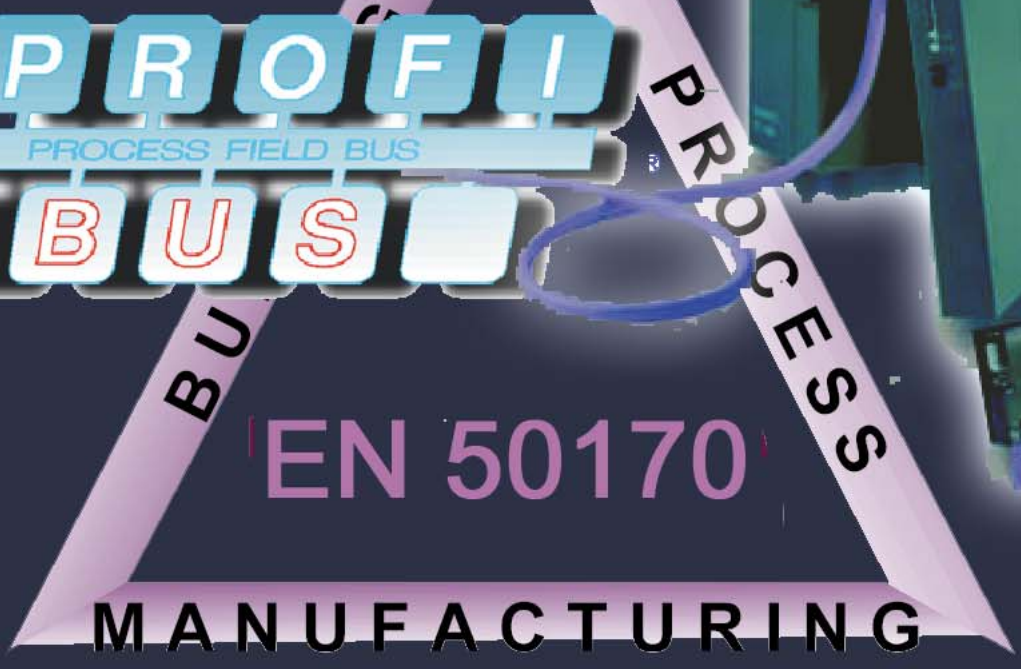
SABCO
BAREZ AUTOMATION CONTROL
SYSTEMS & ELECTRONICS INDUSTRIES

پیکربندی و برنامه نویسی

شبکه PROFIBUS

با نرم افزار STEP7



مولفین:

مهندس محمدرضا ماهر مهندس علی کریم الدینی

بسم الله الرحمن الرحیم

پیکربندی و برنامه نویسی

شبکه PROFIBUS

با نرم افزار STEP7

تالیف : مهندس محمد رضا ماهر مهندس علی کریم الدینی

باهتمام : شرکت صابکو

این صفحه برای مشخصات چاپ مانند تیراژ و شابک و .... رزرو شده است.

پیشگفتار صابکو

پیشگفتار مولفین

سپاس معبود بی همتا را که دست لطفش ما را در تهیه این مجموعه یاری بخشید.
آنچه پیش رو دارید ماحصل نکات مهم و مباحثی است که در مدارک مختلف از جمله مستندات گسترده زیمنس و مدارک موسسه بین المللی Profibus در زمینه شبکه صنعتی Profibus آورده شده است. این مطالب با نگاهی کاربردی و بدون پرداختن به بحث های تئوریک شبکه گرد آوری و عرضه شده است.بمنظور استفاده بهینه از مندرجات این کتاب لازم است خواننده محترم با دو پیش نیاز زیر به اندازه کافی آشنا باشد:

۱. نرم افزار Step7 بویژه ازجنبه پیکر بندی سخت افزار و برنامه نویسی
۲. مفاهیم و اصطلاحات مهم شبکه های صنعتی

شرح مورد اول را باید در کتاب ها و مراجع جداگانه ای یافت ولی دومین موضوع در حد نیاز درانتهای بخشهای اصلی کتاب حاضر بعنوان ضمیمه آورده شده است. توصیه میشود عزیزانی که با مقوله هایی مانند توپولوژی ، تکنیک دسترسی ، مدل OSI و امثال آن کمتر آشنا هستند ابتدا ضمیمه مزبور را مطالعه بفرمایند.
امید است نشر این کتاب که به اهتمام شرکت صابکو انجام شده است بتواند گامی در جهت ارتقاءِ دانش فنی متخصصین عرصه اتوماسیون صنعتی بردارد.از دیدگاههای صاحب نظران که ما را از طریق تماس با پست الکترونیکی reza.maher@gmail.com یا akarimoddini@yahoocom سر افراز نمایند استقبال می کنیم.

مؤ لفین
شهریور ۱۳۸٤

فهرست مطالب

	صفحه
فصل اول – آشنایی با Profibus	
۱-۱ FIELDBUS و تاریخچه آن	۲
۱-۲ PROFIBUS و جایگاه آن	٦
۱-۳ PROFIBUS DP	۹
۱-۳-۱ PROFIBUS DP و نسخه های مختلف آن	۹
۱-۳-۲ تکنولوژی انتقال در PROFIBUS DP	۱۵
۱-۳-۳ تکنولوژی ارتباطات در PROFIBUS DP	۲٤
۱-٤ PROFIBUS FMS	۳۰
۱-۵ PROFIBUS PA	۳۱
فصل دوم – اجزای شبکه Profibus	
۲-۱ اجزای اصلی PROFIBUS DP	٤۰
۲-۱-۱ DP Master	٤۱
۲-۱-۲ DP Slave	٤۳
۲-۲ اجزای اصلی PROFIBUS FMS	٤۷
۲-۳ سایر اجزای شبکه PROFIBUS	۲۵
۲-۳-۱ اجزای شبکه RS485	٤۸
۲-۳-۲ اجزای شبکه فیبر نوری	۵۰
۲-۳-۳ اجزای شبکه بدون سیم	۵٤
فصل سوم – توپولوژی های شبکه PROFIBUS	
۳-۱ توپولوژی های شبکه الکتریکی	۵٦
۳-۲ توپولوژی های شبکه نوری	٦۰
۳-۳ توپولوژی های شبکه بدون سیم	٦٤

فصل چهارم – پیکربندی شبکه PROFIBUS-DP در STEP7

٤-١	بخش های مختلف یک پروژه STEP 7	٦٩
٤-٢	ایجاد پروژه شامل شبکه PROFIBUS -DP	٧٠
٤-٣	ایجاد Master با کارت شبکه	٨٦
٤-٥	استفاده از چند Master System بصورت ترکیبی	٨٨
٤-٦	استفاده از سایر DP Slave ها	٨٩

فصل پنجم – پیکربندی Intelligent Slaves در STEP7

٥-١	پیکر بندی Master با I-Slave	٩٤
٥-٢	پیکر بندی I-Slave با DP Slave	١٠٠
٥-٣	پیکر بندی I-Slave با DP Slave مربوط به DP Master های مختلف	١٠٢
٥-٤	پیکر بندی I-Slave با DP Master	١٠٣

فصل ششم – ساختارهای مختلف شبکه PROFIBUS در STEP7

٦-١	ساختار یک شبکه PROFIBUS در یک پروژه Step7	١٠٦
٦-٢	ساختار چند شبکه PROFIBUS در یک پروژه Step7	١٠٩
٦-٣	ساختار یک شبکه PROFIBUS در چند پروژه Step7	١١٠

فصل هفتم – برنامه نویسی ارتباطات PROFIBUS-DP

٧-١	سرویس DP برای Master / Slave	١١٤
٧-٢	سرویس DP برای Master / I-Slave	١١٦

فصل هشتم –پیکر بندی و برنامه نویسی ارتباط FDL در PROFIBUS

٨-١	ارتباط FDL بین PLC های S7	١٢٨
٨-٢	پیکر بندی ارتباط FDL بین PLC های S7 وS5	١٤٥
٨-٣	ارتباط FDL در Multiproject	١٤٨
٨-٤	ارتباط FDL با Unknown Project	١٤٩

فصل نهم – پیکر بندی و برنامه نویسی شبکه PROFIBUS FMS

۹-۱ شناخت ارتباط FMS ۱۵۲

۹-۲ نحوهٔ پیکر بندی ارتباط FMS ۱۵٤

۹-۳ برنامه نویسی ارتباط FMS ۱۵۸

۹-٤ مثالی از برنامه نویسی ارتباط FMS ۱٦٤

فصل دهم – عیب یابی و مدیریت خطا در PROFIBUS

۱۰-۱ عیب یابی از طریق وسایل تشخیص عیب ۱۷۷

۱۰-۲ عیب یابی از طریق نرم افزارStep7 ۱۸٤

۱۰-۳ مدیریت خطا با استفاده از نرم افزارStep7 ۱۸۷

ضمیمه ۱ – مروری بر مفاهیم و اصطلاحات شبکه های صنعتی

توپولوژی های مختلف شبکه ۱۹۲

واسط های انتقال در شبکه ۱۹٦

تکنیک های دسترسی به شبکه ۲۰۵

لایه های شبکه و مدل OSI ۲۱۰

فاصله همینگ ۲۲۱

ضمیمه ۲ – مقایسه مشخصات برخی از شبکه های فیلدباس ۲۲۸

ضمیمه ۳– پارامترهای شبکه PROFIBUS ۲٤۰

ضمیمه ٤– مقایسه فرمت دیتا در S7 و FMS ۲٤٤

ضمیمه ۵– کد های خطا در ارتباط FMS ۲٤٦

کلمات اختصاری ۲۵٤

منابع و مراجع ۲۵۸

فصل اول – آشنایی با PROFIBUS

مشتمل بر :

۱-۱ FIELDBUS و تاریخچه آن

۱-۲ PROFIBUS و جایگاه آن

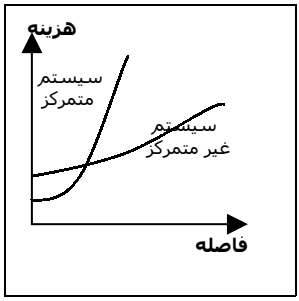
۱-۳ PROFIBUS DP

۱-٤ PROFIBUS FMS

۱-٥ PROFIBUS PA

۱-۱ FIELDBUS و تاریخچه آن

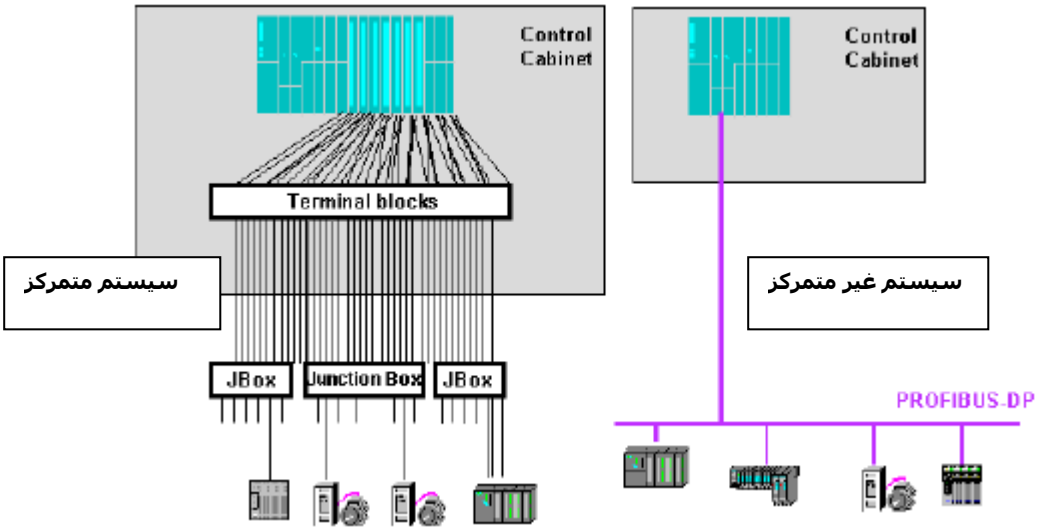
تاقبل از قرن بیستم میلادی سیستم های کنترل فرآیند مبتنی بر تکنولوژی مکانیکی و وسایل آنالوگ بودند . بعد از مدتی تکنولوژی کنترل پنوماتیکی و هیدرولیکی مطرح گردید که کنترل سیستم های Remote را توسط کنترل کننده مرکزی امکان پذیر میساخت. در اوایل ۱۹۶۰ بود که PLC ها پا به عرصه وجود گذاشتند و بتدریج جایگزین مدارات رله ای شدند. با توسعه PLC ها بحث کنترل غیرمتمرکز مطرح شد و اولین سیستم های کنترل غیر متمرکز یا DCS ها در اواسط ۱۹۷۰ عرضه شدند. در همین دوران شبکه های کامپیوتری محلی (LAN) تبادل دیتا بین کامپیوترها را بهبود بخشید و اندیشه استفاده از شبکه جهت مقاصد اتوماسیون صنعتی شکل گرفت



که در دهه ۹۰ با عنوان Fieldbus عملی شد. شبکه های اتوماسیون صنعتی برای کنترل در سطح فیلد که به Fieldbus معروف است مزیتهای بسیاری را برای سیستم اتوماسیون به ارمغان آورد کاهش حجم و عملیات کابل کشی سهولت نصب و کاهش هزینه از آن جمله بود.این کاهش هزینه بویژه وقتی چشمگیر بود که سیگنالهای Field با کنترل کننده فاصله زیادی داشت. نمودار شکل روبرو نشان میدهد که با افزایش فاصله چگونه هزینه های یک سیستم متمرکز افزایش می یابد و این هزینه در مقایسه با استفاده از سیستم توزیع شده چگونه است.

بطور خلاصه مزایای مهم استفاده از Fieldbus را میتوان به صورت زیر بر شمرد:

- کاهش سیم کشی
- کاهش محوطه اشغال شونده جهت نصب
- کنترل صحت اطلاعات و آشکار سازی خطا بدلیل استفاده از سیگنال دیجیتال بجای آنالوگ
- مصونیت بیشتر در مقابل نویز
- تست و عیب یابی راحت تر بدلیل وجود سیستم توزیع شده
- Open بودن سیستم و امکان استفاده از محصولات سازندگان مختلف روی یک شبکه



جایگاه Fieldbus در هرم اتوماسیون

برای سطوح مختلف سیستم های اتوماسیون یک ساختار هرمی شکل مانند شکل صفحه بعد تعریف میشود :

Field Level

در این سطح سنسورها و عملگرها و وسایل ابزار دقیق قرار دارند. حجم دیتا در این سطح کم است ولی زمان ارسال یا دریافت اطلاعات باید کوتاه ودر حد میلی ثانیه باشد.

Control Level

در این سطح سیستم های کنترلی متمرکز (PLC) یا غیرمتمرکز (DCS) ها قرار میگیرند. حجم دیتا نیز در این ناحیه کم و در حد بایت است وزمان تبادل نیز باید کوتاه و کمتر از ثانیه باشد.

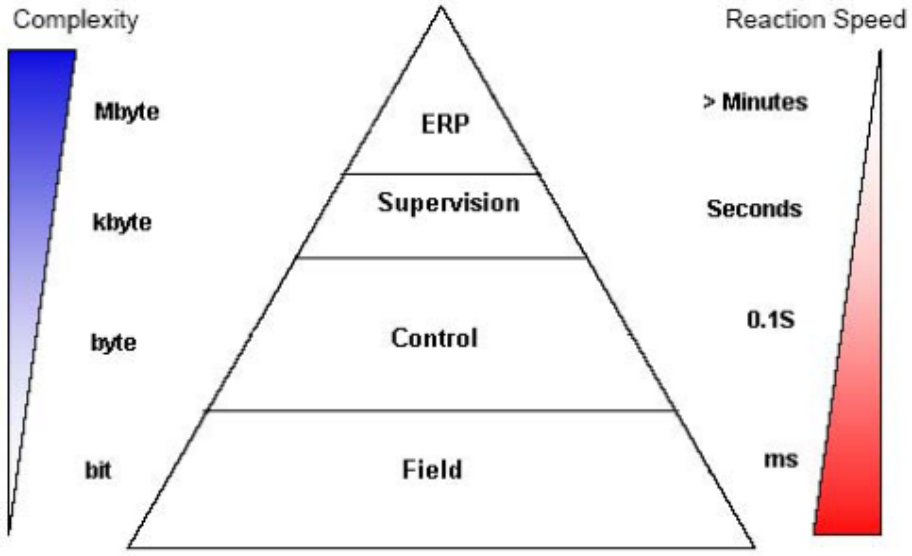
Supervisory Level

در این سطح سیستم های مانیتورینگ یا اصطلاحاً HMI ها قرار میگیرند که اپراتور از طریق آنها وضعیت پروسه را میبینند و فرمانهای لازم را صادر میکند . حجم اطلاعات در حد متوسط و زمان نیز در حد ثانیه است.

ERP Level

سطح Enterprise Resource Planning که به آن اصطلاحات دیگری مانند Factory Level و Management Level نیز اطلاق میشود بالاترین سطح سیستم های اتوماسیون و در واقع سطح اطلاعات مدیریتی است . حجم اطلاعات در این سطح زیاد است بعنوان مثال اطلاعات تولید و تعمیرات و امثال آنها که مربوط به یک شیفت کاری یا یک روز یا یک هفته یا یک ماه است لازم است در این سطح تبادل شود.

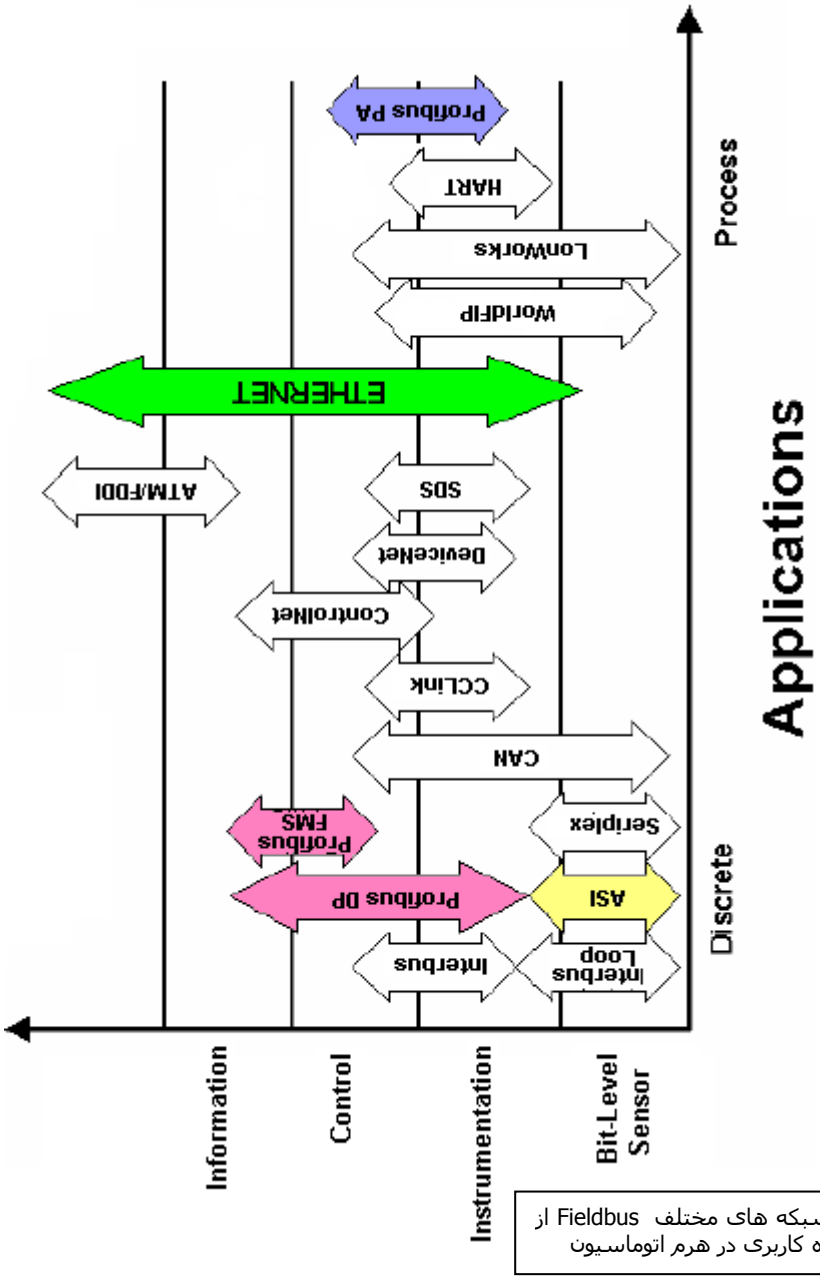
باید دقت کرد در سطوح پایین اگر چه حجم دیتا اندک است ولی زمان دریافت یا ارسال آن بسیار حساس است از اینرو برای این سطوح اصطلاح حساس به زمان یا Real Time بکار برده میشود. بعنوان مثال ممکن است برای یک مدیر در سطح ERP تاخیر چند دقیقه ای برای گرفتن اطلاعات مورد نظر قابل اغماض باشد ولی در سطح کنترل کوچکترین تاخیر زمانی در ارسال اطلاعات از کنترل کننده ممکن است منجر به ایجاد خسارت و توقف فرآیند شود.



وقتی صحبت از جایگاه Fieldbus در هرم اتوماسیون میشود بیش ازهمه ذهن به سطح Field معطوف میشود یعنی شبکه کردن سنسورها و عملگر ها . این ذهنیت اگر چه درست است ولی در عمل پروتکل های مختلفی که تحت عنوان Fieldbus عرضه شده اند بعضاً پا را فراتر گذاشته و در سطح کنترل نیز کاربرد پیدا کرده اند که Profibus نیز یکی از این موارد است . از پروتکل های مشهور در زمینه Fieldbus میتوان موارد زیر را نام برد :

- ASI
- CAN
- EIB
- FOUNDATION FIELDBUS
- HART
- PROFIBUS
- WORLD FIP
- LON
- SDS
- ControlNet
- DeviceNet
- Ethernet
- Interbus

شکل صفحه بعد حوزه کاربری برخی از شبکه های صنعتی را در هرم اتوماسیون نشان میدهد. برای اطلاعات دقیقتر خواننده محترم میتواند مقایسه بین مشخصات مهم برخی از شبکه های معروف فیلدباس را در ضمیمه ۲ مشاهده نماید.



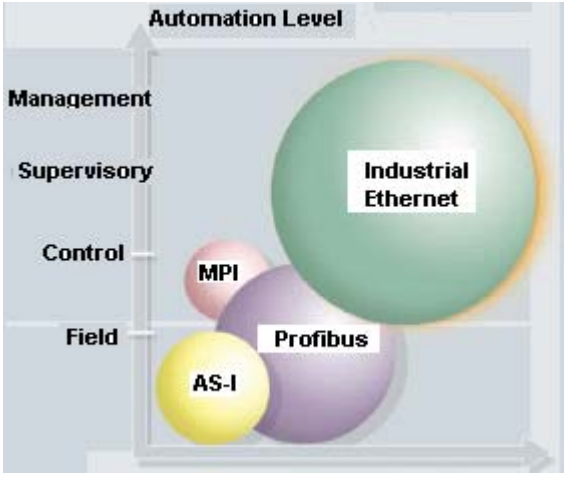
مقایسه شبکه های مختلف Fieldbus از نظر محدوده کاربری در هرم اتوماسیون

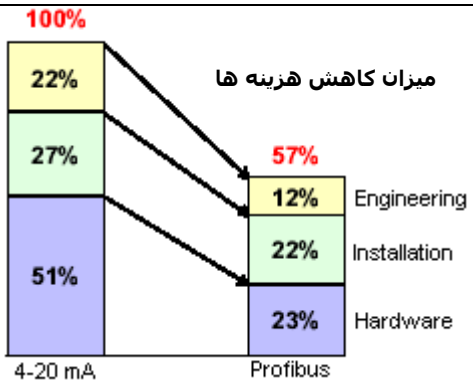
۱-۲ PROFIBUS و جایگاه آن

تاریخچه PROFIBUS که بر گرفته از کلمه *Process Fieldbus* است به سال ۱۹۸۷ برمیگردد. در آن زمان بیش از ۲۰ کمپانی و موسسه آلمانی با یکدیگر پروژه ای را تحت عنوان استاندارد سازی شبکه در سطح فیلد شروع کردند. کمپانی زیمنس نیز در بین آنان بود.هدف پروژه ایجاد یک شبکه باز (Open) بود که بتواندسیستم های کنترل موجود مانند PLC و DCS را پوشش دهد. پس از ۳ سال تلاش در سال ۱۹۹۰ PROFIBUS FMS ارائه گردیدکه برای ارتباطات پیچیده کنترلی بکار میرفت و هنوز نیز کاملا کنار گذاشته نشده است.سپس در سال ۱۹۹۳ PROFIBUS DP طراحی شد که تا امروز ۳ نسخه از آن تحت عنوان DP-V0 , DP-V1 , DP-V2 عرضه شده است. علاوه بر این دو بمنظور پوشش دادن نیازهای مربوط به محیط های خطرناک و انفجاری در سال ۱۹۹۵ PROFIBUS PA پا به عرصه وجود گذاشت.

PROFIBUS اگرچه ابتدا در آلمان مطرح شد ولی بسرعت توسعه پیدا کرد و جایگاهی بین المللی بخود اختصاص داد و سازمانی باعنوان *Profibus Neutzer Organization* یا PNO ایجاد شد که بعدا زیر نظر موسسه بین المللی PROFIBUS International (با نام اختصاری PI) قرار گرفت PI در سال ۱۹۹۵ تشکیل گردید و امروزه بیش از 1100 شرکت معتبر عضو آن هستند . هدف اصلی PI توسعه و بهبود تکنولوژی PROFIBUS برای مقبولیت جهانی است.

شکل زیر جایگاه PROFIBUS را در هرم اتوماسیون نشان میدهد . همانطور که ملاحظه میشود گستردگی آن از سطح فیلد تا سطح کنترل میباشد. در سطح بالاتر یعنی Supervisory اگر چه میتوان از PROFIBUS FMS استفاده کرد ولی امروزه Ethernet صنعتی در این سطح عملا جایگزین PROFIBUS شده و بندرت از آن در سطوح بالاتر از سطح کنترل استفاده میگردد.





PROFIBUS مزایایی که برای خانواده FIELDBUS ذکر میشود را با خود دارد با این وجود ذکر ویژگیهایی که ممکن است بعضاً خاص PROFIBUS باشد خالی از فایده نیست :

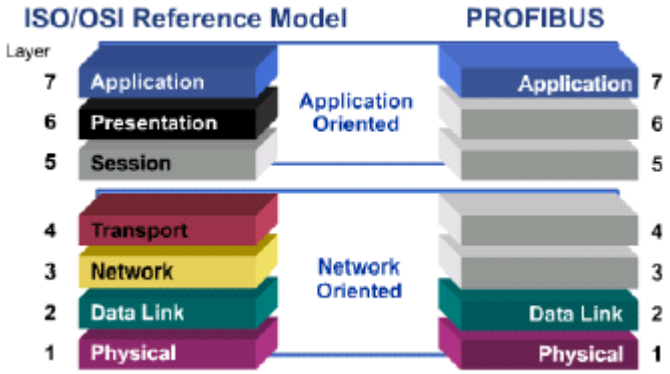
- نویز پذیری کم بعلت استفاده از وسایط انتقال مناسب مانند کابل Twisted Pair
- پهنای باند وسیع بعلت استفاده از روش انتقال مناسب مانند RS485
- تبادل دیتای مطمئن و بدون تداخل بعلت استفاده از روش دسترسی Token Pass
- کاهش هزینه های نصب و راه اندازی بعلت حذف کابل کشی های موازی
- عیب یابی سریع بعلت استفاده از سیستم های توزیع شده
- انعطاف پذیری زیاد جهت توسعه سیستم بعلت Open بودن و عدم انحصار به سازنده خاص

مزایای فوق موجب استقبال از به کار گیری PROFIBUS در شبکه های اتوماسیون صنعتی در سطح جهان شده است . برخی آمارها نشان دهنده سطح این استقبال است مانند جدول زیر:

Bus	User*	Application	Sponsor
CANs	25%	Automotive, Process control	OVDA, Honeywell
PROFIBUS (3 kinds)	26%	Process control	Siemens, ABB
LON	6%	Building systems	Echelon, ABB
Ethernet	50%	Plant bus	all
Interbus-S	7%	Manufacturing	Phoenix Contact
Fieldbus Foundation, HART	7%	Chemical Industry	Fisher-Rosemount, ABB
ASI	9%	Building Systems	Siemens
Modbus	22%	obsolete point-to-point	many
ControlNet	14%	plant bus	Rockwell
*source: ISA, Jim Pinto (1999)		Sum > 100%, since firms support more than one bus	

پروتکل PROFIBUS و انواع آن

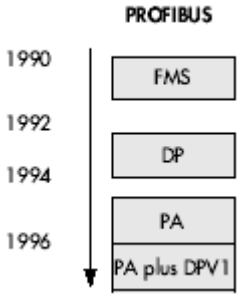
پروتکل PROFIBUS در لایه های خود از مدل ISO / OSI پیروی میکند ولی تمام لایه ها را به کار نمی گیرد شکل زیر نشان میدهد که لایه های ۱ و ۲ و در صورت لزوم لایه هفتم استفاده میشوند.



PROFIBUS در این لایه ها از استاندارد های زیر پیروی میکند:

- EIA RS-485
- IEC 870-5-1
- EN60 870-5-1
- DIN 19245
- IEC 955
- ISO DIS 7498-4

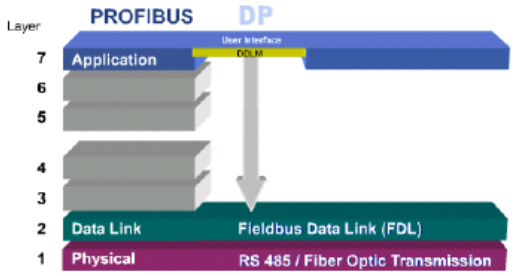
از نظر کاربردی پروتکل PROFIBUS به سه دسته DP و FMS و PA تقسیم میگردد که هریک ویژگی های خاص خود را دارند ودر صفحات آینده به تفصیل شرح داده خواهند شد. توسعه انواع PROFIBUS در شکل زیر آمده است.



PROFIBUS DP ۳-۱

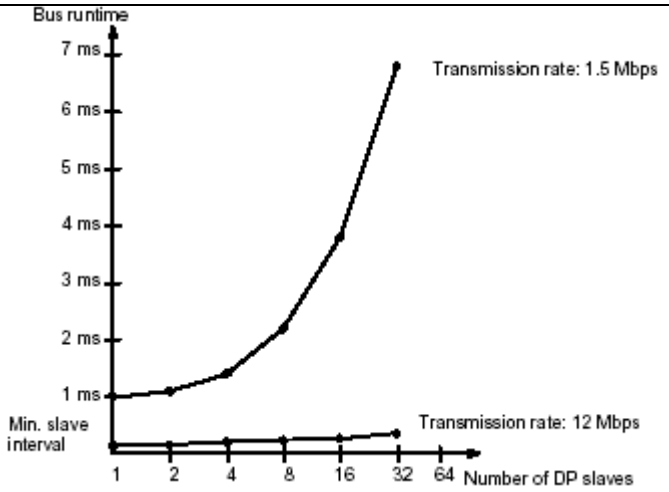
PROFIBUS DP ۱-۳-۱ و نسخه های مختلف آن

PROFIBUS DP از لایه های ۱و۲و همچنین از یک User Interface استفاده می کند، در این پروتکل لایه های ۳ تا ۷ استفاده نمی شوند .این ساختار ساده ،انتقال سریع دیتا را میسر می سازد سرویس DDLM که مخفف کلمه *Direct Data Link Mapper* است امکان دسترسی به لایه دوم را فراهم میکند. توابعی که در این روش به کار می روند و مشخصات سیستم PROFIBUS DP وتجهیزات مورد استفاده در انواع مختلف DP در بخش های بعدی بحث وبررسی می شوند.



USER interface	cyclic exchanged data and status messages	acyclic services - control - diagnostic - parameterizion	 - alarm function - upload - download
DDLM	cyclic and acyclic communication services		
OSI layer 2		Fieldbus Data Link (FDL) Fieldbus Management (FMA)	

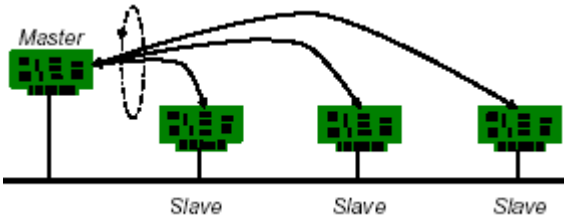
مزیت بزرگDP نسبت به FMS آنست که لایه بالاسری یعنی لایه هفتم در آن حذف شده و اینکار سرعت آن را بهینه کرده است. DP بصورت Master/Slave کار میکند یعنی کنترلر مرکزی یا Master بصورت سیکلی ورودی ها را از Slave ها میخواند و خروجی ها را به آنها میفرستد. به دوره زمانی که در آن این عملیات انجام میشود سیکل باس (Bus Cycle) گفته میشود. این سیکل باید از زمان سیکل برنامه کنترلر مرکزی یعنی Scan Cycle کوتاهتر باشدتا خللی در کار کنترلر پیش نیاید. سرعت DP بگونه ای است که میتواند در عرض یک میلی ثانیه ۵۱۲ بایت دیتای ورودی و ۵۱۲ بایت دیتای خروجی را روی ۳۲ Slave با سرعت 12 Mbps تبادل کند.سرعت انتقال به تعداد ایستگاه و سیکل باس ارتباط دارد . شکل صفحه بعد وابستگی این سه پارامتر را به یکدیگر نشان میدهد.



تاکنون سه نسخه از PROFIBUS DP عرضه شده است DP-V0 و DP-V1 و DP-V2

DP-V0

DP-V0 نسخه پایه است و فقط اجازه ارتباط سیکلی بین Master و Slave را فراهم میسازد. یعنی Master بصورت سیکلی مرتبا Slave ها را یکی پس از دیگری فرا میخواند و با آنها به تبادل دیتا می پردازد. همانطور که میدانیم PROFIBUS DP که از این به بعد ممکن است ما از آن به اختصار با DP یاد کنیم پروتکل پر سرعتی است که بویژه برای ارتباط Master/Slave بهینه سازی شده است. در این ساختار ممکن است فقط یک Master وجود داشته باشد که به آن سیستم Mono Master میگویند و در اینحالت Master بطور نامحدود باس را در اختیار میگیرد. حال اگر بیش از یک Master وجود داشته باشد (Multimaster) در اینصورت Master ها به نوبت باس را در اختیار میگیرند . در هر دو حالت فوق هر Master وقتی باس را در اختیار دارد بطور سیکلی و بترتیب با Slave هایش ارتباط برقرار میکند.



باید توجه داشت که وقتی یک Master میخواهد با Slave صحبت کند ابتدا مراحل آمادگی را انجام داده سپس به خواندن دیتا یا نوشتن آن میپردازد.در مرحله آمادگی (Initialization) سه گام برداشته میشود:

گام ۱: Master درخواست شناسایی وضعیت (Diagnostic) میکند و Slave وضعیت خود را اعلام میکند.

گـام ۲ : Master پارامترهایـی کـه Slave بـاید برای تبادل دیتا از آنها استفاده کند مانند زمان پاسخ دهی را به آن میفرستد . Slave تایید قبول (Acknowledge) را به Master اعلام میکند.

گام ۳ : Master ساختار سخت افزاری پیکربندی شده برای Slave را به آن اعلام میکند. اگر Slave تفاوتی بین آن و ساختار واقعی ببیند به Master اعلام میکند.

پـس از انجام ۳ مرحله فوق کار تبادل سیکلی شروع میشود . یعنی Master درخواست دیتا میکند و Slave به آن پاسخ میدهد. بسته دیتا حداکثر ۲٤٤ بایت و سرویسی که بکار گرفته میشود SRD است.

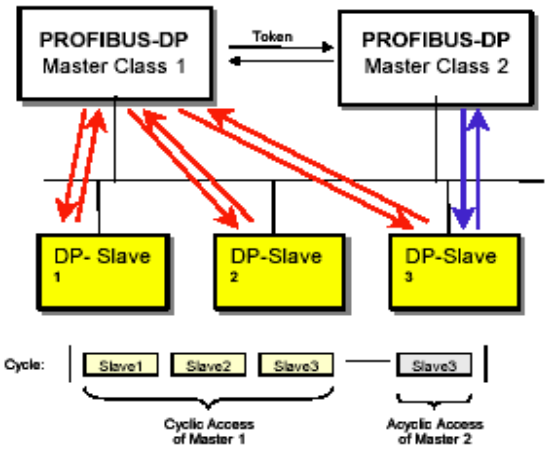
در عیـن حـال کـه تـبادل سـیکلی انجـام میشود میتواند پیغامی توسط Master به تمام Slave ها اعلام شود (یعنی بصـورت Broadcast) و همینطور میتواند پیغامی به گروهی از Slave ها اعلام شود(یعنی بصورت Multicast) که در هر دو حالت از سرویس SDN استفاده میشود یعنی نیاز به تایید دریافت از طرف گیرنده نیست.

DP-V1

نسخه بعدی DP-V1 است که در آن امکان ارتباط غیر سیکلی نیز اضافه شده است برای درک بهتر ارتباط غیر سیکلی باید توجه داشت که در پروتکل DP دونوع Master تعریف شده است :

- DP Master Class 1 کـه DPM1 خوانـده میشود . این همان کنترلر مرکزی است که بصورت سیکلی با Slave ها در دوره زمانی معینی ارتباط برقرار میکند و میتواند یک PLC یا یک PC باشد.
- DP Master Class 2 کـه DPM2 خوانـده میشـود . ایـن وسیله مانند PC یا PG در طول راه اندازی یا تشـخیص عیـب یا برای پیکر بندی و کالیبراسیون و امثال آن به Slave ها متصل میشود. بنابر این ارتباط آن موقت بوده و لازم نیست بطور دائم به باس متصل باشد.

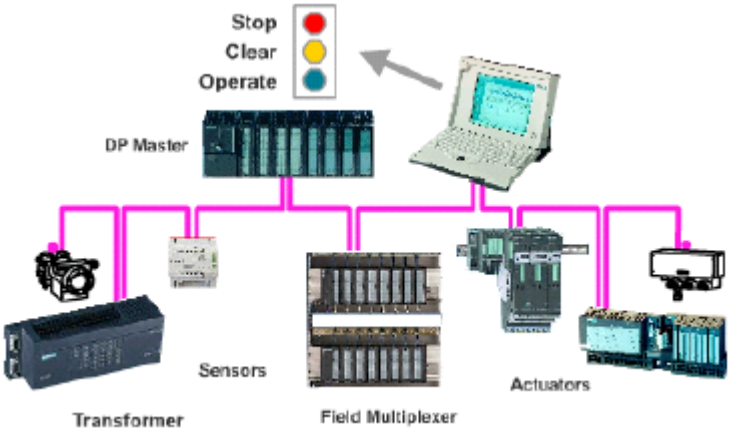
DPM1 بصورت سیکلی و DPM2 بصورت غیر سیکلی ارتباط برقرار میکند :



همانطور که در شکل مشخص است DP Master 1 پرچم را در اختیار گرفته و با Slave هایش ارتباط برقرار میکند وقتی به آخرین Slave رسید پرچم را به DP Master 2 میدهد تا در زمان باقیمانده از سیکل باس با هر Slave که میخواهد بصورت Acyclic ارتباط برقرار کند. در پایان سیکل پرچم دوباره به DP Master 1 برگشت داده میشود.

ویژگی دیگر DP-V1 امکان هدایت خروجی Slave ها به حالت Fail Safe است. برای DPM1 سه مد کاری STOP و CLEAR و OPERATE وجود دارد که قابل تنظیم است.

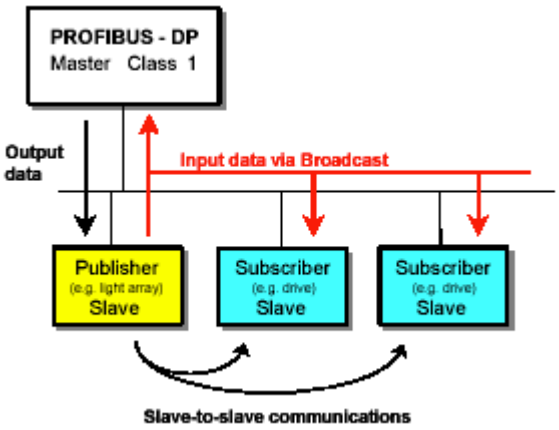
- در مد STOP هیچ تبادل دیتایی با Slave ها وجود ندارد .
- در مد OPERATE کار Master با Slave نرمال است یعنی تبادل دیتا وجود دارد.
- در مد CLEAR اطلاعات ورودی از Slave ها توسط Master خوانده میشود ولی خروجی آنها در حالت Safe قرار میگیرد.



واکنش سیستم در صورت وقوع خطا (مثلا خطا روی یک Slave) بستگی به پارامتر Auto Clear دارد اگر این پارامتر True باشد با بروز خطا روی یک Slave خروجیهای تمام Slave ها توسط Master در حالت Safe قرار میگیرد ولی اگر پارامتر فوق False باشد DPM1 کماکان در وضعیت OPERATE باقی میماند تا اپراتور واکنش مقتضی را به خطا نشان دهد.

DP-V2

بعد از نسخه DP-V1 نسخه DP-V2 عرضه شده است و در آن امکان تبادل دیتای مستقیم بین Slave ها نیز وجود دارد که منجر به صرفه جویی قابل توجهی در زمان میشود زیرا لازم نیست Master دیتا را از یک Slave بگیرد و به Slave دیگر بدهد اینکار مستقیما انجام میشود. این روش بصورت Broadcast است یعنی یک Slave بعنوان Publisher دیتای خود را جهت استفاده سایر Slave ها که Subscriber نامیده میشوند روی باس قرار میدهد. تا در صورت نیاز آنرا خوانده و بعنوان ورودی خود استفاده نمایند.



ویژگی دیگر DP-V2 قابلیت همزمان سازی (Synchronizing) است . اگر لازم باشد که خروجی Slave ها با یکدیگر همزمان شوند میتوان آنها را در این روش در یک حافظه موقت قرار داد سپس با فرمان SYNC بطور همزمان آنها را به خروجی ها ارسال نمود.برای ورودی های Slave ها نیز میتوان عمل همزمان سازی را شبیه خروجی ها ولی با فرمان FREEZE انجام داد.

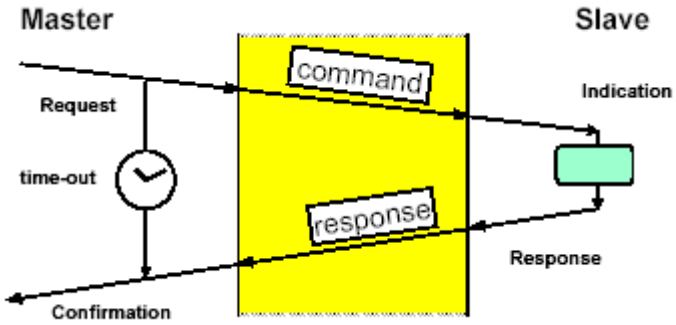
فقط وقتی فرمانهای UNSYNC و UNFREEZE از طرف Master به Slave ها اعلام شود اجازه دارند ورودیها و خروجیهای خود را Update کنند.

فرمانهای SYNC و FREEZE و UNSYNC و UNFREEZE بصورت Broadcast بوده و با سرویس SDN به تمام Slave ها منتقل میشوند.

مکانیسم های حفاظتی

در تمامی نسخه هایV0 و V1 و V2 مکانیسم های حفاظتی برای تبادل دیتا در نظر گرفته شده است. یکی از این مکانیسم ها مانیتور کردن زمان است. در این روش :

- در سمت Master برای DPM1 به ازای هر Slave یک تایمر کنترل دیتای جداگانه وجود دارد. اگر در طول زمان تعیین شده دیتای صحیح از Slave دریافت نشود تایمر مزبور دستور توقف میدهد مگر اینکه مد Auto-Clear فعال شده باشد که در اینصورت DPM1 به کار خود ادامه میدهد و خروجی های Slave مذکور را در حالت Fail Safe نگه میدارد.
- برای هر Slave یک Watchdog منظور شده که خطاهای مربوط به Master یا انتقال دیتا را آشکار می سازد. اگر در طول زمان کنترل شده دیتایی با Master تبادل نشود بطور خودکار Watchdog خروجی های Slave مزبور را به حالت Fail Safe هدایت میکند.

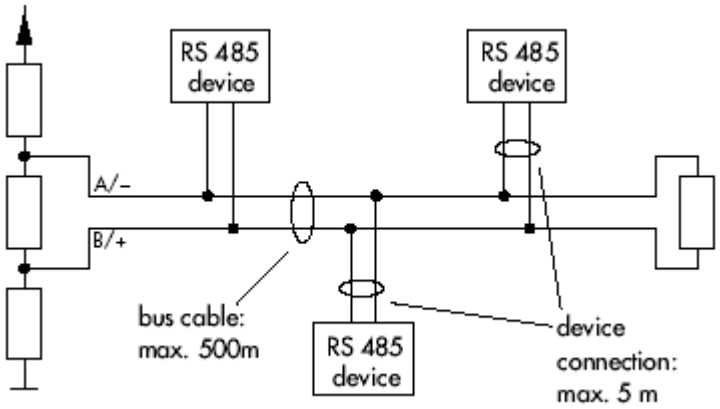


۱-۳-۲ تکنولوژی انتقال در PROFIBUS DP

الف) انتقال با کابل مسی

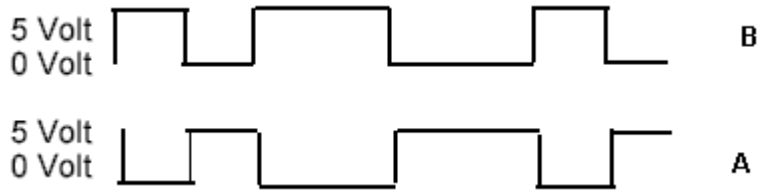
دراین روش در لایه فیزیکی PROFIBUS برای انتقال دیتا از زوج سیم به هم تابیده شیلد دار (STP) مطابق با استاندارد EIA RS 485 موسوم به H2 استفاده می شود.

ارتباط RS485 یک روش ساده و موثر برای انتقال سریع دیتاست که بصورت یک ارتباط دوسیمه Semi-Duplex میباشد یعنی اگر چه چندین وسیله مانند شکل زیر میتواند به آن متصل شود (Multidrop) ولی بعلت وجود ارتباط نیم دوطرفه یا Half-Duplex در هر لحظه فقط یک ایستگاه میتواند فرستنده باشد نه بیشتر. استفاده از RS485 چهار سیمه نیز سیستم را دوطرفه یا Full Duplex نخواهد کرد.زیرا ساختار این استاندارد بصورت نیم دوطرفه است.



سیگنال RS485 برخلاف RS232 که Single_Ended است یک سیگنال تفاضلی است یعنی نسبت به زمین سنجیده نمیشود بلکه سطح ولتاژ تفاضلی بین دوسیم است که صفر یا یک را نشان میدهد. ازاینرو برخلاف RS232 نویز بر آن کمتر تاثیر دارد .

در شکل ٦-١ دو خط انتقال A و Bرا مشاهده می کنید . اگر سیگنال خط B مثبت وسیگنال خط A معکوس سیگنال خط B باشد ،این وضعیت معادل 1 در نظر گرفته می شود.



در این روش:

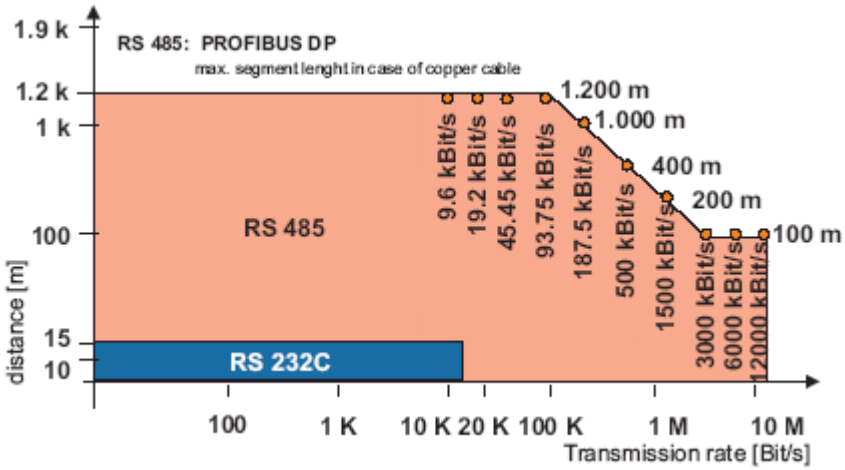
وضعیت	سیگنال تفاضلی بینA و B
1 منطقی	-0.2 to -7 V
0 منطقی	+0.2 to +12 V

همانطور که ذکر شد در استاندارد RS485 یک رشته سیم تک زوج شیلد دار برای انتقال دیتا کافیست. توصیه موسسه بین المللی PROFIBUS بر اینست که ازکابل با مشخصات زیر که کابل نوع A نامیده میشود استفاده گردد :

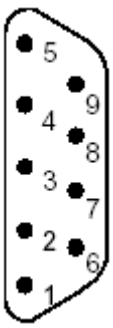
Cable Type A	
Impedance	135-165 Ω
Capacity	<= 30 pf/m
Loop Resistance	<= 110 Ω /km
Wire Diameter	> 0.64 mm
Core Cross-Section	> 0.34 mm2

با این کابل سرعتهای مختلفی بین 9.6 kb/s تا 12 Mb/s قابل انتخاب است. حداکثر تا ٣٢ وسیله را میتوان به کابل فوق متصل نمود که یک Segment را تشکیل میدهد. طول مجاز برای یک سگمنت بستگی به سرعت انتقال دیتایی دارد که می خواهیم در شبکهPROFIBUS برقرار باشد (مطابق جدول زیر) نکته مهم این است که سرعت انتخابی برای یک Segment باید به تمامی تجهیزات متصل به آنSegment اعمال شود.

Data transfer rate in kbit/s	**9.6**	**19.2**	**45.45**	**93.75**	**187.5**	**500**	**1500**	**3000**	**6000**	**12000**
Max. segment length in m	**1200**	**1200**	**1200**	**1200**	**1000**	**400**	**200**	**100**	**100**	**100**

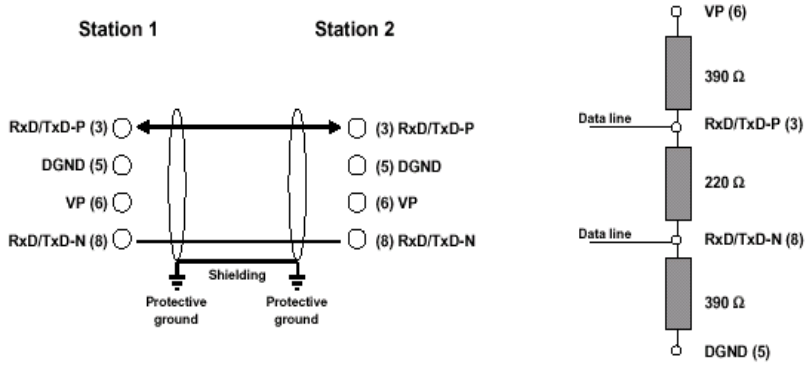


کانکتوری که برای اتصالاتPROFIBUS استفاده میشود طبق استاندارد بین المللی EN 50 170 یک کانکتور 9-Pin Sub D Plug است. به این منظور یک سوکت Sub D بر روی ایستگاه نصب می شود وکابل شبکه از طریق کانکتور Sub D Plug به آن وصل می شود.در جدول ٤-١ مشخصات این کانکتور آمده است.



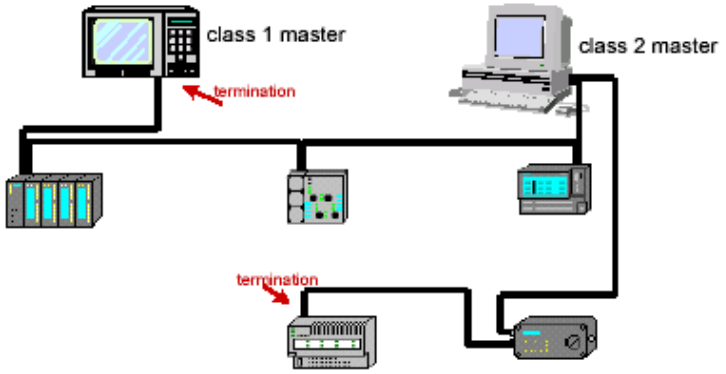
Pin no.	Signal	Significance
1	Shield	Shield/functional ground
2	M24	Ground for +24 V output voltage
3	RxD/TxD-P	Receive/Transmit data – plus (B wire)
4	CNTR-P	Repeater control signal (direction control), RTS signal
5	DGND *)	Data ground (reference potential for VP)
6	VP *)	Supply voltage - plus (P5V)
7	P24	Output voltage +24 V
8	RxD/TxD-N	Receive/Transmit data – minus (A wire)
9	CNTR-N	Repeater control signal (direction control)

ابتدا و انتهای هر سگمنت باید توسط Terminator بسته شود . بر طبق استاندارد RS 485 در انتهای خطوط انتقال دیتا یک مقاومت Pull-Down (در طرف DGND) و یک مقاومت Pull-Up در طرف VP(منبع تغذیه) قرار داده میشود که به آنها Terminator میگویند.



این دو مقاومت باعث می شوند زمانی که باس آزاد است وهیچ کدام از ایستگاهها چیزی روی باس نفرستاده اند یک مقدار ولتاژ معین روی باس قرار گیرد واز بوجود آمدن ولتاژهای تعریف نشده بر روی باس ،جلوگیری شود .ترکیبی که در شکل نمایش داده شده است معمولاً توسط سازنده بر روی کانکتورهای استانداردPROFIBUS تعبیه شده واین امکان وجود دارد که با یک Jumper یا Switch، در انتهای باس Terminator را فعال (ON)یا غیر فعال (OFF) کنیم.

اگر سرعت انتقال دیتا بیش از 1.5 Mbps باشد باید از Terminator هایی با اندوکتانس طولی استفاده کردکه به بار خازنی ایستگاه مورد نظروصل میشوند.این عمل باعث می شود از انعکاس وبازگشت موج دیتا جلوگیری شود.



توپولوژی های قابل اجرا با کابل مسی

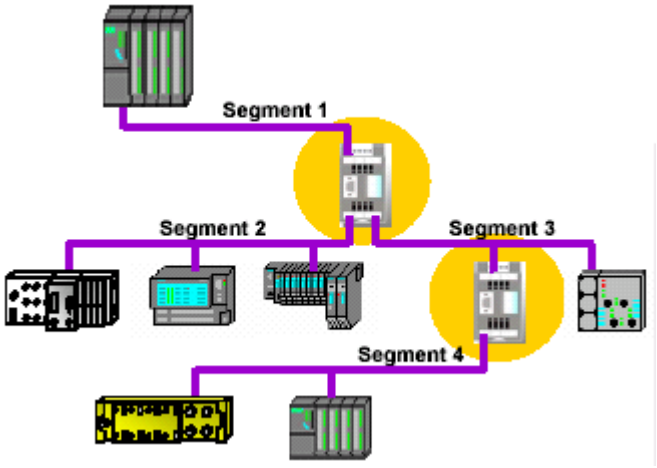
زمانی که محیط انتقال کابل الکتریکی است ونحوه ارسال اطلاعات مطابق با استاندارد RS485 است،امکان پیاده سازی توپولوژی های Bus ، Star و.Tree وجود دارد.اما مرسوم ترین توپولوژی که با کابل الکتریکی پیاده سازی می شود توپولوژی Bus است که در ادامه به شرح آن می پردازیم.

یک سیستم PROFIBUS متشکل از یک Bus که در دو انتهای آنTerminator های فعال تعبیه شده است را در نظر بگیرید.همانطور که قبلاً گفته شد به این باس ،یک سگمنت RS485 گفته می شود.براساس استانداردRS485 ،تا حداکثر ۳۲ ایستگاه RS 485 امکان اتصال به این سگمنت را دارند.به این ایستگاهها در اصطلاح شبکه،Node گفته می شود.

این Node چه Master باشد وچه Slave یک بار جریانی برای سگمنت محسوب می شود .اگر تعداد Node ها بیش از ظرفیت یک سگمنت باشد یعنی سیستم PROFIBUS از تعداد بیش از ۳۲ Node تشکیل شده باشد در اینصورت باید به چندین سگمنت تقسیم شود.این سگمنت های جداگانه ،توسطRepeater به هم متصل می شوند.در واقعRepeater مانند یک تقویت کننده عمل می کند وسطح سیگنال انتقالی را تقویت می نماید.استفاده ازRepeater باعث ایجاد تأخیر واختلاف فاز در سیگنال ارسالی می شود.بنابراین به علت ایجاد اعوجاج وتأخیر در سیگنال ارسالی ،وعدم امکان بازیابی سیگنال ،تعدادRepeater هایی که می توانند به صورت سری به کار برده شوند بر طبق استاندارد EN 50170 حد اکثر ۳ عدد می باشد. این تعداد Repeater،فقط سیگنال را تقویت می کنند واثر چندانی در ایجاد اختلال در سیگنال ارسالی ندارند.

برخی از Repeater ها عمل بازیابی سیگنال (Signal Referesh) را هم به طور محدود انجام می دهند اگر از این نوع Repeater استفاده شود حداکثر ۹ عدد Repeater را میتوان سری کرد و طول کلی شبکه تا ٤ کیلومتر قابل افزایش است. Repeater باکد سفارش 6ES7 971-0A00-0XA0 ساخت زیمنس از این نوع است.

تنها با استفاده ازRepeater امکان دستیابی به ماکزیمم تعدادNode درشبکه PROFIBUS وجود دارد وتنهابا استفاده از Repeater ها است که می توانیم ساختارهای TreeوStarرا ایجاد کنیم.همچنین با بکار بردن Repeater می توانیم یک شبکه بدون Ground داشته باشیم.در این حالت سگمنت ها از یکدیگر ایزوله هستند.



بـه ایـن نکـته توجـه داشته باشید که در ارتباط RS485 ،هر Repeater یک ،Load جریانی محسوب می شود ودر نتیجه در شـمارش Node ها باید منظور گردند.بنابراین با استفاده از هر Repeater از تعداد Node های مجاز یک عددکاسـته مـی شود.به عنوان مثال اگرچه در یک سگمنت می توانیم سی و دو Node داشته باشیم اما با بکار بردن یـک Repeater ،حداکـثر ۳۱ ایسـتگاه مـی تواننـد به این سگمنت وصل شوند .دقت شود Repeater صرفاً از این جهـت یـک Node محسـوب مـی شـود کـه به عنوان یک Load جریانی عمل می کند وگرنه در پیکر بندی شبکه تأثیری ندارد وبه آن آدرسی تخصیص داده نمی شود.

Stub Line

اصطلاح Stub Line به اتصال مستقیم یک ایستگاه به محل اتصال باس می گویند.

در اسـتاندارد EN 50 170 بـرای دسـتیابی بـه سـرعت 1.5Mbps تنها اجازه استفاده از Stub Line با طول حداکثر 6.6m در یک سگمنت داده می شود.اما بهترین حالت این است که از Stub Line استفاده نشود.تنها استثنا ،استفاده از Stub Line برای اتصال موقتی دستگاههای Programmer مثل PG ویا وسایل عیب یابی می باشد.

ایـنکه توصـیه مـی شود از Stub Line استفاده نشود به این علت است که بسته به طول وتعداد Stub Line ها،میزان انعکـاس وبازگشـت سـیگنال اطلاعـات وایجـاد خطـا در طـول سـگمنت بالا می رود و به همین دلیل است که برای سـرعت هـای بالاتـر از 1.5Mbps بـه هـیچ عـنوان ،امکـان اسـتفاده از Stub Line موجـود نمـی باشد.در این حالت تجهیزات عیب یابی یا Programming تنها امکان اتصال به صورت ایستگاه Active را دارند.

ب) انتقال با فیبر نوری

در لایه فیزیکی PROFIBUS امکان استفاده از تکنولوژی فیبر نوری وجود دارد .این روش بر اساس استاندارد های PNO طراحی شده است. کابل نوری این امکان را فراهم می سازد که فاصله ایستگاههای شبکه PROFIBUS تا 15Km هم افزایش یابد نویز های الکترو مغناطیسی بر روی سیگنال نوری اثر نمی کنند . وایستگاههای شبکه از نظر الکتریکی ، کاملاً از یکدیگر ایزوله هستند.

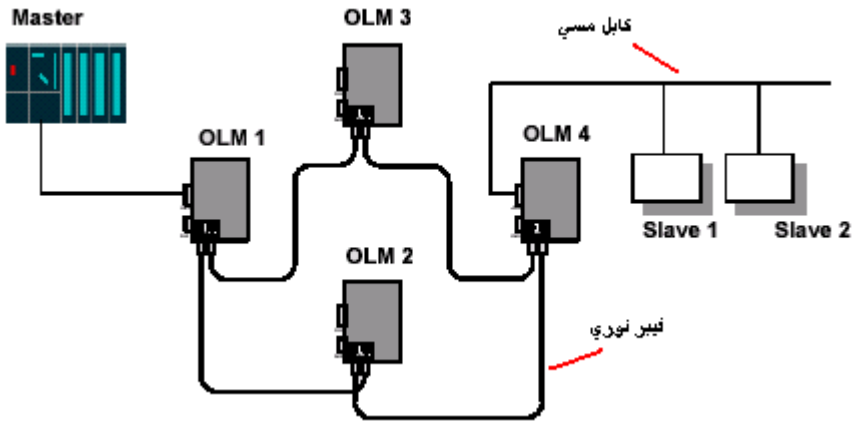
با ساده تر شدن نصب وراه اندازی تجهیزات کابل نوری در سالهای اخیر،این روش انتقال دیتا از محبوبیت خاصی برخوردار شده است.به خصوص تجهیزات کابل نوری پلاستیکی که نسبت به نوع شیشه ای ساده تر وارزانتر است کاربرد بیشتری پیدا کرده اند.کابل های فیبر نوری مشتمل بر فیبر های پلاستیکی وشیشه ای ،به عنوان محیط های انتقال تکنولوژی انتقال نوری استفاده می شوند.فاصله ایستگاهها بسته به نوع کابل نوری که استفاده می شود در نوع شیشه ای تا 15Km ودر نوع پلاستیکی 80m می تواند افزایش یابد.

روش های مختلفی برای اتصال کابل نوری به ایستگاهها وجود دارد:

روش اول: تکنولوژیOLM

مشابه Repeater هایی که در RS485 استفاده می شوند ،OLM دو کابل الکتریکی ایزوله ویک یا دو کانال نوری دارد.OLM از طریق خط RS485 به ایستگاههای شبکه یا سگمنت های الکتریکی وصل می شود. در واقع OLM دو کاربرد می تواند داشته باشد .یکی اینکه به عنوان Repeater در شبکه فیبر نوری استفاده شود، دوم اینکه میتواند شبکه فیبر نوری را به شبکه کابل الکتریکی وصل کند.

در شکل صفحه بعد نحوه استفاده از OLM در یک شبکه PROFIBUS آمده است .همانطور که در این شکل می بینید ایستگاههای Master و Slave با کابل الکتریکی به OLM وصل شده اند وبین خود OLM ها، فیبر نوری کشیده شده است که به کمک این نحوه اتصال حداکثر فاصله مجاز بین ایستگاهها می تواند افزایش یابد .



با استفاده از تکنولوژی فیبر نوری امکان استفاده از توپولوژی های Ring (حلقه) وستاره(Star) ودرخت (Tree) وجود دارد.

مدولهای نوری یعنی OLM ها در ساختار Ring ، هم می توانند به صورت تک حلقه ای و هم به صورت دو حلقه ای استفاده شوند.در حالت دوحلقه ای ،یکی از حلقه ها عملکرد Redundant دارد.به این معنی که اگر برای یک حلقه مشکل پیش بیاید حلقه دیگر وارد عمل می شود.

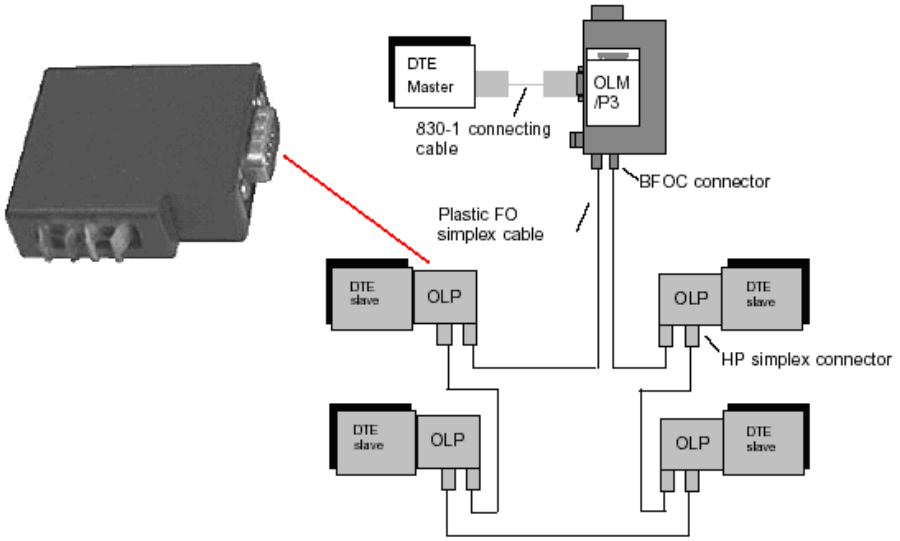
توجه دارید که در حالت تک حلقه ای ،اگر برای یکی از OLM ها مشکلی پیش بیاید یا حلقه قطع شود،کل شبکه ازکار می افتدولی در حالت Redundant اگر برای یکی از حلقه ها مشکل پیش بیاید ،شبکه به کار خودش ادامه می دهد.

حداکثر طول کابل نوری بر اساس نوع آن در جدول زیر آمده است.

Fiber type	Core diameter [µm]	Range
Multimode glass fiber	62.5/125	2-3 km
Singlemode glass fiber	9/125	> 15 km
Plastic fiber	980/1000	< 80 m
HCS® fiber	200/230	approx. 500 m

روش دوم :تکنولوژی OLP

OLP ها می توانند به ایستگاههای Slave در حلقه کابل نوری وصل شوند .این کار از طریق یک کانکتور 9-Pin Sub D Plug در ایستگاه Slave صورت می گیرد.به این ترتیب برای اتصال ایستگاهها ی Slave به شکل فیبر نوری لزومی به استفاده ازOLM نیست. OLP تغذیه خود را از طریق باس متصل به ایستگاه Slave دریافت می کند ونیاز به منبع تغذیه جداگانه ندارد .توجه داشته باشید که در این حالت منبع تغذیه 5V+ در کابل RS 485 باید توان جریان دهی حداقل 80mA را داشته باشد. همانطور که در شکل نمایش داده شده است اتصال ایستگاه Master به یک حلقه OLP به یک عدد OLM نیاز دارد و OLP تنهابرای اتصال Slave به شبکه فیبر نوری استفاده می شود.



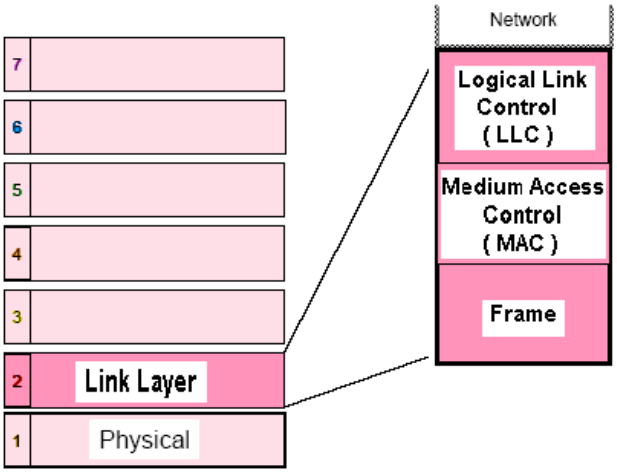
روش سوم :اتصال مستقیم به Intergrated Fiber Optic Connection

بعضی وسایل به صورت مستقیم امکان اتصال به شبکه فیبر نوری را دارند. یعنی یک پورت برای اتصال به شبکه کابل نوری دارند که در نتیجه برای اتصال به شبکه فیبر نوری نیاز به OLM یا OLP ندارند. معمولاً در انتهای نام این وسایل عبارت FO (Fiber Optic) وجود دارد.

CP5613FO ET200S FO

۱-۳-۳ تکنولوژی ارتباطات در PROFIBUS DP

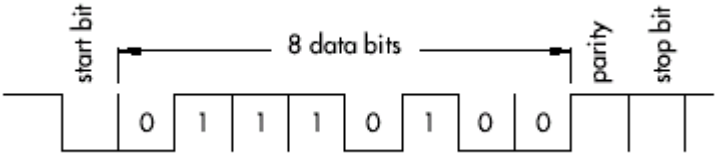
بر طبق مدل OSI نحوه دسترسی به Bus ،امنیت داده ها، پردازش نمودن پروتکل انتقال دیتا از طریق لایه دوم انجام می شود.درPROFIBUS لایه دوم، لایهٔ FDL (FieldBus Data Link) نامیده می شود .فرمت دیتا در این لایه ،امنیت بالایی را در انتقال آن فراهم می کند.



در این لایه کنترل فلوی اطلاعات و کنترل برقراری ارتباط و تشخیص خطا توسط Logical Link Control یا LLC انجام میشود. دسترسی به باس ، چرخش Token توسط MAC (Medium Access Control) صورت میگیرد. به بسته اطلاعات Frame گفته میشود.

الف) فرمت انتقال دیتا و امنیت آن

نحوه آغاز وپایان بسته دیتا وقرار دادن زمان مناسب بین ارسال بسته های دیتا وبه کار بردن بیت Parity وکنترل از جمله مسائلی است که در تعیین دقت وامنیت داده ها مؤثر است .این موارد در شبکه PROFIBUS مطابق با استاندارد IEC870-5-1 تعیین شده اند طبق استاندارد فریم اطلاعاتی از تعدادی بسته دیتا از نوع UART Universal Asynchronous Receiver/Transmitter تشکیل گردیده است . بسته UART بصورت آسنکرون و در ۱۱ بیت مانند شکل صفحه بعد ارسال میشوند.



- ۸ بیت برای دیتا
- یک بیت که شروع دیتا را مشخص میکند Start Bit و همیشه 0 است
- یک بیت که انتهای دیتا را مشخص میکند Stop Bit و همیشه 1 است
- یک بیت که تعداد یک های بسته دیتا را مشخص میکند Parity Bit که زوج بودن تعداد یک ها را چک میکند . Parity=0یعنی تعداد یک ها زوج و Parity=1 یعنی تعداد یک ها زوج نیست.

بدین ترتیب برای یک کاراکتر با کنترل Even Parity امکان آشکار سازی یک خطا وجود دارد اصطلاحا گفته میشود که فاصله همینگ HD=2 است اما همانطور که خواهیم دید یک فریم دیتا صرفاً متشکل از یک بسته UART نیست بلکه دیتاهای دیگری نیز به آن اضافه میگردد. با در نظر گرفتن تمام حالات ممکن ، مشخص شده است که یک فریم اطلاعاتی در Profibus دارای فاصله همینگ HD=4 است. بنابر این تا ۳ خطای متوالی روی بیتها قابل آشکار سازی و یک خطا قابل اصلاح است ازجمله خطاهایی که قابل آشکار سازی است میتوان به موارد زیر اشاره کرد:

- خطا در بیت های شروع
- خطا در بیت های پایان
- خطا در بیت های Parity
- خطا در اندازه فریم
- خطا در بایت کنترلی (FC)

در صورت وقوع خطا اگر چه میتوان تا یک خطا را اصلاح کرد ولی ترجیح داده میشود که بسته دیتا مجدداً ارسال گردد و بصورت اتوماتیک، حداقل یک بار دیگر بسته دیتا ارسال می شود .تکرار ارسال بسته دیتا قابل تنظیم است وحداکثر می تواند مقدار ۸ باشد .این مقدار درپارامترهای باس به عنوان Retryمعرفی می شود که در Setting ها باید آن را، برابر مقدار مطلوب تنظیم کنیم.

با توضیحات فوق اکنون به بررسی ساختار فریم اطلاعاتی می پردازیم . هر بسته اطلاعاتی از تعدادی بسته UART تشکیل شده و ساختاری مانند شکل زیر دارد. قبل از ارسال بسته ، عمل سنکرون سازی انجام میشود.

SYN	SD	DA	SA	FC	DATA	FCS	ED

SD : شروع ارسال دیتا را نشان میدهد و مقدار آن ثابت و برابر A2 هگز است.

DA : آدرس مقصد را نشان میدهد.

SA : آدرس مبدا را نشان میدهد.

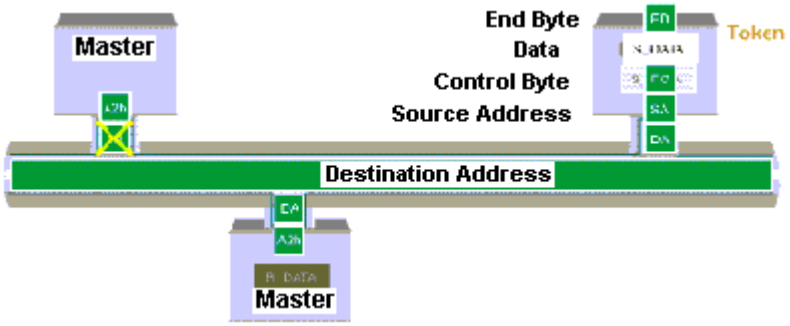
FC : بایت کنترلی است.

DATA : بسته دیتایی که باید ارسال شود.

FCS : وقتی بسته دیتا به چند قسمت تقسیم شده باشد FCS مشخص میکند که این بسته چندمین قسمت است.

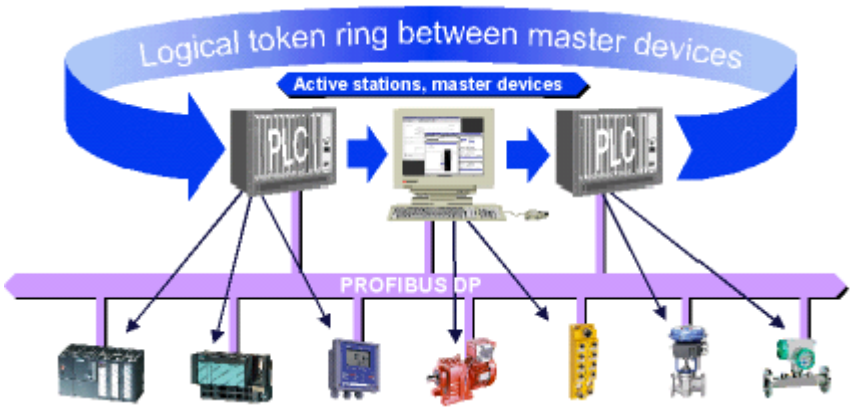
ED : پایان ارسال دیتا را نشان میدهد .

فرستنده دیتا را مطابق با الگوی فوق بسته بندی کرده همراه با Token روی باس قرار میدهد ایستگاه بعدی در حلقه Token آنرا برداشته و آدرس DA را با آدرس خودش تطبیق میدهد اگر یکسان بود بقیه پیام را نیز بازگشایی میکند و اگر آدرس یکی نبود آنرا به ایستگاه بعدی در حلقه Token میفرستد این کار ادامه می یابد تا پیام به گیرنده مورد نظر برسد. گیرنده آنرا از روی الگوی فوق بازگشایی کرده و دیتای اصلی را از بقیه اطلاعات جدا میسازد.



ب) نحوه دسترسی به باس

یکی از مهمترین مسائل در شبکه های صنعتی و از جمله PROFIBUS نحوه ارتباط بین PLC ها با هم ویا با PC هاست.باید هرNode فرصت کافی برای انجام کارهای ارتباطی وانتقال دیتاها درزمانهای معین راداشته باشد.انتقال دیتا بین PLCوPC هاویا انتقال دیتااز Distributed I/O ها باید سریع ودقیق باشد واین نیازمند تعیین یک پروتکل است که بصورت حساب شده ای باس را در اختیار ایستگاهها قرار دهد .به نحوی که ضمن استفاده بهینه از باس ،تداخلی هم بین ارسال اطلاعات بوجود نیاید.در شبکه PROFIBUS نحوه در اختیار گرفتن باس، توسط ایستگاهها به روش Hybride است.این روش ترکیبی است از روشToken Pass(ارتباط بین چند Master) و روش Master-Slave (ارتباط بین یک Master وSlaveهایش) و مطابق با استاندارد EN 50 170 می باشد.



با روش های ذکر شده در فوق ،ترکیب های زیر را می توانیم ایجاد کنیم:

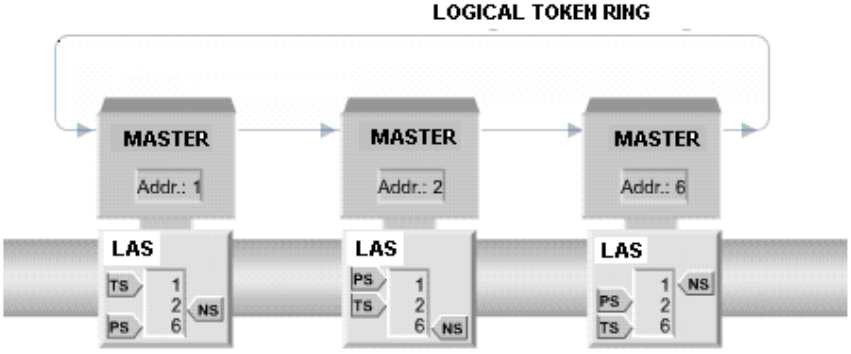
- ارتباط بین چند Master (در روش Token Pass)
- ارتباط بین هر Master با Slave هایش (روش Master-Slave)
- ترکیب دو روش فوق ،یعنی هم ارتباط بین Master ها وهم ارتباط بین Master با Slave هایش (روش Hybride).

تذکر : نحوه در اختیار گرفتن باس ،به محیط انتقال بستگی ندارد وفرقی نمی کند که محیط انتقال کابل مسی باشد یا فیبر نوری.

انتقال اطلاعات بین Node ها ،نیازمند این است که به هر Node یک آدرس یکتا تخصیص دهیم .آدرس های یک شبکه PROFIBUS می توانند بین صفر تا ۱۲۶ تعیین شوند یعنی حداکثر می توانیم ۱۲۷ ایستگاه در یک شبکه PROFIBUS داشته باشیم.

اگر بخواهیم بین یک Master وSlaveهایش ارتباط برقرار کنیم باید از روش Master-Slave استفاده نماییم.همانطور که گفته شد مدیریت باس در این روش بر عهده Master است که تعیین می کند کدام Slave اطلاعات را روی باس بگذارد یا از روی باس بخواند.در واقع با برقراری ارتباط رئیس و مرئوس ،تضمین نموده ایم که هیچگاه تداخل اطلاعات روی باس به وجود نمی آید.

اگر بخواهیم بین چند Master ارتباط برقرار کنیم از روش Token Pass استفاده میکنیم در این روش یک حلقه منطقی (ونه فیزیکی) بین Node ها برقرار می شود.جهت حلقه بر طبق آدرس Node ها از آدرس کمتر به آدرس بیشتر است .Node های تشکیل دهنده این حلقه ،ایستگاههای Master هستند و Token از یک Master به Master با آدرس بالاتر منتقل می شود. واضح است که زمانی که Token به Master با بالاترین آدرس برسد،آن را به Master با پایین ترین آدرس منتقل می کند وبه این ترتیب یک حلقه نرم افزاری تشکیل می شود.



همانطور که در شکل دیده میشود هر ایستگاه دارای یک لیست LAS است که در آن موارد زیر مشخص شده است:

- NS : آدرس ایستگاه بعدی(Next Station) در حلقه Token
- PS : آدرس ایستگاه قبلی(Previous Station) در حلقه Token
- TS : آدرس ایستگاه فعلی(This Station) در حلقه Token

بر اساس این لیست هر ایستگاه میداند Token را از چه ایستگاهی بگیرد و به چه ایستگاهی بفرستد.

فریم Token

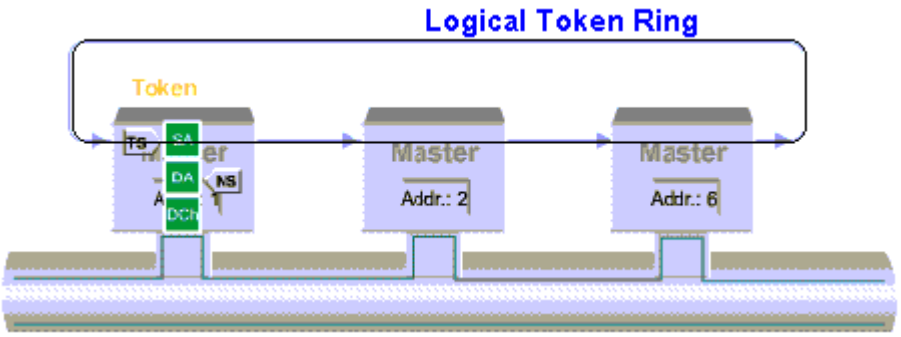
ساختار Token در Profibus بصورت شکل زیر و متشکل از ۳ قسمت است:

SD	DA	SA

SD : شروع ارسال را نشان میدهد و مقدار آن DC هگز است.

DA : آدرس مقصد را نشان میدهد یعنی NS

SA : آدرس مبدا را نشان میدهد یعنی TS



وقتی ایستگاه فعلی (TS)دیتایی برای ارسال ندارد Token را تحویل ایستگاه بعدی (NS) میدهد. ایستگاه بعدی دریافت Token را Acknowledge میکند. اگر ایستگاه فعلی پس از دو بار ارسال Token از ایستگاه بعدی تاییدیه دریافت نکرد در اینصورت NS را از لیست LAS خود حذف کرده و Token را به ایستگاه بعد تر میفرستد.با اطلاعات فوق چرخش Token در حلقه منطقی براحتی انجام میگیرد. لازم بذکر است اگر سیستم monoMaster باشد پرچم Token مرتبا به خود همان Master برگردانده میشود . بعبارت دیگر TS=NS=PS خواهد بود. مدت زمانی که طول می کشد تا Token حلقه را دور زند ودر اختیار همه ایستگاهها قرار گیرد Token Rotation Timeنامیده می شود.ماکزیمم زمانی که Token Rotation Time می تواند طول بکشد ، Ttr (Time Target Rotation) نامیده می شودکه این زمان قابل تنظیم است .همچنین ماکزیمم زمانی که Token به صورت اتوماتیک پس از Configure نمودن سیستم،برای ایستگاه های Master به وجود می آید وبه صورت خودکار ،آدرس هر Node در حلقه Token تعیین می شود .این آدرسها در قسمت (List Of Active Stations) LAS قرار داده می شوند.

۴-۱ PROFIBUS FMS

FMS (*Field Message Specification*) لایه های ۱و۲و۷ مدل OSI را مورد استفاده قرار می دهد . در لایه Application ،سرویس FMS به کار میرود. سرویس قدرتمند FMS در رنج وسیعی از کاربردها ،استفاده می شود وبه خصوص زمانی که ارتباطات پیچیده ای در شبکه برقرار باشد ،سرویس FMS می تواند گزینه مناسبی باشد.

Application Layer Interface (ALI)
Application Layer (7) Field Message Spec (FMS)
Layer 3 to 6 Not Used
Data Link Layer (2) Fieldbus Data Link (FDL) **Physical Layer (1)**

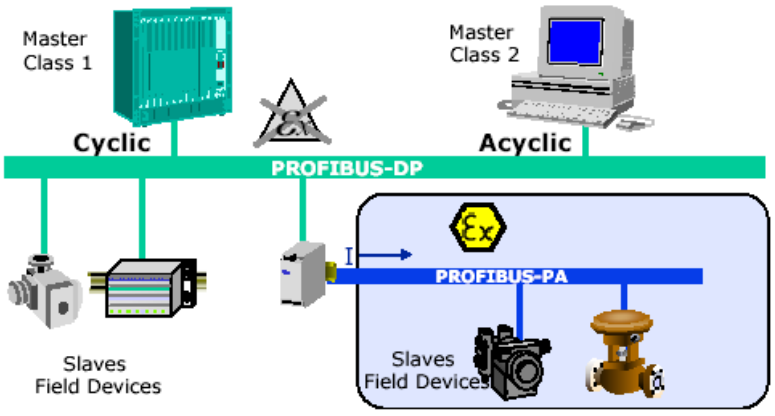
محیط انتقال ونحوه در اختیار گرفتن باس در FMS وDP یکسان می باشد از اینرو می توانند به طور همزمان در یک شبکه قرار گیرند . بطور خلاصه ویژگیهای مهم FMS عبارتند از:

- روش انتقال : کابل مسی (با استاندارد RS485 و با سرعت حداکثر 1500Kbps) یا فیبر نوری
- لایه های مورد استفاده : لایه ۱ و ۲و ۷
- روش دسترسی به باس : Token Pass
- سیگنالینگ : آسنکرون با بسته های ۱۱ بیتی UART (مشابه DP)
- سرویس های ارتباطی : SDA و SDN و SRD

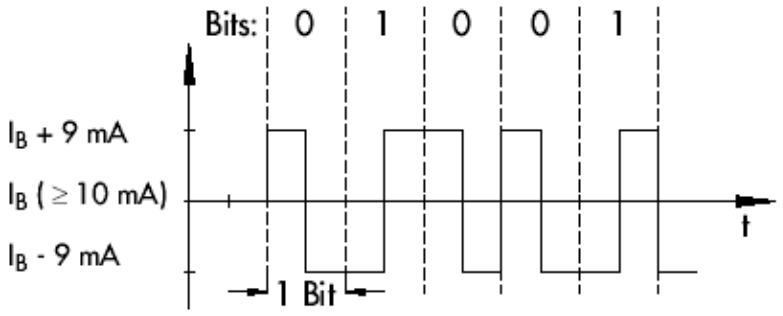
با توضیحات مفصلی که در قسمت قبل راجع به مفاهیم فوق داده شد نیازی به تکرار آنها برای FMS وجود ندارد. معمولاً سرویس FMS را برای حالتی که حجم اطلاعات بالاست به کار می گیرند و جایگاه آن در هرم اتوماسیون بیشتر در سطح Process است. امروزه با توجه به رشد سریع IE(Industrial Ethernet) معمولاً در این سطح به جای استفاده از سرویس FMS از IE استفاده می شود ودر نتیجه سرویس FMS کاربرد محدودتری پیدا کرده است.

PROFIBUS PA ۵-۱

PROFIBUS PA در واقع یک نمونه تکامل یافته PROFIBUS DP است ومعمولاً برای سطح Fleld استفاده می شود. در این روش ،تکنیک انتقال دیتا براساس استاندارد IEC 1158-2 می باشد ودر نتیجه به صورت ذاتی در این روش یک ایمنی بالا به دست می آیدچون تغذیه عناصر متصل به این شبکه مستقیماً از طریق خط ارتباطی تامین میگردد.



انتقال دیتا بر اساس پروتکل Manchester Coding صورت میگیرد که سطح DC برای انتقال دیتا ندارد این روش به MBP یا *Manchester Coded Bus Powered* موسوم ونام دیگر آن H1 است .در پروتکل MBP بیت0 زمانی رخ میدهد که لبه بالا رونده سیگنال جریانی داشته باشیم .وبیت 1زمانی رخ می دهد که لبه پایین رونده سیگنال داشته باشیم مانند شکل زیر :

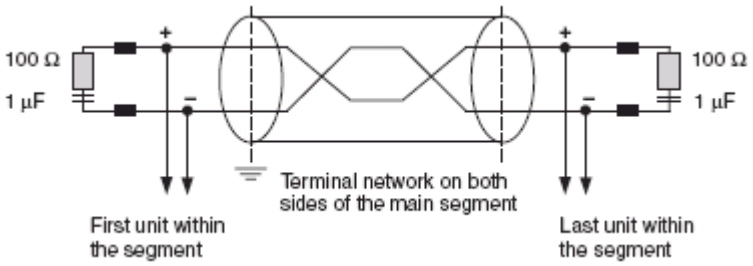


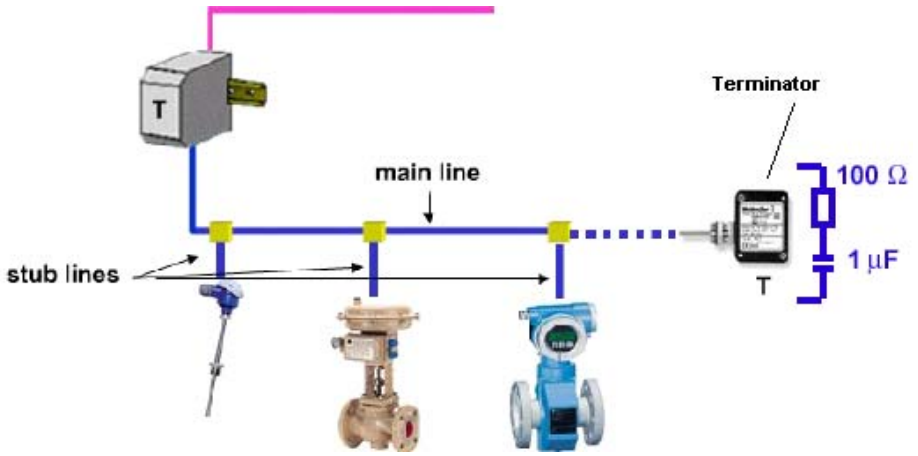
نحـوه انـتقال سیگنال به این صورت است که سطح صفر ویک بصورت 9MA± بر روی جریان باس (IB) مدوله می شوند

در این روش سرعت انتقال دیتا ثابت و برابر 31.25kbps می باشد یعنی به طول کابل بستگی ندارد .محیط انتقال میتواند یک کابل Twisted Pair از نوع شیلد دار (STP) یا بدون شیلد (UTP) باشد. مشخصات فیزیکی و الکتریکی تنها برای چند نوع کابل توسط استاندارد DIN 61158-2 تعیین شده است. این استاندارد برای شبکه PROFIBUS PA کابل های نوع A تا D را پیشنهاد می کند که مشخصات آنها در جدول آمده است .

	Type A	**Type B**	**Type C**	**Type D**
cable design	twisted wire pair, shielded	individual or several twisted wire pairs totally shielded	several twisted wire pairs, not shielded	several not twisted lines, not shielded
wire diameter	0.8 mm^2 (AWG 18)	0.32 mm^2 (AWG 22)	0.13 mm^2 (AWG 26)	1.25 mm^2 (AWG 16)
cable length incl. stub lines	1900 m	1200 m	400 m	200 m

ابتدا و انتهای خط انتقال معمولاً یک مدار RC بعنوان Terminator قرار می دهند.





در شبکه PROFIBUS PA امکان استفاده از ساختارهای Bus،Star، Tree ویا ترکیبی از آنها وجود دارد برای افزایش توانایی سیستم امکان ایجاد یک سیستم Redundant نیز وجود دارد.در حالت ستاره و درختی باید توجه داشت که طول کابل Stub Line باید کمتر از ۳۰ متر باشد.

حداکثر تعداد Node ها که امکان اتصال به یک سگمنت PA را دارند به منبع تغذیه باس، جریان مصرفی Node ها وطول وجنس کابل استفاده شده بستگی دارد.در بیشترین حالت ،۳۲ ایستگاه می تواند به یک سگمنت PA وصل شوند.تعداد ماکزیمم باید با توجه به محدودیتهای زیر محاسبه شود:

محدودیت ها در محیط EEx ia	محدودیت ها در محیط EEx ib
Us=14 to 20 V Ia=110 mA P=1.8 W	Us=14 to 24 V Ia=250 mA P=4.2 W

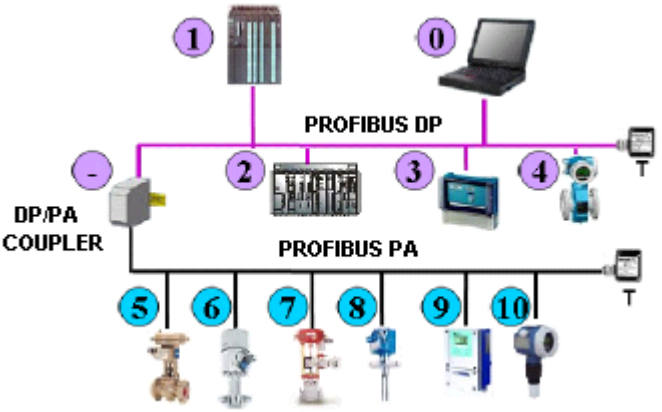
بعنوان مثال فرض کنید سیستمی متشکل از تعدادی Slave مشابه ، قرار است در محیط EEx ia بکار گرفته شود اگر جریان مصرفی هر Slave برابر با ۱۰ میلی آمپر باشد در اینصورت از رابطه زیر تعداد Node ها بدست میآید.

ماکزیمم جریان مجاز = (تعداد Slave) x (جریان Slave) + جریان سیگنال منچستر

۱۱۰ = (تعداد Slave) x (۱۰) + ۹

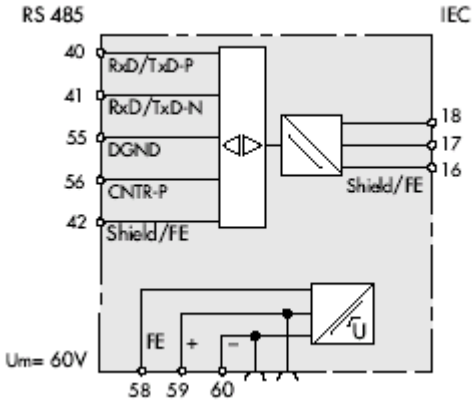
بنابر این ماکزیمم تعداد Slave برابر با ۱۰ بدست می آید. بدیهی است اگر جریان Slave ها متفاوت باشد باید سیگما بکار برد.

میـتوان PROFIBUS PA را از طـریق کوپلـر به PROFIBUS DP وصل کرد.ممکن است عملا این نیاز وجود داشـته باشـد تـا دیتا از شبکه PA به DP یا بر عکس منتقل شود .از آنجا که پروتکل ارتباطی ایندو متفاوت است نـیاز بـه واسـطه ای بـه نـام کوپلـر داریم که در شکل زیر نمایش داده شده است.نکته ای که باید به آن توجه داشت اینسـت کـه پـس از اتصـال دو شبکه به یکدیگر ، آدرس Node ها در کل شبکه باید منحصر به فرد باشند بعلاوه خود کوپلر دارای آدرس نیست.

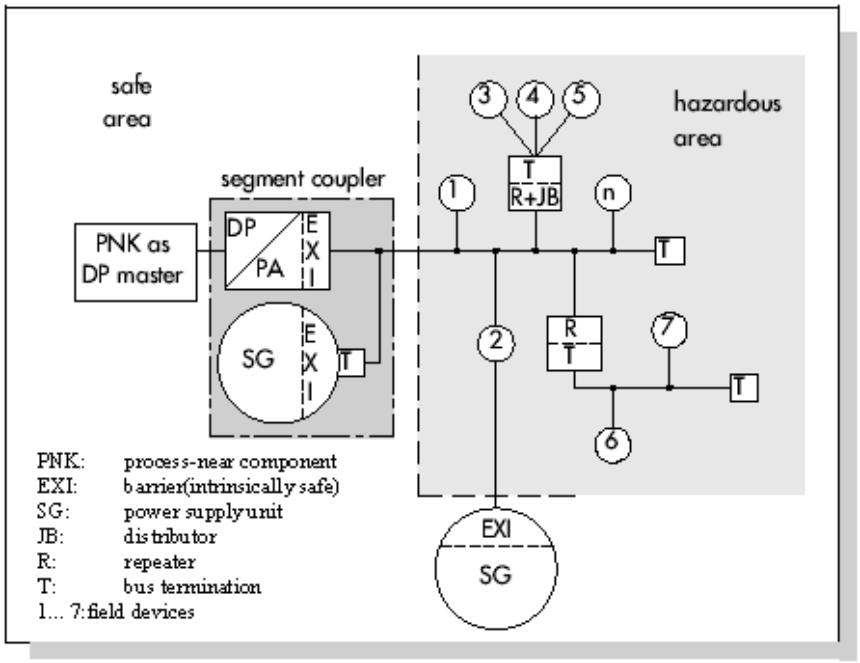


کوپلر معمولا وظایف زیر را بعهده دارد:

- ایزولاسیون الکتریکی بین دو شبکه
- تغذیه کردن شبکه PA
- ایجاد تطابق بین استاندارد های RS485 و IEC 61158-2

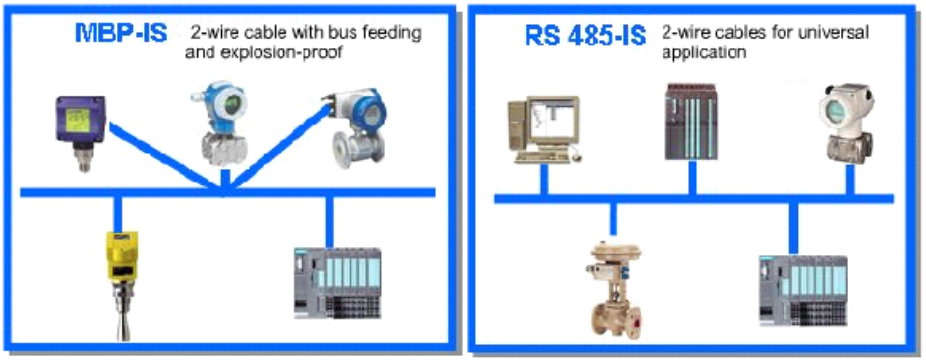


شکل زیر المانهایی که در یک شبکه PROFIBUS PA استفاده میشود را نشان میدهد. جایگاه Terminator ها نیز در آن مشخص است.



RS485-IS

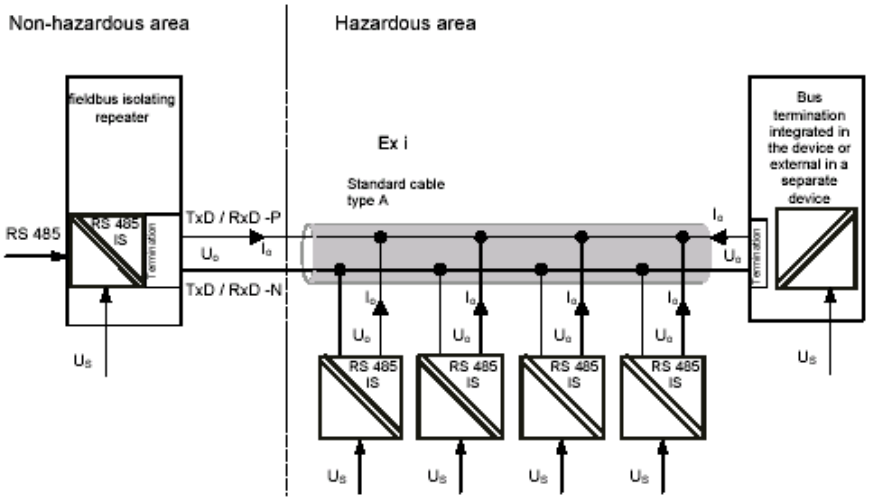
RS485-IS انتخاب جدیدی از خانواده RS485 است و برای محیطهایی که درجه ایمنی بالا نیاز دارند یا اصطلاحا Intrinsically Safe هستند کاربرد دارد هدف از طراحی آن استفاده از سرعت بالا و سایر ویژگیهای RS485 در محیط های خطرناک بوده است .در RS485-IS کابل الکتریکی ٤ سیمه است و سرعت انتقال میتواند تا 1.5 Mbps افزایش یابد بعلاوه تعداد Active Node که در MBP-IS فقط یکی بود ، در این روش تا 32 عدد قابل افزایش است.



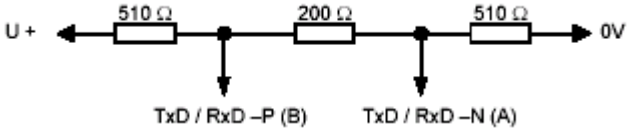
مقایسه ویژگیهای RS485-IS و MBP-IS در جدول زیر آمده است:

	RS485-IS	**MBP**	
Data Transmission	Digital , Differential Singnals according to RS485	Digital , bit-synchronous,Manchester encoding	
Transmission Rate	9.6 to 1500 KBit/s	31.25 KBit/s	
Data Security	HD=4 , Parity bit , start/end delimiter	Preamble , error-protected,start/end delimiter	
Cable	STP , 4wire , cable type A	STP , 2 wire	
Protection Type	Intrinsic Safety (EExib)	Intrinsic Safety (EExia/ib)	
Current / Voltage	Ui=4,2 V Ii=4.8 A Ic<=149 mA/device	EEx ib : Us=14–24 V Ia=250 mA P=4.2 W	EEx ia : Us=14–20 V Ia=110 mA P=1.8 W
Cable Lenght	Max 1000 m	EEx ia : 1900 m	EEx ia : 1000 m
Topology	Line	Line , Tree	
Number of Station	32 per segment , 126 per network	32 per segment , 126 per network	
Number of Repeater	Maximum 9 with signal refresh	Maximum 4	

مدل کلی RS485-IS در شکل زیر آمده است.



ترمینیتور RS485-IS بصورت مقاومتی و مانند شکل زیر است.



فصل دوم – اجزای شبکه PROFIBUS

مشتمل بر :

۲-۱ اجزای اصلی PROFIBUS DP

۲-۲ اجزای اصلی PROFIBUS FMS

۲-۳ سایر اجزای شبکه PROFIBUS

مقدمه

شناخت اجزای سخت افزاری شبکه PROFIBUS پیش نیاز پیکر بندی آنها در محیط نرم افزارStep7 است . شبکه PROFIBUS یک شبکه باز (Open) میباشد و محصولات متنوع سازندگان مختلف با قابلیت اتصال به این شبکه عرضه گردیده است. از اینرو معرفی تمام سخت افزارهای موجود در این مجموعه نه میسر است و نه ضروری. در این قسمت صرفا برخی از المانهای سخت افزاری شبکه PROFIBUS که در نرم افزار Step7 در قالب یک کاتالوگ ارائه شده اند به اجمال معرفی میشوند. دراین خصوص به دو نکته باید توجه داشت:

۱. این المانها مربوط به پروتکل های DP و FMS هستند.
۲. در مورد محصولات سازندگان دیگر (بجز زیمنس) صرفا به نحوه پیکر بندی آنها و آنهم در بخش های بعدی اشاره خواهد شد.

۲-۱ اجزای اصلی PROFIBUS DP

از آنجا که عملکرد DPبر روش Master/Slave استوار است لذا میتوان اجزای این شبکه را به دسته اصلی تقسیم نمود:

- DP Master
- DP Slave

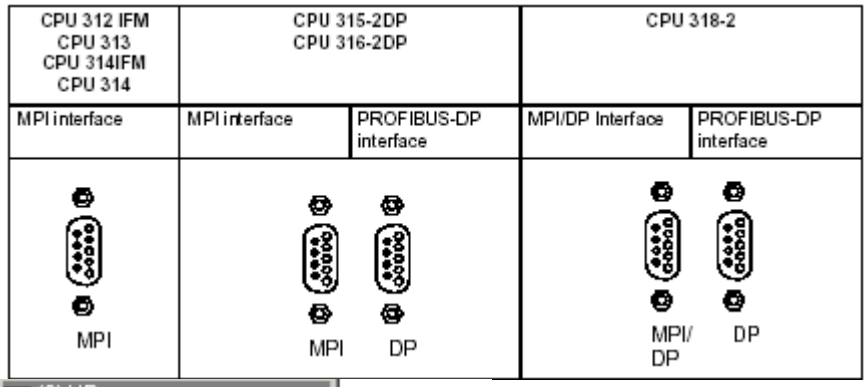
۲-۱-۱ DP Master

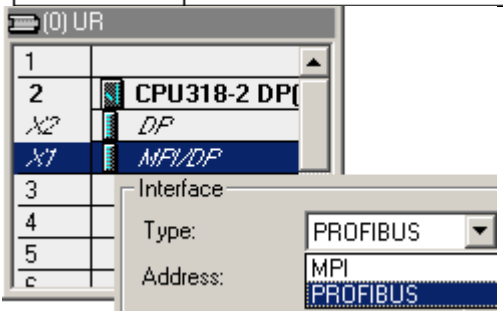
دونوع DP Master در مجموعه S7-300 و S7-400 وجود دارد.

نوع اول : بعضیCPU ها ، یک پورت مخصوص PROFIBUS DP دارند. معمولاً (و نه همیشه) در انتهای نام این CPU ها عبارت 2DP وجود دارد.مثال هایی از این نوع CPU ها به شرح زیر می باشد:

CPU 315-2DP و CPU 318-2DPو CPU 412-1 و CPU 417-4 و...

همه CPU های S7 دارای پورت MPI هستند که برای PC/PG استفاده میشود در عین حال میتوان از آن برای شبکه کردن نیز استفاده کرد. شبکه MPI خاص زیمنس است و استاندارد جهانی ندارد. پورت DP روی همه CPU ها موجود نیست. اگر CPU فاقد پورت DP باشد امکان اتصال مستقیم آن به شبکه PROFIBUS وجود ندارد و برای این منظور باید کارت CP در کنار CPU نصب گردد. پورتهای روی برخی از CPU های 300 در شکل زیر نشان داده شده است.

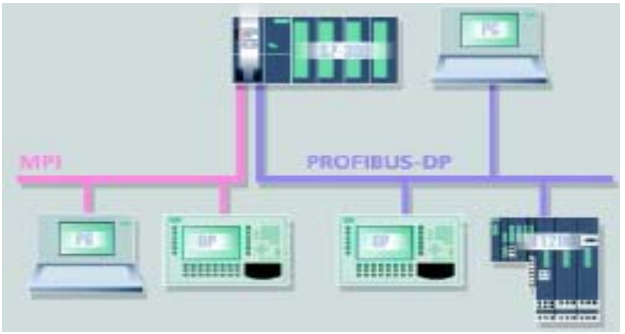




در برخی CPU ها پورت MPI میتواند بصورت DP نیز تنظیم شود . یعنی CPU بعنوان DP Master روی دو شبکه DP که از یکدیگر مستقل هستند عمل میکند. برای این تنظیم کافیست در تنظیمات سخت افزاری CPU در برنامه Hwconfig با کلیک کردن روی MPI/DP قسمت Interface را مانند شکل تغییر دهیم.

باید توجه داشت که در اینحالت ارتباط با PG/PC از طریق پورت DP امکان پذیر است.

نکته دیگری که باید خاطر نشان کرد اینست که وقتی CPU از طریق پورت DP شبکه میشود همزمان میتوان آنرا از طریق پورت MPI نیز به شبکه MPI متصل نمود. مانند شکل زیر.



نوع دوم : اگر CPU فاقد پورت DP باشد یا اگر لازم باشد علاوه بر پورت DP که به یک شبکه PROFIBUS متصل است ، شبکه یا شبکه های DP دیگری نیز داشته باشیم دراینصورت باید از کارتهای زیر استفاده کنیم:

- کارت های CP با قابلیت پشتیبانی PROFIBUS DP مثل :CP 342-5 و CP 343-5
- کارت های IM در S7-400 با قابلیت پشتیبانی PROFIBUS DP مثل IM 467.

تذکر : برخی از CPU های S7-400 دارای اسلاتی برای نصب یک Submodule به نام IF هستندکه از طریق آن میتوان یک شبکه DP جداگانه نیز ایجاد نمود. مثل IF 964-DP در CPU 417-4.

IF964-DP

IM467

CP 342-5

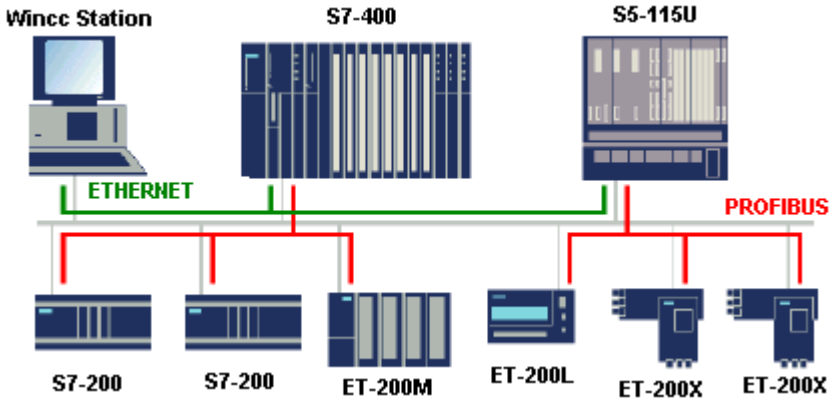
۲-۱-۲ DP Slave

Dp Slave ها در واقع تجهیزات جانبی ونا متمرکز هستند که با Master ارتباط می گیرند .

Slave ها طیف گستردهای از تجهیزات را شامل می شوند. برخی از مهمترین آنها در جدول زیر ذکر شده اند.

عنوان	شرح
Closed Loop Controller	PID کنترلرها
Simoreg	درایوهای DC
Simovert	درایوهای AC
Sipos	عملگرهای برقی
Sensoric	دوربین های صنعتی
Ident	وسیله تشخیص بارکد
IPC	پانل های اپراتوری
NC	دستگاههای کنترل عددی
Switching Device	دژنکتورها
ET	ترمینالهای Remote I/O

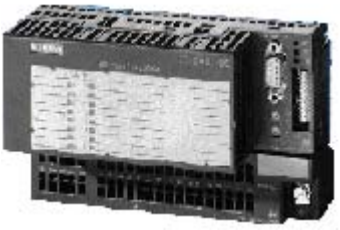
اینها همه تجهیزاتی هستند که امکان اتصال به شبکه PROFIBUS در آنها وجود دارد. اما مشهورترین طیف DP Slave ها،Distributed I/O ها هستند ودر این میان معروفترین Distributed I/O های زیمنس در واقع همان ET ها هستند.وقتی صحبت از Fieldbus میشود معمولا مهمترین مزیت آنرا حذف کابل کشی های موازی و استفاده از Remote I/O برای انتقال سیگنال ذکر میکنند . این کار توسط ET ها انجام میشود. ET ها در سطح فیلد بصورت پراکنده نصب میشوند و در هر منطقه سیگنالهای I/O را جمع آوری کرده و از طریق شبکه به Master انتقال میدهند.



ET ها را می توان به دو دسته Compact و Modular دسته بندی کرد. احتمالاً خواننده با مفاهیم Compact و Modular آشنایی دارد اما برای یادآوری ،وقتی که گفته می شودکه یک وسیله Compact است منظور این است که ساختار آن ثابت است وامکان اضافه یا کم کردن کارت هایی را مانند کارت I/O ندارد اما در نوع Modular ،امکان اضافه کردن تعدادی مدول وجود دارد که به این ترتیب یک ساختار قابل انعطاف در اختیار کاربر قرار می گیرد. مشخصات اصلی برخی از مهمترین ET های زیمنس در جدول زیر آمده است.

ET200L	Compact – مناسب برای تعداد محدودی سیگنال دیجیتال – IP20
ET200B	Compact –مناسب برای تعداد محدودی سیگنال دیجیتال و آنالوگ – IP20
ET200R	Compact –مناسب برای تعداد محدودی سیگنال دیجیتال و آنالوگ –IP65
ET200C	Compact –مناسب برای تعداد محدودی سیگنال دیجیتال و آنالوگ –IP67
ET200M	Modular–مناسب برای حجم زیاد I/O – با کارتهای S7-300
ET200S	Modular– مناسب برای حجم متوسط I/O – دارای استارتر موتور – IP20
ET200X	Modular– مناسب برای حجم متوسط I/O – دارای استارتر موتور و مدول پنوماتیک – IP65
ET200U	Modular–مناسب برای حجم زیاد I/O – با کارتهای S5-100U

ET200R



ET200L

ET200S

ET200M

عناوین فوق هر کدام به یک خانواده اطلاق میشود. هر یک از آنها به چندین نوع با قابلیت های مختلف تقسیم میشوند. عنوان کارت ارتباطی آن IM است و برای هر ET چندین مدل کارت IM وجود دارد.
انتخاب ET مناسب توسط کاربر بسته به نیاز انجام میشود ولی باید خاطر نشان کرد که در بین ET های فوق ET200M پر کاربرد تر ازسایرین است از دلایل این موضوع یکسان بودن کارتهای آن با کارتهای S7-300را میتوان ذکر کرد.
لازم است یادآوری کنیم Distributed I/O هایی که از طریق DP Slave برای DP Master فراهم می شوند دقیقاً مانند I/O هایی که بصورت Central متصل به Master هستند عمل می کنند. تنها استثنا در این مورد CP 342-5 است. این موضوع در هنگام پیکر بندی توسط Step7 روشن تر خواهد شد.

Master با امکان عملکرد Slave

نکته دیگری که در بحث انواع Master و Slave لازم به ذکر است اینست که بعضی DP Interfaceهای متعلق به مجموعه S7-300 مانند CPU 315-2DP ویا CP 342-5 هم می توانند به صورت DP Master و هم به عنوان DP Slave عمل کنند.در حالتی که در این تجهیزات ، حالت DP Slave انتخاب می شود، باید تکنیک در اختیار گرفتن باس را هم معین کنیم. دو مد برای تکنیک در اختیار گرفتن باس قابل تعریف است:

- DP Slave as active node
- DP Slave as passive node

از دیدگاه پروتکل DP نحوه ارتباط وتبادل دیتا در Active DP Slave و Passive DP Slave یکسان است . تنها تفاوت اینست که Active DP Slave علاوه بر ارتباط عادی با Master مربوطه، یک Token هم در اختیار دارد که می تواند با سایر node ها ارتباط بگیرد ومستقیماً به تبادل دیتا بپردازد .این کار از طریق سرویس FDL و توابع S7 صورت می گیرد . قابلیت فوق این امکان را فراهم می سازد که در حالی که شبکه DP در حال تبادل اطلاعات وانجام کارهای مربوطه است ،این وسایل بدون اینکه مزاحمتی برای شبکه DP ایجاد کنند ،از طریق شبکه DP با وسایلی همچون OP ، PG وPC ارتباط بگیرند.به این ترتیب در حالت Passive DP Slave این Master است که تعیین میکند که کدام Passive DP Slave باس را دراختیار بگیرد .اما برای Active DP Slave همانطور که در فوق توضیح داده شد،وضعیت فرق می کند.
تنظیم عملکرد Master یا Slave برای این تجهیزات توسط Step7 انجام میشود که در بخشهای بعدی توضیح داده خواهدشد.

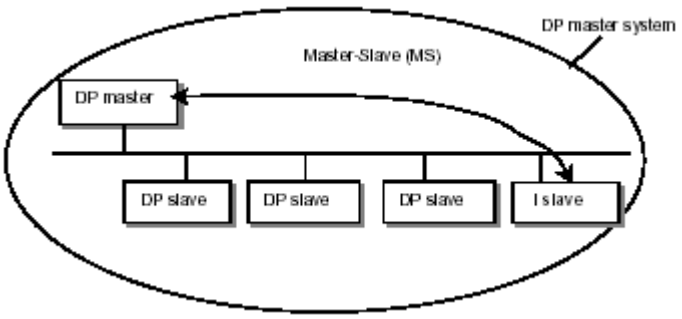
Inteligent Slave

یک نوع DP Slave دیگر نیز وجود دارد که به آن I-Slave گفته می شود و مخفف Intelligent Slave می باشد. انواع I-Slave ها به شرح زیر می باشند:

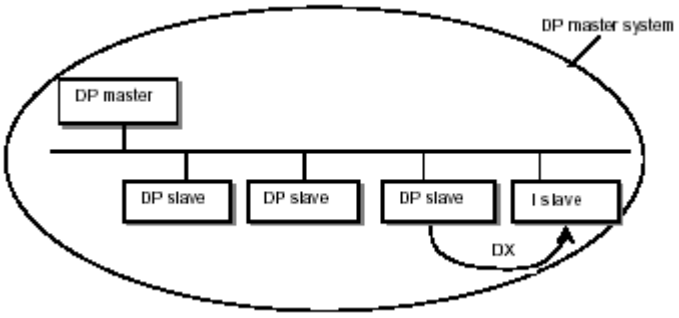
۱- انواع DP Master هایی که قابلیت عمل کردن در مد Slave را هم دارند مثل: , CPU 318-2 , CP 341-5 CPU 316-2DP,CPU 315-2DP

۲- ET های CPU دار مثل ET 200 X و ET 200S

در ارتباط معمولی Master با DP Slave ها،Master مستقیماً به ناحیه I/O در DP Slave دسترسی دارد اما در ارتباط Master با I-Slave ها ،Master مستقیماً به I/O دسترسی ندارد بلکه I-Slave یک پردازش اولیه روی I/O انجام می دهد وسپس تصویر آنها را در اختیار Master قرار می دهد و Master از طریق حافظه I-Slave به I/O های I-Slave دسترسی دارد.



قابلیت دیگر I-Slave امکان تبادل دیتای مستقیم با Slave های دیگر است . قابلیتی که در PROFIBUS DP-V2 منظور شده است.پیکر بندی این روش به تفصیل بیان خواهد شد.



Slave با قابلیت اتصال به فیبر نوری

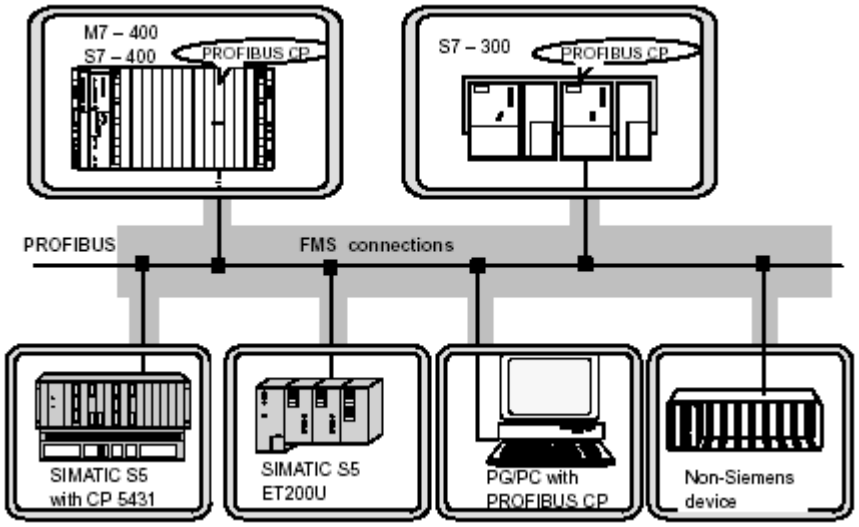
همانطور که قبلا نیز ذکر شد کارتهایی که در انتهای کد آنها کلمه FO نوشته شده قابلیت اتصال به فیبر نوری را مستقیما دارند. در بین ET ها موارد زیر دارای اتصال مستقیم FO هستند:

- ET200M با کارت IM153-2 FO
- ET200S با کارت IM151-1 FO
- ET200x با کارت X-BM 143 FO

۲-۲ اجزای اصلی PROFIBUS FMS

ارتباط FMS همانطور که ذکر شد مبتنی بر تکنیک Token Pass و بین چند Master میباشد. در این روش هر ایستگاه نیازمند کارت شبکه ای است که ارتباط FMS را ساپورت کند.با وجود این کارت سخت افزار خاص دیگری لازم نیست یعنی کابل کانکتور و سایر اجزایی که برای DP استفاده میشوند را نیز میتوان مشترکاً برای FMS نیز بکار برد. برای PLC های S7-300 کارت CP343-5 و برای PLC های S7-400 کارت CP443-5 دارای قابلیت FMS است . با نصب کردن آنها و اتصال به PROFIBUS تبادل اطلاعات با برنامه نویسی قبلی توسط فانکشن های خاص انجام میشود

PLC های S5 ، PG و PC و سخت افزار های غیر زیمنس را نیز با کارت شبکه مناسب میتوان بصورت FMS پیکربندی نمود.



۲-۳ سایر اجزای شبکه PROFIBUS

سایر اجزای مهم شبکه PROFIBUS با تقسیم بندی بر حسب روش انتقال بشرح زیر میباشند:

۲-۳-۱ اجزای شبکه RS485

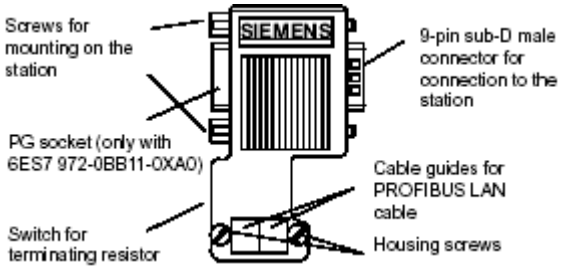
در این روش همانطور که قبلا توضیح داده شد انتقال سیگنال بصورت الکتریکی است. کابل ، کانکتور ، ریپیتر از عمده اجزای این شبکه محسوب میگردند.

کابل شبکه RS485

کابل مسی در شبکه PREOFIBUS یک کابل دو رشته شیلددار یا STP است. مشخصات اصلی این کابل طبق استاندارد در بخش قبل ذکر گردید.امروزه کابل های متنوعی توسط سازندگان مختلف جهت استفاده در شبکه PROFIBUS عرضه میشود. برخی از این کابلها ویژگی های خاص دارند در مقابل آتش سوزی یا تماس با مواد شیمیایی مقاوم تر هستند. بسته به کاربرد باید کابل مورد نظر را انتخاب نمود.

کانکتور RS485

کانکتور های PROFIBUS بصورت ۹ پین male بوده و دارای انواع مختلف هستند که کاربرد آنها متفاوت است بیشتر این کانکتورها دارای ترمینیتور هستند و سوئیچی روی آنها برای ON/OFF کردن ترمینیتور وجود دارد. برخی از انواع کانکتور ها پورتی در پشت دارند که میتوان به آن PG یا PC را نیز متصل نمود.



شایان ذکر است برخی از کابلهای مجهزبه کانکتور نیز عرضه شده اند مانند 830-1 و 830-2

ریپیتر RS485

کاربرد ریپیتر در موارد زیر است :

- وقتی بیش از ۳۲ Node روی شبکه PROFIBUS داشته باشیم.
- وقتی نیاز باشد سگمنت های باس از یکدیگر بصورت الکتریکی ایزوله شوند.
- وقتی تعداد Node ها کمتر از ۳۲ ولی طول کابل به حد ماکزیمم تعیین شده زیر رسیده باشد.

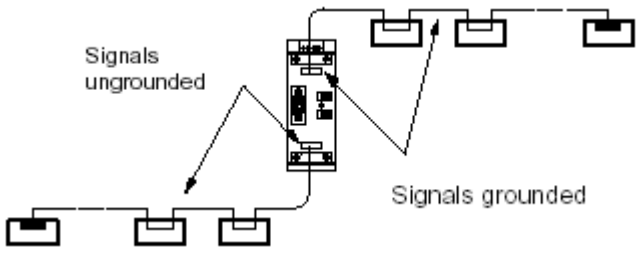
سرعت انتقال	ماکزیمم طول کابل در هر سگمنت بر حسب متر
9.6 to 93.75 Kbps	1000
187.5 Kbps	800
500 Kbps	400
1.5 Mbps	200
3 to 12 Mbps	100

حداکثر ۹ عدد ریپیتر را میتوان در یک شبکه Profibus سری کرد. در اینحالت فاصله بین دو Node نباید از مقادیر مندرج زیر بیشتر باشد.

سرعت انتقال	ماکزیمم طول کابل بین دو Node بر حسب متر
9.6 to 93.75 kbps	10000
187.5 kbps	8000
500 kbps	4000
1.5 kbps	2000
3 to 12 kbps	1000

سرعت انتقال دیتا که باید برای تمام عناصر شبکه یکسان انتخاب شود برای ریپیتر با سوئیچی که روی آن تعبیه شده بین 9.6 Kbps تا 12 Mbps قابل انتخاب است

لازم به ذکر است در محل اتصال هر سگمنت به ریپیتر یک ترمینیتور وجود دارد که با توجه به توپولوژی مورد نظر باید آنها را ON یا OFF نمود. توضیحات بیشتر در بخش بعد آمده است



۲-۳-۲ اجزای شبکه فیبر نوری

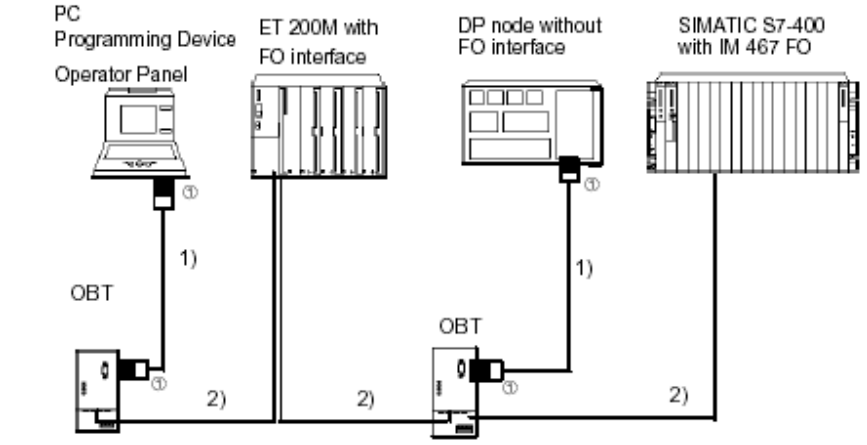
اجزای شبکه فیبر نوری را نیز میتوان به دو دسته Active و Passive تقسیم کرد. OBT و OLM و از اجزای Active و کابل و کانکتور نوری از اجزای Passive هستند که در زیر تشریح شده اند. علاوه بر اینها کارتهای CP و IM هایی که در انتهای کد آنها کلمه FO وجود دارد از اجزای شبکه نوری محسوب میشوند.

OBT

OBT که مخفف Optical Bus Terminal است وسیله ای است که توسط آن میتوان یک Node شبکه الکتریکی RS485 را به شبکه نوری متصل نمود. OBT علاوه براین اتصال نقش یک ریپیتر را نیز بازی میکند. فرض کنید شبکه ای نوری برای PROFIBUS داریم که در آن برخی از تجهیزات مستقیما به آن متصل شده اند. ولی برخی دیگر دارای ارتباط نوری نیستند و فقط پورت RS485 دارند در اینحالت مانند شکل زیر با استفاده از OBT تجهیزات فوق را به فیبر نوری متصل میکنیم. اتصال الکتریکی به پورت ۹ پین RS485 وصل میشود و دو محل اتصال نیز برای فیبر نوری دارد.فقط کابلهای نوری پلاستیکی و پلیمری (PCF) را میتوان به این وسیله متصل کرد و امکان اتصال کابل نوری شیشه ای به آن وجود ندارد از همینجا مشخص است که OBT برای طول محدودی از فیبر نوری میتواند بکار رود (۵۰ متر برای فیبر پلاستیکی و ۳۰۰ متر برای فیبر PCF) .

OBT تمام سرعتهای انتقال دیتا در PROFIBUS را ساپورت میکند یعنی از 9.6 Kbps تا 12 Mbps .

نکته مهم دیگر که باید به آن توجه داشت اینست که دو طرف کابل الکتریکی متصل به OBT باید توسط Terminator بسته شود.



1) Electrical Cable 2)FO Cable

OLM

OLM که مخفف Optical Link Moduleاست نیزوسیله ای است که توسط آن میتوان شبکه الکتریکی RS485 را به شبکه نوری متصل نمود. بر خلاف OBT هر سه نوع فیبر نوری پلاستیکی، پلیمری و شیشه ای را میتوان به آن متصل نمود.

با استفاده از OLM میتوان توپولوژی های باس و ستاره و همچنین حلقوی دوبل برقرار کرد.

بسیاری از OLM ها دارای یک خروجی برای اندازه گیری سیگنال هستند. در این خروجی میتوان کیفیت سیگنال نوری را چک کرد خروجی توسط ولتمتر اندازه گیری میشود و با استفاده از منحنی کیفیت سیگنال که در بخش ۱۰ آمده است وضعیت سیگنال نوری مشخص میگردد.

کانکتور ی که فیبر نوری را به OLM متصل میکند باید از نوع BFOC که در صفحه بعد ذکر شده باشد. OLM ها انواع مختلف دارند که قابلیت های آنها با هم متفاوت است . مهمترین ویژگیهای OLM های عرضه شده توسط زیمنس در جدول زیر مقایسه شده است.

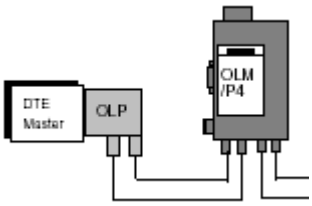
OLM/	P11	P 12	G11	G12 G12-EEC	G11-1300	G12–1300
Number of ports						
– electrical	1	1	1	1	1	1
– optical	1	2	1	2	1	2
Fiber types						
– Plastic optical fibers						
980/1000 µm	80 m	80 m	–	–	–	–
– PCF optical fibers						
200/230 µm	400 m	400 m	–	–	–	–
Quartz glass optical fibers						
10/125 µm	–	–	–	–	15 km	15 km
50/125 µm	–	–	3000 m	3000 m	10 km	10 km
62.5/125 µm	–	–	3000 m	3000 m	10 km	10 km

OLM ها تمام سرعت های انتقال دیتا در شبکه PROFIBUS را ساپورت میکنند یعنی از 9.6 Kbps تا 12 Kbps .

یکی دیگر از ویژگی های مهم OLM توانایی جدا کردن باس در موقع بروز فالت است. یعنی بمحض اینکه خطایی را در کانالی تشخیص دهد آن کانال را بلوکه میکند. این قابلیت باعث میشود که یک شبکه حلقوی در صورت بروز خطا بطور اتوماتیک تبدیل به باس شده و به کار خود ادامه دهد.

شبکه الکتریکی RS485 در موقع اتصال به OLM لازم است توسط ترمینیتور بسته شود.

OLP



OLP که مخفف Optical Link Plugاست کانکتوری است که مستقیماً از یکطرف به پورت RS485 و از طرف دیگر به فیبر نوری پلاستیکی متصل میگردد. OLP میتواند برای ارتباط با OLM استفاده شود. شرایطی که برای استفاده از OLP وجود دارد عبارتند از :

- وسیله مورد نظر دارای پورت RS485 بصورت کانکتور Female نه پین باشد.
- ارتباط PROFIBUS در آن نقطه توانایی تغذیه 80mA را با ولتاژ 5 ولت داشته باشد (پین های ۵ و ۶)
- وسیله مورد نظر یک وسیله Passive روی شبکه Profibus باشد.(یک Slave مانند ET200) البته اگر OLP بصورت نقطه به نقطه به OLM وصل شود میتواند در طرف دیگر به یک Master متصل باشد. جدول زیر امکان ارتباط OLP را به وسایل مختلف نشان میدهد.

Device Name	**M**aster **S**lave	**OLP can be used**
SIMATIC S5		
IM 308–C	M + S	yes
CP 5431 FMS/DP	M	yes
S5–95U/DP	M + S	yes
SIMATIC S7–300		
CP 342–5	M + S	yes
CPU 314	M	yes
CPU 315–2–DP	M + S	yes
SIMATIC S7–400		
CP 343–5	M	yes
CP 443–5	M + S	yes
CPU 413–2 DP	M	no
CPU 414–2 DP	M	no
PC Modules		
CP 5412 A2	M	yes
CP 5411	M	yes
Distributed I/Os		
ET 200M, IM 153	S	yes
ET 200U, IM 318–C	S	yes
ET 200B	S	yes
ET 200L	S	no
ET 200C	S	no
ET 200X	S	no
Miscellaneous		
Repeater RS 485	–	yes
OLM, channel 1	–	no

کابل فیبر نوری

کابل های نوری همانند کابل های الکتریکی در انواع مختلف و توسط سازندگان مخلف عرضه میشوند. برخی از ویژگی هایی که انواع کابلها ی نوری را از هم متمایز میکند عبارتند از :

- جنس هسته
- ضخامت هسته
- تعداد رشته
- پوشش کابل

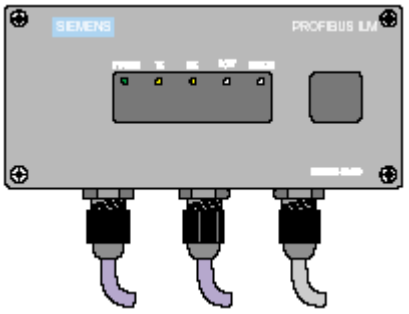
اولین فاکتور در انتخاب کابل های نوری جنس هسته است که میتواند پلاستیکی یا پلیمری یا شیشه ای باشد . نوع پلاستیکی برای مسافتهای کوتاه بکار میرود و ضخامت هسته آن بیش از سایرین (۹۸۰ میکرومتر) است. نوع شیشه ای برای مسافتهای طولانی استفاده میشود و ضخامت هسته آن کمتر از بقیه است (۶۲,۵ میکرومتر). نوع پلیمر ویژگیهایی بین انواع پلاستیکی و شیشه ای دارد.

کانکتورهای فیبر نوری

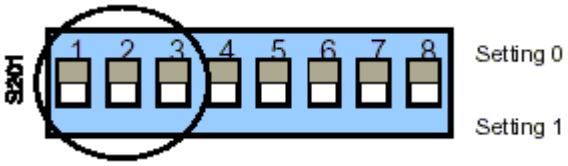
کانکتورهای بصورت Simplex هستند ولی برخی از آنها برای کارتهای خاص مانند IM467 FO و IM153-2 FO باید همراه به آداپتور بسته شوند. برای اتصال دقیق مثلا ارتباط بین دو OLM با مسافت زیاد، کانکتورهای Simplex مناسب نیستند و باید از نوع BFOC استفاده کرد.

BFOC	Simplex with Adaptor	Simplex

۲-۳-۳ اجزای شبکه بدون سیم

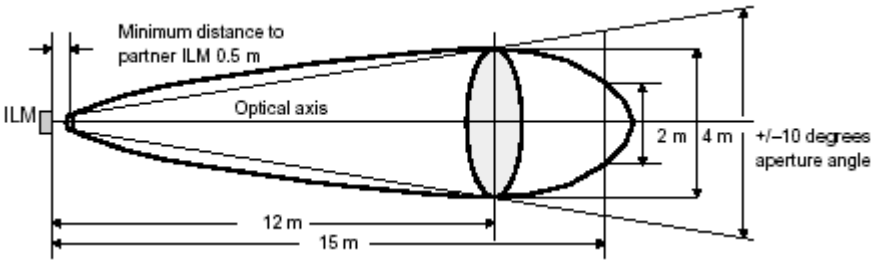


شبکه بدون سیم که برای PROFIBUS استفاده میشود مبتنی بر روش انتقال نور مادون قرمز است. مسافت انتقال سیگنال در این روش کم و حداکثر تا ۱۵ متر است و برای دستگاههای متحرک یا چرخان با فاصله نزدیک مناسب است. وسیله ای که برای انتقال استفاده میشود ILMدارد که مخفف Infrared Link Module است. ILM ها بصورت جفتی بکار میروند و برای تبادل دیتا باید یکدیگر را ببینند. یعنی مانعی بین آنها وجود نداشته باشد.اصول کار بدین نحو است که سیگنال الکتریکی RS485 توسط کابل به ILM ارسال میگردد. ILM آنرا به نور مادون قرمز تبدیل کرده و به ILM دیگر ارسال میدارد. ILM گیرنده سیگنال را باز یابی کرده و مجددا به صورت الکتریکی روی کابل PROFIBUS میفرستد. روش انتقال بصورت Half Duplex است . یعنی در هر لحظه فقط یک ILM میتواند فرستنده باشد.سرعت انتقال بین 9600 bps تا 1.5 Mbps و توسط سوئیچ های 1,2,3 روی ILM قابل تنظیم است مطابق شکل و جدول زیر:



1	2	3	
0	0	0	1.5 Mbps
1	0	0	500 Kbps
0	1	0	187.5 Kbps
1	1	0	93.75 Kbps
0	0	1	19.2 Kbps
0	1	1	9.6 Kbps

ساختار مکانیکی ILM یکپارچه (compact) بوده و دارای IP65 میباشد. هر ILM دارای یک ترمینیتور برای شبکه الکتریکی است که با استفاده از سوئیچ روی ILM میتوان آنرا ON یا OFF کرد. بدیهی است اگر ILM عنصر ابتدایی یا انتهایی باس RS485 باشد لازم است ترمینیتور روی آن را فعال نمود. توضیحات بیشتر را میتوانید روی شکلهای مربوط به توپولوژی ILM در بخش بعد ببینید. فاصله بین دو ILM میتواند بین 0.5 تا 15 متر مطابق شکل زیر باشد.



فصل سوم – توپولوژی های شبکه PROFIBUS

مشتمل بر :

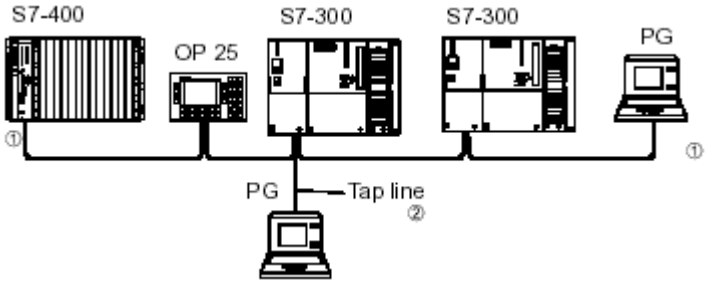
۳-۱ توپولوژی های شبکه الکتریکی
۳-۲ توپولوژی های شبکه نوری
۳-۳ توپولوژی های شبکه بدون سیم

مقدمه

پس از شناخت اجزای مهم شبکه PROFIBUS آشنایی با توپولوژی های مختلف که از ترکیب این اجزا بدست می آید ضروری است. این موضوع در مباحث زیر دنبال شده است.

۳-۱ توپولوژی های شبکه الکتریکی

در شبکه های الکتریکی RS485 توپولوژی میتواند Bus یا Tree باشد. در حالت عادی بصورت باس است.

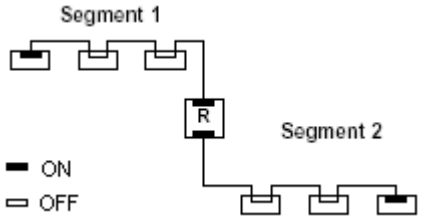


نکات قابل توجه :

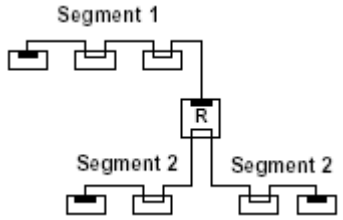
- ابتدای و انتهای باس که در شکل فوق فقط یک سگمنت است باید توسط ترمینیتور بسته شود. پس در این دو نقطه ترمینتور را ON برای سایر اجزای میانی OFF میکنیم. اگر برای یکی از اجزای میانی ترمینیتور ON شود باس کوتاهتر میشود یعنی اجزایی که بعد از آن قرار گرفته اند از باس خارج میشوند.
- PG توسط Tap Line به باس متصل شده است یعنی کانکتور کابل آن به پشت کانکتور S7-300 متصل شده است. در اینحالت نیازی به بستن ترمینیتور در سمت PG نیست.
- در استفاده از Tap Line چه بصورت فوق و چه توسط Bus Terminal ایجاد شود باید به محدودیتهای مندرج در جدول زیر توجه شود:

Transmission rate	ماکزیمم طول Tap Line در هر سگمنت	تعداد Node با Tap Line به طول	
		1.5 m یا 1.6 m	3 m
9.6 – 93.75 Kbps	96 m	32	32
187.5 Kbps	75 m	32	25
500 Kbps	30 m	20	10

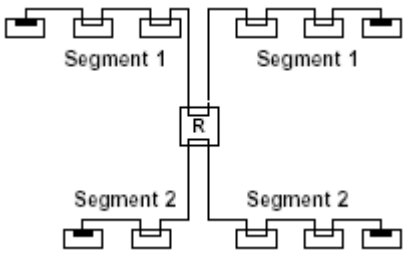
توپولوژی باس با استفاده از ریپیتر



اگر تعداد Node ها بیش از ۳۲ باشد بناچار برای توپولوژی باس باید از ریپیتر مانند شکل روبرو استفاده کرد. توجه شود که ترمینیتور علاوه بر ابتدای سگمنت ها در نقاط اتصال به ریپیتر نیز باید ON شود .



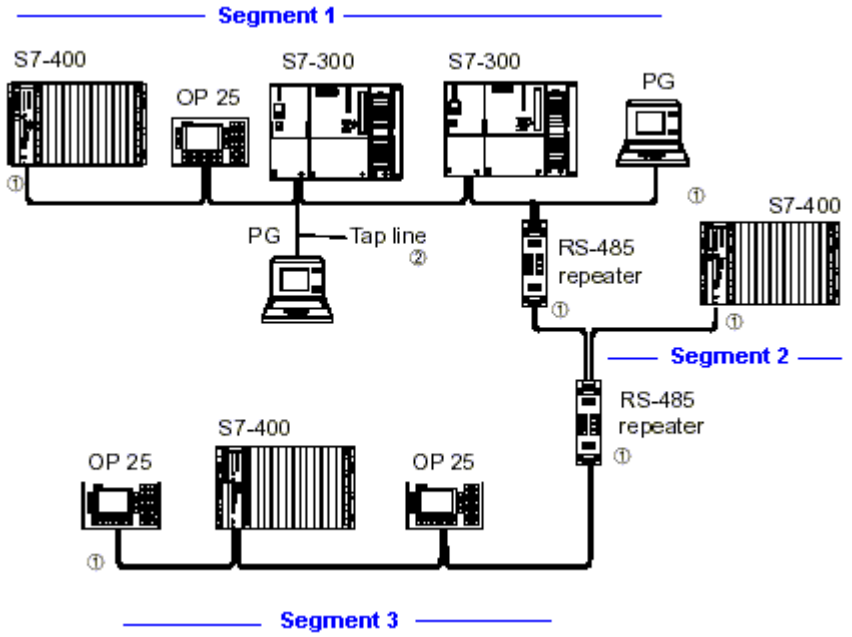
اگر طول کابل به حداکثر طول مجاز برسد باید از ریپیتر استفاده کنیم. در اینحالت سگمنت جدیدی تشکیل نمیشود بلکه همان سگمنت توسط ریپیتر ادامه می یابد مانند سگمنت ۲ در شکل روبرو. توجه شود که ترمینیتور برای سگمنت ۲ باید OFF باشد.



شکل روبرو نیز دو سگمنت را نشان میدهد که هر کدام بصورت باس بوده و توسط ریپیتر ادامه پیدا کرده اند.

توپولوژی درختی با استفاده از ریپیتر

ایجاد توپولوژی Tree فقط با استفاده از ریپیتر امکان پذیر است .شکل زیر ۳ سگمنت را نشان میدهد که از طریق دو ریپیتر ساختار درختی پیدا کرده اند. توجه شود در نقاطی که با علامت ① نشان داده شده ترمینیتور ON می باشد.



نکات مهم زیر را مجدداً یاد آوری میکنیم:

- هر سگمنت باس حداکثر ۳۲ Node میتواند داشته باشد.
- کل باس جمعاً ۱۲۷ Node میتواند داشته باشد.
- ماکزیمم ۹ عدد ریپیتر بصورت سری میتوان استفاده کرد.
- ریپیتر ها در محاسبه تعداد Node ها نیز باید منظور شوند.
- برای سرعت انتقال 1.5 Mbps فاصله بین هر دو Node متوالی نباید از حدی کمتر باشد. جزئیات این موضوع در صفحه بعد آمده است.

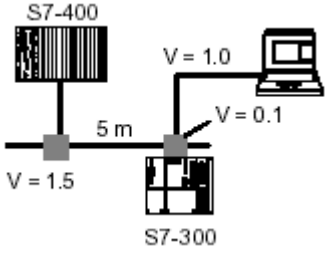
نکات خاص مربوط به سرعت انتقال 1.5 Mbps

در اتصال هر Node به کابل شبکه ،خازنی ایجاد میشود که در سرعتهای پایین تاثیری ندارد ولی برای سرعت 1.5 Mbps به بالا ممکن است مشکلاتی ایجاد شود . برای جلوگیری ازآن در طراحی سیستم باید نکات زیر منظور گردد .

- اگر فاصله بین دو Node مجاور از 10 متر بیشتر باشد شبکه مشکلی ندارد.
- اگر فاصله بین دو Node مجاور از 10 متر کمتر باشد باید مجموع ارزش اتصالات دو Node مجاور را از جدول زیر محاسبه کرد . سپس اگر رابطه زیر برقرار بود شبکه مشکل ندارد PG که توسط Tap Line با باس متصل است را نیز باید در محاسبات منظور کرد.

مجموع ارزش اتصالات دو Node مجاور > *طول کابل بین دو Node بر حسب متر*

- اگر شرایط فوق برقرار نبود باید تغییراتی در طول کابل لحاظ شود تا شبکه مشکل پیدا نکند.



مثال : در شکل روبرو فاصله بین دو ایستگاه مجاور 5 متر است با توجه به جدول و درنظر گرفتن PG مجموع ارزشها برابر است با : 1.5+0.1+1 =2.6

چون طول کابل 5 متر و بزرگتر از 2.6 است این شبکه در سرعت 1.5 Mbps مشکلی نخواهد داشت.

Product Name	Order number	Value (V)
Bus terminal with 1.5 m long tap line	(6GK1 500-0AA00)	1.5
Bus terminal with 1.5 m long tap line, with PG interface	(6GK1 500-0DA00	1.5
Bus terminal with 3.0 m long tap line	(6GK1 500-0BA00)	2.5
Bus connector with 30° cable outlet	(6ES7 972-0BA30-0XA0)	0.7
Bus connector with axial cable outlet Bus connector with axial cable outlet for FastConnect Bus connector with 90° cable outlet Bus connector with 90° cable outlet with PG interface Bus connector with 90° cable outlet for FastConnect Bus connector with 90° cable outlet with PG interface Bus connector with 35° cable outlet Bus connector with 35° cable outlet with PG interface	6GK1 500-0EA02) 6GK1 500-0FC00 6ES7 972-0BA11-0XA0 6ES7 972-0BB11-0XA0 6ES7 972-0BA50-0XA0 6ES7 972-0BB50-0XA0 6ES7 972-0BA40-0XA0) 6ES7 972-0BB40-0XA0	0.1
Bus terminal BT12M	6GK1500-0AA10	0.1
RS-485 repeater (attachment of bus segments)	6ES7 972-0AA01-0XA0)	0.1
PROFIBUS terminator (active RS-485 attachment element)	6ES7 972-0DA01-0AA0	0.1
SIMATIC S5/S7 connecting cable for 12 Mbps PG attachment to PROFIBUS-DP	6ES7 901-4BD00-0XA0	0.5

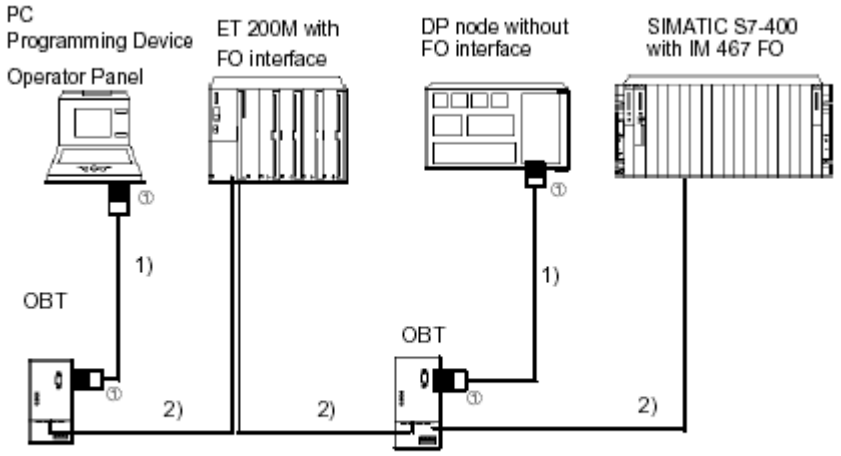
۳-۲ توپولوژی های شبکه نوری

با استفاده از المانهای شبکه فیبر نوری میتوان توپولوژی های مختلفی ایجاد کرد. با OBT می توان فقط توپولوژی باس و با OLM میتوان توپولوژی های باس و ستاره و رینگ را بوجود آورد. نکاتی که باید به آنها توجه داشت اینست که:

- با OBT نمیتوان فیبر شیشه ای استفاده کرد حداکثر فاصله بین دو نقطه با فیبر پلاستیکی 50m و با فیبر PCF 300 متر است.
- با OLM همه انواع کابل های نوری را میتوان استفاده کرد. فاصله بین دو نقطه با کابل شیشه ای 15 km و با کابل پلاستیکی 80 m است.

توپولوژی باس با استفاده از OBT

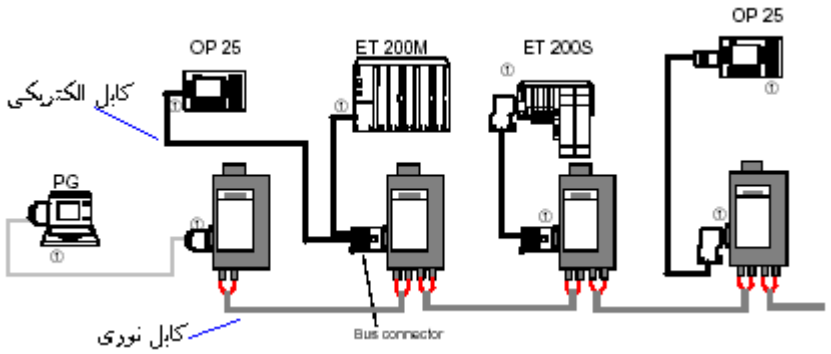
شکل زیر نمونه ای از توپولوژی باس که توسط OBT ایجاد شده را نشان میدهد.کابل نوع 1 الکتریکی و کابل نوع 2 نوری میباشد. توجه شود که در دو طرف کابل الکتریکی ترمینیتور ON می باشد.



① Terminating resistor activted
1) PROFIBUS cable (terminated at both ends)
2) Plastic FO cable or PCF FO cable with two fibers

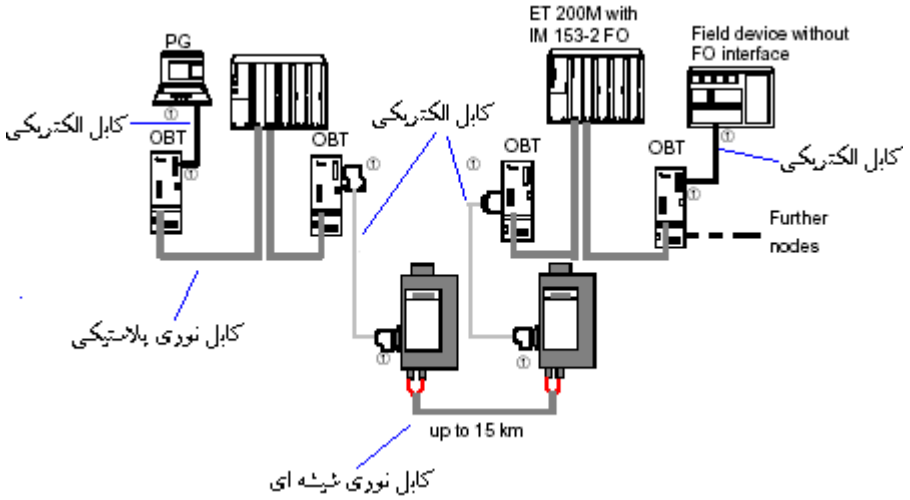
توپولوژی باس با استفاده از OLM

شکل زیر توپولوژی باس که توسط OLM ایجاد شده را نشان میدهد. در این روش میتوان بین دو OLM از فیبر شیشه ای استفاده کرد بنابر این مسافت بین دو Node بسیار بیشتر از نوع قبل (OBT) میباشد.



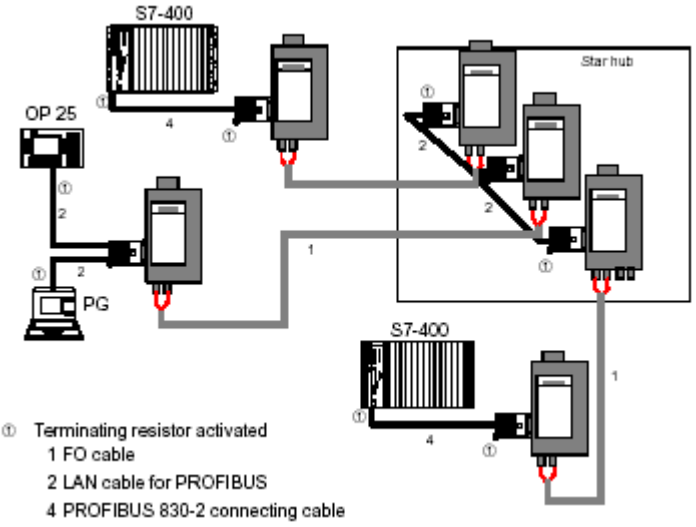
توپولوژی باس با استفاده از OLM و OBT

اگر فاصله برخی ایستگاهها بیشتر از حد مجاز استفاده OBT باشد میتوان بین آن نقاط فقط از OLM استفاده کرد. توجه شود که جایگزینی همه OBT ها با OLM ممکن است مقرون به صرفه نباشد و طراح ترجیح دهد آنها را بصورت ترکیبی استفاده کند.



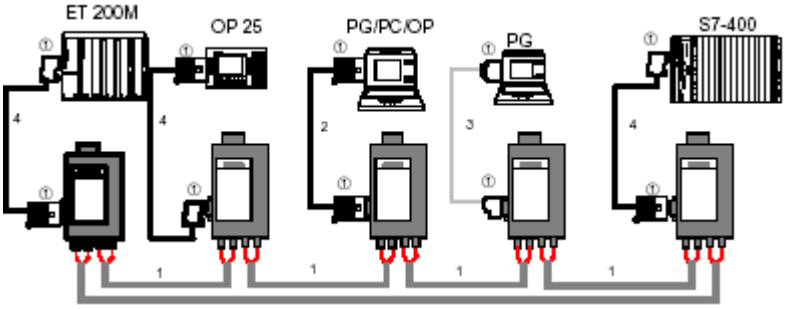
توپولوژی ستاره با استفاده از OLM

این توپولوژی فقط با OLM امکان پذیر است و نمیتوان از OBT استفاده کرد. آرایش OLM ها بگونه ای است که از ترکیب ۳ عدد از آنها یک Hub ایجاد میشود. بدیهی است در این روش با بروز مشکل روی هر ایستگاه بقیه شبکه میتواند بکار خود ادامه دهد. مگر اینکه اشکالی در Hub بوجود آید.

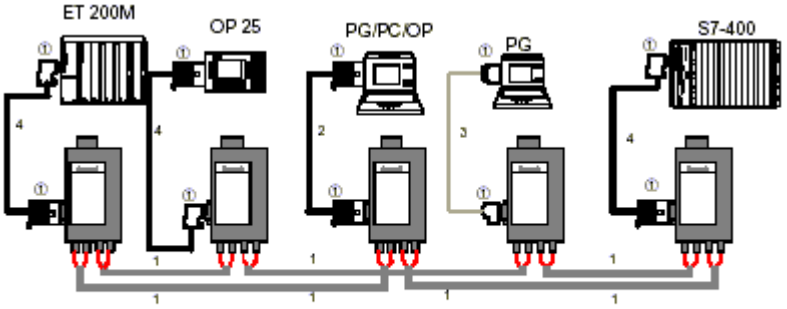


توپولوژی حلقوی با استفاده از OLM

این توپولوژی نیز فقط با OLM امکان پذیر است و نمیتوان از OBT استفاده کرد. با اتصال OLM ها روی شبکه نوری مانند شکل زیر توپولوژی Ring ایجاد میشود . این ساختار قابلیت اطمینانش نسبت به سایرین بیشتر است زیرا با بروز مشکل روی هر قسمت از فیبر یا هر کدام از OLM ها ، توپولوژی به حالت باس در می آید و شبکه به کار خود ادامه میدهد.



اگر فاصله بین دو OLM ابتدا و انتها در شکل فوق خیلی زیاد باشد ممکن است مشکل ایجاد شود زیرا هر نوع OLM تا فاصله خاصی را ساپورت می کند. (برخی تا ۴۰۰ متر و برخی تا ۳ کیلومتر و برخی دیگر تا ۱۵ کیلومتر) لذا میتوان ساختار را با شکل زیر ساده سازی کرد تا مشکل فوق برطرف شود.



① Terminating resistor activated

1 FO cable

2 LAN cable for PROFIBUS

3 PROFIBUS 830-1T connecting cable

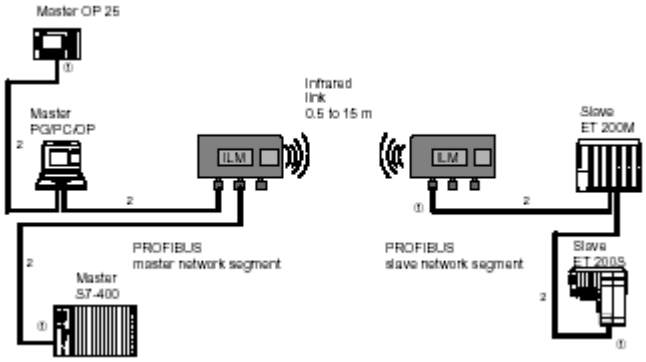
4 PROFIBUS 830-2 connecting cable

۳-۳ توپولوژی های شبکه بدون سیم

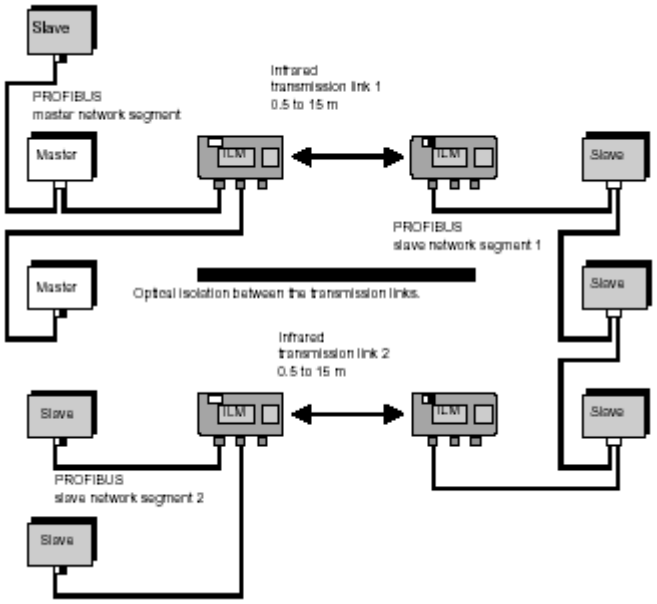
با استفاده از ILM میتوان شبکه PROFIBUS بدون سیم ایجاد کرد که البته ماکزیمم فاصله 15 متر و ماکزیمم سرعت 1.5 Mbps خواهد بود. صرفا دو نوع ساختار میتوان برای ILM داشت.

۱- Point to Point

در این روش Subnet مربوط به Slave ها با Subnet دیگر از طریق دو ILM به هم لینک میشوند.



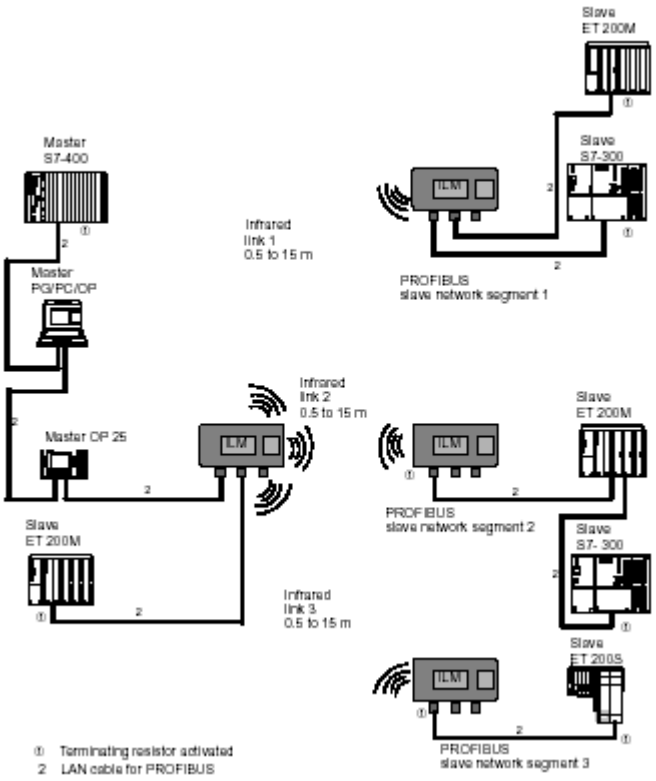
میتوان Subnet دوم را نیز از طریق ILM دیگر بصورت Cascade به Subnet سوم وصل کرد :



Point to multipoint -۲

در این روش Subnet مربوط به Slave ها با Subnet دیگر از طریق دو ILM به هم لینک میشوند.

توجه شود که چه در این روش و چه در روش قبل ، ترمینیتور هر کدام از Subnet ها باید در ابتدا و انتها فعال باشند.



فصل چهارم – پیکربندی شبکه PROFIBUS-DP در STEP7

مشتمل بر :

٤-١ بخش های مختلف یک پروژه STEP 7

٤-٢ ایجاد پروژه شامل شبکه PROFIBUS -DP

٤-٣ ایجاد Master با کارت CP

٤-٤ ایجاد Master با کارت IM

٤-٥ ایجاد Master با کارت IF

٤-٦ استفاده از چند Master System بصورت ترکیبی

٤-٧ استفاده از سایر DP Slave ها

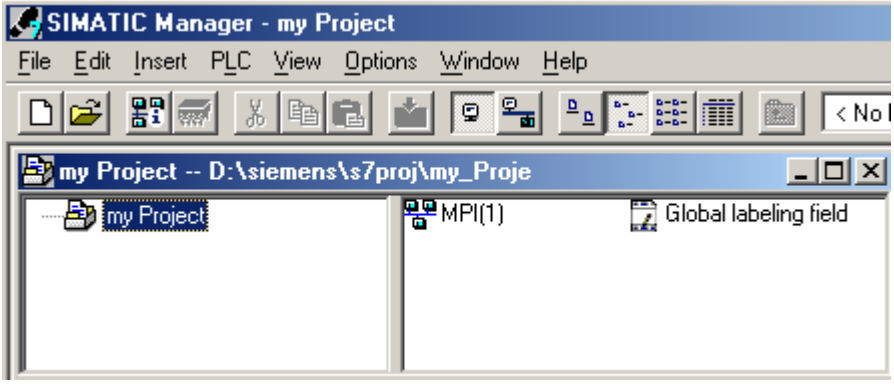
مقدمه

در این بخش ابزارهایی که در STEP 7 برای پیکر بندی وراه اندازی شبکه PROFIBUS نیاز است را معرفی خواهیم کرد. اگرچه آشنایی نسبی خواننده با محیط STEP7 پیش فرض ما در تدوین این کتاب بوده است با این وجود بمنظور یاد آوری به اجمال نگاهی به محیط این برنامه می اندازیم. اگر این مطالب مقدماتی برای خواننده تکراری است میتواند ادامه بحث شبکه را از قسمت ۵-۳ دنبال کند.

همانطور که میدانیم نرم افزار STEP 7 مشتمل بر چندین قسمت کاربردی است که هرکدام برای پوشش دادن یکی از نیازهایی که در طراحی یا پیاده سازی یا عیب یابی سیستم اتوماسیون مطرح میشود استفاده می گردد.

پیکر بندی سخت افزار ، برنامه نویسی و پیکر بندی شبکه و ارتباطات از جمله این نیازها به شمار میروند.

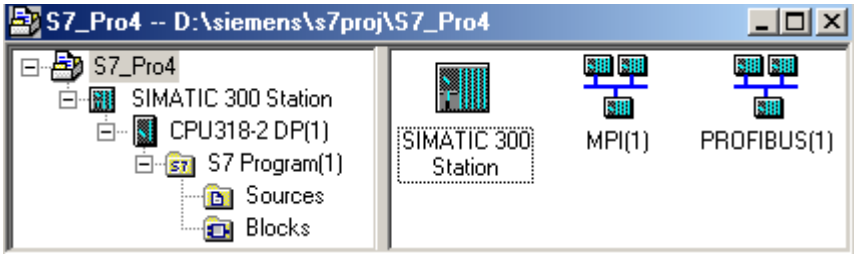
برنامه کاربردی که این مجموعه ابزار ها را به صورت یکجا فراهم نموده است SIMATIC MANAGER نامیده می شود.SIMATIC MANAGER همه اطلاعات مورد نیاز وتنظیمات انجام شده را ذخیره و در داخل یک پروژه گرد آوری می کند . بنابر این اولین قدمی که باید در طراحی یک سیستم جدید برداشته شود ایجاد یک پروژه جدید در Simatic Manager است.



٤-١ بخش های مختلف یک پروژه STEP 7

یک پروژه STEP 7 از یک یا چند Station تشکیل شده است.منظور از Station سخت افزار سیستم است که میتواند شامل اجزایی مانند Rack و CPU و کارت های ورودی و خروجی باشد.

یک Station که از قبل پیکر بندی شده باشد در Simatic Manager دارای ساختار سلسله مراتبی مانند شکل زیر میباشد. یک Folder مخصوص Hardware و یک Folder مخصوص برنامه STEP7 (به نام S7 Program) مجموعه قسمت های اصلی یک Station را تشکیل می دهند



نرم افزار STEP 7 یک نرم افزار شیء گراست .کافی است برای کار با هر یک از اجزاء واشیای موجود در SIMATIC MANAGER برروی آن دابل کلیک نمایید.در این صورت اطلاعات وموارد لازم ومرتبط به آن شی با گرافیک مناسب در اختیار کاربر قرار خواهد گرفت. به جای دابل کلیک بر روی یک شی می توان راست کلیک نموده وسپس Open کنیم.

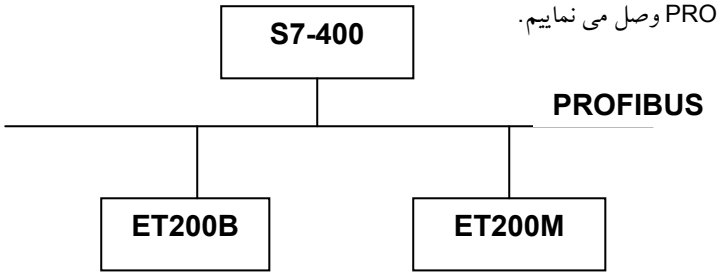
توجه داشته باشید که :

- اطلاعات برنامه نویسی در پوشه Blocks ذخیره میشود با کلیک کردن روی بلاک مورد نظر زیر برنامه ای به نام LAD/STL/FBD اجرا میگردد که توسط آن میتوان برنامه نویسی انجام داد.
- با کلیک کردن روی آیکون SIMATIC Station آیکون Hardware ظاهر میشود که با کلیک روی آن زیر برنامه ای به نام Hwconfig برای پیکر بندی سخت افزار باز می گردد.
- با کلیک کردن روی هر کدام از آیکونهای MPI یا PROFIBUS زیر برنامه ای به نام Netpro که مربوط به پیکر بندی شبکه است باز میشود.

٤-٢ ایجاد پروژه شامل شبکهPROFIBUS –DP

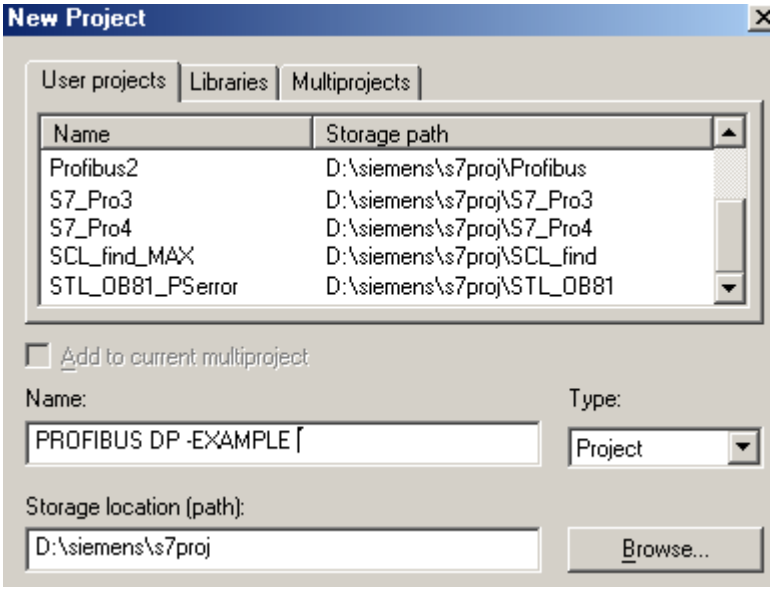
برای اینکه هم نحوه ایجاد یک پروژه در STEP 7 را یاد آوری کرده باشیم وهم ایجاد یک شبکه PROFIBUS را شروع به فراگیری نماییم ،مثال زیر را طرح ودنبال می نماییم.

می خواهیم یک پروژه Station S7-400 ایجاد نموده ودر آن از یک CPU 416-2DP استفاده کنیم.در ادامه برای پیکر بندی یک شبکه Master/Slave یک ET 200B شامل 16DI و 16DO ویک ET 200M به شبکه PROFIBUS وصل می نماییم.

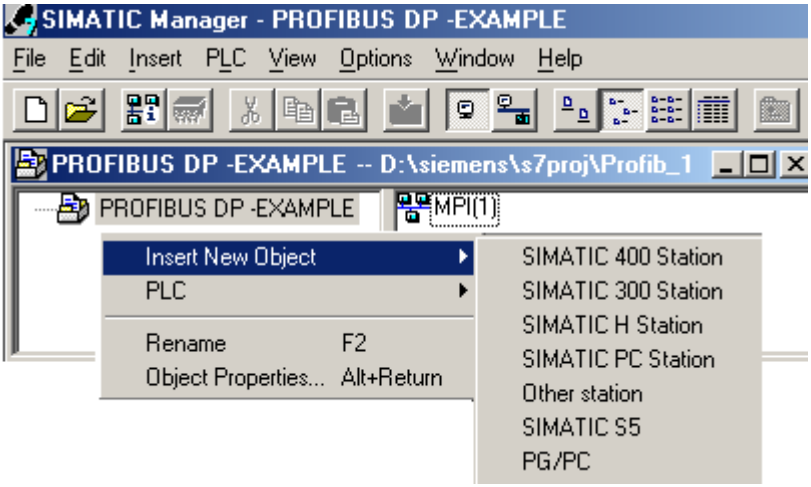


گام اول: ایجاد یک پروژه جدید

برنامه SIMATIC MANAGER را باز کنید سپس در برنامه SIMATIC MANAGER از قسمت File گزینه New... را انتخاب کنید .یک پنجره مطابق شکل باز می شود .در این پنجره، نام دلخواه مثلاً- PROFIBUS DP EXAMPLE را انتخاب کنید .در قسمت Type هم گزینه Project را انتخاب کنید . ودر قسمت Storage Location مسیری را که می خواهید پروژه در آنجا Save شود، تایپ یا آنرا بدون تغییر تاییدکنید.

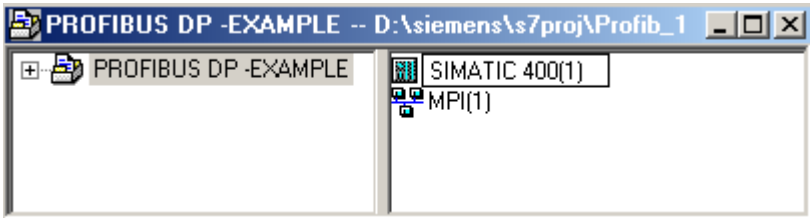


حال صفحه اصلی SIMATIC MANAGER به صورت شکل زیردر خواهد آمد که شامل پروژه PROFIBUS DP -EXAMPLE است. در شرایط فعلی در این پروژه فقط یک شی به نام MPI وجود دارد که شما آن را در نیم صفحه سمت راست می بینید .ارتباط MPI به صورت پیش فرض توسط STEP 7 برای تمام پروژه ها ساخته می شود .شبکه MPI یکی از شبکه های زیمنس است که معمولاً برای اتصال PLC به PG یا PC از آن استفاده می شود،هر چند که برای شبکه های با طول کوتاه وسرعت پایین هم می توان از آن برای ارتباط بین PLC ها استفاده کرد .شبکه MPI موضوع بحث ما نیست.



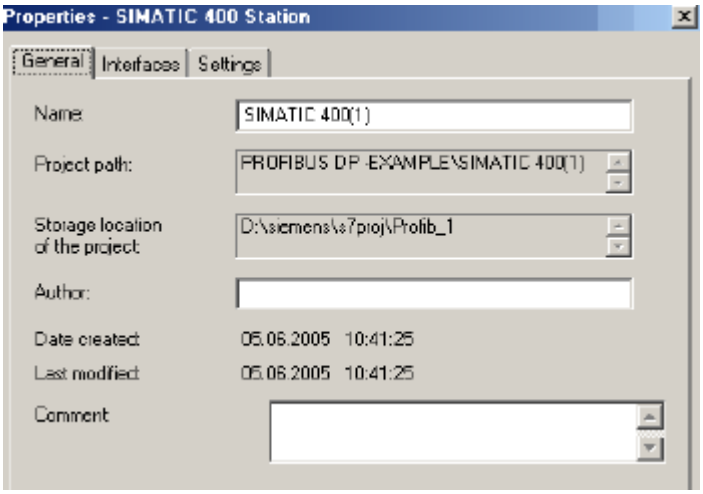
گام دوم :ایجاد یک Station S7-400 در پروژه

بر روی نام پروژه در پنجره سمت چپ صفحه SIMATIC MANAGER راست کلیک نمایید،گزینه Insert New Object را انتخاب وسپس یک Station از نوع SIMATIC 400 ایجاد نمایید.



این Station جدید که ایجاد کردید به عنوان یک شی در پنجره سمت راست ظاهر می شود.می توانید در صورت لزوم ،نام این Station را عوض کنید ویک نام متناسب با عملکرد این Station مثلاً Process1 را انتخاب کنید. برای این کار کافی است که بر روی نام Station کلیک راست کرده وگزینه Rename را انتخاب نمایید.

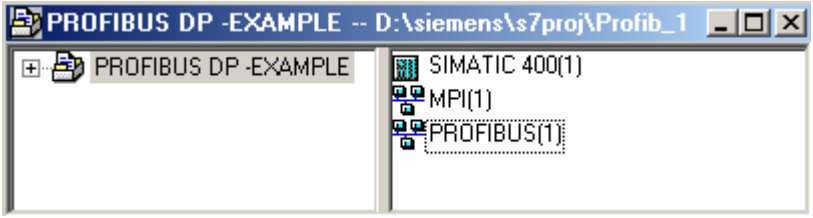
همچنین با راست کلیک بر روی نام این Station وانتخاب Properties یک پنجره باز می شود. همانطور که در شکل هم می بینید در این پنجره شما می توانید یک سری توضیحات شامل نام Station ، Author و...رابنویسید.



گام سوم: ایجاد شبکه PROFIBUS

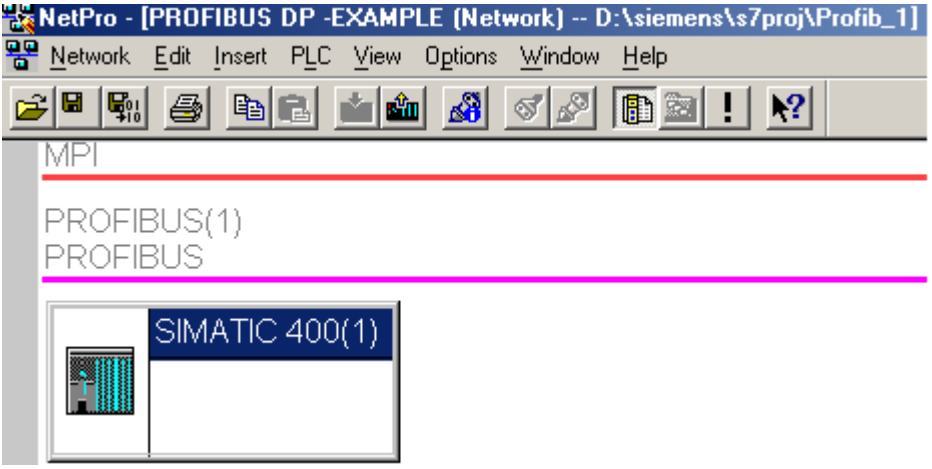
تا اینجا در این پروژه یک Station 400 ایجاد کرده ایم.حال می خواهیم یک شبکه PROFIBUS هم اضافه کنیم راههای مختلفی برای ایجاد شبکه PROFIBUS وجود دارد.در اینجا یک روش برای شروع گفته می شود.روش های بعدی به تدریج ذکر خواهند شد.

شبیه نحوه ایجاد کردن یک Station ،برای ایجاد شبکه PROFIBUS هم می توان در قسمت سمت چپ صفحه SIMATIC MANAGER بر روی نام پروژه راست کلیک کرده ودر قسمت Insert New Object گزینه PROFIBUS را انتخاب کنید. می توانید برای این شبکه PROFIBUS یک نام هم انتخاب کنید وگرنه به صورت خودکار نام(1) PROFIBUS برای آن انتخاب می شود.



برای پیکر بندی شبکه PROFIBUS یک ابزار به نام NetPro وجود دارد.برای مشاهده مجموعه کارهایی که در گام های یک تا سه انجام داده اید می توانید این برنامه را باز نموده وگرافیک کلی کار را ببینید.برای باز کردن این برنامه یا می توانید از منوی Option گزینه Configure Network را انتخاب کنید یا بر روی کلید مربوط به این

گزینه در قسمت نوار ابزار بالای صفحه که به شکل است کلیک نمایید .برنامه Netpro باز می شود.

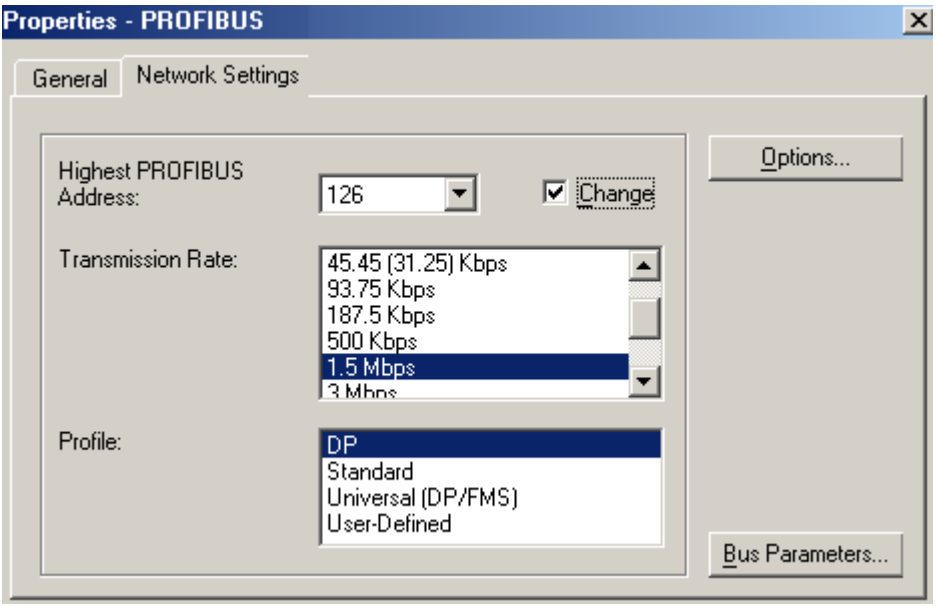


همانطور که در شکل می بینید تا اینجا یک Station 400 به نام SIMATIC 400(1) ویک شبکه PROFIBUS به وجود آمده است.شبکه MPI هم که به صورت اتوماتیک توسط خود STEP 7 ایجاد شده است.Station400 هنوز دارای هیچ سخت افزاری (CPU,I/O,...) نیست وبه هیچ کدام از شبکه های MPI وPROFIBUS وصل نشده است

گام چهارم : تنظیمات شبکه PROFIBUS

برای انجام تنظیمات شبکه PROFIBUS ،در برنامه Netpro روی خط بنفش مربوط به PROFIBUS راست کلیک کنید وگزینه Object Properties را انتخاب کنید .پنجره Network Setting را باز کنید.(شکل صفحه بعد) همچنین برای باز کردن این پنجره می توانید در صفحه اصلی SIMATIC MANAGER ،بر روی شی مربوط به PROFIBUS راست کلیک نمایید.

در پنجره Network Setting فعلاً با فشردن کلید OK ، کلیه تنظیمات پیش فرض را تأیید کنید .در ادامه پارامتر های شبکه PROFIBUS که در این پنجره دیده شدند را بررسی می کنیم. چنانچه برای ادامه کار عجله دارید، می توانیدبدون مطالعه مطالب بعدی، از قسمت ۵-۲-۳ بحث را دنبال کنید.



پارامتر Highest PROFIBUS Address

در استاندارد EN 50 170 این پارامتر موسوم به HSA (Highest- Station Address) می باشد این پارامتر برای بهینه سازی نحوه در اختیار قرار گرفتن باس ومدیریت کردن حلقه Token ،زمانی که چندینMaster به شبکه PROFIBUS وصل هستند وبه کار می رود.این پیکر بندی را در آینده مورد بررسی قرار خواهیم داد. فعلاً در این مثال یک Master بیشتر نداریم ولذا،مقدار پیش فرض یعنی ۱۲٦ را تغییر ندهید.

پارامتر Transmission

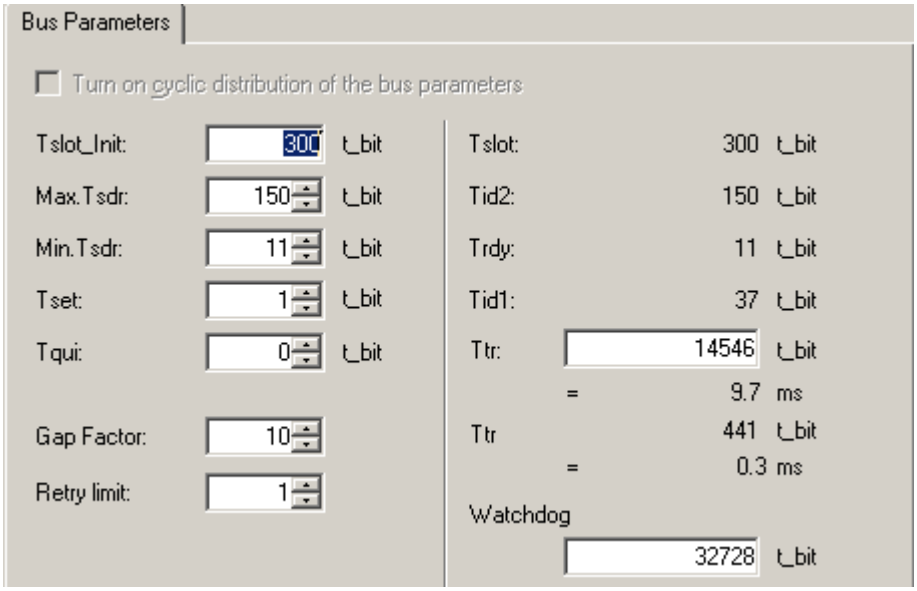
سرعتی که در اینجا برای انتقال اطلاعات انتخاب می کنید برای سرتاسر طول شبکه وکلیه node های متصل به آن اعمال می شود.در نتیجه کلیه تجهیزات وDistributed I/O هایی که به این شبکه وصل می شوند باید روی همین سرعت تنظیم شوند.سرعت انتقال اطلاعات می تواند بین 9.6kbps تا 12Mbps تغییر کند.همانطور که در شکل می بینید سرعت انتقال به صورت پیش فرض 1.5Mbps انتخاب شده است.

Profile های PROFIBUS

برای کاربرد های مختلف شبکه PROFIBUS یک سری پروفایل های استاندارد وجود دارد.در واقع این پروفایل هامجموعه ای از Setting ها هستند که بر روی پارامتر های باس شبکه PROFIBUS انجام می شوند. این پارامتر ها توسط STEP7 محاسبه وSet می شوند وکافی است ما Profile مورد نظر را انتخاب کنیم .بدیهی است که مانند نرخ ارسال ،این پارامتر ها به کل مجموعه شبکه وتجهیزات متصل به آن اعمال می شوند.

در حالتی که نخواهیم از پروفایل های موجود در STEP7 استفاده نماییم میتوانیم گزینه User Defined را انتخاب نموده وسپس متناسب با کاربرد مورد نظرمان پارامتر های باس را تنظیم کنیم.در غیر این صورت می بایست از یکی از پروفایل های Universal(DP/FMS),Standard,DP استفاده نماییم.

توجه اینکه ،تنظیم دستی پارامترهای باس،نیازمند تجربه است و تنها یک مهندس شبکه حرفه ای می تواند از پس آن برآید وترجیح برآن است که از یکی از Profile های موجود استفاده شود.مانند شکل زیر در پنجره Network Setting اگر حالت User Defined را انتخاب نموده وسپس کلید Bus Parameters را فشار دهید یک پنجره جدید مربوط به پارامتر های شبکه PROFIBUS باز می شوداین پارامترهاکه در ضمیمه ۳ شرح مختصری راجع به آنها داده شده است فقط برای پروفایل User Defined قابل تغییر هستند.



در ادامه به بررسی مختصر پروفایل های موجود در STEP7 می پردازیم.

DP Profile

زمانی این پروفایل را انتخاب کنید که سیستم مورد نظر ،فقط شامل شبکهPROFIBUS DP و دارای ساختار Master-Slave باشد که ازیک یا چند Master تشکیل شده و Master ها همگی از خانواده SIMATIC S7 و SIMATIC M7 انتخاب شده باشند. با تغییر در node ها واضافه شدن Subnet های جدید، STEP7 به صورت خودکار پارامترهای باس را با در نظر گرفتن تغییرات جدید،محاسبه می کند.Subnet های جدید می توانند شامل

موارد زیر باشند:

- PC یاPG
- واحد HMI
- S7 node
- ارتباط FDL یا FMS که به صورت non-cyclic پیکر بندی شده باشند.

توجه اینکه تنها آن دسته از node های متصل به شبکه PROFIBUS در محاسبه پارامتری باس لحاظ خواهند شد که به درستی توسط STEP7 پیکر بندی شده باشند.

StandardProfile

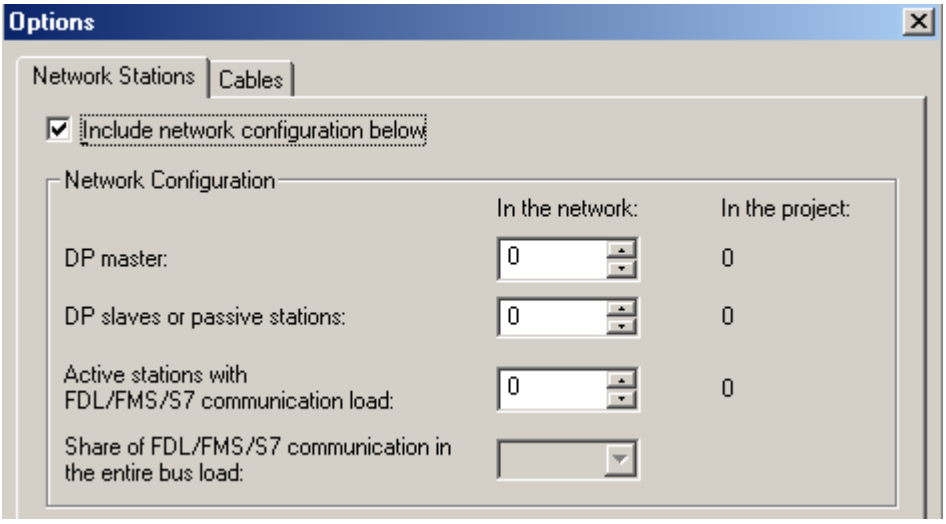
زمانی از این پروفایل استفاده می کنیم که بخواهیم node هایی را به شبکه PROFIBUS متصل کنیم.که توسط STEP7 قابل پیکر بندی نباشند ویا متعلق به پروژه جاری نبوده وبه پروژه دیگری تعلق داشته باشند .در این حالت در پنجره Network Setting در قسمت Profile گزینه Standard را انتخاب نمایید. سپس کلید ..Options را فشار دهید تا پنجره Options باز شود در این پنجره، Network Stations را انتخاب نموده وگزینه Include Network Configuration Below را فعال کنید .در غیر این صورت نحوه محاسبه پارامتر های باس همانند همان الگوریتم محاسبه پارامتر های باس در حالت DP خواهد بود وnode های پیکر بندی نشده در پروژه جاری را در نظر نخواهد گرفت .اما اگر این گزینه فعال شود یک الگوریتم عمومی را به کار خواهد برد که در نتیجه node های پیکر بندی نشده در پروژه جاری را هم در محاسبات پارامتر های باس لحاظ خواهد کرد.

Universal (DP/FMS) Profile

زمانی از این پروفایل استفاده می کنیم که در مجموعه تجهیزات متصل به شبکه PROFIBUS از تجهیزات خانواده SIMATIC S5 مثلاً CP 5431 یا CPU S5-95U استفاده شده باشد.یا زمانی که Station های SIMATIC S7 و SIMATIC S5 به صورت همزمان در یک پروژه به عنوان دو node در یک شبکه PROFIBUS به کار رفته باشند.

پنجره Options

در پنجره Network Setting اگر کلید Options را فشار دهید یک پنجره جدید باز می شود که شامل دوقسمت یکی Network Station ودیگری Cables می باشد.



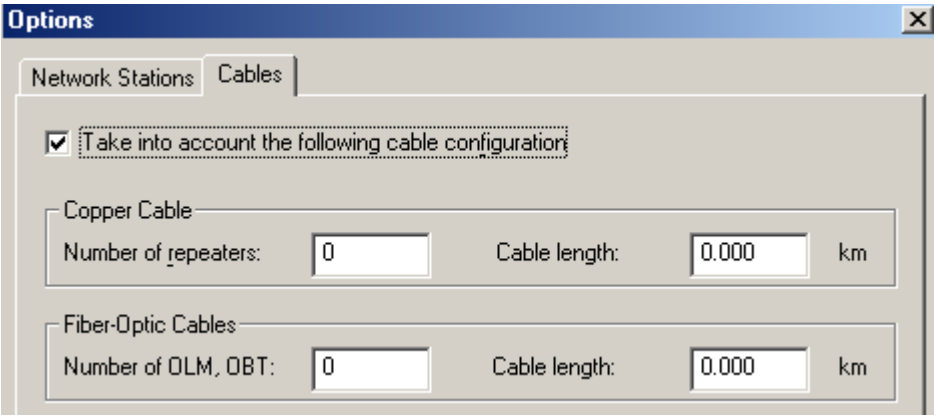
Network Station

قبلاً گفتیم که در بعضی کاربرد ها ،ممکن است node هایی به کار بریم که پیکر بندی آنها توسط STEP7 ممکن نباشد.مثلاً یک S5 را به عنوان node شبکه استفاده کرده باشیم .این موضوع را قبلاً در پروفایل Standard هم بحث کردیم و گفتیم که برای اینکه در محاسبات پارامتر های باس، این node را هم لحاظ کرده باشیم باید گزینه Include Network Configuration Belowرا فعال نماییم.

همان طور که در شکل می بینید در زیر ستون In The Project لیست تعداد node هایی که توسط STEP7 ،پیکر بندی شده اند نشان داده شده است در قسمت سمت چپ در زیر ستون In The Network لیست تعداد node هایی که در محاسبات پارامتر های باس باید لحاظ شوند آمده است که به این ترتیب با وارد کردن مقادیر مناسب ، می توانیم به پارامتر های باس مناسب دست یابیم .زیرا حالا دیگر پارامترهایی که توسط STEP7 محاسبه می شوند با در نظر گرفتن همه node های شبکه اعم از S7 وغیر S7 صورت می گیرد.این صفحه در حالتی که پروفایل DP را انتخاب کرده اید،غیر فعال است.

Cables

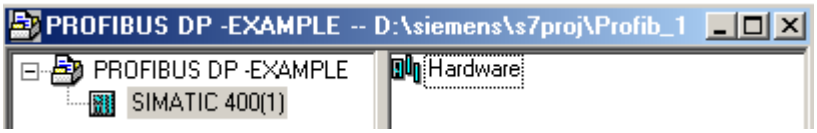
طول کابل ها، تعداد Repeater ها در حالت کابل مسی وتعداد OLM ها در حالت فیبر نوری موارد دیگری هستند که در محاسبه پارامتر های باس مؤثر هستند .در پنجره Option با انتخاب Cables ،مطابق شکل ۱۴-۳ پنجره ای ظاهر می شود که شامل دو قسمت است .ابتدا با انتخاب گزینه Take into account the following cable configuration ،به STEP7 می گوییم که مقادیر مربوط به کابل های شبکه را در محاسبات پارامترهای باس در نظر بگیرید سپس در قسمت پایین این صفحه که مشتمل بر دوقسمت است(یکی مربوط به کابل مسی ودیگری مربوط به کابل نوری) مقادیر مناسب را وارد می کنیم.

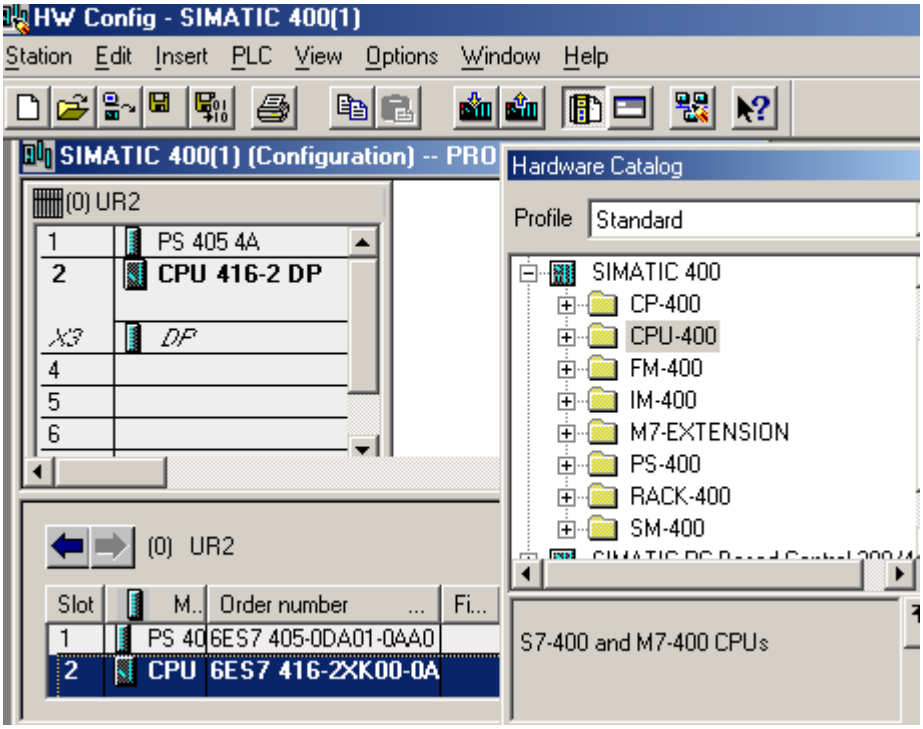


گام پنجم: پیکر بندی سخت افزار سیستم با برنامه HW Config

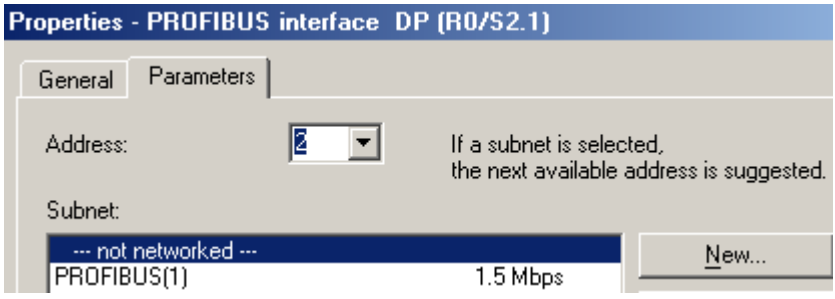
در این مبحث تا حدی که مورد نیاز شبکه PROFIBUS است به موضوع پیکر بندی سخت افزار توسط HW Config می پردازیم. نکته ای که در اینجا باید مطرح شود این است که برنامه HW Config و برنامه Net Pro از دو جنبه مختلف به یک سیستم نظارت می کنند بنابراین وقتی پیکربندی انجام شده در یکی را میخواهیم توسط دیگری باز کنیم قبل از آن حتما باید عمل ذخیره سازی را انجام داده باشیم.

در مثال ذکر شده ، با ذخیره سازی آنچه که در NetPro داریم روی آیکون SIMATIC 400(1) در پنجره NetPro کلیک کرده می بینیم که توسط برنامه HWConfig باز میشود. این کار از روش های دیگر نیز امکان پذیر است. از جمله در Simatic Manager با کلیک کردن روی آیکون Hardware مربوط به Station

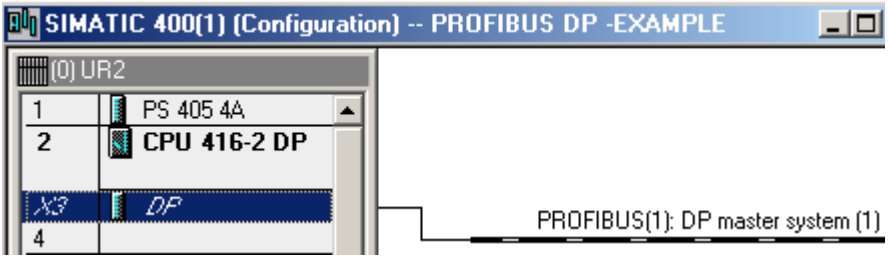




در Hwconfig با استفاده از کاتالوگ پنجره سمت راست از زیر مجموعه SIMATIC 400 ابتدا Rack مورد نظر (مثلا UR2) راانتخاب کرده سپس منبع تغذیه و پس از آن CPU را انتخاب کرده و بترتیب آنها را از اسلات اول به بعد قرار میدهیم. CPU انتخاب شده در مثال فوق CPU 416-2 DP با کد سفارش 6E S7 416-2XK00-0AB0 میباشد. چون این CPU از نوع 2-DP است یک پورت مخصوص ارتباط PROFIBUS DP دارد. وقتی که این CPU را انتخاب می کنید یک پنجره باز و از شما پرسیده می شود که پورت 2-DP مربوط به این CPU را می خواهید به کدام شبکه PROFIBUS وصل نمایید.

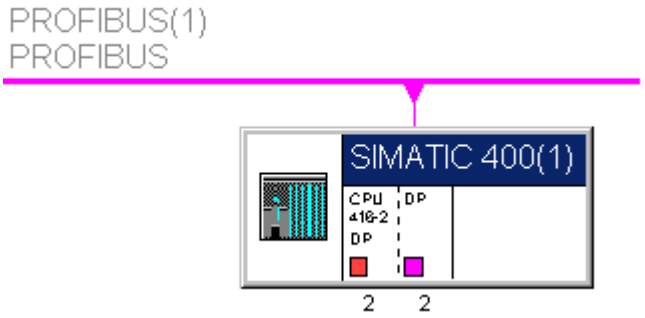


در این پنجره می توانیم یک شبکه DP جدید ایجاد کنیم که با زدن کلید New یک شبکهPROFIBUS جدید ایجاد نموده وهمان مراحلی را که قبلاً برای تعیین پارامترهای باس ذکر شد را باید طی کنیم.ولی در اینجا منظورمان این است که این CPU را به همان شبکه PROFIBUS که قبلاً ایجاد کرده بودیم وصل کنیم بنابر این شبکه PROFIBUS(1) را انتخاب می نماییم.آدرس ۲ برای آدرس Station روی شبکه PROFIBUS بصورت پیش فرض داده شده و لزومی ندارد آنرا تغییر دهیم.



پس از طی این مراحل یک شبکهPROFIBUS که یک ایستگاه Master به آن وصل است ایجاد کرده ایم.در مراحل بعدی تعدادی Slave هم به شبکه اضافه می کنیم. برای غیرفعال کردن یا فعال کردن مجدد شبکه فوق کافیست روی DP در اسلات X3 راست کلیک کنیم و گزینه Disconnect یا Add Master Sytem را انتخاب نماییم.

پس از ذخیره سازی پیکر بندی Hwconfig در برنامه Net Pro نیز می توان،وضعیت موجود را مشاهده کرد.

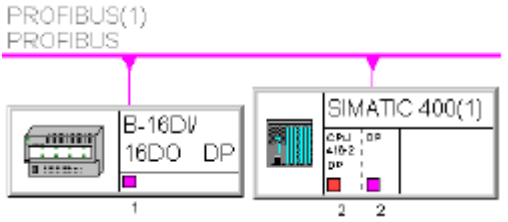


گام ششم:پیکر بندی DP Slave ها

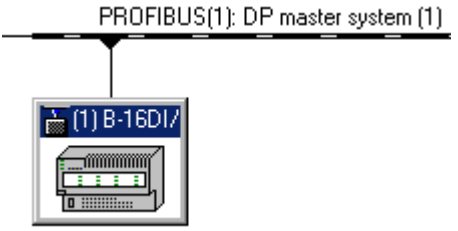
الف:اتصال ET 200B به شبکه PROFIBUS

در این مرحله باید DP Slave ها را به شبکه DP وصل کنیم. ابتدا می خواهیم یک ET 200B به شبکه PROFIBUSوصل کنیم. اینکار به دو روش امکان پذیر است. روش اول در NetPro ابتدا روی پورت DP مربوط به SIMATIC 400 کلیک کرده سپس از پنجره کاتالوگ که در سمت راست برنامه موجود است از زیر مجموعه PROFIBUS DP و از خانواده ET200B روی B-16DI/16DO دابل کلیک میکنیم. مشاهده میکنیم

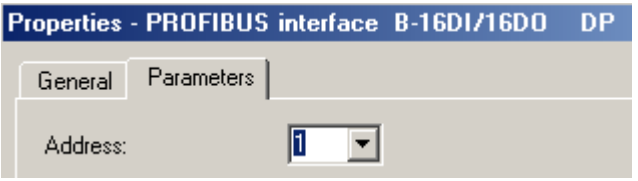
که این المان روی شبکه قرار می گیرد. پنجره کاتالوگ در شکل روبرو نمایش داده شده است.با وارد کردن ET200B مورد نظر در NetPro شکلی شبیه زیر خواهیم داشت. اکنون اگر پس از ذخیره سازی روی المان مربوط به ET200B کلیک کنیم می بینیم که ساختار در محیط Hwconfig نمایش داده خواهد شد.



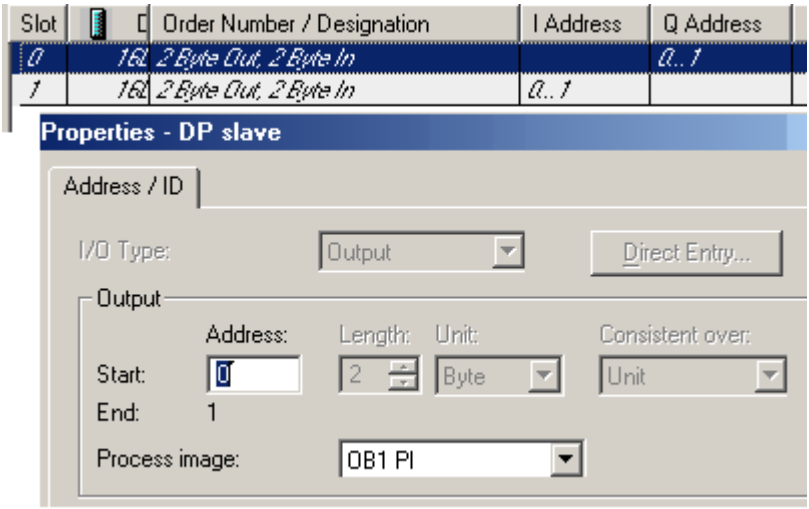
روش دوم آنست که در HWConfig از پنجره کاتالوگ که شبیه پنجره کاتالوگ NetPro میباشد ET200B را انتخاب و روی شبکه PROFIBUS متصل به Master قرار دهیم. اگر پس از ذخیره سازی در HWConfig به Netpro برگردیم شکل بالا را مشاهده خواهیم نمود. لازم به ذکر است در Net pro برای مشاهده شبکه با یا بدون Slave ها از منوی View این برنامه و انتخاب With DP Slaves استفاده مینماییم.



نکته قابل ذکر دیگر آدرس Node است. آدرس ها بصورت اتوماتیک توسط نرم افزار تعیین میشوند. همانطور که در شکل های فوق ملاحظه میشود آدرس 1 به ET200B اختصاص داده شده است. Master نیز بصورت پیش فرض دارای آدرس 2 میباشد. کاربر میتواند در صورت لزوم این آدر سها را در قسمت Properties تغییر دهد.



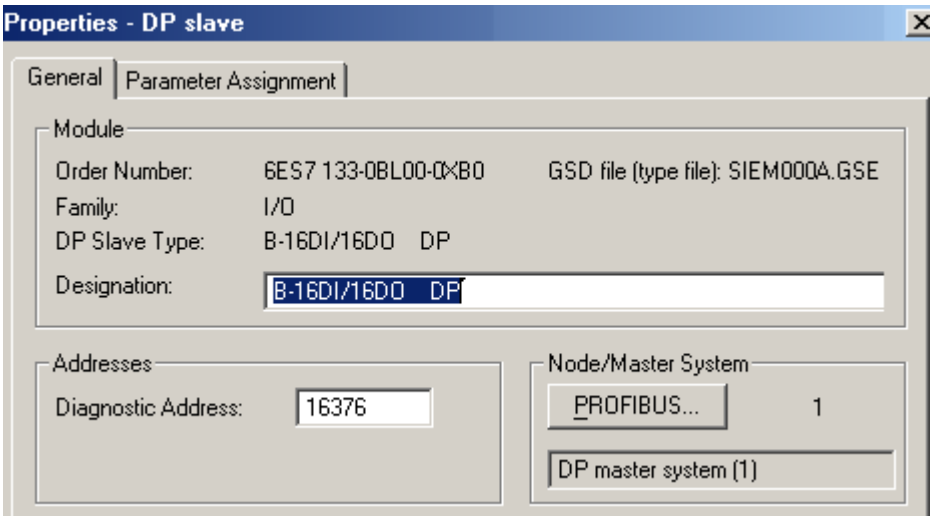
در Hwconfig جزییات بیشتر مربوط به ورودی و خروجیهای ET 200B ظاهر میشود .آدرس آنها در قسمت پایین صفحه HW Config یعنی در Configuration Table دیده می شود. برای تغییر این آدرس ها کافیست در این جدول بر روی کارت مربوطه دابل کلیک نمایید.پنجره Properties DP Slave باز می شود که در قسمت Address I/O می توانید Start Address را تغییر دهید. در اینجا آدرس ورودی ها از I0.0 تا I1.7 و آدرس خروجی ها از Q0.0 تا Q1.7 میباشد. باید توجه داشت که آدرسهای مربوط به ورودی و خروجی ET ها نباید با آدرس کارتهای ورودی و خروجی که روی رک اصلی یا رک اضافی نصب شده اند تلاقی پیدا کند. آدرسهایی که خود سیستم تعیین میکند تلاقی ندارند ولی در صورتی که کاربر بخواهد آنها را تغییر دهد باید توجه به موضوع فوق داشته باشد.



اگر در صفحه اصلی HW Config بر روی ET 200B دابل کلیک نمایید.پنجره Properties DP Slave باز می شود (شکل صفحه بعد) در این صفحه یک سری اطلاعات مانند Order Number ، مشخصات ET 200B شامل نوع وتعداد I/O ها ویک سری مشخصات مهم دیگر را می بینید که در ادامه در مورد آنها بحث می کنیم.

Diagnostic Address

چنانچه این DP Slave مشکلی پیدا کند به CPU اطلاع می دهد واین آدرس را به CPU بر می گرداند.به این ترتیب این آدرس مشخص کننده این است که کدام DP Slave مشکل پیدا کرده است.در این حالت OB 86 صدا زده می شود وآدرسDiagnostic این DP Slave (در این مثال 16376) به OB 86 فرستاده می شود وما می توانیم با برنامه ریزی مناسب این OB، متوجه شویم که کدام DP Slave (در صورت وجود چندین DP Slave) وبه چه دلیل مشکل پیدا کرده است.



Node/Master System

در این قسمت یک سری اطلاعات راجع به شبکه PROFIBUS وپارامتر های آن وآدرس شبکه این DP Slave داده شده است. اگر کاربر بخواهد آدرس Node را عوض کند یا سرعت دیگری بجز 1.5 Mbps انتخاب کند با کلیک کردن روی کلید PROFIBUS میتواند در پنجره بعدی اینکار را انجام دهد.

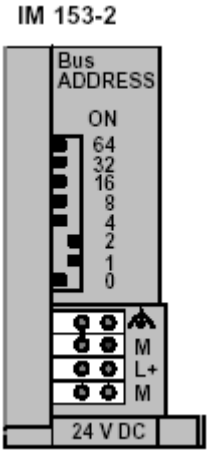
Parameter Assignment

اگر در پنجره Properties به جایGeneral, Parameter Assignment را انتخاب کنید،یک سری پارامترهای مربوط به این DP Slave را می بینید که نحوه تنظیم آنها در کاتالوگ های مربوط به هر DP Slave آمده است.خوشبختانه ، برای DP Slave های متعلق به خانواده SIMATIC S7 نیازی نیست که این پارامترها را تنظیم کنید واین پارامترها توسط HW Confing به طور اتوماتیک تنظیم می شوند.

تا اینجا تنظیمات سخت افزاری که باید در شبکه Master / Slave برای ET200B انجام شود توضیح داده شد. اما یک تنظیم سخت افزاری نیز روی خود ET200B لازم است انجام شود. آدرسی که روی شبکه برای آن منظور شده و در این مثال 1 میباشد لازم است توسط Dip Switch های روی خود ET نیز تنظیم گردد.

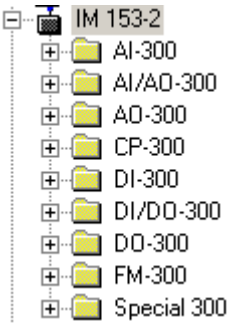
ب :اتصال ET 200M به شبکه PROFIBUS

برای تکمیل بحث ،مطابق با مثال ذکر شده در ابتدای این بخش ،یک ET 200M نیز به شبکه متصل می کنیم.همانطور که می دانید ET 200M برخلاف ET200B مدولار است و می توان تعدادی کارت I/O به آن اضافه کرد.

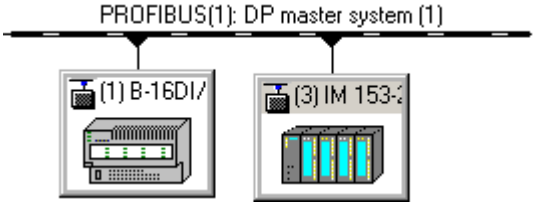


به همان شیوه که ET 200B را به شبکه PROFIBUS وصل کردیم در اینجا نیز ET 200M را با انتخاب نوع IM 153-2 به شبکه متصل میکنیم ودر پنجره ای که باز می شود آدرس شبکه این DP Slave را 3 قرار میدهیم. این تنظیم همانطور که برای ET200B نیز گفته شد روی خود سخت افزار نیز باید انجام شود. روی IM تعدای DIP Switch وجود دارد که در کنار آنها اعدادی نوشته شده است. آدرس مورد نظر را باید با ترکیب اعداد فوق بسازیم و سوئیچ های مربوط به آن اعداد را در وضعیت ON قرار دهیم. در اینجا برای ساختن آدرس 3 باید از 2+1 استفاده کنیم پس همانند شکل سوئیچ های 1 و 2 را ON میکنیم.

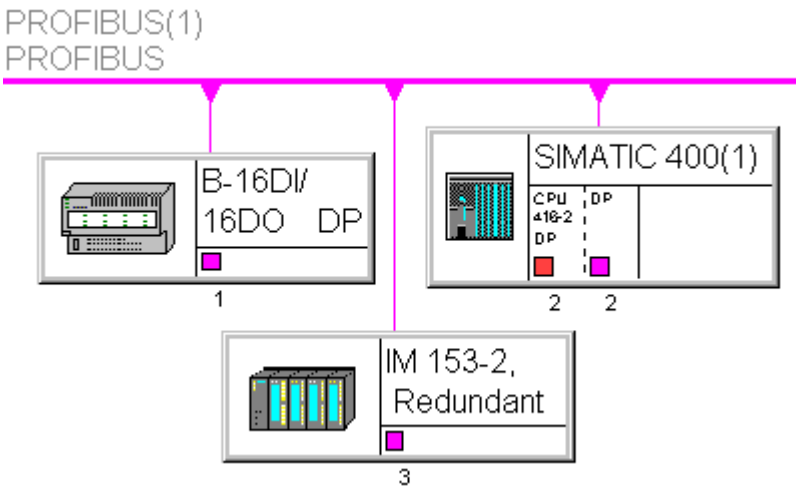
پس از قرار دادن IM153-2 روی شبکه PROFIBUS میبینیم که 11 اسلات ظاهر میشود. از همین جا خواننده باید شباهت ET200M را با یک Station 300 در نظر داشته باشد در اینجا CPU نداریم. کارتهای مختلف ورودی و خروجی آنالوگ و دیجیتال ، کارتهای شبکه و کارتهای FM میتوانند در کاتالوگ از زیر مجموعه IM153-2 مانند شکل روبرو انتخاب شده و در اسلاتهای 4 تا 11 قرار گیرند.تنظیمات زیادی مانند تنظیم آدرس ها و فعال کردن وقفه ها برای این کارتها وجود دارد ولی پرداختن به آنها از حوصله این کتاب خارج است و خواننده محترم میتواند جزئیات موضوع را در تنظیمات سخت افزاری S7-300 در کتاب *راهنمای جامع Step7* ملاحظه نماید.



به این ترتیب پیکر بندی Master/Slave را که شامل یک Master ودو Slave می باشد را انجام دادیم. در برنامه



HW Confing پیکربندی انجام شده را Save نمایید. اگر نتیجه کار را در برنامه NetPro ببینیم شکلی بصورت زیر خواهیم داشت.



باز تأکید می کنیم که مدولهای متصل به DP Slave عیناً همانند مدولهای متصل به Master عمل می کنند یعنی اولاً به لحاظ نوع پذیرش وقفه ها وعملکرد سخت افزاری تفاوتی بین این مدولها و مدولهای Central وجود ندارد،ثانیاً نحوه دسترسی به مدولها عیناً شبیه مدولهای متصل به Master است. برای روشن شدن موضوع فرض کنید یک کارت DI متعلق به یک ET با آدرس شروع ۲۵٦ داریم.این آدرس به صورت خودکار در زمان پیکر بندی توسط STEP7 به این کارت تخصیص داده شده است وهمانطور که دیدید می توان این آدرس را درقسمت پایین صفحه HW Confing مشاهده و در صورت لزوم تغییر داد. در این شرایط برای خواندن سیگنال از این ورودی در برنامه نویسی از دستور زیر استفاده میکنیم:

A I 256.0

دستور فوق ،در یکی از بلاکهای Master CPU مثلاً در بلاک OB1 نوشته می شود واز طریق این CPU به سادگی به Distributed I/O دسترسی پیدا می کنیم.یعنی دقیقاً مثل I/O های متصل به Master با آنها رفتار می کنیم. فرضاً اگر قرار بود به یک کارت DI با آدرس شروع ۲۵٦ برروی خود Master هم دسترسی پیدا کنیم باز هم می نوشتیم A I 256.0 ودر نحوه کار تغییری ایجاد نمی شد.بدیهی است برای سیگنالهای آنالوگ نیز شبیه حالت Central باید از آدرس دهی Peripheral I/O استفاده کنیم مانند :

L PIW0

یا

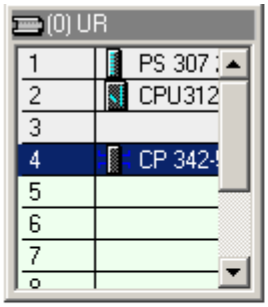
T PQW0

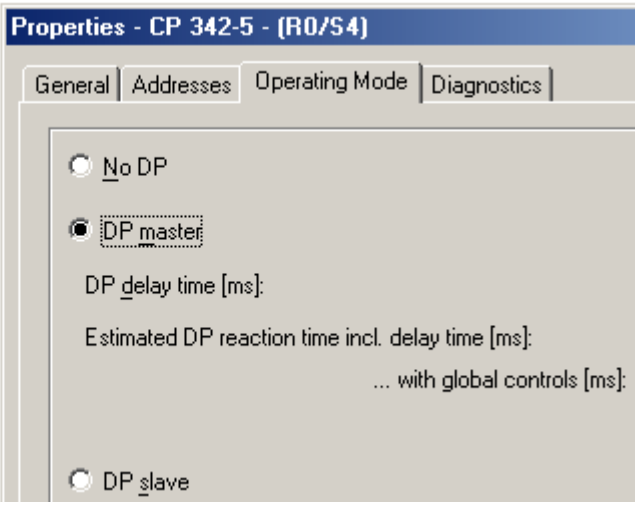
در مثال قبلی برای اتصال Master به شبکه از پورت DP مربوط به CPU استفاده کردیم. برای ایجاد شبکه DP واتصال Master به آن، همانطور که قبلاً گفته شد می توان از مدولهای IM،CP یا IF استفاده کرد.

٤-٣ ایجاد Master با کارت شبکه

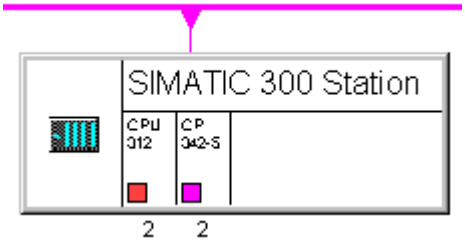
الف) Master با کارت CP

برای حالت Master با کارت CP، یک Station 300 ایجادکنید واز یک CPU معمولی که فاقد پورت DP است مثلاً CPU 312 استفاده کنید وبه آن مدول CP 342-5 را اضافه کنید وقتی این CP را به Rack اضافه می کنید همان پنجره شکل ۱۸-۳ باز می شود کلید New را فشار دهید ونام PROFIBUS (1) را انتخاب کنید .سپس برروی کارت CP دابل کلیک نمایید ودر قسمت Operating Mode حالت DP Master راانتخاب نمایید.



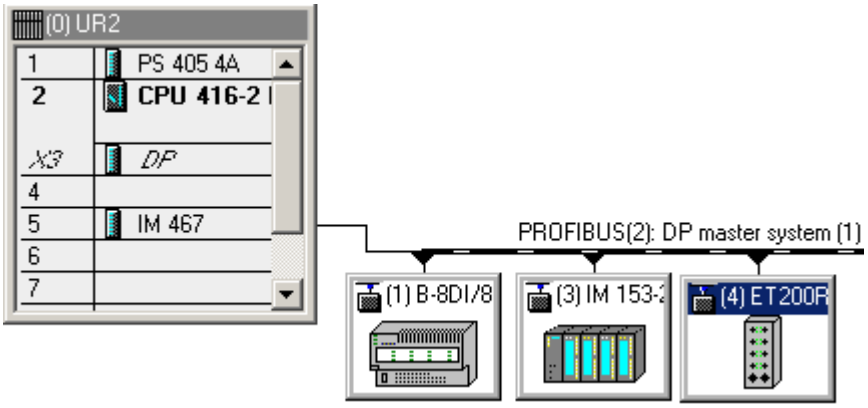


ادامه کار شبیه آنچه قبلا ذکر شد میباشد. میتوان Slave های مختلف روی شبکه ایجاد شده قرار داد و آنها را پیکر بندی کرد. اگرچه ممکن است بسته به نوع کارت CP امکان اتصال برخی از Slave ها وجود نداشته باشد.در اینحالت در Netpro شکل Station 300 بصورت زیر است و مشاهده میشود که اتصال به PROFIBUS از طریق CP انجام شده است.



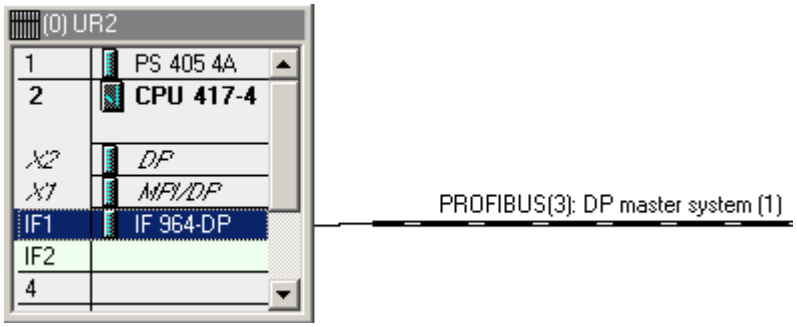
ب) Master با کارت IM

برخی از کارتهای IM و آنهم صرفا برای S7-400 میتوانند نقش Master را ایفا کنند. کارت IM467 از این جمله است. شکل زیر یک Station 400 را با کارت IM467 بعنوان Master نشان میدهد.



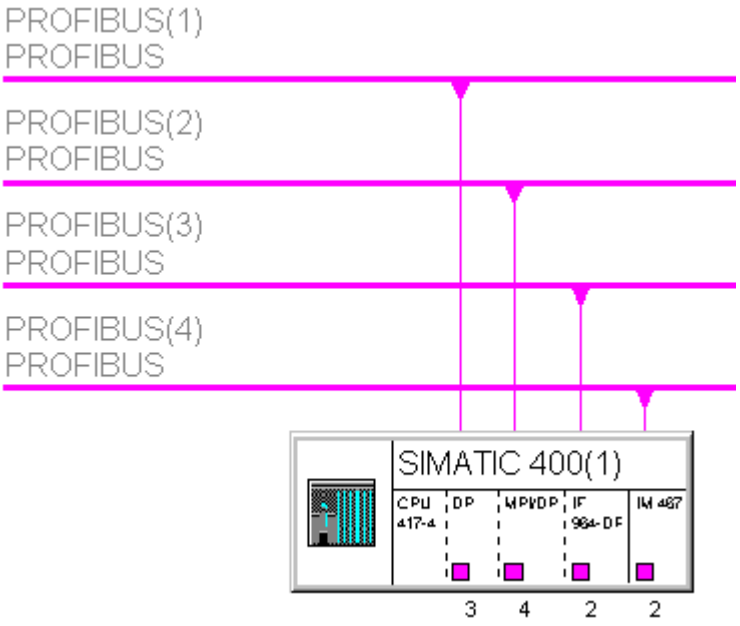
ج) Master با مدول IF

مدول IF نیز خاص S7-400 است و میتواند بعنوان یک Master مستقل از CPU عمل کند بعنوان مثال CPU 417-4 دارای قابلیت نصب کارت IF باشد اگر نوع V1.1 را انتخاب نمایید .همانطور که در شکل می بینید این CPU دو عدد IF را پشتیبانی می کند.در شکل زیر یک عدد مدول IF 964-DP که در پنجره کاتالوگ در زیر مجموعه CPU مربوطه قرار دارد به اسلات مربوطه، کنار CPU اضافه شده و Master System جدیدی ایجاد شده است. مدول IF روی خود CPU نصب میشود.



٤-٤ استفاده از چند Master System بصورت ترکیبی

میتوان در یک Station ترکیبی از Master های فوق را قرار داد و هرکدام را به شبکه جداگانه ای متصل نمود. شاید بتوان گفت در اینحالت قابلیت اطمینان بالاتر است و در صورت قطع شدن یک شبکه سایر شبکه ها میتوانند بکار خود ادامه دهند. شکل زیر ترکیبی از Master ها را برای یک Station 400 نشان میدهد.



تذکر مهم:

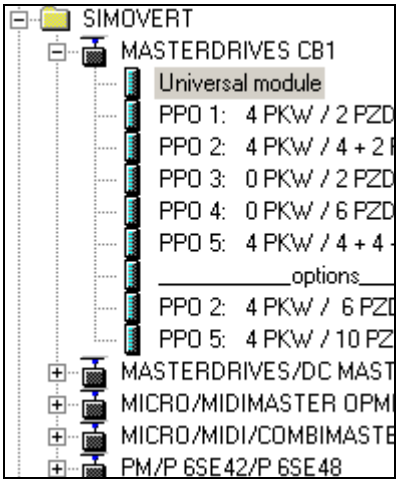
چه در برنامه Hwconfig و چه در برنامه Netpro پس از تکمیل پیکر بندی و ذخیره سازی باید چک سازگاری اجزا (Consistency Check)را انجام دهیم. در HWconfig از منوی Station و در Netpro از منوی Network برای اینکار استفاده میکنیم.بعنوان مثال در Hwconfig اگر ET مدولار به شبکه متصل کرده ولی هیچگونه کارتی در آن قرار ندهیم این یک اشکال است که با اجرای چک توسط برنامه پیغام شکل زیر را مشاهده خواهیم کردیا بعنوان مثال در Netpro اگر آدرس Node ها تلاقی داشته باشد توسط سیستم اعلام خواهد شد.

٤-٥ استفاده از سایر DP Slave ها

DP Slave ها منحصر به ET ها نیستند همانطور که قبلا ذکر شد در پنجره کاتالوگ برنامه های Net Pro و Hwconfig محصولات متنوعی از زیمنس را میبینیم که قابلیت اتصال به شبکه DP را دارند. علاوه بر اینها ممکن است لازم باشد محصولات سازندگان دیگر را نیز به شبکه اضافه کنیم این موارد در زیر مورد بحث قرار گرفته اند:

الف) سایر DP Slave های زیمنس

نحوه قراردادن این تجهیزات روی شبکه DP به همان روش ذکر شده برای ET ها میباشد. یعنی از پنجره کاتالوگ در Hwconfig آنها را برداشته و روی خط شبکه DP قرار میدهیم. نکته ای که باید به آن توجه داشت اینست که بسته به نوع تجهیز ممکن است لازم باشد که کارت اینترفیس مربوط به آن را نیز به نرم افزار معرفی نماییم.

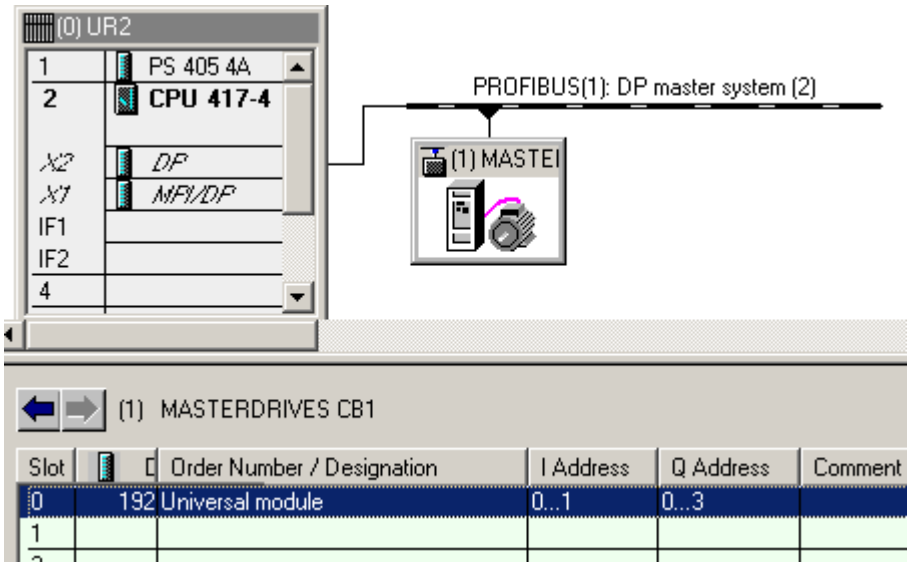


برای روشن شدن موضوع مثالی را مطرح می کنیم . خواننده عزیز با درک موضوع میتواند پیکر بندی سایر DP Slave های زیمنس را شخصا در STEP7 تست کند.

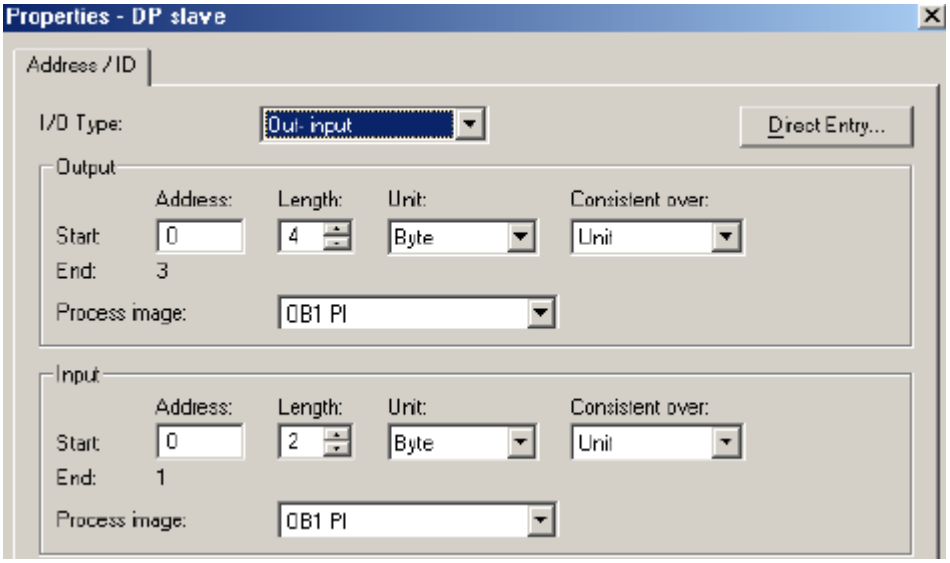
فرض کنید درایو الکتریکی از نوع Simovert که برای کنترل دور موتورهای AC استفاده میشود را قرار است به شبکه PROFIBUS متصل نماییم . در Hwconfig از پنجره کاتالوگ آنرا با ماوس برداشته و روی شبکه DP قرار میدهیم. ولی این کار به تنهایی کافی نیست.

اگر در این حالت Consistency Check را اجرا کنیم با خطایی شبیه آنچه در صفحه قبل ذکر شد مواجه میشویم.لازم است کارت اینترفیس مربوطه را نیز در پنجره کاتالوگ از زیر مجموعه Simovert برداشته و در اسلات مربوطه قرار دهیم.

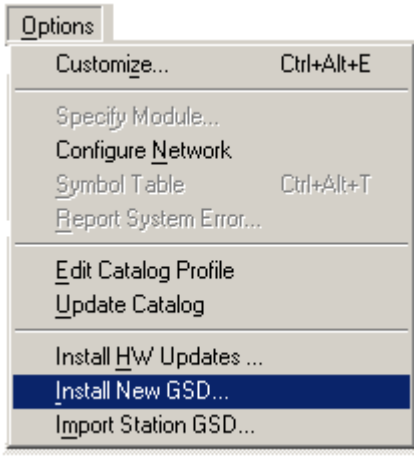
همانطور که در شکل صفحه بعد مشاهده می شود کارت Universal در اسلات مربوطه قرار گرفته است. برای این کارت که در برخی DP Slave های دیگر نیز مشاهده میشود لازم است آدرس های ورودی و خروجی را تعیین کنیم.



برای تنظیم آدرس ها کافیست روی کارت Universal کلیک کرده و در پنجره ای مانند شکل زیر آنها را تعیین نماییم. برخی از کارتهای اینترفیس قبلا با ورودی و خروجی های مشخصی تنظیم شده اند.

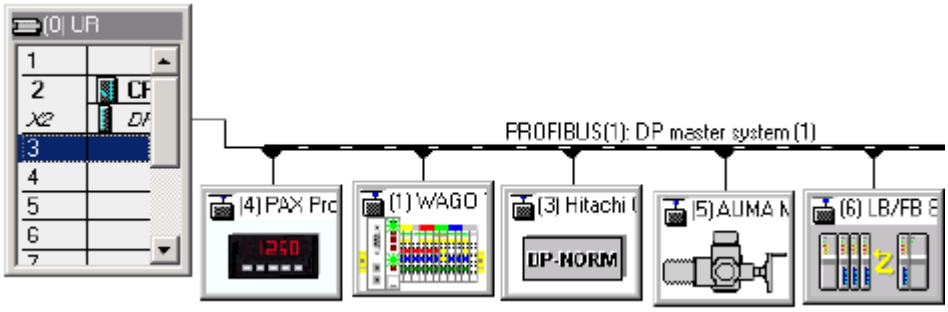


ب) DP Slave های سازندگان دیگر



برای وارد کردن DP Slave های سازندگان دیگر لازم است فایل GSD وسیله مربوطه در دسترس باشد.این فایل را در برنامه Hwconfig توسط منوی Option > Install New GSD به نرم افزار بار میکنیم. پس از آن مشاهده خواهیم کرد که تجهیز مربوطه به پنجره کاتالوگ اضافه میگردد. سعی شود در هنگام نصب GSD سایر زیربرنامه های STEP7 بسته باشد تا پیغام خطا ظاهر نگردد.

شکل زیر DP Slave های مختلفی را از سازندگان غیر زیمنس نشان میدهد که روی شبکه DP مربوط به S7-300 قرار گرفته اند. با کلیک کردن روی هرکدام از این تجهیزات می بینیم که اسلاتهای خالی در پایین برنامه Hwconfig ظاهر میشود. در این اسلاتها باید کارتهای ورودی و خروجی مورد نیاز رااز زیر مجموعه تجهیز مزبور در پنجره کاتالوگ قرار داد.



فصل پنجم – پیکربندی Intelligent Slaves در STEP7

مشتمل بر :

۵-۱ پیکر بندی Master با I-Slave

۵-۲ پیکر بندی I-Slave با DP Slave

۵-۳ پیکر بندی I-Slave با DP Slave مربوط به DP Master های مختلف

۵-٤ پیکر بندی I-Slave با DP Master

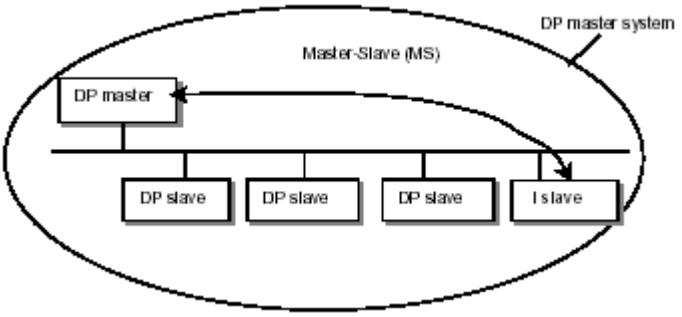
مقدمه :

در بخش های قبلی توضیحاتی در مورد مفاهیم I-Slave وانواع آن دادیم. به طور خلاصه گفتیم I-Slave یک DP Slave هوشمند است از اینرو پردازش اولیه ای روی Distributed I/O ها انجام داده و در صورت لزوم فرمان های محلی را به خروجی ها ارسال می کند.تجهیزات زیر میتوانند بعنوان I-Slave عمل کنند :

- ET200S دارای CPU
- T200X دارای CPU
- CPU از خانواده S7-300 که قابلیت Slave را نیز داشته باشد.
- CPU از خانواده S7-400 که قابلیت Slave را نیز داشته باشد.
- کارت CP از خانواده S7-300 که قابلیت Slave را نیز داشته باشد.
- کارت CP از خانواده S7-400 که قابلیت Slave را نیز داشته باشد.

۵-۱ پیکر بندی Master با I-Slave

در این روش از یک DP Master همراه با یک یا چند I-Slave استفاده میگردد. در عین حال میتوان همراه با آنها DP Slave نیز روی شبکه قرار داد.

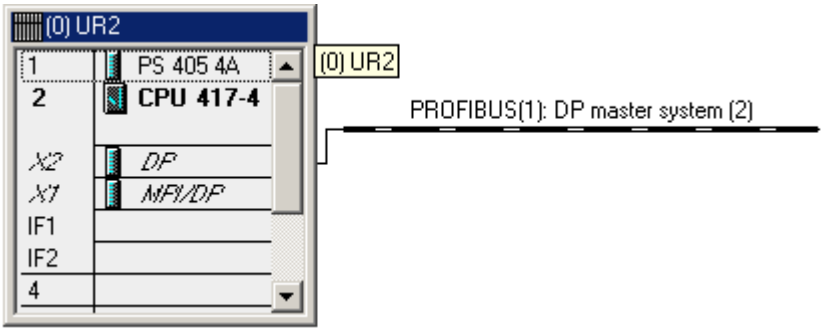


در این روش :

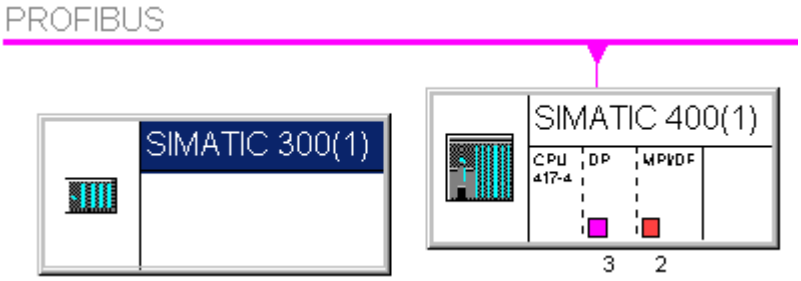
- برنامه اتوماسیون به تعدادی Subtask تقسیم شده و هر قسمت در یکی از I-slave ها قرار دارد.
- DP Master فقط کنترل و نظارت کلی بر I-slave ها دارد . پردازش محلی بعهده I-Slave هاست.
- DP master به I/O های I-Slave ها مستقیماً دسترسی ندارد. و فقط به CPU های I-Slave ها و آنهم صرفاً آدرس هایی که از قبل در پیکر بندی تعریف شده (صرفا جداول PII و PIQ) دسترسی دارد .
- I-Slave ها را نمی توان همانند DP Slave ها به شبکه PROFIBUS متصل نمود. برای اینکار روش خاصی وجود دارد . گام هایی که در این روش باید برداشته شود در صفحه بعد توضیح داده شده است.

گامهایی که برای پیکر بندی سیستم Master<>I-Slave باید برداشت:

گام اول : ایجاد DP Master و فعال کردن آن در Hwconfig



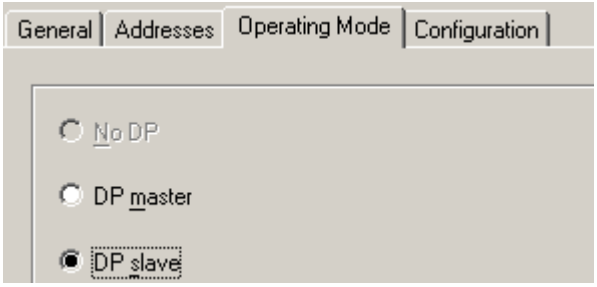
گام دوم : اجرای NetPro و وارد کردن Station 300



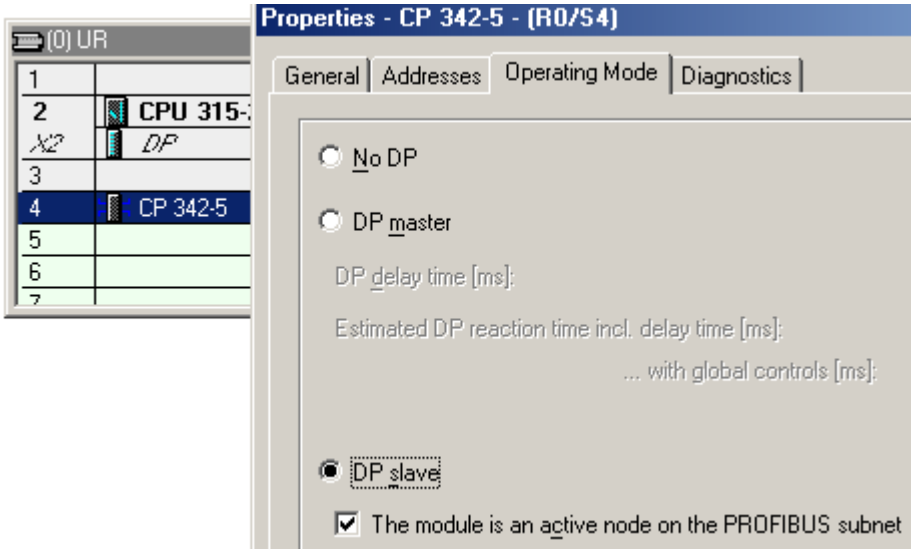
گام سوم : روی Station 300 در محیط Netpro دابل کلیک کرده تا توسط Hwconfig باز شود.

گام چهارم: با بازشدن برنامه Hwconfig بسته به نوع I-Slave یکی از کارهای زیر را انجام میدهیم:

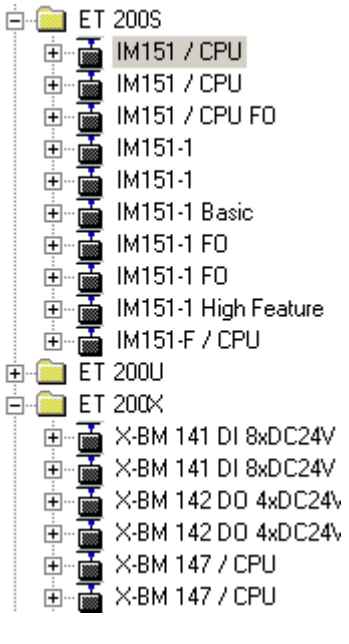
الف) استفاده از CPU 300 با قابلیت Slave مانند CPU315-2DP با کد سفارش 315-2AF82-0AB0 لازم به ذکر است قابلیت Slave شدن را میتوان در توضیحات پایین پنجره کاتالوگ چک کرد. پس از وارد کردن رک و CPU مورد نظر روی اسلات DP کلیک کرده و در پنجره مانند شکل زیر مد Slave را انتخاب میکنیم.

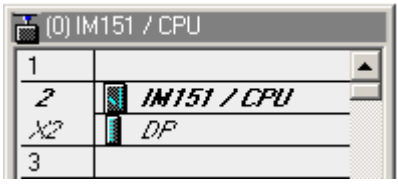


ب) استفاده از کارت CP با قابلیت DP و امکان Master یا Slave شدن مانند کارت CP342-5



ج) استفاده از ET200X یا ET200S دراینحالت در پنجره خالی Hwconfig بدون اینکه رک وارد کنیم از پنجره کاتالوگ مانند شکل روبرو روی یکی از ET های فوق که در انتهای نام آن کلمه CPU نوشته شده دوبارکلیک میکنیم تا وارد Hwconfig شود.



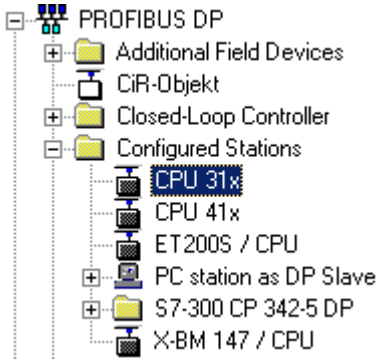


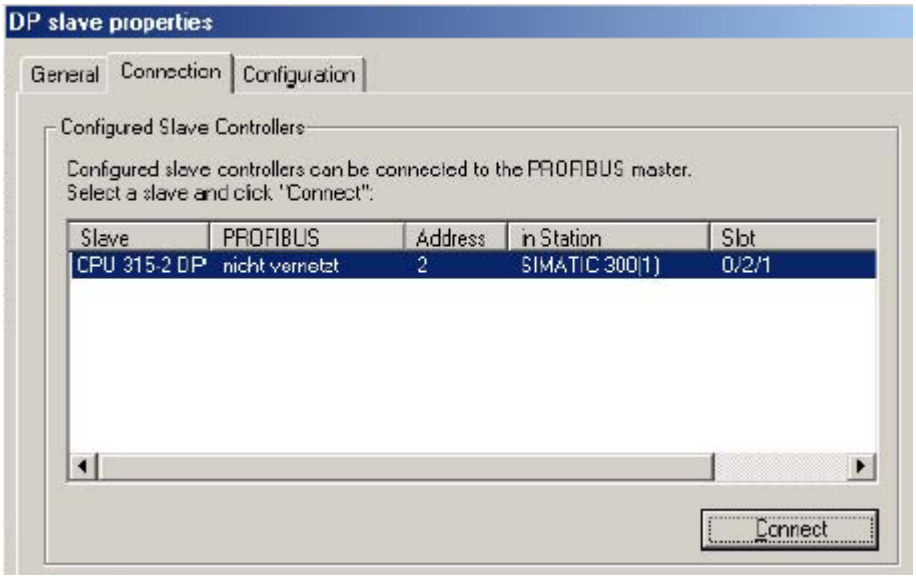
در این حالت نیز روی DP در اسلات مربوطه دابل کلیک کرده و سپس مد DP Slave را برای آن انتخاب میکنیم.

در تمام حالات الف و ب و ج لازم است پیکر بندی را ذخیره نماییم.

گام پنجم : باز کردن DP Master در HWconfig با دو بار کلیک کردن روی آن در Netpro .

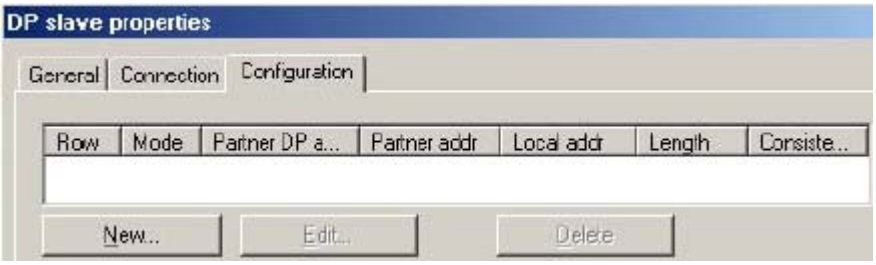
گام ششم : با استفاده از کاتالوگ DP قسمت Configured Station را باز کرده و بسته اینکه کدام I-Slave مد نظر است تجهیز مورد نظر را انتخاب کرده روی شبکه PROFIBUS در DP Master قرار میدهیم. برای X-BM از ET200X 147/CPU استفاده میکنیم. مشاهده میکنیم که پنجره ای مانند شکل زیر باز میشود.



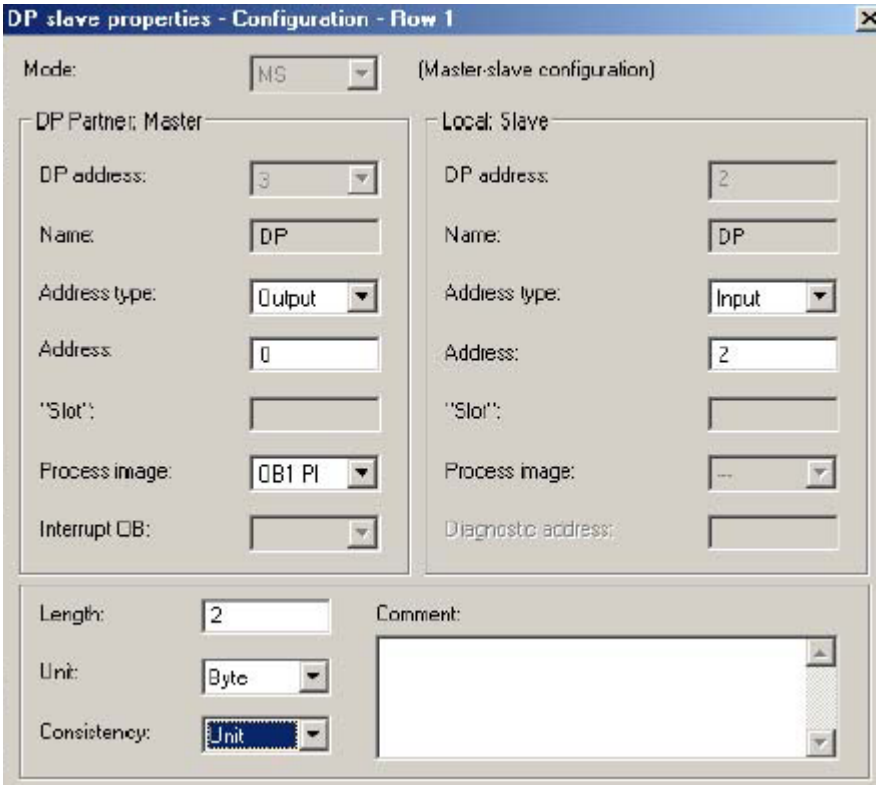


گام هفتم : در پنجره فوق روی Connect کلیک میکنیم با برقراری اتصال تجهیز سطر مربوطه از پنجره حذف میشود. بدیهی است اگر چندین Station 300 بصورت Slave پیکر بندی شده باشد در این لیست چند انتخاب ظاهر میشود.

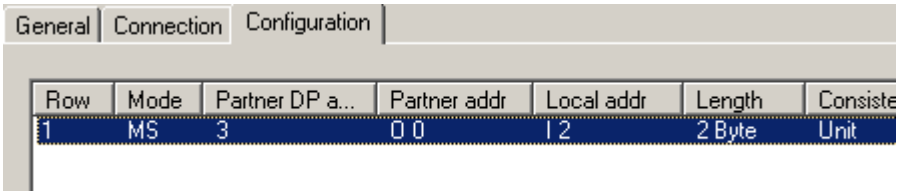
گام هشتم : در پنجره قبل روی Configuration سپس روی New کلیک میکنیم



گام نهم : پنجره ای مانند شکل زیر باز میشود که سمت چپ مربوط به Master و سمت راست مربوط به Slave است. نواحی آدرس ها را در اینجا باید مشخص نمود. اگر در یکطرف Input انتخاب شود در سمت دیگر باید Output را انتخاب نمود. در Address نقطه شروع را مینویسیم. طول دیتا را در قسمت Length مشخص میکنیم که چند بایت یا چند Word باشد. بعنوان مثال اگر آدرس ورودی 0 و طول دیتا 2 بایت باشد یعنی IB0 و IB1. در ضمن لازم نیست که آدرسهای دوطرف هردو از یک عدد شروع شود.



در پنجره قبلی پس از تکمیل کار و کلیک کردن روی OK مشاهده میکنیم که در پنجره Configuration یک سطر مانند شکل زیر ظاهر میگردد. منظور از مد MS در این سطر همان مفهوم Maser/Slave است. در این سطر مشخص است که ناحیه PIQ مربوط به حافظه Master از آدرس 0 با ناحیه PII مربوط به حافظه Slave از آدرس ۲ در تماس است و دیتاهایی با حجم ۲ بایت و با آدرس های شروع فوق الذکر بین آنها تبادل میشود. بهمین روش میتوان سطر دیگری برای تبادل دیتا بین ناحیه ورودی Master و ناحیه خروجی Slave به این جدول اضافه کرد .



با استفاده از این پیکر بندی برنامه نویسی به سهولت انجام میشود. در برنامه Master میتوان از آدرس های تعریف شده فوق استفاده کرد . مانند:

L QB0 یا T QB0

خواننده محترم میتواند شبیه آنچه در مثال فوق برای عملکرد CPU از خانواده S7-300 بعنوان Slave ذکر شد را برای سایر I-Slave ها انجام داده و نتیجه پیکر بندی را مشاهده نماید.

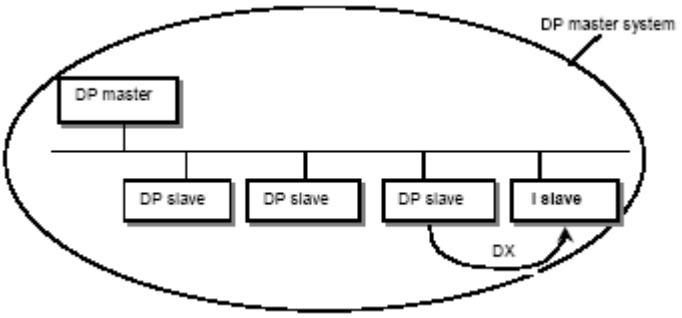
تذکر:

- وقتی یک I-Slave را با یک Master پیکر بندی میکنیم نمیتوان از آن I-Slave برای Master دیگری استفاده کرد مگر اینکه آنرا از Master اول Disconnect کنیم.
- روی یک شبکه PROFIBUSمیتوان چند Master و چند I-slave را بگونه ای پیکر بندی کرد که هر I-Slave با Master خاص خودش در ارتباط باشد. این کار شبیه پیکر بندی DP Slave های مختلف با Master های مختلف میباشد.

۵-۲ پیکر بندی I-Slave با DP Slave

در این روش I-Slave با DP Slave ارتباط مستقیم برقرار میکند که به Direct Data Exchange معروف است و با ارتباط DX نمایش داده میشود.

ارتباط DX قابلیتی است که در نسخه DP-V2 به PROFIBUS اضافه شده است. در این روش لازم نیست که برای تبادل دیتا بین دو Slave پای Master نیز در میان باشد.بنابر این سرعت تبادل بیشتر خواهد بود.

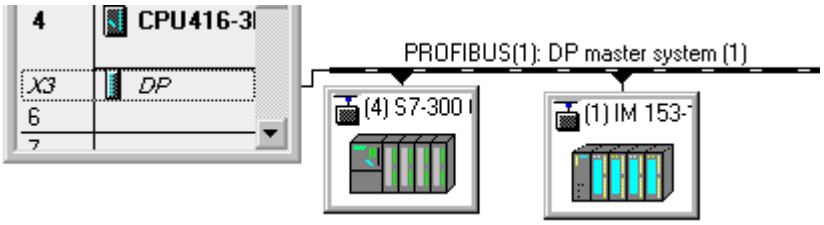


گامهایی که برای پیکر بندی سیستم DP Slave<>I-Slave باید برداشت:

گام اول : ایجاد DP Master و فعال کردن آن در Hwconfig مشابه آنچه برای Master<>Islave بیان شد. تنها نکته مهمی که باید توجه داشت اینست که Master قابلیت DX داشته باشد این قابلیت را میتوان در ویژگیهایی که در زیر پنجره کاتالوگ ظاهر میشود چک کرد.

گام دوم : انتخاب DP Slave وقرار دادن آن روی شبکه DP Master در Hwconfig . بعنوان مثال یک ET200M را در نظر بگیرید که روی شبکه قرار داده شده و کارتهای مربوطه نیز در اسلاتهای آن قرار گرفته اند.

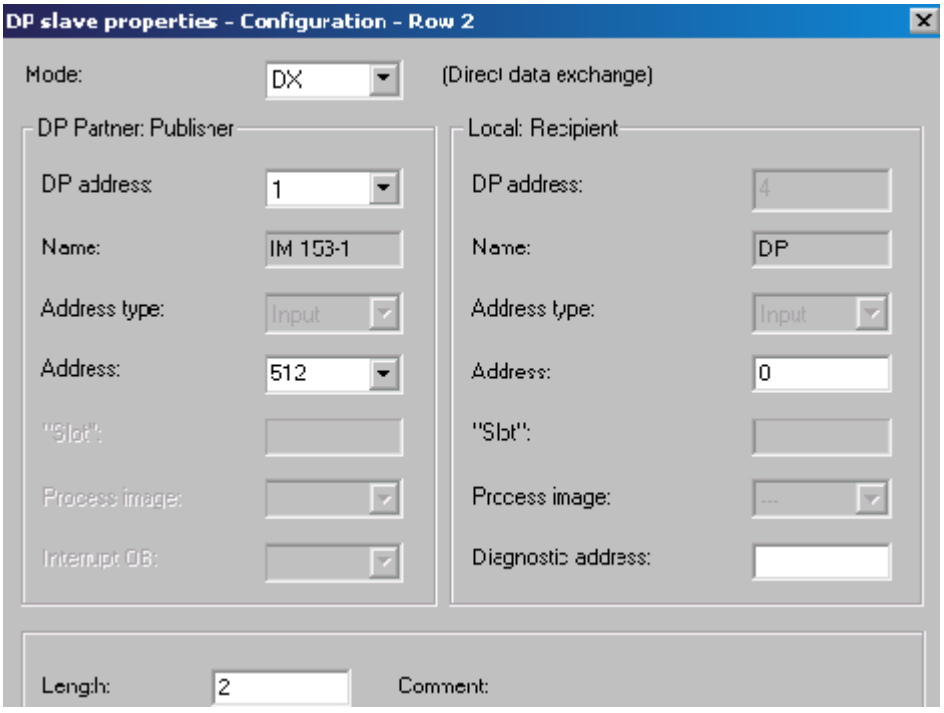
گام سوم : پیکر بندی یک I-Slave به روش ذکر شده در قسمت قبل و قرار دادن آن روی شبکه



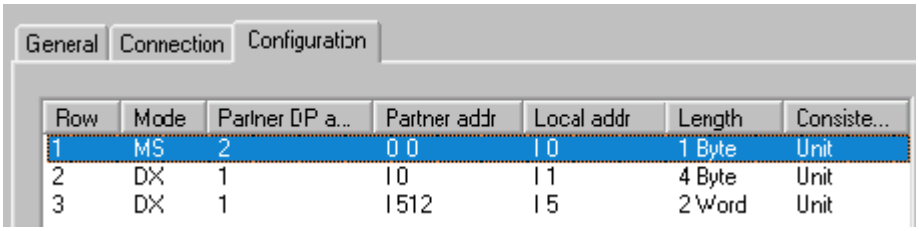
گام چهارم : کلیک کردن روی I-Slave تا پنجره Configuration شبیه گام هشتم ذکر شده قبلی بازشود.

گام پنجم : در پنجره Configurataion روی New کلیک کرده تا پنجره ای مانند شکل زیر باز شود در این پنجره :

- ابتدا باید مد DX را انتخاب کرد.
- اگر بیش از یک DP Slave داشته باشیم در قسمت سمت چپ آدرس Node برای Slave مورد نظر را مشخص میکنیم (در جلوی DP Address)
- آدرسهای مربوط به DP Slave که باید در دسترس I-Slave قرار گیرند را قسمت سمت چپ در جلوی Address مشخص میکنیم. این آدرس ها معمولاً بر اساس آنچه برای DP Slave در HWcofig تنظیم شده ظاهر میگردد.
- در قسمت سمت راست نیز آدرس های مربوط به I-Slave را مشخص میکنیم.
- اندازه دیتا را شبیه قبل با استفاده از Length و Unit مشخص می نماییم.

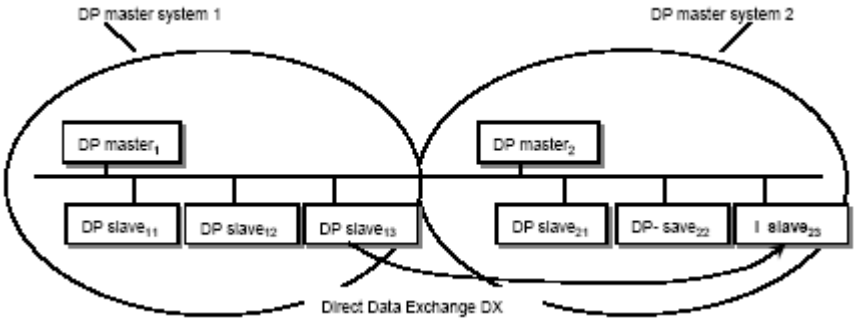


گام ششم : پس از Apply کردن میبینیم که ارتباط DX در پنجره Configuration مربوط به I-slave مانند شکل زیر ظاهر میگردد. در صورت ایجاد چند ارتباط DX یا MS آدرسها نباید تداخل داشته باشند. این کار توسط خود نرم افزار نیز کنترل میشود.



۳-۵ پیکر بندی I-Slave با DP Slave که دارای DP Master های مختلف هستند

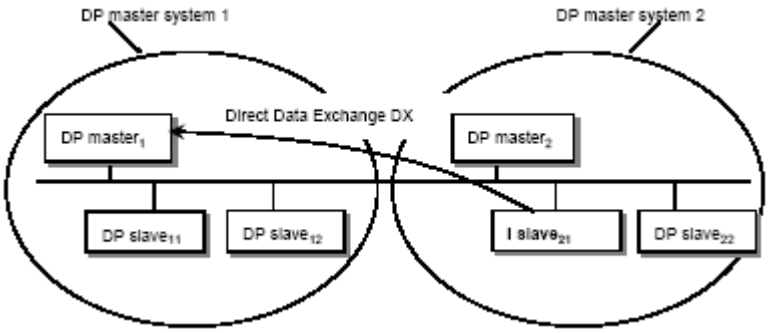
در حالت قبل I-Slave و DP Slave که با یکدیگر ارتباط DX برقرار میکردند هر دو دارای یک DP Master بودند. اکنون حالتی را در نظر بگیرید که هر کدام متعلق به یک DP Master جداگانه باشند ولی مجموعه آنها روی یک شبکه PROFIBUS مانند شکل قرار گرفته باشد. این ساختار نیز تبادل دیتا را سرعت میبخشد . زیرا اگر چنین امکانی وجود نداشت لازم بود در مثال شکل زیر ابتدا I-Slave23 با DP Master2 گفتگو کند سپس DP Master2 با DP Master1 و نهایتا دیتا از DP Master1 به DP Slave23 داده شود. اینراه طولانی با امکان DX فوق بسیار کوتاه شده است.



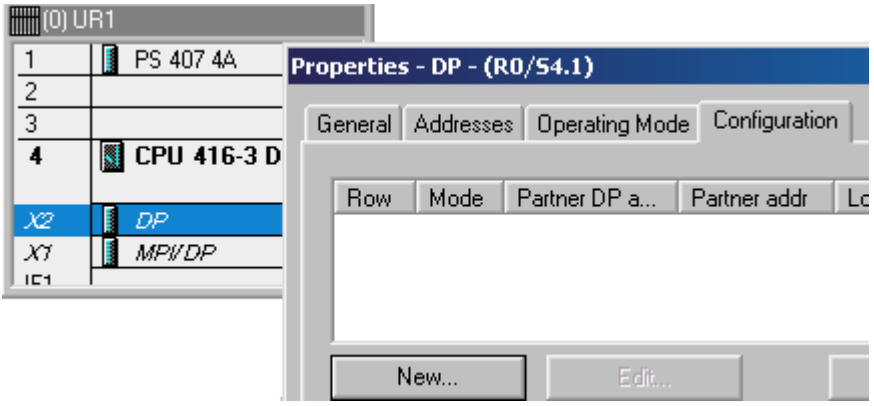
با توضیحاتی که برای روشهای قبلی داده شد پیکر بندی این روش نیز ساده است. کافی است دو DP-Master جداگانه که هر دو امکان DX داشته باشند روی یک باس مشترک ایجاد کرده روی یکی I-slave و روی دیگری DP Slave قرار دهیم. سپس روی I- Slave و آدرس Node مربوط به DP Slave را در پنجره Configuration انتخاب میکنیم . بقیه کارها شبیه قبل است.

٤-٥ پیکر بندی I-Slave با DP Master

در این حالت یک I-Slave با DP Master غیر از DP Master خودش ارتباط مستقیم برقرار میکند.



DP Master باید امکان DX داشته باشد. در اینحالت بجای اینکه روی I-Slave در پنجره HWconfig کلیک کنیم روی اسلات DP مربوط به DP Master کلیک میکنیم تا مانند شکل زیر پنجره Configuration باز شود.



شبیه قبل با کلیک روی New پنجره Cofiguration باز میشود که در آن آدرس Node و آدرس های مربوط به حافظه را انتخاب میکنیم

توجه داشته باشید از سمت DP Master1 میتوانیم آدرس Node های مربوط به Slave های DP Master2 را ببینیم که I-Slave یکی از آنهاست. همینطور از سمت DP Master 2 میتوانیم آدرس Node های مربوط به Slave های DP Master1 را مشاهده کنیم و بین آنها ارتباط برقرار کنیم. بنابراین به این طریق یک DP Master نه تنها با I-Slave که با DP-Slave های Master های دیگر نیز میتواند تبادل دیتا داشته باشد.

فصل ششم – ساختارهای مختلف شبکه PROFIBUS در STEP7

مشتمل بر :

۶-۱ ساختار یک شبکه PROFIBUS در یک پروژه Step7

۶-۲ ساختار چند شبکه PROFIBUS در یک پروژه Step7

۶-۳ ساختار یک شبکه PROFIBUS در چند پروژه Step7

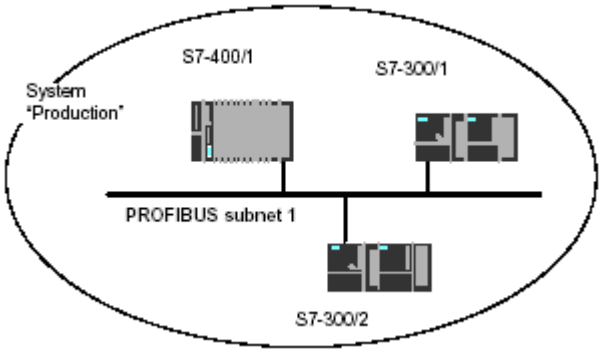
مقدمه :

در این بخش به ساختارهای مختلف شبکه PROFIBUS در STEP7 می پردازیم در اینجا منظور از PROFIBUS صرفا پروتکل DP نیست و FMS را نیز شامل میشود. همانطور که میدانیم این دو پروتکل میتوانند بطور همزمان روی یک شبکه استفاده شوند. ساختارهایی که در این قسمت بحث میشوند عبارتند از :

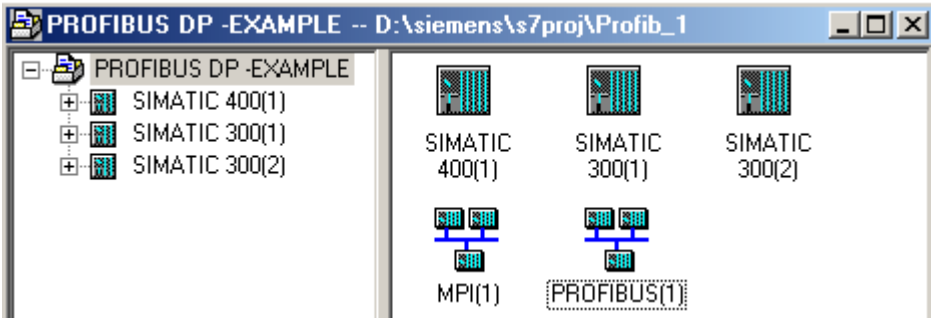
- یک شبکه در یک پروژه
- دو یا چند شبکه در یک پروژه
- یک شبکه در چند پروژه

۱-۶ ساختار یک شبکه PROFIBUS در یک پروژه Step7

این ساختار ساده ترین حالت است . با توضیحاتی که تاکنون داده شده کاربر با این ساختار بخوبی آشناست. Station های مختلف که میتوانند Master یا I-Slave باشند در هنگام پیکر بندی همگی به یک شبکه PROFIBUS متصل میشوند مانند شکل زیر :

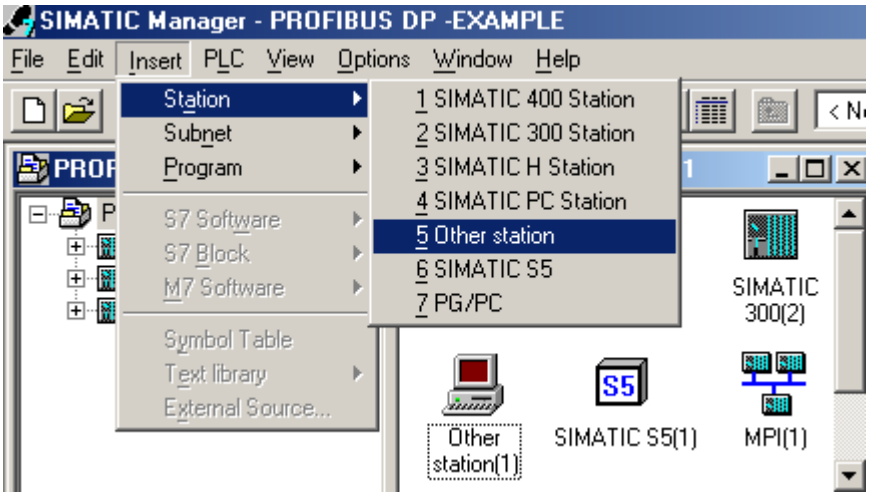


در اینحالت در پنجره Simatic Manager المانهایی مانند شکل زیر خواهیم داشت. همانطور که مشخص است فقط یک المان برای PROFIBUS وجود دارد.

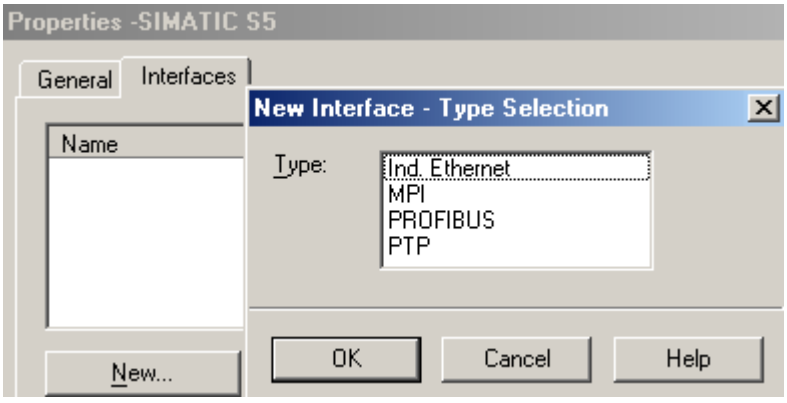


نکته ای که در همین جا لازم است خاطر نشان شود نحوه قراردادن یک Station روی شبکه است که از خانواده S7 نباشد. اینکار هم از طریق Simatic Manager امکان پذیر است و هم از طریق NetPro.

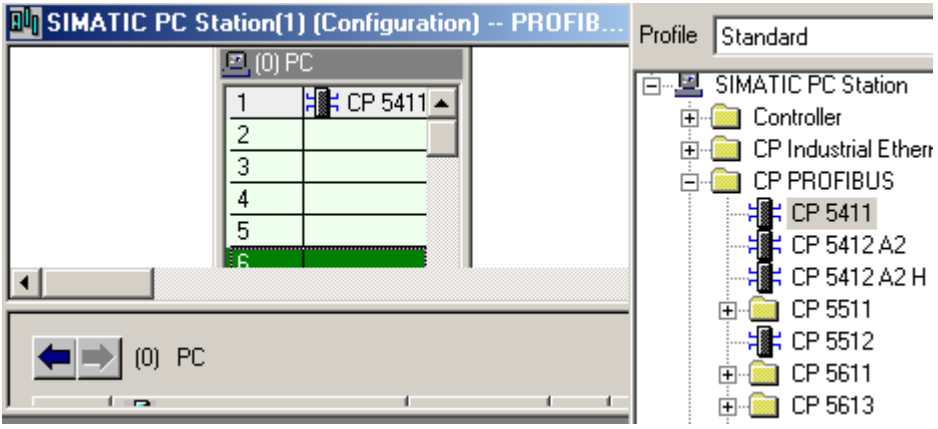
در Simatic Manager با استفاده از منوی Insert > Station استفاده میکنیم. برای S5 از Simatic S5 و برای PC از Simatic PC Station و برای محصولات غیر زیمنس از Other Station استفاده میکنیم.



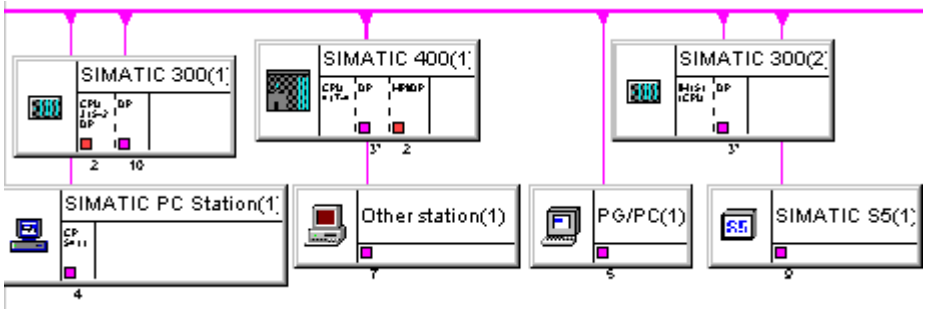
می بینیم که آیکون مربوطه در پنجره ظاهر میشود با کلیک کردن روی هر کدام از این آیکونهای پنجره ای باز میشود که در بخش Interface آن باید نوع ارتباط را تعیین کرد. با انتخاب PROFIBUS و OK کردن امکان اتصال این Station به شبکه مورد نظر فراهم شده و آدرسی برای این Node توسط سیستم داده میشود.



برای اتصال PC نیز از منوی SIMATIC PC Station > Station > Insert استفاده میکنیم. با کلیک روی آیکون آن ، برنامه HWConfig باز میشود و لازم است از پنجره کاتالوگ کارت ارتباطی مورد نظر را از زیر مجموعه SIMATIC PC Station مانند شکل زیر انتخاب کرده و در یکی از اسلاتهای ظاهر شده قرار میدهیم.

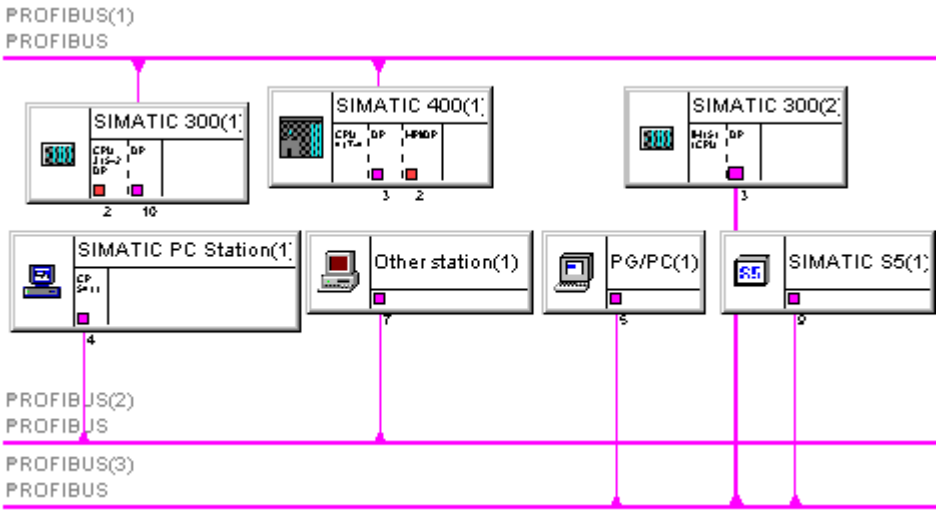


پس از اتمام کار و ذخیره سازی اگر برنامه NetPro را اجرا کنیم شکلی شبیه زیر خواهیم داشت:

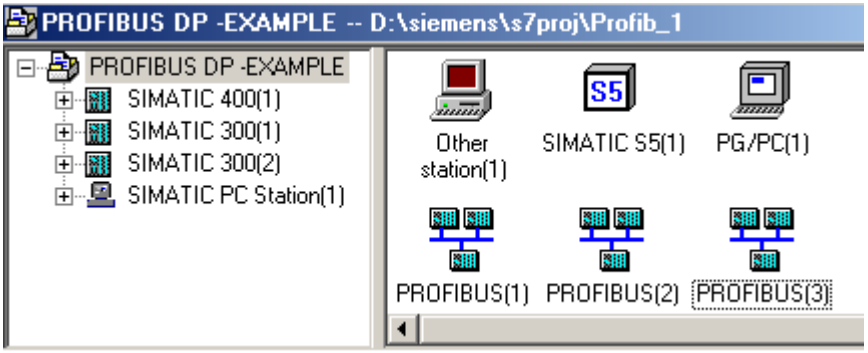


۲-۶ ساختار چند شبکه PROFIBUS در یک پروژه Step7

در یک پروژه میتوان چند شبکه PROFIBUS ایجاد کرد که کاملا از یکدیگر مستقل باشند و هر کدام دارای Master و Slave های خاص خود باشند. اینکار را میتوان در Hwconfig هنگام معرفی Master System انجام داد. روش دیگر نیز استفاده از NetPro و وارد کردن چند Subnet و سپس اتصال Station ها به Subnet های مورد نظر است.

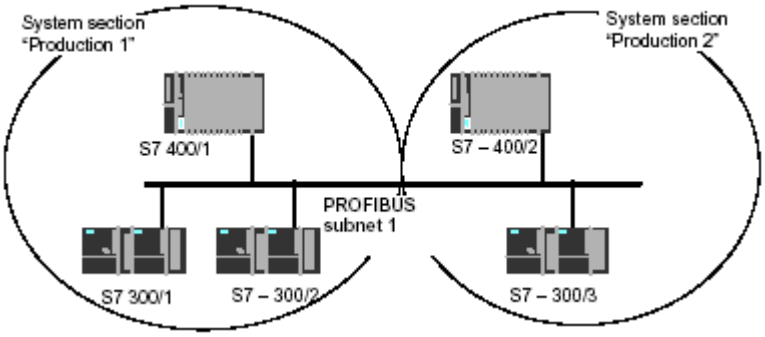


پس از ذخیره سازی اگر به Simatic Manager باز گردیم آیکون مربوط به Subnet های فوق را در پنجره آن مشاهده خواهیم کرد:



۳-۶ ساختار یک شبکه PROFIBUS در چند پروژه Step7

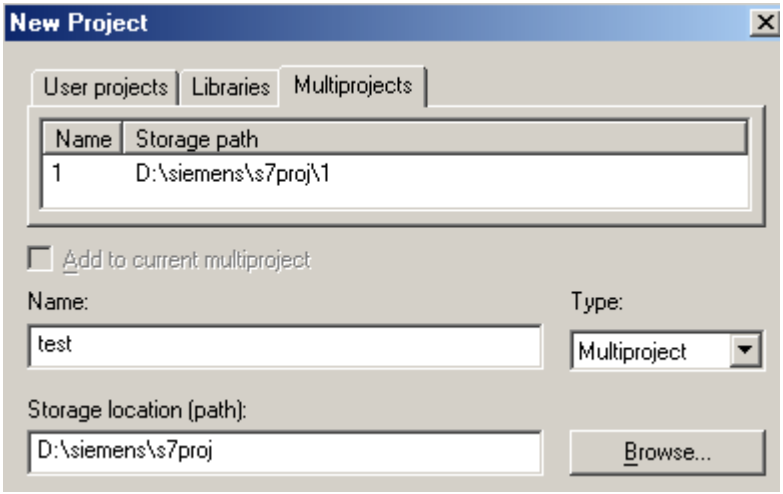
در طراحی ممکن است اتوماسیون یک کارگاه به چند پروژه تقسیم شود ولی در همه آنها یک شبکه PROFIBUS وجود داشته باشد.



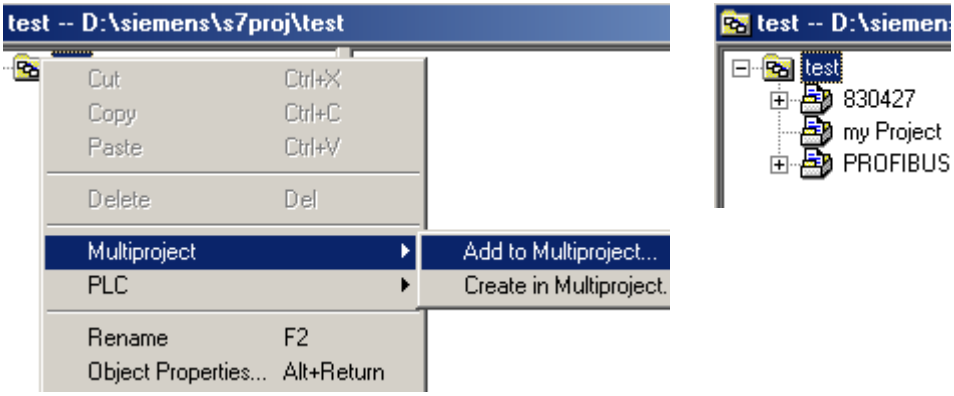
برای این منظور باید یک Multiproject ایجاد کرد سپس شبکه های پروژه ها را باهم ترکیب (Merge) نمود. گامهای زیر لازم است بترتیب برداشته شوند:

گام اول : در Simatic Manager پروژه های مورد نظر را ایجاد کرده و شبکه PROFIBUS آنها را پیکر بندی میکنیم.

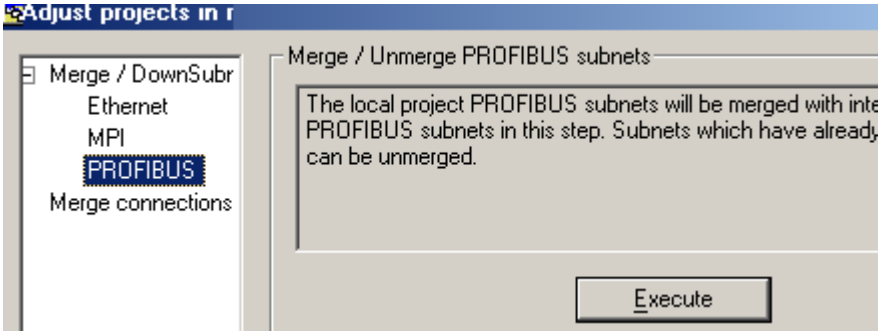
گام دوم : در Simatic Manager از منوی File > New استفاده کرده و در پنجره در قسمت Multiproject اسم دلخواه را وارد میکنیم.



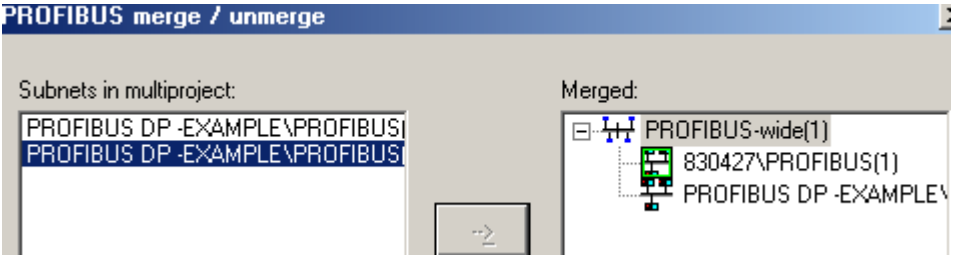
گام سوم : روی اسم پروژه با کلیک راست ماوس مانند شکل زیر Add to Multiproject را انتخاب میکنیم . لیست پروژه های موجود ظاهر میشود. از این لیست پروژه های مورد نظر را انتخاب و به پنجره وارد میکنیم.



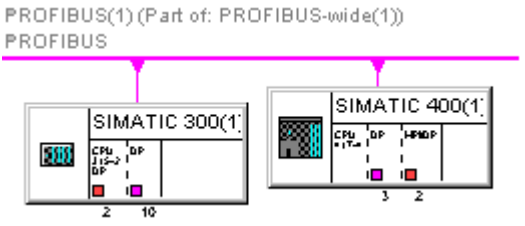
گام چهارم : از منوی File>Multiproject استفاده کرده و Adjust Projects را انتخاب میکنیم پنجره ای مانند شکل زیر ظاهر میشود. این پنجره برای Merge کردن انواع Subnet ها بکار میرود و اختصاص به PROFIBUS ندارد. در اینجا ما PROFIBUS را انتخاب میکنیم سپس روی Execute کلیک میکنیم.



گام پنجم : پنجره ای مانند شکل بالای صفحه بعد باز میشود که در آن لیست شبکه های PROFIBUS موجود در پروژه ها نمایش داده میشود. هر کدام که مد نظر است را به سمت راست منتقل میکنیم. سپس OK و Save مینماییم.



در این مرحله Merge کردن شبکه ها انجام شده است . برای Unmerge کردن نیز همین مسیر را دنبال می نماییم. نتیجه ترکیب شبکه ها را میتوانیم در NetPro ببینیم مشاهده خواهیم کرد که اصطلاح PROFIBUS-wide برای شبکه کلی انتخاب شده است.



نکات قابل توجه :

- توسط برنامه NetPro از منوی Edit>merge/Unmerge Subnetwork نیز میتوان عمل Merge و Unmerge را انجام داد. روش کار و پنجره های مربوطه به همان صورت است که توضیح داده شد.
- پس از Merge کردن لازم است در NetPro چک سازگاری اجزا و عدم تداخل آدرسهای Node ها از منوی Network > Check Interproject Consistency انجام شود.
- عمل Merge کردن نمیتواند برای چند شبکه که داخل یک پروژه قرار دارند انجام شود. در واقع نیازی هم به این کار نیست زیرا کاربر میتواند بسادگی تمام اجزا را به یک شبکه متصل نماید.

فصل هفتم – برنامه نویسی ارتباطات PROFIBUS-DP

مشتمل بر :

۷-۱ سرویس DP برای Master / Slave

۷-۲ سرویس DP برای Master / I-Slave

مقدمه

در بخش های قبل حالت های مختلف پیکر بندی Master / Slave در Profibus توضیح داده شد. در این بخش نحوه برنامه نویسی برای ارتباط DP تشریح میشود. نحوه استفاده از فانکشن های ارتباطی وقتی از کارت CP استفاده شود همراه با مثال مورد بحث قرار می گیرند.

۷-۱ سرویس DP برای Master / Slave

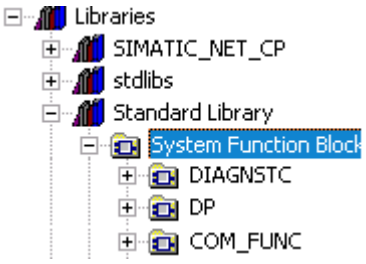
این سرویس نیاز به برنامه نویسی خاصی ندارد. ورودی و خروجی ها شبیه کارتهای I/O که در کنار CPU قرار میگیرند برنامه نویسی میشوند . بعنوان مثال فرض کنید آدرسی که در Hwconfig برای یک ورودی دیجیتال ET200M داده شده آدرس 4 و آدرسی که برای ورودی آنالوگ داده شده آدرس 254 باشد. در اینصورت در برنامه CPU این ورودی ها بصورت زیر استفاده میشوند:

A I 4.0	برای خواندن یک بیت دیجیتال ورودی مثلا کانال اول
L PIW 254	برای خواندن یک ورودی آنالوگ

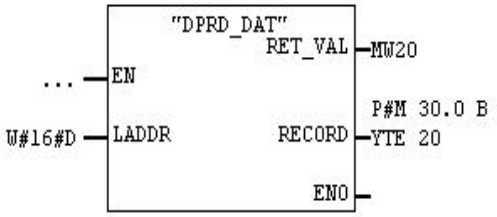
پس بعبارت دیگر :

- مزیت این روش آنست که آدرس دهی شبیه حالت Central با همان فرمت و همان دستورات است..
- عیب این روش آنست که با دستورات Load و Transfer هر بار ماکزیمم چهار بایت را میتوان ارسال یا دریافت کرد.

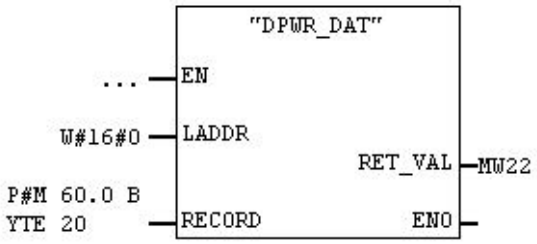
برای رفع عیب فوق الذکر فانکشن هایی توسط زیمنس تعبیه شده که میتوان آنها را در زیر مجموعه System Function پنجره Program Element برنامه LAD/STL/FBD مانند شکل زیر مشاهده و در صورت نیاز استفاده کرد. از جمله این فانکشن ها SFC14 و SFC15 را میتوان نام برد.



SFC14 با نام سمبلیک DPRD-DAT برای دریافت دیتا و SFC15 با نام سمبلیک DPWR-DAT برای ارسال دیتا استفاده می شود.



آنچه در این بلاک ها تحت عنوان LADDR ظاهر می شود آدرس مقصد است که دیتا باید به آن ارسال یا از آن دریافت شود. و باید به فرمت Hex نوشته شود بعنوان مثال آدرس Diagnostic=100 تبدیل به LADDR=W#16#64 خواهد شد.



حجم دیتایی که باید ارسال یا دریافت شود توسط آدرسی که بصورت Pointer در جلوی Record نوشته میشود مشخص میگردد.بعنوان مثال اگر قرار باشد از یک کارت ورودی که روی یک DP Slave مدولار قرار گرفته کل هشت بایت با آدرس شروع صفر را بخواند درینصورت در SFC14 در جلوی Record باید مقدار P#I0.0 byte 8 را قرار داد.باید توجه داشت که برای DP Slave های مدولار در هر Call فقط می توان ادرس یک مدول را تبادل کرد.

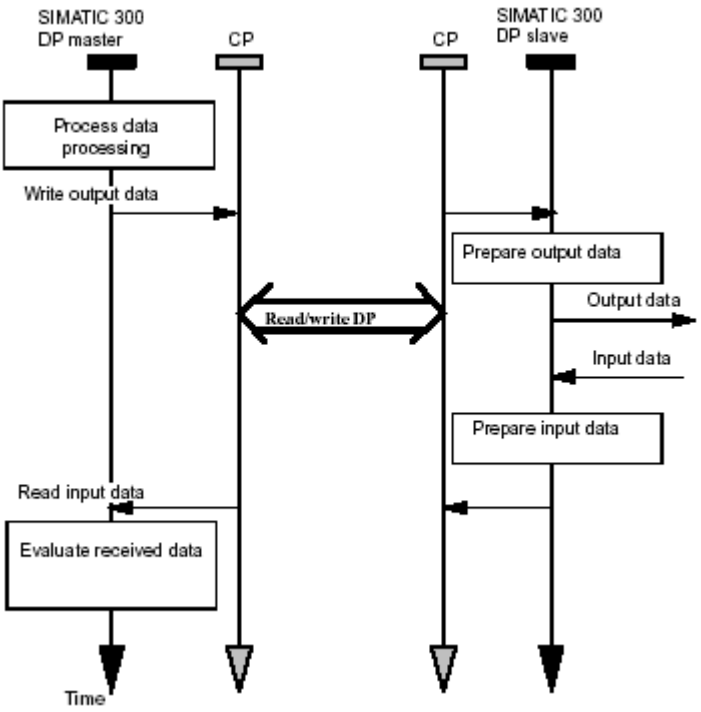
SFC های فوق الذکر علاوه بر DP Slave های معمولی برای Intelligent Slave ها نیز کاربرد دارند. که در قسمت بعد تشریح شده است.

۲-۷ سرویس DP برای Master / I-Slave

در فصل قبل ارتباط بین Master و I-Slave مورد بحث قرار گرفت و حالت های مختلف پیکر بندی آن بیان شد. در روشی که ذکر شد دیتا از یک CPU به CPU دیگر منتقل شده و در برنامه نویسی با دستورات معمول از آن استفاده میگردد. اما دو روش دیگر نیز برای تبادل دیتا در اینحالت وجود دارد . یک روش استفاده از فانکشنهای SFC است که در صفحه قبل آنها را برای ارتباط Master با DP-Slave توضیح دادیم. اصول کار در اینجا نیز مشابه است. تنها تفاوتی که وجود دارد اینست که این فانکشنها در دوطرف میتوانند Call شوند مثال زیر این موضوع را بهتر معرفی میکند.

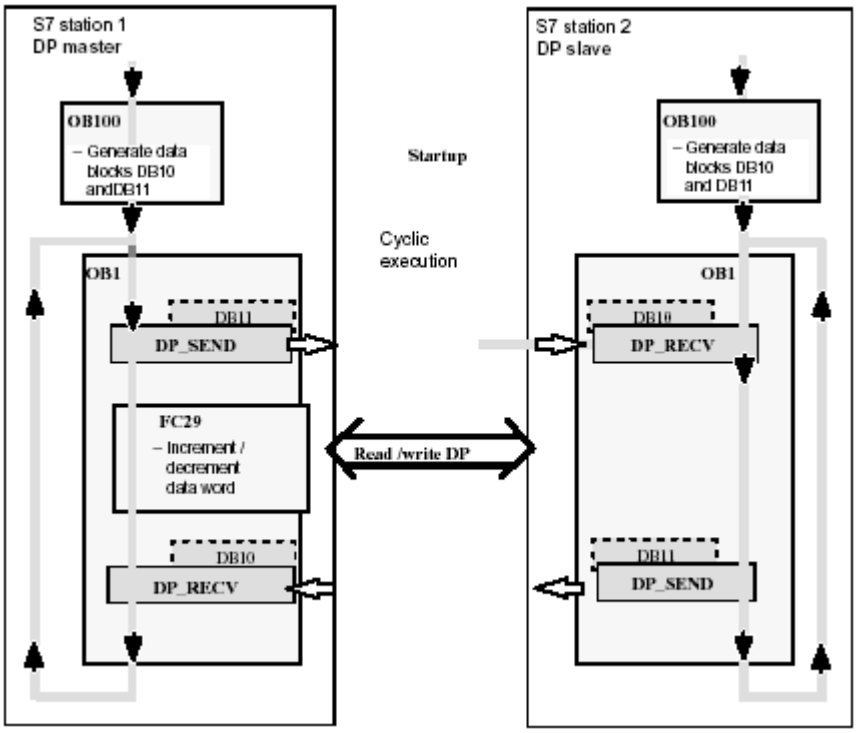
In the DP Slave CPU		In the DP Master CPU	
L 2	Data		
T MB 6	preprocessing in		
L IB 0	DP slave		
T MB 7			
L MW 6	Forward data		
T PQW 310	To DP master		
		L PIB 222	Postprocess
		T MB 50	receive data in
		L PIB 223	DP master
		L B#16#3	
		+ I	
		T MB 51	
		L 10	Data Processing
		+ 3	in DP master
		T MB 60	
		CALL SFC 15	Send data to DP
		LADDR:= W#16#0	slave
		RECORD:= P#M60.0	
		Byte20	
		RET_VAL:=MW 22	
CALL SFC 14	Receive data		
LADDR:=W#16#D	from DP master		
RET_VAL:=MW 20			
RECORD:=P#M30.0			
Byte20			
L MB 30	Postprocess		
L MB 7	receive data		
+ I			
T MW 100			

روش دیگری که میتوان از آن برای تبادل دیتا بین Master و I-Slave از آن استفاده نمود استفاده از فانکشن های خاص در برنامه نویسی است. این روش فقط برای ارتباط از طریق کارت CP بکار میرود. گامهایی که در این روش باید برداشته شود همراه با یک مثال توضیح داده میشود. در این مثال یک S7-300 با CPU314 بعنوان Master با کارت CP342-5 بکار میرود. I-Slave نیز یک S7-300 با همان CPU و همان کارت CP میباشد. روش پیکر بندی قبلاً به تفصیل بیان شد و نیازی به تکرار ندارد . فقط باید توجه داشت که Operating Mode برای Master روی کارت CP بعنوان DP Master انتخاب میشود. بدیهی است این تنظیم برای کارت CP روی I-Slave بصورت DP Slave خواهد بود. اصول کلی تبادل دیتا در شکل زیر نشان داده شده است.



بلوک دیاگرام برنامه در شکل زیر نمایش داده شده همانطور که ملاحظه میشود:

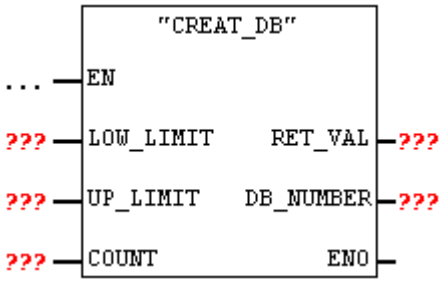
- در سمت Master و Slave پس از راه اندازی دیتا بلاکهای DB10 و DB11 توسط OB100 ایجاد میشوند.
- در سمت Master دیتا از DB11 توسط فانکشن DP_SEND به Slave ارسال میشود.
- Slave دیتای فوق را با فانکشن DP_RECV دریافت کرده و در DB10 مربوط به خود ذخیره میکند.
- در سمت Master فانکشن FC29 که توسط کاربر نوشته شده اجرا میشود. این فانکشن دیتای DB11 را هر سه ثانیه یکبار افزایش یا کاهش میدهد.
- در سمت Slave دیتا از DB11 توسط فانکشن DP_SEND به Master ارسال میشود.
- Master دیتای فوق را با فانکشن DP_RECV دریافت میکند و در DB11 مربوط به خود ذخیره مینماید.



برنامه بلاکهای فوق الذکر در صفحات بعد آمده است.

برنامه OB100 در سمت Master

OB100 برای Warm Restart بکار میرود که معمولاً برای تمام CPU های خانواده S7-300 بصورت پیش فرض بکار میرود. در برنامه Master با هر بار راه اندازی توسط OB100 دیتا بلاکهای DB10 و DB11 ایجاد میشوند. فانکشنی که کار ایجاد دیتا بلاک را انجام میدهد SFC22 نام دارد که نام سمبلیک آن CREAT_DB میباشد. باید توجه داشت این دیتا بلاکها در حافظه CPU ایجاد شده و مورد استفاده قرار میگیرند و در پروژه Offline ظاهر نخواهند شد.



ورودی های این فانکشن عبارتند از :

LOW_LIMIT شماره اولین دیتا بلاک که در مثال فوق عدد ۱۰ میباشد.

UP_LIMIT شماره آخرین دیتا بلاک که در مثال فوق عدد ۱۱ میباشد.

COUNT مقدار بایت رزرو شده برای دیتا بلاک که در مثال فوق ۳۰۰ منظور شده است.

خروجی های این فانکشن عبارتند از :

RET_VAL یک عدد صحیح برمیگرداند که نشان دهنده وجود یا عدم وجود خطا در اجراست.

DB_NUMBER یک word که شماره اولین دیتابلاک ایجاد شده را نشان میدهد.

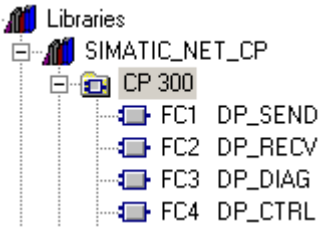
با توضیحات فوق برنامه OB100 بصورت زیر خواهد بود:

```
L    10
T    MW   20
L    11
T    MW   22
L    300
T    MW   24
CALL  SFC   22
 LOW_LIMIT:=MW20
 UP_LIMIT :=MW22
 COUNT    :=MW24
 RET_VAL  :=MW26
 DB_NUMBER:=MW28
  BE
```

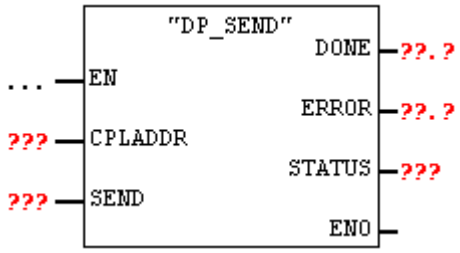
برنامه OB100 در سمت Slave

این برنامه نیز مشابه برنامه فوق میباشد.

فانکشن DP_SEND در سمت Master



فانکشن فوق که نام اصلی آن FC1 است را میتوان از مسیر روبرو در برنامه LAD/STL/FBD از پنجره Program Element وارد کرد و آنرا صدا زد. فانکشن DP_SEND مقدار مشخصی دیتا با آدرس مشخص شده را ارسال میکند. وقتی این فانکشن صدا زده میشود ورودی و خروجیهای آن مانند شکل زیر خواهند بود.



پارامتر CPLADDR

این پارامتر ورودی است و باید در جلوی آن آدرس بیس کارت CP سمت فرستنده را به فرمت Hex نوشت . این آدرس بصورت عدد صحیح در Hwconfig بعنوان آدرس شروع ورودی ذکر میشود .

Slot	Module	Order number	Fi...	MPI.	I address	Q ad
1	PS307 10A	6ES7 307-1KA00-0AA0				
2	CPU 314(1)	6ES7 314-1AE04-0AB0	V1.2	2		
3						
4	DO16xDC24V/0.5A	6ES7 322-1BH01-0AA0				0...1
5	CP 342-5(4)	6GK7 342-5DA02-0XE0	V5.0	3	272...287	272...

در شکل فوق این آدرس ۲۷۲ است که معادل هگز آن 0110 خواهد بود بنابراین در جلوی این ورودی باید نوشت:

CPLADDR:= W#16#0110

پارامتر SEND

این پارامتر آدرس شروع و میزان دیتایی که قرار است ارسال شود را مشخص میکند. بعنوان مثال برای ارسال ۱۰ بایت از آدرس صفر دیتا بلاک شماره ۱۱ مینویسیم:

SEND :=P#DB11.DBX0.0 BYTE 10

طول آدرس نباید از ۲۴۰ بایت بیشتر باشد.

پارامتر های DONE , ERROR , STATUS

این خروجی ها وضعیت ارسال دیتا را مطابق با حالات زیر نمایش می دهند:

حالت اول : ارسال بدون خطا کامل شده است

در این حالت وضعیت سه پارامتر فوق مطابق جدول زیر خواهد بود:

```
DONE   := 1
ERROR  := 0
STATUS := 0
```

حالت دوم : ارسال هنوز کامل نشده و Job فعال است

در این حالت وضعیت سه پارامتر فوق مطابق جدول زیر خواهد بود:

```
DONE   := 0
ERROR  := 0
STATUS := 8180
```

حالت سوم : ارسال بدلیل وجود اشکال قطع شده است

در این حالت وضعیت سه پارامتر فوق مطابق جدول زیر خواهد بود:

```
DONE   := 0
ERROR  := 1
STATUS := x
```

در اینجا کد x مقادیر متفاوتی را بسته به نوع اشکال برمیگرداند مثلاً کد 80D2 هگز نشان دهنده غلط بودن آدرس بیس کارت CP است فرضاً کارت CP را در HWcofigعوض کرده ولی آدرس آنرا تنظیم نکرده اید.
کدهای خطا در جدول صفحه بعد آمده است.

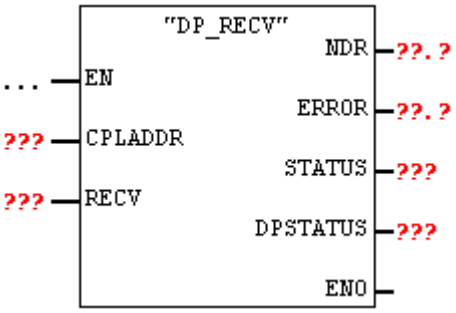
فانکشن DP_SEND در سمت Slave

با توضیحاتی که برای Master داده شد این فانکشن نیز نیاز به شرح اضافی ندارد. طول آدرسی که در جلوی پارامتر SEND نوشته میشود نباید از ۸۶ بایت بیشتر باشد.

DONE	ERROR	STATUS	Meaning
0	1	8183H	No configuration or the DP service has not yet started on the PROFIBUS CP
0	1	8184H	System error or bad parameter type.
0	1	8F22H	Area length error reading a parameter (e.g. DB too short).
0	1	8F23H	Area length error writing a parameter (e.g. DB too short).
0	1	8F24H	Area error reading a parameter.
0	1	8F25H	Area error writing a parameter.
0	1	8F28H	Alignment error reading a parameter.
0	1	8F29H	Alignment error writing a parameter.
0	1	8F30H	Parameter is in the write-protected 1st act. data block.
0	1	8F31H	Parameter is in the write-protected 2nd act. data block.
0	1	8F32H	Parameter contains a DB number that is too high.
0	1	8F33H	DB number error.
0	1	8F3AH	Destination area not loaded (DB).
0	1	8F42H	Timeout reading a parameter from the I/O area.
0	1	8F43H	Timeout writing a parameter to the I/O area.
0	1	8F44H	Address of the parameter to be read is disabled in the access track
0	1	8F45H	Address of the parameter to be written is disabled in the access track
0	1	8F7FH	Internal error, e.g. illegal ANY reference.
0	1	8090H	No module with this address exists.
0	1	8091H	Logical base address not at a double word boundary.
0	1	80A1H	Negative acknowledgment writing to the module.
0	1	80B0H	The module does not recognize the data record.
0	1	80B1H	The specified data record length is incorrect.
0	1	80C0H	The data record cannot be read.
0	1	80C1H	The specified data record is currently being processed.
0	1	80C2H	There are too many jobs pending.
0	1	80C3H	Resources occupied (memory).
0	1	80C4H	Communication error (occurs temporarily, it is usually best to repeat the job in the user program).
0	1	80D2H	Logical base address incorrect.

فانکشن DP_RECV در سمت Master

این فانکشن که نام اصلی آن FC2 است را میتوان به همان روشی که برای DP_SEND ذکر شد در برنامه صدا زد. پس از آن پارامترهای آن مانند شکل زیر ظاهر میگردند:



پارامتر CPLADDR : آدرس بیس کارت CP مشخص میکند.

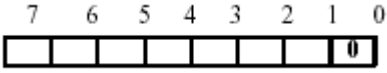
پارامتر RECV : آدرس و مقدار دیتایی که باید ذخیره شود را مشخص میکند. بعنوان مثال برای ذخیره سازی ۱۰ بایت در دیتا بلاک شماره ۱۰ مربوط به Slave با آدرس شروع صفر مینویسیم:

RECV :=P#DB10.DBX0.0 BYTE 10

ماکزیمم مقدار دیتا برای Slave برابر با ۲۴۰ بایت است

پارامترهای STATUS, ERROR , NDR : شبیه آنچه برای DP_SEND گفته شد عمل میکنند. NDR برای RECV مشابه DONE برای SEND میباشد. یعنی اگر 1 شد نشان میدهد که دریافت بدون خطا کامل شده است. در حالتی که ERROR یک شود STATUS کدخطا را طبق جدول صفحه بعد برمیگرداند.

پارامتر DPSTATUS : این پارامتر یک بایت است که وضعیت DP را مطابق جدول زیر برمیگرداند:



Bit	Meaning
7	not used
6	1: received data overflow
5,4	Values for DP STATUS of the DP master: 00 RUN 01 CLEAR 10 STOP 11 OFFLINE
3	1:Cyclic synchronization is active.
2	0: no new diagnostic data exist 1: evaluation of diagnostic list useful; at least one station has new diagnostic data
1	0: all DP slaves in the data transfer phase 1: evaluation of the station list useful
0	DP Mode 0: DP master mode

فانکشن DP_RECV در سمت Slave

این فانکشن نیاز به شرح اضافی ندارد و مشابه فانکشن DP_RECV سمت Master میباشد فقط طول آدرسی که در جلوی پارامتر SEND نوشته میشود نباید از ۸٦ بایت بیشتر باشد.

NDR	ERROR	STATUS	MEANING
0	1	8183H	No configuration or the DP service has not yet started on Profibus
0	1	8184H	System error or bad parameter type.
0	1	8F22H	Area length error reading a parameter (e.g. DB too short).
0	1	8F23H	Area length error writing a parameter (e.g. DB too short).
0	1	8F24H	Area error reading a parameter.
0	1	8F25H	Area error writing a parameter.
0	1	8F28H	Alignment error reading a parameter.
0	1	8F29H	Alignment error writing a parameter.
0	1	8F30H	Parameter is in the write-protected 1st act. data block.
0	1	8F31H	Parameter is in the write-protected 2nd act. data block.
0	1	8F32H	Parameter contains a DB number that is too high.
0	1	8F33H	DB number error.
0	1	8F3AH	Destination area not loaded (DB).
0	1	8F42H	Timeout reading a parameter from the I/O area.
0	1	8F43H	Timeout writing a parameter to the I/O area.
0	1	8F44H	Address of the parameter to be read is disabled in the access track.
0	1	8F45H	Address of the parameter to be read is disabled in the access track.
0	1	8F7FH	Internal error, e.g. illegal ANY reference.
0	1	8090H	No module with this address exists.
0	1	8091H	Logical base address not at a double word boundary.
0	1	80A0H	Negative acknowledgment writing to the module.
0	1	80B0H	The module does not recognize the data record.
0	1	80B1H	The specified data record length is incorrect.
0	1	80C0H	The data record cannot be read.
0	1	80C1H	The specified data record is currently being processed.
0	1	80C2H	There are too many jobs pending.
0	1	80C3H	Resources occupied (memory).
0	1	80C4H	Communication error (occurs temporarily, it is usually best to repeat the job in the user program)
0	1	80D2H	Logical base address incorrect.

با توضیحاتی که در مورد فانکشنهای DP_SEND و DP_RECV داده شد برنامه OB1 برای Master و Slave بصورت زیر خواهد بود .همانطور که مشخص است در برنامه Master از یک FC29 نیز استفاده شده که هر ۳ ثانیه یکبار مقدار دیتایی که قرار است ارسال شود را افزایش یا کاهش میدهد. برنامه FC29 در صفحه بعد آورده شده است.

OB1 in Master

```
CALL  "DP_SEND
 CPLADDR:=W#16#110
 SEND   :=P#DB11.DBX 0.0 BYTE 10
 DONE   :=M1.2
 ERROR  :=M1.3
 STATUS :=MW206

CALL  FC   29

CALL  "DP_RECV"
 CPLADDR :=W#16#110
 RECV    :=P#DB10.DBX 0.0 BYTE 10
 NDR     :=M1.0
 ERROR   :=M1.1
 STATUS  :=MW200
 DPSTATUS:=MB202
BE
```

OB1 in Slave

```
CALL  "DP_RECV"
 CPLADDR :=W#16#110
 RECV    :=P#DB10.DBX 0.0 BYTE 10
 NDR     :=M1.0
 ERROR   :=M1.1
 STATUS  :=MW200
 DPSTATUS:=MB202

L    DB10.DBW   0
T    DB11.DBW   0            // data for confirmation

CALL  "DP_SEND"
 CPLADDR:=W#16#110
 SEND   :=P#DB11.DBX 0.0 BYTE 10
 DONE   :=M1.2
 ERROR  :=M1.3
 STATUS :=MW104
BE
```

FC29 in Master

```
    A    T    4          // time downwards
    JC   runt
st_1: A   M    40.1      //Start Timer 3
    L    S5T#3S          // LED upwards 3 sec.
    SE   T    3
    A    T    3
    R    M    40.1
    JC   hoch             // Jump to upwards
    AN   T    3
    S    M    40.1
    S    M    40.2
runt: A   M    40.2      //Start Timer 4
    L    S5T#3S          // LED downwards 3 sec.
    SE   T    4
    A    T    4
    R    M    40.2
    AN   T    4
    S    M    40.2
    JC   st_1
    A    M    40.0
    L    S5T#100MS
    SE   T    1
    A    T    1
    R    M    40.0
    A    T    1
    JC   weit
    AN   T    1
    S    M    40.0
    L    MW   70
    L    0
    <>D
    JC   los1
    TAK
    +    1
    T    MW   70
    T    DB11.DBW   0
    BEU
los1: L   MW   70
    L    2
    *D
    T    MW   70
    T    DB11.DBW   0
    L    L#32768
    <>D
    BEC
    L    0
    T    MW   70
weit: BEU
```

```
hoch: A    M    40.3
    L    S5T#100MS
    SE   T    1
    A    T    1
    R    M    40.3
    A    T    1
    JC   weit
    AN   T    1
    S    M    40.3
    L    MW   70
    L    0
    <>D
    JC   los2
    TAK
    +    L#32768
    T    MW   70
    T    DB11.DBW   0
    BEU
los2: L   MW   70
    L    2
    /D
    T    MW   70
    T    DB11.DBW   0
    L    1
    <>D
    BEC
    L    0
    T    MW   70
    BE
```

فصل هشتم -پیکر بندی و برنامه نویسی ارتباط FDL در PROFIBUS

مشتمل بر :

۸-۱ ارتباط FDL بین PLC های S7

۸-۲ پیکر بندی ارتباط FDL بین PLC های S7 وS5

۸-۳ ارتباط FDL در Multiproject

۸-٤ ارتباط FDL با Unknown Project

مقدمه :

در بخش های قبل حالت های مختلف پیکر بندی و برنامه نویسی Master / Slave در Profibus تشریح شد, ولی DP تنها سرویس ارتباطی نیست سرویس های ارتباطی دیگری نیز توسط زیمنس برای PROFIBUS ارائه شده است شامل:

- سرویس FDL
- سرویس FMS
- سرویس S7 Function

این سرویسها برای ارتباط بین چند Master بکار میروند و از روش Token Pass استفاده میکنند.

در روش Master/Slave رابطه رئیس و مرئوس برقرار بود وهیچگاه مشکلی برای نحوه ارتباط ودر اختیار گرفتن باس احساس نمی شد. زیرا در آنجا این رئیس (Master) بود که همه ارتباط ها را مدیریت می کرد وتعیین می کرد که چه کسی باس را دراختیار بگیرد. ولی در ارتباط بین چندین Master دیگر نظام رئیس ومرئوس برای در اختیار گرفتن باس وجود ندارد مگر اینکه از طریق سرویس های ارتباطی شیوه Token Pass را پیاده سازی نماییم. البته از طریق توابعی که سیستم در اختیار کاربر قرار می دهد،نیازی نیست که به طور کامل در جریان پیاده سازی شیوه Token Pass قرار گیریم. STEP 7 خودش حلقه Token را پیاده سازی ومدیریت می کند و کافی است که کاربر توابع مربوط به انتقال دیتا را به نحو صحیح صدا بزند ودر برنامه هایش مورد استفاده قرار دهد.

در یک نگاه کلی به سرویسهای فوق میتوان گفت:

- سرویس FDL از لایه ۲ مدل OSI پیروی میکند.با فانکشنهای خاص تا ۲٤۰ بایت را جابجا میکند.
- سرویسFMS از لایه ۷ مدل OSI پیروی میکند . با فانکشنهای خاص حجم دیتای بیشتری را میتواند جابجا کند.
- سرویس S7 Function با استفاده از فانکشنهای سیستم تا ٦٤ بایت را جابجا میکند.
- برای استفاده ار این سرویسها لازم است کارت CP خاص در PLC نصب شود.
- روی یک شبکه PROFIBUS میتوان از سرویس های فوق بطور همزمان استفاده کرد. در جدول صفحه بعد سرویسهای ممکن برای سخت افزار های مختلف ارائه شده است.

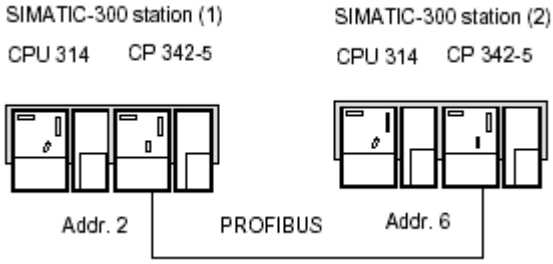
سرویس FDL در این بخش و دو سرویس دیگر در بخشهای بعدی مورد بحث قرار میگیرند.

System	Modules	Services
S5 95U	CPU 95U	FDL, DP (M or S)
S5115/135/155U	CP 5431	FMS, FDL, DP (M)
	IM 308-B/C	DP (M or S)
S7-200	CPU 215	DP (S)
S7-300	CPU 315-2 DP	DP (M or S)
	CP 342-5	S7 functions, FDL, DP (M or S)
	CP 343-5	S7 functions, FDL, FMS
S7-400	CPU 413-2 DP	DP (M)
	CPU 414-2 DP	DP (M)
	CPU 416-2 DP	DP (M)
	IM 467	DP (M or S), (M and S)
	CP 443-5 Basic	S7 functions, FDL, FMS
	CP 443-5 Extended	S7 functions, FDL, DP (M or S)
M7-300/400	IFM submodule	S7 functions, DP (M or S)
C7	CPU 626-DP	DP
OP	OP 5, OP 7, OP 15,	S7 functions
	OP 17; OP 25, OP 35,	S7 functions
	OP 37	S7 functions
PC/PG	CP 5613	S7 functions, FDL, FMS
	CP 5614	S7 functions, FDL, FMS
M = Master	S = Slave	

کارتهای شبکه Profibus و سرویس های آنها

۸-۱ ارتباط FDL بین PLC های S7

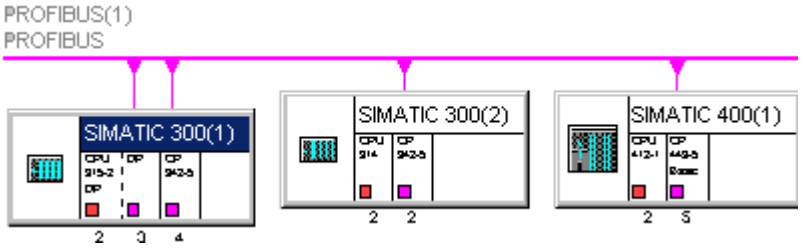
برای ایجاد ارتباط FDL باید از یک CP استفاده نماییم که ارتباط FDL را پشتیبانی کند. در مشخصات این CP باید مشخصه SEND/RECEIVE Interface وجود داشته باشد.مثلاً CP 342-5 این ویژگی را داراست.



با استفاده از این کارت گامهایی که به ترتیب باید برداشته شود عبارتند از :

گام اول:پیکر بندی با کارت CP

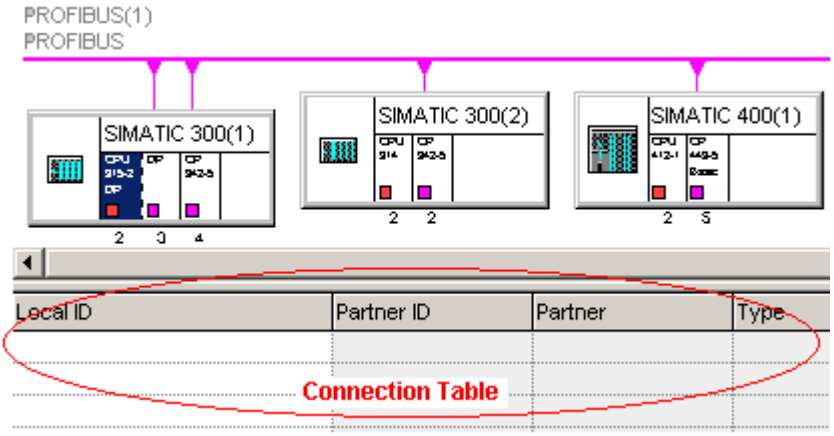
ابتدا چندین Station 300,400 با CP مناسب ایجاد نموده سپس با تنظیم مد کاری کارت CP همه این ایستگاهها را در مد Master پیکربندی مینماییم.نتیجه این پیکر بندی را می توان در برنامه NetPro نیز مشاهده کرد. توجه شود که در اینحالت اگر CPU دارای پورت DP باشد و به همین شبکه نیز متصل شود اشکالی ندارد. زیرا ارتباط FDL میتواند همزمان با DP روی یک شبکه بکار رود اگرچه ایندو مستقل عمل میکنند و با یکدیگر تداخل ندارند.



گام دوم: تشکیل Connection Table

جدول اتصالات یا Connection Table امکان اتصال چند Master را برای ارتباط Token Pass فراهم میسازد. در ارتباط DP حتی در مواردی که از کارت CP و فانکشن جهت تبادل دیتا استفاده میشد نیازی به ایجاد این جدول نبود ولی در ارتباط FDL و FMS وجود این جدول ضروری است.

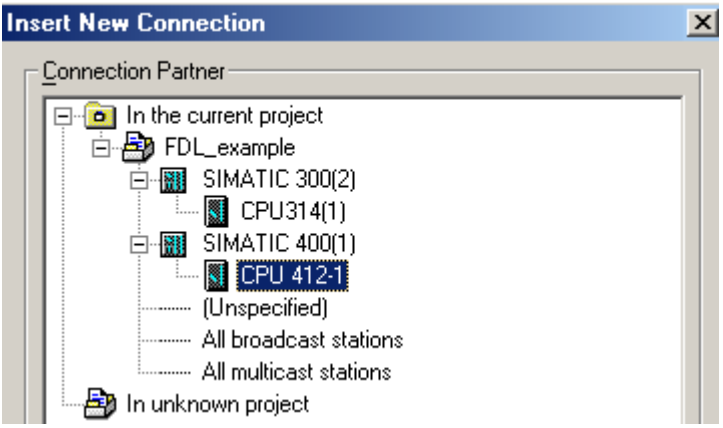
برای ایجاد جدول مزبور در برنامه NetPro در هر کدام از Station ها اگر بر روی قسمت CPU یکبار کلیک کنیم Connection Table ظاهر میشود مانند شکل :



توجه شود روی هر CPU کلیک کنیم یک جدول جداگانه برای آن ظاهر میشود . توسط این جدول میتوان ارتباطات هر CPU رامشاهده و پیکر بندی نمود.

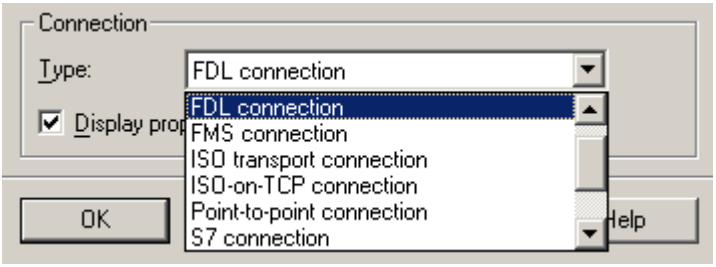
گام سوم: ایجاد ارتباط در Connection Table

در جدول فوق بر روی یکی از سطر ها راست کلیک نموده گزینه Insert New Connection را انتخاب کنید. در این حالت پنجره Insert New Connection باز می شود.



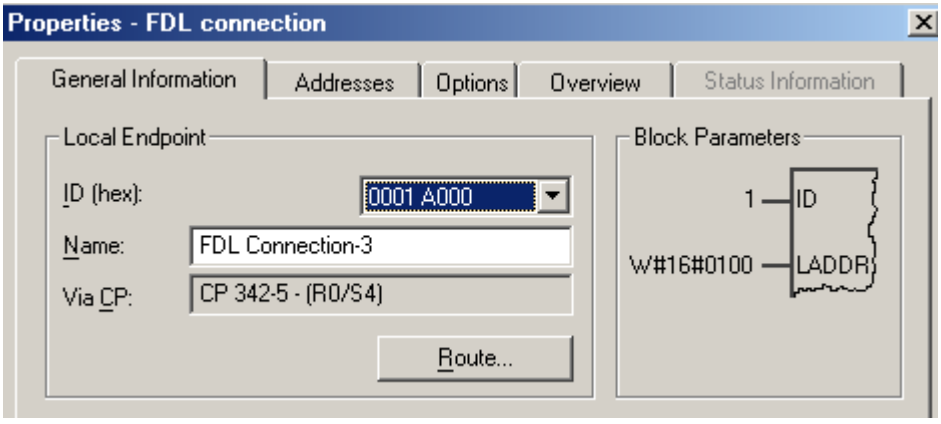
در این پنجره لیست Master هایی که میتوانند با این CPU ارتباط بگیرند قابل مشاهده است . در مثال فوق جدول Connection مربوط به CPU315 می باشد از اینرو در پنجره مزبور میتوان دو CPU دیگر یعنی CPU314 و CPU 412-1 را مشاهده و انتخاب کرد. به هر حال در این پنجره فقط یک مورد قابل انتخاب است.

نکته دیگر اینکه در پایین پنجره فوق باید مانند شکل از لیست Type ، گزینه FDL Connection انتخاب گردد.



پس از انتخاب موارد فوق و OK کردن پنجره مربوط به Property های ارتباط FDL باز می شود(مانند شکل زیر) STEP7 به صورت خودکار به این لینک ارتباطی یک ID تخصیص می دهد. این ID در شکل مقدار 1 دارد. همچنین یک نام هم به این ارتباط داده می شودکه در قسمت Name میتوان آنرا دید.اگر دراین Stationچندین CP وجود داشته باشد در قسمت Via CP باید مشخص کنیم که این ارتباط از طریق کدام CP برقرار می شود.

همچنین یک پارامتر دیگر به نام LADDR به این ارتباط تخصیص داده شده است. در واقع این آدرس ، آدرس شروع مدول CP است که در زمان پیکر بندی سخت افزار ،یعنی زمانی که CP را در Rack قرار داده اید،به آن نسبت داده شده است.دوپارامتر ID وLADDR بعداً در قسمت مربوط به برنامه نویسی این ارتباط به کار می آیند که در ادامه نحوه استفاده از آنها وبرنامه نویسی این ارتباط توضیح داده خواهد شد.



به این ترتیب پس از انجام مراحل فوق یک سطر در جدول Connection Table ظاهر میشود که لینک ارتباطی با ID و LADDR معین بین این CPU و CPU مورد نظر (که در این مثال CPU412-1) بود را مشخص میکند.

با این ارتباط همانطور که خواهیم دید میتوان توسط فانکشنهای خاص در برنامه نویسی تا ۲۴۰ بایت را از CPU315 به CPU412 ارسال کرد یا از آن دریافت نمود. باید توجه کرد بدون ایجاد این ارتباط نمیتوان صرفا با برنامه نویسی و فراخوانی فانکشنهای SEND و RECV به تبادل دیتا پرداخت. این یک تفاوت مهم بین FDL با DP است که با برنامه نویسی آن در بخش قبل آشنا شدید .

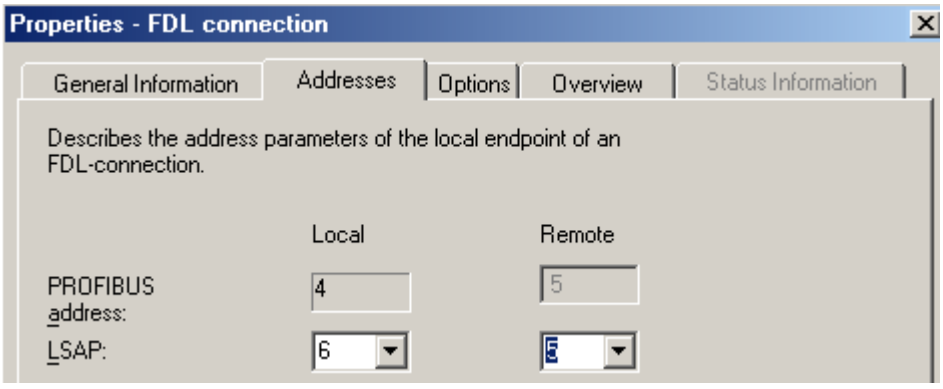
اگر روی سطر دوم جدول Connection راست کلیک کنیم مجدداً میتوانیم ارتباط جدیدی باز با همان CPU قبلی یا با CPU دیگری ایجاد کنیم . ارتباطی که با آن میتوان ۲۴۰ بایت دیگر را تبادل نمود. این اتصالات جدید مانند شکل زیردارای ID های متفاوتی خواهند بود.

در سمت ایستگاههای گیرنده نیز چنین جدولی را خواهیم داشت

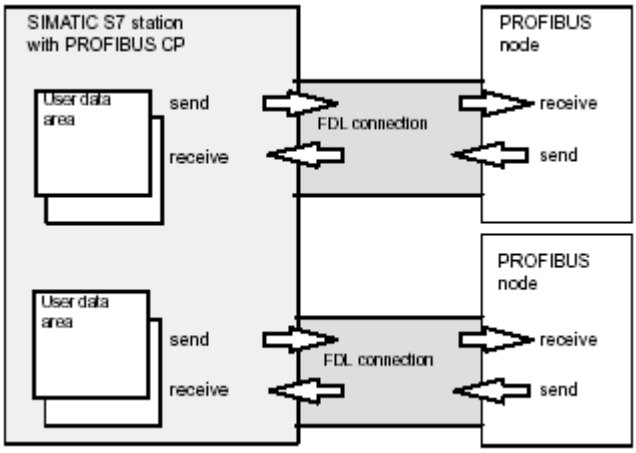
Local ID	Partner ID	Partner	Type
0001 A000	0001 A000	SIMATIC 400(1) / CPU 412-1	FDL connection
0002 A000	0002 A000	SIMATIC 400(1) / CPU 412-1	FDL connection
0003 A000	0001 A000	SIMATIC 300(2) / CPU314(1)	FDL connection

برای تغییر تنظیمات قبلی کافیست روی سطر مورد نظر کلیک کنیم تا پنجره Properties مانند شکل پایین صفحه قبل باز شود. در عمل تغییری لازم نیست در تنظیمات پیش فرض بدهیم. فقط لازم است مقادیر ID و LADDR را یادداشت کنیم تا در برنامه نویسی بکار ببریم.

بخشهای دیگر پنجره Properties اطلاعات دیگری مربوط به ارتباط را نمایش میدهند. مثلاً در بخش Address میتوان آدرس Node های دو ایستگاه را همراه با LSAP مشاهده نمود.



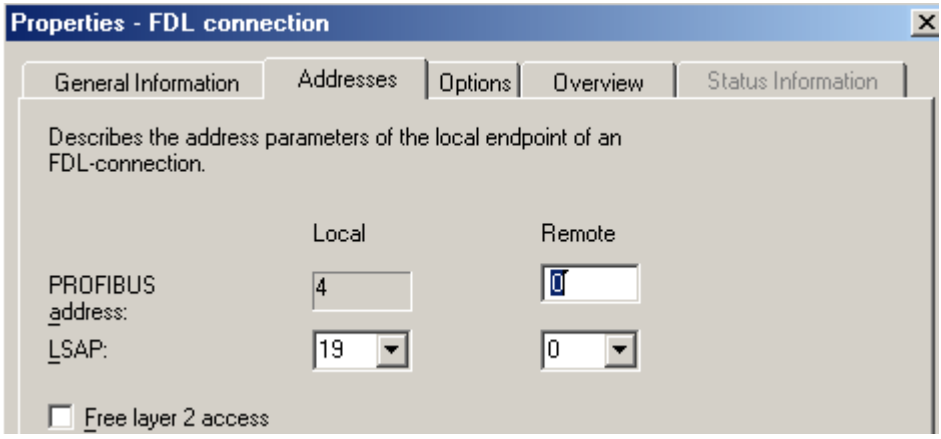
آنچه تاکنون ذکر شد ارتباط بین یک ایستگاه با ایستگاه مشخص دیگر بود که اصطلاحاً **Specified Station** نامیده میشود. شکل کلی این نوع ارتباط بصورت زیر است:



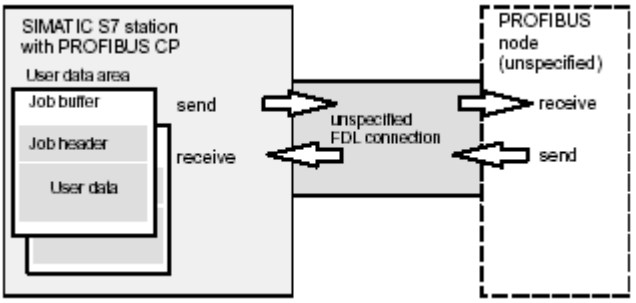
علاوه بر این نوع ارتباط انواع دیگری نیز در پنجره New Connection مشاهده میشوند که عبارتند از :

Unspecified Station

در اینحالت ایستگاه مقابل که قرار است با آن ارتباط برقرار شود در لیست ظاهر نمیشود . مثلاً یک نود Remote است که فقط آدرس Node آن معلوم است. در اینجا باید آدرس مزبور را در زیر Remote از بخش آدرس مانند شکل وارد کرد. این آدرس در حالت قبل غیر فعال بود.

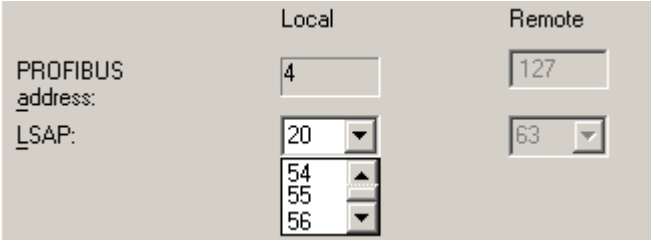


شکل کلی ارتباط Unspecified بصورت زیر است. در این ارتباط ماکزیمم ۲۳٦ بایت میتوان ارسال کرد یعنی ٤ بایت کمتر از نوع Specified. در واقع این ٤ بایت برای اطلاعات اضافی Job header که آدرس Local را مشخص میکند رزرو میگردد.

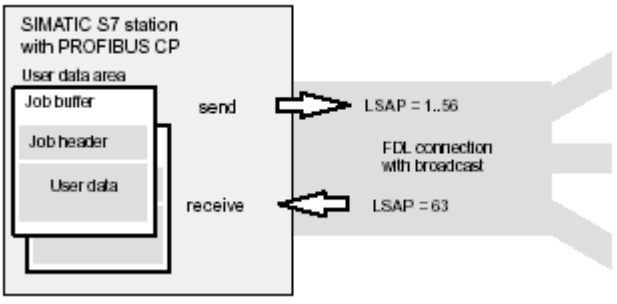


Broadcast Stations

در اینحالت دیتا از فرستنده همزمان به تمام ایستگاههایی که روی باس قرار دارند ارسال میشود این سرویس همانطور که میدانیم SDN است یعنی تاییدیه دریافت از گیرنده به فرستنده ارسال نمیشود.

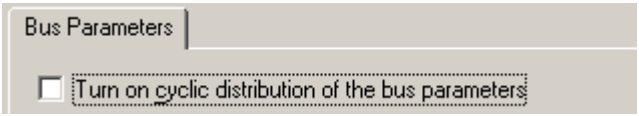


در این حالت دیتا ماکزیمم ۲۳٦ بایت است. آدرس نودهای گیرنده و LSAP آنها نیز در پنجره غیر فعال است. توجه شود که در Connection Table فقط یک ارتباط Broadcast برای هر ایستگاه میتوانیم داشته باشیم.



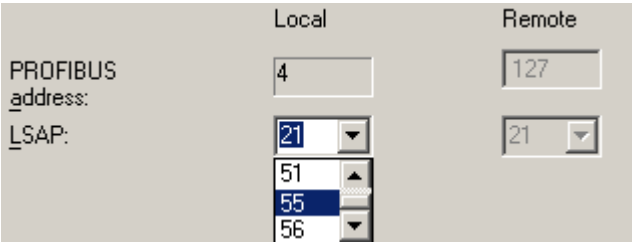
نکته دیگری که در ارتباط Broadcast باید به آن توجه کردانیست که در پارامترهای مربوط به Properties

شبکه PROFIBUS در بخش Network Setting انتخابی مانند شکل زیر وجود دارد که بطور پیش فرض غیر فعال است.اگر فعال شود مرتبا پارامترهای باس در هر سیکل بصورت Broadcast به تمام ایستگاهها ارسال میگردد. برای جلوگیری از ایجاد تداخل بین این بسته ها و بسته هایی که توسط Connection Table بصورت Broadcast ارسال میشوند بهتر است گزینه فوق فعال نشود یا در برنامه بسته هایی با LSAP > 56 را در نظر نگیرد.

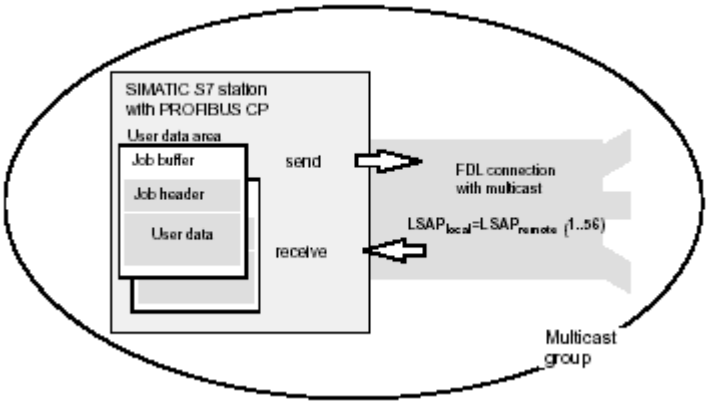


Multicast Stations

در اینحالت دیتا از فرستنده همزمان به گروهی از ایستگاهها که روی باس قرار دارند ارسال میشود این سرویس نیز بصورت SDN است .مقدار دیتا ۲۳٦ بایت است . برای مشخص کردن گروه Multicast لازم است آدرس LSAP برای Local و تمام Remote ها همگی یکسان باشد مثلادر شکل زیر این آدرس برای تمام اعضای گروه باید 21 باشد. به عبارت دیگر برای یکایک CPU های مورد نظر یک ارتباط Multicast تعریف کرده و آدرس همه آنها را 21 انتخاب میکنیم.



در پنجره فوق آدرس برای Local میتواند ۱ تا ٥٦ باشد. هر آدرسی برای Local انتخاب شود میبینیم که بطور اتوماتیک آدرس Remote نیز همان مقدار میشود.



تا این مرحله گامهایی که برداشته شد برای تعیین اتصالات در جدول Connection Table بود . پس از اتمام کار لازم است:

- چک سازگاری از منوی Network در Netpro اجرا شده و Error های احتمالی رفع شوند.
- هر کدام از جداول Connection به CPU مربوطه Download شوند.

تذکر :

اگر بعد از پیکر بندی Connection Table در سخت افزار تغییر بدهیم چه اتفاقی می افتد؟ بعنوان مثال اگر کارت CP را با نوع دیگر تعویض کنیم. در اینصورت

- Connection Table باقی میماند.
- ID بطور اتوماتیک Update میشود.
- LADDR را باید با توجه به آدرس بیس کارت تغییر داد.
- و نهایتاً باید Connection Table را مجدداً Download کرد.

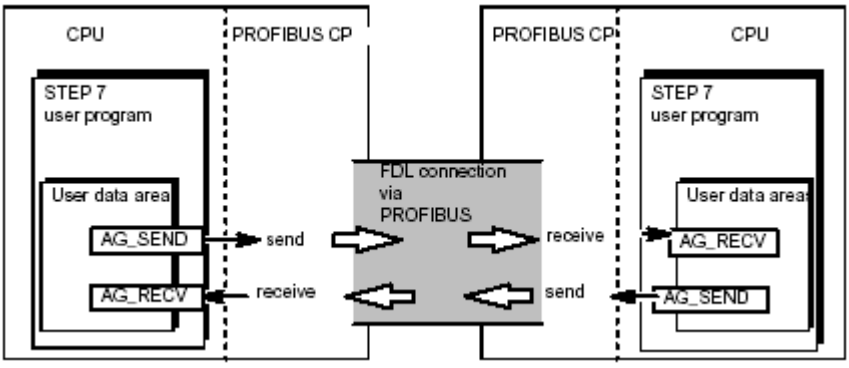
گام سوم : برنامه نویسی سرویس FDL

برای برنامه نویسی FDL باید دو تابع یکی برای فرستادن اطلاعات (SEND) در سمت ایستگاه فرستنده ویکی برای گرفتن اطلاعات (RECIVE) در سمت ایستگاه گیرنده در برنامه های کاربر،قرار گیرد.

در قسمت های قبل یک لینک ارتباطی FDL بین ایستگاه SIMATIC 300(1) و(1)400 SIMATIC برقرار شد اکنون از طریق آن می خواهیم یک سری اطلاعات را مبادله کنیم.توجه اینکه درConnection Table مربوط به این شبکه می توانستیم لینک های ارتباطی دیگری هم بین Station های دیگر داشته باشیم که در این صورت هر لینک ارتباطی ،یک برنامه نویسی مجزا در هر دوسمت فرستنده وگیرنده نیاز دارد.

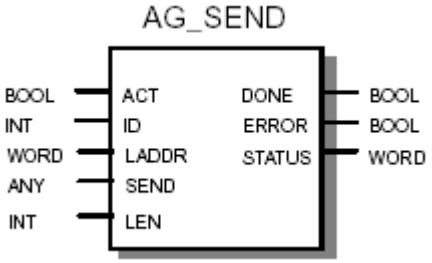
نکات قابل توجه:

- برای فرستادن اطلاعات از تابعی به نام AG_SEND و برای دریافت آن از تابع AG_RECV استفاده می شود .
- لزومی ندارد که این فانکشن ها حتماً در OB1 صدا زده شوند میتوان آنها را در سایر OB ها یا بلاکها صدا زد.
- برای تبادل سیکلی بهترین روش استفاده از OB های Cyclic Interrupt است.
- زمانی که صرف میشود دیتا از طریق کارت CP و شبکه ارسال و جواب دریافت شود جدا از سیکل اسکن است و به Transmission Time معروف است.
- از آنجا که سیکل اسکن سریعتر از Transmission Time است اگر قبل از تکمیل ارسال قبلی مجددا عمل ارسال اتفاق بیفتد دیتای جدید ارسال نمیشود تا ارسال قبلی کامل شود.
- میتوان در بلاک OB چندین بار از فانکشنهای فوق استفاده کرد و هربار دیتا را به آدرس جداگانه ای انتقال داد.
- فانکشنهای فوق شبیه DP_SEND و DP_RECV در زیر مجموعه Libraries>Simatic Net در پنجره Program Element برنامه LAD/STL/FBD قرار دارند و از این مسیر باید Drag شده و در برنامه قرار گیرند.
- برای S7-300 و S7-400 فانکشنهای مزبور متفاوتند و در Folder های جداگانه در آدرس فوق قرار دارند.



فانکشن AG_SEND

این فانکشن که نام غیر سمبلیک آن FC5 است برای ارسال دیتا استفاده میشود. شکل زیر این فانکشن را بصورت بلاک FBD همراه با ورودی و خروجی هایش نشان میدهد.



ورودی ها :

ACT	اگر ACT=1 باشد LEN بایت ارسال میشود. اگر ACT=0 باشد هر سه خروجی update میشوند.
ID	عدد صحیح که در Connection Table مشخص شده است.
LADDR	کد هگز که در Connection Table مشخص شده است.
SEND	آدرس حافظه و مقدار دیتایی که باید ارسال شود را مشخص میکند. این آدرس مربوط به حافظه فرستنده است و میتواند I و Q و M و DB باشد. مانند مثالهای زیر : • P#DB1.DBX0.0 BYTE 240 • P#M0.0 BYTE 100 • P#I0.0 BYTE 200
LEN	مقدار بایتی که باید ارسال شود را مشخص میکند و میتواند بین 1 تا 240 باشد.

خروجی ها :

سه خروجی این فانکشن وضعیت انتقال سیگنال را نشان میدهند مطابق جدول زیر

	STATUS	ERROR	DONE
ارسال بدون خطا کامل شده است	0	0	1
Job هنوز فعال است	8181	0	0
Job در حال اجرا نیست (دیتایی ارسال نشده است)	0000	0	0
قطع شدن ارسال بدلیل وجود خطا با کد x (توضیح در صفحه بعد)	x	1	0

DONE	ERROR	STATUS	Meaning
0	1	7000H	This code is only possible with an S7-400; the FC was called with ACT=0; the job is however not processed
0	1	8183H	No configuration or the FDL service has not yet started on the PROFIBUS CP.
0	1	8184H	Illegal data type specified for the SEND parameter. FDL connection **without** job buffer: system error. FDL connection **with** job buffer: parameter LEN<4 or illegal job header
0	1	8185H	LEN parameter longer than SEND source area.
0	1	8186H	ID parameter invalid. ID!=1,2 to 15,16.
0	1	8301H	SAP not activated on destination station.
0	1	8302H	No receive resources on the destination station; the receiving station cannot process received data quickly enough or has not prepared any receive resources.
0	1	8303H	The PROFIBUS service (SDA **S**end **D**ata with **A**cknowledge) is not supported on this SAP of the destination station.
0	1	8304H	The FDL connection is not established.
0	1	8311H	The destination station is not obtainable at the specified PROFIBUS address or the service is not possible for the specified PROFIBUS address.
0	1	8312H	PROFIBUS error on the CP; for example bus short circuit, local station not in the ring.
0	1	8315H	Internal parameter error on an FDL connection with job header: Parameter LEN<4 or illegal parameter in the job header (with free layer 2 access).
0	1	8F22H	Source area invalid, for example: Area does not exist in the DB LEN parameter < 0
0	1	8F24H	Area error reading a parameter.
0	1	8F28H	Alignment error reading a parameter.
0	1	8F32H	Parameter contains a DB number that is too high.
0	1	8F33H	DB number error.
0	1	8F3AH	Area not loaded (DB).
0	1	8F42H	Timeout reading a parameter from the I/O area.
0	1	8F44H	Address of the parameter to be read is disabled in the access track
0	1	8F7FH	Internal error, e.g. illegal ANY reference. e.g. parameter LEN=0
0	1	8090H	No module with this base address exists. The FC being used does not match the system family being used (remember to use different FC for S7-300 and S7-400).
0	1	8091H	Logical base address not at a double word boundary.
0	1	8092H	In the ANY reference, a type other than BYTE is specified. (S7-400 only)
0	1	80A4H	The communication bus connection between the CPU and CP is not established
0	1	80B0H	The module does not recognize the data record.
0	1	80B1H	Destination area invalid. for example, destination area > 240 bytes.
0	1	80B2H	The communication bus connection between the CPU and CP is not established
0	1	80C0H	The data record cannot be read.
0	1	80C1H	The specified data record is currently being processed.
0	1	80C2H	There are too many jobs pending.
0	1	80C3H	Resources occupied (memory).
0	1	80C4H	Communication error (occurs temporarily, it is usually best to repeat the job)
0	1	80D2H	Module base address incorrect

مثال : می خواهیم ۲۴۰ بایت از آدرس شروع DB 100 در SIMATIC 300(2) به ایستگاه SIMATIC 400(1) ارسال کنیم .همانطور که قبلاً دیدید لینک ارتباطی که بین دو ایستگاه مورد نظر برقرار کردیم به مشخصات ID = 1 و LADDR = W#16#0100 بود این دو پارامتر در برنامه نویسی مربوط به این لینک ارتباطی ،به کار می آیند.
ساده ترین فرم ممکن برای برنامه نویسی مربوط به این ارتباط به صورت زیر است.این برنامه را در قسمت OB 1 در SIMATIC 300(2) می نویسیم.

```
CALL  AG-SEND
  ACT: = M0.0
  ID: = 1
  LADDR: =W#16#0100
  SEND: = P#DB100.DBX0  BYTE  240
  LEN: =240
  DONE: =M0.1
  ERROR: =M0.2
  STATUS: =MW20
```

این که چرا ACT را مستقیماً مقدار قرار نداده ایم به این دلیل است که می خواهیم ارسال اطلاعات را کنترل کنیم.اگر ACT را 1 بگذاریم چون برنامه به صورت سیکلی اجرا می شود هر بار که به خط ارسال اطلاعات برسد ،صرف نظر از اینکه ارسال قبلی در چه وضعیتی قرار دارد وآیا دیتای قبلی به سلامت رسیده است یانه، دیتای جدید را ارسال می کند.در حالیکه این امکان فراهم است که با بررسی مقادیر DONE و ERROR و STATUS به صورت هوشمندانه ای،اطلاعات را ارسال کنیم.
به این ترتیب که در ابتدای کار یک بار اطلاعات را می فرستیم .در واقع در OB 100 که OB راه اندازی است مقدار ACT را 1 قرار می دهیم.به این ترتیب زمانی که CPU به OB1 می رسد چون ACT یک است ،یک بار ارسال را انجام می دهد.پس از ارسال ، مقدار ACT را صفر می کنیم وشروع به ارزیابی ارسال اطلاعات می کنیم اگر مقدار DONE برابر 1 باشد که ارسال به طور کامل انجام شده ،بنابراین مجدداً ACT را 1 می کنیم تا ارسال انجام شود ولی اگرمقدار DONE ،1 نبود و ERROR یک شود یعنی در ارسال اطلاعات خطایی رخ داده است می توانیم یک TAG را یک کنیم تا به سیستم مونیتورینگ خبر داده شودیا اینکه می توانیم ACT را یک کنیم و به انتهای برنامه Jump کنیم به این ترتیب وقتی CPU به ابتدای سیکل OB1 می رسد مجدداً همان اطلاعات را ارسال می کند وخلاصه اینکه هر تدبیر دیگری متناسب با خطا رخ داده می توانید در نظر بگیرید وپیاده سازی نمایید.در اینجا اگر خطایی در ارسال یک بسته دیتا رخ دهد ،می خواهیم خانه حافظه M100.0 یک شود. اگر DONE صفر

باشد وERROR هم صفر باشد،در واقع سیستم در حال ارسال اطلاعات است، پس کار خاصی نباید انجام دهیم و ACT باید مقدار صفر داشته باشد واین وضعیت ادامه می یابد تا اینکه وضعیت ارسال اطلاعات مشخص شود.
برنامه فوق به صورت زیر پیاده سازی شده است:

```
OB100:
        SET
        =M0.0
```

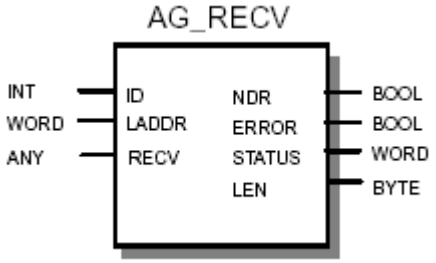
```
OB1:
        CALL  AG_SEND
        ACT:=M0.0
        ID:=1
        LADDR:=W#16#0100
        SEND:=P#DB100.DBX0.0 BYTE 240
        LEN:=240
        DONE:=M0.1
        ERROR:=M0.2
        STATUS:=MW20

        R M0.0
        SET
        A M0.1
        JC  A001
        SET
        A M0.2
        JC  A002
        JU   NEXT
A001:  S M0.0
        JU NEXT
A002:  S M100.0
NEXT :
        ادامه برنامه مورد نظر
        BE
```

در این روش پس از اتمام کار باید تمام بلاکها را به PLC دانلود کرد. پس لازم است دیتا بلاک نیز قبلاً ایجاد شده باشد تا بتوان آنرا دانلود کرد. از آنجا که ایجاد دیتا بلاک و معرفی ۲۴۰ بایت آدرس در آن کمی وقت گیر است میتوان با روشی که در فصل قبل برای برنامه نویسی DP گفته شد توسط SFC22 بطور اتوماتیک آنرا در هنگام راه اندازی (یعنی توسط OB100) ایجاد نمود.

فانکشن AG_RECV

این فانکشن که نام غیر سمبلیک آن FC6 است برای دریافت دیتا استفاده میشود. شکل زیر این فانکشن را بصورت بلاک FBD همراه با ورودی و خروجی هایش نشان میدهد.



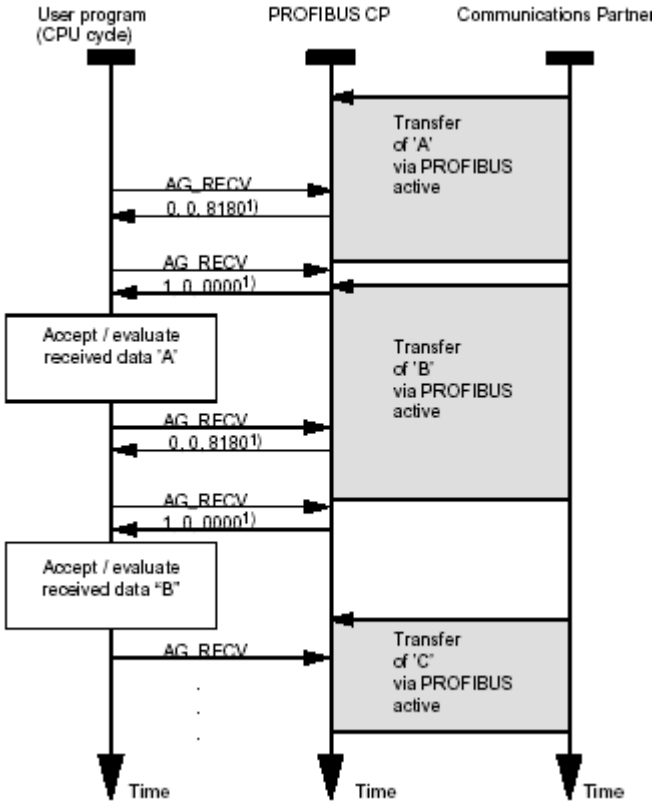
با توضیحاتی که در مورد AG_SEND داده شد بسیاری از پارامترهای AG_RECV نیز مشخص است.فقط نکات زیر قابل توجه است:

- در اینجا نیز ID وLADDR همان معانی بخش قبل را می دهد ولی برای Connection سمت گیرنده.
- در AG-RECV پارامتری به اسم ACT نداریم. در واقع این تابع همیشه منتظر دریافت است نیازی به فعال کردن ندارد.
- RECV آدرس محل ذخیره وطول دیتایی که قرار است دریافت شود را بیان می کند. فرمت آن شبیه SEND است
- LEN تعداد بایت های دریافتی را مشخص می کند.
- NDR همانند DONE در AG_SEND میباشد. اگر ۱ باشد یعنی دریافت کامل وبدون خطا انجام شده است واگر صفر باشد، یا سیستم در حال دریافت اطلاعات است ویا دریافت اطلاعات با مشکل مواجه شده است
- اگر NDR صفر باشد و ERROR یک باشد یعنی در دریافت اطلاعات خطایی رخ داده است که خطای رخ داده از مقدار STATUS قابل استخراج است.

مثال : برنامه ای در سمت گیرنده یعنی ایستگاه SIMATIC 400(1) بنویسید که ۲٤۰ بایتی را که در قسمت قبل ارسال کرده ایم را دریافت نموده ودر DB 20 بریزد.

```
CALL AG-RECV
     ACT: = M0.0
     ID: = 1
     LADDR: =W#16#0100
     SEND: = P#DB20.DBX0.0 BYTE 240
     NDR: =M0.0
     ERROR: =M0.1
     STATUS: =MW100
     LEN=240
```

شکل زیر نتیجه بحث را برای ارسال و دریافت دیتا توسط ارتباط FDL را نشان می دهد.

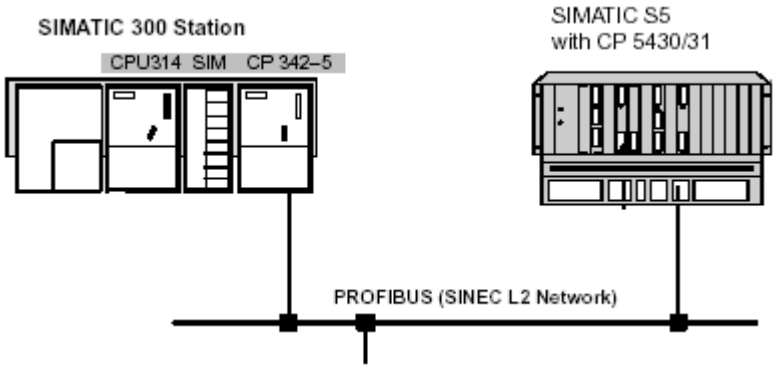


۸-۲ پیکر بندی ارتباط FDL بین PLC های S7 وS5

گام اول : پیکر بندی سخت افزار سیستم

در یک پروژه جدید یک Station 300 ویک Station S5 از نوع DP Master ایجاد کنید.در SIMATIC MANAGER برای استفاده از یک Station S5 در شبکه PROFIBUS ،باید این ایستگاه را در زمان پیکر بندی در نظر بگیریم و در پیکر بندی شبکه لحاظ کنیم. البته جزئیات مربوط به S5 توسط نرم افزار STEP5 صورت می گیرد ودر STEP7 تنها آن بخش از پیکر بندی S5 که مربوط به شبکه PROFIBUS است انجام می شود.

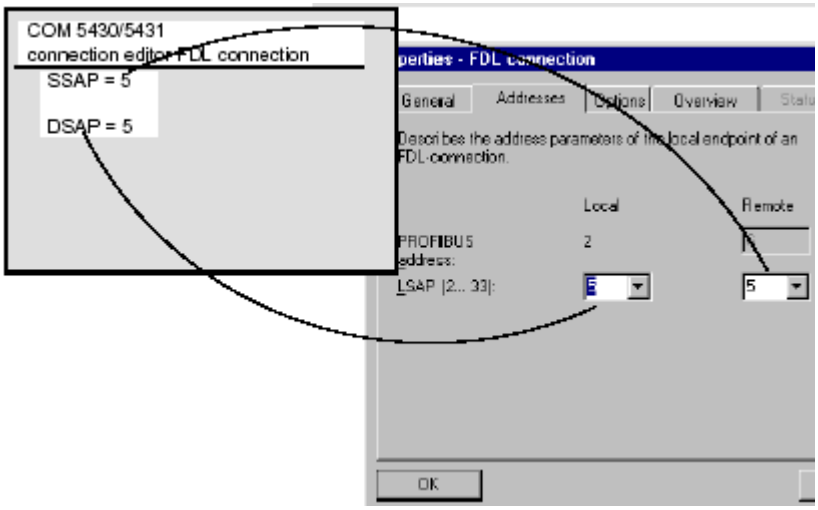
برای ایجاد یک Station S5 در صفحه اصلی SIMATIC MANAGER بر روی نام پروژه راست کلیک نموده واز قسمت Insert New Object گزینه SIMATIC S5 را انتخاب نمایید. در صفحه سمت راست برروی SIMATIC S5(1) راست کلیک نمایید ودر قسمت Interface ، کلید New را فشار دهید. یک صفحه باز می شود وشبکه هایی که توسط S5 پشتیبانی می شوند ، نمایش داده می شوند.گزینه PROFIBUS را انتخاب نموده وOk کنید. یک پنجره باز می شود ،New را بزنید تا یک شبکه PROFIBUS به نام PROFIBUS(1) ایجاد شود.



برای Station 300 یک CP 300 با قابلیت SEND/RECEIVE انتخاب نمایید.مثلاً CPU 314 و CP 342-5 را انتخاب نمایید وآن را به شبکه PROFIBUS(1) که در قسمت قبل ایجاد کردید اضافه نمایید.یادآوری اینکه برای اتصال CP به شبکه PROFIBUS(1) بر روی آن دابل کلیک نمایید ودر قسمت Operating Mode گزینه DP Master را انتخاب نمایید وسپس Save کنید .نتیجه کار را در برنامه NetPro هم ببینید.

گام دوم: ایجاد Connection Table

مانند قبل باید یک لینک FDL بین این دو Station ایجاد نمایید. بر روی CPU 300 کلیک نمایید تا در پایین صفحه جدول Connection Table مشابه قبل به وجود بیاید. بر روی یکی از سطر های این جدول دابل کلیک نمایید مانند قسمت قبل در قسمتType گزینه FDL Connection را انتخاب نمایید وسپس OK کنید تا ID و LADDR مربوط به این لینک ارتباطی را مشاهده نمایید. در ضمن در این پنجره در قسمت Address دو پارامتر LSAP Docal و LSAP Remote را یادداشت نمایید که در بخش بعد مورد استفاده قرار خواهند گرفت.



گام سوم : برنامه نویسی FDL مربوط به ارتباط S7 و S5

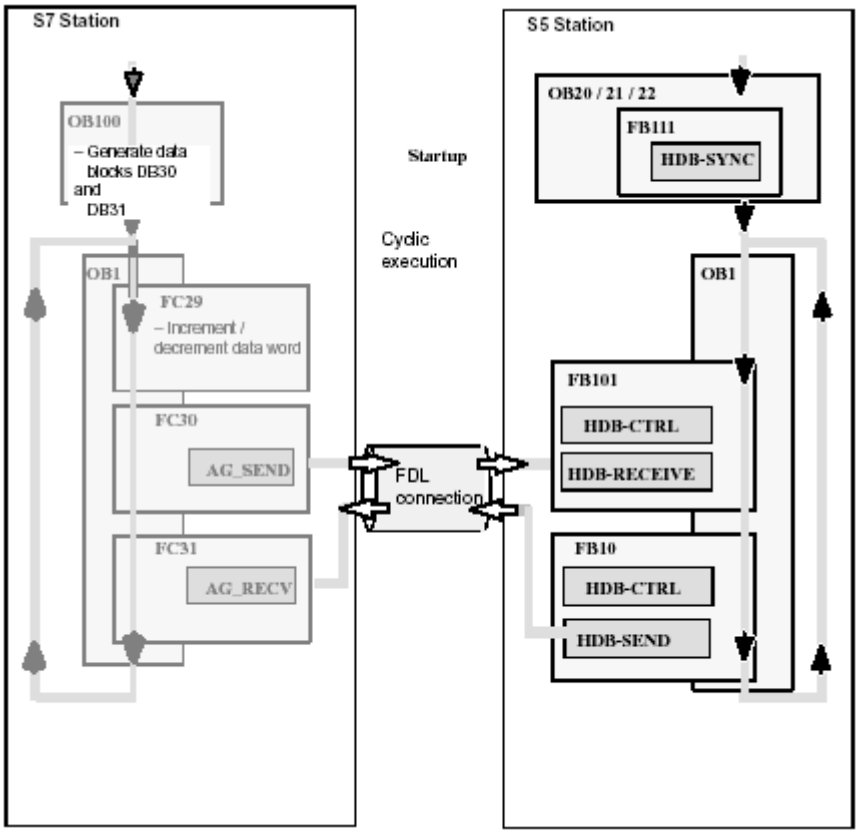
همانند قبل باید دردوسوی این لینک ارتباطی یعنی هم در سمت S7 وهم در سمت S5 برنامه مربوط به دریافت و ارسال اطلاعات را بنویسید. در سمت S7 که همان توابع AG_SEND و AG_RECV استفاده می شوند.اما در سمت S5 از توابع HDB-RECEIVE و HDB-SEND باید استفاده کرد. در S5 Station یک CPU115 U ویک CP که قابلیت پشتیبانی SEND/RECEIVE داشته باشد به کار می بریم.

حال در برنامه STEP 5 برنامهCom PROFIBUS را اجرا نمایید وتنظیمات مربوط به اتصال Station S5 به شبکه را انجام دهید و به CPU مربوطه Download نمایید.تذکر اینکه این تنظیمات از جمله سرعت انتقال، آدرس شبکه و.. باید همان مقادیری را داشته باشند که در برنامه STEP 7 به این Station نسبت داده بودیم.در ضمن در این برنامه پارامترهای LSAP Remote و LSAP Local به نام های SSAP و DSAP شناخته می شوند لذا:

DSAP(S5) = LSAP Local (S7)
SSAP(S5) = LSAP Remote(S7)

در مورد برنامه نویسی مربوط به Station S5 نیز همانطور که گفتیم از توابع HDB-RECEIVE با نام غیر سمبلیک FB101 وHDB-SEND با نام غیر سمبلیک FB10 در برنامه STEP5 استفاده می کنیم.

شکل زیر مثالی را برای ارتباط بین S7 و S5 نشان میدهد.



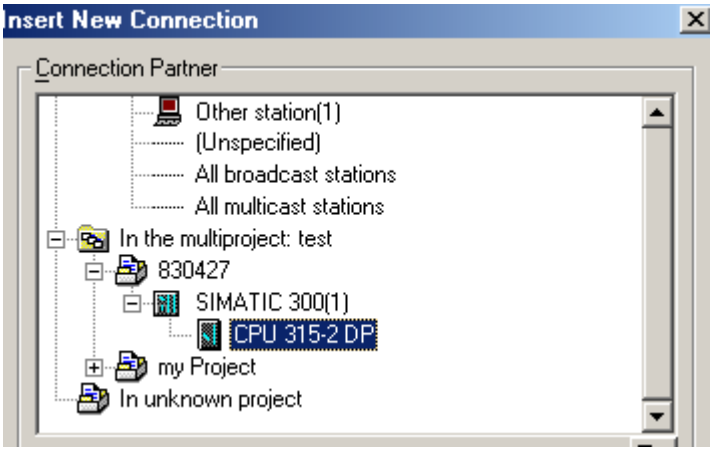
۸-۳ ارتباط FDL در Multiproject

آنچه تا اینجا در مورد ارتباط FDL گفته شد مربوط به Station های داخل یک پروژه بود. میتوان ارتباط فوق را بین پروژه های مختلف که بصورت Multiproject هستند تعریف نمود. برای اینکار :

گام اول : به روشی که در بخش قبل توضیح داده شد یک Multiproject شامل چند پروژه که همگی دارای ایستگاههای Master و به Profibus متصل هستند ایجاد میکنیم.

گام دوم : در ایستگاههای Master کارت CP مناسب برای ارتباط FDL قرار می دهیم

گام سوم : در برنامه Netpro برای یکی از ایستگاهها روی سطر مربوط به Connection Table کلیک کرده تا پنجره زیر باز شود.



گام چهارم : از قسمت In the multiproject استفاده کرده و از زیر مجموعه آن پروژه مورد نظر را انتخاب کرده سپس در آن پروژه Station مورد نظر را انتخاب می نماییم.

گام پنجم: پنجره مربوط به Properties باز میشود . شبیه روال قبل ID و LADRR را در صورت لزوم تنظیم سپس یادداشت میکنیم.

گام ششم : اکنوان اگر در NetPro پروژه مورد نظر (در مثال فوق پروژه 830427) را باز کنیم و روی CPU کلیک کرده تا جدول Connection باز شود می بینیم که یک سطر در آن ایجاد شده است و با کلیک روی آن ID و LADRR مشخص است.

ادامه کار برای برنامه نویسی شبیه قبل است.

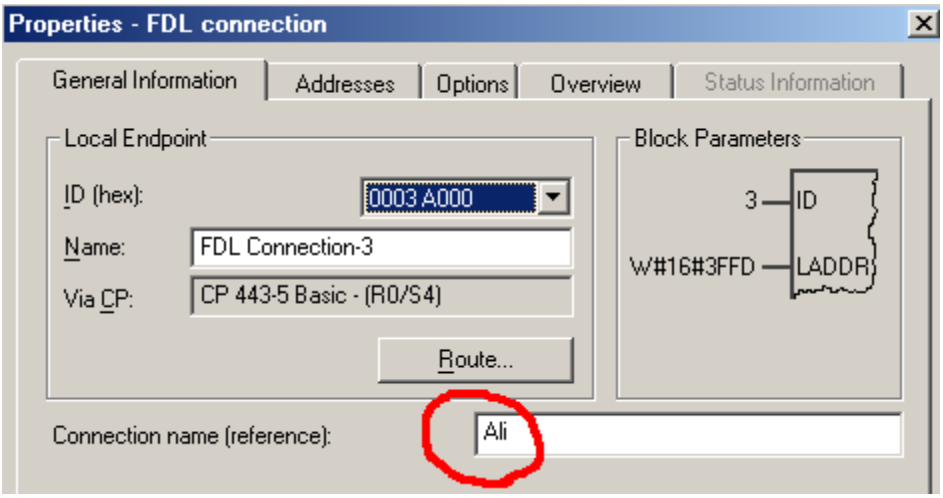
۸-٤ ارتباط FDL با Unknown Project

در این روش بدون اینکه لازم باشد آدرس Node و LSAP ایستگاه مقابل را داشته باشیم صرفاً با استفاده از یک نام ارتباط FDL را برقرار میکنیم. این کار در داخل یک Multiproject انجام میشود. در واقع به این روش اتصال فوق را برای استفاده بعدی رزرو میکنیم.

برا ی این کار مانند روش ذکر شده در صفحه قبل **گام های اول و دوم و سوم** را بر میداریم سپس:

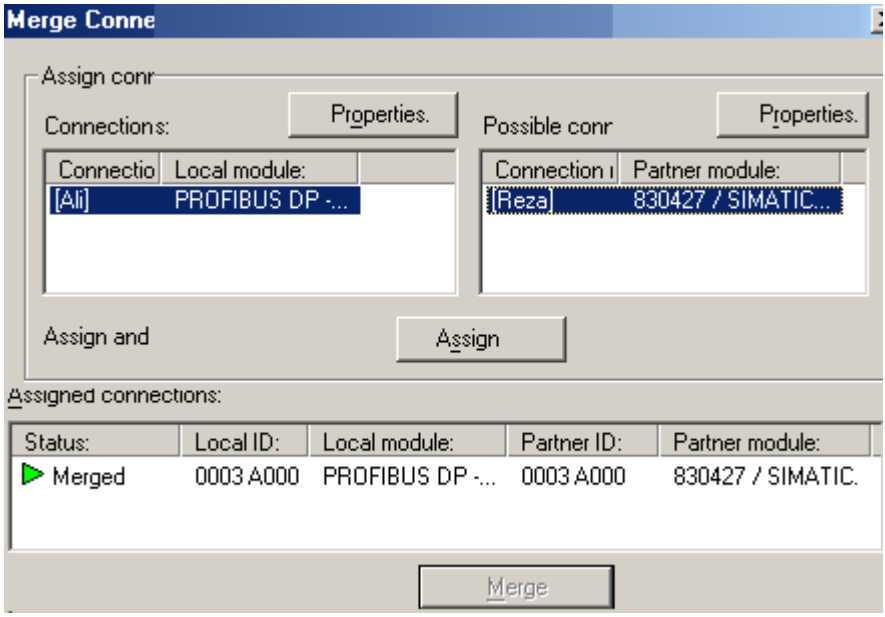
گام چهارم : در پنجره Connection Partner (شکل صفحه قبل) In Unknown Project را انتخاب میکنیم.

گام پنجم : پس از OK کردن در پنجره Properties مانند شکل زیر یک نام Reference بدلخواه وارد میکنیم (مثلا Ali) این نام کلید ارتباط بین این پروژه و پروژه بعدی است. توجه شود اگر در این پنجره به بخش Address مراجعه کنیم خواهیم دید که آدرس Remote فعلاً خالی و غیر فعال است.

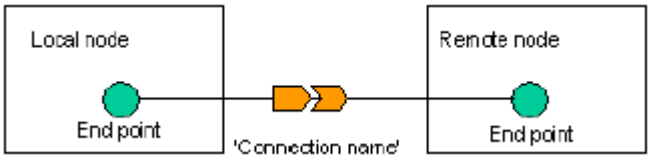


گام ششم : برای پروژه دیگر نیز گام چهارم و پنجم را برداشته و اسم دلخواهی (مانند Reza) به عنوان reference ارتباط معرفی میکنیم.

گام هفتم : در NetPro ازمنوی Edit > Merge Connection استفاده کرده تا پنجره زیر باز شود. در دوطرف این پنجره دو نام reference را میبینیم. با کلید Assign آنها را به هم مربوط می کنیم.پس از آن میبینیم که سطر جدیدی در پنجره پایینی ظاهر میشود با کلیک روی آن و سپس کلیک روی Merge این دو اتصال را به هم لینک می نماییم.



پس از اتمام Merge کردن اگر به جدول Connection هر کدام از CPU ها بر گردیم خواهیم دید که در بخش Address به Remote آدرس اختصاص داده شده است.



ادامه کار برای برنامه نویسی شبیه قبل است.

تذکر:

در تمام موارد پس از پیکر بندی FDL چک سازگاری اجزا و دانلود به PLC ها نباید فراموش شود.

فصل نهم – پیکر بندی و برنامه نویسی شبکه PROFIBUS FMS

مشتمل بر :

۹-۱ شناخت ارتباط FMS

۹-۲ نحوهٔ پیکر بندی ارتباط FMS

۹-۳ برنامه نویسی ارتباط FMS

۹-۴ مثالی از برنامه نویسی ارتباط FMS

مقدمه

در مورد FMS و ویژگیهای آن قبلاً به اختصار مطالبی بیان شد. در این بخش به نحوه پیکر بندی و برنامه نویسی شبکه FMS توسط STEP7 می پردازیم. قبل از هر چیز باید توجه داشت که برای ارتباط FMS نیاز به کارت CP داریم ولازم است این کارت را از زیر مجموعه کارتهای PROFIBUS در Station مورد نظر انتخاب کرده ودر رک مربوطه در Hwconfig قرار دهیم . این کارت CP باید قابلیت FMS را داشته باشد که این موضوع با کلیک کردن روی کارت در پنجره کاتالوگ و مشاهده توضیحات زیر پنجره مشخص می گردد.

کارتهایی که برای این منظور قابل انتخاب هستند عبارتند از :

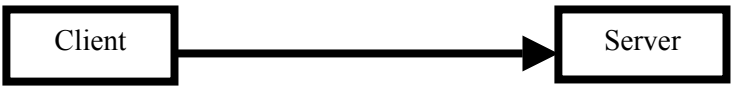
- S7-300 با کارتهای CP343-5
- S7-400 با کارتهای CP443-5 Basic
- S5 با کارت CP 5431
- PC با کارتهای CP5613 و CP5614
- ET200U با IM318C

وسایل سازندگان دیگر به جز زیمنس که قابلیت FMS داشته باشند با وارد کردن فایل GSD یا GSE به برنامه مطابق روالی که قبلاً توضیح داده شد به پنجره کاتالوگ اضافه می شوند وقابل استفاده هستند.

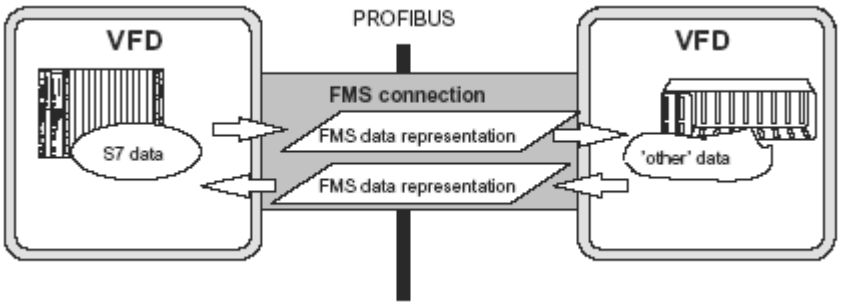
۹-۱ شناخت ارتباط FMS

اتصال FMS ارتباط بین Master را از طریق تکنیک TokenPass فراهم می کند. FMS و DP میتوانند روی یک Media قرار گیرند یعنی باس شبکه می تواند برای هر دوی آنها مشترک باشد . به همان صورت که ارتباط DP و FDL را همزمان استفاده می کردیم.

نکته ای که در ارتباط FMS باید مد نظر قرار گیرد مفهوم دو کلمهٔ Client و Server است.Client عضوی است که درخواست سرویس می کند و Server عضوی است که این سرویس را ارائه می دهد.



مفهوم Client/Server با مفهوم Master/ Slave متفاوت است در Master/Slave اگر چه Master درخواست می کند و Slave پاسخ می دهد ولی باس همیشه در اختیار Master است و Slave بدون اجازهٔ Master نمی تواند آن را در اختیاربگیرد . ولی در Client/Server هم Client و هم Server حق مساوی برای در اختیار گرفتن باس دارند اگر چه بیشتر Client به این کار مبادرت می کند . در FMS دو وسیله ای که با هم ارتباط برقرار می کنند به عنوان VFD (Virtual Field Device) شناخته می شوند.به عنوان مثال دو PLC شکل زیر که یکی S7-300 و دیگری S7-400 است هر دو VFD هستند:



در اتصال FMS کارت CP می تواند برای هر job حداکثر ۲۴۱ بایت اطلاعات را منتقل کند که به این یک بستهٔ PDU (Protocol Data Unit) می گویند . این موضوع اگر چه با ارتباط FDL شبیه است ولی یک فرق اساسی بین FDL و FMS وجود دارد وآن اینکه در FMS در سمت Server یک دیتا بلاک به عنوان منبع اطلاعاتی برای خواندن ونوشتن معرفی می گردد و Client ار آن می خواند یا در آن می نویسد. در سمت Client نیز دیتا بلاک هایی وجود دارد ولی به طور معمول Server نیازی به نوشتن یا خواندن آنها ندارد اگر چه این امر برایش امکان پذیر است از این رو در سمت Server معمولاً نیازی به برنامه نویسی نداریم و برنامه فقط در سمت Client نوشته می شود .در حالیکه در ارتباط FDL در هر دوطرف نیاز به برنامه نویسی داریم . در FMS در سمت Server صرفاً لازم است دیتا بلاک مرجع را به شکلی که اصطلاحاً به آن ایندکس کردن می گویند مشخص کنیم تا در سمت Client قابل شناسایی باشد.

در ارتباط FMS کارت CP وظایف زیر را بعهده دارد:

۱. در سمت فرستنده دریافت دیتا از CPU و تبدیل آن از فرمت S7 به فرمت FMS و ارسال به سمت گیرنده.

۲. در سمت گیرنده دریافت دیتا از شبکه پروفی باس و تبدیل آن از فرمت FMS به فرمت خاص قابل استفاده برای وسیله بعنوان مثال اگر گیرنده یک PLC از نوع S7 باشد کدهای مزبور به فرمت S7 تبدیل می شود .

نحوه تبدیل فرمتهای S7 و FMS به یکدیگر در ضمیمه ٤ آمده است .

۹-۲ نحوهٔ پیکر بندی ارتباط FMS

با توضیحاتی که قبلاً در مورد نحوهٔ پیکر بندی ارتباط FDL داده شد خواننده محترم می تواند به سهولت نسبت به پیکر بندی ارتباط FMS اقدام کند . روش کار مشابه FDL است بنابراین از توضیحات تکراری خودداری می کنیم وصرفاً به عناوین آنها اشاره می کنیم ، گامهایی که باید برداشته شود عبارتند از:

گام اول:

ایجاد Station های مورد نظر و پیکر بندی کارت های آن توسط Hwconfig .بهتر است در نام Station ها از کلمات Server و Client استفاده کنیم تا سریعتر و بدون خطا بتوان مراحل بعدی را انجام داد.

گام دوم:

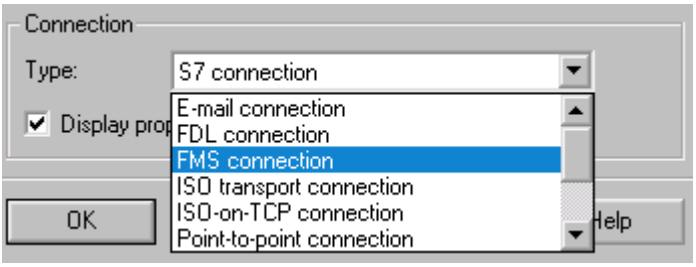
وارد کردن کارت CP مناسب با قابلیت FMS در Station های مورد نظر توسط Hwconfig.

گام سوم:

اجرای Netpro واتصال کارتهای CP به شبکه PROFIBUS .

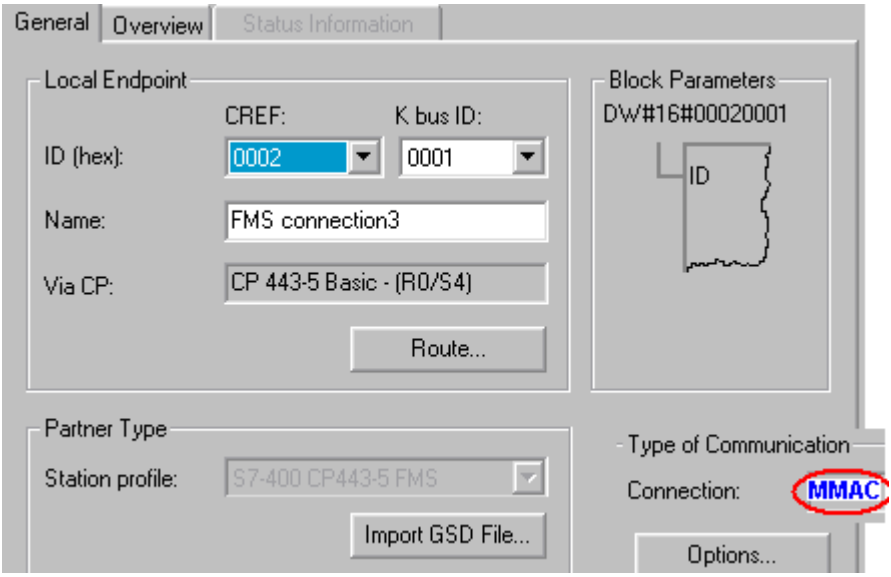
گام چهارم:

دابل کلیک روی سطر Connection table یکی از Station ها وانتخاب ارتباط از نوع FMS

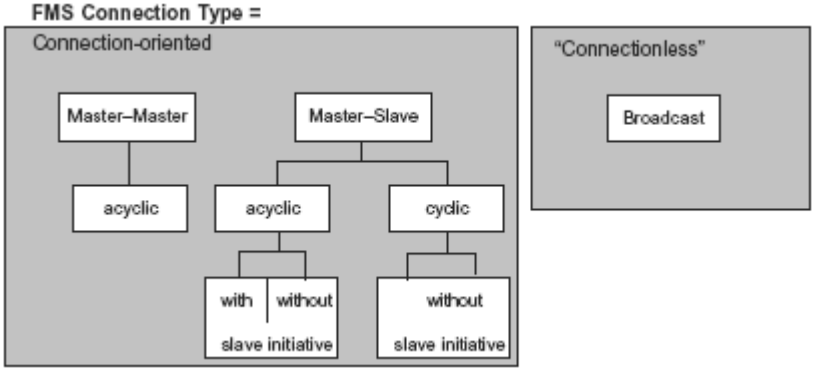


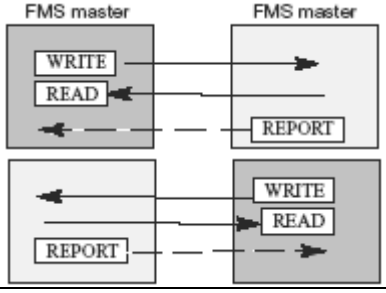
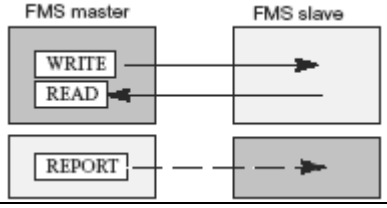
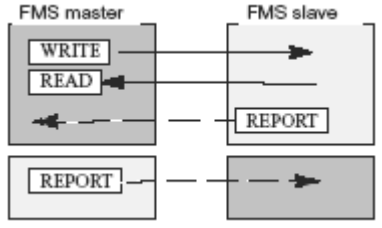
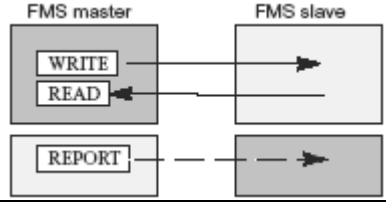
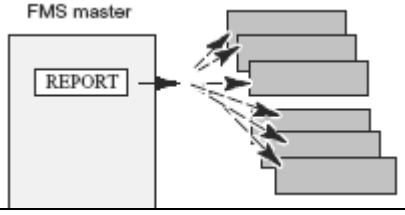
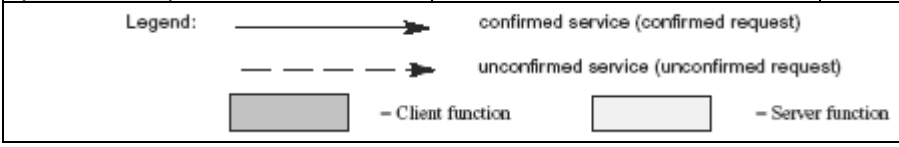
گام پنجم:

یادداشت کردن ID داده شده در پنجرهٔ FMS Connection مانند شکل زیر . همانطور که مشاهده می شود در اینجا ID یک DoubleWord است که از ترکیب CREF+LADDR بدست می آید. باید توجه داشت که LADDR معادل هگز آدرس بیس کارت CP در S7-300 است. برای S7-400 به جای LADDR کلمهٔ KBUSID ظاهر می شود که همان آدرس پایه کارت است.



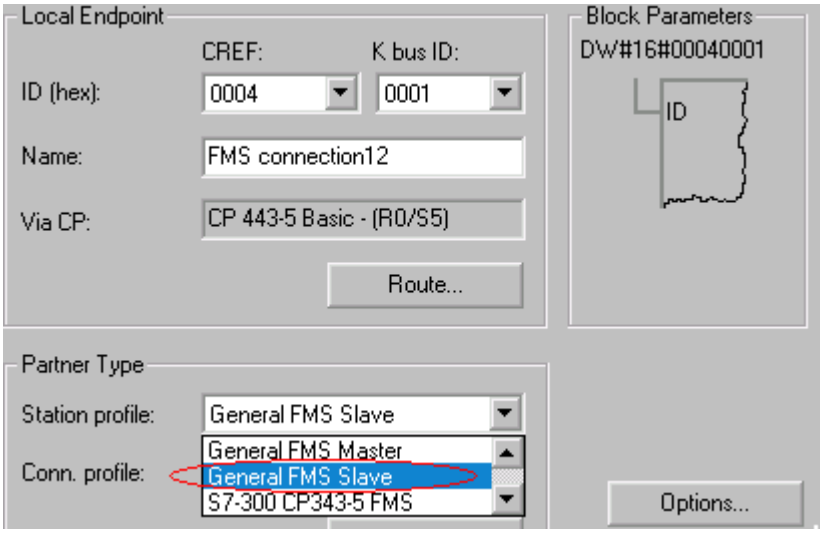
اگر روی Option در پنجره فوق کلیک کنیم پنجرهٔ دیگری ظاهر می شود که در بخش Communication در قسمت Type Of Connection نوع ارتباط کلمهٔ MMAC یا کلمهٔ دیگری مانند MSAC,MSCY,BRCT ظاهر می شود که مفهوم آن درشکل زیر و جدول صفحه بعد آورده شده است.



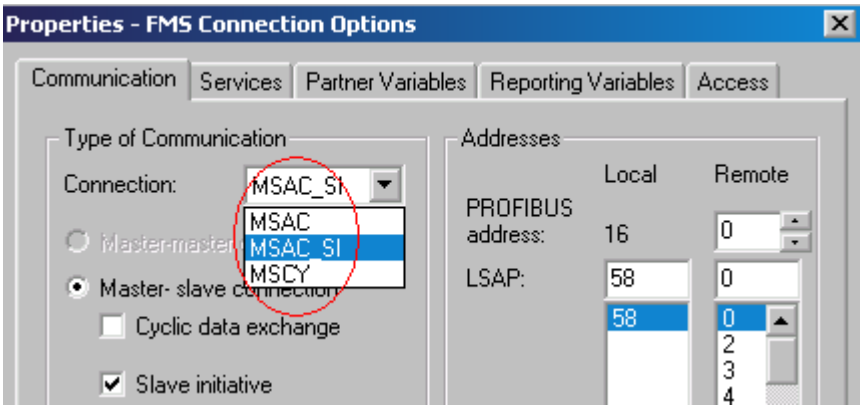
	ارتباط دوطرفه بین دو Master یعنی Write و Read در هر دو جهت و با Acknowledge امکان پذیر است. Report نیز برای هردو عضو وبصورت سرویس SDN امکان پذیر است.	**MMAC** Master-Master Acyclic Connection
	در اینحالت فقط Master است که اجازه Write,Read,Report را دارد. تنظیم این ارتباط یکطرفه در صفحه بعد آمده است.	**MSAC** Master-Slave Acyclic Connection
	اینحالت نیز شبیه MSAC است ولی Slave میتواند در صورتی که Master به او اجازه بدهد Report بفرستد.	**MSAC-SI** Master-Slave Acyclic Connection with Slave Initiative
	اینحالت شبیه MSAC است با این تفاوت که ارتباط سیکلی است.	**MSCY** Master-Slave cyclic Connection without Slave Initiative
	Report از طرف Master به تمام ایستگاهها بصورت SDN ارسال میگردد.	**BRCT** Broadcast
		

نکات قابل توجه در مورد Type of Communication

- اگر از ابتدا در جدول ارتباطات نوع All Broadcast Station انتخاب شده باشد کد BRCT ظاهر میشود.
- بطور معمول برای ارتباط بین دو سیستم نوع MMAC انتخاب میشود.
- اگر ارتباط های نوع MSAC یا MSAC-SI یا MSCY مد نظر باشد لازم است ابتدا در پنجره Properties مانند شکل زیر نوع General FMS Slave را برای طرف مقابل (Partner) انتخاب کرد.



پس از انتخاب فوق اگر روی Optionsکلیک کنیم می توانیم یکی از سه حالت ذکر شده را برای Slave انتخاب نماییم مانند شکل زیر:

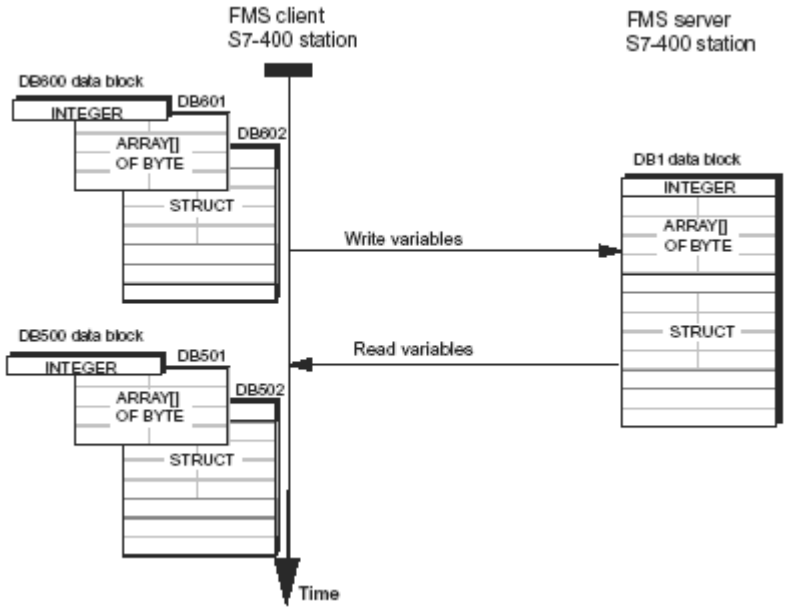


۹-۳ برنامه نویسی ارتباط FMS

همانطور که ذکر شد بطور معمول ما در سمت Server نیاز به برنامه نویسی نداریم. Server صرفاًدیتا بلاک خود را در اختیار Client قرار می دهد.بنابراین:

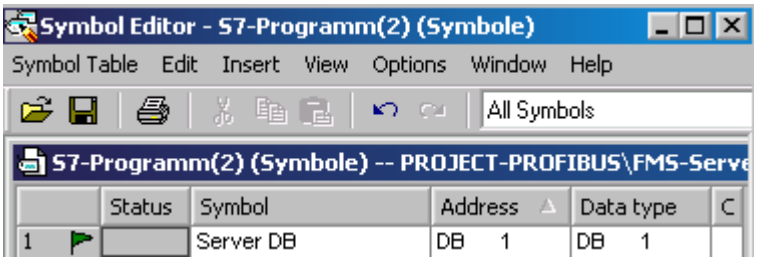
قدم اول:

در سمت Server یک دیتا بلاک با سطر های دلخواه و با نوع متغیر های دلخواه ایجاد می کنیم.باید توجه داشت از آنجا که از سمت Client نیزدیتا از دیتا بلاکی ارسال یا دریافت می شودباید به ساختار آن دقت کرد تا مشابه ساختار سمت Server باشد. شکل زیر:

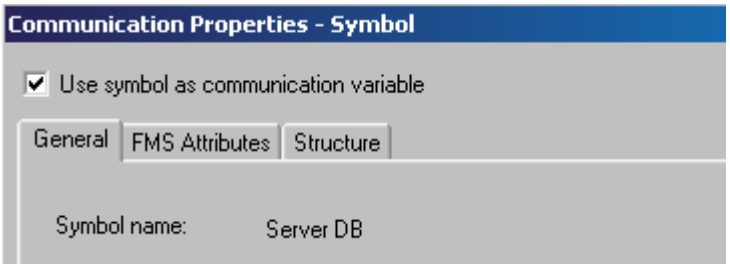


قدم دوم :

در سمت Server جدول سمبلها Symbols Table را باز کرده و DB را با نام سمبلیک دلخواه معرفی می کنیم.



سپس از طریق منوی Edit > Special Object Properties > Communication پنجره ای مانند شکل زیر را باز کرده و گزینهUse Symbol as Communication Partner را فعال میکنیم.

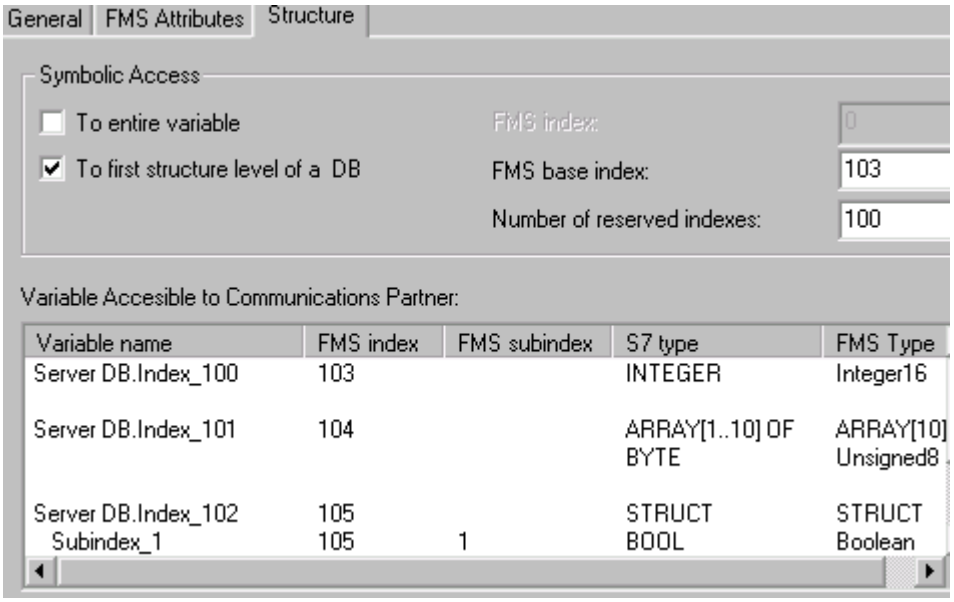


قدم چهارم:

در بخش Struct از پنجره فوق رفته ومشاهده می کنیم که یک شماره Index به دیتا بلاک داده شده است. این شماره را یادداشت کرده تا در سمت Client برای ارتباط با DB از آن استفاده کنیم.این شماره در مثال شکل بالای صفحه بعد 103 میباشد.

می توانیم در سمت Server چند دیتا بلاک برای ارتباط FMS داشته باشیم ، روش ایجاد کردن آنها به همین نحو است و در آخر مشاهده خواهیم کرد که هرکدام یک شماره ایندکس منحصر به فرد دارند.

شماره Index کل DB و شماره Subindex یک سطر از DB را معرفی میکند میتوانیم بسته به نیاز در برنامه نویسی کل DB یا سطر خاصی از آن را آدرس بدهیم تا دیتا از آنجا خوانده یا در آنجا نوشته شود.



قدم پنجم

در سمت Client نیاز به برنامه نویسی داریم روش کار شبیه آنچه برای FDL گفته شد میباشد ولی فانکشن ها متفاوتند.پس بطور خلاصه به نکات مشترک اشاره میکنیم:

- برنامه اصلی میتواند در OB3x یعنی وقفه های سیکلی نوشته شود. از آنجا که در این وقفه ها زمان اجرای بلاک قابل تنظیم است نسبت به OB1 برای کار تبادل دیتا مزیت دارند.
- در OB مربوطه فانکشن بلاک های خواندن (FB3) و نوشتن (FB6) را از مسیر زیر در Program Element وارد برنامه میکنیم.

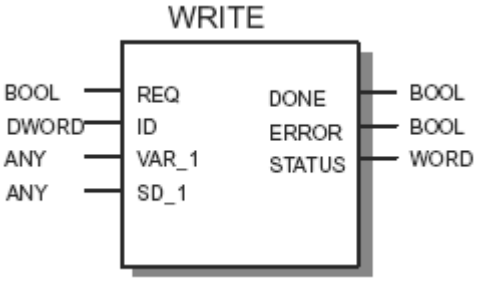
LAD/STL/FBD > Program Element > Libraries> Simatic Net CP

- باید توجه داشت که فانکشنهای 300 و 400 اگرچه هم نام هستند ولی متفاوت میباشند و بسته به CPU باید FB مربوطه را انتخاب نمود.
- معمولا در سمت Server برنامه نویسی لازم نیست . با این وجود میتوان مراحل فوق را در سمت Server نیز دنبال نمود.

بلاک های Read و Write در صفحه بعد توضیح داده شده اند.

فانکشن بلاک Write

این فانکشن بلاک که نام غیر سمبلیک آن FB6 است برای ارسال دیتا استفاده میشود. شکل زیر این فانکشن را بصورت بلاک FBD همراه با ورودی و خروجی هایش نشان میدهد.



ورودی ها :

REQ	یک بیت است که گر 1 باشد دیتا ارسال میشود و اگر 0 باشد خروجیها update میشوند. این بیت را معمولا به یک متغیر حافظه مانند M0.0 میدهیم تا بتوان کار ارسال دیتا را کنترل کنیم.
ID	عدد DWord که در Connection Table هنگام پیکر بندی مشخص شده است. مثال : DW#16#00010001
VAR_1	آدرس index دیتا بلاک سمت Server را در اینجا باید داد ولی مستقیما نمیتوان نوشت مثلا 103 بلکه باید آنرا در یک DB در سمت Client از نوع String معرفی کرده و آدرس سطر DB را در اینجا وارد نماییم مثلاً DB10.test
SD_1	آدرس دیتا بلاک سمت Client و مقدار دیتایی که باید ارسال شود را مشخص میکند. این آدرس میتواند کل DB یا آدرس خاصی از یک DB باشد. مانند DB1.DBW0

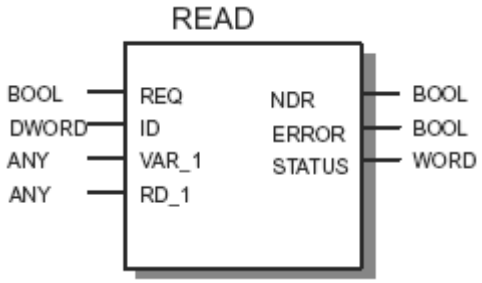
Address	Name	Type	Initial value
0.0		STRUCT	
+0.0	test	STRING[5]	'<100>'

خروجی ها :

عملکرد سه خروجی این فانکشن بلاک شبیه خروجی های فانکشن AG_SEND است که در FDL توضیح داده شدولی کد خطای برگشتی و تفسیر آن متفاوت است . این کدها بدلیل تنوع و تعداد زیاد در ضمیمه ۵ آورده شده اند.

فانکشن بلاک Read

این فانکشن بلاک که نام غیر سمبلیک آن FB3 است برای خواند دیتا استفاده میشود. شکل زیر این فانکشن را بصورت بلاک FBD همراه با ورودی و خروجی هایش نشان میدهد.



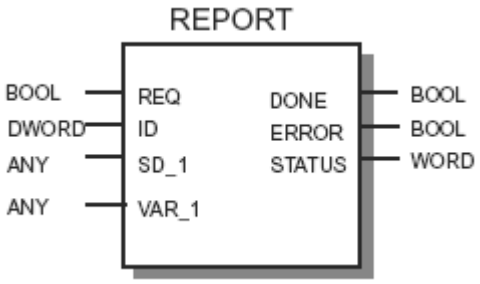
با توجه به شباهت بسیاری از ورودی و خروجی ها ی این بلاک با بلاک FB6 نیازی به توضیح مکرر نیست. ورودی RD_1 این بلاک آدرس محلی از دیتا بلاک سمت Client را میگیرد که قرار است در آنجا دیتا ذخیره شود..خروجی NDR (New Data Received) مشابه این خروجی در AG_RECV مربوط به ارتباط FDL میباشد.

مثالی از صدا زدن این FB در زیر آورده شده است:

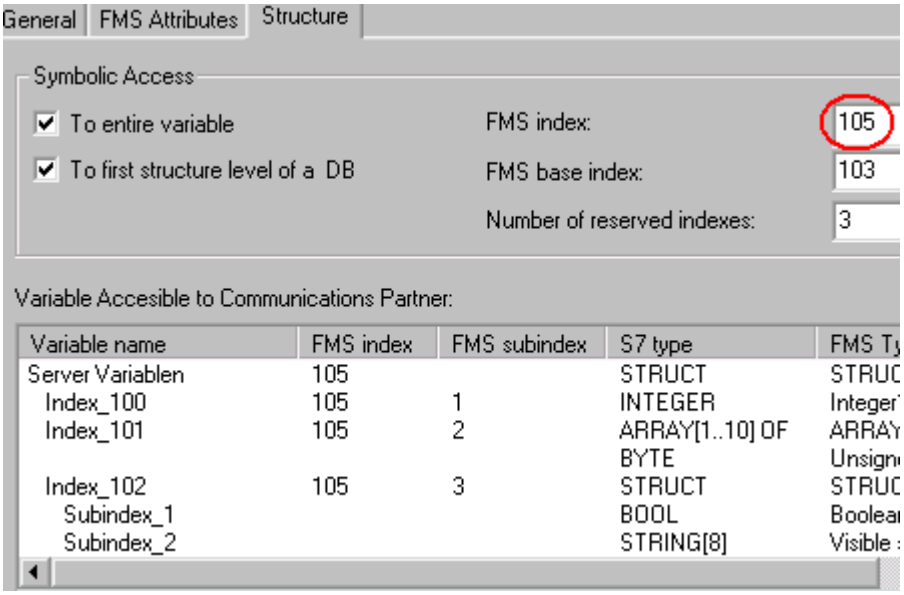
```
//READ block call with instance DB          CALL FB 3, DB 29
//Signal edge change to execute the FB      REQ := M 1.0
//compared with configuration of FMS        ID := DW#16#10001
//addresses K variable that will be read    VAR_1 := “SLAVE2”.INDEX
//addresses data area as destination        RD_1 := “PROCESS”.IMAGE
//confirmation of execution                 NDR := M 1.1
//indicates incorrect execution             ERROR := M 1.2
//detailed error decoding                   STATUS := MW 20
```

فانکشن بلاک Report

این فانکشن بلاک که نام غیر سمبلیک آن FB4 است شبیه Write عمل میکند ولی منتظر تایید نمی ماند.



ورودی ها و خروجی های این بلاک شبیه Write است. ولی باید توجه داشت که برای شماره ایندکس باید از عدد موجود در Structure مربوط به Symbol Table مطابق شکل زیر استفاده شود:



۹-۴ مثالی از برنامه نویسی ارتباط FMS

این مثال برنامه ارتباط FMS را بصورت STL Source نشان میدهد. این برنامه در سمت Client نوشته شده و در سمت Server هیچگونه برنامه نویسی بکار نرفته است.

```
FUNCTION FC 100 : VOID
TITLE =CALL_READ
//This function illustrates an FMS communication job with the READ (FB3)
//block. A variable of the type integer is accessed.
VERSION : 0.1
BEGIN
NETWORK
TITLE =Read variable with index 100.
//Read data access:
//A variable with index 100 is read and entered in the destination data
//DB500).
//Call parameters in DB 100.
//Job handling /job coordination:
//This job is only started when all other jobs have been completed. This
// means that access to the variable is synchronized.
//The job is triggered by a signal edge at the REQ input. REQ is reset
// when the job is completed or an error occurred. Following this, the //
signal is set again.
//If an error occurred, the error number can be read from the status word.
//Only execute job when no other job is active.
      O     DB200.DBX    0.0; //  REQ=1, WRITE /active
      O     DB101.DBX    0.0; //  REQ=1, READ /active
      O     DB201.DBX    0.0; //  REQ=1, WRITE /active
      O     DB102.DBX    0.0; //  REQ=1, READ /active
      O     DB202.DBX    0.0; //  REQ=1, WRITE /active
      JC    ende;
//Calling the READ job:
      CALL FB     3 , DB   300 (
           REQ                       := DB100.DBX    0.0,
           ID                        := DB100.DBD    2,
           VAR_1                     := DB100.Read_VAR_Index,
           RD_1                      := DB500.DBW    0,
           NDR                       := DB100.DBX   13.0,
           ERROR                     := DB100.DBX   13.1,
           STATUS                    := DB100.DBW   14);
// /Query job status
      A     DB100.DBX    0.0; //   REQ=1 =>
//                                  REQ=1 => Job active, query errors
      JC    fehl;
      AN    DB100.DBX    0.0; //   REQ=1 =>
//                                  REQ=1 => Job completed
      S     DB100.DBX    0.0; //    / Set REQ again
      JU    ende;
```

```
//  Error handling
fehl: AN    DB100.DBX   13.1; // Error=0 =>
//                              Error=0 => No error occurred
      JC    ok;
      A     DB100.DBX   13.1; // Error=1 =>
//                              Error=1 => Error occurred
      L     DB100.DBW   14; //   / Load error status
      T     DB100.DBW   16; //  //  Save in buffer
      R     DB100.DBX    0.0; //     / Reset REQ
      JU    ende;     //  / Query job status

ok:   AN    DB100.DBX   13.0; //  Done=0 =>
//                              Done=0 => Job still active
      JC    ende;

      A     DB100.DBX   13.0; //  Done=1 =>
//                              Done=1 => Job complete without error
      R     DB100.DBX    0.0; //   / Reset REQ

ende: BE    ;
END_FUNCTION


FUNCTION FC 101 : VOID
TITLE =CALL_READ
//The function illustrates an FMS communication job with the READ(FB3)
// function block. A variable of the type array is accessed.
VERSION : 0.1
BEGIN
NETWORK
TITLE = Read variable with index 101.
//Read data access:
//The variable with index 101 is read and entered in the destination data
//area (here DB501). Call parameters in DB101.
//
//Job handling /job coordination:
//This job is only started when all other jobs have been completed. This
//means that access to the variable is synchronized.
//The job is triggered by a signal edge at the REQ input. REQ is reset
//when the job is completed or an error occurred. Following this, the.
//signal is set again If an error occurred, the error number can be read
//from the status word.
// Only execute job when no other job is active.
      O     DB200.DBX    0.0; //  REQ=1, WRITE   / active
      O     DB100.DBX    0.0; //  REQ=1, READ   / active
      O     DB201.DBX    0.0; //  REQ=1, WRITE   / active
      O     DB102.DBX    0.0; //  REQ=1, READ   / active
      O     DB202.DBX    0.0; //  REQ=1, WRITE   / active
      JC    ende;
```

```
//Calling the READ job:
      CALL FB     3 , DB   301 (
           REQ                        := DB101.DBX    0.0,
           ID                         := DB101.DBD    2,
           VAR_1                      := DB101.Read_VAR_Index,
           RD_1                       := DB501.Index_101,
           NDR                        := DB101.DBX   13.0,
           ERROR                      := DB101.DBX   13.1,
           STATUS                     := DB101.DBW   14);
// Query job status
      A     DB101.DBX    0.0; //   REQ=1 =>
//                                  REQ=1 => Job active, query errors
      JC    fehl;

      AN    DB101.DBX    0.0; //   REQ=0 =>
//                                  REQ=1 => Job completed
      S     DB101.DBX    0.0; //    / Set REQ again
      JU    ende;
// Error handling

fehl: AN    DB101.DBX   13.1; // Error=0 =>
//                                Error=0 => No error occurred
      JC    ok;
      A     DB101.DBX   13.1; // Error=1 =>
//                                Error=1 => Error occurred

   L     DB101.DBW   14; //load error status
      T     DB101.DBW   16; // Save in buffer
      R     DB101.DBX    0.0; //    Reset REQ
      JU    ende;
//  / Query job status
ok:   AN    DB101.DBX   13.0; //  Done=0 =>
//                                 Done=0 => Job still active
      JC    ende;
      A     DB101.DBX   13.0; //  Done=1 =>
//                                 Done=1 => Job complete without error
      R     DB101.DBX    0.0; //    Reset REQ
ende: BE    ;
END_FUNCTION
```

```
FUNCTION FC 102 : VOID
TITLE =CALL_READ
//This function illustrates an FMS communication job with the READ (FB3)
// function block. A variable of the type STRUCT is accessed.
VERSION : 0.1
BEGIN
NETWORK
TITLE =Variable mit dem Index 102 lesen. Read variable with index 102.
//Read data access:
//A variable with index 102 is read and entered in the destination data
// area (here
//DB502). Call parameters in DB 100.
//Job handling /job coordination:
//This job is only started when all other jobs have been completed. This
// means that access to the variable is synchronized.
//The job is triggered by a signal edge at the REQ input. REQ is reset
// when the job is completed or an error occurred. Following this, the
// signal is set again.If an error occurred, the error number can be read
//from the status word.
// Query whether other jobs still active
   O     DB100.DBX    0.0; //  REQ=1, READ  /active
      O     DB200.DBX    0.0; //  REQ=1, WRITE  /active
      O     DB101.DBX    0.0; //  REQ=1, READ  /active
      O     DB201.DBX    0.0; //  REQ=1, WRITE  /active
      O     DB202.DBX    0.0; //  REQ=1, WRITE  /active
      JC    ende;
/// Calling the READ job
      CALL FB     3 , DB   302 (
           REQ                       := DB102.DBX    0.0,
           ID                        := DB102.DBD    2,
           VAR_1                     := DB102.Read_VAR_Index,
           RD_1                      := DB502.Index_102,
           NDR                       := DB102.DBX   13.0,
           ERROR                     := DB102.DBX   13.1,
           STATUS                    := DB102.DBW   14);
//  /Query the job status
      A     DB102.DBX    0.0; //   REQ=1 =>
//                                  REQ=1 => Job active, query errors
      JC    fehl;

      AN    DB102.DBX    0.0; //   REQ=1 =>
//                                  REQ=1 => Job completed
      S     DB102.DBX    0.0; //     / Set REQ again
      JU    ende;
//  / Error handling
fehl: AN    DB102.DBX   13.1; // Error=0 =>
//                               Error=0 => No error occurred
      JC    ok;
```

```
      A     DB102.DBX   13.1; // Error=1 =>
//                              Error=1 => Error occurred
      L     DB102.DBW   14; //  / Load error status
      T     DB102.DBW   16; // // Save in buffer
      R     DB102.DBX    0.0; //     / Reset REQ
      JU    ende;
//  / Query job status
ok:   AN    DB102.DBX   13.0; //  Done=0 =>
//                              Done=0 => Job still active
      JC    ende;
      A     DB102.DBX   13.0; //  Done=1 =>
//                              Done=1 => Job complete without error
      R     DB102.DBX    0.0; //     / Reset REQ
ende: BE    ;
END_FUNCTION


FUNCTION FC 200 : VOID
TITLE =CALL_WRITE
//This function illustrates an FMS communication job with the WRITE (FB6)
//function
//block. A variable of the type integer is accessed.
VERSION : 0.1
BEGIN
NETWORK
TITLE =Variable mit dem Index 100 schreiben. Write var. with index 100
//Write data access:
//A variable with index 100 is written. It is taken from the source data
// area (here DB600). Call parameters in DB200.
//Job handling /job coordination:
//This job is only started when all other jobs have been completed. This
// means that access to the variable is synchronized.
//The job is triggered by a signal edge at the REQ input. REQ is reset
// when the job is completed or an error occurred. Following this, the.
// signal is set again If an error occurred, the error number can be read
// from the status word.
  //  / Query whether other jobs active
      O     DB100.DBX    0.0; //    REQ=1, WRITE  /active
      O     DB101.DBX    0.0; //    REQ=1, READ  /active
      O     DB201.DBX    0.0; //    REQ=1, WRITE  /active
      O     DB102.DBX    0.0; //    REQ=1, READ  /active
      O     DB202.DBX    0.0; //    REQ=1, WRITE  /active
      JC    ende;
//  / Calling the write job
      CALL FB     6 , DB   400 (
           REQ                    := DB200.DBX    0.0,
           ID                     := DB200.DBD    2,
           VAR_1                  := DB200.Write_VAR_Index,
           SD_1                   := DB600.DBW    0,
           DONE                   := DB200.DBX   13.0,
           ERROR                  := DB200.DBX   13.1,
           STATUS                 := DB200.DBW   14);
```

```
      A     DB200.DBX    0.0; //    REQ=1 =>
//                                  REQ=1 => Job active, query errors
      JC    fehl;
      AN    DB200.DBX    0.0; //    REQ=0 =>
//                                  REQ=1 => Job completed
      S     DB200.DBX    0.0; //      / Set REQ again
      JU    ende;
//   / Error handling
fehl: AN    DB200.DBX   13.1; //  Error=0 =>
//                                Error=0 => No error occurred
      JC    ok;
      A     DB200.DBX   13.1; //  Error=1 =>
//                                Error=1 => Error occurred
      L     DB200.DBW   14; //    / Load error status
      T     DB200.DBW   16; // // Save in buffer
      R     DB200.DBX    0.0; //   / Reset REQ
      JU    ende;

//  / Query job status

ok:   AN    DB200.DBX   13.0; //   Done=0 =>
//                                  Done=0 => Job still active
      JC    ende;
      A     DB200.DBX   13.0; //   Done=1 =>
//                                  Done=1 => Job complete without error
      R     DB200.DBX    0.0; //      / Reset REQ
ende: BE    ;
END_FUNCTION

FUNCTION FC 201 : VOID
TITLE =CALL_WRITE
//This function illustrates an FMS communication job with the WRITE (FB6)
//function
//block. A variable of the type ARRAY is accessed.
VERSION : 0.1
BEGIN
NETWORK
TITLE =Variable mit dem Index 101 schreiben. Write var with index 101.
//Write data access:
//A variable with index 100 is written. It is taken from the source data
// area (here DB601). Call parameters in DB201.
//Job handling /job coordination:
//This job is only started when all other jobs have been completed. This
// means that access to the variable is synchronized.
//The job is triggered by a signal edge at the REQ input. REQ is reset
// when the job is completed or an error occurred. Following this, the
// signal is set again.
//If an error occurred, the error number can be read from the status word.
//  / Query whether other jobs still active
```

```
  O     DB100.DBX    0.0; //    REQ=1, WRITE  /active
      O     DB200.DBX    0.0; //    REQ=1, WRITE  /active
      O     DB101.DBX    0.0; //    REQ=1, READ  /active
      O     DB102.DBX    0.0; //    REQ=1, READ  /active
      O     DB202.DBX    0.0; //    REQ=1, WRITE  /active
      JC    ende;
//   / Calling the write job
      CALL FB     6 , DB   401 (
           REQ                        := DB201.DBX    0.0,
           ID                         := DB201.DBD    2,
           VAR_1                      := DB201.Write_VAR_Index,
           SD_1                       := DB601.Index_101,
           DONE                       := DB201.DBX   13.0,
           ERROR                      := DB201.DBX   13.1,
           STATUS                     := DB201.DBW   14);
//    / Query job status
      A     DB201.DBX    0.0; //    REQ=1 =>
//                                    REQ=1 => Job active, query errors
      JC    fehl;

      AN    DB201.DBX    0.0; //    REQ=0 =>
//                                    REQ=1 => Job completed
      S     DB201.DBX    0.0; //      / Set REQ again
      JU    ende;
//   / Error handling
fehl: AN    DB201.DBX   13.1; //  Error=0 =>
//                                  Error=0 => No error occurred
      JC    ok;
      A     DB201.DBX   13.1; //  Error=1 =>
//                                  Error=1 => Error occurred
      L     DB201.DBW   14; //    / Load error status
      T     DB201.DBW   16; // Save in buffer
      R     DB201.DBX    0.0; //      / Reset REQ
      JU    ende;
//  / Query job status
ok:   AN    DB201.DBX   13.0; //   Done=0 =>
//                                  Done=0 => Job still active
      JC    ende;
      A     DB201.DBX   13.0; //   Done=1 =>
//                                  Done=1 => Job complete without error
      R     DB201.DBX    0.0; //      / Reset REQ
ende: BE    ;
END_FUNCTION
```

```
FUNCTION FC 202 : VOID
TITLE =CALL_WRITE.
//This function illustrates an FMS communication job with the WRITE (FB6)
//function
//block. A variable of the type STRUCT is accessed.
VERSION : 0.1
BEGIN
NETWORK
TITLE =Variable mit dem Index 102 schreiben. Write var with index 102.
//Write data access:
//A variable with index 102 is written. It is taken from the source data
// area (here DB602). Call parameters in DB202.
//
//Job handling /job coordination:
//This job is only started when all other jobs have been completed. This
// means that access to the variable is synchronized.
//The job is triggered by a signal edge at the REQ input. REQ is reset
// when the job is completed or an error occurred. Following this, the
// signal is set again.If an error occurred, the error number can be read
// from the.status word
// / Query whether other jobs active
      O     DB100.DBX    0.0; //  REQ=1, READ  /active
      O     DB200.DBX    0.0; //  REQ=1, WRITE  /active
      O     DB101.DBX    0.0; //  REQ=1, REA  /active
      O     DB201.DBX    0.0; //  REQ=1, WRITE  /active
      O     DB102.DBX    0.0; //  REQ=1, REA  /active
      JC    ende;
//   / Calling the write job
      CALL FB     6 , DB   402 (
           REQ                       := DB202.DBX    0.0,
           ID                        := DB202.DBD    2,
           VAR_1                     := DB202.Write_VAR_Index,
           SD_1                      := DB602.Index_102,
           DONE                      := DB202.DBX   13.0,
           ERROR                     := DB202.DBX   13.1,
           STATUS                    := DB202.DBW   14);
      A     DB202.DBX    0.0; //    REQ=1 =>
//                                   REQ=1 => Job active, query errors
      JC    fehl;

      AN    DB202.DBX    0.0; //    REQ=0 =>
//                                   REQ=1 => Job completed
      S     DB202.DBX    0.0; //      / Set REQ again
      JU    ende;
```

```
//   / Error handling
fehl: AN    DB202.DBX   13.1; //  Error=0 =>
//                                Error=0 => No error occurred
      JC    ok;
      A     DB202.DBX   13.1; //  Error=1 =>
//                                Error=1 => Error occurred
      L     DB202.DBW   14; //    / Load error status
      T     DB202.DBW   16; // //  Save in buffer
      R     DB202.DBX    0.0; //      / Reset REQ
      JU    ende;
//  / Query job status

ok:   AN    DB202.DBX   13.0; //   Done=0 =>
//                                 Done=0 => Job still active
      JC    ende;
      A     DB202.DBX   13.0; //   Done=1 =>
//                                 Done=1 => Job complete without error
      R     DB202.DBX    0.0; //      / Reset REQ
ende: BE    ;
END_FUNCTION

FUNCTION FC 300 : VOID
TITLE =
//This function simulates a signal or value change and checks the write or
// read function via the FMS connection.
VERSION : 0.1
BEGIN
NETWORK
TITLE =
//After writing and reading index 100, the values are compared to check
// that they match. Afterwards the variable is incremented by one.
      L     DB500.DBW    0;
      L     DB600.DBW    0;
      ==I   ;
      JC    inkr;
//  Set memory bit 0.0 if the source and destination do not match.
      S     M      0.0;
// Increment variable with index 100 by one
inkr: L     DB600.DBW    0;
      +     1;
      T     DB600.DBW    0;
      R     M      0.0;
      BE    ;
END_FUNCTION
```

```
ORGANIZATION_BLOCK OB 1
TITLE =
//Functions (FC100..FC300) in which FMS communication jobs are sent are
// called in this block. The FCs are designed so that a coordinated
// sequence is guaranteed .(For the symbolic declaration, refer to the
// entries in the symbol table for this CPU.)
VERSION : 0.1

VAR_TEMP
END_VAR
BEGIN
NETWORK
TITLE =
//Write and read jobs are called sequentially.
      CALL FC   100 ;

      CALL FC   300 ;

      CALL FC   200 ;

      CALL FC   101 ;

      CALL FC   201 ;

      CALL FC   102 ;

      CALL FC   202 ;
END_ORGANIZATION_BLOCK



ORGANIZATION_BLOCK OB 100
TITLE =
VERSION : 0.1
VAR_TEMP
END_VAR

BEGIN
NETWORK
TITLE =

      L     0;
      T     DB500.DBW    0;
      BE    ;
END_ORGANIZATION_BLOCK
```

فصل دهم – عیب یابی و مدیریت خطا در PROFIBUS

مشتمل بر :

۱۰-۱ عیب یابی از طریق وسایل تشخیص عیب

۱۰-۲ عیب یابی از طریق نرم افزار Step7

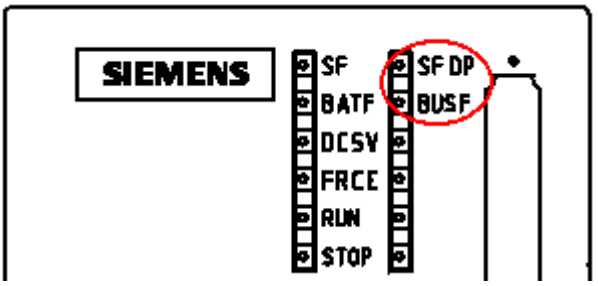
۱۰-۳ مدیریت خطا با استفاده از نرم افزار Step7

مقدمه

Troubleshooting و Diagnostic از مقوله هایی هستند که کاربران اتوماسیون صنعتی بویژه پرسنل که وظیفه آنها نگهداری و تعمیرات این سیستمهاست با آنها به وفور سرو کار دارند. در موقع بروز خطا شناسایی عیب و رفع آن در حداقل زمان ممکن هدفی است که باید محقق شود تا خسارات ناشی از توقف تولید و امثال آن به حداقل کاهش یابد. از اینرو متخصصین اتوماسیون لازم است ابزارهایی را تدارک ببینند تا بتواند اهداف فوق را برآورده سازد. این ابزارها همانگونه که در این بخش خواهیم دید صرفاً ابزارهای سخت افزاری نیستند بلکه استفاده از قابلیت های نرم افزار و برنامه نویسی مناسب نیز از این جمله بشمار می آیند.

در لحظه اول بروز عیب روی شبکه Profibus کاربر ممکن است با توقف CPU با روشن شدن چراغ SF و بطور همزمان چشمک زدن چراغ BF یا Bus Fault روی آن مواجه شود. و همانطور که میدانیم قدم اول ارتباط با PLC و مراجعه به Diagnostic Buffer و نیز دیدن وضعیت شبکه بصورت Online است تا اطلاعات بیشتری نسبت به خطا کسب گردد گرچه بعضاً اطلاعات فوق نیز بطور روشن عیب را مشخص نمی نماید. بعنوان مثال برای یک CPU که از طریق پورت DP به شبکه Profibus متصل است اشکالات زیر با عنوان Bus Access Error ظاهر خواهند شد.

- قطع شدن کابل شبکه
- باز بودن ترمینیتور انتهای باس
- قطع بودن کانکتور DP روی PLC
- عدم تطابق بین شبکه دانلود شده به CPU با آنچه در عمل وجود دارد



۱۰-۱ عیب یابی از طریق وسایل تشخیص عیب

وسایل سخت افزاری مختلف توسط سازندگان مختلف برای عیب یابی شبکه ها طراحی شده اند ولی آنچه در اینجا عنوان میشود صرفاً وسایلی است که زیمنس در مدارک خود به آنها اشاره نموده است.

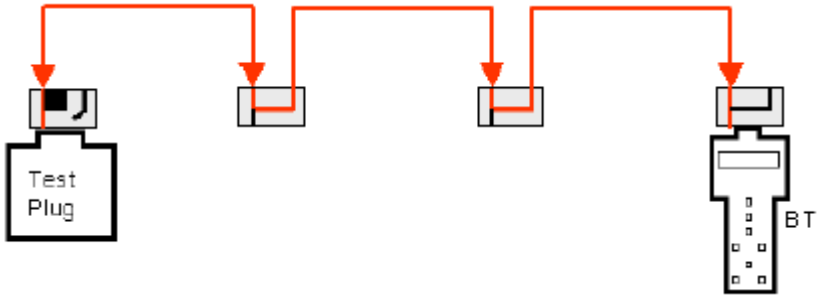
۱۰-۱-۱ وسایل تشخیص عیب در شبکه های الکتریکی

الف) BT200

این وسیله که توسط پورت ۹ پین روی آن به Profibus متصل میگردد قادر است عیوب زیر را در شبکه الکتریکی تشخیص دهد:

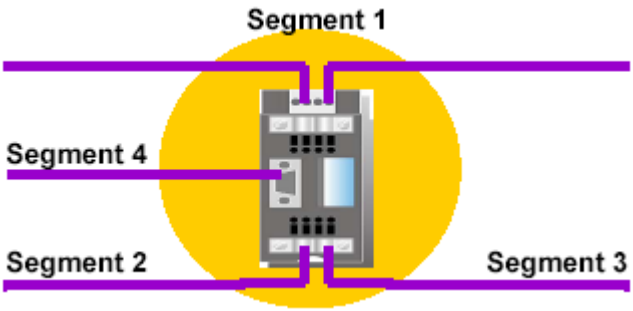
- اتصال کوتاه بین سیم دیتا و شیلد
- قطع شدن خطوط دیتا
- قطع شدن شیلد
- پلاریته بر عکس خطوط A و B
- بازتاب های سیگنال که منجر به خطا میشوند.
- چک کردن ترمینیتورهای فعال

در شبکه ای که قرار است تست شودBT200 را بصورت شکل زیر همراه با یک Test Plug میبندیم. همانطور که مشاهده میشود BT200 در یکطرف و Test Plug در سمت دیگر قرار میگیرد. ترمینیتور روی Test Plug را ON و سایر ترمینیتور ها را OFF می کنیم. BT200 را روشن کرده و کلید Test را فشار میدهیم اگر همه چیز درست باشد پیغام .Cabling O.K ظاهر میشود ولی اگر اشکالی وجود داشته باشد پیغام مرتبط روی صفحه نمایش ظاهر می گردد.



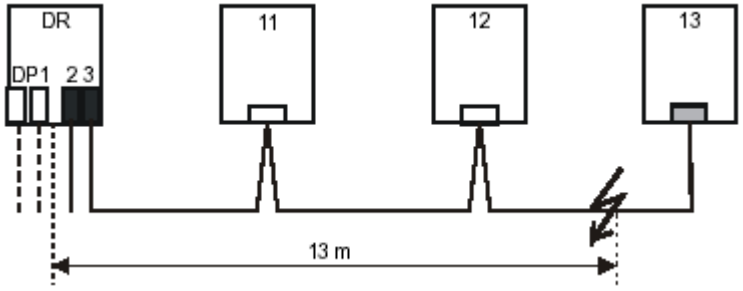
ب) Diagnostic Repeater

این ریپیتر علاوه بر این که کار تقویت سیگنال را روی شبکه الکتریکی RS485 انجام میدهد قادر است اشکالات سگمنت های متصل به خود را آشکار ساخته و CPU را از بروز عیب بطور دقیق مطلع نماید.



همه خطاهایی که توسط BT200 شناسایی میشد توسط این ریپیتر نیز قابل تشخیص است ولی تفاوت آن با BT200 در اینست که این ریپیتر همیشه بصورت On Line به باس متصل است در حالیکه که BT200 فقط در موقع انجام تست به باس وصل میشود. اهم موارد قابل آشکار توسط این وسیله عبارتند از :

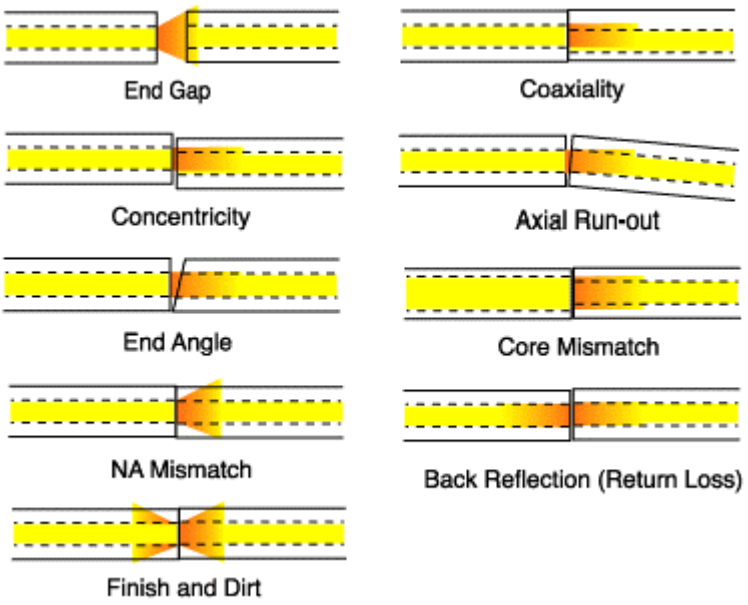
- قطع شدن هر کدام از خطوط
- اتصال کوتاه بین خطوط یا خطوط با شیلد
- پلاریته بر عکس خطوط A و B
- آدرس Node مشکل دار همراه با علت اشکال
- وضعیت ترمینیتورها
- فاصله بین ایستگاهها
- فاصله تا نقطه بروز خطا



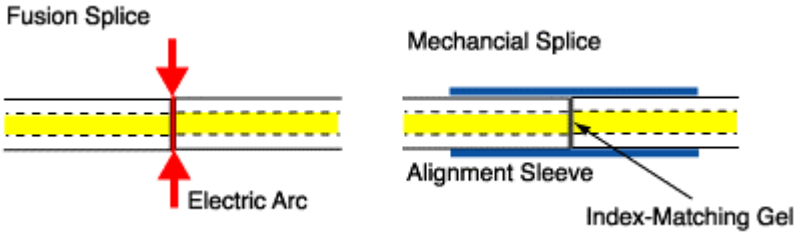
۱۰-۱-۲ وسایل تشخیص عیب در شبکه های نوری

همانطور که در بحث فیبر نوری اشاره شد نصب فیبر نوری نسبت به کابل الکتریکی حساس تر و آسیب پذیرتر است. از جمله اشکالاتی که در حین نصب ممکن است پیش بیاید به موارد زیر میتوان اشاره کرد:

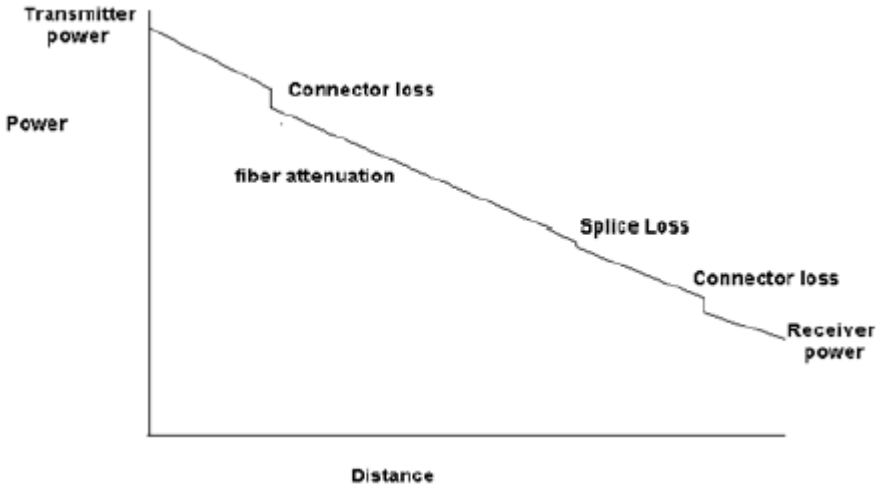
- کشیدن فیبر تحت کشش و استرس زیاد
- خم کردن فیبر بیش از حد مجاز
- اشکال در اتصال کانکتور به فیبر در نقاط اتصال به دستگاه
- عدم دقت در ایجاد Splice ها و وجود اشکالاتی مانند زیر در محل اتصال



عوامل فوق منجر به میرایی سیگنال نوری میشوند. بویژه در نقاط اتصال دو کابل به یکدیگر باید دقت کافی را بکار برد و حتی المقدور بجای Splice مکانیکی از Splice جوشی استفاده نمود.



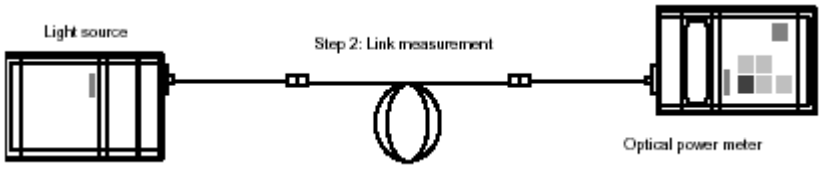
با توجه به توضیحات فوق حتی با وجود دقت در مراحل نصب ، میرایی سیگنال نوری وجود دارد عوامل دیگری نیز مانند طول کابل و جنس کانکتورها بطور اجتناب ناپذیر منجر به میرایی سیگنال میشوند شکل زیر.



بنابراین یکی از مسایل مهم که بعد از اتمام نصب باید بدان توجه شود اندازه گیری میزان میرایی براساس کیفیت سیگنال در سمت گیرنده است که برای این منظور از وسایل مختلفی استفاده میشود. این وسایل خاص دوران نصب و Commissioning نیستند بلکه در هنگام بهره برداری از سیستم نیز ممکن است به آنها نیاز داشته باشیم.

الف) Power Source & Meter

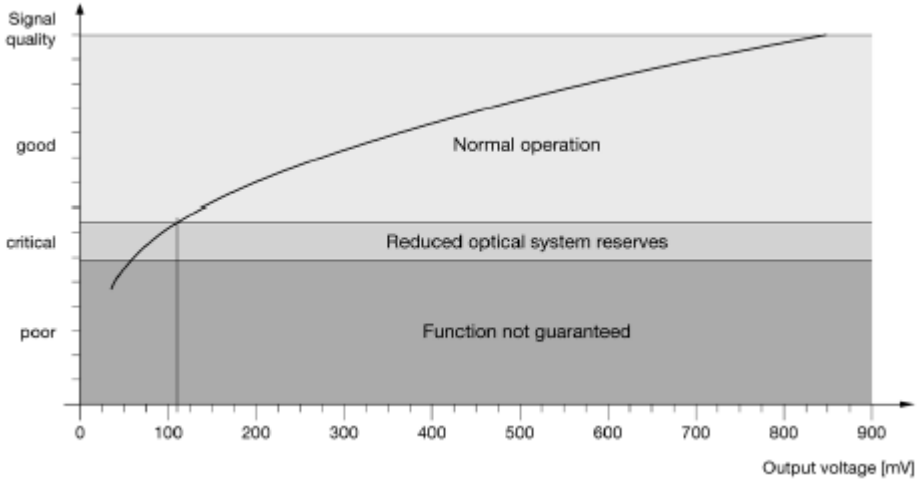
این وسیله متشکل از منبع فرستنده نور است که در یک طرف کابل نوری قرار میگیرد و گیرنده ای دارد که در سمت دیگر فیبر بسته میشود. با ارسال نور در سمت فرستنده و دریافت آن توسط گیرنده میزان میرایی سیگنال اندازه گیری میشود. اگر نتیجه رضایت بخش نبود لازم است با وسیله دیگری بنام OTDR علت اشکال بررسی شود.



ب) استفاده از OLM برای اندازه گیری کیفیت سیگنال

با OLM در بخشهای قبل آشنا شدیم و دیدیم که برای ایجاد شبکه Profibus با فیبر نوری در مسافت های طولانی باید از این وسیله استفاده کرد. در برخی OLM ها (نوع V3) امکان اندازه گیری کیفیت سیگنال در محل فراهم شده است.

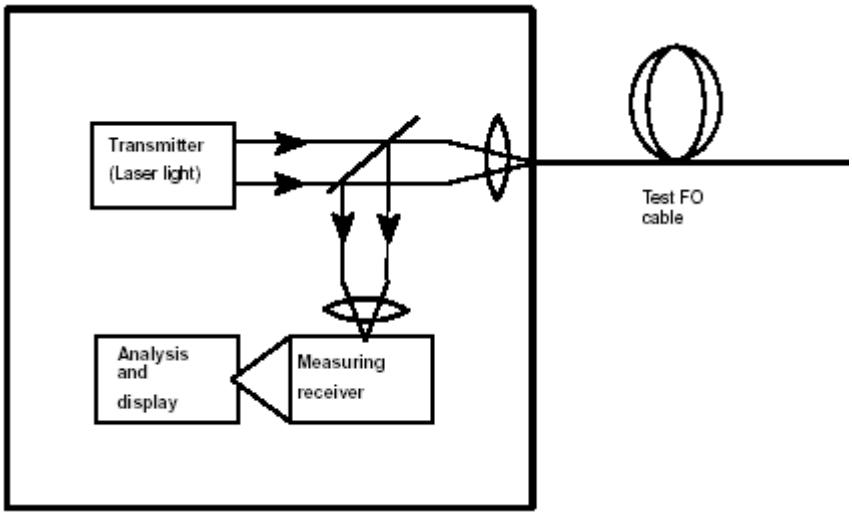
با اتصال یک ولتمتر به OLM میتوان معادل الکتریکی سیگنال نوری را در آن نقطه اندازه گیری نمود سپس با توجه به منحنی زیر وضعیت کیفیت سیگنال را برآورد نمود. نقطه بحرانی ۱۰۰ میلی ولت است.



ج) استفاده از OTDR

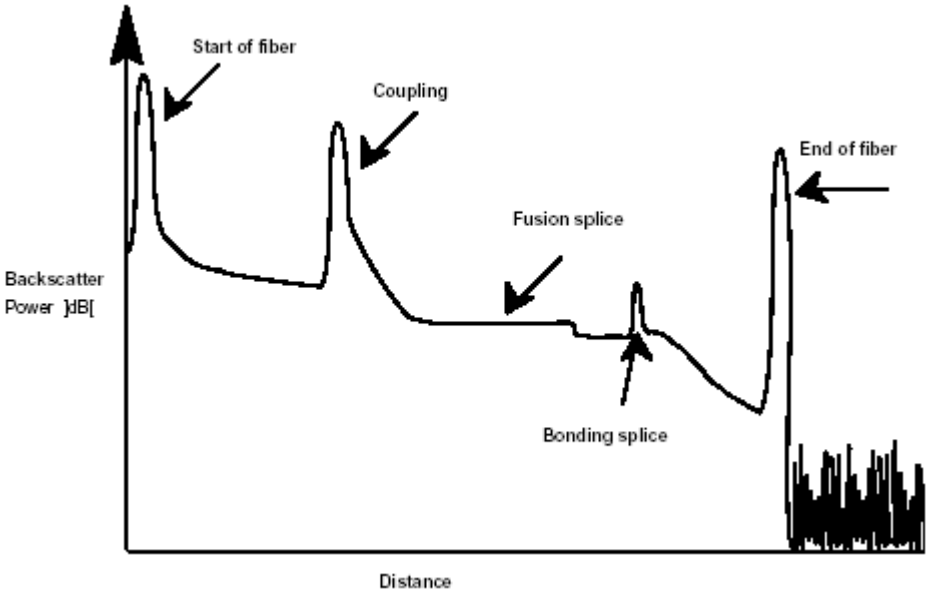
OTDR یا Optical Time Domain Reflectometer وسیله ای است که توسط آن میتوان دقیقترین اطلاعات را در مورد اشکالات فیبر بدست آورد.

روش کار OTDR به این صورت است که نوری با طول موج ۸۵۰ یا ۱۳۰۰ نانومتر ارسال کرده و باز تاب آن را اندازه گیری مینماید. سپس با توجه به شکل موج سیگنال که نمونه ای از آن در صفحه بعد آمده است عیوب فیبر را آنالیز کرده و گزارش میدهد.



بر اساس این شکل موج موارد مختلف قابل آشکار سازی است از جمله :

- نقطه ای از فیبر که آسیب دیده است (در اثر عواملی همچون کشش یا خمش غیر مجاز)
- فاصله نقطه آسیب دیده
- وضعیت Splice ها
- وضعیت کانکتورها
- و نهایتا میزان میرایی سیگنال



۱۰-۲ عیب یابی از طریق نرم افزار

همانند سخت افزار ، نرم افزار های متنوعی نیز توسط سازندگان مختلف جهت آشکار سازی عیوب شبکه طراحی و عرضه شده است. زیمنس نیز علاوه بر Step7 که امکان تشخیص عیوب شبکه را در زیر برنامه های خود دارد نرم افزار های کمکی دیگری را جهت تست و خطا یابی شبکه معرفی نموده است که از این جمله میتوان به نرم افزار AMPROLIZER اشاره کرد ولی آنچه در اینجا مورد بحث ماست صرفاً خطا یابی از طریق Step7 و زیر برنامه های آن می باشد.

الف) استفاده از زیر برنامه Diagnostics

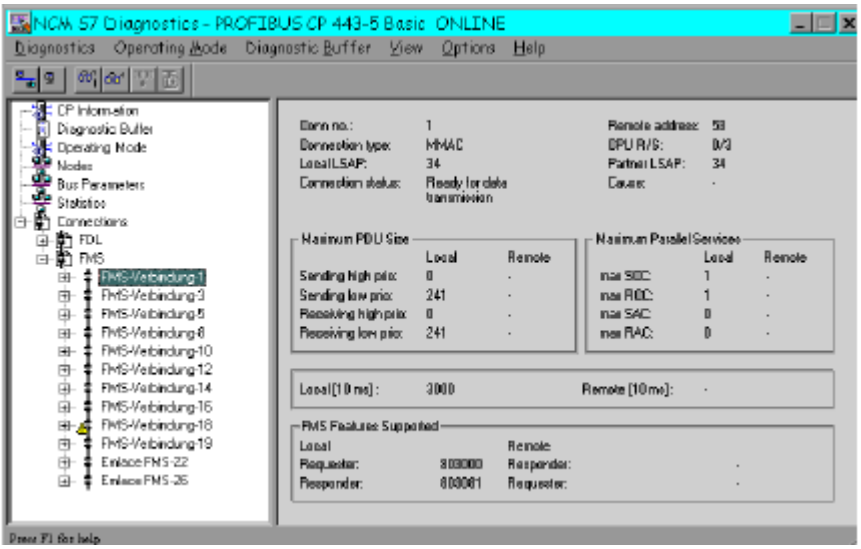
این زیر برنامه پس از نصب Step7 در زیر مجموعه NCM S7 PROFIBUS مانند شکل زیر ظاهر میشود.

توسط این زیر برنامه میتوان تمام ایستگاههایی که توسط کارت CP به شبکه Profibus متصل شده اند را با اطلاعات دقیق مشاهده کرد. بنابراین کاربرد مهم آن در ارتباط FDL و ارتباط FMS است و از اینرو استفاده از این برنامه روی کامپیوتر وقتی امکان پذیر است که دارای کارت CP مناسب باشد.

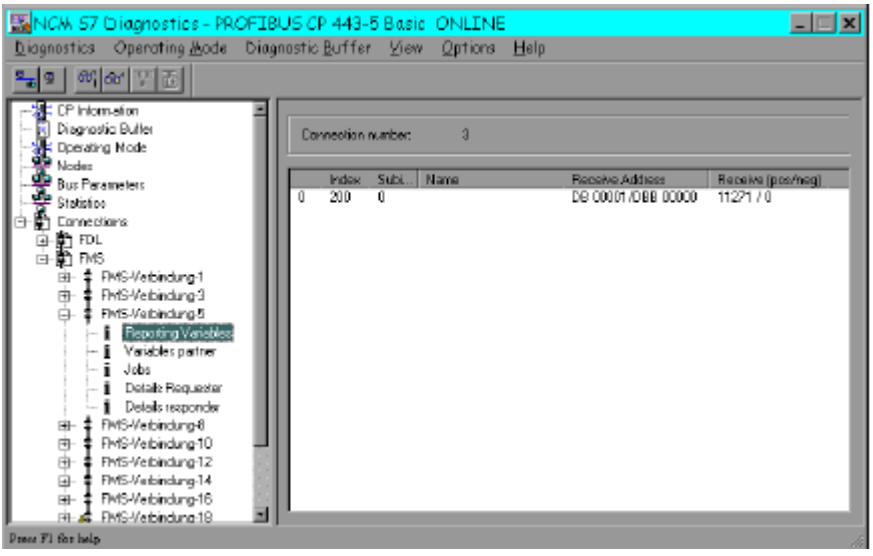
همانطور که در شکل بالای صفحه بعد مشاهده میشود پس از ارتباط On Line با شبکه میتوان موارد زیر را با اطلاعات دقیق مشاهده کرد:

- تمام ایستگاههایی که ارتباط FDL دارند
- تمام ایستگاههایی که ارتباط FMS دارند
- Node های شبکه
- پارامترهای باس
- Diagnostic Buffer

شکل زیر اطلاعات دقیق مربوط به یک ایستگاه FMS را با پارامترهای آن در حالت OnLine نشان میدهد.



وقتی روی یکی از ایستگاههای پنجره فوق کلیک کنیم شاخه های دیگری مانند شکل زیر ظاهر میشود . در این پنجره شماره دیتابلاک ایندکس شده برای Report و میزان دیتای دریافت شده نشان داده میشود.



ب) استفاده از نرم افزار Step7 برای مشاهده وضعیت

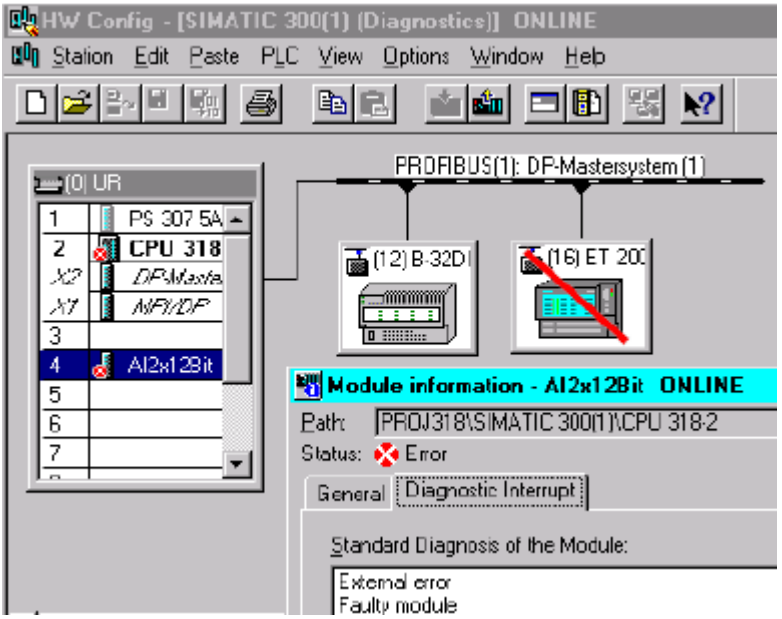
وقتی سخت افزار و شبکه ای توسط Step7 پیکر بندی و به آن دانلود شده باشد اگر با استفاده از زیر برنامه Hwconfig توسط منوی Station > Open Online پنجره OnLine را مشاهده کنیم وضعیت کارتها و مدول های مختلف را همراه با علائمی مانند زیر مشاهده خواهیم کرد.

- این علامت نشان دهنده آنست که CPU کارت یا مدول را نشناخته بعنوان مثال آنچه به CPU دانلود شده با آنچه در عمل وجود دارد یکی نیست.

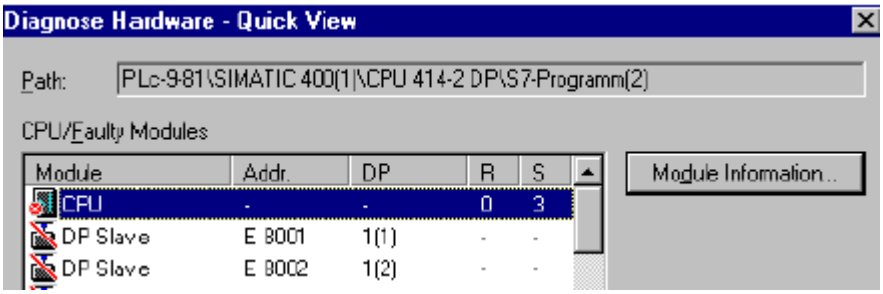
- این علامت نشان دهنده وجود اشکال در مدول است یعنی CPU مدول را شناخته ولی بعنوان مثال نمیتواند از آن دیتا بگیرد یا به آن دیتا بفرستد.

- کارت یا مدول قابلیت Diagnostic ندارد و CPU نمیتواند وضعیت آنرا بصورت Online نشان دهد.

بعنوان مثال در شکل زیر کارت AI دچار اشکال است و در عین حال CPU نتوانسته ET200 قرار گرفته روی شبکه DP را بشناسد . در چنین مواردی معمولا با کلیک کردن روی مدول اطلاعات دقیقتری ارائه خواهند شد.



برای مشاهده وضعیت کلی Node های متصل به شبکه میتوان از Quick View استفاده کرد وهمه آنها را در پنجره ای مانند زیر مشاهده نمود سپس اطلاعات دقیق را با کلیک کردن روی وسیله مورد نظر بدست آورد.



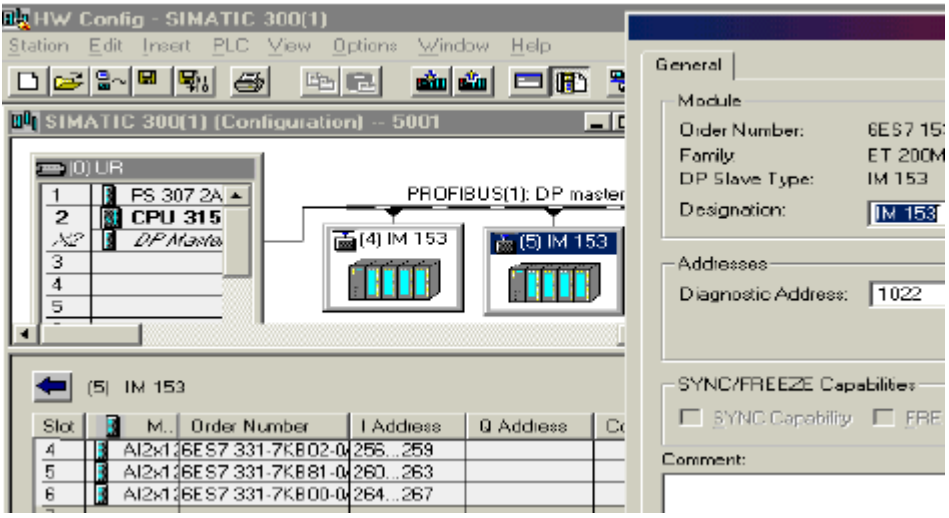
۱۰-۳ مدیریت خطا با استفاده از نرم افزارStep7

همانطور که میدانیم در موقع بروز خطا کار برنامه اصلی CPU یعنی OB1 توسط وقفه ای که به سیستم اعمال میشود قطع شده و اکثراً CPU به مد Stop رفته چراغ SF روی آن روشن میگردد. بدیهی است در چنین شرایطی اتصال به PLC و مشاهده محتویات Diagnostic Buffer آن میتواند تا حدی به تشخیص و رفع عیب کمک کند که البته مستلزم صرف زمان است . زمانی که هر ثانیه آن در فرآیند ارزشمند و گرانبهاست . روش بهتر استفاده از راهکارهای مدیریت خطا توسط برنامه نویسی وقفه هاست کاری که لازم است قبل از ایجاد خطا انجام شود تا در صورت بروز آن ، سیستم به مسیر صحیحی هدایت شده و در کمترین زمان ممکن اطلاعات دقیق خطا در اختیار کاربر قرار گیرد.

این بحث کلی است و اختصاص به شبکه ندارد . Step7 بلاک هایی را تحت عنوان OB های خاص برای این منظور تعبیه کرده است یک برنامه نویس حرفه ای ،باید بتواند از اینOB ها ،برای لینک با قسمت HMI (مونیتورینگ) وسایر کاربردها استفاده نماید. غالب این OB ها ،به گونه ای هستند که اگر صدا زده شوند و Program نشده باشند،CPU به مد STOP می رود و این می تواند باعث سردرگمی کاربر شود. لذا کار با OB های مختلف جزو ملزومات برنامه نویسی S7 در تمامی سطوح می باشد. باید توجه داشت که OB های مربوط به خطا ها بیشترین درجه اولویت را دارند و به این علت میتوانند کار OB1 یا سایر OB ها را قطع کنند. جزئیات مربوط به نحوه کار با این OB ها را باید در مراجع دیگر جستجو کرد در اینجا ما صرفا به OB های مرتبط با خطاهای شبکه می پردازیم و برای تکمیل مطلب خواننده محترم را به *کتاب راهنمای جامع Step7 جلد اول تالیف محمد رضا ماهر* ارجاع میدهیم. بسیاری از OB های بحث شده در کتاب مزبور در شبکه Profibus نیز کاربرد دارند . در اینجا صرفا به دو OB برا ی شناسایی خطا همراه با مثال اشاره می کنیم.

برنامه نویسی OB86 (خطای DP Slave)

OB86 برای آشکار سازی خطاهایی مانند نبود مدول IM یا خرابی یا قطع شدن کابل آن وخطای DP Slave در روی باس DP بکار میرود فرض کنید پروژه ای با CPU315-2DP داریم که دو ET 200M توسط Profibus به آن متصل هستند. آدرس Diagnostic این تجهیزات را با کلیک راست ماوس در بخش Properties میتوان دید که در این مثال این آدرسها 1022, 1020 میباشند.



اکنون OB86 را در پوشه Blocks ایجاد کرده و آنرا باز میکنیم سپس با استفاده از دیتای محلی OB86_MDL_ADDR که در جدول Local Data در بالای بلاک معرفی شده میتوانیم برنامه زیر را بنویسیم تا در صورت بروز خطا درهر کدام از ET ها بیتهای جداگانه ای را در دیتا بلاک ست کند. این بیت ها میتوانند به Tag در سیستم HMI متصل شده و کاربر را از وضعیت آگاه سازد.

```
L    OB86_MDL_ADDR
L    1022
==I
S     DB1.DBX0.0

L    OB86_MDL_ADDR
L    1021
==I
S    DB1.DBX0.1
```

برنامه نویسی OB122 (خطای دسترسی به I/O)

OB122 برای آشکار سازی خطاهای دسترسی به I/O ها بکار میرود چه این خطا ناشی از خراب بودن مدول باشد چه اشکال در شناخته نشدن آدرس توسط CPU باشد.

اگر برای آدرسهای مربوط به یکی از ET های مثال قبلی بخواهیم OB122 را بنویسیم میتوان با استفاده از دیتای محلی OB122_MEM_ADDR که در جدول Local Data در بالای بلاک معرفی شده برنامه زیر را بنویسیم تا در صورت بروز خطا در ورودی ET ها فلاگی ست شود. به آدرس کارتهای I/O در پایین شکل صفحه قبل دقت شود.

```
L   OB122_MEM_ADDR
L   256
>=I
A(
L   OB122_MEM_ADDR
L   268
<=I
)
S   “Err”
```

ضمیمه ۱

مروری بر مفاهیم و اصطلاحات شبکه های صنعتی

مشتمل بر :

- **توپولوژی های مختلف شبکه**
- **واسط های انتقال در شبکه**
- **تکنیک های دسترسی به شبکه**
- **لایه های شبکه و مدل OSI**
- **فاصله همینگ**

توپولوژی های مختلف شبکه

آرایش فیزیکی اجزای یک شبکه و نحوه اتصال آنها به یکدیگر را توپولوژی شبکه میگویند و به سه دسته اصلی BUS و Ring و Star تقسیم میشود با ترکیب این سه دسته میتوان آرایش های دیگری مانند Mesh و Tree داشت پس بطور کلی توپولوژی شبکه میتواند یکی از انواع زیر باشد:

۱. توپولوژی Bus یا Line
۲. توپولوژی Ring
۳. توپولوژی Star
٤. توپولوژی Mesh
٥. توپولوژی Tree

۱- توپولوژی Bus

- متداولترین و ساده ترین شکل شبکه است.
- نصب آن راحت و هزینه اش کمتر از سایر انواع است.
- براحتی قابل توسعه است.
- دوطرف کابل (ابتدا و انتها) لازم است با مقاومت Terminator بسته شود تا از بازتاب سیگنال جلوگیری گردد.
- عیب یابی آن مشکل است.
- قابلیت اطمینان آن بالا نیست زیرا بسته به نوع آن با قطع شدن کابل یا ایستگاه ممکن است کل شبکه از کار بیفتد.

باس ممکن است بصورتهای مختلفی بسته شود. سه نوع توپولوژی باس در زیر مورد مقایسه قرار گرفته است.

توپولوژی باس بصورت Tap and Drop

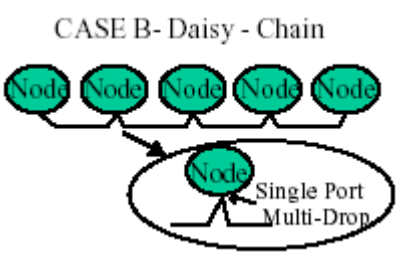
در این توپولوژی همانطور که در شکل زیر مشاهده میشود هر ایستگاه توسط کانکتورهای T شکل به باس متصل شده است. کابلی که ارتباط ایستگاه با باس را برقرار میکند کابل Drop نامیده میشود در این روش :

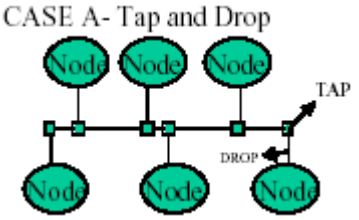
- اگر کابل Drop قطع شود یا ایستگاهی مشکل پیدا کند فقط همان ایستگاه از شبکه خارج میشود ولی کل شبکه قطع نمیشود.
- اگر کابل اصلی شبکه (Trunk) قطع شود کل شبکه از کار می افتد.
- طول کابل Drop نباید از حدی بیشتر باشد.
- فاصله بین دو Tap نباید از حدی کمتر باشد.
- باس باید با مقاومت Termination بسته شود.

توپولوژی باس بصورت Daisy Chain

در این روش کابل Drop وجود ندارد باس شبکه از همان نقطه اتصال به ایستگاه ادامه می یابد بعبارت دیگر این اتصال در کانکتور متصل به ایستگاه برقرار میشود. در این روش:

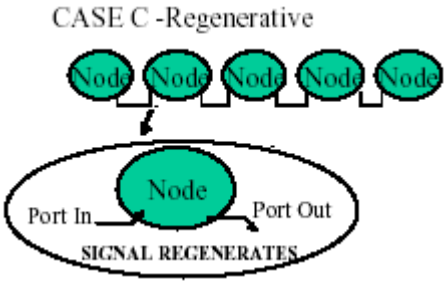
- اگر یک ایستگاه مشکل پیدا کند شبکه قطع نمی شود.
- در اینحالت کابل Drop وجود ندارد.اگر کابل اصلی در نقطه ای قطع شود کل شبکه از کار می افتد.
- باس باید با مقاومت Termination بسته شود.





توپولوژی باس بصورت Regenerative

دراین روش در هر ایستگاه کابل ورودی به یک پورت و کابل خروجی به پورت دیگردستگاه متصل میشود.بنابر این برخلاف دو نوع قبلی در اینجا باس یک پارچه نیست . در این روش اگر یک ایستگاه خراب شود کل شبکه از کار می افتد ولی اگر کابل قطع شود مابقی شبکه میتواند کار کند.

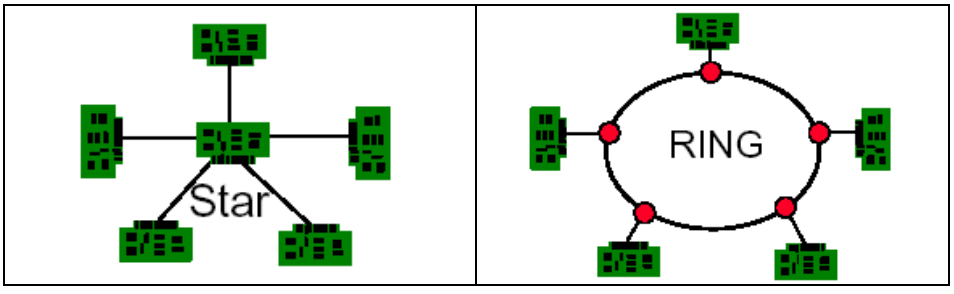


۲- توپولوژی Ring

- اگر ابتدا و انتهای شبکه با توپولوژی باس را بصورت مناسب بهم متصل کنیم توپولوژی Ring بوجود می آید.در واقع هر Node با دو Node در دو طرف خود بصورت Point To Point مرتبط است.
- قابلیت اطمینان آن بیشتر از نوع باس است.در برخی کاربردها شبکه Ring در صورت قطع شدن کابل ارتباطی بصورت باس به کار خود ادامه میدهد.اگر قابلیت اطمینان خیلی بالا نیاز باشد میتوان با سخت افزار مناسب دو Ring ایجاد کرد که بصورت Redundant کار کنند.

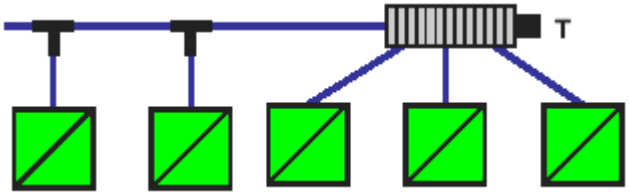
۳- توپولوژی Star

- در این روش هر ایستگاه با کابل مجزا به ایستگاه مرکزی که میتواند یک Hub باشد متصل میشود.
- اگر کابل بین یک ایستگاه تا ایستگاه مرکزی قطع شود فقط همان ایستگاه کنار میرود پس نسبت به Bus و Ring قابلیت اطمینان بالاتری دارد ولی اگر ایستگاه مرکزی قطع شود کل شبکه از کار می افتد.
- پیاده سازی آن اگرچه ساده است ولی نسبت به Bus و Ring کابل کشی بیشتری نیاز دارد.



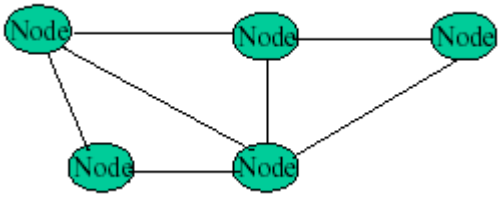
٤- توپولوژی Tree

- این توپولوژی که شکل درختی دارد ترکیبی است از توپولوژی باس و توپولوژی ستاره و مزایای آن دو را با هم داراست.



٥- توپولوژی Mesh یا Hybrid

- در این روش تمام ایستگاهها با یکدیگر از طریق کابل مستقلی ارتباط دارند.
- قابلیت اطمینان آن بسیار بالاست.
- عیب یابی آن مشکل و انعطاف پذیری آن کم است.
- در عمل بندرت استفاده میشود.



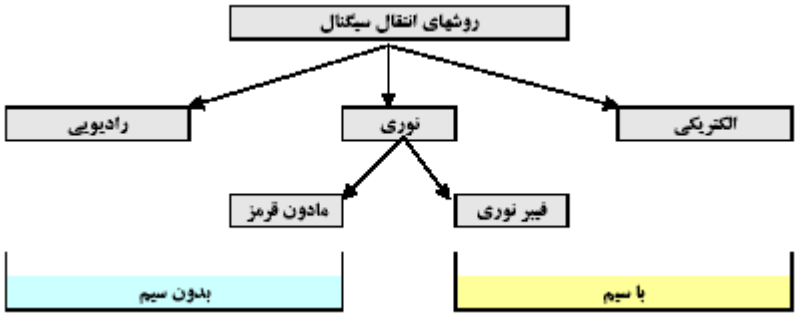
مقایسه توپولوژی های سه گانه اصلی

جدول زیر توپولوژی های BUS و Star و Ring را از برخی جنبه ها مورد مقایسه قرار داده است:

	Star	Ring	Bus
انعطاف پذیری	پایین	پایین	بالا
هزینه	زیاد	متوسط	کم
عیب یابی	راحت تر	مشکل	مشکل
قابلیت اطمینان	زیاد	متوسط	کم

واسط های انتقال در شبکه

واسط های انتقال (Transmission Media) وسایل و عوامل فیزیکی هستند که سیگنال را در یک شبکه جابجا میکنند مانند یک کابل الکتریکی. شکل زیر دسته بندی روشهای انتقال سیگنال را نشان میدهد. در همه این موارد در سمت فرستنده دیتا به انرژی (الکتریسیته و نور و ..) و در سمت گیرنده ، انرژی به دیتا تبدیل میگردد.

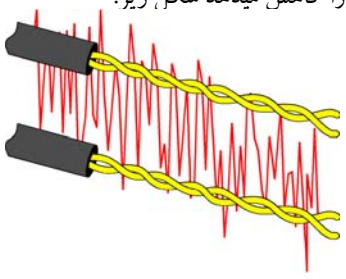


۱- انتقال سیگنال به روش الکتریکی

در این روش از کابل الکتریکی مسی استفاده میشود که دارای دو دسته کلی زیر است یکی زوج سیم بهم تابیده و دیگری کواکسیال.

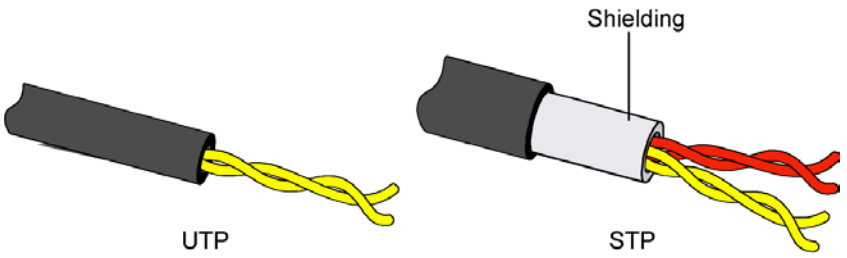
I. کابل با زوج سیم بهم تابیده (TP)

- کابل با زوج سیم به هم تابیده یا Twisted Pair از رشته سیم های روکش دار مسی تشکیل شده که هر جفت آن به هم تابیده شده اند.
- تابیده شدن جفت سیم تا حد زیادی پدیده تداخل الکتریکی با جفت سیم های مجاور که به پدیده Crosstalk موسوم است را کاهش میدهد شکل زیر:



Crosstalk در کابل با زوج سیم بهم تابیده (TP)

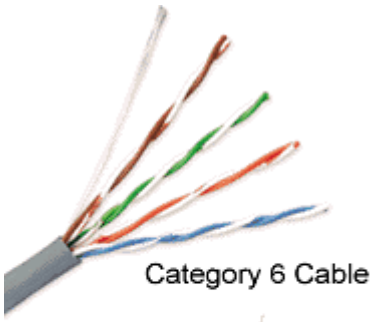
- کابل های TP به به دو دسته بدون شیلد (UTP) و شیلد دار (STP) تقسیم میشوند.
- نوع بدون شیلد UTP یا (Unshielded Twisted Pair) دارای ظرفیت محدودتری است و سرعت انتقال دیتا در آن کمتر از نوع شیلد دار است و در مقابل نویز آسیب پذیر تر است .



مقایسه کابل UTP و STP

- نوع شیلد دار STP یا (Shielded Twisted Pair) دارای انواع مختلفی است ممکن است کل زوجها فقط یک شیلد داشته باشند و ممکن است هر زوج یک شیلد مجزا داشته باشد از اینرو به دسته ها یا اصطلاحاً Category های مختلفی تقسیم میگردد که سرعت و ظرفیت انتقال آنها متفاوت است.
- Category 1 فقط برای ارتباط صوتی بکار میرود و سرعت انتقال در آن کمتر از 1 Mbps است.
- Category 2 برای دیتا ولی با سرعت کم میتواند بکار رود سرعت انتقال در آن 4 Mbps است.
- Category 3 میتواند برای شبکه های کامپیوتری بکار رود سرعت انتقال در آن 10 Mbps است.
- Category 4 برای انتقال دیتا با سرعت 20 Mbps استفاده میشود.
- Category 5 برای انتقال دیتا با سرعت 100 Mbps بکار میرود و در شبکه های محلی LAN مرسوم است.
- Category 5E که زوج سیم ها در آن نسبت به Cate 5 بیشتر به هم تابیده شده اند میتواند برای انتقال دیتا با سرعت 1000 Mbps بکار میرود.

- Category 6 سرعت انتقال دیتا در آن ٦ برابر Cate 5 است.
- Category 7 دارای سرعت انتقال تا حد یک گیگا بیت میباشد



Category 6 Cable

Category 7 Cable

II . کابل Coaxial

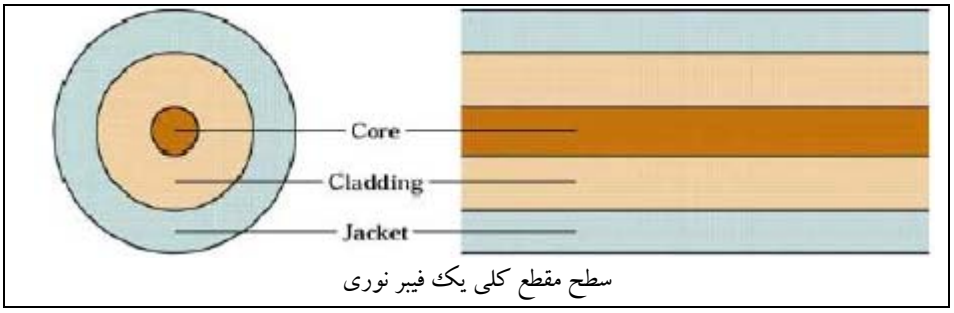
- کابل کوآکسیال یکی از اولین انواع واسط هایی است که در شبکه LAN استفاده میشود.
- معمولا برای شبکه اترنت با توپولوژی باس استفاده میشود.
- این کابل مانند شکل زیر ازدو رشته هم مرکز تشکیل شده که یکی نقش شیلد را نیز دارد.
- شیلد کابل مقاومت آن را در مقابل نویز پذیری زیاد میکند.

۲- انتقال سیگنال توسط فیبر نوری

در این روش از کابل نوری (Fiber Optic) استفاده میشود. اولین انواع کابل نوری از حدود ۱۹۷۶ در آمریکا بکار گرفته شد. در ابتدا تلفات سیگنال در آن زیاد بود و برای مسافتهای کوتاه بکار میرفت با بهبود متریال و تکنولوژی ساخت ،کاربرد آن بتدریج توسعه پیدا کرد و مسافت هایی در حد چندین کیلومتر را پوشش داد.

I. ساختار فیبر نوری

فیبر نوری از یک هسته (Core) که از جنس شیشه یا پلاستیک است و یک پوشش Cladding که هسته را احاطه کرده تشکیل شده است هسته نور را انتقال میدهد و Cladding بازتاب نور را در طول فیبر فراهم میسازد.فیبر دارای یک پوشش نهایی است که Jacket نامیده میشود.

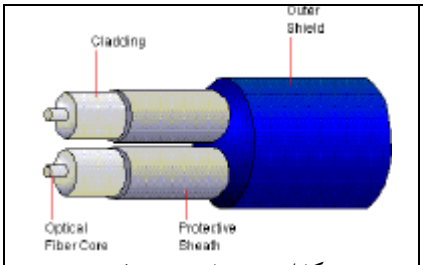


سطح مقطع کلی یک فیبر نوری

بمنظور استحکام مکانیکی فیبر پوشش دیگری نیز بنام Kevlar بین Cladding و Jacket قرار میدهند.

کابل فیبر همیشه تک رشته نیست و ممکن است دارای چندین رشته باشد.

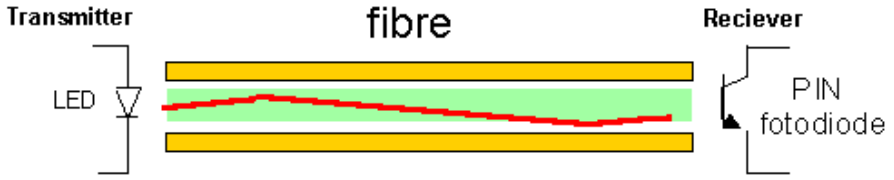
کابل نوری با یک رشته فیبر



کابل نوری با دو رشته فیبر

II. اصول انتقال سیگنال در فیبر نوری

اصول انتقال سیگنال در فیبر نوری بر اساس بازتاب نور در آن است. یک منبع نوری که میتواند یک LED باشد نور را به داخل فیبر می تاباند. نور پس از بازتاب های متوالی به گیرنده که میتواند یک فتودیود باشد میرسد. وجود نور بعنوان 1 و عدم وجودآن بعنوان 0 منطقی تلقی میگردد. بنابراین همانطور که در شکل ملاحظه میشود در سمت فرستنده سیگنال الکتریکی به نور تبدیل میگردد و درسمت گیرنده عمل عکس یعنی تبدیل نور به سیگنال الکتریکی انجام میشود.

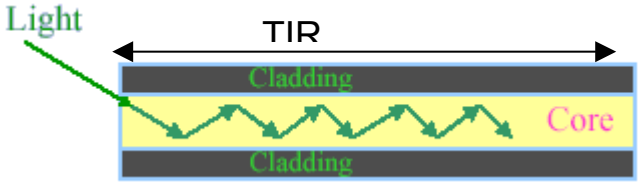


علت بازتاب نور در فیبر آنست که ضریب انکسار محیط Cladding کمتر از ضریب انکسار هسته است و این طبق قانون Snell موجب بازتاب نور به داخل فیبر میشود.ضریب انکسار با حرف n در فرمول زیر بکار میرود:

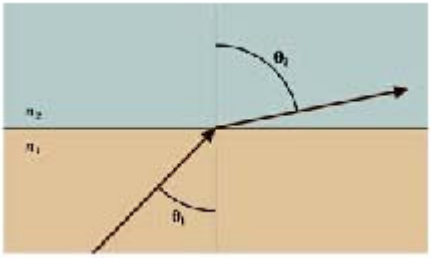
$$n1 \sin\theta1 = n2 \sin\theta2$$

ضریب انکسار یک ماده برابر است با سرعت نور در خلا تقسیم بر سرعت نور در آن ماده این ضریب برای هوا 1.0003 و برای Cladding حدود 1.46 و برای هسته فیبر حدود 1.48 میباشد.

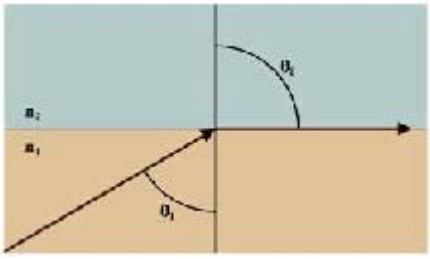
پس بطور خلاصه انتقال سیگنال بر اساس بازتاب کلی داخلی یا TIR (Total Internal Reflection) استوار است.



با توجه به شکلهای صفحه بعد میبینیم که زاویه تابش مهم است . اگر زاویه تابش از حدی که حد بحرانی نامیده میشود کمتر باشد به داخل فیبر بازتاب پیدا نمیکند و در محیط Cladding جذب میشود.

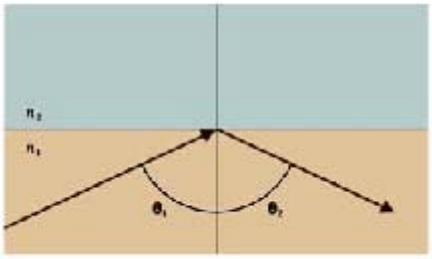


تابش با زاویه کمتر از زاویه بحرانی



تابش با زاویه ای برابر با زاویه بحرانی

پس برای زاویه تابش مناسب محدوده خاصی وجود دارد. نکته قابل توجه دیگر آنست که اگر زاویه تابش مناسب داری شیب تند باشد (یعنی نزدیک به حد بحرانی) در اینصورت تعداد دفعات باز تاب بیشتر خواهد بود و نور باید مسافت بیشتری را طی کند از اینرو سیگنال آهسته تر انتقال می یابد.



تابش با زاویه بیشتر از زاویه بحرانی

مزایای استفاده از فیبر نوری

- ظرفیت و پهنای باند بسیار بالا (بیش از 2 Gb/s برای دهها کیلومتر)
- میرایی (Attenuation) کم سیگنال در مسافتهای طولانی
- کاهش سایز و وزن نسبت به سیم مسی(حدود ۹۰ درصد کاهش)
- انتقال مطمئن دیتا بدون وجود Cross-talk
- ایمن از نظر تداخل رادیویی RFI (Radio Frequency Interference)
- ایمن از نظر تداخل امواج الکترومغناطیسی EMI (Electro Magnetic Interference)
- ایمن از نظر تخلیه الکتروستاتیکی ElectroStatic Discharge (ESD)

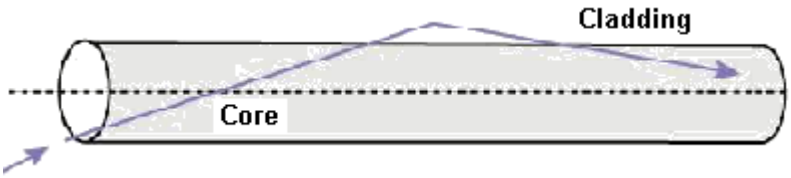
معایب استفاده از فیبر نوری

- پر هزینه است.
- نصب و راه اندازی آن نسبت به الکتریکی پیچیده تر است.
- از نظر فیزیکی آسیب پذیر است و نیاز به محافظت دارد.

دو categoryاصلی فیبر نوری

۱- Singlemode

- ضخامت هسته فیبر نوری نازک است (۱۲,۵ میکرون)
- سیگنال نوری در فیبر فقط با یک زاویه انعکاس می یابد .
- به آن monomode نیز گفته میشود.
- این روش سیگنال برای مسافتهای طولانی بهتر منتقل میشود.
- از LED بعنوان منبع نور در آن نمیتوان استفاده کرد و نیاز به Laser دارد.
- کوپل کردن فیبر با چشمه نور مشکل تر است.
- هزینه اش بیشتر است.



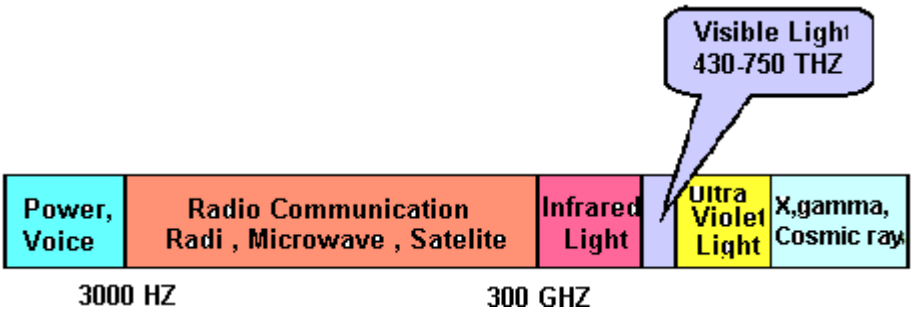
۲- Multimode

- ضخامت هسته فیبر نوری در آن ضخیم تر است (۶۲,۵ میکرون)
- چندین سیگنال نوری که هر کدام زاویه بازتاب متفاوت در هسته دارند را میتواند منتقل کند.
- برای مسافتهای نسبتا کوتاهتر بکار میرود زیرا در مسافتهای طولانی منجر به Dispersion سیگنال میشود.
- چشمه نور در آن میتواند LED باشد.
- Multimode خود به دو دسته Step- index و Graded-index تقسیم میگردد.
- نوع Step-index همان نوع استاندارد فیبر نوری است و قسمت انتقال آن شامل دو لایه Core و Cladding است.
- در نوع Graded-index فیبر چند لایه دارد که ضریب انعکاس آنها متفاوت است . این امر موجب میشود که نوربتواند با سرعتهای متفاوتی از هسته بگذرد.

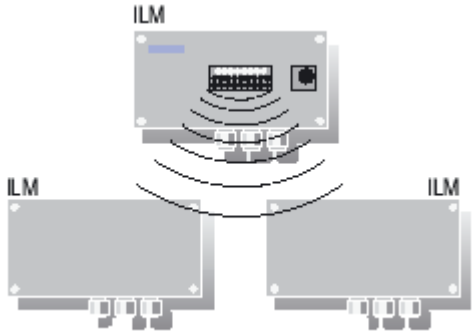
۳- انتقال سیگنال بصورت بی سیم

انتقال بدون سیم به دو روش امکان پذیر است یکی مادون قرمز و دیگری رادیویی.

I. توسط نور مادون قرمز (Infrared)

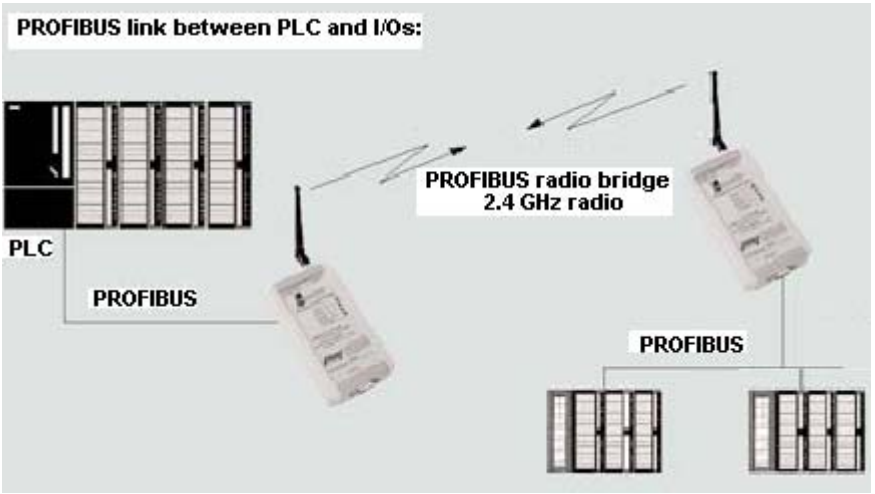


- در این روش نیاز به محیط فیزیکی خاصی نیست.
- نیاز به دید مستقیم بین فرستنده و گیرنده است.
- از موانع و دیوارها نمیگذرد.این موضوع ممکن است در برخی حالات بعنوان حسن و نه عیب تلقی شود زیرا در یک مجموعه بسته از تداخل سیگنال با سیگنالهای بیرونی جلوگیری میکند.
- پهنای باند آن وسیع نیست (100 Kbps تا 16 Mbps)
- مزیت آن عدم نیاز به سیم کشی است ولی خود سیستم هزینه زیاد دارد.
- میرایی سیگنال بستگی به کیفیت نور و شرایط اتمسفری محیط دارد.
- تداخل نویز آن کم است با این وجود نورهایی نزدیک به آن از نظر طول موج میتوانند تداخل ایجاد کنند.
- ارتباط میتواند نقطه به نقطه (Point to Point) باشد یعنی یک فرستنده و یک گیرنده.
- ارتباط میتواند نقطه به چند نقطه (Point to multiPoint) باشد یعنی یک فرستنده و چند گیرنده.
- یکی از کاربردهای این روش در ایستگاههای متحرک (Mobile Equipment)است که فاصله آنها تا ایستگاههای دیگر زیاد نیست.



II. توسط امواج رادیویی

- در این روش از امواج رادیویی استفاده میشود که از موانع و دیوارها نیز میگذرند.
- فرستنده و گیرنده روی فرکانس خاصی تنظیم میشوند.
- برد سیگنال بستگی به قدرت آن دارد. اگر با توان بالا ارسال شود برای مسافتهای طولانی قابل استفاده است.



مقایسه وسایل انتقال سیگنال

	بیسیم – مادون قرمز	کابل نوری	کابل کوآکسیال	کابل TP
آماده سازی و نصب	ساده	پیچیده	ساده	خیلی ساده
نویز پذیری	کم	ندارد	کم	زیاد برای UTP
سرعت انتقال	متوسط	عالی	خوب	خوب
هزینه	متوسط	زیاد	متوسط	کم

تکنیک های دسترسی به شبکه

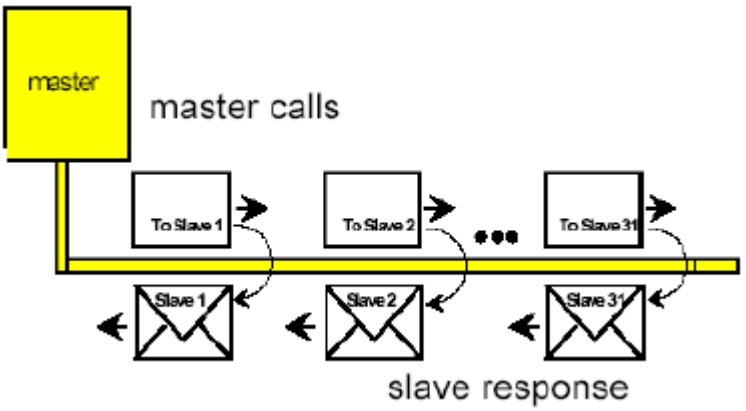
تکنیک دسترسی (Access Technique) یعنی روشی که توسط آن یک ایستگاه میتواند باس شبکه را در اختیار بگیرد و دیتا بگیرد یا بفرستد و به سه نوع زیر تقسیم میشود:

۱. Mastrer / Slave
۲. Token Pass
۳. CSMA

تکنیک دسترسی یک پیکر بندی منطقی (Logic) است و نباید آن را با توپولوژی فیزیکی اشتباه کرد . بعنوان مثال ممکن است یک شبکه با توپولوژی باس از تکنیک Master / Slave استفاده کند و شبکه دیگری با همان توپولوژی باس تکنیک Token را بکار ببرد.

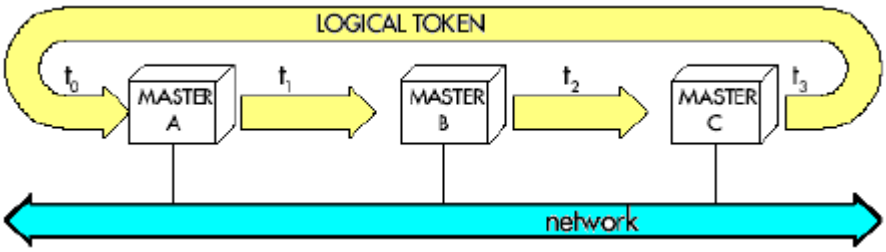
۱- تکنیک Master / Slave یا رئیس مرئوس

- فقط دارای یک Master (واحد فعال) است.
- سایر ایستگاهها بصورت Slave (واحد غیر فعال) هستند.
- باس همواره در اختیار Masterاست و Slave ها اجازه ندارند آنرا در اختیار بگیرند.
- Master است که به Slave ها اجازه میدهد دیتا بگیرند یا دیتا بفرستند.
- Master بصورت سیکلی با Slave ها یکی پس از دیگری ارتباط برقرار میکند.
- ارتباط مستقیم بین Slave ها وجود ندارد.
- هر Slave آدرس خود را میشناسد و اگر چه تمام پیام های Master را دریافت میکند ولی تنها به پیام های مخصوص به خود پاسخ میدهد.

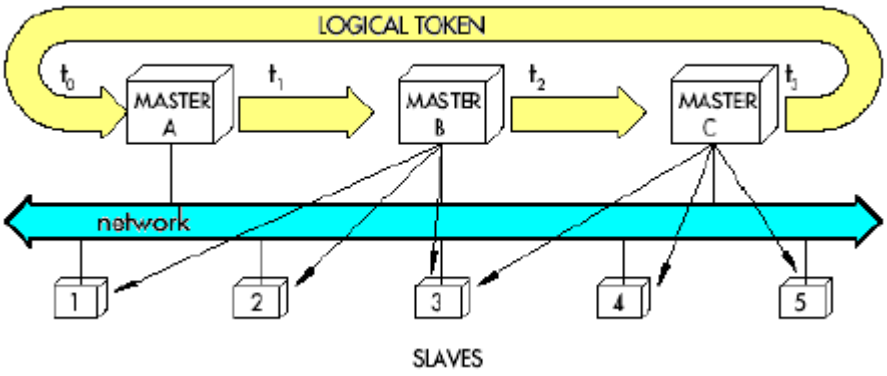


۲- تکنیک Token یا انتقال علامت

- بر خلاف روش Master / Slave میتواند چندین Master (یا عنصر Active) داشته باشد.
- در این روش یک نشانه (Token) بین ایستگاهها داخل یک حلقه میچرخد.
- به حلقه فوق Token Ring میگویند و یک حلقه منطقی است نه یک حلقه فیزیکی.
- نشانه مانند یک پاکت خالی بین ایستگاهها میچرخد.
- این پاکت در اختیار هر ایستگاهی باشد قادر است اطلاعات خود را ارسال کند.
- وقتی پاکت یعنی نشانه در اختیار یک ایستگاه قرار گرفت سایرین نمیتوانند باس را در اختیار بگیرند.
- بدلیل ویژگی فوق عملا هیچ تداخلی بین ایستگاهها در این روش رخ نمی دهد.
- ایستگاه پس از قرار دادن اطلاعات در پاکت و مشخص کردن گیرنده آنرا در حلقه رها می کند.
- هر ایستگاه وقتی پاکت به او رسید آدرس آنرا چک می کند اگر به او مربوط بود آن را بر میدارد و پاکت خالی را در حلقه رها میکند و اگر به او مربوط نبود پاکت را به ایستگاه بعدی میفرستد وسیکل به همین شکل تکرار میشود.
- جهت چرخش معمولا یک طرفه است معمولا نشانه از آدرس با شماره کمتر شروع و به آدرس هایی که دارای شماره بزرگتر هستند تحویل داده میشود. وقتی به بزرگترین آدرس رسید مجددا تحویل کوچکترین آدرس میشود.
- اگر عضوی از حلقه خارج شد یا عضو جدیدی وارد حلقه شد سیستم نوبت دهی جدید برقرار میشود.
- در توپولوژی باس میتوان از روش Token استفاده کرد که در اینصورت به آن Token Bus گفته میشود .



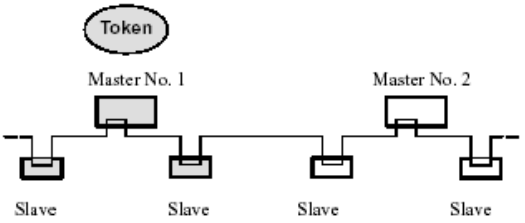
- میتوان روشهای Master / Slave و Token را با هم ترکیب و استفاده کرد که به آن تکنیک Hybrid میگویند. در این روش وقتی نشانه به یک Master رسید اجازه دارد با Slave هایش صحبت کند یا به Master های دیگر دیتا بفرستد.
- Token میتواند همزمان با Master / Slave بکار رود. در شکل زیر علاوه بر اینکه هر Master با Slave های خودش صحبت میکند با سایر Master ها نیز به روش Token می نواند تبادل دیتا داشته باشد.



در صفحه بعد مثال ساده ای از چرخش علامت بین دو Master در روش Hybrid ترسیم شده است.

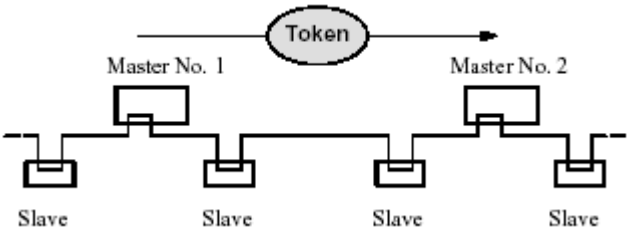
وضعیت ۱ :

نشانه در اختیار Master1 است و میتواند با Slave هایش صحبت کند. در این حالت Master2 به Slave هایش دسترسی ندارد.



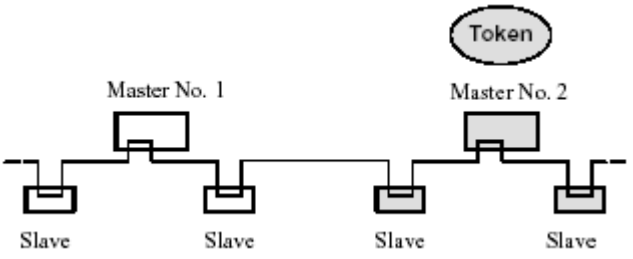
وضعیت ۲ :

Master1 نشانه را به Master2 میفرستد در این مرحله هیچکدام از Master ها به Slave هایشان دسترسی ندارند.



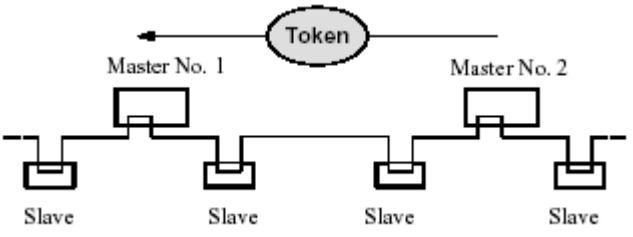
وضعیت ۳ :

نشانه در اختیار Master 2 است و میتواند با Slave هایش صحبت کند. در این حالت Master 1 به Slave هایش دسترسی ندارد.



وضعیت ٤ :

Master2 نشانه را به Master1 میفرستد در این مرحله هیچکدام از Master ها به Slave هایشان دسترسی ندارند.



۳- تکنیک CSMA یا روش گوش دادن به خط

- در روش CSMA (Carrier Sense Multiple Instance) هر وقت باس خالی باشد هر ایستگاهی میتواند آنرا در اختیار بگیرد.
- هر ایستگاه مرتبا به باس گوش میدهد تا از خالی شدن آن اطلاع حاصل کند.
- ممکن است دو ایستگاه همزمان شروع به صحبت کنند که منجر به تصادم میشودو این عیب روش فوق است.
- برای رفع مشکل فوق روش CSMA/CD بکار میرود که در آن CD مخفف Collision Detection یعنی آشکار سازی تصادم است.
- در روش CSMA/CD تصادم با تغییر سطح ولتاژ مستقیم آشکار میشود و تمام ایستگاههای شبکه از جمله ایستگاههای که دیتای آنها با هم برخورد کرده متوجه موضوع میشوند.
- ایستگاههای فرستنده پس از وقوع تصادم هردو ساکت شده و باز طی زمان نامشخصی که بطور رندوم برای هر ایستگاه محاسبه میشود اقدام به ارسال پیام میکنند.
- در طول مدت انتظار (بعد از وقوع تصادم) هیچ ایستگاه دیگری نمیتواند باس را در اختیار بگیرد .
- اگر برخوردهای متعدد در یک شبکه صورت گیرد ایستگاهها زمان انتظار بعد از برخورد را دو برابر میکنند.
- زمان های انتظار بعد از برخورد سرعت موثر شبکه را شدیداً کاهش میدهد.
- بطور خلاصه باید گفت روش CSMA/CD در شرایط عادی کارایی بسیار خوبی دارد ولی در ترافیک بالا کارایی سیستم بدلیل وجود تصادم کم میشود.
- اشکال مزبور در روش CSMA/CD منجر به ابداع روش دیگری به نام CSMA/CA شده که درآن CA مخفف Collision Avoided یا اجتناب از تصادم است.
- در روش CSMA/CD هر ایستگاه یک درجه اولویت دارد. اگر تصادم اتفاق افتاد آنکه درجه اش بالاتر است اجازه دارد باس را در اختیار بگیرد.

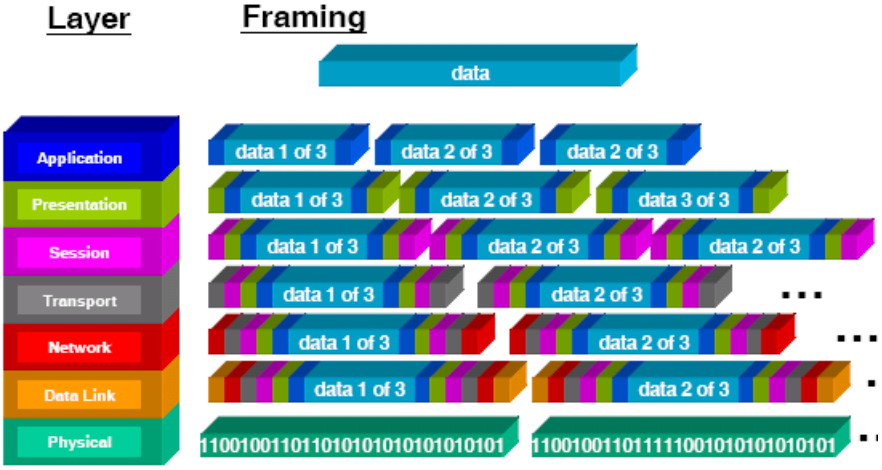
لایه های شبکه و مدل OSI

تا اینجا با ساختار فیزیکی شبکه و وسایل انتقال و تکنیک های دسترسی به شبکه آشنا شدیم اما بحث اصلی نحوه انتقال اطلاعات است. میدانیم اطلاعات بصورت دیجیتال یعنی صفر و یک منتقل میشوند یعنی یک بسته اطلاعاتی شامل تعداد زیادی صفر و یک خواهد بود. در اینجا سوالاتی مطرح میشود مانند :

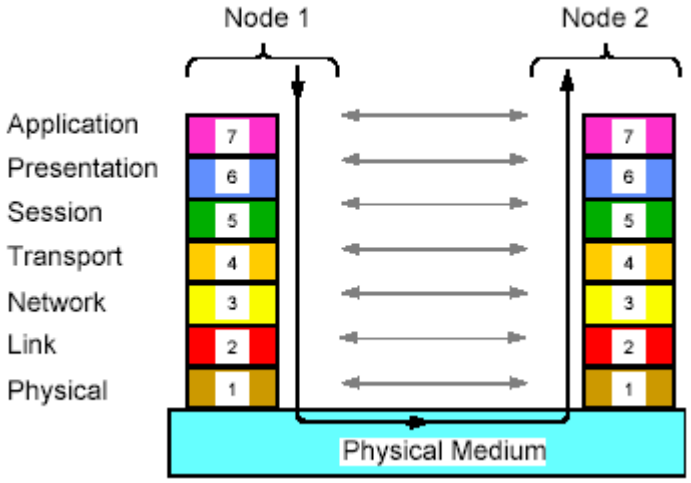
۱. اگر حجم اطلاعات زیاد باشد چگونه میتوان آن را در سمت فرستنده به بسته های کوچکتر شکست و چگونه میتوان این بسته ها را در سمت گیرنده بهم پیوند زد تا کل اطلاعات قابل شناسایی باشد؟
۲. اگر اطلاعات روی شبکه ای که تعداد زیادی ایستگاه به آن متصل هستند ارسال شود چگونه میتوان آدرس گیرنده را همراه با اطلاعات ارسال نمود تا به مقصد مورد نظر برسد؟
۳. چگونه میتوان آدرس فرستنده را همراه با آن ارسال نمود تا گیرنده پاسخ خود را به آن آدرس بفرستد؟
٤. اگر به دلیلی مانند نویز یا اشکال سخت افزاری صفر و یک داخل بسته اطلاعات تغییر کرد چگونه میتوان از وقوع این خطا مطلع شد و چگونه میتوان آن را تصحیح کرد؟

این موارد و سوالات دیگری از این دست پاسخی جز این ندارد که ارسال بسته اطلاعاتی به تنهایی کافی نیست و همراه با آن کدهای دیگری نیز باید ارسال گردد. براساس این نیاز توسط استاندارد ISO یک مدل لایه ای برای شبکه طراحی و ارائه شد که به مدل **OSI** (*Open System Interconnection)* معروف است.

این مدل از ۷ لایه تشکیل شده است . در سمت فرستنده اصل پیام به لایه ۷ داده میشود و از این لایه بتدریج به لایه های پایین تر انتقال می یابد. هر لایه کدهایی را به بسته اضافه میکند تا جوابگوی نیازهایی مانند آنچه در پرسش های صفحه قبل مطرح شد باشد تا نهایتاً به لایه ۱ که محیط انتقال است میرسد و این لایه بسته ها را بصورت صفر و یک به سمت گیرنده ارسال میکند.



در سمت گیرنده ترتیب عملیات برعکس است. یعنی بسته از لایه ۱ دریافت میشود و بتدریج به لایه های بالاتر انتقال می یابد. هر لایه کدهای اضافی را شناسایی و از بسته جدا میکند تا اینکه اصل پیام در بالاترین لایه یعنی لایه ۷ شناسایی میگردد.



اصول کار شبیه آنست که مدیر دو سازمان مختلف بخواهند با یکدیگر مکاتبه کنند . مدیر سازمان اول پیام را مینویسد و

به قسمت دبیرخانه و اداری خود میدهد در این قسمت اطلاعات دیگر مانند شماره و تاریخ و آدرس گیرنده و فرستنده به پیام اضافه شده سپس نامه به سازمان دوم ارسال میگردد. در سازمان دوم عملیات عکس انجام میشود یعنی نامه تحویل دبیرخانه و پس از ثبت و انجام مراحل اداری تحویل مدیریت سازمان میگردد.

پس بطور کلی در سمت فرستنده ترتیب از لایه بالاتر به لایه پایین تر و در سمت گیرنده ترتیب برعکس است.

در اینجا هر یک از لایه های مدل OSI را به اختصار معرفی میکنیم:

لایه ۱ - یا لایه فیزیکی

این لایه که پایین ترین لایه مدل OSI میباشد با مشخصات الکتریکی و مکانیکی محیط انتقال سرو کار دارد و در واقع بستر مناسب را برای انتقال فراهم میکند. وظیفه ارسال و دریافت بیت های بسته های دیتا بعهده این لایه است یعنی اگر فرستنده ۱ بفرستد گیرنده نیز باید 1 دریافت کند نه 0. بدیهی است این بیت ها در این لایه هیچ مفهوم خاصی ندانند. بعبارت دیگر این لایه نمیداند آنچه ارسال یا دریافت میشود اصل پیام است یا بیت های کنترلی است صرفا ارتباط را فراهم میکند. نحوه ارتباط در این لایه ممکن است بصورت سری (یک بیت در لحظه) باشد مانند RS232 . همینطور ممکن است نحوه ارتباط بصورت موازی (چند بیت در لحظه) باشد مانند RS485. قراردادهای مربوط به سطوح ولتاژ و شکل موج های ولتاژ و جریان در خط ، نوع مدولاسیون و فرکانس کار مربوط به این لایه است.

لایه ۲ یا لایه Data Link

این لایه دیگر با بیت ها سرو کار ندارد بلکه اطلاعات را بصورت بسته یا Frame می شناسد. این لایه در سمت فرستنده لایه بالاتر خود را مجبور میکند تا داده ها را به تعدادی بسته (معمولاً) چند صد بایتی تقسیم کند. سپس این بسته ها را به ترتیب برای گیرنده میفرستد.تشخیص محدوده بسته ها با این لایه است و همانطور که ذکر شد لایه فیزیکی این محدوده را تشخیص نمی دهد.

وظایف مهم لایه ۲ عبارتند از:

- تشخیص محدوده بسته ها
- آشکارسازی وتصحیح خطا
- برقراری سرویسهای ارتباطی
- کنترل جریان انتقال دیتا

الف) تشخیص محدوده بسته ها

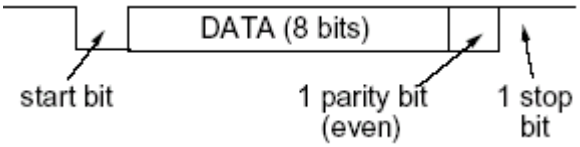
برای تشخیص محدوده بسته ها روش های مختلفی میتواند استفاده شود:

۱. درج فواصل زمانی بین بسته ها

در این روش فرستنده پس از ارسال یک بسته مدتی صبر کرده سپس بسته بعدی را میفرستد. این روش در صورتی مفید است که فرستنده و گیرنده کاملاً با هم از نظر زمانی سنکرون باشند .

۲. استفاده از بیت شروع و پایان برای هر بایت از بسته دیتا

در این روش که به آسنکرون معروف است هر کاراکتر با یک بیت که نشان دهنده آغاز است شروع میگردد و با یک بیت دیگر که نشان دهنده پایان است خاتمه می یابد.



پس برای هر بایت از دیتا حداقل ۲ بیت دیگر باید اضافه کنیم که این کار موجب افزایش بالاسری (overhead) سیگنال میگردد.

۳. استفاده از یک بایت مشخص در ابتدا و انتهای بسته دیتا

در این روش بجای آنکه برای هر بایت کدهای شروع و پایان را بکار ببریم برای کل بسته اطلاعاتی از کاراکتر خاصی که نشان دهنده شروع بسته است استفاده میکنیم. این کاراکتر همواره مقدار مشخص و ثابتی است بعنوان مثال در پروتکل HDLC از کد 01111110 استفاده میگردد.بدیهی است گیرنده در ابتدا باید به دنبال این کد بگردد و پس از یافتن آن ، اطلاعات موجود در بسته را بخواند.این روش نسبت به روش قبلی بالاسری کمتری را به سیگنال تحمیل میکند.

میتوان از ترکیب این روشها استفاده کرد مثلا هر بایت دارای بیت شروع و پایان باشد و کل بسته کاراکتر شروع و پایان داشته باشد.باید توجه داشت مدلOSI صرفاً وظایف لایه ۲ را مشخص کرده است و درمورد اینکه نحوه تشخیص بسته دیتا چگونه باشد دخالتی نمیکند . طراح میتواند ازپروتکل های استاندارد برای این منظور استفاده کند.

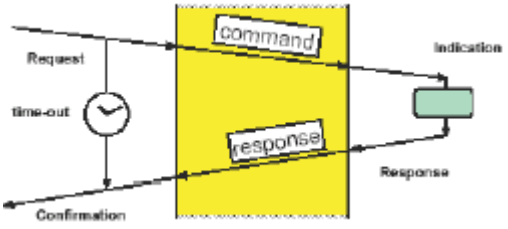
ب) آشکار سازی و تصحیح خطا

وظیفه دیگر لایه دوم آشکار سازی خطا میباشد.خطاهایی مانند :

- گم شدن بسته دیتا
- خراب شدن بسته دیتا

گم شدن بسته دیتا

منظور از گم شدن بسته دیتا اینست که که بدلیل وقوع اشکال سخت افزاری گیرنده نتواند بسته را تحویل بگیرد. برای آشکار سازی این خطا فرستنده با ارسال هر بسته تایمر خاصی را فعال میکند اگر در طول زمان تعیین شده تاییدیه از طرف



گیرنده نرسید فرستنده مجددا اقدام به ارسال همان بسته میکند و اینکار را به تعداد دفعات مشخصی که ازقبل برایش تعریف شده تکرار میکند. باید توجه داشت که در اینحالت بسته با همان شماره ارسال میگردد و لایه دوم گیرنده کنترل میکند که اگر شماره بسته تکراری باشد به لایه بالاتر تحویل داده نشود.

خراب شدن بسته دیتا

منظور از خراب شدن بسته دیتا اینست که در اینحالت گیرنده بسته ای دریافت کرده (پس برخلاف حالت قبل بسته گم نشده است) ولی چگونه میتوان اطمینان داشت که این بسته در حین ارسال دچار اشکال نشده است. میدانیم عواملی مانند نویز و تضعیف میتوانند منجر به خراب شدن بیت های ارسالی شوند. پس به نحوی باید این خطا را کنترل نمود.
یک روش برای آشکار سازی خطا استفاده از بیت Parity همراه با دیتا میباشد. در این روش در انتهای هر بایت از دیتا یک بیت به نام Parity مانند شکل صفحه قبل اضافه می شود که میتواند یکی از دو نوع زیر باشد:

- Even Parity : اگر تعداد یک های موجود در بایت دیتا با احتساب خود بیت Parity زوج باشد این بیت ۱ میشود. بعنوان مثال اگر بایت بصورت 01110011 باشد در اینصورت تعداد یکها با احتساب خود Parity زوج خواهد بود و کد ارسالی بصورت 011100111خواهد بود. یا اگر مثلا بایت بصورت 11101101 باشد در اینصورت کد ارسالی بصورت 111011010 خواهد بود.
- Odd Parity:اگر تعداد یک های موجود در بایت دیتا فرد باشد بیت Parity یک خواهد بود.

ویژگی آشکار سازی خطا به فاصله همینگ(HD) آن بستگی دارد.منظور از فاصله همینگ تعداد بیت های متفاوت در دو دیتای متوالی میباشد .بعنوان مثال دو مقدار 110010 و 110100 که ترکیب دیتا همراه با کد Parity هستند دارای فاصله همینگ ۲ میباشند زیرا دو بیت آنها متفاوت است

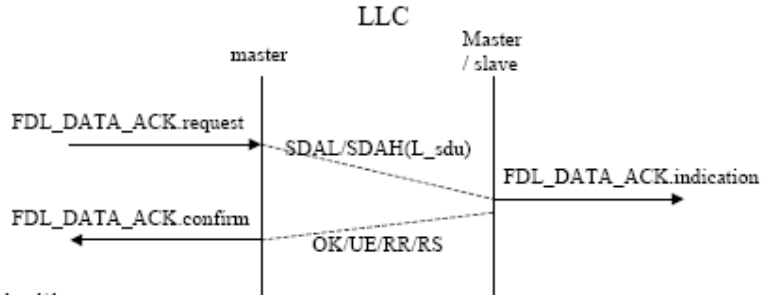
اگر خواننده محترم نیاز به اطلاعات دقیقتری در مورد فاصله همینگ دارد میتوان موضوع را در انتهای این ضمیمه مطالعه نماید.

ج) سرویس های ارتباطی

از جمله وظایف دیگر لایه ۲ برقراری سرویس های ارتباطی است. که عبارتند از :

سرویس SDA

توسط این سرویس دیتا از یک ایستگاه به ایستگاه دیگر (Peer to Peer) منتقل شده و پس از اتمام ارسال تایید دریافت از طرف گیرنده به فرستنده اعلام میشود. این سرویس در DP کاربرد ندارد.



سرویس SDN

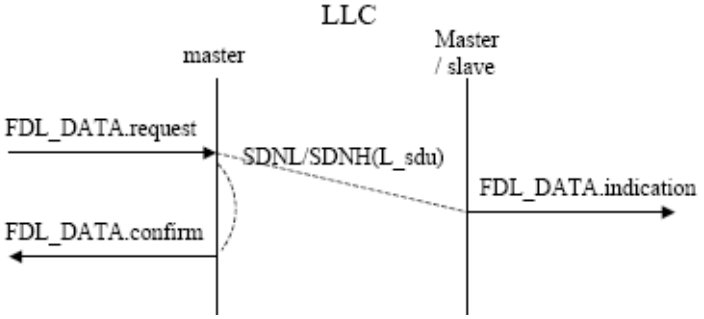
توسط این سرویس دیتا از یک ایستگاه

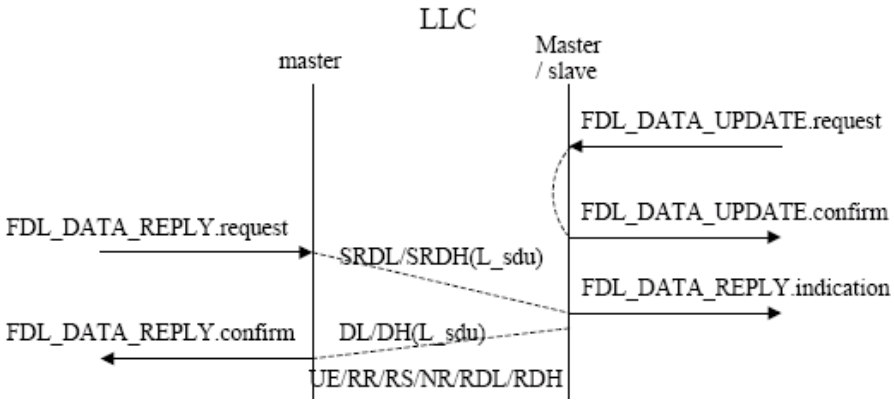
به یک ایستگاه دیگر ارسال میشود (Peer to Peer)

به چند ایستگاه مورد نظر ارسال میشود که به آن Multicast میگویند

به همه ایستگاههای متصل به باس ارسال میشود که به آن Broadcast میگویند

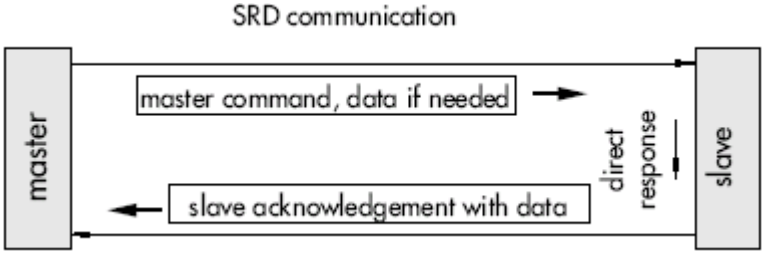
پس از اتمام ارسال ، فرستنده پیغامی مبنی بر اتمام ارسال دریافت میکند ولی از اینکه ایستگاه یا ایستگاههای مورد نظر دیتا را دریافت کرده اند یا نه مطلع نمیشود زیرا دریافت دیتا ،Acknowledge نمی شود.





سرویس SRD

توسط این سرویس دیتا از یک ایستگاه به یک ایستگاه دیگر منتقل شده و بلافاصله از آن ایستگاه دیتا درخواست میشود. فرمان Request از قبل آماده شده و بدنبال فرمان Send ارسال میشود . این کار سرعت عملکرد سیستم را بالا میبرد. اگر در حین کار خطایی رخ دهد تبادل دیتا تکرار میشود.



از سرویس های سه گانه فوق :

پروفی باس DP فقط از سرویسهای SRD و SDN استفاده میکند.

پروفی باس PA فقط از سرویسهای SRD و SDN استفاده میکند.

پروفی باس FMS از هر سه سرویس SDA و SDN و SRD استفاده میکند.

ج) کنترل جریان Flow Control

از وظایف دیگر لایه ۲ کنترل جریان ارسال بسته هایی است که بدنبال هم ارسال میشوند.اگر به دلیلی مانند عدم تطابق سرعت فرستنده و گیرنده ، بافر موجود در گیرنده پر شود باید به نحوی فرستنده را مطلع کند تا از ارسال دیتای جدید خودداری نماید. این موضوع میتواند بطریق نرم افزاری توسط سیگنال Xoff یا اینکه بصورت سخت افزاری مثلا با سیگنالهای سخت افزاری RTS,CTS در RS232 به فرستنده اطلاع داده شود.

با توجه به توضیحاتی که در مورد لایه ۲ داده شد بطور خلاصه میتوان گفت: وظیفه لایه ۲ برقرای ارتباط سالم و بدون خطا بین دو ایستگاه است.

لایه ۳ یا لایه Network

رفع ترافیک و تراکم بسته ها و جلوگیری از بوجود آمدن این مشکل در شبکه بعهده لایه ۳ میباشد. اگر در مسیر بسته دیتا شبکه جدیدی وجود داشته باشد که از نظر پروتکل با شبکه اول فرق کند ایجاد تطابق بعهده این لایه است.
بطور خلاصه : وظیفه این لایه مسیر یابی و هدایت صحیح بسته ها در شبکه است.

لایه ۴ یا لایهTransport

تقسیم داده ها به بسته های کوچکتر که قابل انتقال باشد بعهده این لایه است.

لایه ۵ یا لایه Session

توسط این لایه امکان ایجاد ارتباط مشترک بین چند کاربر متفاوت در شبکه فراهم میشود. مدیریت کنترل ارتباطات بعهده این لایه است بعنوان مثال در روش CSMA جلوگیری از همزمانی شروع بعهده این لایه است. از وظایف دیگر این لایه همزمان سازی و اضافه کردن یک سری نقاط Check Point است تا در صورتی که در وسط کار ارتباط قطع شد لازم نباشد دیتاها از اول ارسال شوند بلکه از جایی که قطع شده ادامه میدهد. این ویژگی برای انتقال اطلاعات حجیم بسیار مفید است.

لایه ۶ یا لایه Presentation

وظیفه اصلی این لایه فرمت بندی و کدگذاری با استاندارد مشخصی روی دیتا هاست تا اطلاعات دریافتی توسط گیرنده قابل استفاده باشد. فشرده سازی و رمز گذاری بمنظور حفاظت اطلاعات نیز بعهده این لایه است.

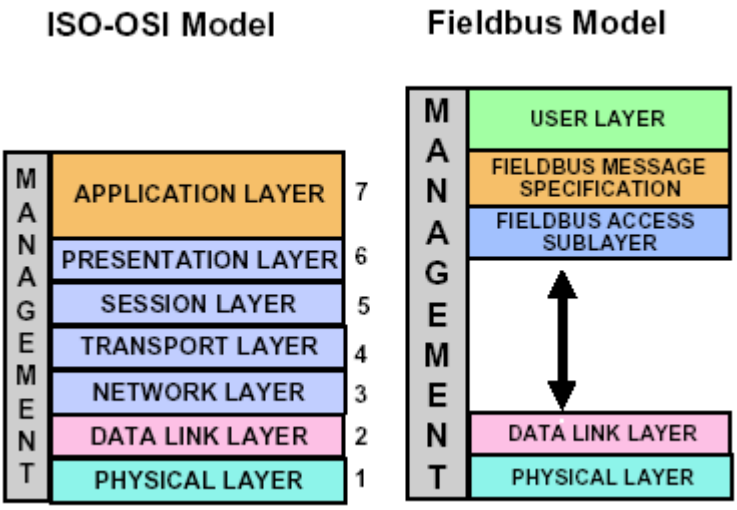
لایه ۷ یا لایه Application

این لایه که بالاترین لایه است واسط ارتباطی بین کاربر (برای ارسال یا دریافت) با سایر لایه هاست. ساختار این لایه به نوع سیستم عامل بستگی دارد. خدمات مربوط به انتقال فایل ، پست الکترونیکی در این لایه قراردارد . پروتکل های FTP و HTTP را میتوان برای این لایه مثال زد.

لایه های OSI در فیلدباس

لایه های مدل OSI که در صفحات قبل ذکر شد ممکن است همگی در یک سیستم بکار گرفته نشوند. فیلد باس نیز ازجمله این سیستم هاست.

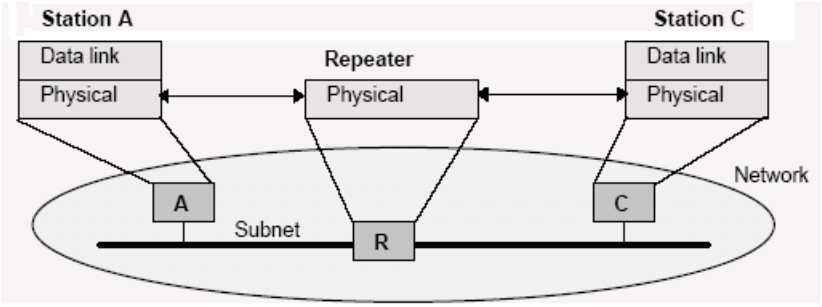
فیلدباس در مدل خود صرفاً از لایه های ۱و۲و۷ استفاده میکند. بعبارت دیگر لایه های میانی حذف شده اند و وظیفه آنها عمدتا به لایه دوم محول شده است.



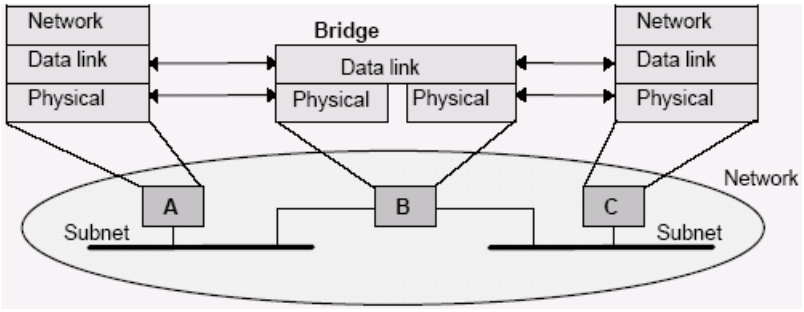
وسایل تطبیق دهنده لایه ها

همانطور که ذکر شد OSI یک مدل است نه یک استاندارد. از اینرو ممکن است فرستنده و گیرنده اگرچه هردو ازمدل OSI استفاده میکنند ولی استانداردهای متفاوتی را در لایه های خود برای کد گذاری و بسته بندی دیتاها بکار ببرند. در این حالات برای امکان ارتباط بین دو شبکه باید از وسایل تطبیق دهنده که سخت افزاری هستند استفاده کنیم. این وسایل بترتیب در زیر ذکر شده اند:

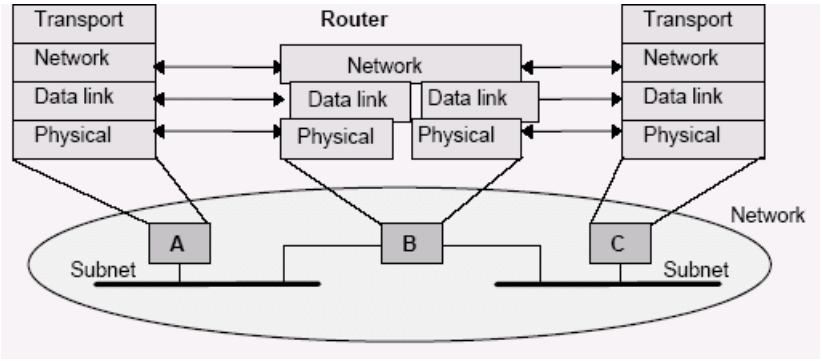
- وقتی لایه ها مشابه باشند برای تقویت سیگنال از Repeater استفاده میکنیم. دراینحالت Repeater بیشتر نقش تقویت کردن دارد تا نقش تطبیق دادن.



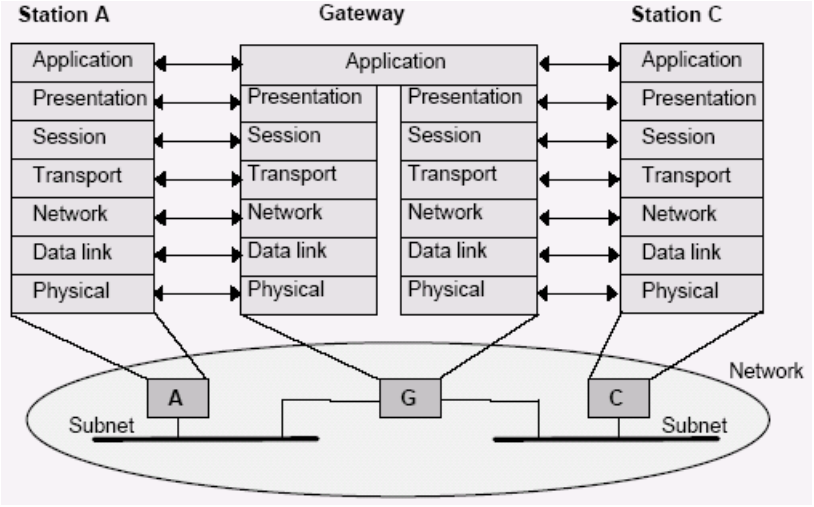
- وقتی لایه ۱ متفاوت باشد از Bridge استفاده میکنیم



- وقتی لایه های ۱و۲ متفاوت باشند از Router استفاده میکنیم



- وقتی لایه های بالاتر نیز متفاوت باشند از Gateway استفاده میکنیم



فاصله همینگ و نقش آن در تشخیص و اصلاح خطا

فاصله همینگ (Hamming Distance) چیست؟

- می دانیم یک کد دیجیتال بصورت تعدادی از صفر و یک ظاهر می شود مانند 10111001
- و میدانیم که در ارسال اطلاعات کدها بصورت رشته های جدا جدا و بدنبال هم ارسال می شوند :

10010010 رشته اول

1010000 رشته دوم

- در دو کد متوالی تعداد موقعیتهایی از بیتها که با یکدیگر متفاوتند فاصله همینگ خوانده می شود

مثال

کد اول	10010010	10010010	10010010
کد دوم	10010000	10010001	10011100
فاصله همینگ	HD=1	HD=2	HD=3

- اگر کدهای متوالی که بدنبال هم ارسال می شوند دارای فاصله همینگ متفاوت باشند کمترین فاصله همینگ بعنوان فاصله همینگ همه آنها محسوب خواهد شد مثال:

<table>
<tr><td rowspan="16">HD=1</td><td rowspan="2">HD=1</td><td>00000000</td></tr>
<tr><td rowspan="2">00000001</td></tr>
<tr><td rowspan="2">HD=2</td></tr>
<tr><td>00000010</td></tr>
<tr><td>HD=1</td><td rowspan="2">00000011</td></tr>
<tr><td rowspan="2">HD=3</td></tr>
<tr><td rowspan="2">00000100</td></tr>
<tr><td rowspan="2">HD=1</td></tr>
<tr><td rowspan="2">00000101</td></tr>
<tr><td rowspan="2">HD=2</td></tr>
<tr><td rowspan="2">00000110</td></tr>
<tr><td rowspan="2">HD=1</td></tr>
<tr><td rowspan="2">00000111</td></tr>
<tr><td rowspan="2">HD=4</td></tr>
<tr><td>00001000</td></tr>
</table>

فاصله همینگ چگونه محاسبه می شود ؟

- با اعمال XOR روی دو کد متوالی و شمارش تعداد یکها در نتیجه می توان تعداد بیتهای متفاوت یا HD را محاسبه نمود .

$$X+Y= X.Y+ X\ Y$$

- اگر X و Y دو کد متوالی باشند در اینصورت

$\overline{Y}$	Y
11110011	00001100

$\overline{X}$	X
10111111	01000000

بر اساس موارد فوق

$\overline{X}.Y= 00001100$

$X.\overline{Y}= 01000000$

$X.\overline{Y} + \overline{X}.Y =01001100 \Longrightarrow$ **فاصله HD=3= تعداد یکها**

چرا فاصله همینگ همیشه یک نیست ؟

شاید تصور شود که در ارسال یک سری اطلاعات باینری فاصله همینگ همیشه یک است (HD=1) ولی واقعیت این است که در ارسال دیتا هر فرم یا قاب دیتا متشکل از بیت های دیتا به اضافه بیت های کنترلی است که به ابتدا و انتهای داده اضافه می شوند . بنابراین بین دو کد متوالی ممکن است تعدادی بیت کنترلی مشابه بوده ولی سایر بیتها متفاوت باشند بطور کلی :

اگر m بیت دیتا داشته باشیم و r بیت برای کنترل به آن اضافه کنیم قاب n بیتی خواهیم داشت که:

$$n=m+r$$

در این قاب :

معمولاً تمام 2^m حالت ممکن که برای دیتا وجود دارد معتبر است (یا معتبر فرض می شود) ولی تمام 2^n حالت ممکن که برای قاب وجود دارد معتبر نیست زیرا ممکن است بیتهای کنترلی ثابت در اثر بروز اشکال از 0 به 1 یا بالعکس تغییر کنند .

بنابراین باید فهرستی از 2^N حالت ممکن را تهیه و فاصله همینگ را در آنها بررسی نمود .

فاصله همینگ چه استفاده ای دارد ؟

از فاصله همینگ می توان برای مقاصد زیر استفاده نمود :

۱-تشخیص خطا

۲- تصحیح خطا

چگونگی تشخیص خطا با استفاده از فاصله همینگ

بطور کلی برای شناسایی d خطا به کدی با فاصله همینگ d+1 نیاز داریم تا گیرنده متوجه یک کد نا معتبر شود .

مثال ۱: اگر HD=1 باشد در این حالت d+1=1

یعنی d=0 است بعبارت دیگر خطایی تشخیص داده نمی شود فرض کنید در سیستمی با HD=1 کد 00000000 و بدنبال آن 00000010 ارسال می شود گیرنده نمی تواند تشخیص دهد که کد دوم دیتای واقعی است یا کدی است که در آن خطا وجود دارد زیرا در سیستم HD=1 کد 00000010 نیز بعنوان یک کد معتبر شناخته می شود .

مثال ۲ : اگر HD=2 باشد در این حالت d+1=2 یعنی یک خطا قابل تشخیص است فرض کنید در سیستمی کد 00000000 و بدنبال آن 00000110 ارسال گردد یعنی فقط دو بیت در دو کد متوالی متفاوت است در این حالت اگر در اثر خطا کد دوم به 00000100 یا به 00000010 تبدیل شود توسط گیرنده قابل تشخیص است ولی اگر در اثر دو خطای همزمان کد دوم به 00000101 تبدیل شود گیرنده تصور می کند که شرایط عادی است و نمی تواند خطا را تشخیص بدهد چون HD=2 هنوز وجود دارد .

تذکر :

معمولاً برای تشخیص خطای تک بیتی از بیت توازن (Parity) استفاده می شود بعنوان مثال در even parity یا توازن زوج اگر تعداد یک های موجود در کد با احتساب بیت توازن زوج باشد بیت توازن ۱ است مثال

$$10110101 \rightarrow 1011010111$$

در غیر این صورت بیت توازن صفر است .

در استفاده از بیت توازن ، فاصله دو کد متوالی HD=2 خواهد بود زیرا اگر در مثال فوق یکی از یک ها صفر شود بیت توازن نیز صفر خواهد شد و دو بیت در کدهای متوالی متفاوت خواهد بود .

بهرحال از بیت توازن برای شناسایی خطای تک بیتی استفاده می گردد.

چگونگی تصحیح خطا با استفاده از فاصله همینگ

بطور کلی برای تصحیح d خطا به کدی با فاصله همینگ 2d+1 نیاز داریم زیرا در اینصورت فاصله بین دو کد متوالی آنقدر بزرگ است که حتی با تغییر d خطا باز میتوان کد صحیح را مشخص نمود.

مثال : کدهای زیر را در نظر بگیرید که بدنبال هم ارسال میشوند :

0000000000
0000011111
1111100000
1111111111

همانطور که ملاحظه می گردد فاصله همینگ بین آنها 5 است .پس بر اساس رابطه2d+1=5 میتوان حداکثر ۲ خطا را اصلاح نمود. فرضاً اگر با بروز خطا کد 0000000111 برسد گیرنده می فهمد که کد اصلی بایستی 0000011111 باشد و آنرا اصلاح می کند .

در مثال فوق اگر ۳ خطا رخ دهد گیرنده نمیتواند کد را بطور صحیح اصلاح کند. مثلا اگر کد 0000000000 در اثر بروز ۳ خطا به کد 0000000111 تبدیل شود گیرنده متوجه وقوع سه خطا نمیشود بلکه شبیه حالت فوق با تصور اینکه ۲ خطا اتفاق افتاده ،آنرا به 0000011111 تصحیح میکند که درست نیست.

با توجه به امکان اشتباه فوق معمولا ترجیح داده میشود تا در صورت تشخیص خطا بجای تصحیح توسط گیرنده ، صرفاً به فرستنده اطلاع داده شود تا مجدداً اقدام به ارسال کد نماید.

ضمیمه ۲

مقایسه مشخصات برخی از شبکه های فیلدباس

BACKGROUND INFORMATION				
Network Specification	Technology Developer	Year Introduced	Governing Standard	Openness
PROFIBUS DP/PA	PNO/PTO	DP-1994, PA-1995	EN 50170 DIN 19245 part 3 (DP) part 4 (PA), IEC 1158-2 (PA)	ASICs from Siemens and Profichip, Products from over 300 vendors
INTERBUS-S	Phoenix Contact	1984	DIN 19258 EN 50.254	Products from over 400 manufacturers
DeviceNet	Rockwell Allen-Bradley	March 1994	ISO 11898 ISO 11519	17 chip vendors, 300+ products, Open specification
ARCNET	Datapoint/ SMC	1977	ANSI/ATA 878.1	Chips, boards, ANSI docs
AS-I	AS-I Consortium	Fall 1993	Submitted to IEC	AS-II.C. Market item
Foundation FieldBus H1	FieldBus Foundation	1995	ISA SP50 IEC TC65	Chips/software/products from multiple vendors
Foundation FieldBus High Speed Ethernet (HSE)	FieldBus Foundation	In development, Preliminary	IEEE 802.3u RFC for IP, TCP and UDP	Chips/software/products from multiple vendors that supply Ethernet products
IEC/ISA SP50 Fieldbus	ISA & FieldBus Foundation	1992 - 1996	IEC 1158 ANSI 850	Multiple chip vendors
Seriplex	APC, Inc. now AEG Modicon	1990	Seriplex spec	Chips available multiple interfaces
WorldFIP	WorldFIP	1988	IEC 1158-2	Multiple chip vendors
LonWorks	Echolon Corp.	March 1991		Public documentation on protocol

BACKGROUND INFORMATION				
Network Specification	Technology Developer	Year Introduced	Governing Standard	Openness
SDS	Honeywell	Jan., 1994	Honeywell Specification, Submitted to IEC, ISO11989	17 chip vendors, 200+ products
ControlNet	Rockwell Allen-Bradley	1996	ControlNet International	Open Specification, 2 chip vendors
CAN Open	CAN in Automation	1995	CiA	17 chip vendors, 300 product vendors, Open specification
Modbus Plus	AEG Modicon	1980's	None	Controlled by AEG Modicon many vendor support through modconnect program
Modbus RTU/ASCII	AEG Modicon	1970's	EN 1434-3 (layer 7) IEC 870-5 (layer 2)	Open specification, uses UART (RS232,422/485), no special hardware required
Industrial Ethernet	Intel/DEC/Xerox	Late 1970's	IEEE802.2	The most open network worldwide thousands of vendors, hundreds of different chip suppliers.

PHYSICAL CHARACTERISTICS				
Network	Network Topology	Physical Media	Max. Devices (nodes)	Max. Distance
PROFIBUS DP/PA	Line, star & ring	Twisted-pair or fiber	127 nodes (124 slaves, 4 segments, 3 repeaters) up to 3 masters	100m between segments @ 12Mbaud; 24 Km (fiber) baud rate and media dependent
INTERBUS-S	Segmented with "T" drops	Twisted-pair, fiber, and slip-ring	256 nodes	400 m/segment, 12.8 Km total
DeviceNet	Trunkline/dropline with branching	Twisted-pair for signal & power	64 nodes	500m (baudrate dependent) 6Km w/repeaters
ARCNET	Star, Bus, distributed star	Twisted-pair, coax, fiber	255 nodes	Coax 2000 feet; Twisted pair 400 feet; Fiber 6000 Feet
AS-I	Bus, ring, tree star, of al	Two wire cable	31 slaves	100 meters, 300 with repeater
FieldBus Foundation H1	Star or bus	Twisted-pair, fiber	240/segment, 65,000 segments	1900m @ 31.25K
FieldBus Foundation HSE	Star	Twisted-pair, fiber	IP addressing - unlimited nodes	100m @ 100Mbaud 2000m @ 100Mbaud fiber full duplex
IEC/ISA SP50 Fieldbus	Star or bus	Twisted-pair fiber, and radio	IS 3-7 non IS 128	1700m @ 31.25K 500M @ 5Mbps
Seriplex	Tree, loop, ring, multi-drop, star	4-wire shielded cable	500+ devices	500+ ft
WorldFIP	Bus	Twisted-pair, fiber	256 nodes	up to 40 Km
LonWorks	Bus, ring, loop, star	Twisted-pair, fiber, power line	32,000/domain	2000m @ 78 kbps
SDS	Trunkline/Dropline	Twisted-pair for	64 nodes,	500m (baudrate dependent)

PHYSICAL CHARACTERISTICS				
Network	Network Topology	Physical Media	Max. Devices (nodes)	Max. Distance
		signal & power	126 addresses	
ControlNet	Linear, Tree, Star, or Combination	Coax, fiber	99 nodes	1000m (coax) 2 nodes 250m with 48 nodes 3km fiber, 30km fiber w/repeaters
CANOpen	Trunkline/Dropline	Twisted Pair + optional signal & power	127 nodes	25-1000m (baud rate dependent)
Modbus Plus	linear	Twisted Pair	32, 64 max. per segment with bridge capabilities	500m per segment
Modbus RTU/ASCII	Line, star, tree, network w/segments	Twisted Pair	250 nodes per segment	350m
Industrial Ethernet	STAR, BUS	10BASE-T, 10-Base-FL (FIBER) 100 Base TX	48 bit address	

TRANSPORT MECHANISM						
Network	Communication Methods	Transmission Properties	Data Transfer Size	Arbitration Method	Error Checking	Diagnostics
PROFIBUS DP/PA	Master/slave peer to peer	DP: 9.6, 19.2, 93.75, 187.5, 500 Kbps, 1.5, 3, 6, 12 Mbps PA 31.25 kbps	0-244 bytes	Token passing	HD4 CRC	Station, module & channel diagnostics
INTERBUS-S	Master/slave with total frame transfer	500kBits/s, full duplex	1-64 bytes data, 246 bytes parameter, 512 bytes h.s., unlimited block	None	16-bit CRC	Segment location of CRC error and cable break
DeviceNet	Master/slave, multi-master, others	500 kbps, 250 kbps, 125 kbps	8-byte variable message	Carrier-Sonac Multiple Access w/ non-destructive bitwise arbitration	CRC check	Bus monitoring
ARCNET	Peer to peer	19.53K to 10M	0-507 bytes	Token passing	16-bit CRC	Built in Acknowledgements at Data link layer
AS-I	Master/slave with cyclic polling	Data and power, EMI resistant	31 slaves with 4 in and 4 out	Master/slave with cyclic polling	Manchester Code, hamming-2	Slave fault, device fault
Fieldbus Foundation H1	Client/server publisher/ subscriber, Event notification	31.25 kbps	128 octets	Scheduler, multiple backup	16-bit CRC	Remote diagnostics, network monitors, parameter status
Fieldbus Foundation	Client/server publisher/	100Mbps	Varies, Uses Standard TCP/IP	CSMA/CD	CRC	

TRANSPORT MECHANISM						
Network	Communication Methods	Transmission Properties	Data Transfer Size	Arbitration Method	Error Checking	Diagnostics
HSE	subscriber, Event notification					
IEC/ISA SP50 Fieldbus	Client/server Publisher/ subscriber	31.25 kbps IS+1, 2.6, 5 Mbps	64 octets high & 256 low priority	Scheduler, tokens, or master	16-bit CRC	Configurable on network management
Seriplex	Master/slave peer to peer	200 Mbps	7680/transfer	Sonal multiplexing	End of frame & echo check	Cabling problems
WorldFIP	Peer to peer	31.25 kbps, 1 & 2.5 Mbps, 6 Mbps fiber	No limit, variables 128 bytes	Central arbitration	16-bit CRC, data "freshness" indicator	Device message time-out, redundant cabling
LonWorks	Master/slave peer to peer	1.25 Mbs full duplex	228 bytes	Carrier Sense, Multiple Access	16-bit CRC	Database of CRC errors and device errors
SDS	Master/slave, peer to peer, multi-cast, multi-master	1Mbps, 500 kbps, 250 kbps, 125 kbps	8-byte variable message	Carrier-Sonac Multiple Access	CRC check	Bus monitoring, Diagnostic slave
ControlNet	Producer / Consumer Device object model	5 Mbps	0-510 bytes variable	CTDMA time slice multiple access	Modified CCITT with 16 bit polynomial	Duplicate Node ID, Device, Slave Faults
CANOpen	Master/slave,	10K, 20K,	8-byte variable	Carrier-Sonac Multiple	CRC check	Error Control &

TRANSPORT MECHANISM						
Network	Communication Methods	Transmission Properties	Data Transfer Size	Arbitration Method	Error Checking	Diagnostics
	multi-master, others	50K, 125K, 250K, 500K, 800K, 1Mbps	message	Access w/ non-destructive bitwise arbitration		Emergency Messages
Modbus Plus	Token Passing	1 MBPS	256 bytes data + header	peer to peer, token passing	16-bit CRC	Local Chip and Software
Modbus RTU/ASCII	Master/Slave	300 bps - 38.4K bps	0-254 bytes			
Industrial Ethernet	CSMA/CD	10MBPs, 100 MBPs	1500 Bytes Data.	Collision Detection	32 bit CRC.	CD, Network Management

PERFORMANCE			
	Cycle Time: 256 Discrete 16 nodes with 16 I/Os	Cycle Time: 128 Analog 16 nodes with 8 I/Os	Block transfer of 128 bytes 1 node
PROFIBUS DP/PA	Configuration dependent typical <2ms	Configuration dependent typical <2ms	not available
INTERBUS-S	1.8 ms	7.4 ms	140 ms
DeviceNet	2.0 ms Master-slave polling	10 ms Master-slave polling	4.2 ms
ARCNET	Application layer dependent	Application layer dependent	Application layer dependent
AS-I	4.7 ms	not possible	not possible
Fieldbus Foundation H1	<100 ms @ 31.25k	<600 ms @ 31.25k	36 ms @ 31.25k
Fieldbus Foundation HSE	Not Applicable; Latency <5ms	Not Applicable; Latency <5ms	<1ms
IEC/ISA SP50	Configuration dependent	Configuration dependent	0.2 ms @ 5 Mbps 1.0 ms @ 1 Mbps
Seriplex	1.32 ms @ 200 kbps, m/s	10.4 ms	10.4 ms
WorldFIP	2 ms @ 1 Mbps	5 ms @ 1 Mbps	5 ms @ 1 Mbps
LonWorks	20 ms	5 ms @ 1 Mbps	5 ms @ 1 Mbps
SDS	<1 ms, event driven	<1 ms per event	2 ms @ 1 Mbps
ControlNet	<0.5ms	<0.5ms	<0.5ms
CANOpen	<1ms	<1ms	<1ms
Modbus Plus			
Modbus RTU/ASCII			
Industrial Ethernet	Not Applicable; Latency <5ms	Not Applicable; Latency <5ms	<1ms

TABLE 1

Type of network	Ethernet	Modbus RTU/ASCII	Profibus	Foundation Fieldbus	DeviceNet	CANopen	J1939
Origin	Digital Equipment Corp., Intel, and Xerox - 1976	Modicon - 1978	German govt. and automation manufacturers - 1989	ISA - 1998	Allen-Bradley - 1994	CAN in Automation - 1993	SAE 1994
Implementation	Produced on chips by many vendors; based on IEEE 802.3	Produced on any medium, but it is typically found on RS-232, -422, or -485; no special ASICs required	Produced on ASICs by multiple vendors; based on RS-485 and the European EN50170	Produced on chips by multiple vendors	Produced on chips by many vendors; based on CAN	Produced on chips by many vendors; based on CAN	Produced on chips by many vendors; based on CAN
Formats	10Base-2, 10Base-T, 100Base-T, 100Base-FX, 1 Gb; copper (twisted pair/thin coaxial), and fiber	Typically RS-232, RS-422, RS-485	Profibus DP (master/slave), Profibus FMS (multimaster/ peer to peer), and Profibus PA (intrinsically safe)	H1 intrinsically safe and High-Speed Ethernet (HSE); based on ISA SP50/ IEC61158			
Connectors	RJ-45 or coaxial	Typically DB9 or terminal block	9-pin D-shell connector (impedance terminated) or 12 mm IP 67 quick disconnect	Application dependent	Mini 18 mm and micro 12 mm waterproof quick disconnect plugs and receptacles; 5-pin Phoenix terminal block	Mini 18 mm and micro 12 mm waterproof quick disconnect plugs and receptacles; 9-pin D-shell	Application dependent
Max. nodes	1024, expandable with routers	250	127	240/ segment; 65,000 possible segments	64	64	30/ segment
Distance	100 m (10Base-T) to 50 km (mono mode, fiber with switches)	350 m for RS-485	100 m (copper, no repeaters, max. speed) to 24 km (with repeaters and fiber optic transmission)	1900 m for H1	100-500 m	100-500 m	40 m
Bit rate	10 Mbps to 1	Can run at any	9600 bps to 12 Mbps	H1 31.25 Kbps	125, 250, and 500	125, 250, and 500	250 Kbps

TABLE 1

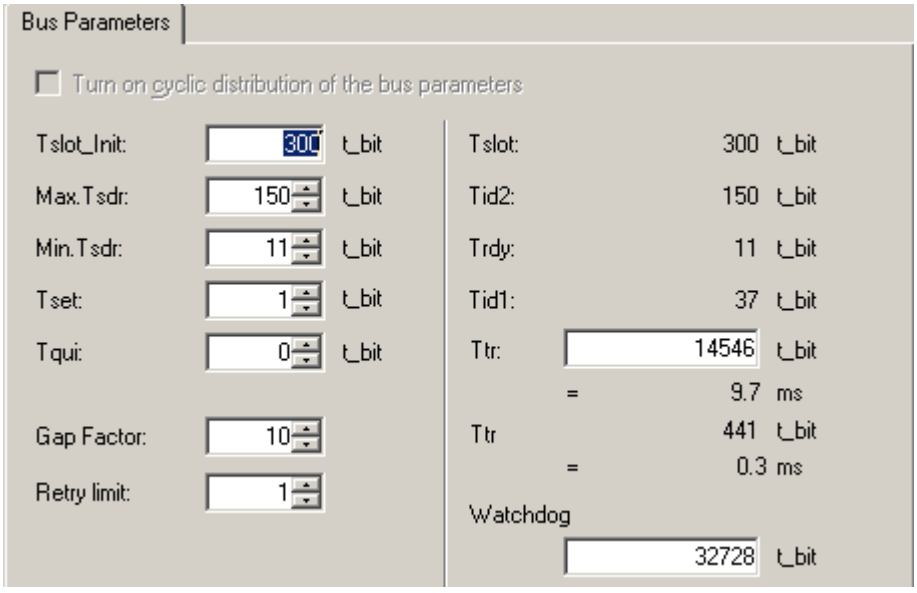
Type of network	Ethernet	Modbus RTU/ASCII	Profibus	Foundation Fieldbus	DeviceNet	CANopen	J1939
	Gbps	speed, but it is most commonly used between 9600 and 38,400 bps		and HSE 100 Mbps	Kbps	Kbps	
Message size	46-1500 bytes	0-254 bytes	Max. 244 bytes/ node / message	128 octets	8 bytes/ node/ message	8 bytes/ node/ message	4-8 bytes/ node/ message
Messaging format	Peer to peer	Master/ slave; discrete and analog I/O and parameters	Polling (DP/PA) and peer to peer (FMS)	Client/ server, publisher/ subscriber, and event notification	Polling, strobing, change-of-state, cyclic; explicit messaging for configur-ation and parameter data; UCMM for peer to peer messaging; producer-consumer-based model	Polling, strobing, change-of-state, cyclic, and others	Broadcast, one-to-one
Supporting trade organi-zation	Industrial Ethernet Assoc. and Industrial Automation Open Networking Assoc.	Modicon/ Groupe Schneider	Profibus Trade Org.	Fieldbus Foundation	Open DeviceNet Vendor Assoc.	CAN In Automation	Society of Automotive Engineers

ضمیمه ۳

پارامترهای شبکه PROFIBUS

یک شبکه PROFIBUS تنها زمانی درست کار می کند که پارامترهای باس به صورت مناسب تعیین شده باشند.همچنین پارامترهای باس که برای یک Node شبکه تعریف می شوند باید برای همه Node های شبکه یکسان باشد.

در حالت کلی پارامتر های باس به نرخ ارسال اطلاعات بستگی دارند.تذکر اینکه نحوه تعیین این پارامترها به صورت تجربی است واکیداً توصیه می شود که تغییر مقادیر این پارامترها توسط افراد با تجربه صورت گیرد.برای دیدن پارامترهای باس همانطور که در فصل چهارم توضیح داده شده است باید در پنجره Network Setting مربوط به شبکه PROFIBUS کلید Bus Parameters رافشار دهید پنجره ای مانند شکل زیر ظاهر خواهد شد:



(Time Target Rotation) Ttr

مدت زمانی است که Token ، حلقه را دور می زند ودر اختیار همه ایستگاههای Master قرار گیرد.در واقع در طول این زمان، همه ایستگاههای Master ، یک بار Token را در اختیار می گیرند و در نتیجه می توانند باس را در اختیار گرفته وتبادل دیتا انجام دهند. محدوده Ttr بصورت 256 t_bit <= Ttr <= 16.777.960 t_bit است:

اگر کاربر بخواهد دستی این پارامتر را تنظیم کند باید حداقل آنرا ۵۰۰۰ برابر آدرس HSA یعنی بالاترین آدرس باس بگیرد.آدرس HAS در پنجره قبل از پنجره فوق ظاهر میگردد.

GAP Factor

با اضافه کردن، ایستگاههای Master جدید. ممکن است یک حلقه پاسخگوی ارتباط مورد نظر نباشد ونیاز باشد که چندین حلقه Token تشکیل شود. GAP Factor در واقع تعداد حلقه های Token را پس از اضافه کردن ایستگاههای جدید،نمایش می دهد و میتواند عددی بین ۱ تا ۱۰۰ باشد.

Retry Limit

پس از اینکه یک پیغام به طور صحیح ارسال شود،گیرنده یک پیغام میدهد واصطلاحاً ایستگاه گیرنده Acknowledge می دهد.اگر چنانچه به هر دلیل ارسال پیام دچار مشکل شود وAcknowledge نشود ویا زمان Time Out سپری شود،پیغام مجدداً ارسال می شود.پارامتر Retry Limit در واقع تعیین کننده ، حداکثر تعداد تکرار ارسال پیام است و میتواند عددی بین ۱ تا ۱۵ باشد.

(Minimum Time Station Delay Response) Min-TSDR

حداقل زمانی است که یک ایستگاه پس از دریافت پیغام باید منتظر بماند وپس ازآن به آن پاسخ دهد.

1 t_bit <= Min. Tsdr <= MIN(255 t_bit, ... Max. Tsdr - 1, 34 + 2*Tset + Tqui)

Max. TSDR

حداکثر زمانی است که یک ایستگاه می تواند به یک پیغام ،پاسخ دهد.

۳۵ + ۲*Tset + Tqui <= Max. Tsdr <= 1.023 t_bit

(Slot Time) Tslot

حداکثر زمانی است که یک فرستنده می تواند منتظر پاسخ گیرنده باشد.

(Setup Time)Test

مدت زمانی که بین دریافت یک پیغام وپاسخ به ان طول می کشد.

1 t_bit <= Tset <= 494 t_bit

(Quiet Time) Tqui

مدت زمانی که طول می کشد تا یک ایستگاه از حالت فرستنده به گیرنده تغییر وضعیت دهد.

0 t_bit <= Tqui <= MIN(31 t_bit, Min. Tsdr - 1)

(Idle Time 1) Tid1

حداقل زمانی که طول می کشد تا یک ایستگاه فرستنده، پس از دریافت پاسخ ،مجدداً پیغام جدید ارسال کند.

Tid1 = 35 + 2*Tset + Tqui

(Idle Time 2) Tid2

اگر پس از ارسال یک پیام،Acknowledge شد وپاسخی دریافت نشد ایستگاه فرستنده یک مدت زمان باید صبر کند سپس پیغام بعدی را ارسال کند،به این زمان Tid2 گفته می شود.

Tid2 = Max. Tsdr

(Ready Time)Trdy

حداقل زمانی که باید بگذرد تا بعد از ارسال یک پیام ،فرستنده بتواند پاسخ را دریافت کند Trdy = Min. Tsdr

در تعاریف فوق منظور از دریافت پاسخ ،Acknowledge شدن پیغام ارسالی است.

پارامترهای فوق دقیقاً باید از سوی تمامی Node ها، رعایت شوند .پارامترهای باس همگی برحسب tBIT (Time-Blt) معین می شوند. یک tBIT زمان چرخش یک بیت در شبکه است وبه صورت زیر محاسبه می شود:

TBIT=1/نرخ ارسال اطلاعات (Bit/s)

یا

TBIT (µsec) = 1/Mbps

به عنوان مثال اگر نرخ ارسال 12Mbps باشد،برابر83ns واگر نرخ ارسال 1/5Mbps باشد ، tBIT برابر667ns می باشد.

تذکر:

در قسمت بالای پنجره Bus Parameters یک گزینه به نام

Turn on cyclic distribution of the bus parameters

وجود دارد که اگر فعال شود مجموعه تنظیمات انجام شده برای Subnet مورد نظر به صورت Cyclic توسط تمام ارتباطات DP فعال در شبکه PROFIBUS منتقل می شود.زمانی این گزینه را فعال کنید که مرتباً می خواهید یک واحد PG به شبکه PROFIBUS که در حال کار است وصل کنید .

ضمیمه ٤

مقایسه فرمت دیتا در S7 و FMS

Convert Data Type >		Bit Length	Number of Bytes	Range of Values	
S7 Type	FMS PDU	S7	in FMS PDU	S7	FMS
BOOL	Boolean	1	1	0,1	0x00, 0xff
BYTE	Unsigned8 [2]	8	1	every 8-bit string	
WORD	Unsigned16 [2]	16	2	every 16-bit string	
DWORD	Unsigned32 [2]	32	4	every 32-bit string	
CHAR	Octet string[1]	8	1	see ISO 646 and ISO 2375: Defining registration number 2 + SPACE	
ARRAY [x..x+n] OF CHAR	Octet string[n+1] 0<=n<=236	[n+1]*8	n+1	see ISO 646 and ISO 2375: Defining registration number 2 + SPACE	
INT	Integer16	16	2	$-2^{15}..2^{15}-1$	
DINT	Integer32	32	4	$-2^{31}..2^{31}-1$	
REAL	Floating-point	32	4	see IEEE Stand. 754	
TIME	Time difference	32	4	see IEC 1131 IS	$0..2^{32}-1$ ms and
DATE	Octet string[2]	16	2	see IEC 1131 IS	see EN 50132
TIME_OF_ DAY or TOD	Time-of-day	32	4 or 6	see IEC 1131 IS	$0..2^{28}-1$ms
S5TIME	Octet string[2]	16	2	See IEC 1131 IS	
DATE_AND _TIME or DT	Date	64	7	see IEC 1131 IS	$0..2^{28}-1$ ms or$0..2^{16}-1$ days
STRING[n] (where 0<n<=237)	Visible string[n]	8n	n	see IEC 1131 IS	
Timer	Octet string[2]	16	2	0 to 65535	
Counter	Octet string[2]	16	2	0 to 65535	

ضمیمه ۵

کد های خطا در ارتباط FMS

Error-Free Job Execution

Meaning	STATUS	ERROR	DONE/NDR
Job completed without errors	0x0000	0	1
Job active	0x000B	0	0

Error Classes

Meaning	Error Class
Indicates errors or problems involving the following: S FB parameter assignment S Block execution in the CPU and CP	Block
Indicates errors or problems on the interface between the user program and FB.	Application
Indicates errors that usually involve inconsistencies between the user program and FMS configuration	Definition
Indicates resource problems on the PROFIBUS CP.	Resources
Indicates errors or problems in conjunction with the requested FMS service.	Service
Indicates denied access to objects due to the following: S Absence of access rights S Hardware problems S Other inconsistencies	Access
Indicates problems accessing the object dictionary of the VFD.	OD (object dictionary)
Unspecified error on the VFD	VFD status
Other errors	Other

Locally Detected Errors

Meaning	STATUS	ERROR	DONE/NDR
Communications problem: For example K bus connection not established.	0x0001	1	0
Function cannot be executed: either negative acknowledgment by the CP or error in the sequence, for example K bus protocol error.	0x0002	1	0
The connection is not configured (invalid ID specified). If the connection is configured, the error message indicates that the permitted parallel job processing limit has been exceeded. Example SAC=0 is configured and a REPORT job is sent.	0x0003	1	0

The receive data area is too short or the data types do not match.	0x0004	1	0
A reset request has been received from the CP (BRCV).	0x0005	1	0
The corresponding job execution on the CP is in the DISABLED state or a reset request has been received from the CP; the transfer is therefore incomplete.	0x0006	1	0
Corresponding job execution on the CP is in the wrong state. With REPORT: the error is specified in greater detail in the diagnostic buffer.	0x0007	1	0
Job execution on the CP signals an error accessing the user memory.	0x0008	1	0
Communications problem: For example K bus connection not established.	0x0001	1	0
Function cannot be executed: either negative acknowledgment by the CP or error in the sequence, for example K bus protocol error.	0x0002	1	0
The connection is not configured (invalid ID specified). If the connection is configured, the error message indicates that the permitted parallel job processing limit has been exceeded. Example SAC=0 is configured and a REPORT job is sent.	0x0003	1	0
The receive data area is too short or the data types do not match.	0x0004	1	0
A reset request has been received from the CP (BRCV).	0x0005	1	0
The corresponding job execution on the CP is in the DISABLED state or a reset request has been received from the CP; the transfer is therefore incomplete.	0x0006	1	0
Corresponding job execution on the CP is in the wrong state. With REPORT: the error is specified in greater detail in the diagnostic buffer.	0x0007	1	0
Job execution on the CP signals an error accessing the user memory.	0x0008	1	0

Application Error Class

Meaning	STATUS	ERROR	DONE/NDR
Unspecified application reference error.	0x0200	1	0
The configured connection cannot be established at present; for example LAN connection not established.	0x0201	1	0

Definition Error Class

Meaning	STATUS	ERROR	DONE/NDR
Unspecified definition error.	0x0300	1	0
Object with requested index/name is not defined.	0x0301	1	0
Object attributes are inconsistent.	0x0302	1	0
Name exists already.	0x0303	1	0

Resources Error Class

Meaning	STATUS	ERROR	DONE/NDR
Unspecified resource error.	0x0400	1	0
No memory available.	0x0401	1	0

Resources Error Class

Meaning	STATUS	ERROR	DONE/NDR
Unspecified resource error.	0x0400	1	0
No memory available.	0x0401	1	0

Service Error Class

Meaning	STATUS	ERROR	DONE/NDR
Unspecified service error.	0x0500	1	0
Conflict due to object status.	0x0501	1	0
Configured PDU size exceeded.	0x0502	1	0
Conflict due to object restrictions.	0x0503	1	0
Inconsistent parameters.	0x0504	1	0
Illegal parameters.	0x0505	1	0

Access Error Class

Meaning	STATUS	ERROR	DONE/NDR
Unspecified access error.	0x0600	1	0
Invalid object.	0x0601	1	0
Hardware error.	0x0602	1	0
Object access was denied.	0x0603	1	0
Invalid address.	0x0604	1	0
Inconsistent object attributes.	0x0605	1	0
Object access not supported.	0x0606	1	0
Object does not exist in OD or GetOD still active	0x0607	1	0
Type conflict or variable content outside permitted range of values	0x0608	1	0
Access using names not supported.	0x0609	1	0

Object Dictionary (OD) Error Class

Meaning	STATUS	ERROR	DONE/NDR
Unspecified OD error.	0x0700	1	0
Permitted name length exceeded.	0x0701	1	0
Overflow of the object dictionary.	0x0702	1	0
Object dictionary is write protected.	0x0703	1	0
Overflow of the extension length.	0x0704	1	0
Overflow of the object description length.	0x0705	1	0
Processing problem.	0x0706	1	0

VFD–Status/Reject Error Class, continued

Meaning	STATUS	ERROR	DONE/NDR
Unspecified VFD status error.	0x0100	1	0
RCC/SAC/RAC error	0x0108	1	0
Service not supported.	0x0106	1	0
PDU length error	0x0105	1	0
Bad FMS-PDU	0x0102	1	0

16 Application Error Class

Meaning	STATUS	ERROR	DONE/NDR
Unspecified error.	0x0800	1	0
Unspecified application reference error.	0x8200	1	0
Application (for example user program) not obtainable.	0x8201	1	0

Definition Error Class

Meaning	STATUS	ERROR	DONE/NDR
Unspecified definition error.	0x8300	1	0
Object with requested index/name is not defined.	0x8301	1	0
Object attributes are inconsistent.	0x8302	1	0
Name exists already.	0x8303	1	0

Resources Error Class

Meaning	STATUS	ERROR	DONE/NDR
Unspecified resource error.	0x8400	1	0
No memory available.	0x8401	1	0

Service Error Class

Meaning	STATUS	ERROR	DONE/NDR
Unspecified service error.	0x8500	1	0
Conflict due to object status.	0x8501	1	0
Configured PDU size exceeded.	0x8502	1	0
Conflict due to object restrictions.	0x8503	1	0
Inconsistent parameters.	0x8504	1	0
Illegal parameters.	0x8505	1	0

Access Error Class

Meaning	STATUS	ERROR	DONE/NDR
Unspecified access error.	0x8600	1	0
Invalid object.	0x8601	1	0
Hardware error.	0x8602	1	0
Object access was denied.	0x8603	1	0

Access Error Class, continued

Meaning	STATUS	ERROR	DONE/NDR
Invalid address.	0x8604	1	0
Inconsistent object attributes.	0x8605	1	0
Object access is not supported.	0x8606	1	0
Object does not exist.	0x8607	1	0
Type conflict or variable content outside permitted range of values	0x8608	1	0
Access using names is not supported.	0x8609	1	0

OD (Object Dictionary) Error Class

Meaning	STATUS	ERROR	DONE/NDR
Unspecified OD error.	0x8700	1	0
Permitted name length exceeded.	0x8701	1	0
Overflow of the object dictionary.	0x8702	1	0
Object dictionary is write protected.	0x8703	1	0
Overflow of the extension length.	0x8704	1	0
Overflow of the object description length.	0x8705	1	0
Processing problem.	0x8706	1	0

VFD Status Error Class

Meaning	STATUS	ERROR	DONE/NDR
Unspecified VFD status error.	0x8100	1	0

Other Error Class

Meaning	STATUS	ERROR	DONE/NDR
Unspecified error detected by partner.	0x8000	1	0

کلمات اختصاری

ASI	Actuator Sensor Interface
BPS	Bits Per Second
BRCT	Broadcast
CP	Communication Processor
CRC	Cyclic Redundancy Check
CSMA	Carrier Sense Multiple Access
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
CSMA/CD	Carrier Sense Multiple Access with Collision Detection
DAE	Destination with Address Extension
DDLM	Direct Data Link Mapper
DP	Decentralized Peripheral (Distributed Peripheral)
DPM1	DP Master Class1
DPM2	DP Master Class2
DSAP	Destination SAP
EDD	Electronic Device Description
EIA	Electronic Industries Association
ERP	Enterprise Resource Planing
ET	Electronic Terminal
FCS	Frame Check Sequence
FDL	Fieldbus Data Link
FDT	Field Device Tool
FDX	Full Duplex
FISCO	Fieldbus Intrinsically Safe Concept
FMA	Fieldbus Management
FMS	Field Message Specification
Gbps	Gigabits Per Second
GSD	General Station Data
HAS	High Station Address
HDLC	High Level Data Link Control
HDX	Half Duplex

IE	Industrial Ethernet
IEC	International electro-technical Commission
IEEE	Institute Of Electrical And Electronics Engineers
ILM	Infrared Linking Module
IP	Internet Protocol
ISDL	Integrated Services Digital Network
ISO	International Organization for Standardization
Kbps	Kilobits Per Second
LAN	Local Area Network
LSAP	Link Service Access Point
MAC	Media Access Control
MAN	Metropolitan Area Network
MBP	Manchester coded Bus Powered
Mbps	Megabits Per Second
MMAC	Master Master Acyclic Connection
MPI	Multipoint Interface
MSAC	Master Slave Acyclic Connection
MSAC-SI	Master Slave Acyclic Connection with Slave Initiative
MSCY	Master Slave Cyclic Connection without Slave Initiative
OBT	Optical Bus Terminal
OLM	Optical Linking Module
OLP	Optical Link Plug
OSI	Open System Interconnection
OTDR	Optical Time Domain Reflectometer
PA	Process Automation
PDU	Protocol Data Unit
PI	Profibus International
PNO	Profibus Neutzer Organization
Profibus	Process FieldBus
PTP	Point To Point

SAE	Source with Address Extension
SAP	Service Access Point
SDA	Send Data with Acknowledge
SDLC	Synchronous Data Link Control
SDN	Send Data with No Acknowledge
SRD	Send and Request Data
SSAP	Source SAP
STP	Shielded Twisted Pair
TCP/IP	Transmission Control Protocol/Internet Protocol
TSAP	Transport Service Access Point
UART	Universal Asynchronous Receiver/Transmitter
UTP	Unshielded Twisted Pair
VFD	Virtual Field Device
VPN	Virtual Private Network
WAN	Wide Area Network
WLAN	Wireless Local Area Network

منابع و مراجع

1.	***Decentralization with Profibus DP/DPV1***	*By:*	*Josef Weigman& Gerhard Kilian*
2.	***Profibus Specification***		*Profibus Neutzer Organization*
3.	***Profibus Technology and Application***		*Profibus International*
4.	***Profibus for Process Automation***		*Profibus International*
5.	***Communicatin Networks***		*Samson*
6.	***Serial Data Transmission***		*Samson*
7.	***Profibus PA***		*Samson*
8.	***Industrial Networks for Communication and Control***		*S. Djiev*
9.	***Fieldbus Overview***		*Brad Harrison*
10.	***Network Fundamentals***		*Chuong Huynh*
11.	***Industrial Communication***		*Gustaf Olsson*
12.	***Standard Fieldbuses***		*H. Kirrmann*
13.	***Field Bus: principles***		*H. Kirrmann*
14.	***Field Bus Operation***		*H. Kirrmann*
15.	***OSI model***		*H. Kirrmann*
16.	***Comparison Of Substation Network Topologies***		*ABB*
17.	***Profibus Network2000***		*Siemens*
18.	***NCM for Profibus Primer***		*Siemens*
19.	***NCM for Profibus Vol 1 & Vol 2***		*Siemens*
20.	***Configuring Hardware with Step7***		*Siemens*
21.	***Communication with Simatic***		*Siemens*
22.	***Profibus Diagnostic & Troubleshooting***		*Siemens*
23.	***Profibus Workshop***		*Siemens*
24.	***Network Solution for Profibus***		*Siemens*
25.	**شبکه های کامپیوتری**		*Andrew S. Tanenbaum* *ترجمه : عین اله جعفر نژاد*