

## برقراری ارتباط مابین دو عدد PLC S7-1200 تحت شبکه PROFINET با دستورات PUT/GET

نویسنده : مهندس مهدی توانا  
مدرس دوره های آموزشی اتوماسیون صنعتی  
عضو هیئت مدیره شرکت مهندسی نادین طیف نگار

واژه های کلیدی : شبکه سازی صنعتی ، پروفینت ، PLC S7-1200

**چکیده :** گسترش روز افزون ارتباطات در سیستم های کنترل صنعتی که منجر به کاهش هزینه های اولیه راه اندازی و همچنین کاهش مشکلات مرتبط با کابل کشی می گردد ، سازندگان مختلف سیستم های کنترل را به سمت توسعه و ساده سازی ارتباطات فیما بین تجهیزات سوق می دهد . شبکه صنعتی پروفینت [۱] که امروزه به عنوان یکی از رایج ترین پروتکل های ارتباطی با سرعت زیاد در حال توسعه می باشد ، یک پروتکل استاندارد صنعتی جهت جمع آوری و انتقال داده ها بر مبنای اتترنت صنعتی می باشد . سیستم های کنترل نوین زیمنس شامل S7-1200 و S7-1500 ، از پروفینت به عنوان پروتکل اصلی ارتباطی بهره می برند . در این مقاله آموزشی به یکی از رایج ترین روشهای ارتباطی مابین دو عدد PLC S7-1200 می پردازیم .

### الف -

PLC های خانواده S7-1200 زیمنس که از اوایل دهه ۹۰ وارد صنعت کشور شده اند ، از پورت ارتباطی پروفینت برای ارتباطات پایه ، از جمله برنامه ریزی توسط نرم افزار TIA PORTAL و همچنین برقراری ارتباط با سایر سیستم های کنترل بهره مند می باشند . این امکان در کنار سایر ویژگی های سخت افزاری ، نرم افزاری و همچنین قیمت تمام شده ، علاوه بر بالابردن انعطاف پذیری [۲] این محصول در کاربردهای صنعتی ، مزایای مهمی را بواسطه برقراری ارتباطات صنعتی تحت پروفینت محقق می نماید که مهم ترین آنها عبارتند از :

۱- پروفینت امکان تبادل داده های صنعتی را با سرعت و دقت بالا در تمامی پلتفرمهایی که از این استاندارد تبعیت می نمایند ، فراهم می سازد . با توجه به این ویژگی امکان ایجاد مفاهیم نوآوارانه برای صنایع تولیدی فراهم می گردد .

۲- لایه باز بودن و انعطاف پذیری بالای این پروتکل ، آزادی عمل گسترده ای را به طراحان سیستم و برنامه نویسان سیستم های کنترل ارائه می دهد که با این توانمندی می توانند دیدگاههای خود را با روش هایی ساده تر و امن تر به اجرا برسانند. این موضوع باعث افزایش بهره وری سیستم به دلیل کاهش زمان راه اندازی سیستم های پیچیده می گردد .

۳- بالا رفتن کیفیت و افزایش سرعت پیاده سازی سیستم های کنترل پیچیده ، در زمینه هایی از جمله کنترل دقیق حرکت مبتنی بر ارتباطات توسعه یافته پروفینت در درایو های پویا ، هماهنگ سازی عملکرد دستگاه ها بواسطه نرخ بالای انتقال داده در پروفینت و ... از مهمترین عوامل کلیدی برای دستیابی به مزیت رقابتی در تولید می باشند .

در PLC های خانواده S7-1200 زیمنس روشهای مختلفی برای برقراری ارتباط با تجهیزات دیگر در بستر ارتباطی پروفینت در نظر گرفته شده است . در این مقاله آموزشی به ایجاد ارتباط مابین دو عدد CPU S7-1200 تحت شبکه پروفینت و با بکارگیری دستورات PUT/GET می پردازیم . دستورات PUT/GET از نوع ارتباطات S7-Connection که یک استاندارد ارتباطی مخصوص سیستم های کنترل STEP7 زیمنس می باشند ، جهت دسترسی به داده ها استفاده می نمایند . این نوع ارتباطات می تواند در بسترهای پروفینت ، پروفیباس یا MPI تعریف شود . سه نوع سرویس ارتباطی که در جدول شکل یک نمایش داده شده اند ، در زیر مجموعه S7-Connection قابل تعریف است .

نام سرویس ارتباطی	توضیحات
PUT/GET	جهت خواندن / نوشتن یک طرفه [۳] اطلاعات برای انتقال حجم کم داده به / از یک ایستگاه کاری
USEND / URCV	سرویس غیر هماهنگ [۴] دو طرفه [۵] برای انتقال حجم متوسط داده بین دو ایستگاه
BSEND / BRCV	یک سرویس دو طرفه بلوک گرا [۶] برای انتقال حجم زیاد داده بین دو ایستگاه.

### شکل ۱

مهمترین مزایای S7-Connection عبارتند از :

- مستقل از بستر انتقال اطلاعات ، امکان تبادل حجم وسیعی از داده ها را فراهم می سازد .
- امکان انتقال داده ها را از بخش های مختلف حافظه فراهم می سازد.
- در هر مرحله امکان انتقال 64Kbyte داده را فراهم می کند.
- با بهره گیری از پروتکل های ویژه ، امکان تایید اتوماتیک دریافت داده را فراهم می کند.
- در هنگام انتقال حجم وسیع داده از درگیر شدن بیش از اندازه CPU و باس های انتقال داده جلوگیری می کند .

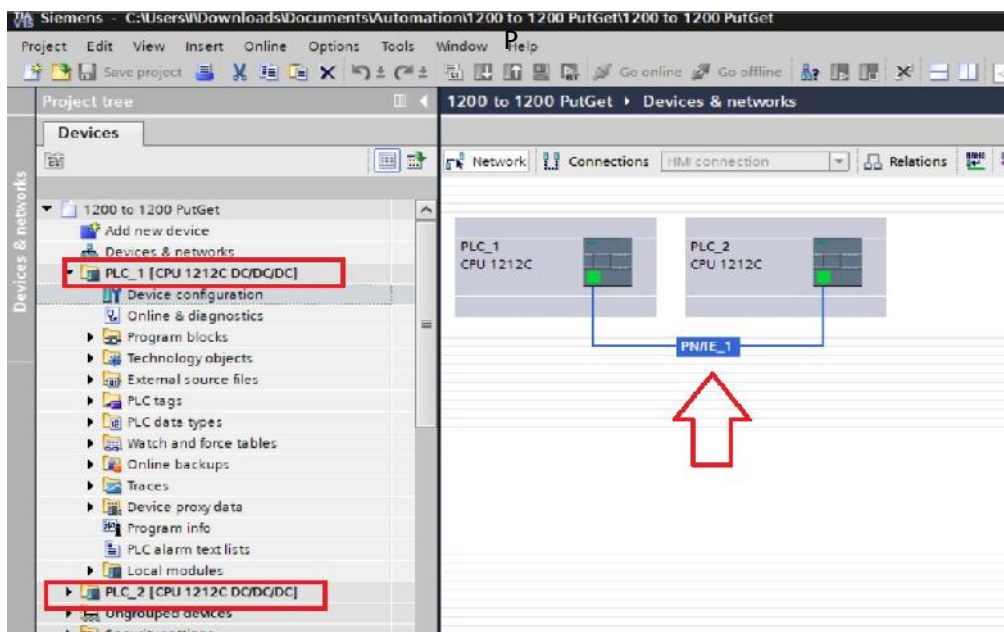
## ب- تنظیمات مورد نیاز برای برقراری ارتباط مابین دو CPU S7-1200 با دستورات PUT/GET

**توجه:** این آموزش مناسب افرادی است که آشنایی مقدماتی با برنامه نویسی S7-1200 در قالب نرم افزار TIA PORTAL دارند. سایر دوستان می توانند به منظور کسب اطلاعات بیشتر به مراجع معرفی شده در انتهای مقاله مراجعه نمایند.



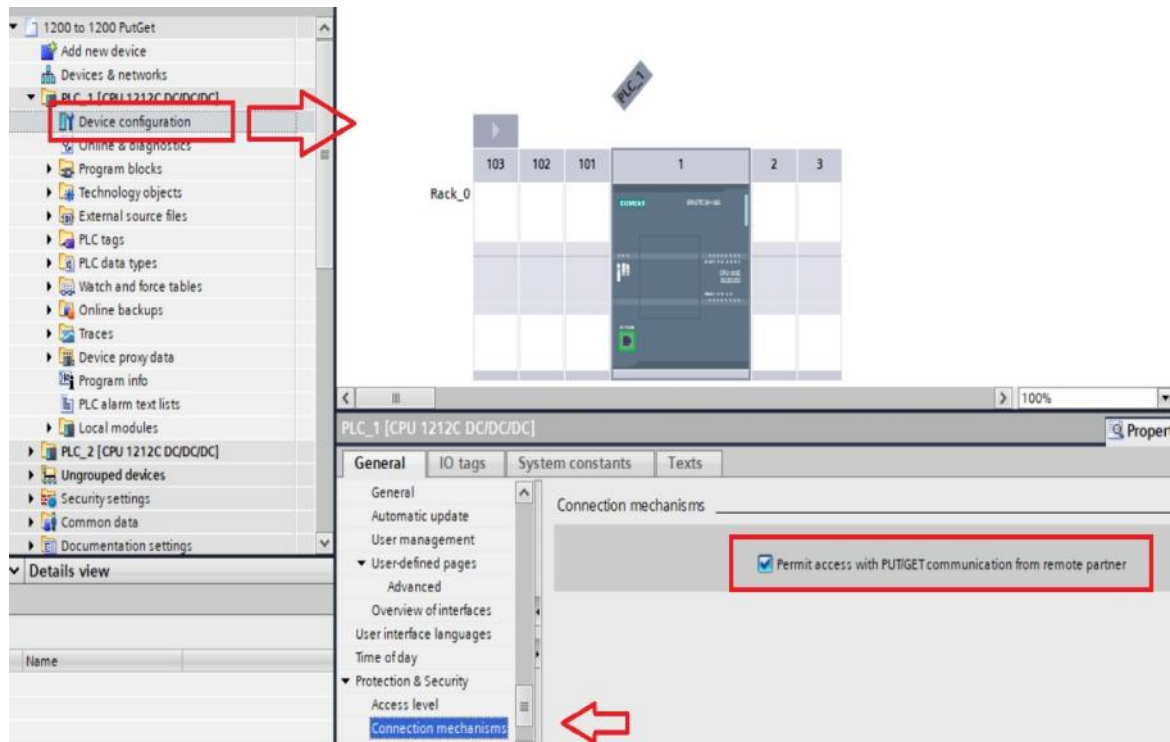
شکل ۲

ابتدا در نرم افزار TIA PORTAL پروژه ای جدیدی با استفاده از CPU1212C ایجاد می کنیم. Address IP مربوط به CPU اول را 192.168.0.1 در نظر می گیریم. سپس در محیط NETWORK View از کتابخانه تجهیزات در سمت راست صفحه، یک عدد CPU1212C دیگر با IP برابر با 192.168.0.2 به پروژه اضافه می کنیم. (توجه به ورژن CPU و مطابقت آن با ورژن CPU های سخت افزاری مهم است.)  
توجه: جهت این پیکربندی می بایست CPU ها دارای ورژن های بالاتر از 4.0 باشند.



شکل ۳

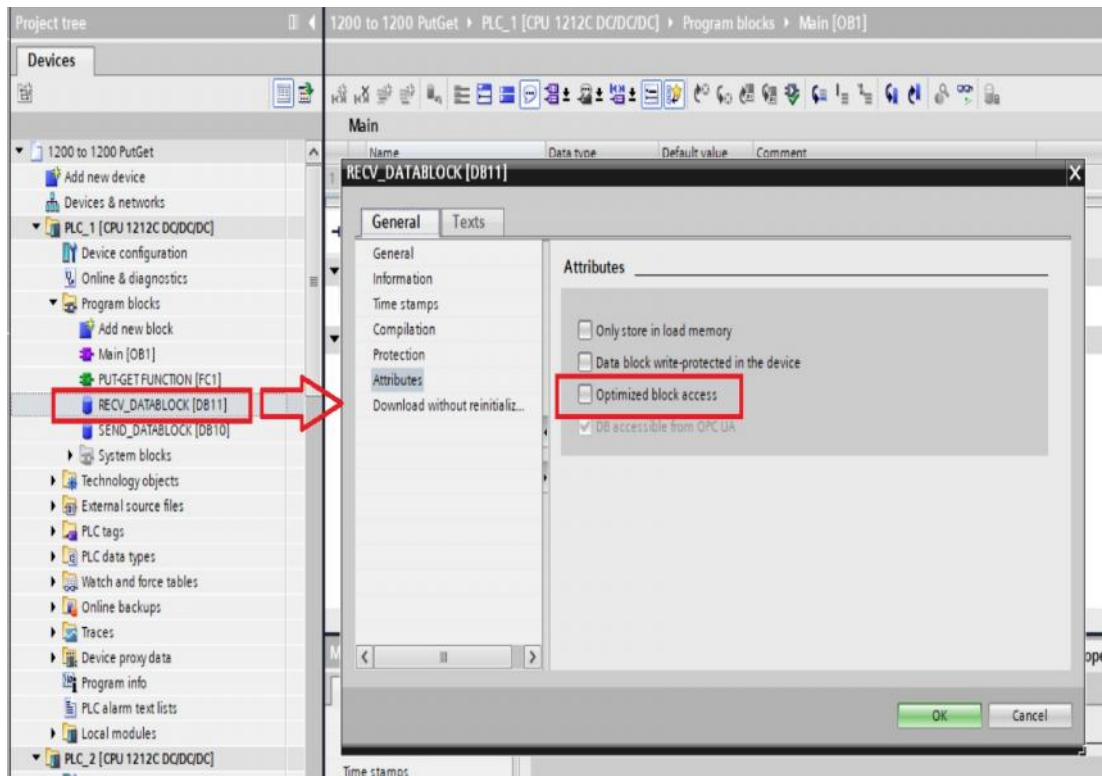
پس از قراردادن هر دو CPU ، در محیط NETWORK VIEW ارتباط آنها را برقرار نمایید . سپس در تنظیمات هر دو CPU تیک مربوط به دسترسی توسط دستورات PUT/GET را فعال نمایید . این مورد در شکل ۴ نمایش داده شده است .



شکل ۴

در مرحله بعد در هر کدام از دو CPU ، دو عدد دیتا بلوک تعریف می کنیم . بهتر است دیتا بلوکهایی با شماره های یکسان در نظر گرفته شود . در این مثال در هر CPU یک دیتابلوک با شماره DB10 و با نام SEND\_DATABLOCK و یک دیتابلوک دیگر با شماره DB11 و با نام RECV\_DATABLOCK تعریف می کنیم.

در تمام دیتابلوکها با کلیک راست بر روی نام دیتابلوک و باز کردن پنجره مشخصات ( همانند شکل ۵ ) تیک مربوط به Optimized Data Block را بر می داریم .



شکل ۵

در داخل Datablock متناسب با نیاز به حجم داده مورد انتقال مابین دو CPU ، تعدادی خانه حافظه ایجاد می کنیم . در این مثال یک متغیر از نوع Struct با نام DATA تعریف نموده و در داخل آن چهار عدد Int با نامهای دلخواه DATA1 تا DATA4 در نظر می گیریم . این عمل برای سایر DATA BLOCK های SEND و RECEIVE دیگر تکرار می گردد.

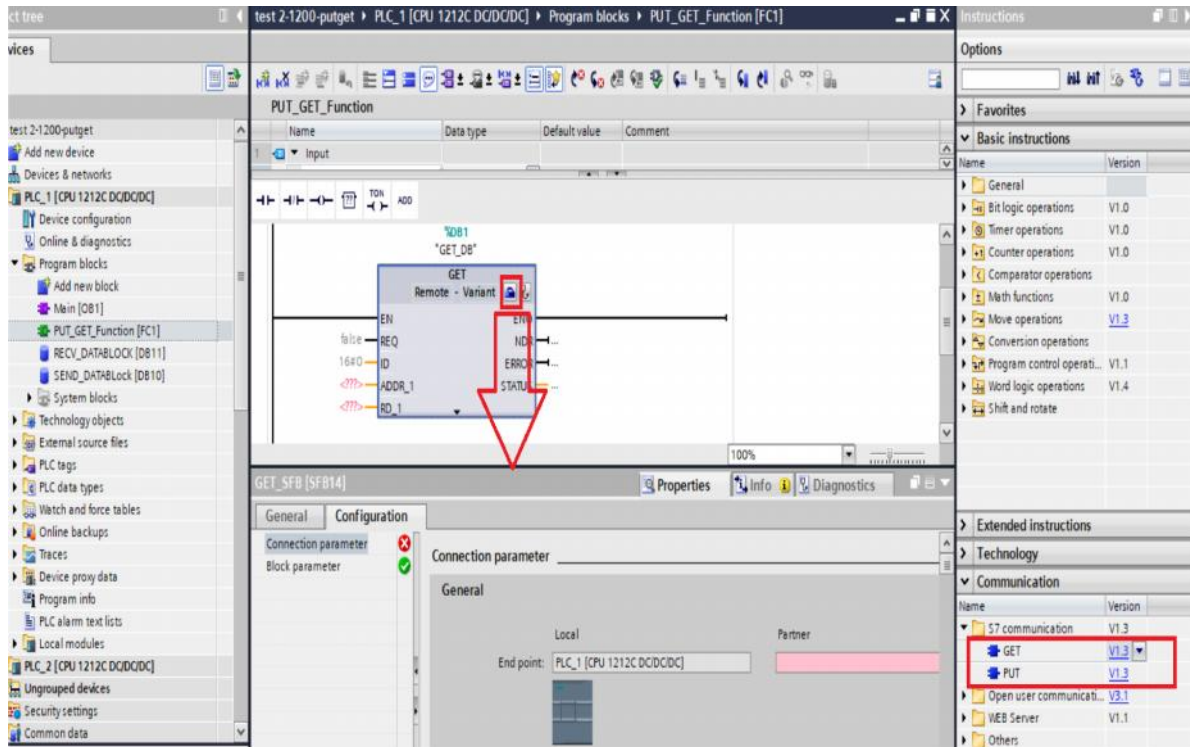
1200 to 1200 PutGet ▶ PLC\_1 [CPU 1212C DC/DC/DC] ▶ Program blocks ▶ RECV\_DATABLOCK [DB11]

Keep actual values Snapshot Copy snapshots to start values Load start values as actual value

RECV_DATABLOCK										
	Name	Data type	Offset	Start value	Retain	Accessible f...	Writa...	Visible in ...	Setpoint	Co
1	Static									
2	DATA	Struct	0.0			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
3	DATA1	Int	0.0	0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
4	DATA2	Int	2.0	0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
5	DATA3	Int	4.0	0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
6	DATA4	Int	6.0	0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		

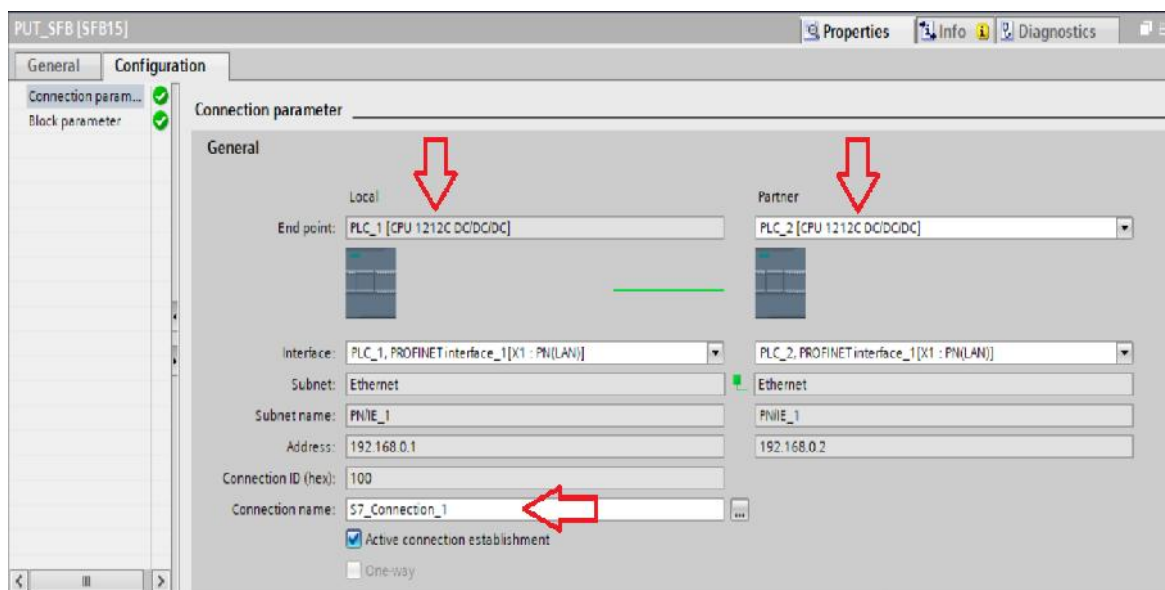
شکل ۶

در یکی از دو CPU که قرار است دستورات PUT و GET در آن نوشته شود، یک FC ایجاد نموده و در داخل آن از کتابخانه دستورات، دستور PUT و GET را بکار می‌بریم. با قراردادن این دستورات می‌توان همانند شکل ۷ پنجره تنظیمات مربوط به Connection را باز نموده و به تنظیمات آن پرداخت.



شکل ۷

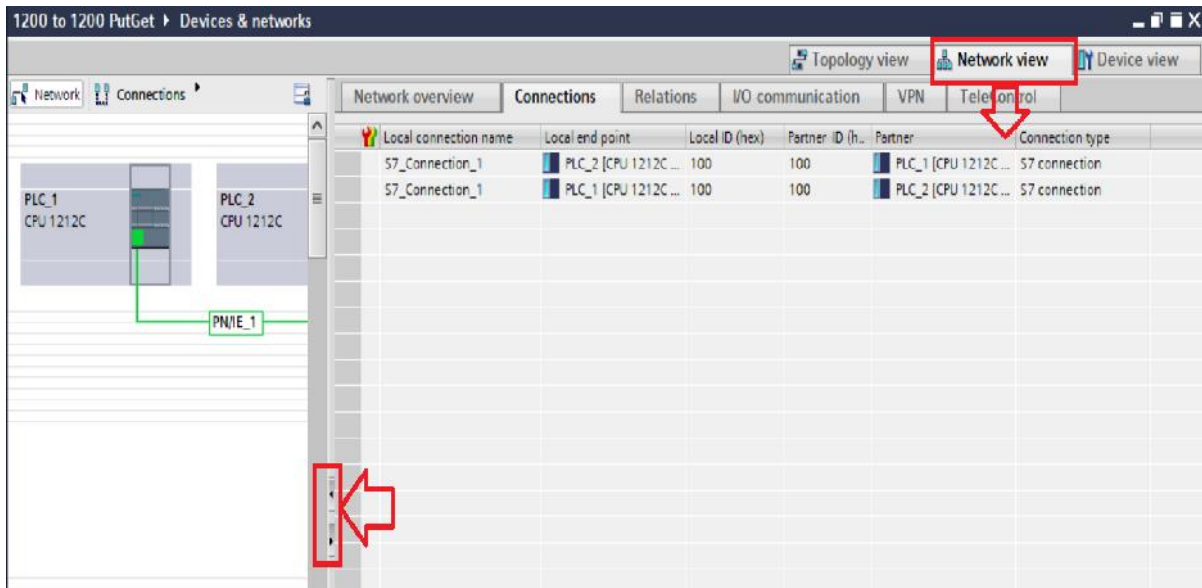
تنظیمات این ارتباط را مطابق با شکل ۸ در برگه Connection Parameter برای هر دو دستور PUT و GET انجام می‌دهیم.



شکل ۸

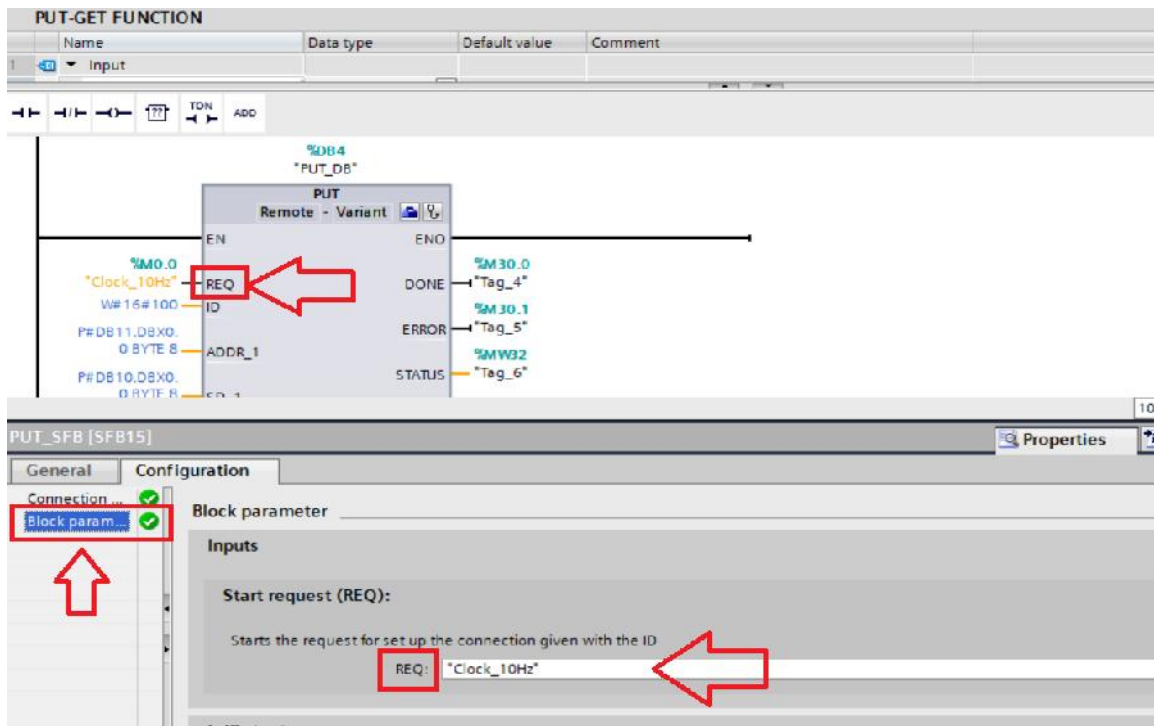


پس از ایجاد این ارتباط نوع کانکشن و جزئیات در محیط network view همانند شکل ۹ قابل مشاهده است .



شکل ۹

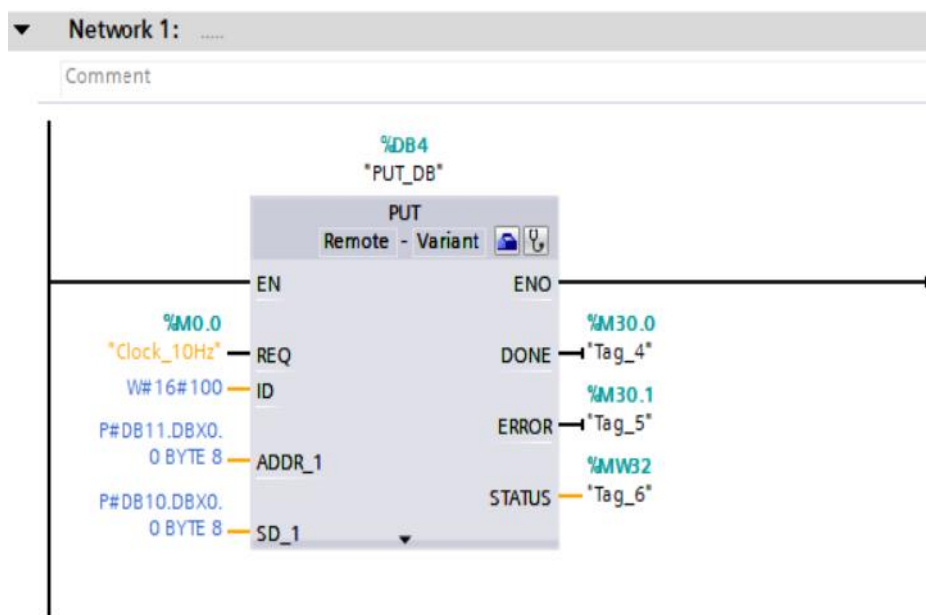
در مرحله بعد می بایست پارامترهای ورودی / خروجی مربوط به بلوکهای PUT و GET مقدار دهی شود . این کار هم با تایپ و مقدار دهی مستقیم در ورودی / خروجی های بلوک در داخل برنامه ممکن بوده و هم می توان از قسمت بلوک پارامتر همانند شکل ۱۰ صورت پذیرد .



شکل ۱۰

پس از ورود مقادیر نتیجه همانند شکل زیر خواهد شد . در ورودی REQ یک پالس ثابت ۱۰ هرتز که توسط CLOCK MEMORY در تنظیمات CPU ایجاد شده است ، جهت اعمال پالس های سنکرون سازی نقل و انتقال داده ، بکار رفته است . ID ، آدرس مربوط به PLC1 بوده که به صورت کد HEX وارد می شود . به دلیل اینکه بلوک PUT جهت ارسال داده از CPU اول ( Master ) یا به CPU دوم (Active Connection) استفاده می شود . لذا در ورودی SD\_1 آدرس دیتابلوک SEND در PLC1 ( DB10 ) و به صورت اشاره گر ( Pointer ) به نقطه شروع داده ها در این دیتابلوک تعریف شده است . در این آدرس تعداد ماکزیمم بایت قابل ارسال هم هشت عدد در نظر گرفته شده است . ( ۴ عدد داده به فرم INT ) . در ورودی ADDR\_1 هم آدرس مربوط به دیتابلوک دریافت کننده اطلاعات در CPU دوم ( Slave ) یعنی DB11 در نظر گرفته شده است .

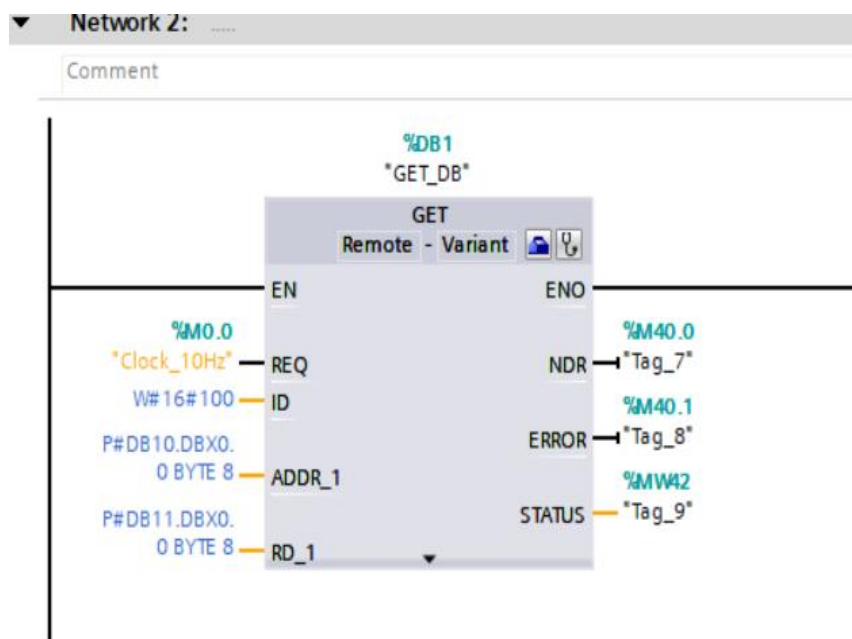
در پارامترهای خروجی هم بیت های مربوط به DONE ( اجرای صحیح بلوک ) ، ERROR ( نمایشگر وجود خطا ) رابه بیت های دلخواه حافظه و خروجی STATUS ( کد خطای ایجاد شده ) را به یک عدد خانه حافظه Word متصل می کنیم .



شکل ۱۱

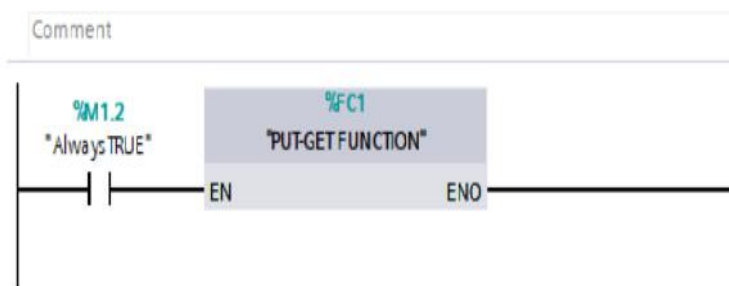
به صورت مشابه برای بلوک GET نیز مقدار دهی متغیرها صورت می پذیرد . شکل ۱۲ این تنظیمات را نمایش می دهد .





شکل ۱۲

پس از ایجاد بلوک ها و ذخیره سازی فانکشن FC1 ، در بدنه اصلی برنامه (OB1) این فانکشن را با شرط همیشه یک فراخوانی می کنیم .



شکل ۱۳

پس از اتمام برنامه نویسی و دانلود سخت افزار و نرم افزار هر دو PLC ، کانکشن برقرار شده و می توان جهت ارسال داده از یک PLC به PLC دیگر آنرا در یکی از خانه های تعریف شده در دیتابلوک SEND مربوط به PLC فرستنده قرار داده و سپس آنرا در خانه مشابه در دیتابلوک RECV در PLC گیرنده دریافت نمود .

## مراجع :

- ۱- مهدی توانا شیروان ، نوید انتظاری - " آموزش کاربردی PLC S7-1200 و شبکه صنعتی PROFINET " - مهدی توانا شیروان ، ۱۳۹۱
- ۲- وب سایت اینترنتی شرکت زیمنس

- 1) **PROFINET** : acronym for Process Field Net
- 2) flexibility
- 3) unidirectional
- 4) uncoordinated
- 5) bidirectional
- 6) block-oriented