

Quick Start Mect Suite



Via Enrico Fermi, 57/59 - 10091 ALPIGNANO (TO)

☎ Telefono: +39 (0)11 9664616 Fax: +39 (0)11 9664610

E-mail: srlmect@mect.it - C.F. e P.I. 04056380019

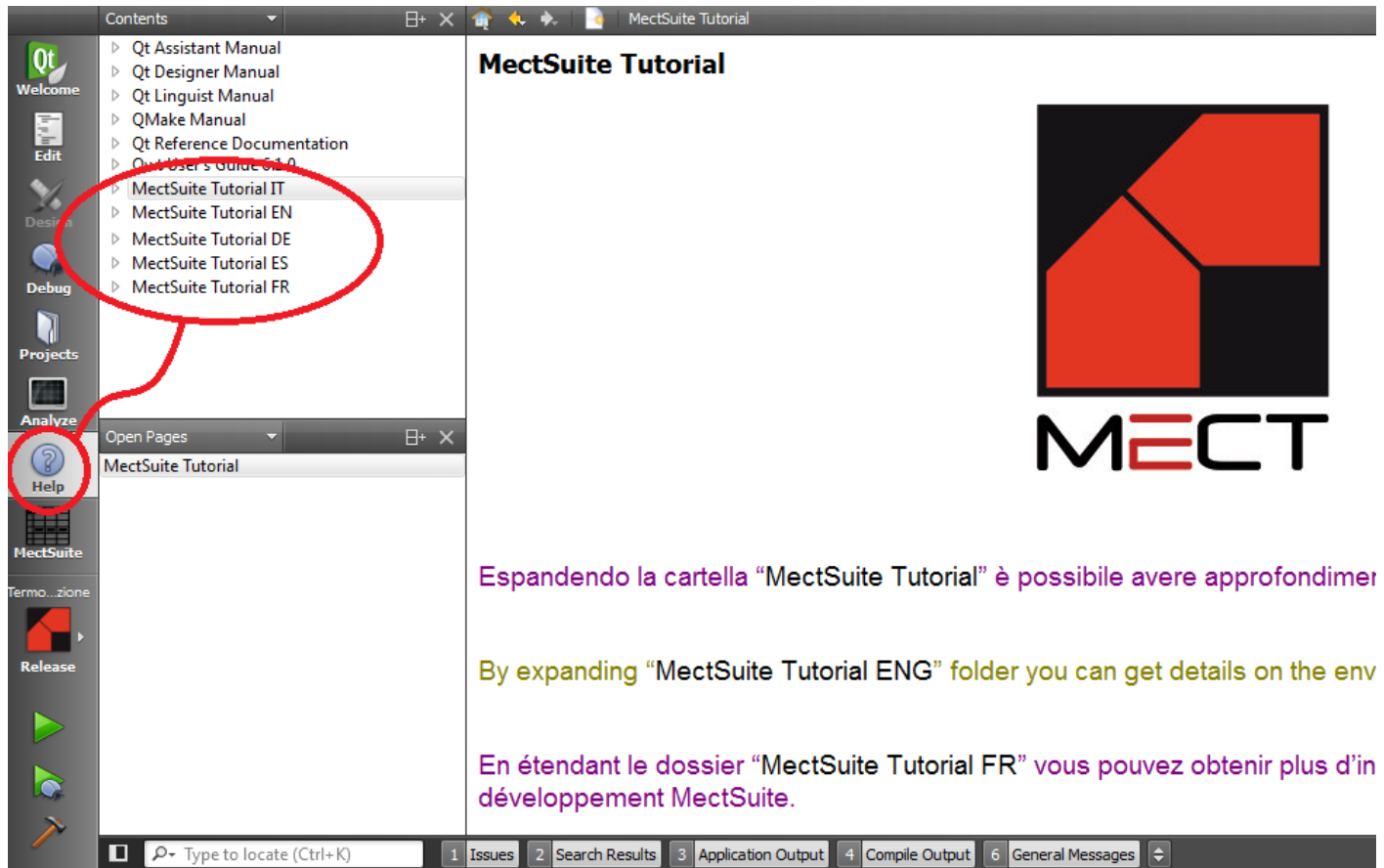
ME7041_24
05/21

INDEX

1	SETUP	1
1.1	Update	1
1.2	Hardware setup.....	2
2	EXERCISE	2
2.1	Description	2
2.2	PROJECT CREATION	3
2.2.1	New project.....	3
2.2.2	Check LAN connection between Operator panel and PC.....	4
2.2.3	Modify “splash” screen.....	5
2.2.4	Variables – Device management.....	5
2.2.4.1	Crosstable creation.....	6
2.2.4.2	System configuration	8
2.2.4.3	Trends configuration.....	9
2.2.5	Add graphics	10
2.2.6	Variables linking	11
2.2.7	Add a new page.....	11
2.2.8	Insert source code.....	13
2.2.9	Activate page change	14
2.2.10	Modifying elements	15
2.2.11	“Oscilloscope” page.....	17
2.2.12	“trend” page	18
2.2.13	“store” page.....	18
2.2.14	“alarms” page.....	19
2.2.15	“recipe_select” page.....	20
2.3	LANGUAGES TRANSLATION.....	21
2.3.1	Add a new file of language	22
2.4	BUILD & RUN	23
2.4.1	Executable creation	23
2.4.2	Send data to operator panel	24
2.4.3	Operator panel web interface	25
2.5	Debug	26
2.6	VNC activation.....	26
2.7	Open an existing project.....	26
2.8	Open the PLC software	27
2.9	Device variables (Local I/O)	32
2.10	Input / output configuration.....	32

1 SETUP

WARNING: read the “MectSuite Tutorial EN” for further details.



1.1 Update

Before upgrade read the “changelog.txt” file on Mect web site in “DOWNLOAD Mect Suite” area.

Update for pc development system:

- **img_sysupdate-3.x.x-ProductName.ext2** and **sysupdate_img_3.x.x.sh**:
 - **By USB:** copy files “img_sysupdate-3.x.x-ProductName.ext2” and “sysupdate_img_3.x.x.sh” on an USB key and restart the device with USB key inserted. For example, with TPAC1007_04_AC device:
 - **img_sysupdate-3.x.x-TPAC1007_04_AC.ext2**
 - **sysupdate_img_3.x.x.sh**
- The update of 3.x version takes about 1.30 minutes. Then switch off device, remove the USB pen drive and restart it. For TPAC1007 and TP1043 models use the micro-USB (type A) → USB adapter.

1.2 Hardware setup

For this tutorial you need:

N.1 24Vdc power supply

N.1 Operator Panel Mect mod. TPAC1007 or TPAC1008

N.1 I/O module with ModBus RTU, Mect mod. MPNC006.

Connect Operator Panel to PC by a Lan cable and switch on the power supply.

See figure 1 and 2 for connection diagram:

Fig1 - Connection for TPAC1008

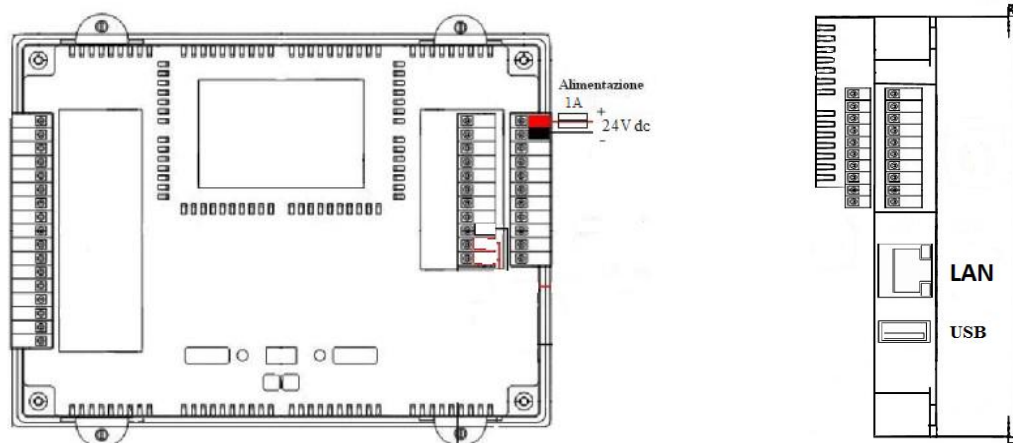
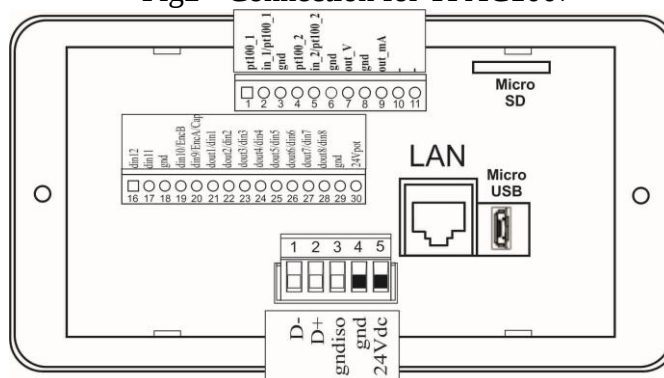


Fig2 - Connection for TPAC1007



2 EXERCISE

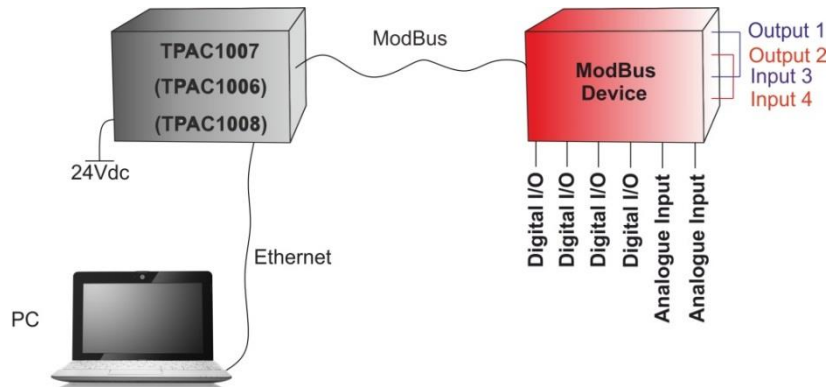
2.1 Description

The example is composed by an operator panel TPAC1007 (or TPAC1008) and an I/O module with ModBus RTU (Mect MPNC006).

The scope is to force 2 digital outputs and read them on 2 digital inputs (connected to digital outputs). One output is forced by PLC application and one by graphic HMI application.

In the graphic HMI we will see how we can display the variable change: with led, with scope, with a recorder.

The I/O module is configured with 2 digital outputs, 6 digital inputs and 2 analogue inputs, output 1 is connected to input 3 and output 2 to input 4.



The first step is the configuration of the operator panel, it is necessary to write which I/O and fieldbus are present in the system (specify also baudrate etc).

In the Mect Suite this is performed modifying **Crosstable** and the **System Editor**.

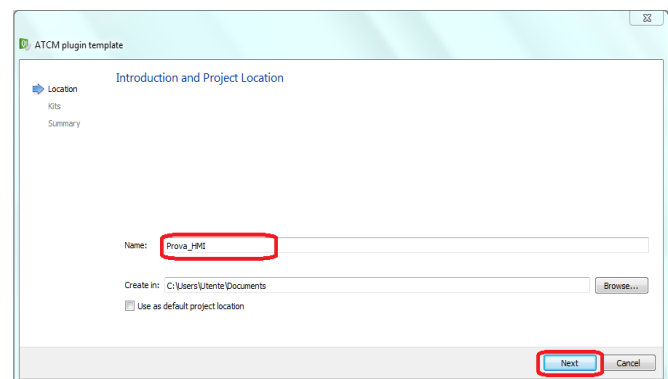
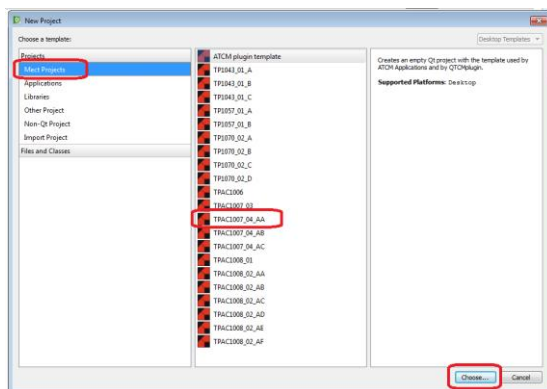
2.2 PROJECT CREATION

Qt Creator is used to write HMI (graphics) application and for configuration file.

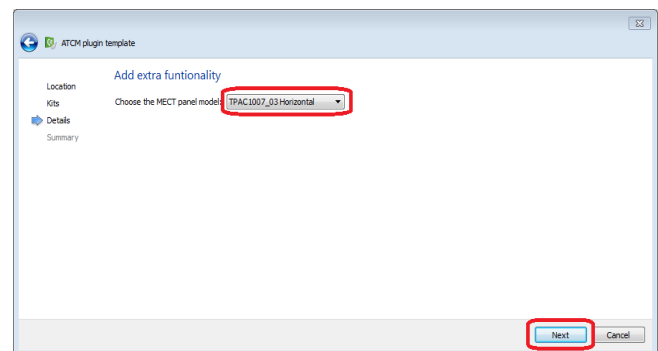
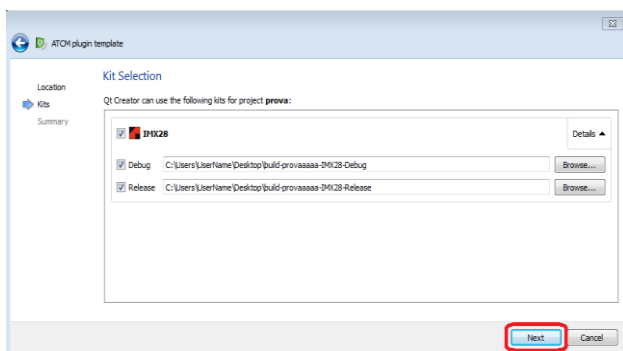
2.2.1 New project

To create a new graphic project is sufficient to open **Qt Creator** and follows few simple steps:

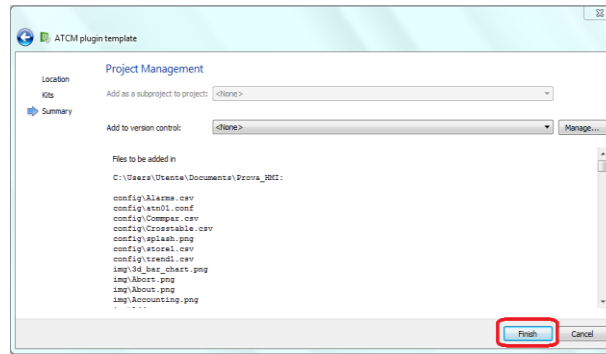
1. Open **Qt Creator**, press on **File** → **New File or Project**.
2. It will appear the window in the figure, choose "Mect Projects", press on "Choose".



3. Insert project name, **without blanks and special characters**, except for the underscore ' _ '.
4. **Warning:** also the path of the project folder must not contain spaces.
5. Select "Next" at the "kit selection".
6. Select product orientation.



7. In the window "Project Management" click on "finish".



2.2.2 Check LAN connection between Operator panel and PC

To access Mect operator panels it is necessary to configure a sub network on the PC.

To do this follow Windows instructions at link: (configure “**Internet Protocol Version 4 (TCP/IPv4)**”)

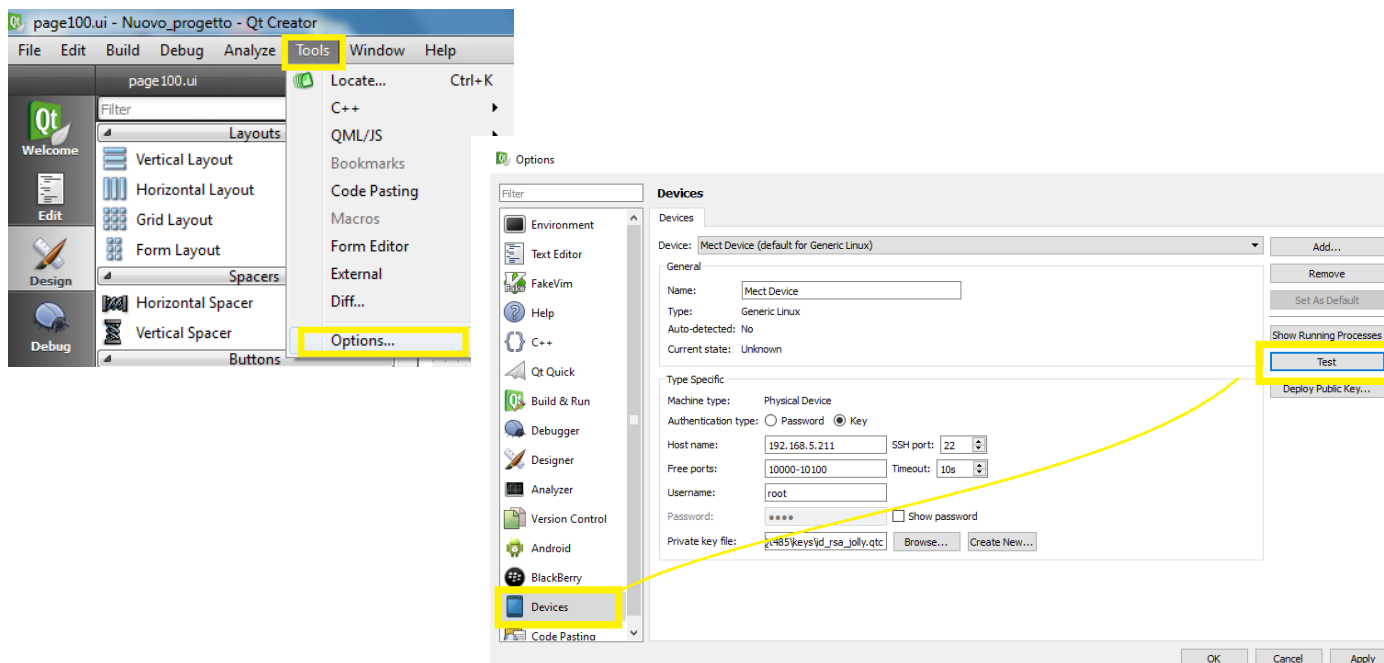
<http://windows.microsoft.com/en-us/windows/change-tcp-ip-settings#1TC=windows-7>

Follow instructions, including step 7 (press “**Advance...**”) and select key “**Add...**” and insert following parameters:

- IP address: 192.168.5.100 (must be different from operator panel one)
- Subnet mask: 255.255.255.0

Power the operator panel and connect through the LAN cable to your PC .

To make the connection test between the PC and the operator panel you must select Tools → Options.



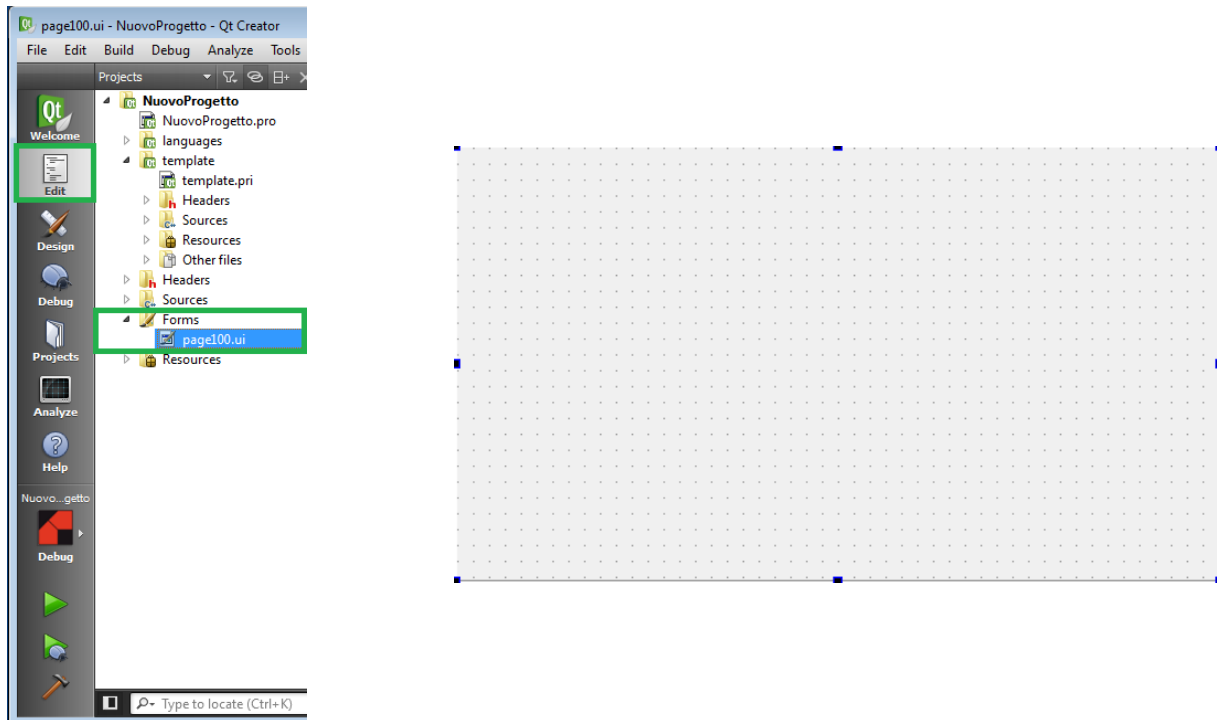
Select “**Devices**” and press “**Test**” for verification.

The operator panel are with default IP addresses: **192.168.5.211**

Passphrase: **Vpn:Update1207**

2.2.3 Modify “splash” screen

In an empty project it is necessary to add the desired pages and to fill them with the graphic elements to use. The new project has the initial page *page100.ui*.



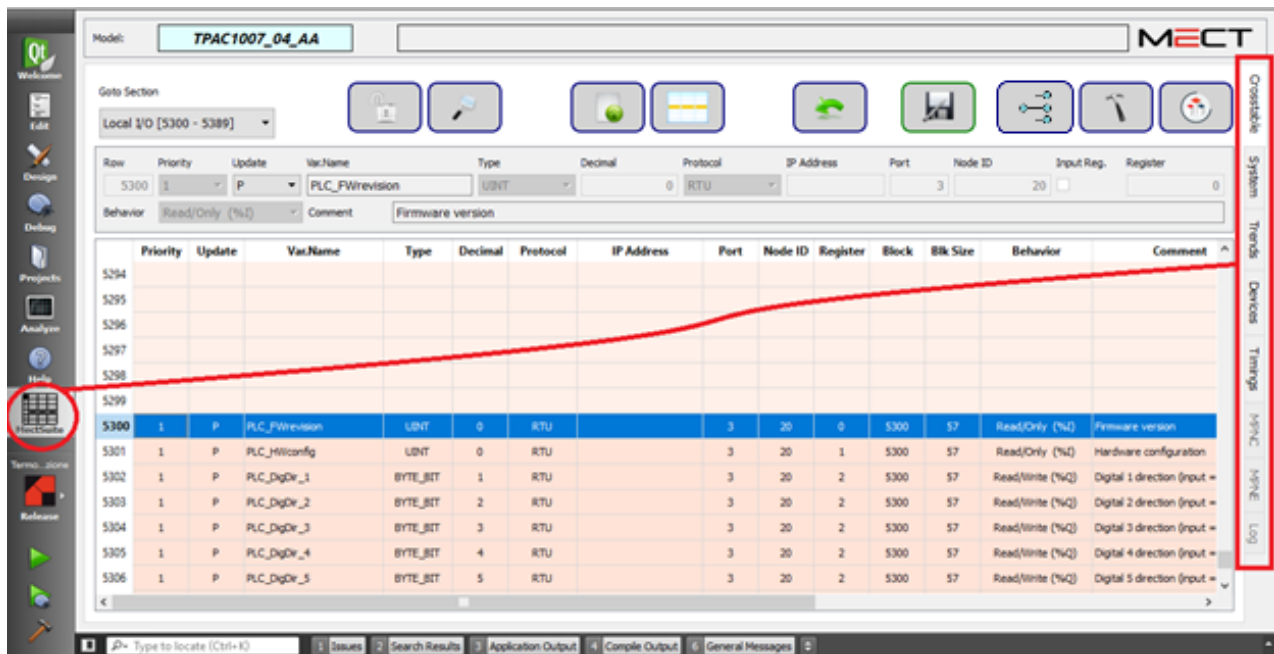
To change the splash screen it is sufficient overwrite the “*splash.png*” file in the “*config*” folder of the project. Splash image must be in the png format, and of the same resolution of the used product:

- TPAC1007/TP1043 => 480x272;
- TPAC1008/TP1070 => 800x480.

2.2.4 Variables – Device management

It is possible to modify some of the project settings:

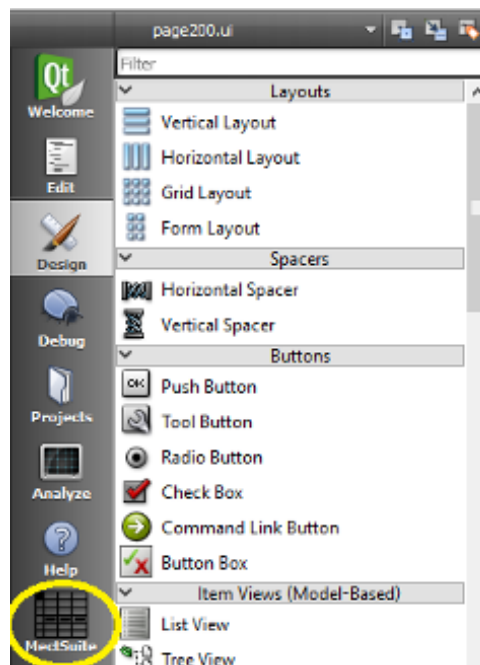
1. **Crosstable Editor** → creation and management of the **variables**;
2. **System Editor** → preliminary setup of the operator panel and setup of communication parameters;
3. **Trends Editor** → configuration of stored variables to show graphics in the time domain;
4. **Devices Editor** → the window provides a topological view of the variables used in the Crosstable;
5. **Timing Editor** → gives a general idea about the time necessary to allow a correct exchange of values between variables with RTU protocol.
6. **MPNC Editor** → configuration of remote I/O nodes of the MPNC family.
7. **MPNE Editor** → configuration of remote I/O nodes of the MPNE family.
8. **Log** → gives a general idea about the stored variables.



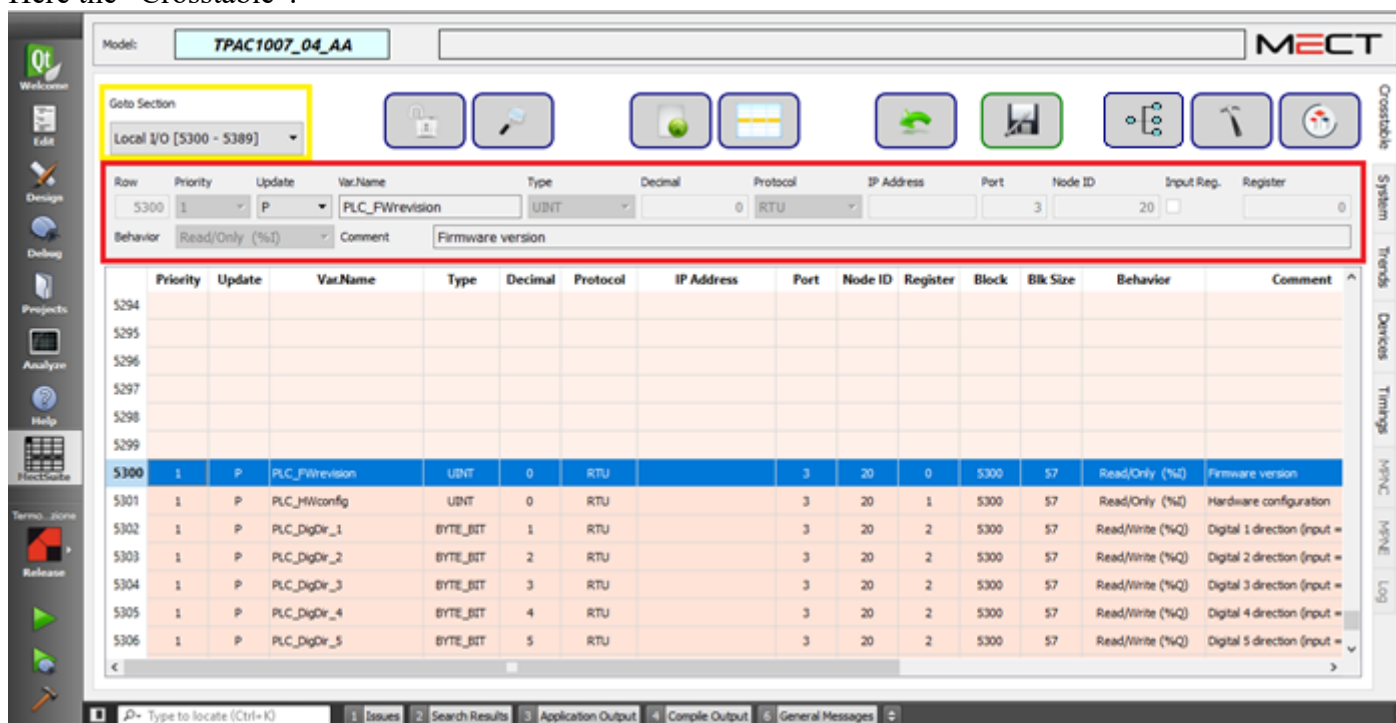
2.2.4.1 Crosstable creation

After project creation in Qt, first of all define variables used in the application. To do this use “Crosstable” that generates a **global variables** table that is shared between HMI and PLC.

To open “Crosstable” see the following figure:



Here the “Crosstable”:



In the “**Goto Section**” section (see yellow frame) it is possible to choose the variables type to see in the table below.

The red selection is where the user can setup to create and modify its project variables.

Moving mouse over each field a brief description appears.

To insert or modify a variable it is necessary:

1. Select table row;
2. Fill / modify existing fields in the red area;
3. Select “**Enter**” key or any row in the table to allow the insertion of the created or modified variable.

NOTE: inside table it is possible to move with the arrow keys as in a spreadsheet program.

Description of variables in the Crosstable of this exercise (not permanent variables).

WARNING: variables on RTU bus are used to manage I/O on Modbus interface (MPNC006). On product manual there is the “Maps of the Modbus RTU registers” chapter where there is the list of available registers to use desired I/O.

In the MPNE and MPNC models, you can import variables directly into the Crosstable (select an empty row and right-click).

Default node address is 1 (to write in the Crosstable “Node ID” field).

Compile the crosstable as follows:

- **Con_TPLC_In_An_1** configures analogue input 1 of the Modbus node.
- **Con_TPLC_In_An_2** configures analogue input 2 of the Modbus node.
- **TPLC_IN_AN_1** analogue input 1 value of the Modbus node.
- **TPLC_IN_AN_2** analogue input 2 value of the Modbus node. Variable historicised (S Update).
- **Con_TPLC_Ou_Di_1** configures digital output 1 of the Modbus node.
- **Con_TPLC_Ou_Di_2** configures digital output 2 of the Modbus node.
- **Con_TPLC_In_Di_3** configures digital input 3 of the Modbus node.
- **Con_TPLC_In_Di_4** configures digital input 4 of the Modbus node.
- **TPLC_IN_3** digital input 3 value of the Modbus node. Variable historicised (F Update).
- **TPLC_IN_4** digital input 4 value of the Modbus node.
- **TPLC_OUT_1** digital output 1 value of the Modbus node.
- **TPLC_OUT_2** digital output 2 value of the Modbus node.

Row	Priority	Update	Var.Name	Type	Decimal	Protocol	IP Address	Port	Node ID	Register	Block	Blk Size	Comment	Behavior	
193	1	P	Con_TPLC_In_An_1	INT	0	RTU		0	1	40	193	12	Configurazione In. Analogico 1 Nodo	Read/Write (%Q)	
Comment: Configurazione In. Analogico 1 Nodo Behavior: Read/Write (%Q)															
Priority	Update	Var.Name	Type	Decimal	Protocol	IP Address	Port	Node ID	Register	Block	Blk Size	Comment	Behavior		
193	1	P	Con_TPLC_In_An_1	INT	0	RTU		0	1	40	193	12	Configurazione In. Analogico 1 Nodo	Read/Write (%Q)	
194	1	P	Con_TPLC_In_An_2	INT	0	RTU		0	1	41	193	12	Configurazione In. Analogico 2 Nodo	Read/Write (%Q)	
195	1	P	TPLC_IN_AN_1	INT	1	RTU		0	1	42	193	12	Valore In. Analogico 1 Nodo	Read/Only (%I)	
196	1	S	TPLC_IN_AN_2	INT	1	RTU		0	1	43	193	12	Valore In. Analogico 2 Nodo	Read/Only (%I)	
197	1	P	Con_TPLC_Ou_Di_1	BYTE_BIT	1	RTU		0	1	44	193	12	Configurazione Out Digitale 1 Nodo	Read/Write (%Q)	
198	1	P	Con_TPLC_Ou_Di_2	BYTE_BIT	2	RTU		0	1	44	193	12	Configurazione Out Digitale 2 Nodo	Read/Write (%Q)	
199	1	P	Con_TPLC_In_Di_3	BYTE_BIT	3	RTU		0	1	44	193	12	Configurazione In. Digitale 3 Nodo	Read/Write (%Q)	
200	1	P	Con_TPLC_In_Di_4	BYTE_BIT	4	RTU		0	1	44	193	12	Configurazione In. Digitale 4 Nodo	Read/Write (%Q)	
201	1	F	TPLC_IN_3	BYTE_BIT	3	RTU		0	1	45	193	12	Valore In. Digitale 3 Nodo	Read/Only (%I)	
202	1	P	TPLC_IN_4	BYTE_BIT	4	RTU		0	1	45	193	12	Valore In. Digitale 4 Nodo	Read/Only (%I)	
203	1	H	TPLC_OUT_1	BYTE_BIT	1	RTU		0	1	46	193	12	Valore Out Digitale 1 Nodo	Read/Write (%Q)	
204	1	H	TPLC_OUT_2	BYTE_BIT	2	RTU		0	1	46	193	12	Valore Out Digitale 2 Nodo	Read/Write (%Q)	

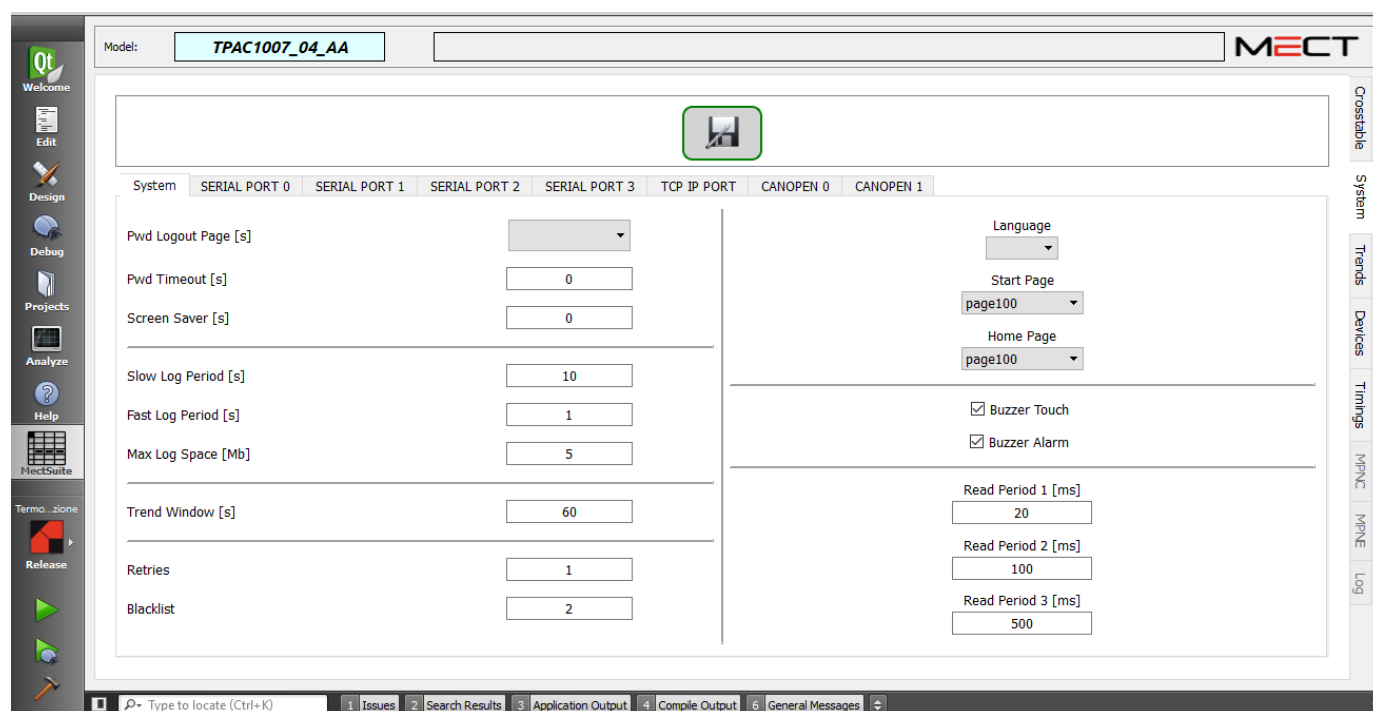
WARNING: in the “**Register**” filed, highlighted in green, there are the numbers of registers of Modbus node according to I/O to use.

NOTE: from 5000 until 5299 the line Crosstable are the system's diagnostic variables while from the line 5300 are the PLC variables (input/output in the TPAC etc ..). See PLC chapter dedicated.

2.2.4.2 System configuration

With *System Editor* it is possible to change some configurations of the graphical interface and the communication parameters between panel and remote I/O.

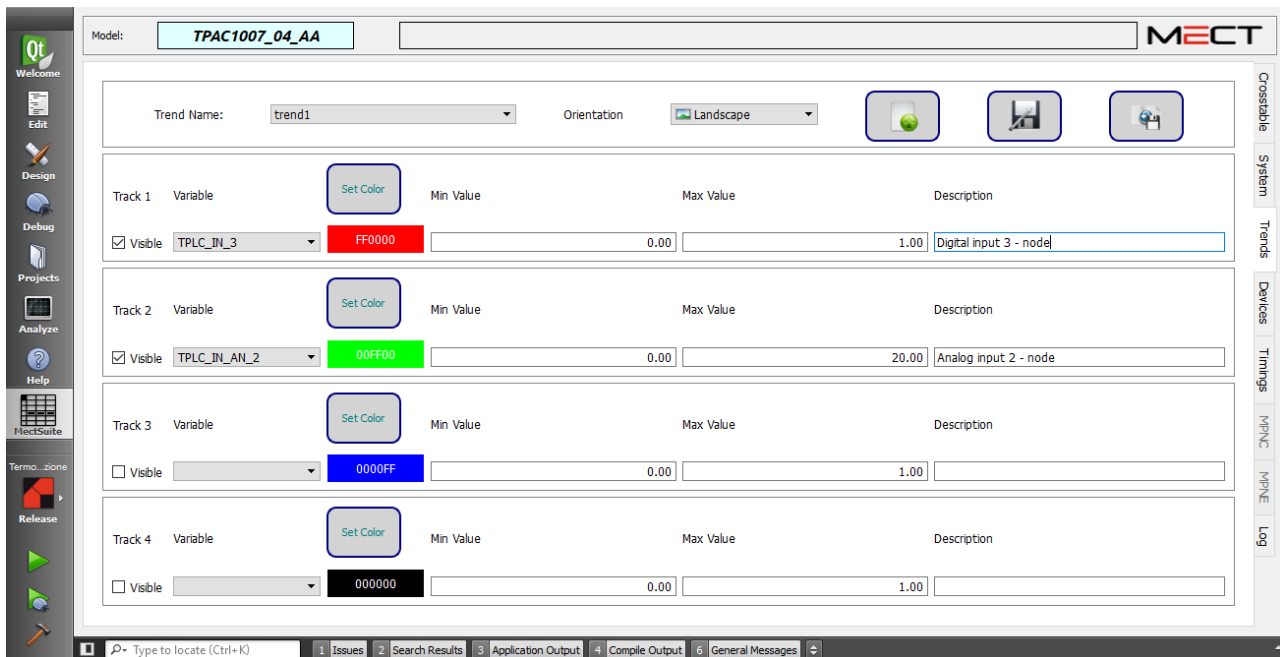
In the example a Modbus RTU protocol has been configured.



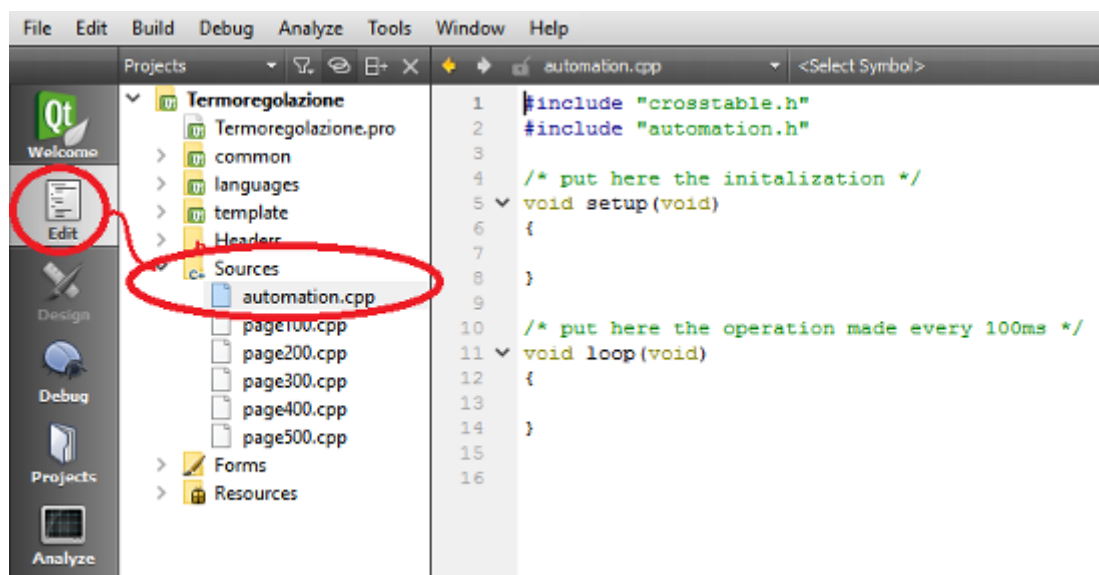
NOTE: in Tab “**Serial Port 0**” it is possible to modify Modbus communication parameters. (**SERIAL PORT 0** corresponds to port number in the Crosstable “**Port**” field).

2.2.4.3 Trends configuration

Trends are historicised graphics showing the relevant variables.
Configure as in the following figure:



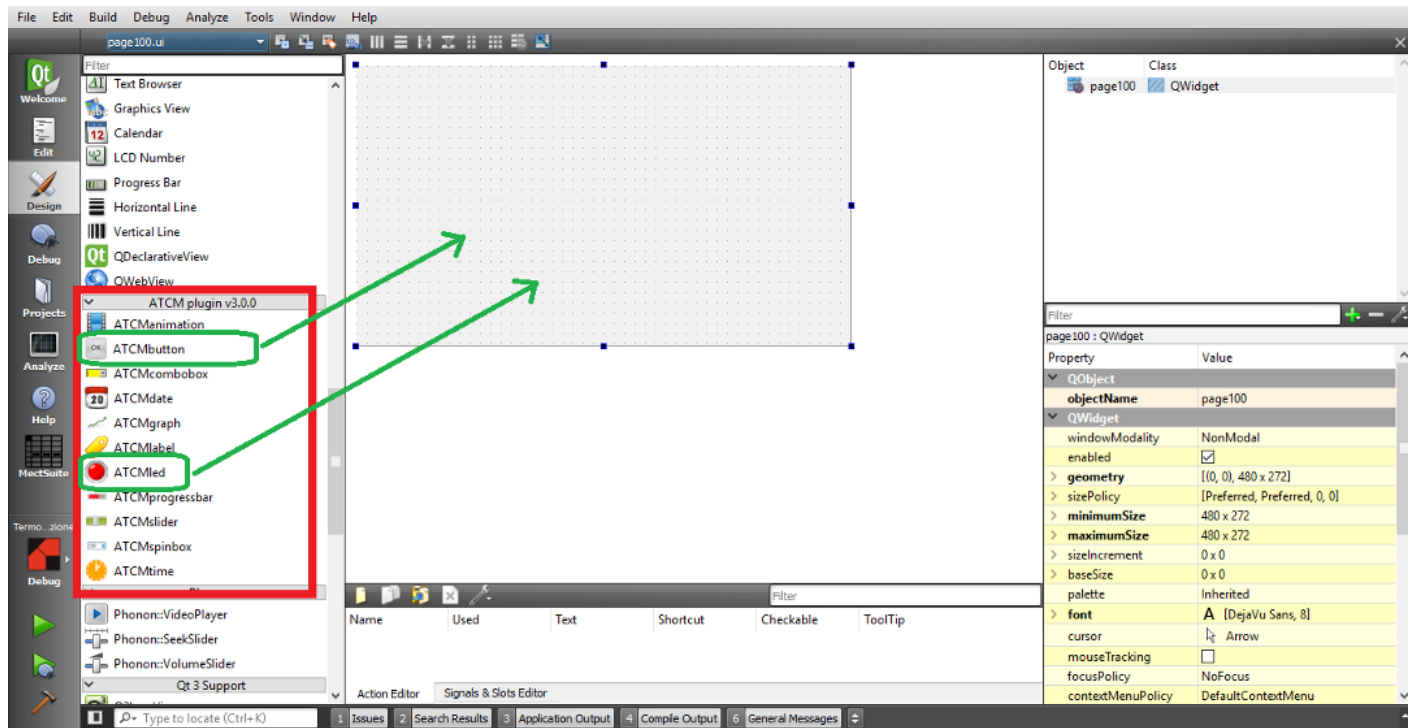
To enable the storage of the history variables insert the **logStart();** function, for example inside the “setup(void)” function in the “automation.cpp” (Edit → Sources) page.



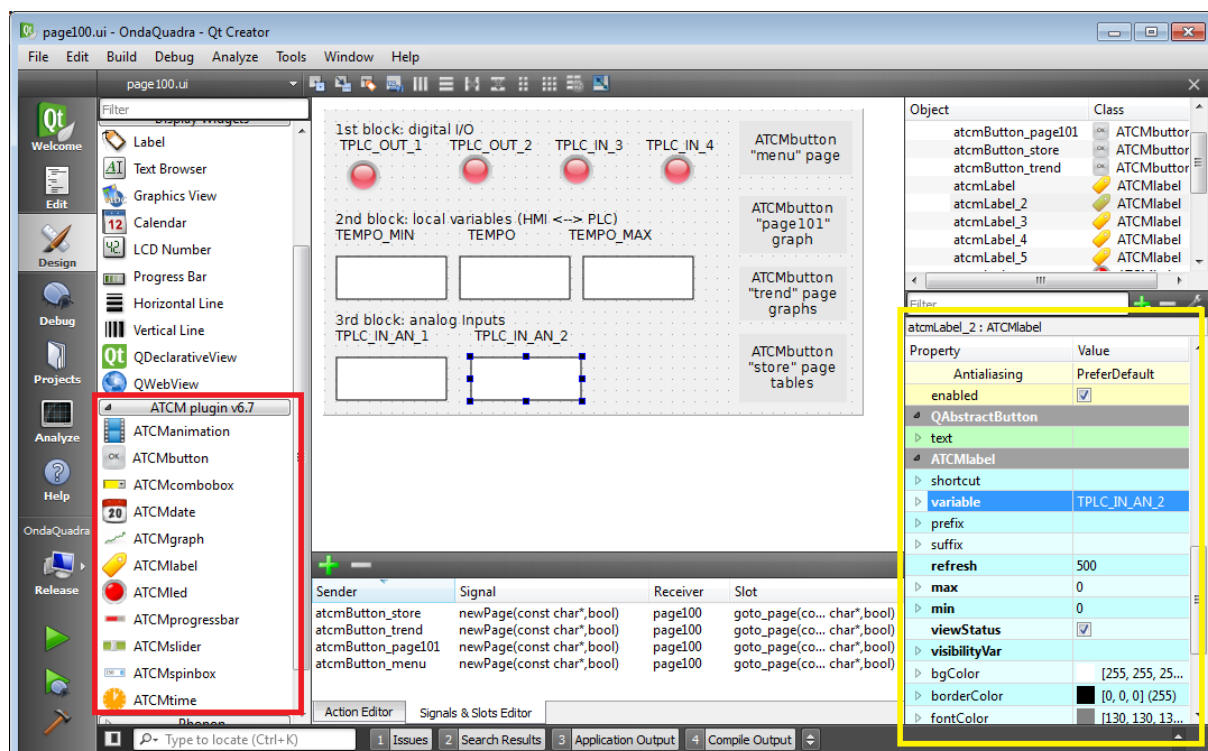
2.2.5 Add graphics

To build pages using in the best way the graphic instruments and minimize the code to write, it is necessary to use the blocks of section “**ATCM plugin vx.y**”, red rectangle.

It is sufficient to drag the elements on the new page.



By selected object it is possible to modify the object property in the right side of the page, see yellow rectangle

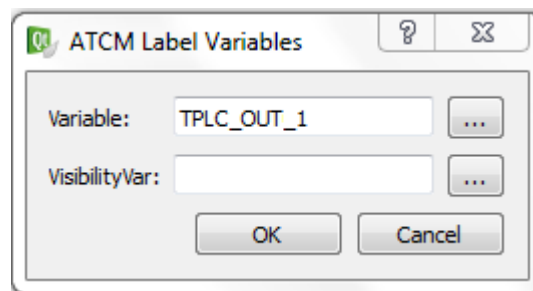


2.2.6 Variables linking

After completion of the graphical elements, it is necessary to link them with the application. In our case we want to link:

- TPLC_OUT_1 → led
- TPLC_OUT_2 → led
- TPLC_IN_3 → led
- TPLC_IN_4 → led
- TPLC_IN_AN_1 → label
- TPLC_IN_AN_2 → label
- PLC_time → label
- PLC_timeMin → label
- PLC_timeMax → label
-

To link it is necessary a double click on the relevant object and select the desired variable, see figure:

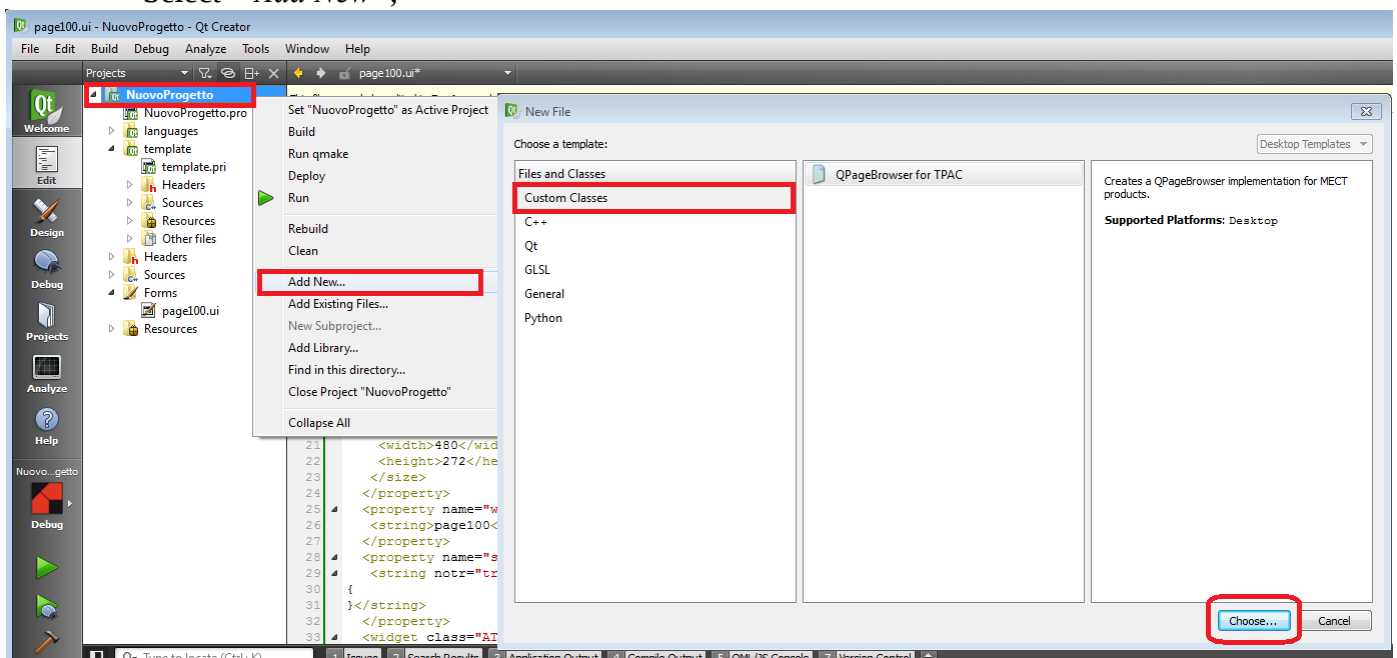


TPLC_OUT_1 e TPLC_OUT_2, has been defined as **H type variables**, that is they are read only if called by an HMI page, in order to **lower the data exchange on modbus bus**, avoiding reading variables not needed at that moment.

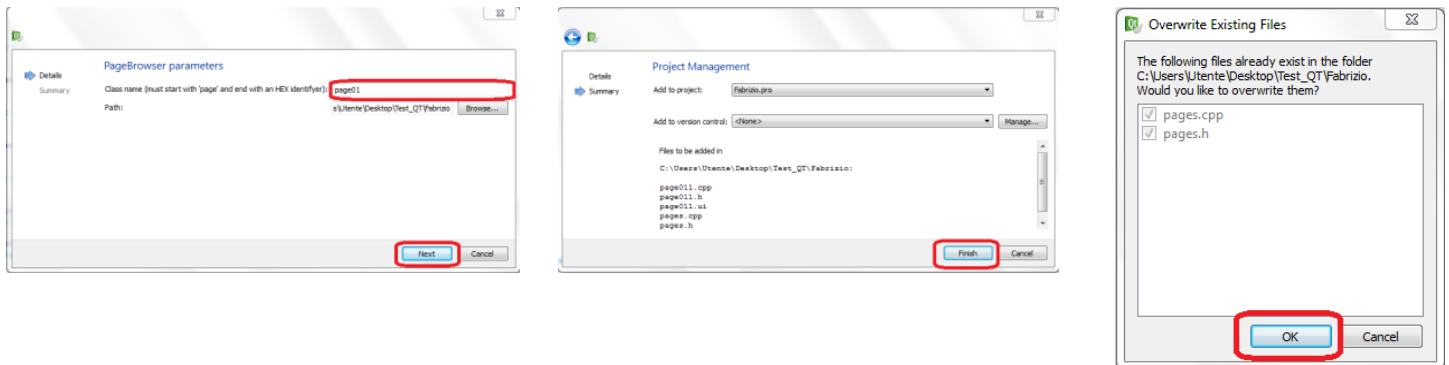
See “**MectSuite Tutorial ENG** → HMI → Crosstable Editor → Description of Crosstable fields → H type Update” to examine in depth the H type variables use.

2.2.7 Add a new page

- Switch in “*Edit*” mode (left column);
- Select the project name (on top) with the right mouse key;
- Select “*Add New*”;

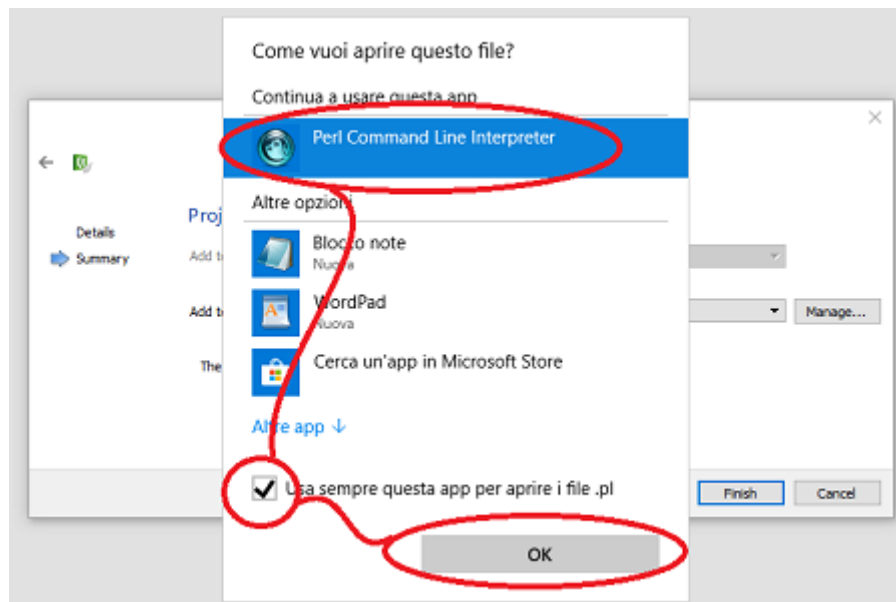


- Press “Choose”;
- Choose the page name and press “Next”;
- **Warning:** the page must have name **page<page number>** (page101);
- In the window “Project Management” click on “finish”;
- Confirm overwrite file “pages.cpp” and “pages.h”.



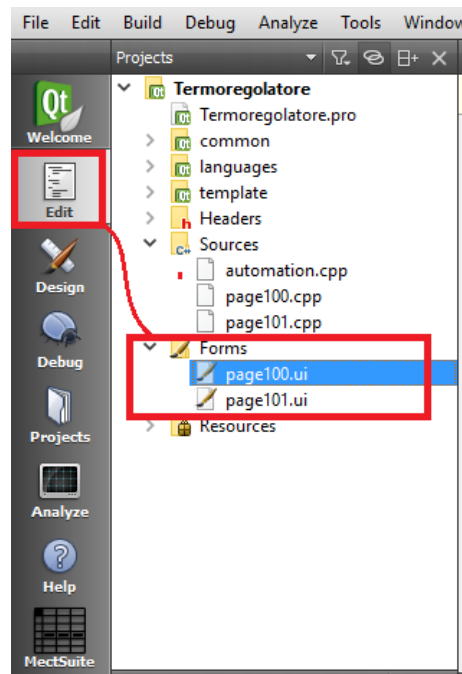
From page 2, you must confirm (press “yes to all”) to the request to reload file “pages.cpp” and “pages.h”. To open a page double click on file corresponding in the “Forms” folder of the project..

Warning: Win10 might bring up this window. Always use the app: Perl Command Line Interpreter. Start from the beginning to create the page.



In our example we create 6 pages, in particular:

- **page100:** “synoptic of I/Os” 4 leds (digital I/Os), 5 labels (2 analogue inputs and 3 for PLC) and 4 buttons for page change.
- **page101:** “oscilloscope” 1 graphic from PLC and 1 button to change page.
- **trend:** “trend graph” uses a prebuild page to show trend of stored variables. This page is already present in the library, to recall it write “trend1” in the “pageName” property of the ATCMbutton.
- **store:** “history table” like trends, but with tables. This page is already present in the library, to recall it write “store” in the “pageName” property of the ATCMbutton.
- **alarms:** alarm signaling of the defined variables. This page is present in the library.
- **recipe_select:** recipes page (defaults settings of some variables). This page is already present in the library, to recall it write “recipe_select” in the “pageName” property of the ATCMbutton.

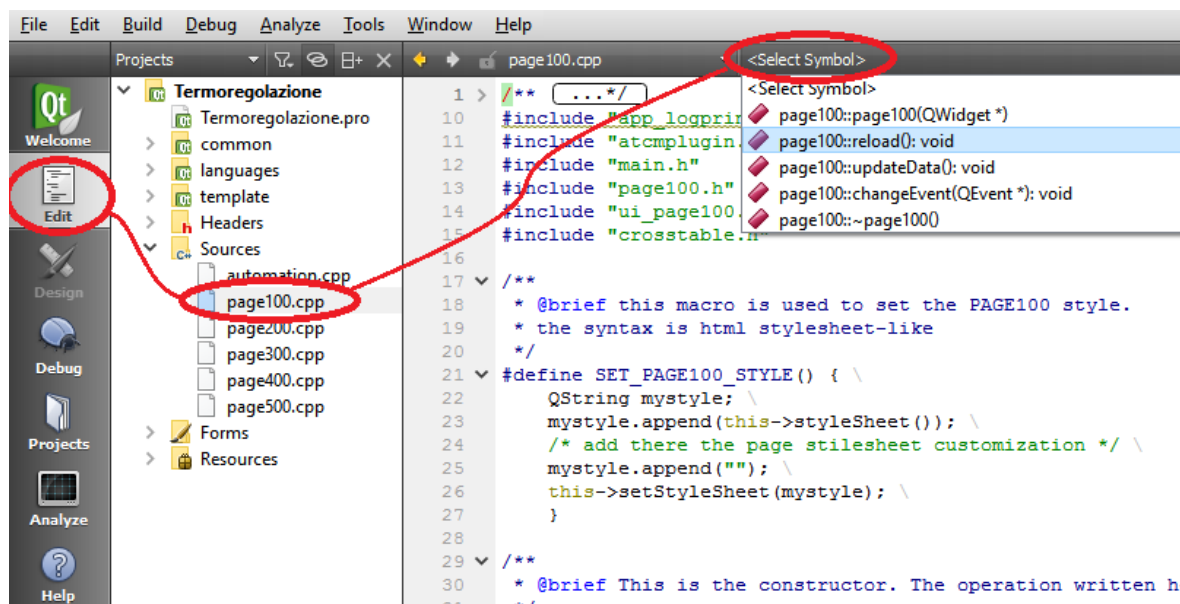


2.2.8 Insert source code

Although it is not necessary to write C++ code, it is possible to do it comunque possibile farlo, for example if you want to manage directly the PLC variables in the HMI application (**Warning:** usually this is done in the PLC application).

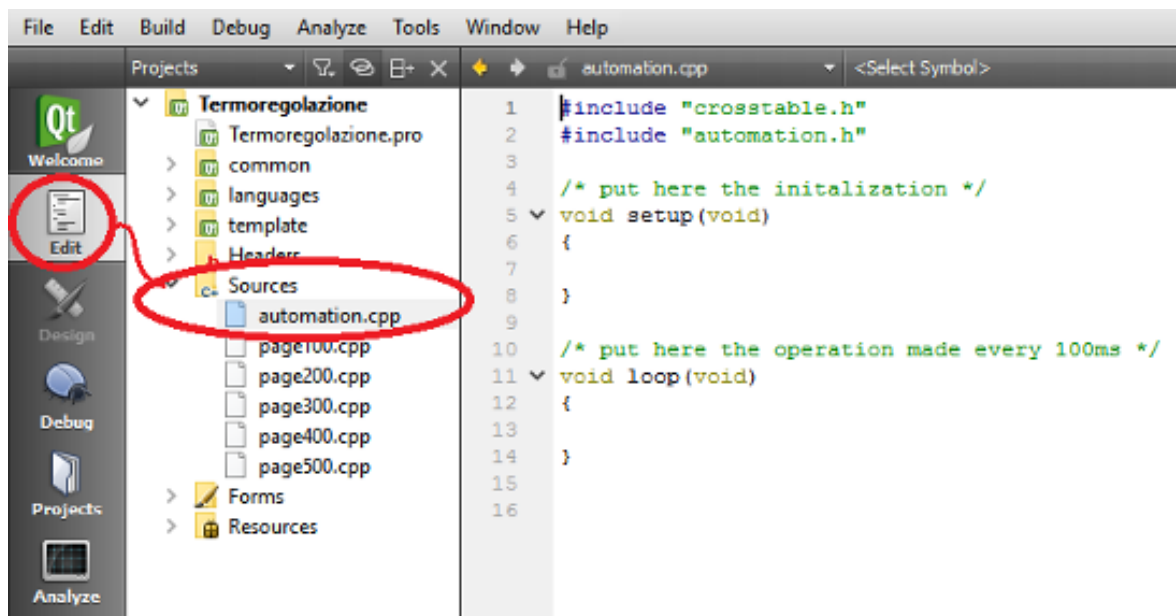
Source code can be linked to each page:

1. in **reload()** method, invoked only when the page is shown on the display;
2. in **updateData()** method, invoked every 500 ms when the page is visible;



Moreover it is possible to write code linked to the application in the file **automation.cpp**:

1. in the **setup()** function, loaded only at the application startup;
2. in the **loop()** function, loaded every 100ms independently of the shown page.



To write a variable in C++ call the function **doWrite_VariableName(value)**, where VariableName is the variable name in the crosstable and value is the value to write. For example to configure as output first digital i/o it is sufficient to write **doWrite_PLC_DigDir_1 (1);**

To assign to the first analogue output the value of the product of first 2 analogue input it is sufficient to write:

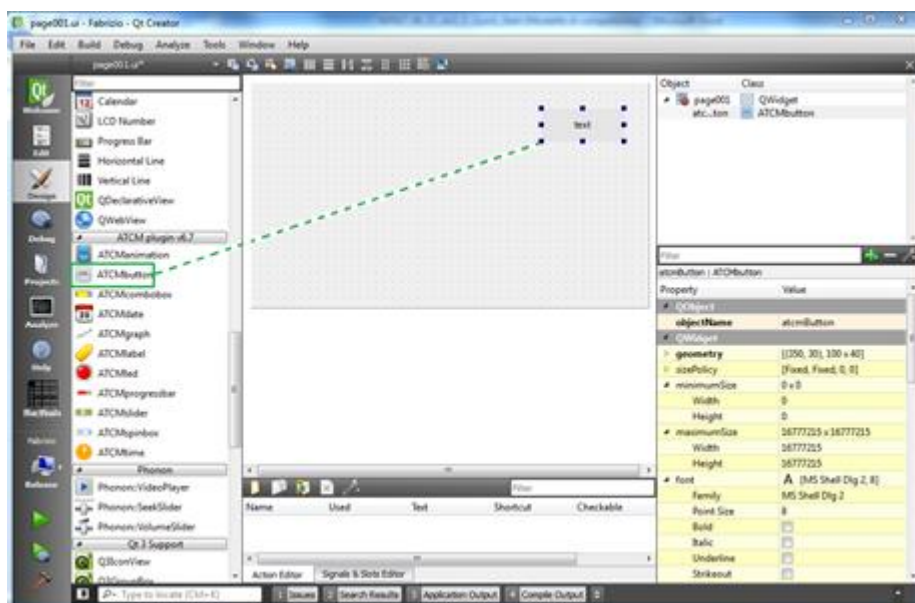
doWrite_PLC_AnOut_1(PLC_AnIn_1 * PLC_AnIn_2);

2.2.9 Activate page change

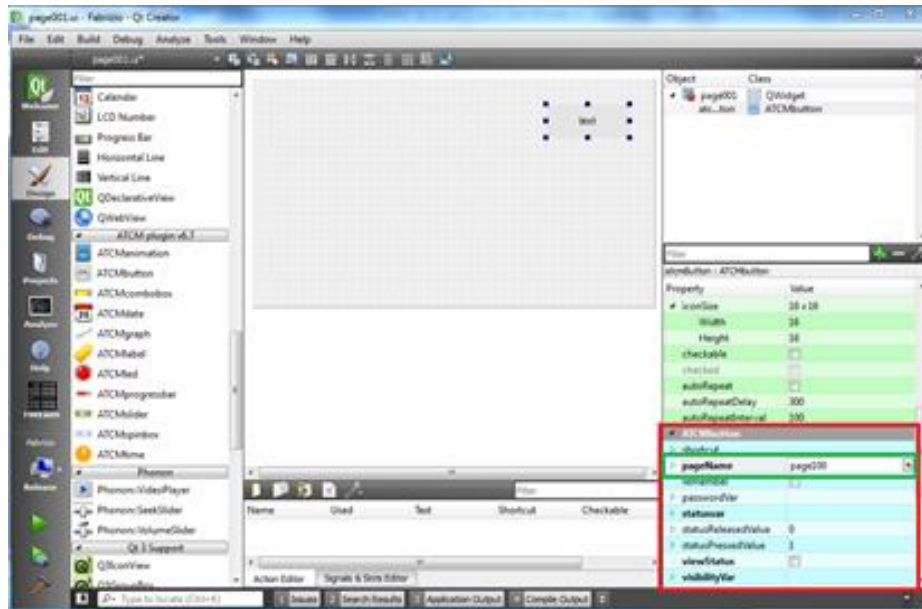
In the page100 there are 4 keys for page change.

To activate a button and **change** the page follow these steps:

1. Drag in the window an ATCMBbutton



2. Select button and put in the field *Property*, ATCMBbutton "**pageName**" the name of the page to go, for example "**page101**".

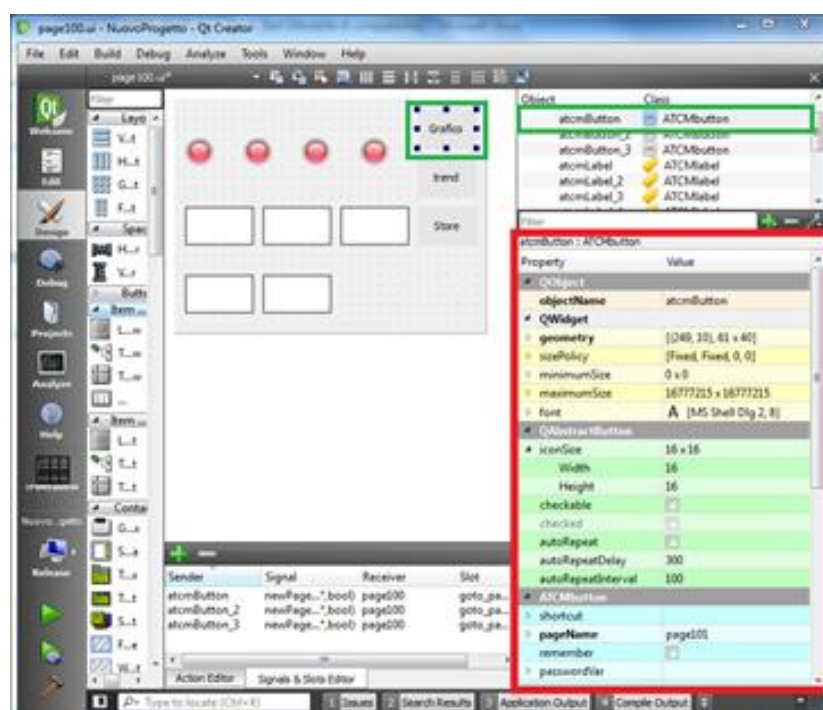


There are some special page names:

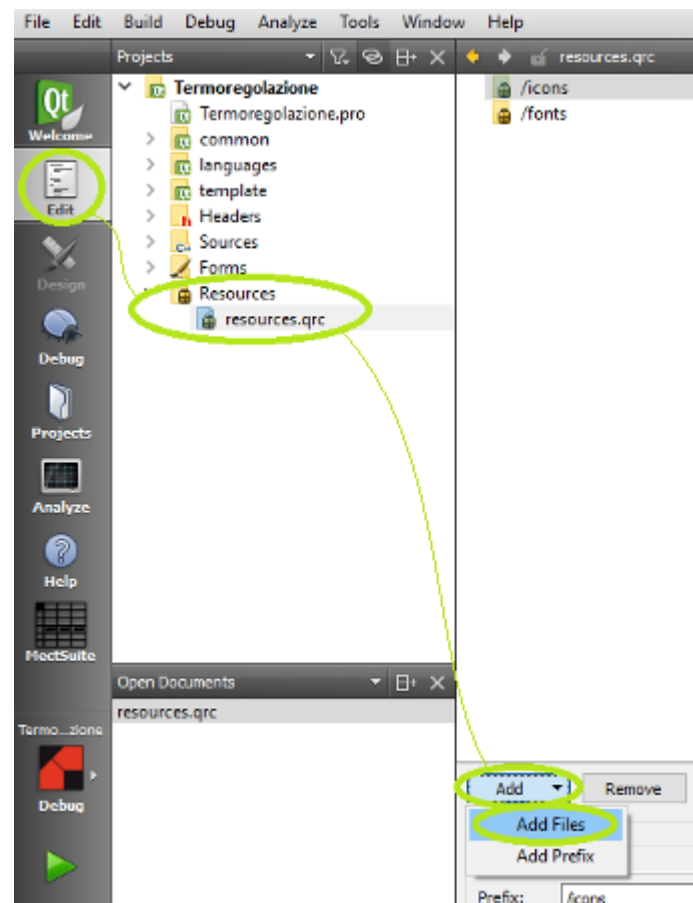
- “HOME”: back to home page defined in system.ini file
- “BACK”: previous page
- “menu”: goes to a general index page
- “trend1”: opens trends page (see below)
- “store”: opens store page (see below)
- “alarms”: opens alarms page (see below)
- “recipe_select”: opens recipe page (see below)
- “time_set”: opens the date and time pre-built page (menu →)
- “display_settings”: opens the brightness monitor page (menu →)
- “info”: opens the information page (menu →)
- “net_conf”: opens the network configuration page (menu →)

2.2.10 Modifying elements

To modify the setup of every single element you must select it and change the “*Property*” field in the right column, see red rectangle.

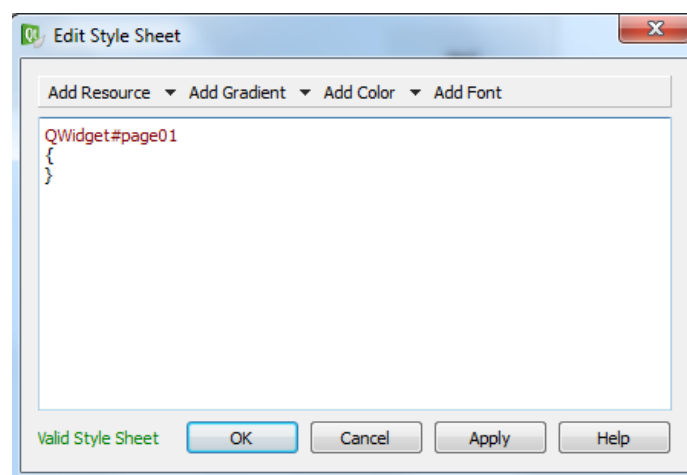


To insert an icon into an ATCMbutton you must select the button and then select the desired image in the *Property: "icon"*. To add its own icon you have to add it to the **"resource.qrc"** resource (Add file) and **save** to make effective the addition of icons.



To change the background of the page, with a color or image, select the page with right click and press "Change styleSheet".

You see the following figure:



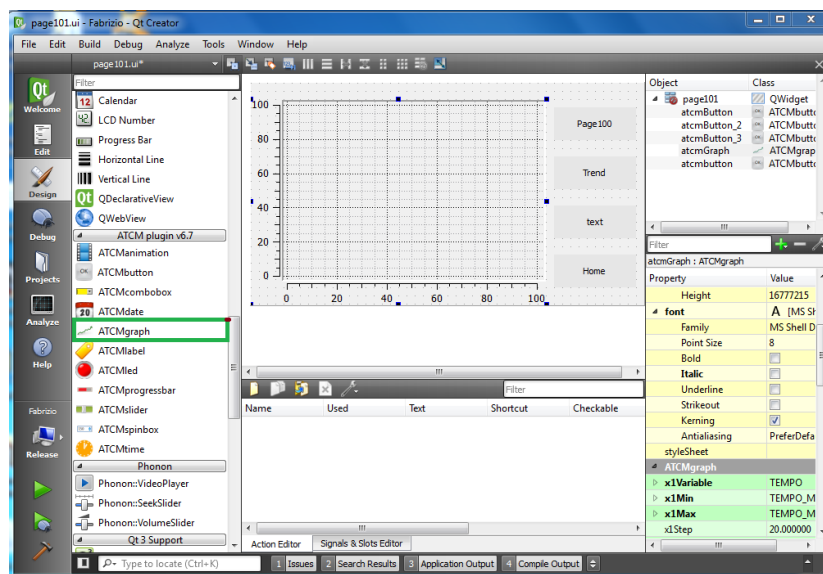
Put the cursor inside curly brackets and select:

- image: "Add Resource → image"
- color: "Add Color → background-color"

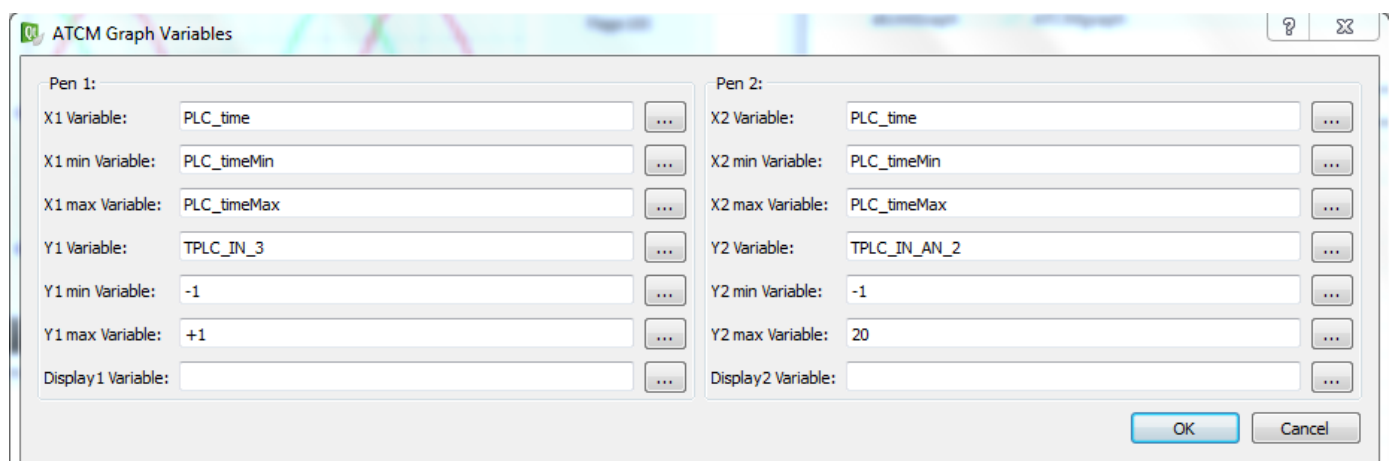
Warning: using the gradient to set the background of a page will slow down the device. It is better to use a defined color or an image.

2.2.11 “Oscilloscope” page

To use scope graph it is necessary to drag into the page the **ATCMgraph**. See the following page:



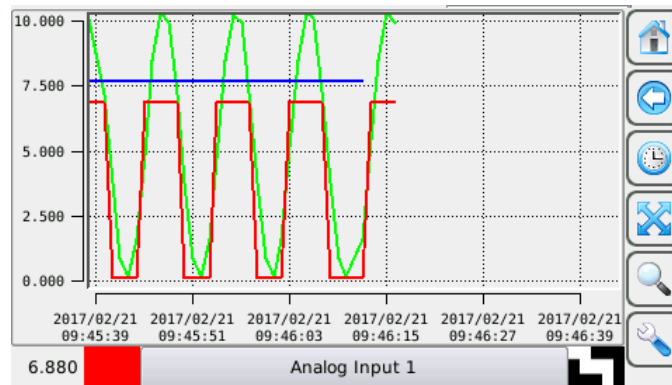
In the graph we want to show in Y axis the digital input “TPLC_IN_3” and the analogue input “TPLC_IN_AN_2”. The X axis is the time of the PLC program. Double click inside the graph, it opens the following page:



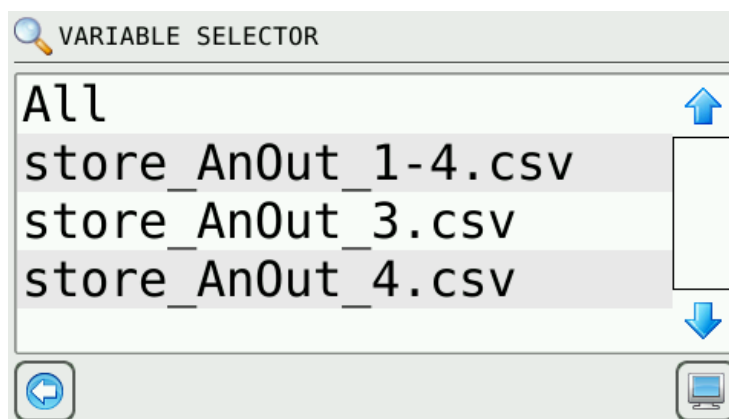
Warning: PLC_tme, PLC_tmeMin, PLC_tmeMax are internal variable of the PLC always available.

2.2.12 “trend” page

If we use a button to change page that opens a trend page It is necessary to write in the field “pageName” of Property, the name of the file of the configuration of the trend (here “trend1”):



2.2.13 “store” page



Upload the new file on the device via web interface (see chapter “[Operator panel web interface](#)”). Optionally you can specify a subset of historicised variables by creating a configuration file (store1.csv) in this way:

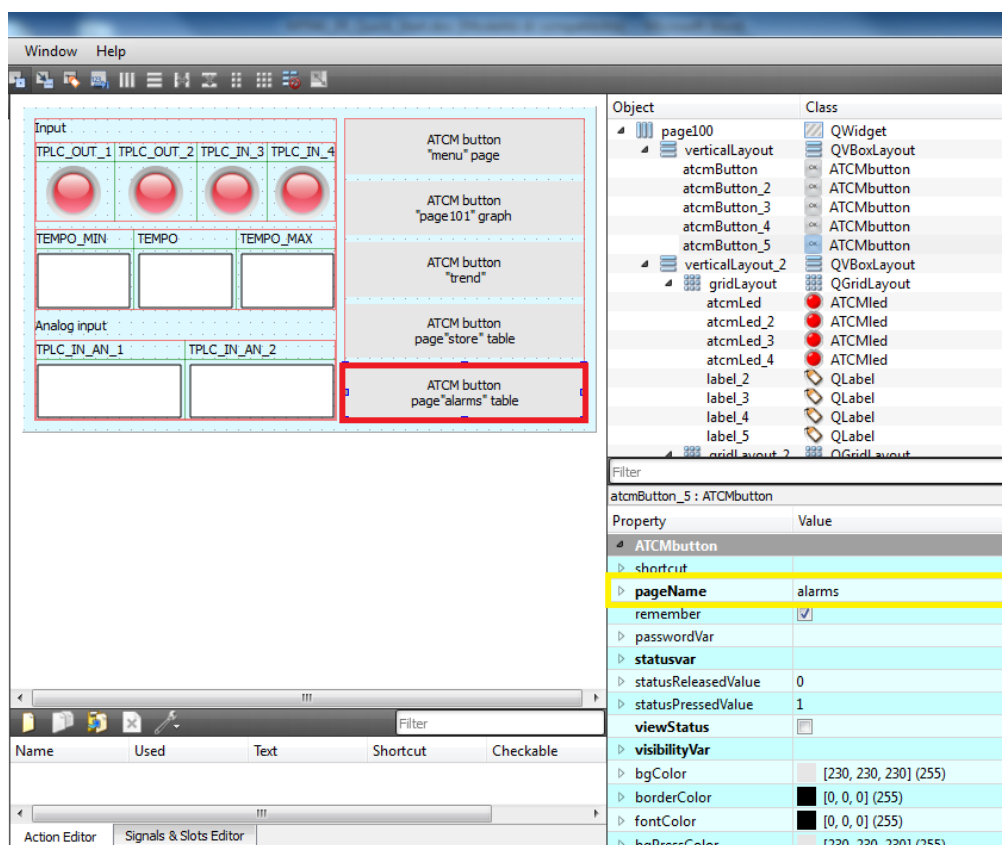
```
date
time
TPLC_IN_3
TPLC_IN_AN_2
```

PAGE store		store1	12/07/2016 - 16:40:23
Filter: [2016/07/12 08:15:00 - 2016/07/12 23:59:59]			
date	time	TPLC_IN_3	
2016/07/12	13:15:40	0	
2016/07/12	13:15:41	0	
2016/07/12	13:15:42	0	
2016/07/12	13:15:43	0	
2016/07/12	13:15:44	0	

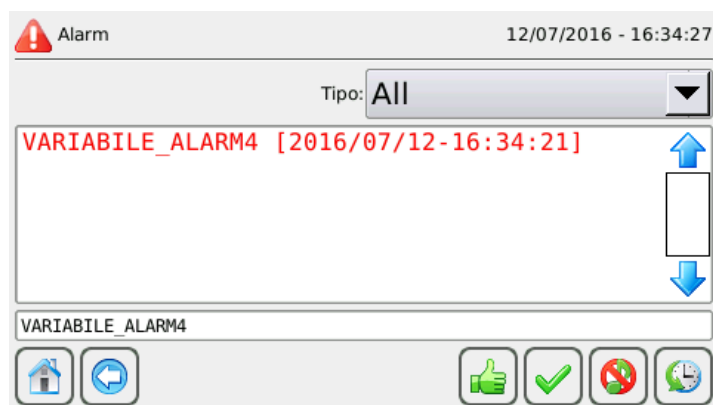
WARNING: remember that to display the history variables use the **logStart()** function.

2.2.14 “alarms” page

If used put a button for page change to “alarms”..



Here the alarms page. from this page it is possible to access the alarms history.



2.2.15 “recipe_select” page

You can use the mechanism of the recipes to write or read or edit a list of preset variables.

Recipes can be made:

- either by the same group of variables, but with different values;
- either by different groups of variables. Those are called “families of recipes”.

Recipes can only be derived from existing recipes on target. New families of recipes can be created by the PC and then uploaded to the target through the web interface described in section “Send data to operator panel”.

The recipes are text file in .csv format. In each line will be specified the variable name and the value to assume.

For example, to create a recipe file that force the output TPLC_OUT_1 to 0 and the output TPLC_OUT_2 to 1 you should write:

```
TPLC_OUT_1; 0; 1  
TPLC_OUT_2; 1; 0
```

It is possible to add all the steps (columns) you want.

Zip the file/folder to upload a recipe or a family of recipes on the device and transfer via web interface (see chapter “[Operator panel web interface](#)”).

Example showing how to create the zip file: it is possible to create one or more folders containing recipes (files with .csv extension): Zip the folders and download them on the device by the transfer web interface.

Displaying / loading / editing recipes has been detailed below into two steps:

1. selection of the recipe to display
2. display / upload / edit the selected recipe

To use the recipe page is necessary to create a button to jump to the page “recipe_select”.

From this window you can browse between the families of recipes and the recipes belonging to them.



Once you have selected the desired recipe, press the button USB (🔌) if you want to save the recipe or family of recipes on the USB or press the recipe to display it.

The recipe are CSV files made of two columns:

1. name of the variable
2. value of variable

RECIPE TPAC1007 03/05/2016 - 12:37:31				
	Tag	1	2	
1	TPLC_OUT_1	0	1	
2	TPLC_OUT_2	1	0	
3				
4				
5				

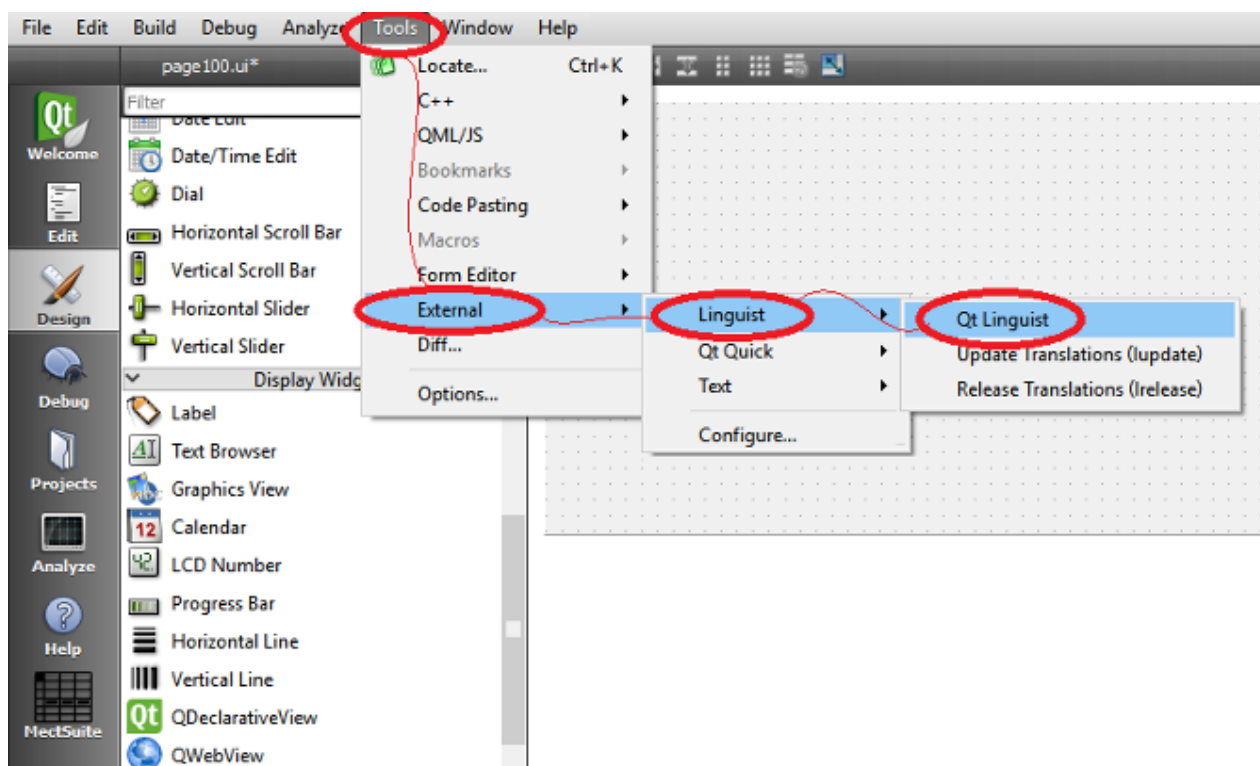
Once you display a recipe, you can do the following actions:

- Change the value of a variable: select the variable.
- Read the update values of the variables of the recipe (on the instruments connected eventually), pressing the Read button (.
- Write on the devices the values displayed in the recipe, pressing the Load button (.
- Save the current recipe: press the Save button () and enter the desired name.

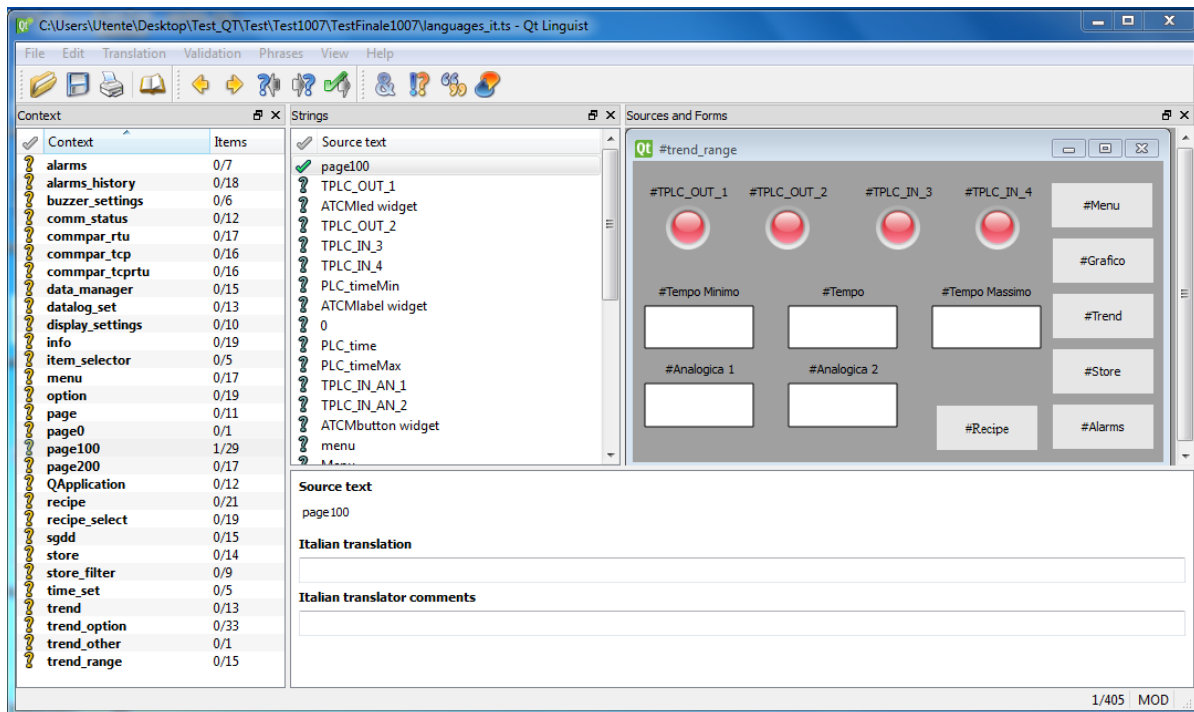
2.3 LANGUAGES TRANSLATION

You can use the tools of **Qt Linguist** in order to have the same user interface with translations in different languages. The template of the project is already configured to be able to fill the files in Italian and English.

To make the translations we use the Qt Linguist tool. See the following image:



From the menu “File → Open” choose the file “languages _ *.ts” of the desired language (the *.ts file is present if you have built your project at least once).

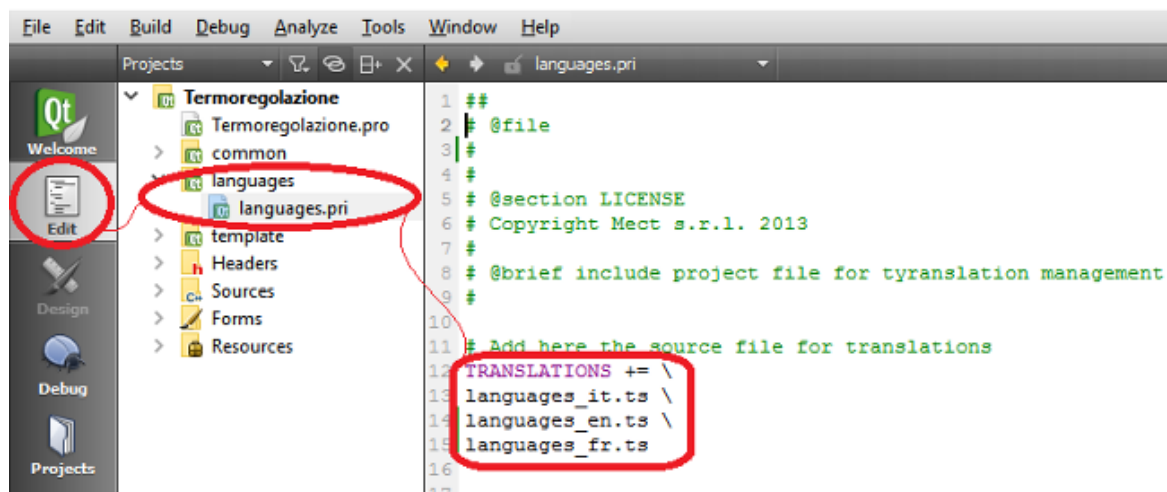


The tool will show you the several text fields used in the interface, edit the translation you need to insert and press Ctrl and Enter to confirm. Once you have finished to fill the translations, save the file and close the Qt Linguist.

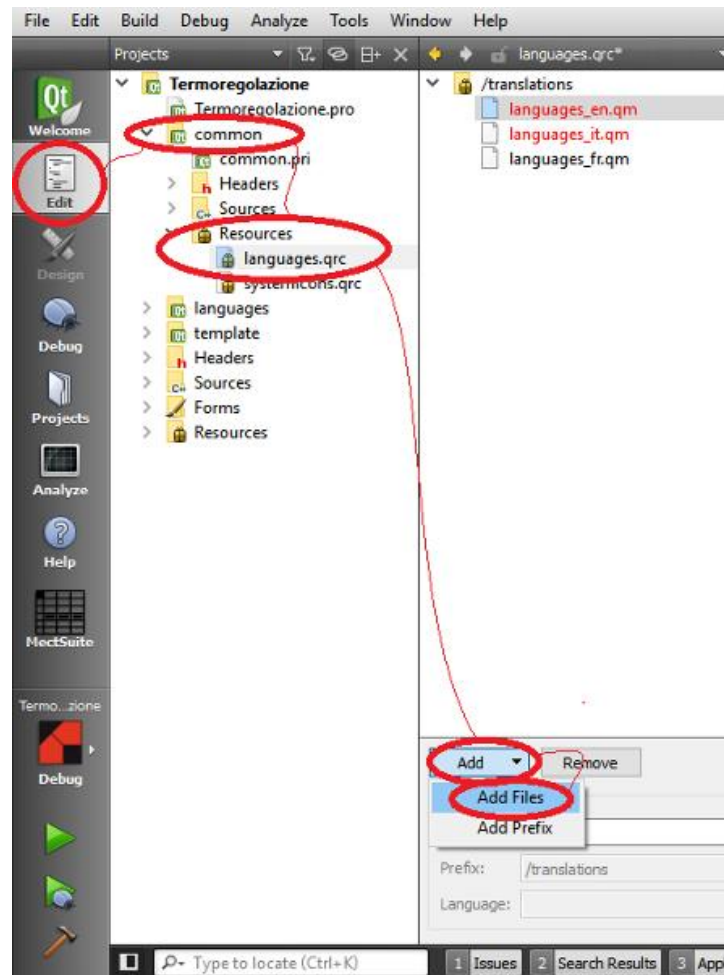
2.3.1 Add a new file of language

To add a new file of language you need to:

- In the Project tree go to “languages”



- Open the file “languages.pri” and add the file of the translation you need. The file must be named “language _ *.ts”, the * symbol must be replaced with the code of the desired language (ex. fr → French).
- Fill to create the translation file on the file system.
- In the Project tree go to “common” → “Resources”.
- Add into the resources file “languages.qrc” the file *.qm born from the previous compiling.



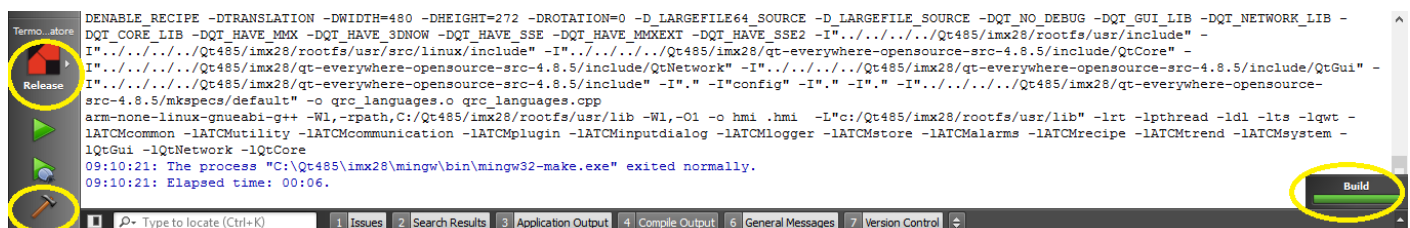
2.4 BUILD & RUN

2.4.1 Executable creation

After the project is complete, it is necessary to build an executable file for the panel architecture and download on the panel itself, as described as follows.

Usually it is sufficient to use the “Release” mode, but it is possible to use the “Debug” mode to have more control on the execution (if you write C++ code).

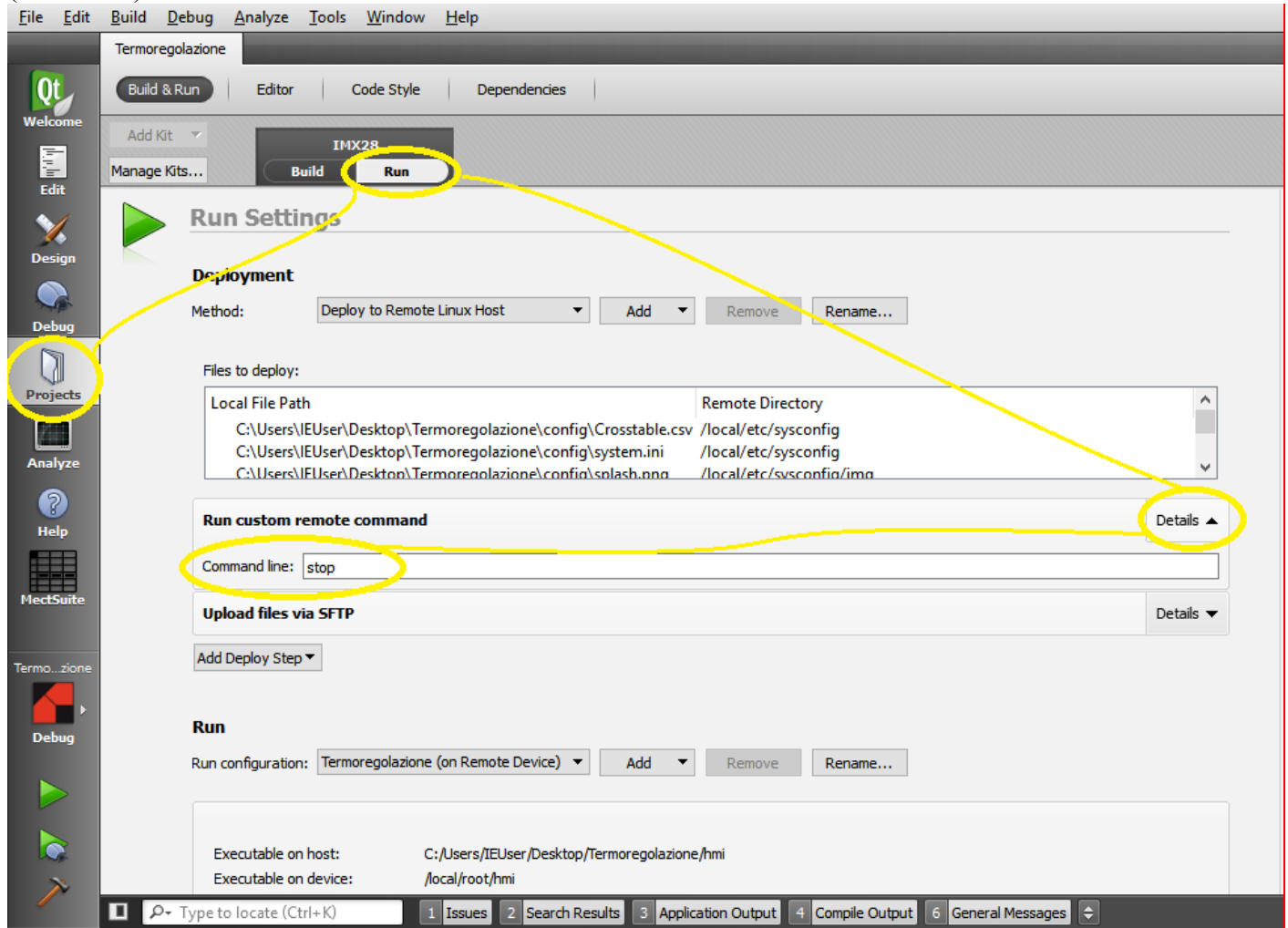
- Configure the program in “Release” mode by acting on symbol in figure, yellow selection.
- Press “Build” icon (“hammer” bottom left).
- If the compilation is successful at the bottom right a green bar appears, as in figure:



2.4.2 Send data to operator panel

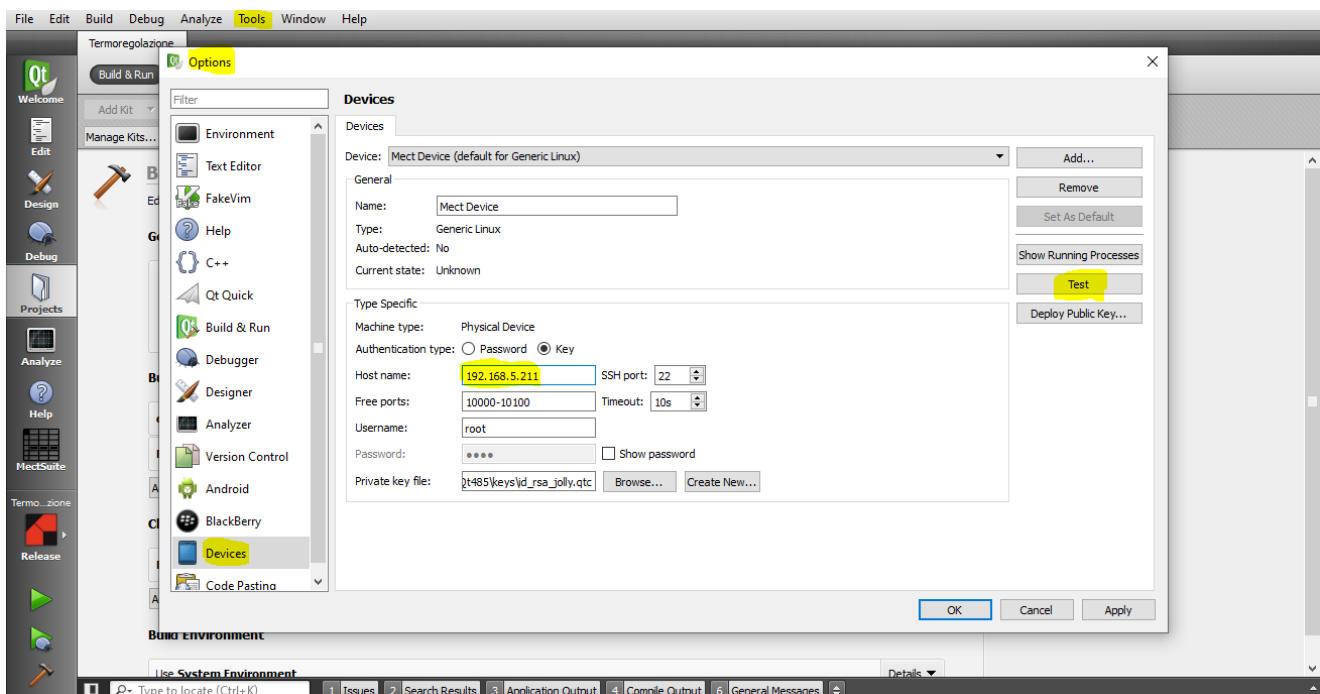
Connect operator panel to PC by a LAN cable.

First time you download the executable file (Deploy) you must select “Projects” → “Run” and write “**stop**” (lowercase) in “Run custom remote command”:



If necessary, it is possible to use a different IP address or a password in the dialog in “Tools → Options → Devices”.

Passphrase: **Vpn:Update1207**



Please check connection between PC and panel. To do it press “Test” key in the right side of the previous figure.

After parameters settings it is possible to send project to the target using the arrow in figure (see yellow circle):



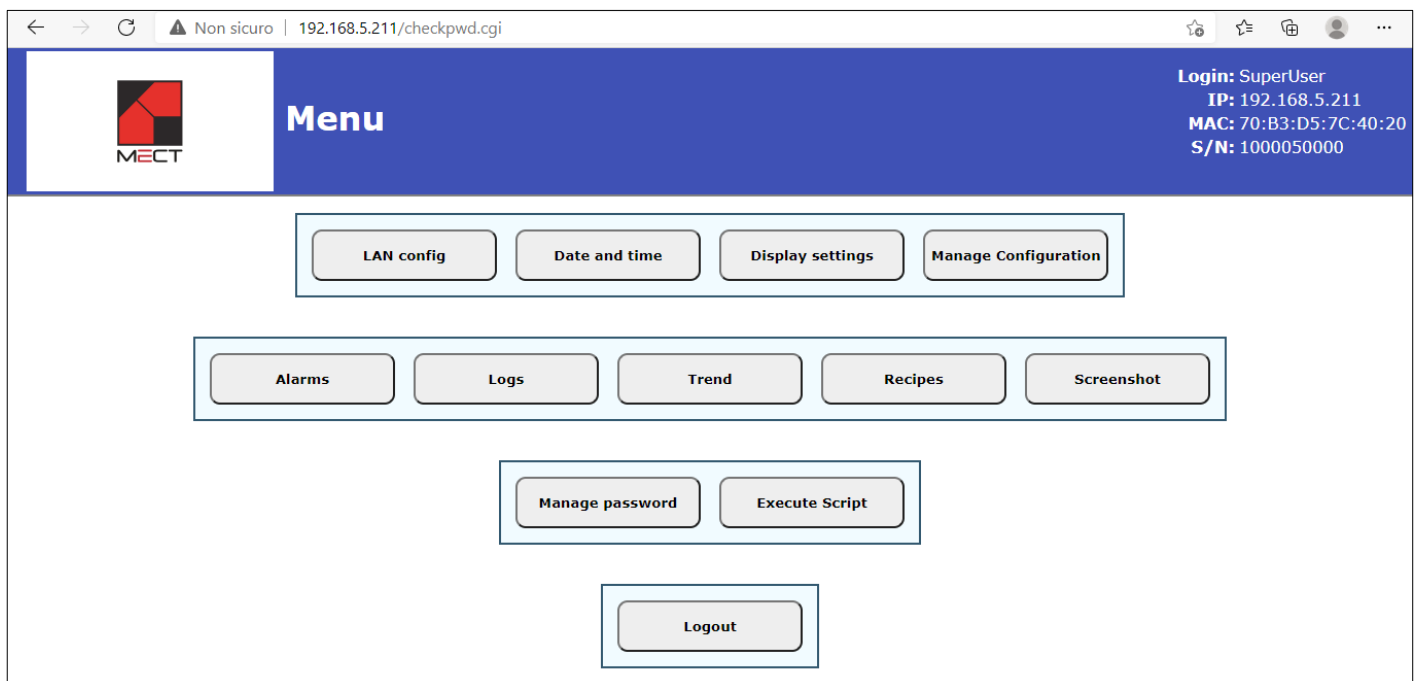
In “[4] Compile Output” you find the copying log (Deploy) and errors.

In “[3] Application Output” you find the application messages.

2.4.3 Operator panel web interface

All panels also a web interface (http) through which it is possible, with a Browser , do some maintenance tasks.

The default address is **192.168.5.211** and the password for the login is **222222**.

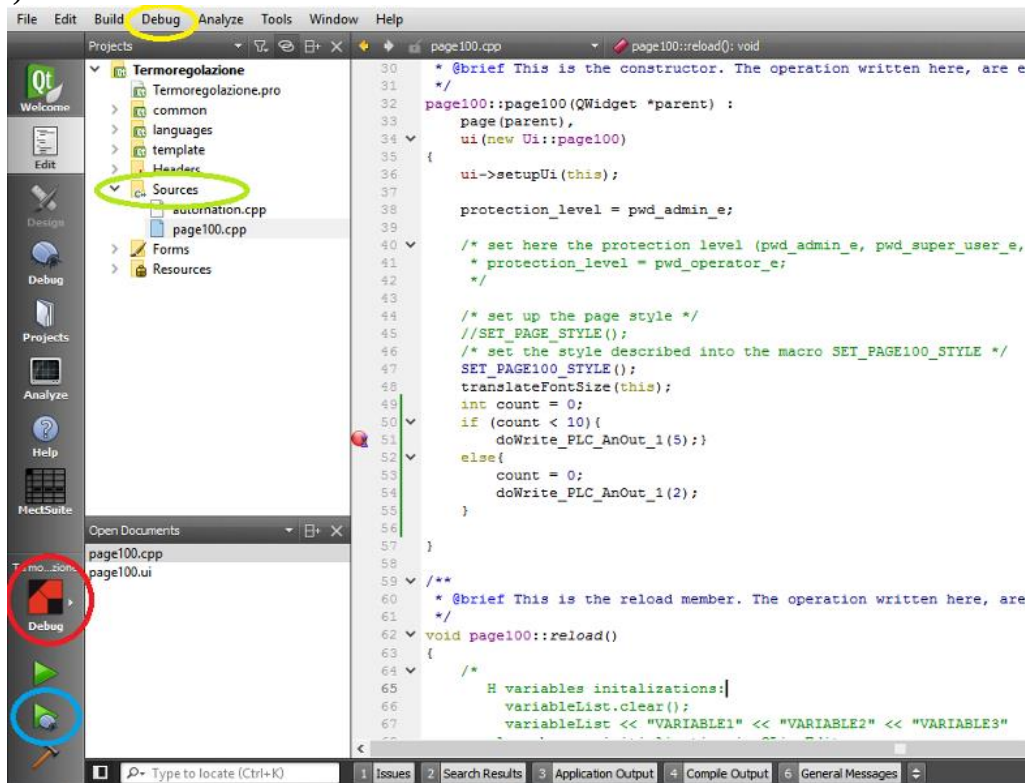


2.5 Debug

It is possible to run the HMI application debugging, compiling all in debug mode (see red circle).

To set the breakpoint in a row is sufficient a click in the gray zone on the left of the row number of the source code: for example in Sources→page100.cpp in the **page::page100()** function.

To run the sequence “*Build, Deploy e Debug*” press F5 or the “green triangle with ladybug” (see blu circle in figure below).

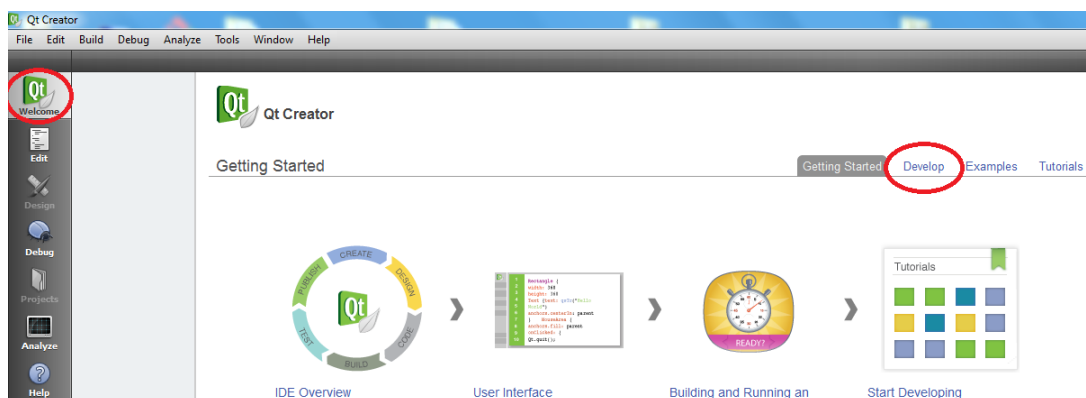


2.6 VNC activation

It is possible to connect to a panel and see/command it in remote mode. This is possible by using the VNC Client “**Ultra VNC**”.

2.7 Open an existing project

Launch “Qt Creator” and select the Tab “**Develop**”



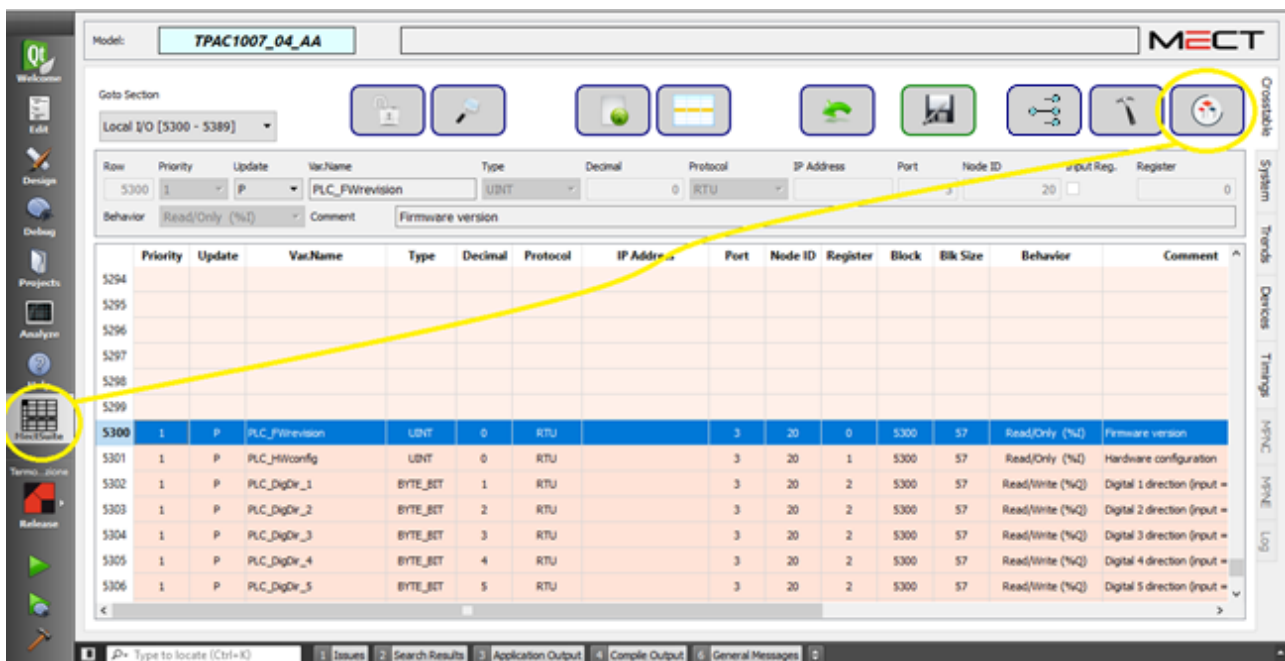
- Select your project from “**Open Project**”
or
- Select your project from “**Recent Projects**”

WARNING: if you have renamed or moved the project folder you must verify that the folder name and the location where it is located, there are no spaces or special characters!.

2.8 Open the PLC software

ATCMcontrol Engineering is the software used for managing PLC applications

PLC software is started by the  Key of the crosstable.



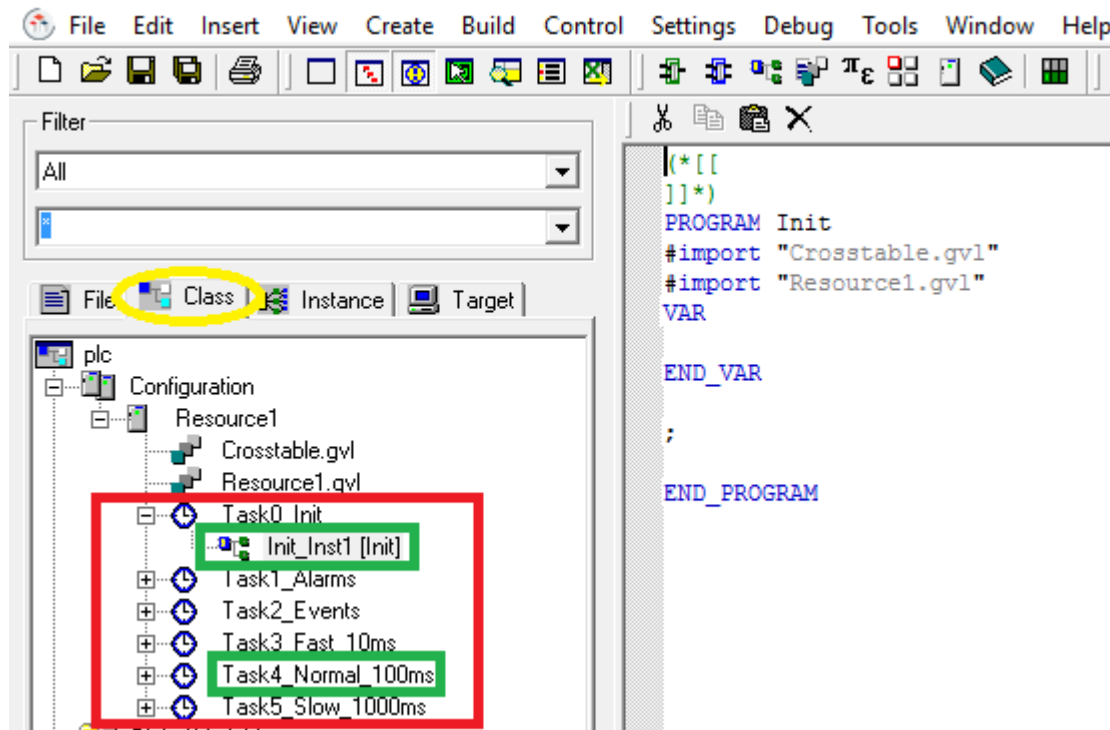
The variables of the Crosstable coincide with PLC software. The refresh of the variables list is possible by:

- Pressing  Key of the Crosstable;
- Building HMI program: 
- Deploying HMI program: 

In the ATCMControl there are the Crosstable variables imported in the PLC program.

The starting point for the PLC application is the file “Init_Inst1.st” and in the file “Crosstable.gvl” there are variables definition of the Crosstable.

See details in “**MectSuite Tutorial EN** → PLC → Insert variables in ST programs”.

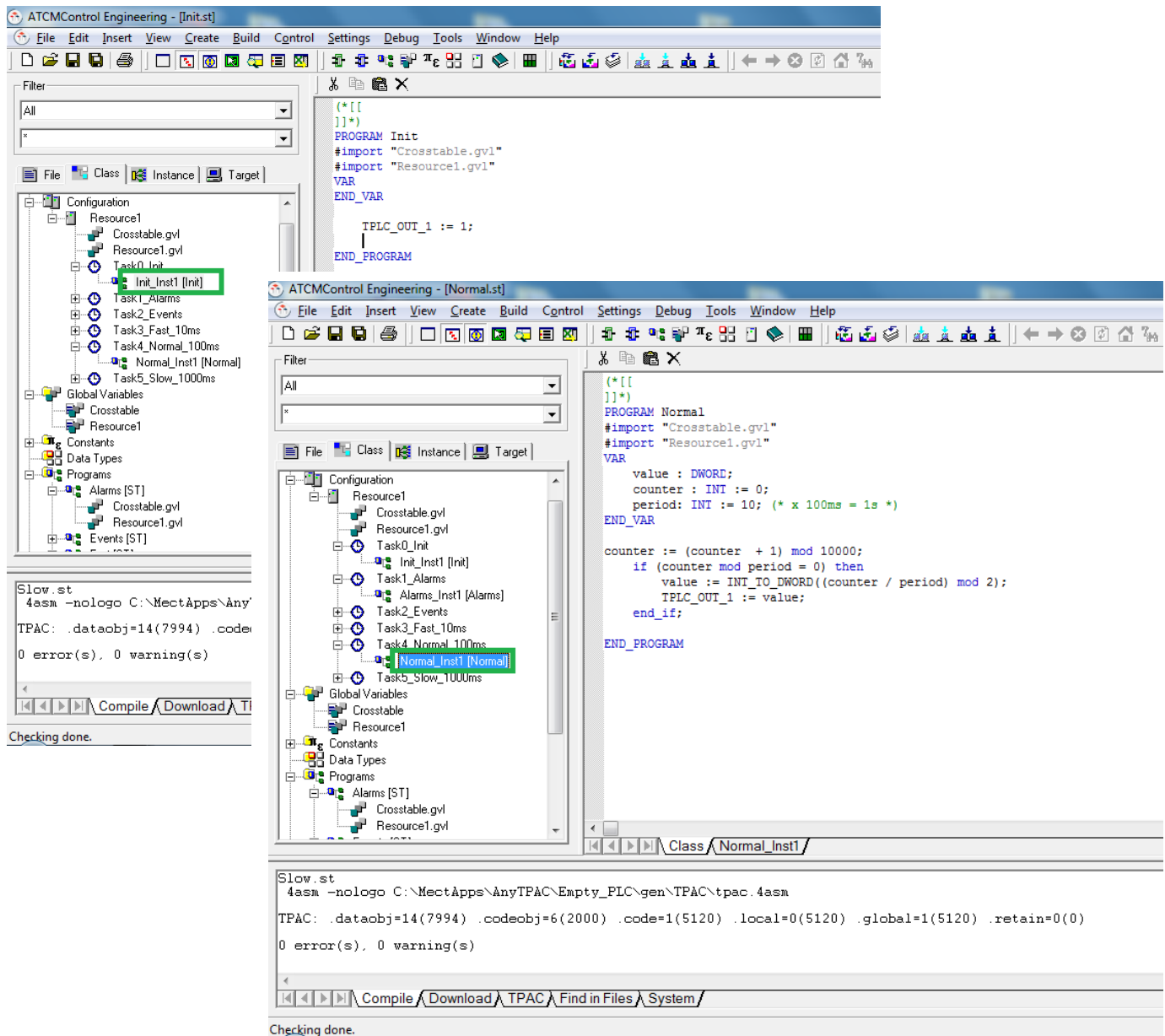


The following PLC program is useful for our exercise, in particular:

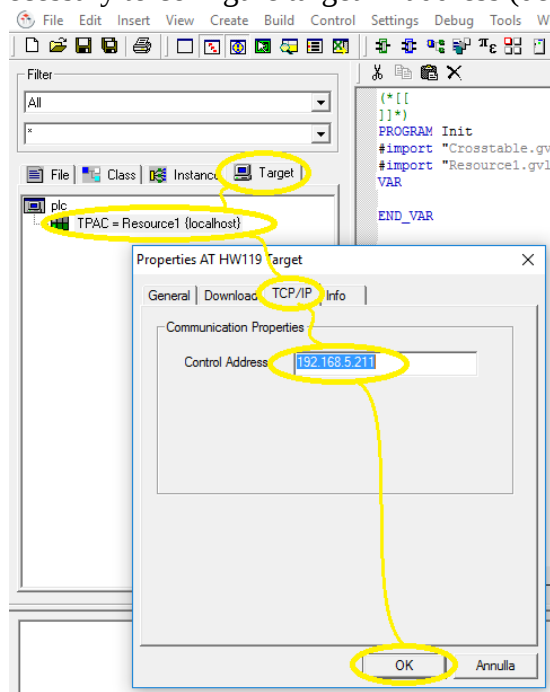
- a plc cycles counter (100 ms default) is held in counter
- the Crosstable variables are updated for graphics (time graph, ...)
- every second the Modbus TPLC_OUT_1 registry is written (square wave on output)

```
PROGRAM Init
#import "Crosstable.gvl"
#import "Resource1.gvl"
VAR
END_VAR;
    TPLC_OUT_1 := 1;
END_PROGRAM
```

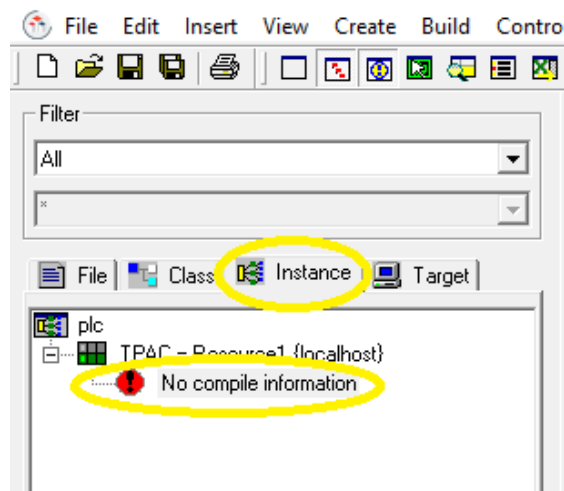
```
PROGRAM Normal
#import "Crosstable.gvl"
#import "Resource1.gvl"
VAR
    value : DWORD;
    counter : INT := 0;
    period: INT := 10; (* x 100ms = 1s *)
END_VAR;
    counter := (counter + 1) mod 10000;
    if (counter mod period = 0) then
        value := INT_TO_DWORD((counter / period) mod 2);
        TPLC_OUT_1 := value;
    end_if;
END_PROGRAM
```



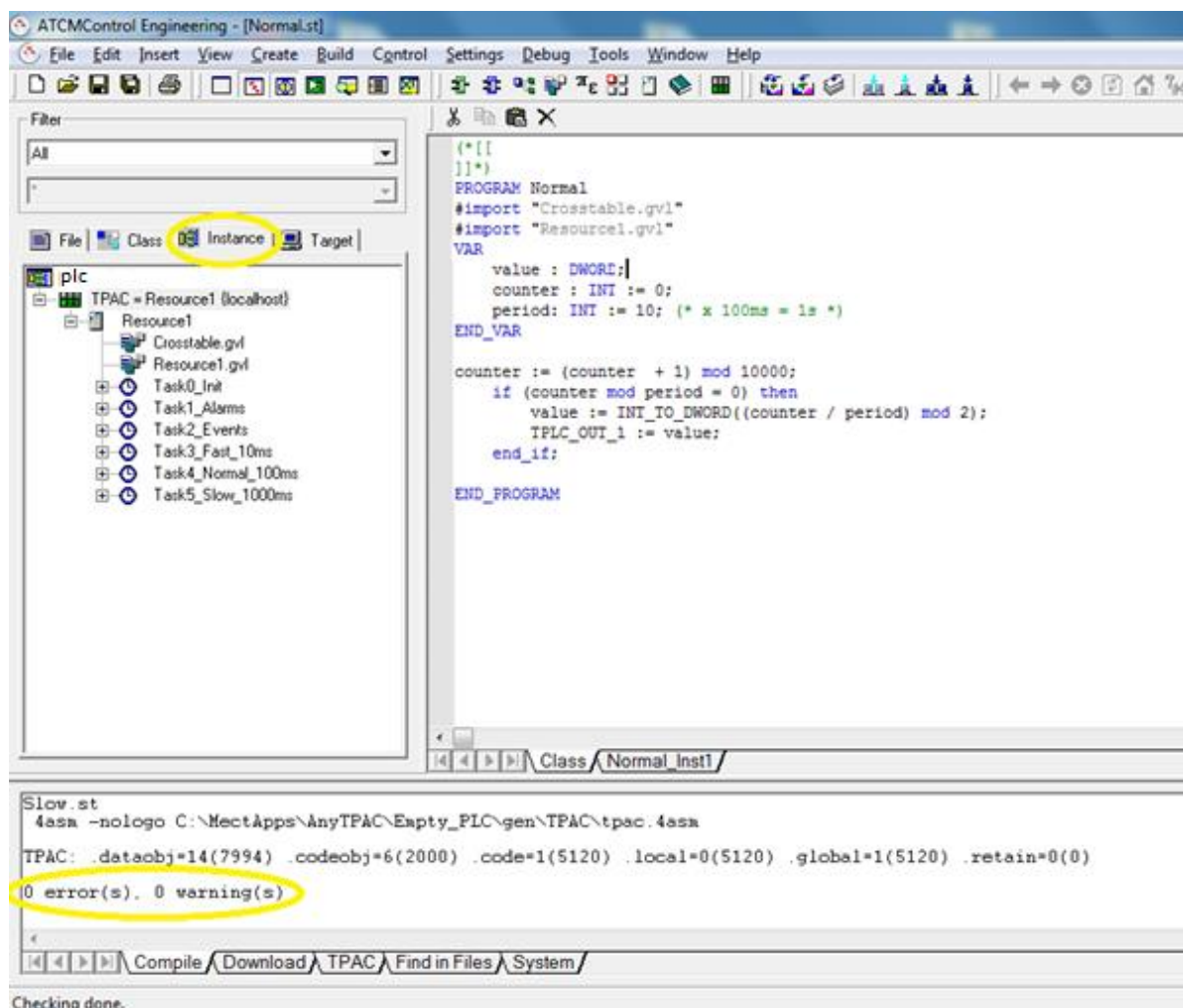
Warning: To read a registry described in the Crosstable is sufficient to use the equivalent variable;
To send PLC to operator panel it is necessary to configure target IP address (default is: 192.168.5.211).



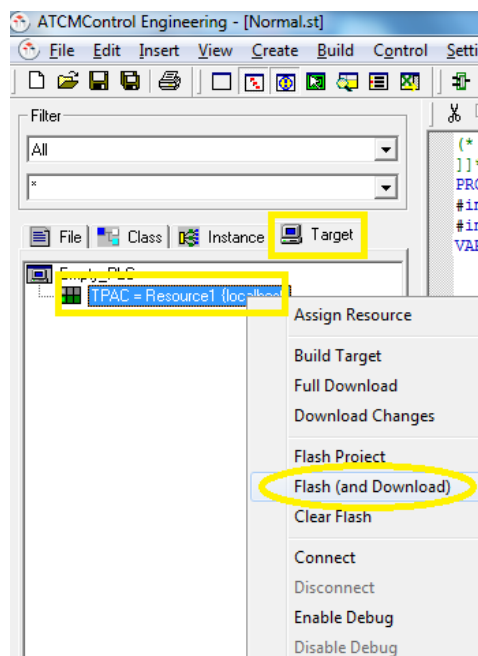
Before the build the following screen appears:



Compile (Build → Build All or F7) and check for 0 errors in the bottom of the screen.

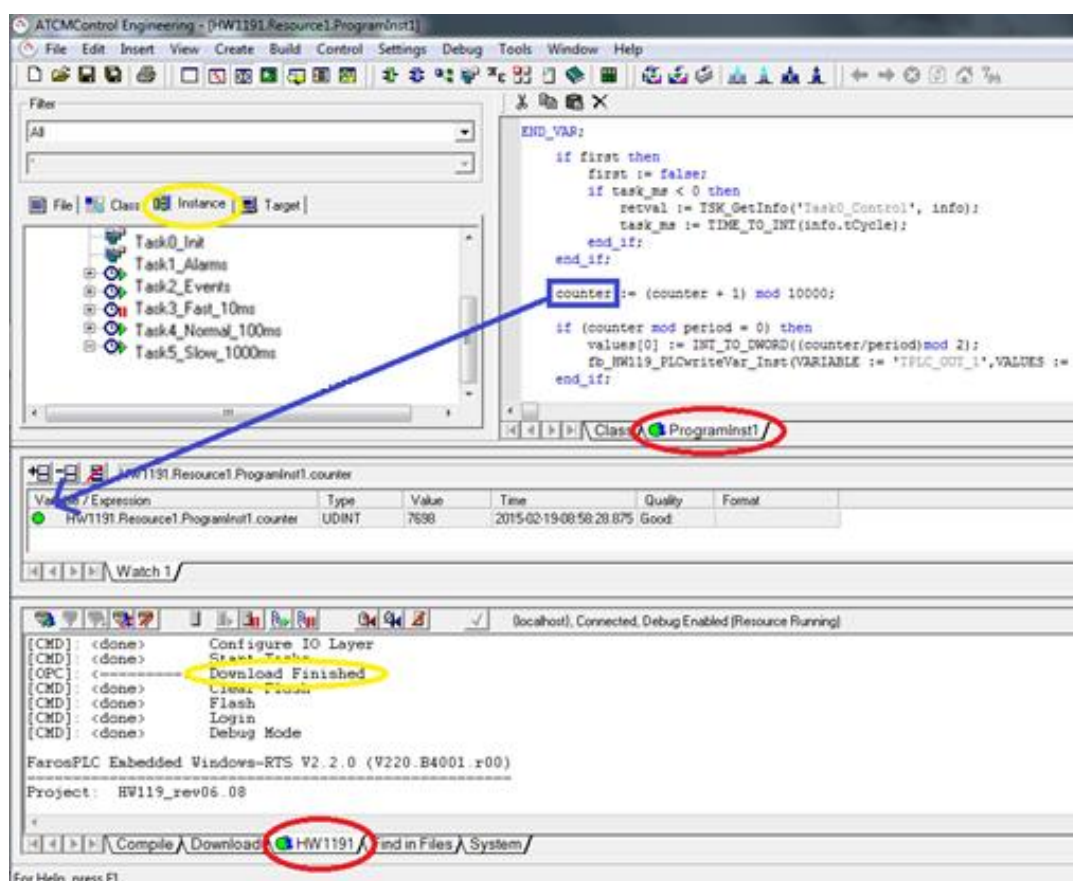


To download on target the compiled program select “**Flash (and Download)**” in the popup menu on target, see figure (accept “debug connection”).



If appears an error dialog should retry to select “**Flash (and Download)**”.
Download finishes with the message “Download Finished”.

At the end of the “Download” and at the connection restore, the “Debug” mode is automatically set.
In Debug mode there is one more tab for the programs running on target (see red circles) and in the “instance” tab the Tasks state is present (active, stopped, ...).
Target circles are green with active connection and yellow with no active connection.
It is possible to activate the refresh of the running program variables by dragging the variable in the “watch window”, see blue arrow in the figure, or in the menu “View → Watch”.
The refresh appears also hovering the mouse on the variable (it is possible to fix it with popup → Set Watchpoint).



2.9 Device variables (Local I/O)

The product variables are present in the Crosstable and are visible starting from row 5300 of “Crosstable Editor”. For example, if in HMI program you want to command an ATCMbutton to change the digital output 1 or to initialize in PLC digital output 1 to a value, use the variable PLC_DigOut_1.

ATTENTION: in the MectSuite Tutorial there is a complete list of variables (Local I/O and System). See the chapter: Device variables (Local I/O)

2.10 Input / output configuration

In devices with bidirectional I / O, with analog inputs or outputs, with Fast I / O or with encoder, it is necessary to tell to the product how these lines need to be used in your application and that is why it is called “configuration”.

It is important that the configuration takes place before their use, otherwise the line can be blocked for self-protection.

ATTENTION: in the MectSuite Tutorial you can view the list of variables used to configure inputs / outputs. See the chapters: Synoptic (Local I/O) or Inputs / outputs configuration