# مقدمه

# LabVIEW

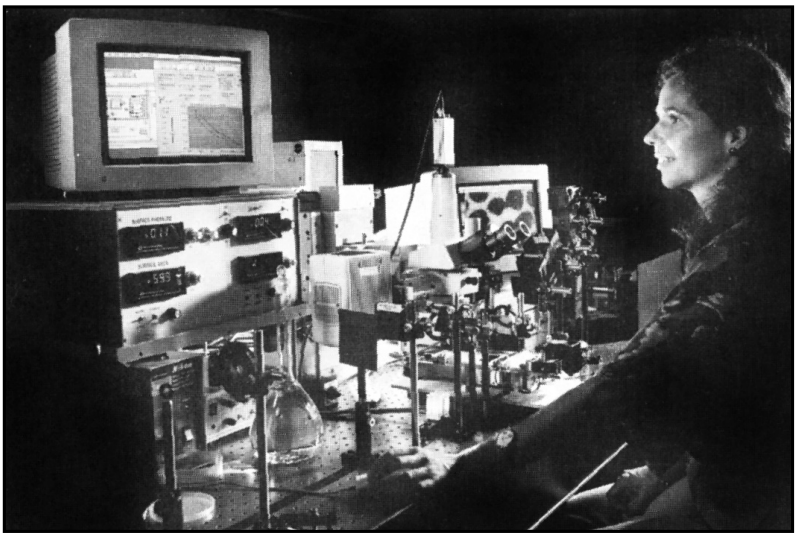
## مقدمه

به دنیای LabVIEW خوش آمدید. نرم‌افزار LabVIEW که مخفف عبارات Laboratory Virtual Instrument Engineering Workbench می‌باشد، یک زبان برنامه‌نویسی گرافیکی است که به صورت گسترده‌ای برای کاربردهای مختلفی در صنایع، تحصیلات، آموزش و تحقیقات آزمایشگاهی به عنوان یک مدل استاندارد برای جمع آوری و پردازش داده‌ها و همچنین وسیله‌ای جهت کنترل و شبیه سازی ابزارهای مجازی در آمده است. این برنامه بر روی تمامی سیستم های Macintosh، Sun SPARC، Hp و رایانه های شخصی تحت نرم‌افزارهای Windows 3.1 و Win95/ NT/ XP قابل اجراست.

این برنامه یک نرم‌افزار قدرتمند و قابل انعطاف جهت تجزیه و تحلیل سیستم های اندازه‌گیری است. همان گونه که می دانید رایانه ها نسبت به ابزارهای اندازه‌گیری استاندارد از قابلیتهای بیشتری برخوردار هستند. ایجاد یک برنامه در محیط LabVIEW بسیار ساده است. ارتباط شهودی و مستقیم کاربری که در حال کار با LabVIEW است نوشتن و استفاده از برنامه های موجود در این بسته‌ی نرم‌افزاری را بسیار جالب و سرگرم کننده می سازد.

عملکرد نرم‌افزار LabVIEW کاملاً از طبیعت ترتیبی و زنجیره‌ای موجود در زبانهای برنامه‌نویسی متنی متداول و مرسوم مجزاست و یک محیط گرافیکی را برای کاربر فراهم ساخته است. در این راه از تمامی ابزارهای لازم جهت جمع آوری، پردازش و تحلیل داده ها و نمایش نتایج استفاده می شود. به کمک این زبان برنامه نویسی گرافیکی که با «G» نشان داده می شود، در برنامه‌ی نوشته شده از یک نمودار بلوکی استفاده

می شود و سپس این نمودار به کدهای ماشین تبدیل می گردد. این نرم افزار برای موارد بیشماری از کاربردهای علمی و مهندسی، ایده آل و عملی است و به شما کمک می کند تا مسائل و مشکلات موجود در برنامه نویسی را در مدت زمان کوتاهی حل کنید.



**دکتر ویولا وُگال در حال بررسی و تحقیق در مورد ساختار و عملکرد غشای نیمه تراوا با استفاده از نرم افزار LabVIEW است. در این پروژه از یک سیستم Mac، یک میکروسکوپ فلوئورسنت و یک سیستم لیزری بسیار قوی استفاده می شود (عکس از دانشگاه واشنگتن).**

مقدمه

## مطالبی در مورد LabVIEW

نرم افزار LabVIEW آن چنان راه خود را در طیف وسیعی از کاربردها و برنامه های «ابزار مجازی» باز کرده است که به سختی می توان دریافت که به راستی این مسیر از کجا آغاز شده است. همان گونه که از نام این نرم افزار استنباط می شود، این برنامه اولین بار از آزمایشگاه نشأت گرفته و هنوز هم در آزمایشگاههای مختلف در زمینه های تحقیق و توسعه، پرطرفدار باقی مانده است. کاربرد این نرم افزار پرانعطاف به آزمایشگاهها خاتمه نمی یابد. رد پای این برنامه را می توان در کلاس های درسی دانشگاههای سراسر دنیا به خصوص در رشته‌های مختلفی از جمله مهندسی برق، مکانیک و فیزیک کاربردی یافت.

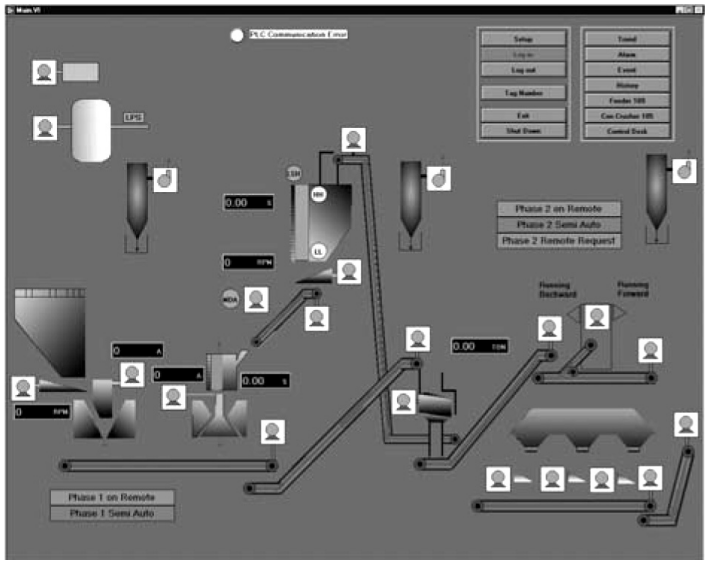
گستردگی LabVIEW در زمینه‌ی آزمایشگاهها دارای ابعاد مختلفی است. سیستم های «ابزار مجازی» نظیر LabVIEW از لحاظ هزینه‌ی پایین آنها در زمینه‌ی سخت افزار و همچنین قابلیت انعطاف بالای آنها در زمینه‌ی نرم افزار کاملاً معروف و شناخته شده‌اند. بنابراین جای تعجب نیست که این سیستم ها تا به این حد عمومی و پرطرفدار هستند.

شاید بهترین دلیل برای گسترش کاربرد برنامه‌ی LabVIEW، عمومی و فراگیربودن آن است. به عنوان مثال در صنایع گوناگون در مواردی که باید اندازه گیری هایی از قبیل دما انجام گیرد، می توان از این نرم افزار استفاده نمود. این دما ممکن است دمای یک کوره، یک سیستم سردکننده مانند یخچال، یک محیط گلخانه ای، یک اتاق

و یا یک دیگ سوپ باشد. علاوه بر دما می توان موارد دیگری نظیر فشار، نیرو، جابه‌جایی، کشش، pH و... را نام برد. از رایانه های شخصی به همراه نرم افزار LabVIEW می توان ابزارهای اندازه گیری حقیقی را در محل های گوناگون به صورت مجازی شبیه سازی کرد. با استفاده از LabVIEW نه‌تنها اندازه گیری ها بسیار ساده تر انجام می گیرد، بلکه قادر خواهید بود کمیت اندازه گیری شده را تجزیه وتحلیل نموده، آن را به نمایش درآورید و یا در صورت تمایل، آن را از طریق رایانه به هر نقطه از جهان ارسال نمایید.

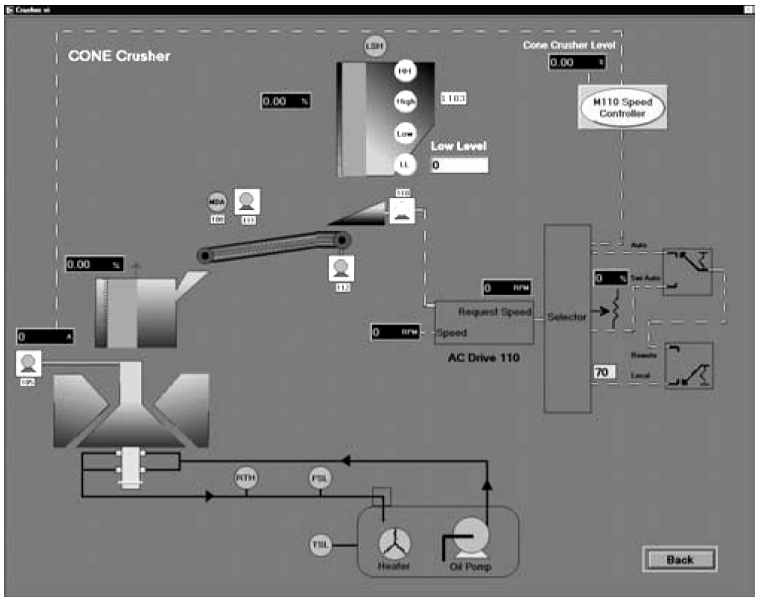
نرم‌افزار LabVIEW را می توان در حقیقت یک کاتالیست فرض نمود که بین رایانه و کمیتهای اندازه‌گیری شونده ارتباط برقرار می کند. مرحله‌ی بعدی پس از اندازه گیری و تحلیل هر کمیت، ایجاد تغییر و یا کنترل آن بر اساس نتایج به‌دست آمده و یا خواسته های ما از سیستم است. به عنوان مثال پس از اندازه گیری دما لازم است که برای کنترل درجه‌ی حرارت محیط، یک سیستم گرم کننده یا سردکننده را در محیط فعال نمود. در این حالت نرم‌افزار LabVIEW باز به میدان می آید و انجام این عمل را ساده‌تر می سازد. نظارت و کنترل فرآیند نیز از توانمندی ها و قابلیتهای این نرم‌افزار است. در برخی موارد این بسته‌ی نرم‌افزاری به طور مستقیم وظیفه‌ی نظارت وکنترل فرآیند را بر عهده دارد. در برخی موارد نیز می توان از کنترل کننده های منطقی برنامه‌پذیر یا سیستم های PLC[1] استفاده نمود که در این حالت سیستم به‌دست آمده را SCADA[2] گویند.

درشکلهای زیر چند صفحه از سیستم HMI در یک کارخانه که با کمک نرم‌افزار LabVIEW ساخته شده، نشان داده شده است. این سیستم با PLC 500 کنترونیک به عنوان کنترلر اصلی در ارتباط می‌باشد و اطلاعات دریافتی از آن را در صفحات به روز می‌رساند و از طرفی فرمانهای اپراتوری را نیز به سیستم کنترل ارسال می‌کند.

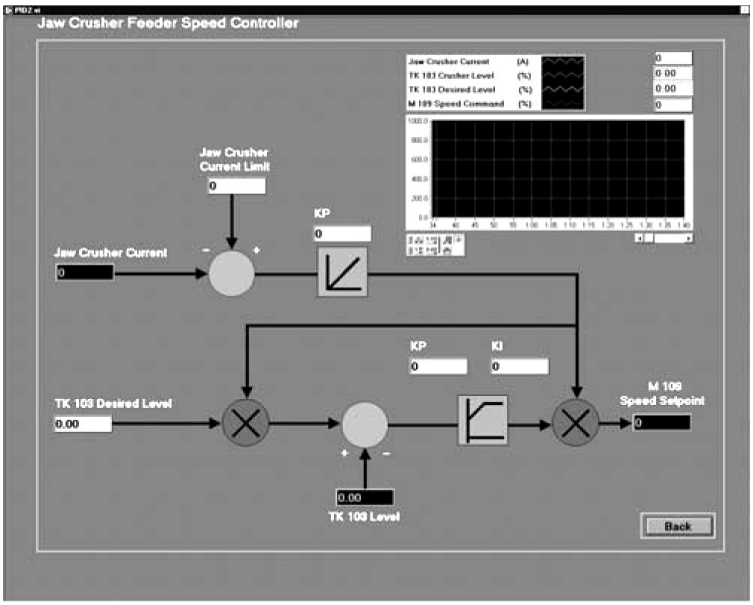



**صفحه‌ی اصلی شمای میمیک فرآیند: در این صفحه گردش مواد در طول خط و در گذر از ماشین‌های مختلف نشان داده می‌شود. وضعیت هر یک از ماشین‌ها به صورت تصویری در اختیار اپراتور قرار می گیرد.**

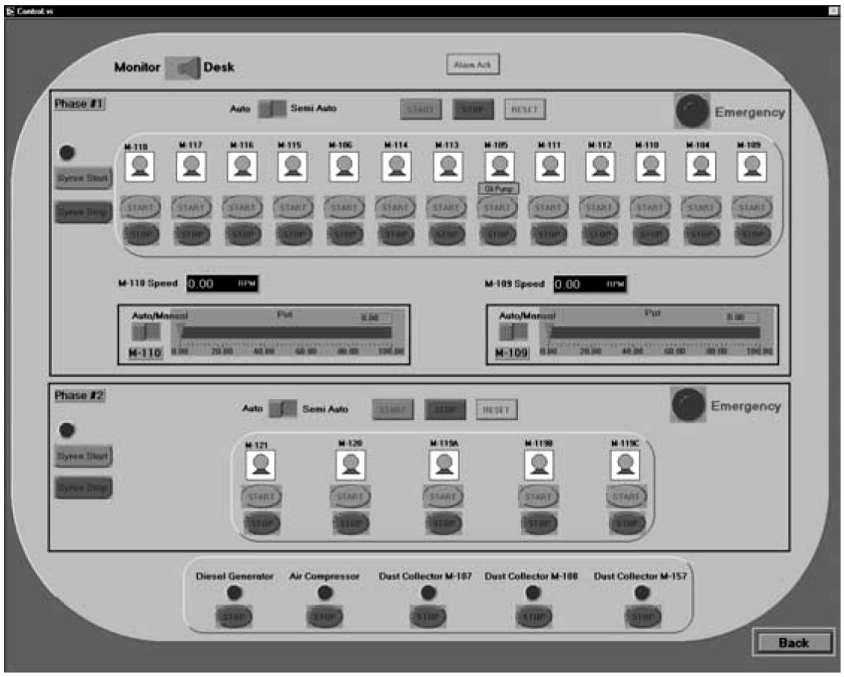
1- Programmable Logic Controller
2- Supervisory Control And Data Acquisition



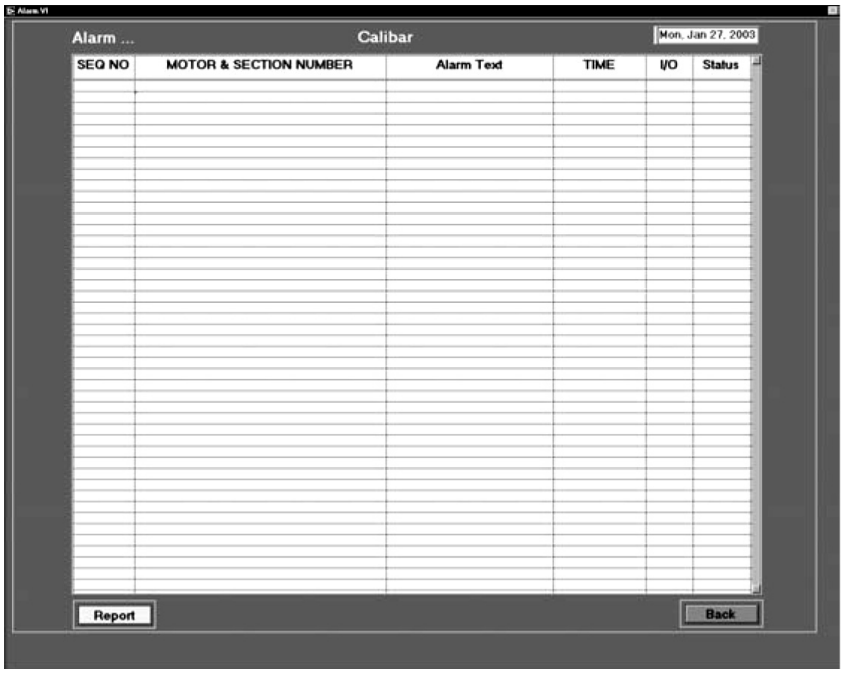

**صفحه‌ی فرعی نمای جزئیات: این صفحه نگاه نزدیکتری به یکی از بخشهای خط دارد تا اپراتور بتواند از نزدیک تغییرات را مشاهده و ارزیابی کند.**




**صفحه‌ی فرعی برای تنظیم پارامترهای کنترل و مشاهده‌ی TREND: دراین صفحه پارامترهای لازم برای کنترل خط تنظیم می‌شوند و نتیجه بر روی منحنی تولید به نمایش در می‌آید.**




**صفحه‌ی فرعی وضعیتها و اعمال فرمانها: دراین صفحه اپراتور بدون نیاز به میز کنترل به‌طور مستقیم از طریق Mouse فرمانهای لازم را برای روشن و خاموش کردن قسمتهای مختلف اعمال می‌کند.**




**صفحه‌ی فرعی ALARMها: در این صفحه کلیه‌ی وقایع اضطراری که در خط اتفاق می‌افتد ضبط می شوند تا در هنگام ضرورت به آن مراجعه شود و مشکلات تولید برطرف گردد .**

در ادامه، تعدادی از کاربردهای این بسته‌ی نرم‌افزاری را ملاحظه می کنید:

- شبیه سازی فعالیت و عملکرد قلب.
- کنترل فرآیند تهیه و بسته بندی مواد غذایی.

- تشخیص وجود نشتی گاز هیدروژن در شاتل های فضایی.
- نظارت بر رژیم غذایی کودکان و نوزادان.
- مدل سازی سیستم های قدرت جهت تجزیه و تحلیل و محاسبه‌ی ضریب قدرت.
- اندازه گیری اثرات فیزیکی آزمایشها بر روی موشهای آزمایشگاهی.
- کنترل موتورهای پله‌ای و سروُ.
- انجام آزمایش بر روی کارت ها و مدارات الکترونیکی در رایانه ها و دیگر وسایل الکترونیکی برای اطمینان از صحت عملکرد آنها.
- کنترل سیستم تهویه مطبوع [3].

## هدف نهایی این کتاب

کتاب «راهنمای جامع LabVIEW» شما را در فراگیری و به‌کارگیری این نرم‌افزار یاری می‌دهد و به شما کمک می‌کند تا محیط برنامه‌نویسی «G» را به راحتی درک کنید. این مجموعه حاوی مثالها و تمرینات اضافی برای معرفی تکنیک‌ها و روشهای کاربردی است و همچنین منابع دیگری را جهت به‌دست آوردن اطلاعات در مورد این بسته‌ی نرم‌افزاری در اختیار خوانندگان قرار می‌دهد. به همراه این کتاب یک لوح فشرده عرضه می‌گردد که به کمک آن می‌توانید برنامه های متنوعی را مشاهده نمایید و پس از ایجاد تغییرات لازم، از آنها برای برنامه های کاربردی و عملی خود استفاده کنید.

در ارائه‌ی توضیح مطالب این کتاب فرض بر این است که خواننده در مورد سیستم عامل رایانه‌ی خود از اطلاعات کافی برخوردار است. در صورتی که در این زمینه تجربه‌ای ندارید، در مورد سیستم عامل رایانه‌ی خود اندکی مطالعه کنید. به عنوان مثال مطالبی از قبیل نحوه‌ی دستیابی به منوها، باز کردن و ذخیره نمودن فایل ها، تهیه‌ی نسخه‌ی پشتیبان و استفاده از ماوس را به خوبی تمرین کنید.

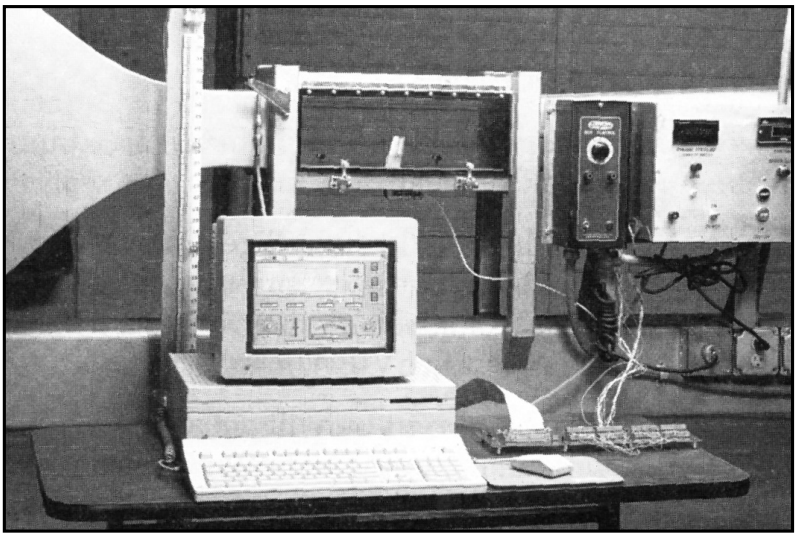
پس از مطالعه‌ی این کتاب و اجرای تمرینات، قادر خواهید بود تا موارد زیر را به آسانی انجام دهید:

- نوشتن برنامه های LabVIEW که در اصطلاح به آنها «ابزار مجازی» یا به اختصار VI گویند.
- به کارگیری تکنیک ها و روشهای مفید و عملی در زمینه‌ی تغییر و اصلاح برنامه.
- اجرای برنامه‌ها و دستورهای داخلی LabVIEW و همچنین برنامه‌های موجود در کتابخانه‌ها.
- ایجاد و ذخیره‌ی برنامه هایی که خود شما نوشته اید به صورتی که بتوانید از آنها به عنوان زیربرنامه یا SubVI نیز استفاده کنید.
- طراحی رابط گرافیکی کاربر[4].
- ذخیره‌ی داده ها در یک فایل و به نمایش درآوردن آنها بر روی گراف و نمودار.



3- HVAC: Heat Ventilation and Air Conditioning
4-GUI: Graphical User Interface

- ایجاد برنامه هایی که از کارت های DAQ[5] و GPIB[6] و یا درگاههای سریال رایانه استفاده می کنند.
- استفاده از توابع و دستورهای تحلیلی برای پردازش داده ها.
- بهینه سازی سرعت و عملکرد برنامه های نوشته شده در محیط LabVIEW.
- به کارگیری تکنیک ها و قابلیتهای پیشرفته نظیر متغیرهای محلی[7]، سراسری[8] و گره های خصوصیت[9].
- استفاده از نرم افزار LabVIEW برای اندازه گیری و کاربردهای آزمایشگاهی.
- افزودن نرم‌افزارهای دیگری به برنامه‌ی LabVIEW و نظارت بر کنترل فرآیندها.



**نرم‌افزار LabVIEW از قابلیتهای گرافیکی و تحلیلی بسیار بالایی برخوردار است. به عنوان مثال می‌توان جهت کنترل و نظارت بر عملکرد و اجرای فرآیندهای مختلف از این ویژگیها استفاده نمود. در این شکل برای کنترل و نظارت بر آزمایش «تونل باد» از قابلیتهای این نرم‌افزار استفاده شده است.**

## نحوه‌ی ارائه‌ی مطالب در این مجموعه

کتاب «راهنمای جامع LabVIEW» به شما کمک می کند تا به راحتی و با سرعت، برنامه‌نویسی در محیط LabVIEW را فراگیرید و به تجزیه و تحلیل برنامه های کاربردی بپردازید. این کتاب به دو بخش مقدماتی و پیشرفته تقسیم می شود. بخش مقدماتی شامل ۹ فصل است و اصول برنامه‌نویسی «G» در نرم‌افزار LabVIEW را آموزش می دهد. بخش پیشرفته مشتمل بر ۶ فصل است که مهارت شما را در برنامه‌نویسی ارتقا می‌دهد و تکنیک های مفید و روشهای بهینه سازی برنامه را معرفی می کند. به شما پیشنهاد می کنیم که مطالعه‌ی این

5- Data Acquisition
6- General Purpose Interface Bus
7- Local Variables
8- Global Variables
9- Attribute Nodes

کتاب را از بخش مقدماتی شروع کنید و پس از فراگیری مطالب اساسی به بخش پیشرفته مراجعه نمایید. در هر دو بخش مقدماتی و پیشرفته جهت سهولت در آموزش مطالب، بخشهای زیر در هر یک از فصول در نظر گرفته شده است:

- مقدمه: این بخش به معرفی اهداف و اصطلاحات کلیدی استفاده شده در هر فصل اختصاص داده شده است.
- توضیح: در این قسمت در مورد عناوین موجود در فصل، توضیحات کامل و مفصل ارائه می گردد.
- مثالها و تمرینات: این تمرینها براساس اطلاعات ارائه شده در همان فصل طراحی شده‌اند.
- خلاصه: در این بخش، مفاهیم مهم و اصلی و همچنین مهارت های آموزش داده شده در آن فصل به صورت خلاصه بازگو می شود.
- تمرینات اضافی: در بسیاری از فصول، انجام این تمرینات برای فعالیت بیشتر پیشنهاد شده است.

## بخش مقدماتی

در فصل ۱ به بررسی چگونگی عملکرد نرم‌افزار LabVIEW می پردازیم و برخی از ویژگیها و قابلیتهای آن را معرفی می کنیم.

فصل ۲ شامل مواردی نظیر روش برقراری ارتباط از طریق درگاههای سریال، کارت های GPIB، DAQ و همچنین تجزیه و تحلیل داده‌ها به کمک LabVIEW می باشد. در این فصل مطالبی در مورد تاریخچه‌ی این نرم‌افزار نیز ارائه می شود.

در فصل ۳ با محیط برنامه‌نویسی LabVIEW آشنا خواهید شد. در این فصل بخشهای اصلی یک برنامه، پنجره‌ی Help، منوها، ابزارها، پالت ها و زیربرنامه ها مورد بررسی قرار می گیرند.

در دو فصل ۴ و ۵ با اصول برنامه‌نویسی در محیط LabVIEW آشنا می شوید. در این دو فصل مفاهیمی از قبیل روش به کارگیری کنترل ها[10]، نشان دهنده ها[11]، اعداد ثابت[12]، جبری[13] و رشته ای[14]، نحوه‌ی سیم کشی[15] و برقرار نمودن ارتباط، ایجاد تغییر و اصلاح، ذخیره‌ی برنامه، ایجاد زیربرنامه و مستندسازی[16] برنامه های نوشته شده مورد بحث و بررسی قرار می گیرند. در این فصول درمی یابید که چرا «G» به عنوان یک زبان برنامه نویسی گرافیکی در نظر گرفته می شود.

در فصل ۶ به ارائه‌ی توضیح در مورد ساختار[17]های برنامه‌نویسی در محیط LabVIEW خواهیم پرداخت.

10- Controls
11- Indicators
12- Constants
13- Booleans
14- Strings
15- Wiring
16- Documentation
17- Structure

ساختارهایی نظیر حلقه[۱۸]، ثبات انتقال[۱۹]، ساختار شرطی[۲۰]، ساختار ترتیبی[۲۱] وگره‌ی فرمولی[۲۲] در این فصل بررسی می شوند و نحوه‌ی کنترل زمان اجرای برنامه ها را نیز فرا خواهید گرفت.

در فصل ۷ مطالبی در مورد روش استفاده از دو ساختار مهم یعنی آرایه و کلاستر ارائه می گردد. در این فصل همچنین با توابع از پیش نوشته شده در LabVIEW در مورد آرایه ها و کلاسترها آشنا خواهید شد.

در فصل ۸ به بررسی گراف ها و نمودارهای گوناگون پرداخته، نحوه‌ی به کارگیری آنها را جهت نمایش داده ها بررسی می کنیم.

در فصل ۹ به بررسی انواع داده های رشته ای و جداول خواهیم پرداخت. در این فصل همچنین روش ذخیره‌ی داده ها و خواندن آنها از یک فایل به کمک دستورهای File I/O در LabVIEW مورد بحث و بررسی قرار می گیرد.

## بخش پیشرفته

در فصل ۱۰ توضیحات بیشتری در مورد درگاه سریال، کارت های GPIB و DAQ ارائه می شود. در این فصل در مورد سخت افزار و همچنین نحوه‌ی نصب نرم افزار پیکربندی کارت های DAQ اطلاعاتی به دست می آورید.

در فصل ۱۱ در مورد نحوه‌ی پردازش داده ها در محیط LabVIEW به کمک کارت های DAQ، GPIB و درگاههای سریال توضیحاتی ارائه می گردد.

فصل ۱۲ به بررسی برخی از ویژگیهای LabVIEW نظیر متغیرهای محلی، سراسری، گره‌ی خصوصیت و نحوه‌ی تبدیل داده ها اختصاص یافته است.

در فصل ۱۳ روش پیکربندی و بررسی عملکرد برنامه و نحوه‌ی دستیابی به المان های موجود در صفحه‌ی پانل با استفاده از صفحه کلید مورد بررسی قرار می گیرد. در این فصل همچنین با دستور بسیار مفید و کاربردی Find و پنجره‌ی Profile نیز آشنا خواهید شد.

در فصل ۱۴ مواردی از قبیل کاربردهای I/O، نحوه‌ی چاپ برنامه و روش برقراری ارتباط بین چند رایانه از طریق شبکه را مورد بررسی قرار خواهیم داد.

در فصل ۱۵ با روش ایجاد پنجره‌ی Help، سفارشی نمودن[۲۳] برنامه با وارد نمودن تصاویر خارجی از محیطهای گرافیکی دیگر آشنا خواهید شد.

18- Loop
19- Shift Register
20- Case Structure
21- Sequence Structure
22- Formula Node
23- Customize

**یک تیم تحقیقاتی در حال بررسی نیروهایی است که در حین ضربه زدن و بازی گلف به بازیکن وارد می‌آید. تمامی نیروهای مذکور و اندازه‌گیری‌های انجام شده به کمک نرم‌افزار LabVIEW و یک سیستم PC-Based ثبت و تحلیل می‌شوند.**

## قراردادهای استفاده شده در این کتاب

| | |
|---|---|
| < > | این براکت‌ها دربرگیرنده‌ی نام کلیدهای صفحه کلید است. به عنوان مثال: کلید <Shift> |
| | این آیکن مواردی را مشخص می‌کند که باید به آنها توجه خاص داشته باشید. |
| | مراقب باشید! این آیکن به شما اخطار می‌دهد که یک خطر یا دام که عموماً کاربران در آن گرفتار می‌شوند در پیش پای شماست و یا مشخص‌کننده‌ی مواردی است که باید از آن برحذر باشید. |
| | این آیکن از شما می‌خواهد تا به راهنمایی‌های مفید و کاربردی در جهت انجام اعمال و وظایف مورد نظر توجه کنید. |

اطلاعات موجود در این بخش منحصر به محیط Mac است.

اطلاعات موجود در این بخش منحصر به محیط Windows است.

نکته‌ای در مورد مسیرها: در سیستم‌های گوناگون از قراردادهای مختلفی جهت تعیین مسیر استفاده می‌گردد. به عنوان مثال مسیر استفاده شده در محیط Windows به صورت C:\LabVIEW\Mine.llb\Bingo.vi است. در صورتی که همین مسیر در محیط Mac به صورت Hard Drive Name:LabVIEW:Mine.llb:Bingo.vi می‌باشد. در این کتاب تنها به روش تعیین مسیر در سیستم عامل Windows اشاره می‌کنیم. در صورت استفاده از سیستم Mac، کاراکتر «\» را با «:» جایگزین کنید.

## توضیحاتی در مورد لوح فشرده‌ی همراه این کتاب

به همراه این کتاب یک لوح فشرده نیز عرضه می‌گردد. این لوح حاوی نسخه‌ی LabVIEW 5.0 و تعدادی از مثالهای کاربردی و جالب برای استفاده‌ی کاربران است. در لوح مذکور همچنین نرم‌افزارهایی جهت پیکربندی کارت‌های DAQ و GPIB وجود دارد.

به دلیل اینکه برخی از مطالب مفید مندرج در این لوح در فایل‌هایی با پسوند .pdf ذخیره شده و برای مطالعه‌ی این مطالب استفاده از نرم‌افزار Acrobat Reader الزامی است، بنابراین نسخه‌ی پنجم این نرم‌افزار یعنی Acrobat Reader 5.0 نیز در این لوح فشرده گنجانده شده است.

## روش نصب نرم‌افزار LabVIEW 5.0

به دلیل اینکه در کشور ما اکثر کاربران از سیستم عامل Windows استفاده می‌کنند، در این کتاب تنها روش نصب نرم‌افزار در این محیط را توضیح خواهیم داد.

حداقل مشخصات سخت‌افزاری لازم جهت نصب نرم‌افزار LabVIEW 5.0 عبارت است از:

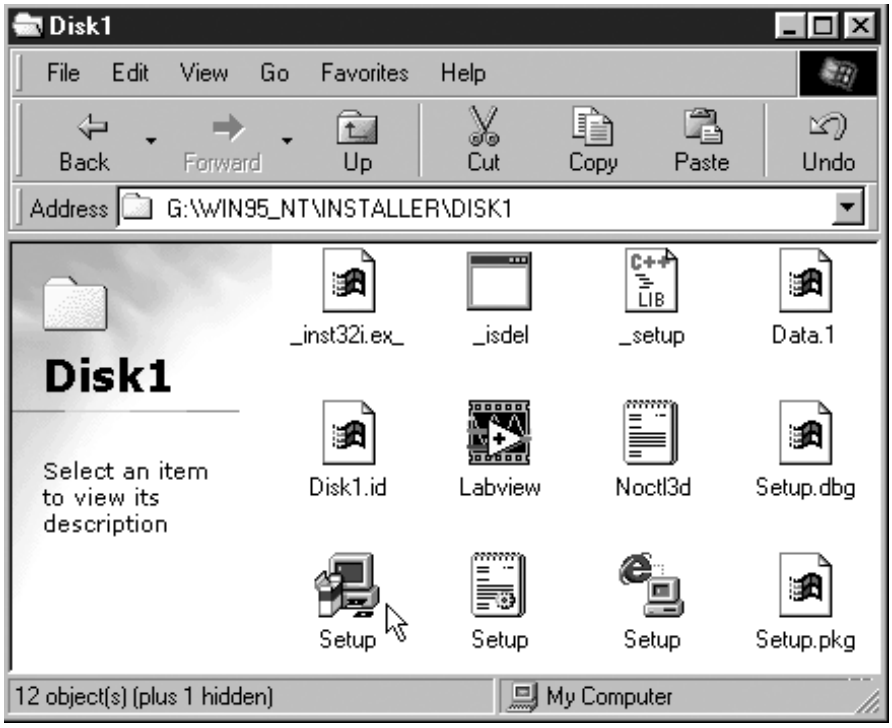
برای نصب این برنامه تحت نرم‌افزار Windows 3.1 رایانه‌ی شما باید دارای ۸ مگابایت حافظه‌ی RAM باشد. نرم‌افزار LabVIEW بر روی سیستم‌های 80386 قابل اجراست. اما پیشنهاد ما بر این است که از یک سیستم 80486 به بالا استفاده کنید.

نسخه‌ی مخصوص Windows 95/ XP تحت هر سیستمی که محیط Windows 95/ XP را پشتیبانی کند قابل اجراست و نیاز به حداقل ۸ تا ۱۲ مگابایت حافظه‌ی RAM دارد.

نسخه‌ی مخصوص Windows NT تحت نسخه‌ی 3.5.1 یا بالاتر اجرا می‌شود. برای اجرای این نرم‌افزار در این حالت نیاز به ۱۲ تا ۱۶ مگابایت حافظه‌ی RAM می‌باشد.

در تمامی موارد فوق حداقل ۷۰ مگابایت از ظرفیت دیسک سخت لازم است[۲۴].

جهت نصب این نرم‌افزار در محیط Windows 95/ NT/ XP کافی است فهرست‌های موجود در لوح فشرده را به ترتیب باز کنید تا به فایل Setup دست یابید. به عنوان مثال اگر لوح فشرده در درایو G قرار گرفته باشد فایل مذکور در مسیر G:\WIN95-NT\INSTALLER\DISK1 قرار دارد. این مسیر در شکل زیر نشان داده شده است.






۲۴- برای نصب بسته‌ی نرم‌افزاری LabVIEW به همراه نرم‌افزارهای پیکربندی کارت‌های DAQ و GPIB در حدود ۲۰۰ مگابایت از حافظه‌ی دیسک سخت مورد نیاز است.

در هنگام نصب نرم‌افزار LabVIEW یک پنجره‌ی محاوره‌ای باز می‌شود و شماره سریال لازم جهت این نرم افزار را از کاربر درخواست می کند. شماره سریال لازم جهت نصب این نرم‌افزار G10X87651 است.

پس از نصب این نرم‌افزار، فهرست Exercise را که حاوی پاسخ تمامی تمرینات موجود در کتاب است، از لوح فشرده در فهرست LabVIEW کپی کنید و پس از انجام هر تمرین، پاسخ خود را با جواب صحیح آن مقایسه نمایید. در ضمن حتماً برنامه های تحلیلگر پیشرفته را از مسیر G:\ANALYSIS\WIN95-NT\DISK1 بر روی سیستم خود نصب کنید[۲۵].



۲۵- در صورت عدم نصب این بخش از نرم‌افزار، زیرپالت Functions >> Analysis به صورت ناقص و با عنوان Functions >> Base Analysis ظاهر می‌شود و در مورد به کارگیری برنامه های موجود در آن با مشکل مواجه خواهید شد.

LabVIEW

# ۱

# شناخت بسته‌ی نرم‌افزاری LabVIEW

## در این فصل با موارد زیر آشنا می‌شوید:

- محیط برنامه‌نویسی LabVIEW
- مفهوم زبان برنامه‌نویسی گرافیکی و اصل جریانِ داده
- مثالهای مقدماتی و کاربردی

# LabVIEW

# ۱

## شناخت بسته‌ی نرم‌افزاری LabVIEW

در این فصل اصول عملکرد و قابلیتهای نرم‌افزار LabVIEW را مورد بررسی قرار می‌دهیم و سعی می کنیم با این محیط برنامه نویسی گرافیکی آشنا شویم.

در این فصل اهداف زیر دنبال می شوند:

- شناخت واقعی نرم‌افزار LabVIEW.
- درک مفهوم زبان برنامه نویسی گرافیکی و اصل جریانِ داده.
- بررسی مثالهای مقدماتی.
- شناخت محیط برنامه نویسی LabVIEW.

اصطلاحات کلیدی[1] موجود در این فصل عبارتند از:

- LabVIEW
- زبان برنامه نویسی «G»
- ابزار مجازی (VI: Virtual Instrument)
- جریانِ داده (Data Flow)
- زبان برنامه نویسی گرافیکی (Graphical Programming)
- صفحه‌ی پانل (Front Panel)

---

1- Key Terms

- نمودار بلوکی (Block Diagram)
- آیکن (Icon)
- کانکتور (Connector)
- نوار ابزار (Toolbar)
- سلسله مراتب (Hierarchy)

## محیط برنامه‌نویسی LabVIEW

شاید بسیار علاقه‌مند باشید که قبل از شروع به کار با نرم‌افزار LabVIEW به اهمیت این نرم‌افزار پی ببرید و دریابید که به کمک این برنامه می‌توانید چه اعمالی انجام دهید و متقابلاً این نرم‌افزار چه کارهایی را برای شما انجام می‌دهد؟

LabVIEW مخفف عبارات Laboratory Virtual Instrument Engineering Workbench می‌باشد و یک محیط برنامه‌نویسی است که به کمک آن قادر خواهید بود تا برنامه‌ی مورد نظر را به صورت گرافیکی ایجاد نمایید. با توجه به این قابلیت، تفاوت این نرم‌افزار با سایر زبانهای برنامه‌نویسی متداول نظیر BASIC و C که در آنها برنامه به صورت متنی نوشته می‌شود، مشخص می‌گردد. با این وجود نرم‌افزار LabVIEW فراتر از یک زبان برنامه‌نویسی است. این بسته‌ی نرم‌افزاری، برنامه‌ای پیشرفته و قابل اجرا برای استفاده‌ی محققان، مهندسان و کارشناسانی است که طراحی و برنامه‌نویسی سیستم‌ها را برعهده دارند. این برنامه در رایانه‌های شخصی تحت نرم‌افزار Windows و همچنین در سیستم‌های Sun SPARC، Mac و HP قابل اجراست.

با استفاده از زبان برنامه‌نویسی قدرتمند «G» و با کمک LabVIEW می‌توانید استعداد برنامه‌نویسی خود را شکوفا کنید. این نرم‌افزار به گونه‌ای طراحی شده است که قابلیت اندازه‌گیری، تجزیه و تحلیل داده‌ها و نمایش نتایج را یک‌جا و با هم برای کاربر به همراه دارد. بنابراین نوشتن برنامه‌هایی که با استفاده از زبانهای برنامه‌نویسی رایج و متداول در عرض چندین هفته و یا چند ماه انجام می‌گیرد، به کمک LabVIEW در عرض چند ساعت به پایان می‌رسد. همچنین به دلیل اینکه این بسته‌ی نرم‌افزاری قابلیت برقراری ارتباط گرافیکی چندجانبه با کاربر را داراست، برنامه‌نویسی در این محیط بسیار ساده است. این نرم‌افزار برای مقاصد شبیه‌سازی، به ظهور رساندن ایده‌ها، برنامه‌نویسی عادی و حتی تدریس و تفهیم مفاهیم اساسی برنامه‌نویسی کاملاً ایده‌آل است.

به دلیل اینکه برنامه‌ی نوشته شده در این محیط بر پایه‌ی نرم‌افزار استوار است، می‌توان از قابلیتهای نرم‌افزاری آن استفاده نمود. به همین دلیل نسبت به تجهیزات استاندارد و آزمایشگاهی از قابلیت انعطاف و کاربرد بیشتری برخوردار است. باکمک LabVIEW قادر خواهید بود تا دقیقاً تجهیزات اندازه‌گیری مورد نیاز خود را به صورت مجازی ایجاد کنید. این عمل مطمئناً از لحاظ هزینه به مراتب کمتر از هزینه‌های مربوط به خرید، تجهیز و نصب وسایل اندازه‌گیری حقیقی خواهد بود. در صورتی که به تجهیزات اندازه‌گیری دیگری

۱

نیاز باشد به کمک این نرم‌افزار می‌توانید در عرض چند دقیقه برنامه‌ها را تغییر دهید و آنها را اصلاح نمایید. بنابراین استفاده نمودن از LabVIEW، منجر به صرفه‌جویی در زمان صرف شده برای برنامه‌نویسی و هزینه‌ی تهیه‌ی لوازم اندازه‌گیری و وسایل آزمایشگاهی می‌گردد. در شکل ۱-۱ نمونه‌ای از کاربرد این نرم‌افزار را ملاحظه می‌کنید.

**شکل ۱-۱: در تحقیقات فضایی جهت تحقیق و مطالعه‌ی ابررساناها در دمای بالا و در یک محیط فاقد جاذبه از کوره‌ی شناور در شاتل‌های سازمان فضایی NASA استفاده می‌شود. با استفاده از نرم‌افزار LabVIEW به راحتی می‌توان این فرآیند را کنترل نمود.**

در نرم‌افزار LabVIEW کتابخانه‌هایی حاوی توابع و دستورهای عملیاتی و زیربرنامه‌های اجرایی وجود دارند که به کاربر کمک می‌کنند تا برنامه‌نویسی را بدون روبه‌رو شدن با مشکلات موجود در اکثر زبانهای برنامه‌نویسی متنی به آسانی دنبال کند. این برنامه همچنین دارای کتابخانه‌هایی شامل ابزارهای جانبی جهت جمع‌آوری داده‌ها، استفاده از کارت‌های GPIB و درگاه سریال، تجزیه و تحلیل، نمایش و ذخیره‌ی داده‌هاست. کتابخانه‌ی Analysis حاوی گروهی از توابع و دستورهای مفید و کاربردی است که جهت تولید و پردازش سیگنال‌ها به‌کار برده می‌شوند. توابع و دستورهای دیگری نظیر فیلترها، آمارگیری، رگرسیون، جبر خطی و حساب آرایه‌ای نیز در این کتابخانه وجود دارند.

به دلیل وجود قابلیت گرافیکی در بسته‌ی نرم‌افزاری LabVIEW، طبیعتاً از این برنامه جهت نمایش داده‌ها نیز استفاده می‌شود. در این نرم‌افزار، خروجیها به صور مختلف بر روی صفحه به نمایش در می‌آیند. گراف‌ها، نمودارها و محیطهای گرافیکی تعریف شده توسط کاربر تنها گوشه‌ای از ویژگیها و قابلیتهای موجود به منظور نمایش خروجی می‌باشند. در این کتاب با نحوه‌ی نمایش داده‌ها به هر یک از اشکال مذکور آشنا خواهید شد.

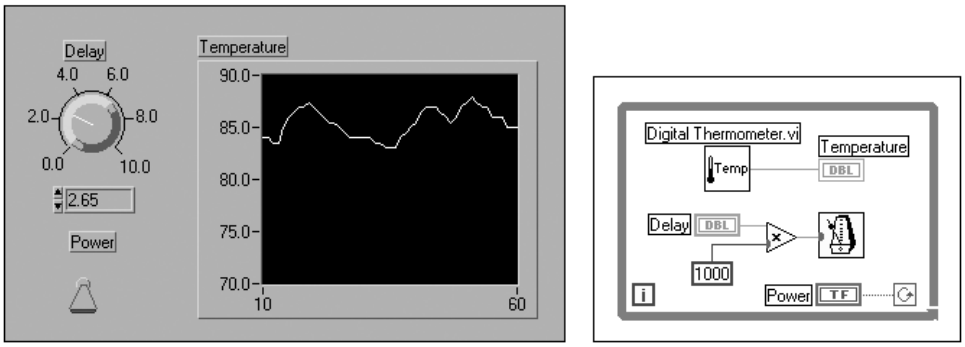
برنامه‌ی نوشته شده در محیط LabVIEW را می‌توان در سیستم[۲]های مختلف استفاده و جابه‌جا نمود.

2- Platform

به عنوان مثال می توانید برنامه‌ای را در یک سیستم Mac نوشته و بدون هیچ گونه تغییری آن را بر روی یک رایانه‌ی شخصی تحت نرم افزار Windows بارگذاری و اجرا نمایید.

## زبان برنامه نویسی «G» و اصل جریانِ داده

محیط برنامه‌نویسی LabVIEW از یک نظر کاملاً با زبانهای برنامه‌نویسی متداول نظیر BASIC و C متفاوت است. در حالی که در دیگر زبانهای برنامه‌نویسی از محیطهای متنی برای نوشتن برنامه استفاده می شود، LabVIEW یک زبان برنامه نویسی گرافیکی موسوم به«G» را به کار می گیرد. در این محیط، برنامه ها به صورت گرافیکی در صفحه‌ای موسوم به نمودار بلوکی ایجاد می شوند. به کمک این روش گرافیکی می توانید عبور و جریان داده ها را در برنامه‌ی نوشته شده ملاحظه نمایید. دو صفحه‌ی رابط کاربر و کدهای گرافیکی مربوط به یک برنامه‌ی کاملاً ساده و ابتدایی در محیط LabVIEW را در شکل ۲-۱ ملاحظه می کنید.




الف: رابط کاربر (User Interface) ب: کدهای گرافیکی (Graphical Code)

شکل ۲-۱

در نرم‌افزار LabVIEW از اصطلاحات فنی، آیکن‌ها و ایده‌های آشنا برای محققان و مهندسان استفاده شده است. برای برنامه‌نویسی در این نرم‌افزار به جای زبان متنی از نمادهای گرافیکی استفاده می‌شود. براساس اصل جریانِ داده، توابع و دستورهای موجود در برنامه، تنها پس از دریافت داده های مهم و ضروری اجرا می شوند.

حتی اگر قبلاً تجربه‌ی برنامه نویسی نداشته باشید به راحتی می توانید برنامه نویسی به کمک LabVIEW را فرا بگیرید. اما در صورتی که از اصول برنامه نویسی آگاهی داشته باشید در فراگیری این نرم افزار موفق‌تر خواهید بود.

## نرم‌افزار LabVIEW چگونه عمل می کند؟

به دلیل اینکه ظاهر و عملکرد برنامه های نوشته شده در محیط LabVIEW از تجهیزات و ابزار حقیقی پیروی می‌نمایند و در واقع این برنامه ها، تجهیزات اندازه گیری را شبیه سازی می کنند، آنها را ابزار مجازی یا VI[۳] نامند. این برنامه ها با توابع،

۳- بخوانید «وی-آی» و این نماد را با عدد VI که معادل با عدد ۶ در سیستم عددنویسی رومی است اشتباه نگیرید.

دستورها، برنامه ها و زیربرنامه های زبانهای برنامه نویسی متنی نظیر BASIC و C مطابقت دارند. از این به بعد به برنامه های نوشته شده در محیط LabVIEW نام VI را اختصاص می دهیم. برنامه های نوشته شده در این محیط با پسوند vi. مشخص می گردند. هر برنامه شامل سه بخش اساسی زیر است:

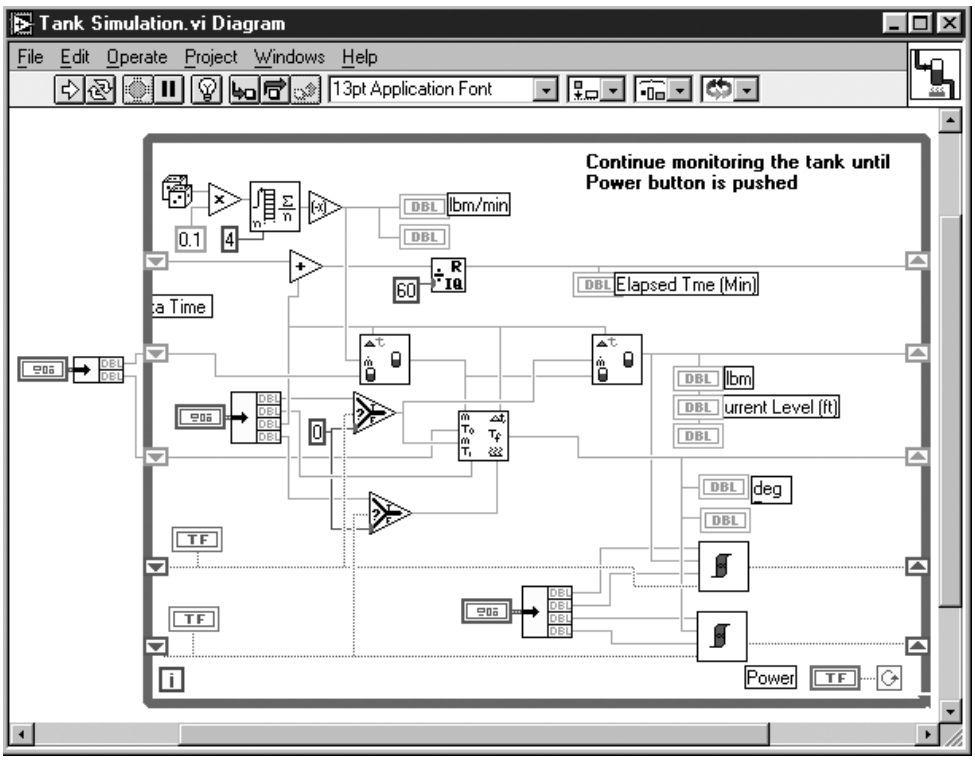
۱- صفحه‌ی پانل یا Front Panel بخشی است که کاربر و نرم‌افزار LabVIEW به راحتی می توانند بایکدیگر ارتباط برقرار کند و بر یکدیگر اثر متقابل داشته باشند. این نام از آن جهت به این صفحه اختصاص داده شده است که در این بخش، پانل یا «صفحه‌ی جلویی» یک سیستم اندازه گیری یا فیزیکی شبیه سازی می گردد. دراین صفحه می توان از کلید ها، دکمه ها و دیگر المان های ورودی که در مجموع به آنها کنترل[4] گوییم و همچنین از المان های خروجی برنامه موسوم به نشان دهنده[5] استفاده نمود. در این صفحه با استفاده از صفحه کلید یا ماوس، داده ها را وارد نموده، نتایج حاصل را بر روی صفحه‌ی نمایشگر رایانه مشاهده می کنیم. در شکل ۳-۱ صفحه‌ی پانل یک برنامه نشان داده شده است.

**شکل ۳-۱: صفحه‌ی پانل در برنامه‌ی Tank Simulation.vi**

۲- صفحه‌ی نمودار بلوکی یا Block Diagram بخش دیگری از برنامه می‌باشد که محل قرار گرفتن کدها و نمادهایی است که زیربنای زبان برنامه‌نویسی گرافیکی «G» هستند. صفحه‌ی نمودار بلوکی در حقیقت بخش اجرایی برنامه است. این صفحه شامل دستورها، توابع، ثوابت و ساختارهاست. در این صفحه جهت نشان دادن عبور داده‌ها و برقراری ارتباط بین عناصر از سیم[6] استفاده می‌شود. هر یک از المان‌ها و عناصر

4- Control
5- Indicator
6- Wire

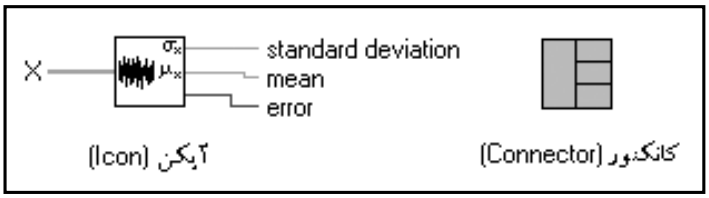
استفاده شده در صفحه‌ی پانل دارای یک ترمینال منحصر به فرد و متناظر بر روی صفحه‌ی نمودار بلوکی است. در شکل ۴-۱ صفحه‌ی نمودار بلوکی یک برنامه را ملاحظه می‌کنید. این صفحه‌ی نمودار بلوکی با صفحه‌ی پانل نشان داده شده در شکل ۳-۱ متناظر است. در هنگام ایجاد هر صفحه‌ی پانل، یک صفحه‌ی نمودار بلوکی متناظر و همنام با آن نیز ایجاد می‌گردد. به نام مندرج در نوار عنوان[7] هر دو صفحه توجه کنید. همان گونه که ملاحظه می‌کنید، عنوان در نظر گرفته شده برای این دو صفحه یکسان است. تنها تفاوت موجود بین عناوین دو صفحه‌ی مذکور در عبارت «Diagram» می‌باشد.




**شکل ۴-۱: صفحه‌ی نمودار بلوکی در برنامه‌ی Tank Simulation.vi**

۳- برای این که بتوانیم یک برنامه را به عنوان زیربرنامه در صفحه‌ی نمودار بلوکی برنامه‌های دیگر به‌کاربریم، باید از آیکن و کانکتور استفاده کنیم. هرگاه یک برنامه در داخل برنامه‌ی دیگری به کار برده شود، در اصطلاح، زیربرنامه یا SubVI خوانده می‌شود. زیربرنامه، نظیر Subroutine در زبانهای برنامه‌نویسی متنی است. آیکن، یک نماد گرافیکی است و به عنوان یک عنصر در صفحه‌ی نمودار بلوکی برنامه‌ی سطح بالاتر مورد استفاده قرار می‌گیرد. در صورتی که یک برنامه به عنوان زیربرنامه به کار برده شود، جهت برقراری ارتباط داده‌ها بین صفحه‌ی نمودار بلوکی و برنامه‌ی اصلی یا برنامه‌ی سطح بالاتر، ازکانکتور استفاده می‌شود. همانند پارامترهای Subroutine، کانکتور نیز معرف ورودیها و خروجیهای برنامه است. در شکل ۵-۱ نمای یک زوج آیکن/کانکتور را ملاحظه می‌کنید.

7- Title Bar




**شکل ۵-۱: تصویر یک زوج آیکن/کانکتور در یک برنامه**

VIها همگی از اصل سلسله مراتب[8] پیروی می‌کنند. بنابراین می‌توانید آنها را طبق اصل سلسله مراتب به عنوان برنامه‌ی اصلی و یا به صورت زیربرنامه استفاده کنید. نوشتن برنامه بدین روش را برنامه‌نویسی مدولار[9] گویند.

بنابراین برای نوشتن یک برنامه کافی است ابتدا آن را به تعدادی زیربرنامه تقسیم نمایید. سپس برای هر یک از زیربرنامه‌ها یک برنامه بنویسید و با ترکیب زیربرنامه‌ها و قراردادن آنها در صفحه‌ی نمودار بلوکی برنامه‌ی سطح بالاتر، VI یا برنامه‌ی اصلی را تکمیل کنید.

برنامه‌نویسی مدولار دارای مزایای بسیاری است. زیرا شما می‌توانید هر یک از زیربرنامه‌ها را به تنهایی اجرا کنید. این قابلیت باعث تسهیل در رفع اشکال برنامه نیز می‌گردد. علاوه بر این بسیاری از زیربرنامه‌های قرار گرفته در سطوح پایین‌تر اغلب انجام وظایفی را با چند برنامه‌ی کاربردی[10] دیگر به صورت مشترک برعهده می‌گیرند یا می‌توانند به صورت مستقل با هر یک از برنامه‌های کاربردی مورد استفاده قرار گیرند.

در جدول ۱-۱ تعدادی از اصطلاحات نرم‌افزار LabVIEW به همراه اصطلاحات معادل آنها در زبانهای برنامه‌نویسی متنی متداول آورده شده است.

**جدول ۱-۱: اصطلاحات نرم‌افزار LabVIEW به همراه اصطلاحات معادل آنها در زبانهای برنامه‌نویسی متداول**

| LabVIEW | زبانهای برنامه‌نویسی متداول |
|---|---|
| VI | Program |
| Function | Function |
| SubVI | Subroutine, Subprogram |
| Front Panel | User Interface |
| Block Diagram | Program Code |
| G | C, Pascal, BASIC, etc. |

## بررسی مثالها

بسیار خوب! دیگر مطالعه کافی است. برای درک نحوه‌ی عملکرد نرم‌افزار LabVIEW کافی است برنامه‌های موجود را باز و اجرا کنید.

8- Hierarchy
9- Modular Programming
10- Application

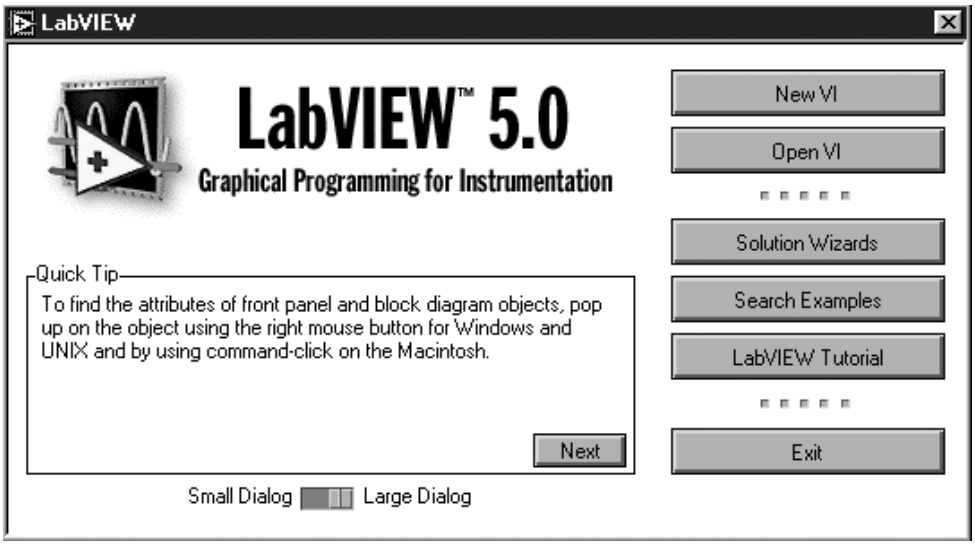
با توجه به توضیحات ارائه شده در بخش مقدمه، نرم‌افزار LabVIEW را نصب نمایید و مثالهای موجود در فهرست Exercise را ملاحظه کنید. این فهرست حاوی مثالها و تمرینات موجود در این کتاب است. پس از نصب نرم‌افزار LabVIEW برای مشاهده و اجرای هر یک از مثالها بر روی گزینه‌ی ...Open از منوی File کلیک نموده و مثال مورد نظر را انتخاب نمایید. همچنین جهت ایجاد یک برنامه‌ی جدید می توانید گزینه‌ی New را از همان منو انتخاب کنید.

در این کتاب هرگاه صحبت از کلیک کردن توسط ماوس به میان می‌آید منظور، استفاده از کلید سمت چپ ماوس است. بنابراین در حالت کلی و تا زمانی که بر استفاده از کلید سمت راست ماوس تأکید نشده است از کلید سمت چپ استفاده نمایید. در محیط Mac در صورت نیاز به کلید سمت راست ماوس می‌توانید کلید <Cmd> را پایین نگه داشته، سپس کلیک کنید.

## تمرین ۱-۱: اندازه‌گیری درجه حرارت

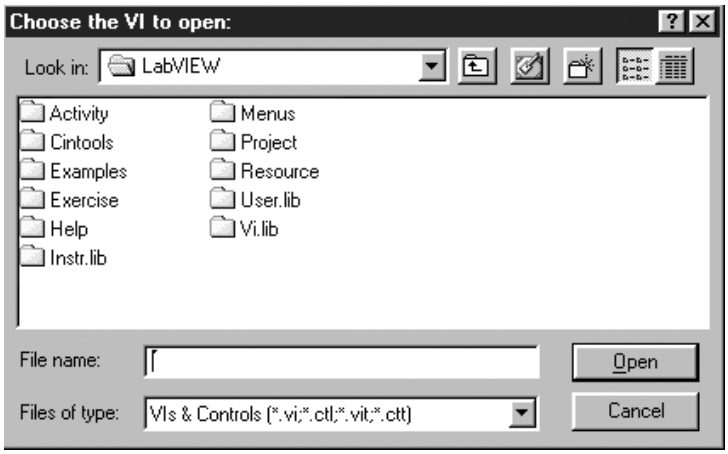
یکی از مثالهای موجود با عنوان Temperature System Demo.vi را به ترتیب زیر باز و اجرا نمایید:

۱- در صورتی که هنوز نرم‌افزار LabVIEW را نصب نکرده‌اید، این عمل را انجام دهید. در این حالت صفحه‌ای مطابق شکل ۶-۱ ظاهر می گردد.



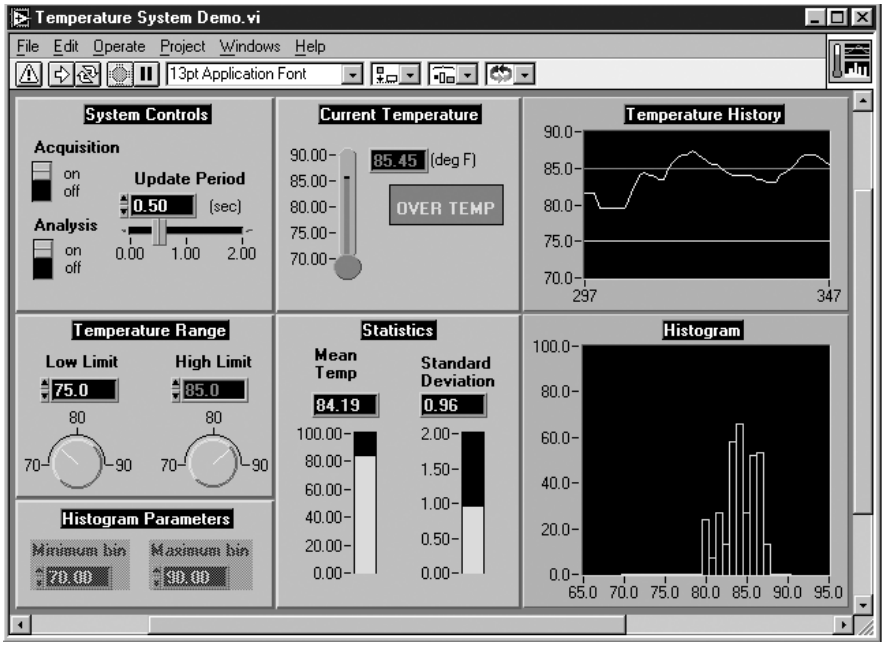

شکل ۶-۱

۲- برای باز کردن این برنامه بر روی دکمه‌ی Open VI کلیک کنید. روش دیگر برای انجام این عمل آن است که ابتدا بر روی دکمه‌ی New VI کلیک کرده، سپس در صفحه‌ی ظاهر شده، گزینه‌ی ...Open را از منوی File انتخاب نمایید. در این حالت صفحه‌ای مطابق شکل ۷-۱ ظاهر می گردد.




**شکل ۷-۱**

۳- فهرست Exercise را با دو بار کلیک کردن بر روی آن باز کنید. سپس کتابخانه‌ی Ch1.llb[11] را که حـاوی مثـالها و برنـامه‌های فصل ۱ می‌باشد انتخاب نمایید و در مرحـلـه‌ی بعدی برنـامه‌ی Temperature System Demo.vi را باز کنید. این برنامه در کتابخانه‌ی tempsys.llb نیز ذخیره شده است. کتابخانه‌ی مذکور در مسیر LabVIEW\Examples\Apps قرار دارد. پس از چند لحظه صفحه‌ی پانل برنامه‌ی مذکور مطابق شکل ۸-۱ ظاهر خواهد شد. همان گونه که ملاحظه می‌کنید در این صفحه المان‌های کنترل عددی، کلیدهای جبری، کلیدهای لغزشی، گراف، نمودار و یک نشان‌دهنده‌ی درجه حرارت یا ترمومتر وجود دارد.




**شکل ۸-۱: صفحه‌ی پانل در برنامه‌ی Temperature Syetem Demo.vi**

۱۱-کتابخانه یک نوع فایل به خصوص در محیط LabVIEW است که برای ذخیره نمودن برنامه‌ها مورد استفاده قرار می‌گیرد. در مورد فایل‌های کتابخانه‌ای در فصلهای بعدی به تفصیل سخن خواهیم گفت.

Run Button

Run Button (active)

۴- با کلیک کردن بر روی دکمه‌ی Run، برنامه را اجرا کنید. همان گونه که ملاحظه می کنید شکل ظاهری این دکمه در حالت فعال تغییر خواهد کرد و بیانگر این مطلب است که برنامه‌ی موجود بر روی صفحه در حال اجراست. به دلیل اینکه در برنامه‌ی فعال و در حال اجرا نمی توان تغیراتی ایجاد نمود، نوار ابزار یا Toolbar که محل قرار گرفتن ابزارها و دکمه هاست نیز تغییر شکل می دهند و المان های موجود در آن تعویض می شوند.

Abort Button

توجه داشته باشید که در این حالت دکمه‌ی Abort در نوار ابزار فعال می گردد. با فشاردادن این دکمه می توانید اجرای برنامه را متوقف کنید. همان گونه که احتمالاً استنباط کرده اید برنامه‌ی Temperature System Demo.vi فرآیند کنترل درجه‌ی حرارت را شبیه سازی می کند. در این برنامه، درجه حرارت اندازه گیری شده بر روی نمودارها و نشان دهنده های درجه‌ی حرارت به نمایش در می آید. اگرچه در این مثال، مقادیر شبیه سازی شده اندازه گیری می شوند و این برنامه به صورت نمایشی اجرا می گردد، به آسانی می توانید برنامه‌ی مذکور را برای اندازه گیری مقادیر حقیقی تغییر دهید و آن را اصلاح کنید.

کلید لغزشی Update Period مقادیر قرائت شده را ذخیره می کند. به عبارت دیگر پریود نمونه برداری و ذخیره کردن مقادیر اندازه گیری شده توسط این کلید تعیین می شود. این برنامه همچنین می تواند حداکثر و حداقل مقادیر اندازه گیری شده را بر روی صفحه نمایش دهد. با استفاده از کلیدهای Temperature Range می توان این حدود را تغییر داد. در صورتی که مقادیر قرائت شده، خارج از محدوده‌ی مجاز باشند نشان دهنده‌ی OVER TEMP روشن می شود.

تا زمانی که کلید Acquisition را در وضعیت off قرار ندهید، این برنامه به طور پیوسته اجرا می‌شود. همچنین می توانید تجزیه و تحلیل داده ها را به کمک کلید Analysis فعال یا غیر فعال کنید. بخش آماری برنامه، میانگین و انحراف استاندارد مقادیر قرائت شده را نشان می دهد. در ضمن نمودار فرکانسی که بر حسب تغییرات دما تنظیم شده است، در Histogram ترسیم می شود.

## ایجاد تغییر در مقادیر و پارامترهای موجود در صفحه

Operating Tool

Enter Button

۵- با استفاده از نشانگر ماوس که درحالت اجرای برنامه به ابزار Operating Tool تغییر شکل می دهد، می توان مقادیر و پارامترهای موجود را تغییر داد. در این حالت به کمک این ابزار مقادیر حداکثر و حداقل دما را تغییر دهید. بدین ترتیب که با دو بار کلیک کردن بر روی پارامتری که قصد تغییر آن را دارید، مقادیر حداکثر و حداقل آن را متمایز[12] کنید. این عمل با کلیک کردن و کشیدن ابزار Operating Tool در طول پارامتری که قصد دارید آن را تغییر دهید نیز امکان پذیر است. سپس مقادیر جدید را به کمک صفحه کلید وارد نمایید و بر روی دکمه‌ی Enter کلیک کنید. این دکمه پس از دکمه‌ی Run در نوار ابزار واقع شده است.

12- Highlight

۶- با قرار دادن ابزار Operating Tool بر روی کلید لغزشی، مقدار پارامتر Update Period را تغییر دهید. سپس کلیک کرده، آن را به محل جدید حرکت دهید.

برای انجام عمل مذکور روش دیگری نیز وجود دارد. بدین ترتیب که می‌توانید به کمک ابزار Operating Tool بر روی نقطه‌ای از کلید لغزشی که پارامتر مورد نظر باید مطابق با آن مقدار تنظیم گردد کلیک کنید. در ضمن می‌توانید بر روی پیکان افزایش یا کاهش کلیک کنید و کلید لغزشی را به آرامی در جهت پیکان‌های افزایش یا کاهش حرکت دهید یا بر روی نشان‌دهنده‌ی دیجیتالی[۱۳] کلیک نموده، مقدار جدید را وارد کنید.

اگرچه در صورت انجام اعمال مذکور اعداد به نمایش درآمده بر روی صفحه‌ی نمایش دیجیتالی تغییر می‌کنند، اما تا زمانی که بر روی دکمه‌ی Enter کلیک نکرده باشید پارامترهای جدید در برنامه اعمال نخواهند شد.

۷- به همین روش سعی کنید تا مقادیر پارامترهای دیگر را نیز تغییر دهید.

۸- با استفاده از کلید Acquisition اجرای برنامه را متوقف سازید.

## بررسی صفحه‌ی نمودار بلوکی

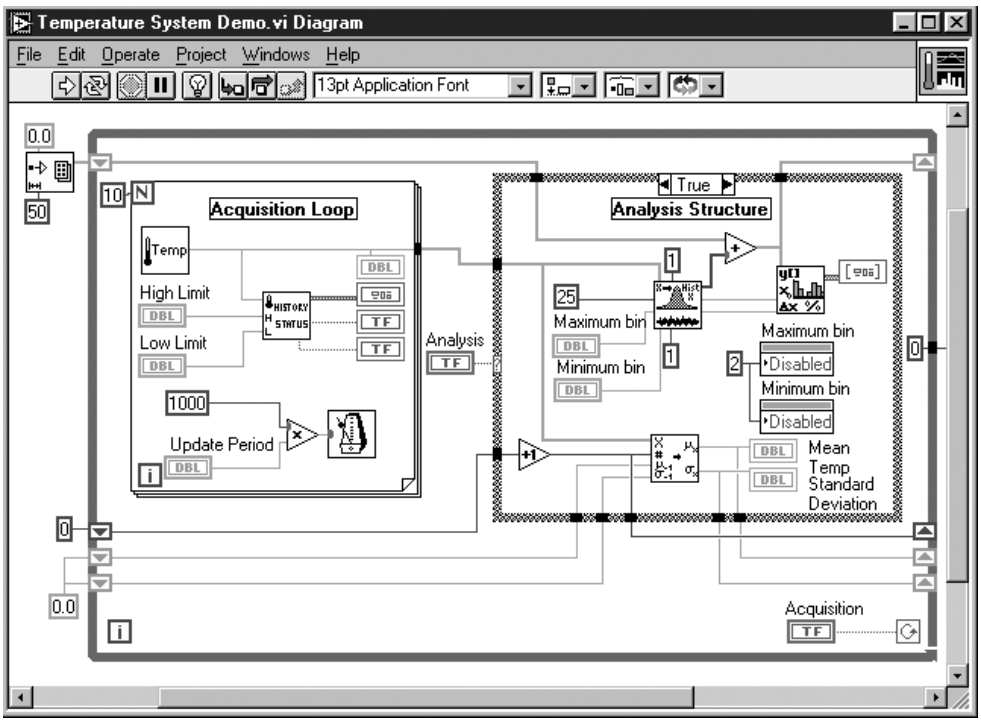
در صفحه‌ی نمودار بلوکی شکل ۹-۱ نحوه‌ی عملکرد یک برنامه نشان داده شده است. در این مرحله نیازی به شناخت تمامی المان‌های موجود در این صفحه نیست. در فصول آینده با هر یک از اجزای این صفحه آشنا خواهید شد. هدف از بررسی صفحه‌ی نمودار بلوکی، تنها درک ماهیت واقعی این پنجره است. اگر در این مرحله واقعاً به نحوه‌ی عملکرد این بلوک پی برده باشید به جرأت می‌توان گفت که شما قادر هستید بدون مطالعه‌ی مطالب این فصل، به فصل بعدی مراجعه کنید.

۹- با انتخاب گزینه‌ی Show Diagram از منوی Windows صفحه‌ی نمودار بلوکی برنامه‌ی Temperature System Demo.vi را باز کنید.

از این به بعد جهت انتخاب یکی از گزینه‌های موجود در منو برای خلاصه‌نویسی قرارداد زیر را می‌پذیریم:

به عنوان مثال، انتخاب گزینه‌ی Show Diagram از منوی Windows را به صورت Windows >> Show Diagram نشان می‌دهیم.

13- Digital Display




**شکل ۹-۱: صفحه‌ی نمودار بلوکی در برنامه‌ی Temperature Syetem Demo.vi**

توجه داشته باشید که در نوار عنوان صفحه‌ی نمودار بلوکی، عبارت «Diagram» پس از نام برنامه درج شده است.

۱۰- عناصر و اجزای موجود در این بلوک را بررسی کنید. از مشاهده‌ی توابع و دستورهای موجود در این صفحه مضطرب نشوید. در این کتاب مرحله به مرحله با اجزای استفاده شده در این صفحه آشنا خواهید شد.

۱۱- با انتخاب گزینه‌ی Help >> Show Help پنجره‌ی Help را باز کنید. با قرار دادن نشانگر ماوس بر روی هر یک از عناصر و المان‌های موجود در این بلوک، توضیحات مندرج در این پنجره را ملاحظه کنید. در صورتی که نشانگر ماوس بر روی یک زیربرنامه و یا یک دستور قرار گیرد، علاوه بر ارائه‌ی توضیحات در پنجره‌ی Help، در مورد ورودیها و خروجیهای آن نیز توضیحاتی در اختیار کاربر قرار داده می‌شود.

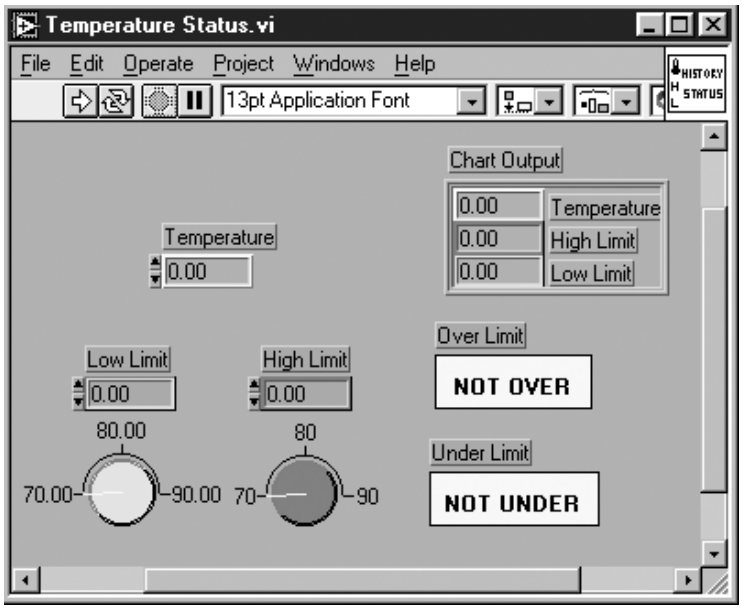
## سلسله مراتب

قدرت نرم‌افزار LabVIEW در طبیعت سلسله مراتبی برنامه‌های آن نهفته است. با توجه به اصل سلسله مراتب، پس از ایجاد یک برنامه می‌توانید آن را به عنوان یک زیربرنامه در صفحه‌ی نمودار بلوکی برنامه‌ی مراتب بالاتر به‌کار برید. در ضمن در مورد تعداد لایه‌های ترتیبی برای این برنامه‌ها محدودیتی وجود ندارد. برای درک این قابلیت یکی از زیربرنامه‌های استفاده شده در برنامه‌ی Temperature System Demo.vi را بررسی کنید.

۱۲- صفحه‌ی نمودار بلوکی برنامه‌ی Temperature System Demo.vi را باز نموده، سپس صفحه‌ی پانل زیربرنامه‌ی Temperature Status.vi را با دو بار کلیک کردن بر روی آیکن آن باز کنید.

**شکل ۱۰-۱: آیکن زیربرنامه‌ی Temperature Status.vi**

در این حالت ملاحظه می کنید که صفحه‌ی پانل این زیربرنامه مطابق شکل ۱۱-۱ بر روی صفحه ظاهر می گردد.

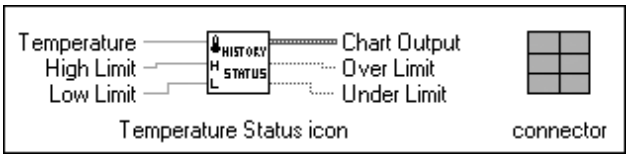



**شکل ۱۱-۱: صفحه‌ی پانل در زیربرنامه‌ی Temperature Status.vi**

## آیکن/کانکتور

در صورتی که از یک برنامه به عنوان زیربرنامه در صفحه‌ی نمودار بلوکی برنامه های دیگر استفاده گردد، برای نمایش گرافیکی آن، از آیکن/کانکتور استفاده می‌شود. این دو نماد در گوشه‌ی بالایی سمت راست صفحه‌ی پانل قرار دارند. البته در هر لحظه تنها یکی از این دو نماد قابل رؤیت است.

از لحاظ گرافیکی، یک آیکن نمایش‌دهنده‌ی برنامه در صفحه‌ی نمودار بلوکی برنامه‌های دیگر است. از طریق ترمینال های کانکتور می توانید ورودی و خروجیها را به آیکن ارتباط دهید. این ترمینال ها با پارامترهای Subroutine یا Function در زبانهای برنامه‌نویسی متنی متناظر هستند. برای هر یک از کنترل‌ها و نشان‌دهنده‌های موجود در صفحه‌ی پانل، یک ترمینال منحصر به فرد در کانکتور در نظر گرفته می شود. در شکل ۱۲-۱ یک زوج آیکن/کانکتور را ملاحظه می کنید.




**شکل ۱۲-۱: شمای آیکن و کانکتور در زیربرنامه‌ی Temperature Status.vi**

با استفاده از زیربرنامه ها می توانید ساختار صفحه‌ی نمودار بلوکی را مدولار و کنترل‌پذیر کنید. خاصیت

مدولار بودن باعث می شود تا بتوان به راحتی برنامه ها را درک کرد و آنها را تغییر داد. علاوه بر این، می توان یک زیربرنامه را برای انجام عملیاتی که توسط چند برنامه صورت می گیرد، ایجاد نمود. در حالی که پنجره‌ی زیربرنامه‌ی Temperature Status.vi قابل رؤیت است، برنامه‌ی اصلی را اجرا نمایید. در این حالت به تغییر مقادیر و پارامترهای زیربرنامه در زمانی که چند بار توسط برنامه‌ی اصلی فراخوانی می شود توجه کنید.

۱۳- با انتخاب گزینه‌ی File >> Close در صفحه‌ی پانل زیربرنامه‌ی Temperature Status.vi، این زیربرنامه را ببندید. در این حالت از ذخیره نمودن تغییرات ایجاد شده خودداری کنید.

۱۴- با انتخاب گزینه‌ی File >> Close در صفحه‌ی پانل برنامه‌ی Temperature System Demo.vi، این برنامه را ببندید. در این حالت نیز از ذخیره نمودن تغییرات ایجاد شده خودداری کنید.

انتخاب گزینه‌ی File >> Close در پنجره‌ی نمودار بلوکی تنها منجر به بسته شدن همین پنجره می‌گردد. برای بستن هر دو پنجره‌ی پانل و نمودار بلوکی، گزینه‌ی File >> Close را در صفحه‌ی پانل انتخاب کنید.

## تمرین ۲-۱: بررسی پاسخ فرکانسی

در این مثال به بررسی پاسخ فرکانسی یک بلوک ناشناخته می پردازیم. در مباحث برق و الکترونیک به این بلوک ناشناخته در اصطلاح «جعبه‌ی سیاه[14]» گفته می شود. در این حالت یک منبع مولد موج، یک موج سینوسی را به ورودی «جعبه‌ی سیاه» اعمال می کند. شایان ذکر است که بلوک ناشناخته‌ی مورد بحث شامل یک فیلتر میان گذر است و تنها اجازه‌ی عبور به محدوده‌ی خاصی از فرکانس ها را صادر می کند. یک مولتی متر دیجیتال جهت اندازه گیری ولتاژ خروجی این بلوک به کار برده می شود. در این برنامه از چند زیربرنامه جهت مواردی چون شبیه سازی منبع مولد موج و مولتی متر دیجیتال استفاده شده است. به کمک این زیربرنامه ها می توان به جای انجام عملیات بر روی داده های شبیه سازی شده، اعمالی نظیر جمع آوری و انتقال داده ها به کارت های GPIB و یا درگاههای سریال رایانه را جهت دریافت و ارسال داده های حقیقی انجام داد.

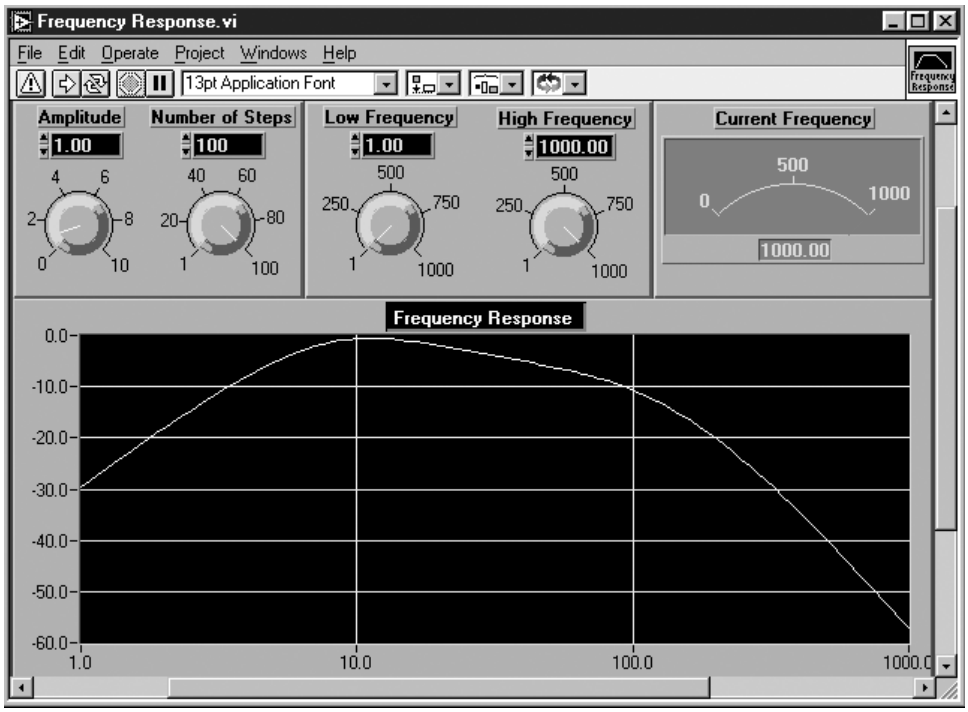
این برنامه را به ترتیب زیر باز نموده، اجرای آن را ملاحظه کنید.

۱- گزینه‌ی File >> Open ... را انتخاب نمایید.

۲- کتابخانه‌ی Ch1.llb را از فهرست Exercise انتخاب نموده، سپس با دو بار کلیک نمودن بر روی برنامه‌ی Frequency Response.vi، پنجره‌ی پانل آن را باز کنید. برای دستیابی به این برنامه می توانید به مسیر LabVIEW\Examples\Apps\freqresp.llb نیز مراجعه نمایید. در این حالت یک صفحه‌ی پانل مطابق شکل ۱۳-۱ ظاهر می گردد.

14- Black Box

۱




**شکل ۱۳-۱: صفحه‌ی پانل در برنامه‌ی Frequency Response.vi**

Run Button

۳- با کلیک نمودن بر روی دکمه‌ی Run، این برنامه را اجرا کنید. در برنامه‌ی مذکور با تغییر پارامترهای Amplitude و Number of Steps، به ترتیب دامنه‌ی موج سینوسی و تعداد مراحلی را که جهت تعیین پاسخ فرکانسی توسط برنامه مورد استفاده قرار می‌گیرند تغییر دهید و مجدداً برنامه را اجرا کنید. در این برنامه همچنین می‌توانید به کمک دکمه‌های High Frequency و Low Frequency حداکثر و حداقل فرکانس ورودی را جهت تعیین فرکانس جاروب[15] تغییر دهید. حال با تغییر دکمه‌های مذکور و اجرای مجدد برنامه، تأثیر آنها را بر روی خروجی «جعبه‌ی سیاه» ملاحظه کنید.

۴- با انتخـاب گزینه‌ی Windows >> Show Diagram پنجره‌ی نمودار بلوکی این برنامه را باز نموده، المان‌های موجود در آن را مشاهده کنید.

۵- گزینه‌ی File >> Close را انتخاب کنید و این برنامه را ببندید.

این تمرینات به شما ایده‌ای در مورد محیط برنامه‌نویسی «G» خواهند داد. به زودی به کمک زبان برنامه‌نویسی «G» می‌توانید برنامه‌های سطح بالا و پیشرفته بنویسید و به رفع اشکال آنها بپردازید.

## خلاصه

برنامه‌ی LabVIEW یک نرم‌افزار قابل انعطاف و کاربردی جهت شبیه‌سازی تجهیزات و ابزار آزمایشگاهی و اندازه‌گیری است. این نرم‌افزار همچنین قابلیت تجزیه و تحلیل داده‌ها و نتایج به‌دست آمده را داراست.

15- Frequency Sweep

جهت ایجاد برنامه در این نرم‌افزار از زبان برنامه‌نویسی گرافیکی موسوم به «G» استفاده می‌شود. برنامه‌های نوشته شده در محیط LabVIEW را در اصطلاح «ابزار مجازی» یا به طور خلاصه VI گوییم.

کاربر از طریق صفحه‌ای به نام پانل با برنامه ارتباط برقرار می‌کند. هر صفحه‌ی پانل دارای یک صفحه‌ی نمودار بلوکی است. این صفحه محل قرار گرفتن کدها و نمادهای گرافیکی در برنامه می‌باشد. شایان ذکر است که دو صفحه‌ی مذکور در تناظر یک به یک قرار دارند. به عبارت دیگر هر صفحه‌ی پانل متناظر با یک صفحه‌ی نمودار بلوکی می‌باشد و برعکس. در این نرم‌افزار جهت نوشتن برنامه، دستورها و توابع متعددی در کتابخانه‌ها وجود دارد که برنامه‌نویسی را بسیار آسان و سریع می‌سازند. برای برقراری ارتباط بین المان‌ها از سیم استفاده می‌شود. در فصل بعدی روش به کارگیری قابلیتهای LabVIEW را مورد بررسی قرار می‌دهیم.

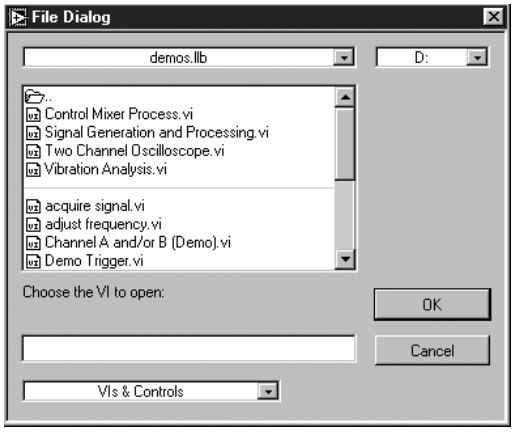
پاسخ تمامی مثالها و تمرینات موجود در این کتاب در فهرست Exercise گنجانده شده است. ابتدا سعی کنید تا پاسخ تمرینات را به‌دست آورید و سپس آنها را با جوابهای موجود در فهرست مذکور مقایسه کنید.

## تمرینات اضافی

### تمرین ۳-۱: برنامه‌های نمایشی[16]

در این تمرین قصد داریم تا با ملاحظه‌ی تعدادی از برنامه‌های نمایشی در محیط LabVIEW برخی از قابلیتهای این بسته‌ی نرم‌افزاری را بررسی کنیم. در این کتاب متوجه خواهید شد که تنها با مشاهده و بررسی دقیق مثالهای موجود می‌توان مطالب بسیاری را در مورد این نرم‌افزار استنباط نمود.

۱- برای انجام این تمرین برنامه‌های موجود در کتابخانه‌ی demos.llb را اجرا نمایید. این کتابخانه در مسیر LabVIEW\Examples\Apps قرار دارد. در شکل ۱۴-۱ کتابخانه‌ی مذکور را به همراه برنامه‌های موجود در آن ملاحظه می‌کنید.




**شکل ۱۴-۱: پنجره‌ی محاوره‌ای File Dialog در کتابخانه‌ی demos.llb**

16- Demo

۲- حال برنامه‌ی Control Mixer Process.vi را باز نموده، با کلیک کردن بر روی دکمه‌ی Run، آن را اجرا کنید.

۳- برای کسب اطلاعات بیشتر در مورد نحوه‌ی عملکرد این برنامه بر روی دکمه‌ی ...MORE INFO در پنجره‌ی پانل آن کلیک کنید. در ضمن با کلیک کردن بر روی دکمه‌ی MANUAL می توانید اجرای این برنامه را به صورت دستی کنترل نمایید.

همان‌گونه که در شکل ۱۴-۱ ملاحظه می کنید برنامه های موجود در کتابخانه ها با پسوند vi. ذخیره شده‌اند. اما این برنامه ها به دو بخش تقسیم می شوند:

بخش اول که شامل برنامه های سطح بالا یا اصلی است و بخش دوم که توسط یک خط افقی از بخش قبلی جدا شده است و شامل زیربرنامه های استفاده شده در برنامه های سطوح بالاتر است.

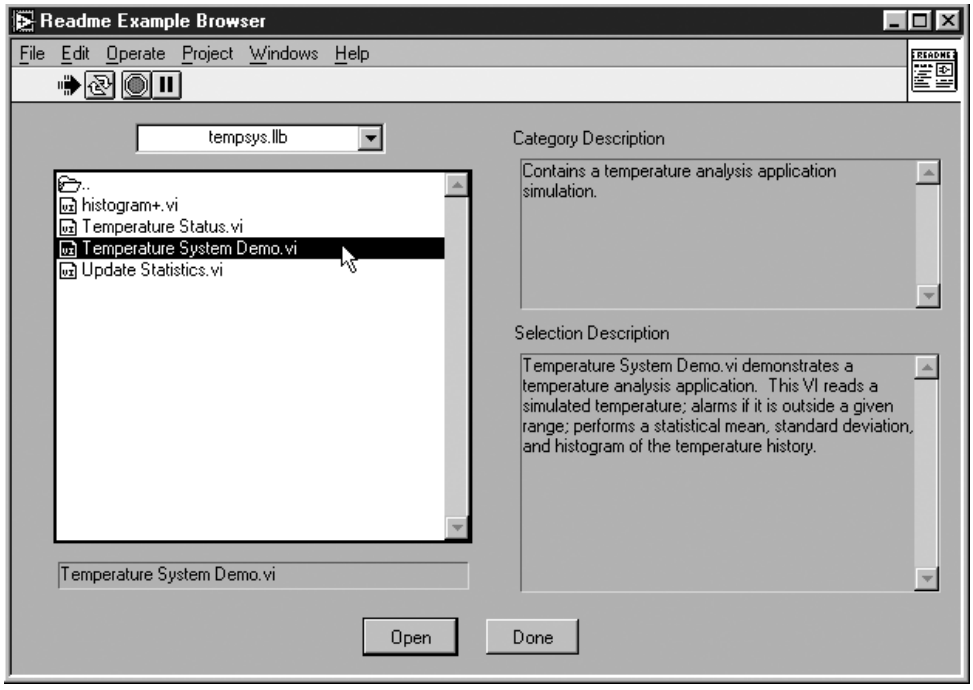
حتماً با انتخاب گزینه‌ی Windows >> Show Diagram، پنجره‌ی نمودار بلوکی این برنامه را نیز مشاهده کنید. با این روش می توانید نحوه‌ی عملکرد این برنامه را درک کنید.

## تمرین ۴-۱ : بررسی چند مثال

در این تمرین، چند مثال کاربردی را در محیط LabVIEW بررسی می کنیم.

۱- برنامه‌ی readme.vi را از فهرست LabVIEW\Examples باز کنید.

۲- با کلیک کردن بر روی دکمه‌ی Run، این برنامه را اجرا نمایید. با استفاده از برنامه‌ی readme.vi می توانید در میان فهرست ها و کتابخانه های موجود به جستجو بپردازید و توضیحات ارائه شده در مورد هر یک را ملاحظه کنید.




**شکل ۱۵-۱ : صفحه‌ی پانل در برنامه‌ی readme.vi**

در صورت انتخاب هر یک از برنامه‌ها محتویات، اعمال انجام گرفته و وظایف محوله به این برنامه در بخش Selection Description توضیح داده می‌شوند. با دو بار کلیک نمودن بر روی یکی از فهرست‌ها آن را باز نموده، به زیرفهرست‌ها و برنامه‌های موجود در آن دسترسی پیدا کنید. پس از بررسی برنامه‌های موجود در هر فهرست، برای خروج از برنامه‌ی readme.vi دکمه‌ی Done را فشار دهید.

۳- با انتخاب گزینه‌ی Windows >> Show Diagram پنجره‌ی نمودار بلوکی هر یک از برنامه‌ها را ملاحظه کنید.

۴- حال دیگر برنامه‌های موجود در فهرست Examples را اجرا نموده، نظر خود را در مورد محیط برنامه‌نویسی LabVIEW بیان کنید.

اکنون به بررسی فهرست‌های General، Apps و daq بپردازید و سعی کنید با بررسی نحوه‌ی عملکرد برنامه‌های موجود در این فهرست‌ها، قابلیتهای محیط LabVIEW را درک کنید. همچنین می‌توانید این برنامه‌ها را تغییر داده، برای کاربردهای متعددی از آنها استفاده کنید.

۵- پس از بررسی برنامه‌ها، گزینه‌ی File >> Close را انتخاب کنید و صفحات برنامه‌های انتخاب شده را ببندید. از ذخیره نمودن تغییرات ایجاد شده‌ی احتمالی خودداری کنید.

LabVIEW

# ۲

# برقراری ارتباط بین رایانه و دنیای خارج

## در این فصل با موارد زیر آشنا می‌شوید:

- ماهیت واقعی و اجزای سیستم‌های DAQ و GPIB
- درگاههای سریال رایانه
- قابلیتهای مفید مربوط به دستورها و توابع تحلیلی
- باس VXI
- نحوه‌ی تبادل داده‌ها بین چند رایانه یا چند برنامه‌ی کاربردی در محیط LabVIEW

# LabVIEW

# ۲

# برقراری ارتباط بین رایانه و دنیای خارج

سیستم اندازه گیری مجازی یا به اصطلاح Virtual Instrumentation، پایه و اساس آزمایشگاههای مدرن و امروزی است. مجموعه‌ی سیستم اندازه گیری مجازی شامل یک دستگاه رایانه، نرم افزار مربوط و کارت هایی است که تجهیزات و ابزار آزمایشگاهی را شبیه سازی می کنند. این نرم افزار همان برنامه ای است که ما آن را LabVIEW می نامیم. به دلیل اینکه عملکرد این سیستم توسط کاربر و به صورت نرم افزاری تعیین می شود، کاملاً قابل انعطاف و قدرتمند است و از لحاظ هزینه، نسبت به ابزار های اندازه گیری حقیقی در سطحی بسیار پایین تر می باشد. در این فصل در مورد نحوه‌ی برقراری ارتباط بین رایانه و دنیای خارج به بحث می نشینیم.

در فصول مقدماتی تنها اشاره‌ی مختصری به این موضوع خواهیم داشت. در بخش پیشرفته کتاب در مورد نحوه‌ی جمع آوری داده ها، کنترل و نظارت بر اندازه گیری ها و شبکه کردن یک رایانه با رایانه های دیگر اطلاعات بیشتری در اختیار شما قرار خواهیم داد. در خلال مطالعه‌ی این فصل متوجه خواهید شد که نرم افزار LabVIEW چگونه در طول سالهای اخیر متحول شده است.

در این فصل اهداف زیر دنبال می شوند:

- شناخت ماهیت واقعی و اجزای سیستم های DAQ و GPIB.
- فراگیری مطالبی در مورد درگاههای سریال رایانه.
- فراگیری قابلیتهای مفید مربوط به دستورها و توابع تحلیلی.
- کسب اطلاعات مختصری در زمینه‌ی VXI.
- بررسی نحوه‌ی تبادل داده ها بین چند رایانه یا چند برنامه‌ی کاربردی در محیط LabVIEW.

- فراگیری مطالبی در مورد برخی از ابزارهایی که می توان به نرم افزار LabVIEW اضافه نمود و بدین ترتیب قابلیتهای آن را افزایش داد.

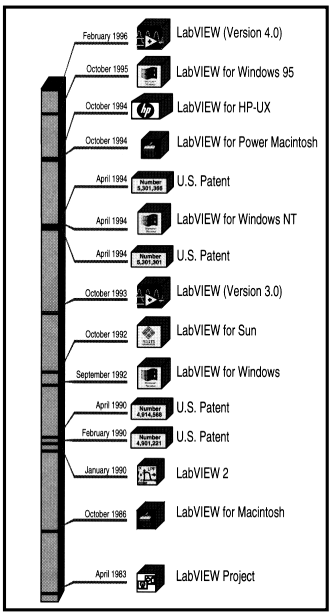
اصطلاحات کلیدی موجود در این فصل عبارتند از:

- جمع آوری دادها یا DAQ (Data Acquisition)
- باس ارتباطی همه منظوره یا GPIB (General Purpose Interface Bus)
- استاندارد IEEE 488.2
- درگاههای سریال رایانه (Serial Ports)
- برقراری ارتباط از طریق شبکه (Networking)
- DLL (Dynamic Link Library)
- CIN (Code Interface Node)
- OLE (Object Linking and Embedding)
- ابزارهایی که می توان به LabVIEW اضافه نمود و بدین ترتیب قابلیتهای آن را افزایش داد (Toolkits)

۲

## سیر تکامل تدریجی نرم افزار LabVIEW

در سال ۱۹۸۳ شرکت NI[1] تحقیقات خود را برای یافتن راهی جهت به حداقل رساندن زمان لازم برای برنامه نویسی سیستم های اندازه گیری آغاز نمود. در نتیجه‌ی این تحقیقات، نظریه‌ی «ابزار مجازی» یا در اصطلاح Virtual Instrument ظهور کرد. در این نظریه از یک زبان برنامه نویسی گرافیکی استفاده شده بود. در این زبان برنامه نویسی، از صفحه‌ای موسوم به «پانل» جهت برقراری ارتباط با کاربر استفاده می شد. همچنین صفحه‌ی دیگری به نام «نمودار بلوکی» که در آن کدهای برنامه نویسی قرار داشتند به کار برده شد. در شکل ۱-۲ سیر تکامل بسته‌ی نرم افزاری LabVIEW را ملاحظه می کنید.




**شکل ۱-۲ : سیر تکامل نرم افزار LabVIEW**

اولین نسخه‌ی این نرم افزار یا LabVIEW 1 در سال ۱۹۸۶ پا به عرصه‌ی دنیای نرم افزار گذاشت. این برنامه تنها در محیط Mac قابل استفاده بود. اگرچه سیستم Mac جهت اهداف و مقاصد آزمایشگاهی و اندازه گیری به طور گسترده مورد استفاده قرار نمی گرفت، اما به دلیل سازگاری طبیعت گرافیکی این سیستم با ماهیت گرافیکی و تکنولوژی استفاده شده در LabVIEW، در ابتدای امر مورد توجه قرار گرفت تا اینکه سیستم های عامل دیگری توانستند این بسته‌ی نرم افزاری را پشتیبانی کنند.

---

۱- از این به بعد شرکت National Instruments را به اختصار، شرکت NI می خوانیم.

تا سال ۱۹۹۰ شرکت NI نسخه های دیگری از نرم‌افزار LabVIEW را با ویرایش های جدید و با استفاده از تکنیک های نرم‌افزاری جدید و بهره گیری از راهنمایی و پیشنهادهای کاربران عرضه نمود. نکته‌ی شایان ذکر آن است که در خلال این سالها، نسخه‌ی دوم این نرم‌افزار یعنی LabVIEW 2 به بازار آمد و سرعت اجرای VIها را افزایش داد تا جایی که سرعت اجرای VIها با سرعت برنامه‌های نوشته شده در محیط برنامه‌نویسی C قابل مقایسه بود.

در حالی که سیستم های عامل دیگری در محیطهای گرافیکی و غیرگرافیکی ظهور نمودند، شرکت NI نیز تحقیقات خود را جهت به اجرا در آوردن برنامه‌ی LabVIEW در رایانه های شخصی و سیستم های دیگر ادامه داد. در سال ۱۹۹۲ این شرکت اعلام نمود که نرم‌افزار LabVIEW بر روی رایانه های شخصی و سیستم های Sun نیز قابل اجراست.

نسخه‌ی شماره‌ی ۳ این نرم‌افزار یعنی LabVIEW 3 در سال ۱۹۹۴ برای سیستم های عامل Mac، Windows و Sun به بازار عرضه شد. از امکانات قابل توجه این نسخه، قابلیت برنامه‌نویسی در یکی از سیستم[۲]ها و به اجرا در آوردن آن برنامه تحت همان سیستم و یا سیستم های دیگر بود. این قابلیت به کاربر اجازه می داد تا سیستم عامل مورد نظر خود را از بین سیستم های مختلف انتخاب نماید.

در سال ۱۹۹۴ سیستم های عامل دیگری نظیر Windows NT و HP در فهرست سیستم های پشتیبانی کننده‌ی نرم‌افزار LabVIEW قرار گرفتند. با ظهور سیستم عامل Win 95 در سال ۱۹۹۵ این سیستم نیز به فهرست مذکور اضافه شد.

نسخه‌ی شماره‌ی۴ این نرم‌افزار یا LabVIEW 4.0 که در سال ۱۹۹۶ عرضه شد دارای قابلیتها و امکانات پیشرفته‌ی بسیاری بود. به گونه‌ای که کاربران می توانستند محیط برنامه‌نویسی خود را مطابق با سطح معلومات، تجربه و نیازهای خود ایجاد نمایند. علاوه بر این، نسخه‌ی مذکور دارای ابزارهای قدرتمندی جهت ایجاد تغییر و انجام اصلاحات در سیستم های پیشرفته‌ی اندازه گیری بود.

نسخه‌ی LabVIEW 5.0 نیز در سال ۱۹۹۷ عرضه شد. تمامی سیستم های عامل از جمله Windows XP این نرم‌افزار را پشتیبانی می کنند.

اگرچه این بسته‌ی نرم‌افزاری، ابزاری قدرتمند جهت شبیه‌سازی داده‌هاست، اما در اکثر موارد برای جمع آوری داده ها و برقراری ارتباط بین رایانه و دنیای خارج نیز مورد استفاده قرار می گیرد. به همین منظور برنامه های بسیار زیادی جهت انجام اعمال مذکور در محیط LabVIEW تعبیه شده‌اند. به عنوان مثال به کمک نرم‌افزار LabVIEW می توان از کارت های DAQ جهت جمع آوری داده ها و یا تولید سیگنال های آنالوگ و دیجیتال استفاده نمود.

به کمک این کارت‌ها و نرم‌افزار LabVIEW می‌توان دمای یک محیط را تحت کنترل و نظارت قرار داد، سیگنال‌هایی را به یک سیستم دیگر ارسال نمود و یا مقدار فرکانس یک سیگنال ناشناخته را تعیین کرد.

2- Platform

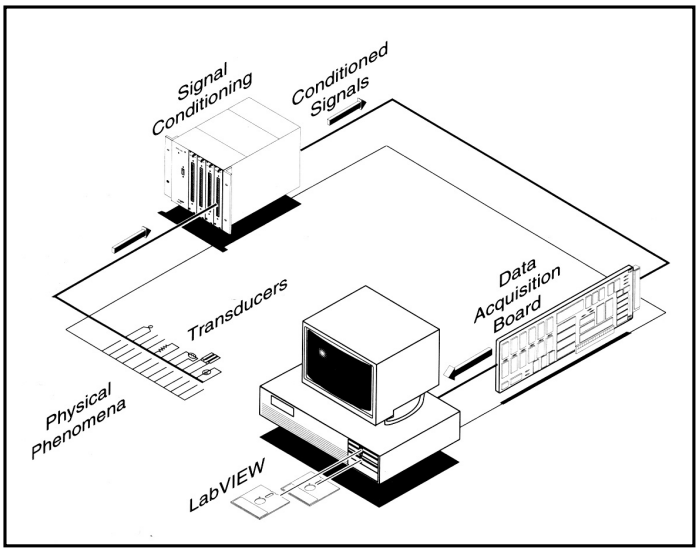
با به‌کارگیری نرم‌افزار LabVIEW و کارت‌های GPIB یا درگاههای سریال رایانه می‌توان داده‌ها را به آسانی جمع‌آوری نمود. جهت برقراری ارتباط بین رایانه و اسیلوسکوپ، مولتی‌متر، اسکنر و همچنین برای بهره‌برداری از سیستم‌های اندازه‌گیری که در فواصل دور از سیستم کنترل خود قرار دارند می‌توان از کارت‌های GPIB استفاده نمود. نرم‌افزار LabVIEW همچنین می‌تواند سخت‌افزار سیستم‌های اندازه‌گیری پیچیده‌ی VXI را تحت کنترل در آورد. با به‌کارگیری این بسته‌ی نرم‌افزاری می‌توانید پس از جمع‌آوری داده‌ها با کمک چند VI که جهت آنالیز و تحلیل داده‌ها در LabVIEW تعبیه شده‌اند، داده‌ها را پردازش کنید و آنها را به نمایش در آورید.

به کمک نرم‌افزار LabVIEW می‌توانید برای سهولت در امر پردازش، داده‌های جمع‌آوری شده را بین چند رایانه تقسیم کنید. در این بسته‌ی نرم‌افزاری توابع و دستورهایی از پیش نوشته شده وجود دارند که انجام این عمل را تسهیل می‌نمایند، همچنین چند پروتکل شبکه را پشتیبانی می‌کنند. در این نرم‌افزار می‌توان از CIN یا DLL نیز جهت فراخوانی کدهای خارجی استفاده نمود. در ادامه‌ی این فصل به بررسی امکانات و قابلیتهای ویژه‌ی LabVIEW می‌پردازیم.

۲

## جمع‌آوری داده‌ها (DAQ)

DAQ به طور خلاصه به اندازه‌گیری یک سیگنال حقیقی نظیر ولتاژ و ارسال آن به رایانه جهت پردازش، تجزیه و تحلیل، ذخیره و اعمال تغییرات اطلاق می‌گردد. در شکل ۲-۲ اجزای یک سیستم DAQ نشان داده شده است.




**شکل ۲-۲: اجزای یک سیستم DAQ**

در شکل ۲-۲ منظور از اصطلاح pHysical pHenomena کمیت فیزیکی مورد نظر یا همان سیگنال است. این سیگنال ممکن است سرعت، درجه حرارت، رطوبت، فشار، pH، مقدار فلو، سیگنال وضعیت روشن و خاموش شدن یک سیستم، میزان رادیواکتیویته‌ی یک عنصر، شدت نور و... باشد.

سنسورها[3] و مبدلها[4] کمیت فیزیکی مورد نظر را می‌سنجند و یک سیگنال الکتریکی متناسب با مقدار اندازه گیری شده تولید می کنند. به عنوان مثال ترموکوپل هایی که می توان آنها را نوعی مبدل فرض کرد، درجه حرارت را به ولتاژ تبدیل می کنند. این سطوح ولتاژ توسط «مبدل آنالوگ به دیجیتال» که به طور خلاصه آن را با A/D نشان می دهیم قابل اندازه گیری خواهند بود. نمونه های دیگری از مبدلها عبارتند از: گِیج[5]های کشش، فلومترها و مبدلهای فشار که به ترتیب جهت اندازه گیری نیرو، مقدار فلو و فشار مورد استفاده قرار می گیرند. در هر یک از حالات مذکور، سیگنال الکتریکی ایجاد شده توسط مبدل با کمیت فیزیکی مورد نظر متناظر است.

به کمک نرم افزار LabVIEW و کارت های DAQ می توان سیگنال های آنالوگ را با استفاده از مبدل A/D قرائت نمود. با به کارگیری این نرم افزار و کارت های مذکور می توان در موارد زیر از قابلیتهای LabVIEW استفاده کرد:

قرائت سیگنال های آنالوگ با استفاده از مبدل A/D، تولید سیگنال های خروجی آنالوگ به کمک مبدل D/A، قرائت و یا ارسال سیگنال های دیجیتال، اعمال تغییر در شمارنده های موجود در کارت ها جهت اندازه گیری فرکانس، تولید پالس و... .

در صورتی که کمیت مورد نظر یا سیگنال مورد بحث به صورت آنالوگ باشد، ولتاژ تولید شده توسط سنسور یا مبدل، به کارت های DAQ در رایانه ارسال می‌شود و پس از تبدیل آنها به سیگنال دیجیتال، عملیات ذخیره، پردازش و... بر روی داده ها یا سطوح ولتاژ تولید شده انجام می گیرد.

«مدول های تطبیق دهنده‌ی سیگنال[6]» سیگنال های تولید شده توسط مبدلها را با سیگنال ها و سطوح ولتاژ قابل قبول برای کارت های DAQ منطبق می سازند. به عنوان مثال فرض کنید که می خواهید ولتاژهای فشار قوی سیستم روشنایی شهر را مورد بررسی قرار دهید و آنها را تجزیه و تحلیل کنید. در صورت عدم استفاده از این مدول ها، رایانه، کارت های DAQ و اجزای دیگر مورد استفاده در این پروژه از بین خواهند رفت و موجب وارد آمدن خسارات جبران ناپذیری خواهد شد.

مدول های تطبیق دهنده‌ی سیگنال اعمالی نظیر تقویت، تضعیف، خطی سازی، فیلتر کردن، ایزوله‌سازی و... را انجام می دهند. در اکثر موارد استفاده از این مدول ها الزامی است. مگر در موارد معدودی که سیگنال های مورد بحث از نوع دیجیتال بوده، سطح ولتاژ آنها نیز پایین باشد. در هنگام استفاده از این مدول ها مراقب باشید که از نظر الکتریکی به رایانه و کارت های DAQ آسیبی وارد نگردد. در مورد استفاده از مدول های تطبیق دهنده، حالت امتحانی یا انتخابی وجود ندارد. بدین مفهوم که برای به کارگیری یک مدول تطبیق دهنده نمی توان از روش «سعی و خطا» استفاده کرد. بلکه باید قبل از شروع به کار حتماً شرایط و انطباق پارامترها و موارد خواسته شده با مدول مربوط را بررسی نمود.

جهت انجام عملیات DAQ در آزمایشگاه به کمک سیستم «ابزار مجازی» به موارد زیر نیاز دارید:

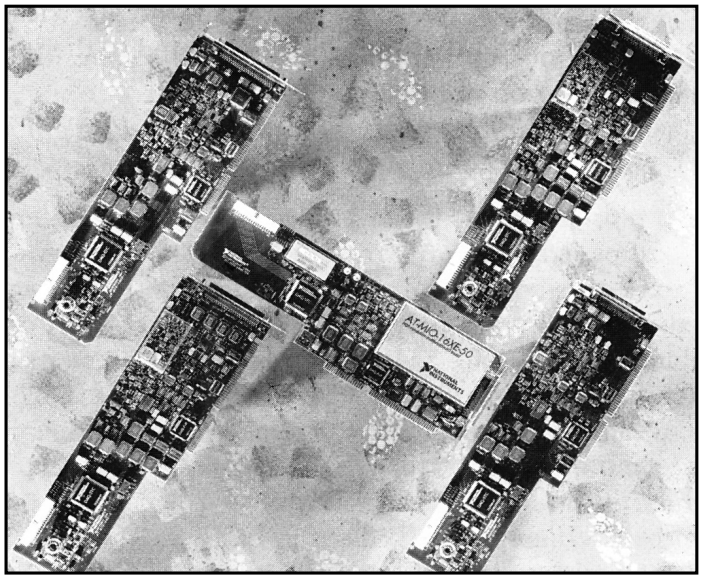
کارت DAQ، یک دستگاه رایانه به همراه نرم افزار LabVIEW، نرم افزار راه انداز[7] کارت های DAQ و برخی

3- Sensors
4- Transducers
5- Gauge
6- Signal Conditioning Modules
7- Driver

از ابزارهای لازم برای برقراری ارتباط بین سیگنال‌های تولید شده توسط مبدلها و کارت‌های DAQ نظیر مدول‌های تطبیق دهنده‌ی سیگنال، کارت‌های آزمایشگاهی، کابل یا سیم و... .

همان گونه که قبلاً ذکر شد استفاده از مدول‌های تطبیق دهنده به شرایط، نوع سیگنال و سطح ولتاژ سیگنال تولیدشده توسط مبدلها بستگی دارد. به عنوان مثال فرض کنید که کمیت فیزیکی مورد بحث، «دمای محیط» باشد. برای اندازه گیری دما توسط نرم‌افزار LabVIEW باید ابتدا سنسور گرمایی را به یکی از کانال‌های ورودی آنالوگ در کارت DAQ ارتباط دهیم. همان گونه که می دانید کارت‌های DAQ همانند کارت‌های دیگر بر روی کارت اصلی[8] رایانه نصب می شوند. توجه داشته باشید که جهت برقراری ارتباط مذکور در اغلب موارد استفاده از یک مدول تطبیق دهنده الزامی است. نوع این مدول نیز به سطح سیگنال ایجاد شده و نوع سنسور بستگی دارد. پس از انتخاب یک مدول مناسب، با استفاده از VIهای مربوط به DAQ که در نرم‌افزار LabVIEW تعبیه شده‌اند، می توان سطح ولتاژ کانال کارت DAQ را قرائت نمود. در مراحل بعدی و جهت تجزیه و تحلیل درجه حرارت اندازه گیری شده می توان داده‌ها و اطلاعات جمع آوری شده را بر روی صفحه به نمایش در آورد و این داده‌ها را در یک فایل ذخیره نمود و سپس به بررسی و تجزیه و تحلیل آنها پرداخت.

در شکل ۳-۲ تعدادی از کارت‌های DAQ که همگی محصول شرکت NI می باشند را ملاحظه می کنید. شایان ذکر است که پیکربندی تمامی این کارت‌ها با استفاده از نرم‌افزار انجام می‌شود و در رایانه‌های EISA و PCAT قابل استفاده‌اند.



**شکل ۳-۲: تعدادی از کارت‌های DAQ**

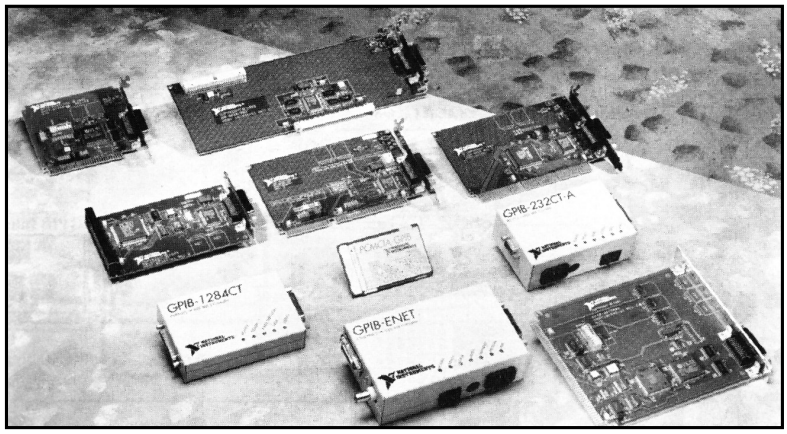
VIهای مورد استفاده جهت انجام فرآیند DAQ تنها با محصولات شرکت NI سازگارند. در صورت استفاده از محصولات شرکتهای دیگر حتماً باید از نرم‌افزار راه‌انداز آنها استفاده کنید. در غیر این صورت باید کدهای راه‌انداز مربوط را نوشته، سپس آنها رابه کمک DLL در نرم‌افزار LabVIEW فراخوانی کنید.

8- Main Board

## سیستم GPIB

شرکت Hewlet Packard در اواخر دهه‌ی ۱۹۶۰ برای تسهیل در امر برقراری ارتباط بین رایانه و ابزار و تجهیزات آزمایشگاهی، سیستم «خطوط ارتباطی همه منظوره» یا GPIB را ابداع نمود.

همان گونه که می دانید واژه‌ی bus به طور اختصار به سیستمی اطلاق می گردد که بین اجزای مختلف رایانه ارتباط برقرار می کند. در مورد GPIB نیز می توان گفت که سیستم خطوط ارتباطی همه منظوره بین رایانه و ابزار و تجهیزات اندازه گیری ارتباط برقرار می‌سازد. GPIB مسیری هموار جهت برقراری ارتباط مذکور فراهم نموده است. انجمن مهندسین برق و الکترونیک یا IEEE[9] در سال ۱۹۷۵ سیستم GPIB را به رسمیت شناخت و آن را به عنوان یک سیستم استاندارد معرفی نمود. از آن پس این سیستم به عنوان سیستم استاندارد IEEE 488.2 شناخته شد. هدف اصلی از به کارگیری سیستم GPIB، استفاده از رایانه به همراه تجهیزات آزمایشگاهی جهت پردازش داده هاست. کاربرد این سیستم از موارد مذکور فراتر رفته و در ارتباط دو رایانه با یکدیگر و همچنین برقراری ارتباط بین رایانه و مولتی متر، اسکنر، اسیلوسکوپ و... نیز به کار برده می شود. در شکل ۴-۲ تعدادی از این کارت ها را ملاحظه می کنید.




**شکل ۴-۲ : تعدادی از کارت‌های GPIB**

GPIB شامل یک باس موازی حاوی ۲۴ خط دیجیتال است. این باس مشتمل بر ۸ خط داده، ۵ خط کنترل باس (EOI, IFC, REN, SRQ, ATN)، ۳ خط Handshake و ۸ خط اتصال به زمین[10] می باشد. در سیستم GPIB از روشهای مختلفی نظیر ارسال ۸ بیت موازی، یک بایت سری و همچنین روش انتقال غیر همزمان داده ها استفاده می شود. به دلیل اینکه داده ها در این سیستم به صورت بایتی انتقال می یابند، پیام های ارسال شده اکثراً به صورت رشته های ASCII کدگذاری می شوند.

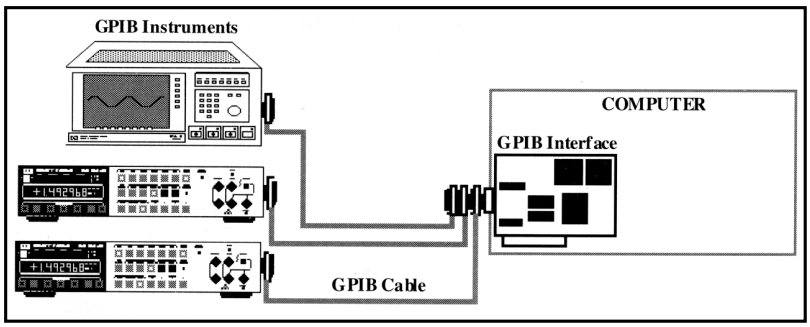
تنها در صورتی می توانید از روش ارتباط GPIB در رایانه‌ی خود استفاده نمایید که کارت یا جعبه‌ی GPIB[11] را

9- Institute of Electrical and Electronic Engeering
10- Ground

۱۱- این نام از آن جهت انتخاب شده است که المان ارتباط دهنده در داخل جعبه‌ای در خارج از رایانه واقع شده و از طریق درگاههای ورودی/خروجی با گیرنده یا فرستنده ارتباط برقرار می سازد.

به همراه نرم‌افزار راه‌انداز GPIB دراختیار داشته باشید. در این سیستم می توان چند رایانه و ابزار اندازه گیری را به باس GPIB متصل نمود. هر یک از تجهیزات یا هر وسیله ای که با کارت GPIB در ارتباط است، باید یک آدرس منحصر به فرد از شماره‌ی 0 الی 30 داشته باشد. به کمک این آدرس می توان مبدأ و مقصد داده ها را مشخص نمود. آدرس 0 معمولاً به کارت GPIB اختصاص داده می شود. برای تجهیزات مرتبط با باس GPIB می توان شماره های 1 الی 30 را در نظر گرفت.

جهت کنترل نمودن سیستم GPIB، از یک کنترل کننده که معمولاً همان رایانه‌ی شخصی است استفاده می شود. این کنترل کننده، عملکرد خطوط را کنترل می کند. کنترل کننده‌ی مذکور جهت انتقال داده ها و فرمانهای ارسال شده از تجهیزات و ابزار مرتبط با کارت GPIB، یکی از دو طرف را به عنوان فرستنده و دیگری را به عنوان گیرنده آدرس دهی می کند. سپس انتقال داده ها در طول باس از فرستنده به سمت گیرنده یا گیرنده ها صورت می گیرد. نظیر VIهای تعبیه شده در نرم افزار LabVIEW جهت انجام فرآیند DAQ، در این مورد نیز VIهای خاصی جهت به کارگیری سیستم GPIB در نظر گرفته شده است. این برنامه ها آدرس و عملکرد خطوط باس را به صورت خودکار بررسی می کنند. در شکل ۵-۲ نحوه‌ی ارتباط بین تجهیزات آزمایشگاهی و اندازه گیری با رایانه از طریق کارت GPIB نشان داده شده است.




**شکل ۵-۲: نحوه‌ی برقراری ارتباط بین تجهیزات آزمایشگاهی و اندازه‌گیری با رایانه از طریق کارت GPIB**

اگرچه در هر دو سیستم DAQ و GPIB از کارت هایی استفاده می شود که بر روی کارت اصلی رایانه قرار می گیرند و به کارگیری سیستم GPIB نیز روشی جهت انتقال داده ها به رایانه است، اما اساساً عملکرد این دو سیستم متفاوت می باشد. همان گونه که قبلاً عنوان شد سیستم GPIB برای برقراری ارتباط بین یک رایانه با یک ابزار اندازه گیری یا یک رایانه‌ی دیگر به کار برده می شود. حال آن که سیستم DAQ یک سیگنال تولید شده توسط مبدل را مستقیماً به کارت DAQ در رایانه مرتبط می سازد. در مورد این دو سیستم در فصول ۱۰ و ۱۱ به تفصیل به بحث خواهیم نشست. برای به کارگیری GPIB به عنوان بخشی از سیستم اندازه گیری و ابزار مجازی به موارد زیر نیاز دارید:

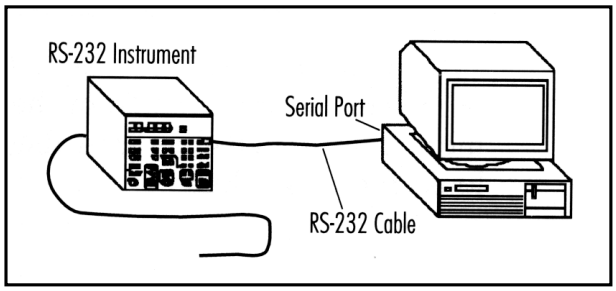
یک کارت یا جعبه‌ی خارجی GPIB، کابل ارتباطی مخصوص GPIB، نرم افزار LabVIEW، یک دستگاه رایانه و تجهیزاتی که با استاندارد IEEE 488.2 سازگار باشند.

شایان ذکر است که برای استفاده از کارت های GPIB حتماً باید نرم افزار راه انداز همان کارت را بر روی رایانه نصب نمایید. دستورالعمل نصب این نرم افزار به همراه کارت در اختیار شما قرار می گیرد.

VIهای مربوط به سیستم GPIB در این نرم‌افزار تنها با محصولات شرکت NI سازگارند. در صورتی که از تولیدات شرکتهای دیگر استفاده می‌کنید، باید کدهای راه‌انداز آن را بنویسید و آنها را به کمک DLL در LabVIEW فراخوانی کنید.

## برقراری ارتباط از طریق درگاههای سریال

برقراری ارتباط از طریق درگاههای سریال روشی مأنوس، قابل درک و عامه‌پسند جهت انتقال داده‌ها بین دو رایانه یا بین یک رایانه و یک ابزار قابل برنامه‌ریزی است. در این روش برای برقراری ارتباط، از درگاه سریال رایانه کمک گرفته می‌شود. در روش مذکور در هر لحظه تنها یک بیت از داده‌ها از طریق هر خط سریال بین فرستنده و گیرنده انتقال می‌یابد. از این روش در مواردی استفاده می‌شود که میزان سرعت ارسال داده‌ها پایین بوده، یا لازم است داده‌ها به فواصل دور انتقال یابند. سرعت انتقال اطلاعات در این روش پایین‌تر از سیستم GPIB بوده، همچنین ضریب اطمینان این روش نسبت به سیستم GPIB کمتر است. تنها مزیت روش ارتباط سریال نسبت به سیستم GPIB عدم نیاز به استفاده از کارت‌های ارتباطی GPIB است. همچنین لازم نیست تا ابزار و تجهیزات اندازه‌گیری مرتبط با رایانه با استاندارد IEEE 488.2 سازگار باشند. در شکل ۶-۲ یک سیستم ارتباط سریال نشان داده شده است.




**شکل ۶-۲: نحوه‌ی برقراری ارتباط بین تجهیزات آزمایشگاهی و اندازه‌گیری با رایانه از طریق درگاه سریال**

به دلیل اینکه در اکثر رایانه‌ها یک یا دو درگاه سریال تعبیه شده است، روش ارتباط سریال کاملاً در دسترس است و می‌توان بدون خرید تجهیزات و کارت‌های ارتباط دهنده‌ی خاص نظیر کارت‌های DAQ یا GPIB به تبادل داده‌ها پرداخت. در بسیاری از تجهیزات و ابزارهای آزمایشگاهی و اندازه‌گیری که در سیستم ارتباطی GPIB مورد استفاده قرار می‌گیرند، یک یا دو درگاه سریال نیز تعبیه شده است. بدین ترتیب می‌توان از آنها برای برقراری ارتباط به روش سریال نیز استفاده نمود. برخلاف سیستم GPIB، از طریق درگاه سریال تنها امکان برقراری ارتباط با یک ابزار آزمایشگاهی یا اندازه‌گیری وجود دارد که این مورد می‌تواند محدودیتی در برخی از کاربردهای صنعتی و اندازه‌گیری ایجاد نماید. همان گونه که اشاره شد سرعت انتقال داده‌ها در روش ارتباط سریال پایین بوده، همچنین فاقد قابلیت بررسی خطاهای ایجاد شده در انتقال داده‌هاست. با این وجود روش ارتباط سریال در برخی موارد بسیار کاربردی بوده و از نظر اقتصادی نیز کاملاً مقرون به صرفه است.

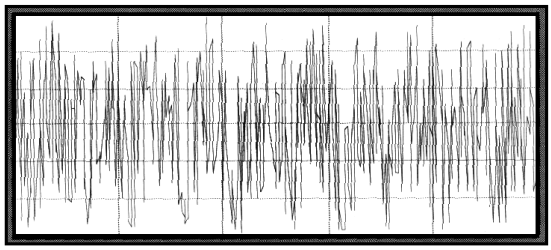
در نرم‌افزار LabVIEW توابع و دستورهای خاصی جهت به کارگیری درگاههای سریال تعبیه شده است. در صورتی که یک کابل سریال و یک ابزار اندازه گیری در اختیار دارید، می توانید نحوه‌ی برقراری ارتباط از طریق درگاههای سریال را بررسی نمایید.

## تجزیه و تحلیل داده‌ها

پس از انتقال داده‌ها به رایانه ممکن است قصد داشته باشید تا آنها را با یکی از روشهای موجود پردازش کنید. امروزه پردازنده‌های سیگنال دیجیتال به سیستم‌های تحلیلگر نیز مجهز شده‌اند. تعداد معدودی از برنامه‌های تحلیلی تعبیه شده در محیط LabVIEW نیز دارای قابلیت پردازش داده‌های پزشکی، سنتز، تشخیص صوت و پردازش تصویر هستند.

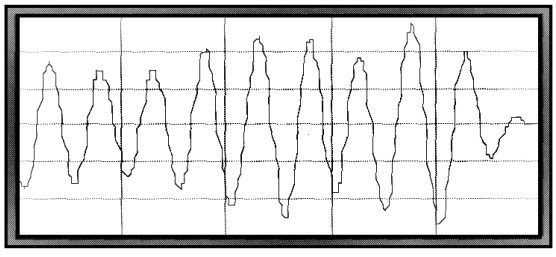
اهمیت به کارگیری و استفاده از توابع و برنامه‌های تحلیلگر در آزمایشگاههای تحقیقاتی، بر هیچ کس پوشیده نیست. داده‌های خام جمع‌آوری و ارسال شده توسط کارت‌های DAQ یا GPIB همواره حاوی اطلاعات و داده‌های مفید و سودمند نمی‌باشند. در اغلب موارد لازم است تا پس از ارسال سیگنال به رایانه، نویزها و اختلالات وارد شده را حذف کرده، اطلاعات و داده‌های از دست رفته توسط عملکرد نادرست دستگاهها و ابزارهای ارسال کننده یا تأثیر عوامل محیطی نظیر رطوبت، دما و... را تصحیح کنید.

شکل ۷-۲ را در نظر بگیرید. با ملاحظه‌ی این تصویر دلایل استفاده از توابع و دستورهای تحلیلگر را خواهید یافت.



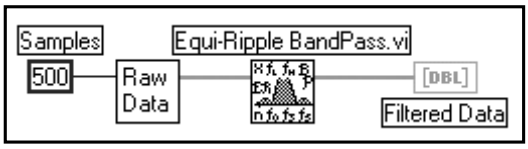
**شکل ۷-۲: داده‌های خام که شامل نویز و اطلاعات غیر مفید می‌باشند.**

با تحلیل و پردازش داده‌های دیجیتال می توان داده‌های مفید را از نویز جدا نمود و آنها را به صورتی قابل درک و معقول‌تر نسبت به داده‌های خام نمایش داد. پس از پردازش داده‌های نشان داده شده در شکل ۷-۲ می توان آنها را به صورت شکل ۸-۲ به نمایش درآورد.



**شکل ۸-۲: داده‌های پردازش شده پس از اعمال دستورهای تحلیلگر**

نحوه‌ی اعمال توابع و دستورهای تحلیلگر بر روی داده های خام و به دست آوردن داده های پردازش شده را در شکل ۹-۲ ملاحظه می کنید.




**شکل ۹-۲: نحوه‌ی اعمال توابع و دستورهای تحلیلگر بر روی داده‌های خام و به‌دست آوردن داده‌های پردازش شده**

همان گونـــه که در شکل ۹-۲ نشان داده شده است به دلیل اینکه دستـــورهای تحـــلیلگر در محیط LabVIEW، تکنیک ها و روشهای تجزیه و تحلیل داده ها را در VIهای جداگانه در اختیار شما قرار می دهند، می توان از آنها جهت آنالیز داده ها استفاده نمود. در محیط برنامه نویسی LabVIEW، به جای نگرانی در مورد جزییات برنامه های آنالیزکننده که اغلب گریبان گیر برنامه نویسان در اکثر زبانهای برنامه نویسی است، انرژی خود را بر روی حل مسائل و مشکلات موجود در مسیر آنالیز داده ها متمرکز کنید.

VIهای آنالیزکننده در محیط LabVIEW برای استفاده‌ی متخصصین و کارشناسان تعبیه شده است. با استفاده از این برنامه ها می توانید به کمک تکنیک های موجود در پردازش سیگنال دیجیتال یا DSP[12]، فیلترهای دیجیتالی، آمارگیری و آنالیز عددی، عملیات پردازش و آنالیز داده ها را انجام دهید. به کارگیری این برنامه ها جهت انجام محاسبات پیچیده برای افراد مبتدی نیز بسیار ساده است.

برنامه های آنالیزکننده[13] در LabVIEW برای پردازش داده ها به کار برده می شوند. در موارد زیر می توان از این برنامه ها برای تجزیه و تحلیل داده ها استفاده نمود:

- پردازش سیگنال های دیجیتال یا DSP
- طراحی فیلتر دیجیتال
- آنالیز آماری
- رسم نمودار
- جبر خطی
- آنالیز عددی

## مطالب مختصری در مورد VXI

باس VXI مخفف عبارت VMEbus eXtensions for Instrumentation است. این باس برای اولین بار در سال ۱۹۸۷ عرضه شد. VXI یک باس بسیار مفید جهت استفاده در سیستم های اندازه گیری است. این باس شامل یک جعبه‌ی اصلی[14] است که حاوی شکاف هایی برای قرار گرفتن ابزارهای مدولار بر روی کارت هاست.

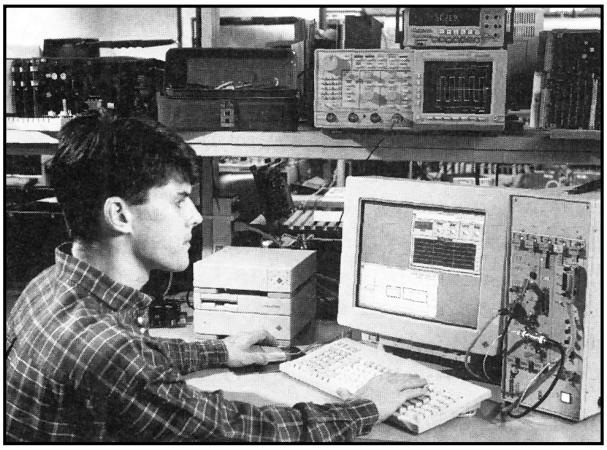
12- Digital Signal Processing

14- Main Frame

۱۳- تمامی برنامه‌های آنالیزکننده در زیرپالت Functions >> Analysis قرار دارند.

شرکتهای سازنده، جعبه های مذکور و ابزارهای مدولار را در اندازه های متفاوتی تولید می‌کنند و آنها را در اختیار کاربران قرار می دهند. در باس VXI می توان از مدول های VME نیز استفاده کرد. VXI کابردهای بسیاری در زمینه‌ی اندازه گیری و تست خودکار[۱۵] دارد.

دلیل عامه پسند بودن این باس، امکان به کارگیری آن به عنوان یک سیستم عامل جهت جمع آوری و تجزیه و تحلیل داده ها در موارد تحقیقاتی و اتوماسیون صنعتی است. بسیاری از کاربران، باس VXI را از طریق کارت های DAQ یا GPIB با سیستم های دیگر تلفیق نموده، از این مجموعه جهت مقاصد خاص و کاربردی خود استفاده می کنند. در شکل ۱۰-۲ نمونه ای از کاربرد سیستم VXI را ملاحظه می کنید.



**شکل ۱۰-۲ : در این تصویر، یک کارشناس در حال بررسی و اندازه‌گیری پاسخ فرکانسی به کمک نرم‌افزار LabVIEW و باس VXI است.**

VXI تعدادی از بالاترین تکنولوژی های به کار رفته در سیستم های DAQ، GPIB و رایانه های مدرن و پیشرفته‌ی امروزی را در خود جمع کرده است. این باس همانند سیستم GPIB، ابزار و تجهیزات قدرتمندی در اختیار کاربران قرار می‌دهد و همچون سیستم DAQ مدولار بودن، قابلیت انعطاف و ویژگیهای خاص دیگری را به ارمغان می آورد.

به دلیل اینکه در باس VXI، یک محیط اندازه گیری پیچیده با رایانه های مدرن امروزی تلفیق شده است، تجهیزات استفاده شده در این باس به دلیل استفاده از بهترین تکنیک های به کار برده شده در کارت های DAQ و GPIB از قابلیت برقراری ارتباط در سرعت های بسیار بالا برخوردار هستند.

VXI Plug & Play نامی است که به برخی از تولیدات VXI که دارای قابلیتهای استاندارد ویژه ای هستند اطلاق می شود. این تجهیزات به همراه یک نرم افزار استاندارد بوده، شامل صفحه‌ی پانل و نرم افزارهای راه‌انداز مخصوص این ابزارها می باشند. به کمک این نرم‌افزارهای استاندارد، برنامه‌نویسی بسیار ساده می گردد. از نرم افزار LabVIEW می توان جهت به کارگیری ابزار و تجهیزات VXI Plug & Play نیز استفاده نمود.

15- ATE : Automated Test Equipment

در لوح فشرده‌ی همراه این کتاب مطالب مفیدی در مورد این سیستم گنجانده شده است. برای مطالعه‌ی این مطالب، لوح فشرده را در داخل گرداننده قرار داده، سپس فایل Vxibook.pdf را از فهرست Manuals باز کنید. جهت باز کردن فایل‌های موجود در این فهرست به نرم‌افزار Acrobat Reader نیاز دارید. نرم‌افزار Acrobat Reader 5.0 نیز برای کمک به خوانندگان عزیز بر روی این لوح ذخیره شده است. بنابراین ابتدا این نرم‌افزار را نصب کرده، سپس مطالب مفید و ارزشمند موجود در لوح فشرده را مطالعه کنید.

۲

## نحوه‌ی برقراری ارتباط

در برخی موارد تقسیم داده‌ها بین چند رایانه و یا چند برنامه الزامی است. در نرم‌افزار LabVIEW توابع و دستورهایی در نظر گرفته شده است که انجام این اعمال را تسهیل می‌کند. این VIها برقراری ارتباط از طریق شبکه را به کمک فراخوانی کدهای خارجی[۱۶] و DLLها امکان‌پذیر می‌سازند.

## برقراری ارتباط از طریق شبکه

در این کتاب واژه‌ی «شبکه» برای برقراری ارتباط بین چند فرآیند به کار برده می‌شود. پس از برقراری ارتباط می‌توان آنها را به طور دلخواه بر روی رایانه‌های مجزا اجرا نمود. این ارتباط معمولاً از طریق شبکه‌ی سخت‌افزاری Ethernet یا Local Talk برقرار می‌گردد. یکی از کاربردهای اساسی شبکه آن است که امکان استفاده‌ی یک یا چندین برنامه را از سرویس و خدمات برنامه‌های دیگر فراهم می‌کند.

برای برقراری ارتباط بین چند فرآیند در حال اجرا، باید از یک زبان ارتباطی مشترک استفاده گردد که در اصطلاح به آن «پروتکل» گفته می‌شود. هر یک از پروتکل‌ها از استاندارد خاصی پیروی می‌کنند که عموماً با پروتکل‌های دیگر سازگار نیست. در ادامه اسامی چند پروتکل و موارد استفاده‌ی آنها آورده شده است:

- TCP: قابل استفاده در تمامی رایانه‌ها.
- UDP: قابل استفاده در تمامی رایانه‌ها.
- DDE: قابل استفاده در رایانه‌های شخصی برای برقراری ارتباط بین برنامه‌های تحت Windows.
- AppleEvents: قابل استفاده در محیط Mac جهت دریافت و ارسال اطلاعات بین برنامه‌های تحت سیستم Mac.

در فصل ۱۴ در مورد شبکه و ارتباط شبکه‌ای به تفصیل سخن خواهیم گفت.

16- External Codes

## DLL و CIN

برای افزایش قابلیتهای نرم‌افزار LabVIEW می‌توان از برنامه‌های کد شده‌ی متنی خارجی یا DLLها استفاده نمود و آنها را در اجرای برنامه به کار برد. DLL مجموعه‌ای از دستورها و توابع است که یک برنامه می‌تواند در زمان اجرا به آن متصل[17] شود.

نرم‌افزار LabVIEW همچنین از یک ساختار بلوکی خاص به نام CIN جهت برقراری ارتباط بین کدهای متنی و قراردادی با VIها استفاده می‌کند. در حالی که یک گره در حال اجراست، نرم‌افزار LabVIEW یک کد اجرایی را فراخوانی نموده، داده‌های ورودی را از کدهای اجرایی به سمت صفحه‌ی نمودار بلوکی عبور می‌دهد. به طریق مشابه در صورتی که برنامه‌های شما تحت Windows به اجرا در می‌آیند برای فراخوانی DLL می‌توانید از دستور Call Library استفاده کنید.

## پروتکل OLE[18]

پروتکل OLE یک زبان مشترک برای برقراری ارتباط بین برنامه‌های کاربردی است. به عنوان مثال به کمک این پروتکل استاندارد می‌توان یک برنامه‌ی صفحه‌گسترده نظیر Excel را با یک محیط واژه‌پرداز نظیر Word مرتبط ساخت. در این پروتکل از مدل Client/Server استفاده می‌شود. به طور کلی Client یا مصرف‌کننده برنامه‌ای است که برقراری ارتباط را درخواست نموده است. سرویس‌دهنده یا Server برنامه‌ی کاربردی دیگری است که با مصرف‌کننده در ارتباط است. برنامه‌های نوشته شده در محیط LabVIEW از نوع مصرف‌کننده هستند و با برنامه‌های کاربردی دیگر نظیر Word، Excel و Access ارتباط برقرار می‌سازند.

برقراری ارتباط بین LabVIEW و برنامه‌های کاربردی مذکور از طریق پروتکل OLE تنها در رایانه‌های شخصی و تحت سیستم عامل Windows امکان‌پذیر است.

## ابزارهایی که می‌توان به LabVIEW اضافه نمود

در نرم‌افزار LabVIEW برای افزایش قابلیت انعطاف و کارایی می‌توان از ابزارهای دیگری نیز استفاده نمود. برای کسب اطلاعات بیشتر در مورد ابزارهای مذکور با شرکت NI مکاتبه کنید.

برخی از ابزارهای مذکور عبارتند از:

- Application Builder
- Test Executive Toolkit
- JTFA Toolkit
- Digital Filter Design Toolkit
- Third Octave Analyzer Toolkit
- Signal Processing Suite
- Picture Control Toolkit
- SPC Toolkit
- SQL Toolkit-DatabaseVIEW

17- Link
18- Object Linking and Embedding

## خلاصه

دستورها و توابع داخلی در نظر گرفته شده در نرم‌افزار LabVIEW، برقراری ارتباط با ابزار و تجهیزات اندازه‌گیری و دنیای خارج را آسان‌تر می‌سازند. برخی از برنامه‌های بسته‌ی نرم‌افزاری LabVIEW جهت جمع‌آوری و تبادل داده‌ها می‌توانند از انواع سخت‌افزارهای جانبی استفاده نمایند. این سیستم‌های سخت‌افزاری عبارتند از: کارت‌های DAQ و GPIB، درگاههای سریال رایانه و سیستم VXI.

جهت خواندن و همچنین تولید سیگنال‌های ورودی آنالوگ، سیگنال‌های دیجیتال، زمان‌سنجی و شمارش از کارت‌های DAQ به همراه نرم‌افزار LabVIEW استفاده می‌شود. برقراری ارتباط از طریق سیستم GPIB و یا باس VXI نیز در نرم‌افزار LabVIEW امکان‌پذیر است. اگر هیچ یک از سخت‌افزارهای مذکور را در اختیار ندارید، جای نگرانی نیست؛ زیرا LabVIEW می‌تواند از طریق درگاههای سریال تعبیه شده در رایانه با دنیای خارج ارتباط برقرار سازد.

پس از انتقال داده‌ها به رایانه می‌توانید به کمک VIهای آنالیزکننده در LabVIEW به پردازش و تجزیه و تحلیل داده‌ها بپردازید. برای برقراری ارتباط بین دو رایانه که از طریق شبکه به یکدیگر مرتبط شده‌اند و یا ارتباط چندین برنامه در یک رایانه نیز می‌توانید از دستورها و توابع مخصوص برقراری ارتباط و اتصال در LabVIEW استفاده کنید. LabVIEW دارای VIهایی جهت پشتیبانی از چندین پروتکل شبکه بوده، همچنین قابلیت OLE داشته و می‌تواند DLL و کدهای متنی خارجی را فراخوانی کند.

در صورتی که قصد دارید تا قابلیتهای نرم‌افزار LabVIEW را افزایش دهید، می‌توانید ابزارهای مورد نظر را برای انجام اعمال خاص تهیه کنید و آنها را به این بسته‌ی نرم‌افزاری اضافه نمایید. ابزارهای مذکور جهت ایجاد برنامه‌های LabVIEW، طراحی فیلتر دیجیتال، آنالیز و تجزیه و تحلیل سیگنال‌ها، ایجاد تصاویر دلخواه، پردازش آماری، کنترل‌کننده‌ی PID و... به کار برده می‌شوند.

در فصل بعد به ذکر اصول برنامه‌نویسی در بسته‌ی نرم‌افزاری LabVIEW می‌پردازیم. بنابراین خود را برای نوشتن نمادها، علائم و کدهای خاص در محیط LabVIEW آماده سازید.

LabVIEW

# ۳

# محیط برنامه‌نویسی LabVIEW و نحوه‌ی ایجاد برنامه

## در این فصل با موارد زیر آشنا می‌شوید:

- صفحات پانل، نمودار بلوکی و بخشهای آیکن/کانکتور
- المان‌های کنترل و نشان‌دهنده و درک تفاوت بین آنها
- ترمینال‌های عناصر کنترل و نشان‌دهنده در صفحه‌ی نمودار بلوکی
- برنامه‌نویسی بر اساس اصل جریانِ داده
- منوهای پایین کشیدنی و کرکره‌ای در نرم‌افزار LabVIEW
- قابلیتها و موارد به‌کارگیری نوار ابزار، پالت‌های Tools، Controls، Functions و زیرپالت‌های موجود در آنها

# LabVIEW

# ۳

# محیط برنامه‌نویسی LabVIEW و نحوه‌ی ایجاد برنامه

در این فصل به بررسی محیط برنامه‌نویسی LabVIEW پرداخته، چگونگی عملکرد سه بخش اساسی یک برنامه یعنی پنجره‌ی پانل، نمودار بلوکی و آیکن/کانکتور را فرا می گیرید. هنگامی که سه بخش مذکور به طور مناسب با یکدیگر آماده بهره‌برداری شوند یک VI در اختیار دارید که می توان از آن به عنوان یک زیربرنامه در برنامه‌ی اصلی و یا در برنامه های دیگر استفاده نمود.

در این فصل در مورد منوهای پایین کشیدنی و کرکره‌ای، پالت‌های شناور، زیرپالت‌ها، نوار ابزار و نحوه‌ی استفاده از پنجره‌ی Help مطالبی خواهید آموخت. در پایان این فصل نیز در مورد امکانات و قابلیتهای زیربرنامه ها و دلیل استفاده از آنها سخن خواهیم گفت.

در این فصل اهداف زیر دنبال می شوند:

- آموزش محیط برنامه‌نویسی LabVIEW و شناخت صفحات پانل، نمودار بلوکی و بخشهای آیکن/کانکتور.
- شناخت المان های کنترل و نشان دهنده و درک تفاوت بین آنها.
- تشخیص تفاوت بین ترمینال های عناصر کنترل و نشان دهنده در صفحه‌ی نمودار بلوکی.
- فراگیری برنامه‌نویسی بر اساس اصل جریانِ داده.
- آشنایی با منو های پایین کشیدنی و کرکره ای در نرم‌افزار LabVIEW.
- درک قابلیتها و موارد به کارگیری نوار ابزار، پالت های Tools، Controls، Functions و زیرپالت های موجود در آنها.

- نحوه‌ی استفاده از پنجره‌ی Help در محیط LabVIEW.
- فراگیری ماهیت SubVI و اهمیت به کارگیری آن.
- انجام تمرینات اضافی در انتهای این فصل جهت درک ماهیت محیط برنامه نویسی LabVIEW.

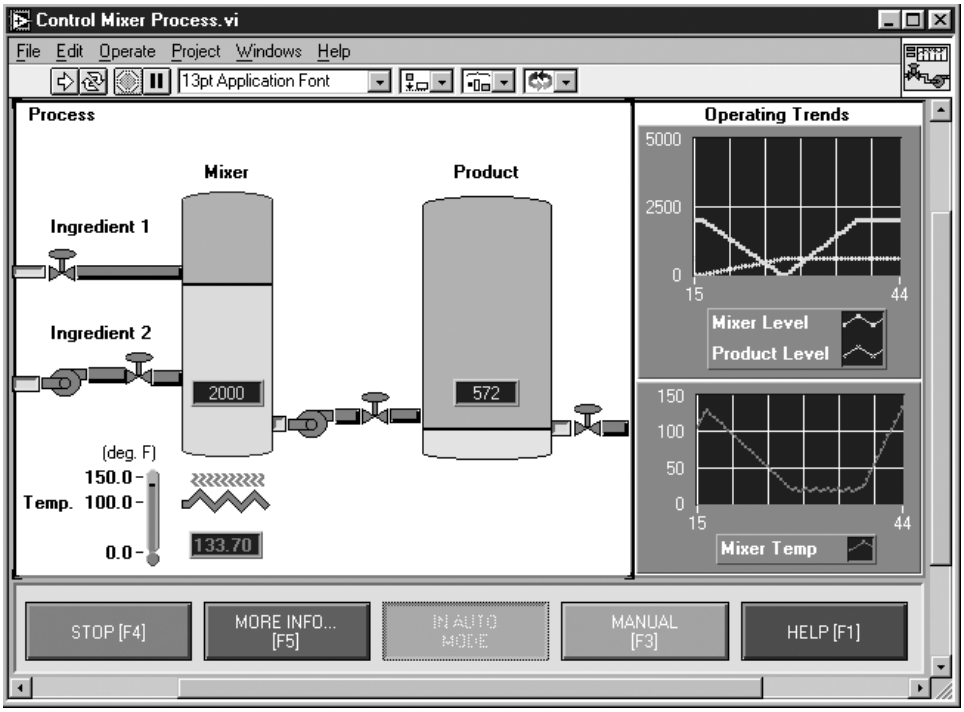
اصطلاحات کلیدی موجود در این فصل عبارتند از:

- کنترل (Control)
- نشان دهنده (Indicator)
- سیم (Wire)
- زیربرنامه (SubVI)
- ترمینال (Terminal)
- گره (Node)
- جریانِ داده (Data Flow)
- منوی پایین کشیدنی (Pull-down Menu)
- منوی کرکره ای (Pop-up Menu)
- پالت (Palette)
- زیرپالت (Subpalette)
- پنجره‌ی Help

## صفحه‌ی پانل

به بیان ساده می توان گفت که صفحه‌ی پانل، پنجره‌ای است که کاربر از طریق آن، با برنامه ارتباط برقرار می کند. برای اجرای یک VI ابتدا باید صفحه‌ی پانل آن را باز نموده، سپس ورودیهای لازم جهت اجرای برنامه را از طریق این پنجره وارد کنید. به دلیل اینکه این صفحه، محل نمایش خروجیهای برنامه است در بسیاری موارد ناگزیر به استفاده از آن هستید. در شکل ۱-۳ نمونه‌ای از صفحه‌ی پانل را مشاهده می کنید.

۳




**شکل ۱-۳: صفحه‌ی پانل در برنامه‌ی Control Mixer Process.vi**

## المان‌های کنترل و نشان‌دهنده

عناصر و المان‌های تشکیل دهنده‌ی صفحه‌ی پانل اصولاً کنترل‌ها و نشان‌دهنده‌ها هستند. کنترل‌ها، المان‌های ورودی نظیر کلیدها و سوییچ‌ها را که علائم قراردادی در مورد تجهیزات ورودی هستند شبیه سازی می‌کنند. کاربر به کمک المان‌های کنترل می‌تواند ورودیها را اعمال نماید.

هر یک از این عناصر دارای یک ترمینال بر روی صفحه‌ی نمودار بلوکی بوده، از این طریق داده‌های لازم برای ایجاد این صفحه را نیز فراهم می‌کنند. نشان‌دهنده‌ها نیز خروجیهای تولید شده توسط برنامه را نمایش می‌دهند. رابطه‌ی ساده‌ی زیر را برای درک ماهیت المان‌های کنترل و نشان‌دهنده در نظر بگیرید:

کنترل‌ها: ورودیهای اعمال شده از سوی کاربر = ترمینال‌های مبدأ
نشان‌دهنده‌ها: خروجیهای تولید شده برای کاربر = ترمینال‌های مقصد

به طور کلی این دو مفهوم تعویض‌پذیر نیستند. بنابراین حتماً تفاوت بین این دو گروه را به طور کامل درک کنید.

۳

با انتخاب این المان‌ها از زیرپالت‌های موجود در پالت Controls می‌توانید آنها را در محل مورد نظر بر روی صفحه‌ی پانل قرار دهید. پس از قرار دادن این المان‌ها بر روی صفحه‌ی پانل به راحتی می‌توان اندازه، شکل، محل، رنگ و خصوصیات دیگر آنها را تغییر داد.

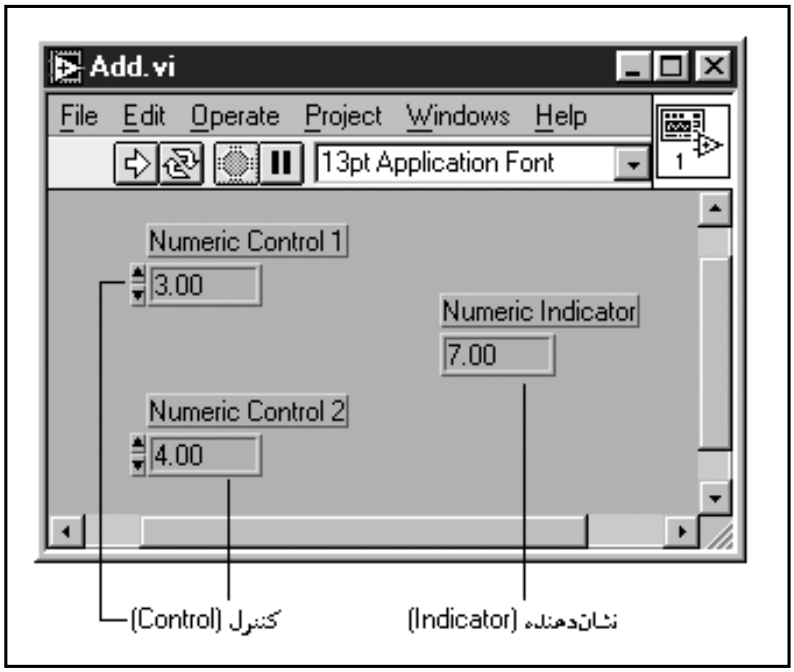
## صفحه‌ی نمودار بلوکی

پنجره‌ی نمودار بلوکی محل نگهداری و ذخیره‌ی کدها و علائم گرافیکی برنامه در محیط برنامه نویسی LabVIEW است.

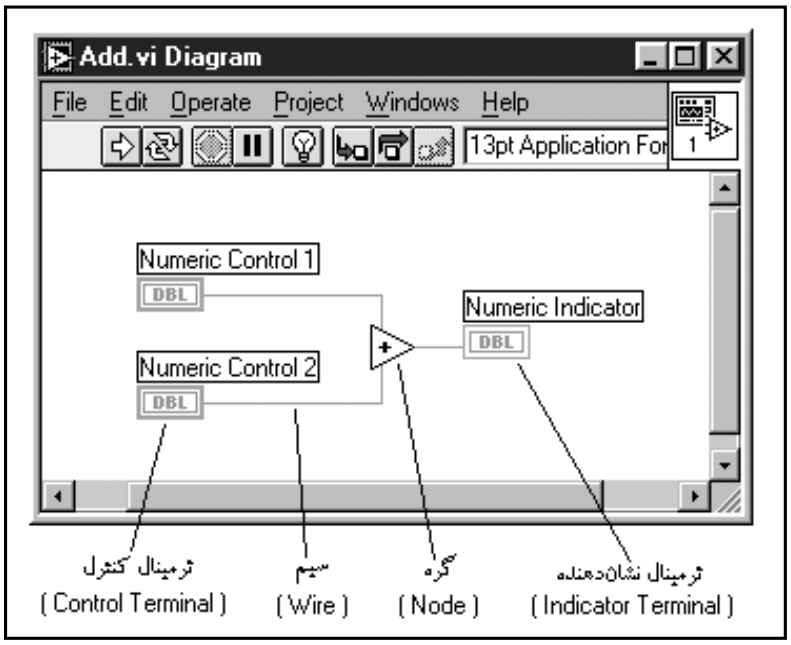
این کدها در زبان برنامه‌نویسی گرافیکی «G» نوشته می‌شوند. نمودار بلوکی به‌کار برده شده در محیط LabVIEW متناظر با سطرهای برنامه‌های متنی در زبانهای برنامه‌نویسی متداول و رایج نظیر BASIC و C می‌باشد. در حقیقت می‌توان گفت که این بلوک در برگیرنده‌ی کدهای اجرایی واقعی در محیط LabVIEW است.

در پیکربندی و ایجاد صفحه‌ی نمودار بلوکی باید المان‌هایی که در ارتباط با یکدیگر عمل خاصی را انجام می‌دهند، به یکدیگر سیم‌کشی نمود. در این بخش در مورد المان‌های نمودار بلوکی نظیر ترمینال‌ها، گره‌ها و سیمها به بحث خواهیم نشست.

برنامه‌ی ساده و ابتدایی نشان داده شده در شکل ۲-۳ برای محاسبه‌ی مجموع دو عدد به‌کار می‌رود. در شکل ۳-۳ صفحه‌ی نمودار بلوکی این برنامه را ملاحظه می‌کنید. به ترمینال‌های کنترل و نشان‌دهنده، گره و سیمهای موجود در این صفحه توجه کنید.




**شکل ۲-۳: صفحه‌ی پانل در برنامه‌ی Add.vi**




**شکل ۳-۳ : صفحه‌ی نمودار بلوکی در برنامه‌ی Add.vi**

## ترمینال ها

هنگامی که المان های کنترل یا نشان دهنده را بر روی صفحه‌ی پانل قرار می دهید، نرم افزار LabVIEW به صورت خودکار ترمینال های متناظر با آن ها را بر روی صفحه‌ی نمودار بلوکی ایجاد می کند. بنابراین

حذف ترمینال موجود در صفحه‌ی نمودار بلوکی که متناظر با یکی از المان های کنترل یا نشان دهنده باشد امکان پذیر نیست. ترمینال ایجاد شده در صفحه‌ی نمودار بلوکی تنها در صورتی حذف می گردد که المان متناظر آن یعنی کنترل یا نشان دهنده در صفحه‌ی پانل را حذف نمایید. برای درک تفاوت بین این دو نوع ترمینال، آنها را به عنوان درگاههای ورودی و خروجی و یا مبدأ و مقصد در نمودار بلوکی در نظر بگیرید.

حاشیه‌ی ترمینال های کنترل، پهن می‌باشد در حالی که حاشیه‌ی ترمینال های نشان دهنده باریک است. بنابراین تشخیص تفاوت بین آنها در هنگامی که از لحاظ عملیاتی و ساختاری یکسان نیستند، بسیار ساده است. در شکل ۴-۳ دو ترمینال کنترل و نشان دهنده نشان داده شده است. به تفاوت بین حاشیه‌ی آنها توجه کنید.

**شکل ۴-۳: تفاوت بین حاشیه های دو ترمینال کنترل و نشان دهنده**

به صفحه‌ی پانل برنامه‌ی Add.vi توجه کنید. اعدادی که در المان های Numeric Control 1 و Numeric Control 2 وارد می کنید، از صفحه‌ی پانل خارج می‌شوند و از طریق ترمینال‌های Numeric Control 1 و Numeric Control 2 که در صفحه‌ی نمودار بلوکی قرار گرفته‌اند به این بلوک اعمال می شوند. این داده‌ها از طریق سیم جریان پیدا می‌کنند و به ترمینال ورودی دستور Add وارد می شوند. هنگامی که دستور Add عملیات محاسبه را به پایان رسانید، یک عدد جدید در ترمینال خروجی خود ایجاد می کند. سپس عدد حاصل از طریق ترمینال Numeric Indicator عبور نموده، مجدداً به صفحه‌ی پانل بازگشته و بر روی صفحه نمایش داده می شود.

## گره‌ها

واژه‌ی گره یا Node تنها یک نام فرضی برای یک المان اجرایی برنامه است. گره‌ها متناظر با سطرهای برنامه، عملگرها، دستورها و زیربرنامه‌ها در زبانهای برنامه‌نویسی استاندارد می باشند. به عنوان مثال توابع Add و Subtract نمایش دهنده‌ی یک نوع گره هستند.

در نرم‌افزار LabVIEW، ساختارها[1] می توانند نظیر حلقه‌ها و حلقه های شرطی در زبانهای برنامه‌نویسی متداول و رایج نظیر BASIC و C، یک برنامه‌ی دلخواه را به طور متناوب به اجرا در آورند. بنابراین می توان ساختارها را نیز نوعی گره فرض کرد. در بسته‌ی نرم‌افزاری LabVIEW گره های خاصی تعبیه شده است که به آنها گره های محاسباتی یا گره های فرمولی[2] گویند. از این گره‌ها برای اعمال نمودن فرمول‌ها و اصطلاحات ریاضی استفاده می شود.

1- Structures
2- Formula Nodes

## سیمها

در نرم‌افزار LabVIEW برای برقراری ارتباط بین گره‌ها و ترمینال‌ها از سیم استفاده می شود. در حقیقت می توان گفت سیمها مسیر عبور داده‌ها را بین ترمینال های مبدأ[3] و مقصد[4] فراهم می سازند. سیمها، اطلاعات و داده‌ها را از یک ترمینال مبدأ به یک یا چند ترمینال مقصد انتقال می دهند. در صورتی که به یک سیم بیش از یک ترمینال مبدأ متصل شود و یا برای یک قطعه سیم، ترمینال مبدأ در نظر گرفته نشود، سیم مذکور به صورت شکسته[5] و منقطع نشان داده می شود و این بدان معنی است که برقراری ارتباط به درستی صورت نگرفته است.

روش برقراری ارتباط و سیم کشی بین ترمینال های مبدأ و مقصد دلیل غیر قابل تعویض بودن المان های کنترل و نشان دهنده را به خوبی روشن می سازد. بدین مفهوم که همواره المان های کنترل به عنوان ترمینال های مبدأ بوده، در حالی که عناصر نشان دهنده به عنوان ترمینال های مقصد تلقی می شوند.

هر یک از سیمها دارای شکل و رنگ خاصی می باشند. شکل و رنگ سیم به نوع اطلاعاتی که از طریق آن سیم انتقال می یابند بستگی دارد. در صفحه‌ی نمودار بلوکی نشان داده شده در شکل ۳-۳ ملاحظه نمودید که سیم حامل اعداد اسکالر به صورت یک خط باریک نمایش داده می شود. برای جلوگیری از بروز هرگونه اشتباه در هنگام سیم کشی ترمینال های مختلف در صفحه‌ی نمودار بلوکی، رنگ و شکل ظاهری سیم را برای هر نوع داده به خاطر بسپارید. در جدول ۱-۳ چند نوع سیم به همراه مشخصات خاص، از قبیل شکل و رنگ آنها نشان داده شده است.

۳

**جدول ۱-۳: مشخصات ظاهری سیمها که برای انتقال داده‌های مختلف به کار برده می شوند.**

| نوع داده | رنگ | عدد اسکالر | بردار یک بعدی | بردار دو بعدی |
|---|---|---|---|---|
| عدد اعشاری | نارنجی | | | |
| عدد صحیح | آبی | | | |
| مقدار جبری | سبز | | | |
| مقدار رشته‌ای | ارغوانی | | | |

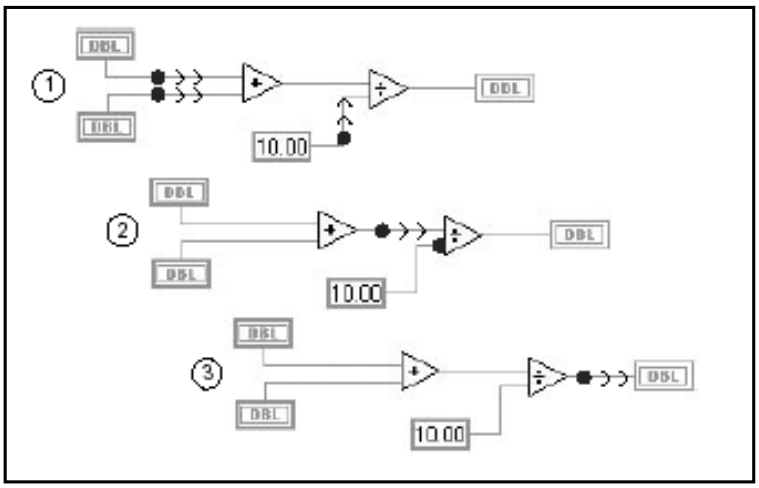
## برنامه نویسی به روش جریان داده

به دلیل اینکه نرم افزار LabVIEWیک محیط برنامه نویسی متنی نیست، بنابراین نمادها و کدهای آن نظیر زبانهای برنامه نویسی متنی سطر به سطر اجرا نمی شوند. قاعده‌ای که منجر به اجرای دستورها در زبان برنامه‌نویسی گرافیکی «G» می شود، اصل جریانِ داده خوانده می شود. به عبارت دیگر در برنامه‌نویسی به روش جریان داده،

3- Source Terminal
4- Sink or Destination Terminal
5- Broken

یک گره در حالتی قابل اجراست که داده‌ها و اطلاعات به تمامی ترمینال‌های ورودی آن اعمال گردد.

پس از اجرای عملیات و دستورهای خاص درگره، داده‌ها و نتایج حاصله در ترمینال‌های خروجی گره ظاهر می‌شوند. در برنامه‌نویسی به روش جریان داده، مقادیر داده‌ها بلافاصله از ترمینال‌های مبدأ به سمت ترمینال‌های مقصد جریان می‌یابند. این مطلب در شکل ۵-۳ نشان داده شده است.




**شکل ۵-۳: مفهوم اصل جریان داده**

۳

روش جریان داده کاملاً با روش اجرای برنامه‌های متنی که به «کنترلِ داده[6]» موسوم است تفاوت دارد. در روش کنترل داده ترتیب اجرای دستورها برحسب تقدم و تأخر آنها در سطرهای نوشته شده در برنامه است. اختلاف مذکور ممکن است باعث شود تا در موارد بسیاری، نحوه‌ی به‌کارگیری دو روش جریانِ داده و کنترلِ داده نیز متفاوت باشد. زیرا روند اجرای برنامه‌های متنی متداول و رایج برحسب ترتیب قرار گرفتن دستورهاست، در حالی که اجرای برنامه به روش جریان داده به داده‌ها وابسته است.

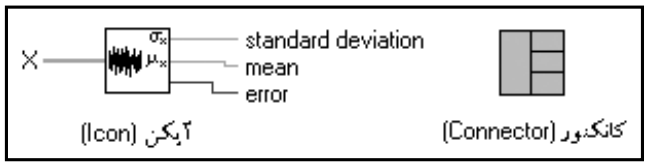
## آیکن/کانکتور

هنگامی‌که از یک VI به عنوان زیربرنامه در برنامه‌های دیگر استفاده می‌شود، المان‌های کنترل و نشان‌دهنده داده‌ها را از برنامه‌ی مرحله‌ی بالاتر که در آن، فراخوانی زیربرنامه صورت گرفته است، دریافت می‌کنند و نتایج حاصله را نیز به همان برنامه باز می‌گردانند. آیکن در واقع نمایانگر یک VI به عنوان زیربرنامه در صفحه‌ی نمودار بلوکی برنامه‌های دیگر است. یک آیکن ممکن است شامل نمادگرافیکی یا توضیح نوشتاری مختصری از برنامه و یا ترکیبی از هر دو مورد باشد.

دستورها و توابع مربوط به کانکتور در برنامه بسیار شبیه به لیست پارامترها در حین فراخوانی برنامه در زبانهای C یا Pascal هستند. ترمینال‌های کانکتور تا اندازه‌ای شبیه به پارامترهای گرافیکی برای انتقال داده‌ها به/از یک زیربرنامه عمل می‌کنند. هر ترمینال، با یکی از المان‌های کنترل یا نشان‌دهنده‌ی موجود در صفحه‌ی

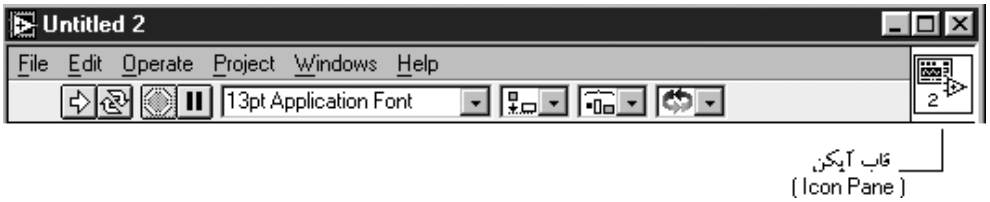
6- Control Flow

پانل متناظر است. در حین فراخوانی زیربرنامه، پارامترهای ورودی به کنترل‌های مربوط انتقال یافته، سپس زیربرنامه به اجرا در می‌آید. در پایان اجرای برنامه، مقادیر نشان دهنده‌ها نیز به پارامترهای خروجی در کانکتور انتقال می‌یابند. در شکل ۶-۳ یک آیکن را به همراه کانکتور متناظر با آن ملاحظه می‌کنید.




**شکل ۶-۳: تصویر یک زوج آیکن/کانکتور**

در هنگام ایجاد هر برنامه یک آیکن به صورت پیش‌فرض برای آن در نظر گرفته می‌شود. این آیکن در گوشه‌ی بالایی سمت راست صفحات پانل و نمودار بلوکی و در قاب آیکن به نمایش در می‌آید. در شکل ۷-۳ تصویر پیش‌فرض یک آیکن را ملاحظه می‌کنید.




**شکل ۷-۳: تصویر پیش‌فرض یک آیکن که در قاب آیکن به نمایش در می‌آید.**

قاب کانکتور[7] مربوط به هر برنامه در زیر قاب آیکن[8] آن قرار گرفته است. برای مشاهده‌ی قاب کانکتور، منوی کرکره‌ای را بر روی قاب آیکن در صفحه‌ی پانل باز نموده، سپس گزینه‌ی Show Connector را انتخاب کنید. در آینده در مورد منوی کرکره‌ای به طور کامل توضیح خواهیم داد. هنگامی که برای اولین بار یک کانکتور را ظاهر می‌سازید، نرم‌افزار LabVIEW یک الگو برای کانکتور مورد بحث پیشنهاد می‌دهد که در آن برای هر یک از المان‌های کنترل و نشان‌دهنده‌ی موجود در صفحه‌ی پانل، یک ترمینال منحصربه فرد در نظر گرفته شده است. در صورتی که به الگوی دیگری نیاز دارید می‌توانید آن را انتخاب نمایید. در هر کانکتور می‌توان حداکثر از ۲۸ ترمینال استفاده نمود.

## تمرین ۱-۳: آغاز برنامه‌نویسی

بسیار خوب! دیگر مطالعه کافی است. حال زمان آن رسیده تا کمی تمرین کنید. در این تمرین قصد داریم تا با ایجاد یک برنامه‌ی ساده در محیط LabVIEW با صفحات پانل، نمودار بلوکی و بخشهای آیکن/کانکتور آشنا شویم. این برنامه، یک عدد تصادفی تولید نموده، سپس مقدار آن را بر روی نمودار ترسیم می‌کند. در فصل بعدی در مورد مراحل نوشتن و اجرای برنامه مطالب بیشتری فرا خواهید گرفت؛ اما اکنون تنها قصد داریم تا شما را با محیط برنامه‌نویسی LabVIEW آشنا سازیم.

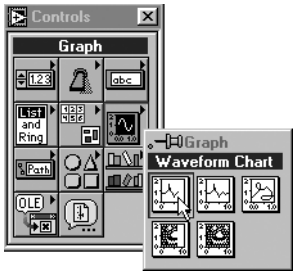
در صورتی که هنوز نرم‌افزار LabVIEW را بر روی رایانه‌ی خود نصب نکرده‌اید، این عمل را انجام دهید. اگر

7- Connector Pane
8- Icon Pane

بسته‌ی نرم‌افزاری LabVIEW در محیط Windows نصب شده است، پس از بارگذاری این نرم‌افزار، بر روی کلید New VI در پنجره‌ی محاوره‌ای کلیک کرده، خود را برای نوشتن یک برنامه‌ی جدید آماده کنید.

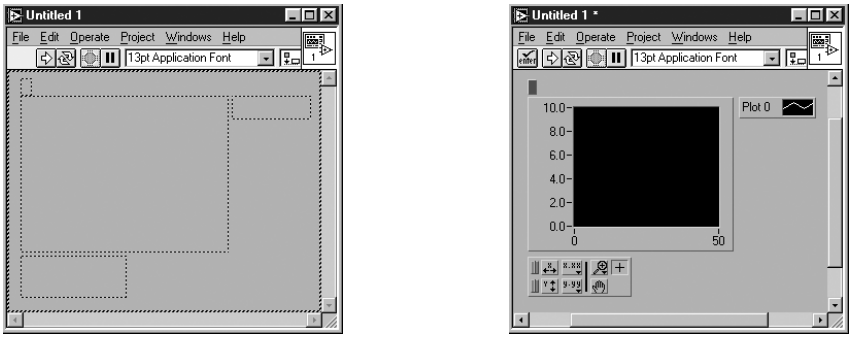
اگر در مورد انجام تمرینات این فصل با مشکل مواجه می‌شوید، به شما پیشنهاد می‌کنیم که ابتدا مطالب فصل ۴ را مطالعه نموده، سپس به این بخش بازگردید.

۱- در این حالت یک پنجره‌ی پانل با عنوان Untitled 1 بر روی صفحه‌ی نمایشگر رایانه ظاهر می‌گردد. اکنون برای دستیابی به زیرپالت‌های موجود در پالت Controls بر روی یکی از زیرپالت‌های آن کلیک کنید. در صورتی که پالت Controls بر روی صفحه‌ی نمایش وجود نداشته باشد، برای ظاهر ساختن آن گزینه‌ی Windows >> Show Controls Palette را انتخاب نمایید. در این حالت از فعال بودن صفحه‌ی پانل اطمینان حاصل کنید. زیرا در صورت فعال بودن صفحه‌ی نمودار بلوکی، به جای پالت Controls، پالت Functions ظاهر می‌گردد. اکنون ماوس را بر روی زیرپالت‌های موجود در پالت Graph حرکت دهید و با کلیک کردن بر روی Waveform Chart، آن را انتخاب کنید. همان گونه که در شکل ۸-۳ ملاحظه می‌کنید با حرکت دادن و عبور ماوس بر روی آیکن‌های قرار گرفته در پالت Controls و زیرپالت های آن، نام آیکن یا دکمه‌ی انتخاب شده در بالای پنجره‌ی پالت به نمایش در می‌آید. در شکل ۸-۳ نحوه‌ی انتخاب Waveform Chart را ملاحظه می‌کنید.




**شکل ۸-۳: نحوه‌ی انتخاب Waveform Chart**

حال نشانگر ماوس را به نقطه‌ای دلخواه در صفحه‌ی پانل ببرید و کلیک کنید. نمودار شکل موج یا Waveform Chart مطابق شکل ۹-۳ در محل انتخاب شده ظاهر خواهد شد.

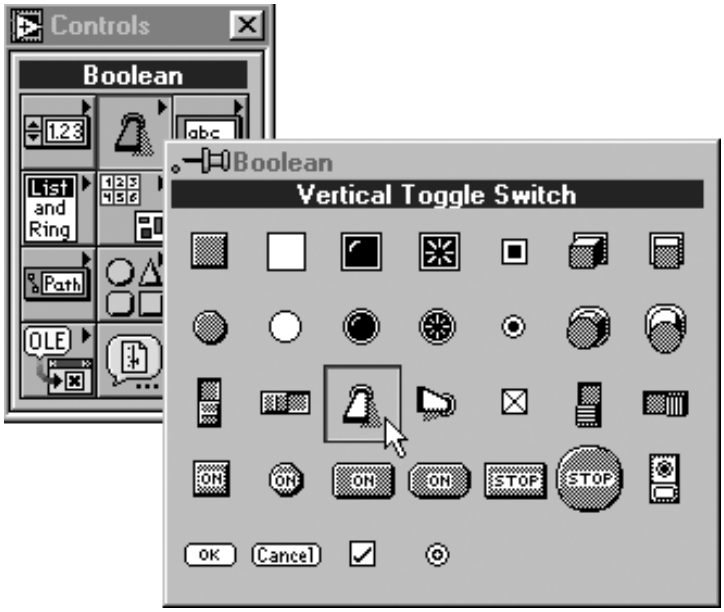



**نمای ظاهری نمودار پس از کلیک کردن بر روی صفحه‌ی پانل  نمای ظاهری نمودار قبل از کلیک کردن بر روی صفحه‌ی پانل**
**شکل ۹-۳**

Positioning Tool

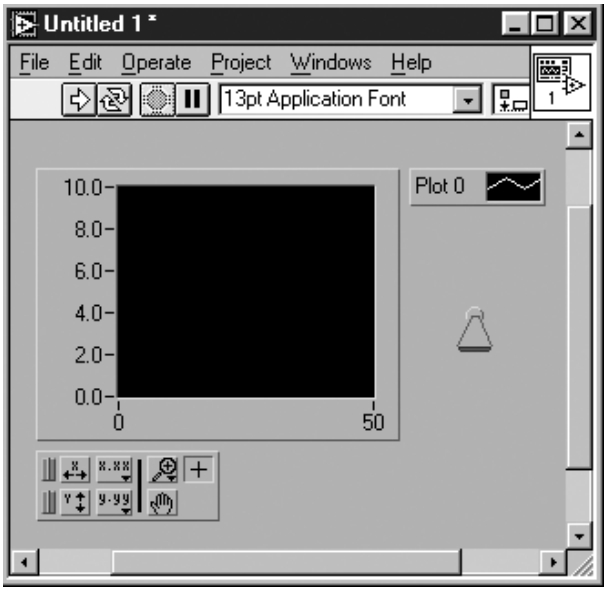
برای جابه‌جا کردن نمودار ابتدا ابزار PositioningTool را از پالت Tools انتخاب کرده، سپس به کمک آن، نمودار شکل موج را به مکان جدید حرکت دهید. در صورت عدم وجود پالت Tools بر روی صفحه‌ی نمایشگر رایانه، گزینه‌ی Windows >> Show Tools Palette را انتخاب کنید و بدین ترتیب آن را ظاهر سازید.

۲- در این حالت به پالت Controls بازگردید و مطابق شکل ۱۰-۳، کلید Vertical Toggle Switch را از زیرپالت Boolean >> Functions انتخاب کنید.




**شکل ۱۰-۳: نحوه‌ی انتخاب المان Vertical Toggle Switch**

مطابق شکل ۱۱-۳ این کلید را در مجاورت نمودار قرار دهید.




**شکل ۱۱-۳**

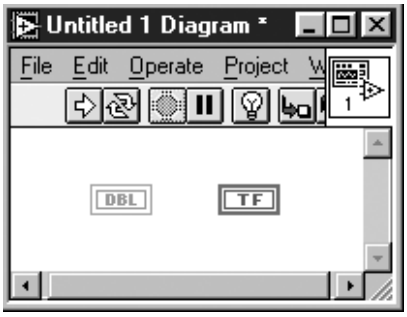
۳

۳- ابزار Operating Tool را از پالت Tools انتخاب کنید.

**شکل ۱۲-۳: نحوه‌ی انتخاب ابزار Operating Tool از پالت Tools**

حال به کمک ابزار انتخاب شده درجه‌بندی[9] یا حد بالایی نمودار را تغییر دهید. برای انجام این عمل، با دو بار کلیک کردن و یا کلیک کردن وکشیدن ماوس بر روی عدد 10.0، آن را متمایز کنید. اکنون عدد 1.0 را تایپ نموده، بر روی دکمه‌ی Enter که در نوار ابزار قرار دارد کلیک کنید.

۴- حال با انتخاب گزینه‌ی Windows >> Show Diagram، صفحه‌ی نمودار بلوکی این برنامه را بررسی نمایید. همان گونه که ملاحظه می کنید این صفحه با عنوان «Untitled 1 Diagram*[10]» ظاهر می گردد. در صفحه‌ی نمودار بلوکی این برنامه حتماً باید دو ترمینال نشان داده شده در شکل ۱۳-۳ را ببینید. دو ترمینال TF و DBL به ترتیب مربوط به کلید و گراف می باشند.




**شکل ۱۳-۳: ترمینال‌های TF و DBL مربوط به کلید و گراف که بر روی صفحه‌ی نمودار بلوکی دیده می‌شوند.**

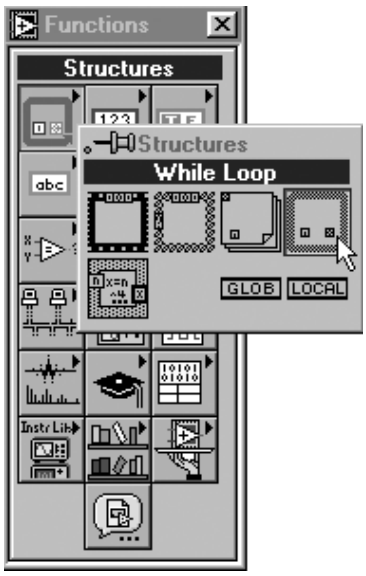
از این به بعد به عنوان قرارداد می پذیریم که به طور مثال اگر لازم باشد زیرپالت Structures را از پالت Functions انتخاب کنید به اختصار گوییم: زیرپالت Functions >> Structures را انتخاب نمایید.

۵- حال این دو ترمینال را در داخل چهارچوب یک حلقه‌ی While قرار دهید تا اجرای این بخش از برنامه به طور متناوب تکرار شود. برای انجام این عمل مطابق شکل ۱۴-۳ آیکن While Loop را از زیرپالت Functions >> Structures انتخاب کنید. در صورت عدم وجود پالت Functions بر روی صفحه‌ی

9- Scale

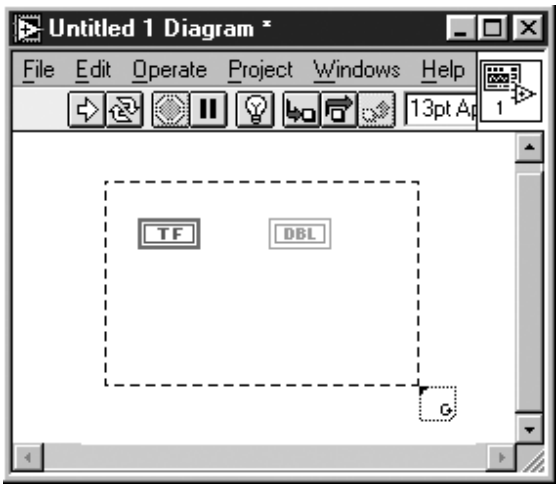
۱۰- کاراکتر «*» در نوار عنوان بیانگر آن است که کاربر در برنامه تغییراتی ایجاد نموده اما هنوز آن را ذخیره نکرده است.

نمایشگر رایانه، گزینـه‌ی Windows >> Show Functions Palette را انتخاب کنید و بدین ترتیب آن را ظاهر سازید. در این حالت از فعال بودن صفحه‌ی نمودار بلوکی اطمینان حاصل کنید. زیرا در صورت فعال بودن صفحه‌ی پانل، پالت Controls را به جای پالت Functions مشاهده خواهید کرد.




**شکل ۱۴-۳: نحوه‌ی انتخاب حلقه‌ی While**

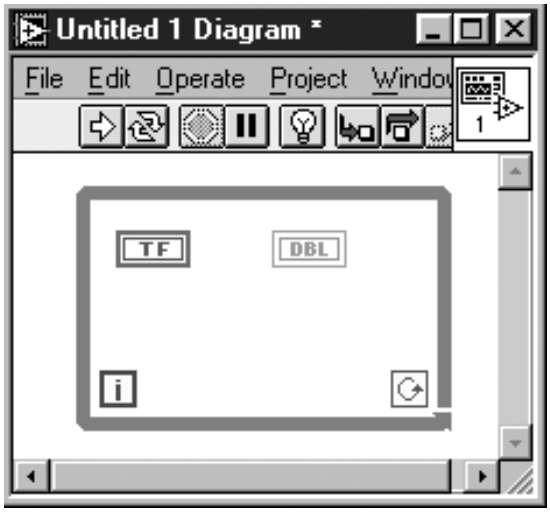
در این حالت، نشانگر ماوس به شکل یک «حلقه‌ی کوچک» ظاهر می گردد. اکنون دو ترمینال TF و DBL را در داخل چهارچوب حلقه قرار دهید. برای انجام این عمل مطابق شکل ۱۵-۳ در حالی که دکمه‌ی ماوس را پایین نگه داشته اید، ماوس را به صورت قطری و از گوشه‌ی بالایی سمت چپ صفحه به گوشه‌ی پایینی سمت راست آن حرکت دهید به صورتی که دو ترمینال مذکور درون چهارچوب حلقه قرار گیرند.




**شکل ۱۵-۳**

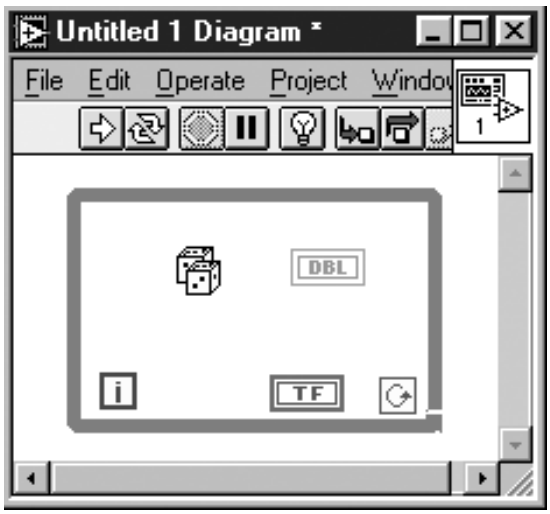
به محض رها نمودن دکمه‌ی ماوس، نوار خط چین که پیرامون دو ترمینال مورد بحث قرار گرفته است

طبق شکل ۱۶-۳ به چهارچوب حلقه‌ی While تبدیل می‌گردد. برای قرار دادن المان‌های دیگر در حلقه، مقداری فضای خالی در نظر بگیرید.




شکل ۱۶-۳

۶- در این حالت آیکن Random Number (0-1) << Numeric << Functions را انتخاب کنید. سپس مطابق شکل ۱۷-۳ آن را در داخل چهارچوب حلقه‌ی While قرار دهید.




شکل ۱۷-۳

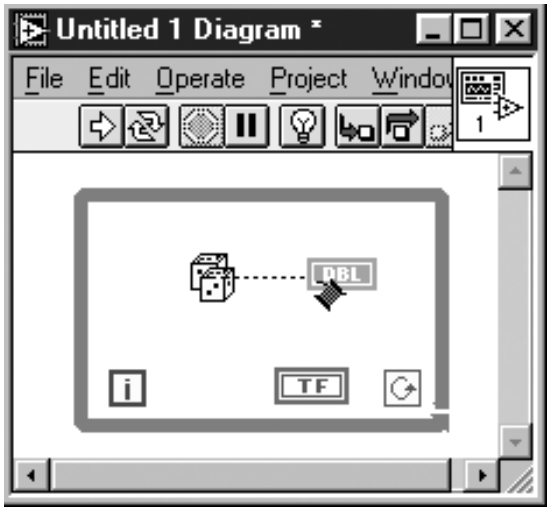
حلقه‌ی While یکی از ساختارهای خاص محیط برنامه‌نویسی LabVIEW است که کدها و دستورهای محصور در چهارچوب حلقه را تا زمانی که مقدار False دریافت نکرده باشد، به طور مداوم تکرار می‌کند. این حلقه معادل با حلقه‌ی DO-While در زبانهای برنامه‌نویسی متداول و رایج است. در فصل ۶ مطالب بیشتری درمورد حلقه‌ها فرا خواهید گرفت.

Positioning Tool

۷- با انتخاب ابزار Positioning Tool از پالت Tools المان‌های موجود در حلقه‌ی While را مطابق شکل ۱۷-۳ مرتب کنید.

Wiring Tool

۸- حال برای برقراری ارتباط بین ترمینال‌های موجود در حلقه، ابزار Wiring Tool را از پالت Tools انتخاب کنید. به کمک این ابزار یک بار بر روی آیکن Random Number (0-1) کلیک کنید و ماوس را به سمت ترمینال DBL حرکت دهید. سپس مطابق شکل ۱۸-۳ یک بار نیز بر روی ترمینال DBL کلیک کنید.



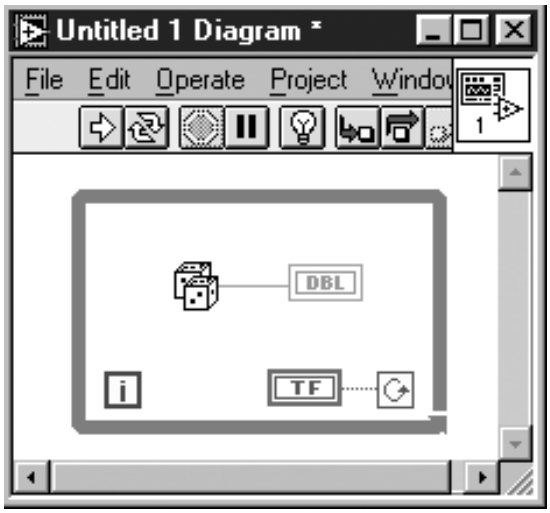

**شکل ۱۸-۳: نحوه‌ی سیم‌کشی بین آیکن Random Number (0-1) و ترمینال DBL**

اکنون باید یک سیم نارنجی رنگ که دو آیکن مذکور را به یکدیگر مرتبط کرده است بر روی صفحه ظاهر شده باشد. در صورتی که موفق به انجام این کار نشده‌اید و یا به عبارت دیگر سیم‌کشی را به خوبی انجام نداده‌اید، به کمک ابزار Positioning Tool تمام طول سیم یا یک قطعه از آن را انتخاب نموده، سپس با فشار دادن کلید <Delete> قطعه سیم موردنظر را حذف کنید. پس از سیم‌کشی و برقراری ارتباط بین دو آیکن مورد بحث، بار دیگر با انتخاب ابزار Wiring Tool ترمینال جبری TF را به ترمینال شرطی[11] در حلقه‌ی While متصل کنید. اکنون باید یک قطعه سیم سبز رنگ نقطه‌چین که ترمینال جبری TF را به ترمینال شرطی حلقه مرتبط کرده است بر روی صفحه ظاهر شده باشد.

این حلقه تا زمانی اجرا می‌شود که کلید قرار گرفته درصفحه‌ی پانل در حالت True یا در وضعیت بالا قرار گرفته باشد. به محض تغییر وضعیت این کلید به حالت False، اجرای حلقه متوقف می‌گردد. صفحه‌ی تکمیل شده‌ی نمودار بلوکی این برنامه در شکل ۱۹-۳ نشان داده شده است.

۳

11- Conditional Terminal

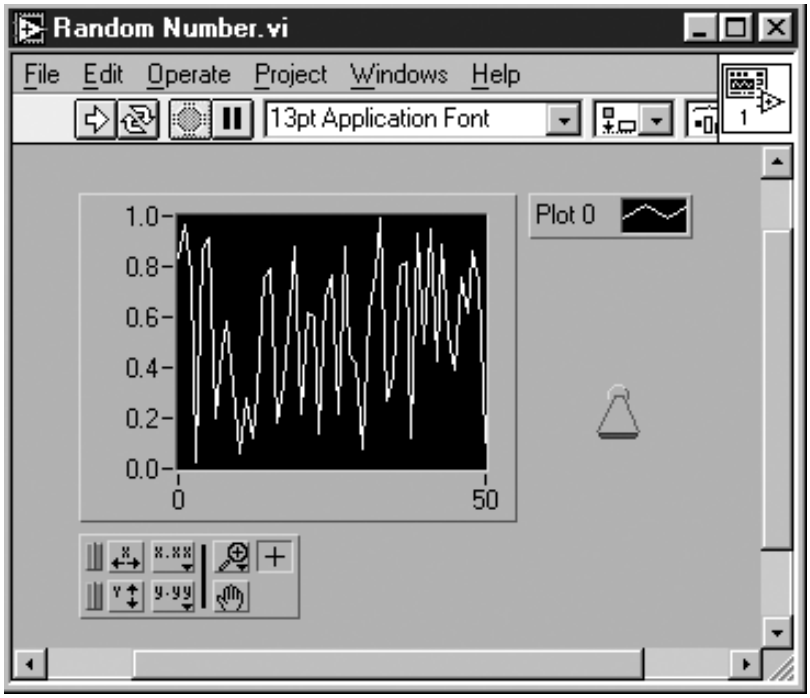



شکل ۱۹-۳: صفحه‌ی تکمیل شده‌ی نمودار بلوکی برنامه

Operating Tool

Run Button

اکنون برای اجرا کردن این برنامه آماده شوید. با انتخاب گزینه‌ی Windows >> Show Panel به صفحه‌ی پانل بازگردید. با استفاده از ابزار Operoting Tool، کلید را در وضعیت بالا قرار دهید. حال برای اجرا نمودن برنامه بر روی دکمه‌ی Run کلیک کنید. مطابق شکل ۲۰-۳ ملاحظه می کنید که مجموعه‌ای از اعداد تصادفی بر روی صفحه‌ی نمودار در پنجره‌ی پانل ترسیم می شوند. برای متوقف کردن اجرای برنامه، وضعیت کلید را به کمک ابزار Operating Tool تغییر دهید.




شکل ۲۰-۳

۱۰- در فهرست LabVIEW یک زیرفهرست به نام MY Activity ایجاد کنید. سپس با انتخاب گزینه‌ی File >> Save این برنامه را با عنوان Random Number.vi در زیرفهرست LabVIEW\My Activity ذخیره کنید.

برای دسترسی سریع و آسان به برنامه‌ها و تمرینات نوشته شده تمامی آنها را در زیرفهرست LabVIEW\My Activity ذخیره کنید. در صورتی که نرم‌افزار LabVIEW را تحت Windows 3.1 به‌کار می‌برید، به جای ایجاد زیرفهرست، یک کتابخانه با عنوان MyAct ایجاد نمایید (به یاد داشته باشید که نام فایل‌ها در این محیط حداکثر شامل ۸ کاراکتر است). کتابخانه‌ها محل ذخیره‌ی VIها هستند و به شما اجازه می‌دهند تا محدودیت‌های سیستم عامل نظیر نام‌گذاری فایل‌ها با حداکثر ۸ کاراکتر را در نرم‌افزار Windows 3.1 مرتفع سازید. فایل‌های کتابخانه‌ای را در فصل ۵ مورد بررسی قرار خواهیم داد. برای ایجاد یک کتابخانه و ذخیره‌ی این برنامه، گزینه‌ی File >> Save را انتخاب کنید. مطئمن شوید که حتماً در فهرست LabVIEW قرار دارید. حال با کلیک کردن بر روی دکمه‌ی New VI Library در پنجره‌ی محاوره‌ای گزینه‌ی Save، یک کتابخانه‌ی جدید ایجاد کنید. نام این کتابخانه را MyAct.llb انتخاب نموده، بر روی دکمه‌ی VI Library کلیک کنید. اکنون کتابخانه‌ای با عنوان MyAct.llb در اختیار دارید و می‌توانید این برنامه را در آن ذخیره کنید.

۳

برای حصول اطمینان از صحت عملکرد برنامه‌های نوشته شده، جوابهای خود را با پاسخ صحیح تمرینات که در فهرست Exercise گنجانده شده‌اند مقایسه کنید.

به شما تبریک می‌گوییم. زیرا که موفق شده‌اید تا اولین برنامه‌ی خود را در محیط LabVIEW بنویسید. به زودی قادر خواهید بود تا برنامه‌های قوی‌تر و با کاربرد بیشتری را در این محیط بنویسید و آنها را برای مقاصد و اهداف کاربردی خود مورد استفاده قرار دهید.

## منوهای پایین کشیدنی

قابلیتها و توانمندی‌های نرم‌افزار LabVIEW بسیار زیاد و متنوع است. در این کتاب سعی شده تا حد امکان تعدادی از قابلیتهای این برنامه را مورد بحث قرار دهیم. در این مجموعه سعی داریم تا سرعت شما را در انجام تمرینات افزایش دهیم. اگر قصد دارید تمامی اطلاعات و جزییات را درباره‌ی نرم‌افزار LabVIEW کسب کنید به شما توصیه می‌کنیم که این مطالب را در پنجره‌ی Online Reference یا از فهرست Manuals بر روی لوح فشرده مطالعه نموده، یا با شرکت NI ارتباط برقرار کنید. در ضمن می‌توانید اطلاعات مفیدی در مورد این برنامه از طریق اینترنت به‌دست آورید. در صورتی که با موضوعات و عناوین این فصل و زیرفصلهای آن آشنایی دارید، به راحتی می‌توانید به فصل بعدی مراجعه کنید.

در نرم‌افزار LabVIEW دو نوع منو وجود دارد: منوهای پایین کشیدنی و کرکره‌ای.

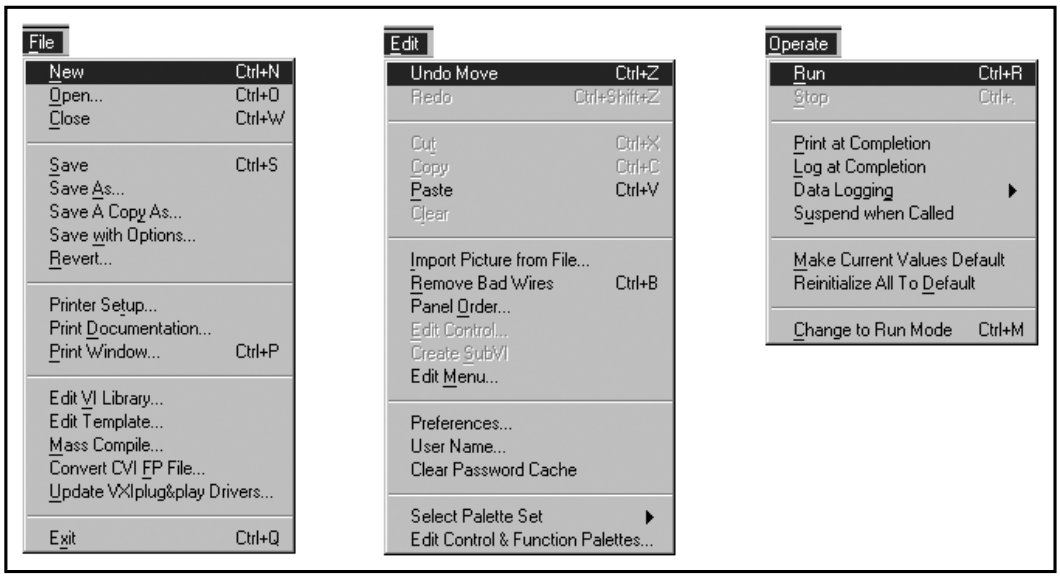
در تمرین قبل برخی از این منوها را ملاحظه نمودید. در تمرینات بعدی هر دو منو را استفاده خواهید کرد. در این بخش منوهای پایین کشیدنی را به اختصار مورد بررسی قرار می‌دهیم. اگر در خلال ارائه‌ی این توضیحات، منوهای مذکور را بر روی صفحه‌ی نمایشگر رایانه ملاحظه کنید، درک این منوها و گزینه‌های تعبیه شده در آنها بسیار ساده‌تر خواهد بود. در ضمن سعی کنید تا با منوهای مورد بحث تمرین کنید تا به نحوه‌ی به‌کارگیری و همچنین گزینه‌های موجود در آنها به خوبی آشنا شوید.

نوار حاوی منو یا به اصطلاح Menu Bar شامل چند منوی پایین کشیدنی است. هنگامی که بر روی یکی از عبارات موجود در نوار حاوی منو کلیک می‌کنید، یک منو در زیر نوار مذکور ظاهر می‌گردد. در اکثر منوهای پایین کشیدنی گزینه‌های عمومی نظیر Paste، Copy، Save، ...Open و دستورهای کاربردی دیگر وجود دارند. در این بخش در مورد برخی از دستورهای اصلی موجود در این منوها توضیح می‌دهیم و در مورد قابلیتهای پیشرفته‌ی آنها در فصول آینده صحبت خواهیم کرد.

در بسیاری از منوها و در کنار برخی از گزینه‌ها، نام بعضی از کلیدهای صفحه کلید نیز دیده می‌شود. اگر با محیط Windows در ارتباط هستید برای برقراری ارتباط از طریق صفحه کلید، کلید مربوط را به همراه کلید <Ctrl> فشار دهید. در صورتی که سیستم Mac در اختیار دارید، کلید مورد نظر را به همراه کلید <Cmd> فشار دهید.

در اغلب موارد کلیدهای مذکور در سمت راست گزینه‌های موجود در منوها دیده می‌شوند. به جای کلیک کردن بر روی منوها و انتخاب گزینه‌ی مورد نظر توسط ماوس، اسامی کلیدهای مذکور را به خاطر بسپارید و از آنها برای انتخاب گزینه‌ی مورد نظر استفاده کنید و بدین ترتیب سرعت عملکرد خود را افزایش دهید.

۳




**شکل ۲۱-۳: منوهای File، Edit و Operate**

## منوی File

با کلیک کردن بر روی عبارت File در نوار حاوی منوها، این منو را باز کنید. منوی File شامل دستورها و فرمانهای عمومی نظیر Save و Print می باشد. به کمک گزینه‌های New یا ...Open در این منو می توانید به ترتیب یک برنامه‌ی جدید ایجاد نموده، یا یکی از برنامه های موجود را باز کنید.

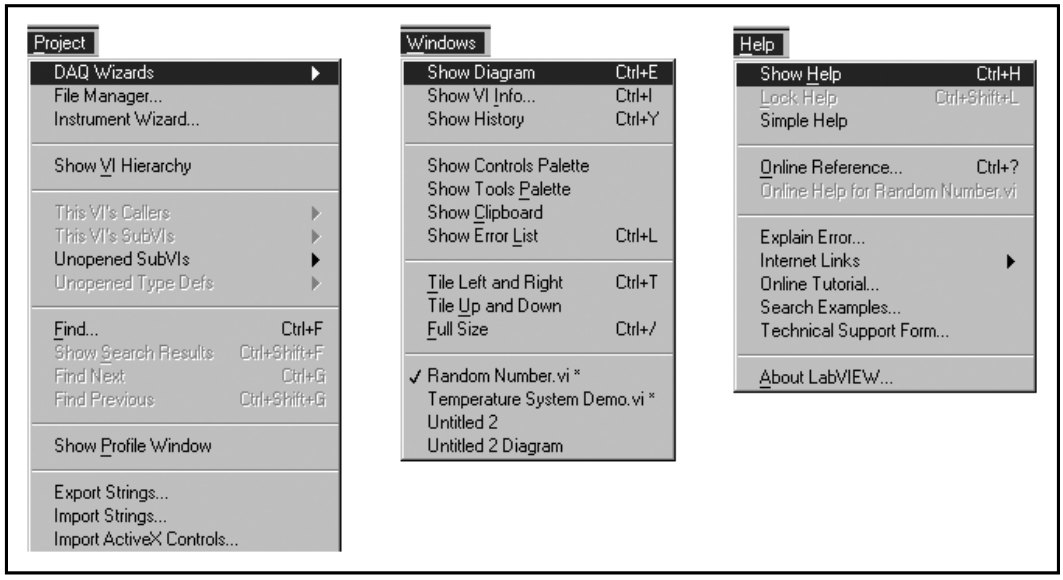
همان گونه که ذکر شد در رایانه های شخصی و تحت سیستم عامل Windows برای انتخاب گزینه‌ی File >> New، می توانید کلید <N> را به همراه کلید <Ctrl> فشار دهید.

## منوی Edit

منوی Edit را باز کنید. این منو شامل دستورهایی نظیر Paste، Copy، Cut و گزینه های دیگر است که به کاربر اجازه می دهند تا با استفاده از این گزینه ها صفحه‌ی برنامه‌ی خود را ویراستاری نموده، آن را اصلاح کند. همچنین می توانید با استفاده از گزینه‌ی Remove Bad Wires در این منو، سیم کشی نادرست یا در اصطلاح، bad wire را حذف نمایید.

## منوی Operate

اگرچه معمولاً برای صدور فرمانهای اجرا و توقف برنامه از کلیدها و دکمه های موجود در نوار ابزار استفاده می گردد، با استفاده از گزینه های موجود در این منو نیز می توانید این فرمانها را صادر کنید. به کمک این منو همچنین می توانید مقادیر پیش فرض برای یک برنامه را تغییر داده، نحوه‌ی چاپ و ذخیره‌ی داده ها را کنترل نمایید. به کمک گزینه های موجود در این منو نیز می توانید بین دو مد Run و Edit تغییر موقعیت دهید.




**شکل ۲۲-۳ : منوهای Project، Windows و Help**

## منوی Project

منوی Project شامل دستورهایی جهت استفاده در مجموعه های عظیم VI می باشد. منظور از مجموعه‌ی VI، برنامه ای با چندین مرحله زیربرنامه است. با استفاده از گزینه های موجود در این منو می توانید سلسله مراتب VIها را ملاحظه کنید. همچنین می توانید متن یا المان خاصی را در برنامه ها جستجو کنید.

## منوی Windows

منوی Windows را باز کنید. با استفاده از گزینه های موجود در این منو می توانید بین دو صفحه‌ی پانل و نمودار بلوکی تغییر موقعیت دهید. به کمک برخی از این گزینه ها نیز می توان لیست خطاهای احتمالی موجود در برنامه را مشاهده نمود.

همچنین با استفاده از گزینه های موجود در این منو می توانید دو پنجره را به گونه ای قرار دهید تا همزمان محتویات هر دو را ملاحظه نموده، یا بین صفحات مربوط به چندین VI که بر روی صفحه‌ی نمایشگر رایانه قرار دارند تغییر موقعیت دهید. در مجموع می توان گفت که به کمک گزینه های موجود در این منو می توانید اطلاعاتی در مورد برنامه یا زیربرنامه به دست آورید.

۳

## منوی Help

با استفاده از این منو می توانید محتویات پنجره‌ی Help را ملاحظه کنید. همچنین می توانید پنجره‌ی Help را مخفی کنید و یا آن را به طور ثابت بر روی صفحه‌ی نمایش نگه دارید. در ضمن به کمک این منو دسترسی به پنجره‌ی Online Reference در نرم افزار LabVIEW امکان پذیر است.

## پالت های شناور [12]

در نرم افزار LabVIEW چند پالت شناور وجود دارد که اغلب مورد استفاده قرار می گیرند. منظور از پالت شناور آن است که می توانید پالت مذکور را در نقطه‌ی دلخواهی قرار دهید به گونه ای که به راحتی قابل بررسی باشد. این پالت ها عبارتند از: Controls، Functions و Tools.

با کلیک کردن بر روی نوار عنوان یا Title Bar در هر پالت و حرکت دادن ماوس می توانید پالت مورد نظر را جابه‌جا کنید. روش بستن این پالت ها نیز مانند بستن پنجره های Windows می باشد. کافی است بر روی علامت «X» که در گوشه‌ی بالایی سمت راست پالت قرار دارد کلیک کنید. در صورتی که قصد داشته باشید مجدداً آنها را ظاهر سازید، گزینه‌ی Windows >> Show ... Palette را انتخاب کنید.

12- Floating Palettes

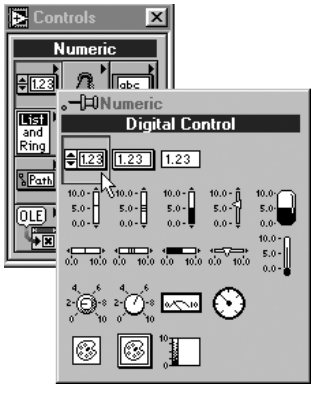
## پالت های Controls و Functions

در خلال برنامه نویسی در محیط LabVIEW، از پالت Controls مکرراً استفاده می کنید. زیرا تمامی المان های کنترل و نشان دهنده که درصفحه‌ی پانل نیاز دارید در این پالت قرار دارند. پالت Functions نیز شامل ساختارها و توابع خاص برای ایجاد صفحه‌ی نمودار بلوکی است.

پالت Controls تنها در صورتی قابل رؤیت است که صفحه‌ی پانل فعال باشد و به همین ترتیب در صورتی که صفحه‌ی نمودار بلوکی، فعال باشد می توان پالت Functions را مشاهده کرد. هریک از این پالت‌ها شامل چند زیرپالت است. این زیرپالت‌ها حاوی المان‌ها و عناصری هستند که برای ایجاد یک برنامه به آنها نیاز دارید. هنگامی که نشانگر ماوس بر روی دکمه‌های موجود درپالت‌های Controls یا Functions قرار می گیرد، نام زیرپالت مربوط در بالای پنجره ظاهر می گردد. همان‌گونه که در شکل ۲۳-۳ ملاحظه می کنید، هنگامی که نشانگر ماوس بر روی زیرپالت Boolean قرار می گیرد نام این زیرپالت که یکی از زیرپالت‌های موجود در پالت Controls است در بالای پنجره ظاهر می شود.

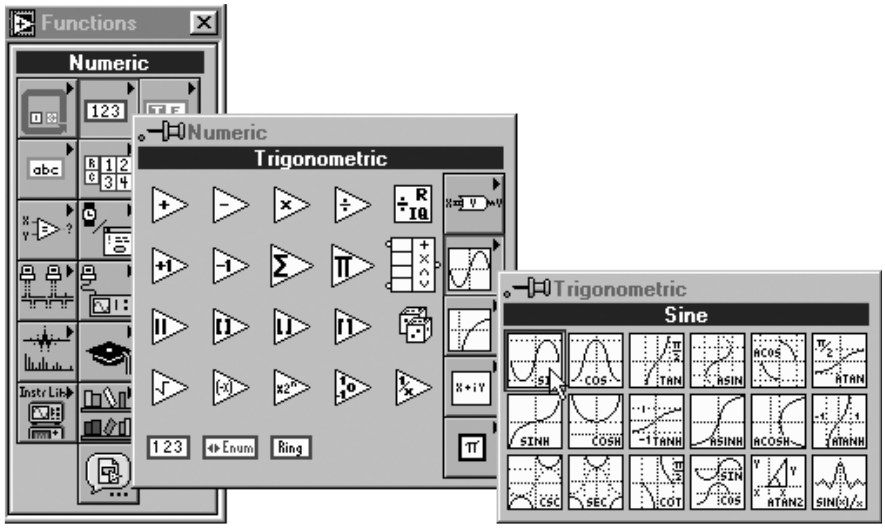
شکل ۲۳-۳

اگر بر روی دکمه‌ای که مربوط به یکی از زیرپالت‌هاست کلیک کرده، دکمه‌ی ماوس را در همان حالت نگه دارید، المان‌های موجود در آن بر روی صفحه ظاهر می‌شوند. برای انتخاب المانی که در یکی از زیرپالت‌ها قرار دارد، دکمه‌ی ماوس را بر روی آن المان رها کنید. سپس بر روی صفحه‌ی پانل یا نمودار بلوکی و در محلی که قصد دارید المان مذکور در آن‌جا قرار گیرد کلیک کنید. همان گونه که در مورد اسامی المان‌های زیرپالت ذکر شد، هنگامی که ماوس را مطابق شکل ۲۴-۳ بر روی المان‌های موجود در زیرپالت عبور می دهید، اسامی آنها در بالای پنجره به نمایش در می آید.




شکل ۲۴-۳

توجه داشته باشید که برخی از زیرپالت‌ها شامل زیرپالت‌های دیگری هستند که در آنها نیز المان‌های دیگری وجود دارد که در اصطلاح به آنها زیرپالت‌های مرکب یا سلسله مراتبی گویند. در گوشه‌ی بالایی سمت راست این زیرپالت‌ها یک پیکان قرار گرفته است و نسبت به المان‌های دیگر کمی بالاتر و برجسته‌تر به نظر می‌رسند. در فصل بعد در مورد برخی از زیرپالت‌های خاص و المان‌های موجود در آنها به بحث خواهیم نشست.




**شکل ۲۵-۳: شمای ظاهری یک زیرپالت مرکب**

۳

برای دستیابی به پالت‌های Controls یا Functions منوی کرکره‌ای را در ناحیه‌ای خالی از صفحات پانل یا نمودار بلوکی باز کنید. روش دیگری نیز برای باز کردن این منو با به‌کارگیری ابزار Pop-up Tool وجود دارد که در آینده در مورد آن توضیح خواهیم داد.

## پالت‌های سفارشی[۱۳]

در صورتی که نمای ظاهری و ساختار از پیش تعیین شده‌ی پالت‌های Functions و Controls نیازهای شما را برآورده نمی‌سازد، می‌توانید با توجه به نیاز خود این پالت‌ها را به دلخواه تغییر دهید و در اصطلاح آنها را سفارشی کنید.

برای انجام این عمل ابتدا گزینه‌ی ...Edit >> Edit Control and Function Palettes را انتخاب کنید. حال می‌توانید پالت‌های مورد نیاز خود را ایجاد نموده، با افزودن زیرپالت‌های جدید به برخی از پالت‌ها آنها را سفارشی کنید. همچنین می‌توانید برخی از المان‌های زیرپالت‌ها را پنهان سازید و یا المان‌های موجود در یک زیرپالت را به زیرپالت‌های دیگر انتقال دهید. به عنوان مثال اگر برای ایجاد یک برنامه به دستورها و توابع مثلثاتی نیاز دارید، می‌توانید برای دستیابی آسان‌تر به این دستورها، آنها را در زیرپالت Trigonometric قرار دهید.

13- Custom Palettes

در صورتی که برخی از دستورها و توابع مورد نیاز که به دفعات زیاد از آنها استفاده می کنید در زیرپالت های داخلی یا به عبارت دیگر در زیرپالت های مراحل پایین تر قرار دارند برای سهولت در دستیابی به دستورهای مذکور نیز می توانید آنها را در پالت ها و یا زیرپالت های طبقات بالاتر قرار دهید. بدین ترتیب برای دستیابی به دستورهای مذکور مجبور نیستید تا چندین طبقه از زیرپالت ها را جستجو کنید. همچنین در مورد دستورها و توابع غیر قابل استفاده که به ندرت به کارگرفته می شوند می توانید ترتیبی دهید تا دستورهای مذکور به زیرپالت های طبقات پایین تر انتقال یابند. در فصل ۴ در مورد نحوه‌ی سفارشی نمودن این پالت ها توضیحات بیشتری ارائه خواهیم داد.

## پالت Tools

ابزار یاToolدر حقیقت یک مد عملیاتی نشانگر ماوس می باشد. در صورت انتخاب یک ابزار، نشانگر ماوس به شکل دیگری در می آید و به کمک آن می توانید دستورهای عملیاتی و اصلاحی خاصی را به اجرا در آورید. این ابزارها به ترتیب نشان داده شده در شکل ۲۶-۳ در پالت Tools قرار گرفته اند.

**شکل ۲۶-۳: پالت Tools**

همان گونه که در مورد پالت های Controls و Functions اشاره شد، پالت Tools نیز شناور است. بدین مفهوم که می توانید آن را در هر نقطه‌ی دلخواه از صفحه‌ی نمایشگر رایانه قرار دهید و یا آن را ببندید. برای انتخاب یک ابزار، بر روی دکمه‌ی متناظر با آن در پالت Tools کلیک نمایید. در این حالت مشاهده می کنید که نشانگر ماوس به شکل ابزار انتخاب شده ظاهر می شود. اگر در مورد برخی از ابزارهای موجود در این پالت و کاربرد آنها اطلاعات کافی در دست ندارید نشانگر ماوس را بر روی ابزار مورد نظر قرار دهید. ملاحظه می کنید که توضیح مختصری در مورد ابزار انتخاب شده ارائه می گردد. حال کاربرد ابزارهای موجود در پالت Tools را مورد بررسی قرار می دهیم:

Operating Tool

با استفاده از ابزار Operating Tool می توانید مقادیر المان های کنترل و نشان دهنده را بر روی صفحه‌ی پانل تغییر دهید. همان گونه که از نام این ابزار استنباط می گردد، به کمک آن می توانید دکمه ها، کلیدها و دیگر المان ها را به کار بیندازید. این ابزار تنها ابزاری است که در هنگام اجرای برنامه در صفحه‌ی پانل فعال می باشد.

Positioning Tool

به کمک ابزار Positioning Tool می توان المان ها را انتخاب نمود و جابه‌جا کرد. همچنین تغییر اندازه‌ی برخی از المان ها با استفاده از این ابزار امکان پذیر است.

ابزار Labeling Tool جهت تعیین برچسبهای متنی مورد استفاده قرار می گیرد.

Labeling Tool

ابزار Wiring Tool جهت سیم کشی و اتصال المان های موجود در صفحه‌ی نمودار بلوکی به کار برده می شود. از این ابزار همچنین جهت انتساب المان های کنترل و نشان دهنده‌ی موجود بر روی صفحه‌ی پانل به ترمینال های کانکتور استفاده می گردد.

Wiring Tool

ابزار Color Tool به شما اجازه می دهد تا پیش زمینه و پس زمینه‌ی المان ها را رنگ آمیزی کنید. در صورتی که توسط این ابزار بر روی یک المان کلیک کنید، یک پالت شامل صدها رنگ بر روی صفحه ظاهر می گردد. سپس می توانید رنگ مورد نظر را انتخاب کنید.

Color Tool

هنگامی که ابزار Pop-up Tool را بر روی یک المان کلیک می کنید، منوی کرکره‌ای آن المان باز می شود.

Pop-up Tool

ابزار Scroll Tool به شما کمک می کند تا پنجره‌ی فعال را بلغزانید.

Scroll Tool

ابزار Breakpoint Tool به شما اجازه می دهد تا جهت سهولت در رفع اشکال برنامه چندین نقطه‌ی توقف یا breakpoint در برنامه ایجاد کنید. کلیک نمودن بر روی این ابزار سبب می شود تا اجرای برنامه به طور موقت متوقف گردد. در این حالت می توانید مقادیر ورودیهای برنامه را تغییر دهید و یا مراحل مختلف برنامه را بررسی کنید.

Breakpoint Tool

ابزار Probe Tool، پروپ هایی بر روی سیمهای ارتباط دهنده‌ی المان ها ایجاد می کند. به کمک این ابزار می توانید عبور و مقادیر داده ها در سیمهای رابط در یک برنامه‌ی در حال اجرا را بررسی کنید.

Probe Tool

با استفاده از ابزار Color Copy Tool می توانید رنگ یکی از المان‌های موجود را نمونه برداری کرده، سپس توسط آن رنگ، المان های دیگر را رنگ آمیزی کنید.

Color Copy Tool

برای دستیابی به ابزار مورد نظر در پالت Tools به جای کلیک کردن بر روی آن ابزار می‌توانید از کلید <Tab> برای حرکت در پالت مذکور استفاده نمایید. درضمن زمانی که صفحه‌ی پانل فعال است با فشار دادن کلید <Space bar> می‌توانید بین دو ابزار Positioning Tool و Operating Tool تغییر حالت دهید. همچنین در صورت فعال بودن صفحه‌ی نمودار بلوکی با فشار دادن کلید <Space bar> می‌توان بین دو

ابزار Positioning Tool و Wiring Tool تغییر حالت داد. دو کلید <Tab> و <Space bar> مراحل انتخاب و تعویض ابزارها را در مواردی که از این ابزارها مکرراً استفاده می‌کنید خلاصه و ساده‌تر می‌سازند. به‌کارگیری دو کلید مذکور موجب صرفه‌جویی در زمان می‌گردد. اگر باور ندارید امتحان کنید!

## نوار ابزار

نوار ابزار که در بالای صفحه‌ی برنامه قرار دارد شامل کلیدهایی جهت کنترل اجرای برنامه، انتخاب فونت‌های متنی و همچنین دستورهایی جهت کنترل، تنظیم فاصله و تراز نمودن المان‌هاست. توجه داشته باشید که تعداد کلیدها و گزینه‌های نوار ابزار در صفحه‌ی نمودار بلوکی بیشتر از گزینه‌های این نوار در صفحه‌ی پانل است. شایان ذکر است که تعدادی از گزینه‌ها و کلیدهایی که برای ایجاد تغییر و اصلاح در برنامه به کار می‌روند در هنگام اجرای برنامه ناپدید می‌شوند.

در صورتی که با عملکرد گزینه‌های این نوار آشنا نیستید و قصد دارید اطلاعات مختصری در مورد هر یک از گزینه‌ها و کلیدها به‌دست آورید کافی است نشانگر ماوس را بر روی گزینه‌ی مورد نظر قرار دهید. در این حالت توضیح مختصری را در مورد آن ملاحظه خواهید کرد. در شکل ۲۷-۳ نوار ابزار در صفحه‌ی نمودار بلوکی نشان داده شده است.

**شکل ۲۷-۳: نوار ابزار در صفحه‌ی نمودار بلوکی**

Run Button

شکل ظاهری دکمه Run شبیه به یک پیکان است. هنگامی که بر روی آن کلیک می‌کنید اجرای برنامه آغاز می‌گردد.

Run Button (active)

در زمانی که برنامه در حال اجرا باشد شکل ظاهری این دکمه تغییر می‌کند.

Run Button (broken)

در صورتی که برنامه کامپایل نگردد، این کلید به صورت یک «پیکان شکسته» نشان داده می‌شود و بیانگر آن است که این برنامه قابل اجرا نیست.

Continuous Run Button

فشار دادن این دکمه باعث می‌شود که برنامه‌ی مورد نظر به طور پیوسته اجرا شود. اجرای برنامه تا زمانی که دکمه‌ی Stop یا Abort را فشار نداده‌اید ادامه می‌یابد. عملکرد این کلید شبیه به دستور GO TO در زبانهای برنامه نویسی است.

Abort Button

دکمه‌ی Abort یا Stop به راحتی قابل تشخیص است. زیرا شکل ظاهری این دکمه به علامت «توقف» شبیه است. در هنگامی که اجرای یک برنامه آغاز می‌گردد این کلید نیز فعال می‌شود. در غیر این صورت به رنگ خاکستری ظاهر می‌شود و نشانگر غیرفعال بودن این دکمه است. برای متوقف ساختن اجرای برنامه کافی است بر روی این دکمه کلیک کنید.

کلیک کردن بر روی دکمه‌ی Abort نظیر فشار دادن کلید <Break> بر روی صفحه کلید است. به محض فشار دادن این دکمه، اجرای برنامه بلافاصله متوقف می‌گردد. متوقف کردن برنامه به این روش باعث از دست دادن بخشی از داده‌ها می‌شود. در بخشهای بعدی نشان خواهیم داد که چگونه می‌توان برنامه را با به‌کارگیری روشهای بهتری متوقف ساخت.

Pause Button

کلیک کردن بر روی این دکمه باعث می‌شود که در اجرای برنامه وقفه‌ای ایجاد گردد. بدین ترتیب می‌توانید گزینه‌های تک‌مرحله‌ای Step Into، Step Over و Step Out را به‌کار برید. با فشار دادن مجدد این کلید، اجرای برنامه ادامه می‌یابد.

Step Into Button

Step Out Button

Step Over Button

دکمه‌های تک مرحله‌ای Step Into، Step Over و Step Out سبب می‌شوند تا در هر لحظه تنها یک مرحله از برنامه به اجرا در آید. با استفاده از این دکمه‌ها می‌توانید به رفع اشکالات احتمالی موجود در برنامه‌ی خود بپردازید و یا اجرای برنامه را مرحله به مرحله مشاهده کنید. در فصل ۵ در مورد استفاده از این دکمه‌ها به تفصیل سخن خواهیم گفت.

Execution Highlighting Button

کلیک کردن بر روی این دکمه باعث می‌شود تا مسیر عبور داده‌ها در هنگام عبور در صفحه‌ی نمودار بلوکی از سایر مسیرها متمایز گردد. در این حالت می‌توانید مشاهده کنید که داده‌ها در چه مرحله‌ای از صفحه‌ی نمودار بلوکی قرار دارند. در صورتی که این کلید را فشار دهید به نظر می‌رسد که چراغ نشان داده شده در شکل روشن می‌شود. در این حالت قادر خواهید بود مقادیر داده‌ها را در صفحه‌ی نمودار بلوکی ملاحظه کنید.

Warning Button

این کلید در هنگامی ظاهر می‌شود که برنامه را به گونه‌ای پیکربندی کنید که در صورت وجود خطاهای ناخواسته و غیرعمدی در حین نوشتن برنامه به شما هشدار داده شود. با کلیک نمودن بر روی این دکمه می‌توانید لیست هشدارهای موجود را ملاحظه کنید. هشدار به منزله‌ی خطا نیست و تنها به شما اطلاع می‌دهد عملی را انجام داده‌اید که احتمالاً قصد انجام آن را نداشته‌اید. به عنوان مثال در صفحه‌ی پانل، یک المان کنترل وجود دارد که فراموش کرده‌اید آن را در صفحه‌ی نمودار بلوکی به المان‌های دیگر سیم‌کشی کنید.

با استفاده از منوی حلقوی[۱۴] Font که در نوار ابزار قرار دارد می‌توانید نوع، اندازه، شکل و رنگ فونت استفاده شده در محیط LabVIEW را تغییر دهید. زیرا در برخی موارد لازم است تا برای المان‌ها برچسب تهیه نموده و از فونت‌های مختلف استفاده کنید.

**شکل ۲۸-۳ : منوی حلقوی Font**

14- Menu Ring

منوی حلقوی Alignment

منوی حلقوی Distribution

در نرم‌افزار LabVIEW روشی خودکار جهت تنظیم کردن فاصله‌ی المان‌ها وجود دارد که به شما کمک می‌کند تا المان‌های موجود در صفحه را مرتب و تراز نمایید و حتی فاصله‌ی آنها را به طور یکنواخت تنظیم کنید. برای انجام این عمل، به کمک ابزار Positioning Tool در اطراف المان‌هایی که قصد مرتب کردن آنها را دارید یک چهارچوب کشیده، سپس یکی از موارد Left Edges، Top Edges و Vertical Centers را بسته به نیاز خود از منوی حلقوی Alignment انتخاب کنید. در صورتی که قصد دارید تا فضای یکنواختی بین المان‌های موجود در صفحه ایجاد گردد، به همین روش از منوی حلقوی Distribution استفاده نمایید.

منوی حلقوی Reorder

با استفاده از منوی حلقوی Reorder می‌توانید المان‌ها را در جلو، پشت، بالا یا پایین المان‌های دیگر قرار دهید. انجام این عمل در مواردی مفید است که چند المان، همدیگر را می‌پوشانند و شما قصد دارید نحوه‌ی به نمایش درآمدن آنها را بر روی صفحه تغییر دهید.

## مدهای Run و Edit

زمانی که یک برنامه را باز می‌کنید، این برنامه در محیط Edit بارگذاری می‌شود. در این حالت می‌توانید برنامه را به دلخواه تغییر دهید و یا آن را اصلاح کنید. هنگامی که برنامه را اجرا می‌کنید، به صورت خودکار به محیط Run تغییر حالت می‌دهید. در این حالت ایجاد تغییر در برنامه مجاز نیست. در زمانی که برنامه در حال اجراست، فقط ابزار Operating Tool در صفحه‌ی پانل فعال است. هنگامی که اجرای برنامه به پایان برسد، مجدداً به محیط Edit باز می‌گردید. مگر آن که به صورت دستی آن را در محیط Run برده باشید. جهت انتخاب محیط Run بر روی گزینه‌ی Operate >> Change to Run Mode کلیک کنید. برای تغییر محیط به مد Edit گزینه‌ی Change to Edit Mode را از همان منو انتخاب نمایید. در صورتی که برنامه‌ای در محیط Run به اجرا در آمده باشد، بدین مفهوم است که این برنامه با موفقیت کامپایل شده و عاری از اشکال است.

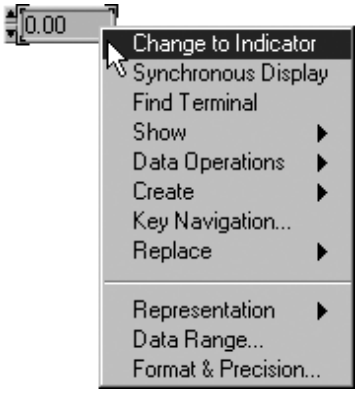
در صورتی که ترجیح می‌دهید برنامه‌ی خود را در محیط Run باز کنید، گزینه‌ی ...Edit >> Preferences را انتخاب نمایید. سپس گزینه‌ی Miscellaneous را از منوی حلقوی موجود در پنجره‌ی محاوره‌ای انتخاب نموده، زیرگزینه‌ی Open VIs in run mode را فعال کنید.

## منوهای کرکره‌ای

در این بخش قصد داریم تا در مورد نوع دیگری از منوهای نرم‌افزار LabVIEW یعنی منوهای کرکره‌ای به بحث بپردازیم. در بسته‌ی نرم‌افزاری LabVIEW مکرراً از این منوها استفاده خواهید کرد. برای باز کردن این منو، نشانگر ماوس را بر روی المانی که قصد دارید منوی کرکره‌ای آن را باز کنید قرار داده، سپس دکمه‌ی سمت راست ماوس را در محیط Windows فشار دهید. در محیط Mac کلید <Cmd> را

پایین نگه داشته، سپس کلیک کنید.

برای باز کردن این منو همچنین می توانید از ابزار Pop-up Tool کمک بگیرید. برای انجام این عمل ابتدا ابزار مذکور را از پالت Tools انتخاب نموده، سپس بر روی المان مورد نظر کلیک کنید تا منوی کرکره‌ای ظاهر گردد. در شکل ۲۹-۳ گزینه های موجود در منوی کرکره‌ای المان Digital Control را ملاحظه می کنید.




**شکل ۲۹-۳: گزینه‌های موجود در منوی کرکره‌ای المان Digital Control**

هر یک از المان‌های LabVIEW دارای یک منوی کرکره‌ای می‌باشد که شامل گزینه‌ها و فرمانهای خاص همان المان است. گزینه‌های موجود در این منو به نوع المان بستگی دارد. به عنوان مثال منوی کرکره‌ای مربوط به دو المان Numeric Control و Waveform Chart با یکدیگر متفاوت‌اند. علاوه بر این، گزینه‌های موجود در این منو در دو محیط Run و Edit نیز تفاوت دارند.

در صورتی که در فضایی خالی از صفحات پانل و نمودار بلوکی منوی کرکره‌ای را باز کنید، به ترتیب به یکی از پالت های Controls و Functions دست می یابید. بنابراین به کمک منوی کرکره‌ای و به روش مذکور می توان به دو پالت Controls و Functions دست یافت و دیگر لازم نیست تا برای باز کردن دو پالت مذکور، گزینه‌ی مربوط به آنها را از منوی Windows انتخاب کنید.

در این کتاب در مورد انتخاب برخی از فرمانها و دستورها، استفاده نمودن از منوی کرکره‌ای الزامی است. همان گونه که قبلاً اشاره شد از این منو مکرراً استفاده خواهید کرد. بنابراین سعی کنید تا نحوه‌ی باز کردن این منو را به خوبی فرا گیرید.

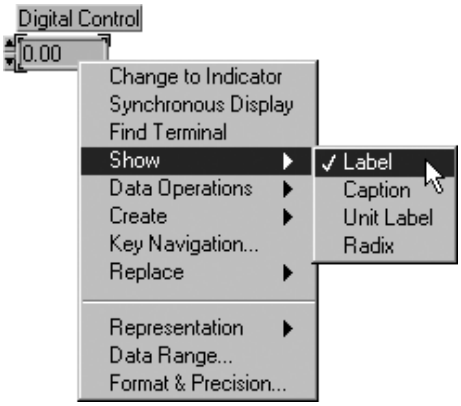
در صورتی که ابزار Color Tool فعال باشد، هنگامی که قصد باز کردن منوی کرکره‌ای را دارید به جای ظاهر شدن این منو، یک پالت رنگ ظاهر می‌شود.

## نکاتی در مورد منوی کرکره‌ای

بسیاری از گزینه‌های موجود در منوهای کرکره‌ای خود شامل زیرمنوهای دیگری هستند که به آنها منوهای ترتیبی[۱۵] گویند. این گزینه‌ها با یک پیکان که در سمت راست آنها قرار گرفته است متمایز می‌شوند.

۱۵- در برخی موارد به این منوها، منوهای چندطبقه‌ای یا سلسله مراتبی نیز گویند.

منوهای ترتیبی در برخی موارد چند گزینه‌ی انتخابی دارند. گزینه‌ی انتخاب شده در این منوها توسط علامت «✓» که درکنار آن قرار گرفته است مشخص می‌شود. در گزینه‌های تصویری، گزینه‌ی انتخاب شده توسط یک چهارچوب احاطه می‌شود.




**شکل ۳۰-۳: نحوه‌ی انتخاب یکی از گزینه‌های موجود در منوی ترتیبی**

برخی از گزینه‌های موجود در منوهای کرکره‌ای دارای پنجره‌های محاوره‌ای هستند. این پنجره‌ها نیز شامل گزینه‌هایی می‌باشند که توسط آنها می‌توانید ترکیب یا ساختار المان مورد نظر را تعیین کنید. گزینه‌هایی که حاوی پنجره‌ی محاوره‌ای هستند، با علامت «...» که درکنار آنها قرار دارد متمایز می‌شوند.

گزینه‌های دیگر که فاقد یکی از دو علامت «✓» یا «...» می‌باشند، معمولاً دستورهایی هستند که پس از انتخاب، بلافاصله اجرا می‌شوند. این فرمان‌ها غالباً نظیر دستور Change to Indicator با یک فعل[16] آغاز می‌شوند. پس از انتخاب این فرمانها ممکن است برخی از دستورهای موجود در منو، به گزینه‌ها و فرمانهای معکوس خود تغییر وضعیت دهند. به عنوان مثال پس از انتخاب گزینه‌ی Change to Indicator، همین گزینه به دستور Change to Control تغییر می‌یابد.

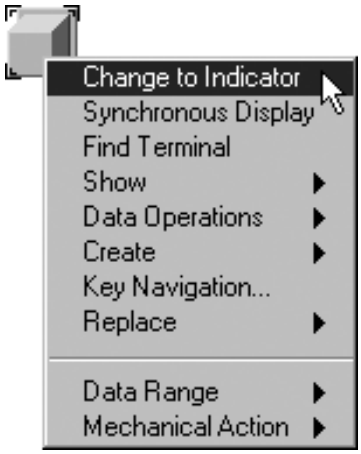
در برخی موارد، بخشهای مختلف یک المان، منوهای کرکره‌ای گوناگونی دارند. برای مثال اگر منوی کرکره‌ای مربوط به برچسب یا Label را در مورد یک المان باز کنید، این منو تنها شامل گزینه‌ی Size to Text می‌باشد. حال آن که پس از باز نمودن منوی کرکره‌ای بر روی المان یک منوی کامل حاوی چند گزینه ظاهر می‌گردد. بنابراین اگر در مواردی اقدام به باز نمودن منوی کرکره‌ای کردید و منوی مورد نظر خود را نیافتید، منوی کرکره‌ای را در مکانهای دیگر بر روی المان باز کنید.

## قابلیتهای منوی کرکره‌ای

به کمک منوی کرکره‌ای می‌توان چندین مشخصه از المان را تعیین نمود. در بسیاری از منوها گزینه‌های خاصی وجود دارند. در ادامه به ارائه‌ی توضیح در مورد آنها می‌پردازیم. به نظر ما توضیح این موارد برای شما

16- Verb

کافی است. در شکل ۳۱-۳ منوی کرکره‌ای المان Square Push Button را ملاحظه می کنید.




**شکل ۳۱-۳: منوی کرکره‌ای المان جبری Square Push Button**

## گزینه‌های Change to Control و Change to Indicator

با انتخاب گزینه‌ی Change to Control می توانید یک المان نشان‌دهنده را به یک المان کنترل تبدیل کنید. در صورت انتخاب گزینه‌ی Change to Indicator فرآیند مذکور به صورت معکوس انجام می‌گیرد. یعنی المان کنترل به نشان‌دهنده تبدیل می شود. منوی کرکره‌ای یک المان کنترل شامل گزینه‌ی Change to Indicator است. در حالی که این گزینه در منوی کرکره‌ای یک المان نشان‌دهنده به صورت Change to Control ظاهر می گردد.

به دلیل اینکه گزینه‌های Change to Control/Indicator اولین گزینه‌های موجود در منوی کرکره‌ای هستند ممکن است با قرار دادن ماوس بر روی این گزینه‌ها به طور تصادفی و ناخواسته آنها را انتخاب کنید. همان گونه که می‌دانید در صفحه‌ی نمودار بلوکی، عملاً کنترل‌ها و نشان‌دهنده‌ها غیر قابل تعویض هستند. بدین مفهوم که نمی‌توان هر یک را به جای دیگری مورد استفاده قرار داد. به خاطر داشته باشید که در صفحه‌ی نمودار بلوکی حاشیه‌ی المان کنترل نسبت به حاشیه‌ی المان نشان‌دهنده پهن‌تر است.

## گزینه‌های Find Control/Indicator و Find Terminal

در صورتی که گزینه‌ی Find Terminal را از منوی کرکره‌ای بر روی یک المان در صفحه‌ی پانل انتخاب کنید، نرم‌افزار LabVIEW ترمینال متناظر با آن را در صفحه‌ی نمودار بلوکی نشان می‌دهد و آن را متمایز می کند. در صورت انتخاب گزینه‌ی Find Control/Indicator از منوی کرکره‌ای بر روی یک ترمینال درصفحه‌ی نمودار بلوکی، المان متناظر با آن در صفحه‌ی پانل متمایز می گردد.

### گزینه‌ی Show

گزینه‌ی Show در منوی کرکره‌ای بسیاری از المان‌ها وجود دارد. به کمک این دستور می‌توان برخی از بخشهای مربوط به المان نظیر Label، Scrollbar، Unit و Terminals را به نمایش در آورد و یا آنها را مخفی ساخت.

در صورت انتخاب این گزینه، زیرمنوی دیگری ظاهر می‌شود که در آن لیستی از مواردی را که می‌توان به نمایش در آورد مشاهده می‌کنید. زیرگزینه‌های موجود در این لیست به نوع المان بستگی دارند. اگر در کنار گزینه‌های این زیرمنو علامت «✓» وجود داشته باشد، بدین معنی است که در حال حاضر این گزینه قابل رؤیت است. فقدان علامت «✓» بدین مفهوم است که این گزینه مخفی شده است. برای تغییر دادن وضعیت یک گزینه یا به عبارت دیگر برای حذف یا نمایش علامت «✓» در کنار یک گزینه، کافی است بر روی آن کلیک کنید.

### گزینه‌ی Data Operations

در گزینه‌ی Data Operations چند زیرگزینه‌ی کاربردی و مفید وجود دارد. این زیرگزینه‌ها به شما کمک می‌کنند تا داده‌ها و اطلاعات را در المان کنترل یا نشان‌دهنده، تحت نفوذ خود در آورید. توجه داشته باشید که این گزینه‌ها تنها گزینه‌های فعال و در دسترس منوی کرکره‌ای در محیط Run می‌باشند. در ادامه به بررسی زیرگزینه‌های موجود در این گزینه می‌پردازیم:



- گزینه‌ی Reinitialize to Default مقادیر پیش‌فرض را به المان مورد نظر نسبت می‌دهد.
- گزینه‌ی Make Current Value Default مقادیر فعلی را با داده‌های پیش‌فرض جایگزین می‌کند.
- به کمک گزینه‌های Paste Data، Copy Data و Cut Data می‌توان داده‌ها را از یک المان کنترل یا نشان‌دهنده کپی کرد و یا داده‌ها را به آنها اعمال نمود.
- با انتخاب گزینه‌ی Description یک پنجره‌ی محاوره‌ای بر روی صفحه ظاهر می‌شود که به کمک آن می‌توانید توضیحاتی که در مورد برخی از المان‌ها وجود دارد مطالعه کرده، یا این توضیحات را وارد کنید. در محیط Run، تنها می‌توانید توضیحات موجود را مشاهده کنید و قادر به ایجاد تغییر در آنها نخواهید بود.

### گزینه‌ی Show or Hide Control/Indicator

به کمک این گزینه می‌توانید المان کنترل یا نشان‌دهنده را که متناظر با ترمینال موجود در صفحه‌ی نمودار بلوکی است در صفحه‌ی پانل به نمایش در آورید یا آن را مخفی کنید. این عمل در مواردی سودمند و کاربردی است که کاربر به ترمینال‌های متناظر با المان‌ها در صفحه‌ی نمودار بلوکی نیاز دارد ولی قصد ندارد آن المان‌ها را در صفحه‌ی پانل ظاهر سازد.

### گزینه‌ی Create

این گزینه روشی ساده جهت ایجاد یک متغیر محلی و گره‌ی خصوصیت را برای یک المان فراهم می کند. این عناوین در فصل ۱۲ به تفصیل مورد بررسی قرار خواهند گرفت.

### گزینه‌ی ... Key Navigation

با استفاده از گزینه‌ی ... Key Navigation می توان مجموعه ای از کلیدهای صفحه کلید را در مورد المان های موجود در صفحه‌ی پانل به‌کار برد. هنگامی که کاربر کلیدهای صفحه کلید را در مورد یک برنامه‌ی در حال اجرا فشار می دهد، درست شبیه به حالتی عمل می کند که وی به کمک ماوس بر روی المان مذکور کلیک کرده باشد. بنابراین به کمک این گزینه می توان کلیک کردن ماوس بر روی المان ها را با استفاده از کلیدهای صفحه کلید انجام داد.

### گزینه‌ی Replace

این گزینه به شما اجازه می دهد تا به پالت های Controls و Functions دست یابید. البته دسترسی شما به پالت های مذکور به فعال بودن صفحات پانل یا نمودار بلوکی بستگی دارد. این گزینه همچنین به شما اجازه می دهد تا المانی را که منوی کرکره‌ای آن باز شده است با المان دیگر جایگزین کنید. در صورت تعویض یک المان سیمهای متصل به آن سالم و بدون تغییر باقی می مانند.

اگر نتوانسته اید تمامی این گزینه ها را به خاطر بسپارید جای نگرانی نیست. زیرا در برنامه هایی که در آینده خواهید نوشت به تدریج با این گزینه ها و قابلیتهای دیگر آنها آشنا خواهید شد.

همان گونه که قبلاً اشاره شد گزینه‌های موجود در منوی کرکره‌ای در دو محیط Run و Edit متفاوت‌اند. اگر موفق به یافتن گزینه‌ای خاص در منوی کرکره‌ای نشدید، به محیط دیگر تغییر وضعیت دهید و یا اینکه منوی کرکره‌ای را در محل دیگری بر روی المان باز کنید.

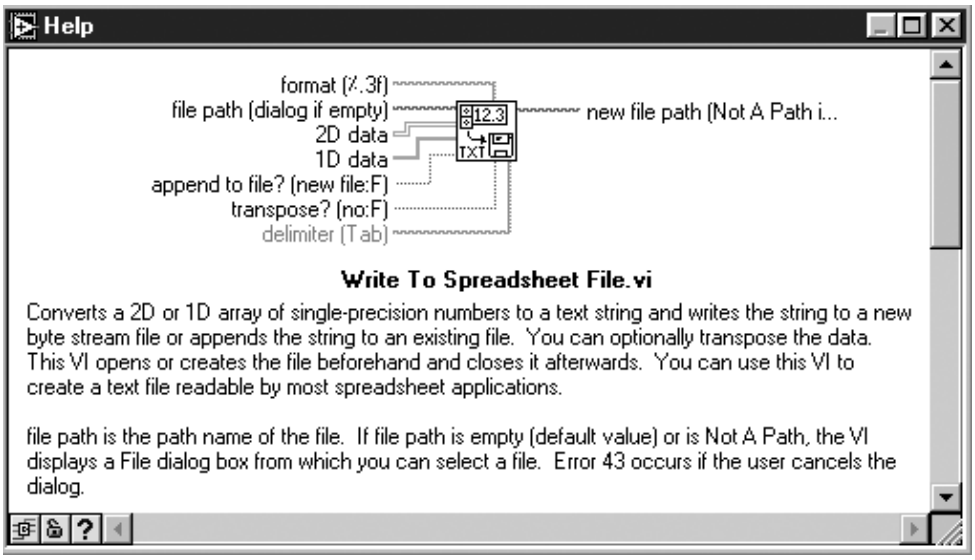
### پنجره‌ی Help

پنجره‌ی Help در نرم‌افزار LabVIEW اطلاعات مفیدی در رابطه با دستورها، اعداد ثابت، زیربرنامه ها و المان های کنترل و نشان‌دهنده در اختیار کاربر قرار می دهد. برای مشاهده‌ی این پنجره گزینه‌ی Help >> Show Help را انتخاب نمایید.

برای انتخاب این پنجره به کمک صفحه کلید می توانید در دو محیط Windows و Mac، به ترتیب از ترکیب کلیدهای <Ctrl-H> و <Cmd-H> استفاده کنید. اگر کلید <Help> بر روی صفحه کلید رایانه‌ی شما وجود دارد، برای باز کردن پنجره‌ی مذکور، مستقیماً این کلید را فشار دهید. در ضمن می توانید اندازه‌ی این پنجره را تغییر داده،

یا آن را در محل دلخواه قرار دهید.

در شکل ۳۲-۳ پنجره‌ی Help را در مورد برنامه‌ی Write To Spreadsheet File.vi ملاحظه می کنید.




**شکل ۳۲-۳ : پنجره‌ی Help در مورد برنامه‌ی Write To Spreadsheet File.vi**

هنگامی که نشانگر ماوس بر روی یک دستور، زیربرنامه و یا آیکن برنامه قرار گیرد، آیکن مربوط به آن دستور، برنامه یا زیربرنامه به همراه سیمها، نقاط اتصال و نوع داده های مربوط به هر ترمینال، در پنجره‌ی Help به نمایش در می آیند. توجه داشته باشید که سیمهای ورودی به سمت چپ و سیمهای خروجی به سمت راست آیکن اتصال می یابند. نام هر ترمینال نیز در کنار هر یک از سیمها دیده می شود. اگر در مورد برنامه‌ی مربوط توضیح خاصی وجود داشته باشد، این توضیحات نیز به نمایش در می آیند.

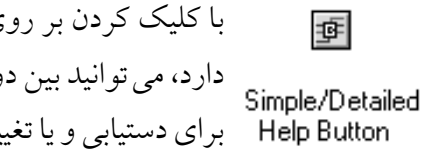
Lock Button

در صورت انتخاب گزینه‌ی Lock Help << Help یا با کلیک کردن بر روی دکمه‌ی Lock در پنجره‌ی Help می توان محتویات و اطلاعات موجود در پنجره‌ی Help را ثابت نگه داشت به طوری که با حرکت ماوس و قرار گرفتن آن بر روی المان های دیگر اطلاعات دیگری در پنجره‌ی Help نمایش داده نشود.

اگر ابزار Wiring Tool را بر روی گره‌ای از یک دستور یا زیربرنامه قرار دهید، پنجره‌ی Help در مورد گره‌ی متناظر به نمایش در می آید. این عمل در هنگام انجام سیم کشی و برقراری ارتباط بین المان ها بسیار مفید است؛ زیرا در این حالت می توانید صحت انجام عملیات سیم کشی را بررسی نموده، از صحت برقراری ارتباط المان مورد نظر با المان های مجاور اطمینان حاصل کنید. برخی مواقع ممکن است به دلیل گستردگی حجم اطلاعات موجود در پنجره‌ی Help ناگزیر به استفاده از نوار پیمایش باشید تا بدین ترتیب بتوانید تمامی جزییات را مطالعه کنید.

در مورد VIها و دستورهایی که تعداد ورودیها و خروجیهای آنها زیاد است، می توانید محتویات پنجره‌ی Help

را به دو صورت مختصر یا مفصل مورد بررسی قرار دهید. با انتخاب حالت Simple Help تنها اتصالات مهم و اساسی مورد بررسی قرار می‌گیرند و نشان داده می‌شوند. برای مطالعه‌ی دقیق و بررسی جزییات، حالت Help Detailed را انتخاب کنید.


Simple/Detailed Help Button

با کلیک کردن بر روی این دکمه که در گوشه‌ی پایینی سمت چپ پنجره‌ی Help قرار دارد، می‌توانید بین دو نوع بررسی مختصر و مفصل تغییر وضعیت دهید. روش دیگری برای دستیابی و یا تغییر موقعیت بین دو حالت مذکور وجود دارد. برای مشاهده‌ی این دو حالت گزینه‌ی Help >> Simple Help را فعال یا غیر فعال نمایید.

در بررسی مختصر، اتصالات الزامی[17] با فونت پهن[18] و اتصالات پیشنهادی[19] با فونت ساده[20] به نمایش در می‌آیند. در این حالت، اتصالات اختیاری[21] نمایش داده نمی‌شوند. تنها سرسیمهای مربوط به ورودیها و خروجیهایی که اتصالات آنها به نمایش در نیامده‌اند، در پنجره‌ی Help نشان داده می‌شود. وجود این سرسیمها جهت آگاه کردن شما از وجود اتصالات دیگری است که برای این المان، دستور یا زیربرنامه در نظر گرفته شده است و می‌توانید آنها را در حالت بررسی مفصل مشاهده کنید.

در حالت بررسی مفصل، اتصالات الزامی با فونت پهن، اتصالات پیشنهادی با فونت ساده و اتصالات اختیاری به رنگ خاکستری نشان داده می‌شوند.

اگر در مورد ورودی یک دستور نیاز به سیم کشی نباشد، در اغلب موارد مقدار پیش‌فرض در پرانتز و بعد از نام ورودی ظاهر می‌شود. در صورتی که دستوری بتواند چند نوع داده را قبول کند، معمول‌ترین نوع داده در پنجره‌ی Help به نمایش در می‌آید.

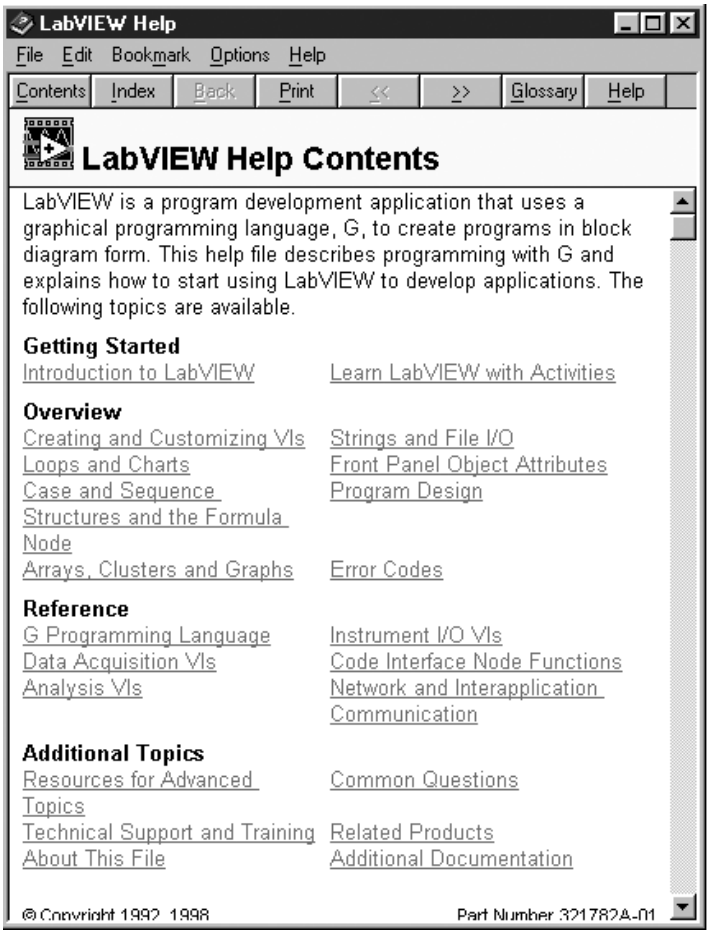
## پنجره‌ی Online Reference

پنجره‌ی Online Reference به کاربر اجازه می‌دهد تا در مورد دستورها، برنامه‌ها و المان‌های کنترل و نشان‌دهنده به یک منبع اطلاعاتی ارزشمند دست یابد. برای دستیابی به این اطلاعات، گزینه‌ی Help >> Online References... را انتخاب نمایید. روش دیگر جهت دستیابی به این اطلاعات، کلیک کردن بر روی دکمه‌ی Online Help در پنجره‌ی Help می‌باشد. در صورت انتخاب این گزینه پنجره‌ای مطابق شکل ۳۳-۳ ظاهر می‌شود.

برای دستیابی به اطلاعات مورد نظر در پنجره‌ی Online Reference، می‌توانید موارد دلخواه را از لیست عناوین موجود بیابید و یا آنها را جستجو کنید. در فصل ۱۵ خواهید دید که چگونه می‌توانید برای برنامه‌هایی که خودتان نوشته‌اید توضیحاتی به صورت Online Help اضافه نمایید.

17- Required
18- Bold
19- Recommended
20- Plain
21- Optional

در حال حاضر برای تمامی VIها، پنجره‌ی Online Help در نظر گرفته شده است. در صورتی که برای یک VI، پنجره‌ی Online Help تعریف نشده باشد، گزینه‌های موجود در منوی Online Help و همچنین کلید Help به رنگ خاکستری ظاهر می‌شوند که به مفهوم غیرفعال بودن آنهاست.




**شکل ۳۳-۳: پنجره‌ی Online Reference در محیط LabVIEW**

## مطالب مختصری در مورد زیربرنامه‌ها

اگر قصد دارید از تمام قابلیتها و توانمندی های نرم افزار LabVIEW بهره ببرید باید طبیعت و ماهیت سلسله مراتبی برنامه ها را درک و از آن استفاده کنید. یک زیربرنامه به بیان ساده، یک برنامه‌ی مستقل است که می تواند در برنامه های دیگر نیز مورد استفاده قرار گیرد. پس از ایجاد یک برنامه، می توانید با انتساب یک آیکن به آن و همچنین تعریف اتصالات و کانکتور، از آن به عنوان یک زیربرنامه در برنامه های سطوح بالاتر استفاده کنید.

همان گونه که قبلاً اشاره شد، زیربرنامه در نرم‌افزار LabVIEW متناظر و مشابه با Subroutine در زبانهای برنامه نویسی متنی متداول نظیر C است. همان‌طور که در تعداد Subroutine های استفاده شده در زبان برنامه نویسی C هیچ محدودیتی وجود ندارد، در مورد تعداد زیربرنامه ها در نرم‌افزار LabVIEW نیز محدودیتی دیده نمی شود[۲۲].

در صورتی که در صفحه‌ی نمودار بلوکی برنامه آیکن های زیادی وجود داشته باشد، می توانید برای ساده‌تر کردن صفحه‌ی مذکور، آنها را در یک یا چند زیربرنامه دسته‌بندی نمایید. همچنین می توانید از یک زیربرنامه جهت انجام یک دستور مشترک در برنامه های چند مرحله‌ی بالاتر استفاده کنید.

خاصیت مدولار بودن در LabVIEW موجب سهولت فراگیری، تغییر، اصلاح و رفع اشکال برنامه‌ها می گردد. در مورد نحوه‌ی ایجاد زیربرنامه در آینده توضیح خواهیم داد.

## تمرین ۲-۳: بررسی صفحات پانل و نمودار بلوکی

در این بخش، برای درک محیط برنامه نویسی LabVIEW به بررسی چند تمرین ساده می پردازیم. سعی کنید تا اعمال زیر را به درستی انجام دهید. اگر در حین اجرای دستورها با مشکل مواجه شدید، برای کسب اطلاعات و راهنمایی، به مطالب ذکر شده در همین فصل بازگردید.

۱- با انتخاب گزینه‌ی New << File یک برنامه‌ی جدید باز نموده، سپس بین دو صفحه‌ی پانل و نمودار بلوکی تغییر وضعیت دهید.

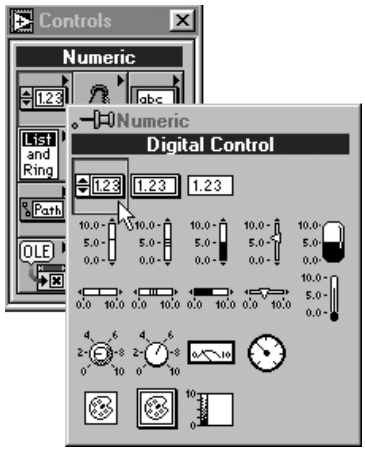
برای تمرین بیشتر چند مرتبه دو صفحه‌ی پانل و نمودار بلوکی را به کمک ماوس و صفحه کلید باز کنید.

۲- اندازه‌ی پنجره ها را طوری تنظیم کنید تا همزمان هر دو صفحه قابل رؤیت باشند. زیرا ممکن است به هر دو نیاز داشته باشید.

برای مشاهده‌ی هر دو پنجره، از گزینه‌های Tile Left and Right یا Tile Up and Down در منوی Windows کمک بگیرید.

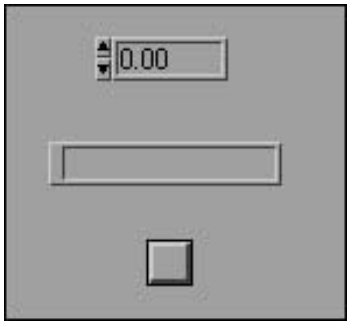
۳- المان‌های Digital Control، String Control و Square Button را از پالت Controls انتخاب نموده، آنها را بر روی صفحه‌ی پانل قرار دهید. برای دستیابی به المان Digital Control مطابق شکل ۳۴-۳ بر روی دکمه‌ی مربوط به زیرپالت Numeric در پالت Controls کلیک نموده، سپس از زیرپالت ظاهر شده گزینه‌ی Digital Control را انتخاب کنید.

۲۲- البته تا حدی که حافظه گنجایش داشته باشد.




**شکل ۳۴-۳: نحوه‌ی انتخاب المان Digital Control**

در این حالت، ماوس را در محلی از صفحه‌ی پانل که قصد دارید المان Digital Control در آن مکان قرار گیرد کلیک کنید. حال به همین روش، المان‌های String Control و Square Button را به ترتیب از زیرپالت‌های String & Table و Boolean انتخاب کنید و آنها را بر روی همین صفحه قرار دهید تا صفحه‌ای نظیر آنچه در شکل ۳۵-۳ نشان داده شده است حاصل گردد.




**شکل ۳۵-۳**

توجه داشته باشید هنگامی که یک المان را در صفحه‌ی پانل قرار می‌دهید، نرم‌افزار LabVIEW ترمینال متناظر با آن را در صفحه‌ی نمودار بلوکی ایجاد می‌کند. همچنین به خاطر داشته باشید که ترمینال‌های اعداد اعشاری به رنگ نارنجی، ترمینال‌های اعداد صحیح به رنگ آبی، اعداد جبری به رنگ سبز و رشته‌ها به رنگ صورتی می‌باشند. این مشخصات و تنوع رنگ‌ها شما را در تشخیص نوع داده‌ها یاری خواهند داد.

۴- اکنون منوی کرکره‌ای را بر روی المان Digital Control باز کرده، سپس گزینه‌ی Change to Indicator را انتخاب نمایید. به تغییر شکل المان مذکور توجه نموده، دقت کنید که در صورت اجرای این دستور پیکان‌های قرار گرفته در کنار جعبه‌ی مربوط به این المان حذف می‌شوند. حال تغییر یافتن این المان‌ها را در صفحه‌ی نمودار بلوکی ملاحظه کنید. توجه داشته باشید که حاشیه‌ی ترمینال مربوط به المان نشان‌دهنده نازک‌تر است. اکنون این المان را به حالت اولیه‌ی خود برگردانید. برای انجام این عمل، منوی کرکره‌ای را بر روی این المان در صفحه‌ی پانل باز نموده، سپس گزینه‌ی Change to Control را انتخاب کنید.

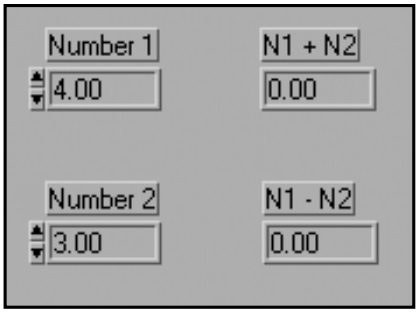
حال چندین مرتبه تمامی مراحل فوق را انجام دهید تا جایی که بتوانید به راحتی تفاوت بین ترمینال های مربوط به المان های کنترل و نشان دهنده را تشخیص دهید. توجه کنید که در مورد برخی از المان ها (نظیر تعدادی از المان های جبری) المان های کنترل و نشان دهنده در صفحه‌ی پانل به یک صورت ظاهر می شوند، اما ترمینال مربوط به آنها در صفحه‌ی نمودار بلوکی همواره متفاوت است.

اکنون تمامی مراحل فوق را در مورد دو المان دیگر تکرار کنید و در صورت اجرای دستورهای Change to Control / Indicator تفاوت ها را در مورد ترمینال ها و تغییر المان ها به دقت بررسی نمایید.

Positioning Tool

۵- ابزار Positioning Tool را از پالت Tools انتخاب کرده، سپس به کمک آن یکی از المان های موجود در صفحه‌ی پانل را انتخاب نمایید. برای حذف کردن این المان کلید <Delete> را بر روی صفحه کلید فشار دهید. حال به همین روش تمامی المان های موجود در این صفحه را حذف کنید تا جایی که در صفحه‌ی پانل هیچ المانی وجود نداشته باشد.

۶- اکنون مجدداً المان Digital Control را از زیرپالت Numeric << Controls انتخاب نموده، سپس آن را بر روی صفحه‌ی پانل قرار دهید. در صورتی که پس از انجام این عمل در هیچ نقطه ای کلیک نکنید، یک جعبه‌ی کوچک را در بالای چهارچوب مستطیل شکل مربوط به المان مشاهده می کنید. حال به کمک کلید های صفحه کلید عبارت Number 1 را تایپ کنید. ملاحظه می کنید که این عبارت در داخل جعبه‌ی مذکور ظاهر می گردد. برای وارد کردن متن نوشته شده بر روی دکمه‌ی Enter که در نوار ابزار قرار گرفته است کلیک کنید. با انجام این عمل در حقیقت برای المان Digital Control یک برچسب ایجاد کرده اید. حال به همین روش یک المان دیگر از نوع Ddigital Control با برچسب Number 2 و دو المان Digital Indicator با برچسبهای N1 + N2 و N1 - N2 ایجاد نمایید تا نمای ظاهری صفحه‌ی پانل این برنامه به صورت شکل ۳۶-۳ در آید.




**شکل ۳۶-۳: صفحه‌ی پانل در برنامه‌ی Add.vi**

Operating Tool

اکنون به کمک ابزار Operating Tool بر روی پیکان افزایش دهنده‌ی مربوط به Number 1 تا حدی کلیک کنید که در جعبه‌ی مربوط عدد 4.00 ظاهر شود. به همین روش عدد 3.00 را برای Number 2 ایجاد کنید.

۷- به صفحه‌ی نمودار بلوکی بازگردید. حال آیکن دستور Add را از زیرپالت Functions >> Numeric انتخاب کنید و آن را بر روی صفحه‌ی پانل قرار دهید. روش انجام این عمل نیز همانند روش قرار دادن المان‌ها بر روی صفحه‌ی پانل است. اکنون به همین روش آیکن دستور Subtract را بر روی صفحه‌ی پانل قرار دهید.

۸- منوی کرکره‌ای را بر روی دستور Add باز نموده، سپس گزینه‌ی Show Terminals را انتخاب نمایید. قبل از انتخاب این گزینه دقت کنید که علامت «✓» در کنار آن قرار ندارد و بدین معنی است که در حال حاضر ترمینال‌های مربوط به این دستور مخفی شده و یا نشان داده نشده‌اند. پس از ظاهر شدن ترمینال‌ها، به نحوه و ترتیب قرار گرفتن ترمینال‌های ورودی و خروجی دقت کنید. سپس مجدداً گزینه‌ی Show Terminals را انتخاب کنید. این بار در کنار گزینه علامت «✓» وجود دارد و بیانگر آن است که در حال حاضر ترمینال‌ها به نمایش در آمده‌اند.

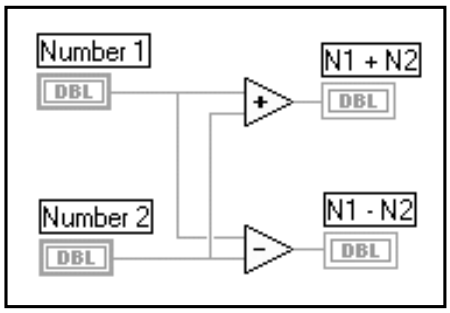
۹- با استفاده از میانبرهای صفحه کلید و یا با انتخاب گزینه‌ی Help >> Show Help، پنجره‌ی Help را فعال کنید. نشانگر ماوس را بر روی آیکن دستور Add در صفحه‌ی نمودار بلوکی قرار دهید. در این حالت در پنجره‌ی Help اطلاعاتی در مورد نحوه‌ی استفاده از این دستور و همچنین اتصالات ورودی و خروجی آن را ملاحظه می‌کنید. حال نشانگر ماوس را بر روی آیکن دستور Subtract قرار داده، در مورد این دستور نیز اطلاعاتی به دست آورید.

۳

Wiring Tool

۱۰- در صورت نیاز می‌توانید با استفاده از ابزار Positioning Tool محل قرار گرفتن برخی از ترمینال‌ها را تغییر داده، صفحه‌ی نمودار بلوکی این برنامه را به صورت شکل ۳-۳۷ در آورید. سپس به کمک ابزار Wiring Tool ترمینال‌ها را مطابق شکل ۳-۳۷ به یکدیگر ارتباط دهید. ابتدا این ابزار را از پالت Tools انتخاب نموده، سپس یک بار بر روی ترمینال مورد نظر در آیکن دستور Add یا Subtract و بار دیگر بر روی ترمینال DBL کلیک کنید. اگر سیم‌کشی را به درستی انجام داده باشید یک خط نارنجی رنگ ظاهر خواهد شد. در غیر این صورت به جای خط نارنجی رنگ، یک قطعه سیم سیاه خط چین بر روی صفحه ملاحظه می‌کنید. وجود این سیم سیاه رنگ خط چین بیانگر آن است که سیم‌کشی را به طور صحیح انجام نداده‌اید. در این صورت بخش خط چین را با استفاده از ابزار Positioning Tool انتخاب نموده، سپس کلید <Delete> را فشار دهید. این بار سیم‌کشی را بادقت انجام دهید.

برای انجام سیم‌کشی به طور صحیح ابتدا بر روی ترمینال مورد نظر کلیک نموده، سپس دکمه‌ی ماوس را رها کنید. این نقطه ابتدای سیم را نشان می‌دهد. سپس ماوس را حرکت دهید تا این ابزار بر روی نقطه‌ی مقصد قرار گیرد. حال مجدداً کلیک کنید تا سیم‌کشی پایان یابد.




**شکل ۳۷-۳: نحوه‌ی سیم‌کشی بین المان‌ها و ترمینال‌های موجود در صفحه‌ی نمودار بلوکی**

توجه داشته باشید که در زمانی که ابزار Wiring Tool را بر روی آیکن یکی از دستورهای Add یا Subtract قرار می‌دهید، سرسیمهای کوچکی در اطراف دستور نمایان می‌شوند. این سرسیمها نشان‌دهنده‌ی محل قرار گرفتن ترمینال‌ها هستند. علاوه بر این در صورتی که این ابزار را بر روی ترمینال‌ها قرار دهید نام آنها در کنار ترمینال ظاهر می‌شود.

انجام سیم‌کشی و برقرار کردن ارتباط بین عناصر، همانند فراگیری ماشین‌نویسی در ابتدا ممکن است مشکل به نظر برسد. به مرور زمان و با تمرینات بیشتر قادر خواهید بود به راحتی این عمل را در مدت زمانی کوتاه و بدون هیچ گونه خطایی انجام دهید. بنابراین اگر در حال حاضر در برقرار نمودن ارتباط بین ترمینال‌ها دچار مشکل هستید جای هیچ نگرانی نیست.

۱۱- حال به صفحه‌ی پانل بازگشته، منوی کرکره‌ای را بر روی قاب آیکن این برنامه باز کنید. این آیکن درگوشه‌ی بالایی سمت راست صفحه‌ی پانل قرار دارد. گزینه‌ی Show Connector را از این منو انتخاب کنید. ملاحظه می‌کنید که به جای آیکن، یک کانکتور ظاهر می‌شود. اگر قادر نیستید تا کانکتور را به کمک منوی کرکره‌ای فعال کنید، احتمالاً در صفحه‌ی نمودار بلوکی هستید و سعی دارید منوی کرکره‌ای را بر روی آیکن موجود در این صفحه باز کنید.

**شکل ۳۸-۳: نحوه‌ی انتخاب گزینه‌ی Show Connector از منوی کرکره‌ای مربوط به آیکن در صفحه‌ی پانل برنامه**

اکنون منوی کرکره‌ای را بر روی قاب کانکتور باز نموده، گزینه‌های موجود در آن را بررسی کنید. همان گونه که قبلاً اشاره شد، اتصالات کانکتور، پارامترهای ورودی و خروجی را در یک VI تعریف می‌کند به صورتی که بتوانید آن را به عنوان SubVI در برنامه‌های دیگر مورد استفاده قرار دهید و از طریق این پارامترها مقادیر ورودی و خروجی را به آن اعمال نموده، یا از آن دریافت کنید. بسته به تعداد پارامترهای مورد نیاز،

می توانید الگوهای مختلفی را برای یک کانکتور مورد استفاده قرار دهید. برای مشاهده‌ی قاب آیکن، گزینه‌ی Show Icon را از منوی کرکره‌ای قاب کانکتور انتخاب کنید تا آیکن مربوط ظاهر شود. به خاطر داشته باشید که آیکن، تنها یک نماد گرافیکی برای نمایش برنامه است. هنگامی که یک برنامه را به عنوان زیربرنامه به کار می برید، در حقیقت این آیکن را به ترمینال های دیگری که در صفحه‌ی نمودار بلوکی مربوط به برنامه‌ی سطح بالاتر قرار دارند سیم کشی می کنید. روش سیم کشی و برقراری ارتباط نظیر همان روشی است که در مورد دستور Add توضیح داده شد.

Run Button

۱۲- با فشار دادن دکمه‌ی Run، این برنامه را اجرا کنید. المان های نشان دهنده که با دو برچسب N1+ N2 و N1-N2 متمایز شده اند باید به ترتیب اعداد 7.00 و 1.00 را نمایش دهند. حال مقادیر ورودیها یعنی المان های کنترل که با برچسبهای N1 و N2 مشخص شده اند را تغییر داده و چند بار این برنامه را اجرا کنید.

۱۳- با انتخاب گزینه‌ی File >> Save این برنامه را با عنوان Add.vi در زیرفهرست LabVIEW\My Activity ذخیره نمایید.

## خلاصه

محیط برنامه نویسی LabVIEW شامل سه بخش اساسی پانل، نمودار بلوکی و آیکن/ کانکتور است. پانل، صفحه‌ای است که کاربر از طریق آن با برنامه ارتباط برقرار می کند. در این صفحه می توانید ازطریق المان های کنترل مقادیر داده های ورودی را به برنامه اعمال کرده و از طریق المان های نشان دهنده مقادیر خروجی را مشاهده کنید.

۳

هنگامی که از پالت Controls المانی را انتخاب می‌کنید و آن را بر روی صفحه‌ی پانل قرار می دهید، یک ترمینال متناظر با آن در صفحه‌ی نمودار بلوکی ایجاد می گردد. برای برقراری ارتباط بین گره ها از سیم استفاده می شود. گره ها المان های اجرایی نرم افزار LabVIEW هستند. یک گره، تنها در صورتی اجرا می شود که برای تمامی ورودیهای آن مقداری تعیین شده باشد. این قاعده را اصل جریانِ داده گویند.

هر برنامه دارای یک زوج آیکن /کانکتور است. در صورت استفاده از یک برنامه به عنوان زیربرنامه در برنامه های مراتب بالاتر، این آیکن نمایشگر برنامه می باشد. قاب کانکتور در زیر قاب آیکن قرار دارد و پارامترهای ورودی و خروجی زیربرنامه را تعریف می کند.

در نرم افزار LabVIEW دو نوع منوی پایین کشیدنی و کرکره‌ای وجود دارد. منوهای پایین کشیدنی در نوار بالای پنجره قرار دارند. در حالی که منوی کرکره‌ای در مورد هر المان و در هر محل قابل دستیابی است. برای باز کردن این منو کافی است در محیط Windows توسط کلید سمت راست ماوس بر روی المان مربوط کلیک کنید. در مورد محیط Mac این عمل را با پایین نگه داشتن کلید <Cmd> و سپس کلیک کردن انجام دهید. منوهای پایین کشیدنی شامل دستورهای عمومی و کلی هستند. اما فرمانها و دستورهای موجود در منوی کرکره‌ای تنها در مورد المانی که این منو را بر روی آن باز نموده اید به کار می رود.

پالت Tools امکان دستیابی به مدهای مختلف عملیاتی نشانگر ماوس را فراهم می سازد. از ابزارهای

موجود در این پالت می توانید جهت اعمال تغییرات و اصلاحات خاص و اجرای دستورها استفاده کنید. روش به کارگیری این ابزارها نظیر روش استفاده از آنها در محیط Paint است.

تمامی المان های کنترل و نشان دهنده که در صفحه‌ی پانل قرار می گیرند در پالت Controls یا زیرپالت های آن قرار دارند. اما اعداد ثابت، دستورها و ساختارهایی که در صفحه‌ی نمودار بلوکی مورد استفاده قرار می گیرند در پالت Functions یا زیرپالت های آن وجود دارند.

پنجره‌ی Help حاوی اطلاعاتی در مورد دستورها، نحوه‌ی سیم کشی و برقرار نمودن ارتباط با المان های دیگر است. از طریق منوی Help می توانید به اطلاعات مذکور دسترسی پیدا کنید. در نرم‌افزار LabVIEW می توان با انتخاب گزینه‌ی ...Help >> Online References و یا کلیک کردن بر روی دکمه‌ی Online Help در پنجره‌ی Help، به اطلاعات ارزشمندی دست یافت.

در نرم‌افزار LabVIEW می توانید هر یک از برنامه ها را با ایجاد آیکن و کانکتور به عنوان زیربرنامه در برنامه های مراتب بالاتر مورد استفاده قرار دهید. این زیربرنامه ها کاملاً مدولار هستند و ویژگیهایی همچون سهولت در رفع اشکال و درک ساده تر عملکرد برنامه های پیچیده را به ارمغان می آورند.

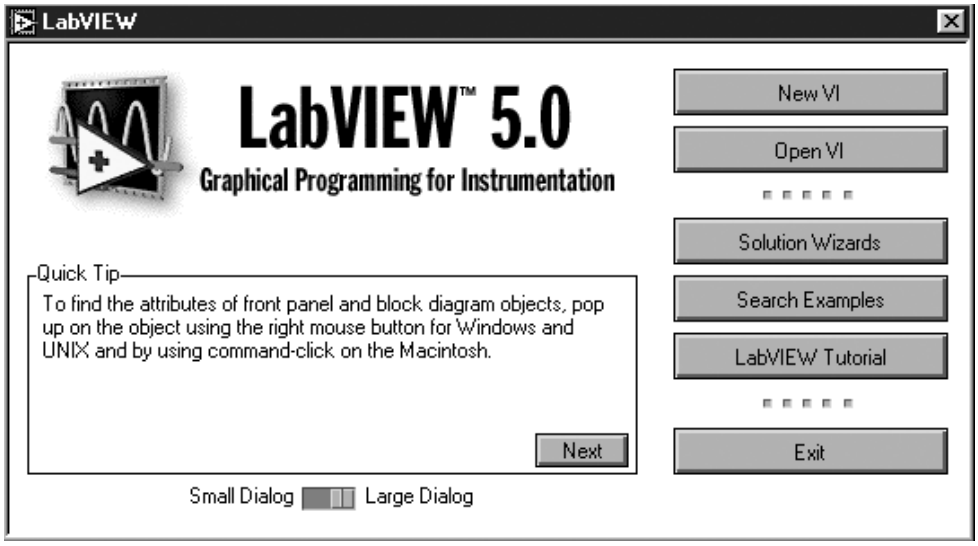
اگر نمی توانید تمامی مطالب فوق را به خاطر بسپارید، جای هیچ نگرانی نیست. زیرا به تدریج و در حین انجام تمرینات با این مطالب به طور کامل آشنا خواهید شد.

## تمرینات اضافی

## تمرین ۳-۳: آموزش LabVIEW

در این تمرین قصد داریم با مراجعه به لوح فشرده‌ی همراه این کتاب، مطالبی را در مورد نرم‌افزار LabVIEW بیاموزیم.

۱- نرم‌افزار LabVIEW را بارگذاری کنید. در این حالت پنجره ای مطابق شکل ۳۹-۳ ظاهر می شود.



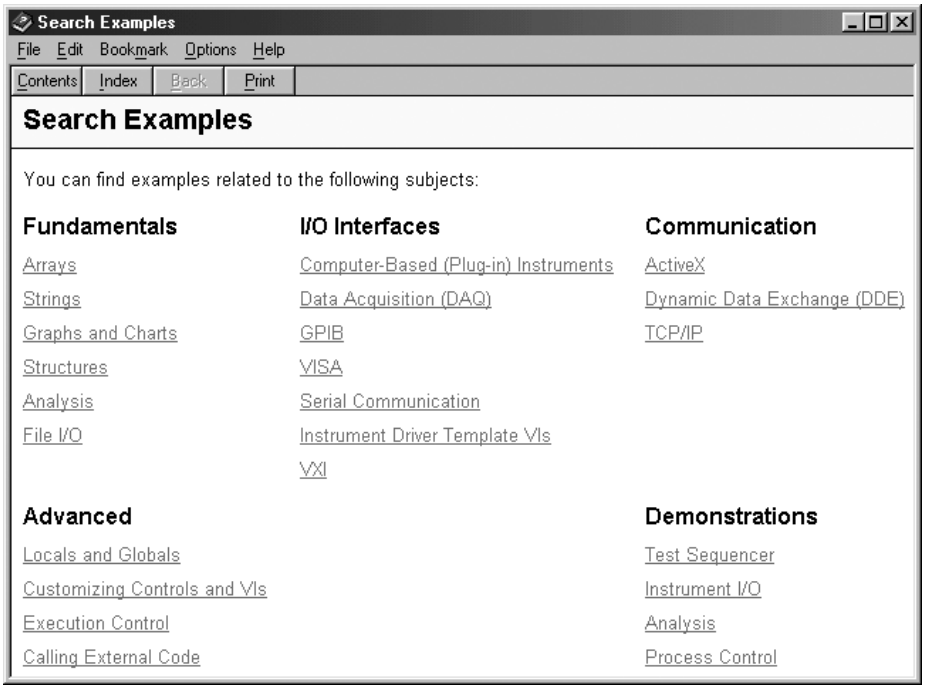

شکل ۳۹-۳

۲- لوح فشرده را در داخل گرداننده قرار داده، سپس بر روی دکمه‌ی LabVIEW Tutorial کلیک کنید. در این حالت در مورد معرفی و کاربردهای این نرم‌افزار توضیحاتی به صورت صوتی و تصویری ارائه می‌شود. از این آموزش لذت ببرید.

## تمرین ۳-۴: جستجوی مثالها و برنامه‌ها در محیط LabVIEW

۱- نرم‌افزار LabVIEW را بارگذاری کنید. در این حالت پنجره‌ای مطابق شکل ۳-۳۹ باز می‌شود.

۲- سپس بر روی دکمه‌ی Search Examples کلیک کنید. در این حالت پنجره‌ای مطابق شکل ۳-۴۰ ظاهر می‌گردد.



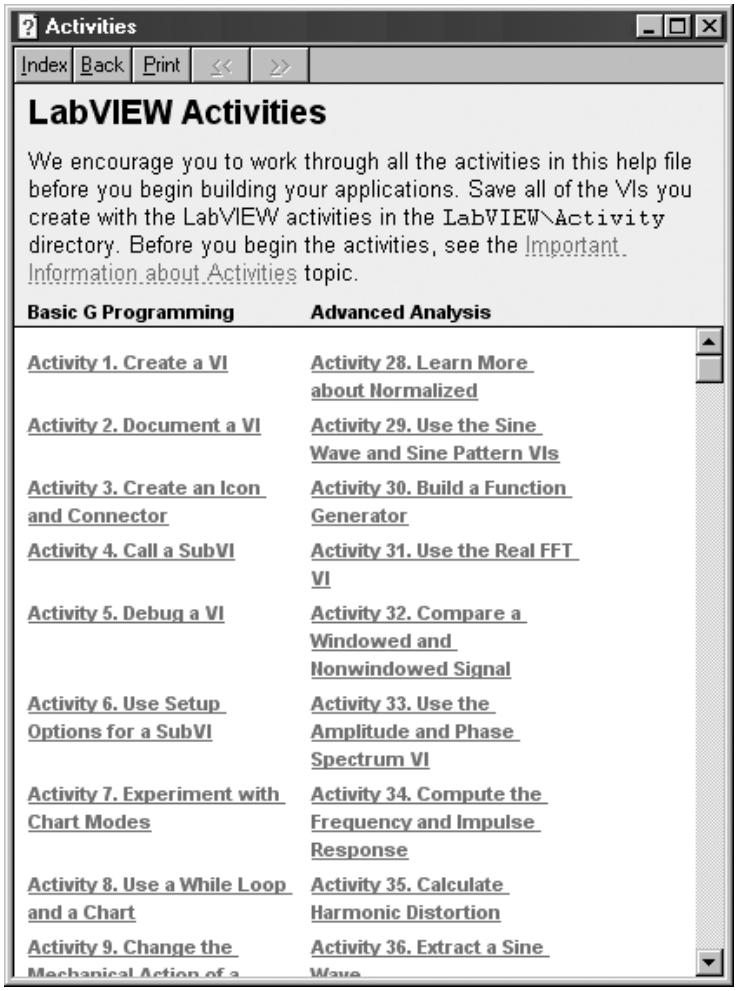

**شکل ۳-۴۰: پنجره‌ی محاوره‌ای Search Examples**

۳- اکنون یکی از موضوعات مندرج در شکل ۳-۴۰ را انتخاب نموده، سپس توضیحات و مثالهای مربوط را ملاحظه کنید. همان گونه که قبلاً نیز عنوان شد با مطالعه‌ی مثالها و برنامه‌های نوشته شده در محیط LabVIEW می‌توان به نکاتی بسیار ارزشمند دست یافت.

## تمرین ۳-۵: فراگیری برنامه‌نویسی با استفاده از تمرینات

۱- گزینه‌ی Help >> Online References... را انتخاب کنید.

۲- پس از ظاهر شدن پنجره‌ی محاوره‌ای LabVIEW Help، گزینه‌ی Learn LabVIEW with Activities را از این پنجره انتخاب نمایید. در این حالت یک پنجره‌ی محاوره‌ای مطابق شکل ۳-۴۱ ظاهر می‌گردد.




**شکل ۴۱-۳: پنجره‌ی محاوره‌ای Activities**

۳- تمرینات موجود در این بخش را به دقت بررسی کنید و پس از مطالعه‌ی هر فصل تمرینات مربوط به آن را گام به گام دنبال نمایید.

## تمرین ۶-۳: شناخت نرم‌افزار LabVIEW

۱- لوح فشرده‌ی همراه این کتاب را در داخل گرداننده قرار داده، سپس فایل Manuals >> Quckstrt.pdf را باز کنید.

۲- مطالب مندرج در این فایل را به دقت مطالعه نموده، مراحل عنوان شده در آن را برای ایجاد یک برنامه در محیط LabVIEW دنبال کنید. به دلیل اینکه توضیحات مندرج در این فایل با توجه به نسخه های قبلی ارائه شده است، ممکن است منوها، گزینه ها و همچنین پالت های شناور موجود در شکلهای ترسیم شده در این فایل، با آنچه که شما بر روی صفحه‌ی نمایشگر رایانه‌ی خود مشاهده می کنید اندکی متفاوت باشد.

سه تمرین اخیر در حقیقت خودآموز محیط برنامه‌نویسی LabVIEW محسوب می‌شوند. در لوح فشرده‌ی همراه این کتاب مطالب بسیار ارزشمند دیگری وجود دارد. بنابراین سعی کنید با مطالعه‌ی این مطالب برنامه‌نویسی در محیط LabVIEW را به سرعت فرا گیرید.

به شما تبریک می گوییم. تا این مرحله تعدادی از دستورهای مهم و اساسی و همچنین جزییات صفحات پانل و نمودار بلوکی را فرا گرفته‌اید.

LabVIEW

# ۴

# ایجاد برنامه و اجرای آن در محیط LabVIEW

## در این فصل با موارد زیر آشنا می‌شوید:

- روشهای ایجاد تغییر و اصلاح برنامه در نرم‌افزار LabVIEW
- المان‌های کنترل و نشان‌دهنده و گزینه‌های آنها
- اصول ایجاد یک برنامه و روش سیم‌کشی و ویرایش برنامه
- روش ایجاد یک برنامه‌ی ساده و اجرا نمودن آن

# LabVIEW

۴

## ایجاد برنامه و اجرای آن در محیط LabVIEW

خود را برای فراگیری اصول برنامه نویسی در محیط LabVIEW آماده کنید. در این فصل سعی داریم تا به بررسی انواع مختلف داده‌ها و روش ایجاد آنها، انجام سیم‌کشی و نحوه‌ی اجرای برنامه بپردازیم. در این قسمت همچنین روشهای میانبری را برای سرعت بخشیدن به پیشرفت‌تان خواهید آموخت. قبل از به کارگیری این اصول، از فراگیری و درک کامل آنها اطمینان حاصل کنید. زیرا درک کامل اصول مذکور منجر به پیشرفت شما در فراگیری بسته‌ی نرم‌افزاری LabVIEW می گردد.

در این فصل اهداف زیر دنبال می شوند:

- آشنایی با روشهای ایجاد تغییر و اصلاح در نرم‌افزار LabVIEW.
- آشنایی با المان های کنترل و نشان دهنده و گزینه های مربوط به هر کدام.
- فراگیری اصول ایجاد یک برنامه، انجام سیم کشی و ویرایش برنامه.
- ایجاد یک برنامه ی ساده و اجرا نمودن آن.

اصطلاحات کلیدی موجود در این فصل عبارتند از:

- اولویتها (Preferences)
- مقادیر عددی (Numeric)
- مقادیر رشته ای (String)
- مقادیر جبری (Boolean)
- مسیر (Path)
- منوی حلقوی (Ring)
- فرمت و دقت ارقام اعشاری (Format and Precision)
- نمایش عددی (Numeric Representation)
- برچسب (Label)

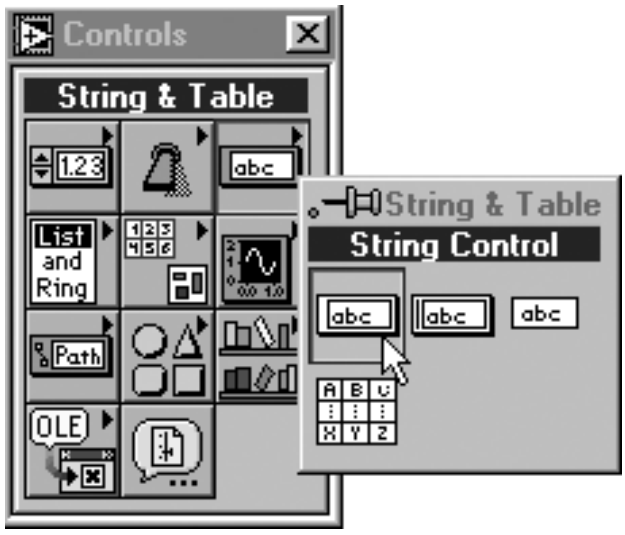
## ایجاد یک برنامه

تاکنون تعدادی از اصول برنامه نویسی در محیط LabVIEW را بررسی نموده‌ایم. حال قصد داریم تا روش ایجاد یک برنامه راگام به گام دنبال کنیم. به منظور یادگیری بیشتر به شما توصیه می‌کنیم تا این دستورها را مرحله به مرحله بر روی رایانه‌ی خود اجرا نمایید. بدین ترتیب فراگیری تکنیک‌ها و روشهای برنامه نویسی بسیار ساده خواهد بود.

## قرار دادن المان‌ها بر روی صفحه‌ی پانل

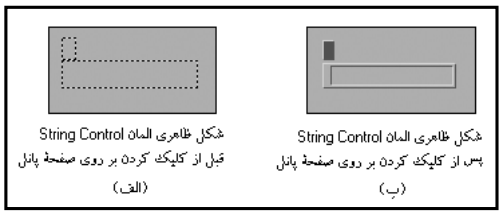
معمولاً در آغاز برنامه نویسی و برای تعریف ورودیها و خروجیهای برنامه لازم است تا المان‌های کنترل و نشان‌دهنده را بر روی صفحه‌ی پانل قرار دهید. در فصل قبل، یک یا دو مرتبه این عمل را انجام داده‌اید. اما در این فصل، انجام این عمل را به طور کامل مورد بررسی قرار می‌دهیم.

پالت Controls را باز نموده، نشانگر ماوس را به آرامی بر روی دکمه‌های موجود در این پالت و زیرپالت‌های آن حرکت دهید. ملاحظه می‌کنید که نام زیرپالت‌ها در بالای پنجره‌ی پالت Controls ظاهر می‌شود. برای مشاهده المان‌های موجود در یک زیرپالت، بر روی آن زیرپالت کلیک کنید. برای انتخاب المان مطلوب از زیرپالت، ماوس را بر روی آن رها کنید. توجه داشته باشید هنگامی که نشانگر ماوس بر روی المان مذکور قرار دارد نام آن در بالای پنجره‌ی زیرپالت انتخاب شده ظاهر می‌گردد. در شکل ۱-۴ نحوه انتخاب المان String Control نشان داده شده است.




**شکل ۱-۴: نحوه انتخاب المان String Control**

حال بر روی صفحه‌ی پانل و در محلی که قصد دارید المان مورد نظر در آن جا قرار گیرد، کلیک کنید. ملاحظه می‌کنید که المان مذکور در محل مطلوب ایجاد می‌شود. در شکل ۲-۴ (الف) و (ب) شکل ظاهری المان String Control را قبل و بعد از کلیک کردن بر روی صفحه‌ی پانل ملاحظه می‌کنید.




**شکل ۲-۴**

روش دیگر برای دستیابی به پالت Controls، باز کردن منوی کرکره‌ای در فضایی خالی از صفحه‌ی پانل است.

حال که با روش قرار دادن المان‌ها بر روی صفحه‌ی پانل آشنا شدید، یک پنجره‌ی جدید باز کنید و یک المان Digital Control را بر روی صفحه‌ی پانل قرار دهید.

به خاطر داشته باشید هنگامی که یک المان را بر روی صفحه‌ی پانل قرار می‌دهید، ترمینال متناظر با آن بر روی صفحه‌ی نمودار بلوکی ایجاد می‌گردد. برای مشاهده هر دو صفحه‌ی پانل و نمودار بلوکی، گزینه Windows >> Tile Left and Right را انتخاب کنید.

## تعیین برچسب برای المان

برچسب یا Label، متن کوتاهی است که توسط آن می‌توان المان‌های موجود در صفحات پانل و نمودار بلوکی را نام‌گذاری کرد. هنگامی که المانی را بر روی صفحه‌ی پانل قرار می‌دهید، یک مستطیل کوچک که محل نوشتن برچسب است در بالای المان ظاهر می‌گردد. در صورتی که قصد داشته باشید در این مرحله، المان مذکور را برچسب‌گذاری کنید، متن یا برچسب مورد نظر را با استفاده از صفحه کلید وارد نمایید. اگر تصمیم دارید تا در مراحل بعدی برای المان مورد بحث برچسب تعیین کنید و یا اصلاً قصد تعیین برچسب برای آن المان را ندارید، ماوس را در هر نقطه‌ی دیگری کلیک کنید. ملاحظه می‌کنید که مستطیل کوچک ناپدید می‌شود. پس از اینکه تایپ کردن برچسب مورد نظر را به پایان رساندید، با انجام هر یک از مراحل زیر برچسب‌گذاری را تکمیل کنید. توجه داشته باشید که در صورت عدم انجام یکی از مراحل زیر، برچسب مذکور وارد نمی‌گردد.

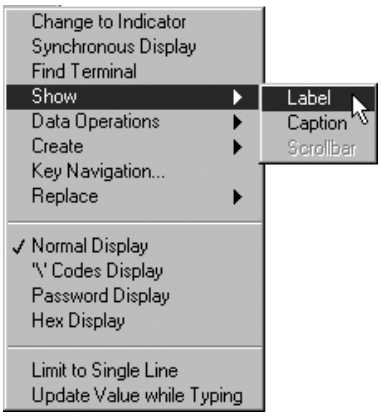
- کلید <Enter> را که در بخش عددی بر روی صفحه کلید قرار دارد فشار دهید.
- در محلی خارج از مستطیل کوچک و بر روی یکی از صفحات پانل یا نمودار بلوکی کلیک نمایید.
- بر روی دکمه‌ی Enter در نوار ابزار کلیک کنید.
- به کمک صفحه کلید در دو محیط Windows و Mac به ترتیب از ترکیب کلیدهای <Shift-Enter> و <Shift-Return> استفاده کنید. نیاز به توضیح است که در این مورد باید کلیدهای <Enter> و <Return> که در بخش حروف الفبایی صفحه کلید قرار دارند را به کار برید.

در این حالت برچسب مورد نظر در بالای المان در صفحه‌ی پانل و همچنین در بالای ترمینال متناظر با آن در

صفحه‌ی نمودار بلوکی ظاهر می‌شود. در نرم‌افزار LabVIEW دو نوع برچسب وجود دارد: برچسبهای اختصاصی[1] و آزاد[2].

برچسبهای اختصاصی به همان المان تعلق دارند و همراه با آن جا به‌جا می‌شوند. این برچسبها تنها برای نام‌گذاری یک المان خاص به‌کار برده می‌شوند. هنگامی که یک المان کنترل یا نشان‌دهنده را بر روی صفحه‌ی پانل قرار می‌دهید، یک مستطیل کوچک در بالای المان که محل قرار گرفتن برچسب اختصاصی آن المان است ظاهر می‌شود و آماده دریافت متن یا برچسب مورد نظر است. برچسب اختصاصی در مورد المان قرار گرفته در صفحه‌ی پانل و ترمینال متناظر با آن در صفحه‌ی نمودار بلوکی یکسان است؛ در حالی که برچسب آزاد وابسته به المان خاصی نیست و می‌توان آن را هر لحظه و در هر مکان ایجاد کرد و سپس آن را حذف نمود.

با انتخاب گزینه‌ی Show >> Label از منوی کرکره‌ای مربوط به هر المان می‌توان برچسبی را که درحال حاضر قابل رؤیت نیست، ایجاد نمود و یا آن را تغییر داد. همچنین می‌توانید برچسبهای اختصاصی المان را مخفی کنید. به خاطر داشته باشید که نمی‌توانید برچسب اختصاصی یک المان را جدای از المان کپی یا حذف نمایید. در شکل ۳-۴ روش دستیابی به برچسب المان String Control نشان داده شده است.




**شکل ۳-۴: روش ظاهر ساختن برچسب المان String Control**

برای هر یک از ساختارها و دستورها یک برچسب به صورت پیش‌فرض در نظر گرفته شده است. این برچسبها تا زمانی که قصد نمایش دادن آنها را نداشته باشید، مخفی باقی می‌مانند. در ضمن می‌توانید جهت مشخص کردن عملیات انجام گرفته و یا برای شرح وظایف المان، برچسب آن را تغییر دهید. در مورد زیربرنامه تنها می‌توانید برچسب آن را به نمایش در آورید. این برچسب در حقیقت نام زیربرنامه است. در این حالت تغییر دادن برچسب امکان‌پذیر نیست.

حال برای المان Digital Control برچسب Number 1 را تعیین کنید. در صورتی که پس از ایجاد المان در محلی از صفحه‌ی پانل کلیک کرده باشید، حتماً مستطیل کوچکی که در بالای المان و محل قرار گرفتن برچسب می‌باشد حذف شده است. برای به نمایش در آوردن آن، منوی کرکره‌ای را بر روی المان مورد نظر باز نموده، گزینه‌ی Show >> Label را انتخاب کنید.

1- Owned Labels
2- Free Labels

## ایجاد برچسب آزاد

همان گونه که قبلاً اشاره شد برچسبهای آزاد، وابسته به هیچ المانی نیستند و می توانید آنها را به طور مجزا ایجاد نموده، حرکت دهید و یا حذف کنید. این برچسبها معمولاً جهت نام گذاری بخش‌هایی از صفحات پانل و نمودار بلوکی مورد استفاده قرار می گیرند. برای ایجاد این برچسبها از ابزار Labeling Tool استفاده نمایید. به کمک این ابزار قادر خواهید بود تا برچسب مورد نظر را ایجاد کنید یا آن را تغییر دهید.

Labeling Tool

برای تعیین برچسب آزاد، ابزار Labeling Tool را از پالت Tools انتخاب نموده و در یک نقطه‌ی خالی از صفحه‌ی مورد نظر کلیک کنید. در این حالت ملاحظه می کنید که یک مستطیل کوچک حاشیه‌دار ظاهر می شود. برچسب مورد نظر را تایپ کنید و آن را به یکی از ۴ روش ذکر شده وارد نمایید. در صورتی که در مستطیل مذکور هیچ گونه عبارتی را تایپ نکنید، به محض کلیک کردن ماوس در هر نقطه‌ی دیگر، آن برچسب ناپدید می گردد. حال یک برچسب آزاد با عنوان Free Label بر روی صفحه‌ی پانل ایجاد کنید.

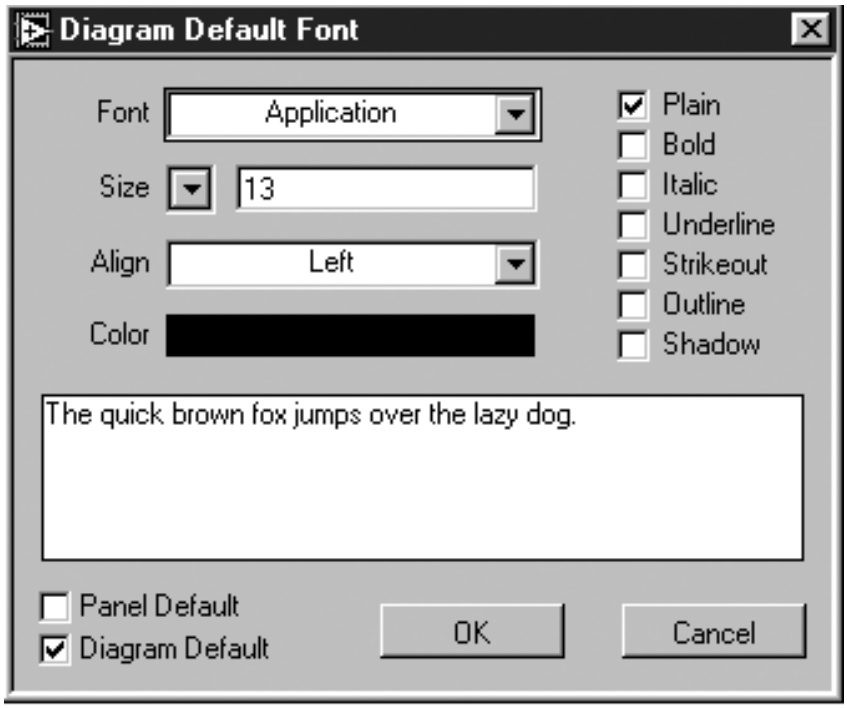
## تغییر رنگ، شکل واندازه‌ی فونت‌ها

با کمک منوی حلقوی Font می توانید مشخصات فونت ها را تغییر دهید. برای تغییر فونت ابتدا المان مورد نظر را توسط ابزار Positioning Tool انتخاب نموده، یا با استفاده از ابزارهای Labeling Tool یا Operating Tool متن مورد نظر را متمایز کنید. سپس گزینه‌ای را از منوی حلقوی Font انتخاب نمایید. متن متمایز شده مطابق با گزینه‌ای که انتخاب نموده‌اید تغییر می یابد. در صورت عدم انتخاب متن برای انجام تغییرات، مشخصات پیش فرض برای فونت ها در نظر گرفته می شود. در شکل ۴-۴ منوی حلقوی Font را ملاحظه می کنید.

**شکل ۴-۴ : منوی حلقوی Font**

حال اندازه‌ی فونت استفاده شده در برچسب Free Label را از اندازه‌ی پیش فرض 13 به 18 تغییر دهید. در صورت انتخاب گزینه‌ی ...Font Dialog از این منوی حلقوی، یک پنجره‌ی محاوره‌ای مطابق شکل ۵-۴ بر روی صفحه ظاهر می شود. به کمک این پنجره می توانید چند مشخصه‌ی فونت را تغییر دهید.

همان گونه که در شکل ۵-۴ ملاحظه می کنید برای هر یک از صفحات پانل و نمودار بلوکی می توان مشخصات فونت ها را به طور جداگانه تعیین نمود. در صورت در اختیار داشتن فونت های زبان فارسی می توان از این فونت ها نیز در تعیین برچسب المان ها استفاده نمود. در نرم‌افزار LabVIEW در موارد مختلف از سه نوع فونت با نامهای Dialog، Application و System استفاده می شود. مشخصات این فونت ها به صورت پیش فرض بوده، هرگونه تغییر در آنها باعث ایجاد تغییر در تمامی مواردی خواهد شد که از این فونت ها استفاده شده است.




**شکل ۵-۴: پنجره محاوره‌ای Font Dialog که برای انتخاب مشخصات فونت ظاهر می شود.**

- فونت Application به صورت پیش فرض انخاب شده و در مورد پالت های Controls، Functions و متون نوشتاری برای المان های کنترل و نشان دهنده‌ی جدید به کار برده می شود.
- فونت System در مورد منوها به کار برده می شود.
- فونت Dialog در پنجره های محاوره‌ای استفاده می شود.

## قرار دادن المان‌ها در صفحه‌ی نمودار بلوکی

اگر برنامه ای برای پشتیبانی صفحه‌ی پانل وجود نداشته باشد استفاده از آن امکان پذیر نخواهد بود. در حقیقت برنامه واقعی با قرار دادن دستورها، زیربرنامه ها و ساختارها در صفحه‌ی نمودار بلوکی ایجاد می گردد. برای قراردادن المان در صفحه‌ی مذکور به همان روش ذکر شده در مورد پالت Controls گزینه‌ی مورد نظر را از زیرپالت های موجود در پالت Functions انتخاب کنید و آن را با یک کلیک بر روی صفحه‌ی نمودار بلوکی قرار دهید.

۴

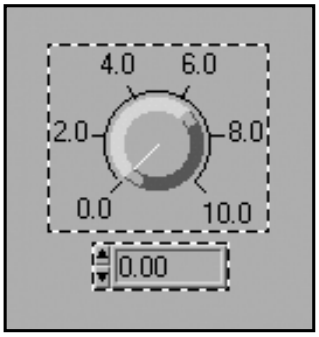
## ویرایش برنامه

وقتی که چند المان در صفحه وجود داشته باشند ممکن است نیاز به جابه‌جا کردن، کپی برداری، تغییر رنگ، حذف و... داشته باشید. در ادامه به بررسی این موارد و روشهای ویراستاری برنامه می پردازیم:

## انتخاب المان‌ها

Positioning Tool

قبل از جابه‌جا کردن هر المان ابتدا باید آن را انتخاب کنید. برای انتخاب هر یک از المان‌ها، به کمک Positionirg Tool بر روی المان مورد نظر کلیک نمایید. پس از انتخاب، المان مورد نظر با یک حاشیه‌ی خط‌چین متحرک احاطه می‌گردد[۳]. این حاشیه در شکل ۶-۴ نشان داده شده است.




شکل ۶-۴

برای انتخاب بیش از یک المان، کلید <Shift> را فشار دهید و به کمک ابزار Positioning Tool بر روی المان ها کلیک کنید. برای اعلام انصراف از انتخاب یک المان نیز می توانید کلید <Shift> را فشار دهید و بر روی آن المان کلیک کنید.

روش دیگر برای انتخاب یک/ چند المان، محصور نمودن آن/آنها در داخل یک مستطیل انتخابی است. برای انجام این عمل، به کمک ابزار Positionirg Tool در فضایی خالی در اطراف المان/المان‌های مورد نظر کلیک کنید و ماوس را به صورت قطری حرکت دهید تا المان/المان های مورد نظر در داخل مستطیل ظاهر شده قرار گیرند. هنگامی که دکمه‌ی ماوس را رها می کنید، مستطیل انتخابی ناپدید می‌شود و هر یک از المان های انتخاب شده توسط حاشیه‌ی خط چین متحرک محصور می گردند. به دلیل اینکه حاشیه‌ی مذکور به صورت خط چین و متحرک ظاهر می شود گاهی به آن «مورچه های رژه رونده» نیز گویند. پس از انتخاب المان های مورد نظر می توانید آنها را حرکت داده، کپی و یا حذف کنید.

توجه داشته باشید که انتخاب همزمان یک المان از صفحه‌ی پانل و المان دیگر از صفحه‌ی نمودار بلوکی امکان پذیر نیست. اما می توانید همزمان چند المان را از صفحه‌ی پانل یا نمودار بلوکی انتخاب نمایید.

کلیک کردن بر روی یک المان انتخاب شده و یا بر روی ناحیه ای خالی از صفحه، به منزله‌ی انصراف از انتخاب تمامی المان های انتخاب شده می باشد. در یک مجموعه از المان های انتخاب شده برای انصراف از عدم انتخاب یک یا چند المان، کلید <Shift> را پایین نگه دارید و سپس بر روی المان/المان های مورد نظر کلیک کنید.

حال که با روش انتخاب المان ها آشنا شدید، المان Digital Control را که قبلاً ایجاد نمودید انتخاب کنید.

۳- به این حاشیه‌ی خط چین متحرک در اصطلاح marquee گفته می شود.

## حرکت دادن المان‌ها

برای حرکت دادن یک المان ابتدا آن را انتخاب نموده، سپس آن را به محل مورد نظر انتقال دهید. اگر کلید <Shift> را فشار دهید و سپس المان مورد نظر را حرکت دهید، جهت حرکت به صورت افقی یا عمودی محدود می گردد. جهت انتخاب شده توسط LabVIEW به جهتی بستگی دارد که اولین بار المان را به آن سمت حرکت داده‌اید. همچنین می توانید به کمک کلیدهای مکان‌نما، المان مورد نظر را در فواصل کوتاه و بسیار دقیق حرکت دهید. بدین ترتیب که کلید <Shift> را پایین نگه دارید و در همان حال یکی از کلیدهای مکان‌نما را برای حرکت دادن المان در فواصل کوتاه و دقیق به کار برید.

اگر در حین حرکت دادن المان، نظر خود را در مورد جابه‌جا نمودن المان مذکور تغییر دادید، نشانگر ماوس را به خارج از تمامی پنجره های موجود بر روی صفحه انتقال دهید تا مستطیل خط چین حذف شود. سپس دکمه‌ی ماوس را رها کنید تا المان مذکور به مکان اولیه‌ی خود باز گردد. اگر المانی در حین جابه‌جا شدن بر روی لبه‌ی پنجره واقع شده باشد، پنجره به صورت خودکار می لغزد تا المان جابه‌جا شده به خوبی رؤیت شود.

حال که با نحوه حرکت دادن المان‌ها در صفحه آشنا شدید، المان Digital Control را به طرف دیگر صفحه انتقال دهید.

## کپی برداری[4] از المان‌ها

Positioning Tool

پس از انتخاب هر المان می توانید در صورت نیاز آن را کپی کنید. برای انجام این عمل، گزینه‌ی Edit >> Copy را انتخاب کنید و در محلی که قصد دارید المان جدید در آن جا قرار گیرد کلیک کنید. سپس گزینه‌ی Paste را از همان منو انتخاب نمایید. برای انجام این عمل به کمک صفحه کلید می توانید به ترتیب در دو محیط Windows و Mac، کلیدهای <Ctrl> و <Option> را فشار دهید و در همان حال با استفاده از ابزار Positioning Tool بر روی المانی که قصد کپی برداری از آن را دارید کلیک کنید. سپس در حالی که دکمه‌ی ماوس را پایین نگه داشته‌اید، ماوس را حرکت دهید. حال می توانید المان کپی شده را که در اطراف آن یک مستطیل انتخابی وجود دارد به نقطه‌ی دلخواه انتقال دهید.

توجه داشته باشید که المان اولیه در جای خود باقی می‌ماند. در ضمن می توانید المان‌های موجود در صفحات پانل و نمودار بلوکی را از یک برنامه به برنامه‌های دیگر کپی کنید. در صورتی که قسمتی از دستورهای موجود در صفحه‌ی نمودار بلوکی را به صفحه‌ی دیگری انتقال دهید، اتصالات و سیمهای مربوط به آن بخش نیز به همراه المان‌های دیگر منتقل می شوند. نکته‌ی جالب توجه این است که المان‌های متناظر با ترمینال‌های انتقال یافته در صفحه‌ی نمودار بلوکی، بر روی صفحه‌ی پانل ایجاد می شوند.

به خاطر داشته باشید که کپی برداری از المان‌های کنترل و نشان‌دهنده که مربوط به صفحه‌ی پانل می‌باشند از یک برنامه به صفحه‌ی نمودار بلوکی در برنامه‌ی دیگر امکان‌پذیر نیست. بلکه باید این المان‌ها را در صفحه‌ی پانل کپی نمود.

۴

4- Cloning

حال المان Digital Control را به هر دو روش ذکر شده یعنی با استفاده از ماوس و صفحه کلید کپی کنید. پس از انجام این عمل باید سه المان Digital Control با برچسبهای Number 1 Copy 2، Number 1 Copy و Number 1 بر روی صفحه‌ی پانل و سه ترمینال متناظر با آنها بر روی صفحه‌ی نمودار بلوکی وجود داشته باشد. برای اینکه متوجه شوید که هر یک از ترمینال ها مربوط به کدام یک از المان های Digital Control است، منوی کرکره‌ای را بر روی یک المان کنترل و یا یک ترمینال باز نموده، گزینه‌ی Find Terminal یا Find Control را انتخاب کنید. در این حالت، ترمینال یا المان متناظر در صفحه‌ی دیگر از سایر المان ها متمایز می گردد.

## حذف نمودن المان ها

برای حذف یک المان، ابتدا آن را انتخاب نموده، سپس گزینه‌ی Edit >> Clear را انتخاب کنید. با استفاده از صفحه کلید نیز می توانید پس از انتخاب المان مورد نظر در دو محیط Windows و Mac به ترتیب کلیدهای <Delete> و <Back space> را فشار دهید.

در مورد حذف المان ها توجه داشته باشید که تنها می توانید المان های کنترل و نشان دهنده را از صفحه‌ی پانل حذف نمایید. در صورتی که سعی در حذف ترمینال های موجود در صفحه‌ی نمودار بلوکی داشته باشید، فرمان «حذف» انجام نخواهد شد.

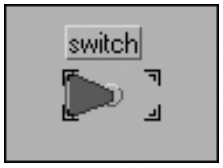
شایان ذکر است که حذف برخی از اجزای المان های کنترل و نشان دهنده نظیر برچسبها و نشان دهنده های دیجیتالی امکان پذیر نیست. به جای حذف این اجزا می توانید با انتخاب گزینه‌ی Show از منوی کرکره‌ای و سپس غیرفعال نمودن زیرگزینه‌ی مورد نظر هر یک از اجزای موجود در منو را مخفی کنید.

حال که روش حذف کردن المان ها را آموختید یکی از المان های Digital Control را حذف کنید.

## تغییر اندازه‌ی المان ها

Positioning Tool

در نرم‌افزار LabVIEW می توان اندازه‌ی اکثر المان ها را تغییر داد. برای انجام این عمل ابزار Positioning Tool را بر روی المانی که قابلیت تغییر اندازه دارد قرار دهید. همان گونه که در شکل ۷-۴ ملاحظه می کنید در اطراف المان مذکور علائمی ظاهر می شود که بیانگر تغییر اندازه است.




شکل ۷-۴

هنگامی که ابزار Positioning Tool را بر روی یکی از علائم تغییر اندازه قرار دهید، نشانگر ماوس مطابق

شکل ۸-۴ به ابزار Resizing Tool تغییر شکل می دهد. حال برای تغییر اندازه، کلیک کنید و ماوس را تا جایی که حاشیه‌ی نقطه چین شده به اندازه‌ی دلخواه شما در آمده است حرکت دهید.

**شکل ۸-۴ : تغییر شکل ابزار Positioning Tool به Resizing Tool**

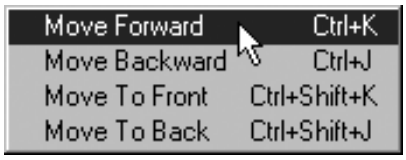
در صورتی که در حین تغییر اندازه، نظر شما در مورد انجام این عمل عوض شد به حرکت دادن ماوس ادامه دهید تا این که نشانگر ماوس به خارج از پنجره انتقال یابد. در این حالت مشاهده می کنید که چهار چوب نقطه چین شده ناپدید می شود. سپس دکمه‌ی ماوس را رها کنید. در این حالت المان مورد بحث به همان اندازه‌ی اولیه‌ی خود باز می گردد.

اکنون که با روش تغییر اندازه‌ی المان ها آشنا شدید، اندازه‌ی یکی از المان های Digital Control را تغییر دهید.

توجه داشته باشید که در مورد برخی از المان‌های کنترل و نشان‌دهنده نمی‌توان فرمان تغییر اندازه را به صورت کلی و عمومی اجرا نمود. بدین معنی که اندازه برخی از المان‌ها تنها به صورت افقی یا عمودی تغییر می‌کند.

## قرار دادن المان‌ها در صفحه‌ی پس‌زمینه و پیش‌زمینه

ممکن است در یک برنامه لازم باشد تا برخی از المان ها را نه در داخل صفحه بلکه بر روی صفحه قرار دهید. انجام این عمل نیز به نوبه خود ممکن است المان های دیگری را مخفی نماید؛ زیرا ممکن است المان های قرار گرفته بر روی صفحه، المان های موجود در داخل صفحه را بپوشانند. مطابق شکل ۹-۴ در منوی حلقوی Reorder چند دستور وجود دارد که به کمک آنها می توانید المان ها را نسبت به یکدیگر حرکت دهید.




**شکل ۹-۴ : دستورهای موجود در منوی حلقوی Reorder**

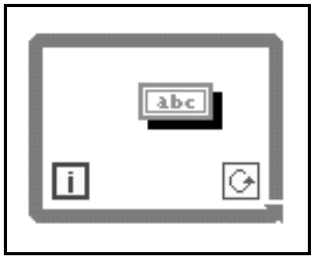
در ادامه به بررسی گزینه های موجود در این نوار می پردازیم:

دستور Move Forward : المان انتخاب شده را یک طبقه بالاتر از المان های دیگر قرار می دهد.

دستور Move To Front : المان انتخاب شده را بر روی دسته ای از المان های دیگر قرار می دهد.

دو دستور Move Backward و Move To Back شبیه به دو دستور قبلی عمل می کنند با این تفاوت که المان انتخاب شده را به جای حرکت به سمت طبقه‌ی بالاتر، به سمت طبقات پایین تر و در زیر المان های دیگر حرکت می دهند.

از دستورهای مذکور می توانید برای یافتن المان های مخفی شده نیز کمک بگیرید. اگر در صفحه ای یک یا چند المان که با چهارچوب سایه دار محاط شده اند وجود داشته باشد بدین معنی است که این المان ها نه در داخل صفحه بلکه بر روی صفحه قرار دارند. در شکل ۴-۱۰ المان String Control را که در یک چهارچوب سایه دار محاط شده است ملاحظه می کنید. توجه داشته باشید که در این حالت، المان مذکور در داخل حلقه قرار نگرفته است بلکه بر روی آن قرار دارد.



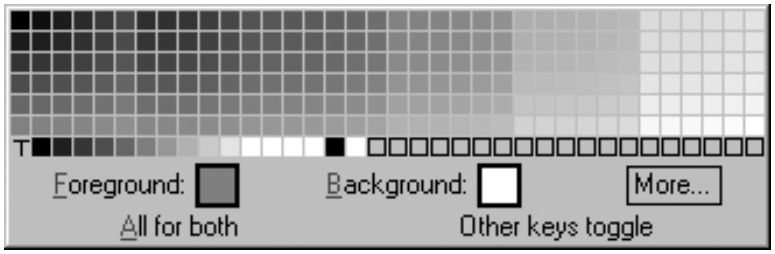

شکل ۴-۱۰

## رنگ آمیزی المان ها

المان های به کار برده شده و همچنین صفحات موجود در LabVIEW بسته به قابلیت صفحه‌ی نمایشگر رایانه به رنگهای سیاه و سفید، خاکستری یا رنگهای متنوع ظاهر می شوند. در محیط برنامه نویسی LabVIEW تغییر رنگ اکثر المان ها (و نه تمام آنها) امکان پذیر نیست. به طور مثال، رنگ ترمینال های قرار گرفته در صفحه‌ی نمودار بلوکی و همچنین رنگهای خاص مربوط به سیمهای ارتباط دهنده‌ی ترمینال ها را که حاوی انواع مختلفی از داده ها هستند نمی توان تغییر داد.

Color Tool

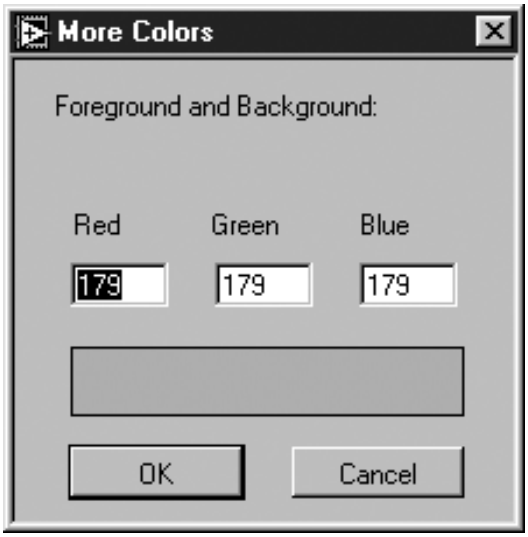
برای تغییر دادن رنگ یک المان و یا پس‌زمینه‌ی آن، منوی کرکره‌ای آن المان یا پس‌زمینه‌ی مربوط را به کمک ابزار Color Tool باز کنید. در این حالت یک پالت رنگ مطابق شکل ۴-۱۱ ظاهر خواهد شد.




شکل ۴-۱۱: پالت رنگ

در حالی که ماوس را بر روی پالت رنگ حرکت می دهید، دکمه‌ی ماوس را فشار دهید. در این حالت المان یا پس‌زمینه‌ی انتخاب شده به همان رنگی که نشانگر ماوس بر روی آن قرار دارد تغییر می یابد. اگر دکمه‌ی ماوس را بر روی یکی از رنگها رها کنید، المان یا پس‌زمینه‌ی مذکور به رنگ انتخاب شده در می آید. برای ملغی کردن عملیات رنگ آمیزی، قبل از رها کردن دکمه‌ی ماوس، نشانگر ماوس را به ناحیه ای خارج از

پالت رنگ حرکت دهید. با انتخاب دکمه‌ی ...More در پالت رنگ، مطابق شکل ۱۲-۴ یک پنجره‌ی محاوره‌ای جهت ایجاد رنگهای سفارشی ظاهر می شود.




**شکل ۱۲-۴ : پنجره‌ی محاوره‌ای More Colors جهت ایجاد رنگهای سفارشی**

برخی از المان‌ها دارای هر دو صفحه‌ی پیش‌زمینه و پس‌زمینه هستند که می‌توانید آنها را به طور جداگانه رنگ‌آمیزی کنید. نشان‌دهنده‌ی موجود در بخش پایینی پالت رنگ مشخص می‌کند که اکنون در حال رنگ‌آمیزی پیش‌زمینه، پس‌زمینه و یا هر دو صفحه هستید. برای رنگ‌آمیزی صفحه‌های پیش‌زمینه یا پس‌زمینه، به ترتیب کلیدهای <F> یا <B> را فشار دهید. جهت رنگ‌آمیزی هر دو صفحه‌ی پیش‌زمینه و پس‌زمینه، کلید <A> را فشار دهید. فشار دادن هر کلید دیگر بر روی صفحه کلید منجر به تغییر وضعیت در انتخاب دو صفحه‌ی پیش‌زمینه و پس‌زمینه می گردد.

به کمک ابزار Set Color در پالت Tools می‌توانید بر روی مربع نشان داده شده در پالت رنگ (Forground یا Background) کلیک کنید و یکی از رنگهای پالت رنگ را انتخاب نمایید. هنگامی که بر روی المان کلیک می کنید، المان مذکور به رنگ انتخاب شده در می آید.

حال منوی کرکره‌ای را بر روی یکی از المان های Digital Control باز نموده، با انتخاب یک رنگ دلخواه، آن را رنگ‌آمیزی کنید. سپس رنگ المان دیگر را با استفاده از ابزارهای موجود در پالت Tools تغییر دهید.

۴

## نمونه‌برداری و کپی کردن رنگها

Color Copy Tool

در برخی موارد، یکسان کردن رنگهای استفاده شده بسیار مشکل است. در نرم‌افزار LabVIEW می توانید رنگ یک المان را به المان دیگر کپی کنید. برای انجام این عمل ابزار Color Copy Tool را از پالت Tools انتخاب کنید. شکل ظاهری این ابزار نظیر «قطره چکان» است. برای کپی برداری، ابتدا بر روی المانی که قصد کپی نمودن رنگ آن را دارید کلیک نموده، سپس به کمک ابزار Color Tool المان های دیگر را رنگ آمیزی کنید.

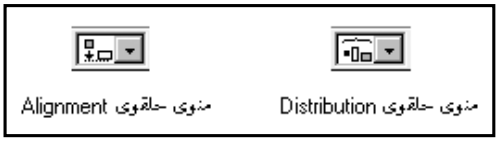
روش دیگر برای کپی برداری از رنگ، فشار دادن دکمه های <Ctrl> و <Option> در دو محیط Windows و Mac و سپس کلیک کردن توسط ابزار Color Tool بر روی المانی است که قصد کپی برداری رنگ آن را دارید. سپس کلیدهای صفحه کلید را رها کرده، به کمک ابزار Color Tool بر روی المان های دیگر کلیک کنید تا به رنگ المان اولیه در آیند.

### شفاف سازی[5]

در هنگام رنگ آمیزی یک المان در صورت انتخاب جعبه‌ای که با حرف T بر روی پالت رنگ مشخص شده است، نرم افزار LabVIEW المان مذکور را شفاف می سازد. این ویژگی که در اصطلاح، رنگ آمیزی فرانمایی نامیده می شود برای لایه بندی[6] المان ها مورد استفاده قرار می گیرد. برای مثال می توانید المان های کنترل غیر قابل رؤیت را در بالای المان های نشان دهنده قرار دهید. ویژگی شفاف سازی تنها بر ظاهر المان تأثیرگذار است. بدین مفهوم که خواص و ویژگیهای استفاده شده در برنامه در مورد آن المان تغییر نمی کند. بنابراین می توانید دستورها و فرمانهای لازم را در مورد المان شفاف شده به کمک ماوس و صفحه کلید اعمال کنید.

## تراز کردن المان ها و تنظیم نمودن فاصله‌ی بین آنها

ممکن است در برنامه هایی که المان های بسیاری در صفحه‌ی آنها وجود دارد، قصد داشته باشید تا المان ها را تراز نموده، فاصله‌ی آنها را تنظیم کنید. در نرم افزار LabVIEW دو منوی حلقوی Alignment و Distribution برای انجام این اعمال تعبیه شده است.




شکل ۱۳-۴

۴

برای تراز کردن المان ها، ابتدا به کمک ابزار Positioning Tool آنها را انتخاب نمایید. بهتر است که به دور المان های مورد نظر یک مستطیل انتخابی ایجاد کنید. سپس یکی از گزینه های موجود در منوی حلقوی Alignment را با توجه به نحوه‌ی تراز کردن مورد نظر، انتخاب نمایید. برای تنظیم فاصله بین المان ها و قرار دادن آنها به صورت متقارن، از منوی حلقوی Distribution استفاده کنید.

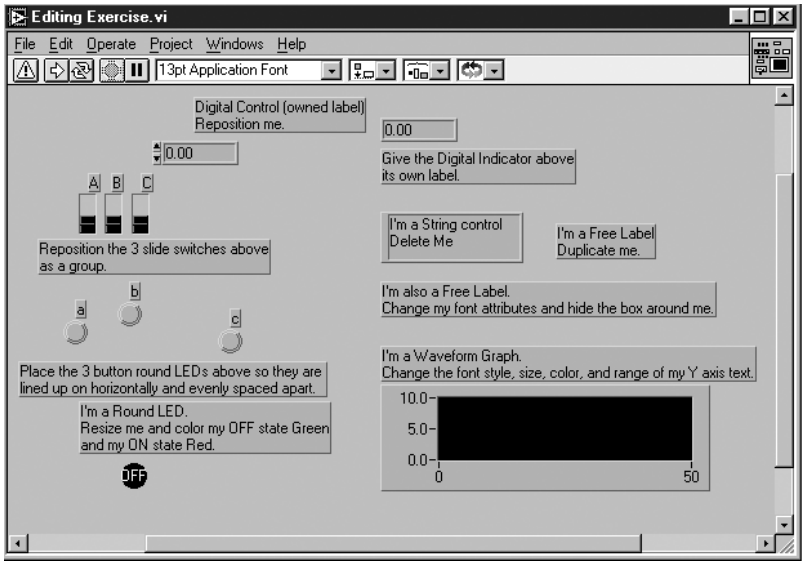
در هنگام به کارگیری این دستورها دقت داشته باشید. زیرا در برخی موارد و پس از اجرای این دستورها ملاحظه می کنید که برخی از المان ها ناپدید شده‌اند. به عنوان مثال، اگر سه دکمه در یک ردیف قرار داشته باشند و آنها را از لبه‌ی سمت چپ (Left Edge) تراز کنید، تمامی آنها بر روی یکدیگر قرار می گیرند. در صورت بروز چنین اتفاقی هر یک از المان ها را با استفاده از ابزار Positioning Tool جابه‌جا کنید.

5- Transparency
6- Layering

## تمرین ۱-۴ : ویرایش صفحه‌ی پانل

در این بخش قصد داریم تا روشهای تغییر و اصلاح المان ها و پالت های موجود در نرم‌افزار LabVIEW را مورد بررسی قرار دهیم. به خاطر داشته باشید که پالت های Controls یا Functions در صورتی قابل رؤیت هستند که به ترتیب، پنجره های پانل یا نمودار بلوکی فعال باشند.

۱- برنامه‌ی Editing Exercise.vi را از مسیر LabVIEW\Exercise\Ch4.llb باز کنید. در صفحه‌ی پانل این برنامه تعدادی المان وجود دارد. هدف ما در این تمرین، تغییر دادن ظاهر این صفحه است.




**شکل ۱۴-۴ : صفحه‌ی پانل در برنامه‌ی Editing Exercise.vi**

Positioning Tool

۲- محل قرار گرفتن المان Digital Control را تغییر دهید. برای انجام این عمل ابتدا ابزار Positioning Tool را از پالت Tools انتخاب کنید. سپس به کمک این ابزار بر روی المان مذکور کلیک کنید و آن را به نقطه‌ی دیگری انتقال دهید. توجه داشته باشید که برچسب مربوط به این المان نیز به همراه آن حرکت می کند. همان گونه که احتمالاً استنباط کرده اید، این برچسب از نوع برچسب اختصاصی است. حال برای اعلام انصراف خود از انتخاب المان، در نقطه ای خالی بر روی صفحه‌ی پانل کلیک کنید. سپس بر روی برچسب این المان کلیک کنید و آن را به نقطه‌ای دیگر انتقال دهید. توجه کنید که در این حالت، تنها برچسب المان به حرکت در می‌آید و المان مورد بحث در جای خود باقی می ماند. بنابراین می توان نتیجه گرفت که جابه‌جا کردن برچسب اختصاصی، بدون حرکت دادن المان امکان‌پذیر نیست. اما در صورت حرکت دادن المان، برچسب آن نیز به همراه المان انتقال می یابد.

۳- اکنون محل قرار گرفتن سه کلید لغزشی یا Slide Switch را به صورت گروهی تغییر دهید. بدین ترتیب که با استفاده از ابزار Positioning Tool در ناحیه ای نزدیک به کلیدها کلیک کنید و در حالی که دکمه‌ی

ماوس را فشار داده‌اید، آن را به حرکت در آورید تا تمامی المان های مذکور در داخل مستطیل انتخابی قرار گیرند. حال بر روی گروه انتخاب شده کلیک نموده، آنها را به نقطه‌ی دیگری انتقال دهید.

۴- المان String Control را با استفاده از ابزار Positioning Tool انتخاب نموده، سپس با فشردن کلید <Delete> یا انتخاب گزینه Edit >> Clear، آن را حذف کنید.

۵- حال برچسب آزاد را کپی کنید. برای انجام این عمل، در دو محیط Windows و Mac به ترتیب کلیدهای <Ctrl> و <Option> را پایین نگه دارید و در همان حالت بر روی برچسب مذکور کلیک کنید، سپس نسخه‌ی کپی آن را به نقطه‌ی دیگری انتقال دهید.

Labeling Tool

۶- مشخصات و ظاهر فونت استفاده شده برای نوشتن متن برچسب آزاد را تغییر دهید. برای ایجاد تغییر در مشخصات فونت، ابتدا به کمک ابزار Labeling Tool متن مذکور را متمایز کنید. برای انجام این عمل یکی از دو روش زیر را انتخاب نمایید:

- بر روی متن مورد نظر دو بار کلیک کنید.
- بر روی متن مذکور یک بار کلیک کنید و ماوس را بر روی متن حرکت دهید.

Color Tool

سپس با استفاده از گزینه های موجود در منوی حلقوی Font مشخصات آن را تغییر دهید. حال جعبه‌ای را که اطراف برچسب قرار گرفته است مخفی کنید. برای انجام این عمل با استفاده از ابزار Color Tool، منوی کرکره‌ای را بر روی جعبه‌ی مذکور باز نموده، از پالت رنگ، گزینه‌ی T را برای شفاف سازی انتخاب کنید.

۷- حال با استفاده از گزینه های موجود در منوی حلقوی Font اندازه، رنگ و ظاهر متن مربوط به محور Y در Waveform Graph را تغییر دهید.

Enter Button

۸- یک برچسب اختصاصی برای المان Digital Indicator تعیین کنید. برای انجام این عمل، ابتدا منوی کرکره‌ای را بر روی این المان باز نموده، سپس گزینه‌ی Label >> Show را از این منو انتخاب کنید. در داخل جعبه‌ی در نظر گرفته شده برای محل برچسب، عبارت Digital Indicator را تایپ کنید. برای وارد کردن متن تایپ شده کلید <Enter> در بخش عددی صفحه کلید را فشار دهید یا بر روی دکمه‌ی Enter در نوار ابزار کلیک کنید.

Positioning Tool

۹- اندازه‌ی LED مدور را تغییر دهید. برای انجام این عمل ابتدا ابزار Positioning Tool را بر روی یکی از گوشه های LED قرار دهید تا نشانگر ماوس به نشانگر تغییر اندازه یا Resizing Cursor تغییر شکل دهد. حال برای تغییر اندازه‌ی LED، بر روی آن کلیک کنید و نشانگر را حرکت دهید تا LED به اندازه‌ی دلخواه در آید. در صورتی که قصد دارید نسبت طول و عرض یا نسبت ابعاد افقی/عمودی LED ثابت باقی بماند، در حالی که کلید <Shift> را پایین نگه داشته‌اید، تغییر اندازه را انجام دهید.

۴

Color Tool

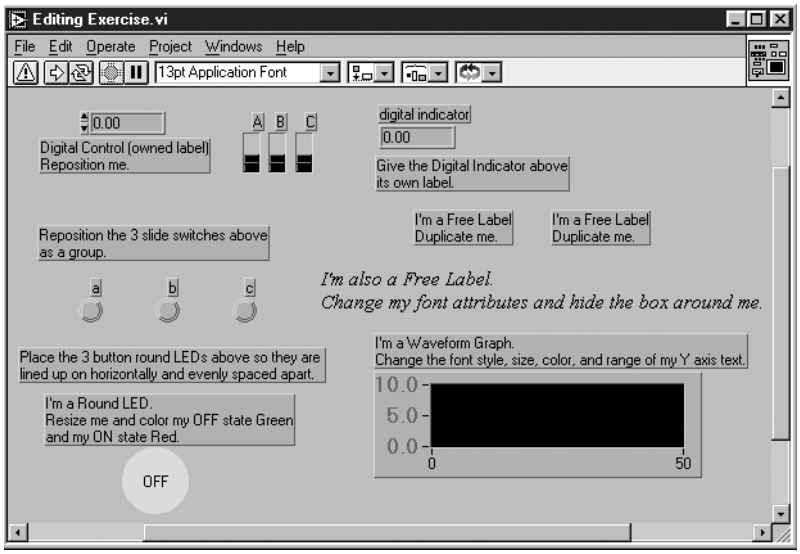
۱۰- رنگ LED مدور را تغییر دهید. برای انجام این عمل ابتدا به کمک ابزار Color Tool منوی کرکره‌ای این المان را باز کنید. سپس در حالی که دکمه‌ی ماوس را فشار داده‌اید، یکی از رنگهای موجود در پالت رنگ را انتخاب نمایید. پس از رها کردن دکمه ی ماوس،آخرین رنگ انتخاب شده از پالت رنگ که ماوس بر روی آن قرار داشته است برای رنگ آمیزی LED انتخاب می شود. حال به کمک ابزار Operating Tool بر روی LED کلیک کنید و وضعیت آن را به حالت ON تغییر دهید. در این حالت نیز رنگ دیگری برای وضعیت ON انتخاب کنید.

Vertical Centers Axis

Horizontal Centers Axis

۱۱- موقعیت سه LED را طوری تنظیم نمایید که هر سه به صورت افقی و به فاصله‌ی مساوی از یکدیگر قرار گیرند. برای انجام این عمل، با استفاده از ابزار Positioning Tool در ناحیه‌ای نزدیک به آنها کلیک کرده، در اطراف آنها یک مستطیل انتخابی محاط کنید. با انتخاب گزینه‌ی Vertical Centers Axis از منوی حلقوی Alignment، آنها را تراز کنید. سپس با انتخاب گزینه‌ی Horizontal Centers Axis از منوی حلقوی Distribution، فاصله المان های مذکور را به طور مساوی تنظیم کنید.

۱۲- پس از اعمال تمامی تغییرات مذکور، ظاهر صفحه‌ی پانل به صورت شکل ۱۵-۴ در می آید.




**شکل ۱۵-۴ : نمای ظاهری صفحه‌ی پانل اصلاح شده در برنامه Editing Exercise.vi**

۱۳- با انتخاب گزینه‌ی File >> Close، پنجره‌ی این برنامه را ببندید. توجه داشته باشید که انتخاب گزینه‌ی File >> Close در صفحه‌ی نمودار بلوکی تنها منجر به بسته شدن صفحه‌ی نمودار بلوکی می گردد. اما انتخاب همین گزینه در صفحه‌ی پانل باعث بسته شدن برنامه می شود. در این حالت از ذخیره نمودن تغییرات انجام شده خودداری کنید.

## المان‌های کنترل و نشان‌دهنده‌ی اصلی

در این بخش قصد داریم تا المان‌های موجود در پالت Controls را مورد بررسی قرار دهیم. در نرم‌افزار LabVIEW، ۴ نوع المان کنترل و نشان‌دهنده‌ی اصلی تعبیه شده است. این المان‌ها عبارتند از: المان‌های عددی، جبری، رشته‌ای و مسیر. در حین نوشتن برنامه ممکن است از انواع پیچیده‌تر داده‌ها نظیر آرایه، کلاستر، جدول، نمودار و گراف استفاده کنید. در آینده در مورد این داده‌ها به تفصیل سخن خواهیم گفت.

بار دیگر خاطر نشان می‌کنیم تنها در صورتی پالت Controls قابل رؤیت است که صفحه‌ی پانل فعال باشد. در غیر این صورت تلاش شما برای یافتن این پالت بیهوده است.

برای اعمال مقادیر عددی یا متنی به المان‌های کنترل یا نشان‌دهنده می‌توان از ابزارهای Operating Tool یا Labeling Tool استفاده نمود. پس از تایپ اعداد یا متن مورد نظر حتماً باید یکی از سه روش وارد کردن داده‌ها یعنی فشردن کلید <Enter> بر روی بخش عددی صفحه کلید یا کلیک کردن برروی دکمه‌ی Enter در نوار ابزار یا کلیک نمودن در نقطه‌ای خالی از صفحه‌ی پانل را جهت وارد کردن اعداد یا متن مذکور انجام دهید. توجه داشته باشید در صورت عدم انجام یکی از اعمال مذکور، اعداد یا متن تایپ شده به المان‌های کنترل یا نشان‌دهنده اعمال نخواهد شد.

فشار دادن کلید <Enter> و <Return> به ترتیب در دو محیط Windows و Mac که در بخش حروف الفبایی قرار گرفته‌اند، منجر به ثبت تغییرات انجام شده نمی‌گردد. بنابراین برای وارد کردن اعداد و متون تایپ شده حتماً از کلیدهای بخش عددی صفحه کلید استفاده کنید. اگر مجبور به استفاده از بخش حروف الفبایی صفحه کلید هستید برای وارد کردن متون و اعداد در دو محیط Windows و Mac به ترتیب کلیدهای <Shift-Enter> و <Shift-Return> را فشار دهید.

۴

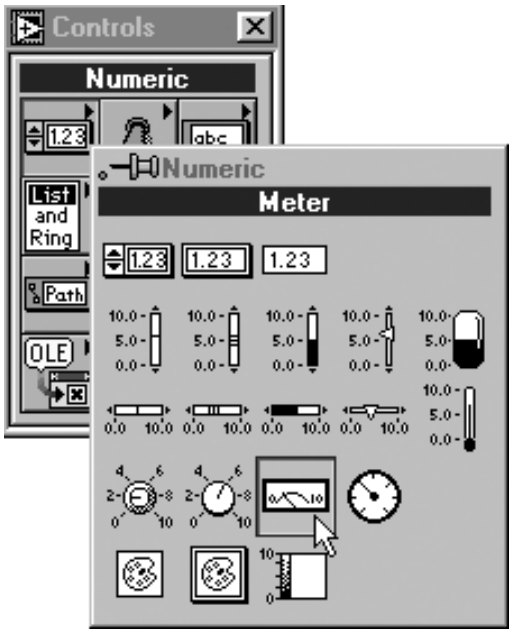
## المان‌های کنترل و نشان‌دهنده‌ی عددی

برای اعمال مقادیر عددی به برنامه، حتماً باید از المان‌های کنترل عددی استفاده کنید. المان‌های نشان‌دهنده‌ی عددی نیز مقادیر عددی حاصل را به نمایش در می‌آورند. در نرم‌افزار LabVIEW تعداد المان‌های عددی بسیار زیاد است. این المان‌ها عبارتند از: کلید چرخشی[۷]، کلید لغزشی[۸]، مخزن[۹]، دماسنج[۱۰] و نشان‌دهنده‌ی دیجیتالی[۱۱].

برای به‌کارگیری این المان‌های عددی، آنها را از زیرپالت Numeric << Controls انتخاب کنید. تمامی

7- Knob
8- Slide
9- Tank
10- Thermometer
11- Digital Display

المان های عددی به هر یک از دو صورت کنترل و نشان دهنده وجود دارند. اما ممکن است برخی از المان ها به صورت پیش فرض به یکی از دو صورت مذکور ظاهر شوند. به طور مثال یک ترمومتر به صورت پیش فرض به صورت نشان دهنده ظاهر می شود. زیرا در اکثر موارد ترمومتر را به عنوان یک عنصر نشان دهنده یا یک المان خروجی استفاده می کنیم. در مقایسه با ترمومتر، یک کلیدچرخشی به صورت یک المان کنترل بر روی صفحه‌ی پانل ظاهر می گردد. زیرا این کلیدها معمولاً عناصر ورودی هستند. در شکل ۱۶-۴ نحوه‌ی انتخاب المان Meter را ملاحظه می کنید.




**شکل ۱۶-۴: نحوه‌ی انتخاب المان Meter**

برای مشاهده حالت دیگر المان مورد نظر، گزینه‌ی Change to Control/Indicator را از منوی کرکره‌ای آن انتخاب نمایید.

۴

## روش نمایش مقادیر عددی

شکل ظاهری ترمینال المان های عددی بر روی صفحه‌ی نمودار بلوکی به نحوه‌ی نمایش داده ها بستگی دارد. نمایش متفاوت داده ها امکان ذخیره‌ی آنها را به روشهای مختلفی فراهم می سازد. با به کارگیری این روشها می توان با کارایی و راندمان بیشتر، از حافظه استفاده نمود.

در روشهای مختلف نمایش مقادیر عددی، مقادیر متفاوتی از فضای حافظه جهت ذخیره‌ی داده ها مورد استفاده قرار می گیرد. در این روشها می توان به عنوان مثال داده ها را در دو فرمت اعداد با علامت (با قابلیت نمایش اعداد منفی) و یا اعداد بدون علامت (با قابلیت نمایش اعداد مثبت و صفر) بررسی نمود. در این روش ترمینال های قرار گرفته در صفحه‌ی نمودار بلوکی برای مقادیر صحیح به رنگ آبی و برای مقادیر اعشاری به

رنگ نارنجی ظاهر می شوند. این ترمینال ها معمولاً شامل چند حرف الفبایی برای ارائه‌ی توضیح مختصری در مورد نوع داده ها هستند. به عنوان مثال ترمینال در نظر گرفته شده برای مقادیر اعشاری با دقت مضاعف، با حروف DBL[12] نشان داده می شود.

در جداول ۴-۱ و ۴-۱ روشهای مختلف نمایش داده ها را ملاحظه می کنید. در این دو جدول همچنین اندازه‌ی داده ها بر حسب بایت و ترمینال به کار برده شده برای طرق مختلف نمایش داده ها نشان داده شده است.

جدول ۴-۱

| Representation | Abbreviation | Terminal | Size (bytes) |
|---|---|---|---|
| Byte | I8 | I8 | 1 |
| Unsigned Byte | U8 | U8 | 1 |
| Word | I16 | I16 | 2 |
| Unsigned Word | U16 | U16 | 2 |
| Long | I32 | I32 | 4 |

جدول ۴-۲

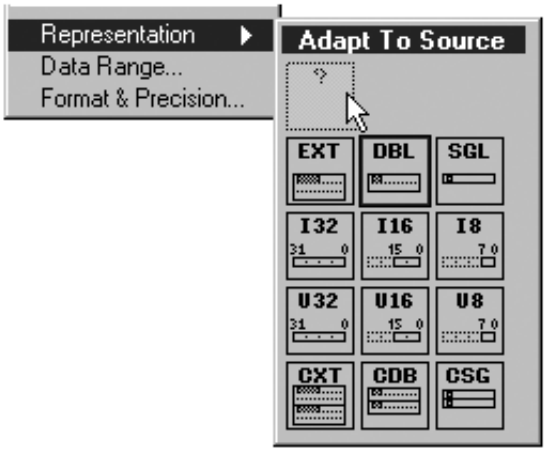
| Data Type | Abbreviation | Representation | Size (bytes) |
|---|---|---|---|
| Unsigned Long | U32 | U32 | 4 |
| Single Precision | SGL | SGL | 4 |
| Double Precision | DBL | DBL | 8 |
| Extended Precision | EXT | EXT | 10(a)/ 12(b) /16(c) |
| Complex Single | CSG | CSG | 8 |
| Complex Double | CDB | CDB | 16 |
| Complex Extended | CXT | CXT | 20(a)/ 24(b)/ 32(c) |

(a) Windows
(b) Macintosh
(c) Sun and HP-UX

۴

پس از باز نمودن منوی کرکره‌ای بر روی المان های ثابت عددی، المان های کنترل و نشان دهنده و انتخاب گزینه‌ی Representation، طرز نمایش اعداد را تغییر دهید. پس از انتخاب گزینه‌ی Representation یک پالت ظاهر می شود که با استفاده از آن می توانید گزینه‌ی مورد نظر را برای نمایش اعداد در المان های ثابت عددی،کنترل و نشان دهنده انتخاب کنید.

12- Double Precision




**شکل ۱۷-۴: گزینه‌های موجود در پالت Representation**

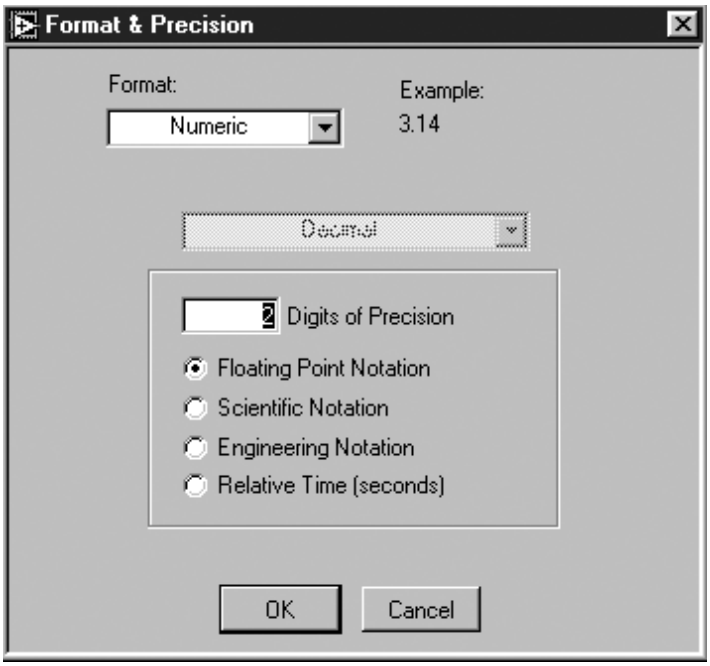
در صورتی که به استفاده‌ی بهینه از حافظه‌ی سیستم خود علاقه‌مند هستید، مطمئناً سعی دارید تا داده‌های خود را به بهترین صورت ممکن نمایش دهید به گونه‌ای که هم اطلاعات مورد نظر از دست نرود و هم از کمترین حجم حافظه جهت ذخیره کردن داده‌ها استفاده شود. دستور Adapt To Source نحوه‌ی نمایش داده‌ها را به صورت خودکار به نشان‌دهنده‌ی استفاده شده نسبت می‌دهد. این دستور، بسیار مفید و کاربردی بوده و سعی کنید استفاده از آن به صورت عادت در آید. در نرم‌افزار LabVIEW همچنین دستورهایی وجود دارد که یک نوع داده را به انواع دیگر تبدیل می‌کند. در مورد این دستورها در دو فصل ۹ و ۱۲ به بحث خواهیم نشست.

## فرمت و دقت نمایش داده‌ها

نرم‌افزار LabVIEW این امکان را برای شما فراهم می‌آورد تا فرمت نشان‌دهنده‌های دیجیتالی را برای نمایش مقادیر عددی یا زمان و تاریخ تعیین کنید. به عنوان مثال در صورتی که قصد داشته باشید تا نشان‌دهنده‌های دیجیتالی را برای نمایش مقادیر عددی مورد استفاده قرار دهید، می‌توانید فرمت نمایش را به یکی از حالات اعشاری[13]، علمی[14] و مهندسی[15] انتخاب نمایید و یا در صورتی که می‌خواهید از نشان‌دهنده‌ی دیجیتالی برای نمایش زمان استفاده کنید می‌توانید در فرمت مورد نظر، «ثانیه» را هم به نمایش در آورید. در ضمن می‌توانید دقت نشان‌دهنده‌ی دیجیتالی را نیز تعیین کنید. منظور از دقت، تعداد ارقام اعشار در سمت راست نقطه‌ی اعشاری است که از 0 تا 20 متغیر می‌باشد. دقت یا تعداد ارقام اعشار، تنها بر روی صفحه‌ی نمایش دهنده‌ی دیجیتالی تأثیرگذار است و دقت واقعی به طرز نمایش اعداد بستگی دارد. برای تعیین فرمت و دقت می‌توانید منوی کرکره‌ای را بر روی المان مورد نظر باز نموده، گزینه‌ی ...Format & Precision را انتخاب کنید. در این مرحله یک پنجره‌ی محاوره‌ای مطابق شکل ۱۸-۴ ظاهر می‌شود.

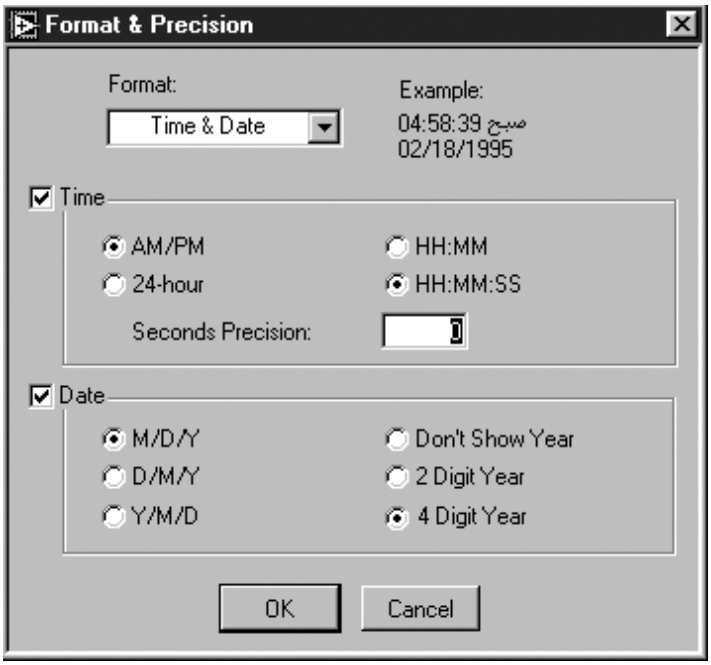
۴

13- Floating Point
14- Scientific
15- Engineering




**شکل ۱۸-۴ : نمای ظاهری پنجره‌ی محاوره‌ای Format & Precision در صورت انتخاب گزینه‌ی Numeric از منوی حلقوی Format**

در صورتی که قصد دارید از نشان‌دهنده دیجیتالی برای نمایش تاریخ و زمان استفاده کنید، گزینه‌ی Time & Date را از منوی حلقوی Format انتخاب نمایید. پس از انتخاب این گزینه، محتویات پنجره‌ی محاوره‌ای نیز مطابق شکل ۱۹-۴ تغییر می‌کند.

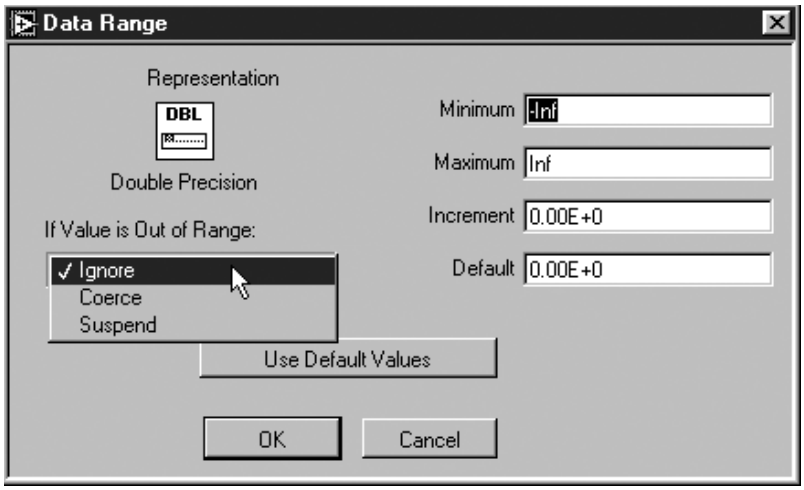



**شکل ۱۹-۴ : نمای ظاهری پنجره‌ی محاوره‌ای Format & Precision در صورت انتخاب گزینه‌ی Time & Date از منوی حلقوی Format**

در برخی موارد تشخیص مقدار واقعی اعداد نشان داده شده توسط المان‌های کنترل و نشان‌دهنده‌ی گرافیکی نظیر کلید چرخشی، گراف و ترمومتر چندان ساده نیست. در این حالت می توانید از نشان دهنده های دیجیتالی استفاده کنید. برای ظاهر ساختن این نشان دهنده، گزینه‌ی Show>>Digital Display را از منوی کرکره‌ای المان مورد نظر انتخاب نمایید. ملاحظه می کنید که یک پنجره‌ی نشان دهنده‌ی دیجیتالی در کنار المان مذکور ظاهر می شود که در آن، مقدار عددی مربوط نشان داده شده است. توجه داشته باشید که نشان دهنده دیجیتالی به عنوان بخشی از المان می‌باشد و فاقد ترمینال جداگانه در صفحه‌ی نمودار بلوکی است.

## تعیین حدود داده‌ها

در نرم‌افزار LabVIEW می توانید مقادیر عددی را در فاصله‌ی معینی محدود کنید. همچنین می توانید فاصله‌ی افزایش داده‌ها را تعیین نمایید. به عنوان مثال قصد دارید که مقادیر ورودی را بین دو عدد 0 و 100 محدود کنید و افزایش اعداد نیز 2 واحد بوده، به طوری که همواره ورودی به صورت یک عدد زوج باشد. برای انجام این عمل منوی کرکره‌ای را بر روی مقدار عددی متناظر باز کرده، گزینه‌ی Data Range... را انتخاب کنید. در این حالت یک پنجره‌ی محاوره‌ای مطابق شکل ۲۰-۴ ظاهر شود.




**شکل ۲۰-۴: پنجره‌ی محاوره‌ای Data Range**

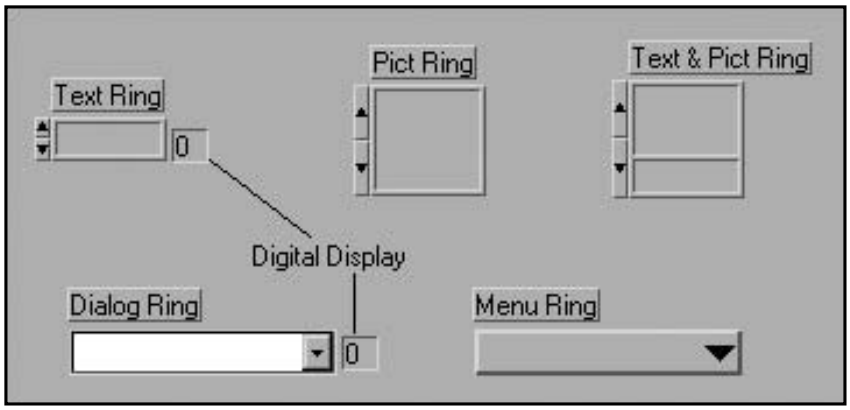
در پنجره‌ی محاوره‌ای فوق می توانید پارامترهایی نظیر نحوه‌ی نمایش داده‌ها[16]، حداکثر[17]، حداقل[18] و فاصله‌ی افزایش داده‌ها[19] را تعیین کنید. در این پنجره همچنین می توانید مقدار پیش‌فرض المان را توسط گزینه‌ی Default تغییر دهید. در ضمن در صورتی که مقدار ورودی، خارج از محدوده‌ی تعیین شده باشد می توانید با انتخاب یکی از سه گزینه‌ی نشان داده شده در شکل ۲۰-۴ عملکرد برنامه را در ارتباط با عدد مذکور تغییر دهید:

16- Representation
17- Maximum
18- Minimum
19- Increment

- در صورت انتخاب گزینه‌ی Ignore، مقادیر خارج از محدوده بدون تغییر باقی می مانند. با کلیک کردن بر روی پیکان های افزایش دهنده یا کاهش دهنده که در کنار المان کنترل واقع شده‌اند، مقدار عددی المان به اندازه‌ی فاصله‌ی افزایشی که توسط کاربر تعیین شده است، افزایش یا کاهش می‌یابد تا به حداکثر یا حداقل مقدار خود برسد. در صورت انتخاب گزینه‌ی مذکور می توانید اعداد خارج از محدوده‌ی مشخص شده را در این قسمت وارد کنید.
- در صورت انتخاب گزینه‌ی Coerce، تمامی مقادیر کمتر از حد می نیمم و یا بیشتر از حد ماکزیمم به ترتیب به مقادیر حداقل و حداکثر در نظر گرفته شده گرد می شوند. در صورتی که در این حالت مقدار افزایشی نادرست باشد، این مقدار نیز گرد خواهد شد.
- در صورت انتخاب گزینه‌ی Suspend و در صورتی که داده‌ها معتبر و قابل قبول نباشند، LabVIEW اجرای برنامه را آغاز نمی‌کند و یا اجرای آن را متوقف می سازد. در این حالت المان‌های کنترلِ فاقد اعتبار توسط حلقه‌ای قرمزرنگ احاطه شده، دکمه‌ی Run نیز در زیر یک دایره‌ی قرمزرنگ پنهان می‌شود. درون دایره‌ی مذکور یک خط مورب به صورت « / » قرار دارد. توجه داشته باشید که این علامت شبیه به «تابلوی ورود ممنوع» در علائم راهنمایی و رانندگی می‌باشد و بدین مفهوم است که در حال حاضر اجرا نمودن این برنامه امکان‌پذیر نیست. پس از اعمال داده‌های صحیح و معتبر به المان‌های کنترل، دایره‌های قرمزرنگ ناپدید می گردند.

## منوی حلقوی

Ringها یا منوهای حلقوی، المان‌های عددی خاصی هستند که به همراه اعداد صحیح بدون علامت ۱۶ بیتی با رشته‌ها، تصاویر یا هر دو ظاهر می شوند. این المان‌ها در زیرپالت Controls >> List &Ring قرار دارند. المان‌های مذکور جهت انتخاب گزینه‌های انحصاری نظیر مدهای عملیاتی، دستورهای محاسباتی و... بسیار مفید می باشند. در شکل ۲۱-۴ چند منوی حلقوی را ملاحظه می کنید.




شکل ۲۱-۴

پس از ایجاد منوی حلقوی، قدم بعدی وارد کردن یک متن یا الحاق نمودن یک تصویر به منوی حلقوی است. در این مرحله به هر یک از متون یا تصاویر، عدد معینی نسبت داده می‌شود (عدد 0 برای اولین متن یا تصویر، عدد 1 برای متن یا تصویر بعدی و...). برای دستیابی و مشاهده‌ی اعداد مذکور، گزینه‌ی Show >> Digital Display را از منوی کرکره‌ای منوی حلقوی انتخاب نمایید. در شکل ۲۱-۴ بخش Digital Display نشان داده شده است.

یک منوی حلقوی جدید تنها شامل یک المان با مقدار صفر و نشان‌دهنده‌ی فاقد المان می‌باشد. در صورتی که قصد دارید تا اعداد و متناظر با آن اعداد، پیام‌ها و متون دیگری را اضافه کنید یکی از دو گزینه‌ی Add Item After یا Add Item Before را از منوی کرکره‌ای آن انتخاب کنید. در این حالت یک پنجره برای وارد کردن اقلام مذکور ظاهر می‌گردد. در این پنجره می‌توانید به کمک ابزار Labeling Tool متن مورد نظر را تایپ نمایید یا یک تصویر را وارد کنید.

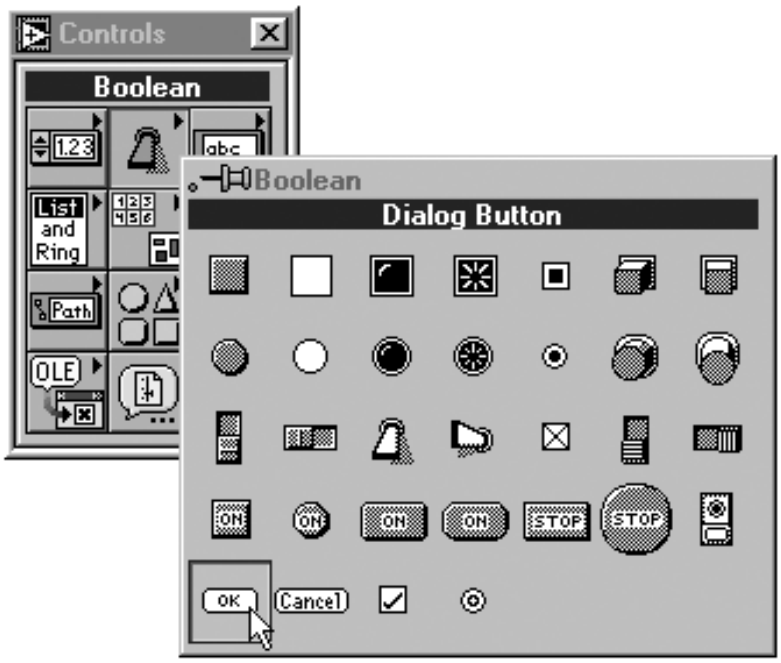
در صورتی که به کمک ابزار Operating Tool بر روی یک منوی حلقوی کلیک کنید، تمامی پیغام‌ها یا تصاویر موجود در آن به نمایش در می‌آیند. استفاده از منوهای حلقوی در مواردی که قصد داشته باشید گزینه‌ای متناظر با مقدار عددی موجود در صفحه‌ی نمودار بلوکی انتخاب نمایید بسیار مفید است.

حال که با این المان آشنا شدید، یک منوی حلقوی بر روی صفحه‌ی پانل ایجاد نموده، سپس بخش نشان‌دهنده‌ی دیجیتالی آن را ظاهر سازید و چند المان دیگر نیز به آن اضافه کنید.

## مقادیر جبری

Booleans نامی است که به احترام «جرج بول» ریاضی‌دان و منطق‌دان انگلیسی که تلاش‌ها و مطالعات او موجب پیدایش منطق جبر بول شده انتخاب شده است. برای اهداف برنامه نویسی می‌توانید این عبارت را تنها به عنوان یک واژه‌ی فرضی برای دو حالت on و off، روشن و خاموش، درست و نادرست یا 1 و 0 در نظر بگیرید. زیرا همان گونه که می‌دانید عبارات جبری دارای دو حالت درست یا نادرست یا 1 و 0 می‌باشند.

در نرم‌افزار LabVIEW تعداد زیادی کلید، LED و دکمه برای نشان دادن المان‌های کنترل و نشان‌دهنده‌ی جبری تعبیه شده است که همگی آنها در زیرپالت Controls >> Boolean قرار دارند. برای تغییر حالت یک المان جبری کافی است توسط ابزار Operating Tool بر روی آن کلیک کنید. همان گونه که در مورد المان‌های کنترل و نشان‌دهنده‌ی عددی توضیح داده شد، هر یک از المان‌های جبری نیز به صورت پیش‌فرض و طبق کاربرد و موارد استفاده آن به یکی از دو حالت کنترل یا نشان‌دهنده ظاهر می‌شود. به عنوان مثال کلیدها به صورت المان‌های کنترل و LEDها به صورت المان‌های نشان‌دهنده ظاهر می‌شوند. در صورت نیاز و برای تبدیل المان کنترل به نشان‌دهنده و یا برعکس، گزینه‌ی Change to Control/Indiacator را از منوی کرکره‌ای آن المان انتخاب کنید. در شکل ۲۲-۴ نحوه‌ی انتخاب المان جبری Dialog Button را ملاحظه می‌کنید.




**شکل ۲۲-۴ : نحوه انتخاب المان جبری Dialog Button**

Boolean Terminal

ترمینال‌های مربوط به المان‌های جبری بر روی صفحه‌ی نمودار بلوکی به رنگ سبز ظاهر می‌شوند و شامل دو حرف F و T می‌باشند. حرف T نشان‌دهنده حالت درست یا 1 بوده، در حالی که حرف F بیانگر حالت نادرست یا 0 است.

بار دیگر توجه شما را به این نکته جلب می‌کنیم که حاشیه‌ی ترمینال‌های کنترل پهن بوده، در حالی که حاشیه‌ی ترمینال‌های نشان‌دهنده باریک است. به دلیل اینکه این دو از لحاظ عملیاتی یکسان نیستند تشخیص و شناخت آنها حائز اهمیت است و باز یادآوری می‌کنیم که المان‌های کنترل و نشان‌دهنده به ترتیب به عنوان ورودی و خروجی تلقی می‌شوند.

۴

در ادامه به بررسی برخی از المان‌های جبری و قابلیتهای آنها می‌پردازیم.

## دکمه‌های برچسب‌دار[20]

در نرم‌افزار LabVIEW دو نوع دکمه‌ی برچسب‌دار وجود دارد. این دو دکمه به صورت مدور و چهارگوش می‌باشند. برچسبها یا پیامهای متنی که به همراه این دکمه‌ها ظاهر می‌شوند برای کاربر بسیار مفید هستند.

برای هر یک از دکمه‌های برچسب‌دار می‌توان دو پیام متنی تعیین نمود. یک پیام برای حالت True و دیگری برای حالت False. هنگامی که دکمه‌ای را برای اولین بار بر روی صفحه‌ی پانل قرار می‌دهید، به

20- Labeled Buttons

صورت پیش فرض، وضعیت True به منزله روشن بودن دکمه و وضعیت False به مفهوم خاموش بودن آن در نظر گرفته می شود. در صورت نیاز می توانید به کمک ابزار Labeling Tool هریک از پیام ها و متون نوشتاری را تغییر دهید.

در منوی کرکره ای تمامی المان های جبری، گزینه‌ی Boolean Text >> Show وجود دارد. به کمک این گزینه و با توجه به وضعیت دکمه می توان یکی از عبارات OFF یا ON را نمایش داد.

## حرکت مکانیکی کلیدهای جبری[۲۱]

در منوی کرکره‌ای هر یک از المان های کنترل جبری گزینه‌ی بسیار مفید و کمکی Mechanical Action به چشم می خورد. این گزینه به شما اجازه می دهد تا نحوه‌ی عملکرد و رفتار المان جبری را در هنگام کلیک کردن بر روی آن تعیین کنید. به عنوان مثال می توانید مشخص کنید که تغییر وضعیت دکمه در هنگام فشار دادن آن و یا در هنگام رها کردن آن انجام گیرد. در فصل ۸ در مورد حرکت مکانیکی کلیدهای جبری به طور مفصل به بحث خواهیم پرداخت.

## حدود داده ها در المان های جبری

اگر قصد دارید نادرست بودن یک مقدار جبری را مشخص و آشکار سازید، زیرمنوی Data Range را از منوی کرکره‌ای المان جبری مورد نظر باز نموده، سپس یکی از دو گزینه‌ی Suspend if True یا Suspend if False را انتخاب کنید. همان گونه که در مورد داده های عددی و در مورد گزینه‌ی Suspend ذکر شد، در صورت ظاهر شدن مقادیر دور از انتظار اجرای برنامه متوقف می گردد.

## سفارشی نمودن المان های جبری به کمک تصاویر وارد شده

در LabVIEW انتساب المان های جبری کنترل و نشان دهنده برای دو حالت True و False امکان پذیر است. این عمل توسط وارد کردن تصاویر انجام می گیرد. در فصل ۱۵ در مورد این ویژگی کاربردی به تفصیل توضیح خواهیم داد.

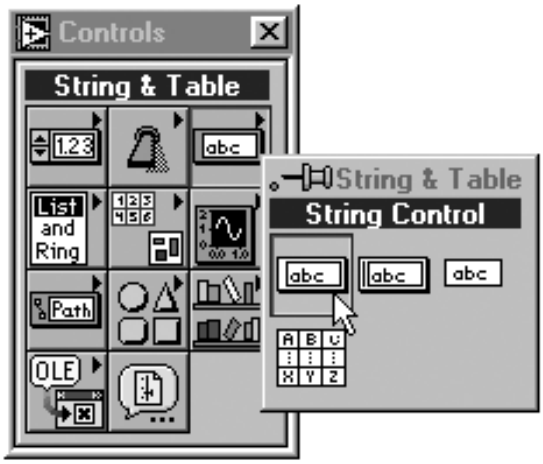


## المان های رشته ای

به بیان ساده می توان گفت که المان های کنترل و نشان دهنده‌ی رشته ای برای نمایش داده های متنی مورد استفاده قرار می گیرند. داده های رشته ای معمولاً به صورت کدهای ASCII می باشند. همان گونه که می دانید کد ASCII روشی استاندارد جهت ذخیره نمودن کاراکترهای الفبایی، عددی و... است. ترمینال های رشته ای و

21- Mechanical Action

سیمهایی که حامل داده‌های رشته‌ای هستند بر روی صفحه‌ی نمودار بلوکی به رنگ صورتی ظاهر می‌شوند. ترمینال‌های رشته‌ای در صفحه‌ی نمودار بلوکی حاوی سه حرف «abc» هستند. برای دستیابی به المان‌های رشته‌ای، زیرپالت String & Table << Controls را مطابق شکل ۲۳-۴ باز نموده، المان مورد نظر را انتخاب کنید.




**شکل ۲۳-۴: نحوه‌ی انتخاب المان String Control**

اگرچه المان‌های کنترل و نشان‌دهنده‌ی رشته‌ای می‌توانند حاوی کاراکترهای عددی باشند، اما برای ذخیره نمودن کاراکترهای عددی از این المان‌ها استفاده نمی‌شود. زیرا پردازش و انجام عملیات عددی بر روی داده‌های رشته‌ای امکان‌ناپذیر است. در صورت نیاز به داده‌های عددی که به صورت رشته‌ای ذخیره شده‌اند، ابتدا باید آنها را به کمک دستورهایی که در فصل ۹ اشاره خواهد شد به فرم عددی تبدیل کنید.

درک عملکرد المان‌های کنترل و نشان‌دهنده‌ی رشته‌ای نسبتاً ساده است و در منوی کرکره‌ای آنها تنها چند گزینه‌ی خاص وجود دارد. در مورد رشته‌ها و جداول در فصل ۹ به تفصیل سخن خواهیم گفت.

۴

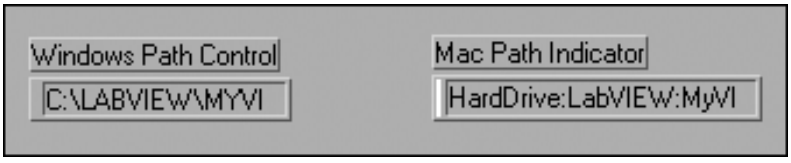
### مسیر

مسیر نوع خاصی از داده‌هاست که به صورت مستقل و غیروابسته به نوع سیستم برای نشان دادن محل ذخیره شدن فایل‌ها و فهرست‌ها به‌کار برده می‌شود. عبارت « مستقل و غیروابسته به نوع سیستم[۲۲] » بدین معنی است که محل قرار گرفتن یک فایل یا فهرست در سیستم‌های مختلف از قبیل Windows، Mac و... به نوع سیستم وابسته نیست.

در صفحه‌ی نمودار بلوکی، سیمها و ترمینال‌های مربوط به مسیر به رنگ سبز متمایل به آبی نشان داده

22- Platform

می‌شوند. همان گونه که می‌دانید جهت نمایش دادن مسیر، ابتدا نام درایو و سپس نام فهرست اصلی و در ادامه نام زیرفهرست (در صورت وجود) و در انتها نام فایل نوشته می‌شود. در دو محیط Windows و Mac نام درایو، فهرست و فایل، به ترتیب توسط دو کاراکتر «\» و «:» جدا می‌شوند. در فصل ۱۴ در مورد مسیرها توضیحات بیشتری ارائه خواهیم داد. در شکل ۲۴-۴ روش نمایش مسیر در دو محیط Windows و Mac را ملاحظه می‌کنید.




**شکل ۲۴-۴: نحوه‌ی نمایش مسیر در دو محیط Windows و Mac**

## زیباسازی صفحه‌ی پانل

تنها برای سرگرمی و زیباسازی ظاهر صفحه‌ی پانل می‌توانید از زیرپالت Decorations << Controls استفاده کنید. این زیرپالت، تنها شامل دستورها و توابع مخصوص برای زیباتر کردن صفحه‌ی پانل است. المان‌های موجود در این زیرپالت تنها عناصر موجود در پالت Controls هستند که فاقد ترمینال‌های متناظر بر روی صفحه‌ی نمودار بلوکی می‌باشند.

## ایجاد المان‌های کنترل و نشان‌دهنده‌ی سفارشی

برای هرچه جالب‌تر شدن محیط برنامه نویسی LabVIEW، می‌توان المان‌های کنترل و نشان‌دهنده‌ی مورد نظر را در محیطهای گرافیکی دیگر نظیر Paint طراحی نموده، سپس آنها را در صفحه‌ی پانل وارد کرد. بنابراین در صورتی که المان مورد نظر و مطلوب شما در محیط LabVIEW وجود ندارد، می‌توانید آن المان را ایجاد کنید. در فصل ۱۵ نحوه‌ی انجام این عمل را مورد بررسی قرار خواهیم داد.

۴

## خلاصه‌ای در مورد المان‌های کنترل و نشان‌دهنده‌ی اصلی

در نرم‌افزار LabVIEW، ۴ نوع المان کنترل و نشان‌دهنده‌ی اصلی وجود دارد:

- المان‌های عددی: این عناصر حاوی مقادیر عددی استاندارد هستند.
- المان‌های جبری: این المان‌ها تنها می‌توانند در یکی از دو وضعیت on یاoff، True یاFalse و 1 یا 0 قرار گیرند.
- المان‌های رشته‌ای: این المان‌ها حاوی داده‌های متنی می‌باشند. اگرچه می‌توان برای ذخیره کردن کاراکترهای عددی از این عناصر استفاده نمود، ولی برای انجام هرگونه عملیات و محاسبه بر روی این داده‌ها ابتدا باید داده‌های رشته‌ای را به داده‌های عددی تبدیل کرد. جهت ذخیره کردن کاراکترهای متنی از

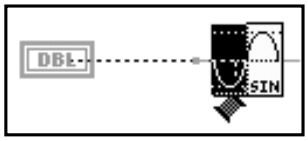
کدهای ASCII استفاده می شود.

- مسیر: محل ذخیره شدن فایل ها را نشان می دهد.

## سیم کشی و برقرار نمودن ارتباط

اگر صفحه‌ی پانل یک برنامه مملو از المان های کنترل و نشان دهنده باشد و این المان ها به طور منظم و با ترتیب خاصی آراسته شده باشند، تا زمانی که ترمینال های متناظر با این المان ها را در صفحه‌ی نمودار بلوکی به یکدیگر متصل نکنید، این برنامه اجراء نخواهد شد. در این بخش سعی داریم تا تمامی اطلاعات لازم در مورد نحوه‌ی سیم کشی و برقراری ارتباط را در اختیار شما قرار دهیم.

جهت برقراری ارتباط بین ترمینال ها، از ابزار Wiring Tool استفاده می شود. در این حالت نشانگر ماوس به صورت یک قرقره ظاهر می گردد. برای سیم کشی از یک ترمینال به ترمینال دیگر، توسط ابزار Wiring Tool بر روی اولین ترمینال کلیک کنید و این ابزار را به سمت ترمینال دوم حرکت دهید؛ سپس بر روی ترمینال دوم کلیک کنید. در این حالت هیچ اهمیتی ندارد که ابتدا بر روی کدام ترمینال کلیک می کنید. زمانی که انتهای آزاد قرقره دقیقاً بر روی ترمینال قرار می گیرد، ترمینال مذکور چشمک می زند. در این حالت، کلیک کردن ماوس باعث اتصال سیم به ترمینال می گردد. در شکل ۲۵-۴ نحوه‌ی سیم کشی بین ترمینال DBL و ترمینال ورودی آیکن دستور Sine نشان داده شده است.

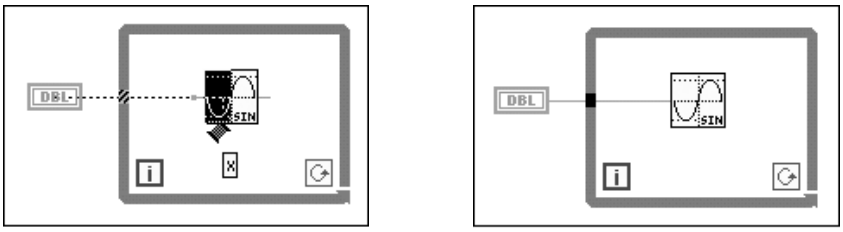



**شکل ۲۵-۴: نحوه‌ی سیم کشی بین ترمینال DBL و ترمینال ورودی آیکن دستور Sine**

به محض برقرار نمودن اولین اتصال، همزمان با حرکت دادن ماوس، یک قطعه سیم بر روی صفحه‌ی نمودار بلوکی به نمایش در می آید. در این حالت لازم نیست تا دکمه‌ی ماوس را فشرده نگه دارید و حرکت کنید. تنها کلیک نمودن اولیه‌ی ماوس بر روی ترمینال کافی است تا سیم به ترمینال مورد نظر متصل شود. در این حالت می توانید دست را از روی دکمه‌ی ماوس بردارید و آن را حرکت دهید. برای امتداد دادن یک قطعه سیم می توانید همین عمل را انجام دهید. در این حالت هنگامی که انتهای آزاد قرقره به طورکامل بر روی قطعه سیم قرار می گیرد، سیم مذکور چشمک می زند. این عمل در حقیقت گره زدن یا لحیم کردن انتهای آزاد قرقره به قطعه سیم موجود می باشد.

در ضمن می توانید از ترمینالی که خارج از حلقه قرار دارد به یک ترمینال داخلی حلقه سیم کشی کنید. در فصل ۶ در مورد این روش سیم کشی توضیح خواهیم داد. همان گونه که در شکل ۲۶-۴ ملاحظه می کنید در نقطه ای از چهارچوب حلقه که محل عبور سیم به داخل حلقه است یک تونل[۲۳] ایجاد می شود.

23- Tunnel



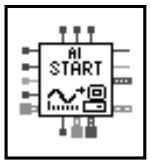

شکل ظاهری تونل پس از انجام سیم‌کشی (ب)

شکل ظاهری تونل در حین انجام سیم‌کشی (الف)

**شکل ۲۶-۴**

## سیم‌کشی المان‌های پیچیده

هنگامی که قصد دارید یک گره‌ی پیچیده یا آیکن یک زیربرنامه را سیم‌کشی کنید، توجه به سرسیمها و توضیحات مختصری که با نزدیک شدن ابزار Wiring Tool به آیکن ظاهر می‌شوند، می‌تواند شما را در انجام عملیات سیم‌کشی یاری دهد. رنگ، ضخامت و شکل سرسیمهایی که در اطراف آیکن زیربرنامه وجود دارند مشخص کننده‌ی نوع داده‌ی مورد نیاز در این ترمینال‌هاست. سرسیمهایی که برای اعمال ورودیها در نظر گرفته شده‌اند با نقطه‌ای که در انتهای آنها قرار دارد متمایز می‌گردند. سرسیمهایی که در انتهای آنها نقطه وجود ندارد محل اتصال خروجیها هستند. اگر به رعایت نظم و ترتیب در صفحه‌ی نمودار بلوکی برنامه علاقه‌مند هستید، حتماً سیم‌کشی را در جهات پیشنهاد شده و طبق ترتیب قرارگرفتن سرسیمها برروی آیکن مربوط انجام دهید. در شکل ۲۷-۴ یک آیکن به همراه سرسیمهای ورودی و خروجی آن را ملاحظه می‌کنید. به نقطه‌ای که در انتهای سرسیمهای ورودی وجود دارد توجه کنید.




**شکل ۲۷-۴ : نمای ظاهری آیکن یک دستور پیچیده که سرسیمهای ورودی و خروجی آن با نزدیک شدن ابزار Wiring Tool به آیکن آن متمایز شده‌اند.**

با به‌کارگیری پنجره‌ی Help می‌توان هر یک از ترمینال‌های کانکتور را متمایز نمود. هنگامی که ابزار Wiring Tool را بر روی ترمینال قرار می‌دهید، ترمینال قرار گرفته در پنجره‌ی Help نیز چشمک می‌زند. بدین ترتیب می‌توانید از اتصال سیم به محل دقیق آن اطمینان حاصل کنید. همچنین به کمک پنجره‌ی Help می‌توانید نوع اتصالات هر دستور را از لحاظ الزامی، پیشنهادی یا اختیاری دریابید.

## سیم‌کشی نادرست

زمانی که در حین سیم‌کشی دچار اشتباه می‌شوید به جای یک قطعه سیم رنگی، یک خط نقطه‌چین سیاه‌رنگ که در اصطلاح به آن «سیم منقطع [۲۴]» گوییم ظاهر می‌شود. تا زمانی که تمامی سیمهای منقطع را

24- Broken Wire

از بین نبرده‌اید، دکمه‌ی Run به صورت «پیکان شکسته» ظاهر می‌شود و برنامه‌ی مورد نظر کامپایل نخواهد شد. برای از بین بردن قطعه سیم منقطع، ابتدا آن را انتخاب نموده، سپس آن را حذف کنید. برای حذف تمامی سیمهای منقطع که در صفحه‌ی نمودار بلوکی وجود دارند، گزینه‌ی Edit >> Remove Bad Wires را انتخاب نمایید. برای انجام این عمل به کمک صفحه کلید، از ترکیب کلیدهای<Ctrl-B> و<Cmd-B> به ترتیب در دو محیط Windows و Mac استفاده کنید. نیز

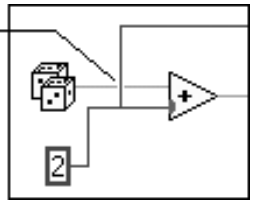
در برخی موارد، سیمهای منقطع به صورت تکه‌تکه بوده، یا در زیر یکی از المان‌ها قرار می‌گیرند یا به قدری کوچک هستند که به راحتی دیده نمی‌شوند. در این حالت بهترین روش برای حذف این سیمها، انتخاب گزینه‌ی Edit >> Remove Bad Wires است.

برای کسب اطلاع در مورد علت شکستگی سیم، بر روی دکمه‌ی Run که در این حالت نیز به صورت پیکان شکسته ظاهر می‌شود کلیک نمایید یا منوی کرکره‌ای را بر روی سیم منقطع باز کنید. پس از کلیک کردن بر روی دکمه‌ی Run یا انتخاب گزینه‌ی List Errors از منوی کرکره‌ای سیم منقطع، یک پنجره‌ی محاوره‌ای باز شده، در آن علت شکستگی سیم توضیح داده می‌شود.

## مطالبی در مورد روش سیم کشی

رعایت موارد اشاره شده در لیست زیر، شما را در انجام سیم کشی سریع و آسان یاری می دهد:

- می توانید تنها یک بار بدون کلیک کردن، سیم را تا ۹۰° بچرخانید.
- با کلیک کردن ماوس می توانید انتهای سیم را متصل یا ثابت کرده، سپس جهت امتداد سیم را تغییر دهید.
- با فشار دادن کلید <Space bar> می توانید جهتی را که سیم از یک نقطه خارج می شود عوض کنید.
- برای تعیین نقطه‌ی آغاز یا پایان سیم می توانید در ناحیه‌ای خالی از صفحه، دو مرتبه کلیک کنید.
- در مواردی که دو سیم همدیگر را قطع می کنند یک شکاف در قطعه سیمی که در ابتدا رسم شده است ظاهر می شود. این مطلب در شکل ۲۸-۴ نشان داده شده است. دقت داشته باشید که در این شکل ابتدا المان Random Number (0-1) به دستور Add سیم کشی شده و در مرحله بعدی سیم کشی قطعه‌ی دیگر انجام گرفته است، بنابراین به هنگام عبور دو قطعه سیم از روی یکدیگر، بر روی قطعه سیم اول یک شکاف برای عبور قطعه سیم دوم ایجاد می گردد.




شکل ۲۸-۴

۴

- برای حذف یک قطعه سیم در حین انجام سیم کشی، با استفاده از دکمه‌ی سمت راست ماوس بر روی آن کلیک کنید. در محیط Mac این عمل با فشار دادن کلید <Cmd> و سپس کلیک کردن بر روی سیم مورد نظر امکان پذیر است.
- به کمک پنجره‌ی Help می توان اطلاعات بیشتری در مورد المان و نحوه‌ی سیم کشی صحیح آن به دست آورد.

حال که با برخی از تکنیک های سیم کشی آشنا شدید، دو المان Numeric Control بر روی صفحه‌ی پانل ایجاد نموده، سپس آنها را به ورودیهای آیکن دستور Add سیم کشی کنید. به دلیل اینکه هنوز مبحث سیم کشی خروجیها را مورد بررسی قرار نداده ایم، از سیم کشی خروجی آیکن دستور Add خودداری کنید. توجه داشته باشید که سیم کشی های ذکر شده را باید در صفحه‌ی نمودار بلوکی انجام دهید.

## امتداد دادن سیم

در نرم افزار LabVIEW می توان المان های سیم کشی شده را جداگانه یا به صورت گروهی حرکت داد. برای حرکت دادن سیمها ابتدا آنها را به کمک ابزار Positioning Tool انتخاب نموده، سپس آنها را به محل جدید انتقال دهید. سیمهای متصل به المان های انتخاب شده در صورت حرکت و جابه‌جا شدن المان ها، به صورت خودکار امتداد می یابند. در صورت کپی برداری از المان ها و یا انتقال آنها از یک صفحه به صفحه یا زیرنمودار [۲۵] دیگر (به عنوان مثال انتقال از صفحه‌ی نمودار بلوکی به داخل یک حلقه‌ی While) سیمهای متصل شده به المان ها در همان محل اولیه باقی می مانند مگر اینکه آنها را نیز انتخاب کرده باشید.

در برخی موارد، امتداد دادن سیمها باعث به وجود آمدن سرسیمهای کوتاه و منقطع و یا سیمهای بدون انتها می گردد. قبل از اجرای برنامه حتماً با استفاده از گزینه‌ی Edit >> Remove Bad Wires و یا به کمک صفحه کلید تمامی سیمهای مذکور را حذف کنید.

حال به کمک ابزار Positioning Tool آیکن دستور Add را حرکت دهید و نحوه‌ی امتداد یافتن سیمها را ملاحظه کنید.

## انتخاب و حذف نمودن سیمها

قبل از وارد شدن به این مبحث، به ارائه‌ی توضیح در مورد چند واژه می پردازیم:

یک قطعه سیم[۲۶] تکه سیمی است که به صورت افقی یا عمودی قرار گرفته است. نقطه ای که ۳ یا ۴ قطعه سیم به یکدیگر اتصال می یابند، نقطه‌ی اتصال[۲۷] نامیده می شود. نقطه‌ی پیچیدگی یا زانو[۲۸] محلی است که دو قطعه سیم به یکدیگر اتصال می یابند. شاخه‌ی سیم[۲۹] شامل تمامی قطعه سیمهای موجود از یک نقطه‌ی اتصال تا نقطه‌ی اتصال دیگر، از ترمینال به نقطه‌ی اتصال یا از ترمینال به ترمینال دیگر می باشد. البته به شرطی که در این میان هیچ نقطه‌ی اتصال دیگری وجود نداشته باشد.

25- Subdiagram
26- Wire Segment
27- Wire Junction
28- Bend
29- Wire Branch

Positioning Tool

برای انتخاب یک قطعه سیم، توسط ابزار Positioning Tool بر روی آن کلیک کنید. دو بار کلیک کردن منجر به انتخاب شدن یک شاخه سیم می گردد. سه بار کلیک کردن موجب انتخاب یک سیم کامل می شود. برای حذف قطعه سیم انتخاب شده در محیط Windows یا Mac، به ترتیب کلید <Delete> یا <Back space> را فشار دهید.

اکنون که با روشهای انتخاب و حذف سیمها آشنا شدید یکی از قطعه سیمهای موجود در صفحه را انتخاب نموده، سپس آن را حذف کنید. حال مجدداً آن را سیم کشی کنید.

## حرکت دادن سیمها

Positioning Tool

در نرم افزار LabVIEW می توان یک یا چند قطعه سیم را جابه‌جا نمود. برای انجام این عمل ابتدا سیمها را به کمک ابزار Positioning Tool انتخاب نموده، سپس آنها را حرکت دهید. همچنین برای حرکت در فواصل کوتاه و دقیق می توانید از کلیدهای مکان نما استفاده کنید. بدین ترتیب با فشار دادن هر یک از کلیدهای مکان نما، قطعات انتخاب شده در هر ضربه تنها به اندازه‌ی یک نقطه[3] حرکت می کنند. در جابه‌جا کردن سیمها، LabVIEW قطعه سیمهای انتخاب شده‌ی مجاور را نیز برای منطبق ساختن با تغییرات انجام شده امتداد می دهد. به هنگام حرکت دادن یک تونل، معمولاً اتصال سیم بین تونل و گره‌ی سیم کشی شده بدون تغییر باقی می ماند.

حال که با روشهای حرکت دادن سیمها آشنا شدید ابتدا با استفاده از ابزار Positioning Tool و سپس به کمک کلیدهای مکان نما، یک قطعه سیم را جابه‌جا کنید.

## سیم کشی به نواحی خارج از صفحه

اگر صفحه‌ی نمودار بلوکی به اندازه‌ای باشد که نتوان تمامی المان های آن را در یک صفحه قرار داد، با استفاده از کلیدهای پیمایش می توان هر یک از المان ها را به ناحیه‌ای در خارج از صفحه حرکت داد. کشیدن و حرکت دادن ابزار Wiring Tool در لبه‌ی صفحه‌ی نمودار بلوکی در حین انجام سیم کشی سبب می شود که صفحه‌ی نمودار بلوکی به طور خودکار بلغزد.

۴

## گزینه‌های Create

جهت ایجاد اعداد ثابت و المان های کنترل و نشان دهنده، ابتدا آنها را از زیرپالت های موجود در پالت Controls انتخاب نموده، سپس آنها را به ترمینال مورد نظر سیم کشی می کنید. روش دیگر برای انجام این عمل، باز کردن منوی کرکره‌ای بر روی ترمینال مورد نظر و انتخاب یکی از گزینه‌های Create Constant، Create Control یا Create Indicator است. با این عمل، المان مذکور به همراه نوع داده‌ی متناسب با ترمینال مورد نظر ایجاد

30- Pixel

می‌شود و به صورت خودکار به آن سیم‌کشی می گردد. این قابلیت را در ضمن پیشرفت در برنامه نویسی به خاطر داشته باشید، زیرا این روش باعث تسهیل در امر برنامه نویسی شده، همچنین نوشتن برنامه جالب و جذاب می گردد.

حال با ایجاد یک المان نشان‌دهنده، نتایج حاصل از آیکن دستور Add را به نمایش در آورید. برای انجام این عمل منوی کرکره‌ای را بر روی آیکن دستور Add باز نموده، گزینه‌ی Create Indicator را انتخاب کنید. ملاحظه می کنید که در صفحه‌ی نمودار بلوکی، یک ترمینال نشان‌دهنده که به خروجی آیکن دستور Add سیم‌کشی شده است به همراه یک المان نشان‌دهنده در صفحه‌ی پانل ایجاد می شود. بنابراین ملاحظه کردید که این روش تا چه اندازه باعث صرفه جویی در زمان و سهولت برنامه نویسی می گردد.

## اجرای برنامه

Run Button

برای اجرای یک برنامه می توانید فرمان Run >> Operate را در صفحه‌ی پانل انتخاب نموده، یا بر روی دکمه‌ی Run کلیک کنید. در هنگامی که برنامه در حال اجراست، شکل ظاهری دکمه‌ی Run تغییر می کند.

Run Button (active)

در صورتی که دکمه‌ی Run به رنگ سیاه ظاهر شود نشانگر آن است که هم‌اکنون برنامه‌ی اصلی یا بالاترین مرتبه‌ی برنامه در حال اجراست. در این حالت به نظر می رسد که پیکان موجود در دکمه‌ی Run در حال حرکت است.

Run Button (subVI)

در صورتی که در داخل پیکان موجود در دکمه‌ی Run، پیکان کوچکتری قرار گرفته باشد بیانگر آن است که این برنامه به عنوان زیربرنامه در حال اجراست و بدین معنی است که در حال حاضر برنامه‌ی مذکور توسط برنامه‌ی دیگری فراخوانی شده است.

Continuous Run Button

اگر قصد دارید که یک برنامه به صورت پیوسته به اجرا در آید، دکمه‌ی Continuous Run Button را فشار دهید. اما توجه داشته باشید که این عمل، روش مناسبی برای اجرا نمودن برنامه نیست. زیرا ممکن است برنامه‌ی مورد بحث در یک حلقه‌ی بی پایان قرار گیرد و برای خارج شدن از آن مجبور شوید رایانه‌ی خود را ری‌ست کنید. اگر با این مشکل مواجه شدید، سعی کنید تا با فشار دادن کلیدهای <Ctrl> و <Cmd> به ترتیب در دو محیط Windows و Mac، فرمان اجرای برنامه را لغو کنید.

Abort Button

برای لغو اجرای برنامه‌ی اصلی یا به عبارت دیگر برنامه‌ای که در بالاترین سطح قرار گرفته است دکمه‌ی Abort را فشار دهید. در صورتی که برنامه‌ای توسط برنامه های سطوح بالاتر فراخوانی شده باشد یا به عبارت دیگر اگر برنامه‌ای به صورت زیربرنامه در برنامه های دیگر مورد استفاده قرار گرفته باشد، این دکمه به رنگ خاکستری ظاهر می شود و بدین معنی است که در این برنامه، استفاده نمودن از دکمه‌ی مذکور امکان‌پذیر نیست. کلیک کردن بر روی این دکمه منجر به توقف فوری برنامه‌ی اصلی می گردد.

۴

شایان ذکر است که این عمل، روش مناسبی برای متوقف کردن برنامه نیست. زیرا ممکن است اطلاعات و داده‌های برنامه را از دست بدهید. در آینده خواهید دید که برای متوقف ساختن برنامه از روشهای دیگری استفاده می‌شود. در این روشها، اجرای برنامه با ظرافت خاصی به پایان می‌رسد.

Pause Button

فشار دادن دکمه‌ی Pause باعث ایجاد وقفه در اجرای برنامه می‌گردد. با فشار دادن مجدد این دکمه، اجرای برنامه ادامه می‌یابد.

در محیط LabVIEW اجرا نمودن چند برنامه به طور همزمان امکان‌پذیر است. پس از اجرای اولین برنامه می‌توانید به صفحه‌ی پانل یا نمودار بلوکی برنامه‌ی دیگر بروید و آن را نیز به اجرا در آورید. توجه داشته باشید در صورت اجرای یک زیربرنامه به عنوان برنامه‌ی اصلی، در تمامی برنامه‌هایی که این زیربرنامه را فراخوانی نموده‌اند، دکمه‌ی Run به صورت پیکان شکسته ظاهر می‌شود و بیانگر آن است که در حال حاضر اجرای برنامه‌های سطوح بالاتر امکان‌پذیر نیست. در ضمن، اجرای همزمان یک زیربرنامه، هم به عنوان زیربرنامه و هم به عنوان برنامه‌ی اصلی امکان‌ناپذیر است.

## تمرین ۲-۴ : ساخت یک ترمومتر

حال سطح معلومات شما به حدی رسیده که قادر هستید برنامه‌ای که عملاً وظیفه‌ای را انجام می‌دهد، ایجاد کنید. این برنامه سطح ولتاژ ارسال شده توسط یک کارت DAQ را قرائت می‌کند و آن را بر روی صفحه‌ی پانل به نمایش در می‌آورد. در صورتی که کارت DAQ در اختیار ندارید، این برنامه با استفاده از تکنیک شبیه‌سازی، درجه حرارت قرائت شده توسط یک ترمومتر را بر روی صفحه‌ی پانل نمایش می‌دهد. در مورد استفاده از کارت DAQ، باید کانال 0 را به یک سنسور گرمایی و یا سطح ولتاژی مشابه (ترجیحاً بین 0 تا 1 ولت) ارتباط دهید. در صورتی که کارت DAQ در اختیار دارید، روش نصب و پیکربندی[31] کارت را از دفترچه‌ی راهنمای آن مطالعه کنید. روش پیکربندی کارت‌های DAQ را در دو فصل ۱۰ و ۱۱ بررسی خواهیم کرد.

۴

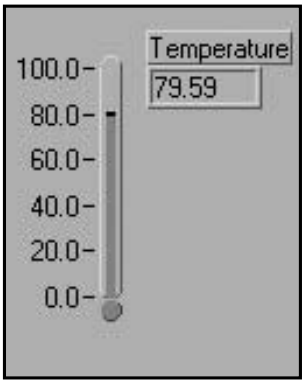
حتماً این تمرین را ذخیره نمایید. زیرا در تمرینات بعدی مطالب و المان‌های دیگری به آن اضافه خواهیم کرد. در صورتی که این برنامه را ذخیره نمی‌کنید، می‌توانید آن را با عنوان Thermometer.vi در مسیر LabVIEW\Exercise\Ch4.llb بیابید.

۱- یک برنامه‌ی جدید ایجاد کنید.

۲- با انتخاب آیکن Controls >> Numeric >> Thermometer، یک ترمومتر بر روی صفحه‌ی پانل قرار دهید. به محض ظاهر شدن ترمومتر، برچسب Thermometer را در جعبه‌ای که محل وارد نمودن برچسب اختصاصی است وارد کنید. سپس به کمک یکی از دو ابزار Operating Tool یا Labeling Tool بر روی

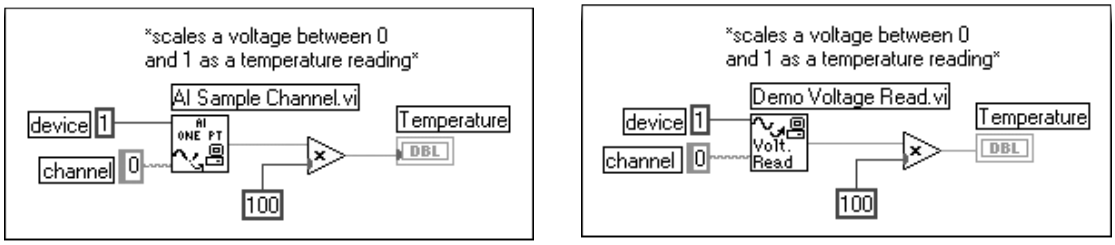
31- Configuration

عدد 10.0 در بخش درجه‌بندی ترمومتر کلیک کنید و ماوس را بر روی آن حرکت دهید. سپس برای ایجاد تغییر در مقدار حداکثر آن، عدد 100.0 را وارد نمایید. توجه کنید که مقادیر میانی نیز مطابق شکل ۲۹-۴ به صورت خودکار تغییر می یابند.




**شکل ۲۹-۴**

۳- یکی از دو صفحه‌ی نمودار بلوکی نشان داده شده در شکل ۳۰-۴ را ایجاد کنید. برای مشاهده همزمان هر دو صفحه‌ی پانل و نمودار بلوکی گزینه‌ی Windows >> Tile Left and Right را انتخاب کنید. اگر کارت DAQ در اختیار دارید صفحه‌ی (الف) را ایجاد کنید و در صورتی که به داده های شبیه سازی شده نیاز دارید صفحه‌ی (ب) را بسازید.




**صفحه‌ی نمودار بلوکی (الف)**
**در صورت در اختیار داشتن کارت DAQ**

**صفحه‌ی نمودار بلوکی (ب)**
**در صورت استفاده از داده‌های شبیه‌سازی شده**

**شکل ۳۰-۴**

همان گونه که ملاحظه می کنید در شکل (الف) از برنامه‌ی AI Sample Channel.vi استفاده شده است. این برنامه در زیرپالت Functions >> Data Acquisition >> Analog Input قرار دارد. در صورتی که کارت DAQ در اختیار ندارید، برای ایجاد شکل (ب) از برنامه‌ی Demo Voltage Read.vi که در زیرپالت Functions >> Tutorial قرار دارد استفاده نمایید. به خاطر داشته باشید که با استفاده از پنجره‌ی Help می توانید اطلاعات مفیدی در زمینه‌ی روش سیم کشی دقیق و صحیح ترمینال ها به دست آورید.

توجه داشته باشید در هنگامی که ابزار Wiring Tool را بر روی یک گره قرار می‌دهید، ترمینال متناظر با آن در پنجره‌ی Help چشمک می زند. این بدان معنی است که شما در حال سیم کشی در محل دقیق و صحیح می باشید.

با باز نمودن منوی کرکره‌ای بر روی ترمینال‌های کانکتور در برنامه‌های Demo Voltage Read.vi یا AI Sample Channel.vi و انتخاب گزینه‌ی Create Constant، یک عدد ثابت ایجاد کنید. مطمئن شوید که منوی کرکره‌ای را در مورد ترمینالی که قصد سیم کشی آن را دارید باز نموده‌اید. در غیر این صورت عدد ثابت ایجاد شده برای ترمینال دیگر منظور می گردد.

در ادامه به بررسی اجزای موجود در صفحه‌ی نمودار بلوکی می پردازیم:

برای تخصیص کانال ورودی آنالوگ در کارت DAQ به یک ثابت رشته‌ای نیاز دارید. ثابت رشته‌ای یا String Constant در زیرپالت Functions >> Strings قرار دارد. همان گونه که در شکل ۳۰-۴ ملاحظه می کنید ممکن است این کانال، کانال شماره‌ی 0 یا هر کانال دیگری باشد. در برنامه Demo Voltage Read.vi از انتساب این مقدار صرف نظر شده است. ولی ما برای شبیه سازی واقعی، عدد 0 را به کانال مربوط اختصاص داده‌ایم. اگرچه ثابت رشته‌ای حاوی یک کاراکتر عددی (نظیر 0 یا هر عدد دیگری) است، توجه داشته باشید که این عدد ثابت حتماً باید از نوع رشته‌ای باشد. در صورتی که عدد مذکور از نوع عددی باشد، سیم کشی و برقرار نمودن ارتباط با این المان امکان پذیر نخواهد بود.

زمانی که یک عدد ثابت رشته‌ای را بر روی صفحه‌ی نمودار بلوکی قرار می دهید، این رشته فاقد هر نوع داده است. اگر پس از قرار دادن المان مذکور بر روی صفحه در هیچ نقطه‌ی دیگری کلیک نکرده باشید، می توانید بلافاصله یک داده‌ی رشته‌ای را تایپ نمایید. سپس برای وارد نمودن مقدار تایپ شده کلید <Enter> را در بخش عددی صفحه کلید فشار دهید یا بر روی دکمه‌ی Enter در نوار ابزار کلیک کنید. ضمناً در صورت نیاز می توانید مقدار ثابت رشته‌ای را به کمک ابزارهای Operating Tool یا Labeling Tool تغییر دهید.

این ثابت عددی که در زیرپالت Functions >> Numeric قرار دارد، شماره‌ی کارت DAQ را مشخص می کند. این ثابت عددی ممکن است مقدار 1 یا هر عددی دیگری را اختیار کند.

۴

در محیط Windows، شماره کارت توسط نرم‌افزار NI-DAQ Configuration Utility تعیین می‌شود (برای انتساب این پارامتر، عدد 1 در نظر گرفته می‌شود مگر اینکه از چندین کارت استفاده شده باشد). در محیط Mac، شماره‌ی کارت در حقیقت شماره‌ی شیاری است که کارت مورد نظر در آن قرار می‌گیرد. با اجرای برنامه‌ی NI-DAQ می‌توانید این شماره را مشاهده کنید. برای کسب اطلاعات بیشتر در مورد شماره‌ی کارت، به دفترچه‌ی راهنمای آن مراجعه نمایید و یا مطالب مندرج در فصل ۱۰ را مطالعه کنید. در شکل (الف) فرض بر آن است که یک سنسور گرمایی به کانال شماره‌ی 0 در یک کارت DAQ متصل شده است.

این ثابت عددی، سطح ولتاژ را (که معمولاً مفهومی غیر واقعی و فرضی است) به درجه حرارت تبدیل می کند. در صورتی که ولتاژ در دسترس شما بین 0 تا 1 ولت نباشد، بهتر است برای نزدیک تر کردن دما به مقدار واقعی این عدد ثابت را تغییر دهید.

۴- با استفاده از پنجره‌ی Help، اتصالات و سرسیمها را ملاحظه کنید. به رنگ سیمها توجه داشته باشید. رنگهای متفاوت، نشانگر داده های مختلف هستند. به یاد داشته باشید که داده های عددی به رنگ آبی یا نارنجی، رشته ها به رنگ صورتی و مقادیر جبری به رنگ سبز ظاهر می شوند.

Run Button

۵- با کلیک نمودن بر روی دکمه‌ی Run، این برنامه را اجرا کنید. ملاحظه می کنید که ترمومتر، ولتاژ قرائت شده از کارت و یا مقادیر شبیه سازی شده را نمایش می دهد. در صورتی که دکمه‌ی Run در این برنامه به صورت پیکان شکسته ظاهر شده است، فصل بعد را مطالعه کنید و پس از ایجاد تغییرات لازم و رفع اشکالات موجود، مجدداً برنامه را اجرا کنید. اگر در مورد جمع آوری داده ها توسط کارت DAQ با مشکل مواجه شده‌اید، سعی کنید تا برنامه‌ی شبیه ساز را اجرا نمایید. برای جایگزین کردن برنامه‌ی شبیه ساز Demo Voltage Read.vi با AI Sample Channel.vi، منوی کرکره‌ای را بر روی آیکن این برنامه باز نموده، سپس گزینه‌ی Replace را انتخاب کنید. در این حالت ملاحظه می کنید که پالت Functions بر روی صفحه ظاهر می شود. اکنون می توانید برنامه‌ی Demo Voltage Read.vi را از زیرپالت Functions >> Tutorial انتخاب کنید. پس از انجام این عمل، آیکن برنامه‌ی AI Sample Channel.vi با آیکن برنامه‌ی Demo Voltage Read.vi جایگزین می گردد. در این بخش سعی داریم تا شما را با نحوه‌ی برنامه نویسی آشنا سازیم. هدف ما در این مثال درگیر نمودن شما با مسائلی پیشرفته از قبیل جمع آوری داده ها از طریق کارت DAQ که باعث سردرگمی شما در این فصول مقدماتی می گردد نیست.

۶- با انتخاب گزینه‌ی File >> Save این برنامه را با عنوان Thermometer.vi در زیرفهرست LabVIEW\ My Activity ذخیره کنید. در تمرینات بعدی از این برنامه به عنوان زیربرنامه استفاده خواهیم کرد.

## نکاتی سودمند در مورد برنامه نویسی

۴

همزمان با فراگیری تدریجی برنامه نویسی در محیط LabVIEW، استفاده نمودن از برخی روشهای میانبر در برنامه نویسی می تواند نقش به سزایی در پیشرفت شما داشته باشد. بنابراین به تمامی مطالب و روشهایی که در این بخش عنوان می شود توجه نموده، سعی کنید آنها را به خاطر بسپارید. مطمئن باشید به مرحله‌ای می رسید که می گویید : «ای کاش این مطالب و روشهای میانبر را پیش از این فرا گرفته بودم.»

### میانبرهای صفحه کلید

همان گونه که قبلاً ذکر شد در مورد بسیاری از گزینه های موجود در منوهای LabVIEW، میانبرهای صفحه کلید تعبیه شده است. به عنوان مثال برای ایجاد یک برنامه‌ی جدید می توان گزینه‌ی File >> New را انتخاب نمود یا

کلیدهای معادل آن یعنی <Ctrl-N> و <Cmd-N> را به ترتیب در دو محیط Windows و Mac مورد استفاده قرار داد. در جدول ۳-۴ تعدادی از میانبرهای کلیدی را که اغلب مورد استفاده قرار می گیرند ملاحظه می کنید.

جدول ۳-۴

| Windows | Mac | عملکرد |
|---|---|---|
| <Ctrl-B> | <Cmd-B> | از میان برداشتن سیم‌های منفطع |
| <Ctrl-E> | <Cmd-E> | تغییر موقعیت بین دو صفحه ی پانل و نمودار بلوکی |
| <Ctrl-F> | <Cmd-F> | جستجوی یک المان، دستور یا متن در بین برنامه‌ها |
| <Ctrl-G> | <Cmd-G> | جستجوی رویداد۱ بعدی در مورد یک المان یا متن |
| <Ctrl-H> | <Cmd-H> | ظاهر ساختن/مخفی نمودن پنجره ی Help |
| <Ctrl-N> | <Cmd-N> | ایجاد یک برنامه ی جدید |
| <Ctrl-Q> | <Cmd-Q> | خروج از برنامه ی فعال |
| <Ctrl-R> | <Cmd-R> | اجرای برنامه ی فعال |
| <Ctrl-W> | <Cmd-W> | بستن پنجره ی برنامه ی فعال |
| <Ctrl-.> | <Cmd-.> | متوقف ساختن برنامه ی فعال |

## مثالها

به مثالهای موجود در فهرست Examples نگاهی کوتاه داشته باشید. این مثالها به همین صورت، قابل استفاده می باشند. در صورت نیاز می توانید آنها را برای برآوردن نیازهای خود تغییر دهید. قبل از بررسی مثالها، برنامه‌ی readme.vi را باز و اجرا نمایید. این برنامه نیز در همین فهرست قرار دارد و در مورد محتوا و نحوه‌ی عملکرد هر یک از مثالها توضیحاتی در اختیار شما قرار می دهد. پس از مطالعه این جزییات و یافتن برنامه و مثال مورد نظر، آن را باز کنید و مورد بررسی قرار دهید.

## تعویض ابزارها

هنگامی که در محیط Edit قرار داشته باشید، با فشار دادن کلید <Tab> می توانید بین ابزارهای موجود در پالت Tools تغییر وضعیت دهید. در صورت فعال بودن صفحه‌ی پانل با فشار دادن کلید <Tab>، ابزارها به صورت‌چرخشی و به ترتیب:

Operating Tool ← Positioning Tool ← Labeling Tool ← Color Tool تغییر می یابند. در صورت فعال بودن صفحه‌ی نمودار بلوکی نیز ترتیب چرخش و تغییر به همان صورت است؛ به استثناء اینکه در این چرخه، ابزار Wiring Tool به جای Color Tool قرار می گیرد.

در ضمن فشار دادن کلید <Space bar> در دو صفحه‌ی پانل یا نمودار بلوکی به ترتیب منجر به تغییر وضعیت بین دو ابزار Positioning Tool و Operating Tool یا دو ابزار Positioning Tool و Wiring Tool می گردد.

### تعویض جهت یک قطعه سیم

در حین انجام سیم کشی، فشار دادن کلید <Space bar> جهت امتداد یافتن قطعه سیم را از آخرین نقطه‌ی اتصال تغییر می دهد. بنابراین اگر به طور اتفاقی در جهت افقی حرکت کرده ولی واقعاً قصد داشته اید که ابتدا به سمت پایین حرکت نمایید، با فشار دادن این کلید جهت اولیه را از افقی به عمودی تغییر دهید.

### حذف یک قطعه سیم

اگر در حین انجام سیم کشی تصمیم به حذف یک قطعه سیم گرفتید، در محیط Windows دکمه‌ی سمت راست ماوس را کلیک کنید و در محیط Mac سیم را به بیرون از صفحه حرکت دهید و سپس کلیک کنید.

### حذف آخرین نقطه‌ی توقف سیم

در صورتی که در حال انجام سیم کشی هستید، کلیک کردن ماوس باعث توقف سیم می گردد. برای انجام این عمل با استفاده از صفحه کلید در دو محیط Windows و Mac به ترتیب کلیدهای <Ctrl> و <Cmd> را فشار داده، سپس کلیک کنید. انجام مجدد این عمل باعث حذف نقطه‌ی قبلی از آخرین نقطه‌ی توقف سیم می گردد. اگر این عمل را چندین بار تکرار کنید تا به آخرین نقطه‌ی توقف که محل اتصال سیم به ترمینال است برسید، در این حالت سیم به طور کامل حذف می شود.

### اضافه کردن یک المان به سیمهای موجود

در LabVIEW اضافه نمودن یک المان نظیر دستور محاسباتی یا منطقی به سیمهایی که قبلاً اتصال یافته اند امکان پذیر است. شایان ذکر است که این عمل بدون جدا کردن سیمها و یا انجام سیم کشی مجدد صورت می گیرد. برای انجام این عمل، منوی کرکره ای را بر روی قطعه سیمی که قصد دارید المان مورد نظر را به آن اضافه کنید باز نموده، گزینه‌ی Insert را انتخاب کنید. سپس با انتخاب المان مورد نظر از پالت Functions یا زیرپالت های آن، المان مورد بحث را در محل مطلوب اضافه نمایید. برای حرکت دادن یک المان در فواصل کوتاه و دقیق ابتدا آن را انتخاب نمایید. سپس با فشار دادن کلیدهای مکان نما در هر ضربه، المان انتخاب شده را تنها به اندازه‌ی یک نقطه جابه‌جا کنید. برای تکرار عملیات مذکور کلید مکان نما را پایین نگه دارید تا المان انتخاب شده به حرکت خود ادامه دهد. برای حرکت دادن المان در فواصل بیشتر ابتدا کلید <Shift> را پایین نگه داشته، سپس با توجه به جهت حرکت، یکی از کلیدهای مکان نما را فشار دهید.

### تغییر سریع اعداد در المان Digital Control

اگر در حالی که بر روی پیکان های افزایش یا کاهش در یک المان Digital Control کلیک نموده اید، کلید <Shift> را نیز پایین نگه دارید، در این حالت افزایش یا کاهش مقدار نشان داده شده سریعتر صورت می گیرد.

### وارد کردن المان‌ها به یک منوی حلقوی

برای اضافه کردن سریع المان‌ها به یک منوی حلقوی پس از تایپ نمودن نام آنها، در دو محیط Windows و Mac به ترتیب از ترکیب کلیدهای <Shift-Enter> و <Shift-Return> استفاده کنید تا المان مورد نظر در نوار پذیرفته شود. سپس نشانگر ماوس را حرکت دهید تا المان بعدی اضافه گردد.

### کپی برداری از یک المان

برای کپی برداری از یک المان، المانی را که قصد کپی برداری از آن را دارید انتخاب نموده، سپس در دو محیط Windows و Mac به ترتیب کلید <Ctrl> و <Option> را پایین نگه دارید و کپی آن را به محل جدید انتقال دهید. توجه داشته باشید که المان اولیه در محل خود باقی می ماند. می توانید المان /المان های انتخاب شده را به برنامه های دیگر نیز کپی کنید.

### جابه‌جا نمودن المان در یک جهت

اگر در هنگام حرکت دادن المان، کلید <Shift> را نیز پایین نگه دارید، جهت حرکت به صورت افقی یا عمودی محدود می شود. بسته به اینکه ابتدا ماوس را در چه جهتی حرکت داده‌اید، جهت حرکت تعیین می گردد.

### یکسان کردن رنگ

برای برداشتن نمونه‌ی رنگ یک المان، توسط یکی از دو ابزار Color Tool یا Copy Color Tool بر روی المان مورد نظر کلیک کرده، سپس با ابزار Color Tool بر روی المان /المان‌هایی که قصد رنگ آمیزی آن /آنها را دارید کلیک کنید.

### جایگزین کردن المان‌ها

به راحتی می توانید با باز نمودن منوی کرکره‌ای بر روی یکی از المان های موجود در صفحات پانل یا نمودار بلوکی و انتخاب گزینه‌ی Replace، آن را با المان دیگری جایگزین کنید. در این حالت بسته به این که کدام یک از صفحات پانل یا نمودار بلوکی فعال باشد یکی از دو پالت Controls یا Functions ظاهر می گردد. پس از انتخاب دستور مذکور، المان یا دستور جدید جایگزین المان یا دستور قبلی می شود و سیمهایی که وجود آنها هنوز جایز است بر جای خود باقی می مانند.

### ایجاد فضای بیشتر در صفحه

برای اضافه کردن فضای بیشتر در صفحه‌ی پانل یا نمودار بلوکی، توسط ابزار Positioning Tool بر روی صفحه کلیک کنید و آن را به خارج از صفحه حرکت دهید. ملاحظه می کنید که صفحه‌ی پانل یا نمودار بلوکی،

جابه‌جا شده و می لغزد. در این حالت یک مستطیل را با مرزهای نقطه چین شده ملاحظه می کنید. این مستطیل مشخص کننده‌ی فضای جدید است.

## ایجاد پالت های سفارشی

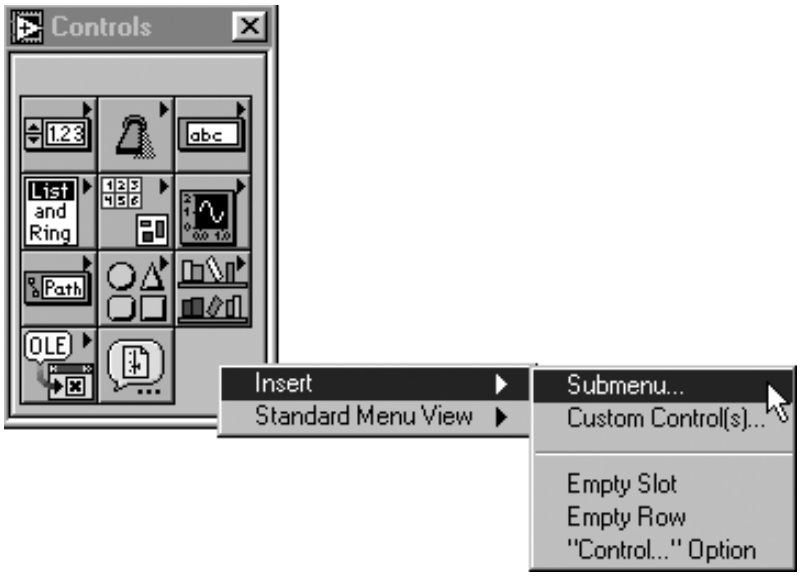
اگر در برنامه های خود از یک المان یا دستور خاص در صفحه‌ی پانل استفاده می کنید، می توانید پالت حاوی المان یا دستور مورد نظر خود را همواره باز نگه دارید. در محیط LabVIEW می توانید به منظور دسترسی سریع، برنامه ها یا المان های کنترل سفارشی را که خود ایجاد نموده‌اید به پالت های استاندارد اضافه کنید.

برای ایجاد یک پالت سفارشی مراحل زیر را انجام دهید:

۱- گزینه‌ی ...Edit >> Edit Control Function Palettes را انتخاب کنید.

۲- در این حالت گزینه‌ی new setup را انتخاب نموده، سپس نام پالت یا زیرپالت مورد نظر را که قصد تغییر و یا اضافه نمودن المان به آن را دارید وارد کنید.

۳- منوی کرکره‌ای را بر روی پالت مورد نظر باز نموده، مطابق شکل ۳۱-۴ گزینه‌ی مطلوب را از زیرمنوی Insert انتخاب کنید.



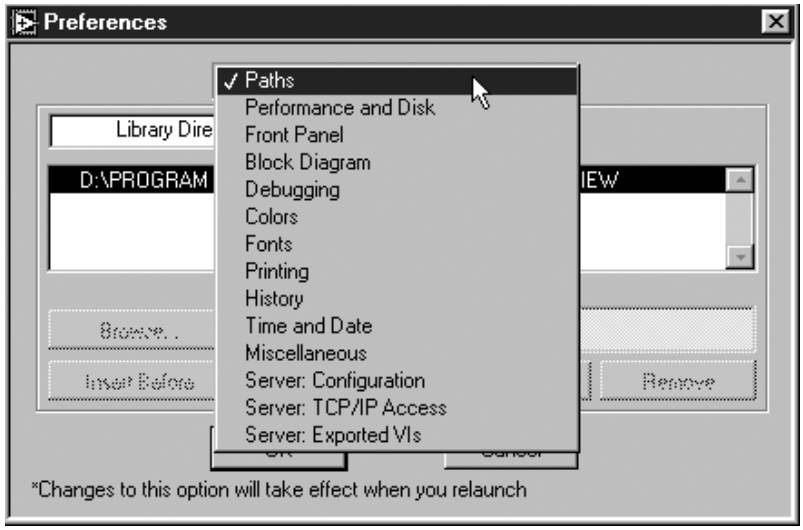

**شکل ۳۱-۴: نحوه‌ی ایجاد پالت سفارشی**

۴- در این حالت یک پنجره‌ی محاوره‌ای باز می گردد که در آن در مورد انتخاب برنامه یا المان کنترل سفارشی از شما سؤال می شود.

۵- پس از اتمام عملیات، با تخصیص نام برای پالت های جدید، آنها را ذخیره کنید. در آینده می توانید با انتخاب گزینه‌ی Edit >> Select Palette Set بین دو پالت سفارشی و پالت پیش فرض تغییر وضعیت دهید.

## تعیین اولویت

در نرم‌افزار LabVIEW اولویتهای بسیاری وجود دارد که بنا به سلیقه و برای ساده‌تر کردن عملیات می‌توانید با استفاده از گزینه‌ی ...Edit >> Preferences، آنها را انتخاب کنید. این اولویتها در منوی حلقوی موجود در بالای پنجره‌ی محاوره‌ای Preferences قرار دارند.




**شکل ۳۲-۴ : اولویتهای موجود در پنجره‌ی محاوره‌ای Preferences**

برای کسب اطلاعات بیشتر در مورد گزینه‌های Preferences، مطالب موجود در فهرست Manuals یا پنجره‌ی Online Help را مطالعه کنید.

## خلاصه

در نرم‌افزار LabVIEW روشها و ابزارهای بسیاری وجود دارد که همگی در محیط گرافیکی عمل می‌کنند. به عنوان مثال ابزار Operating Tool مقدار المان را تغییر می‌دهد. ابزار Positioning Tool برای انتخاب، حذف یا جابه‌جا کردن المان‌ها به‌کار برده می‌شود. از ابزار Wiring Tool برای سیم‌کشی و برقراری ارتباط در صفحه‌ی نمودار بلوکی استفاده می‌شود و به کمک ابزار Labeling Tool می‌توان برچسبهای اختصاصی یا آزاد را ایجاد نمود و یا آنها را تغییر داد. همان طورکه از نام برچسبهای اختصاصی استنباط می‌گردد، این برچسبها به المان خاصی تعلق ندارند و حذف یا جابه‌جایی آنها به طور مستقل امکان‌پذیر نیست. در حالی که در مورد برچسب آزاد چنین محدودیتی وجود ندارد.

در LabVIEW، چهار نوع المان کنترل و نشان‌دهنده‌ی اصلی وجود دارد که عبارتند از: المان‌های عددی، جبری، رشته‌ای و مسیر. هر یک از این المان‌ها حاوی نوع خاصی از داده‌ها هستند و دارای گزینه‌های بخصوصی در منوی کرکره‌ای می‌باشند. رنگ و شکل سیمها و ترمینال‌های قرار گرفته در صفحه‌ی نمودار بلوکی بسته به نوع داده‌ها متفاوت است. اعداد اعشاری به رنگ نارنجی، اعداد صحیح به رنگ آبی، مقادیر

جبری به رنگ سبز، رشته ها به رنگ ارغوانی و مسیرها به رنگ سبز متمایل به آبی ظاهر می شوند.

با استفاده از پالت های Controls یا Functions و زیرپالت های آنها می توان المان ها و دستورها را در صفحه‌ی پانل یا نمودار بلوکی قرار داد. برای دستیابی به این پالت ها همچنین می توان منوی کرکره‌ای را در ناحیه‌ای خالی از صفحه‌ی پانل یا نمودار بلوکی باز نمود.

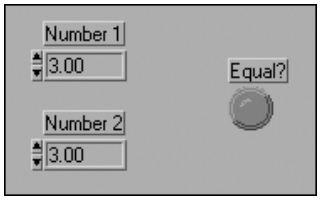
برای اجرای برنامه می توان دکمه‌ی Run را در نوار ابزار فشار داد و یا گزینه‌ی Operate >> Run را انتخاب نمود. در صورتی که دکمه‌ی Run در یک برنامه به صورت پیکان شکسته ظاهر شود، بیانگر آن است که این برنامه خالی از اشکال نیست. در فصل بعد روشهای اشکال زدایی[۳۳] برنامه را مورد بررسی قرار خواهیم داد.

## تمرینات اضافی

### تمرین ۴-۳ : مقایسه‌ی دو عدد

با ایجاد یک برنامه، دو عدد ورودی را مقایسه کنید. در صورت تساوی دو مقدار ورودی، LED موجود در صفحه‌ی پانل روشن می‌شود. این برنامه را با عنوان Comparasion Practice.vi در زیرفهرست LabVIEW\My Activity ذخیره کنید.

صفحه‌ی پانل در این برنامه باید به صورت شکل ۴-۳۳ باشد.

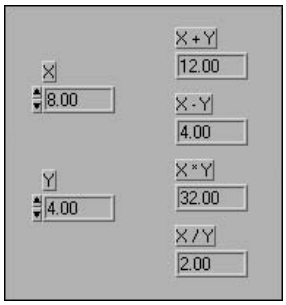



**شکل ۴-۳۳ : صفحه‌ی پانل در برنامه‌ی Comparison Practice.vi**

### تمرین ۴-۴ : طراحی یک ماشین حساب بسیار ساده

با ایجاد یک برنامه، مقدار حاصل جمع، تفریق، حاصل‌ضرب و تقسیم دو عدد ورودی را به دست آورده، نتایج حاصل را بر روی صفحه‌ی پانل به نمایش در آورید. این برنامه را با عنوان Very Simple Calculator.vi در زیرفهرست LabVIEW\My Activity ذخیره کنید.

صفحه‌ی پانل در این برنامه باید به صورت شکل ۴-۳۴ باشد.




**شکل ۴-۳۴ : صفحه‌ی پانل در برنامه‌ی Very Simple Calculator.vi**

۴

33- Debugging

LabVIEW

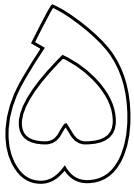
# ۵

# اصول برنامه‌نویسی و اشکال‌زدایی برنامه

## در این فصل با موارد زیر آشنا می‌شوید:

- روش بارگذاری و ذخیره‌ی برنامه
- روشهای مختلف رفع اشکال برنامه در نرم‌افزار LabVIEW
- نحوه‌ی ایجاد زیربرنامه و نحوه‌ی استفاده از آن
- شیوه‌ی مستندسازی برنامه و ارائه‌ی توضیح در مورد برنامه و المان‌های موجود در آن

# LabVIEW



## اصول برنامه‌نویسی و اشکال‌زدایی برنامه

در این فصل اصول بیشتری در مورد LabVIEW فرا می‌گیرید. در اینجا سعی داریم در مورد دستورها و روشهای ذخیره و بارگذاری فایل‌های کتابخانه‌ای در نرم‌افزارLabVIEW، روشهای اشکال‌زدایی، نحوه‌ی استفاده از زیربرنامه‌ها و مستندسازی برنامه‌ها به بحث بپردازیم.

در این فصل اهداف زیر دنبال می‌شود:

- فراگیری روش بارگذاری و ذخیره‌ی برنامه.
- به‌کارگیری روشهای مختلف رفع اشکال[1] برنامه در نرم‌افزار LabVIEW.
- ایجاد زیربرنامه و نحوه‌ی استفاده از آن.
- مستندسازی برنامه و ارائه‌ی توضیح در مورد برنامه و المان‌های موجود در آن.

اصطلاحات کلیدی موجود در این فصل عبارتند از:

- فایل کتابخانه‌ای (VI Library)
- برنامه‌ی معیوب (Broken VI)
- روش اجرای تک‌مرحله‌ای (Single Step Mode)
- گره (Node)
- روش اجرای متمایز (Execution Highlighting)
- پروب (Probe)

---

1- Debug

- نقطه‌ی توقف (Breakpoint)
- زیربرنامه (SubVI)
- محیط طراحی آیکن (Icon Editor)
- اتصالات الزامی، پیشنهادی و اختیاری (Required, Recommended, Optional Inputs)

## بارگذاری و ذخیره‌ی برنامه‌ها

در خلال برنامه نویسی و پس از آن، مطمئناً بارگذاری و ذخیره نمودن برنامه های نوشته شده الزامی است. در نرم‌افزار LabVIEW قابلیتها و گزینه های بسیار زیادی جهت ذخیره‌ی فایل ها و برنامه های مورد نیاز شما تعبیه شده است. این بخش به بررسی روش استفاده از این قابلیتها اختصاص داده شده است. برای بارگذاری یا به عبارت ساده‌تر برای بازکردن یک برنامه کافی است گزینه‌ی ...File >> Open را انتخاب کنید. در این حالت یک پنجره‌ی محاوره‌ای ظاهر می شود. به کمک این پنجره می توانید برنامه‌ی مورد نظر را از فهرست ها یا کتابخانه های موجود، بارگذاری نموده، یا عملیات بارگذاری را لغو کنید.

با انتخاب گزینه‌ی Save (یا گزینه های مشابه دیگر) از منوی File می توان برنامه های نوشته شده را ذخیره نمود. در این حالت یک پنجره‌ی محاوره‌ای ظاهر می شود و به شما اجازه می دهد تا نام و محل ذخیره شدن برنامه را تعیین کنید. در صورت ذخیره نمودن برنامه ها به صورت فایل های مجزا و نه در کتابخانه، نام انتخاب شده برای برنامه باید از قواعد نام‌گذاری فایل ها در سیستم عامل پیروی کند. به عنوان مثال در نرم‌افزار Windows 3.1 نام فایل ها حداکثر شامل ۸ کاراکتر است.

برای اجتناب از این محدودیت می توانید برنامه ها را به صورت فشرده و در یک نوع فایل به خصوص در محیط LabVIEW ذخیره نمایید. این نوع فایل را در اصطلاح «فایل کتابخانه ای[2]» گویند. به خاطر داشته باشید که برای باز کردن برنامه ها، LabVIEW براساس نام آنها به محل ذخیره‌ی برنامه ها مراجعه می کند. بنابراین وجود دو برنامه با نام یکسان در یک مسیر و به طور همزمان امکان پذیر نیست.

توجه داشته باشید که در نوار عنوان[3] برنامه هایی که در آنها تغییراتی ایجاد کرده ولی هنوز آنها را ذخیره ننموده اید، مطابق شکل ۱-۵ یک علامت «*» دیده می شود.

شکل ۱-۵

ما اطمینان داریم که همگی شما به اهمیت ذخیره نمودن برنامه ها به خوبی واقف هستید؛ اما به شما توصیه می کنیم که در هنگام نوشتن برنامه و پس از ایجاد هرگونه تغییر، برنامه را در چند مرحله ذخیره کنید و پس از آن، مجدداً به اعمال تغییرات و اصلاحات بپردازید؛ زیرا از زمان بروز اشکال احتمالی در سیستم خود و یا قطع احتمالی برق شهر اطلاع ندارید. واضح است که در صورت عدم ذخیره‌ی برنامه و بروز هرگونه اشکال در

۵

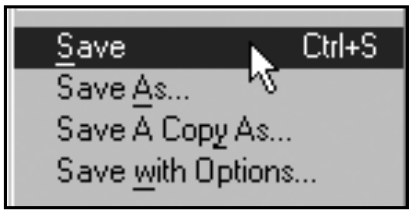
2- VI Library
3- Title Bar

سیستم الکتریکی رایانه، دستیابی به برنامه‌ی مذکور امکان ناپذیر است.

از ذخیره نمودن برنامه‌های خود در فهرست Vi.lib خودداری کنید. این فهرست در هر نسخه‌ی LabVIEW توسط شرکت NI به روز رسانده می‌شود. واضح است که در صورت ذخیره کردن برنامه‌ها در فهرست مذکور و نصب نسخه‌ی جدید نرم‌افزار LabVIEW، برنامه‌های خود را از دست خواهید داد.

## گزینه‌های مختلف برای ذخیره کردن برنامه

برای ذخیره نمودن برنامه ها می توانید در موارد مختلف یکی از چهار گزینه‌ی موجود در منوی File را به کار برید.




**شکل ۲-۵: گزینه‌های موجود در منوی File برای ذخیره نمودن برنامه‌ها**

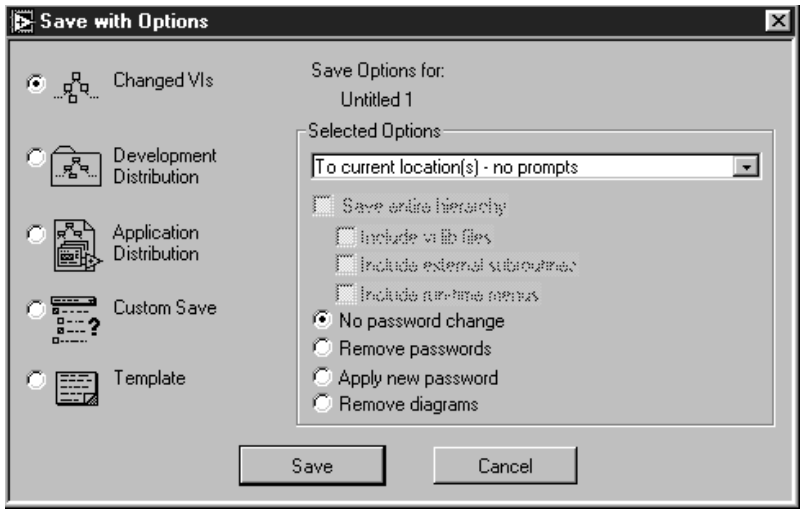
جهت ذخیره کردن یک برنامه‌ی جدید گزینه‌ی Save را انتخاب نموده، سپس با اختصاص یک نام به برنامه‌ی جدید و تعیین محل قرار گرفتن آن، برنامه را ذخیره کنید. از این گزینه نیز می توان برای ذخیره نمودن تغییرات انجام شده در یک برنامه که در محل قرار گرفتن آن تغییری ایجاد نشده است استفاده نمود.

با انتخاب گزینه‌ی ...Save As می توان نام جدیدی در حافظه به برنامه نسبت داد و یک نسخه‌ی کپی از آن را با نام جدید ذخیره نمود. در صورتی که یک نام جدید به برنامه‌ی خود اختصاص دهید، این نسخه بر روی نسخه‌ی قبلی بازنویسی نمی شود. بلکه LabVIEW، آن را با نامی جدید (و احتمالاً در محلی دیگر بسته به انتخاب شما) ذخیره می کند. در صورت عدم تعویض نام برنامه، یک پنجره‌ی محاوره‌ای باز شده، در مورد تمایل شما نسبت به بازنویسی بر روی فایل اولیه سؤال می شود.

در صورت انتخاب گزینه‌ی ...Save A Copy As، یک نسخه از برنامه‌ی موجود در حافظه در محلی دیگر که در پنجره‌ی محاوره‌ای تعیین می گردد ذخیره می شود. این گزینه برخلاف گزینه‌ی ...Save As، بر روی نام برنامه‌ی موجود در حافظه تأثیرگذار نیست. بدین معنی که با انتخاب این گزینه می توان یک نسخه از برنامه را با همان نام و در محلی دیگر ذخیره نمود.

با انتخاب گزینه‌ی ...Save with Options یک پنجره‌ی محاوره‌ای مطابق شکل ۳-۵ باز می شود که در آن می توان نحوه‌ی ذخیره شدن برنامه را تعیین نمود. به کمک زیرگزینه های موجود در این پنجره می توانید تمام طبقات و سلسله مراتب برنامه‌ی مورد نظر را ذخیره کنید. همچنین با استفاده از زیرگزینه های مذکور می توان

برنامه‌ی مورد نظر را به صورت سفارشی ذخیره نمود و یا آن را بدون صفحه‌ی نمودار بلوکی ذخیره کرد. البته قبل از ذخیره کردن برنامه بدین روش، حتماً یک نسخه‌ی کپی از صفحه‌ی نمودار بلوکی را در محلی دیگر ذخیره نمایید تا در صورت نیاز بتوانید تغییرات لازم را در آن ایجاد کنید.




**شکل ۳-۵: پنجره‌ی محاوره‌ای Save with Options**

توجه داشته باشید که پس از ذخیره نمودن برنامه بدون صفحه‌ی نمودار بلوکی، ویراستاری و ایجاد تغییر در آن امکان‌پذیر نیست. بنابراین همواره قبل از ذخیره‌ی برنامه بدون صفحه‌ی نمودار بلوکی، یک نسخه‌ی کپی از برنامه‌ی اصلی را در محلی خاص ذخیره نمایید.

### گزینه‌ی ...Revert

با انتخاب گزینه‌ی ...Revert << File امکان بازگشت به آخرین نسخه‌ی ذخیره شده از برنامه‌ای که تغییراتی را در آن اعمال نموده‌اید وجود دارد. در این حالت یک پنجره‌ی محاوره‌ای ظاهر می‌شود که در مورد حذف آخرین تغییرات انجام شده و تأیید یا عدم تأیید آنها در برنامه از کاربر سؤال می‌شود.

### پنجره‌های محاوره‌ای برای بارگذاری و ذخیره‌ی برنامه

در محیط LabVIEW پنجره‌های محاوره‌ای بسیاری وجود دارد. به عنوان مثال در هنگام ذخیره نمودن برنامه، یک پنجره‌ی محاوره‌ای ظاهر می‌شود و محل ذخیره‌ی برنامه را از کاربر سؤال می‌کند. در این پنجره کاربر می‌تواند با انتخاب دکمه‌ی Cancel عملیات ذخیره‌ی برنامه را لغو کند. در صورتی که بر روی نام یک کتابخانه کلیک کنید، یک پنجره‌ی محاوره‌ای باز می‌شود که به کمک آن می‌توانید برنامه‌های مورد نظر را انتخاب نمایید.

به دلیل اینکه انتخاب و ذخیره نمودن فایل ها در کتابخانه معمولاً از طریق پنجره‌ی محاوره‌ای مربوط به سیستم عامل انجام می گیرد، در صورت استفاده از کتابخانه ها ممکن است ترجیح دهید از پنجره های محاوره‌ای نرم افزار LabVIEW استفاده کنید. برای تعیین این اولویت، گزینه‌ی ...Edit >> Preferences را انتخاب کرده، سپس گزینه‌ی Miscellaneous >> Use native file dialogs را از منوی حلقوی موجود در بالای پنجره‌ی محاوره‌ای Preferences غیرفعال کنید.

## فایل های کتابخانه‌ای

کتابخانه ها، فایل های خاص نرم‌افزار LabVIEW می‌باشند که همانند فهرست ها قابلیت بارگذاری و ذخیره‌ی برنامه های LabVIEW را دارا هستند. در بسته‌ی نرم‌افزاری LabVIEW می توانید چند برنامه را در یک گروه دسته‌بندی کنید و آنها را در یک کتابخانه ذخیره نمایید. کتابخانه ها مزایا و معایب متعددی دارند. به عنوان مثال آنها تنها می توانند برنامه ها را به صورت فشرده در خود ذخیره سازند و شامل داده های مستقل و یا فایل های مجزا نیستند.

به طور کلی می توان گفت که سیستم عامل، کتابخانه را به عنوان یک فایل منفرد در نظر می گیرد که محتوای این فایل، تنها به کمک نرم‌افزار LabVIEW قابل دستیابی است. مطالب زیر را با دقت مطالعه کنید تا دریابید که کدام یک از دو روش ذخیره‌ی برنامه نیاز های شما را برآورده می سازد.

## دلایل استفاده از کتابخانه

- در صورتی که نرم‌افزار LabVIEW تحت Window 3.1 بر روی رایانه‌ی شما اجرا گردد و یا تصمیم دارید برنامه را به محیط Window 3.1 انتقال دهید، به جای استفاده از حداکثر ۸ کاراکتر برای نام گذاری فایل ها، امکان به کارگیری حداکثر ۲۵۵ کاراکتر وجود دارد.
- در صورتی که قصد دارید برنامه های خود را به سیستم عامل دیگری انتقال دهید،(به عنوان مثال قصددارید تا برنامه های خود را از محیط Mac به Windows انتقال دهید) استفاده از کتابخانه به مراتب ساده تر از فایل های مجزاست.
- ذخیره‌ی برنامه ها در کتابخانه ها به صورت فشرده انجام می گیرد و مسلماً در این حالت فضای کمتری از دیسک اشغال می گردد.

## دلایل ذخیره‌ی برنامه ها در فایل های مجزا

- با استفاده از قابلیتهای سیستم عامل رایانه‌ی خود و بدون نیاز به محیط LabVIEW می توانید اعمالی نظیر کپی برداری، انتقال، تغییر نام، تهیه‌ی نسخه‌ی پشتیبان و... را انجام دهید.
- در کتابخانه ها استفاده از قابلیتهای سلسله مراتب امکان پذیر نیست. به عبارت دیگر نمی توان برنامه ها

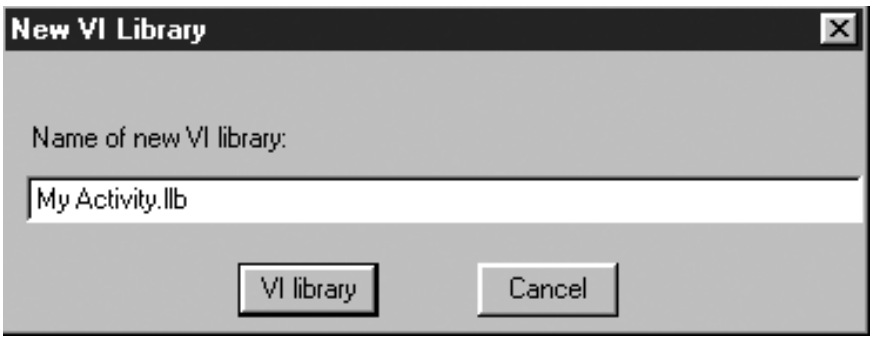
را در فهرست یا زیرفهرست ها ذخیره نمود. زیرا کتابخانه به صورت یک فایل مستقل عمل نموده و فاقد هرگونه زیرفهرست یا اجزای کوچک تر است.

- بارگذاری و ذخیره نمودن برنامه ها در فایل های جداگانه سریع تر از کتابخانه ها صورت می گیرد.
- ذخیره کردن برنامه ها در فایل های مجزا از پشتوانه محکم تری نسبت به ذخیره نمودن آنها در یک فایل کتابخانه ای برخودار است.

در این کتاب از شما خواسته می شود تا پس از انجام تمرینات در تمامی سیستم ها، برنامه‌ی ایجاد شده را در زیرفهرست LabVIEW\My Activity ذخیره نمایید. البته بسته به نظر خود می توانید برنامه ها را در کتابخانه‌ی LabVIEW\My Activity.llb نیز ذخیره کنید. در صورت استفاده از محیط Windows 3.1 به دلیل وجود محدودیت در تعداد کاراکترهای نام فایل، یک کتابخانه با عنوان MyAct.llb در فهرست LabVIEW ایجاد کنید.

## روش استفاده از کتابخانه ها

برای ایجاد یک کتابخانه‌ی جدید در سیستم عامل Windows بر روی دکمه‌ی New VI Library در یکی از پنجره های محاوره‌ای مربوط به گزینه های Save، ...Save As یا ...Save A Copy As کلیک کنید. سپس در پنجره‌ی محاوره‌ای نشان داده شده در شکل ۴-۵ نامی دلخواه با پسوند llb. را برای این کتابخانه وارد نمایید. حال بر روی دکمه‌ی VI Library کلیک کنید تا یک کتابخانه‌ی جدید ایجاد گردد. اگر پسوند llb. را وارد نکنید، نرم افزار LabVIEW این پسوند را به صورت خودکار به نام کتابخانه اضافه می کند.



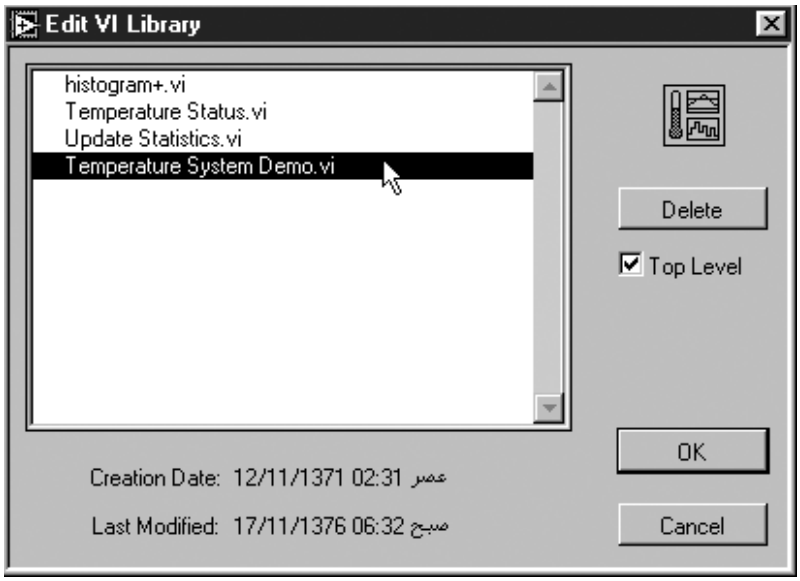

**شکل ۴-۵: پنجره‌ی محاوره‌ای New VI Library**

پس از ایجاد یک کتابخانه می توانید برنامه ها را در آن ذخیره نموده، تنها از طریق LabVIEW به آنها دسترسی پیدا کنید. به خاطر داشته باشید که به کمک سیستم عامل رایانه‌ی خود دستیابی به برنامه های مجزا در این کتابخانه امکان ناپذیر است.

برای ایجاد یک کتابخانه‌ی جدید در سیستم عامل Mac، بر روی دکمه‌ی Use LLBs در پنجره‌ی محاوره‌ای گزینه‌ی Save کلیک کنید و در پنجره‌ی محاوره‌ای دیگری که بعداً ظاهر می شود گزینه‌ی New را انتخاب کنید.

## پنجره‌ی محاوره‌ای Edit VI Library

به دلیل اینکه به کمک سیستم عامل رایانه، ویراستاری و اعمال تغییرات بر روی برنامه های ذخیره شده در کتابخانه امکان‌پذیر نیست، به ناچار باید از پنجره‌ی محاوره‌ای Edit VI Library استفاده کنید. برای استفاده از این پنجره گزینه‌ی ... Edit VI Library << File را انتخاب نمایید. پس از انتخاب این گزینه پنجره‌ای باز می شود. پس از انتخاب یکی از کتابخانه ها، پنجره‌ی محاوره‌ای Edit VI Library مطابق شکل ۵-۵ باز شده، در آن لیستی از برنامه های موجود در کتابخانه نشان داده می شود. در پایین این پنجره زمان و تاریخ ایجاد برنامه و همچنین زمان و تاریخ انجام آخرین تغییرات بر روی برنامه‌ی انتخاب شده به نمایش در می آید.




**شکل ۵-۵: پنجره‌ی محاوره‌ای Edit VI Library**

در صورت فعال نمودن گزینه‌ی Top Level در این پنجره یا به عبارت دیگر انتخاب VI به صورت برنامه‌ی اصلی، به هنگام باز نمودن کتابخانه، برنامه‌ی اصلی به طور خودکار بارگذاری می‌شود. در یک کتابخانه می‌توانید بیش از یک برنامه را به عنوان برنامه‌ی سطح بالا انتخاب کنید. نام برنامه‌های اصلی در بخش جداگانه‌ای در بالای پنجره‌ی محاوره‌ای ظاهر می‌شود. بنابراین تشخیص سطوح مختلف برنامه ساده‌تر خواهد بود. بدین ترتیب می‌توانید به راحتی تشخیص دهید که کدام برنامه به صورت برنامه‌ی اصلی و کدام یک به صورت زیربرنامه به‌کار برده شده است.

## روشهای اشکال زدایی برنامه

آیا تا به حال موفق شده‌اید برنامه‌ای عاری از اشکال را ایجاد نمایید؟ اگر تاکنون موفق به انجام این کار شده اید، به شما تبریک می گوییم. اما معمولاً در برنامه های نوشته شده اشکالاتی وجود دارد.

نرم‌افزار LabVIEW قابلیتهای بسیار زیادی دارد که به شما کمک می کند تا به رفع اشکال برنامه های خود بپردازید. این بخش به بررسی روشهای رفع اشکال برنامه اختصاص یافته است.

## رفع اشکال یک برنامه‌ی معیوب

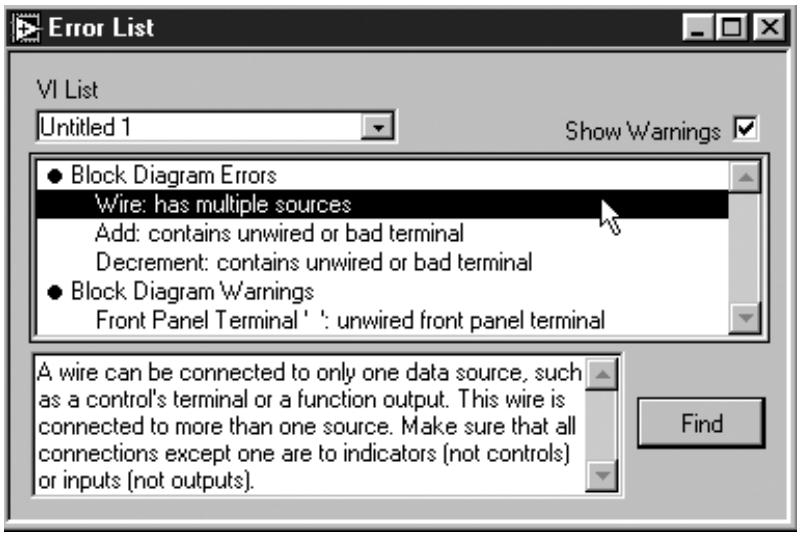
Run Button (broken)

یک برنامه‌ی معیوب برنامه‌ای است که نتوان آن را اجرا نمود. برای نشان دادن وجود اشکال در برنامه، دکمه‌ی Run در این حالت به صورت یک «پیکان شکسته» ظاهر می گردد.

در هنگام ایجاد یا ویرایش برنامه و همچنین در زمان انجام عملیات سیم کشی در صفحه‌ی نمودار بلوکی، شکسته بودن دکمه‌ی Run و یا معیوب بودن برنامه امری کاملاً عادی و طبیعی است. زیرا در مد Edit قاعدتاً نباید برنامه را اجرا کنید.

در برخی موارد برای رفع اشکال برنامه‌ی معیوب ممکن است لازم باشد که سیمهای نادرست و اشتباه را حذف کنید. برای انجام این عمل گزینه‌ی Edit >> Remove Bad Wires را انتخاب کنید. اما مراقب باشید که سیمهای مورد نیاز خود را حذف نکنید.

برای درک علت معیوب بودن برنامه یا به عبارت دیگر برای مشاهده‌ی خطاهای موجود در برنامه کافی است بر روی دکمه‌ی Run که در این حالت به صورت پیکان شکسته ظاهر می شود کلیک کنید یا گزینه‌ی Windows >> Show Error List را انتخاب نمایید. در این حالت مطابق شکل ۶-۵ پنجره‌ای با عنوان Error List که شامل لیستی از اشکالات و خطاهای موجود در برنامه است ظاهر می گردد.




**شکل ۶-۵: پنجره‌ی Error List**

برای کسب اطلاعات بیشتر در مورد خطاهای موجود در برنامه بر روی آنها کلیک کنید. در این حالت پنجره‌ی Error List اطلاعات بیشتری در مورد خطای انتخاب شده در اختیار شما قرار می دهد. برای تشخیص منشأ ایجاد خطا در برنامه بر روی خطای موجود در لیست دو بار کلیک کنید یا اینکه ابتدا خطای موجود در لیست را انتخاب نموده، سپس دکمه‌ی Find را فشار دهید. در این حالت پنجره‌ی مربوط فعال می‌شود و المانی که باعث ایجاد خطا شده است از سایر المان ها متمایز می گردد. برای مطالعه‌ی فهرست تمامی خطاهای مندرج در پنجره‌ی Error List به فایل Appendices >> Appendix 1 بر روی لوح فشرده مراجعه کنید.

## هشدارها

Warning Button

در صورت نیاز به اطلاعات بیشتر جهت رفع اشکال برنامه، گزینه‌ی Show Warnings را در پنجره‌ی Error List فعال کنید. یک هشدار، موردی خارج از قاعده نیست و موجب معیوب شدن برنامه نمی‌شود؛ اما برای LabVIEW محسوس و قابل درک نیست. نظیر ترمینال یک المان کنترل در صفحه‌ی نمودار بلوکی که به هیچ یک از ترمینال های دیگر سیم کشی نشده است. در صورت فعال بودن گزینه‌ی Show Warning در پنجره‌ی Error List، دکمه‌ی Warning در نوار ابزار ظاهر می شود. برای مشاهده‌ی هشدارهای موجود در پنجره‌ی Error List، بر روی این دکمه کلیک کنید.

همچنین برای ظاهر ساختن هشدارها به صورت پیش‌فرض می‌توانید از اولویتهای LabVIEW استفاده نمایید. برای انجام این عمل ابتدا گزینه‌ی ...Edit >> Preferences را انتخاب نموده، سپس گزینه‌ی Debugging >> Show warnings in error box را از منوی حلقوی موجود در بالای پنجره‌ی محاوره‌ای Preferences فعال کنید.

## خطاها و اشکالات معمول

معمولاً دسته‌ای از اشتباهات و خطاها در بین مبتدیان مشترک می‌باشند. بنابراین برای سهولت در رفع اشکال برنامه آنها را به صورت یک لیست تنظیم کرده‌ایم. در صورتی که دکمه‌ی Run در برنامه‌ای به صورت پیکان شکسته ظاهر گردد، ممکن است یکی از عوامل زیر باعث ایجاد خطا شده باشد.

- ممکن است یک یا چند ترمینال مربوط به دستورهای LabVIEW در صفحه‌ی نمودار بلوکی سیم کشی نشده باشند. در هنگام ایجاد برنامه مجاز نیستید تا ترمینال های موجود در صفحه‌ی نمودار بلوکی را بدون سیم کشی رها کنید و حتماً باید این ترمینال ها را به ترمینال ها یا المان های دیگر در این صفحه ارتباط دهید.
- ممکن است درصفحه‌ی نمودار بلوکی یک قطعه سیم منقطع یا در اصطلاح bad wire وجود داشته باشد. این اشکال ممکن است در اثر عدم هماهنگی سیم حامل داده ها با نوع داده‌ای که از آن عبور می کند ایجاد شود. همچنین ممکن است در یک قطعه سیم، قطعی یا ناپیوستگی وجود داشته و یا انتهای سیم به هیچ ترمینالی متصل نشده باشد. در ضمن امکان دارد که سیمهای منقطع در زیر المان های دیگر مخفی شده و یا به قدری کوچک باشند که قادر به رؤیت آنها نباشید. برای از بین بردن این سیمها گزینه‌ی Edit >> Remove Bad Wires را انتخاب کنید.
- ممکن است یک زیربرنامه که در برنامه‌ی اصلی استفاده شده است، معیوب باشد و یا امکان دارد که پس از قرار دادن آیکن این زیربرنامه در صفحه‌ی نمودار بلوکی، کانکتور آن را تغییر داده باشید. برای برقراری ارتباط مجدد این زیربرنامه یکی از دو گزینه‌ی Relink To SubVI یا Replace را از منوی کرکره‌ای آیکن زیربرنامه انتخاب نمایید.

- ممکن است اشکال ایجاد شده مربوط به المانی باشد که غیرفعال یا غیرقابل رؤیت است. همچنین امکان دارد که المانی در اثر به کارگیری گره‌ی خصوصیت تغییر یافته باشد. این مطلب را در فصل ۱۳ مورد بررسی قرار خواهیم داد.
- ممکن است دو المان کنترل را ناخواسته به یکدیگر سیم‌کشی نموده، یا آنها را به یک المان نشان‌دهنده ارتباط داده باشید. در صورت بروز چنین اشکالی در پنجره‌ی Error List پیغام «Signal: has multiple sources» را ملاحظه می کنید. در اکثر موارد با تعویض یکی از دو المان مذکور با یک المان نشان دهنده می توان این اشکال را رفع نمود.

## اجرای برنامه در مد تک‌مرحله‌ای

Pause Button

برای رفع اشکال یک برنامه ممکن است لازم باشد تا اجرای برنامه را در صفحه‌ی نمودار بلوکی از یک گره تا گره‌ی بعدی بررسی کنید. واژه‌ی گره می تواند دربرگیرنده‌ی یکی از موارد زیربرنامه، دستور، حلقه، گره‌ی فرمولی یا گره‌ی خصوصیت باشد. برای اجرای مرحله به مرحله‌ی برنامه، یکی از دکمه های Single Step را به جای دکمه‌ی Run فشار دهید. برای ایجاد وقفه بر روی دکمه‌ی Pause کلیک کنید و یا نقاط توقف را در برنامه قرار دهید. برای آغاز اجرای مجدد برنامه در حالت عادی، مجدداً کلید Pause را فشار دهید.

ممکن است قصد داشته باشید در صورت اجرای هر مرحله، آن بخش در برنامه را از سایر بخش‌ها متمایز کنید. بدین ترتیب می توانید عبور داده ها را از گره ها دنبال کنید. در مورد این روش در بخش بعدی توضیح می دهیم. هنگامی که برنامه را به صورت تک مرحله ای به اجرا در می آورید با فشار دادن هر یک از سه دکمه‌ی فعال می توانید مرحله‌ی بعدی را اجرا کنید. دکمه‌ی انتخاب شده نحوه‌ی اجرای مرحله‌ی بعدی را مشخص می کند.

Step Into Button

برای اجرای اولین مرحله از یک برنامه، زیربرنامه یا یک حلقه و سپس توقف در مرحله بعدی، دکمه‌ی Step Into را فشار دهید یا با استفاده از کلیدهای مکان نما، کلید <Down> را به همراه کلید <Ctrl> و <Cmd> به ترتیب در دو محیط Windows و Mac به کار برید.

Step Over Button

برای اجرای یک حلقه یا زیربرنامه و سپس توقف در گره‌ی بعدی، دکمه‌ی Step Over را فشار دهید یا با استفاده از کلیدهای مکان نما، کلید <Right> را به همراه کلید <Ctrl> و <Cmd> به ترتیب در دو محیط Windows و Mac به کار برید.

Step Out Button

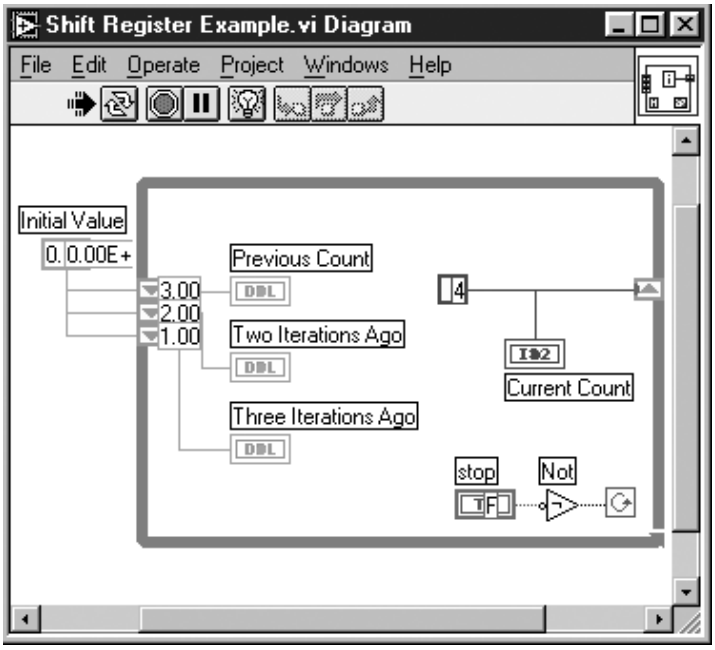
برای اجرای برنامه یا حلقه و سپس توقف آن، دکمه‌ی Step Out را فشار دهید یا با استفاده از کلیدهای مکان نما، کلید <Up> را به همراه کلید <Ctrl> و <Cmd> به ترتیب در دو محیط Windows و Mac به کار برید.

## روش اجرای متمایز

Execution Highlighting Button

در برخی موارد رؤیت محل دقیق داده‌ها در صفحه‌ی نمودار بلوکی و دانستن این مطلب که در حال حاضر چه عملی بر روی آنها انجام می گیرد بسیار جالب است. در نرم‌افزار LabVIEW می توانید تصویری زنده از اجرای برنامه را بر روی صفحه‌ی نمودار بلوکی ملاحظه کنید. برای فعال ساختن این قابلیت، بر روی دکمه‌ی Execution Highlighting کلیک کنید.

در حالی که داده‌ها از یک گره به گره‌ی بعدی انتقال می یابند، جابه‌جایی داده‌ها در داخل سیم با حرکت حباب هایی در سیم حامل داده‌ها نشان داده می شود. توجه داشته باشید که در این محیط، سرعت اجرای برنامه به شدت کاهش می یابد. برای بازگشت به حالت اجرای عادی برنامه، مجدداً بر روی این دکمه کلیک کنید. در شکل ۷-۵ برنامه‌ای را که در این محیط در حال اجراست ملاحظه می کنید.




**شکل ۷-۵: نحوه‌ی به اجرا در آمدن برنامه‌ی Shift Register Example.vi در مد اجرای متمایز یا Execution Highlighting**

در صورت انتخاب گزینه‌ی Auto probe during execution highlighting << Debugging در منوی حلقوی موجود در پنجره‌ی محاوره‌ای Preferences، مقادیر گره به طور خودکار نشان داده می شود. برای درک نحوه‌ی عبور داده‌ها از گره‌ها بهتر است دو مد اجرای متمایز و تک مرحله‌ای را همزمان به کار برید.

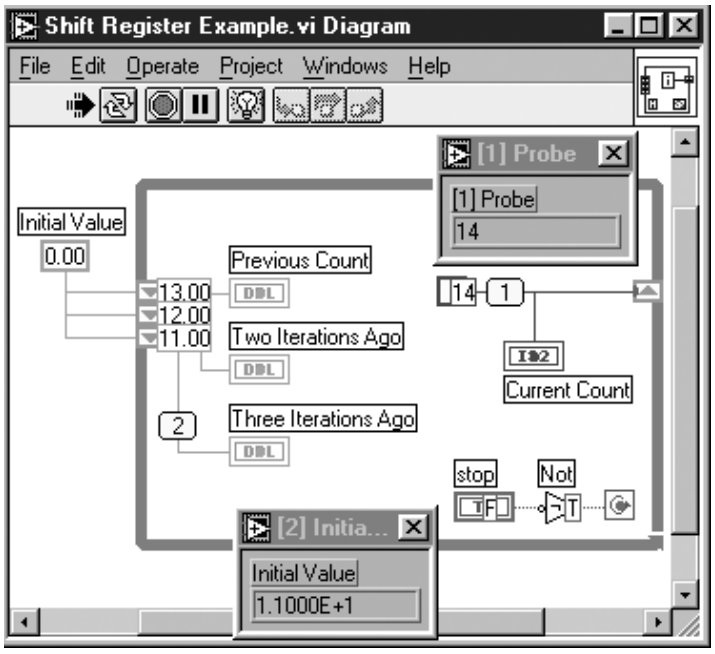
## به کارگیری پروپ

Probe Tool

با استفاده از پروپ می توان به بررسی مقادیر میانی در یک برنامه پرداخت. استفاده از پروب در مواردی که برنامه‌ی مورد نظر اجرا شده ولی نتایج دور از انتظار و سؤال برانگیز ایجاد می کند بسیار مفید است. به عنوان مثال یک صفحه‌ی نمودار بلوکی را در نظر بگیرید

که در آن چند عملیات محاسباتی انجام می گیرد. هر یک از اعمال ممکن است منجر به ایجاد داده های خروجی نادرست گردد. برای بررسی مقادیر مذکور می توانید یک المان نشان دهنده را بر روی سیم حامل داده ها ایجاد کنید و نتایج حاصل را به نمایش در آورید یا اینکه از یک پروب برای مشاهده‌ی نتایج به دست آمده در نقاط مختلف برنامه استفاده کنید. برای انجام این عمل، ابزار Probe Tool را از پالت Tools انتخاب نموده، سپس بر روی سیم مورد نظر کلیک کنید یا منوی کرکره‌ای را بر روی سیم مورد نظر باز نموده، گزینه‌ی Probe را انتخاب کنید. صفحه‌ی نمایش پروب به صورت یک پنجره‌ی شناور است.

در صورتی که برنامه در حال اجرا نباشد، صفحه‌ی نمایش پروب فاقد هرگونه مقدار است. به محض اجرای برنامه، مقادیر عبور داده شده توسط سیم مذکور در این صفحه به نمایش در می آیند. در شکل ۸-۵ محل و نحوه‌ی استفاده از Probe را ملاحظه می کنید.

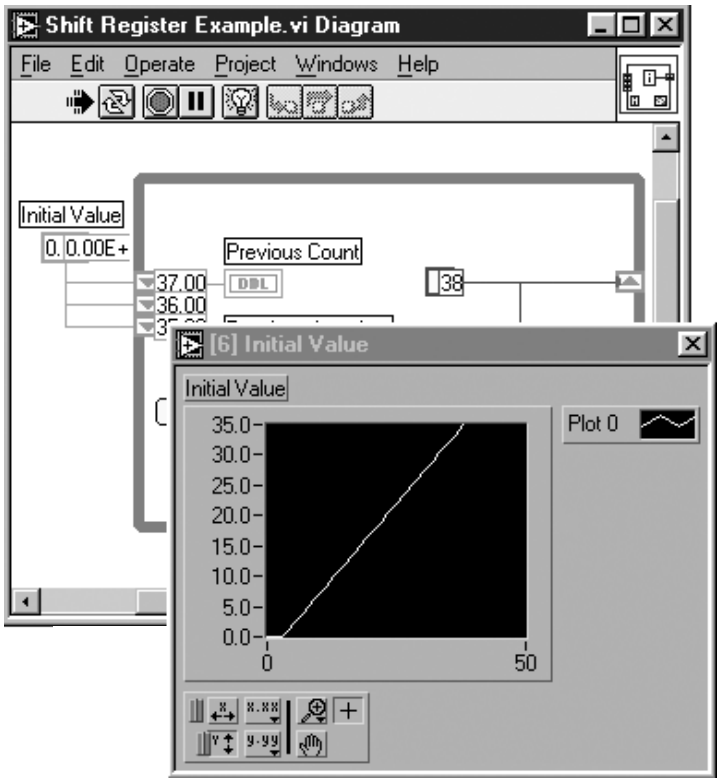



**شکل ۸-۵: نحوه‌ی استفاده از Probe**

برای بررسی بیشتر مقادیر، بهتر است پروب را به همراه دو مد اجرای متمایز و تک مرحله ای به کار برید. نرم‌افزار LabVIEW به هر پروب و سیم مربوط به آن، شماره‌ی منحصر به فردی اختصاص می دهد. این شماره ها به کاربر کمک می کند تا به راحتی آنها را در صفحه‌ی نمودار بلوکی ردیابی کند. اگر مسیر یک پروب را گم کردید و ندانستید که کدام پروب در ارتباط با کدام سیم می باشد، منوی کرکره‌ای را بر روی پروب یا سیم باز نموده، به ترتیب گزینه‌ی Find Wrie یا Find Probe را انتخاب کنید. در این حالت سیم یا پروب مورد نظر از سایر المان ها متمایز می شود.

شایان ذکر است که ایجاد تغییر در مقادیر داده ها توسط پروب امکان‌پذیر نیست. به عبارت دیگر توسط پروب تنها می توان مقادیر داده های عبور کننده از سیمها را مشاهده کرد.

همچنین می توانید صفحه‌ی نمایش دهنده را به همراه پروب به کار برید. برای انجام این عمل، منوی کرکره‌ای را بر روی سیم مورد نظر باز نموده، گزینه‌ی Custom Probe را انتخاب کنید. سپس صفحه‌ی نمایش دهنده‌ی مورد نظر را که قصد دارید نتایج به دست آمده توسط پروب را توسط آن ملاحظه کنید انتخاب نمایید. به عنوان مثال برای به نمایش در آوردن نمو یک متغیر در حلقه می توانید از یک نمودار که نوعی المان نشان دهنده است استفاده کنید. در شکل ۹-۵ ملاحظه می کنید که نتایج حاصل از پروب بر روی یک نمودار به نمایش در آمده است.




**شکل ۹-۵: به نمایش در آوردن نتایج حاصل از پروب بر روی نمودار**

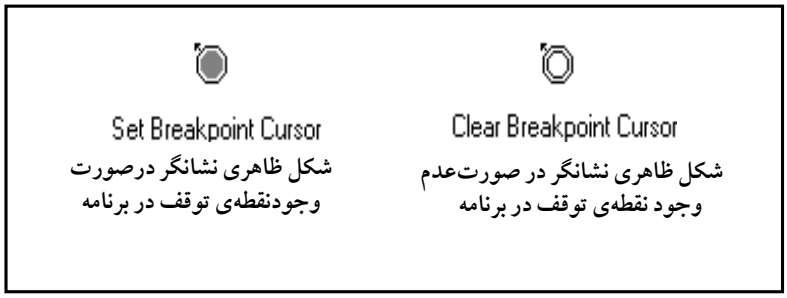
## قرار دادن نقاط توقف در برنامه

نگران نباشید! گنجاندن نقطه‌ی توقف در برنامه باعث توقف مطلق برنامه نمی شود. این نقطه تنها اجرای برنامه را به صورت موقت به تعویق می‌اندازد تا شما بتوانید به راحتی به رفع اشکال یا بررسی برنامه بپردازید. در صورتی که قصد داشته باشید تا ورودیهای یک برنامه، گره یا سیم را در خلال اجرای برنامه بررسی کنید، به کارگیری این نقاط بسیار مفید و کمکی خواهد بود.

به محض رسیدن اجرای برنامه به نقاط توقف در صفحه‌ی نمودار بلوکی، دکمه‌ی Pause به صورت خودکار فعال می شود. در این حالت می توانید اجرای تک مرحله ای برنامه را مورد بررسی قرار دهید یا برای مشاهده‌ی مقادیر داده ها در سیم از پروب استفاده کنید و حتی مقادیر المان های موجود در صفحه‌ی پانل را تغییر دهید. برای ادامه‌ی اجرای برنامه کافی است یکی از دو دکمه‌ی Pause یا Run را فشار دهید.

Breakpoint Tool

برای ایجاد یک نقطه‌ی توقف، ابتدا ابزار Breakpoint Tool را از پالت Tools انتخاب نموده، سپس بر روی المانی از صفحه‌ی نمودار بلوکی که قصد توقف برنامه در آن نقطه را دارید کلیک کنید. کلیک کردن مجدد بر روی همان المان باعث حذف نقطه‌ی توقف می گردد. شکل ظاهری نشانگر Breakpoint مشخص کننده‌ی وجود یا عدم وجود آن در صفحه‌ی نمودار بلوکی است.




**شکل ۱۰-۵**

بسته به اینکه نقاط توقف درچه محلی از صفحه‌ی نمودار بلوکی قرار گیرند، رفتار متفاوتی از خود نشان می دهند.

- در صورت قرارگرفتن نقطه‌ی توقف بر روی صفحه‌ی نمودار بلوکی، یک حاشیه‌ی قرمزرنگ در اطراف صفحه ظاهر می‌شود و توقف هنگامی صورت می گیرد که اجرای بلوک خاتمه یابد.
- در صورت قرارگرفتن نقطه‌ی توقف بر روی یک گره، یک حاشیه‌ی قرمزرنگ در اطراف گره ظاهر شده، دقیقاً قبل از اجرای گره، برنامه متوقف می شود.
- در صورت قرارگرفتن نقطه‌ی توقف بر روی یک سیم، یک گلوله‌ی قرمزرنگ بر روی سیم ظاهر می شود. اگر بر روی سیم از پروب استفاده شده باشد پروب نیز توسط یک حاشیه‌ی قرمزرنگ احاطه می گردد. در این حالت پس از عبور داده ها از سیم، توقف صورت می گیرد.

زمانی که به دلیل وجود نقطه‌ی توقف، در یک برنامه وقفه ای ایجاد می گردد، صفحه‌ی نمودار بلوکی در پنجره‌ی پیش زمینه قرار می گیرد و المانی که باعث ایجاد وقفه شده یا به عبارت دیگر المانی که نقطه‌ی توقف بر روی آن قرار گرفته است، توسط حاشیه‌ی خط چین متحرک از سایر المان ها متمایز می شود. نقاط توقف به همراه برنامه‌ی مربوط ذخیره شده، ولی تنها در هنگام اجرای برنامه فعال می شوند.

با انتخاب گزینه‌ی VI Setup... >> Execution Options >> Suspend When Called امکان فعال و غیر فعال کردن نقاط توقف وجود دارد. یادآوری می کنیم که گزینه‌ی VI Setup... یکی از گزینه های موجود در منوی کرکره‌ای آیکن برنامه درصفحه‌ی پانل است. انتخاب گزینه‌ی Suspend When Called سبب می شود تا نقاط توقف در برنامه فعال شده و آن را متوقف کنند. در صورتی که یک زیربرنامه دو مرتبه در یک صفحه‌ی نمودار بلوکی فراخوانی شده باشد، اجرای هر دو زیربرنامه متوقف می گردد. در فصل ۱۳ در مورد گزینه‌ی VI Setup... مطالب بیشتری خواهید آموخت.

در صورتی که قصد دارید که نقطه‌ی توقف، تنها در مواردی خاص در زیربرنامه باعث ایجاد وقفه گردد، نقاط توقف را با استفاده از گزینه‌ی SubVI Node Setup... در زیربرنامه قرار دهید. برای دستیابی به این گزینه منوی کرکره‌ای

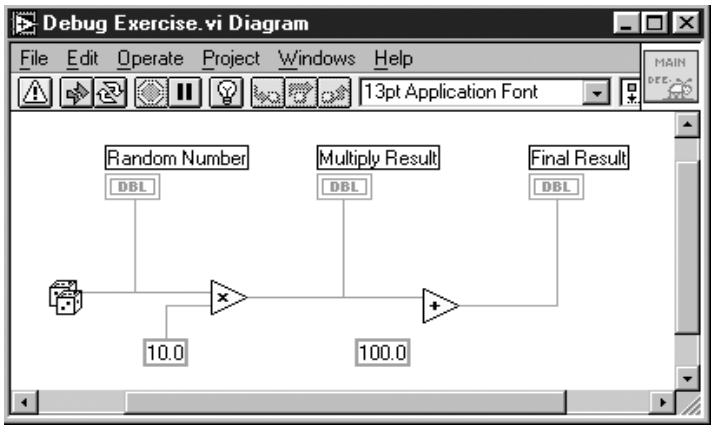
را در صفحه‌ی نمودار بلوکی و بر روی آیکن برنامه‌ای که این زیربرنامه را فراخوانی نموده است باز کنید.

## تمرین ۱-۵ : رفع اشکال برنامه

در این تمرین به رفع اشکال یک برنامه‌ی معیوب می پردازیم. سپس روشهای دیگر رفع اشکال برنامه نظیر استفاده از مدهای اجرای متمایز و تک مرحله ای و به کارگیری پروب را مورد بررسی قرار می دهیم.

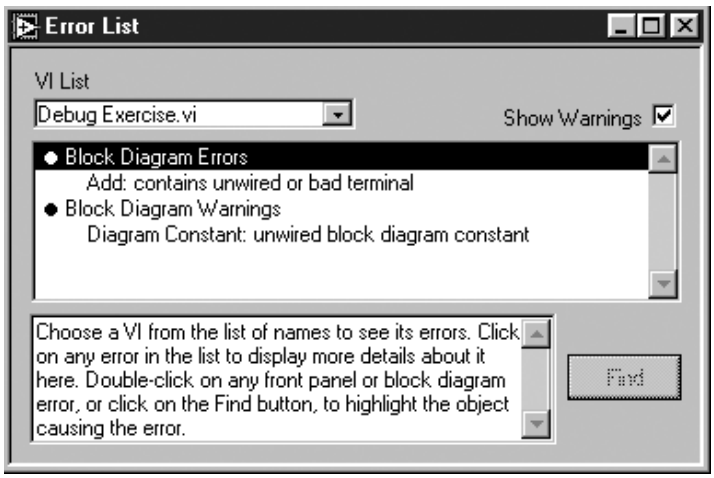
۱- برنامه‌ی Debug Exercise.vi را از مسیر LabVIEW\Exercise\Ch5.llb باز کنید.

۲- به صفحه‌ی نمودار بلوکی تغییر وضعیت دهید. در این حالت ملاحظه می کنید که دکمه‌ی Run به صورت پیکان شکسته ظاهر می‌شود و بیانگر معیوب بودن برنامه است. ابتدا علت عدم اجرای این برنامه را یافته، سپس برنامه را به گونه ای اصلاح کنید تا به اجرا در آید.




**شکل ۱۱-۵: صفحه‌ی نمودار بلوکی در برنامه‌ی معیوب Debug Exercise.vi**

۳- بر روی دکمه‌ی Run که به صورت پیکان شکسته ظاهر شده است کلیک کنید. در این حالت پنجره‌ی محاوره ای Error List ظاهر می گردد که در آن خطاهای موجود در این برنامه نشان داده می شود.




**شکل ۱۲-۵: پنجره‌ی محاوره‌ای Error List در برنامه‌ی معیوب Debug Exercise.vi**

۴- بر روی خطای «Add: contains unwired or bad terminal» کلیک کنید. در این حالت پنجره‌ی Error List اطلاعات بیشتری در مورد این خطا در اختیارتان قرار می‌دهد. حال برای یافتن منشأ ایجاد آن بر روی این خطا در پنجره‌ی Error List دو بار کلیک کنید و یا دکمه‌ی Find را فشار دهید. برای کمک به تشخیص محل المان یا دستوری که باعث ایجاد این خطا شده است، المان یا دستور مذکور متمایز می‌شود.

۵- سیم متصل نشده در ترمینال آیکن دستور Add را به ثابت عددی 100.0 ارتباط دهید. در این حالت دکمه‌ی Run باید به حالت اولیه‌ی خود بازگردد. در غیر این صورت سیم‌کشی را مجدداً انجام دهید. قبل از انجام سیم‌کشی مجدد، دستور Edit >> Remove Bad Wires را برای حذف سیمهای منقطع اجرا کنید.

۶- به صفحه‌ی پانل بازگردید و چند بار برنامه را اجرا کنید.

۷- با استفاده از گزینه‌ی Windows >> Tile Left and Right، دو صفحه‌ی پانل و نمودار بلوکی را به گونه‌ای قرار دهید که همزمان قابل رؤیت باشند.

Execution Highlighting Button

حال مد اجرای متمایز را فعال سازید و با فشار دادن دکمه‌های مربوط به حالت اجرای تک مرحله‌ای، این برنامه را به همراه دو مد اجرایی مذکور به اجرا در آورید.

۸- جهت اجرای یک گره کافی است بر روی دکمه‌ی Step Over کلیک کنید. برای خاتمه دادن به اجرای صفحه‌ی نمودار بلوکی، دکمه‌ی Step Out را فشار دهید. توجه داشته باشید که پس از اجرای هر مرحله، داده‌های حاصل از همان مرحله بر روی صفحه‌ی پانل ظاهر می‌شوند.

Step Over Button

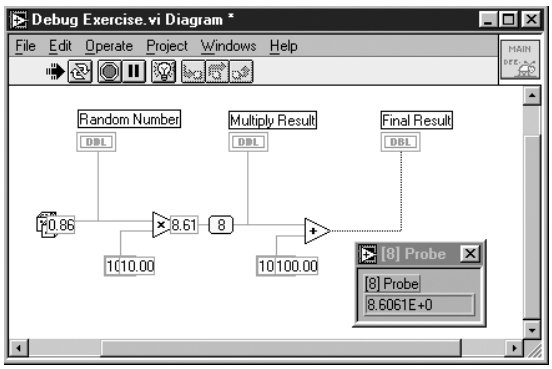
Step Into Button

Step Out Button

همان گونه که احتمالاً با ملاحظه‌ی صفحه‌ی نمودار بلوکی استنباط نموده‌اید روند اجرا، بدین ترتیب است که این برنامه ابتدا یک عدد تصادفی ایجاد می‌کند و پس از ضرب نمودن آن در عدد ثابت 10.0 مقدار حاصل ضرب را با عدد ثابت 100.0 جمع می‌کند. به یاد داشته باشید که با فشار دادن دکمه‌ی Pause می‌توانید از حالت اجرای تک مرحله‌ای خارج شوید و برنامه را به صورت عادی اجرا کنید.

۹- حال منوی کرکره‌ای را بر روی یکی از سیمها باز کنید و گزینه‌ی Probe را انتخاب نمایید.

۱۰- اکنون مجدداً برنامه را به صورت تک مرحله‌ای اجرا نموده، به نمایش داده‌های عبور داده شده توسط سیم بر روی صفحه‌ی نمایش دهنده‌ی پروب توجه کنید.




**شکل ۱۳-۵: اجرا نمودن برنامه‌ی Debug Exercise.vi به صورت تک‌مرحله‌ای و مشاهده‌ی نتایج به‌دست آمده با استفاده از پروب**

۱۱- حالت اجرای متمایز را با کلیک کردن بر روی دکمه‌ی مربوط به آن در نوار ابزار غیر فعال کنید.
۱۲- برنامه‌ی تکمیل شده را با انتخاب گزینه‌ی ...File >> Save As و با عنوان Debugged Exercise.vi در زیرفهرست LabVIEW\My Activity ذخیره کنید. با این روش، برنامه‌ی تکمیل شده بر روی برنامه‌ی اولیه بازنویسی نخواهد شد.
۱۳- با انتخاب گزینه‌ی File >> Close در صفحه‌ی پانل، برنامه را ببندید.

## ایجاد زیربرنامه

بسیاری از ویژگیها و قابلیتهای نرم‌افزار LabVIEW ناشی از خاصیت مدولار بودن آن است. بر همین اساس می توان یک برنامه‌ی پیچیده را ابتدا به بخشهای جداگانه تقسیم نمود و سپس با ایجاد چند زیربرنامه هر قسمت را برنامه نویسی کرد. همان گونه که قبلاً ذکر شد یک زیربرنامه در حقیقت برنامه‌ای است که توسط برنامه‌ی دیگری در سطح بالاتر فراخوانی شده است. یک زیربرنامه که دارای آیکن/کانکتور است و در صفحه‌ی نمودار بلوکی برنامه‌ی مرتبه‌ی بالاتر قرار گرفته است، مشابه Subroutine در زبانهای برنامه نویسی متداول نظیر C و BASIC می باشد که در برنامه‌ی اصلی فراخوانی می شود.

استفاده از یک برنامه به عنوان زیربرنامه در برنامه‌های مراتب بالاتر به شرطی امکان‌پذیر است که آیکن برنامه را ایجاد نموده و کانکتور مربوط به آن را نیز مشخص کرده باشید. برای قرار دادن یک برنامه به عنوان زیربرنامه در صفحه‌ی نمودار بلوکی برنامه مرتبه‌ی بالاتر، زیرپالت ...Functions >> Select a VI را انتخاب کنید. پس از انتخاب این گزینه یک پنجره‌ی محاوره‌ای باز می شود. در این پنجره می توانید هر یک از برنامه‌های موجود را انتخاب کنید. پس از انتخاب هر برنامه، آیکن آن در صفحه‌ی نمودار بلوکی برنامه‌ی مرتبه‌ی بالاتر قرار می گیرد.

یک برنامه نمی‌تواند خود را به طور مستقیم یا غیرمستقیم فراخوانی کند. این ویژگی را خاصیت اجرای بازگشتی[4] گویند که نرم‌افزار LabVIEW قادر به انجام آن نیست. بنابراین استفاده از یک برنامه در صفحه‌ی نمودار بلوکی خودش و یا صفحات زیربرنامه‌های آن امکان‌پذیر نیست.

قبل از به کارگیری یک برنامه به عنوان زیربرنامه باید راهی پیدا کنید تا بتوانید داده‌ها را از برنامه‌ی فرا خوانده شده دریافت نموده، همچنین اطلاعات و داده‌ها را به آن ارسال کنید. برای انجام این عمل باید المان‌های کنترل و نشان‌دهنده‌ی موجود در برنامه را به ترمینال‌های قرار گرفته بر روی کانکتور نسبت دهید. همچنین باید یک آیکن برای نمایش دادن برنامه‌ی مذکور طراحی کنید.

۵

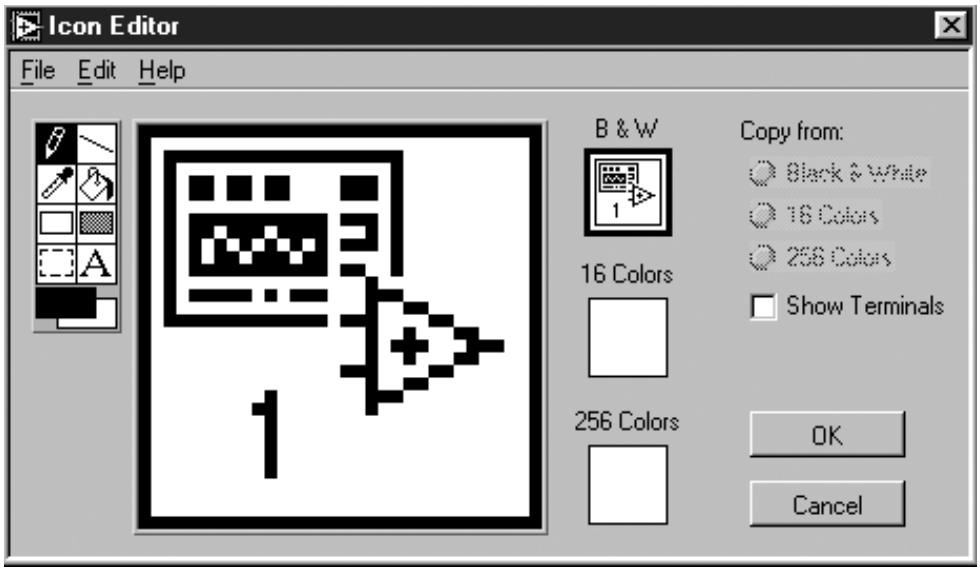
4- Recursive Execution

## طراحی آیکن

به‌کارگیری آیکن زیربرنامه برای به نمایش در آوردن آن در صفحه‌ی نمودار بلوکی برنامه‌ی فراخوان الزامی است. آیکن در حقیقت یک نماد گرافیکی از برنامه است. با انتخاب گزینه‌ی ...Edit Icon از منوی کرکره‌ای بر روی قاب آیکن می توان یک آیکن طراحی نمود. قاب آیکن در منتهی‌الیه سمت راست و در بالای صفحه‌ی پانل قرار گرفته است. دستیابی به منوی کرکره‌ای جهت ایجاد آیکن تنها در محیط Edit امکان پذیر است.

**شکل ۱۴-۵: نحوه‌ی انتخاب گزینه‌ی ...Edit Icon از منوی کرکره‌ای بر روی قاب آیکن**

برای دستیابی به پنجره‌ی Icon Editor به روش دیگر، بر روی قاب آیکن مورد نظر دو بار کلیک کنید تا پنجره‌ای مطابق شکل ۱۵-۵ ظاهر شود. برای طراحی آیکن دلخواه می توانید از ابزارهای موجود در این پنجره استفاده نمایید.




**شکل ۱۵-۵: پنجره‌ی Icon Editor**

در ادامه به بررسی ابزارهای موجود در این پنجره می پردازیم:

از این ابزار جهت ترسیم نقطه به نقطه استفاده می شود.

Pencil

از این ابزار جهت ترسیم خطوط مستقیم استفاده می شود. با پایین نگه داشتن کلید <Shift> می توان ترسیم خطوط را به صورت افقی، عمودی یا مورب انجام داد.

Line

از این ابزار جهت کپی برداری از رنگ صفحه‌ی پیش زمینه‌ی یک المان استفاده می شود. با پایین نگه داشتن کلید <Shift> می توان به کمک این ابزار رنگ پس‌زمینه‌ی یک المان را کپی برداری نمود.

Dropper

این ابزار جهت پر کردن یک ناحیه‌ی محصور شده با رنگ پیش زمینه به کار برده می شود.

Fill Bucket

از این ابزار جهت ترسیم یک مستطیل در صفحه‌ی پیش‌زمینه استفاده می گردد. با دوبار کلیک کردن بر روی این ابزار می توان آیکن را در صفحه‌ی پیش‌زمینه قرار داد. با استفاده از کلید <Shift> می توانید این ابزار مستطیل شکل را به صورت مکعب در آورید.

Rectangle

به کمک این ابزار می توان مستطیلی رسم نمود که اضلاع آن به رنگ پیش زمینه بوده، با رنگ پس زمینه پر شده باشد. با دوبار کلیک کردن بر روی این ابزار می توان آیکن را به رنگ پیش زمینه در آورده و آن را با رنگ پس‌زمینه پُر نمود.

Filled Rectangle

از این ابزار جهت انتخاب ناحیه‌ای از آیکن برای حرکت دادن و انجام تغییرات دیگر استفاده می شود.

Select

این ابزار برای وارد کردن متون نوشتاری که در طراحی آیکن مورد استفاده قرار می گیرند به کار می رود. برای تغییر در اندازه، شکل و رنگ فونت بر روی این ابزار دوبار کلیک کنید.

Text

Foreground/Background

این ابزار رنگ فعلی دو صفحه‌ی پیش‌زمینه و پس‌زمینه را نشان می‌دهد. با کلیک کردن بر روی هر یک از دو صفحه‌ی مذکور، یک پالت رنگ ظاهر می‌شود که می‌توان رنگ دلخواه را انتخاب نمود.

در سمت راست پنجره‌ی Icon Editor دو دکمه دیده می‌شود که در ادامه به تشریح آنها می‌پردازیم:

OK

با کلیک نمودن بر روی این دکمه می‌توان طراحی انجام شده را به عنوان آیکن مربوط به برنامه‌ی مورد نظر ذخیره نمود و به صفحه‌ی پانل بازگشت.

Cancel

کلیک کردن برروی این دکمه موجب می‌شود تا بدون ذخیره نمودن طراحی انجام شده، به صفحه‌ی پانل بازگردید.

بسته به نوع و قابلیت صفحه‌ی نمایشگر رایانه می‌توانید یک آیکن را در یکی از مدهای تک‌رنگ، ۱۶ رنگ یا ۲۵۶ رنگ طراحی کنید.

می‌توان آیکن‌های طراحی شده در مدهای مختلف را به طور جداگانه ذخیره نمود. همچنین می‌توانید یک آیکن را با استفاده از دکمه‌های موجود در بخش Copy from از حالت رنگی به تک‌رنگ و برعکس کپی کنید.

در صورتی که آیکن را در مد رنگی طراحی کرده باشید، طرح تکمیل شده در صفحه‌ی پانل نیز به صورت رنگی ظاهر می‌شود. واضح است که آیکن‌های رنگی در مانیتورهای تک‌رنگ قابل رؤیت نمی‌باشند. در صورت عدم طراحی آیکن، LabVIEW یک آیکن را به صورت پیش‌فرض در قاب آیکن به نمایش در می‌آورد.

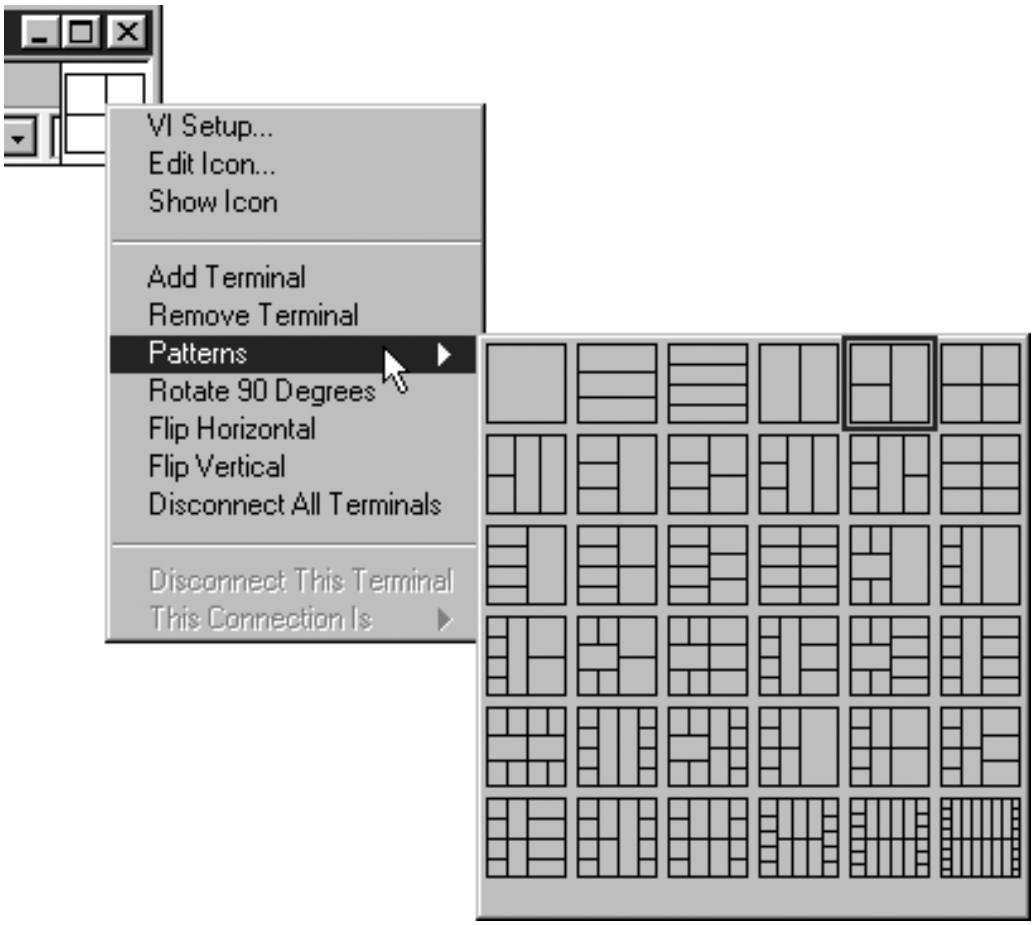
## انتساب ترمینال‌های کانکتور

قبل از اینکه بتوانید یک برنامه را به عنوان زیربرنامه به کار برید باید ترمینال‌های کانکتور آن را به المان‌های موجود در صفحه‌ی پانل نسبت دهید. به کارگیری کانکتور روشی برای عبور داده‌ها به/از یک زیربرنامه است.

انتساب ترمینال‌های کانکتور در محیط LabVIEW شبیه به تعریف پارامترهای Subroutine در زبانهای برنامه نویسی متنی است. منظور از انتساب کانکتور در یک برنامه، نسبت دادن المان‌های کنترل و نشان‌دهنده‌ی موجود در صفحه‌ی پانل آن به ترمینال‌های کانکتور است.

جهت تعیین کانکتور، منوی کرکره‌ای را بر روی قاب آیکن در صفحه‌ی پانل باز نموده، سپس گزینه‌ی Show Connector را انتخاب کنید. برای مشاهده‌ی مجدد آیکن، گزینه‌ی Show Icon را از همین منو انتخاب نمایید.

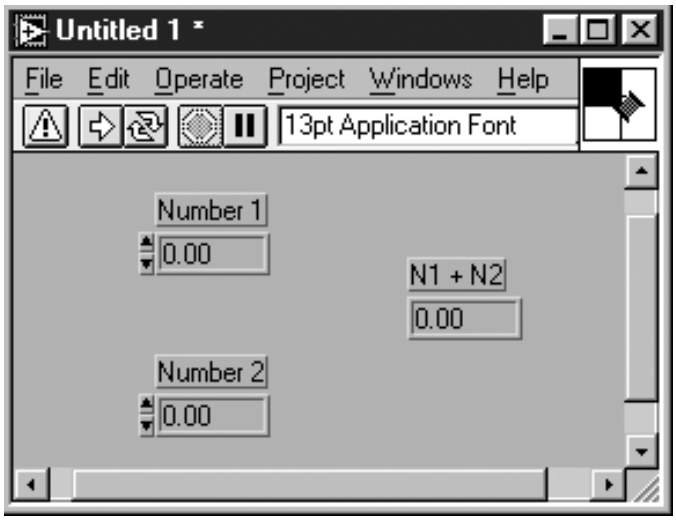
نرم افزار LabVIEW با توجه به تعداد المان های کنترل و نشان دهنده‌ی موجود در صفحه‌ی پانل، به صورت پیش فرض یک الگوی کانکتور برای برنامه پیشنهاد می دهد. در صورتی که قصد داشته باشید از الگوی دیگری استفاده کنید می توانید آن را از پالت Patterns در منوی کرکره‌ای کانکتور انتخاب نمایید. همچنین با استفاده از گزینه های موجود در این منو می توانید الگوی انتخاب شده را در صورتی که فاقد جهت های معمول و متداول است، چرخانده یا وضعیت آن را تغییر دهید. در شکل ۱۶-۵ گزینه های موجود در منوی کرکره ای کانکتور را به همراه الگوهای تعبیه شده در پالت Patterns ملاحظه می کنید.




**شکل ۱۶-۵: گزینه های موجود در منوی کرکره‌ای کانکتور به همراه الگوهای تعبیه شده در پالت Patterns**

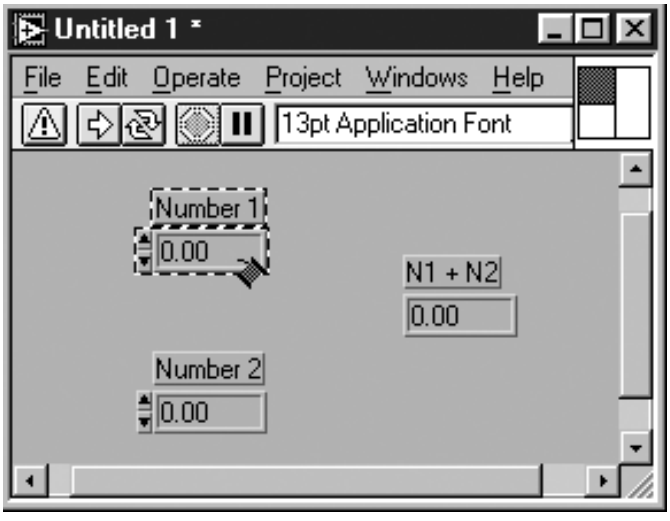
مراحل زیر را برای انتساب ترمینال کانکتور به یکی از المان های کنترل یا نشان دهنده دنبال کنید:

۱- بر روی یکی از ترمینال های کانکتورکلیک کنید. در این حالت نشانگر ماوس به ابزار Wiring Tool تغییر شکل می‌دهد و ترمینال انتخاب شده به رنگ سیاه در می آید. این مطلب در شکل ۱۷-۵ نشان داده شده است.




شکل ۱۷-۵

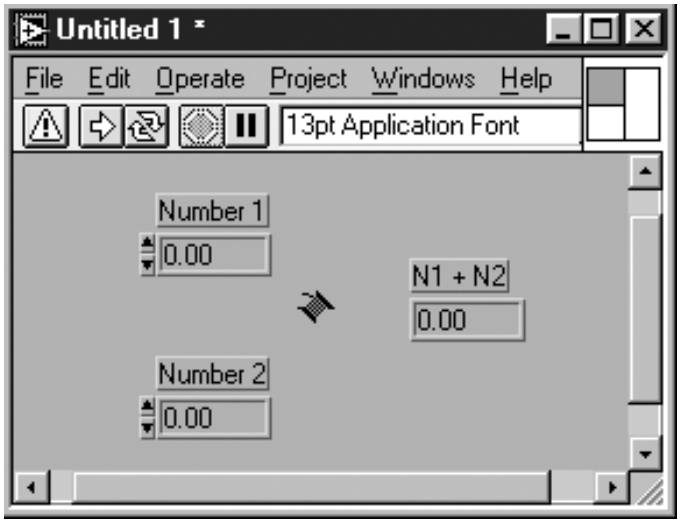
۲- سپس بر روی المان کنترل یا نشان‌دهنده‌ای که قصد دارید به ترمینال انتخاب شده نسبت داده شود کلیک کنید. همان‌طور که در شکل ۱۸-۵ مشاهده می کنید المان انتخاب شده توسط حاشیه‌ی خط چین متحرک احاطه می گردد.




شکل ۱۸-۵

۳- حال در ناحیه‌ای فاقد المان بر روی صفحه‌ی پانل کلیک کنید تا حاشیه‌ی خط چین متحرک ناپدید گردد. در این حالت ترمینال نسبت داده شده به صورت رنگی (در این حالت به رنگ نارنجی) ظاهر

می شود. این رنگ، نشانگر نوع داده‌ی اختصاص داده شده به المان مورد نظر است.




**شکل ۱۹-۵: تکمیل عملیات انتساب ترمینال ورودی کانکتور به المان** Number 1

در صورتی که ترمینال به رنگ سیاه یا سفید ظاهر گردد بدین مفهوم است که اعمال فوق را به درستی انجام نداده‌اید. در صورت لزوم می توانید مراحل فوق را مجدداً تکرار کنید. همان طور که قبلاً اشاره شد برای یک برنامه می توانید حداکثر ۲۸ ترمینال در نظر بگیرید.

می‌توانید ترتیب اجرای دو مرحله‌ی ۱ و۲ را عوض کنید. بدین مفهوم که ابتدا المان کنترل یا نشان‌دهنده را از صفحه‌ی پانل انتخاب نموده، سپس بر روی ترمینال مورد نظر در الگوی کانکتور کلیک کنید.

## گزینه‌ی Create SubVI

در برخی موارد تا زمانی که اقدام به ایجاد یک زیربرنامه در برنامه‌ی اصلی ننموده‌اید، ممکن است متوجه نباشید که باید بخشی از دستورها را با یک زیربرنامه جایگزین می کردید. خوشبختانه در نرم‌افزار LabVIEW امکاناتی وجود دارد که می توان با تبدیل بخشی از دستورها و کدهای موجود در برنامه، یک زیربرنامه ایجاد نمود. برای انتخاب بخش مورد نظر ابتدا با استفاده از ابزار Positioning Tool در اطراف قسمتی از برنامه که قصد جایگزینی آن را با یک زیربرنامه دارید یک مستطیل محاط کرده، سپس گزینه‌ی Edit >> Create SubVI را انتخاب نمایید. ملاحظه می کنید که بخش مذکور با یک زیربرنامه جایگزین می گردد. پس از اجرای این دستور، زیربرنامه به همراه آیکن مربوط و سیم کشی دقیق به نمایش در می آید.

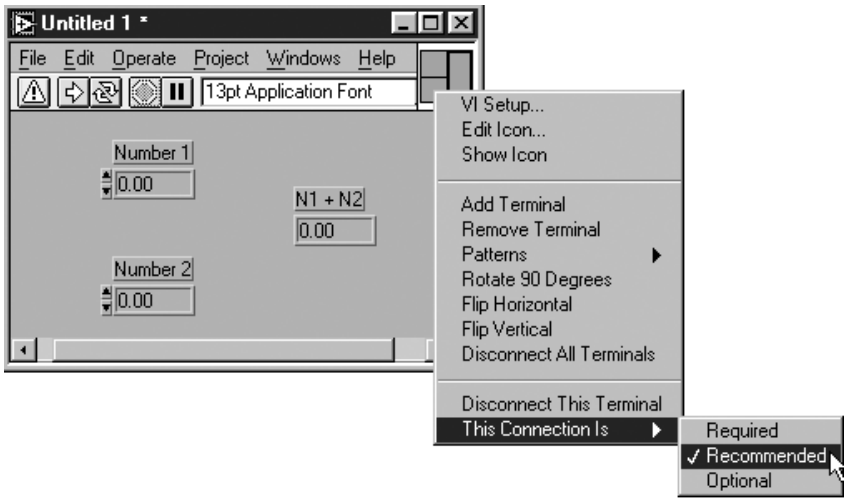
برای مشاهده‌ی صفحه‌ی پانل زیربرنامه‌ی جدید، همچنین تغییر آیکن و مشاهده‌ی اتصالات آن بر روی آیکن زیربرنامه دو بار کلیک کنید. به هنگام استفاده از گزینه‌ی Create SubVI با احتیاط عمل کنید؛ زیرا ممکن است به نتایجی دور از انتظار دست یابید. ویژگیهای این دستور را در فصل ۱۳ مورد بررسی قرار خواهیم داد.

## پنجره‌ی Help در مورد زیربرنامه‌ها

اگر درصفحه‌ی نمودار بلوکی، پنجره‌ی Help را بر روی یک گره‌ی زیربرنامه فعال کنید، در مورد زیربرنامه، الگوی سیم‌کشی و اتصالات آن توضیحاتی ظاهر می‌گردد. در این پنجره ملاحظه می‌کنید که ورودیها در سمت راست و خروجیها در سمت چپ نمایش داده می‌شوند. برای مشاهده‌ی پارامترها و توضیحات برنامه، پنجره‌ی Help را بر روی قاب آیکن فعال کنید. در بخش بعدی نحوه‌ی ویراستاری و ایجاد تغییر در مورد توضیحات ارائه شده در پنجره‌ی Help را فرا می‌گیرید.

در صورتی که یک ورودی الزامی را در زیربرنامه سیم‌کشی نکرده باشید، LabVIEW به صورت خودکار این مطلب را تشخیص می‌دهد و تا زمانی که این ارتباط را برقرار نکرده‌اید اجرای برنامه‌ی اصلی را آغاز نمی‌کند. همان گونه که در مورد دستورها اشاره شد، در مورد زیربرنامه‌ها نیز می‌توانید از ورودیهای الزمی، پیشنهادی و اختیاری استفاده نمایید. در صورتی که یک ورودی از نوع الزامی باشد، بدون سیم‌کشی دقیق آن قادر نخواهید بود که برنامه را به عنوان زیربرنامه استفاده کنید. اگر یک ورودی یا خروجی پیشنهادی را سیم‌کشی نکرده باشید، برنامه اجرا می‌گردد و در صورتی که گزینه‌ی Show Warnings در پنجره‌ی Error List فعال باشد به شما هشدار داده می‌شود که یکی از اتصالات پیشنهادی سیم‌کشی نشده است. اگر یک ورودی به صورت اختیاری باشد در مورد سیم‌کشی یا عدم سیم‌کشی آن محدودیتی وجود ندارد.

برای مشاهده‌ی نوع اتصالات از نظر الزامی، پیشنهادی یا اختیاری، منوی کرکره‌ای را بر روی ترمینال کانکتور باز نموده، گزینه‌ی This Connection Is را انتخاب کنید. در زیرمنوی بعدی وجود علامت «✓» درکنار هر یک از گزینه‌های Required، Recommended یا Optional وضعیت فعلی آنها را تعیین می‌کند.




**شکل ۲۰-۵: روش مشاهده‌ی نوع اتصالات کانکتور**

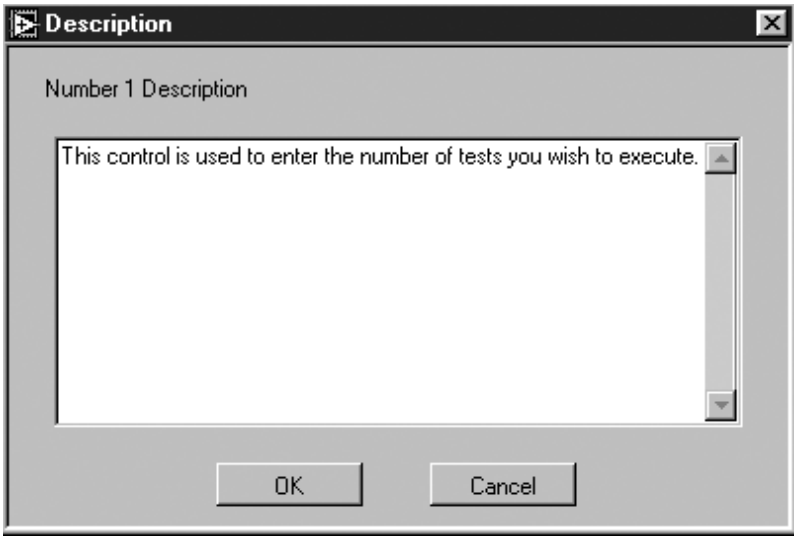
در پنجره‌ی Detailed Help، اتصالات الزامی و پیشنهادی به ترتیب با حروف پهن و ساده نشان داده می‌شوند. اتصالات اختیاری نیز به رنگ خاکستری ظاهر می‌شوند. در صورت استفاده از پنجره‌ی Simple Help، اتصالات اختیاری به نمایش در نمی‌آیند.

## اضافه کردن توضیح در برنامه

مستندسازی یا اضافه کردن توضیح در برنامه از اهمیت بالایی برخودار است. زیرا با این عمل، کاربران دیگر به راحتی می توانند عملکرد این برنامه ها را درک کنند. با وجود توضیحات در مورد هر برنامه فراموش نخواهید کرد که به عنوان مثال چرا چنین کاری را انجام داده اید و یا اینکه برنامه چگونه عمل می کند. این بخش به ذکر روشهای اضافه کردن توضیح در مورد برنامه اختصاص یافته است.

## ارائه‌ی توضیح در مورد المان ها

در صورتی که قصد دارید تا در مورد یکی از المان های کنترل یا نشان دهنده‌ی موجود در برنامه توضیح دهید، ابتدا منوی کرکره ای را بر روی المان مورد نظر باز کرده، سپس گزینه‌ی ...Description >> Data Operations را انتخاب نمایید. ملاحظه می کنید که یک پنجره‌ی محاوره ای با عنوان Description ظاهر می شود. سپس متن یا توضیح مورد نظر را در پنجره وارد کرده، برای ذخیره‌ی آن بر روی دکمه‌ی OK کلیک کنید. در شکل ۲۱-۵ یک نمونه از این پنجره‌ی محاوره ای را ملاحظه می کنید.

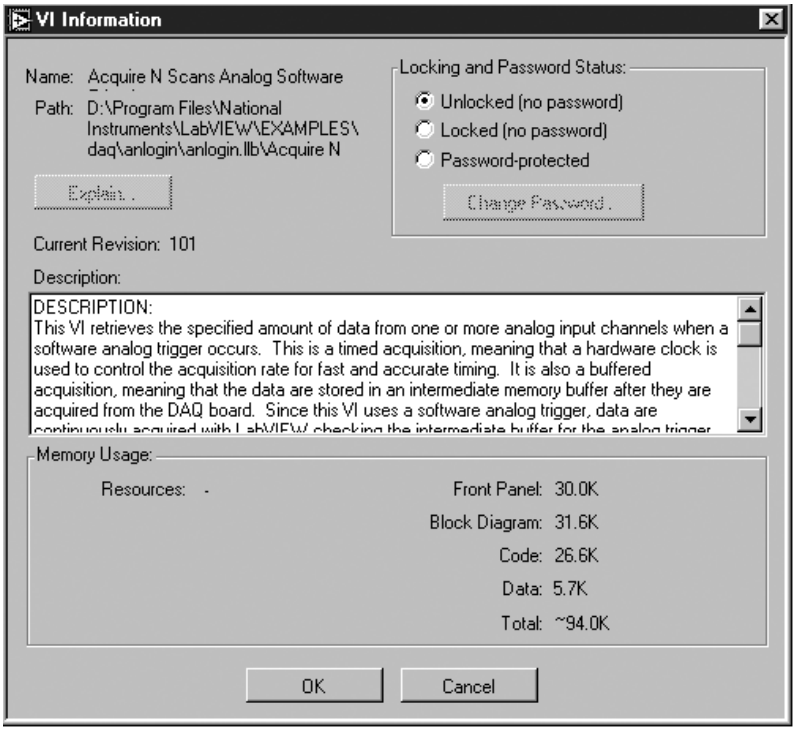



**شکل ۲۱-۵: پنجره‌ی محاوره‌ای Description**

پس از انتخاب گزینه‌ی ...Description از منوی کرکره ای المان، همین متن در مورد المان مورد نظر ظاهر می شود. در صورت قرار گرفتن ماوس بر روی این المان در صفحه‌ی پانل، همین توضیحات در پنجره‌ی Help نیز به نمایش در می آیند. استفاده از این قابلیت برای ارائه‌ی توضیح در مورد چگونگی به کارگیری دستورها و زیربرنامه ها در صفحه‌ی نمودار بلوکی بسیار مفید است. شایان ذکر است که توضیحات مندرج در پنجره‌ی Description در مورد دستورها و زیربرنامه‌ها، در پنجره‌ی Help به نمایش در نمی آیند و برای مشاهده‌ی این توضیحات حتماً باید از پنجره‌ی Description استفاده کنید. برای ایجاد پنجره‌ی Online Help در مورد یک برنامه، بهتر است در مورد تمامی المان های آن در پنجره‌ی Description توضیح دهید.

## ارائه‌ی توضیح در مورد برنامه

LabVIEW روشی ساده برای ارائه‌ی توضیح در مورد کل برنامه در پیش پای شما قرار می دهد. با انتخاب گزینه‌ی ...Windows >> Show VI Info مطابق شکل ۲۲-۵ یک پنجره‌ی محاوره‌ای با عنوان VI Information برای برنامه‌ی فعلی باز می شود.




**شکل ۲۲-۵: پنجره‌ی محاوره‌ای VI Information**

با استفاده از این پنجره‌ی محاوره‌ای قادر به انجام موارد زیر خواهید بود:

- می توانید در مورد برنامه توضیح دهید. ناحیه‌ی مخصوص وارد کردن توضیح، دارای یک نوار پیمایش است که به کمک آن می توانید متن ها و توضیحات طولانی را به راحتی مشاهده کنید یا آنها را تغییر دهید. در صورت استفاده از برنامه به عنوان زیربرنامه، هنگامی که نشانگر ماوس بر روی آیکن مربوط در صفحه‌ی نمودار بلوکی برنامه‌ی سطح بالاتر قرار گیرد، این توضیحات در صفحه‌ی Help به نمایش در خواهند آمد.
- با فشار دادن دکمه‌ی Explain فهرستی از تغییرات ایجاد شده در برنامه از زمان آخرین عملیات ذخیره‌ی برنامه را مشاهده خواهید کرد.
- می توانید مسیر یا به عبارت دیگر محل ذخیره شدن برنامه را مشاهده کنید.
- می توانید مقدار فضای اشغال شده توسط این برنامه را ملاحظه کنید. در بخش Memory Usage فضای اشغال شده‌ی دیسک سخت نشان داده می شود. در این بخش تنها فضای اشغال شده توسط برنامه

نشان داده شده است. در پنجره‌ی VI Information، رؤیت مقدار فضای اشغال شده توسط زیربرنامه های این برنامه امکان پذیر نیست.

با انتخاب گزینه‌ی ...Description << Data Operations در منوی کرکره‌ای هر المان می‌توان در مورد المان‌های مجزا توضیح داد. حال آن که انتخاب نمودن گزینه‌ی ...Show VI Info << Windows منجر به ظاهر شدن پنجره‌ای می‌گردد که در آن توضیحات کلی در مورد عملکرد برنامه ارائه می‌شود.

## مطالب مختصری در مورد نحوه‌ی چاپ

در نرم‌افزار LabVIEW سه روش چاپ وجود دارد که در موارد مختلف می‌توانید برای چاپ برنامه از آنها استفاده کنید:

- با انتخاب گزینه‌ی ...File << Print Window می‌توانید محتویات پنجره‌ی فعال را چاپ کنید.
- با انتخاب گزینه‌ی ...File << Print Documentation می‌توانید اطلاعاتی در مورد صفحات پانل و نمودار بلوکی، زیربرنامه های موجود در برنامه و... را چاپ نمایید. با انتخاب این گزینه، یک پنجره‌ی محاوره‌ای باز می‌شود که در آن می‌توانید فرمت دلخواه را برای چاپ انتخاب کنید. این روش چاپ برنامه را در فصل ۱۴ مورد بررسی قرار خواهیم داد.
- با انتخاب گزینه‌ی Operate << Print at Completion می‌توانید از قابلیتهای چاپ در نرم‌افزار LabVIEW استفاده کنید و صفحه‌ی پانل را تحت کنترل خود در آورید. با انتخاب این گزینه، محتویات صفحه‌ی پانل پس از اتمام اجرای آن چاپ می‌گردد. در صورتی که از یک برنامه به عنوان زیربرنامه استفاده شده باشد، عمل چاپ پس از اجرای زیربرنامه و قبل از برگشتن به برنامه‌ی فراخوان اصلی انجام می‌گیرد. این گزینه و قابلیتهای آن را در فصل ۱۴ مورد بررسی قرار خواهیم داد.

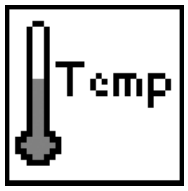
## تمرین ۲-۵ : ایجاد یک زیربرنامه

حال زمان آن رسیده است که مجدداً به رایانه‌ی خود مراجعه کنید. در این تمرین قصد داریم تا برنامه‌ی Thermometer.vi را که در فصل قبل ایجاد نمودیم به یک زیربرنامه تبدیل کنیم. بدین ترتیب قادر خواهیم بود این (زیر) برنامه را در صفحه‌ی نمودار بلوکی برنامه های دیگر مورد استفاده قرار دهیم.

۱- برنامه‌ی Thermometer.vi را که در تمرین ۲-۴ ایجاد نمودید، باز کنید. همان طور که به شما پیشنهاد دادیم اگر این برنامه را در زیرفهرست LabVIEW\My Activity ذخیره کرده باشید دستیابی به آن بسیار ساده است. در صورتی که این برنامه را ذخیره نکرده‌اید می‌توانید آن را در مسیر LabVIEW\Exercise\Ch4.llb بیابید.

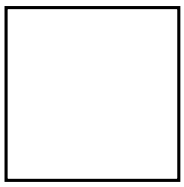
۲- برای این برنامه یک آیکن طراحی کنید. برای انجام این عمل منوی کرکره‌ای را بر روی قاب آیکن این برنامه در صفحه‌ی پانل باز کرده، گزینه‌ی ...Edit Icon را برای وارد شدن به محیط Icon Editor انتخاب

نمایید. به یاد داشته باشید که برای وارد شدن به محیط مذکور نیز می توانید دو بار بر روی قاب آیکن کلیک کنید. با استفاده از ابزارهای موجود در این پنجره یک آیکن مطابق شکل ۲۳-۵ ایجاد کنید و با فشار دادن دکمه‌ی OK به برنامه‌ی اصلی بازگردید.




**شکل ۲۳-۵: آیکن برنامه‌ی Thermometer.vi**

۳- حال منوی کرکره‌ای را بر روی قاب آیکن باز کنید و گزینه‌ی Show Connector را جهت ایجاد کانکتور انتخاب نمایید. به دلیل اینکه بر روی صفحه‌ی پانل، تنها یک المان نشان دهنده وجود دارد، الگوی کانکتور این برنامه باید فقط دارای یک ترمینال و به صورت یک مربع سفیدرنگ باشد. این کانکتور در شکل ۲۴-۵ نشان داده شده است.

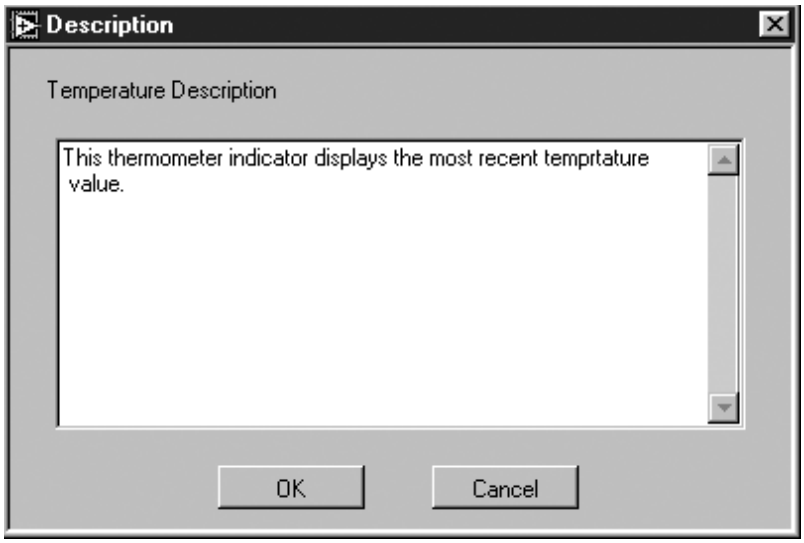


**شکل ۲۴-۵: کاتکتور برنامه‌ی Thermometer.vi**

۴- اکنون ترمینال کانکتور را به المان نشان دهنده‌ی Temperature نسبت دهید. برای انجام این عمل، توسط ابزار Wiring Tool بر روی ترمینال کانکتور کلیک کنید. ملاحظه می کنید که ترمینال به رنگ سیاه در می آید. سپس بر روی نشان دهنده‌ی Temperature کلیک کنید.

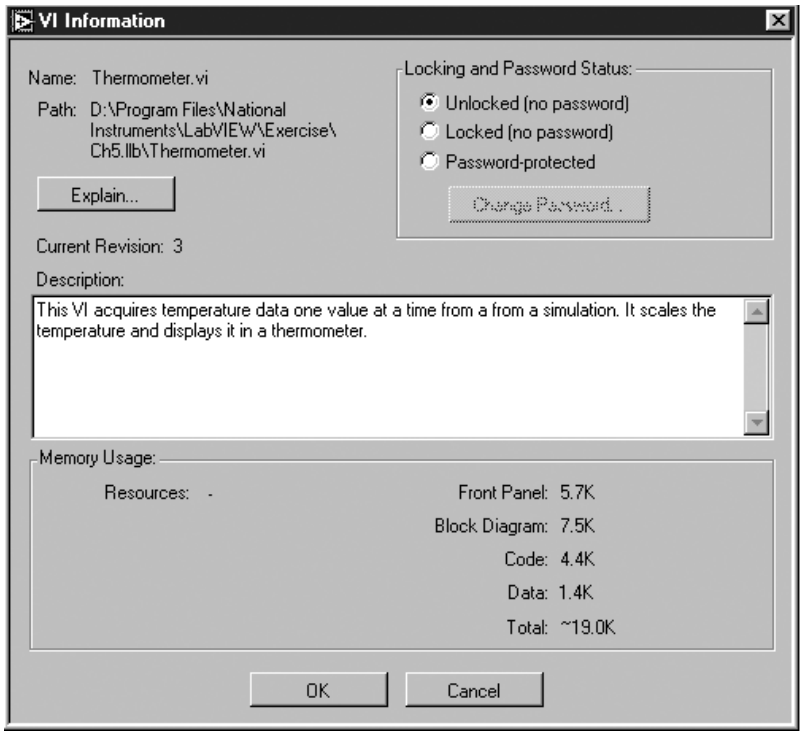
مشاهده می کنید که این المان توسط یک حاشیه‌ی خط چین متحرک احاطه می شود. حال در ناحیه‌ای فاقد المان بر روی صفحه‌ی پانل کلیک کنید. در این حالت حاشیه‌ی خط چین، ناپدید می‌شود و رنگ ترمینال انتخاب شده از سیاه به نارنجی تغییر می‌یابد و بیانگر آن است که یک المان با نوع نمایش عدد صحیح به این ترمینال نسبت داده شده است. برای مشاهده‌ی قاب آیکن، گزینه‌ی Show Icon را از منوی کرکره‌ای قاب آیکن انتخاب کنید.

۵- در این مرحله در مورد المان Temperature توضیح دهید. برای انجام این عمل، منوی کرکره‌ای را بر روی این المان باز نموده، سپس گزینه‌ی ...Description << Data Operations را انتخاب نمایید. متن مندرج در پنجره‌ی نشان داده شده در شکل ۲۵-۵ را تایپ نموده، سپس برای ذخیره‌ی این توضیحات بر روی دکمه‌ی OK کلیک کنید.




**شکل ۲۵-۵: توضیحات وارد شده در پنجره‌ی Description در مورد عملکرد المان Temperature**

۶- با انتخاب گزینه‌ی ...Show VI Info << Windows، در مورد برنامه‌ی Thermometer.vi توضیح دهید. برای انجام این عمل توضیحات مندرج در پنجره‌ی نشان داده شده در شکل ۲۶-۵ را تایپ کرده، سپس دکمه‌ی OK را فشار دهید.




**شکل ۲۶-۵: توضیحات وارد شده در پنجره‌ی VI Information در مورد عملکرد برنامه‌ی Thermometer.vi**

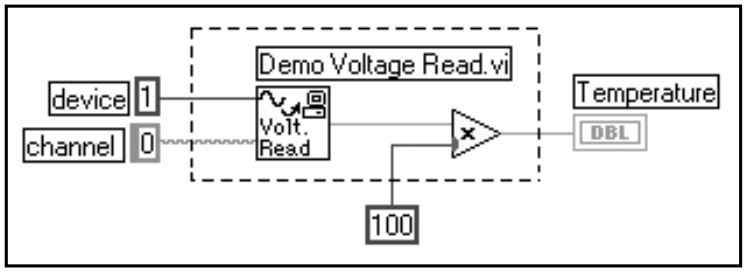
۷- حال با انتخاب گزینه‌ی Help << Show Help، پنجره‌ی Help را فعال سازید. هنگامی که نشانگر ماوس را بر روی آیکن این برنامه قرار می دهید، اتصالات ورودی و خروجی و توضیحاتی در مورد عملکرد برنامه را در این پنجره ملاحظه می کنید. در صورتی که هیچ برچسبی برای المان

نشان‌دهنده‌ی Temperature در برنامه تعیین نکرده باشید، این المان در پنجره‌ی Help نیز فاقد برچسب خواهد بود. حال منوی کرکره‌ای را بر روی المان نشان‌دهنده‌ی Temperature باز نموده، سپس گزینه‌ی Data Operations >> Description... را برای مشاهده‌ی توضیحات وارد شده در مورد این المان انتخاب کنید.

۸- در صورتی که یک چاپگر در اختیار دارید با انتخاب گزینه‌ی File >> Print Window... محتویات صفحه‌ی فعال را چاپ کنید. در این روش چاپ، با انتخاب کردن هر یک از صفحات پانل یا نمودار بلوکی به عنوان صفحه‌ی فعال، می توانید محتویات آنها را چاپ کنید.

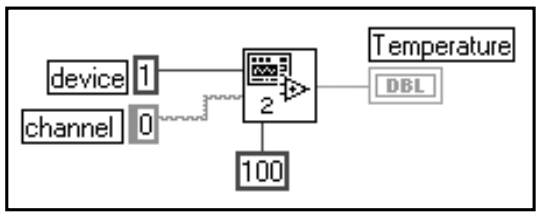
۹- با انتخاب گزینه‌ی File >> Save تمامی تغییرات انجام شده را ذخیره کنید. این برنامه را در زیرفهرست LabVIEW\My Activity ذخیره کنید. در ضمن نسخه‌ای از این برنامه نیز با عنوان Thermometer.vi در مسیر LabVIEW\Exercise\Ch5.llb ذخیره شده است. در تمرینات فصول آینده از این برنامه استفاده خواهیم نمود.

۱۰- با استفاده از ابزار Positioning Tool بخشی از صفحه‌ی نمودار بلوکی را که در شکل ۵-۲۷ نشان داده شده است انتخاب کنید.




**شکل ۵-۲۷: نحوه‌ی انتخاب بخشی از صفحه‌ی نمودار بلوکی به منظور ایجاد یک زیربرنامه**

سپس برای تبدیل این بخش از برنامه به زیربرنامه، گزینه‌ی Edit >> Create SubVI را انتخاب نمایید. همان گونه که در شکل ۵-۲۸ ملاحظه می کنید المان Temperature به همان صورت در برنامه‌ی فراخوان باقی می‌ماند. حال برای مشاهده‌ی صفحه‌ی پانل زیربرنامه‌ی جدید بر روی قاب آیکن آن دو بار کلیک کنید.




**شکل ۵-۲۸: تبدیل شدن بخشی از صفحه‌ی نمودار بلوکی به یک زیربرنامه با استفاده از گزینه‌ی Edit >> Create SubVI**

۱۱- اکنون برنامه‌ی Thermometer.vi و زیربرنامه‌ی ایجاد شده را ببندید. در این حالت از ذخیره نمودن تغییرات انجام شده خودداری کنید.

در اغلب موارد، بسیاری از کاربران پس از ایجاد یک برنامه، انتساب ترمینال‌های کانکتور را فراموش می‌کنند. در صورت عدم انتساب ترمینال‌های کانکتور به المان‌های موجود در صفحه‌ی پانل، به هنگام استفاده از این برنامه به عنوان زیربرنامه، سیم‌کشی ورودیها و خروجیهای برنامه‌ی مرتبه‌ی بالاتر به این زیربرنامه امکان‌پذیر نیست.

## خلاصه

در نرم‌افزار LabVIEW چند روش برای ذخیره‌ی برنامه‌ها وجود دارد. در این بسته‌ی نرم‌افزاری می‌توان برنامه‌ها را در کتابخانه ذخیره نمود. کتابخانه، فایل مخصوصی در LabVIEW می‌باشد که شامل تعدادی برنامه است. سیستم عامل رایانه، فایل کتابخانه‌ای را به عنوان یک فایل منفرد در نظر می‌گیرد. تنها به کمک LabVIEW می‌توان به فایل‌های مجزای موجود در کتابخانه دست یافت. در صورت استفاده از Windows 3.1 با محدودیت نام‌گذاری فایل‌ها با حداکثر ۸ کاراکتر مواجه هستید. با به‌کارگیری فایل‌های کتابخانه‌ای می‌توانید برای نام‌گذاری فایل‌ها و برنامه‌های خود حداکثر از ۲۵۵ کاراکتر استفاده نمایید. قبل از ذخیره‌ی برنامه باید از مزایا و معایب استفاده از کتابخانه‌ها به خوبی آگاه باشید. صرف نظر از روش ذخیره‌ی برنامه سعی کنید که در حین برنامه نویسی و یا ایجاد تغییر در برنامه چندین مرتبه برنامه‌ی خود را ذخیره کنید؛ زیرا در صورت عدم ذخیره‌ی تغییرات انجام شده، پس از قطع برق یا ایجاد اشکال در سیستم الکتریکی رایانه تمامی تغییرات تا آخرین زمان ذخیره نادیده فرض می‌شوند.

اگر برنامه‌ی مورد نظر شما به اجرا در نیامده ویا نتایج به‌دست آمده از آن نادرست و سؤال برانگیز است، از روشهای رفع اشکال برنامه در نرم‌افزار LabVIEW استفاده کنید. این روشها عبارتند از: اجرای برنامه در مد تک‌مرحله‌ای، اجرای متمایز، قرار دادن نقاط توقف در برنامه، استفاده از پروب بر روی سیمهای حامل داده جهت نمایش مقادیر عبور داده شده توسط آنها و... . این روشها به شما کمک می‌کنند تا نگاهی دقیق‌تر و بهتر به برنامه و مقدار داده‌های آن داشته باشید.

در نرم‌افزار LabVIEW، عملکرد زیربرنامه با Subroutine در زبانهای برنامه نویسی متنی معادل است. تمامی زیربرنامه‌ها باید دارای آیکن و کانکتور باشند. برای ایجاد یک زیربرنامه می‌توانید تمام یا بخشی از برنامه را مورد استفاده قرار دهید. برای اجتناب از بروز اشکالات و خطاهای احتمالی در سیم‌کشی زیربرنامه همواره از پنجره‌ی Help کمک بگیرید. به کمک این پنجره می‌توان ورودیهای الزامی، پیشنهادی و اختیاری را تشخیص داد و آنها را سیم‌کشی نمود. با انتخاب زیرپالت ...Functions >> Select a VI می‌توانید یک زیربرنامه را در صفحه‌ی نمودار بلوکی برنامه‌های دیگر قرار دهید. استفاده از زیربرنامه‌ها ویژگی مدولار بودن را برای LabVIEW به ارمغان می‌آورد. می‌توان از زیربرنامه‌ها در برنامه‌های سطوح بالاتر استفاده نمود.

نظیر سایر زبانهای برنامه‌نویسی، مستندسازی و ارائه‌ی توضیح در مورد برنامه بسیار مفید است. برای

ارائه‌ی توضیح در مورد یک برنامه، گزینه‌ی ...Windows >> Show VI Info را انتخاب نموده، سپس متن مورد نظر را در پنجره‌ی VI Information وارد کنید. در صورتی که نشانگر ماوس را بر روی آیکن این برنامه قرار دهید، این توضیحات در پنجره‌ی Help به نمایش در می آیند. در نرم‌افزار LabVIEW ارائه‌ی توضیح در مورد المان‌ها نیز امکان‌پذیر است. برای انجام این عمل ابتدا منوی کرکره ای را بر روی المان مورد نظر باز نموده، سپس گزینه‌ی ...Data Operations >> Description را انتخاب کنید. پس از این مرحله پنجره‌ای ظاهر می‌شود که می‌توان متن و توضیح مورد نظر را در آن وارد نمود. در صورت قرار گرفتن نشانگر ماوس بر روی المان‌های موجود در صفحه‌ی پانل، این توضیحات در پنجره‌ی Help به نمایش در می آیند.

در نرم‌افزار LabVIEW چندین گزینه برای چاپ برنامه‌ها وجود دارد. با انتخاب گزینه‌های...Print Window، ...Print Documentation و Print at Completion به ترتیب می‌توانید پنجره‌ی فعال، اطلاعاتی در مورد صفحات پانل و نمودار بلوکی، دستورهای موجود در برنامه و نتایج کامل برنامه را پس از اجرای آن چاپ کنید.

به شما تبریک می‌گوییم. اکنون اصول بنیادین نرم‌افزار LabVIEW را فرا گرفته‌اید و برای فراگیری ساختارها و دیگر دستورهای LabVIEW آمادگی بیشتری دارید.

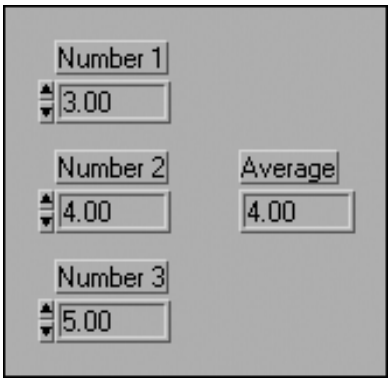
## تمرینات اضافی

در این مرحله دو تمرین برای شما در نظر گرفته‌ایم. اگر واقعاً در یافتن پاسخ با مشکل مواجه شدید پاسخ خود را با برنامه‌های موجود در مسیر LabVIEW\Exercise\Ch5.llb مقایسه کنید.

## تمرین ۳-۵: محاسبه‌ی میانگین

با ایجاد یک برنامه، میانگین سه عدد ورودی را محاسبه کرده، نتیجه را در صفحه‌ی پانل اعلام کنید. به یاد داشته باشید که برای این برنامه حتماً آیکن و کانکتور ایجاد نمایید. سپس این برنامه را با عنوان Find the Average.vi در زیرفهرست LabVIEW\My Activity ذخیره کنید.

صفحه‌ی پانل این برنامه به صورت شکل ۲۹-۵ است.



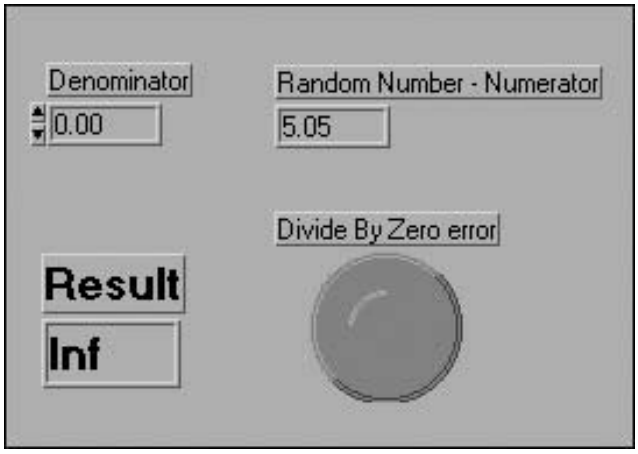

**شکل ۲۹-۵: صفحه‌ی پانل در برنامه‌ی Find the Average.vi**

## تمرین ۴-۵: خطای تقسیم بر صفر

در این تمرین قصد داریم با ایجاد یک برنامه، یک عدد تصادفی بین 0 تا 10 ایجاد کنیم و پس از تقسیم نمودن آن بر یک عدد ورودی، نتیجه را بر روی صفحه‌ی پانل نمایش دهیم. در صورتی که عدد 0 به عنوان ورودی انتخاب شود، یک LED با برچسب Divide By Zero error روشن می‌شود و «خطای تقسیم بر صفر» را آشکار می‌سازد. این برنامه را با عنوان Devide by Zero.vi در زیرفهرست LabVIEW\My Activity ذخیره کنید.

برای ایجاد این برنامه از دستور ?Equal استفاده کنید. این دستور در زیرپالت Functions >> Comparison قرار دارد.

صفحه‌ی پانل این برنامه به صورت شکل ۳۰-۵ است.




**شکل ۳۰-۵: صفحه‌ی پانل در برنامه‌ی Divide by Zero.vi**

LabVIEW

# ۶

# کنترل اجرای برنامه با استفاده از ساختارها

## در این فصل با موارد زیر آشنا می‌شوید:

- موارد استفاده از While Loop و For Loop و درک تفاوت بین آنها
- تشخیص ضرورت استفاده از ثبات انتقال در برنامه های گرافیکی
- ساختار شرطی عددی و جبری
- عملکرد ساختار ترتیبی
- روش استفاده از گره‌ی فرمولی جهت انجام محاسبات و اعمال فرمول‌های پیچیده‌ی ریاضی
- نحوه‌ی ایجاد یک پنجره‌ی محاوره‌ای در محیط LabVIEW
- روش به‌کارگیری برخی از دستورهای ساده جهت کنترل زمان اجرای برنامه

# LabVIEW

۶

## کنترل اجرای برنامه با استفاده از ساختارها

ساختارها[1] از مهم‌ترین گره‌ها در نرم‌افزار LabVIEW هستند. این گره‌ها بر روند اجرای یک برنامه نظارت دارند و آن را فرماندهی می‌کنند. این المان‌ها در بسته‌ی نرم‌افزاری LabVIEW درست شبیه به ساختارهای کنترلی در زبانهای برنامه‌نویسی استاندارد و متداول می‌باشند. در این فصل با چهار ساختار اصلی در LabVIEW آشنا خواهید شد. این ساختارها عبارتند از: While Loop، For Loop، Case Structure و Sequence Structure.

در این فصل همچنین چگونگی اعمال فرمول‌های پیچیده‌ی ریاضی با استفاده از گره‌ی فرمولی را فرا خواهید گرفت و در ضمن با روشهای کنترل زمان اجرای برنامه آشنا می‌شوید.

در این فصل اهداف زیر دنبال می‌شوند:

- فراگیری موارد استفاده از While Loop و For Loop و درک تفاوت بین آنها.
- تشخیص ضرورت استفاده از ثبات انتقال در برنامه‌های گرافیکی.
- آشنایی با ساختار شرطی عددی و جبری.
- فراگیری عملکرد ساختار ترتیبی.
- استفاده از گره‌ی فرمولی جهت انجام محاسبات و اعمال فرمول‌های پیچیده‌ی ریاضی.
- فراگیری نحوه‌ی ایجاد یک پنجره‌ی محاوره‌ای در محیط LabVIEW.
- درک روش به‌کارگیری برخی از دستورهای ساده جهت کنترل زمان اجرای برنامه.

---

1- Structures

۶

اصطلاحات کلیدی موجود در این فصل عبارتند از:

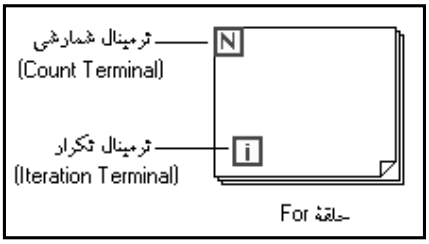
- حلقه‌ی For
- حلقه‌ی While
- ترمینال تکرار (Iteration Terminal)
- ترمینال شرطی (Conditional Terminal)
- ترمینال شمارشی (Count Terminal)
- تونل (Tunnel)
- نقطه‌ی اجباری (Coercion Dot)
- ثبات انتقال (Shift Register)
- ساختار شرطی (Case Structure)
- ترمینال انتخابی (Selector Terminal)
- پنجره‌ی محاوره‌ای (Dialog Box)
- ساختار ترتیبی (Sequence Structure)
- متغیر محلی ترتیبی (Sequence Local)
- گره‌ی فرمولی (Formula Node)

## حلقه‌ها

اگر تا به حال با زبانهای برنامه‌نویسی مرسوم و متداول، برنامه‌ای را نوشته باشید احتمالاً بخشی از برنامه را با استفاده از حلقه‌ها تکرار نموده‌اید. در نرم‌افزار LabVIEW دو ساختار While Loop و For Loop جهت تکرار بخشی از برنامه در نظر گرفته شده اند. حلقه‌ی For، برنامه را به تعداد دفعات مشخصی اجرا می کند. اما اجرای حلقه‌ی While تا زمانی ادامه می یابدکه یک شرط تعیین شده در برنامه در وضعیت True قرار داشته باشد. هر دو حلقه‌ی مذکور در زیرپالت Functions >> Structures قرار دارند.

## حلقه‌ی For

یک حلقه‌ی For کدها و دستورهای داخل چهارچوب حلقه را که به آنها Subdiagram گویند به تعداد دفعات کران داری اجرا می کند. این عدد کران دار به ترمینال شمارشی حلقه نسبت داده می شود. برای تنظیم و مقداردهی به این ترمینال می توانید عددی از خارج از حلقه به ترمینال مذکور سیم کشی کنید. واضح است که اگر عدد 0 را به این ترمینال سیم کشی کنید، حلقه اجرا نخواهد شد. در شکل ۱-۶ یک حلقه‌ی For را به همراه دو ترمینال تکرار و شمارشی آن ملاحظه می کنید.




**شکل ۱-۶: شکل ظاهری حلقه‌ی For و ترمینال‌های موجود در چهارچوب آن**

ترمینال تکرار تعداد حلقه‌هایی را که تاکنون تکمیل شده‌اند نشان می‌دهد. بدین ترتیب که عدد 0 در خلال اولین تکرار حلقه در این ترمینال قرار می گیرد، عدد 1 در طول دومین تکرار حلقه و به همین

۶

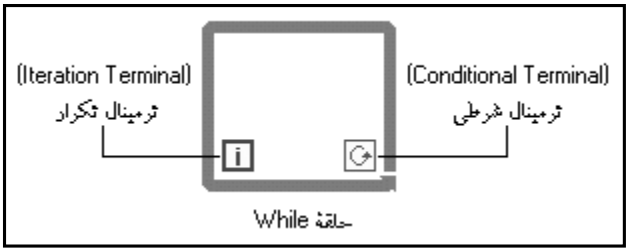
ترتیب تا N-1. N عددی است که به ترمینال شمارشی حلقه سیم‌کشی شده است و به تعداد آن، حلقه به اجرا در می‌آید.

عملکرد حلقه‌ی For معادل با کدهای فرضی زیر است:

```
for i = 0 to N-1
    Execute subdiagram
```

## حلقه‌ی While

در حلقه‌ی While، تا زمانی که مقدار جبری مربوط به حلقه به حالت False تغییر وضعیت نداده باشد، دستورها و کدهای موجود در داخل چهارچوب حلقه اجرا می‌شوند. این مقدار جبری از خارج از چهارچوب حلقه به ترمینال شرطی سیم‌کشی می‌شود. LabVIEW مقدار موجود در ترمینال مذکور را در پایان هر تکرار حلقه بررسی می‌کند و در صورتی که این مقدار در حالت True باشد، حلقه یک بار دیگر تکرار می‌گردد. مقدار جبری پیش‌فرض برای این ترمینال، False می‌باشد. بنابراین در صورتی که مقداری را به آن سیم‌کشی نکرده باشید دستورات داخل چهارچوب حلقه فقط برای یک مرتبه به اجرا در می‌آیند. ترمینال تکرار در While Loop دقیقاً شبیه به همین ترمینال در For Loop رفتار می‌کند. در شکل ۲-۶ یک حلقه‌ی While به همراه دو ترمینال تکرار و شرطی آن نشان داده شده است.



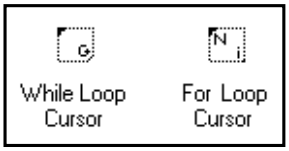

**شکل ۲-۶: شکل ظاهری حلقه‌ی While و ترمینال‌های موجود در چهارچوب آن**

عملکرد حلقه‌ی While معادل با کدهای فرضی زیر است:

```
Do
        Execute subdiagram (which sets condition)
While condition is True
```

## روش قرار دادن المان‌ها در داخل ساختار

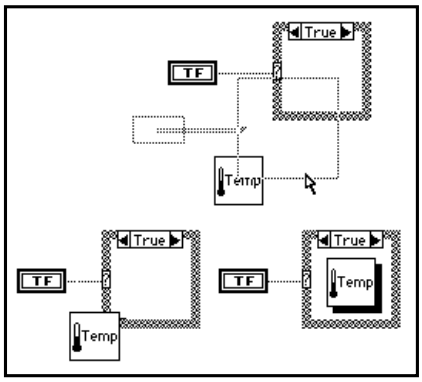
زمانی که یک ساختار را از زیرپالت Functions >> Structures انتخاب می‌کنید، نشانگر ماوس به صورت تصویر کوچکی از ساختار انتخاب شده در می‌آید. به عنوان مثال در صورت انتخاب حلقه‌های While و For، نشانگر ماوس به اشکال زیر در می‌آید.




**شکل ۳-۶: شکل ظاهری نشانگر ماوس در صورت انتخاب حلقه‌های While و For**

سپس در محلی از صفحه‌ی نمودار بلوکی که قصد دارید گوشه‌ی بالایی ساختار در آن جا قرار گیرد کلیک کنید و در همان حالت ماوس را برای تعیین اندازه‌ی ساختار به صورت قطری حرکت دهید. در صورت رها کردن دکمه‌ی ماوس، ساختار مورد نظر به همراه دو ترمینال موجود در آن ظاهر می شود. پس از ظاهر شدن ساختار بر روی صفحه‌ی نمودار بلوکی، می توانید المان های موجود در صفحه‌ی نمودار بلوکی را انتخاب کنید و آنها را در داخل ساختار قرار دهید. در ضمن می توانید المان های موجود در صفحه‌ی نمودار بلوکی را نیز به داخل ساختار حرکت دهید. در این حالت چهارچوب ساختار متمایز می شود. این مطلب در مورد خارج کردن المان از ساختار نیز صادق است. بدین مفهوم که به هنگام خارج نمودن المان از ساختار، چهارچوب آن متمایز می گردد.

برای تغییر اندازه‌ی حلقه، ابزار Positioning Tool را بر روی یکی از گوشه های چهارچوب آن قرار دهید تا نشانگر ماوس به شکل ابزار تغییر اندازه یا Resizing Tool در آید. در این حالت با حرکت دادن ماوس، اندازه‌ی حلقه را به دلخواه تغییر دهید. اگر در هنگام تغییر اندازه و یا حرکت دادن ساختار، به طور ناخواسته بخشی از المان دیگری را بپوشاند، المان مذکور بر روی چهارچوب ساختار قرار می گیرد و قابل رؤیت خواهد بود. اگر در هنگام تغییر اندازه و یا حرکت دادن ساختار، به طور کامل بر روی المانی قرار گیرد، المان مذکور با سایه‌ی پهن ظاهر می‌شود و نشانگر آن است که این المان در بالا یا در زیر ساختار قرار گرفته است و نه در داخل آن. هر دو مورد مذکور در شکل ۴-۶ نشان داده شده است.




شکل ۴-۶

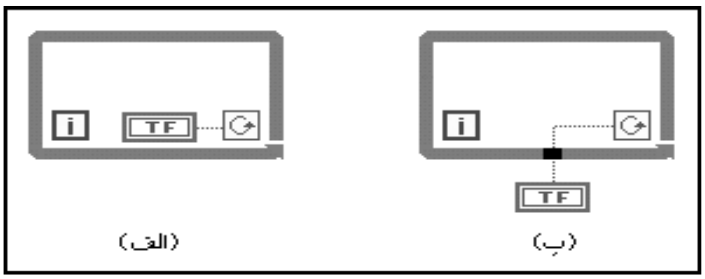
## ترمینال های موجود در داخل چهارچوب حلقه ها

جریان داده ها از داخل حلقه به خارج از آن یا برعکس، از طریق یک جعبه‌ی کوچک که بر روی چهارچوب حلقه قرار دارد انجام می گیرد. به این جعبه‌ی کوچک در اصطلاح تونل گویند. به دلیل اینکه برنامه های نوشته شده در محیط LabVIEW بر اساس اصل جریانِ داده عمل می کنند، بنابراین مقادیر ورودی حلقه باید قبل از شروع به کار حلقه آماده و در دسترس باشند.

نیاز به توضیح است که دستیابی به مقادیر خروجی حلقه تنها پس از تکمیل تمامی تکرارهای آن امکان‌پذیر است. با توجه به اصل جریانِ داده، اگر قصد دارید تا در هر تکرار حلقه مقدار یک ترمینال،

۶

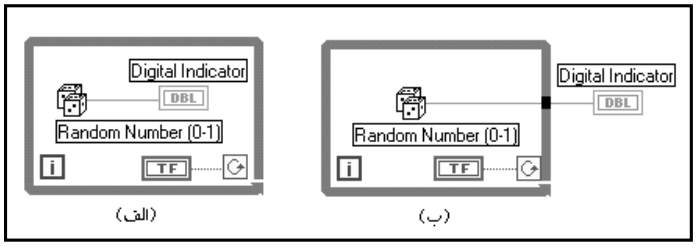
بررسی شده یا به روز رسانده شود، باید در داخل حلقه قرار گرفته باشد. همان گونه که در شکل ۵-۶ (الف) نشان داده شده است، مقدار ترمینال TF در انتهای هر یک از تکرارهای حلقه‌ی While بررسی می‌شود و در صورت قرائت مقدار False از این ترمینال، اجرای حلقه پایان می یابد.




شکل ۵-۶

در صورتی که مطابق شکل ۵-۶ (ب) ترمینال TF را خارج از چهارچوب حلقه‌ی While قرار دهید، در این حالت طبق اصل جریانِ داده مقدار ترمینال جبری قبل از ورود به حلقه قرائت می شود و نه در طول حلقه و یا پس از تکمیل آن.

به طریق مشابه، المان Digital Indicator در شکل ۶-۶ (الف) در طول هر یک از تکرارهای حلقه به روز رسانده می شود. در حالی که این عمل در مورد المان Digital Indicator که در شکل ۶-۶ (ب) قرار دارد، یک مرتبه و تنها پس از تکمیل حلقه انجام می گیرد. این المان حاوی مقدار عدد تصادفی ایجاد شده در آخرین تکرار حلقه است.




شکل ۶-۶

در صورتی که قصد دارید چهارچوب حلقه را به همراه المان‌های قرار گرفته در داخل آن حذف کنید، منوی کرکره‌ای را بر روی چهارچوب حلقه باز نموده، گزینه‌ی Remove While Loop یا Remove For Loop را انتخاب کنید. اگر به کمک ابزار Positioning Tool، یک حلقه را انتخاب نموده، سپس آن را حذف کنید، تمامی المان‌های قرار گرفته در داخل چهارچوب حلقه نیز به همراه آن حذف خواهند شد.

به کمک ویژگی «اندیس گذاری خودکار[2]» در LabVIEW می توانید آرایه هایی حاوی مؤلفه های عددی را در یک حلقه ایجاد نموده، آنها را بر روی چهارچوب حلقه ذخیره کنید. در فصل ۷، آرایه ها و قابلیتهای آنها را مورد بررسی قرار خواهیم داد.

2- Auto-Indexing

به یاد داشته باشید که در اولین تکرار حلقه‌ی For یا While مقدار ترمینال تکرار برابر صفر است. در صورتی که قصد داشته باشید تعداد تکرارهای انجام شده در حلقه را به‌دست آورید باید یک واحد به مقدار «ترمینال تکرار» اضافه کنید.

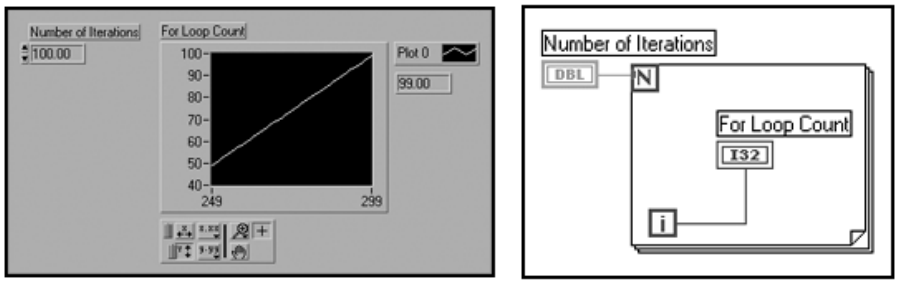
۶

## تمرین ۱-۶: شمارش به کمک حلقه‌ها

در این تمرین، یک حلقه‌ی For ایجاد می کنید که شمارش تعداد حلقه‌های انجام شده‌ی آن بر روی صفحه‌ی پانل نشان داده می‌شود. در این برنامه، ابتدا کاربر تعداد تکرارهای حلقه را در المان Number of Iterations تعیین می کند و حلقه از عدد صفر تا یک واحد کمتر از مقدار وارد شده تکرار می شود. در مرحله‌ی بعدی به ایجاد یک حلقه‌ی While می‌پردازیم. تا زمانی که با تغییر وضعیت یک کلید جبری به وضعیت False، این حلقه را متوقف نکرده‌اید به شمارش ادامه می‌دهد. تنها برای سرگرمی و همچنین برای توضیح یک نکته‌ی بسیار مهم تأثیر قرار دادن المان‌های کنترل و نشان‌دهنده را در خارج از حلقه‌ی While در این برنامه ملاحظه خواهید نمود.

۱- با انتخاب گزینه‌ی New << File یک برنامه‌ی جدید ایجاد کنید.

۲- صفحات پانل و نمودار بلوکی این برنامه را مطابق شکل ۷-۶ ایجاد نمایید. بهتر است برای مشاهده‌ی هر دو صفحه از فرمان های Tile Up and Down یا Tile Left and Right که در منوی Windows قرار دارند کمک بگیرید. گزینه‌ی Waveform Chart را از زیرپالت Graph << Controls انتخاب کنید و آن را بر روی صفحه‌ی پانل قرار دهید. پس از ظاهر شدن نمودار، برچسب For Loop Count را برای آن تعیین کنید (در مورد گراف ها و نمودارها در فصل ۸ به طور مفصل توضیح خواهیم داد). یک المان DigitalControl را از زیرپالت Numeric << Controls انتخاب نموده، برچسب Number of Iterations را برای آن ایجاد کنید.




**شکل ۷-۶: صفحات پانل و نمودار بلوکی در برنامه‌ی For Loop Count**

۳- منوی کرکره‌ای را بر روی Waveform Chart باز نموده، سپس گزینه‌ی Autoscale << Y Scale را انتخاب کنید. بدین ترتیب حد بالایی انتخاب شده در نمودار با اعداد پدید آمده توسط For Loop سازگار است و می‌تواند آنها را تحت پوشش قرار دهد. سپس گزینه‌ی Digital Display << Show را از منوی کرکره‌ای Waveform Chart انتخاب نمایید. با استفاده از ابزار Operating Tool یک عدد دلخواه (به غیر از 0) به المان Number of Iterations نسبت داده، سپس برنامه را اجرا کنید. توجه داشته باشید که در المان Digital Indicator مقادیر 0 تا N-1 به نمایش در می‌آیند (N عددی است که توسط کاربر وارد می‌شود). با هر بار اجرای حلقه،

۶

تعداد دفعات اجرای آن یا به عبارت دیگر تعداد تکرارهای حلقه بر روی محور Y ترسیم می شود. در این تمرین، محور X نشانگر زمان است و هر واحد زمانی بیانگر یک بار تکرار حلقه می باشد.

۴- به نقطه‌ی خاکستری رنگ موجود در محل اتصال ترمینال شمارشی و سیم مربوط به ترمینال Number of Iterations توجه کنید. این نقطه را در اصطلاح «نقطه‌ی اجباری[3]» گویند. در مورد این نقطه در بخش بعد صحبت خواهیم کرد. حال منوی کرکره‌ای را بر روی المان Number of Iterations باز کنید و گزینه‌ی Representation >> I32 را انتخاب نمایید.

۵- ذخیره نمودن این برنامه الزامی نیست، زیرا از این برنامه در تمرینات بعدی استفاده نخواهیم کرد. در صورت تمایل، این برنامه را با عنوان For Loop Count در زیرفهرست LabVIEW\My Activity ذخیره کنید.

حال برای بررسی حلقه‌ی While یک برنامه‌ی جدید ایجاد کنید.

۶- صفحات پانل و نمودار بلوکی را مطابق شکل ۸-۶ ایجاد نمایید. به خاطر داشته باشید که در هنگام قرار دادن المان های جبری بر روی صفحه‌ی پانل، این المان ها به صورت پیش فرض در وضعیت False قرار می گیرند.

شکل ۸-۶

Operating Tool

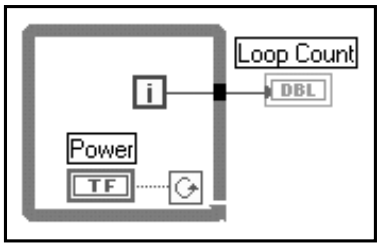
۷- به کمک ابزار Operating Tool بر روی کلید موجود در صفحه‌ی پانل کلیک کنید وآن را در وضعیت بالا یا True قرار دهید. حال برنامه را اجرا کنید. برای متوقف نمودن اجرای برنامه، توسط ابزار Operating Tool بر روی همین کلید کلیک کنید و آن را در وضعیت پایین یا False قرار دهید. در این برنامه، المان Loop Count در طول هر یک از تکرارهای حلقه به روز رسانده می شود.

۸- در حالتی که کلید در وضعیت False قرار دارد، مجدداً برنامه را اجرا کنید. توجه کنید که حلقه‌ی While تنها یک مرتبه اجرا می شود. برای درک این مطلب به یاد آورید که ترمینال شرطی در انتهای هر تکرار حلقه بررسی می شود و نه در ابتدای آن. بنابراین در صورتی که این ترمینال به هیچ المان دیگری سیم کشی نشده باشد، حلقه‌ی While حداقل یک مرتبه اجرا می گردد.

۹- با انتخاب گزینه‌ی Windows >> Show Diagram به صفحه‌ی نمودار بلوکی تغییر وضعیت داده، المان Loop Count را مطابق شکل ۹-۶ به خارج از چهارچوب حلقه حرکت دهید. مشاهده می کنید که به

3- Coercion Dot

محض عبور سیم از حلقه، یک تونل به صورت خودکار در دیواره‌ی حلقه ایجاد می گردد.




**شکل ۹-۶**

۱۰- اکنون کلید را در وضعیت True قرار دهید و برنامه را اجرا کنید. ملاحظه می کنید که برنامه به طور دائم اجرا می شود. در حقیقت تا زمانی که برنامه را متوقف نکنید، این برنامه اجرا می شود. حال با کلیک کردن بر روی کلید Power توسط Operating Tool و قرار دادن این کلید در حالت False، اجرای برنامه را متوقف کنید و محتویات نشان دهنده‌ی Loop Count را ببینید. این المان تنها پس از تکمیل اجرای حلقه به روز رسانده شده و تعداد تکرارهای نهایی حلقه را نشان می دهد. این مقدار پس از تکمیل اجرای حلقه به دست می آید. در فصل ۷ مطالب بیشتری در مورد داده های مرکب نظیر آرایه ها و کلاسترها خواهید آموخت. تا آن زمان سعی نکنید تا یک داده‌ی عددی را از حلقه‌ی For نظیر آنچه که در مورد حلقه‌ی While انجام دادید، استخراج نمایید؛ زیرا در این صورت به دلیل عدم هماهنگی سیم انتخاب شده با نوع داده، با اشکالاتی نظیر bad wire مواجه خواهید شد که در حال حاضر در مورد علت بروز آن اطلاعات کافی در دست ندارید.

۱۱- پس از انجام تمرین و اجرای برنامه، آن را با عنوان Loop Count.vi در زیرفهرست LabVIEW\My Activity ذخیره کنید.

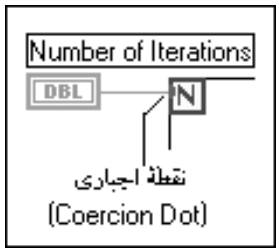
۱۲- حال تنها برای درک اعمالی که نباید در خلال برنامه انجام دهید مراحل زیر را دنبال نمایید.

کلید را به خارج از حلقه حرکت دهید اما سیمهای رابط آن را در همان حالت نگه دارید. سپس کلید را در وضعیت True قرار دهید و برنامه را اجرا کنید. با کلیک کردن مجدد بر روی کلید مذکور اجرای برنامه را متوقف کنید. ملاحظه می کنید که اجرای برنامه متوقف نخواهد شد. در این حالت برنامه در داخل یک حلقه‌ی بی پایان قرار می گیرد و تنها به کمک دکمه‌ی Abort می توان به اجرای آن پایان داد؛ زیرا هنگامی که LabVIEW اجرای حلقه را آغاز می کند مقادیر المان های کنترل قرار گرفته در خارج از حلقه را بررسی نمی کند. نظیر همین مطلب را در مورد به روز درآوردن المان های نشان دهنده پس از تکمیل اجرای حلقه که در خارج از حلقه قرار گرفته اند، خاطر نشان کردیم. برای متوقف نمودن اجرای برنامه، دکمه‌ی Abort را فشار دهید. اگر در هنگام شروع به کار حلقه، کلید در وضعیت False قرار داشته باشد، حلقه تنها یک مرتبه اجرا می شود. در این حالت برنامه را ببندید و از ذخیره نمودن تغییرات انجام شده خودداری کنید.

۶

## نقطه‌ی اجباری

به شکل ۱۰-۶ توجه کنید. آیا نقطه‌ی خاکستری رنگ موجود در محل اتصال ترمینال شمارشی حلقه‌ی For و سیم مربوط به ترمینال Number of Iterations در تمرین قبل را به خاطر می‌آورید؟




**شکل ۱۰-۶: نقطه‌ی اجباری که در اثر ناهماهنگی بین نوع نمایش داده‌ها در محل اتصال سیم و ترمینال ظاهر می‌شود.**

همان طور که قبلاً اشاره شد به این نقطه، نقطه‌ی اجباری گویند. این نام از آن جهت انتخاب شده است که LabVIEW به اجبار از یک نمایش عددی برای پوشش دادن نمایش عددی دیگر استفاده می‌کند. در صورتی که دو ترمینال از نمایش‌های عددی مختلف به یکدیگر سیم کشی شده باشند، LabVIEW یکی از آنها را مطابق با ترمینال دیگر تبدیل می‌کند.

ترمینال شمارشی حلقه در تمرین قبل دارای یک نمایش عدد صحیح ۳۲ بیتی بوده، در حالی که المان Number of Iterations یک عدد اعشاری است. همان گونه که عنوان شد یکی از دو حالت نمایشی به دیگری تبدیل می‌گردد. در این حالت، عدد اعشاری به یک عدد صحیح تبدیل می‌شود. در انجام این عمل، LabVIEW یک نسخه‌ی کپی از اعداد موجود در حافظه تهیه می‌کند که این نسخه‌ی کپی نیز به نوبه‌ی خود مقداری از فضای حافظه را اشغال می‌کند.

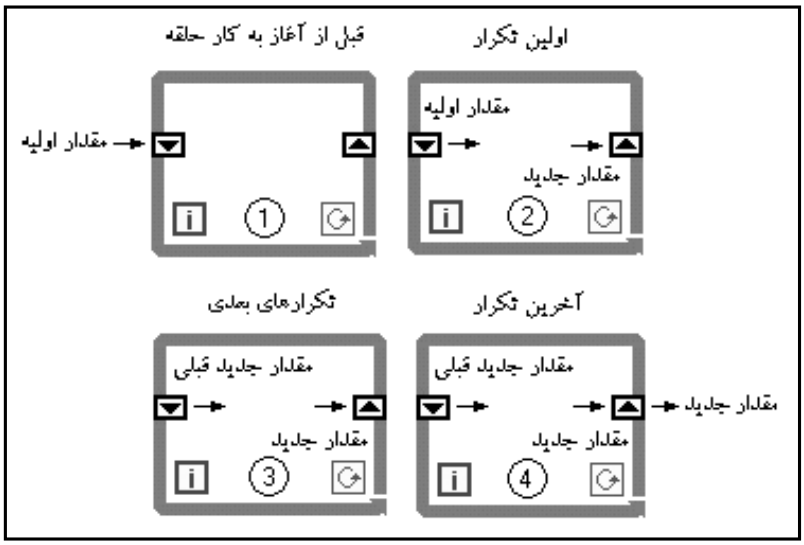
اگر در حالتی که فضای اضافی برای اعداد اسکالر قابل اغماض باشد از آرایه‌ها استفاده کنید، LabVIEW می‌تواند این فضاهای اضافی را به سرعت با یکدیگر جمع کند و مقدار حافظه‌ی اشغال شده توسط آنها را به حداقل برساند. بنابراین با تغییر نمایش المان‌های کنترل و نشان‌دهنده به نمایش واقعی و هماهنگ با نوع داده‌هایی که متناظر با آنها در نظر گرفته شده است، سعی کنید تا تعداد نقاط اجباری را در آرایه‌های بزرگ به حداقل برسانید. در صورت تبدیل عدد اعشاری به عدد صحیح، LabVIEW آنها را به نزدیکترین عدد صحیح گرد می‌کند.

روش ساده برای ایجاد یک ترمینال ورودی شمارشی متناسب با نوع و نمایش داده‌ها، باز نمودن منوی کرکره‌ای بر روی ترمینال شمارشی حلقه و انتخاب گزینه‌ی Create Constant (برای یک عدد ثابت در صفحه‌ی نمودار بلوکی) و یا انتخاب گزینه‌ی Create Control (برای یک المان کنترل در صفحه‌ی پانل) می‌باشد.

## ثبات انتقال

ثبات انتقال نوعی متغیر است که جهت انتقال مقادیر از یک تکرار حلقه به تکرار بعدی در For Loop و While Loop مورد استفاده قرار می‌گیرد. به کارگیری و وجود این متغیرها در ساختارهای گرافیکی در محیط LabVIEW لازم و منحصر به فرد است. برای ایجاد یک ثبات انتقال کافی است منوی کرکره‌ای را بر روی دیواره‌ی سمت چپ یا راست حلقه باز نموده، سپس گزینه‌ی Add Shift Register را انتخاب کنید.

ثبات انتقال از یک زوج ترمینال تشکیل شده است که دقیقاً در دو جهت مخالف یکدیگر و بر روی اضلاع عمودی چهارچوب حلقه قرار می‌گیرند. ترمینال سمت راست، داده‌ها را تا زمان تکمیل یک تکرار حلقه ذخیره می‌کند. همان گونه که در شکل ۱۱-۶ ملاحظه می‌کنید این داده‌ها در انتهای هر تکرار، انتقال داده شده، در ابتدای تکرار بعدی در ترمینال سمت چپ ظاهر می‌شوند.

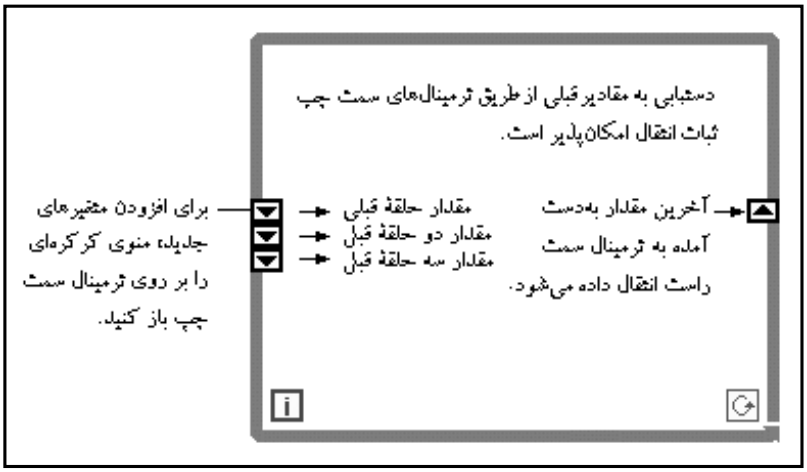



**شکل ۱۱-۶: انتقال داده‌ها در ترمینال‌های ثبات انتقال**

ثبات انتقال می‌تواند هر نوع داده اعم از جبری، عددی، رشته‌ای، آرایه‌ای و... را در خود نگه‌داری کند. یک ثبات خود را به طور خودکار با نوع داده‌ی اولین المانی که به آن سیم‌کشی شده است وفق می‌دهد. پس از ایجاد ثبات انتقال و قرار گرفتن آن بر روی دیواره‌ی حلقه، ترمینال آن به رنگ سیاه ظاهر می‌شود ولی در مراحل بعدی رنگ نوع داده‌ای را که به آن سیم‌کشی شده است به خود می‌گیرد. همان گونه که در شکل ۱۲-۶ نشان داده شده است می‌توانید از ثباتهای انتقال برای مراجعه به مقادیر حاصل از تکرارهای قبلی استفاده کنید. استفاده از این قابلیت در حالتی مفید است که قصد دارید میانگین داده‌های به‌دست آمده از تکرارهای مختلف را محاسبه کنید. برای دستیابی به مقادیر حاصل از تکرارهای قبلی، منوی کرکره‌ای را بر روی ترمینال سمت چپ باز نموده، با انتخاب گزینه‌ی Add Element یک ترمینال دیگر بر روی دیواره‌ی سمت چپ حلقه اضافه کنید. در شکل ۱۲-۶ ملاحظه

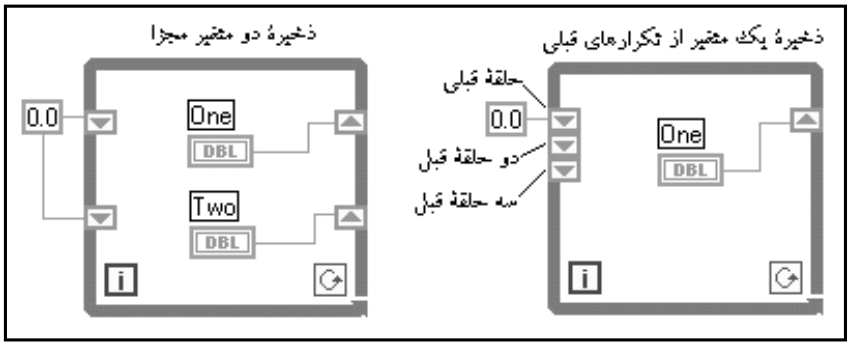
۶

می کنید که برای نگه داشتن مقادیر به‌دست آمده از سه تکرار قبلی حلقه، از سه ترمینال ثبات انتقال بر روی دیواره‌ی سمت چپ حلقه استفاده شده است.



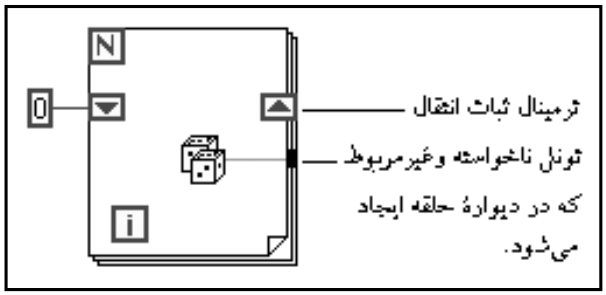

**شکل ۱۲-۶: ایجاد سه ترمینال ثبات انتقال بر روی دیواره‌ی سمت چپ حلقه برای نگهداشتن مقادیر به‌دست آمده از سه تکرار قبلی حلقه**

در یک حلقه استفاده نمودن از چندین ثبات انتقال برای ذخیره‌ی متغیرهای مختلف امکان پذیر است. برای انجام این عمل منوی کرکره‌ای را بر روی دیواره‌ی حلقه باز نموده و گزینه‌ی Add Shift Register را انتخاب کنید. نیاز به توضیح است که در هر مرتبه، یک زوج متغیر بر روی حلقه ایجاد می گردد. در این حالت همواره ترمینال سمت چپ به صورت موازی با ترمینال سمت راست قرار می گیرد. در صورت حرکت دادن یکی از ترمینال ها، هر دوی آنها به موازات یکدیگر حرکت می کنند. بنابراین اگر تعداد ثباتهای موجود بر روی حلقه زیاد باشد و نتوانید ترمینال های متناظر با یکدیگر را تشخیص دهید کافی است یکی از آنها را انتخاب نمایید. ملاحظه می کنید که ترمینال متناظر با آن نیز انتخاب می گردد یا اینکه یکی از ترمینال ها را حرکت دهید و جابه‌جایی ترمینال متناظر با آن را ببینید. مراقب باشید که مفهوم ذخیره‌ی چندین متغیر در چند ثبات انتقال و مفهوم ذخیره‌ی یک متغیر منفرد از چند تکرار قبلی در یک ثبات انتقال را با یکدیگر اشتباه نگیرید. در شکل ۱۳-۶ تفاوت این دو مورد نشان داده شده است.




**شکل ۱۳-۶: تفاوت بین دو مفهوم ذخیره‌ی چندین متغیر در چند ثبات انتقال و ذخیره‌ی یک متغیر منفرد از چند تکرار قبلی در یک ثبات انتقال**

حتماً ترمینال ثبات انتقال را به طور صحیح سیم کشی کنید. در غیر این صورت مطابق شکل ۱۴-۶ یک تونل ناخواسته و نامربوط در دیواره‌ی حلقه ایجاد می گردد.



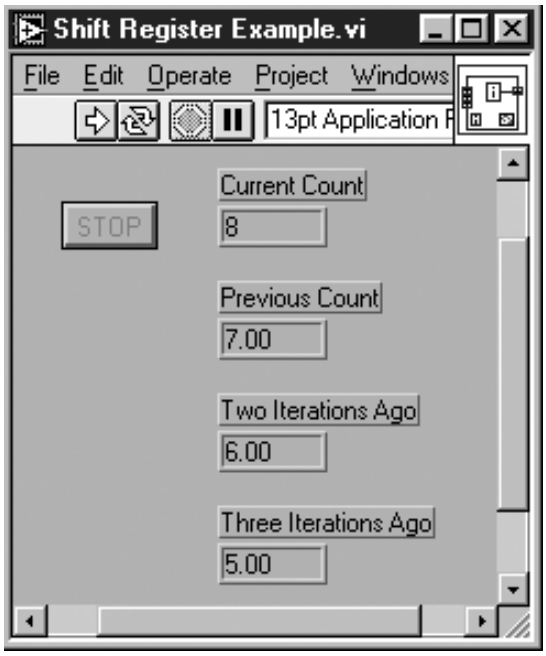

**شکل ۱۴-۶: به وجود آمدن یک تونل ناخواسته و نامربوط در صورت عدم انجام سیم‌کشی صحیح ترمینال ثبات انتقال**

اگر در مورد درک مطالب مندرج در این بخش با مشکل مواجه هستید جای هیچ نگرانی نیست. ثبات انتقال یک مفهوم کاملاً جدید است و با مفاهیم موجود در زبانهای برنامه نویسی متداول و رایج تفاوت دارد. انجام مرحله به مرحله‌ی تمرین بعدی، شما را در فراگیری این مفهوم یاری می دهد.

## تمرین ۲-۶: به‌کارگیری ثبات انتقال

در این برنامه برای درک بهتر عملکرد ثباتهای انتقال، موارد استفاده‌ی آنها را جهت دستیابی به مقادیر تکرار قبلی در یک حلقه بررسی می کنیم. در این تمرین به کمک یک ثبات انتقال، مقادیر ترمینال شمارشی را از تکرارهای قبل به دست می آوریم.

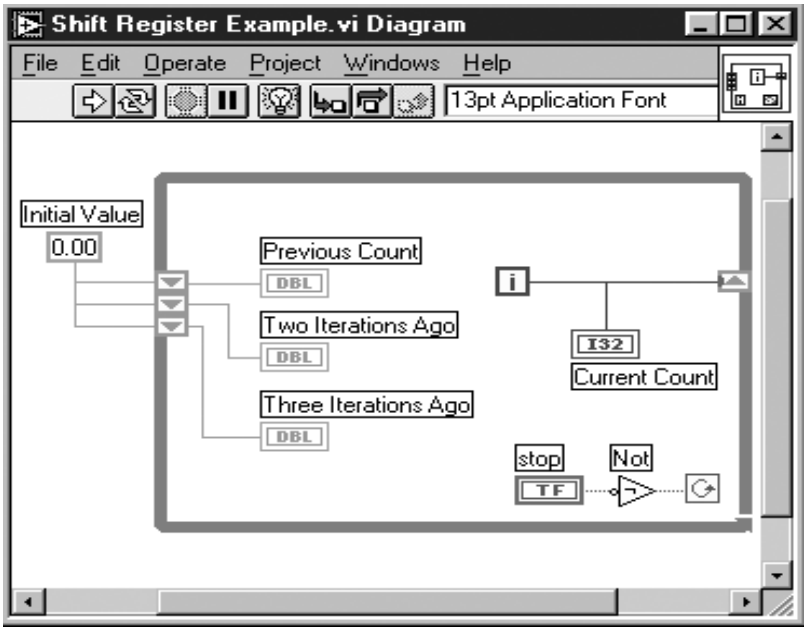
۱- برنامه‌ی Shift Register Example.vi را از مسیر LabVIEW\Exercise\Ch6.llb باز کنید.




**شکل ۱۵-۶: صفحه‌ی پانل در برنامه‌ی Shift Register Example.vi**

همان گونه که در شکل ۱۵-۶ ملاحظه می کنید در صفحه‌ی پانل این برنامه چهار المان Digital Indicator وجود دارد. نشان دهنده‌ی Current Count مقدار ترمینال شمارشی حلقه‌ی فعلی را نشان می دهد. در صفحه‌ی نمودار بلوکی نشان داده شده در شکل ۱۶-۶ ملاحظه می کنید که این نشان‌دهنده به ترمینال تکرار حلقه سیم کشی شده است. المان Previous Count مقدار ترمینال شمارشی حلقه را در یک تکرار قبل نشان می دهد. به همین ترتیب المان های Two Iterations Ago و Three Iterations Ago مقادیر ترمینال شمارشی حلقه را در دو و سه تکرار قبل نمایش می دهند.

۲- با انتخاب گزینه‌ی Windows >> Show Diagram صفحه‌ی نمودار بلوکی را به عنوان پنجره‌ی پیش زمینه انتخاب کنید.




**شکل ۱۶-۶: صفحه‌ی نمودار بلوکی در برنامه‌ی Shift Register Example.vi**

همان گونه که ملاحظه می کنید عدد 0.00 به عنوان مقدار اولیه یا Initial Value به ثبات انتقال نسبت داده شده است. در ابتدای تکرار بعدی، مقدار Current Count به ترمینال بالایی در سمت چپ حلقه منتقل می‌شود تا مقدار Previous Count را ایجاد کند. به همین ترتیب مقدار Previous Count به ترمینال Two Iterations Ago منتقل می گردد و... .

۳- پس از بررسی صفحه‌ی نمودار بلوکی با انتخاب گزینه‌ی Windows >>Tile Left and Right دو صفحه‌ی پانل و نمودار بلوکی را در کنار یکدیگر قرار دهید تا هر دو صفحه قابل رؤیت باشند.

۴- با کلیک نمودن بر روی دکمه‌ی Execution Highlighting، مد اجرای متمایز را در این برنامه فعال کنید.

Execution Highlighting Button

Step Into Button

۵- برنامه را اجرا کنید و عبور حباب‌های موجود در سیمهای حامل داده را به دقت ملاحظه کنید. در صورتی که سرعت عبور حباب‌ها بالاست، اجرای برنامه را متوقف نموده، برای فعال کردن مد اجرایی تک مرحله‌ای، بر روی دکمه‌ی Step Into کلیک کنید. برای اجرای هر مرحله از برنامه بر روی این دکمه کلیک کرده، چگونگی تغییرات نشان‌دهنده‌های موجود در صفحه‌ی پانل را مشاهده کنید.

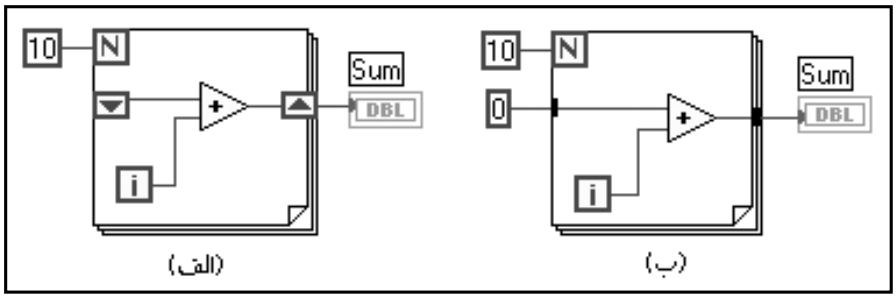
توجه داشته باشید که این برنامه در هر تکرار حلقه‌ی While برای ذخیره‌ی مقادیر قبلی در ترمینال‌های سمت چپ ثبات انتقال از الگوریتم FIFO[4] استفاده می‌کند. در هر تکرار حلقه، به مقدار ترمینال شمارشی که به ترمینال سمت راست ثبات انتقال سیم کشی شده است، یک واحد اضافه می‌شود. این مقدار در ابتدای تکرار بعدی به ترمینال سمت چپ متغیر Previous Count منتقل می‌گردد. در این برنامه، تنها سه مقدار آخر نگهداری می‌شوند. برای نگه‌داری مقادیر بیشتر، منوی کرکره‌ای را بر روی دیواره‌ی سمت چپ حلقه باز نموده، با انتخاب گزینه‌ی Add Element المان‌های دیگری را به ترمینال سمت چپ ثبات انتقال اضافه کنید.

با فشار دادن دکمه‌ی STOP بر روی صفحه‌ی پانل، اجرای برنامه را متوقف کنید. اگر برنامه را به صورت تک مرحله‌ای اجرا می‌کنید، چند مرتبه دکمه‌های مربوط به این مد اجرایی را فشار دهید تا اجرای برنامه به پایان رسد.

۶- این برنامه را ببندید و از ذخیره نمودن تغییرات ایجاد شده خودداری کنید.

## دلیل استفاده از ثبات انتقال چیست؟

به شکل ۱۷-۶ توجه کنید. در شکل (الف) مجموع تکرارهای انجام شده در حلقه محاسبه می‌گردد. در هر بار تکرار حلقه، حاصل جمع جدید در ثبات انتقال ذخیره می‌شود. در پایان حلقه، مقدار حاصل جمع کلی یعنی عدد 45 در نشان‌دهنده‌ی عددی به نمایش در می‌آید. در حلقه‌ی (ب) از ثبات انتقال استفاده نمی‌شود و در نتیجه، ذخیره نمودن مقادیر میانی تکرارها به کمک این حلقه امکان‌پذیر نیست. در این حالت در هر یک از تکرارهای حلقه، عدد 0 با مقدار «i» جمع می‌شود و تنها مقدار نهایی 9 از این حلقه به دست می‌آید.




**شکل ۱۷-۶: به‌دست آوردن مقادیر متفاوت در یک حلقه در صورت استفاده یا عدم استفاده از ثبات انتقال**

4- First In First Out

۶

مبحث به کارگیری ثبات انتقال را با طرح یک سؤال ادامه می دهیم:

در حالتی که نیاز به محاسبه‌ی میانگین از مقادیر تکرارهای حلقه باشد، چه باید کرد؟ (به عنوان مثال ممکن است در یک فرآیند قصد داشته باشید در هر ثانیه دمای محیط را قرائت نموده، سپس میانگین این مقادیر را در عرض یک ساعت محاسبه کنید) با استفاده از طبیعت گرافیکی LabVIEW چگونه می توان بدون استفاده از ثبات انتقال، مقدار به دست آمده در یک تکرار حلقه را به تکرار بعدی مرتبط ساخت؟

پاسخ به این سؤال، همان دلیل استفاده نمودن از ثبات انتقال را در محیط برنامه‌نویسی LabVIEW روشن می‌سازد.

## انتساب مقدار اولیه به ثبات انتقال

برای اجتناب از عملکرد نادرست و دور از انتظار برنامه باید یک مقدار اولیه به ثبات انتقال اختصاص دهید، مگر اینکه بنا به دلایلی خاص این عمل را انجام ندهید و از عواقب آن اطلاع داشته باشید. برای اختصاص یک مقدار اولیه به ثبات انتقال، مقدار مورد نظر را از خارج از حلقه به ترمینال سمت چپ ثبات انتقال سیم کشی کنید.

در شکل ۱۸-۶ نحوه‌ی اتصال مقدار اولیه رابه ترمینال سمت چپ حلقه ملاحظه می کنید. در صورتی که برای ثبات انتقال، مقدار اولیه‌ای در نظر گرفته نشود، LabVIEW یک مقدار اولیه‌ی پیش‌فرض مطابق با نوع داده‌ی استفاده شده در اولین اجرای برنامه به آن نسبت می دهد. در مراحل بعدی تکرار حلقه، ثبات انتقال حامل مقادیر به‌دست آمده از تکراهای قبلی خواهد بود.

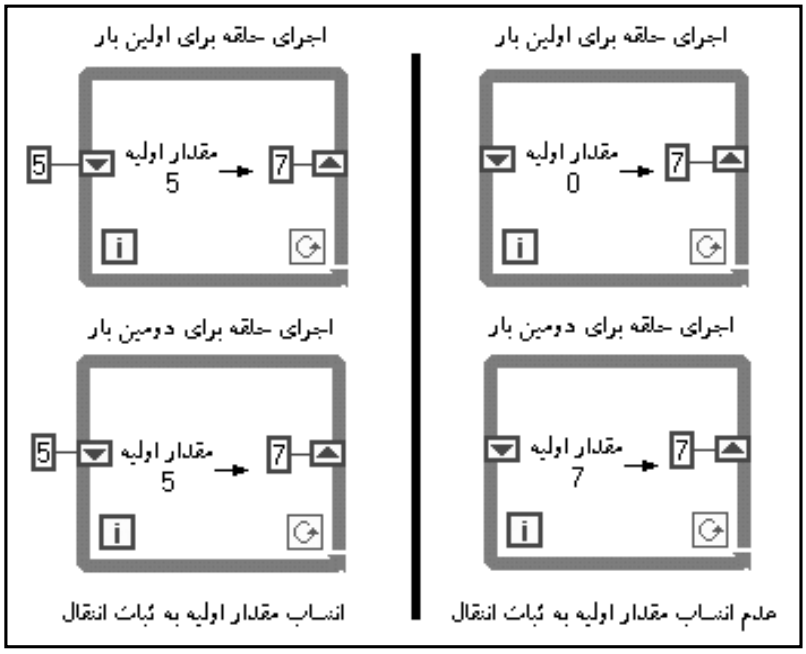
به عنوان مثال اگر داده های ثبات انتقال از نوع جبری باشند، در اولین اجرای حلقه، مقدار اولیه‌ی False به صورت پیش فرض برای ترمینال سمت چپ در نظر گرفته می شود. به طریق مشابه اگر داده های ثبات از نوع عددی باشند، مقدار اولیه‌ی 0 در اولین اجرای حلقه به آن اختصاص داده خواهد شد. در صورت اجرای حلقه برای دومین بار، ثباتهای انتقال حاوی مقادیر به دست آمده از اولین اجرا خواهند بود. شکل ۱۸-۶ را جهت درک بهتر عملکرد و مفهوم انتساب مقدار اولیه به ثبات انتقال ملاحظه کنید.

دو حلقه‌ی موجود در ستون سمت چپ عملکرد و محتوای یک ثبات انتقال را که مقدار اولیه‌ای به آن اختصاص داده شده است[۵] پس از دو تکرار نشان می دهند. در ستون سمت راست نیز عملکرد و محتوای یک ثبات انتقال را که برای آن مقدار اولیه ای در نظر گرفته نشده است[۶] پس از دو تکرار ملاحظه می کنید. به مقادیر اولیه‌ی در نظر گرفته شده برای ثباتهای انتقال که در دو حلقه‌ی ردیف بالا قرار دارند، توجه کنید.

---

۵- به این نوع ثبات در اصطلاح Initialized Shift Register گفته می شود.

۶- به این نوع ثبات در اصطلاح Uninitialized Shift Register گفته می شود.




شکل ۱۸-۶

LabVIEW مقادیر ذخیره شده در ثباتهای انتقال را تا زمانی که برنامه را نبسته و آن را از حافظه دور نکرده باشد، حذف نخواهد کرد. به بیان دیگر در صورت اجرای برنامه‌ای که در آن برای یک ثبات انتقال مقدار اولیه‌ای در نظر گرفته نشده باشد، مقدار اولیه‌ی لازم برای اجرای هر مرحله، از اجرای مرحله‌ی قبل به‌دست می‌آید. از این ویژگی به‌ندرت در محیط برنامه‌نویسی LabVIEW استفاده می‌شود؛ زیرا تحلیل مقادیر به‌دست آمده در این حالت بسیار مشکل خواهد بود.

## ساختار شرطی

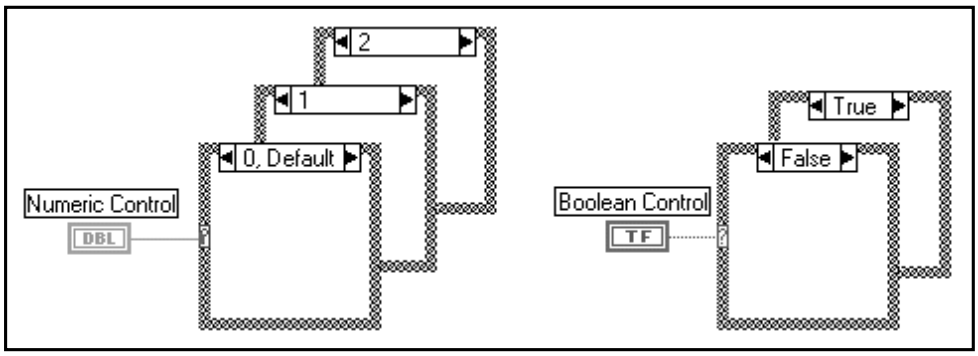
تا اینجا هرچه در مورد حلقه‌ها گفته‌ایم کافی است! اکنون قصد داریم در مورد کاربرد ساختارها سخن بگوییم. یک ساختار شرطی روشی برای اجرای متون شرطی در محیط LabVIEW است. عملکرد این ساختار نظیر دستور «if-then-else» در سایر زبانهای برنامه‌نویسی متداول متنی است. این ساختار در زیرپالت Functions >> Structures قرار دارد.


Selector Terminal

همان گونه که در شکل ۱۹-۶ ملاحظه می کنید این ساختار شرطی از چند شرط یا حالت تشکیل شده است که در اصطلاح به هریک از آنها Case گوییم. بسته به مقدار عددی یا جبری که به ترمینال انتخابی[۷] سیم کشی نموده‌اید تنها یکی از آنها به اجرا در می آید.

7- Selector Terminal

۶




**شکل ۱۹-۶: ساختار شرطی**

در صورتی که یک مقدار جبری به ترمینال انتخابی سیم‌کشی شده باشد، این ساختار دارای دو Case به صورت False و True می‌باشد. اگر یک مقدار عددی به ترمینال مذکور سیم‌کشی شده باشد این ساختار می‌تواند از 0 تا $2^N$-1 حالت یاCase داشته باشد. در مورد ساختارهایی که یک مقدار عددی به ترمینال انتخابی در آنها سیم‌کشی شده است، شماره‌ی Case فعلی در بالای چهارچوب نمایش داده می‌شود. در ابتدا تنها Caseهای با شماره‌ی 0 و 1 در دسترس هستند اما به سادگی می‌توانید تعداد آنها را اضافه کنید. هنگامی که برای اولین بار ساختار شرطی را بر روی صفحه‌ی پانل قرار می‌دهید به فرم جبری ظاهر می‌شود. بدین معنی که ترمینال انتخابی آن به صورت پیش‌فرض در موقعیت False قرار می‌گیرد. به محض سیم‌کشی یک مقدار عددی به ترمینال مذکور، ساختار شرطی مقادیر عددی را به خود می‌گیرد.

همان گونه که عنوان شد ساختار شرطی می‌تواند شامل چندین Case باشد. اما نظیر دسته‌ای از برگه‌های مرتب شده در هر لحظه تنها یکی از آنها قابل رؤیت است (برخلاف شکل قبل که در آن چندین Case در کنار یکدیگر به نمایش در آمده‌اند). با کلیک کردن بر روی پیکان کاهش و یا افزایش که به ترتیب درسمت چپ و راست در بالای چهارچوب قرار دارند می‌توان Caseهای قبلی و بعدی را (البته در صورت وجود) ملاحظه کرد. برای مشاهده‌ی تمامی Case های موجود در یک ساختار شرطی بر روی نشان‌دهنده‌ی بالای چهارچوب کلیک کرده، سپس Case مورد نظر را انتخاب نمایید. روش دیگر برای تغییر وضعیت به Caseهای موجود، انتخاب گزینه‌ی ...Show Case از منوی کرکره‌ای بر روی دیواره‌ی ساختار است.

در صورت اتصال یک عدد اعشاری به ترمینال انتخابی، LabVIEW آن مقدار را به نزدیک‌ترین عدد صحیح گرد می‌کند.

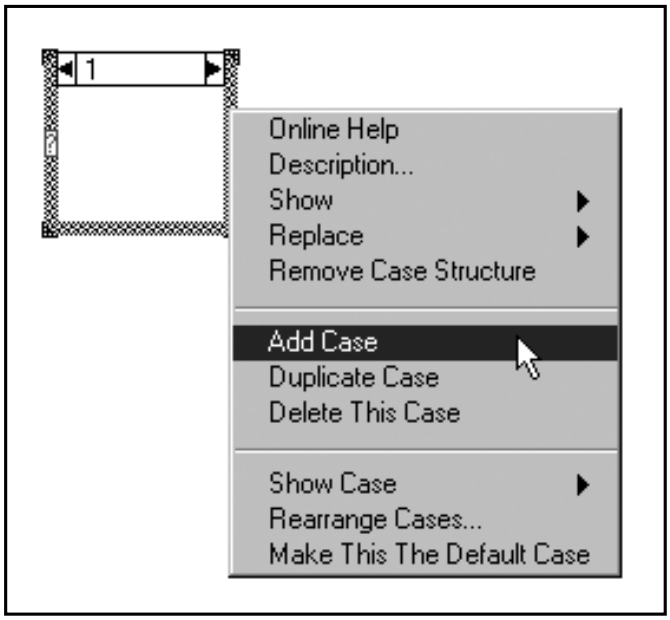
می‌توانید ترمینال انتخابی را در هر نقطه از دیواره‌ی سمت چپ درCase قرار دهید. پس از انجام این عمل ترمینال مذکور به صورت خودکار نوع داده‌ها را به خود می‌گیرد. اگر نوع داده‌ی سیم‌کشی شده به ترمینال انتخابی را از مقدار عددی به جبری تغییر دهید، Caseهای 0 و 1 به False و True تغییر می‌یابند. در صورت وجود Caseهای دیگر (از شماره‌ی 2 تا N)، LabVIEW آنها را حذف نمی‌کند. در این Caseها تغییر نوع داده به صورت تصادفی صورت می‌گیرد. قبل از اجرای ساختار باید این Caseها را حذف کنید.

## سیم کشی ترمینال های ورودی و خروجی

دستیابی به داده های ترمینال های ورودی ساختار شرطی (تونل ها و ترمینال انتخابی) در تمامی Caseها امکان پذیر است. لزومی ندارد که Caseها از داده های ورودی استفاده نموده، یا داده های خروجی تولید کنند. اما در صورتی که هر یک از Caseها یک مقدار خروجی تولید نماید، تمامی Caseها نیز باید همان مقدار خروجی را ایجاد کنند. هنگامی که یک خروجی را از یک Case سیم کشی نمودید، یک تونل سفیدرنگ در همان محل و در تمامی Caseها ظاهر می شود و تا زمانی که داده ای را به این تونل سیم کشی نکرده باشید دکمه‌ی Run به صورت پیکان شکسته ظاهر می گردد و بدین معنی است که قادر به انجام این برنامه نخواهید بود. پس از انجام سیم کشی، تونل مذکور به رنگ سیاه در می آید و دکمه‌ی Run نیز به حالت اولیه‌ی خود باز می گردد که بیانگر عدم وجود خطا در برنامه است.

## اضافه نمودن Case

در منوی کرکره‌ای بر روی دیواره‌ی Case چند گزینه جهت افزودن، کپی کردن و حذف نمودن Caseها تعبیه شده است. گزینه های موجود در این منوی را در شکل ۲۰-۶ ملاحظه می کنید. جهت اضافه کردن تعداد Caseها از گزینه‌ی Add Case استفاده نمایید. با انتخاب گزینه‌ی Duplicate Case می توانید Case فعلی را کپی کنید. در صورت انتخاب گزینه‌ی Remove Case Structure، تنها Case فعلی حذف می‌شود و المان های موجود در آن در محل فعلی خود باقی می مانند. در صورت انتخاب گزینه‌ی Delete This Case المان های موجود در آن نیز به همراه Case حذف خواهند شد.



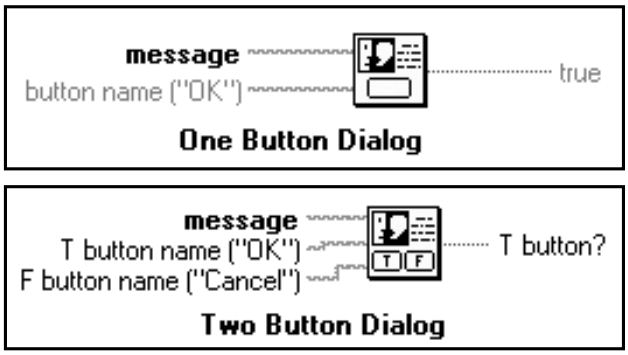

**شکل ۲۰-۶: گزینه های موجود در منوی کرکره‌ای بر روی دیواره‌ی Case**

۶

## پنجره‌های محاوره‌ای

لحظه‌ای سرعت خود را کم کنید. در این بخش قصد داریم موضوع بحث را تغییر دهیم و پنجره‌های محاوره‌ای مورد استفاده در محیط LabVIEW را بررسی کنیم؛ زیرا در تمرین بعدی از آنها استفاده خواهیم کرد.

هر یک از دستورهای One Button Dialog و Two Button Dialog یک پنجره‌ی محاوره‌ای را که حاوی پیام مورد نظر شماست ظاهر می سازند. این دو دستور در زیرپالت Functions >> Time & Dialog قرار دارند. در شکل ۲۱-۶ این دو دستور را ملاحظه می کنید.




**شکل ۲۱-۶: دو دستور تعبیه شده در محیط LabVIEW به منظور ایجاد پنجره‌های محاوره‌ای**

تا زمانی که بر روی دکمه‌ی OK در پنجره‌ی محاوره‌ای One Button Dialog کلیک نکنید، این پنجره بر روی صفحه باقی می ماند. در پنجره‌ی محاوره‌ای Two Button Dialog با کلیک کردن بر روی یکی از دو دکمه‌ی OK یا Cancel می توانید این پنجره را ببندید. همچنین با توجه به موارد استفاده و کاربرد این دکمه ها در برنامه های مختلف می توانید با وارد کردن نام مورد نظر در ورودی button name، نام آنها را تغییر دهید. در صورت باز بودن هر یک از این پنجره ها نمی توانید هیچ پنجره‌ی دیگری را در محیط LabVIEW باز کنید یا ببندید.

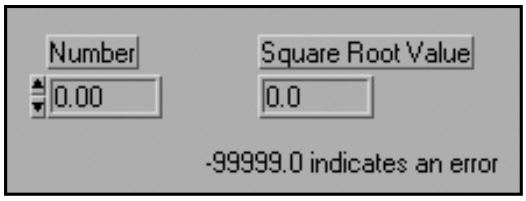
پنجره های مذکور برای ابلاغ پیام به درخواست کردن ورودی از اُپراتور برنامه بسیار مفید و کاربردی هستند. نظیر این پنجره ها را حتماً در سیستم عامل Windows ملاحظه نموده اید و اطلاع دارید تا زمانی که با کلیک کردن بر روی یکی از دکمه های OK یا Cancel این پنجره ها را نبسته اید، فعال کردن پنجره های دیگر امکان پذیر نیست.

## تمرین ۳-۶: محاسبه‌ی جذر یک عدد

در این تمرین، ساختار شرطی و پنجره های محاوره ای را مورد بررسی قرار می دهیم. در این برنامه قصد داریم تا ریشه‌ی دوم یک عدد مثبت را محاسبه نماییم. در صورتی که عدد ورودی، منفی باشد یک پنجره‌ی محاوره ای باز شده، یک پیغام خطا ظاهر می شود.

۱- با انتخاب گزینه‌ی File >> New یک برنامه‌ی جدید باز کنید.

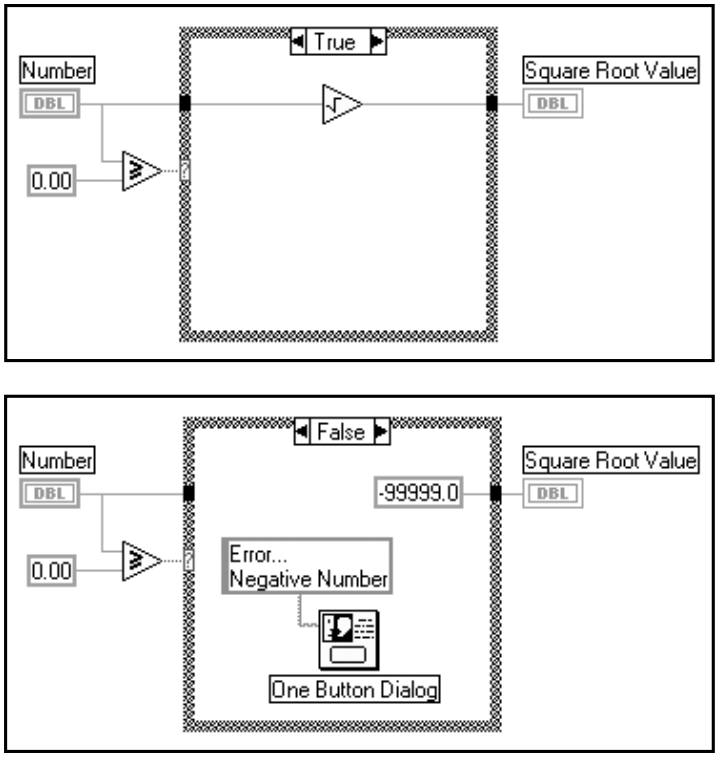
۲- صفحه‌ی پانل این برنامه را مطابق شکل ۲۲-۶ ایجاد نمایید.




**شکل ۲۲-۶: صفحه‌ی پانل در برنامه‌ی Square Root.vi**

عدد ورودی توسط المان Digital Control که با برچسب Number مشخص شده است در برنامه اعمال می‌شود و المان Digital Indicator با برچسب Square Root Value جذر عدد ورودی را نشان می دهد. بهتر است با انتخاب گزینه‌ی ...Format& Precision از منوی کرکره‌ای این المان، تعداد ارقام اعشاری آن را افزایش دهید.

۳- حال به صفحه‌ی نمودار بلوکی تغییر وضعیت دهید و دستورها و گره‌های نشان داده شده در شکل ۲۳-۶ را در این صفحه ایجاد کنید.




**شکل ۲۳-۶: صفحه‌ی نمودار بلوکی در برنامه‌ی Square Root.vi**

۴- ساختار شرطی یا Case Structure را از زیرپالت Structures << Functions انتخاب نموده، سپس آن را بر روی صفحه‌ی نمودار بلوکی قرار دهید. نظیر آنچه که در مورد For Loop و While Loop عنوان شد، پس از انتخاب این ساختار از زیرپالت مذکور، نشانگر ماوس را در نقطه‌ای از صفحه‌ی نمودار بلوکی که قصد دارید گوشه‌ی بالایی سمت چپ ساختار در آن محل قرار گیرد کلیک کرده، سپس ماوس را به صورت قطری حرکت دهید و بدین ترتیب اندازه‌ی چهارچوب آن را تعیین کنید.

دستور Greater Or Equal? یک مقدار جبری به دست می دهد. بنابراین ساختار شرطی در همان حالت

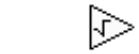
۶

از پیش تعیین شده‌ی خود یعنی حالت جبری باقی می‌ماند. به خاطر داشته باشید که در هر لحظه تنها می‌توانید یکی از Caseها را مشاهده کنید. برای مشاهده‌ی Caseهای دیگر بر روی پیکان‌های موجود در بالای چهارچوب کلیک کنید. در شکل ۲۳-۶ هر دو Case را ملاحظه می‌کنید. این تصویر بدین جهت به این صورت نشان داده شده است تا متوجه شوید که چه المان‌هایی در دو Case ایجاد نمایید.

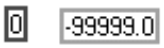
۵- المان‌های موجود در شکل ۲۳-۶ را از زیرپالت‌های مربوط انتخاب کنید و آنها را مطابق شکل سیم‌کشی کنید. برای سیم‌کشی ترمینال‌های ورودی و خروجی بهتر است از پنجره‌ی Help کمک بگیرید.

Greater Or Equal?
Function

دستور Greater Or Equal? را از زیرپالت Functions >> Comparison انتخاب کنید. دستور مذکور منفی بودن عدد ورودی را بررسی می‌کند. در صورتی که عدد ورودی بزرگتر یا مساوی صفر باشد، این دستور مقدار جبری True را در خروجی خود به دست می‌دهد.


Square Root
Function

دستور Square Root را از زیرپالت Functions >> Numeric انتخاب کنید. این دستور جذر عدد ورودی را به دست می‌دهد.

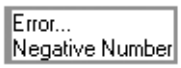


Numeric Constants

این دو المان ثابت عددی یا Numeric Constant را از زیرپالت Functions >> Numeric انتخاب کنید. در صورت بروز خطا یا به عبارت دیگر در صورت منفی بودن عدد ورودی، مقدار ثابت 99999.0- به عنوان خروجی دستور Square Root ظاهر می‌شود. عدد ثابت 0 نیز مبنایی برای تشخیص منفی بودن عدد ورودی است.

One Button
Dialog Function

دستور One Button Dialog را از زیرپالت Functions >> Time & Dialog انتخاب کنید. در این تمرین، پس از اجرای این دستور یک پنجره‌ی محاوره‌ای با پیغام Error... Negative Number ظاهر می‌شود.



String Constant

ثابت رشته‌ای یا String Constant را از زیرپالت Functions >> String انتخاب کنید. سپس به کمک یکی از دو ابزار Operating Tool یا Labeling Tool متن دلخواه را وارد نمایید. به عنوان مثال در این تمرین عبارت «Error... Negative Number» را وارد می‌کنید. در مورد رشته‌ها در فصل ۹ به تفصیل سخن خواهیم گفت.

پس از اعمال ورودی، این برنامه تنها یکی از دو حالت True یا False در ساختار شرطی را اجرا می‌کند. در صورتی که عدد ورودی بزرگتر یا مساوی صفر باشد، Case مربوط به حالت True اجرا می‌شود. در این حالت جذر عدد مذکور محاسبه می‌شود و نتیجه‌ی حاصل بر روی نشان‌دهنده‌ی Square Root Value در صفحه‌ی پانل به نمایش در می‌آید. اگر عدد ورودی منفی باشد، Case مربوط به حالت False اجرا می‌شود و بر روی صفحه‌ی نشان‌دهنده‌ی مذکور عدد ثابت 99999.0- ظاهر می‌گردد. در این حالت یک پنجره‌ی محاوره‌ای بر روی صفحه باز می‌شود که حاوی متن «Error... Negative Number» است.

به یاد داشته باشید که حتماً برای هر Case یک تونل خروجی تعیین کنید. به همین دلیل است که برای Case، خطای خروجی 99999.0- را در نظر گرفته‌ایم. پس از ایجاد تونل خروجی در Case، تونل مذکور در همان محل و در Caseهای دیگر نیز ظاهر می‌گردد. در صورتی که به یک تونل سیم‌کشی نشده باشد، به صورت مستطیل سفیدرنگ ظاهر می‌شود.

۶

۶- حال به صفحه‌ی پانل بازگردید و برنامه را اجرا کنید. هر دو دسته اعداد مثبت و منفی را در المان Number وارد نموده، نتیجه را ملاحظه کنید. شایان ذکر است که می‌توانید اعداد ورودی را با دو رقم اعشار وارد کنید و نتیجه را نیز تا یک رقم اعشار محاسبه نمایید.

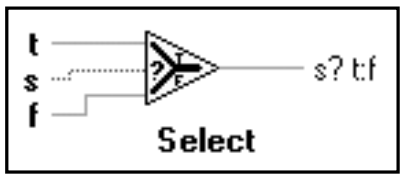
۷- این برنامه را با عنوان Square Root.vi درزیرفهرست LabVIEW\My Activity ذخیره نموده، سپس آن را ببندید.

عملکرد این برنامه معادل با کدهای فرضی زیر است:

```
if (Number <= 0) then
        Square Root Value = SQRT (Number)
        else
        Square Root Value = -99999.0
        Display Message Error... Negative Number
end if
```

## دستور Select

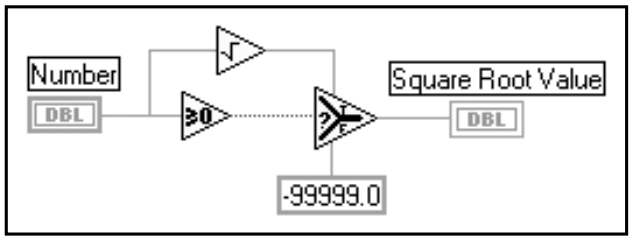
دستور Select در LabVIEW شبیه به دستور «if-then-else» در زبانهای برنامه‌نویسی متداول متنی است. عملکرد این دستور با ساختار شرطی نیز مشابه است. دستور مذکور در زیرپالت Functions >> Comparison قرار دارد.




**شکل ۲۴-۶: آیکن دستور Select به همراه اتصالات آن**

در صورتی که ورودی s در وضعیت True باشد، این دستور مقدار اتصال یافته به ورودی t را به خروجی اعمال می کند. اگر ورودی s در وضعیت False باشد، مقدار متصل شده به ورودی f، به خروجی اعمال می‌شود. از این دستور می‌توانید به جای ساختار شرطی در تمرین قبل استفاده کنید. با این تفاوت که در این حالت پنجره‌ی محاوره‌ای باز نمی‌شود. در شکل ۲۵-۶ برنامه‌ی محاسبه‌ی جذر یک عدد به کمک این دستور آمده است.

۶



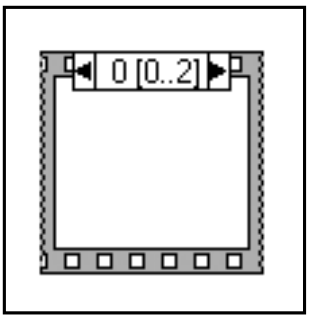

**شکل ۲۵-۶: روش محاسبه‌ی جذر یک عدد با استفاده از دستور Select**

## ساختار ترتیبی

تعیین نمودن ترتیب اجرای یک برنامه توسط آرایش دادن المان های آن در ترتیب ها و مراحل مشخص را در اصطلاح «جریانِ کنترل» گویند. این اصل، جزء جدانشدنی زبانهای برنامه نویسی متنی نظیر C و BASIC است، زیرا سطرهای برنامه به همان ترتیبی که در برنامه قرار دارند به اجرا در می آیند. در LabVIEW برای اجرای این اصل از ساختار ترتیبی یا Sequence Structure استفاده می شود. در این ساختار هر یک از مراحلی را که باید به ترتیب به اجرا درآیند یک فریم (Frame) گویند. در یک ساختار ترتیبی ابتدا Frame 0 به اجرا در آمده، سپس Frame 1، Frame 2 و به همین ترتیب الی آخر. تنها در صورت اجرای آخرین فریم، داده ها و اطلاعات حاصل از ساختار ترتیبی به دست می آیند.

ساختار ترتیبی نشان داده شده در شکل ۲۶-۶ شبیه به یک فریم فیلم است. این ساختار در زیرپالت Functions >> Structures قرار دارد. نظیر آنچه که در مورد ساختارهای شرطی ذکر شد در این مورد هم در هر لحظه تنها یک فریم قابل رؤیت است. برای مشاهده‌ی فریم های دیگر بر روی پیکان های قرار گرفته در بالای چهارچوب ساختار کلیک کنید و فریم مورد نظر را از لیست فریم های موجود انتخاب نمایید. روش دیگر برای انجام این عمل، باز نمودن منوی کرکره ای بر روی دیواره‌ی ساختار و انتخاب گزینه‌ی Show Frame... است.

هنگامی که برای اولین بار این ساختار را از زیرپالت Structures انتخاب می‌کنید و آن را بر روی صفحه‌ی نمودار بلوکی قرار می دهید، تنها دارای یک فریم است و در بالای این ساختار هیچ عدد یا پیکانی که نشانگر شماره‌ی این فریم و وجود فریم های دیگر باشد، دیده نمی شود. برای ایجاد فریم های جدید، منوی کرکره ای را بر روی دیواره‌ی ساختار باز نموده، یکی از دو گزینه‌ی Add Frame After یا Add Frame Berore را انتخاب کنید. همان گونه از نام این ساختار استنباط می شود، از ساختارهای ترتیبی جهت کنترل ترتیب اجرای گره هایی استفاده می شود که برای اجرا شدن به داده های گره های دیگر وابسته نیستند. ترتیب اجرای گره ها در داخل هر فریم توسط وابستگی داده ها به یکدیگر تعیین می شود. در شکل ۲۶-۶، Frame 0 در یک ساختار ترتیبی نشان داده شده است. اعداد موجود در ضلع فوقانی این فریم مشخص می کنند که این ساختار ترتیبی شامل سه فریم است.




**شکل ۲۶-۶: یکی از Caseهای ساختار ترتیبی**

برخلاف ساختارهای شرطی که هر Case باید یک خروجی داشته باشد در ساختارهای ترتیبی تونل های خروجی تنها می توانند یک منبع داشته باشند. این خروجی می تواند از هر یک از فریم ها نشأت گرفته باشد؛ اما به خاطر داشته باشید تنها در صورت انجام کامل ساختار، داده ها به خارج از آن انتقال می یابند و نه در زمانی که فریم های مجزا به پایان می رسند. درضمن داده‌ی مربوط به تونل ورودی برای تمامی فریم ها معتبر است.

## ترمینال ترتیبی محلی[8]

برای عبور داده‌ها از یک فریم به فریم‌های بعدی از ترمینالی به نام ترمینال ترتیبی محلی استفاده می‌شود. برای دستیابی به این ترمینال، منوی کرکره‌ای را بر روی چهارچوب ساختار باز نموده، گزینه‌ی Add Sequence Local را انتخاب کنید.

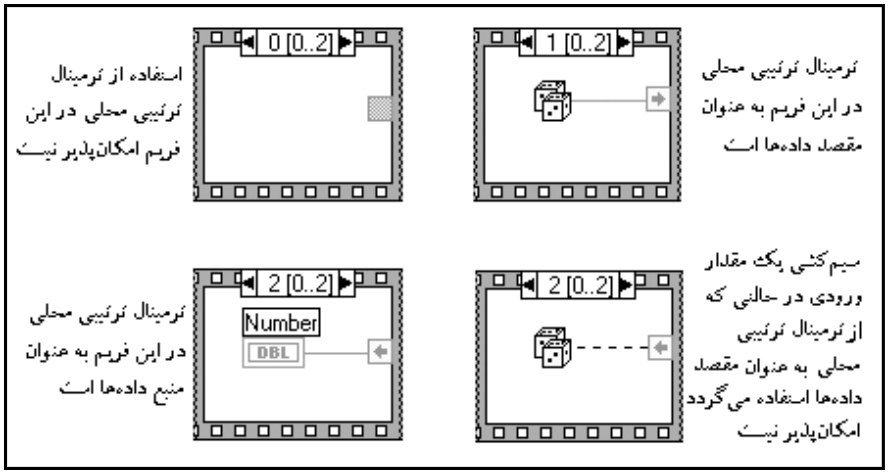
جابه‌جایی این ترمینال به هر نقطه ای از چهارچوب ساختار که توسط المان های دیگر اشغال نشده باشد امکان‌پذیر است. برای حذف این ترمینال، ابتدا آن را به کمک ابزار Positioning Tool انتخاب نموده، سپس کلید <Delete> را فشار دهید و یا منوی کرکره ای را بر روی آن باز نموده، گزینه‌ی Remove را انتخاب کنید.

هنگامی که این ترمینال برای اولین بار بر روی صفحه‌ی نمودار بلوکی قرار می گیرد، به صورت یک جعبه‌ی زردرنگ کوچک ظاهر می شود.

در شکل ۲۷-۶ ترمینال محلی در چند حالت نمایش داده شده است. هنگامی که داده ای را به این ترمینال سیم کشی می کنید، یک پیکان به سمت خارج چهارچوب ساختار بر روی ترمینال ظاهر می شود و نشان دهنده‌ی آن است که این ترمینال حاوی منبع داده است. این ترمینال در فریم های بعدی به صورت یک پیکان به سمت داخل چهارچوب فریم می باشد. در شکل ۲۷-۶ این مطلب به وضوح نشان داده شده است.

8- Sequence Local Terminal

۶

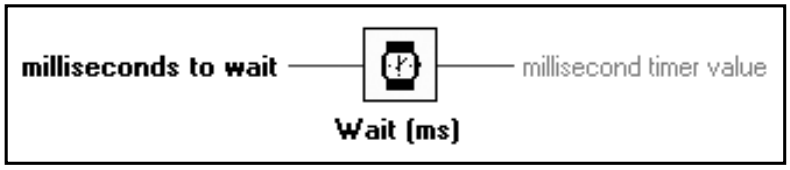



**شکل ۲۷-۶: شکل ظاهری ترمینال ترتیبی محلی در چند فریم از یک ساختار ترتیبی**

## زمان‌بندی در برنامه

در برخی موارد لازم است تا زمان اجرای برنامه را کنترل کنید. برای انجام این عمل می توان از گزینه های Wait (ms)، Tick Count (ms) و Wait Until Next ms Multiple استفاده نمود. گزینه‌های مذکور در زیرپالت Functions >> Time & Dialog قرار دارند. در تمامی گزینه‌های مذکور، زمان تأخیر برحسب میلی‌ثانیه تعیین می‌شود.

به کارگیری گزینه‌ی Wait (ms) سبب می شود که اجرای برنامه با تأخیر انجام گیرد.




**شکل ۲۸-۶: آیکن دستور Wait (ms) به همراه اتصالات آن**

دستور Wait Until Next ms Multiple باعث می‌شود LabVIEW تا زمانی که کلاک داخلی مساوی یا بزرگتر از مقدار ورودی millisecond multiple نشده است قبل از اجرا نمودن برنامه منتظر بماند. این دستور جهت اجرا نمودن حلقه‌ها در فواصل زمانی مشخص و همچنین همزمان کردن فعالیت‌ها بسیار مفید و کاربردی است. این دو دستور با یکدیگر مشابهند اما یکسان نمی‌باشند. به عنوان مثال دستور Wait Until Next ms Multiple بسته به مقدار کلاک در هنگام شروع ممکن است کمتر از زمان مشخص شده در اولین حلقه منتظر بماند.

**شکل ۲۹-۶: آیکن دستور Wait Until Next ms Multiple به همراه اتصالات آن**

دستور Tick Count (ms) مقدار کلاک داخلی سیستم عامل رایانه را بر حسب میلی ثانیه به دست می دهد.

**شکل ۳۰-۶: آیکن دستور Tick Count (ms) به همراه اتصالات آن**

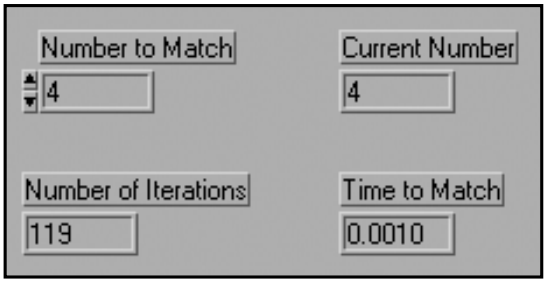
همان گونه که در مثال بعد ملاحظه خواهید کرد این دستور معمولاً جهت محاسبه‌ی زمان سپری شده مورد استفاده قرار می گیرد. توجه داشته باشید که کلاک داخلی همواره دقت کافی را ندارد. حداکثر زمان یک Tick در کلاک در سیستم های Windows 3.1، Windows 95/NT و Mac به ترتیب ۵۵، ۱۰ و ۱۷ میلی ثانیه است. با توجه به محدودیت مذکور در سیستم های عامل، عملکرد LabVIEW بسیار دقیق نخواهد بود.

## تمرین ۴-۶: تطبیق اعداد

حال که با مفاهیم ساختارها و زمان بندی آشنا شدید، فرصت دارید تا این تمرین را با استفاده از ساختارهای ترتیبی و یکی از توابع زمان‌بندی انجام دهید. در این تمرین قصد داریم تا با ایجاد یک برنامه مدت زمانی را که سپری می شود تا عدد تصادفی تولید شده در برنامه با یک عدد ورودی برابر شود محاسبه نماییم. در صورت نیاز به زمان‌سنجی در برنامه های مختلف می توانید الگوریتم به کار برده شده در این تمرین را به کار برید.

۱- با انتخاب گزینه‌ی File >> New یک برنامه‌ی جدید باز کنید.

۲- صفحه‌ی پانل این برنامه را مطابق شکل ۳۱-۶ ایجاد کنید.

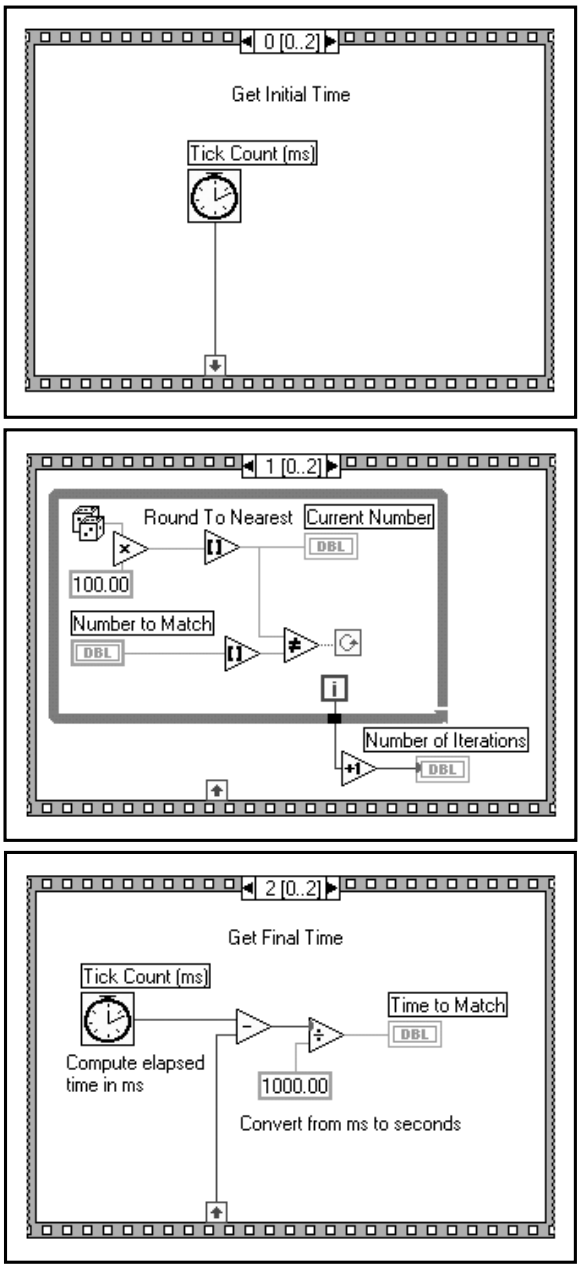



**شکل ۳۱-۶: صفحه‌ی پانل در برنامه‌ی Time to Match.vi**

۳- تعداد ارقام اعشار در المان های Number to Match، Current Number و Number of Iterations را با باز نمودن منوی کرکره ای بر روی هر یک و انتخاب گزینه‌ی ...Format & Precision به 0 کاهش دهید. بدین ترتیب در المان های مذکور، رقم اعشار حذف می‌شود و به آنها تنها عدد صحیح بدون نقطه‌ی اعشاری نسبت می دهید. در صورتی که از یک رایانه‌ی با سرعت بالا استفاده می کنید تعداد ارقام اعشار در المان Time to Match را به 4 یا 6 افزایش دهید.

۴- حال به صفحه‌ی نمودار بلوکی تغییر وضعیت دهید و فریم های نشان داده شده در شکل ۳۲-۶ را ایجاد کنید.

۶




**شکل ۳۲-۶: فریم‌های موجود در صفحه‌ی نمودار بلوکی برنامه‌ی Time to Match.vi**

۵- ساختار ترتیبی یا Sequence Structure (Functions >> Structures) را بر روی صفحه‌ی نمودار بلوکی قرار دهید. نظیر آنچه که در مورد For Loop و While Loop عنوان شد، پس از انتخاب این ساختار از زیرپالت مذکور، نشانگر ماوس را در نقطه‌ای از صفحه‌ی نمودار بلوکی که قصد دارید گوشه‌ی بالایی سمت چپ ساختار در آن محل قرار گیرد کلیک کرده، سپس ماوس را به صورت قطری حرکت داده و بدین ترتیب اندازه‌ی چهارچوب آن را تعیین کنید.

همان‌گونه که در شکل ۳۲-۶ نشان داده شده است در این تمرین باید سه فریم مجزا در ساختار ترتیبی ایجاد نمایید. جهت ایجاد یک فریم جدید، منوی کرکره‌ای را بر روی چهارچوب فریم باز نموده، گزینه‌ی After

Add Frame را انتخاب کنید.

۶- حال منوی کرکره‌ای را بر روی ضلع پایینی چهارچوب Frame 0 باز نموده، با انتخاب گزینه‌ی Add Sequence Local ترمینال ترتیبی محلی را ایجاد کنید. در این حالت ترمینال مذکور به صورت یک مستطیل خالی ظاهر می‌شود. پیکان‌های قرار گرفته در داخل این مستطیل پس از سیم‌کشی این ترمینال به صورت خودکار ظاهر خواهند شد.

۷- سایر المان‌های موجود در شکل ۳۲-۶ را در صفحه‌ی نمودار بلوکی قرار دهید. در ادامه به ارائه‌ی توضیح در مورد برخی از دستورهای استفاده شده در این تمرین می‌پردازیم. در هنگام سیم‌کشی جهت نمایش دادن ترمینال‌های ورودی و خروجی بهتر است از پنجره‌ی Help استفاده کنید.

Tick Count Function

دستور Tick Count (ms) (Functions >> Time & Dialog) مقدار کلاک داخلی رایانه را به دست می‌دهد.

Random Number (0-1) Function

دستور Random Number (0-1) (Functions >> Numeric) در هر یک از تکرارهای حلقه‌ی While، یک عدد تصادفی در فاصله‌ی [0,1] تولید می‌کند.

Multiply Function

دستور Multiply (Functions >> Numeric) در هر یک از تکرارهای حلقه‌ی While، عدد تصادفی تولید شده را در مقدار ثابت 100 ضرب می‌کند. واضح است که حاصل این عملیات، یک عدد تصادفی در فاصله‌ی [0,100] خواهد بود.

Round to Nearest Function

دستور Round to Nearest (Functions >> Numeric) عدد تصادفی حاصل از دستور Multiply را به نزدیکترین عدد صحیح گرد می‌کند.

Not Equal? Function

دستور Not Equal? (Functions >> Comparison) عدد حاصل از دستور Round to Nearest را با عدد تعیین شده در المان Number to Match مقایسه می‌کند و در صورتی که این دو عدد نامساوی باشند مقدار True را در خروجی خود به‌دست می‌دهد. در غیر این صورت حاصل این دستور، False خواهد بود.

Increment Function

دستور Increment (Functions >> Numeric) برای تولید مقدار Number of Iterations، یک واحد به ترمینال شمارشی حلقه اضافه می‌کند. همان گونه که می‌دانید اندیس گذاری حلقه‌ها از عدد صفر انجام می‌شود؛ لذا برای به دست آوردن مقدار صحیح تعداد تکرارهای حلقه یا Number of Iterations، یک واحد به ترمینال شمارشی حلقه افزوده می‌شود.

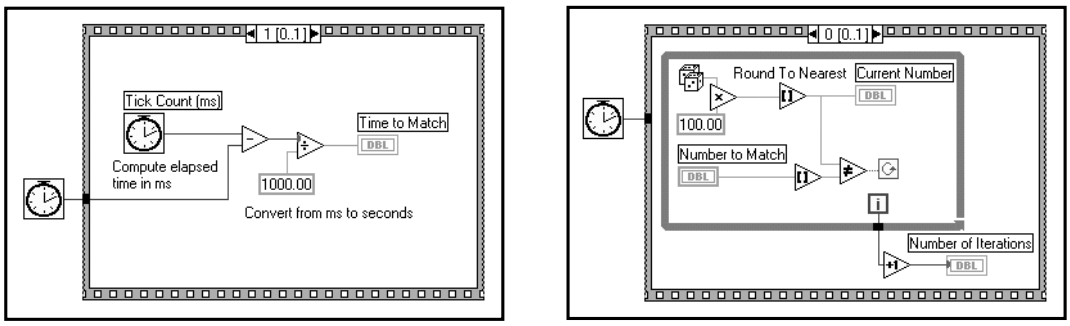
۶

عملکرد این برنامه به شرح زیر است:

در Frame 0، دستور Tick Count (ms) مقدار کلاک داخلی رایانه را برحسب میلی‌ثانیه به‌دست می‌دهد. این مقدار به ترمینال ترتیبی محلی موجود در این فریم سیم‌کشی می‌شود و دستیابی به آن در فریم‌های بعدی نیز امکان‌پذیر است.

در Frame 1 تا زمانی که عدد تعیین شده در صفحه‌ی پانل با عدد حاصل از دستور Round to Nearest مساوی نباشد، حلقه‌ی While به طور متناوب اجرا می‌شود. در Frame 2 دستور Tick Count (ms) زمان جدیدی را برحسب میلی‌ثانیه به‌دست می‌دهد. همان گونه که ملاحظه می‌کنید در این فریم، جهت به‌دست آوردن زمان سپری شده، تفاضل زمان جدید و زمان قبلی که از Frame 0 و از طریق ترمینال ترتیبی محلی به‌دست آمده است محاسبه می‌گردد. سپس عدد به‌دست آمده برای تبدیل میلی‌ثانیه به ثانیه بر عدد ثابت 1000 تقسیم می‌شود.

با استفاده از یک ساختار ترتیبی نیز می‌توان این تمرین را انجام داد و زمان سپری شده را محاسبه نمود. این برنامه در شکل ۳۳-۶ نشان داده شده است. همان گونه که ملاحظه می‌کنید این ساختار شامل دو فریم است و در این روش از ترمینال ترتیبی محلی استفاده نشده است.




**شکل ۳۳-۶: ایجاد صفحه‌ی نمودار بلوکی برنامه‌ی Time to Match.vi با استفاده از دو فریم**

۸- با کلیک کردن بر روی دکمه‌ی Execution Highlighting مد اجرای متمایز را در این برنامه فعال کرده، سپس با استفاده از دکمه‌ی Run این برنامه را به اجرا در آورید. همان گونه که قبلاً ذکر شد سرعت اجرای برنامه در این حالت کاهش می‌یابد و برای مشاهده‌ی روند مقادیر تولید شده بر روی صفحه‌ی پانل فرصت کافی در اختیار دارید. واضح است که زمان به‌دست آمده در این حالت بیانگر زمان مطلوب نخواهد بود.

۹- یک عدد صحیح مثبت در فاصله‌ی [0,100] را در المان Number to Match وارد نموده، برنامه را اجرا کنید. در صورتی که قصد دارید سرعت عملکرد المان‌ها و اجرای برنامه را افزایش دهید با کلیک کردن مجدد بر روی دکمه‌ی Execution Highlighting برنامه را در حالت عادی به اجرا در آورید.

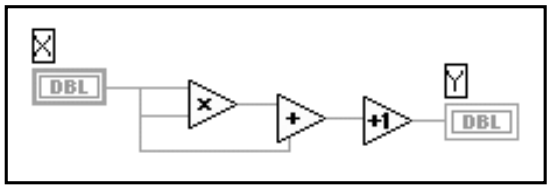
۱۰- این برنامه را با عنوان Time to Match.vi در زیرفهرست LabVIEW\My Activity ذخیره نموده، سپس آن را ببندید.

۶

## گره‌ی فرمولی

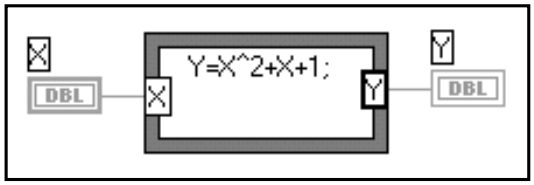
حال که تا اندازه‌ای با چهار ساختار کنترلی در محیط LabVIEW آشنا شدید، به معرفی ساختاری می‌پردازیم که تأثیری بر روند برنامه ندارد. گره‌ی فرمولی شامل یک چهارچوب است که قابلیت تغییر اندازه دارد و با استفاده از آن می‌توانید فرمول‌های ریاضی را مستقیماً در صفحه‌ی نمودار بلوکی وارد نمایید. به کارگیری این ساختار در مواردی که نیاز به حل یک معادله‌ی پیچیده داشته باشید نیز بسیار مفید و کاربردی است. به عنوان مثال معادله‌ی بسیار ساده‌ی $Y = X^2 + X + 1$ را در نظر بگیرید.

در صورتی که قصد داشته باشید برای وارد نمودن این تساوی از دستورهای معمول و توابع محاسباتی متداول در LabVIEW استفاده کنید دنبال نمودن آن در صفحه‌ی نمودار بلوکی اندکی مشکل‌تر از حالتی است که این معادله را با استفاده از فرمول‌های ریاضی وارد می‌کنید. در شکل ۶-۳۴ نحوه‌ی وارد نمودن این تساوی در صفحه‌ی نمودار بلوکی را با استفاده از توابع و دستورهای متداول در محیط برنامه‌نویسی LabVIEW ملاحظه می‌کنید.




**شکل ۶-۳۴: نحوه‌ی وارد نمودن تساوی $Y = X^2 + X + 1$ در صفحه‌ی نمودار بلوکی با استفاده از توابع و دستورهای متداول در محیط برنامه‌نویسی LabVIEW**

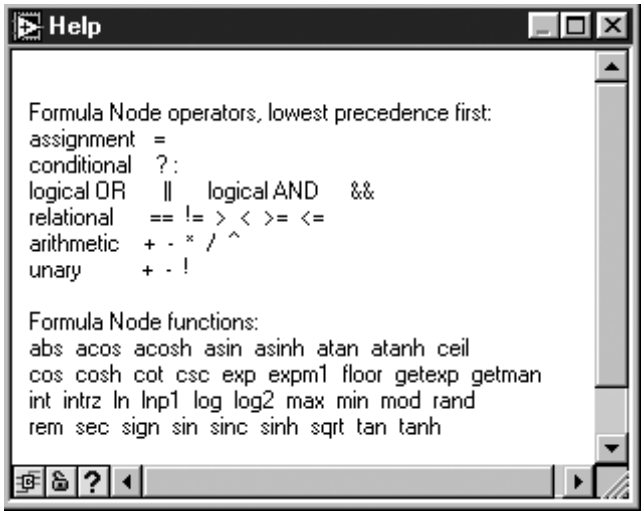
همین تساوی با استفاده از گره‌ی فرمولی مطابق شکل ۶-۳۵ اعمال می‌گردد.




**شکل ۶-۳۵: وارد نمودن تساوی $Y = X^2 + X + 1$ در صفحه‌ی نمودار بلوکی با استفاده از گره‌ی فرمولی**

به کمک گره‌ی فرمولی می‌توانید به جای ایجاد بخشهای پیچیده در صفحه‌ی نمودار بلوکی، فرمول‌های ریاضی را مستقیماً وارد کنید. برای انجام این عمل کافی است فرمول مورد نظر را در داخل چهارچوب ساختار گره‌ی فرمولی وارد نمایید. برای تعیین متغیرهای ورودی یا خروجی (در این تساوی متغیرهای ورودی و خروجی به ترتیب X و Y می‌باشند) کافی است یکی از دو گزینه‌ی Add Input یا Add Output را از منوی کرکره‌ای بر روی چهارچوب ساختار گره‌ی فرمولی انتخاب کنید. سپس نام متغیرها را در جعبه‌های ورودی و خروجی وارد نمایید. در انتهای هر فرمول حتماً باید از کاراکتر «;» استفاده شود. گره‌ی فرمولی در زیرپالت Functions >> Structures قرار دارد. در شکل ۶-۳۶ عملگرهای قابل استفاده در گره‌ی فرمولی را ملاحظه

می کنید. فهرست تمامی توابع ریاضی، مثلثاتی و... که در گره‌ی فرمولی قابل استفاده‌اند در فایل Appendix 2 << Appendices آورده شده است.

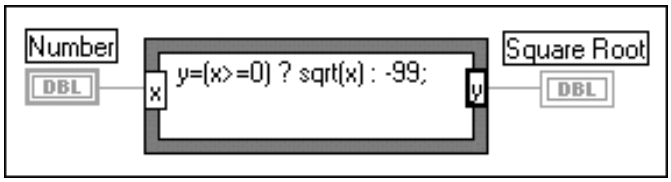



**شکل ۳۶-۶: عملگرهای قابل استفاده در گره‌ی فرمولی**

در مثال زیر عملیات یک شاخه‌ی شرطی نشان داده شده است. این شاخه‌ی شرطی را می توانید با استفاده از گره‌ی فرمولی برنامه‌نویسی کنید. کدهای فرضی زیر را در نظر بگیرید (نظیر تمرین ۳-۶). همان گونه که ملاحظه می کنید در صورت مثبت بودن عدد x، جذر آن محاسبه شده و این مقدار به متغیر y نسبت داده می شود. اگر x عددی منفی باشد مقدار 99- به y نسبت داده می شود.

```
if (x >= 0) then
    y = sqrt (x)
else
    y = -99
end if
```

در شکل ۳۷-۶ نحوه‌ی به کارگیری گره‌ی فرمولی جهت محاسبه‌ی جذر متغیر x نشان داده شده است. این برنامه معادل با کدهای فرضی فوق نوشته شده است.



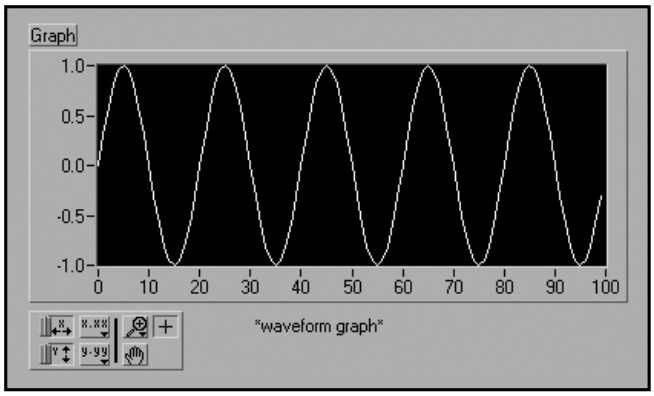

**شکل ۳۷-۶: نحوه‌ی به کارگیری گره‌ی فرمولی جهت محاسبه‌ی جذر متغیر x**

## تمرین ۵-۶: ترسیم منحنی $y = \sin(x)$

در این تمرین قصد داریم تا معادله‌ی y = sin (x) را با استفاده از ساختار گره‌ی فرمولی اعمال نموده، منحنی آن را رسم کنیم.

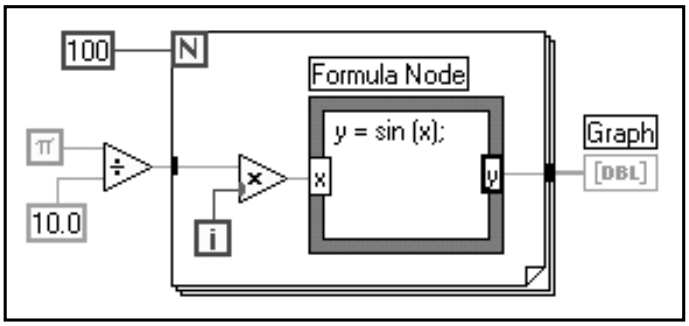
۱- یک برنامه‌ی جدید باز کنید. سپس گزینه‌ی Waveform Graph را از زیرپالت Controls >> Graph انتخاب کنید و آن را بر روی صفحه‌ی پانل قرار دهید. سپس آن را با برچسب Graph نام‌گذاری کنید. منحنی معادله‌ی y = sin (x)

بر روی صفحه‌ی این گراف به نمایش در می آید. در فصل ۸ در مورد گراف‌ها به تفصیل سخن خواهیم گفت. انجام این تمرین بدون نمایش تصویری گراف بسیار مضحک و بیهوده به نظر می‌رسد. بنابراین در این برنامه قصد داریم تا قبل از معرفی و بررسی گراف‌ها از آنها استفاده کنیم. با استفاده از گزینه‌ی Show در منوی کرکره‌ای گراف، بخش Legend را مخفی کنید. سپس اندازه‌ی صفحه را به صورت افقی افزایش دهید و با کمک بخش Palette یا منوی کرکره‌ای، گزینه‌های AutoScale X و AutoScale Y را فعال کنید.




**شکل ۳۸-۶: صفحه‌ی پانل در برنامه‌ی Formula Node Exercise.vi**

۲- صفحه‌ی نمودار بلوکی این برنامه را مطابق شکل ۳۹-۶ ایجاد کنید.




**شکل ۳۹-۶: صفحه‌ی نمودار بلوکی در برنامه‌ی Formula Node Exercise.vi**

به کمک Formula Node (Functions >> Structures) می توانید فرمول های ریاضی را مستقیماً در صفحه‌ی نمودار بلوکی وارد کنید. جهت ایجاد متغیر ورودی، منوی کرکره‌ای را بر روی دیواره‌ی ساختار گره‌ی فرمولی باز کنید و گزینه‌ی Add Input را انتخاب نمایید. برای ایجاد متغیر خروجی نیز به طریق مشابه گزینه‌ی Add Output را از همان منو انتخاب کنید. نام این متغیرها باید دقیقاً با متغیرهای استفاده شده در فرمول هماهنگی داشته باشد.

Pi Constant

عدد ثابت $\pi$ در زیرپالت Functions >> Numeric >> Additional Numeric Constants قرار دارد. در خلال هر یک از تکرارهای حلقه، ترمینال تکرار در عدد ثابت ($\pi/10$) ضرب می شود. نتیجه‌ی حاصل ضرب این دو مقدار به گره‌ی فرمولی سیم کشی می شود و مقدار سینوس آن محاسبه می گردد. سپس نتیجه در یک آرایه در دیواره‌ی For Loop ذخیره می شود. در فصل ۷ با آرایه ها آشنا

خواهیدشد. آرایه‌ی مذکور پس از اتمام اجرای حلقه‌ی For ترسیم می شود.

۳- حال به صفحه‌ی پانل بازگردید و برنامه را اجرا کنید. توجه داشته باشید که در این تمرین به جای به‌کارگیری گره‌ی فرمولی می‌توانید از تابع Sine که در زیرپالت Functions >> Numeric >> Trigonometric قرار دارد نیز استفاده کنید. اما این نکته را نیز به خاطر داشته باشید که در محیط LabVIEW برای هر فرمول مورد نیاز شما یک دستور از پیش نوشته شده وجود ندارد.

۴- این برنامه را با عنوان Formula Node Exercise.vi در زیرفهرست LabVIEW\My Activity ذخیره کنید.

عملکرد این برنامه معادل با کدهای فرضی زیر است:

```
for i = 0 to 99
    x = i * (π/10)
    y = sin (x)
    array [i] = y
next i
Graph (array)
```

برای درک بهتر مطالب این فصل، مثالهای ارائه شده در مورد ساختارها را بررسی نمایید. این مثالها در مسیر LabVIEW\Examples\General\structs.llb قرار دارند.

## خلاصه

در نرم‌افزار LabVIEW برای تکرار بخشی از صفحه‌ی نمودار بلوکی، دو حلقه‌ی For و While تعبیه شده است. هر دو حلقه شامل چهارچوبی هستند که قابلیت تغییر اندازه دارد و برای تکرار بخش مورد نظر باید المان‌های آن، در داخل چهارچوب مذکور قرار گیرند. اجرای While Loop تا زمانی ادامه می یابد که ترمینال شرطی در حالت True قرار داشته باشد، حال آن‌که تعداد تکرارها در For Loop عدد ثابت و مشخصی است و توسط کاربر تعیین می گردد.

جهت انتقال مقادیر از انتهای یک تکرار حلقه به ابتدای تکرار حلقه‌ی بعدی در For Loop و While Loop از المانی به نام ثبات انتقال استفاده می شود. از این المان ها می توانید برای دستیابی به مقادیر چندین تکرار قبل نیز استفاده کنید. برای فراخوانی و یا دستیابی به مقادیر به‌دست آمده در هر تکرار حلقه، باید یک المان جدید به ترمینال سمت چپ آن اضافه کنید. در ضمن جهت ذخیره‌ی چند متغیر در حلقه می توانید از چند ثبات انتقال استفاده نمایید.

در بسته‌ی نرم‌افزاری LabVIEW جهت کنترل نمودن جریان داده‌ها دو ساختار شرطی و ترتیبی در نظر گرفته شده است. در هر لحظه تنها یک Case یا یک فریم از ساختارهای مذکور قابل رؤیت است. برای مشاهده‌ی Caseها و فریم های دیگر، با استفاده از ابزار Operating Tool بر روی پیکان های قرار گرفته در بالای چهارچوب ساختار کلیک کنید و یا از گزینه های موجود در منوی کرکره‌ای کمک بگیرید.

بسته به ورودی اعمال شده به ترمینال انتخابی در ساختار ترتیبی می توان از آن برای انشعاب به بخشهای موجود در صفحه‌ی نمودار بلوکی استفاده نمود. این عمل شبیه به استفاده از دستور «if- then- else» در زبانهای

۶

برنامه نویسی متنی است. بخشی از صفحه‌ی نمودار بلوکی را که قصد اجرای آن را دارید در داخل چهارچوب هر یک از Caseها قرار داده، سپس یک ورودی به ترمینال انتخابی هر Case سیم کشی کنید. ساختار شرطی می تواند یکی از دو حالات جبری یا عددی را اختیار کند. زمانی که یک المان جبری یا عددی را به ترمینال انتخابی ارتباط می دهید، LabVIEW به صورت خودکار نوع داده ها را تشخیص می دهد.

در برخی از موارد اصول حاکم بر «جریانِ داده» دقیقاً عملکردی را که از برنامه انتظار دارید برآورده نمی سازند و شما نیاز به تعیین اولویت را در حین اجرای برنامه احساس می کنید. ساختارهای ترتیبی به شما کمک می کنند تا اولویت اجرای دستورها و توابع موجود در برنامه را تعیین کنید. در ساختار ترتیبی بخشی از صفحه‌ی نمودار بلوکی را که باید در ابتدا به اجرا در آید در اولین فریم یا Frame 0 قرار دهید. بخش بعدی را در Frame 1 قرار دهید و به همین ترتیب الی آخر. برای تبادل مقادیر بین فریم های ساختارهای ترتیبی از ترمینال ترتیبی محلی استفاده می گردد. داده های یک ترمینال ترتیبی محلی تنها در فریم های بعدی قابل دسترسی است نه در فریم های قبل از آن.

به کمک گره‌ی فرمولی می توان فرمول های ریاضی را مستقیماً در صفحه‌ی نمودار بلوکی وارد نمود. این قابلیت در مورد معادلات و توابع ریاضی پیچیده بسیار مفید است. به یاد داشته باشید که هر فرمول باید با کاراکتر «;» خاتمه یابد.

زیرپالت Functions >> Time & Dialog حاوی دستورهایی است که به کمک آنها می توان پنجره های محاوره‌ای را که شامل پیام های دلخواه کاربر می باشد باز نمود. همچنین در این زیرپالت چندین دستور برای زمان بندی برنامه وجود دارد. با انتخاب گزینه های One Button Dialog و Two Button Dialog پنجره هایی باز می شوند که در آنها از کلیدهای OK یا Cancel استفاده شده است و می توانید پیغام دلخواه خود را وارد نمایید. تا زمانی که با کلیک کردن بر روی یکی از دو دکمه‌ی OK یا Cancel این پنجره ها را نبندید، باز نمودن پنجره های دیگر در محیط LabVIEW امکان پذیر نخواهد بود.

دستور Wait (ms) باعث ایجاد تأخیر در اجرای برنامه می گردد. دستور Wait Until Next ms Multiple سبب می شود تا حلقه ها در یک فاصله‌ی زمانی مشخص همراه با وقفه به اجرا در آیند. این وقفه تا زمانی ادامه می یابد که کلاک داخلی مساوی یا بزرگتر از مضربی از مقدار ورودی گردد. این دو دستور مشابهند ولی عملکرد آنها یکسان نیست. لزومی ندارد که اکنون اختلاف این دو دستور را به طور کامل بدانید. دستور Tick Count (ms) مقدار کلاک داخلی رایانه را به دست می دهد. در تمامی این دستورها مقدار تأخیر یا کلاک داخلی بر حسب میلی ثانیه است.

## تمرینات اضافی

### تمرین ۶-۶: بررسی معادلات و تساوی ها

در این تمرین قصد داریم تا با استفاده از گره‌ی فرمولی معادلات زیر را در صفحه‌ی نمودار بلوکی وارد کنیم:

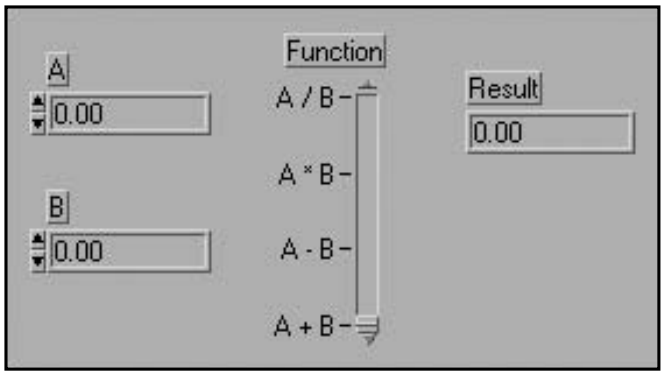
$$y_1 = x^3 + x^2 + 5$$
$$y_2 = m*(x) + b$$

۶

برای هر دو تساوی تنها از یک گره‌ی فرمولی استفاده کنید. به یاد داشته باشید که در انتهای هر فرمول کاراکتر «;» را به کار برید. سپس این برنامه را با عنوان Equations.vi در زیرفهرست LabVIEW\My Activity ذخیره کنید.

## تمرین ۷-۶: طراحی یک ماشین حساب ساده

در این تمرین سعی داریم تا با ایجاد یک برنامه،چهار عمل اصلی ریاضی را به کمک یک ماشین حساب بسیار ساده انجام دهیم. جهت وارد کردن دو عدد ورودی و به نمایش در آوردن مقدار حاصل از عملیات مورد نظر، از دو المان Digital Control و یک المان Digital Indicator در صفحه‌ی پانل استفاده می گردد. عملیاتی که بر روی دو عدد ورودی انجام می گیرد توسط لغزاندن یک دکمه‌ی لغزشی انتخاب می شود. برای آغاز برنامه نویسی از صفحه ای مطابق شکل ۴۰-۶ استفاده کنید.




**شکل ۴۰-۶: صفحه‌ی پانل در برنامه‌ی Calculator.vi**

این برنامه را با عنوان Calculator.vi در زیرفهرست LabVIEW\My Activity ذخیره کنید.

برای انجام این تمرین به دستور Text Labels در مورد کلید لغزشی یا Slide Control نیاز دارید. برای تعیین یکی از اعمال ( / , * , - , + ) منوی کرکره‌ای را بر روی کلید لغزشی باز نموده، سپس گزینه‌ی Text Labels را انتخاب کنید. در صورت عدم مشاهده‌ی گزینه‌ی مذکور در منوی کرکره‌ای، به احتمال زیاد منوی کرکره‌ای را در قسمتهای دیگر کلید لغزشی نظیر بخش درجه‌بندی آن باز کرده‌اید. کلیدهای لغزشی به همراه Text Labels شبیه به منوهای حلقوی متنی عمل می‌کنند. هنگامی که برای اولین بار Text Labels را انتخاب می‌کنید، تنها دارای دو مقدار max و min می‌باشد. به کمک ابزار Labeling Tool می‌توانید این عبارات را به دلخواه تغییر دهید. جهت اضافه کردن متون دیگر، منوی کرکره‌ای را بر روی صفحه‌ی نشان دهنده‌ی Text باز نموده، پس از انتخاب یکی از دو گزینه‌ی Add Item After یا Add Item Before، متن مورد نظر برای نشان‌دهنده را وارد کنید.

## تمرین ۸-۶: کنترل اجرای برنامه توسط حلقه‌ی While

با استفاده از یک حلقه‌ی While برنامه‌ای ایجاد کنید که عملکرد حلقه‌ی For را شبیه سازی کند. بدین ترتیب که در هنگام رسیدن به عدد «N» (که در المان کنترل بر روی صفحه‌ی پانل تعیین می گردد) و یا در صورت فشرده شدن دکمه‌ی STOP توسط کاربر، اجرای حلقه‌ی While متوقف گردد.

این برنامه را با عنوان Combo For/While Loop.vi در زیرفهرست LabVIEW\My Activity ذخیره کنید.

فراموش نکنید که حلقه‌ی While تنها در صورتی که ترمینال شرطی آن در وضعیت True قرار گرفته باشد به طور متناوب اجرا می گردد. در این تمرین به دستور And (Functions >> Boolean) نیاز دارید. در ضمن به یاد داشته باشید که در حال اجرای حلقه، LabVIEW المان‌های کنترل یا نشان‌دهنده‌ی قرار گرفته در خارج از چهارچوب حلقه را به روز نمی‌رساند یا قرائت نمی‌کند. بنابراین در صورتی که انتظار دارید تا این برنامه به درستی عمل کند ترمینال مربوط به دکمه‌ی STOP را در داخل حلقه قرار دهید.

## تمرین ۹-۶: پنجره‌ی محاوره‌ای

در این تمرین برنامه‌ای ایجاد کنیدکه مقدار کلید موجود در صفحه‌ی پانل را قرائت کند. سپس یک پنجره‌ی محاوره‌ای باز شود و وضعیت روشن یا خاموش بودن کلید مذکور را نشان دهد.

این برنامه را با عنوان Display Dialog.vi در زیرفهرست LabVIEW\My Activity ذخیره کنید.

# ۷

# آرایه و کلاستر

## در این فصل با موارد زیر آشنا می‌شوید:

- روش ایجاد المان‌های کنترل و نشان‌دهنده‌ی آرایه‌ای
- نحوه‌ی به‌کارگیری روش اندیس‌گذاری خودکار جهت ایجاد یک آرایه
- دستورهای موجود در مورد آرایه‌ها
- مفهوم Polymorphism
- روش استفاده از کلاستر و نحوه‌ی دسته‌بندی و جداسازی آن

# LabVIEW

۷

## آرایه و کلاستر

در این فصل به معرفی دو نوع داده‌ی مرکب و جدید یعنی آرایه و کلاستر می پردازیم. این دو المان قابلیت ذخیره و انجام عملیات بر روی داده های گروهی را به ارمغان می آورند. همچنین در این فصل موارد استفاده از این داده ها و چگونگی به کارگیری آنها را خواهید دید. در ضمن تأثیر توابع و دستورهای موجود در نرم افزار LabVIEW را بر روی این داده های گروهی بررسی خواهیم نمود.

در این فصل اهداف زیر دنبال می شوند:

- ایجاد المان های کنترل و نشان دهنده‌ی آرایه ای و فراگیری روش استفاده از آنها.
- به کارگیری روش اندیس گذاری خودکار جهت ایجاد یک آرایه.
- آشنایی با دستورهای موجود در مورد آرایه ها.
- درک مفهوم Polymorphism.
- فراگیری روش استفاده از کلاستر و نحوه‌ی دسته بندی و جداسازی آن.
- فراگیری تفاوت بین آرایه وکلاستر.

اصطلاحات کلیدی موجود در این فصل عبارتند از:

- آرایه (Array)
- اندیس گذاری خودکار (Auto-Indexing)
- اصل Polymorphism
- کلاستر (Cluster)
- دسته بندی (Bundle)
- جداسازی (Unbundle)

## آرایه

در فصول قبل تنها اعداد اسکالر را مورد بررسی قرار دادیم. عدد اسکالر یک نوع داده‌ی ساده است که شامل یک مقدار می‌باشد و یا در اصطلاح غیرآرایه‌ای می‌باشد. اکنون زمان آن رسیده است تا با انواع داده‌های پیچیده و مرکب آشنا شوید.

آرایه شامل مجموعه‌ای از داده‌هاست که همه از یک نوع هستند. نظیر این آرایه‌ها را در زبانهای برنامه‌نویسی متنی متداول دیده‌اید. یک آرایه ممکن است یک یا چند بعد و در هر بعد حداکثر $2^{31}$ مؤلفه داشته باشد.[1] یک مؤلفه‌ی آرایه نیز ممکن است هر نوع داده به استثناء یک آرایه‌ی دیگر، یک نمودار و یا یک گراف باشد.

دسترسی به مؤلفه‌های آرایه از طریق اندیس‌های آنها امکان‌پذیر است. هر اندیس، عددی بین 0 تا N-1 است که N تعداد کل مؤلفه‌های آرایه می‌باشد. یک آرایه‌ی یک‌بعدی که به اختصار آن را با 1D[2] نشان می‌دهیم در شکل ۱-۷ نشان داده شده است. توجه داشته باشید که اندیس اولین مؤلفه آرایه 0 است، اندیس دومین مؤلفه 1 و ... .

| اندیس | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| آرایهٔ یک‌بعدی با ۱۰ مؤلفه | 12 | 32 | 82 | 8.0 | 4.8 | 5.1 | 6.0 | 1.0 | 2.5 | 1.7 |

**شکل ۱-۷: اندیس‌ها و مؤلفه‌های یک آرایه‌ی یک‌بعدی**

در بخشهای بعدی خواهید دید که برای ذخیره نمودن اغلب شکل موجها می‌توان از آرایه استفاده نمود. در این حالت هر یک از نقاط شکل موج متناظر با یک مؤلفه‌ی آرایه است. بدین مفهوم که به عنوان مثال هر نقطه با مختصات (x,y) می‌تواند مؤلفه‌ای از آرایه‌ی دوبعدی باشد. در ضمن از آرایه‌ها می‌توان برای ذخیره نمودن داده‌ها و اطلاعات به‌دست آمده از حلقه‌ها استفاده نمود. در این حالت در هر تکرار حلقه یکی از مؤلفه‌های آرایه تولید می‌شود.

## روش ایجاد المان‌های کنترل و نشان‌دهنده‌ی آرایه‌ای

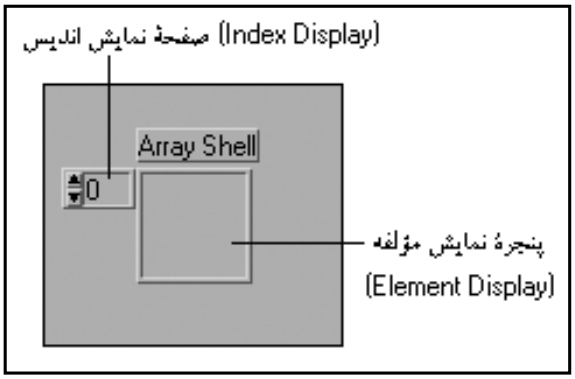
جهت ایجاد المان‌های کنترل و نشان‌دهنده برای داده‌های پیچیده و مرکب نظیر آرایه‌ها و کلاسترها باید دو مرحله را پشت سر گذاشت.

برای ایجاد کنترل یا نشان‌دهنده‌ی آرایه‌ای باید یک نوع داده را در یک پوسته‌ی آرایه قرار داد. این داده ممکن است یکی از انواع عددی، جبری، رشته‌ای، مسیر یا کلاستر باشد. پوسته‌ی آرایه یا Array Shell در زیرپالت Controls >> Array & Cluster قرار دارد.

---

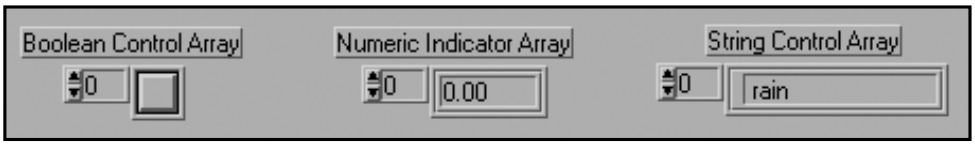
۱- البته این مقدار به گنجایش حافظه نیز بستگی دارد.

2- 1 Dimensional Array



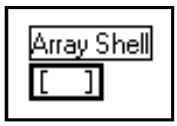

**شکل ۲-۷: شکل ظاهری پوسته‌ی آرایه در صفحه‌ی پانل قبل از قرار دادن مؤلفه‌ها در داخل پنجره‌ی Element Display**

جهت ایجاد یک آرایه کافی است یک نوع داده را در داخل پنجره‌ی Element Display قرار دهید. همان گونه که در شکل ۳-۷ ملاحظه می کنید اندازه‌ی این پنجره برای مطابقت یافتن با مؤلفه‌ی جدید تغییر می یابد. اما تا زمانی که داده‌ها را به آن وارد نکرده‌اید به رنگ خاکستری ظاهر می شود. این حالت بیانگر غیرفعال بودن پوسته‌ی آرایه است. توجه داشته باشید که تمامی مؤلفه های یک آرایه باید همگی از نوع کنترل بوده، یا همگی از نوع نشان دهنده باشند نه ترکیبی از هر دو.




**شکل ۳-۷: شکل ظاهری پوسته‌ی آرایه در صفحه‌ی پانل پس از قرار دادن مؤلفه‌ها در داخل پنجره‌ی Element Display**

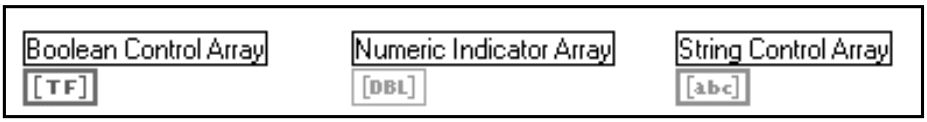
هنگامی که در ابتدای کار پوسته‌ی آرایه را بر روی صفحه‌ی پانل قرار داده و هنوز هیچ مؤلفه‌ای را در داخل پنجره‌ی Element Display قرار نداده‌اید، ترمینال متناظر با آن در صفحه‌ی نمودار بلوکی به رنگ سیاه ظاهر می شود که بیانگر عدم تعریف نوع داده در پوسته‌ی آرایه است. همان گونه که در شکل ۴-۷ نشان داده شده است در داخل ترمینال مذکور، تنها دو براکت قرار دارد. وجود این براکت ها روشی جهت نمایش و تعریف آرایه در محیط LabVIEW است.




**شکل ۴-۷: شکل ظاهری پوسته‌ی آرایه در صفحه‌ی نمودار بلوکی قبل از قرار دادن مؤلفه در داخل پنجره‌ی Element Display**

پس از قرار دادن یک کنترل یا نشان دهنده در پنجره‌ی Element Display، ترمینال متناظر با آن در صفحه‌ی نمودار بلوکی رنگ داده های جدید را به خود می گیرد. مطابق شکل ۵-۷ در داخل براکت نیز حروفی ظاهر می شود که نشان دهنده‌ی نوع المان است. به عنوان مثال ترمینال [TF] نشان می دهد که مؤلفه های این آرایه المان های جبری هستند. توجه داشته باشید که سیم حامل داده های آرایه ای از سیم

حامل داده های عددی و تک مقداری ضخیم تر است.




**شکل ۵-۷: شکل ظاهری پوسته‌ی آرایه در صفحه‌ی نمودار بلوکی پس از قرار دادن مؤلفه در داخل پنجره‌ی Element Display**

پس از انتساب نوع داده به آرایه می توانید داده ها را در آرایه وارد کنید. برای وارد کردن مقادیر، از ابزارهای Labeling Tool یا Operating Tool کمک بگیرید. اگر داده ها از نوع عددی هستند برای کاهش یا افزایش اعداد، بر روی پیکان های موجود در بخش نمایش دهنده‌ی اندیس یا Index Display کلیک کنید.

در صورتی که قصد دارید اندازه‌ی المان را در پنجره‌ی Element Display تغییر دهید از ابزار Positioning Tool استفاده نمایید. در این حالت زمانی که نشانگر ماوس را بر روی یکی از گوشه های پنجره قرار می دهید، نشانگر ماوس به براکت‌های تغییر اندازه یا Resizing Brackets تغییر شکل می دهد. اگر تصمیم دارید مؤلفه های بیشتری را همزمان به نمایش در آورید، ابزار Positioning Tool را در اطراف گوشه های پنجره حرکت دهید تا به نشانگر صفحه‌ی شطرنجی یا Grid Cursor دسترسی پیدا کنید.

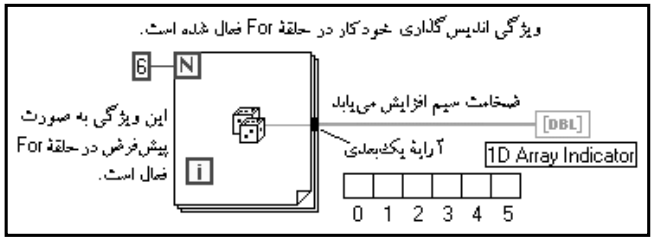
سپس آن را در جهت افقی یا عمودی گسترش دهید در این حالت مؤلفه های بسیاری قابل رؤیت خواهند بود. در این حالت همواره مؤلفه ای که نزدیک به بخش Index Dislay قرار گرفته است متناظر با اندیس نمایش داده شده می باشد.

برای ایجاد ثوابت آرایه ای به همان روش ذکر شده در مورد ثوابت عددی، جبری و رشته ای اقدام کنید. برای انجام این عمل گزینه‌ی Array Constants را از زیرپالت Functions >> Array انتخاب کنید تا پوسته‌ی آرایه ظاهر شود. سپس یک نوع داده‌ی مناسب را که معمولاً عدد ثابت است در داخل پوسته قرار دهید. روش انجام این عمل نظیر روش قرار دادن نوع داده بر روی صفحه‌ی پانل می باشد. این ویژگی در زمانی که قصد انتساب یک مقدار اولیه را به ثبات انتقال داشته باشید و یا بخواهید یک نوع داده را برای یک فایل فراهم سازید بسیار مفید است. این مطلب را در آینده مورد بررسی قرار خواهیم داد. اگر بخواهید یک المان کنترل، نشان دهنده یا نوع داده‌ی ثابت را از آرایه حذف نمایید منوی کرکره ای را بر روی Index Display باز نموده، گزینه‌ی Data Operations >> Empty Array را انتخاب کنید.

## اندیس گذاری خودکار

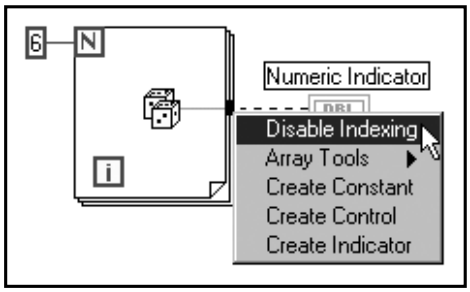
حلقه های For و While می توانند آرایه ها را در دیواره های خود به طور خودکار اندیس گذاری کنند. در این حالت در هر تکرار حلقه یک مؤلفه‌ی جدید ایجاد و اندیس گذاری می شود. این قابلیت را اندیس گذاری

خودکار گویند. مطلب حائز اهمیت آن است که این ویژگی به صورت پیش فرض در For loop فعال و در While loop غیرفعال می باشد. در شکل ۶-۷ یک حلقه‌ی For نشان داده شده است که یک آرایه را در داخل چهارچوب خود به صورت خودکار اندیس گذاری می کند. همان گونه که قبلاً ذکر شد هر تکرار حلقه باعث ایجاد و اندیس گذاری یکی از مؤلفه های آرایه می گردد. تا زمانی که اجرای حلقه به اتمام نرسیده است، دسترسی به مؤلفه های آرایه امکان پذیر نیست. به شکل ۶-۷ دقت کنید. هنگامی که نوع داده ها در خارج از حلقه تغییر می‌کنند و به داده های آرایه ای تبدیل می شوند، ضخامت سیم حامل آنها افزایش می یابد.




**شکل ۶-۷: ایجاد یک آرایه با استفاده از ویژگی اندیس گذاری خودکار در حلقه‌ی For**

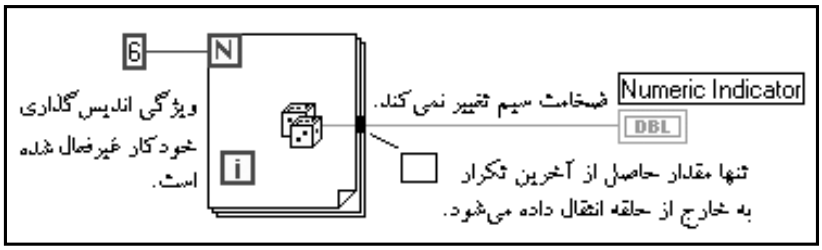
در صورتی که قصد داشته باشید تنها یک مقدار اسکالر را به خارج از حلقه‌ی For ارتباط دهید، منوی کرکره‌ای را بر روی تونل موجود در دیواره‌ی حلقه باز نموده، گزینه‌ی Disable Indexing را انتخاب کنید.




**شکل ۷-۷: نحوه‌ی غیر فعال کردن ویژگی اندیس گذاری خودکار در حلقه‌ی For**

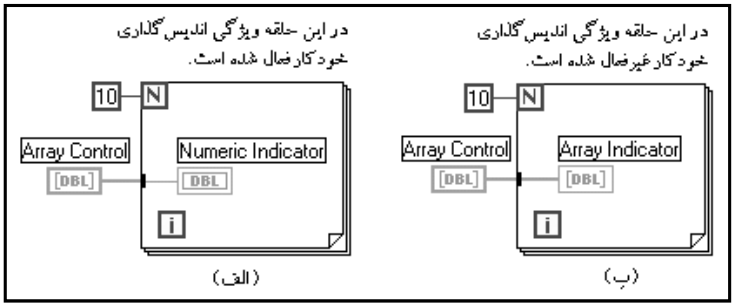
به دلیل اینکه ویژگی اندیس گذاری خودکار در حلقه‌ی While به صورت پیش فرض غیرفعال شده است، در صورت نیاز به استخراج و سیم کشی داده های آرایه ای از این حلقه، منوی کرکره ای را بر روی تونل موجود در دیواره‌ی حلقه باز کنید و گزینه‌ی Enable Indexing را انتخاب نمایید.

در شکل ۸-۷ ویژگی اندیس گذاری خودکار در حلقه‌ی For غیرفعال شده است و تنها آخرین عدد تصادفی حاصل از دستور Random Number (0-1) در حلقه به خارج از آن راه می یابد. به ضخامت سیم در داخل و خارج از حلقه دقت کنید. ملاحظه می کنید که ضخامت سیم در هنگام عبور از دیواره‌ی حلقه تغییر نمی کند.




**شکل ۸-۷: غیر فعال شدن ویژگی اندیس گذاری خودکار در حلقه‌ی For**

از ویژگی اندیس گذاری خودکار جهت سیم کشی آرایه به داخل حلقه نیز استفاده می شود. اگر این ویژگی در حلقه فعال شده باشد، در هر بار تکرار، تنها یکی از مؤلفه های آرایه اندیس گذاری می شود. این مطلب در شکل ۹-۷ (الف) نشان داده شده است. به تغییر ضخامت سیم در هنگام عبور از دیواره‌ی حلقه دقت کنید. در خارج از حلقه، داده ها از نوع آرایه ای هستند و سیم ضخیم تر است. درحالی که پس از عبور از دیواره‌ی حلقه ضخامت سیم کاهش می یابد و بیانگر عبور داده های اسکالر به داخل حلقه است. در شکل ۹-۷ (ب) این ویژگی غیر فعال شده است و کل آرایه یک‌جا در حلقه وارد می شود.

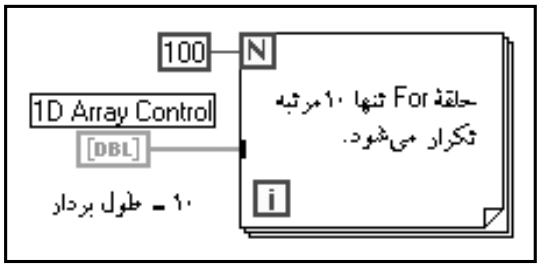



**شکل ۹-۷**

به دلیل اینکه برای پردازش آرایه‌ها اغلب از For loop استفاده می‌شود، زمانی که یک آرایه به/ازحلقه‌ی For سیم‌کشی می گردد، LabVIEW ویژگی اندیس‌گذاری خودکار را به صورت پیش‌فرض فعال می‌کند. همان گونه که قبلاً ذکر شد این ویژگی در حلقه‌ی While به صورت پیش‌فرض غیرفعال است. در صورت نیاز به این خاصیت در حلقه‌ی While، منوی کرکره‌ای را بر روی تونل موجود در دیواره‌ی حلقه باز نموده، سپس گزینه‌ی Enable Indexing را انتخاب کنید.

درهنگام استفاده از ویژگی اندیس گذاری خودکار جهت وارد کردن داده‌ها به داخل یک حلقه‌ی For، ترمینال شمارشی حلقه به طور خودکار به اندازه‌ی مقدار مؤلفه های آرایه تنظیم می گردد. بنابراین در این حالت سیم کشی و انتساب مقدار به ترمینال شمارشی حلقه الزامی نیست. اگر در این حالت به ترمینال مذکور مقداری اختصاص دهید و یا دو آرایه با طول متفاوت به حلقه اعمال کنید، LabVIEW کوچکترین مقدار را به عنوان تعداد تکرارهای حلقه‌ی For انتخاب می کند. در شکل ۱۰-۷، به دلیل این که طول آرایه از مقدار سیم کشی شده به ترمینال شمارشی حلقه‌ی For

کمتر است، تعداد تکراهای حلقه توسط طول آرایه تعیین می شود.

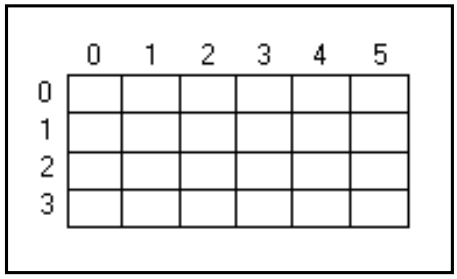



شکل ۱۰-۷

## آرایه‌ی دوبعدی

۷

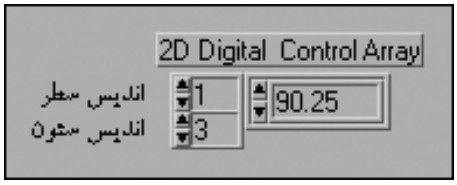
آرایه‌ی دوبعدی که به اختصار آن را به صورت 2D[3] نشان می‌دهیم، همان ماتریس[4] است که مؤلفه‌ها را در یک جدول دوبعدی ذخیره می‌کند. بنابراین برای مشخص کردن محل یک مؤلفه به دو اندیس یعنی اندیس سطر و اندیس ستون نیاز داریم. هر دو دسته اندیس مذکور از عدد صفر آغاز می‌شوند و تنها شامل اعداد صحیح می‌باشند. در شکل ۱۱-۷ یک آرایه‌ی دوبعدی با ۶ ستون و ۴ سطر که در مجموع شامل ۲۴ مؤلفه می‌باشد نشان داده شده است.




شکل ۱۱-۷: یک آرایه‌ی دوبعدی با ۶ ستون و ۴ سطر

برای اضافه کردن ابعاد یک آرایه، منوی کرکره‌ای را بر روی صفحه‌ی نمایش اندیس باز نموده، گزینه‌ی Add Dimension را انتخاب کنید. در صورتی که منوی کرکره‌ای را بر روی صفحه‌ی نمایش مؤلفه باز نمایید، گزینه‌ی مذکور را نخواهید دید. بنابراین در مورد به‌کارگیری منوی کرکره‌ای و گزینه‌های آن و همچنین محل باز نمودن منو دقت کنید.

در شکل ۱۲-۷ یک آرایه‌ی دوبعدی را ملاحظه می‌کنید. توجه داشته باشید که در این حالت برای تعیین موقعیت هر مؤلفه از دو اندیس سطر و ستون استفاده می‌گردد.




شکل ۱۲-۷

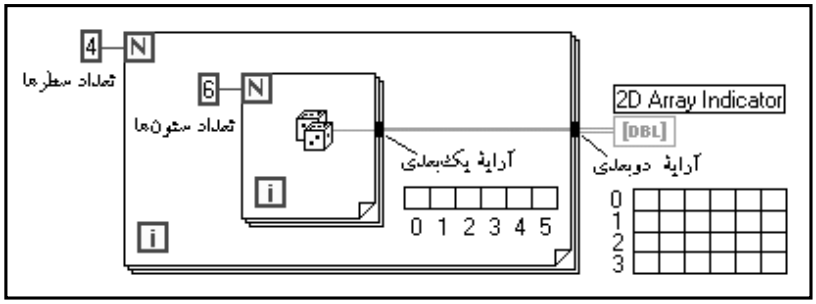
3- 2 Dimensional Array
4- Matrix

Grid Cursor
(Positioning Tool)

با استفاده از ابزار Positioning Tool که در این حالت به نشانگر صفحه‌ی شطرنجی یا Grid Cursur تبدیل می‌شود می‌توانید صفحه‌ی نمایش آرایه را گسترش دهید. پس از تغییر اندازه‌ی صفحه‌ی نمایش آرایه می‌توان مؤلفه‌های بیشتری را مشاهده کرد. جهت حذف ابعاد اضافی و ناخواسته، منوی کرکره‌ای را بر روی بخش نمایش‌دهنده‌ی اندیس باز نموده، گزینه‌ی Remove Dimension را انتخاب کنید.

برای ذخیره‌ی شکل موجهایی از چند کانال در کارت DAQ، می‌توان از یک آرایه‌ی دوبعدی استفاده نمود. در این حالت داده‌های هر یک از کانال‌ها در یکی از ستونهای آرایه‌ی دوبعدی ذخیره می‌شود.

جهت ایجاد یک آرایه‌ی دوبعدی می‌توانید مقادیر مؤلفه‌ها را در صفحه‌ی پانل وارد کنید. روش دیگر برای ایجاد این آرایه استفاده از دو حلقه‌ی For می‌باشد که یکی از آنها در داخل دیگری قرار گرفته است. حلقه‌ی داخلی به ایجاد سطر می‌پردازد و حلقه‌ی بیرونی سطرهای ایجاد شده را جهت پرکردن ستونهای آرایه‌ی دوبعدی مورد استفاده قرار می‌دهد. در شکل ۱۳-۷ دو حلقه‌ی For را ملاحظه می‌کنید که به کمک روش اندیس‌گذاری خودکار یک آرایه‌ی دوبعدی با ۶ ستون و ۴ سطر ایجاد می‌کنند. به عبارت دیگر این دو حلقه یک ماتریس ۶×۴ تولید می‌کنند. در شکل ۱۳-۷ ملاحظه می‌کنید که سیم حامل داده‌های آرایه‌ی دوبعدی ضخیم‌تر از سیم حامل داده‌های آرایه‌ی یک‌بعدی است.



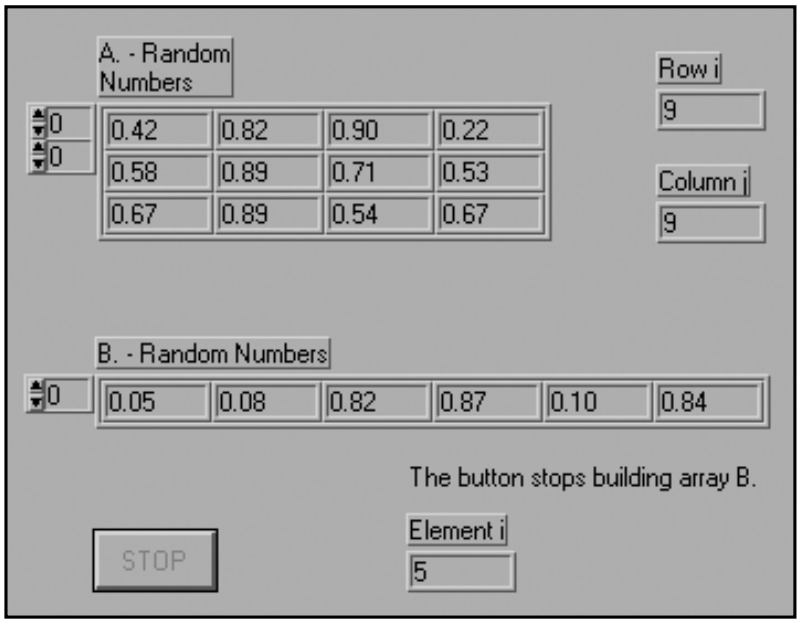

شکل ۱۳-۷

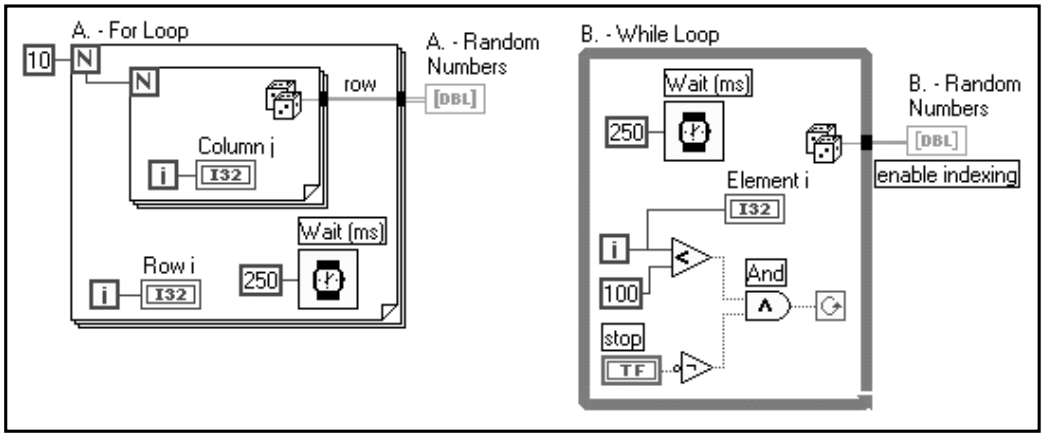
## تمرین ۱-۷: ایجاد آرایه با استفاده از ویژگی اندیس‌گذاری خودکار

در این تمرین قصد داریم تا به کمک روش اندیس‌گذاری خودکار به ایجاد یک آرایه بپردازیم و با روش مذکور آشنا شویم. برای اجرای این تمرین نیاز به برنامه‌ای است که در آن از دو حلقه‌ی For و یک حلقه‌ی While استفاده شده باشد. این برنامه با عنوان Building Array.vi در مسیر LabVIEW\Exercise\Ch7.llb قرار دارد. در این تمرین از دو حلقه‌ی For و یک حلقه‌ی While به ترتیب جهت ایجاد آرایه‌های دوبعدی و یک‌بعدی استفاده شده است. همان گونه که می‌دانید حلقه‌ی For به تعداد دفعات مشخصی اجرا می‌شود. اما برای متوقف کردن حلقه‌ی While باید کلید STOP را که در صفحه‌ی پانل گنجانده شده است فشار دهید؛ در غیر این صورت اجرای حلقه‌ی مذکور در این برنامه پس از ۱۰۱ تکرار متوقف می‌گردد.

۱- برنامه‌ی Building Array.vi را باز کنید.

۲- هر دو صفحه‌ی پانل و نمودار بلوکی را به دقت مورد بررسی قرار دهید. توجه داشته باشید که حلقه‌های داخلی و خارجی در For Loop به ترتیب، سطر و ستون آرایه‌ی دوبعدی را به کمک روش اندیس گذاری خودکار ایجاد می‌کنند. همچنین به تفاوت ضخامت سیمهای حامل داده‌های آرایه‌ی دوبعدی و یک بعدی توجه کنید.




**شکل ۱۴-۷: صفحات پانل و نمودار بلوکی در برنامه‌ی Building Array.vi**

۳- قبل از اینکه بتوان داده‌های آرایه‌ای را از یک حلقه‌ی While استخراج نمود، حتماً باید ویژگی اندیس گذاری خودکار را در آن فعال ساخت. برای این کار منوی کرکره‌ای را بر روی تونل موجود در دیواره‌ی حلقه باز نموده، سپس گزینه‌ی Enable Indexing را انتخاب کنید. برای مشاهده‌ی عملکرد و تأثیر این دستور بر روی حلقه، مجدداً منوی کرکره‌ای را بر روی تونل مذکور باز کنید و گزینه‌ی

Disable Indexing را انتخاب نمایید. ملاحظه خواهید کرد که سیم خارج شده از حلقه به صورت منقطع درآمده و بیانگر وجود اشکال در نحوه‌ی تعریف یکی از پارامترهای حلقه است. بنابراین برای استخراج داده های آرایه ای از حلقه‌ی While، حتماً ویژگی اندیس گذاری خودکار را فعال کنید.

حال مجدداً گزینه‌ی Enable Indexing را از همان منو انتخاب کنید تا سیم خارج شده از حلقه به حالت اولیه‌ی خود باز گردد. همان گونه که درصفحه‌ی نمودار بلوکی ملاحظه می کنید در صورتی که کاربر دکمه‌ی STOP را در حلقه‌ی While فشار ندهد، جهت اطمینان از متوقف شدن برنامه از یک الگوریتم منطقی استفاده شده است. در این صورت اجرای حلقه‌ی While پس از ۱۰۱ تکرار متوقف می گردد. حال این سؤال پیش می آید که چرا به جای ۱۰۰ مرتبه، حلقه به تعداد ۱۰۱ مرتبه تکرار می شود؟

به یاد آورید که در حلقه‌ی While وضعیت ترمینال شرطی در انتهای هر تکرار بررسی می‌شود. در انتهای صدمین تکرار، i=۹۹ می باشد و به دلیل اینکه این مقدار از عدد ۱۰۰ کمتر است، اجرای حلقه ادامه می یابد. در انتهای صدویکمین تکرار i=۱۰۰ است و این مقدار از عدد ۱۰۰ کوچکتر نیست. بنابراین اجرای حلقه متوقف می گردد.

۴- حال که با نحوه‌ی عملکرد و المان های این برنامه آشنا شدید، آن را اجرا کنید. به یاد داشته باشید که برای متوقف کردن حلقه‌ی While، دکمه‌ی STOP را فشار دهید؛ زیرا نشان دهنده های قرار گرفته در صفحه‌ی پانل تا زمانی که آرایه به صورت کامل ایجاد نشده باشد به روز رسانده نمی شوند.

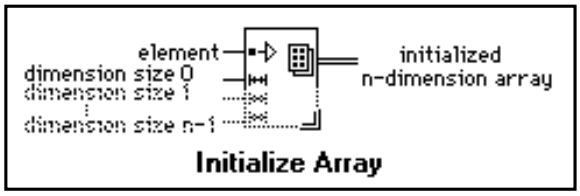
۵- اکنون برنامه را بسته و از ذخیره نمودن تغییرات ایجاد شده خودداری کنید.

## انجام عملیات بر روی آرایه ها

تا کنون مطالبی در مورد آرایه ها آموخته اید. حال قصد داریم دستورهای موجود در مورد آنها را بررسی کنیم. در نرم افزار LabVIEW دستورهای بسیاری برای انجام عملیات بر روی آرایه ها تعبیه شده است. این دستورها در زیرپالت Array << Functions قرار دارند. برای اجتناب از بروز برخی مشکلات عمومی در بین مبتدیان همواره به خاطر داشته باشید که آرایه ها و دیگر ساختارهای موجود در بسته‌ی نرم افزاری LabVIEW همگی Zero-Indexed هستند یا به عبارت دیگر اولین المان یا مؤلفه‌ی آنها دارای اندیس 0 است.

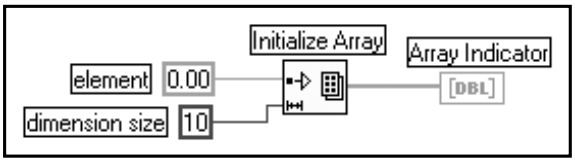
برخی از دستورهای مشترک در بین آرایه ها در این بخش مورد بررسی قرار می گیرند. اما به شما توصیه می کنیم که تمامی دستورهای موجود در زیرپالت Array را مرور کنید و اطلاعاتی در مورد روش به کارگیری و تأثیر اعمال آنها به دست آورید. در بررسی این دستورها حتماً از پنجره‌ی Help استفاده کنید.

دستور Initialize Array یک آرایه‌ی nبعدی را ایجاد می‌کند و مقادیر تعیین شده در ورودی element را اندیس گذاری نموده، آنها را در مؤلفه های آرایه قرار می دهد. برای افزایش تعداد ورودیها، اندازه‌ی آیکن این دستور را با استفاده از ابزار Positioning Tool تغییر دهید. بدین ترتیب بر تعداد ورودیهای dimension size افزوده می شود. دستور مذکور جهت گنجاندن آرایه هایی با طول معین در حافظه بسیار مفید است.



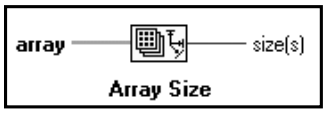

**شکل ۱۵-۷: آیکن دستور Initialize Array به همراه اتصالات آن**

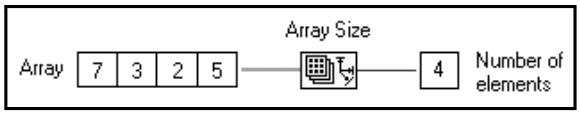
درشکل ۱۶-۷ ملاحظه می کنید که دستور Initialize Array یک آرایه‌ی یک بعدی را ایجاد می کند. این آرایه دارای ۱۰ مؤلفه است که هر مؤلفه‌ی آن حاوی عدد 0 است.




**شکل ۱۶-۷: نمونه‌ای از کاربرد دستور Initialize Array**

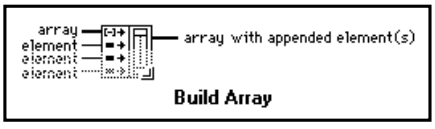
دستور Array Size تعداد مؤلفه ها را در آرایه‌ی ورودی به دست می دهد. در صورتی که آرایه‌ی ورودی n بعدی باشد این دستور n آرایه‌ی یک بعدی به دست می دهد که هر مؤلفه‌ی آن حاوی اندازه‌ی ابعاد یکی از آرایه هاست. در شکل ۱۷-۷ آیکن این دستور را به همراه یک نمونه از کاربرد آن ملاحظه می کنید.



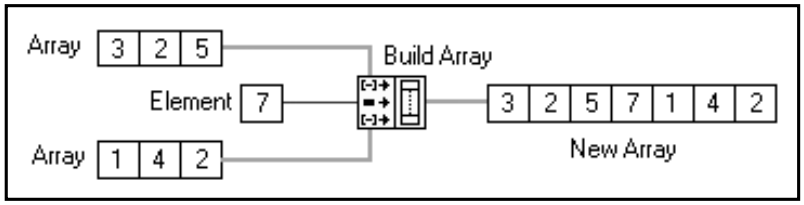

**شکل ۱۷-۷: آیکن دستور Array Size به همراه اتصالات و نمونه‌ای از کاربرد آن**

دستور Build Array دو یا چند آرایه را با یکدیگر ترکیب کرده و حاصل آنها را که به صورت یک زنجیره درآورده است در یک آرایه قرار می‌دهد. همچنین این دستور می تواند مؤلفه های دیگری را به یک آرایه اضافه کند. هنگامی که دستور مذکور را برای اولین بار بر روی صفحه‌ی نمودار بلوکی قرار می دهید به صورت آیکن نشان داده شده در شکل سمت راست ظاهر می شود. برای افزایش تعداد ورودیها می توان اندازه‌ی آن را تغییر داد. دستور Build Array دارای دو نوع ورودی آرایه و مؤلفه است و می تواند از پیوند دادن آرایه و مؤلفه های مجزا، یک آرایه‌ی جدید ایجاد کند.




**شکل ۱۸-۷: آیکن دستور Build Size به همراه اتصالات آن**

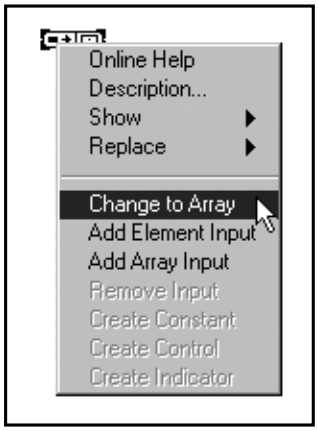
در شکل ۱۹-۷ دستور Build Array، دو آرایه و یک المان مجزا را به ترتیبی که ملاحظه می کنید به صورت یک زنجیره درآورده و یک آرایه‌ی جدید ایجاد کرده است.




**شکل ۱۹-۷: به کارگیری دستور Build Array جهت پیوند دادن دو آرایه و یک مؤلفه‌ی مجزا**

به ورودیهای دستور Build Array دقت کنید. همان گونه که در شکل ۱۹-۷ ملاحظه می کنید ورودیهای آرایه‌ای دارای براکت هستند در حالی که ورودیهای مجزا یا مؤلفه‌ها فاقد براکت می‌باشند. این دو دسته ورودی قابل تعویض نبوده و در صورتی که مراقب نباشید ممکن است با اشکالاتی نظیر bad wire مواجه شوید.

جهت انتخاب نوع ورودی اعم از آرایه‌ای یا مؤلفه‌ی مجزا، منوی کرکره‌ای را بر روی آیکن دستور Build Array باز نموده، یکی از دو گزینه‌ی Change to Array یا Change to Element را انتخاب کنید.

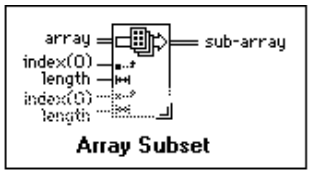


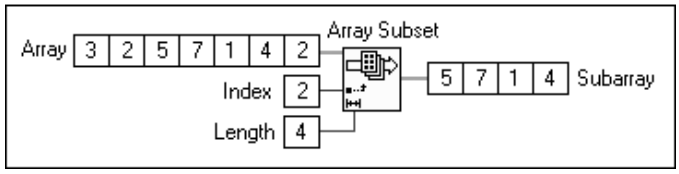

**شکل ۲۰-۷: گزینه‌های موجود در منوی کرکره‌ای دستور Build Array جهت انتخاب نوع ورودی**

به مرور زمان و با به کارگیری دستور Build Array متوجه خواهید شد که به کمک این دستور می توان یک یا چند مؤلفه را به یک آرایه‌ی چندبعدی اضافه نمود. جهت اضافه کردن یک مؤلفه به آرایه‌ی چندبعدی باید بعد مؤلفه‌ی مذکور یک واحد کوچکتر از بعد آرایه‌ی اصلی باشد. به عنوان مثال می توانید یک آرایه‌ی 1D را به یک آرایه‌ی 2D اضافه کنید. واضح است که برای اضافه کردن المان به آرایه‌ی 1D، المان مذکور حتماً باید مؤلفه‌ی تک مقداری باشد.

دستور Array Subset زیرمجموعه‌ای از آرایه را به دست می دهد. بدین ترتیب که اندیس اولین مؤلفه‌ی زیرمجموعه‌ی مورد نظر در ورودی index و طول زیرمجموعه نیز در ورودی lengh مشخص می گردد. در مورد

وارد نمودن اعداد در ورودیهای مذکور به یاد داشته باشید که اندیس اولین مؤلفه همواره 0 است. به عنوان مثال اندیس سومین مؤلفه‌ی آرایه، 2 می باشد.

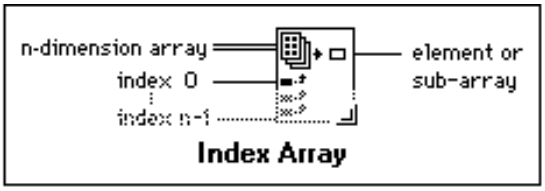


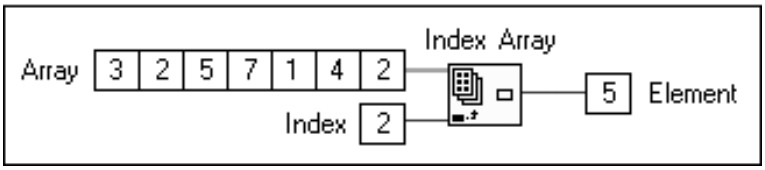




**شکل ۲۱-۷: آیکن دستور Array Subset به همراه اتصالات و یک نمونه از کاربرد آن**

در شکل ۲۱-۷ دستور Array Subset یک آرایه‌ی یک بعدی با 4 مؤلفه را از آرایه‌ی اصلی جدا می کند. اولین مؤلفه‌ی آرایه‌ی حاصل المانی است که با اندیس 2 در آرایه‌ی اصلی وجود دارد.

به کمک دستور Index Array می توان به مؤلفه های معینی از آرایه دست یافت. در این دستور برای دستیابی به مقدار مؤلفه‌ی مورد نظر، اندیس آن را در ورودی index وارد کنید. در شکل ۲۲-۷ نحوه‌ی به کارگیری این دستور برای دستیابی به سومین مؤلفه‌ی یک آرایه نشان داده شده است.



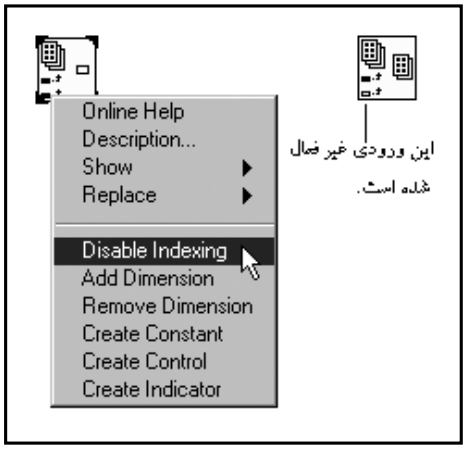

**شکل ۲۲-۷: آیکن دستور Index Array به همراه اتصالات و یک نمونه از کاربرد آن**

همان گونه که در شکل ۲۲-۷ ملاحظه نمودید با استفاده از دستور Index Array یک مؤلفه‌ی اسکالر از یک آرایه جدا می گردد. از این دستور همچنین می توان جهت جدا کردن یک سطر، یک ستون و یا یک مؤلفه‌ی اسکالر از یک آرایه‌ی دوبعدی استفاده نمود. برای استخراج یک مؤلفه از یک آرایه‌ی دوبعدی لازم است تا ابتدا اندازه‌ی دستور مذکور را افزایش دهید، به صورتی که قابلیت پذیرفتن دو ورودی را داشته باشد. برای انجام این عمل منوی کرکره‌ای را بر روی ورودی index باز نموده، سپس گزینه‌ی Add Dimension را انتخاب کنید. ملاحظه می کنید که در این حالت یک ترمینال دیگر جهت اعمال دومین ورودی به آیکن دستور مذکور اضافه می شود. برای بازگشت به حالت اولیه، گزینه‌ی Remove Dimension را از همان منو انتخاب کنید.

برای استخراج یک مؤلفه از آرایه‌ی دوبعدی، اندیس سطر و ستون مؤلفه‌ی مورد نظر را به ترتیب به

ترمینال های ورودی بالایی و پایینی در آیکن دستور Index Array وارد کنید.

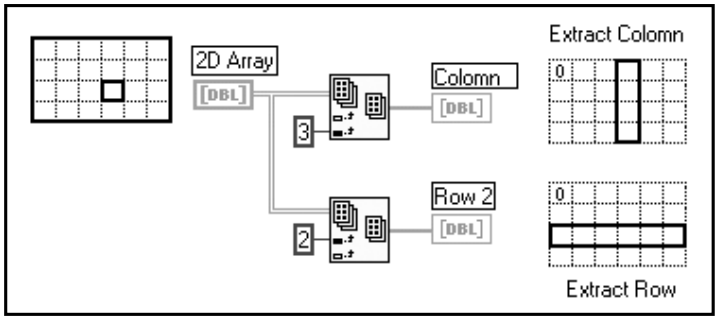
جهت جدا کردن یک سطر یا یک ستون از آرایه‌ی دوبعدی، مطابق شکل ۲۳-۷ منوی کرکره‌ای را بر روی ورودی index باز نموده، سپس گزینه‌ی Disable Indexing را انتخاب کنید.




**شکل ۲۳-۷: روش جدا کردن یک سطر آرایه‌ی دوبعدی**

همان گونه که در شکل ۲۳-۷ نشان داده شده است پس از اجرای دستور Disable Indexing جعبه‌ی سیاه رنگ مربوط به ورودی index به یک جعبه‌ی سفیدرنگ توخالی تغییر شکل می دهد.

با استفاده از این دستور می توانید زیرآرایه هایی از آرایه‌ی اصلی استخراج کنید. در شکل ۲۴-۷ روش جداسازی آرایه های یک بعدی سطری و ستونی را از یک آرایه‌ی دوبعدی ملاحظه می کنید. برای استخراج یک ستون، گزینه‌ی Disable Indexing را از منوی کرکره‌ای بر روی ترمینال بالایی که برای ورودی سطر در نظر گرفته شده است انتخاب کرده، سپس اندیس ستون مورد نظر را به ورودی پایینی سیم کشی نمایید. جهت استخراج یک سطر، گزینه‌ی Disable Indexing را از منوی کرکره‌ای بر روی ترمینال پایینی که جهت ورودی ستون در نظر گرفته شده است انتخاب نموده، سپس اندیس سطر مورد نظر را به ورودی بالایی سیم کشی کنید.




**شکل ۲۴-۷: نحوه‌ی جدا کردن یک سطر یا یک ستون از آرایه‌ی دوبعدی**

همان گونه که عنوان شد برای استخراج یک مؤلفه از یک آرایه‌ی دوبعدی باید اندیس سطر و ستون آن را به ترتیب به ورودیهای دو ترمینال index در آیکن دستور Index Array وارد نمود. توجه داشته باشید که در این حالت هردو ترمینال مذکور باید فعال باشند. برای فعال نمودن این ویژگی گزینه‌ی Enable Indexing را از منوی

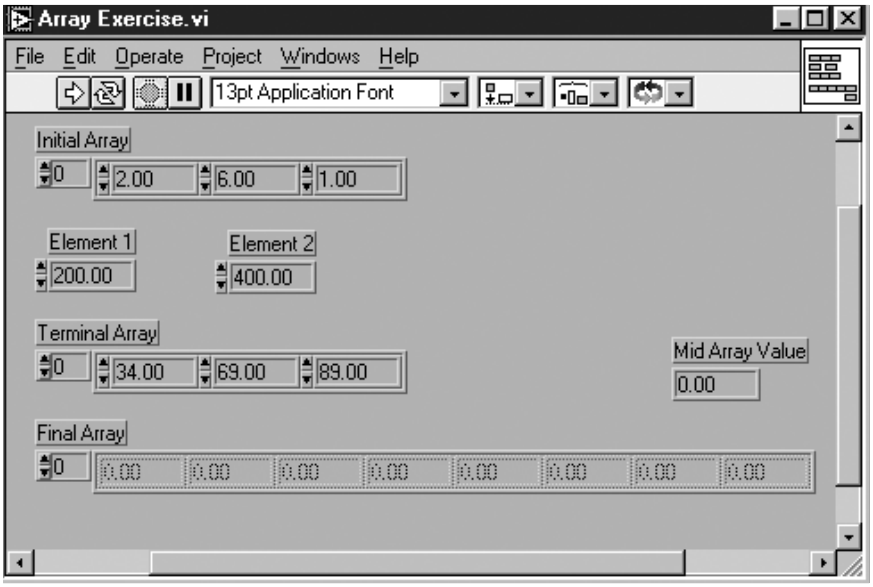
کرکره‌ای ترمینال غیرفعال انتخاب نمایید تا مستطیل سفیدرنگ به رنگ سیاه در آید.

## تمرین ۲-۷: پیوند دادن آرایه‌ها و مؤلفه‌های مجزا

در این تمرین قصد داریم تا با پیوند دادن دو آرایه و دو مؤلفه‌ی مجزا یک آرایه‌ی جدید به‌دست آورده، سپس مؤلفه‌ی میانی آرایه‌ی حاصل را استخراج کنیم.

۱- برنامه‌ی Array Exercise.vi را از مسیر LabVIEW\Exercise\Ch7.llb باز کنید. در صفحه‌ی پانل این برنامه دو آرایه‌ی ورودی (که هر یک شامل ۳ مؤلفه هستند) و دو المان Digital Control را که در این برنامه نقش مؤلفه‌های مجزا را بر عهده دارند به همراه آرایه‌ی حاصل ملاحظه می‌کنید. این برنامه دو آرایه‌ی ورودی و مؤلفه‌های مجزا را به صورت زیر به یکدیگر پیوند داده، یک آرایه‌ی جدید با ۸ مؤلفه به‌دست می‌دهد.

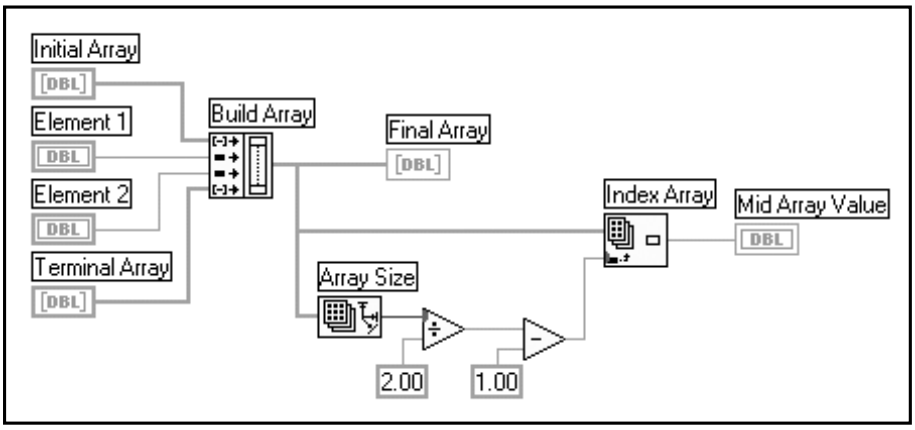
Initial Array + Element 1 + Element 2 + Terminal Array




**شکل ۲۵-۷: صفحه‌ی پانل در برنامه‌ی Array Exercise.vi**

همان گونه که ملاحظه می‌کنید صفحه‌ی پانل به طور کامل ایجاد شده است. وظیفه‌ی شما در این تمرین، تکمیل کردن صفحه‌ی نمودار بلوکی است. توجه داشته باشید تا زمانی که به المان‌های کنترل و نشان‌دهنده‌ی مربوط به آرایه داده‌ای نسبت داده نشود، به رنگ خاکستری ظاهر می‌شوند. این حالت بیانگر غیرفعال بودن المان‌های مذکور است.

۲- صفحه‌ی نمودار بلوکی نشان داده شده در شکل ۲۶-۷ را ایجاد کنید. در هنگام سیم‌کشی برای یافتن ترمینال‌های صحیح از پنجره‌ی Help استفاده کنید.




**شکل ۲۶-۷: صفحه‌ی نمودار بلوکی در برنامه‌ی Array Exercise.vi**

۷

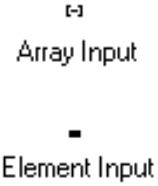
در ادامه به بررسی برخی از دستورهای استفاده شده در صفحه‌ی نمودار بلوکی این برنامه می پردازیم:

در این تمرین دستور Build Array (Functions >> Array) داده های ورودی را به ترتیب زیر به یکدیگر پیوند می‌دهد و یک آرایه‌ی جدید ایجاد می کند:

Initial Array + Element 1 + Element 2 + Terminal Array

هنگامی که آیکن دستور Build Array را برای اولین بار بر روی صفحه‌ی نمودار بلوکی قرار می‌دهید به صورت شکل نشان داده شده در سمت راست ظاهر می‌شود. ابزار Positioning Tool را بر روی گوشه‌ی پایینی سمت راست آیکن قرار داده، اندازه‌ی آن را به گونه‌ای تغییر دهید که ظرفیت دریافت ۴ ورودی را داشته باشد.

برای تغییر اندازه به روش دیگر، منوی کرکره‌ای را بر روی ترمینال ورودی آیکن باز کنید و یکی از دو گزینه‌ی Add Element Input یا Add Array Input را انتخاب نمایید. ملاحظه می کنید که یک ورودی دیگر به ورودی های آیکن این دستور اضافه می‌شود. برای افزایش ورودیها به تعداد ۴ ورودی، باید این عمل را ۳ مرتبه انجام دهید. این مطلب را تنها به عنوان تمرین بیشتر انجام دهید. برای تغیر اندازه، بهتر است از روش اول یعنی از ابزار Positioning Tool استفاده کنید.




اولین ورودی دستور Build Array را از حالت ورودی المانی یا Element Input به حالت ورودی آرایه‌ای یا Array Input تغییر دهید. برای انجام این عمل منوی کرکره‌ای را بر روی ورودی مذکور باز نموده، سپس گزینه‌ی Change to Array را انتخاب کنید. در مورد آخرین ورودی نیز همین عمل را تکرار کنید. اشکال سمت راست نحوه‌ی نمایش ورودیهای المانی و آرایه‌ای را نشان می دهند.

دستور Array Size (Functions >> Array) تعداد مؤلفه های آرایه‌ی جدید را به‌دست می دهد.

از دستور Index Array (Functions >> Array) جهت به‌دست آوردن مؤلفه‌ی میانی آرایه‌ی جدید استفاده می‌شود.

در این برنامه ابتدا با استفاده از دستور Build Array یک آرایه‌ی جدید ایجاد شده، سپس اندیس مؤلفه‌ی میانی آن محاسبه می‌گردد. برای به‌دست آوردن اندیس مؤلفه‌ی میانی طول آرایه‌ی جدید بر ۲ تقسیم شده، یک واحد از آن کاسته می‌شود. تفریق یک واحد به دلیل آغاز شدن اندیس‌ها از عدد 0 است. به دلیل اینکه تعداد مؤلفه‌های آرایه‌ی جدید زوج است مؤلفه‌ی میانی یکی از دو مؤلفه‌ی وسطی می‌باشد.

۳- به صفحه‌ی پانل بازگردید و برنامه را اجرا کنید. سپس این برنامه را در مورد چند ترکیب متفاوت از آرایه‌ها و المان‌ها به کار برید.

۴- این برنامه را در زیرفهرست LabVIEW\My Activity ذخیره کنید.

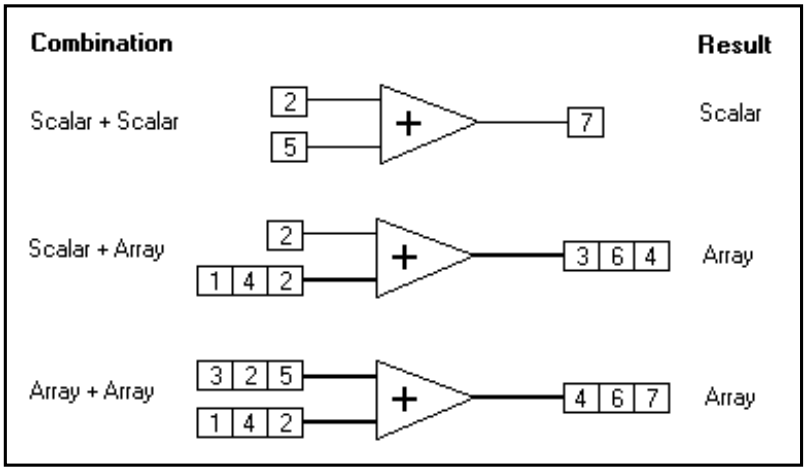
مثالها و برنامه‌های دیگری در مسیر LabVIEW\Examples\General\arrays.llb قرار دارند. به شما توصیه می‌کنیم که حتماً این برنامه‌ها را بررسی کنید.

## مفهوم Polymorphism

ویژگی کاربردی دیگری که در دستورها و توابع محاسباتی نظیر Add، Multiply و Divide در LabVIEW دیده می‌شود، قابلیت Polymorphism می‌باشد. این واژه بیان کننده‌ی اصل زیر است:

«ورودیهای اعمال شده به توابع و دستورهای مذکور می‌توانند در اندازه‌ها و انواع نمایش مختلف ظاهر شوند.»

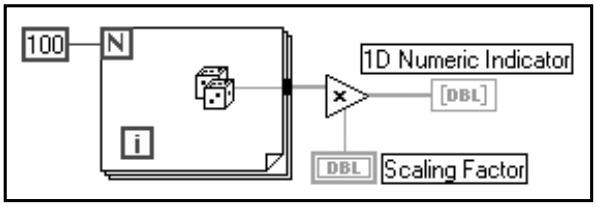
به عنوان مثال می‌توان یک عدد اسکالر را با یک آرایه جمع نمود یا به کمک این اصل می‌توان دو آرایه را با یکدیگر جمع کرد. در شکل ۲۷-۷ برخی از ترکیبات مختلف مربوط به دستور Add را ملاحظه می‌کنید.




**شکل ۲۷-۷: استفاده از اصل Polymorphism در مورد دستور Add**

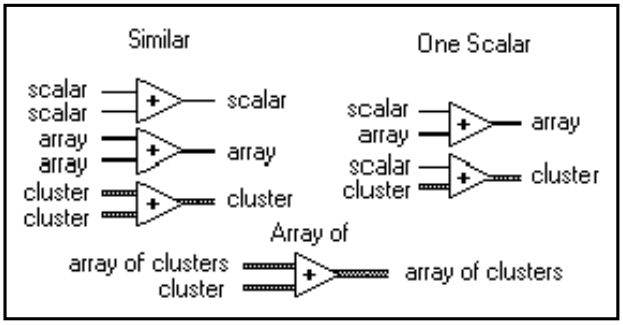
همان گونه که در شکل ۷-۲۷ مشاهده می کنید در اولین ترکیب، دو عدد اسکالر با یکدیگر جمع شده و عدد اسکالر دیگری به دست آمده است. در دومین ترکیب یک عدد اسکالر به هر یک از مؤلفه های آرایه اضافه شده و حاصل نیز یک آرایه است. در ترکیب سوم مؤلفه های متناظر دو آرایه با یکدیگر جمع شده و یک آرایه‌ی جدید به دست آمده است. در تمامی موارد مذکور از دستور Add استفاده شده است. اما ملاحظه می کنید که در موارد مختلف نتیجه‌ی ترکیب متفاوت است.

در شکل ۷-۲۸ در هر تکرار حلقه‌ی For یک عدد تصادفی در فاصله‌ی [0,1] ایجاد شده و در آرایه‌ای که در چهارچوب حلقه ساخته شده است ذخیره می گردد. پس از اتمام اجرای حلقه هر یک از مؤلفه های آرایه‌ی مذکور توسط دستور Multiply در عددی که توسط کاربر تعیین می‌شود ضرب می گردد. این عدد به Scaling Factor معروف است. سپس المان نشان‌دهنده‌ی آرایه‌ای یا Array Indicator در صفحه‌ی پانل، آرایه‌ی حاصل را به نمایش در می آورد.



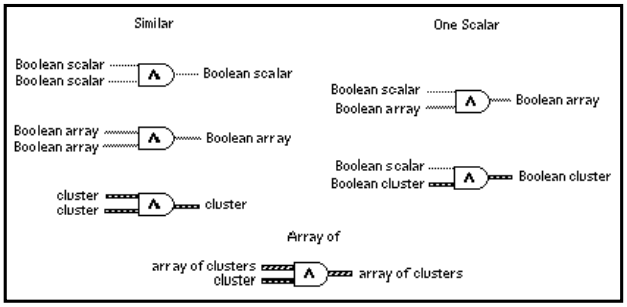

**شکل ۷-۲۸**

در شکل ۷-۲۹ ترکیبات مختلف دستور Add را ملاحظه می کنید.




**شکل ۷-۲۹: ترکیبات مختلف دستور Add**

این قابلیت در مورد دستورهای جبری نیز وجود دارد. در شکل ۷-۳۰ ترکیبات مختلف دستور And نشان داده شده است.



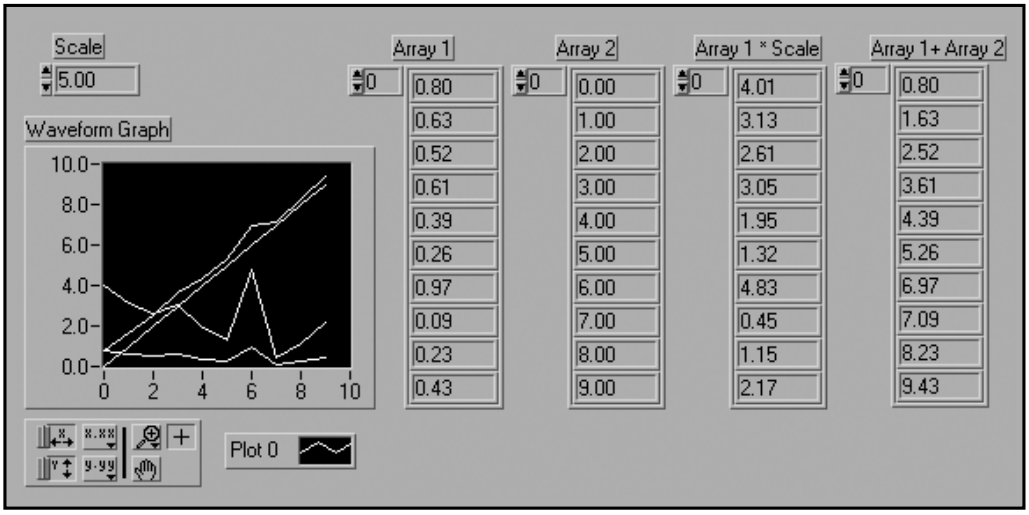

**شکل ۷-۳۰: ترکیبات مختلف دستور And**

اگر دستورهای محاسباتی را در مورد دو آرایه با تعداد مؤلفه‌های متفاوت به‌کار برید، تعداد مؤلفه‌های آرایه‌ی حاصل برابر با تعداد مؤلفه‌های آرایه‌ی کوچکتر است. در این حالت عملیات بر روی مؤلفه‌های متناظر دو آرایه انجام می‌گیرد و از مؤلفه‌های باقی مانده در آرایه‌ی طولانی‌تر چشم‌پوشی می‌شود.

## تمرین ۳-۷: بررسی مفهوم Polymorphism

در این تمرین به ایجاد یک برنامه می‌پردازیم و به کمک آن، مفهوم Polymorphism را در مورد آرایه‌ها بررسی می‌کنیم.

۱- یک برنامه‌ی جدید باز کنید و صفحه‌ی پانل آن را مطابق شکل ۳۱-۷ ایجاد نمایید.




**شکل ۳۱-۷: صفحه‌ی پانل در برنامه‌ی Polymorphism Example.vi**

Grid Cursor (Positioning Tool)

ابتدا آرایه ها را ایجاد کنید. برای انجام این عمل، ابتدا پوسته‌ی آرایه یا Array Shell را از زیرپالت Controls >> Array & Cluster انتخاب نموده، سپس یک نشان دهنده‌ی عددی یا Numeric Indicator را در پنجره‌ی آن قرار دهید. برای مشاهده‌ی مؤلفه های دیگر آرایه به کمک ابزار Positioning Tool بر روی گوشه‌ی پنجره‌ی نمایش کلیک کنید و ماوس را حرکت دهید. در این حالت ابزار مذکور به نشانگر صفحه‌ی شطرنجی یا Grid Cursor تبدیل می‌شود. با این روش می‌توانید تعدادی از مؤلفه های آرایه‌ی یک‌بعدی را به صورت افقی یا عمودی نمایش دهید.

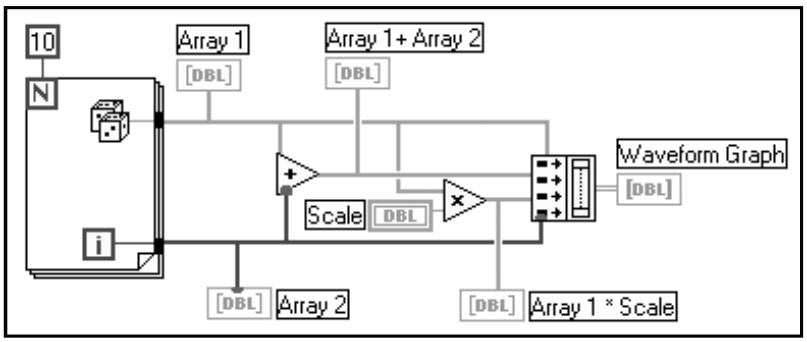
برای هر یک از چهار آرایه‌ی موجود در صفحه‌ی پانل، یک المان نشان‌دهنده در نظر گرفته شده است. برای اجتناب از سردرگمی به هر یک از آنها یک برچسب منحصر به فرد اختصاص دهید. اگر واقعاً تشخیص نمی‌دهید که المان موجود در صفحه‌ی پانل متناظر با کدام یک از المان‌های صفحه‌ی نمودار بلوکی است، منوی کرکره‌ای را بر روی آن باز نموده، گزینه‌ی Find Terminal را انتخاب کنید. ملاحظه می‌کنید که المان

متناظر با آن در صفحه‌ی نمودار بلوکی از سایر المان ها متمایز می گردد.

۲- پس از ایجاد آرایه ها، گزینه‌ی Waveform Graph را از زیرپالت Graph << Controls انتخاب نمایید. اگرچه گراف ها در فصل بعدی مورد بحث و بررسی قرار خواهند گرفت، اما در این تمرین قصد داریم تا پیش زمینه ای برای شما ایجاد کنیم.

۳- یک المان کنترل عددی (Numeric Control) بر روی صفحه‌ی پانل قرار دهید و آن را با برچسب Scale نام‌گذاری کنید.

۴- صفحه‌ی نمودار بلوکی را مطابق شکل ۳۲-۷ ایجاد نمایید. در انجام سیم کشی و برقراری ارتباط بین ترمینال ها از پنجره‌ی Help استفاده کنید. به یاد داشته باشید که خاصیت اندیس گذاری خودکار به صورت پیش فرض در حلقه‌ی For فعال است. بنابراین آرایه ها به صورت خودکار ایجاد می شوند.




**شکل ۳۲-۷: صفحه‌ی نمودار بلوکی در برنامه‌ی Polymorphism Example.vi**

۵- دستورهای Add، Multiply و Random Number (0-1) را از زیرپالت Numeric << Functions انتخاب نمایید.

۶- دستور Build Array را از زیرپالت Array << Functions انتخاب کنید. با استفاده از ابزار PositioningTool اندازه‌ی آن را به گونه ای تغییر دهید که ظرفیت پذیرفتن ۴ ورودی را داشته باشد. تمامی ورودیها را به صورت المانی انتخاب کنید. آرایه‌ی حاصل از دستور Build Array، یک آرایه‌ی دوبعدی خواهد بود که هر آرایه‌ی ورودی در یکی از سطرهای آن قرار گرفته است. بنابراین آرایه‌ی دوبعدی حاصل در خروجی، از ۴ سطر که در هر سطر آن ۱۰ مؤلفه وجود دارد تشکیل می شود. به عبارت دیگر آرایه‌ی حاصل یک ماتریس ۱۰×۴ خواهد بود.

۷- حال برنامه را اجرا کنید. ملاحظه می کنید که هر یک از مؤلفه های ۴ آرایه‌ی مذکور نسبت به اندیس های خود بر روی یک گراف ترسیم می شوند. ۴ آرایه‌ی ترسیم شده عبارتند از:

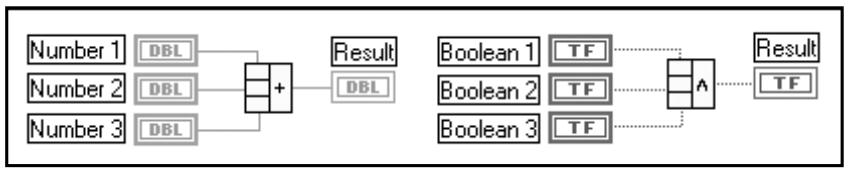
Array 1 , Array 2 , Array 1 * Scale , Array 1 + Array 2

آرایه‌ی خروجی چندین کاربرد Polymorphism را در نرم افزار LabVIEW نشان می دهد.

۸- این برنامه را با عنوان Polymorphism Example.vi در زیرفهرست LabVIEW\ My Activity ذخیره نموده، سپس آن را ببندید.

## دستور Compound Arithmetic

برای انجام محاسبات مرکب و پیچیده، از دستور Compound Arithmetic استفاده می‌شود. همان گونه که در مثال قبل روش انجام عملیات جمع بر روی آرایه هایی با طول های متفاوت را ملاحظه نمودید، این دستور به شما اجازه می دهد تا به طور همزمان عملیات و محاسبات را بر روی بیش از دو مقدار ورودی انجام دهید. دستور مذکور نیازهای شما را در مورد انجام اعمال Add، Multiply، AND و OR بر روی چند مقدار عددی یا جبری برآورده می سازد. در شکل ۳۳-۷ دو نمونه از کاربردهای این دستور را ملاحظه می کنید.




**شکل ۳۳-۷ : دو نمونه از کاربردهای دستور Compound Arithmetic**

این دستور در هر دو زیرپالت Numeric و Boolean از پالت Functions وجود دارد. همان طور که در مورد دستورهای دیگر اشاره شد، برای تغییر اندازه‌ی آیکن این دستور و افزایش تعداد ترمینال های ورودی از ابزار Positioning Tool استفاده نمایید.

دستور Compound Arithmetic تنها به یک حالت ظاهر می شود. برای انجام اعمال محاسباتی گوناگون بنا به نیاز خود، مد آن را تغییر دهید. برای تغییر مد دستور مذکور و انتخاب هر یک از حالات، منوی کرکره‌ای را بر روی ترمینال خروجی باز نموده، گزینه‌ی Change Mode را انتخاب کنید. روش دیگر برای تعویض مد عملیاتی، کلیک کردن بر روی دستور مذکور به کمک ابزار Operating Tool می باشد. برای معکوس کردن علامت ورودیها و خروجیهای عددی یا تغییر وضعیت مقادیر جبری از False به True یا برعکس، منوی کرکره‌ای را بر روی ورودی یا خروجی مورد نظر باز نموده، گزینه‌ی Invert را انتخاب کنید. معکوس شدن علامت یا مقدار، با وجود یک دایره‌ی کوچک در ورودی و خروجی مشخص می گردد.

**شکل ۳۴-۷ : استفاده ار گزینه‌ی Invert جهت معکوس نمودن علامت یا مقدار در دستور Compound Arithmetic**

## نکاتی در مورد عملیات جبری

در نرم‌افزار LabVIEW دستورهایی نظیر And، OR، NOT، Exclusive OR، Not Exclusive OR، Not And و Not OR جهت انجام عملیات جبری در نظر گرفته شده‌اند. در این کتاب به ندرت از عملیات جبری استفاده شده است. اگر تاکنون در مورد این اعمال هیچ مرجعی را مطالعه نکرده‌اید به شما توصیه می‌کنیم حتماً یکی از کتاب‌های مفید و کاربردی در زمینه‌ی طراحی مدارات منطقی را مطالعه کنید. اما در این بخش قصد داریم تنها به عنوان مقدمه، برخی از اصول عملیات جبری را خاطر نشان کنیم. برای کسب اطلاع در مورد عملکرد این دستورها از پنجره‌ی Help کمک بگیرید.

دستور NOT ساده‌ترین دستور منطقی است که وضعیت ورودی را معکوس می‌کند. در صورتی که ورودی این دستور در وضعیت True قرار داشته باشد، خروجی آن در وضعیت False خواهد بود و برعکس.

خروجی دستور And تنها در صورتی در وضعیت True خواهد بود که ارزش تمام ورودیهای آن نیز True باشد.

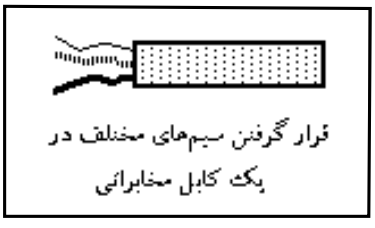
خروجی دستور OR در صورتی True خواهد بود که حداقل یکی از ورودیهای آن در وضعیت True باشد.

۷

## کلاستر

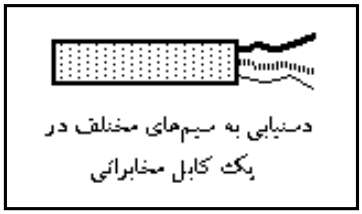
اکنون که مفهوم آرایه را تا اندازه‌ای دریافته‌اید، فراگیری مفهوم کلاستر بسیار ساده خواهد بود. نظیر آرایه، کلاستر نیز ساختاری است که در آن، داده‌ها به صورت گروهی قرار گرفته‌اند. اما برخلاف آرایه، در یک کلاستر می‌توان انواع مختلفی از داده‌ها را دسته‌بندی نمود. به عنوان مثال می‌توان مقادیر جبری و عددی را در یک کلاستر قرار داد.

این ساختار در نرم‌افزار LabVIEW متناظر با Record و Struct در زبانهای برنامه‌نویسی Pascal و C است. برای درک بهتر مفهوم کلاستر آن را به صورت دسته‌ای سیم در یک کابل مخابراتی در نظر بگیرید. هر یک از سیمها در این کابل نمایانگر یک المان مجزا در کلاستر است. به دلیل اینکه در صفحه‌ی نمودار بلوکی برای انتقال داده‌های کلاستری از یک رشته سیم نشان داده می‌شود (حتی اگر حامل چندین مقدار از انواع متخلف داده‌ها باشد) از ازدحام سیمهای موجود در صفحه کاسته می‌شود و تعداد ترمینال‌های کانکتور مورد نیاز در زیربرنامه‌ها نیز کاهش می‌یابد. در مورد رسم داده‌ها بر روی گراف یا نمودار مکرراً از کلاستر استفاده می‌شود. در شکل ۳۵-۷ مفهوم دسته‌بندی داده‌ها در کلاستر به قرار گرفتن سیم های مختلف در یک کابل مخابراتی تشبیه شده است.




**شکل ۳۵-۷: مفهوم دسته‌بندی داده‌ها در کلاستر**

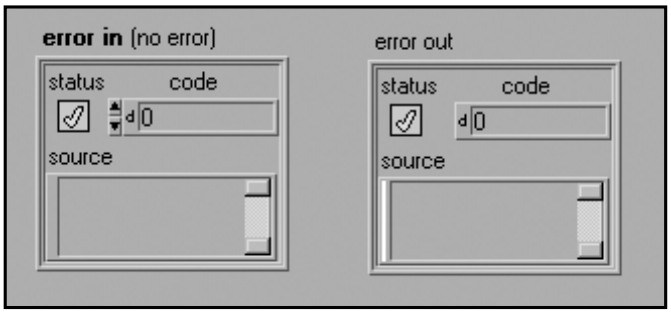
برای دستیابی به المان های دسته بندی شده در کلاستر باید آنها را از هم جدا نمود و هر یک ازآنها را اندیس گذاری کرد. جدا کردن المان های موجود در یک کلاستر را نظیر باز کردن کابل مخابراتی در نظر بگیرید. واضح است که پس از باز کردن کابل تلفن می توان به هر یک از سیمهای رنگی موجود در آن دست یافت. به طریق مشابه پس از جدا کردن المان های موجود در کلاستر دستیابی به هر یک از المان ها امکان پذیر است. در شکل ۳۶-۷ مفهوم جداسازی المان ها در کلاستر به باز کردن یک کابل مخابراتی و دستیابی به سیمهای موجود در آن تشبیه شده است.




**شکل ۳۶-۷: مفهوم جداسازی داده‌ها در کلاستر**

بر خلاف آرایه که می تواند به صورت دینامیکی تغییر اندازه یابد، کلاستر دارای اندازه‌ی ثابتی می‌باشد یا به عبارت دیگر تعداد سیمهای موجود در آن ثابت است. برای اتصال دو کلاستر به یکدیگر باید تعداد المان های موجود در دو کلاستر یکسان باشند همچنین المان های متناظر از نظر نوع داده و ترتیب قرار گرفتن در کلاستر با یکدیگر مطابقت داشته باشند. تا زمانی که نوع داده ها در دو کلاستر با یکدیگر هماهنگی داشته باشند، اصل Polymorphism در مورد کلاستر ها نیز همانند آرایه ها صادق است.

کلاستر ها اغلب در مورد نگه داری خطاها مورد استفاده قرار می گیرند. در شکل ۳۷-۷ کلاستر های خطا را ملاحظه می کنید. این دو کلاستر عبارتند از: کلاستر خطای ورودی (Error In.ctl) و کلاستر خطای خروجی (Error Out.ctl) که برای ثبت خطاهای ایجاد شده بین برنامه های موجود در صفحه‌ی نمودار بلوکی به کار برده می شوند.



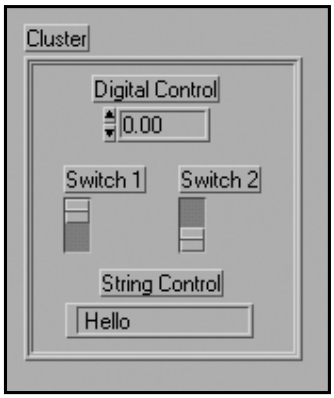

**شکل ۳۷-۷: کلاستر خطای ورودی و خروجی**

به عنوان مثال بسیاری از برنامه های مربوط به DAQ و File I/O حاوی کلاستر های خطا هستند. موارد استفاده از کلاستر های خطا به قدری زیاد است که برای سهولت دستیابی به آنها، در زیرپالت Controls >> Array & Cluster گنجانده شده اند.

## روش ایجاد المان‌های کنترل و نشان‌دهنده در کلاستر

برای ایجاد یک کلاستر ابتدا پوسته‌ی آن یا Cluster Shell را از زیرپالت Controls >> Array & Cluster انتخاب کنید و آن را بر روی صفحه‌ی پانل قرار دهید. سپس هر یک از المان‌های صفحه‌ی پانل را در داخل پوسته‌ی کلاستر قرار دهید. همان گونه که در مورد آرایه اشاره شد برای قرار دادن المان‌ها در داخل کلاستر می‌توانید مستقیماً المان‌ها را از پالت Controls یا زیرپالت‌های آن انتخاب کنید و یا المان‌های موجود در صفحه‌ی پانل را به داخل کلاستر انتقال دهید.

المان‌های داخل یک کلاستر باید همگی از نوع کنترل بوده، یا همگی از نوع نشان‌دهنده باشند. قرار دادن هر دو نوع المان کنترل و نشان‌دهنده در یک کلاستر امکان‌پذیر نیست. زیرا کلاستر باید از نوع ورودی یا از نوع خروجی باشد. اولین المانی که در کلاستر قرار می‌گیرد نوع آن را از لحاظ ورودی یا خروجی تعیین می‌کند. در صورت نیاز می‌توانید اندازه‌ی پنجره‌ی کلاستر را به کمک ابزار Positioning Tool تغییر دهید. در شکل ۳۸-۷ یک کلاستر ورودی را ملاحظه می‌کنید که در داخل آن ۴ المان کنترل قرار گرفته است.



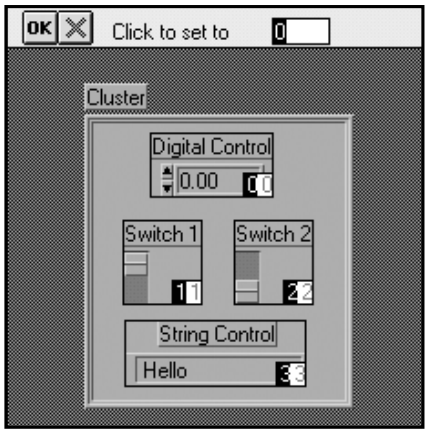

**شکل ۳۸-۷: یک کلاستر ورودی که حاوی ۴ المان کنترل است.**

برای گنجاندن مقادیر ثابت در یک کلاستر، آیکن Cluster Constant (Functions >> Cluster) را در صفحه‌ی نمودار بلوکی قرار داده، سپس مقادیر ثابت را در آن قرار دهید. در صورتی که بخواهید اندازه‌ی پنجره‌ی کلاستر با اندازه‌ی المان‌های قرارگرفته در آن مطابقت داشته باشد، منوی کرکره‌ای را بر روی چهارچوب کلاستر باز کنید و گزینه‌ی Autosizing را انتخاب نمایید.

## ترتیب قرارگرفتن المان‌ها در کلاستر

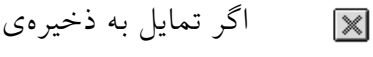
المان‌های کلاستر با توجه به یک ترتیب منطقی و مستقل از محل آنها، در داخل پوسته‌ی کلاستر قرار دارند. اولین المان قرار گرفته در داخل پوسته، المان شماره‌ی 0 است و دومین المان داخل پوسته المان شماره‌ی 1 و به همین ترتیب الی آخر. در صورتی که یکی از المان‌ها را حذف کنید ترتیب آنها به طور خودکار تنظیم می‌گردد. اگر قصد دارید دو کلاستر را به یکدیگر متصل کنید باید ترتیب قرار گرفتن المان‌ها در دو

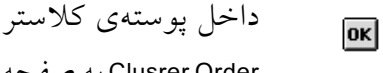
کلاستر را حتماً در نظر بگیرید. این دو کلاستر باید از نظر نوع داده ها و ترتیب قرار گرفتن آنها در داخل کلاستر یکسان باشند. در صورتی که بخواهید المان های کلاستر را جدا کنید، باید بدانید که کدام مقدار متناظر با کدام خروجی در کلاستر است. در بخشهای بعد، این مطلب را بررسی خواهیم کرد. برای ایجاد تغییر در ترتیب قرار گرفتن المان های موجود در پوسته‌ی کلاستر، منوی کرکره‌ای را بر روی چهارچوب کلاستر باز نموده، گزینه‌ی ...Cluster Order را انتخاب کنید. در این حالت ملاحظه می کنید که تعدادی دکمه‌ی جدید در نوار ابزار ظاهر می‌شوند و شکل ظاهری کلاستر نیز مطابق شکل ۳۹-۷ تغییر می کند.




**شکل ۳۹-۷: تغییر شکل ظاهری کلاستر در صورت انتخاب گزینه‌ی ...Cluster Order**

اعداد مندرج در مستطیل های سفیدرنگ در شکل ۳۹-۷ ترتیب فعلی المان ها را در داخل کلاستر نشان می دهند. اعداد موجود در مستطیل های سیاه رنگ نیز ترتیب قرار گرفتن جدید المان ها را مشخص می کنند. در صورتی که با استفاده از نشانگر Cluster Order بر روی یکی از المان ها کلیک کنید، ترتیب قرار گرفتن آن المان در داخل پوسته‌ی کلاستر در نوار ابزار نشان داده می شود. قبل از کلیک کردن بر روی المان مورد نظر شماره ای را در نوار ابزار تایپ کنید و بدین صورت ترتیب قرار گرفتن المان را در داخل پوسته‌ی کلاستر تغییر دهید.


Revert Button


OK Button

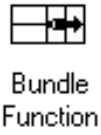
اگر تمایل به ذخیره‌ی تغییرات انجام شده ندارید با کلیک کردن بر روی دکمه‌ی Revert به ترتیب قبلی بازگردید. پس از تنظیم ترتیب قرار گرفتن المان ها در داخل پوسته‌ی کلاستر بر روی دکمه‌ی OK کلیک کنید و با خارج شدن از محیط Clusrer Order به صفحه‌ی پانل بازگردید.

## به کارگیری کلاستر جهت ارسال یا دریافت داده ها در زیربرنامه

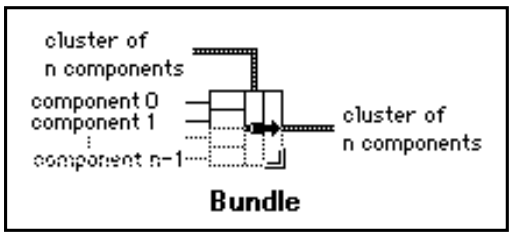
همان گونه که قبلاً اشاره شد کانکتور یک برنامه حداکثر ۲۸ ترمینال دارد. در هنگام فراخوانی زیربرنامه معمولاً سیم کشی تمامی ترمینال های کانکتور الزامی نیست؛ زیرا در این صورت، سیم کشی و برقرار نمودن ارتباط، بسیار مشکل و کسل کننده خواهد بود. با دسته بندی چندین المان کنترل یا نشان دهنده در یک کلاستر

می توانید از یک ترمینال استفاده نموده، آن را برای ارسال چند مقدار به/از زیربرنامه سیم کشی کنید. واضح است که در این حالت انجام سیم کشی به مراتب ساده تر است و در نتیجه احتمال بروز خطا کاسته خواهد شد.

## دسته بندی داده ها




دستور Bundle (Functions >> Cluster) المان های مجزا را در یک کلاستر دسته بندی می کند و به شما اجازه می دهد تا ترتیب قرار گرفتن المان ها را در کلاستری که قبلاً به وجود آمده است تغییر دهید. زمانی که برای اولین بار آیکن این دستور را بر روی صفحه‌ی نمودار بلوکی قرار می دهید به صورت شکل نشان داده شده در سمت راست ظاهر می شود. برای افزایش تعداد ورودیها، به کمک ابزار Positioning Tool بر روی یکی از گوشه های آیکن مذکور کلیک کنید و ماوس را حرکت دهید. هنگامی که هر یک از ترمینال ها را سیم کشی می کنید یک نماد که بیانگر نوع داده‌ی مربوط به المان سیم کشی شده است بر روی ترمینال مذکور ظاهر می گردد. ترتیب قرار گرفتن المان ها در کلاستر همان ترتیب اعمال ورودیها در دستور Bundle می باشد.

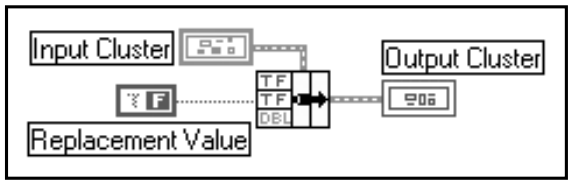



**شکل ۴۰-۷: آیکن دستور Bundle به همراه اتصالات آن**

اگر تصمیم به ایجاد یک کلاستر جدید ندارید، سیم کشی به ورودی مرکزی آیکن دستور Bundle الزامی نیست. تنها در صورت تغییر ترتیب قرار گرفتن المان ها در کلاستر، سیم کشی به ورودی مذکور ضروری است.

## جابه‌جا کردن یک المان در کلاستر

برای جابه‌جا کردن یک یا چند المان در کلاستر، ابتدا اندازه‌ی آیکن دستور Bundle را برای در برگرفتن تمامی ترمینال های ورودی که در واقع المان های موجود در کلاستر هستند، تغییر دهید. این کلاستر باید با تعداد ورودیها هم‌اندازه باشد؛ در غیر این صورت با خطای bad wire مواجه خواهید شد. سپس کلاستر را به ترمینال مرکزی آیکن دستور Bundle سیم کشی کنید. حال مقادیر جدید المان‌ها را به ورودیهای متناظر سیم کشی نمایید. تنها ترمینال هایی که قصد تغییر آنها را دارید سیم کشی کنید. در شکل ۴۱-۷ نحوه‌ی جابه‌جا کردن یک المان جبری در کلاستر را ملاحظه می کنید.



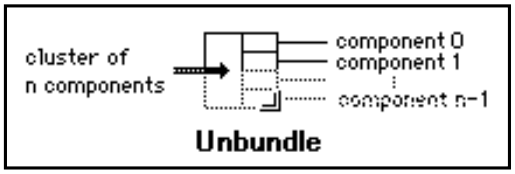

**شکل ۴۱-۷ : نحوه‌ی جابه‌جا کردن یک المان جبری در کلاستر با استفاده از دستور Bundle**

## جدا کردن اجزای یک کلاستر

دستور Unbundle (Functions >> Cluster) هر کلاستر را به اجزای تشکیل دهنده‌ی آن تجزیه می کند. خروجیهای این دستور به همان ترتیبی که در کلاستر قرار گرفته‌اند مرتب می شوند. اگر داده های مربوط به المان ها همه از یک نوع باشند ترتیب قرار گرفتن آنها در کلاستر تنها راه برای تشخیص آنها از یکدیگر است.

Unbundle
Function

هنگامی که آیکن این دستور را برای اولین بار بر روی صفحه‌ی نمودار بلوکی قرار می دهید، به صورت شکل نشان داده شده درسمت راست ظاهر می گردد. برای تغییر اندازه و افزایش تعداد خروجیها، با استفاده از ابزار Positioning Tool بر روی یکی از گوشه های این آیکن کلیک کرده، سپس ماوس را حرکت دهید. این تغییر اندازه باید به گونه ای انجام گیرد که تعداد المان های خروجی با تعداد المان های ورودی کلاستر برابر باشد. در غیر این صورت خطای bad wire بروز خواهد کرد. هنگامی که کلاستر ورودی را به آیکن دستور Unbundle سیم کشی می کنید، ترمینال های خروجی که قبلاً خالی بوده‌اند نماد داده های موجود در کلاستر را به خود می گیرند.




**شکل ۴۲-۷ : آیکن دستور Unbundle به همراه اتصالات آن**

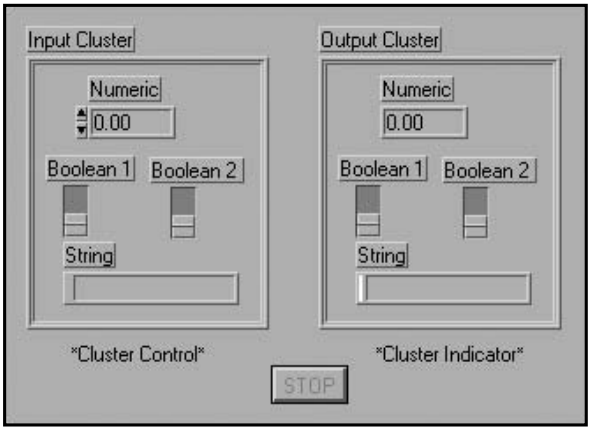
برای دستیابی به داده‌های موجود در کلاستر با استفاده از دستورهای Bundle و Unbundle، اطلاع داشتن از ترتیب قرار گرفتن آنها در کلاستر بسیار مهم و ضروری است. به عنوان مثال اگر دو مقدار جبری در یک کلاستر وجود داشته باشد به دلیل اینکه به المان های کلاستر در دستور Unbundle بر حسب ترتیب قرار گرفتن و نه بر حسب نام آنها مراجعه می‌شود، به راحتی ممکن است در اثر یک اشتباه به المان دیگری دسترسی پیدا کنید. در این حالت در برنامه‌ی شما هیچ خطایی وجود ندارد؛ اما نتایج نادرستی به‌دست خواهید آورد.

در هنگام کار با این دستور مراقب باشید. در صورتی که بدون تغییر دادن اندازه‌ی آیکن دستور Unbundle المانی را به کلاستر اضافه کنید، برنامه‌ی شما معیوب خواهد بود. واضح است که در این حالت دکمه‌ی Run به صورت پیکان شکسته ظاهر می شود.

در نرم‌افزار LabVIEW می توان المان های یک کلاستر را بر حسب نام آنها دسته‌بندی نمود یا آنها را از یکدیگر جدا کرد. در بخشهای آینده این روش را مورد بررسی قرار خواهیم داد.

## تمرین ۴-۷: یک مثال در مورد کلاسترها

در این تمرین به ایجاد یک برنامه پرداخته، روش کار با کلاسترها را خواهید آموخت. در این برنامه پس از ایجاد یک کلاستر، المان های آن را جدا نموده، مجدداً آنها را دسته‌بندی می کنیم. سپس مقادیر المان ها را در یک کلاستر دیگر نمایش می دهیم. پنجره‌ی پانل این برنامه به صورت شکل ۴۳-۷ است.




**شکل ۴۳-۷: صفحه‌ی پانل در برنامه‌ی Cluster Exercise.vi**

۱- یک برنامه‌ی جدید باز نموده، پس از قرار دادن یک پوسته‌ی کلاستر یا ClusterShell (Controls >> Array & Cluster) بر روی صفحه‌ی پانل، برچسب Input Cluster را برای آن تعیین کنید. سپس اندازه‌ی پوسته‌ی کلاستر را افزایش دهید.

۲- در داخل پوسته‌ی Input Cluster یک المان Digital Control، دو کلید Vertical Switch و یک المان String Control قرار دهید.

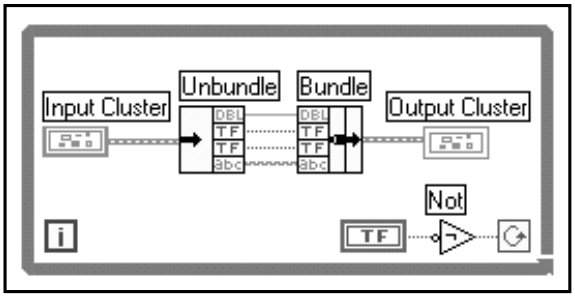
۳- حال پوسته‌ی Output Cluster را از پوسته‌ی Input Cluster کپی برداری کنید. سپس منوی کرکره‌ای را بر روی هر یک از المان های داخلی کلاستر و یا بر روی چهارچوب کلاستر باز نموده، گزینه‌ی Change to Indicator را انتخاب کنید. برچسب کلاستر جدید را نیز تغییر دهید.

برای ایجاد Output Cluster می توانید از همان روشی که در مورد Input Cluster استفاده نمودید کمک بگیرید. البته در این مرحله به جای المان های کنترل از نشان دهنده ها استفاده کنید. حتماً المان ها را به همان ترتیبی که در کلاستر ورودی قرار گرفته اند ایجاد کنید.

۴- ترتیب قرار گرفتن المان ها را در کلاسترهای Input Cluster و Output Cluster بررسی کنید. برای انجام این عمل منوی کرکره‌ای را بر روی چهارچوب هر یک از کلاسترها باز نموده، گزینه‌ی Cluster Order... را انتخاب کنید. در صورتی که ترتیب قرار گرفتن المان ها متفاوت است، ترتیب المان ها را در یکی از دو کلاستر به گونه‌ای تغییر دهید تا ترتیب قرار گرفتن آنها در هر دو کلاستر یکسان باشد.

۵- المان Rectangular Stop Button (Controls >> Boolean) را بر روی صفحه‌ی پانل قرار دهید. توجه داشته باشید که این دکمه به صورت پیش فرض در وضعیت False قرار دارد. وضعیت این دکمه را تغییر ندهید.

۶- صفحه‌ی نمودار بلوکی این برنامه را مطابق شکل ۴۴-۷ ایجاد کنید. توجه کنید که اگرچه در هر یک از کلاسترها ۴ المان وجود دارد، اما تنها یک ترمینال را برای هر کلاستر مشاهده می کنید.




**شکل ۴۴-۷: صفحه‌ی نمودار بلوکی در برنامه‌ی Cluster Exercise.vi**

در ادامه، عملکرد برخی از دستورهای استفاده شده در صفحه‌ی نمودار بلوکی را مورد بررسی قرار می دهیم:

Unbundle Function

دستور Unbundle (Functions >> Cluster) کلاستر را به اجزای تشکیل دهنده‌ی آن تجزیه می کند و دسترسی به المان‌های موجود در آن را امکان‌پذیر می سازد. اندازه‌ی آیکن این دستور را به گونه‌ای تغییر دهید که گنجایش ۴ خروجی را داشته باشد. پس از سیم کشی این دستور به Input Cluster، برچسب نوع داده‌ها بر روی آیکن دستور Unbundle ظاهر می گردد.

Bundle Function

دستور Bundle (Functions >> Cluster) مجدداً کلاستر را بازسازی می کند. اندازه‌ی آیکن این دستور را نیز به گونه ای افزایش دهید تا قابلیت دریافت ۴ ورودی را داشته باشد.

Not Function

دستور Not (Functions >> Boolean) وضعیت دکمه‌ی STOP را معکوس می کند. بنابراین قبل از شروع برنامه، کلیک کردن بر روی دکمه‌ی True (حرف T در مقدار جبری) الزامی نیست. به عبارت دیگر در صورتی که دکمه‌ی STOP در وضعیت False باشد دستور Not مقدار جبری True را به ترمینال شرطی در حلقه‌ی While ارسال می کند و هنگامی که دکمه‌ی مذکور تغییر وضعیت می‌دهد و به حالت True در می آید، ترمینال شرطی در حلقه‌ی While مقدار جبری False را از دستور Not دریافت می کند و حلقه را متوقف می سازد. به خاطر داشته باشید که المان های جبری به صورت پیش فرض در وضعیت False قرار دارند.

برای دستیابی به دستورهای Bundle و Unbundle منوی کرکره‌ای را بر روی ترمینال کلاستری که قصد سیم‌کشی آن را دارید، باز کنید و دستور مورد نظر را از زیرمنوی Cluster Tools انتخاب نمایید. در این حالت آیکن دستور انتخاب شده همراه با تعداد دقیق ترمینال‌های ورودی و خروجی بر روی صفحه‌ی نمودار بلوکی ظاهر می‌شود. بنابراین در این حالت افزایش اندازه‌ی آیکن الزامی نیست.

۷- به صفحه‌ی پانل بازگردید و برنامه را اجرا کنید. مقادیر مختلفی را در کلاستر ورودی وارد نموده و ملاحظه کنید که کلاستر خروجی همان مقادیر را نشان می‌دهد. برای متوقف کردن اجرای برنامه دکمه‌ی STOP را فشار دهید.

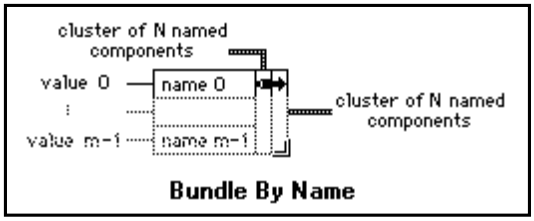
ممکن است در حین اجرای این تمرین متوجه شده باشید که با سیم‌کشی کلاستر ورودی به کلاستر خروجی دقیقاً همین عمل انجام می‌شود؛ اما هدف ما در این برنامه بررسی و درک عملکرد دو دستور Bundle و Unbundle است.

۸- این برنامه را با عنوان Cluster Exercise.vi در زیرفهرست LabVIEW\ My Activity ذخیره نموده، سپس آن را ببندید.

## دسته‌بندی و جداسازی اجزای کلاستر با توجه به نام آنها

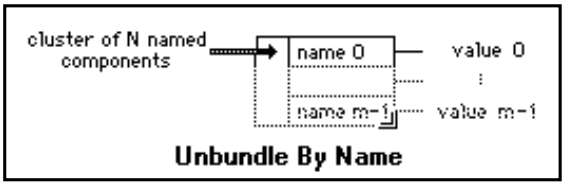
در برخی موارد دسته‌بندی یا جداسازی تمامی اجزای کلاستر الزامی نیست و این اعمال باید در مورد یک یا چند المان صورت گیرد. برای انجام این عمل دو دستور کاربردی دیگر در نرم‌افزار LabVIEW تعبیه شده است که در ادامه به بررسی آنها می‌پردازیم:

دستور Bundle By Name (Functions >> Cluster) به المان‌های موجود در کلاستر برحسب نام آنها (و نه براساس ترتیب و موقعیت قرار گرفتن آنها) مراجعه می‌کند. برخلاف آنچه که در مورد دستور Bundle ذکر شد، می‌توانید به المان/المان‌هایی که نیاز دارید دسترسی پیدا کنید. شایان ذکر است که ایجاد یک کلاستر جدید با استفاده از این دستور امکان‌پذیر نیست. به کمک دستور مذکور تنها می‌توان موقعیت یک یا چند المان را در کلاستری که از قبل ایجاد شده است تغییر داد. برخلاف دستور Bundle در مورد این دستور حتماً باید ترمینال مرکزی آیکن Bundle By Name سیم کشی شده باشد تا به این دستور اعلام شود که المان را باید در کدام کلاستر جابه‌جا کند. در شکل ۴۵-۷ آیکن این دستور نشان داده شده است.




شکل ۴۵-۷: آیکن دستور Bundle By Name به همراه اتصالات آن

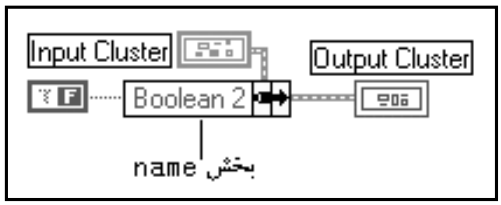
دستور Unbundle By Name (Functions >> Cluster) المان یا المان هایی را که توسط کاربر مشخص می گردد از کلاستر جدا می‌کند. در مورد این دستور اطلاع داشتن از ترتیب قرار گرفتن المان ها در کلاستر و یا اندازه‌ی دقیق و صحیح آن الزامی نیست. در شکل ۴۶-۷ آیکن این دستور را ملاحظه می کنید.




**شکل ۴۶-۷: آیکن دستور Unbundle By Name به همراه اتصالات آن**

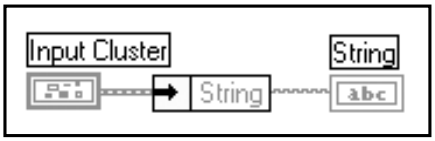
توجه داشته باشید که جهت دسته‌بندی و جداسازی المان‌های موجود در کلاستر با استفاده از دو دستور Bundle By Name و Unbundle By Name تمامی المان‌ها باید دارای یک برچسب منحصر به فرد باشند. واضح است که در صورت عدم انتساب برچسب به المان‌ها، دستیابی به اجزای کلاستر از طریق نام آنها امکان‌پذیر نیست.

به عنوان مثال در صورتی که قصد داشته باشید تا مقدار Boolean 2 را به کمک دستور Bundle By Name در تمرین قبل تغییر دهید به راحتی و بدون در نظر گرفتن اندازه یا ترتیب قرار گرفتن این المان در کلاستر می توانید مطابق شکل ۴۷-۷ این عمل را انجام دهید.




**شکل ۴۷-۷: تغییر مقدار Boolean 2 با استفاده از دستور Bundle By Name**

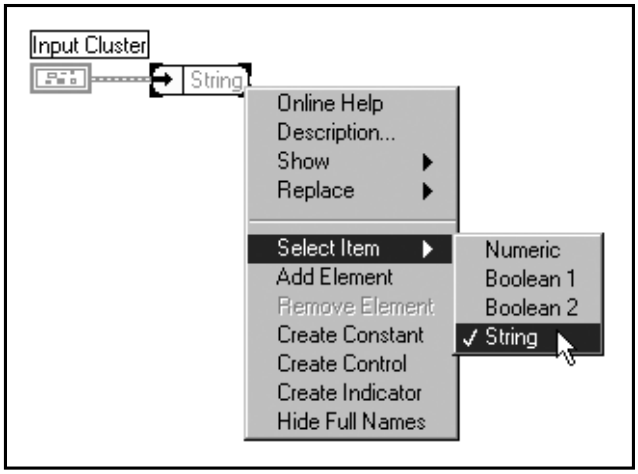
به طریق مشابه برای دسترسی به مقدار String می توانید مطابق شکل ۴۸-۷ از دستور Unbundle By Name کمک بگیرید.




**شکل ۴۸-۷: دسترسی به مقدار String با استفاده از دستور Unbundle By Name**

به محض سیم‌کشی کلاستر ورودی در دو دستور Bundle By name و Unbundle By Name نام اولین المان در کلاستر در بخش name ظاهر می گردد. برای دستیابی به المان‌های دیگر، توسط ابزار Operating Tool یا Labeling Tool بر روی محل در نظر گرفته شده برای نام المان کلیک کنید. در این حالت لیستی از اسامی المان‌های برچسب‌دار در کلاستر ظاهر می شود. سپس المان مورد نظر را از لیست مذکور انتخاب نمایید. نام این المان در محل

مذکور به نمایش در می‌آید. روش دیگر برای انجام این عمل، باز کردن منوی کرکره‌ای بر روی محل درج نام و انتخاب گزینه‌ی Select Item می‌باشد. در این حالت می‌توان المان مورد نظر را از زیرمنوی ظاهر شده انتخاب نمود.

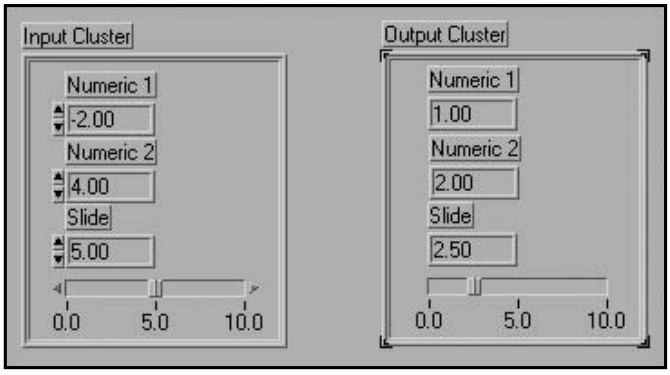



**شکل ۴۹-۷: دستیابی به المان‌های کلاستر ورودی در دو دستور Bundle By name و Unbundle By Name با استفاده از منوی کرکره‌ای**

اندازه‌ی هر دو دستور مذکور برای گنجایش و هماهنگی با تعداد المان‌های مورد نیاز شما قابل تغییر است.

## تمرین ۵-۷: تمرین بیشتر با کلاسترها

در این تمرین برنامه‌ای می‌نویسیم که مقدار یک المان Digital Control را که با برچسب Number 1 مشخص شده است نسبت به صفر می‌سنجد و بررسی می‌کند که مقدار مذکور بزرگتر یا مساوی صفر است. در صورتی که این عدد کوچکتر از صفر باشد، قدر مطلق تمامی المان‌های کنترل محاسبه می‌شود و اگر مقدار Number 1 بزرگتر یا مساوی صفر باشد قدر مطلق هیچ یک از المان‌ها محاسبه نمی‌گردد. صرف نظر از مقدار Number 1، در این برنامه تمامی مقادیر در عدد ثابت 0.5 ضرب شده، حاصل این عملیات در Output Cluster نمایش داده می‌شود. این بخش از برنامه نشان می‌دهد که می‌توان اصل Polymorphism را در مورد کلاسترها نیز به کار برد. پنجره‌ی پانل این برنامه به صورت شکل ۵۰-۷ است.




**شکل ۵۰-۷: صفحه‌ی پانل در برنامه‌ی Cluster Comparison.vi**

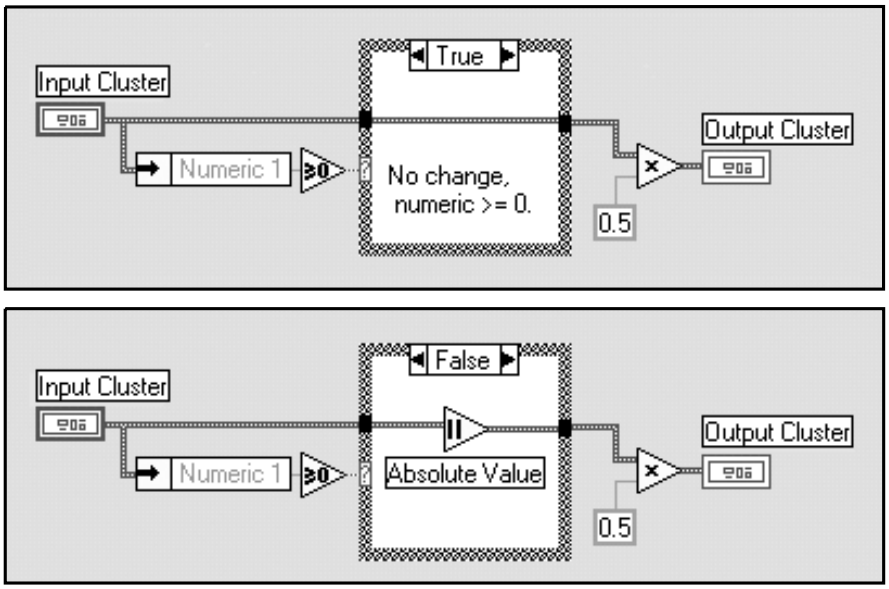
۱- یک برنامه‌ی جدید باز نموده، سپس یک پوسته‌ی آرایه یا Cluster Shell (Controls >> Array & Cluster)

بر روی صفحه‌ی پانل قرار دهید. این کلاستر را با برچسب Input Cluster نام گذاری کنید.

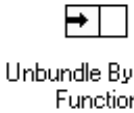
۲- سه المان Digital Control از زیرپالت Numeric << Controls انتخاب کنید و آنها را در داخل پوسته‌ی کلاستر قرار دهید. سپس آنها را با برچسبهای Number 1، Number 2 و Slide نام گذاری کنید. حتماً المان‌های مذکور را به ترتیبی که در شکل ۷-۵۰ نشان داده شده است در پوسته‌ی کلاستر قرار دهید.

۳- اکنون کلاستر دیگری با برچسب Output Cluster ایجاد کنید و سه المان Digital Indicator با برچسبهای تعیین شده در کلاستر Input Cluster در آن قرار دهید. ترتیب قرار گرفتن المان‌ها در هر دو پوسته باید یکسان باشد. روش دیگر برای ایجاد کلاستر Output Cluster، کپی برداری از کلاستر Input Cluster و سپس تغییر برچسب آن است. در این حالت باید گزینه‌ی Change to Indicator را از منوی کرکره‌ای هر یک از المان‌ها انتخاب نموده، بدین ترتیب المان‌های کنترل را به نشان‌دهنده تبدیل کنید.

۴- صفحه‌ی نمودار بلوکی این برنامه را مطابق شکل ۷-۵۱ ایجاد کنید.




**شکل ۷-۵۱: صفحه‌ی نمودار بلوکی در برنامه‌ی Cluster Comparison.vi**




دستور Unbundle By Name (Functions >> Cluster) مقدار المان Number 1 را از کلاستر Input Cluster استخراج می‌کند. پس از به‌دست آوردن مقدار المان مذکور می‌توان آن را با عدد 0 مقایسه نمود. اگر عبارت Number 1 در خروجی این دستور دیده نمی‌شود، با ابزار Operating Tool بر روی آن کلیک نموده، آن را از لیست المان‌های موجود در کلاستر Input Cluster انتخاب کنید.

دستور Greater Or Equal to 0? (Functions >> Comparison) مقدار المان Number 1 را با عدد 0 مقایسه می کند.

دستور Absolute Value (Functions >> Numeric) در صورتی که عدد ورودی بزرگتر یا مساوی صفر باشد، همان مقدار را به‌دست می‌دهد و اگر این عدد کوچکتر از صفر باشد خروجی این دستور قرینه‌ی عدد ورودی خواهد بود. در این تمرین از دستور Absolute Value برای محاسبه‌ی قدر مطلق تمامی المان‌های کلاستر استفاده می‌شود.

۵- به صفحه‌ی پانل بازگردید و پس از اعمال مقادیر مثبت یا منفی به المان Number 1، برنامه را اجرا کنید. در این تمرین به کاربرد اصل Polymorphism در مورد ضرب تمامی المان‌های کلاستر در عدد ثابت 0.5 به طور همزمان و همچنین محاسبه‌ی قدرمطلق تمامی المان‌های کلاستر توجه کنید.

۶- این برنامه را با عنوان Cluster Comparison.vi در زیرفهرست LabVIEW\My Activity ذخیره نموده، سپس آن را ببندید.

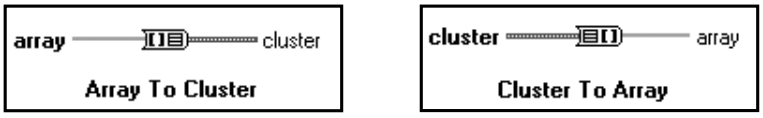
۷

## قابلیت تبدیل آرایه و کلاستر به یکدیگر

برخی مواقع تبدیل نمودن آرایه به کلاستر و برعکس، انجام بعضی از اعمال و دستورها را بسیار ساده‌تر می‌سازد. انجام این عمل بسیار مفید و کاربردی است؛ زیرا در محیط LabVIEW تعداد دستورهایی که بر روی آرایه‌ها اعمال می‌شوند بیشتر از دستورهایی است که در مورد کلاسترها وجود دارد.

به عنوان مثال ممکن است در برنامه‌ای قصد داشته باشید تا وضعیت تمامی دکمه‌های موجود در کلاستر را که در صفحه‌ی پانل قرار دارند معکوس کنید. در این حالت کافی است از دستور Reverse 1D Array استفاده نمایید؛ اما مشکل اینجاست که دستور مذکور تنها بر روی آرایه‌ها قابل اعمال است.

جای هیچ نگرانی نیست؛ زیرا می‌توانید جهت تبدیل یک کلاستر به آرایه از دستور Cluster to Array استفاده نموده، سپس به کمک دستور Reverse 1D Array تمامی مقادیر مذکور را معکوس کنید. پس از انجام این عمل به کمک دستور Array to Cluster مجدداً از آرایه به کلاستر تغییر حالت دهید.

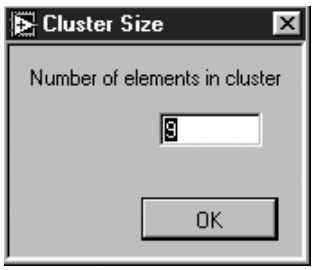



**شکل ۵۲-۷: آیکن دو دستور Cluster to Array و Array to Cluster**

دستور Cluster to Array، یک کلاستر شامل N المان را که همگی دارای یک نوع داده هستند به یک آرایه با N مؤلفه از همان نوع داده تبدیل می‌کند. اندیس المان موجود در آرایه، متناظر با شماره‌ی همان المان در کلاستر است. به عنوان مثال المان شماره‌ی 0 درکلاستر متناظر با مؤلفه‌ی آرایه با اندیس 0 خواهد بود. شایان ذکر است که استفاده از این دستور در مورد کلاستری که المان‌های آن را آرایه‌ها تشکیل می‌دهند امکان‌پذیر نیست؛ زیرا در محیط LabVIEW ایجاد آرایه‌ای که مؤلفه‌های آن نیز آرایه باشد مجاز نیست. توجه داشته باشید برای استفاده از این دستور باید همه‌ی المان‌های موجود در

کلاستر از یک نوع باشند.

دستور Array to Cluster یک آرایه‌ی یک بعدی حاوی N المان را به یک کلاستر حاوی N المان از همان نوع داده تبدیل می کند. در مورد این دستور حتماً باید منوی کرکره‌ای را بر روی دستور Array to Cluster باز کنید و جهت تعیین اندازه‌ی کلاستر خروجی، گزینه‌ی ...Cluster Size را انتخاب نمایید؛ زیرا بر خلاف آرایه، کلاستر اندازه‌ی خود را به طور خودکار تغییر نمی دهد. در صورت انتخاب این گزینه یک پنجره‌ی محاوره‌ای مطابق شکل ۵۳-۷ جهت تعیین تعداد المان های موجود در کلاستر باز می شود.




**شکل ۵۳-۷: پنجره‌ی محاوره‌ای Cluster Size**

همان گونه که در شکل ۵۳-۷ ملاحظه می کنید، اندازه‌ی کلاستر به صورت پیش فرض برای ۹ المان در نظر گرفته می شود. اگر تعداد مؤلفه های آرایه کمتر از تعداد تعیین شده در اندازه‌ی کلاستر باشد، LabVIEW به طور خودکار مقادیر المان های اضافی کلاستر را با مقادیر از پیش تعیین شده از همان نوع داده در کلاستر پر می کند. در صورتی که تعداد مؤلفه‌های آرایه‌ی ورودی بیش از تعداد مشخص شده در اندازه‌ی کلاستر باشد، سیم منتهی به کلاستر خروجی در صفحه‌ی نمودار بلوکی به صورت منقطع ظاهر خواهد شد و تا زمانی که اندازه‌ی پنجره را تغییر نداده‌اید به همین صورت باقی می ماند.

در صورتی که قصد داشته باشید تا المان های کنترل یا نشان دهنده‌ی موجود در کلاستر را در صفحه‌ی پانل به نمایش در آورید، استفاده نمودن از این دو دستور بسیار مفید خواهد بود؛ البته به شرطی که مقادیر اندیس المان ها را در صفحه‌ی نمودار بلوکی مشخص کنید. برای سهولت دستیابی، این دو دستور مفید و کاربردی در هر دو زیرپالت Array و Cluster در پالت Functions گنجانده شده اند.

## خلاصه

یک آرایه مجموعه‌ای از المان هاست که همگی از نظر نوع داده یکسان می‌باشند و به ترتیب خاصی که به آن اندیس گویند در پوسته‌ی آرایه قرار گرفته‌اند. در نرم‌افزار LabVIEW مؤلفه های آرایه از هر نوع ممکن است باشند به استثنای نمودار، گراف و یا آرایه دیگر. برای ایجاد یک آرایه لازم است تا دو مرحله را پشت سر بگذارید. ابتدا پوسته‌ی آرایه یا Array Shell (Controls >> Array & Cluster) را بر روی صفحه‌ی پانل قرار داده، سپس المان های کنترل یا نشان دهنده را به پوسته‌ی آرایه اضافه کنید. در بسته‌ی نرم‌افزاری LabVIEW دستورهای بسیاری نظیر Index Array و Build Array (Functions >> Array) جهت انجام عملیات بر روی آرایه ها تعبیه شده است.

حلقه های For و While می توانند با استفاده از روش اندیس گذاری خودکار، آرایه ها را بر روی چهارچوب خود ایجاد کنند. روش اندیس گذاری خودکار یک ویژگی مهم و کاربردی جهت ایجاد، پردازش و انجام عملیات بر روی آرایه هاست. به یاد داشته باشید که روش اندیس گذاری خودکار به صورت پیش فرض در دو حلقه‌ی For و While به ترتیب فعال و غیرفعال می باشد.

اصل Polymorphism به مفهوم قابلیت انجام یک دستور بر روی داده‌های ورودی با اندازه و نوع متفاوت است. در این فصل ویژگی مذکور را تنها در مورد توابع و دستورهای محاسباتی مورد بررسی قرار دادیم؛ اما این اصل در مورد دستورها و توابع دیگر نیز صادق است.

برای دسته بندی داده ها از کلاستر استفاده می شود؛ اما برخلاف آرایه در کلاستر می توان انواع مختلف داده ها را به کار برد. جهت ایجاد کلاستر انجام دو مرحله لازم است. ابتدا یک پوسته‌ی کلاستر یا Cluster Shell (Controls >> Array & Cluster) را بر روی صفحه‌ی پانل قرار داده، سپس المان های کنترل یا نشان دهنده را به پوسته اضافه کنید. به یاد داشته باشید که المان های یک کلاستر باید همگی از نوع کنترل و یا همگی از نوع نشان دهنده باشند. به عبارت دیگر به کارگیری هر دو المان کنترل و نشان دهنده در یک کلاستر مجاز نیست.

کلاسترها جهت کاهش دادن تعداد سیمها و ترمینال های موجود در برنامه نیز به کار می روند. به عنوان مثال فرض کنید که در برنامه‌ای تعداد زیادی المان کنترل و نشان دهنده وجود دارد که سیم کشی تمامی آنها بسیار وقت گیر است. در این حالت می توانید آنها را در کلاسترهای جداگانه دسته بندی کنید و سیم کشی را با تعداد ترمینال کمتری انجام دهید.

دستور Unbundle (Functions >> Cluster) یک کلاستر را به اجزای تشکیل دهنده‌ی آن تجزیه می کند. دستور Unbundle By Name نیز نظیر دستور Unbundle عمل می کند. با این تفاوت که به المان های موجود در کلاستر بر حسب نام آنها دسترسی پیدا می کند. برای دسترسی به یک یا چند المان از دستور Unbundle By Name استفاده کنید و برای دستیابی به تمام المان های موجود در کلاستر دستور Unbundle را به کار برید. در مورد استفاده از دستور Unbundle حتماً ترتیب قرار گرفتن المان ها و همچنین اندازه‌ی صحیح ترمینال ها را در نظر داشته باشید.

دستور Bundle (Functions >> Cluster) المان های مجزا را در یک کلاستر دسته بندی می‌کند یا موقعیت یک المان را در کلاستر تغییر می دهد. ایجاد کلاستر به کمک دستور Bundle by Name امکان پذیر نیست؛ اما با استفاده از این دستور می توان موقعیت المان های مجزا در کلاستر را بدون دستیابی به کل کلاستر تغییر داد. در هنگام به کارگیری دستور Bundle By Name در نظر گرفتن ترتیب المان ها اهمیتی ندارد. در صورتی که قصد دارید المان ها را با استفاده از دستورهای Bundle By Name یا Unbundle By Name دسته بندی نموده، یا آنها را جدا کنید، حتماً باید تمامی المان ها برچسب منحصر به فردی داشته باشند.

## تمرینات اضافی

### تمرین ۶-۷: معکوس نمودن ترتیب مؤلفه‌های آرایه

برنامه‌ای ایجاد کنید که ترتیب قرار گرفتن مؤلفه‌های یک آرایه را که شامل 100 عدد تصادفی است معکوس کند. به عنوان مثال مؤلفه‌ی اندیس 0 جایگزین مؤلفه‌ی اندیس 99 شود (و برعکس) و به همین ترتیب الی آخر. این برنامه را با عنوان Reverse Random Array.vi در زیرفهرست LabVIEW \ My Activity ذخیره کنید.

### تمرین ۷-۷: استخراج یک زیرآرایه از آرایه‌ی اصلی

برنامه‌ای ایجاد کنید که ابتدا یک آرایه با 100 مؤلفه تولید کند (مؤلفه‌ها اعداد تصادفی هستند)، سپس بخشی از این آرایه را به نمایش در آورد. این برنامه را با عنوان Random Array Subset.vi در زیرفهرست LabVIEW \ My Activity ذخیره کنید.

جهت استخراج زیرمجموعه‌ای از یک آرایه، از دستور Array Subset (Functions >> Array) استفاده کنید.

### تمرین ۸-۷: شبیه‌سازی پرتاب تاس

برنامه‌ای ایجاد کنید که عملکرد یک تاس را شبیه‌سازی کند. همچنین تعداد پرتاب‌های تاس را در خود ذخیره نماید. ورودی برنامه تعداد دفعاتی است که تاس پرتاب می‌شود و خروجی نیز تعداد دفعاتی است که تاس بر روی اعداد ۱ الی ۶ قرار می‌گیرد. این برنامه را با عنوان Dice Roller.vi در زیرفهرست LabVIEW \ My Activity ذخیره کنید.

جهت نگهداشتن مقدار به‌دست آمده از هر تکرار حلقه و انتقال آن به تکرار بعدی از ثبات انتقال استفاده کنید.

### تمرین ۹-۷: ضرب کردن مؤلفه‌های آرایه در یکدیگر

برنامه‌ای ایجاد کنید که مؤلفه‌های یک آرایه‌ی یک‌بعدی را به عنوان ورودی دریافت کند. سپس آنها را دوبه‌دو در هم ضرب نموده، نتیجه را در یک آرایه‌ی خروجی قرار دهد. به عنوان مثال اگر مؤلفه‌های آرایه‌ی ورودی به صورت ۱, ۲۳, ۱۰, ۵, ۷, ۱۱ باشند مؤلفه‌های آرایه‌ی خروجی به صورت ۲۳, ۵۰, ۷۷ خواهند بود. این برنامه را با عنوان Array Pair Multiplier.vi در زیرفهرست LabVIEW \ My Activity ذخیره کنید.

LabVIEW

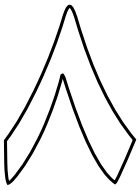
# ۸

# گراف و نمودار

## در این فصل با موارد زیر آشنا می‌شوید:

- روش به‌کارگیری نمودار و گراف
- مدهای نمایشی در یک نمودار (Sweep، Scope و Strip)
- حرکات مکانیکی در کلیدهای جبری
- انواع داده‌های قابل قبول برای گراف و نمودار جهت ترسیم نقشه‌های منفرد و چندگانه
- سفارشی کردن شکل ظاهری نمودار وگراف با تغییر مقیاس و به‌کارگیری بخشهای Palette، Legend و نشانگرها
- قابلیت شدت رنگ در گراف و نمودار جهت ترسیم نقشه‌های سه‌بعدی

# LabVIEW



## گراف و نمودار

در نرم‌افزار LabVIEW برای نمایش گرافیکی داده‌ها و نتایج به‌دست آمده، از گراف یا نمودار استفاده می‌شود. با استفاده از نمودار می‌توان داده‌های جدید را به همراه داده‌های قدیمی نمایش داد؛ اما گراف‌ها مقادیر از پیش ایجاد شده در آرایه‌ها را بدون نگه داشتن داده‌های تولید شده‌ی قبلی ترسیم می‌کنند. در این فصل مطالبی در مورد گراف، نمودار، انواع داده‌ی مناسب برای هر یک، روش به‌کارگیری آنها و برخی از ویژگیهای خاص و قابل توجه آنها را فرا خواهید گرفت. همچنین با قابلیت خاص گراف و نمودار در ترسیم داده‌های سه‌بعدی بر روی صفحه‌ی دوبعدی آشنا خواهید شد.

در این فصل اهداف زیر دنبال می‌شوند:

- فراگیری کاربردها و روش به‌کارگیری نمودار و گراف.
- توانایی تشخیص سه مد نمایشی در یک نمودار (Sweep، Scope و Strip).
- فراگیری حرکات مکانیکی در کلیدهای جبری.
- درک تفاوت عملکرد گراف و نمودار.
- به‌دست‌آوردن اطلاعاتی در مورد انواع داده‌های قابل قبول برای گراف و نمودار جهت ترسیم نقشه‌های منفرد و چندگانه.
- سفارشی کردن شکل ظاهری نمودار و گراف با تغییر مقیاس و به‌کارگیری بخشهای Palette، Legend و نشانگرها.
- استفاده از قابلیت شدت رنگ در گراف و نمودار جهت ترسیم نقشه‌های سه‌بعدی.

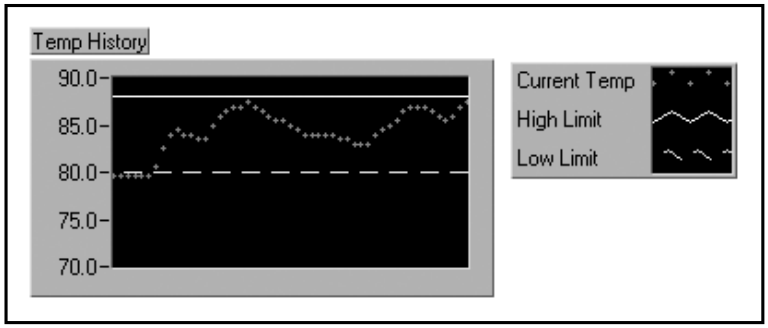
اصطلاحات کلیدی موجود در این فصل عبارتند از:

- منحنی (Plot)
- نمودار شکل موج (Waveform Chart)
- راهنمای نقشه (Legend)
- مدهای مختلف نمایشی در نمودار (Sweep Mode، Scope Mode و Strip Mode)
- نشانگرها (Cursors)
- Palette
- حرکت مکانیکی کلیدهای جبری (Mechanical Action)
- گراف شکل موج (Waveform Graph)
- XY Graph
- نمودار و گراف شدت رنگ (Intensity Charts and Graphs)

## نمودار

به بیان ساده نمایش دادن مقادیر Y در مقابل مقادیر X را ترسیم نمودار گویند. در اکثر نمودارها مقدار داده‌ها و نتایج به دست آمده بر روی محور Y به نمایش در می‌آیند. حال آن که مقادیر X برای نشان دادن زمان به کار می‌روند. گزینه‌ی Waveform Chart که در زیرپالت Controls >> Graph قرار دارد، شامل یک صفحه‌ی نمایش است که می‌تواند یک یا چند نمودار از داده‌ها و نتایج به دست آمده را نمایش دهد. در اکثر مواقع ترمینال این المان در داخل حلقه‌ها مورد استفاده قرار گرفته، داده‌های به‌دست آمده‌ی قبلی را نیز در خود نگه می‌دارد و آنها را به نمایش در می‌آورد. در نمودارها قابلیت اضافه کردن داده‌های جدید به اطلاعات قبلی تعبیه شده است و در صورتی که داده‌های جدید در دسترس باشند می‌تواند آنها را به اطلاعات قبلی اضافه کند.

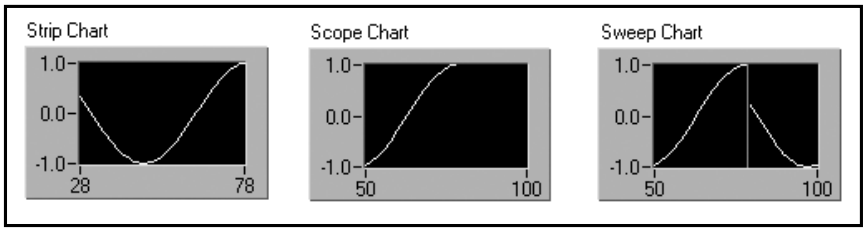
همان گونه که قبلاً ذکر شده مقادیر X و Y در یک نمودار به ترتیب برای نمایش داده‌ها و زمان به کار برده می‌شوند. در اکثر موارد مقادیر Y، در تکرارهای یک حلقه تولید می‌شوند. بنابراین مقدار X نشان‌دهنده‌ی زمان لازم برای تکرار یک حلقه خواهد بود. در نرم‌افزار LabVIEW تنها یک نوع نمودار وجود دارد. اما جهت به روز رساندن داده‌ها در نمودار، سه مد متفاوت در نظر گرفته شده است. در شکل ۱-۸ یک نمونه از نمودار شکل موج چندگانه نشان داده شده است. همان گونه که ملاحظه می‌کنید در این حالت چند منحنی بر روی صفحه‌ی نمودار ترسیم شده است.




**شکل ۱-۸: نمودار شکل موج چندگانه**

## مدهای مختلف نمایش نمودار

در نمودار شکل موج، جهت به روز رساندن داده ها از سه مد متفاوت استفاده می شود. این حالات عبارتند از: Strip Chart Mode، Scop Chart Mode و Sweep Chart Mode. در شکل ۲-۸ این مدها را ملاحظه می کنید.




**شکل ۲-۸: مدهای مختلف نمودار جهت به روز رساندن داده ها**

برای تعویض مدهای مذکور منوی کرکره‌ای را بر روی Wavefrom Chart باز کنید و مد دلخواه را از زیرمنوی Data Operations >> Update Mode انتخاب نمایید. در صورتی که در هنگام اجرای برنامه قصد تعویض مدهای مذکور را داشته باشید گزینه‌ی Update Mode را از منوی کرکره‌ای نمودار انتخاب کنید. همان گونه که قبلاً ذکر شد، ممکن است منوهای موجود در دو مد Edit و Run تغییرات اندکی داشته باشند.

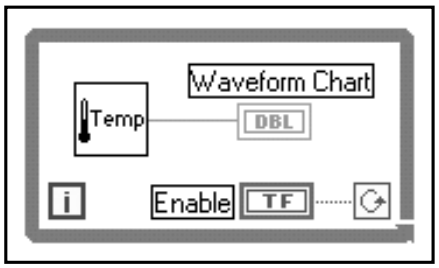
مد Strip Chart دارای یک صفحه‌ی نمایش دهنده‌ی لغزشی است. مدهای Scope و Sweep دارای صفحات نمایش دنبال کننده شبیه به یک اسیلوسکوپ هستند. در مد Scope Chart وقتی که منحنی به مرز سمت راست در ناحیه‌ی ترسیم می‌رسد و صفحه‌ی نمایش پر می شود، نمودار به طور کامل پاک می‌شود و مجدداً از سمت چپ صفحه آغاز می گردد. عملکرد مد Sweep Chart نیز شبیه به مد Scope Chart است. با این تفاوت که هنگامی که صفحه‌ی نمایش کاملاً پر می شود صفحه‌ی نمایش به طور کامل پاک نمی گردد. همان گونه که در شکل ۲-۸ ملاحظه می کنید در این مد، یک خط موازی با مرزهای سمت چپ و راست صفحه بر روی صفحه‌ی نمایش وجود دارد. این خط عمودی نقطه‌ی شروع نمایش داده های جدید را مشخص می کند و در حالی که داده های جدید به نمایش در می آیند این خط نیز به سمت چپ حرکت می کند. درک تفاوت بین مدهای مذکور در زمانی که دقیقاً عملکرد آنها را مشاهده کنید، ساده تر خواهد بود. در صورتی که تفاوت بین این مدها را به خوبی درک نکرده اید جای هیچ نگرانی نیست؛ زیرا در تمرین بعدی با مدهای مختلف به روز رسانی داده ها به طور کامل آشنا خواهید شد. به دلیل این که در هنگام دنبال نمودن یک نمودار سربار[1] کمتری وجود دارد، سرعت عملکرد دو مد Scope Chart و Sweep Chart بیش از سرعت مد Strip Chart می باشد.

## نمودار شکل موج منفرد

ساده‌ترین روش جهت استفاده از یک نمودار، اتصال یک مقدار عددی به ترمینال آن در صفحه‌ی نمودار بلوکی است. در این حالت در هر یک از تکرارهای حلقه تنها یک نقطه به صفحه‌ی نمایش نمودار اضافه می شود. در صفحه‌ی نمودار بلوکی نشان داده شده در شکل ۳-۸، در هر یک از تکرارهای حلقه‌ی While

1- Overhead

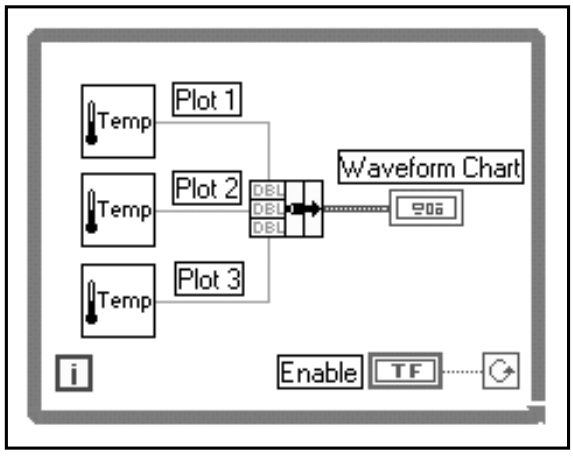
مقدار دمای اندازه‌گیری شده توسط زیربرنامه‌ی Digital Thermometer.vi بر روی صفحه‌ی نمودار نقش می‌بندد.




**شکل ۳-۸: صفحه‌ی نمودار بلوکی در یک نمودار شکل موج منفرد**

## نمودار شکل موج چندگانه

بر روی یک نمودار می‌توان بیش از یک منحنی را به نمایش در آورد؛ اما به دلیل اینکه سیم‌کشی چندین مبدأ داده در صفحه‌ی نمودار بلوکی به یک ترمینال منفرد در نمودار مجاز نیست، ابتدا باید به کمک دستور Bundle (Functions >> Cluster) داده‌ها را دسته‌بندی کنید. در شکل ۴-۸ ملاحظه می‌کنید که خروجیهای حاصل از سه زیربرنامه‌ی متفاوت که دمای نقاط مختلف را اندازه‌گیری می‌کنند توسط دستور Bundle در یک کلاستر دسته‌بندی شده است. بدین ترتیب می‌توان هر سه منحنی را بر روی یک نمودار ترسیم نمود. توجه داشته باشید که می‌توان منحنیهای بیشتری را بر روی همین نمودار ترسیم کرد. برای انجام این عمل ابتدا به کمک ابزار Positioning Tool اندازه‌ی کلاستر را تغییر داده، تعداد ترمینال‌های ورودی دستور Bundle را افزایش دهید. سپس خروجی حاصل از گره‌های دیگر را سیم‌کشی کنید. به تغییرات ظاهری ترمینال نمودار در زمانی که دستور Bundle به آن اتصال می‌یابد توجه کنید.

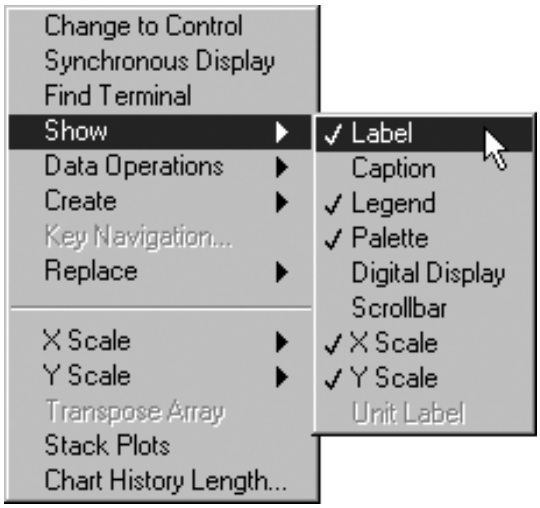



**شکل ۴-۸: صفحه‌ی نمودار بلوکی در یک نمودار شکل موج چندگانه**

جهت کسب اطلاعات بیشتر در مورد نمودار، مدهای مختلف آن و انواع داده‌های متناسب با هر یک، برنامه‌ی Charts.vi را از مسیر LabVIEW\Exercise\Ch8.llb باز و اجرا کنید. برای درک عملکرد هر یک از نمودارهای موجود در صفحه‌ی پانل این برنامه، با کلیک کردن بر روی دکمه‌ی Execution Highlighting در نوار ابزار صفحه‌ی نمودار بلوکی، مد اجرای متمایز را در برنامه‌ی مذکور فعال کرده، سپس آن را اجرا کنید. برای

ایجاد تغییر در سرعت به روز رساندن داده ها از کلید لغزشی update/Pts of # استفاده کنید.

در ادامه‌ی بحث به بررسی برخی از گزینه های موجود در منوی کرکره‌ای نمودار می پردازیم. این گزینه ها را در شکل ۵-۸ ملاحظه می کنید.




شکل ۵-۸

## گزینه های موجود در منوی کرکره‌ای نمودار

### صفحه‌ی نمایش دیجیتالی

۸

در نمودار نیز شبیه به اکثر نشان دهنده های عددی، گزینه ای جهت ظاهر ساختن یا مخفی نمودن صفحه‌ی نمایش دیجیتالی وجود دارد. برای ظاهر ساختن صفحه‌ی مذکور، منوی کرکره‌ای را بر روی نمودار باز نموده، گزینه‌ی Show >> Digital Display را انتخاب کنید. مقدار آخرین داده‌ی ترسیم شده بر روی نمودار، در صفحه‌ی نمایش دیجیتالی نیز نشان داده می شود.

### نوار پیمایش[۲]

در نمودار یک نوار پیمایش وجود دارد که می توانید آن را ظاهر سازید و یا مخفی کنید. به کمک این نوار می توانید داده های قبلی را که از صفحه‌ی نمایش خارج شده‌اند ملاحظه کنید. برای ظاهر ساختن این نوار، منوی کرکره‌ای را بر روی نمودار باز نموده، گزینه‌ی Show >> Scrollbar را انتخاب کنید

### پاک کردن صفحه‌ی نمایش در نمودار

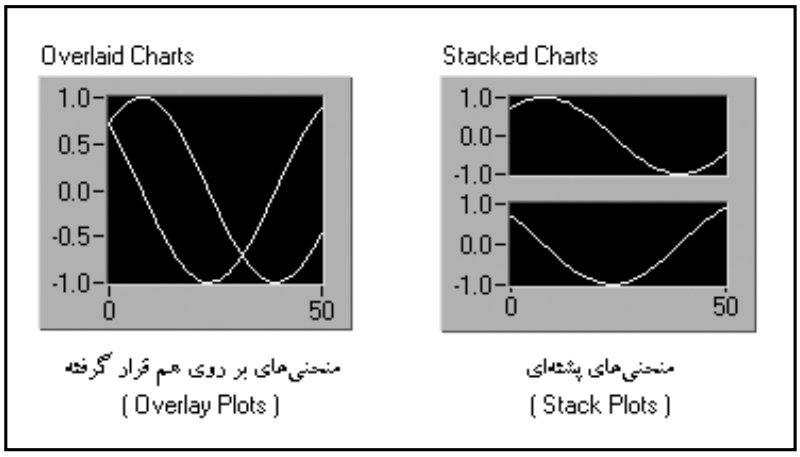
برخی از مواقع لازم است تا تمامی داده های قبلی را از صفحه‌ی نمایش پاک کنید. جهت پاک کردن نمودار در مد Edit، منوی کرکره‌ای را بر روی نمودار باز نموده، گزینه‌ی Data Operations >> Clear Chart را انتخاب کنید. برای پاک کردن نمودار در مد Run، گزینه‌ی Clear Chart را از منوی کرکره‌ای بر روی نمودار انتخاب

2- Scrollbar

نمایید. همان گونه که قبلاً عنوان شد در دو مد Run و Edit تغییرات اندکی در منوها و زیرمنوهای موجود در نرم‌افزار LabVIEW دیده می‌شود. به عنوان مثال گزینه‌ی Clear Chart در مد Run به صورت گزینه‌ای از منوی کرکره‌ای است در حالی که در مد Edit به صورت زیرگزینه‌ای از گزینه‌ی Data Operations می‌باشد.

## گزینه‌های Stack Plots و Overlay Plots

در صورتی که بر روی صفحه‌ی نمایش نمودار بیش از یک منحنی وجود داشته باشد می‌توانید تمامی آنها را بر روی یک صفحه به نمایش در آورید. در این حالت به آن‌ها منحنیهای بر روی هم قرارگرفته (Overlay Plots) گویند. روش دیگر برای ترسیم نمودارها، تخصیص محورهای جداگانه‌ی Y به هر یک از آنهاست که در این حالت به آنها منحنیهای پشته‌ای (Stack Plots) گفته می‌شود. برای ترسیم چند منحنی بر روی صفحه‌ی نمایش یک نمودار، گزینه‌ی Overlay Plots به صورت پیش‌فرض انتخاب می‌شود. بنابراین مطابق شکل ۵-۸ گزینه‌ی Stack Plots در منوی کرکره‌ای نمودار به چشم می‌خورد. برای تغییر وضعیت صفحه‌ی نمایش به حالت دیگر، گزینه‌ی Stack Plots را انتخاب کنید. در صورتی که در این حالت منوی کرکره‌ای را بر روی نمودار باز نمایید، ملاحظه می‌کنید که این بار گزینه‌ی Overlay Plots برای تغییر وضعیت به حالت دیگر در این منو وجود دارد. در شکل ۶-۸ تفاوت بین دو حالت مذکور نشان داده شده است.

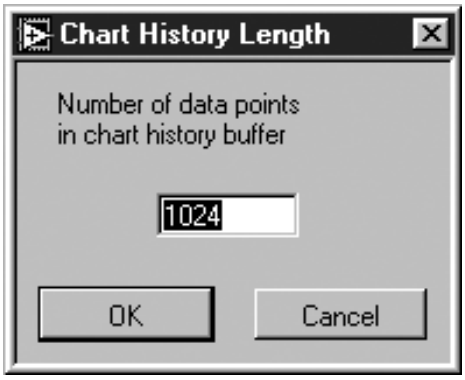



شکل ۶-۸

## گزینه‌ی Chart History Length...

یک نمودار قابلیت ذخیره نمودن حداکثر 1024 نقطه را داراست. این مقدار به صورت پیش‌فرض برای نمودار در نظر گرفته شده است. در صورتی که قصد داشته باشید این مقدار را تغییر دهید و تعداد نقاط کمتر یا بیشتری را ذخیره کنید، منوی کرکره‌ای را بر روی نمودار باز کنید و گزینه‌ی Chart History Length... را انتخاب نمایید. در این حالت یک پنجره‌ی محاوره‌ای نظیر شکل ۷-۸ باز خواهد شد که در آن می‌توانید تعداد نقاط مورد نظر را حداکثر تا 100000 نقطه افزایش دهید. تغییر این مقدار که در یک بافر ذخیره می‌گردد، تأثیری بر تعداد نقاط نشان داده شده در صفحه‌ی نمایش نمودار ندارد. جهت افزایش یا کاهش تعداد نقاط نمایش داده

شده بر روی نمودار، اندازه‌ی صفحه‌ی آن را تغییر دهید. ایجاد تغییر در ظرفیت بافر تنها منجر به تغییر تعداد داده‌هایی می‌شود که می‌توانید با استفاده از نوار پیمایشی به آنها دسترسی پیدا کنید.



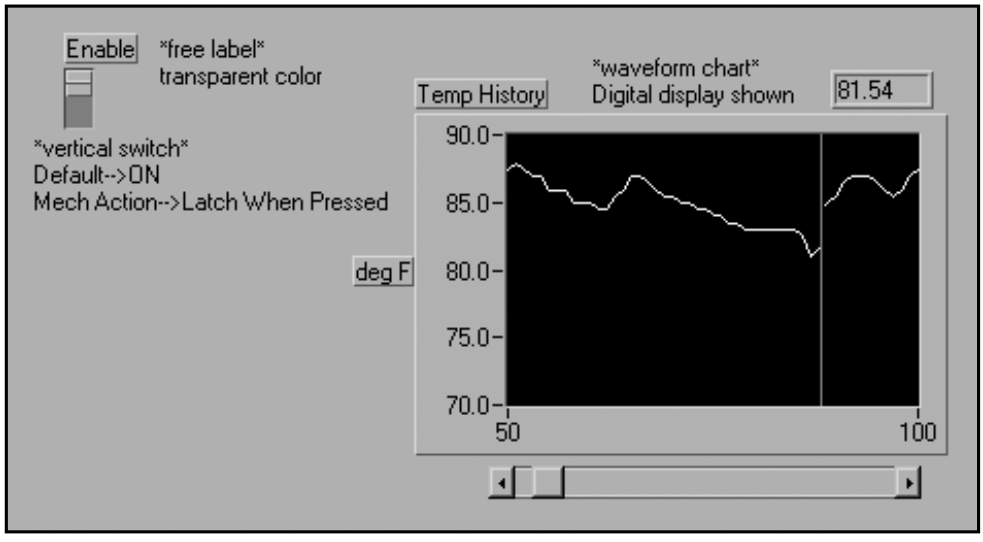

**شکل ۷-۸: پنجره‌ی محاوره‌ای Chart History Length**

## تمرین ۱-۸: اندازه‌گیری دما و به نمایش در آوردن آن

در این تمرین قصد داریم تا با ایجاد یک برنامه، دمای محیط را اندازه‌گیری کنیم و آن را بر روی نمودار نمایش دهیم. در این برنامه اندازه‌گیری دما با استفاده از زیربرنامه‌ی Thermometer.vi انجام می‌شود.

۸

۱- یک برنامه‌ی جدید باز نموده، صفحه‌ی پانل آن را مطابق شکل ۸-۸ ایجاد کنید. تایپ کردن توضیحات مندرج در این شکل الزامی نیست. این توضیحات تنها جهت راهنمایی شما به صفحه‌ی پانل اضافه شده‌اند.




**شکل ۸-۸: صفحه‌ی پانل در برنامه‌ی Temperature Monitor.vi**

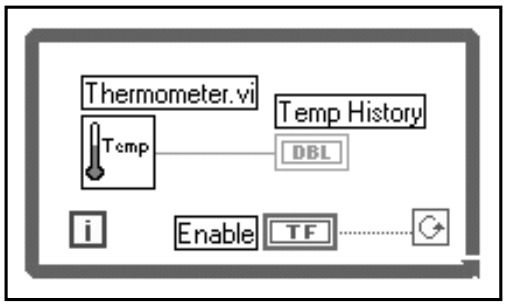
۲- یک کلید عمودی یا Vertical Switch (Controls >> Boolean) را بر روی صفحه‌ی پانل قرار داده، سپس آن را با برچسب Enable نام گذاری کنید. از این کلید برای متوقف نمودن فرآیند اندازه‌گیری دما استفاده می‌شود.

۳- یک نمودار شکل موج یا Waveform Chart (Controls >> Graph) را بر روی صفحه‌ی پانل قرار داده، سپس آن را با برچسب Temp History نام گذاری کنید. این نمودار، درجه حرارت یا دمای کنونی را بلافاصله[3] نشان می دهد.

۴- همان گونه که قبلاً اشاره شد، برای هر نمودار یک صفحه‌ی نمایش دهنده‌ی دیجیتالی در نظرگرفته شده است که به کمک آن می توان مقدار آخرین نمونه‌ی اندازه‌گیری شده را مشاهده نمود. برای ظاهر ساختن صفحه‌ی مذکور، منوی کرکره‌ای را بر روی نمودار باز نموده، سپس گزینه‌ی Show >> Digital Display را انتخاب کنید.

۵- به دلیل اینکه در این برنامه فرض بر آن است که سنسور موجود در سیستم، دمای اتاق را بر حسب درجه‌ی فارنهایت اندازه‌گیری می کند، درجه‌بندی نمودار را به گونه‌ای تغییر دهید تا با تغییرات دمای اتاق سازگار باشد. در غیر این صورت با توجه به درجه‌بندی ترمومتری که در این برنامه استفاده شده است، مقادیر اندازه‌گیری شده خارج از محدوده خواهند بود. برای تغییر درجه‌بندی ترمومتر، توسط ابزار Labeling Tool بر روی عدد 10.0 در مقیاس نمودار کلیک کرده، عدد 90.0 را تایپ کنید. سپس برای وارد نمودن این مقدار بر روی دکمه‌ی Enter در نوار ابزار کلیک کنید. برای وارد کردن مقدار مذکور می توانید از کلید <Enter> در بخش عددی صفحه کلید نیز استفاده کنید. به همین روش حد پایینی درجه‌بندی یعنی مقدار 0.0 را به 70.0 تغییر دهید. بدین ترتیب درجه‌بندی نمودار از 70.0 تا 90.0 تنظیم می گردد.

۶- حال به صفحه‌ی نمودار بلوکی تغییر وضعیت داده، آیکن ها و گره های نشان داده شده در شکل ۹-۸ را در این صفحه قرار دهید.




**شکل ۹-۸: صفحه‌ی نمودار بلوکی در برنامه‌ی Temperature Monitor.vi**

۷- یک حلقه‌ی While (Functions >> Structures) بر روی صفحه‌ی نمودار بلوکی قرار داده، اندازه‌ی آن را مطابق شکل ۹-۸ و با توجه به تعداد المان های قرار گرفته در آن افزایش دهید.

۸- دو ترمینال نشان داده شده در شکل ۹-۸ را در چهارچوب حلقه قرار دهید.

۹- برنامه‌ی Thermometer.vi را به عنوان زیربرنامه در این صفحه وارد کنید. برای انجام این عمل گزینه‌ی Functions >> Select a VI... را انتخاب کنید. به احتمال زیاد این برنامه در فهرست یا کتابخانه‌ی

3- Realtime

My Activity ذخیره شده است. اگر این برنامه را ذخیره نکرده‌اید نگران نباشید. می‌توانید یکی از دو برنامه‌ی Thermometer.vi یا Digital Thermometer.vi را به ترتیب از مسیر LabVIEW\Exercise\Ch5.llb یا زیرپالت Functions >> Tutorial مورد استفاده قرار دهید.

Thermometer.vi

برنامه‌ی Thermometer.vi مقدار دمای اندازه‌گیری شده از سنسور گرمایی یا سیستم شبیه‌سازی شده را به‌دست می‌دهد. اگر به خاطر داشته باشید، این برنامه را در فصل ۴ ایجاد نمودید و در فصل ۵ نیز در آن تغییراتی ایجاد کردید.

۱۰- المان‌ها و ترمینال‌های موجود در صفحه‌ی نمودار بلوکی را مطابق شکل ۹-۸ سیم‌کشی کنید.

۱۱- اکنون به صفحه‌ی پانل بازگردید و برنامه را اجرا کنید. قبل از کلیک کردن بر روی دکمه‌ی Run وضعیت کلید عمودی را به کمک ابزار Operating Tool تغییر دهید. درغیر این صورت حلقه‌ی While تنها یک مرتبه اجرا می‌شود. به یاد آورید که While Loop یک حلقه‌ی بی‌پایان است. بدین معنی که زیربرنامه‌ی قرار گرفته در داخل چهارچوب این حلقه تا زمانی که شرط مشخص شده در برنامه در وضعیت True قرار داشته باشد به اجرا در می‌آید. در این تمرین تا زمانی که کلید عمودی در وضعیت True قرار گرفته باشد، زیربرنامه‌ی Thermometer.vi یک نمونه‌ی جدید را اندازه‌گیری می‌کند و آن را بر روی صفحه‌ی نمودار به نمایش در می‌آورد.

۱۲- برای متوقف کردن عملیات اندازه‌گیری و قرائت دما، بر روی کلید عمودی کلیک کنید. این عمل وضعیت کلید را به حالت False تغییر می‌دهد و باعث توقف اجرای حلقه می‌گردد.

۱۳- همان طور که قبلاً اشاره شد در نمودارها جهت ذخیره‌ی نقاطی که از صفحه‌ی نمایش خارج شده‌اند یک بافر تعبیه شده است. برای مشاهده‌ی مقادیر خارج از صفحه، منوی کرکره‌ای را بر روی نمودار باز نمایید و گزینه‌ی Show >> Scrollbar را برای ظاهر ساختن نوار پیمایش انتخاب کنید. برای تغییر اندازه و تنظیم نمودن نوار مذکور از ابزار Positioning Tool استفاده نمایید. برای مشاهده‌ی نقاط خارج از صفحه‌ی نمایش بر روی هر یک از پیکانهای موجود در نوار پیمایش کلیک کنید. در ضمن می‌توانید با انتخاب گزینه‌ی Chart History Length... از منوی کرکره‌ای و تعیین تعداد نقاط ذخیره شده در بافر، تعداد نقاط بیشتری از داده‌های قبلی را مشاهده کنید. برای پاک کردن بافر و صفحه‌ی نمایش دهنده، منوی کرکره‌ای را بر روی نمودار باز کنید و در محیط Edit گزینه‌ی Data Operations >> Clear Chart را انتخاب کنید. برای پاک کردن بافر مذکور در محیط Run، گزینه‌ی Clear Chart را از منوی کرکره‌ای انتخاب نمایید.

۱۴- با اطمینان از قرار گرفتن کلید عمودی در وضعیت True، مجدداً برنامه را اجرا کنید. این بار مد استفاده شده برای به‌روز رساندن داده‌ها را تغییر دهید. برای انجام این عمل منوی کرکره‌ای را بر روی نمودار باز نموده، سپس گزینه‌ی Update Mode >> Scope Chart را انتخاب کنید. به تفاوت موجود در دو نمودار توجه کنید. حال گزینه‌ی Update Mode >> Sweep Chart را از همان منو انتخاب کنید و عملکرد این مد را نیز ملاحظه کنید.

## بهره‌گیری از حرکات مکانیکی کلیدهای جبری

لحظه‌ای از ادامه‌ی این تمرین دست بکشید. حتماً تا به حال دقت کرده‌اید که جهت اجرا نمودن این برنامه باید قبل از کلیک کردن بر روی دکمه‌ی Run، وضعیت کلید عمودی Enable را به حالت True تغییر دهید. در غیر این صورت حلقه‌ی While تنها یک مرتبه اجرا خواهد شد. برای تغییر عملکرد این کلید و از میان برداشتن این عمل بهتر است عملکرد مکانیکی آن را اصلاح کنید. در نرم‌افزار LabVIEW، ۶ حالت مختلف برای حرکت مکانیکی یک المان کنترل جبری[4] تعبیه شده است. در ادامه به بررسی این حالات می‌پردازیم:

Switch When Pressed

در حالت Switch When Pressed هر مرتبه که با استفاده از ابزار Operating Tool بر روی المان کنترل جبری کلیک می‌کنید مقدار آن المان تغییر می‌یابد. این حالت به صورت پیش‌فرض برای المان‌های جبری در نظر گرفته شده و نظیر همان عملی است که برای روشن کردن چراغ اتاق انجام می‌دهید.

Switch When Released

در حالت Switch When Released تنها در زمانی که دکمه‌ی ماوس را رها می‌کنید وضعیت المان جبری تغییر می‌کند. این حرکت به این موضوع بستگی ندارد که هر چند وقت یک بار برنامه، مقدار المان جبری را قرائت می‌کند. این حالت شبیه به حرکتی است که در هنگام کلیک کردن بر روی علامت «✓» در پنجره‌های محاوره‌ای انجام می‌گیرد. در این حالت، مورد انتخاب شده فعال می‌گردد ولی تا زمانی که دکمه‌ی ماوس را رها نکرده‌اید تغییری در آن ایجاد نمی‌شود.

Switch Until Released

در حالت Switch Until Released زمانی که بر روی المان جبری کلیک می‌کنید مقدار آن تغییر می‌یابد. در این حالت تا زمانی که دکمه‌ی ماوس را رها نکرده‌اید مقدار جدید در المان نگهداری می‌شود. این حالت شبیه به عملکرد بوق هشدار دهنده‌ی درب ورودی است.

Latch When Pressed

در حالت Latch When Pressed در هنگام کلیک کردن بر روی المان، وضعیت آن تغییر می‌کند. این حالت تا زمانی که برنامه مقدار جدید را یک مرتبه قرائت کند و المان کنترل جبری به وضعیت پیش‌فرض خود بازگردد در المان جبری باقی می‌ماند. عملکرد این حالت شبیه به یک سیستم قطع کننده‌ی مدار بوده و (نظیر متوقف نمودن حلقه‌ی While توسط کلید STOP) در مواردی مفید است که قصد داشته باشید که برنامه تنها یک مرتبه مقدار المان کنترل جبری را قرائت کند.

Latch When Released

در حالت Latch When Released تنها پس از رها کردن دکمه‌ی ماوس، وضعیت المان جبری تغییر می‌یابد. هنگامی که برنامه، المان کنترل جبری را یک مرتبه می‌خواند المان

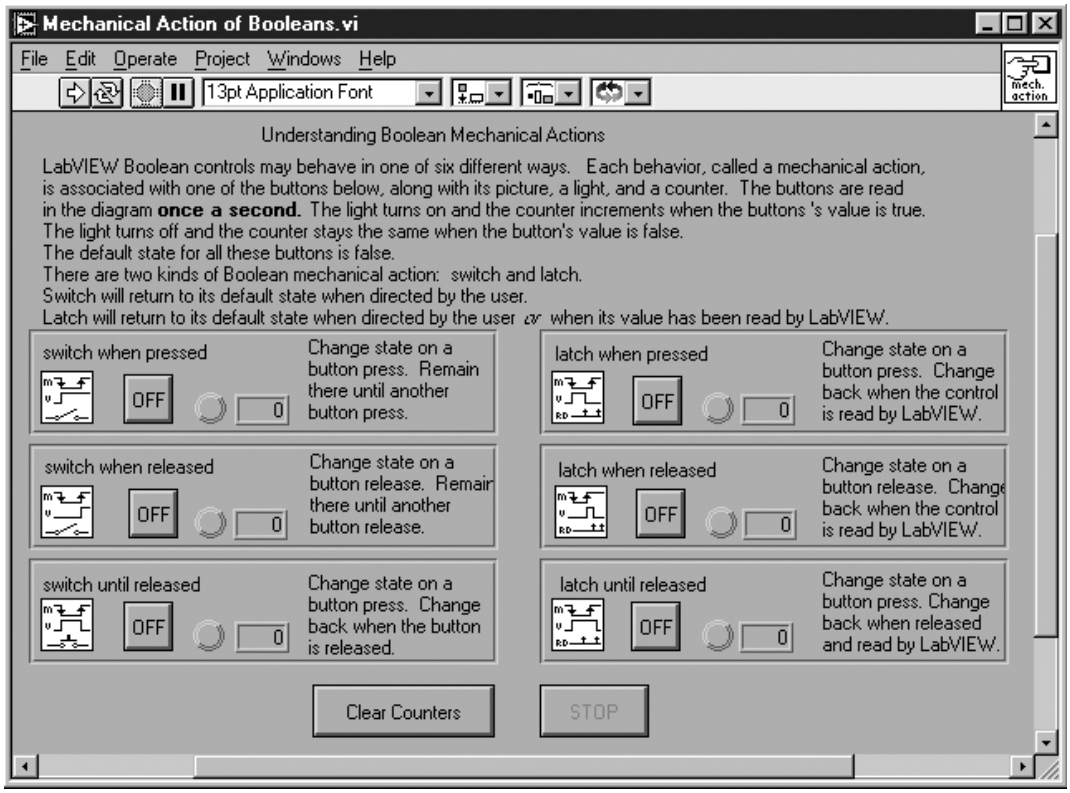
۸

4- Boolean Control

کنترل به وضعیت قبلی خود باز می گردد. این حالت شبیه به عملکرد دکمه های موجود در پنجره های محاوره ای است. همان گونه که می دانید دکمه های مذکور در زمانی که بر روی آنها کلیک می کنید فعال می‌شوند و تا زمان رها کردن ماوس مقدار قرائت شده را در خود نگه می دارند.

در حالت Latch Until Released هنگامی که بر روی المان جبری کلیک می کنید مقدار آن تغییر می کند. تا زمانی که برنامه یک مقدار جدید را بخواند و یا تا زمانی که دکمه‌ی ماوس را رها نکرده اید، المان جبری مقدار قبلی را در خود نگه می دارد. بسته به این که کدام یک از دو شرط مذکور دیرتر اتفاق بیفتد تغییر مقدار المان انجام می گیرد.

برای درک عملکرد المان کنترل جبری در هر یک از حالات مذکور برنامه‌ی Mechanical Action of Booleans.vi را از مسیر LabVIEW\Examples\General\controls\booleans.llb باز و اجرا نمایید.




**شکل ۱۰-۸: صفحه‌ی پانل برنامه‌ی Mechanical Action of Booleans.vi**

حال کلید عمودی استفاده شده در تمرین قبل را در نظر بگیرید. این المان به صورت پیش فرض در وضعیت False یا off قرار دارد.

۱۵- وضعیت این کلید را به گونه ای تغییر دهید تا برای هر بار به اجرا در آوردن برنامه مجبور به تغییر حالت آن به موقعیت True نباشید.

برای انجام این عمل مراحل زیر را دنبال نمایید:

الف- کلید عمودی را به حالت True تغیر وضعیت دهید.

ب- منوی کرکره‌ای را بر روی کلید مذکور باز کنید و برای تغییر دادن حالت پیش‌فرض کلید به موقعیت True یا on، گزینه‌ی Data Operations >> Make Current Value Default را انتخاب نمایید.

ج- منوی کرکره‌ای را بر روی کلید مذکور باز نمایید و گزینه‌ی Mechanical Action >> Latch When Pressed را انتخاب کنید.

۱۶- اکنون برنامه را اجرا کنید. برای متوقف کردن عملیات اندازه‌گیری دما بر روی کلید Enable کلیک کنید. ملاحظه می‌کنید که کلید مذکور به حالت off تغییر وضعیت می‌دهد و پس از اینکه ترمینال شرطی وضعیت False را قرائت نمود، مجدداً به طور خودکار به حالت on باز می‌گردد.

در مورد متغیرهای محلی نمی‌توان از خواص و ویژگیهای حرکات مکانیکی کلیدهای جبری استفاده نمود. دلیل این امر را در فصل ۱۲ خاطر نشان خواهیم کرد.

## کنترل زمان اجرای برنامه

در تمرین قبل ملاحظه نمودید که پس از کلیک کردن بر روی دکمه‌ی Run، برنامه بلافاصله به اجرا در می‌آمد. در برخی موارد ممکن است قصد داشته باشید تا داده‌ها را در فواصل زمانی معینی نمونه برداری کنید. به عنوان مثال تصمیم دارید اندازه‌گیری دما یک مرتبه در هر ثانیه و یا یک مرتبه در هر دقیقه انجام گیرد. به کمک دستور Wait Until Next ms Multiply (Functions >> Time & Dialog) می‌توان زمان اجرای حلقه را کنترل نمود. حال مجدداً به برنامه‌ی تمرین ۱-۸ بازگردید.

۱۷- با اضافه نمودن بخش نشان داده شده در شکل ۱۱-۸ به حلقه‌ی While، برنامه را به گونه‌ای تغیر دهید که در هر ثانیه دو مرتبه دمای محیط را اندازه‌گیری نماید یا به عبارت دیگر در هر نیم ثانیه یک مرتبه دمای اتاق اندازه‌گیری شود.

شکل ۱۱-۸

Wait Until Next ms Multiple Function

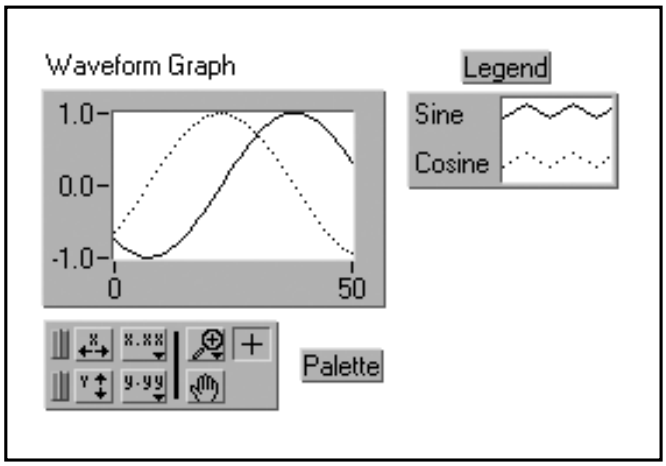
دستور Wait Until Next ms Multiple باعث می‌شود که هر تکرار حلقه پس از مدت زمان تعیین شده توسط کاربر انجام گیرد. در این تمرین، زمان مذکور نیم ثانیه است. به دلیل اینکه مقدار ورودی این دستور باید بر حسب میلی‌ثانیه اعمال گردد، بنابراین عدد 500 به عنوان ورودی دستور مذکور در نظر گرفته شده است.

۱۸- برنامه را به اجرا در آورید. حال زمان تأخیر تکرار حلقه را تغییر دهید و مجدداً برنامه را اجرا نمایید.

۱۹-این برنامه را با عنوان Temperature Monitor.vi در زیرفهرست LabVIEW\My Activity ذخیره نموده، سپس آن را ببندید.

## گراف

برخلاف نمودار که داده‌های جدید را به همراه داده‌های قبلی بر روی صفحه‌ی نمایش ترسیم می‌کند، در گراف آرایه‌های از پیش تولید شده یک‌جا رسم می گردند. به عبارت دیگر قابلیت به روز رساندن داده‌ها در گراف وجود ندارد. برای افزایش قابلیت انعطاف و کاربرد بیشتر این المان در نرم‌افزار LabVIEW، سه نوع گراف تعبیه شده است. این گراف‌ها عبارتند از: Waveform Graph، XY Graph و Intensity Graph. در این بخش به بحث در مورد گراف‌های Waveform و XY می‌پردازیم و بررسی گراف‌های Intensity را به بخش بعد موکول می کنیم. شکل ظاهری هر دو گراف Waveform و XY بر روی صفحه‌ی پانل یکسان بوده ولی عملکرد آنها متفاوت است. در شکل ۱۲-۸ نمونه‌ای از یک گراف را ملاحظه می کنید.



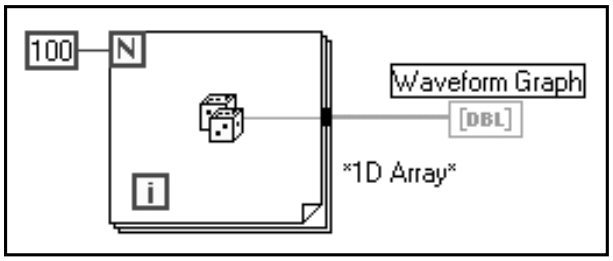

**شکل ۱۲-۸: یک گراف شکل موج چندگانه**

هر دو نوع گراف Waveform و XY در زیرپالت Controls >> Graph قرار دارند. گراف شکل موج یا Waveform Graph تنها برای ترسیم توابع یک‌به‌یک استفاده می شود. در این توابع نقاط متغیر X با فاصله‌ی مساوی از یکدیگر قرار دارند. این نوع گراف برای ترسیم داده هایی که در آنها نقاط به صورت متقارن پراکنده شده اند مناسب است. XY Graph یک گراف همه منظوره در مختصات دکارتی می‌باشد و برای ترسیم داده هایی که به صورت پراکنده قرار می‌گیرند و یا مبدأ زمانی[5] متفاوتی دارند ایده‌آل است. از این نوع گراف همچنین برای رسم داده هایی نظیر اشکال دایره‌ای که در آنها به هر مقدار X چند مقدار Y اختصاص می یابد استفاده می گردد. همان گونه که ذکر شد شکل ظاهری این دو نوع گراف یکسان می‌باشد اما نوع داده های ورودی آنها متفاوت است. بنابراین در هنگام کار با آنها مراقب باشید.

5- Time Base

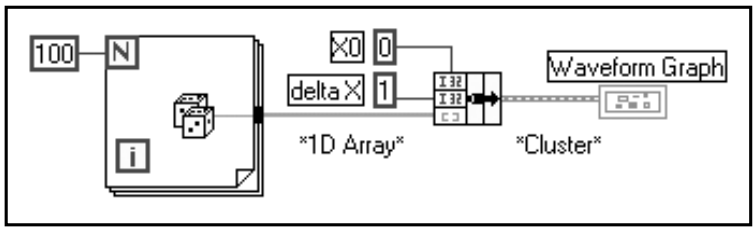
## گراف شکل موج منفرد

جهت ترسیم یک گراف شکل موج منفرد کافی است مطابق شکل ۱۳-۸ آرایه‌ی حاوی مقادیر Y را مستقیماً به ترمینال Waveform Graph سیم کشی کنید. در این روش فرض بر آن است که مقدار اولیه‌ی X و مقدار اختلاف دو مقدار X که با X0 و ΔX نمایش داده می شوند به ترتیب 0 و 1 هستند. مطابق شکل ۱۳-۸ در این حالت ترمینال گراف در صفحه‌ی نمودار بلوکی به صورت نشان دهنده‌ی آرایه ای یا Array Indicator ظاهر می گردد.




شکل ۱۳-۸

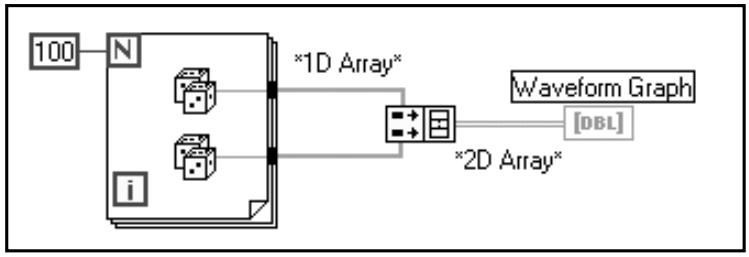
در برخی موارد لازم است که مبدأ زمان را در گراف تغییر دهید. به عنوان مثال قصد دارید در لحظه ای به غیر از X0 = 0 نمونه برداری را آغاز کنید یا اینکه ممکن است فواصل زمانی نمونه برداری کمتر یا بیشتر از یک واحد باشد. جهت تغییر مبدأ زمان و مقادیر X0 و ΔX در آرایه، این داده ها را در یک کلاستر دسته بندی نمایید سپس کلاستر را به گراف سیم کشی کنید. همان طور که در شکل ۱۴-۸ ملاحظه می کنید در این حالت ترمینال گراف به صورت نشان دهنده‌ی کلاستری یا Cluster Indicator ظاهر می شود.




شکل ۱۴-۸

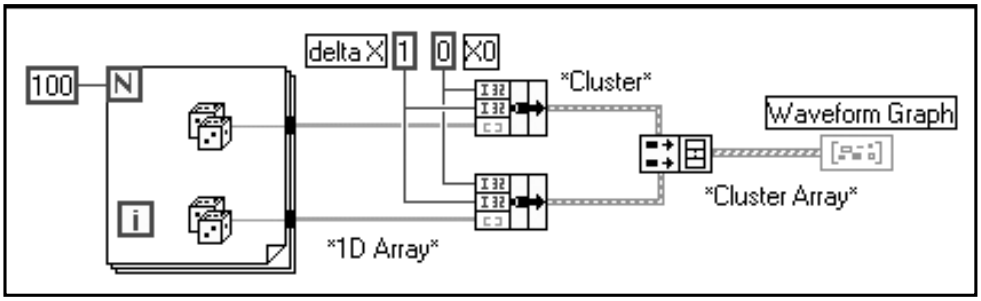
## گراف شکل موج چندگانه

با ایجاد یک آرایه‌ی دوبعدی از داده های استفاده شده در گراف های تک‌مقداری می توان بیش از یک گراف را بر روی صفحه ترسیم نمود. در این حالت توجه داشته باشید که شکل ظاهری ترمینال های گراف چگونه بسته به ساختار (آرایه، کلاستر و...) و نوع داده‌ی متصل شده به آنها (I16, DBL, ... ) تغییر می یابد.




شکل ۱۵-۸: روش ایجاد یک آرایه‌ی دوبعدی و ترسیم آن بر روی گراف شکل موج چندگانه

در شکل ۱۵-۸ فرض بر این است که مقادیر X0 و ΔX در هر دو آرایه به ترتیب 0 و1 می‌باشند. دستور Build Array از دو آرایه‌ی 1D، یک آرایه‌ی 2D ایجاد می‌کند. توجه داشته باشید که این آرایه‌ی دوبعدی یک ماتریس ۱۰۰ × ۲ است. به عبارت دیگر این آرایه دارای ۲ سطر است که هر سطر آن شامل ۱۰۰ ستون می‌باشد. داده‌های هر یک از سطرهای آرایه‌ی 2D به صورت پیش‌فرض بر روی یک گراف ترسیم می‌شوند. در صورتی که داده‌ها به صورت ستونی سازماندهی شده باشند باید قبل از ترسیم آرایه، آن را ترانهاده کنید. در اینجا منظور از آرایه‌ی ترانهاده، آرایه‌ای است که جای سطر و ستون آن عوض شده باشد. به عنوان مثال ترانهاده‌ی یک آرایه‌ی دوبعدی با ۳ سطر و ۱۰ ستون، یک آرایه‌ی دوبعدی با ۱۰ سطر و ۳ ستون است. در نرم‌افزار LabVIEW دستوری خاص برای انجام این عمل تعبیه شده است. برای به‌دست آوردن آرایه‌ی ترانهاده منوی کرکره‌ای را بر روی گراف باز نمایید و گزینه‌ی Transpose Array را انتخاب کنید. شایان ذکر است در صورتی که یک آرایه‌ی 2D به این گراف سیم‌کشی نشده باشد، گزینه‌ی مذکور به رنگ خاکستری ظاهر می‌شود و بیانگر غیر فعال بودن آن است. برای ترانهاده کردن یک آرایه‌ی 2D می‌توانید از دستور Transpose 2D Array (Functions >> Array) نیز استفاده کنید.



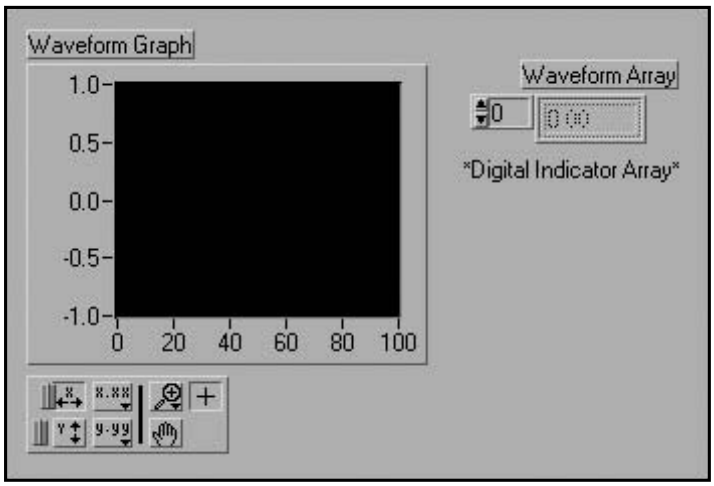

شکل ۱۶-۸

در شکل ۱۶-۸ ، دستور Build Array (Functions >> Array) از دو کلاستر ورودی خود یک آرایه ایجاد می‌کند. هر یک از کلاسترهای ورودی شامل یک آرایه و دو مقدار عددی است. نتیجه‌ی نهایی، آرایه‌ای از کلاسترهاست که می‌توان از آن به عنوان ورودی گراف در ترسیم منحنی استفاده نمود. در این حالت مقادیر X0 و ΔX همان مقادیر از پیش تعیین شده هستند. اما همواره می‌توانید مقادیر دیگری را نیز به‌کار برید.

## تمرین ۲-۸ : ترسیم شکل موج سینوسی به کمک Waveform Graph

در این تمرین قصد داریم تا با ایجاد یک برنامه آرایه‌ای حاوی نقاط موج سینوسی تولید نماییم، سپس آن را بر روی یک Waveform Graph ترسیم کنیم. در مرحله‌ی بعد با ایجاد تغیراتی در این برنامه یک گراف چندگانه رسم خواهیم کرد.

۱- یک برنامه‌ی جدید باز و صفحه‌ی پانل آن را مطابق شکل ۱۷-۸ ایجاد کنید.

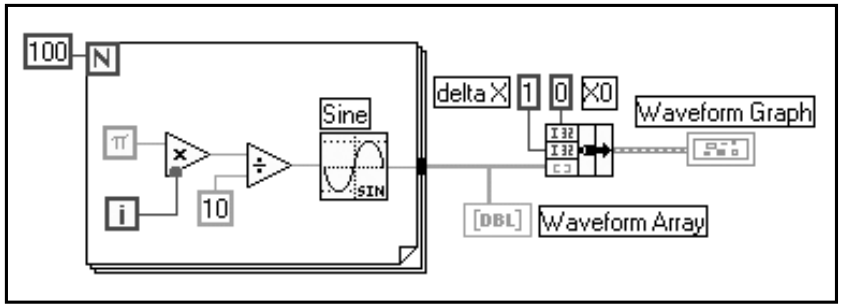



شکل ۱۷-۸

۲- یک پوسته‌ی آرایه یا Array Shell (Controls >> Array & Cluster) بر روی صفحه‌ی پانل قرار دهید. سپس آن را با برچسب Waveform Array نام گذاری کنید. برای به نمایش در آوردن محتویات آرایه‌ی مذکور، یک نشان‌دهنده‌ی دیجیتالی (Controls >> Numeric >> Digital Indicator) در داخل پنجره‌ی پوسته‌ی آرایه قرار دهید.

۳- یک Waveform Graph (Controls >> Graph) بر روی صفحه‌ی پانل قرار دهید و آن را با برچسب Waveform Graph نام گذاری کنید. سپس به کمک ابزار Positioning Tool بر روی یکی از گوشه های آن کلیک کنید و با حرکت دادن ماوس اندازه‌ی صفحه‌ی نمایش آن را افزایش دهید. منوی کرکره‌ای را بر روی گراف باز نمایید و با کلیک نمودن برروی گزینه‌ی Legend ✓ << Show، بخش Legend را مخفی کنید. منوی کرکره‌ای را بر روی گراف باز کنید و با کلیک کردن بر روی گزینه‌ی Autoscale Y ✓ << Y Scale سیستم درجه بندی خودکار را غیرفعال کنید. با استفاده از ابزار Labling Tool ابتدا مقادیر حداکثر و حداقل درجه بندی محور Y را انتخاب نموده، سپس مقادیر جدید را وارد کنید. به دلیل اینکه به محور Y مقادیر سینوسی نسبت داده می شود، بنابراین مقادیر حداکثر و حداقل درجه بندی این محور را به ترتیب 1.0 و 1.0- انتخاب کنید. در بخش بعد در مورد سیستم درجه بندی خودکار به بحث خواهیم پرداخت.

۴- صفحه‌ی نمودار بلوکی این برنامه را مطابق شکل ۱۸-۸ ایجاد نمایید.




شکل ۱۸-۸

Sine Function

تابع Sine (Functions >> Numeric >> Trigonometric) مقدار Sin (X) را محاسبه می‌کند و یک نقطه از شکل موج سینوسی را به دست می دهد. ورودی تابع Sine یک مقدار اسکالر است و برحسب رادیان می باشد. در این تمرین در هر تکرار حلقه مقدار X تغییر می‌کند و منحنی ترسیم شده به صورت یک موج سینوسی خواهد بود.

Pi Constant

عدد ثابت $\pi$ (Functions >> Numeric >> Additional Numeric Constants) آرگومان ورودی را برای دستور Sine فراهم می کند.

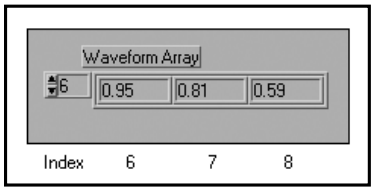
Bundle Function

دستور Bundle (Functions >> Cluster) اجزای گراف را در یک کلاستر دسته‌بندی می‌کند. این اجزا عبارتند از: مقدار X0 (عدد ثابت 0)، مقدار ΔX (عدد ثابت 1) و آرایه‌ی حاوی مقادیر Y (داده‌های Waveform Graph). برای افزایش اندازه‌ی این دستور، توسط ابزار Positioning Tool بر روی یکی از گوشه‌های آیکن دستور مذکور کلیک کنید و ماوس را حرکت دهید. در هر یک از تکرارهای حلقه‌ی For تنها یک نقطه از موج سینوسی تولید می‌شود و در آرایه‌ی Y که در چهارچوب حلقه ایجاد شده است ذخیره می گردد. پس از تکمیل اجرای حلقه، دستور Bundle مقادیر X0، ΔX و آرایه‌ی Y را برای ترسیم شکل موج سینوسی دسته‌بندی می کند.

۵- اکنون به صفحه‌ی پانل بازگردید و برنامه را اجرا کنید. در این لحظه تنها یک شکل موج را بر روی گراف ملاحظه می کنید.

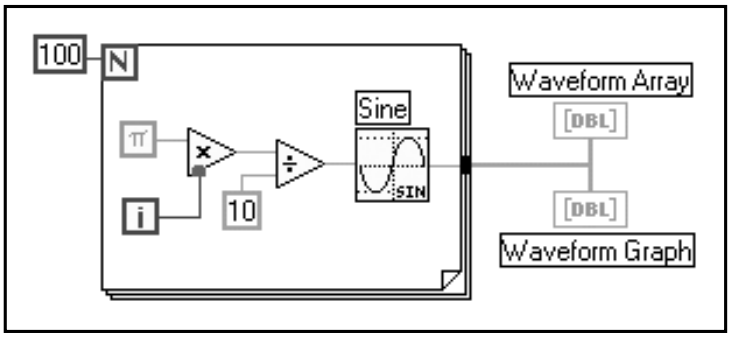
۶- حال مقادیر X0 و ΔX را به ترتیب به 20 و 0.5 تغییر دهید و مجدداً برنامه را اجرا نمایید. توجه کنید که در این حالت باز هم 100 نقطه بر روی صفحه‌ی گراف ترسیم می گردد. اما این بار داده‌ها از نقطه‌ی 20 آغاز می‌شوند و میزان تغییرات مقادیر X نیز 0.5 واحد است.

۷- ابزار Positioning Tool را بر روی گوشه‌ی پایینی سمت راست آرایه‌ی Waveform Array قرار دهید تا نشانگر ماوس به صورت صفحه‌ی شطرنجی یا Grid Cursur در آید. سپس ماوس را در جهت افقی یا عمودی حرکت دهید. در این حالت در صفحه‌ی نمایش‌دهنده، چندین مؤلفه با اندیس‌هایی که از سمت چپ به راست یا از بالا به پایین سیر صعودی دارند ظاهر می‌شوند. اولین المان از المان‌های مذکور متناظر با اندیس تعیین شده در صفحه‌ی نشان‌دهنده‌ی اندیس است. همان گونه که در شکل ۱۹-۸ ملاحظه می کنید در صفحه‌ی نمایش‌دهنده‌ی آرایه، سه مؤلفه با اندیس‌های 6، 7 و 8 دیده می‌شوند. اولین مؤلفه با اندیس 6 نزدیکترین المان به صفحه‌ی نمایش دهنده‌ی اندیس است. به یاد داشته باشید که با وارد کردن اندیس هر یک از مؤلفه‌های آرایه می توان مقدار مؤلفه‌ی متناظر با آن را مشاهده کرد. در صورتی که اندیس وارد شده بزرگتر از طول آرایه باشد صفحه‌ی نمایش تاریک خواهد شد.



شکل ۱۹-۸

در صفحه‌ی نمودار بلوکی قبلی مقادیر X0 و ΔX را به ترتیب 0 و 1 انتخاب نمودید. در این حالت می توانید آرایه را مستقیماً به ترمینال گراف سیم کشی کنید و از مقادیر پیش فرض X0 و ΔX استفاده نمایید. این مطلب در شکل ۲۰-۸ نشان داده شده است.



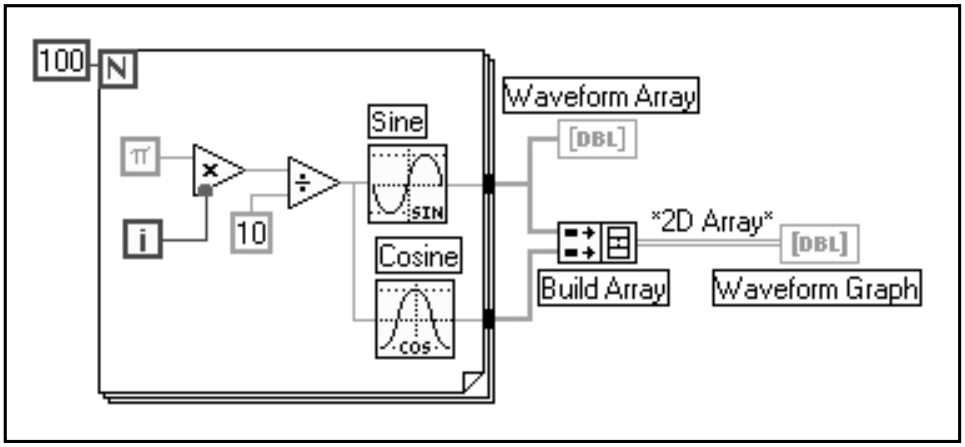

شکل ۲۰-۸

۸- اکنون به صفحه‌ی نمودار بلوکی بازگردید و دستور Bundle و اعداد ثابت سیم کشی شده به آن را حذف کنید. سپس گزینه‌ی Edit >> Remove Bad Wires را انتخاب کنید و سیمهای اضافی را حذف کنید. حال سیم کشی را مطابق شکل ۲۰-۸ تکمیل نمایید.

۹- برنامه را اجرا کنید. توجه داشته باشیدکه این بار گراف شکل موج با مقادیر پیش فرض X0 و ΔX یعنی مقادیر 0 و 1 ترسیم می شود. در این حالت برای ایجاد صفحه‌ی نمودار بلوکی زحمت کمتری متحمل شده اید.

با سیم کشی داده های حاصل از گراف های منفرد به دستور Bundle Array و سپس با سیم کشی به ترمینال گراف می توانید یک گراف چندگانه ایجاد کنید. اگرچه امکان ایجاد یک آرایه از آرایه های دیگر وجود ندارد، اما می توانید آرایه های ورودی را به عنوان سطرهای یک آرایه‌ی 2D در نظر گرفته و یک آرایه‌ی دوبعدی ایجاد کنید.

۱۰- صفحه‌ی نمودار بلوکی این برنامه را مطابق شکل ۲۱-۸ تغییر دهید.




شکل ۲۱-۸

Build Array Function

دستور Bundle Array (Functions >> Array) داده های مناسب را جهت ترسیم دو آرایه بر روی یک گراف ایجاد می کند. با کلیک کردن توسط ابزار Positioning Tool بر روی یکی از گوشه های این دستور و حرکت دادن ماوس، اندازه‌ی آن را به گونه ای تغییر دهید تا ظرفیت دریافت دو ورودی را داشته باشد. در اینجا لازم است برای دستیابی به یک آرایه‌ی 2D در دستور Bundle Array به جای ورودیهای آرایه ای از ورودیهای مؤلفه ای استفاده کنید. در صورت استفاده از ورودیهای آرایه ای، این دستور با پیوند دادن آرایه‌ی دوم به انتهای آرایه‌ی اول، یک آرایه‌ی یک بعدی طولانی ایجاد می کند.

Cosine Function

دستور Cosine (Functions >> Numeric >> Trigonometric) مقدار Cos (X) را محاسبه می کند و به ازای هر مقدار X یک نقطه از شکل موج کسینوسی را به دست می دهد. ورودی این دستور یک عدد اسکالر است و برحسب رادیان می باشد. در این تمرین در هر تکرار حلقه، مقدار ورودی X تغییر می کند و منحنی ترسیم شده به صورت یک موج کسینوسی خواهد بود.

۱۱- حال به صفحه‌ی پانل بازگردید و برنامه را اجرا کنید. توجه داشته باشید که هر دو شکل موج سینوسی و کسینوسی بر روی یک گراف ترسیم می‌شوند و مقادیر X0 و ΔX به صورت پیش فرض یعنی 0 و 1 برای هر دو مجموعه‌ی داده ها در نظر گرفته شده است.

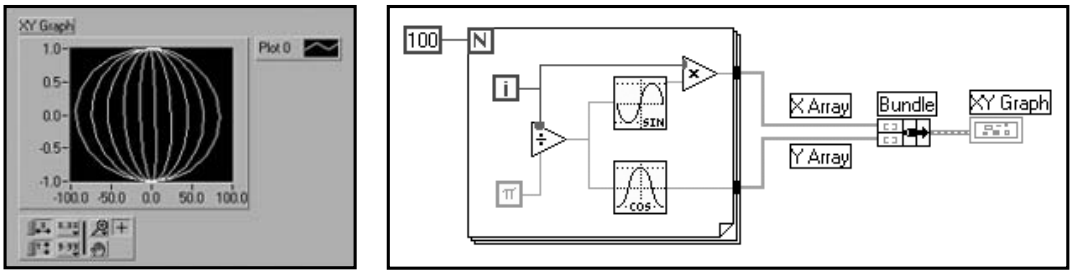
۱۲- اکنون منوی کرکره ای را بر روی گراف باز نموده و گزینه‌ی Transpose Array را انتخاب کنید. ملاحظه می کنید که در صورت تعویض جای سطر و ستون و یا به عبارت دیگر ترانهاده کردن آرایه، شکل ظاهری گراف نیز تغییر می کند. با انتخاب مجدد گزینه‌ی مذکور به حالت اولیه بازگردید.

۱۳- این برنامه را با عنوان Graph Sine Array.vi در زیرفهرست LabVIEW\My Activity ذخیره کنید، سپس آن را ببندید.

## گراف XY

تا اینجا تنها Waveform Graph را مورد بررسی قرار دادیم. همان گونه که عنوان شد این گراف برای ترسیم منحنیهایی که به صورت یکسان و متقارن نمونه برداری شده اند به کار برده می شود. حال فرض کنید که فواصل نمونه برداری، نامنظم هستند و یا اینکه قصد دارید یک تابع ریاضی را که در آن به هر یک از مقادیر X چند مقدار Y اختصاص داده شده است ترسیم کنید. معمولاً در این موارد، هر یک از نقاط با استفاده از مختصات (X,Y) و به صورت زوج مرتب مشخص می شوند. جهت ترسیم این نوع منحنیها از XY Graph استفاده می شود.

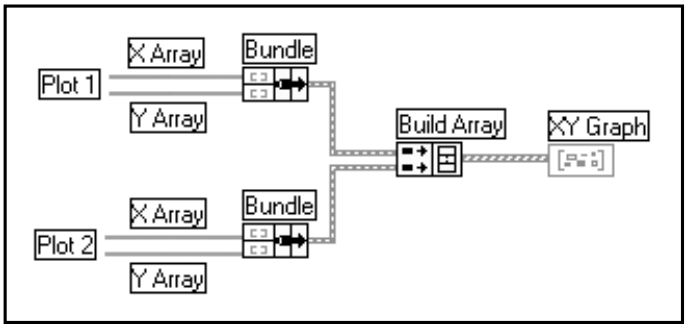
برخلاف Waveform Graph در این نوع گراف، داده ها از انواع متفاوتی هستند. در شکل ۲۲-۸ صفحه‌ی پانل یک گراف منفرد XY را به همراه صفحه‌ی نمودار بلوکی متناظر با آن ملاحظه می کنید.




**شکل ۲۲-۸: صفحات پانل و نمودار بلوکی در یک گراف منفرد XY**

همان گونه که در شکل ۲۲-۸ نشان داده شده است ورودیهای گراف XY، دو آرایه‌ی دسته بندی شده‌ی X (ورودی بالایی) و Y (ورودی پایینی) می باشند. دستور Bundle (Functions >> Cluster) دو آرایه‌ی X و Y را در یک کلاستر دسته بندی می کند. در مرحله‌ی بعد کلاستر ایجاد شده به گراف XY سیم کشی می شود.

در این حالت ترمینال گراف XY به صورت نشان دهنده‌ی کلاستری یا Cluster Indicator ظاهر می شود. در یک گراف چندگانه‌ی XY می توان از کلاسترهای مقادیر X و Y که در گراف XY منفرد استفاده شده اند یک آرایه ایجاد کرد. این مطلب در شکل ۲۳-۸ نشان داده شده است.



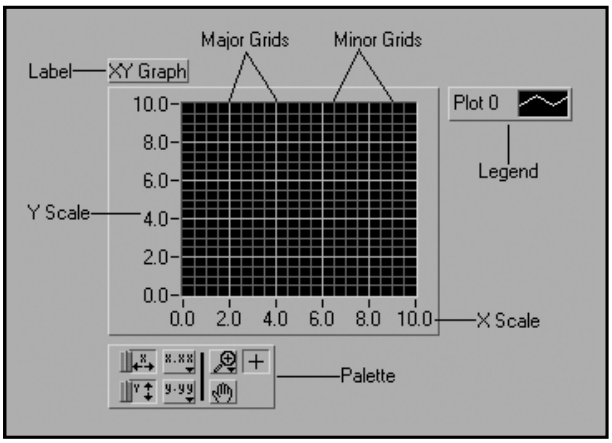

**شکل ۲۳-۸**

در هنگام دسته بندی داده ها برای ترسیم یک گراف مراقب باشید که دو دستور Bundle و Bundle Array را به جای یکدیگر استفاده نکنید. توجه کنید که به کدام یک نیاز دارید.

برای درک عملکرد گراف‌ها، برنامه‌های Waveform Graph.vi و X vs Y Graph.vi را از مسیر LabVIEW\Exercise\Ch 8.llb اجرا کنید.

## اجزای نمودار و گراف

صفحات نمایشی نمودار و گراف قابلیتهای خاصی دارند که می توان برای سفارشی نمودن نحوه‌ی نمایش مقادیر از آنها استفاده نمود. این بخش به بررسی نحوه‌ی به کارگیری قابلیتهای مذکور اختصاص داده شده است.




**شکل ۲۴-۸: اجزای نمودار و گراف**

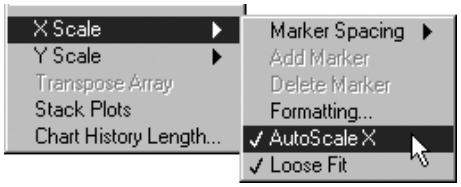
## تغییر درجه‌بندی

برای به نمایش در آوردن تمامی نقاط با بالاترین درجه‌ی وضوح[6] در نمودار و گراف، می توانید درجه‌بندی محورهای عمودی و افقی را به طور خودکار تغییر دهید. با انتخاب یکی از دو گزینه‌ی AutoScale X و AutoScale Y از زیرمنوهای X Scale و Y Scale << Data Operations در منوی کرکره‌ای صفحه‌ی نمایش نمودار یا گراف، می توان قابلیت درجه‌بندی خودکار[7] را فعال یا غیر فعال ساخت.

همچنین می توانید به کمک بخش Palette، درجه‌بندی خودکار را کنترل کنید. سیستم درجه‌بندی خودکار به صورت پیش‌فرض در گراف‌ها فعال و در نمودارها غیر فعال در نظر گرفته شده است. شایان ذکر است که استفاده از این قابلیت ممکن است سرعت به روز رساندن داده‌ها را در نمودار یا گراف کاهش دهد. البته این مطلب به نوع و سرعت رایانه و همچنین کارت گرافیکی استفاده شده در رایانه بستگی دارد. اگر قصد ندارید از این ویژگی استفاده کنید، با استفاده از ابزارهای Operating Tool یا Labeling Tool یک مقدار جدید را به عنوان مقیاس انتخاب کنید و مستقیماً درجه‌بندی محورهای عمودی یا افقی را تغییر دهید. واضح است که در این حالت سیستم درجه‌بندی خودکار غیر فعال می گردد.

## منوهای X Scale و Y Scale

در هر یک از منوهای X Scale و Y Scale یک زیرمنو به صورت زیر وجود دارد.

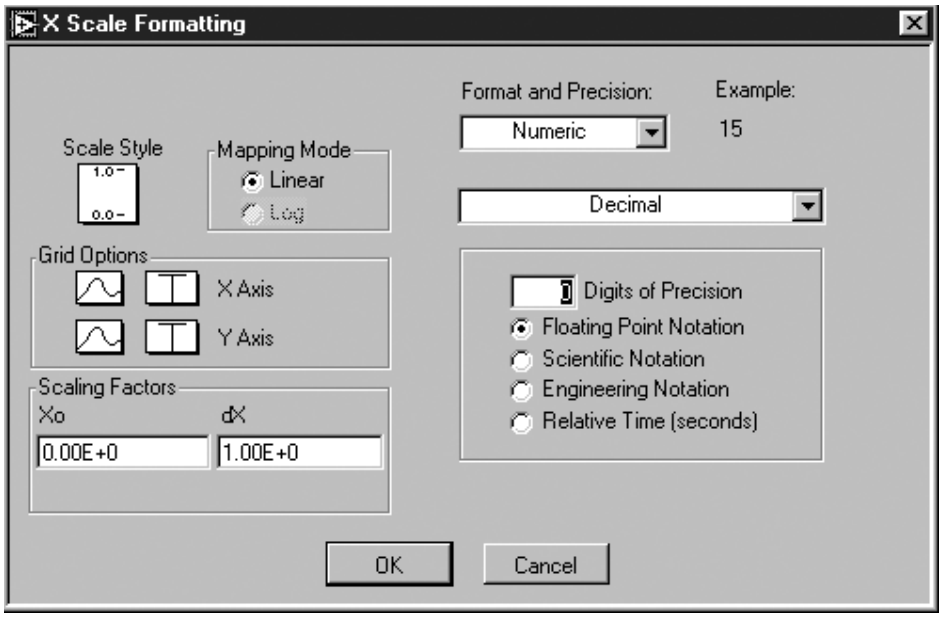



**شکل ۲۵-۸: گزینه‌های موجود در منوی X Scale**

6- Resolution
7- Auto-Scaling

گزینه های AutoScale X و AutoScale Y جهت فعال یا غیر فعال کردن قابلیت درجه‌بندی خودکار مورد استفاده قرار می گیرند. در صورت استفاده از ویژگی درجه‌بندی خودکار، درجه‌بندی محورها به گونه‌ای تنظیم می شود که داده‌ها به بهترین صورت بر روی محورها قرار گیرند. در صورتی که قصد دارید درجه بندی محورها به اعداد مناسبی گرد شوند، گزینه‌ی Loose Fit را از این منو انتخاب کنید. در صورت انتخاب این گزینه، درجه‌بندی محورها به مضاربی از میزان افزایش درجات گرد می شوند. به عنوان مثال اگر میزان افزایش درجه‌بندی محورها 5 واحد باشد، آنگاه مقادیر حداکثر و حداقل درجه‌بندی محور به مضاربی از عدد 5 گرد می شوند.

با انتخاب گزینه‌ی ...Formatting یک پنجره‌ی محاوره‌ای مطابق شکل ۲۶-۸ باز می شود که در آن می توانید پارامترها و گزینه های لازم در تعیین فرمت درجه بندی محور را تغییر دهید.




**شکل ۲۶-۸: پنجره‌ی محاوره‌ای Scale Formatting**

در ادامه به بررسی گزینه های موجود در پنجره‌ی محاوره ای Formatting می پردازیم:

- به کمک زیرگزینه‌های موجود در منوی Scale Style می توانید علائم موجود بر روی محورها را تغییر دهید. این علائم، درجه‌بندی محورها را مشخص می کنند. با انتخاب هر یک از زیرگزینه‌ها تغییر فرمت درجه‌بندی را مشاهده کنید.
- در بخش Mapping Mode نیز می توان درجه بندی محورها را به یکی از دو حالت خطی یا لگاریتمی انتخاب نمود. در قسمت Grid Options می توانید نحوه‌ی خط کشی و به عبارت دیگر شطرنجی بودن صفحه را انتخاب نمایید. در این بخش می توان رنگ خطوط را نیز انتخاب نمود. برای مشاهده‌ی گزینه ها و رنگهای مختلف، بر روی دکمه های موجود در این بخش کلیک کنید.
- در بخش Scaling Factors می توان مقادیر X0 و dX را تعیین نمود.
- در منوی حلقوی Format and Precision می توانید یکی از دو فرمت Numeric یا Time & Date را انتخاب کنید. در صورت انتخاب فرمت Numeric می توانید تعداد ارقام اعشار و همچنین طرز نمایش اعداد مندرج در

محورها را به یکی از صورتهای اعشاری، علمی یا مهندسی انتخاب نمایید. به کمک زیرگزینه‌های موجود در این بخش می توانید اعداد را در یکی از مبناهای دو، هشت، ده یا شانزده نمایش دهید. در صورت انتخاب فرمت Time & Date می توان نحوه‌ی به نمایش در آمدن زمان و تاریخ را تعیین نمود.

## به‌کارگیری بخش Legend

واژه‌ی Legend در اصل به معنای راهنمای نقشه است. این واژه به علائمی اطلاق می شود که در کنار نقشه به عنوان راهنما در نظر گرفته شده‌اند.

در بخش Legend می توانید برای هر نمودار یا گراف برچسب تعیین نموده، رنگ و فرم خطوط ترسیم شده و حالت نقاط مندرج در منحنیها را مشخص کنید. تا زمانی که برای نمودار یا گراف، یک فرم دلخواه انتخاب نکرده‌اید، گزینه های پیش فرض آنها فعال هستند.

به کمک زیرمنوی Show در منوی کرکره‌ای نمودار یا گراف، می توان بخش Legend را ظاهر یا مخفی ساخت. همچنین در این بخش می توانید به هر منحنی یک نام اختصاص دهید. در شکل ۲۷-۸ بخش Legend را ملاحظه می کنید.

**شکل ۲۷-۸: بخش Legend**

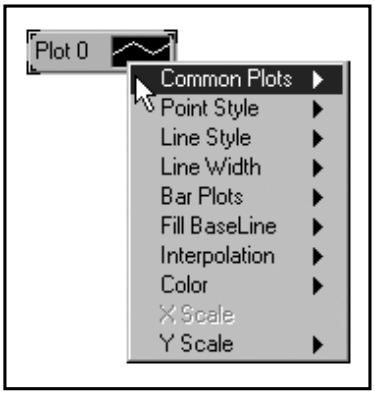
پس از ظاهر شدن بخش Legend تنها یک منحنی در این جعبه نشان داده می شود. برای به نمایش در آوردن منحنیهای دیگر، توسط ابزار Positioning Tool بر روی یکی از گوشه های جعبه‌ی Legend کلیک کنید و در حالی که دکمه‌ی ماوس را فشار داده اید، ماوس را حرکت داده، بدین ترتیب اندازه‌ی جعبه را افزایش دهید. پس از تنظیم و تعیین مشخصات مطلوب در بخش Legend، منحنی مورد نظر صرف نظر از وجود یا فقدان بخش Legend بر روی صفحه، تمام تنظیمات را در خود نگه می دارد. در صورتی که در بخش Legend بیش از یک منحنی در نمودار یا گراف ترسیم شده باشد، سایر منحنیها با مشخصات پیش فرض به نمایش در می آیند.

هنگامی که نمودار یا گراف را بر روی صفحه‌ی پانل حرکت می دهید، بخش Legend نیز به همراه آن جابه‌جا می شود. با حرکت دادن بخش Legend می‌توانید موقعیت و محل آن را نسبت به نمودار یا گراف تغییر دهید. برای داشتن فضای بیشتر جهت تعیین برچسب منحنیها اندازه‌ی جعبه‌ی Legend را از سمت چپ تغییر دهید و در صورتی که قصد دارید بخش بیشتری از منحنی را در جعبه‌ی Legend به نمایش در آورید، اندازه‌ی این جعبه را از سمت راست افزایش دهید.

همان گونه که در شکل ۲۷-۸ ملاحظه نمودید منحنیهای موجود در بخش Legend به صورت پیش فرض دارای برچسبهایی هستند که از شماره‌ی 0 شروع می شوند. می توانید این برچسبها را به همان

روش ذکر شده در مورد تغییر برچسب المان های دیگر تغییر دهید. برای تغییر دادن برچسب کافی است با استفاده از ابزار Labeling Tool برچسب مورد نظر را تایپ نموده، سپس این برچسب را به یکی از روشهای ذکر شده وارد کنید.

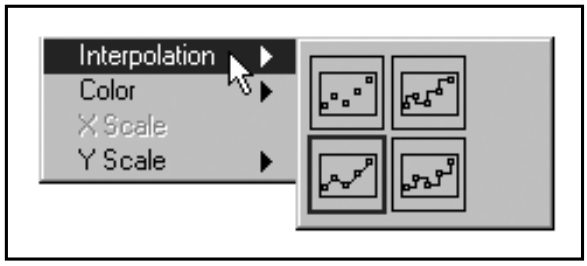
برای هر منحنی یک منوی خاص در نظر گرفته شده است که به کمک گزینه های موجود در آن می توانید شکل، فرم و رنگ خطوط و نقاط موجود در منحنی را تغییر دهید. برای دستیابی به منوی مذکور مطابق شکل ۲۸-۸، توسط ابزار Operating Tool بر روی منحنی مورد نظر در بخش Legend کلیک کنید. در ادامه به بررسی گزینه های موجود در منوی مذکور می پردازیم.




شکل ۲۸-۸

## گزینه های موجود در منوی کرکره‌ای بر روی بخش Legend

- به کمک گزینه های موجود در زیرمنوی Common Plots می توانید منحنی مورد نظر را به یکی از ۶ حالت عمومی نمایش دهید. گزینه های موجود در این زیرمنو به شما اجازه می دهند تا شکل ظاهری نقاط و خطوط منحنی را تغییر دهید. برای کسب اطلاعات بیشتر در مورد این گزینه ها به تمرین ۶-۸ مراجعه نمایید.
- گزینه های Point Style، Line Style و Line Width برای تعیین مشخصات نقاط و خطوط منحنی به کار می روند. به عنوان مثال گزینه‌ی Line Width شامل زیرگزینه‌ی خط باریک یا hair Line می باشد که ضخامت این خط به اندازه‌ی یک نقطه بوده، با چاپگرهای مختلف سازگار است.
- گزینه‌ی Bar Plots به کاربر اجازه می دهد تا عرض منحنیهای میله‌ای را از لحاظ افقی و عمودی تعیین نماید.
- گزینه‌ی Interpolation نحوه‌ی ترسیم خطوط را بین نقاط موجود در منحنی تعیین می کند. در زیرمنوی این گزینه مطابق شکل ۲۹-۸ موارد دیگری نیز وجود دارد. در صورت انتخاب اولین گزینه، هیچ خطی ترسیم نمی‌شود. در این حالت تنها تعدادی نقطه را بر روی صفحه‌ی منحنی مشاهده می کنید. در صورت انتخاب گزینه‌ی پایینی سمت چپ، یک خط مستقیم بین نقاط ترسیم می شود. دو گزینه‌ی دیگر، نقاط را با خطوطی که زاویه‌ی قائمه دارند به یکدیگر متصل می کنند. این دو گزینه برای ترسیم هیستوگرام مناسب بوده، به گزینه های پله ای معروف اند. گزینه‌ی پله ای بالایی ابتدا در جهت محور Y پیش می‌رود در حالی که گزینه‌ی پله ای پایینی ابتدا به ترسیم خط در جهت محور X می پردازد.



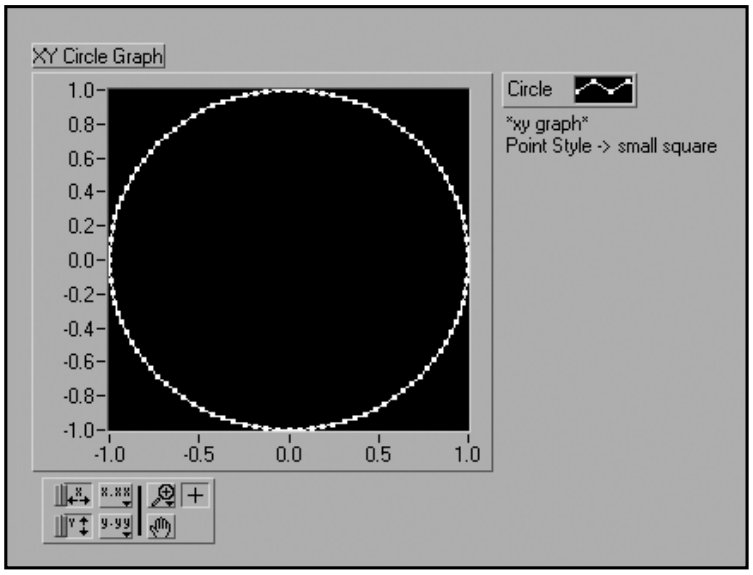

**شکل ۲۹-۸: گزینه‌های موجود در پالت Interpolation**

گزینه‌ی Color به کاربر کمک می‌کند تا پالت رنگ را ظاهر ساخته، سپس رنگ مورد نظر را جهت رنگ‌آمیزی خطوط منحنی انتخاب نمایید. ایجاد تغییر در رنگ خطوط منحنی در حین اجرای برنامه نیز امکان‌پذیر است.

## تمرین ۳-۸: ترسیم یک دایره به کمک XY Graph

در این تمرین قصد داریم تا با ایجاد یک برنامه و با کمک آرایه‌های مستقل X و Y، یک دایره بر روی گراف XY ترسیم کنیم.

۱- یک برنامه‌ی جدید باز نموده، صفحه‌ی پانل آن را مطابق شکل ۳۰-۸ ایجاد کنید. البته دایره‌ی نشان داده شده در این شکل را در مراحل بعدی بر روی گراف XY به نمایش در می‌آورید.




**شکل ۳۰-۸: صفحه‌ی پانل در برنامه‌ی Graph Circle.vi**

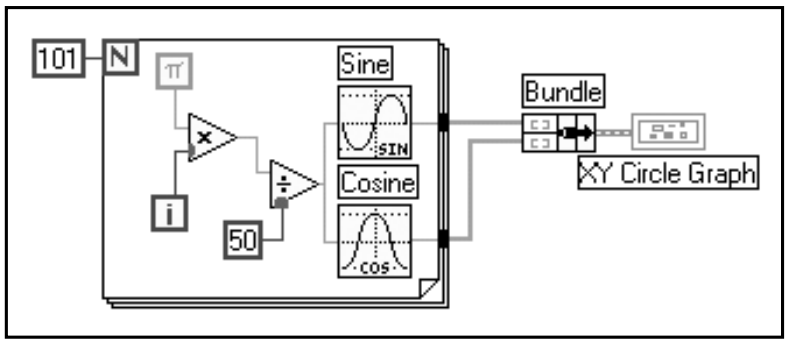
۲- یک XY Graph (Controls >> Graph) بر روی صفحه‌ی پانل قرار دهید و آن را با برچسب XY Circle Graph نام گذاری کنید.

۳- به کمک ابزار Positioning Tool بر روی یکی از گوشه‌های صفحه‌ی گراف کلیک کنید و با حرکت دادن ماوس اندازه‌ی آن را افزایش دهید. سعی کنید تا سطح صفحه‌ی گراف را تا حد امکان به صورت مربع در آورید.

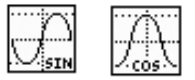
با پایین نگه داشتن کلید <Shift> و حرکت دادن ماوس به صورت قطری، نسبت طول و عرض صفحه را ثابت نگه دارید.

۴- منوی کرکره‌ای را بر روی گراف باز نموده، گزینه‌ی Legend >> Show را انتخاب کنید. اندازه‌ی جعبه‌ی Legend را از سمت چپ تغییر دهید و با استفاده از ابزار Labeling Tool برچسب Circle را برای منحنی تعیین کنید. حال منوی کرکره‌ای را بر روی جعبه‌ی Legend باز نموده، گزینه‌ی small square را از زیرمنوی Point Style انتخاب کنید. سپس برای رنگ‌آمیزی منحنی، به دلخواه یک رنگ از زیرمنوی Color انتخاب نمایید.

۵- اکنون صفحه‌ی نمودار بلوکی این برنامه را مطابق شکل ۳۱-۸ ایجاد کنید.




**شکل ۳۱-۸: صفحه‌ی نمودار بلوکی در برنامه‌ی Graph Circle.vi**



Sine Function Cosine Function

دستورهای Sine و Cosine (Controls >> Numeric >> Trigonometric) به ترتیب مقادیر سینوس و کسینوس اعداد ورودی را محاسبه می‌کنند. در این تمرین این دو دستور را در یک حلقه‌ی For قرار داده، سعی می‌کنیم تا آرایه‌هایی از یک سیکل کامل در هر دو موج سینوسی و کسینوسی به دست آوریم.

Bundle Function

دستور Bundle (Controls >> Cluster) آرایه‌ی سینوس (مقادیر X) را به همراه آرایه‌ی کسینوس (مقادیر Y) دسته‌بندی می‌کند تا بتوان آرایه‌ی سینوس را در مقابل آرایه‌ی کسینوس ترسیم نمود.

Pi Constant

عدد ثابت π (Functions >> Nnmeric >> Additional Numeric Constants) مقدار زاویه‌ی ورودی را برحسب رادیان برای دو دستور Sine و Cosine فراهم می‌سازد.

با استفاده از دستور Bundle می‌توانید یک سیکل از آرایه‌ی سینوس را در مقابل یک سیکل از آرایه‌ی کسینوس ترسیم کنید. واضح است که این عمل منجر به ترسیم یک دایره می‌شود.

۶- حال به صفحه‌ی پانل بازگردید و برنامه را اجرا کنید.

این برنامه را با عنوان Graph Circle.vi در زیرفهرست LabVIEW\My Activity ذخیره نموده، سپس آن را ببندید.

## به‌کارگیری بخش Palette

بخش Palette شامل جعبه‌ای کوچک است که به کمک آن می‌توانید چندین مشخصه‌ی منحنی را حتی در حین اجرای برنامه تغییر دهید. به عنوان مثال می‌توانید صفحه‌ی نمایش نمودار یا گراف را پاک کرده، یا محورهای X و Y را درجه‌بندی کنید. در هر لحظه می‌توانید فرمت نمایش اعداد مندرج بر روی محورها را عوض کنید. به یاد داشته باشید که انجام این عمل به کمک گزینه‌های X Scale و Y Scale نیز امکان‌پذیر است. به کمک گزینه‌های بخش Palette می‌توان عملیات بزرگ‌نمایی را در ناحیه‌ای از صفحه‌ی نمایش گراف یا نمودار انجام داد. برای ظاهر ساختن Palette، منوی کرکره‌ای را بر روی نمودار یا گراف باز نموده، سپس گزینه‌ی Show >> Palette را انتخاب کنید. در شکل ۳۲-۸ بخش Palette را ملاحظه می‌کنید.

**شکل ۳۲-۸: بخش Palette**

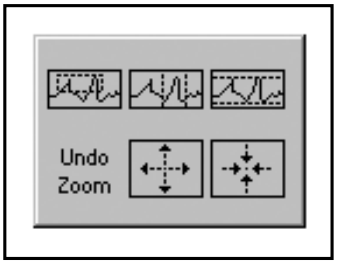
در ادامه به بررسی موارد موجود در این جعبه می‌پردازیم:

در صورتی که بر روی دکمه‌های Autoscale X Data یا Autoscale Y Data کلیک کنید، LabVIEW به صورت خودکار به ترتیب اطلاعات و داده‌های محورهای X یا Y را درجه‌بندی می‌کند. اگر قصد دارید تا درجه‌بندی تعیین شده بر روی هر یک از محورها به طور دائم به همین صورت باقی بماند بر روی دکمه‌ی Lock Switch متناظر با هر یک از محورها کلیک کنید. همان طور که در جعبه‌ی Palette ملاحظه می‌کنید برای هر یک از محورها یک دکمه‌ی Lock Switch درنظر گرفته شده است. این کلید در وضعیت روشن و خاموش به ترتیب در سمت راست و چپ قرار می‌گیرد.

دو دکمه‌ی نشان داده شده در سمت راست صفحه به کاربر اجازه می‌دهند تا در زمان اجرای برنامه و ترسیم گراف، نحوه‌ی درجه‌بندی دو محور X و Y را کنترل کند.

سه دکمه‌ی نشان داده شده در سمت راست صفحه به کاربر اجازه می‌دهند تا مد عملیاتی را تغییر دهد. همان‌طور که می‌دانید در حالت عادی در مد عملیاتی استاندارد یا Standard Operate Mode قرار دارید. بدین معنی که می‌توانید بر روی نشانگرهای گراف کلیک نموده، آنها را جابه‌جا کنید. در صورتی که دکمه‌ی Pan Button را فشار دهید قادر خواهید بود تا داده‌ها و اطلاعات قابل‌رؤیت را با حرکت دادن بخشهایی از گراف به کمک نشانگر Pan Cursor بلغزانید.

پس از کلیک کردن بر روی دکمه‌ی Zoom Button یک پنجره‌ی محاوره‌ای باز می‌شود و به کاربر اجازه می‌دهد تا چندین حالت بزرگ‌نمایی را انتخاب کند. این حالات در شکل ۳۳-۸ نشان داده شده‌اند.




**شکل ۳۳-۸: گزینه‌های مختلف برای بزرگ‌نمایی**

در ادامه‌ی بحث، گزینه‌ها و روشهای مختلف بزرگ‌نمایی را بررسی می‌کنیم:

Zoom by Rectangle

حالت Zoom by Rectangle. در این روش بزرگ‌نمایی، نشانگر را به گونه‌ای حرکت دهید تا در اطراف ناحیه‌ای که قصد بزرگ‌نمایی آن را دارید یک مستطیل ایجاد گردد. پس از رها کردن دکمه‌ی ماوس، درجه‌بندی محورها به گونه‌ای تغییر می‌یابد که تنها ناحیه‌ی انتخاب شده به نمایش در آید.

Zoom by Rectangle (restricted to X data)

حالت Zoom by Rectangle in X. در این روش، عملیات بزرگ‌نمایی تنها به داده‌های موجود بر روی محور X محدود می‌شود. به عبارت دیگر درجه‌بندی محور Y ثابت و بدون تغییر باقی می‌ماند.

Zoom by Rectangle (restricted to Y data)

حالت Zoom by Rectangle in Y. در این روش، عملیات بزرگ‌نمایی تنها به داده‌های موجود بر روی محور Y محدود می‌شود. به عبارت دیگر درجه‌بندی محور X ثابت و بدون تغییر باقی می‌ماند.

Undo Last Zoom

حالت Undo last Zoom. همان گونه که از نام این گزینه استنباط می‌گردد با کلیک کردن بر روی این دکمه، درجه‌بندی محورها به حالت قبلی تغییر می‌یابد. به عبارت دیگر آخرین عملیات بزرگ‌نمایی را در مورد محورها ری‌ست می‌کند.

Zoom In About a Point

حالت Zoom In About a Point. در صورتی که بر روی این دکمه کلیک کنید و سپس دکمه‌ی ماوس را بر روی یک نقطه‌ی خاص از گراف فشار دهید تا زمانی که دکمه‌ی ماوس را رها نکرده‌اید، عملیات بزرگ‌نمایی در اطراف نقطه‌ی مذکور به طور پیوسته انجام می‌گیرد.

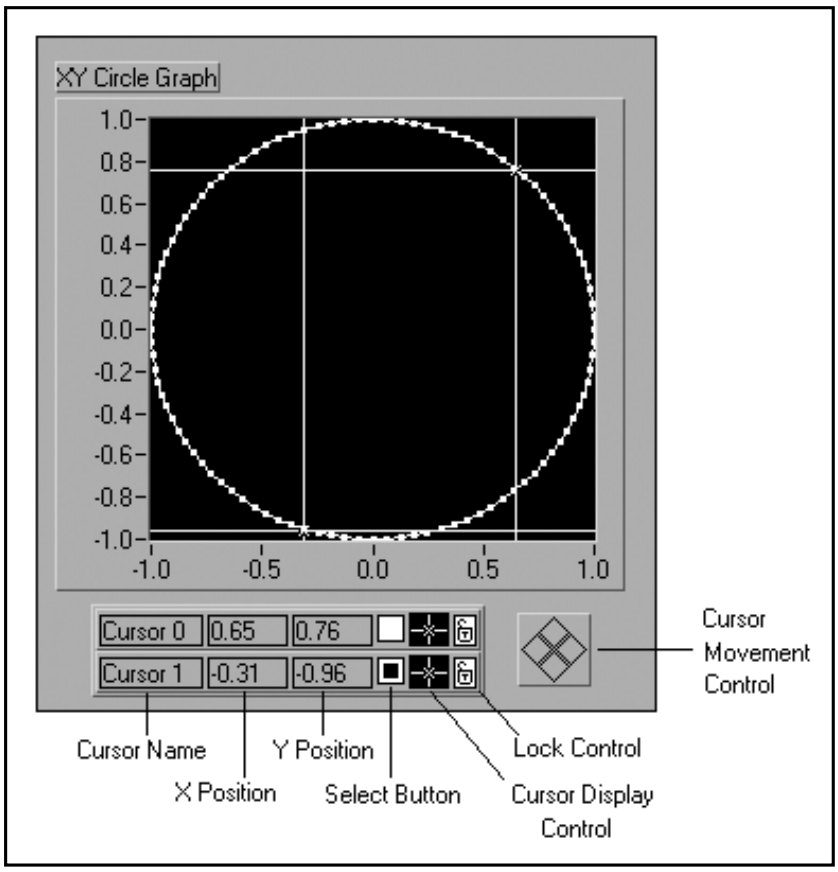
Zoom Out About a Point

حالت Zoom Out About a Point. در صورتی که بر روی این دکمه کلیک کنید و سپس دکمه‌ی ماوس را بر روی یک نقطه‌ی خاص از گراف فشار دهید، تا زمانی که دکمه‌ی ماوس را رها نکرده‌اید، عملیات کوچک‌نمایی در اطراف نقطه‌ی مذکور به طور پیوسته انجام می‌گیرد.

در دو حالت اخیر یعنی Zoom Out About a Point و Zoom In با پایین نگه داشتن کلید <Shift> و کلیک نمودن بر روی ماوس، هر یک از اعمال مذکور به صورت معکوس انجام می‌گیرند.

## نشانگرهای گراف

در نرم افزار LabVIEW برای نشان دادن و مشخص کردن محل نقاط بر روی منحنی چندین نشانگر یا Cursor تعبیه شده است. شایان ذکر است که در نمودارها هیچ گونه نشانگری وجود ندارد.




**شکل ۳۴-۸: پالت Cursor**

جهت دستیابی به پالت Cursor، منوی کرکره‌ای را بر روی گراف باز نموده، سپس گزینه‌ی Cursor Display << Show را انتخاب کنید. هنگامی که این پالت برای اولین مرتبه ظاهر می گردد به رنگ خاکستری دیده می شود و بدین معنی است که در حال حاضر پالت Cursor غیر فعال می باشد. برای فعال کردن نشانگرها بر روی دکمه‌ی Select Button کلیک کنید یا به کمک ابزار Labeling Tool در ناحیه‌ی در نظرگرفته شده برای نام نشانگر (Cursor Name) به دلخواه یک نام را برای نشانگر مورد نظر تایپ کنید. در فصل ۱۲ خواهید دید که چگونه می توان نشانگرها را به صورت دستی یا از طریق برنامه و به کمک گره‌ی خصوصیت حرکت داد.

برای حرکت دادن نشانگر بر روی گراف، از ابزار Operating Tool استفاده کنید. در صورتی که نشانگر ماوس را بر روی محل تقاطع خطوط عمودی و افقی ببرید و آن را جابه‌جا کنید، می توانید نشانگر را در تمامی جهات به حرکت در آورید.

شایان ذکر است که محل تقاطع خطوط عمودی و افقی در واقع محل قرار گرفتن نشانگر است. در صورت به حرکت در آوردن خطوط عمودی یا افقی مربوط به نشانگر به ترتیب می توانید نشانگر مذکور را تنها در یکی از دو جهت عمودی یا افقی حرکت دهید. در ضمن می توانید با استفاده از کلید Cursor Movement Control، نشانگر را در جهات بالا، پایین، چپ یا راست حرکت دهید.

در مورد تعداد نشانگرها در یک گراف محدودیتی وجود ندارد. با استفاده از پالت Cursor می توانید آنها را ردیابی نموده، محل قرار گرفتن هر یک را ملاحظه کنید. جهت افزایش تعداد نشانگرها، اندازه‌ی این پالت را تغییر دهید. در ادامه به بررسی بخشهای مختلف پالت Cursor می پردازیم:

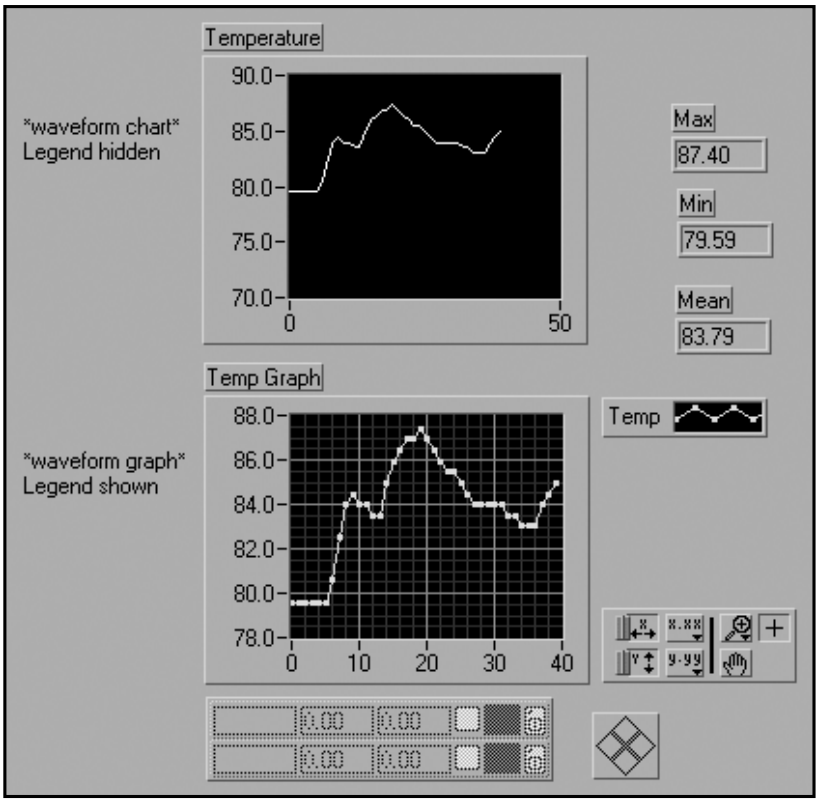
اولین جعبه در سمت چپ این پالت برای تعیین برچسب نشانگر اختصاص داده شده است. دو جعبه‌ی بعدی به ترتیب مختصه‌ی X و Y مربوط به نشانگر را نشان می دهند. برای فعال نمودن و همچنین به حرکت در آوردن نشانگر، بر روی جعبه‌ی سفیدرنگ متناظر با آن (Select Button) کلیک کنید. در حالتی که یکی از نشانگرها انتخاب شده باشد، یک مربع سیاه‌رنگ در داخل جعبه‌ی متناظر با آن ظاهر می شود. هنگامی که بر روی دکمه‌ی Cursor Movement Control کلیک می کنید، تمامی نشانگرهای فعال به حرکت در می آیند. برای تعیین مواردی نظیر شکل، ظاهر و رنگ نشانگرها، توسط ابزار Operating Tool بر روی دکمه‌ی Cursor Display Control کلیک کنید.

پس از کلیک کردن توسط ابزار Operating Tool بر روی جعبه‌ی کوچک در منتهی الیه سمت راست در پالت Cursor، پنجره‌ای باز می شود که به کمک گزینه های موجود در آن می توان جابه‌جایی نشانگر را تنها بر روی نقاط ترسیم شده در گراف محدود کرد. انجام این عمل، جابه‌جایی و حرکت نشانگر را محدود می کند و بدین ترتیب نشانگر تنها می تواند بر روی نقاط منحنی قرار گیرد. جهت دستیابی به زیرمنوی این گزینه، توسط ابزار Operating Tool بر روی آن کلیک کنید. مراقب باشید که اشتباهاً منوی کرکره‌ای را بر روی این جعبه باز نکنید، وگرنه منوی دیگری ظاهر می شود.

## تمرین ۴-۸: تجزیه و تحلیل دمای اندازه‌گیری شده

در این تمرین قصد داریم تا با ایجاد یک برنامه، دمای محیط را در مدت ۱۰ ثانیه، ۴۰ مرتبه اندازه‌گیری کنیم. به عبارت دیگر در هر ۲۵۰ میلی ثانیه یک مرتبه عملیات نمونه برداری را انجام دهیم. در حین جمع آوری داده ها، مقادیر اندازه گیری شده بر روی Waveform Chart به نمایش در می آیند. پس از اتمام عملیات جمع آوری و قرائت داده ها، مقادیر اندازه گیری شده بر روی یک گراف، ترسیم شده، و مقادیر حداکثر، حداقل و میانگین درجه حرارت محاسبه می شود.

۱- یک برنامه‌ی جدید باز نموده، صفحه‌ی پانل آن را مطابق شکل ۳۵-۸ ایجاد کنید.




**شکل ۳۵-۸: صفحه‌ی پانل در برنامه‌ی Temperature Analysis.vi**

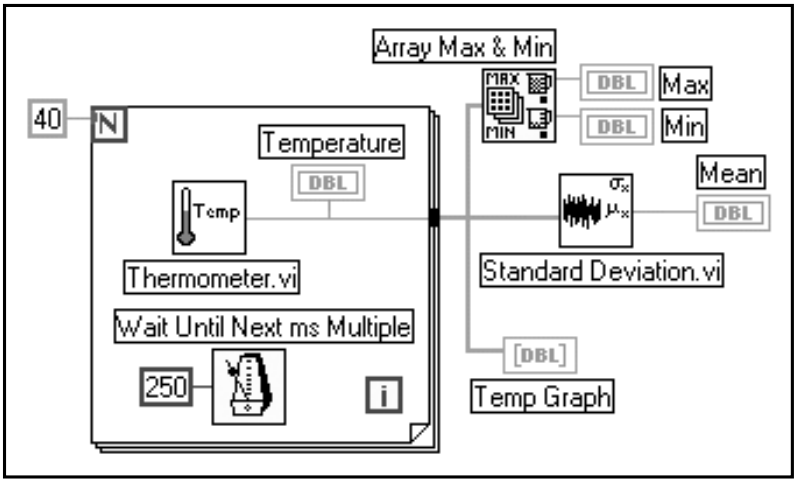
۲- درجه‌بندی محور Y را به گونه‌ای تغییر دهید که فاصله‌ی [70.0 ,90.0] را تحت پوشش قرار دهد. همچنین از فعال بودن سیستم درجه‌بندی خودکار بر روی دو محور X و Y اطمینان حاصل کنید.

۳- منوی کرکره‌ای را بر روی محورهای X و Y باز نموده، با انتخاب گزینه‌ی ...Formatting خطوط شطرنجی را به گراف اضافه کنید. خطوط شطرنجی Major و Minor را در مورد هر دو محور X و Y انتخاب نموده، سپس این خطوط را رنگ‌آمیزی کنید.

۴- جعبه‌ی Legend را بر روی گراف ظاهر سازید. اندازه‌ی این جعبه را از سمت چپ افزایش دهید تا فضای بیشتری برای نوشتن متن در اختیار داشته باشید. حال به کمک ابزار Labeling Tool برچسب Temp را وارد کنید. سپس منوی کرکره‌ای را بر روی بخش نمایش منحنی Temp در جعبه‌ی Legend باز کنید و شکل نقاط یا Point Style را به گونه‌ای تغییر دهید تا نقاط موجود بر روی گراف به صورت مربع‌های کوچک ظاهر شوند. سپس با انتخاب یک رنگ دلخواه، منحنی را رنگ‌آمیزی کنید.

نمودار Temperature مقادیر اندازه‌گیری شده را بلافاصله و در همان زمان قرائت آنها به نمایش در می‌آورد. پس از اتمام عملیات نمونه‌برداری و قرائت داده‌ها، یک منحنی کامل بر روی Temp Graph ترسیم می‌شود. همچنین مقادیر میانگین، ماکزیمم و می‌نیمم به ترتیب در نشان‌دهنده‌های دیجیتالی Mean، Max و Min به نمایش در می‌آیند.

۵- اکنون صفحه‌ی نمودار بلوکی این برنامه را مطابق شکل ۳۶-۸ ایجاد کنید. برای سیم‌کشی و رؤیت ترمینال‌های ورودی و خروجی توابع و دستورهای موجود در شکل، از پنجره‌ی Help کمک بگیرید. در غیر این صورت ممکن است درسیم‌کشی و برقراری ارتباط بین ترمینال‌ها با مشکل مواجه شوید.




**شکل ۳۶-۸: صفحه‌ی نمودار بلوکی در برنامه‌ی Temperature Analysis.vi**

Thermometer.vi

زیربرنامه‌ی Thermometer.vi. اگر به خاطر داشته باشید در یکی از تمرینات فصول قبل این برنامه را ایجاد نمودیم. با انتخاب زیرپالت ...Functions >> Select a VI به این برنامه دسترسی پیدا کنید. به احتمال زیاد برنامه‌ی مذکور در زیرفهرست LabVIEW\My Activity ذخیره شده است. در صورتی که این برنامه را ذخیره نکرده‌اید، یکی از دو برنامه‌ی Thermometer.vi یا Digital Thermometer.vi را که به ترتیب در مسیر LabVIEW\Exercise\Ch5.llb یا زیرپالت Functions >> Tutorial قرار دارند مورد استفاده قرار دهید.

Wait Until Next ms Multiple Function

دستور Wait Until Next ms Multiple (Functions >> Time & Dialog) سبب می‌شود که حلقه‌ی For در هر ۲۵۰ میلی‌ثانیه یک مرتبه به اجرا در آید.

Array Max & Min Function

دستور Array Max & Min (Functions >> Array) حدود ماکزیمم و می‌نیمم آرایه‌ای را که مقادیر اندازه‌گیری شده در آن گنجانده شده است به‌دست می‌دهد.

Standard Deviation.vi

زیربرنامه‌ی Standard Deviation.vi (Functions >> Analysis >> Probability and Statistics) مقدار میانگین دماهای اندازه‌گیری شده را به‌دست می‌دهد.

عملکرد این برنامه به شرح زیر است:

حلقه‌ی For در هر ۲۵۰ میلی‌ثانیه یک مرتبه و در مجموع ۴۰ مرتبه به اجرا در می‌آید. دستور Wait Until Next ms Multiple سبب می‌شود که حلقه‌ی For در هر ۲۵۰ میلی‌ثانیه یک مرتبه تکرار شود.

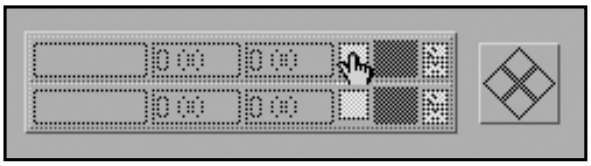
این برنامه، مقادیر قرائت شده را با استفاده از قابلیت اندیس گذاری خودکار در یک آرایه که در چهارچوب حلقه‌ی For ایجاد می گردد ذخیره می کند. پس از تکمیل تکرارهای حلقه‌ی For، اطلاعات موجود درآرایه‌ی مذکور به چندین گره که در شکل ۳۶-۸ نشان داده شده اند پخش می شود. دستور Array Max & Min و زیربرنامه‌ی Standard Deviation.vi به ترتیب حداکثر، حداقل و میانگین مقادیر اندازه گیری شده را به دست می دهند.

۶- اکنون به صفحه‌ی پانل بازگردید و برنامه را اجرا کنید.

Zoom Button

۷- حال با استفاده از بخش Palette تعداد ارقام اعشاری اعداد موجود بر روی محور Y را به ۳ رقم افزایش دهید. اکنون بر روی دکمه‌ی Zoom Button کلیک کنید و با انتخاب یکی از حالات، عملیات بزرگ نمایی را بر روی بخشی از منحنی انجام دهید.

۸- منوی کرکره‌ای را بر روی گراف باز نموده، گزینه‌ی Show >> Display Cursor را انتخاب کنید. همان گونه که قبلاً اشاره شد در ابتدای امر، پالت Cursor به رنگ خاکستری ظاهر می‌شود و بیانگر غیر فعال بودن آن است. برای فعال نمودن این پالت و همچنین برای فعال کردن اولین نشانگر، مطابق شکل ۳۷-۸ بر روی دکمه‌ی Select Button در نوار بالایی این صفحه کلیک کنید.



**شکل ۳۷-۸: نحوه‌ی فعال نمودن پالت Cursor**

به کمک ابزار Operating Tool نشانگر را در صفحه‌ی گراف به حرکت در آورید. در این حالت به تغییرات مقادیر نمایش دهنده های مربوط به موقعیت نشانگر توجه کنید. این مقادیر در تعیین مختصات یک نقطه‌ی خاص بر روی گراف به شما کمک می‌کنند. حال به کمک دکمه های موجود در بخش Cursor Movement Control نشانگر را در جهات بالا، پایین، چپ و راست به حرکت در آورید. دومین نشانگر را نیز با کلیک کردن بر روی دکمه‌ی Select Button در نوار پایینی پالت Cursor فعال سازید. اکنون با استفاده از دکمه های موجود در بخش Cursor Movement Control هر دو نشانگر را حرکت دهید. حال به کمک ابزار Operating Tool بر روی Cursor Display Control کلیک کنید و رنگ یکی از نشانگرها را به دلخواه تغییر دهید.

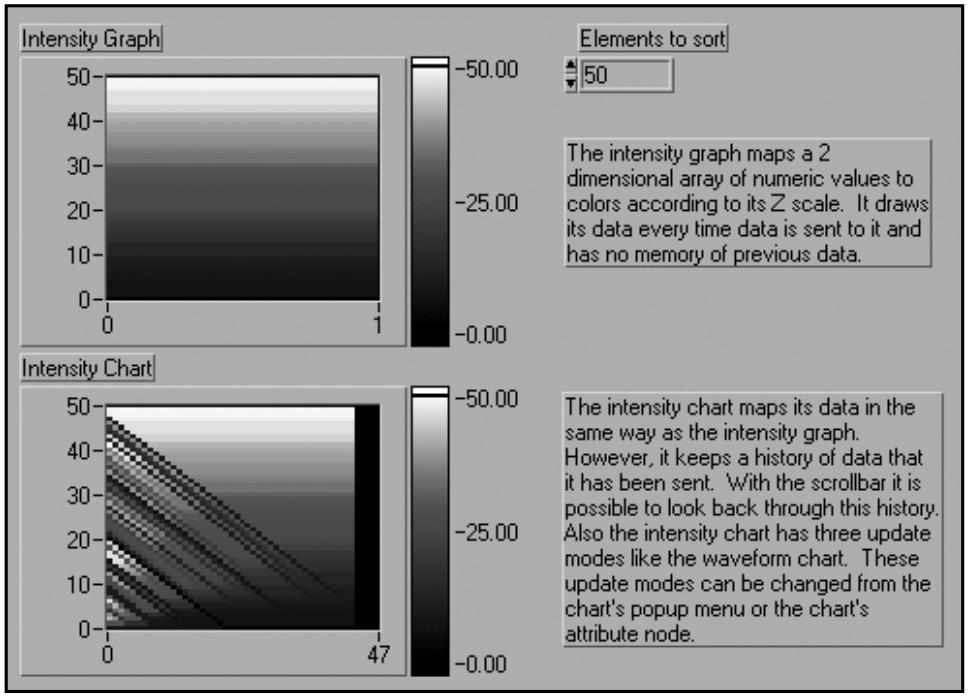
اکنون برچسب Cursor 0 را در بخش در نظر گرفته شده برای نام نشانگر وارد کنید. سپس با ابزار Operating Tool بر روی دکمه‌ی Lock button کلیک کرده، سپس گزینه‌ی Allow Drag را در منوی ظاهر شده غیر فعال کنید. در این حالت گزینه‌ی Lock to Plot را از همان منو انتخاب نمایید. با انتخاب این گزینه قادر خواهید بود نشانگر مذکور را تنها بر روی نقاط منحنی حرکت دهید. با وجود این سعی کنید با کمک دکمه های موجود در بخش Cursor Movement Control نشانگر مذکور را به حرکت در آورید. متوجه خواهید شد که در این حالت، نشانگر تنها بر روی نقاط منحنی جابه‌جا می‌شود.

۹- این برنامه را با عنوان Temperature Analysis.vi در زیرفهرست LabVIEW\My Activity ذخیره نموده، سپس آن را ببندید.

## ترسیم نقشه های سه بعدی در نمودار و گراف

ممکن است این سؤال در ذهن شما ایجاد شده باشد که: «در صورتی که بخواهیم به جای دو متغیر، سه متغیر را نسبت به یکدیگر به تصویر درآوریم تکلیف چیست؟»

پاسخ این سؤال بسیار ساده است. به کمک منحنیهای Intensity به راحتی می توانید این عمل را انجام دهید. در منحنیهای مذکور برای نمایش بعد سوم از رنگهای مختلف استفاده می شود. در نمودار Intensity یک صفحه‌ی نمایش لغزشی در نظر گرفته شده است. حال آن که صفحه‌ی نمایش گراف Intensity فاقد این ویژگی می‌باشد و ثابت و بدون لغزش است. منحنیهای Intensity برای به نمایش در آوردن داده‌های الگوبندی شده نظیر نقشه‌ی زمین و... که شدت رنگ در آنها مشخص کننده‌ی ارتفاع در فضای دوبعدی است بسیار مفید و کاربردی می‌باشند و یا در رویه‌های گرمایی که در آنها شدت رنگ نشان‌دهنده‌ی نحوه‌ی توزیع و پخش شدن گرماست. در شکل ۳۸-۸ نمونه‌ای از منحنیهای Intensity را ملاحظه می کنید.

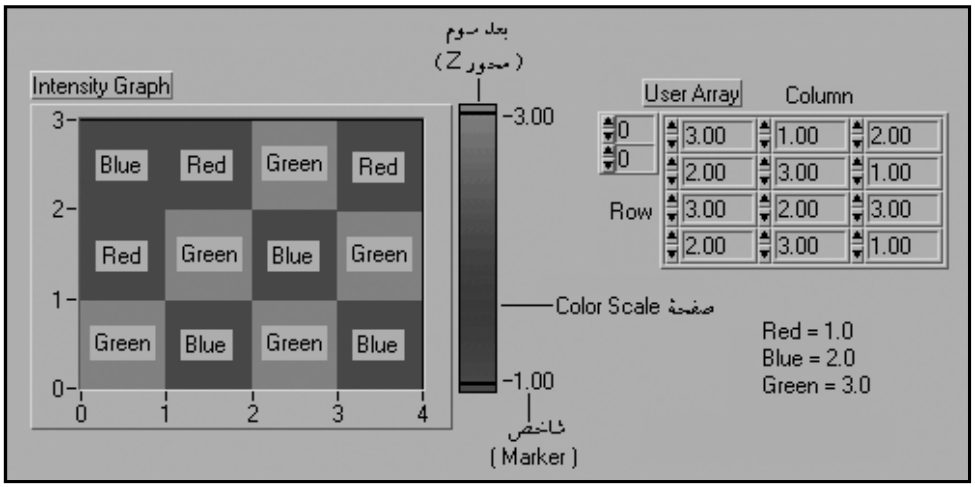



**شکل ۳۸-۸**

منحنیهای Intensity در اغلب موارد نظیر نمودارها و گراف ها عمل می‌کنند با این تفاوت که در آنها برای نمایش بعد سوم از رنگ استفاده می شود. برای نمایش سومین بعد در نقشه یک مقیاس رنگ یا Color Scale تعبیه شده است که به کمک آن می توانید رنگها را تنظیم نموده، سپس آنها را به نمایش در آورید.

ورودی این منحنیها یک آرایه‌ی دوبعدی عددی است که هر یک از مؤلفه های آن با یک رنگ تصویر می‌شود و اندیس مؤلفه ها در آرایه نیز مشخص کننده‌ی محل رنگ بر روی گراف یا نمودار است. رنگ نقشه‌ی اعداد را نه تنها با استفاده از مقیاس رنگ یا Color Scale می توانید تعیین کنید، بلکه تغییر و تعریف رنگها از طریق برنامه و با کمک گره‌ی خصوصیت نیز امکان پذیر است. در مورد این مطلب در فصل ۱۲ سخن خواهیم گفت.

در شکل ۳۹-۸ نمونه‌ی ساده‌ای از یک ماتریس ۳ x ۴ را که در مورد یک Intensity Graph استفاده شده است ملاحظه می کنید. در این شکل هر یک از رنگهای قرمز، آبی و سبز به ترتیب با شماره های 1.0، 2.0 و 3.0 مشخص شده اند.




شکل ۳۹-۸

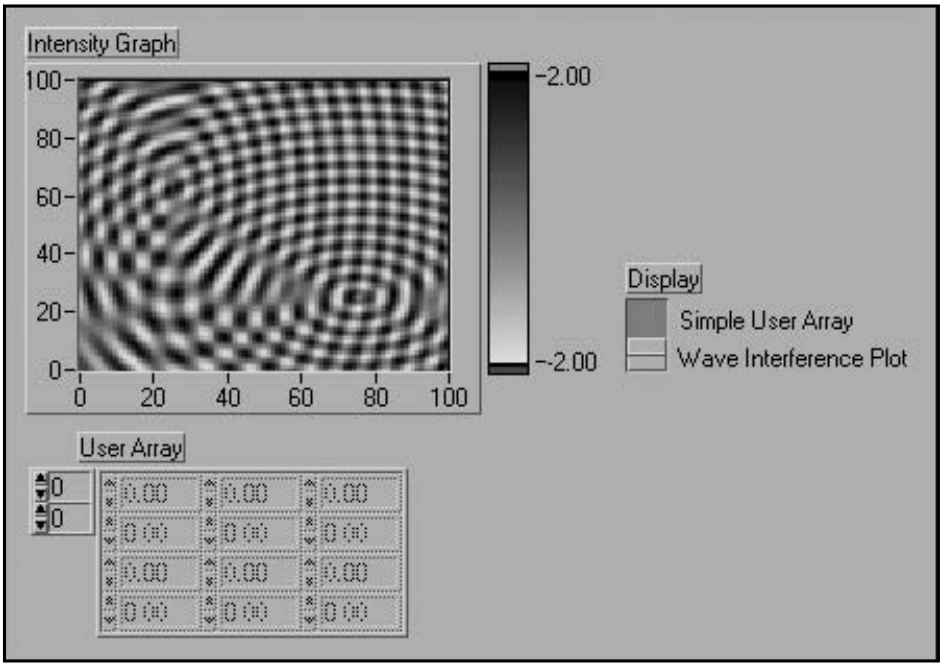
جهت انتساب رنگ به یک مقدار در صفحه‌ی Color Scale، منوی کرکره‌ای را بر روی شاخص (Marker) متناظر با آن باز نموده، گزینه‌ی Marker Color را انتخاب کنید. در نمودارها و گراف های Intensity برای محور Z چند شاخص به صورت پیش فرض در نظر گرفته می شود. بنابراین با حرکت دادن یکی از شاخص ها به کمک ابزار Operating Tool می توانید شیب تدریجی[8] تغییر رنگها را تغییر دهید. برای افزایش تعداد شاخص ها، گزینه‌ی Add Marker را از منوی کرکره ای در Color Scale انتخاب کنید. سپس آنها را به نقاط دلخواه ببرید و یک رنگ جدید به آنها اختصاص دهید.

## تمرین ۵-۸: بررسی Intensity Graph

در این تمرین قصد داریم تا با ایجاد یک برنامه، رویه های موج را به نمایش درآوریم. این برنامه همچنین نحوه‌ی به نمایش درآمدن آرایه‌ی ورودی دوبعدی را در یک گراف شدت رنگ نشان می دهد.

۱- برای درک نحوه‌ی عملکرد و گراف شدت رنگ، برنامه‌ی Intensity Graph Example.vi را از مسیر LabVIEW\Exercise\Ch8.llb باز و اجرا نمایید. پس از اجرای برنامه، رویه های موج را بر روی گراف ملاحظه کنید. محدوده‌ی رنگها در صفحه‌ی نمودار بلوکی و با کمک گره‌ی خصوصیت در گراف شدت رنگ تعیین می گردد. با کلیک کردن توسط ابزار Operating Tool بر روی جعبه های رنگ در اولین فریم ساختار ترتیبی، محدوده‌ی رنگها را تغییر دهید. سپس از صفحه‌ی ظاهر شده یک رنگ جدید انتخاب نموده، مجدداً برنامه را اجرا کنید.

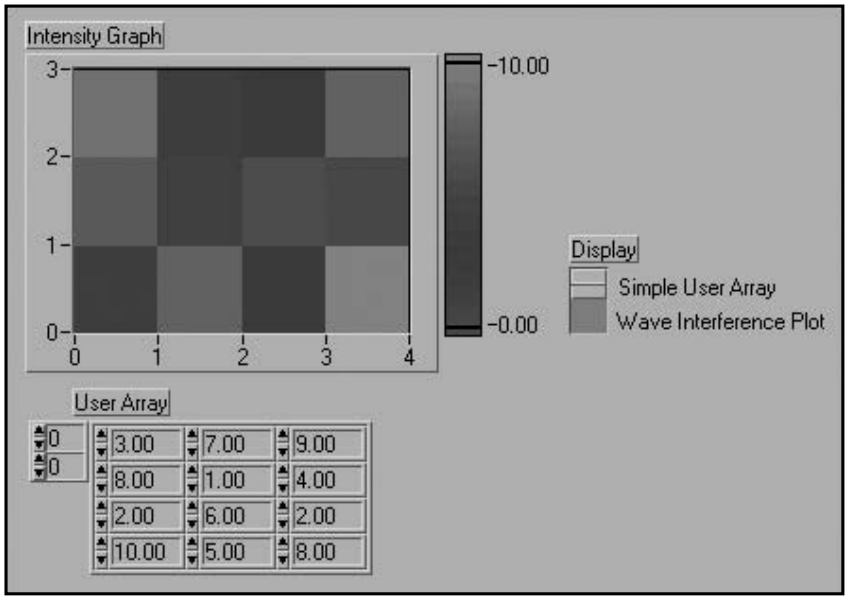
8- Gradient




**شکل ۴۰-۸: صفحه‌ی پانل در برنامه‌ی Intensity Graph Example.vi**

۲- وضعیت کلید Display را به حالت Simple User Array تغییر دهید و در المان User Array مقادیری از 0.0 تا 10.0 را به دلخواه وارد کنید.

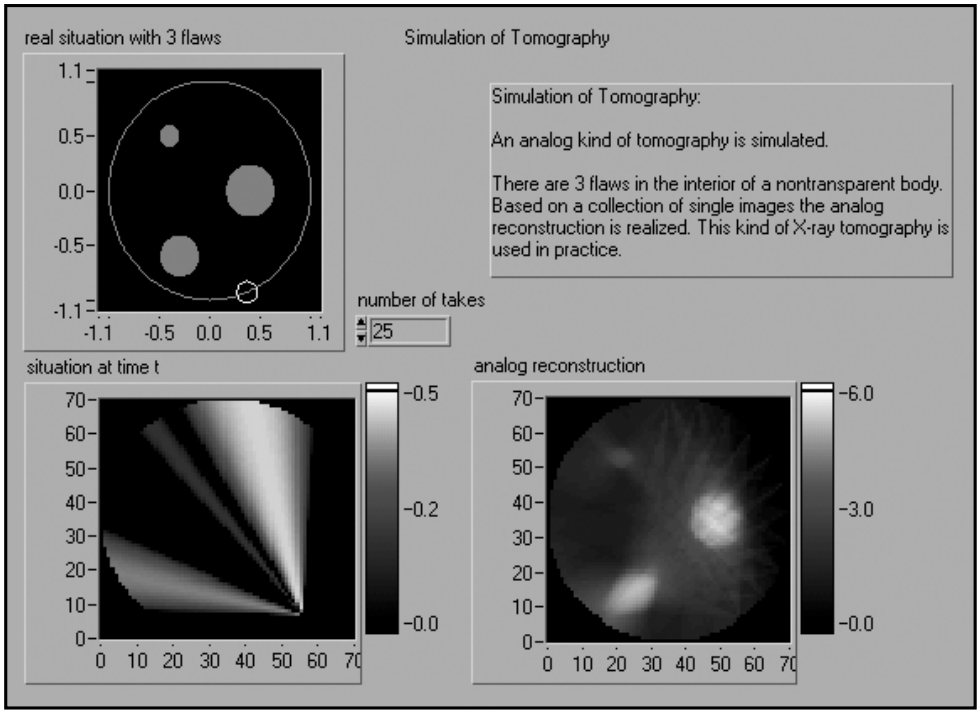
محدوده‌ی رنگها در صفحه‌ی نمودار بلوکی به کمک گره‌ی خصوصیت در Intensity Graph، تعریف شده است. بدین صورت که به دو رنگ آبی و قرمز به ترتیب دو مقدار 0.0 و 10.0 اختصاص داده شده است. پس از وارد نمودن تمامی مقادیر، برنامه را اجرا کنید. توجه داشته باشید که چگونه اندازه‌ی هر یک از المان ها بر روی گراف به تصویر کشیده می شود. حال مقادیر وارد شده را تغییر دهید و مجدداً برنامه را اجرا کنید.




**شکل ۴۱-۸: انتساب رنگها در صفحه‌ی پانل برنامه‌ی Intensity Graph Example.vi**

۳- برای درک نحوه‌ی عملکرد این برنامه صفحه‌ی نمودار بلوکی آن را نیز به دقت بررسی کنید.

۴- حال برنامه را ببندید و از ذخیره نمودن تغییرات ایجاد شده خودداری کنید. برای مشاهده مثالهای دیگری از منحنیهای شدت رنگ، به مسیر LabVIEW\Examples\General\graphs\intgraph.llb مراجعه کنید. همچنین می توانید برنامه های Simulation of Tomography و Heat Equation Example را از مسیر LabVIEW\Examples\Analysis\mathxmpl.llb اجرا کنید.




**شکل ۴۲-۸: صفحه‌ی پانل در برنامه‌ی Simulation of Tomography.vi**

## خلاصه

در نرم‌افزار LabVIEW می‌توانید داده‌ها و نتایج به‌دست آمده را با به‌کارگیری نمودار و گراف به صورت گرافیکی نمایش دهید. اضافه نمودن داده های جدید به داده های قدیمی در نمودار امکان‌پذیر است و می توان در هر لحظه یک نقطه یا مجموعه‌ای از نقاط را به داده های قبلی اضافه نمود. به عبارت دیگر نمودار، نقاط منحنی را بلادرنگ ترسیم می کند. بنابراین می توانید مقادیر کنونی را به همراه مقادیر قبلی ملاحظه کنید. حال آن که گراف، یک بلوک از داده های جمع آوری شده را یک‌جا به نمایش در می آورد.

در بسته‌ی نرم‌افزاری LabVIEW سه نوع گراف Waveform، XY و Intensity وجود دارد. Waveform Graph تنها جهت ترسیم توابع یک به یک که در آنها مقادیر X به صورت متقارن توزیع شده‌اند به کار برده می شود. از این گونه توابع می توان امواج سینوسی را نام برد.

XY Graph یک گراف همه‌منظوره است که برای ترسیم توابع چند مقداری نظیر اشکال دایره‌ای در

مختصات دکارتی به کار برده می شود. در این گراف، آرایه‌ی حاوی مقادیر Y در مقابل آرایه‌ی حاوی مقادیر X ترسیم می شود.

به دلیل اینکه منحنیهای Intensity قادر به نمایش داده های سه متغیره بر روی یک صفحه‌ی دوبعدی هستند، می توان از آنها برای نمایش رویه ها استفاده نمود. در این منحنیها برای به تصویر کشیدن بعد سوم، از رنگ استفاده می شود. ورودی این منحنیها یک آرایه‌ی دوبعدی عددی است که هر یک از مؤلفه های آن با یک رنگ تصویر شده و اندیس مؤلفه ها در آرایه نیز مشخص کننده‌ی محل رنگ بر روی گراف یا نمودار می باشد. در اغلب موارد عملکرد منحنیهای Intensity نظیر نمودارها و گراف های دوبعدی استاندارد می باشد.

به کمک دو بخش Legend و Palette می توان شکل ظاهری گراف و نمودار را تغییر داد. همچنین برای انطباق داده ها می توان درجه بندی محورها را تغییر داد و از چند نشانگر بر روی منحنی استفاده نمود. ترسیم چند منحنی بر روی صفحات نمایش نمودار و گراف امکان پذیر است.

حرکات مکانیکی کلیدهای جبری به شما اجازه می دهد تا رفتار و عملکرد آنها را در هنگام کلیک نمودن بر روی المان های جبری کنترل کنید. به عنوان مثال می توانید عملکرد یک کلید را به گونه ای تنظیم نمایید که پس از قرائت مقدار جدید، مجدداً به حالت پیش فرض خود باز گردد و با این روش همواره برای عملکرد مجدد خود آماده باشد. به کمک حرکات مکانیکی همچنین می توانید عملکرد ماوس را به گونه ای تنظیم نمایید که کلیک کردن بر روی ماوس در هنگام فشار دادن دکمه‌ی ماوس و یا در لحظه‌ی رها کردن آن به ثبت برسد.

۸

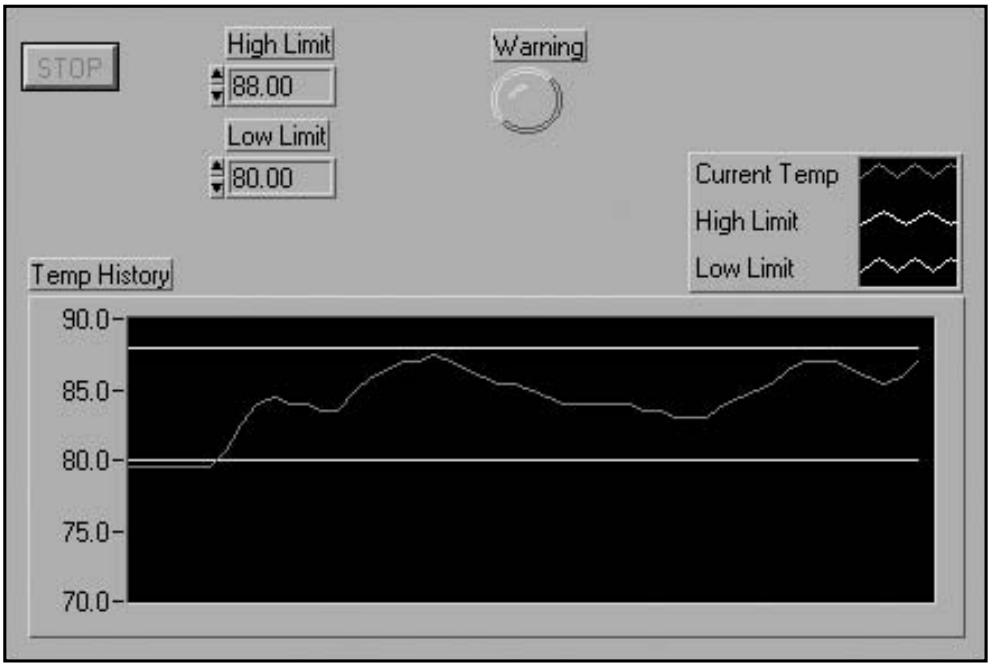
## تمرینات اضافی

### تمرین ۶-۸: گراف های میله‌ای

برنامه‌ی Array to Bar Graph Demo.vi را از مسیر LabVIEW\Exercise\Ch8.llb باز کنید. حال این برنامه را اجرا نموده، به عملکرد آن دقت کنید. پارامترهای آن را تغییر دهید و مجدداً برنامه را اجرا کنید؛ سپس صفحه‌ی نمودار بلوکی آن را بررسی نمایید.

### تمرین ۷-۸: اندازه گیری دما

برنامه ای ایجاد کنید که دما را هر ثانیه یک مرتبه اندازه گیری نموده، نمودار آن را در مد Scope ترسیم کند. در صورتی که دمای اندازه گیری شده خارج از محدوده‌ی از پیش تعیین شده باشد، LED موجود در صفحه‌ی پانل روشن می شود. در صفحه‌ی پانل، باید مقادیر حداکثر و حداقل دما به همراه منحنی به نمایش در آیند. این برنامه باید به گونه ای ایجاد شود که کاربر بتواند در صورت لزوم مقادیر حداکثر و حداقل دما را از صفحه‌ی پانل تغییر دهد. برای آغاز برنامه نویسی از صفحه‌ی پانل نشان داده شده در شکل ۴۳-۸ کمک بگیرید.




**شکل ۴۳-۸: صفحه‌ی پانل در برنامه‌ی Temperature Limit.vi**

این برنامه را با عنوان Temperature Limit.vi در زیرفهرست LabVIEW\My Activity ذخیره کنید.

## تمرین ۸-۸: حداکثر و حداقل دما

برنامه‌ی ایجاد شده در تمرین ۷-۸ را به گونه‌ای تغییر دهید که حداکثر و حداقل دمای اندازه‌گیری شده را نشان دهد. این برنامه را با عنوان Temp Limit (max/min).vi در زیرفهرست LabVIEW\My Activity ذخیره نمایید.

برای انجام این تمرین، از ثبات انتقال و دستور Array Max & Min (Functions >> Array) استفاده کنید.

## تمرین ۹-۸: ترسیم یک آرایه‌ی دوبعدی

برنامه‌ای ایجاد کنید که یک آرایه‌ی دوبعدی ۱۰×۳ را که شامل اعداد تصادفی است تولید نماید. پس از تولید آرایه، اندیس هر یک از سطرها را غیرفعال نموده، مؤلفه‌های هر سطر را بر روی یک گراف ترسیم کنید. بنابراین درصفحه‌ی پانل این برنامه باید سه گراف ترسیم شود.

این برنامه را با عنوان Extract 2D Array.vi در زیرفهرست LabVIEW\My Activity ذخیره کنید.

# ۹

# داده‌های رشته‌ای و ورودی/خروجی فایل

## در این فصل با موارد زیر آشنا می‌شوید:

- المان‌های کنترل و نشان‌دهنده‌ی رشته‌ای
- نحوه‌ی به‌کارگیری دستورهای خاص در مورد داده‌های رشته‌ای
- روش تبدیل داده‌های عددی به رشته‌ای و برعکس
- شیوه‌ی استفاده از برنامه‌های File I/O جهت ذخیره‌ی داده‌ها در یک فایل و همچنین خواندن آنها

# LabVIEW

۹

# داده‌های رشته‌ای و ورودی/خروجی فایل

در این فصل به معرفی و بررسی داده‌های رشته‌ای می‌پردازیم. در بسته‌ی نرم‌افزاری LabVIEW دستورهای بسیاری در مورد رشته‌ها تعبیه شده است. این دستورها تقریباً شبیه به دستورهای استفاده شده در مورد آرایه‌ها هستند و به شما اجازه می‌دهند تا بر روی داده‌های رشته‌ای اعمال مختلفی انجام دهید. در این فصل نحوه‌ی خواندن و نوشتن داده‌ها را در یک فایل مورد بررسی قرار می‌دهیم.

در این فصل اهداف زیر دنبال می‌شوند:

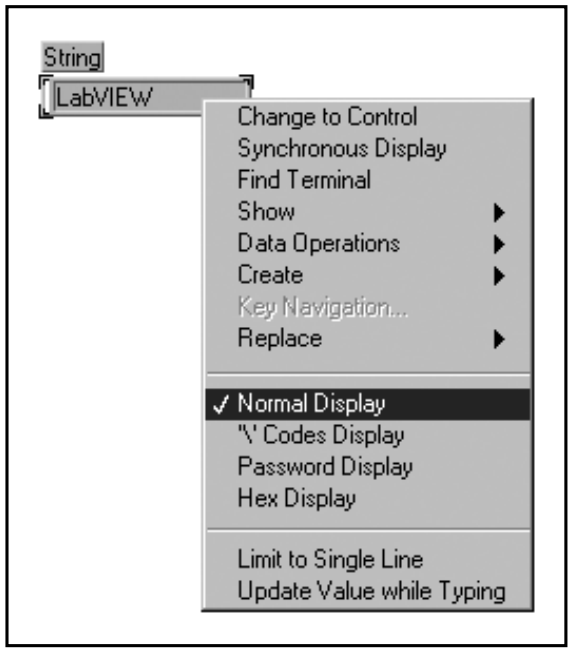
- فراگیری مطالبی در مورد المان‌های کنترل و نشان‌دهنده‌ی رشته‌ای.
- درک نحوه‌ی به‌کارگیری دستورهای خاص در مورد داده‌های رشته‌ای.
- چگونگی تبدیل داده‌های عددی به رشته‌ای و برعکس.
- استفاده از برنامه‌های File I/O جهت ذخیره‌ی داده‌ها در یک فایل و سپس خواندن آنها.

اصطلاحات کلیدی موجود در این فصل عبارتند از:

- نوار پیمایش (Scrollbar)
- جدول (Table)
- فایل صفحه‌گسترده (Spreadsheet File)

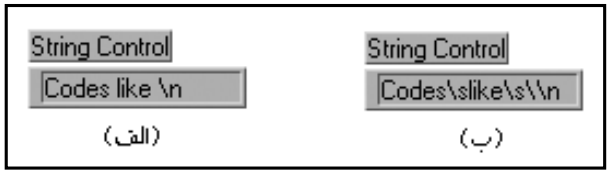
## مطالبی در مورد رشته‌ها

در فصل ۴ به معرفی رشته‌ها پرداختیم و خاطر نشان کردیم که یک رشته به طور ساده مجموعه‌ای از کاراکترهای ASCII است. در اغلب موارد می‌توانید از رشته‌ها به جای پیغامهای متنی استفاده نمایید. به عنوان مثال در کنترل ابزارهای اندازه‌گیری می‌توانید داده‌های عددی را به صورت کاراکترهای رشته‌ای درآورید، سپس جهت انجام عملیات و پردازش بر روی داده‌ها، این رشته‌ها را به اعداد تبدیل کنید. برای ذخیره نمودن داده‌های عددی بر روی دیسک نیز می‌توانید از رشته‌ها کمک بگیرید. در بسیاری از برنامه‌های موجود در مورد ورودی/خروجی فایل، قبل از ذخیره نمودن مقادیر عددی در یک فایل، داده‌های عددی به رشته‌ای تبدیل می‌شوند. در ادامه به بررسی برخی از گزینه‌های موجود در منوی کرکره‌ای یک المان رشته‌ای می‌پردازیم:




**شکل ۱-۹: گزینه‌های موجود در منوی کرکره‌ای یک المان رشته‌ای**

در منوی کرکره‌ای المان‌های کنترل و نشان‌دهنده‌ی رشته‌ای، گزینه‌های کاربردی و مفیدی در نظر گرفته شده است. به عنوان مثال، نمایش دادن کاراکترهایی نظیر Return، Enter، Tab و Back space که به طور معمول غیر قابل نمایش می‌باشند امکان‌پذیر است. اگر به جای روش نمایش معمول یا Normal Display از روش نمایش Codes Display '\' استفاده کنید، کاراکترهای غیر قابل نمایش نظیر Back space به همراه کد متناظر با آن ظاهر می‌شوند. در شکل ۲-۹ نمونه‌ای از کاربرد این روش نمایش، نشان داده شده است.



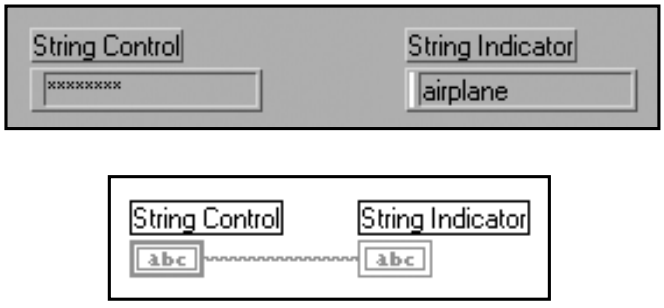

**شکل ۲-۹**

در شکل ۹-۲ (الف) در یک المان کنترل رشته ای، کدهای Backslash در مد Normal Display نمایش داده شده‌اند. پس از انتخاب گزینه‌ی Codes Display[1]، فاصله‌ی در نظر گرفته شده بین کاراکترهای مندرج در این رشته مطابق شکل ۹-۲ (ب) نشان داده می شوند. در جدول ۹-۱ مفهوم کدهای Backslash را ملاحظه می کنید.

**جدول ۹-۱: کدهای Backslash**

| LabVIEW Interpretation | Code |
|---|---|
| Hexadecimal value of an 8-bit character; alphbetical characters must be upper case | \00-\FF |
| Backspace (ASCII BS, equivalent to \08) | \b |
| Formfeed (ASCII FF, equivalent to \OC) | \f |
| New Line (ASCII LF, equivalent to \OA) | \n |
| Return (ASCII CR, equivalent to \0D) | \r |
| Tab (ASCII HT, equivalent to \09) | \t |
| Space (equivalent to \20) | \s |
| Backslash (ASCII \, equivalent to \5C) | \\ |

در صورت انتخاب گزینه‌ی Password Display از منوی کرکره‌ای در المان‌های کنترل یا نشان‌دهنده، برای هر یک از کاراکترهای رشته‌ای تنها یک کاراکتر «*» ثبت می‌گردد. بدین ترتیب رؤیت مطالب و کاراکترهای تایپ شده امکان‌پذیر نخواهد بود. همان گونه که در شکل ۹-۳ ملاحظه می‌کنید در حالی که در صفحه‌ی پانل تنها یک رشته از کاراکترهای «********» ظاهر می‌شود، در صفحه‌ی نمودار بلوکی داده‌های واقعی رشته قرائت می‌شوند. به طور کلی می‌توان گفت که این طرز نمایش در صورتی که قصد داشته باشید برای تمام یا بخشی از برنامه، «کلمه‌ی عبور[1]» در نظر بگیرید بسیار مفید است.




**شکل ۹-۳: استفاده از گزینه‌ی Password Display جهت غیر قابل رؤیت ساختن کلمه‌ی عبور**

با انتخاب گزینه‌ی Hex Display از منوی کرکره‌ای المان رشته‌ای، کاراکترهای آن در مبنای ۱۶ به نمایش در می آیند.

1- Password

## رشته‌های تک‌خطی

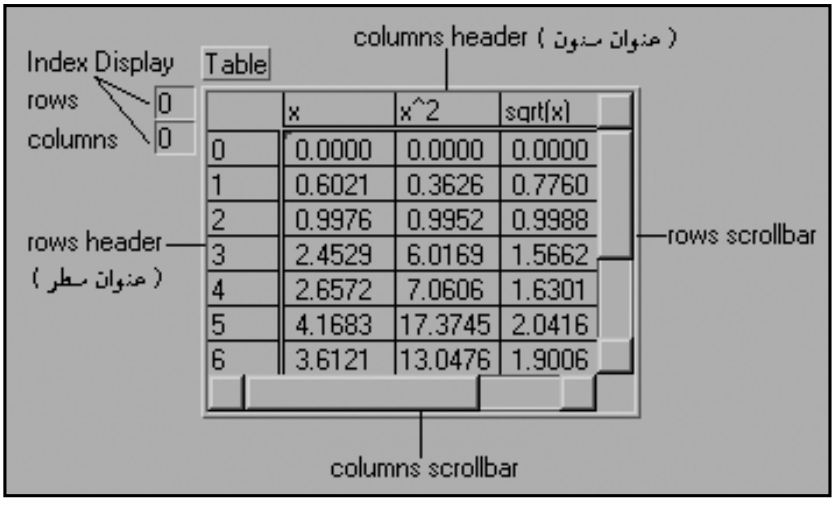
در صورت انتخاب گزینه‌ی Limit to Single Line از منوی کرکره‌ای رشته، متن وارد شده برای رشته‌ی مورد نظر تنها به یک سطر محدود می‌شود. بدین معنی که با فشار دادن کلید <Enter> یا <Return>، متن وارد شده، پایان یافته تلقی می‌شود. در صورت عدم انتخاب این گزینه یا به بیان دیگر در صورت عدم وجود محدودیت در نوشتن رشته در یک سطر، فشار دادن کلید <Enter> یا <Return> سبب می‌شود تا نشانگر به سطر بعدی پرش کند و به شما اجازه دهد تا مطالب بیشتری را تایپ نمایید.

## نوار پیمایش

در صورت انتخاب گزینه‌ی Scrollbar << Show از منوی کرکره‌ای رشته، یک نوار پیمایش بر روی المان کنترل یا نشان‌دهنده‌ی رشته‌ای ظاهر می‌شود. از این گزینه می‌توانید برای به حداقل رساندن فضای اشغال شده توسط المان‌های کنترل رشته‌ای که حاوی یک متن بزرگ در صفحه‌ی پانل هستند استفاده نمایید. توجه داشته باشید تا زمانی که اندازه‌ی رشته را به حد کافی برای گنجایش نوار پیمایش افزایش ندهید این گزینه به رنگ خاگستری ظاهر می شود و بدین معنی است که گزینه‌ی مذکور غیر فعال است.

## جدول

جدول یک ساختار خاص در بسته‌ی نرم‌افزاری LabVIEW است که جهت نمایش یک آرایه‌ی دوبعدیِ حاوی رشته‌ها مورد استفاده قرار می‌گیرد. این گزینه در زیرپالت String & Table << Controls قرار دارد. در شکل ۴-۹ یک جدول با تمامی اجزای آن به طور کامل نشان داده شده است.




**شکل ۴-۹: شمای یک جدول**

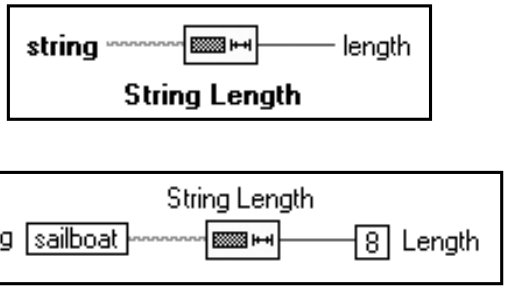
همان گونه که در شکل ۴-۹ ملاحظه می کنید در هر جدول دو عنوان سطر و ستون در نظر گرفته شده است که کاربر به دلخواه می تواند آنها را ظاهر سازد یا مخفی نماید. عناوین مذکور، توسط یک فضای باریک از محل در نظر گرفته شده برای قرار گرفتن داده‌ها جدا می شوند. نظیر موارد ذکر شده‌ی قبلی می توانید با به کارگیری یکی از دو ابزار Operating Tool یا Labeling Tool عناوین دلخواه را در این جداول تایپ و سپس وارد نمایید. همچنین می توانید این عناوین را به کمک گره‌ی خصوصیت بخوانید و آنها را به روز برسانید. در فصل ۱۲ در مورد این موضوع سخن خواهیم گفت.

صفحه‌ی نمایش دهنده‌ی اندیس در جدول مشخص می کند که کدام یک از اجزای جدول در گوشه‌ی بالایی سمت چپ جدول دیده می شود. برای بررسی نحوه‌ی به کارگیری جدول، برنامه‌ی Building Tables.vi را از مسیر LabVIEW\Exercise\Ch9.llb باز و اجرا کنید.

## به کارگیری دستورهای مخصوص رشته‌ها

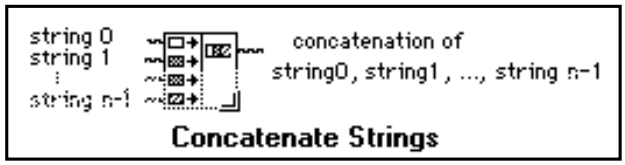
در این بخش به بررسی برخی از دستورها و توابع مربوط به رشته‌ها می پردازیم. این دستورها در زیرپالت String >> Functions قرار دارند. به شما توصیه می کنیم تمامی توابع و دستورهای موجود در این زیرپالت را بررسی کنید. جهت درک عملکرد دستورهای مذکور بهتر است از پنجره‌ی Help استفاده نمایید.

دستور String Length تعداد کاراکترهای یک رشته را به دست می دهد. همان طور که در شکل ۵-۹ ملاحظه می کنید آیکن این دستور تعداد کاراکترهای رشته‌ی sailboat را با عدد 8 مشخص کرده است.



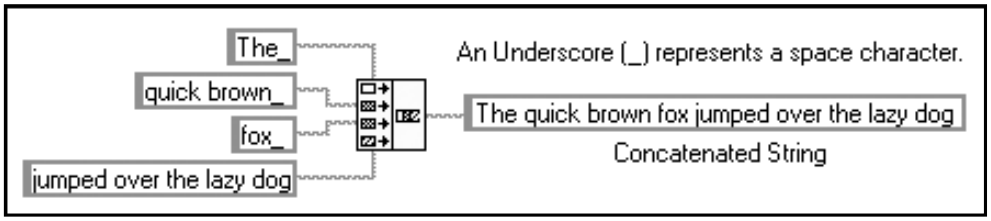

**شکل ۵-۹: آیکن دستور String Length به همراه اتصالات و یک نمونه از کاربرد آن**

دستور Concatenate Strings تمامی ورودیهای یک رشته را به صورت یک زنجیره به یکدیگر متصل می کند و آنها را در خروجی به صورت یک رشته‌ی منفرد نمایش می دهد.



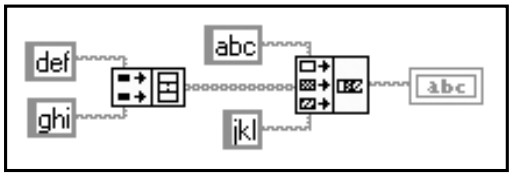

**شکل ۶-۹: آیکن دستور Concatenate Strings به همراه اتصالات آن**

هنگامی که آیکن این دستور را برای اولین بار بر روی صفحه‌ی نمودار بلوکی قرار می دهید به صورت شکل نشان داده شده در سمت راست ظاهر می شود. برای افزایش تعداد ورودیها، اندازه‌ی این آیکن را با استفاده از ابزار Positioning Tool افزایش دهید. در شکل ۷-۹ ملاحظه می کنید که این دستور، ۴ رشته‌ی ورودی را به صورت زنجیره در می آورد و آنها را توسط یک رشته در خروجی نشان می دهد.



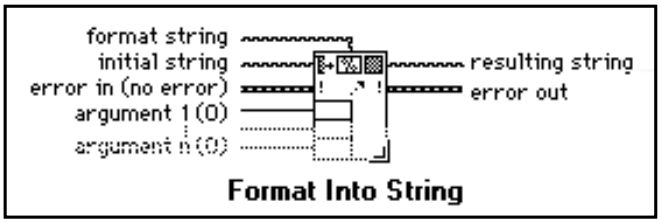

**شکل ۷-۹: نمونه‌ای از کاربرد دستور Concatenate Strings**

علاوه بر آن که می توانید رشته های ساده‌ای را نظیر آنچه در شکل ۷-۹ نشان داده شده است به عنوان ورودی به این دستور اعمال نمایید، دستور مذکور قابلیت پذیرفتن آرایه‌ی یک‌بعدی از رشته ها را به عنوان ورودی نیز داراست. همان گونه که قبلاً عنوان شد خروجی این دستور یک رشته خواهد بود که اجزای آن رشته های آرایه‌ای هستند. در شکل ۸-۹ ترکیبی از ورودیهای رشته‌ای ساده و رشته های آرایه‌ای را به عنوان ورودی این دستور ملاحظه می کنید.



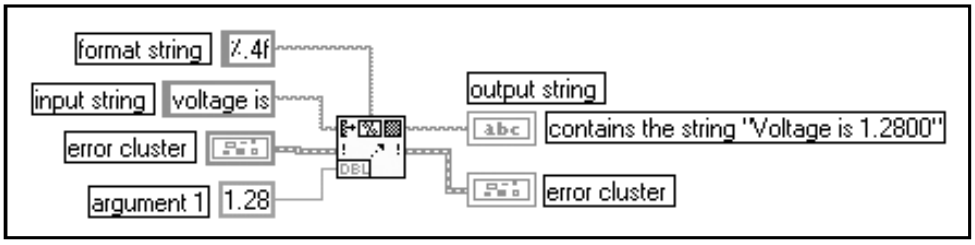

**شکل ۸-۹: ترکیبی از ورودیهای رشته‌ای ساده و رشته های آرایه‌ای به عنوان ورودی دستور Concatenate Strings**

در بسیاری موارد لازم است رشته ها را به اعداد یا اعداد را به رشته ها تبدیل نمایید. دو دستور Format Into String و Scan From String جهت انجام تبدیلات مذکور به کار می روند. البته دستورهای دیگری نیز جهت انجام این اعمال وجود دارند. اما در اینجا تنها این دو دستور را مورد بررسی قرار می دهیم.




**شکل ۹-۹: آیکن دستور Format Into String به همراه اتصالات آن**

همان گونه که احتمالاً از شکل ۹-۹ استنباط نموده‌اید دستور Format Into String داده‌های عددی را به رشته‌ای تبدیل می‌کند. در شکل ۹-۱۰ نمونه‌ای از کاربرد این دستور را ملاحظه می‌کنید.




**شکل ۹-۱۰: نمونه‌ای از کاربرد دستور Format Into String**

در مثال ساده‌ی فوق ملاحظه می‌کنید که دستور Format Into String عدد اعشاری 1.28 را به یک رشته‌ی ۶ بایتی 1.2800 تبدیل کرده است. دستور Format Into String آرگومان ورودی را که به فرمت عددی است، مطابق با فرمت مشخص شده در ورودی format string به یک داده‌ی رشته‌ای تبدیل می‌کند. این دستور، رشته‌ی تبدیل شده‌ی اخیر را در صورت وجود به ورودی initial string الحاق می‌کند و خروجی را در resulting string نمایش می‌دهد. در جدول ۹-۲ چند نمونه از عملکرد دستور Format Into String را ملاحظه می‌کنید.

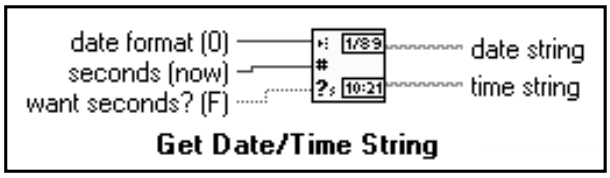
**جدول ۹-۲**

| رشته ی اولیه | فرمت رشته | عدد | رشته ی به‌دست آمده |
|---|---|---|---|
| (empty) | score=%2d%% | 87 | score=87% |
| score=_ | %2d%% | 87 | score=87% |
| (empty) | level=%7.2eV | 0.03642 | level=3.64E-2V |
| (empty) | %5.3f | 5.67N | 5.670N |

در نمونه‌های نشان داده شده در جدول ۹-۲ وجود کاراکتر underline یا «_» در انتهای هر رشته بیانگر یک فضای خالی است. کاراکتر «%» نشان‌دهنده‌ی مشخصات و نحوه‌ی فرمت کردن است. به عنوان مثال در «% number 1 . number2» طول رشته توسط «number 1» مشخص می‌شود و «number 2» نیز تعداد ارقام اعشار را بیان می‌کند. کاراکتر «f» عدد ورودی را به یک عدد اعشاری فرمت می‌کند در حالی که کاراکتر «d» عدد ورودی را به فرمت یک عدد صحیح دهدهی درمی‌آورد. کاراکتر «e» نیز عدد ورودی را به صورت یک عدد اعشاری با نماد علمی فرمت می‌کند.

برای افزایش تعداد ورودیها به منظور تبدیل همزمان چند مقدار به یک رشته، اندازه‌ی آیکن دستور Format Into String را با استفاده از ابزار Positioning Tool تغییر دهید.

در صورت نیاز به تاریخ و زمان فعلی از دستور Get Date/Time String (Functions >> Time & Dialog) استفاده نمایید. این دستور برای مشخص کردن زمان و تاریخ داده‌ها بسیار مفید است. توجه داشته باشید که سیم‌کشی ورودی به آیکن این دستور الزامی نیست. دستور مذکور مقادیر پیش‌فرض زمان و تاریخ سیستم عامل رایانه را مورد استفاده قرار می‌دهد. در شکل ۹-۱۱ آیکن این دستور را به همراه اتصالات ورودی و خروجی آن ملاحظه می‌کنید.

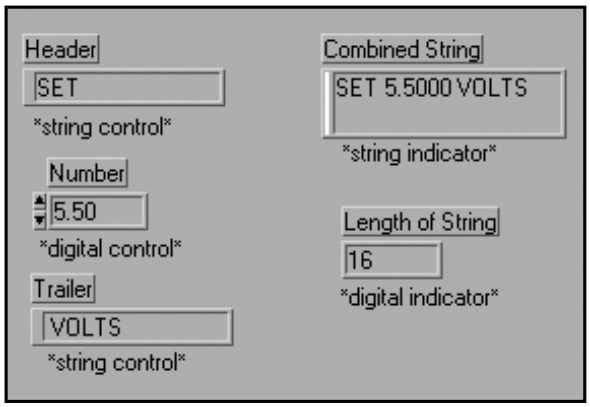



**شکل ۱۱-۹: آیکن دستور Get Date/Time String به همراه اتصالات آن**

## تمرین ۱-۹: ترکیب کردن رشته‌ها

در این تمرین قصد داریم تا یک عدد را به یک رشته تبدیل نموده، سپس رشته‌ی حاصل را با چند رشته‌ی دیگر ترکیب و تمامی آنها را به صورت یک رشته‌ی منفرد تبدیل کنیم. در این برنامه همچنین قصد داریم تا طول رشته‌ی خروجی حاصل را نیز به دست آوریم.

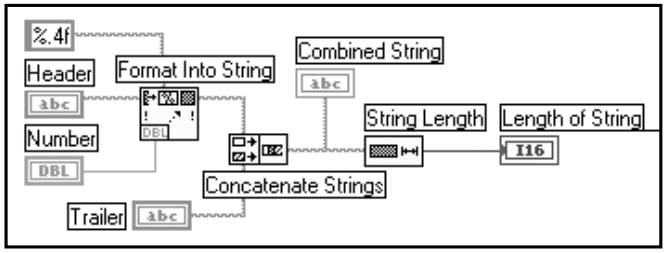
۱- یک برنامه‌ی جدید باز نموده و صفحه‌ی پانل آن را مطابق شکل ۱۲-۹ ایجاد کنید.




**شکل ۱۲-۹: صفحه‌ی پانل در برنامه‌ی Build String.vi**

همان گونه که در شکل ۱۲-۹ ملاحظه می کنید در این برنامه دو المان رشته‌ای (String Control) و یک المان کنترل دیجیتالی (Digital Control) به صورت یک زنجیره در می‌آید و نتیجه‌ی حاصل در نشان‌دهنده‌ی رشته‌ای (String Indicator) نمایش داده می‌شود و درنهایت یک نشان‌دهنده‌ی دیجیتالی (Digital Indicator) که با برچسب Length of String مشخص شده است، تعداد کاراکترهای رشته‌ی حاصل را نشان می‌دهد. گزینه‌ی Representation را از منوی کرکره‌ای نشان‌دهنده‌ی دیجیتالی انتخاب کنید و نوع نمایش داده‌ها را به فرمت I16 تغییر دهید.

۲- صفحه‌ی نمودار بلوکی این برنامه را مطابق شکل ۱۳-۹ ایجاد نمایید.




**شکل ۱۳-۹: صفحه‌ی نمودار بلوکی در برنامه‌ی Build String.vi**

در ادامه به بررسی برخی از دستورهای موجود در صفحه‌ی نمودار بلوکی این برنامه می پردازیم:

Format Into String Function

دستور Format Into String (Functions >> String) عددی را که توسط کاربر در المان Number وارد می شود به یک داده‌ی رشته‌ای با ۴ رقم اعشار تبدیل می‌کند.

Concatenate Strings Function

دستور Concatenate Strings (Functions >> String) تمامی رشته های ورودی را به یک رشته‌ی زنجیره‌ای در خروجی تبدیل می‌کند. برای افزایش تعداد ورودیها، اندازه‌ی آیکن این دستور را با استفاده از ابزار Positioning Tool تغیر دهید.

String Length Function

دستور String Length (Functions >> String) تعداد کاراکترهای رشته‌ی خروجی را به‌دست می دهد.

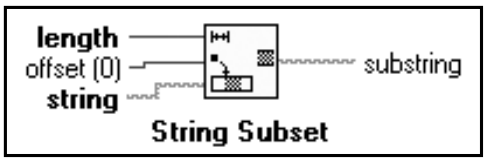
۳- اکنون به صفحه‌ی پانل بازگردید و در فضای در نظر گرفته شده برای دو داده‌ی رشته‌ای و عدد ورودی، به دلخواه دو رشته و یک عدد وارد کنید. حتماً در انتهای رشته‌ی Header و در ابتدای رشته‌ی Trailer فضای خالی در نظر بگیرید وگرنه کاراکترها و اعداد بر روی صفحه‌ی نشان دهنده‌ی رشته‌ی خروجی، بدون فاصله به نمایش در می آیند. اکنون برنامه را اجرا کنید.

۴- این برنامه را با عنوان Build String.vi در زیرفهرست LabVIEW\My Activity ذخیره کنید، سپس آن را ببندید.

## دستورهای تجزیه کننده‌ی رشته ها

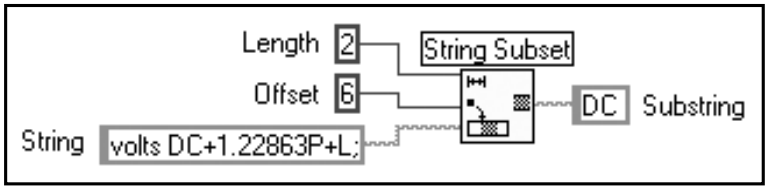
برخی مواقع لازم است تا قسمتی از رشته را جدا نمایید یا آن را به داده های عددی تبدیل کنید. دستورهای تجزیه کننده به شما کمک می کنند تا این اعمال را انجام دهید.

برای دستیابی به بخش معینی از رشته، دستور String Subset را به کار برید. به بیان ساده می توان گفت که این دستور شبیه به دستور Array Subset عمل می کند.




**شکل ۱۴-۹: آیکن دستور String Subset به همراه اتصالات آن**

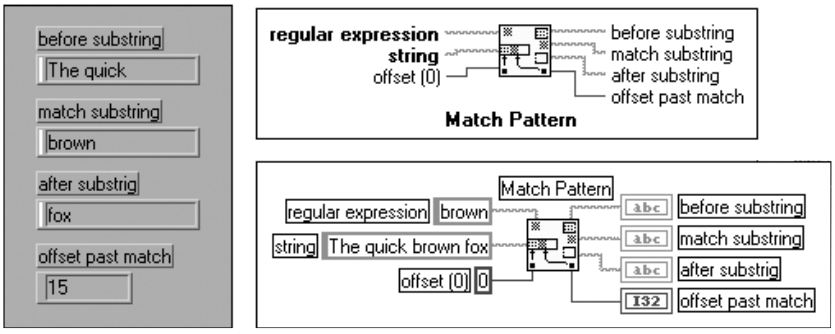
دستور String Subset زیرمجموعه‌ای از رشته‌ی ورودی را به‌دست می دهد که محل اولین کاراکتر آن در ورودی offset و طول کاراکترهای این زیررشته در ورودی length مشخص می گردد.




**شکل ۱۵-۹: نمونه‌ای از کاربرد دستور String Subset**

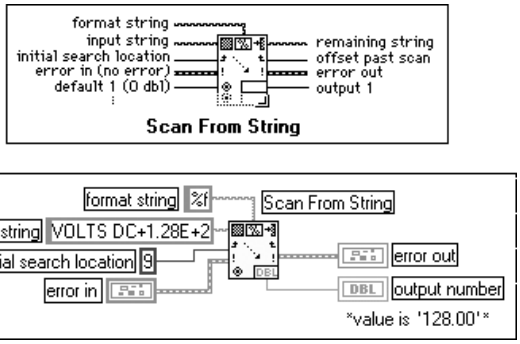
همان گونه که در شکل ۱۵-۹ ملاحظه می کنید در ورودیهای length و offset به ترتیب دو عدد 2 و 6 مشخص شده است. بدین ترتیب ششمین کاراکتر از رشته‌ی ورودی، اولین کاراکتر از رشته‌ی خروجی خواهد بود. به علاوه، رشته‌ی خروجی تنها شامل ۲ کاراکتر است. با توجه به کاراکترهای رشته‌ی ورودی و اعداد مندرج در ورودیهای length و offset، زیررشته‌ی DC در خروجی ظاهر می شود.

دستور Match Pattern جهت جستجو کردن یک الگوی مطلوب از کاراکترهای موجود در یک رشته به کار برده می شود. این دستور الگوی منطبق با الگوی مطلوب را جستجو می کند و در صورت وجود، آن را به دست می دهد. دستور مذکور کاراکترهای مشخص شده در ورودی reguler expression در رشته‌ی ورودی string را جستجو می کند. این الگو با کاراکتری که در ورودی offset نشان داده شده است آغاز می گردد. در صورتی که الگوی مطلوب پیدا شود، این دستور رشته را به سه زیررشته‌ی after substring, match, offset تقسیم می کند. در غیر این صورت در زیررشته‌ی match substring هیچ کاراکتری درج نمی‌شود و در خروجی offset past match مقدار «-1» ظاهر می‌شود. در شکل ۱۶-۹ آیکن دستور Match Pattern را به همراه یک نمونه از کاربرد آن ملاحظه می کنید.




**شکل ۱۶-۹: آیکن دستور Match Pattern به همراه اتصالات و نمونه‌ای از کاربرد آن**

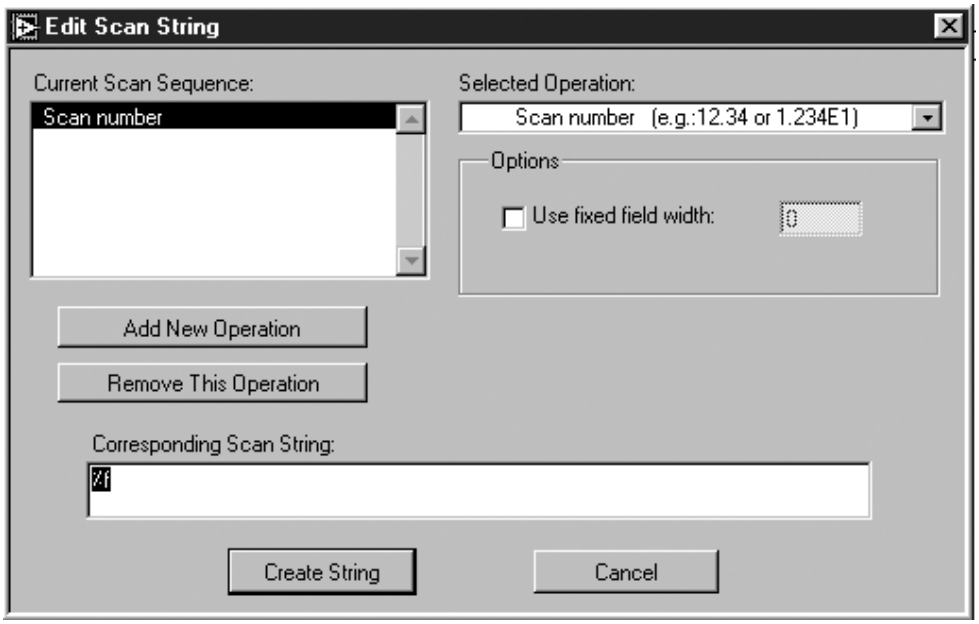
دستور Scan From String برخلاف دستور Format Into String رشته ای را که شامل کاراکترهای عددی است به داده های عددی تبدیل می کند. دستور مذکور، رشته‌ی ورودی را از محلی که در ورودی initial search location آغاز می شود بررسی می کند و مطابق با فرمت مشخص شده در ورودی format string، رشته‌ی ورودی را به اعداد تبدیل می نماید. می توان اندازه‌ی این دستور را برای تبدیل همزمان چند مقدار تغییر داد. در شکل ۱۷-۹ آیکن دستور Scan From String را به همراه یک نمونه از کاربرد آن ملاحظه می کنید.




**شکل ۱۷-۹: آیکن دستور Scan From String به همراه اتصالات و نمونه‌ای از کاربرد آن**

در شکل ۹-۱۷ دستور Scan From String، رشته‌ی «VOLTS DC+1.28E+2» را به عدد 128.00 تبدیل می کند. این دستور تبدیل داده های رشته‌ای به عددی را از کاراکتر شماره‌ی 8 که در اینجا کاراکتر «+» است آغاز می کند. همان گونه که در شکل ۹-۱۷ ملاحظه می کنید، عدد 9 به ورودی initial seareh location نسبت داده شده است و به دلیل اینکه offset اولین کاراکتر صفر است، کاراکتر شماره‌ی 8 یعنی «+» آغازگر زیررشته خواهد بود.

در مورد هریک از دو دستور Format Into String و Scan From String یک پنجره‌ی محاوره‌ای در نظر گرفته شده است که از طریق آن می توانید فرمت رشته را برای ورودی format string تعیین نمایید. در این پنجره‌ی محاوره‌ای می توانید فرمت، دقت، نوع داده و طول مقدار تبدیل شده را تعیین کنید. برای دستیابی به پنجره های مذکور بر روی دستور مربوط دو بار کلیک نمایید یا منوی کرکره‌ای را بر روی دستور باز کرده، گزینه‌ی Edit Format String را جهت دسترسی به پنجره های Edit Scan String یا Edit Format String انتخاب کنید. در شکل ۹-۱۸ پنجره‌ی محاوره‌ای Edit Scan String نشان داده شده است.



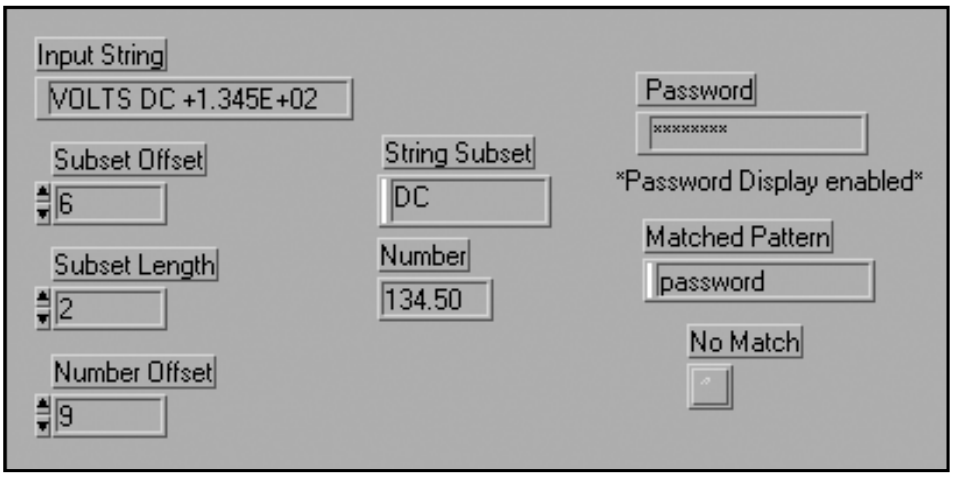

**شکل ۹-۱۸: پنجره‌ی محاوره‌ای Edit Scan String**

پس از تعیین فرمت رشته و کلیک کردن بر روی دکمه‌ی Create String در این پنجره‌ی محاوره‌ای، یک رشته با فرمت مورد نظر ایجاد شده و به ورودی format string اعمال خواهد شد.

## تمرین ۹-۲: تجزیه‌ی رشته ها

در این تمرین قصد داریم با جدا کردن زیرمجموعه‌ای از یک رشته، داده های یک رشته‌ی طولانی را تجزیه نماییم و کاراکترهای عددی موجود در رشته را به مقادیر عددی تبدیل کنیم.

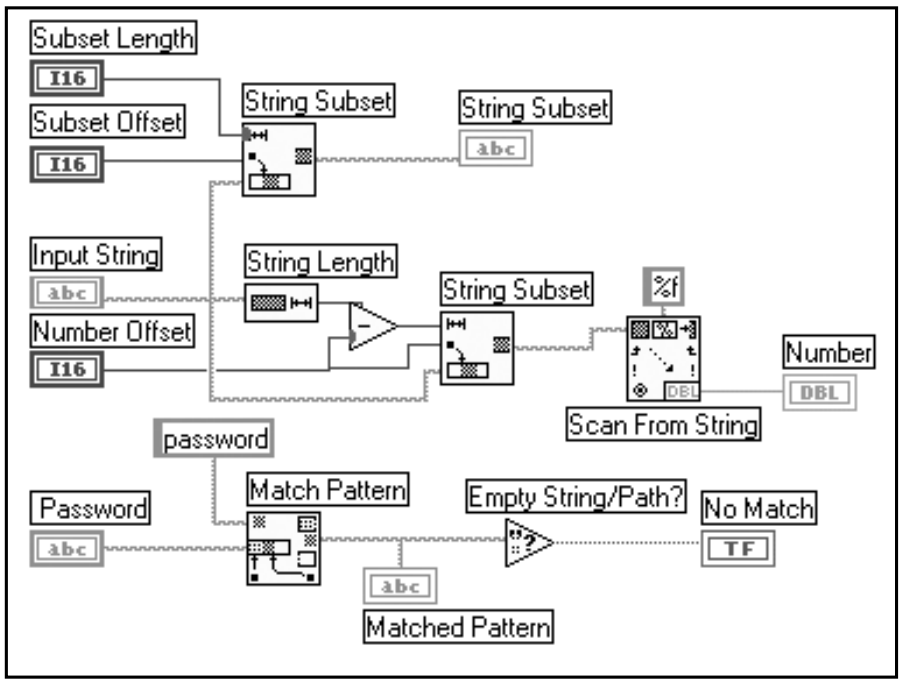
۱- یک برنامه‌ی جدید باز نمایید و صفحه‌ی پانل آن را مطابق شکل ۹-۱۹ ایجاد کنید.




**شکل ۱۹-۹: صفحه‌ی پانل در برنامه‌ی Parse String.vi**

۲- با انتخاب گزینه‌ی Password Display از منوی کرکره‌ای رشته، رشته‌ای با برچسب Password ایجاد کنید که در آن تنها کاراکترهای «*» به نمایش در آید.

۳- صفحه‌ی نمودار بلوکی این برنامه را مطابق شکل ۲۰-۹ ایجاد کنید.




**شکل ۲۰-۹: صفحه‌ی نمودار بلوکی در برنامه‌ی Parse String.vi**

در ادامه به بررسی برخی از دستورها و توابع موجود در صفحه‌ی نمودار بلوکی این برنامه می پردازیم:

String Subset Function

دستور String Subset (Functions >> String) یک زیررشته با تعداد کارکترهای معین را از رشته‌ی ورودی جدا می کند. این زیررشته از کاراکتری که در ورودی offset مشخص شده است آغاز می گردد.

دستور Scan From String (Functions >> String) رشته‌ی حاوی کاراکترهای عددی (نظیر E, e, -, +, 0 الی 9) را به داده های عددی تبدیل می کند.

دستور Match Pattern (Functions >> String) رشته‌ی وارد شده توسط کاربر را با رشته‌ی داده شده مقایسه می کند. در صورتی که دو رشته‌ی مذکور یکسان باشند، کلمه‌ی عبور به نمایش در می آید. در غیر این صورت نشان دهنده‌ی رشته‌ای فاقد رشته است و یک رشته‌ی خالی را نمایش می دهد.

دستور Empty String/Path? (Functions >> Comparison) در صورتی که از خروجی match substring در دستور Match Pattern یک رشته‌ی خالی دریافت کند، مقدار جبری True را در خروجی خود به دست می دهد.

دستور String Length (Functions >> String) تعداد کاراکترهای رشته را به دست می دهد.

۴- اکنون مقادیر موجود درصفحه‌ی پانل را مطابق شکل ۱۹-۹ وارد نموده و برنامه را اجرا کنید. توجه کنید که چگونه زیررشته‌ی «DC» از رشته‌ی ورودی جدا می شود. همچنین دقت کنید که کاراکترهای عددی در رشته به عدد تبدیل می‌شوند و در بخش Number به نمایش در می آیند. حال مقادیر دیگری را به رشته‌ی ورودی نسبت دهید و در صفحه‌ی پانل وارد کنید. سپس برنامه را مجدداً اجرا نمایید. این مطلب را به خاطر داشته باشید که در رشته ها نیز همانند آرایه ها همواره اندیس اولین المان 0 است.

در صورتی که فاصله‌ی کاراکترهای موجود در رشته‌ی Input String مطابق شکل ۱۹-۹ نباشد کاراکترهای مورد نظر در String Subset به نمایش در نمی آیند.

همچنین دقت کنید که چگونه رشته‌ی Password تنها کاراکترهای «*» را نشان می‌دهد. دستور Match Pattern کلمه‌ی عبور وارد شده توسط کاربر را با رشته‌ی داده شده که در اینجا شامل کاراکترهای «password» می باشد مقایسه می‌کند و در صورت تطابق آنها، یک رشته‌ی مطابق به‌دست می‌دهد. در غیر این صورت حاصل این دستور یک رشته‌ی خالی خواهد بود.

۵- این برنامه را با عنوان Parse String.vi در زیرفهرست LabVIEW\My Activity ذخیره کنید، سپس آن را ببندید.

## برنامه های ورودی/خروجی فایل (File I/O)

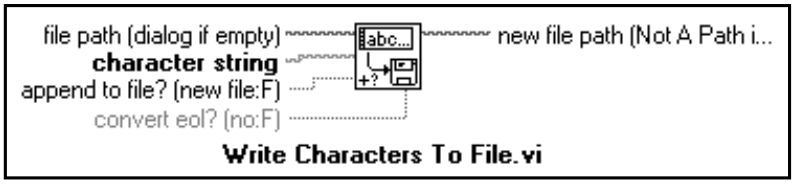
در بسته‌ی نرم‌افزاری LabVIEW برنامه های بسیار زیادی در مورد ورودی/خروجی فایل وجود دارد. در این فصل این برنامه ها را مورد بررسی قرار می دهیم. برنامه های مذکور در زیرپالت Functions >> File I/O قرار دارند.

در برنامه های ورودی/خروجی فایل، یک ورودی برای تعیین مسیر فایل در نظر گرفته شده است. این مسیر شباهت بسیاری به یک رشته دارد. همان گونه که قبلاً ذکر شد، مسیر نوع خاصی از داده است که برای تعیین

محل ذخیره‌ی فایل به‌کار می‌رود. در مورد مسیرها به اختصار در فصل ۴ توضیحاتی ارائه شد. در فصل ۱۴ مفهوم و کاربرد مسیر را به طور مفصل بررسی خواهیم نمود. در صورتی که مسیر فایل را در این دستورها سیم‌کشی نکرده باشید، یک پنجره‌ی محاوره‌ای باز خواهد شد که از شما می‌خواهد تا نام یک فایل را وارد نمایید و یا آن را از لیست فایل‌های موجود انتخاب کنید. در زمان فراخوانی، برنامه‌های ورودی/خروجی فایل، یک فایل را باز یا آن را ایجاد می‌کنند. سپس داده‌ها را از آن می‌خوانند یا در آن می‌نویسند و پس از اتمام عملیات، آن را می‌بندند. تمامی فایل‌های ایجاد شده از نوع فایل‌های متنی عادی هستند.

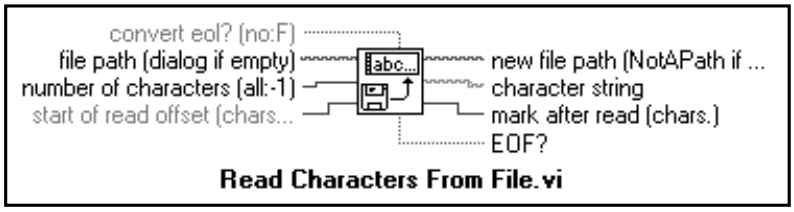
پس از ذخیره‌ی داده‌ها در یک فایل می‌توانید با به‌کارگیری یکی از برنامه‌های واژه‌پرداز یا صفحه‌گسترده نظیر Microsoft Word یا Microsoft Excel فایل مذکور را بازکنید و داده‌ها را ملاحظه نمایید. در اکثر فایل‌های صفحه‌گسترده، کلید <Tab> و کاراکترهای EOL به ترتیب برای جدا کردن ستونها و سطرها مورد استفاده قرار می‌گیرند. دو برنامه‌ی Write To Spreadsheet File.vi و Read Form Spreadsheet File.vi در مورد فایل‌های با فرمت صفحه‌گسترده به‌کار برده می‌شوند. در ادامه‌ی بحث برخی از برنامه‌های موجود در زیرپالت File I/O را مورد بررسی قرار می‌دهیم:

برنامه‌ی Write Characters To File.vi رشته ای از کاراکترها را در داخل یک فایل جدید می‌نویسد یا آن را به یکی از فایل‌های موجود الحاق می‌کند.



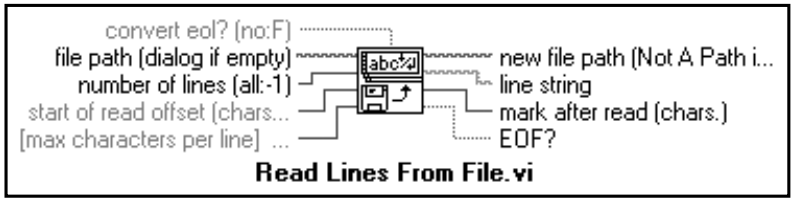

**شکل ۲۱-۹: آیکن برنامه‌ی Write Characters To File.vi به همراه اتصالات آن**

برنامه‌ی Read Characters From File.vi تعداد معینی از کاراکترهای یک فایل را قرائت می‌کند. خواندن داده‌ها از کاراکتری که در ورودی start of read offset مشخص شده است آغاز می‌گردد.



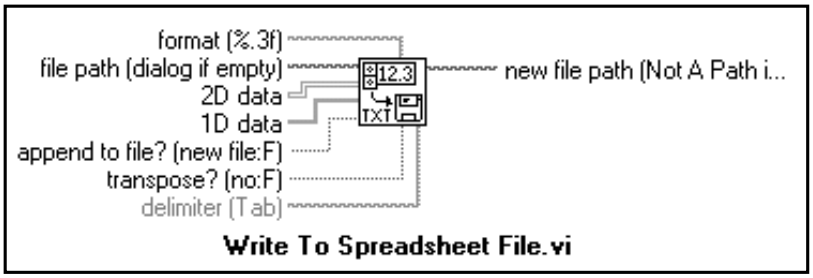

**شکل ۲۲-۹: آیکن برنامه‌ی Read Characters From File.vi به همراه اتصالات آن**

برنامه‌ی Read Lines From File.vi تعداد معینی از خطوط یک فایل را قرائت می‌کند. عملیات قرائت از کاراکتری که در ورودی start of read offset مشخص شده است آغاز می‌گردد.



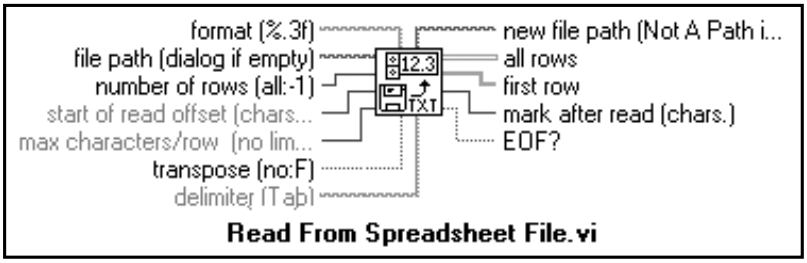

**شکل ۲۳-۹: آیکن برنامه‌ی Read Lines From File.vi به همراه اتصالات آن**

برنامه‌ی Write To Spreadsheet File.vi یک آرایه‌ی یک‌بعدی یا دوبعدی حاوی مؤلفه‌های اعشاری را به داده‌های رشته‌ای متنی تبدیل می‌نماید، سپس رشته‌ی مذکور را در یک فایل جدید می‌نویسد یا آن را به یک فایل موجود الحاق می‌کند. در صورت تمایل می‌توانید آرایه‌ی ورودی را ترانهاده کنید. شایان ذکر است که اتصال همزمان هر دو آرایه‌ی 1D و 2D امکان‌پذیر نیست. در صورت انجام این عمل یکی از این دو آرایه نادیده فرض می‌شود. خواندن فایل‌های متنی ایجاد شده توسط این برنامه در اکثر نرم‌افزارهای واژه‌پرداز و صفحه گسترده نظیر Microsoft Word و Microsoft Excel امکان‌پذیر است.




**شکل ۲۴-۹: آیکن برنامه‌ی Write To Spreadsheet File.vi به همراه اتصالات آن**

برنامه‌ی Read Form Spreadsheet File.vi فایل‌های صفحه گسترده را که در فرمت متنی ذخیره شده‌اند می‌خواند. خواندن داده‌ها از کاراکتری که در ورودی start of read offset مشخص شده است آغاز می‌گردد. سپس داده‌ها را به یک آرایه‌ی 2D که حاوی مؤلفه‌های اعشاری با دقت یک رقم اعشار است تبدیل می‌کند. در صورت تمایل می‌توانید آرایه‌ی ورودی را ترانهاده کنید.




**شکل ۲۵-۹: آیکن برنامه‌ی Read Form Spreadsheet File.vi به همراه اتصالات آن**

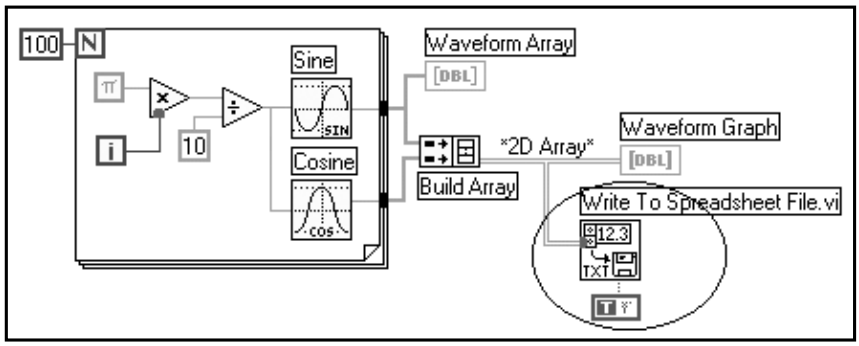
دستورهای مذکور بسیار سطح بالا بوده، ولی از لحاظ کاربرد بسیار ساده هستند. در نرم‌افزار LabVIEW دستورهای پیچیده‌تر دیگری نیز وجود دارند که در فصل ۱۴ مورد بررسی قرار می‌گیرند.

## تمرین ۳-۹: نوشتن در یک فایل صفحه گسترده

در این تمرین قصد داریم تا با ایجاد تغییر در یکی از برنامه‌های موجود، داده‌ها را در یک فایل جدید با فرمت ASCII ذخیره نماییم. پس از آن می‌توانیم از طریق برنامه‌های صفحه گسترده به محتویات این فایل دسترسی پیدا کنیم.

۱- برنامه‌ی Graph Sine Array.vi را که در فصل ۸ ایجاد نمودید باز کنید. اگر برنامه‌ی مذکور را ذخیره نکرده‌اید می‌توانید این برنامه را در مسیر LabVIEW\Exercise\Ch8.llb بیابید. در صورتی که به یاد داشته باشید این برنامه پس از ایجاد دو آرایه، آنها را بر روی یک گراف ترسیم می‌کرد. در این تمرین باید برنامه‌ی مذکور را به گونه‌ای تغییر دهید تا دو آرایه در یک فایل نوشته شوند؛ به صورتی که هر ستون شامل یک آرایه باشد. به بیان دیگر هر یک از آرایه‌ها در یک ستون ذخیره گردد.

۲- حال با اضافه نمودن بخش نشان داده شده در شکل ۲۶-۹، صفحه‌ی نمودار بلوکی این برنامه را بازسازی کنید.




**شکل ۲۶-۹: صفحه‌ی نمودار بلوکی در برنامه‌ی Graph Sine Array to File.vi**

در ادامه به بررسی بخش اضافه شده در شکل ۲۶-۹ می‌پردازیم:

Write to Spreadsheet File VI

برنامه‌ی Write to Spreadsheet File.vi (Functions >> File I/O) آرایه‌ی دوبعدی را به یک رشته‌ی صفحه گسترده تبدیل می‌کند، سپس آن را در یک فایل می‌نویسد. در صورتی که هیچ مسیری مشخص نشده باشد (همان گونه که در این تمرین اعمال شده است) یک پنجره‌ی محاوره‌ای باز خواهد شد و از کاربر نام فایلی که قرار است داده‌های رشته‌ای در آن نوشته شود درخواست می‌گردد.

Boolean Constant

ثابت جبری یا Boolean Constant (Functions >> Boolean) ترانهاده بودن آرایه‌ی دوبعدی را قبل از نوشته شدن در داخل فایل کنترل می‌کند. برای تغییر وضعیت این ثابت جبری به حالت True، توسط ابزار Operatig Tool بر روی آن کلیک کنید. همان گونه که می‌دانید این المان به صورت پیش فرض در حالت False قرار دارد. در صورتی که این المان در وضعیت True باشد بدین معنی است که قصد دارید ترانهاده‌ی آرایه‌ی مورد نظر را در برنامه استفاده کنید. به دلیل اینکه آرایه‌های حاوی داده‌ها به صورت سطری قرار گرفته‌اند و در برنامه‌ی جدید قصد دارید که هر ستون از فایل صفحه گسترده حاوی داده‌های لازم برای ترسیم یک شکل موج باشد، ابتدا باید آرایه‌ی 2D را ترانهاده کنید.

۳- به صفحه‌ی پانل بازگشته و برنامه را اجرا کنید. پس از ایجاد آرایه‌های حاوی داده‌ها، یک پنجره‌ی محاوره‌ای باز می‌شود و نام فایل جدیدی که قصد دارید داده‌ها در آن نوشته شوند درخواست

می شود. یک نام دلخواه را به عنوان مثال Sine & Cosine Wave Data.dat در محل در نظر گرفته شده تایپ و سپس بر روی دکمه‌ی OK کلیک کنید. نام و محل ذخیره‌ی فایل را به خاطر بسپارید، زیرا در تمرین بعدی قصد داریم این داده ها را از آن بخوانیم.

از ذخیره نمودن فایل‌های حاوی داده در کتابخانه‌ها خودداری کنید. زیرا این عمل منجر به نوشتن داده‌ها بر روی برنامه‌های قبلی می‌شود و مسلماً برنامه‌های قبلی شما را نابود خواهد کرد.

۴- این برنامه را با عنوان Graph Sine Array to File.vi در زیرفهرست LabVIEW\ My Activity ذخیره کنید و سپس آن را ببندید.

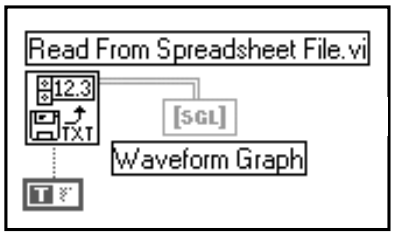
۵- در صورت دسترسی به نرم‌افزارهای Word یا Excel از آنها استفاده کنید و فایلی را که اخیراً ایجاد نموده‌اید باز کنید. باید دو ستون مجزا را که هر یک حاوی ۱۰۰ المان است مشاهده کنید.

## تمرین ۴-۹: خواندن یک فایل صفحه گسترده

در این تمرین قصد داریم تا با ایجاد یک برنامه، داده ها را از فایل ایجاد شده در تمرین ۳-۹ بخوانیم و آنها را بر روی یک گراف رسم کنیم.

۱- یک برنامه‌ی جدید باز کنید و در صفحه‌ی پانل آن یک Waveform Graph قرار دهید. از فعال بودن سیستم درجه بندی خودکار در گراف اطمینان حاصل کنید.

۲- صفحه‌ی نمودار بلوکی این برنامه را مطابق شکل ۲۷-۹ ایجاد کنید.



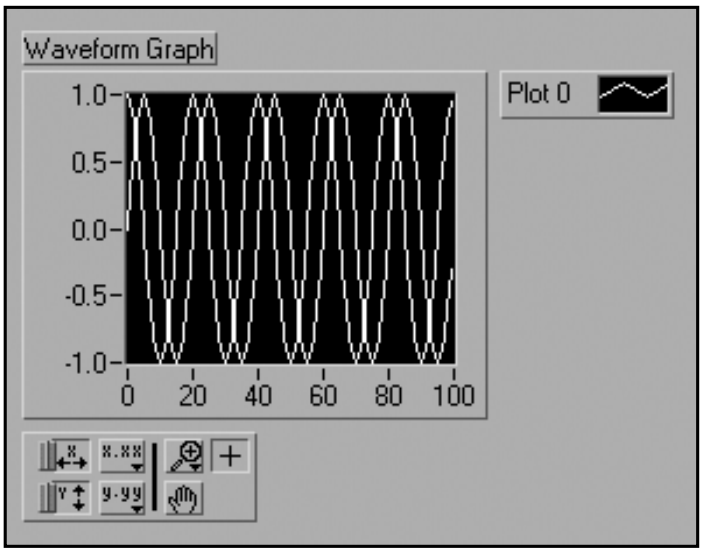

**شکل ۲۷-۹: صفحه‌ی نمودار بلوکی در برنامه‌ی Read File.vi**

همان گونه که ملاحظه می کنید برای خواندن داده ها و ترسیم آنها بر روی گراف از (زیر) برنامه‌ی Read Form Spreadsheet File.vi استفاده شده است. توجه داشته باشید که به محض ایجاد Waveform Graph در صفحه‌ی پانل، حروف DBL در ترمینال متناظر با آن در صفحه‌ی نمودار بلوکی دیده می شود اما به محض سیم کشی آرایه‌ی 2D به آن، حروف SGL در این ترمینال به نمایش در می آید.

۳- ثابت جبری یا Boolean Constant را در وضعیت True قرار دهید. به دلیل اینکه داده ها در فایل به صورت ستونی ذخیره شده ولی در گراف، داده ها به صورت سطری ترسیم می شوند، حتماً با قرار دادن ثابت جبری در وضعیت True، آرایه را ترانهاده کنید. در صورتی که در تمرین قبل برای

ذخیره نمودن داده ها به صورت ستونی، آرایه را ترانهاده نکرده اید، اکنون نیز ترانهاده کردن مجدد این آرایه ضروری نیست.

۴- برنامه را اجرا کنید. به دلیل اینکه مسیر فایل را مشخص نکرده اید، یک پنجره‌ی محاوره‌ای باز می‌شود و نام فایل درخواست می گردد. در این مرحله نام فایل ایجاد شده در تمرین ۳-۹ را وارد کنید یا منوی کرکره‌ای را بر روی ورودی file path در آیکن برنامه‌ی Read Form Spreadsheet File.vi باز کنید و پس از انتخاب گزینه‌ی Create Constant، نام فایل را وارد کنید. این برنامه ابتدا داده ها را از فایل مورد نظر قرائت می‌نماید، سپس هر دو شکل موج را بر روی گراف ترسیم می کند.




**شکل ۲۸-۹: دو شکل موج ترسیم شده بر روی گراف در صفحه‌ی پانل برنامه‌ی Read File.vi**

۵- این برنامه را با عنوان Read File.vi در زیرفهرست LabVIEW\My Activity ذخیره کنید.

## خلاصه

۹

در بسته‌ی نرم افزاری LabVIEW دستورهای زیادی جهت انجام عملیات بر روی رشته ها تعبیه شده است. این دستورها در زیرپالت Functions >> String قرار دارند. به کمک این دستورها می توانید تعداد کاراکترهای رشته را تعیین نموده، دو رشته را با هم ترکیب کنید و آنها را به صورت زنجیره در آورید. همچنین می توانید زیرمجموعه‌ای از رشته را جدا نمایید و داده های عددی را به رشته‌ای (و برعکس) تبدیل کنید.

با به کارگیری دستورهای موجود در زیرپالت Functions >> File I/O می توانید داده ها را از یک فایل بخوانید یا بر روی آن بنویسید. برنامه‌ی Write Characters To File.vi رشته‌ی متنی را در فایل ذخیره می کند. دو برنامه‌ی Read Characters From File.vi و Read Lines From File.vi داده‌ها را از فایل می‌خوانند. برای نوشتن داده ها در یک فایل صفحه گسترده، برنامه‌ی Write To Spreadsheet File.vi را به کار برید. با استفاده ازبرنامه‌ی Read From Spreadsheet File.vi می توانید این داده ها را خوانده و آنها را به فرمت عددی تبدیل نمایید.

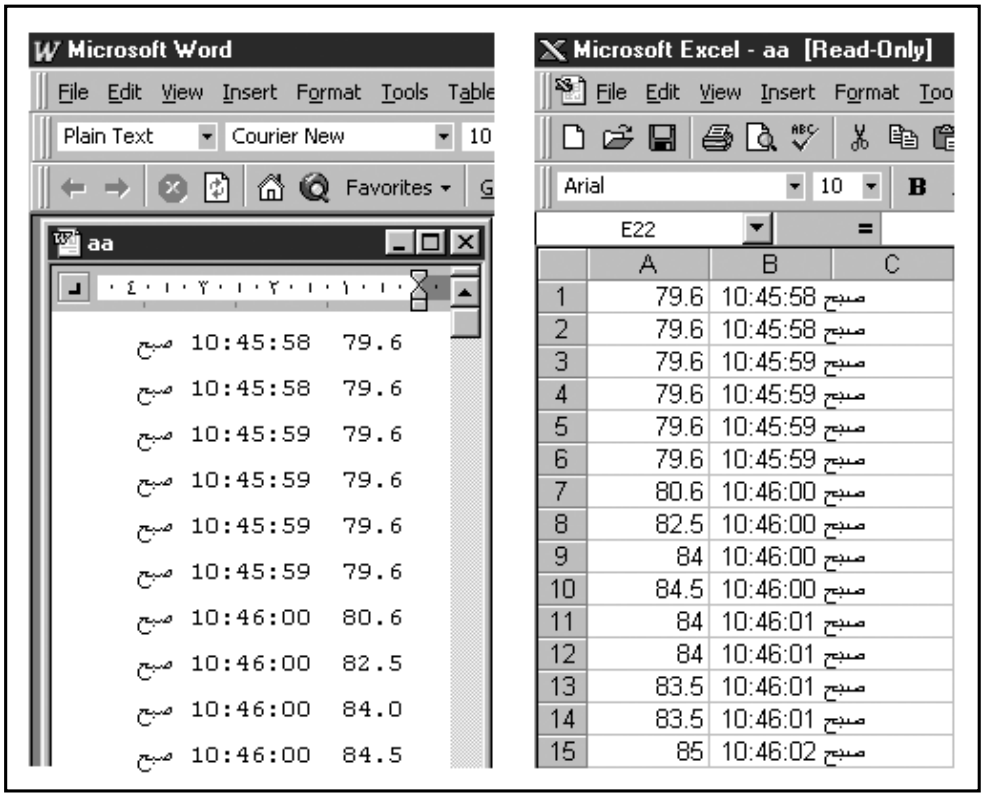
به شما تبریک می گوییم. اکنون با اصول برنامه نویسی LabVIEW به خوبی آشنا شده‌اید. بخش پیشرفته‌ی کتاب که از فصل بعدی آغاز می گردد ویژگیها و قابلیتهای پیچیده‌ی بسته‌ی نرم‌افزاری LabVIEW را آموزش می دهد. به کمک قابلیتهای مذکور، برنامه نویسی بسیار ساده تر خواهد شد.

## تمرینهای اضافی

### تمرین ۵-۹: اندازه گیری درجه حرارت و ثبت زمان اندازه گیری

برنامه‌ای ایجاد کنید که به کمک یک حلقه، ۵۰ مرتبه درجه حرارت را اندازه گیری کند، سپس نمونه ها را بر روی نمودار نمایش دهد. در مرحله‌ی بعد، داده ها را به رشته تبدیل نماید و کاراکتر Tab، زمان اندازه گیری داده ها و کاراکتر End of Line را به صورت زنجیره در آورد. فاصله بین هر دو نمونه، ۲۵۰ میلی ثانیه است. این برنامه تمامی داده ها را در یک فایل ذخیره می کند. برنامه‌ی مذکور را با عنوان Temperature Log.vi در زیرفهرست LabVIEW\My Activity ذخیره کنید.

با استفاده از برنامه های Word یا Excel می توانید محتویات فایل را بررسی کنید. آنچه مشاهده می کنید باید تقریباً مطابق شکل ۲۹-۹ باشد.




**شکل ۲۹-۹: محتویات ذخیره شده در فایل که توسط دو محیط Word و Excel به نمایش درآمده است.**

ثابتهای Tab و End of Line در زیرپالت String قرار دارند.

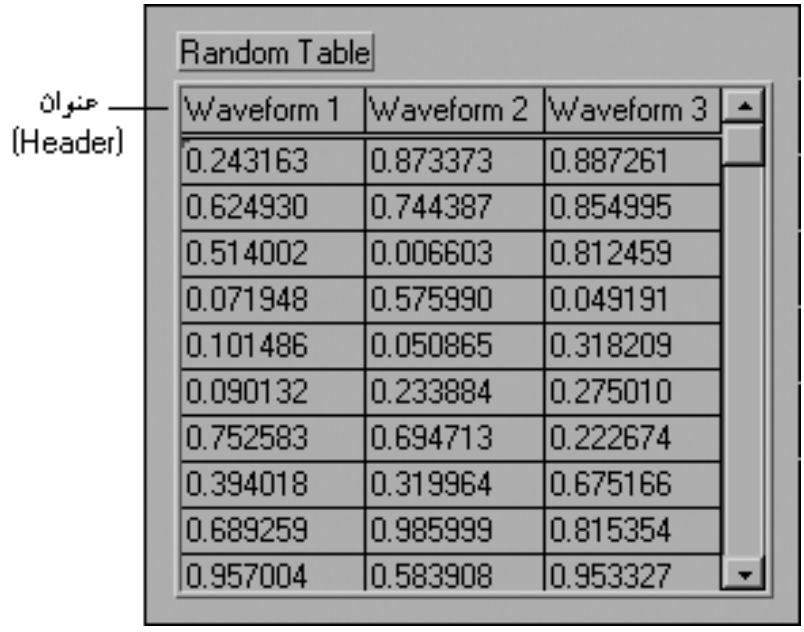
برای پیوند دادن تمامی رشته‌ها، از دستور Concatenate Strings استفاده کنید.

برای ذخیره نمودن داده‌ها، از برنامه‌ی Write Characters To File.vi استفاده نمایید.

می‌توانید در هر لحظه یک سطر از داده‌ها را در فایل بنویسید. اما بهتر است که با استفاده از ثبات انتقال و دستور Concatenate Strings ابتدا تمامی داده‌ها را در یک رشته‌ی طولانی جمع‌آوری کنید، سپس آنها را یک‌جا در یک فایل بنویسید.

## تمرین ۶-۹: تمرین بیشتر با فایل صفحه‌گسترده

برنامه‌ای ایجاد کنید که یک آرایه‌ی دوبعدی تولید نماید، سپس ترانهاده‌ی آن را در یک فایل صفحه‌گسترده ذخیره کند. آرایه‌ی مورد نظر یک ماتریس ۱۰×۳ است و مؤلفه‌های آن نیز اعداد تصادفی هستند. همان گونه که در شکل ۳۰-۹ ملاحظه می‌کنید در این فایل برای هر یک از ستونها عنوان[2] جداگانه‌ای در نظر گرفته شده است. برای انجام این تمرین، دستورهای موجود در زیرپالت‌های String و File I/O را به کار برید. این برنامه را با عنوان Spreadsheet Exercise.vi در زیرفهرست LabVIEW\My Activity ذخیره کنید.




**شکل ۳۰-۹: نمای ظاهری صفحه‌ی پانل برنامه‌ی Spreadsheet Exercise.vi پس از اجرای برنامه**

برای درج عنوان در بالای هر یک از ستونها، از برنامه‌ی Write Characters To File.vi استفاده کنید. سپس برای نوشتن داده‌های عددی در همان فایل از برنامه‌ی Write to Spreadsheet File.vi کمک بگیرید.

2- Header

# ۱۰

# جمع آوری داده‌ها

## در این فصل با موارد زیر آشنا می‌شوید:

- سخت‌افزارهای موجود جهت جمع‌آوری داده‌ها
- فرضیه‌های موجود در مورد سیگنال‌ها نظیر طبقه‌بندی و روشهای مختلف اندازه‌گیری سیگنال‌ها، تطبیق دادن و نمونه‌برداری آنها
- نحوه‌ی انتخاب و نصب کارت DAQ
- عملکرد سیستم GPIB
- درگاههای سریال در رایانه و روش استفاده از آنها در فرآیند DAQ

# LabVIEW

۱۰

## جمع آوری داده‌ها

در این فصل به مطالب عنوان شده در فصل ۲ نگاهی عمیق‌تر خواهیم داشت. علت اصلی استفاده از نرم‌افزار LabVIEW، قابلیت جمع‌آوری داده‌ها یا Data Acquisition است. همان‌طور که می‌دانید به کمک این بسته‌ی نرم‌افزاری و با جمع‌آوری داده‌ها از دنیای خارج از رایانه توسط کارت‌های DAQ، GPIB و... می‌توان رایانه را به یک سیستم اندازه‌گیری مجازی تبدیل نمود.

در این فصل ابتدا به بررسی حالات مختلف جهت ایجاد یا جمع‌آوری داده‌ها می‌پردازیم، سپس نحوه‌ی به‌کارگیری ابزارها و کارت‌های DAQ، GPIB و درگاههای سریال را مورد بحث قرار می‌دهیم. همچنین با برخی از فرضیه‌های موجود در مورد سیگنال‌ها و نوع سخت‌افزار لازم برای سیستم‌های اندازه‌گیری مختلف آشنا خواهید شد.

در این فصل اهداف زیر دنبال می‌شوند:

- آشنایی با سخت‌افزارهای موجود جهت جمع‌آوری داده‌ها.
- آشنایی و فراگیری برخی از فرضیه‌های موجود در مورد سیگنال‌ها نظیر طبقه‌بندی و روشهای مختلف اندازه‌گیری سیگنال‌ها، تطبیق دادن و نمونه‌برداری آنها.
- فراگیری نحوه‌ی انتخاب و نصب کارت DAQ.
- درک عملکرد سیستم GPIB.
- بررسی درگاههای سریال در رایانه و روش استفاده از آنها در فرآیند DAQ.

اصلاحات کلیدی موجود در این فصل عبارتند از:

- جمع آوری داده ها (DAQ)
- باس ارتباطی همه منظوره (GPIB)
- درگاههای سریال (Serial Ports)
- سیگنال متصل شده به زمین (Grounded Signal)
- سیگنال شناور (Floating Signal)
- مرجع زمین (Ground Reference)
- نرخ نمونه برداری (Sampling Rate)
- فرکانس نایکوییست (Nyquist Frequency)
- تطبیق دادن سیگنال (Signal Conditioning)
- نرم افزار راه انداز کارت های DAQ
- روش اندازه گیری تفاضلی (Differential Measurement)
- روش اندازه گیری RSE
- روش اندازه گیری NRSE

## توضیحاتی در مورد حروف و کلمات اختصاری

«با ما همراه شوید تا با به کارگیری ابزار VHDL و با استفاده از استاندارد جدید MXR، به طراحی کارت رابط PCMCIA بپردازیم.»

حتماً تا به حال در جلسه یا کنفرانسی حضور داشته اید که در آن جلسه، شخصی در حین صحبت در یک بحث تخصصی از حروف اختصاری (نظیر حروف و کلمات موجود در جمله‌ی قبل) استفاده نموده و همه‌ی حضار وانمود کرده‌اند که مفهوم آنها را درک نموده اند. زیرا هیچ کس تمایل ندارد که در میان جمع سئوال کند که به عنوان مثال حروف اختصاری MXR مخفف چه کلمات یا عباراتی است.

حال فرصتی پیش آمده است تا مفهوم تمامی حروف و علائم اختصاری موجود در این فصل را فرا بگیرید و دریابید که این حروف اختصاری مخفف چه کلماتی هستند و یا به جای چه عباراتی مورد استفاده قرار می گیرند. در ادامه، فهرستی از حروف اختصاری موجود در این فصل را ملاحظه می کنید که در مقابل هر یک توضیح مربوط آورده شده است.

AC (Alternaing Current): جریان متناوب. این حروف اختصاری اصولاً به جریان الکتریکی در یک سیستم اطلاق می شود. به بیان ساده می توان گفت که جریان AC همان جریان موجود در پریز دیوار اتاق است و DC همان جریان باطری هاست. اما امروزه حروف اختصاری AC به هر نوع سیگنالی (نه فقط جریان الکتریکی) که نسبت به زمان، به سرعت تغییر کند اطلاق می گردد.

ADC (Analog-to-Digital Conversion): مبدل سیگنال آنالوگ به دیجیتال. این مبدل، یک سیگنال خارجی آنالوگ را به عنوان ورودی دریافت می‌نماید و در خروجی، آن را به صورت مجموعه‌ای از بیت ها تبدیل می کند. این بیت ها برای سیستم های دیجیتالی نظیر رایانه قابل درک هستند. در اغلب موارد این مبدل به طور اختصار با A/D نشان داده می‌شود و تراشه‌ای که برای انجام این تبدیل به کار می رود، ADC خوانده می شود.

DAQ (Data AcQuisition): این عبارت به طور کلی به جمع‌آوری داده‌ها اطلاق می‌گردد و معمولاً همراه با مبدل A/D به کار برده می‌شود. برخی مواقع نظیر موارد استفاده شده در این کتاب عبارت DAQ در مورد تولید داده‌ها نیز به‌کار برده می‌شود. مراقب باشید که عبارت DAQ را با عبارت DAC اشتباه نگیرید.

DAC (Digital-to-Analog Conversion) یا D/A به مبدلی گفته می‌شود که سیگنال دیجیتال را به آنالوگ تبدیل می‌کند. معمولاً تراشه‌ای که برای انجام این تبدیل به کار می‌رود، DAC خوانده می‌شود.

DC (Direct Current): جریان مستقیم. این نوع سیگنال برخلاف AC است. برخی مواقع این عبارت در مواردی به کار برده می‌شود که سیگنال مورد نظر ثابت و فرکانس آن صفر باشد. در موارد دیگر نظیر بحث DAQ، عبارت DC برای سیگنال‌هایی با فرکانس خیلی پایین (کمتر از یک ثانیه) نیز به کار برده می‌شود.

DMA (Direct Memory Access): دسترسی مستقیم به حافظه DMA. به شما اجازه می‌دهد تا داده‌های جمع‌آوری شده را مستقیماً به حافظه‌ی RAM در رایانه‌ی خود انتقال دهید. واضح است که در این حالت سرعت انتقال داده‌ها افزایش می‌یابد. بدون وجود DMA، انتقال داده‌ها به حافظه با زمان طولانی‌تر و طی مراحل بیشتری انجام می‌گیرد. جهت جمع‌آوری داده‌ها می‌توانید از کارت‌های DAQ که دارای DMA داخلی هستند استفاده کنید و یا یک کارت مجزای DMA تهیه نمایید.

GPIB (General Purpose Interface Bus): باس ارتباطی همه منظوره. این باس که برخی مواقع با HP-IB[1] یا باس IEEE 488.2[2] نیز خوانده می‌شود، به صورت یک باس ارتباطی استاندارد جهانی برای ابزارها و دستگاههایی که با رایانه در ارتباط هستند به‌کار برده می‌شود. این باس برای اولین بار در دهه‌ی ۱۹۶۰ توسط شرکت Hewlet Packard جهت برقراری ارتباط بین ابزارهای اندازه‌گیری و رایانه ابداع گردید. باس استاندارد IEEE نیز به همراه باس مذکور به کار برده می‌شود و کمک می‌کند تا پروتکل‌های سخت‌افزاری خاص بتوانند با رایانه ارتباط برقرار سازند.

MXI (Multisystem eXtension Interface): یک سیستم ارتباطی استاندارد جهت اتصال رایانه و جعبه‌های VXI به کارت‌های GPIB است.

RS-232 (Recommended Standard # 232): این سیستم استاندارد توسط انجمن مهندسین ابزار دقیق در مورد ارتباط سریال عرضه شده است. اگرچه عبارت ارتباط سریال یا Serial Communication به مفهوم ارسال یا دریافت یک بیت در هر لحظه است، اما گاهی این عبارت را مترادف با RS-232 در نظر می‌گیریم. استانداردهای دیگری که در مورد ارتباط سریال وجود دارند و ممکن است با آنها برخورد کرده باشید عبارتند از: RS-485، RS-423 و RS-422.

SCXI (Signal Conditioning eXtensions for Instrumentation): این سیستم یک وسیله‌ی تطبیق‌دهنده با قابلیت و کارایی بالاست که به صورت یک جعبه‌ی جداگانه موجود است و شامل مدول‌های I/O جهت تطبیق دادن و یا مالتی‌پلکس کردن[3] سیگنال می‌باشد. این سیستم توسط شرکت NI ابداع شده است. جعبه‌ی خارجی مذکور به کارت DAQ در رایانه متصل می‌گردد.

۱۰

1- Hewlet Packard Interface Bus
2- Institute of Electrical and Electronic Engineers Standard 488.2
3- Multiplexing

VISA (Virtual Instrument Standard Architecture): این عبارت به نرم‌افزارهای راه‌انداز که توسط شرکت NI ابداع شده است اطلاق می‌گردد. هدف از به‌کارگیری این نرم‌افزار راه‌انداز، هماهنگ نمودن استانداردهای نرم‌افزاری در مورد دستگاههایی است که توسط یکی از سیستم‌های DAQ، GPIB، VXI یا RS-232 با رایانه در ارتباط هستند.

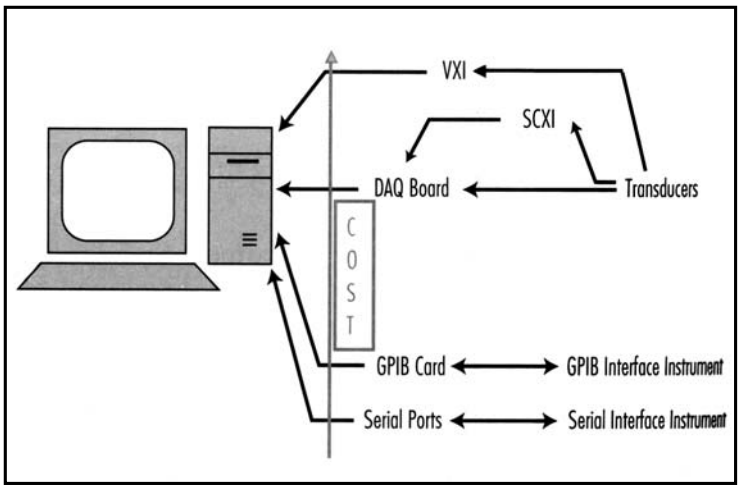
VXI: این واژه خود شامل حروف اختصاری دیگری است. VME eXtensions for Instrumentation که VME خود مخفف عبارات Versa-Modular Eurocard می‌باشد. VXI یک سیستم با قابلیت و کارایی بسیار بالاست که برای اندازه‌گیری و جمع‌آوری داده‌ها به‌کار برده می‌شود. جعبه‌ی VXI معمولاً حاوی یک کارت اصلی رایانه است. بنابراین استفاده از رایانه در این سیستم الزامی نیست. VXI توسط شرکتهایی بجز NI نیز پشتیبانی می‌شود.

در لوح فشرده‌ی همراه این کتاب (در فایل Vxibook.pdf << Manuals) مطالب بسیار جامع و ارزشمندی در مورد این سیستم گنجانده شده است. در صورتی که در فرآیند DAQ از این سیستم استفاده می‌کنید، حتماً مطالب مذکور را مطالعه کنید.

## روش برقراری ارتباط بین رایانه و دنیای خارج

در صورتی که یک رایانه‌ی شخصی با سرعت و قابلیت بالا در اختیار داشته باشید، شور و اشتیاق شما برای کار در محیط LabVIEW بیشتر خواهد شد. در این حالت می‌توانید با استفاده از نرم‌افزار LabVIEW نمونه‌هایی را در خارج از رایانه‌ی خود اندازه‌گیری نمایید. به عنوان مثال ممکن است قصد داشته باشید تا یک نوار مغزی تهیه کنید یا برای آزمایش در مورد کشش الاستیکی لاستیک‌های جدید منحنیهای نیرو را نسبت به جابه‌جایی ترسیم نمایید و یا ممکن است تعداد سیگنال‌های کنترلی در کنترل فرآیند تولید نیمه رساناها به قدری زیاد باشد که بخواهید با استفاده از کارت‌های مخصوص، تعداد سیگنال‌ها را افزایش دهید.

صرف نظر از نوع فرآیند و نحوه‌ی عملکرد برنامه، ابتدا باید روشی جهت جمع‌آوری و ارسال داده‌ها به رایانه پیشنهاد کنید. مسلماً برای گردآوری داده‌ها روشهای بسیاری وجود دارد. اما بهترین و پرکاربردترین روش، به نوع داده‌های جمع‌آوری شده و امکانات موجود و قابل دسترس شما بستگی دارد. قبل از اقدام به خرید سخت‌افزار لازم سعی کنید ابتدا سیگنال‌های مورد نظر را بررسی نموده، آنها را تجزیه و تحلیل کنید. زیرا در برخی موارد نظیر جمع‌آوری داده‌ها از طریق درگاههای سریال ممکن است حتی تهیه‌ی سخت‌افزار اضافی الزامی نباشد. در شکل ۱-۱۰ روشهای مختلف جمع‌آوری داده‌ها و ارسال آنها به رایانه نشان داده شده است. به میزان هزینه‌ی هر سیستم در این شکل توجه کنید.




**شکل ۱-۱۰: روشهای مختلف جمع‌آوری داده‌ها و ارسال آنها به رایانه**

یکی از اولین مواردی که باید در هنگام انتخاب و طراحی سیستم جمع‌آوری داده‌ها یا Data Acquisition در نظر داشته باشید آن است که آیا از یک وسیله‌ی اندازه‌گیری حقیقی متداول نظیر یک مولتی‌متر استفاده می‌کنید و یا اینکه قصد دارید در محیط LabVIEW برنامه‌ای ایجاد کرده، سپس به کمک این نرم‌افزار یک ابزار اندازه‌گیری را به صورت مجازی شبیه‌سازی نمایید و با به‌کارگیری کارت DAQ یا ... داده‌ها را جمع‌آوری کنید.

مواردی نظیر هزینه، زمان و قابلیت انعطاف سیستم اندازه‌گیری مورد استفاده‌ی شما نیز می‌توانند نقش اساسی در تصمیم‌گیری و انتخاب سیستم ایفا کنند. به عنوان مثال اگر قصد داشته باشید که مجموعه‌ای از داده‌هایی را که سطح ولتاژ پایین دارند قرائت کنید، می‌توانید از یک کارت ساده‌ی DAQ استفاده نمایید.

به طور کلی می‌توان گفت که هزینه‌ی تهیه یک کارت DAQ بسیار پایین‌تر از یک ابزار اندازه‌گیری حقیقی است. از طرفی در مقایسه‌ی به‌کارگیری مولتی‌متر و برنامه‌ی ایجاد شده در محیط LabVIEW، در صورت دسترسی به سیستم ارتباطی GPIB، اندازه‌گیری و قرائت داده‌ها و سپس ارسال آنها توسط باس GPIB بسیار ارزان‌تر و مقرون به صرفه‌تر از به‌کارگیری مولتی‌متر است.

در برخی موارد نیز استفاده از رایانه اصلاً امکان‌پذیر نیست. به عنوان مثال در مورد بررسی طیف جرم یک جسم هیچ کارتی که قابلیت انجام این عمل را داشته باشد وجود ندارد.

عامل مهم و اساسی دیگری که ممکن است در تصمیم‌گیری شما در اندازه‌گیریها تأثیرگذار باشد، دسترسی آسان به یک راه‌انداز ابزار اندازه‌گیری یا Instrument Driver است. بنا به تعریف کلی یک «راه‌انداز ابزار اندازه‌گیری» معمولاً شامل تعدادی زیربرنامه است که هر یک فرمانی مشخص و یا دسته‌ای از فرمانها را به ابزار اندازه‌گیری ارسال می‌کند. این زیربرنامه‌ها همگی شامل کدهای سطح پایین هستند و کاربر به راحتی و با سرعت می‌تواند با به‌کارگیری آنها یک برنامه‌ی کامل ایجاد نماید. شرکت NI راه‌اندازهای بسیاری که همگی جهت برقراری ارتباط از GPIB، درگاههای سریال یا سیستم VXI استفاده می‌کنند عرضه کرده است . در برخی موارد نیز تولیدکنندگان ابزارهای اندازه‌گیری، راه‌اندازهای خاصی را برای ابزار خود ارائه می‌دهند.

در صورتی که قصد خرید یک کارت DAQ را دارید یا تاکنون آن را تهیه نموده‌اید با دقت از آن مراقبت و استفاده کنید. بسیاری از کاربران در هنگام استفاده از کارت DAQ از میزان توانایی و قابلیتهای آن اطلاعات کافی در اختیار ندارند. فرض کنید که قصد دارید تا یک سیگنال AC را مورد بررسی قرار دهید. در این حالت نیازی نیست تا به فکر تهیه‌ی یک اسیلوسکوپ باشید. کافی است توسط یکی از برنامه های DAQ که به همراه نرم افزار LabVIEW در اختیار شما قرار می گیرد سیگنال مذکور را بر روی صفحه‌ی نمایشگر رایانه‌ی خود ملاحظه کنید. به کمک یک کارت DAQ می توانید چند سیگنال را در کانال های مختلف اندازه گیری کنید و در صورت نیاز به تعویض و ارتقای کارت یا انتقال آن به سیستم دیگر، لزومی ندارد که در صفحه‌ی نمودار بلوکی تغییری ایجاد کنید.

در لوح فشرده‌ی همراه این کتاب، در فایل Daqbasic.pdf << Manuals مطالب بسیار جامع و ارزشمندی در مورد این سیستم گنجانده شده است. در صورتی که در فرآیند جمع‌آوری داده‌ها از سیستم DAQ استفاده می کنید، حتماً مطالب مذکور را مطالعه کنید.

در ادامه‌ی این فصل به بررسی دو بخش می‌پردازیم. این دو بخش عبارتند از: DAQ و کنترل ابزار اندازه گیری[4]. در بخش DAQ به بررسی سینگال های مختلف، نوع سخت افزار لازم برای جمع آوری داده ها و نحوه‌ی به کارگیری و پیکربندی آن می‌پردازیم و در بخش کنترل ابزار اندازه گیری در مورد پروتکل های سریال و باس GPIB به بحث خواهیم نشست.

## نقش زمان در فرآیند DAQ

اگرچه ممکن است در نگاه اول زمان انجام نمونه برداری چندان مهم به نظر نرسد، اما معمولاً بحرانی ترین و مهمترین پارامتر در اغلب اندازه گیریها زمان نمونه برداری است. هر موردی را که بخواهید اندازه گیری یا مشاهده کنید به نحوی با زمان در ارتباط است. به عنوان مثال چگونگی تغییرات دمای یک موتور نسبت به زمان یا بررسی سیگنال رادیویی فیلتر شده نسبت به زمان همگی به نحوی با زمان در ارتباط هستند. منظور ما از بررسی یک فرآیند در حقیقت بررسی زمان وقوع آن است، نه اینکه واقعاً چه اتفاقی روی داده است.

همانند فرآیندهای دیگر، زمان نیز یک عامل تعیین کننده و اساسی در عملیات DAQ است. زمان بندی در تصمیم گیری و اتخاذ سیستم DAQ به دو دلیل حائز اهمیت است:

اولاً درمورد نرم افزار و... نیاز دارید تا نرخ نمونه برداری را تعیین کنید. ثانیاً باید بدانید که در مورد وظایف و اعمال دیگری نظیر ورودی/خروجی فایل چه مدت زمانی را به پردازنده اختصاص دهید.

اگر تنها لازم باشد تا داده ها را در یک ثانیه چندین مرتبه قرائت کنید و نیاز به زمان بندی دقیق بین



4- Instrument I/O

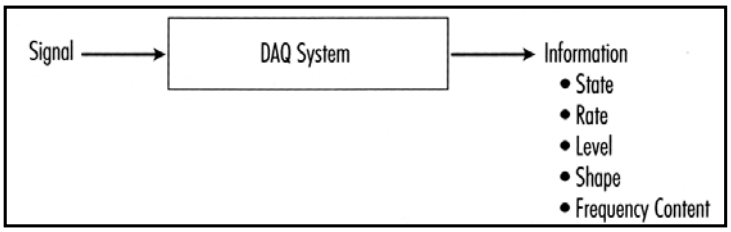
فواصل نمونه‌برداری احساس نشود، می‌توانید مستقیماً نرخ نمونه‌برداری را با استفاده از توابع و دستورهای زمان‌بندی در نرم‌افزار LabVIEW کنترل کنید. به عنوان مثال می‌توانید در یک برنامه که به جمع آوری داده‌ها اختصاص داده شده است، دستور Wait را که از دستورهای زمان‌بندی در بسته‌ی نرم‌افزاری LabVIEW می‌باشد در داخل یک حلقه قرار دهید و بدین ترتیب زمان اجرای تکرارهای حلقه را تعیین کنید. در مورد سیگنال‌های AC یا برای مواردی که دقت بیشتری را می‌طلبند، می‌توانید با پیکربندی کارت DAQ نرخ نمونه‌برداری را تعیین کنید. در فصل ۱۱ این مطالب را همراه با مثالهای بیشتری فرا خواهید گرفت.

## طبقه‌بندی سینگال‌ها

قبل از ورود به مبحث DAQ قصد داریم ابتدا ماهیت سیگنال‌های مختلف را مورد بررسی قرار دهیم. یک سیگنال به طور ساده یک کمیت فیزیکی است که می‌توان اطلاعاتی را از اندازه و یا تغییرات آن نسبت به زمان یا برخی از متغیرهای دیگر به‌دست آورد.

فرض کنید که می‌خواهید یک کمیت را اندازه‌گیری کنید. جهت تطابق سازی سیگنال و یا برای اندازه‌گیری کمیت فیزیکی مورد اندازه‌گیری توسط یک کارت DAQ، ابتدا بایدکمیت مورد نظر را به یک سیگنال الکتریکی نظیر ولتاژ یا جریان تبدیل کنید. برای تبدیل کمیت فیزیکی به یک سیگنال الکتریکی از مبدل استفاده می شود.

به عنوان مثال برای اندازه‌گیری دما باید به هر حال دما را به صورت ولتاژ درآورید (به صورتی که برای کارت DAQ قابل قرائت باشد). برای تبدیل فیزیکی دما به یک سیگنال الکتریکی چند مبدل گرمایی وجود دارد. این مبدلها برای تبدیل مذکور برخی از خواص فیزیکی گرمایی را مورد استفاده قرار می‌دهند. پس از تبدیل کمیت فیزیکی به سیگنال الکتریکی می‌توانید سیگنال‌ها را اندازه‌گیری کنید و اطلاعات مفیدی نظیر وضعیت[۵]، نرخ[۶]، سطح[۷]، شکل[۸] و مؤلفه‌های فرکانسی[۹] را از آنها به‌دست آورید. در شکل ۲-۱۰ این مطلب نشان داده شده است.

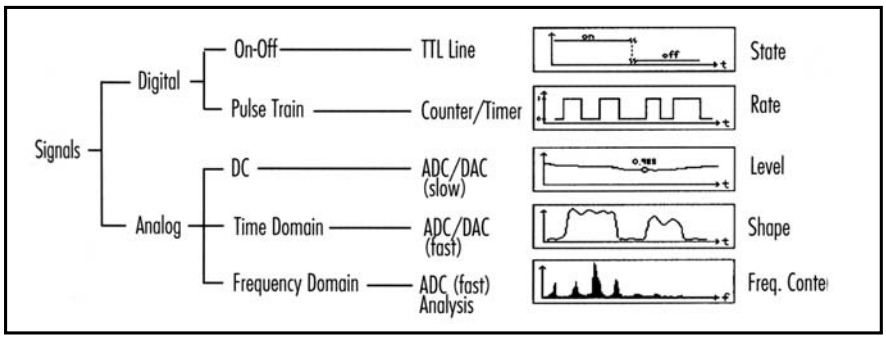



**شکل ۲-۱۰: اطلاعات مختلفی از قبیل وضعیت، نرخ، دامنه، شکل و مؤلفه‌های فرکانسی که از سیگنال اندازه‌گیری شده به‌دست می‌آیند.**

۱۰

5- State
6- Rate
7- Level
8- Shape
9- Frequency Content

در مبحث اندازه گیری سیگنال باید سیگنال مورد نظر را در یکی از پنج دسته‌ی تعریف شده طبقه بندی کنید. دسته بندی باید به گونه ای انجام شود که تمامی اطلاعات مورد نیاز را در بر داشته باشد. برای طبقه بندی سیگنال ابتدا آن را در یکی از دو گروه آنالوگ یا دیجیتال قرار دهید. همان گونه که می دانید یک سیگنال دیجیتال یا باینری تنها دو سطح مجاز دارد. یکی سطح بالا یا روشن و دیگری سطح پایین یا خاموش. سیگنال آنالوگ حاوی اطلاعاتی است که نسبت به زمان تغییر می کند. در شکل ۳-۱۰ نحوه‌ی طبقه بندی سیگنال ها را ملاحظه می کنید.




**شکل ۳-۱۰: طبقه بندی سیگنال ها**

سیگنال های دیجیتال به دو گروه عمده و سیگنال های آنالوگ به سه گروه اصلی تقسیم می شوند. دو گروه سیگنال دیجیتال عبارتند از:

- سیگنال دیجیتال on-off
- قطار پالس

سه گروه سیگنال آنالوگ عبارتند از:

- سیگنال DC آنالوگ
- سیگنال آنالوگ در حوزه‌ی زمان
- سیگنال آنالوگ در حوزه‌ی فرکانس

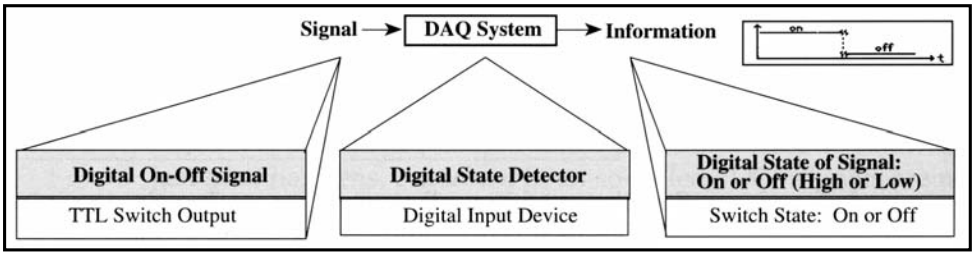
هر یک از این گروه‌ها حاوی اطلاعات منحصر به فردی هستند. همان گونه که در شکل ۳-۱۰ ملاحظه می کنید پنج گروه نشان داده شده متناظر با پنج دسته‌ی وضعیت، نرخ، دامنه، شکل و مؤلفه های فرکانسی می باشند.

## سیگنال های دیجیتال

### سیگنال دیجیتال on-off

سیگنال های دیجیتال یا باینری به صورت رشته ای از بیت ها هستند. اولین دسته از سیگنال های دیجیتال، سیگنال on-off است. این سیگنال، اطلاعاتی در مورد وضعیت دیجیتالی یک سیگنال در بر دارد. بنابراین برای اندازه گیری این نوع سیگنال می توان از یک «تشخیص دهنده‌ی پالس» استفاده نمود. خروجی یک سوییچ TTL مثال ساده‌ای از یک سیگنال دیجیتال on-off است. در شکل ۴-۱۰ این نوع سیگنال و اطلاعاتی را که می توان از آن به دست آورد به همراه

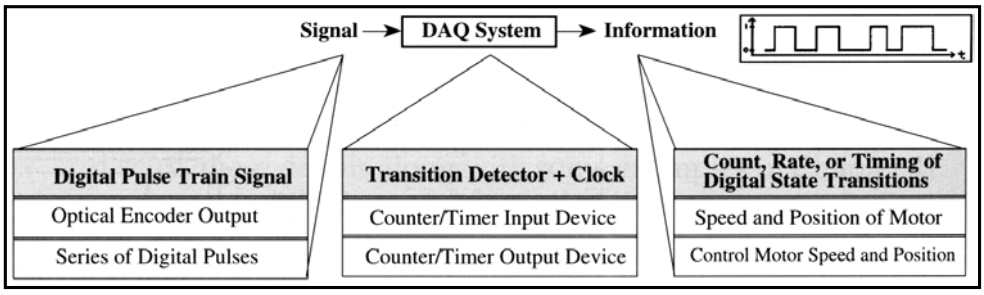
مشخصات سیستم DAQ استفاده شده جهت اندازه گیری این گروه از سیگنال ها ملاحظه می کنید.




**شکل ۴-۱۰: سیگنال دیجیتال on-off به همراه مشخصات سیستم DAQ استفاده شده جهت اندازه گیری این نوع سیگنال**

## سیگنال دیجیتال قطار پالس

سیگنال دیجیتال قطار پالس دومین گروه از سیگنال های دیجیتال است. همان گونه که از نام این گروه استنباط می شود این نوع سیگنال شامل تعدادی پالس می باشد. اطلاعات موجود در این نوع سیگنال از تعداد و نرخ تغییر حالات به دست می آیند. سیگنال خروجی یک انکودر نوری[10] که بر روی محور یک موتور نصب شده است نمونه‌ی ساده‌ای از این نوع سیگنال می باشد. برخی از دستگاهها برای آغاز به کار خود به ورودی پالس نیاز دارند. به عنوان مثال، جهت کنترل موقعیت و سرعت یک موتور پله‌ای[11]، ورودی آن حتماً باید به صورت تعدادی پالس دیجیتال باشد. در شکل ۵-۱۰ این نوع سیگنال و اطلاعاتی که می توان از آن به دست آورد به همراه مشخصات سیستم DAQ استفاده شده جهت اندازه گیری این گروه از سیگنال ها نشان داده شده است.




**شکل ۵-۱۰: سیگنال دیجیتال قطار پالس به همراه مشخصات سیستم DAQ استفاده شده جهت اندازه گیری این نوع سیگنال**

## سیگنال های آنالوگ

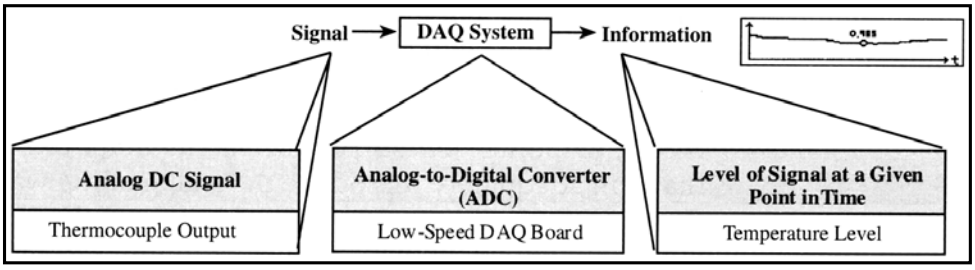
### سیگنال DC آنالوگ

سیگنال های DC آنالوگ سیگنال هایی هستند که تغییرات آنها نسبت به زمان بسیار آهسته صورت می گیرد. مهمترین پارامتری که در مورد این سیگنال ها وجود دارد، میزان سطح یا دامنه‌ی سیگنال در یک فاصله‌ی زمانی معین است. به دلیل اینکه تغییرات سیگنال DC آنالوگ نسبت به زمان به آهستگی صورت می گیرد، دقت در دامنه‌ی سیگنال اندازه گیری شده مهم تر از فاصله‌ی زمانی است که در آن، عمل نمونه برداری انجام می شود.

۱۰

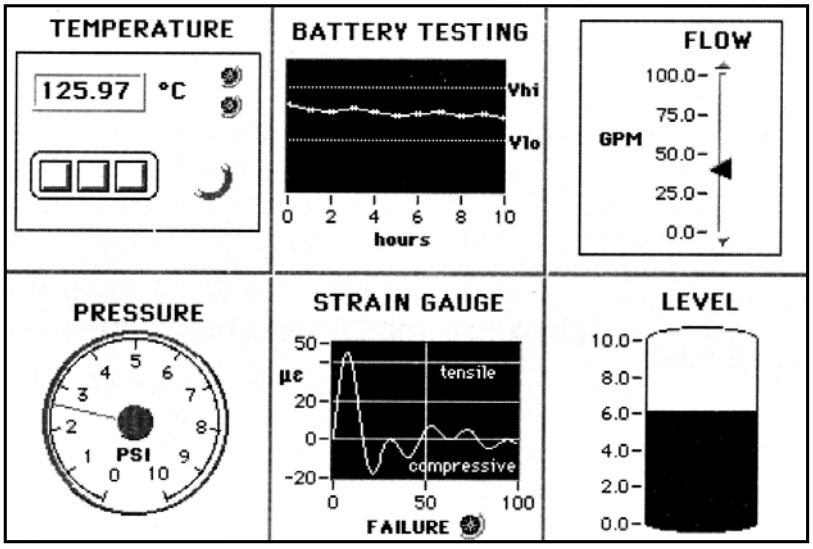
10- Optical Encoder
11- Stepper Motor

ابزار یا کارت DAQ که برای جمع آوری این نوع سیگنال مورد استفاده قرار می گیرد شبیه به مبدل ADC عمل می‌نماید و سیگنال الکتریکی آنالوگ را به مقادیر دیجیتالی قابل درک برای رایانه تبدیل می کند. در شکل ۶-۱۰ این نوع سیگنال و اطلاعاتی را که می توان از آن به دست آورد به همراه مشخصات سیستم DAQ استفاده شده جهت اندازه گیری این گروه از سیگنال ها ملاحظه می کنید.




**شکل ۶-۱۰: سیگنال DC آنالوگ به همراه مشخصات سیستم DAQ استفاده شده جهت اندازه گیری این نوع سیگنال**

در شکل ۷-۱۰ نمونه هایی از سیگنال های DC آنالوگ نظیر دما، سطح ولتاژ باطری، میزان فلو، فشار، نشان دهنده‌ی کشش و سطح مایعات داخل مخزن را ملاحظه می کنید. در هر یک از موارد مذکور، کارت DAQ در هر لحظه یک مقدار را که بیانگر سطح سیگنال در آن لحظه است به دست می دهد. بنابراین برای به نمایش در آوردن این سیگنال ها در نرم افزار LabVIEW اغلب از نشان دهندهایی نظیر خط کشهای عددی، گیج ها و... استفاده می گردد.




**شکل ۷-۱۰: چند نمونه از سیگنال های DC آنالوگ**

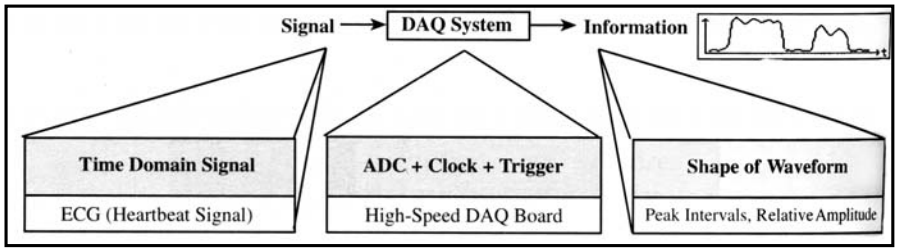
سیستم یا کارت DAQ که جهت جمع آوری سیگنال های DC آنالوگ مورد استفاده قرار می گیرد باید دارای مشخصات زیر باشد:

- دقت بالا داشته باشد. به گونه‌ای که بتواند سطح سیگنال را با دقت کافی اندازه گیری نماید.
- پهنای باند آن کم باشد. به طوری که بتواند سیگنال را با نرخ پایین نمونه برداری کند.

## سیگنال آنالوگ در حوزه‌ی زمان

اطلاعات به‌دست آمده و حائز اهمیت در مورد سیگنال‌های آنالوگ در حوزه‌ی زمان نه تنها در سطح یا دامنه‌ی سیگنال وجود دارد، بلکه این اطلاعات از نحوه‌ی تغییر سطح سیگنال با زمان نیز به‌دست می‌آیند. به همین دلیل این سیگنال‌ها تا حدودی با سیگنال‌های دیگر تفاوت دارند. در هنگام اندازه‌گیری یک سیگنال در حوزه‌ی زمان، برخی از خواص و مشخصات سیگنال نظیر شیب، موقعیت و فرم نقاط ماکزیمم و مینیمم و... جالب و حائز اهمیت است.

برای اندازه‌گیری یک سیگنال در حوزه‌ی زمان باید دنباله‌ی زمانی دقیقی از دامنه‌ی نقاط اندازه‌گیری شده در اختیار داشته باشید. این اندازه‌گیریها باید به گونه‌ای انجام گیرند که فرم شکل موج را به طور کامل به‌دست دهند. همچنین فرآیند اندازه‌گیری باید در زمانی خاص و معین آغاز گردد. به طوری که حتماً بخشهای مفید و حاوی اطلاعات سیگنال جمع‌آوری شوند. بنابراین ابزار یا کارت DAQ که جهت اندازه‌گیری سیگنال‌های در حوزه‌ی زمان به‌کار می‌رود شامل یک مبدل ADC، یک پالس نمونه‌برداری و یک تریگرکننده است. پالس نمونه‌برداری، زمان انجام تبدیلات سیگنال آنالوگ به دیجیتال را تنظیم می‌کند. برای اطمینان از جمع‌آوری بخش مفید و حاوی اطلاعات در سیگنال مورد نظر، فرآیند اندازه‌گیری بر اساس برخی از شرایط خارجی در زمان معین و مناسب تریگر می‌شود. در شکل ۸-۱۰این نوع سیگنال و اطلاعاتی را که می‌توان از آن به‌دست آورد به همراه مشخصات سیستم DAQ استفاده شده جهت اندازه‌گیری این گروه از سیگنال‌ها ملاحظه می‌کنید.

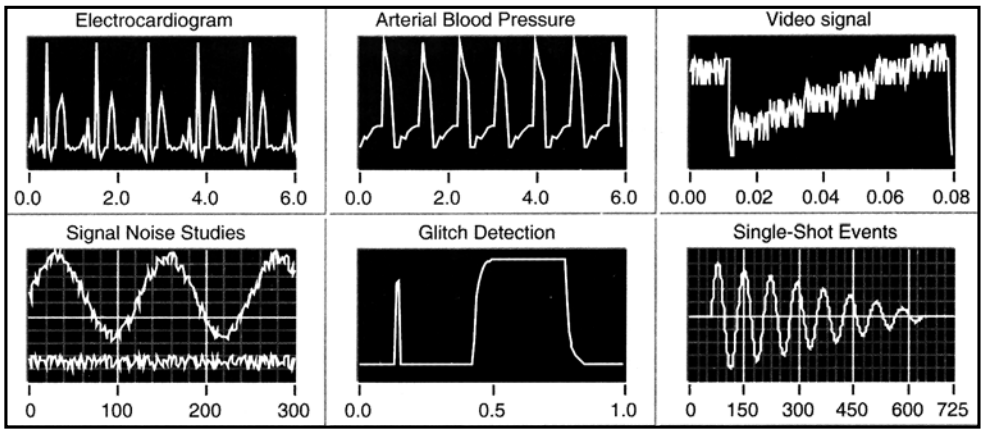



**شکل ۸-۱۰: سیگنال آنالوگ در حوزه‌ی زمان به همراه مشخصات سیستم DAQ استفاده شده جهت اندازه‌گیری این نوع سیگنال**

سیستم DAQ استفاده شده جهت جمع‌آوری سیگنال‌های در حوزه‌ی زمان باید دارای خصوصیات زیر باشد:

- پهنای باند وسیع داشته باشد. به گونه‌ای که بتواند عمل نمونه‌برداری را با نرخ بالا انجام دهد.
- دقت پالس نمونه‌برداری بالا باشد. برای اینکه بتواند نمونه‌برداری را در فواصل زمانی دقیق انجام دهد.
- تریگرکننده باشد. به‌دلیل اینکه اندازه‌گیریها را در زمان دقیق آغاز نماید و در این فرآیند، اطلاعات ارزشمند سیگنال از دست نرود.

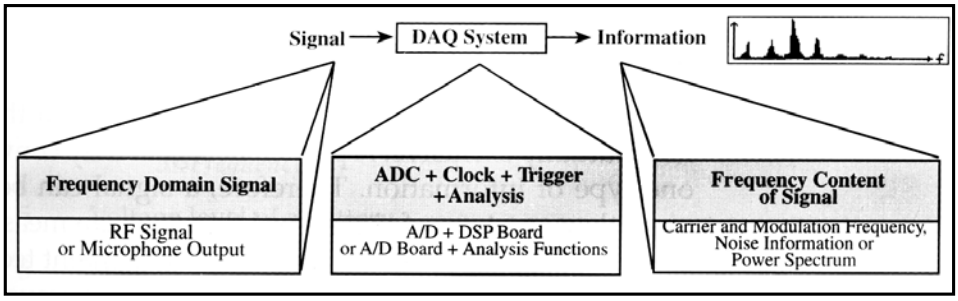
تعدادی بیشماری از سیگنال‌های آنالوگ در حوزه‌ی زمان وجود دارند که تعدادی از آنها را در شکل ۹-۱۰ مشاهده می‌کنید. وجه مشترک موجود بین این سیگنال‌ها آن است که سطح و دامنه‌ی موج سیگنال‌ها نسبت به زمان مهمترین ویژگی قابل توجه آنهاست.




**شکل ۹-۱۰: چند نمونه از سیگنال های آنالوگ در حوزه‌ی زمان**

## سیگنال آنالوگ در حوزه‌ی فرکانس

سیگنال های آنالوگ در حوزه‌ی فرکانس نیز شبیه به سیگنال های آنالوگ در حوزه‌ی زمان هستند؛ زیرا این سیگنال ها نیز اطلاعاتی در موردچگونگی تغییر سیگنال نسبت به زمان در اختیار ما قرار می دهند. با این تفاوت که اطلاعات به دست آمده از سیگنال های آنالوگ در حوزه‌ی فرکانس بر اساس مؤلفه های فرکانسی است که برخلاف شکل، فرم و خواص متغیر با زمان شکل موج می باشد. نظیر سیگنال های آنالوگ در حوزه‌ی زمان، ابزاری که جهت اندازه گیری یک سیگنال در حوزه‌ی فرکانس به کار برده می شود برای در برگرفتن تمامی شکل موج باید شامل یک مبدل ADC، یک پالس نمونه برداری و یک تریگرکننده باشد. علاوه بر این، ابزار مذکور باید قدرت آنالیز لازم جهت استخراج اطلاعات فرکانسی از سیگنال را داشته باشد. برای پردازش سیگنال آنالوگ می توانید نرم افزارهای کاربردی و یا سخت افزارهای طراحی شده‌ی خاص را جهت تجزیه و تحلیل سیگنال با سرعت و راندمان بالا به کار گیرید. در شکل ۱۰-۱۰ این نوع سیگنال و اطلاعاتی را که می توان از آن به دست آورد به همراه مشخصات سیستم DAQ استفاده شده جهت اندازه گیری این گروه از سیگنال ها ملاحظه می کنید.

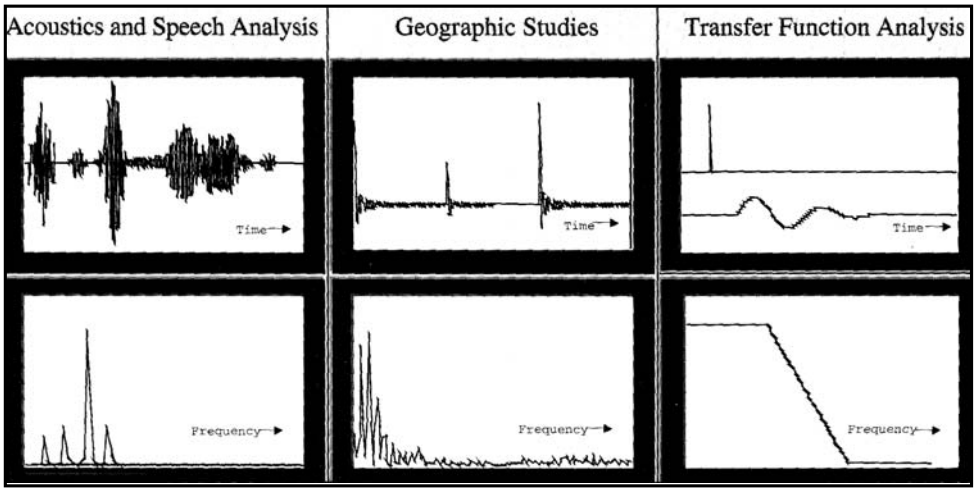



**شکل ۱۰-۱۰: سیگنال آنالوگ در حوزه‌ی فرکانس به همراه مشخصات سیستم DAQ استفاده شده جهت اندازه گیری این نوع سیگنال**

سیستم DAQاستفاده شده جهت جمع‌آوری سیگنال‌های آنالوگ در حوزه‌ی فرکانس باید دارای خصوصیات زیر باشد:

- پهنای باند وسیع داشته باشد. به گونه‌ای که بتواند عمل نمونه‌برداری را با نرخ بالا انجام دهد.
- دقت پالس نمونه‌برداری بالا باشد. برای اینکه بتواند نمونه‌برداری را در فواصل زمانی دقیق انجام دهد.
- تریگرکننده باشد. به‌دلیل اینکه اندازه‌گیریها را در زمان دقیق آغاز نماید و در این فرآیند اطلاعات ارزشمند سیگنال از دست نرود.
- حاوی توابع و دستورهای تحلیلگر باشد تا بتواند اطلاعات زمانی را به اطلاعات فرکانسی تبدیل کند.

در شکل ۱۱-۱۰ چند نمونه از سیگنال‌های آنالوگ در حوزه‌ی فرکانس نشان داده است. همان گونه که ملاحظه می‌کنید هر یک از نمونه‌ها شامل یک گراف از سیگنال اندازه‌گیری شده‌ی اصلی است که نسبت به زمان متغیر می‌باشد. در زیر هر گرافِ زمانی، طیف فرکانسی سیگنال نیز ترسیم شده است. معمولاً جهت بررسی بسیاری از سیگنال‌ها و کاربردهای خاص از این روش تجزیه و تحلیل استفاده می‌شود. برخی از سیگنال‌ها و کاربردهای مذکور عبارتند از: صحبت و صوت، سیگنال‌های ژئوفیزیکی، لرزش و توابع انتقال سیستم.



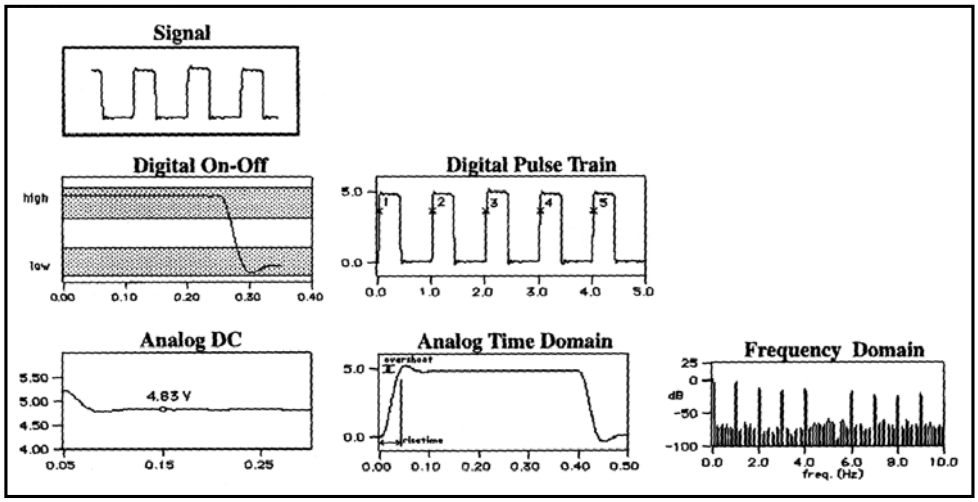

**شکل ۱۱-۱۰: چند نمونه از سیگنال‌های آنالوگ در حوزه‌ی فرکانس**

## یک سیگنال و پنج روش اندازه‌گیری

طبقه‌بندی پنج گانه‌ی سیگنال که در این بخش بیان شد انحصاری نیست. بدین معنی که هر سیگنال حتماً نباید تنها در یکی از این گروهها قرار گیرد؛ زیرا ممکن است یک سیگنال حاوی اطلاعات مختلفی باشد. بنابراین ممکن است یک سیگنال در دو یا چند گروه از سیگنال‌های مذکور قرار گیرد. واضح است که برای اندازه‌گیری آن نیز بیش از یک روش وجود دارد. بنابراین می‌توانید با استفاده از روشهای اندازه‌گیری ساده‌تر نظیر سیگنال دیجیتال on-off، قطار پالس و سیگنال‌های DC آنالوگ، فرآیند DAQ را انجام دهید. زیرا روشهای مذکور در مورد این سیگنال‌ها ساده‌تر از روشهای عنوان شده در مورد سیگنال‌های آنالوگ در حوزه‌ی زمان

یا فرکانس هستند. بنابراین با توجه به موارد مذکور می توانید فرآیند DAQ را به کمک سیستم های مختلفی از کارت های ساده‌ی ورودی دیجیتالی تا سیستم های پیچیده‌ی آنالیزکننده‌ی فرکانسی انجام دهید.

نوع اطلاعاتی که قصد استخراج آنها را از سیگنال مورد بحث دارید در انتخاب روش اندازه گیری سیگنال مؤثر و تعیین کننده است. در شکل ۱۲-۱۰ ملاحظه می کنید که یک سیگنال که شامل تعدادی پالس ولتاژ است به چه صورت اطلاعاتی برای ۵ طبقه سیگنال عنوان شده در بخش قبل را در بر دارد.




**شکل ۱۲-۱۰: یک سیگنال و پنج روش اندازه گیری**

## تمرین ۱-۱۰: طبقه بندی سیگنال ها

سیگنال های زیر را در یکی از ۵ گروه عنوان شده طبقه بندی کنید. در برخی موارد نیز ممکن است یک سیگنال در بیش از یک گروه قرار گیرد. در این حالت دسته ای را برای سیگنال مذکور در نظر بگیرید که به نظر شما تطبیق بیشتری با سیگنال مورد بحث دارد.

۱- سیگنال دیجیتال on-off

۲- سیگنال دیجیتال قطار پالس

۳- سیگنال DC آنالوگ

۴- سیگنال آنالوگ در حوزه‌ی زمان

۵- سیگنال آنالوگ در حوزه‌ی فرکانس

- سطح ولتاژ یک باطری ۱ ۲ ۳ ۴ ۵
- وضعیت یک رله ۱ ۲ ۳ ۴ ۵
- داده های درگاه موازی در رایانه در حین چاپ کردن ۱ ۲ ۳ ۴ ۵
- نویز جرقه در یک منبع قدرت ۱ ۲ ۳ ۴ ۵
- تابع انتقال یک فیلتر دیجیتال ۱ ۲ ۳ ۴ ۵

- نوار مغزی ۱ ۲ ۳ ۴ ۵
- صحبت در یک میکروفون ۱ ۲ ۳ ۴ ۵
- فشار مطلق در سیلندر یک موتور ۱ ۲ ۳ ۴ ۵
- فشار اتمسفر ۱ ۲ ۳ ۴ ۵

پاسخهای خود را با جواب صحیح این تمرین که در انتهای فصل وجود دارد مقایسه کنید.

## مبدلها

در صورت استفاده از یک سیستم DAQ به خاطر داشته باشید که کمیت اندازه گیری شده در نهایت باید به سیگنال ولتاژ یا جریان تبدیل شود. برای تبدیل کمیت فیزیکی مورد بحث نظیر دما، نیرو، صوت، شدت نور و... به یک سیگنال الکتریکی، از مبدل استفاده می شود. در جدول ۱-۱۰ تعدادی از مبدلهای عمومی نشان داده شده است. این مبدلها برای تبدیل کمیت فیزیکی به کمیتهای قابل اندازه گیری یا سیگنال های قابل درک برای رایانه به کار می روند.

**جدول ۱-۱۰: تعدادی از مبدلهای عمومی**

| **Phenomena** | **Transducer** |
|---|---|
| Temperature | Thermocouple<br>Resistance Temperature Detector (RTD)<br>Thermistor<br>Integrated Circuit Sensor |
| Light | Photomultiplier Tube<br>Photoconductive Cell |
| Sound | Microphone |
| Force and Pressure | Strain Gauge<br>Piezoelectric Transducer<br>Load Cell |
| Position (Displacement) | Potentiometer<br>Linear Voltage Differential Transformer (LVDT)<br>Optical Encoder |
| Fluid Flow | Differential Pressure Flowmeter<br>Rotational Flowmeter<br>Ultrasonic Flowmeter |
| PH | PH Electrode |

## تطبیق دادن سیگنال

«بسیار خوب! اکنون که نحوه‌ی جمع آوری داده ها را فراگرفته اید کافی است خروجی مبدل را مستقیماً به کارت DAQ متصل کنید.»

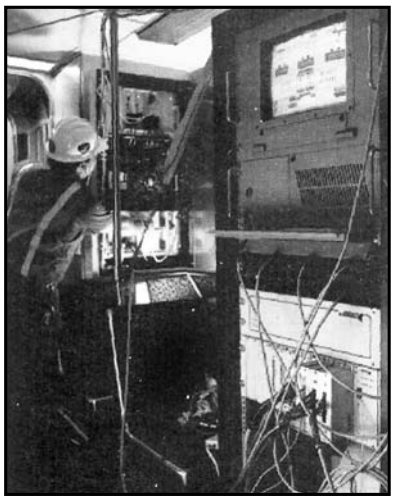
آیا واقعاً همین طور است؟

شاید در ۵۰٪ موارد یا حتی بیشتر، این مطلب صحیح نیست. همواره این نکته را در نظر داشته باشید که ما در دنیایی زندگی می کنیم که مملو از نویزهای الکتریکی است. بنابراین در زمانی که سیگنال اندازه گیری شده‌ی مورد بحث در راه رسیدن به کارت DAQ می باشد ممکن است تحت تأثیر نویز یا عوامل دیگر به یک نمونه‌ی بی حاصل تبدیل شود. معمولاً در مورد سیگنال های آنالوگ که بیانگر پدیده های فیزیکی هستند، تطبیق دادن سیگنال نیز انجام می گیرد.

به طور ساده می توان گفت که تطبیق دادن سیگنال یک عملیات ساده بر روی سیگنال مورد نظر است به گونه‌ای که برای پردازش در یک کارت DAQ آماده شود. سیگنال مورد بحث باید با وضوح کامل، بدون نویز و با دقت کافی با ولتاژی معمولاً در حدود ±5V یا 0-10V و جریانی در حدود 20mA به کارت DAQ وارد شود. همچنین برای به کارگیری یک مبدل لازم است در مورد آن اطلاعاتی در اختیار داشته باشید. به عنوان مثال جهت تطبیق دادن یک سیگنال صوتی با استفاده از یک میکروفون شاید اتصال سیستم به زمین و یا استفاده از یک فیلتر پایین گذر کافی به نظر رسد. اما اگر قصد داشته باشید سطوح یونیزاسیون را در یک غشای پلاسما در سطح ولتاژ 800V اندازه گیری نمایید و نمی خواهید رایانه‌ی خود را کباب کنید، باید سیستم های تطبیق دهنده‌ی پیچیده تری را به کار برید. در این سیستم ها معمولاً از تقویت کننده های ایزوله کننده با بهره‌ی پایین استفاده می شود.

شرکت NI در مورد سیگنال هایی که نیاز به تطبیق داشته یا برای سیستم هایی که دارای تعداد زیادی سیگنال هستند، اقدام به طراحی و ساخت سیستم های SCXI نموده است. یک سیستم SCXI شامل یک جعبه است که واحدهای مدولار در داخل آن نصب می شوند. این واحدهای مدولار شامل مالتی پلکسرهای ورودی آنالوگ، کارت های خروجی آنالوگ، مدول های تطبیق دهنده و... می باشند. بنابراین برای تطبیق دادن سیگنال حتماً از سیستم SCXI استفاده کنید.

**شکل ۱۳-۱۰: جعبه‌ای که در سمت چپ رایانه قرار گرفته است همان سیستم SCXI می باشد. به مدول‌های نصب شده در داخل جعبه توجه کنید. همان گونه که ملاحظه می کنید نرم‌افزار LabVIEW در حال اجرا بر روی رایانه است. این برنامه جهت آزمایش نیروی محرکه‌ی اتومبیل طراحی شده است. در این آزمایش، داده‌ها از طریق سیستم SCXI جمع آوری می شوند.**

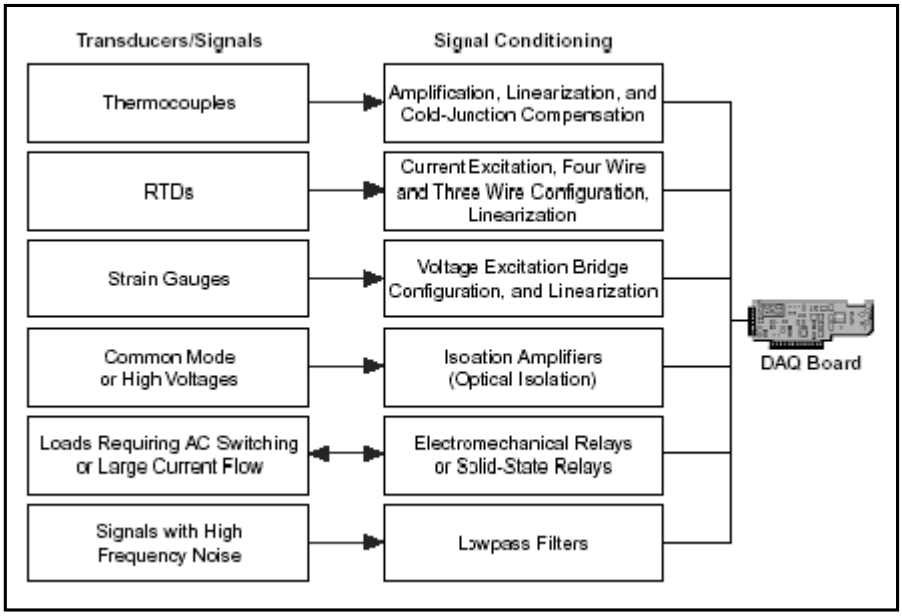


**شکل ۱۴-۱۰: همان گونه که در این شکل ملاحظه می کنید در محیطهای صنعتی نیز می توان از سیستم SCXI استفاده نمود.**

برخی از صورتهای عمومی در تطبیق دادن سیگنال عبارتند از:

- تقویت کردن[۱۲]
- خطی کردن[۱۳]
- ایزوله سازی[۱۴]
- فیلتر کردن[۱۵]

در شکل ۱۵-۱۰ انواع مبدلها به همراه صورتهای مختلف تطبیق سیگنال نشان داده شده است.




**شکل ۱۵-۱۰: چند مبدل به همراه صورتهای مختلف تطبیق سیگنال**

۱۰

12- Amplification
13- Linearization
14- Isolation
15- Filtering

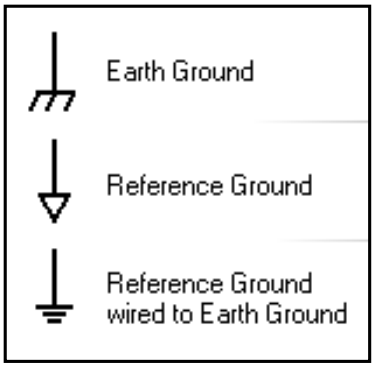
## نقطه‌ی مرجع اندازه‌گیری

برخی از کمیتهای فیزیکی نظیر جرم جسم، مطلق[۱۶] هستند. اما ولتاژ یا پتانسیل الکتریکی یک جسم مطلق نیست. به عبارت دیگر پتانسیل الکتریکی جسم یک کمیت نسبی است. برای تعیین پتانسیل الکتریکی باید ولتاژ یک جسم نسبت به یک نقطه‌ی مرجع سنجیده شود. در حقیقت، مقدار ولتاژ به صورت اختلاف پتانسیل الکتریکی بین دو جسم تعریف می شود. معمولاً یکی از این دو جسم به عنوان مرجع در نظر گرفته شده و برای آن، پتانسیل ۰ V در نظر گرفته می شود. در اکثر موارد، از ذکر مرجع ولتاژ یا نقطه‌ی ۰ خودداری می شود. به دلیل اینکه معمولاً در اندازه گیری اختلاف پتانسیل، از زمین به عنوان مرجع استفاده می شود، این مطلب منجر به بروز برخی اشتباهات می گردد. در سیستم های اندازه گیری از واژه‌ی «زمین» برای بیان دو مفهوم تقریباً متفاوت استفاده می شود. در ادامه به بررسی این دو مفهوم می پردازیم:

نقطه‌ی زمین[۱۷] به پتانسیل زمین زیر پایتان اطلاق می گردد. اکثر خروجیهای الکتریکی دارای سیمی هستند که به زمین متصل شده است و معمولاً برای رعایت نکات ایمنی به سیستم الکتریکی ساختمان اتصال می یابد. در بسیاری از سیستم های اندازه گیری نیز این نقطه برای اتصال یافتن به زمین تعبیه شده است. به احتمال زیاد، اکثر شما عبارت «System Ground» را بر روی پانل آنها دیده‌اید. همان گونه که عنوان شد دلیل اصلی استفاده از نقطه‌ی زمین، ایمنی بیشتر است نه اینکه واقعاً به عنوان مرجعی برای محاسبه‌ی اختلاف پتانسیل مورد استفاده قرار گیرد. بنابراین هنگامی که به یک مرجع پتانسیل نیاز داریم منظور ما زمین سیستم نیست.

زمین مرجع[۱۸] که گاهی با نام زمین مشترک خوانده می شود، همان نقطه‌ی مرجع جهت اندازه گیری اختلاف پتانسیل است. ممکن است زمین مرجع به زمین سیستم نیز متصل باشد (یا نباشد). این نقطه همان نقطه ای است که در اکثر ابزارها، دستگاهها و برای اندازه گیری پتانسیل الکتریکی مورد استفاده قرار می گیرد.

نماد های نشان داده شده در شکل ۱۶-۱۰ جهت نمایش مراجع زمین به کار برده می شوند. در هنگام استفاده از این نمادها کاملاً مراقب باشید، زیرا حتی برخی از کارشناسان نیز در هنگام استفاده و به کارگیری این علائم در نقشه ها دچار اشتباه می شوند.




**شکل ۱۶-۱۰: نمادهای مختلف جهت نمایش مراجع زمین**

۱۰

۱۶- البته همگی ما می دانیم در صورتی که سرعت جسم به سرعت سیر نور نزدیک شود، جرم آن نیز تغییر می کند.

17- Earth Ground
18- Reference Ground

واضح است که کارت هایی که برای انجام عملیات DAQ بر روی کارت اصلی رایانه نصب می شوند نیز جهت اندازه گیری ولتاژ به مرجع مشخصی نیاز دارند. حال ممکن است این سؤال در ذهن شما ایجاد شده باشد که:

«در مورد این کارت ها باید از چه مرجعی استفاده کنیم؟»

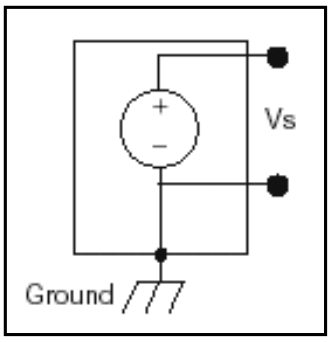
پاسخ به این سؤال بسیار ساده و تا حدی گمراه کننده است، زیرا بسته به نوع سیگنال می توانید یکی از مراجع زمین را مورد استفاده قرار دهید. برای درک این مطلب، سیگنال ها را به دو دسته‌ی عمده تقسیم می کنیم:

- سیگنال های متصل شده به زمین
- سیگنال های شناور

در ادامه به بررسی این دو نوع سیگنال می پردازیم.

## سیگنال متصل شده به زمین

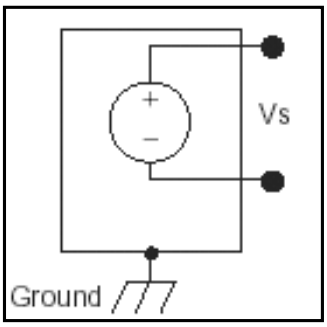
در سیگنال های اتصال یافته به زمین، مقدار ولتاژ نسبت به زمین سیستم اعم از زمین ساختمان یا زمین مرجع سنجیده می شوند. به دلیل استفاده از زمین سیستم در آنها، این سیگنال ها با کارت های DAQ دارای زمین مشترک هستند. بهترین مثال این سیگنال ها، مولدهای سیگنال یا منابع تغذیه‌ای هستند که از طریق پریزهای دیواری به زمین ساختمان متصل می شوند. در شکل ۱۷-۱۰ نحوه‌ی نمایش این نوع سیگنال و طریقه‌ی اتصال آن به زمین سیستم نشان داده شده است.




**شکل ۱۷-۱۰: نحوه‌ی نمایش سیگنال متصل شده به زمین و طریقه‌ی اتصال آن به زمین سیستم**

## سیگنال شناور

در سیگنال شناور مقدار ولتاژ نسبت به هیچ نقطه‌ی مشترکی نظیر زمین ساختمان یا زمین مرجع سنجیده نمی‌شود. برخی از منابع سیگنال شناور عبارتند از: باطری‌ها، ترموکوپل ها، ترانسفورماتورها و تقویت کننده ها. همان گونه که در شکل ۱۸-۱۰ ملاحظه می کنید هیچ یک از ترمینال های این منابع به زمین متصل نشده اند. بنابراین می توان گفت که هر ترمینال مستقل از زمین سیستم می باشد.




**شکل ۱۸-۱۰: نحوه‌ی نمایش منبع سیگنال شناور**

## اندازه‌گیری اختلاف پتانسیل

جهت اندازه‌گیری یک سیگنال می‌توانید همواره کارت DAQ را به گونه‌ای استفاده کنید که در یکی از سه گروه زیر قرار گیرد:

- سیستم اندازه‌گیری تفاضلی
- سیستم اندازه‌گیری RSE[19]
- سیستم اندازه‌گیری NRSE[20]

در ادامه‌ی بحث، این مفاهیم را مورد بررسی قرار می‌دهیم.

## سیستم اندازه‌گیری تفاضلی

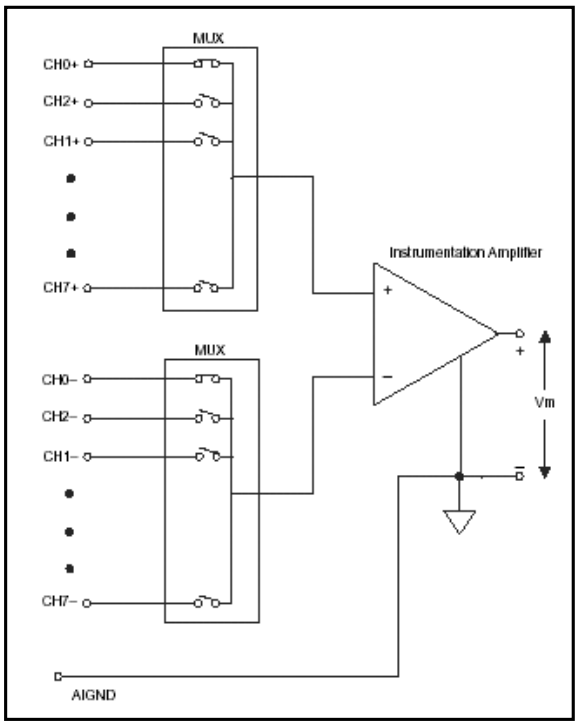
در یک سیستم اندازه‌گیری تفاضلی هیچ یک از ورودیها به مرجع ثابتی نظیر زمین ساختمان یا زمین مرجع متصل نمی‌شوند. اکثر کارت‌های DAQ به همراه یک تقویت‌کننده می‌توانند به عنوان سیستم‌های اندازه‌گیری تفاضلی مورد استفاده قرار گیرند.

در شکل ۱۹-۱۰ یک سیستم اندازه‌گیری تفاضلی با ۸ کانال نشان داده شده است که در کارت‌های سری MIO-16 به‌کار برده می‌شود. مالتی‌پلکسرهای آنالوگ به کمک یک تقویت‌کننده، تعداد کانال‌های اندازه‌گیری را افزایش می‌دهند. در مورد این کارت، پین مشخص شده با برچسب AIGND[21] همان نقطه‌ی زمین سیستم اندازه‌گیری است.

شایان ذکر است که در سیستم‌های SCXI همواره از سیستم اندازه‌گیری تفاضلی استفاده می‌شود؛ در حالی که لازم نیست در کارت‌های DAQ همواره از سیستم مذکور استفاده نمایید.

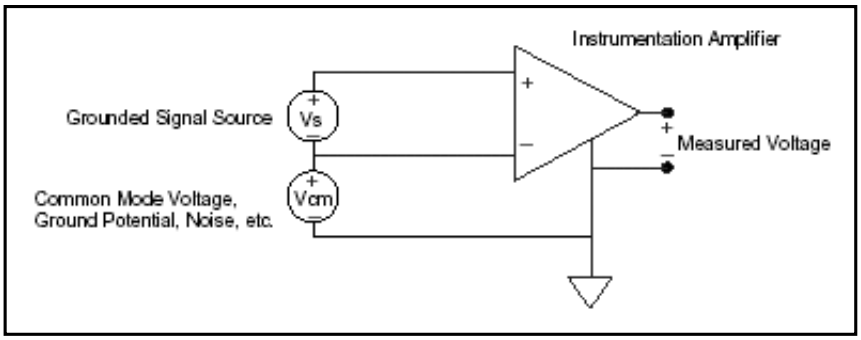
۱۰

19- Reference Signal-Ended
20- Nonreference Signal-Ended
21- Analog Input Ground




**شکل ۱۹-۱۰: یک سیستم اندازه‌گیری تفاضلی با ۸ کانال**

یک سیستم اندازه‌گیری تفاضلی ایده‌آل تنها اختلاف پتانسیل بین دو ترمینال «+» و «-» را قرائت می‌کند. اختلاف پتانسیل ورودی تقویت‌کننده نسبت به نقطه‌ی زمین تقویت‌کننده، ولتاژ مد مشترک[۲۲] خوانده می‌شود. یک سیستم اندازه‌گیری تفاضلی ایده‌آل به طور کلی ولتاژ مد مشترک را اندازه‌گیری می‌کند. در عمل، تقویت‌کننده‌های موجود، فاقد قابلیت مذکور می‌باشند. مقدار ولتاژ مد مشترک، ولتاژ مجاز ورودیها را نسبت به زمین سیستم اندازه‌گیری محدود می‌کند. مشخصات محدوده‌ی ولتاژ مد مشترک، قدرت و قابلیت کارت DAQ را در عملکرد مد تفاضلی برای حذف ولتاژهای مد مشترک تعیین می‌کند. در شکل ۲۰-۱۰ نحوه‌ی اتصال یک تقویت‌کننده به نقطه‌ی زمین را ملاحظه می‌کنید.




**شکل ۲۰-۱۰: نحوه‌ی اتصال یک تقویت‌کننده به نقطه‌ی زمین**

۱۰

22- Common Mode Voltage

ولتاژ مد مشترک یا $V_{cm}$ نسبت به نقطه‌ی زمین کارت DAQ اندازه گیری می شود. برای محاسبه‌ی سطح ولتاژ مذکور از رابطه‌ی زیر استفاده می شود:

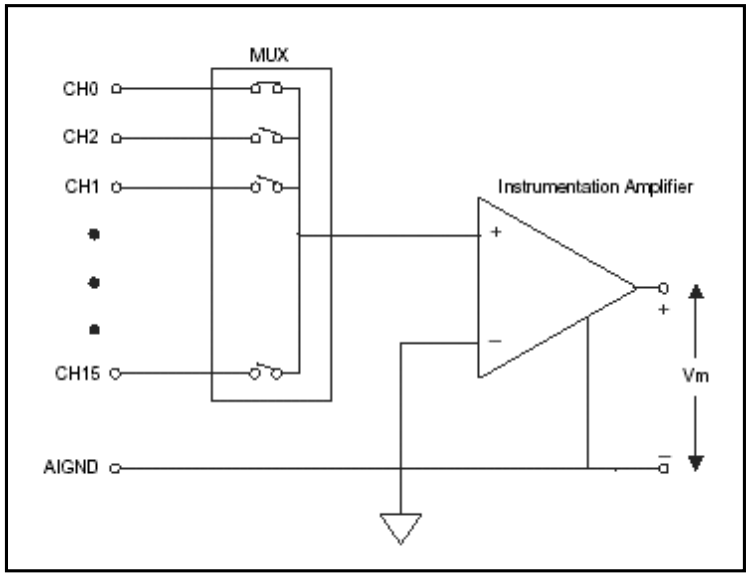
$$V_{cm} = \frac{V^+ + V^-}{2}$$

که $V^+$ و $V^-$ به ترتیب ولتاژ دو ترمینال غیر معکوس کننده و معکوس کننده در سیستم اندازه گیری نسبت به زمین تقویت کننده می باشند.

در هنگام اندازه‌گیری با یک کارت DAQ، ولتاژ مد مشترک نباید حالت اختیاری و انتخابی داشته باشد. تمامی کارت‌های DAQ دارای یک مشخصه‌ی ولتاژ عملیاتی ماکزیمم هستند که بنا به تعریف، حداکثر ولتاژ قابل تحمل کارت DAQ است، به شرطی که در اندازه‌گیری سیگنال خطایی ایجاد نشود.

## سیستم اندازه‌گیری RSE

سیستم اندازه‌گیری RSE که در برخی موارد سیستم اندازه‌گیری زمین نیز خوانده می‌شود، شبیه به یک منبع سیگنال متصل شده به زمین است که در آن، اندازه‌گیری سیگنال نسبت به زمین انجام می‌گیرد. در شکل ۲۱-۱۰ یک سیستم اندازه‌گیری RSE با ۱۶ کانال را ملاحظه می کنید.

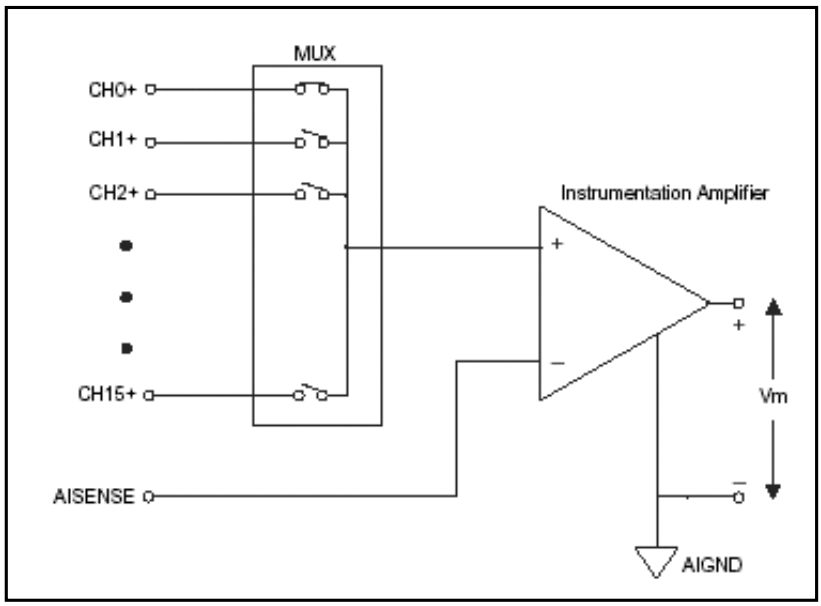



**شکل ۲۱-۱۰: یک سیستم اندازه‌گیری RSE با ۱۶ کانال**

در هنگام اتصال یک منبع ولتاژ یا جریان به کارت DAQ کاملاً مراقب باشید که منبع سینگال مورد بحث از حداکثر ولتاژ یا جریان قابل تحمل کارت DAQ تجاوز نکند. در غیر این صورت ممکن است به رایانه و کارت آسیب وارد شود.

## سیستم اندازه‌گیری NRSE

در یک سیستم اندازه‌گیری NRSE تمامی اندازه‌گیریها نسبت به یک مرجع مشترک انجام می‌گیرد. اما ممکن است نقطه‌ی مذکور با زمین مرجع یکسان نباشد. شکل ۲۲-۱۰ یک سیستم اندازه‌گیری NRSE را نشان می‌دهد. در این شکل پین‌های AISENSE و AIGND به‌ترتیب نقاط مرجع مشترک برای اندازه‌گیریها و زمین سیستم می‌باشند.




**شکل ۲۲-۱۰: یک سیستم اندازه‌گیری NRSE با ۱۶ کانال**

نحوه‌ی استفاده از کارت DAQ، نوع وچگونگی به‌کارگیری سیستم اندازه‌گیری را تعیین می‌کند. اکثر کارت‌های تولید شده توسط شرکت NI می‌توانند به کمک یک نرم‌افزار جانبی با عنوان NI-DAQ همراه با سیستم‌های اندازه‌گیری تفاضلی، RSE و NRSE مورد استفاده قرار گیرند. در برخی از کارت‌های قدیمی‌تر جهت پیکربندی باید از جابه‌جا نمودن jumper بر روی کارت استفاده نمود. هنگامی‌که یک کارت DAQ را برای یک نوع سیستم اندازه‌گیری استفاده می‌کنید تمامی کانال‌های ورودی آن از قوانین حاکم بر سیستم اندازه‌گیری مورد نظر پیروی می‌کنند.

معمولاً در سیگنال‌های متصل شده به زمین یکی از دو سیستم تفاضلی یا NRSE مورد استفاده قرار می‌گیرند. سیستم RSE نیز در مورد منابع سیگنال شناور به‌کار برده می‌شود.

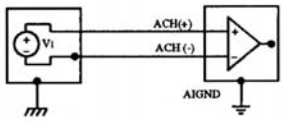
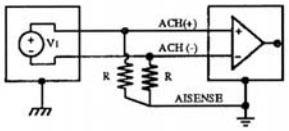
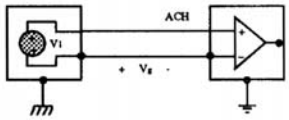
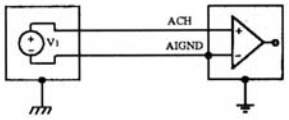
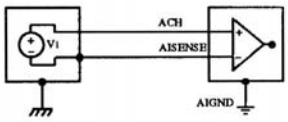
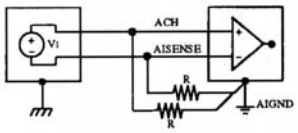
استفاده از سیستم RSE به همراه یک سیگنال متصل شده به زمین، باعث به وجود آمدن حلقه‌های زمین[۲۳] می‌گردد که ممکن است منجر به ایجاد خطا در اندازه‌گیری شود. به طریق مشابه به‌کارگیری یک سیستم تفاضلی یا NRSE به همراه یک منبع سیگنال شناور باعث ایجاد جریانهای بایاس شده که سبب می‌شود ولتاژ ورودی از محدوده‌ی مجاز و قابل تحمل کارت DAQ تجاوز کند. برای رفع این مشکل می‌توانید بین ورودی تا

۱۰

23- Ground Loops

زمین، مقاومتهای بایاس قرار دهید. در جدول ۲-۱۰ چند نمونه از روشهای اندازه گیری برای انواع سیگنال‌ها نشان داده شده است.

**جدول ۲-۱۰: چند نمونه از روشهای اندازه‌گیری**

| Input Configuration | Signal Source Type | |
|---|---|---|
| | Grounded Signal Source | Floating Signal Source (Not Connected to Building Ground) |
| | Examples<br>• Instruments with nonisolated inputs | Examples<br>• Thermocouples<br>• Signal conditioning with isolated outputs<br>• Battery devices |
| Differential (DIFF) |  |  <br>Two resistors (10 kΩ<R<100 kΩ) provide return paths to ground for bias currents |
| Single-Ended — Referenced (RSE) | NOT RECOMMENDED<br>  <br>Ground-loop losses, $V_g$, are added to measured signal |  |
| Single-Ended — Nonreferenced (NRSE) |  |  |

## تمرین ۲-۱۰: تعیین سیستم اندازه‌گیری مناسب

بهترین سیستم اندازه گیری را برای منابع سیگنال معرفی شده‌ی زیر انتخاب کنید.

۱- سیستم تفاضلی

۲- سیستم RSE

۳- سیستم NRSE

- یک دستگاه Function Synthesizer در اختیار دارید که از پریز معمولی دیوار اتاق تغذیه می شود. خروجیهای دستگاه مورد بحث و منبع تغذیه‌ی دستگاه نسبت به یک مرجع زمین سنجیده می شوند. ۱ ۲ ۳

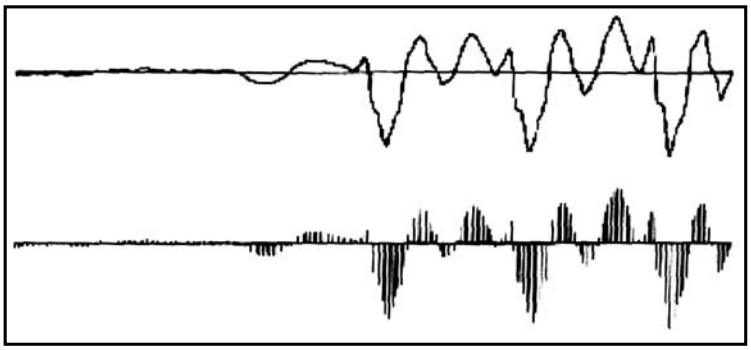
۱۰

- برای اندازه گیری میزان لرزش یک موتور از یک زوج شتاب سنج[۲۴] استفاده شده است. در این حالت مبدلها فاقد منبع تغذیه هستند. ۱ ۲ ۳
- قصد داریم ۱۴ ترموکوپل را به کارتی متصل کنیم که حداکثر ۱۶ کانال ورودی آنالوگ دارد. ۱ ۲ ۳

جهت بررسی صحت پاسخهای خود، آنها را با پاسخ صحیح این تمرین که در انتهای فصل گنجانده شده است مقایسه کنید.

## فرضیه‌ی نایکوییست

آخرین و شاید مهمترین بخش از تئوری سیگنال‌ها که در این فصل به بحث در مورد آن می‌پردازیم، روش تعیین پریود نمونه‌برداری است. همان گونه که می‌دانید سیگنال‌های موجود در دنیای طبیعی، همگی به صورت آنالوگ هستند. جهت پردازش و به نمایش در آوردن این سیگنال ها در رایانه، باید آنها را به سیگنال دیجیتال تبدیل کرد. بدین ترتیب این سیگنال‌ها برای رایانه قابل درک خواهند بود. این عمل را تبدیل آنالوگ به دیجیتال می‌گویند که با A/D نشان داده می‌شود. در مرحله‌ی بعد، رایانه نقاط گسسته را مرتب می‌کند و شکلی شبیه به سیگنال اولیه در اختیار کاربر قرار می‌دهد. در شکل ۲۳-۱۰ در قسمت بالایی، نمونه‌ای از سیگنال صحبت نشان داده شده است. در قسمت پایین شکل، فرم گسسته‌ی آن را ملاحظه می‌کنید.



**شکل ۲۳-۱۰: نمونه‌ای از سیگنال صحبت به صورت آنالوگ به همراه فرم گسسته‌ی آن**

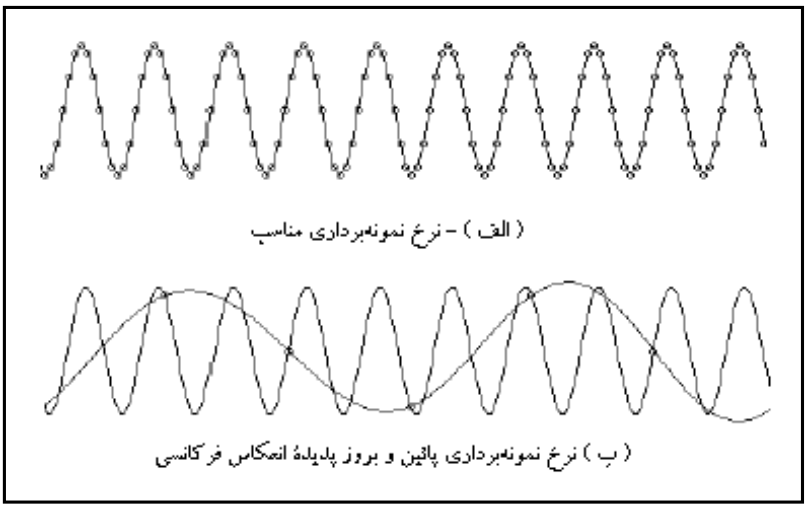
نرخ نمونه‌برداری در یک سیستم بیانگر آن است که هرچند وقت یک بار تبدیل سیگنال آنالوگ به دیجیتال روی می‌دهد. هر یک از خطوط عمودی در شکل ۲۳-۱۰ نمایش دهنده‌ی یک تبدیل A/D است. اگر سیستم DAQ در هر ثانیه دو مرتبه این تبدیل را انجام دهد، گوییم نرخ نمونه‌برداری ۲ نمونه بر ثانیه یا ۲ Hz است. در نتیجه می‌توانیم پریود نمونه‌برداری را نیز تعریف کنیم. در مثال قبلی پریود نمونه‌برداری ۰/۵ ثانیه است. واضح است که نرخ نمونه‌برداری تأثیر به سزایی در شکل سیگنال دیجیتال دارد.

هنگامی که نرخ نمونه‌برداری به اندازه‌ی کافی بالا نباشد پدیده‌ی انعکاس فرکانسی روی می‌دهد. در شکل ۲۴-۱۰ اهمیت میزان نرخ نمونه‌برداری نشان داده شده است. در شکل (الف) نرخ نمونه‌برداری به مقدار مناسب انتخاب شده است. در شکل (ب) ملاحظه می‌کنید که به دلیل انتخاب نرخ پایین در نمونه‌برداری،

۱۰

24- Accelerometer

پدیده‌ی انعکاس فرکانسی[۲۵] روی داده است.




**شکل ۲۴-۱۰: تأثیر نرخ نمونه‌برداری در بروز پدیده‌ی انعکاس فرکانسی**

پدیده‌ی انعکاس فرکانسی باعث بروز مؤلفه‌های فرکانسی بالا در داده‌ها می‌گردد که در سیگنال‌های طبیعی وجود ندارند. هنگامی که پدیده‌ی انعکاس فرکانسی در مورد داده‌ها روی دهد، هیچ راه بازگشتی وجود ندارد. به عبارت دیگر نمی‌توان انعکاس فرکانسی را از داده‌ها حذف نمود. به همین دلیل نرخ نمونه‌برداری باید به اندازه‌ی کافی بالا انتخاب گردد. حال ممکن است این سؤال در ذهن شما پیش آید که:

برای پرهیز از بروز پدیده‌ی انعکاس فرکانسی نرخ نمونه‌برداری چه اندازه باید انتخاب شود؟

شخصی به نام نایکوییست پاسخ این سؤال را یافته است. فرضیه‌ای با عنوان فرضیه‌ی نایکوییست وجود دارد که بدین ترتیب بیان می‌شود:

«جهت پرهیز از پدیده‌ی انعکاس فرکانسی، نرخ نمونه‌برداری باید بیش از دو برابر بزرگترین مؤلفه‌ی فرکانسی در سیگنال مورد بحث انتخاب گردد.»

به عنوان مثال اگر می‌دانید که حداکثر فرکانس سیگنال مورد نظر شما ۱۰۰۰ Hz است، نرخ نمونه‌برداری برای این سیگنال باید بیش از ۲ kHz باشد. در فرضیه‌ی نایکوییست فرض بر آن است که از مقدار بزرگترین مؤلفه‌ی فرکانسی موجود در سیگنال مورد اندازه‌گیری اطلاع دارید. بنابراین اطلاع داشتن از این مقدار ضروری است. در غیر این صورت بهتر است ابتدا با استفاده از یک فیلتر پایین گذر، مؤلفه های فرکانس بالا را حذف کنید.

در بسیاری از موارد، سیگنال‌ها شامل نویز، جرقه و... هستند که از حدود فرکانس اندازه‌گیری شده تجاوز می‌کنند. به عنوان مثال یک سیگنال ساده‌ی پزشکی نظیر نوار الکتروکاردیوگرام، ولتاژی است که متناسب با ضربان قلب می‌باشد. اگرچه این نوع سیگنال به ندرت دارای مؤلفه‌ای بیش از ۲۵۰ Hz است، اما سیمهای الکترود آنها به راحتی نویزهای فرکانس رادیویی در حدود ۱۰۰ kHz را به خود می‌گیرند. برای نمونه‌گیری در



25- Aliasing

فرکانس‌های بسیار بالا، سیستم DAQ از تعدادی فیلتر پایین‌گذر استفاده می‌نماید و امواج بالاتر از ۲۵۰ Hz را حذف می‌کند. در این حالت سیستم DAQ به راحتی عمل می‌کند و به عنوان مثال نمونه‌برداری را با فرکانس ۶۰۰ Hz انجام می‌دهد (۲۵۰×۲ < ۶۰۰).

در سیگنال‌های DC نظیر فشار یا دما دقت در انتخاب نرخ نمونه‌برداری از اهمیت‌چندانی برخوردار نیست. ماهیت فیزیکی سیگنال‌های مذکور به گونه‌ای است که نمی‌توانند بیش از یک یا دو بار در ثانیه تغییر کنند. در این موارد می‌توانید نمونه‌برداری را با فرکانس ۱۰ Hz انجام دهید.

## نتیجه‌ی کلی

تاکنون مطالب بسیاری در مورد مسیر پدیده‌های فیزیکی از مبدأ پیدایش کمیت مورد اندازه‌گیری تا کارت DAQ ذکر نموده‌ایم. اگر برخی از این مفاهیم را درک نکرده‌اید، جای هیچ نگرانی نیست. DAQ یک مبحث بسیار پیچیده است و برای فهم تمامی مطالب آن حتماً باید در یک محیط کاملاً مهندسی برخی از آزمایشها و تمرینات را خودتان انجام دهید.

به طور خلاصه دیدید که چگونه سیگنال‌ها طبقه‌بندی می‌شوند و در این رابطه از چه نوع مبدلهایی استفاده می‌شود. همچنین اهمیت تطبیق دادن سیگنال و تفاوت سیگنال‌های اتصال یافته به زمین و شناور و مفهوم فرضیه‌ی نمونه‌برداری نایکوییست را درک نمودید. جهت درک تمامی مفاهیم و مطالب عنوان شده در مورد جمع‌آوری داده‌ها و اندازه‌گیری آنها باید چندین واحد درسی از مجموعه‌ی مهندسی برق را بگذرانید. در این بخش، تنها راه را برای مطالعه‌ی شما هموار ساخته‌ایم. همین مطالب برای آغاز عملیات اندازه‌گیری کافی است. برای کسب اطلاعات در این زمینه، مطالب عنوان شده در فایل Manuals >> DAQbasic.pdf را از لوح فشرده‌ی همراه این کتاب مطالعه کنید.

## نحوه‌ی انتخاب و به‌کارگیری یک کارت DAQ

### انتخاب سخت‌افزار

پس از به‌دست آوردن اطلاعات کافی در مورد سیگنال‌هایی که قصد جمع‌آوری یا تولید آنها را دارید زمان آن فرا می‌رسد تا یک کارت DAQ انتخاب کنید. ما استفاده از محصولات شرکت NI را پیشنهاد می‌کنیم. در ضمن استفاده از کارت‌های تولید شده توسط شرکتهای دیگر به همراه نرم‌افزار LabVIEW نیز امکان‌پذیر است. به دلیل کثرت کارت‌ها و خدمات جانبی توسط شرکت NI موارد انتخاب زیادی پیش روی شما قرار دارد که می‌توانید با در نظر گرفتن هزینه‌ی تهیه‌ی کارت و نوع سیستم، کارت مطلوب را انتخاب نمایید.

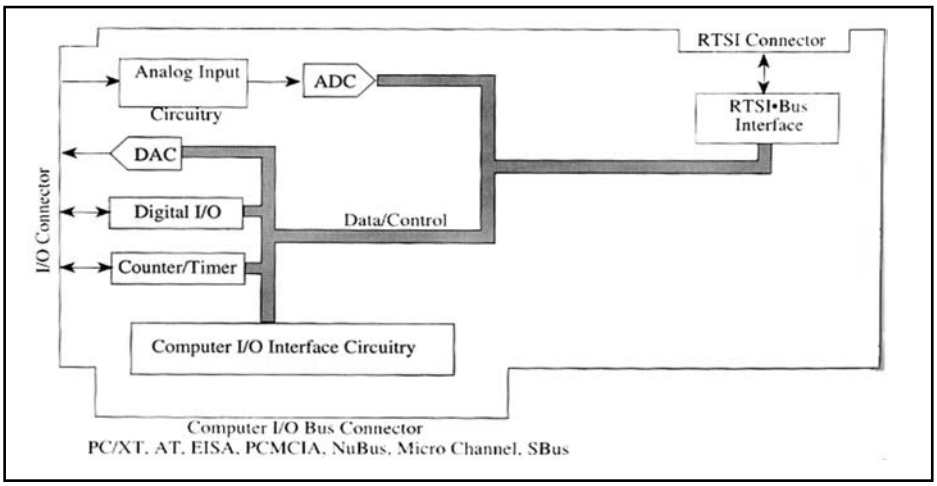
ابزار دیگری که ممکن است به کمک آن بتوانید کارت موردنظر را انتخاب کنید، DAQ Designer نام دارد. این ابزار یک نرم‌افزار است که پس از دریافت اطلاعاتی در مورد سیستم و نیازهایتان، سخت‌افزار مناسب را پیشنهاد می‌کند. اما تنها نسخه‌ی تحت Windows این نرم‌افزار عرضه شده است.

۱۰

جهت انتخاب سخت‌افزار بهینه برای سیستم باید نیازهای سیستم خود را به خوبی درک کنید. به خصوص تعداد ورودیها و خروجیهای آن را در نظر داشته باشید. در لیست زیر تقریباً تمامی اطلاعاتی که باید جهت انتخاب یک کارت مناسب از سیستم خود در اختیار داشته باشید وجود دارد.

- از چه نوع سیستمی استفاده می کنید؟ (HP-UX، Sun، Windows یا Mac)
- چه نوع باس یا کانکتوری در دسترس است؟ (EISA یا AT برای سیستم های X86، PCI یا NuBus در سیستم های Mac، PCMCIA در مورد Laptopها و...)
- تعداد ورودیهای آنالوگ مورد نیاز چه اندازه است؟ (در صورت نیاز به ورودی‌های تفاضلی این تعداد باید دو برابر شود)
- تعداد خروجیهای آنالوگ موردنیاز چه اندازه است؟
- آیا ورودیهای آنالوگ به صورت سیگنال جریان یا ولتاژ هستند؟ حدود این سیگنال ها چه مقدار است؟
- چه تعداد ورودی و خروجی دیجیتال مورد نیاز است؟
- آیا نیاز به سیگنال زمان‌سنج یا شمارشگر وجود دارد؟ اگر پاسخ مثبت است تعداد آنها چه اندازه است؟
- آیا تطبیق دادن سیگنال های آنالوگ ورودی/خروجی الزامی است؟ (به عنوان مثال سیگنال های دما، فشار و...)
- آیا سیگنال های آنالوگ ورودی/خروجی از 10 V یا 20 mA تجاوز می کنند؟
- کمترین نرخ اسکن برای تمام کانال ها چقدر است؟
- درجه‌ی وضوح مورد نیاز چه اندازه است؟
- آیا هزینه‌ی تهیه‌ی کارت و لوازم جانبی، ایزوله سازی و... در انتخاب سیستم تأثیرگذار هستند؟ در صورت مثبت بودن پاسخ، تأثیر آنها تا چه حدی است؟
- آیا نیازهای خود را، به عنوان مثال، گسترش قابلیت استفاده از سیستم های DAQ جدیدتر را در آینده در نظر گرفته اید؟

پس از شناخت سیستم و همچنین سیگنال مورد نظر، برای انتخاب کارت DAQ آمادگی دارید. عمومی ترین محصولات شرکت NI کارت های MIO می باشند. از این سری کارت ها می توان کارت AT-MIO-16E-10 را نام برد که شامل ۱۶ کانال ورودی آنالوگ، ۲ کانال خروجی آنالوگ، ۸ کانال I/O دیجیتال و ۲ شمارنده است. کارت مذکور شامل ترکیبات متفاوتی از مبدلهای ADC و DAC، خطوط دیجیتال و مدارات تایمر/شمارنده است که آن را برای کاربردهای مختلفی آماده می سازد. اتصالات داخلی کارت، سیگنال های لازم برای اعمال تریگر و زمان‌بندی را در بین کارت ها انتقال می‌دهند و عملکرد چندین کارت را به طور همزمان امکان پذیر می سازند. در شکل ۲۵-۱۰ شمای بلوکی یک کارت DAQ چندمنظوره را ملاحظه می کنید.




**شکل ۲۵-۱۰: شمای بلوکی یک کارت DAQ چندمنظوره**

این کارت ها در مواردی که چندین ورودی آنالوگ و گاهی یک خروجی یا سیگنال دیجیتال مورد احتیاج باشد، نیازهای شما را به خوبی برآورده می سازند. در صورت نیاز به خروجیهای آنالوگ بیشتر می توانید از کارت هایی نظیر AT-AO-10 استفاده کنید. این کارت شامل ۱۰ خروجی آنالوگ می باشد. همچنین می توانید از کارت های زمان بندی، کارت های با تعداد I/O بالا، کارت های PCMCIA برای Laptopها و تطبیق دهنده های درگاههای موازی و یا هر ترکیب یا قابلیتی که مدنظرتان است استفاده نمایید. بخش ورودی آنالوگ در کارت MIO شامل مبدل ADC و مدارات ورودی آنالوگ می باشد. مدار ورودی آنالوگ شامل مالتی پلکسرهای آنالوگ، تقویت کننده ها و مدارات نمونه بردار و ذخیره ساز[۲۶] است. جزییات لازم در مورد هر یک از کارت ها در دفترچه‌ی راهنمای آن وجود دارد. علاوه بر بخشهایی که در شکل ۲۵-۱۰ ملاحظه نمودید، ممکن است در کارت های DAQ قدیمی تر اجزای دیگری نظیر jumper و dipswitch وجود داشته باشد. این اجزا برای تعیین پارامترهای سخت افزاری کارت نظیر آدرس پین ها، کانال های DMA و سطوح وقفه به کار برده می شوند. در بخش بعد اجزای مذکور را مورد بررسی قرار خواهیم داد.

در مورد قیمت کارت چگونه برخورد می کنید؟ قیمت اکثر کارت های DAQ بین ۷۰۰ تا ۲۰۰۰ دلار است. به طور کلی دو عامل نرخ نمونه برداری و تعداد کانال های ورودی/خروجی آنالوگ به صورت مستقیم در قیمت یک کارت تأثیرگذار هستند.

اگر سیگنال مورد بحث از نوع DC باشد نیازی نیست تا هزینه‌ی اضافی متحمل شوید و به عنوان مثال کارتی با نرخ نمونه برداری ۱MHz انتخاب نمایید که مسلماً به دلیل قابلیت نمونه برداری با نرخ بالا گران قیمت است. کارت های حاوی ورودی/خروجی دیجیتال اغلب ارزان قیمت هستند. اگر به عنوان مثال برای یک آزمایشگاه دانشجویی به دنبال کارت های مقرون به صرفه و ارزان قیمت هستید می توانید از محصولات شرکت NI نظیر Lab-PC+ استفاده کنید.

۱۰

26- Sample-and-Hold Circuitry

در مقابل خدمات ارائه شده توسط کارت به شما، قیمت کارت چندان بالا نیست. اگر مجبور باشید وسایل اندازه‌گیری نظیر اسلیوسکوپ، طیف‌نگار و صدها وسیله و ابزار اندازه‌گیری حقیقی و غیرمجازی را تهیه نمایید، هزینه‌ای که برای این کار اختصاص می‌دهید چه اندازه است؟

اگر پس از مطالعه و بررسی دفترچه‌ی راهنمای کارت باز هم در انتخاب و خرید کارت مردد هستید، در لیست زیر چند عامل اساسی ذکر شده که شما را در انتخاب و تصمیم‌گیری یاری می‌دهد.

۱- دقیقاً نمی‌دانم که سیستم مورد نظر من به چه کارتی نیاز دارد. یک کارت قوی، قابل انعطاف و با قیمت مناسب برای شروع کار نیاز دارم.

◆ در این مورد می‌توانید یکی از کارت‌های چند منظوره نظیر AT-MIO-16E-10 (برای سیستم‌های X86)، NB-MIO-16 (در باس NuBus در Mac) و یا PCI-MIO-16XE-50 (در محیط Mac) را به‌کار برید. در این کارت‌ها تعداد ورودیهای آنالوگ زیاد است. اما موارد دیگر نظیر تعداد خروجی آنالوگ، ورودی/خروجی دیجیتال و شمارنده کمتر می‌باشند.

۲- تعداد کانال‌های آنالوگ که برای نمونه‌برداری نیاز دارم بسیار زیاد است (در حدود ۱۶ تا ۲۵۶ کانال).

◆ در این مورد می‌توانید یک کارت استاندارد MIO تهیه نمایید و یک مالتی‌پلکسر آنالوگ خارجی به آن اضافه کنید. این مالتی‌پلکسر هر یک از ورودیهای کارت را به ۶۴ کانال افزایش می‌دهد. توجه داشته باشید که در این حالت و با توجه به استفاده از مالتی‌پلکسر، از مقدار حداکثر نرخ نمونه‌برداری برای این کانال‌ها کاسته خواهد شد.

۳- تعداد کانال‌های آنالوگ مورد نیاز برای نمونه‌برداری بسیار زیاد است. (از ۲۵۶ تا ۴۰۹۶ کانال)

◆ با این تعداد کانال حتماً باید یک سیستم SCXI را به‌کار برید. استفاده از سیستم SCXI در مواردی که تعداد کانال‌های آنالوگ نمونه‌برداری بسیار زیاد و تطبیق دادن سیگنال‌های مورد نظر الزامی باشد، ایده‌آل و مفید است.

۴- به یک سیستم DAQ قابل حمل و نقل نیاز دارم که آن را به همراه Laptop مورد استفاده قرار دهم.

◆ یکی از کارت‌های DAQ سری PCMCIA (نظیر DAQ Card 1200) را انتخاب کنید.

۵- برای یک سیستم بسیار ساده تنها به یک زوج کانال ورودی/خروجی نیاز دارم.

◆ یکی از کارت‌های ارزان قیمت نظیر PC-AO-2DC (با ۲ کانال آنالوگ خروجی) یا کارت‌های I/O چندمنظوره برای باس NuBus در محیط Mac نظیر Lab-NB را به‌کار برید.

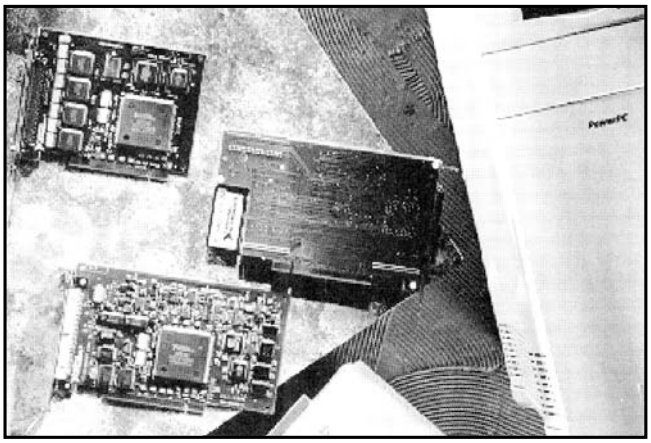
۶- برای یک سیستم کنترل به تعداد زیادی سیگنال دیجیتال نیاز دارم.

◆ در این حالت می‌توانید از کارت‌هایی نظیر PC-DIO-96 یا NB-DIO-96 که ۹۶ خط دیجیتال در اختیار شما قرار می‌دهند استفاده نمایید.

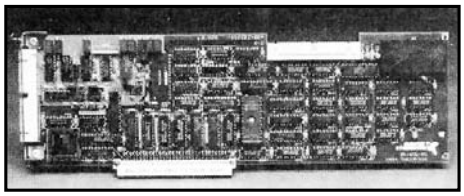
۷- بنا به دلایلی خاص نمی‌توانم از هیچ نوع کارتی استفاده کنم.

◆ شاید لازم باشد تا سیستم رایانه‌ی خود را تغییر دهید. اما شرکت NI سیستمی تعبیه نموده است که داده‌ها را از طریق درگاه موازی و به کمک کارتی نظیر DAQ Pad-1200 به رایانه انتقال می‌دهد.

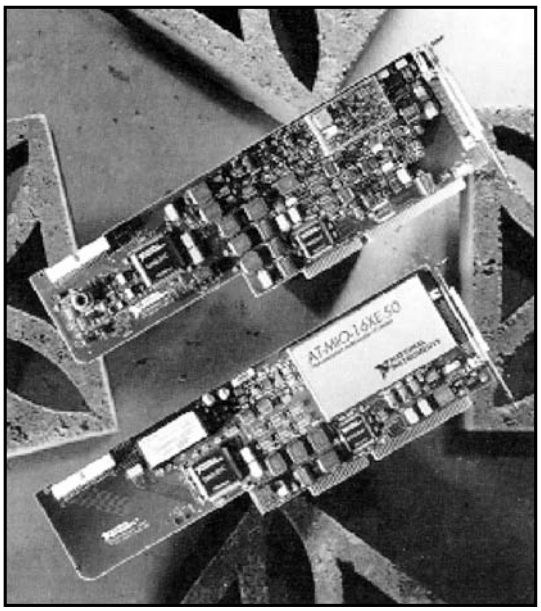
توجه داشته باشید که موارد مذکور تنها به عنوان راهنمای عمومی بوده و نیازهای خاص هر شخص و هر سیستم متفاوت است. قبل از تهیه هر کارت به شما توصیه می‌کنیم تا با شرکت NI تماس حاصل کنید و در مورد انتخاب بهترین سخت‌افزار از مشاوران این شرکت کمک بگیرید. در شکلهای زیر چند نمونه از محصولات این شرکت را ملاحظه می‌کنید.



**شکل ۲۶-۱۰: کارت‌های سری PCI-MIO، کارت‌های چندمنظوره هستند که در محیط Mac به همراه باس PCI استفاده می‌شوند.**



**شکل ۲۷-۱۰: کارت‌های سری NB-MIO، کارت‌های چندمنظوره هستند که در محیط Mac به همراه باس Nu Bus مورد استفاده قرار می‌گیرند.**




**شکل ۲۸-۱۰: کارت‌های سری AT-MIO، کارت‌های چندمنظوره هستند که در محیط Windows به همراه باس AT به کار می‌روند.**

محصولات سازندگان دیگر تنها در صورتی به همراه نرم‌افزار LabVIEW به درستی عمل می‌کنند که راه‌انداز کارت در LabVIEW به همراه کارت DAQ ارائه شده باشد. قبل از تهیه‌ی کارت مورد نظر ابتدا مشخصات آن را با نیازهای سیستم خود مقایسه کنید. اسامی، مشخصات و قابلیتهای کارت‌های مختلف در فایل Appendix 3 << Appendices آورده شده است.

## تمرین ۳-۱۰: نمونه‌برداری

در این تمرین دو مورد از مسائل اندازه‌گیری و جمع‌آوری داده‌ها آورده شده است. وظیفه‌ی شما در این تمرین تعیین اطلاعات مورد نیاز است. برای هر مورد به سئوالات زیر به طور کامل پاسخ دهید:

۱- نوع سیگنال مورد اندازه‌گیری کدام است؟

۲- کدام یک از روشهای اندازه‌گیری و جمع‌آوری سیگنال را توصیه می‌کنید؟

۳- برای این سیگنال‌ها فرکانس نایکوییست و نرخ نمونه‌برداری را تعیین نمایید.

۴- آیا تطبیق دادن سیگنال‌ها الزامی است؟ در صورت مثبت بودن پاسخ، نوع تطبیق سیگنال را بیان کنید.

۵- کارت یا کارت‌هایی از دفترچه‌ی راهنمای شرکت NI که نیازهای شما را در مورد این سیگنال برآورده می‌سازد انتخاب کنید. در مورد انتخاب کارت حتماً مطابقت با سیستم[۲۷] را در نظر بگیرید.

الف) پروفسور هَری فِیس در یک آزمایشگاه مهندسی پزشکی قصد دارد سیگنال‌های ضربان قلب انسان را تجزیه و تحلیل کند. وی می‌خواهد نوار ضربان قلب را در هر لحظه از دو نقطه تهیه نماید. در هر یک از نقاط نیز چهار الکترود در مواضع مختلف به بدن متصل شده است. هدف از انجام این آزمایش، اندازه‌گیری اختلاف پتانسیل بین هر یک از سه الکترود مذکور با چهارمین الکترود است که در این حالت به عنوان مرجع در نظر گرفته می‌شود. سیمهای الکترود متصل به کارت DAQ ایزوله نیستند و فاقد پوسته‌ی زمین[۲۸] می‌باشند. حداکثر پهنای موج در حد 2 ms است. محدوده‌ی ولتاژ سیگنال‌ها در فاصله‌ی 0.02-1 mV می‌باشد.

ب ) آقای دکتر جانسون قصد دارد تا مقاومت یک ماده‌ی قابل انعطاف را تحت فشار و دمای بالا اندازه‌گیری کند. برای انجام این آزمایش یک اتاقک در نظر گرفته شده است که درجه حرارت داخل آن نیز قابل کنترل است و شبیه به فرگاز عمل می‌کند. در داخل این اتاقک دستگاهی تعبیه شده است که ماده‌ی قابل انعطاف مذکور را پیوسته تحت فشار قرار داده، آن را می‌پیچاند یا می‌کشد.

وی ۴۸ رشته از این ماده را در داخل اتاقک قرار داده است و قصد دارد تا مقاومت تمامی آنها را در شرایط موجود در اتاقک بررسی کند. مقاومت هر رشته با اعمال یک سطح ولتاژ مشخص به هر یک از رشته‌ها و سپس اندازه‌گیری جریان آنها محاسبه می‌گردد. دمای اتاق توسط یک ترموکوپل، اندازه‌گیری می‌شود و خاموش و روشن شدن دستگاه اعمال‌کننده‌ی فشار نیز توسط یک رله صورت می‌گیرد. در ضمن تعداد سیکل‌های دستگاه اعمال‌کننده‌ی فشار نیز باید اندازه‌گیری شود.

پاسخهای صحیح در انتهای فصل گنجانده شده‌اند.

۱۰

۲۷- مطابقت با سیستم به معنی به کارگیری کارت‌های متناظر با سیستم‌های Mac، Windows و... است.

28- Ground Shield

## روش نصب کارت

تمامی کارت‌هایی که بر روی کارت اصلی رایانه نصب می‌شوند به نرم‌افزار راه‌انداز نیاز دارند. خوشبختانه به همراه محصولات شرکت NI، نرم‌افزار راه‌انداز نیز وجود داردکه به همراه بسته‌ی نرم‌افزاری LabVIEW نصب می‌شوند. نرم‌افزار راه‌انداز برای محصولات شرکت NI، NI-DAQ نام دارد. در مورد کارت‌های GPIB این نرم‌افزار راه‌انداز با عنوان NI-488.2 عرضه می‌گردد. در صورت خرید هر کارت، حتماً آخرین نسخه از نرم‌افزار راه‌انداز آن را نیز دریافت کنید.

همان گونه که قبلاً عنوان شد کارت‌های قدیمی‌تر در محیط Windows به صورت Plug & Play نیستند و ممکن است در این کارت‌ها تعدادی jumper وجود داشته باشد که به منظور آدرس‌دهی و تعیین سطوح وقفه تنظیم می‌شوند. تمامی کارت‌های سری E تحت سیستم عامل Win 95 و همچنین تمامی کارت‌های مورد استفاده در محیط Mac خود را به صورت خودکار پیکربندی می‌کنند. در هر صورت باید دستورالعمل موجود در دفترچه‌ی راهنمای کارت را با دقت مطالعه و دنبال نمایید و نرم‌افزار NI-DAQ را برای اطمینان از شناسایی کارت‌ها توسط سیستم اجرا کنید. نرم‌افزار NI-DAQ به هر یک از کارت‌ها یک شماره‌ی منحصر به فرد اختصاص می‌دهد. این شماره توسط بسته‌ی نرم‌افزاری LabVIEW جهت شناسایی کارت در برنامه‌های DAQ مورد استفاده قرار می‌گیرد.

نرم‌افزارهایی که جهت پیکربندی کارت‌های DAQ به‌کار برده می‌شوند با سرعت چشمگیری در حال پیشرفت می‌باشند. تمامی نرم‌افزارهایی که در این بخش معرفی می‌شوند حاصل تحول در مقطع زمانی کوتاه و فشرده‌ای بوده‌اند. به شما توصیه می‌کنیم دفترچه‌ی راهنمای کارت DAQ را با مشخصات و دستورالعملهای موجود در این فصل و فایل Appendix 3 << Appendices مقایسه کنید. زیرا ممکن است برخی از خصوصیات کارت‌ها به خصوص در محیط Windows تغییر کرده باشند.

## پارامترهای مرتبط با رایانه در مورد کارت DAQ

در هر یک از کارت‌های DAQ، سه پارامتر برای تعیین نحوه‌ی برقراری ارتباط با رایانه در نظر گرفته شده است. این پارامترها عبارتند از: آدرس I/O، سطوح وقفه و کانال‌های DMA. در ادامه به بررسی پارامترهای مذکور می‌پردازیم:

## آدرس I/O

کارت DAQ ابتدا از طریق ثباتهای داخلی خود با رایانه ارتباط برقرار می‌سازد. نرم‌افزار راه‌انداز، اطلاعاتی را به منظور پیکربندی کارت بر روی ثباتهای تعبیه شده برای این منظور ثبت می‌کند. سپس این نرم‌افزار اطلاعات و داده‌های ثبات را به منظور به‌دست آوردن موقعیت و حالت کارت یا جهت اندازه‌گیری سیگنال از کارت قرائت می‌کند. آدرس I/O محل قرار گرفتن ثباتهای کارت را در I/O های رایانه تعیین می‌کند.

۱۰

## سطوح وقفه

روش دیگری که از آن طریق، کارت DAQ با رایانه ارتباط برقرار می‌سازد «وقفه های پردازنده» است. همان گونه که می‌دانید در عملکرد پردازشگرها، وقفه از اهمیت به سزایی برخودار است. وقفه‌ها توانایی پاسخ دادن و نشان دادن عکس‌العمل سریع در مقابل اولویتها را به پردازنده می‌دهند.

وقفه‌ی پردازنده را به عنوان زنگ درب منزل در نظر بگیرید. در صورتی که برای درب منزل شما زنگ تعبیه نشده باشد مجبور هستید که هر از گاهی به درب منزل مراجعه کنید و ببینید که آیا کسی به منزل شما مراجعه کرده است یا خیر؟ در صورت وجود زنگ، تنها در زمان به صدا در آمدن آن به درب منزل مراجعه می‌کنید و مطمئن هستید که شخصی در پشت درب منتظر شماست.

به طریق مشابه و در یک کارت DAQ نیز معقول به نظر نمی‌رسد که پردازنده‌ی رایانه همواره وجود و فراهم بودن داده های کارت DAQ را به منظور قرائت آنها بررسی کند. یک کارت DAQ می تواند از وقفه به عنوان زنگ درب منزل استفاده نماید و بدین روش، آماده بودن تمامی شرایط برای خواندن داده‌ها را اعلام کند. هر یک از کارت‌هایی که از وقفه‌های پردازشی استفاده می‌کنند باید دارای سطح وقفه‌ی متفاوتی باشند. در غیر این صورت در قرائت داده‌ها اغتشاش و سردرگمی به وجود می آید.

## کانال‌های DMA

سومین روش برقراری ارتباط بین کارت DAQ با رایانه، از طریق DMA است. استفاده از DMA یکی از روشهای انتقال اطلاعات است که در آن، داده‌ها مستقیماً به حافظه‌ی رایانه ارسال می‌شوند. در این حالت پردازنده از میان برداشته می‌شود و در اصطلاح، بایپاس می‌گردد. در DMA انتقال داده‌ها با حداکثر سرعت انجام می‌شود. بنابراین در مواردی که زمان، به عنوان یکی از عوامل تعیین کننده در انتقال داده‌ها مد نظر باشد، به کارگیری کارت‌هایی با قابلیت DMA بسیار مفید است. به هر یک از کارت‌هایی که از قابلیت DMA استفاده می‌کنند باید یک کانال DMA به صورت مجزا اختصاص داده شود. به برخی از کارت‌ها از قبیل کارت‌های سری MIO می‌توان بیش از یک کانال DMA اختصاص داد.

## پیکربندی کارت DAQ

بسته به نوع سیستمی که از آن استفاده می‌کنید پارامترهای فوق‌الذکر به روشهای زیر تنظیم می‌شوند:

در مورد باس PC/XT/AT و در کارت‌های DAQ که از نوع Plug & Play نیستند جهت تنظیم پارامترهای آدرس I/O، سطوح وقفه و کانال‌های DMA، از jumper یا dipswitch استفاده می‌شود. تنظیم موارد مذکور باید به صورتی انجام گیرد که تنظیمات فعلی برای jumper یا dipswitch ها توسط سخت افزار دیگری اشغال نشده باشد. در صورت تغییر دادن تنظیمات، همین تغییرات را باید در مورد نرم‌افزار پیکربندی نیز اعمال کنید. در صورتی که کارت DAQ از نوع Plug & Play باشد این تنظیمات تنها بر روی نرم‌افزار انجام می‌گیرد.

در باس NuBus/PCI در محیط Mac، آدرس I/O، سطوح وقفه و کانال DMA به صورت خودکار پیکربندی می‌شوند. در این محیط تنها کافی است شماره‌ی شکافی از کارت اصلی رایانه که کارت DAQ در آن نصب شده است را بدانید. برای دستیابی به شماره‌ی شکاف مذکور از نرم‌افزار NI-DAQ استفاده کنید.

## تنظیم ورودی/خروجی آنالوگ

در هر یک از کارت‌های DAQ چند پارامتر ورودی/خروجی آنالوگ وجود دارد که عملکرد مبدلهای ADC و DAC را کنترل می‌کنند. در برخی از کارت‌ها تنظیم این پارامترها تنها از طریق نرم‌افزار صورت می‌گیرد، در حالی که در کارت‌های دیگر، تنظیمات مذکور از طریق jumper انجام می‌شود. نرم‌افزارهایی که اغلب برای تنظیم پارامترهای مذکور در کارت‌هایی که دارای این قابلیت هستند به‌کار برده می‌شوند، به ترتیب در سیستم‌های Win 95، Windows 3.1 و Mac عبارتند از: NI-DAQ Configuration Utility، WDAQCONF و NI-DAQ. در مورد کارت‌هایی که تنظیمات مذکور از طریق jumper صورت می‌گیرد، علاوه بر تغییر موقعیت jumper‌ها، تنظیمات مذکور باید در مورد نرم‌افزار نیز انجام گیرد.

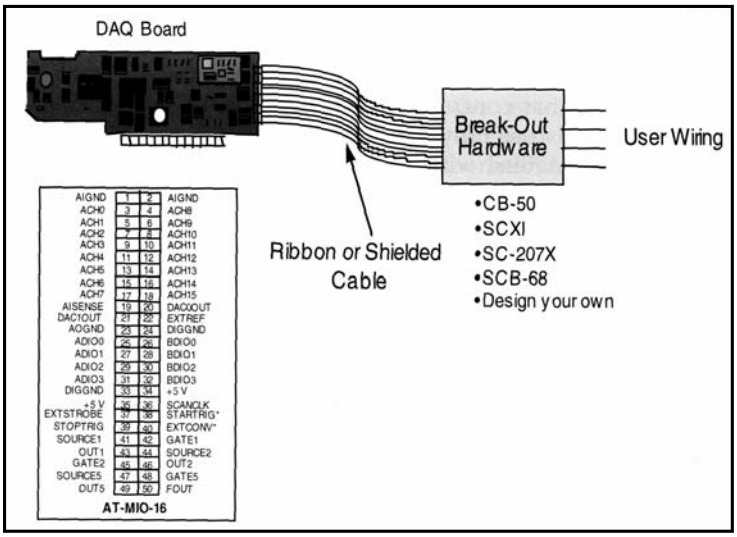
به عنوان مثال برای پیکربندی هر یک از کارت‌های AT-MIO-16 یا NB-MIO-16 تنظیمات لازم هم بر روی نرم‌افزار و هم در مورد jumper‌ها صورت می‌گیرد. در لیست زیر پارامترهای مذکور در مورد ورودی/خروجی آنالوگ را ملاحظه می‌کنید:

- ADC Input Range
  - Unipolar 0 V to +10 V
  - Bipolar +-5 V
  - Bipolar +-10 V (factory setting)
- ADC Input Mode
  - Ground-referenced signal-ended
  - Nonreferenced signal-ended
  - Differential (factory setting)
- DAC Reference
  - Internal (factory setting)
  - External
- DAC Polarity
  - Unipolar-Straight binary mode
  - Bipolar-Twos complement mode (factory setting)

برعکس در کارت AT-MIO-16E-2 که از نوع Plug & Play می‌باشد، این تنظیمات تنها در نرم‌افزار انجام می‌گیرد.

## اتصال سیگنال

جهت اتصال سیگنال به کارت DAQ می‌توانید از کانکتورهایی نظیر CB-50 یا SC-207X استفاده کنید. در صورت استفاده از سیستم SCXI ترمینال متناسب با آن را به‌کار برید. در شکل ۲۹-۱۰ اتصالات کارت AT-MIO-16 را ملاحظه می‌کنید.

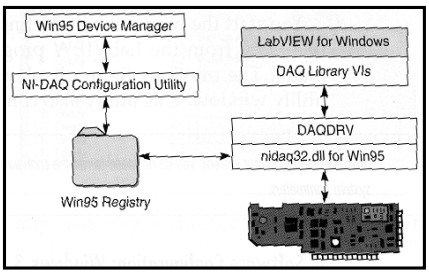



شکل ۲۹-۱۰

## نرم‌افزار پیکربندی در Win95 [۲۹]

بسته‌ی نرم‌افزاری LabVIEW در مورد برنامه‌های DAQ مطابق شکل ۳۰-۱۰ از یک نرم‌افزار راه‌انداز با نام DAQDRV جهت دستیابی به DLL استاندارد ۳۲ بیتی در Win 95 استفاده می‌کند. نرم‌افزار NI-DAQ در Win 95 تمامی محصولات شرکت NI اعم از کارت‌های AT/XT/EISA و سیستم SCXI را پشتیبانی می‌کند.

بخش مدیریت پیکربندی در Win 95 تمامی سخت‌افزارهای نصب شده در رایانه از قبیل کارت‌های DAQ را تشخیص می‌دهد و وجود یا عدم وجود آنها را بررسی می‌کند. در صورتی که یک کارت Plug & Play در اختیار دارید با استفاده از گزینه‌ی Add New Hardware در Control Panel، آن را به صورت دستی تعریف کنید. در شکل ۳۰-۱۰ شمای بلوکی پیکربندی کارت DAQ را در Win 95 ملاحظه می‌کنید.

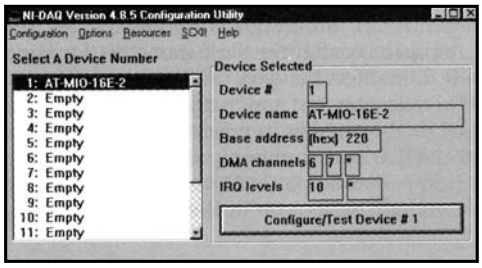



**شکل ۳۰-۱۰: شمای بلوکی پیکربندی کارت DAQ در Win 95**

نرم‌افزار جانبی NI-DAQ Configuration Utility برنامه‌ای است که جهت تعریف پارامترها در کارت DAQ مورد استفاده قرار می‌گیرد. پس از نصب کارت DAQ بر روی کارت اصلی رایانه و پیکربندی کارت توسط بخش مدیریت پیکربندی Win 95، باید این برنامه را اجرا کنید. برنامه‌ی مذکور اطلاعات ثبت شده در ثبات را می‌خواند و یک شماره به هر یک از کارت‌های DAQ اختصاص می‌دهد. برای مراجعه به کارت در LabVIEW، از این شماره استفاده می‌شود. در شکل ۳۱-۱۰ نمونه‌ای از پنجره‌ی محاوره‌ای این برنامه نشان داده شده است.

۱۰

29- Windows 95 Configuration Manager

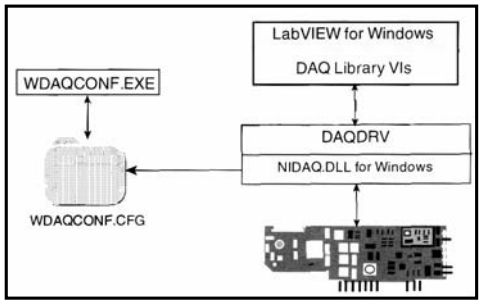



**شکل ۳۱-۱۰: پنجره‌ی محاوره‌ای نرم‌افزار NI-DAQ Configuration Utility**

پارامترهای مربوط به کارت که به کمک این نرم‌افزار جانبی تنظیم می‌شوند به کارت بستگی دارند. همان گونه که عنوان شد برخی از کارت ها کاملاً به صورت نرم افزاری پیکربندی می‌شوند، حال آن که تنظیم و تعریف پارامترها در برخی دیگر مستلزم تنظیم jumperهاست. این برنامه‌ی جانبی شماره‌ی کارت و پارامترهای پیکربندی شده را در یک ثبات ذخیره می کند. برای پیکربندی کارت، برنامه‌ی NI-DAQ Configuration Utility را از منوی Start >> Programs >> LabVIEW >> NI-DAQ اجرا نمایید. پس از پیکربندی پارامترهای مورد نظر، اجرای مجدد این برنامه‌ی الزامی نیست مگر اینکه قصد داشته باشید پارامترهای کارت را تغییر دهید.

## نرم‌افزار پیکربندی در Windows 3.1

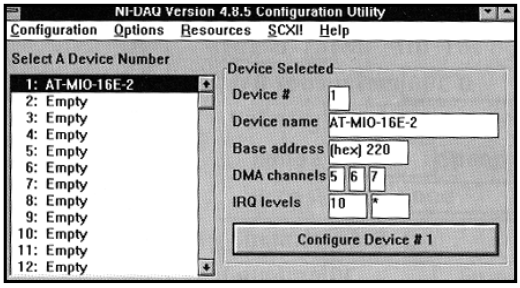
بسته‌ی نرم‌افزاری LabVIEW در مورد برنامه‌های DAQ مطابق شکل ۳۲-۱۰ از یک نرم‌افزار راه‌انداز با نام DAQDRV جهت دستیابی به DLL استاندارد در Windows استفاده می کند. برنامه‌ی جانبی NI-DAQ تمامی محصولات شرکت NI از قبیل کارت های AT/XT/EISA و سیستم SCXI را پشتیبانی می کند. در شکل ۳۲-۱۰ شمای بلوکی پیکربندی کارت DAQ را در Windows 3.1 ملاحظه می کنید.




**شکل ۳۲-۱۰: شمای بلوکی پیکربندی کارت DAQ در Windows 3.1**

نرم افزار WDAQCONF یک برنامه‌ی جانبی است که به هر کارت یک شماره‌ی منحصر به فرد اختصاص می‌دهد. به کمک این شماره می‌توان به پیکربندی پارامترهای کارت DAQ پرداخت. این پارامترها به نوع کارت بستگی دارند. پس از انتساب یک شماره‌ی منحصر به فرد به هر کارت، برای ارجاع به آن کارت از شماره‌ی آن استفاده می شود. این برنامه پارامترهای پیکربندی و شماره‌ی کارت ها را در یک فایل با نام WDAQCONF.CFG ذخیره می کند. همان گونه که عنوان شد از برنامه‌ی WDAQCONF می توان برای پیکربندی سیستم SCXI نیز استفاده نمود. نمونه‌ای از پنجره‌ی این برنامه در شکل ۳۳-۱۰ نشان داده شده است.

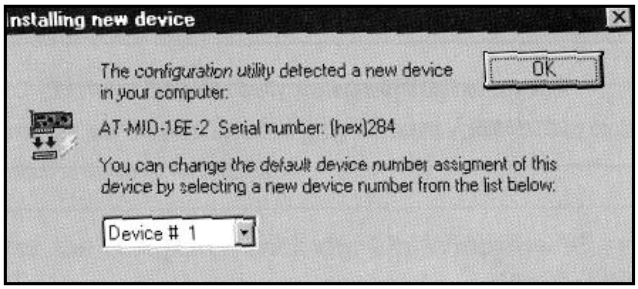
۱۰




**شکل ۳۳-۱۰: پنجره‌ی محاوره‌ای نرم‌افزار WDAQCONF**

پس از پیکربندی پارامترهای موردنظر، اجرای مجدد برنامه‌ی WDAQCONF الزامی نیست مگر اینکه قصد داشته باشید پارامترهای کارت را تغییر دهید.

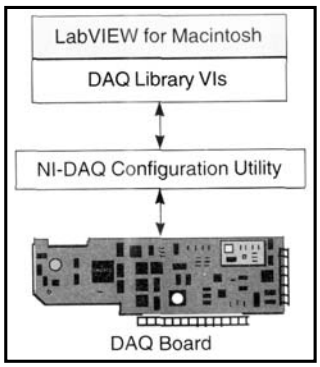
NI-PNP یک برنامه‌ی جانبی است که توسط WDAQCONF و برای انتخاب و پیکربندی کارت‌های Plug & Play به طور خودکار اجرا می گردد. پس از نصب یکی از کارت‌های سری E و اجرای برنامه‌ی WDAQCONF، این برنامه کارت نصب شده را تشخیص می‌دهد. در این حالت یک پنجره‌ی محاوره‌ای باز می‌شود و در آن، شماره‌ی کارت از کاربر سؤال می‌شود. در شکل ۳۴-۱۰ نمونه‌ای از این پنجره را ملاحظه می کنید.




**شکل ۳۴-۱۰**

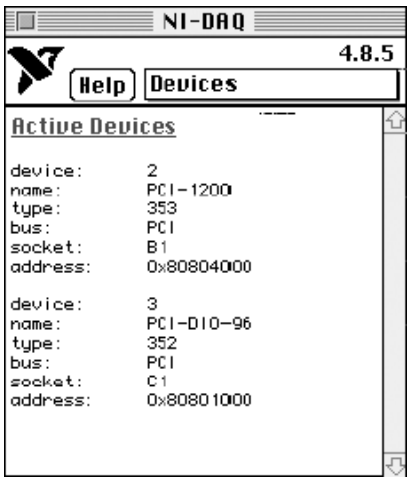
## نرم‌افزار پیکربندی در سیستم عامل Mac

بسته‌ی نرم‌افزاری LabVIEW در مورد برنامه‌های DAQ در محیط Mac از نرم‌افزار NI-DAQ استفاده می کند. این نرم‌افزار تمامی کارت‌های سری Lab، PCI، NB و سیستم SCXI را در محیط Mac پشتیبانی می کند. در شکل ۳۵-۱۰ شمای بلوکی پیکربندی کارت DAQ را در محیط Mac ملاحظه می کنید.



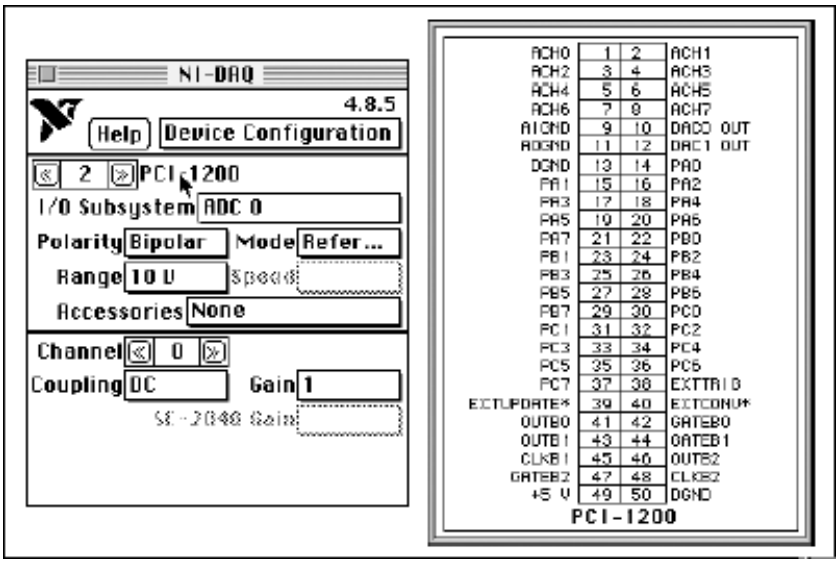

**شکل ۳۵-۱۰: شمای بلوکی پیکربندی کارت DAQ در محیط Mac**

نرم‌افزار NI-DAQ یک برنامه‌ی جانبی است که جهت پیکربندی کارت‌های DAQ مورد استفاده قرار می‌گیرد. همان گونه که در شکل ۳۶-۱۰ ملاحظه می‌کنید به کمک گزینه‌ی Devices در این برنامه می‌توان لیستی از کارت‌های نصب شده در سیستم Mac را مشاهده نمود.



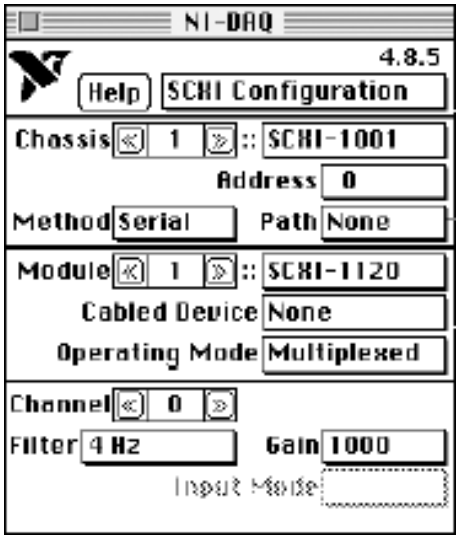

**شکل ۳۶-۱۰: فهرست کارت‌های نصب شده در محیط Mac**

به کمک گزینه‌ی Device Configuration می‌توان هر یک از کارت‌های نصب شده را پیکربندی نمود. پارامترهایی از کارت DAQ که شما می‌توانید به کمک NI-DAQ تنظیم نمایید به کارت بستگی دارد. همان گونه که عنوان شد پیکربندی این پارامترها در برخی از کارت‌ها به صورت نرم‌افزاری انجام می‌شود در حالی که این عمل در کارت‌های دیگر توسط تنظیم jumper ها صورت می‌گیرد. در شکل ۳۷-۱۰ نحوه‌ی پیکربندی کارت PCI-1200 به همراه اتصالات این کارت نشان داده شده است.




**شکل ۳۷-۱۰: نحوه‌ی پیکربندی کارت PCI-1200 به همراه اتصالات این کارت**

همچنین با استفاده از گزینه‌ی SCXI Configuration می‌توان سیستم SCXI را پیکربندی نمود. در شکل ۳۸-۱۰ نحوه‌ی پیکربندی سیستم SCXI را ملاحظه می‌کنید.




**شکل ۳۸-۱۰: نحوه‌ی پیکربندی سیستم SCXI**

پس از پیکربندی پارامترهای مورد نیاز، اجرای مجدد برنامه‌ی NI-DAQ الزامی نیست مگر اینکه قصد داشته باشید پارامترهای کارت را تغییر دهید.

## خلاصه‌ی مطالب در مورد نصب کارت DAQ

۱- اگر تعریف پارامترها درکارت DAQ از طریق تنظیم jumper صورت می‌گیرد، جهت تنظیم آدرس I/O، سطوح وقفه، کانال‌های DMA به دفترچه‌ی راهنمای کارت مراجعه کنید و با کمک آن تنظیمات مذکور را انجام دهید.

۲- رایانه را از پریز برق جدا کنید، سپس کارت را در شکاف متناسب با آن نصب کنید. ممکن است لازم باشد تا کمی کارت را فشار دهید تا در جای خود قرار گیرد. با این حال به کارت زیاد فشار وارد نکنید.

قبل از نصب هر کارت بر روی کارت اصلی رایانه، حتماً رایانه را از پریز برق جدا کنید. عدم انجام این عمل ممکن است منجر به صدمه دیدن کارت DAQ یا کارت اصلی رایانه شود.

۳- رایانه را به پریز برق وصل و آن را روشن کنید. در صورتی که کارت نصب شده از نوع Plug & Play باشد یک پنجره‌ی محاوره‌ای در محیط Win 95 باز می‌شود و شما را از نصب شدن یک سخت‌افزار جدید آگاه می‌سازد.

۴- برنامه‌ی جانبی پیکربندی کارت DAQ (متناظر با سیستم عامل رایانه‌ی خود) را اجرا کنید. بررسی کنید که کارت به طور دقیق نصب شده است.

۵- شماره‌ی اختصاص داده شده به کارت را به خاطر بسپارید. در برنامه های DAQ برای دستیابی به هر یک از کارت ها از شماره‌ی آن استفاده می شود. همان گونه که ذکر شد در محیط Mac شماره‌ی اختصاص داده شده به کارت شماره‌ی شیاری است که کارت در آن قرار گرفته است.

۶- نوع اندازه گیری سیگنال را تعریف کنید. در صورت استفاده از کارتی که نیاز به تنظیم jumper دارد، از هماهنگی تنظیمات مذکور با نوع اندازه گیری سیگنال اطمینان حاصل کنید.

۷- با کلیک کردن بر روی دکمه‌ی Test (از طریق DAQ Devices >> Configure >> System) یا از صفحه‌ی SCXI Devices برنامه‌ی آزمایش پیکربندی کارت را اجرا نمایید. این برنامه جهت ارزیابی صحت پیکربندی کارت به کار می رود. در محیط Mac انجام این آزمایش امکان پذیر نیست.

همچنین با استفاده از دیگر گزینه های موجود در منوی Test می توانید ورودیها و خروجیهای آنالوگ و دیجیتال را آزمایش کنید. به عنوان مثال می توانید ولتاژ 5 V را در یکی از کانال های کارت ایجاد نمایید و با اندازه گیری ولتاژ توسط ولت متر، از صحت نصب و عملکرد کارت اطمینان حاصل کنید.

## کارت GPIB

کارت GPIB جهت کنترل و برقراری ارتباط بین چند ابزار اندازه گیری که دارای باس GPIB هستند مورد استفاده قرار می گیرد. باس GPIB که در دهه‌ی ۱۹۶۰ توسط شرکت HP عرضه شد اکنون به صورت یکی از پرطرفدارترین و پرکاربردترین گذرگاههای ارتباطی استاندارد درآمده است.

تمامی تجهیزات ساخت شرکت HP و بسیاری از دستگاههای دیگر، این سیستم ارتباطی را پشتیبانی می کنند. انجمن مهندسین برق و الکترونیک، باس GPIB را به صورت استاندارد معرفی کرده است. به همین دلیل این باس گاهی با عنوان IEEE 488.2 خوانده می شود.

برخی از قابلیتهای جالب و قابل توجه GPIB عبارتند از:

- داده ها را به صورت موازی (۱ بایت یا ۸ بیت) در هر لحظه انتقال می دهد.
- چند دستگاه (حداکثر ۱۵ عدد) می توانند از یک باس استفاده کنند.
- انتقال داده ها با سرعت بالا صورت می گیرد (۸۰۰ کیلو بایت بر ثانیه یا بیشتر).

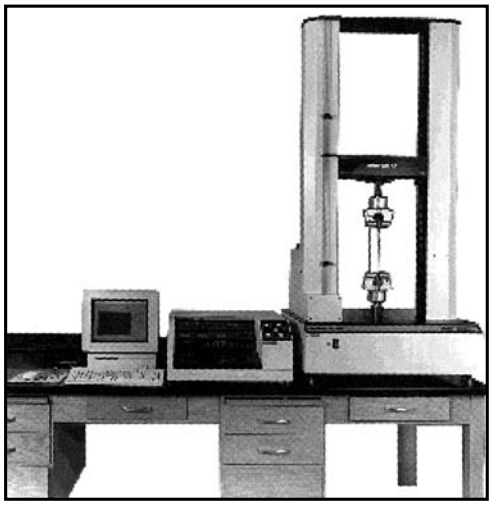
استفاده از کارت GPIB در مورد هر نوع باس، سیستم عامل و همچنین کارت های ارتباطی خارجی که درگاه سریال، درگاه موازی یا درگاه SCSI را به GPIB تبدیل می کنند امکان پذیر است. در شکل ۳۹-۱۰ یک اسیلوسکوپ را ملاحظه می کنید که از طریق کارت GPIB با رایانه مرتبط شده است.

**شکل ۳۹-۱۰: برقراری ارتباط بین اسیلوسکوپ و رایانه از طریق کارت GPIB**

عامل دیگری که سبب شده تا این کارت از طرف همگان مورد استفاده قرار گیرد آن است که به کمک این کارت، رایانه و دستگاه مورد نظر با استفاده از کدهای ساده‌ی ASCII با یکدیگر ارتباط برقرار می کنند. به عنوان مثال در صورت برقراری ارتباط بین یک رایانه‌ی شخصی و یک مولتی متر دیجیتال مدل HP 3458A ممکن است از کدهای زیر استفاده شود:

**PC:** IDN; [Identity-i.e.,who are you]
**HP:** HP 3458A
**PC:** RMEM 1; [Recall Memory Register 1]
**HP:** +4.23789

راه‌اندازهای GPIB با استفاده از تعدادی زیربرنامه با دستگاههای دیگر ارتباط برقرارمی سازند. در این زیربرنامه ها از کدهای ASCII استفاده می شود. بهترین روش برای نصب یک کارت GPIB دنبال نمودن مراحلی است که در دفترچه‌ی راهنمای کارت درج شده است. در شکل ۴۰-۱۰ نحوه‌ی برقراری ارتباط بین رایانه‌ی شخصی و یک ماشین تست فشار و کشش از طریق باس GPIB نشان داده شده است.



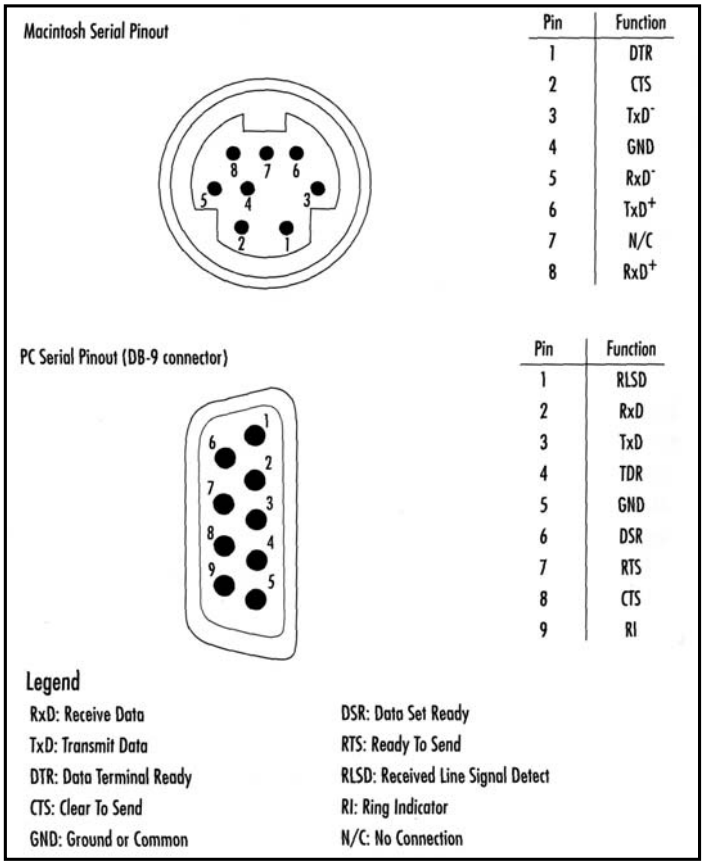
**شکل ۴۰-۱۰: برقراری ارتباط بین یک ماشین تست فشار و رایانه از طریق باس GPIB**

## برقراری ارتباط از طریق درگاه سریال

مزیت استفاده از درگاه سریال، قیمت پایین و در دسترس بودن آن است. به دلیل اینکه در اکثر رایانه‌ها حداقل دو درگاه سریال وجود دارد، در این مورد برای برقراری ارتباط سریال به ندرت به سخت‌افزار دیگری نیاز خواهید داشت. اگرچه در مورد نحوه‌ی اتصال سیم، کانکتورها و استانداردهای معروفی نظیر RS-232 و RS-485 وجود دارند، اما سازندگان دستگاهها اغلب این استانداردها را به طریق نادرست استعمال می‌کنند یا آنها را نادیده می‌گیرند.

در این روش، داشتن اطلاعات در مورد دستگاهی که از طریق درگاه سریال با آن ارتباط برقرار می‌شود، پین‌های خروجی آن و پارامترهای دیگری نظیر سرعت انتقال[۳۰]، بیت توازن[۳۱]، بیت‌های توقف[۳۲] و... حائز اهمیت است. همچنین باید در مورد کاربرد و چگونگی به‌کارگیری خطوط مختلف درگاه سریال، اطلاعات کافی در اختیار داشته باشید.

در شکل ۴۱-۱۰ پین‌های درگاه سریال در رایانه‌ی شخصی و سیستم Mac نشان داده شده است. در بسیاری از حالات، تنها تعداد کمی از این خطوط مورد استفاده قرار می‌گیرند.




**شکل ۴۱-۱۰: پین‌های درگاه سریال در رایانه‌ی شخصی و سیستم Mac**

30- Baud Rate
31- Parity Bit
32- Stop Bits

در ادامه به بررسی برخی از خطوط نشان داده شده در شکل ۴۱-۱۰ می پردازیم:

- Transmit (TxD): داده ها را از رایانه به دستگاه مرتبط با آن ارسال می کند.
- Receive (RxD): داده ها را از دستگاه به رایانه ارسال می کند.
- Ground (GND): نقطه‌ی زمین مرجع.
- Clear To Send (CTS): رایانه از طریق این خط به دستگاه اطلاع می دهد که در چه زمانی آماده‌ی دریافت داده هاست.
- Ready To Send (RTS): رایانه از طریق این خط به دستگاه اطلاع می دهد که در چه زمانی آماده‌ی ارسال داده هاست.

اگر در مورد برقراری ارتباط با دستگاه از طریق درگاه سریال با مشکل مواجه هستید روشهای زیر را برای رفع اشکال به کار برید:

- خطوط TxD و RxD را بررسی کنید. برای انجام این عمل می توانید از یک کابل مخصوص به نام null modem کمک بگیرید.
- هماهنگی سرعت انتقال داده ها، بیت توازن، بیت های توقف، handshaking و پارامترهای دیگر سریال را در دستگاه مورد نظر و رایانه‌ی خود بررسی کنید.
- در اکثر موارد دستگاهی که از طریق درگاه سریال با رایانه مرتبط است دارای یک منبع تغذیه‌ی جداگانه می باشد. بنابراین از روشن بودن آن اطمینان حاصل کنید.
- از صحت اتصال خطوط CTS و RTS اطمینان حاصل نمایید. در مورد برخی از دستگاههای سریال، اتصال این دو خط الزامی است ولی در مورد برخی از دستگاهها اتصال آنها لزومی ندارد.
- مطمئن شوید که درگاه سریال برای کاربرد دیگری استفاده نشده است.
- به دلیل اینکه در هر رایانه حداقل دو درگاه وجود دارد مطمئن شوید که حتماً درگاه مورد نظرتان را استفاده نموده‌اید نه درگاه دیگر را.
- مطمئن شوید که کاراکترهای «EOL» را به طور صحیح ارسال نموده‌اید.

یک روش ساده برای اطمینان از صحت تنظیم پین های درگاه سریال در رایانه آن است که دو رایانه را از طریق درگاه سریال به یکدیگر ارتباط دهید و سپس یک برنامه را بر روی هر دو رایانه اجرا کنید. می توانید با استفاده از صفحه کلید یکی از رایانه ها کاراکترهایی را تایپ کنید و در صورت صحت عملکرد و تنظیم پین های درگاه سریال، کاراکترهای تایپ شده را بر روی صفحه‌ی نمایش رایانه‌ی دیگر ملاحظه کنید (و برعکس).

## خلاصه

۱۰

در این فصل فرضیه‌ی سیگنال ها و روشهای جمع‌آوری داده‌ها مورد بررسی قرار گرفت. جهت اندازه‌گیری سیگنال های مختلف می توان آنها را به دیجیتال on-off، قطار پالس دیجیتال، آنالوگ DC،

آنالوگ AC، و سیگنال های فرکانسی دسته بندی کرد. سیگنال ها به دو صورت متصل شده به زمین یا شناور وجود دارند. سیگنال های متصل شده به زمین معمولاً از دستگاههایی صادر می شوندکه به نحوی به زمین ساختمان یا زمین مرجع متصل شده‌اند. به عنوان مثال برای سیگنال های شناور می توان از ترموکوپل ها و شتاب سنج ها نام برد. سه نوع سیستم اندازه گیری وجود دارد که در موارد مختلف می توان یکی از سیستم ها را بسته به نوع و خواص سیگنال به کار برد. این سیستم های اندازه گیری عبارتند از: سیستم اندازه گیری تفاضلی، سیستم RSE و NRSE.

نرخ نمونه برداری در سیستم DAQ برای سیگنال های آنالوگ حائز اهمیت است. بر اساس فرضیه‌ی نایکوییست، نرخ نمونه برداری باید بیش از دو برابر بزرگترین مؤلفه‌ی فرکانسی سیگنال مورد اندازه گیری باشد.

گام بعدی، انتخاب یک کارت DAQ است. این کارت باید نیازهای ما را در فرآیند DAQ برآورده سازد و انتقال داده ها را به درستی انجام دهد. تعداد کارت های DAQ بسیار زیاد است، بنابراین می توان با توجه به کاربرد و هزینه، کارت مطلوب را انتخاب کرد. در صورتی که تعداد سیگنال ها زیاد باشد می توان از سیستم SCXI استفاده نمود. مراحل نصب و راه‌اندازی یک کارت DAQ نسبت به کارت های قدیمی ساده‌تر شده است. اما در هر حال انجام این عمل نیازمند داشتن اطلاعات کافی در مورد نحوه‌ی تنظیم پارامترهای کارت به کمک نرم‌افزار جانبی NI-DAQ است.

باس GPIB، یک پروتکل استاندارد قابل قبول و فراگیر در بسیاری از ابزارهای اندازه گیری است. برقراری ارتباط از طریق درگاه سریال بسیار کم هزینه و ساده است. اما در عمل ممکن است نیازمند رفع اشکال باشد.

اگر برخی از مطالب این فصل را به خوبی درک نکرده‌اید نگران نباشید. فرآیند DAQ یک مبحث پیچیده است و در اغلب موارد جهت جمع آوری داده ها و برقراری ارتباط صحیح، مهارت و تجربه را در سطح یک مهندس برق و ابزار دقیق می طلبد.

## پاسخ تمرینات

۱-۱۰: ۳،۴،۵،۴،۴،۵،۴،۲،۱،۳

۲-۱۰: ۳ یا ۱،۱،۲.

۳-۱۰: الف) ۱- سیگنال آنالوگ AC (در حوزه‌ی زمان) با دامنه‌ی کوتاه که می توان آن را به عنوان یک سیگنال شناور فرض نمود.

۲- سیستم اندازه گیری تفاضلی (به دلیل دامنه‌ی کم نسبت به زمین)

۳- فرکانس نایکوییست: نمونه برداری باید با نرخ حداقل ۱۰۰۰ نمونه برثانیه انجام گیرد.

$$f_n = \frac{1}{2\ ms} = \frac{1}{2 * 10^{-3} ms} = 500\ Hz$$

۴- تطبیق دادن سیگنال الزامی است. در این حالت تقویت کردن سیگنال (به دلیل دامنه‌ی کوتاه سیگنال ها) جداسازی (به دلیل ایمنی بیشتر) و احتمالاً استفاده از فیلترهای پایین گذر

(به دلیل وجود نویز) لازم است.

۵- در این مورد می توان هر یک از کارت های سری MIO را انتخاب نمود.

ب) ۱- قرائت ترموکوپل و مقدار مقاومت : ورودی آنالوگ DC

ولتاژ تحریک : خروجی آنالوگ DC

روشن و خاموش کردن ماشین توسط رله : خروجی دیجیتال on-off

شمارش تعداد سیکل های انجام شده توسط ماشین : ورودی قطار پالس دیجیتال

۲- در این حالت به دلیل تعداد زیاد کانال ها (۴۸ کانال) به جای استفاده از سیستم اندازه گیری تفاضلی، سیستم اندازه گیری RSE به کار برده می شود.

۳- به دلیل اینکه سیگنال مورد بحث از نوع DC می‌باشد و نیازی به سرعت بالا نیست بنابراین انتخاب فرکانس نمونه برداری ۱۰ Hz بسیار مناسب است.

۴- تطبیق دادن سیگنال ها الزامی است. برای قرائت ترموکوپل ها استفاده از یک مبدل مناسب ضروری است.

۵- در این مورد به دلیل تعداد زیاد کانال ها و نیاز به تطبیق سیگنال ها بهتر است از سیستم SCXI استفاده کنیم.

LabVIEW

# ۱۱

# جمع آوری داده‌ها و کنترل ابزار اندازه گیری

## در این فصل با موارد زیر آشنا می‌شوید:

- برنامه‌های موجود در پالت DAQ
- مراحل لازم جهت اندازه‌گیری و جمع‌آوری داده‌ها در مورد سیگنال‌های دیجیتال و آنالوگ
- نحوه‌ی جمع‌آوری داده‌ها
- برنامه‌های موجود در پالت Instrument I / O
- روش کنترل ابزارهای اندازه‌گیری توسط GPIB

# LabVIEW

# ۱۱ جمع آوری داده‌ها و کنترل ابزار اندازه گیری

در این فصل به بررسی دو مبحث جمع آوری داده ها و کنترل دستگاهها و ابزارهای اندازه گیری در محیط LabVIEW می پردازیم. در ضمن با برخی از برنامه های موجود در پالت های DAQ و Instrument I/O آشنا می شویم و ورودیها و خروجیهای آنالوگ، ورودیها و خروجیهای دیجیتال، نحوه‌ی کنترل ابزار و وسایل اندازه گیری با استفاده از GPIB و ارتباط از طریق درگاه سریال را مورد بحث و بررسی قرار می دهیم. در بخشهای مختلف این فصل به بحث درباره‌ی مراحل لازم جهت جمع آوری داده ها می‌پردازیم و شما را در جهتی که بتوانید برنامه های پیشرفته ای را در مورد DAQ و کنترل دستگاهها ایجاد نمایید سوق خواهیم داد.

در این فصل اهداف زیر دنبال می شوند:

- آشنایی با برنامه های موجود در پالت DAQ.
- فراگیری مراحل لازم جهت اندازه گیری و جمع آوری داده ها در مورد سیگنال های دیجیتال و آنالوگ.
- آشنایی با نحوه‌ی جمع آوری داده ها از طریق ارائه‌ی چند مثال.
- آشنایی با برنامه های موجود در پالت Instrument I/O.
- فراگیری نحوه‌ی کنترل ابزارهای اندازه گیری توسط GPIB.
- مروری بر ارتباط از طریق درگاه سریال.

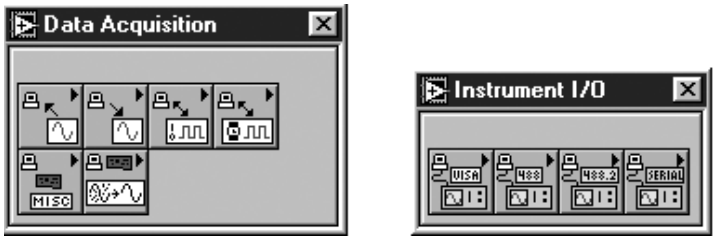
۱۱

اصطلاحات کلیدی موجود در این فصل عبارتند از:

- دستگاه اندازه گیری (Device)
- کانال (Channel)
- بافر (Buffer)
- تریگر (Trigger)
- نمونه (Sample)
- ورودی/خروجی آنالوگ (Analog I/O)
- اسکن (Scan)
- ورودی/خروجی دیجیتال (Digital I/O)

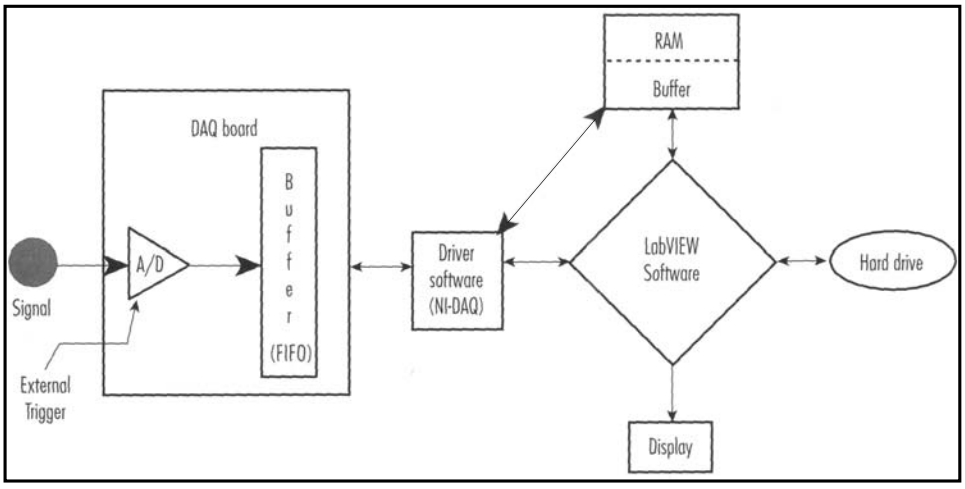
## تعاریف، راه‌اندازها و ابزارها

به دو پالت نشان داده شده در شکل ۱-۱۱ توجه کنید. برنامه های موجود در این دو زیرپالت در سطحی فراتر از دیگر برنامه های نرم‌افزار LabVIEW هستند. این برنامه ها به کاربر اجازه می دهند تا از طریق کارت های DAQ و GPIB با دنیای خارج از رایانه ارتباط برقرار نماید و به کمک آنها داده ها را بخواند یا بر روی آنها فرمانهایی را ارسال کند. برنامه های موجود در این دو زیرپالت همچنین امکان اندازه گیری و روشن و خاموش کردن وسایل را از طریق کارت های مذکور فراهم می سازند.




**شکل ۱-۱۱: زیرپالت های Instrument I/O و Data Acquisition**

در این فصل قصد داریم تا توابع و دستورهای آنالوگ، دیجیتال، نحوه‌ی برقراری ارتباط از طریق کارت های GPIB و درگاههای سریال را مورد بررسی قرار دهیم. در شکل ۲-۱۱ نحوه‌ی برقراری ارتباط بین LabVIEW و کارت های DAQ را ملاحظه می کنید.




**شکل ۲-۱۱: نحوه‌ی برقراری ارتباط بین LabVIEW و کارت های DAQ**

۱۱

همان گونه که در شکل ۲-۱۱ ملاحظه می کنید آغاز فرآیند DAQ مستلزم فراخوانی برنامه‌ی NI-DAQ توسط LabVIEW است. این برنامه‌ی جانبی، نرم‌افزار راه‌انداز کارت را نیز در بر دارد. فراخوانی برنامه‌ی NI-DAQ به نوبه‌ی خود به سخت‌افزار فرمان می دهد تا I/O را بررسی کند. در کارت های DAQ جهت ذخیره‌ی داده های جمع آوری شده در یک مکان واسطه و موقت، از بافر استفاده می شود. توجه داشته باشید که نرم‌افزار، تنها عامل آغازکننده‌ی عملیات I/O نیست. بلکه یک قطعه‌ی سخت‌افزاری خارجی نیز می تواند عملیات مذکور را تریگر کند. دو سؤال مهم و اساسی زیر، شما را در طبقه‌بندی نوع فرآیند DAQ یاری می کند:

- آیا از بافر استفاده می کنید؟
- آیا برای شروع، توقف و یا همزمان ساختن فرآیند DAQ از تریگرکننده‌ی خارجی استفاده می شود؟

در ادامه به بررسی دو مفهوم بافر و تریگر می پردازیم.

## بافر

بافر به مفهومی که در این کتاب از آن استفاده شده است، ناحیه‌ای از حافظه در رایانه است که جهت ذخیره‌ی موقت داده‌ها قبل از ارسال به محلی دیگر مورد استفاده قرار می گیرد. به عنوان مثال فرض کنید که قصد دارید تا هزاران داده را در عرض یک ثانیه نمونه‌برداری کنید. به نمایش در آوردن تمامی این داده‌ها در عرض این مدت کوتاه بسیار مشکل و حتی غیر ممکن است. اما با فرمان دادن به کارت DAQ مبنی بر جمع آوری داده‌ها در یک بافر به راحتی می توانید ابتدا داده‌ها را در بافر ذخیره نمایید، سپس برای تجزیه و تحلیل یا نمایش داده‌ها، آنها را از بافر فراخوانی کنید. به خاطر داشته باشید که استفاده از بافرها به سرعت و حجم عملیات DAQ بستگی دارد. اگر کارت مورد استفاده‌ی شما قابلیت DMA داشته باشد عملیات ورودی آنالوگ یک مسیر سخت‌افزاری بسیار کوتاه و سریع در رسیدن به RAM رایانه در پیش دارد. با استفاده از قابلیت DMA داده های جمع آوری شده مستقیماً به داخل حافظه راه می یابند.

در صورت عدم استفاده از بافر باید اعمالی از قبیل ترسیم بر روی گراف، ذخیره‌ی داده ها بر روی دیسک، تجزیه و تحلیل داده ها و... را در هر لحظه تنها بر روی یک نمونه انجام دهید و به دلیل اینکه هیچ محلی جهت نگهداری و ذخیره‌ی موقت نمونه های دیگر در اختیار ندارید، این نمونه همان داده‌ای است که اخیراً جمع آوری شده است.

در موارد زیر از I/O همراه بافر استفاده کنید:

- در حالتی که قصد دارید تا نمونه هایی را تولید یا جمع آوری نمایید و سرعت جمع آوری یا تولید داده ها بیش از سرعت و قابلیت نمایش داده ها بر روی گراف، ذخیره‌ی داده ها بر روی دیسک یا تجزیه و تحلیل آنها در زمان فعلی باشد.
- در مواردی که قصد دارید داده های AC را با سرعتی بیش از ۱۰ نمونه در ثانیه جمع آوری نمایید و به تجزیه و تحلیل یا نمایش داده ها بپردازید.

- در حالتی که لازم است در طول مراحل مختلف فرآیند DAQ، نمونه‌گیری با پریود بسیار دقیق و به صورت یکنواخت انجام گیرد.

در موارد زیر می‌توانید از I/O بدون بافر استفاده کنید:

- در حالتی که مجموعه‌ی داده‌ها بسیار کوچک است (به عنوان مثال جمع‌آوری داده‌ها با سرعت ۱ نمونه در هر ثانیه از ۲ کانال).
- در مواردی که قصد داشته باشید حجم سطوح بالایی حافظه را کاهش دهید. زیرا بافر بخشی از فضای حافظه را اشغال می‌کند.

در بخشهای بعد در مورد بافرها به تفصیل به بحث خواهیم نشست.

## تریگر کردن

تریگر کردن به روشی اطلاق می‌گردد که به کمک آن، فرآیند DAQ را آغاز نموده، متوقف می‌کنید یا آن را با عملیات یا اعمال دیگر همزمان می‌سازید. سیگنال تریگرکننده معمولاً یک سیگنال آنالوگ یا دیجیتال است که ابتدا شرایط آن بررسی می‌شود و سپس بر اساس آن، زمان انجام شدن تمام یا بخشی از عملیات مورد نظر تعیین می‌شود.

در مورد تریگر کردن یک عملیات، دو روش نرم‌افزاری و سخت‌افزاری وجود دارد. متداولترین و ساده‌ترین روش، تریگر نرم‌افزاری است. در این روش برای شروع یا متوقف نمودن فرآیند DAQ، مستقیماً از نرم‌افزار (نظیر یک المان جبری که در صفحه‌ی پانل قرار گرفته است) استفاده می‌شود. در تریگر سخت‌افزاری، مدارات موجود در کارت، جمع‌آوری داده‌ها را همراه با زمان‌بندی دقیق فرآیند DAQ کنترل می‌کنند.

تریگر سخت‌افزاری را می‌توان به دو دسته‌ی تریگر داخلی و خارجی تقسیم نمود. یک مثال از تریگر داخلی، برنامه‌ریزی کارت بدین صورت است که در هنگامی که یک کانال ورودی آنالوگ به سطح ولتاژ معینی می‌رسد، در خروجی، یک پالس دیجیتال ایجاد می‌کند. نمونه‌ای از تریگر خارجی، انتظار کارت برای دریافت یک پالس از یکی از ابزارهای اندازه‌گیری خارجی جهت آغاز فرآیند DAQ است.

در تمامی محصولات شرکت NI، یک پین تریگر خارجی تعبیه شده است. این پین یک پالس دیجیتال را دریافت می‌کند و تریگر را به صورت خارجی انجام می‌دهد. بسیاری از ابزارهای اندازه‌گیری نیز یک خروجی دیجیتال دارند که اغلب با نام Trigger Out خوانده می‌شود. این خروجی جهت تریگر کردن دستگاهها یا ابزارهای دیگر و در اینجا برای تریگر کردن کارت DAQ در نظر گرفته شده است.

در موارد زیر از تریگر نرم‌افزاری استفاده کنید:

- در حالتی که کاربر قصد دارد کنترل همزمان بر روی عملکرد تمامی فرآیندهای DAQ داشته باشد.
- در مواردی که لازم نیست تا زمان‌بندی در فرآیند DAQ بسیار دقیق باشد. نظیر حالتی که عملیات ورودی آنالوگ آغاز می‌گردد.

۱۱

در موارد زیر از تریگر سخت‌افزاری استفاده کنید:

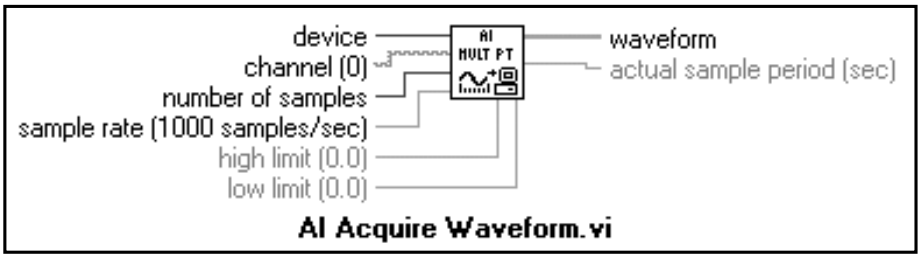
- در حالتی که لازم است زمان‌بندی در فرآیند DAQ بسیار دقیق باشد.
- در مواردی که قصد داشته باشید سربار[1] سخت‌افزاری را کاهش دهید. به عنوان مثال می‌توانید یک حلقه‌ی While که بر وقوع و انجام فرآیند DAQ نظارت دارد را از میان بردارید.
- در حالتی که لازم است تا فرآیند DAQ توسط یک سیگنال خارجی تریگر شود. در این حالت باید تریگر کردن به صورت سخت‌افزاری و از نوع تریگر خارجی انجام گیرد.

در بخش بعد خواهید دید که چگونه می‌توان از برنامه‌های مخصوص DAQ جهت پیکربندی عملکرد I/O استفاده نمود و همچنین نوع تریگر را تعیین کرد.

## ورودی/خروجی آنالوگ

در این بخش فرض بر این است که کاربر می‌تواند به راحتی یک کارت DAQ را نصب نماید و آن را پیکربندی کند. در صورت نیاز به راهنمایی در انجام اعمال مذکور می‌توانید مطالب ارائه شده در فصل ۱۰ را به دقت مطالعه نمایید یا به دفترچه‌ی راهنمای کارت DAQ مراجعه کنید. در تمرینات نیز فرض بر این است که منابع سیگنال مختلف و همچنین یک سیستم کاملاً پیشرفته‌ی LabVIEW در اختیار دارید.

قبل از ورود به بحث DAQ و برنامه‌های مربوط به آن به معرفی چند اصطلاح می‌پردازیم. واژه‌ها و اصطلاحات تعریف شده در این بخش دائماً در ورودی یا خروجی برنامه‌ها دیده می‌شوند. بنابراین در صورتی که قصد دارید به کمک این برنامه‌ها به جمع‌آوری داده‌ها بپردازید درک مفهوم آنها الزامی است. در شکل ۳-۱۱ یکی از برنامه‌های مخصوص DAQ به همراه ورودیها و خروجیهای آن نشان داده شده است.




**شکل ۳-۱۱: آیکن برنامه‌ی AI Acquire Waveform.vi به همراه اتصالات آن**

device یا device number: شماره‌ی کارت. شماره‌ای است که توسط برنامه‌ی جانبی NI-DAQ به کارت DAQ اختصاص داده می‌شود. این عدد همان شماره‌ی شیاری از کارت اصلی در محیط Mac است که کارت DAQ در آن نصب می‌شود. این پارامتر به LabVIEW اطلاع می‌دهد که فرآیند DAQ توسط کدام یک از کارت‌ها انجام می‌شود. به کمک این پارامتر، برنامه‌های مخصوص فرآیند DAQ مستقل از نوع کارت عمل می‌کنند. بدین مفهوم که اگر در آینده یکی از کارت‌های سازگار را با کارت فعلی جایگزین کنید و همان شماره را به آن اختصاص دهید، تمامی برنامه‌های تعریف شده با شماره‌ی مذکور، بدون نیاز به اصلاح و تغییر به راحتی با

1- Overhead

کارت جدید هماهنگی دارند و فرآیند DAQ را توسط آن انجام می دهند.

sample: نمونه. بیانگر یک تبدیل A/D است. نمونه تنها یک نقطه یا به عبارت دیگر یک مقدار عددی متناظر با مقدار آنالوگ حقیقی در زمانی است که اندازه گیری انجام شده است.

channel: کانال. بیانگر آن است که داده ها از طریق چه منبعی به داخل کارت راه می یابند. به عبارت دیگر شماره‌ای است که به منبع فیزیکی و منشأ ورود نمونه ها اختصاص داده شده است. به عنوان مثال توسط یک کارت با ۱۶ کانال ورودی آنالوگ، می توانید در هر لحظه ۱۶ دسته داده را همزمان جمع آوری کنید.

در برنامه های DAQ، یک کانال یا دسته‌ای از کانال ها به روشی بسیار ساده و ابتدایی به صورت رشته مشخص می شوند. دلیل استفاده از فرم رشته‌ای به جای نوع عددی جهت نمایش کانال ها، به‌کارگیری دو کاراکتر غیر عددی «:» و «,» است. استفاده از این دو کاراکتر جهت نشان دادن دسته‌ای از کانال ها لازم و ضروری است. در جدول ۱-۱۱ نمونه‌ای از روش به کارگیری دو کاراکتر مذکور را ملاحظه می کنید.

**جدول ۱-۱۱: نمونه‌ای از روش به‌کارگیری دو کاراکتر غیرعددی «:» و «,»**

| Channels | Channel String |
|---|---|
| Channel 5 | 5 |
| Channels 0 thru 4 | 0:4 |
| Channels 1, 8, and 10 thru 13 | 1, 8, 10:13 |

در برخی موارد در برنامه های سطح پایین DAQ، از آرایه های رشته‌ای برای نشان دادن دسته‌ای از کانال ها استفاده می شود. در این آرایه ها هر مؤلفه مشخص کننده‌ی یک کانال است.

scan: اسکن. قرائت تمام کانال هایی است که در فرآیند DAQ مورد استفاده قرار گرفته‌اند.

waveform: شکل موج. مجموعه‌ای از نمونه های یک کانال است که در فاصله‌ی زمانی معینی جمع آوری شده است. ترتیب جمع آوری نمونه ها نیز بر حسب زمان اندازه گیری آنهاست. معمولاً ولی نه همواره، پریود نمونه برداری در یک شکل موج ثابت است.

در اغلب موارد مبتدیان در تشخیص دو مفهوم اسکن و شکل موج دچار مشکل می‌شوند. اسکن مجموعه‌ای از نمونه‌هاست که در هر لحظه تنها یک نمونه از هر کانال جمع‌آوری می‌شود. در حالی که شکل موج، مجموعه‌ای از نمونه های یک کانال در یک فاصله‌ی زمانی معین است.

high limit و low limit: حداقل و حداکثر مقادیر ولتاژی هستند که در مورد سیگنال خود انتظار دارید. این مقادیر به صورت پیش فرض در حد 10 V+ و 10 V- در نظر گرفته می شوند. با تغییر این مقادیر می توان بهره‌ی[2] سیستم DAQ را تغییر داد. به عنوان مثال اگر این مقادیر را در حد 5 V+ و 5 V- تنظیم کنید بهره‌ی کارت استفاده شده، 2 خواهد بود و در صورتی که آنها را به صورت 1 V+ و 1 V- تنظیم نمایید به بهره‌ی 10 دست خواهید یافت. برای محاسبه‌ی بهره‌ی کارت از رابطه‌ی زیر استفاده کنید:

(|حد پایین| - |حد بالا|)/مقدار ولتاژ ورودی کارت = بهره

در هنگام تعیین حدود ورودی کارت و محاسبه‌ی بهره دقت داشته باشید، زیرا بهره‌ی بعضی از کارت‌های DAQ

2- Gain

۱۱

به صورت پیش‌فرض تنظیم شده است. اگر مقدار بهره را به‌گونه‌ای تنظیم کنید که خارج از حدود از پیش تعیین شده باشد، LabVIEW به صورت خودکار این مقدار را به نزدیکترین بهره‌ی پیش‌فرض گرد می‌کند. بهره‌های قابل قبول برای یک کارت نوعی عبارتند از: 100, 50, 20, 10, 5, 2, 1, 0.5 .

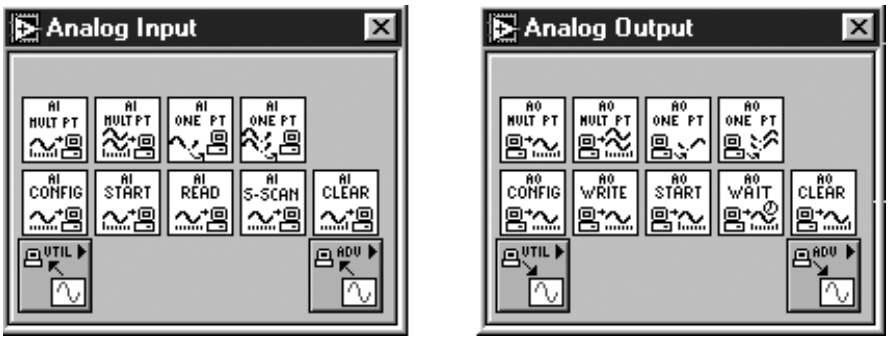
taskID : یک عدد صحیح ۳۲ بیتی (از نوع عددی I32) است که توسط برخی از برنامه‌های DAQ جهت تعریف و شناسایی عملکرد I/O استفاده می‌شود. بسیاری از برنامه‌ها برای ارتباط با برنامه‌های قبلی و بعدی به taskID in و taskID out نیاز دارند. با سیم‌کشی به این ترمینال‌ها تعریف نمودن اطلاعاتی در مورد کارت، نرخ نمونه‌برداری، حدود بالا و پایین و... در تمامی برنامه‌های مرتبط الزامی نیست؛ تنها کافی است این اطلاعات را در اولین برنامه تعریف نمایید. سپس این برنامه تمامی مقادیر را در ورودی tasKID in دریافت می‌کند و در خروجی taskID out نیز همین مقادیر را به برنامه‌ی بعدی انتقال می‌دهد. به همین ترتیب هر یک از برنامه‌ها این اطلاعات را از برنامه‌ی قبلی دریافت می‌کند، سپس آنها را در اختیار برنامه‌ی بعدی قرار می‌دهد.

در صورتی که در مورد درک این تعاریف دچار مشکل شده‌اید نگران نباشید. پس از بررسی چند مثال، این مفاهیم را به راحتی فرا خواهید گرفت.

## برنامه‌های ساده در مورد ورودی/خروجی آنالوگ

اگر امکانات لازم برای آزمایش نمودن کارت DAQ در اختیار ندارید، برای شروع می‌توانید برخی از برنامه‌های ساده را مورد استفاده قرار دهید. حتی اگر در حال جمع‌آوری داده‌های پیچیده هستید، بهتر است که به منظور درک ماهیت این برنامه‌ها، ابتدا برنامه‌های ساده‌ی زیر را استفاده کنید.

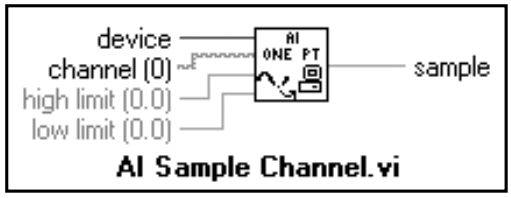
تمامی برنامه‌های موجود در ردیف بالایی در پالت‌های Analog Input و Analog Output، برنامه‌های ساده هستند. در ادامه به بررسی این برنامه‌ها می‌پردازیم.




**شکل ۴-۱۱: زیرپالت‌های Analog Input و Analog Output**

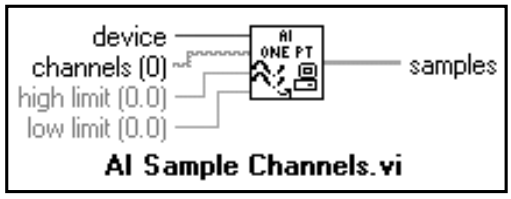
## ورودی آنالوگ

برنامه‌ی AI Sample Channel.vi از کانال مشخص شده در ورودی رشته‌ای channel یک نمونه دریافت می‌کند و سپس آن را در خروجی sample ظاهر می‌سازد.



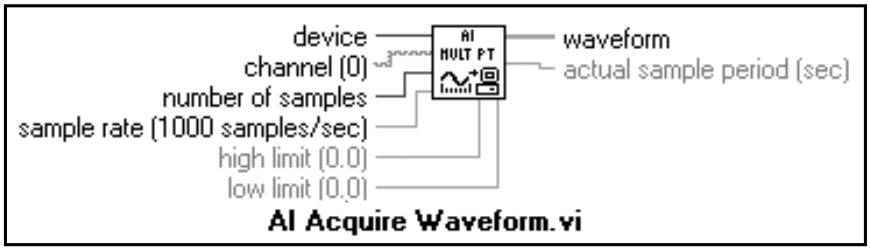

شکل ۵-۱۱

برنامه‌ی AI Sample Channels.vi همانند برنامه‌ی قبلی عمل می کند. با این تفاوت که این برنامه از هر یک از کانال های مشخص شده در ورودی رشته ای channels یک نمونه دریافت می کند و سپس آنها را به صورت یک آرایه در خروجی samples ظاهر می سازد.




شکل ۶-۱۱

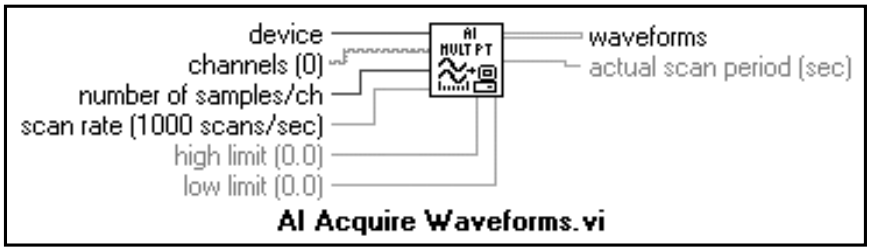
برنامه‌ی AI Acquire Waveform.vi یک شکل موج یا به عبارت دیگر نمونه های به دست آمده از یک کانال در یک بازه‌ی زمانی معین را جمع آوری می کند. این نمونه ها به صورت یک آرایه در خروجی waveform نشان داده می شوند. مؤلفه های این آرایه، نمونه های جمع آوری شده از یک کانال هستند که با پریود خاصی نمونه برداری شده اند.




شکل ۷-۱۱

برنامه‌ی AI Acquire Waveforms.vi نیز همانند برنامه‌ی قبلی عمل می کند، با این تفاوت که در این برنامه از هر یک از کانال های تعیین شده در ورودی رشته ای channels، یک شکل موج به دست می آید. نمونه های جمع آوری شده به صورت یک آرایه‌ی دوبعدی در خروجی waveforms نشان داده می شوند. ترتیب قرارگرفتن نمونه ها برحسب شماره‌ی کانال و زمان نمونه برداری است. داده‌های هر یک از کانال ها در یک ستون ذخیره می شود. ترتیب قرارگرفتن نمونه ها در سطرها بر حسب زمان نمونه برداری آنهاست، به صورتی که نمونه های اولیه در سطرهای بالا و نمونه های انتهایی در سطرهای پایین تر قرار می گیرند.

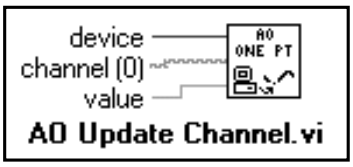
۱۱




شکل ۸-۱۱

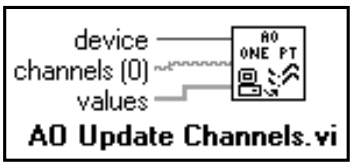
## خروجی آنالوگ

برنامه‌ی AO Update Channel.vi مقدار تعیین شده در ترمینال ورودی value را در خروجی کانال مشخص شده در ترمینال ورودی channel فراهم می سازد. این مقدار تا زمانی که توسط کاربر تغییر داده نشده یا کارت یا شماره‌ی آن تغییر نکرده باشد ثابت باقی می ماند.



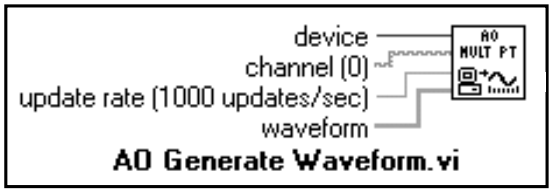

شکل ۹-۱۱

برنامه‌ی AO Update Channels.vi همانند برنامه‌ی قبلی عمل می کند. با این تفاوت که این برنامه مقادیر تعیین شده در آرایه‌ی ورودی values را در خروجی کانال های مشخص شده در ورودی رشته‌ای channels فراهم می سازد. این مقادیر تا زمانی که توسط کاربر تغییر داده نشده یا کارت یا شماره‌ی آن تغییر نکرده باشد ثابت باقی می مانند.



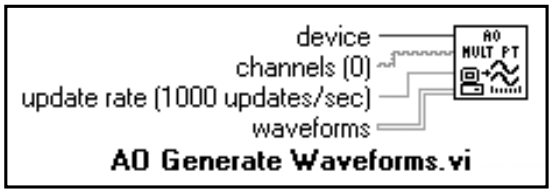

شکل ۱۰-۱۱

برنامه‌ی AO Generate Waveform.vi در خروجی کانال تعیین شده در ورودی رشته‌ای channel یک شکل موج تولید می کند. ورودی waveform حاوی یک آرایه‌ی یک بعدی است. ورودی update rate سرعت به روز رسانی داده ها یا به عبارت دیگر زمان بین نقاط موجود در شکل موج را تعیین می کند.




شکل ۱۱-۱۱

برنامه‌ی AO Generate Waveforms.vi همانند برنامه‌ی قبلی عمل می‌کند. با این تفاوت که این برنامه در خروجی کانال‌های تعیین شده در ورودی رشته‌ای channels، شکل موجهایی تولید می‌کند. این شکل موجها به صورت یک آرایه‌ی دوبعدی در ورودی waveforms ظاهر می‌شوند. هر یک از شکل موجها در یکی از ستونهای آرایه‌ی مذکور ذخیره می‌شوند.



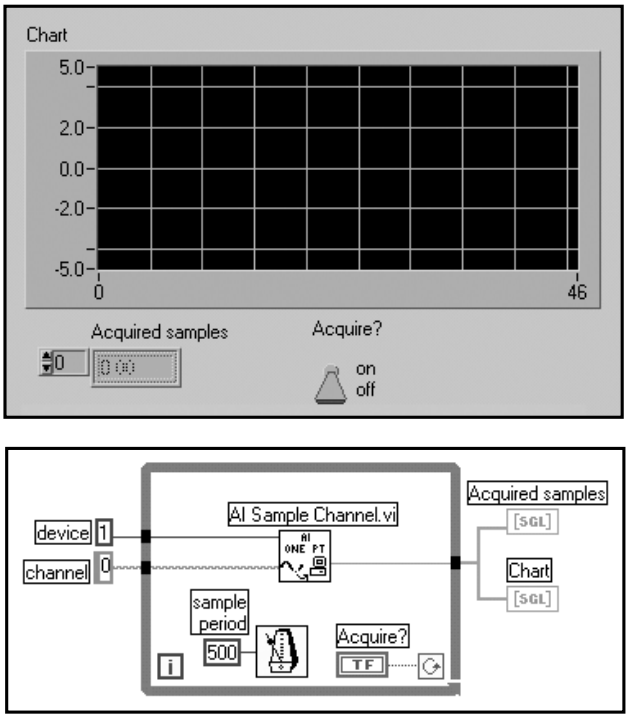

**شکل ۱۲-۱۱**

برای درک عملکرد و روش استفاده از این برنامه‌ها تمرین ساده‌ی زیر را انجام دهید.

## تمرین ۱-۱۱: سربار نرم‌افزاری

۱- یک منبع ولتاژ نظیر دستگاه مولد موج[3] را به کانال 0 در کارت DAQ متصل کنید. در این مرحله می‌توانید حتی از ولتاژ ۱/۵ ولتی حاصل از یک باطری نیز استفاده کنید.

۲- صفحات پانل و نمودار بلوکی این برنامه را مطابق شکل ۱۳-۱۱ ایجاد کنید.




**شکل ۱۳-۱۱: صفحات پانل و نمودار بلوکی در برنامه‌ی Quick Analog In.vi**

3- Function Generator

۱۱

برنامه‌ی AI Sample Channel.vi را از زیرپالت Functions >> Data Acquisition >> Analog Input انتخاب کنید. اگر در نرم‌افزار جانبی NI-DAQ شماره‌ی کارت استفاده شده با عدد 1 مشخص نشده است، شماره‌ی کارت را به گونه‌ای تغییر دهیدکه با سیستم هماهنگی داشته باشد.

۳- این برنامه را با عنوان Quick Analog In.vi در زیرفهرست LabVIEW\My Activity ذخیره کنید.

۴- با تغییر وضعیت کلید جبری ?Acquire به حالت on، برنامه را اجرا کنید و پس از چند ثانیه با خاموش کردن کلید مذکور برنامه را متوقف سازید.

۵- آرایه‌های حاصل در Acquired samples را ملاحظه کنید. مؤلفه‌های آرایه را یک به یک بررسی نمایید.

احتمالاً استنباط کرده‌اید که حلقه‌ی While برای کنترل عملیات نمونه‌برداری در صفحه‌ی نمودار بلوکی این برنامه گنجانده شده است. با این عمل در حقیقت سربار نرم‌افزاری غیر ضروری را به صفحات پانل و نمودار بلوکی افزوده‌اید. به‌کارگیری این حلقه در این مورد و برای مروری گذرا بر داده‌ها مسأله‌ای نیست. اما در یک صفحه‌ی نمودار بلوکی که حاوی چندین زیربرنامه است نیازی نیست تا برای اعمالی نظیر کنترل عملیات نمونه‌برداری، از نرم‌افزار استفاده نمایید. برای انجام اعمالی این چنین، کارت‌های DAQ و راه‌اندازهای خاصی طراحی شده‌اند. این روش جمع‌آوری داده‌ها، تحت شرایط زیر به خوبی عمل می‌کند و نیازهای شما را برآورده می‌سازد:

- نرخ نمونه‌برداری پایین باشد (۱ نمونه در هر ثانیه یا کمتر).
- دیگر عواملی که با سیستم عامل در ارتباط هستند و به زمان بستگی دارند، در زمانی که این برنامه در حال اجراست روی ندهند.
- تغییر سرعت نمونه‌برداری تدریجی و در حد قابل قبول باشد.

به عنوان مثال فرض کنید که با استفاده از توابع زمان‌بندی در LabVIEW در حال اجرای یک برنامه هستید که در این برنامه (نظیر مثال قبل) از یک حلقه جهت انجام نمونه‌برداری با سرعت ۱ نمونه درثانیه استفاده شده است. حال تصمیم دارید در حین اجرای برنامه، صفحه‌ی پانل را کمی به اطراف حرکت دهید. در هنگامی که شما در حال حرکت دادن صفحه‌ی برنامه هستید عملیات DAQ انجام نمی‌گیرد و ممکن است در این فاصله، شکستگی یا ناپیوستگی در نمودار ایجاد گردد.

فرض کنید که عمل انتقال و حرکت صفحه در عرض ۵ ثانیه انجام گیرد. پس از رها کردن دکمه‌ی ماوس، برنامه به نمونه‌برداری و جمع‌آوری داده‌ها ادامه می‌دهد. اما هیچ نشانه‌ای مبنی بر عدم انجام نمونه‌برداری در مدت مذکور وجود ندارد. بنابراین هنگامی که از برنامه‌هایی استفاده می‌کنید که جهت زمان‌بندی خود از حلقه استفاده می‌کنند، باید مطمئن شوید که برنامه‌های دیگری به صورت همزمان اجرا نشوند. زیرا اجرای همزمان این برنامه‌ها باعث ایجاد تداخل زمانی در فرآیند نمونه‌برداری می‌گردد. بنابراین از انجام اعمالی نظیر حرکت دادن ماوس که باعث بروز تداخل زمانی در فرآیند DAQ می‌گردد خودداری نمایید.

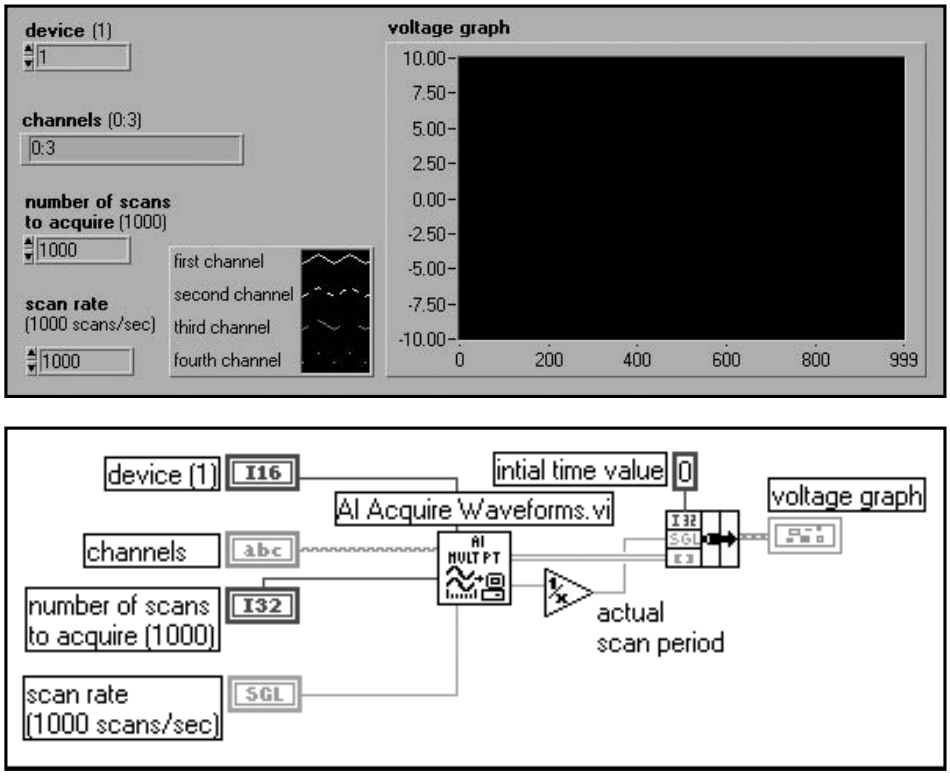
برای نظارت و کنترل بیشتر می‌توانید از توابع دیگری نظیر Tick Count (ms) جهت مشخص کردن دقت زمان‌بندی در حلقه استفاده کنید.

حال به بررسی مثال دیگری می‌پردازیم. در این برنامه ابتدا داده‌های چند کانال، جمع‌آوری شده و در مرحله‌ی بعد، همه‌ی آنها بر روی یک گراف به نمایش در می‌آیند.

## تمرین ۲-۱۱: نمونه‌برداری از چند کانال

۱- ۴ منبع ولتاژ DC یا ۴ منبع با فرکانس پایین را به کانال‌های 3-0 بر روی کارت DAQ متصل کنید. اگر دسترسی به ۴ منبع سیگنال متفاوت امکان‌پذیر نیست می‌توانید یک منبع را به ۴ کانال ارتباط دهید یا از یک شبکه‌ی مقاومتی استفاده نمایید و با تقسیم ولتاژ نسبت به مقاومتهای مختلف، ولتاژهای متفاوتی را به کانال‌های مذکور اعمال کنید.

۲- صفحات پانل و نمودار بلوکی این برنامه را مطابق شکل ۱۴-۱۱ ایجاد کنید.




**شکل ۱۴-۱۱: صفحات پانل و نمودار بلوکی در برنامه‌ی Acquire Multiple Channels.vi**

منوی کرکره‌ای را بر روی گراف باز کنید و با انتخاب گزینه‌ی Transpose Array، ترانهاده‌ی آرایه‌ی ورودی را بر روی گراف ترسیم نمایید.

برنامه‌ی AI Acquire Waveforms.vi یک آرایه‌ی دوبعدی به دست می‌دهد که هر ستون آن حاوی ولتاژهای نمونه‌برداری شده از یک کانال می‌باشد. همان گونه که می‌دانید آرایه‌ی ورودی هر گراف معمولاً به صورت سطری ترسیم می‌شود. بنابراین برای نمایش مقادیر ولتاژ بر روی محور Y لازم است تا ابتدا آرایه‌ی ورودی را ترانهاده کنید.

۳- مقادیر ورودیهای scan rate، channels و number of scans to acquire را طبق اعداد نشان داده شده در شکل ۱۴-۱۱ تنظیم نمایید، سپس برنامه را اجرا کنید.

۴- این برنامه را با عنوان Acquire Multiple Channels.vi ذخیره کنید.

صفحه‌ی نمودار بلوکی این برنامه بسیار ساده است. برنامه‌ی AI Acquire Waveforms.vi تمامی اعمال را انجام می‌دهد. به کاربرد دستور Bundle در این برنامه توجه کنید. از این دستور برای دسته‌بندی داده‌ها به منظور به نمایش در آوردن تمامی آنها بر روی یک گراف استفاده شده است. خروجی actual scan rate از برنامه‌ی AI Acquire Waveforms.vi برای ترسیم نقاط با مقیاس زمانی دقیق به کار برده شده است. توجه داشته باشید که این برنامه از یک مبدل ADC بدون بافرکه به طریقه‌ی نرم‌افزاری تریگر شده است استفاده می کند.

اگر قصد ندارید تا با کمک نرم‌افزار LabVIEW به جمع‌آوری داده‌ها بپردازید و یا امکانات لازم نظیر کارت‌های DAQ و ... را در اختیار ندارید، از مطالعه‌ی سایر بخشهای این فصل چشم‌پوشی کنید.

## برنامه‌های ورودی/خروجی آنالوگ سطح متوسط

برنامه‌های ورودی/خروجی آنالوگ سطح متوسط نسبت به برنامه‌های ساده‌ای که در بخش قبل مورد بررسی قرار گرفتند از قابلیت انعطاف بیشتری برخوردارند. بنابراین در برخی موارد بهتر است از برنامه‌های سطح متوسط استفاده نماییم. به عنوان مثال با هر بار فراخوانی برنامه‌ی AI Sample Channel.vi که در ردیف برنامه‌های ساده قرار دارد لازم است تا پارامترهایی نظیر نرخ نمونه‌برداری را جهت انواع روشهای اندازه‌گیری تنظیم کنید. در نتیجه اگر تعداد نمونه‌های جمع‌آوری شده بسیار زیاد باشد لازم است تا پارامترهای مذکور را در هر مرحله تنظیم نمایید. برای رفع این مشکل می‌توان از برنامه‌های سطح متوسط استفاده نمود. این برنامه‌ها در ردیف میانی زیرپالت Functions >> Data Acquisition >> Analog Input قرار دارند. برنامه‌های مذکور، روند برنامه‌نویسی را بسیار ساده می‌کنند و سرعت جمع‌آوری و نمونه‌گیری داده‌ها را افزایش می‌دهند. در این برنامه‌ها می‌توانید سربارهای نرم‌افزاری غیر ضروری را ایجاد کنید.

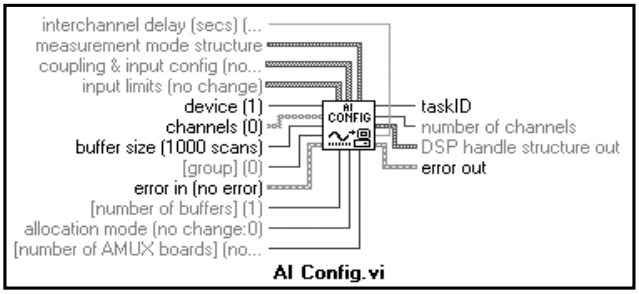
برنامه‌های سطح متوسط، قابلیت انعطاف و کارایی بیشتری جهت ارتقای برنامه‌های کاربردی از خود نشان می‌دهند. این برنامه‌ها قابلیتهایی نظیر قدرت کنترل نرخ نمونه‌برداری مقادیر میانی، استفاده از تریگر خارجی، انجام عملیات مداوم I/O را دارا هستند.

در بخش بعد به بررسی برنامه‌های سطح متوسط می‌پردازیم. توجه داشته باشید که در برخی از این برنامه‌ها تعداد ورودیها و خروجیها بسیار زیاد است. در این مورد می‌توانید تنها ترمینال‌های مورد نیاز را سیم‌کشی کنید. در اکثر موارد سیم‌کشی اتصالات اختیاری الزامی نیست. همان گونه که قبلا عنوان شد این اتصالات در پنجره‌ی Detialed Help به رنگ خاکستری ظاهر می‌شوند.

## ورودی آنالوگ

۱۱

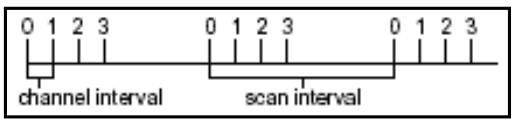
برنامه‌ی AI Config.vi عملیات ورودی آنالوگ را در مورد مجموعه‌ای از کانال‌ها پیکربندی می‌کند و برای انجام این عملیات یک بافر در حافظه‌ی رایانه اختصاص می‌دهد.




شکل ۱۵-۱۱

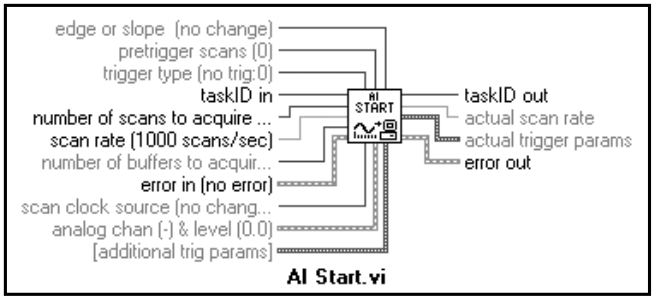
شماره‌ی اختصاص داده شده به کارت DAQ در ورودی device تعیین می‌گردد. شماره‌ی کانال‌های ورودی آنالوگ به صورت آرایه‌ی رشته‌ای در ورودی channels مشخص می‌شوند. ورودی input limits محدوده‌ی سیگنال ورودی را تعیین می‌کند و مسلماً بر مقدار بهره‌ی کارت DAQ تأثیرگذار است. مقدار buffer size با توجه به تعداد اسکن‌ها مشخص می‌شود و کنترل می‌کند که چه مقدار از حافظه‌ی رایانه توسط برنامه‌ی AI Config.vi جهت فرآیند DAQ در نظر گرفته شده است. ورودی interchannel delay میزان تأخیر در نظر گرفته شده بین زمان نمونه‌برداری کانال‌ها را تعیین می‌کند.

در اینجا به بررسی دو مفهوم Channel Interval و Scan Interval می‌پردازیم. در شکل ۱۶-۱۱ تفاوت بین این دو مطلب نشان داده شده است. اعداد مندرج در این شکل شماره‌ی کانال‌های یک کارت DAQ هستند.




**شکل ۱۶-۱۱: تفاوت بین دو مفهوم Channel Interval و Scan Interval**

برنامه‌ی AI Start.vi عملیات دریافت اطلاعات ورودی را آغاز می‌کند. این برنامه سرعت فرآیند DAQ و تعداد نقاطی که باید جمع‌آوری شوند را کنترل می‌کند و می‌توان هر یک از دو حالت تریگر سخت‌افزاری (تریگر داخلی و خارجی) را در مورد این برنامه انتخاب کرد.

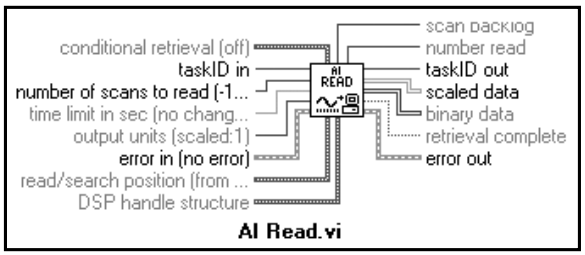



شکل ۱۷-۱۱

۱۱

دو ورودی مهم و حائز اهمیت این برنامه عبارتند از: scan rate و ...number of scans to acqiure. در ورودی scan rate تعداد اسکن‌هایی که در طول یک ثانیه در مورد هر یک از کانال ها باید انجام گیرد تعیین می شود. در ورودی ...number of scans to acquire تعداد اسکن هایی که باید از لیست کانال ها انجام گیرد مشخص می گردد.

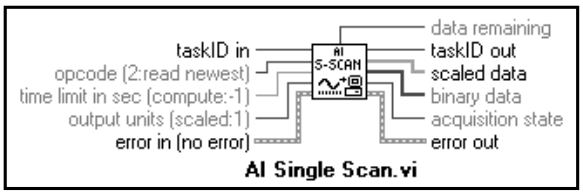
برنامه‌ی AI Read.vi داده ها را از بافری که توسط برنامه‌ی AI Config.vi در حافظه اختصاص داده شده است می خواند. این برنامه، محل داده‌ها در بافر و همچنین تعداد نقاطی که باید از بافر خوانده شود را کنترل می کند.




شکل ۱۸-۱۱

همچنین به کمک این برنامه می توان تعیین کرد که داده‌ها در کدام یک از دو خروجی binary data یا scaled data ظاهر شوند. خروجی این برنامه یک آرایه‌ی دوبعدی است که هر ستون آن شامل داده های یکی از کانال های موجود در لیست کانال هاست.

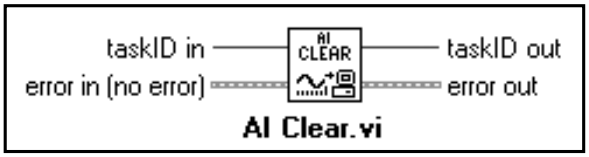
برنامه‌ی AI Single Scan.vi یک اسکن از داده ها در خروجی خود را به دست می دهد. دو خروجی binary data یا scaled data حاوی مقادیر ولتاژ قرائت شده از هر یک از کانال های موجود در لیست کانال ها هستند.




شکل ۱۹-۱۱

این برنامه تنها به همراه برنامه‌ی AI Config.vi به کار می رود. در صورت به کارگیری این برنامه نیازی به استفاده از برنامه های AI Start.vi و AI Read.vi نمی باشد.

برنامه‌ی AI Clear.vi عملیات ورودی آنالوگ را به پایان می‌برد، همچنین بافر را از حافظه‌ی رایانه حذف می کند و به طور کلی هر گونه منبعی از قبیل شمارنده‌ها را که در طی فرآیند DAQ مورد استفاده قرار گرفته است، ری‌ست می کند.



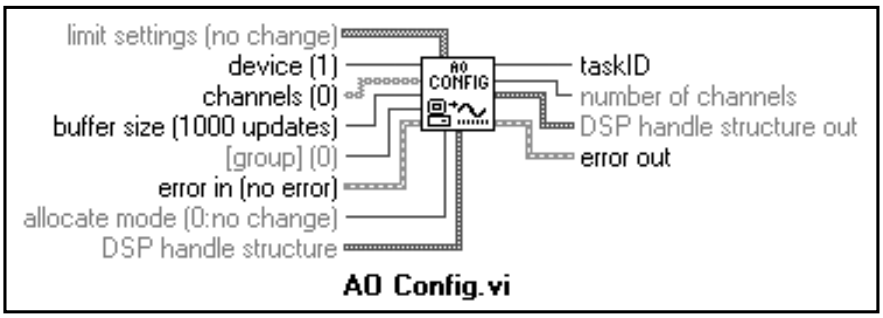

شکل ۲۰-۱۱

برنامه‌ی AI Config.vi اولین برنامه‌ای است که همواره در آغاز فرآیند DAQ در برنامه‌های ورودی آنالوگ استفاده می‌شود. این برنامه اطلاعات لازم جهت پیکربندی را در taskID قرار می‌دهد و همچنین یک کلاستر خطا ایجاد می‌کند.

دیگر برنامه‌های ورودی آنالوگ، اطلاعات taskID in را به عنوان ورودی دریافت می‌کنند و بدین ترتیب شماره‌ی کارت و کانال یا کانال‌هایی که باید فرآیند DAQ بر روی آنها انجام گیرد برای تمامی برنامه‌ها تعیین می‌گردد. با این روش، تعریف نمودن پارامترهای مذکور در تمامی برنامه‌ها الزامی نیست. هر یک از برنامه‌ها پس از اتمام عملیات، اطلاعات مذکور را در خروجی taskID out آماده می‌سازند. به دلیل اینکه taskID به عنوان ورودی و خروجی در تمامی برنامه‌های آنالوگ وجود دارد، این پارامتر بین برنامه‌های موجود در فرآیند DAQ وابستگی اطلاعاتی[4] برقرار می‌کند و بدین ترتیب، روند اجرای فرآیند کنترل می‌شود. با این روش برنامه‌های ورودی آنالوگ به درستی و با ترتیب صحیح اجرا می‌شوند.

## خروجی آنالوگ

برنامه‌ی AO Config.vi عملیات خروجی آنالوگ را در مورد مجموعه‌ای از کانال‌ها پیکربندی می‌کند و برای انجام این عملیات یک بافر در حافظه‌ی رایانه اختصاص می‌دهد.



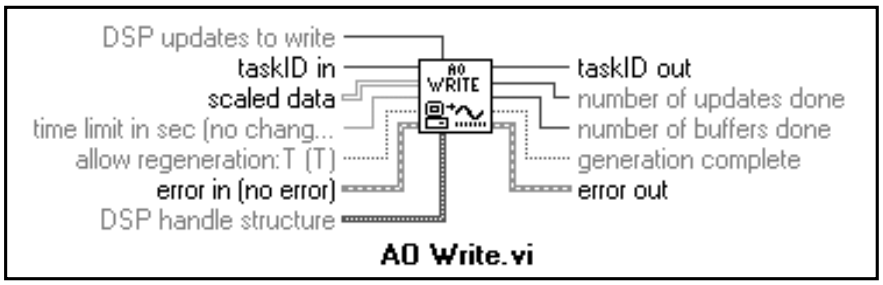

شکل ۲۱-۱۱

شماره‌ی اختصاص داده شده به کارت DAQ، در ورودی device تعیین می‌گردد. شماره‌ی کانال‌های خروجی آنالوگ به صورت آرایه‌ی رشته‌ای در ورودی channels مشخص می‌شود. ورودی limit settings حدود سیگنال‌های خروجی را تعیین می‌کند. خروجی taskID نیز توسط تمامی برنامه‌های خروجی آنالوگ مرتبط با این برنامه دریافت می‌شود و بدین ترتیب پارامترهایی نظیر شماره‌ی کارت و کانال یا کانال‌هایی که فرآیند DAQ در مورد آنها صورت می‌گیرد به تمامی برنامه‌های آنالوگ خروجی انتقال داده می‌شود.

برنامه‌ی AO Write.vi آرایه‌ی ورودی scaled data را در بافر تعیین شده جهت انجام عملیات خروجی آنالوگ ثبت می‌کند. داده‌های ورودی مذکور به صورت آرایه‌ی دوبعدی هستند که داده‌های هر ستون در یکی از کانال‌های موجود در لیست کانال‌ها نوشته می‌شود.

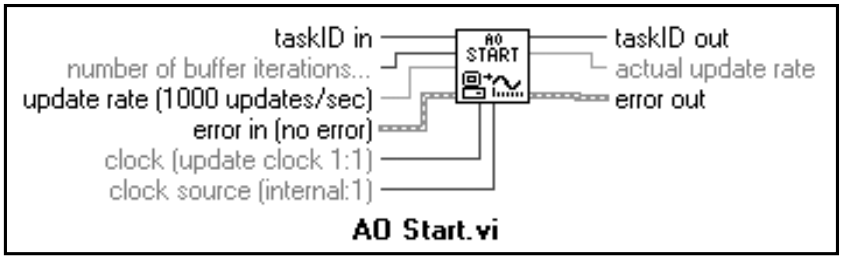
4- Data Dependency

۱۱




شکل ۲۲-۱۱

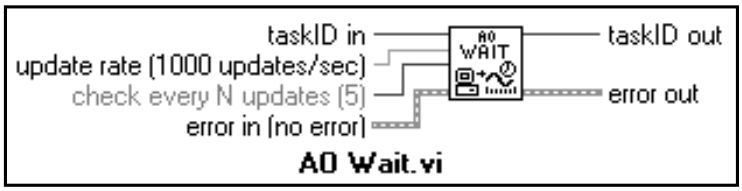
برنامه‌ی AO Start.vi عملیات خروجی آنالوگ را آغاز می کند. در ورودی update rate میزان به روز رسانی داده ها در یک ثانیه تعیین می گردد.




شکل ۲۳-۱۱

در صورتی که عدد 0 را به ورودی number of buffer iterations... سیم‌کشی کنید تا زمانی که برنامه‌ی AO Clear.vi را به اجرا در نیاورید، همواره محتویات بافر در خروجی کارت DAQ ظاهر خواهد شد.

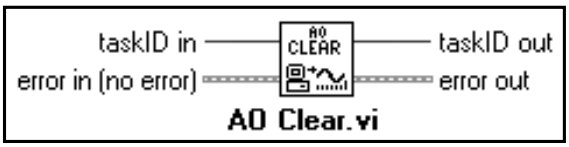
برنامه‌ی AO Wait.vi باعث ایجاد تأخیر در عملکرد سیستم می گردد. این تأخیر تا پایان فرآیند تولید شکل موج ادامه می یابد.




شکل ۲۴-۱۱

این برنامه وضعیت عملیات را بررسی می‌کند و برای آزادسازی پردازنده جهت انجام اعمال دیگر بین بازه‌های زمانی منتظر می‌ماند. زمان انتظار با تقسیم کردن ورودی اختیاری Check every N updates (که به صورت پیش‌فرض 5 در نظر گرفته شده است) به ورودی update rate محاسبه می شود.

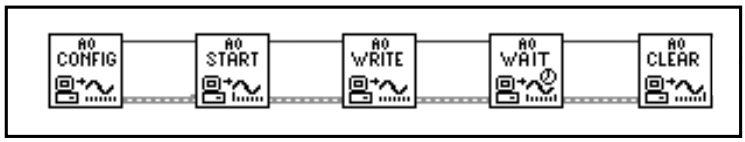
برنامه‌ی AO Clear.vi عملیات خروجی آنالوگ را متوقف ساخته، همچنین بافر را از حافظه‌ی رایانه حذف می کند و به طور کلی هر گونه منبعی از قبیل شمارنده ها را که در طی فرآیند DAQ مورد استفاده قرار گرفته است، ری‌ست می سازد.




شکل ۲۵-۱۱

۱۱

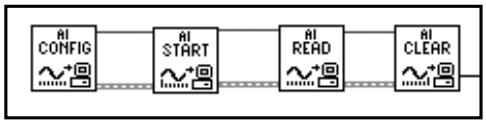
تمامی برنامه هایی که در دو بخش قبل مورد بررسی قرار گرفتند در سیستم جریانِ داده به طور تنگاتنگ با یکدیگر در ارتباط می باشند. همان گونه که احتمالاً تاکنون دریافته اید، ترتیب قرار گرفتن این برنامه ها از قاعده‌ی خاصی پیروی می کند. برنامه های آنالوگ ورودی/خروجی از طریق taskID و لیست خطاها زنجیروار به یکدیگر متصل شده اند. در شکل ۲۶-۱۱ ترتیب این برنامه ها را در فرآیند تولید داده ها ملاحظه می کنید.




شکل ۲۶-۱۱: ترتیب برنامه ها در فرآیند تولید داده ها

۱- AO Config.vi : بافر و کانال ها را تعیین می کند.
۲- AO Start.vi : عملیات تولید داده‌ها را آغاز می کند.
۳- AO Write.vi : داده های جدید را در بافر ثبت می کند.
۴- AO Wait.vi : (اختیاری است) تا زمان خالی شدن بافر به انتظار می ماند.
۵- AO Clear.vi : بافر را پاک می کند و تمامی منابع را آزاد می سازد.

در شکل ۲۷-۱۱ ترتیب این برنامه ها در فرآیند جمع آوری داده ها نشان داده شده است.




شکل ۲۷-۱۱: ترتیب برنامه ها در فرآیند جمع آوری داده ها

۱- AI Config.vi : بافر و کانال ها را تعیین می کند.
۲- AI Start.vi : عملیات جمع آوری داده ها را آغاز می کند.
۳- AI Read.vi : داده ها را از بافر می خواند.
۴- AI Clear.vi : بافر را پاک می کند و تمامی منابع را آزاد می سازد.

## ورودی آنالوگ همراه بافر

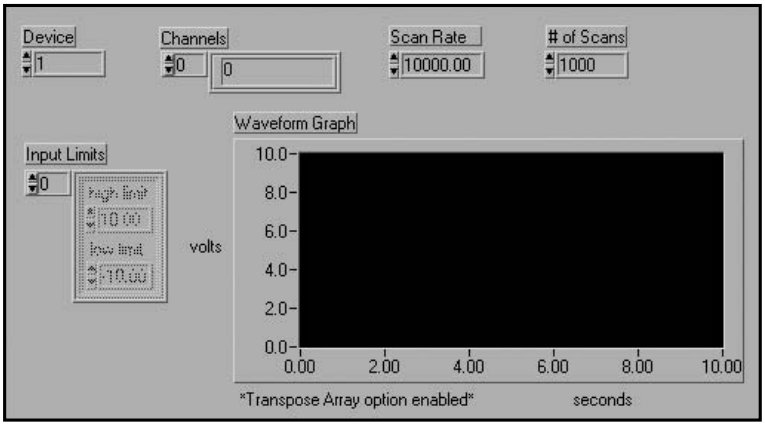
آیا بخش I/O همراه بافر را که در ابتدای این فصل به بررسی آن پرداختیم به خاطر می آورید؟ در این بخش می توانید نحوه‌ی اجرای آن را ملاحظه کنید. در تمرین ۱-۱۱ برای جمع آوری نمونه های حاصل از چند کانال،

۱۱

از یک حلقه‌ی While استفاده نمودیم. حال برای انجام این عمل از برنامه‌های سطح متوسط استفاده نموده و خواهید دید که چگونه با پیکربندی سخت‌افزاری و بدون استفاده از حلقه‌ی While می‌توان به جمع‌آوری مجموعه‌ای از داده‌ها پرداخت.

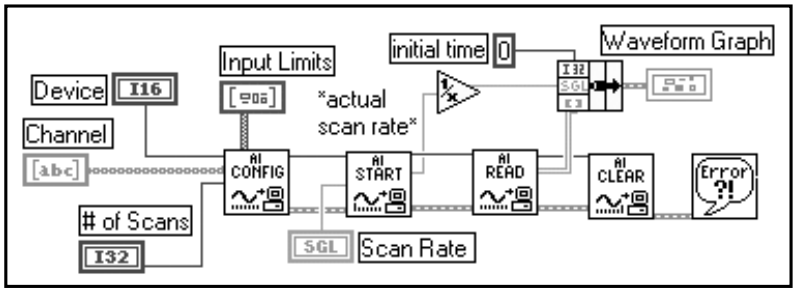
## تمرین ۳-۱۱: انجام عملیات DAQ با استفاده از بافر

این تمرین نمونه‌ای از انجام عملیات DAQ به کمک بافر است. ابتدا صفحه‌ی پانل این برنامه را مطابق شکل ۲۸-۱۱ ایجاد کنید. در این صفحه با به کارگیری یک گراف ساده ابتدا داده‌ها را بررسی می‌کنیم.




**شکل ۲۸-۱۱: صفحه‌ی پانل در برنامه‌ی Buffered Analog In.vi**

توجه داشته باشید که ورودی Input Limits، یک آرایه حاوی داده‌های کلاستری است که هر یک از اجزای کلاستر شامل دو المان Numeric Control می‌باشد. برای ایجاد آن، منوی کرکره‌ای را بر روی ورودی input limits در برنامه‌ی AI Config.vi باز و گزینه‌ی Create Control را انتخاب کنید. سپس یک منبع ولتاژ به کانال 0 در کارت DAQ ارتباط دهید و به ترتیبی که قبلاً اشاره شد برنامه‌های ورودی آنالوگ را مطابق شکل ۲۹-۱۱ در صفحه‌ی نمودار بلوکی این برنامه قرار دهید.




**شکل ۲۹-۱۱: صفحه‌ی نمودار بلوکی در برنامه‌ی Buffered Analog In.vi**

این برنامه را با عنوان Buffered Analog In.vi در زیرفهرست LabVIEW\My Activity ذخیره کنید.

با بررسی عملکرد این برنامه درمی‌یابید که به کمک دستورها و برنامه‌های موجود می‌توان یک دستگاه اسیلوسکوپ بسیار ساده ساخت. در ادامه به بررسی برخی از دستورها و همچنین المان‌های استفاده شده در برنامه می‌پردازیم:

ورودی Input Limits جهت تنظیم بهره‌ی کارت به کار برده می‌شود. به عنوان مثال اگر حدود بالا و پایین را به ترتیب 100 Vm+ و 100 Vm- تنظیم کنید، به بهره‌ی 100 دست خواهید یافت.

برنامه‌ی AI Start.vi فرآیند DAQ را تریگر می کند. به محض اتمام اجرای این برنامه، داده ها در داخل بافر بارگذاری می شوند. تا زمانی که تصمیم به خواندن داده های بافر نگرفته اید دسترسی به آنها امکان پذیر نیست. به همین دلیل برنامه‌ی AI Read.vi بلافاصله پس از این برنامه قرار گرفته است. بدین ترتیب می توان قبل از اینکه داده های دیگری بر روی داده های بارگذاری شده در بافر نوشته شده یا داده های قبلی به کلی از بافر پاک شوند، به آنها دست یافت و داده ها را قرائت نمود.

عکس مقدار actual scan rate در برنامه‌ی AI Start.vi برای ترسیم داده ها بسیار مفید است. زیرا به کمک آن می توان مقدار $\Delta X$ یا فاصله‌ی بین داده ها را در منحنی ترسیم شده تعیین نمود. اگر مقدار actual scan rate با نرخ اسکن استفاده شده در برنامه سازگار نباشد، این عدم تطابق در گراف منعکس می گردد.

به کارگیری برنامه‌ی AI Clear.vi بسیار مهم و قابل توجه است. زیرا همان گونه که می دانید این برنامه داده های موجود در بافر را پاک می کند. در صورت عدم استفاده از این برنامه، برنامه‌ی AI Read.vi بعدی ممکن است داده های قبلی موجود در بافر را بخواند.

## تمرین ۴-۱۱: نمونه‌برداری و ذخیره‌ی داده‌ها

در این تمرین قصد داریم تا برنامه‌ی قبلی را اصلاح نماییم و در آن تغییراتی ایجاد کنیم. در این تمرین با ایجاد یک برنامه به ذخیره و سپس نمایش داده ها می پردازیم. صفحه‌ی پانل این برنامه شبیه به همین صفحه در تمرین قبل است. با این تفاوت که در این حالت می خواهیم تمامی داده ها را در یک فایل صفحه گسترده ذخیره نماییم. این فایل باید به گونه‌ای باشد که کاربر بتواند نام آن را تغییر داده و به دلخواه آن را نام گذاری کند. این برنامه باید قدرت استخراج داده های قدیمی را از فایل مذکور داشته و بتواند آنها را بر روی نمودار ترسیم کند.

پاسخ این تمرین با عنوان Acquire and Save.vi در مسیر LabVIEW\Exercise\Ch11.llb وجود دارد.

## تمرین ۵-۱۱: شبیه‌سازی دستگاه مولد موج

در این تمرین قصد داریم تا با ایجاد یک برنامه، عملکرد دستگاه مولد موج را شبیه سازی کنیم. این برنامه یک موج ورودی را از کارت DAQ دریافت می کند و در خروجی آنالوگ، موج تجزیه شده را به دست می دهد. در این برنامه کاربر می تواند یکی از اشکال موج سینوسی، مربعی و دندانه اره‌ای را انتخاب نماید و آنها را در خروجی مشاهده کند.

پاسخ این تمرین با عنوان Function Generator.vi در مسیر LabVIEW\Exercise\Ch11.llb وجود دارد.

۱۱

## مباحث پیشرفته در ورودی/خروجی آنالوگ

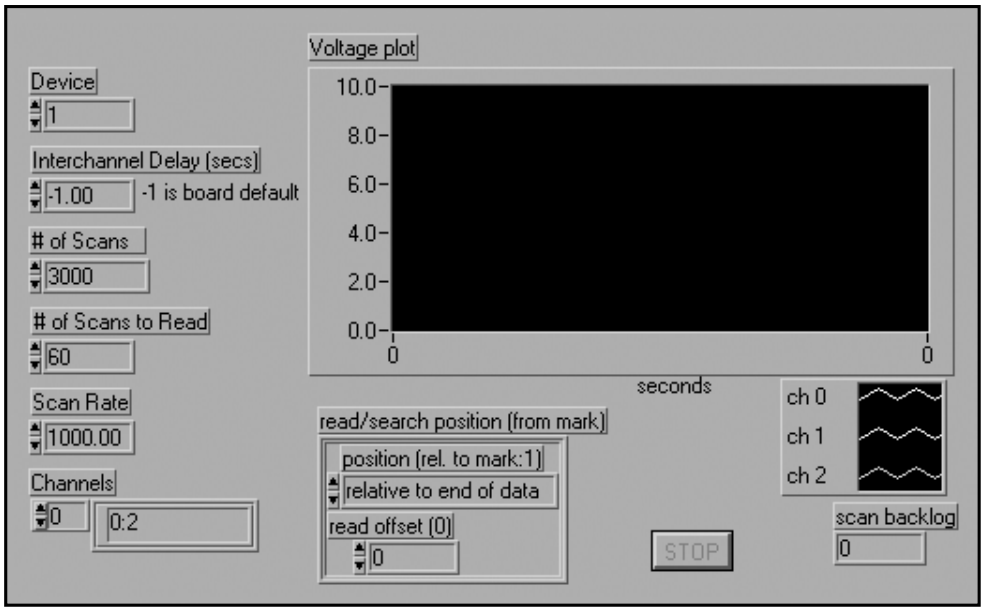
در این مرحله به ارائه‌ی توضیح در مورد برخی از اصطلاحات و مفاهیم نظیر بافر چرخشی[5]، جمع آوری داده‌ها به طور پیوسته، تریگر سخت‌افزاری و... می‌پردازیم. دانستن این اصطلاحات بسیار مفید و ضروری است. با ارائه‌ی یک مثال این مفاهیم را مورد بحث و بررسی قرار خواهیم داد.

## جمع آوری داده‌ها به طور پیوسته

در جمع آوری داده‌ها به طور پیوسته، فرآیند DAQ بدون ایجاد وقفه انجام می گیرد. در این فرآیند از یک بافر چرخشی استفاده می شود. بدین ترتیب که کارت DAQ داده‌ها را جمع آوری می‌نماید و آنها را در این بافر ذخیره می کند. به محض پرشدن فضای بافر، داده‌های جدید بر روی داده‌های ذخیره شده‌ی قبلی نوشته می شوند، خواه این داده‌ها توسط LabVIEW خوانده شده باشند یا خیر. این فرآیند برای مواردی نظیر جاری ساختن داده‌ها[6] به دیسک و نمایش داده‌ها به صورت بلادرنگ[7] بسیار مفید است.

### تمرین ۶-۱۱: جمع آوری داده‌ها به طور پیوسته

۱- یک برنامه‌ی جدید باز نمایید و صفحه‌ی پانل آن را مطابق شکل ۳۰-۱۱ ایجاد کنید.



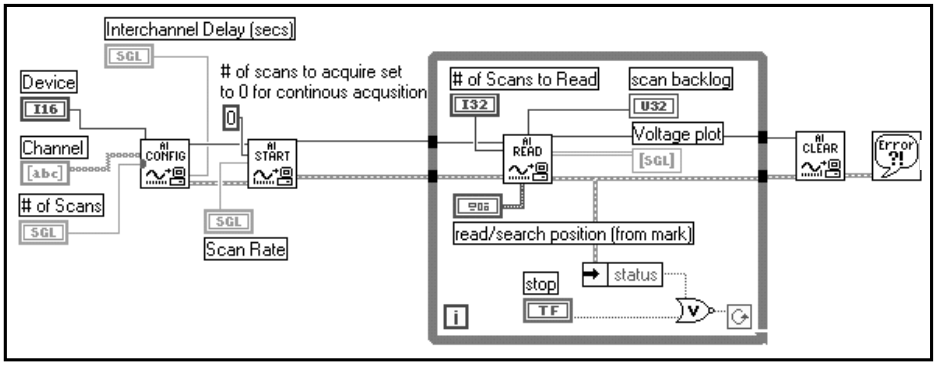

**شکل ۳۰-۱۱: صفحه‌ی پانل در برنامه‌ی Continuous Acquisition.vi**

برای ایجاد کلاستر read/search position، منوی کرکره‌ای را بر روی ورودی read/search position در برنامه‌ی AI Read.vi باز نمایید، سپس گزینه‌ی Create Indicator را انتخاب کنید.

5- Circular Buffer
6- Streaming Data
7- Realtime

۱۱

۲- این برنامه را با عنوان Continuous Acquisition.vi در زیرفهرست LabVIEW\My Activity ذخیره کنید. سپس برنامه های ورودی آنالوگ را مطابق شکل ۳۱-۱۱ در صفحه‌ی نمودار بلوکی آن قرار دهید.




**شکل ۳۱-۱۱: صفحه‌ی نمودار بلوکی در برنامه‌ی Continuous Acquisition.vi**

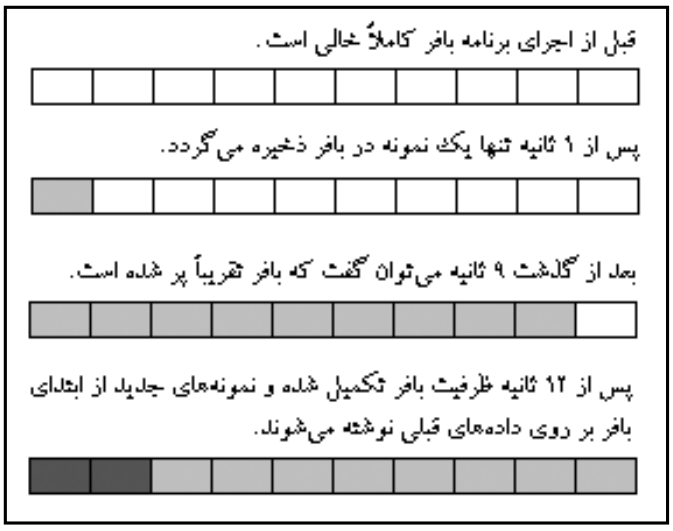
۳- با استفاده از برنامه‌ی AI Start.vi، داده ها را به طور پیوسته جمع آوری کنید. برای انجام این عمل، به ورودی ...number of scans to acquire در برنامه‌ی AI Start.vi عدد 0 نسبت دهید. این نوع فرآیند DAQ را عملیات غیرهمزمان[8] گویند. بدین مفهوم که اعمال دیگر LabVIEW می توانند بدون ایجاد مشکل در طول انجام فرآیند DAQ به اجرا در آیند.

در صفحه‌ی نمودار بلوکی این برنامه ملاحظه می کنید که برنامه‌ی AI Read.vi در یک حلقه‌ی While قرار دارد و بدین روش داده ها را از بافر می خواند. پس از آن می توانید برای نمایش داده ها، آنها را به گراف ارسال کنید. برنامه‌ی AI Clear.vi فرآیند DAQ را متوقف نموده، بافر را تخلیه می کند و هرگونه منبعی از کارت که درحین انجام فرآیند DAQ مورد استفاده قرار گرفته است را ریست می کند. به تفاوت بین این برنامه که عملیات DAQ را به طور پیوسته انجام می دهد و برنامه‌ای که داده ها را برای رسم یک شکل موج جمع آوری می کرد (برنامه‌ی مخصوص فرآیند DAQ به صورت معمولی که در بخش قبل مورد بررسی قرارگرفت) توجه کنید. همان گونه که احتمالاً دریافته اید تفاوت این دو برنامه، در استفاده از حلقه‌ی While، انتساب عدد 0 به ورودی ...number of scans to acquire در برنامه‌ی AI Start.vi و پارامتر number of scans to read است.

برنامه‌ی قبلی برای جمع آوری داده ها به طور پیوسته طراحی شده بود. برای انجام این برنامه لازم است ابتدا اطلاعات جمع آوری شده در محلی از حافظه‌ی رایانه ذخیره شوند. در صورتی که داده ها به طور پیاپی در حافظه ذخیره شوند پس از مدتی حافظه‌ی مذکور مملو از داده های جمع آوری شده می گردد. به همین دلیل یک بافر چرخشی با طول ثابت در حافظه در نظر گرفته شده است. در هنگام انجام فرآیند DAQ، داده ها در این بافر از ابتدا تا انتها ذخیره می شوند و پس از اتمام ظرفیت آن، مجدداً داده های جدید از ابتدا بر روی داده های قبلی نوشته می شوند. به طور مثال فرض کنید که یک برنامه نمونه‌ها را با نرخ ۱ نمونه در ثانیه جمع آوری می کند و بافر اختصاص داده شده جهت ذخیره‌ی داده ها می‌تواند حداکثر ۱۰ نمونه را در خود ذخیره کند. شکل ۳۲-۱۱ عملکرد این بافر را به هنگام قرائت داده ها در چند مرحله نشان می دهد.

8- asynchronous

۱۱




**شکل ۳۲-۱۱: عملکرد بافر چرخشی به هنگام قرائت داده‌ها در مراحل مختلف**

در تمرین ۶-۱۱ جهت بهره‌گیری از قابلیتها و مزایای زمان‌بندی سخت‌افزاری و استفاده‌ی بهینه از حافظه در LabVIEW، از برنامه‌های سطح متوسط استفاده شد. اگر در زمانی که برنامه‌ی AI Single Channel.vi در حال اجراست، پردازنده درگیر انجام یکی از اعمال خاص سیستم عامل شود برنامه‌های سطوح میانی از قابلیت DMA (در صورت وجود) و بافر موجود در کارت DAQ استفاده می‌کنند و بدون ایجاد وقفه در انجام عملیات پردازنده، فرآیند DAQ را ادامه می‌دهند.

DMA به سخت‌افزار کارت DAQ اجازه می‌دهد تا بدون ایجاد وقفه در عملیات پردازنده داده‌ها را مستقیماً به حافظه‌ی رایانه انتقال دهد. تنها در صورتی که زمان انجام عملیات توسط سیستم عامل بیش از مدت زمان قابل تحمل توسط بافر FIFO و بافر DMA باشد، تعدادی از داده‌ها از دست خواهند رفت. برای درک بیشتر این مطلب مجدداً به شکل ۳۲-۱۱ نگاه کنید. ابتدا فرض کنید که کارت DAQ فاقد بافر FIFO است اما دارای ویژگی DMA است. بنابراین کارت DAQ داده‌های جمع‌آوری شده را به طور پیوسته در بافر ذخیره می‌کند.

برنامه‌های استفاده شده در LabVIEW نیز به طور دائم نمونه‌هایی را که قبل از آخرین داده‌ی ذخیره شده در بافر قرار دارند قرائت می‌کنند. حال فرض کنید پس از یک ثانیه در حالی که LabVIEW اولین داده‌ی ذخیره شده در بافر را خوانده است، پردازنده درگیر انجام عملیات دیگری می‌شود. اگر پردازنده مشغول انجام عملیات باشد کارت DAQ به جمع‌آوری داده‌ها ادامه می‌دهد و آنها را در بافر ذخیره می‌کند. اگر پس از ۱۰ ثانیه باز هم پردازنده درگیر باشد و انجام عملیات پردازش توسط پردازنده به اتمام نرسیده باشد، داده‌ی جدیدی بر روی اولین داده از سمت چپ نوشته می‌شود که ممکن است حاوی اطلاعات مفیدی باشد. در صورت بروز این مشکل برنامه‌های سطح متوسط، خطای ایجاد شده را آشکار می‌سازند.

با دقت در صفحه‌ی نمودار بلوکی برنامه‌ی Continuous Acquisition.vi ملاحظه می‌کنید که یک خروجی scan backlog در برنامه‌ی AI Read.vi در نظر گرفته شده است. این نشان‌دهنده در گوشه‌ی پایینی سمت راست در صفحه‌ی پانل نیز به چشم می‌خورد. حتماً تاکنون دلیل استفاده از این نمایشگر را استنباط نموده‌اید؟

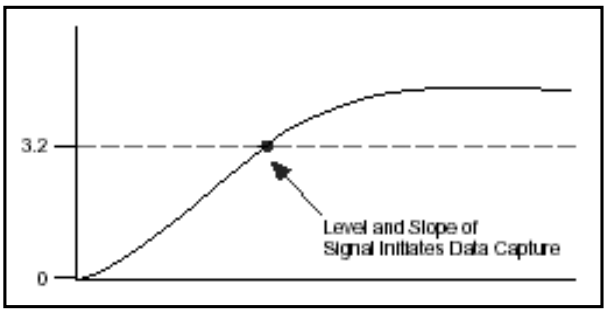
دانستن این مطلب که آیا در هنگام انجام فرآیند DAQ نمونه هایی از دست رفته‌اند بسیار مهم است. در صورتی که سرعت پر شدن بافر، بیش از سرعت استخراج داده‌ها از بافر باشد، تعدادی از داده‌ها از دست خواهند رفت و به دلیل اینکه داده‌های جدید مجدداً بر روی داده‌های ذخیره شده در بافر نوشته می‌شوند خروجی scan backlog این مسأله را آشکار می‌سازد.

## تریگر

برای آغاز نمودن فرآیند DAQ دو روش وجود دارد: تریگر به روش نرم‌افزاری و سخت‌افزاری.

در تریگر نرم‌افزاری آغاز فرآیند DAQ منوط به صدور دستورهای نرم‌افزاری است. به عنوان مثال در تمامی برنامه‌های ساده‌ی DAQ از تریگر نرم‌افزاری استفاده می‌شود. در این روش به محض اجرا شدن برنامه، فرآیند جمع‌آوری و تولید داده‌ها آغاز می‌گردد. تمامی کارت‌های DAQ را می‌توان به روش نرم‌افزاری تریگر نمود.

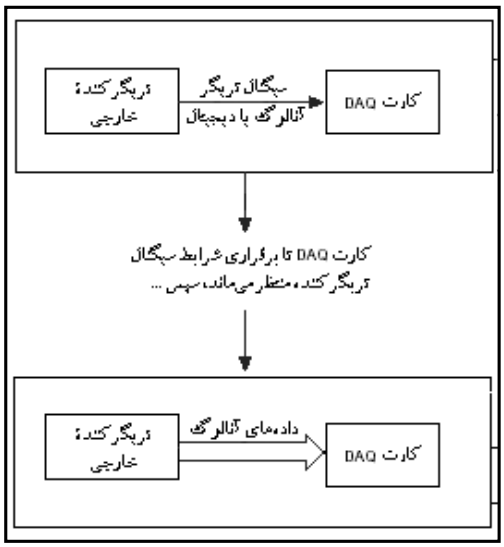
در تریگر سخت‌افزاری سیگنال آغازکننده‌ی فرآیند DAQ توسط سخت‌افزار صادر می‌شود و برای شروع نمونه‌برداری باید منتظر وقوع یک حادثه بود. به عبارت دیگر آغاز فرآیند DAQ منوط به وقوع یک حادثه‌ی خارجی خاص است. در روش مذکور معمولاً فرمان آغاز فرآیند DAQ توسط وضعیت، شیب، دامنه‌ی سیگنال و... صادر می‌شود. در شکل ۳۳-۱۱ ملاحظه می‌کنید که ابتدا شرایط یک سیگنال آنالوگ (که در اینجا مقدار دامنه و شیب سیگنال است) بررسی شده، سپس عملیات مورد نظر تریگر می‌شود.




**شکل ۳۳-۱۱: بررسی شرایط یک سیگنال آنالوگ برای تریگر کردن عملیات مورد نظر**

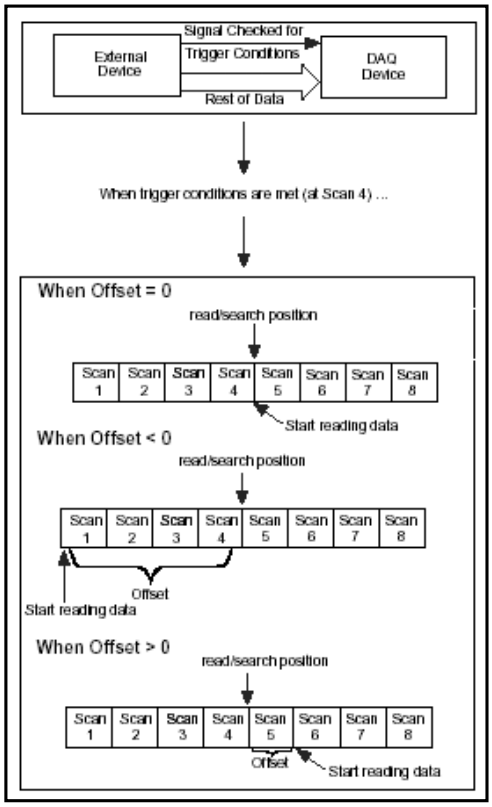
در بعضی از کارت‌ها از تریگر آنالوگ استفاده می‌شود، در حالی که برخی از کارت‌ها تریگر دیجیتال را به کار می‌برند. بدین معنی که فرمان آغاز فرآیند DAQ توسط یک سیگنال دیجیتال ارسال می‌گردد. پین ورودی کارت DAQ که جهت اعمال تریگر سخت‌افزاری به کارت در نظر گرفته شده است با یکی از عناوین START TRIG یا EXT TRIG مشخص می‌گردد. در تمامی کارت‌های سری MIO و Lab (محصولات شرکت NI) از تریگر دیجیتالی استفاده می‌شود. در شکل ۳۴-۱۱ عملکرد سیگنال تریگر خارجی نشان داده شده است.

۱۱




**شکل ۳۴-۱۱: عملکرد سیگنال تریگر خارجی**

همچنین می‌توانید برای آغاز نمودن عملیات قرائت داده‌های موجود در بافر، یک سیگنال آنالوگ خارجی را به عنوان تریگرکننده مورد استفاده قرار دهید. این نوع تریگر موسوم به دستیابی شرطی[9] است. در شکل ۳۵-۱۱ عملکرد این نوع تریگر را ملاحظه می کنید.



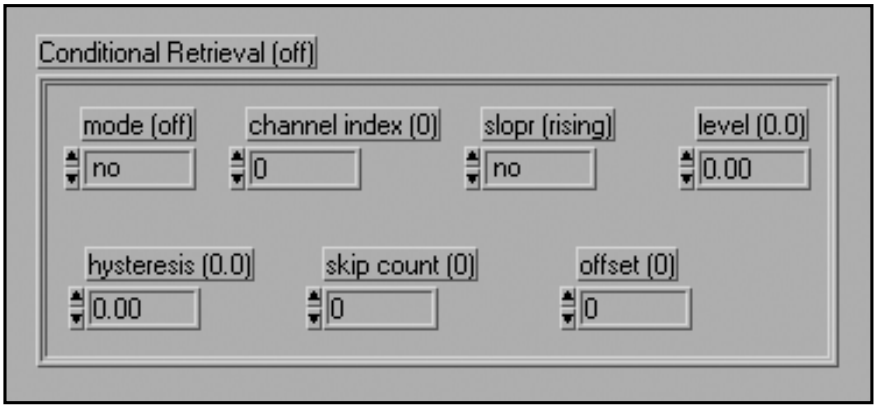

**شکل ۳۵-۱۱: دستیابی شرطی**

9- Conditional Retrieval

۱۱

در صورت استفاده از روش دستیابی شرطی، تریگر همواره به روش نرم‌افزاری صورت می گیرد. انجام فرآیند DAQ به روش دستیابی شرطی منوط به دریافت یک سیگنال با دامنه‌ی معین و شیب خاص می باشد. بدین ترتیب تا زمانی که نمونه‌ای با سطح دامنه‌ی معین و شیب خاص جمع آوری نشده باشد، خواندن داده ها آغاز نمی شود. مراقب باشید که دو مفهوم تریگر سخت افزاری و دستیابی شرطی را با یکدیگر اشتباه نگیرید. در سیستم‌هایی که در آنها از تریگر آنالوگ استفاده می شود، ذخیره‌ی داده‌ها در بافر تا زمان برقراری شرایط تریگر صورت نمی گیرد. اما در سیستم‌هایی که از روش دستیابی شرطی استفاده می کنند داده‌ها در بافر ذخیره می شوند اما در صورت عدم برقراری شرایط تریگر، قرائت داده ها انجام نمی گیرد.

شکل ۳۶-۱۱ نحوه‌ی عملکرد یک کلاستر را در روش دستیابی شرطی نشان می دهد. در کلاستر مذکور شرایط لازم جهت دستیابی به داده های بافر تعیین می شود. این کلاستر یکی از ورودیهای برنامه‌ی AI Read.vi است. پس از آغاز فرآیند، کارت DAQ نمونه برداری از سیگنال را به طور پیوسته انجام می دهد و هر یک از نمونه ها را با شرایط دستیابی مقایسه می کند. پس از برقراری شرایط، برنامه‌ی AI Read.vi به اندازه‌ی تعداد مشخص شده در ورودی number of scans to read داده ها را از بافر می‌خواند و آنها را در خروجی خود ظاهر می سازد.




**شکل ۳۶-۱۱: نحوه‌ی عملکرد یک کلاستر در روش دستیابی شرطی**

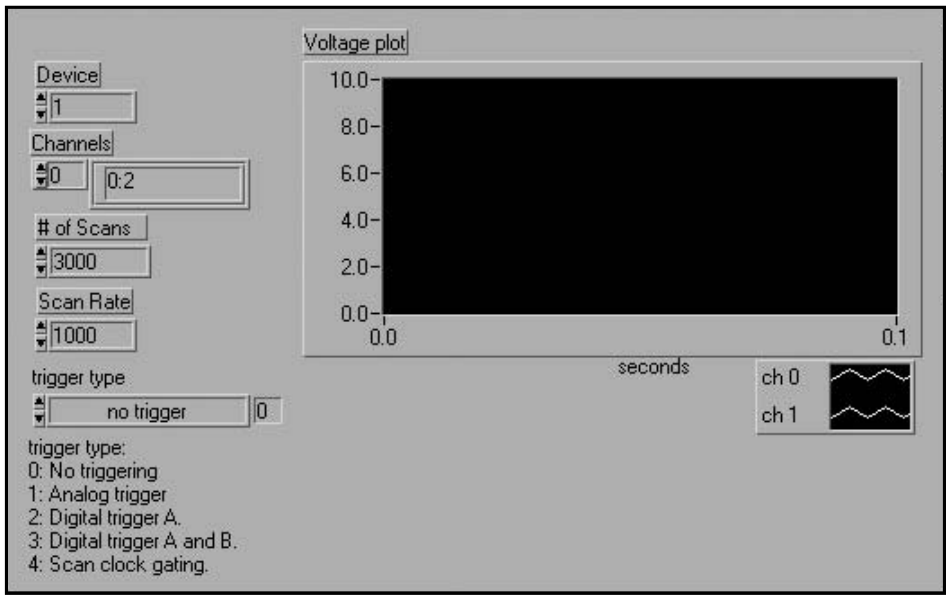
## تمرین ۷-۱۱: اعمال تریگر سخت‌افزاری

یک برنامه‌ی ورودی آنالوگ ایجاد کنید که در آن برای آغاز و اتمام فرآیند DAQ از تریگر سخت‌افزاری استفاده شده باشد.

۱- یک کلید را با یک سیگنال TTL به پین EXT TRIG یا پین های مشابه بر روی کارت DAQ متصل نمایید و یک زوج سیگنال آنالوگ به آن اعمال کنید.

۲- صفحه‌ی پانل این برنامه را مطابق شکل ۳۷-۱۱ ایجاد کنید.

۱۱

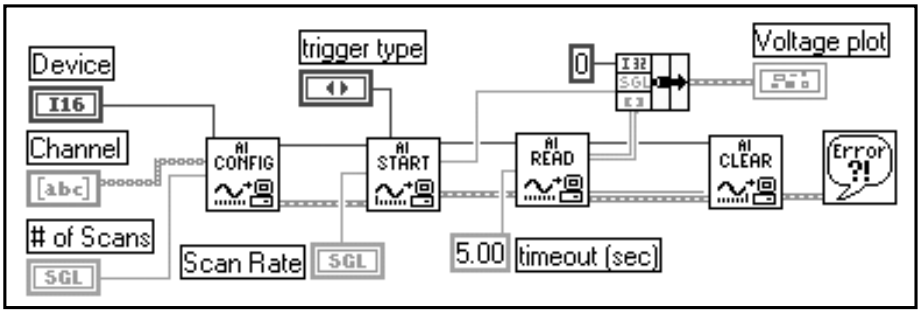



**شکل ۳۷-۱۱: صفحه‌ی پانل در برنامه‌ی Triggered Analog In.vi**

المان trigger type یک منوی حلقوی است که به شما اجازه می دهد تا نوع تریگر را به صورت زیر مشخص کنید:

0: No triggering (default)
1: Analog trigger
2: Digital trigger A
3: Digital trigger A and B
4: Scan clock gating

۳- صفحه‌ی نمودار بلوکی این برنامه را مطابق شکل ۳۸-۱۱ ایجاد نمایید.




**شکل ۳۸-۱۱: صفحه‌ی نمودار بلوکی در برنامه‌ی Triggered Analog In.vi**

با دقت در این صفحه ملاحظه می کنید که تفاوت بین صفحات نمودار بلوکی در این تمرین و ورودی آنالوگ ساده به همراه بافر، تنها در سیم کشی ورودی trigger type در برنامه‌ی AI Start.vi است.

۴- این برنامه را با عنوان Triggered Analog In.vi در زیرفهرست LabVIEW\My Activity ذخیره کنید.

۵- پس از انتخاب حالات مختلف تریگر، این برنامه را اجرا نمایید و در هر حالت، داده های جمع آوری شده را بر روی گراف ملاحظه کنید.

## جاری‌ساختن داده‌ها در فایل

تاکنون تمریناتی را انجام داده‌اید که در آنها، داده‌ها به فایل صفحه‌گسترده ارسال می‌شدند. در این تمرینات ملاحظه نمودید که پس از تکمیل فرآیند DAQ، LabVIEW داده‌ها را به فرمت صفحه‌گسترده تبدیل می‌نماید و آنها را در یک فایل ASCII ذخیره می‌کند. روش دیگر و بعضاً در برخی از حالات روش بهتر آن است که در هنگام انجام فرآیند DAQ داده‌ها در دیسک سخت ذخیره شوند. این روش را جاری‌ساختن داده‌ها در فایل گویند. در این روش در حالی که فرآیند DAQ در حال انجام است همواره جریانی از داده‌ها به سمت دیسک سخت وجود دارد. مزیت استفاده از این روش سرعت بالای آن است. بنابراین می توانید چند فرآیند DAQ را به طور پیوسته انجام دهید و در عین حال یک نسخه‌ی کپی از تمامی داده‌های نمونه‌برداری شده بر روی دیسک داشته باشید.

سرعت دستیابی داده‌ها به بافر و سپس انتقال آنها به دیسک یک پارامتر بحرانی و تعیین‌کننده است. سرعت خواندن و سپس جاری‌ساختن داده‌ها باید به اندازه‌ی کافی بالا باشد به طوری که داده‌های جدید مجدداً بر روی داده‌های قبلی در بافر چرخشی نوشته نشوند. جهت افزایش سرعت دستیابی و دریافت داده‌ها باید در هنگام نمونه‌برداری، از به‌کارگیری توابع و دستورهای دیگری نظیر توابع آنالیزکننده خودداری نمود. بنابراین تا حد امکان از انجام همزمان فرآیند DAQ و آنالیز داده‌ها خودداری کنید.

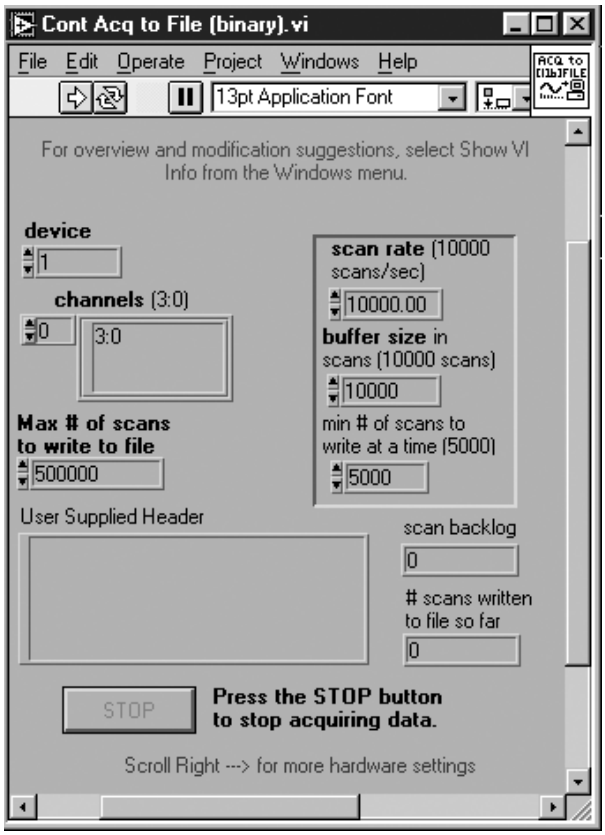
همچنین برای افزایش سرعت دستیابی به داده‌ها و جاری‌ساختن آنها می توان به جای خروجی scaled data در برنامه‌ی AI Read.vi از خروجی binary data استفاده نمود. هنگامی که برنامه‌ی AI Read.vi را به‌گونه‌ای استفاده می‌کنید که در خروجی آن تنها داده‌های باینری تولید شود، سرعت استخراج و خواندن داده‌ها بیش از حالتی است که ابتدا داده‌ها را در یک آرایه جمع‌آوری نمایید و سپس از برنامه‌های File I/O استفاده کنید. یکی از معایب خواندن و جاری ساختن داده‌ها در فایل آن است که خواندن فایل توسط کاربر یا برنامه‌های دیگر به آسانی امکان‌پذیر نیست.

به راحتی می توانید برنامه‌های DAQ را مطابق با ویژگیهای روش جاری ساختن داده‌ها تغییر دهید و آنها را اصلاح کنید. اگر تا به حال با فایل‌ها و داده‌های باینری تمرین نکرده‌اید نگران نباشید. در فصل ۱۴ این داده‌ها را بررسی خواهیم نمود. در تمرین بعد نیز با ایجاد یک برنامه، داده‌ها و فایل‌های باینری را مورد بررسی قرار می دهیم.

### تمرین ۸-۱۱: جاری ساختن داده‌ها در فایل

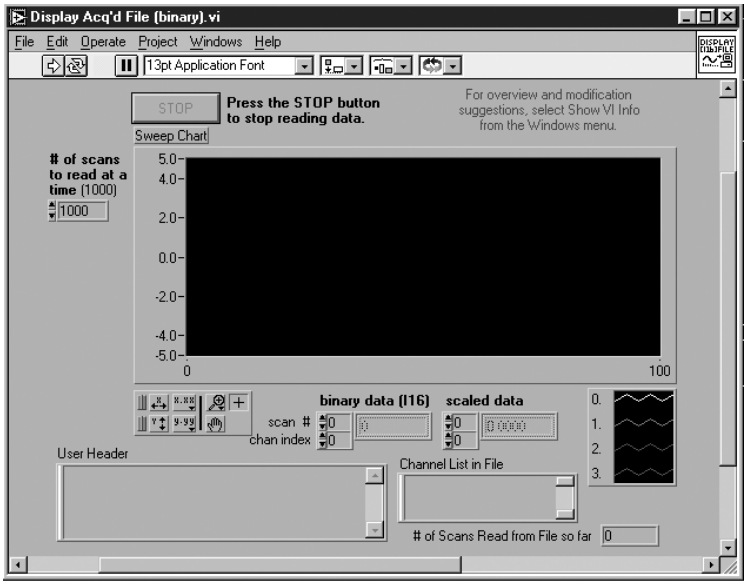
برنامه‌ی Cont Acq to File (binary).vi را از مسیر LabVIEW\ Examples\ daq\ anlogin\ strmdisk.llb باز کنید. این برنامه نحوه‌ی جریان یافتن داده‌ها به فایل را نشان می دهد.

۱۱




**شکل ۳۹-۱۱: صفحه‌ی پانل در برنامه‌ی Cont Acq to File (binary).vi**

پس از اتصال یک سیگنال به کارت DAQ، این برنامه را به مدت چند ثانیه اجرا نمایید. برای مشاهده‌ی جریان داده‌ها به فایل، برنامه‌ی دیگری با عنوان Display Acq'd File (binary).vi را از همان مسیر باز کنید.




**شکل ۴۰-۱۱: صفحه‌ی پانل در برنامه‌ی Display Acq'd File (binary).vi**

۱۱

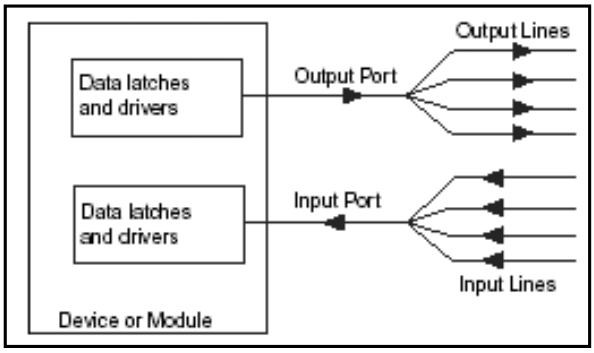
این برنامه را با انتخاب نام فایلی که در برنامه‌ی قبلی تعیین نمودید اجرا کنید. توجه داشته باشید اولین برنامه‌ای که داده‌ها را به داخل فایل انتقال می‌دهد در ترسیم داده‌ها بر روی گراف هیچ نقشی ندارد. مسأله‌ی مهم و اساسی در روش جاری ساختن داده‌ها، جمع آوری و انتقال سریع داده‌ها به فایل است. بررسی و تجزیه و تحلیل داده‌ها به مراحل بعدی و فراغت از انجام فرآیند DAQ موکول می‌گردد. برای درک نحوه‌ی عملکرد هر دو برنامه، صفحات نمودار بلوکی آنها را به دقت بررسی کنید.

## ورودی/خروجی دیجیتال

به طور کلی می‌توان گفت که تجزیه و تحلیل و به کارگیری سیگنال‌های دیجیتال به مراتب از سینگال‌های آنالوگ ساده‌تر است. همان گونه که می‌دانید سیگنال‌های دیجیتال به یکی از دو حالت 0 یا 1، خاموش یا روشن، پایین یا بالا دیده می‌شوند. در این بخش جهت مطالعه‌ی سیگنال‌های دیجیتال، داشتن اطلاعات کافی در مورد فایل‌ها و داده‌های باینری الزامی است. بنابراین قبل از ورود به این مبحث به تعریف چند اصطلاح می‌پردازیم:

Line: خط دیجیتال. هم‌ارز با یک کانال آنالوگ است و طبق تعریف، مسیری است که در آن یک سیگنال دیجیتال منفرد دریافت یا ارسال می‌گردد. خطوط دیجیتال اغلب به یکی از دو حالت ورودی یا خروجی هستند. اما برخی مواقع ممکن است این خطوط دوجهتی باشند. در برخی از کارت‌ها که خطوط دیجیتال نمی‌توانند در هر لحظه هم به عنوان ورودی و هم به عنوان خروجی عمل کنند باید نوع خطوط را قبل از آغاز فرآیند DAQ تعیین نمود.

Port: درگاه. مجموعه‌ای از خطوط دیجیتال است که هم‌جهت هستند یعنی همگی یا به صورت ورودی یا خروجی می‌باشند و می‌توان در هر لحظه همگی آنها را با هم استفاده نمود. تعداد خطوط دیجیتال در هر درگاه به نوع کارت بستگی دارد. اما اکثر درگاهها شامل ۴ یا ۸ خط می‌باشند. به عنوان مثال کارت AT-MIO-16E-10 دارای ۸ خط دیجیتال است که می‌توان این خطوط را به صورت یک درگاه با ۸ خط، دو درگاه با ۴ خط و یا حتی ۸ درگاه با یک خط استفاده نمود. داده‌های درگاه به صورت رشته‌ای می‌باشند. در شکل ۴۱-۱۱ درگاههای ورودی و خروجی کارت MIO-16 نشان داده شده است. دقت کنید که در این حالت ۸ خط دیجیتال به صورت دو درگاه که هریک حاوی ۴ خط دیجیتال هستند به کار برده شده‌اند.




شکل ۴۱-۱۱: درگاههای ورودی و خروجی کارت MIO-16

۱۱

Port Width: عرض درگاه. تعداد خطوط به کاربرده شده در یک درگاه را گویند.

State: وضعیت. به یکی از دو حالت ممکن در مورد یک خط دیجیتال اطلاق می شود (1 یا on و 0 یا off).

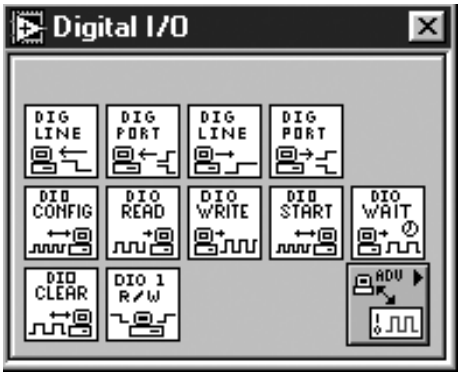
Pattern: این واژه در کل به معنای الگو بوده، در این بحث دنباله‌ای از حالات دیجیتالی است که اغلب به صورت یک عدد باینری بیان می‌شود و وضعیت هر یک از خطوط درگاه را نشان می دهد. به عنوان مثال در صورتی که الگوی یک درگاه با ۴ خط به صورت عدد باینری 1101 بیان شود بدین معنی است که ارزش خطوط اول، سوم و چهارم True یا 1 بوده، در حالی که خط دوم در وضعیت False یا 0 می باشد.در این حالت به ترتیب شماره گذاری بیت ها و خطوط توجه داشته باشید.

اولین یا کم ارزش‌ترین بیت[10] در منتهی‌الیه سمت راست الگو قرار دارد. در حالی که آخرین یا پرارزش ترین بیت[11] که در این مثال بیت چهارم است در منتهی الیه سمت چپ الگو قرار می گیرد. برای نمایش الگو می توان از مبنای دهدهی استفاده نمود و به عنوان مثال این الگو را به صورت عدد دهدهی 13 به نمایش در آورد.

شایان ذکر است که در تمامی کارت های شرکت NI از منطق مثبت TTL استفاده می شود. بدین معنی که در این کارت ها خطوط حامل سطوح ولتاژ پایین در حد 0-0.8 V معادل با وضعیت 0 یا False و خطوط حامل سطوح ولتاژ 2.2-5.5 V معادل با وضعیت 1 یا True در نظر گرفته می شوند.

## برنامه های ساده در مورد ورودی/خروجی دیجیتال

در مورد I/O دیجیتال ساده، برنامه های بسیاری در زیرپالت Functions >> Data Acquisition >> Digital I/O وجود دارد که استفاده از آنها نیز بسیار ساده است.



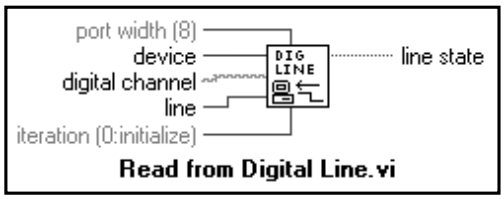

**شکل ۴۲-۱۱: زیرپالت Digital I/O**

در ادامه به بررسی هر یک از این برنامه ها می پردازیم:

برنامه‌ی Read from Digital Line.vi حالت منطقی یک خط دیجیتال را می‌خواند. در دو ورودی device و digital channel به ترتیب شماره‌ی کارت DAQ و شماره‌ی کانال حاوی خط دیجیتال مورد نظر تعیین می شود. ورودی line خط دیجیتالی را که باید وضعیت آن خوانده شود مشخص می کند. خروجی line state وضعیت

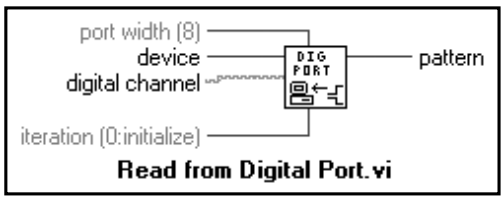
10- LSB: Least Significant Bit
11- MSB: Most Significant Bit

خط دیجیتال را به‌دست می دهد. وضعیت این خروجی به یکی از دو صورت high یا low می باشد.




شکل ۴۳-۱۱

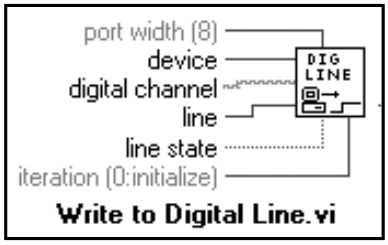
برنامه‌ی Read from Digital Port.vi وضعیت تمامی خطوط یک درگاه را می خواند.




شکل ۴۴-۱۱

در ورودی digital channel شماره‌ی درگاه دیجیتالی که باید قرائت شود مشخص می گردد. در خروجی pattern وضعیت خطوط دیجیتال به صورت یک عدد دهدهی نشان داده می شود. در صورتی که این عدد را به فرمت باینری تبدیل کنید به سادگی می توانید وضعیت هر یک از خطوط درگاه را مشاهده کنید.

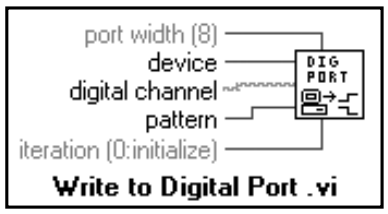
برنامه‌ی Write to Digital Line.vi یکی از دو حالت high یا low را در یکی از خطوط درگاه می نویسد.




شکل ۴۵-۱۱

در دو ورودی deveice و digital channel به ترتیب شماره‌ی کارت و شماره‌ی کانال حاوی خط دیجیتال مورد نظر تعیین می گردد. در ورودی line خطی که باید وضعیت آن در یکی از دو حالت high یا low قرار گیرد مشخص می شود. ورودی line state وضعیتی را که باید به خط مورد نظر اعمال شود مشخص می کند.

برنامه‌ی Write to Digital Port.vi یک الگوی دیجیتال یا به عبارت دیگر وضعیت دسته ای از خطوط را به یک درگاه اعمال می کند.




شکل ۴۶-۱۱

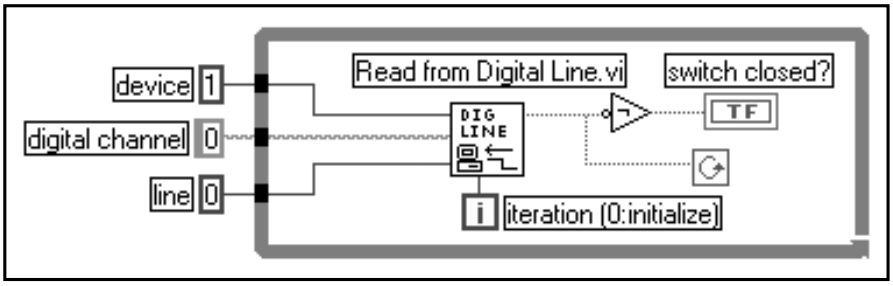
۱۱

ورودی digital channel شماره‌ی درگاهی را که اطلاعات مورد نظر باید بر روی آن نوشته شود مشخص می کند. ورودی pattern نیز به صورت عدد دهدهی بیان می‌شود و بیانگر الگویی است که باید به خطوط درگاه اعمال شود.

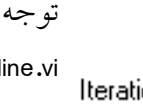
در تمامی برنامه های فوق از I/O های دیجیتال لچ نشده[۱۲] استفاده می شود. بدین مفهوم که به عنوان مثال به محض اجرا شدن برنامه‌ی Write to Digital Line.vi، خط مورد نظر در برنامه بلافاصله در یکی از دو وضعیت high یا low قرار می گیرد و تا زمانی که این مقدار توسط فراخوانی مجدد برنامه تغییر نکند همواره در همین وضعیت باقی می ماند. در اینجا لازم است در مورد ورودی iteration که در تمامی برنامه های فوق دیده می شود اندکی توضیح دهیم.

در هنگام فراخوانی این برنامه ها، جهت و نوع درگاههای دیجیتال به صورت پیش فرض و به طور مناسب تعریف می گردد. به طور معمول تعریف نمودن این پارامترها در مورد خواندن یا نوشتن داده ها تنها برای یک مرتبه کافی است. اگر یکی از این برنامه ها را به صورت تکراری استفاده کنید (به عنوان مثال آن را در یک حلقه قرار دهید) می توانید با سیم کشی هر عدد غیر صفر به ورودی iteration، از تعریف و پیکربندی درگاه به صورت مداوم خودداری کنید. مثال زیر نحوه‌ی به کارگیری این برنامه ها را روشن می‌سازد.

فرض کنید که یک سوییچ ولتاژ بالا به همراه یک رله‌ی دیجیتال در اختیار دارید. رله‌ی مذکور وضعیت باز یا بسته بودن کلید را نشان می دهد. خط دیجیتال که جهت نشان دادن وضعیت این رله استفاده شده است در حالت عادی، باز یا در وضعیت high می باشد. در صورت بسته شدن کلید، وضعیت خط دیجیتال به low[۱۳] تغییر می یابد. برای ایجاد برنامه ای که وضعیت باز یا بسته بودن این کلید را اطلاع دهد می توانید یک صفحه‌ی نمودار بلوکی مطابق شکل ۴۷-۱۱ ایجاد کنید. همان گونه که ملاحظه می کنید کلید مذکور به خط دیجیتال 0 از کانال 0 در کارت DAQ متصل شده است.




شکل ۴۷-۱۱




توجه کنید که در این برنامه ترمینال تکرار حلقه به ورودی iteration در برنامه‌ی Read from Digital line.vi سیم کشی شده است. در اولین تکرار حلقه، مقدار i برابر با 0 بوده و برنامه‌ی Read from Digital Line.vi به پیکربندی درگاه دیجیتال می پردازد. در حلقه های بعدی i>0 بوده و برنامه فقط مقدار خط را می خواند و به حلقه اجازه می دهد تا در زمان کمتری اجرا شود.

12- Nonlatched

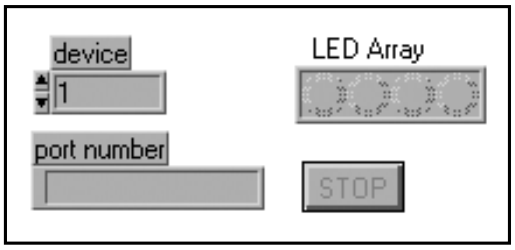
۱۳- در منطق منفی یا low، برای حالت باز و بسته کلید به ترتیب دو وضعیت low و high در نظر گرفته می شود و این بر خلاف تصورات ما است. زیرا معمولاً کلید بسته به مفهوم روشن بودن در ذهن ما نقش بسته است. این مفهوم در منطق مثبت یا high مورد استفاده قرار می گیرد. در بسیاری از رله ها و کنتاکتورها از منطق منفی استفاده می شود.

هنگامی که خط دیجیتال در وضعیت False قرار گیرد یا به عبارت دیگر کلید بسته شود، اجرای حلقه پایان می‌یابد و مقدار جبری ?switch closed به وضعیت on تغییر حالت می‌دهد. بدین معنی که اکنون کلید بسته شده است.

## تمرین ۹-۱۱: روشن کردن LEDها

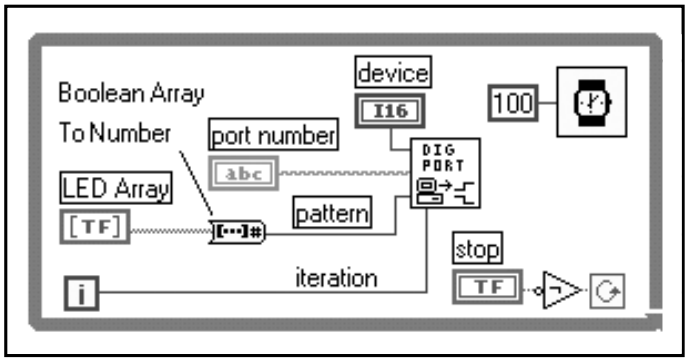
۱۱

برنامه‌ای بنویسید که در صفحه‌ی پانل آن ۴ عدد LED، یک المان Digital Control با برچسب device، یک المان String Control با برچسب port number و یک کلید STOP وجود داشته باشد. در دو المان device و port number به ترتیب شماره‌ی کارت DAQ و درگاه دیجیتال تعیین می‌گردد. با باز نمودن منوی کرکره‌ای بر روی این المان‌ها و انتخاب گزینه‌ی Change to Control، آنها را به المان‌های کنترلی تبدیل کنید. هدف از نوشتن این برنامه روشن کردن تعدادی LED است که هر یک از آنها به یکی از خطوط دیجیتال بر روی کارت DAQ متصل شده‌اند. با هر یک از LEDها یک مقاومت به طور سری قرار دهید. این برنامه باید به‌گونه‌ای نوشته شود که کاربر بتواند LEDهای واقعی را با کمک LEDهای مجازی موجود در صفحه‌ی پانل روشن و خاموش کند. می‌توانید به جای فرمان نوشتن بر روی هر یک از خطوط دیجیتال، برنامه‌ی Write to Digital Port.vi را در مورد کل درگاه اجرا کنید. این برنامه را با عنوان Digital Port.vi در زیرفهرست LabVIEW\My Activity ذخیره کنید. در شکل ۴۸-۱۱ صفحه‌ی پانل این برنامه را ملاحظه می‌کنید.




**شکل ۴۸-۱۱: صفحه‌ی پانل در برنامه‌ی Digital Port.vi**

ایجاد این برنامه با استفاده از دستور Boolean Array To Number (Functions >> Boolean) بسیار ساده‌تر خواهد شد. به کمک این دستور می‌توانید مقادیر باینری موجود در یک آرایه‌ی جبری را به عدد دهدهی تبدیل نمایید و سپس آن را به ورودی pattern سیم‌کشی کنید. پنجره‌ی نمودار بلوکی این برنامه در شکل ۴۹-۱۱ نشان داده شده است. به نحوه‌ی استفاده و همچنین عملکرد دستور Boolean Array To Number توجه کنید.



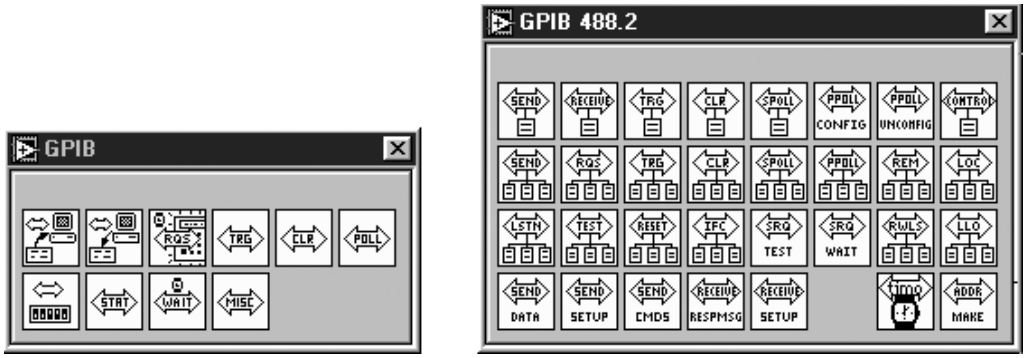

**شکل ۴۹-۱۱: صفحه‌ی نمودار بلوکی در برنامه‌ی Digital Port.vi**

۱۱

## برقراری ارتباط از طریق GPIB

گستردگی مطالب در مورد GPIB به اندازه‌ای است که برای بررسی کامل آن می توانیم یک فصل کامل یا حتی یک کتاب اختصاص دهیم. با این وجود سعی می کنیم تا از مطالب ابتدایی و پایه به سرعت بگذریم تا برای مطالعه‌ی جزییات این مبحث فرصت کافی داشته باشیم.

برای ارتباط از طریق GPIB دو زیرپالت در نظر گرفته شده است. همان گونه که در شکل ۵۰-۱۱ ملاحظه می‌کنید یکی از این دو زیرپالت، GPIB و زیرپالت دیگر که دستورهای بیشتری در آن قرار دارد، GPIB 488.2 نام دارد. این دو زیرپالت در زیرپالت Functions >> Instrument I/O قرار دارند.




**شکل ۵۰-۱۱: زیرپالت‌های GPIB 488.2 و GPIB**

GPIB 488.2 که گاهی IEEE 488.2 نامیده می شود، آخرین نسخه‌ی استاندارد ارتباطی انجمن مهندسین برق و الکترونیک است. استاندارد 488.2 در برگیرنده‌ی قواعدی در مورد ارتباطات می‌باشد و حاوی فرمانها و دستورهایی جهت یکنواخت کردن و هماهنگ سازی سیستم ها و دستگاههایی است که در آنها از باس GPIB استفاده شده است. دستورهای موجود در زیرپالت GPIB را می توان در اکثر کارت های GPIB به کار برد. اما در صورت نیاز به استانداردهای IEEE 488.2 حتماً باید از دستورهای موجود در زیرپالت GPIB 488.2 استفاده نمایید.

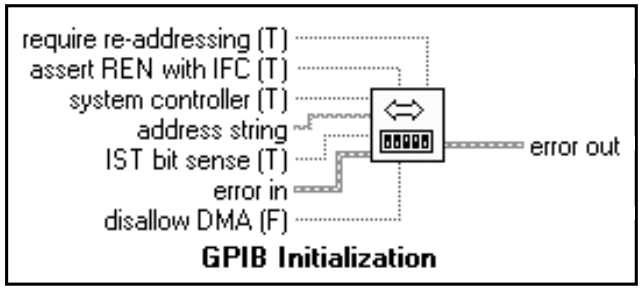
در برخی موارد می‌توان دستورهای GPIB را به جای دستورهای موجود در زیرپالت VISA به‌کار برد. دستورهای موجود در زیرپالت VISA از کاربرد و عمومیت بالاتری برخوردار هستند و به کمک آنها می‌توان با سایر دستگاهها و سیستم‌ها نظیر VXI نیز ارتباط برقرار ساخت.

به یاد داشته باشید که تمامی دستگاههای GPIB دارای یک آدرس منحصر به فرد هستند. همان گونه که قبلاً عنوان شد آدرس مذکور عددی بین 0 الی 30 است. این عدد هر یک از دستگاهها را از ابزارهای دیگر متمایز می‌سازد. بعضی از دستگاههای اندازه‌گیری در برخی موارد دارای آدرس ثانویه نیز می باشند. در تمامی برنامه های GPIB برای برقراری ارتباط با ابزار اندازه گیری حتماً باید آدرس آن را به فرمت رشته ای بیان کنید. این امر، به کارگیری یک برنامه را برای برقراری ارتباط با دستگاههای مختلفی

که همگی بر روی یک باس GPIB قرار دارند ساده تر می سازد.

مراحلی که در مورد ارتباط GPIB طی می شوند عبارتند از: آغاز نمودن ارتباط، ارسال فرمانها، دریافت پاسخ، تریگر کردن و قطع ارتباط. با به‌کارگیری مجموعه‌ای از برنامه هایی که در زیر آورده شده‌اند می توانید این اعمال را به راحتی انجام دهید. این دستورها در ردیف فرمانهای ساده‌ی GPIB هستند و همگی در زیرپالت GPIB قرار دارند.

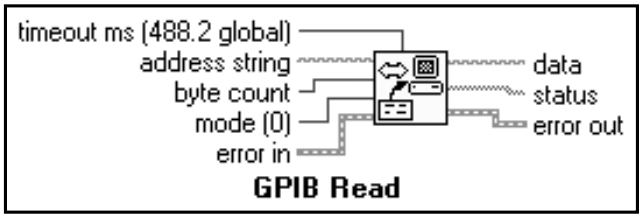
دستور GPIB Initialization یکی از کارت های GPIB را که آدرس آن در ورودی address string ظاهر شده است به کار می اندازد.




شکل ۵۱-۱۱

همان گونه که ملاحظه می کنید این آدرس به صورت رشته‌ای اعمال می شود، زیرا ممکن است حاوی کاراکترهای غیر عددی باشد.

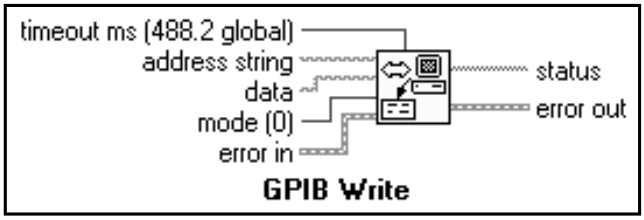
دستور GPIB Read داده ها را ازکارتی که آدرس آن در ورودی address string مشخص شده است می خواند.




شکل ۵۲-۱۱

در ورودی byte count تعداد بایت هایی که باید خوانده شوند تعیین می گردد. داده های قرائت شده در خروجی رشته ای data ظاهر می شوند.

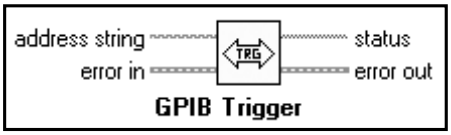
دستور GPIB Write داده های رشته ای در ورودی data را به یکی از کارت ها ارسال می کند.




شکل ۵۳-۱۱

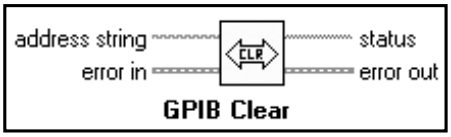
۱۱

دستور GPIB Trigger یک فرمان تریگر به کارت ارسال می‌کند. به عبارت دیگر این دستور کارت GPIB را تریگر می‌کند.



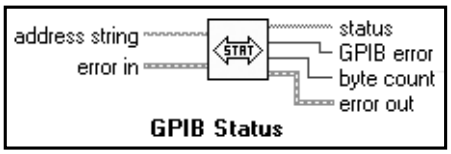

شکل ۱۱-۵۴

دستور GPIB Clear کارت را پاک یا در اصطلاح ری‌ست می‌کند.



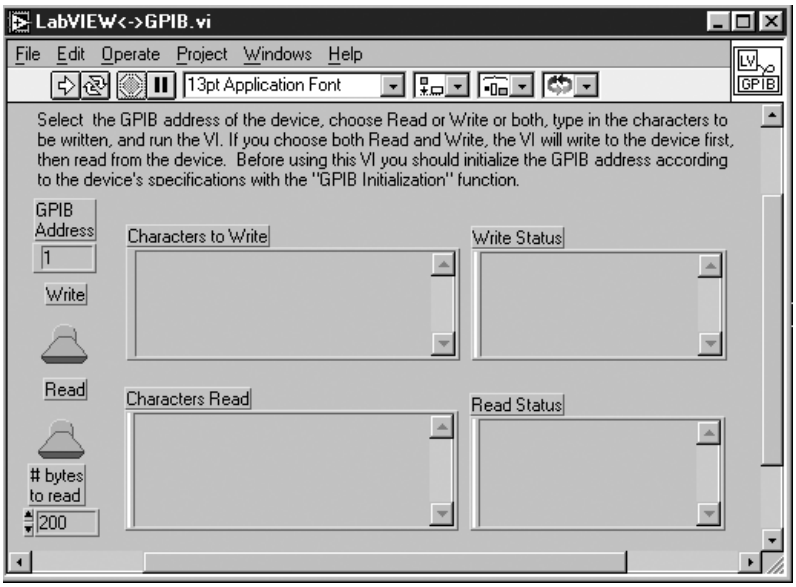

شکل ۱۱-۵۵

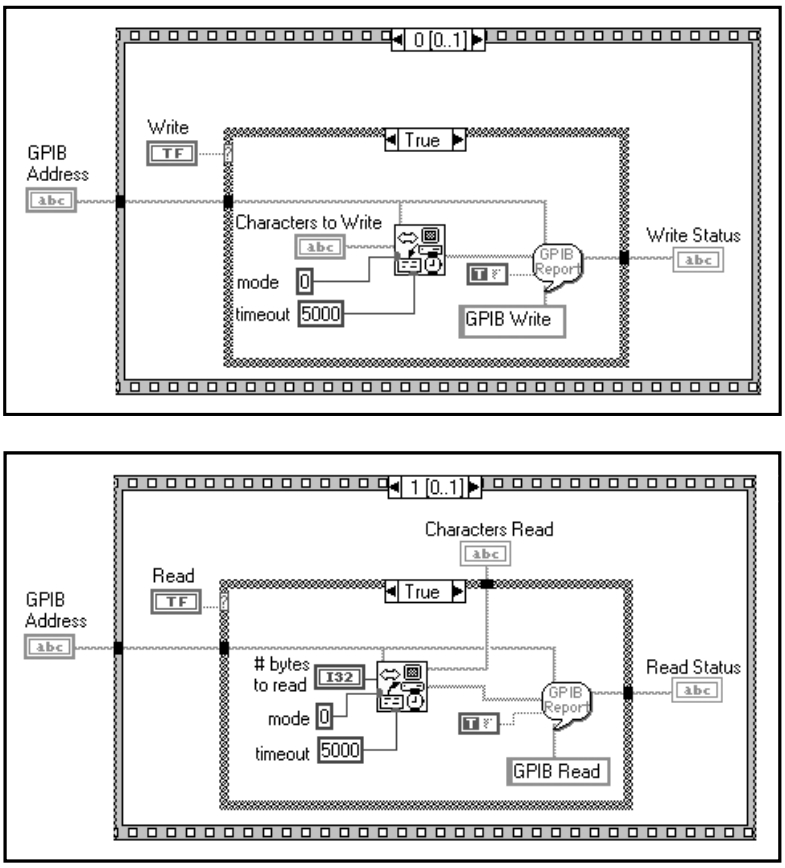
دستور GPIB Status وضعیت کنترل‌کننده‌ی GPIB را که با آدرس رشته‌ای address string مشخص شده است به همراه تعداد بایت‌های ارسال شده در عملیات GPIB قبلی به دست می‌دهد.




شکل ۱۱-۵۶

برای درک این مفاهیم و نحوه‌ی استفاده از دستورهای مذکور برنامه‌ی GPIB.vi <-> LabVIEW را از مسیر LabVIEW\Examples\Instr\smplgpib.llb اجرا کنید. در شکل ۱۱-۵۷ صفحات پانل و نمودار بلوکی این برنامه را ملاحظه می‌کنید.






**شکل ۵۷-۱۱: صفحات پانل و نمودار بلوکی در برنامه‌ی GPIB.vi <-> LabVIEW**

برنامه‌ی GPIB Error Report.vi که به عنوان یک زیربرنامه در صفحه‌ی نمودار بلوکی برنامه‌ی GPIB.vi <-> LabVIEW استفاده شده است، برنامه‌ای بسیار مفید می‌باشد که خطای احتمالی موجود در عملیات GPIB را گزارش می‌دهد. در صورت وجود هر گونه خطا یک پنجره‌ی محاوره‌ای باز می‌شود و خطاهای موجود را اعلام می‌کند. این برنامه نیز در همین مسیر قرار دارد.

در حال حاضر مجموعه‌ای از برنامه‌ها نوشته شده‌اند که به کمک آنها می‌توان با صدها دستگاه اندازه‌گیری مختلف ارتباط برقرار نمود. این برنامه‌ها به برنامه‌های راه‌انداز[۱۴] معروفند و اغلب آنها به صورت رایگان در اختیار شما قرار می‌گیرند.

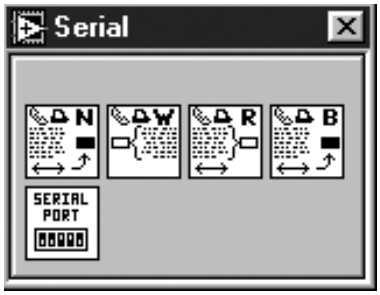
برای دستیابی به برنامه‌های راه‌انداز می‌توانید آنها را به همراه نسخه‌ی کامل LabVIEW از شرکت NI دریافت کنید یا آنها را از طریق شبکه‌ی اینترنت بر روی رایانه‌ی خود نصب نمایید. به‌کارگیری راه‌اندازها موجب صرفه‌جویی در زمان برنامه‌نویسی می‌گردد. زیرا این برنامه‌ها حاوی کدهای سطح پایین GPIB و فرمانهای لازم برای دستگاههای خاص هستند و تعریف‌کردن و نوشتن این کدها توسط کاربر الزامی نیست.

14- Instrument Driver

۱۱

## برقراری ارتباط از طریق درگاه سریال

برقراری ارتباط از طریق درگاه سریال که از این به بعد برای سهولت به اختصار آن را ارتباط سریال می‌نامیم، ساده‌ترین روش ایجاد ارتباط بین رایانه و دستگاههای اندازه‌گیری است. در شکل ۵۸-۱۱ برنامه‌های موجود در زیرپالت Serial را ملاحظه می‌کنید.




**شکل ۵۸-۱۱: زیرپالت Serial**

در صورت نادیده گرفتن استانداردهای سخت‌افزاری موجود در این روش ممکن است در برقرار نمودن ارتباط دچار مشکل شوید یا مجبور به برنامه‌نویسی و طراحی پروتکل‌های پیچیده باشید. در ضمن در این روش سرعت انتقال داده‌ها بسیار پایین است. اکثر مشکلاتی که کاربران در برقرار نمودن ارتباط در این روش با آن دست به گریبان هستند به ندرت به LabVIEW مربوط است.

در این بخش نگاهی کوتاه به نحوه‌ی استفاده از برنامه‌های ارتباط سریال داشته و شما را در برقراری ارتباط از طریق درگاه سریال یاری خواهیم داد. اگر قبلاً از درگاههای سریال استفاده نکرده‌اید لازم است با مفاهیم استفاده شده در این روش آشنا شوید. اگر تاکنون از مودم استفاده نموده‌اید و با اصطلاحاتی نظیر سرعت انتقال، بیت‌های توقف و بیت توازن آشنا هستید اطلاعات شما در این زمینه کافی است و می‌توانید این بحث را آغاز کنید. در غیر این صورت بهتر است ابتدا مطالبی در مورد ارتباط سریال مطالعه نمایید.

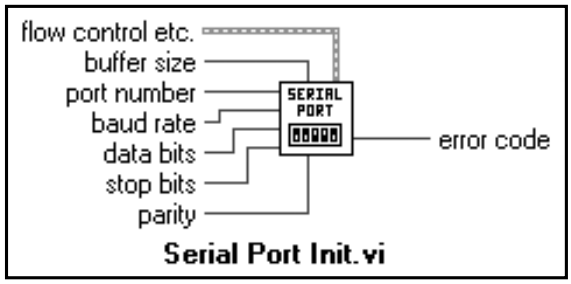
در نرم‌افزار LabVIEW می‌توان هر یک از درگاههای رایانه را جهت برقراری ارتباط سریال به کار برد. در هر یک از سیستم‌های عامل واژه‌ی port number بسته به نوع سیستم مفهومی متفاوت دارد. در جدول ۲-۱۱ شماره درگاهها و کاربرد آنها را در چند سیستم عامل ملاحظه می‌کنید.

**جدول ۲-۱۱: شماره‌ی درگاههای مختلف رایانه در چند سیستم عامل**

| Windows | Mac | Unix | |
|---|---|---|---|
| | | Solaris 1 | Solaris 2 |
| 0: COM1<br>1: COM2<br>2: COM3<br>…<br>8: COM9<br>9: LPT1<br>10: LPT2, etc. | 0: modem port<br>1: printer port | 0: /dev/ttya<br>1: dev/ttyb, etc. | 0: /dev/cua/a<br>1: /dev/cua/b, etc. |

در ادامه به بررسی هر یک از برنامه های موجود در زیرپالت Serial می پردازیم:

برنامه‌ی Serial Port Init.vi درگاه سریال مطلوب را طبق پارامتر های تنظیم شده توسط کاربر به کار می اندازد.

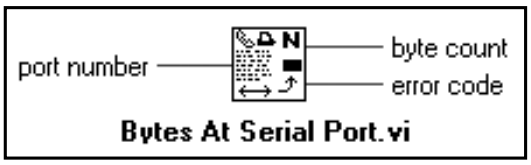



شکل ۵۹-۱۱

در اغلب موارد می توانید از سیم کشی به ورودیهای این برنامه صرف نظر کنید و تنها از مقادیر پیش فرض استفاده نمایید. این مقادیر پیش فرض عبارتند از:

**port number :** 0
**baud rate :** 9600
**data bits :** 8
**stop bits :** 1
**parity :** none

برنامه‌ی Bytes At Serial Port.vi تعداد بایت های موجود در بافر ورودی درگاه سریال را که شماره‌ی آن در port number تعیین شده است در خروجی byte count ظاهر می سازد.



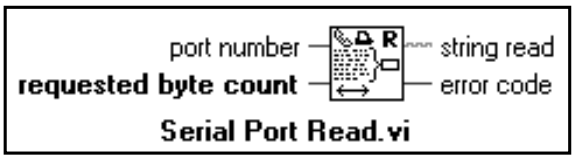

شکل ۶۰-۱۱

برنامه‌ی Serial Port Write.vi داده هایی را که به صورت رشته ای در ورودی string to write تعیین شده اند در درگاه مورد نظر می نویسد.

شکل ۶۱-۱۱

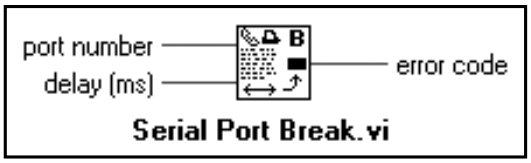
برنامه‌ی Serial Port Read.vi تعداد کاراکتر های تعیین شده در ورودی requested byte count را از درگاهی که توسط ورودی port number مشخص شده است می خواند.

۱۱



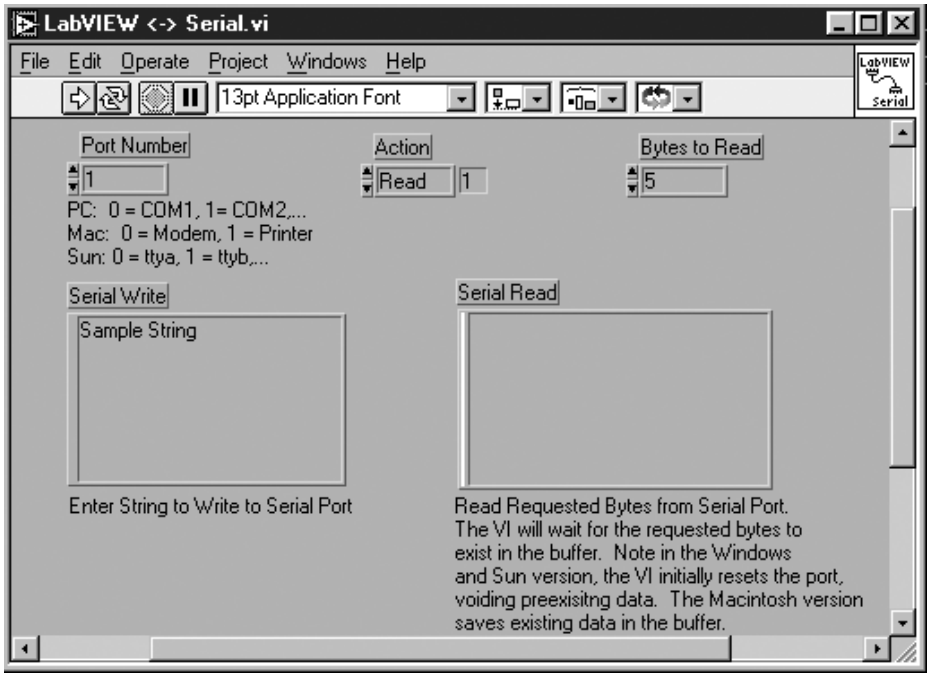

شکل ۶۲-۱۱

برنامه‌ی Serial Port Break.vi تأخیری با حداقل زمان تعیین شده در ورودی deley (ms) را به درگاه مشخص شده در ورودی port number اعمال می کند.



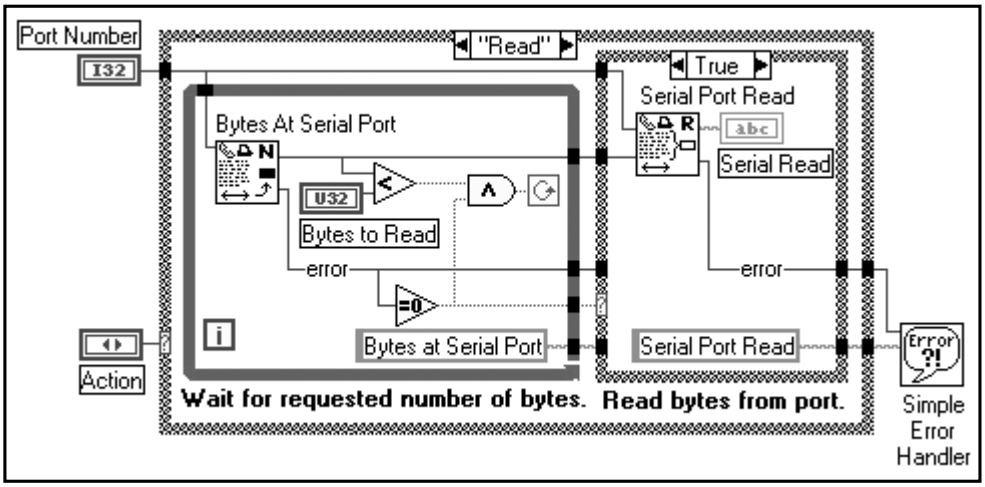

شکل ۶۳-۱۱

هر بایت سریال معادل با یک کاراکتر ASCII است. در اکثر درگاههای ارتباطات سریال از این قاعده پیروی می شود. مطلبی که باید در رابطه با رایانه‌ی خود بدانید آن است که در رایانه، یک بافر مخصوص درگاه سریال تعبیه شده است و احتمالاً دستگاهی که از طریق آن نیز به ارتباط سریال می پردازید نیز حاوی این بافر می باشد. بافر FIFO کاراکترها را به محض ورود در خط سریال ذخیره می کند. برای درک مفاهیم و برنامه های ارتباط سریال، برنامه‌ی LabVIEW <-> Serial Port.vi را از مسیر LabVIEW\ Examples\ Instr\ smplserl.llb اجرا کنید.



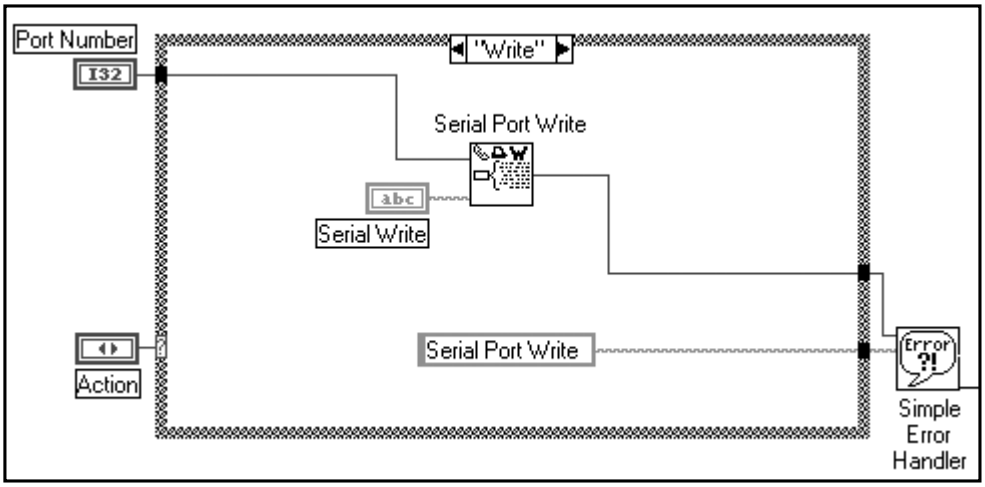

شکل ۶۴-۱۱ : صفحه‌ی پانل در برنامه‌ی LabVIEW <-> Serial Port.vi

توجه داشته باشید که این برنامه در هر لحظه قادر است تنها یکی از دو عمل خواندن از درگاه سریال یا نوشتن بر روی آن را انجام دهد. برای خواندن داده‌ها، این برنامه منتظر می‌ماند تا شماره‌ی بایت‌های مورد انتظار در بافر درگاه سریال را پیدا کند. انتظار کشیدن برای دریافت تمامی بایت‌ها کاملاً معقول است. زیرا بدین ترتیب از دریافت تمامی داده‌ها از دستگاهی که ارتباط سریال با آن برقرار می‌شود اطمینان حاصل می‌گردد. در غیر این صورت به نتایجی نادرست دست خواهید یافت. در شکل ۶۵-۱۱ صفحه‌ی نمودار بلوکی این برنامه را در حالت خواندن داده‌ها ملاحظه می‌کنید.




**شکل ۶۵-۱۱: فرآیند خواندن داده‌ها در صفحه‌ی نمودار بلوکی برنامه‌ی LabVIEW <-> Serial Port.vi**

فرآیند نوشتن داده‌ها در این برنامه ساده‌تر است. این عمل را می‌توان با ارسال داده‌ی رشته‌ای از درگاه سریال انجام داد. صفحه‌ی نمودار بلوکی این برنامه در حالت نوشتن داده‌ها در شکل ۶۶-۱۱ نشان داده شده است.




**شکل ۶۶-۱۱: فرآیند نوشتن داده‌ها در صفحه‌ی نمودار بلوکی برنامه‌ی LabVIEW <-> Serial Port.vi**

در ادامه، چند نکته در مورد نوشتن برنامه‌های ارتباط سریال آورده شده است. این نکات می‌تواند راهگشای شما در نوشتن برنامه یا رفع اشکالات احتمالی در برقراری ارتباط باشد.

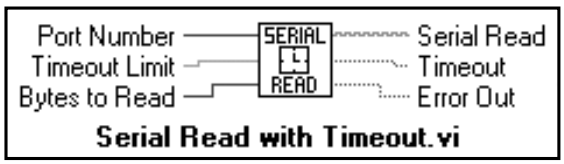
۱۱

۱- از هماهنگی ساختارها و پارامترهای تعریف شده در برنامه‌ی Serial Port Init.vi و درگاه سریال بر روی دستگاهی که قصد ارتباط با آن را دارید اطمینان حاصل کنید. (پارامترهایی نظیر سرعت انتقال داده ها، handshaking و...) اگر حتی یکی از این پارامترها با دیگری مطابقت نداشته باشد برقراری ارتباط امکان‌پذیر نخواهد بود.

۲- در بسیاری از دستگاههای سریال باید یک کاراکتر carriage return یا line feed پس از هر فرمان رشته‌ای[15] وارد شود. LabVIEWبه صورت خودکار این کاراکترها را به رشته اضافه نمی‌کند. برای افزودن این کاراکترها به فرمان رشته‌ای، منوی کرکره‌ای را بر روی المان رشته‌ای باز کنید و گزینه‌ی Codes Display '\' را انتخاب نمایید. سپس کاراکتر «\r» یا «\n» را تایپ کنید.

۳- برنامه‌ی Bytes At Serial Port.vi یکی از برنامه های کاربردی و مفید در برقراری ارتباط سریال است. فرض کنید که از تعداد کاراکترهای مورد انتظار در یک دستگاه سریال اطلاعاتی در اختیار ندارید. با استفاده از این برنامه می توانید به تعداد کاراکترهای مذکور دست یابید و بدین ترتیب تعداد دقیق کاراکترها را بخوانید. خروجی این برنامه را می توانید مستقیماً به ورودی requested byte count در برنامه‌ی Serial Port Read.vi سیم کشی کنید.

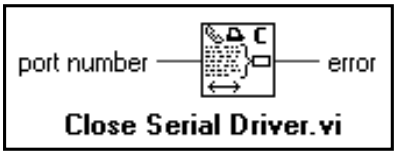
۴- یک برنامه‌ی کاربردی دیگر با عنوان Serial Read with Timeout.vi وجود دارد. این برنامه در مسیر LabVIEW\Examples\Instr\smplserl.llb قرار گرفته است. برنامه‌ی مذکور نظیر برنامه‌ی Serial Port Read.vi عمل می‌کند با این تفاوت که در این برنامه در صورت عدم دریافت پاسخ از درگاه سریال دستگاه اندازه‌گیری، می توانید خطای timeout تعیین کنید. این برنامه در مواردی که برنامه‌ی شما توقف[16] نموده و منتظر دریافت پاسخ از درگاه سریال می باشد بسیار مفید و کاربردی است.




**شکل ۶۷-۱۱: آیکن برنامه‌ی Serial Read with Timeout.vi به همراه اتصالات آن**

۵- پس از فراخوانی هر یک از برنامه های ارتباط سریال توسط LabVIEW، به طور معمول تا زمانی‌که از محیط نرم‌افزاری LabVIEW خارج نشده‌اید درگاه سریال استفاده شده آزاد نمی شود. این مطلب بدین مفهوم است که تا زمانی که درگاه سریال را نبندید، هیچ دستگاه دیگری نظیر مودم نمی تواند بر روی درگاه مذکور قرار گیرد. به دلائلی که در اینجا از ذکر آن خودداری می کنیم، برنامه‌ی Serial Port Close.vi در زیرپالت Serial گنجانده نشده است. می توانید برنامه‌ی Close Serial Driver.vi که عملکردی مشابه با برنامه‌ی مذکور دارد از مسیر LabVIEW\Vi.lib\Instr\sersup.llb مورد استفاده قرار دهید. اگر این برنامه را مکرراً مورد استفاده قرار می دهید به شما توصیه می کنیم آن را در زیرپالت Serial قرار دهید.

15- String Command
16- Hang




شکل ۶۸-۱۱: آیکن برنامه‌ی Close Serial Driver.vi به همراه اتصالات آن

۱۱

## خلاصه

در این فصل روشهای جمع آوری اطلاعات و نحوه‌ی برقراری ارتباط با دنیای خارج از رایانه به کمک کارت های DAQ، GPIB و درگاههای سریال مورد بررسی قرار گرفت. در بخشهایی از این فصل نحوه‌ی ذخیره‌ی داده ها در بافر و مفهوم تریگر کردن و انواع آن را بررسی نمودیم. در این فصل، روش ذخیره نمودن داده ها را به صورت موقت در بخشی از حافظه که اصطلاحاً به آن بافر گوییم نیز مورد مطالعه قرار دادیم. در مواردی که نیاز به دقت و سرعت بالا در نمونه برداری باشد از این روش استفاده می شود. تریگر کردن نیز روشی برای آغاز نمودن یا پایان دادن به فرآیند DAQ است. سیگنال تریگرکننده ممکن است نرم افزاری یا سخت افزاری باشد.

در زیرپالت های Analog Input و Analog Output برنامه هایی برای I/O های آنالوگ تعبیه شده است که از نظر پیچیدگی در سطوح متفاوتی قرار دارند. استفاده از برنامه های ساده در مورد ورودی/خروجی های ساده بسیار مفید است. برنامه های سطح متوسط کنترل بیشتری بر فرآیند DAQ اعمال می کنند. با قرار دادن این برنامه ها در یک ترتیب خاص می توانید ابتدا پارامترهای فرآیند DAQ را تعریف نموده، سپس عملیات DAQ را آغاز کنید و داده ها را بخوانید. پس از اتمام فرآیند نیز بافر و تمامی مواردی که در این عملیات مورد استفاده قرار گرفته اند را ری‌ست کنید. در برنامه های سطح متوسط می توان از I/O همراه بافر استفاده نمود. همچنین می توانید داده ها را به سمت یک فایل به جریان در آورید. استفاده از برنامه های سطح پیشرفته به طور کلی برای مبتدیان مشکل است. بنابراین از ذکر آنها در این کتاب خودداری کرده ایم.

زیرپالت Digital I/O شامل برنامه هایی است که به کمک آنها می توان داده ها را از خطوط دیجیتالی کارت DAQ خواند یا بر روی آنها اطلاعاتی ثبت نمود. عملیات خواندن و نوشتن را می توان بر روی هر یک از خطوط یا بر روی کل درگاه (بسته به نوع کارت در ۴ یا ۸ خط دیجیتال) انجام داد.

برنامه های لازم جهت برقراری ارتباط با دستگاههایی که در آنها از باس GPIB استفاده شده است در دو زیرپالت GPIB و GPIB 488.2 قرار دارند. در زیرپالت Serial نیز برنامه هایی جهت برقراری ارتباط سریال تعبیه شده است.

LabVIEW

# ۱۲

# دستورها و ساختارهای پیشرفته

## در این فصل با موارد زیر آشنا می‌شوید:

- مفهوم متغیرهای محلی و سراسری و نحوه‌ی استفاده از آنها
- روش سفارشی کردن شکل ظاهری و عملکرد المان‌های موجود در صفحه‌ی پانل با استفاده ازگره‌های خصوصیت
- دستورهای پیشرفته در نرم‌افزار LabVIEW
- عملکرد نرم‌افزار LabVIEWدر مورد فراخوانی کدهای خارجی و برقراری ارتباط با زبانهای برنامه‌نویسی متنی نظیر C
- نحوه‌ی تبدیل داده‌ها از نوعی به نوع دیگر

# LabVIEW

# ۱۲

## دستورها و ساختارهای پیشرفته

در این فصل روش به‌کارگیری برخی از دستورها و ساختارهای پیشرفته و قدرتمند LabVIEW را مورد بررسی قرار می‌دهیم و به تعریف دو متغیر محلی و سراسری می‌پردازیم. این دو متغیر در زبانهای برنامه‌نویسی متداول نیز به چشم می‌خورند. در این فصل خواهید دید که چگونه می‌توان با استفاده از گره‌های خصوصیت، قابلیت انعطاف و کارایی بیشتری به المان‌های کنترل و نشان‌دهنده بخشید. گره‌های مذکور شکل ظاهری و عملکرد المان‌های موجود در صفحه‌ی پانل را تغییر می‌دهند. علاوه بر این به بحث و بررسی در مورد توابع و دستورهای پیشرفته نظیر فراخوانی کدهای خارجی نوشته شده در زبانهای برنامه‌نویسی دیگر و پنجره‌های محاوره‌ای می‌پردازیم. در پایان نیز با نگاهی کوتاه به تبدیلات پیشرفته‌ی انواع داده‌ها دلیل استفاده از آنها را درک خواهید نمود.

در این فصل اهداف زیر دنبال می‌شوند:

- درک مفهوم متغیرهای محلی و سراسری و نحوه‌ی استفاده از آنها.
- فراگیری روش سفارشی کردن شکل ظاهری و عملکرد المان‌های موجود در صفحه‌ی پانل با استفاده از گره‌های خصوصیت.
- آشنایی با برخی از دستورهای پیشرفته در نرم‌افزار LabVIEW.
- بررسی عملکرد و قدرت نرم‌افزار LabVIEW در مورد فراخوانی کدهای خارجی و برقراری ارتباط با زبانهای برنامه‌نویسی متنی نظیر C.
- تبدیل داده‌ها از نوعی به نوع دیگر.

اصطلاحات کلیدی موجود در این فصل عبارتند از:

- متغیر محلی (Local Variable)
- متغیر سراسری (Global Variable)
- مد خواندن و نوشتن (Read and Write Mode)
- حالت رقابتی (Race Condition)
- گره‌ی خصوصیت (Attribute Node)
- رویداد (Occurrence)
- CIN (Call Interface Node)
- رشته‌ی متنی (ASCII String)
- رشته‌ی باینری (Binary String)

۱۲

## متغیرهای محلی و سراسری

متغیرهای محلی و سراسری دو ساختار پیشرفته در بسته‌ی نرم‌افزاری LabVIEW می‌باشند. اگر تاکنون در یکی از محیطهای برنامه‌نویسی رایج نظیر C یا Pascal برنامه‌نویسی کرده باشید، مسلماً با این مفاهیم آشنایی کامل دارید. تا اینجا ملاحظه نمودید که اعمالی نظیر خواندن از /نوشتن به المان‌های موجود در صفحه‌ی پانل از طریق ترمینال‌های متناظر با آنها در صفحه‌ی نمودار بلوکی انجام می‌گرفت. همان گونه که می‌دانید هر المان در صفحه‌ی پانل تنها یک ترمینال بر روی صفحه‌ی نمودار بلوکی دارد. حال فرض کنید که در یک برنامه لازم است تا مقدار یک المان را از چند نقطه‌ی مختلف در صفحه‌ی نمودار بلوکی و یا در چند برنامه‌ی مجزا بخوانید و یا مقدار آن را به روز برسانید. در این موارد چه باید کرد؟

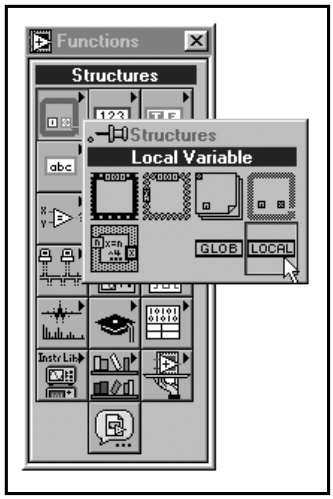
متغیر محلی دستیابی به المان‌های موجود در صفحه‌ی پانل را از چند نقطه‌ی مختلف در صفحه‌ی نمودار بلوکی در یک برنامه امکان‌پذیر می‌سازد. به کارگیری این متغیرها در زمانی کاملاً مفید و حیاتی است که نمی‌خواهید یا به دلایلی نمی‌توانید به ترمینال متناظر با المان مورد نظر در صفحه‌ی نمودار بلوکی سیم‌کشی کنید.

با استفاده از متغیر سراسری، دستیابی به مقادیر هر نوع داده یا در صورت نیاز، چند نوع داده در یک زمان بین چند برنامه امکان‌پذیر می‌گردد. این متغیرها در مواردی استفاده می‌شوند که سیم‌کشی به گره‌های زیربرنامه امکان‌ناپذیر بوده، یا اینکه چند برنامه به صورت همزمان در حال اجرا باشند.

می‌توان گفت که دو متغیر محلی و سراسری در بسیاری موارد به طور مشابه عمل می‌کنند. با این تفاوت که استفاده از متغیرهای سراسری به یک برنامه محدود نیستند و این متغیرها می‌توانند مقادیر را بین چند برنامه انتقال دهند. در بخش بعدی به بحث درباره‌ی مزایای استفاده از متغیرهای محلی و سراسری می‌پردازیم.

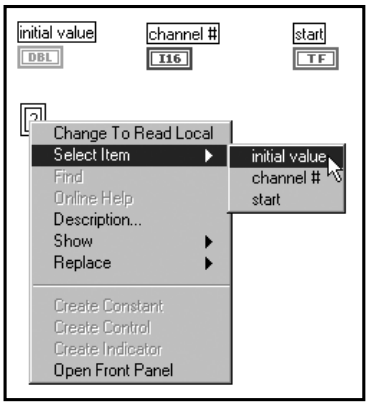
## متغیرهای محلی

متغیرهای محلی از المان های پیشرفته در بسته‌ی نرم‌افزاری LabVIEW هستند و دسترسی به آنها از طریق زیرپالت Structures << Functions امکان‌پذیر است.




**شکل ۱-۱۲: نحوه‌ی ایجاد متغیر محلی با استفاده از زیرپالت Structures**

هنگامی که یک المان را به عنوان متغیر محلی انتخاب می کنید ابتدا یک گره با علامت «?» ظاهر می گردد. علامت مذکور بیانگر آن است که این متغیر محلی هنوز تعریف نشده است. پس از کلیلک کردن بر روی این گره به کمک ابزار Operating Tool، لیستی از المان های کنترل و نشان دهنده‌ی برچسب دار در پنجره‌ی فعال ظاهر می شود. برای تعریف نمودن متغیر محلی کافی است یکی از المان های موجود در لیست مذکور را انتخاب کنید. روش دیگر جهت دستیابی به لیست مذکور، باز نمودن منوی کرکره‌ای بر روی متغیر محلی و سپس انتخاب گزینه‌ی Select Item از این منو است. روش دیگر برای ایجاد یک متغیر محلی، باز نمودن منوی کرکره‌ای بر روی ترمینال متناظر با المان مورد نظر و انتخاب گزینه‌ی Local Variable << Create می باشد. در شکل ۲-۱۲ نحوه‌ی ایجاد متغیر محلی را در این روش ملاحظه می کنید.



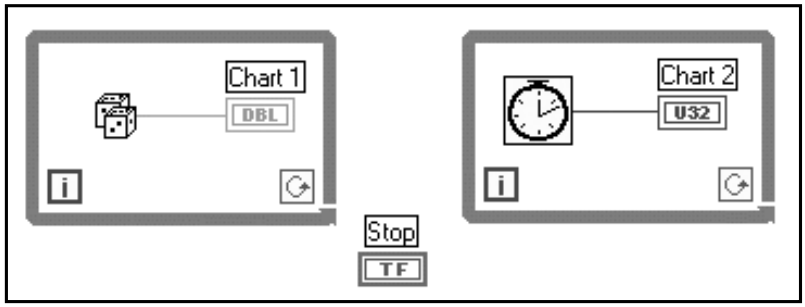

**شکل ۲-۱۲: نحوه‌ی ایجاد متغیر محلی با استفاده از منوی کرکره‌ای**

حداقل دو دلیل عمده برای توجیه به کارگیری متغیرهای محلی در برنامه وجود دارد:

- می توانید اعمالی نظیر کنترل حلقه های موازی را به کمک متغیر محلی انجام دهید. در صورت عدم استفاده از متغیر محلی این عمل امکان پذیر نیست.
- به صورت مجازی می توان هر یک از المان های کنترل و نشان دهنده را به جای یکدیگر استفاده نمود.

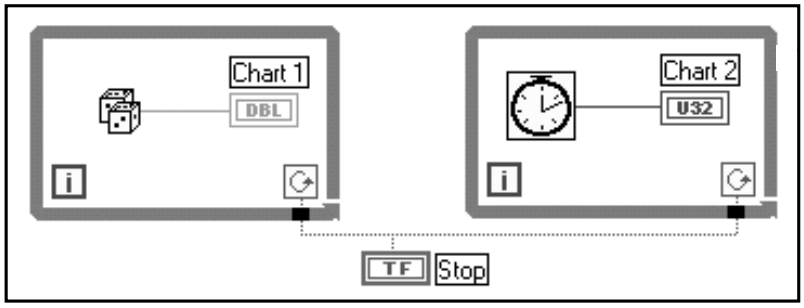
## کنترل حلقه های موازی

در صورتی که به یاد داشته باشید در فصول ابتدایی این مجموعه عنوان شد که نرم افزار LabVIEW چگونه ترتیب اجرای برنامه را از طریق اصل جریانِ داده کنترل می کند. در حقیقت اصل جریانِ داده جزء جدا نشدنی نرم افزار LabVIEW است. به کمک این اصل، برنامه نویسی در محیط LabVIEW بسیار ساده می گردد. با این وجود در برخی موارد لازم است تا عملیات خواندن و نوشتن داده ها را در مورد المان های کنترل و نشان دهنده‌ی موجود در صفحه‌ی پانل بدون سیم کشی مستقیم به ترمینال متناظر آنها در صفحه‌ی نمودار بلوکی انجام دهید. به عنوان مثال فرض کنید که قصد داریم توسط یک المان جبری Stop، اجرای دو حلقه‌ی مستقل While را متوقف سازیم. در شکل ۳-۱۲ این مسأله نشان داده شده است.




شکل ۳-۱۲

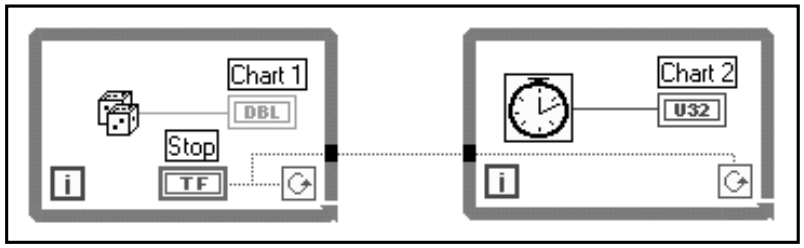
چگونه می توان این عمل را انجام داد؟ ممکن است پاسخ دهید که: «دکمه‌ی Stop را به ترمینال های دو حلقه سیم کشی می کنیم.» دقیقاً فکر کنید در صورتی که دکمه‌ی Stop از بیرون حلقه به ترمینال های داخلی آن سیم کشی شده باشد، مقدار این دکمه چند مرتبه بررسی می شود. اتصال دکمه‌ی Stop از خارج حلقه به ترمینال های داخلی آن در شکل ۴-۱۲ نشان داده شده است.




شکل ۴-۱۲

به دلیل اینکه المان های کنترل در خارج از حلقه تنها یک بار قبل از به اجرا در آمدن حلقه خوانده می شوند، ارتباط دادن دکمه‌ی Stop از بیرون حلقه ها به دو ترمینال شرطی داخل حلقه کاری از پیش نمی برد. در صورتی که دکمه‌ی Stop در وضعیت False باشد این حلقه تنها یک مرتبه به اجرا در می آید و اگر دکمه‌ی Stop در وضعیت True باشد این حلقه به طور متناوب اجرا می گردد.

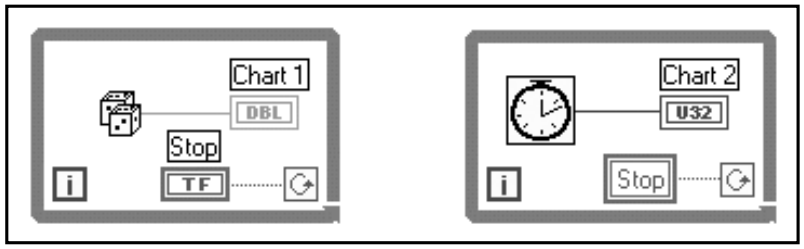
حال ممکن است پاسخ دهید: «می توانیم دکمه‌ی Stop را در داخل یکی از حلقه ها قرار دهیم و سپس آن را به ترمینال های شرطی هر دو حلقه سیم کشی کنیم». آیا این طرح به درستی عمل می کند؟ این مطلب در شکل ۵-۱۲ نشان داده شده است.




شکل ۵-۱۲

قرار دادن دکمه‌ی Stop در داخل یکی از حلقه ها و سپس برقراری ارتباط بین ترمینال شرطی حلقه‌ی دیگر و کلید مذکور نیز به همان دلیل گفته شده نیاز ما را برآورده نمی سازد. زیرا کلید Stop همواره در خارج یکی از حلقه ها قرار گرفته است و مجدداً مشکل قبلی مطرح می شود. در این حالت اجرای حلقه‌ی دوم پس از اتمام اجرای حلقه‌ای که دکمه‌ی Stop در داخل آن قرار گرفته است آغاز می شود. آیا مبحث وابستگی اطلاعاتی[1] را به خاطر دارید؟

پاسخ به این معما همان گونه که احتمالاً حدس زده اید به کارگیری متغیر محلی است. متغیر های محلی در حقیقت یک رونوشت از داده های ترمینال های دیگر در صفحه‌ی نمودار بلوکی ایجاد می کنند. این متغیرها همواره حاوی داده های به روز رسانده شده از المان های متناظر خود در صفحه‌ی پانل هستند. با این روش می توانید بدون انجام سیم کشی به ترمینال های متناظر با المان های کنترل یا نشان دهنده، از چند نقطه در صفحه‌ی نمودار بلوکی به آنها دسترسی پیدا کنید. در شکل ۶-۱۲ نحوه‌ی استفاده از متغیر محلی و چگونگی نمایش آن در صفحه‌ی نمودار بلوکی نشان داده شده است.




شکل ۶-۱۲

1- Data Dependency

مطابق شکل ۶-۱۲ می توان کلید Stop را از داخل حلقه به یکی از ترمینال های شرطی سیم کشی نمود و متغیر محلی متناظر با این کلید را در حلقه‌ی دوم به ترمینال شرطی دیگر ارتباط داد.

در مورد ایجاد متغیرهای محلی جبری یک حالت استثناء وجود دارد. المان‌های جبری موجود در صفحه‌ی پانل نباید از طریق گزینه‌های Mechanical Action به وضعیت latch برده شوند. اگرچه در ابتدای امر، این حالت قابل رؤیت نیست، اما به‌کارگیری یک المان جبری در مد latch همراه با ایجاد یک متغیر محلی در مورد آن المان در مد Read یک وضعیت مبهم ایجاد می‌کند. بنابراین در صورت استفاده از متغیر محلی برای یک المان جبری در مد latch، دکمه‌ی Run به صورت پیکان شکسته در می‌آید و بیانگر وجود خطا در برنامه است.



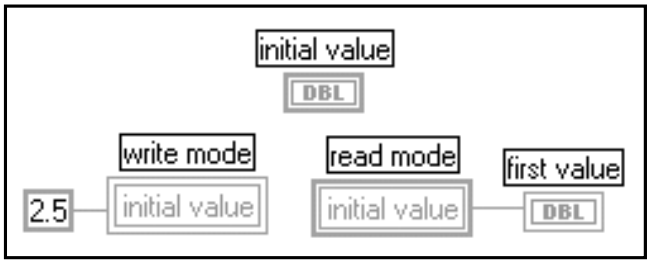
یکی از ویژگیهای قابل توجه و بسیار جالب در مورد متغیرهای محلی آن است که به شما اجازه می دهند تا داده‌ای را بر روی یک المان کنترل بنویسید یا داده های یک نشان دهنده را بخوانید. همان گونه که می دانید انجام چنین عملی درمورد ترمینال های معمول و متداول یک المان امکان‌پذیر نیست. متغیرهای محلی دو مد خواندن و نوشتن دارند. هر متغیر محلی در هر لحظه تنها می تواند در یکی از مدهای مذکور قرار گیرد. اما ایجاد ترمینال محلی دوم در مد دیگر امکان‌پذیر است. در مد خواندن می توانید مقدار ترمینال محلی را به همان شیوه‌ی قرائت یک المان کنترل بخوانید. در مد نوشتن نیز می توانید داده‌ها را به همان روش به روز رساندن المان نشان‌دهنده بر روی ترمینال محلی بنویسید. قراردادهای زیر را جهت سیم‌کشی متغیرهای محلی به خاطر بسپارید:

مد خواندن = المان کنترل

مد نوشتن = المان نشان دهنده

روش دیگر برای به خاطر سپردن مفهوم فوق، در نظر گرفتن یک متغیر محلی در مد خواندن به عنوان یک منبع اطلاعاتی است. حال آن که متغیر محلی در مد نوشتن را می توان به عنوان یک مقصد اطلاعاتی فرض نمود.

برای تعیین مد متغیر محلی، منوی کرکره‌ای را بر روی ترمینال متغیر محلی باز نموده، سپس گزینه‌ی ...Change to را انتخاب کنید. همان گونه که در مورد تفاوت شکل ظاهری بین دو ترمینال المان های کنترل و نشان دهنده عنوان شد، ضخامت حاشیه‌ی ترمینال متغیر محلی در حالت خواندن پهن تر از حالت نوشتن است. در شکل ۷-۱۲ این مورد به وضوح نشان داده شده است.




شکل ۷-۱۲

برای حصول اطمینان از صحت انجام سیم کشی در مد صحیح، در هنگام برقراری ارتباط بین ترمینال های متغیر های محلی، به ضخامت حاشیه‌ی ترمینال های مربوط توجه کنید. اگر به عنوان مثال سعی کنید تا در مد خواندن، عملیات نوشتن را انجام دهید، سیم ارتباط دهنده به صورت منقطع در می‌آید و نشانگر وجود خطا در انجام سیم کشی است. نکته‌ی آخر این که برای تعریف متغیر محلی در مورد المان های کنترل یا نشان دهنده حتماً باید ابتدا آنها را برچسب گذاری نمایید. در غیر این صورت تعریف متغیر محلی برای آنها امکان‌پذیر نیست.

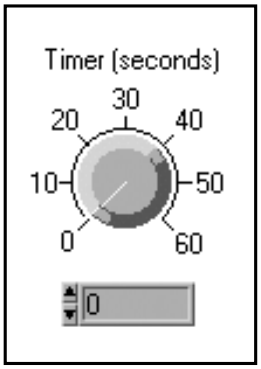
## تمرین ۱-۱۲: شبیه‌سازی عملکرد تایمر

۱۲

در این تمرین قصد داریم عملکرد یک تایمر را توسط یک دکمه‌ی چرخشی شبیه سازی کنیم، به صورتی که بتوانیم زمان تایمر را توسط این دکمه تنظیم نماییم و چرخش دکمه را نیز مشاهده کنیم. نظیر این تایمرها را حتماً در اجاق گاز آشپزخانه دیده‌اید.

نتیجه‌ای که از توضیحات فوق به دست می آید آن است که به دلیل اینکه کاربر می تواند زمان را تنظیم کند، المان موجود در صفحه‌ی پانل باید از نوع کنترل باشند. این المان جهت نمایش سپری شدن زمان باید قابلیت دوران و چرخش نیز داشته باشد.

۱- صفحه‌ی پانل این برنامه را مطابق شکل ۸-۱۲ ایجاد کنید.



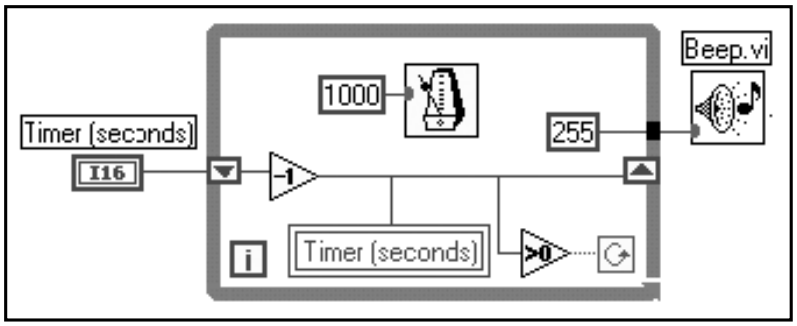

**شکل ۸-۱۲: صفحه‌ی پانل در برنامه‌ی Kitchen Timer.vi**

ابتدا یک دکمه‌ی چرخشی[2] با برچسب Timer (seconds) بر روی صفحه قرار دهید. این دکمه در زیرپالت Controls >> Numeric قرار دارد. سپس جهت ظاهر ساختن صفحه ی نمایش دهنده‌ی دیجیتالی، گزینه‌ی Show >> Digital Display را از منوی کرکره‌ای این دکمه انتخاب کنید. درجه‌بندی این دکمه را نیز مطابق شکل ۸-۱۲ تنظیم نمایید. برای تغییر درجه‌بندی، نشانگر ماوس را بر روی مقدار حداکثر درجه‌بندی این المان قرار دهید و با دو بار کلیک کردن بر روی آن، مقدار مورد نظر را فعال کنید. سپس مقدار مطلوب را وارد نمایید. نوع نمایش کلید چرخشی Timer (seconds) را نیز به فرمت I16 تغییر دهید. برای انجام این عمل گزینه‌ی Representation >> I16 را از منوی کرکره‌ای این المان انتخاب نمایید.

2- Knob

۲- با انتخاب گزینه‌ی Local Variable از زیرپالت Functions >> Structures، یک متغیر محلی ایجاد کنید. پس از ظاهر شدن آیکن متغیر محلی، به کمک ابزار Operating Tool بر روی این آیکن کلیک نمایید و گزینه‌ی Timer (seconds) را انتخاب کنید. همان گونه که قبلا عنوان شد با کلیک کردن بر روی آیکن متغیر محلی لیستی از المان های کنترل یا نشان دهنده‌ی برچسب دار که در صفحه‌ی پانل قرار دارند ظاهر می شوند. این متغیر محلی باید به صورت پیش فرض در مد نوشتن تعریف شده باشد. دلیل این موضوع را در چند سطر آینده متوجه خواهید شد. از تغییر دادن این مد خودداری کنید.

۳- صفحه‌ی نمودار بلوکی این برنامه را مطابق شکل ۹-۱۲ ایجاد کنید.




**شکل ۹-۱۲: صفحه‌ی نمودار بلوکی در برنامه‌ی Kitchen Timer.vi**

عملکرد این برنامه به شرح زیر است:

در این برنامه ابتدا مقدار المان Timer (seconds) در ثبات انتقال قرار می‌گیرد. این مقدار همان عددی است که توسط کاربر در صفحه‌ی پانل تنظیم می گردد. پس از به اجرا در آمدن حلقه، مقدار ذخیره شده در ثبات انتقال به ازای هر ثانیه یک واحد کاهش می یابد. مقدار مذکور در متغیر محلی Timer (seconds) قرار داده می شود. به همین دلیل این متغیر محلی در مد نوشتن انتخاب شده است. کلید چرخشی موجود در صفحه‌ی پانل جهت منعکس نمودن مقدار تغییر یافته در جهت خلاف عقربه های ساعت به چرخش در می آید.

۴- حال تایمر را بر روی عددی دلخواه تنظیم کنید و برنامه را به اجرا درآورید. به چرخش این دکمه در جهت خلاف عقربه های ساعت توجه کنید. این حرکت تا زمانی ادامه می یابد که دکمه بر روی عدد 0 قرار گیرد.

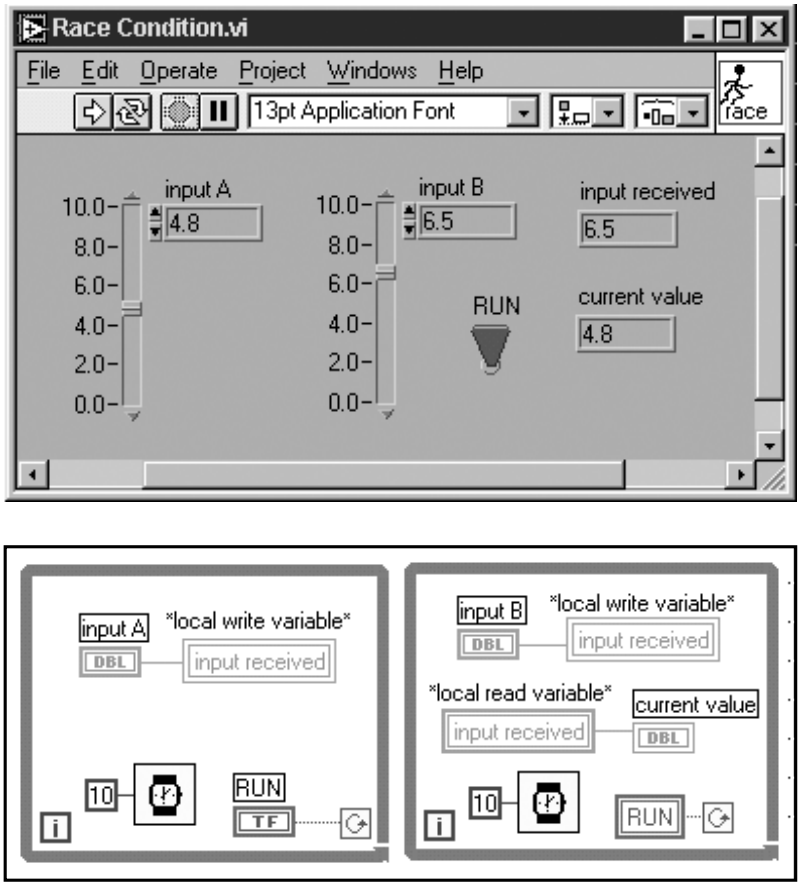
۵- این برنامه را با عنوان Kitchen Timer.vi در زیرفهرست LabVIEW\My Activity ذخیره کنید.

در اکثر تایمرهای به کار برده شده در موارد مهم و حساس، برای آگاه نمودن کاربر از سپری شدن زمان، از یک آلارم هشداردهنده استفاده می شود. به احتمال زیاد نظیر این حالت را در زمان سنج اجاق گاز آشپزخانه دیده اید. اضافه نمودن قابلیت تولید صدا به این برنامه می تواند کاربرد آن را گسترش دهد. در این فصل خواهید دید که چگونه می توان در محیط LabVIEW به تولید صدا پرداخت.

نکته‌ای که باید در هنگام به‌کارگیری متغیرهای محلی بدان توجه شود بروز یک مشکل عمومی است که در اصطلاح به آن حالت رقابتی[3] گوییم. این حالت در شرایطی روی می‌دهد که دو یا چند کپی از یک متغیر محلی در مد نوشتن وجود داشته باشند و پیشگویی در مورد ترتیب به اجرا در آمدن آنها و یا به عبارت دیگر نوشته شدن بر روی آنها امکان ناپذیر باشد.

۱۲

در هنگام استفاده از متغیرهای محلی و سراسری ممکن است یک حالت رقابتی به طور ناخواسته ایجاد گردد. برای درک این مطلب صفحات پانل و نمودار بلوکی را مطابق شکل ۱۰-۱۲ ایجاد نمایید. در این برنامه کنترل دو حلقه‌ی While توسط المان جبری RUN و متغیر محلی RUN صورت می گیرد. توجه داشته باشید که متغیر محلی input received در دو حلقه در مد نوشتن استفاده شده است.




**شکل ۱۰-۱۲: صفحات پانل و نمودار بلوکی در برنامه‌ی Race Condition.vi**

با تغییر وضعیت المان جبری RUN به حالت False و قراردادن دکمه های لغزشی بر روی اعداد متفاوت، برنامه را اجرا کنید. به دلیل قرار گرفتن المان جبری در حالت False، هر دو حلقه تنها یک مرتبه اجرا می شوند.

مقدار ظاهر شده در نشان دهنده‌ی Current Value چه عددی است؟ در این حالت امکان پیشگویی وجود

3- Race Condition

ندارد، زیرا ممکن است عدد ظاهر شده مربوط به یکی از دو مقدار input A یا input B باشد. در این موارد هیچ روشی نمی تواند تعیین کننده‌ی ترتیب اجرای حلقه ها به عنوان مثال از چپ به راست یا از بالا به پایین باشد.

اگر با قرار دادن المان جبری RUN در وضعیت True، این برنامه را مجدداً به اجرا در آورید، ملاحظه خواهید نمود که مقدار ظاهر شده در نشان دهنده‌ی Current Value دائماً بین دو مقدار ورودی که توسط کلیدهای لغزشی تنظیم شده‌اند پرش می کند. برای جلوگیری از پیدایش حالت رقابتی باید ترتیب اجرا شدن دو یا چند متغیر محلی در مد نوشتن، توسط روشهایی نظیر اصل جریانِ داده، ساختارهای ترتیبی و... تعیین شود. بنابراین در هنگام به کارگیری متغیرهای محلی برای اجتناب از بروز حالت رقابتی حتماً صفحه‌ی نمودار بلوکی و المان های موجود در آن را به منظور بررسی نحوه‌ی اجرای حلقه ها و... بازبینی کنید.

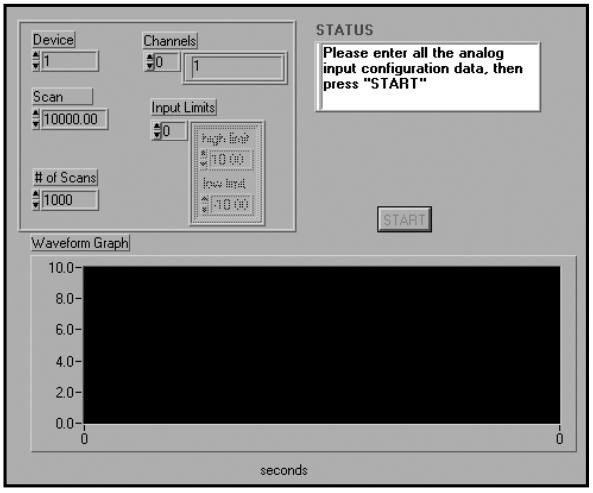
## ۱۲ تمرین ۲-۱۲ : نشان‌دهنده‌ی وضعیت

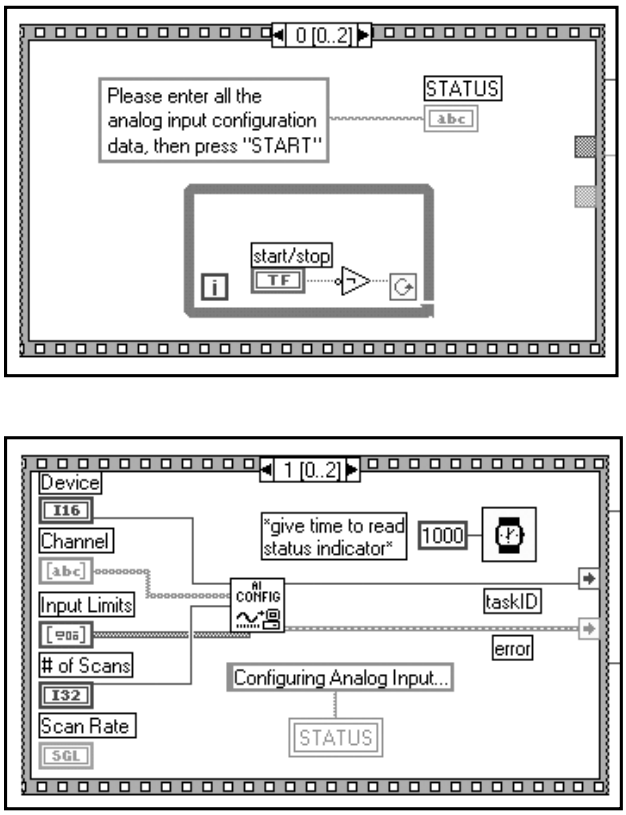
در مواردی که جهت ایجاد برنامه به یک نشان دهنده‌ی وضعیت نیاز دارید می توانید از متغیرهای محلی استفاده کنید. به عنوان مثال ممکن است به یک نشان دهنده‌ی رشته ای نیاز داشته باشید که توسط یک متن به روز رسانده می شود یا اینکه هر زمان که اتفاقی در برنامه روی می دهد این رشته، یک ورودی را درخواست کند.

۱- یک برنامه‌ی ساده‌ی DAQ را نظیر برنامه‌ی Buffered Analog In.vi که در فصل ۱۱ ایجاد نمودید بازنویسی کنید. اما این بار برنامه را به گونه ای اصلاح کنید که حاوی یک نشان دهنده‌ی رشته‌ای باشد و اطلاعات زیر را در اختیار کاربر قرار دهد.

- مدت زمانی که برنامه منتظر دریافت ورودی است.
- مدت زمانی که برنامه‌ی در حال جمع آوری داده هاست.
- مدت زمانی که برنامه در حال ترسیم داده هاست.
- مدت زمانی که برنامه متوقف شده است.

برای راهنمایی شما در نوشتن برنامه، در شکل ۱۱-۱۲ صفحه‌ی پانل این برنامه به همراه دو فریم اولیه از ساختار ترتیبی آن در صفحه‌ی نمودار بلوکی نشان داده شده است.






**شکل ۱۱-۱۲: صفحه‌ی پانل و دو فریم از نمودار بلوکی در برنامه‌ی Status Indicator.vi**

۲- این برنامه را با عنوان Status Indicator.vi در زیرفهرست LabVIEW\My Activity ذخیره کنید.

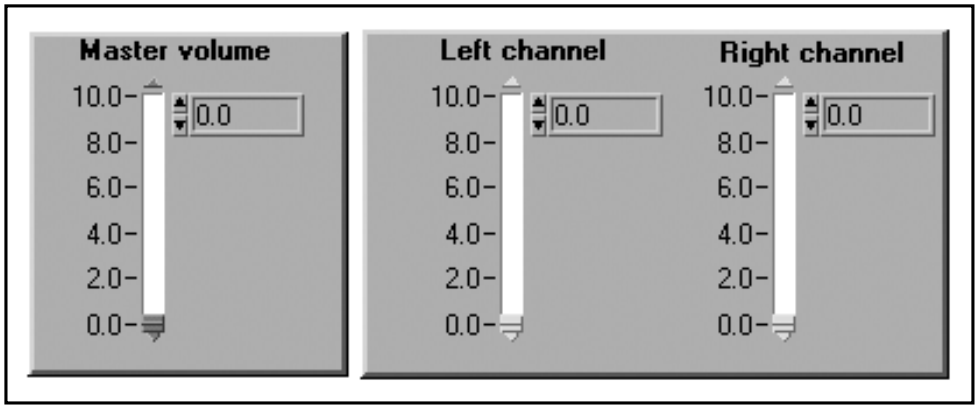
## تمرین ۳-۱۲: المان کنترل اصلی و فرعی

در صورت نیاز به تعریف و به کارگیری یک المان کنترل اصلی[4] می توانید از متغیرهای محلی استفاده کنید. این المان کنترل می‌تواند مقادیر المان های دیگر را تغییر دهد یا اصلاح کند. در حقیقت با ایجاد تغییر در المان کنترل اصلی، المان های کنترل دیگر به تبعیت از المان اصلی تغییر می کنند.

فرض کنید که به کمک یک پانل ساده قصد دارید تا ولوم های استریوی منزل خود را کنترل نمایید و رایانه به نحوی با ولوم های استریو در ارتباط است. در برنامه‌ی نشان داده شده در شکل ۱۲-۱۲ یک پانل کنترل صدا را که شامل ۳ کلید لغزشی است ملاحظه می کنید. این ۳ کلید با عناوین ولوم اصلی، کانال سمت چپ و کانال سمت راست مشخص شده‌اند. کانال های سمت راست و سمت چپ را می توان مستقل از یکدیگر تنظیم نمود. اما حرکت دادن کلید لغزشی اصلی باعث تغییر وضعیت کلیدهای سمت چپ و راست می شود.

۱- صفحه‌ی پانل این برنامه را مطابق شکل ۱۲-۱۲ ایجاد کنید.

4- Master Control




**شکل ۱۲-۱۲: صفحه‌ی پانل در برنامه‌ی Master and Slave.vi**

۲- حال برای این برنامه، یک صفحه‌ی نمودار بلوکی ایجاد نمایید. نکته‌ی جالب توجه در این برنامه آن است که با حرکت دادن کلید اصلی، جابه‌جا شدن کلیدهای سمت راست و چپ را بر روی صفحه مشاهده می کنید.

۳- این برنامه را با عنوان Master and Slave.vi در زیرفهرست LabVIEW\My Activity ذخیره کنید.

برای نوشتن این برنامه از ثبات انتقال کمک بگیرید.

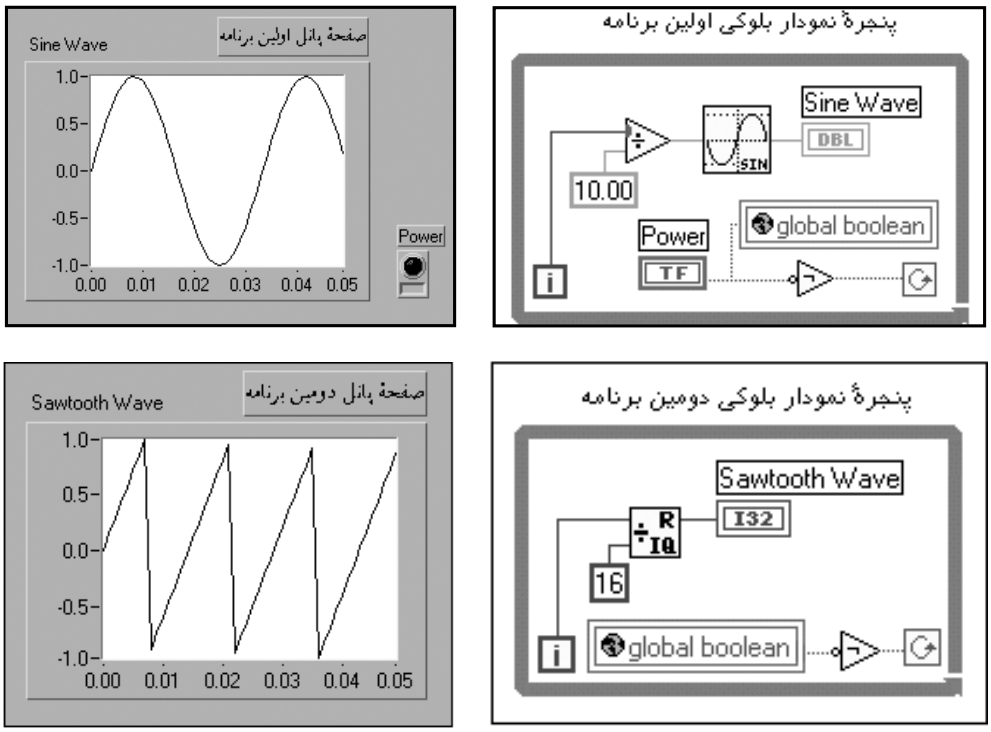
## متغیرهای سراسری

در محیط برنامه نویسی LabVIEW به ندرت از متغیرهای سراسری استفاده می گردد. این متغیرها اغلب اما نه همواره باعث ایجاد اشکال و بروز رفتارهای دور از انتظار می شوند. با این حال در برخی شرایط به اجبار از این متغیرها استفاده می کنیم.

همان طور که در بخش قبل عنوان شد برای دستیابی به المان های موجود در صفحه‌ی پانل از نقاط مختلف صفحه‌ی نمودار بلوکی، از متغیرهای محلی استفاده می شود. دستیابی به این متغیرها تنها در یک برنامه امکان پذیر است. حال فرض کنید که قصد دارید به داده های چند برنامه که به نحوی در ارتباط با یکدیگر می باشند دسترسی پیدا کنید، یا اینکه فرض کنید سیم کشی و برقراری ارتباط بین آیکن زیربرنامه ها در صفحه‌ی نمودار بلوکی برنامه‌ی اصلی امکان پذیر نیست. در این موارد چه باید کرد؟

برای انجام این اعمال می توان از متغیر های سراسری استفاده نمود. در بسیاری موارد عملکرد متغیرهای سراسری شبیه به متغیرهای محلی است. با این تفاوت که در مورد متغیرهای سراسری محدودیت به کارگیری در یک برنامه وجود ندارد. برای درک بیشتر این مطلب مثال زیر را در نظر بگیرید:

فرض کنید که دو برنامه‌ی نشان داده شده در شکل ۱۳-۱۲ همزمان با یکدیگر به اجرا در می آیند.




شکل ۱۳-۱۲

در هر بار اجرای این برنامه‌ها یک نقطه بر روی نمودار نقش می‌بندد. در صفحه‌ی پانل اولین برنامه، یک دکمه با برچسب Power جهت پایان دادن به اجرای هر دو برنامه قرار داده شده است. در صورتی که داده‌های دو حلقه‌ی مستقل در یک برنامه بر روی یک نمودار به نمایش در می‌آمدند نیاز به تعریف متغیر محلی داشتیم. اما در شرایط حاضر که حلقه‌ها در دو برنامه‌ی مجزا قرار دارند به ناچار باید از متغیر سراسری استفاده کنیم. با دقت در شکل ۱۳-۱۲ ملاحظه می‌کنید که آیکن متغیر سراسری نیز شبیه به آیکن متغیر محلی است، با این تفاوت که در داخل آیکن آن یک «کره‌ی زمین» دیده می‌شود.

متغیر سراسری نیز نظیر متغیر محلی در زیرپالت Functions >> Structures قرار دارد. ترمینال یک متغیر سراسری نظیر متغیر محلی می‌تواند در یکی از مدهای خواندن یا نوشتن انتخاب شود. اما برخلاف متغیرهای محلی، برنامه‌های مختلف می‌توانند یک متغیر سراسری را مستقل از یکدیگر فراخوانی کنند. به‌کارگیری متغیرهای سراسری در حقیقت روشی برای تقسیم نمودن داده‌ها بین چند برنامه بدون نیاز به سیم‌کشی و اتصال داده‌ها از یک برنامه به برنامه‌های دیگر است. داده‌های متغیرهای سراسری، مستقل از برنامه‌ها ذخیره می‌شود. بدین معنی که اگر در برنامه‌ای مقدار یک متغیر سراسری تغییر یابد، پس از این تغییر، هر برنامه یا زیربرنامه‌ی دیگر، مقادیر به روز رسانده شده را قرائت می‌کند.

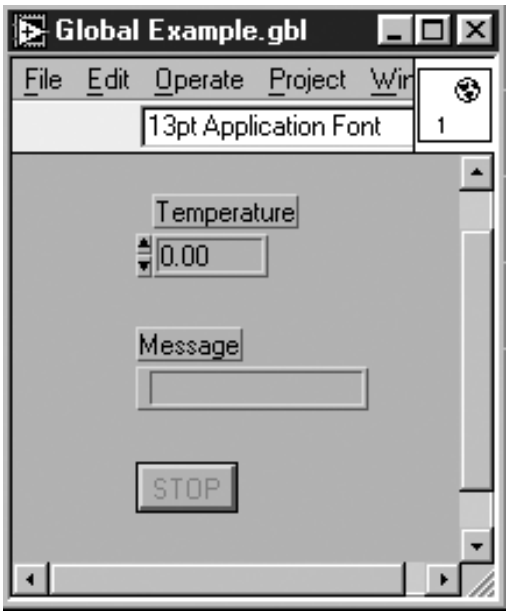
Undefined Global

هنگامی که متغیر سراسری را برای اولین بار انتخاب می‌کنید و آن را بر روی صفحه‌ی نمودار بلوکی قرار می‌دهید آیکن آن به صورت شکل سمت راست ظاهر می‌شود. این آیکن بیانگر یک متغیر سراسری است که هنوز تعریف نشده است. با دو بار کلیک کردن بر

روی این آیکن، یک منوی کرکره‌ای باز می‌شود که تمامی المان‌های برچسب‌دار موجود در صفحه‌ی پانل در آن دیده می‌شوند. در حقیقت می‌توان متغیرهای سراسری را نوع خاصی از برنامه در نظر گرفت که در صفحه‌ی پانل آن می‌توان هر نوع ساختار را به تعداد دلخواه قرار داد، با این تفاوت که المان‌های موجود در این صفحه‌ی پانل فاقد ترمینال متناظر در صفحه‌ی نمودار بلوکی هستند. نحوه‌ی قرار دادن المان‌های کنترل یا نشان‌دهنده بر روی صفحه‌ی پانل متغیر سراسری شبیه به قرار دادن آنها در صفحه‌ی پانل برنامه است.

به دلیل اینکه خواندن از/نوشتن بر روی متغیرهای سراسری امکان‌پذیر است، انتخاب نوع داده‌ی مورد نظر برای یک المان کنترل یا نشان‌دهنده تفاوتی ندارد. در پایان ذکر این نکته ضروری است که حتماً برای المان‌های موجود در صفحه‌ی پانل برچسب تعیین کنید. زیرا تعیین متغیر سراسری برای المان‌های فاقد برچسب امکان‌پذیر نیست. همان گونه که در شکل ۱۴-۱۲ ملاحظه می‌کنید در صفحه‌ی پانل این متغیر سراسری، یک متغیر عددی، یک المان کنترل رشته‌ای و یک المان جبری قرار دارند.

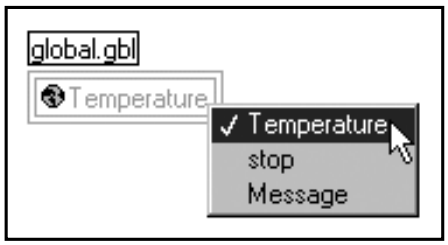





**شکل ۱۴-۱۲: صفحه‌ی پانل یک متغیر سراسری**

روش ذخیره نمودن یک متغیر سراسری نظیر روش ذخیره‌ی یک برنامه است. بسیاری از کاربران در هنگام نامگذاری این متغیرها برای سهولت دستیابی به آنها از پسوند gbl. استفاده می‌کنند. جهت استفاده از یک متغیر سراسری ذخیره شده در صفحه‌ی نمودار بلوکی برنامه، ابتدا آن را از زیرپالت Functions >> Select a VI... انتخاب کنید. پس از انجام این عمل یک ترمینال که تمامی متغیرهای سراسری در آن نشان داده می‌شود، بر روی صفحه ظاهر می‌گردد. برای انتخاب متغیری که قصد به کارگیری آن را دارید منوی کرکره‌ای را بر روی ترمینال مذکور باز کنید و گزینه‌ی Select Item را انتخاب نمایید یا توسط ابزار Operating Tool بر روی ترمینال مذکور کلیک کنید. در هر لحظه می‌توانید تنها یک متغیر را بر روی ترمینال متغیر سراسری انتخاب کنید. برای به کارگیری المان‌های دیگر به عنوان متغیر سراسری، یک

ترمینال دیگر برای متغیر سراسری مورد نظر ایجاد کنید




**شکل ۱۵-۱۲: نحوه‌ی انتخاب یک متغیر سراسری**

۱۲

متغیرهای سراسری نظیر متغیرهای محلی می توانند در یکی از دو مد نوشتن یا خواندن مورد استفاده قرار گیرند. جهت انتخاب مد، منوی کرکره‌ای را بر روی ترمینال مورد نظر باز نمایید و گزینه‌ی ...Change to را انتخاب کنید. در مورد متغیرهای محلی خاطر نشان کردیم که هر متغیر محلی را در دو مد خواندن و نوشتن می توان به ترتیب متناظر با المان کنترل و نشان دهنده در نظر گرفت. همچنین عنوان کردیم که حاشیه‌ی ترمینال های متغیرهای محلی در مد خواندن پهن تر از مد نوشتن می باشد. در مورد متغیرهای سراسری نیز تمامی این مطالب صادق هستند و می توان متغیرهای سراسری را نیز در دو حالت خواندن و نوشتن، معادل با منبع و مقصد در نظر گرفت.

برخی از نکات مهم در استفاده از متغیرهای سراسری عبارتند از:

۱- همواره متغیرهای سراسری را از داخل صفحه‌ی نمودار بلوکی تعریف کنید. این نوع متغیرها با خارج شدن از محیط LabVIEW مجدداً مقادیر پیش فرض خود را اختیار می کنند.

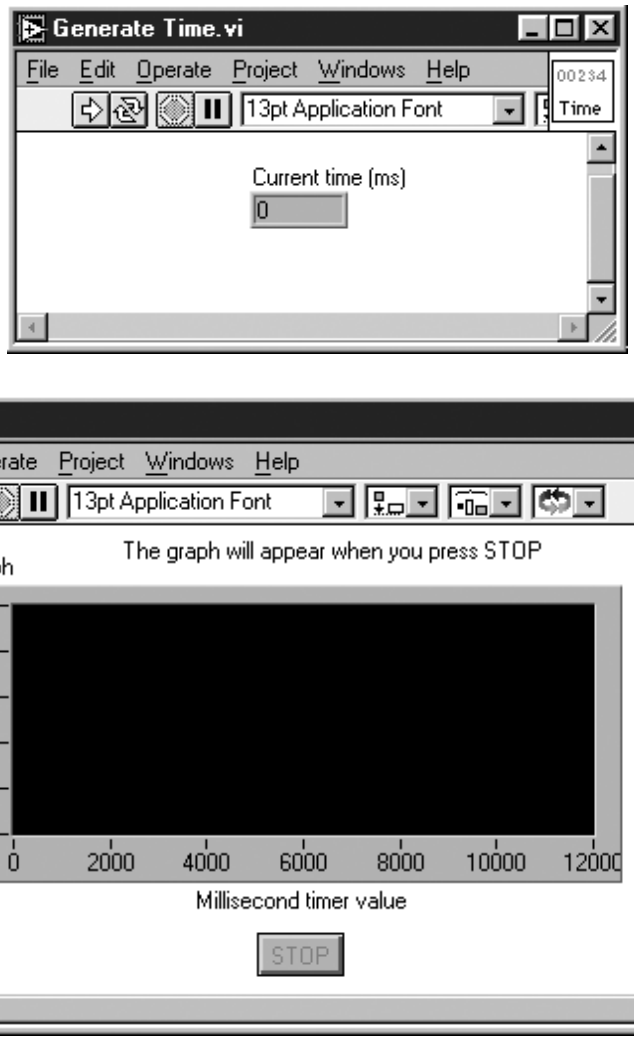
۲- هیچ گاه عمل خواندن و نوشتن را به صورت همزمان بر روی یک متغیر سراسری انجام ندهید(به عنوان مثال در مواردی که ترتیب انجام گرفتن اعمال تعریف نشده است). در این صورت حالت رقابتی پیش می آید.

۳- به دلیل اینکه یک متغیر سراسری می تواند چند نوع داده‌ی مختلف را در خود ذخیره کند به جای استفاده و ذخیره‌ی داده ها در چند متغیر سراسری ابتدا آنها را دسته بندی کنید، سپس تمامی آنها را در یک متغیر سراسری قرار دهید.

۴- به نامهای اختصاص داده شده به متغیرهای داخل هر مجموعه‌ی سراسری توجه کنید. زیرا تمام برنامه هایی که متغیرهای موجود در یک مجموعه را فراخوانی می کنند، متغیر را توسط نام آن می شناسند. به عبارت دیگر مبنای مراجعه به یک متغیر در حقیقت نام آن است. بنابراین از انتساب نامهای مشابه به المان های کنترل یا نشان دهنده در یک مجموعه‌ی سراسری خودداری کنید.

حال به بررسی مشکلی نظیر مسأله‌ای که در مورد دو حلقه‌ی مستقل While عنوان شد می پردازیم. تصور کنید که به جای دو حلقه‌ی مستقل در صفحه‌ی نمودار بلوکی دو زیربرنامه‌ی مستقل داریم که با هم به اجرا در می آیند. در شکل ۱۶-۱۲ صفحات پانل این زیربرنامه ها نشان داده شده است. دو زیربرنامه‌ی Plot.vi و Generate Time.vi به گونه ای طراحی شده اند که همزمان با یکدیگر اجرا شوند. زیربرنامه‌ی Generate Time.vi به طور پیوسته ضربات پالس را از کلاک داخلی رایانه دریافت می کند. این کلاک ها از زمان به اجرا در آمدن

برنامه ایجاد می‌شوند. زیربرنامه‌ی Plot.vi تا زمان فشرده شدن دکمه‌ی STOP در هر لحظه یک عدد تصادفی تولید می‌کند. سپس تمامی ضربات پالس‌های تولید شده را از برنامه‌ی Generate Time.vi دریافت و منحنی اعداد تصادفی را نسبت به زمان ترسیم می‌کند.

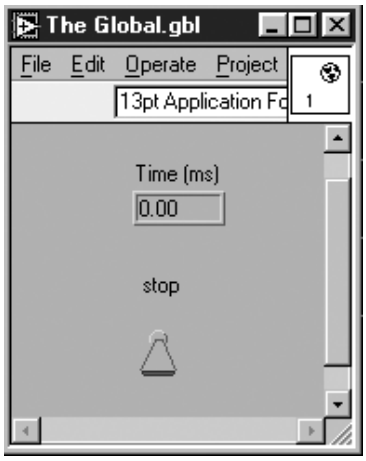



**شکل ۱۶-۱۲: صفحات پانل در زیربرنامه‌های Generate Time.vi و Plot.vi**

این دو زیربرنامه از طریق متغیر سراسری با یکدیگر ارتباط برقرار می‌کنند. روند اجرای برنامه‌ی اصلی به گونه‌ای است که زیربرنامه‌ی Plot.vi، آرایه‌ی حاوی مقادیر زمانی ایجاد شده توسط زیربرنامه‌ی Generate Time.vi را دریافت می‌نماید و آن را ترسیم می‌کند.

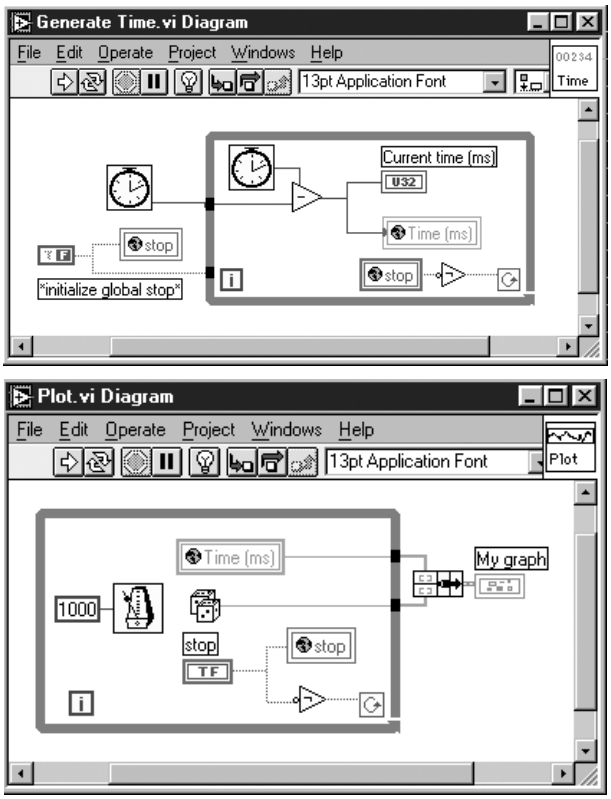
نکته‌ی مهم و دلیل به کارگیری متغیر سراسری در این مثال آن است که قصد داریم تا توسط یک کلید جبری اجرای هر دو زیربرنامه را متوقف سازیم. بنابراین ابتدا به ایجاد یک متغیر سراسری که مشتمل بر ۲ المان است می‌پردازیم. جهت ایجاد یک متغیر سراسری جدید ابتدا المان Global Variable را از زیرپالت Functions >> Structures انتخاب نمایید، سپس برای تعریف اجزای آن بر روی آیکن «کره‌ی زمین» دو بار کلیک کنید. در این تمرین

اجزای متغیر سراسری را به صورت مجموعه‌ای از المان عددی (Time (ms و المان جبری stop تعریف کنید و متغیر سراسری را نیز با عنوان The Global.gbl نام‌گذاری کنید. در شکل ۱۷-۱۲ صفحه‌ی پانل این متغیر سراسری را ملاحظه می کنید.




**شکل ۱۷-۱۲: صفحه‌ی پانل در متغیر سراسری The Global.gbl**

حال باید اجزای تعریف شده برای این متغیر سراسری را در مکانهای مناسب در صفحات نمودار بلوکی هر دو زیربرنامه به کار بریم. در شکل ۱۸-۱۲ مکانهای مذکور و نحوه‌ی به کارگیری آنها نشان داده شده است.




**شکل ۱۸-۱۲: نحوه‌ی به‌کارگیری متغیر سراسری در صفحه‌ی نمودار بلوکی دو زیربرنامه‌ی Generate Time.vi و Plot.vi**

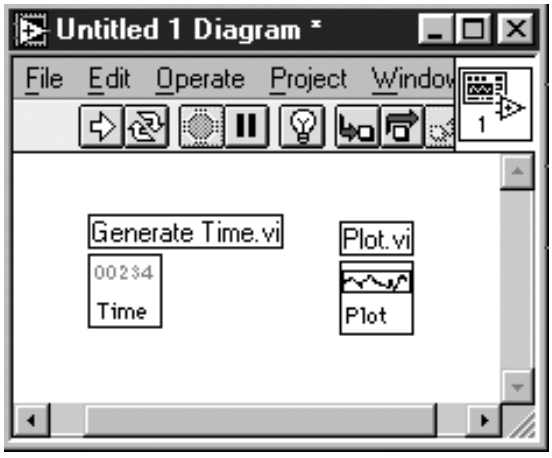
عملکرد این دو زیربرنامه به صورت زیر است:

دکمه‌ی stop در صفحه‌ی پانل زیربرنامه‌ی Plot.vi قرار دارد و به متغیر سراسری و المان جبری stop در این زیربرنامه فرمان می‌دهد. این متغیر نیز به نوبه‌ی خود جهت متوقف ساختن حلقه‌ی While در زیربرنامه‌ی Generate Time.vi به کار گرفته شده است. همان گونه که عنوان شد به هنگام فراخوانی زیربرنامه‌ی Plot.vi می توان از طریق صفحه‌ی پانل، به دکمه‌ی stop دست یافت. هنگامی که این دکمه در زیربرنامه‌ی Plot.vi فشار داده شود، هر دو زیربرنامه متوقف می‌شوند. به طریق مشابه، فرآیند تولید مقادیر زمانی توسط زیربرنامه‌ی Time.vi Generate که به متغیر سراسری Time (ms) انتقال داده می شود و در زیربرنامه‌ی Plot به کار برده شده است، نیز متوقف می گردد. به عبارت دیگر تولید آرایه ای که حاوی مقادیر زمانی ایجاد شده توسط زیربرنامه‌ی Generate Time.vi است متوقف می‌شود.

۱۲

با بررسی این مثال، مفهوم و روش به کارگیری متغیرهای سراسری را به خوبی فرا خواهید گرفت. متوقف ساختن این دو زیربرنامه بدون استفاده از متغیرهای سراسری نیز امکان پذیر است. اما در اینجا هدف از ارائه‌ی این مثال تفهیم متغیرهای محلی و روش به کارگیری آنها در صفحه‌ی نمودار بلوکی است. حال به ذکر روش دیگری برای انجام این تمرین می پردازیم:

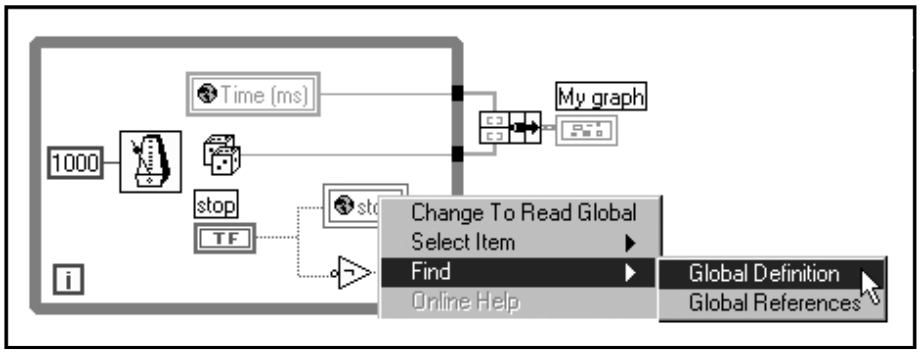
در صفحه‌ی نمودار بلوکی نشان داده شده در شکل ۱۹-۱۲، همان دو زیربرنامه را ملاحظه می کنید. در اینجا در ارتباط با به کارگیری متغیرهای سراسری مشکل دیگری ظهور می کند. مشکل این است که در به کارگیری این متغیرها هیچ سیمی وجود ندارد. به عبارت دیگر در این صفحه‌ی نمودار بلوکی هیچ سیمی مشاهده نمی شود. به دلیل اینکه هیچ گونه اطلاعاتی در مورد نحوه‌ی ارتباط دو زیربرنامه در اختیار ندارید، اصل جریانِ داده به صورت مبهم در می آید. حتی در صورت مشاهده‌ی یک متغیر سراسری بر روی صفحه‌ی نمودار بلوکی نمی دانید که این متغیر از چه نقطه ای فرمان گرفته است. خوشبختانه در نسخه‌ی LabVIEW 4.0 و نسخه های بالاتر این مشکل با گنجاندن قابلیت جستجوی المان های یک متغیر سراسری برطرف شده است.




شکل ۱۹-۱۲

همان گونه که عنوان شد در نسخه های بالاتر از LabVIEW 4.0، جستجوی المان های یک متغیر سراسری

برای کسب اطلاع در مورد منشأ ارسال فرمان به متغیر مورد نظر امکان‌پذیر است. برای انجام این عمل، منوی کرکره‌ای را بر روی متغیر سراسری مورد نظر بازکرده، سپس گزینه‌ی Find >> Global Definition را انتخاب کنید. با انتخاب این گزینه، محل تعریف و منشأ پیدایش متغیر سراسری در صفحه‌ی پانل مشخص می‌شود. با انتخاب گزینه‌ی Find >> Global References لیستی از تمامی برنامه‌هایی که متغیر مذکور در آنها به کار رفته است ظاهر می‌شود.

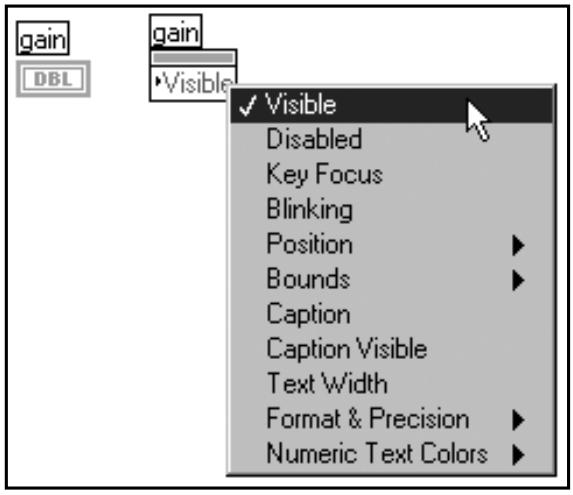



**شکل ۲۰-۱۲: نحوه‌ی یافتن منشأ پیدایش یک متغیر سراسری**

## گره‌های خصوصیت

به کمک گره‌های خصوصیت می‌توان قدرت و جذابیت برنامه‌های نوشته شده را افزایش داد. این گره‌ها به شما اجازه می‌دهند تا خصوصیات المان‌های موجود در صفحه‌ی پانل را کنترل کنید. این خصوصیات عبارتند از: رنگ، قابلیت به نمایش در آمدن، محل، فرمت عددی و.... به عنوان مثال می‌توانید رنگ یک المان dial را در صورت افزایش مقدار عددی آن تغییر دهید و با افزایش مقدار المان، یکی از رنگهای آبی، سبز و قرمز را برای آن در نظر بگیرید یا می‌توانید قابلیت نمایش برخی از المان‌ها را تعیین کنید. بدین مفهوم که در صورت فشار دادن برخی از دکمه‌ها، المان‌های از پیش تعیین شده ظاهر یا ناپدید شوند. به عبارت دیگر قابلیت به نمایش در آمدن المان‌ها بر روی صفحه به صورت انتخابی باشد. به گونه‌ای که کاربر بتواند ظهور یا عدم ظهور آنها را بر روی صفحه توسط یک یا چند کلید تعیین کند. همچنین می‌توانید برای به نمایش در آوردن برخی از فرآیندهای فیزیکی، با حرکت دادن یک المان بر روی صفحه، بخشی از صفحه‌ی برنامه‌ی خود را متحرک سازید.

برای ایجاد یک گره‌ی خصوصیت منوی کرکره‌ای را بر روی المان موجود در صفحه‌ی پانل یا بر روی ترمینال متناظر با آن در صفحه‌ی نمودار بلوکی باز کنید و سپس گزینه‌ی Create >> Attribute Node را انتخاب کنید. در این صورت یک ترمینال همنام با متغیر مورد نظر بر روی صفحه‌ی نمودار بلوکی ظاهر خواهد شد. برای مشاهده‌ی خصوصیاتی که می‌توانید در یک گره‌ی خصوصیت برای یک متغیر انتخاب نمایید به کمک ابزار Operating Tool بر روی گره‌ی مذکور کلیک کنید. روش دیگر برای دستیابی به این خصوصیات، باز کردن منوی کرکره‌ای بر روی آن گره و انتخاب گزینه‌ی Select Item می‌باشد. پس از مشاهده‌ی خصوصیات مذکور می‌توانید یک یا چند مورد را انتخاب نمایید. در شکل ۲۱-۱۲ تعدادی از این خصوصیات نشان داده شده است.

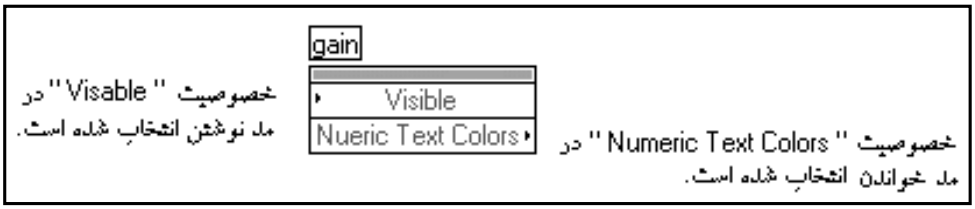



شکل ۲۱-۱۲

۱۲

نظیر متغیرهای محلی می توانید خصوصیات را در مورد یک المان بخوانید یا بر روی آن بنویسید. برای ایجاد تغییر در یک گره‌ی خصوصیت، منوی کرکره‌ای را بر روی آن باز نموده، سپس گزینه‌ی ...Change to را انتخاب کنید. مکان قرار گرفتن پیکان موجود در ترمینال یک گره‌ی خصوصیت، مد آن گره را نشان می دهد. در صورتی که گره‌ی خصوصیت در مد نوشتن انتخاب شده باشد، این پیکان در سمت چپ ترمینال دیده می شود و نشانگر آن است که داده‌ها در حال ورود به گره هستند. در صورتی که این پیکان در سمت راست ترمینال ظاهر شود نشان‌دهنده‌ی مد خواندن می‌باشد و بیانگر خروج داده‌ها از گره است. قاعده‌ی ذکر شده در مورد متغیرهای محلی و سراسری در اینجا نیز صادق است. بدین مفهوم که یک گره‌ی خصوصیت را در دو مد خواندن و نوشتن می توان متناظر با دو المان کنترل و نشان‌دهنده فرض نمود.

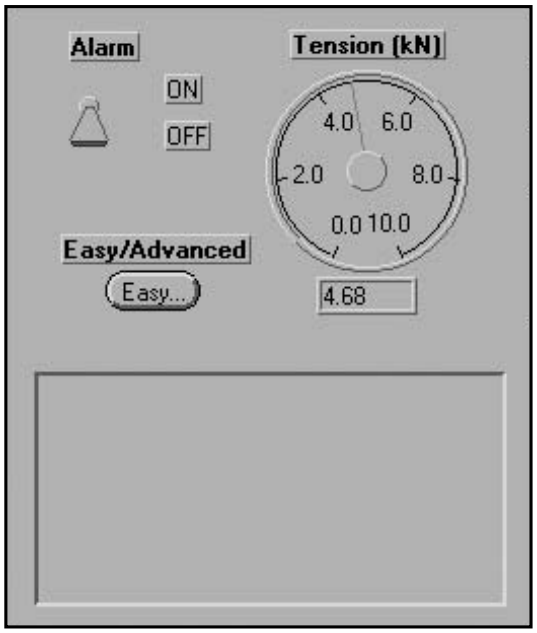
ویژگی جالب توجه در گره‌های خصوصیت آن است که می توانید برای یک ترمینال موجود در صفحه‌ی نمودار بلوکی، چند خصوصیت را به‌طور همزمان درنظر بگیرید. این عمل تنها بر روی المان کنترل یا نشان‌دهنده‌ی متناظر با همان ترمینال تأثیرگذار است. برای افزودن خصوصیات جدید به یک گره، اندازه‌ی ترمینال مورد نظر را به کمک ابزار Positioning Tool تغییر دهید به گونه‌ای که گنجایش تمامی خصوصیات مورد نیاز شما را داشته باشد. در شکل ۲۲-۱۲ ملاحظه می کنید که دو خصوصیت برای یک ترمینال المان کنترل عددی در نظر گرفته شده است.




شکل ۲۲-۱۲

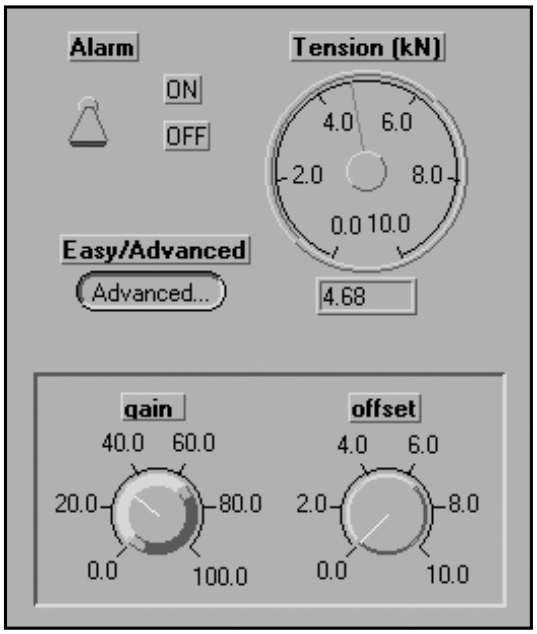
حال به بررسی یک مثال ساده می‌پردازیم. فرض کنید که قصد دارید برخی از المان‌های کنترل موجود در صفحه‌ی پانل تا زمانی که شما به آنها نیاز پیدا نکرده‌اید مخفی باقی بمانند یا در صفحه ظاهر نشوند. در شکل ۲۳-۱۲ یک نشان‌دهنده‌ی تنش Tension (kN) و یک کلید جبری Alarm را در صفحه‌ی پانل ملاحظه می کنید. در این صفحه یک دکمه با برچسب

Easy / Advanced تعبیه شده است که در صورت فشار دادن آن برخی از گزینه‌های پیشرفته و پیچیده ظاهر می شوند.



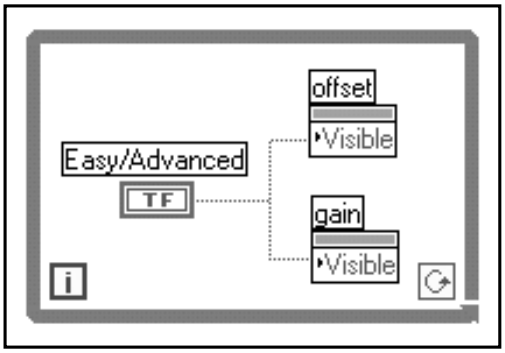

**شکل ۲۳-۱۲: شمای صفحه‌ی پانل قبل از فشردن دکمه‌ی Easy/Advanced**

در صفحه‌ی پانل این برنامه دو المان دیگر با برچسبهای gain و offset در نظر گرفته شده است که در شکل ۲۳-۱۲ قابل رؤیت نیستند. مخفی ساختن این دو المان با تغییر وضعیت گره‌ی خصوصیت Visible به حالت False انجام گرفته است. البته پس از فشرده شدن دکمه‌ی Easy/Advanced دو المان مذکور بر روی صفحه ظاهر می شوند. در شکل ۲۴-۱۲ شمای کلی صفحه‌ی پانل را پس از فشردن دکمه‌ی Easy/Advanced ملاحظه می کنید.




**شکل ۲۴-۱۲: شمای صفحه‌ی پانل پس از فشردن دکمه‌ی Easy/Advanced**

صفحه‌ی نمودار بلوکی این برنامه در شکل ۲۵-۱۲ نشان داده شده است. در این شکل ملاحظه می کنید که برای قابل رؤیت ساختن یا مخفی نمودن دو کلید چرخشی مذکور در صفحه‌ی پانل، ترمینال دکمه‌ی Easy/Advanced به همراه دو گره‌ی خصوصیت در مد نوشتن، در چهارچوب یک حلقه‌ی While قرار گرفته است. جهت ساده‌سازی صفحه‌ی نمودار بلوکی از گنجاندن سایر بخشها و ترمینال‌های دیگر در این شکل خودداری شده است.




**شکل ۲۵-۱۲: نحوه‌ی استفاده از خصوصیت Visible برای قابل رؤیت ساختن دو المان offset و gain**

در اغلب موارد نیاز دارید تا بیش از یک گزینه را در گره‌ی خصوصیت المان استفاده کنید. در این حالت می توانید به جای ایجاد یک گره‌ی خصوصیت دیگر، پس از تغییر اندازه‌ی ترمینال مورد نظر چند خصوصیت را توسط ابزار Operating Tool انتخاب کنید. در صورت انتخاب گزینه‌ی جدید در یک گره، این خصوصیت در زیرِ گزینه های قبلی ظاهر می شود. ترتیب قرار گرفتن خصوصیات یک گره به همان ترتیب انتخاب شدن آنهاست. در صورتی که قصد دارید ترتیب قرار گرفتن گزینه های مذکور را در یک گره‌ی خصوصیت تغییر دهید توسط ابزار Operating Tool بر روی گزینه‌ی مورد نظر کلیک کنید یا گزینه‌ی Select Item را از منوی کرکره‌ای انتخاب کنید. در ادامه به بررسی گزینه های اصلی موجود در یک گره‌ی خصوصیت می پردازیم:

- Visible : به کمک این گزینه می توان قابلیت رؤیت یا به تصویر در آمدن یک المان را تعیین نمود. در صورتی که وضعیت True در مورد این خصوصیت انتخاب شده باشد المان مورد بحث قابل رؤیت‌است و در غیر این صورت غیر قابل رؤیت خواهد بود. به دلیل اینکه المان هایی که با روش شفاف سازی یا فرانمایی، رنگ آمیزی می شوند ممکن است به طور تصادفی و ناخواسته انتخاب شوند، به کارگیری این خصوصیت در مورد یک المان خیلی بهتر و پرکاربردتر از روش شفاف سازی است.
- Disabled : این گزینه وضعیت دستیابی کاربر به یک المان کنترل را مشخص می کند. مقدار 0، المان کنترل را فعال می‌سازد و کاربر می تواند به آن دسترسی داشته باشد. مقدار 1، بدون اینکه هیچ نشانه‌ای از عدم فعالیت المان بر روی صفحه دیده شود، آن را غیر فعال می کند. مقدار 2، المان کنترل را غیر فعال می‌کند و آن را به رنگ خاکستری (غیر فعال) نمایش می دهد. تفاوت بین دو حالت اخیر یعنی مقادیر 1 و 2 تنها در وجود نشانه‌ای مبنی بر عدم فعالیت المان مذکور است وگرنه در هر دو صورت المان مورد نظر غیر فعال می گردد.
- Key Focus: در صورتی که این خصوصیت در وضعیت True باشد، می توان المان کنترل را توسط صفحه

کلید انتخاب نمود. در این حالت می توان با استفاده از کلید <Tab> بین نقاط مختلف صفحه حرکت کرد. این ویژگی در مواردی که استفاده از ماوس امکان پذیر نیست بسیار مفید و کاربردی می باشد.

- Position: این گزینه حاوی کلاستری مشتمل بر دو عدد است. این دو عدد مختصات گوشه‌ی بالایی سمت چپ المان موجود در صفحه‌ی پانل را نشان می دهند.
- Size: این گزینه حاوی کلاستری مشتمل بر دو عدد است. این دو عدد به ترتیب طول[5] و عرض[6] المان موجود در صفحه‌ی پانل را نشان می دهند.
- Blinking: در صورتی که این خصوصیت در وضعیت True باشد، المان موجود در صفحه‌ی پانل چشمک می زند.
- Format and Precision: این گزینه، دقت و فرمت اعداد مندرج در المان های کنترل یا نشان دهنده را تعیین می کند. کلاستر ورودی مربوط به این گزینه حاوی دو عدد است که یکی از آنها برای تعیین فرمت و دیگری جهت تنظیم دقت اعداد مورد استفاده قرار می گیرد. دستیابی به این خصوصیت از طریق گزینه‌ی موجود در منوی کرکره ای المان نیز امکان پذیر است.
- Color: به کمک این گزینه می توان بسته به نوع المان، متن، پس زمینه، پیش زمینه و... را رنگ آمیزی کرد.

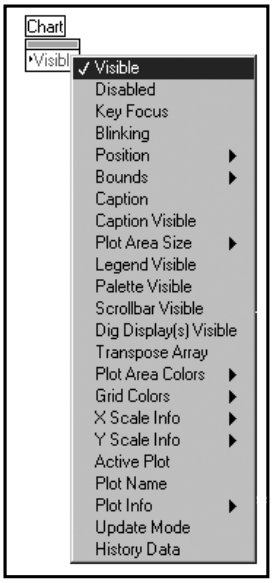
در هنگام استفاده از گره‌ی خصوصیت به کارگیری پنجره‌ی Help بسیار مفید و کاربردی است. اگر نشانگر ماوس را بر روی ترمینال یک گره‌ی خصوصیت حرکت دهید، مفهوم خصوصیت انتخاب شده در پنجره‌ی Help ظاهر می گردد. در این صفحه نوع داده‌ای که برای گره‌ی مذکور قابل قبول است نیز به نمایش در می آید. برای انجام سیم کشی و تعیین نوع داده‌ی قابل قبول برای المان، منوی کرکره‌ای را بر روی ترمینال گره‌ی خصوصیت باز کنید، سپس گزینه‌ی Create Constant را انتخاب کنید. پس از انجام این عمل، نوع داده‌ی قابل قبول بلافاصله به ترمینال مورد نظر سیم کشی می گردد. به کارگیری این گزینه در صورت نیاز به سیم کشی یک کلاستر بسیار مفید است.

تقریباً تمامی المان های کنترل یا نشان دهنده دارای خصوصیات پایه‌ی مذکور می باشند؛ اما در اکثر آنها به خصوص جداول و گراف ها تعداد این خصوصیات بیشتر است. حداکثر تعداد این گزینه ها ۷۳ خصوصیت می باشد. به دلیل این که به ندرت با گزینه های خاص سروکار خواهید داشت، در این کتاب از ذکر جزییات در مورد آنها خودداری می کنیم. بهترین روش برای یادگیری و درک جزییات در رابطه با گره های خصوصیت انجام تمرینات مختلف در مورد آنهاست. پس از ایجاد و به کارگیری این گره ها متوجه خواهید شد که خصوصیات مذکور در افزایش قابلیت انعطاف برنامه نقش بسزایی دارند.

## گره های خصوصیت در گراف و نمودار

تعداد گره های خصوصیت مربوط به گراف و نمودار بسیار زیاد است. همان گونه که عنوان شد برای دستیابی و مشاهده‌ی گزینه های موجود کافی است منوی کرکره‌ای را بر روی ترمینال دلخواه باز کنید. در شکل ۲۶-۱۲ گزینه های موجود در مورد گره‌ی خصوصیت یک نمودار را ملاحظه می کنید.

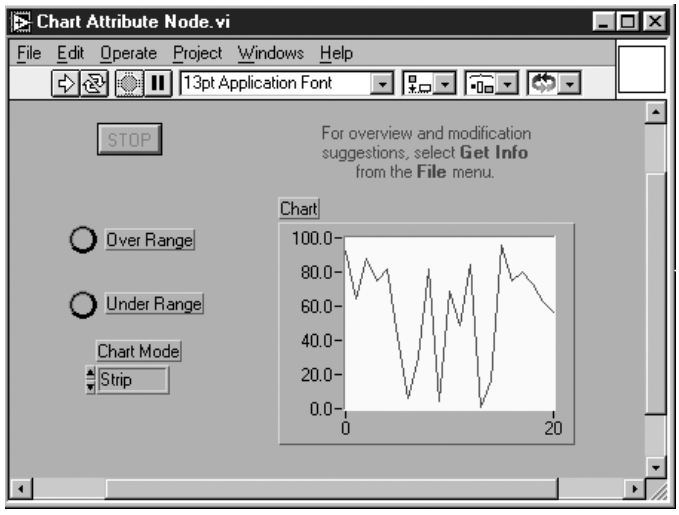
5- Height
6- Width




**شکل ۲۶-۱۲: گره‌های خصوصیت مربوط به نمودار**

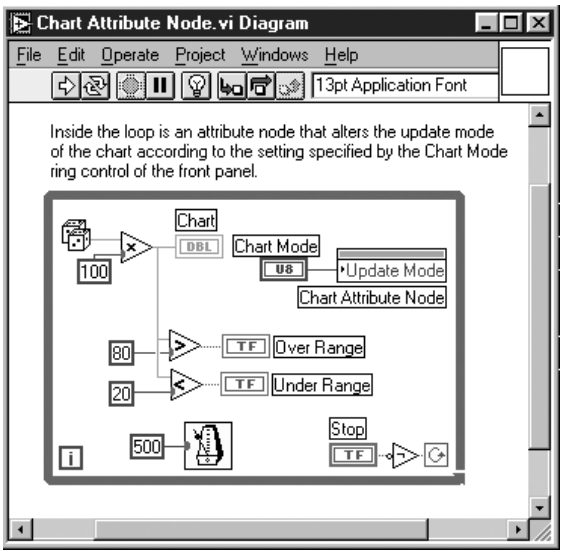


برخی از گزینه های موجود در گره‌ی خصوصیت خود دارای چند زیرگزینه هستند. این گره ها را گره های خصوصیت مرکب گویند. همان گونه که در شکل ۲۶-۱۲ ملاحظه می کنید در سمت راست این گزینه ها یک پیکان وجود دارد و بیانگر آن است که گزینه‌ی مذکور دارای یک زیرمنو می باشد. در این حالت می توان خصوصیت موجود را به صورت کلی انتخاب نمود، یا اینکه پس از دسترسی به هر یک از زیرگزینه های آن، گزینه‌ی دلخواه را انتخاب کرد. برای درک نحوه‌ی استفاده از گره های خصوصیت، برنامه‌ی Chart Attribute Node.vi را از مسیر LabVIEW\Examples\General\atribute.llb اجرا کنید. این برنامه به شما اجازه می دهد تا هر نمودار را به یکی از سه مد Sweep، Scope و Strip به نمایش در آورید. در صورتی که عملکرد و رفتار نمودار را در هر یک از مدهای مذکور فراموش کرده‌اید، مجدداً مطالب عنوان شده در فصل ۸ را مطالعه کنید. صفحه‌ی پانل این برنامه در شکل ۲۷-۱۲ نشان داده شده است.




**شکل ۲۷-۱۲: صفحه‌ی پانل در برنامه‌ی Chart Attribute Node.vi**

با تغییر دادن مد نمایشی نمودار در این برنامه می توانید تغییر یافتن نحوه‌ی نمایش را در حین اجرای برنامه نیز مشاهده کنید. در شکل ۲۸-۱۲ صفحه‌ی نمودار بلوکی این برنامه را ملاحظه می کنید.

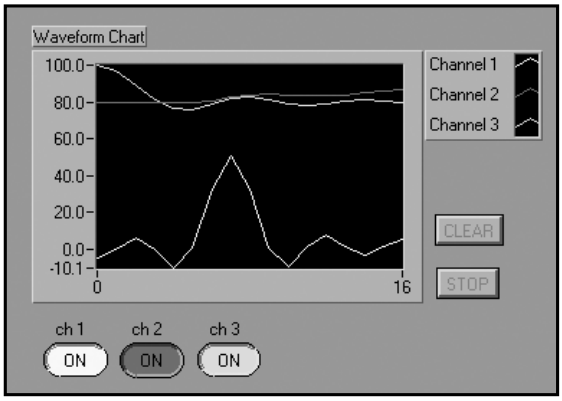



شکل ۲۸-۱۲: صفحه‌ی نمودار بلوکی در برنامه‌ی Chart Attribute Node.vi

## تمرین ۴-۱۲: ترسیم گراف با استفاده از گره‌ی خصوصیت

۱- برنامه‌ای ایجاد کنید که داده‌های سه کانال را رسم کند. این داده‌ها ممکن است به صورت داده‌های جمع آوری شده از طریق کارت DAQ و یا اعداد تصادفی باشند. در صفحه‌ی پانل این برنامه سه کلید قرار دهید که کاربر با انتخاب هر یک از آنها بتواند نمودار مربوط را بر روی صفحه به نمایش در آورد و یا آن را از روی صفحه‌ی نمایش حذف کند. به دلیل این که این داده‌ها بر روی نمودار ترسیم می شوند، داده‌های جدید به همراه داده‌های قبلی بر روی یک صفحه به نمایش در می آیند.

۲- یک دکمه با برچسب CLEAR بر روی صفحه‌ی پانل قرار دهید که به کمک آن بتوان اطلاعات کانال‌ها را حذف نمود و صفحه‌ی نمایش نمودار را پاک کرد. برای شروع برنامه نویسی از صفحه‌ی پانل نشان داده شده در شکل ۲۹-۱۲ استفاده کنید.

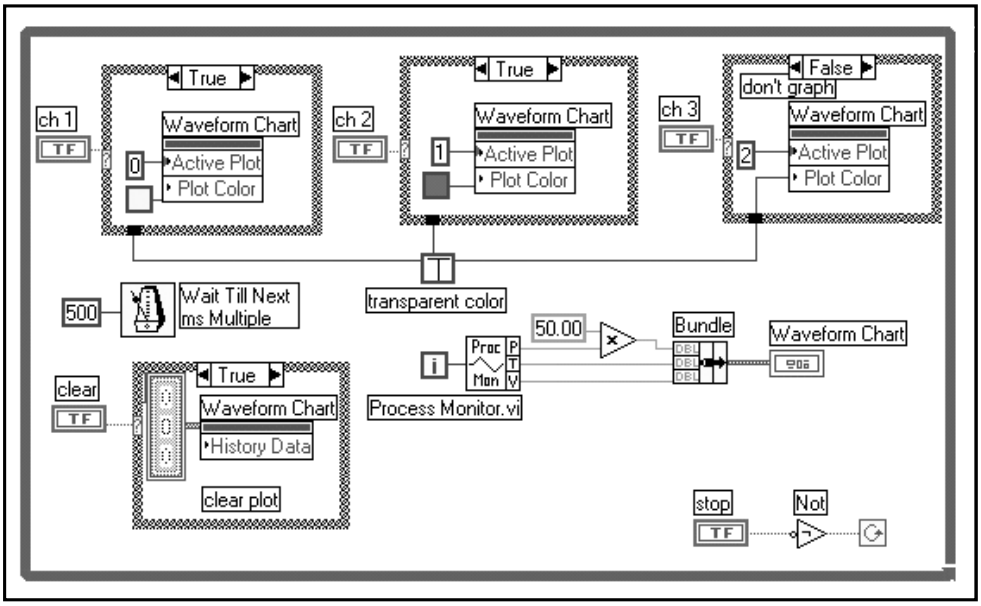



شکل ۲۹-۱۲: صفحه‌ی پانل در برنامه‌ی Attr Node-Graph.vi

همان‌طور که می‌دانید در نرم‌افزار LabVIEW خاصیت رنگ‌آمیزی فرانمایی وجود دارد. برای تنظیم گره‌ی خصوصیت مربوط به این نحوه‌ی رنگ‌آمیزی، یک کلاستر حاوی Color Box Constant ایجاد نمایید. Color Box Constant در واقع یک نوع داده‌ی عددی است که در زیرپالت Functions >> Numeric >> Additional Numeric Constants قرار دارد. جهت انتخاب رنگ، توسط ابزار Operating Tool بر روی Color Box Constant کلیک کنید. به یاد داشته باشید که رنگ‌آمیزی فرانمایی با حرف T نشان داده می‌شود.

صفحه‌ی نمودار بلوکی این برنامه در شکل ۳۰-۱۲ نشان داده شده است.






**شکل ۳۰-۱۲: صفحه‌ی نمودار بلوکی در برنامه‌ی Attr Node-Graph.vi**

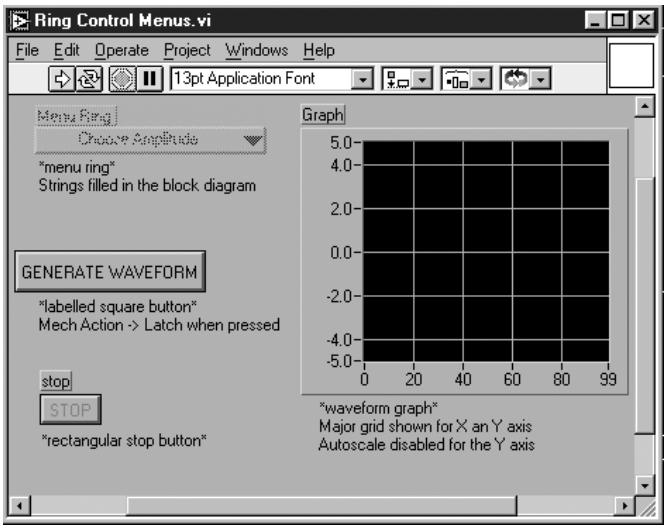
در هنگام ترسیم چندین شکل موج بر روی یک نمودار یا گراف، در هر لحظه تنها تعیین و یا ایجاد تغییر در خصوصیات یک منحنی امکان پذیر است. جهت دستیابی به منحنیهای مختلف و تمایز آنها در تعیین خصوصیات، آنها را به ترتیب (0,1,2,... ,N) شماره گذاری کنید. یکی از خواص مهم و مخصوص در مورد گراف و نمودار جهت انتخاب منحنی دلخواه برای تعیین و تغییر خصوصیات مورد نظر، Active Plot نام دارد.

جهت پاک کردن نمودار، از گزینه‌ی History Data در گره‌ی خصوصیت استفاده نمایید. سپس یک آرایه‌ی فاقد المان (Empty Array) را به این گره‌ی خصوصیت سیم‌کشی کنید.

۳- این برنامه را با عنوان Attr Nodes-Graph.vi در زیرفهرست LabVIEW\ My Activity ذخیره کنید.

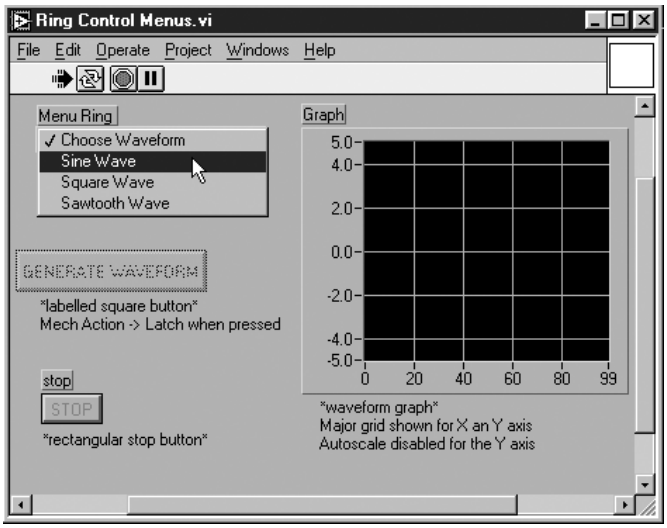
## تمرین ۵-۱۲: منوی حلقوی

منوهای حلقوی[۷] به همراه گره‌های خصوصیت برنامه‌های قدرتمند و پرکاربردی ایجاد می‌کنند. برای درک این مطلب، برنامه‌ی Ring Control Menus.vi را از مسیر LabVIEW\Exercise\Ch12.llb باز کنید.



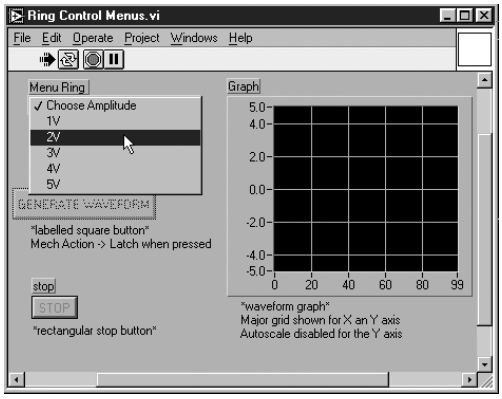

**شکل ۳۱-۱۲: صفحه‌ی پانل در برنامه‌ی Ring Control Menus.vi**

یک منوی حلقوی کرکره‌ای که در ابتدا فاقد هر گونه المان است برای سئوال کردن از کاربر در مورد نوع داده‌های خروجی در این برنامه مورد استفاده قرار می‌گیرد. پس از کلیک کردن بر روی دکمه‌ی Run، هنگامی که دکمه‌ی GENERATE WAVEFORM فشار داده شود منوی حلقوی مذکور فعال می‌شود و از کاربر می‌خواهد تا یکی از گزینه‌های موج سینوسی، مربعی یا دندانه اره‌ای را از نوار Choose Waveform انتخاب کند.




**شکل ۳۲-۱۲: منوی حلقوی موجود در صفحه‌ی پانل برنامه‌ی Ring Control Menues.vi جهت انتخاب نوع خروجی**

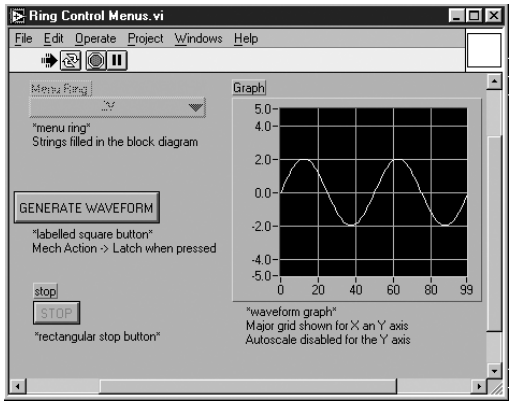
در مرحله‌ی بعد، محتویات این منو مطابق شکل ۱۲-۳۳ تغییر می کند. در این حالت کاربر می تواند دامنه‌ی خروجی را تعیین کند.




**شکل ۱۲-۳۳: منوی حلقوی موجود در صفحه‌ی پانل برنامه‌ی Ring Control Menues.vi جهت تعیین دامنه‌ی خروجی**

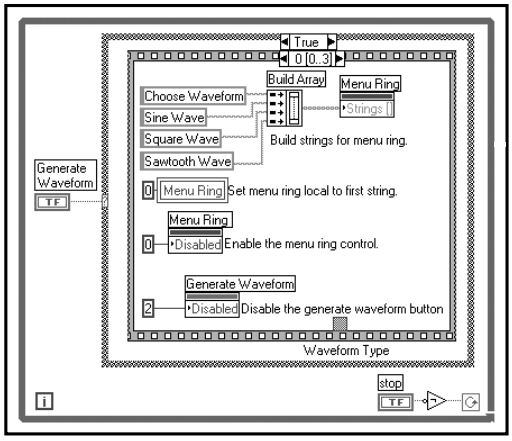
۱۲

در پایان، شکل موج مطلوب در کارت DAQ تولید می‌شود و بر روی صفحه به نمایش در می آید.




**شکل ۱۲-۳۴: شکل موج ایجاد شده در کارت DAQ**

در شکل ۱۲-۳۵ بخشی از صفحه‌ی نمودار بلوکی این برنامه نشان داده شده است.




**شکل ۱۲-۳۵: یکی از فریم‌های موجود در صفحه‌ی نمودار بلوکی برنامه‌ی Ring Control Menus.vi**

## امکانات دیگر در محیط LabVIEW

همواره و نه فقط در زمینه‌ی LabVIEW، صرف نظر کردن از ذکر یک دسته از دستورها و المان های فراموش شده مشکل است. بسیاری از دستورهایی که هنوز به ذکر آنها نپرداخته‌ایم در زیرپالت Functions >> Advanced قرار دارند.

در این بخش برخی از دستورها و توابع موجود در این زیرپالت را مورد بررسی قرار می دهیم. این دستورها به ندرت مورد استفاده قرار گرفته ولی کاملاً مفید هستند. به کارگیری این دستورها، برنامه نویسی را آسان تر می سازد.

۱۲

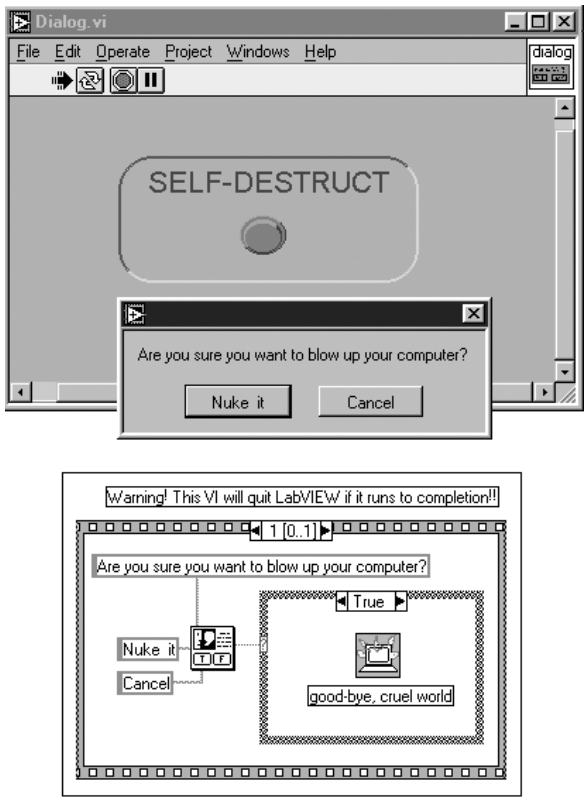
## پنجره‌های محاوره‌ای

استفاده از پنجره‌ی محاوره‌ای روشی برای برقراری ارتباط با نرم‌افزار LabVIEW و هر نرم‌افزار کاربردی دیگر است. در محیط LabVIEW ایجاد پنجره های محاوره‌ای امکان‌پذیر است. می توانید این پنجره‌ها را به گونه‌ای ایجاد کنید که حاوی پیامهای مخصوص و مورد نظر شما باشند. در اکثر این پنجره‌ها کلیدهایی نظیر OK یا Cancel جهت تأیید یا انصراف از قبول پیام دیده می شوند. در فصول قبل، جهت برقراری ارتباط و محاوره به ذکر توابع و دستورهای محاوره‌ای پرداختیم. در این بخش نیز دستورهای مذکور را بررسی می‌کنیم و علاوه بر این دستورها، اطلاعات مفید دیگری در اختیار شما قرار می دهیم.

همان گونه که قبلاً عنوان شد دستورهای محاوره‌ای در زیرپالت Functions >> Time & Dialog قرار دارند. در پنجره های محاوره‌ای نرم‌افزار LabVIEW، یک یا دو دکمه وجود دارد. در هر دو حالت می توانید پیامهای دلخواه را در پنجره‌ی محاوره‌ای وارد نمایید و برچسب دکمه ها را به دلخواه تعیین کنید. در صورت استفاده از پنجره‌ی محاوره‌ای دو دکمه‌ای، دستور مذکور یک مقدار جبری به دست می دهد که مشخص کننده‌ی آن است که کدام دکمه فشرده شده است.

در این حالت تا زمانی که کاربر با فشار دادن یکی از دو دکمه به پیام مندرج در پنجره‌ی محاوره‌ای پاسخ ندهد، LabVIEW اجرای برنامه را متوقف می سازد. تا زمانی که کاربر به پنجره‌ی محاوره‌ای پاسخ ندهد، نمی تواند هیچ عملی را از طریق ماوس یا صفحه کلید در مورد برنامه های در حال اجرا یا پنجره های باز و فعال در محیط LabVIEW انجام دهد.

به عنوان مثال فرض کنید که قصد دارید یک پنجره‌ی محاوره‌ای ایجاد نمایید تا کاربر در موارد بحرانی گزینه‌ای را تأیید کند. در مثال زیر هنگامی که کاربر دکمه‌ی SELF-DESTRUCT را فشار می دهد، یک پنجره‌ی محاوره‌ای ظاهر می‌شود و تا زمانی که کاربر بر روی یکی از دو کلید موجود در پنجره کلیک نکرده باشد، دسترسی و انجام هر گونه عملیات یا اعمال فرمان بر روی پنجره‌ی اصلی توسط ماوس یا صفحه کلید امکان ناپذیر است. پنجره‌ی پانل و بخشی از صفحه‌ی نمودار بلوکی این برنامه را در شکل ۳۶-۱۲ ملاحظه می کنید.




**شکل ۳۶-۱۲: صفحه‌ی پانل و بخشی از صفحه‌ی نمودار بلوکی در برنامه‌ی Dialog.vi**

در نرم افزار LabVIEW پنجره های محاوره‌ای دیگری نیز جهت خواندن و ذخیره کردن فایل ها وجود دارد. در مورد پنجره های مذکور و نحوه‌ی ایجاد آنها در فصل ۱۴ سخن خواهیم گفت. به یاد داشته باشید که برنامه های LabVIEW یک ارتباط گرافیکی با کاربر برقرار می کنند. بنابراین وجود دیاگرام ها می تواند باعث ازدحام و شلوغ شدن برنامه گردد. پس بهتر است که پنجره های محاوره‌ای را برای برقراری ارتباط کاربر با برنامه و تأیید یا انصراف از انجام برخی از اعمال حساس و بحرانی استفاده کنیم. همچنین به خاطر داشته باشید که در این پنجره ها تنها یک یا دو دکمه وجود دارد. برای ایجاد پنجره های محاوره‌ای پیشرفته تر که در آنها دکمه های بیشتری وجود دارد، بهتر است ابتدا یک زیربرنامه ایجاد کنید و گزینه های محاوره‌ای مورد نظر خود را در صفحه‌ی پانل آن قرار دهید. می توانید این زیربرنامه را در صورت نیاز باز نمایید و حتی نمای پنجره‌ی آن را سفارشی کنید و آن را به دلخواه خود در آورید. در فصل ۱۴ نحوه‌ی انجام این عمل را بررسی خواهیم کرد.

## دستورهای Occurrence

توابع و دستورهایی نظیر Generate Occurrence، Wait On Occurrence و Set Occurrence همگی دارای اسامی نامأنوس و ناشناخته‌ای هستند و شاید به همین دلیل است که کمتر کسی از این دستورها استفاده می کند. این دستورها ابزارهای قدرتمندی در برنامه نویسی محسوب می شوند.

به کمک این دستورها و بدون استفاده از حلقه های While یا متغیرهای سراسری، می توانید در بخشی از

صفحه‌ی نمودار بلوکی تأخیر ایجاد کنید. در صورتی که برای ایجاد تأخیر در برخی از بخشهای صفحه‌ی نمودار بلوکی از حلقه های متعددی استفاده نمایید، مسلماً بخشی از زمان، صرف بررسی و مقایسه نمودن مقادیر ابتدایی و مقادیر حاصله در حلقه ها می شود. بنابراین به کارگیری این دستورها، زمان لازم جهت بررسی این مقادیر را کاهش می‌دهد و بدین ترتیب سرعت اجرای برنامه افزایش می یابد.

دستورهای مذکور در زیرپالت Functions >> Advanced >> Synchronization >> Occurrences قرار دارند. به دلیل اینکه دستورهای مذکور دارای دو وضعیت می باشند می توان آنها را شبیه به المان های جبری در نظر گرفت. در ادامه به بررسی هر یک از دستورهای موجود در این زیرپالت می پردازیم:

Generate Occurence

دستور Generate Occurrence یک شماره‌ی رویداد یا Occurrence refnum ایجاد می کند. این المان، عددی مجازی است که LabVIEW جهت تعقیب و انجام دستورهای Occurrence به کار می‌برد و آن را به دستورهای دیگر انتقال می دهد.

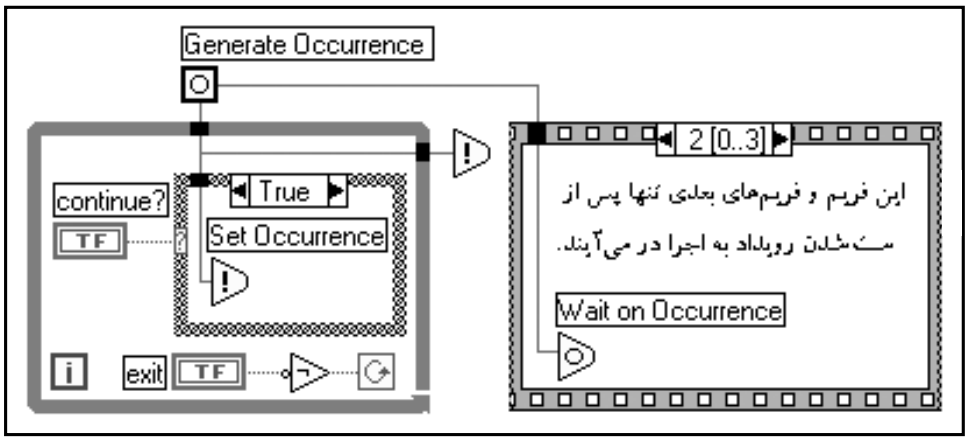
Wait On Occurence

دستور Wait On Occurrence باعث می شود اجرای بخشی از صفحه‌ی نمودار بلوکی تا زمان وقوع رویداد یا برقراری شرط متوقف گردد. در صورت تمایل می توانید از این دستور یک خطای timeout استخراج کنید. بدین ترتیب در صورت عدم وقوع شرط یا رویداد مورد نظر، در خروجی این دستور خطای timeout ایجاد می گردد.

Set Occurence

دستور Set Occurrence شرط یا رویداد مورد نظر را تریگر می کند. بدین معنی که وضعیت رویداد انتقال داده شده به آن را تغییر می دهد. هنگامی که این دستور اجرا می گردد، دیگر دستورهای Wait On Occurtence که به همان شماره‌ی رویداد مرتبط شده اند آن را تشخیص می‌دهند و بدون توجه به اینکه در چه بخشی از صفحه‌ی نمودار بلوکی قرار گرفته اند به ساختاری که در آن قرار دارند اجازه‌ی اجرا می دهند.

دستورهای Occurrence نظیر ساختارهای شرطی یا ترتیبی تنها بر روی قسمتی از صفحه‌ی نمودار بلوکی که در آن قرار گرفته اند تاثیرگذارند. در مثال بعدی در صورتی که المان جبری continue? در وضعیت True قرار گیرد (فرض کنید که این عمل توسط فشردن یک دکمه درصفحه‌ی پانل انجام می گیرد) یک رویداد ست می شود. بدین ترتیب هنگامی که ساختار ترتیبی نشان داده شده در سمت راست شکل ۱۲-۳۷ به فریم شماره‌ی 2 می رسد، تا زمانی که این اتفاق ست نشود، مرحله یا فریم بعدی انجام نمی شود. بنابراین الزاماً این رویداد تنها به آن بخش ساختار ترتیبی اجازه می دهد تا به صورت موازی با حلقه‌ی While به اجرا در آید. حال آنکه بخشهای دیگر یعنی فریم شماره‌ی 2 به بعد باید تا زمانی که حلقه‌ی While به پایان رسیده یا دکمه‌ی continue? فشار داده شود منتظر بمانند. فشردن دکمه‌ی continue? در این برنامه به معنی قرار دادن این المان جبری در وضعیت True و در نتیجه برقراری شرط یا وقوع رویداد مذکور و به همراه آن ادامه یافتن اجرای فریم های بعدی می باشد. این مطلب در شکل ۱۲-۳۷ نشان داده شده است.




**شکل ۳۷-۱۲: نحوه‌ی به کارگیری دستورهای Occurences**

۱۲

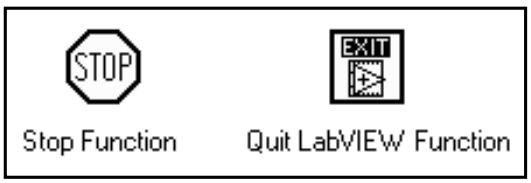
توجه کنید که در شکل ۳۷-۱۲ هر دو دستور Wait On Occurrence و Set Occurrence ورودی خود را از دستور Generate Occurrence دریافت می کنند.

به کارگیری دستور Generate Occurrence و ارسال ورودی به دو دستور دیگر الزامی است. زیرا در صورت وجود چند رویداد، دو دستور Wait On Occurrence و Set Occurrence باید توسط ورودی دریافت شده، از نوع و چگونگی رویداد مورد نظر اطلاع حاصل کنند. به عبارت دیگر دستور Generate Occurrence در واقع به غیر از یک شماره‌ی رویداد که توسط دو دستور دیگر مورد استفاده قرار می گیرد، پارامتر دیگری را تولید نمی کند. نکته‌ی مهم دیگر در شکل ۳۷-۱۲ نحوه‌ی سیم کشی به دومین دستور Set Occurrence است به گونه‌ای که پس از اتمام حلقه‌ی While به اجرا در آید. حال فکر کنید در صورتی که این عمل را انجام نمی دادیم چه اتفاقی ممکن بود روی دهد.

در اغلب موارد می توانید نتایج حاصل از به کارگیری دستورهای Occurrence را با استفاده از متغیرهای محلی نیز به دست آورید. مزیت استفاده از این دستورها، صرفه جویی در حافظه و زمان برنامه نویسی است.

## متوقف نمودن برنامه

در نرم افزار LabVIEW دو دستور Stop و Quit LabVIEW جهت متوقف ساختن اجرای برنامه تعبیه شده‌اند. این دو دستور در زیرپالت Functions >> Application Control قرار دارند.




**شکل ۳۸-۱۲**

دستور Stop دارای یک ورودی جبری است و در حالتی که این پارامتر در حالت True باشد اجرای برنامه و تمامی زیربرنامه های آن متوقف می گردد. تغیر وضعیت این عبارت جبری به حالت True نظیر فشرده شدن دکمه‌ی Abort در نوار ابزار است. در صورتی که ورودی دستور Quit LabVIEW نیز در وضعیت True باشد عملکردی شبیه به دستور Stop دارد.

در هنگام استفاده از دستورهای مذکور کاملاً مراقب باشید. به یاد داشته باشید که استفاده نمودن از این دو دستور روش مناسبی جهت متوقف ساختن برنامه نیست. سعی کنید تا جهت متوقف ساختن و خروج از برنامه از روشهای معقول و بهتری نظیر گنجاندن دکمه های Stop در برنامه استفاده نمایید.

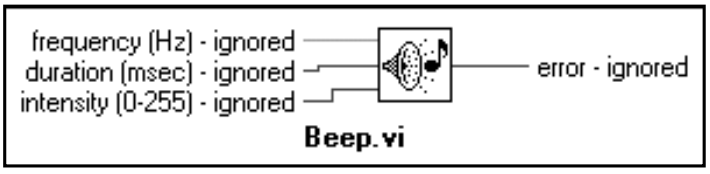
قبل از اجرای دستور Stop از تکمیل وظایف و برنامه‌های خود اطمینان حاصل کنید. در غیر این صورت ممکن است خطاهای I/O و یا رفتارهای غیر منتظره ایجاد شوند و یا نتایج حاصله از دست بروند.

۱۲

## نحوه‌ی تولید صدا

یکی از ویژگیهای جالب توجه در نرم افزار LabVIEW قابلیت تولید صدا در برنامه هاست. تولید یک آلارم در مواردی که اپراتور قادر نیست در خلال برخی از آزمایشها و فرآیندها به طور پیوسته به صفحه نمایش نگاه کند بسیار مفید و کاربردی است. از طرفی حتی صدای یک بوق هشداردهنده ممکن است برای آگاه کردن اپراتور آزاردهنده باشد. بنابراین در هنگام استفاده از این قابلیت مراقب باشید که از قابلیت تولید صدا در چه محلی و چند دفعه استفاده می کنید.

LabVIEW به شما اجازه می دهد به کمک دستور Beep به بوق هشداردهنده‌ی سیستم عامل خود دسترسی پیدا کنید. دستور Beep در زیرپالت Functions >> Advanced واقع شده است.




**شکل ۳۹-۱۲: آیکن دستور Beep به همراه اتصالات آن**

در محیط Mac می توانید فرکانس، شدت و طول زمان پخش صدا را تعیین کنید.

در محیط Windows از ورودیهای نشان داده شده در شکل ۳۹-۱۲ چشم‌پوشی شده است و بوق هشداردهنده‌ی سیستم، مورد استفاده قرار می گیرد. فراخوانی فایل‌های صوتی با پسوند WAV. در محیط LabVIEW امکان‌پذیر است. برای درک نحوه‌ی تولید صدا در نرم‌افزار LabVIEW، برنامه‌ی Play Sound.vi را از مسیر LabVIEW\Examples\Dll\sound\plysnd.llb اجرا کنید. این برنامه کدهای خارجی را جهت استفاده از سیستم مولتی مدیا به کار می برد.

## فراخوانی کدهای خارجی از زبانهای برنامه نویسی دیگر

در صورتی که قصد داشته باشید کدهای نوشته شده در یکی از زبانهای برنامه نویسی متداول و رایج نظیر C، Pascal، FORTRAN و BASIC را در محیط LabVIEW به کار ببرید چه می کنید؟

LabVIEW امکاناتی جهت برقراری ارتباط با کدهای خارجی در دیگر زبانهای برنامه نویسی در اختیار کاربر قرار می دهد. اگر قصد دارید تمامی برنامه های خود را به زبان C یا $C^{++}$ بنویسید باید از نرم افزار Lab Windows/CVI استفاده کنید. این نرم افزار یک محیط برنامه نویسی است که بسیار شبیه به LabVIEW می باشد. تنها تفاوت بین نرم افزار LabVIEW و بسته‌ی نرم افزاری مذکور آن است که در این بسته‌ی نرم افزاری کدهای نوشته شده در زبان C، با صفحه‌ی نمودار بلوکی جایگزین می‌شوند و تنها کافی است از کدهای زبانهای برنامه نویسی متداول متنی استفاده نمایید.

۱۲

در صورتی که با روش نوشتن کدهای برنامه‌نویسی در زبانهای دیگر نظیر C آشنا نیستید و یا به برقراری ارتباط با کدهای خارجی در نرم‌افزار LabVIEW نیاز ندارید، می توانید از مطالعه‌ی این بخش صرف نظر کنید.

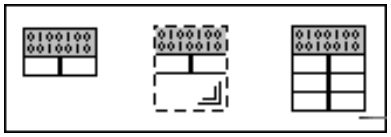
برای فراخوانی کدهای خارجی در محیط Windows دو راه در اختیار دارید: می توانید با به کارگیری دستور Call Library یک DLL را فراخوانی نمایید یا اینکه می توانید مستقیماً با استفاده از CIN، کدهای اجرایی را که خارج از محیط LabVIEW ایجاد شده‌اند فراخوانی کنید. هر دو دستور مذکور در زیرپالت Functions >> Advanced قرار دارند. در دو محیط Windows 3.1 و Win 95 به ترتیب می توانید DLL های ۱۶ و ۳۲ بیتی را فراخوانی کنید. LabVIEW به شما اجازه می دهد تا با تبدیل دستورهای زبان C به زیربرنامه ها، کدهای خارجی را از صفحه‌ی پانل در Lab Windows/CVI فراخوانی کنید.

در محیط Mac جهت فراخوانی کدهای خارجی اجرایی از CIN استفاده نمایید.

برای استفاده از CIN به روش زیر عمل کنید:

۱- آیکن CIN را از زیرپالت Functions >> Advanced انتخاب کنید و آن را بر روی صفحه‌ی نمودار بلوکی قرار دهید.

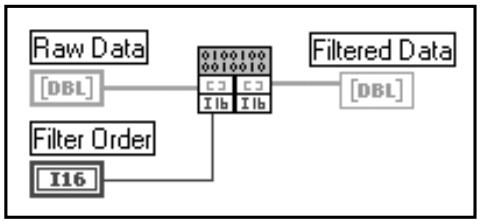
۲- آیکن CIN چند ترمینال جهت عبور و انتقال ورودیها و خروجیها دارد. به صورت پیش فرض برای هر CIN تنها یک زوج ترمینال در نظر گرفته شده است. در صورت نیاز به ورودی و خروجیهای بیشتر، مطابق شکل ۴۰-۱۲ اندازه‌ی این آیکن را توسط ابزار Positioning Tool افزایش دهید.



شکل ۴۰-۱۲: نحوه‌ی تغییر اندازه‌ی آیکن CIN

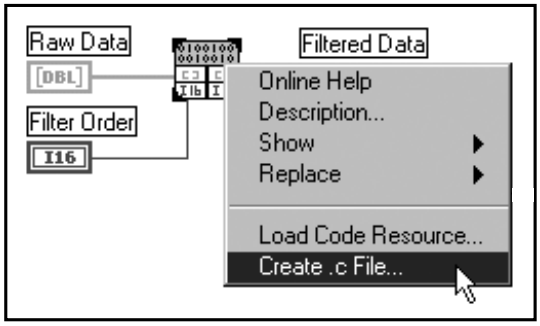
به صورت پیش فرض هر جفت از ترمینال های موجود در CIN به صورت یک زوج ورودی و خروجی تعریف شده‌اند. به عبارت دیگر ترمینال سمت چپ و راست به ترتیب ورودی و خروجی متناظر با یکدیگر هستند. در صورتی که تعداد خروجیها از تعداد ورودیهای یک دستور خاص تجاوز نمایند یا اینکه در یک دستور خاص، ترمینال ورودی وجود نداشته باشد، منوی کرکره‌ای را بر روی ترمینال مورد نظر باز کنید و با انتخاب گزینه‌ی Output Only ورودی متناظر با آن را در نظر نگیرید.

۳- ورودیها و خروجیها را به آیکن CIN سیم کشی کنید. در هنگام سیم کشی به ترمینال CIN به شرطی که پارامتر سیم کشی شده به ترمینال های CIN با نوع داده‌ی فراخوانی شده از زبان C یا هر زبان برنامه نویسی دیگر همخوانی داشته باشد، اتصال هر نوع داده به ترمینال های ورودی و خروجی امکان‌پذیر است. ترتیب اتصال جفت ترمینال های CIN متناظر با ترتیب قرار گرفتن پارامترهای موجود در کدهای خارجی است. در شکل ۴۱-۱۲ یک CIN را فراخوانی می کنیم که یک آرایه‌ی حاوی داده‌های خام با برچسب Raw Data را به عنوان ورودی دریافت می کند و داده‌ها را پس از فیلتر کردن در خروجی آرایه‌ای Filtered Data ظاهر می سازد. توجه داشته باشید که چگونه ترمینال های سیم کشی شده، نوع داده هایی را که انتقال می دهند مشخص می کنند.




شکل ۴۱-۱۲: نحوه‌ی اتصال ترمینال های آیکن CIN

۴- مطابق شکل ۴۲-۱۲ با انتخاب گزینه‌ی ...Create .c File از منوی کرکره‌ای آیکن CIN، یک فایل با پسوند .c ایجاد کنید. این فایل که توسط LabVIEW ایجاد می شود در قالب زبان برنامه نویسی C است و می توانید کدهای زبان C را در آن بنویسید.




شکل ۴۲-۱۲

۵- کدهای CIN را کامپایل کنید. این مرحله را با توجه به نوع سیستم، کامپایلر و ابزار مورد استفاده و با دقت کافی انجام دهید. در این مرحله ابتدا توسط یک کامپایلر مناسب که LabVIEW از آن پشتیبانی می کند داده ها را کامپایل کنید، سپس با استفاده از نرم افزارهای جانبی LabVIEW، کدها را به گونه ای تغییر دهید که در LabVIEW قابل استفاده باشند.

۶- با انتخاب گزینه‌ی ...Load Code Resource از منوی کرکره ای آیکن CIN، کدها را از حافظه بارگذاری کنید. پس از کامپایل نمودن کدها، پسوند lsb. را برای فایل ایجاد شده در نظر بگیرید. وقتی که تمامی این مراحل با دقت و موفقیت به اتمام رسید کد زبان C را به همان روش فراخوانی یک زیربرنامه صدا کنید. با این تفاوت که فراخوانی CIN ها به طور همزمان انجام می گیرد. بدین معنی که برخلاف وظایف و دیگر اعمال در LabVIEW، زمان پردازش را با وظایف دیگر تقسیم نمی کنند. به عنوان مثال اگر در صفحه‌ی نمودار بلوکی برنامه، یک حلقه‌ی For و یک CIN وجود داشته باشند، حلقه‌ی For پس از اتمام اجرای CIN به اجرا در می آید. این نکته ای است که در مواردی که پردازنده در حال انجام چند برنامه می‌باشد و زمان‌بندی از اهمیت خاصی برخوردار است باید مد نظر قرار گیرد.

۱۲

نکته‌ی مهم و قابل توجه در مورد اکثر CINها آن است که انتقال آنها از یک سیستم به سیستم های دیگر امکان پذیر نیست. به عنوان مثال اگر کدهای LabVIEW را توسط CIN در یک سیستم Mac کامپایل نمایید و سپس سعی کنید تا آنها را تحت محیط Windows اجرا کنید، تلاش شما بی نتیجه است. در این حالت باید مجدداً یک CIN مطابق با سیستم مورد نظر ایجاد کنید. کامپایلرهای سازگار با LabVIEW[۸] در زبان برنامه نویسی C عبارتند از:

**جدول ۱-۱۲: کامپایلرهای سازگار با LabVIEW در زبان برنامه نویسی C**

| | |
|---|---|
| Windows 3.1 | Watcom C |
| Windows 95/NT | Microsoft Visual C++, Microsoft Win32 SDK C/C++ compiler |
| Mac | THINK C (ver. 5 or above), Symatnec C++ (ver. 8 or above), Metrowerks CodeWarrior, MPW from Apple |
| Solaris | Sun ANSI C compiler |
| HP-UX | HP-UX C/ANSI C compiler |

مسائل پیچیده‌ی بسیاری در مورد برقراری ارتباط با کدهای خارجی در LabVIEW وجود دارد. به دلیل اینکه اکثر آنها به سرعت پردازنده، سیستم عامل و کامپایلر استفاده شده بستگی دارند، در بحث پیرامون CIN یا فراخوانی DLL، بیش از این فراتر نخواهیم رفت. جهت کسب اطلاعات بیشتر در مورد فراخوانی کدهای خارجی در محیط برنامه نویسی LabVIEW، مطالب مندرج در فایل Cirmbook.pdf << Manuals را از لوح فشرده‌ی همراه این کتاب مطالعه کنید.

## تبدیل داده‌ها از نوعی به نوع دیگر

آیا مبحث Polymorphism را که در فصول ابتدایی این مجموعه مورد بررسی قرار گرفت به یاد دارید؟ این ویژگی یکی از مهمترین و برجسته ترین قابلیتهای نرم افزار LabVIEW است که به شما اجازه می دهد تا در بسیاری

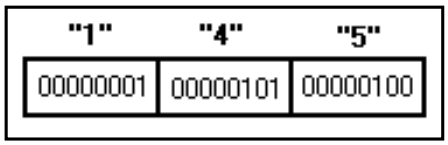
۸- کامپایلرهای مندرج در این جدول، کامپایلرهای موجود در زمان چاپ و انتشار این کتاب هستند و ممکن است تعداد آنها در آینده افزایش یابد.

از دستورها و توابع بدون در نظر گرفتن تفاوت بین نوع داده‌ها، انواع مختلف داده‌ها را با یکدیگر استفاده و ترکیب کنید. در صورتی که قصد داشته باشید این عمل را در کامپایلرهای دیگر به غیر از LabVIEW انجام دهید (به عنوان مثال یک عدد ثابت را با یک آرایه جمع کنید)این فرآیند برای کامپایلر تعریف نشده است. هنگامی که LabVIEW در یک یا چند دستور با ورودیهای از نوع متفاوت اما سازگار با یکدیگر مواجه شود، تبدیلات داخلی داده‌ها را به طور خودکار انجام می دهد.

اگر سعی دارید برنامه‌ای بنویسید که به کمک آن به کنترل دستگاهها بپردازید و بین برنامه‌ها ارتباط برقرار نماید و یا اینکه اطلاعات را از طریق شبکه انتقال دهد بهتر است از داده‌های رشته‌ای استفاده کنید. در اغلب موارد، حالتی پیش می آید که لازم است داده‌های عددی خود نظیر یک آرایه‌ی حاوی اعداد اعشاری را به یک رشته تبدیل کنید. این مطلب را به اختصار در فصل ۹ مورد بحث و بررسی قرار دادیم و خاطر نشان کردیم که تشخیص و تمییز دو رشته‌ی ASCII و باینری بسیار حائز اهمیت است.

۱۲

همان گونه که می دانید در روش نمایش ASCII، برای نمایش هر کاراکتر از ۸ بیت یا ۱ بایت استفاده می شود. با توجه به این اصل عدد 145 در تبدیل به کد ASCII ابتدا به سه کاراکتر مجزای 5,4,1 شکسته می‌شود و در مرحله‌ی بعد معادل هر یک از این کاراکترها منظور می گردد. در مواردی نظیر آرایه‌ها که در آنها تعداد اعداد زیاد است جهت جدا کردن اعداد و کاراکترها ممکن است از یک کاراکتر واسطه نظیر Space نیز استفاده شود. این نحوه‌ی نمایش رشته برای اعداد، در مورد کنترل دستگاهها در GPIB بسیار متداول است. در شکل ۴۳-۱۲ نحوه‌ی نمایش کاراکترها به صورت رشته نشان داده شده است.




شکل ۴۳-۱۲

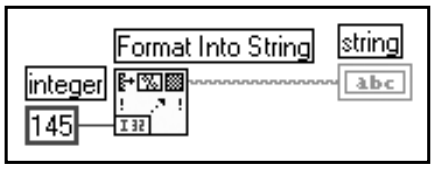
رشته های باینری ممکن است در مرحله‌ی اول شما را فریب دهند. زیرا پس از خواندن آنها به عنوان کاراکتر ASCII، نمی توانید تشخیص دهید که چه داده‌ای را نمایش می دهند. در این روش، جهت نمایش هر عدد از الگوی باینری آن استفاده می شود. بنابراین عدد 145 با طرز نمایش به صورت ۸بیت یا نماد I8 تنها به صورت یک بایت منفرد در رشته‌ی باینری است که تصادفاً با کاراکتر «ë» در کد ASCII متناظر است. ما نیز اذعان داریم که همه‌ی افراد قدرت تشخیص این عدد را ندارند و نمی توانند به راحتی تشخیص دهند که به عنوان مثال عدد 145 بدین صورت نمایش داده می شود. رشته های باینری در مواردی که به استفاده از حداقل حافظه نیاز دارید بسیار کاربردی و مفید هستند. در این فرآیند، تبدیل داده‌ها به رشته‌ی باینری سریعتر انجام می‌گیرد و مسلماً حجم کمتری از حافظه اشغال می‌شود.

شکل ۴۴-۱۲

در نهایت تمامی داده ها در حافظه‌ی رایانه به صورت اعداد باینری ذخیره می شوند. حال یک سؤال مطرح است که: LabVIEW چگونه نوع داده‌ی ذخیره شده اعم از رشته‌ای، جبری، اعداد اعشاری، آرایه‌ای یا عدد صحیح را تشخیص می دهد؟

هر یک از داده های ذخیره شده در LabVIEW شامل دو بخش است. بخش اول شامل داده می باشد و بخش دوم مبین نوع داده[9] است. بخش دوم یا مبین نوع داده یک آرایه از نوع I16 است که نوع داده‌ی بخش اول اعم از اعداد صحیح، رشته ای، اعشاری، جبری و... را تعیین می کند. این بخش حاوی اطلاعاتی در مورد طول، نوع و ساختار داده است. این اطلاعات به صورت کدهایی در بخش مبین داده سازماندهی می شوند. هر یک از انواع داده ها دارای کد بخصوصی است.

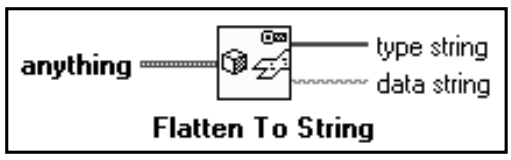
در تبدیل داده ها از نوعی به نوع دیگر (نظیر تبدیل یک عدد به یک رشته‌ی دهدهی) بخش دوم یا مبین نوع داده تغیر می کند و بخش اول نیز به طریقی اصلاح می شود. در اکثر تبدیلات، معمولاً یک عدد به رشته‌ی ASCII تبدیل می گردد. در شکل ۴۵-۱۲ تبدیل مذکور نشان داده شده است. این عمل با استفاده از دستور Format Into String انجام می گیرد.




**شکل ۴۵-۱۲: نمونه‌ای از کاربرد دستور Format Into String**

عناوین و مطالب موجود در ادامه‌ی این فصل ممکن است برای مبتدیان بسیار گمراه کننده باشد. اگر به تبدیل رشته‌های باینری نیاز ندارید از مطالعه‌ی ادامه‌ی این فصل صرف نظر نمایید.

در برخی موارد ممکن است قصد داشته باشید داده ها را به رشته های باینری تبدیل کنید. در نرم افزار LabVIEW می توان با کمک دستور Flatten To String داده ها را به رشته های باینری تبدیل نمود. این دستور در زیرپالت Functions >> Advanced >> Data Manipulation قرار دارد.



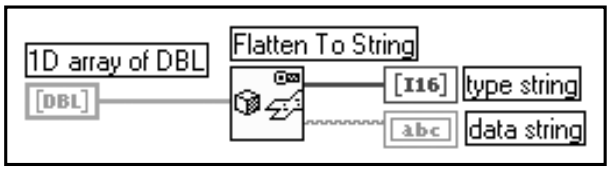

**شکل ۴۶-۱۲: آیکن دستور Flatten To String به همراه اتصالات آن**

همان گونه که در شکل ۴۶-۱۲ ملاحظه می کنید ورودی دستور Flatten to String ممکن است هر نوع داده‌ای اعم از انواع داده های پیچیده نظیر کلاستر باشد. این دستور دارای دو خروجی است. خروجی data string نمایش دهنده‌ی داده است. خروجی دیگر یعنی type string در واقع یک داده‌ای رشته‌ای نیست و یک آرایه با

9- Data Type Descripter

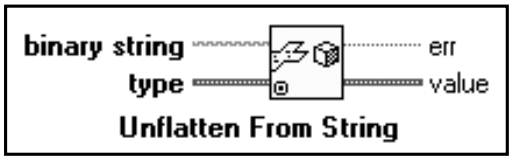
فرمت I16 می‌باشد و به دلیل هماهنگی در به کارگیری اصطلاحات، به رشته موسوم شده است. این خروجی، اطلاعاتی در مورد نوع داده در بر دارد و در حقیقت مبین نوع داده است.

رشته‌ی باینری حاصل خروجی دستور Flatten To String نه تنها شامل داده‌ها به فرم فشرده است بلکه حاوی ۴ بایت اطلاعات عنوان[10] برای داده می باشد. این اطلاعات که در ابتدای رشته‌ی باینری قرار دارند شامل اطلاعاتی نظیر طول، نوع، ساختار داده و... است. به عنوان مثال در شکل ۴۷-۱۲ ملاحظه می کنید که یک آرایه‌ی حاوی اعداد DBL، به عنوان ورودی دستور Flatten To String اعمال شده است. خروجی type string که همواره یک آرایه از نوع I16 می باشد حاوی اطلاعات عنوان است.




**شکل ۴۷-۱۲: نمونه‌ای از کاربرد دستور Flatten To String**

به کارگیری این رشته‌های باینری برای مبتدیان گمراه کننده است. در اکثر مواقع نیازی نیست تا داده‌های یک رشته را مستقیماً بررسی نمایید و یا بر روی آن عملیاتی انجام دهید. بخصوص در مواقعی که نیاز به صرفه‌جویی در حافظه دارید، تنها کافی است این رشته‌ها را به یک فایل، شبکه، ابزار اندازه‌گیری و... انتقال دهید. سپس برای خواندن و انجام هرگونه عملیات بر روی داده‌ها مجدداً آنها را از نوع رشته‌ای به نوع دلخواه تبدیل کنید.

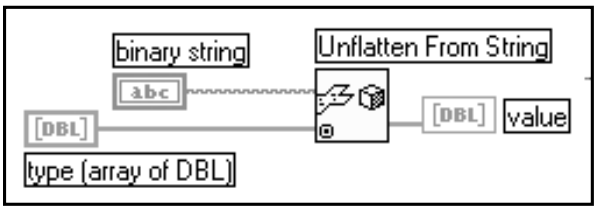



**شکل ۴۸-۱۲: آیکن دستور Unflatten From String به همراه اتصالات آن**

برای برگرداندن رشته‌های باینری، از دستور Unflatten From String استفاده می شود. همان گونه که در شکل ۴۸-۱۲ ملاحظه می کنید این دستور دارای دو ورودی و دو خروجی است. ورودی type یک نوع داده‌ی فرضی در LabVIEW و از همان نوع نمایش ورودی binary string می باشد. خروجی value در حقیقت حاوی همان داده‌های ورودی binary string است که به نوع مشخص شده در ورودی type درآمده‌اند. در صورتی که تبدیل مذکور به درستی انجام نگرفته باشد خروجی err به حالت True تغییر وضعیت می‌دهد و بیانگر آن است که در انجام تبدیل مورد نظر خطایی رخ داده است.

در شکل ۴۹-۱۲ نحوه‌ی برگرداندن داده‌های رشته‌ی باینری به آرایه‌ی DBL نشان داده شده است. در این شکل ملاحظه می کنید که به دلیل انجام دقیق و موفقیت آمیز تبدیل داده‌ها، خروجی err غیر فعال می‌شود و در وضعیت False قرار می گیرد.

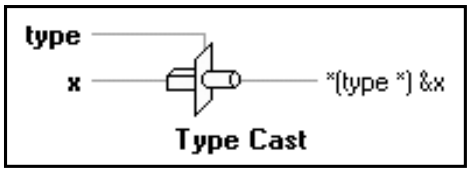
10- Header




**شکل ۴۹-۱۲: نمونه‌ای از کاربرد دستو ر Unflatten From String**

در شکل ۴۹-۱۲ ملاحظه می کنید که در ورودی دستور Unflatten Form String دو ورودی binary string و یک آرایه‌ی یک بعدی DBL جهت تعیین نوع داده به دستور سیم کشی شده‌اند و بدین ترتیب در خروجی، آرایه‌ی DBL اولیه به دست آمده است. توجه داشته باشید که به کمک دو دستور مذکور، چگونه داده های باینری را به رشته تبدیل نمودیم و سپس مجدداً آنها را با انجام عملیات معکوس به دست آوردیم و در این فرآیند حتی نیاز به جداسازی یا آشکارسازی داده های باینری احساس نشد.

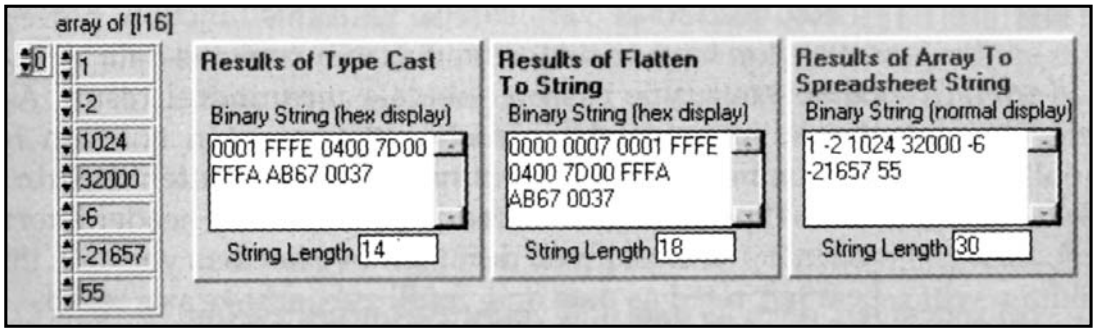
۱۲




**شکل ۵۰-۱۲: آیکن دستور Type Cast به همراه اتصالات آن**

برای انجام تبدیلات سریع داده ها دستور دیگری در LabVIEW در نظر گرفته شده است. دستور Type Cast به شما اجازه می دهد تا تنها مبین نوع داده را بدون انجام هرگونه تغییراتی در داده عوض کنید. در این دستور هیچ تغییری در داده ایجاد نمی شود. در ورودی این دستور می توانید هر یک از انواع داده اعم از رشته‌ای، جبری، عددی، کلاستر و آرایه‌ای را استفاده نمایید و در خروجی نیز هر نوع داده‌ی دیگر را دریافت کنید. مزیت به کارگیری این دستور صرفه جویی در حافظه است. زیرا به هنگام اجرای این دستور نسخه‌ی کپی دیگری از داده ها در حافظه ایجاد نمی شود.

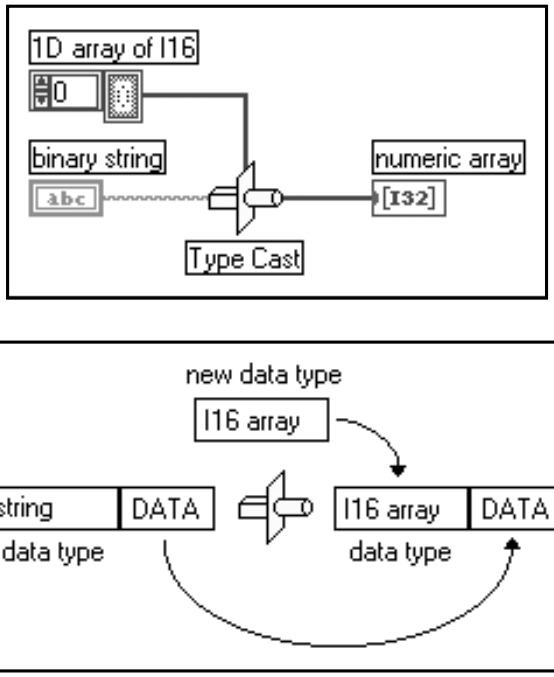
در شکل ۵۱-۱۲ نتایج تبدیل یک آرایه به رشته را در سه روش مختلف ملاحظه می کنید.




**شکل ۵۱-۱۲: نتیجه‌ی به‌دست آمده از تبدیل داده‌ها در سه روش متفاوت**

کاربرد عمومی دستور Type Cast در شکل ۵۲-۱۲ نشان داده شده است. به خصوص در مواردی که برخی از دستگاههای اندازه گیری، داده های قرائت شده را به صورت رشته های باینری در خروجی خود تبدیل می‌نمایند و نیاز دارند برای انجام عملیات، آنها را به آرایه‌ای حاوی داده های عددی تبدیل کنند، کاربرد این دستور بسیار زیاد است. در این روش فرض بر این است که رشته‌ی باینری به صورت یک آرایه‌ی حاوی اعداد صحیح با فرمت I16 نمایش داده می شود.



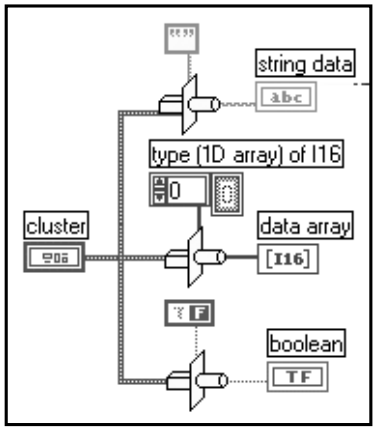



**شکل ۵۲-۱۲: نحوه‌ی عملکرد دستور Type Cast در تبدیل داده‌ها**

عملکرد دستور Type Cast در تبدیل نوع داده‌ها از اعداد اسکالر یا آرایه‌ی حاوی اعداد اسکالر به یک رشته، نظیر عملکرد دستور Flatten To String است. یک تفاوت آشکار و بدیهی بین به کارگیری دو دستور و دو تبدیل مذکور آن است که برخلاف دستورهای Flatten و Unflatten که ۴ بایت عنوان ایجاد می کنند، در فرآیند تبدیل به وسیله‌ی دستور Type Cast هیچ عنوانی (Header) ایجاد نمی‌شود. ورودی type در دستور Type Cast دقیقاً یک پارامتر فرضی است که تنها برای تعریف نوع داده به کار می‌رود و هرگونه داده‌ی حقیقی در این ورودی نادیده فرض می شود.

در هنگام استفاده از دستور Type Cast حتماً در مورد نوع داده‌ای که نمایش داده می‌شود اطلاعات کاملی در اختیار داشته باشید. زیرا در انجام تبدیل به این روش، وجود خطا بررسی نمی‌شود. ترتیب قرار گرفتن بایت ها به نوع سیستم بستگی دارد. بنابراین حتماً از چگونگی ترتیب قرار گرفتن داده‌های خود اطلاع حاصل کنید.

برای بررسی نوع نتایج به‌دست آمده سعی کنید به کمک دستور Type Cast چند تمرین انجام دهید. در شکل ۵۳-۱۲ نمونه‌ای از کاربرد این دستور را ملاحظه می‌کنید.




**شکل ۵۳-۱۲: نمونه‌ای از کاربرد دستور Type Cast**

۱۲

## خلاصه

در این فصل به بررسی مفاهیمی نظیر متغیرهای محلی و سراسری و گره‌های خصوصیت پرداختیم. همچنین برخی از قابلیتهای پیشرفته نظیر دستورهای Occurrences، پنجره‌های محاوره‌ای، ایجاد صدا، فراخوانی کدهای خارجی، تبدیل رشته‌های باینری و تبدیل نوع داده‌ها را مورد بررسی قرار دادیم. قدرت و کارایی این دستورها به شما اجازه می‌دهد تا برنامه‌نویسی را با سرعت و علاقه‌ی بیشتری انجام دهید.

به کمک متغیرهای محلی می‌توان یک یا چند کپی از المان‌های موجود در صفحه‌ی پانل را در صفحه‌ی نمودار بلوکی ایجاد نمود، سپس عمل خواندن یا نوشتن را از طریق کپی‌های ایجاد شده در نقاط مختلف صفحه‌ی نمودار بلوکی انجام داد. متغیرهای سراسری نیز شبیه به متغیرهای محلی عمل می‌کنند با این تفاوت که این متغیرها داده‌های خود را مستقل از برنامه ذخیره می‌نمایند و به کاربر اجازه می‌دهند تا متغیرها را بدون سیم‌کشی آنها به یکدیگر بین چند برنامه‌ی مجزا تقسیم کند. به کارگیری متغیرهای محلی در کنترل حلقه‌های موازی یا به روز رسانی یک المان موجود در صفحه‌ی پانل از چند نقطه در صفحه‌ی نمودار بلوکی بسیار مفید و کاربردی است. متغیر سراسری نیز یک ساختار قدرتمند و مفید است. در هنگام به کارگیری متغیر سراسری بسیار مراقب باشید. زیرا ممکن است مسائل و مشکلات غیرمنتظره‌ای ظهور کند.

به کمک گره‌های خصوصیت می‌توانید بر عملکرد و ظاهر المان‌های کنترل و نشان‌دهنده، کنترل بیشتری داشته باشید. از طریق این گره‌ها می‌توانید خصوصیات المان‌ها را تغیر دهید. در مورد برخی از المان‌ها تعداد این خصوصیات بسیار زیاد است.

در نرم‌افزار LabVIEW با به کارگیری CIN می‌توانید از کدهای خارجی در زبان C استفاده کنید. برای فراخوانی کدهای خارجی در محیط Windows می‌توانید به طور مستقیم از DLL‌ها استفاده کنید.

دستورهای Flatten To String، Unflatten From String و Type Cast جهت تبدیل انواع داده‌ها به یکدیگر به کار برده می‌شوند.

LabVIEW

# ۱۳

# قابلیتهای پیشرفته در نرم‌افزار LabVIEW

## در این فصل با موارد زیر آشنا می‌شوید:

- روش به‌کارگیری گزینه‌های Preferences جهت سفارشی کردن محیط برنامه‌نویسی LabVIEW
- گزینه‌های موجود در منوی VI Setup
- نحوه‌ی نمایش اعداد در مبناهای مختلف و انتساب واحد به متغیرهای عددی
- روش ایجاد یک زیربرنامه از بخشی از صفحه‌ی نمودار بلوکی
- دستورها و توابع موجود در منوی Project

# LabVIEW

# ۱۳

# قابلیتهای پیشرفته در نرم‌افزار LabVIEW

حال زمان آن فرا رسیده تا برای سهولت برنامه‌نویسی و ایجاد یک برنامه‌ی قوی و کاربردی، از قابلیتهای پیشرفته در نرم‌افزار LabVIEW استفاده کنید. با به‌کارگیری گزینه‌های Preferences می‌توانید این محیط برنامه‌نویسی را به‌گونه‌ای سفارشی کنید که بهترین عملکرد را برای شما به همراه داشته باشد و بتوانید حداکثر بهره را از این نرم‌افزار ببرید. در این فصل مطالبی در مورد گزینه‌های VI Setup فرا می‌گیرید. همچنین خواهید دید که چگونه می‌توان اعداد را در مبناهای دو، هشت، ده و شانزده نمایش داد. علاوه بر این می‌توانید برای هر یک از اعداد و متغیرهای عددی، واحد به‌خصوصی تعریف نمایید و از ویژگی تبدیل واحدها در LabVIEW استفاده کنید. در بخشهای انتهایی این فصل خواهید دید که چگونه می‌توان بخشی از صفحه‌ی نمودار بلوکی را به یک زیربرنامه تبدیل نمود. در ضمن با ابزارهای پیشرفته و مفید موجود در منوی Project نیز آشنا خواهید شد.

در این فصل اهداف زیر دنبال می‌شوند:

- به کارگیری گزینه‌های Preferences جهت سفارشی کردن محیط برنامه‌نویسی LabVIEW.
- آشنایی با گزینه‌های موجود در منوی VI Setup که به کمک آنها می‌توانید ظاهر و نحوه‌ی اجرای برنامه را به دلخواه تنظیم کنید.
- به‌کارگیری صفحه کلید جهت دستیابی به المان‌های موجود درصفحه‌ی پانل.
- فراگیری نحوه‌ی نمایش اعداد در مبناهای مختلف و روش انتساب واحد به متغیرهای عددی.
- آشنایی با نحوه‌ی ایجاد یک زیربرنامه از بخشی از صفحه‌ی نمودار بلوکی.
- آشنایی با دستورها و توابع موجود در منوی Project.

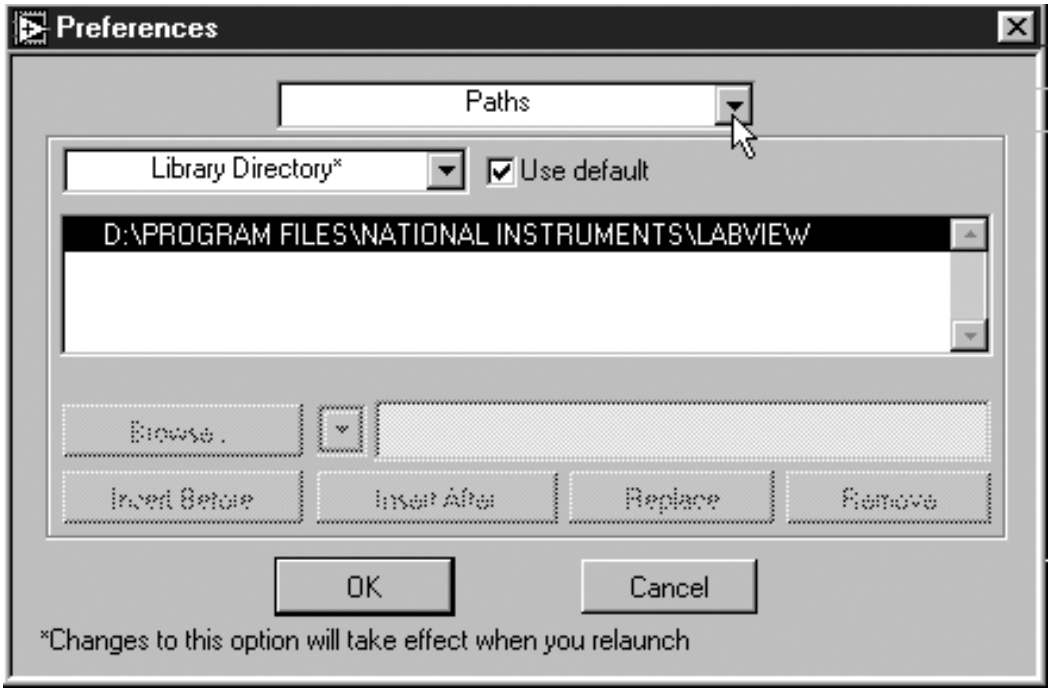
اصطلاحات کلیدی موجود در این فصل عبارتند از:

- گزینه های Preferences
- منوی VI Setup
- منوی SubVI Node Setup
- اولویت (Priority)
- مبنا (Radix)
- واحد (Unit)

## پنجره‌ی محاوره‌ای Preferences

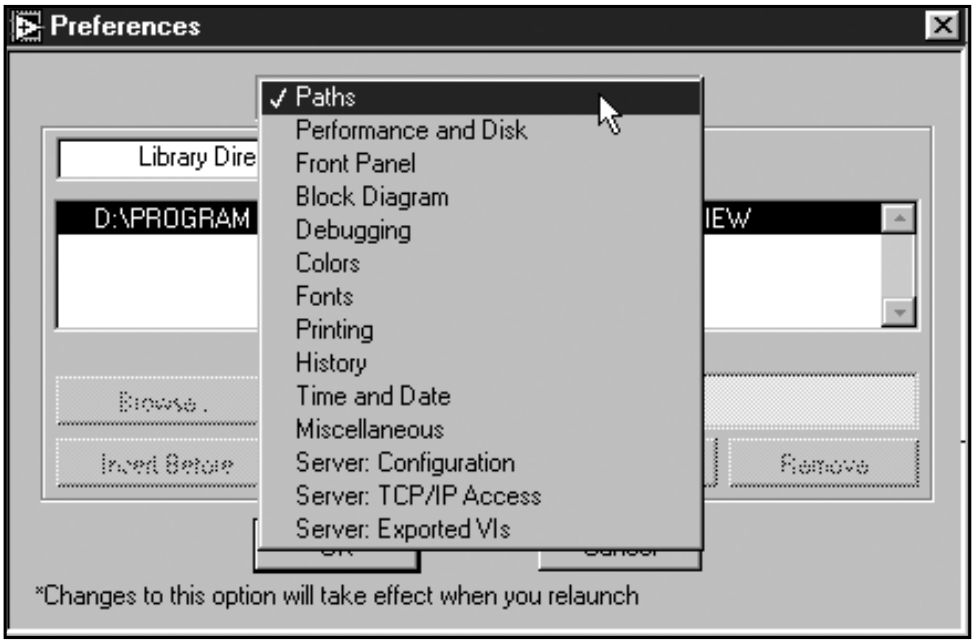
ممکن است تا به حال گزینه‌ی ...Edit >> Preferences را مورد استفاده قرار داده باشید. با انتخاب این گزینه یک پنجره‌ی محاوره‌ای مطابق شکل ۱-۱۳ باز می شود. در این بخش قصد داریم دستورها و گزینه های موجود در این پنجره را بررسی کنیم. به کمک گزینه های موجود در این پنجره می توانید مسیرهای گوناگون را جستجو کنید، رنگ و فونت استفاده شده در صفحه‌ی پانل را به دلخواه تغییر دهید و نحوه‌ی نمایش زمان و تاریخ را مشخص کنید.






**شکل ۱-۱۳: پنجره‌ی محاوره‌ای Preferences**

پس از کلیک کردن بر روی منوی حلقوی موجود در بالای این پنجره، گزینه های این منو مطابق شکل ۲-۱۳ ظاهر می شوند.



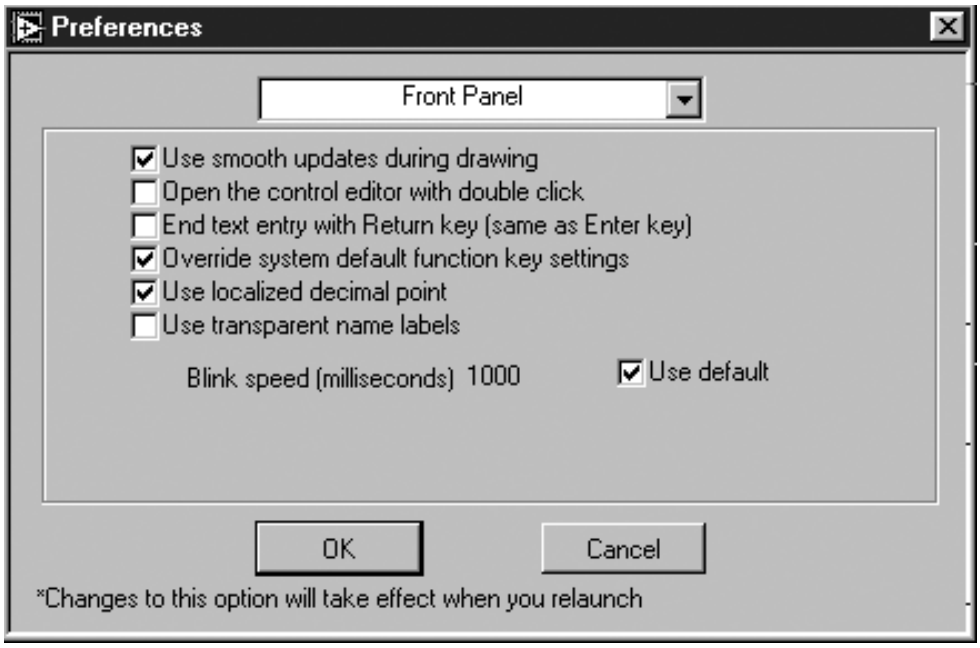

**شکل ۲-۱۳: گزینه‌های موجود در پنجره‌ی محاوره‌ای Preferences**

در صورت انتخاب هر یک از گزینه های نشان داده شده در شکل ۲-۱۳ زیرگزینه های دیگری ظاهر خواهند شد که به کمک آنها می توانید برخی از پارامترها را اولویت بندی نمایید و تغییر دهید.

برخی از گزینه‌های موجود در این منو بسیار مهم و کاربردی هستند. در ادامه به بررسی این گزینه‌ها می‌پردازیم:

- به کمک گزینه‌ی Path می توانید عنوان فهرست و مسیر قرار گرفتن برنامه های نوشته شده در محیط LabVIEW را تغییر دهید. به طور پیش‌فرض نام LabVIEW برای این فهرست انتخاب شده است. در صورتی که قصد دارید تا برنامه های نوشته شده‌ی خود را به منظور دسترسی سریع و آسان به آنها در محلی دیگر ذخیره نمایید، می توانید مسیر قرار گرفتن آنها را تغییر دهید.
- در گزینه‌ی Front Panel چند زیرگزینه‌ی مهم وجود دارد. به عنوان مثال گزینه‌ی Use smooth updates during drawing چراغ هشداردهنده‌ای را که با سرعت زیاد و بر روی برخی از گراف ها و نمودارها به طور منظم و متناوب چشمک می زند خاموش می کند. به کمک گزینه های این گروه می توانید سرعت چشمک زدن یک المان چشمک‌زن را تغییر دهید. شاید تا به حال تصور می کردید که این عمل امکان‌پذیر نیست اما در اینجا مشاهده می کنید که چگونه می توان سرعت چشمک زدن یک المان را تغییر داد[1]. در شکل ۳-۱۳ زیرگزینه های موجود در گزینه‌ی Front Panel را ملاحظه می کنید.
- گزینه‌ی Colors : با به کارگیری زیرگزینه های موجود در این گزینه می توانید رنگ صفحات پانل، نمودار بلوکی و... را که به صورت پیش‌فرض تعیین شده‌اند تغییر دهید. اگر قصد دارید صفحه‌ی پس‌زمینه‌ی خود را رنگ آمیزی کنید به کمک زیرگزینه های موجود می توانید این عمل را انجام دهید.

۱- البته انجام این عمل در صورتی امکان‌پذیر است که المان مورد بحث دارای خصوصیت Blinking باشد.




**شکل ۳-۱۳: زیرگزینه‌های موجود در گزینه‌ی Front Panel**

- گزینه‌ی Printing : به کمک زیرگزینه‌های موجود در این گزینه می‌توانید یکی از سه نحوه‌ی چاپ Bitmap، Standard و PostScript را انتخاب کنید. ممکن است بسته به نوع چاپگر در دسترس خود قصد داشته باشید تا جهت داشتن بهترین نتیجه‌ی چاپ، یکی از این گزینه‌ها را انتخاب نمایید.

حال گزینه‌های دیگر این منو را بررسی نمایید. به کمک این اولویتها می‌توانید شکل، فونت، نحوه‌ی نمایش زمان و تاریخ و... را تعیین کنید.

۱۳

## منوی VI Setup

ممکن است تا به حال این نیاز را در برنامه‌های خود احساس کرده باشید که به عنوان مثال در صورت برقرار شدن یک شرط خاص یک پنجره‌ی جدید باز گردد، یا اینکه فرض کنید در برنامه‌ای یک صفحه‌ی پانل اصلی در اختیار دارید و ساختار این برنامه به گونه‌ای تعریف شده باشد که فشار دادن هر یک از دکمه‌های تعبیه شده در صفحه‌ی اصلی منجر به باز شدن صفحه یا صفحات جدید دیگری شود. در این صفحات نیز می‌توان دکمه‌ای نظیر دکمه‌ی Exit در نظر گرفت که با فشار دادن آن به صفحه‌ی اصلی بازگشت.

به دلیل اینکه صفحه‌ای که مقابل روی شماست معمولاً صفحه‌ی پانل می‌باشد ممکن است قصد داشته باشید تا قابلیتها و ویژگیهای ظاهری این صفحه را سفارشی کنید و شکل ظاهری آن را مطابق میل خود در آورید. چندین گزینه وجود دارد که از طریق آنها می‌توانید شکل ظاهری و عملکرد برنامه‌ها را به میل خود در آورید یا تغییر دهید. برای انجام این عمل می‌توان از دو روش متفاوت استفاده نمود: یکی از طریق منوی VI Setup و دیگری به کمک منوی SubVI Node Setup.

جهت دستیابی به گزینه‌های موجود در منوی VI Setup ابتدا منوی کرکره‌ای را بر روی قاب آیکن برنامه که در گوشه‌ی بالایی سمت راست پنجره‌ی پانل قرار دارد باز کرده، سپس گزینه‌ی ...VI Setup را انتخاب نمایید. برای ایجاد

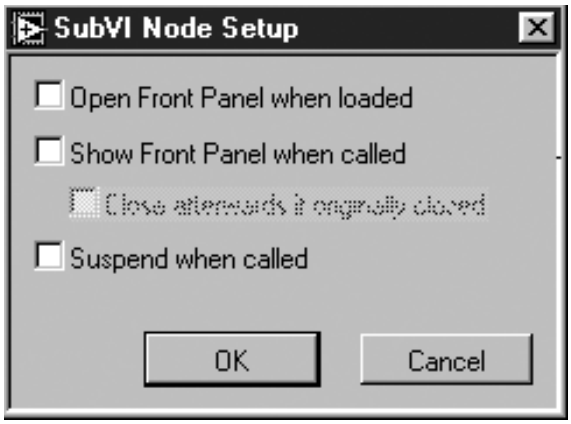
تغییر در ظاهر و عملکرد برنامه از طریق گزینه های موجود در منوی SubVI Node Setup، منوی کرکره‌ای را بر روی آیکن زیربرنامه در صفحه‌ی نمودار بلوکی باز نموده، سپس گزینه‌ی ...SubVI Node Setup را انتخاب کنید.

به تفاوت اساسی بین گزینه های این دو منو و نحوه‌ی عملکرد آنها دقت کنید. به کمک زیرگزینه های موجود در منوی SubVI Node Setup تنها می توان شکل ظاهری و نحوه‌ی اجرای بخشی از برنامه را که توسط زیربرنامه فراخوانی شده است تغییر داد. حال آن که زیرگزینه های موجود در منوی VI Setup همواره بر روی برنامه‌ی اصلی یا بالاترین سطح برنامه تأثیرگذارند.

به دلیل اینکه تعداد زیرگزینه های موجود در منوی SubVI Node Setup کمترند و درک آنها نیز ساده‌تر است، ابتدا به بررسی این منو و گزینه های موجود در آن می پردازیم:

## منوی SubVI Node Setup

در صورت انتخاب گزینه‌ی ...SubVI Node Setup از منوی کرکره‌ای آیکن زیربرنامه، یک پنجره‌ی محاوره‌ای مطابق شکل ۴-۱۳ باز می گردد.




**شکل ۴-۱۳: پنجره‌ی محاوره‌ای SubVI Node Setup**

در ادامه به بررسی گزینه های موجود در این پنجره می پردازیم:

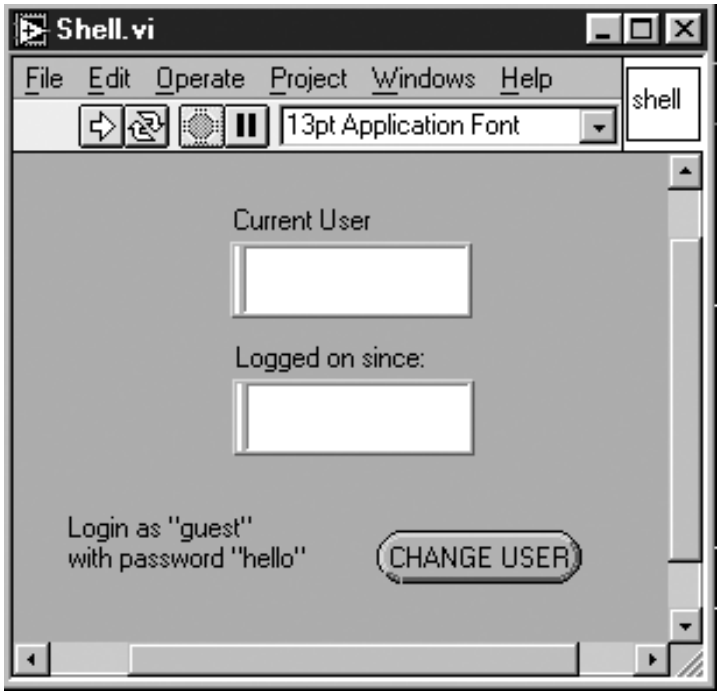
- Open Front Panel when loaded : در صورت انتخاب این گزینه در هنگام بارگذاری برنامه‌ی اصلی، صفحه‌ی پانل زیربرنامه نیز باز می شود (نظیر حالتی که برنامه ای را باز می کنید که درآن چند زیربرنامه فراخوانی شده است).
- Show Front Panel when called : در صورت انتخاب این گزینه در هنگام به اجرا در آمدن زیربرنامه، صفحه‌ی پانل برنامه ای که این زیربرنامه را فراخوانی نموده است نیز باز می شود.
- Close afterwards if originally closed : انتخاب این گزینه سبب می شودکه پس از به پایان رسیدن اجرای زیربرنامه، صفحه‌ی پانل آن نیز بسته شود. این گزینه تنها در صورتی فعال و در دسترس خواهد بود که گزینه‌ی Show Front Panel when called نیز انتخاب شده باشد.

- Suspend when called : در صورت انتخاب این گزینه، یک نقطه‌ی توقف در برنامه ایجاد شده، باعث می‌شود تا اجرای برنامه به هنگام فراخوانی زیربرنامه به طور موقت متوقف گردد. از این گزینه می‌توانید به عنوان یک ابزار برای رفع اشکال استفاده نمایید. بدین ترتیب ورودیهای اعمال شده در هر زیربرنامه تنها بر روی عملکرد و نحوه‌ی اجرای همان زیربرنامه و در همان محل فراخوانی شده تأثیرگذار هستند. انتخاب این گزینه بر روی زیربرنامه‌ی مورد بحث که در برنامه‌های دیگر فراخوانی شده است تأثیری ندارد.

## تمرین ۱-۱۳: ایجاد صفحه‌ی Login

در این تمرین با استفاده از گزینه‌های موجود در منوی SubVI Node Setup به ایجاد یک صفحه‌ی Login می‌پردازیم. این صفحه را می‌توان در برنامه‌های دیگر نیز استفاده نمود.

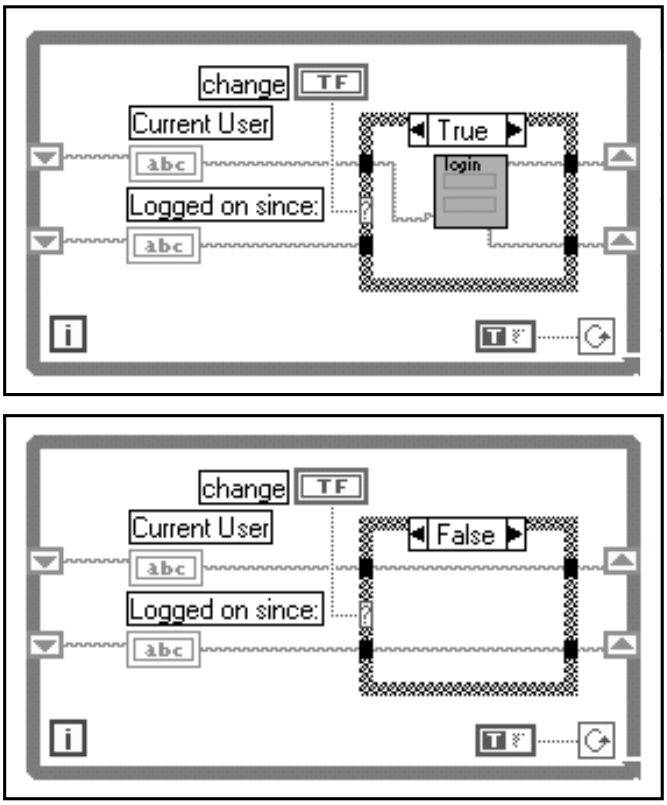
۱- صفحه‌ی پانل این برنامه را مطابق شکل ۵-۱۳ ایجاد نمایید.




**شکل ۵-۱۳: صفحه‌ی پانل در برنامه‌ی Shell.vi**

همان طور که ملاحظه می‌کنید در این صفحه یک کلید با برچسب CHANGE USER تعبیه شده است. در صورت فشار دادن این کلید، یک زیربرنامه فراخوانی می‌شود. برنامه‌ی اصلی را با عنوان Shell.vi ذخیره کنید.

۲- صفحه‌ی نمودار بلوکی نشان داده شده در شکل ۶-۱۳ را ایجاد کنید.

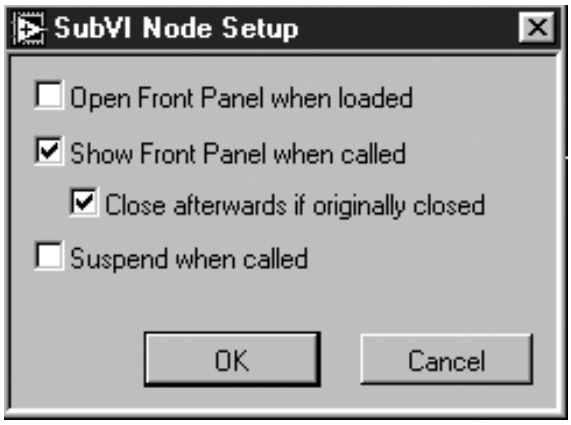



**شکل ۶-۱۳: صفحه‌ی نمودار بلوکی در برنامه‌ی Shell.vi**

۱۳

همان گونه که در این شکل ملاحظه می کنید زیربرنامه‌ی Login.vi در صفحه‌ی نمودار بلوکی برنامه‌ی اصلی فراخوانی شده است. این زیربرنامه در مسیر LabVIEW\Exercise\Ch13.llb قرار دارد.

۳- منوی کرکره‌ای را بر روی آیکن زیربرنامه‌ی Login باز کنید و گزینه‌ی ...SubVI Node Setup را انتخاب کنید. در این حالت، یک پنجره‌ی محاوره‌ای SubVI Node Setup مطابق شکل ۷-۱۳ ظاهر می گردد.




**شکل ۷-۱۳: پنجره‌ی محاوره‌ای SubVI Node Setup مربوط به زیربرنامه‌ی Login**

هنگامی که برنامه‌ی Shell.vi اجرا می‌شود، آخرین مقادیر تغییر داده شده در Current User به نمایش در می‌آیند و تا زمانی که مجدداً زیربرنامه‌ی Login.vi توسط کلید CHANGE USER فراخوانی و اصلاح نگردد، این مقادیر بدون تغییر باقی می‌مانند. قبل از به اجرا در آمدن این تمرین، حتماً صفحه‌ی Login را ببندید. در غیر این صورت پس از اتمام اجرای برنامه، صفحه‌ی Login بسته نخواهد شد. دلیل این امر، انتخاب گزینه‌ی Close afterwards if originally closed در پنجره‌ی محاوره‌ای SubVI Node Setup است.

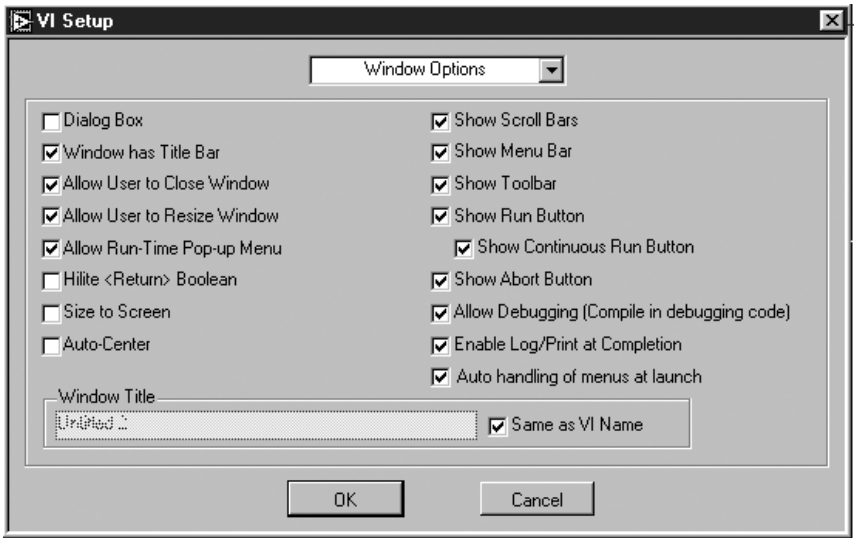
## گزینه‌های موجود در منوی VI Setup

تعداد گزینه‌های موجود در منوی VI Setup بیش از گزینه‌های عنوان شده در منوی SubVI Node Setup است. پس از انتخاب گزینه‌ی ...VI Setup از منوی کرکره‌ای آیکن برنامه، یک پنجره‌ی محاوره‌ای ظاهر خواهد شد که سه مجموعه گزینه در اختیار شما قرار می‌دهد. این مجموعه‌ها عبارتند از: Windows Options، Execution Options و Documentation. در ادامه به بررسی زیرگزینه‌های موجود در هر یک از این مجموعه‌ها می‌پردازیم.

## مجموعه‌ی Windows Options

۱۳

به کمک گزینه‌های موجود در مجموعه‌ی Windows Options می‌توانید جزییات ظاهری و شکل پنجره‌ی برنامه را تغییر دهید. در شکل ۸-۱۳ گزینه‌های موجود در این مجموعه را ملاحظه می‌کنید.




**شکل ۸-۱۳: گزینه‌های موجود در مجموعه‌ی Windows Options**

برخی از گزینه‌های موجود در این صفحه نیازی به توضیح نداشته و توضیح آنها در خود گزینه مستتر است. به هر حال به ذکر برخی از نکات و موارد لازم و ضرروی در مورد برخی از گزینه‌ها می‌پردازیم:

- در مورد غیر فعال نمودن گزینه‌هایی نظیر Show Abort Button با دقت و آگاهی کامل عمل کنید.

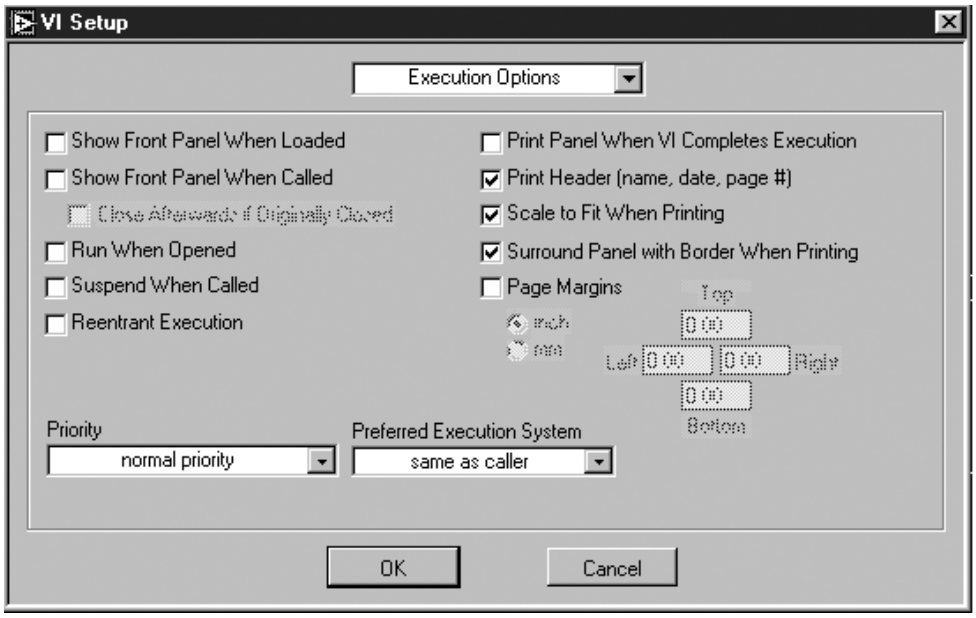
در صورت غیر فعال کردن این گزینه، دکمه‌ی Abort در نوار ابزار حذف می‌شود و پس از به اجرا در آمدن برنامه، حتی به کمک صفحه کلید نیز قادر به متوقف ساختن اجرای آن نخواهید بود.

- در صورت انتخاب گزینه‌ی Dialog Box شکل ظاهری پنجره‌ی برنامه به صورت پنجره‌های محاوره‌ای سیستم عامل در خواهد آمد و از دسترسی به پنجره‌های دیگر در LabVIEW در هنگام فعال بودن پنجره‌ی محاوره‌ای جلوگیری به عمل می آید.
- در صورت فعال نمودن گزینه‌ی Hilite <Return> Boolean، المان جبری کنترل که به کلید <Enter> یا <Return> نسبت داده شده است متمایز می گردد. این المان توسط یک حاشیه‌ی سیاه‌رنگ که در اطراف آن قرار می گیرد متمایز می‌شود.
- در صورت انتخاب گزینه‌ی Allow Debugging می توانید در حین به اجرا در آمدن برنامه به ابزارهای رفع اشکال در LabVIEW دسترسی پیدا کنید. این گزینه به صورت پیش‌فرض فعال شده است. در صورت غیر فعال بودن این گزینه سرعت اجرای برنامه اندکی افزایش یافته، حجم اشغال شده‌ی حافظه‌ی توسط برنامه نیز اندکی کاهش می یابد.

## مجموعه‌ی Execution Options

۱۳

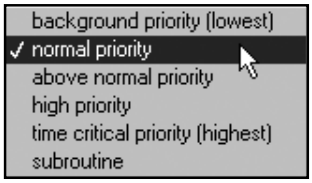
پنجره‌ی Execution Options حاوی همان گزینه‌های موجود در منوی SubVI Node Setup می‌باشد که چند گزینه‌ی دیگر نیز به آنها اضافه شده است. در میان گزینه‌های موجود در این پنجره تنها به بررسی دو مورد Priority و Reentrant Execution می پردازیم. درک این دو گزینه بسیار مشکل است و به ندرت مورد استفاده قرار می گیرند. در شکل ۹-۱۳ گزینه‌های موجود در مجموعه‌ی Execution Options نشان داده شده است.




**شکل ۹-۱۳: گزینه‌های موجود در مجموعه‌ی Execution Options**

هنگامی که چند برنامه به صورت موازی به اجرا در می‌آیند، LabVIEW آنها را در یک ردیف مرتب می‌کند و بنابر اولویت، مدت زمانی را جهت به اجرا در آمدن هر یک اختصاص می‌دهد. هر برنامه در زمان در نظر گرفته شده به اجرا در می‌آید و پس از آن به انتهای صف منتقل می‌شود. سپس برنامه‌ی بعدی موجود در صف اجرا می‌گردد.

گزینه‌ی Priority زمان اختصاص داده شده توسط پردازنده به برنامه را تعیین می‌کند. گزینه‌های قابل انتخاب در مورد تعیین اولویت را با کلیک کردن بر روی منوی حلقوی Priority مشاهده کنید. در شکل ۱۰-۱۳ ملاحظه می‌کنید که گزینه‌ی normal priority به صورت پیش‌فرض انتخاب شده است. با انتخاب گزینه‌های دیگر می‌توان زمان اختصاص داده شده توسط پردازنده به برنامه را تعیین نمود.




**شکل ۱۰-۱۳: گزینه‌های موجود در منوی حلقوی Priority**

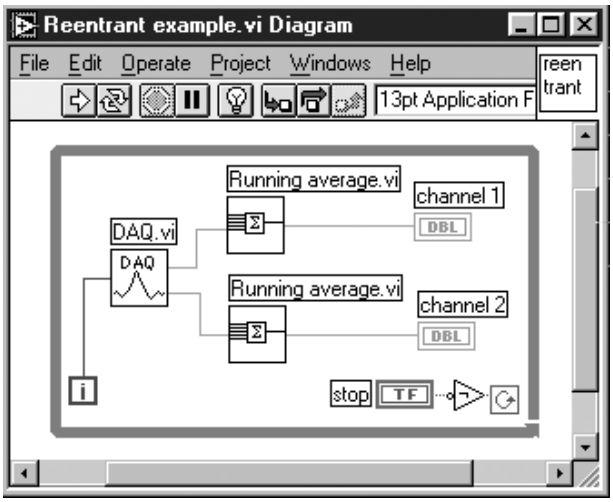
همان گونه که می دانید در هنگام اجرای موازی، برنامه هایی که درجه‌ی اولویت بالاتری دارند قبل از برنامه های با سطح اولویت پایین تر به اجرا در می آیند. به عنوان مثال سه برنامه که به صورت موازی اجرا می شوند را در نظر بگیرید. در این برنامه‌ها دو برنامه دارای درجه‌ی اولویت high priority هستند و برنامه‌ی سوم با اولویت normal priority است. ترتیب اجرای این برنامه ها به صورت زیر است:

ابتدا هر دو برنامه که به آنها درجه‌ی تقدم high priority نسبت داده شده است به صورت همزمان به اجرا در می‌آیند و پس از تکمیل آنها، برنامه‌ی سوم که دارای درجه‌ی اولویت normal priority است اجرا می‌شود. در صورتی که برنامه های با درجه‌ی تقدم high priority در داخل حلقه های پایان ناپذیر قرار گیرند، برنامه‌ی با درجه‌ی اولویت normal priority ممکن است هیچ گاه به اجرا در نیاید.

به طور معمول لزومی ندارد که حتماً خود را در مورد به اجرا در آوردن همزمان چند برنامه محدود کنید. در صورتی که قصد دارید مدت زمان بیشتری را جهت به اجرا در آوردن یک برنامه اختصاص دهید، قبل از به کارگیری گزینه های Priority، از دستورهای Wait استفاده کنید.

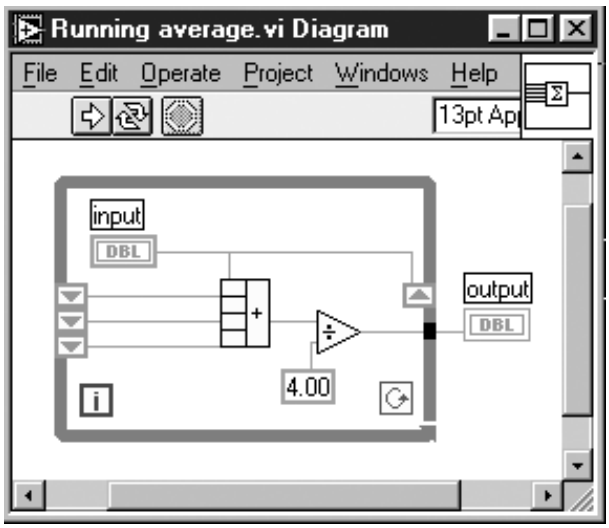
گزینه‌ی Reentrant Execution نیز حائز اهمیت است. به طور معمول در صورتی که فراخوانی یک زیربرنامه به صورت موازی در دو یا چند محل صورت گیرد، LabVIEW برای هر یک از برنامه‌هایی که زیربرنامه را فراخوانی کرده‌اند، فضایی مجزا جهت ذخیره‌ی اطلاعات اختصاص می‌دهد. در برخی موارد لازم است تا داده های مربوط به هر یک از زیربرنامه ها نیز به صورت جداگانه ذخیره شوند. به عنوان مثال در شکل ۱۱-۱۳ ملاحظه می کنید که در صفحه‌ی نمودار بلوکی برنامه‌ی Reentrant example.vi در مورد هر کانال یک مرتبه از زیربرنامه‌ی Running Average.vi استفاده شده است.

۱۳




**شکل ۱۱-۱۳: صفحه‌ی نمودار بلوکی در برنامه‌ی Reentrant example.vi**

ثباتهای انتقال در زیربرنامه‌ی Running Average.vi به صورت uninitialized استفاده شده‌اند. بدین مفهوم که این نوع ثباتها حتی پس از تکمیل اجرا یا توقف برنامه و اجرای مجدد آن، اطلاعات خود را حفظ می کنند (در صورتی که مطالب عنوان شده در مورد ثباتهای انتقال را فراموش کرده‌اید به فصل ۶ مراجعه کنید). اگر این برنامه در مد پیش‌فرض (مد غیر از Reentrant) به اجرا در آید، نتایج دور از انتظار و نادرستی به‌دست خواهد آمد. زیرا ورودیهای هر زیربرنامه، داده‌های موجود در ثباتهای مربوط به فراخوانی زیربرنامه‌ی قبلی هستند. اما در صورت انتخاب گزینه‌ی Reentrant Execution، برای ذخیره‌ی داده‌های هر زیربرنامه، فضای جدا گانه‌ای اختصاص داده می شود. درست نظیر حالتی که این دو زیربرنامه کاملاً متفاوت و جدا از هم باشند. در شکل ۱۲-۱۳ صفحه‌ی نمودار بلوکی زیربرنامه‌ی Running Average.vi را ملاحظه می کنید.



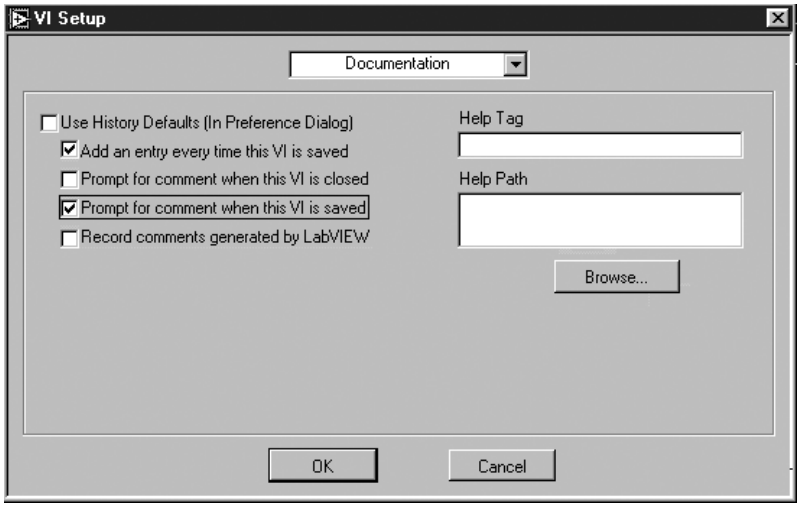

**شکل ۱۲-۱۳: صفحه‌ی نمودار بلوکی در زیربرنامه‌ی Ruuning Average.vi**

## تمرین ۲-۱۳: بررسی تأثیر انتخاب گزینه‌ی Reentrant Execution در اجرای برنامه

به عنوان یک مثال جالب و قابل توجه، ابتدا برنامه‌ی Running Average.vi را از مسیر LabVIEW\ Exercise\Ch13.llb اجرا نمایید و عملکرد و نحوه‌ی اجرای آن را بررسی کنید. سپس برنامه‌ی Reentrant example.vi را از همان مسیر اجرا نمایید. برنامه‌ی مذکور زیربرنامه‌ی Running Average.vi را دو مرتبه فراخوانی می‌کند. در مرحله‌ی بعد، گزینه‌ی Reentrant Execution را در مورد زیربرنامه‌ی Running Average.vi انتخاب نمایید و پس از اجرای مجدد برنامه‌ی Reentrant example.vi، نتایج به‌دست آمده را با یکدیگر مقایسه کنید. اکنون دلیل وجود اختلاف بین نتایج به‌دست آمده از دو حالت مذکور را توجیه کنید.

## مجموعه‌ی Documentation

گزینه‌های موجود در مجوعه‌ی Documentation ابزارهای پیشرفته‌ی برنامه‌نویسی در ارتباط با پنجره‌ی VI History می‌باشند. دسترسی به این پنجره از طریق منوی Windows نیز امکان پذیراست. همان گونه که از نام پنجره‌ی VI History استنباط می‌گردد، با انتخاب این گزینه توضیحات اضافه شده توسط کاربر در حین نوشتن برنامه ضبط می‌گردد. در شکل ۱۳-۱۳ گزینه‌های موجود در این مجموعه را ملاحظه می‌کنید.



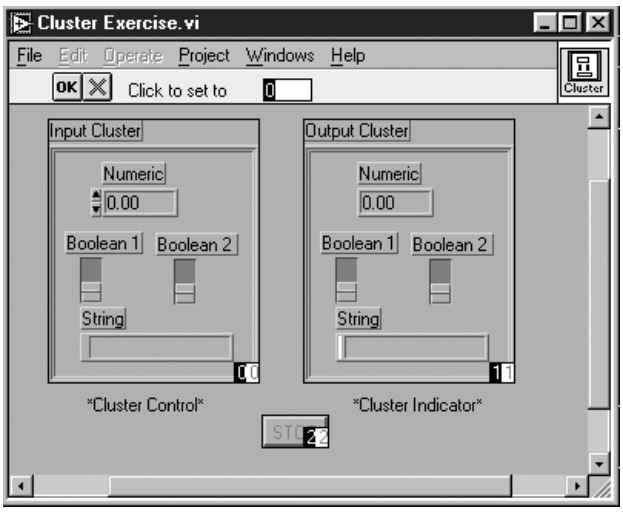

**شکل ۱۳-۱۳: گزینه‌های موجود در مجموعه‌ی Documentation**

## به‌کارگیری صفحه کلید جهت برقراری ارتباط با صفحه‌ی پانل

اگر شما نیز جزء آن دسته از کاربرانی هستید که هنوز به استفاده از ماوس عادت نکرده‌اید و یا به کارگیری صفحه کلید برایتان راحت‌تر است برای شما خبر خوشی داریم. در محیط LabVIEW می‌توانید ساختار برنامه‌های خود را به گونه‌ای تعیین نمایید که از طریق کلید <Tab> به المان‌های کنترل درصفحه‌ی پانل دسترسی پیدا کنید. دسترسی به المان‌های نشان‌دهنده به این روش امکان‌پذیر نیست. زیرا در مورد این المان‌ها لازم نیست تا به آنها دسترسی

پیدا کنید و به عنوان مثال مقادیر ورودی را به آنها اعمال کنید. یک المان کنترل انتخاب شده که در این حالت به آن Key Focus گفته می شود در یک حاشیه‌ی مستطیلی قرار می گیرد. در این حالت می توانید مقادیر لازم را با استفاده از کلیدهای مناسب به المان مذکور اعمال کنید. در هنگام به کارگیری صفحه کلید رعایت نکات زیر الزامی است.

- اگر مقادیر دلخواه را مستقیماً در المان کنترل انتخاب شده تایپ می کنید، پس از اتمام تایپ، جهت وارد کردن مقادیر تایپ شده حتماً کلید <Enter> را فشار دهید.
- در مورد المان های کنترل عددی و منوهای حلقوی برای دستیابی به مقدار دلخواه می توانید از کلیدهای مکان نما استفاده کنید. فشار دادن کلید <Shift> به همراه یکی از دو کلید <Up> یا <Down>، سرعت افزایش یا کاهش مقدار مورد نظر را افزایش می دهد. برای تعیین مقدار افزایش داده ها از گزینه‌ی ...Data Range استفاده کنید. این گزینه در منوی کرکره ای المان کنترل قرار دارد.
- در مورد المان های کنترل جبری، کلید <Enter> وضعیت المان جبری را تغییر می دهد. بدین مفهوم که با هر بار فشار دادن کلید مذکور حالت المان جبری انتخاب شده عوض می شود.
- ترتیب گذر از یک المان کنترل به المان دیگر به کمک کلید <Tab>، به ترتیب ایجاد شدن آنها در صفحه‌ی پانل بستگی دارد. در برنامه هایی که در صفحه‌ی پانل آنها چندین المان کنترل وجود دارد ممکن است قصد داشته باشید که ترتیب گذر از این المان ها را به دلخواه تغییر دهید. بدین معنی که در صورت انتخاب شدن یک المان کنترل پس از فشار دادن کلید <Tab> یکی از المان های موجود در صفحه به دلخواه شما فعال شود. در LabVIEW به این عمل، تعیین اولویت ترتیب المان های پانل گویند. برای تعویض ترتیب در پانل، گزینه‌ی ...Panel Order >> Edit را انتخاب کنید. عملکرد این گزینه نظیر گزینه‌ی ...Cluster Order است. در صورتی که به یاد داشته باشید عملکرد گزینه‌ی ...Cluster Order را در فصل ۷ مورد بحث قرار دادیم. در شکل ۱۴-۱۳ تغییرات ظاهری صفحه‌ی پانل در صورت انتخاب گزینه‌ی ...Panel Order نشان داده شده است.

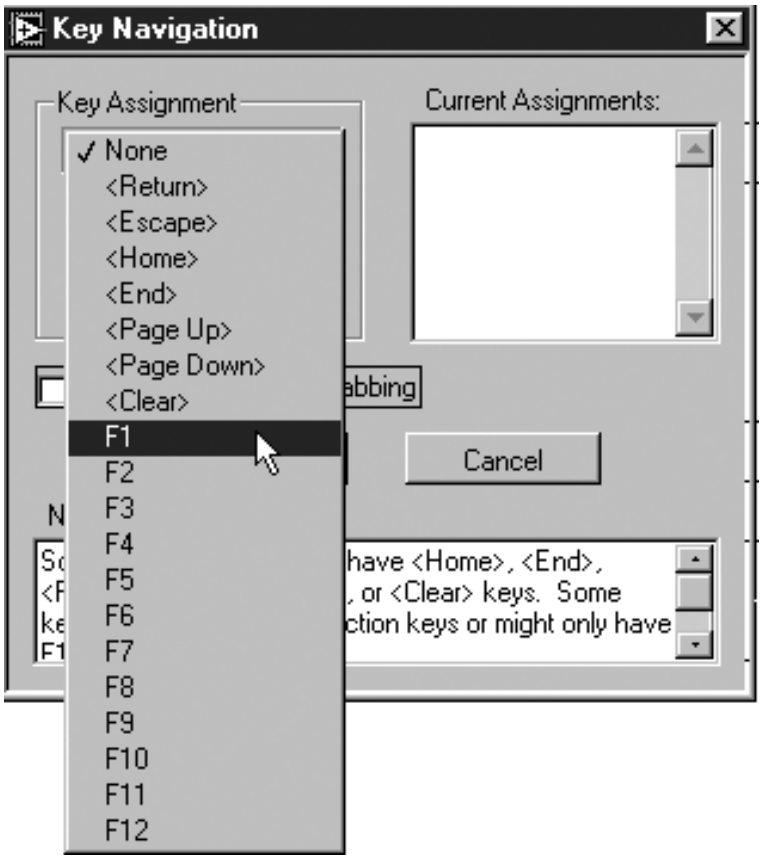



**شکل ۱۴-۱۳: نمای ظاهری صفحه‌ی پانل در صورت انتخاب گزینه‌ی ...Panel Order**

۱۳

همان گونه که ملاحظه می کنید هر یک از المان های کنترل در داخل یک مستطیل قرار می گیرند. در گوشه‌ی پایینی سمت راست این مستطیل دو عدد وجود دارد. عددی که در مستطیل سفیدرنگ دیده می شود، مشخص کننده‌ی اولویت قبلی المان در صفحه‌ی پانل است. در حالی که عدد قرار گرفته در مستطیل سیاه رنگ ترتیب اخیر المان در صفحه‌ی پانل را نشان می دهد. به عبارت دیگر این عدد، اولویت جدید المان را در صفحه‌ی پانل مشخص می کند. برای انتساب یک عدد جدید به المان در صفحه‌ی پانل در نوار منو یک عدد را تایپ نموده، سپس بر روی المان مورد نظر کلیک کنید. در این صورت عدد تایپ شده به المان مورد نظر اختصاص داده می شود. پس از انجام تمامی تغییرات در ترتیب المان های موجود در صفحه‌ی پانل برای تأیید و انتساب تمامی این اعداد و اولویتها، بر روی دکمه‌ی OK کلیک کنید. برای اعلام انصراف از انجام تغیرات انجام شده و بازگشت به ترتیب اولویت قبلی المان ها در صفحه‌ی پانل بر روی دکمه‌ی X کلیک نمایید.

همچنین می توانید به هر یک از المان ها یک کلید خاص از صفحه کلید را اختصاص دهید. برای انجام این عمل منوی کرکره‌ای را بر روی المان کنترل مورد نظر باز و گزینه‌ی ...Key Navigation را انتخاب کنید. در این صورت یک پنجره‌ی محاوره‌ای نظیر شکل ۱۵-۱۳ ظاهر خواهد شد.



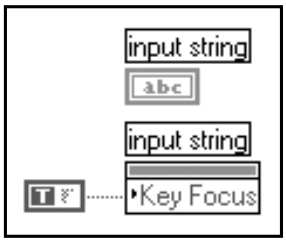

**شکل ۱۵-۱۳: پنجره‌ی محاوره‌ای Key Navigation**

همان گونه که در شکل ۱۵-۱۳ ملاحظه می کنید می توان به هر یک از المان های کنترل یکی از کلیدهای <F1> ، <F2> یا حتی کلیدهایی نظیر <Ctrl> ، <Home> و... را اختصاص داد. در صورت انتساب کلیدی خاص به یک المان، در صورت فشار دادن کلید مذکور المان مورد نظر فعال شده و به اصطلاح Key Focus می گردد. در این حالت، استفاده از کلید <Tab> برای فعال کردن المان های کنترل الزامی نیست. استفاده از گزینه‌ی Key Navigation... در مواردی بسیار مفید است که المان های کنترل بسیار زیادی در صفحه‌ی پانل وجود داشته باشند اما تعداد کمی از آنها مورد استفاده قرار می گیرند.

شایان ذکر است که امکان فعال یا غیر فعال کردن یک المان کنترل از طریق گره‌ی خصوصیت نیز امکان پذیر است.

خصوصیت Key Focus در مورد یک المان کنترل، یک مقدار جبری است و در صورتی که در وضعیت True قرار گرفته باشد بدین معنی است که المان مورد نظر انتخاب شده است و آماده‌ی دریافت ورودی از طریق صفحه کلید می‌باشد. در شکل ۱۶-۱۳ طریقه‌ی فعال و غیر فعال شدن المان input string از طریق خصوصیت Key Focus نشان داده شده است.






شکل ۱۶-۱۳

همان گونه که ملاحظه می کنید در صورتی که مقدار جبری نسبت داده شده به گره‌ی خصوصیت در وضعیت True باشد، این المان آماده‌ی دریافت ورودی از طریق صفحه کلید است.

## تمرین ۳-۱۳: استفاده از خصوصیت Key Focus

برنامه‌ی Login.vi را که در تمرین ۱-۱۳ ایجاد نمودید به صورتی بازسازی کنید که ابتدا جعبه‌ی LOGIN توسط گره‌ی خصوصیت Key Focus فعال شود و آماده‌ی دریافت ورودی گردد. این برنامه باید به گونه‌ای نوشته شود که پس از فشرده شدن یکی از دو کلید <Enter> یا <Return> توسط کاربر، نشانگر به بخش PASSWORD انتقال یابد. شایان ذکر است که کلمه‌ی عبور وارد شده در بخش PASSWORD نباید قابل رؤیت باشد. پنجره‌ی پانل این برنامه باید در وسط صفحه ظاهر گردد (به نوارهای پیمایش موجود در کنار پنجره‌ی پانل دقت کنید).

**شکل ۱۷-۱۳: صفحه‌ی پانل در برنامه‌ی Login.vi**

۱- برای انجام این تمرین حتماً از گره‌ی خصوصیت Key Focus استفاده کنید.

۲- منوی کرکره‌ای را بر روی المان PASSWORD باز کنید و گزینه‌ی Password را برای رشته انتخاب نمایید.

۳- منوی کرکره‌ای را بر روی المان LOGIN باز کنید و گزینه‌ی Limit to Single Line را انتخاب نمایید.

این برنامه چگونه اطلاع پیدا می کند که ورودی رشته‌ای LOGIN به اتمام رسیده است؟ یا به عبارت دیگر چه موقع کلید <Enter> یا <Return> فشار داده شده است؟ برای حل این مشکل باید یک متغیر محلی تعریف کنید که دائماً بررسی کند که آیا کاراکتر <Enter> یا <Return> در ورودی درج شده است یا خیر. یافتن پاسخ این سؤال را به خوانندگان عزیز واگذار می کنیم.

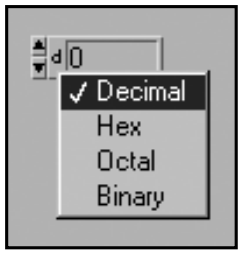
## مبنا و واحد

یکی از ویژگیهای قابل توجه در بسته‌ی نرم‌افزاری LabVIEW آن است که می توان اعداد را در مبناها و واحدهای مختلف استفاده نمود. برای هر یک از اعداد استفاده شده در المان های کنترل و نشان‌دهنده یک نمایش اصلی به نام مبنا تعریف شده است. گزینه های اختیاری دیگری نیز در مورد اعداد مذکور به نام واحد در نظر گرفته شده است. به طور معمول اعداد در صفحه‌ی پانل به صورت مبنای ده به نمایش در می آیند. اما در برخی از برنامه ها لازم است تا این اعداد در مبناهای دیگر نظیر مبنای دو، هشت یا شانزده به کار برده شوند.

به طریق مشابه و در مواردی که واحد اعداد استفاده شده یکسان نبوده و قصد داشته باشید تا واحدها را به یکدیگر تبدیل کنید، می توانید واحد آنها (نظیر متر، کالری، درجه‌ی سانتیگراد و... ) را در کنار المان‌ها به نمایش در آورید.

## مبناها

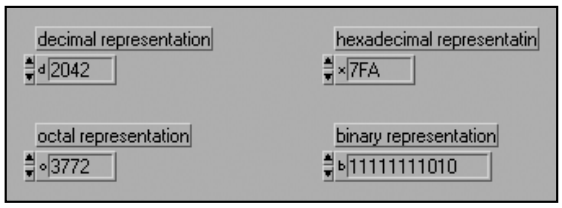
در نرم‌افزار LabVIEW نمایش دهنده‌های اعداد از قابلیت انعطاف بالایی برخوردارند. این اعداد را می‌توان در یکی از مبناهای دو، هشت، ده یا شانزده به نمایش در آورد. به عنوان مثال کاربری که در حال کار با I/O دیجیتال می‌باشد ممکن است جهت به نمایش در آوردن اعداد بر روی صفحه‌ی پانل، مبنای دو را ترجیح دهد. زیرا با این روش می‌تواند بین هر بیت و خط دیجیتال مربوط تناظر یک‌به‌یک برقرار نماید. برای نمایش یک عدد در مبناهای مختلف منوی کرکره‌ای را بر روی آن باز نموده و گزینه‌ی Show >> Radix را انتخاب کنید. سپس به کمک ابزار Operating Tool بر روی حروف کوچکی که در کنار صفحه‌ی نمایش دهنده‌ی عدد ظاهر می‌گردد، کلیک کنید و از منوی مذکور مبنای دلخواه را انتخاب کنید. در شکل ۱۸-۱۳ نحوه‌ی انتخاب مبنا نشان داده شده است.




**شکل ۱۸-۱۳: نحوه‌ی انتخاب مبنا**

در صورتی که المان کنترل یا نشان‌دهنده دارای هیچ نوع نمایش عدد صحیح نباشد، در این منو تنها گزینه‌ی Decimal فعال خواهد بود.

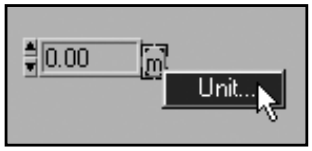
در شکل ۱۹-۱۳ نحوه‌ی نمایش عدد دهدهی 2042 را در مبناهای مختلف ملاحظه می کنید.




**شکل ۱۹-۱۳: نحوه‌ی نمایش عدد دهدهی 2042 در مبناهای مختلف**

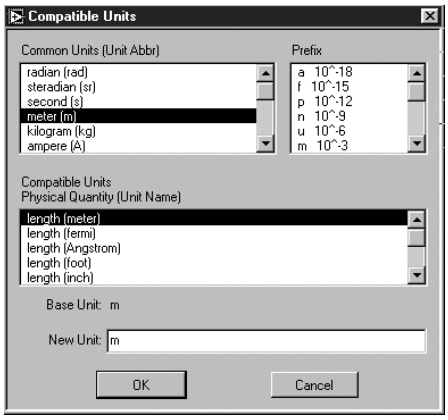
## واحدها

برای هر یک از اعداد اعشاری یا صحیح در المان کنترل یا نشان‌دهنده می‌توان واحدی نظیر متر، ثانیه یا ... در نظر گرفت. برای برچسب گذاری واحدها یا به عبارت دیگر برای تعیین واحد، منوی کرکره‌ای را بر روی المان مورد نظر باز و گزینه‌ی Show >> Unit Label را انتخاب کنید. در شکل ۲۰-۱۳ طریقه‌ی دستیابی به این واحدها نشان داده شده است.




**شکل ۲۰-۱۳: روش دستیابی به واحد**

پس از به نمایش در آمدن برچسب واحد، با استفاده از علائم اختصاری نظیر m برای متر، ft برای فوت و.. واحد دلخواه را به عدد مذکور اختصاص دهید. اگر واحد ناشناخته‌ای را به یکی از اعداد نسبت دهید برچسب مذکور با یک علامت ؟ متمایز می گردد. اگر قصد دارید از انواع واحدهای معتبر و شناخته شده در LabVIEW اطلاع پیدا کنید یک واحد ساده نظیر m را به یکی از اعداد اختصاص دهید. سپس گزینه‌ی ...Unit را از منوی کرکره‌ای بر روی بخش نشان‌دهنده‌ی واحد انتخاب نمایید. در این صورت یک پنجره‌ی محاوره‌ای مطابق شکل ۲۱-۱۳ باز خواهد شد که تمامی واحدهای قابل قبول و سازگار با آن به نمایش در می آیند.

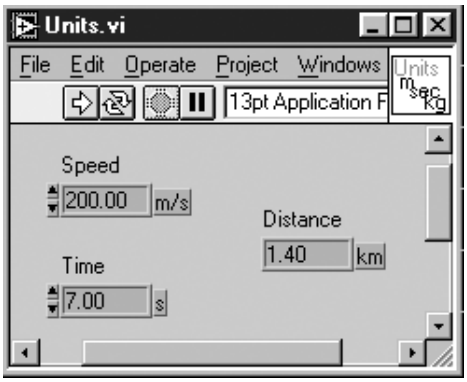



**شکل ۲۱-۱۳ : پنجره‌ی محاوره‌ای Compatible Units**

هر عدد به همراه یک واحد به عنوان یک عدد اعشاری تلقی می شود.

۱۳

در انجام اعمال محاسباتی بر روی داده‌ها، واحدهای آنها نیز منظور می شود. همان‌گونه که در شکل ۲۲-۱۳ ملاحظه می کنید برقرار نمودن ارتباط، تنها بین دو منبع و مقصد هم‌واحد یا سازگار امکان‌پذیر است. در این شکل دو عدد با کمیتهای سرعت و زمان در یکدیگر ضرب شده و حاصل آنها در خروجی با کمیت مسافت نشان داده شده است ($x = v.t$).

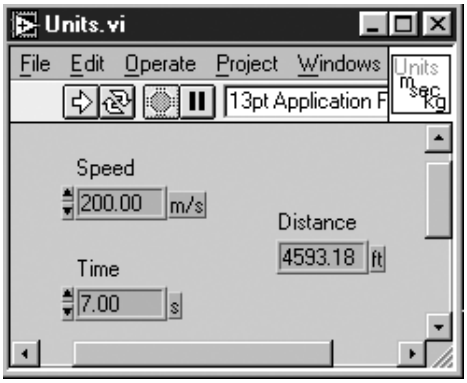



**شکل ۲۲-۱۳**

همان گونه که در شکل ۲۲-۱۳ ملاحظه می کنید واحد انتخاب شده برای سرعت، زمان و مسافت به ترتیب m/s، s و km می باشد. پس از انجام عمل ضرب بر روی دو کمیت سرعت و زمان با واحدهای 200 m/s و 7 s، مقدار

مسافت 1400 m به دست می آید. اما به دلیل اینکه برای این کمیت، واحد km انتخاب شده است، این عدد به صورت 1.40 km در خروجی نشان داده می شود.

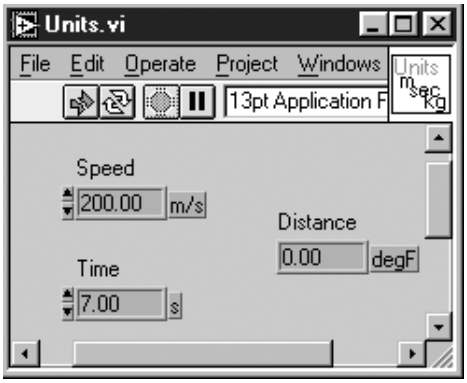
هنگامی که واحدهای سازگار را در مورد خروجی به کار می بریم منظور ما این است که واحد در نظر گرفته شده برای خروجی باید از نظر فیزیکی یا ریاضی معنی دار باشد. به عبارت دیگر واحد اختصاص داده شده برای خروجی نباید حتماً به یک سیستم خاص نظیر SI و یا سیستم انگلیسی متعلق باشد. به عنوان مثال هر دو واحد باید واحد اندازه گیری طول باشند. همان گونه که در شکل ۲۳-۱۳ نشان داده شده است، می توانید واحدهای سازگار را در سیستم های مختلف به یکدیگر تبدیل کنید.




شکل ۲۳-۱۳

برخی از دستورها و توابع LabVIEW نسبت به واحدها، مبهم و دو پهلو عمل می‌کنند. به عنوان مثال اگر واحدی برای اندازه گیری مسافت اختصاص داده‌اید، مشخص نیست که در دستور Add One باید یک متر، یک کیلومتر، یک فوت یا... به عدد مورد نظر افزوده شود.

همان گونه که در شکل ۲۴-۱۳ ملاحظه می کنید نمی توان واحدهای ناسازگار را برای اعداد در نظر گرفت. به عنوان مثال در این شکل المان Speed با واحد m/s در المان Time با واحد s ضرب شده و به المان Distance نیز واحد degF اختصاص داده شده است. در صورت انجام این عمل، پنجره‌ی Error List خطایی را با عنوان Wire: unit conflict نشان خواهد داد.




شکل ۲۴-۱۳

در مواردی که اعداد واحددار و بدون‌واحد با یکدیگر ترکیب می‌شوند، با احتیاط عمل کنید. به عنوان مثال می‌توانید یک عدد بدون‌واحد را در یک عدد واحددار ضرب کنید. واضح است که در این حالت عدد واحددار دیگری به دست خواهد آمد. اما اضافه کردن یک عدد بدون واحد به یک عدد واحددار مجاز نیست. برای انجام این عمل مطابق شکل ۲۵-۱۳ ابتدا به عدد بدون‌واحد، یک واحد متناظر با عدد دیگر نسبت داده، سپس عملیات جمع را انجام دهید. جهت انتساب واحد به عدد مورد نظر از دستور Convert Unit استفاده کنید. این دستور در زیرپالت Functions >> Numeric >> Conversion قرار دارد.

شکل ۲۵-۱۳

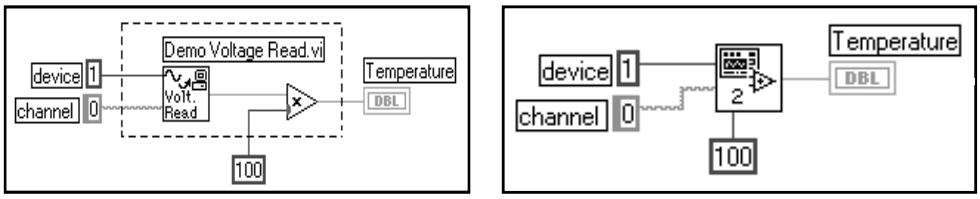
برای تعیین واحد، ابتدا واحد مطلوب را در داخل مستطیل ترمینال تایپ کنید و یا در صورتی که المان مذکور از نوع کنترل یا نشان‌دهنده باشد، منوی کرکره‌ای را بر روی ترمینال مربوط باز نموده، سپس گزینه‌ی ...Unit را انتخاب کنید. در این صورت لیستی از واحدهای معتبر و قابل قبول در LabVIEW به نمایش در می‌آیند. برای حذف واحد از اعداد واحددار نیز می‌توانید از دستور Convert Unit استفاده کنید. تمامی واحدهای اصلی و فرعی در جداول مندرج در فایل Appendix 4 >> Appendices آورده شده است.

۱۳

## ایجاد یک زیربرنامه از بخشی از صفحه‌ی نمودار بلوکی

یکی از روشهای برنامه‌نویسی صحیح و بدون اشکال، استفاده از قابلیت مدولار بودن است. منظور از مدول در LabVIEW در واقع همان زیربرنامه است. در هنگام نوشتن یک برنامه‌ی بزرگ و پیچیده می‌توان ابتدا آن را به چند زیربرنامه تقسیم کرد و سپس با هماهنگ کردن زیربرنامه‌ها برنامه‌ی اصلی را بازسازی نمود. به کمک گزینه‌ی Edit >> Create SubVI می‌توان بخشهایی از صفحه‌ی نمودار بلوکی را به زیربرنامه تبدیل کرد.

اگر برنامه‌ی وسیعی در اختیار دارید و قصد دارید تا قسمتی از آن را به زیربرنامه تبدیل کنید نگران نباشید. برای انجام این عمل کافی است بخشی از صفحه‌ی نمودار بلوکی را انتخاب نموده، سپس گزینه‌ی Edit >> Create SubVI را انتخاب کنید. در این حالت بخش مورد نظر به زیربرنامه تبدیل شده و المان‌های کنترل و نشان‌دهنده به صورت خودکار برای زیربرنامه‌ی جدید ایجاد می‌شوند. سپس زیربرنامه‌ی مذکور جایگزین بخش انتخاب شده در صفحه‌ی نمودار بلوکی برنامه‌ی اصلی شده و زیربرنامه‌ی جدید به صورت خودکار به سیمهای موجود در صفحه سیم‌کشی می‌گردد. به کارگیری این قابلیت به خصوص در مواردی که نیاز دارید تا بخشی از صفحه‌ی نمودار بلوکی یک برنامه را در برنامه یا برنامه‌های دیگر تکرار کنید، بسیار مفید و کاربردی است. در شکل ۲۶-۱۳ نمونه‌ای از کاربرد این گزینه را ملاحظه می‌کنید.




شکل ۲۶-۱۳

اگرچه در ابتدای این مجموعه به ذکر این قابلیت پرداختیم اما حال قصد داریم تا برخی از قواعد و نکات به کارگیری دستور Create SubVI را ذکر کنیم. به دلیل اینکه تبدیل برخی از صفحات نمودار بلوکی به زیربرنامه ممکن است باعث تغییر رفتار و عملکرد برنامه‌ی شما گردد، مواردی وجود دارد که استفاده از این قابلیت مجاز نیست. این موارد در لیست زیر آورده شده‌اند:

- در حالتی که تعداد ورودیها و خروجیهای زیربرنامه بیش از ۲۸ مورد باشد. زیرا حداکثر تعداد ورودیها و خروجیهای یک کانکتور ۲۸ عدد است. در این حالت پیشنهاد می کنیم که ابتدا بخشهای کوچکتر را به آرایه یا کلاستر تبدیل نموده و پس از کاهش تعداد ورودیها و خروجیها، بخش مورد نظر در صفحه‌ی نمودار بلوکی را به زیربرنامه تبدیل کنید.
- در حالتی که المان های داخل و یا خارج از یک ساختار[2] جهت تبدیل به زیربرنامه انتخاب شده باشند، در حالی که هنوز خود ساختار انتخاب نشده باشد. به عبارت دیگر تبدیل بخشهای داخلی یا خارجی یک ساختار به زیربرنامه بدون خود ساختار امکان پذیر نیست.
- به دلیل اینکه گره های خصوصیت به المان های کنترل در همان برنامه مربوط هستند، این گره ها به زیربرنامه ها کپی نمی شوند. در عوض گره های خصوصیت در برنامه‌ی فراخوان، باقی می‌مانند و خصوصیات مورد نظر در صورت نیاز به گره های خصوصیت، توسط المان های کنترل جدید از همان نوع به / از زیربرنامه منتقل می شوند. اذعان داریم که درک این مورد بسیار پیچیده است. برای درک این مطلب یک مورد را آزمایش کنید.
- علاوه بر مورد قبلی در صورتی که یک گره‌ی خصوصیت در داخل یک ساختار (نظیر یک حلقه‌ی While) قرار گرفته باشد، امکان تبدیل آن بخش از صفحه‌ی نمودار بلوکی به زیربرنامه وجود ندارد. به دلیل اینکه خصوصیات تنها یک مرتبه به زیربرنامه انتقال داده می شوند، گره های خصوصیت موجود در داخل حلقه و یا یک زیربرنامه به روز رسانده نمی شوند.
- در حالتی که بخش انتخاب شده از صفحه‌ی نمودار بلوکی شامل متغیرهای محلی است اما المان کنترل متناظر را در بر نمی گیرد و یا در حالتی که یک متغیر محلی متناظر با یک المان کنترل در برنامه‌ی فراخوان باقی می‌ماند و داده ها را از/ به زیربرنامه انتقال می دهد. در این حالت اولین متغیر محلی برای یک المان کنترل در داخل زیربرنامه به صورت یک المان کنترل جدید تلقی می‌شود و متغیرهای محلی مراحل پایین تر به المان کنترل جدید در زیربرنامه ارجاع داده می شوند.

۱۳

2- Structure

- در مواردی که بخش انتخاب شده از صفحه‌ی نمودار بلوکی شامل متغیرهای محلی یا ترمینال های صفحه‌ی پانل باشد که در داخل حلقه قرار گرفته‌اند و مقادیری که توسط آنها اندازه‌گیری می‌شود ممکن است در هر کجای صفحه‌ی نمودار بلوکی در حین اجرای حلقه تغییر کنند. در این حالت اگر حلقه را به یک زیربرنامه تبدیل کنید، احتمال دارد که عملکرد بخش انتخاب شده تغییر کند. اگر تعدادی از (و نه تمام) متغیرهای محلی یا ترمینال های پانل را انتخاب کرده باشید، LabVIEW به شما هشدار می‌دهد و از شما می خواهد تا بین ادامه و انصراف از انجام عملیات، یک مورد را انتخاب کنید.

در صورت انتخاب بخشی از برنامه و تبدیل آن به زیربرنامه هیچ راه بازگشتی وجود ندارد. بنابراین قبل از انجام این عمل حتماً یک نسخه‌ی کپی از برنامه‌ی خود تهیه نمایید و تبدیلات را بر روی نسخه‌ی کپی انجام دهید.

استفاده از گزینه‌ی Create SubVI در برنامه های عظیم و پیچیده بسیار مفید و کاربردی است. تبدیل بخشهایی از صفحه‌ی نمودار بلوکی به زیربرنامه علاوه بر قابلیت مدولار بودن، فضای بیشتری برای ایجاد دستورها و استفاده‌ی بهینه از صفحه‌ی برنامه نویسی ایجاد می کند.

## منوی Project

۱۳

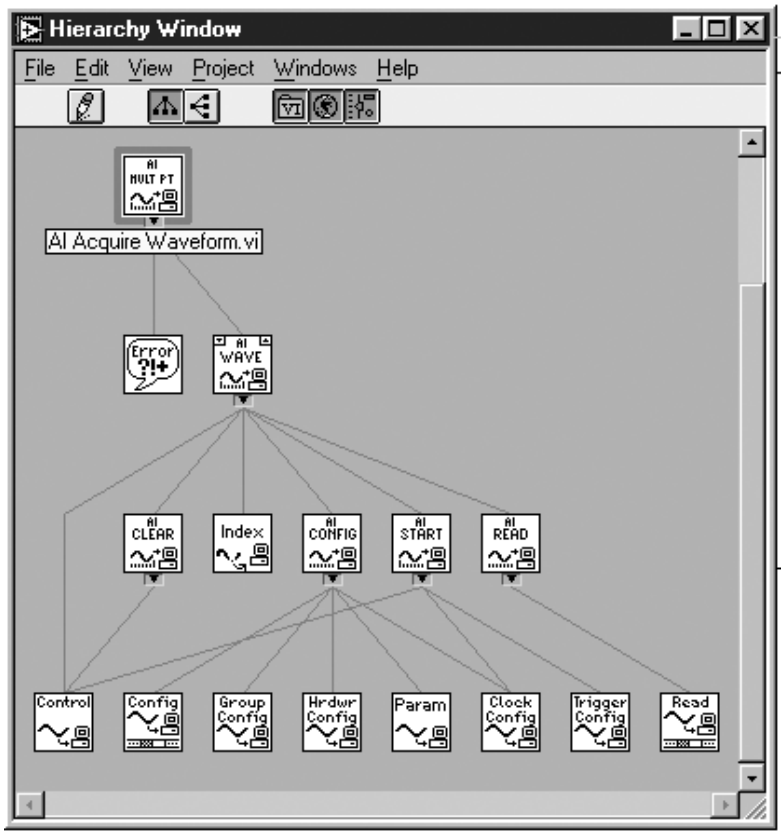
منوی Project دارای قابلیتهای بسیار زیادی است که جهت رفع اشکال و مدیریت در برنامه های وسیع و پیچیده بسیار مفید می باشند. در ادامه به بررسی برخی از گزینه های موجود در این منو می‌پردازیم :

- به کمک گزینه‌ی Show VI Hierarchy می‌توانید سلسله مراتبِ حلقه ها و ساختارهای یک برنامه و زیربرنامه های آن را مشاهده کنید.
- با استفاده از دستور Find می‌توانید المان های کنترل، نشان دهنده، دستورها، متون و... را در یک برنامه یا دسته ای از برنامه ها جستجو کنید.
- در حین اجرای هر برنامه از طریق گزینه‌ی Show Profile Window می‌توانید اطلاعاتی در مورد فراخوانیها، زمان‌بندی و ظرفیت اشغال شده‌ی حافظه توسط برنامه و زیربرنامه های آن به دست آورید.

## گزینه‌ی Show VI Hierarchy

به طور کلی استفاده از گزینه‌ی Show VI Hierarchy در برنامه های عظیم و پیچیده بسیار مفید است. این پنجره به شما کمک می کند تا دریابید که فراخوانی زیربرنامه ها از چه نقطه یا نقاطی صورت گرفته است یا اینکه کدام برنامه یا برنامه ها آنها را فراخوانی کرده‌اند.

برای مشاهده‌ی پنجره‌ی Hierarchy Window در یک برنامه‌ی فعال، گزینه‌ی Project >> Show VI Hierarchy را انتخاب کنید. این پنجره تمامی زیربرنامه های فراخوانی شده توسط برنامه‌ی مورد نظر را نشان می دهد.



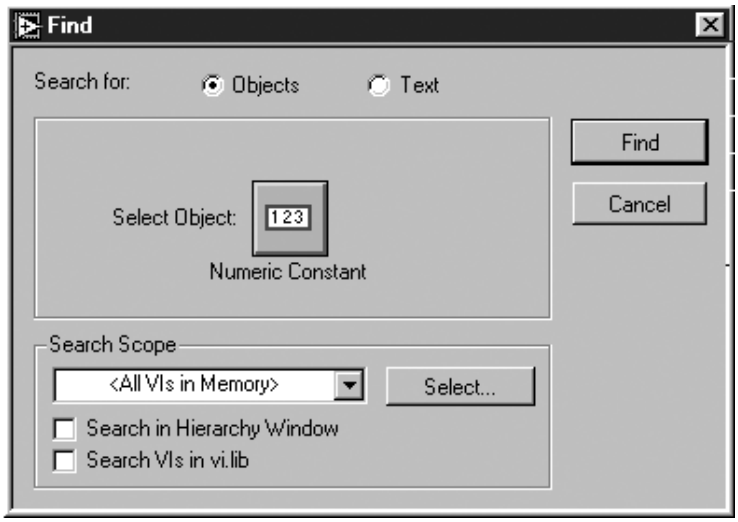

**شکل ۲۷-۱۳ : نمونه‌ای از پنجره‌ی Hierarchy**

۱۳

دکمه های موجود در نوار ابزار این پنجره به کاربر اجازه می دهند تا جنبه های مختلف نمایش زیربرنامه ها را مشاهده کند. به عنوان مثال می توان زیربرنامه های موجود را به صورت افقی یا عمودی نمایش داد و یا برخی از مراتب را مخفی یا ظاهر ساخت. با استفاده از این دکمه ها می توانید ساختار این صفحه را به گونه ای تعیین کنید که تعاریف و متغیرهای سراسری را نیز در برگیرد و یا فاقد موارد مذکور باشد. نکته‌ی جالب و قابل توجه در مورد این پنجره آن است که با دو بار کلیک کردن بر روی آیکن یک زیربرنامه، صفحه‌ی پانل آن ظاهر می شود.

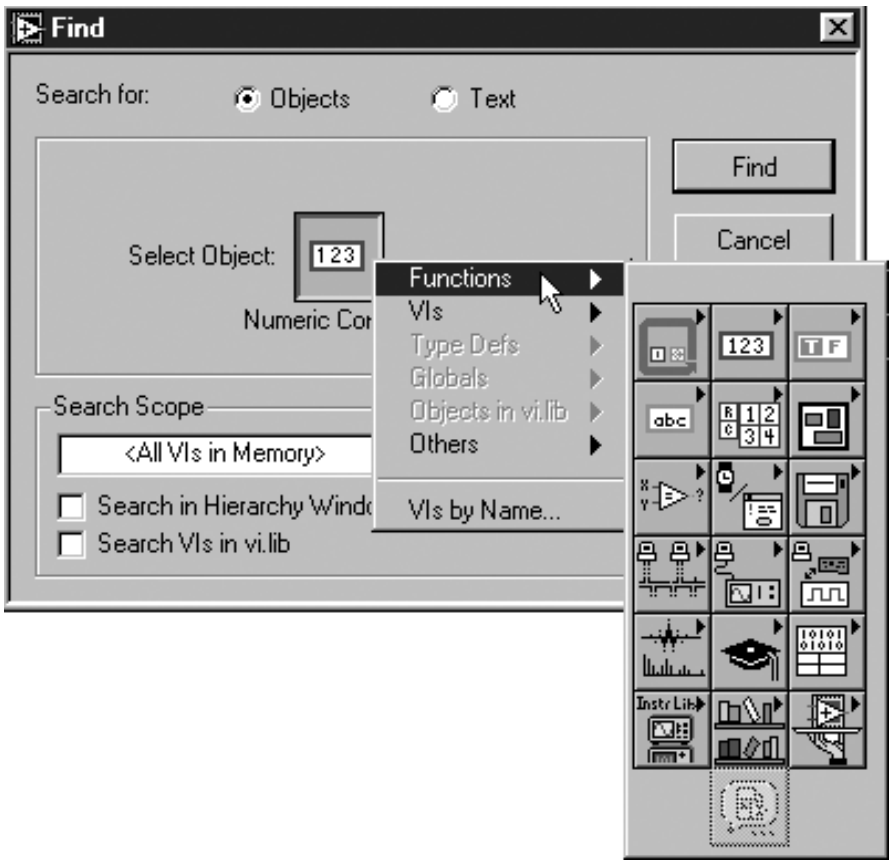
## گزینه‌ی Find

قابلیت جستجوی یک المان در نسخه‌های بالاتر از LabVIEW 4.0 گنجانده شده است. به کمک دستور Project>>Find... می توان هریک از دستورها، زیربرنامه ها، متغیرهای سراسری، گره های خصوصیت، ترمینال یا متن را در صفحه‌ی نمودار بلوکی جستجو کرد و محل آن را تعیین نمود. می توانید عملیات جستجو را در یک برنامه، مجموعه ای از برنامه ها و یا تمام برنامه های موجود در حافظه انجام دهید. در صورت انتخاب این دستور یک پنجره‌ی محاوره ای مطابق شکل ۲۸-۱۳ باز می شود. به عنوان مثال در این پنجره عمل جستجو در مورد المان Numeric Constant انجام گرفته است.




**شکل ۲۸-۱۳: جستجوی المان Numeric Constant در پنجره‌ی محاوره‌ای Find**

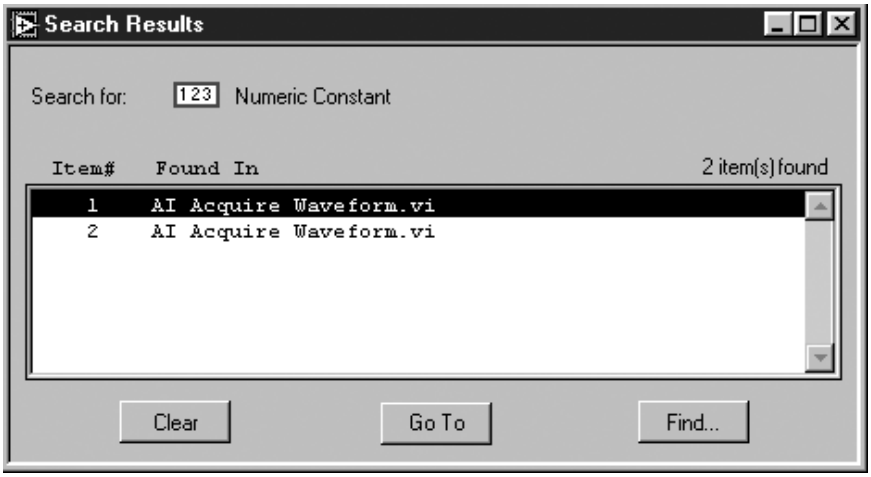
برای جستجوی المان یا متن مورد نظر یکی از دو گزینه‌ی Objects یا Text را در این پنجره انتخاب نمایید. در صورت انتخاب گزینه‌ی Objects بر روی بخش Select Object کلیک کنید و دستور یا سایر موارد مطلوب را از منوی ظاهر شده انتخاب نمایید.




**شکل ۲۹-۱۳: نحوه‌ی انتخاب المان یا دستور مورد نظر در پنجره‌ی محاوره‌ای Find**

در صورتی که متن خاصی را جستجو می کنید متن مورد نظر را تایپ کنید و دکمه‌ی More Options را فشار دهید. در این حالت عملیات جستجو بر روی برچسب المان ها و... نیز انجام می گیرد.

در صورتی که پس از جستجو تنها یک مورد پیدا شده باشد، صفحه‌ی برنامه‌ی حاوی المان یا متن مطلوب باز می‌شود و المان یا متن مورد نظر از سایر اجزای موجود در آن صفحه متمایز می گردد. اما اگر چندین مورد پیدا شده باشد، یک پنجره‌ی محاوره‌ای مطابق شکل ۳۰-۱۳ باز می‌شود و نتایج جستجو را نشان می دهد.



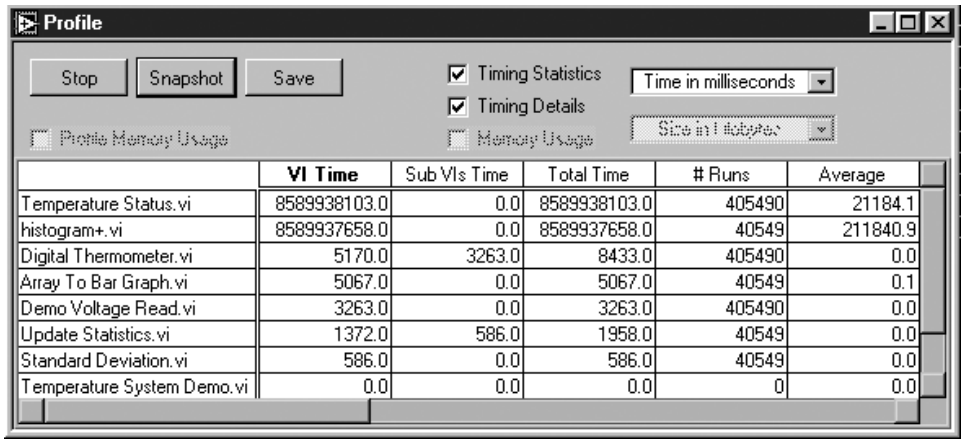

شکل ۳۰-۱۳

به عنوان مثال در شکل ۳۰-۱۳ نتایج حاصل از جستجوی المان Numeric Constant را ملاحظه می کنید. در این حالت از المان مورد نظر تنها در ۲ مورد استفاده شده است. در چنین مواردی می توانید بر روی هر یک از اقلام موجود در پنجره دو بار کلیک کنید و به محل قرار گرفتن آن در برنامه دست یابید.

## گزینه‌ی Show Profile Window

در نرم افزار LabVIEW ابزار قدرتمند و پیشرفته‌ی دیگری به نام Profile Window تعبیه شده است. این پنجره زمان صرف شده جهت اجرای برنامه و همچنین مقدار حافظه‌ی اشغال شده توسط برنامه را نشان می دهد. این اطلاعات در پروژه های وسیع که نیاز به زمان بندی دقیق و استفاده‌ی بهینه از حافظه دارند بسیار مفید است. جهت دستیابی به اطلاعات مذکور، گزینه‌ی Project >> Show Profile Window را در یک برنامه‌ی فعال انتخاب کنید. در این صورت پنجره‌ای باز می شود که در آن تعدادی ستون را با عناوینی نظیر Vi Time، SubVIs Time، Runs و... مشاهده می کنید. هر یک از ستونها بیانگر مشخصات آماری اندازه گیری شده توسط این پنجره است.

سطرهای موجود در این پنجره متناظر با هر یک از زیربرنامه های فراخوانده شده توسط برنامه‌ی اصلی می باشند. برنامه‌ی اصلی یا بالاترین سطح اجرایی با حروف پهن نشان داده می شود. در شکل ۳۱-۱۳ نمونه ای از این پنجره را مشاهده می کنید.




| | VI Time | Sub VIs Time | Total Time | # Runs | Average |
|---|---|---|---|---|---|
| Temperature Status.vi | 8589938103.0 | 0.0 | 8589938103.0 | 405490 | 21184.1 |
| histogram+.vi | 8589937658.0 | 0.0 | 8589937658.0 | 40549 | 211840.9 |
| Digital Thermometer.vi | 5170.0 | 3263.0 | 8433.0 | 405490 | 0.0 |
| Array To Bar Graph.vi | 5067.0 | 0.0 | 5067.0 | 40549 | 0.1 |
| Demo Voltage Read.vi | 3263.0 | 0.0 | 3263.0 | 405490 | 0.0 |
| Update Statistics.vi | 1372.0 | 586.0 | 1958.0 | 40549 | 0.0 |
| Standard Deviation.vi | 586.0 | 0.0 | 586.0 | 40549 | 0.0 |
| Temperature System Demo.vi | 0.0 | 0.0 | 0.0 | 0 | 0.0 |

**شکل ۳۱-۱۳: پنجره‌ی Profile**

برای استفاده از این پنجره مراحل زیر را دنبال کنید:

۱- با انتخاب گزینه های موجود در بالای این پنجره یکی از موارد جمع آوری مشخصات آماری در مورد زمان اجرا یا مقدار حافظه‌ی اشغال شده و یا هر دو را انتخاب کنید. تجزیه و تحلیل مقدار حافظه‌ی اشغال شده، به طور قابل ملاحظه ای سرعت اجرای برنامه را کاهش می دهد.

۲- دکمه‌ی Start را فشار دهید.

۳- اگر برنامه‌ی مورد نظر در حال اجرا نیست آن را به اجرا در آورید.

۴- در زمانی که برنامه در حال اجراست برای مشاهده‌ی مشخصات آماری، دکمه‌ی Snapshot را فشار دهید. برای مشاهده‌ی چگونگی تغیرات مشخصات آماری، چندین دفعه بر روی این دکمه کلیک کنید.

۵- در صورتی که قصد دارید داده های موجود در این پنجره را ذخیره کنید، دکمه‌ی Save را فشار دهید. در این صورت داده های موجود در این پنجره در یک فایل متنی ذخیره می شود.

۶- برای شروع آمارگیری مجدد دکمه‌ی Stop را فشار دهید.

به کارگیری اطلاعات مندرج در پنجره‌ی Profile یک مبحث پیشرفته و مرتبط با استفاده‌ی بهینه از حافظه و همچنین ارتقای راندمان برنامه می‌باشد و از حوصله‌ی این کتاب خارج است.

## خلاصه

این فصل به بحث پیرامون تعدادی از قابلیتهای پیشرفته و مفید نرم‌افزار LabVIEW اختصاص‌داده شده است. در بخشهای مختلف این فصل به بررسی منوهای Perferences، VI Setup، SubVI Node Setup، Key Focus، مبناها و واحدها، نحوه‌ی ایجاد زیربرنامه از بخشی از صفحه‌ی نمودار بلوکی و منوی Project پرداختیم.

گزینه های موجود در منوی Perferences به کاربر اجازه می دهند تا در محیط برنامه نویسی LabVIEW کنترل بیشتری داشته باشد و مواردی نظیر فونت، رنگ، نوع چاپ، مسیرها و... را به دلخواه انتخاب کند.

منوهای VI Setup و SubVI Node Setup حاوی گزینه های مفید و کاربردی جهت سفارشی نمودن جنبه هایی از یک برنامه یا زیربرنامه می باشند. به کمک گزینه های VI Setup می توانید دکمه های موجود در نوار ابزار را ظاهر یا مخفی سازید و نحوه‌ی اجرای برنامه را کنترل نمایید. گزینه های SubVI Node Setup نیز به شما اجازه می دهند تا پنجره‌ی زیربرنامه را به هنگام فراخوانی آن باز کنید و یا آن را ببندید.

Key Focus یک خصوصیت جبری است که در مورد یک المان کنترل به کار می رود. به کمک این خصوصیت می توانید با استفاده از صفحه کلید، المان مورد نظر را انتخاب نمایید و ورودیهای دلخواه را به آن اعمال کنید. همچنین می توانید کلیدهای خاصی نظیر کلیدهای <F1> و <Ctrl> را برای تغییر موقعیت بین المان های کنترل موجود در صفحه‌ی پانل اختصاص دهید.

مبنا و واحد دو ویژگی کاربردی در مورد اعداد می باشند. یک المان عددی را می توان در یکی از مبناهای دو، هشت، ده یا شانزده نمایش داد. واحدهای اختیاری المان های عددی نیز به شما اجازه می دهد تا اعمال ریاضی را بر روی اعداد واحددار انجام داده، تبدیلات خودکار را در سیستم های اندازه گیری مختلف انجام دهید.

خاصیت مدولار بودن LabVIEW به شما اجازه می دهد تا در صورت وجود برخی شرایط، با استفاده از گزینه‌ی Edit >> Create SubVI قسمتهایی از صفحه‌ی نمودار بلوکی را به زیربرنامه تبدیل کنید.

منوی Project شامل تعدادی ابزار پیشرفته نظیر Hierarchy Window، دستور Find و Profile Window می باشد. به کمک Profile Window می توان مقدار حافظه‌ی اشغال شده توسط برنامه و زمان صرف شده جهت اجرای آن را مورد بررسی قرار داد.

۱۳

# ۱۴

# برقراری ارتباط و ورودی/خروجی فایل پیشرفته

## در این فصل با موارد زیر آشنا می‌شوید:

- فایل‌های باینری، متنی و datalog
- نحوه‌ی به‌کارگیری دستورهای I/O سطح پایین
- برقراری ارتباط بین LabVIEW و نرم‌افزارهای دیگر با استفاده از پروتکل‌هایی نظیر DDE
- ایجاد پنجره‌ی Help در مورد برخی از المان‌های موجود در صفحه
- پروتکل‌های شبکه‌ای (نظیر TCP/IP) در LabVIEW
- نحوه‌ی چاپ برنامه

# LabVIEW

# ۱۴ برقراری ارتباط و ورودی/خروجی فایل پیشرفته

در این فصل تعدادی از دستورهای پیشرفته‌ی نرم‌افزار LabVIEW را در رابطه با فایل ها و برقراری ارتباط با رایانه های دیگر فرا خواهید گرفت. در اختیار داشتن اطلاعات کافی از ساختار سه نوع فایل موجود در LabVIEW یعنی فایل متنی، datalog و باینری به شما کمک می کند تا بهترین نوع فایل را در مورد برنامه های خود انتخاب نمایید. LabVIEW همچنین به شما اجازه می دهد تا با به کارگیری قابلیتهای سیستم عامل نظیر DDE (در محیط Windows) و AppleEvents (در محیط Mac) با برنامه های دیگر ارتباط برقرار سازید. به کمک دستورهای داخلی در LabVIEW، برقراری ارتباط با رایانه های دیگر از طریق اینترنت و پروتکل TCP/IP امکان پذیر است. در انتهای این فصل نیز روشهای مختلف چاپ برنامه را بررسی می کنیم.

در این فصل اهداف زیر دنبال می شوند:

- درک تفاوت بین فایل های باینری، متنی و datalog.
- نحوه‌ی به کارگیری دستورهای I/O سطح پایین.
- برقراری ارتباط بین LabVIEW و نرم‌افزارهای دیگر با استفاده از پروتکل هایی نظیر DDE.
- آشنایی با پروتکل های شبکه‌ای (نظیر TCP/IP) در LabVIEW.
- کسب اطلاعات در مورد نحوه‌ی چاپ برنامه.

اصطلاحات کلیدی موجود در این فصل عبارتند از:

- فایل باینری (Binary File)
- فایل متنی (ASCII File)
- فایل Datalog
- پروتکل های TCP/IP، AppleEvents، OLE و DDE

## ورودی/خروجی فایل پیشرفته

در فصل ۹ روش ذخیره‌ی داده‌ها در یک فایل متنی را با استفاده از دستورهای موجود در زیرپالت File I/O مورد بررسی قرار دادیم. یکی از مزایای ذخیره‌ی این داده‌ها در فرمت متنی یا ASCII، سهولت جابه‌جایی آنهاست. هر رایانه‌ای که با هر سیستم عاملی کار کند می‌تواند یک فایل متنی را بخواند یا داده‌ها را در آن ذخیره کند. با این وجود این فایل‌ها دو اشکال اساسی دارند. اولاً فضای زیادی از حافظه را اشغال می‌کنند. ثانیاً در صورتی که داده‌ای که باید ذخیره شود در فرمت متنی نبوده و به عنوان مثال به صورت یک گراف باشد، جهت تبدیل و پردازش آن مدت زمان زیادی لازم است. در این فصل به بررسی دو فایل دیگر جهت ذخیره‌ی داده‌ها می‌پردازیم. این دو فایل عبارتند از: فایل datalog و فایل باینری.

فایل‌های datalog نوع خاصی از فایل‌های باینری هستند که جهت ذخیره نمودن اطلاعات صفحه‌ی پانل مورد استفاده قرار می‌گیرند.

در مورد ذخیره‌ی تمامی اطلاعات صفحه‌ی پانل در فایل‌های datalog می‌توانید فرض کنید که صفحه‌ی برنامه به صورت فشرده در آمده است. در صورت ایجاد یک فایل datalog، تمامی مقادیر المان‌های کنترل و نشان‌دهنده در فایل مذکور ضبط می‌شوند. شایان ذکر است که مقادیر ذخیره شده مربوط به همان لحظه‌ای هستند که فرآیند ثبت داده‌ها صورت گرفته است. در صورتی که این فایل را بعداً در برنامه‌ی خود بارگذاری کنید می‌توانید همان مقادیر ذخیره شده را مجدداً توسط المان‌های موجود درصفحه‌ی پانل برنامه به نمایش در آورید. حتی می‌توانید در یک فایل datalog چند دسته از مقادیر موجود در صفحه را ذخیره کنید. خواندن و نوشتن فایل‌های datalog تنها توسط نرم‌افزار LabVIEW امکان‌پذیر است. کاربرد این فایل‌ها بسیار زیاد است. زیرا می‌توانید بدون نوشتن دستورها و کدهای خاص، اعمال دلخواه را بر روی آنها انجام دهید.

فایل‌های باینری[1] که در برخی موارد به آنها فایل‌های Stream نیز گفته می‌شود شامل تصویر بایت به بایت داده‌های ذخیره شده در حافظه است. تا زمانی که واقعاً از فرمت فایل اطلاع نداشته باشید نمی‌توانید با کمک یک ویرایشگر متنی و یا هر برنامه‌ی دیگر یک فایل باینری را بخوانید. برای خواندن یک فایل باینری باید از نوع و فرمت داده‌ها اطلاع داشته باشید (فصل ۱۲ را ببینید).

مزایای استفاده از فایل باینری آن است که به دلیل عدم نیاز به تبدیل داده‌ها کمترین سربار را به همراه دارد و در مقایسه با فایل متنی فضای کمتری از حافظه را اشغال می‌کند. به عنوان مثال جهت ذخیره‌ی آرایه‌ای حاوی ۱۰۰ عدد ۸ بیتی در یک فایل باینری، تقریباً ۱۰۰ بایت از حافظه اشغال می‌گردد. حال آنکه برای ذخیره‌ی همین آرایه در فایل متنی بیش از ۴۰۰ بایت از حافظه نیاز است. دلیل این تفاوت آن است که برای ذخیره‌ی هر عدد ۸بیتی در فرمت باینری تنها یک بایت به‌کار برده می‌شود. اما همین عدد در فرمت متنی، ۳ یا ۴ بایت از فضای حافظه را اشغال می‌کند (یک بایت برای هر کاراکتر ASCII و سپس در نظر گرفتن یک کاراکتر Space جهت جداسازی هر یک از اعداد). در جدول ۱-۱۴ مزایا و معایب ۳ نوع فایل مذکور نسبت به یکدیگر آورده شده است.

۱- در برخی موارد به این نوع فایل‌ها، فایل‌های Byte Stream نیز گفته می‌شود.

جدول ۱-۱۴: مزایا و معایب فایل های متنی، باینری و datalog نسبت به یکدیگر

| متنی | باینری | Datalog |
|---|---|---|
| سهولت در استفاده<br>سازگار با برنامه‌های کاربردی، سهولت در<br>بررسی و اعمال تغییرات<br>جهت ذخیره و تبدیل داده‌ها مقدار زیادی از<br>ظرفیت دیسک سخت اشغال می گردد.<br>برای مجموعه ی کوچک تا متوسط داده‌هایی که<br>با برنامه‌های کاربردی نظیر برنامه‌های صفحه<br>گسترده مورد استفاده قرار می گیرند ایده‌آل است. | در اختیار داشتن اطلاعات دقیق از نحوه ی ذخیره<br>شدن داده‌ها الزامی است.<br>استفاده ی بهینه از ظرفیت دیسک سخت و<br>همچنین زمان<br>به دلیل وجود قابلیت جاری ساختن داده‌ها در فایل<br>عملکردی سریع دارد.<br>برای ذخیره نمودن داده‌ها به صورت بلادرنگ در<br>فایل‌های حجیم مناسب است. | سهولت در استفاده<br>دسترسی به این فایل‌ها تنها در<br>محیط LabVIEW<br>امکان‌پذیر است.<br>برای ذخیره نمودن اطلاعات<br>صفحه ی پانل یا انواع دیگر<br>داده‌ها بسیا مناسب است. |

## مشخص کردن مسیر برای ذخیره‌ی فایل

همان گونه که در فصول ابتدایی این کتاب ذکر شد برای مشخص کردن محل قرار گرفتن یک فایل در LabVIEW، از مسیر یا Path استفاده می شود. المان کنترل و نشان دهنده‌ی مربوط به مسیر فایل را در شکل ۱-۱۴ ملاحظه می کنید. این المان ها که شباهت بسیار زیادی به المان های کنترل و نشان دهنده‌ی رشته‌ای دارند در زیرپالت Controls >> Path & Refnum قرار گرفته اند.

File Path Control　　File Path Indicator

شکل ۱-۱۴

برخلاف المان های رشته‌ای در یک المان کنترل مربوط به مسیر، تنها می توانید طبق قواعد حاکم در سیستم عامل، مسیر یا نام فایل را وارد کنید. اگر با نحوه‌ی وارد کردن نام فایل و مسیر در سیستم عامل خود آشنایی ندارید، دفترچه‌ی راهنمای سیستم عامل رایانه‌ی خود را مطالعه کنید. در جدول ۲-۱۴ نحوه‌ی مشخص کردن مسیر برای دستیابی به فایل Sample.txt در چند سیستم عامل نشان داده شده است.

۱۴

جدول ۲-۱۴: نحوه‌ی نمایش مسیر برای دستیابی به فایل Sample.txt در چند سیستم عامل

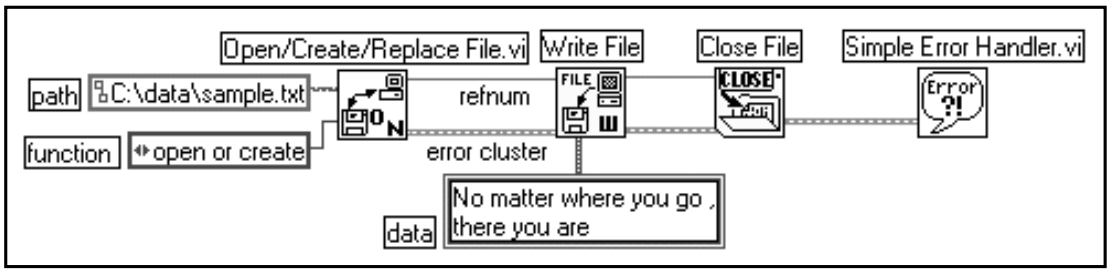
| | |
|---|---|
| Windows | C:\TRAVIS\DATA\SAMPLE.TXT |
| Mac | PowerHD:Travis:Data:Sample Text file |
| Sun and HP | usr/travis/data/sample_text_file |

برای ایجاد یا باز کردن هر فایل ابتدا باید مسیر قرار گرفتن آن را تعیین نمایید. در صورت عدم سیم کشی مسیر در دستورهایی نظیر Open File یا New File، یک پنجره‌ی محاوره‌ای باز می‌شود و از کاربر می خواهد تا مسیر قرار گرفتن فایل را مشخص کند. در صورت به کارگیری دستور File Dialog نیز LabVIEW نام فایل را از کاربر درخواست می کند.

## فرآیند سه مرحله‌ای

هنگام کار با دستورهای File I/O حتماً باید سه مرحله‌ی باز کردن، خواندن یا نوشتن و بستن را دنبال نمایید. برای درک این فرآیند سه مرحله‌ای، یک کشوی پر از پرونده را در نظر بگیرید. در صورتی که بخواهید پرونده‌ای را از کشو خارج کنید یا پرونده‌ای را در داخل آن قرار دهید، ابتدا باید کشوی مربوط را باز کنید. سپس با پرونده‌های موجود کار کنید و اعمال مورد نیاز خود را انجام دهید. پس از اتمام کار آیا درب کشو را باز می‌گذارید؟ ممکن است شخص دیگری برخی از پرونده‌ها را بردارد و یا نظم آنها را بر هم زند. بستن فایل در رایانه نیز نظیر بستن کشوی حاوی پرونده‌ها باعث حصول اطمینان از سلامت و ایمنی فایل‌ها می‌شود.

صفحه‌ی نمودار بلوکی در شکل ۲-۱۴ این فرآیند سه مرحله‌ای را نمایش می‌دهد. تمام توابع و دستورهای Write File، Open/ Create/ Replace File و Close File همگی در زیرپالت Functios >> File I/O قرار دارند. به وابستگی اطلاعاتی[2] موجود در بین دستورها توجه داشته باشید. پس از اتمام عملیات Open/Create/Replace File، LabVIEW یک عدد مرجع فایل یا refrum تولید می‌کند که به دستور Write File انتقال داده می‌شود. هنگامی که رشته به داخل فایل نوشته می‌شود، دستور Write File نیز به نوبه‌ی خود عدد refnum را به دستور Close File انتقال می‌دهد. در پایان این فرآیند، استفاده نمودن از برنامه‌ی Simple Error Handler.vi (Functions >> Time & Dialog) الزامی است. این برنامه به شما کمک می‌کند تا خطاهای احتمالی را در صورت وجود مشاهده کنید.



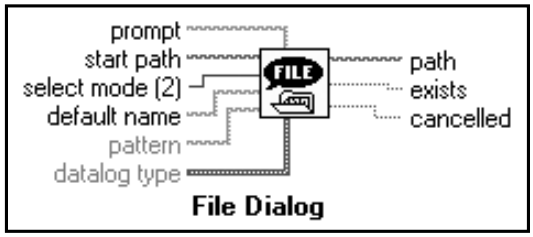

شکل ۲-۱۴

refunm عددی است که جهت دنبال نمودن و یافتن فایلی که در برنامه فراخوانی شده است به کار برده می‌شود. لزومی ندارد تا از ماهیت یا عملکرد این اعداد اطلاع پیدا کنید. تنها کافی است که این اعداد را به دستورهای File I/O سیم کشی کنید. سیم کشی ترمینال های refnum در بین این برنامه‌ها علاوه بر مزایای فوق، ویژگی وابستگی اطلاعاتی را نیز به ارمغان می‌آورد. بدین معنی که اعمال مورد نظر به صورت خودکار و با ترتیب صحیح انجام می‌گیرند. در ادامه‌ی بحث به بررسی تعدادی از دستورهای File I/O می‌پردازیم:

دستور File Dialog (Functions >> File I/O >> Advanced File Functions) یک پنجره‌ی محاوره‌ای برای انتخاب فایل ظاهر می‌سازد. این پنجره برای انتخاب فایل‌ها یا فهرست های جدید یا موجود مورد استفاده قرار می‌گیرد. در ورودی prompt می‌توانید یک پیغام دلخواه تعیین کنید.

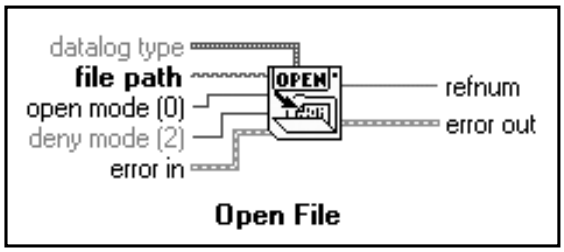
۱۴

2-Data Dependency



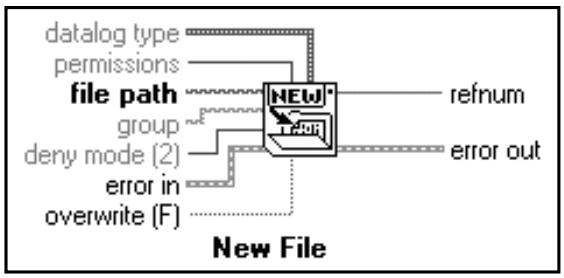

شکل ۳-۱۴

دستور Open File (Functions >> File I/O >> Advanced File Functions) جهت باز کردن یکی از فایل های موجود به کار برده می شود. جهت دستیابی و باز کردن فایل باید مسیری معتبر و صحیح را به ورودی file path اعمال کنید. انجام اعمالی نظیر ایجاد و یا جابه‌جا نمودن فایل توسط این دستور امکان پذیر نیست. همان گونه که از نام این دستور استنباط می گردد، از دستور مذکور تنها جهت باز کردن فایل های موجود استفاده می شود. ورودی datalog type تنها در صورتی مورد استفاده قرار می گیرد که قصد باز کردن فایل های datalog در LabVIEW را داشته باشید.



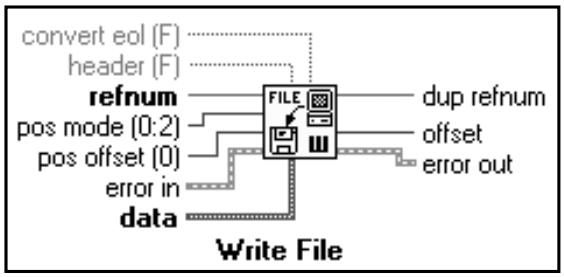

شکل ۴-۱۴

دستور New File (Functions >> File I/O >> Advanced File Funotions) یک فایل جدید را ایجاد می‌نماید، سپس آن را برای خواندن و یا نوشتن باز می کند. برای ایجاد یک فایل جدید باید نام فایل سیم کشی شده در ورودی file path نامی غیر از نام فایل های موجود باشد. ورودی datalog type تنها در صورتی مورد استفاده قرار می گیرد که قصد داشته باشید فایل های datalog را در LabVIEW ایجاد نمایید.



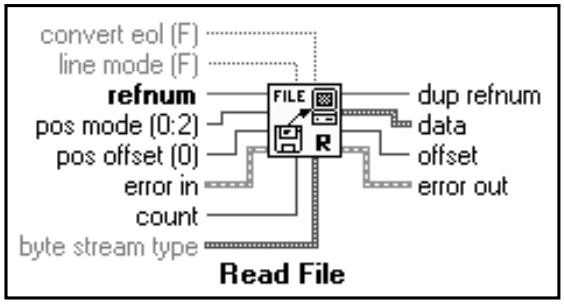

شکل ۵-۱۴

دستور Write File (Functions >> File I/O) داده ها را در یک فایل باز شده می نویسد. عملکرد این دستور، بسته به نوشتن داده ها در یک فایل باینری یا datalog متفاوت است. ورودی header تنها در مورد فایل های باینری استفاده می‌شود و در مورد فایل های متنی از سیم کشی این ورودی صرف نظر می شود.



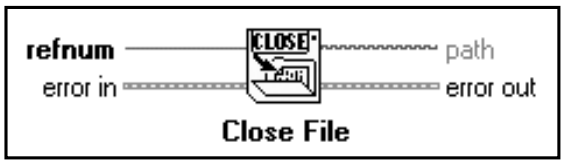

شکل ۶-۱۴

دستور Read File (Functions >> File I/O) داده ها را از یک فایل باز شده می خواند. در صورت قرائت داده ها از یک فایل باینری حتماً باید از ورودی byte stream type استفاده کنید. در بخشهای بعد، فایل باینری را مورد بررسی قرار خواهیم داد.



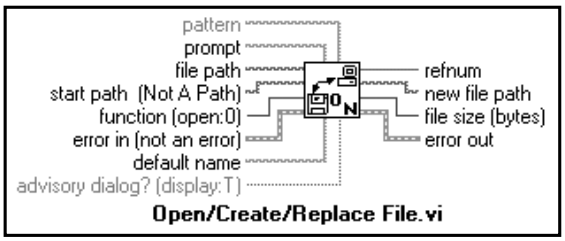

شکل ۷-۱۴

دستور Close File (Functions >> File I/O) فایلی را که با شماره‌ی refnum مشخص شده است می بندد.




شکل ۸-۱۴

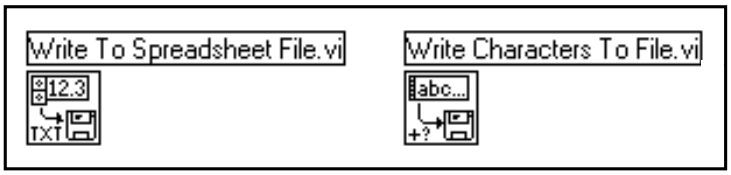
دستور Open / Create / Replace File (Functions >> File I/O) یک دستور جانبی و کمکی است که به شما اجازه می دهد تا یک فایل را باز نموده، یک فایل جدید ایجاد نمایید یا یک فایل جدید را با نام قدیمی آن جایگزین کنید. این دستور معمولاً پیش از سایر دستورهای ذکر شده، فراخوانی و استفاده می شود.




شکل ۹-۱۴

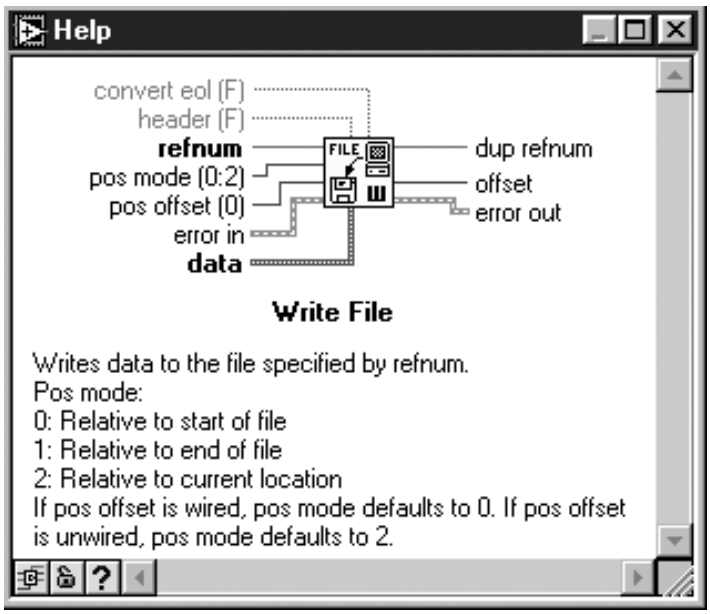
## خواندن و نوشتن فایل های متنی

اگر قصد دارید برای ذخیره‌ی داده ها و اطلاعات خود از فایل های متنی یا ASCII استفاده کنید و در انجام این عمل سرعت بالا نیاز دارید، به شما توصیه می کنیم تا یکی از دو برنامه‌ی نشان داده شده در شکل ۱۰-۱۴ را به کار گیرید. در صورتی که به یاد داشته باشید این برنامه ها را در فصل ۹ مورد بررسی قرار دادیم. این دو برنامه تمامی اعمال اعم از باز کردن و بستن فایل را انجام می‌دهند، همچنین وجود یا عدم وجود خطا را بررسی می کنند.




شکل ۱۰-۱۴

در برخی موارد لازم است تا داده ها را بر روی فایل متنی بنویسید. این عمل توسط دستور Write File انجام می گیرد.




شکل ۱۱-۱۴

ورودی data در دستور Write File هر نوع داده اعم از داده های عددی، رشته ای، کلاستر یا... را می پذیرد. به عبارت دیگر این ورودی دارای ویژگی Polymorphism است. نوع داده‌ی سیم کشی شده به این دستور تعین کننده‌ی آن است که فایل مورد نظر با چه فرمتی استفاده می شود. واضح است که برای فایل های متنی باید یک رشته را به این ورودی سیم کشی کنید.

## مدیریت فایل در فایل های متنی

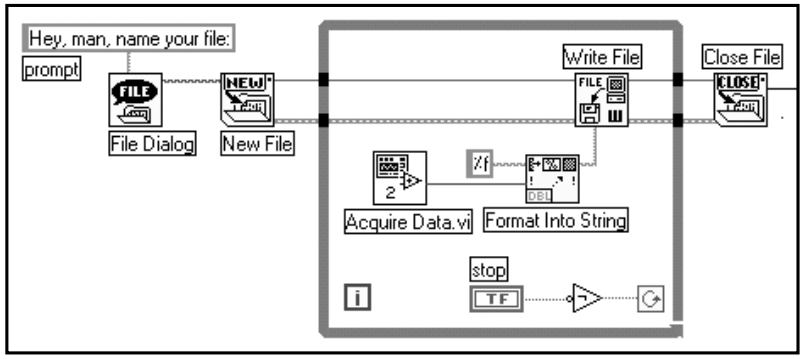
مدیریت فایل در مورد فایل های متنی یا ASCII بسیار ساده است. این عمل شامل بازکردن، نوشتن و بستن فایل می باشد. اما در هنگام استفاده از این فایل ها ممکن است سؤالات زیر در ذهن شما ایجاد شود.

۱- مسیر و نام فایل متنی ازچه طریقی به دست می آید؟ آیا نرم افزار LabVIEW به طور خودکار آن را ایجاد می نماید یا اینکه کاربر از طریق صفحه‌ی پانل یا پنجره‌ی محاوره‌ای دیگری آن را وارد می کند؟

۲- خطاهایی نظیر پر شدن ظرفیت دیسک، مسیر نادرست و... در یک برنامه چگونه تشخیص داده می شود؟ آیا در این حالت برنامه متوقف می گردد؟ یا اینکه اجرای برنامه ادامه می‌یابد و کاربر توسط یک پیغام از وجود خطا آگاه می شود؟

مطمئناً همگی شما با روش تبدیل داده ها به رشته آشنایی دارید. خوشبختانه در نرم افزار LabVIEW تعداد زیادی دستور جهت تبدیل داده های رشته‌ای و انجام عملیات بر روی آنها وجود دارد (زیرپالت Functions >> String را بررسی نمایید یا به فصل ۹ مراجعه کنید). اکنون فرض کنید که یک برنامه‌ی ساده را با عنوان Acquire Data.vi ایجاد نموده اید. این برنامه ابتدا داده ها را می‌خواند و سپس متناظر با هر داده، یک عدد اعشاری به دست می دهد. نحوه‌ی ذخیره‌ی داده ها به صورت زیر است:

تا زمانی که دکمه‌ی Abort در نوار ابزار فشار داده نشود، فرآیند DAQ و جاری ساختن داده ها به فایل ادامه می‌یابد و کاربر می تواند نام فایل ذخیره کننده‌ی داده ها را در حین و یا قبل از اجرای برنامه تعیین کند. در شکل ۱۲-۱۴ صفحه‌ی نمودار بلوکی این برنامه را ملاحظه می کنید.




شکل ۱۲-۱۴

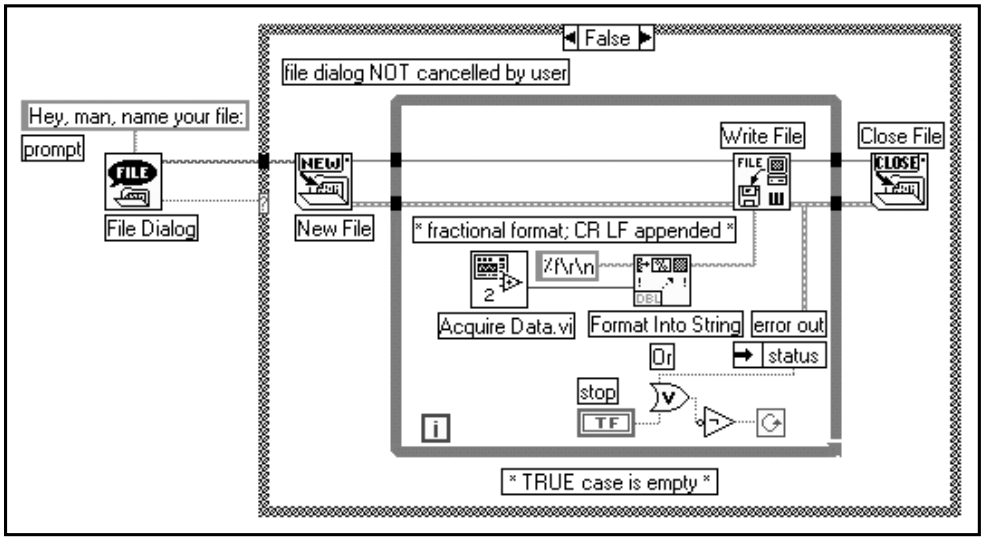
روند اجرای برنامه نیز به صورت زیر است:

ابتدا از کاربر درخواست می شود تا نام یک فایل را وارد کند. سپس فایل مورد نظر ایجاد می‌شود و اجرای حلقه‌ی While آغاز می گردد. در پایان اجرای هر تکرار حلقه، با استفاده از دستور Format Into String تنها یک داده به فرمت رشته ای تبدیل می‌شود و به داخل فایل مشخص شده نوشته می شود. پس از فشرده شدن دکمه‌ی Abort اجرای حلقه‌ی While پایان می‌یابد و فایل مذکور بسته می شود.

بسیار ساده است. آیا این‌طور نیست؟ این برنامه می تواند به تنهایی مورد استفاده قرار گیرد. اما مدیریت فایل

در این برنامه کمی ضعیف است که می توان آن را به روش زیر ارتقا داد. خوانندگان مبتدی توجه داشته باشند که در این برنامه هیچ عملی جهت جداسازی هر یک از رشته‌ها که بیانگر یک داده‌ی مجزا می باشد، انجام نگرفته است. در حقیقت تعدادی رشته‌ی درهم ریخته و مخلوط در این فایل ذخیره شده‌اند. در این حالت اگر کاربر دکمه‌ی Cancel را در پنجره‌ی محاوره‌ای دستور File Dialog فشار دهد چه اتفاقی روی می‌دهد؟

در صفحه‌ی نمودار بلوکی شکل ۱۳-۱۴، با افزودن برخی از امکانات و قابلیتهای دیگر در این برنامه بر قدرت مدیریت فایل افزوده شده است. توجه داشته باشید که در این برنامه کاراکترهای carriage return و linefeed (کاراکترهای \n و \r) به هر یک از اعداد اضافه می شوند. این عمل باعث می شود تا هر عدد در انتهای سطر فایل مذکور قرار گیرد. ما احتمال فشرده شدن دکمه‌ی Cancel توسط کاربر را نیز با اضافه کردن یک ساختار شرطی براساس خروجی دستور File Dialog در نظر گرفته ایم. جهت بررسی وجود خطای ورودی /خروجی فایل نیز یک کلاستر خطا تعبیه شده است که در صورت وجود خطا، برنامه را متوقف خواهد کرد.




**شکل ۱۳-۱۴**

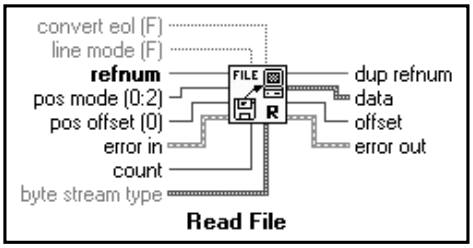
۱۴

اما هنوز در رابطه با این برنامه یک مشکل دیگر وجود دارد که قابل تأمل است. این مسأله در رابطه با دستور Write File می باشد که در داخل حلقه‌ی While قرار گرفته است. آیا می توانید این مشکل را حدس بزنید؟

بله. درست حدس زده اید. تا زمانی که دستور Wait در برنامه‌ی Acquire Data.vi وجود دارد این برنامه به هر تعداد دفعه‌ای که بتواند از پردازنده استفاده کند، داده ها را در فایل می نویسد و ممکن است تا هزار بار در ثانیه موفق به انجام این عمل گردد. شما باید برای محدود نمودن تعداد دفعات مورد نیاز جهت نوشتن داده ها در فایل، از یک دستور Wait در داخل حلقه استفاده کنید. این دستور سبب می شود تا نوشتن داده ها در فایل در هر ثانیه یک مرتبه و یا در حدی قابل قبول و معقول صورت گیرد. راه حل بهتر نیز آن است که داده های جمع آوری شده را با استفاده از خاصیت اندیس گذاری خودکار در یک آرایه ذخیره کنید و همه را یک‌باره در خارج از حلقه‌ی While بر روی فایل بنویسید.

همان‌طور که ملاحظه می‌کنید در هنگام انجام عملیات با دستورهای File I/O ممکن است مشکلات و خطاهای بسیار زیادی روی دهد. در اکثر موارد برای خواندن و نوشتن فایل‌های متنی می‌توانید از برنامه‌های ساده‌ی موجود در زیرپالت File I/O استفاده نمایید. اگر واقعاً قصد دارید که از فایل‌های متنی استفاده کنید، حتماً تمامی شرایط را در نظر بگیرید.

خواندن یک فایل متنی شبیه به نوشتن بر روی آن انجام می گیرد. دستور Read File را به همان روش استفاده از دستور Write File در برنامه ها به کار برید. در شکل ۱۴-۱۴ این دستور را به همراه ورودیها و خروجیهای آن مشاهده می کنید.

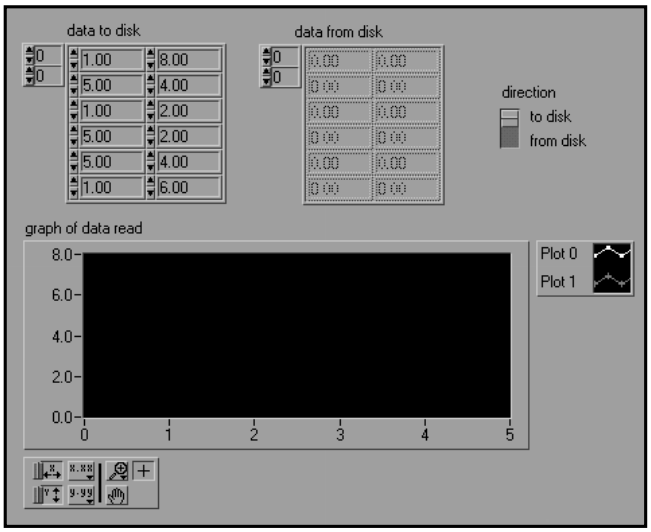



شکل ۱۴-۱۴

در صورت عدم سیم‌کشی به ورودی count اطلاعات کل فایل خوانده می‌شود. عدد سیم‌کشی شده در ورودی count تعداد بایت هایی را که باید خوانده شوند تعیین می کند.

## تمرین ۱-۱۴: خواندن و نوشتن فایل های متنی

برنامه‌ای ایجاد کنید که قابلیت خواندن و نوشتن فایل‌های متنی را داشته باشد. داده‌ی ورودی یک آرایه‌ی دوبعدی است که مؤلفه‌های آن را اعداد تشکیل می‌دهند. خواندن از /نوشتن به یک فایل از طریق وضعیت یک کلید جبری که در صفحه‌ی پانل قرار گرفته است تعیین می گردد. صفحه‌ی پانل این برنامه در شکل ۱۵-۱۴ نشان داده شده است.




شکل ۱۵-۱۴: صفحه‌ی پانل در برنامه‌ی ASCII Read/Write.vi

این برنامه را با عنوان ASCII Read / Write.vi در زیرفهرست LabVIEW\My Activity ذخیره کنید. پاسخ این تمرین در مسیر LabVIEW\Exercise\Ch14.llb گنجانده شده است.

برای خوانـــدن فایل، از برنـامه‌های سـاده‌ی موجـود در زیـرپالت Functions >> File I/O (نظیر Read Characters From File.vi) استفاده کنید.

## خواندن و نوشتن فایل‌های Datalog

فایل‌های datalog داده‌های المان‌های موجود در صفحه‌ی پانل را در یک فرمت باینری خاص ذخیره می‌کنند. هرگاه که به نوشتن در یک فایل datalog اقدام کنید، LabVIEW به طور معمول داده‌ها را ضبط می‌کند و در حقیقت اطلاعات ضبط شده از المان‌ها را در فایل datalog می‌نویسد. هر دسته از مجموعه‌ی اطلاعات ذخیره شده را یک «رکورد[3]» گویند. رکورد یک نوع فایل است که در داخل فایل دیگری قرار گرفته است. سیستم عامل رایانه تنها فایل اصلی را تشخیص می‌دهد. اما دستیابی به رکوردهای ذخیره شده‌ی درون فایل اصلی تنها به کمک نرم افزار LabVIEW امکان پذیر است. نکته‌ی جالب توجه در مورد این ساختار آن است که دسترسی تصادفی به رکوردهای یک فایل نیز امکان‌پذیر است. مزیت دیگر استفاده از رکوردها آن است که برای دستیابی به مجموعه‌ای خاص از داده‌ها تنها کافی است از شماره‌ی رکورد اطلاع داشته باشید.

فایل‌های datalog به خصوص برای ذخیره‌ی داده‌های مرکب نظیر آرایه‌ها بسیار مفید و مناسب‌اند. برای ایجاد یک فایل datalog دو روش زیر وجود دارد:

۱- ثبت تمامی اطلاعات صفحه‌ی پانل: برای انجام این عمل از ویژگیها و قابلیتهای داخلی ثبت اطلاعات که در منوی Operate گنجانده شده‌اند کمک بگیرید. دستورهای موجود در این منو به شما اجازه می‌دهند تا بدون نیاز به استفاده از دستورهای File I/O تمامی داده‌های صفحه‌ی پانل را در یک فایل ثبت کنید. با استفاده از گزینه‌های موجود در این منو می‌توانید داده‌های صفحه‌ی پانل را در مرحله‌ی اتمام برنامه و یا در اواسط اجرای برنامه ذخیره نمایید.

۲- ثبت اطلاعات المان‌های دلخواه موجود در صفحه‌ی پانل: در این حالت می‌توانید مشخص کنید که به جای ذخیره‌ی تمامی اطلاعات صفحه‌ی پانل، داده‌های کدام المان/المان‌ها ذخیره شوند. در ضمن در این حالت می‌توانید محل ذخیره شدن داده‌ها را نیز تعیین کنید. انجام این عمل با استفاده از دستورهای Read File و Write File امکان‌پذیر است.



## ثبت تمامی اطلاعات صفحه‌ی پانل

ثبت تمامی اطلاعات صفحه‌ی پانل بسیار ساده است. به هنگام ثبت داده‌ها، LabVIEW داده‌های تمامی المان‌های کنترل و نشان‌دهنده‌ی موجود در صفحه‌ی پانل را به همراه زمان ثبت آنها در یک

3- Record

فایل datalog ذخیره می کند. در این حالت می توانید چندین فایل مجزا داشته باشید که درون هر یک، داده های ذخیره شده یا به عبارت دیگر رکوردهای زمانهای مختلف قرار گرفته باشد. به بیان دیگر در یک فایل می توانید داده های مربوط به زمانهای مختلف را ثبت کنید. سپس در صورت نیاز می توانید به راحتی این اطلاعات را استخراج کنید. در ضمن می توانید این داده ها را با استفاده از دستورهای File I/O در برنامه های دیگر نیز به کار برید.

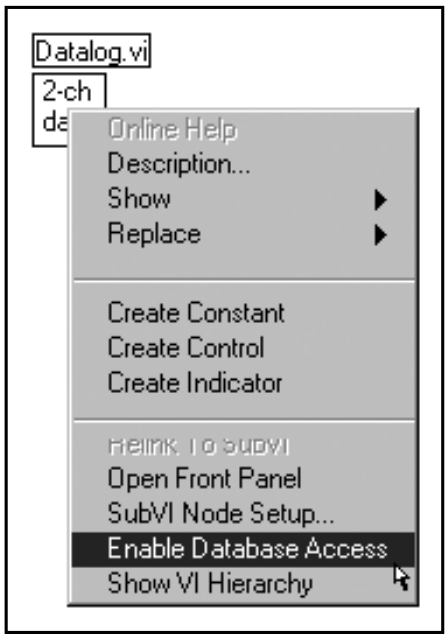
برای ثبت کل اطلاعات صفحه‌ی پانل گزینه‌ی ...Log << Data Logging << Operate را انتخاب کنید. در صورتی که از این دستور برای اولین بار استفاده می کنید یک پنجره‌ی محاوره‌ای باز می‌شود و نام فایل مورد نظر که این داده ها باید در آن ذخیره شوند از کاربر درخواست می شود. برای ثبت اطلاعات در زمان اتمام اجرای برنامه گزینه‌ی Operate >> Log at Completion را انتخاب کنید. هر بارکه داده ها در یک فایل datalog ثبت می شوند، در حقیقت یک رکورد جدید در آن فایل ایجاد می گردد.

برای بررسی و مشاهده‌ی داده های ثبت شده، گزینه‌ی ...Retrieve << Data logging << Operate را انتخاب کنید. در این حالت شکل ظاهری نوار ابزار مطابق شکل ۱۶-۱۴ تغییر می کند. در ضمن مقادیر تمامی المان های کنترل و نشان دهنده‌ی موجود در صفحه‌ی پانل نیز تغییر می‌کنند و مقادیر ثبت شده را نشان می دهند.

**شکل ۱۶-۱۴**

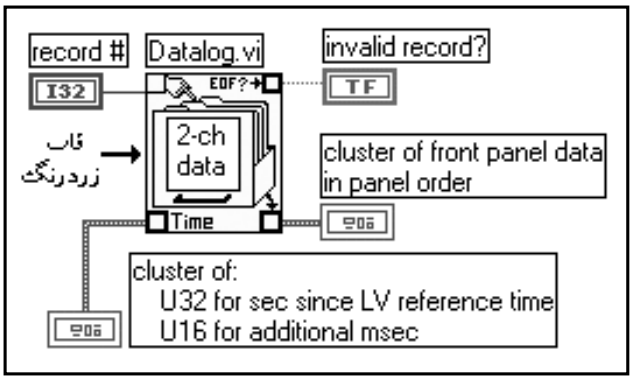
در شکل ۱۶-۱۴، شماره‌ی مندرج در داخل کادر موجود در نوار ابزار به کاربر اطلاع می دهد که در حال حاضر اطلاعات کدامین رکورد بر روی صفحه‌ی پانل به نمایش در آمده است. آخرین شماره‌ی قرار گرفته در براکت، نشان دهنده‌ی تعداد رکوردهای ذخیره شده در فایل است. به عنوان مثال عدد 1 در براکت موجود در شکل ۱۶-۱۴ نشان می دهد که تاکنون 2 رکورد یعنی رکوردهای 0 و 1 در این فایل ذخیره شده‌اند. برای تغییر وضعیت به رکوردهای دیگر و در نتیجه مشاهده‌ی داده های ثبت شده‌ی دیگر از پیکانهای موجود در نوار ابزار استفاده نمایید. به محض تغییر شماره‌ی رکورد، المان های موجود در صفحه‌ی پانل نیز تغییر می‌کنند و داده های متناظر با رکورد مورد نظر را نمایش می دهند. در سمت راست براکت و در انتهای نوار ابزار، زمان و تاریخ ثبت داده ها نشان داده می شود.

برای حذف رکوردهای مجزا شماره‌ی رکورد را انتخاب نموده، سپس بر روی دکمه ای که به شکل سطل زباله است کلیک کنید. برای خروج از این صفحه بر روی کلید OK کلیک نمایید. روش دیگر جهت بررسی و استخراج داده های ثبت شده در یک فایل datalog، در حالتی است که این برنامه به صورت زیربرنامه استفاده شده باشد. برای انجام این عمل مطابق شکل ۱۷-۱۴، گزینه‌ی Enable Database Access را از منوی کرکره ای زیربرنامه انتخاب کنید.




شکل ۱۷-۱۴

در صورت انتخاب این گزینه، مطابق شکل ۱۸-۱۴ یک قاب زردرنگ در اطراف آیکن این برنامه ظاهر می گردد.




شکل ۱۸-۱۴

۱۴

در این حالت اگر این زیربرنامه را اجرا کنید، به اجرا در نمی‌آیند و در عوض، داده‌های ذخیره شده از پنجره‌ی پانل خود را با توجه به شماره‌ی رکوردی که به ورودی سیم کشی شده است در قالب یک کلاستر به دست می دهد. این کلاستر حاوی تمامی داده های صفحه‌ی پانل است. ترتیب قرار گرفتن اجزای کلاستر نظیر ترتیب قرار گرفتن المان ها در صفحه‌ی پانل است.

## ثبت اطلاعات برخی از المان‌های موجود در صفحه‌ی پانل

علاوه بر ذخیره‌ی کل اطلاعات موجود در صفحه‌ی پانل می توان به کمک دستورهای Read File و Write File ثبت داده ها را در سطح بالاتری انجام داد. به کمک این دستورها نیازی به ذخیره‌ی تمامی داده های صفحه‌ی

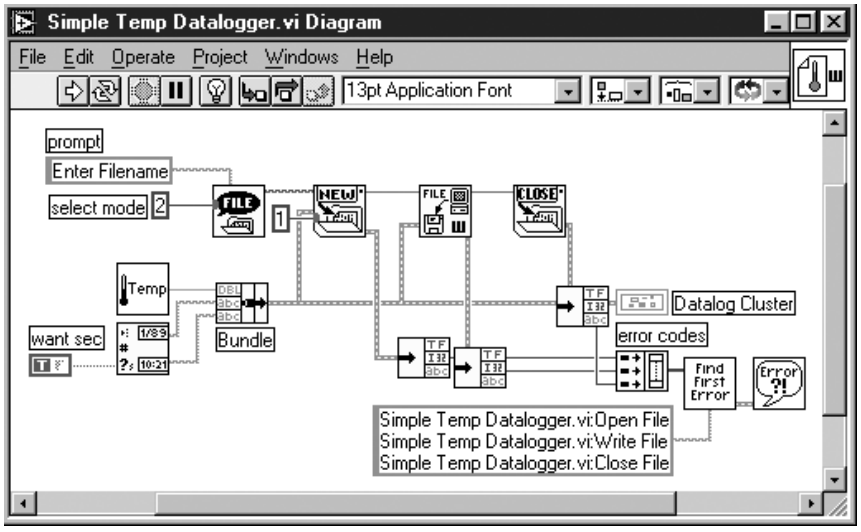
پانل نیست بلکه می توانید تنها داده های برخی از المان ها را ثبت کنید. در حقیقت لزومی ندارد که داده‌ی مورد نظر حتماً بر روی صفحه‌ی پانل قرار گرفته باشد، بلکه ممکن است این داده در صفحه‌ی نمودار بلوکی ایجاد شده باشد.

اگرچه تنها می توان یک نوع داده را در یک فایل datalog ذخیره و ثبت نمود، اما شما می توانید چندین متغیر مختلف را در یک کلاستر دسته‌بندی کنید یا اینکه داده ها می توانند از نوع آرایه، رشته، جبری یا عددی باشند.

در صورت استفاده از فایل های datalog همواره ورودی data type را در تمامی دستورهای فایل نظیر New File، Open File و... به استثنای دستور Read File سیم کشی کنید. در مورد دستور Read File باید از سیم کشی این ورودی خودداری کنید تا بدین ترتیب بتوانید یک فایل datalog را مشخص سازید.

در مورد فایل های باینری، ورودی data type را به یکی از دو دستور Read File یا Write File سیم کشی کنید. در مورد این فایل ها باید از سیم کشی ورودی data type در دستورهای New File، File Dialog و... خودداری کنید تا بدین ترتیب بتوانید یک فایل باینری را مشخص سازید.

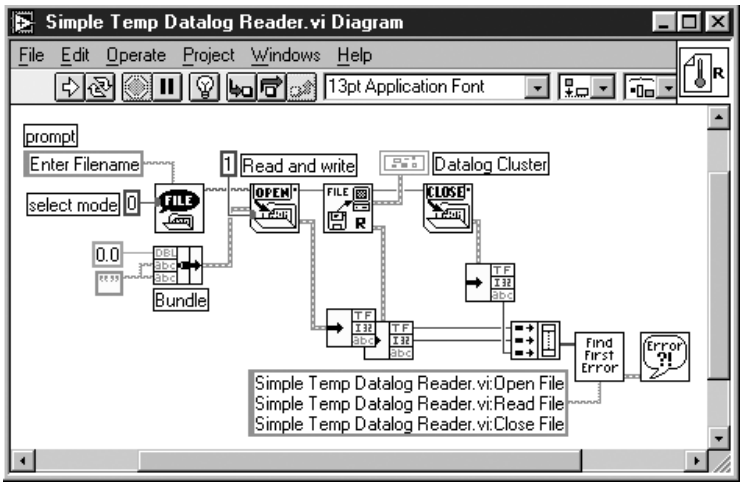
در شکل ۱۹-۱۴ صفحه‌ی نمودار بلوکی یک برنامه‌ی ساده‌ی datalog با عنوان Simple Temp Datalogger.vi نشان داده شده است. این برنامه در مسیر LabVIEW\Examples\File\datalog.llb قرار دارد.




**شکل ۱۹-۱۴: صفحه‌ی نمودار بلوکی در برنامه‌ی Simple Temp Datalogger.vi**

این برنامه فایل های datalog را ذخیره می کند. در این فایل ها درجه حرارت قرائت شده به همراه تاریخ و زمان قرائت داده ها ثبت می شوند. شایان ذکر است که این داده ها به صورت عددی هستند و به همین صورت نیز ذخیره و ثبت می شوند.

برای استخراج و مشاهده‌ی داده‌های ذخیره شده در فایل های datalog از برنامه‌ی Simple Temp Datalog Reader.vi استفاده کنید. صفحه‌ی نمودار بلوکی این برنامه را در شکل ۲۰-۱۴ ملاحظه می کنید.

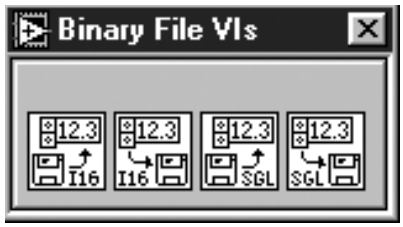



**شکل ۲۰-۱۴: صفحه‌ی نمودار بلوکی در برنامه‌ی Simple Temp Datalog Reader.vi**

در شکل ۲۰-۱۴ دقت کنید و ببینید که چگونه از ثابت مناسب یعنی ثابت کلاستری جهت آماده‌سازی ورودی data type استفاده شده است. در این موارد کاملاً مراقب باشید. زیرا لازم است که حتماً نوع داده‌ای را که در ابتدای امر ذخیره شده است به این ورودی سیم کشی کنید. برای اطمینان از دستیابی به متغیر صحیح و مورد نظر، اطلاع داشتن از ترتیب قرار گرفتن المان ها درکلاستر از اهمیت بالایی برخوردار است.

## خواندن و نوشتن فایل های باینری

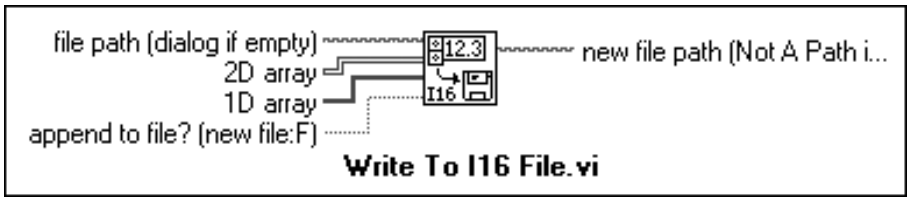
فایل های باینری نوع خاصی از فایل های متنی هستند که از لحاظ حجم بسیار کوچکتر و از لحاظ عملکرد تبدیل یا ذخیره‌ی داده ها بسیار سریعتر می باشند. اشکال استفاده از این نوع فایل ها آن است که برای خواندن داده های یک فایل، اطلاع داشتن از جزییات روش نوشته شدن فایل الزامی است. در غیر این صورت باید از خواندن فایل صرف نظر کنید. زیرا در فایل های باینری هیچ گونه اطلاعات ارزشمندی در مورد نوع داده و یا علائم قراردادی نظیر عنوان[۴] وجود ندارد تا به کمک آنها بتوانید از محتویات فایل آگاه شوید. به طور خلاصه می توان گفت که استفاده از فایل های باینری در مقایسه با سایر فایل ها چندان ساده نیست. در بسته‌ی نرم‌افزاری LabVIEW برنامه هایی جهت خواندن و نوشتن داده های عددی در فایل باینری تعبیه شده است. این دستورها در زیرپالت Functions >> File I/O >> Binary File VIs قرار دارند.




**شکل ۲۱-۱۴: زیرپالت Binary File VIs**

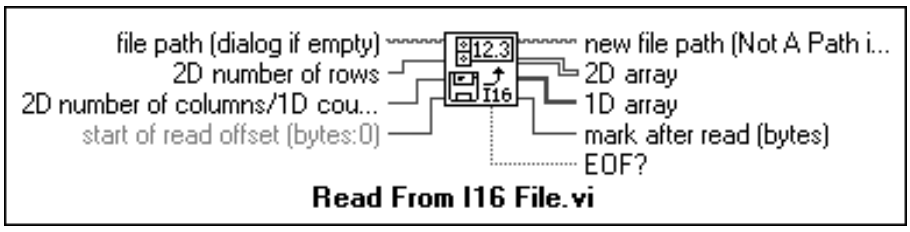
اولین و دومین زوج از این برنامه ها به ترتیب جهت خواندن و نوشتن داده های عددی با فرمت I16 و SGL به کار می روند. در تمامی این برنامه ها فرض بر آن است که داده ها به صورت آرایه ای وجود دارند. در ادامه به بررسی این برنامه ها می پردازیم:

برنامه‌ی Write To I16 File.vi یک آرایه‌ی یک بعدی یا دو بعدی از اعداد صحیح بدون علامت با فرمت I16 را به فایل باینری که مسیر آن در ورودی file path مشخص شده است می نویسد. با انتساب مقادیر جبری False یا True به ورودی append to file? می توانید یک فایل جدید ایجاد نمایید یا اینکه داده ها را به یک فایل قدیمی الحاق کنید.



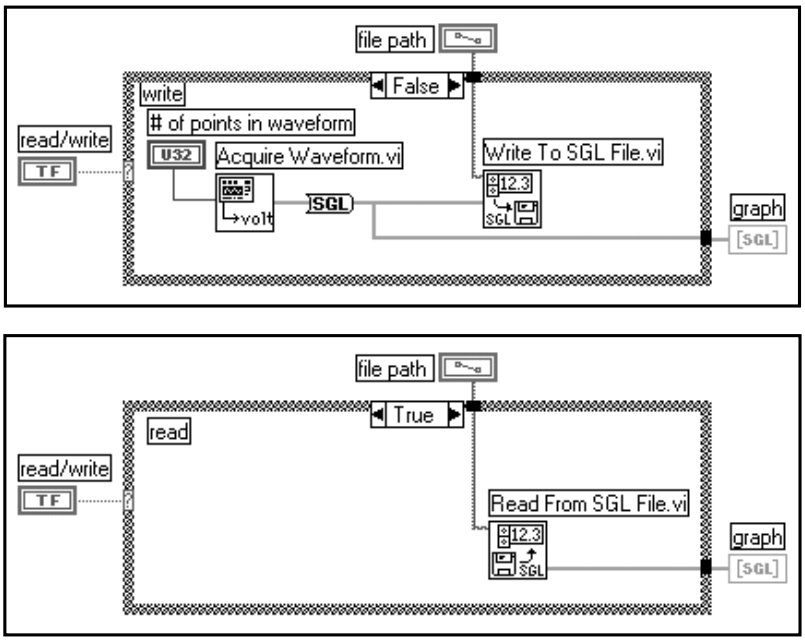

شکل ۲۲-۱۴

برنامه‌ی Read From I16 File.vi یک فایل باینری را که داده های آن، آرایه ای از اعداد صحیح بدون علامت با فرمت I16 می باشد می‌خواند. در صورتی که از تعداد سطرها و ستونهای آرایه‌ی دوبعدی که در خروجی ظاهر می‌شود اطلاع دارید این مقادیر را به ورودیهای مربوط سیم کشی کنید تا بدین ترتیب اندازه‌ی صحیح آرایه را در خروجی 2D array به دست آورید. در غیر این صورت از سیم کشی این ورودیها صرف نظر کنید تا بدین ترتیب تمامی محتویات فایل، خوانده شوند و در خروجی 1D array ظاهر شوند. برای دستیابی تصادفی به داده های فایل، شماره‌ی بایت مورد نظر را برای آغاز نمودن فرآیند خواندن داده ها به ورودی start of read offest سیم کشی کنید.




شکل ۲۳-۱۴

عملکرد دو برنامه‌ی دیگر یعنی Write To SGL File.vi و Read From SGL File.vi با دستورهای مذکور کاملاً مشابه است. با این تفاوت که در این برنامه تمامی عملیات بر روی داده های عددی با فرمت SGL صورت می گیرد. برای درک عملکرد این دستورها، برنامه‌ی Binary Read/Write.vi را از کتابخانه‌ی Ch14.llb ملاحظه کنید. این برنامه داده های مربوط به یک شکل موج را جمع آوری می‌نماید و آنها را در یک فایل ذخیره می کند. همچنین به کمک این برنامه می توانید این داده ها را از فایل استخراج کنید. در حالت دوم، داده ها به یک گراف انتقال داده می‌شوند و بر روی آن ترسیم می شوند. در شکل ۲۴-۱۴ صفحه‌ی نمودار بلوکی این برنامه نشان داده شده است.

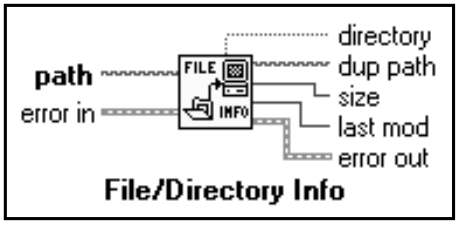



**شکل ۲۴-۱۴: صفحه‌ی نمودار بلوکی در برنامه‌ی Binary Read/Write.vi**

به نحوه‌ی تبدیل نوع داده به SGL در شکل ۲۴-۱۴ دقت کنید. همان گونه که قبلاً عنوان شد در مورد فایل های باینری سیم کشی نوع داده‌ی مناسب از اهمیت بسیار بالایی برخوردار است. در غیر این صورت استخراج داده ها از این فایل ها امکان ناپذیر است.

## تمرین ۲-۱۴: مقایسه‌ی عملکرد فایل های باینری، متنی و datalog

با استفاده از برنامه های مخصوص DAQ، سه برنامه بنویسید که عملیات خواندن و نوشتن را به ترتیب بر روی فایل های باینری، متنی و datalog انجام دهند. سپس با کمک برنامه‌ی File/Directory Info.vi که در زیرپالت Functions >> File I/O >> Advanced File Functions قرار دارد اندازه‌ی هر یک از فایل های تولید شده را تعیین کنید.




**شکل ۲۵-۱۴: آیکن دستور File/Directory Info**

نتیجه‌ی این مقایسه بسیار جالب است. در مرحله‌ی بعد روشی پیشنهاد کنید که به کمک آن بتوان زمان لازم جهت خواندن یا نوشتن فایل را اندازه گیری نمود. این برنامه را با عنوان File Type Comparison.vi در زیرفهرست LabVIEW\My Activity ذخیره کنید.

## برقراری ارتباط با برنامه های دیگر

این مبحث را با طرح دو سؤال زیر آغاز می کنیم:

آیا انتقال داده های یک برنامه از محیط LabVIEW مستقیماً به داخل یک برنامه‌ی صفحه گسترده نظیر Microsoft Excel امکان پذیر است؟

آیا می توان برنامه‌ای نوشت که از آن به عنوان یک رابط گرافیکی سطح بالا برای اجرای برنامه های کاربردی دیگر استفاده نمود؟

خبر خوشی برای شما داریم و آن این است که در محیط برنامه‌نویسی LabVIEW انجام این اعمال و حتی اعمال پیچیده‌تر امکان‌پذیر است. اما خبر ناخوشایند آن است که نوشتن این برنامه ها برای برقراری ارتباط داخلی، مستلزم مطالعه‌ی جزییات در مورد فرمت مناسب داده ها و پروتکل های مناسب برای تبادل موفقیت آمیز داده ها بین محیط LabVIEW و برنامه های کاربردی دیگر است.

این بخش به بررسی روش برقراری ارتباط بین محیط LabVIEW و برنامه های کاربردی اختصاص داده شده است. در بخش بعد نیز به بررسی نحوه‌ی برقراری ارتباط شبکه‌ای بین چند رایانه می پردازیم. بین این دو مبحث ارتباط تنگاتنگی وجود دارد. بنابراین هنگامی که در این بخش، از برقراری ارتباط با برنامه های دیگر صحبت می شود الزاماً این برنامه ها بر روی یک رایانه وجود ندارند، بلکه ممکن است این برنامه ها بر روی یک شبکه‌ی رایانه‌ای قرار داشته باشند. در این بخش فرض بر این است که سیستم عامل رایانه‌ی مورد بحث به راحتی قادر است از طریق شبکه با محیط LabVIEW ارتباط برقرار سازد. به عبارت دیگر LabVIEW پروتکل استفاده شده در شبکه‌ی رایانه‌ای را پشتیبانی می کند. قبل از ورود به بحث به بررسی پروتکل ها می پردازیم.

## نکاتی در مورد پروتکل ها

پروتکل یک زبان عمومی است که در ارتباطات رایانه‌ای مورد استفاده قرار می گیرد. یک پروتکل ارتباطی، روشی تعریف شده است که به شما اجازه می دهد تا مشخص کنید که چه داده‌ای را به کجا ارسال کنید بدون اینکه خود را در مورد نحوه‌ی رسیدن داده ها به مقصد درگیر سازید. برای برقراری ارتباط، چندین پروتکل به صورت استاندارد جهانی پذیرفته شده‌اند. به دلیل اینکه تغییر و اصلاح یک برنامه برای پذیرش یک پروتکل جدید چندان ساده نیست در انتخاب پروتکل مناسب دقت داشته باشید.

۱۴

در جدول ۲-۱۴چند پروتکل معتبر و قابل قبول در محیط LabVIEW را ملاحظه می کنید.

**جدول ۲-۱۴: پروتکل های سازگار با محیط LabVIEW**

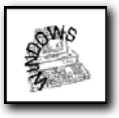
| Protocol | Win 95/NT | Win 3.1 | Mac | Sun & HP |
|---|---|---|---|---|
| OLE (Object Linking and Embedding) | ✓ | | | |
| DDE (Dynamic Data Exchange) | ✓ | ✓ | | |
| AppleEvents | | | ✓ | |
| PPC (Program-to-Program Connection) | | | ✓ | |
| TCP/IP (Transmission Control Protocol/ Internet Protocol) | ✓ | ✓ | ✓ | ✓ |
| UDP (User Datagram Protocol) | ✓ | ✓ | ✓ | ✓ |

## مدل Client/Server

مدل ها ما را در بررسی نحوه‌ی برقراری ارتباط با برنامه های کاربردی یاری می دهند. عمومی ترین این مدل ها مدل Client/Server است. در این مدل یک برنامه‌ی کاربردی یا یک فرآیند که در اصطلاح به آن مصرف کننده یا Client گویند از برنامه‌ی کاربردی یا فرآیند دیگر که به آن سرویس دهنده یا Server گفته می شود سرویس دهی را درخواست می کند. به عنوان مثال اگر برنامه‌ای بنویسید که داده ها را به صورت بلادرنگ جمع آوری نماید و آنها را مستقیماً در نرم افزار Microsoft Excel ذخیره کند، برنامه‌ی مورد بحث و برنامه‌ی صفحه گسترده به ترتیب به عنوان مصرف کننده و سرویس دهنده عمل می کنند. زیرا برنامه‌ای که شما در محیط LabVIEW نوشته‌اید از برنامه‌ی Excel درخواست سرویس نموده است و این سرویس در حقیقت عملیات ذخیره‌ی داده هاست.

در برنامه های کاربردی که بر روی شبکه اجرا می شوند معمولاً برنامه‌ی سرویس دهنده بر روی یک رایانه و برنامه های مصرف کننده بر روی رایانه های دیگر که در فواصل دور از رایانه‌ی سرویس دهنده قرارگرفته اند به اجرا در می آیند. به دلیل اینکه برنامه های نوشته شده در محیط LabVIEW از برنامه های کاربردی دیگر درخواست سرویس می کنند در اکثر موارد این برنامه ها در قالب برنامه های مصرف کننده نوشته می شوند. البته می توان ساختار یک VI را به گونه‌ای تعریف نمود که به عنوان سرویس دهنده مورد استفاده قرار گیرد که به احتمال زیاد تنها می تواند به برنامه های نوشته شده در محیط LabVIEW سرویس دهد.

## پروتکل OLE[5]



پروتکل OLE یک استاندارد جهت برقراری ارتباط بین برنامه های کاربردی در محیط Windows است. در این پروتکل دستیابی به داده ها از یک برنامه‌ی کاربردی که در اصطلاح به آن Object گفته می شود امکان‌پذیر است. عمومی ترین مثال در مورد OLE که ممکن است تاکنون با آن برخورد کرده باشید ارتباط یک فایل صفحه گسترده در Microsoft Excel به یک محیط متنی در نرم افزار Microsoft Word می‌باشد. در این مثال، فایل صفحه گسترده به عنوان Object می‌باشد که به طور معمول به نرم افزار Word تعلق ندارد اما در محیط Word قابل دستیابی است و می توان آن را تغییر داد.

پروتکل OLE یک مفهوم ساده نیست و فراتر از مبحث این کتاب می‌باشد. برنامه های پروتکل OLE در نسخه‌ی LabVIEW 5.0 در زیرپالت ActiveX << Communication << Functions قرار دارند.

**شکل ۲۶-۱۴: زیرپالت ActiveX و برنامه های پروتکل OLE**

۱۴

5- Object Linking and Embedding

برای درک عملکرد این پروتکل و مطالعه‌ی مثالهایی در مورد آن، برنامه‌های موجود در مسیر LabVIEW\Examples\Comm\ole-excel8.llb را بررسی کنید.

نکته‌ی آخر این که درحال حاضر برنامه‌های نوشته شده در محیط LabVIEW را می‌توان تنها به صورت مصرف کننده استفاده نمود. این مطلب بدان معنی است که برنامه های کاربردی دیگر را می توان به کمک LabVIEW کنترل نمود اما برنامه های دیگر قادر به کنترل LabVIEW نمی باشند.

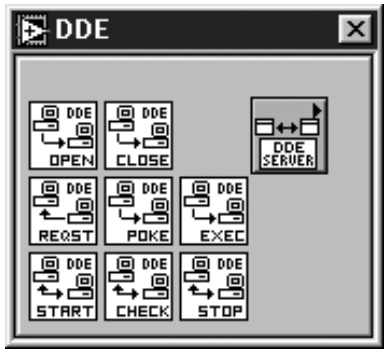
## پروتکل DDE[6]

استاندارد DDE یک پروتکل قدیمی برای تبادل اطلاعات بین برنامه های کاربردی در نرم افزار Windows است. می توانید از DDE برای برقراری ارتباط بین برنامه های کاربردی بر روی یک رایانه یا بر روی یک شبکه‌ی رایانه‌ای استفاده کنید. این پروتکل تنها توسط سیستم عامل Win 95/NT پشتیبانی می‌شود.

برای برقراری ارتباط با رایانه‌هایی که سیستم عامل آنها به غیر از Windows می‌باشد (نظیر Mac) استفاده از برنامه های خاص پروتکل DDE مجاز نیست. در صورت نیاز به برقراری ارتباط با سیستم‌های عامل دیگر از پروتکل‌های عمومی نظیر TCP/IP استفاده نمایید.

در پروتکل DDE، برنامه های کاربردی جهت تبادل اطلاعات، برای یکدیگر پیام ارسال می کنند. به عنوان مثال یک برنامه در محیط LabVIEW می‌تواند برای متصل شدن به محیط Excel یک «پیام اتصال[7]» ارسال کند. سپس فرمانهایی مبنی بر درخواست اطلاعات از فایل صفحه گسترده به محیط Excel ارسال می کند و در پایان با ارسال فرمان Close به این عملیات خاتمه می دهد.

با استفاده از برنامه های DDE می‌توانید برنامه‌ای بنویسید که عملکرد آن شبیه به یک مصرف کننده‌ی ساده باشد. برنامه های خاص DDE در زیرپالت Functions >> Communication >> DDE قرار دارند.



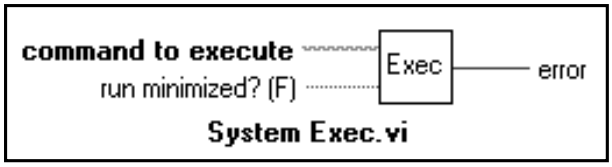

**شکل ۲۷-۱۴: زیرپالت DDE**

6- Dynamic Data Exchange
7- Connect Message

در لوح فشرده‌ی همراه این کتاب چند مثال جالب و مفید وجود دارد که داده‌های یک جدول را به نرم افزار Excel ارسال می‌کنند یا داده‌ها را از آن می‌خوانند. این مثالها در مسیر LabVIEW\Examples\Comm\ddeexamp.llb قرار دارند.

## ارسال فرمانهای سیستم به رایانه از طریق LabVIEW

فرض کنید که می‌خواهید با نوشتن برنامه‌ای در محیط LabVIEW، بدون کنترل نمودن برنامه‌های کاربردی دیگر، آنها را بسته یا باز کنید. در نرم افزار LabVIEW برنامه‌ای بسیار ساده برای اجرای فرمانهای سیستم تعبیه شده است. این برنامه با عنوان System Exec.vi در زیرپالت Functions >> Communication قرار دارد.




**شکل ۲۸-۱۴: آیکن برنامه‌ی System Exec.vi به همراه اتصالات آن**

این برنامه فرمان مورد نظر را به راحتی اجرا می‌کند. برای انجام عمل مورد نظر کافی است فرمان مورد نظر را همانند یک دستور در خط فرمان DOS تایپ کنید.

## پروتکل AppleEvents

استاندارد AppleEvents یک پروتکل خاص جهت برقراری ارتباط بین برنامه‌های کاربردی در محیط Mac است. این پروتکل برای باز کردن یک برنامه، درخواست اطلاعات، انجام عملیات‌چاپ و... به برنامه‌های کاربردی دیگر یا حتی به سیستم عامل پیامهایی را ارسال می‌کند. یک برنامه‌ی کاربردی می‌تواند به خود، به برنامه‌های کاربردی دیگر بر روی همان رایانه یا بر روی رایانه‌های دیگر پیامهایی ارسال نماید.

برای برقراری ارتباط با رایانه‌هایی که سیستم عامل آنها به غیر از Mac می‌باشد (نظیر Windows) استفاده از برنامه‌های خاص AppleEvents مجاز نیست. در صورت نیاز به برقراری ارتباط با سیستم‌های عامل دیگر از پروتکل‌های عمومی نظیر TCP/IP استفاده کنید.

برای درک پیامهای ارسال شده توسط پروتکل AppleEvents بهتر است که ابتدا با سیستم عامل Mac به طور کامل آشنا شوید. برنامه‌های خاص AppleEvents در زیرپالت Functions >> Communication >> AppleEvents قرار دارند.

به دلیل اینکه لوح فشرده‌ی همراه کتاب تنها حاوی نرم‌افزار LabVIEW در محیط Windows است، زیرپالت AppleEvents در این محیط وجود ندارد.

## پروتکل PPC[8]

استاندارد PPC که در اصطلاح به آن پروتکل «ارتباط برنامه با برنامه» گویند یک پروتکل ارتباط داخلی سطح پایین در محیط Mac است که به برنامه‌های کاربردی اجازه می‌دهد تا داده‌ها را به صورت بلوکی ارسال یا دریافت دارند.

برای برقراری ارتباط از طریق این پروتکل برنامه‌های خاصی در زیرپالت Functions >> Communications >> PPC تعبیه شده‌اند. پروتکل PPC بسیار پیشرفته‌تر و پیچیده‌تر از پروتکل AppleEvents است. در این کتاب از ذکر جزییات در مورد این پروتکل قدرتمند خودداری می‌کنیم.

به دلیل اینکه لوح فشرده‌ی همراه کتاب تنها حاوی نرم‌افزار LabVIEW در محیط Windows است، زیرپالت PPC در این محیط وجود ندارد.

## برقراری ارتباط با رایانه‌های دیگر از طریق پروتکل TCP/IP

حال خود را برای به‌کارگیری نرم‌افزار LabVIEW در یک شبکه‌ی رایانه‌ای آماده کنید. در این بسته‌ی نرم‌افزاری تعدادی برنامه‌ی از پیش نوشته شده برای ارتباط شبکه‌ای تعبیه شده است. در این ارتباطات از پروتکل TCP/IP استفاده می‌گردد. مزایای این پروتکل نسبت به پروتکل‌های دیگر عبارت است از:

- به کمک این پروتکل برقرار نمودن ارتباط بین چندین رایانه که از سیستم‌های عامل متفاوت (HP، Sun، Mac و Windows) استفاده می‌کنند امکان‌پذیر است.
- برقراری ارتباط همزمان با چند رایانه از طریق این پروتکل امکان‌پذیر است.
- رایانه‌هایی که به صورت شبکه با یکدیگر ارتباط دارند می‌توانند در فواصل بسیار دور نسبت به یکدیگر قرار داشته باشند.

به بیان دیگر به کمک این پروتکل می‌توانید فرآیند DAQ را بر روی یک رایانه‌ی شخصی و تحت سیستم عامل Windows در ایالت نیوجرسی انجام دهید، سپس نتایج حاصل را از طریق برنامه‌های تحلیلگر بر روی یک سیستم Mac در ایالت نیویورک ارسال نمایید و این در حالی است که یکی از دوستان شما که به عنوان مثال در ایران به سر می‌برد قادر است تمامی فرآیند را از ابتدا تا انتها به صورت بلادرنگ بر روی یک سیستم Sun مشاهده کند.

البته هیچ لزومی ندارد که رایانه‌های شبکه شده در فواصل دور نسبت به یکدیگر قرار گیرند. آنچه که در مورد این پروتکل ارتباطی نیاز دارید سخت‌افزار و نرم‌افزار مناسب جهت اتصال به یک شبکه‌ی رایانه‌ای است.

شایان ذکر است که این شبکه باید پروتکل TCP/IP را پشتیبانی کند. تمامی سیستم‌های عامل

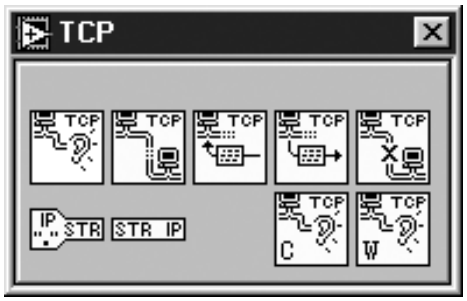


8- Program-to-Program Communication

Winsows 95/NT، Mac، Sun و HP-UX از این پروتکل پشتیبانی می کنند. در ادامه‌ی بحث فرض بر آن است که در مورد پروتکل TCP/IP اطلاعات کافی در اختیار دارید.

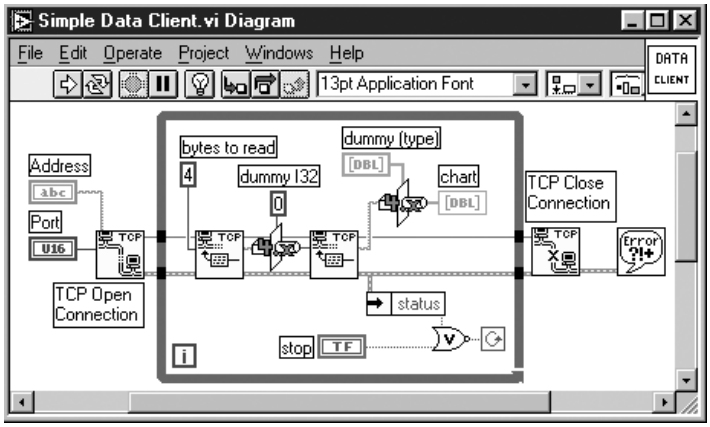
TCP و IP به ترتیب مخفف عبارات Transmission Control Protocol و Internet Protocol می‌باشند. IP داده های مورد نظر را به بسته هایی که در اصطلاح به آنها datagram گویند تقسیم نموده، سپس آنها را ارسال می کند. IP به صورت هوشمند عمل نمی کند، بدین معنی که هیچ گونه سیگنال handshaking را برای رایانه‌ی مقصد ارسال نمی کند. این محدودیت می تواند باعث بروز مشکلات عدیده‌ای گردد. عملکرد پروتکل IP را نظیر ارسال یک نامه از طریق پست معمولی در نظر بگیرید. در ارسال نامه به روش معمولی تضمینی مبنی بر رسیدن بسته به مقصد وجود ندارد. در مورد پروتکل IP نیز هیچ ضمانتی مبنی بر دریافت datagram توسط گره‌ی مقصد وجود ندارد. پس از آن، TCP ظهور نمود و به پروتکل IP افزوده شد. TCP عملیات handshaking را انجام می دهد و ارسال datagramها را به مقصد تضمین می کند. عملکرد پروتکل TCP را مانند ارسال نامه از طریق پست سفارشی در نظر بگیرید.

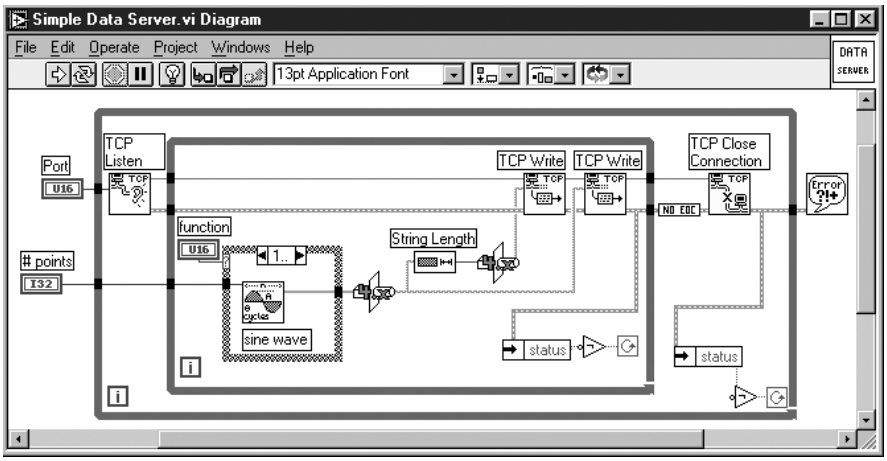
در بسته‌ی نرم‌افزاری LabVIEW، برنامه های خاصی جهت برقراری ارتباط از طریق این پروتکل در زیرپالت Functions >> Communication >> TCP تعبیه شده است.




**شکل ۲۹-۱۴: زیرپالت TCP**

به کمک این برنامه ها می توانید فرمانهای مرتبط با پروتکل TCP نظیر برقراری ارتباط از طریق یک IP، خواندن یا نوشتن داده ها و... را اجرا کنید. برای مطالعه‌ی مثالهای بیشتر به مسیر LabVIEW\Examples\Comm\tcpex.llb مراجعه کنید. در شکل ۳۰-۱۴ صفحه‌ی نمودار بلوکی دو برنامه‌ی Simple Data Client.vi و Simple Data Server.vi را ملاحظه می کنید.

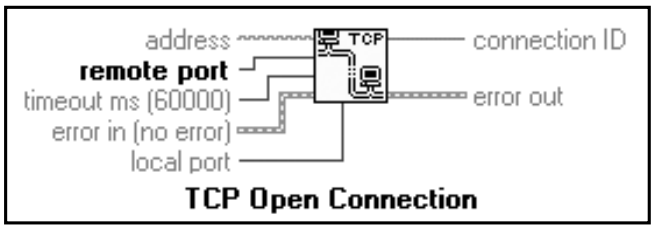





**شکل ۳۰-۱۴: صفحه‌ی نمودار بلوکی دو برنامه‌ی Simple Data Client.vi و Simple Data Server.vi**

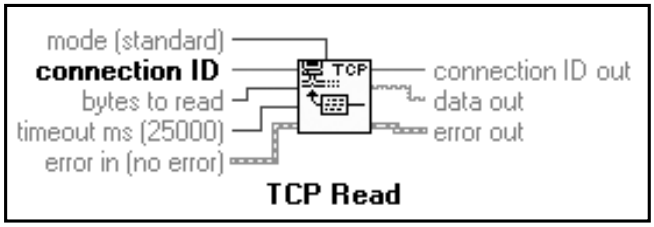
با دقت در این نمودارهای بلوکی می توانید مطالبی را جهت نوشتن برنامه های Client/Server فرا گیرید. مراحل فرآیند برقراری ارتباط در یک برنامه‌ی مصرف کننده به صورت زیر است:

۱- درخواست برقراری ارتباط TCP. در صورت عدم ارسال پاسخ توسط برنامه‌ی سرویس دهنده، برای جلوگیری از توقف[9] برنامه‌ی خود می توانید خطای timeout را در برنامه قرار دهید.



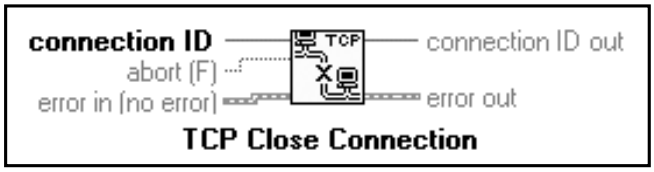

**شکل ۳۱-۱۴**

۲- خواندن داده ها. به یاد داشته باشید که داده ها همواره به صورت رشته ای انتقال می یابند.




**شکل ۳۲-۱۴**

۳- قطع ارتباط TCP.



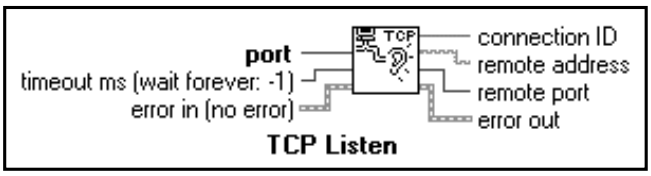

**شکل ۳۳-۱۴**

۱۴

9- Hang

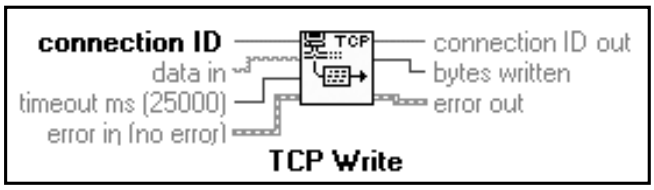
مراحل فرآیند برقراری ارتباط در یک برنامه‌ی سرویس دهنده به صورت زیر است:

۱- انتظار برای دریافت یک درخواست ارتباطی TCP.



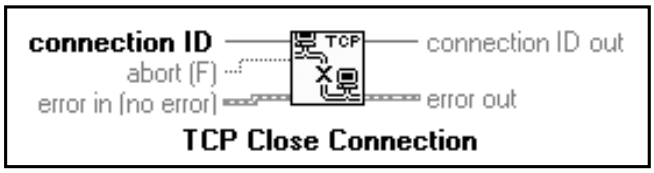

شکل ۱۴-۳۴

۲- نوشتن داده ها. به یاد داشته باشید که داده ها همواره به صورت رشته ای انتقال می یابند.




شکل ۱۴-۳۵

۳- قطع ارتباط TCP.




شکل ۱۴-۳۶

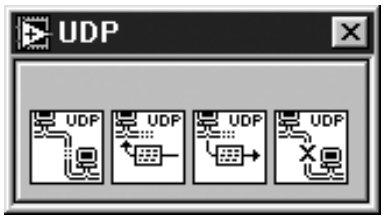
به دلیل اینکه در پروتکل TCP/IP تمامی داده ها به صورت رشته ای انتقال می یابند، حتماً باید داده ها را به نوع رشته ای تبدیل کنید. همان گونه که در مثال قبل ملاحظه شد ساده ترین روش برای انجام این عمل استفاده از دستور Type Cast می باشد.

استفاده از برنامه های کاربردی به همراه شبکه در مواردی که قصد دارید تا برنامه ای را برای کنترل یک سیستم گسترده و عظیم بنویسید بسیار ایده‌آل است. برنامه های کاربردی جهت کنترل فرآیند از این نوع برنامه ها هستند.

پروتکل دیگری با نام UDP[10] وجود دارد که LabVIEW آن را پشتیبانی می کند و هنوز از آن صحبتی به میان نیاورده ایم. عملکرد این پروتکل تقریباً شبیه به پروتکل IP است، بدین مفهوم که عملیات handshaking را انجام نمی دهد اما می تواند داده ها را به چند رایانه ارسال نماید و تا حدودی نسبت به پروتکل IP سریعتر عمل می کند.

10- User Datagram Protocol

به دلیل اینکه کاربرد و تعداد کاربران این پروتکل اندک است از ذکر جزییات در مورد این پروتکل خودداری می کنیم. برنامه های خاص پروتکل UDP در زیرپالت Functions >> Communication >> UDP قرار دارند.




شکل ۳۷-۱۴: زیرپالت UDP

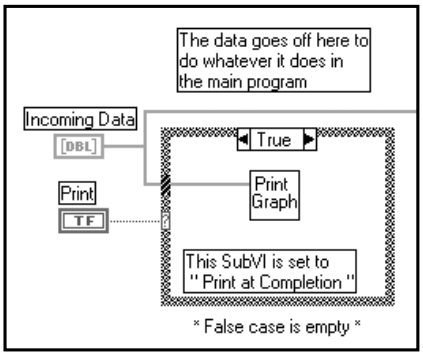
## روش چاپ برنامه در محیط LabVIEW

همان گونه که در فصل ۵ عنوان شد برای چاپ پنجره‌ی فعال در LabVIEW می توان از گزینه‌ی File >> Print Window... استفاده نمود. چاپ برنامه پس از اتمام اجرای آن نیز امکان پذیر است. برای انجام این عمل کافی است گزینه‌ی Operate >> Print at Completion را انتخاب کنید.

حال اگر قصد داشته باشید بخشی از صفحه‌ی پانل را چاپ کنید یا اینکه صفحه‌ی برنامه را در زمانی خاص و نه در صورت پایان اجرای آن به چاپ برسانید، چه باید کرد؟

برای انجام این عمل نیز می توانید از گزینه‌ی Operate >> Print at Completion استفاده کنید. برای درک عملکرد این دستور یک زیربرنامه ایجاد کنید و در صفحه‌ی پانل آن، گراف، المان های جبری و یا هر موردی که قصد چاپ آن را دارید قرار دهید. سپس گزینه‌ی Operate >> Print at Completion را در مورد این زیربرنامه انتخاب نمایید. در اینجا نیازی نیست تا هیچ المانی را بر روی صفحه‌ی نمودار بلوکی قرار دهید. سپس زیربرنامه‌ی مذکور را در برنامه‌ی اصلی خود قرار دهید و آن را به داده‌ای که قصد چاپ آن را دارید سیم کشی کنید. به دلیل اینکه هیچ دستوری در صفحه‌ی نمودار بلوکی زیربرنامه قرار ندارد، اجرای آن فوراً به پایان می رسد و در حالی که اجرای برنامه‌ی اصلی ادامه می یابد، صفحه‌ی پانل زیربرنامه جهت انجام عملیات چاپ به چاپگر ارسال می شود.

در شکل ۳۸-۱۴ نمونه ای از این روش چاپ را ملاحظه می کنید. در این زیربرنامه کاربر قادر است که با فشار دادن دکمه‌ی Print، گراف ترسیم شده را در زمان دلخواه چاپ کند.




شکل ۳۸-۱۴

۱۴

در این حالت بر روی صفحه‌ی پانل در برنامه‌ی Print graph.vi، تنها یک گراف وجود دارد. ساختار این برنامه به کمک گزینه‌ی Print at Completion تعریف شده است. به خاطر داشته باشید برای سیم کشی داده ها به گراف، ابتدا باید با استفاده از گزینه‌ی Change to Control در منوی کرکره‌ای گراف، آن را به یک المان کنترل تبدیل کنید.

## خلاصه

در این فصل برخی از عناوین پیشرفته‌ی I/O را مورد بحث و بررسی قرار دادیم. این مباحث عبارتند از: ورودی/خروجی فایل، ارتباط برنامه های کاربردی با یکدیگر، ارتباط شبکه‌ای از طریق پروتکل ها و روش چاپ برنامه.

در محیط LabVIEW می توان از سه نوع فایل متنی، باینری و datalog استفاده نمود. به کارگیری فایل های متنی ساده تر از سایر فایل ها می‌باشد اما حجم اشغال شده توسط فایل و همچنین زمان لازم جهت ذخیره‌ی آن بیشتر است. استفاده از فایل های datalog برای ثبت داده های صفحه‌ی پانل ایده آل است. به کارگیری فایل های باینری مشکل تر از سایر فایل هاست اما کمترین سربار را در مورد سرعت و فضای اشغال شده به همراه دارد. در مورد هر سه نوع فایل می توان از دستورهای اصلی موجود در زیرپالت Functions >> File I/O نظیر Read File، New File، Open File و Close استفاده نمود.

قابلیت برقراری ارتباط نرم‌افزار LabVIEW با برنامه های کاربردی دیگر، قدرت برنامه های نوشته شده در این محیط را افزایش می دهد. دستورهای تعبیه شده در این بسته‌ی نرم‌افزاری به کاربر اجازه می دهند تا برای برقراری ارتباط با برنامه های کاربردی نظیر برنامه های صفحه گسترده یا پردازشگرهای متنی، در محیط Windows از پروتکل DDE و در محیط Mac از پروتکل های PPC یا AppleEvents استفاده کند.

به کمک دستورهای مربوط به پروتکل TCP/IP از طریق شبکه‌ی اینترنت یا هر شبکه‌ی دیگری که توسط پروتکل پشتیبانی شود، دسترسی به رایانه هایی که در فواصل دور قرار گرفته‌اند امکان‌پذیر است. برنامه‌ی نوشته شده در محیط LabVIEW را می توان تحت پروتکل TCP/IP به صورت مصرف کننده یا سرویس‌دهنده استفاده نمود.

در این فصل ملاحظه نمودید که چگونه می توان با استفاده از یک زیربرنامه‌ی ساختگی و گزینه‌ی Operate >> Print at Completlion بخشهای دلخواه برنامه را چاپ کرد.

۱۴

LabVIEW

# ۱۵

# ارتقای کیفی محیط برنامه‌نویسی LabVIEW

## در این فصل با موارد زیر آشنا می‌شوید:

- روشهای پیشنهادی جهت ایجاد یک صفحه‌ی گرافیکی بسیار جالب و کاملاً حرفه‌ای
- واردکردن تصاویر از محیطهای گرافیکی دیگر به صفحه‌ی پانل
- سفارشی کردن المان‌های کنترل با استفاده از محیط Control Editor
- روش ایجاد پنجره‌ی Help در مورد برخی از المان‌های موجود در صفحه

# LabVIEW

# ۱۵ ارتقای کیفی محیط برنامه‌نویسی LabVIEW

در این فصل به بررسی تکنیک‌ها و روشهای پیشنهادی جهت هرچه بهتر کردن محیط برنامه‌نویسی LabVIEW می‌پردازیم. در ابتدا شکل ظاهری صفحه‌ی پانل را مورد بحث قرار می‌دهیم و تا حد امکان سعی در زیباسازی و سازماندهی المان‌های قرار گرفته در آن خواهیم داشت. سپس مفهوم کلی رابط گرافیکی کاربر[1] را در بسته‌ی نرم‌افزاری LabVIEW مورد بررسی قرار می‌دهیم. در این فصل سعی خواهیم داشت تا روشهای مفید و کاربردی قابل استفاده در LabVIEW را پیش پای شما قرار دهیم تا به کمک آنها بتوانید المان‌ها و صفحه‌ی پانل را مرتب نمایید، به زیباسازی فضای برنامه‌نویسی بپردازید و عملکرد المان‌های کنترل را به دلخواه خود در آورید. همچنین در این فصل خواهید دید که چگونه می‌توان در مورد هر یک از اجزای موجود در صفحه، یک پنجره‌ی حاوی اطلاعات Online Help اضافه نمود.

در این فصل اهداف زیر دنبال می‌شوند:

- آشنایی با روشهای کلی و راههای پیشنهادی جهت ایجاد یک صفحه‌ی گرافیکی بسیار جالب و کاملاً حرفه‌ای.
- واردکردن تصاویر از محیطهای گرافیکی دیگر (نظیر Paint) به صفحه‌ی پانل.
- سفارشی کردن المان‌های کنترل با استفاده از محیط Control Editor.
- ایجاد پنجره‌ی Help در مورد برخی از المان‌های موجود در صفحه.

اصطلاحات کلیدی موجود در این فصل عبارتند از:

- زیباسازی (Decorations)
- المان‌های کنترل و نشان‌دهنده‌ی سفارشی (Custom Controls and Indicators)
- پنجره‌ی Control Editor

1- Graphical User Interface

## نمای ظاهری یک برنامه‌ی گرافیکی

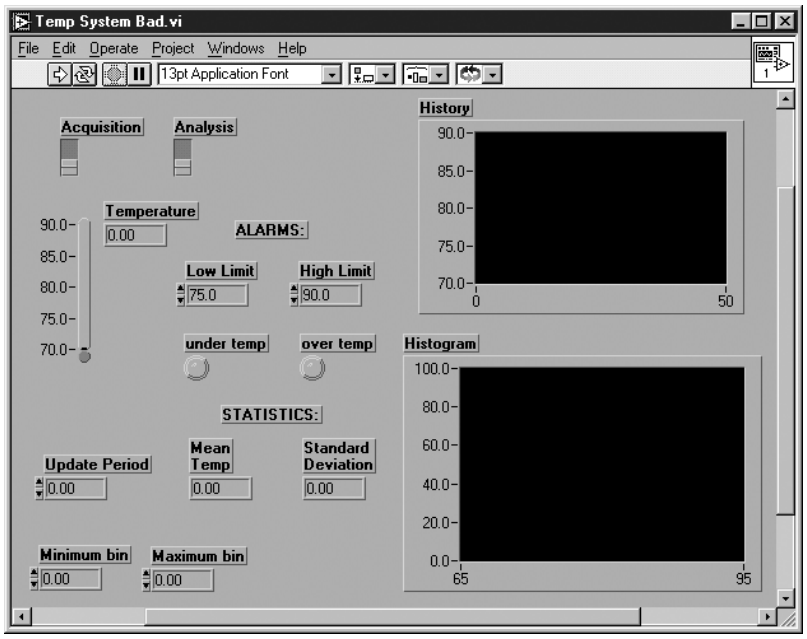
در فرهنگ ما شکل ظاهری اشیا گویای موارد بسیاری است. اکثر مردم تحت تأثیر نمای ظاهری صفحه‌ی برنامه قرار می‌گیرند و در برخی شرایط حتی به عملکرد آن توجه نمی‌کنند. در اینجا به ذکر دو دلیل برای زیباسازی صفحه‌ی برنامه می‌پردازیم:

۱- تأثیر گذاشتن و رساندن مفهوم و مقصود کلی و همچنین انتقال کیفیت و قدرت برنامه‌نویسی برنامه‌ی شما به اشخاص دیگر.

۲- ساده‌تر کردن نمای ظاهری برنامه جهت استفاده‌ی کاربران دیگر.

علاوه بر موارد مذکور، نمای صفحه‌ی برنامه زیباتر می‌شود و کارکردن با این صفحه به مراتب خوشایندتر است. المان‌های موجود در صفحه‌ی پانل در محیط برنامه‌نویسی LabVIEW نظیر دکمه‌های چرخشی، LEDها، دکمه‌های لغزشی و... به اندازه‌ی کافی زیبا هستند. اما این فصل برخی نکات و روشهای کاربردی جهت سازماندهی المان‌ها را به شما آموزش می‌دهد، به گونه‌ای که با قرار دادن و مرتب کردن آنها بتوانید صفحه‌ای زیباتر ایجاد کنید. همچنین خواهید دید که چگونه می‌توان برای راهنمایی کردن کاربران دیگر در مورد عملکرد هر یک از المان‌ها، یک پنجره‌ی Online Help طراحی نمود. مزیت به کارگیری این پنجره آن است که کاربران دیگر بدون نیاز به راهنمایی می‌توانند درمورد عملکرد برنامه، زیربرنامه‌ها و المان‌های موجود در صفحه‌ی پانل اطلاعاتی کسب نمایند.

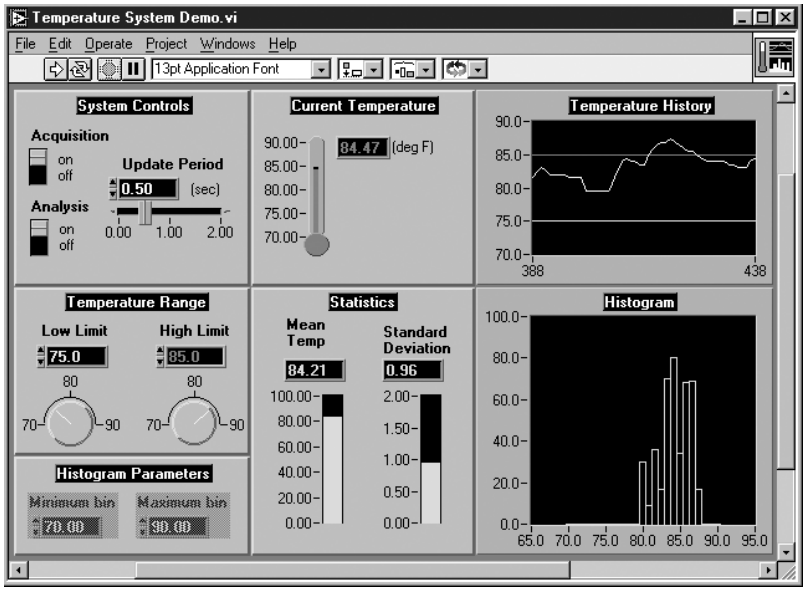
بنابراین نتیجه می‌گیریم که مزایای طراحی یک صفحه‌ی ارتباط دهنده‌ی مرتب و زیبا تنها محدود به موارد ذکرشده نیست. بلکه باعث صرفه‌جویی در زمان صرف شده جهت کسب اطلاعات برای کاربران دیگر نیز خواهد شد. حال به برنامه‌ی زیر توجه کنید. این صفحه مربوط به برنامه‌ی نظارت و کنترل بر یک سیستم گرمایی است. همان گونه که ملاحظه می‌کنید المان‌های موجود در شکل ۱-۱۵، بدون نظم و ترتیب خاصی در صفحه‌ی پانل قرار گرفته‌اند.




شکل ۱-۱۵

۱۵

اما تقریباً همه با این نظر موافق‌اند که صفحه‌ی پانل نشان داده شده در شکل ۲-۱۵ که دقیقاً عملکرد صفحه‌ی قبل را دارد، هم از لحاظ ظاهری زیباتر بوده و هم به دلیل نظم و ترتیب حاکم بر المان های موجود در صفحه کارکردن با آن ساده تر است.

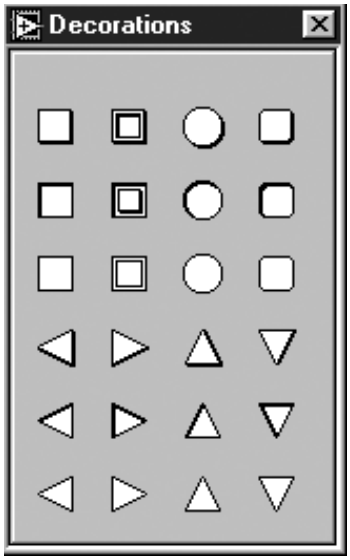



شکل ۲-۱۵

## مرتب کردن المان‌ها و زیباسازی صفحه‌ی پانل

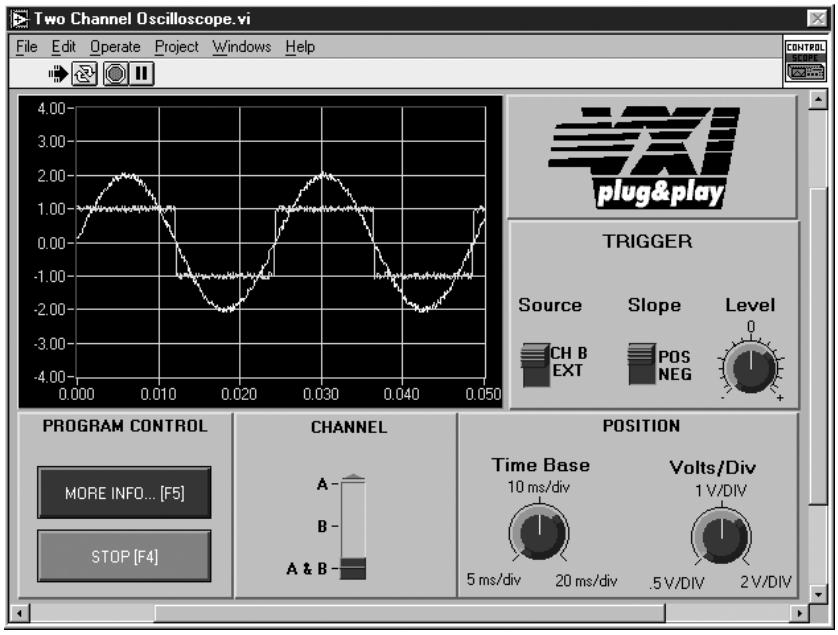
یکی از روشهای بهبود و ارتقای کیفی نمای ظاهری صفحه‌ی پانل، سازماندهی و مرتب کردن المان هاست. برای انجام این عمل ابتدا المان ها را نسبت به نوع عملکرد و اندازه‌ی آنها سازماندهی کنید و در مکان مناسب قرار دهید. این عمل را می توانید با توزیع کردن المان‌ها و قرار دادن آنها به صورت متقارن در صفحه یا با دسته بندی المان هایی که از نظر منطقی به هم وابسته اند انجام دهید.

همان گونه که در فصل ۴ عنوان شد در محیط برنامه نویسی LabVIEW می توانید المان های موجود در صفحه را در یک ردیف تراز نمایید و آنها را در صفحه توزیع کنید. برای تراز کردن المان ها حالتهای انتخابی بسیاری وجود دارد. به عنوان مثال می توانید المان ها را در ردیفهای افقی یا عمودی قرار دهید. نحوه‌ی توزیع المان ها نیز بسته به فضای در نظر گرفته شده بین آنها متفاوت است. به خاطر داشته باشید که می توان برچسب هر المان را در هر نقطه از صفحه قرار داد. اما برچسب مذکور وابسته و متعلق به المان است و می توان از آن به عنوان یک نقطه‌ی مرجع جهت مرتب سازی و توزیع المان ها استفاده نمود؛ بدین مفهوم که به عنوان مثال برای توزیع یا تراز کردن المان ها، می توان برچسب آنها را مرتب نمود. زیرپالت Decorations << Controls حاوی المان هایی است که برای زیباسازی صفحه به کار برده می شوند.



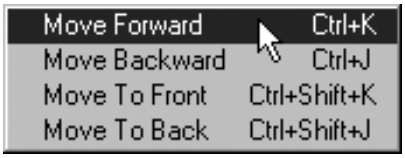

**شکل ۳-۱۵: زیرپالت Decorations**

المان های موجود در زیرپالت مذکور در صفحه‌ی پانل قرار می گیرند. اما از لحاظ عملیاتی هیچ عملی انجام نمی‌دهند و فاقد ترمینال متناظر در صفحه‌ی نمودار بلوکی می باشند. می توانید این المان ها را جهت دسته بندی المان های کنترل و نشان دهنده به کار برید. با استفاده از این المان ها می توانید نمای ظاهری صفحه‌ی پانل را به صورت نمای جلویی دستگاههای اندازه گیری در آورید. به عنوان مثال در شکل ۴-۱۵ ملاحظه می کنید که به کمک ابزارها و المان های موجود در این زیرپالت، پانل یک دستگاه اسیلوسکوپ شبیه سازی شده است.




**شکل ۴-۱۵: شبیه سازی پانل یک دستگاه اسلوسکوپ**

به دلیل اینکه المان‌های موجود در صفحه‌ی پانل معمولاً بر روی این ابزارها قرار می‌گیرند، برای قرار دادن ابزارهای زیباسازی در پشت المان‌های موجود در صفحه‌ی پانل از گزینه‌های موجود در منوی حلقوی Reorder (نظیر Move to Back) استفاده کنید. دلیل استفاده از این گزینه‌ها نیز کاملاً روشن است. زیرا شما یک المان تزییناتی را برای دسته‌بندی مجموعه‌ای از المان‌های کنترل یا نشان‌دهنده استفاده می‌کنید. این ابزارهای تزییناتی به صورت معمول فاقد خاصیت فرانمایی[2] هستند و در صورتی که پس از ایجاد صفحه‌ی پانل، ابزارهای مذکور را استفاده نمودید واضح است که تمامی المان‌های موجود در صفحه‌ی پانل را می‌پوشانند و بر روی آنها قرار می‌گیرند. در شکل ۵-۱۵ گزینه‌های موجود در این منوی حلقوی را ملاحظه می‌کنید.

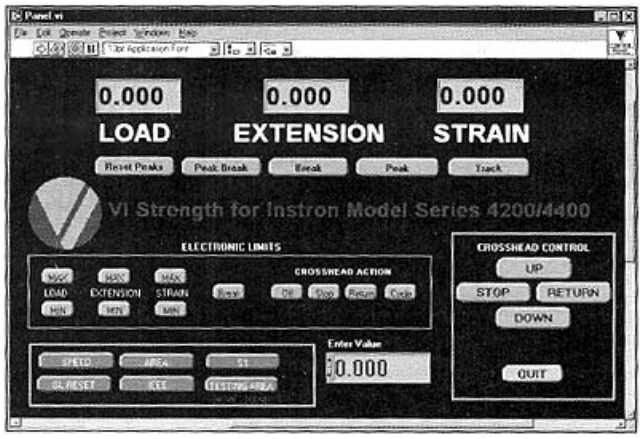



**شکل ۵-۱۵: گزینه‌های موجود در منوی حلقوی Reorder**

کلیک کردن بر روی المان‌های تزییناتی نظیر کلیک کردن بر روی بخشهای فاقد المان در صفحه‌ی پانل نیست. در هنگام استفاده از این ابزارها ممکن است با کلیک کردن بر روی این المان‌ها به طور ناخواسته آنها را انتخاب کنید.

## وارد کردن تصاویر به محیط LabVIEW

در نرم‌افزار LabVIEW می‌توانید تصاویری را از خارج از محیط برنامه‌نویسی به LabVIEW وارد کنید و آنها را در صفحه‌ی پانل قرار دهید. به عنوان مثال ممکن است قصد داشته باشید نام یک شرکت را بر روی صفحه‌ی پانل ایجاد کنید یا اینکه ممکن است بخواهید برای نشان دادن مراحل کنترل یک فرآیند، لوله‌ها یا شیرها را بر روی بخشی از صفحه به نمایش در آورید. تصویر شیرها را می‌توانید شبیه به المان‌های کنترل جبری فرض کنید. این مطلب را در بخش بعد مورد بررسی قرار می‌دهیم.



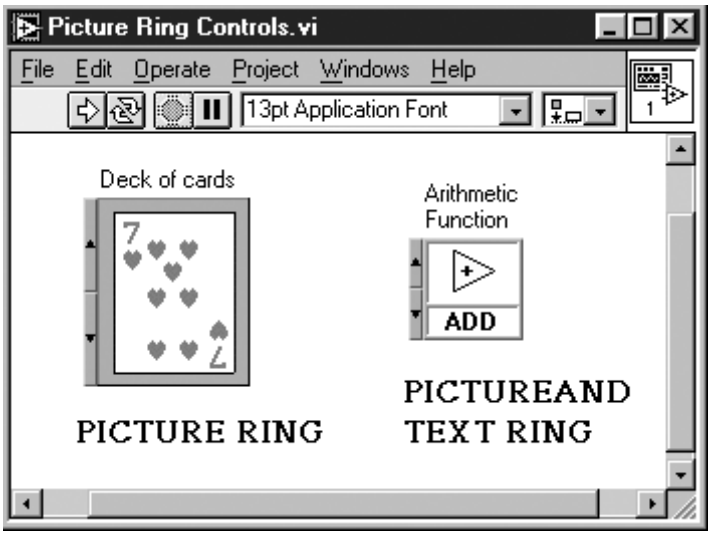

**شکل ۶-۱۵: در این برنامه آرم شرکت VI Technology به عنوان یک تصویر درصفحه‌ی پانل قرار گرفته است.**

2- Transparency

برای وارد کردن یک تصویر به محیط برنامه نویسی LabVIEW در دو محیط Windows و Mac، ابتدا تصویر مورد نظر را از محیط پردازشگر تصویر (به عنوان مثال محیط Paint) انتخاب کنید. سپس گزینه‌ی Edit >> Copy را انتخاب و در صفحه‌ی پانل بر روی گزینه‌ی Edit >> Paste کلیک کنید. در این حالت LabVIEW تصویر وارد شده را شبیه به یک ابزار تزییناتی فرض می کند. بنابراین می توانید اندازه‌ی آن را تغییر دهید یا آن را در پشت یا بر روی المان های دیگر قرار دهید. اما تغییر و اصلاح این تصویر در محیط LabVIEW امکان پذیر نیست.

در سیستم عامل Mac، تصویر وارد شده به فرمت PICT تبدیل می گردد. این فرمت از لحاظ درجه‌ی وضوح[3] در سطح بالایی است. در سیستم عامل Windows تصویر وارد شده به فرمت bmp در می آید. اشکال استفاده از تصویر با فرمت bmp در محیطهای گرافیکی دیگر آن است که به هنگام تغییر اندازه، درجه‌ی وضوح آن نیز تغییر می کند. در ضمن، LabVIEW تصاویر metafile را که فرمت گرافیکی جدیدتری در Windows می باشند پشتیبانی می کند. اندازه‌ی یک تصویر metafile را می توان بدون ایجاد تغییر در میزان وضوح آن افزایش داد. این تصاویر دارای قابلیت فرانمایی نیز می باشند. بنابراین در مورد هماهنگ کردن و تطبیق دادن رنگ پس زمینه‌ی صفحه‌ی پانل جای هیچ نگرانی نیست. در صورت وارد کردن یک تصویر metafile در LabVIEW، این تصویر بدون تبدیل به فرمت bmp در محیط LabVIEW پذیرفته می شود.

محل دیگری که می توان تصاویر را در آن قرار داد منوهای حلقوی Picture Ring و Picture and Text Ring می باشد. این دو المان در زیرپالت Controls >> List & Ring قرار دارند. در شکل ۷-۱۵ دو نمونه از کاربرد این حلقه ها را ملاحظه می کنید. به کارگیری Picture Ring به شما اجازه می دهد تا از بین تصاویر ایجاد شده در این منوی حلقوی گزینه‌ی دلخواه را انتخاب نمایید.



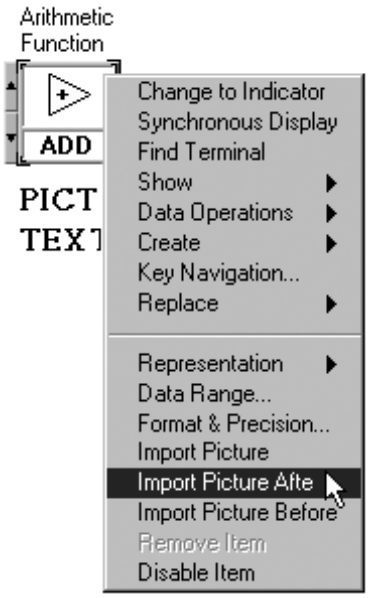

شکل ۷-۱۵

جهت اضافه کردن یک تصویر به Picture Ring ابتدا تصویر مورد نظر را در داخل کلیپ بورد[4]، کپی کنید. سپس منوی کرکره‌ای را بر روی نوار حلقوی باز کنید و گزینه‌ی Import Picture را انتخاب نمایید.

۱۵

3- Resolution
4- Clipboard

برای اضافه کردن تصاویر بیشتر و مرتب کردن آنها مطابق شکل۸-۱۵ یکی از دو گزینه‌ی Import Picture After یا Import Picture Before را انتخاب کنید. این دو گزینه در مراحل بعدی به ترتیب به گزینه های Add Item After و Add Item Before تغییر می یابند.

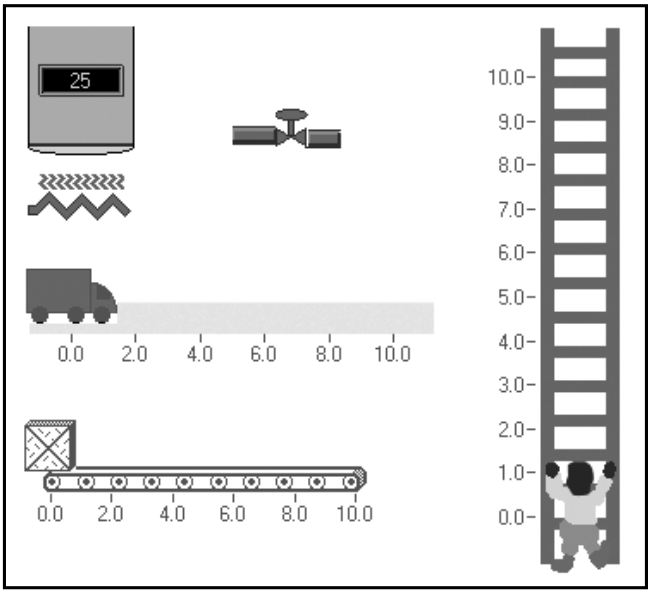



شکل ۸-۱۵

## المان‌های کنترل و نشان‌دهنده‌ی سفارشی[5]

این مبحث را با طرح یک سؤال آغاز می کنیم:

سؤال: وجه مشترک المان های زیر در چیست؟




شکل ۹-۱۵

۱۵

5- Custom Controls and Indicators

پاسخ: همگی، المان های کنترل در محیط LabVIEW می باشند.

برای تأثیرگذاری و داشتن صفحه‌ی بهتر به هنگام نمایش دادن صفحه‌ی پانل می توانید المان های کنترل یا نشان دهنده را به صورت دلخواه در آورید. به عنوان مثال ممکن است قصد داشته باشید تا به منظور نشان دادن مراحل تولید، موقعیت جعبه ها را بر روی نوار نقاله به عنوان جایگاه مواد تولید شده بر روی خط به نمایش در آورید، یا اینکه ممکن است بخواهید با کلیک کردن بر روی بخشی از تصویر کوره که به صورت شعله های آتش نشان داده شده است، آن را روشن یا خاموش کنید. می توانید یک المان کنترل یا نشان دهنده‌ی سفارشی را که از این به بعد به اختصار آن را المان کنترل یا نشان دهنده می خوانیم، در یک فهرست یا کتابخانه ای از برنامه ها (نظیر همان عملی که در مورد برنامه های معمولی و متغیرهای سراسری انجام می دادید) ذخیره کنید. پسوند قراردادی استاندارد برای نامگذاری یک المان کنترل سفارشی به صورت ctl. است. استفاده نمودن از المان کنترل ایجاد شده در یک برنامه، در برنامه های دیگر نیز امکان پذیر است. اگر نیاز به استفاده از المان ایجاد شده درچندین برنامه‌ی دیگر وجود دارد، می توانید یک نسخه‌ی اصلی از آن ایجاد نمایید و آن را به صورت type definition ذخیره کنید. در صورت ایجاد تغییر در نسخه‌ی اصلی، در تمامی برنامه هایی که از آن المان استفاده نموده اند، تغییر مورد نظر به صورت خودکار ایجاد و به روز رسانده می‌شود.

برای قرار دادن المان کنترل سفارشی در پالت Controls گزینه‌ی ...Edit >> Edit Control & Function Palettes را انتخاب نمایید و مراحل سفارشی نمودن پالت را دنبال کنید.

## تمرین ۱-۱۵: وارد کردن تصاویر در صفحه‌ی پانل

معمولاً برای نمایش المان های کنترل سفارشی، تصاویر مورد نظر را از صفحات و محیطهای دیگر به صفحه‌ی پانل وارد می کنید. بنابراین در ابتدای امر باید فایل حاوی تصویر را آماده کنید. ممکن است قصد داشته باشید آن را در یک محیط گرافیکی ایجاد نمایید. متأسفانه بسته‌ی نرم‌افزاری LabVIEW فاقد محیط پردازشگر تصویر است.

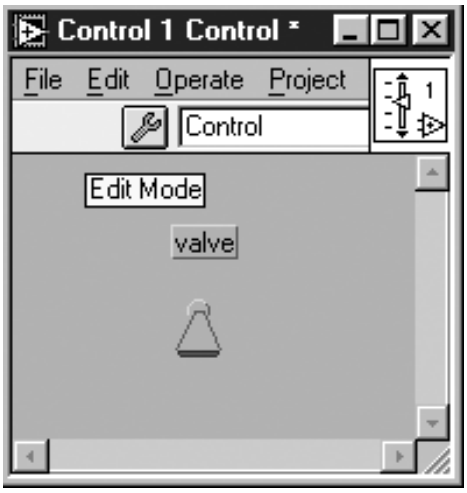
در این تمرین قصد داریم یک المان جبری دلخواه ایجاد کنیم که باز و بسته شدن یک شیر را با دو رنگ متفاوت نشان دهد.

۱- یک المان جبری (البته در این حالت یک کلید) را بر روی صفحه‌ی پانل قرار دهید و آن را با برچسب valve نام گذاری کنید. سپس این المان را با کلیک کردن توسط ابزار Positioning Tool انتخاب نمایید.

۲- با انتخاب گزینه‌ی ...Edit >> Edit Control محیط Control Editor را بارگذاری کنید.

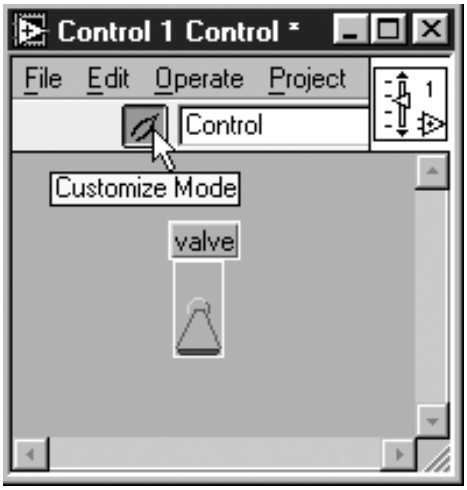
۳- المان جبری انتخاب شده در محیط Control Editor به نمایش در می آید.

۴- پنجره‌ی Control Editor در مد Edit دقیقاً شبیه به صفحه‌ی پانل عمل می کند. بنابراین می توانید برای تنظیم مقادیری از قبیل درجه بندی، دقت و... از منوی کرکره‌ای المان جبری انتخاب شده استفاده کنید.




شکل ۱۰-۱۵

۵- با کلیک کردن بر روی دکمه‌ی Tool به مد Customize دسترسی پیدا کنید. در مد مذکور می‌توانید پارامترهایی نظیر رنگ، اندازه و مکان اجزای مختلف المان کنترل را تغییر دهید.

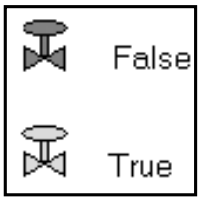



شکل ۱۱-۱۵

۶- تصویر یک «شیر بسته» را کپی نمایید یا از فایلی که در لوح فشرده وجود دارد استفاده کنید (در محیط Windows با نام Closed.bmp و در محیط Mac با نام Closed.pct). برای انجام این عمل منوی کرکره‌ای را در مد Customize بر روی المان جبری باز کنید و در حالتی که المان جبری در موقعیت False قرار گرفته است گزینه‌ی Import Picture را انتخاب نمایید. بدین مفهوم که این شکل، بیانگر تصویر شیر در حالت بسته است و معادل با موقعیت False در کلید جبری متناظر با آن می‌باشد.

شکل ۱۲-۱۵

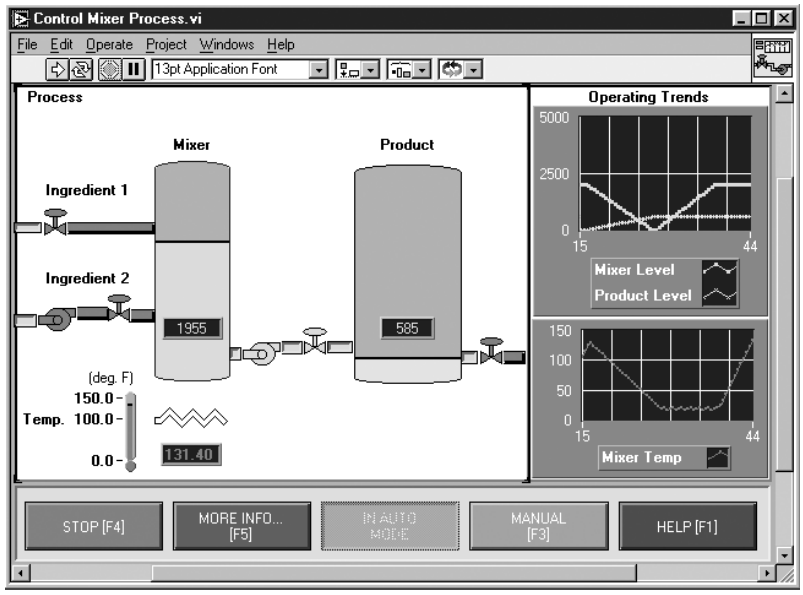
۷- با به کارگیری یک رنگ دیگر، مرحله ۶ را برای وضعیت True تکرار کنید. تصویر شیر باز یا متناظر با آن، وضعیت True در المان جبری نیز در لوح فشرده وجود دارد (در محیط Windows با نام Open.bmp و در محیط Mac با نام Open.pct). در شکل ۱۳-۱۵ شیر مورد نظر را در دو رنگ متفاوت که نشان دهنده‌ی دو موقعیت باز و بسته است ملاحظه می کنید.




شکل ۱۳-۱۵

۸- المان کنترل سفارشی را با استفاده‌ی از گزینه‌ی File >> Save ذخیره کنید. طبق قرارداد برای این المان ها پسوند ctl. در نظر بگیرید.

۹- صفحه‌ی پانل نشان داده شده در شکل ۱۴-۱۵ در مسیر LabVIEW\Examples\Apps\demos.llb\ControlMixerProcess.vi قرار دارد. در این صفحه المان کنترل سفارشی ایجاد شده را به همراه سایر اجزای برنامه ملاحظه می کنید. برنامه‌ی مذکور را باز نموده و المان های موجود در پنجره‌ی پانل آن را بررسی کنید.

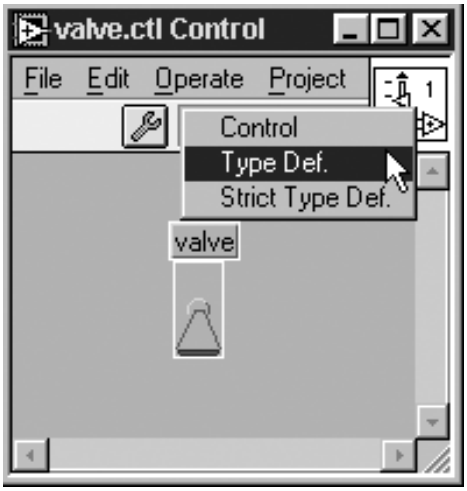



شکل ۱۴-۱۵

نکته‌ی مهم و قابل توجه در مورد المان های کنترل سفارشی آن است که ایجاد تغییر در عملکرد آنها امکان پذیر نیست و تنها می توانید ظاهر آنها را تغییر دهید. به عنوان مثال یک المان سفارشی که بر مبنای کلید لغزشی طراحی شده است، همواره دارای پارامتری است که در یک راستا می لغزد.

برای ایجاد یک نسخه‌ی اصلی از المان کنترل سفارشی، مطابق شکل ۱۵-۱۵ گزینه‌ی Type Def. را از نوار

ابزار در پنجره‌ی Control Editor انتخاب کنید. بدین ترتیب در صورت ایجاد هر گونه تغییر در نسخه‌ی اصلی، تمامی نسخه های موجود در برنامه هایی که از این المان استفاده کرده‌اند به طور خودکار اصلاح می شوند.




شکل ۱۵-۱۵

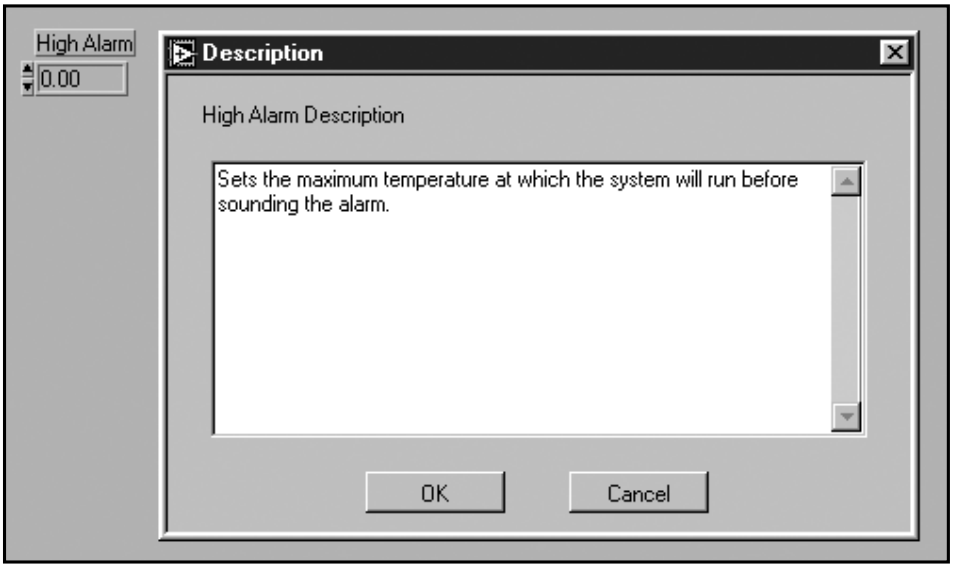
## روش ایجاد پنجره‌ی Online Help

با استفاده از پنجره‌ی Help در LabVIEW همواره می توان اطلاعات لازم را در مورد عملکرد برنامه‌ها، گره‌ها، دستورها و... به دست آورد. برخی از کاربران عادت دارند تا به هنگام سیم کشی در صفحه‌ی نمودار بلوکی از این پنجره استفاده کنند. استفاده نمودن از این پنجره روش مناسبی جهت ارتقای سطح برنامه نویسی و جلوگیری از بروز خطاهای احتمالی در حین انجام سیم کشی است. علاوه بر این برای درک مفاهیم و مطالب موجود در برنامه برای خود و کاربران دیگر می توانید توضیحاتی را در پنجره‌ی Help وارد کنید. برای اضافه کردن پنجره‌ی Help دو روش یا دو سطح متفاوت وجود دارد:

- ارائه‌ی توضیح در مورد عملکرد المان های استفاده شده در برنامه : بدین ترتیب در هنگامی که نشانگر ماوس بر روی المان های کنترل یا نشان دهنده قرار می گیرد، همین توضیحات در پنجره‌ی Help ظاهر می شود.
- ارائه‌ی توضیح در مورد عملکرد برنامه : در این بخش در مورد عملکرد برنامه توضیحاتی ارائه می گردد.

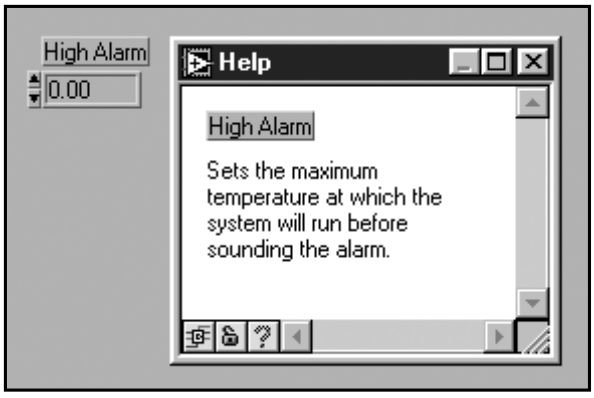
ایجاد پنجره‌ی Help بسیار ساده است. همان گونه که در فصل ۵ عنوان شد، برای انجام این عمل، منوی کرکره‌ای را بر روی المان کنترل یا نشان دهنده‌ی مورد نظر باز کنید و گزینه‌ی ...Data Operations >> Description را انتخاب کنید. در این حالت یک پنجره‌ی محاوره‌ای با عنوان Description ظاهر می شود. متن یا توضیح مورد نظر را در این پنجره وارد کنید و برای ذخیره نمودن آن کلید OK را فشار دهید. از این زمان به بعد در صورتی که پنجره‌ی Help را باز کنید و نشانگر را بر روی المان مذکور قرار دهید، همین توضیحات در پنجره‌ی Help ظاهر می شود.

در شکل ۱۶-۱۵ چگونگی وارد کردن توضیحات را در مورد یک المان ملاحظه می کنید. این توضیحات در پنجره‌ی محاوره‌ای Description وارد می شود.




شکل ۱۶-۱۵

در شکل ۱۷-۱۵ پس از باز شدن پنجره‌ی Help، توضیحات وارد شده در صفحه‌ی Description در این پنجره ظاهر می شود.




شکل ۱۷-۱۵

اگر در حین یا پس از برنامه‌نویسی در مورد تمامی المان‌های قرار گرفته در پنجره‌ی پانل یعنی تمامی المان‌های کنترل و نشان‌دهنده، در پنجره‌ی Description توضیحات لازم را وارد کنید، کاربران دیگر نیز می‌توانند پس از باز نمودن پنجره‌ی Help و مشاهده‌ی توضیحات مورد نیاز در این پنجره عملکرد هر یک از المان‌ها را درک کنند. در نرم‌افزار LabVIEW می‌توانید در مورد عملکرد کلی برنامه نیز توضیحاتی وارد کنید. برای انجام این عمل گزینه‌ی ...Windows >> Show VI Info را انتخاب نمایید و متن مورد نظر را در پنجره‌ی محاوره‌ای ظاهر شده تایپ کنید.

در صورتی که نشانگر ماوس بر روی آیکن این برنامه قرار گیرد، این توضیحات در پنجره‌ی Help به نمایش در می‌آیند. استفاده از این پنجره در هنگام سیم‌کشی در صفحه‌ی نمودار بلوکی بسیار مفید است. همچنین می‌توانید پنجره‌ی Help را از طریق برنامه باز کرده، محل قرار گرفتن آن را تعیین کنید و یا آن را ببندید. برای انجام این عمل از دو دستور Control Help Window و Get Help Window Status استفاده کنید. این دو دستور در زیرپالت Functions >> Application Control >> Help قرار دارند. در ادامه به بررسی این دو دستور می‌پردازیم:

دستور Control Help Window وضعیت باز یا بسته بودن پنجره‌ی Help را تعیین می‌کند. ورودی Top Left Corner حاوی کلاستری است که شامل دو نمایش‌دهنده‌ی عددی می‌باشد. این دو عدد مختصات گوشه‌ی بالایی سمت چپ پنجره‌ی Help را نشان می‌دهند.

شکل ۱۸-۱۵

دستور Get Help Window Status حالت باز یا بسته بودن و همچنین محل قرار گرفتن پنجره‌ی Help را به‌دست می‌دهد. در خروجی Show وضعیت باز یا بسته بودن پنجره نشان داده می‌شود و خروجی Top Left Corner نیز مختصات گوشه‌ی بالایی سمت چپ پنجره‌ی Help را به‌دست می‌دهد.

شکل ۱۹-۱۵

## خلاصه

در این فصل زیباسازی محیط برنامه‌نویسی LabVIEW مورد بررسی قرار گرفت. در بخشهای این فصل در رابطه با زیباسازی صفحه‌ی پانل و دلایل تزیین و زیباسازی این صفحه به بحث پرداختیم و عنوان کردیم که زیباسازی صفحه‌ی پانل، سهولت در انجام عملیات را برای کاربران دیگر نیز به ارمغان می‌آورد. برای تزیین این صفحه می‌توان المان‌ها را تراز نمود و فاصله‌ی آنها را تنظیم کرد و در صورت لزوم آنها را در لایه‌های مختلف قرار داد. در ضمن وارد نمودن تصاویر از خارج از محیط LabVIEW نیز امکان‌پذیر است. در ضمن با سفارشی کردن المان‌های کنترل و نشان‌دهنده می‌توان حرکت اشیا را در صفحه‌ی پانل شبیه‌سازی کرد.

اضافه نمودن توضیح در مورد برنامه یا المان‌های موجود در صفحه‌ی پانل آن نه تنها برای خود شما مفید است، بلکه این عمل سبب می‌شود تا کاربران دیگر عملکرد برنامه و نقش المان‌های استفاده شده در آن را به خوبی درک کنند.