



MECT
mechatronic thinking

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

Telefono: +39 011 9664616 Fax: +39 011 9664610

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

MECT suite

Corso Base

Versione 1.5

Sommario

Strutturazione corso MECT suite.....	4
Famiglia ARM9 vs Famiglia Cortex A72	5
Differenza tra TP, TPX e TPAC.....	5
Famiglia ARM9.....	5
TPAC1007	5
Elenco Ingressi uscite.....	6
Morsettiere.....	6
TPAC1008	8
Morsettiere.....	9
Famiglia Cortex A72.....	10
TPX4100_01A.....	10
Dispositivi collegabili ai TPAC/TPX/TP	12
MPNC006 per interfaccia Modbus	12
Schema di principio	12
Dispositivi per l'estensione degli ingressi uscite	12
MPNE1001 per interfaccia Modbus	13
Schema di principio	13
Dispositivi per l'estensione degli ingressi uscite	13
Software	14
Struttura del software	14
Realizzazione di un semplice programma	14
Specifiche del progetto.....	14
HMI: software per la grafica	14
Creazione progetto.....	15
Impostazioni preliminari.....	16
Passo 1: modifica Logo	20
Passo 2: definizione variabili	20
Prima pagina: page100	23
Seconda pagina Setup: page200.....	28

Terza pagina: page300.....	33
Storici.....	36
Allarmi.....	39
Trasferimento progetto HMI sul pannello operatore	40
ATCM control.....	40
Progetto PLC.....	40
Analisi del progetto PLC.....	42
Regolazione 1: FBD	42
Help di ATCMControl	45
Regolazione 2: ST.....	46
Assegnazione ingressi/uscite.....	47
Inizializzazioni	48
ATCM Plugin	49

Strutturazione corso MECT suite

- Presentazione HW
 - Distinzione tra TP, TPX e TPAC
 - Panoramica sui TPAC con descrizione IO disponibili
 - Panoramica sui dispositivi MECT collegabili ai TPAC/TPX/TP
- Installazione sw
- Descrizione struttura del software ed integrazione con hardware
 - Presentazione Configuratore
 - Presentazione QT
 - Presentazione ATCM Control
 - Presentazione Crosstable
- Creazione di un progetto HMI
 - Creazione di pagine
 - Utilizzo ATCM plugin
 - Log
 - Download del progetto
- Creazione tabella variabili
 - Descrizione campi Crosstable
- Sviluppo progetto HMI
- Creazione e sviluppo progetto PLC
 - Visione delle parti di Crosstable predefinite
- Strumenti per le lingue
- Ricette
- Allarmi
- Trend

Famiglia ARM9 vs Famiglia Cortex A72

La prima versione di dispositivi MECT TP1xxx è basata su processore ARM926, la generazione successiva invece è basata su processore Cortex A72.

Le due famiglie di dispositivi ARM9 e CortexA72 sono identificate rispettivamente dalle sigle TP-1xxxx e TP-3xxx

Differenza tra TP, TPX e TPAC

Nella nomenclatura MECT i TP sono pannelli operatori che possiedono interfacce verso bus di campo come Modbus, CANopen e TCP, ma non hanno a bordo nessun ingresso o uscita, i TPX sono una estensione dei TP ai quali si aggiungono un numero limitato di IO digitali, i TPAC mettono a disposizione anche ingressi e uscite analogiche.

Per quanto riguarda le caratteristiche software invece, a parità di processore, le tre classi di prodotto sono identiche.

Famiglia ARM9

TPAC1007

Il TPAC1007 è un pannello operatore monitor touch-screen da 4.3" ed una risoluzione 480 x 272 pixel a 262.000 colori.

Il TPAC1007 permette la supervisione di dispositivi connessi in rete Modbus RTU e Modbus TCP. Le reti sono gestite contemporaneamente dal TPAC1007, ed i dati provenienti da una rete possono essere inviati su un'altra creando quindi un bridge tra le due reti.

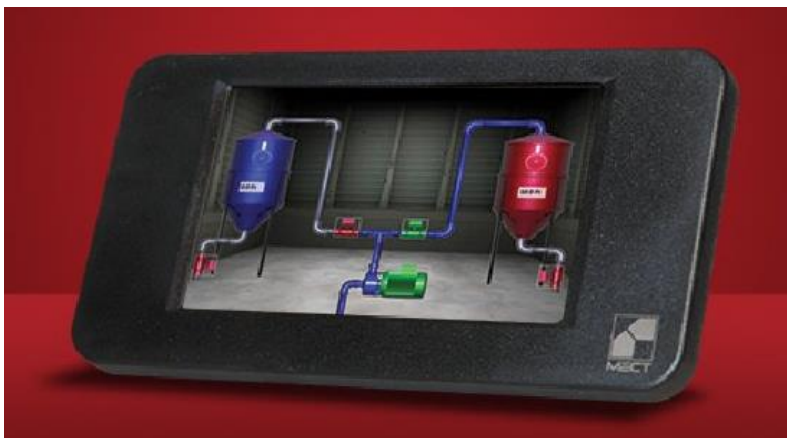
Sul TPAC1007 è presente una porta Micro-USB host che permette l'uso di una chiavetta per aggiornamento software e la storicizzazione di dati.

Sul TPAC1007 sono presenti fino a 1kByte di variabili ritentive memorizzate su flash.

Sul dispositivo è disponibili anche una Micro SD, capacità massima 64GB, per memorizzare i dati.

Un orologio calendario alimentato da batteria che permette il mantenimento della data e dell'ora fino a quattro mesi con dispositivo spento.

Lo strumento può essere richiesto in esecuzione orizzontale oppure in esecuzione verticale con l'opzione "V".

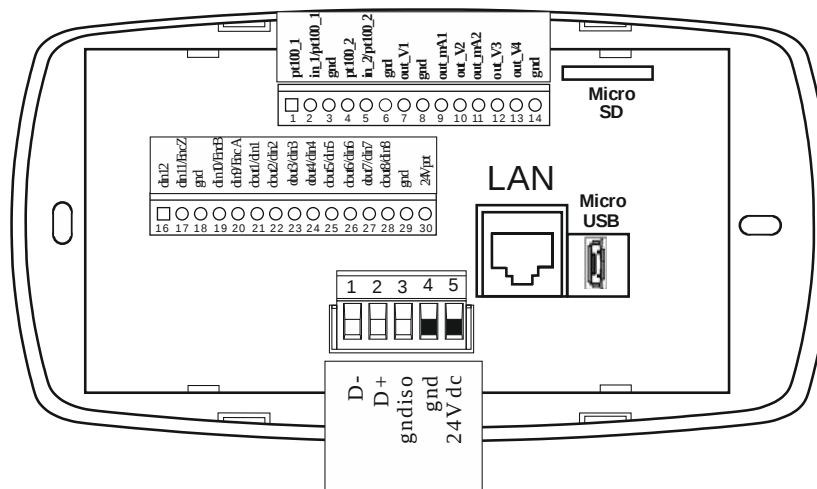


Elenco Ingressi uscite

Caratteristiche Espansione TPAC1007 04 AA

Ingressi analogici universali	N° 2 (in1 – in2)	Tipo di ingresso	Risoluzione	Bit	Note
		0÷20 mA	0.01mA	12	Impedenza ingresso 8 Ω
		0÷10ΩV	0.005V	12	Impedenza ingresso 500k Ω
		termocoppie J(0°C ÷ 600°C), T(0°C ÷ 400°C), K(0°C ÷ 1200°C) S(0°C ÷ 1710°C) B(0°C ÷ 1800°C) R(0°C ÷ 1500°C)	1°C	12	Compensazione del giunto freddo
		PT100 r -40.0°C ÷ 200.0°C	0.1°C	12	
		PT100 E -40°C÷800°C	1°C	12	
Uscite analogiche	N° 2 (out1-out2)	Tipo di uscita	Risoluzione	Bit	Note
		0÷20 mA	0.01mA	12	Impedenza Max: 400Ω
		0÷10V	0.01V	12	Impedenza Min: 1K Ω
		PWM @250Hz	1%		Impedenza Min: 1K Ω
Uscite analogiche	N° 2 (out3-out4)	0-10V	0.01V	12	Impedenza Min: 1K Ω
		PWM @250Hz	1%		Impedenza Min: 1KΩ
Ingressi digitali	N°4 (din9-din12)	Tipo di ingresso	Risoluzione		Note
		PNP o NPN (impostazione da fabbrica)	Tempo di ciclo plc		Ingressi din9 e din10 possono essere usati come encoder Fmax ingresso: 40kHz din9: ingresso A din10: ingresso B
I/O digitali configurabili	N°8 (din1-din8 / dout1 - dout8)	Tipo di I/O	Risoluzione		Note
		PNP	Tempo di ciclo plc		Max 200mA per ogni uscita. 2 A max in totale.
Alimentazione pannello		12 ÷ 28Vdc			
Alimentazione I/O		12 ÷ 28Vdc			
ModBus RTU		Tempo ciclo minimo: 10msec			
Micro USB A		2.0			
Ethernet		Bit rate: 100Mbit/sec			

Morsettiere



MECT Suite corso base

Del TPAC1007 sono stati realizzati diversi modelli che differiscono per numero e tipo di ingressi e uscite.

- TPAC1007 04 AA
- TPAC1007 04 AB
- TPAC1007 04 AC
- TPAC1007 04 AD
- TPAC1007 04 AE

Con lo stesso fattore di forma sono disponibili i seguenti modelli di TP e TPX:

- TP1043 02 A
- TP1043 02 B
- TPX1043 03 C

Si rimanda al manuale specifico del dispositivo per maggiori dettagli.

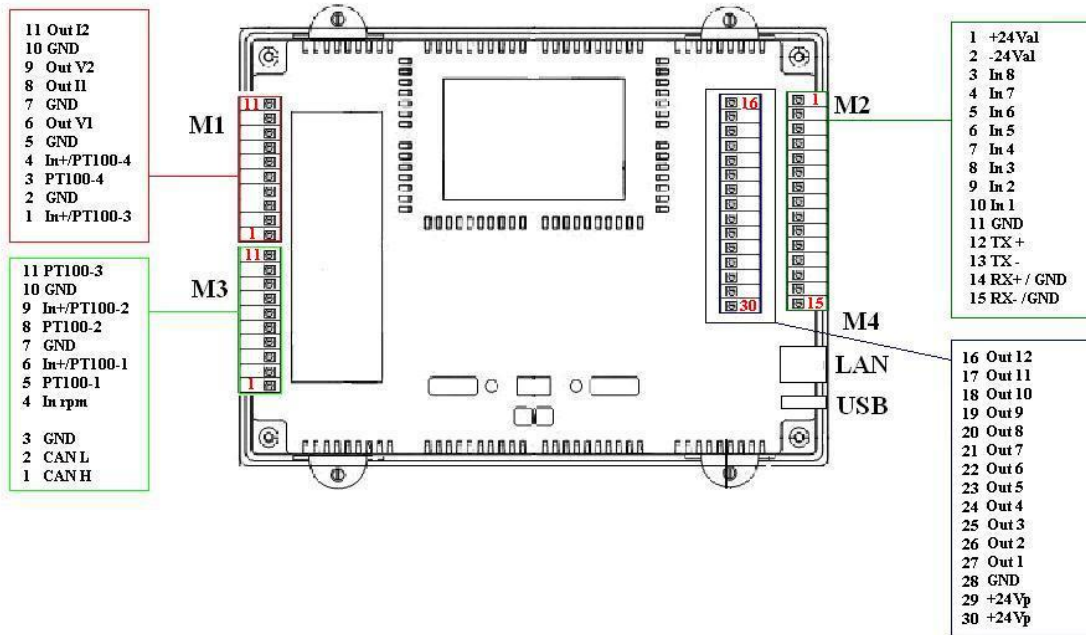
TPAC1008

Il TPAC1008 02 è un dispositivo composto da un PLC ed un HMI con monitor touch-screen da 7" ed una risoluzione 800 x 480 pixel a 262.000 colori. Il TPAC1008 02 è corredato di ingressi e uscite sia analogiche sia digitali, bus di campo come CanOpen e Modbus, una porta USB ed un'interfaccia Ethernet da 100Mbit/s.

Lo strumento può essere richiesto in esecuzione orizzontale oppure in esecuzione verticale con l'opzione "V".

TPAC1008 02 AA				
Ingressi analogici	N° 4	Tipo ingresso	Risoluzione	Note
		0 ÷ 20 mA	0.01mA	Impedenza ingresso 8Ω
		0 ÷ 10V	0.005V	Impedenza ingresso 500KΩ
		termocoppie J(0°C ÷ 600°C), T(0°C ÷ 400°C), K(0°C ÷ 1200°C) S(0°C ÷1710°C) B(0°C ÷ 1800°C) R(0°C ÷ 1500°C)	1°C	Compensazione giunto freddo
		PT100 ridotta -40.0°C ÷ 200.0°C	0.1°C	
		PT100 espansa -40°C ÷ 800°C	1°C	
Uscite analogiche	N° 2	Tipo uscita	Risoluzione	Note
		0 ÷ 20 mA	0.01mA	Impedenza massima: 400Ω
		0 ÷ 10V	0.01V	Impedenza minima: 1KΩ
		PWM @250Hz	1%	
Ingresso (rpm)	N° 1	Tipo ingresso	Risoluzione	Note
		rpm	Frequenza max 1kHz (60000 rpm)	
Ingressi digitali/Ingressi Encoder	N°8	Tipo ingresso	Risoluzione	Note
		PNP	Frequenza max 500Hz	Gli ingressi 2 e 3 possono essere usati come input da encoder incrementale Fmax 40kHz D2: A D3: B
Uscite digitali	N°12	Tipo uscita	Risoluzione	Note
		PNP	Frequenza max 500Hz	Max 500mA per ogni uscita. 2 A max totali
Alimentazione pannello		12÷36Vdc 400mA		
Alimentazione I/O		24Vdc@15%		
CANOpen		1 canale		
		Bit rate massimo: 1Mbit/sec		
		Tempo ciclo minimo: 10msec		
USB A		2.0		
Ethernet		Bit rate: 100Mbit/sec		
Uscite seriali		RS485 full duplex (tramite configurazione hardware)		

Morsettiere



Del TPAC1008 sono stati realizzati diversi modelli che differiscono per numero e tipo di ingressi e uscite.

- TPAC1008 02 AA
- TPAC1008 02 AB
- TPAC1008 02 AD
- TPAC1008 02 AD
- TPAC1008 02 AE
- TPAC1008 02 AF
- TPAC1008 03 AC
- TPAC1008 03 AD

Con lo stesso fattore di forma sono disponibili i seguenti modelli di TP e TPX:

- TP1070 01 A
- TP1070 01 B
- TP1070 01 C
- TP1070 02 F
- TPX1070 03 D
- TPX1070 03 E

Si rimanda al manuale specifico del dispositivo per maggiori dettagli.

Famiglia Cortex A72

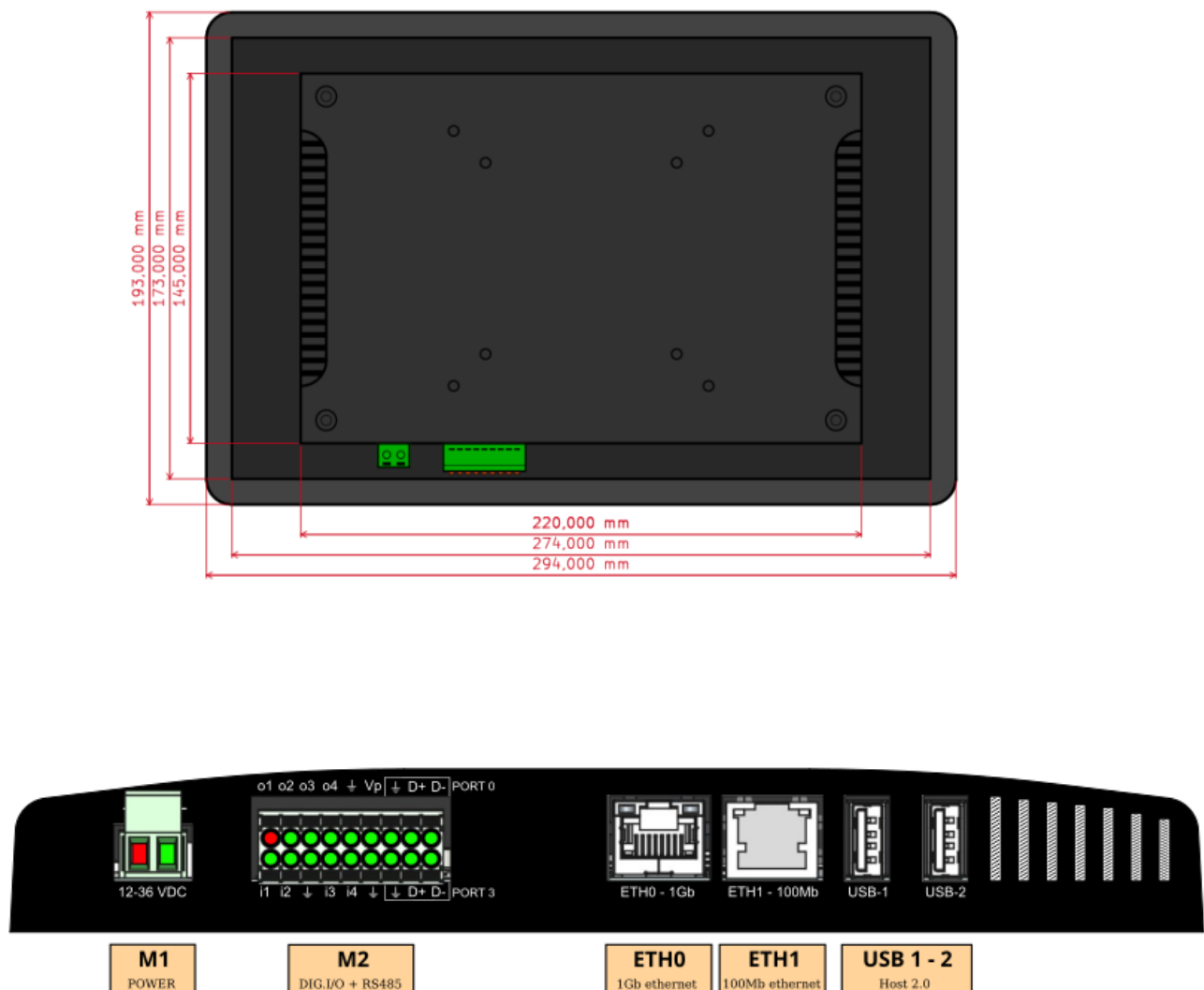
Della famiglia basata su Cortex A72 (CM4) si sono sviluppati alcuni modelli con le seguenti caratteristiche:

- TPX4100_01A
- TPX4120_01A
- TPX4150_01A
- TPX4180_01A

I dispositivi si differenziano tra loro per la dimensione del display e la risoluzione grafica.

TPX4100_01A

Il TPX4100_01A è un dispositivo composto da un PLC ed un HMI con monitor touch-screen da 10" ed una risoluzione 1280 x 800 pixel a 16M colori. Il TPX4100 è corredato di ingressi e uscite digitali, bus di campo come Modbus, due porte USB due interfaccia Ethernet, una da 100Mbit/s, una da 1Gbit.



ID	TPX4100	
M1	Alimentazione pannello	24Vdc 500mA
M2		1 DOUT 1 2 DOUT 2 3 DOUT 3 4 DOUT 4 5 GND 6 VPOT 7 GND 8 D0 + (UART3) M.S. SERIAL PORT 0 9 D0 – (UART3) M.S. SERIAL PORT 0 10 DIN 1 11 DIN 2 12 GND 13 DIN 3 14 DIN 4 15 GND 16 GND 17 D1+ (UART4) M.S. SERIAL PORT 3 18 D1- (UART4) M.S. SERIAL PORT 3
ETH0	Ethernet 1Gb (IP default in DHCP)	Bit rate max: 1Gbit/sec
ETH1	Ethernet 100Mb (IP default:192.168.5.211)	Bit rate max: 100Mbit/sec
USB 1	USB A 1	Host 2.0
USB 2	USB A 2	Host 2.0

Dispositivi collegabili ai TPAC/TPX/TP

Se gli ingressi o le uscite presenti sui TPAC non sono sufficienti è possibile estenderli attraverso dispositivi remoti che comunicano via bus di campo.

I pannelli operatori MECT sono collegabili a dispositivi di terze parti che comunichino su bus con i seguenti standard:

- Modbus TCP
- Modbus RTU
- CANopen

MECT produce dispositivi che si possono interfacciare ai propri pannelli attraverso i suddetti bus di campo.

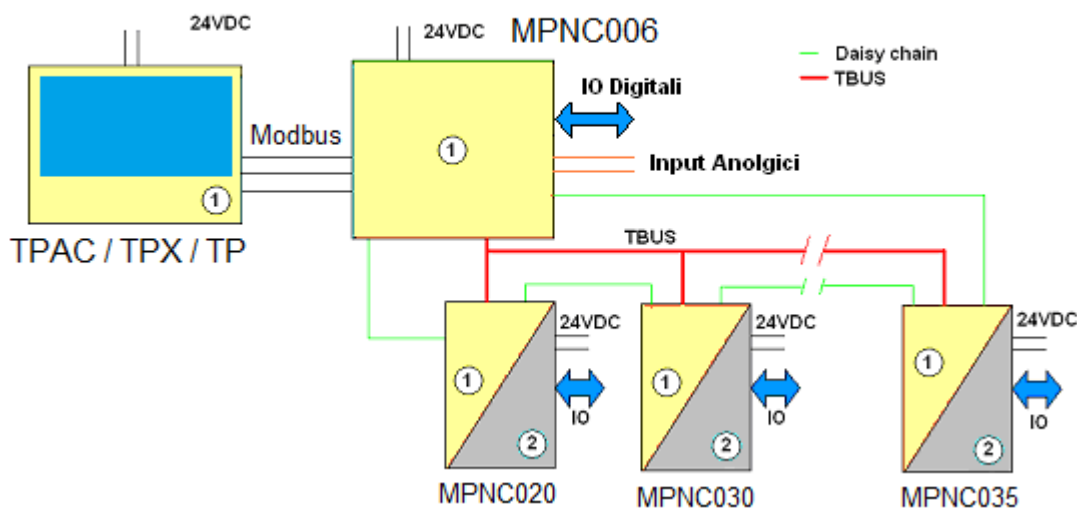
A seconda del tipo di bus di campo sono disponibili dispositivi opportuni.

MPNC006 per interfaccia Modbus

MPNC006 e MPNE1001 sono dispositivi programmabili con una interfaccia Modbus RTU slave.

Schema di principio

Questi dispositivi sono l'interfaccia, affiancati ad essi si possono inserire diversi moduli (che quindi sono indipendenti dal bus) per estendere la dotazione di ingressi/uscite disponibili, come mostrato schematicamente nella figura seguente.



Dispositivi per l'estensione degli ingressi uscite

Sono disponibili i seguenti moduli per l'estensione degli ingressi e delle uscite dei pannelli operatori:

- MPNC020-1: modulo con 16 uscite digitali PNP
- MPNC020-2: modulo con 16 ingressi digitali PNP
- MPNC030: modulo con 4 ingressi analogici configurabili

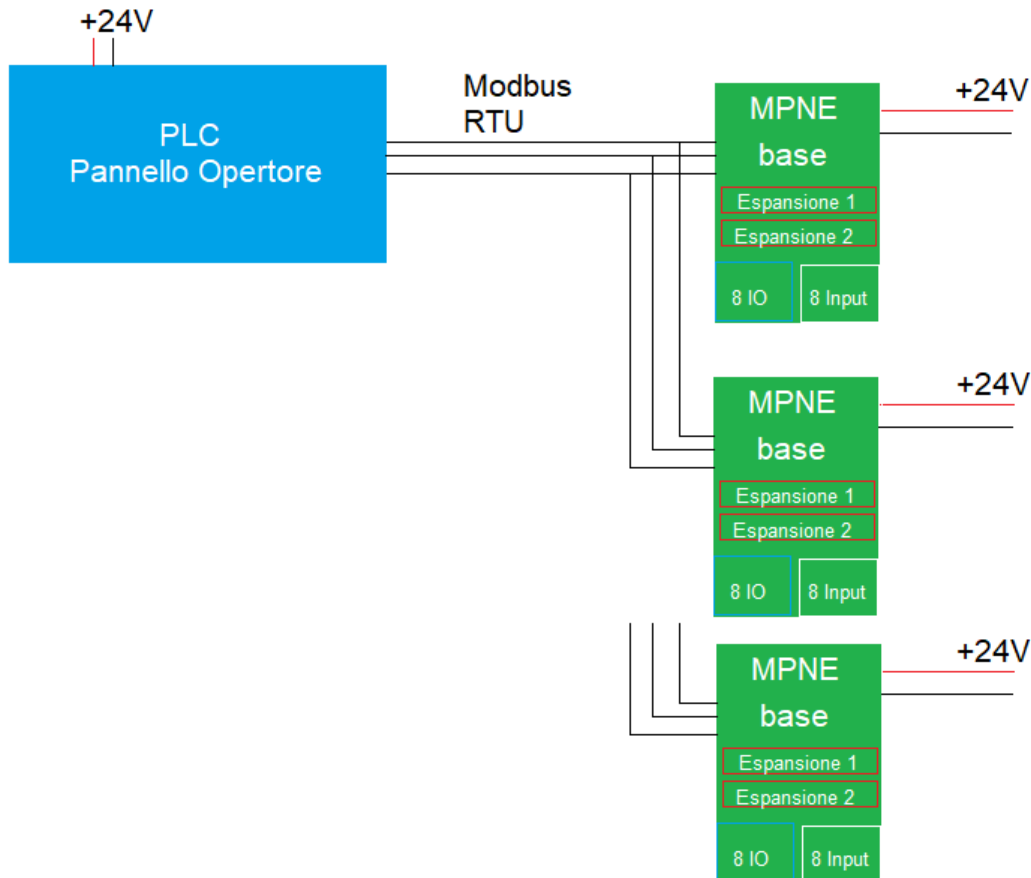
- MPNC035: modulo con 4 uscite analogiche configurabili

MPNE1001 per interfaccia Modbus

MPNE1001 sono dispositivi programmabili con una interfaccia Modbus RTU slave.

Schema di principio

Questi dispositivi sono l'interfaccia, affiancati ad essi si possono inserire diversi moduli (che quindi sono indipendenti dal bus) per estendere la dotazione di ingressi/uscite disponibili, come mostrato schematicamente nella figura seguente



Dispositivi per l'estensione degli ingressi uscite

Sono disponibili i seguenti moduli per l'estensione degli ingressi e delle uscite dei pannelli operatori:

- MPNE1001 : modulo con 8 uscite digitali PNP e 8 linee digitali configurabili
- MPNE1001-01: modulo con 8 ingressi digitali PNP
- MPNE1001-02: modulo con 8 uscite digitali PNP
- MPNE1001-05: modulo con 2 ingressi analogici configurabili e una uscita analogica configurabile

Software

Il software per la programmazione dei dispositivi è denominato Mect Suite SDK composto da diversi moduli preposti alla programmazione della grafica, del PLC ed alla configurazione.

Il software è scaricabile dal sito MECT: <https://www.mect.it/it/software-e-servizi/mect-suite>

Il software da installare espanderà i seguenti moduli:

- HMI: relativo alla parte grafica
- PLC: relativo all'automazione

Struttura del software

Il centro della struttura software è la tabella delle variabili denominata CROSSTABLE. In questa tabella sono definite tutte le variabili che devono essere scambiate tra PLC e HMI o che devono essere usate per comunicare con dispositivi su bus di campo quindi definisco una variabile in Crosstable se:

- La variabile deve essere vista tra PLC e HMI
- La variabile deve essere usata per comunicare con un dispositivo remoto

Quando si crea un nuovo progetto la Crosstable è già riempita con le variabili di sistema del dispositivo che si va ad utilizzare perciò se si crea un progetto per il TPAC1008 nella Crosstable iniziale saranno presenti anche tutte le variabili di ingresso/uscita del pannello operatore incluse quelle di diagnostica.

Realizzazione di un semplice programma

Per presentare l'utilizzo del software, proviamo ora a realizzare un semplice programma di automazione che sia composto da una parte grafica, un PLC e che comunichi con un dispositivo remoto via Modbus RTU. I dispositivi hardware sono il TPAC1008 02 AA e un MPNC006 collegato al pannello via seriale RS485 Modbus.

Specifiche del progetto

Si richiede di realizzare due regolazioni, una di tipo ON-OFF con isteresi, una proporzionale. Si leggono due ingressi analogici in tensione e si forniscono due uscite di regolazione: una digitale su dispositivo remoto per la regolazione ON-OFF, una analogica 0-10V per la regolazione proporzionale.

Dal punto di vista grafico si realizzano/richiamano sette pagine, più la navigazione:

- Pagina per la scelta setup o grafici
- Pagina per la lettura delle temperature
- Pagina di impostazione ingressi e setup di regolazione
- Pagina di grafici
- Pagina di gestione log
- Pagina degli allarmi

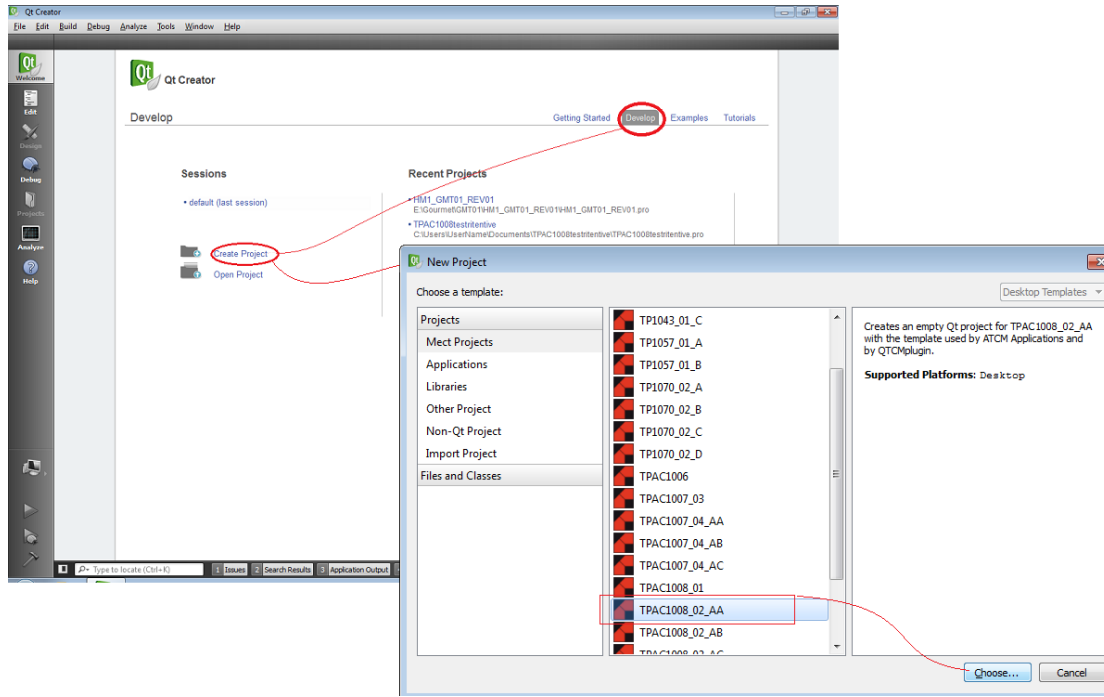
HMI: software per la grafica

Una volta installato il software HMI ci si trova sul desktop l'icona QtCreator e quindi si può iniziare a creare le applicazioni.

Creazione progetto

Aprire il programma, quindi in sequenza premere:

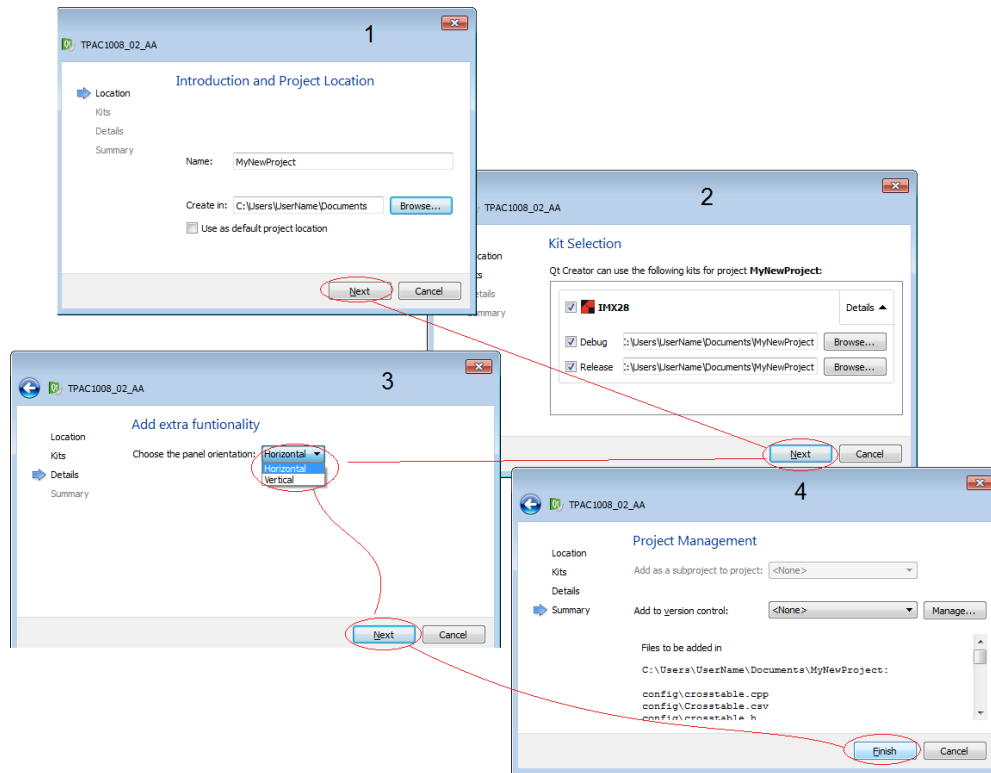
- Develop
- Create Project



Nella finestra che appare, selezionare il pannello con il quale si deve effettuare il progetto e successivamente premere **Choose...**

Nella finestra che appare:

- inserire il nome del progetto (pagina 1)
- scegliere eventualmente il nome della cartella nella quale si vuole salvare il progetto,
- Premere Next



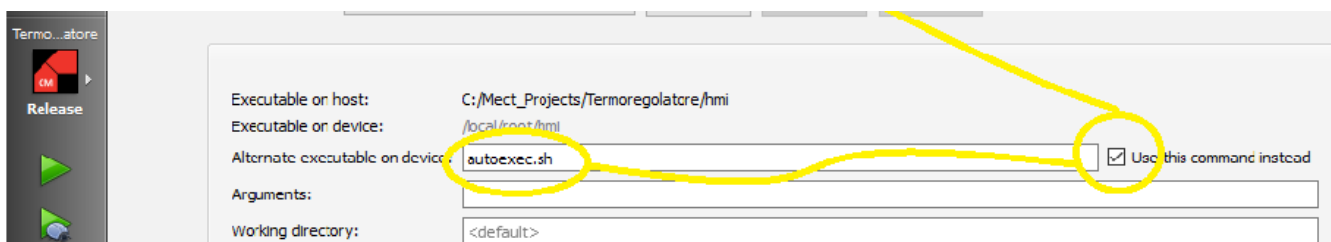
- Nella finestra successiva (2) premere Next
- Nella finestra 3 è possibile selezionare se il layout delle pagine deve essere verticale o orizzontale, quindi premere Next. Notare che se si sceglie di effettuare un progetto in orizzontale non sarà possibile poi riportarlo in verticale, e viceversa
- Nella pagina 4 premere Finish per concludere il processo di creazione del progetto

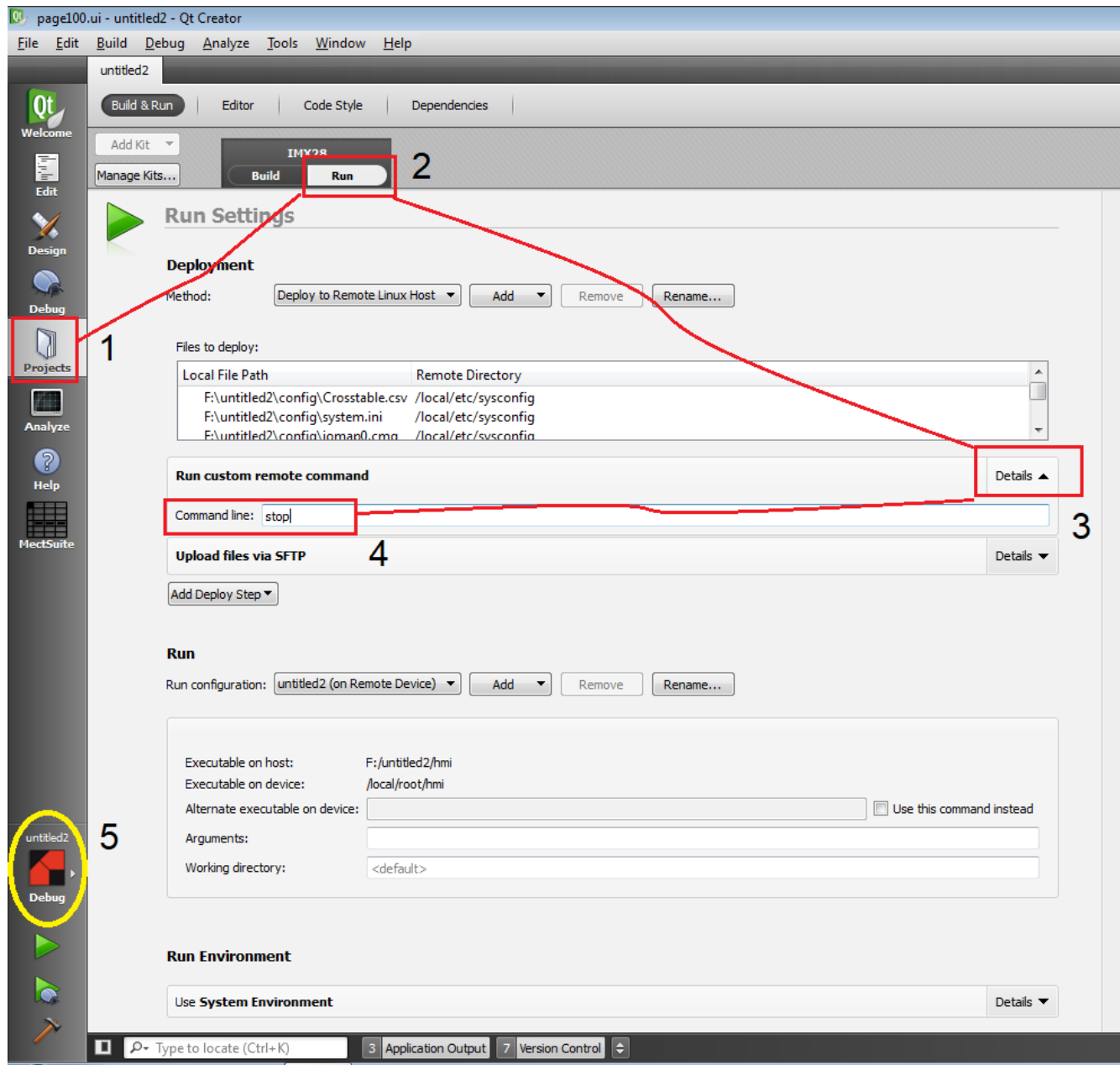
Impostazioni preliminari

Una volta creato il progetto è necessario impostare alcuni parametri che saranno necessari nel download del progetto sul pannello operatore.

1. Quindi premere sull'icona **Projects** sul lato sinistro dello schermo
2. Selezionare **Run**
3. Selezionare **Details** della riga Run custom remote command
4. Scrivere **stop** nella casella
5. Selezionare la modalità "Release"

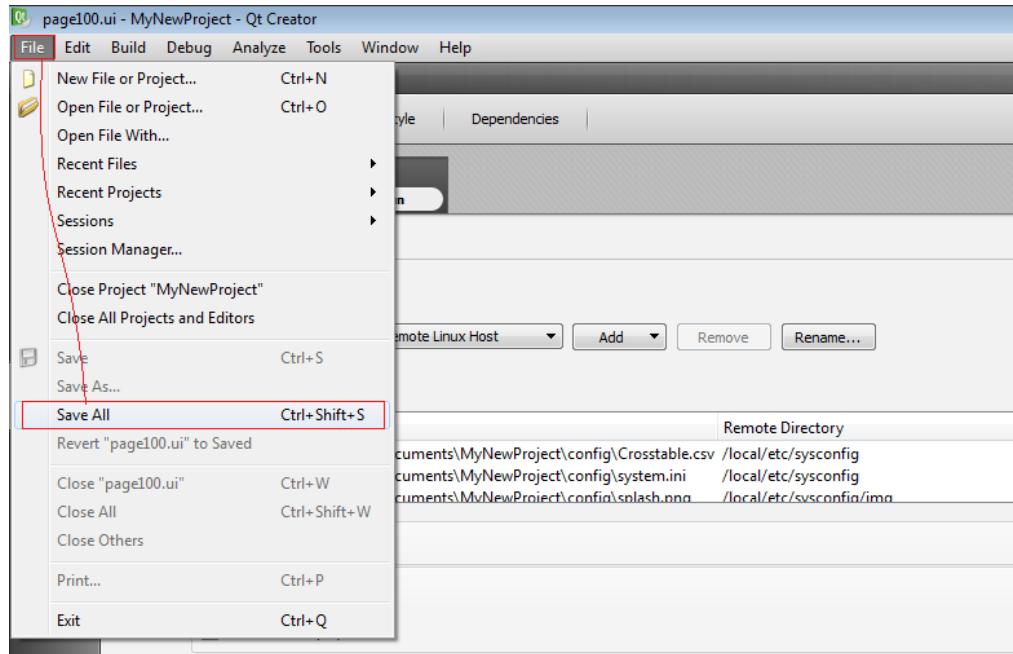
Per i dispositivi della serie CM (**TPX41xx**) nel punto 4. occorre scrivere **stop.sh** e scrivere **autoexec.sh**





Al termine di queste operazioni,

- selezionare dal menu: **File**
- Selezionare **Save All**



Un'ultima operazione prima di poter iniziare a realizzare il progetto è l'impostazione dell'indirizzo IP al quale risponde il pannello operatore.

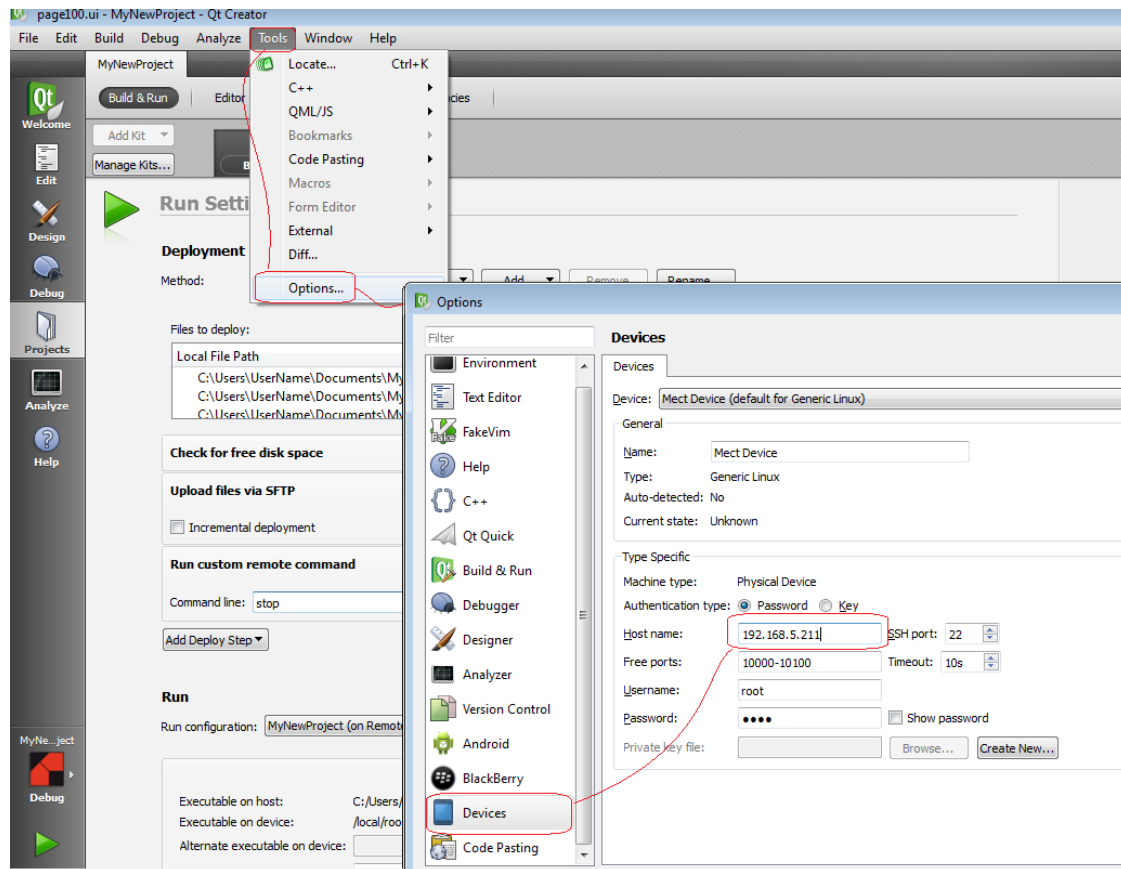
Per far ciò è necessario:

- Selezionare **Tools** dal menu
- Quindi selezionare **Options...** (l'ultima riga)

Nella finestra che appare

- selezionare  **Devices**
- quindi impostare l'indirizzo IP del pannello operatore (indirizzo di fabbrica 192.168.5.211)

MECT Suite corso base



A questo punto possiamo incominciare.

Passo 1: modifica Logo

Ogni pannello operatore parte con la schermata a tempo di presentazione del logo aziendale, per modificare la schermata inserendo un proprio logo è necessario seguire i seguenti passi.

Creare un file chiamato **splash.png** con le dimensioni del pannello operatore scelto:

TPAC1007	480 x 272
TPAC1008	800 x 480
TPX4100 – TPX4120	1280 x 800
TPX4150 - TPX4190	1920 x 1080

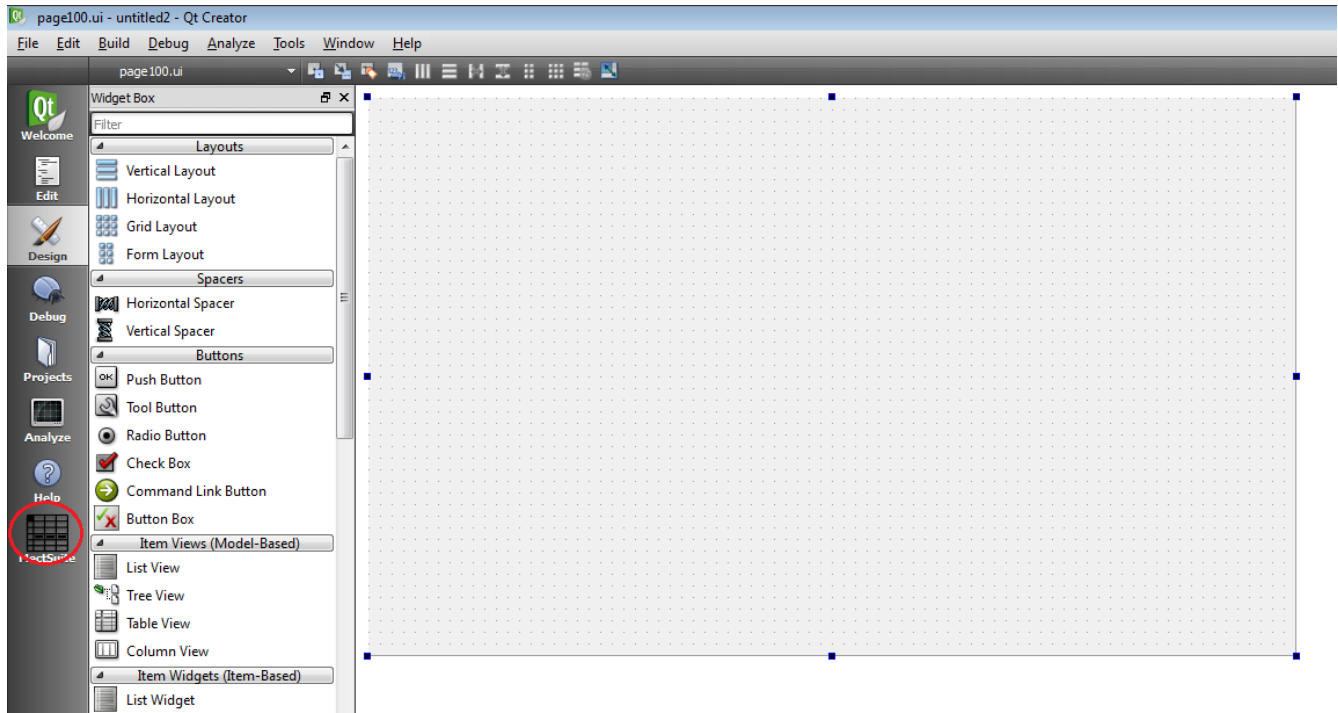
Copiare il file nella directory <nome_progetto>\config nella quale c'è già un file splash.png che quindi sarà sovrascritto.

Passo 2: definizione variabili

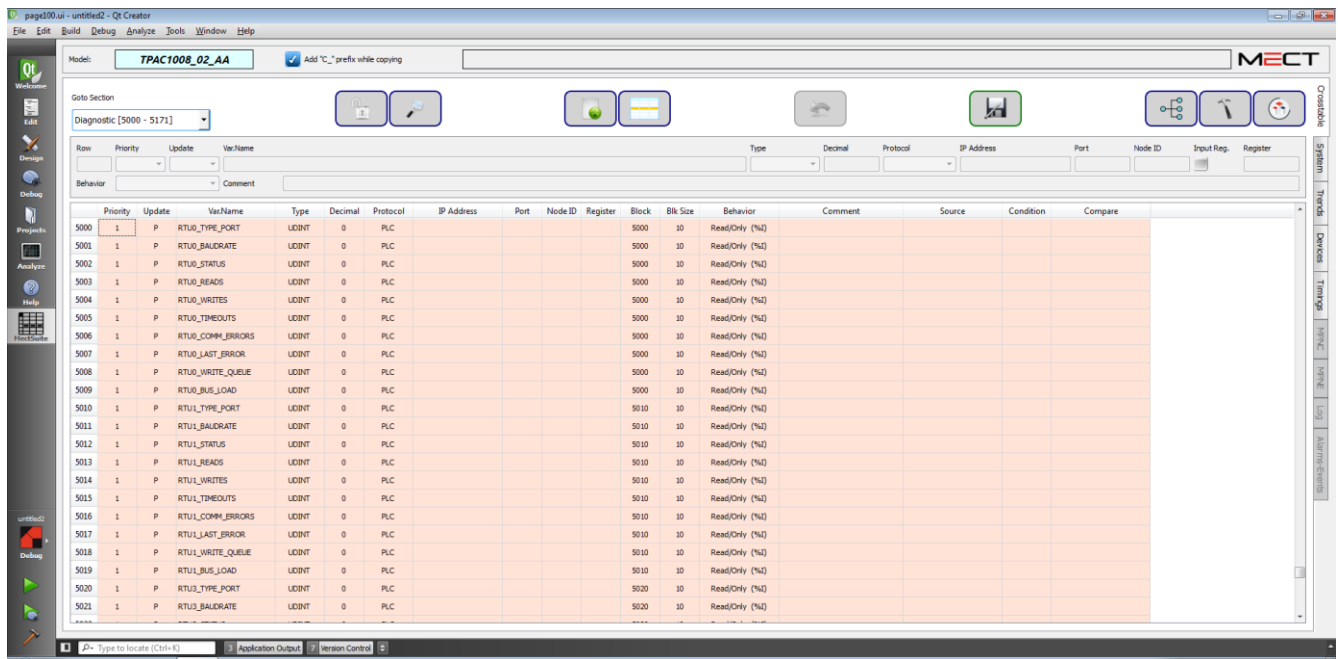
Iniziamo definendo le variabili di cui abbiamo bisogno nel nostro progetto:

Nome variabile	Tipo	Definizione
ingresso_1	INT	Variabile condizionata dell'ingresso 1
ingresso_2	INT	Variabile condizionata dell'ingresso 2
uscita_1	BIT	Variabile di controllo in uscita 1
uscita_2	INT	Variabile di controllo in uscita 2
setpoint_1	REAL	Setpoint ingresso 1
setpoint_2	REAL	Setpoint ingresso 2
Isteresi	REAL	Isteresi su controllo ON OFF
Proporzionale	REAL	Costante per regolazione proporzionale

A questo punto dobbiamo definire le variabili nella Crosstable, premendo il tasto MectSuite presente nella barra laterale



Premendo **Crosstable Editor** viene lanciato un programma che permette la configurazione delle variabili. Il programma permette di impostare per ciascuna variabile le proprietà.



La Cross table è divisa in 6 gruppi di variabili:

- Ritentive
- Non Ritentive
- Diagnostica

- Stato dei nodi
- I/O locali
- Sistema

1.	Priority	Imposta la priorità associata alla variabile: sono definite 3 livelli di priorità, i livelli di priorità indicano i tassi di rinfresco delle variabili	
2.	Update	Indica il tipo di rinfresco della variabile:	
		P	la variabile viene aggiornata ad ogni ciclo PLC
		H	la variabile viene aggiornata solo se visualizzata
		S	la variabile è aggiornata e storicizzata con tempo Slow Log Period
		F	la variabile è aggiornata e storicizzata con tempo Fast Log Period
		V	la variabile è acquisita e storicizzata solo alla variazione
		X	la variabile è richiamata dalla funzione log_shot
3.	Name	Si inserisce il nome della variabile: il nome della variabile NON può avere più di 16 caratteri	
4.	Type	Si inserisce il tipo della variabile da utilizzare, i tipi definiti sono:	
		BIT	la variabile rappresenta un bit
		BYTE_BIT	Seleziona un bit all'interno di un byte
		WORD_BIT	Seleziona un bit all'interno di una word (16 bit)
		DWORD_BIT	Seleziona un bit all'interno di una double word (32 bit)
		UINT	Intero senza segno 16 bit
		UINTBA	Intero senza segno 16 bit con byte invertiti
		INT	Intero con segno 16 bit
		INTBA	Intero con segno 16 bit con byte invertiti
		UDINT	Intero senza segno 32 bit
		UDINTBADC	Intero senza segno 32 bit con byte invertiti
		UDINTCDAB	Intero senza segno 32 bit con byte invertiti
		UDINTDCBA	Intero senza segno 32 bit con byte invertiti
		DINT	Intero con segno 32 bit
		DINTBADC	Intero con segno 32 bit con byte invertiti
		DINTCDAB	Intero con segno 32 bit con byte invertiti
		DINTDCBA	Intero con segno 32 bit con byte invertiti
		REAL	reale
		REALBADC	Reale con byte invertiti
		REALCDAB	Reale con byte invertiti
		REALDCBA	Reale con byte invertiti
5.	Decimal	Per le variabili di tipo intero o reale indica il numero di decimali dopo la virgola con cui viene rappresentata la variabile	
6.	Protocol	Indica il protocollo con cui viene scambiata la variabile	
		PLC	Variabile di interscambio tra PLC e HMI
		RTU	Variabile presente su un dispositivo remoto su Modbus RTU
		TCP	Variabile presente su un dispositivo remoto su Modbus TCP
		TCPRTU	Variabile presente su un dispositivo remoto su Modbus TCPRTU
		CANOPEN	Variabile presente su un dispositivo remoto su CANopen
		RTU_SRV	Variabile che risponde ad una richiesta di un master ModbusRTU
		TCP_SRV	Variabile che risponde ad una richiesta di un master Modbus TCP
		TCPRTU_SRV	Variabile che risponde ad una richiesta di un master ModbusTCPRTU
		MECT	Variabile su protocollo MECT
7.	Port	In caso di variabili NON PLC indica quale porta di comunicazione utilizzare: la porta di comunicazione dipende dal dispositivo utilizzato	
8.	Node ID	Indirizzo del nodo del dispositivo remoto. Nel caso di indirizzo 0 la scrittura è di tipo Broadcast.	
9.	Register	Indirizzo Modbus del registro a cui fare accesso.	

10.	Block	Indirizzo del primo registro del blocco. Per registri facenti parte di un blocco la lettura viene richiesta accorpata.
11.	Block size	Indica il numero di variabili appartenenti allo stesso blocco
12.	Comment	Utilizzato come campo di commento per la variabile. Non viene utilizzato dal sistema.
13.	Behaviour	Definisce se la variabile è di tipo "lettura", "lettura/scrittura", "allarme" o "evento".
14.	Source	In caso di variabile di allarme definisce quale variabile si tiene sotto controllo
15.	Condition	Si indica la condizione di allarme scegliendo tra: > >= < <= == != Rising Falling
16.	Compare	Valore o variabile di confronto

Inseriamo ora le variabili che abbiamo individuato in precedenza.

Una volta compilati i campi necessari, premendo , la variabile è inserita nella Crosstable.

Definiamo ora tutte le variabili evidenziate prima ed inseriamole nella Crosstable.

Name	Ritentive / Non ritentive	Priority	Update	Type	Decimal	Protocol	Comment	Behaviour
ingresso_1	NON rit	1	F	INT	3	RTU	IN1	Read/write
ingresso_2	NON rit	1	S	INT	3	RTU	IN2	Read/write
uscita_1	NON rit	1	P	BIT	0	RTU	OUT1	Read/write
uscita_2	NON rit	1	P	INT	2	RTU	OUT2	Read/write
setpoint_1	Ritentiva	2	H	REAL	2	PLC		Read/write
setpoint_2	Ritentiva	2	H	REAL	2	PLC		Read/write
Isteresi	Ritentiva	2	H	REAL	2	PLC		Read/write
Proporzionale	Ritentiva	2	H	REAL	2	PLC		Read/write
Password	Ritentiva	2	P	DWORD	0	PLC		Read/write
TempoCiclo1	Ritentiva	1	P	UINT	0	PLC		Read/write
TempoCiclo2	Ritentiva	1	P	UINT	0	PLC		Read/write
Abilita	NON rit	1	P	BIT	0	RTU		Read/write
Controllo_Allarm	Ritentiva	1	H	REAL	2	PLC		Read/write
Allarme	NON rit	1	P	BIT	0	PLC		Alarm

Quando inseriamo la variabile "Allarme" creiamo la condizione di allarme: if "ingresso_1" è maggiore di "Controllo_Allarm".

Premendo  viene richiesto se si vuole salvare le variabili anche nel progetto PLC.

Costruiamo ora le pagine grafiche del nostro progetto.

Prima pagina: page100

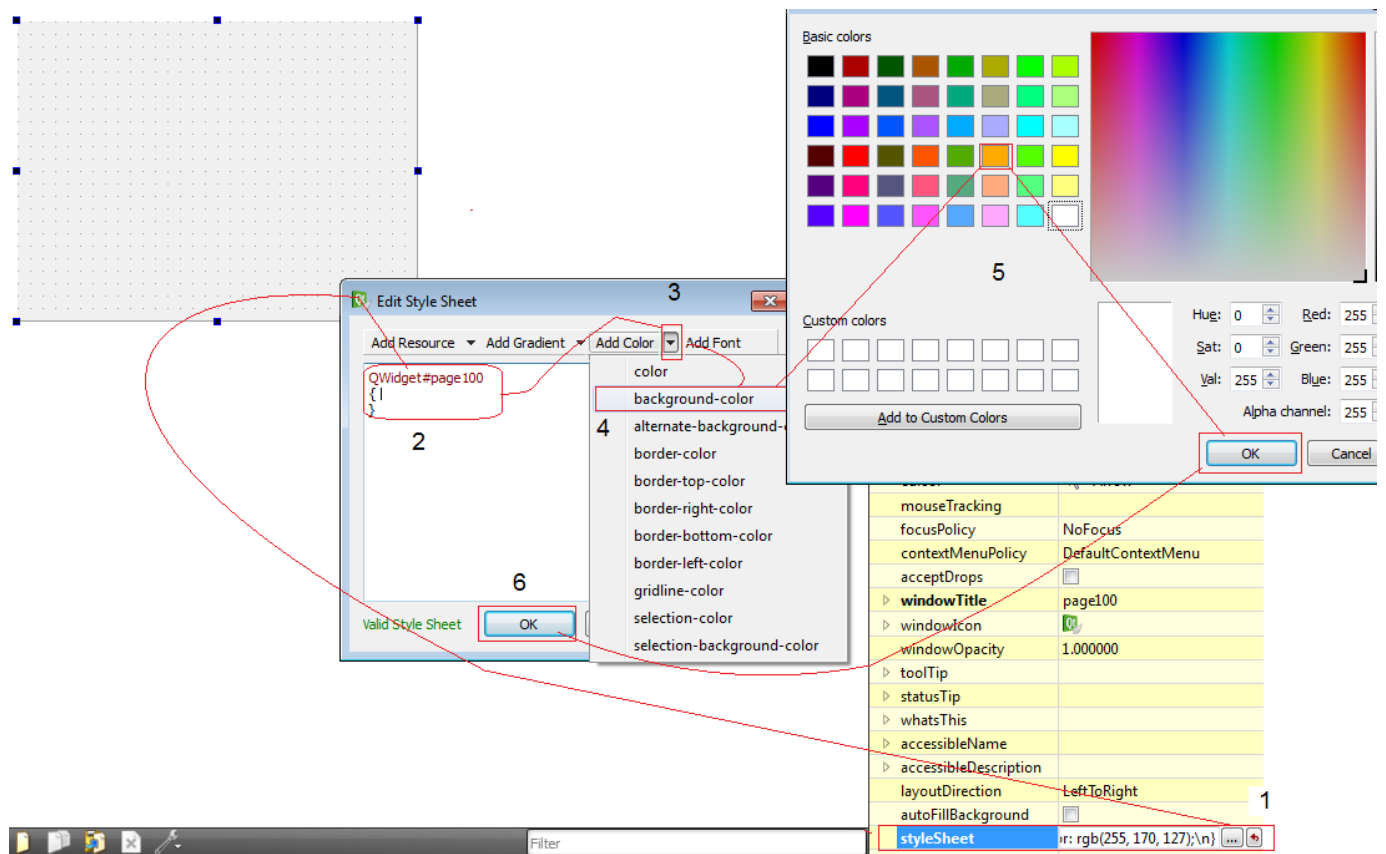
Alla creazione di un nuovo progetto il wizard, genera già la prima pagina: page100.

Modifica proprietà della pagina

Per esercizio proviamo a modificare alcune proprietà della pagina, ad esempio lo sfondo.

Per modificare lo sfondo ed inserire un colore differente rispetto al grigio di default:

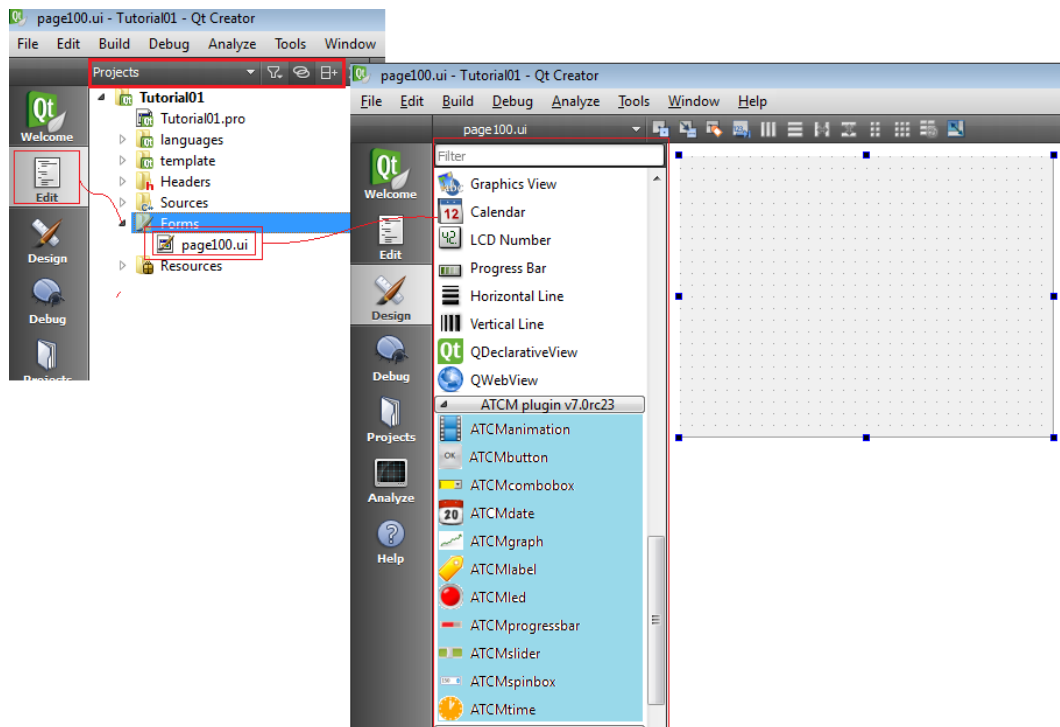
1. Selezionare **styleSheet** e premere il tasto ...
2. Nella finestra che appare posizionare il cursore tra le parentesi graffe {
3. Selezionare  nel menu **Add Color**
4. Selezionare **background-color**
5. Nella finestra che appare scegliere il colore e premere **OK**
6. Quindi premere **OK**



Inserimento oggetti

Le pagine sono costruite inserendo gli oggetti necessari; per visualizzarli è necessario:

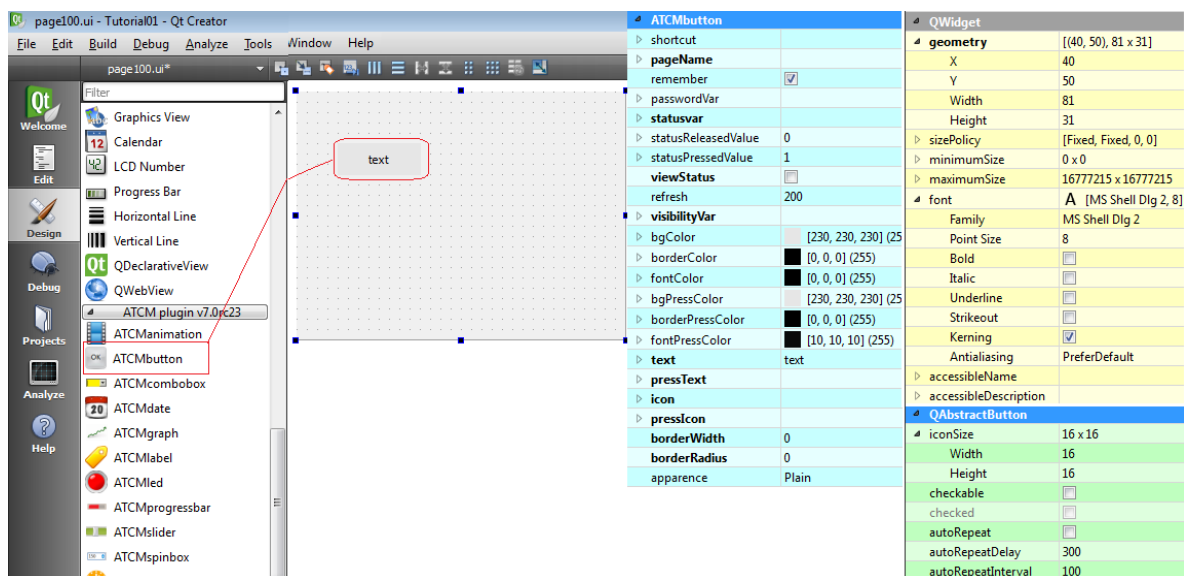
- Selezionare “Edit” ed espandere il ramo **Forms** nell’albero Projects
- Selezionare la pagina.ui nella quale si vogliono inserire gli oggetti, e fare un doppio click con il mouse



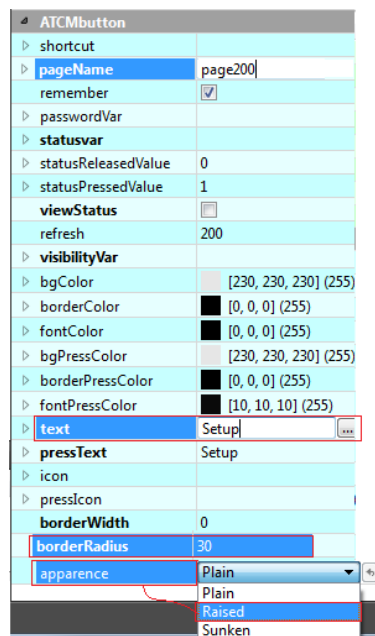
Nella parte sinistra dello schermo appaiono una serie di oggetti che sono trasportabili nella nostra pagina.

MECT ha realizzato degli oggetti semplici da utilizzare e associabili alle variabili definite nella Crosstable senza la necessità di scrivere codice C++. Gli oggetti realizzati sono nella sezione **ATCM plugin**. Per una descrizione specifica sugli ATCM plugin si rimanda al Tutorial della MectSuite.

Nella **page100** inseriamo tre bottoni che rimandano ad altre pagine del progetto. L’inserimento di un bottone avviene selezionando l’oggetto e trascinandolo sulla pagina.



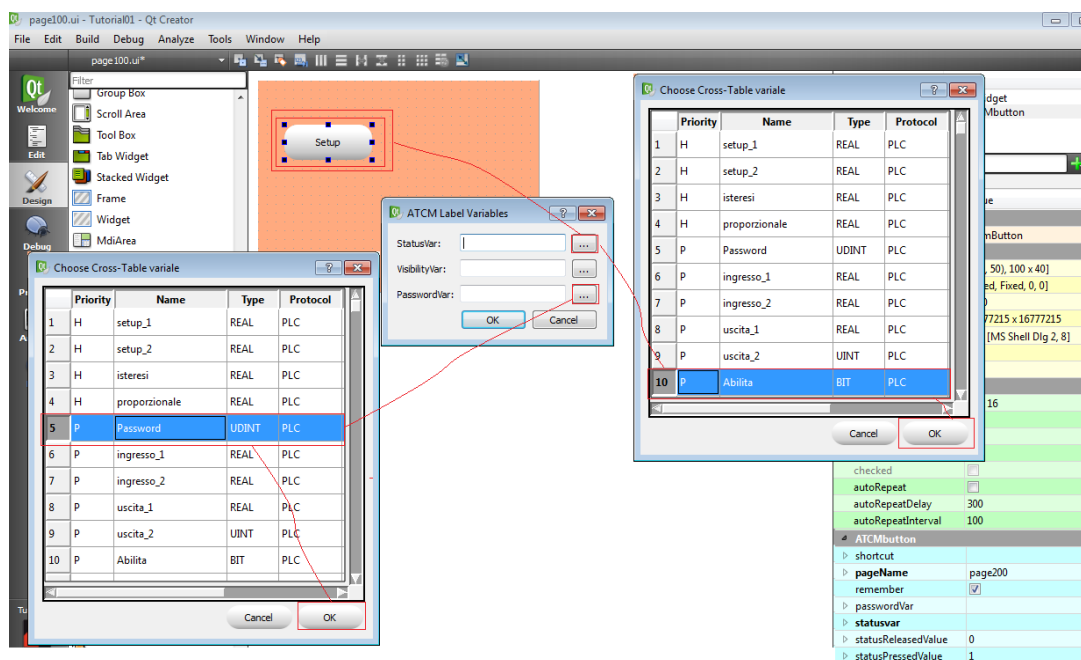
Una volta creato il bottone 1 (bottone che manda alla pagina setup) ne modifichiamo le caratteristiche nella sezione ATCMbutton (quella in azzurro):



- Modifica del testo:
 - selezioniamo la riga text ed inserire il testo desiderato
- Modifica della forma:
 - selezionare apparence e scegliere Raised
 - selezionare borderRadius per arrotondarne gli spigoli
- Salto al cambio pagina
 - Selezionare PageName ed inserire la pagina alla quale si vuole saltare una volta premuto il bottone. Mettiamo page200.

Ad ogni tasto è comunque possibile associare anche la scrittura di una variabile o la possibilità che al cambio pagina venga richiesta una password. Condizioniamo per esempio l'ingresso alla pagina Setup a seguito di una password il cui valore è immagazzinato nella variabile password.

- Fare doppio click sul bottone
- Nella finestra che appare: **ATCM Label Variables**, selezionare il tasto “...” e quindi selezionare la variabile da associare al bottone una volta premuto, quindi premere **OK**
- Per associare al bottone una variabile di password e quindi condizionare l'apertura della finestra all'inserimento corretto della password, selezionare “...” nella riga **PasswordVar**, quindi selezionare la variabile che si vuole utilizzare come password



Inseriamo un secondo bottone ed analogamente ne modifichiamo le caratteristiche:

- Text = Grafici
- borderRadius = 30
- apparence = Raised
- pageName page300

Inseriamo infine un terzo bottone che va alle pagine di informazioni e impostazioni di sistema preconfezionate da MECT mandandolo alla pagina **options**.

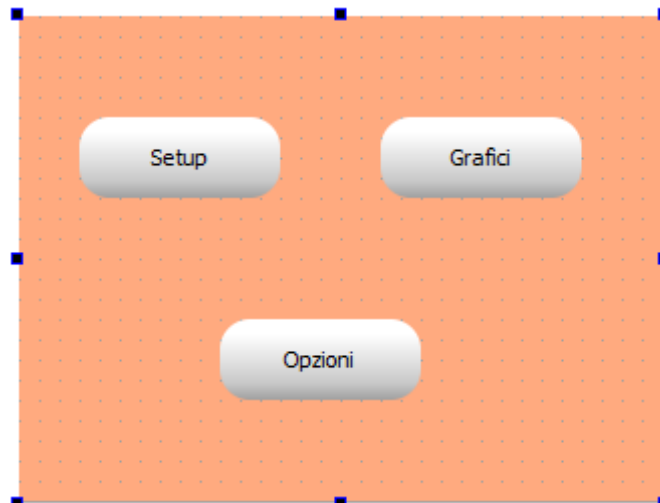
- Text = Opzioni
- borderRadius = 30
- apparence = Raised
- pageName = options

Queste pagine non seguono le regole delle pagine standard e quindi hanno nomi che non iniziano per page, ma sono precostituiti.

Pagine di sistema	
Nome pagina	Descrizione
menu	è la pagina con i 6 bottoni (page100,200,300, 400, LOGIN, DATA (data_manager), OPTION, INFO)
data_manager	pagina con i bottoni ALARM, STORE, RECIPE, TREND
options	pagina con i bottoni DATE AND TIME, SYSTEM INI, PASSWORDS, DISPLAY, LANGUAGE
system_ini	va nella pagina di configurazione
display_settings	pagina che gestisce backlight e screen saver del monitor
info	pagina che riassume le informazioni di sistema
time_set	pagina di impostazione data e ora
alarms	pagina per la gestione allarmi
alarms_history	storico allarmi
store	pagina per la gestione dei log
store_filter	filtraggio storici
trend	pagina per la gestione dei trend
trend_option	opzioni delle tracce da disegnare
trend_range	range da disegnare
recipe	pagina per la gestione delle ricette
recipe_select	pagina selezione ricette

Per una spiegazione dettagliata vedere il Tutorial della MectSuite.

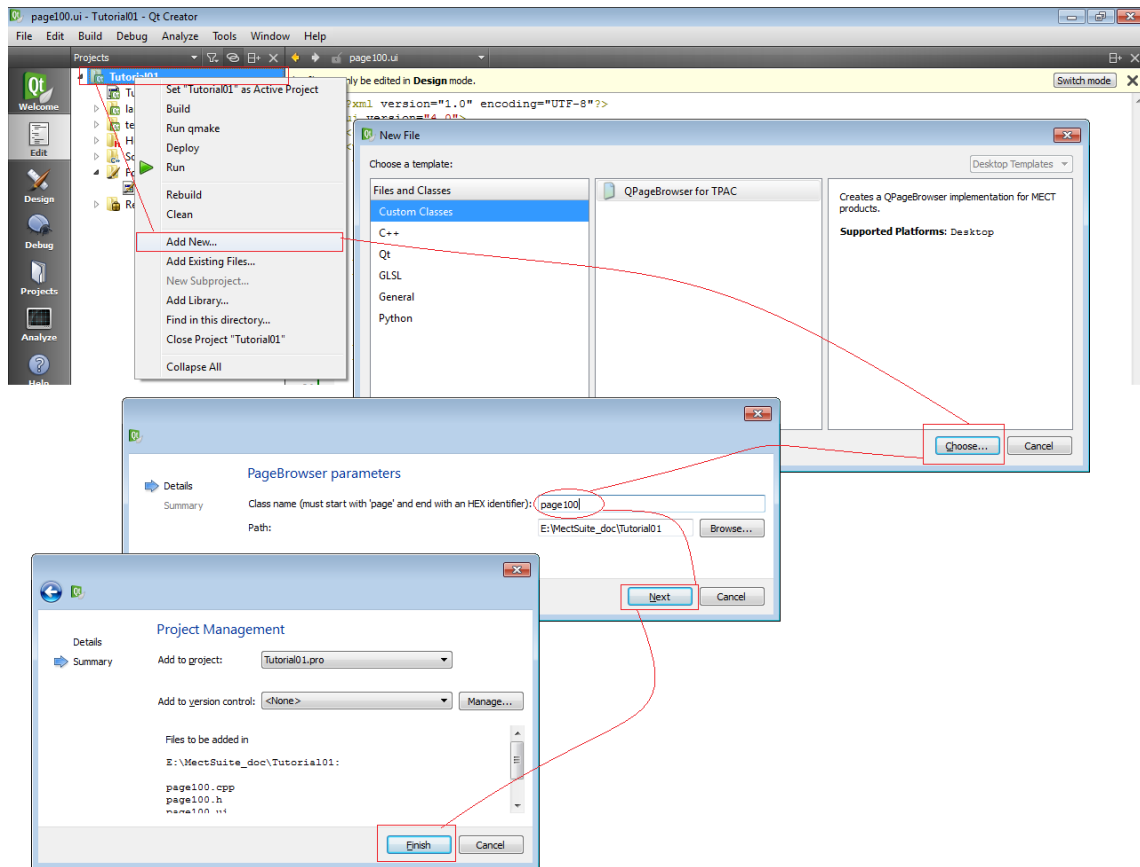
Inseriamo gli altri oggetti nella pagina, che alla fine risulta come segue:



Seconda pagina Setup: page200

Inseriamo la pagina nel progetto:

- Selezionare il nome del progetto e premere il tasto destro del mouse
- Selezionare **Add New...**
- Nella finestra che appare premere **Choose...**
- Inserire il nome della pagina secondo queste regole:
 - Il nome della pagina DEVE iniziare con **page** (tutto minuscolo) seguito da un numero, il programma propone come nome pagina page100, che di default è la pagina considerata di start e home
 - Non è possibile chiamare due pagine con lo stesso nome
- Inserito il nome della pagina (page200) , premere **Next**
- Infine premere **Finish**



La seconda pagina è quella di Setup, quindi:

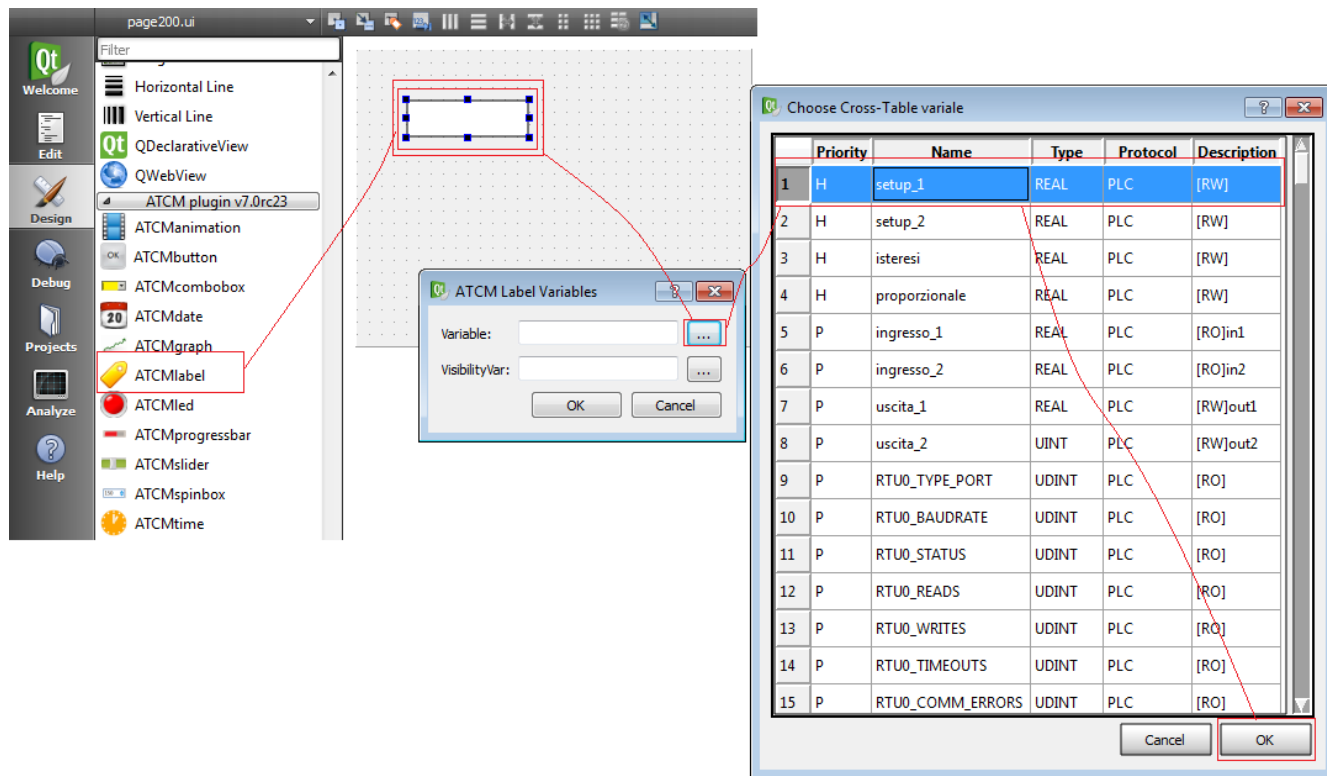
- Creiamo la pagina come fatto in precedenza e la nominiamo page200
- Modifichiamo eventualmente lo sfondo
- Inseriamo gli oggetti per le impostazioni
 - Setpoint 1
 - Setpoint 2
 - Isteresi
 - Costante Proporzionale

ATCMLabel

Per impostare i parametri di configurazione è necessario inserire delle ATCMLabel

- Selezionare ATCMLabel nella sezione ATCM plugin
- Trascinare l'oggetto nella posizione desiderata all'interno della pagina
- Selezionare la label e fare doppio click con il mouse

- Questo aprirà la finestra **ATCM Label Variables** premendo il pulsante “...” nella riga **Variable:** si aprirà la finestra **Choose Cross-Table variable**
- Scegliere la variabile che si vuole associare alla label quindi premere **OK**



Queste operazioni fanno sì che associamo alla label la variabile quindi possiamo leggere e scrivere il contenuto della variabile da interfaccia grafica. Infatti dall'applicazione, se tocchiamo la label, apparirà automaticamente una tastiera numerica, che permetterà l'inserimento del valore desiderato.

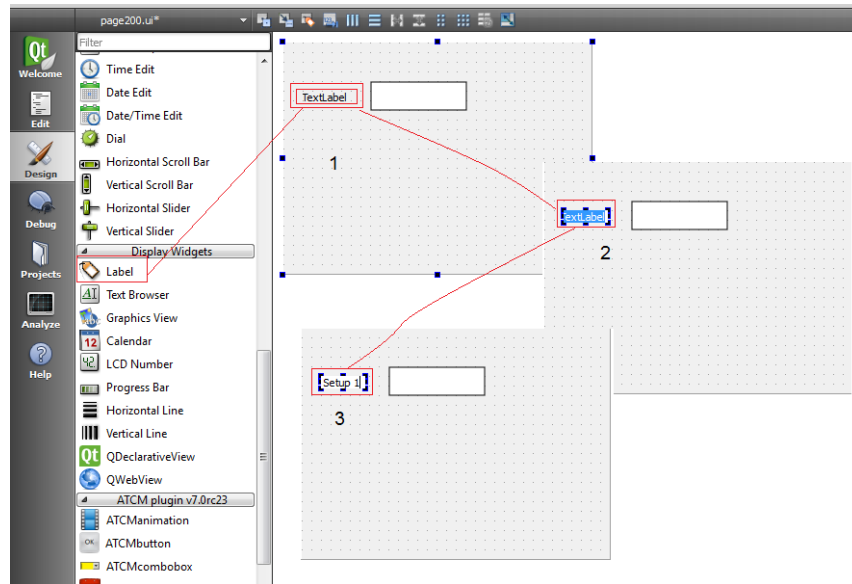
Come per il bottone anche per la label è possibile impostare alcuni parametri per la rappresentazione tra cui i più impiegati sono:

- Refresh: indica ogni quanto il valore viene letto dalla sorgente
- Max, min impostano i valori di massimo e di minimo che il parametro può assumere. Quando appare la tastiera si effettua un controllo che il valore immesso sia dentro i limiti impostati, in caso contrario viene emesso un messaggio di allarme e il parametro non viene aggiornato
- visibilityVar: se impostato, la visibilità della variabile è condizionata ad una altra variabile, per esempio se vogliamo vedere la label solo se è a 1 l'ingresso di abilitazione, questo meccanismo ce lo permette

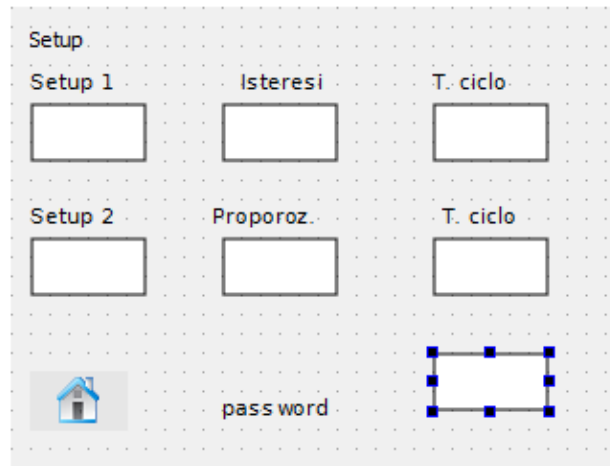
Inseriamo ora un prefisso alla ATCM label.

- Selezionare **Label** e trascinarla sulla pagina
- Per modificare il messaggio da Textlabel a qualcosa di diverso
 1. Fare doppio click sull'oggetto
 2. Scrivere il testo richiesto

3. Premere invio



Inseriamo ora gli altri oggetti della pagina Setup, ripetendo le operazioni precedenti. Al termine la pagina si presenta come mostrato in figura:



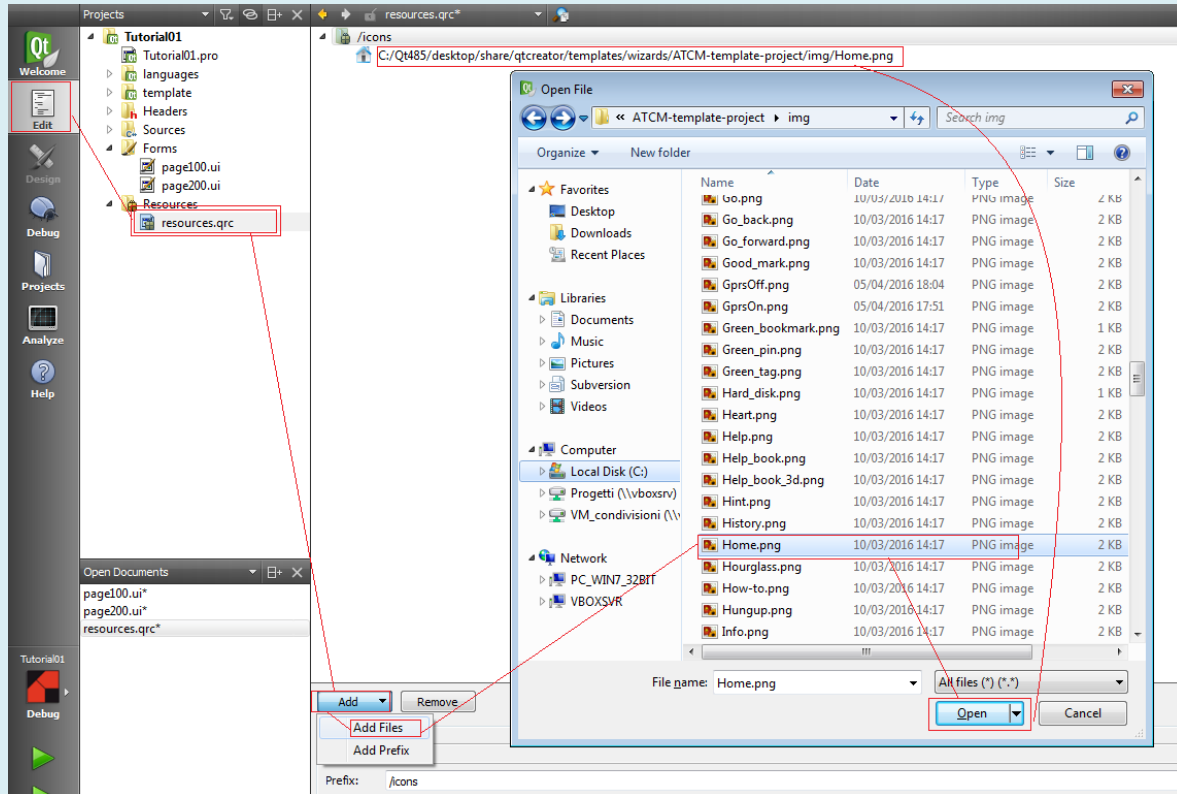
Notare che nella pagina è stato inserito un bottone con un'icona, vediamo ora come sono stati inseriti e la loro funzionalità:

Il bottone è un normale ATCMbutton e quindi si inserisce come visto in precedenza. Una volta trascinato nella pagina, affinché esegua il ritorno alla pagina iniziale è sufficiente inserire nella proprietà **pageName** della sezione ATCMbutton: **HOME**, se invece si vuole ritornare solo al livello precedente (in caso di più livelli di pagina) si deve scrivere **BACK**.

Per inserire l'immagine invece è necessario per prima cosa aggiungere l'immagine alle risorse del sistema.

Aggiunta di una risorsa

- Premere sull'icona **Edit**
- Espandere l'albero **Resources**
- Fare doppio click su **resources.qrc**
- Selezionare **Add**, quindi **Add Files**
- Selezionare il file che si vuole importare (le immagini DEVONO essere di tipo png)
- Premere **Open**



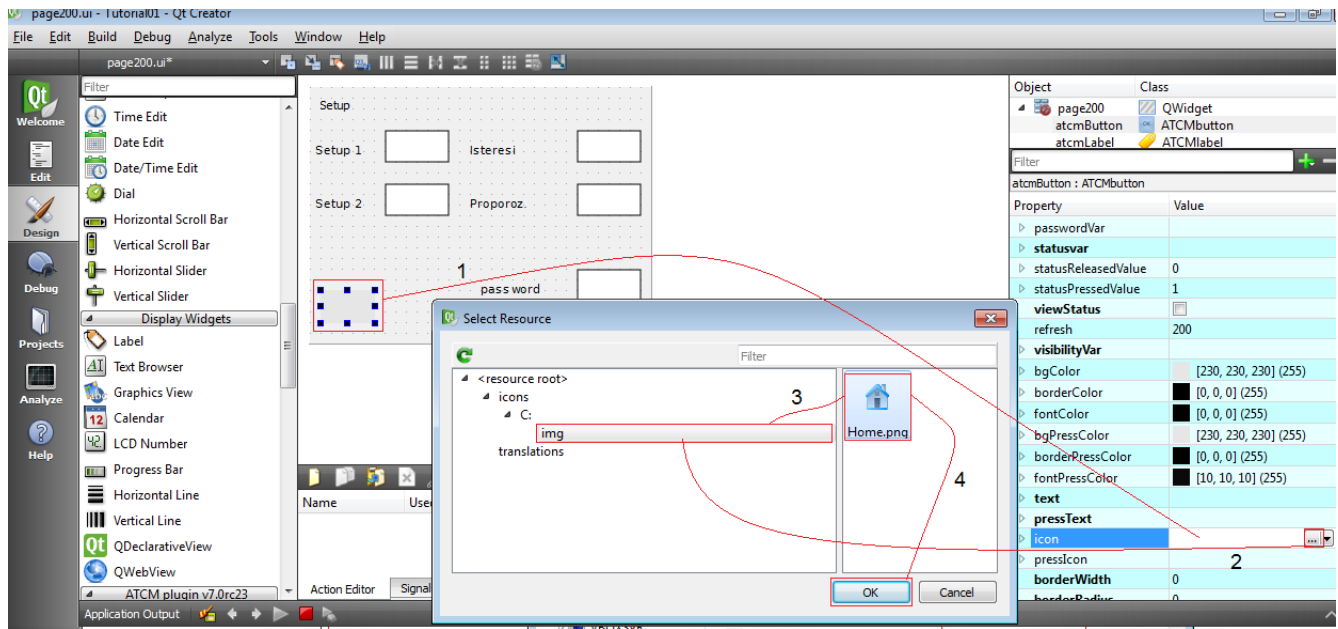
Per concludere:

- Selezionare **File** dal menu, premere **Close All** ed alla richiesta di salvare premere **Save All**

L'immagine sarà ora disponibile nel progetto.

Una volta aggiunta la risorsa di sistema aggiungere l'immagine al bottone come segue:

1. Selezionare il bottone
2. **Icon** nella sezione ATCMbutton
3. Dalla finestra che appare selezionare l'immagine che si desidera inserire nel bottone
4. Premere OK



Ora nel bottone di fianco al testo apparirà l'immagine selezionata.

A questo punto è possibile: eliminare il testo e ridimensionare l'immagine.

Per eliminare il testo bisogna selezionare **text** nella sezione ATCMbutton e cancellare il testo presente.

Per ridimensionare l'icona dalla sezione QAbstractButton selezionare iconSize e quindi impostare la dimensione richiesta.

QAbstractButton	
iconSize	32 x 32
Width	32
Height	32
checkable	☐
checked	☐
autoRepeat	☐
autoRepeatDelay	300
autoRepeatInterval	100

Terza pagina: page300

La terza pagina contiene i grafici, la lettura delle variabili di ingresso e uscita e la possibilità di accedere alle pagine di trend, allarmi e storici:

- Creiamo la pagina come fatto in precedenza e la nominiamo page300
- Modifichiamo eventualmente lo sfondo
- Inseriamo gli oggetti necessari
 - Grafico delle grandezze in ingresso
 - Label per la rappresentazione numerica delle grandezze in ingresso
 - Bottone per la storicizzazione delle variabili

- Bottone per accedere alla pagina dei trend
- Bottone per accedere alla schermata degli allarmi
- Visualizzazione data e ora
- Rappresentazione grafica delle grandezze di regolazione
 - LED per la grandezza ON-OFF (uscita 1)
 - ProgressBar per la grandezza proporzionale (uscita 2)

Inserimento del grafico

Per inserire un grafico in una pagina è sufficiente selezionare ATCMgraph dalla lista degli oggetti ATCM plugin e trascinarlo nella pagina. Su ciascun grafico sono rappresentabili al più due tracce.

Per l'impostazione dei parametri iniziamo con le variabili da rappresentare:

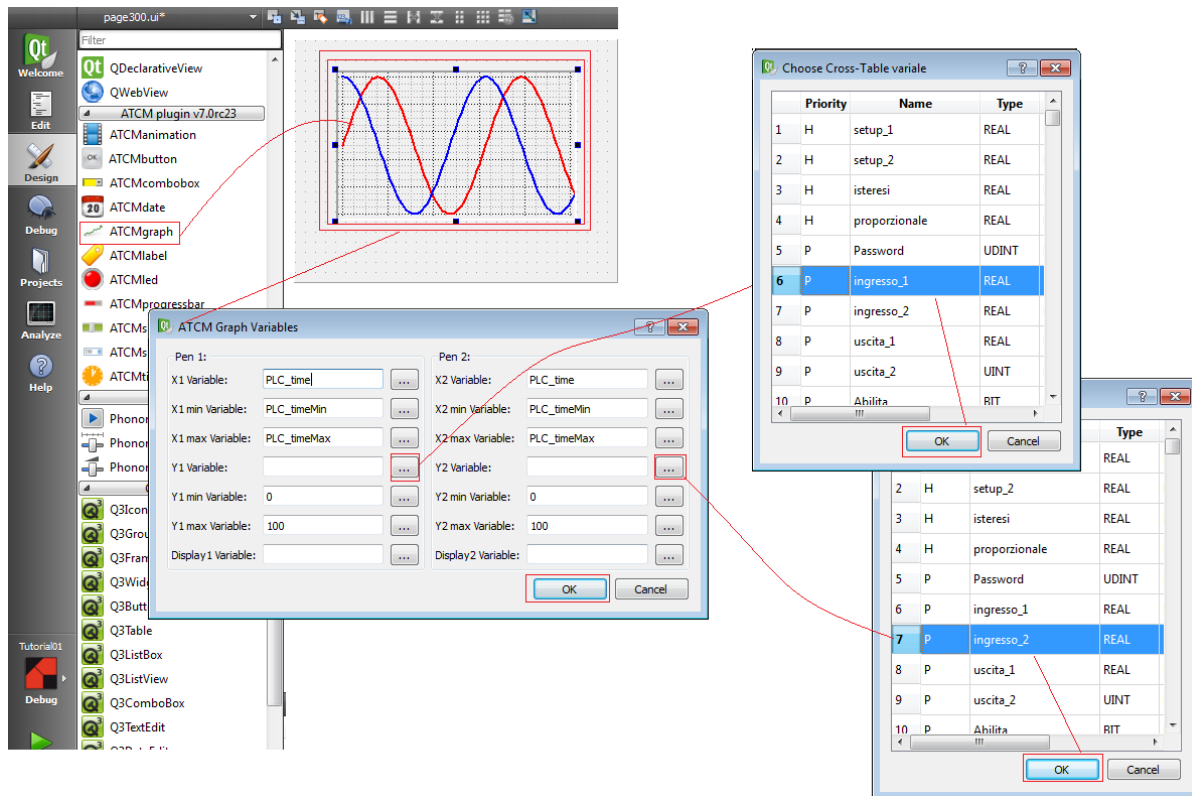
- Fare doppio click sul grafico
- Nella finestra che appare sono disponibili due insiemi di proprietà da impostare, una per ogni traccia rappresentabile. I parametri da impostare sono:
 - **Y Variable** che indica quale variabile si deve rappresentare sull'asse delle Y
 - **Ymin** e **Ymax** che indicano i limiti inferiore e superiore dell'asse Y (nel nostro progetto ingresso_1 per quanto riguarda la traccia 1 e ingresso_2 per quanto riguarda la traccia 2)

Per quanto riguarda invece l'asse dei tempi lasciando inalterati i parametri preimpostati il grafico viene aggiornato automaticamente.

Impostati i parametri premere **OK** nella finestra **ATCM Graph Variables**

Se si vuole invece modificare la dimensione della finestra di grafico, ed impostarla ad un determinato valore è possibile scrivere nella variabile **PLC_timeWin** la dimensione necessaria.

Per far ciò inserire una ATCMlabel ed associarla alla variabile PLC_timeWin.



Fatto ciò i grafici sono già utilizzabili ma è possibile impostare altri parametri per rappresentare meglio le tracce, vediamo alcuni parametri che può essere utile modificare:

- x1Step
- x1TickLabelVisible
- x1TicklabelFormat
- x1Label
- y1Step
- pen1Color

Modificate le proprietà del grafico, inseriamo le label che rappresentano numericamente i valori degli ingressi e delle uscite e i bottoni rimanenti:

- inseriamo la ATCMLabel per la lettura degli ingressi 1 e 2
- inseriamo le label di descrizione e ne modifichiamo il contenuto
- inseriamo il led per la rappresentazione dell'uscita ON-OFF
 - il led si comporta come una label quindi, una volta trascinato sulla pagina, facendo un doppio click, si apre la finestra per la selezione delle variabili, quindi scegliere la variabile uscita_1
- Inseriamo la ATCMprogressbar per la rappresentazione della variabile uscita_2
 - Analogamente alla ATCMLabel, una volta selezionata e trascinata sulla pagina, gli associamo la variabile
- Inseriamo un ATCMbutton per abilitare la comunicazione con il dispositivo remoto
- Inseriamo un ATCMbutton che sia visibile solamente quando c'è una condizione di allarme
- Inseriamo un ATCMbutton per accedere alla pagina degli storici

Inseriamo la gestione della storicizzazione dei dati.

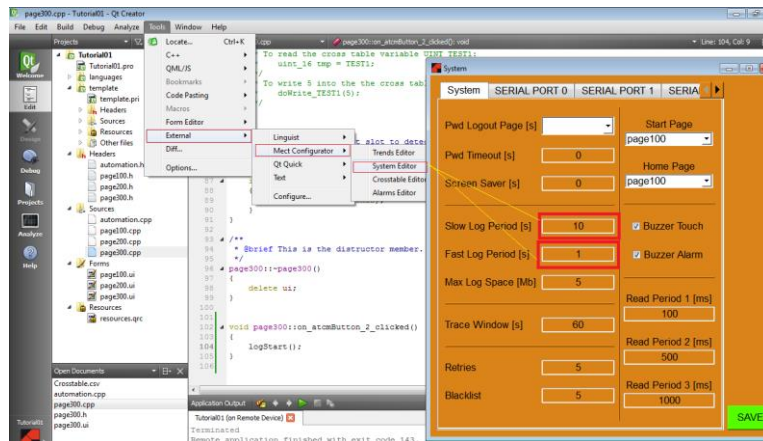
Storici

Per la gestione degli storici sono necessarie due cose:

- Dichiarare nella Crosstable le variabili di cui si vuole fare il log come S oppure F, dove S sta per slow mentre F equivale a fast.
- Aggiungere un bottone che quando premuto fa partire la registrazione e quando rilasciato la fa terminare

All'inizio del progetto quando abbiamo dichiarato le variabili input_1 e input_2 sono state definite come S, questo farà sì che una volta fatto partire il log le due variabili saranno salvate in memoria.

Per definire il tempo di campionamento delle variabili di tipo S o F è necessario aprire il programma **System editor** e da lì modificare i valori, se necessario.

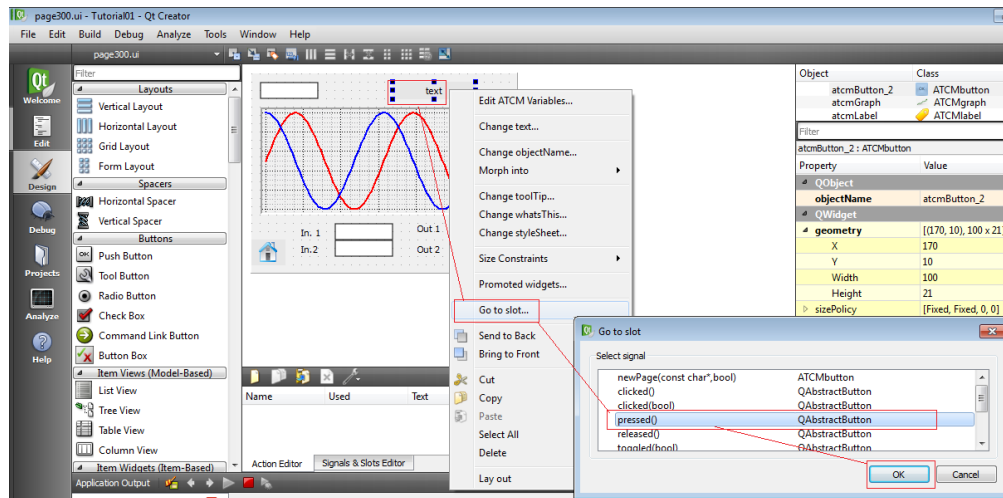


Ora inseriamo il bottone per l'avvio e la fermata dei log.

1. Inseriamo un ATCMbutton nella pagina nel modo già conosciuto;
2. Modifichiamo la proprietà del bottone **checkable** spuntando la casella;

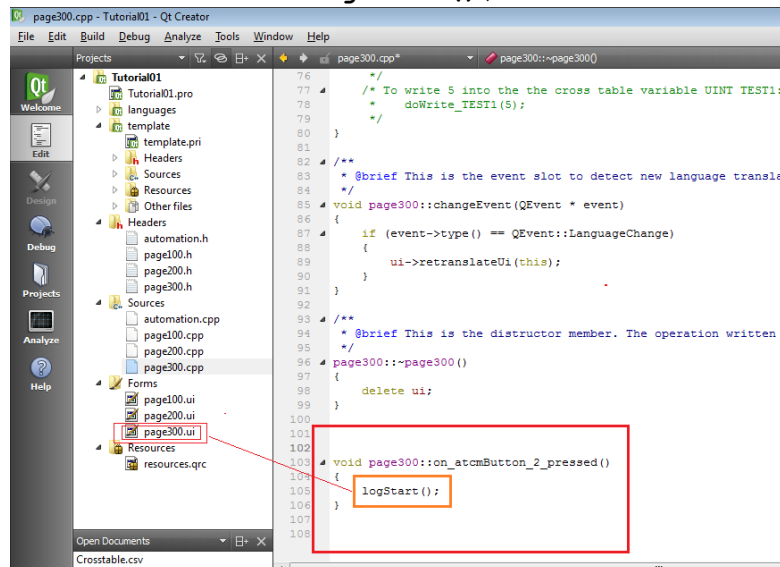
QAbstractButton	
iconSize	16 x 16
Width	16
Height	16
checkable	☒
checked	☐
autoRepeat	☐
autoRepeatDelay	300
autoRepeatInterval	100

3. Selezioniamo il bottone e premiamo il tasto destro
4. Nel menu che appare selezionare **Go to slot**
5. Selezionare la riga **pressed()**, quindi premere **OK**



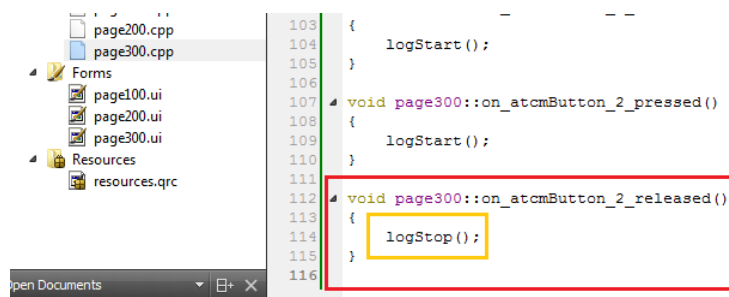
Dopo aver premuto OK si viene mandati su una finestra che contiene il codice sorgente della pagina. Nel punto dove è posizionato il cursore inserire la riga di codice:

logStart () ;



Abbiamo creato quindi la possibilità di far partire la registrazione dei dati alla pressione del tasto. Tornati nella pagina ripetere i passi da 3 e 4 per creare l'azione dello stop del log ad una nuova pressione del tasto. Arrivati qui selezionare **released()**, quindi nella finestra di codice inserire la linea

logStop () ;



Terminate queste azioni possiamo, una volta trasferito il programma sul pannello, acquisire e salvare i dati delle variabili richieste.

Visualizzazione degli storici

Per visualizzare sul pannello i dati acquisiti è sufficiente richiamare la pagina di sistema **store**. Come abbiamo visto ci sono alcune pagine di sistema che permettono di realizzare funzioni già preconfezionate.

Inseriamo perciò un **ATCMButton** che richiama la pagina **store**.

La pagina sul pannello si presenta nel modo seguente:

date	time	Test_ID	Test_RTU	Test_USB	IO_Errors	RTU_Errors	USBA_Errors	USBB_Errors	serverToClt
2025/05/09	14:07:44	0	0	0	0	0	0	0	1
2025/05/09	14:07:49	0	0	0	0	0	0	0	1
2025/05/09	14:07:54	0	0	0	0	0	0	0	1
2025/05/09	14:07:59	0	0	0	0	0	0	0	1
2025/05/09	14:08:04	0	0	0	0	0	0	0	1
2025/05/09	14:08:09	0	0	0	0	0	0	0	1
2025/05/09	14:08:14	0	0	0	0	0	0	0	1
2025/05/09	14:08:19	0	0	0	0	0	0	0	1
2025/05/09	14:08:24	0	0	0	0	0	0	0	1
2025/05/09	14:08:29	0	0	0	0	0	0	0	1
2025/05/09	14:08:34	0	0	0	0	0	0	0	1
2025/05/09	14:08:39	0	0	0	0	0	0	0	1
2025/05/09	14:08:44	0	0	0	0	0	0	0	1
2025/05/09	14:08:49	0	0	0	0	0	0	0	1
2025/05/09	14:08:54	0	0	0	0	0	0	0	1
2025/05/09	14:08:59	0	0	0	0	0	0	0	1
2025/05/09	14:09:04	0	0	0	0	0	0	0	1
2025/05/09	14:09:09	0	0	0	0	0	0	0	1

Oltre ad avere la visualizzazione dei dati acquisiti con il relativo istante temporale, sono presenti alcuni bottoni che effettuano azioni specifiche:

- I bottoni   riportano rispettivamente alle pagine di HOME e BACK;
- I bottoni     permettono di navigare all'interno dei dati;
- Il bottone  si attiva quando è inserita una chiavetta USB e permette il salvataggio dei dati sul supporto esterno;
- Il bottone  manda alla pagina che permette il filtraggio dei dati e quindi di visualizzare solo quelli di un preciso periodo.

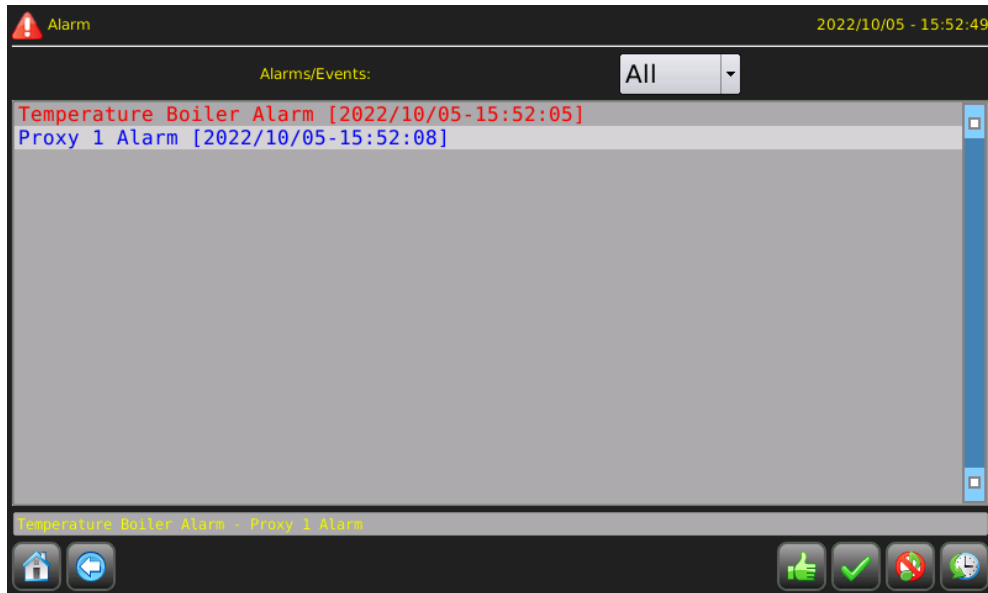
Allarmi

Per vedere lo stato di allarmi/eventi sono disponibili due pagine di libreria:

- La pagina “**alarms**” implementa la visualizzazione degli allarmi.
- La pagina “**alarms_history**” implementa la visualizzazione dello storico degli allarmi.

Scriviamo **alarms** nella proprietà **pageName** del ATCMbutton relativo alla variabile “Allarme”

Pagina alarms



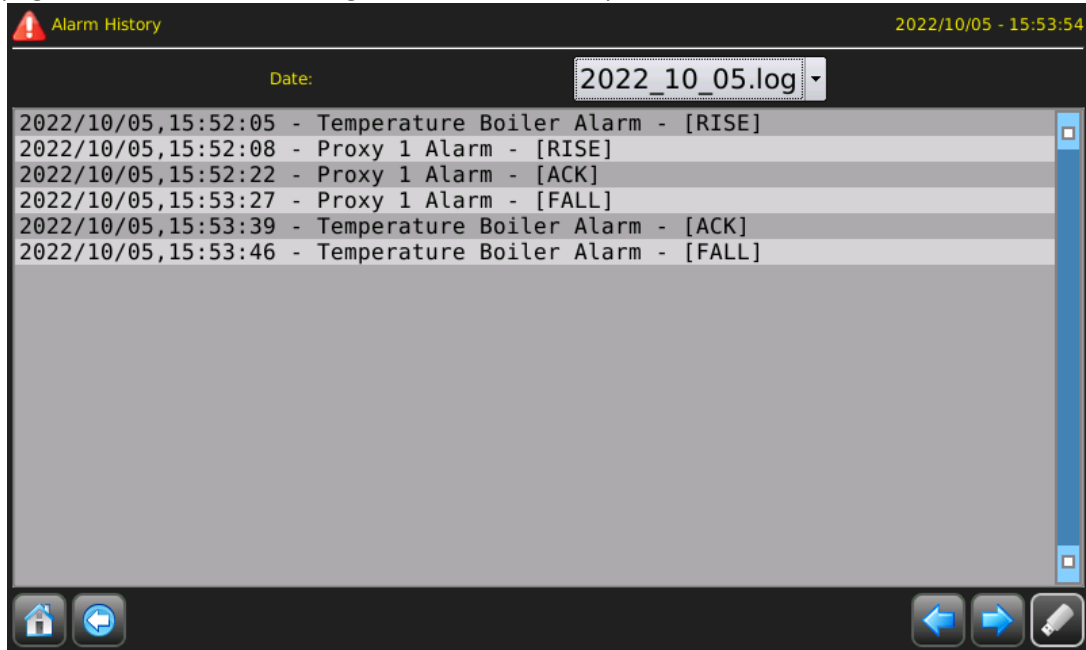
In alto, sono presenti 2 combobox nelle quali si può selezionare il livello (1 più alto 32 più basso) e il tipo (All, Alarm e Event). Queste selezioni saranno usate come filtri per gli allarmi visualizzati nella lista scorrevole sottostante. In questo oggetto, sono elencati gli allarmi attualmente visibili. Subito sotto è presente un banner che mostra a turno i vari allarmi visibili attualmente.

Sul fondo, accanto ai soliti tasti di Home e Back ci sono anche i seguenti tasti:

1.  tasto per fare il riconoscimento di un allarme precedentemente selezionato dalla lista scorrevole soprastante;
2.  tasto per fare il riconoscimento di tutti gli allarmi visibili;
3.  tasto per disabilitare il suono dell'allarme;
4.  tasto per passare alla pagina dello storico degli allarmi.

Pagina alarms_history

In questa pagina sono visualizzati tutti gli allarmi avvenuti in precedenza.



E' possibile filtrarli per giorno, livello o per tipo (allarmi/eventi) usando gli appositi menù a tendina. Inoltre è possibile scaricare i dati su penna USB premendo il bottone  che si attiverà all'inserzione di una penna USB.

A questo punto abbiamo terminato la grafica, trasferiamo il progetto sul pannello.

Trasferimento progetto HMI sul pannello operatore

Il trasferimento del progetto avviene semplicemente premendo il tasto  dopo aver collegato il pannello al personal computer. Terminato il trasferimento il pannello si riavvia automaticamente con la nuova interfaccia.

T

trasferito il progetto HMI iniziamo con il PLC!

ATCM control

Il programma per la realizzazione del PLC si chiama ATCMcontrol; è un editor per PLC che supporta i linguaggi IEC-61131-3.

Progetto PLC

Una volta creato il progetto si genera anche il corrispondente progetto PLC. Per lanciare il programma basta

premere il tasto  nel cross table editor. Il progetto PLC è preformato con alcuni task e programmi associati ai task, inoltre le variabili create in cross table sono già automaticamente presenti tra le variabili globali del PLC.

MECT Suite corso base

```
VAR_GLOBAL
setup_1 AT %QD0.4:REAL;
setup_2 AT %QD0.8:REAL;
isteresi AT %QD0.12:REAL;
proporzionale AT %QD0.16:REAL;
Password AT %QD0.20:UDINT;
Tempociclo1 AT %QW0.24:UINT;
Tempociclo2 AT %QW0.28:UINT;
ingresso_1 AT %ID0.772:REAL;
ingresso_2 AT %ID0.776:REAL;
uscita_1 AT %QW0.780:UINT;
uscita_2 AT %QW0.784:UINT;
Abilita AT %QX0.788.0:BOOL;
RTU0_TYPE_PORT AT %ID0.20000:UDINT;
RTU0_BAUDRATE AT %ID0.20004:UDINT;
RTU0_STATUS AT %ID0.20008:UDINT;
RTU0_READS AT %ID0.20012:UDINT;

...

PLC_DigIn_1 AT %IX0.21200.0:BOOL;
PLC_DigIn_2 AT %IX0.21204.0:BOOL;
PLC_DigIn_3 AT %IX0.21208.0:BOOL;
PLC_DigIn_4 AT %IX0.21212.0:BOOL;
PLC_DigIn_5 AT %IX0.21216.0:BOOL;
PLC_DigIn_6 AT %IX0.21220.0:BOOL;
PLC_DigIn_7 AT %IX0.21224.0:BOOL;
PLC_DigIn_8 AT %IX0.21228.0:BOOL;
PLC_Reserved_1 AT %IX0.21232.0:BOOL;
PLC_Reserved_2 AT %IX0.21236.0:BOOL;
PLC_Reserved_3 AT %IX0.21240.0:BOOL;
PLC_DigOut_1 AT %QX0.21244.0:BOOL;
PLC_DigOut_2 AT %QX0.21248.0:BOOL;
PLC_DigOut_3 AT %QX0.21252.0:BOOL;
PLC_DigOut_4 AT %QX0.21256.0:BOOL;
PLC_DigOut_5 AT %QX0.21260.0:BOOL;
PLC_DigOut_6 AT %QX0.21264.0:BOOL;
PLC_DigOut_7 AT %QX0.21268.0:BOOL;
PLC_DigOut_8 AT %QX0.21272.0:BOOL;
PLC_DigOut_9 AT %QX0.21276.0:BOOL;
PLC_DigOut_10 AT %QX0.21280.0:BOOL;
PLC_DigOut_11 AT %QX0.21284.0:BOOL;
PLC_DigOut_12 AT %QX0.21288.0:BOOL;
PLC_AnIn_1 AT %IW0.21292:INT;
PLC_AnIn_2 AT %IW0.21296:INT;
PLC_AnIn_3 AT %IW0.21300:INT;
PLC_AnIn_4 AT %IW0.21304:INT;
PLC_Tamb AT %IW0.21308:INT;
PLC_EncoderLo AT %IW0.21312:UINT;
PLC_EncoderHi AT %IW0.21316:UINT;
PLC_StatusReg AT %IW0.21320:UINT;
PLC_RPM AT %IW0.21324:UINT;
PLC_AnOut_1 AT %QW0.21328:INT;
PLC_AnOut_2 AT %QW0.21332:INT;
PLC_AnInConf AT %QW0.21336:UINT;
PLC_AnOutConf AT %QW0.21340:UINT;
PLC_ControlWord AT %QW0.21344:UINT;
PLC_EncPreset_Lo AT %QW0.21348:UINT;
PLC_EncPreset_Hi AT %QW0.21352:UINT;
CH0_NETRUN AT %IX0.21368.0:BOOL;
CH0_NETGOOD AT %IX0.21372.0:BOOL;
CH0_NETERR AT %ID0.21376:UDINT;
CH0_NETRST AT %QX0.21380.0:BOOL;
CH0_NETDIS AT %QX0.21384.0:BOOL;
CH0_01_NODERUN AT %IX0.21388.0:BOOL;
CH0_01_NODEGOOD AT %IX0.21392.0:BOOL;
CH0_01_NODEERR AT %ID0.21396:UDINT;
CH0_01_NODERST AT %QX0.21400.0:BOOL;
CH0_01_NODEDIS AT %QX0.21404.0:BOOL;
PLC_time AT %ID0.21560:REAL;
PLC_timeMin AT %ID0.21564:REAL;
PLC_timeMax AT %ID0.21568:REAL;
PLC_timeWin AT %QD0.21572:REAL;
PLC_Version AT %IW0.21576:UINT;
PLC_EngineStatus AT %IW0.21580:UINT;
PLC_ResetValues AT %QX0.21584.0:BOOL;
END_VAR
```

Come possiamo notare oltre alle variabili che abbiamo definito sono presenti nel file anche quelle di sistema e quelle di ingressi/uscite presenti sul pannello operatore.

Se durante lo svolgimento del progetto dovessimo aggiungere o togliere una variabile dalla Crosstable, è sufficiente ripetere i passi appena descritti in questo paragrafo.

A questo punto dobbiamo realizzare l'automazione. Come detto il programma permette di scrivere l'automazione utilizzando diversi linguaggi.

Il template di MECT parte dal linguaggio ST come programma principale.

Nel nostro progetto partiremo dal programma principale e ci costruiremo dei blocchi funzione in FBD e in ST.

Analisi del progetto PLC

Per prima cosa analizziamo cosa dobbiamo realizzare e quindi vediamo come farlo.

Il PLC deve effettuare due regolazioni, una su `ingresso_1` e `uscita_1`, la seconda su `ingresso_2` e `uscita_2`.

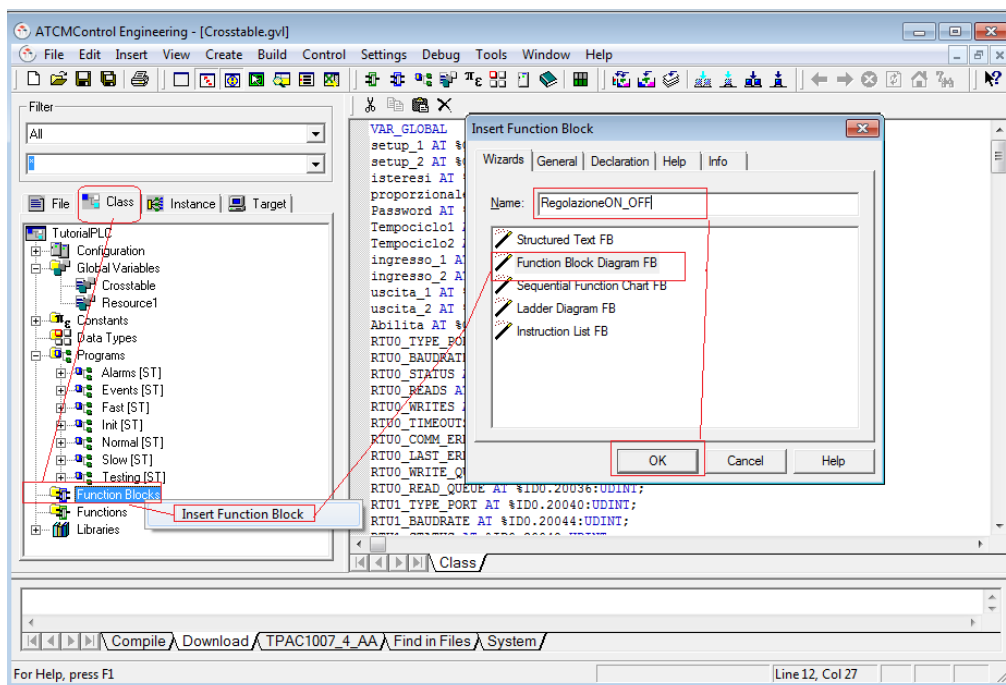
Abilitiamo le regolazioni a seguito di un contatto in ingresso che associamo alla variabile di sistema:

`PLC_DigIn_1`

Regolazione 1: FBD

Per effettuare la regolazione di tipo ON-OFF creiamo un blocco funzione di tipo LD:

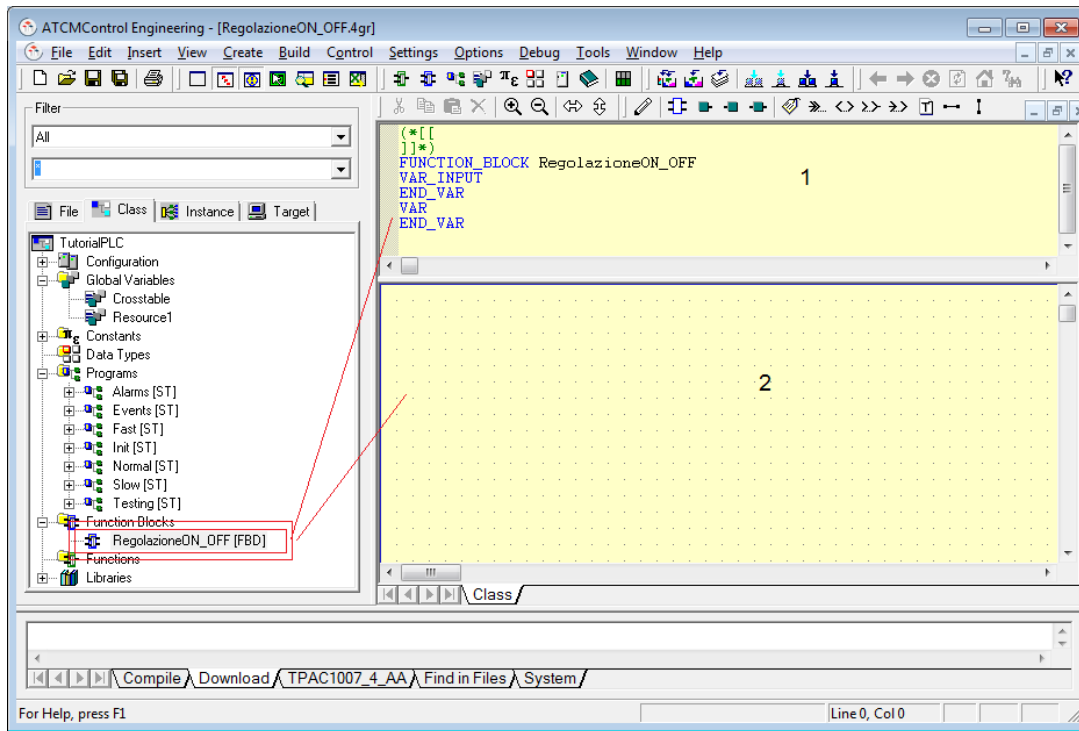
- Selezionare Class
- Selezionare nell'albero **Function Blocks** e premere il tasto destro del mouse
- Selezionare e premere **Insert Function Block**
- Nella finestra che appare selezionare **Ladder Diagram FB**
- Inserire il nome che si vuol dare al blocco funzione (per esempio: `RegolazioneON_OFF`)
- Premere **OK**



Il blocco funzione è così stato creato, infatti espandendo il ramo **Function Blocks** dell'albero del progetto vediamo il ramo.

Per entrare nel blocco funzione bisogna selezionare e fare doppio click sul nome del blocco funzione. La finestra che appare è composta da due zone:

- Zona 1 di definizione delle variabili
- Zona 2 di codice



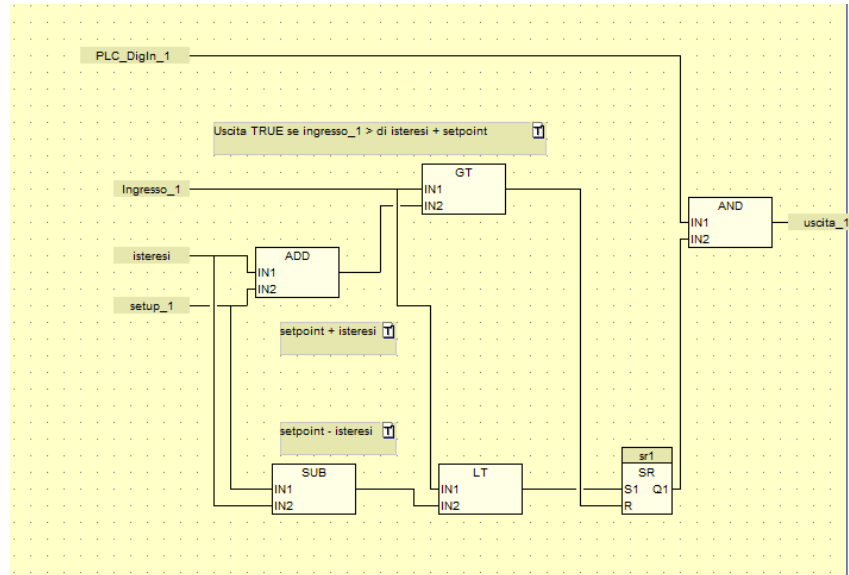
Come primo passo facciamo in modo che il blocco funzione veda le variabili definite in Crosstable. Per far ciò posizioniamoci con il mouse nella zona1 (che cambia colore di sfondo), subito dopo il nome del blocco funzione, ed inseriamo la scritta: `#import "Crosstable.gvl"` come si può vedere in figura.

```
(*[[
]])*)
FUNCTION_BLOCK RegolazioneON_OFF
#import "Crosstable.gvl"
VAR_INPUT
END_VAR
VAR
END_VAR
```

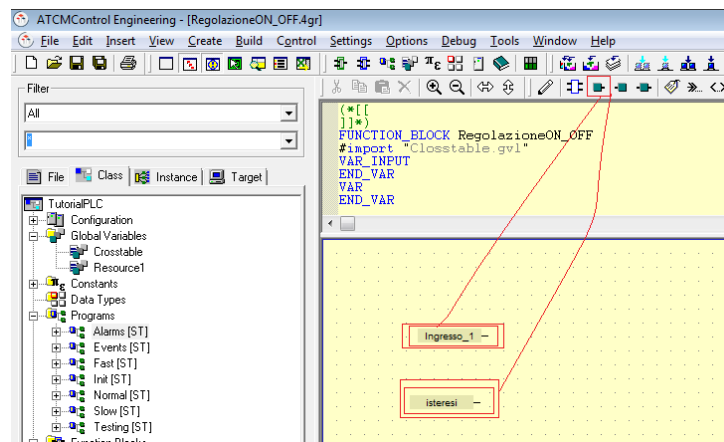
A questo punto posizioniamoci sulla zona 2 per inserire il codice di programma.

La regolazione dovrebbe in pratica fare in modo che quando `ingresso_1` supera il valore della variabile `setup_1 + isteresi` spegne `uscita_1`, che si riaccende quando `ingresso_1` scende sotto `setup_1 - isteresi`.

Lo schema del blocco funzione risulta come mostrato in figura:



- Inseriamo come primo passo le variabili di cui abbiamo necessità trascinando l'icona **Insert Var Read** nella finestra di zona 2



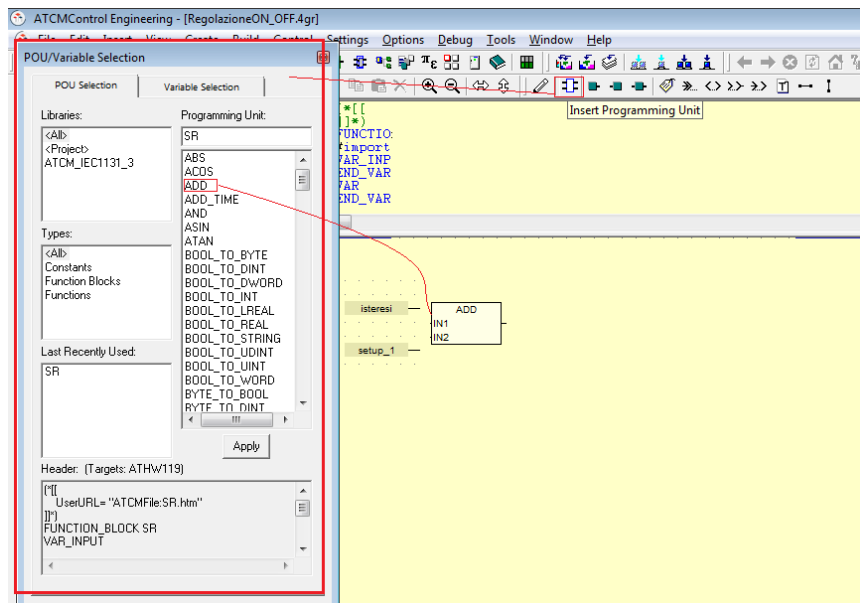
- Facciamo un doppio click sulla variabile immessa nella finestra e modifichiamone il nome inserendo quello della variabile alla quale vogliamo associarla, per esempio `ingresso_1` o `isteresi`.

Inserite le variabili, inseriamo gli operatori.

Se selezioniamo l'icona **Insert Programming Unit** apparirà la finestra dalla quale è possibile scegliere gli operatori di libreria. Selezionando e trascinando l'operatore necessario (per esempio l'operatore ADD che fa la somma tra due variabili), lo si porta nella zona 2.

Per collegare le variabili all'operatore selezionare l'icona, a questo punto è sufficiente effettuare i collegamenti come in un normale editor per schemi elettrici.

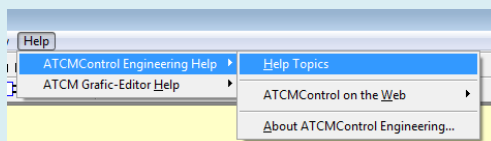
Il risultato ottenuto è quello mostrato in precedenza



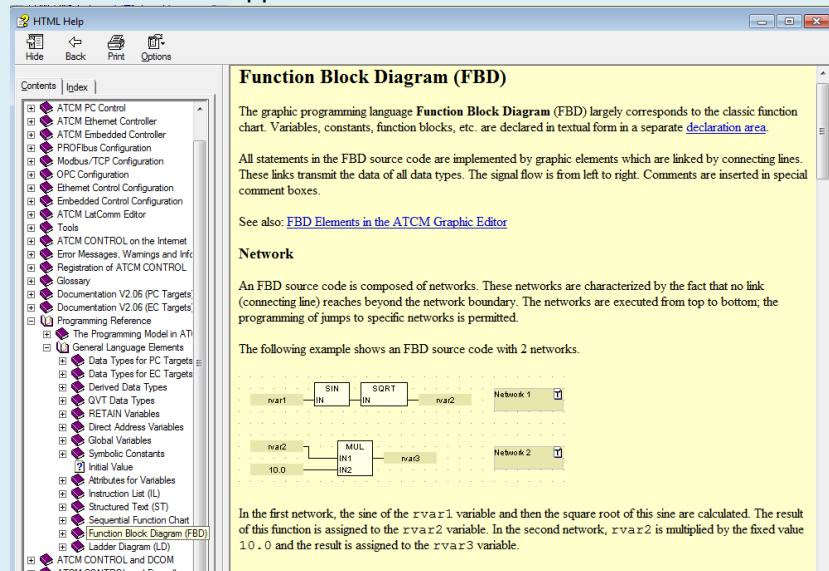
Help di ATCMControl

Per i dettagli sull'utilizzo del linguaggio FBD di ATCMcontrol è presente un help in linea (in inglese) che spiega come utilizzare i vari elementi e operatori.

Per raggiungere Help di ATCMControl dal menu selezionare:
Help, ATCMControl Engineering Help; Help Topics



Nella finestra che appare selezionare "Indice" e scrivere "FBD"



Regolazione 2: ST

Per effettuare la regolazione proporzionale invece usiamo il linguaggio ST.

Creiamo anche questa volta un blocco funzione nello stesso modo visto in precedenza, solo che invece di selezionare **Functional Block**, selezioniamo **Structured Text**. Anche con questo linguaggio come primo passo facciamo l'import delle variabili della Crosstable.

```
(*[[
]])*)
FUNCTION_BLOCK Regolazione_Prop
#import "Crosstable.gv1"
VAR_INPUT
END_VAR
VAR
END_VAR

;
END_FUNCTION_BLOCK
```

La regolazione che effettua questo blocco funzione è di tipo proporzionale, in pratica si moltiplica per la costante `proporzionale`, il valore di differenza tra `setpoint_2` e `ingresso_2`. Il risultato del calcolo viene inviato sull'`uscita_2`.

Il codice è semplicemente così composto:

- Inseriamo una variabile di appoggio che chiamiamo `uscita`, di tipo REAL tra le espressioni VAR e END_VAR.
- Scriviamo il codice

```
(*[[
]])*)
FUNCTION_BLOCK Regolazione_Prop
#import "Crosstable.gv1"
VAR_INPUT
END_VAR
VAR
    uscita:REAL;
END_VAR

uscita:= proporzionale*(setup_2 - ingresso_2);
IF uscita > 1000.0 THEN
    uscita_2:=1000.0;
ELSIF uscita < 0.0 THEN
    uscita_2:=0.0;
ELSE
    uscita_2:=uscita;
END_IF;
END_FUNCTION_BLOCK
```

Assegnazione ingressi/uscite

Finora abbiamo utilizzato variabili interne come uscita_1 ingresso_2 ecc., ora vogliamo assegnare queste variabili agli effettivi ingressi e uscite fisiche del dispositivo.

Facciamo la corrispondenza tra variabile fisica e variabile interna:

ingresso_1	PLC_AnIn_1	Ingresso analogico 1 del pannello operatore
ingresso_2	PLC_AnIn_2	Ingresso analogico 2 del pannello operatore
uscita_2	PLC_AnOut_1	Uscita analogica 1 del pannello operatore

Per quanto riguarda l'uscita_1, anche se utilizziamo il dispositivo remoto TPLC005 la scrittura sulla variabile è automaticamente trasferita sull'uscita del dispositivo. In sostanza non è necessario utilizzare una funzione per scrivere su un dispositivo remoto ma è sufficiente impostare la variabile.

Fatte queste associazioni logiche scriviamo il codice per effettuarle. Il codice lo scriviamo nel programma principale **Normal**.

Per entrare nel programma Normal è sufficiente selezionarlo dall'albero dei progetti e fare un doppio click. La pagina che viene mostrata è quella di codice.

Inseriamo innanzitutto le variabili nel seguente modo:

```
ingresso_1:=INT_TO_REAL(PLC_AnIn_1);
ingresso_2:=INT_TO_REAL(PLC_AnIn_2);
PLC_AnOut_1:=REAL_TO_INT(uscita_2);
```

Nel programma dobbiamo poi metterci in grado di utilizzare i blocchi funzione che abbiamo realizzato. Per far ciò prima dobbiamo dichiarare i blocchi funzione esattamente come una qualsiasi variabile, usando per esempio, la seguente sintassi:

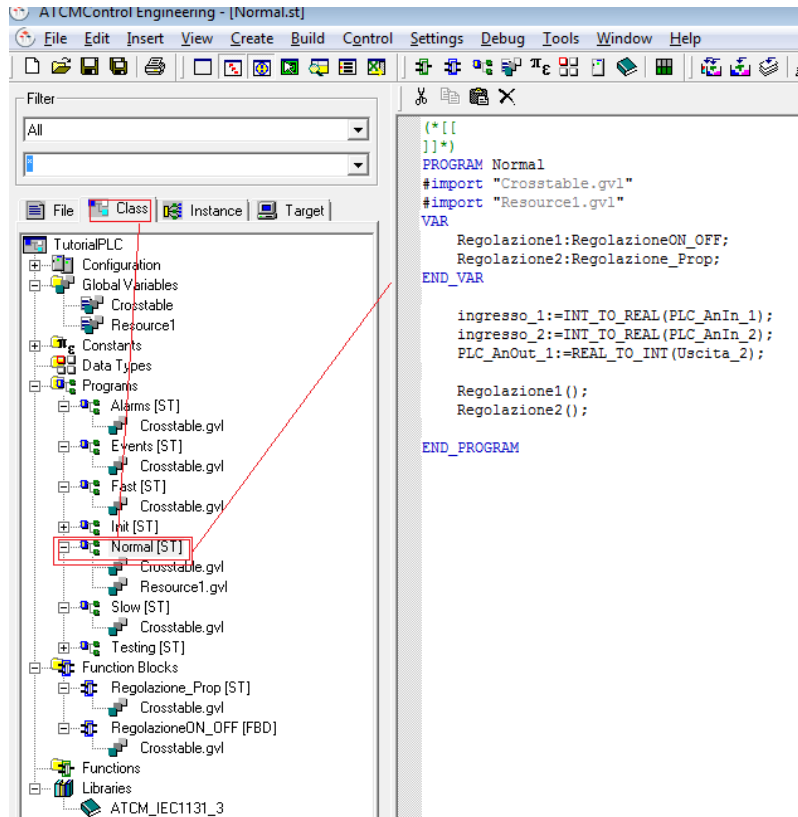
Regolazione1: Regolazione ON_OFF;

Infine inseriamo il codice per l'esecuzione del programma che risulta come riportato di sotto.

```
(*[[
]])*)
PROGRAM Normal
#import "Crosstable.gvl"
#import "Resource1.gvl"
VAR
    Regolazione1:RegolazioneON_OFF;
    Regolazione2:Regolazione_Prop;
END_VAR

    ingresso_1:=INT_TO_REAL(PLC_AnIn_1);
    ingresso_2:=INT_TO_REAL(PLC_AnIn_2);
    PLC_AnOut_1:=REAL_TO_INT(Uscita_2);

    Regolazione1();
    Regolazione2();
END_PROGRAM
```



Inizializzazioni

L'ultimo passo prima della compilazione è quello di inizializzare le variabili alla partenza.

Per esempio dobbiamo inizializzare le variabili per definire di che tipo sono gli ingressi e le uscite che andiamo ad utilizzare. Infatti poiché gli ingressi possono accettare diversi tipi di sonde definiamo quali sono quelle usate nel nostro progetto.

Le inizializzazioni sono effettuate nel programma **Init** al quale si accede nella stessa maniera di come si è entrati nel programma **Normal**:

- lo si seleziona dall'albero dei progetti
- si effettua un doppio click con il mouse

Le inizializzazioni da effettuare sono le seguenti:

- configurazione ingresso analogico 1
- configurazione ingresso analogico 2
- configurazione uscita analogica 2

Il codice per effettuare queste inizializzazioni è il seguente:

```
(*[[
]])*)
PROGRAM Init
#import "Crosstable.gv1"
#import "Resource1.gv1"
VAR
END_VAR
    PLC_AnInConf_1 := 2;
    PLC_AnInConf_2 := 2;
    PLC_AnOutConf_1 := 2;
END_PROGRAM
```

I numeri da assegnare alle variabili sono descritti nel manuale del dispositivo. Nel nostro esempio abbiamo tutte le analogiche che sono in tensione.

Terminati questi passi è possibile compilare il programma e quindi trasferirlo al pannello operatore.

Appendice 1 ATCM plugins

ATCM Plugin

Gli ATCM plugin sono undici oggetti che possono essere utilizzati per la costruzione delle pagine grafiche:

- ATCManimation
- ATCMbutton
- ATCMcombobox
- ATCMdate
- ATCMgraph
- ATCMlabel
- ATCMled
- ATCMprogressbar
- ATCMslider
- ATCMspinbox
- ATCMtime