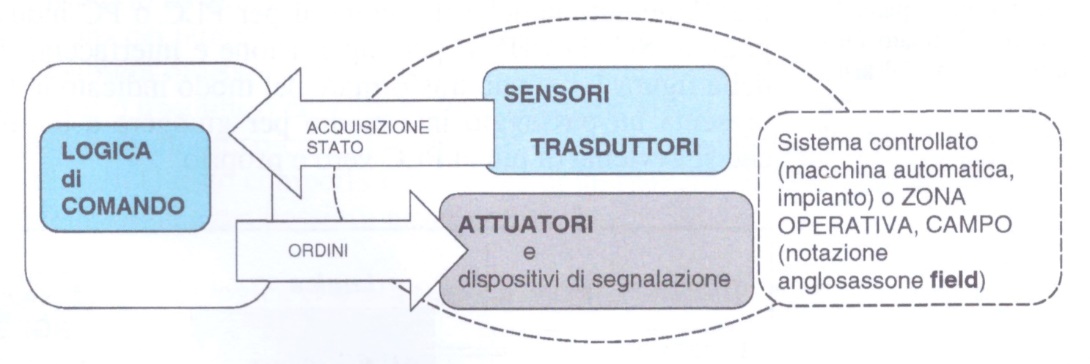
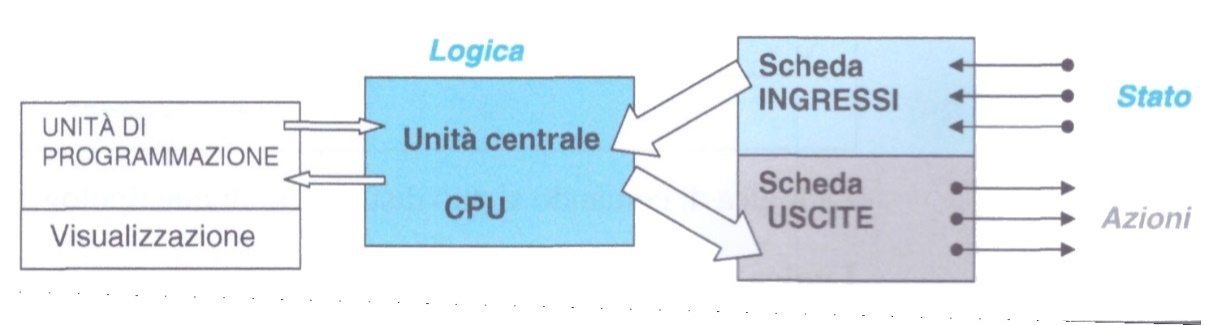
MANUALE TIA PORTAL

Dalla configurazione del progetto alla programmazione avanzata PLC /HMI

PLC

Il PLC (controllore logico programmabile) è il più diffuso dispositivo usato nell'automazione industriale, si tratta di un apparato progettato e costruito per poter operare in condizioni elettriche anche ostili (24/24 h in condizioni critiche di temperatura, umidità e sbalzi di tensione). Trattandosi di un elaboratore a tutti gli effetti, il PLC è costituito da una parte hardware formata da cavi e altri oggetti fisici e da una parte software costituita, dalle istruzioni che formano il programma che deve essere eseguito.   
Esso, acquisisce i segnali d'ingresso che gli giungono dai sensori di campo e in base al loro valore e a quanto stabilito dal programma, emette i necessari segnali di uscita indirizzandoli verso gli attuatori.





Architettura dei PLC

Un controllore a logica programmabile è costituito da un’unità centrale e da un insieme di moduli periferici. Tali moduli sono montati su guide profilate e collegati attraverso appositi cavi o bus1. Generalmente un sistema PLC si compone di:

• Modulo processore

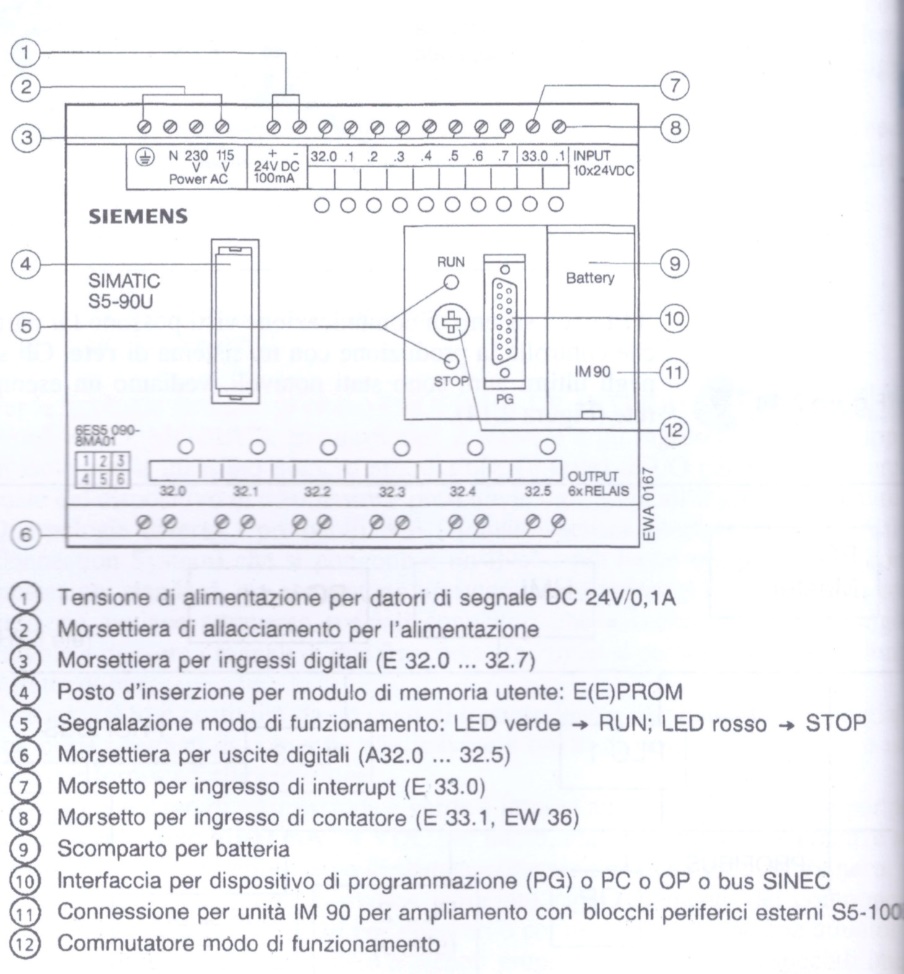
• Moduli di input/output

• Modulo alimentatore

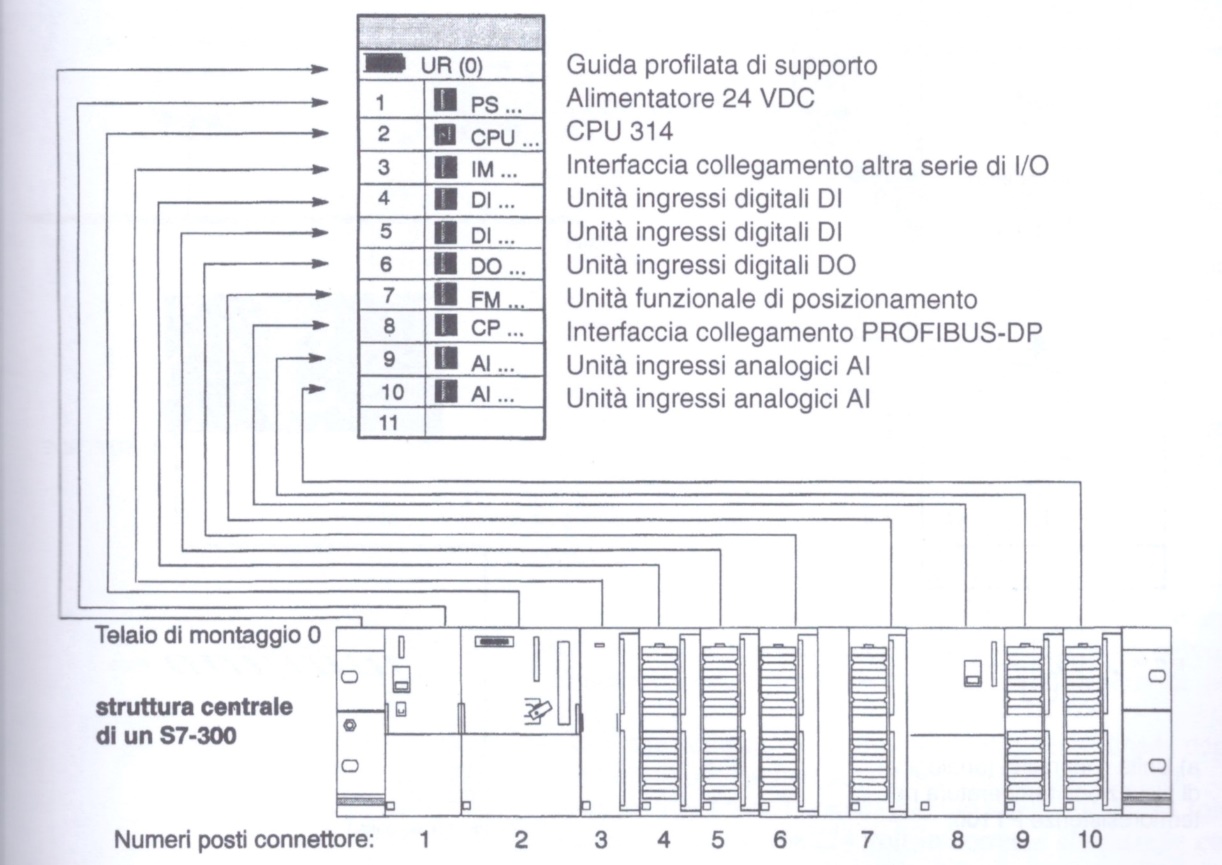
Questi componenti possono essere disposti in modo differente a seconda della tipologia del PLC, esistono sistemi che contengono tutto all’interno di un unico blocco e sistemi che sono composti da più blocchi.

In entrambi i casi i corpi principali possono essere espansi per aggiungere delle funzionalità utilizzando appositi moduli, questi moduli possono anche non trovarsi direttamente a contatto con il corpo principale ma possono essere dei moduli remoti.

PLC con tutti i componenti in un unico blocco (Es: 1200 Siemens)



PLC con più moduli (es: 1500 Siemens)

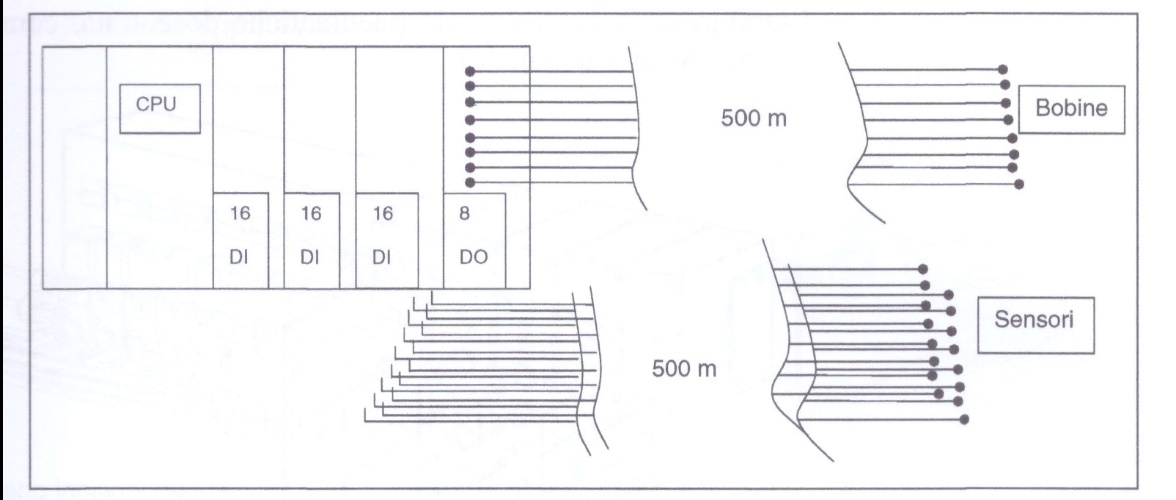


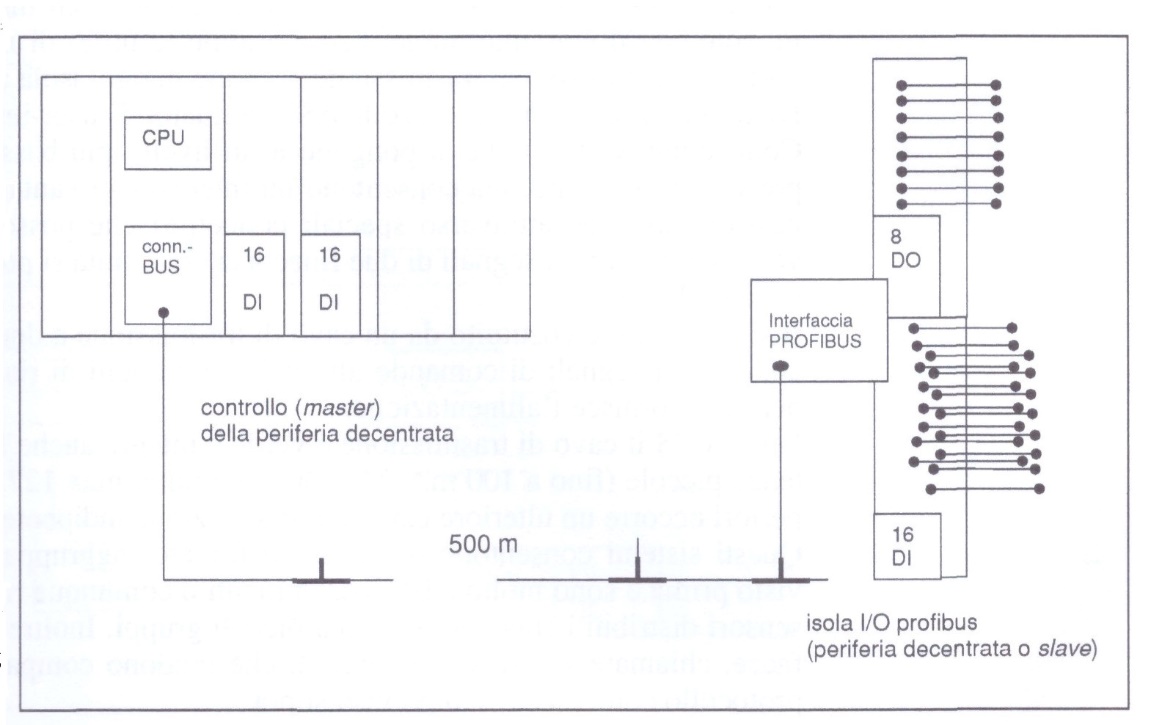
Oltre alle schede necessarie per il funzionamento di base esistono anche altre tipi di schede che devono essere utilizzare per scopi specifici come:

* schede di conteggio veloce.
* schede dedicate a trasduttori particolari, ad esempio le schede per i trasduttori di temperatura.
* Schede di comunicazione
* Schede dedicate al posizionamento assi
* Schede dedicate alla gestione di camme elettroniche

Come detto in precedenza i moduli possono trovarsi anche a grande distanza dal corpo principale,

questi tipi di moduli vengono utilizzati quando si hanno trasduttori o dispositivi di comando molto distanti dal quadro elettrico principale e utilizzando questi moduli remoti si limita l’utilizzo di cavo e si semplifica la ricerca guasti.





Modulo processore

La CPU (Central Processing Unit) `e il cuore del PLC: racchiude una scheda con uno o più microprocessori che eseguono il sistema operativo e i programmi sviluppati dall’utente e una memoria dove questi programmi sono salvati. Il modulo dispone anche di una memoria RAM per i dati e di una batteria di backup, data la sua volatilità.

La memoria

La memoria del PLC è tipicamente costituita da dispositivi a semiconduttore di diverso genere a seconda della funzione che svolgono:

* RAM (Random Access Memory): è un tipo di memoria ad accesso veloce e riscrivibile ma in caso di mancanza di tensione il contenuto viene perso (questo rende la RAM una memoria volatile).
* ROM (Read Only Memory): è una memoria di sola lettura che non può essere modificata, non è volatile e non è riscrivibile
* EPROM (Erasable Programmable Read Only Memory): è una memoria di sola lettura che non necessita di alimentazione. Può essere cancellata solo completamente usando una tecnica a raggi ultravioletti (scrittore di PROM) e riscritta elettricamente.
* EEPROM (Erasable Programmable Read Only Memory): è una memoria di tipo ROM che non necessita di alimentazione e può essere riscritta e cancellata elettricamente a piacere; quindi è la memoria perfetta per contenere un programma.

In particolare essa può essere suddivisa in:

* Memoria di sistema (ROM): La memoria di sistema contiene tutte le informazioni che servono per la gestione ed il controllo del funzionamento della CPU, esse costituiscono il sistema operativo (firmware) del PLC.

Data la primaria importanza di tale contenuto, i costruttori utilizzano generalmente memorie di sistema di tipo ROM in modo da evitarne l’accidentale cancellazione.

* Memoria di programma: Contiene il programma utente da quando viene trasferito al PLC. Non può essere una memoria volatile perché ovviamente questi dati non possono essere persi; quindi potrà essere o una RAM tamponata dalla batteria oppure una EPROM o EEPROM.
* Memoria di sistema: In questa sezione della memoria vengono memorizzati i risultati temporanei delle elaborazioni, i contenuti dei contatori, dei temporizzatori, dei bit interni che rappresentano lo stato logico degli ingressi e delle uscite.

Tale memoria è necessariamente realizzata con memorie RAM o FLASH e, durante il funzionamento normale del controllore, è accessibile solo dal processore.

Per favorire la messa a punto del software di controllo molti produttori danno la possibilità di attivare una modalità di lavoro nella quale all’utente è consentito di accedere alla memoria dati per simulare lo stato del sistema e verificare il corretto funzionamento del programma.

Struttura memoria di sistema

Come detto prima all’interno della memoria di sistema sono presenti vari tipi di dati.

Abbiamo un’area di memoria che è dedicata alla lettura degli ingressi e alla scrittura delle uscite.

Prendiamo come esempio un pulsante collegato all’ingresso 0.0, all’interno del PLC il suo segnale verrà collocato all’interno dell’area di memoria I0.0, mentre una lampadina collegata all’uscita 1.2 verrà collocata nell’area di memoria Q1.2.

Oltre all’area di memoria riservata alla parte hardware abbiamo a disposizione un’area di memoria “virtuale” presente all’interno del PLC.

Questa area di memoria è divisa in bit e può essere raggruppata in Byte e in Word.

Il bit è l’unità più basica all’interno dell’informatica e può rappresentare solo 2 valori, 1 e 0.

(le memorie di ingresso e uscite sono bit)

Il Byte contiene 8 bit, e questo permette al byte di rappresentare più valori rispetto a un bit.

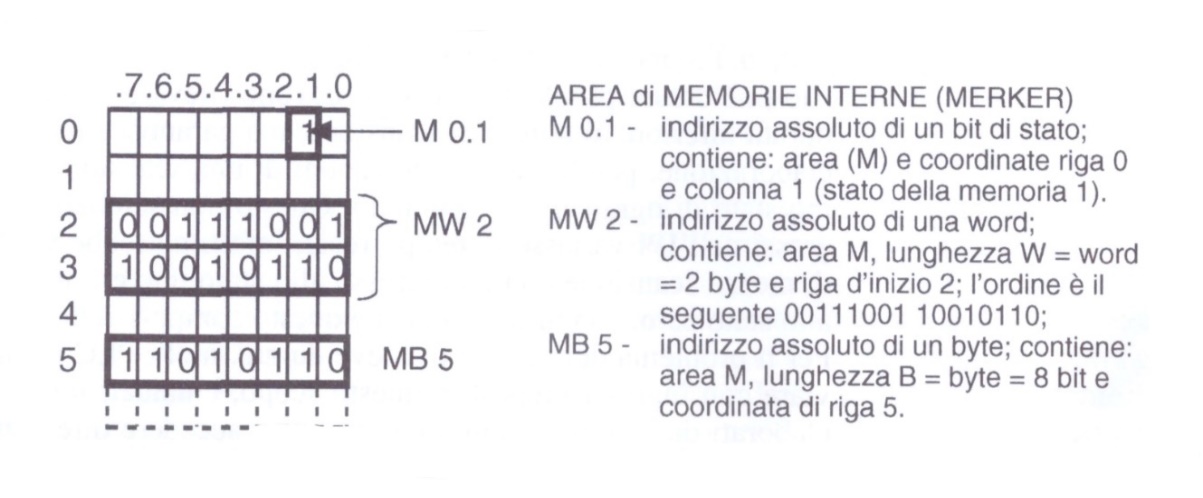
Il valore massimo che può rappresentare è 255 (partendo da 0) oppure può partire da -128 ed arrivare a 127 , questo perché come abbiamo detto prima l’unita di base è il bit che ha solo 2 valori, quindi il sistema di numerazione che si utilizza è in base 2.

Es: 255 = 11111111

1\* 2^7 + 1\* 2^6 + 1\* 2^5 + 1\* 2^4 + 1\* 2^3 + 1\* 2^2 + 1\* 2^1 + 1\* 2^0

La Word è un’area di memoria ancora più grande ed è composta da 2 byte, cioè 16 bit, il valore massimo che può rappresentare è 65535 (partendo da 0) oppure può partire da – 32768 ed arrivare a 32767.

Durante le esercitazioni andremo a vedere quando usare queste diverse aree di memorie, possiamo già anticipare qui che quando andremo ad utilizzare un sensore analogico, e quindi non ci basterà più ragionare con valore 0 o 1, andremo ad utilizzare una Word, visto che si tratta di un ingresso in questo caso andremo a lavorare sull’area di memoria degli ingressi ---> IW

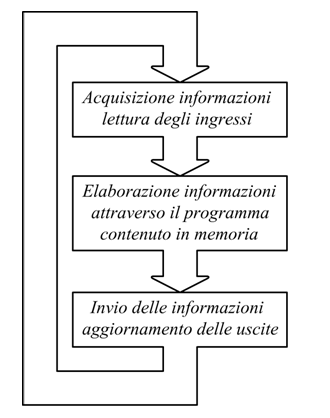


Ciclo programma

La particolarità che distingue il funzionamento del PLC da quello degli altri dispositivi a microprocessore `e la ciclicità con cui il programma viene eseguito.

Durante ogni ciclo il sistema esegue una sequenza di azioni:

• legge gli ingressi;

• esegue il programma utente;

• scrive i valori sulle uscite;

Il Ciclo programma può avere un tempo fisso definito dall’utente oppure il tempo ciclo può essere variabile perché viene definito ciclo per ciclo dal PLC in base al carico di lavoro che deve elaborare.

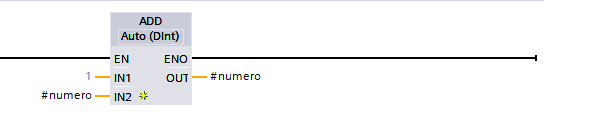
Siemens principalmente utilizza il secondo metodo ma vedremo come impostare manualmente il tempo ciclo.

In ogni PLC è presente il “Whatchdog timer” cioè un timer che monitora la durate del ciclo, se il programma richiede

più tempo di quello che è stato a disposizione il PLC andrà in blocco.

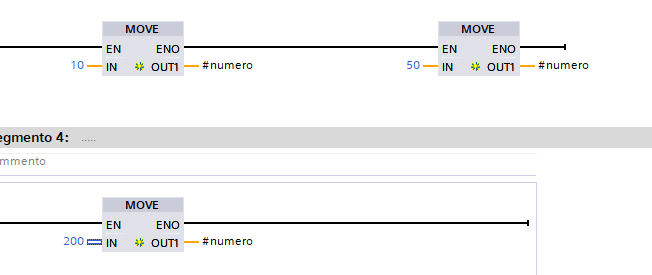
ESEMPI LETTURA CODICE

La variabile #numero inizialmente avrà valore 0, alla prima lettura del codice verrà incrementata di uno, alla seconda verrà ancora incrementata di 1 e a questo girò varrà 2 e così via.



Bisogna stare molto attenti all’ordine con il quale vengono dati i comandi, in questo esempio il PLC imposterà prima la variabile #numero a 10 e sempre sulla stessa riga dopo la imposterà a 50, quando il PLC passerà alla prossima riga il valore finale sarà 50 proprio perché è stato l’ultimo comando.

Se proseguendo con il programma vado a modificare ancora il valore della stessa variabile il risultato finale che avrò a fine ciclo non sarà più 50 ma 200.



Linguaggi di programmazione

La progettazione e lo sviluppo del software di controllo per PLC presenta alcune problematiche dovute principalmente alla sua dipendenza dall’hardware del controllore da programmare. Il mercato dei controllori programmabili per l’industria è costituito da un gran numero di produttori, ognuno dei quali mette a disposizione dei propri clienti un ambiente di sviluppo proprietario. Sebbene questi software siano per certi versi simili, le differenze tra le architetture hardware

dei vari dispositivi e le funzionalità per le quali questi sono progettati possono rendere molto differenti i set di istruzioni disponibili al programmatore nei vari ambienti di sviluppo.

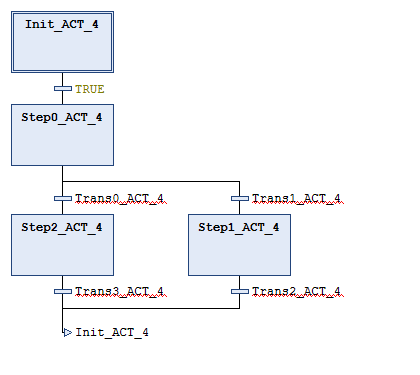
Per favorire una convergenza fra i diversi costruttori nello sviluppo di software per la programmazione dei PLC è stato introdotto dall’organismo internazionale IEC (Internation Electrotechnical Commission) uno standard che si propone di definire gli aspetti descrittivi e di programmazione dei dispositivi di controllo per l’Automazione Industriale.

Tale norma è denominata IEC 1131 e risale al 1993.

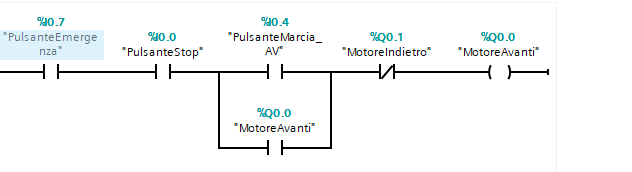
In particolare nella sua terza parte definisce un gruppo di linguaggi di programmazione e per ciascuno di essi indica i principali campi di applicazione e le regole sintattiche e semantiche da utilizzare. La normativa definisce in particolare

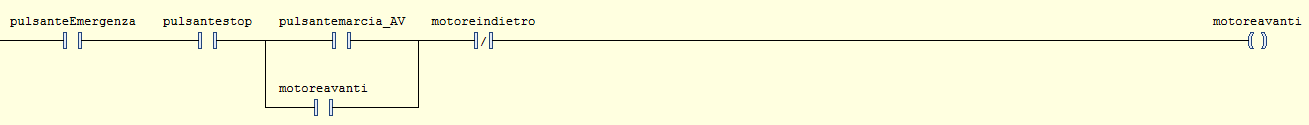
i seguenti cinque tipi di linguaggi:

• Sequential Function Charts (SFC)

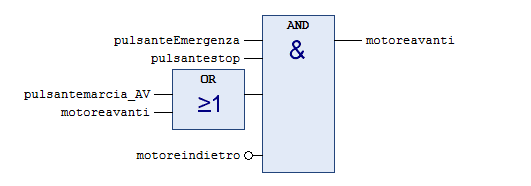
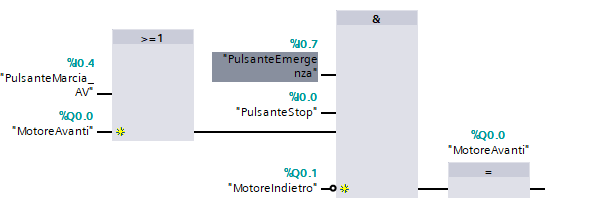


• Ladder Diagram (LD)

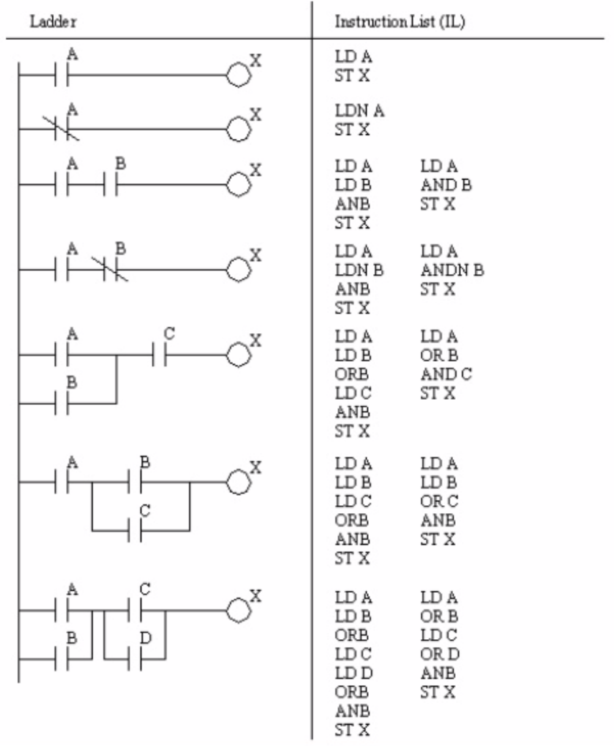




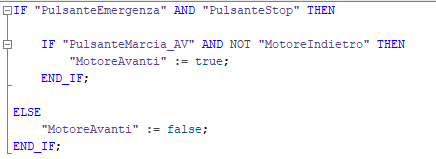
• Function Block Diagram (FBD)

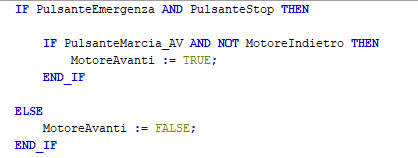


• Instruction List (IL)



• Structured Text (ST)





La norma consente inoltre, di sviluppare applicazioni per PLC combinando più tipi di linguaggio nello stesso programma.

Generalmente i costruttori non forniscono tutti e cinque i linguaggi per ciascun PLC in quanto non necessario se il linguaggio fornito è conforme allo standard.

Per quanto riguarda la trasportabilità del codice tra un costruttore ed un altro lo standard non prevede nessuna regola e non sono nemmeno previste novità in questo senso. Lo scopo principale della Norma IEC 1131-3 è quello di rendere disponibili dai diversi costruttori linguaggi di programmazione sempre più simili per ridurre i costi e gli sforzi nell’addestramento dei programmatori.

Terminale di programmazione

Il PLC non prevede tastiere e schermi quindi la sua programmazione avviene tramite dispositivi esterni. Esistono dei terminali dalle funzionalità limitate che, tramite un piccolo display e una tastiera, permettono di programmare direttamente la memoria del PLC. Generalmente, però, la programmazione avviene tramite un personal computer con software forniti dal costruttore del PLC che permette funzionalità di programmazione molto più complesse. Il PC può rimanere connesso ai PLC tramite una rete informatica permettendo all’utente di eseguire il monitoraggio del sistema, oltre alla possibilità di sviluppare e caricare direttamente i

programmi.

Confronto logica cablata/PLC

Alcuni fra i principali vantaggi nell’uso dei PLC sono riassunti nei seguenti punti.

• Flessibilità: Prima dell’introduzione di questi dispositivi le macchine automatiche richiedevano l’utilizzo di sistemi di controllo realizzati con componenti elettromeccanici sulla base di uno schema elettrico specifico per ogni tipo di macchina. Con l’introduzione dei controllori programmabili il cablaggio dei sistemi di controllo risulta molto più semplice e lo stesso tipo

di PLC, seppur con programmi differenti, può essere montato su macchine per mansioni molto differenti.

• Facile gestione dei cambiamenti e correzione degli errori: Nella logica cablata per modificare il sistema di controllo è necessario mettere mano a tutto il circuito e cambiare i componenti che lo compongono. Quando viene effettuata una modifica ad un PLC generalmente non è necessario modificare lo schema elettrico, ma è sufficiente modificare il programma con un costo in

termini di tempo assai minore.

• Prova di esecuzione: A differenza dei sistemi in logica cablata quelli programmabili possono essere facilmente testati prima di essere montati sul campo.

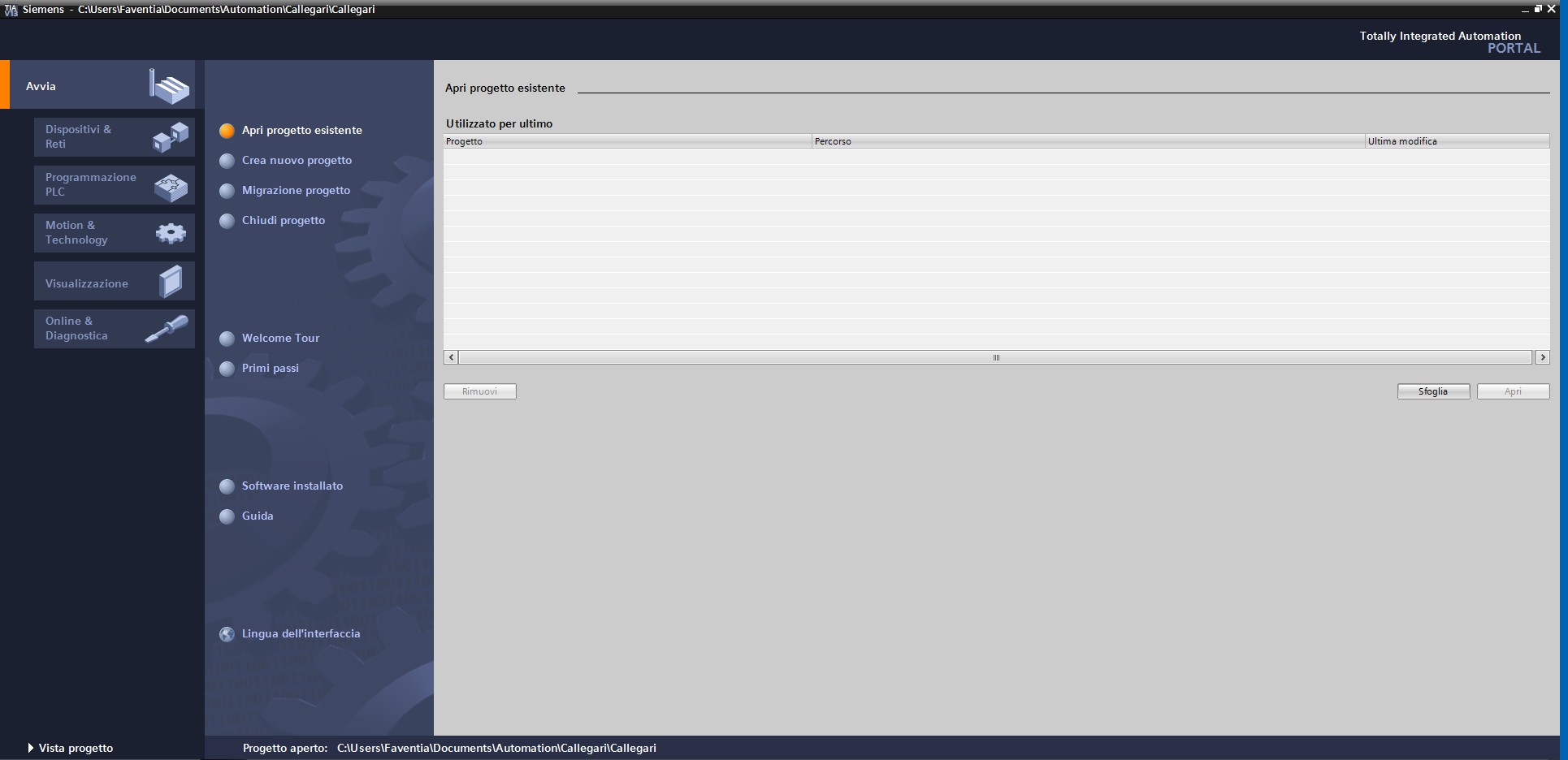
• Velocità nelle operazioni.

• Facilità di riprogrammazione: Il PLC può rimanere collegato al terminale di programmazione ed essere riprogrammato velocemente a seconda delle necessità di produzione.

* Possibilità di raccolta dati per industria 4.0 e mondo del web

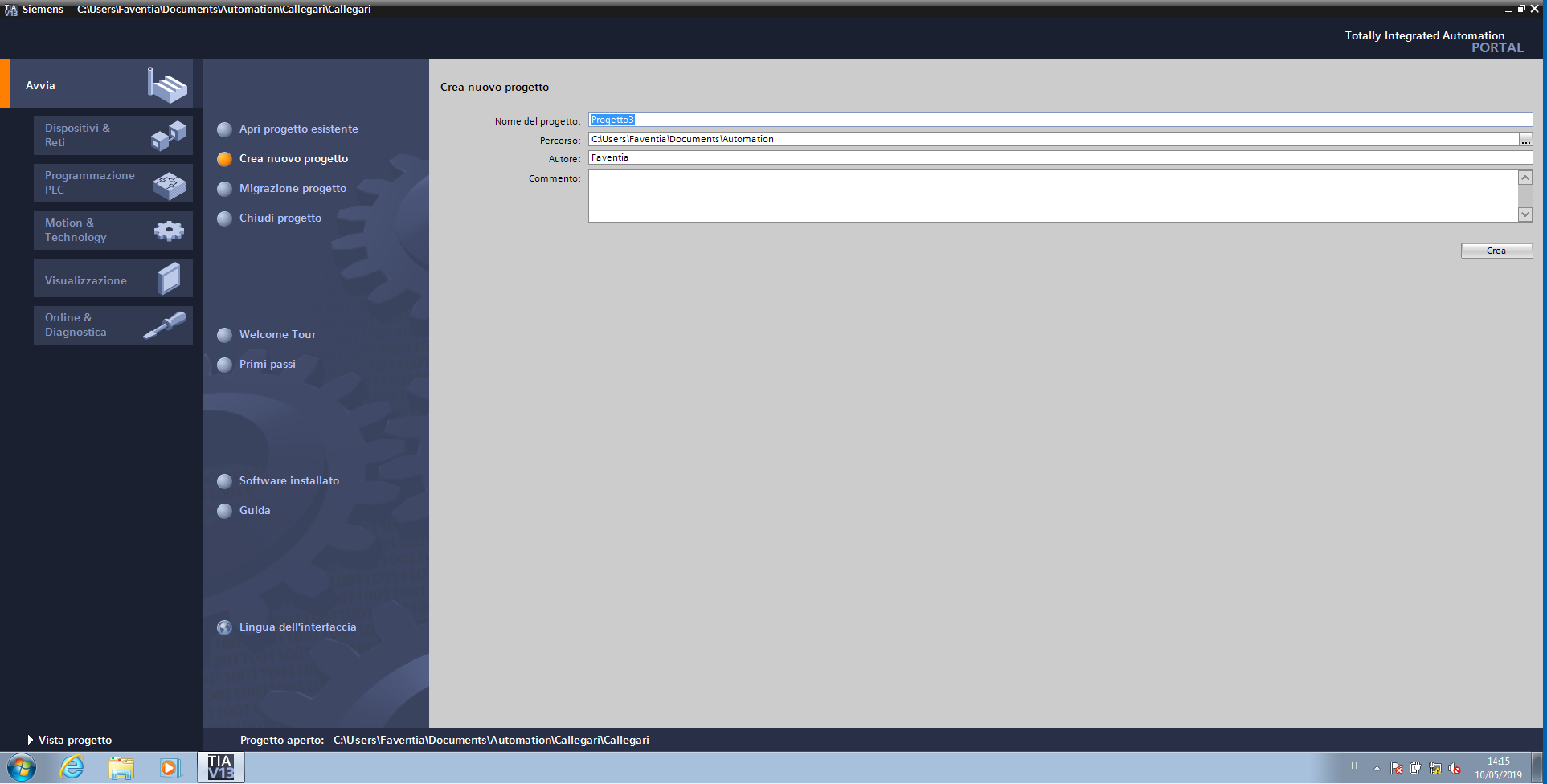
TIA PORTAL: Procedure passo per passo per configurare il progetto

* 1. **Schermata di Avvio**



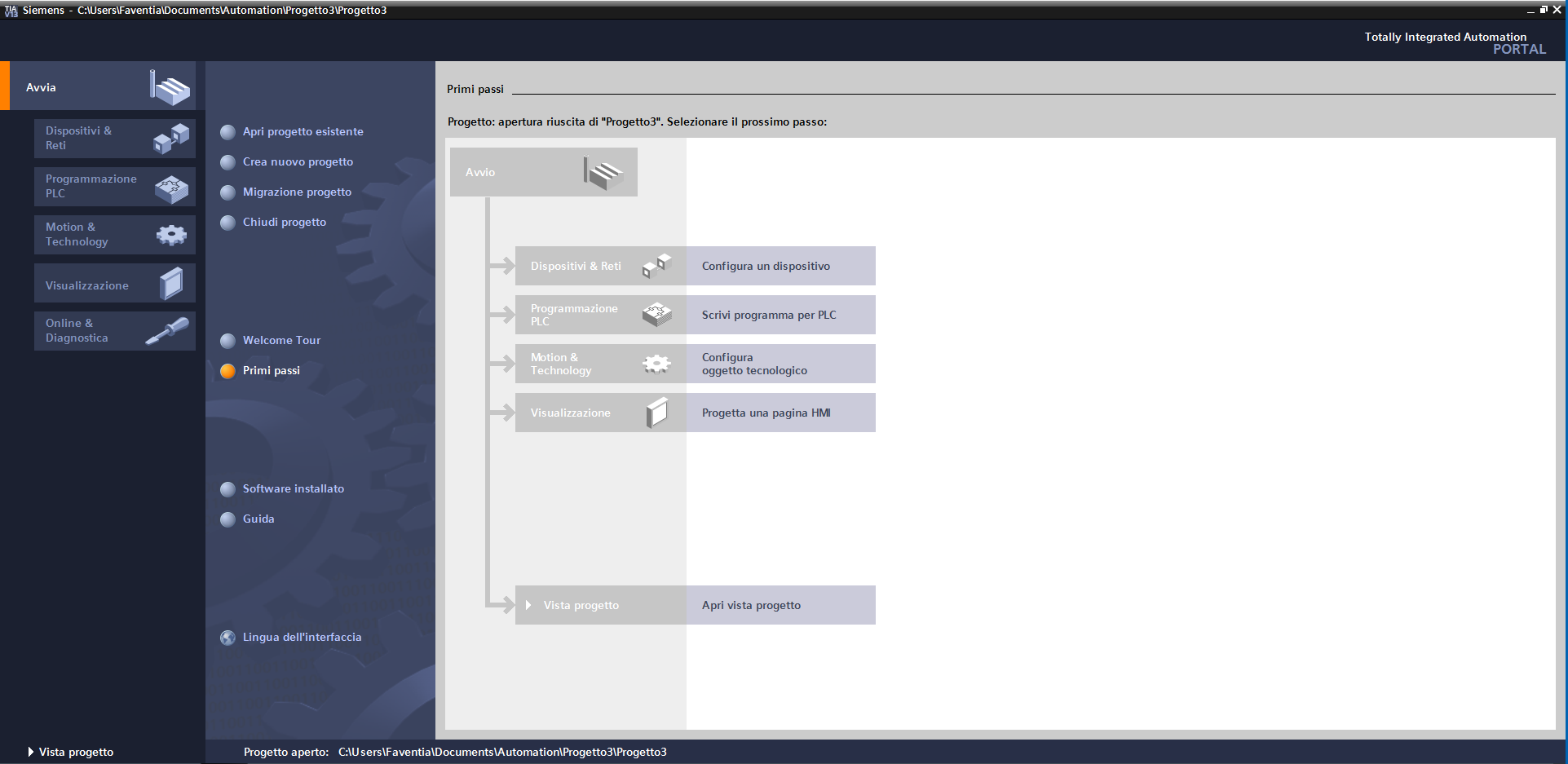
Questa è la prima schermata, da qui si può aprire un progetto già esistente o crearne uno nuovo.

* Per creare un nuovo progetto Cliccare su: “Crea nuovo progetto”.
  1. **Schermata crea nuovo progetto**



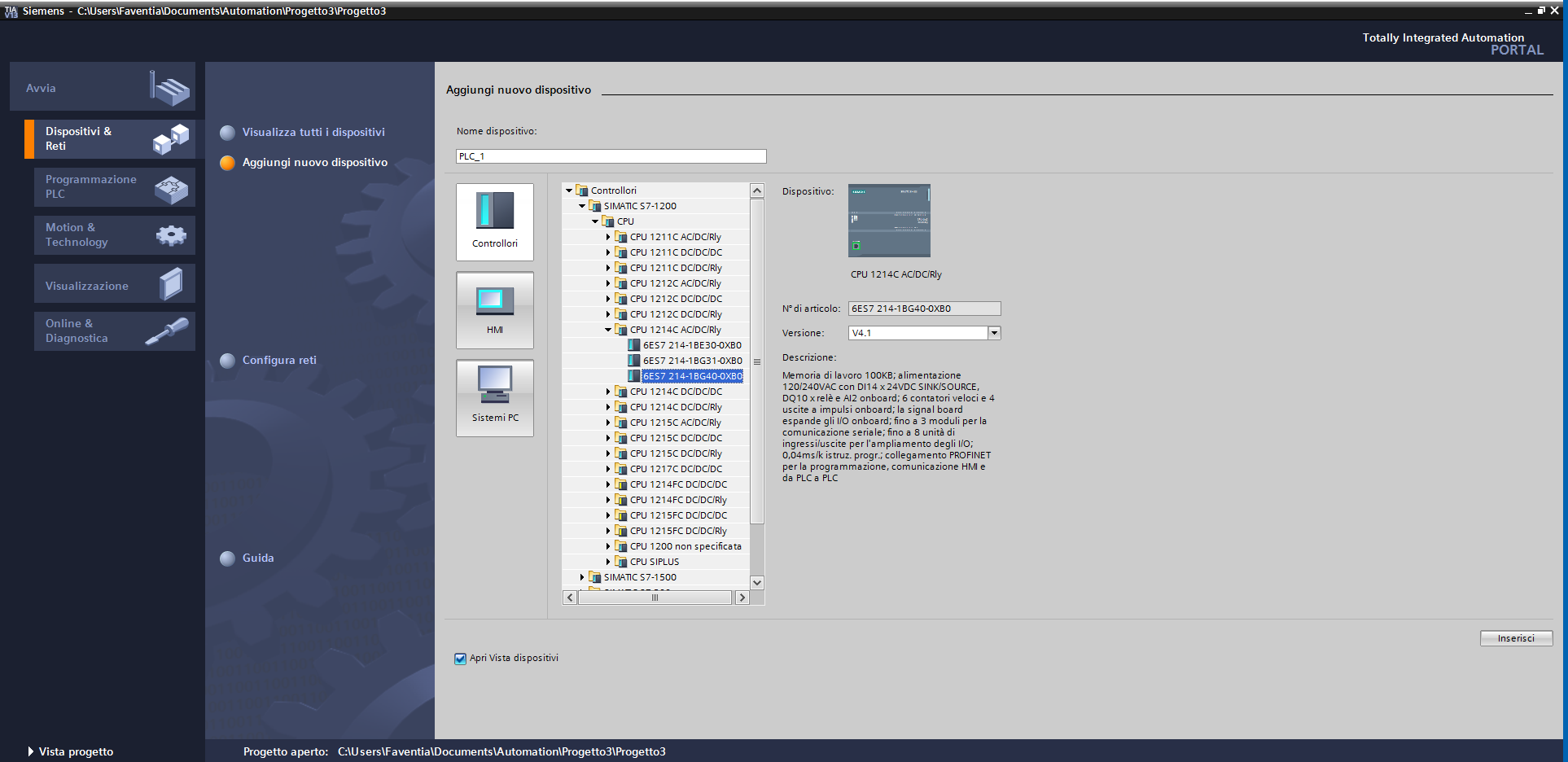
Nella schermata di creazione di un nuovo progetto possiamo dare il nome al progetto e decidere dove salvare il file.

* Per procedere e creare il progetto cliccare sul tasto “Crea”.
  1. **Primi Passi**



Questa è la prima pagina che ci si presenta dopo aver aperto un progetto, da qui possiamo iniziare a configurare il nostro progetto se si tratta di un nuovo progetto [Freccia Rossa] oppure nel caso fosse un progetto già in corso passare direttamente alla schermata di lavoro cliccando su apri vista progetto [Freccia Verde].

* 1. **Aggiungere un dispositivo**



Da questa schermata possiamo aggiungere il nostro PLC e il nostro pannello HMI.

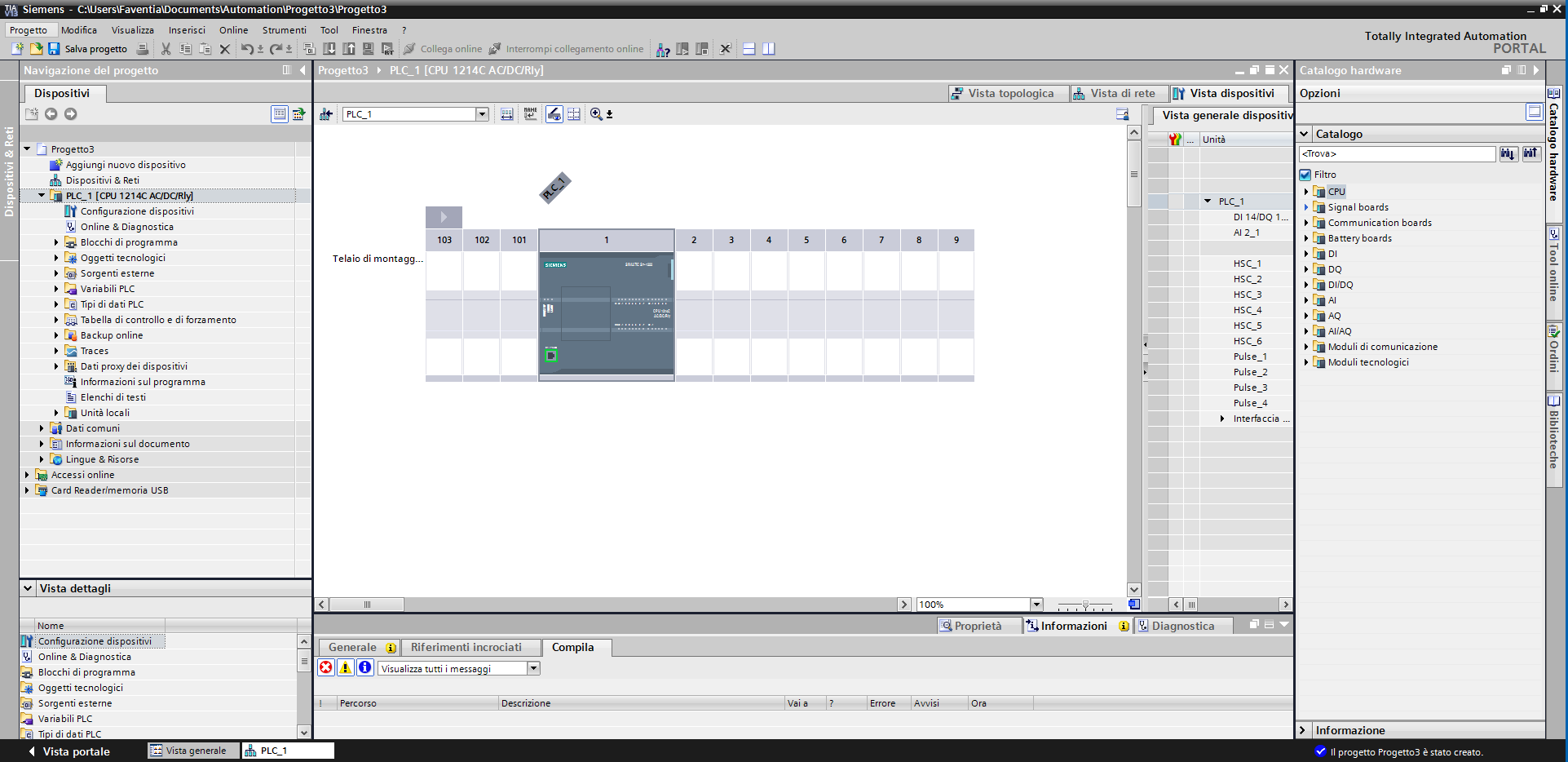
Passaggi per aggiungere il PLC:

* cliccare su controllori.
* controllare sul fianco del PLC il modello e il codice
* in questo caso il nostro modello è:  
  SIMATIC S7-1200;

CPU 1214C AC/DC/RLY;

6ES7 214-1BG40-0XB0;

* Per finire clicchiamo su Inserisci.
  1. **Prima schermata progetto e interfaccia di lavoro di TiaPortal**

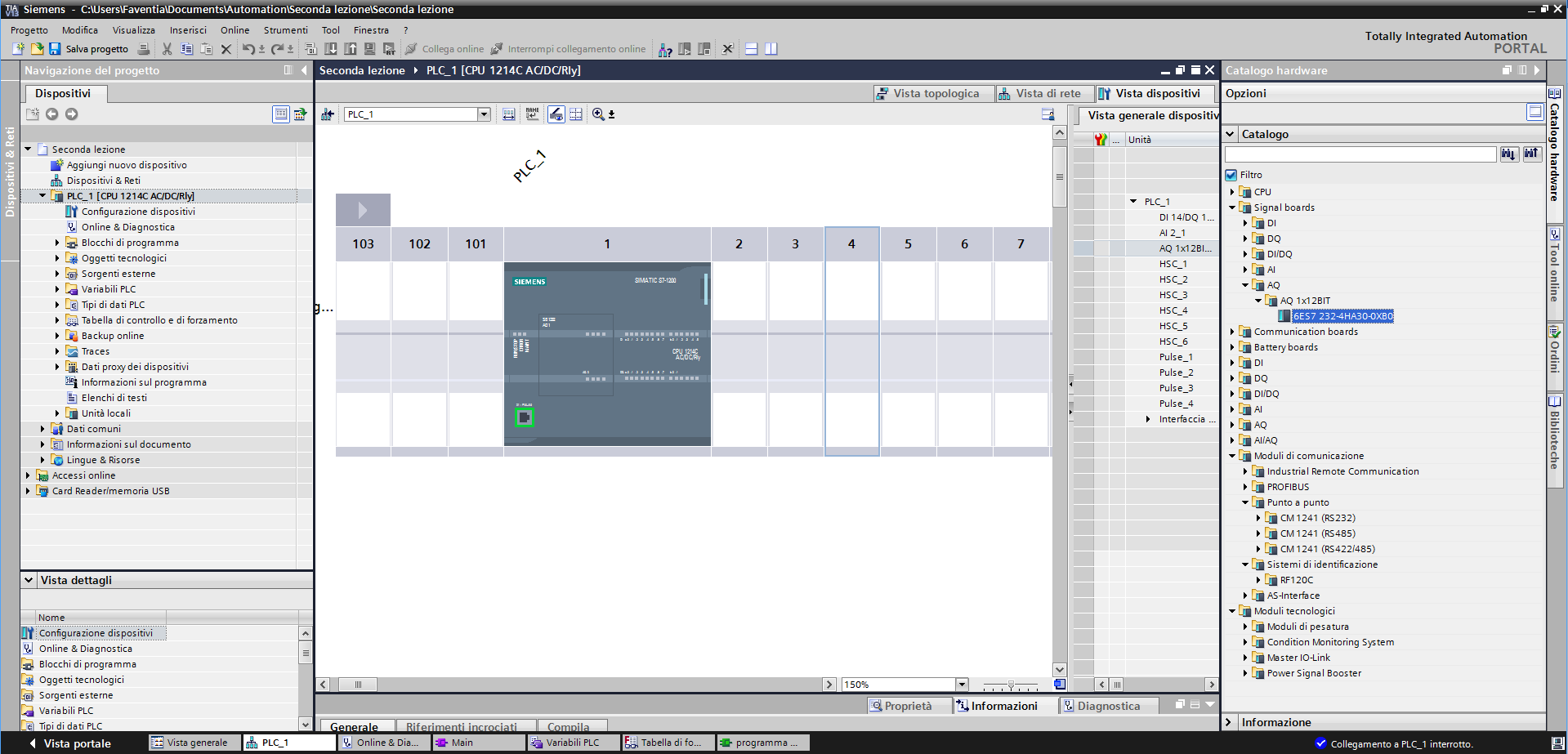


Questa è la schermata che ci si presenta dopo aver inserito il PLC ed è l’interfaccia con cui lavoreremo sul progetto per tutto il tempo.

* A Sinistra [Freccia Rossa] abbiamo il riepilogo del nostro progetto, con tutti i componenti e con tutti i vari strumenti per lavorare con essi.
* Nella parte superiore [Freccia Verde] abbiamo vari menu a tendina e la barra delle scorciatoie con i comandi più usati e i comandi per dialogare con il PLC.
* A Destra [Freccia Blu] abbiamo il menù contestuale che cambia in base a cosa stiamo facendo, ad esempio, nello screenshot siamo nella schermata di configurazione Hardware e ci mostrerà il catalogo hardware, se fossimo nella progettazione HMI avremo tutti gli strumenti per lavorare sul pannello.
* Nella parte Inferiore [Freccia Gialla] abbiamo un altro menù contestuale che ci permette di controllare le proprietà e le informazioni di qualsiasi componente e in oltre consente al programma di dialogare con l’utente, mostrando informazioni e problemi, come ad esempio tutti gli errori in fase di compilazione.
  1. **Aggiunta modulo di espansione uscite analogiche (se presente\*)**

Se nel PLC è presente un modulo di espansione anche esso dovrà essere aggiunto nella nostra configurazione del progetto.

Nel caso dei PLC del Callegari abbiamo un modulo di espansione che aggiunge al PLC alcune uscite analogiche.

****

Essendo nella pagina di configurazione dispositivi nel menu contestuale sulla destra ci verrà mostrato il catalogo hardware contenente ogni tipo di modulo di espansione per il nostro PLC.

Per trovare il modello del nostro modulo bisogna leggere il codice, come fatto nel passaggio prima con il PLC, che si trova fisicamente su esso.

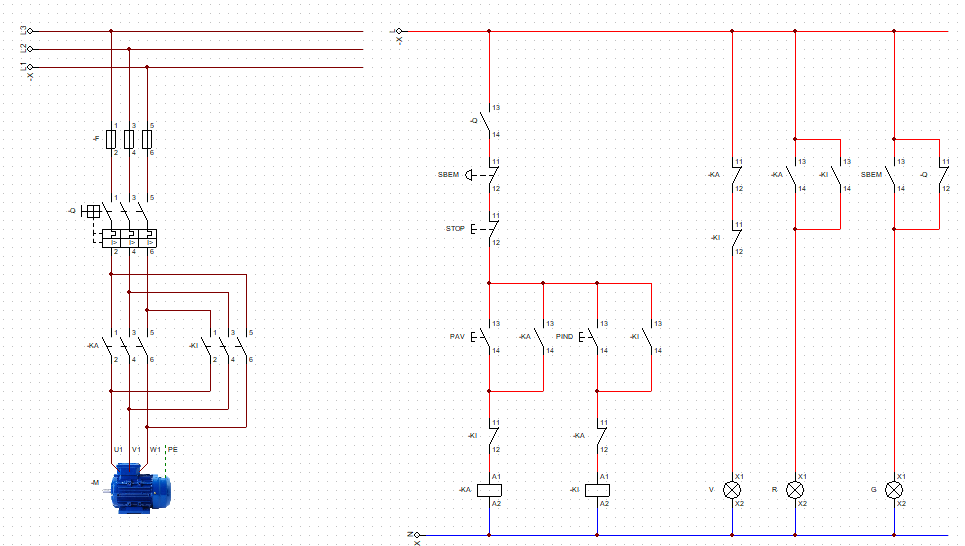
In questo caso abbiamo una Signal Boards che aggiunge uscire analogiche (AQ) con codice: 6ES7 232-4HA30-0XB0.

Trovato il componente lo si aggiunge tramite un doppio click.

Adesso che il PLC del nostro progetto è perfettamente allineato con il PLC fisico possiamo iniziare a lavorare sul nostro software.

1. **RIPRODURRE LA LOGICA CABLATA ALL’INTERNO DEL PLC**

Como primo esercizio andremo a riprodurre il funzionamento di uno schema elettrico all’interno del nostro PLC, come esempio utilizzeremo lo schema di un semplice motore avanti – indietro con interblocco.

Ecco lo schema elettrico che prenderemo come riferimento:

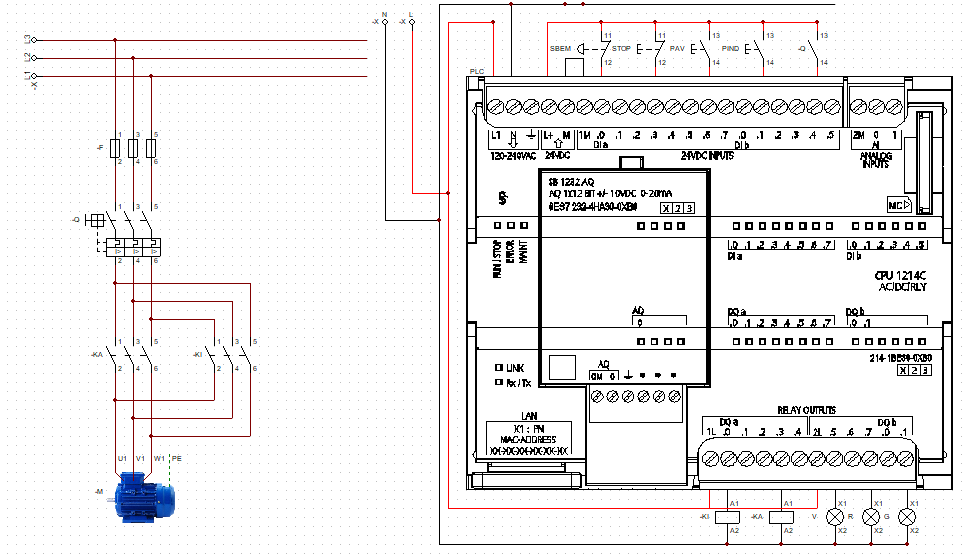
A sinistra abbiamo la parte di potenza dove abbiamo:

* la linea trifase dal quale il nostro motore verrà alimentato;
* a seguire i fusibili e l’interruttore magnetotermico, che proteggeranno il nostro motore in caso di malfunzionamento;
* i due teleruttori, uno per la marcia avanti e l’altro per la marcia avanti;
* il motore.

A destra abbiamo la parte dell’impianto in bassa tensione dove è presente la logica di lavoro dell’impianto dove abbiamo:

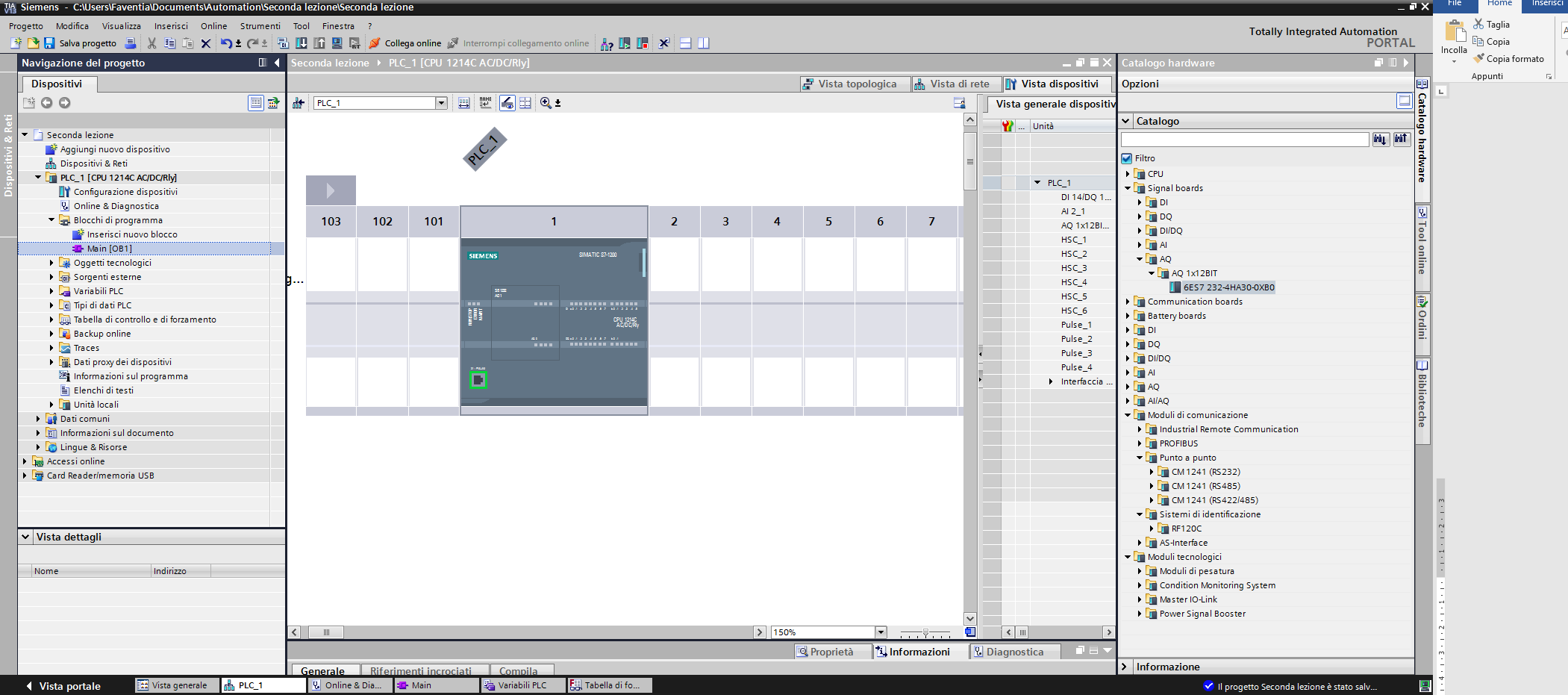
* Un pulsante di emergenza, il contatto ausiliario della protezione termica e il pulsante di stop per fermare il motore;
* I due pulsanti di start per l’avanti e l’indietro;
* Le due bobine dei due teleruttori;
* Tre lampadine per segnalare la marcia in avanti e indietro e un eventuale allarme;

Ecco come sarà il nostro schema elettrico con l’utilizzo di un PLC, si può notare che la parte di potenza è rimasta invariata mentre la parte di logica è totalmente diversa.



Avremo sempre i nostri dispositivi di comando come i vari tasti o le bobine di comando dei teleruttori ma al posto della logica cablata fisica avremo il nostro PLC con il programma che andremo a scrivere tramite Tia Portal che gestirà i nostri dispositivi e il funzionamento dell’impianto.

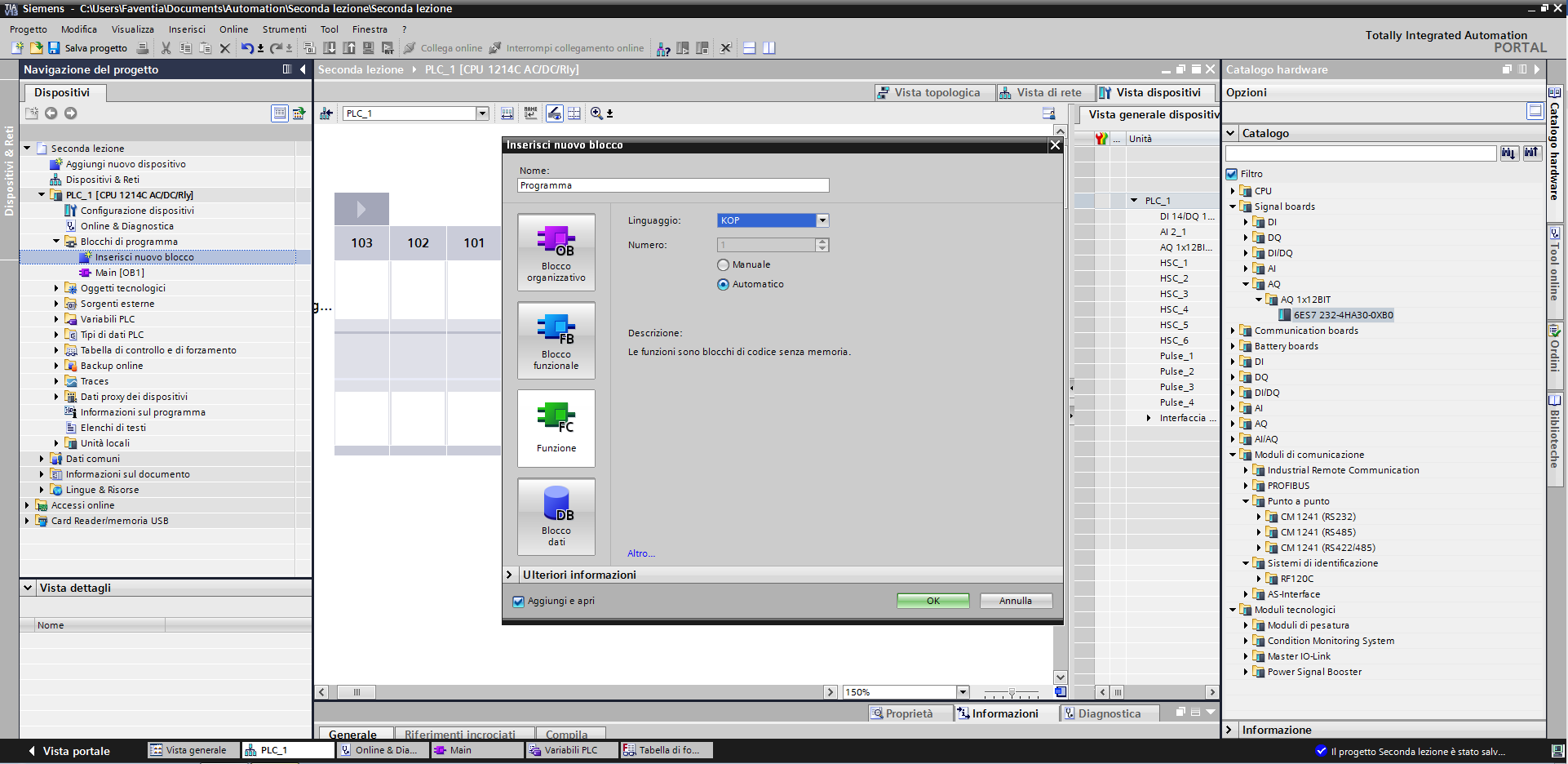
* 1. **INIZIO PROGRAMMAZIONE**

Torniamo su Tia Portal e iniziamo a impostare il nostro programma, per prima cosa andiamo nel menù “Blocchi di programma” e clicchiamo su “Inserisci nuovo blocco”.

Ci ritroveremo nel menù di creazione del blocco da dove è possibile dare il nome al nostro programma e selezionare che tipo di linguaggio utilizzare, Per questo esempio useremo il KOP (Ladder) perché è il più semplice da utilizzare e visivamente il più simile ai simboli dei

schemi elettrici.

Per scrivere il nostro programma utilizzeremo una Funzione (FC).



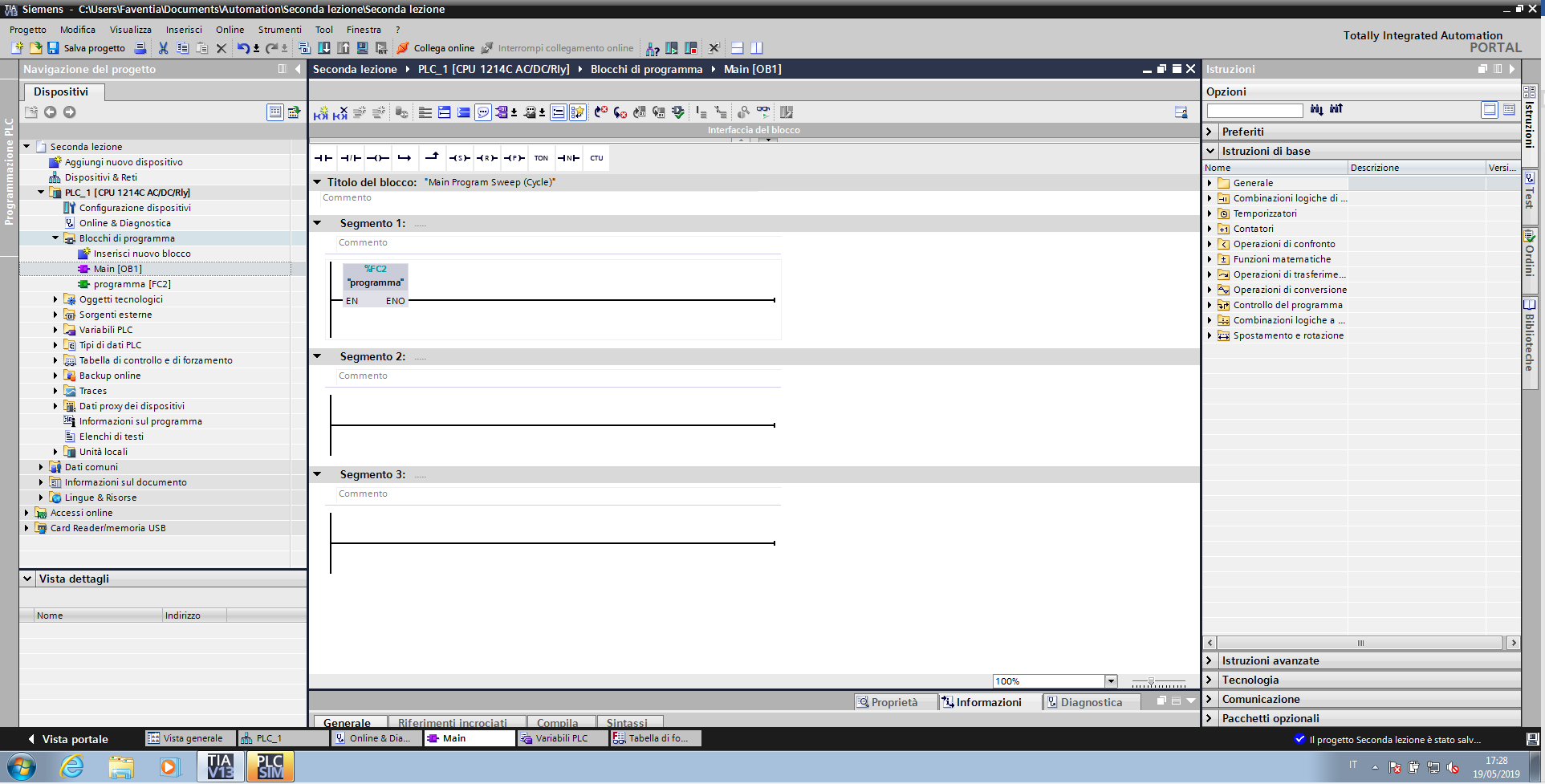
Cliccando ok vedremo che il nostro programma verrà aggiunto all’interno della cartella “blocchi di programma”.

Ora abbiamo creato la nostra funzione dove scriveremo il nostro programma, ma prima di fare ciò dobbiamo segnalare al nostro PLC l’esistenza della funzione appena creata.

Per far ciò utilizzeremo il blocco “Main” già presente da prima della creazione della nostra FC.

Il blocco “Main” è un blocco di sistema del PLC messa a disposizione della Siemens per gestire vari parametri del PLC e gestire le varie FC del nostro programma.

Per segnalare l’esistenza al nostro PLC del nostro programma non dobbiamo fare altro che aprire il blocco “Main” e trascinare all’interno di esso la nostra FC “Programma”.

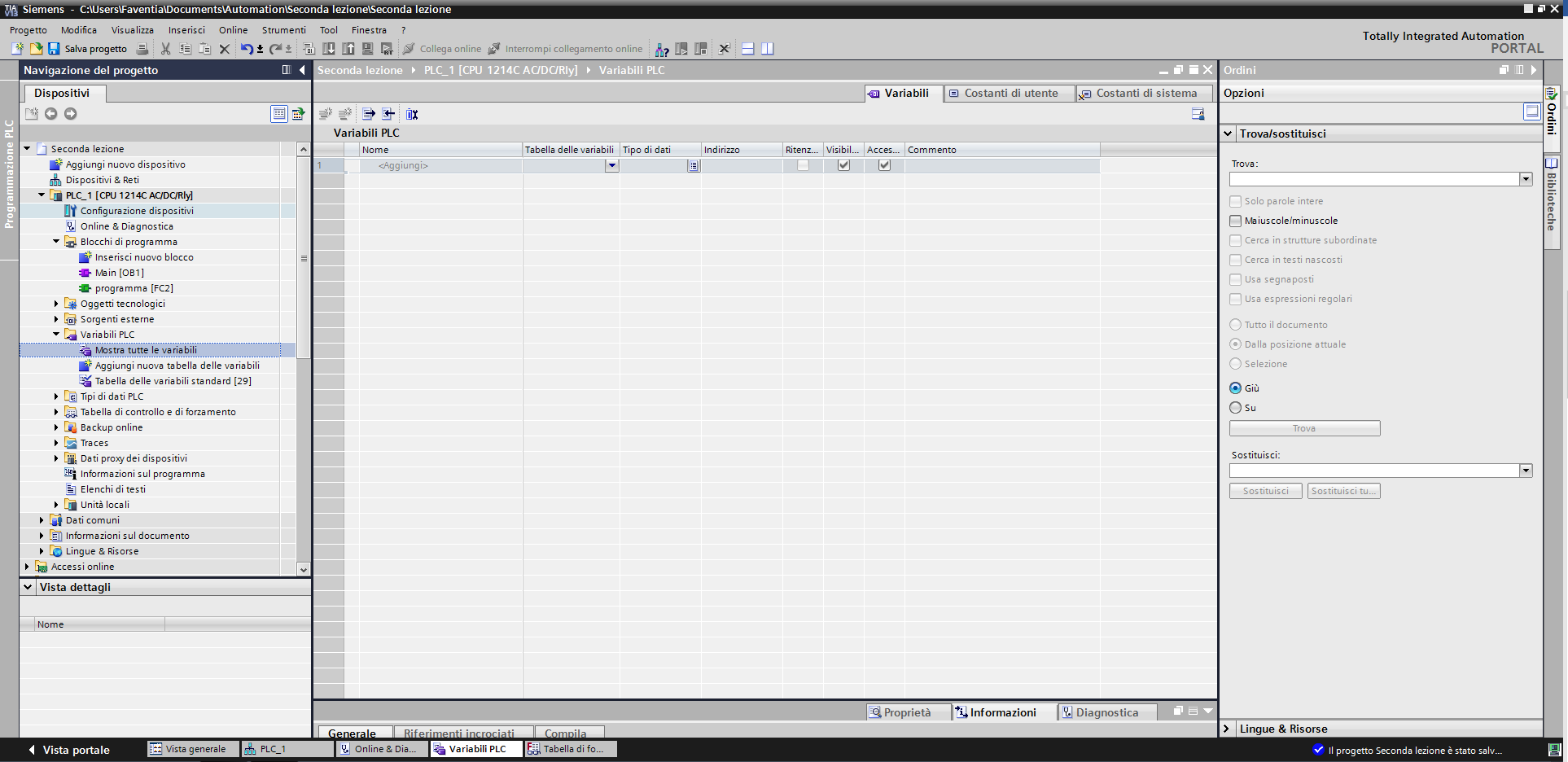


Ora che abbiamo inserito la nostra FC all’interno del Main siamo sicuri che essa verrà eseguita dal PLC.

* 1. **VARIABILI**

Ora che il nostro programma può essere eseguito passiamo all’inserimento all’interno del nostro programma dei vari componenti che sono collegati al nostro PLC.

Per far ciò andiamo nella cartella “Variabili PLC” e clicchiamo su “Mostra tutte le variabili.



Ci ritroveremo davanti a una tabella vuota, ed è proprio tramite questa tabella che andremo ad aggiungere i nostri dispositivi di Input e Output.

Ma prima di iniziare a inserirle facciamo un attimo mente locale e andiamo a controllare che dispositivi avevamo nello schema elettrico preso prima come esempio.

abbiamo come Input:

* 2 tasti di start;
* Un tasto di emergenza;
* Un tasto stop;
* Il contatto del magnetotermico.

Mentre come output:

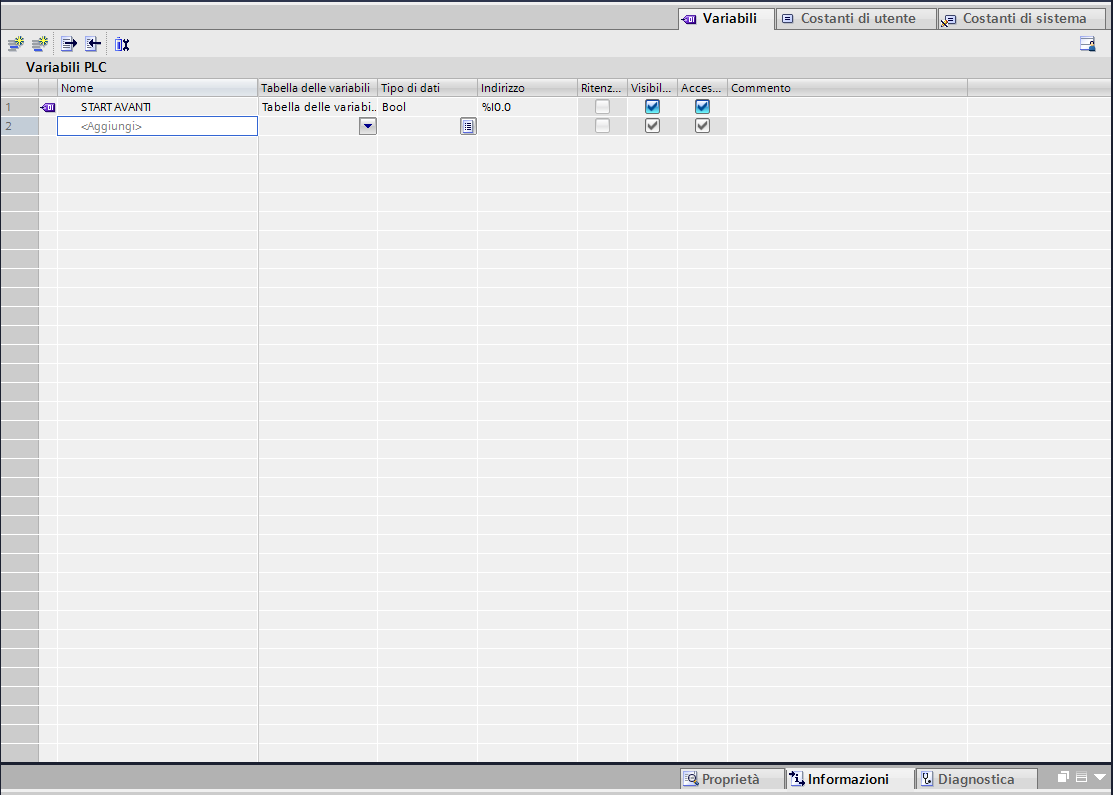
* Le due bobine dei teleruttori;
* Le due lampadine.

Per prima cosa andiamo a inserire i dispositivi di Input.

Prendiamo come esempio il pulsante start avanti, clicchiamo su aggiungi e inseriamo il suo nome e clicchiamo invia per creare la variabile.

Come si può notare oltre al nome abbiamo altri campi che interessano la nostra variabile:

* abbiamo la voce tabella delle variabili che indica dove è stato dichiarare la variabile;
* abbiamo il tipo di dato che indica con che tipo di variabile stiamo lavorando, in questo caso è di tipo BOOL, ciò vuol dire che può avere solo due stati, 0 o 1 (come i tasti possono essere aperti e chiusi) e
* l’indirizzo, nell’indirizzo andiamo a definire in che ingresso è cablato il nostro tasto.

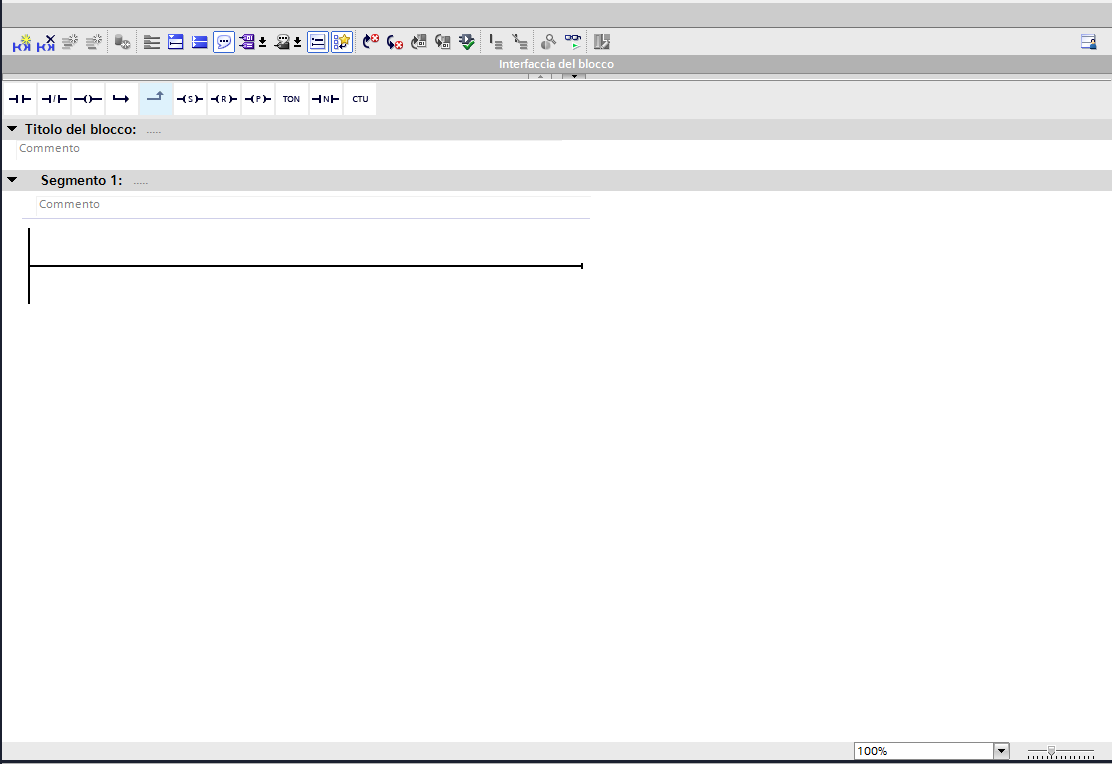


Dopo aver inserito tutti i nostri dispositivi di Input inserire anche i dispositivi di Output, Fare Attenzione all’Indirizzo, non sarà più una I (ingresso) ma una Q(uscita).

* 1. **PROGRAMMAZIONE LADDER**

Ora che abbiamo inserito tutte le nostre variabili possiamo finalmente a scrivere il nostro programma.

Aprendo la nostra FC programma ci ritroveremo davanti all’editor di programmazione.



In alto all’editor possiamo trovare i comandi base che ci serviranno durante la programmazione.

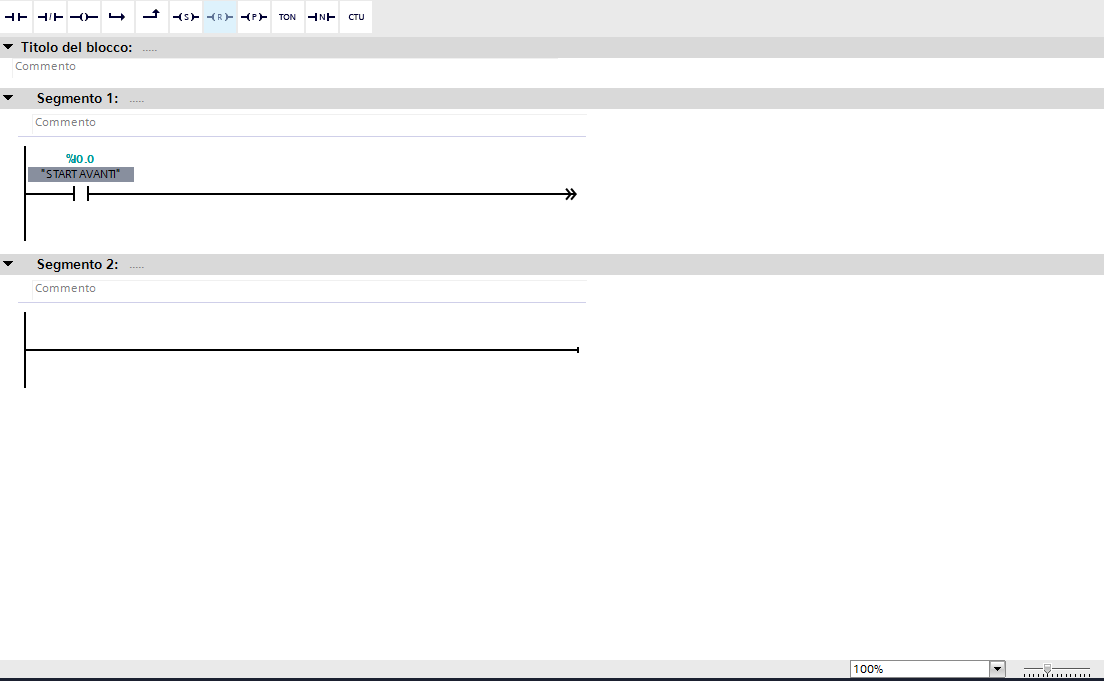


Qui abbiamo i principali:

* Il primo è l’equivalente di un contatto normalmente aperto che alla chiusura/attivazione fa passare corrente.
* Il secondo è l’equivalente di un contatto normalmente chiuso che permette sempre il passaggio di corrente tranne nel caso fosse aperto/attivato.
* Il terzo è l’equivalente di un comando a bobina
* Mentre gli ultimi due sono due comandi di Tia Portal che ci permettono di modellare i vari rami.

Essi possono essere aggiunti trascinandoli all’interno del segmento trascinandoli all’interno del segmento.

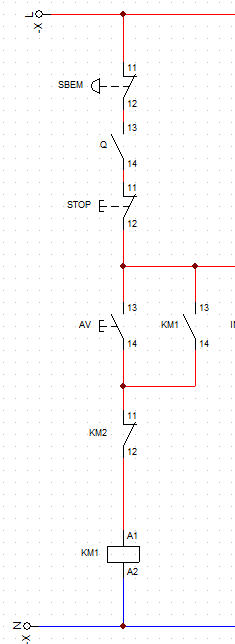
Dopo aver inserito un simbolo, prendiamo come esempio un contatto aperto, ci verrà chiesto di inserire il nome di una variabile, è proprio in questo punto che possiamo iniziare ad utilizzare le variabili inserite prima nell’apposita tabella, essendo un contatto aperto proviamo a inserire uno dei nostri start.



Ora che sappiamo come inserire i nostri comandi all’interno dello schema Ladder possiamo iniziare a trasferire la nostra logica cablata all’interno del PLC.

* 1. **Comparazione logica cablata/ Ladder**

Adesso che abbiamo copiato tutto lo schema elettrico andiamo a vedere le differenze fra logica cablata e logica dello schema ladder prendendo come esempio solo un ramo, quello del motore avanti.

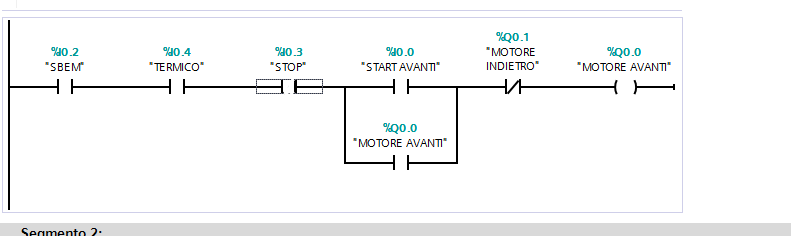
Come possiamo vedere visivamente sono molto simili, ma se si osserva con attenzione la gestione delle sicurezze, quindi i primi tre contatti ci sono delle differenze.

Questo perché i dispositivi di sicurezza devono essere sempre a contatto chiuso

E per questo la logica dei contatti all’interno del PLC è inversa.

Il tasto di emergenza e il contatto del termico vengono considerati dispositivi di emergenza e vengono sempre cablati elettricamente come contatti chiusi, questo vuol dire che il PLC rileverà sempre il passaggio di corrente ai loro ingressi e a sua volta la loro variabile sarà sempre a uno, proprio per questo mettiamo il contatto aperto di essi, fino a quando sono a uno permetteranno il passaggio di corrente, nel momento che verranno aperti elettricamente il PLC non rileverà più la corrente e i contatti interni al PLC andranno a 0 interrompendo il flusso di corrente nel nostro Programma.

Il pulsante di stop non viene considerato un dispositivo di sicurezza ma viene sempre collegato tramite contatto chiuso (in caso di guasto la macchina non partirebbe), quindi fino a quando non verrà premuto il tasto fisico stop il suo contatto aperto, ma attivo, permetterà il passaggio di corrente.

****

Questa è una delle regole fondamentali da conoscere nella programmazione dei PLC, I dispositivi di sicurezza vanno sempre collegati elettricamente come contatti chiusi e nel Software devono essere sempre trattati come contatti aperti.

1. **Barra degli strumenti**

4

6

5

3

2

1



1. Crea nuovo progetto, Apri progetto e Salva progetto;
2. Taglia, Copia, Incolla e Elimina;
3. Annulla operazione e Ripeti;

* Compila progetto: Ci permette di trovare errori di “**grammatica**” nel codice che abbiamo scritto, ogni errore verrà segnalato nel menù contestuale inferiore.
* (**ATTENZIONE: Trova gli errori grammaticali e non di logica di programmazione**).
* Download programma in PLC: Carica il programma all’interno della memoria del PLC.
* Upload da PLC: Scarica il programma dal PLC al Computer.
* Simula Dispositivo: Permette di simulare il PLC o un pannello HMI, possono essere simulati in simultanea per testare il funzionamento completo del programma.
* Collega online: Collega il PC al PLC senza eseguire nessun download, ci permette di controllare cosa sta accadendo all’interno del programma.
* Interrompi collegamento online: Scollegare il PC dal PLC, da fare assolutamente quando si sconnette il cavo Ethernet.
* Controlla rete: Cerca e trova tutti i dispositivi all’interno della rete.
* Start CPU: Inizia a far ciclare il programma nel PLC.
* Stop CPU: Smette di far ciclare il programma nel PLC, la CPU può andare in stop automaticamente nel caso di qualche problema grave non dovuto alla logica del programma ma dovuto a qualche malfunzionamento in qualche suo parametro interno.
  1. **Scaricare il programma e testarlo**

Adesso che abbiamo scritto il nostro programma siamo pronti per testarlo.

Per testarlo dobbiamo scaricare all’interno del PLC il nostro Software.

Nel caso non si abbia un PLC a disposizione Tia Portal ci mette a disposizione un simulatore che emula il funzionamento del plc.

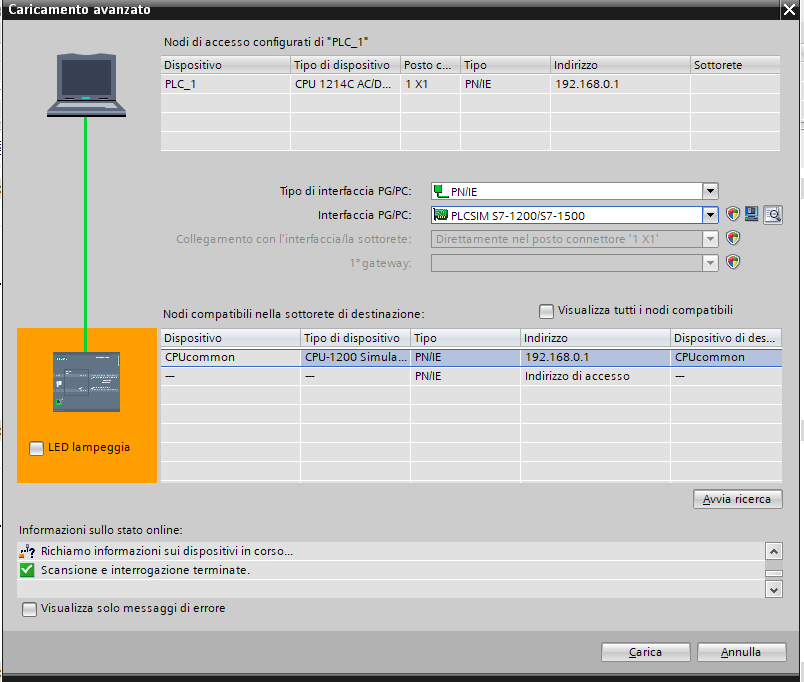
La procedura del Download è uguale in entrambi i casi, cambia solo la procedura con cui ci si connette.

Se si possiede un PLC basta collegarlo tramite cavo Ethernet al nostro computer mentre se non se ne ha a disposizione basta lanciare la simulazione.

 **Simulazione**

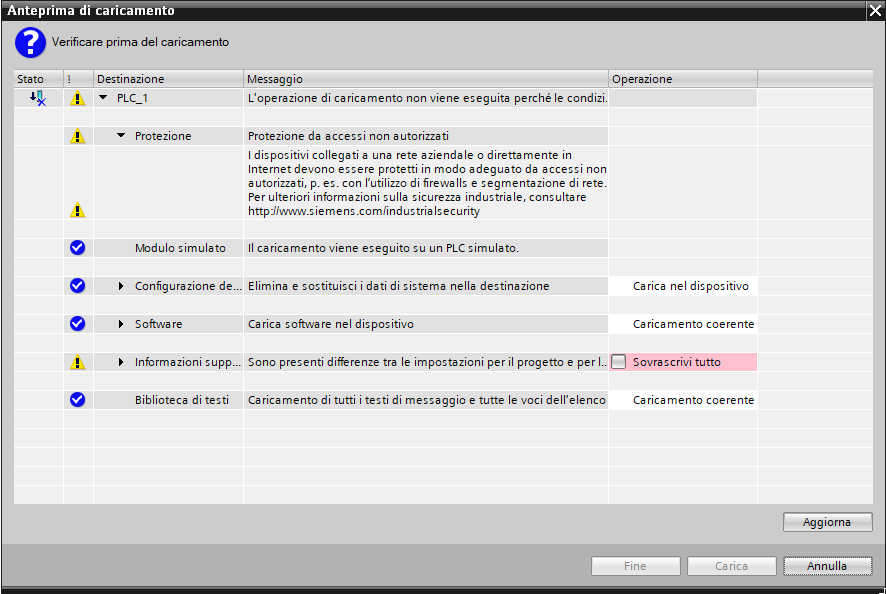
Per prima cosa bisogna compilare il software tramite tasto apposito e in caso di nessun errore caricare il software tramite il tasto ” Carica nel dispositivo ”.

Questa è la prima schermata che ci si presente da dove andremo a selezionare il dispositivo in cui caricare il software, per prima cosa premere su avvia ricerca e dopo aver trovato il PLC all’interno della rete premere su Carica.

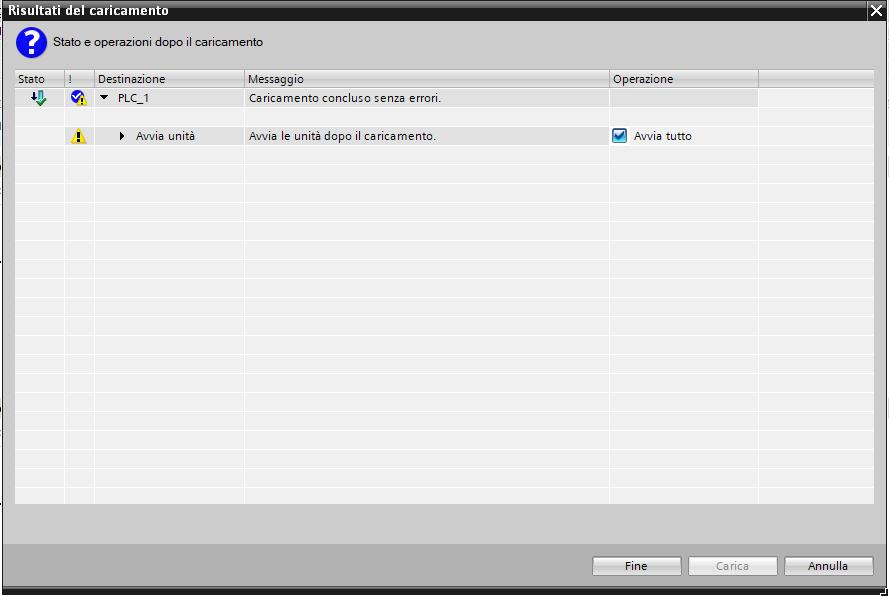


Nella schermata successiva ci verranno mostrate alcune informazioni in merito al download.

Cliccare su “Sovrascrivi tutto” in modo da cancellare il progetto precedentemente presente sul PLC e caricare il nostro e cliccare su carica.

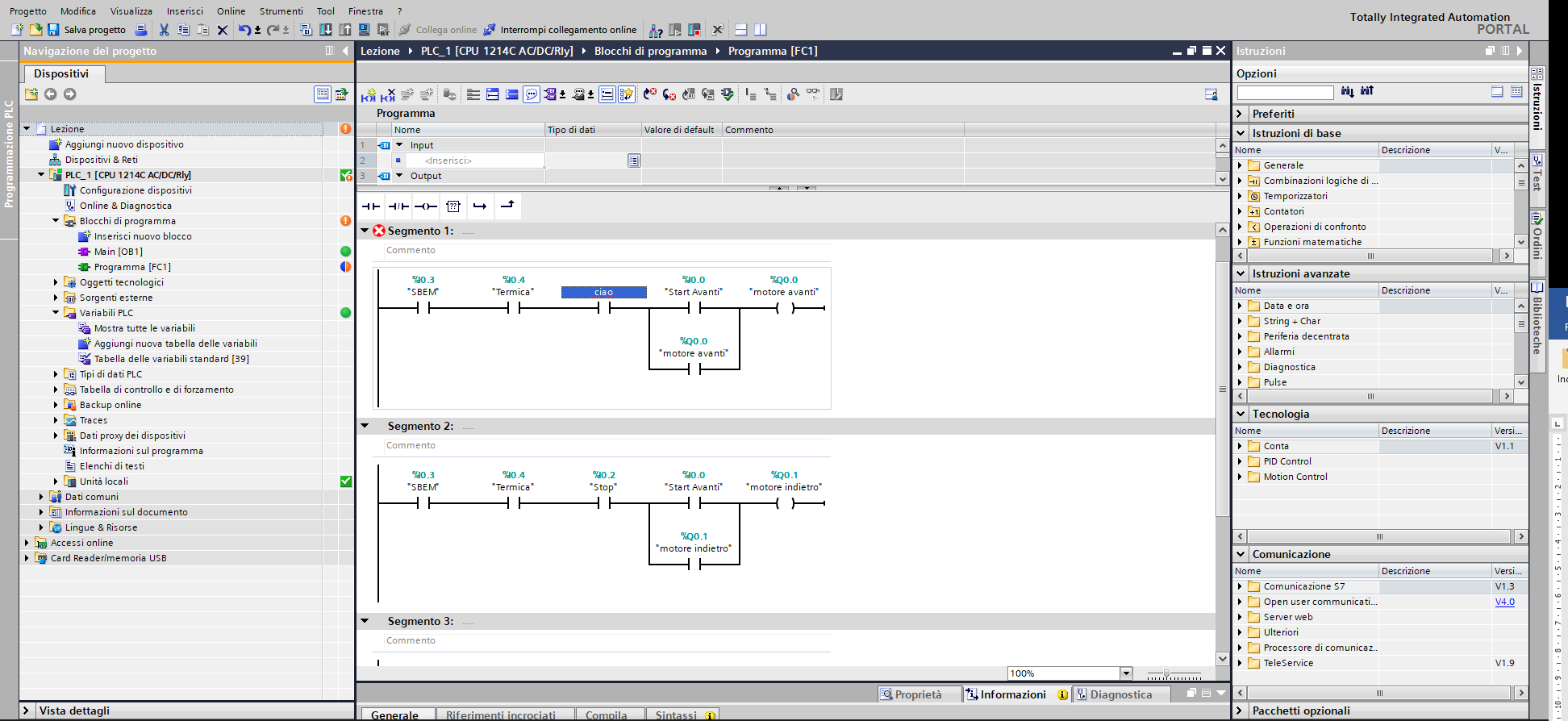


Nella pagina successiva spuntare “Avvia tutto” per mettere la CPU in Run e cliccare su Fine.



Dopo esser tornati su nella schermata principale di Tia Portal clicchiamo su Collega online nella barra superiore per collegarci al PLC.

Dopo esserci collegati nel menù a sinistra appariranno dei pallini verdi che stanno a indicare che il nostro progetto è allineato con quello all’interno del PLC, nel caso si facessero delle modifiche il pallino diventerà metà arancione e metà blu, e in questo caso bisogna rifare nuovamente il download in modo da allineare i due progetti.

****

Ora che siamo collegati al PLC possiamo vedere cosa sta succedendo all’interno del nostro software, per far ciò dobbiamo cliccare sull’icona di controllo in alto nell’editor Ladder.

Bene ora siamo online e possiamo iniziare a provare il nostro software e verificare che il tutto funzioni come dovuto.

Se si è in possesso di un PLC basterà muovere i vari pulsanti collegati negli ingressi per lavorare sul nostro programma, mentre se non si è in possesso di un PLC ma si è in simulazione dovremo simulare anche gli ingressi fisici.

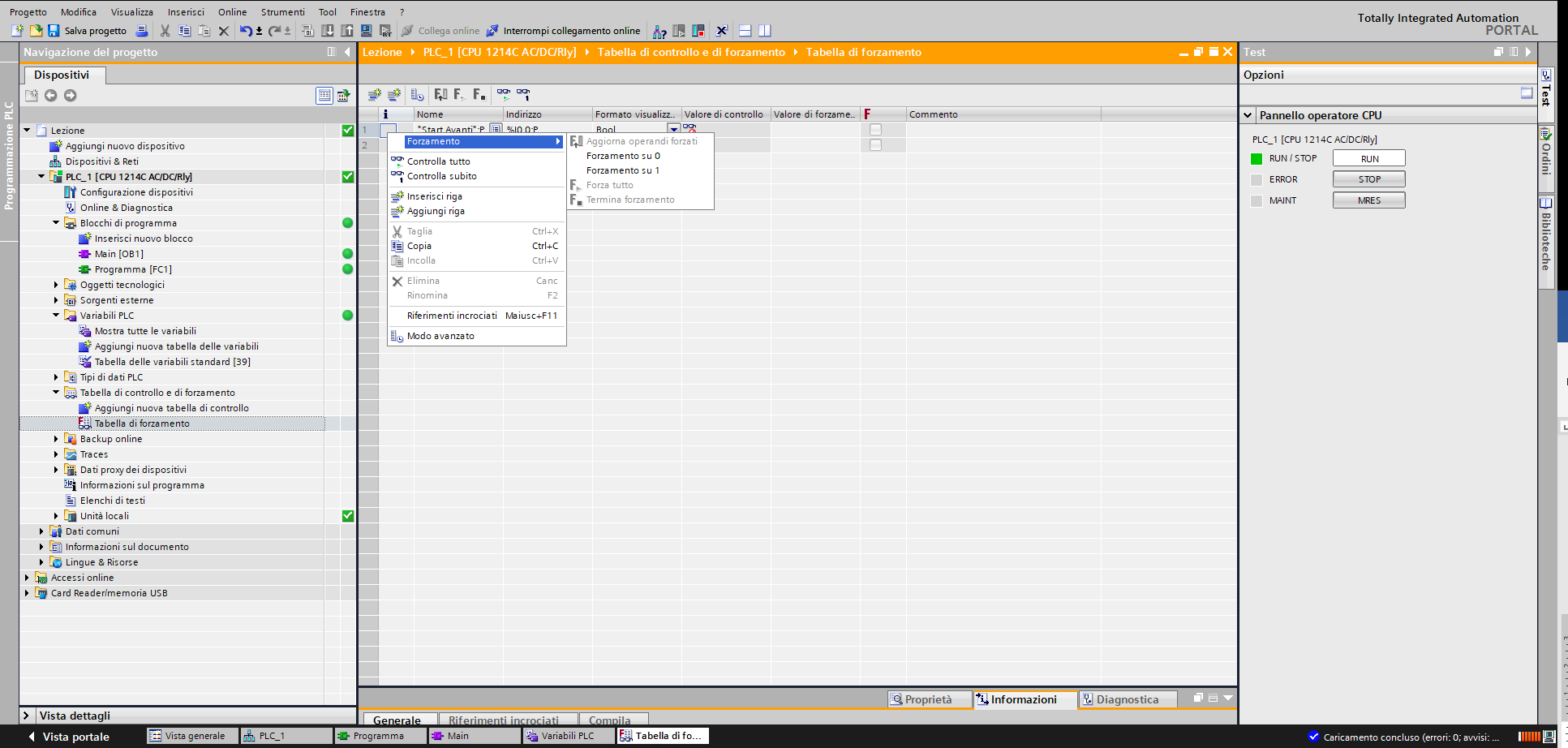
Per far ciò utilizzeremo la Tabella di Forzamento.

* 1. **Tabella di Forzamento**

Troviamo la tabella di forzamento nella cartella nel menù a sinistra “Tabella di controllo e di forzamento”.

Aprendola ci troveremo a una tabella simile a quella delle variabili, qui possiamo inserire le nostre variabili che vogliamo controllare, come esempio prendiamo il pulsante start avanti,

inserisco il suo nome all’interno della tabella e tramite il tasto destro del mouse apro il menu a tendina e vado su “forzamento”, qui posso decidere in che stato deve essere il mio pulsante, per far partire il motore metterò il pulsante ad uno.

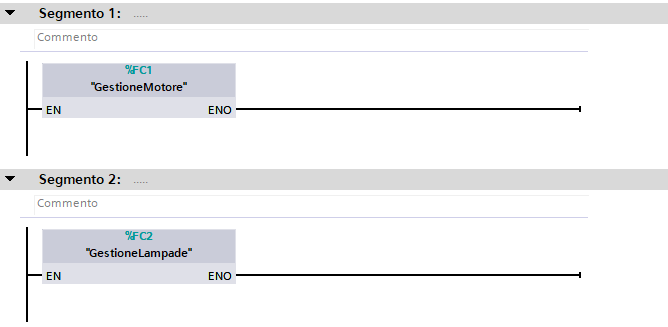
****

Per fermare il forzamento di tutte le variabili basta cliccare su “termina il forzamento” (Freccia Blu).

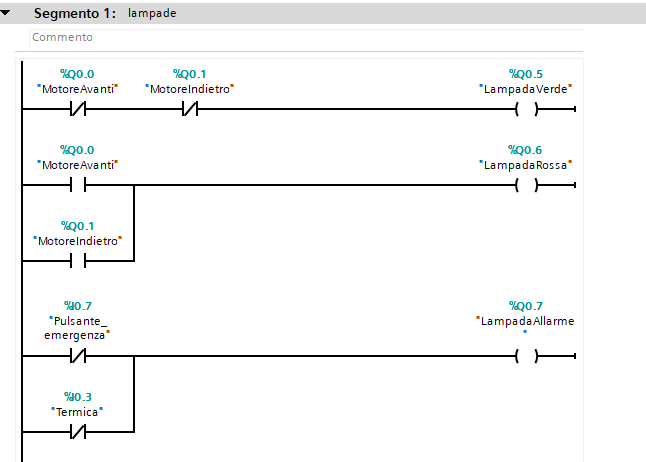
* 1. **FC Gestione Lampadine**

Adesso che abbiamo testato il funzionamento della logica dell’accensione di entrambi i sensi di marcia possiamo finire la copiatura della logica cablata.

Quindi adesso creeremo la FC per gestire le lampadine e la andremo a inserire nel nostro OB1 in modo che anche essa venga eseguita

****

E dopo di che copiare la logica come abbiamo fatto con la marcia avanti e indietro

****

* 1. **Gestione allarmi**

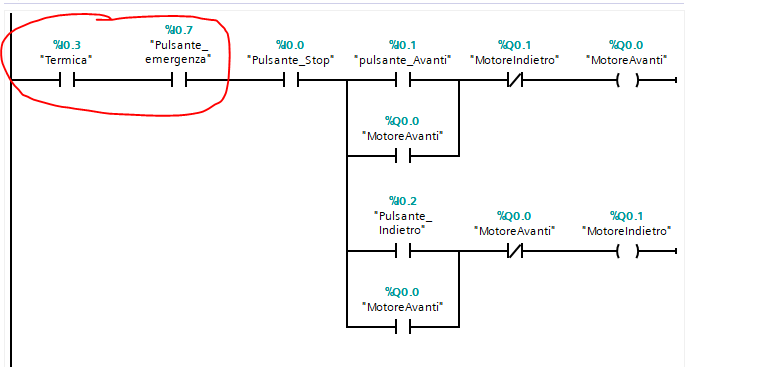
Fino ad adesso abbiamo ricopiato la logica dallo schema elettrico, adesso è arrivato il momento di iniziare a sfruttare le potenzialità del PLC.

In un macchinario o in un impianto industriale il verificarsi di un determinato allarme deve essere registrato e salvato in memoria in modo da informare l’operatore dell’eventuale problema, una volta che il problema verrà risolto l’operatore dovrà “dire” al PLC che l’allarme è stato risolto e che la macchina può ripartire.

Per far ciò dovremo iniziare a lavorare con le memorie interne del PLC.

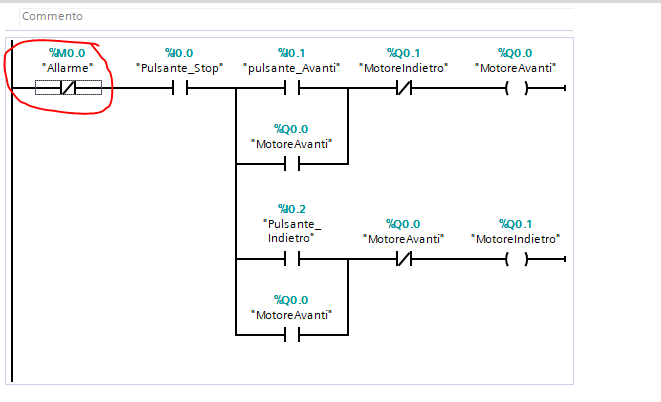
Come prima cosa dobbiamo andare a sostituire tutte le nostre condizioni di allarme con un’unica condizione, cioè un bit cumulativo di allarme.

Questo renderà molto più leggibile il nostro software in caso in cui abbiamo molti dispositivi e condizioni di allarme.

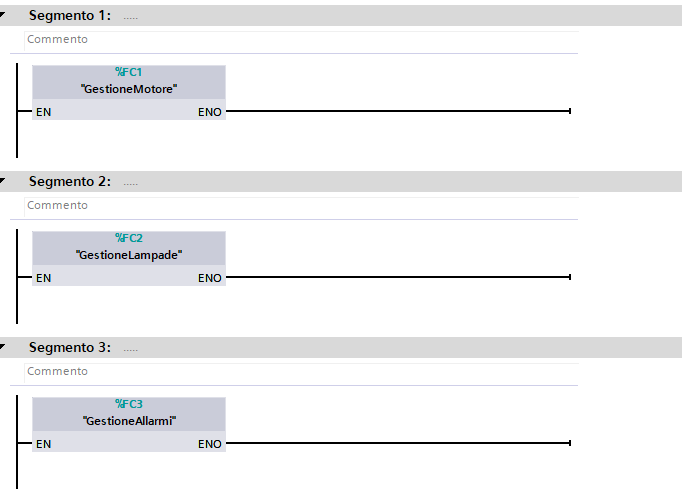
****

Il bit che andremo a creare dovrà essere dichiarato e istanziato in una determinata area di memoria con indirizzo a nostra libera scelta, questo perché come vedremo prossimamente, il pannello operatore ha bisogno di questa area di memoria come riferimento per poter mostrare i messaggi dei relativi allarmi all’operatore.

Nell’esempio la memoria allarme è stata dichiarata all’indirizzo M0.0 e quindi farà parte dell’area di memoria MW1 che è composta da 16 bit. (M0.0 / M1.7)

****

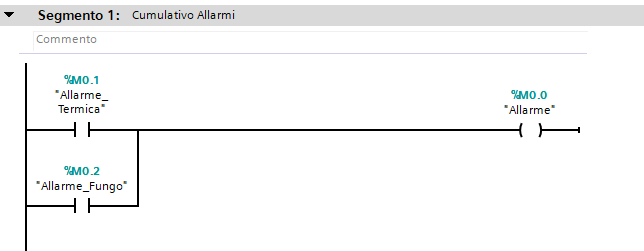
Ora che abbiamo creato e posizionato il nostro bit cumulativo di allarmi non ci rimane che gestire la logica relativa ad esso, per far ciò creeremo un’altra funzione destinata solo alla gestione degli allarmi. Ovviamente dobbiamo ricordarci di inserirla all’interno del nostro OB1.



Per prima cosa dovremo occuparci della gestione del nostro bit cumulativo di allarme.

Come detto in precedenza all’interno del PLC gli allarmi devono essere registrati e salvati perciò dovrò dichiarare e istanziare dei bit per ogni allarme possibile della macchina, i bit dovranno essere istanziati nella stessa area di memoria del bit cumulativo per poter essere mostrati successivamente a pannello.

Essendo un bit cumulativo esso verrà attivato da varie condizioni, in questo caso le nostre condizioni saranno tutti gli allarmi del nostro macchinario collegato tramite logica “OR”.

****

Adesso che abbiamo creato le memorie dei nostri singoli allarmi non ci resta che gestirli.

Abbiamo detto che un allarme deve rimanere in memoria e una volta risolto deve essere azzerato dall’operatore, per far ciò introduciamo 2 nuovi comandi relativi ai bit, il Set e il Reset di una memoria.

Al contrario di una bobina dove sia l’accensione che lo spegnimento vengono gestiti all’interno dello stesso “ramo” di logica utilizzando questi due comandi la gestione dell’accensione e dello spegnimento vengono separati.

Questo semplifica molto la lettura del programma, soprattutto quando la logica inizia a essere abbastanza complessa.

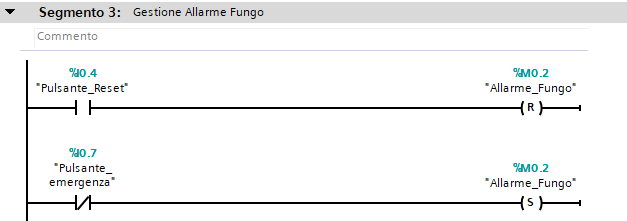
Per prima cosa ci occuperemo del comando di reset, di solito l’azzeramento degli allarmi avviene tramite un apposito tasto fisico oppure tramite pannello HMI, in questo caso useremo un tasto fisico.

Visto che non avevamo ancora considerato questo tipo di gestione degli allarmi dovremo aggiungere alla nostra configurazione hardware un altro pulsante su un altro ingresso libero.

Ora ci basterà mettere il tasto appena aggiunto in serie al comando di reset.

Mentre per l’attivazione dell’allarme ci basterà mettere le condizioni in serie al comando di set, in questo caso ci basterà mettere la mancata lettura del segnale elettrico del fungo di emergenza.

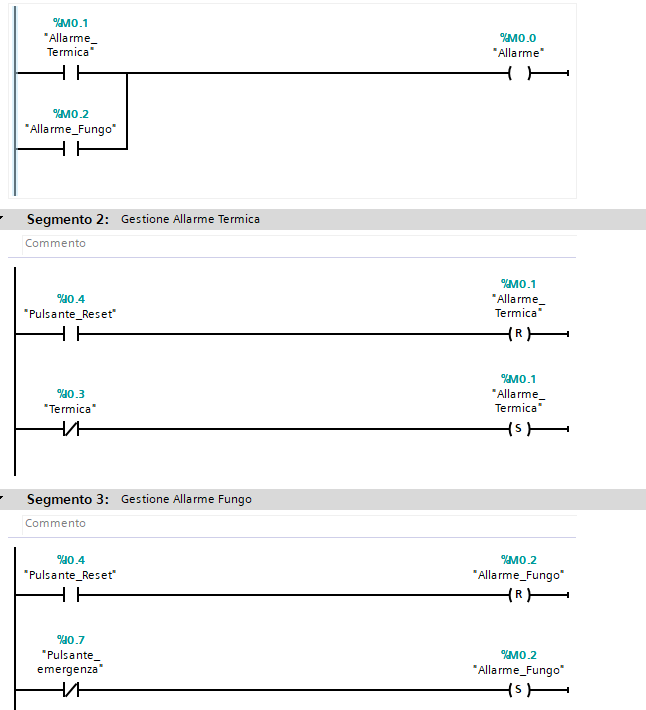
Ed ecco il risultato finale.

****

È bene ricordare che per quanto riguarda gli allarmi bisogna sempre inserire prima la condizione di reset di un allarme e poi il suo Set, questo per evitare problemi relativi alla lettura del programma da parte del PLC.

In questo modo anche se il pulsante di reset verrà premuto continuamente, in ogni caso la funzione di set dell’allarme avrà la prevalenza, quindi al termine del ciclo sarà che la memoria di allarme verrà attivata.

Ecco la FC gestione allarmi con anche l’allarme della protezione termica.

****

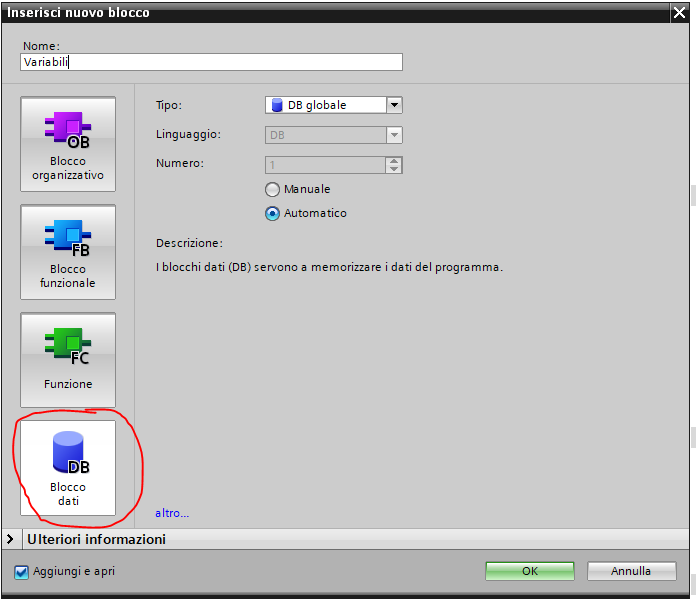
* 1. **Aggiunta conteggio allarmi e aggiunta contenitore variabili**

Se volessimo contare quante volte un determinato allarme è stato attivato avremmo bisogno di una funzione che effettua il conteggio e di una variabile per contenere tale informazione numerica.

Fino ad ora abbiamo lavorato solo con dei bit e non abbiamo avuto nessun problema nell’allocare la loro memoria ma andando a lavorare con dei valori numerici è molto più probabile fare degli errori di allocazione, per quanto Tia Portal cerchi di aiutarci.

Per questo andremo a vedere un metodo di creazione variabili senza bisogno di allocare l’area di memoria, questo strumento si chiama DB.

Per aggiungere una DB non dobbiamo far altro che aggiungere un nuovo blocco programma come se stessimo aggiungendo una FC

****

Una volta aggiunta la ritroveremo nella radice del nostro progetto e aprendola ci troveremo davanti a una tabella per l’inserimento variabili.

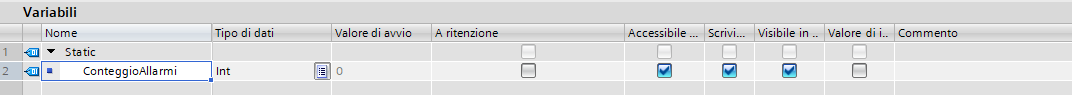
Come per la pagina variabili globali dove abbiamo dichiarato i nostri input, output e memorie anche qui possiamo dichiarare le nostre variabili dandogli un nome e la tipologia di dato.

In più possiamo decidere se la nostra variabile deve avere un determinato valore all’avvio del PLC oppure che quella variabile venga tenuta in memoria anche dopo lo spegnimento del PLC.

Come si può notare non c’è stato bisogno di allocare la variabile in nessuna area di memoria, ci penserà il programma a farlo automaticamente.

Consiglio di lavorare sempre con le DB per qualsiasi variabile del progetto a parte ovviamente degli ingressi e le uscite e l’area di memoria per gli allarmi.

È possibile creare più DB, questo rende anche più ordinato il software, in questo modo si possono dividere le variabili in base alla loro appartenenza e al loro utilizzo.

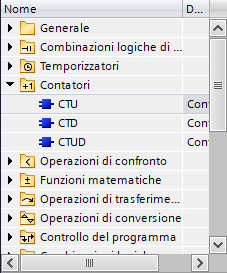
****

Ecco come accedere alla variabile appena creata, prima di tutto bisogna scrivere il nome della nostra DB e dopo di che aggiungendo un punto ci verrà mostrato l’elenco di tutte le variabili al suo interno.

****

Ora che abbiamo creato la nostra variabile di tipo “int” per tenere il conteggio degli allarmi non ci resta che aggiungere il nostro contatore.

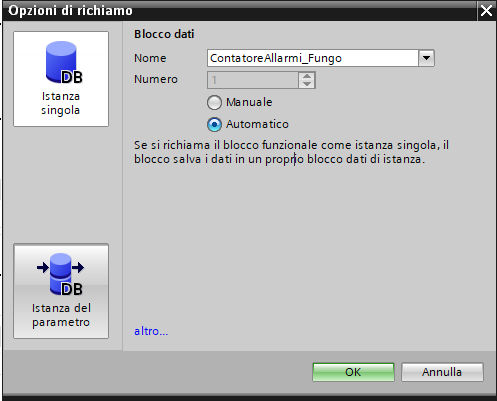
Nel rung subito dopo l’allarme del fungo di emergenza aggiungere un “CTU”, lo si può trovare nel menu contestuale a destra.

****

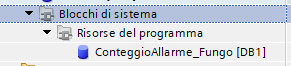
Una volta inserito il contatore nel nostro programma ci apparirà una finestra.

Questa finestra ci chiederà di creare una istanza del nostro contatore, questo perché il contatore che stiamo per utilizzare dovrà avere vita propria e ha bisogno della sua propria area di memoria per salvare le sue informazioni. Questo ovviamente varrà per ogni altro contatore che inseriremo nel programma.

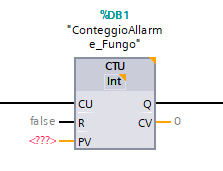
Infatti se si guarda attentamente lo screenshot qui sotto il programma ci sta chiedendo di creare una DB con nome proprio per quel determinato contatore.



Una volta creata la sua istanza Tia Portal inserirà in un sottomenù la DB che abbiamo appena creato, questa voce la si può trovare in fondo ai nostri blocchi programma.



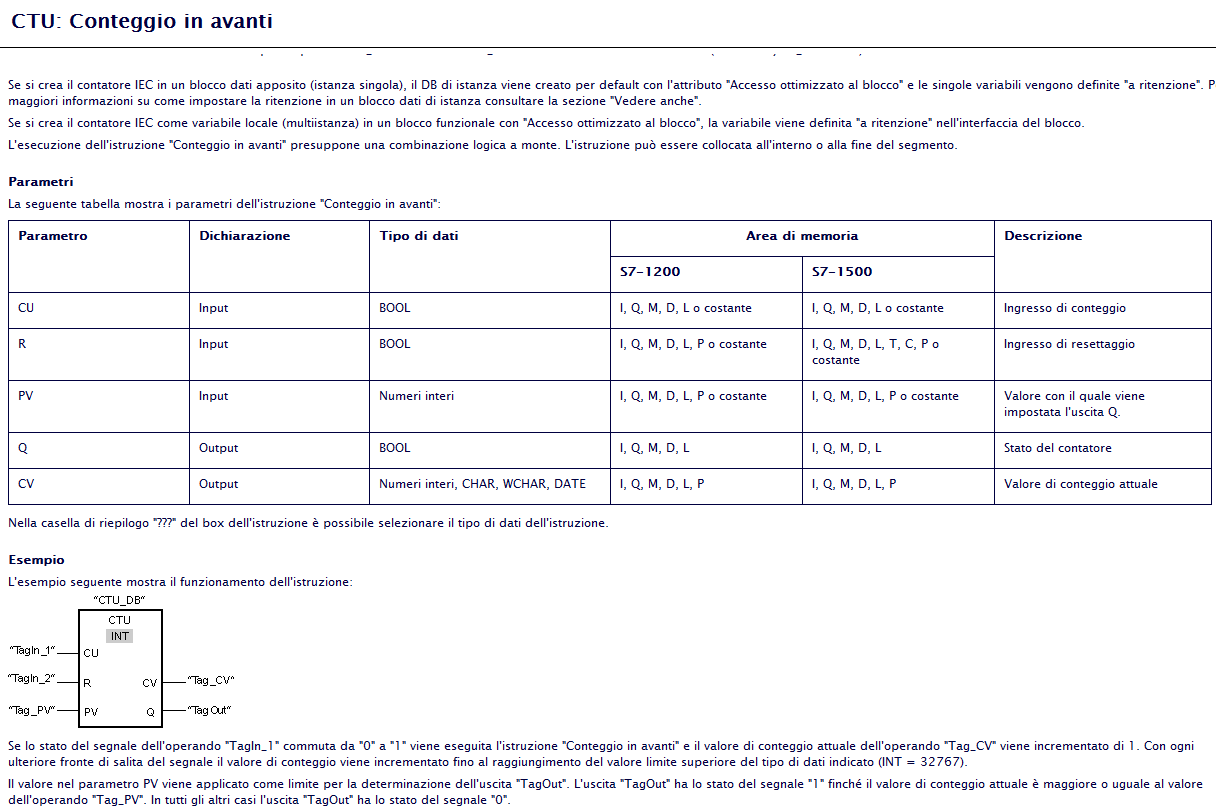
Dopo aver istanziato il nostro contatore ci ritroveremo il suo blocco all’interno del nostro programma ladder, come si può notare il nome dell’istanza che abbiamo appena dato è proprio su di lui proprio per indicare che stiamo utilizzando quel contatore.



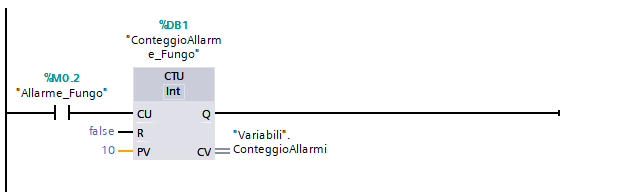
La funzione contatore ci richiedere 3 ingressi e ci restituisce 2 uscite.

A prima vista ovviamente non si capisce come utilizzare questo blocco, Siemens ha incorporato dentro Tia Portal una guida dove è presente la spiegazione per qualsiasi funzione, la guida è accessibile cliccando F1

Ecco uno screenshot preso direttamente dalla guida dove viene spiegato come utilizzare il contatore.



Eccome come parametrizzare un contatore in modo da contare il numero di allarmi:



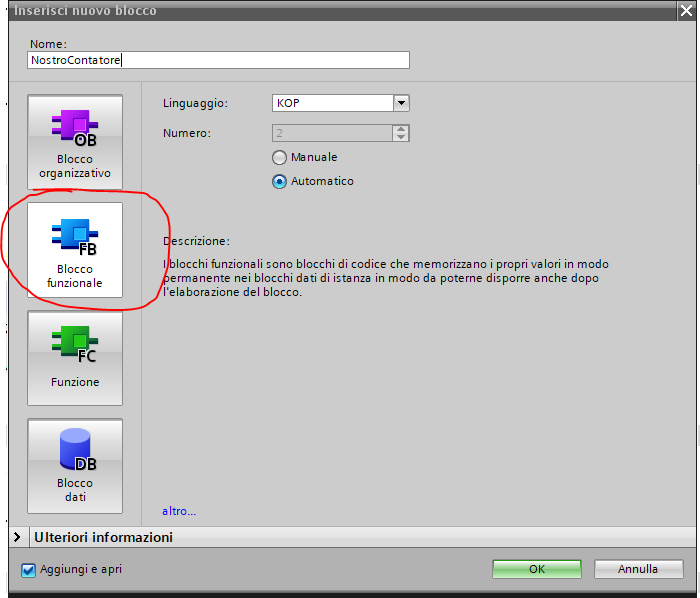
* 1. **Creazione funzione ritentiva**

Gli oggetti che abbiamo a nostra disposizione per programmare, come ad esempio il contatore appena utilizzato, sono oggetti creati dalla Siemens per aiutarci e semplificare la programmazione.

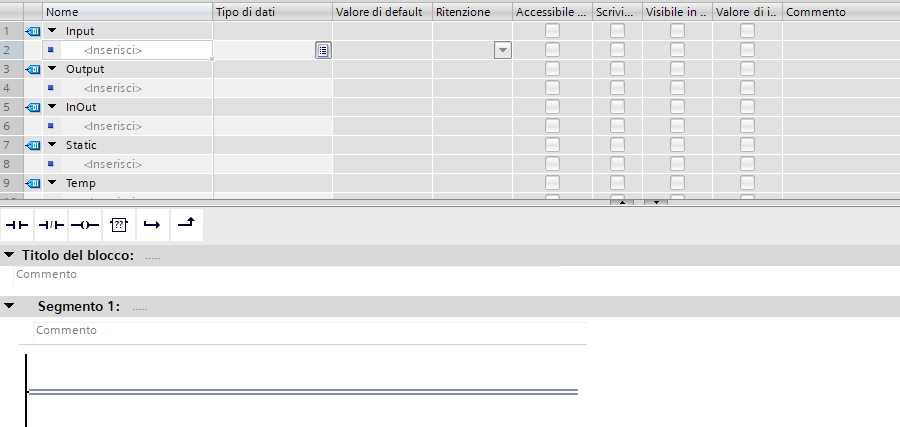
Tramite TiaPortal possiamo creare i nostri oggetti personali che potremo utilizzare allo stesso modo in cui usiamo gli oggetti inclusi nel programma.

Per creare un nostro blocco programma dobbiamo inserire un nuovo tipo di blocco, questo tipo di blocco si chiama FB (Function Block).

Da notare che la FB viene rappresentata con lo stesso disegno di una FC ma con il colore di una DB, questo per indicare che sarà una funzione con memoria allocata.

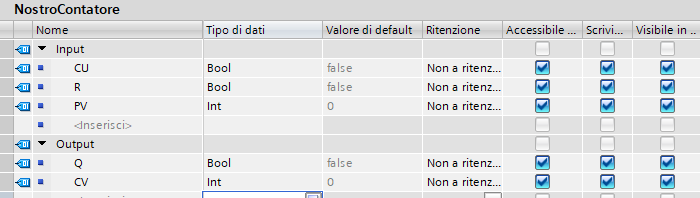
****

Una volta creata la FB ci ritroveremo davanti il nostro programma vuoto, proprio come se stessimo scrivendo una FC.



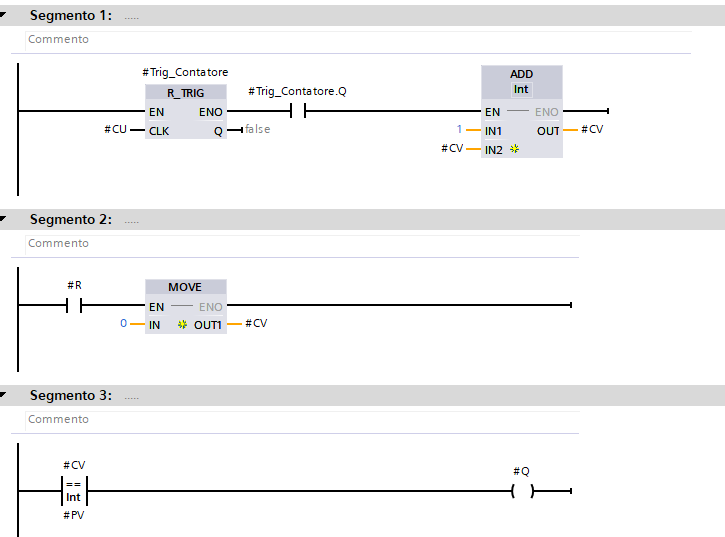
La prima cosa da fare in una FB è dichiarare i suoi ingressi e le sue uscite, nel menù collocato sopra i segmenti della logica del programma è possibile dichiarare le variabili unicamente appartenenti alla funzione che abbiamo appena inserito.

In questo caso ho parametrizzato gli ingressi e le uscite in modo che siano uguali a quelle del contatore che abbiamo appena utilizzato, ovviamente avrei potuto chiamare le mie variabili in qualsiasi altro modo.



Dopo aver creato i miei ingressi e uscite non mi resta che iniziare a lavorare alla logica, in questo caso è stata riprodotta la logica del contatore.

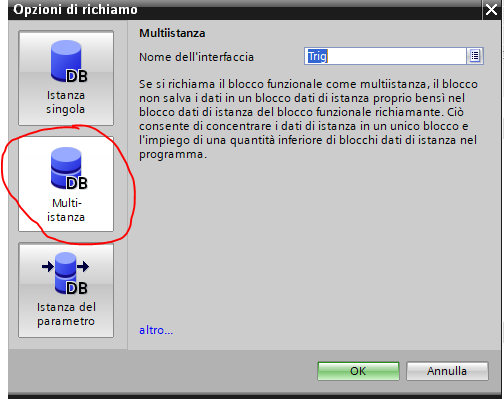
Da notare che sono state utilizzate solo le variabili interne della FB, infatti una variabile interna è sempre preceduta dal simbolo #.



NB: Quando all’interno di una FB si utilizzano oggetti che hanno la propria area di memoria (come ad esempio il blocco “R\_Trig” qui sopra riportato) bisogna stare attenti a dove allocare la sua memoria.

Quando andiamo a inserire uno di questi oggetti ci si aprirà la solita finestra che ci chiederà di dare un nome alla sua istanza, ma in questo caso dovremo selezionare la voce “Multi-istanza”, questo farà il modo che l’istanza di quell’oggetto sia posizionata all’interno della nostra FB e non nella cartella generale “blocchi di sistema”.

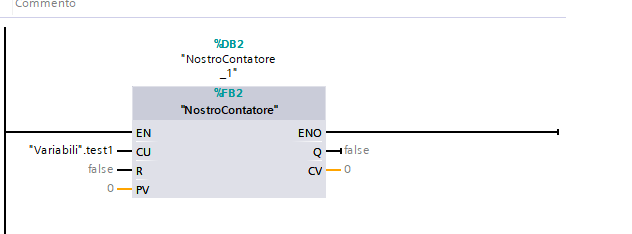
In questo modo ogni FB da noi creata conterrà al suo interno l’istanza di tutte le FB contenute al suo interno. Se non facessimo questo passaggio e allocassimo la memoria dell’oggetto nelle memorie generali del programma, ogni FB da noi creata utilizzerebbe la stessa istanza di quell’oggetto, andando a compromettere il funzionamento della nostra logica.



Istanza dell’oggetto appena creato all’interno della FB. All’interno della voce Static possono essere dichiarate sia altre FB che semplici variabili destinate all’uso interno (quindi non sono ne input ne output)

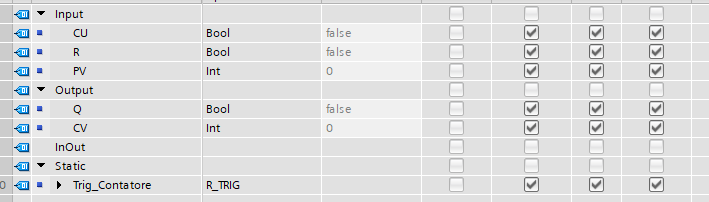


Una volta finito con la programmazione del nostro blocco non ci resta che richiamarlo, ovviamente essendo un oggetto con propria memoria al suo inserimento ci verrà chiesto di allocare una DB per il suo funzionamento.



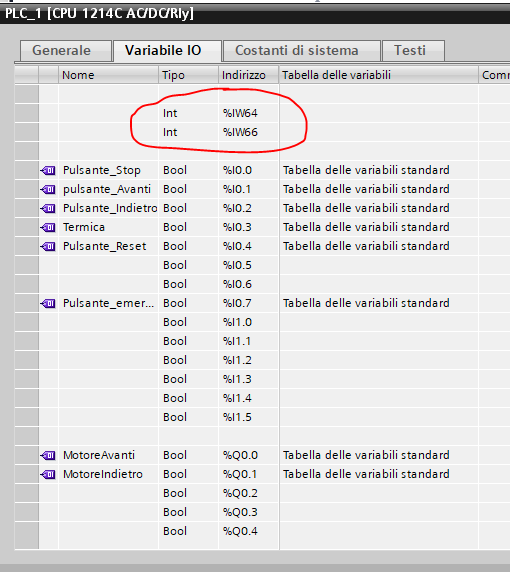
La DB non verrà più posizionata nella cartella “blocchi di sistema” ma verrà inserita nella cartella principale dei nostri programmi. Potremo aprire l’istanza della FB e da qui potremo controllare qualsiasi valore all’interno di quella istanza, anche il valore di tutti gli oggetti istanziati al suo interno.





* 1. **Configurare un sensore analogico**

Il nostro PLC ha a disposizione due ingressi analogici, è possibile trovare il loro indirizzo fisico all’interno delle proprietà del PLC alla voce “variabili I/0”. Qui saranno disponibili tutti gli indirizzi fisici disponibile nella nostra configurazione hardware.

****

Come si può notare l’area di memoria messa a disposizione per un ingresso analogico non è più un bit ma una variabile intera, questo perché ovviamente un sensore analogico può rappresentare valori.

L’area di memoria ha dimensioni di una Word, quindi può rappresentare 65535 valori.

In questo caso la tipologia di variabile utilizzata è un integer, quindi il valore può essere anche negativo, in questo caso il valore rappresentato andrà da – 32768 a 32767.

Il nostro ingresso lavora in un range da 0 a 10 volt quindi in teoria il valore rappresentato all’interno del PLC non dovrebbe mai essere inferiore allo 0.

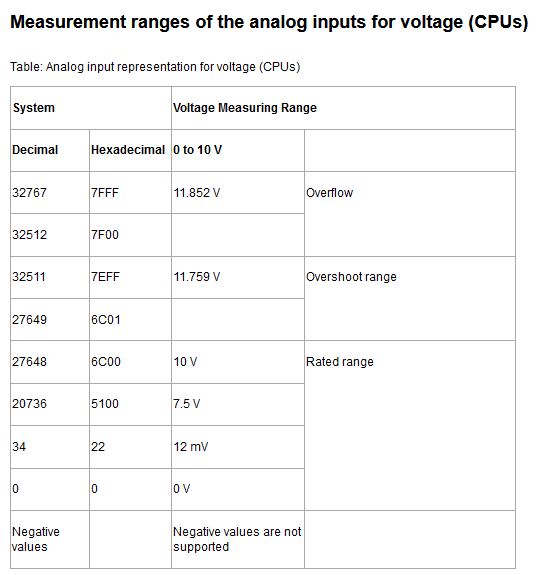
Quindi il valore in tensione di 0-10 volt verrà rappresentato all’interno del PLC in un range da 0 a 32767

**NB per Siemens:** in tutti gli altri dispositivi il range è da 0 a 32767, in Siemens questo range viene ristretto da 0 a 27648, questo avviene perché ci viene lasciato un range di Overflow, cioè ci viene lasciata un’area di valori per poter leggere errori relativi a una sovratensione.

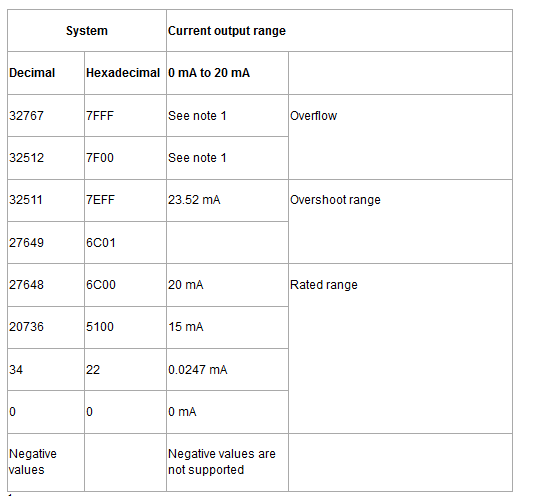
Ad esempio se in ingresso avrò una tensione di 11 volt il mio valore interno andrà sopra il limite di 27648 e grazie a questa informazione posso capire che c’è un problema elettrico.

Se il range fosse 32767 e avessi una tensione di ingresso di 11 volt non avrei modo di notarlo.

**Tabella dei valori:**

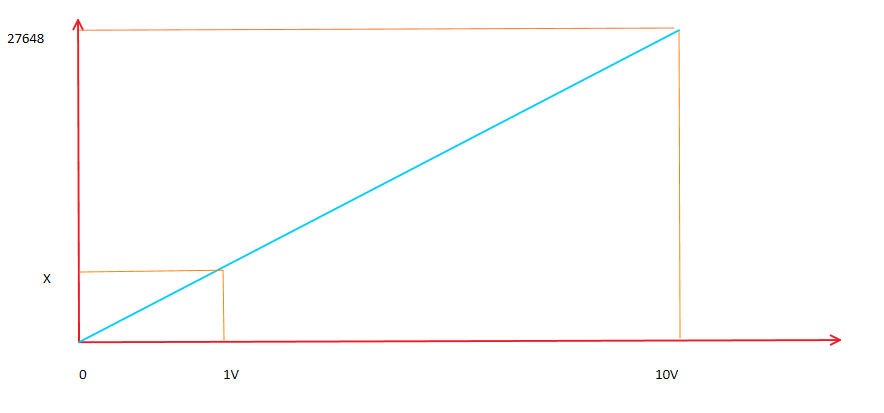


**Lo stesso discorso vale anche per le uscite analogiche.**



Dopo aver creato la variabile e averlo allocata sull’ingresso analogico %IW non ci resta che andare a calcolare il valore dei volt che effettivamente abbiamo in ingresso.

Come detto prima sappiamo che quando avremo 10 Volt in ingresso la nostra variabile dovrà valere 27648, per trovare il valore del singolo volt mi basta fare una semplice operazione.



27648 = 10v \* X

X = 27648 / 10 X = 2764,8

Ora che ho trovato il coefficiente angolare non mi resta che prendere il mio ingresso analogico e dividerlo per quel valore e posizionare il risultato su una terza variabile interna di tipo Real, la variabile deve essere di questo tipo perché le variabili Real sono in grado di mostrare anche i valori con le virgole.

ValoreSensore := IngressoSensore / 2764,8

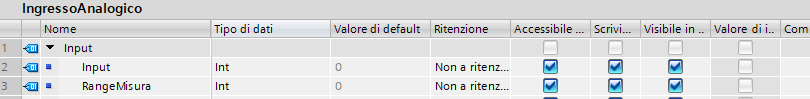
In questo caso se ho in ingresso 1V la variabile “IngressoSensore” varrà esattamente 2764,8, quindi il valore della terza variabile “ValoreSensore” sarà uguale a 1.

Nel caso avessi in ingresso 0.5V la variabile “IngressoSensore” varrà 1382,4 e quindi il mio risultato sarà 0.5

Ora che sappiamo il range dei valori con il quale lavorare andiamo a costruire la nostra funzione per gestire un sensore analogico.

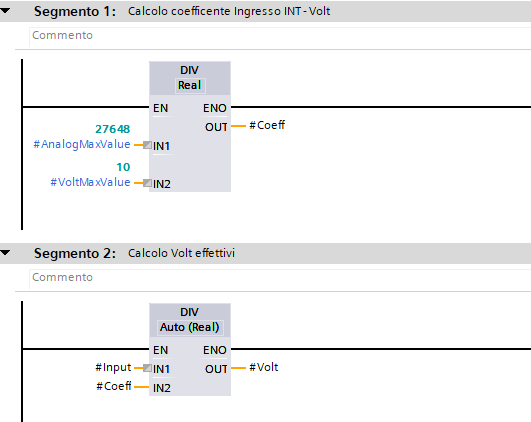
In questo caso creerò una FB in modo da poter utilizzare la stessa funzione per N sensori analogici e per evitare qualsiasi tipo di problema.

Dopo aver aggiunto la Function Block nel programma devo dichiarare gli ingressi e le uscite.



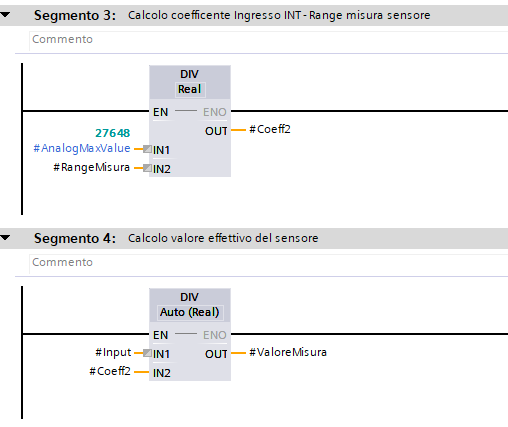


Una volta dichiarate le variabili non mi resta che sviluppare la logica, in questo caso userò una variabile di appoggio che deve essere dichiarata all’interno del blocco nella voce “Costant”



In questo modo avrò in output il valore in Volt.

Dopo aver calcolato il range in volt non ci rimane che calcolare il range del valore effettivo della misura rilevata dal sensore (pressione, temperatura, ecc,), Ovviamente dovrò dichiarare e calcolare un altro coefficiente.



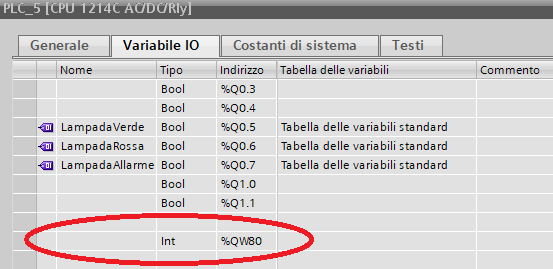
Una volta finito con al scrittura della nostra FB inserirla all’interno del programma dichiarando la sua istanza e provare a parametrizzarla, la funzione può essere simulata anche fornendo in ingresso una semplice variabile di tipo INT e non per forza un sensore analogico.



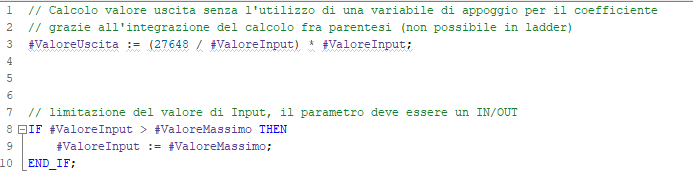
* 1. **Uscita analogica**

Ora che abbiamo lavorato con un ingresso digitale andiamo a creare una FB per una uscita analogica.

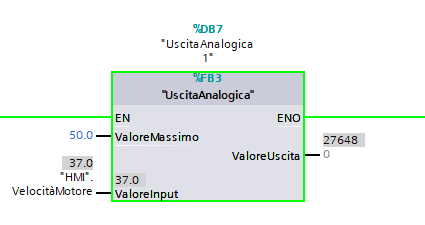
Prima di tutto dobbiamo trovare l’indirizzo della nostra uscita analogica e dichiarare una variabile in tale indirizzo, lo si può trovare sempre nel menu “Variabili IO” nelle proprietà del PLC.



Dopo aver dichiarato una variabile non ci resta che creare una FB dedicata alle uscite analogiche, la nostra FB avrà bisogno di due ingressi, un ingresso per indicare il limite del valore reale che la nostra uscita dovrà avere, una variabile sia in ingresso che in uscita per indicare il valore effettivo del nostro segnale analogico (deve essere in/out perché dobbiamo avere la possibilità di poter correggere questo valore) e in uscita ci sarà il nostro valore in Integer per comandare l’uscita.



Ecco il risultato finale:

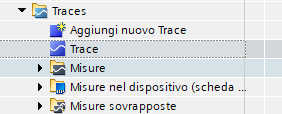


* 1. **Trace e monitoraggio valori**

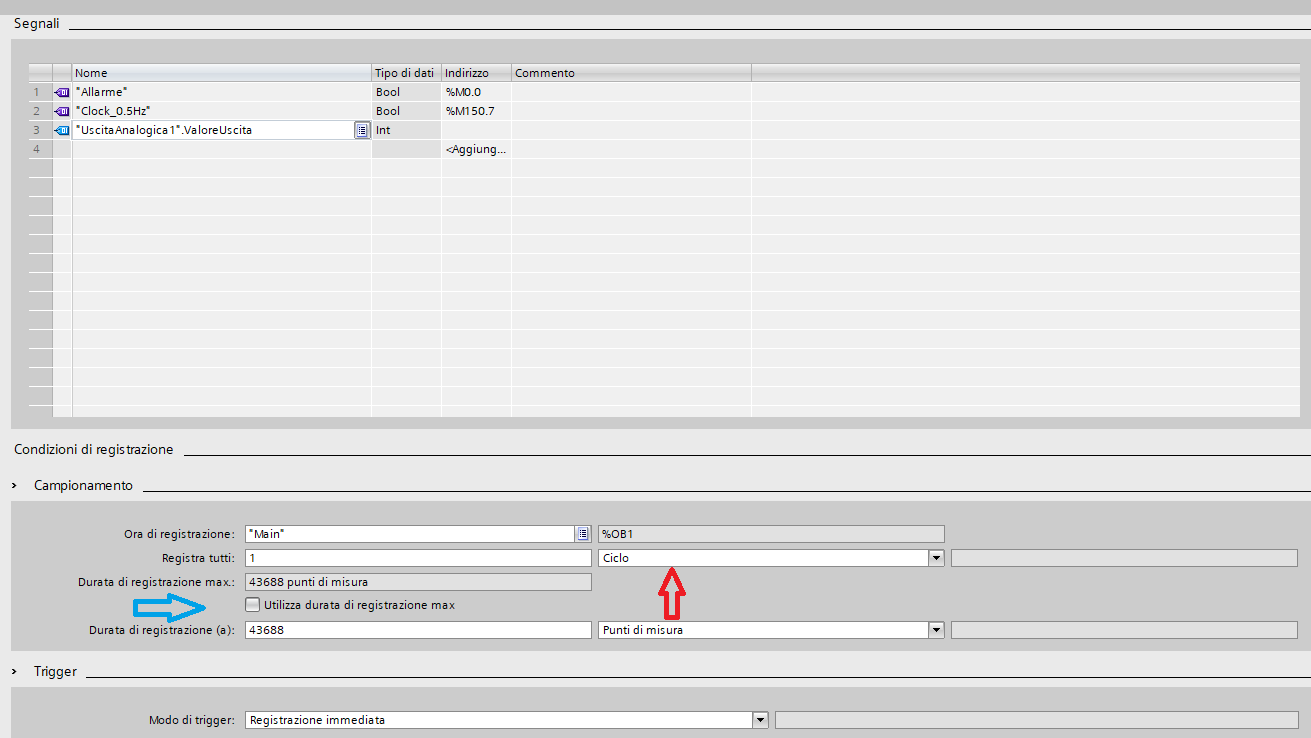
All’interno di TiaPortal è presente un particolare oggetto che ci permette di tenere traccia nel tempo dell’andamento di più valori simultaneamente, questo è molto utile per tenere sotto controllo il comportamento dei nostri vari dispositivi e delle nostre memorie nel tempo.

**NB:** Non tutti i PLC posso usufruire di questa funzione, se il controllore che state utilizzando non supporta la funzione Trace la sua voce non sarà proprio presente all’interno della radice di progetto in TiaPortal.

Una volta trovata la cartella “Traces” basta aggiungere un nuovo Trace e come se avessimo creato una nuova FC il nuovo trace apparirà nel nostro progetto.



Appena creato il nostro Trace ci apparirà la sua schermata di configurazione



* Nella parte superiore è possibile aggiungere tutte le variabili del nostro progetto che vogliamo monitorare simultaneamente nello stesso grafico, ovviamente possono essere dati di tipologia diversa.
* Nella parte centrale “Campionamento” è possibile impostare la frequenza di campionamento e il tempo di durate del Trace.
* Nella prima voce “Ora di registrazione” si indica quale funzione di sistema deve essere presa come riferimento per conteggiare i cicli di registrazione, (per il momento nel nostro progetto è presente solo il nostro Main OB).
* Alla seconda voce “Registra Tutti” si può impostare la frequenza di registrazione, in questo caso viene impostato che la registrazione avvenga ad ogni ciclo del nostro Main.
* Togliendo la spunta alla voce “Utilizza durata di registrazione max” è possibili decidere la durata della registrazione.

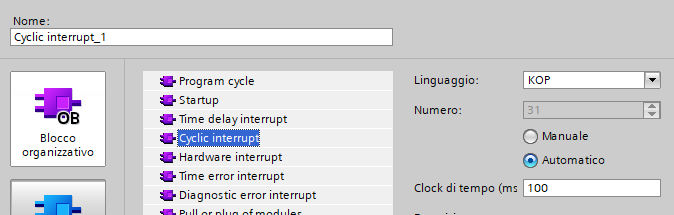
Da notare che la durata della registrazione è basata sull’unità di misura che viene indicata nella voce “Registra tutti” ad esempio nel nostro caso il nostro singolo punto di misura è il singolo ciclo del nostro programma principale.

* Nell’ultimo menù “Trigger” è possibile decidere se far partire manualmente la registrazione (Registrazione immediata) oppure far partire la registrazione dopo l’attivazione di un determinato evento (Es: un bit a 1, una determinata variabile a un determinato valore, ecc)

Visto che la nostra funzione di sistema Main verrà eseguita dal PLC con un tempo prestabilito automaticamente e potenzialmente variabile ciclo per ciclo è molto difficile prestabilire quanto durerà il nostro tracciamento delle variabili.

Per aggirare questo problema si può aggiungere al progetto un’altra funzione di sistema di diversa tipologia, una funzione “Cyclic Interrupt”

È possibile aggiungere una funzione di sistema sempre dalla stessa finestra dalla quale abbiamo aggiunto le nostre FC, DB ed FB.

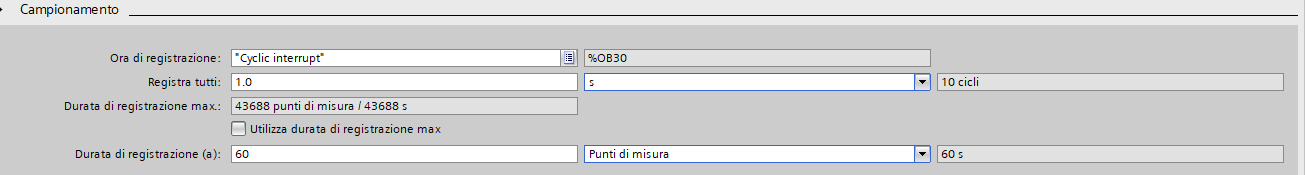


Come si può notare, oltre al nome e al tipo di linguaggio è presente un parametro per determinare ogni quanto lasso di tempo il nostro controllore dovrà richiamare la nostra funzione Cyclic interrupt.

Quando questa funzione viene richiamata, il PLC smetterà di eseguire temporaneamente il programma principale e si dedicherà ad eseguire le istruzioni all’interno del nostro nuovo blocco di sistema, una volta finito il controllore tornerà a rileggere il programma principale proprio nel punto dove si era fermato.

Una volta aggiunta la nostra funzione possiamo tornare nella configurazione del Trace e sostituire in “ora di registrazione” il nostro Main con la funzione ciclica.

Una volta fatto ciò nella voce inferiore sarà possibile selezionare l’unità di misura, adesso oltre al ciclo è presente anche “s” (secondi).

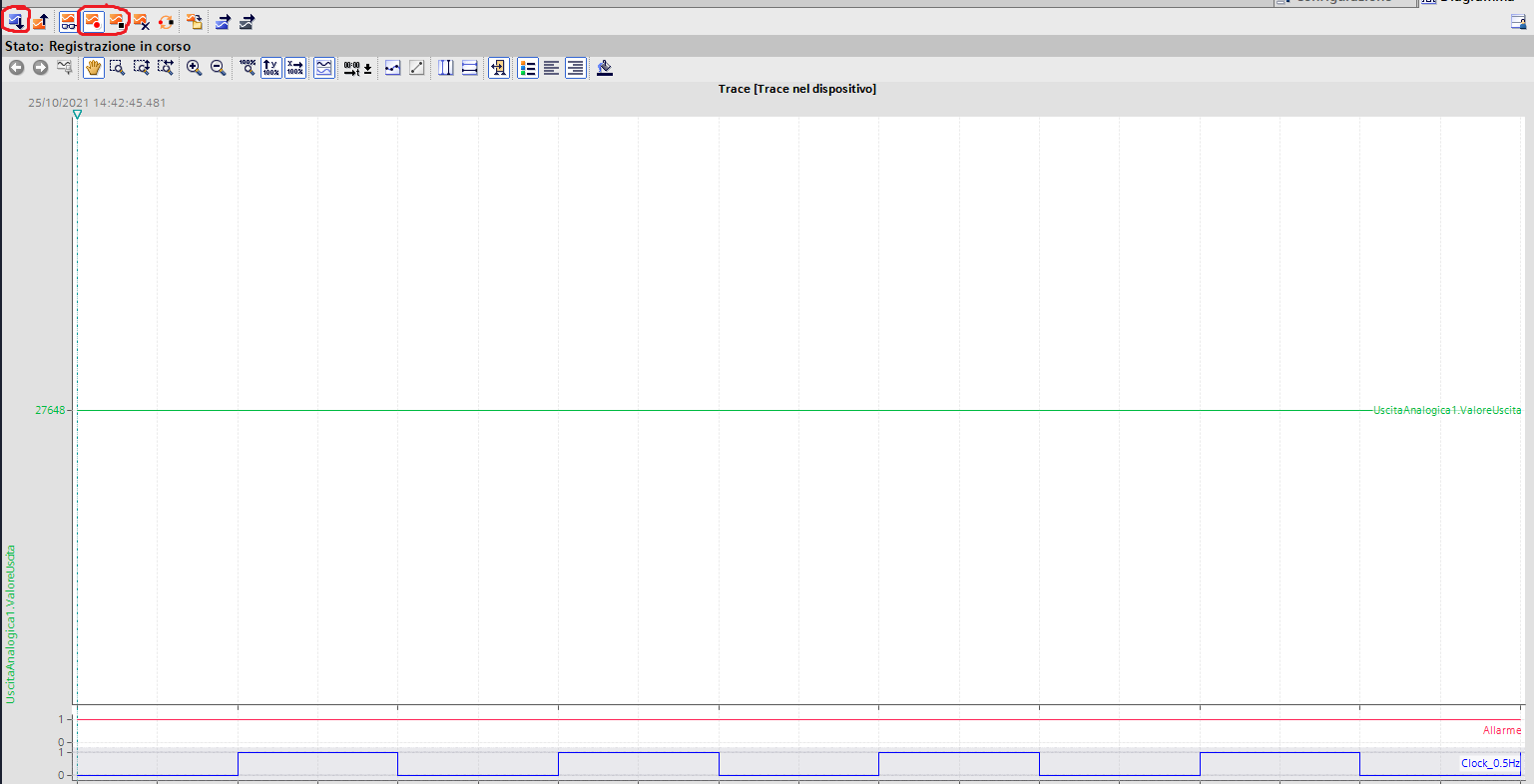


Una volta finito con la configurazione si può passare al grafico, in alto a destra è possibile passare al menu “diagramma”.

Una volta arrivata all’interfaccia del diagramma il primo passo da eseguire è fare il download del Trace, per far ciò basta cliccare sulla prima icona a sinistra.

Il download del trace non interferirà in nessun modo con il funzionamento del controllore, questo perché è una funzione che viene eseguita in parallelo.

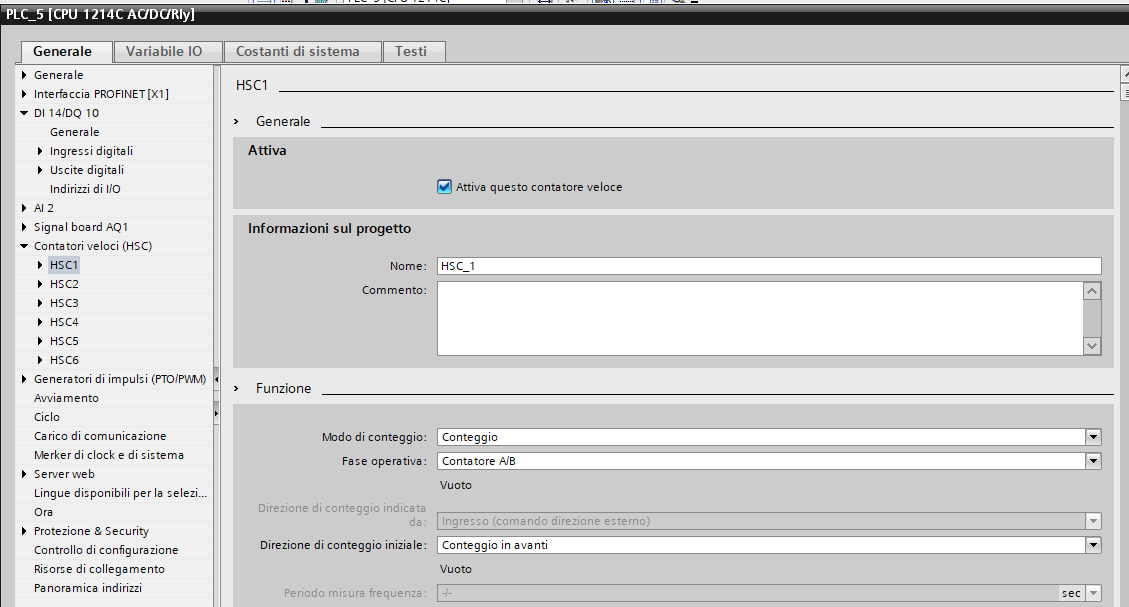
Una volta eseguito il download non resta che far partire la registrazione tramite l’apposito pulsante.



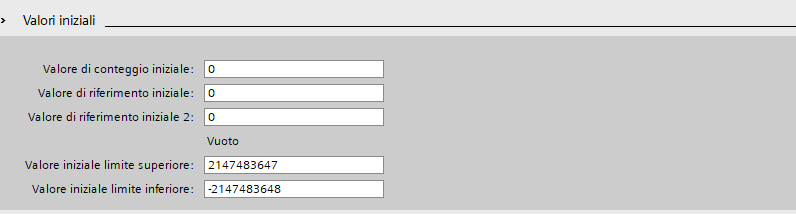
* 1. **Gestione Encoder**

Dopo aver cablato l’encoder è necessario attivare la funzione interna dedicata al conteggio, se il nostro PLC supporta questa funzione è possibile abilitarla nelle sue proprietà alla voce HSC come allegato nello screenshot.

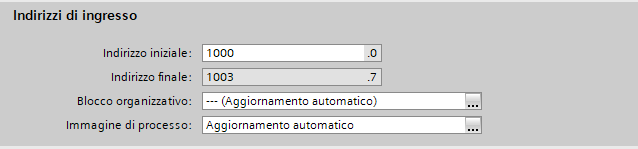
Una volta attivato è possibile iniziare a configurarlo, il primo menù di impostazioni è relativo alla tipologia di cablaggio ed encoder che si andrà ad utilizzare.



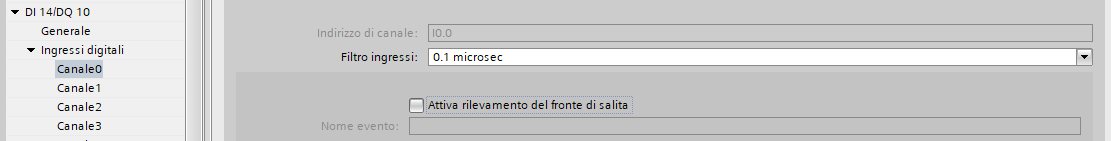
Dopo di che si potrà impostare il valore da dove il nostro conteggio ad accensione PLC dovrà essere inizializzato e gli ingressi che andranno a leggere i segnali.

****

E come ultima impostazione abbiamo la possibilità di modificare l’area di memoria dove il valore del nostro conteggio verrà salvata. L’area di memoria in questo caso partirà da %ID1000.

****

Una volta terminata la configurazione del nostro contatore HSC dobbiamo modificare il tempo di lettura dei due ingressi che sono stati selezionati, questa impostazione è possibile trovarla all’interno delle impostazioni di ogni singolo ingresso sempre all’interno delle proprietà del PLC.

****

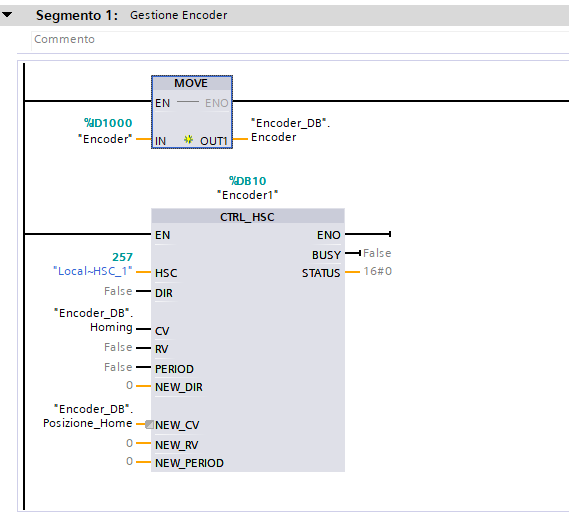
Una volta configurato il tutto è possibile iniziare a gestire via software il nostro conteggio.

Per poter Azzerare il conteggio del nostro encoder dopo una procedura di inizializzazione della posizione dell’oggetto collegato al nostro encoder dobbiamo utilizzare la FB di sistema “ CTRL\_HSC”.

Per prima cosa dobbiamo collegare l’area di memoria speciale dedicata al conteggio per la gestione del nostro encoder, queste aree di memorie le si possono trovare nel menù “costanti di sistema” all’interno del menù delle variabili interne del nostro PLC”.

La variabile “Local HSC1” è la variabile che fa riferimento al contatore che abbiamo appena attivato e deve essere inserito come primo ingresso alla nostra FB.

Dopo di che dovremo utilizzare l’ingresso CV e l’ingresso New\_CV, il primo è un ingresso booleano e qui dovrà essere collegata l’informazione di fine procedura di inizializzazione, mentre nell’ ingresso New\_CV è possibile andare ad indicare la posizione che dovrà essere impostata all’interno dell’area di memoria del contatore dopo l’attivazione dell’ingresso CV.

****

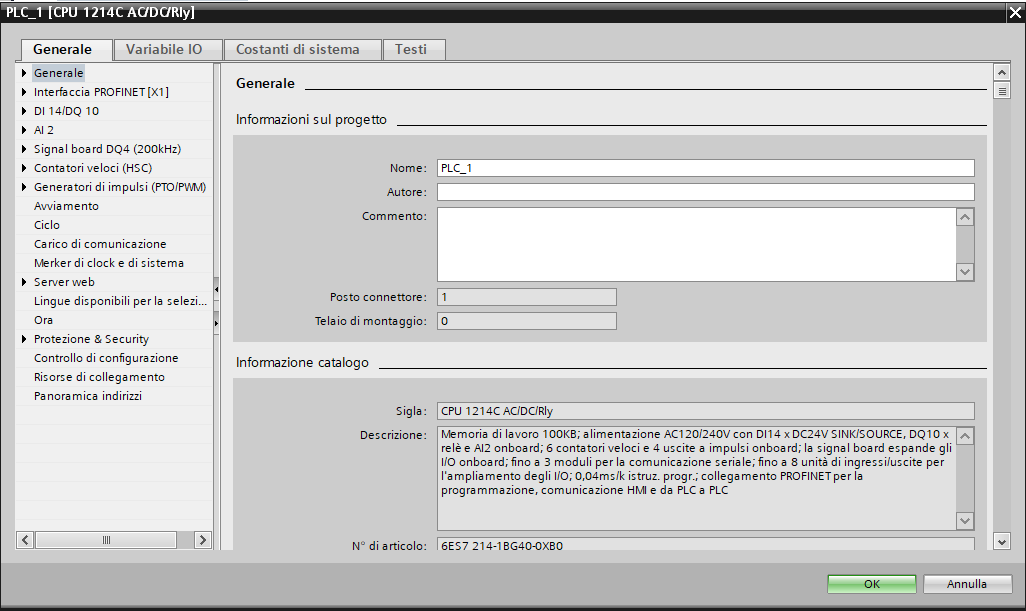
* 1. **Configurazione per il download**

Adesso che abbiamo finito di scrivere il nostro programma non ci resta che testarlo, nella prima parte della guida abbiamo simulato il nostro programma tramite l’apposito simulatore, questa volta andremo a eseguire il download su un PLC reale.

Per far ciò prima devono essere configurati alcuni parametri.

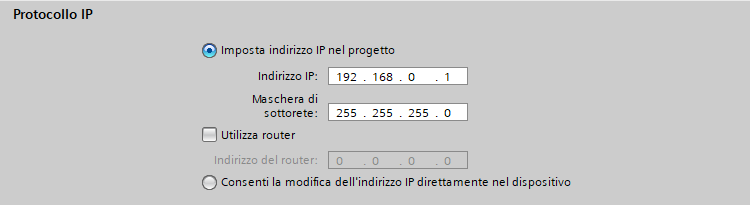
Cliccando nella cartella principale del nostro programma PLC con il tasto destro si può accede alle sue proprietà





La prima modifica che è possibile effettuare è la modifica del nome del PLC, in questo modo quando andremo a ricercare in rete la nostra CPU potremo riconoscerla velocemente.

La seconda modifica necessaria per effettuare il download è relativa all’indirizzo IP del dispositivo, questa impostazione la si può trovare nel secondo menu “Interfaccia PROFINET” (Profinet è il protocollo di comunicazione proprietario Siemens)

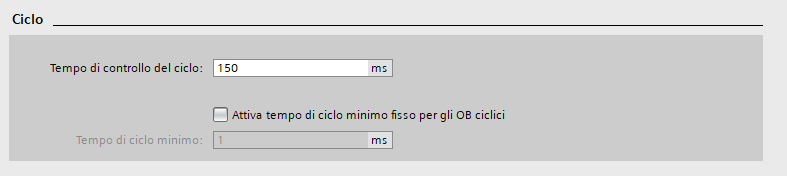


Dopo aver impostato l’indirizzo IP e la relativa maschera di sottorete è possibile eseguire il download eseguendo gli stessi passaggi che sono stati fatti per il download utilizzando il simulatore incorporato.

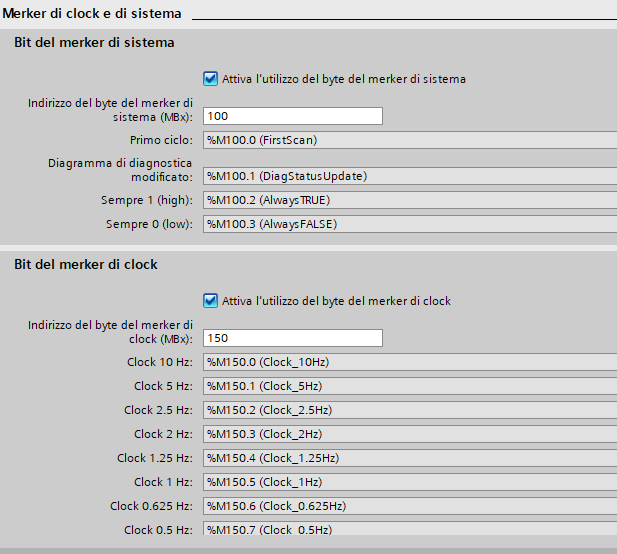
Visto che ormai siamo all’interno delle proprietà del PLC esploriamo qualche altro menù molto importante.

Ciclo: Qui è possibile impostare il tempo di controllo (WhatchDog), questo tempo è la durata massima che può avere un ciclo di PLC, in caso di superamento di questo limiti il sistema andrà in blocco.

Possono allungare il tempo ciclo varie chiamate ad altre OB di sistema che verranno chiamate nel ben mezzo della lettura del programma principale (Ad esempio le OB con interrupt ciclico) oppure degli errori di scrittura del programma, ad esempio in caso di utilizzo di una funzione ricorsiva (effetto matriosca).



**Merker di clock e di sistema**: in questo menù è possibile abilitare alcune memorie speciali messe a disposizioni da Siemens.



Il primo gruppo, i bit di sistema, sono 4 bit che hanno queste specifiche funzioni:

* Primo ciclo: è un bit che al primo ciclo dopo l’avviamento del PLC è a True e a partire dal secondo ciclo in poi sarà sempre a False e non sarà più possibile riportarlo all’altro stato, Può essere molto utile per eseguire parametrizzazioni a inizio macchina.
* Diagramma di diagnostica modificato: Questo bit rileva il fronte di salita di un errore grave del dispositivo o del sistema, il fronte di salita può essere sfruttato per segnalare un allarme e magari salvare determinati parametri in modo da effettuare una “fotografia” dello stato di alcune variabili in modo da semplificare il debug.
* Sempre 1 e Sempre 0: Sono bit che per l’appunto rimangono sempre attivi o disattivati e il loro stato non può essere alterato.

Sono molto utili per forzare o bypassare porzioni di programma o dispositivi di ingressi e uscite.

In alternativa a questi bit si può utilizzare direttamente il comando “False” o “True”, effettivamente a lato pratico avrebbero lo stesso effetto, il problema è che non essendo delle variabili il loro utilizzo non può essere tracciato all’interno del programma.

I Merker di clock invece sono memorie speciali che lampeggeranno con frequenza variabili e possono essere usati nei casi più disparati, possono essere utilizzati per far lampeggiare una lampadina oppure per far si che alcune funzioni vengano chiamate con una determinata frequenza.

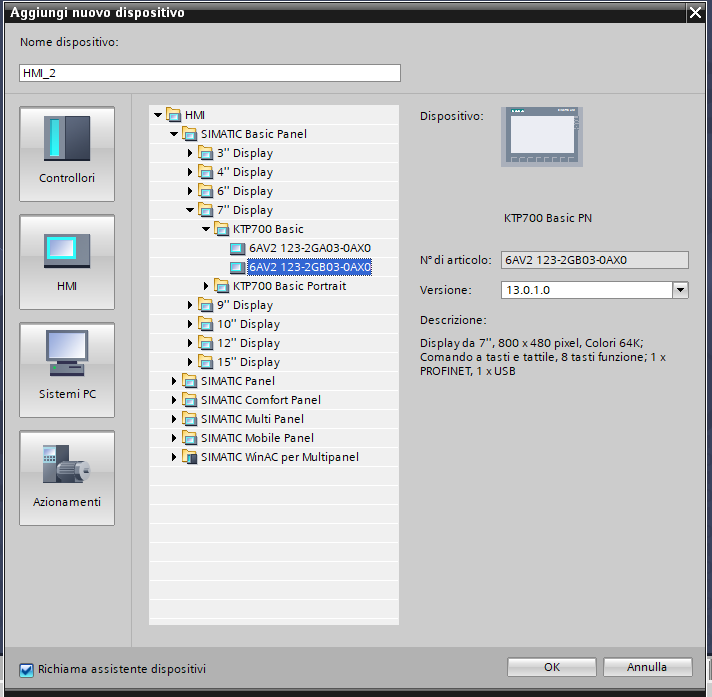
Come si può notare all’attivazione di queste memorie speciali ci verrà chieste il loro indirizzo di allocazione, ovviamente bisogna stare attenti a dove posizionare la loro area di memoria.

**4. Aggiunta pannello**

Per aggiungere un pannello HMI dobbiamo andare nel menù laterale di sinistra e cliccare sulla voce in alto “Aggiungi nuovo dispositivo”.

Si aprirà lo stesso menù che abbiamo utilizzato per aggiungere il nostro PLC ma stavolta dovremo andare non più sulla voce “Controllori” ma sulla voce HMI.

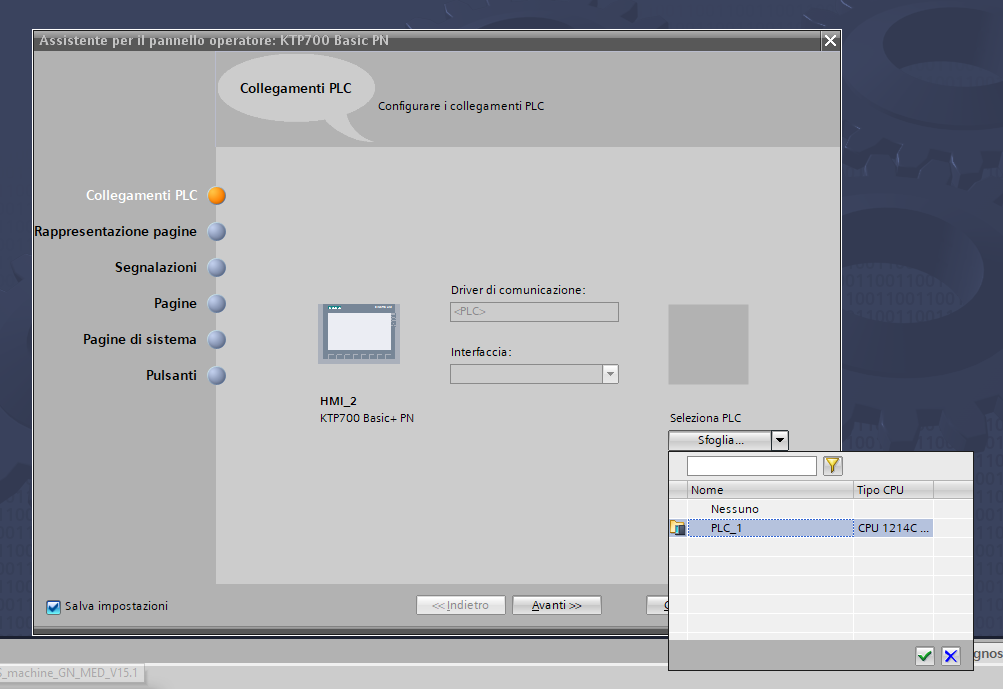
Ora possiamo selezionare il nostro pannelloleggendo il codice che è presente sul retro di esso, trovato il modello premere ok e subito dopo verremo portati nel Wizard di configurazione HMI.

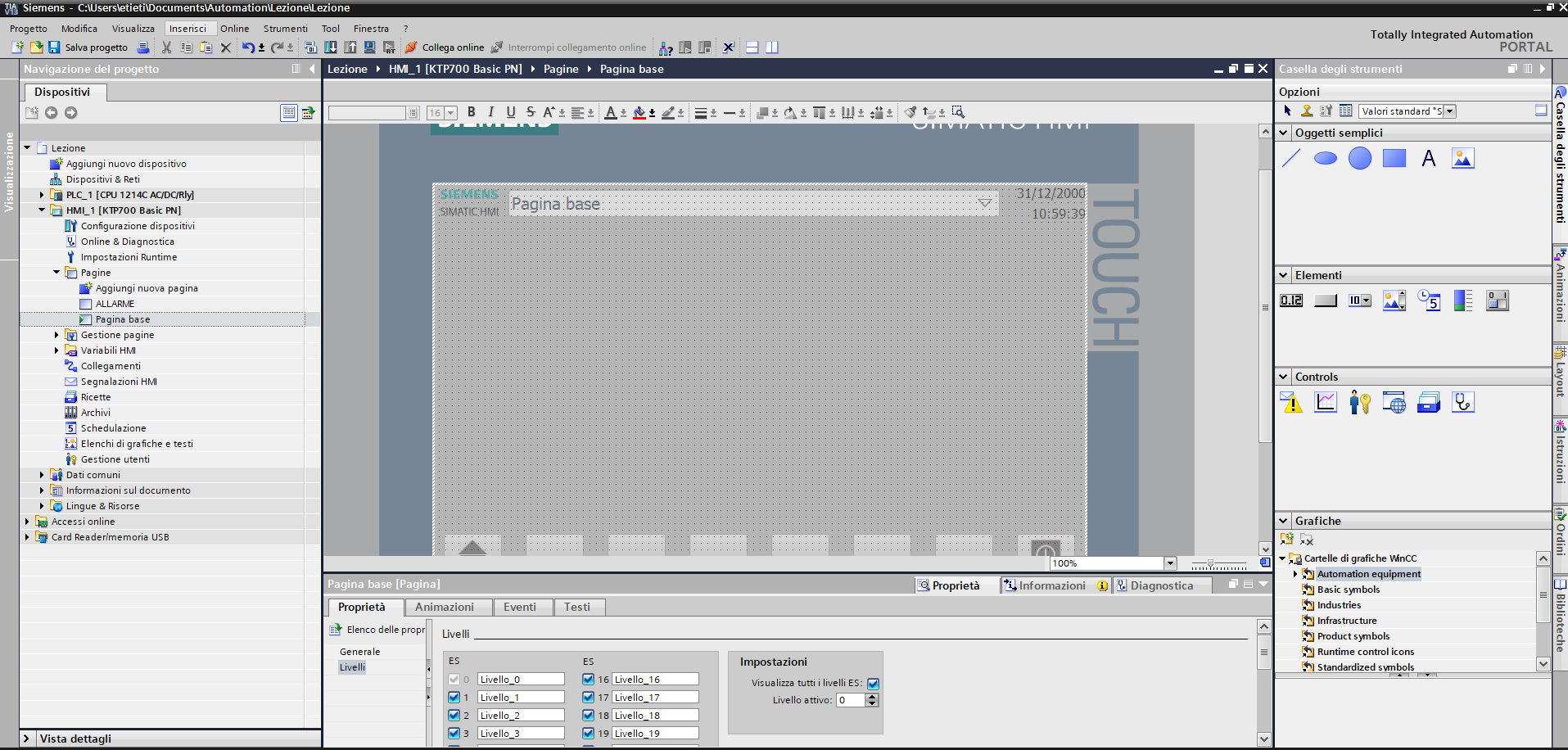


* 1. **Wizard**

Questa è la prima schermata di configurazione, da qui andremo a stabilire la connessione fra pannello e PLC, per far ciò basta cliccare sul pulsante sfoglia e dopo di che selezionare il PLC come da screenshot.

Dopo questo passaggio si può cliccare direttamente sul tasto completa per passare direttamente all’editor oppure si può continuare la configurazione guidata passo per passo, in ogni caso tutte le impostazioni presenti nei menù successivi sono accessibili anche all’interno dell’editor, come ad esempio la possibilità di aggiungere pagine.

****

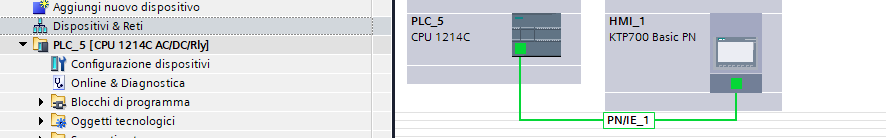
* 1. **Editor HMI**

Questa è la schermata che ci si presenta dopo aver inserito il pannello, come si può notare è molto simile a quella della progettazione del software del PLC.

* A Sinistra [Freccia Rossa] abbiamo il riepilogo di tutti i componenti presenti nel nostro pannello, le varie impostazioni, le pagine, le variabili e altri tipi di dati, come le grafiche.
* Nella parte superiore [Freccia Verde] abbiamo varie impostazioni per modificare la conformazione dei nostri testi e oggetti, da qui potremo allineare degli oggetti oppure renderli tutta della stessa dimensione tramite un semplice click.
* A Destra [Freccia Blu] abbiamo il menù contestuale che cambia in base a cosa stiamo facendo, essendo ora nella progettazione HMI abbiamo tutti gli strumenti per lavorare sul pannello, da oggetti semplici come varie forme e testi a i controlli veri e propri del pannello, come tasti e visualizzatori di parametri.
* Nella parte Inferiore [Freccia Gialla] abbiamo un altro menù contestuale che ci permette di controllare le proprietà e le informazioni di qualsiasi componente, da qui sarà possibile cambiare ogni aspetto della conformazione di un oggetto, le sue proprietà di visualizzazione, come ad esempio varie animazioni e si potranno impostare vari tipi di eventi alla sua attivazione e disattivazione.
  1. **Collegamento variabili PLC**

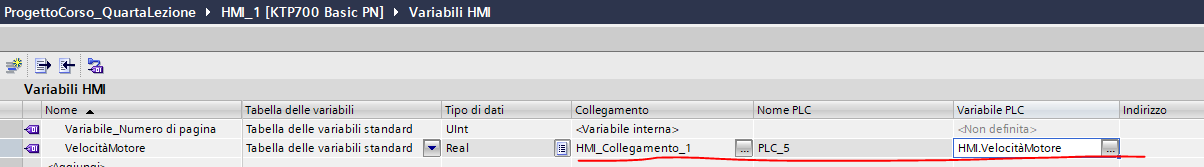
Durante la configurazione del pannello è stata attivata il collegamento con il PLC, se quel passaggio è stato saltato è possibile collegare i due dispositivi anche dopo la configurazione del HMI.

Per far ciò basta andare nel menù “Dispositivi e reti” e collegare le due porte ethernet dei due dispositivi trascinando con il mouse.



Una volta che il collegamento è stato eseguito andare nelle variabili del pannello operatore e creare le variabili da dover utilizzare, dopo di che selezionare il tipo di collegamento, il PLC e selezionare la variabile direttamente dal progetto del controllore.

Come si può notare dallo screenshot la prima variabile è stata dichiarata all’interno del pannello operatore mentre l’altra è si stata creata all’interno del pannello operatore ma fa riferimento a una variabile all’interno di una DB del progetto PLC.

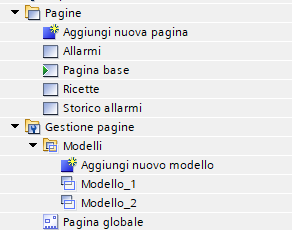
****

Esiste anche un altro metodo per aggiungere le variabili dal progetto PLC al pannello operatore, basterà aprire un qualsiasi contenitore di variabili, ad esempio una DB e trascinare all’interno di una pagina del pannello la variabile, automaticamente verrà creato un oggetto con una variabile interna del pannello collegata alla variabile PLC appena trascinata

* 1. **Pagine e modelli**

È possibile aggiungere quante pagine si vuole tramite l’apposita voce. Ogni pagina sarà basata su un modello e questo modello potrà essere condiviso fra più pagine.

Inizialmente nel progetto è presente un solo modello, quello contenente l’intestazione e la barra principale di navigazione, ma possono esserne creati altri.

****

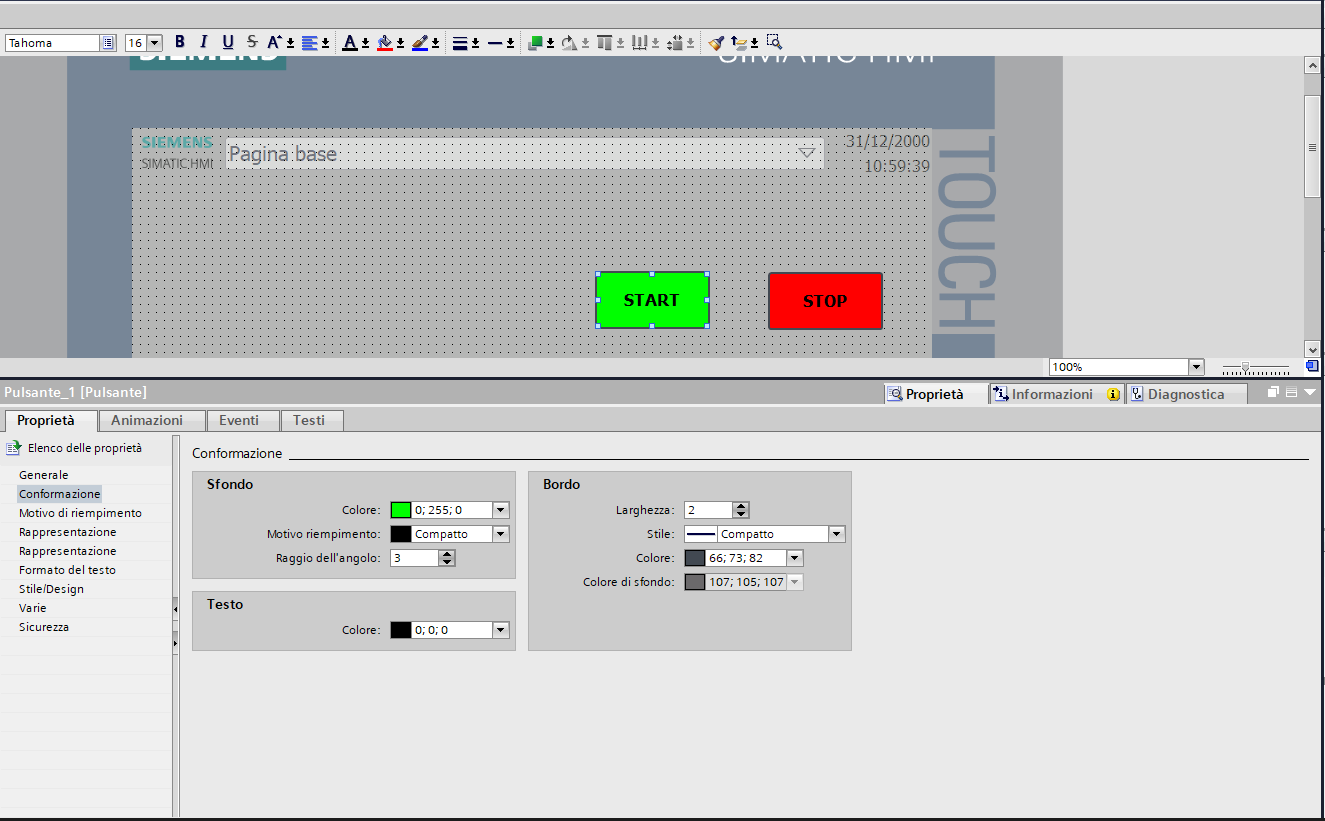
Per controllare e modificare il modello sul quale una pagina è basata basta andare nelle proprietà della relativa pagina.

****

* 1. **Esempio modifiche conformazione oggetti**

Proviamo ad aggiungere un oggetto, ad esempio un tasto, andiamo sul menù contestuale a sinistra e trasciniamo sulla pagina un pulsante che si trova nella categoria “elementi”.

Cliccando su esso ed aprendo il menù proprietà in basso ci troveremo davanti un elenco di proprietà, da qui possiamo andare a modificare ogni aspetto della conformazione del nostro tasto, dalla grandezza al colore del testo, al colore di riempimento e al bordo del tasto stesso.



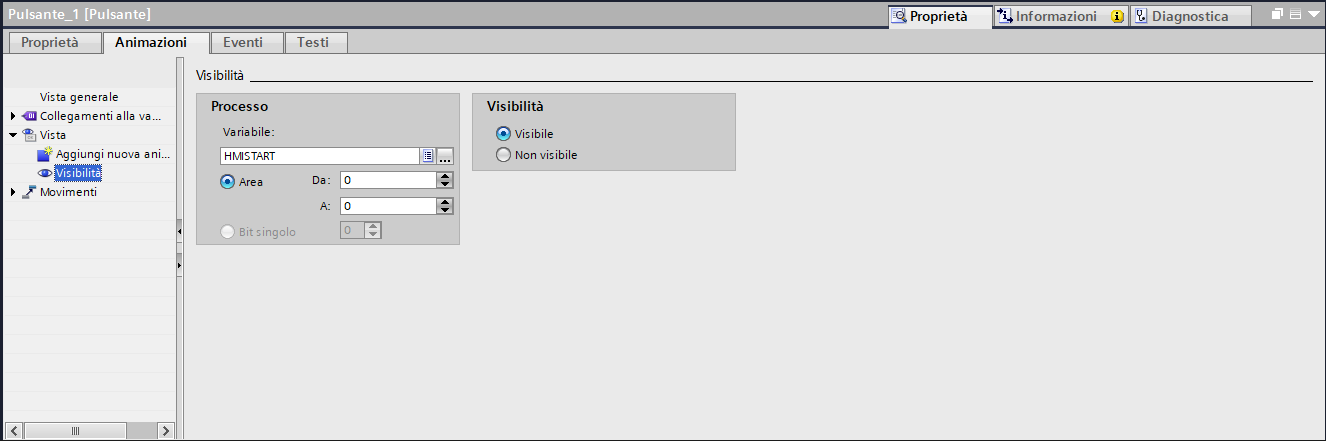
* 1. **Menù animazioni**

Affianco del menu proprietà troviamo in menù animazioni, da qui è possibile andare a impostare varie impostazioni di visualizzazione, cioè quando mostrare o no un qualsiasi oggetto oppure dei movimenti.

Nell’esempio mostrato nello screenshot ho aggiunto una nuova animazione chiamata visibilità all’interno del menù “vista”.

Qui abbiamo due voci, la prima cioè processo, ci permette di legare la variabile che andrà a influenzare la nostra visibilità, in questo caso ho scelto una variabile booleana, quindi gli unici due valori possibili sono 0/1, ma avrei potuto usare qualsiasi altro tipo di variabile come ad esempio una variabile intera.

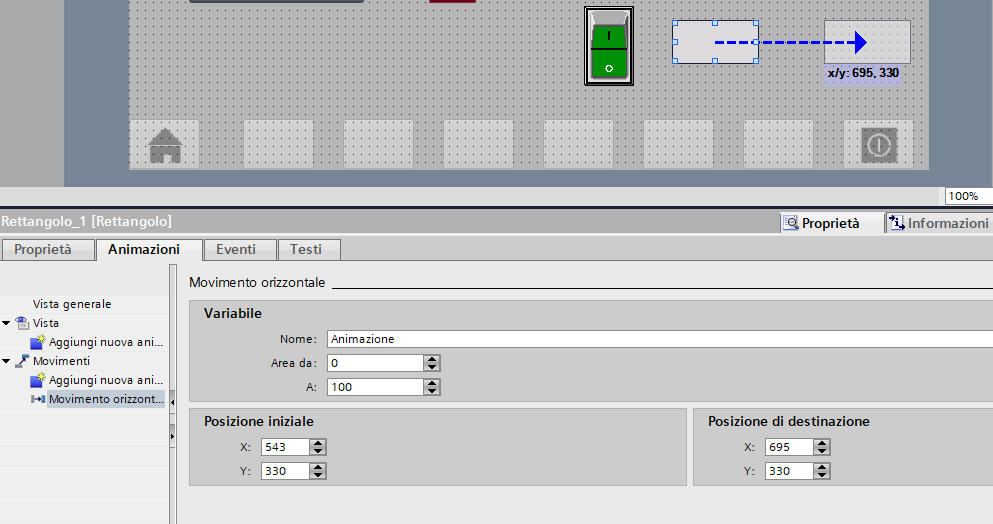
È possibile impostare l’area di lavoro della nostra visibilità dal menù inferiore “area”, nell’esempio ho impostato da 0 a 0, in questo caso quando il bit “HMISTART” sarà a zero il tasto sarà visibile perché come si può notare dello screenshot nella voce affianco “Visibilità” ho selezionato visibile, ovviamente avrei potuto fare anche il contrario selezionando “Non visibile nel caso il bit “HMISTART” fosse a 1.

****

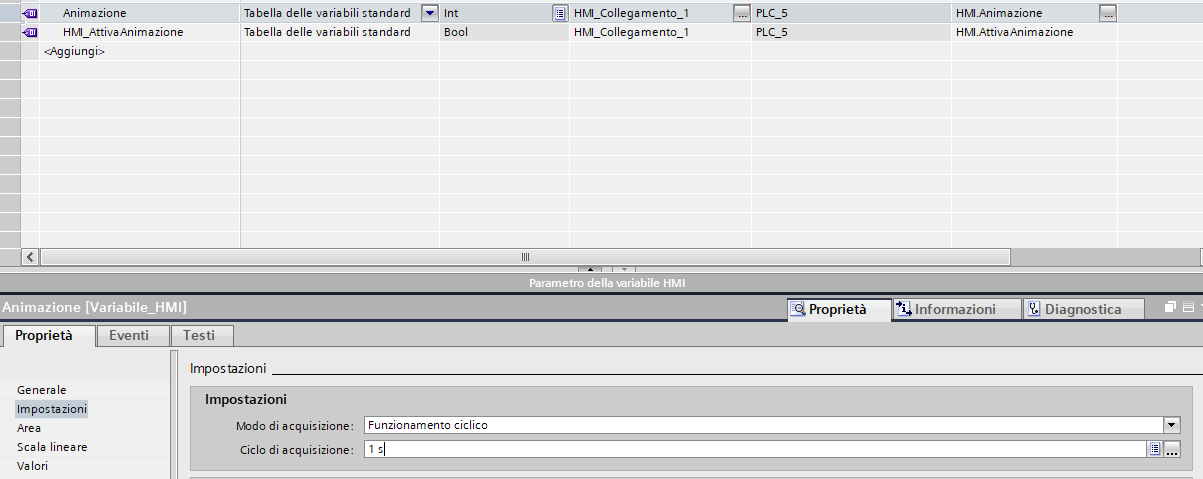
**Esempio animazione**

Aggiungendo un’animazione di tipo movimento è possibile decidere di quanto il nostro oggetto si debba muovere e che tipo di movimento deve eseguire (orizzontale, verticale, ecc).

Il movimento avviene collegato una variabile e selezionando il range che questa variabile deve avere per completare tutto lo spostamento appena prestabilito

****

Una volta configurato il movimento bisogna aumentare la frequenza di aggiornamento della variabile legata ad esso, questo serve per rendere il movimento più fluido. Per far ciò bisogna andare nelle proprietà di quella variabile ed andare in “impostazioni”, qui è possibile selezionare ogni quanto la variabile deve essere aggiornata, il parametro standard è un secondo e il minimo possibile è 100 ms.

****

* 1. **Menù eventi**

da questo menù è possibile ad impostare i vari comandi che il tasto, o qualsiasi oggetto, andrà ad eseguire quando verrà attivato.

Abbiamo varie tipologie di attivazione, le principali sono al “premi” e al “rilascia”.

Nel primo caso appena appoggerò il dito sul tasto verrà eseguito il comando che io avrò imposto, nel secondo caso il comando verrà eseguito quando alzerò il dito da sopra di esso.

(Clic, attiva e disattiva sono comandi praticamente simili a questi due però per conformità di solito non vengono mai utilizzati).

In questo caso io ho aggiunto un comando sia nel momento in cui io premo il pulsante e sia quando lo rilascio, al “premi” ho inserito il comando ImpostaBit, quindi in questo caso Setto il bit a uno, e al “rilascio” ho inserito il comando ResettaBit, perciò subito dopo metterò a zero lo stesso bit, così facendo ho creato un tasto monostabile, cioè che funzionerà solamente quando sarà premuto.

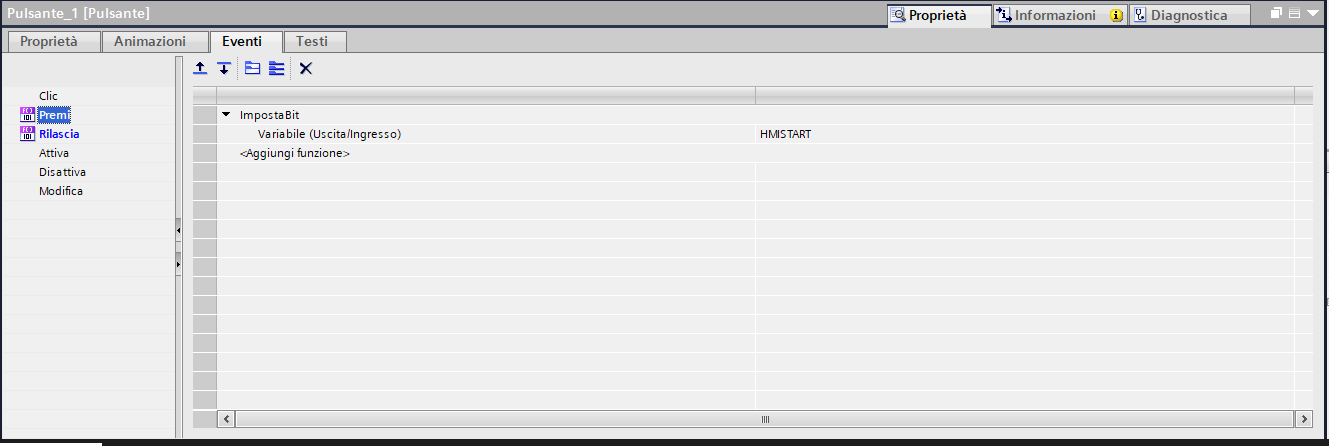
Ci sono altri tipi di comando oltre all’ImpostaBit e al ResettaBit come ad esempio:

InvertiBit, che ci permette di invertire lo stato di un bit, se è a 0 lo metto ad uno e se a 1 lo metto a zero, in questo caso al posto dell’ImpostaBit e il ResettaBit avrei potuto usare sia alla pressione che al rilascio solamente questo comando e avrei ottenuto lo stesso risultato, ovviamente è preferibile usare gli altri due comandi per un fattore di “Trasperenza” del comando.

AttivaPagina, che ci permette di saltare ad un’altra pagina, lo troviamo già di default nei tasti che vengono creati al momento della configurazione del HMI quando andiamo a creare altre pagine.

ImpostaValore, che ci permette di impostare il valore a una variabile intera, Real o Time.

È possibile aggiungere più comandi al singolo evento, ad esempio insieme al impostaBit avrei potuto impostare un valore a una variabile oppure cambiare pagina nel momento in cui avrei premuto sul tasto.

****

* 1. **Campo I/O**

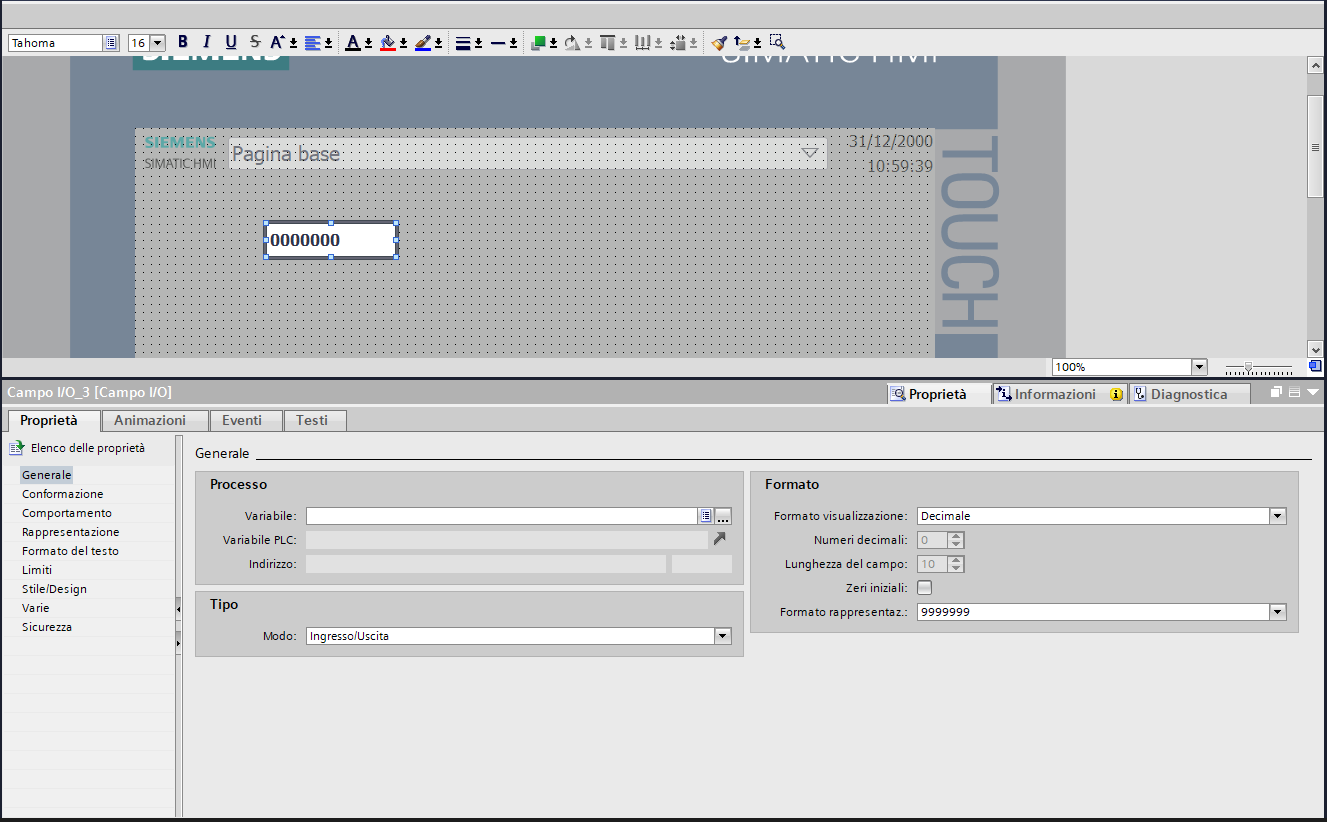
Il campo I/O ci permette di visualizzare e impostare un valore a una variabile.

Tramite il menù proprietà, nella prima voce generale, possiamo andare a collegare la variabile che vogliamo visualizzare/comandare tramite la voce “Processo”, subito sotto troviamo la voce “Tipo, è proprio da qui che possiamo decidere se vogliamo solo visualizzare il valore, selezionando “Uscita”, oppure se vogliamo controllare il valore, selezionando Ingresso/Uscita.

Il Tipo Ingresso non viene mai usato perché ci permette solamente di inserire il valore ma non ci permette di vedere a quanto ammonta in quel momento.

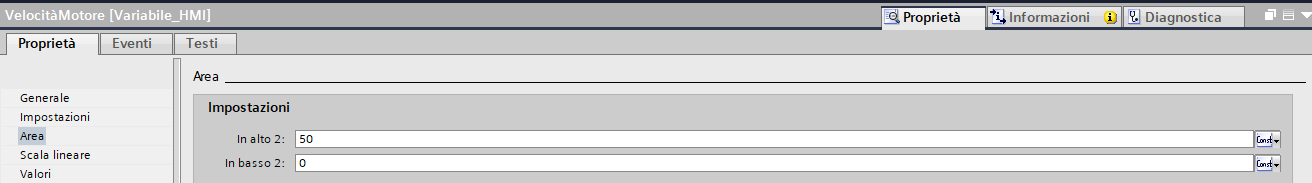
Subito affianco abbiamo la voce “Formato”, da qui possiamo andare a decidere il formato di visualizzazione e quante cifre o caratteri vogliamo che vengano mostrati.

Anche nel Campo I/O come il pulsante è possibile cambiare ogni aspetto grafico ed aggiungere animazioni, nel menù eventi non è possibile aggiungere comandi visto che non si tratto di un oggetto in grado di eseguire comandi**.**



È possibile impostare dei limiti di inserimento per qualsiasi variabile e campo di testo, per far ciò basta andare nelle variabili del pannello operatore, selezionare la variabile che si vuole limitare e andare nelle sue proprietà, nella voce Area è possibile impostare il valore massimo e il valore minimo, in questo caso sono stati impostati due limiti basati su un valore costante, ma è possibile anche impostare due variabili al loro posto.

**NB**: questa opzione limiterà il range di valori che sarà possibile impostare tramite il tastierino del pannello operatore, non andrà a limitare il valore della variabile nel nostro programma PLC, quindi se a PLC la variabile verrà impostata a un valore esterno del nostro range di limitazione esso non verrà corretto.

****

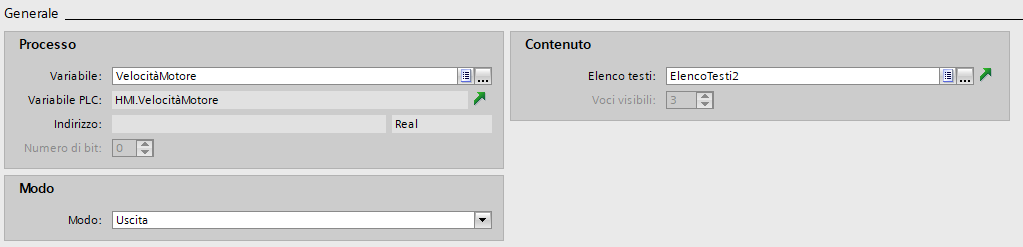
* 1. **Campo I/O Simbolico**

Questo tipo di oggetto a differenza del Campo visto precedentemente non accetta valori tramite un tastierino ma lavora utilizzando una lista di testi prestabilita.

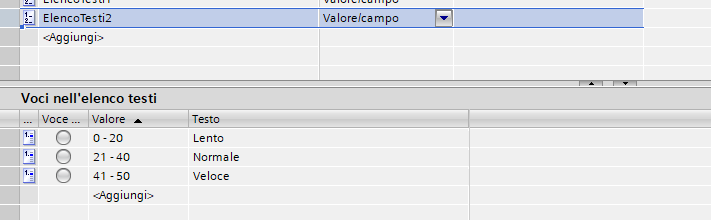
Il primo oggetto in base al testo selezionato imposterà una variabile un determinato valore, il secondo oggetto al contrario in base al valore di una variabile mostrerà un determinato testo.

****

Nelle proprietà di questi oggetti bisogna collegare la variabile e l’elenco testi che si vuole visualizzare, in più si può decidere se il nostro oggetto è un oggetto come il primo raffigurato, cioè un campo di input, oppure un campo di visualizzazione, quindi di uscita.

****

Per creare un elenco di testo andare nella voce “elenchi di testo e grafiche” e creare un nuovo elenco. Nelle proprietà del nuovo elenco è possibile aggiungere i vari testi e vari range di valori per visualizzare i testi appena immessi.

****

* 1. **Grafiche e oggetti Siemens**

È possibile aggiungere anche varie grafiche all’interno del pannello, ad esempio tramite il menù elementi semplici si possono aggiungere vari poligoni di base come delle linee, cerchi o quadrati, modificando la loro conformazione e con po' di fantasia si possono fare molte cose.

Sempre dal menù oggetti semplici è possibile importare

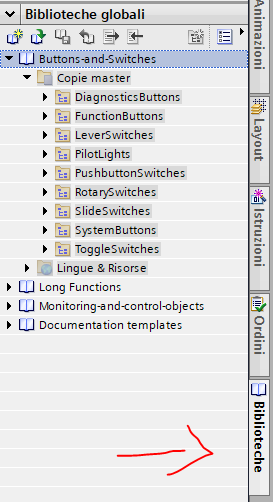
un’immagine dal nostro computer utilizzando una “Vista Grafica”.

Tia Portal ci mette già a disposizione delle grafiche standard, è possibile accedere ad esse tramite il menù “Grafiche”, cliccando su “Cartelle di grafiche WinCC” ci verranno mostrate varie cartelle dove vengono suddivise per categorie tutte le grafiche.

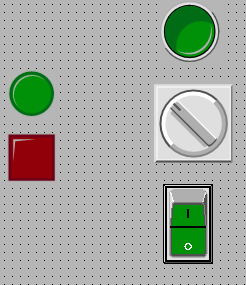
Ad esempio, all’interno della cartella “Automation equipment” troviamo le grafiche di vari componenti del mondo dell’automazione, cliccandoci sopra e andando nella sotto cartella “Motors” troveremo delle grafiche con vari tipi di motori.

Quando apriremo la cartella “Motors” ci troveremo davanti altre 4 cartelle, in ognuna sono presenti lo stesso tipo e numero di grafiche solo che il colore di riempimento sarà differente, avremo le grafiche tutte nere, quelle in bianco e nero e quelle colorate.

Selezioniamo una grafica in bianco e nero e una colorata dello stesso motore e trasciniamole all’interno della pagina per il prossimo paragrafo dove andremo a lavorare con le animazioni di visualizzazione.

Oltre a delle grafiche Siemens ci ha dato a disposizione oggetti precostruiti fra cui pulsanti, selettori e spie.

Questi oggetti si possono trovare all’interno della voce Biblioteche, basterà trascinarli in una pagina come i vari oggetti base e poi configurarli.

****

* 1. **Esempio visualizzazione**

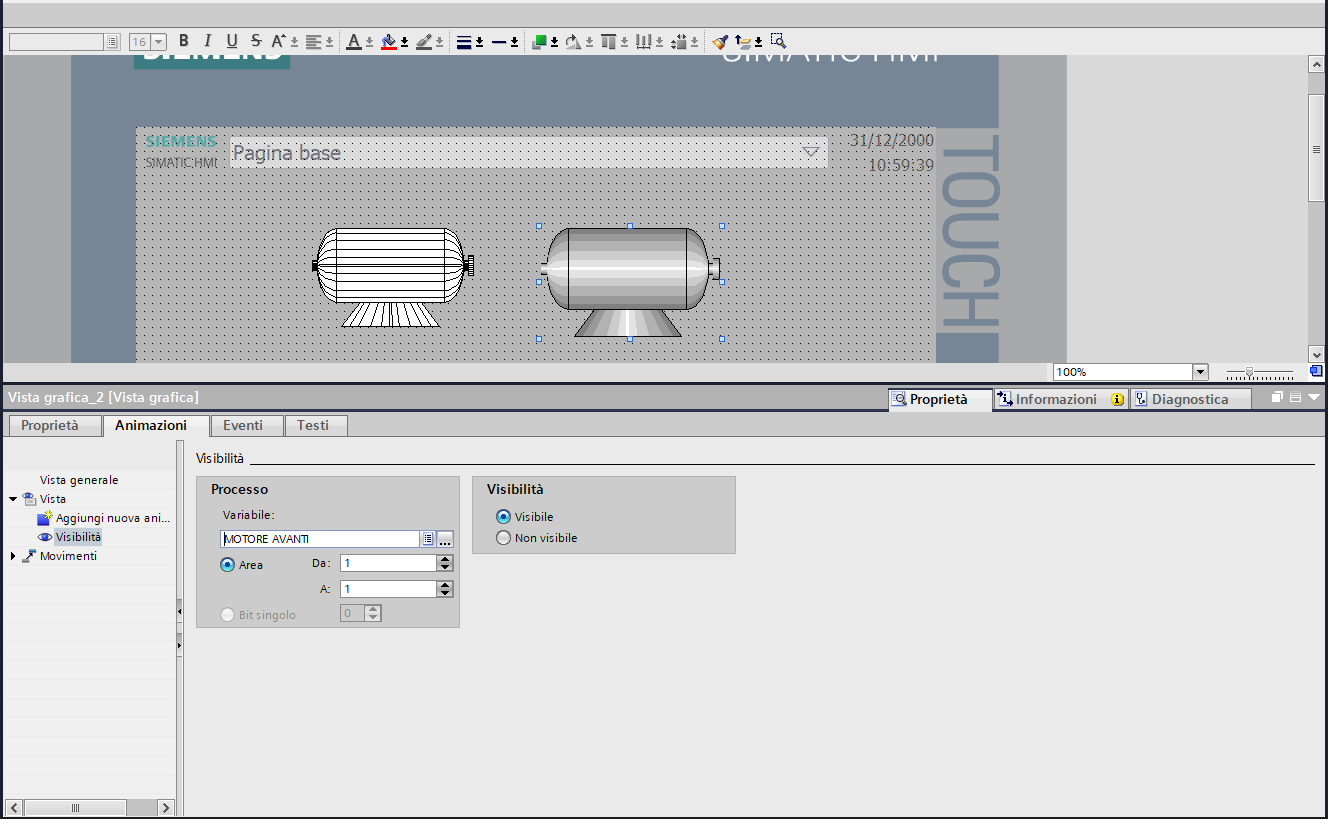
Adesso andremo ad usare le animazioni di visualizzazione per mostrare all’operatore se il motore è in marcia o è fermo, per far ciò andremo a sovrapporre le due grafiche che saranno visualizzate in base allo stato del motore.

Per il momento mettiamo le due grafiche una di fianco all’altra in modo che sia più semplice lavorare su entrambe.

Selezioniamo la grafica colorato e andiamo nel menù “animazioni” ed aggiungiamo una animazione di tipo visibilità come nell’esempio qui sotto.

Voglio che la grafica colorata venga visualizzata se il motore in marcia, quindi andrò a collegare la variabile che mi indica che il motore è in marcia prelevandola dal mio PLC e indico che quando quel Bit è a 1 la grafica deve essere visibile.

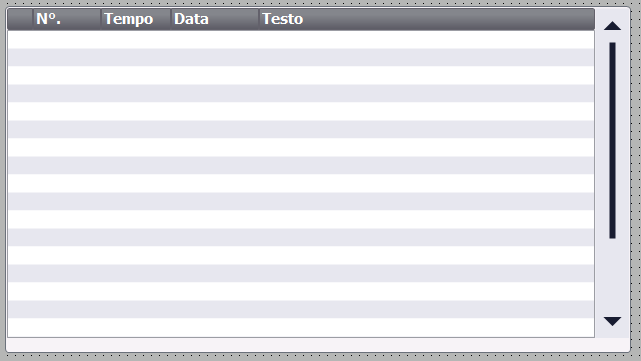
Ripeto lo stesso procedimento per la grafica in bianco e nero mettendo che quando il Bit è a 1 non deve essere visibile, oppure che quando il Bit è a 0 deve essere visibile (Stesso Risultato) e dopo di che le sovrappongo.

In questo modo avrò due grafiche della stessa dimensione e nella stessa posizione che non verranno mai mostrare nello stesso momento ma solo quando una delle dune condizioni è attiva.

* 1. **Allarmi**

Per inserire il visualizzatore degli allarmi andare nella voce “controls” del menù contestuale e trascinare l’oggetto in una pagina del pannello, ci si ritroverà davanti a una tabella.

****

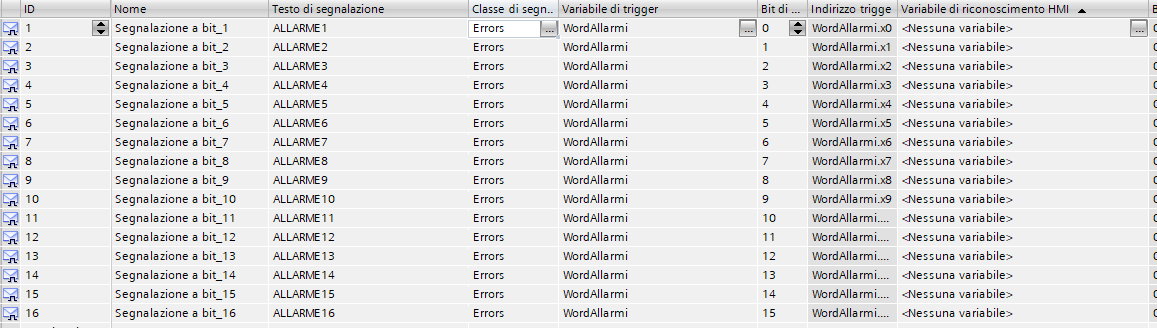
****

Dopo aver inserito la tabella andare nelle sue proprietà nel menù “barra degli strumenti” e attivare il pulsante “Riconosci”, questo pulsante permetterà di eliminare la visualizzazione di un determinato allarme una volta che la sua causa di attivazione è stata risolta.

Dopo di che andare in “Colonne” ed attivare “stato della segnalazione”, in questo modo è possibile visualizzare lo stato di ogni singolo allarme. Ad esempio un allarme può essere solamente attivo oppure può essere riconosciuto ma ancora attivo.

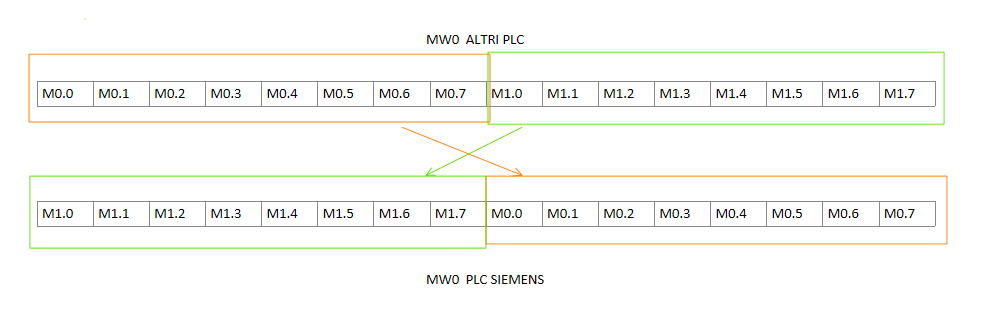
Una volta che l’oggetto per visualizzare i nostri allarmi è stato parametrizzato non ci resta che creare le nostre segnalazioni. Per far ciò bisogna andare nel menù del progetto HMI “Segnalazioni HMI”.

Una volta arrivati in questo menù non ci resta che aggiungere i nostri allarmi. La voce “Testo di segnalazione” contiene il testo che verrà visualizzato nell’oggetto precedentemente aggiungo, la voce “Variabile di trigger” contiene la Word al quale all’interno i vari bit di allarmi sono stati creati, e nella voce successiva “bit di trig” bisogna indicare quale bit della word è associato al relativo allarme.

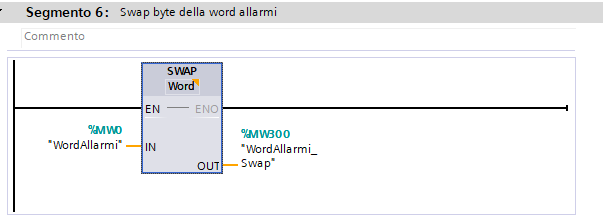
****

NB: nei PLC Siemens i byte delle word sono invertiti, quindi bisogna stare molto attenti nella configurazione degli allarmi.

Ad esempio, il primo bit della word MW0 non è m0.0 ma sarà m1.0.

****

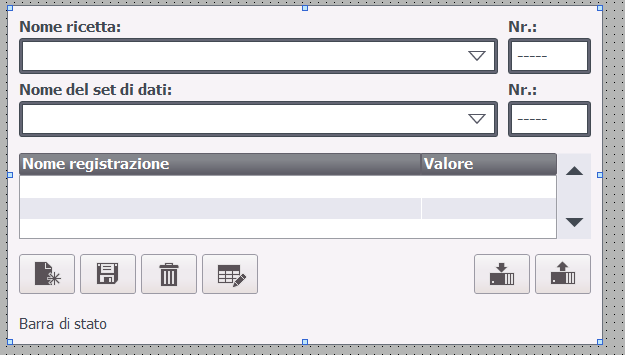
Per correggere questo problema si può utilizzare la funzione Swap che invertirà i byte di una word e li inserirà in un’altra area di memoria, dopo di che collegare alla configurazione degli allarmi la seconda word.

****

* 1. **Gestione ricette**

Nel pannello operatore è possibile gestire più set di valori per vari gruppi di variabili per semplificare la parametrizzazione del nostro macchinario in modo da poter far cambiare molto velocemente il suo comportamento in base alla lavorazione che deve effettuare.

Per far questo Siemens ci mette a disposizione l’oggetto gestione ricette che è reperibile nello stesso menù dove abbiamo trovato l’oggetto per gestire gli allarmi.

****

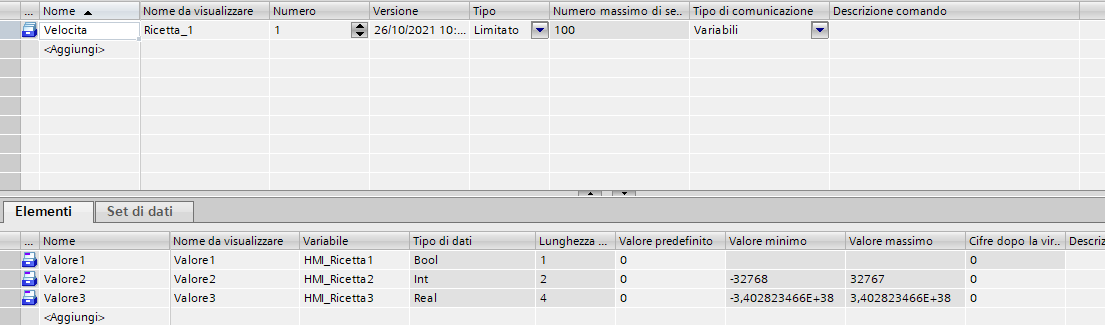
Una volta inserito l’oggetto non ci resta che configurare le nostre ricette, per far ciò andare nel menù del progetto HMI “Ricette”.

Nella prima tabella è possibile aggiungere più ricette, ad esempio nello screenshot qui allegato è stata creata la ricetta “Velocità”.

Ogni ricetta può avere i propri parametri personali, ad esempio qui sono stati aggiunti e collegati tre parametri dal nostro progetto PLC.

Inoltre ogni ricetta può avere più set di dati per ogni valore al suo interno.

Le modifiche possono essere effettuate direttamente dal progetto di TiaPortal, oppure dall’oggetto inserito precedentemente all’interno delle pagine del pannello.

****