کنترونیک

CONTRONIC

آغاز کار با نسخه آزمایشی Multiprog

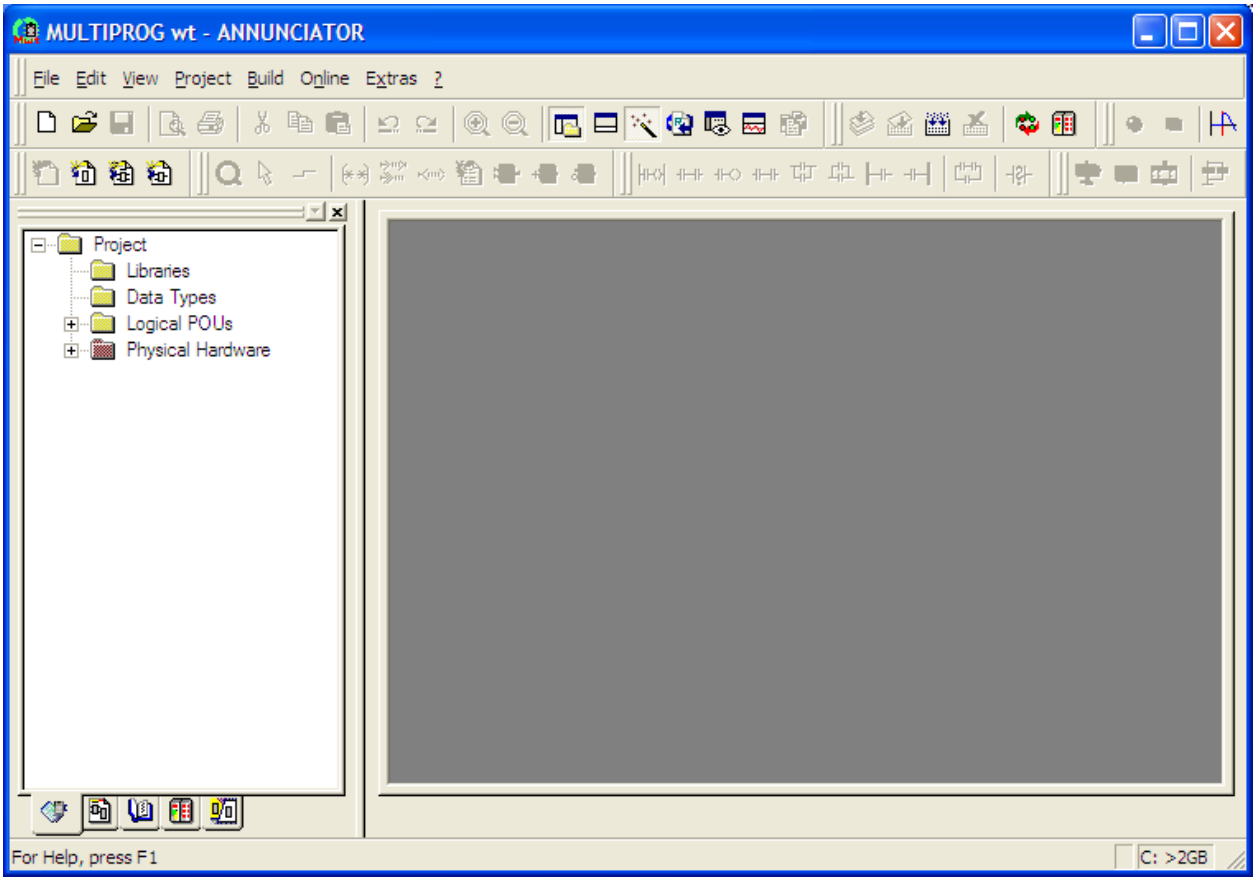
Getting Started with Multiprog Demo Version

شهریور ۱۳۹۰

آغاز کار با نسخه آزمایشی Multiprog

پس از نصب نرم افزار Multiprog، یک آیکون برروی desktop کامپیوتر شما نصب میشود. از این آیکون استفاده کرده و برنامه را اجرا کنید.

پس از نمایش پنجره ای که یادآوری برای نسخه demo است، نمای اصلی محیط برنامه نویسی Multiprog بشکل زیر نمایان می شود.



شکل ۱- نمای عمومی Multiprog

میله منوی نرم افزار در بالاترین بخش صفحه نمایش قرار داشته و گزینه های File، Edit، View، Project، Build، Online، Extras ، و ? در آن قرار دارند.

قبل از اینکه به ساخت یک برنامه plc بپردازیم، ذکر چند مطلب در مورد برنامه نویسی بر اساس استاندارد IEC61131-3 ضروری است. اول اینکه برنامه نویسی از تولید یک Project شروع می شود. در تولید پروژه لازم است ابتدا مشخص کنیم نام پروژه ما چیست؟ سخت افزار (Hardware) ما کدام plc است؟ وظایف یا Task های پروژه کدامند و چه نام دارند، برنامه هایی که باید اجرا شوند چیستند و ...

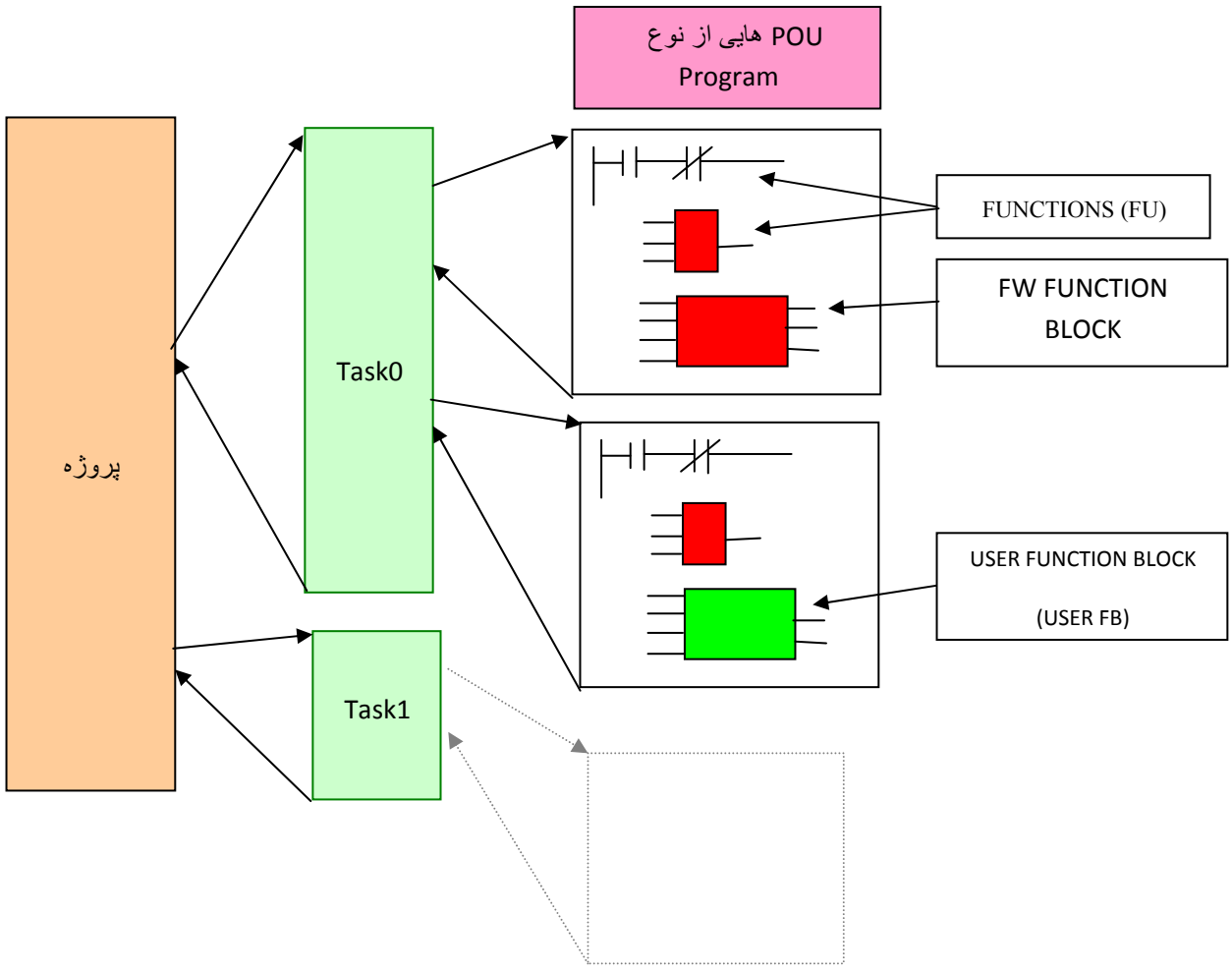
پس لازم است در اولین گام با این اسامی آشنا شویم.

ساختار پروژه

شکل زیر ساختار نرم افزاری یک پروژه را نشان میدهد.

- ✓ هر پروژه تعدادی Task را که شما تدوین می کنید اجرا می کند.
- ✓ هر Task تعدادی Program را فراخوانده واجرا می کند.
- ✓ در هر Program شما کارهای منطقی یا محاسباتی ای را که میخواهید مینویسید. برنامه ها در این بخش به یکی از زبانهای برنامه نویسی نوشته می شوند. این کار بشکل قرار دادن توابع در صفحه کار و برقراری اتصال و ارتباط بین توابع (FUNCTIONS) و بلوکهای تابع (FUNCTION BLOCKS) نوشته می شوند.

نکته: برخی از توابع در حافظه دائمی plc قرار دارند و شما صرفا از آنها استفاده می کنید که به آنها توابع Firmware و یااختصارا FW می گویند. برخی دیگر را کاربر تعریف می کند که توابع USER نامیده می شوند. مثلا تابع AND بدلیل کثرت کاربرد در حافظه دائمی قرار داشته و از نوع FW FUNCTION است در حالیکه ممکن است کاربر برای خود تابعی بنام MY_LOGIC تعریف کند که کار ویژه ای انجام دهد. به این نوع توابع USER FUNCTION گفته می شود. به همین ترتیب بلوکهای تابع (FUNCTION BLOCK) میتوانند از نوع FW و یا USER باشند. فرق بین FUNCTION و FUNCTION-BLOCK را در آینده توضیح خواهیم داد.



شکل ۲: ساختار اجرای برنامه در استاندارد IEC61131-3

همانطور که دیده می شود هنگام اجرای پروژه، Task ها برای اجرا فراخوانده می شوند. نحوه فراخواندن Taskها بستگی به نوع آنها دارد.

انواع Task ها عبارتند از:

- ✓ **CYCLIC TASKS:** این Task ها بصورت پی در پی و در مقاطع زمانی معینی برای اجرا فراخوانده می شوند. زمان اجرا برای هر Task دورانی قابل تنظیم است.
- ✓ **SYSTEM TASKS:** این Task ها در مواقع خاصی برای اجرا فراخوانده می شوند. مثلا هنگام روشن کردن دستگاه، هنگام راه اندازی سرد یا گرم و ...
- ✓ **EVENT TASKS:** این Task ها در صورت وقوع اتفاق خاصی، مثلا هنگام وقوع یک وقفه، برای اجرا فراخوانده می شوند.

برای هر Task برنامه هایی که قرار است با آن اجرا شوند را باید اختصاص داد. با فرارسیدن نوبت اجرای Task، برنامه های اختصاص یافته به آن Task اجرا میشوند.

POU ها در استاندارد IEC61131-3

عبارت POU مخفف Program Organization Unit است و نام عمومی بلوکهای برنامه PLC میباشند. POU ها واحدهای کوچک و مستقل نرم افزاریی هستند که کدهای برنامه را در خود جای میدهند.

در استاندارد IEC61131-3، سه نوع POU براساس کاربرد خاص آنها قابل تفکیک اند که عبارتند از:

- Program (PROG)
- Function Block (FB)
- Function (FU)

هر POU از دو بخش تشکیل شده است. بخش اعلان متغیرها (declaration part) و بخش متن برنامه یا Code (body). نا گفته پیداست که در بخش اعلان متغیرها، تمام متغیرهای مورد نیاز POU تعریف می شوند. دستورالعمل ها و کدهای برنامه ی هر POU نیز در بخش مربوطه و به یکی از زبانهای دلخواه نوشته میشوند.

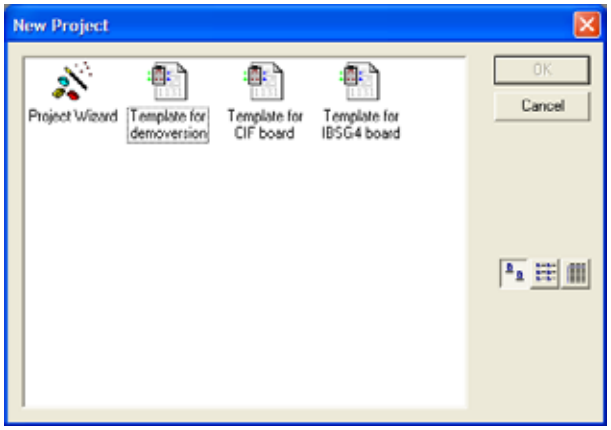
شکل فراخوانی POU ها

- o POU هایی از جنس PROG توسط Task ها فراخوانده می شوند. بنابراین نمی توان اجرای FB یا FU را به Task نسبت داد. این موضوع بسیار بدیهی است زیرا اینها در حقیقت قطعاتی هستند که کاربر برای برنامه نویسی بکار میبرد.
- o POU هایی از جنس FB و FU در POU هایی از جنس برنامه یا PROG بکار میروند.
- o هنگام تدوین FB یا FU توسط کاربر برای ساختن USER FB/FU، بدیهی است که میتوان از سایر FB/FU های دیگر بهره برد.

حالا که با اصلی ترین عبارات رایج در استاندارد آشنا شدیم، وقت آن رسیده که پروژه کوچکی ساخته و کارکرد آن را بررسی نماییم. هنگام ساختن پروژه چنانچه توضیحات دیگری لازم باشد در همانجا بیان خواهد شد.

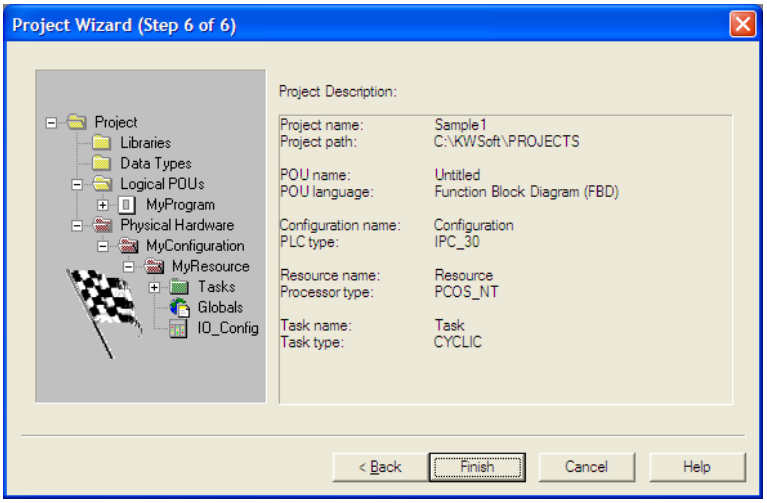
ساختن پروژه نمونه

در برنامه Multiprog و از منوی File، گزینه New Project را انتخاب کنید. پنجره زیر باز می شود.



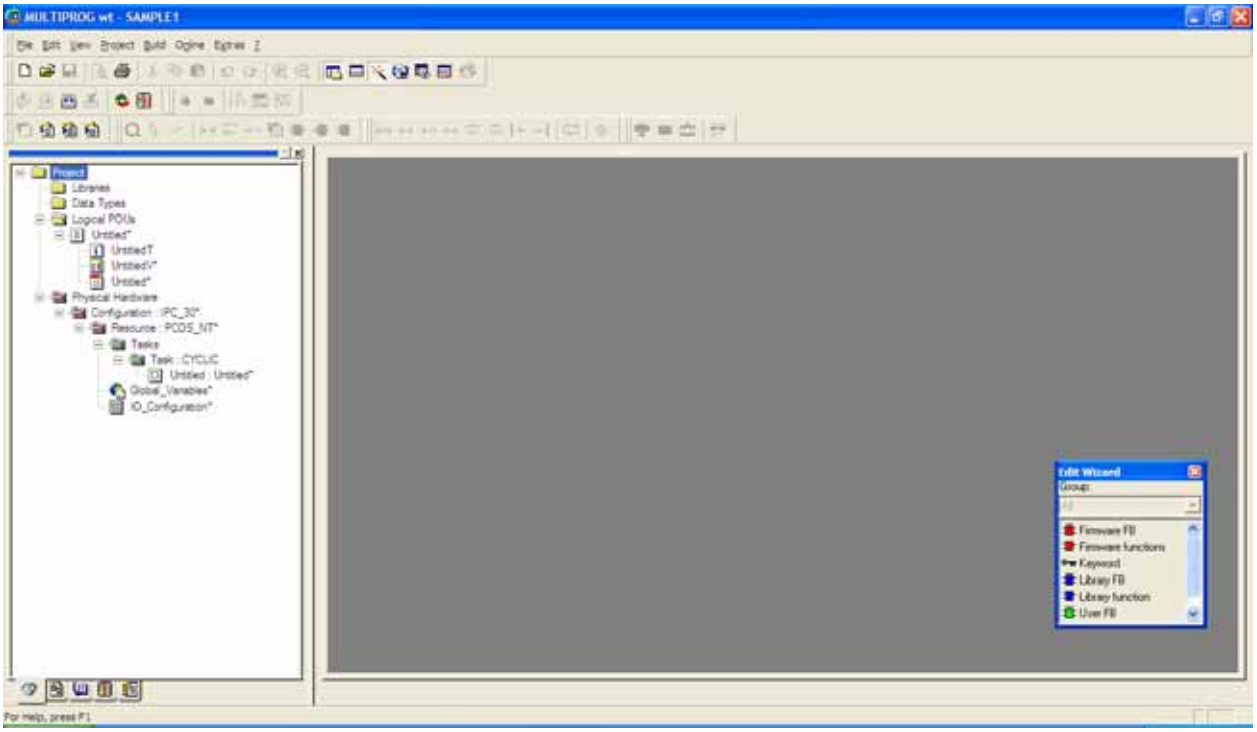
اولین گزینه یعنی Project Wizard بمعنی "جادوگر ساخت پروژه" را انتخاب کرده و کلید OK را بزنید. در ادامه پنجره های متوالی ظاهر شده و سوالاتی در مورد نام پروژه، نام POU اصلی و زبان برنامه نویسی، سخت افزار یا Configuration، منابع پردازش یا Resource و نام Task پرسیده میشود. در پاسخ به این سوالات همان مقایر پیش فرض را قبول کنید تا پروژه تکمیل شود.

دقت کنید که نام پروژه را Sample1 (بدون حرف خالی یا Space) بگذارید. در صورتیکه اشتباهی نکرده باشید، آخرین پنجره، مطابق شکل زیرخلاصه ای از انتخابها را نمایش داده و منتظر تصمیم نهایی شما می شود.



با کلیک روی کلید Finish، پروژه ساخته می شود.

پنجره های Project Wizard تمام شده و اسکلت اصلی پروژه مانند شکل زیرساخته خواهد شد.



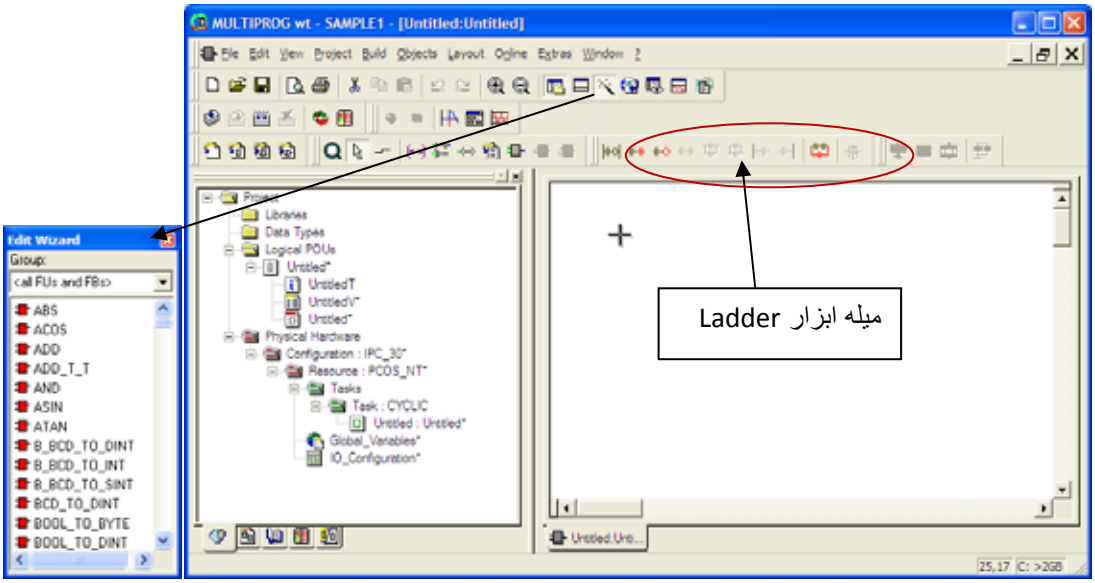
در پنجره سمت چپ که Project Tree Window یا پنجره درخت پروژه نام دارد، ساختار اصلی پروژه دیده میشود. در درخت پروژه یک POU از جنس PROG بنام Untitled در زیر شاخه Logical POUs ساخته شده است. در زیر شاخه Physical Hardware مشاهده میشود که Configuration ای براساس کنترلرهای IPC_30 و پردازنده ای (Resource) از نوع PCOS_NT برای آن منظور شده است. از این پردازنده خواسته شده که Task ای بنام Task از نوع Cyclic را پردازش نماید. از Task نیز خواسته شده که نسخه ای از برنامه ای بنام Untitled را اجرا نماید.

با انتخاب بعضی ازگره ها و کلیک راست برروی آن می توان خواص (Properties) و تنظیمات (Settings) آن را مشاهده کرد. فعلا از هر گونه تغییر در آنها خودداری نمایید.

درگره Logical POUs و زیر شاخه Untitled سه زیر شاخه بنامهای UntitledT با آیکونی برنگ آبی، UntitledV با رنگ زرد و Untitled به رنگ سرخ دیده میشوند. این ها در حقیقت صفحاتی هستند که باید ویرایش شوند. کاربری این سه صفحه عبارتند از:

- Untitled به رنگ سرخ برای متن اصلی برنامه بکار میرود
- UntitledV به رنگ زرد برای معرفی و اعلان متغیرهای برنامه بکار میرود
- UntitledT به رنگ آبی برای نوشتن متون راهنمای برنامه و در صورت نیاز بکار میرود

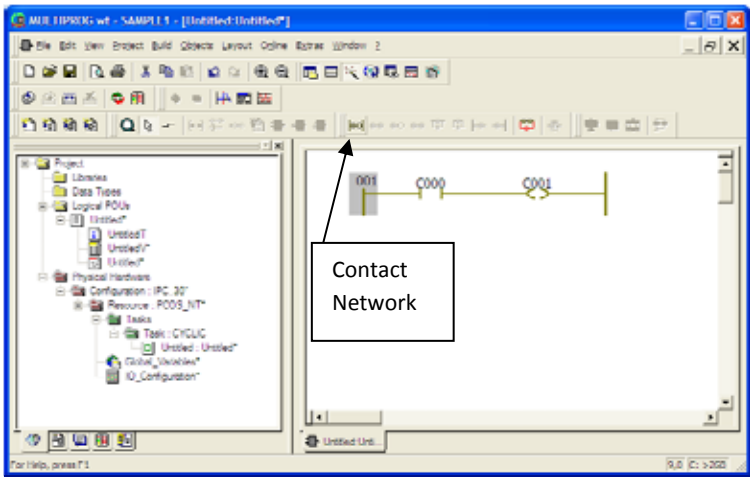
برروی Untitled سرخ رنگ کلید دوبل کنید تا صفحه کارآن باز شود. در بالا و سمت چپ صفحه (Worksheet) در محلی دلخواه کلیک کنید. علامتی بشکل + در همان محل ظاهر میشود. اینجا مکانی است که از این پس برنامه شما نوشته میشود.



توابع لازم برای برنامه نویسی در پنجره ای بنام Edit Wizard قرار دارند. برای برنامه نویسی بزبان Ladder نیز میتوان ازمیله ابزاری که در بالای صفحه قرار دارند استفاده کرد.

نکته: *پنجره ی Edit Wizard یا جادوگر ویرایش را میتوان پیدا و پنهان کرد. این کار با انتخاب آیکون مربوطه که بشکل چوب جادوگریست انجام میشود.*

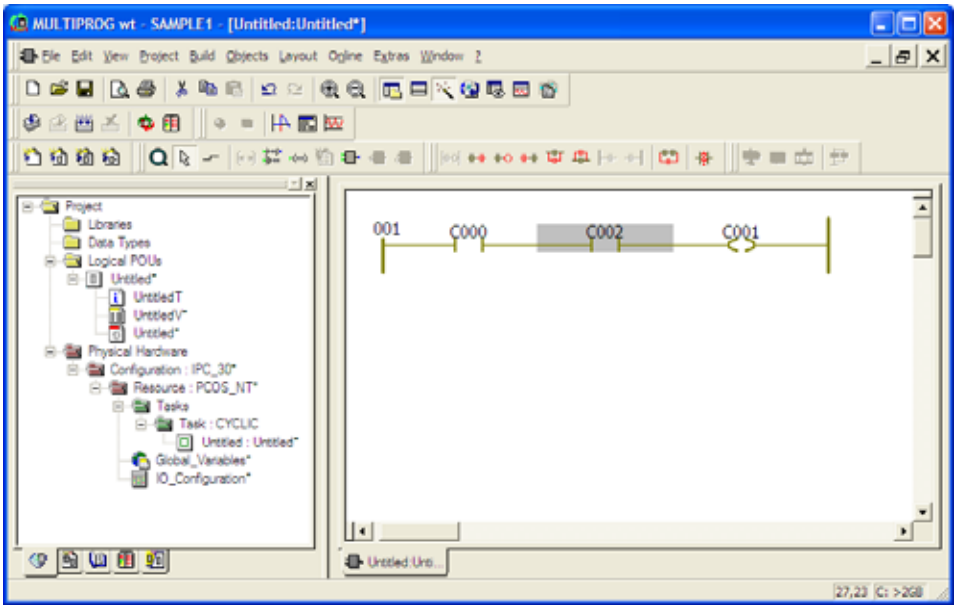
از میله ابزار Ladder گزینه Contact Network را انتخاب کنید. در همان محلی از صفحه که انتخاب شده بود یک Network بزبان LD یا همان Ladder ایجاد میشود.



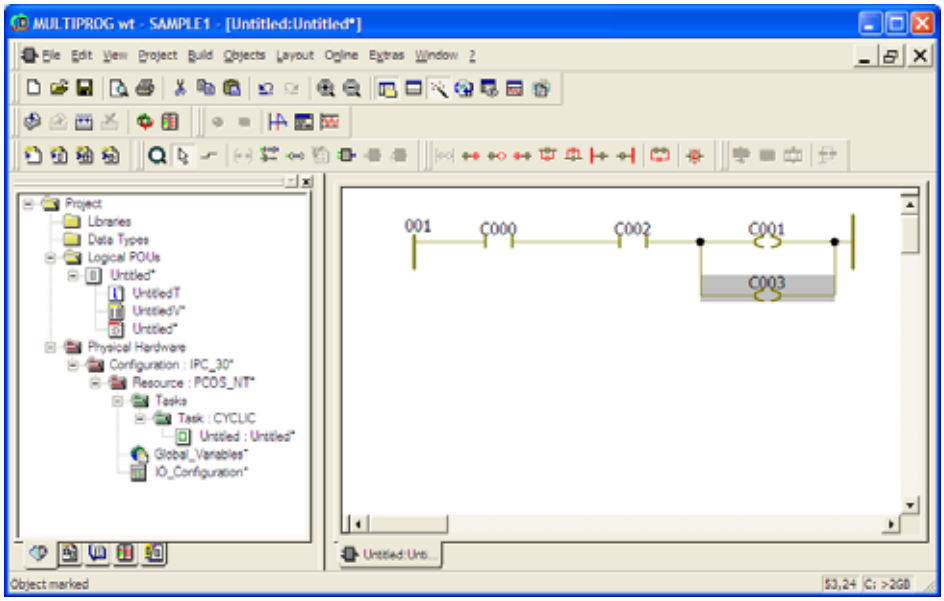
یک Network بزبان LD با دو متغیر به اسامی پیش فرض به اسامی C000 و C001 تشکیل میشود. این متغیرها هنوز نهایی نشده و نا مشخص اند. تعریف آنها را بعدا کامل خواهیم کرد.

چنانچه ماوس را روی هر یک از قطعات قرار دهید و کلیک (تک کلیک) کنید، آن قطعه Select شده و رنگ آن طوسی میشود.

کنتاکت C000 را Select کرده و از روی میله ابزار LD یک کنتاکت در سمت راست آن قرار دهید. این کار بوسیله کلید Add Contact Right انجام میشود. کنتاکت C002 مطابق شکل زیر اضافه خواهد شد.



در مرحله بعد Coil خروجی بنام C001 را select کرده وکلید Add contact/coil below را بزنید. یک خروجی دیگر بنام C003 در زیر Coil قبلی ایجاد می شود. شکل زیر:

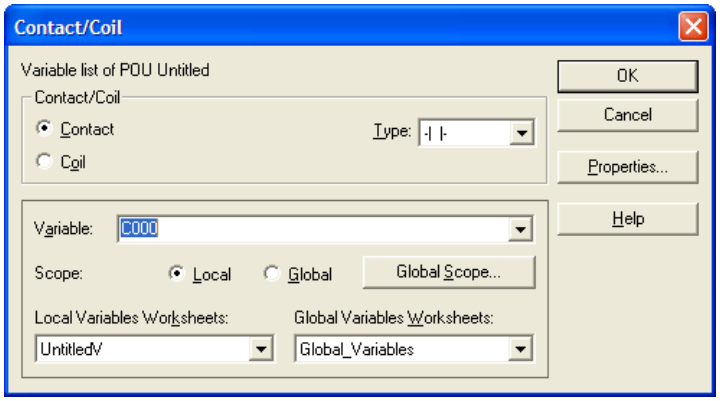


با Select بودن C003، کلید Toggle properties of coil/contact را یکبار کلیک کنید. درخروجی C003 علامت "/" قرار گرفته و بمعنی Not شدن آن است. با کلیک های بعدی علامتهای "S" بمعنی Set و "R" بمعنی Reset نیز ظاهر خواهد شد. با کلیک های متوالی همان گزینه Not با علامت "/" را انتخاب کنید.

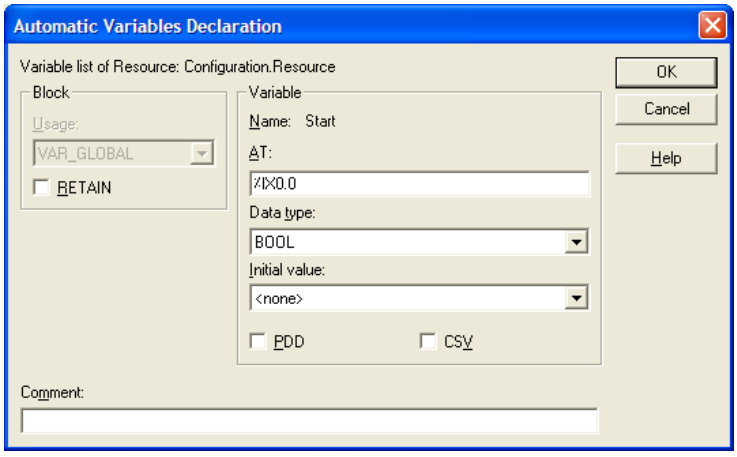
اعلان متغیرها:

همانطور که میدانید متغیرهای C000 تا C003 فقط نمادین بوده و هنوز تعریف نشده اند. برای تعریف آنها به شرح زیر عمل کنید.

روی C001 کلیک دوبل کنید. پنجره زیر باز میشود.

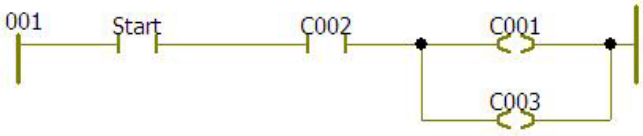


ابتدا باید در قسمت Variable نام مناسبی انتخاب کنید. نام متغیر را به Start تغییر دهید. در بخش Scope گزینه Global را انتخاب کنید. با زدن کلید OK پنجره شکل زیر نیز باز میشود.



در قسمت Variable و در بخش AT میتوان آدرس متغیر را وارد کرد. فرض کنید که Start از ورودی I0.0 وارد میشود. پس آدرس آن %IX0.0 است. در مورد آدرس دهی این نوع نامگذاری بیشتر توضیح خواهیم داد*.

در قسمت Data type نیز همان گزینه BOOL مناسب است. سایر مقادیر و گزینه ها را فعلا تغییر ندهید. کلید OK را بزنید تا پنجره بسته شود. برنامه بشکل زیر تغییر خواهد کرد.



*آدرس دهی متغیر ها

در IEC61131-3 متغیرهای پایه به شرح زیر معرفی شده اند.

Data type	Description	Size	Range	Default initial value
BOOL	Boolean	1	0...1	0
SINT	Short integer	8	-128...127	0
INT	Integer	16	-32768...32767	0
DINT	Double integer	32	-2.147.483.648 up to 2.147.483.647	0
USINT	Unsigned short integer	8	0 up to 255	0
UINT	Unsigned integer	16	0 up to 65535	0
UDINT	Unsigned double integer	32	0 up to 4.294.967.295	0
REAL	Real numbers	32	1.18 x 10^-38 up to 3.40 x 10^38	0.0
TIME	Duration	32		t#0s
BYTE	Bit string of length 8	8		0
WORD	Bit string of length 16	16		0
DWORD	Bit string of length 32	32		0

متغیرهای تک بیتی (BOOL) به یک بیت فضا نیاز دارند. متغیرهای نوع BYTE، SINT و USINT به ۸ بیت یا یک بایت ومتغیرهای نوع WORD، INT و UINT به دو بایت فضا نیاز دارند و ...

هنگامی که محل حضور متغیری را می خواهید مشخص کنید باید نوع متغیر و اندازه فضای اختصاص یافته با هم سازگار باشند. برای اختصاص فضا که به یکی از اندازه های bit، byte، word و double word است از حروف اختصاری زیر استفاده می نماییم.

Size متغیر	در فضای ورودی	در فضای خروجی	در فضای حافظه
BOOL	%IX	%QX	%MX
BYTE	%IB	%QB	%MB
WORD	%IW	%QW	%MW
DWORD	%ID	%QD	%MD

مثلا برای قرار دادن متغیری بنام Press_100 از نوع INT در فضای حافظه M به آدرس ۱۰۰ باید در قسمت AT ازعبارت %MW100 استفاده کرد.

آدرس متغیرهای بیتی، هم به آدرس بایت و هم شماره بیت نیاز دارند. مثلا متغیری بنام Press_Fault را میتوان در آدرس %MX12.3 قرار داد. یعنی بیت سوم از بایت ۱۲ فضای M.

هنگام آدرس دهی متغیرها باید به هم پوشانی فضاها دقت کرد. مثلا %MW100 شامل MB100 و MB101 است.

برای متغیرهایی که از جنس ورودی (I) و خروجی (Q) نیستند، میتوان آدرس آن را مشخص نکرد. در این صورت کامپایلر برنامه بصورت اتوماتیک و با سایز مناسب فضا اختصاص داده و همپوشانی بوجود نخواهد آمد.

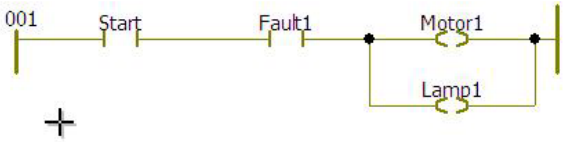
تکمیل برنامه:

برروی متغیر C002 کلیک دوبل کرده و به روشی که بیان شد آن را ورودی (I) بنام Fault1 از نوع BOOL و در آدرس %IX0.1 و با منطق معکوس تعریف کنید. خروجی ها را هم مانند جدول زیر تکمیل کنید.

Scope	AT	Type	Variable Name
Global	%IX0.0	BOOL	Start
Global	%IX0.1	BOOL	Fault
Global	%QX0.0	BOOL	Motor1
Global	%QX0.1	BOOL	Lamp1

توجه داشته باشید که هنگام اعلان متغیرها، قوانین نامگذاری و آدرس دهی را باید رعایت نمود. مثلا درنام متغیر نباید از علائم خاصی مثل ?,- ,+ ,/ ,* و فاصله استفاده کرد. هنگام آدرس دهی (AT) نیز باید از حروف بزرگ استفاده کرد.

چنانچه تمام متغیرها را بخوبی ویرایش نموده باشید، برنامه بشکل زیر درخواهد آمد.



اما متغیرهایی که شما ویرایش کرده اید نیز در فایل Global_Variables وارد شده اند. برای دیدن آنها به درخت پروژه مراجعه کرده و روی Global_Variables کلیک دوبل نمایید. فایل مربوطه باز شده و متن زیر را در آن مشاهده می کنید.

```
(* Here you may define your global variables, using the
        VAR_GLOBAL ... END_VAR construct . *)

(* Defines for System flags  *)

VAR_GLOBAL
        PLCMODE_ON              AT %MX 1.0.0  :  BOOL;
        PLCMODE_RUN             AT %MX 1.0.1  :  BOOL;
        PLCMODE_STOP            AT %MX 1.0.2  :  BOOL;
        PLCMODE_HALT            AT %MX 1.0.3  :  BOOL;
        PLCDEBUG_BPSET          AT %MX 1.1.4  :  BOOL;
        PLCDEBUG_FORCE          AT %MX 1.2.0  :  BOOL;
        PLCDEBUG_POWERFLOW      AT %MX 1.2.3  :  BOOL;
        PLC_TICKS_PER_SEC       AT %MW 1.44   :  INT;
        PLC_SYS_TICK_CNT        AT %MD 1.52   :  DINT;
END_VAR

VAR_GLOBAL (*AUTOINSERT *)
        Start                   AT %IX0.0   : BOOL;
        Fault1                  AT %IX0.1   : BOOL;
        Motor1                  AT %QX0.0   : BOOL;
        Lamp1                   AT %QX0.1   :BOOL;
END_VAR
```

به بخش پائین آن توجه کنید. اسامی ۴ متغیری که شما ویرایش نموده اید دیده می شوند. محل استقرار آنها در بخش AT و type متغیرها در آخرین قسمت آمده است. تمام این ۴ متغیر در بخشی محصور شده اند که با عبارت VAR_GLOBAL شروع شده و با عبارت END_VAR بپایان رسیده است.

در بالای صفحه تعدادی متغیر سیستمی نیز وجود دارند که به صورت اتوماتیک برای این plc خاص تعریف شده و در صورت نیاز میتوانید از آنها نیز استفاده کنید. فعلا با آنها کاری نداریم.

اینها متغیرهای عام یا Global هستند. همانطور که قبلا گفته شد، هر POU ای که شما ویرایش می کنید، صفحه گسترده یا Worksheet ای بنام برنامه و با پسوند V دارد. مثلا برنامه Untitled دارای یک Worksheet بنام UntitledV دارد.

متغیرهای برنامه شما در این صفحه درج میشوند. برای دیدن آنها به درخت پروژه رفته و روی UntitledV کلیک دوبل نمایید. صفحه باز شده و به متن زیر برخواهید خورد.

```
VAR_EXTERNAL (*AUTOINSERT *)
        Start    :        BOOL;
        Fault1   :        BOOL;
        Motor1   :        BOOL;
        Lamp1    :        BOOL;
END_VAR
```

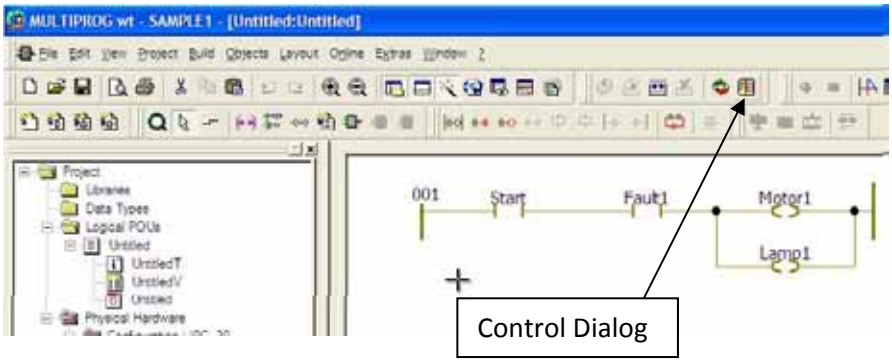
ملاحظه می کنید که نسخه ای از همان متغیرها در این صفحه گسترده نیز آمده ولی تنها شامل نام و نوع متغیرهاست. این متغیرها در محدوده ای که با VAR_EXTERNAL شروع شده و با END_VAR پایان می یابد، محصور شده اند. عبارت VAR_EXTERNAL میگوید که تعریف آنها در خارج از این Worksheet (دراین مثال در Global_Variables) انجام شده است.

نکته: *فعلا POU ی ما دارای متغیری از نوع محلی (Local) نیست. اگر چنین بود، آنها نیز در همین worksheet یعنی در UntitledV درج میشد و با عبارات VAR و END_VAR محصور می گردیدند. برای شرح کامل این موضوع به Help نرم افزار و یا مدارک PLC500 Nseries مراجعه نمایید.*

عیب یابی برنامه:

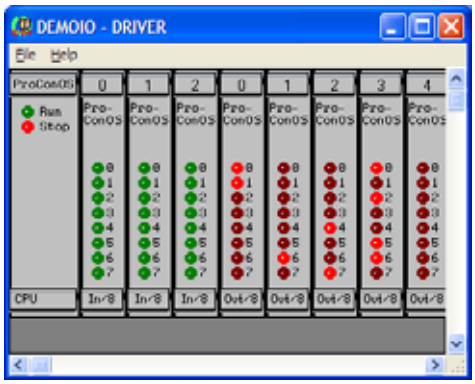
وقت آن رسیده که برنامه نوشته شده را مورد ارزیابی قرار دهیم. از منوی Build گزینه Make را انتخاب کنید. برنامه کامپایل شده و نتیجه آن در پنجره Message Window ظاهر خواهد شد. برای دیدن اخطارها (Warnings) و خطاها (Errors) به زیرمجموعه های مربوطه مراجعه کنید.

اگر برنامه بدون خطا باشد میتوان آن را برای اجرا به plc ارسال کرد. برای ارسال به شکل زیر اقدام کنید.

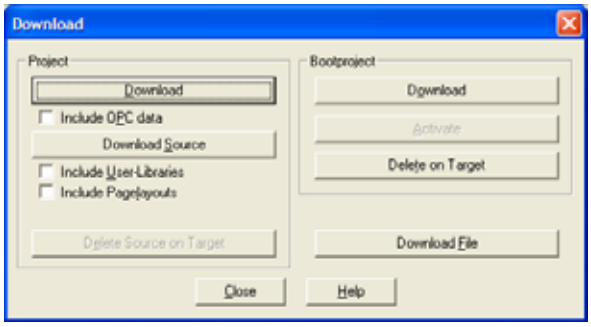


با زدن کلید Control Dialog، دیالوگ یا پنجره ارتباط با plc بنام Resource به شکل زیر باز میشود.

با پیدا شدن این پنجره شکل گرافیک plc فرضی نیز ظاهر می شود. اندازه پنجره ها را طوری تنظیم کنید که بخشی از برنامه و plc دیده شوند.

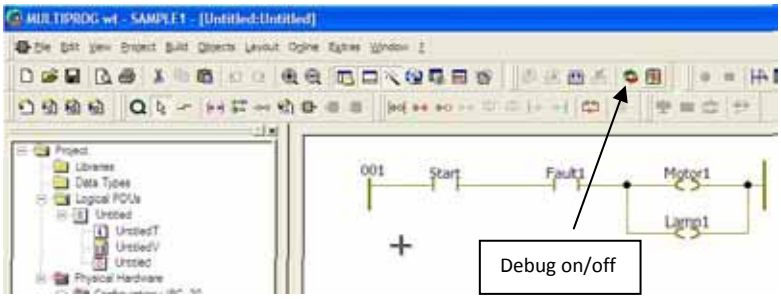


برای ارسال برنامه از پنجره Resource گزینه download را بزنید. پنجره download باز میشود.

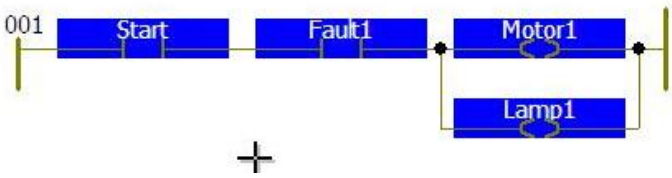


از سمت چپ بالا و در ناحیه Project کلید Download را کلیک کنید. برنامه به حافظه plc منتقل می شود. بعد از پنجره ی Resource گزینه Cold بمعنی راه اندازی سرد را بزنید. دستگاه plc راه اندازی شده و برنامه را اجرا میکند.

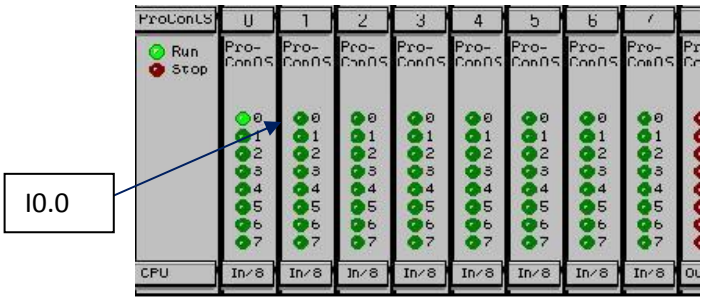
برای مشاهده Status یا وضعیت اجرای برنامه از کلید Debug on/off که در شکل زیر دیده میشود استفاده کنید.



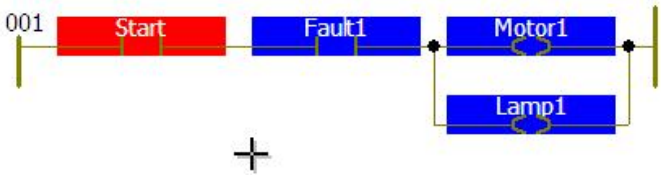
بازدن این کلید Status همزمان اجرای برنامه مطابق شکل زیر دیده میشود.



ورودی I0.0 بر روی سیمولاتور plc را مطابق شکل زیرکلیک کنید. ورودی روشن می شود.



در صفحه Multiprog هم وضعیت ورودی با رنگ قرمز مشخص میشود.



ورودی Fault یعنی I0.1 (ورودی پایین تر) را هم فعال کنید و وضعیت را بررسی کنید.

با زدن مجدد کلید Debug on/off ارتباط با plc قطع میشود لیکن برنامه در plc همچنان اجرا میشود. برای خاتمه دادن به اجرای برنامه سیمولاتور باید در Task bar ویندوز برنامه PcSim32 را Terminate کنید. اگر این کار انجام نشد ابتدا پنجره Resource را ببندید و سپس PcSime32 را Terminate کنید.

سیمولاتور plc مورد بحث درحقیقت برنامه Demo ی یک plc نرم افزاری بنام ProConOs (Process Control OS) است که به عنوان plc های PC-Based بکار میرود.

بیشتر بیاموزید:

یک POU دیگراز نوع PROG بنام Second بنویسید و به Task اضافه کنید. کدهای این برنامه را بزبان FBD بنویسید. مراحل برنامه نویسی، تعریف متغیرها، کامپایل، انتقال به plc و مشاهده Status را برای این برنامه تکرار کنید.

هرچند سیمولاتور اخیر با منابع محدود کار کرده و برنامه های بزرگ را نمی توان با آن آزمایش کرد، لیکن برای آموختن نکات بسیاری میتواند مورد استفاده قرار گیرد. پیشنهاد میشود با خواندن Help نرم افزار به بخشهای مختلف آن سرکشی کرده و دانش برنامه نویسی خود را ارتقا دهید. پس از آن هم می توانید نرم افزار واقعی را از شرکت کنترونیک تهیه نمایید.