

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

## Modulo TIA Portal 010-010

### Avvio alla programmazione con 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](http://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](http://siemens.com/sce/contact)

## Ulteriori informazioni su SCE

[siemens.com/sce](http://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 adattati 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 (TIA) 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](mailto: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.

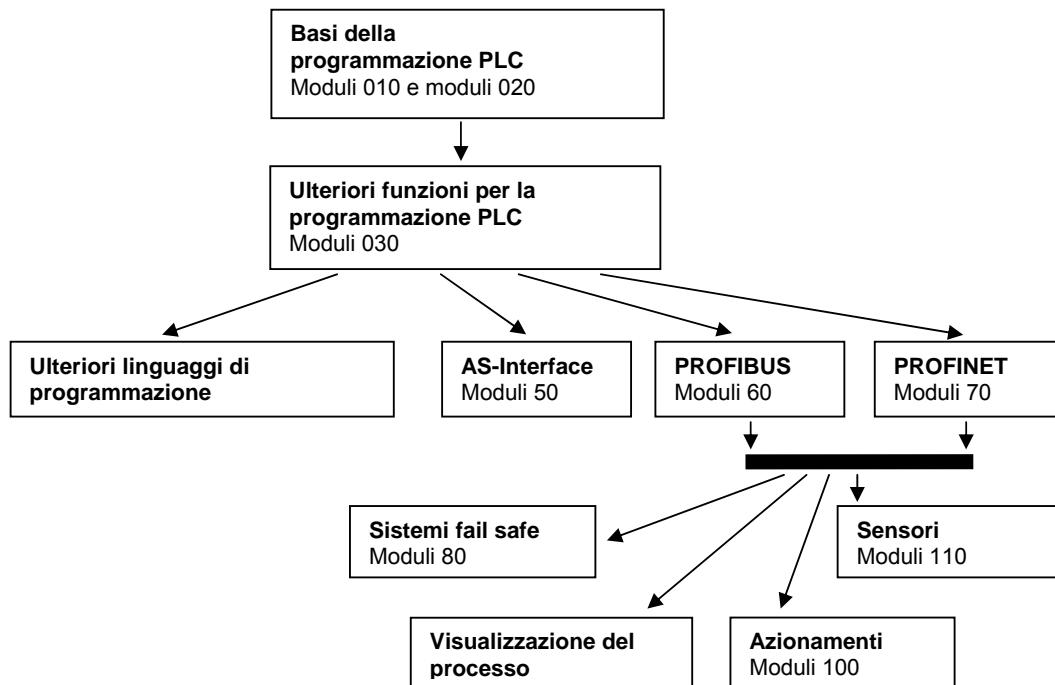
Ringraziamo Michael Dziallas Engineering e tutte le persone coinvolte nella creazione di questa documentazione.

**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.	Installazione del software STEP 7 Professional V11 (TIA Portal V11) .....	7
4.	Collegamento con la CPU tramite TCP/IP e reset alle impostazioni di fabbrica.....	8
5.	Cos'è un PLC e perché viene utilizzato.....	14
5.1	Cosa significa PLC? .....	14
5.2	Come comanda il processo il PLC? .....	14
5.3	Il PLC da dove riceve le informazioni sugli stati del processo? .....	15
5.4	Qual è la differenza tra contatti normalmente aperti e normalmente chiusi?.....	15
5.5	In che modo il SIMATIC S7-1200 indirizza i singoli segnali di ingresso/uscita?.....	16
5.6	Come viene elaborato il programma nel PLC? .....	17
5.7	Come sono le combinazioni logiche nel programma del PLC? .....	18
5.7.1	Combinazione logica AND .....	18
5.7.2	Combinazione logica OR .....	20
5.7.3	Negazione.....	21
5.8	Come si crea il programma del PLC? E come lo si salva nella memoria del PLC? .....	22
6.	Configurazione e utilizzo del controllore SIMATIC S7-1200.....	23
7.	Esempio di programmazione: comando di una pressa.....	27
8.	Programmazione della pressa per il SIMATIC S7-1200 .....	28
8.1.	Vista portale.....	28
8.2.	Vista progetto .....	29

## 1. Prefazione

Il modulo SCE\_DE\_010-010 appartiene per contenuti all'unità didattica **'Basi della programmazione PLC'** e rappresenta un **rapido approccio** alla programmazione di SIMATIC S7 1200 con TIA Portal.



### Obiettivo didattico:

L'obiettivo di questo modulo è insegnare al lettore come programmare un controllore a logica programmabile (PLC) SIMATIC S7-1200 con il tool di programmazione TIA Portal. Il modulo fornisce le nozioni di base e spiega il procedimento con un esempio dettagliato nei passi seguenti:

- Installazione del software e impostazione dell'interfaccia di programmazione
- Cos'è un PLC e come funziona
- Configurazione e utilizzo del PLC SIMATIC S7-1200
- Creazione, comando e test di un programma di esempio

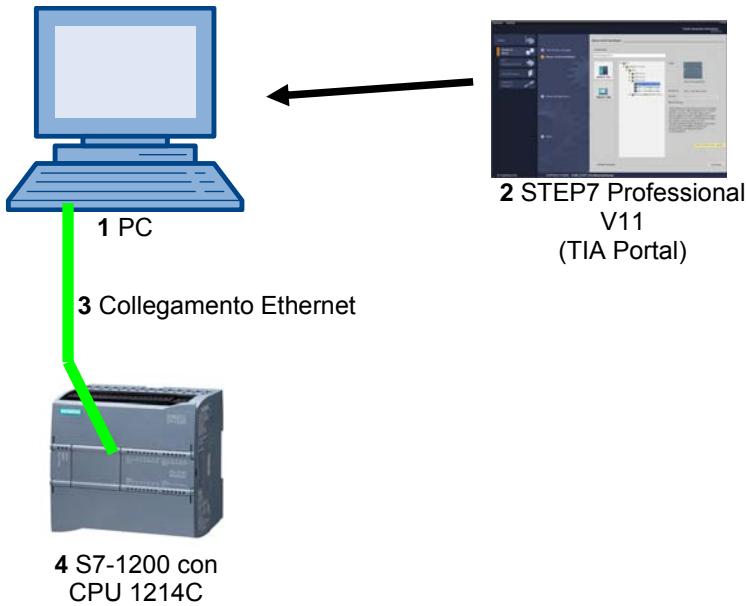
### Presupposti:

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

- Esperienza nell'uso di Windows

**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 operatori

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

### 3. Installazione del software STEP 7 Professional V11 (TIA Portal V11)

STEP 7 Professional viene fornito su DVD.

Per installare STEP 7 Professional procedere nel modo seguente:

1. Inserire il DVD di STEP 7 Professional nell'apposita unità.
2. Si avvierà automaticamente il programma Setup. In caso contrario, avviarlo facendo doppio clic sul file '**→ START.exe**'.
3. Il programma Setup guida l'utente nel corso dell'intera installazione di STEP 7 Professional
4. Per l'utilizzo di STEP 7 Professional è necessario installare sul computer una chiave di licenza.

La chiave di licenza si può trasferire sul computer durante l'installazione da una chiavetta USB in dotazione. In seguito sarà possibile ritrasferire questa chiave di licenza su altri supporti dati con l'applicazione '**Automation License Manager**'. Questa chiave di licenza può trovarsi anche su un altro computer ed essere interrogata da una rete.

#### 4. Collegamento con la CPU tramite TCP/IP e reset alle impostazioni di fabbrica

Per poter programmare un SIMATIC S7-1200 da un PC, un PG o un laptop è necessario un collegamento TCP/IP.

Perché PC e SIMATIC S7-1200 possano comunicare tra loro è importante anche che gli indirizzi IP dei due dispositivi siano compatibili.

Ecco come impostare l'indirizzo IP del computer.

1. Da '**System control**' richiamare '**Network connections**'. Quindi selezionare la voce '**Properties**' della connessione LAN (→ Start → Impostazioni → Pannello di controllo → Connessioni di rete → LAN → Proprietà)
2. Selezionare la voce '**Properties**' da '**Internet Protocol (TCP/IP)**' (→ Protocollo Internet (TCP/IP) → Proprietà)
3. Ora è possibile impostare i campi '**IP address**' e '**Subnet screen form**' e confermare con '**OK**' (→ Usa il seguente indirizzo IP → Indirizzo IP: 192.168.0.99 → Subnet mask 255.255.255.0 → OK → Chiudi)

**Avvertenze per il collegamento alla rete Ethernet (ulteriori informazioni v. appendice V della documentazione didattica):****Indirizzo MAC:**

L'indirizzo MAC si compone di una parte fissa e di una parte variabile. La parte fissa ("indirizzo MAC di base") indica il produttore (Siemens, 3COM...). La parte variabile dell'indirizzo MAC distingue i diversi nodi Ethernet e deve essere univoca a livello mondiale. Su ogni unità è stampato un indirizzo MAC preassegnato dalla fabbrica.

**Campo di valori per l'indirizzo IP:**

L'indirizzo IP è costituito da 4 numeri decimali compresi in un campo di valori da 0 a 255 e separati da un punto, ad es. 141.80.0.16.

**Campo di valori per la maschera di sottorete:**

Questa maschera viene utilizzata per poter rilevare se un nodo - ovvero il suo indirizzo IP - appartiene alla sottorete locale o se è accessibile solo da un router.

La maschera di sottorete è costituita da 4 numeri decimali compresi in un campo di valori da 0 a 255 e separati da un punto, ad es. 255.255.0.0

Nella rappresentazione binaria i 4 numeri decimali della maschera di sottorete devono presentare da sinistra una serie continua di valori "1" e da destra una serie continua di valori "0".

I valori "1" determinano il campo dell'indirizzo IP per il numero di rete. I valori "0" determinano il campo dell'indirizzo IP per l'indirizzo del nodo.

Esempio:

**Valori corretti:** 255.255.0.0 decimale = 1111 1111.1111 1111.0000 0000.0000 0000 binario  
255.255.128.0 decimale = 1111 1111.1111 1111.1000 0000.0000 0000 binario

255.254.0.0 decimale = 1111 1111.1111 1110.0000 0000.0000.0000 binario

**Valore scorretto:** 255.255.1.0 decimale = 1111 1111.1111 1111.0000 0000.0000 0000 binario  
0001.0000 0000 binario

**Campo di valori per l'indirizzo di accoppiamento ad altra rete (router):**

L'indirizzo è costituito da 4 numeri decimali compresi in un campo di valori da 0 a 255 e separati da un punto, ad es. 141.80.0.1.

**Relazione tra indirizzi IP, indirizzo del router e maschera di sottorete:**

L'indirizzo IP e l'indirizzo del router possono differire soltanto nei punti in cui nella maschera di sottorete si trova lo "0".

Esempio: sono stati inseriti: 255.255.255.0 per la maschera di sottorete; 141.30.0.5 per l'indirizzo IP e 141.30.128.1 per l'indirizzo del router.

L'indirizzo IP e l'indirizzo del router possono avere un valore diverso solo nel 4° numero decimale. Nell'esempio tuttavia la posizione 3 è già diversa.

Nell'esempio è necessario modificare alternativamente:

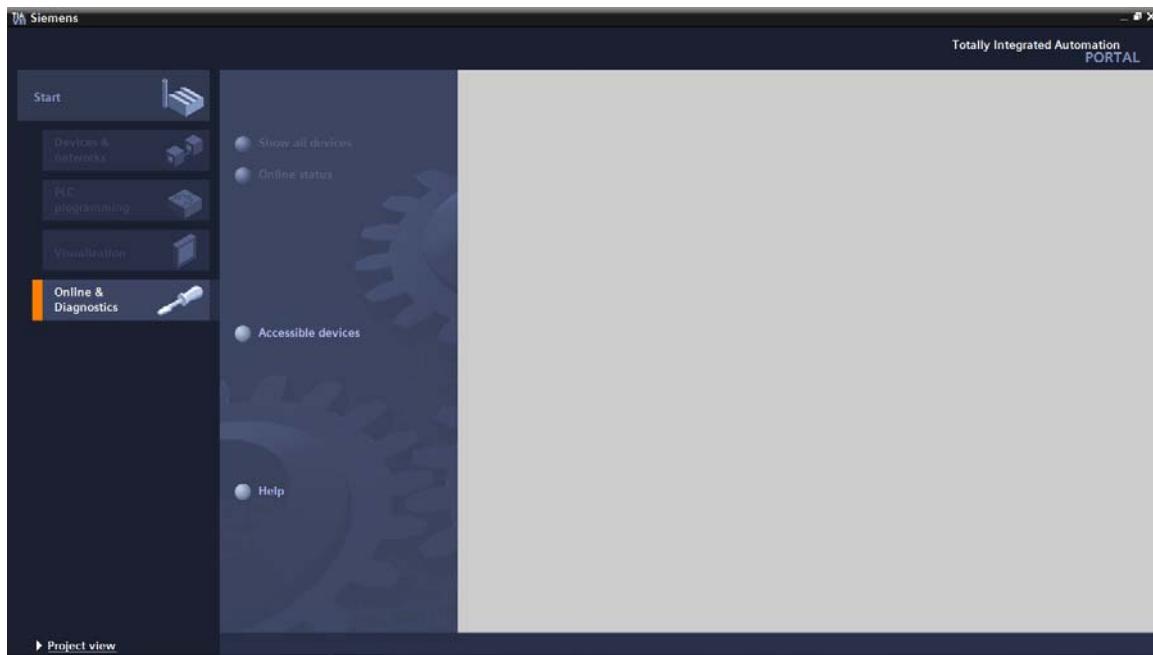
- la maschera della sottorete a: 255.255.0.0 oppure
- l'indirizzo IP a: 141.30.128.5 oppure
- l'indirizzo del router a: 141.30.0.1

L'indirizzo IP del SIMATIC S7-1200 si imposta nel modo seguente.

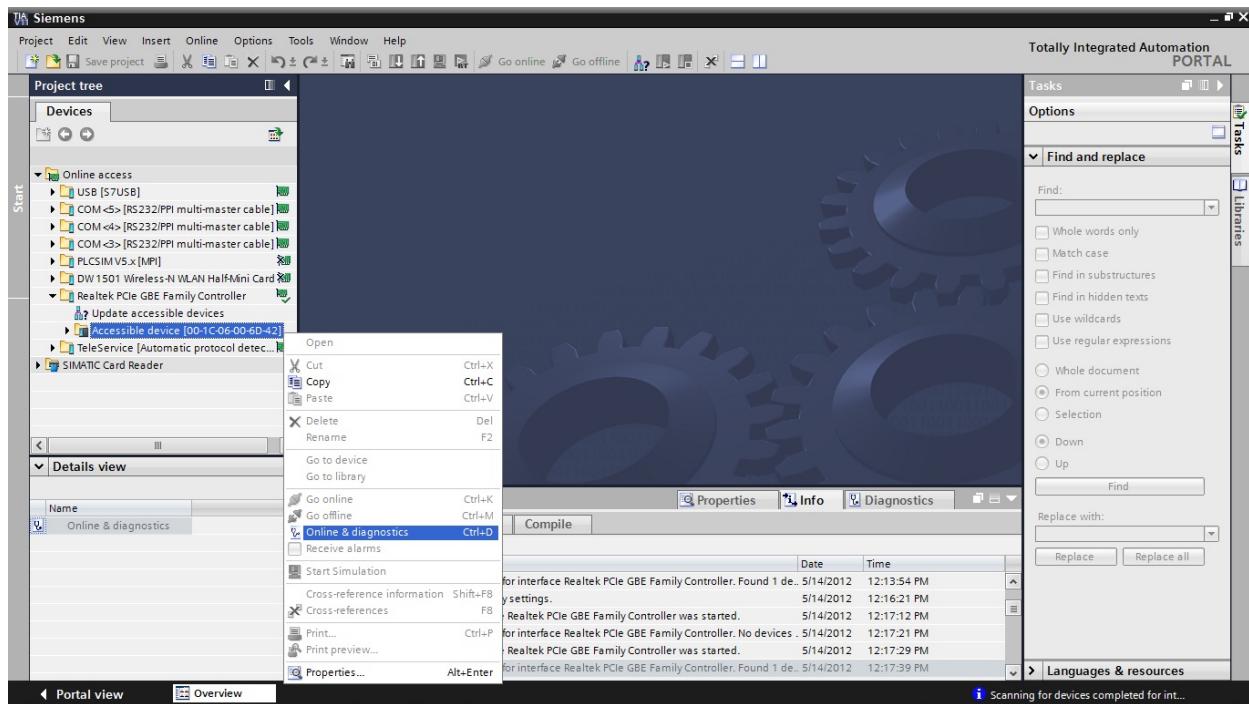
4. Selezionare '**Totally Integrated Automation Portal**', che si richiama qui con un doppio clic. (→ TIA Portal V11)



5. Selezionare '**Online & Diagnostics**' e aprire '**Project view**'. (→ Online & Diagnostica → Vista progetto)



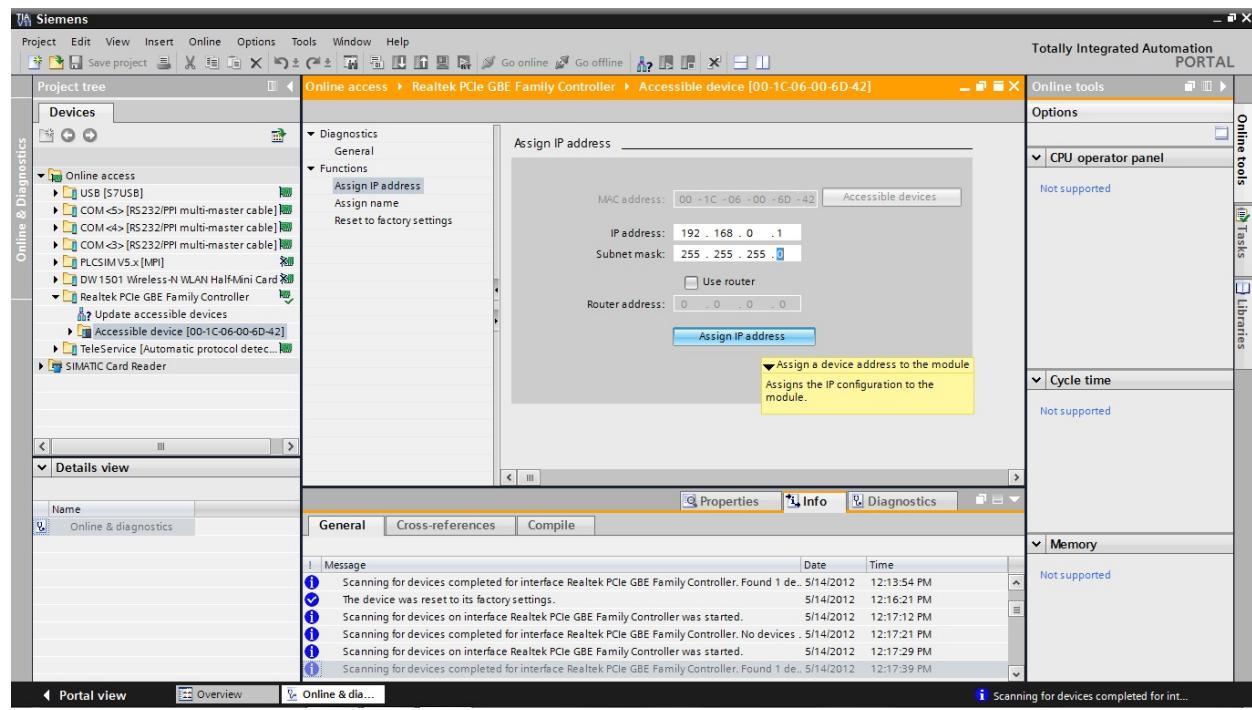
6. Nella navigazione del progetto selezionare alla voce '**Online access**' la scheda di rete già impostata precedentemente. Facendo clic su '**Update accessible devices**' si visualizza l'indirizzo MAC del SIMATIC S7-1200 collegato. Selezionare qui '**Online & Diagnostics**'. ( → Accessi online → ... connessione di rete → Aggiorna nodi accessibili → MAC= ..... → Online & Diagnostica)



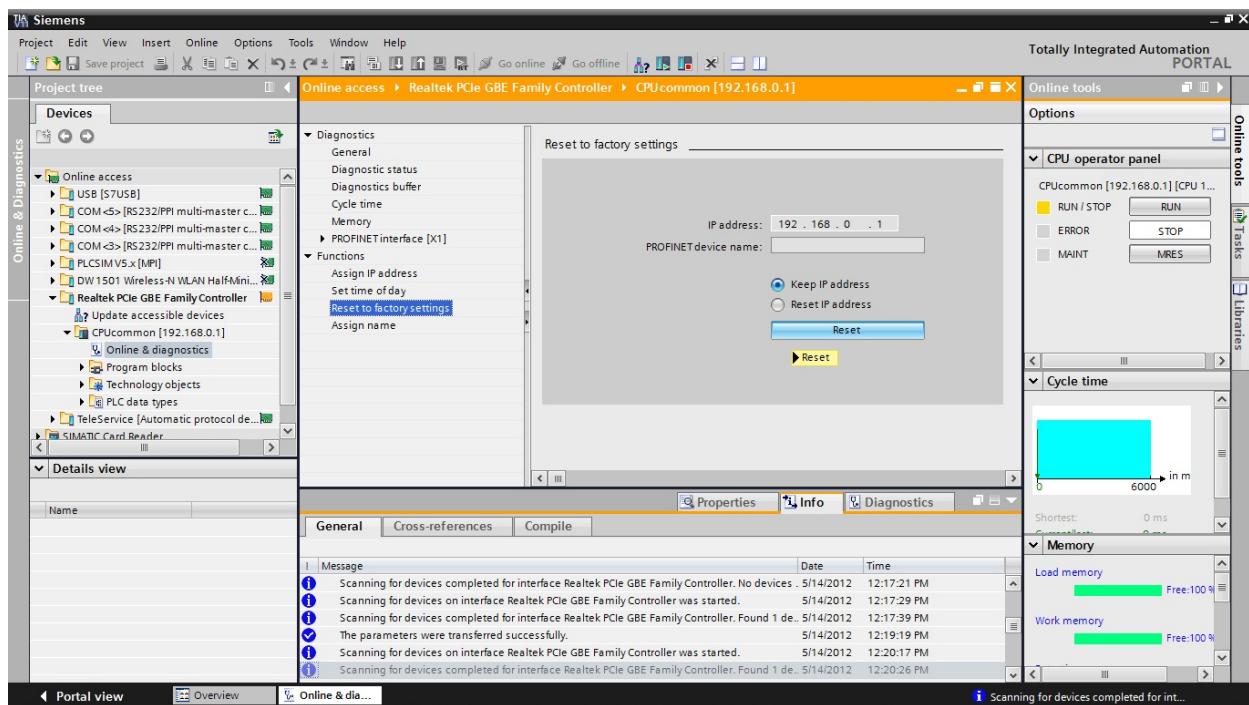
#### Avvertenza:

Se per la CPU è già stato impostato un indirizzo IP viene visualizzata la CPU invece dell'indirizzo MAC.

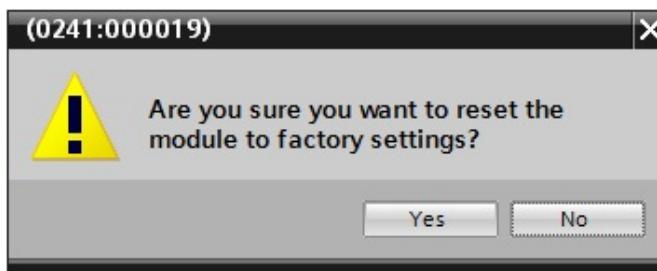
7. Alla voce **'Functions'** selezionare **'Assign IP address'**. Inserire qui **'IP address'** e **'Subnet mask'**. Fare clic su **'Assign IP address'** per assegnare il nuovo indirizzo al controllore SIMATIC S7-1200. ( → Funzioni → Assegna indirizzo IP → Indirizzo IP: 192.168.0.1 → Maschera di sottorete: 255.255.255.0 → Assegna indirizzo IP)



8. Selezionare ora 'Functions' e 'Reset to factory settings'. Lasciare attivata l'impostazione 'Keep IP address' e fare clic su 'Reset'. ( → Funzioni → Resetta alle impostazioni di fabbrica → Mantieni indirizzo IP → Resetta)



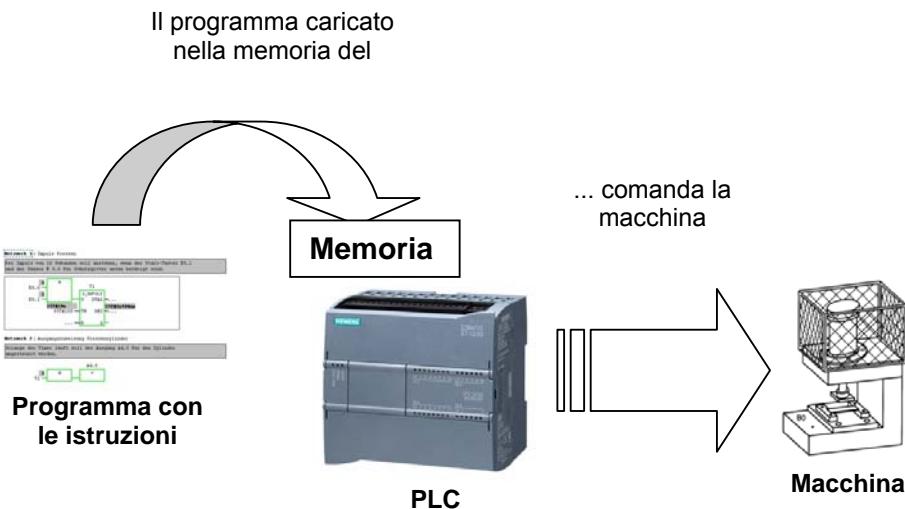
9. Confermare il reset alle impostazioni di fabbrica scegliendo 'OK'. ( → OK)



## 5. Cos'è un PLC e perché viene utilizzato

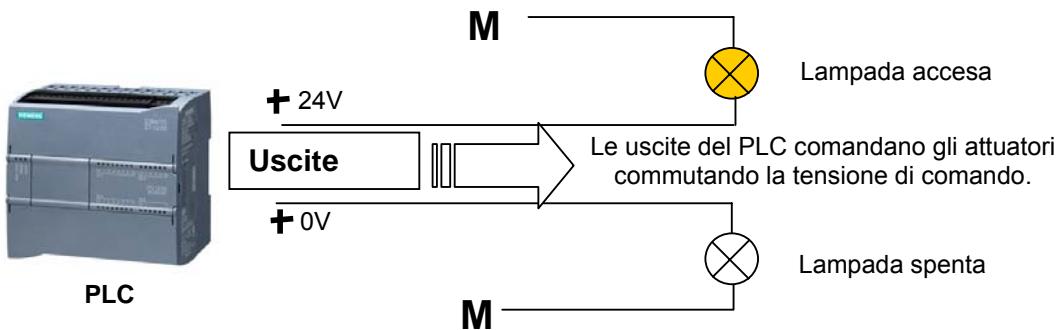
### 5.1 Cosa significa PLC?

**PLC** è la sigla di **Programmable Logic Controller** (controllore a logica programmabile). Questo termine descrive un dispositivo che comanda un processo (ad es. una macchina per la stampa di giornali, un impianto per il riempimento di sacchi di cemento, una pressa per pezzi stampati in plastica ecc... ). Il PLC funziona seguendo le istruzioni di un programma contenuto in una memoria del dispositivo.



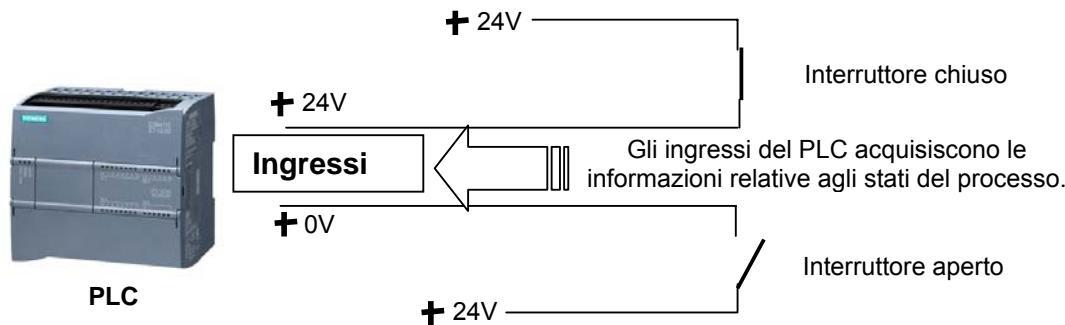
### 5.2 Come comanda il processo il PLC?

Il PLC comanda il processo nel modo seguente: cosiddetti **attuatori** vengono collegati con una tensione di comando - ad es. a 24V - dalle connessioni del PLC definite **uscite**. In questo modo è possibile accendere e spegnere un motore, aprire e chiudere una valvola o accendere e spegnere una lampada.



### 5.3 Il PLC da dove riceve le informazioni sugli stati del processo?

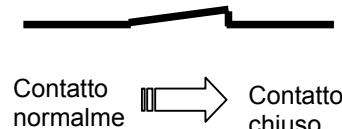
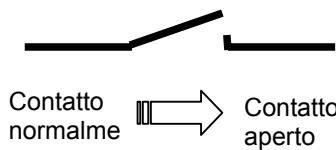
Il PLC riceve le informazioni sul processo dai cosiddetti **datori di segnale**, che sono cablati con gli **ingressi** del PLC. Questi datori di segnale possono essere ad es. sensori che riconoscono se un pezzo si trova in una determinata posizione o anche semplici interruttori o tasti che possono essere aperti o chiusi. Inoltre essi si distinguono in **contatti normalmente chiusi**, che a riposo sono chiusi, e **contatti normalmente aperti**, che a riposo sono aperti.



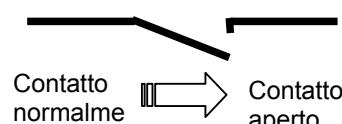
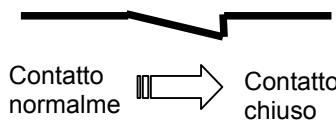
### 5.4 Qual è la differenza tra contatti normalmente aperti e normalmente chiusi?

I datori di segnale si suddividono in **contatti normalmente aperti** e **contatti normalmente chiusi**.

L'interruttore qui raffigurato è un contatto normalmente aperto, ovvero che si chiude esattamente quando viene azionato.



L'interruttore qui raffigurato è un contatto normalmente chiuso, ovvero che si chiude esattamente quando non è azionato.



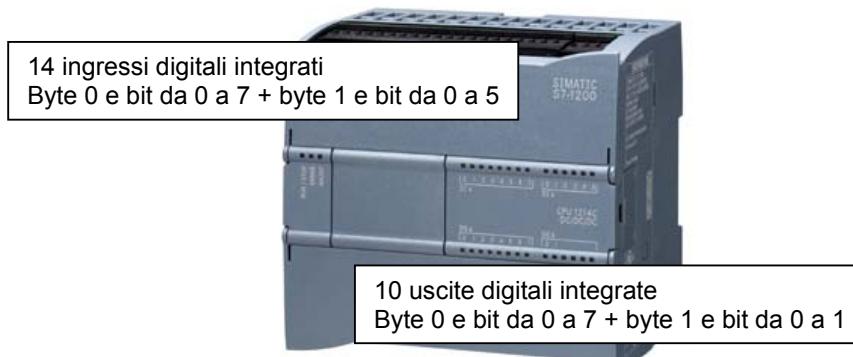
## 5.5 In che modo il SIMATIC S7-1200 indirizza i singoli segnali di ingresso/uscita?

L'indicazione di un determinato ingresso o di una determinata uscita all'interno del programma è definita indirizzamento.

Gli ingressi e le uscite dei PLC sono per lo più riuniti in gruppi di 8 su unità di ingressi o di uscite digitali. Questi otetti sono definiti **byte**. A ciascun gruppo di questo tipo viene assegnato un numero, il cosiddetto **indirizzo a byte**.

Per indirizzare i singoli ingressi e uscite all'interno di un byte ogni byte viene suddiviso in otto singoli **bit** numerati dal bit 0 al bit 7. In questo modo si ottiene l'**indirizzo a bit**.

Il PLC qui rappresentato ha i byte di ingresso 0 e 1 e i byte di uscita 0 e 1.



Ora, per indirizzare ad es. il quinto ingresso digitale, occorre indicare l'indirizzo seguente:

**%I 0 . 4**

**%I** identifica il tipo di indirizzo come ingresso, **0** l'indirizzo a byte e **4** l'indirizzo a bit.  
Indirizzo a byte e indirizzo a bit sono sempre separati da un punto.

**Avvertenza:** per l'indirizzo a bit qui nel quinto ingresso è indicato un **4** perché si inizia a contare da 0.

Ora, per indirizzare ad es. la decima uscita, occorre indicare l'indirizzo seguente:

**%Q 1 . 1**

**%Q** identifica il tipo di indirizzo come uscita, **1** l'indirizzo a byte e **1** l'indirizzo a bit.  
Indirizzo a byte e indirizzo a bit sono sempre separati da un punto.

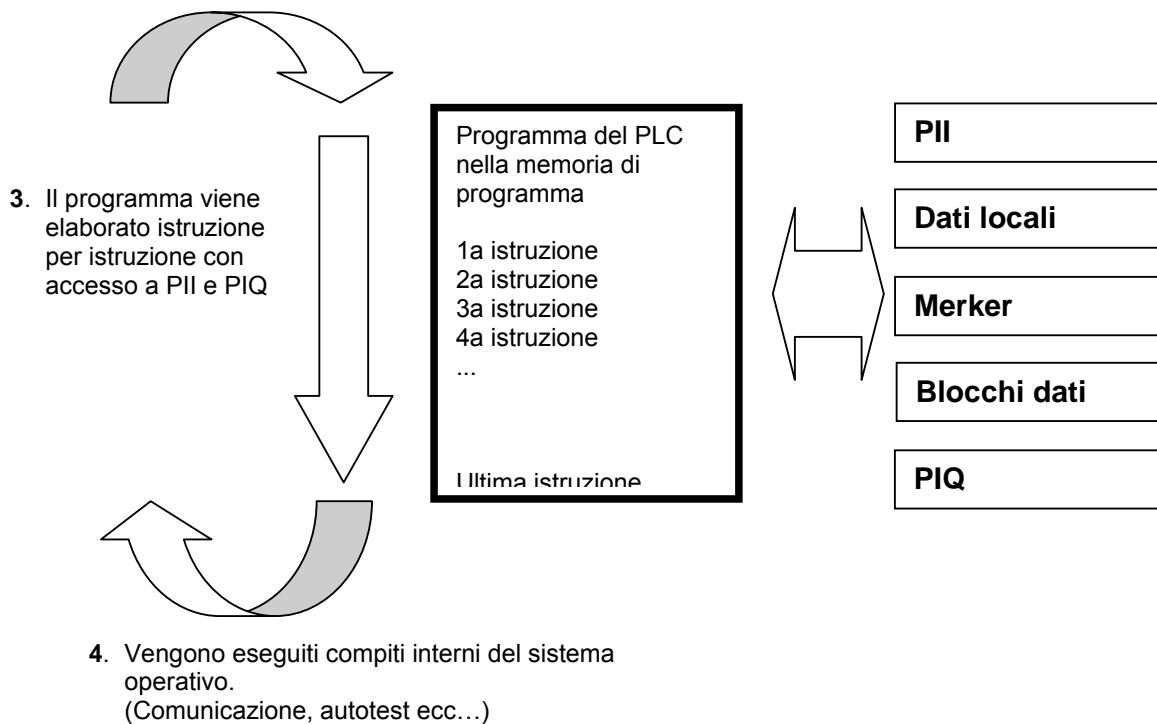
**Avvertenza:**

per l'indirizzo a bit qui nella decima uscita è indicato un **1** perché si inizia a contare da 0.

## 5.6 Come viene elaborato il programma nel PLC?

In un PLC il programma viene elaborato ciclicamente nella sequenza qui riportata:

1. In primo luogo viene trasferito lo stato dall'immagine di processo delle uscite (**PIQ**) alle uscite e queste vengono attivate o disattivate.
  2. Successivamente il **processore**, che praticamente costituisce il cervello del PLC, verifica se i singoli ingressi portano la tensione o meno. Questo stato degli ingressi viene salvato nell'immagine di processo degli ingressi (**PII**). Per gli ingressi che portano la tensione viene salvata l'informazione 1 o "High", per quelli che non portano la tensione viene salvata l'informazione 0 o "Low".
  3. Questo processore elabora poi il programma salvato nella memoria di programma. Il programma consiste in un elenco di combinazioni logiche e istruzioni che vengono elaborate consecutivamente. Per ottenere l'informazione di ingresso necessaria il processore accede alla PII letta in precedenza e i risultati logici combinatori vengono scritti in una cosiddetta immagine di processo delle uscite (**PIQ**). Se necessario, il processore accede anche ad altre aree di memoria durante l'elaborazione del programma, ad es. ai dati locali dei sottoprogrammi, a blocchi dati e merker.
  4. Per finire vengono eseguiti altri compiti interni del sistema operativo, come l'autotest e la comunicazione. In seguito l'elaborazione continua con il punto 1.
1. Lo stato viene trasferito dalla PIQ alle uscite.
  2. Lo stato degli ingressi viene salvato nella PII.



### Avvertenza:

Il tempo impiegato dal processore per eseguire questa sequenza è definito tempo di ciclo.

Il tempo di ciclo a sua volta varia sia in funzione del numero e del tipo di istruzioni sia della potenza del processore.

## 5.7 Come sono le combinazioni logiche nel programma del PLC?

Le combinazioni logiche vengono utilizzate per definire le condizioni di attivazione di un'uscita.

Nel programma del PLC le istruzioni si possono creare con i linguaggi di programmazione schema a contatti (**KOP**) o schema funzionale (**FUP**).

Per ragioni di semplificazione ci limiteremo in questo contesto al linguaggio **FUP**.

Esistono numerose e svariate combinazioni logiche che si possono utilizzare nei programmi dei PLC.

Tuttavia, le combinazioni logiche **AND** e **OR**, così come la **NEGAZIONE** di un ingresso, sono le più utilizzate e pertanto sono spiegate brevemente qui di seguito sulla base di alcuni esempi.

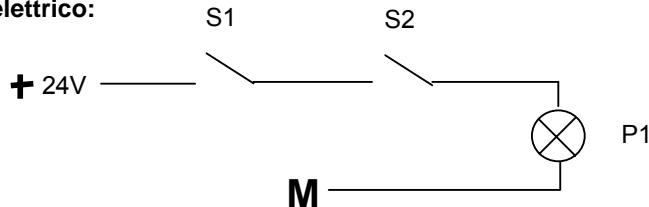
**Avvertenza:** tutte le informazioni sulle ulteriori combinazioni logiche sono riportate nella Guida in linea, di facile e rapida consultazione.

### 5.7.1 Combinazione logica AND

#### Esempio di combinazione logica AND:

Una lampada deve accendersi quando due interruttori vengono azionati contemporaneamente come contatti normalmente aperti.

##### Schema elettrico:



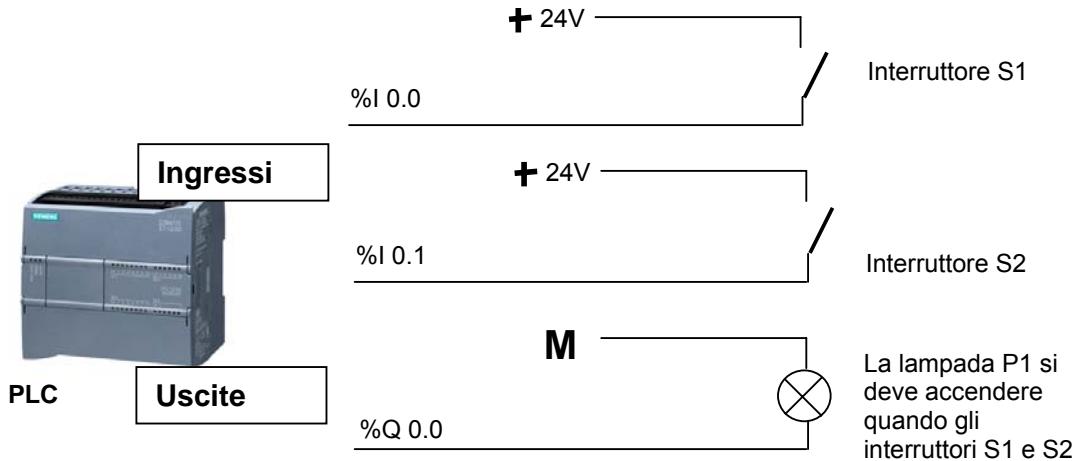
##### Spiegazione:

La lampada si accende esattamente quando vengono azionati entrambi gli interruttori.

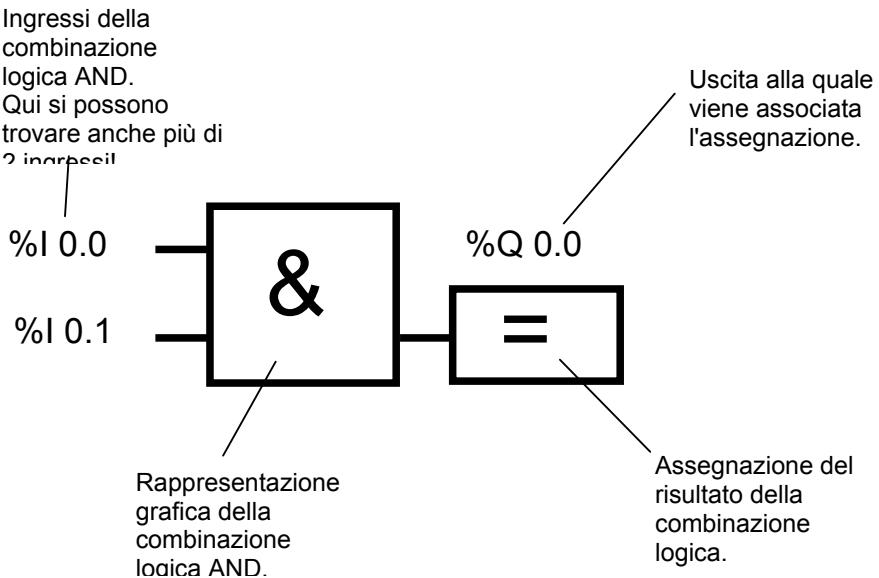
Quindi quando gli interruttori S1 e S2 sono attivati la lampada P1 è accesa.

**Collegamento del PLC:**

Per applicare questa logica al programma di un PLC naturalmente entrambi gli interruttori devono essere cablati con gli ingressi del PLC. Qui S1 è cablato con l'ingresso I 0.0 e S2 con l'ingresso I 0.1. Inoltre, la lampada P1 deve essere collegata a un'uscita, ad es. Q 0.0.

**Combinazione logica AND in FUP:**

Nello schema funzionale FUP la combinazione logica AND si programma con un linguaggio grafico ed è rappresentata nel modo seguente:

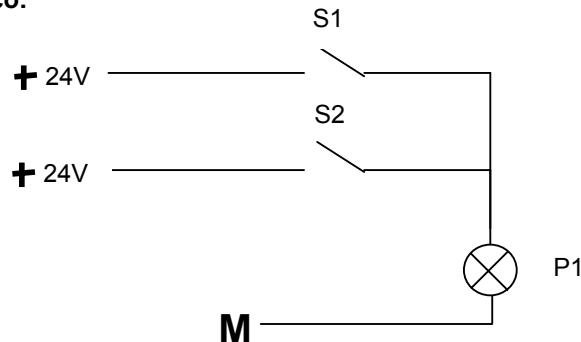


## 5.7.2 Combinazione logica OR

### Esempio di combinazione logica OR:

Una lampada deve accendersi quando uno o entrambi gli interruttori vengono azionati come contatti normalmente aperti.

#### Schema elettrico:

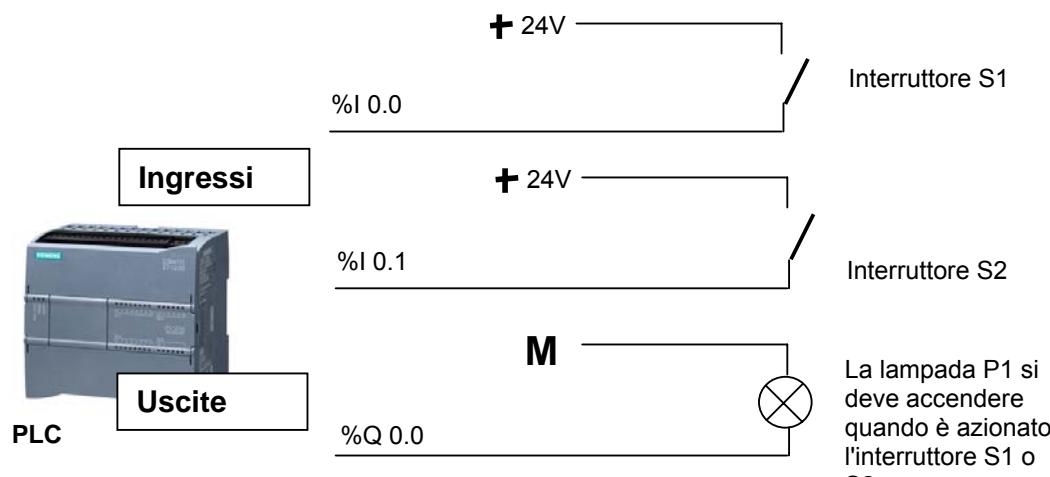


#### Spiegazione:

La lampada si accende esattamente quando uno o entrambi gli interruttori vengono azionati. Quindi quando viene azionato l'interruttore S1 o S2 la lampada P1 è accesa.

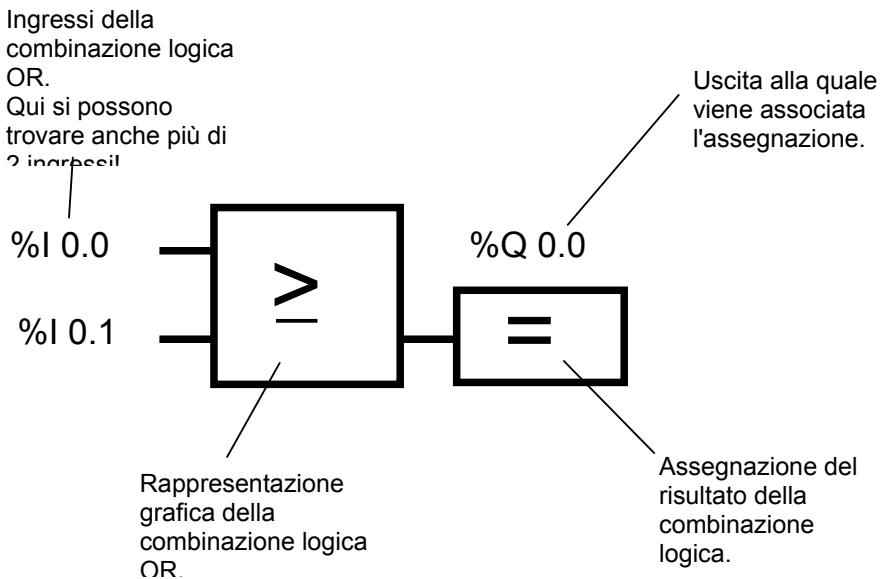
#### Collegamento del PLC:

Per applicare questa logica al programma di un PLC naturalmente entrambi gli interruttori devono essere cablati con gli ingressi del PLC. Qui S1 è cablato con l'ingresso I 0.0 e S2 con l'ingresso I 0.1. Inoltre, la lampada P1 deve essere collegata a un'uscita, ad es. Q 0.0.



### Combinazione logica OR in FUP:

Nello schema funzionale FUP la combinazione logica OR si programma con un linguaggio grafico ed è rappresentata nel modo seguente:

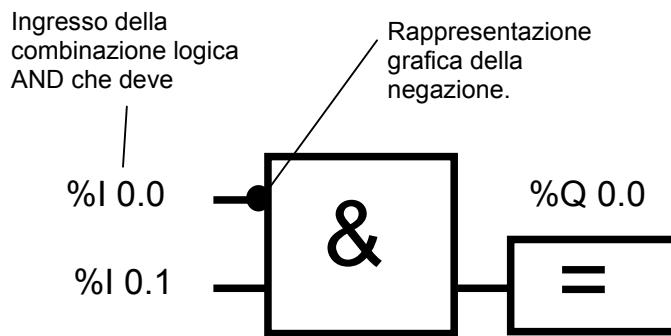


### 5.7.3 Negazione

Nelle combinazioni logiche è spesso necessario verificare se un **contatto normalmente aperto** **NON è azionato** o se un **contatto normalmente chiuso** **è azionato** e quindi nell'ingresso corrispondente non passa la tensione.

Questa interrogazione è possibile programmando una **negazione** nell'ingresso della combinazione logica AND oppure OR.

Nello schema funzionale FUP la negazione di un ingresso in una combinazione logica AND si programma con la seguente rappresentazione grafica:



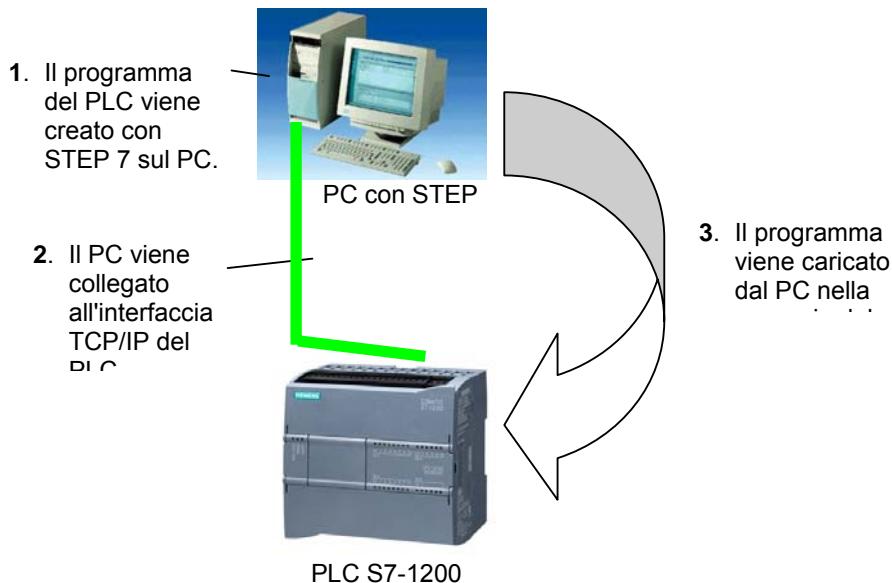
Qui perciò viene applicata la tensione all'uscita %Q 0.0 esattamente quando %I 0.0 non è collegato e %I 0.1 è collegato.

## 5.8 Come si crea il programma del PLC? E come lo si salva nella memoria del PLC?

Il programma del PLC si crea con l'applicazione software STEP 7 su un PC nel quale viene memorizzato temporaneamente.

Dopo aver collegato il PC all'interfaccia TCP/IP del PLC è possibile caricare il programma nella memoria del PLC con una funzione di caricamento.

Per la successiva elaborazione del programma nel PLC il PC non è più necessario.



### Avvertenza:

La sequenza esatta è descritta passo per passo nei capitoli che seguono.

## 6. Configurazione e utilizzo del controllore SIMATIC S7-1200

### Gamma delle unità:

SIMATIC S7-1200 è un sistema di automazione modulare e offre la seguente gamma di unità:

- Unità centrali (CPU) con diversa potenza, ingressi/uscite integrati e interfaccia PROFINET (ad es. CPU1214C)



- Alimentatore di rete (PM) con ingresso 120/230V AC, 50Hz / 60Hz, 1.2A / 0.7A e uscita 24V DC / 2.5A



- Signal board (SB) per l'inserimento di ingressi/uscite analogici o digitali senza tuttavia modificare le dimensioni della CPU.  
(Le signal board si possono utilizzare con le CPU 1211C / 1212C e 1214C.)



- Unità di ingresso/uscita (SM) per ingressi e uscite digitali e analogici  
(Con le CPU 1212C si possono utilizzare max. 2 SM e con le 1214C max. 8 SM.)



- Moduli di comunicazione (CM) per la comunicazione seriale RS232 / RS 485  
(Con le CPU 1211C / 1212C e 1214C si possono utilizzare fino a 3 CM.)



- Compact Switch Module (CSM) con 4x prese RJ45 10/ 100 Mbit/s



- SIMATIC memory card da 2MB o 24 MB per la memorizzazione dei dati del programma e una rapida sostituzione delle CPU in caso di manutenzione

**Avvertenza:**

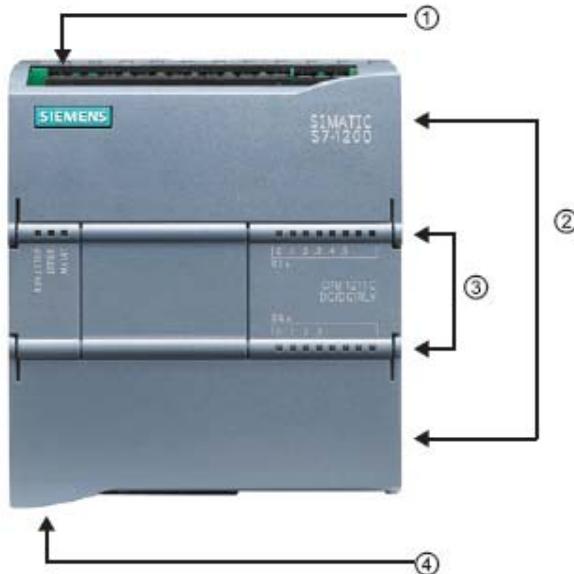
Per questo modulo è sufficiente una qualsiasi CPU con ingressi e uscite digitali integrati.

**Elementi importanti della CPU:**

Con un'alimentazione integrata (connessione 24V) e ingressi e uscite integrati, la CPU S7-1200 è pronta per l'uso senza componenti aggiuntivi.

Per la comunicazione con un dispositivo di programmazione la CPU è dotata di una porta TCP/IP integrata.

Attraverso una rete ETHERNET la CPU può così comunicare con pannelli operatore HMI o altre CPU.



- ① Connessione 24V
- ② Morsettiera a innesto per il cablaggio dell'applicazione (dietro le coperture di protezione)
- ③ LED di stato per gli I/O integrati e lo stato di funzionamento della CPU
- ④ Porta TCP/IP (sul lato inferiore della CPU)

La **SIMATIC memory card (MC)** salva il programma, i dati, i dati di sistema, i file e i progetti. Questa scheda di memoria si può utilizzare per:

- Trasferire un programma in diverse CPU
- Aggiornare il firmware di CPU, unità di ingressi/uscite SM e moduli di comunicazione CM



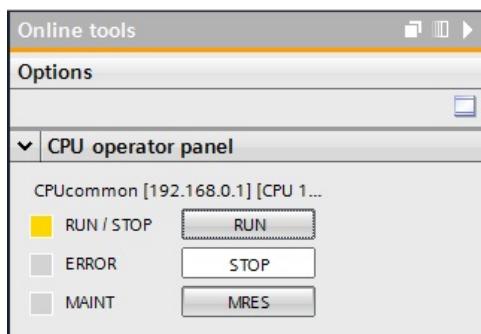
## Stati di funzionamento della CPU

La CPU ha i tre stati di funzionamento seguenti:

- In stato di funzionamento **STOP** la CPU non esegue il programma ed è possibile caricare un progetto.
- In stato di funzionamento **STARTUP** la CPU si avvia.
- In stato di funzionamento **RUN** viene eseguito ciclicamente il programma. Quando la CPU è in stato di funzionamento RUN non è possibile caricare i progetti.

La CPU non dispone di un interruttore fisico con il quale modificare lo stato di funzionamento.

Lo stato di funzionamento (**STOP** o **RUN**) si modifica con il pulsante sul pannello di comando del software STEP7 Basic. Inoltre questo pannello comprende un pulsante **MRES** per la cancellazione totale della memoria e mostra i LED di stato della CPU.



Il **LED di stato RUN/STOP** sulla parte frontale della CPU segnala lo stato di funzionamento attuale della CPU cambiando colore.



- La luce **gialla** indica lo stato di funzionamento **STOP**.
- La luce **verde** indica lo stato di funzionamento **RUN**.
- Una luce che **lampeggia** indica lo stato di funzionamento **STARTUP**.

Inoltre sono disponibili i LED **ERROR** per segnalare gli errori e **MAINT** per segnalare la necessità di manutenzione.

## 7. Esempio di programmazione: comando di una pressa

Il nostro primo programma consiste nella programmazione del comando di una pressa.

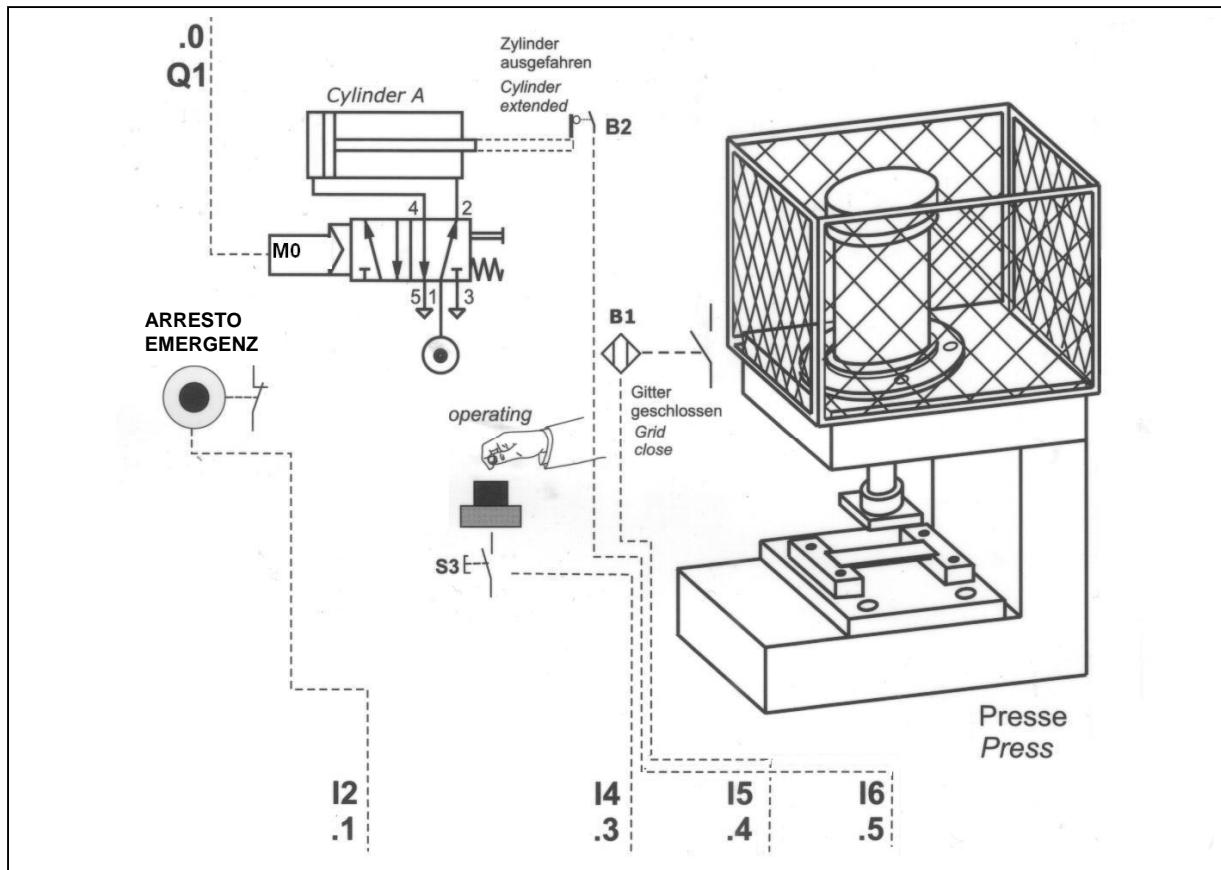
Una pressa dotata di dispositivo di protezione deve essere attivata con un tasto START S3 solo quando la griglia di protezione è chiusa. Questo stato viene controllato da un sensore B1 'Griglia di protezione chiusa'.

Quando la griglia è chiusa, viene azionata una valvola 5/2 M0 per il cilindro della pressa in modo che il pezzo stampato di plastica possa essere compresso.

La pressa deve risalire quando viene azionato il tasto ARRESTO DI EMERGENZA (NC), se il sensore B1 'Griglia di protezione' non risponde più o se risponde il sensore B2 'Cilindro estratto'.

### **Lista di attribuzione:**

Indirizzo	Simbolo	Commento
%I 0.1	EMERG_OFF	Tasto ARRESTO DI EMERGENZA NC
%I 0.3	S3	Tasto di avvio S3 NO
%I 0.4	B1	Sensore 'Griglia di protezione chiusa' NO
%I 0.5	B2	Sensore 'Cilindro estratto' NO
%Q 0.0	M0	Estrae il cilindro A



## 8. Programmazione della pressa per il SIMATIC S7-1200

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

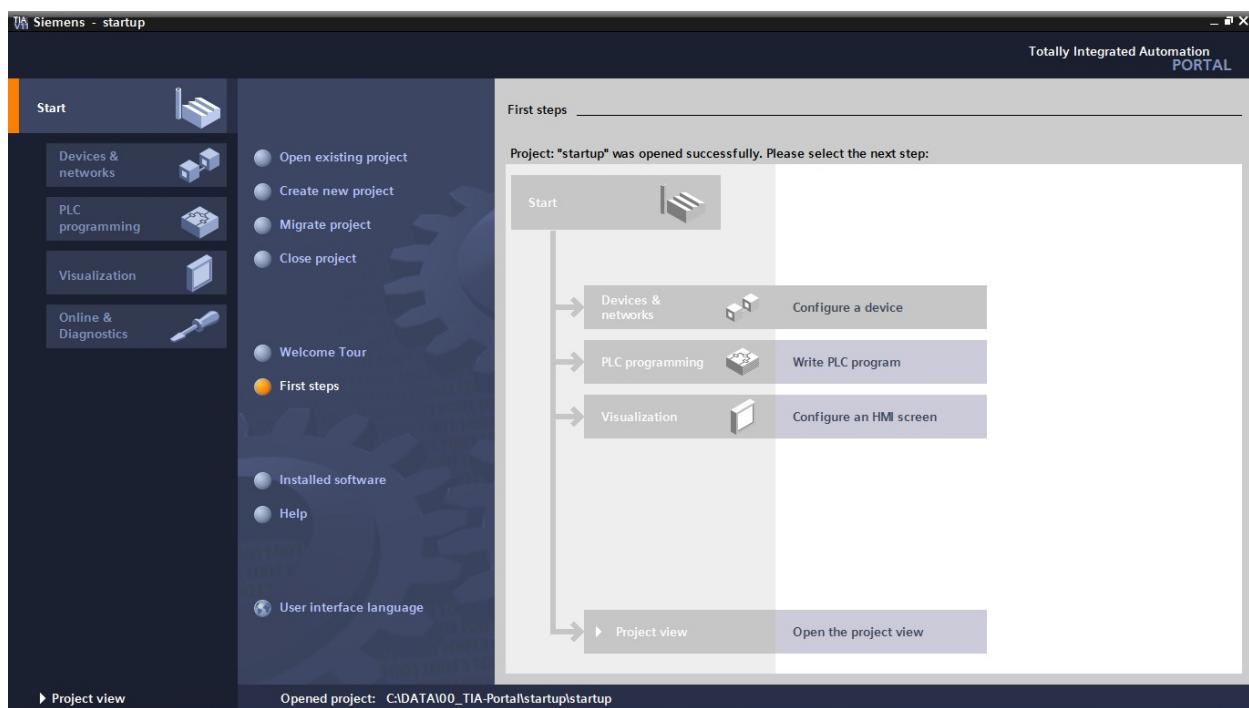
Qui si creano, si parametrizzano e si programmano con un'interfaccia utente unificata i componenti come controllore, visualizzazione e collegamento in rete della soluzione di automazione.

Per la diagnostica degli errori sono disponibili diversi tool online.

Il software '**Totally Integrated Automation Portal**' offre due viste diverse: la vista portale e la vista progetto.

### 8.1. Vista portale

La vista portale è una vista degli strumenti orientata alle attività per l'elaborazione del progetto. Da qui è possibile decidere rapidamente quali operazioni eseguire e quali strumenti richiamare per ogni diverso ordine. Se necessario, la vista commuta automaticamente alla vista progetto per il compito attualmente selezionato. Qui lo scopo principale è agevolare l'accesso e i primi passi.



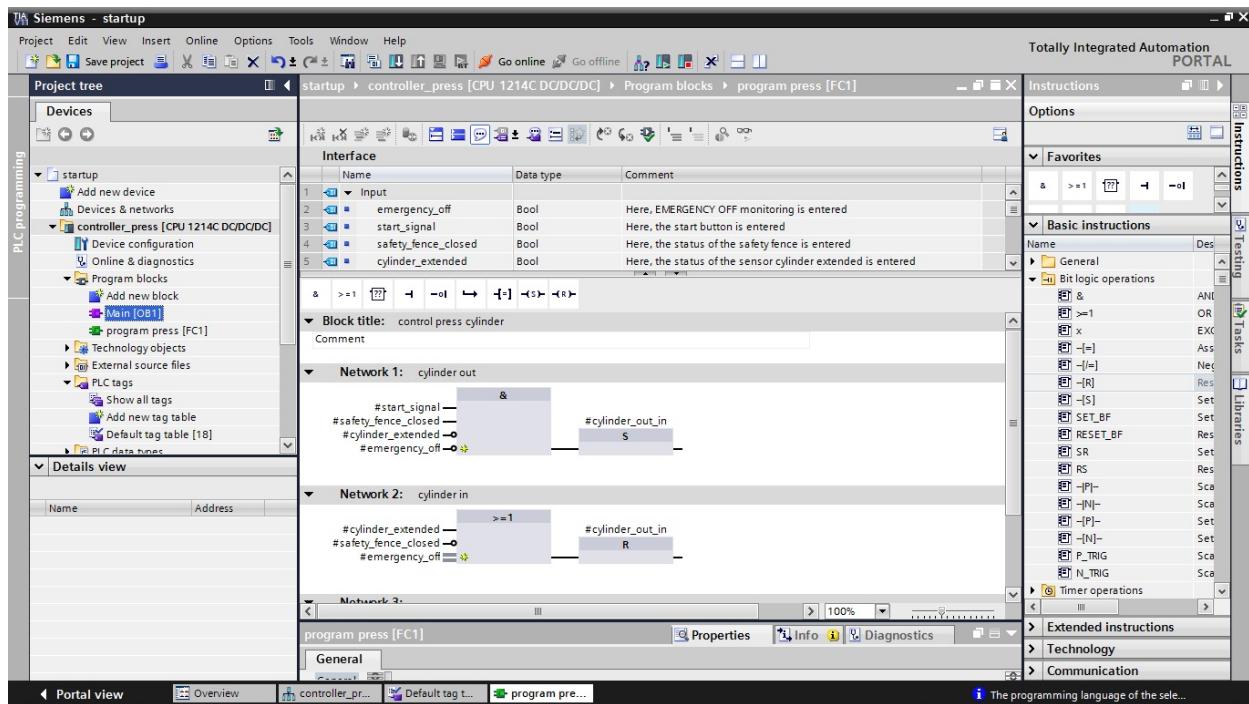
#### Avvertenza:

In basso a sinistra è possibile saltare direttamente dalla vista portale alla vista progetto.

## 8.2. Vista progetto

La vista progetto è una vista di tutti i componenti del progetto ordinati in una struttura. Per default la vista comprende la barra dei menu e le barre degli strumenti in alto, a sinistra la navigazione del progetto con tutti i componenti di un progetto e a destra le cosiddette 'task card' che contengono ad es. istruzioni e biblioteche.

Quando si seleziona un elemento nella navigazione del progetto (qui ad es. il blocco di programma FC1) esso viene visualizzato al centro e può essere elaborato da qui.



### Avvertenza:

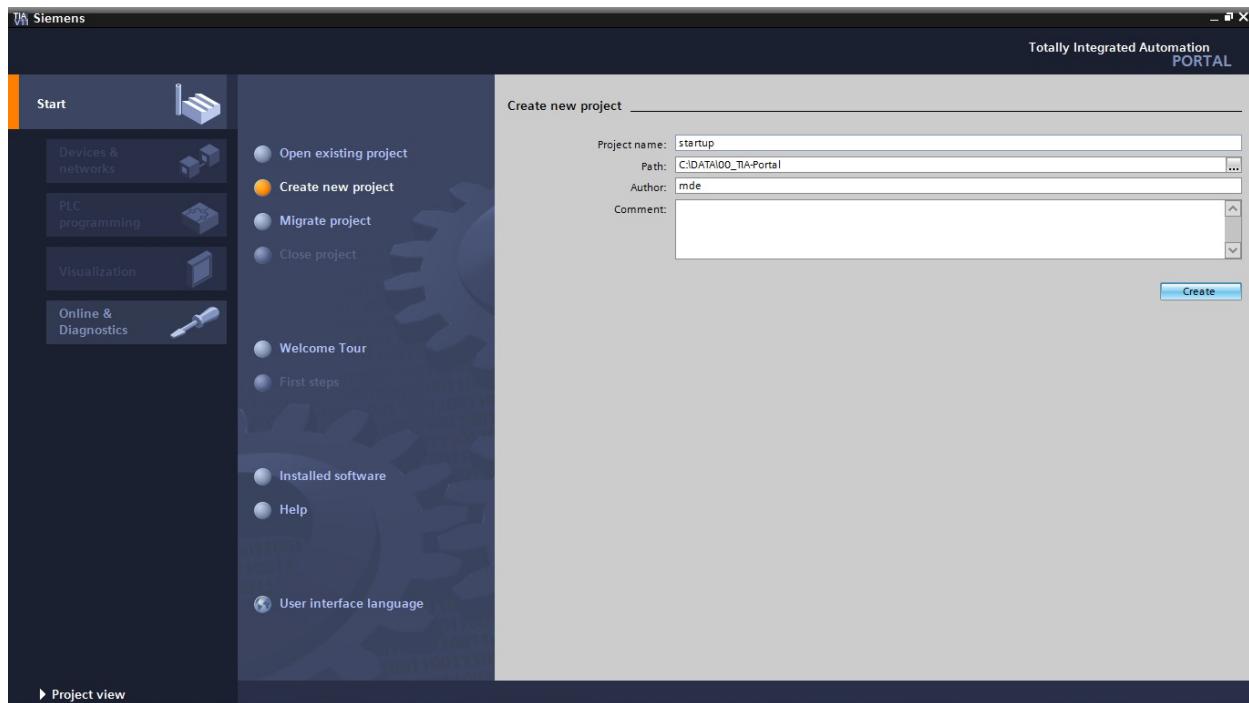
In basso a sinistra è possibile saltare direttamente dalla vista progetto alla vista portale.

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

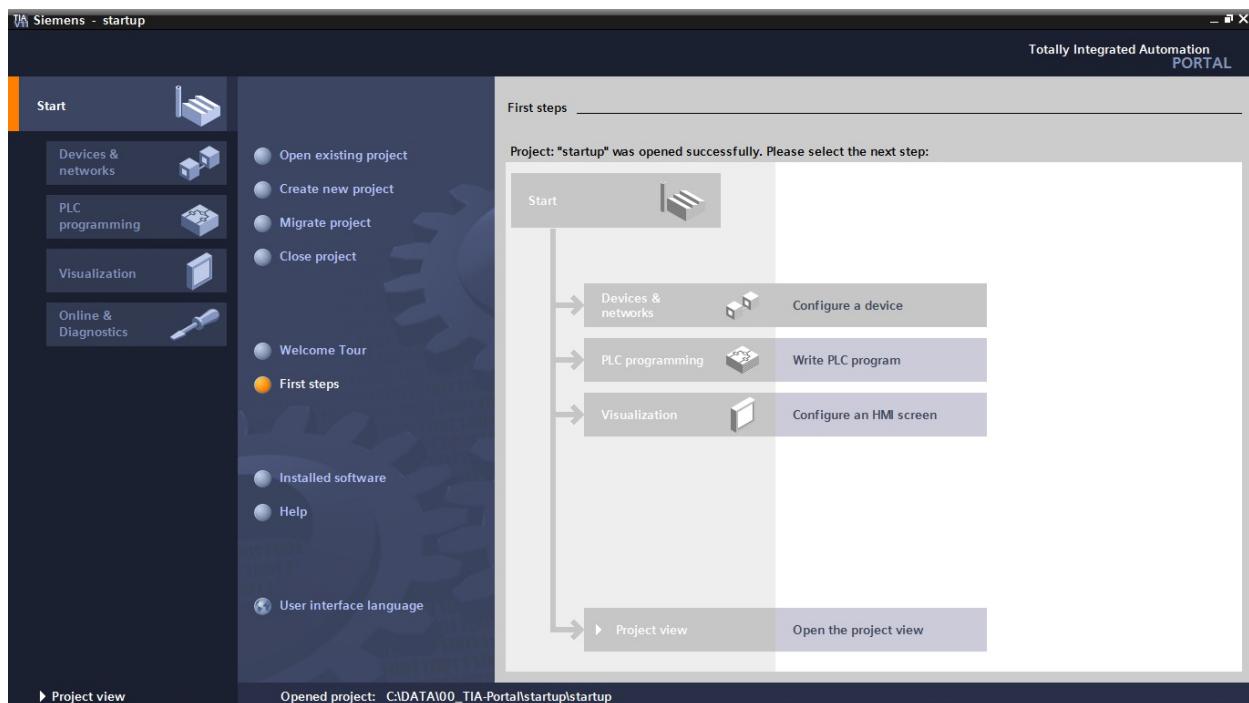
1. Il tool principale è '**Totally Integrated Automation Portal**', che si richiama qui con un doppio clic. ( → **Totally Integrated Automation Portal V11**)



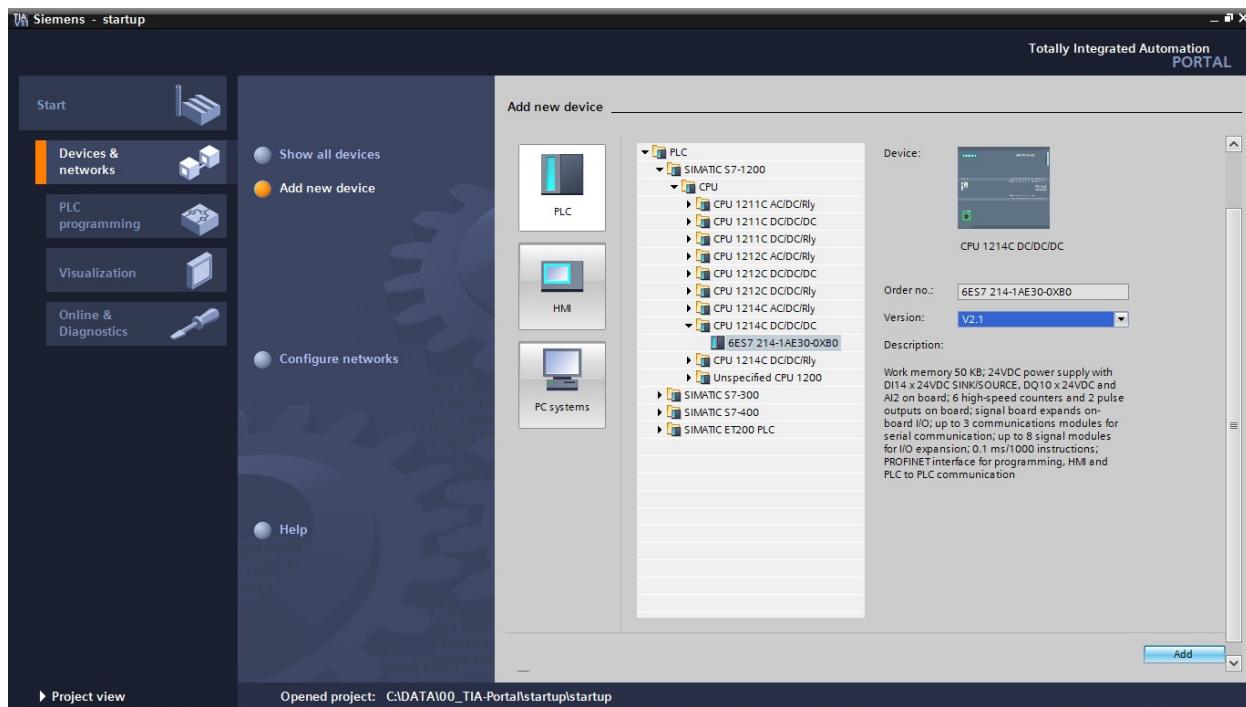
2. I programmi per il SIMATIC S7-1200 vengono gestiti in progetti. Uno di questi progetti verrà ora creato nella vista portale ( → **Create new project** (Crea nuovo progetto) → **startup** → **Create** (Crea))



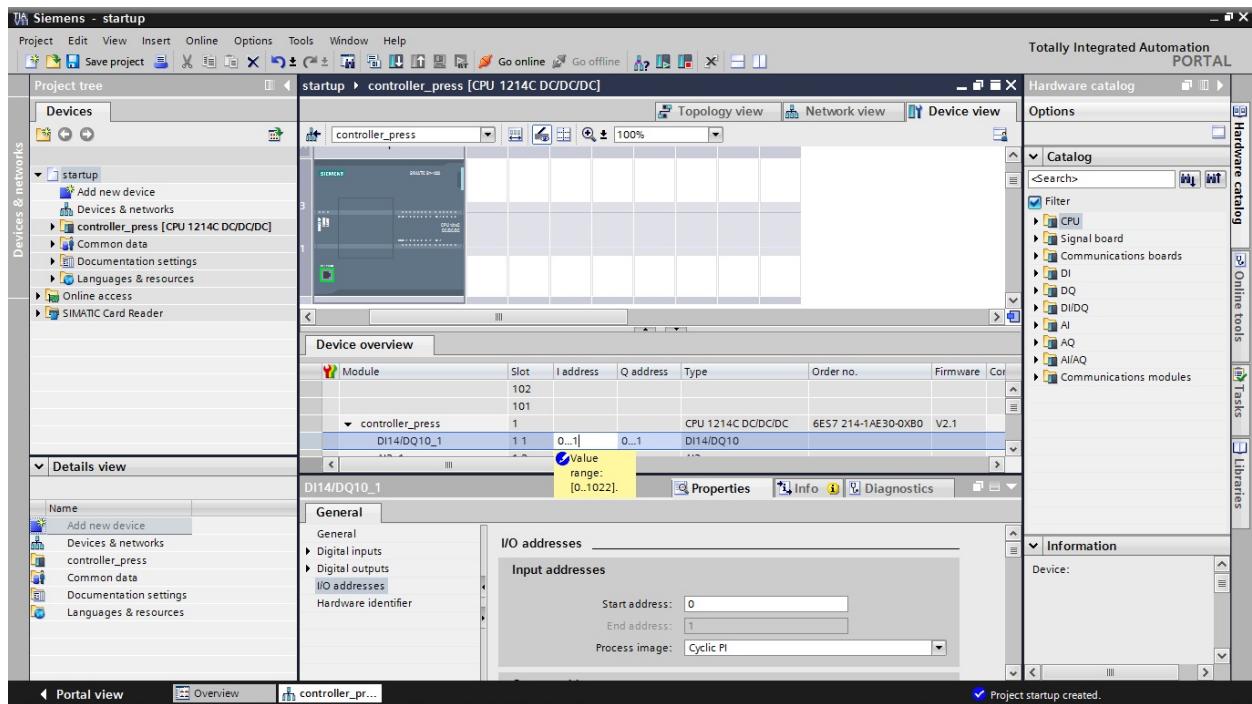
3. Ora alla voce '**First steps**' vengono proposti i primi passi per la progettazione. Innanzitutto vogliamo creare un dispositivo con '**Configure a device**'. (→ Primi passi → Configura un dispositivo)



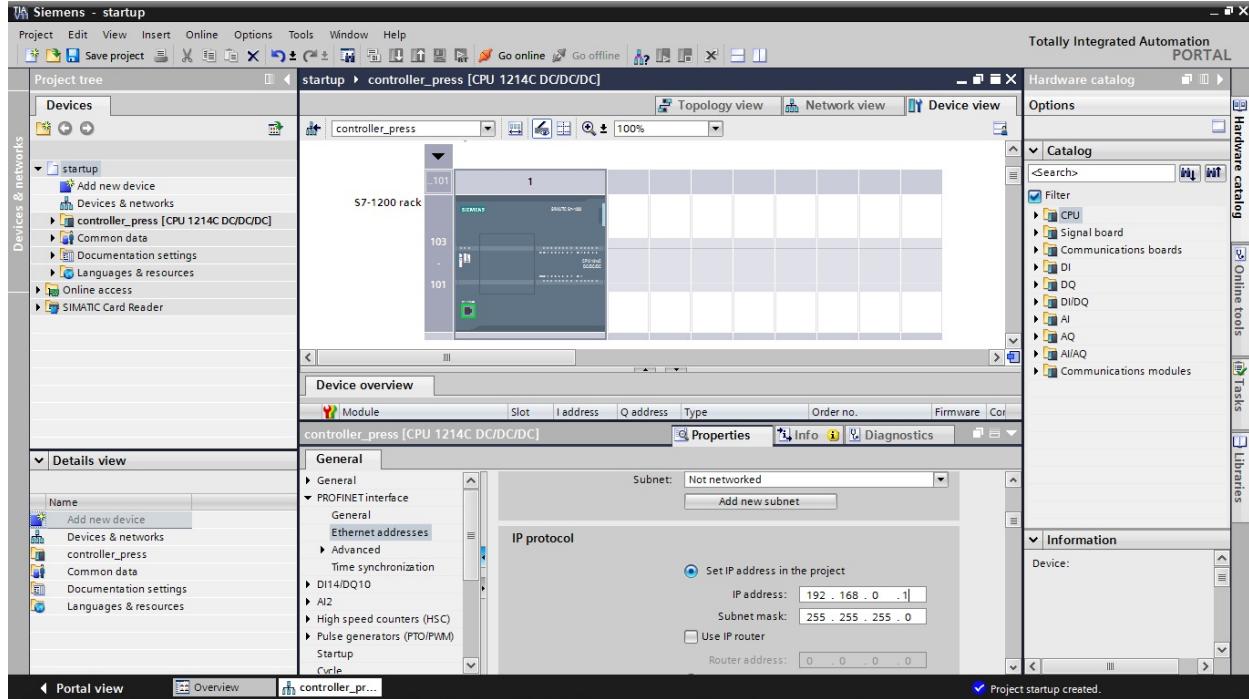
4. Successivamente inseriremo con 'Add new device' un nuovo dispositivo denominato '**controller\_press**'. Selezionare dal catalogo la '**CPU1214C**' con il numero di ordinazione adeguato. ( → Aggiungi nuovo dispositivo → Comando\_pressa → CPU1214C → 6ES7 ..... → Aggiungi)



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)  
(vedere anche: capitolo 3 sull'impostazione dell'interfaccia di programmazione.)



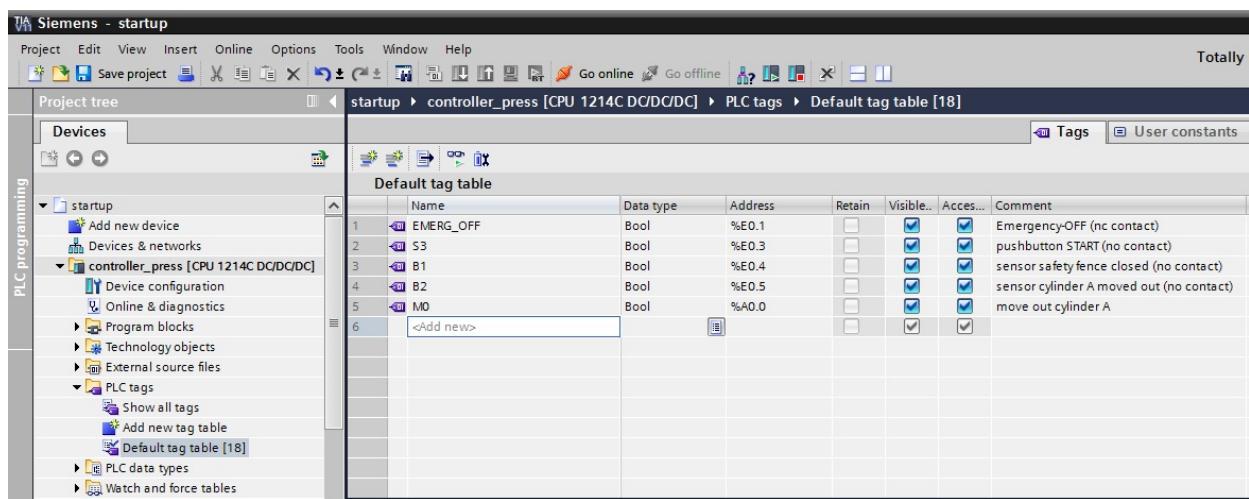
- Poiché nella moderna programmazione non si utilizzano gli indirizzi assoluti ma delle variabili, qui è necessario definire le **variabili PLC globali**.

Queste variabili PLC globali sono nomi descrittivi corredata 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\_press[CPU1214C DC/DC/DC]'** e poi **'PLC tags'**. Aprire la tabella **'Default tag table'** con un doppio clic e inserire i nomi degli ingressi e delle uscite come nella figura seguente.

( → Comando\_pressa[CPU1214C DC/DC/DC] → Variabili PLC → Tabella delle variabili standard)



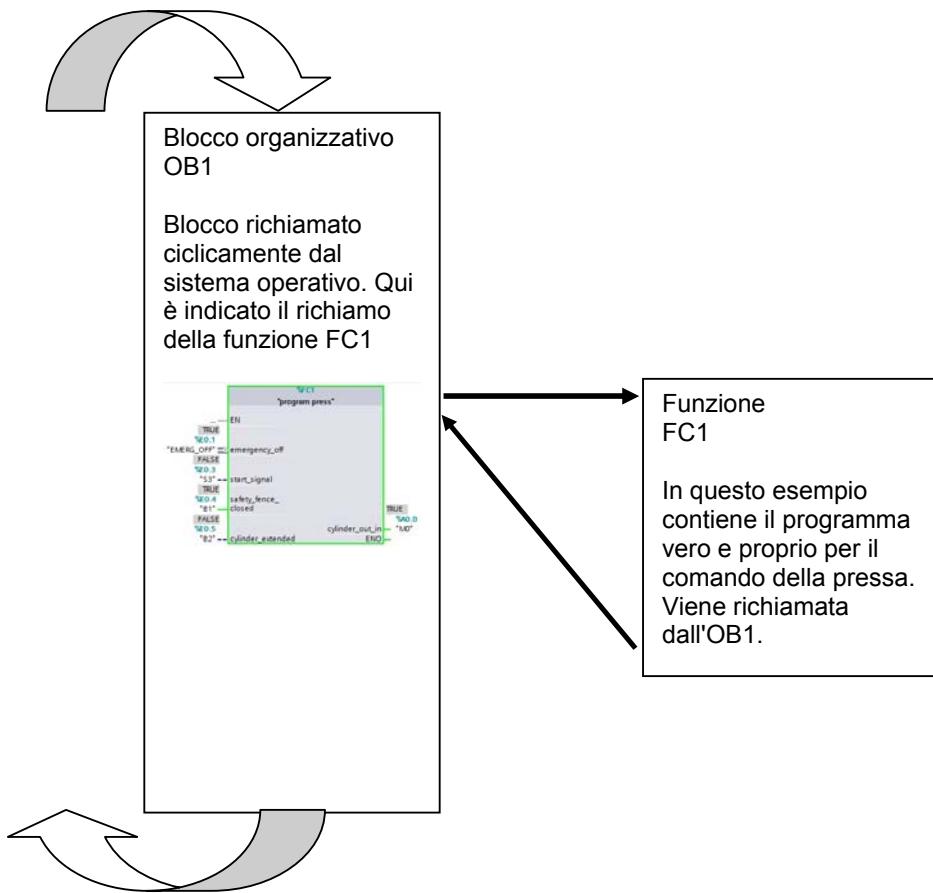
8. La sequenza del programma viene scritta in cosiddetti blocchi. Per default è già disponibile il blocco organizzativo OB1.

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

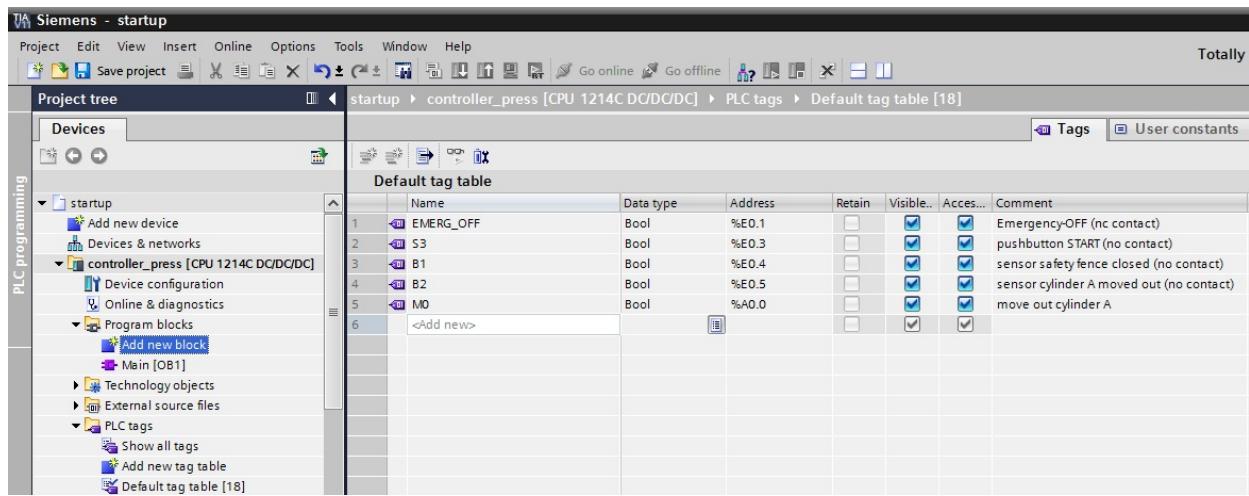
D'altra parte da questo blocco organizzativo è possibile richiamare ulteriori blocchi, come ad es. la funzione FC1, per strutturare la programmazione.

Lo scopo è quello di suddividere un compito complessivo in compiti più piccoli che sono più facili da risolvere e da testare nella loro funzionalità.

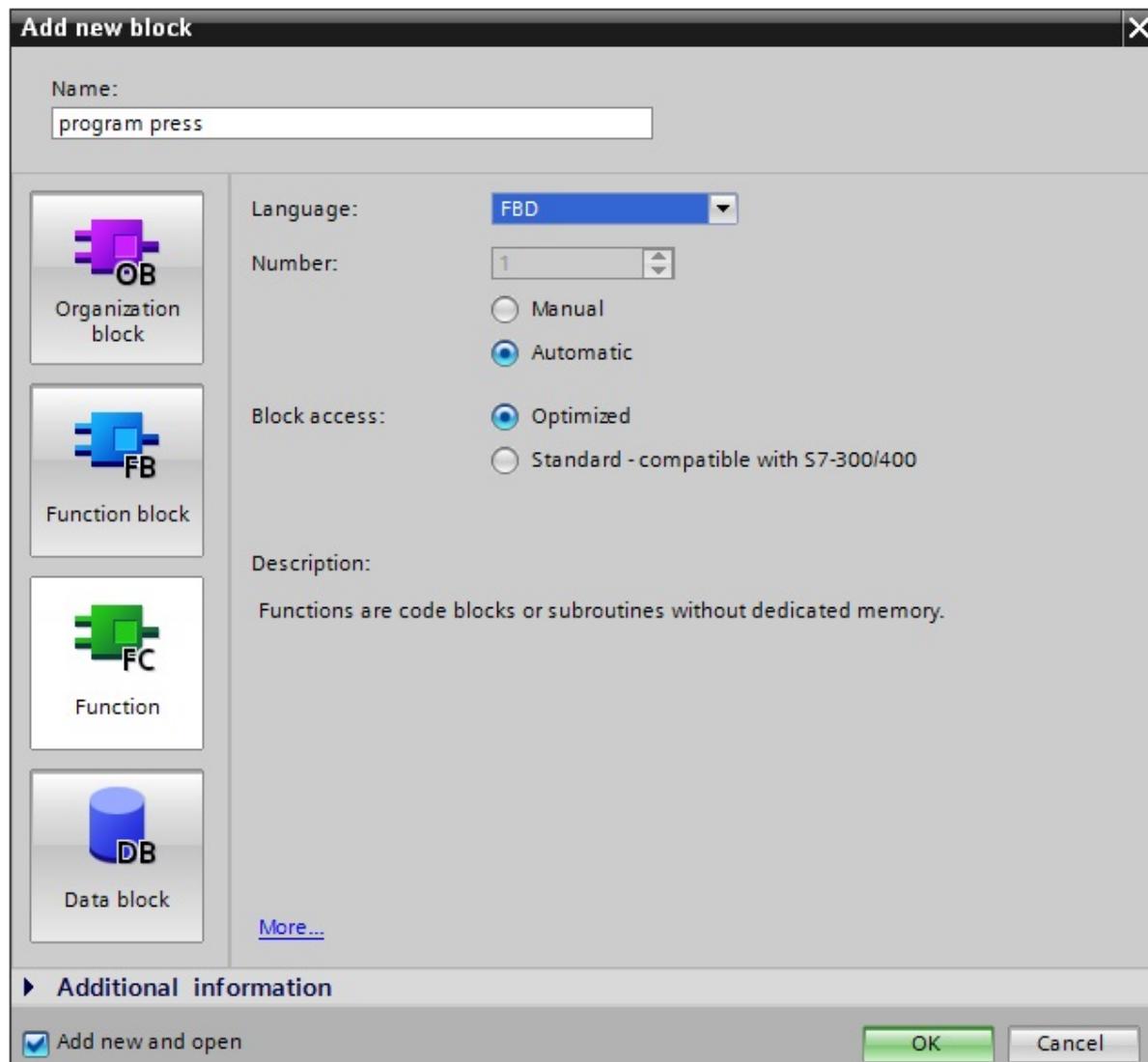
### Struttura del programma di esempio:



9. Per creare la funzione FC1 selezionare nella navigazione del progetto '**controller\_press**[CPU1214C DC/DC/DC]' e quindi '**Program blocks**'. Fare doppio clic su '**Add new block**'. ( → Comando\_pressa[CPU1214C DC/DC/DC] → Blocchi di programma → Inserisci nuovo blocco)



10. Selezionare 'Function (FC)' e assegnare il nome 'program press'. Come linguaggio di programmazione è preimpostato lo schema funzionale 'FBD'. La numerazione è automatica. Poiché questa FC1 in un secondo tempo verrà richiamata comunque con il nome simbolico, il numero non ha più una grande importanza. Applicare i dati inseriti con 'OK'. ( → Funzione (FC) → Programma pressa → FUP → OK)

**Avvertenza:**

Normalmente nel caso di una funzione con memoria andrebbe utilizzato un blocco funzionale con corrispondente blocco dati di istanza.  
Tuttavia, poiché nel nostro programma di esempio il salvataggio ha luogo al di fuori del blocco attraverso una variabile OUT e quindi nella PIQ, qui è possibile impiegare una funzione (FC).

11. Il blocco '**program press[FC1]**' 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

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

**Input:**

emergency_off	Qui viene immessa la funzione di arresto di emergenza
start_signal	Qui viene immesso il comando di avvio
safety_fence_closed	Qui viene interrogata la griglia di protezione chiusa
cylinder_extended	Qui viene interrogato il cilindro estratto

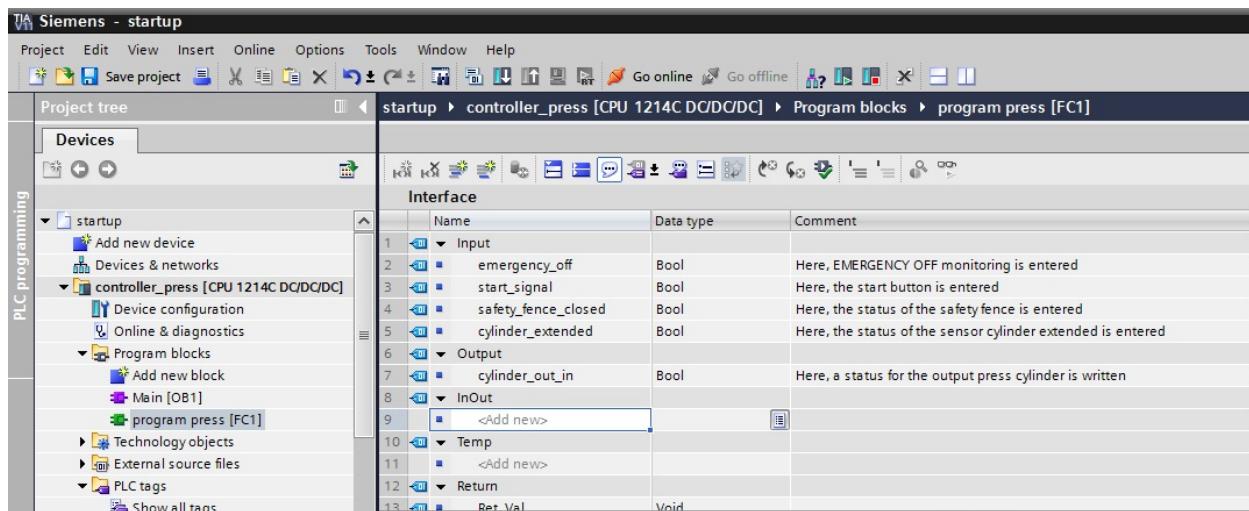
**Output:**

cylinder_out_in	Qui viene scritto uno stato per l'uscita del cilindro della pressa
-----------------	--

In questo caso tutte le variabili sono del tipo 'Bool', ovvero variabili binarie che possono avere solo lo stato

0' (false) o '1' (true).

Per una migliore comprensione tutte le variabili locali dovrebbero essere scritte a lettere minuscole ed essere corredate di un commento sufficientemente esplicativo.



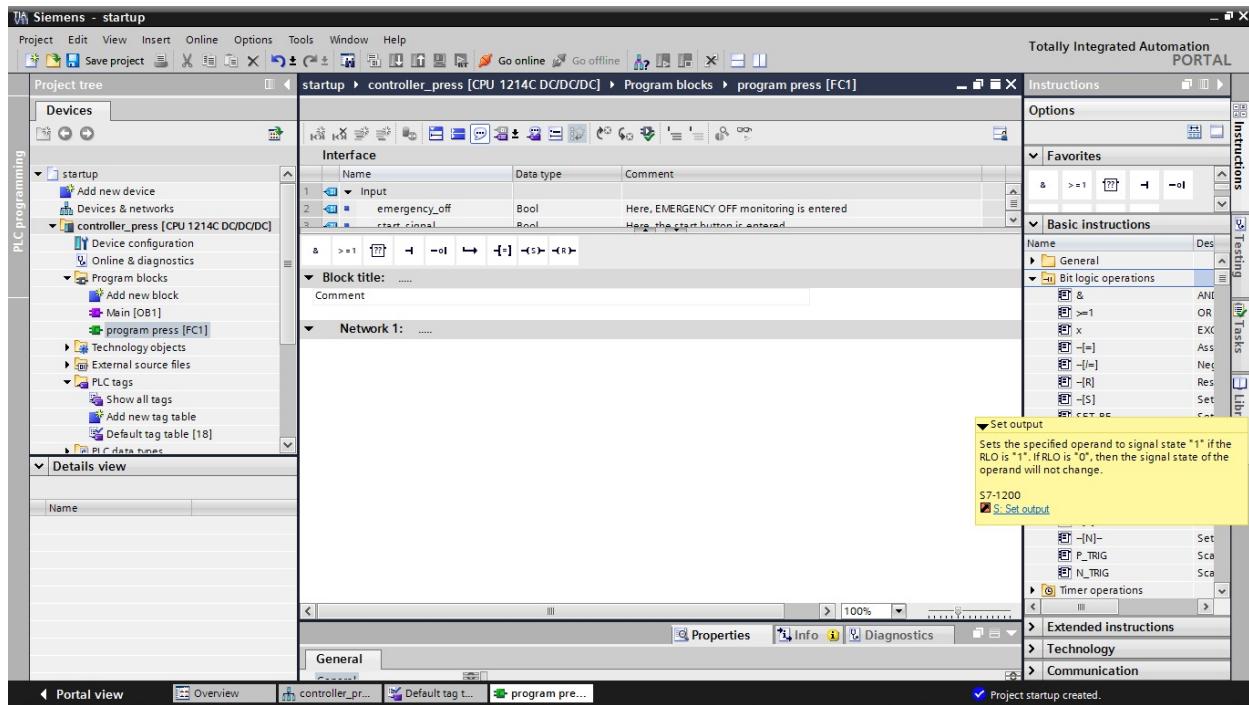
13. Dopo aver dichiarato le variabili locali è possibile iniziare la programmazione. Per garantire un migliore orientamento all'interno della programmazione vengono utilizzati i segmenti. Per inserire

un nuovo segmento è sufficiente fare clic con il mouse sul simbolo **'Insert network'** (**Inserisci segmento**). Ogni segmento - così come il blocco stesso - deve essere documentato almeno nella riga del titolo. Se per la descrizione è necessario un testo più lungo è possibile anche

utilizzare il campo **'Comment' (Commento)**. (→ 

Per creare la nostra soluzione ora è necessaria una **'S'** per impostare l'uscita. Questa si trova in **'Basic instructions'** nella cartella **'Bit logic operations'**. Puntando il mouse su un oggetto, come ad es. **S**, viene visualizzata un'informazione dettagliata sull'oggetto.

(→ Istruzioni di base → Combinazioni logiche di bit → S)



14. Evidenziando un oggetto e premendo il tasto '**F1**' sulla tastiera del PC è possibile visualizzare una finestra con la Guida in linea di questo oggetto. ( → F1)

### S: Set output



#### Description

You can use the "Set output" instruction to reset the signal state of a specified operand to "1".

The instruction is only executed if the result of logic operation (RLO) at the box input is "1". If the box input has the signal state "1", the specified operand is set to "1". If there is an RLO of "0" at the box input, the signal state of the specified operand remains unchanged.

Executing the instruction does not influence the RLO. The RLO at the box input is transferred directly to the box output.

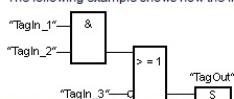
#### Parameters

The following table shows the parameters of the "Set output" instruction:

Parameters	Declaration	Data type	Memory area	Description
<Operand>	Output	BOOL	I, Q, M, D, L	Operand that is set when RLO = "1".

#### Example

The following example shows how the instruction works:



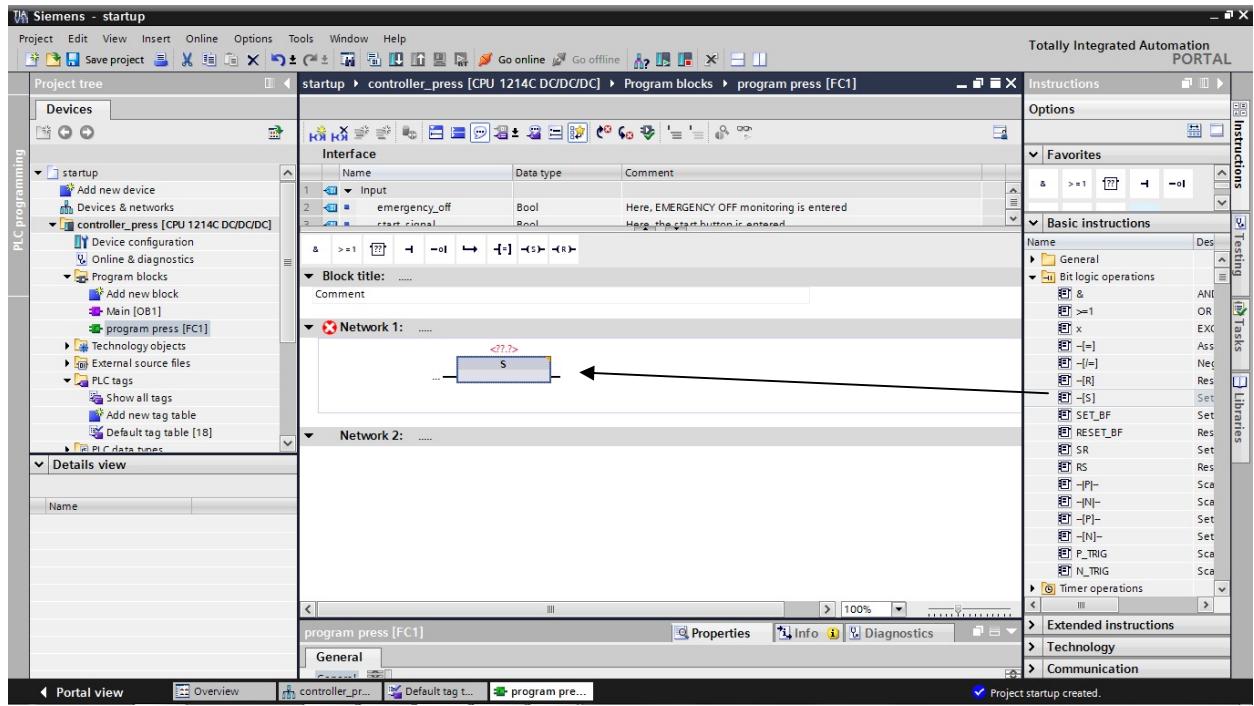
The operand "TagOut" is reset when one of the following conditions is fulfilled:

- The operands "TagIn\_1" and "TagIn\_2" have signal state "1".
- The operand "TagIn\_3" has the signal state "0".

### Avvertenza:

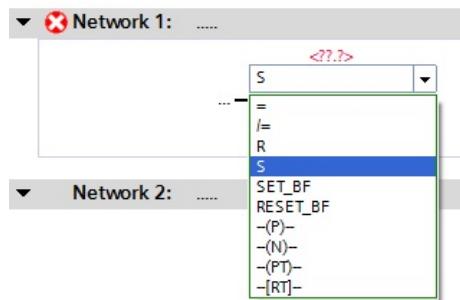
Leggere attentamente nella Guida in linea le informazioni sulla funzione e il collegamento dell'istruzione **S** (Imposta uscita).

15. Ora trascinare la **S** con il mouse sotto il commento nel segmento 1. ( → S)

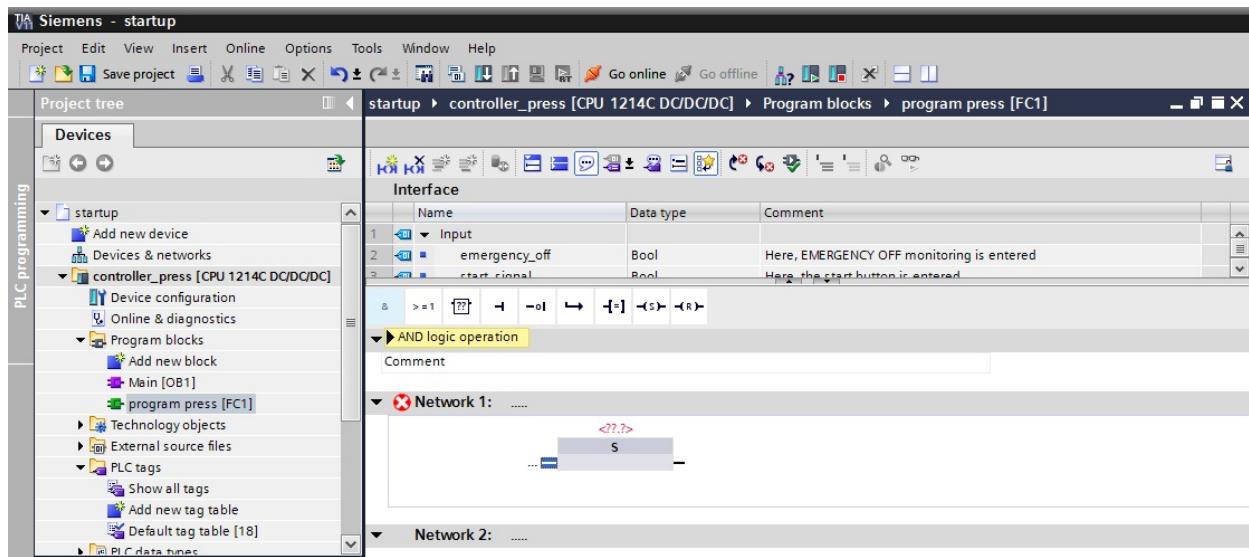


#### Avvertenza:

Facendo clic sull'angolo arancione in alto a destra è possibile modificare la funzione del blocco anche a posteriori utilizzando il menu.

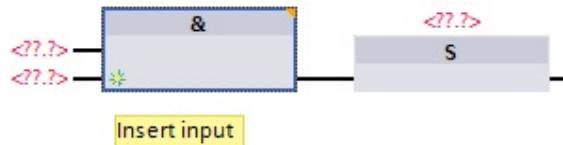


16. Successivamente evidenziare l'ingresso del blocco S a sinistra e fare doppio clic su  AND nei Preferiti. ( → Ingresso a sinistra → Preferiti →  AND)



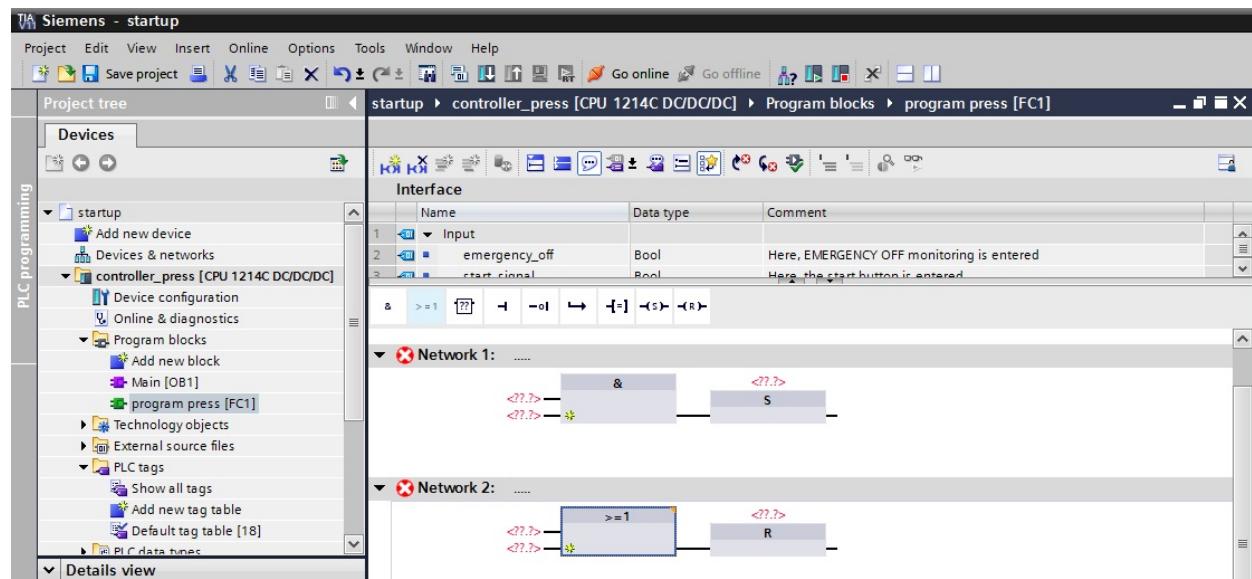
**Avvertenza:**

Facendo clic sull'asterisco giallo in basso a sinistra si possono aggiungere ulteriori ingressi.  
Facendo clic sull'angolo arancione in alto a destra è possibile modificare la funzione del blocco anche qui utilizzando il menu.

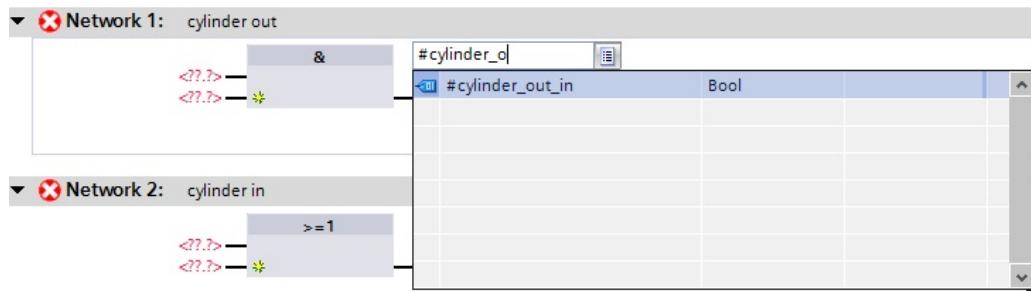


17. allo stesso modo posizionare sotto **Network 2 (Segmento 2)** la combinazione  OR per R  
'Resetta uscita'

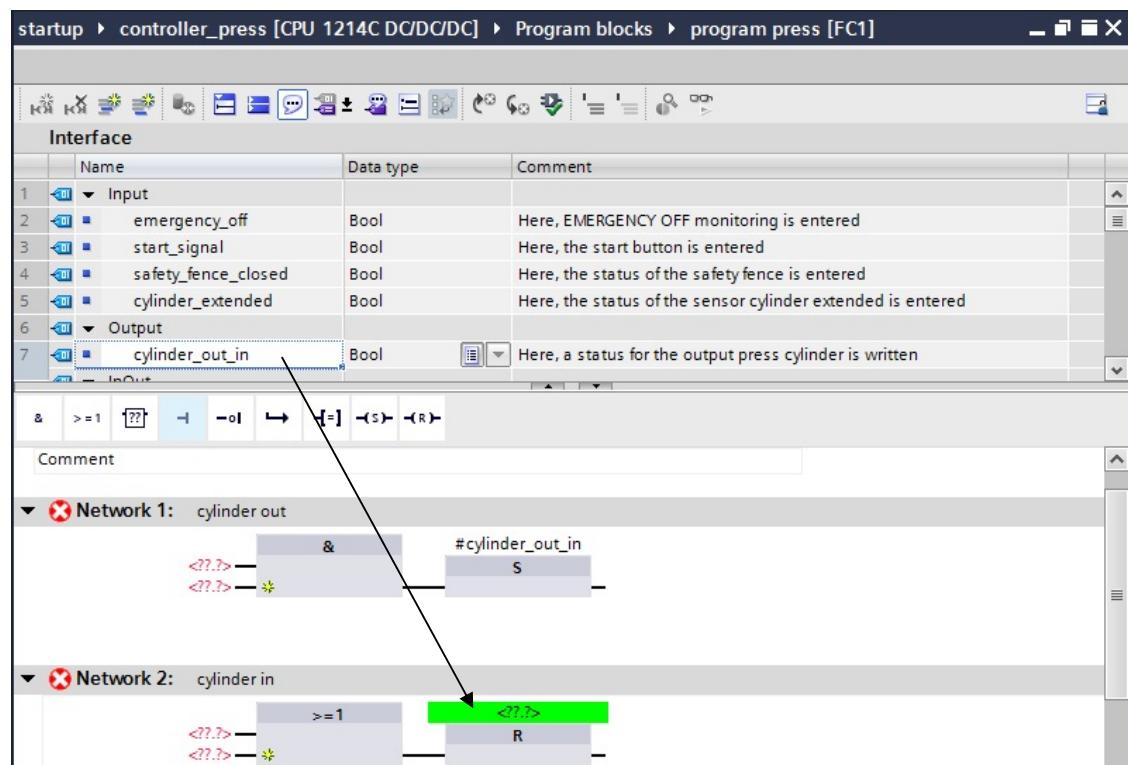
( → R →  OR )



18. A questo punto inserire le variabili locali. È sufficiente inserire la prima lettera della variabile locale nei campi in corrispondenza dei comandi. Sarà poi possibile scegliere la variabile desiderata da una casella di riepilogo. Le variabili locali sono sempre identificate dal simbolo '#' davanti al nome. ( → #cylinder\_out\_in ).



Le variabili locali si possono anche trascinare dall'interfaccia.

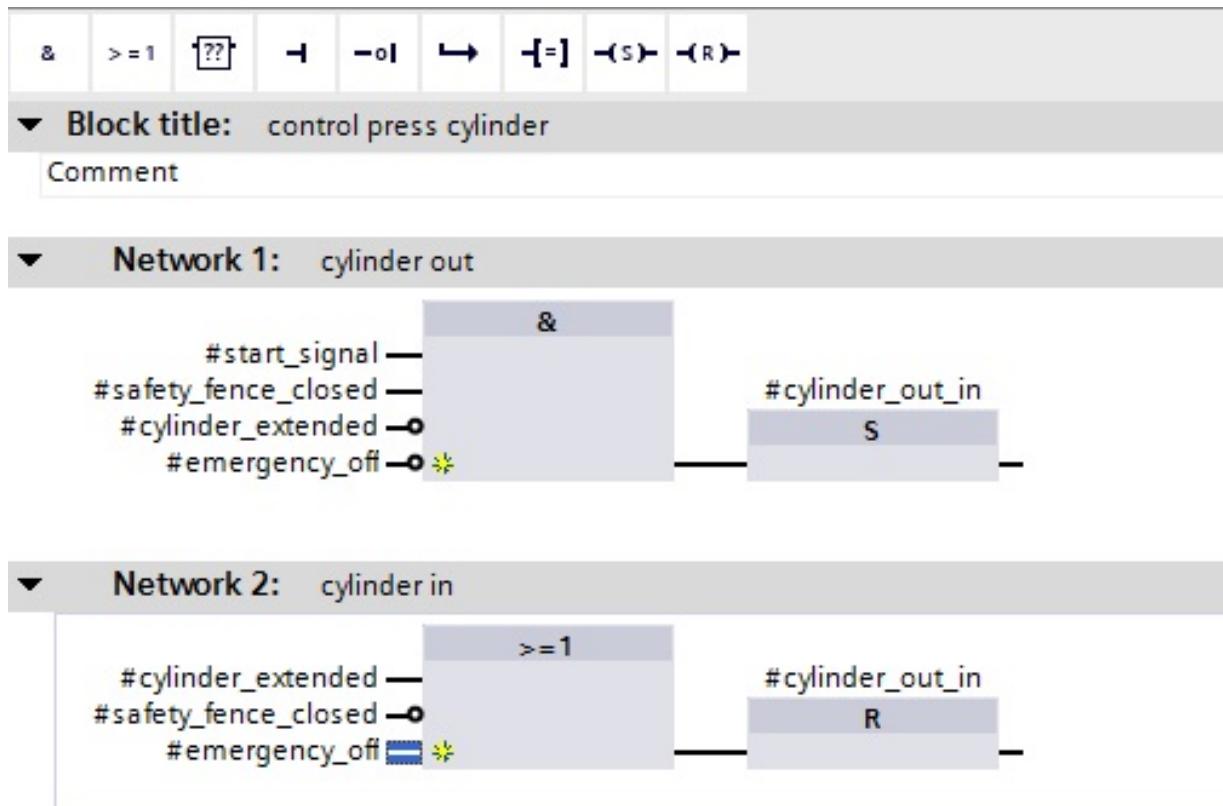


19. Seguendo lo stesso procedimento aggiungere le altre variabili locali. Nei blocchi AND e OR devono essere inseriti ulteriori ingressi. Fare clic sull'asterisco giallo o selezionare l'ultimo ingresso in basso con il tasto destro del mouse e selezionare '**Insert input**'. (→ Inserisci ingresso)

20. Per invertire l'ingresso selezionarlo e fare clic su

o trascinare il simbolo

della negazione da '**Favorites**' (**Preferiti**) sull'ingresso

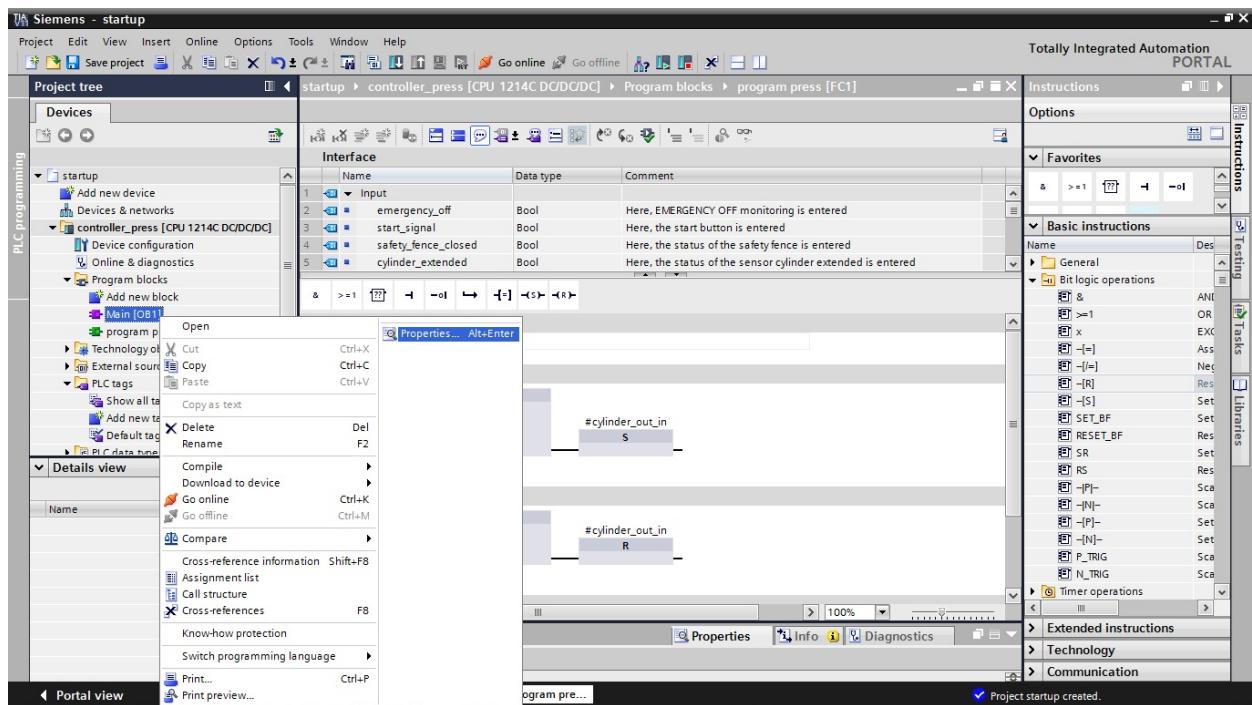


#### Descrizione della funzione del blocco FC

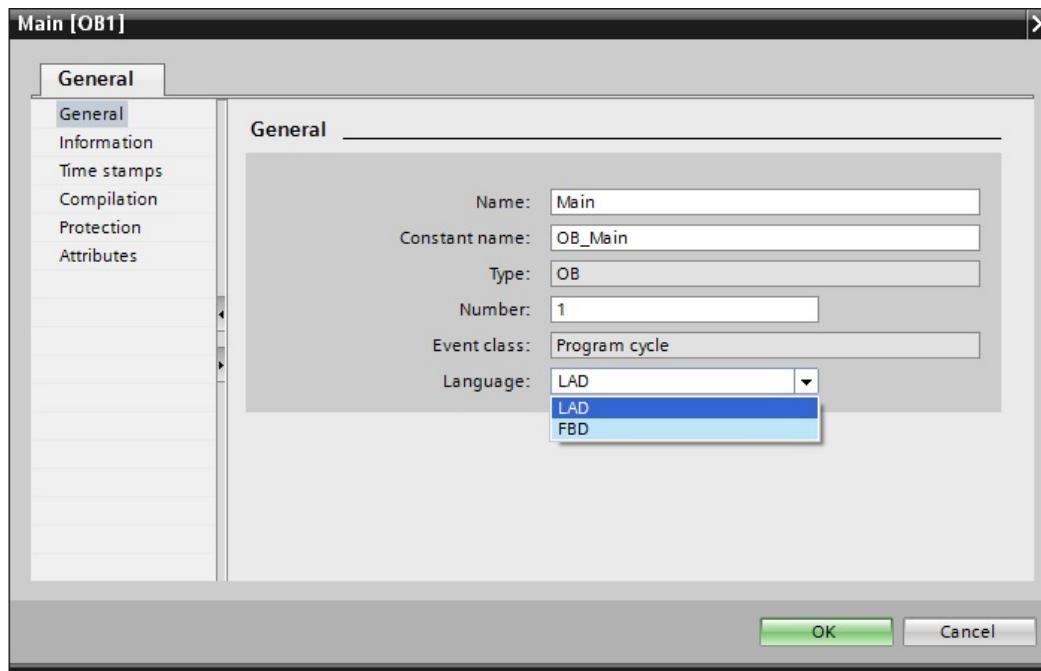
Il cilindro deve essere estratto quando viene dato il comando di avvio **e** la griglia di protezione è chiusa **e non** se il cilindro è già estratto **e non** se è attivo un arresto di emergenza.

Il cilindro deve essere inserito nuovamente quando è estratto **o** quando la griglia di protezione **non** è più chiusa **o** se è attivo un arresto di emergenza.

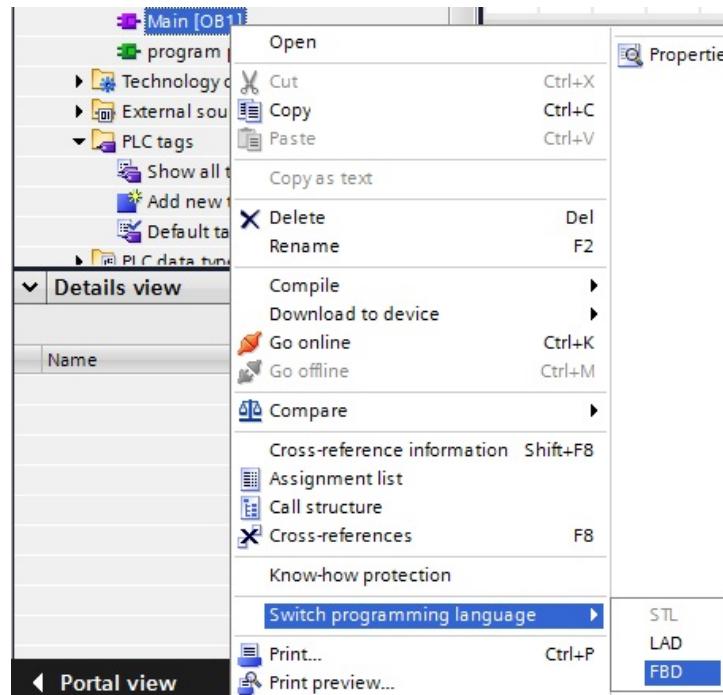
21. Successivamente occorre selezionare le proprietà 'Properties' del blocco 'Main[OB1]' elaborato ciclicamente. Le proprietà dei blocchi sono modificabili. ( → Proprietà → Main[OB1] )



22. Selezionare nelle proprietà il linguaggio di programmazione ('**Language**') schema funzionale '**FBD**'.  
( → FUP → OK)

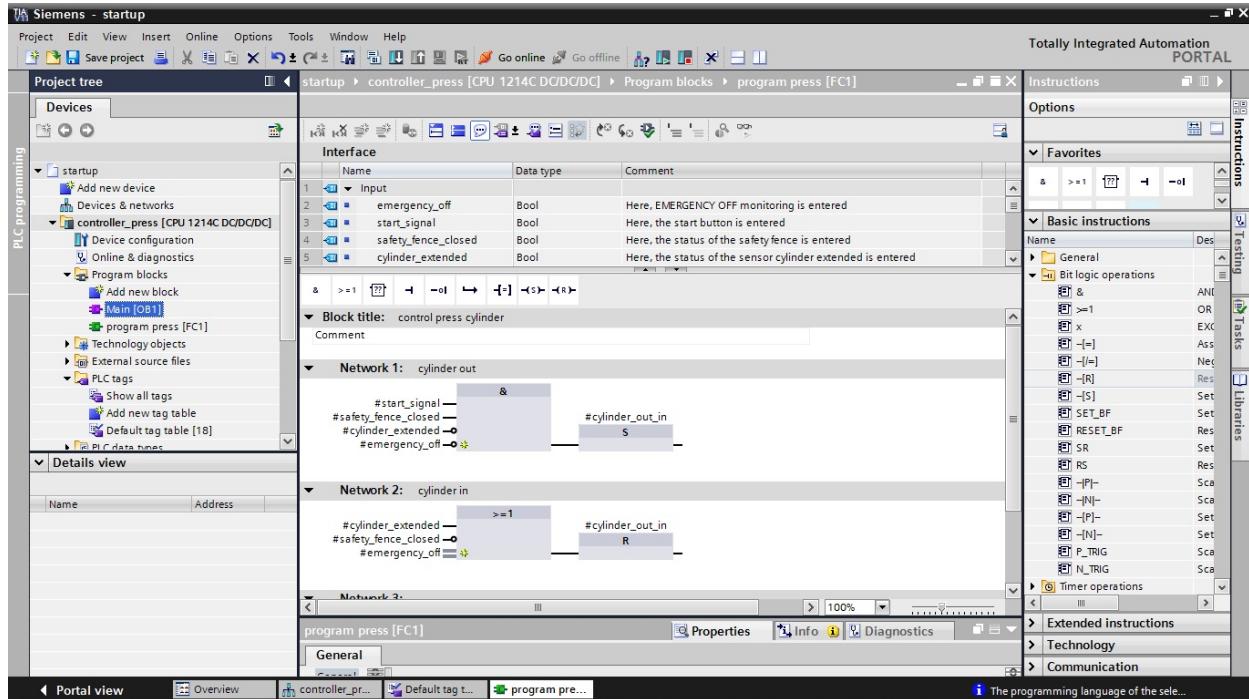


In alternativa è possibile scegliere '**Switch programming language**' (Commuta linguaggio di programmazione).

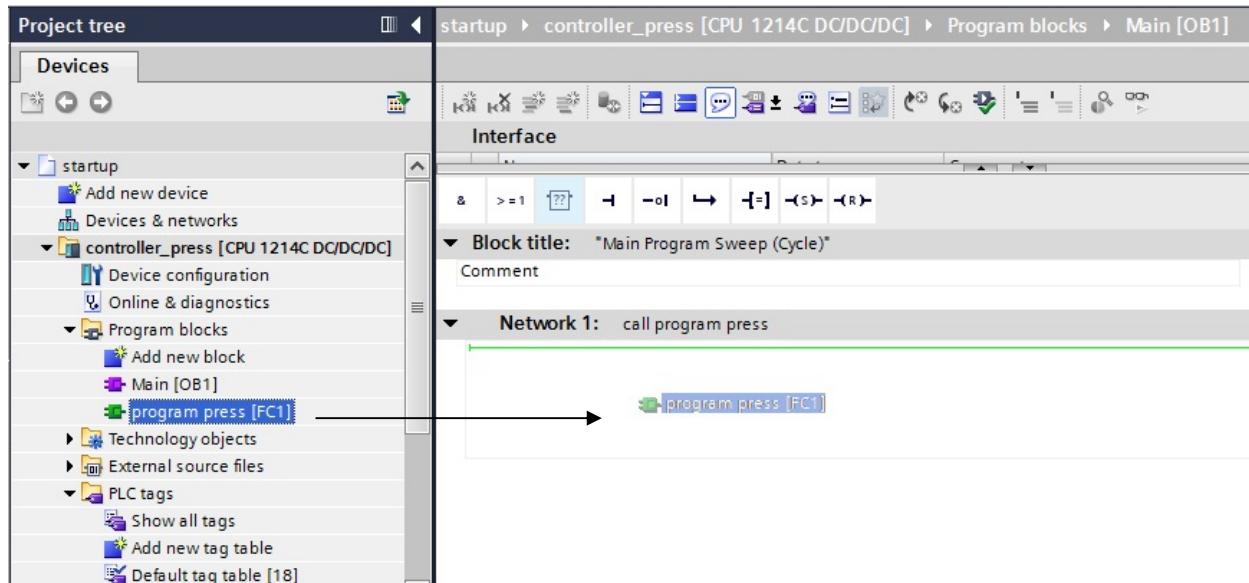


23. Come accennato in precedenza è necessario richiamare il blocco "program press" (Programma pressa) dal blocco di programma Main[OB1]. In caso contrario il blocco non verrebbe proprio elaborato.

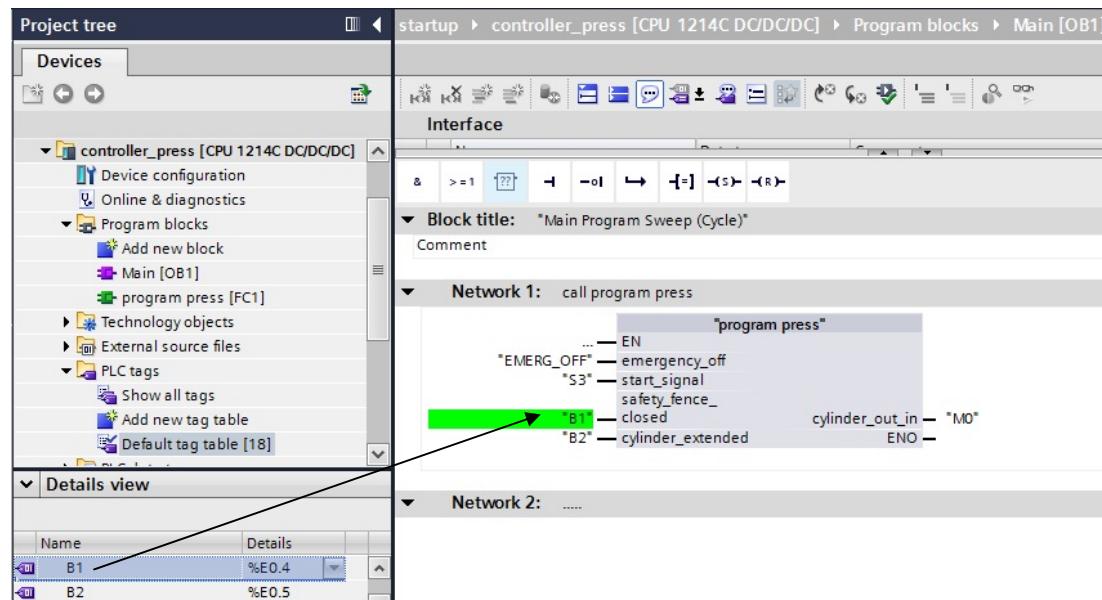
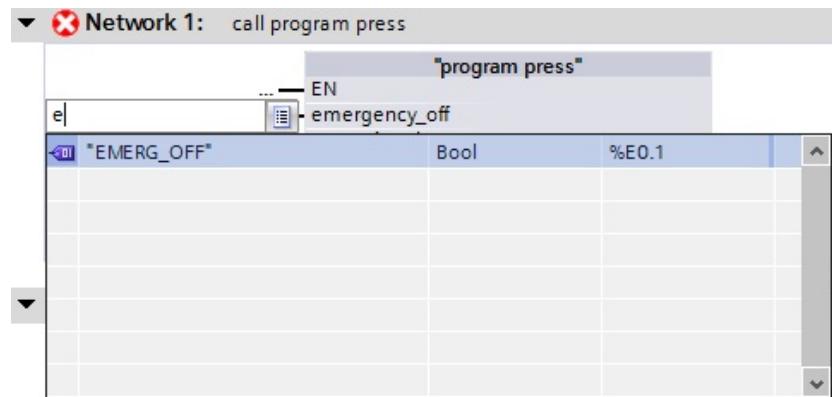
Aprire il blocco facendo un doppio clic su '**Main[OB1]**'. ( → Main[OB1] )



24. Il blocco "program press" può essere semplicemente trascinato con il mouse nel segmento 1 del blocco Main[OB1]. Non dimenticare di documentare i segmenti anche nel blocco Main[OB1]. (→ Programma pressa)

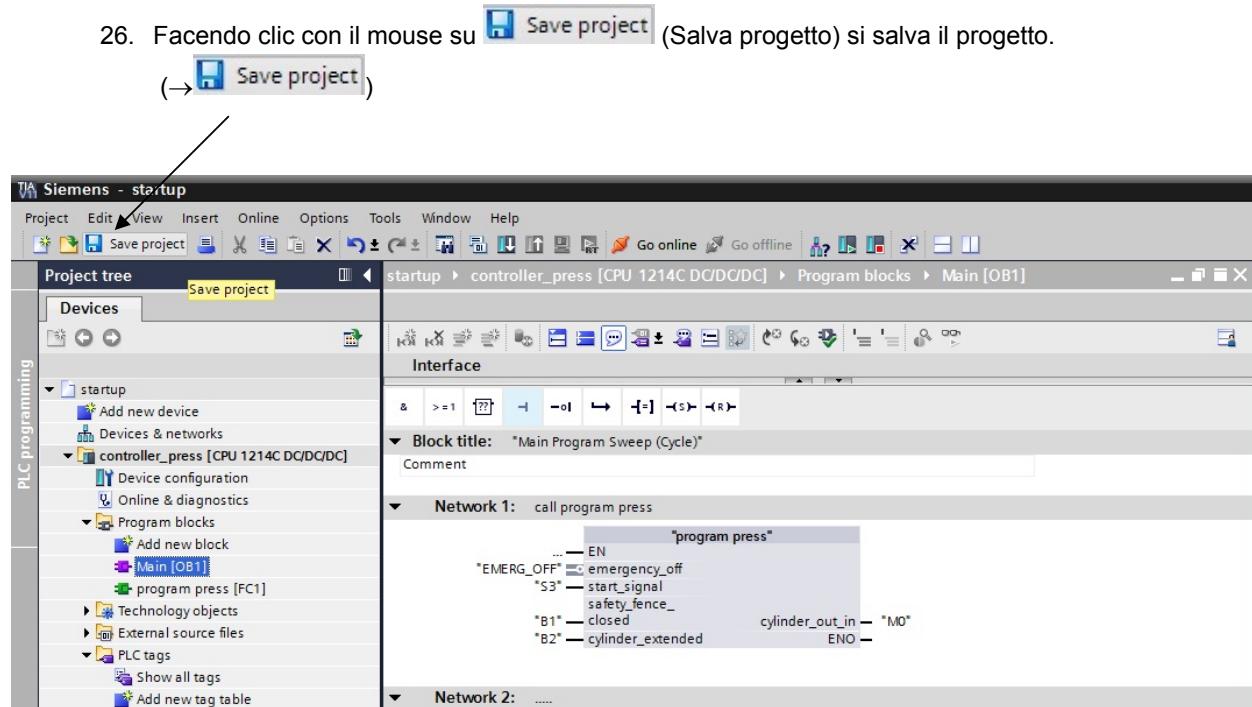


25. I parametri dell'interfaccia del blocco "program press" ora devono essere collegati alle variabili PLC globali. È sufficiente inserire la prima lettera e selezionare l'operando desiderato dall'elenco. ( → "EMERG\_OFF" ) oppure selezionare la tabella delle variabili standard e trascinare le variabili dalla vista dettagli nel blocco.

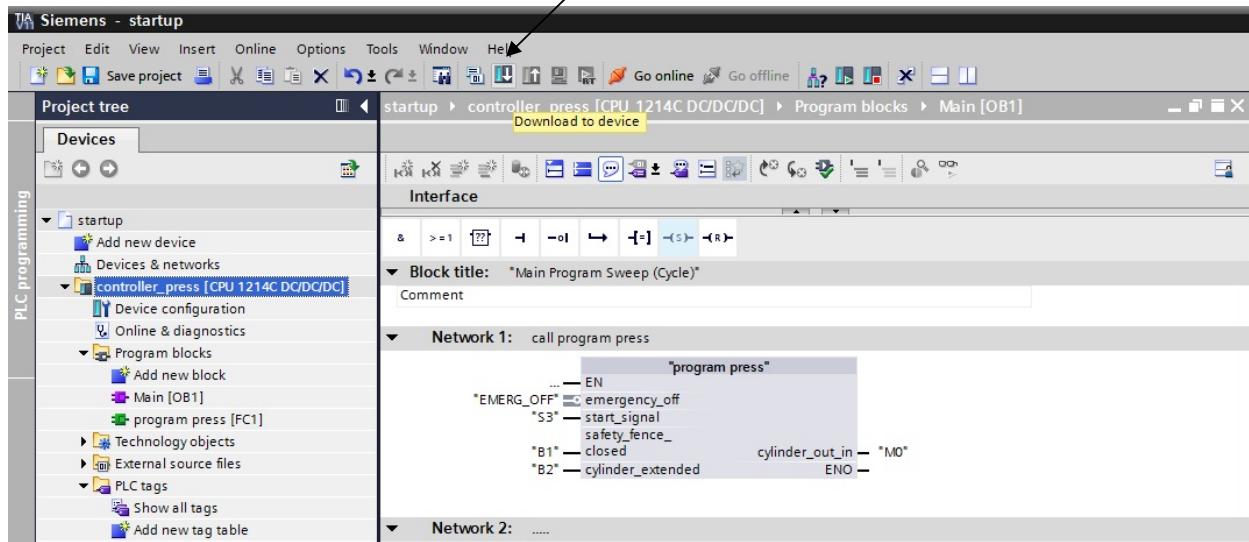


### Avvertenza

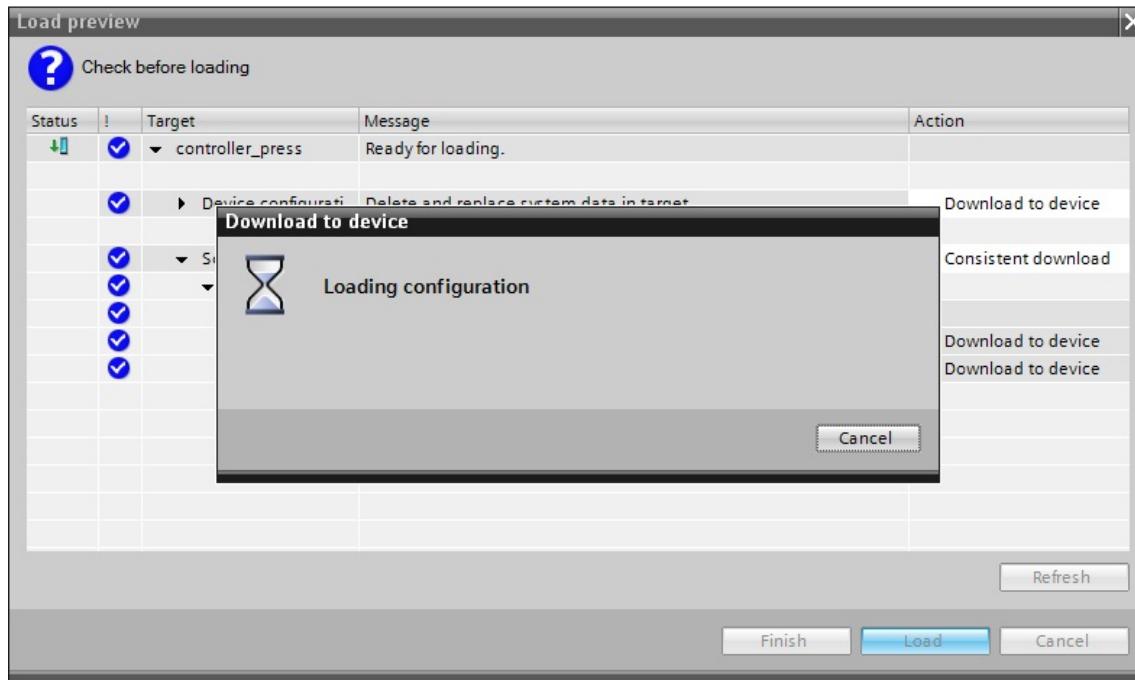
L'ARRESTO DI EMERGENZA è un contatto normalmente chiuso (NC) e pertanto deve essere negato; in altri termini, se si attiva l'arresto di emergenza, nell'ingresso I0.1 non è più presente il segnale. Nel connettere i blocchi con gli indirizzi reali occorre prestare attenzione ai sensori (NO, NC).



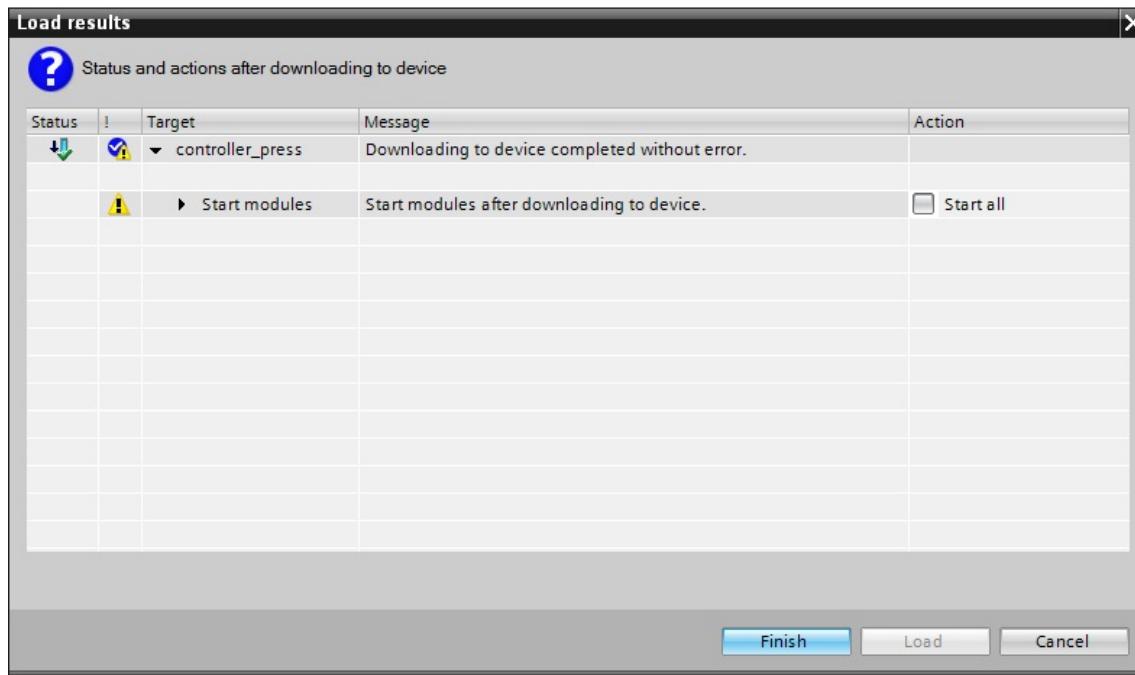
27. Per caricare l'intero programma nella CPU selezionare prima la cartella '**controller\_press**' quindi fare clic sul simbolo  (Carica nel dispositivo). (→ Comando\_pressa → 



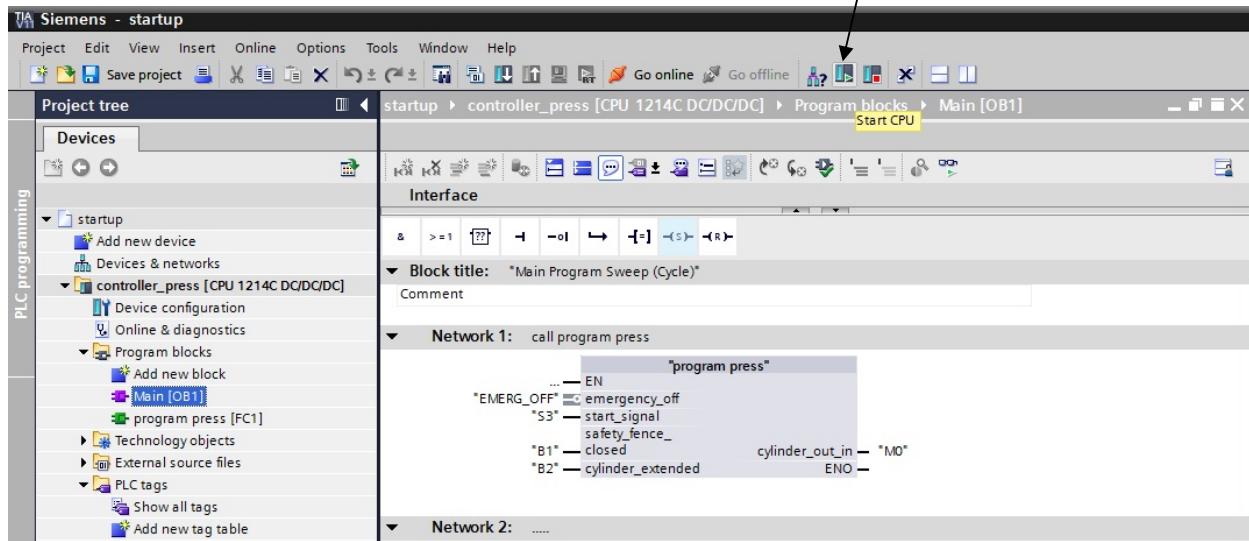
28. Durante il caricamento lo stato viene visualizzato in una finestra.



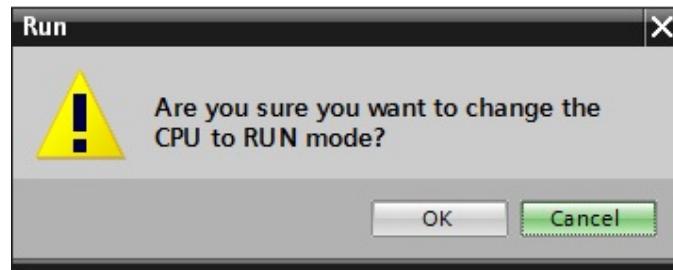
29. Il caricamento completato senza errori viene visualizzato in una finestra. Fare clic con il mouse su 'Finish'. ( → Fine)



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



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



32. Con un clic del mouse sul simbolo  Monitoring on/off (Controllo on/off) è possibile controllare lo stato delle variabili di ingresso e di uscita nel blocco "program press" durante il test del programma. ( →  )

