



SIEMENS

Cooperates
with Education



SIEMENS

Automation

Documentazione didattica SCE per una soluzione di automazione omogenea Totally Integrated Automation (TIA)

Modulo TIA Portal 010-020

Tipi di blocchi del controllore SIMATIC S7-1200

Trainer Package SCE adatti a questa documentazione

- **SIMATIC S7-1200 AC/DC/RELAIS pacchetti da 6 postazioni PLC "TIA Portal"**
Nr. di ordinazione: 6ES7214-1BE30-4AB3
- **SIMATIC S7-1200 DC/DC/DC pacchetti da 6 postazioni PLC "TIA Portal"**
Nr. di ordinazione: 6ES7214-1AE30-4AB3
- **SIMATIC S7-SW for Training STEP 7 BASIC V11 Upgrade per 6 pacchetti STEP 7 Basic (per S7-1200) "TIA Portal"**
Nr. di ordinazione: 6ES7822-0AA01-4YE0

Tenere presente che questi Trainer Package potrebbero essere sostituiti da successivi pacchetti. Potete consultare i pacchetti SCE attualmente disponibili su: siemens.com/sce/tp

Corsi di formazione

Per corsi di formazione regionali di Siemens SCE contattare il partner di contatto SCE regionale siemens.com/sce/contact

Ulteriori informazioni su SCE

siemens.com/sce

Avvertenza importante sulla traduzione

La presente documentazione didattica è stata tradotta sulla base dei documenti redatti in tedesco. Gli screenshot sono stati riprodotti dalla lingua inglese. Per agevolare la comprensione, anche all'interno del testo sono stati adottati i comandi di programma in inglese con traduzione nella lingua straniera tra parentesi.

Avvertenze per l'impiego

La documentazione di formazione per una soluzione di automazione omogenea Totally Integrated Automation (T I A) è stata creata per il programma "Siemens Automation Cooperates with Education (SCE)" specialmente per scopi di formazione per enti di formazione, di ricerca e di sviluppo pubblici. La Siemens AG declina qualunque responsabilità riguardo ai contenuti di questa documentazione.

Questa documentazione può essere utilizzata solo per la formazione base di prodotti e sistemi Siemens. Ciò significa che può essere copiata in parte, o completamente, e distribuita agli studenti nell'ambito della loro formazione professionale. La riproduzione, distribuzione e divulgazione di questa documentazione è consentita solo all'interno di istituzioni di formazione pubbliche e a scopo di formazione professionale.

Qualsiasi eccezione richiede un'autorizzazione scritta dal partner di riferimento di Siemens AG: Sig. Roland Scheuerer roland.scheuerer@siemens.com.

Le trasgressioni obbligano al risarcimento dei danni. Tutti i diritti sono riservati, incluso anche quelli relativi alla distribuzione e in particolare quelli relativi ai brevetti e ai marchi GM.

L'utilizzo per corsi rivolti a clienti del settore industria è esplicitamente proibito e non è inoltre permesso l'utilizzo commerciale della documentazione.

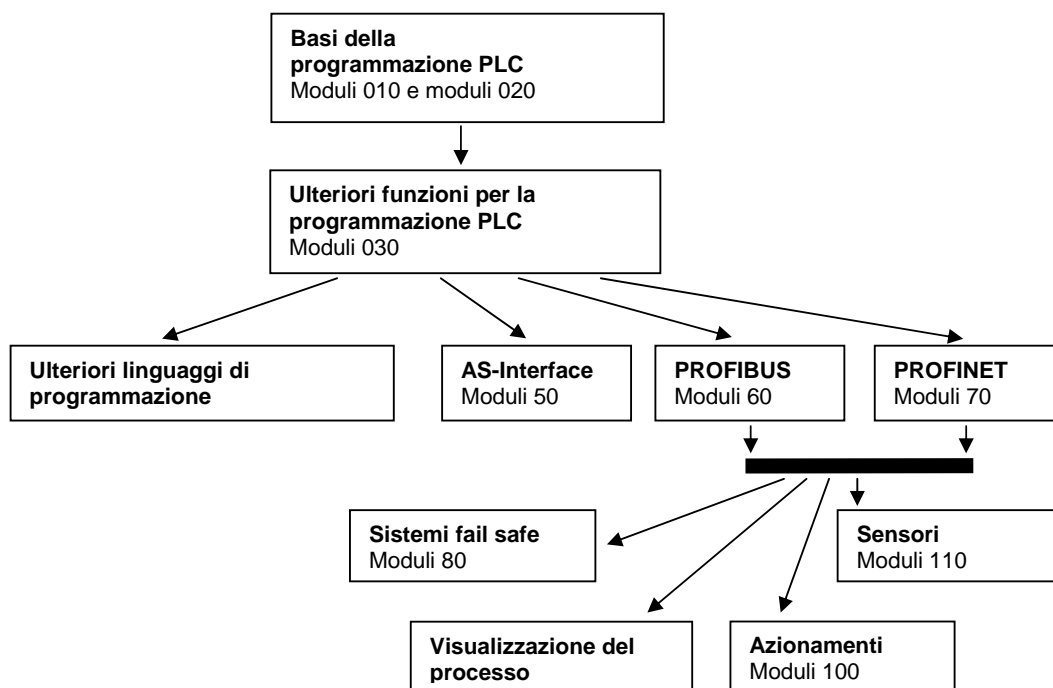
Ringraziamo Michael Dziallas Engineering e tutte le persone coinvolte nella creazione di questa documentazione.

Pagina:

1.	Prefazione	4
2.	Avvertenze sulla programmazione di SIMATIC S7-1200.....	6
2.1	Sistema di automazione SIMATIC S7-1200.....	6
2.2	Software di programmazione STEP 7 Professional V11 (TIA Portal V11)	6
3.	Tipi di blocchi per il controllore SIMATIC S7-1200.....	7
3.1	Programmazione lineare	7
3.2	Programmazione strutturata.....	8
3.2.	Blocchi utente per il controllore SIMATIC S7-1200.....	9
3.2.1	Blocchi organizzativi	10
3.2.2	Funzioni	11
3.2.3	Blocchi funzionali	11
3.2.4	Blocchi dati.....	12
4.	Esempio di programmazione: blocco funzionale per il comando di un nastro trasportatore	13
5.	Programmazione del comando di un nastro trasportatore per il SIMATIC S7-1200	14

1. Prefazione

Il modulo SCE_DE_010-020 appartiene per contenuti all'unità didattica '**Basi della programmazione PLC**' e rappresenta un **rapido approccio** alla programmazione di SIMATIC S7 1200 con il TIA Portal.



Obiettivo didattico:

L'obiettivo di questo modulo è far conoscere al lettore i diversi blocchi per la programmazione del controllore SIMATIC S7-1200 con il tool di programmazione TIA Portal. Il modulo descrive i diversi tipi di blocchi e spiega il procedimento per creare un programma in un blocco funzionale con i passi seguenti:

- Creazione di un blocco funzionale
- Definizione delle variabili interne
- Programmazione con variabili interne nel blocco funzionale
- Richiamo e parametrizzazione del blocco funzionale nell'OB1

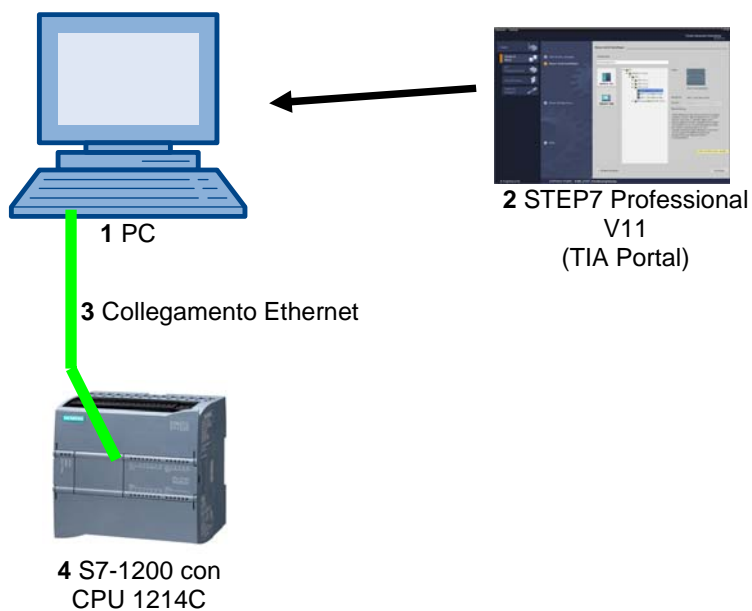
Presupposti:

Per una corretta elaborazione di questo modulo si presuppongono le conoscenze seguenti:

- Esperienza nell'uso di Windows
- Basi della programmazione di PLC con TIA Portal (ad es. Modulo_10-10_R1201 - Avvio alla programmazione con SIMATIC S7-1200 con TIA Portal V11)

Requisiti hardware e software

- 1 PC Pentium 4, 1.7 GHz 1 (XP) – 2 (Vista) GB RAM, memoria su disco rigido ca. 2 GB
Sistema operativo Windows XP Professional SP3 / Windows 7 Professional / Windows 7 Enterprise / Windows 7 Ultimate / Windows 2003 Server R2 / Windows Server 2008 Premium SP1, Business SP1, Ultimate SP1
- 2 Software STEP7 Professional V11 SP1 (Totally Integrated Automation (TIA) Portal V11)
- 3 Collegamento Ethernet tra PC e CPU 315F-2 PN/DP
- 4 PLC SIMATIC S7-1200, ad es. CPU 1214C.
Gli ingressi devono essere condotti su un quadro di comando esterno.



2. Avvertenze sulla programmazione di SIMATIC S7-1200

2.1 Sistema di automazione SIMATIC S7-1200

Il sistema di automazione SIMATIC S7-1200 è un mini controllore modulare per la fascia di potenzialità medio-bassa.

Un'ampia gamma di unità consente di adeguarlo in modo ottimale al compito di automazione specifico. Il controllore S7 è costituito da un alimentatore, da una CPU e da unità di ingressi e uscite per i segnali digitali e analogici.

Eventualmente è possibile aggiungere processori di comunicazione e moduli funzionali per compiti speciali, come ad es. un blocco di comando motore passo-passo.

Il controllore programmabile (PLC) controlla e comanda con il programma S7 una macchina o un processo. Nel programma S7 le unità I/O vengono interrogate attraverso gli indirizzi di ingresso (%I) e indirizzate dagli indirizzi di uscita (%Q).

Il sistema si programma con il software STEP 7.

2.2 Software di programmazione STEP 7 Professional V11 (TIA Portal V11)

Il software STEP 7 Professional V11 (TIA Portal V11) è il tool per la programmazione dei sistemi di automazione

- SIMATIC S7-1200
- SIMATIC S7-300
- SIMATIC S7-400
- SIMATIC WinAC

Con STEP 7 Professional V11 è possibile utilizzare le seguenti funzioni per l'automazione di un impianto:

- Configurazione e parametrizzazione dell'hardware
- Definizione della comunicazione
- Programmazione
- Test, messa in servizio e Service con le funzioni operative e di diagnostica
- Documentazione
- Creazione di visualizzazioni per i SIMATIC Basic Panel con WinCC Basic integrato.
- Con ulteriori pacchetti WinCC è possibile realizzare soluzioni di visualizzazione anche per PC e altri pannelli operatore

Tutte le funzioni sono supportate da una dettagliata Guida in linea.

3. Tipi di blocchi per il controllore SIMATIC S7-1200

Il programma per il controllore SIMATIC S7-1200 viene scritto in cosiddetti blocchi. Per default è già disponibile il blocco organizzativo Main[OB1].

Questo OB rappresenta l'interfaccia con il sistema operativo della CPU, dal quale viene richiamato automaticamente ed elaborato ciclicamente.

Per i compiti di automazione complessi si suddivide il programma in blocchi di programma più piccoli, comprensibili e ordinati per funzioni.

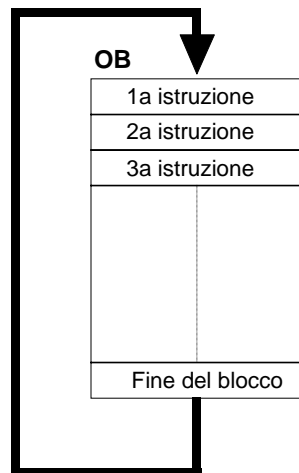
Questi blocchi vengono poi richiamati dai blocchi organizzativi. Alla fine del blocco si torna nuovamente nel blocco organizzativo richiamante, e precisamente nella riga dopo il richiamo.

3.1 Programmazione lineare

Per la programmazione lineare le istruzioni vengono salvate in un blocco ed elaborate nello stesso ordine in cui sono memorizzate nella memoria del programma. Raggiunta la fine del programma (fine del blocco) l'elaborazione del programma ricomincia da capo.

In questo caso si parla di elaborazione ciclica.

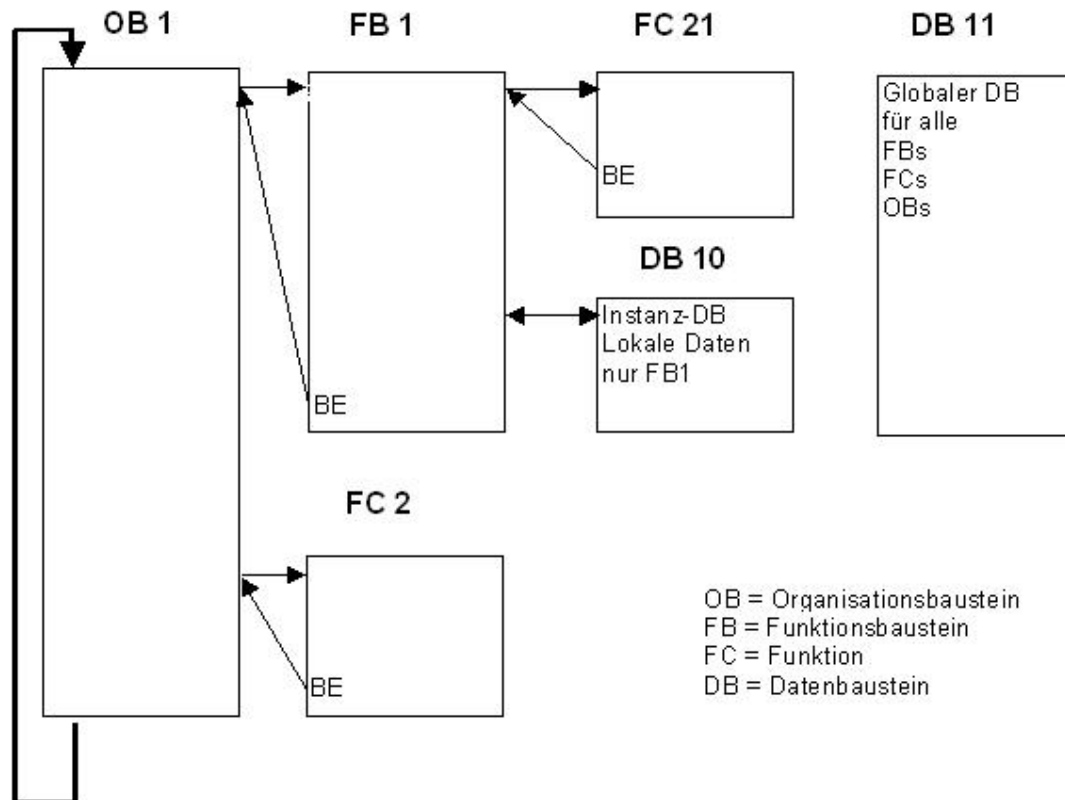
Il tempo impiegato da un dispositivo per elaborare una volta tutte le istruzioni è definito tempo di ciclo. L'elaborazione lineare del programma viene impiegata per lo più per controllori semplici, non troppo complessi, e può essere realizzata in un unico OB.



3.2 Programmazione strutturata

Per i compiti di automazione complessi si suddivide il programma in blocchi di programma più piccoli, comprensibili e ordinati per funzioni. Il vantaggio è quello di poter testare le singole parti del programma e di riunirle in un'unica funzione complessiva se funzionano.

I blocchi di programma devono essere richiamati dal blocco di livello superiore. Quando viene riconosciuta la fine del blocco (BE) il programma continua a essere elaborato nel blocco richiamante dopo il richiamo.



3.2. Blocchi utente per il controllore SIMATIC S7-1200

Per la programmazione strutturata sono disponibili i seguenti blocchi utente:

- OB (blocco organizzativo):

Un OB viene richiamato ciclicamente dal sistema operativo e perciò costituisce l'interfaccia tra programma utente e sistema operativo. In questo OB viene comunicato all'unità di controllo del PLC, attraverso i comandi di richiamo dei blocchi, quali blocchi di programma deve elaborare.

- FB (blocco funzionale):

Un FB deve avere un'area di memoria assegnata per ogni istanza (di richiamo). Quando un FB viene richiamato è possibile assegnargli ad es. un blocco dati (DB) come DB di istanza.

I dati di questo DB di istanza saranno poi accessibili dalle variabili dell'FB.

Se un FB viene richiamato più volte occorre assegnargli diverse aree di memoria.

In un blocco funzionale si possono anche richiamare ulteriori FB e FC.

- FC (funzione):

Una FC non dispone di un'area di memoria assegnata. Una volta elaborata la funzione i suoi dati locali vanno persi.

In una funzione si possono anche richiamare ulteriori FB e FC.

- DB (blocco dati):

I DB vengono utilizzati per creare spazio di memoria per le variabili dei dati. Esistono due tipi di blocchi dati: i DB globali, in cui tutti gli OB, gli FB e le FC possono leggere i dati memorizzati o scrivere essi stessi dati nel DB, e i DB di istanza, che sono assegnati a un determinato FB.

Avvertenza:

Se durante la programmazione delle FC e degli FB sono state utilizzate solo variabili interne sarà possibile utilizzarli più volte in forma di blocchi standard.

Sarà quindi possibile richiamarli un numero qualsiasi di volte. Tuttavia agli FB deve essere assegnata un'area di memoria - una cosiddetta istanza (ad es. un DB) - a ogni richiamo.

3.2.1 Blocchi organizzativi

I blocchi organizzativi (OB) costituiscono l'interfaccia tra il sistema operativo e il programma utente. Vengono richiamati dal sistema operativo e comandano le seguenti operazioni:

Comportamento del sistema di automazione all'avvio

- Elaborazione ciclica del programma
- Elaborazione del programma comandata da un allarme
- Trattamento degli errori

È possibile programmare liberamente i blocchi organizzativi e determinare così il comportamento della CPU.

Esistono diverse possibilità di utilizzare i blocchi organizzativi nel proprio programma:

- **OB di avvio, OB di ciclo, OB di errore temporale e OB di allarme di diagnostica:**

Questi blocchi organizzativi si possono facilmente inserire e programmare nel progetto. Non occorre né parametrizzarli né richiamarli.

- **OB di interrupt di processo e OB di schedulazione orologio:**

Questi blocchi organizzativi devono ancora essere parametrizzati dopo essere stati inseriti nel programma. Inoltre gli OB di interrupt di processo si possono assegnare e nuovamente separare da un evento in runtime utilizzando rispettivamente le istruzioni ATTACH e DETACH.

- **OB di allarme di ritardo:**

L'OB di allarme di ritardo si può facilmente inserire e programmare nel progetto. Deve inoltre essere richiamato nel programma utente mediante l'istruzione SRT_DINT. Non è necessario parametrizzarlo.

Informazione di avvio

All'avvio di alcuni blocchi organizzativi il sistema operativo emette informazioni che possono essere valutate nel programma utente.

Queste informazioni possono essere molto utili in particolare per la diagnostica degli errori.

Per sapere se e quali informazioni vengono emesse è possibile leggere le descrizioni dei blocchi organizzativi.

3.2.2 Funzioni

Una funzione contiene un programma che viene eseguito quando la funzione viene richiamata da un altro blocco di codice.

Le funzioni (FC) sono blocchi di codice senza memoria. Dopo l'elaborazione della funzione i dati delle variabili temporanee vanno persi. Per il salvataggio permanente dei dati delle funzioni si possono utilizzare blocchi dati globali.

Le funzioni possono essere impiegate ad es. per i seguenti scopi:

- Restituzione dei valori della funzione al blocco richiamante, ad es. nel caso delle funzioni matematiche
- Esecuzione di funzioni tecnologiche, ad es. controlli singoli con operazioni binarie

Una funzione può essere richiamata anche più volte in diversi punti all'interno di un programma. Questa possibilità semplifica la programmazione di funzioni complesse che ricorrono spesso.

3.2.3 Blocchi funzionali

I blocchi funzionali contengono sottoprogrammi che vengono eseguiti ogni volta che un blocco funzionale viene richiamato da un altro blocco di codice.

I blocchi funzionali sono blocchi di codice che memorizzano i propri valori in modo permanente in blocchi dati di istanza in modo che restino a disposizione anche dopo l'elaborazione del blocco. I loro parametri di ingresso, uscita e transito vengono memorizzati in modo permanente nei blocchi dati di istanza in modo che restino disponibili anche dopo l'elaborazione del blocco. Per questo motivo vengono definiti anche blocchi con "memoria".

I blocchi funzionali vengono utilizzati per quei compiti che non si possono realizzare con le funzioni:

- ogni volta che nei blocchi sono necessari temporizzatori e contatori (vedere il modulo M3)
- ogni volta che un'informazione deve essere salvata nel programma.
Un esempio è la preselezione del modo di funzionamento con un tasto.

Un blocco funzionale può essere richiamato anche più volte in diversi punti all'interno di un programma. Questa possibilità semplifica la programmazione di funzioni complesse che ricorrono spesso.

Istanze dei blocchi funzionali

Il richiamo di un blocco funzionale viene definito istanza.

A ogni istanza di un blocco funzionale viene assegnata un'area di memoria che contiene i dati utilizzati dal blocco funzionale. Questa memoria viene messa a disposizione da blocchi dati creati automaticamente dal software. È anche possibile rendere disponibile la memoria per diverse istanze in un blocco dati come **multiistanza**.

3.2.4 Blocchi dati

Diversamente dai blocchi di codice i blocchi dati non contengono istruzioni ma fungono da memoria per i dati utente.

I blocchi dati contengono quindi dati variabili che vengono utilizzati dal programma utente.

I **blocchi dati globali** contengono dati che possono essere utilizzati da tutti gli altri blocchi.

Le dimensioni max. variano in funzione della CPU. La struttura dei blocchi dati globali si può definire liberamente.

Esempi di applicazione:

- Salvataggio di informazioni relative a un sistema di gestione magazzino: "dove si trova il tale prodotto".
- Salvataggio di ricette per determinati prodotti.

Ogni blocco funzionale, ogni funzione e ogni blocco organizzativo può leggere o scrivere dati in un blocco dati globale. Questi dati vengono mantenuti nel blocco dati anche quando si esce dal blocco.

Il richiamo di un blocco funzionale viene definito istanza. A ogni richiamo di un blocco funzionale con assegnazione di parametri viene assegnato un **blocco dati di istanza** che funge da memoria dei dati. Qui vengono memorizzati i parametri attuali e i dati statici del blocco funzionale.

Le dimensioni max. dei blocchi dati di istanza varia in funzione della CPU. Le variabili dichiarate nel blocco funzionale determinano la struttura del blocco dati di istanza.

Un blocco dati globale e un blocco dati di istanza possono essere aperti contemporaneamente.

4. Esempio di programmazione: blocco funzionale per il comando di un nastro trasportatore

Per creare dei blocchi che devono fungere per così dire da "scatola nera" in qualsiasi programma è necessario programmarli utilizzando delle variabili. La regola vuole che in questi blocchi non siano mai utilizzati ingressi/uscite, merker ecc. indirizzati in modo assoluto. All'interno del blocco si possono utilizzare solo variabili e costanti.

Nell'esempio seguente verrà creato un blocco funzionale con dichiarazione delle variabili che contiene il comando di un nastro trasportatore in funzione del modo di funzionamento.

Con il tasto 'S1' si deve selezionare il modo di funzionamento 'manuale' e con il tasto 'S2' il funzionamento 'automatico'.

Nel modo di funzionamento 'manuale' il motore è acceso finché è azionato il tasto 'S3' ma il tasto 'S4' non deve essere assolutamente azionato.

Nel modo di funzionamento 'automatico' deve essere possibile accendere il motore del nastro con il tasto 'S3' e spegnerlo con il tasto 'S4' (contatto normalmente chiuso).

Lista di attribuzione:

Indirizzo	Simbolo	Commento
%I 0.0	S1	Tasto per modo di funzionamento manuale S1 NO
%I 0.1	S2	Tasto per modo di funzionamento automatico S2 NO
%I 0.2	S3	Tasto ON S3 NO
%I 0.3	S4	Tasto OFF S4 NC
%Q 0.2	M1	Motore del nastro M1

Avvertenza:

Il tasto OFF S4 qui è un contatto normalmente chiuso per garantire la protezione dalla rottura conduttore. Ciò significa che in caso di rottura conduttore in questo tasto l'impianto si arresta automaticamente. Diversamente non si potrebbe arrestare l'impianto in caso di rottura conduttore. Nella tecnica di comando perciò tutti i tasti di arresto/OFF e tutti gli interruttori devono sempre essere contatti normalmente chiusi.

5. Programmazione del comando di un nastro trasportatore per il SIMATIC S7-1200

Per la gestione del progetto e la programmazione si utilizza il software '**Totally Integrated Automation Portal**'.

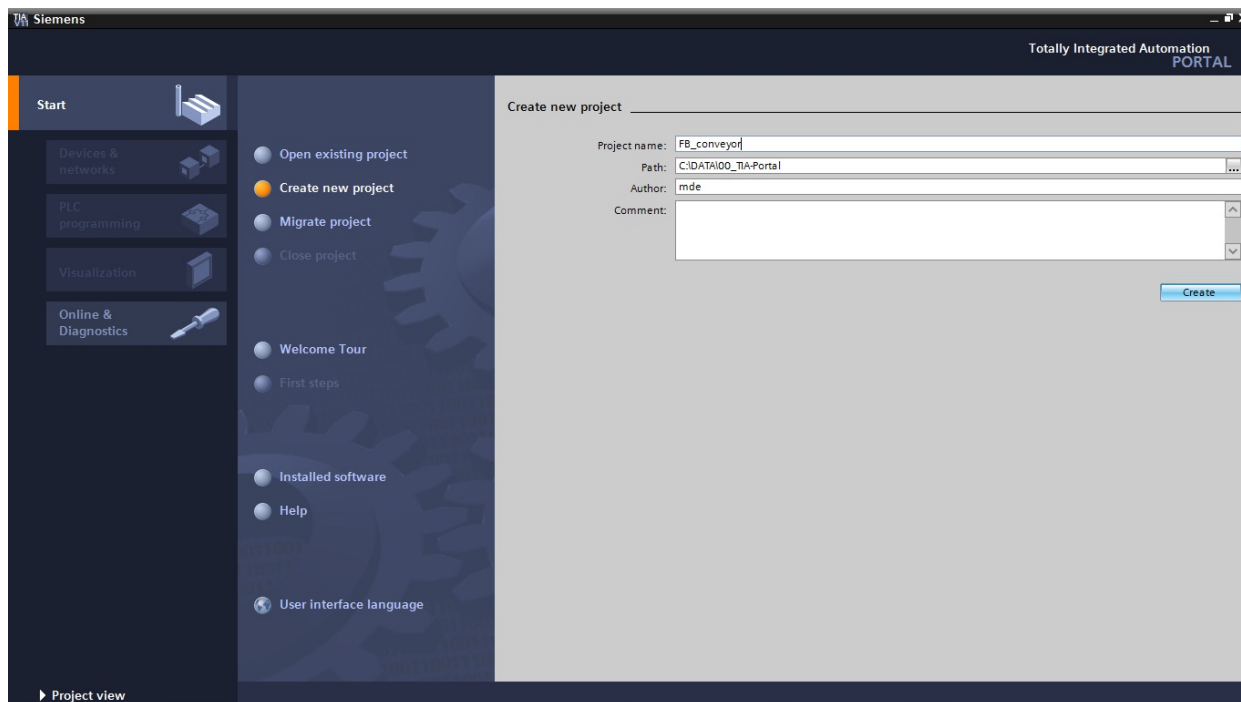
Qui si creano, si parametrizzano e si programmano con un'interfaccia utente unificata i componenti come controllore, visualizzazione e collegamento in rete della soluzione di automazione. Per la diagnostica degli errori sono disponibili diversi tool online.

Con i passi seguenti è possibile creare un progetto per il SIMATIC S7-1200 e programmare la soluzione del compito che ci siamo proposti:

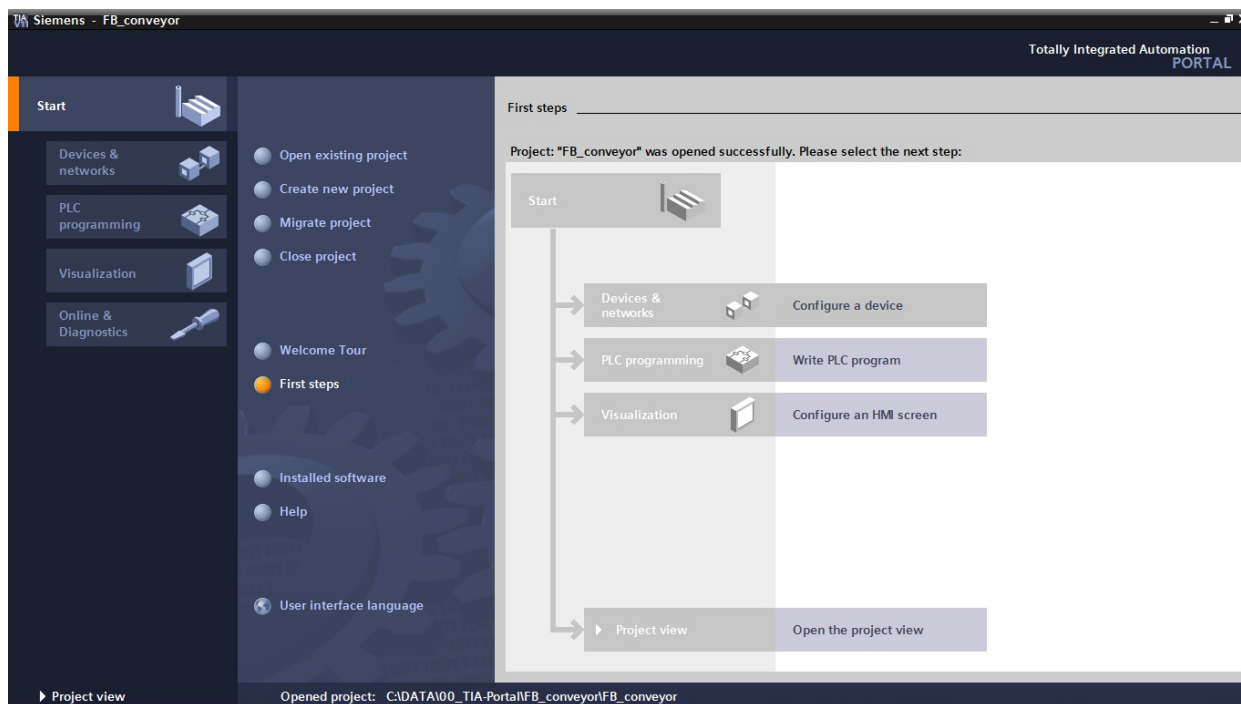
1. Il tool principale è '**Totally Integrated Automation Portal**', che si richiama qui con un doppio clic. (→ Totally Integrated Automation Portal V11)



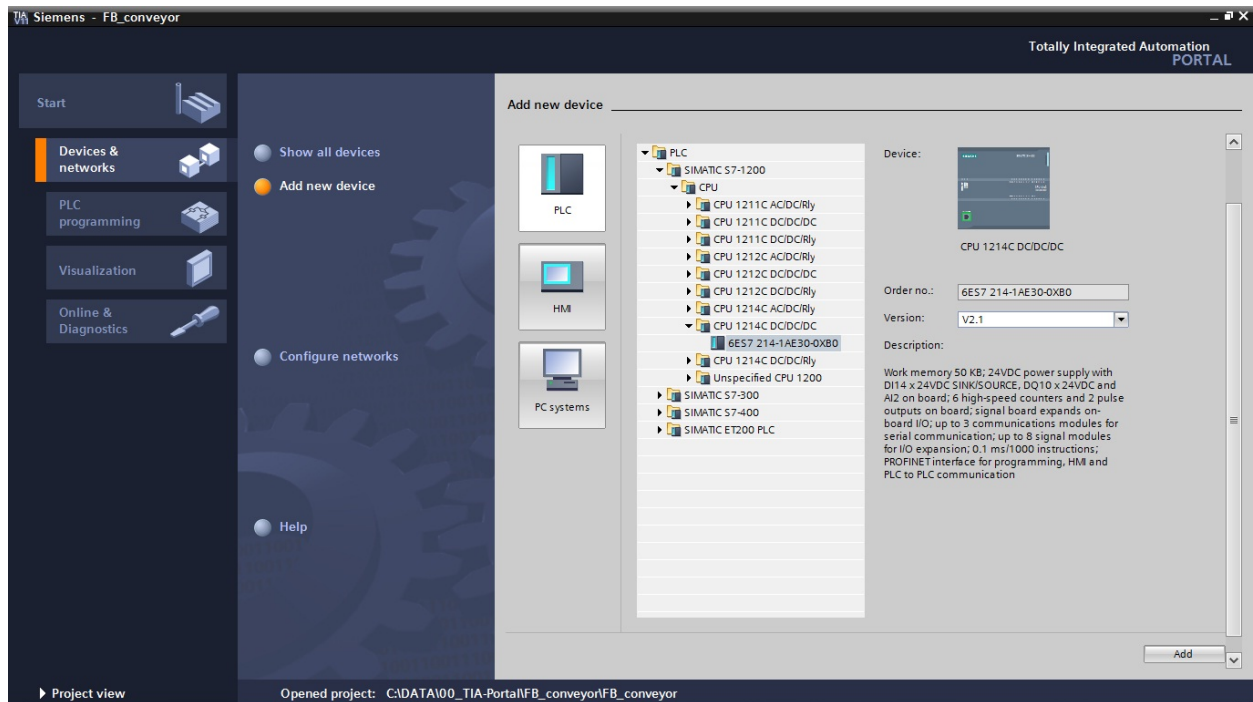
2. I programmi per il SIMATIC S7-1200 vengono gestiti in progetti. Uno di questi progetti verrà ora creato nella vista portale (→ Create new project (Crea nuovo progetto) → FB_conveyor (FB_nastro) → Create (Crea))



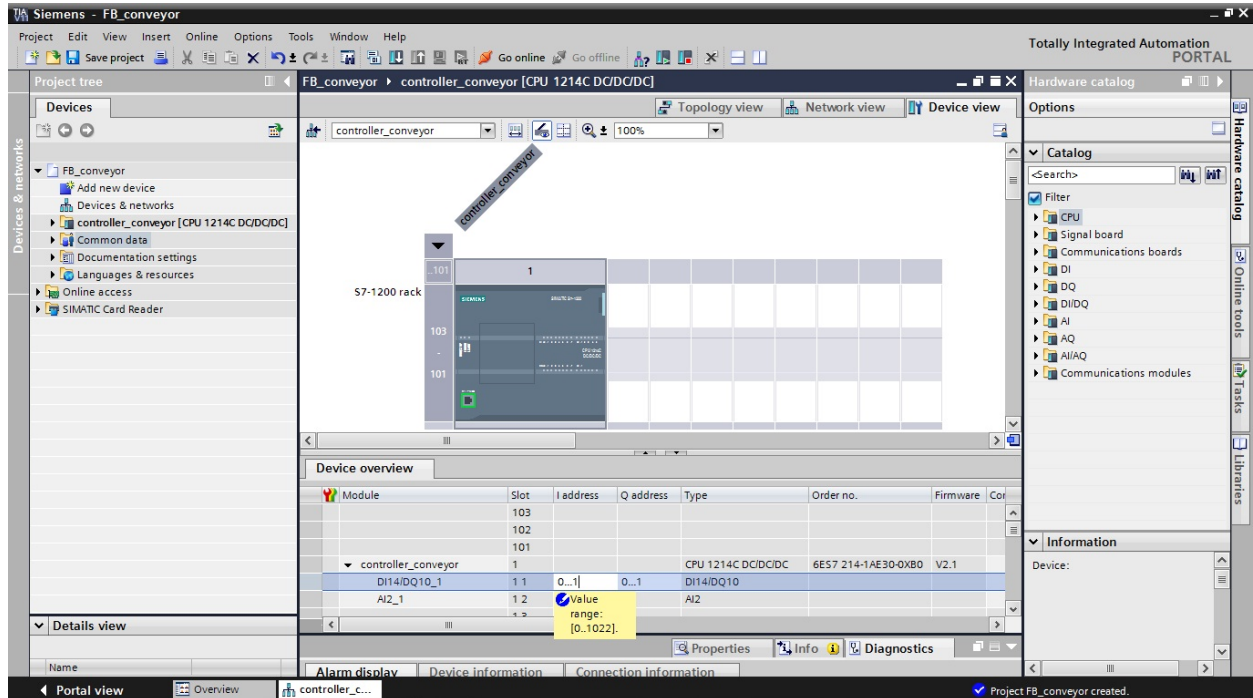
3. Ora alla voce **'First steps'** vengono proposti i primi passi per la progettazione. Innanzitutto vogliamo creare un dispositivo con **'Configure a device'**. (→ Primi passi → Configura un dispositivo)



4. Successivamente inseriremo con **'Add new device'** un nuovo dispositivo denominato **'controller_conveyor'**. Selezionare dal catalogo la **'CPU1214C'** con il numero di ordinazione adeguato. (→ Aggiungi nuovo dispositivo → Comando_nastro → CPU1214C → 6ES7 → Aggiungi)

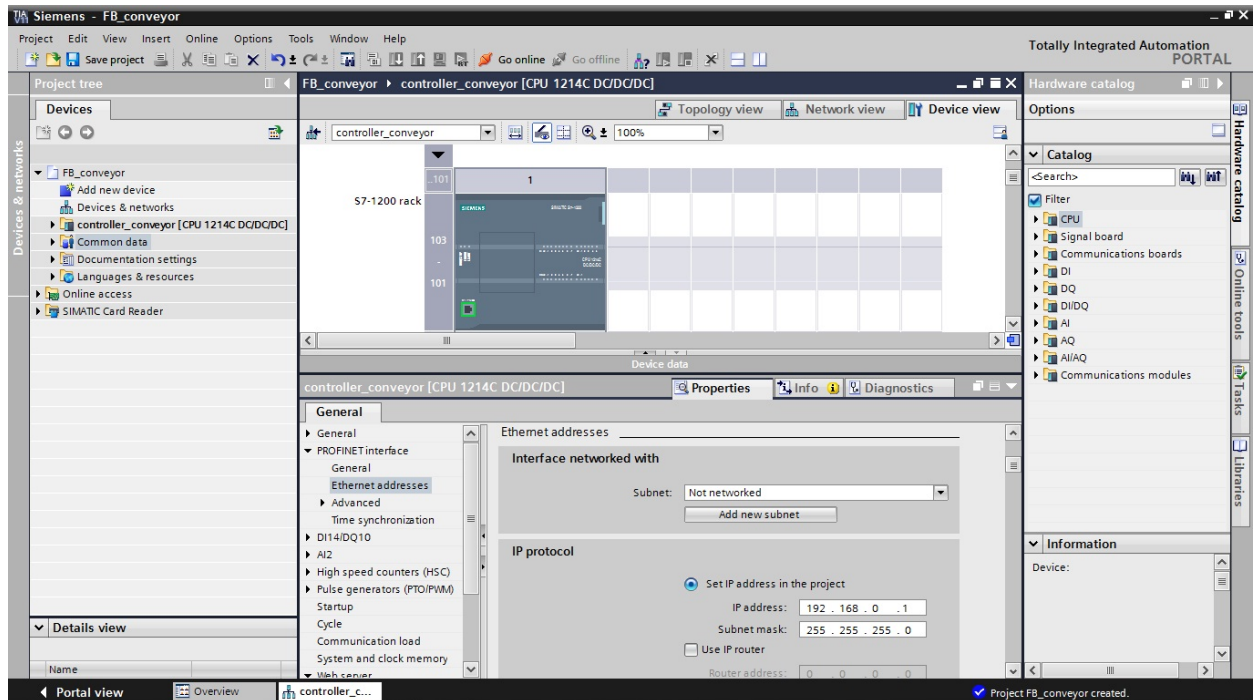


5. Ora il software passa automaticamente alla vista progetto con la configurazione hardware aperta. Qui è possibile aggiungere ulteriori moduli prelevandoli dal catalogo hardware (a destra!) e impostare gli indirizzi degli ingressi e delle uscite nella **'Device overview'**. Qui gli ingressi integrati della CPU hanno gli indirizzi %I0.0 - %I1.5 e le uscite integrate gli indirizzi %Q0.0 - %Q1.1 (→ Vista generale dispositivi → DI14/DO10 → 0...1)



6. Per far sì che in un secondo tempo il software acceda alla CPU corretta è necessario impostare l'indirizzo IP e la maschera di sottorete.

(→ Properties (Proprietà) → General (Generale) → PROFINET interface (Interfaccia PROFINET) → IP address (Indirizzo IP): 192.168.0.1 → Subnet mask (Maschera di sottorete): 255.255.255.0)



7. Poiché nella moderna programmazione non si utilizzano gli indirizzi assoluti ma delle variabili, qui è necessario definire le **variabili PLC globali**.

Queste variabili PLC globali sono nomi descrittivi corredati di commento per gli ingressi e le uscite che vengono utilizzati nel programma. In un secondo momento, durante la programmazione, da questi nomi si potrà accedere alle variabili PLC globali.

Queste variabili globali si possono utilizzare in tutto il programma in tutti i blocchi.

Nella navigazione del progetto selezionare prima '**controller_conveyor[CPU1214C DC/DC/DC]**' e poi '**PLC tags**'. Aprire la tabella '**PLC tags**' con un doppio clic e inserire i nomi degli ingressi e delle uscite come nella figura seguente. (→ Comando_nastro[CPU1214C DC/DC/DC]' → Variabili PLC → Variabili PLC)

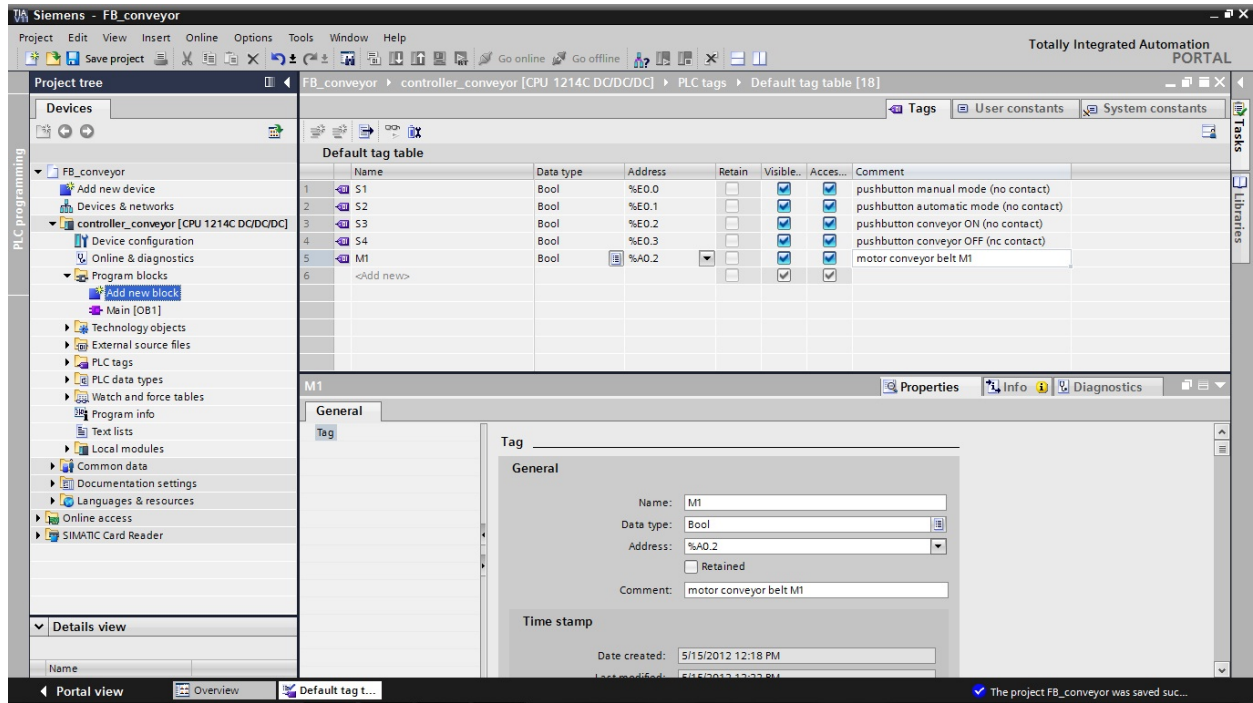
The screenshot displays the Siemens TIA Portal interface for configuring PLC tags. The 'Default tag table' is shown with the following data:

Name	Data type	Address	Retain	Visible	Access...	Comment
1 S1	Bool	%E0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	pushbutton manual mode (no contact)
2 S2	Bool	%E0.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	pushbutton automatic mode (no contact)
3 S3	Bool	%E0.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	pushbutton conveyor ON (no contact)
4 S4	Bool	%E0.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	pushbutton conveyor OFF (nc contact)
5 M1	Bool	%A0.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	motor conveyor belt M1
6 <Add new>						

The 'Properties' dialog box for tag M1 shows the following configuration:

- Name: M1
- Data type: Bool
- Address: %A0.2
- Retained:
- Comment: motor conveyor belt
- Date created: 5/15/2012 12:18 PM

8. Per creare il blocco funzionale FB1 selezionare nella navigazione del progetto 'controller_conveyor[CPU1214C DC/DC/DC]' e quindi 'Program blocks'. Fare doppio clic su 'Add new block'. (→ Comando_nastro[CPU1214C DC/DC/DC] → Blocchi di programma → Inserisci nuovo blocco)



9. Selezionare **'Function block (FB)'** e assegnare il nome **'conveyor'**. Come linguaggio di programmazione è preimpostato lo schema funzionale **'FBD'**. La numerazione è automatica. Poiché questo FB1 in un secondo tempo verrà richiamato comunque con il nome simbolico, il numero non ha più una grande importanza. Applicare i dati inseriti con **'OK'**. (→ Blocco funzionale (FB1) → Nastro → FUP → OK)

Add new block

Name: conveyor

Language: FBD

Number: 1

Manual
 Automatic

Block access: Optimized
 Standard - compatible with S7-300/400

Description: Function blocks are code blocks that store their values permanently in instance data blocks, so that they remain available after the block has been executed.

[More...](#)

► **Additional information**

Add new and open

OK Cancel

10. Il blocco '**conveyor[FB1]**' si aprirà automaticamente. Prima di poter scrivere il programma è necessario dichiarare l'interfaccia del blocco.

Con la dichiarazione dell'interfaccia si definiscono le variabili locali, conosciute solo in questo blocco.

Le variabili si suddividono in due gruppi:

- Parametri del blocco che costituiscono l'interfaccia del blocco per il richiamo nel programma.

Tipo	Denominazione	Funzione	Disponibile in
Parametri di ingresso	Input	Parametri i cui valori vengono letti dal blocco.	Funzioni, blocchi funzionali e alcuni tipi di blocchi organizzativi
Parametri di uscita	Output	Parametri i cui valori vengono scritti dal blocco.	Funzioni e blocchi funzionali
Parametri di transito	InOut	Parametri il cui valore viene letto dal blocco al momento del richiamo e riscritto nello stesso parametro dopo l'elaborazione.	Funzioni e blocchi funzionali

- Dati locali che permettono il salvataggio di risultati intermedi.

Tipo	Denominazione	Funzione	Disponibile in
Dati locali temporanei	Temp	Variabili che permettono il salvataggio di risultati intermedi temporanei. I dati temporanei vengono mantenuti solo per un ciclo.	Funzioni, blocchi funzionali e blocchi organizzativi
Dati locali statici	Static	Variabili che permettono il salvataggio di risultati intermedi statici nel blocco dati di istanza. I dati statici vengono mantenuti anche per più cicli finché non vengono riscritti.	Blocchi funzionali

11. Per la dichiarazione delle variabili locali nel nostro esempio sono necessarie le seguenti variabili.

Input:

manual	Qui viene immesso il segnale per la selezione del modo di funzionamento manuale
automatic	Qui viene immesso il segnale per la selezione del modo di funzionamento automatico
on	Qui viene immesso il segnale di avvio
off	Qui viene immesso il segnale di arresto

Output:

motor	Qui viene scritto un segnale di uscita per il motore del nastro
-------	---

Static (è presente solo nei blocchi funzionali FB):

mem_automatic	Qui viene salvato il modo di funzionamento preselezionato
mem_motor	Qui viene memorizzata l'informazione se il motore è stato avviato in funzionamento automatico

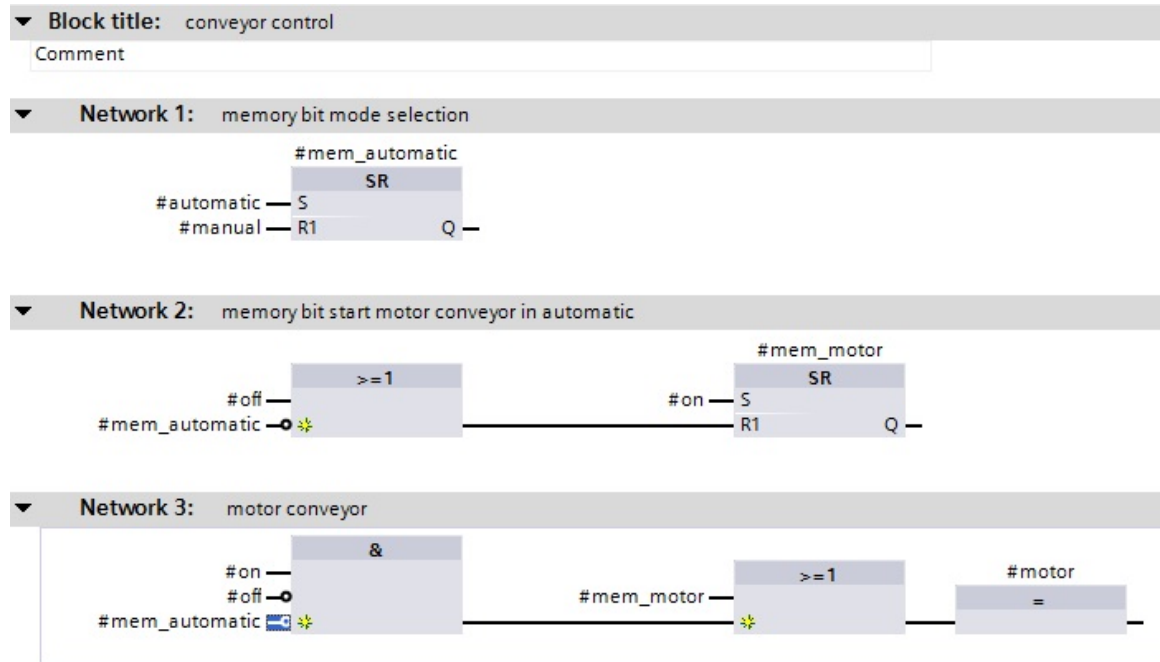
In questo caso tutte le variabili sono del tipo 'Bool', ovvero variabili binarie che possono avere solo lo stato '0' (false) o '1' (true).

In questo esempio è importante notare che lo stato nelle due variabili 'mem_automatic' e 'mem_motor' deve essere memorizzato per un periodo prolungato. Perciò qui è necessario utilizzare il tipo di variabile '**Static**'. Questo tipo di variabile d'altra parte esiste solo in un blocco funzionale FB.

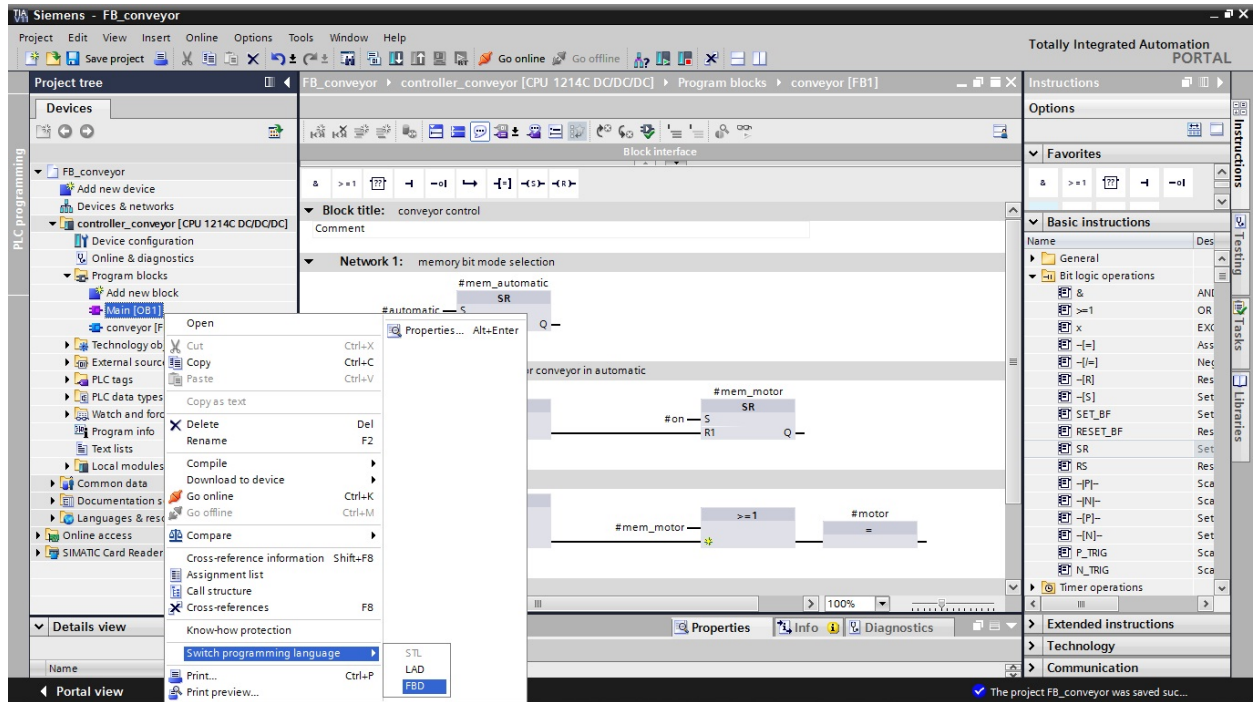
Per una migliore comprensione, tutte le variabili locali dovrebbero anche essere corredate di un commento sufficientemente esplicativo.

Name	Data type	Default value	Retain	Visible in ...	Comment
Input					
1 manual	Bool	false	Non-retentive	<input checked="" type="checkbox"/>	signal select manual mode
2 automatic	Bool	false	Non-retentive	<input checked="" type="checkbox"/>	signal select automatic mode
3 on	Bool	false	Non-retentive	<input checked="" type="checkbox"/>	start signal
4 off	Bool	false	Non-retentive	<input checked="" type="checkbox"/>	stop signal
Output					
5 motor	Bool	false	Non-retentive	<input checked="" type="checkbox"/>	write signal to motor conveyor
InOut					
Static					
6 mem_automatic	Bool	false	Non-retentive	<input checked="" type="checkbox"/>	memory bit mode selection
7 mem_motor	Bool	false	Non-ret...	<input checked="" type="checkbox"/>	memory bit motor conveyor ON
Temp					
8					
9					
10					
11					
12					
13					

12. Dopo aver dichiarato le variabili locali è possibile immettere il programma utente utilizzando i nomi delle variabili. (Le variabili sono identificate dal simbolo '#.'). Per l'esempio in FUP il programma potrebbe essere questo.



13. Fare clic con il tasto destro del mouse sul blocco 'Main[OB1]'.
 Selezionare alla voce 'Switch programming language' (Commuta linguaggio di programmazione) lo schema funzionale 'FBD' (FUP).



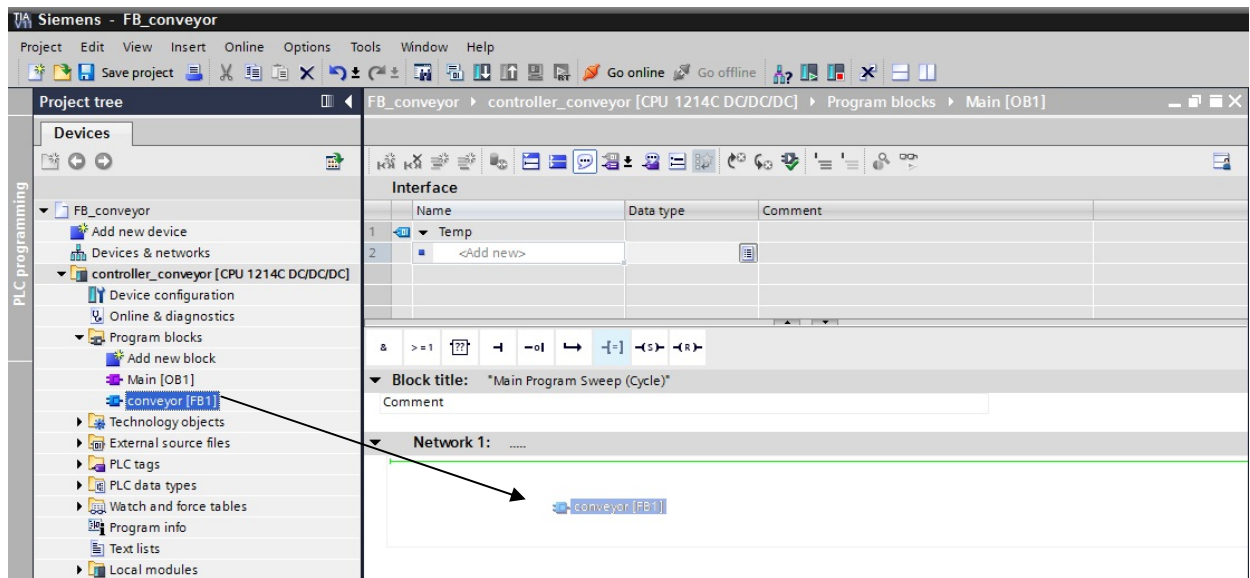
14. Il blocco "conveyor" ora deve essere richiamato dal blocco di programma Main[OB1]. In caso contrario il blocco non verrebbe proprio elaborato.

Aprire il blocco facendo un doppio clic su **'Main[OB1]'**. (→ Main[OB1])

The screenshot displays the Siemens TIA Portal interface for a project named 'FB_conveyor'. The left-hand 'Project tree' shows the hierarchy: 'FB_conveyor' > 'controller_conveyor [CPU 1214C DC/DC/DC]' > 'Program blocks' > 'Main [OB1]'. The main workspace shows the 'conveyor [FB1]' block with the following ladder logic:

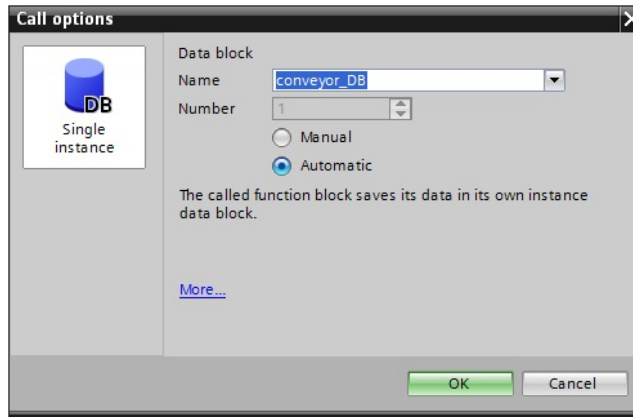
- Block title:** conveyor control
- Network 1: memory bit mode selection**
 - Inputs: #automatic (S), #manual (R1)
 - Logic: An SR (Set Reset) coil is set by #automatic and reset by #manual.
- Network 2: memory bit start motor conveyor in automatic**
 - Inputs: #off, #mem_automatic, #on (S), #mem_motor (R1)
 - Logic: A normally open contact for #off is in series with a normally closed contact for #mem_automatic. This combination is connected to an SR coil set by #on and reset by #mem_motor.
- Network 3: motor conveyor**
 - Inputs: #on, #off, #mem_automatic, #mem_motor, #motor
 - Logic: A normally open contact for #on is in series with a normally closed contact for #off. This combination is connected to a normally open contact for #mem_motor, which is then connected to an SR coil set by #mem_motor and reset by #motor.
- Network 4:**

16. Il blocco "**conveyor[FB1]**" può essere semplicemente trascinato con il mouse nel segmento 1 del blocco Main[OB1]. (→ Nastro[FB1])

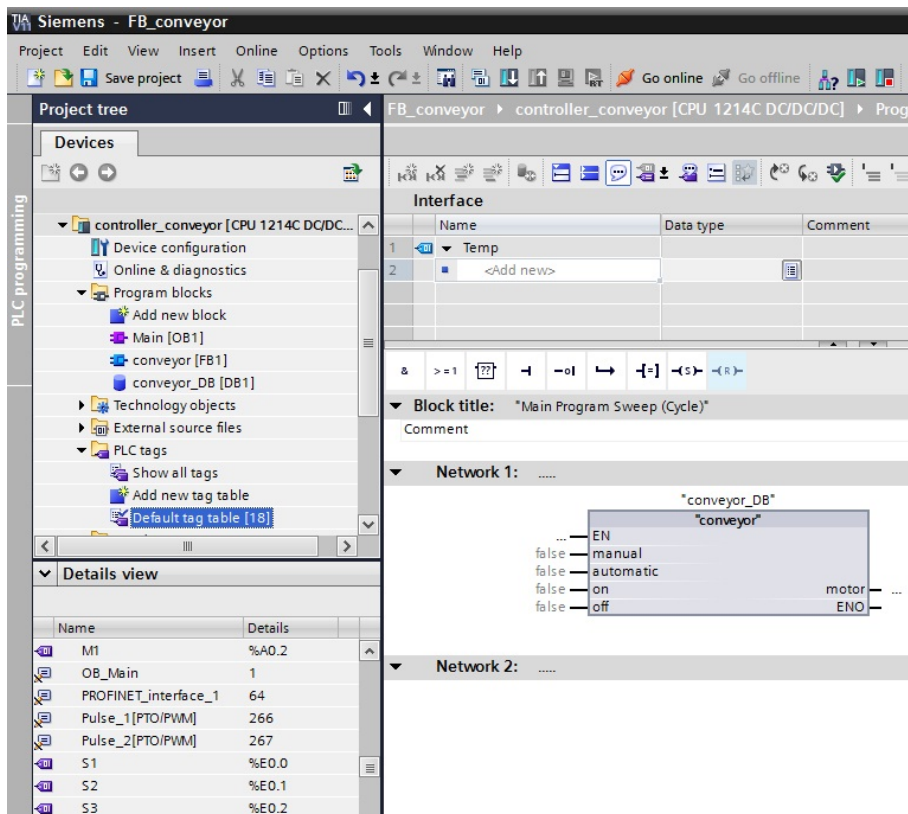


17. Poiché abbiamo a che fare con un blocco funzionale è necessario mettergli a disposizione una memoria. Come memoria in SIMATIC S7-1200 sono disponibili i blocchi dati. Un blocco dati assegnato di questo tipo viene definito **blocco dati di istanza**.

Qui questo blocco deve essere definito e creato 'automaticamente'. (→ Automatic (automatico) → OK)





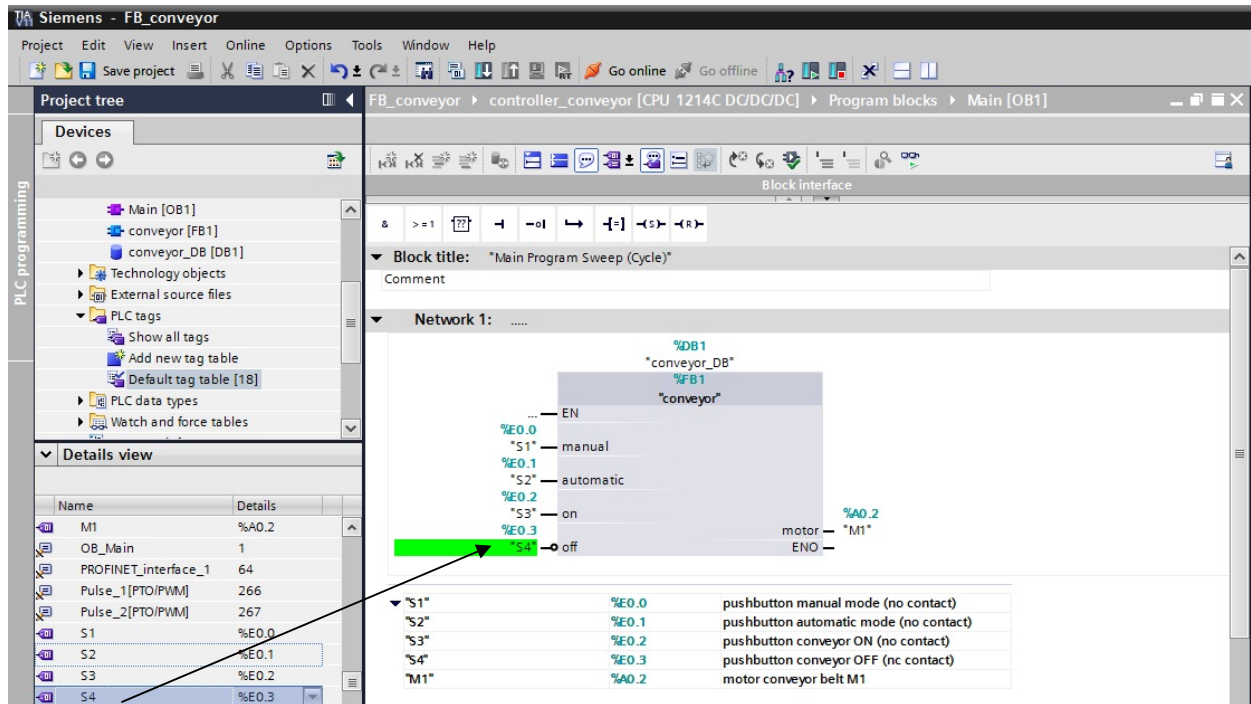
Selezionare la tabella delle variabili standard



18. Nell'OB1 le variabili di ingresso e di uscita ora vanno interconnesse con le variabili PLC qui visualizzate.

È sufficiente trascinare le variabili PLC sulle variabili del blocco.



Facendo clic con il mouse su  Save project (Salva progetto) si salva il progetto. (→ "S1" → "S2" → "S3" → "S4" → "M1" →  Save project)

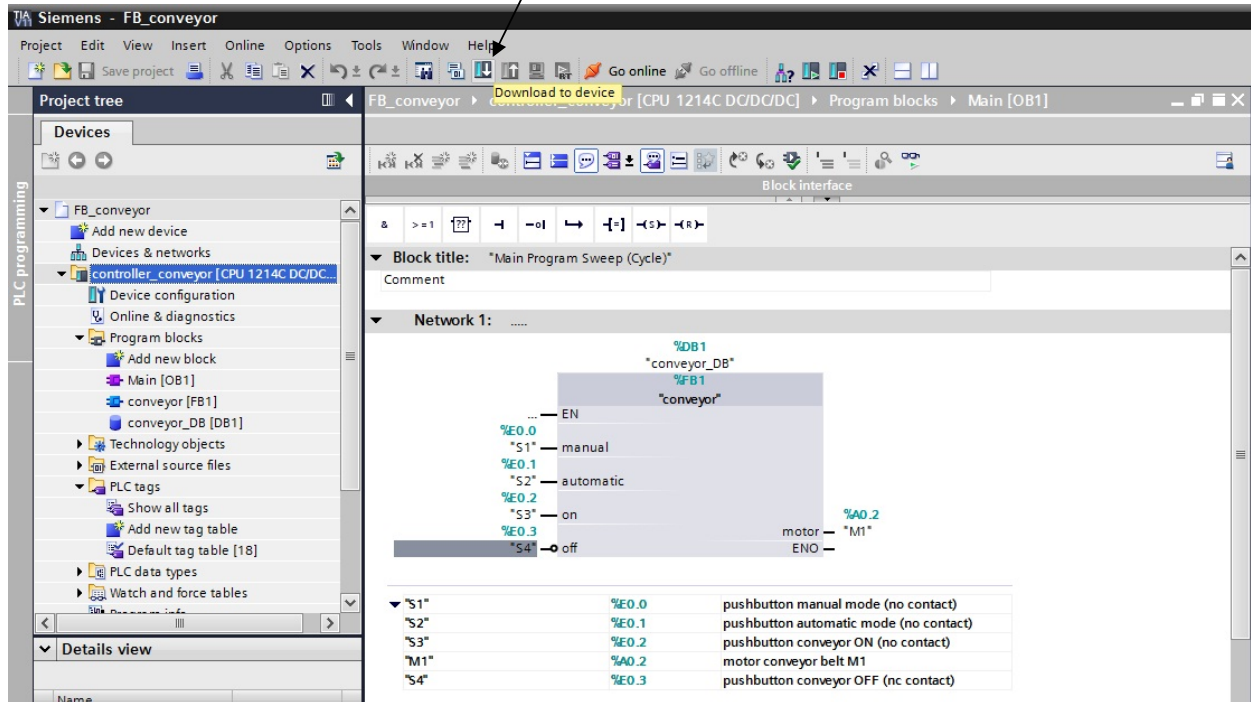


Attenzione!

Il tasto OFF S4 è un contatto normalmente chiuso (NC) e perciò deve essere negato quando viene collegato nel blocco.

Ciò significa che la funzione OFF nel blocco è presente quando il tasto OFF S4 è azionato e quindi nel morsetto %I0.3 non è più presente il segnale.

19. Per caricare l'intero programma nella CPU selezionare prima la cartella **'controller_conveyor'** quindi fare clic sul simbolo  (Carica nel dispositivo). (→ Comando_nastro → )

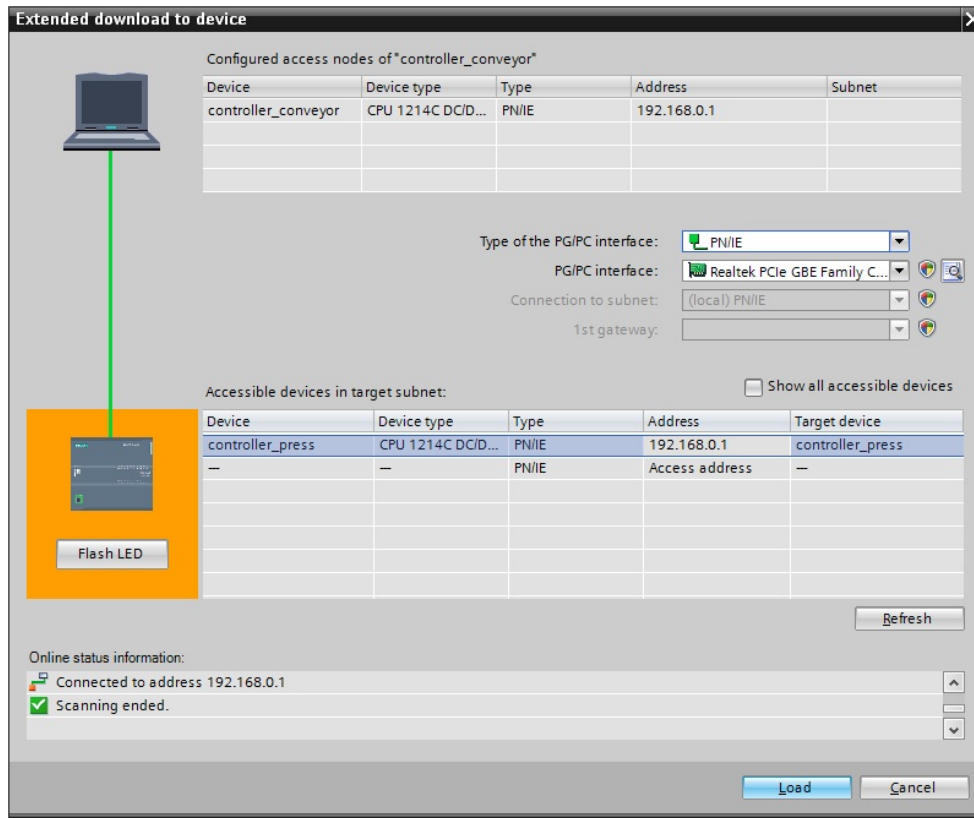


The screenshot shows the Siemens TIA Portal interface. The 'Project tree' on the left has 'controller_conveyor [CPU 1214C DC/DC...]' selected. The main workspace displays the 'Block interface' for 'Main Program Sweep (Cycle)'. The network diagram shows a function block 'conveyor' with inputs %E0.0 (manual), %E0.1 (automatic), %E0.2 (on), and %E0.3 (off), and an output %AO.2 (motor M1). A table below the diagram lists the tag definitions:

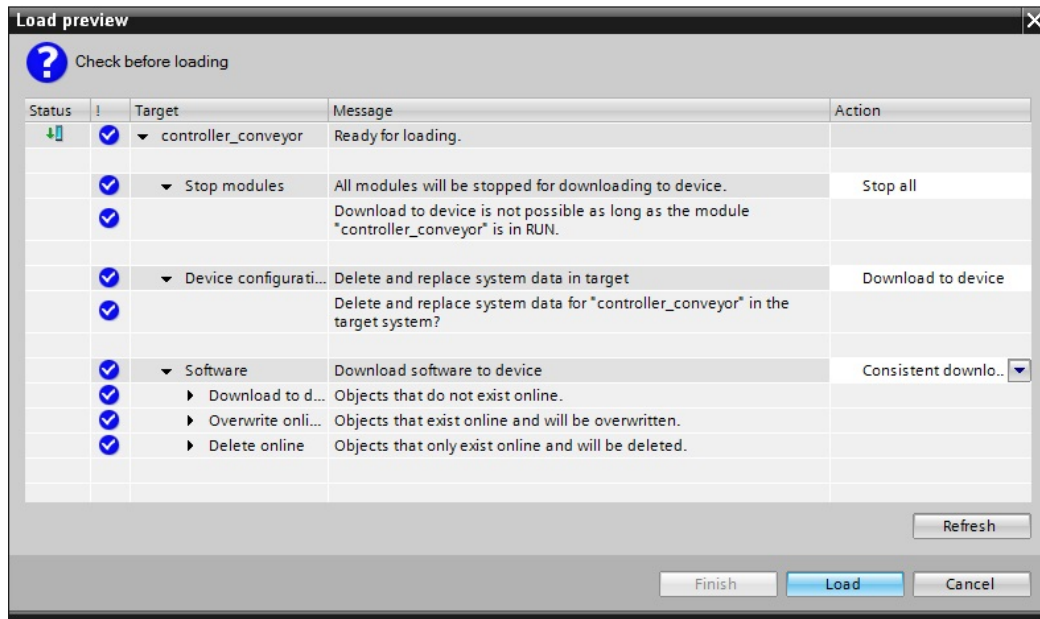
*S1"	%E0.0	pushbutton manual mode (no contact)
*S2"	%E0.1	pushbutton automatic mode (no contact)
*S3"	%E0.2	pushbutton conveyor ON (no contact)
*M1"	%AO.2	motor conveyor belt M1
*S4"	%E0.3	pushbutton conveyor OFF (nc contact)

20. Se si è dimenticato di definire prima l'interfaccia PG/PC viene visualizzata una finestra nella quale è possibile procedere all'operazione.

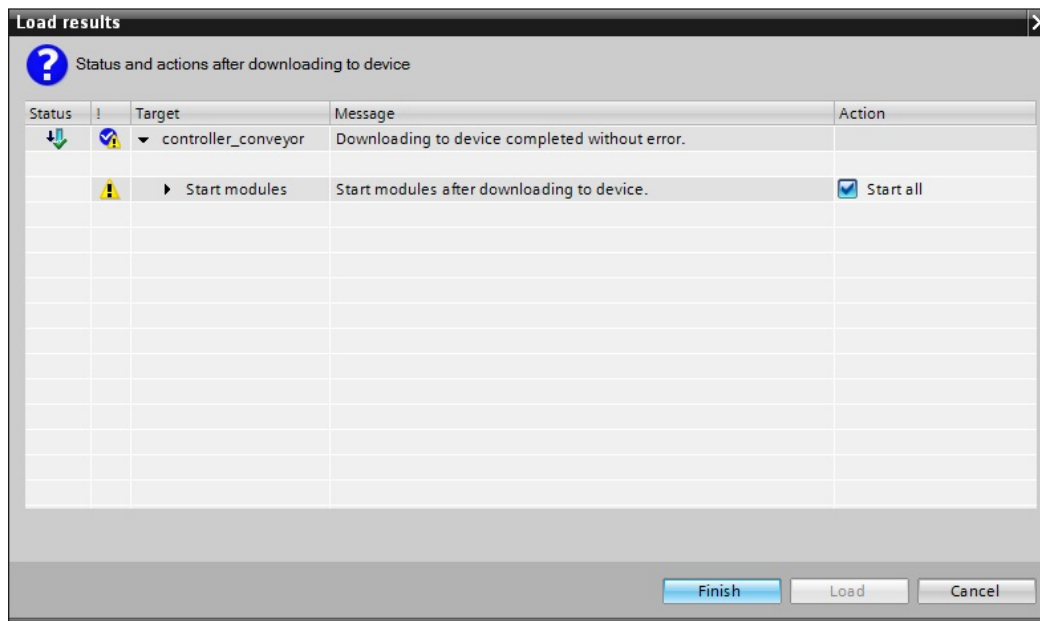
(→ Type of the PG/PC interface → Load)





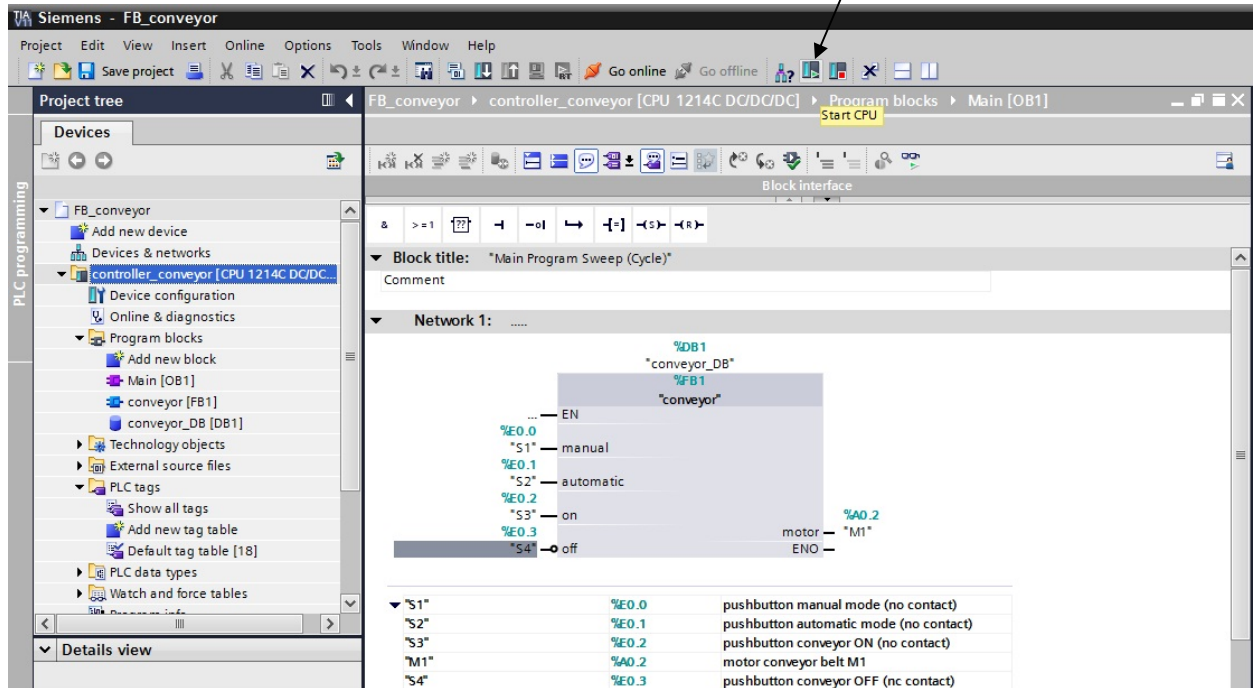
21. Fare clic nuovamente su 'Load'. Durante il caricamento lo stato viene visualizzato in una finestra. (→ Carica)



22. Il caricamento completato senza errori viene visualizzato in una finestra. Fare clic con il mouse su 'Finish'. (→ Fine)



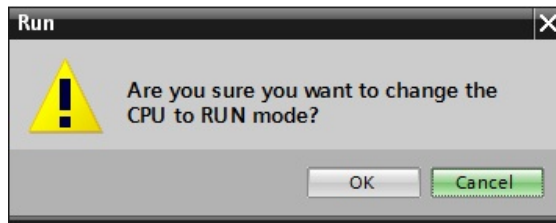
23. Avviare la CPU facendo clic con il mouse sul simbolo . (→ )





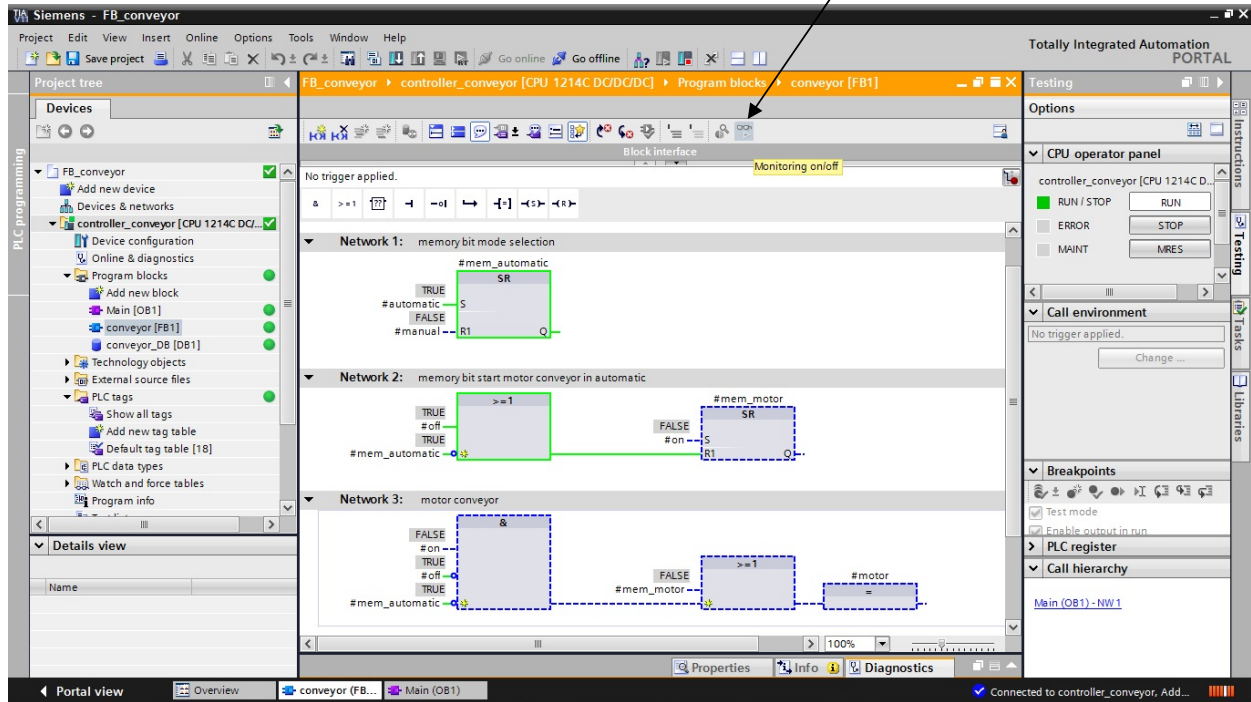
The screenshot shows the Siemens TIA Portal interface. In the top toolbar, the 'Start CPU' button (represented by a green play icon) is highlighted with a yellow box. An arrow points from this button to the text in step 23. The main workspace displays a ladder logic network titled 'Main Program Sweep (Cycle)'. The network contains a function block 'conveyor' (FB1) with inputs '%S1' (manual), '%S2' (automatic), '%S3' (on), and '%S4' (off). The output is '%M1' (motor conveyor belt M1). Below the network, a table lists the variables and their descriptions:

Variable	Address	Description
%S1	%E0.0	pushbutton manual mode (no contact)
%S2	%E0.1	pushbutton automatic mode (no contact)
%S3	%E0.2	pushbutton conveyor ON (no contact)
%M1	%A0.2	motor conveyor belt M1
%S4	%E0.3	pushbutton conveyor OFF (nc contact)

24. Confermare l'avvio della CPU con 'OK'. (→ OK)



25. Con un clic del mouse sul simbolo  Monitoring on/off (Controllo on/off) è possibile controllare sia lo stato delle variabili di ingresso e di uscita nel blocco "conveyor" sia il ciclo di programma nel blocco "conveyor" durante il test del programma. (→ Nastro[FB1] → )



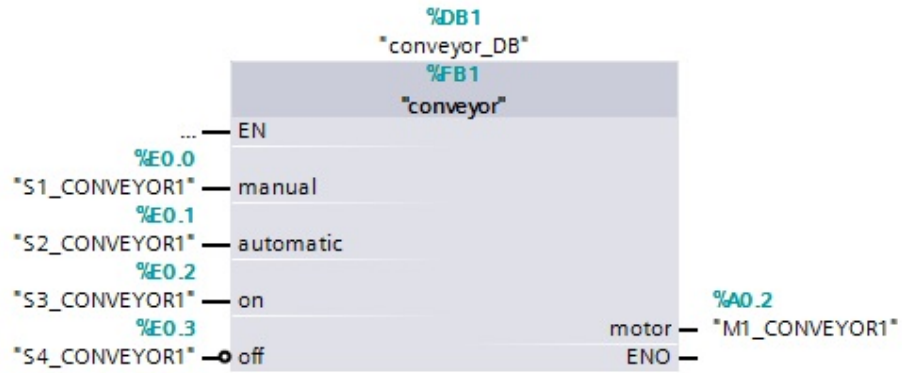
26. Poiché il nostro blocco "conveyor" è stato creato secondo le regole per i blocchi standard (nessuna variabile globale va utilizzata all'interno del blocco!!!!), ora può essere utilizzato e richiamato un numero qualsiasi di volte.

Qui è rappresentata una tabella delle variabili PLC ampliata con gli ingressi e le uscite per due nastri trasportatori.

Default tag table							
	Name	Data type	Address	Retain	Visible..	Acces...	Comment
1	S1_CONVEYOR1	Bool	%E0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	conveyor1 pushbutton manual mode (no contact)
2	S2_CONVEYOR1	Bool	%E0.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	conveyor1 pushbutton automatic mode (no contact)
3	S3_CONVEYOR1	Bool	%E0.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	conveyor1 pushbutton conveyor ON (no contact)
4	S4_CONVEYOR1	Bool	%E0.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	conveyor1 pushbutton conveyor OFF (nc contact)
5	M1_CONVEYOR1	Bool	%A0.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	conveyor1 motor conveyor belt M1
6	S1_CONVEYOR2	Bool	%E0.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	conveyor2 pushbutton manual mode (no contact)
7	S2_CONVEYOR2	Bool	%E0.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	conveyor2 pushbutton automatic mode (no contact)
8	S3_CONVEYOR2	Bool	%E0.6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	conveyor2 pushbutton conveyor ON (no contact)
9	S4_CONVEYOR2	Bool	%E0.7	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	conveyor2 pushbutton conveyor OFF (nc contact)
10	M1_CONVEYOR2	Bool	%A0.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	conveyor2 motor conveyor belt M1
11	<Add new>			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

27. Ora è possibile richiamare il blocco "conveyor" nell'OB1 anche due volte sempre con un diverso collegamento. Per ogni richiamo viene definito un diverso blocco dati di istanza.

▼ Network 1: call program conveyor 1



▼ Network 2: call program conveyor 2



Dallo stesso blocco 'conveyor' è ora possibile comandare i due nastri trasportatori separatamente
 È solo necessario assegnare un diverso blocco dati di istanza a ciascun richiamo.

