

SIEMENS

SIMATIC

Primi passi ed esercitazioni con STEP 7

Getting Started

Introduzione a STEP 7	1
SIMATIC Manager	2
Programmazione con nomi simbolici	3
Creazione di un programma nell' OB1	4
Creazione di un programma con FB e DB	5
Configurazione delle unità centrali	6
Caricamento e test del programma	7
Programmazione di una funzione (FC)	8
Programmazione di un blocco dati globali	9
Programmazione di una multiistanza	10
Configurazione della periferia decentrata	11
Appendice A	A

Il presente manuale fa parte del pacchetto di documentazione con il numero di ordinazione:
6ES7810-4CA10-8EW0

Avvertenze di legge

Concetto di segnaletica di avvertimento

Questo manuale contiene delle norme di sicurezza che devono essere rispettate per salvaguardare l'incolumità personale e per evitare danni materiali. Le indicazioni da rispettare per garantire la sicurezza personale sono evidenziate da un simbolo a forma di triangolo mentre quelle per evitare danni materiali non sono precedute dal triangolo. Gli avvisi di pericolo sono rappresentati come segue e segnalano in ordine decrescente i diversi livelli di rischio.

 PERICOLO

questo simbolo indica che la mancata osservanza delle opportune misure di sicurezza provoca la morte o gravi lesioni fisiche.
--

 AVVERTENZA

il simbolo indica che la mancata osservanza delle relative misure di sicurezza può causare la morte o gravi lesioni fisiche.

 CAUTELA
--

con il triangolo di pericolo indica che la mancata osservanza delle relative misure di sicurezza può causare lesioni fisiche non gravi.

CAUTELA

senza triangolo di pericolo indica che la mancata osservanza delle relative misure di sicurezza può causare danni materiali.
--

ATTENZIONE

indica che, se non vengono rispettate le relative misure di sicurezza, possono subentrare condizioni o conseguenze indesiderate.
--

Nel caso in cui ci siano più livelli di rischio l'avviso di pericolo segnala sempre quello più elevato. Se in un avviso di pericolo si richiama l'attenzione con il triangolo sul rischio di lesioni alle persone, può anche essere contemporaneamente segnalato il rischio di possibili danni materiali.

Personale qualificato

Il prodotto/sistema oggetto di questa documentazione può essere adoperato solo da **personale qualificato** per il rispettivo compito assegnato nel rispetto della documentazione relativa al compito, specialmente delle avvertenze di sicurezza e delle precauzioni in essa contenute. Il personale qualificato, in virtù della sua formazione ed esperienza, è in grado di riconoscere i rischi legati all'impiego di questi prodotti/sistemi e di evitare possibili pericoli.

Uso conforme alle prescrizioni di prodotti Siemens

Si prega di tener presente quanto segue:

 AVVERTENZA

I prodotti Siemens devono essere utilizzati solo per i casi d' impiego previsti nel catalogo e nella rispettiva documentazione tecnica. Qualora vengano impiegati prodotti o componenti di terzi, questi devono essere consigliati oppure approvati da Siemens. Il funzionamento corretto e sicuro dei prodotti presuppone un trasporto, un magazzinaggio, un' installazione, un montaggio, una messa in servizio, un utilizzo e una manutenzione appropriati e a regola d' arte. Devono essere rispettate le condizioni ambientali consentite. Devono essere osservate le avvertenze contenute nella rispettiva documentazione.
--

Marchio di prodotto

Tutti i nomi di prodotto contrassegnati con ® sono marchi registrati della Siemens AG. Gli altri nomi di prodotto citati in questo manuale possono essere dei marchi il cui utilizzo da parte di terzi per i propri scopi può violare i diritti dei proprietari.

Esclusione di responsabilità

Abbiamo controllato che il contenuto di questa documentazione corrisponda all'hardware e al software descritti. Non potendo comunque escludere eventuali differenze, non possiamo garantire una concordanza perfetta. Il contenuto di questa documentazione viene tuttavia verificato periodicamente e le eventuali correzioni o modifiche vengono inserite nelle successive edizioni.

Benvenuti in STEP 7 ...

... il software di base SIMATIC per la creazione di programmi PLC in KOP, FUP oppure AWL per le stazioni SIMATIC S7-300/400.

Informazioni relative a questo Getting Started

In questo libro apprenderemo i principi fondamentali di SIMATIC STEP 7. Vi mostreremo le finestre di dialogo principali ed i modi di procedere sulla base di esercitazioni pratiche strutturate in modo da permetterVi di cominciare da un capitolo qualsiasi.

Ogni sottocapitolo è costituito da una parte teorica a sfondo grigio e da una parte pratica a sfondo verde. Le istruzioni operative iniziano con una freccia nella barra di scorrimento verticale verde e in alcuni casi si estendono per diverse pagine, terminando con un punto ed un rimando a maggiori informazioni.

Può essere utile avere già dimestichezza con il mouse, la tecnica delle finestre, i menu a discesa ecc. e essere in possesso delle nozioni di base del PLC.

Nei corsi di addestramento STEP 7 è possibile approfondire il proprio know-how esulando dal Getting Started, e apprendere come vengono create soluzioni di automazione complete con STEP 7.

Presupposti per operare con Getting Started

Per poter svolgere le esercitazioni pratiche di STEP 7 contenute in questo Getting Started sono necessari

- un dispositivo di programmazione Siemens o un PC
- il pacchetto software STEP 7 e il License Key relativo
- un sistema di automazione SIMATIC S7-300 oppure S7-400 (per il capitolo 7 "Caricamento e test del programma")

Ulteriore documentazione relativa a STEP 7

- Nozioni fondamentali STEP 7
- Nozioni di riferimento STEP 7

E' possibile accedere ai manuali elettronici dopo l'installazione di STEP 7 tramite il menu **Start > Simatic > Documentazione** o ordinarli presso qualsiasi concessionario Siemens. Tutte le informazioni contenute nei manuali possono essere richiamate in STEP 7 mediante la Guida online.

Buon lavoro

SIEMENS AG

Contenuto

1	Introduzione a STEP 7	
1.1	Materia di apprendimento	7
1.2	Interazione tra software e hardware	9
1.3	Come procedere con STEP 7	10
1.4	Installazione di STEP 7	11
2	SIMATIC Manager	
2.1	Come avviare SIMATIC Manager e creare il progetto	13
2.2	Struttura del progetto nel SIMATIC Manager e richiamo della Guida a STEP 7	16
3	Programmazione con nomi simbolici	
3.1	Indirizzo assoluto	19
3.2	Come programmare con nomi simbolici	20
4	Creazione di un programma nell'OB1	
4.1	Come aprire la finestra del programma KOP/AWL/FUP e l'OB1	23
4.2	Programmazione dell'OB1 in KOP	26
4.3	Programmazione dell'OB1 in AWL	30
4.4	Programmazione dell'OB1 in FUP	33
5	Creazione di un programma con FB e DB	
5.1	Come creare ed aprire un blocco funzionale	37
5.2	Programmazione dell'FB1 in KOP	39
5.3	Programmazione dell'FB1 in AWL	43
5.4	Programmazione dell'FB1 in FUP	46
5.5	Come creare i blocchi dati di istanza e modificare i valori attuali	50
5.6	Come programmare un richiamo di blocco in KOP	52
5.7	Come programmare un richiamo di blocco in AWL	55
5.8	Come programmare un richiamo di blocco in FUP	57

Nei capitoli 3 - 5 viene creato un programma semplice.

1 Introduzione a STEP 7

1.1 Materia di apprendimento

Sulla base di esercitazioni pratiche illustreremo come è semplice programmare in KOP, FUP o AWL con STEP 7.

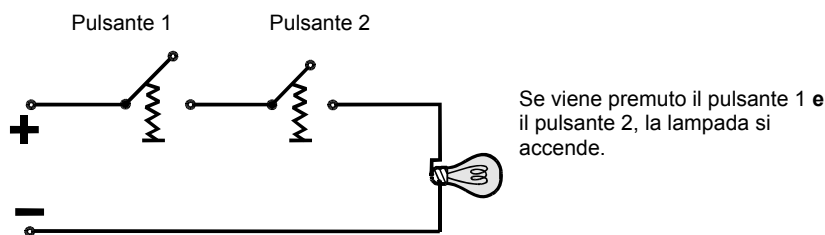
Istruzioni operative dettagliate nei singoli capitoli mostrano passo per passo le molteplici possibilità di utilizzo di STEP 7.

Creazione di un programma con operazioni logiche binarie

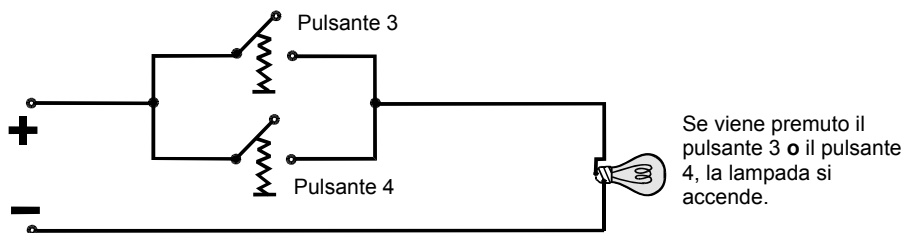
Nei capitoli 2 - 7 viene creato un programma con operazioni logiche binarie. Mediante le operazioni logiche combinatorie programmate vengono indirizzati gli ingressi e le uscite della CPU (se presente).

Gli esempi di programmazione in "Getting Started" si basano su tre operazioni logiche combinatorie binarie fondamentali.

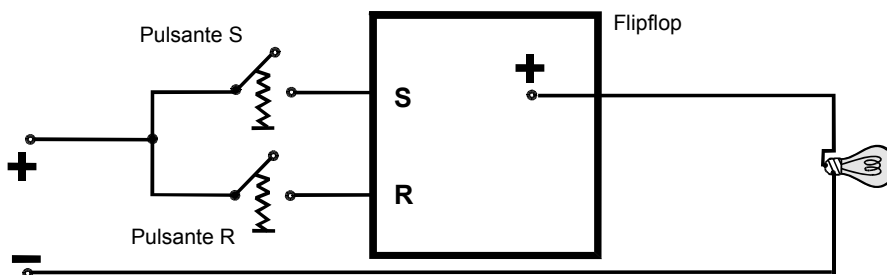
La prima operazione logica combinatoria binaria che programmerete è la funzione AND. La funzione AND può essere spiegata con l'esempio di un circuito elettrico a due pulsanti.



La seconda operazione logica combinatoria binaria è la funzione OR. Anche per spiegare la funzione OR viene utilizzato l'esempio di un circuito elettrico.



La terza operazione logica combinatoria binaria è il flipflop. La funzione SR reagisce in un circuito elettrico a determinati stati di tensione e li trasmette in modo corrispondente.

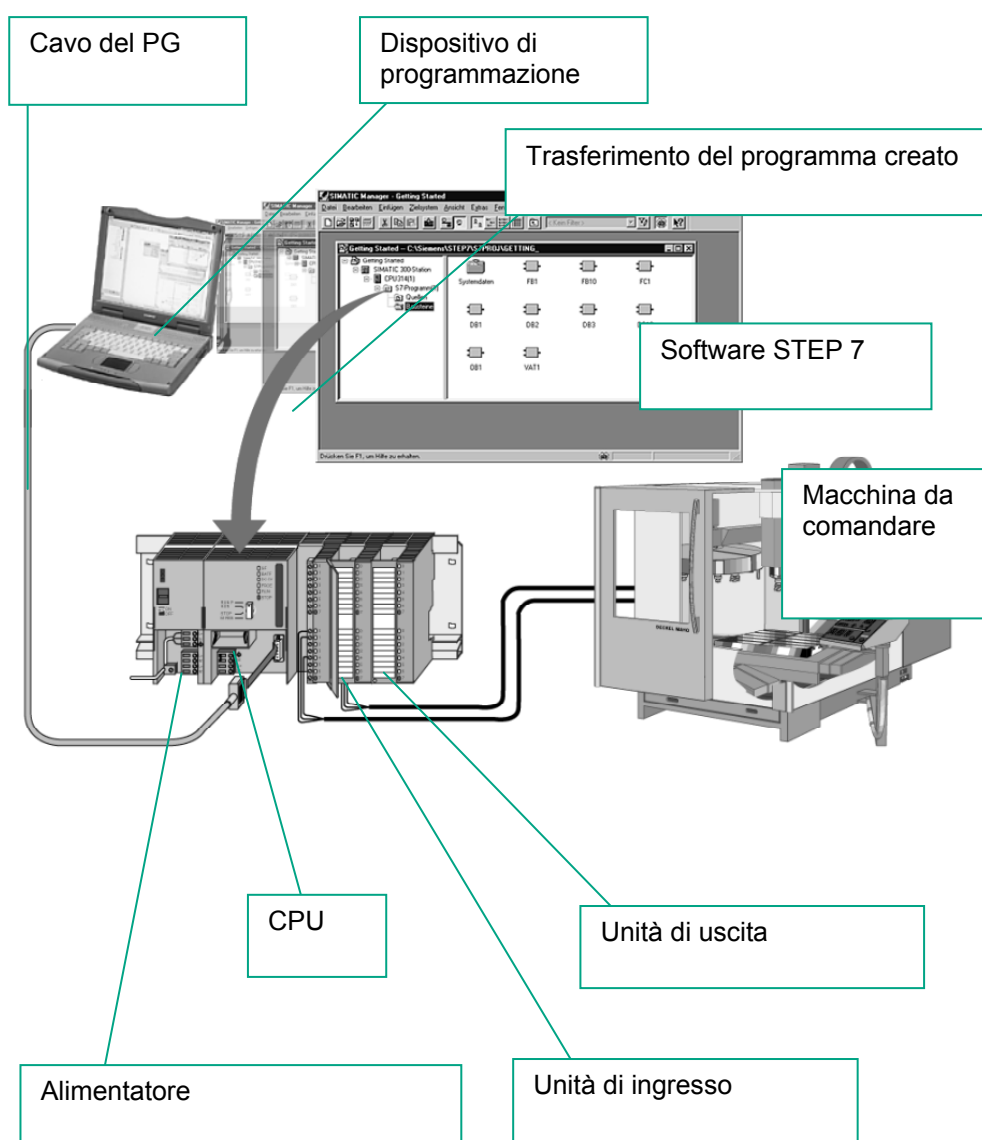


Se viene premuto il pulsante S, la lampada rimane accesa finché non viene premuto il pulsante R.

1.2 Interazione tra software e hardware

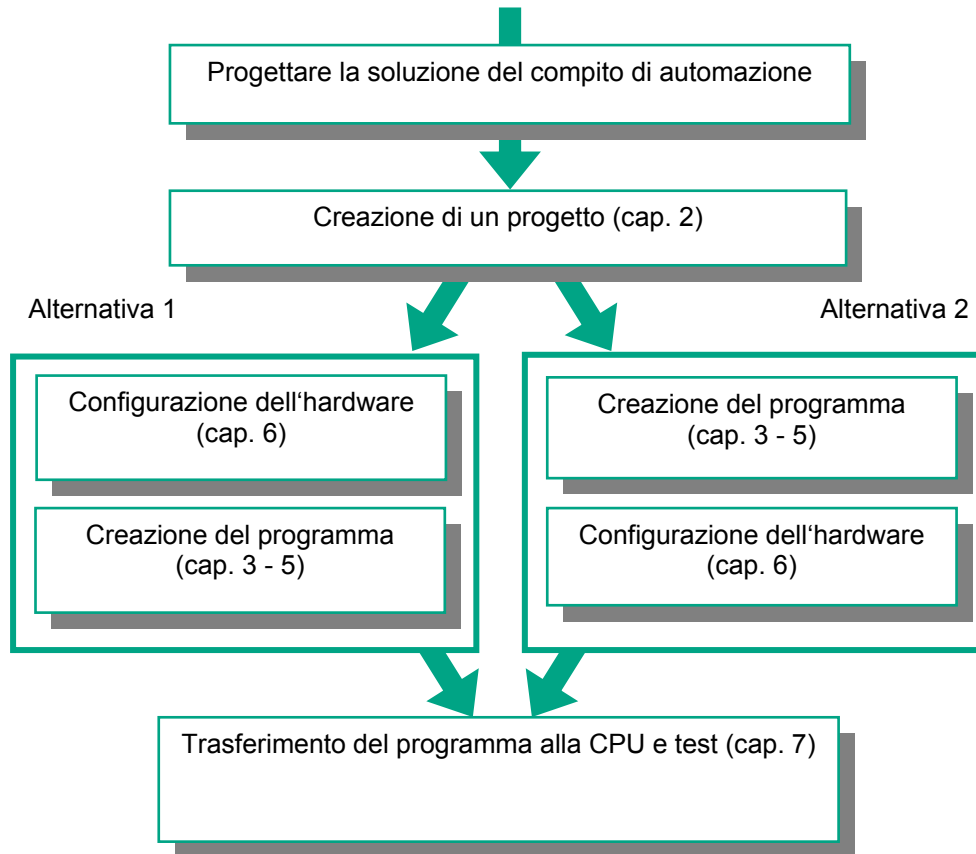
Con il software STEP 7 creiamo il programma S7 all'interno di un progetto. Il controllo S7 è costituito da un alimentatore, da una CPU e da unità di ingresso/uscita (unità I/O).

Il controllore programmabile (PLC) controlla e comanda la Vostra macchina con il programma S7. Le unità I/O vengono indirizzate nel programma S7 mediante gli indirizzi.



1.3 Come procedere con STEP 7

Prima di creare un progetto è utile sapere che i progetti STEP 7 possono essere creati in ordine diverso.



In caso di programmi voluminosi caratterizzati da numerosi ingressi e uscite si raccomanda di procedere in primo luogo alla configurazione dell'hardware. Ciò presenta il vantaggio che STEP 7 visualizza gli indirizzi possibili nell'editor dell'applicazione Configurazione HW.

In alternativa, l'utente stesso definisce gli indirizzi in questione a seconda dei componenti scelti e non è possibile richiamare questi indirizzi mediante STEP 7.

Con la configurazione hardware è possibile non solo stabilire gli indirizzi ma anche modificare i parametri e le proprietà delle unità modulari. Per il funzionamento di più CPU occorre ad esempio adattare gli indirizzi MPI delle CPU.

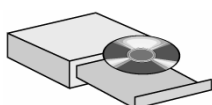
Dato che in „Getting Started“ utilizziamo solo pochi ingressi e uscite, trascureremo per il momento la configurazione hardware ed inizieremo subito con la programmazione.

1.4 Installazione di STEP 7

Che si inizi con la programmazione o con la configurazione dell'hardware, è necessario prima di tutto installare STEP 7. Nel caso utilizzate un SIMATIC PG, STEP 7 è preinstallato.



Per l'installazione del software STEP 7 su un PG/PC senza software STEP 7 preinstallato, rispettare i presupposti software e hardware descritti nel file Leggimi.wri sul CD di STEP 7 sotto **<drive>:\STEP 7\Disk1**.



Se invece si deve ancora installare STEP 7, inserire il CD di STEP 7 nel lettore per CD-ROM. Il programma di installazione viene avviato automaticamente. Seguire le istruzioni di installazione.

Nel caso non si riesca ad avviare automaticamente l'installazione, il programma di installazione si trova anche sul CD-ROM alla directory **<drive>:\STEP 7\Disk1\setup.exe**.



SIMATIC Manager

Terminata l'installazione e riavviato il computer, sul Desktop di Windows compare il simbolo "SIMATIC Manager".

Facendo doppio clic sul simbolo "SIMATIC Manager", viene avviato automaticamente l'Assistente di STEP 7.

Per altre istruzioni relative all'installazione vedere il file Leggimi.wri sul CD di STEP 7 alla directory **<drive>:\STEP 7\Disk1\Leggimi.wri**.

2 SIMATIC Manager

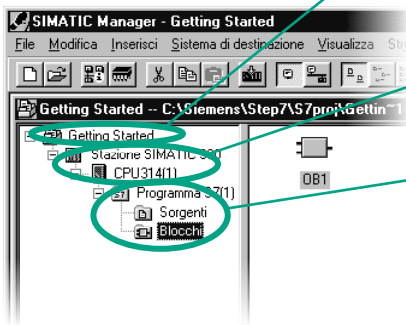
2.1 Come avviare il SIMATIC Manager e creare il progetto

Il SIMATIC Manager viene attivato come finestra centrale dopo l'avvio di STEP 7. Nella preimpostazione viene contemporaneamente avviato l'Assistente di STEP 7, che Vi assiste nella creazione di un progetto STEP 7. La struttura del progetto serve a memorizzare in modo ordinato tutti i dati ed i programmi in questione.

I dati vengono memorizzati sotto forma di oggetti in una struttura gerarchica all'interno di un progetto.

I dati di configurazione ed i parametri dell'hardware si trovano all'interno della stazione SIMATIC e della CPU.

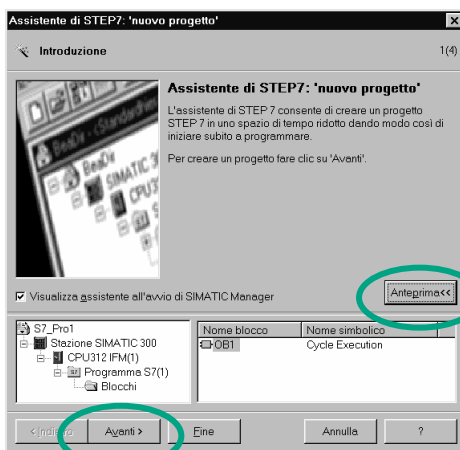
Il programma S7 comprende tutti i blocchi con i programmi necessari per il comando della macchina.

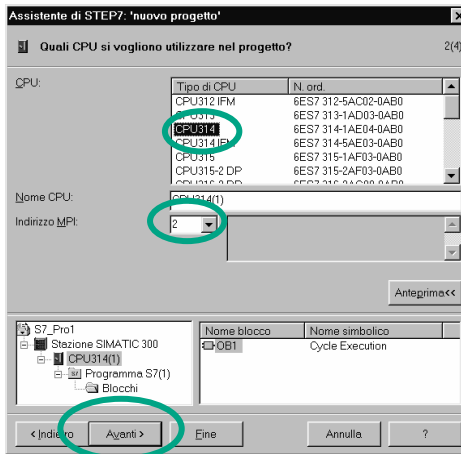


Fare doppio clic sul simbolo **SIMATIC Manager** nel desktop di Windows. Se l'Assistente non si avvia automaticamente, richiamare il comando di menu **File > Assistente "Nuovo progetto"**.

Con **Anteprima** è possibile attivare e disattivare la struttura del progetto che viene creato.

Con **Avanti** si giunge alla seconda finestra di dialogo





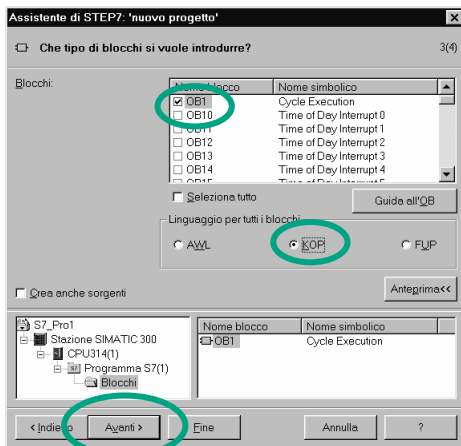
Per il progetto di esempio "Getting Started" selezionare la CPU 314. L'esempio è stato concepito in modo da consentire di selezionare in qualsiasi momento la CPU disponibile.

L'indirizzo MPI 2 è preimpostato.

Con **Avanti** si confermano le impostazioni e si giunge alla finestra di dialogo successiva.

Ogni CPU ha determinate proprietà, p.e. per quanto riguarda la configurazione della memoria o le aree di operandi. Pertanto la CPU deve essere selezionata prima della programmazione.

L'indirizzo MPI (Multi Point Interface) è necessario per la comunicazione della CPU con il PG/PC.



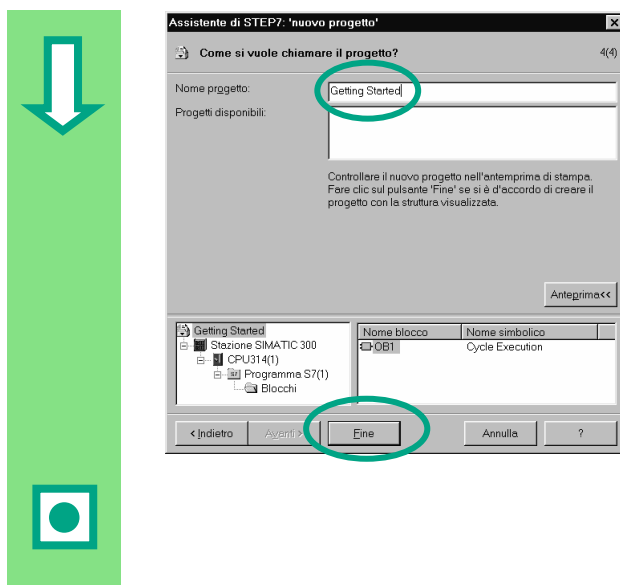
Selezionare il blocco organizzativo **OB1** (a meno che questo non sia già selezionato).

Scegliere il linguaggio di programmazione **KOP**, **FUP** o **AWL**.

Confermare le impostazioni con **Avanti**.

L'OB1 rappresenta il livello più elevato del programma e organizza gli altri blocchi del programma S7.

La scelta del linguaggio di programmazione può essere modificata in un secondo momento.



Nella casella "Nome progetto" scegliere con doppio clic il nome proposto e sovrascriverlo con "Getting Started".

Con **Fine** il nuovo progetto viene generato come nell'anteprima e memorizzato.

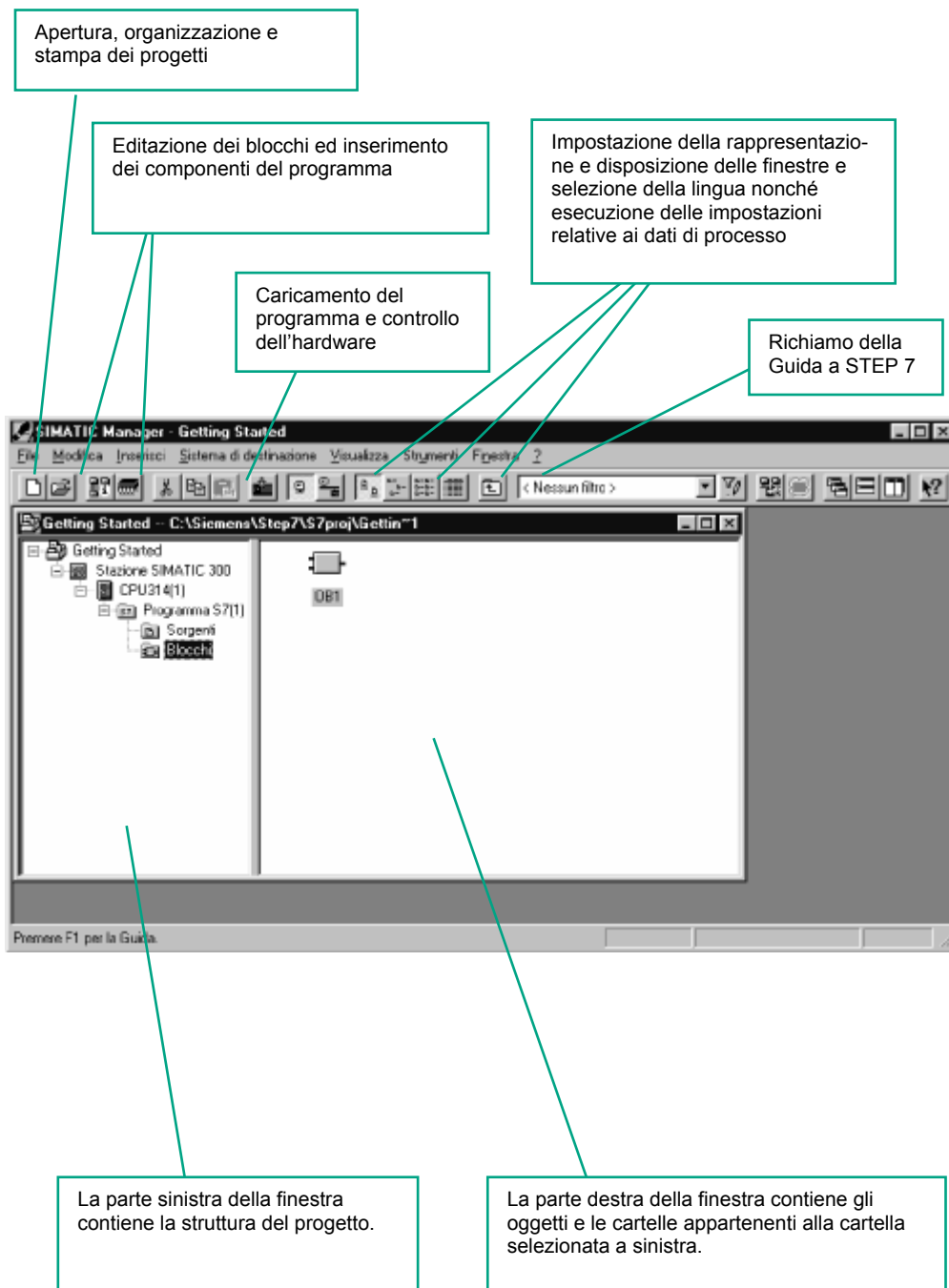
Dopo il comando **Fine** viene aperto il SIMATIC Manager con la finestra del progetto "Getting Started" creato. Nelle pagine seguenti Vi illustreremo il significato dei file e delle cartelle creati e come utilizzarli in modo efficace.

L'Assistente di STEP 7 viene attivato ad ogni avviamento del programma. Questa preimpostazione può essere disattivata nella prima finestra di dialogo dell'Assistente. Se create progetti senza l'Assistente di STEP 7, dovete tuttavia generare da Voi ogni directory all'interno del progetto.

Per maggiori informazioni vedere [?](#) > **Argomenti della Guida**
"Preparazione e elaborazione del progetto".

2.2 Struttura del progetto nel SIMATIC Manager e richiamo della Guida a STEP 7

Non appena è chiuso l'Assistente di STEP 7, compare il SIMATIC Manager con la finestra di progetto "Getting Started" aperta. Da questa finestra vengono richiamate tutte le funzioni e le finestre di STEP 7.



Richiamo della Guida a STEP 7

F1

Alternativa 1:

Selezionare un comando di menu qualsiasi e premere il tasto funzione **F1**. Comparirà la Guida sensibile al contesto relativa al comando di menu selezionato.

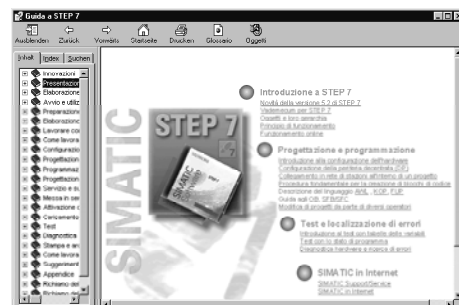
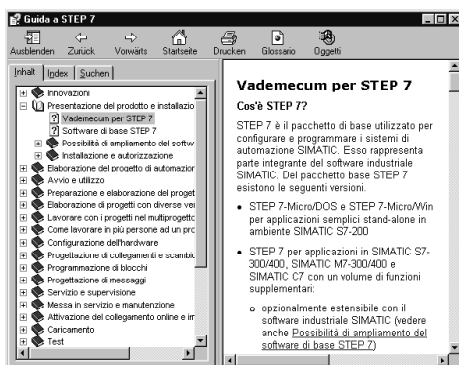
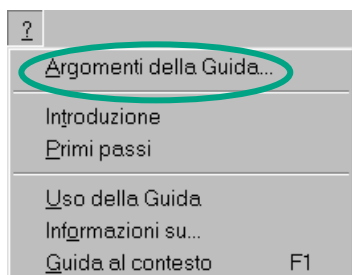
Alternativa 2:

Mediante il menu si accede alla Guida di STEP 7.

Nella sezione sinistra della finestra compare l'indice con i diversi argomenti della Guida e nella parte destra viene visualizzato l'argomento selezionato.

Navigare all'argomento desiderato facendo clic su **+** nel **Sommario**. Nella sezione destra viene visualizzato parallelamente il contenuto dell'argomento selezionato.

Con **Indice** e **Trova** si possono introdurre le voci di ricerca e cercare in modo mirato l'argomento desiderato.



Alternativa 3:

Nella Guida a STEP 7 il simbolo "Pagina iniziale" apre un portale informativo che consente l'accesso rapido ad argomenti fondamentali della Guida quali:

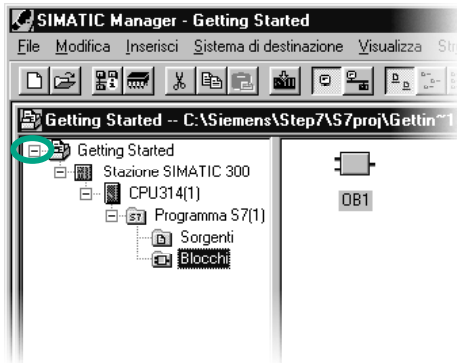
- primi passi con STEP 7
- progettazione e programmazione
- test e ricerca degli errori
- SIMATIC in Internet.

Alternativa 4:

Fare clic sul puntatore della Guida. Facendo clic su un determinato oggetto viene attivata la Guida.



Come navigare all'interno della struttura del progetto



Il progetto appena creato viene ora visualizzato con la stazione S7 selezionata e la CPU.

Fare clic su + oppure – per aprire o chiudere le cartelle desiderate.

Mediante i simboli visualizzati nella sezione destra della finestra si richiamano poi altre funzioni.

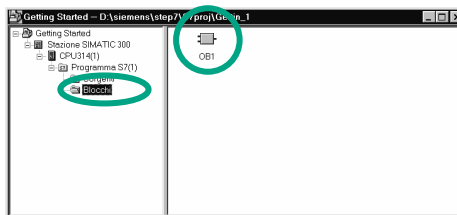


Fare clic sulla cartella

Programma S7 (1). Essa contiene tutti i componenti del programma necessari.

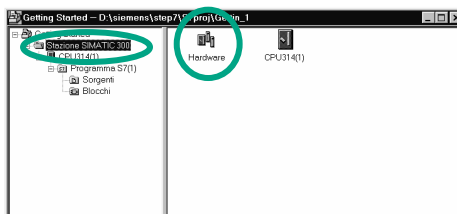
Nel capitolo 3 vengono assegnati nomi simbolici agli indirizzi mediante i simboli.

La cartella Sorgenti serve per memorizzare i programmi sorgente. I programmi sorgente non vengono trattati nel "Getting Started".



Fare clic sulla cartella **Blocchi**. Essa contiene l'**OB1** precedentemente creato e, successivamente, tutti gli altri blocchi.

Mediante i blocchi si giunge nei capitoli 4 e 5 all'introduzione del programma in KOP, FUP o AWL.



Fare clic sulla cartella **Stazione**

SIMATIC 300. Qui vengono memorizzati tutti i dati di progetto relativi all'hardware.

Mediante **Hardware** specificare al capitolo 6 i parametri del Vostro sistema di automazione.

Nel caso si voglia utilizzare un software di ampliamento SIMATIC per il compito di automazione, come per esempio i pacchetti opzionali PLC-SIM (programma di simulazione hardware) oppure S7-GRAPH (linguaggio di programmazione grafico), anche questi vengono integrati in STEP 7. Mediante il SIMATIC Manager è possibile aprire i relativi oggetti, p.es. un blocco funzionale S7-GRAPH, direttamente dal SIMATIC Manager.

Per maggiori informazioni vedere [?](#) > **Argomenti della Guida** "Elaborazione del progetto di automazione" e "Nozioni per lo sviluppo di strutture di programma".

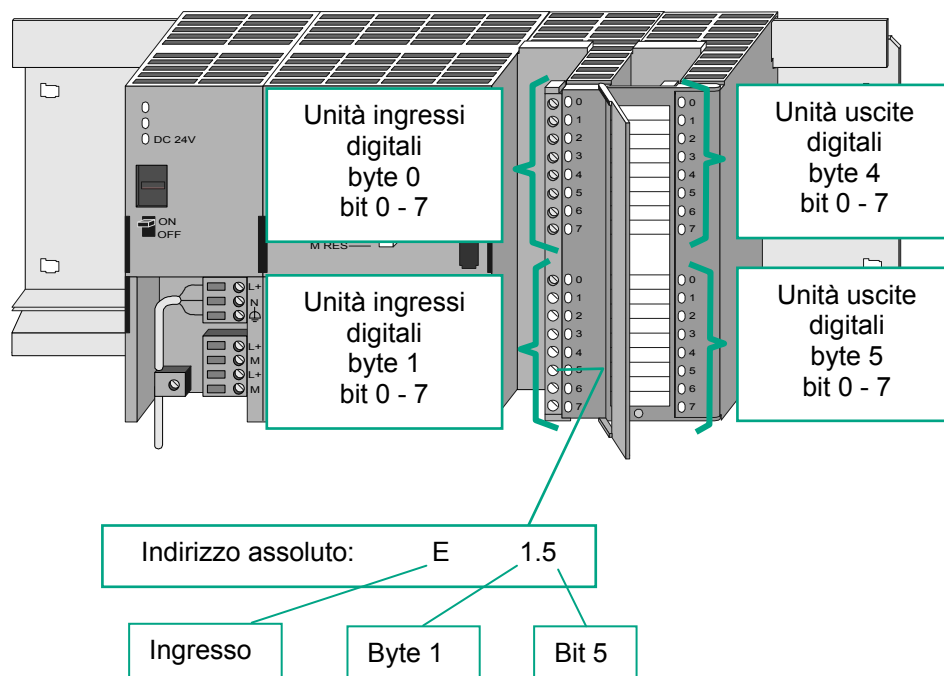
Per maggiori informazioni relative ai pacchetti vedere il catalogo SIMATIC ST 70 "Componenti per la Totally Integrated Automation".

3 Programmazione con nomi simbolici

3.1 Indirizzo assoluto

Ogni ingresso e ogni uscita ha un indirizzo assoluto predefinito dalla configurazione hardware. Questo viene indicato in modo diretto, ovvero assoluto.

L'indirizzo assoluto può essere sostituito da nomi simbolici a libera scelta.



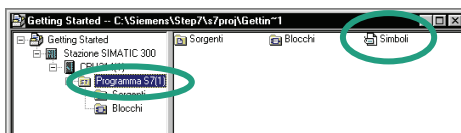
La programmazione assoluta dovrebbe essere utilizzata solamente se si vogliono indirizzare pochi ingressi e uscite nel programma S7.

3.2 Come programmare con nomi simbolici

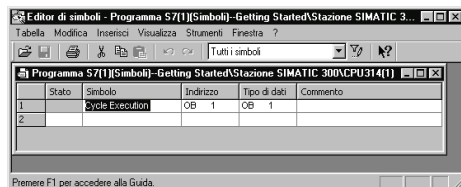
Nella tabella dei simboli assegnare un nome simbolico ed il tipo di dati a tutti gli indirizzi assoluti che verranno indirizzati nel Vostro programma, p.es. per l'ingresso E0.1 il simbolo Pulsante 1. Questi nomi valgono per tutte le parti del programma e vengono identificati come variabili globali.

Per mezzo della programmazione simbolica è possibile migliorare notevolmente la leggibilità del programma S7 da Voi creato.

Come lavorare con l'editor di simboli



Nella finestra del progetto "Getting Started" navigare a **Programmi S7 (1)** e con un doppio clic aprire **Simboli**.



La tabella dei simboli è costituita momentaneamente solo dal blocco organizzativo predefinito OB1.

Stato	Simbolo	Indirizzo	Tipo di dati	Commento
1	Cycle Execution	OB 1	OB 1	
2				

Fare clic su **Cycle Execution** e sovrascriverlo, per il nostro esempio, con "Programma principale".

Stato	Simbolo	Indirizzo	Tipo di dati	Commento
1	Programma principale	OB 1	OB 1	
2	Lampadina verde	A 4.0	BOOL	

Nella riga 2 introdurre "Lampadina verde" e "A 4.0". Il tipo di dati viene aggiunto automaticamente.



Fare clic nella colonna di commento della riga 1 o 2 per introdurre un commento relativo al simbolo in questione. Le introduzioni di una riga vengono concluse con **Invio** per inserire una nuova riga.

Stato	Simbolo	Indirizzo	Tipo di dati	Commento
1	Programma principale	OB 1	OB 1	
2	Lampadina verde	A 4.0	BOOL	
3	Lampadina rossa	A 4.1	BOOL	

Introdurre nella riga 3 "Lampadina rossa" e "A 4.1" e terminare l'introduzione con **Invio**.

In questo modo viene assegnato un nome simbolico dell'indirizzo assoluto a tutti gli ingressi e le uscite di cui necessita il programma.



Salvare le introduzioni o le modifiche nella tabella dei simboli e chiudere la finestra.

Dato che vi sono innumerevoli nomi per l'intero progetto "Getting Started" copieremo nel capitolo seguente la tabella dei simboli nel progetto "Getting Started".



Stato	Simbolo	Indirizzo	Tipo di dati	Commento
	Automatico_on	E 0.5	BOOL	per la funzione di memoria (Inserisci)
	Benzina	DB 1	FB 1	Dati per il motore a benzina
	Diesel	DB 2	FB 1	Dati per il motore diesel
	Disinserisci_MD	E 1.1	BOOL	Disinserisci il motore a benzina
	Disinserisci_MD	E 1.5	BOOL	Disinserisci motore diesel
	Funzionamento automatico	A 4.2	BOOL	Uscita con funzione di memoria
	G_dati	DB 3	DB 3	Blocco dati globale
	Inserisci_MB	E 1.0	BOOL	Inserisci il motore a benzina
	Inserisci_MD	F 1.4	BOOL	Inserisci motore diesel
	Lampadina rossa	A 4.1	BOOL	Risultato dell'interrogazione OR
	Lampadina verde	A 4.0	BOOL	Risultato dell'interrogazione AND
	Manuale on	E 0.6	BOOL	Per la funzione di memoria (disinserisci)
	MB_guasto	E 1.2	BOOL	Guasto del motore a benzina
	MB_on	A 5.0	BOOL	Inserisci comando per il motore a benzina
	MD_guasto	E 1.6	BOOL	Guasto del motore diesel
	MD_on	A 5.4	BOOL	Inserisci comando per il motore diesel
	Motore	FB 1	FB 1	Comando del motore
	Numero giri_attuale_MD	MV 2	INT	Numero di giri effettivo del motore a benzina
	Numero giri_attuale_MD	MV 4	INT	Numero di giri effettivo del motore diesel
	Prefissato_raggiunto_MB	A 5.1	BOOL	Visualizzazione numero giri prefissato raggiunto
	Prefissato_raggiunto_MD	A 5.5	BOOL	Visualizzazione numero giri prefissato raggiunto
	Programma principale	OB 1	OB 1	In questo blocco si trova il programma utente
	Pulsante 1	F 0.1	BOOL	per l'interrogazione AND
	Pulsante 2	E 0.2	BOOL	per l'interrogazione AND
	Pulsante 3	E 0.3	BOOL	per l'interrogazione OR
	Pulsante 4	E 0.4	BOOL	per l'interrogazione OR
	Tempo inerzia_MB	T 1	TIMER	Tempo per il funzionamento di inerzia del ventilatore
	Tempo inerzia_MD	T 2	TIMER	Tempo per il funzionamento di inerzia del ventilatore
	Ventilatore	FC 1	FC 1	Comando del ventilatore
	Ventilatore_MB_on	A 5.2	BOOL	Inserisci il comando per il ventilatore del motore a benzina
	Ventilatore_MD_on	A 5.8	BOOL	Inserisci comando per il ventilatore del motore diesel

Qui viene rappresentata in via esemplificativa la tabella dei simboli per il programma S7 per l'esempio "Getting Started" in AWL. Generalmente viene creata una tabella dei simboli per ciascun programma S7, vale a dire indipendentemente dal linguaggio di programmazione selezionato. Nella tabella dei simboli sono ammessi tutti i caratteri stampabili (p.es. dieresi, spaziature, ecc.).

Il tipo di dati prima inserito automaticamente nella tabella dei simboli stabilisce il tipo di segnale che la CPU deve elaborare. STEP 7 utilizza anche i tipi di dati seguenti:

BOOL BYTE WORD DWORD	Dati di questo tipo sono combinazioni di bit. Bit (tipo BOOL) fino a 32 Bit (DWORD).
CHAR	Dati di questo tipo occupano esattamente 1 carattere del set di caratteri ASCII.
INT DINT REAL	Sono disponibili per l'elaborazione di valori numerici (p.es. per il calcolo delle espressioni aritmetiche).
S5TIME TIME DATE TIME_OF_DAY	Dati di questo tipo rappresentano i diversi valori di data e ora all'interno di STEP 7 (p. es. per l'impostazione della data o per l'introduzione dell'ora).

Per maggiori informazioni vedere ? >
Argomenti della Guida "Programmazione di blocchi" e "Definizione di simboli".

4 Creazione di un programma nell'OB1

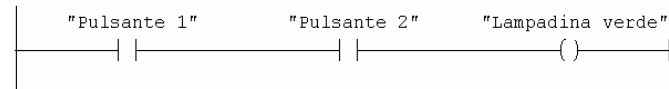
4.1 Come aprire la finestra del programma KOP/AWL/FUP e l'OB1

Scegliere tra KOP, AWL o FUP

Con STEP 7 i programmi S7 vengono creati nei linguaggi di programmazione standard KOP, AWL oppure FUP. Nell'uso pratico, così come in questo capitolo, si deve scegliere uno di questi linguaggi.

KOP (Schema a contatti)

è adatto per es. per utenti del settore elettrotecnico.



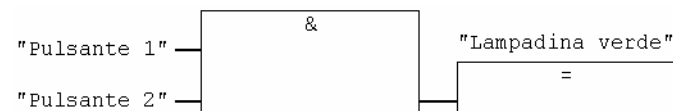
AWL (Lista istruzioni)

è adatto per es. per utenti del settore informatico.

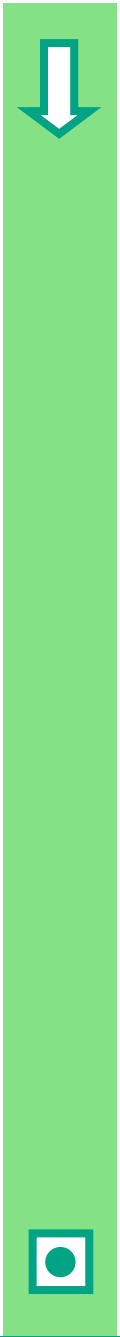
```
U    "Pulsante 1"  
U    "Pulsante 2"  
=    "Lampadina verde"
```

FUP (Schema logico)

è adatto per es. per utenti del settore della tecnica circuitale.



Il blocco viene ora aperto nel linguaggio usato per creare il blocco OB1 con l'assistente del progetto. E' possibile tuttavia modificare il linguaggio di programmazione in un secondo momento.



Come copiare la tabella dei simboli ed aprire l'OB1



Aprire il "Getting Started". Facendo clic sul simbolo **Apri**. Selezionare il "Getting Started" creato, e confermare con **OK**.

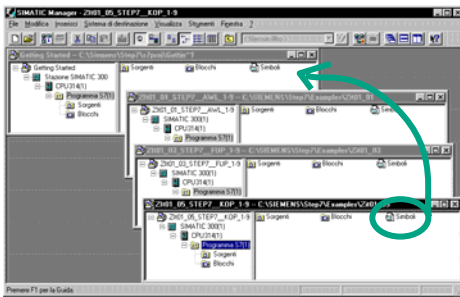
A seconda del linguaggio di programmazione scelto, aprire inoltre nella scheda "Progetti di esempio" il progetto:

- ZIt01_05_STEP7_**KOP**_1-9,
- ZIt01_01_STEP7_**AWL**_1-9 o
- ZIt01_03_STEP7_**FUP**_1-9

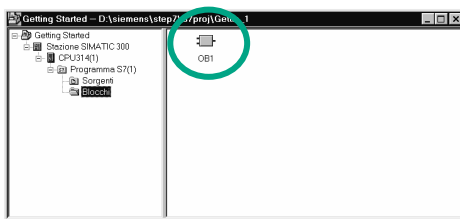
Vengono qui indicati a titolo dimostrativo tutti e tre i progetti di esempio.

In "ZIt01_XXX" Esempio navigare ai **Simboli** e copiarli con Drag & Drop nella finestra del progetto "Getting Started", Cartella **Programma S7**.

Chiudere poi la finestra "ZIt01_XXX" Esempio.



Drag & Drop significa che è possibile fare clic su un oggetto a piacere e trascinarlo in un'altra posizione tenendo premuto il tasto del mouse. Rilasciando il tasto del mouse l'oggetto viene depositato in quella posizione.



Fare doppio clic su **OB1** nel progetto "Getting Started". Si apre la finestra del programma KOP/AWL/FUP.

In STEP 7 l'OB1 viene elaborato ciclicamente dalla CPU. La CPU legge riga per riga i comandi del programma e li esegue. Quando la CPU inizia di nuovo dalla prima riga del programma, significa che ha eseguito esattamente un ciclo. Il tempo impiegato viene definito tempo di ciclo.

A seconda del linguaggio di programmazione selezionato, leggere come programmare con KOP al capitolo 4.2, con AWL al capitolo 4.3 e con FUP al capitolo 4.4.

Per maggiori informazioni vedere ? > **Argomenti della Guida** "Programmazione di blocchi" e "Creazione di blocchi e biblioteche".

La finestra del programma KOP/AWL/FUP

Nella finestra del programma KOP/AWL/FUP vengono programmati tutti i blocchi. Come esempio di linguaggio di programmazione abbiamo qui raffigurato il programma in KOP.

Inserimento di un nuovo segmento

Attivazione e disattivazione delle schede "Elementi di programma" e "Struttura di richiamo" (la finestra può essere collocata in qualsiasi posizione all'interno della finestra del programma)

Elementi principali del programma KOP e FUP

Modifica della visualizzazione del linguaggio di

Elementi del programma (qui di KOP) e Struttura di richiamo

Nome	Tipo di dati	Indirizzo	Commento
OB1_EV_CLASS	Byte	0.0	Bits 0-3 = 1 (Coming event)...
OB1_SCAN_1	Byte	1.0	1 (Cold restart scan 1 of O...
OB1_PRIORITY	Byte	2.0	Priority of OB Execution
OB1_...			
OB1_...			
OB1_...			
OB1_...			
OB1_...			
OB1_...			
OB1_...			
OB1_...			
OB1_DATE_TIME	Date_A...	12.0	Date and time OB1 started

La visualizzazione delle variabili sintetica e quella analitica contengono i parametri e le variabili locali del blocco

Casella del titolo e del commento al blocco o al segmento

Riga di introduzione del programma (anche segmento, montante)

Informazione sull'elemento selezionato del programma

Le diverse schede della finestra "Dettagli" consentono la visualizzazione di messaggi di errore e di informazioni sugli operandi, sulla modifica di simboli, sul controllo di operandi, sul confronto di blocchi e sulla modifica delle definizioni di errore per la diagnostica di processo.

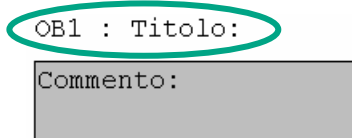
4.2 Programmazione dell'OB1 in KOP

Nel seguito programmeremo un circuito in serie, un circuito in parallelo e la funzione di memoria Imposta e Resetta in KOP (Schema a contatti).

Come programmare un circuito in serie in KOP



Impostare il linguaggio di programmazione **KOP** tramite il menu **Visualizza**.



Fare clic nell'area **Titolo** dell'OB1 e introdurre per es. "Programma principale elaborato ciclicamente".



Selezionare il montante per il primo elemento.



Fare clic sul simbolo nella barra degli strumenti ed inserire un contatto normalmente aperto.



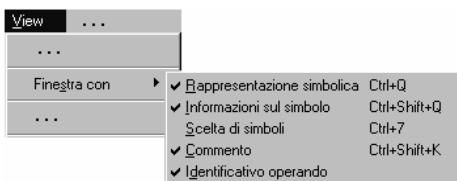
Inserire analogamente un secondo contatto normalmente aperto.



All'estremità destra del montante inserire una bobina.



Nel circuito in serie manca ancora l'indirizzamento dei contatti normalmente aperti e della bobina.



Verificare se è attivata la rappresentazione simbolica.



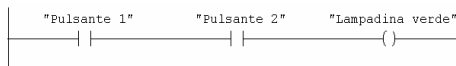
Fare clic su **???** e introdurre il nome simbolico "Pulsante 1" (tra virgolette).
In alternativa, scegliere il nome nella casella di riepilogo visualizzata.
Confermare premendo **Invio**.



Introdurre il nome simbolico "Pulsante 2" per il secondo contatto normalmente aperto.



Introdurre il nome "Lampadina verde" per la bobina.



Il circuito in serie è ora completamente programmato.



Salvare il blocco se non vi sono più simboli contrassegnati in rosso.

I simboli vengono contrassegnati in rosso se per es. il simbolo non è contenuto nella tabella dei simboli oppure vi è un errore sintattico.





Come programmare un circuito in parallelo in KOP

Segmento 1: Titolo:
 Commento:

Selezionare il **segmento 1**.



Inserire un nuovo segmento.



Selezionare di nuovo il montante.



Inserire un contatto normalmente aperto e una bobina.



Selezionare il ramo verticale del montante.



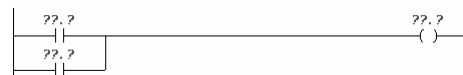
Inserire un ramo parallelo.



Nel ramo parallelo viene inserito un altro contatto normalmente aperto.



Chiudere la diramazione (se necessario selezionare la punta inferiore della freccia).



Nel circuito in parallelo manca solo l'indirizzamento.

Per l'indirizzamento simbolico procedere analogamente al circuito in serie.



Sovrascrivere il contatto normalmente aperto superiore con "Pulsante 3", il contatto normalmente aperto inferiore con "Pulsante 4" e la bobina con "Lampadina rossa".



Salvare il blocco.

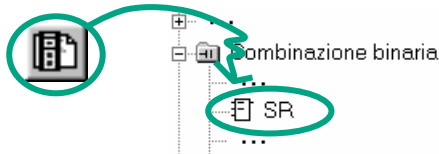
Come programmare la funzione di memoria in KOP



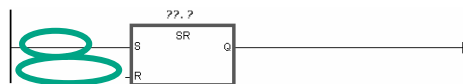
Selezionare il segmento 2 ed inserire un altro segmento.



Selezionare di nuovo il montante.



Nel catalogo degli elementi del programma navigare all'elemento **SR** attraverso **Combinazioni binarie**. Con doppio clic esso viene inserito.



Prima di ogni ingresso S e R inserire un contatto normalmente aperto.

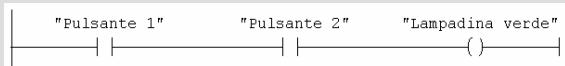


Introdurre i nomi simbolici seguenti per l'elemento SR: contatto normalmente aperto superiore "Automatico On", contatto normalmente aperto inferiore "Manuale On", elemento SR "Funzionamento automatico".



Salvare il blocco e chiudere la finestra.

Per vedere la differenza tra indirizzamento assoluto ed indirizzamento simbolico, disattivare nel menu **Visualizza > Finestra con > Rappresentazione simbolica**.



Esempio:
Indirizzamento simbolico in KOP



Esempio:
Indirizzamento assoluto in KOP

L' "a capo automatico" dell'indirizzamento simbolico viene modificato nella finestra del programma KOP/AWL/FUP con **Strumenti > Impostazioni > KOP/FUP > Larghezza campo operando**. È qui possibile impostare l' "a capo automatico" tra il 10° e 26° carattere.

Per maggiori informazioni vedere **? > Argomenti della Guida** "Programmazione di blocchi", "Creazione di blocchi di codice" e "Editazione di istruzioni KOP"

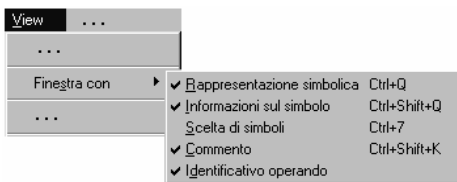
4.3 Programmazione dell'OB1 in AWL

Nel seguito programmeremo una istruzione AND, una istruzione OR e le istruzioni di memoria Imposta o Resetta in AWL (Lista istruzioni).

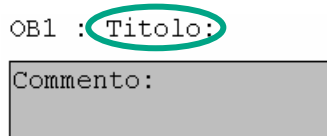
Come programmare l'istruzione AND in AWL



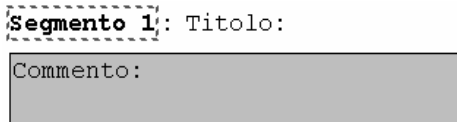
Impostare il linguaggio di programmazione **AWL** tramite il menu **Visualizza**.



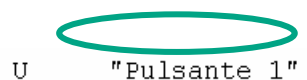
Verificare se è attivata la rappresentazione simbolica.



Fare clic nell'area **Titolo** dell'OB1 ed introdurre p. es. "Programma principale elaborato ciclicamente".



Selezionare l'area per la prima istruzione.



Introdurre nella prima riga del programma una U (AND) con uno spazio ed il simbolo "Pulsante 1" (tra virgolette).

Terminare la riga con **Invio**. Il cursore salta alla nuova riga.

```

U   "Pulsante 1"
U   "Pulsante 2"
=   "Lampadina verde"

```

Allo stesso modo completare l'istruzione AND.



L'istruzione AND è programmata completamente. Salvare il blocco nel caso non vi siano simboli contrassegnati in rosso.

I simboli vengono contrassegnati in rosso quando per es. il simbolo non è contenuto nella tabella dei simboli oppure vi è un errore sintattico.

Come programmare un'istruzione OR in AWL

Segmento 1: Titolo:

Selezionare il **segmento 1**.

Commento:



Inserire un nuovo segmento e selezionare l'area di introduzione.

```

O   "Pulsante 3"

```

Introdurre una O (OR) ed il simbolo "Pulsante 3" (analogamente a AND).

```

O   "Pulsante 3"
O   "Pulsante 4"
=   "Lampadina rossa"

```

Completare l'istruzione OR e salvarla.



Come programmare l'istruzione di memoria in AWL



Selezionare il segmento 2 ed inserire un altro segmento.

U "Automatico On"

Nella prima riga trascrivere l'istruzione U con il nome simbolico "Automatico On".

```
U "Automatico On"
S "Funzionamento automatico"
U "Manuale On"
R "Funzionamento automatico"
```

Completare l'istruzione di memoria e salvarla. Chiudere il blocco.

Per vedere la differenza tra indirizzamento assoluto e indirizzamento simbolico disattivare nel menu **Visualizza > Finestra con > Rappresentazione simbolica**.

```
U "Pulsante 1"
U "Pulsante 2"
= "Lampadina verde"
```

Esempio:
indirizzamento simbolico in AWL

```
U E 0.1
U E 0.2
= A 4.0
```

Esempio:
indirizzamento assoluto in AWL

Per maggiori informazioni vedere ? > **Argomenti della Guida** "Programmazione di blocchi", "Creazione di blocchi di codice" e "Editazione di istruzioni AWL"

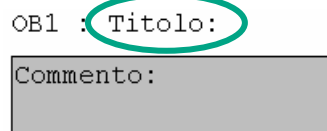
4.4 Programmazione dell'OB1 con FUP

Nel seguito programmiamo una funzione AND, una funzione OR e una funzione di memoria in FUP (Schema logico).

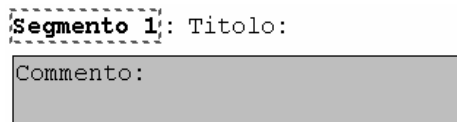
Come programmare una funzione AND in FUP



Impostare il linguaggio di programmazione **FUP** mediante il menu **Visualizza**.



Fare clic nell'area **Titolo** dell' OB1 ed introdurre per es. "Programma principale elaborato ciclicamente".



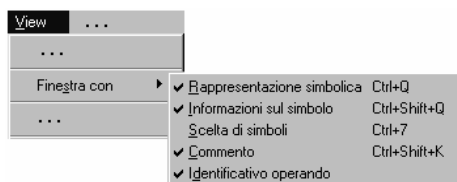
Selezionare l'area di introduzione della funzione AND (sotto il campo di commento).



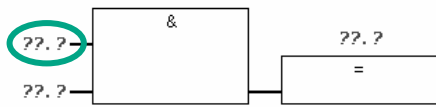
Inserire un box AND (&) ed una assegnazione (=).



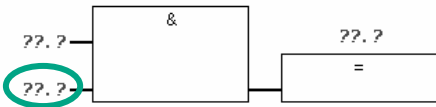
Nella funzione AND manca solo l'indirizzamento degli elementi.



Verificare se la rappresentazione simbolica è attivata.



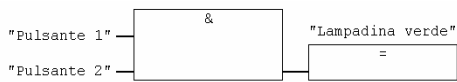
Fare clic su **??.** ed introdurre il nome simbolico "Pulsante 1" (tra virgolette).
In alternativa, scegliere il nome nella casella di riepilogo visualizzata.
Confermare premendo **Invio**.



Introdurre il nome simbolico "Pulsante 2" per il secondo ingresso.



Introdurre il nome "Lampadina verde" per l'assegnazione.



La funzione AND è programmata completamente.



Se non vi sono più simboli contrassegnati in rosso, è possibile salvare.

I simboli vengono contrassegnati in rosso quando per es. il simbolo non è contenuto nella tabella dei simboli oppure vi è un errore sintattico.

Come programmare una funzione OR in FUP



Inserire un nuovo segmento.

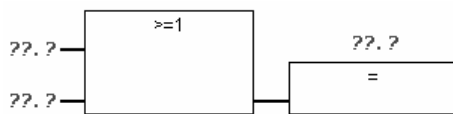
Segmento 2: Titolo:

Commento:

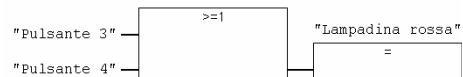
Selezionare di nuovo l'area di introduzione per la funzione OR.



Inserire un box OR - (≥ 1) e una assegnazione (=).



Nella funzione OR manca ancora l'indirizzamento. Procedere come per la funzione AND.



Introdurre "Pulsante 3" per l'ingresso superiore "Pulsante 4" per l'ingresso inferiore e "Lampadina rossa" per l'assegnazione.



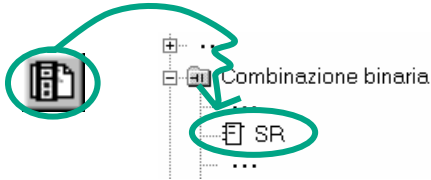
Salvare il blocco.



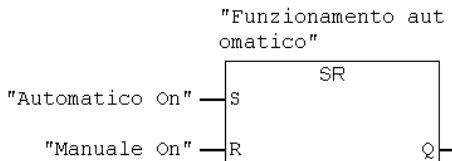
Come programmare una funzione di memoria in FUP



Selezionare il segmento 2 ed inserire un altro segmento. Selezionare di nuovo l'area di introduzione (sotto il campo del commento).



Nel catalogo degli elementi del programma navigare all'elemento **SR** tramite le **Combinazione binarie**. Con doppio clic questo elemento viene inserito.

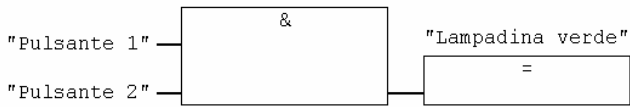


Introdurre i nomi simbolici seguenti per l'elemento SR :
 Imposta "Automatico On",
 Resetta "Manuale On",
 Merker "Funzionamento automatico".

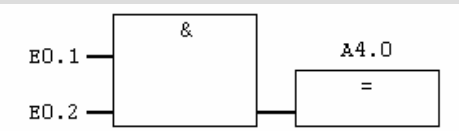


Salvare il blocco e chiudere la finestra.

Per vedere la differenza tra indirizzamento assoluto e indirizzamento simbolico disattivare nel menu **Visualizza > Finestra con > Rappresentazione simbolica**.



Esempio:
indirizzamento simbolico in FUP



Esempio:
indirizzamento assoluto in FUP

L' "a capo automatico" dell'indirizzamento simbolico viene modificato nella finestra del programma KOP/AWL/FUP con **Strumenti > Impostazioni > KOP/FUP > Larghezza campo operando**. È qui possibile impostare "l' a capo automatico" tra il 10° e 26° carattere.

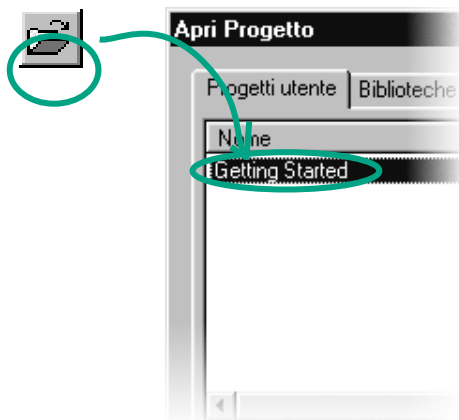
Per maggiori informazioni ? > **Argomenti della Guida** "Programmazione di blocchi", "Creazione di blocchi di codice" e "Editazione di istruzioni FUP".

5 Creazione di un programma con FB e DB

5.1 Come creare ed aprire un blocco funzionale

Il blocco funzionale (FB) è subordinato al blocco organizzativo. Esso contiene una parte del programma che può essere richiamata all'interno dell'OB1 ogni volta che lo si desidera. Tutti i parametri formali e i dati statici del blocco funzionale vengono memorizzati in un blocco dati DB separato che è assegnato al blocco funzionale.

Il blocco funzionale (FB1, nome simbolico "Motore", vedere tabella dei simboli pag. 3-3) viene programmato nella già nota finestra del programma KOP/AWL/FUP. È consigliabile utilizzare lo stesso linguaggio di programmazione del capitolo 4 (Creazione di un programma nell'OB1).



La tabella dei simboli deve precedentemente essere stata copiata nel progetto "Getting Started". In caso contrario, leggere a pagina 4-2 come copiare la tabella dei simboli e poi ritornare a questa pagina.

Se necessario, aprire il progetto "Getting Started".

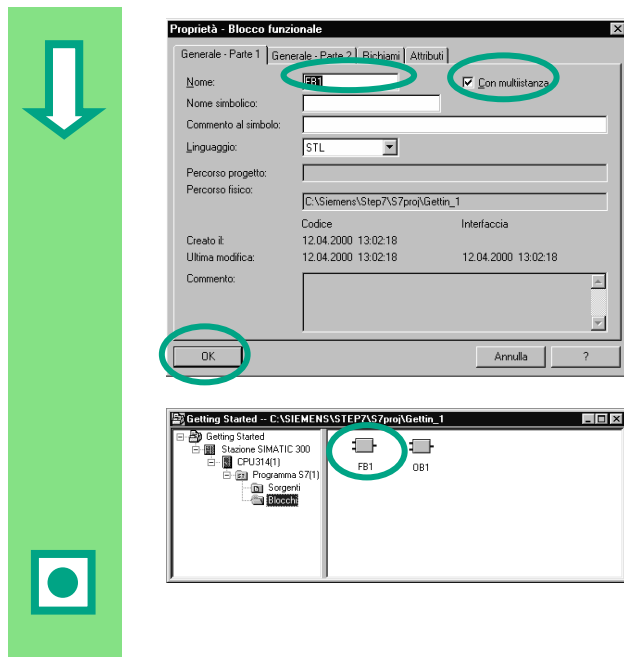


Navigare alla cartella **Blocchi** e aprirla.

Con il tasto destro del mouse fare clic nella sezione destra della finestra.



Il menu di scelta rapida del tasto destro del mouse contiene i comandi principali della barra dei menu. Inserire come nuovo oggetto un **blocco funzionale**.



Nella finestra di dialogo "Proprietà – Blocco funzionale" selezionare il linguaggio di programmazione, attivare la **Proprietà multiistanza** e con **OK** acquisire le altre preimpostazioni.

Il blocco funzionale **FB1** è stato inserito nella cartella Blocchi.

Facendo doppio clic sull'FB1 si accede alla finestra del programma KOP/AWL/FUP.

A seconda del linguaggio di programmazione scelto, leggere come procedere al capitolo 5.2 per KOP, al capitolo 5.3 per AWL e al capitolo 5.4 per FUP.

Per maggiori informazioni vedere ? > **Argomenti della Guida** "Programmazione di blocchi" e "Creazione di blocchi e biblioteche".

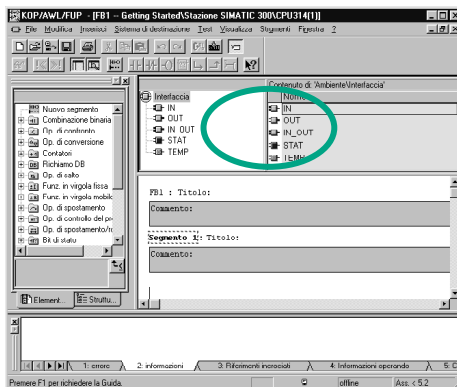
5.2 Programmazione dell'FB1 in KOP

Vi mostriamo come programmare un blocco funzionale che per es. comanda e controlla un motore a benzina e un motore diesel con un blocco dati ciascuno.

Tutti i segnali specifici del motore vengono trasferiti come parametri di blocco dal blocco organizzativo al blocco funzionale e devono perciò essere riportati nella tabella di dichiarazione delle variabili come parametri di ingresso e di uscita (dichiarazione "in" e "out").

Si dovrebbe già sapere come introdurre un circuito in serie, un circuito parallelo e una funzione di memoria con STEP 7.

Come dichiarare / definire le variabili



La finestra del programma KOP/ABL/FUP è aperta ed è stato richiamato il comando **Visualizza > KOP** (linguaggio di programmazione).

La riga del titolo contiene ora la dicitura "FB1", poiché la finestra del programma è stata aperta con doppio clic sull'FB1.

L'area per la dichiarazione delle variabili comprende una visualizzazione delle variabili sintetica (sezione sinistra) ed una visualizzazione analitica (sezione destra).

Nella visualizzazione sintetica selezionare in sequenza i tipi di dichiarazione "IN", "OUT" e "STAT" ed introdurre nei dettagli della variabile le dichiarazioni seguenti.

Nella visualizzazione analitica delle variabili fare clic sulle celle corrispondenti ed acquisirvi quanto illustrato nel seguito. Il tipo di dati può essere scelto nella casella di riepilogo visualizzata.





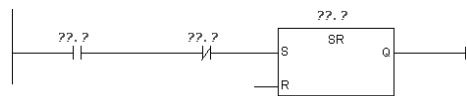
Contenuto di: 'Ambiente\Interfaccia\IN'					
	Nome	Tipo di dati	Indirizzo	Valore iniziale	Commento
Switch_On	Bool	0.0	FALSE	Inserisci motore	
Switch_Off	Bool	0.1	FALSE	Disinserisci motore	
Failure	Bool	0.2	FALSE	Disturbo del motore, causa la disinserzione	
Actual_Speed	Int	2.0	0	Numero di giri effettivo del motore	

Contenuto di: 'Ambiente\Interfaccia\OUT'					
	Nome	Tipo di dati	Indirizzo	Valore iniziale	Commento
Engine_On	Bool	4.0	FALSE	Il motore viene inserito	
Preset_Speed_Reached	Bool	4.1	FALSE	Numero di giri prefissato raggiunto	

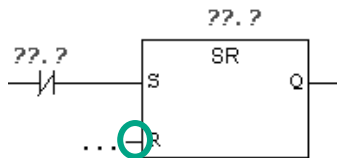
Contenuto di: 'Ambiente\Interfaccia\STAT'					
	Nome	Tipo di dati	Indirizzo	Valore iniziale	Commento
Preset_Speed	Int	6.0	1500	Numero di giri del motore richiesto	

Se nei dettagli delle variabili non sono presenti tutte le colonne necessarie, visualizzarle mediante il menu di scelta rapida (facendo un clic con il tasto destro del mouse).
Per i nomi dei parametri dei blocchi nei dettagli delle variabili sono ammesse solo lettere, cifre e il carattere di sottolineatura.

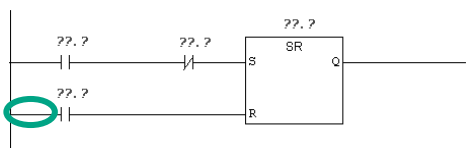
Come programmare l'inserzione e la disinserzione di un motore



Nel segmento 1 inserire in successione un contatto normalmente aperto, un contatto normalmente chiuso e un elemento SR mediante i simboli corrispondenti oppure dal catalogo degli elementi del programma.



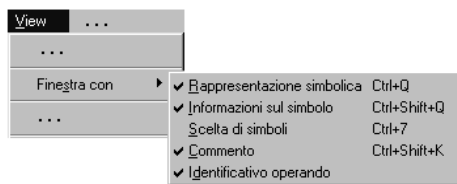
Selezionare il montante subito prima dell'ingresso R.



Inserire un altro contatto normalmente aperto. Selezionare il montante subito prima del contatto normalmente aperto.



Inserire un contatto normalmente chiuso parallelamente al contatto normalmente aperto.

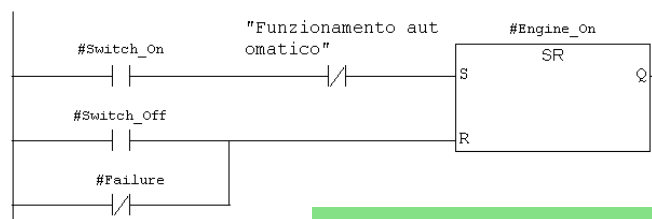


Verificare se è attivata la rappresentazione simbolica.

Selezionare i punti interrogativi ed introdurre i relativi nomi nella tabella di dichiarazione (# viene assegnato automaticamente).

Per il contatto normalmente chiuso del circuito in serie introdurre il nome simbolico "Funzionamento automatico".

Infine salvare il programma.



Le variabili locali di blocco vengono identificate con # e sono valide solo nel blocco.

Le variabili globali sono racchiuse tra virgolette. Vengono definite nella tabella dei simboli e sono valide nell'intero programma.

Lo stato di segnale "Funzionamento automatico" viene definito nell'OB1 (segmento 3, vedere pagina 4-7) da un altro elemento SR e interrogato ora nell'FB1.





Come programmare il controllo del numero di giri



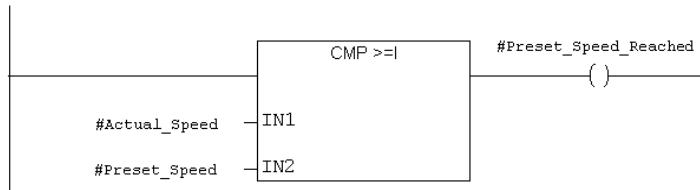
Inserire un nuovo segmento e selezionare il montante.
 Nel catalogo degli elementi del programma navigare a **Op. di confronto** ed inserire un **CMP>=I**.



Nel montante inserire inoltre una bobina.

Selezionare di nuovo i punti interrogativi e trascrivere i nomi della bobina e del comparatore prendendoli dalla dichiarazione delle variabili.

Salvare il programma.



Quando viene inserito e disinserto il motore?

Se la variabile #Inserisci ha lo stato di segnale "1" e (AND) la variabile "Funzionamento automatico" ha lo stato di segnale "0", il motore viene inserito. Solo la negazione (contatto normalmente chiuso) di "Funzionamento automatico" permette questa funzionalità.

Se la variabile #Disinserisci ha lo stato di segnale "1" o (OR) la variabile #Guasto ha lo stato di segnale "0", il motore viene disinserto. La funzionalità desiderata viene di nuovo conseguita con la negazione di #Guasto (#Guasto è un segnale "zeroattivo" e in caso normale ha valore "1", in caso di guasto ha valore "0").

Come controlla l'operazione di confronto il numero di giri del motore?

Mediante l'operazione di confronto vengono confrontate le variabili #Actual_Speed e #Preset_Speed e viene assegnato il risultato delle variabili #Preset_Speed_Reached (stato di segnale 1).

Per maggiori informazioni vedere ? > **Argomenti della Guida** "Programmazione di blocchi", "Creazione di blocchi di codice" e "Editazione della dichiarazione delle variabili" o "Editazione di istruzioni KOP".

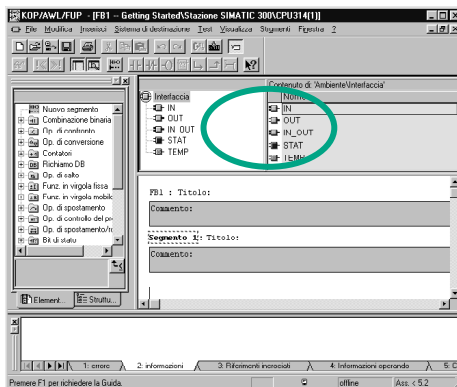
5.3 Programmazione dell'FB1 in AWL

Vi mostriamo come programmare un blocco funzionale che per es. comanda e controlla un motore a benzina e un motore diesel con un blocco dati ciascuno.

Tutti i segnali specifici del motore vengono trasferiti come parametri di blocco dal blocco organizzativo al blocco funzionale e devono perciò essere riportati nella tabella di dichiarazione delle variabili come parametri di ingresso e di uscita (dichiarazione "in" e "out").

Si dovrebbe già sapere come introdurre con STEP 7 una istruzione AND, una istruzione OR e le istruzioni di memoria Imposta e Resetta.

Come dichiarare / definire le variabili



La finestra del programma KOP/AWL/FUP è aperta ed è attivato **Visualizza > AWL** (linguaggio di programmazione).

Fare attenzione alla riga di intestazione: vi si legge ora FB1, poiché la finestra del programma è stata aperta con doppio clic sull'FB1.

L'area per la dichiarazione delle variabili comprende una visualizzazione delle variabili sintetica (sezione sinistra) ed una visualizzazione analitica (sezione destra).

Nella visualizzazione sintetica selezionare in sequenza i tipi di dichiarazione "IN", "OUT" e "STAT" ed introdurre nei dettagli della variabile le dichiarazioni seguenti.

Nella visualizzazione analitica delle variabili fare clic sulle celle corrispondenti ed acquisirvi quanto illustrato nel seguito. Il tipo di dati può essere scelto nella casella di riepilogo visualizzata.



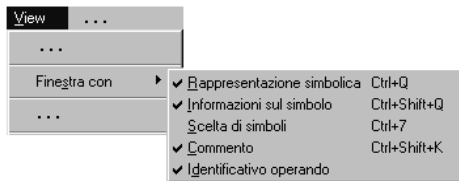
Contenuto di: 'Ambiente\Interfaccia\IN'					
	Nome	Tipo di dati	Indirizzo	Valore iniziale	Commento
	Switch_On	Bool	0.0	FALSE	Inserisci motore
	Switch_Off	Bool	0.1	FALSE	Disinserisci motore
	Failure	Bool	0.2	FALSE	Disturbo del motore, causa la disinserzione
	Actual_Speed	Int	2.0	0	Numero di giri effettivo del motore

Contenuto di: 'Ambiente\Interfaccia\OUT'					
	Nome	Tipo di dati	Indirizzo	Valore iniziale	Commento
	Engine_On	Bool	4.0	FALSE	Il motore viene inserito
	Preset_Speed_Reached	Bool	4.1	FALSE	Numero di giri prefissato raggiunto

Contenuto di: 'Ambiente\Interfaccia\STAT'					
	Nome	Tipo di dati	Indirizzo	Valore iniziale	Commento
	Preset_Speed	Int	6.0	1500	Numero di giri del motore richiesto

Per i nomi dei parametri di blocco nella dichiarazione delle variabili sono ammesse solo lettere, cifre e il carattere di sottolineatura.

Come programmare l'inserzione e la disinserzione di un motore



Verificare se è attivata la rappresentazione simbolica.

```

U      #Switch_On
UN     "Automatic_Mode"
S      #Engine_On
O      #Switch_Off
ON     #Failure
R      #Engine_On
    
```

Introdurre nel segmento 1 le istruzioni corrispondenti.

Le variabili locali di blocco vengono identificate con # e sono valide solo nel blocco.
 Le variabili globali sono racchiuse tra virgolette. Vengono definite nella tabella dei simboli e sono valide nell'intero programma.
 Lo stato di segnale "Funzionamento automatico" viene definito nell'OB1 (segmento 3, vedere pagina 4-10) da un altro elemento SR e interrogato ora nell'FB1.



Come programmare il controllo del numero di giri

```
L   #Actual_Speed
L   #Preset_Speed
>=I
=   #Preset_Speed_Reached
```

Inserire un nuovo segmento ed introdurre le istruzioni corrispondenti. Salvare infine il programma.



Quando viene inserito e disinserito il motore?

Se la variabile #Inserisci ha lo stato di segnale "1" e (AND) la variabile "Funzionamento automatico" ha lo stato di segnale "0", il motore viene inserito. Solo la negazione (contatto normalmente chiuso) di "Funzionamento automatico" permette questa funzionalità.

Se la variabile #Disinserisci ha lo stato di segnale "1" o (OR) la variabile #Guasto ha lo stato di segnale "0", il motore viene disinserito. La funzionalità desiderata viene di nuovo conseguita con la negazione di #Guasto (#Guasto è un segnale "zeroattivo" e in caso normale ha valore "1", in caso di guasto ha valore "0").

Come controlla l'operazione di confronto il numero di giri del motore?

Mediante l'operazione di confronto vengono confrontate le variabili #Actual_Speed e #Preset_Speed e viene assegnato il risultato delle variabili #Preset_Speed_Reached (stato di segnale 1).

Per maggiori informazioni vedere ? > **Argomenti della Guida** "Programmazione di blocchi", "Creazione di blocchi di codice" e "Editazione della dichiarazione delle variabili" o "Editazione di istruzioni AWL".

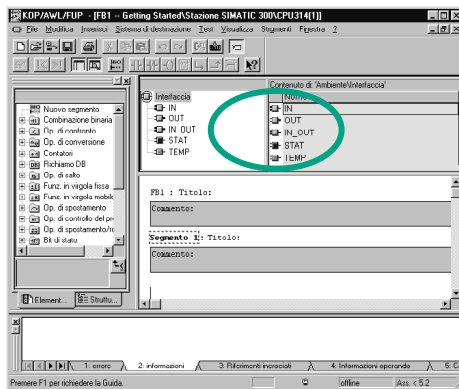
5.4 Programmazione dell'FB1 in FUP

Vi mostriamo come programmare un blocco funzionale che per es. comanda e controlla un motore a benzina e un motore diesel con un blocco dati ciascuno.

Tutti i segnali specifici del motore vengono trasferiti come parametri di blocco dal blocco organizzativo al blocco funzionale e devono perciò essere riportati nella dichiarazione delle variabili come parametri di ingresso e di uscita (dichiarazione "in" e "out").

Si dovrebbe già sapere come introdurre con STEP 7 una funzione AND, una funzione OR ed una funzione di memoria STEP 7.

Come dichiarare / definire le variabili



La finestra del programma KOP/AWL/FUP è aperta ed è attivato **Visualizza > FUP** (linguaggio di programmazione).

Fare attenzione alla riga di intestazione; vi si legge ora FB1, poiché la finestra del programma è stata aperta con doppio clic sull'FB1.

L'area per la dichiarazione delle variabili comprende una visualizzazione delle variabili sintetica (sezione sinistra) ed una visualizzazione analitica (sezione destra).

Nella visualizzazione sintetica selezionare in sequenza i tipi di dichiarazione "IN", "OUT" e "STAT" ed introdurre nei dettagli della variabile le dichiarazioni seguenti.

Nella visualizzazione analitica delle variabili fare clic sulle celle corrispondenti ed acquisirvi quanto illustrato nel seguito. Il tipo di dati può essere scelto nella casella di riepilogo visualizzata.



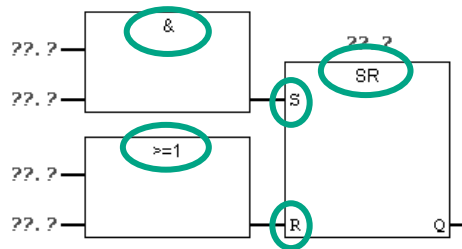
Contenuto di: 'Ambiente\Interfaccia\IN'					
Nome	Tipo di dati	Indirizzo	Valore iniziale	Commento	
Switch_On	Bool	0.0	FALSE	Inserisci motore	
Switch_Off	Bool	0.1	FALSE	Disinserisci motore	
Failure	Bool	0.2	FALSE	Disturbo del motore, causa la disinserzione	
Actual_Speed	Int	2.0	0	Numero di giri effettivo del motore	

Contenuto di: 'Ambiente\Interfaccia\OUT'					
Nome	Tipo di dati	Indirizzo	Valore iniziale	Commento	
Engine_On	Bool	4.0	FALSE	Il motore viene inserito	
Preset_Speed_Reached	Bool	4.1	FALSE	Numero di giri prefissato raggiunto	

Contenuto di: 'Ambiente\Interfaccia\STAT'					
Nome	Tipo di dati	Indirizzo	Valore iniziale	Commento	
Preset_Speed	Int	6.0	1500	Numero di giri del motore richiesto	

Per i nomi dei parametri dei blocchi nella dichiarazione delle variabili sono ammesse solo lettere, cifre e il carattere di sottolineatura.

Come programmare l'inserzione e disinserzione del motore



Nel segmento 1 inserire una funzione SR mediante il catalogo degli elementi del programma (cartella Combinazione binaria).

Occupare l'ingresso S (Imposta) con un box AND, l'ingresso R (Resetta) con un box OR.

View ...

- ...
- Finestra con ...
- ...
- Rappresentazione simbolica Ctrl+Q
- Informazioni sul simbolo Ctrl+Shift+Q
- Scelta di simboli Ctrl+7
- Commento Ctrl+Shift+K
- Identificativo operando

Verificare se la rappresentazione simbolica è attivata.



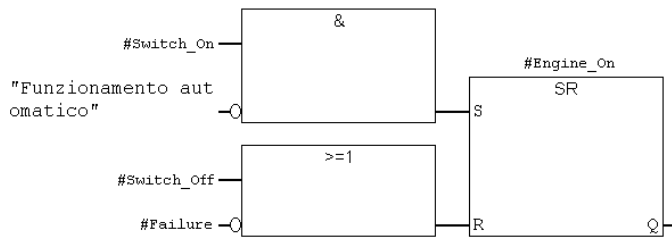


Fare clic su ??? e introdurre i nomi corrispondenti della tabella di dichiarazione (# viene assegnato automaticamente).

Fare attenzione che un ingresso della funzione AND venga indirizzato con il nome simbolico "Funzionamento automatico".

Negare ancora gli ingressi "Funzionamento automatico" e #Guasto con il simbolo corrispondente (pulsante della barra degli strumenti).

Salvare il programma.



Le variabili locali di blocco vengono identificate con # e sono valide solo nel blocco.

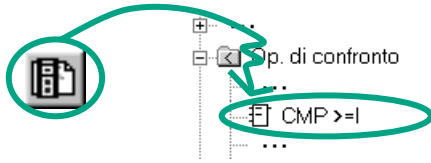
Le variabili globali sono racchiuse tra virgolette. Vengono definite nella tabella dei simboli e sono valide nell'intero programma.

Lo stato di segnale "Funzionamento automatico" viene definito nell'OB1 (segmento 3, vedere pagina 4-14) da un'altra funzione SR e interrogato ora nell'FB1.





Come programmare il controllo del numero di giri

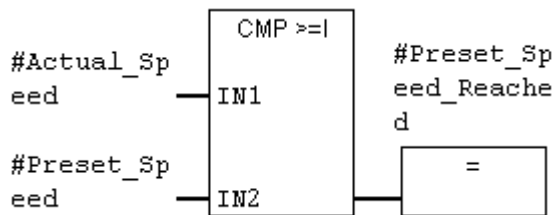


Inserire un nuovo segmento e selezionare l'area di introduzione.

Nel catalogo degli elementi del programma navigare a **Op. di confronto** ed inserire un **CMP>=I**.

Aggiungere un'assegnazione di uscita all'operazione di confronto ed indirizzare gli ingressi con i nomi presi dalla dichiarazione delle variabili.

Salvare infine il programma.



Quando viene inserito e disinserto il motore?

Se la variabile #Inserisci ha lo stato di segnale "1" e (AND) la variabile "Funzionamento automatico" ha lo stato di segnale "0", il motore viene inserito. Solo la negazione (contatto normalmente chiuso) di "Funzionamento automatico" permette questa funzione.

Se la variabile #Disinserisci ha lo stato di segnale "1" o (OR) la variabile #Guasto ha lo stato di segnale "0", il motore viene disinserto. La funzionalità desiderata viene di nuovo conseguita con la negazione di #Guasto (#Guasto è un segnale "zeroattivo" e in caso normale ha valore "1", in caso di guasto ha valore "0").

Come controlla l'operazione di confronto il numero di giri del motore?

Mediante l'operazione di confronto vengono confrontate le variabili #Actual_Speed e #Preset_Speed e viene assegnato il risultato delle variabili #Preset_Speed_Reached (stato di segnale 1).

Per maggiori informazioni vedere ? > **Argomenti della Guida** "Programmazione di blocchi", "Creazione di blocchi di codice" e "Editazione di dichiarazione delle variabili" o "Editazione di istruzioni FUP".

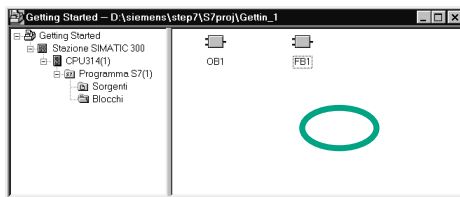
5.5 Come creare i blocchi dati di istanza e modificare i valori attuali

Abbiamo appena programmato il blocco funzionale FB1 ("Motore") e definito, tra le altre cose, i parametri specifici del motore nella tabella di dichiarazione delle variabili.

Per poter programmare successivamente il richiamo (CALL) dell'FB nell'OB1, è necessario creare il relativo blocco dati. Ad un FB è sempre assegnato un blocco dati di istanza (DB).

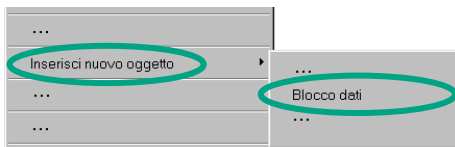
L'FB deve comandare e controllare un motore a benzina o un motore diesel. I diversi numeri di giri prefissati dei motori vengono memorizzati in due DB separati modificando il valore attuale corrispondente (#Numero giri_prefissato).

Programmando solo una volta in modo centrale il blocco funzionale, si riducono i tempi di programmazione.

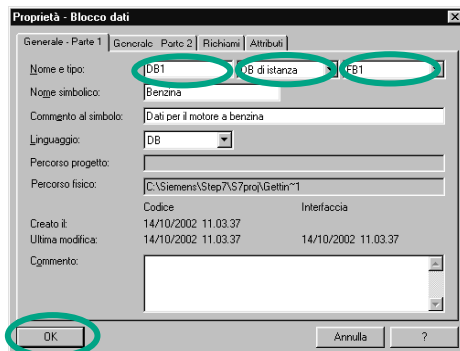


Nel SIMATIC Manager è aperto il progetto "Getting Started".

Navigare alla cartella **Blocchi** e fare clic nella parte destra della finestra con il tasto destro del mouse.



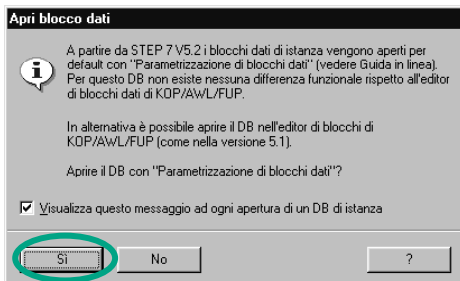
Con il menu di scelta rapida del tasto destro del mouse inserire un **blocco dati**.



Nella finestra di dialogo "Proprietà – Blocco dati" acquisire il nome DB1, scegliere nella casella di riepilogo la voce "DB di istanza" ed acquisire il nome del blocco funzionale "FB1" assegnato. Confermare le preimpostazioni con **OK**.

Il blocco dati DB1 viene aggiunto nel progetto "Getting Started".

Aprire il **DB1** con doppio clic.



Confermare la finestra di dialogo visualizzata con **Sì** per parametrizzare il blocco dati di istanza.

	Indirizzo	Dichiarazion	Nome	Tipo	Valore iniziale	Valore attuale	Commento
1	0.0	in	Switch_On	BOOL	FALSE	FALSE	Imparisci motore
2	0.1	in	Switch_Off	BOOL	FALSE	FALSE	Disinserisci motore
3	0.2	in	Failure	BOOL	FALSE	FALSE	Disturbo del motore, causa la disconnessione
4	2.0	in	Actual_Speed	INT	0	0	Numero di giri effettivo del motore
5	4.0	out	Engine_On	BOOL	FALSE	FALSE	Il motore viene inserito
6	4.1	out	Presst_Speed_Reached	BOOL	FALSE	FALSE	Numero di giri prefissato raggiunto
7	6.0	stat	Presst_Speed	INT	1500	1500	Numero di giri del motore richiesto

Introdurre ora per il motore a benzina nella colonna Valore attuale il valore "1500" (nella riga "Numero giri_prefissato"). In tal modo è stato definito il numero di giri massimo per questo motore.

Salvare il DB1, e chiudere la finestra del programma.

A questo punto creare analogamente al DB1 un altro blocco dati DB2 per l'FB1.

Introdurre ora per il motore diesel il valore attuale "1200".

Salvare il DB2, e chiudere la finestra del programma.

	Indirizzo	Dichiarazion	Nome	Tipo	Valore iniziale	Valore attuale	Commento
1	0.0	in	Switch_On	BOOL	FALSE	FALSE	Imparisci motore
2	0.1	in	Switch_Off	BOOL	FALSE	FALSE	Disinserisci motore
3	0.2	in	Failure	BOOL	FALSE	FALSE	Disturbo del motore, causa la disconnessione
4	2.0	in	Actual_Speed	INT	0	0	Numero di giri effettivo del motore
5	4.0	out	Engine_On	BOOL	FALSE	FALSE	Il motore viene inserito
6	4.1	out	Presst_Speed_Reached	BOOL	FALSE	FALSE	Numero di giri prefissato raggiunto
7	6.0	stat	Presst_Speed	INT	1500	1200	Numero di giri del motore richiesto



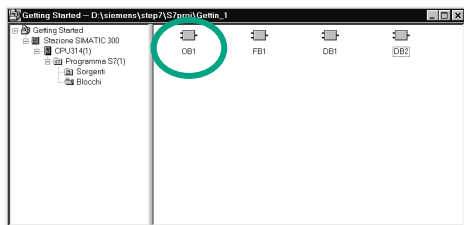
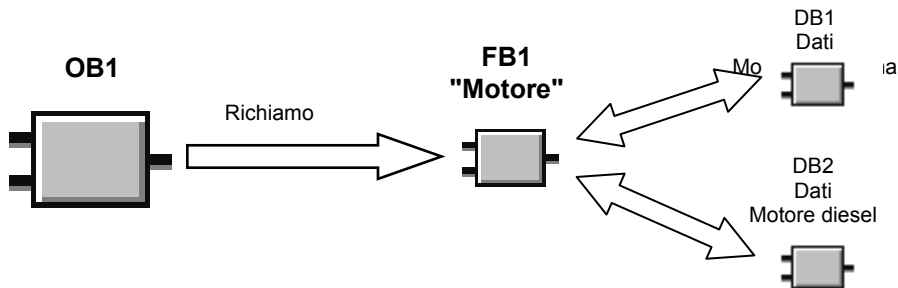
Modificando i valori attuali sono terminate le operazioni preliminari per comandare due motori con un solo blocco funzionale. Per comandare altri motori è sufficiente creare altri blocchi dati.

Per programmare ora il richiamo dell'FB nell'OB1, leggere a seconda del linguaggio di programmazione utilizzato il paragrafo 5.6 per KOP, il paragrafo 5.7 per AWL e il paragrafo 5.8 per FUP.

Per maggiori informazioni vedere ? > **Argomenti della Guida** "Programmazione di blocchi" e "Creazione di blocchi dati".

5.6 Come programmare un richiamo di blocco in KOP

L'intera programmazione di un FB non è efficace senza il richiamo nell'OB1. Per ciascun richiamo dell'FB viene utilizzato un blocco dati; vengono così comandati i due motori.

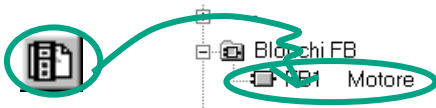


Il SIMATIC Manager è aperto con il suo progetto "Getting Started".

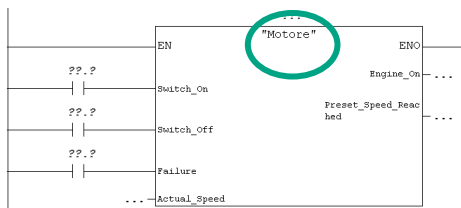
Navigare alla cartella **Blocchi**, ed aprire l'**OB1**.



Nella finestra del programma KOP/AWL/FUP selezionare il segmento 3 ed inserire il segmento 4.

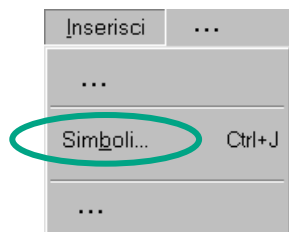


Nel catalogo degli elementi del programma navigare infine all'**FB1** ed inserirlo.



Inserire un contatto normalmente aperto prima di Inserisci, Disinserisci e Guasto.

Fare clic sui ??? sopra "Motore", e subito dopo fare clic con il tasto destro del mouse nella casella di editazione.

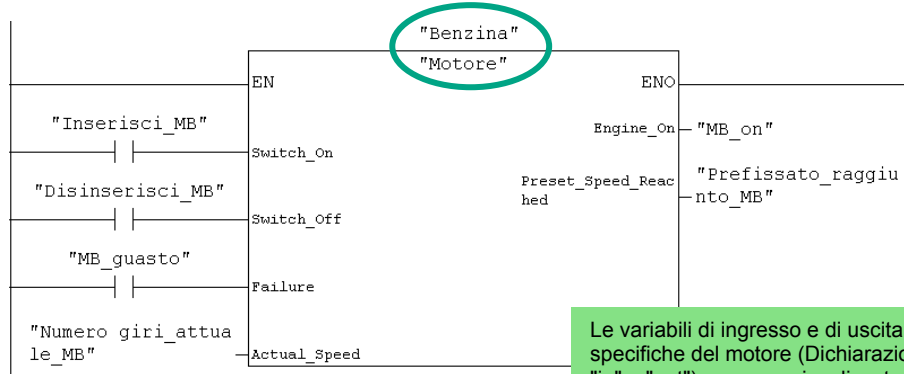


Nel menu di scelta rapida del tasto destro del mouse fare clic su **Inserisci simboli**. Si apre una casella di riepilogo.

Benzina	FB	DB	1	Dati per il motore
---------	----	----	---	--------------------

Fare doppio clic sul blocco dati **Benzina**. Esso viene acquisito automaticamente nella casella di editazione e posto tra virgolette.

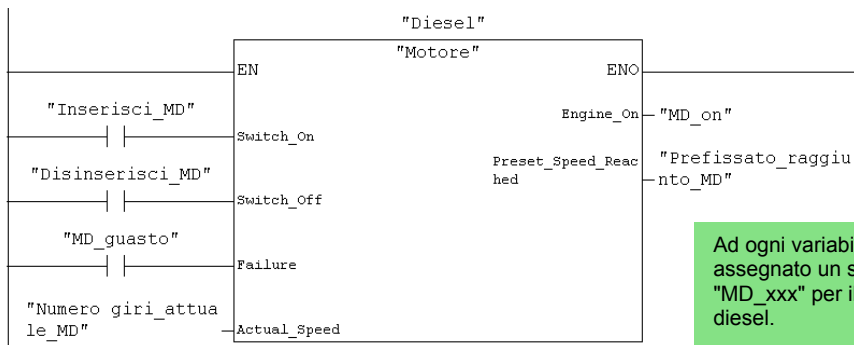
Fare clic sui punti interrogativi e, dopo aver digitato le virgolette, indirizzare tutti gli altri parametri del blocco funzionale con i nomi simbolici corrispondenti utilizzando la casella di riepilogo.



Le variabili di ingresso e di uscita specifiche del motore (Dichiarazione "in" e "out") vengono visualizzate nell'FB "Motore".

A ogni variabile viene assegnato un segnale "MB_XXX" per il motore a benzina.

Programmare in un nuovo segmento il richiamo del blocco funzionale "Motore" (FB1) con il blocco dati "Diesel" (DB2) e acquisire i relativi indirizzi dalla casella di riepilogo.



Ad ogni variabile viene assegnato un segnale "MD_xxx" per il motore diesel.



Salvare il programma e chiudere il blocco.

Se vengono create strutture di programma con OB, FB e DB si dovrebbe programmare il richiamo di un blocco subordinato (p.es. FB1) nel blocco sovraordinato (p.es. OB1). Il modo di procedere è sempre lo stesso.

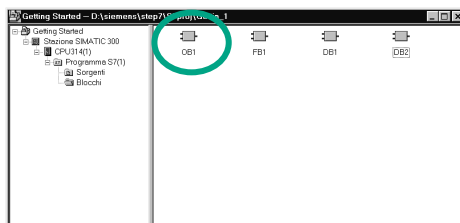
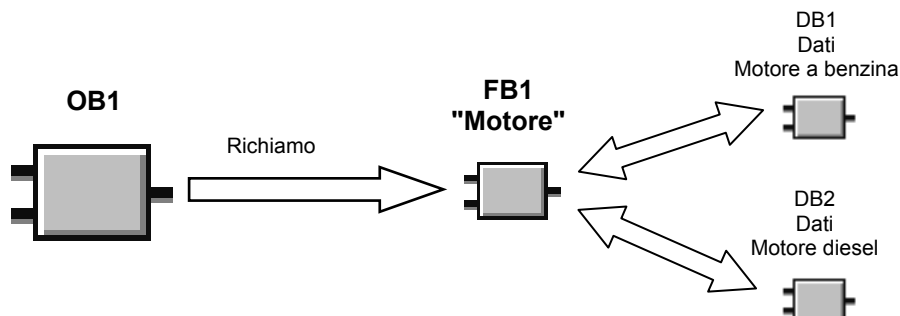
Nella tabella dei simboli si possono anche assegnare nomi simbolici a diversi blocchi (p.es. FB1 ha il nome "Motore" e DB1 il nome "Benzina").

E' possibile in qualsiasi momento archiviare o stampare i blocchi programmati. Le funzioni corrispondenti possono essere richiamate nel SIMATIC Manager mediante i comandi di menu **File > Archivia** o **File > Stampa**.

Per maggiori informazioni vedere ? > **Argomenti della Guida** "Richiamo delle Guide di riferimento", "Linguaggio di programmazione KOP" e "Operazioni di controllo del programma".

5.7 Come programmare un richiamo di blocco in AWL

L'intera programmazione di un blocco funzionale non è efficace senza il richiamo nell'OB1. Per ciascun richiamo del blocco funzionale viene utilizzato un blocco dati e vengono così comandati i due motori.



Il SIMATIC Manager è aperto con il suo progetto "Getting Started".

Navigare alla cartella **Blocchi**, ed aprire l'OB1.



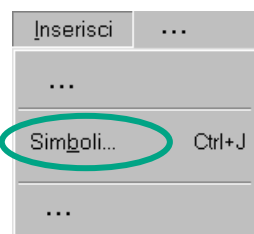
Nella finestra del programma KOP/AWL/FUP selezionare il segmento 3 e inserire il segmento 4.

```
CALL "Motore" , "Benzina"
Switch_On      :=
Switch_Off     :=
Failure        :=
Actual_Speed   :=
Engine_On      :=
Preset_Speed_Reached :=
```

Scrivere nella parte istruzioni **CALL "Motore", "Benzina"** e premere **Invio**.

Vengono visualizzati tutti i parametri del blocco funzionale "Benzina".

Posizionare il cursore dopo il segno di uguale di Inserisci e premere il tasto destro del mouse.



Fare clic nel menu di scelta rapida del tasto destro del mouse su **Inserisci simboli**. Si apre una casella di riepilogo.



Disinserisci_MD	BOOL	E	1
Funzionamento automatico	BOOL	A	4
Inserisci_MB	BOOL	E	1
Inserisci_MD	BOOL	E	1
Lampadina rossa	BOOL	A	4

Fare clic sul nome **Inserisci_MB**. Esso viene acquisito automaticamente dalla casella di riepilogo e posto tra virgolette.

```
CALL "Motore" , "Benzina"
Switch_On      := "Inserisci_MB"
Switch_Off     := "Disinserisci_MB"
Failure        := "MB_guasto"
Actual_Speed   := "Numero giri_attuale_MB"
Engine_On      := "MB_on"
Preset_Speed_Reached := "Prefissato_raggiunto_MB"
```

Mediante la casella di riepilogo assegnare tutti gli indirizzi necessari alle variabili del blocco funzionale.

A ogni variabile viene assegnato un segnale "MB_xxx" per il motore a benzina.

```
CALL "Motore" , "Diesel"
Switch_On      := "Inserisci_MD"
Switch_Off     := "Disinserisci_MD"
Failure        := "Disinserisci_MD"
Actual_Speed   := "Numero giri_attuale_MD"
Engine_On      := "MD_on"
Preset_Speed_Reached := "Prefissato_raggiunto_MD"
```

In un nuovo segmento programmare il richiamo del blocco funzionale "Motore" (FB1) con il blocco dati "Diesel" (DB2). Procedere come per il richiamo sopra indicato.



Salvare il programma e chiudere il blocco.

Se si creano strutture di programma con OB, FB e DB si dovrebbe programmare il richiamo di un blocco subordinato (p.es. FB1) nel blocco sovraordinato (p.es. OB1). Il modo di procedere è sempre lo stesso.

Nella tabella dei simboli si possono anche assegnare nomi simbolici a diversi blocchi (p.es. FB1 ha il nome "Motore" e DB1 il nome "Benzina").

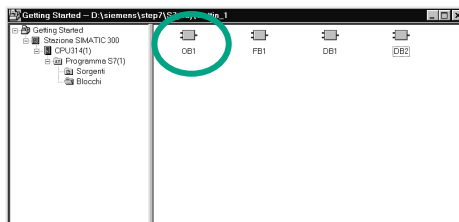
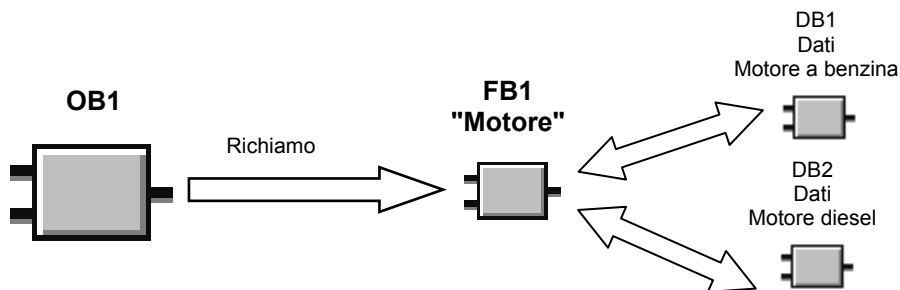
E' possibile in qualsiasi momento archiviare o stampare i blocchi programmati. Le funzioni corrispondenti possono essere richiamate dal SIMATIC con i comandi di menu

File > Archivia o **File > Stampa**.

Per maggiori informazioni vedere ? > **Argomenti** "Richiamo delle Guide di riferimento", "Descrizione del linguaggio AWL" e "Controllo del programma".

5.8 Come programmare un richiamo di blocco in FUP

L'intera programmazione di un blocco funzionale non è efficace senza il richiamo nell'OB1. Per ciascun richiamo del blocco funzionale viene utilizzato un blocco dati e vengono così comandati i due motori.



Il SIMATIC Manager è aperto con il suo progetto "Getting Started".

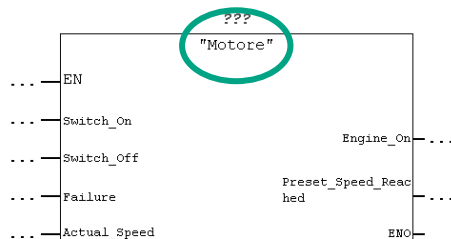
Navigare alla cartella **Blocchi**, ed aprire l'**OB1**.



Nella finestra del programma KOP/AWL/FUP selezionare il segmento 3 e inserire il segmento 4.

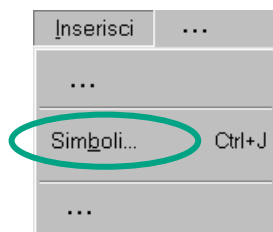


Nel catalogo degli elementi del programma navigare infine all'**FB1** ed inserirlo.



Vengono visualizzate tutte le variabili di ingresso e di uscita specifiche del motore.

Fare clic sui **???** sopra "Motore", e subito dopo fare clic con il tasto destro del mouse nella casella di editazione.



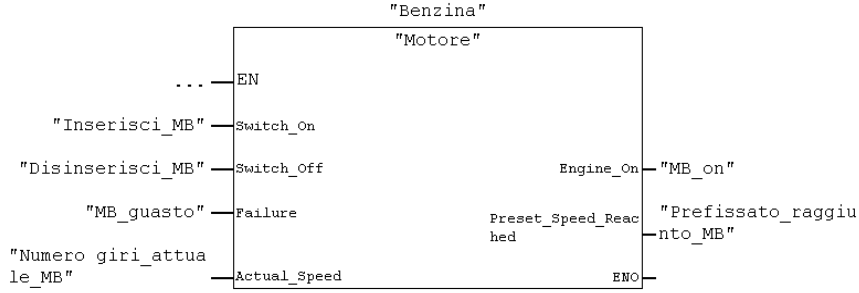
Nel menu di scelta rapida del tasto destro del mouse fare clic su **Inserisci simboli**. Si apre una casella di riepilogo. Quando viene effettuata per la prima volta, questa operazione richiede un certo tempo.



Benzina	FB	1	DB	1	Dati per il m
---------	----	---	----	---	---------------

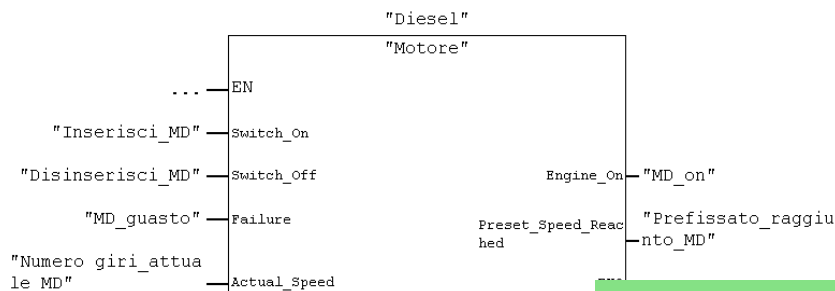
Fare doppio clic sul blocco dati **Benzina**. Esso viene trasferito automaticamente dalla casella di riepilogo alla casella di editazione e posto tra virgolette.

Mediante la casella di riepilogo indirizzare tutti gli altri parametri del blocco funzionale con i nomi simbolici corrispondenti.



A ogni variabile viene assegnato un segnale "MB_xxx" per il motore a benzina.

Programmare in un nuovo segmento il richiamo del blocco funzionale "Motore" (FB1) con il blocco dati "Diesel" (DB2) e acquisire i relativi indirizzi dalla casella di riepilogo.



Ad ogni variabile viene assegnato un segnale "MD_xxx" per il motore diesel.



Salvare il programma e chiudere il blocco.

Se si creano strutture di programma con OB, FB e DB si dovrebbe programmare il richiamo di un blocco subordinato (p.es. FB1) nel blocco sovraordinato (p.es. OB1). Il modo di procedere è sempre lo stesso.

Nella tabella dei simboli si possono anche assegnare nomi simbolici ai diversi blocchi (p.es. FB1 ha il nome "Motore" e DB1 il nome "Benzina").

E' possibile in qualsiasi momento archiviare o stampare i blocchi programmati. Le funzioni corrispondenti possono essere richiamate nel SIMATIC Manager con i comandi di menu

File > Archivia o **File > Stampa**.

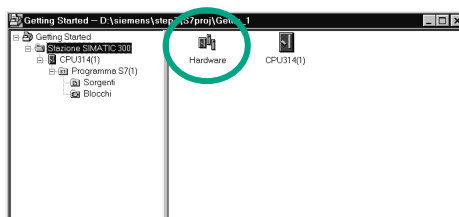
Per maggiori informazioni vedere ? > **Argomenti della Guida** "Richiamo delle Guide di riferimento", "Descrizione del linguaggio FUP" e "Operazioni di controllo del programma".

6 Configurazione delle unità centrali

6.1 Come configurare l'hardware

L'hardware può essere configurato se è stato creato un progetto con una stazione SIMATIC. La struttura del progetto creata con l' "Assistente di STEP 7" al capitolo 2.1 soddisfa tutti i presupposti.

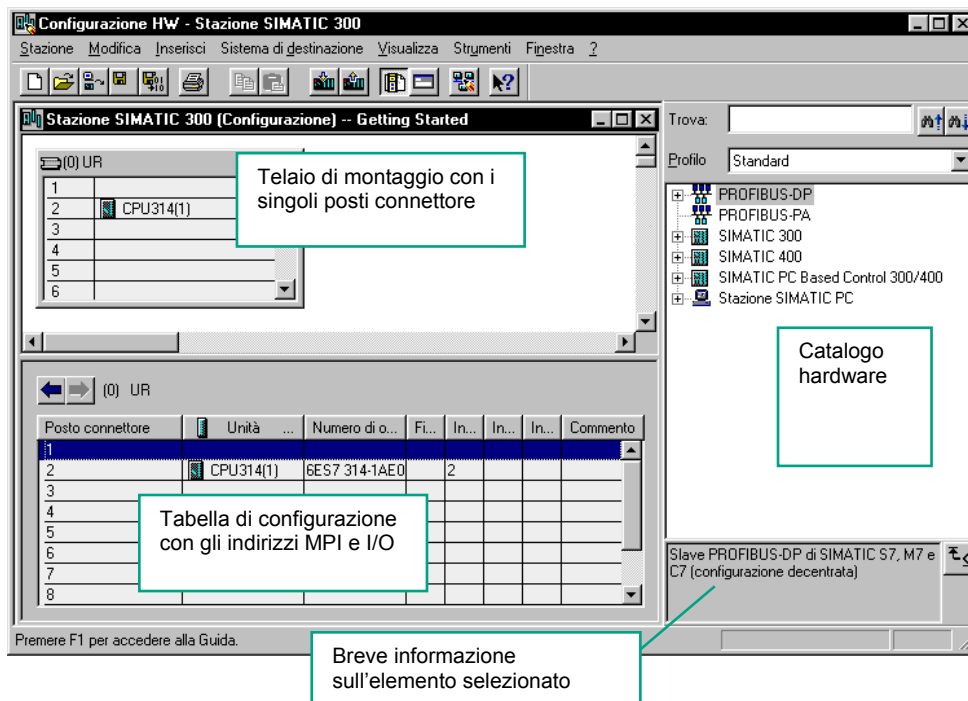
L'hardware viene configurato con STEP 7. I dati di configurazione vengono trasferiti al sistema di automazione successivamente durante il "Caricamento" (vedere il capitolo 7).

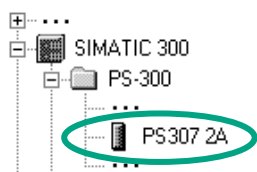


Punto di partenza è il SIMATIC Manager aperto, con il progetto "Getting Started".

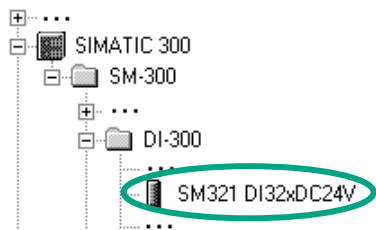
Aprire la cartella **Stazione SIMATIC 300**, e fare doppio clic sul simbolo **Hardware**.

Si apre la finestra "Configurazione HW" e viene visualizzata la CPU scelta durante la creazione del progetto. Per il "Getting Started" si tratta della CPU314.

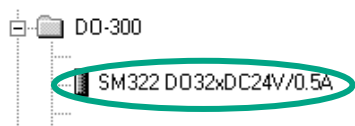




E' necessario inserire prima di tutto un alimentatore. Navigando nel catalogo, posizionarsi su **PS307 2A** ed inserire l'alimentatore sul posto connettore 1 mediante Drag & Drop.



Per un'unità di ingresso (DI, Digital Input) navigare a **SM321 DI32xDC24V** ed inserirla sul posto connettore 4. Il posto connettore 3 rimane libero.

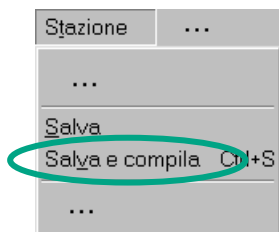


Analogamente inserire sul posto connettore 5 l'unità di uscita **SM322 DO32xDC24V/0.5A**.

Per modificare i parametri (p.es. Indirizzo) di un'unità all'interno del progetto, aprirla con doppio clic. È opportuno modificare i parametri solo se si è certi degli effetti che le modifiche producono sul PLC.

Per il progetto "Getting Started" non sono necessarie modifiche.

Connettore	Unità	Numero di ordinazione	Indirizzo MPI	Indirizzo E	Indirizzo...	Commento
1	PS307 2A	6ES7 307-1BA00-0AA0				
2	CPU314(1)	6ES7 314-1AE04-0AB0	2			
3						
4	DI32xDC24V	6ES7 321-1BL00-0AA0		0...3		
5	DO32xDC24V/0.5A	6ES7 322-1BL00-0AA0			4..7	
6						
7						
8						
9						
10						
11						



Mediante **Salva e compila** i dati vengono preparati per il trasferimento nella CPU.

Una volta chiusa la "Configurazione HW", nella cartella Blocchi viene visualizzato il simbolo Dati di sistema.



Con il comando di menu **Stazione > Verifica coerenza** è inoltre possibile controllare se vi sono errori nella configurazione. In caso di eventuali errori STEP 7 offre diverse soluzioni.

Per maggiori informazioni vedere ? > **Argomenti della Guida** "Configurazione dell'hardware" e "Configurazione delle unità centrali".

7 Caricamento e test del programma

7.1 Come stabilire il collegamento online

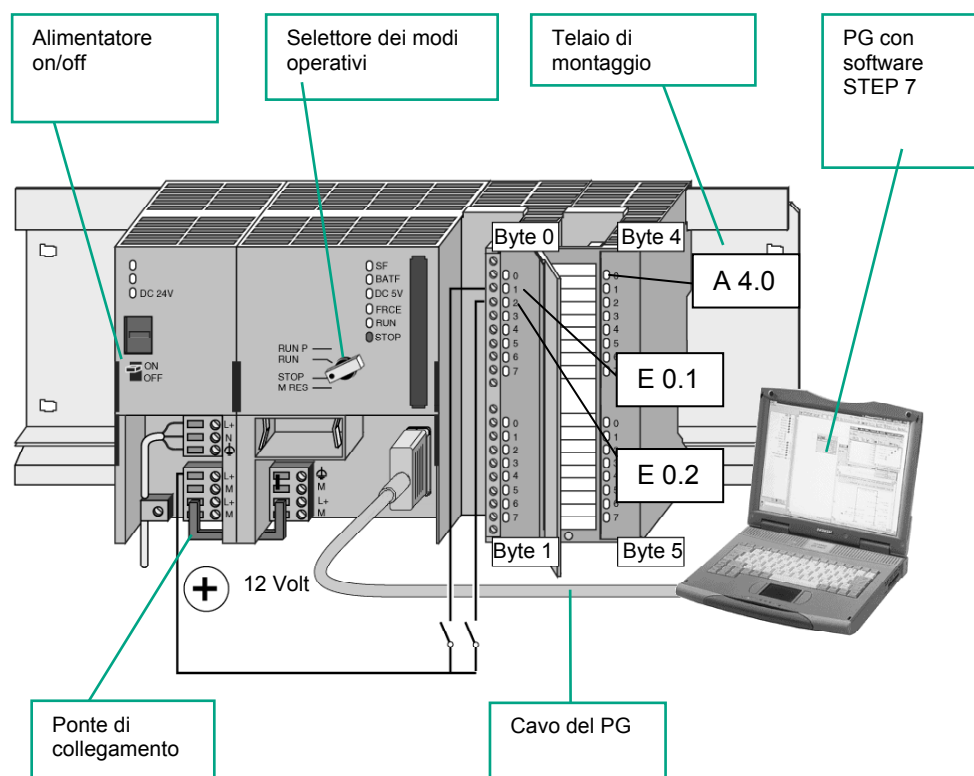
Sulla base del progetto "GS-KOP_Esempio" in dotazione o del "Getting Started" finora creato e una semplice configurazione di prova Vi mostriamo come caricare il programma nel sistema di automazione e come testarlo.

È necessario:

- avere configurato l'hardware per il "Getting Started" (vedere il capitolo 6)
- configurare l'hardware come indicato nel manuale.

Esempio per il circuito in serie (funzione AND):

L'uscita A 4.0 si deve accendere (sull'unità uscite digitali si illumina il diodo A 4.0) solo quando il pulsante E 0.1 e il pulsante E 0.2 sono premuti. Realizzare la configurazione di prova con l'ausilio dei cavi e della CPU.





Come configurare l'hardware

Per montare un'unità sulla guida profilata procedere nel modo seguente.

- Innestare l'unità sul connettore di bus
- Agganciare l'unità e ruotarla verso il basso
- Fissare a vite l'unità
- Montare le unità rimanenti
- Dopo aver montato tutte le unità, inserire la chiave nella CPU

Il test può anche essere eseguito con un hardware diverso da quello illustrato. È sufficiente rispettare l'indirizzamento degli ingressi e delle uscite.

STEP 7 offre diverse possibilità di test, per es. mediante lo stato del programma o la tabella delle variabili.

Per maggiori informazioni sulla configurazione delle unità centrali consultare i manuali "S7-300 – Configurazione e dati della CPU" e "S7-400/M7-400 – Configurazione"

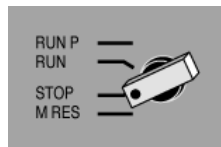
7.2 Caricamento del programma nel sistema di destinazione

Il presupposto per il caricamento del programma è la presenza del collegamento online.

Come applicare la tensione

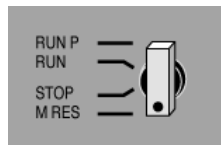


Inserire la rete sull'interruttore ON/OFF. Si accende il diodo "DC 5V" sulla CPU.



Ruotare il selettore dei modi operativi su STOP (a meno che non sia già su STOP). Si accende il LED rosso "STOP".

Come eseguire la cancellazione totale della CPU e commutarla in RUN



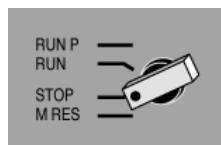
Ruotare il selettore dei modi operativi in posizione **MRES** e tenerlo premuto per almeno 3 sec. finché il LED rosso "STOP" non lampeggia lentamente.

Rilasciare il selettore dei modi operativi dopo max. 3 sec. ruotarlo di nuovo in posizione **MRES**. Quando il LED "STOP" lampeggia velocemente è stata eseguita la cancellazione totale della CPU.

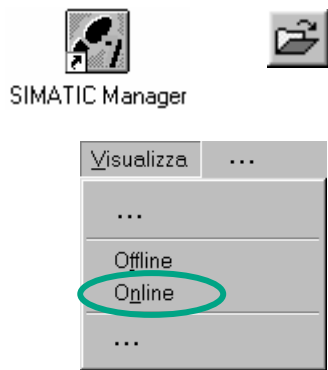
Se "STOP" non lampeggia velocemente, ripetere l'operazione.

La cancellazione totale cancella tutti i dati della CPU. La CPU si trova ora nello stato iniziale.

Come caricare il programma sulla CPU

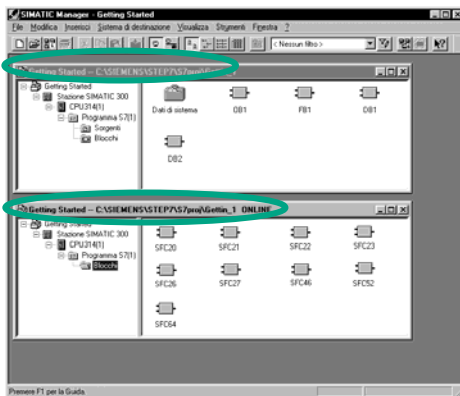


Per il caricamento del programma ruotare ora di nuovo il selettore dei modi operativi su "STOP".



Avviare il SIMATIC Manager e aprire il progetto "Getting Started" (a meno che non sia già aperto) mediante la finestra di dialogo "Apri".

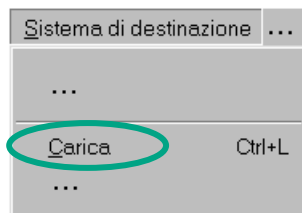
Oltre alla finestra "Getting Started Offline" richiamare la finestra "Getting Started Online". Lo stato Offline/Online è identificato da un cambiamento di colore nella riga di intestazione.



Nelle due finestre navigare alla cartella **Blocchi**.

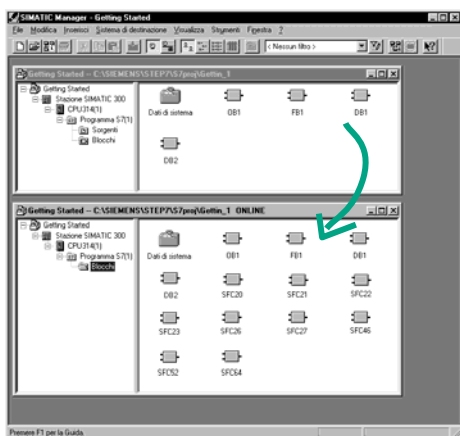
La finestra "Offline" mostra la situazione sul PG, la finestra "Online" mostra la situazione sulla CPU.

Malgrado la cancellazione totale, nella CPU si trovano le funzioni di sistema (SFC). Queste funzioni del sistema operativo sono messe a disposizione dalla CPU. Esse non devono essere caricate, ma non possono nemmeno essere cancellate.



Selezionare nella finestra "Offline" la cartella **Blocchi** e caricare il programma nella CPU tramite **Sistema di destinazione > Carica**.

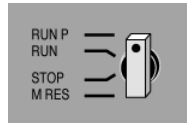
Confermare l'interrogazione con **OK**.



Nella finestra "Online" vengono visualizzati, dopo il caricamento, i blocchi del programma.

Il comando di menu **Sistema di destinazione > Carica** può essere richiamato anche mediante il simbolo corrispondente nella barra degli strumenti oppure mediante il menu di scelta rapida del tasto destro del mouse

Come accendere la CPU e controllare lo stato di funzionamento



Girare il selettore di modi operativi su **RUN-P**. Il LED verde "RUN" è acceso e si spegne il LED rosso "STOP". La CPU è pronta al funzionamento.

Se è acceso il LED verde, è possibile iniziare con il test del programma.

Se il LED rosso continua ad essere acceso, si è verificato un errore. Per una diagnostica dell'errore è necessario analizzare il buffer di diagnostica.

Caricamento dei singoli blocchi

Per poter reagire prontamente agli errori, i singoli blocchi possono essere trasferiti mediante Drag & Drop nella CPU.

Durante il caricamento dei blocchi il selettore dei modi operativi della CPU deve essere impostato o su "RUN-P" o su "STOP". I blocchi caricati nello stato di funzionamento "RUN" vengono attivati immediatamente. Si tenga presente che:

- se blocchi corretti vengono sovrascritti da blocchi errati, ciò può dare luogo a un funzionamento errato dell'impianto. Per prevenire questa situazione, testare i blocchi prima di caricarli;
- se non è stato rispettato l'ordine di caricamento dei blocchi – vengono caricati prima i blocchi inferiori e poi quelli superiori – la CPU passa allo stato di funzionamento "STOP". Per prevenire questa situazione, caricare l'intero programma sulla CPU.

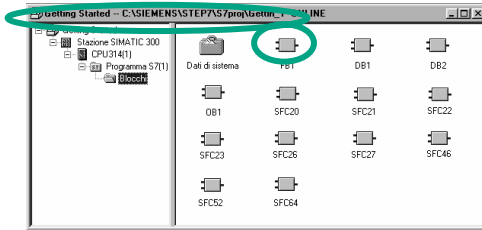
Come programmare online

Per finalità di test può essere necessario, nel caso pratico, modificare blocchi che sono già stati caricati nella CPU. Fare doppio clic sul blocco desiderato nella finestra "Online" per aprire la finestra del programma KOP/AWL/FUP e programmare il blocco secondo le consuete modalità. Tenere presente che il blocco programmato viene attivato immediatamente nella CPU.

Per maggiori informazioni vedere ? >
Argomenti della Guida "Caricamento" e
 "Attivazione di collegamenti online e
 impostazione della CPU".

7.3 Come testare il programma controllandolo

Mediante la funzione di controllo dello stato del programma viene testato il programma di un blocco. Presupposto è che vi sia un collegamento online con la CPU, la CPU si trovi in RUN o RUN-P ed il programma sia stato caricato nella CPU.



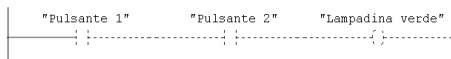
Aprire l'OB1 nella finestra di progetto "Getting Started Online".

Viene aperta la finestra del programma KOP/AWL/FUP.



Attivare la funzione **Test > Controlla**.

Come testare con KOP



Viene visualizzato il circuito in serie nel segmento 1 in KOP. Il montante viene rappresentato come linea continua fino al pulsante 1 (E 0.1), il che significa che qui è applicata la tensione.

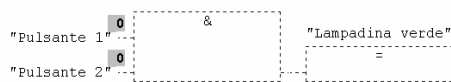
Come testare con AWL

	RLC	STA	STANDARD
U "Pulsante 1"	0	0	0
U "Pulsante 2"	0	0	0
= "Lampadina verde"	0	0	0

Per AWL

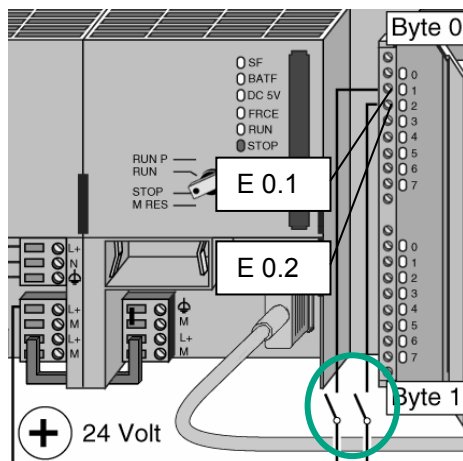
- il risultato logico combinatorio (RLC)
 - il bit di stato (STA)
 - lo stato standard (STANDARD)
- vengono visualizzati in forma tabellare.

Come testare con FUP



Lo stato del segnale viene identificato con "0" e "1". La linea tratteggiata significa che non vi sono risultati logici combinatori.

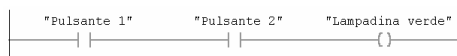
Mediante **Strumenti > Impostazioni** è possibile modificare il modo di rappresentazione del linguaggio di programmazione durante il test.



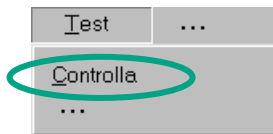
Collegare ora i due pulsanti alla configurazione da testare.

Sull'unità di ingresso si accendono i diodi per gli ingressi E 0.1 e E 0.2.

Sull'unità di uscita il diodo per l'uscita A 4.0.



	RLC	STA	STANDARD	
U	1	1		0
U	1	1		0
=	1	1		0



Nei linguaggi di programmazione grafici KOP / FUP è possibile seguire l'andamento del risultato del test dal cambiamento di colore nel segmento programmato. Il cambiamento di colore sta ad indicare che il risultato logico combinatorio è fino a questo punto soddisfatto.

Nel linguaggio di programmazione AWL cambia la visualizzazione nella colonna STA e nella colonna RLC in caso di risultato logico combinatorio soddisfatto.

Disattivare **Test > Controlla** e chiudere la finestra.

Chiudere nel SIMATIC Manager la finestra "Online".

Si raccomanda di non caricare ed eseguire mai completamente programmi voluminosi nella CPU, poiché la diagnostica degli errori risulta più complessa a causa delle molteplici fonti di errore. Per una maggiore chiarezza si dovrebbero caricare separatamente i singoli blocchi e testarli.

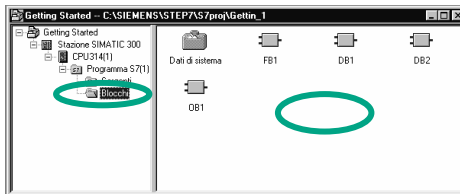
Per maggiori informazioni vedere ? > **Argomenti della Guida "Test" e "Test con lo stato di programma"**

7.4 Come testare il programma con la tabella delle variabili

Le singole variabili del programma vengono testate quando vengono controllate e comandate. Presupposto è che vi sia un collegamento online con la CPU, che quest'ultima si trovi in stato di funzionamento RUN-P e che il programma sia stato caricato.

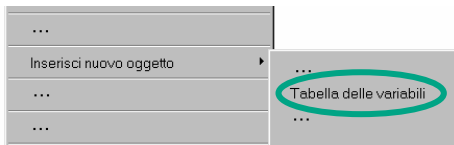
Come durante il test con lo stato del programma, è possibile controllare gli ingressi e le uscite del segmento 1 (circuiti di serie o funzione AND) nella tabella delle variabili. Preimpostando poi un numero di giri ATTUALE è possibile testare l'operazione di confronto per il numero di giri del motore nell'FB1.

Come creare la tabella delle variabili

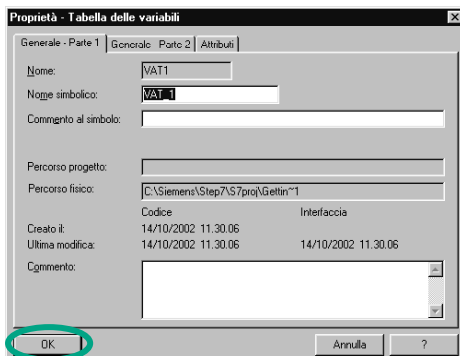


Punto di partenza è di nuovo il SIMATIC Manager con il progetto "Getting Started Offline" aperto.

Navigare alla cartella **Blocchi**, e con il tasto destro del mouse fare clic nella parte destra della finestra.

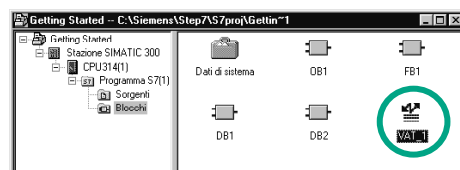


Con il menu di scelta rapida del tasto destro del mouse inserire la **Tabella delle variabili**.



Acquisire le preimpostazioni chiudendo la finestra di dialogo "Proprietà" con **OK**.

È possibile assegnare alla tabella delle variabili un nome simbolico e corredarla di un commento al simbolo.



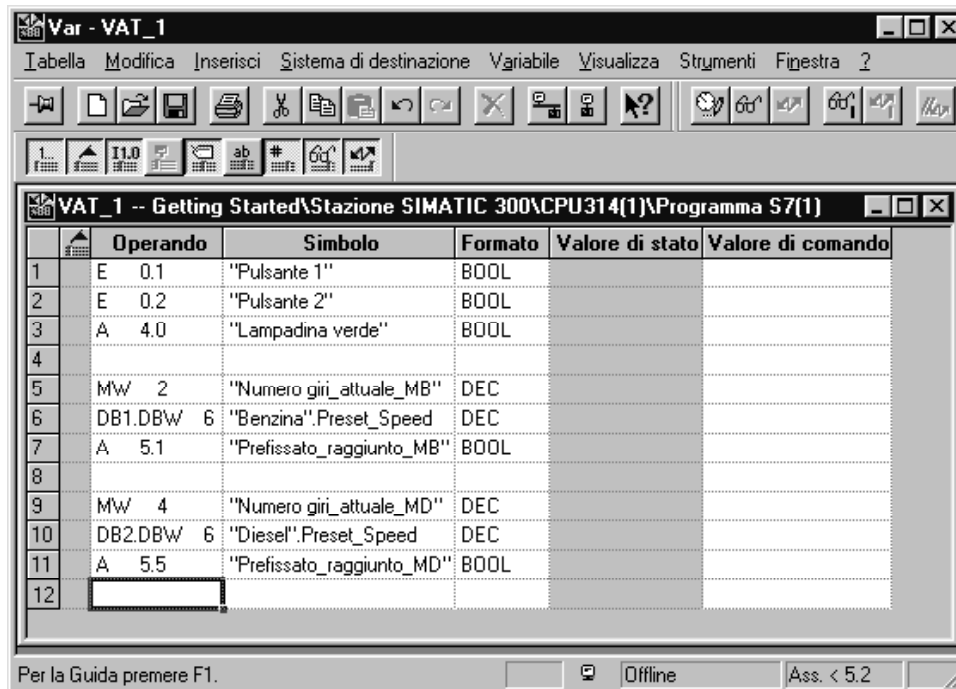
Una VAT1 (tabella delle variabili) viene creata nella cartella Blocchi.

Aperto **VAT1** (con doppio clic) si passa alla finestra "Controlla e comanda variabili".



La tabella delle variabili è ora vuota. Per l'esempio "Getting Started" inserire i nomi simbolici o l'operando secondo l'illustrazione. Le altre informazioni vengono completate terminando una delle introduzioni con Invio.

Impostare per tutti i valori relativi al numero di giri il formato di visualizzazione DEC. Fare clic sulla cella corrispondente e selezionare il formato DEC nel menu di scelta rapida (clic con il tasto destro del mouse).



Salvare la tabella delle variabili.

Collegare online la tabella delle variabili



Creare un collegamento con la CPU progettata. Nella barra di stato viene visualizzato il funzionamento della CPU.



Impostare l'interruttore a chiave della CPU su **RUN-P** (se non è già stato precedentemente impostato).





Come controllare le variabili



Fare clic su **Controlla variabile**.

	Operando	Simbolo	Formato	Valore di stato	Valore di comando
1	E 0.1	"Pulsante 1"	BOOL	true	
2	E 0.2	"Pulsante 2"	BOOL	true	
3	A 4.0	"Lampadina verde"	BOOL	true	
4					
5	MW 2	"Numero giri_attuale_MB"	DEC	0	

Collegare il pulsante 1 ed il pulsante 2 nella configurazione di prova e controllare il risultato nella tabella delle variabili.

I valori di stato nella tabella delle variabili cambiano da false a true.

Come comandare le variabili

Nella colonna Valore di comando per l'operando MW2 introdurre il valore "1500" e per l'operando MW4 il valore "1300".

	Operando	Simbolo	Formato	Valore di stato	Valore di comando
1	E 0.1	"Pulsante 1"	BOOL	true	
2	E 0.2	"Pulsante 2"	BOOL	true	
3	A 4.0	"Lampadina verde"	BOOL	true	
4					
5	MW 2	"Numero giri_attuale_MB"	DEC	0	1500
6	DB1.DBW 6	"Benzina".Preset_Speed	DEC	1500	
7	A 5.1	"Prefissato_raggiunto_MB"	BOOL	true	
8					
9	MW 4	"Numero giri_attuale_MD"	DEC	0	1300
10	DB2.DBW 6	"Diesel".Preset_Speed	DEC	1200	
11	A 5.5	"Prefissato_raggiunto_MD"	BOOL	true	
12					



Trasferire i valori di comando alla CPU.



Dopo il trasferimento questi valori vengono elaborati nella CPU e viene visualizzato il risultato del confronto.

Terminare Controlla variabili e chiudere la finestra. Ad una eventuale interrogazione rispondere **Si** oppure **OK**.

	Operando	Simbolo	Formato	Valore di stato	Valore di comando
1	E 0.1	"Pulsante 1"	BOOL	true	
2	E 0.2	"Pulsante 2"	BOOL	true	
3	A 4.0	"Lampadina verde"	BOOL	true	
4					
5	MW 2	"Numero giri_attuale_MB"	DEC	1500	1500
6	DB1.DBW 6	"Benzina".Preset_Speed	DEC	1500	
7	A 5.1	"Prefissato_raggiunto_MB"	BOOL	true	
8					
9	MW 4	"Numero giri_attuale_MD"	DEC	1300	1300
10	DB2.DBW 6	"Diesel".Preset_Speed	DEC	1200	
11	A 5.5	"Prefissato_raggiunto_MD"	BOOL	true	
12					

MPI = 2(diretto) RUN Ass. < 5.2



Spesso una tabella delle variabili molto voluminosa non può essere visualizzata integralmente a causa dei limiti dello schermo.

Se si dovessero avere tabelle delle variabili voluminose, raccomandiamo di creare con STEP 7 diverse tabelle per un programma S7. Le tabelle delle variabili possono essere adattate alle esigenze di prova.

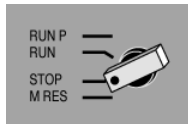
Analogamente ai blocchi, è possibile assegnare nomi individuali alle tabelle delle variabili. (p.es. invece di VAT1 il nome OB1_segmento). L'assegnazione dei nomi avviene mediante la tabella dei simboli.

Per maggiori informazioni vedere ? > **Argomenti della Guida "Test" e "Test con la tabella delle variabili"**.

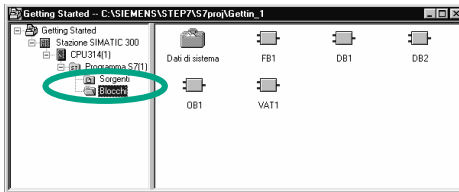
7.5 Come valutare il buffer di diagnostica

Nel caso in cui la CPU dovesse commutare in STOP durante l'elaborazione di un programma S7 oppure la CPU non potesse ritornare allo stato di funzionamento RUN dopo il caricamento del programma, si può dedurre la causa dell'errore dagli eventi elencati nel buffer di diagnostica.

Presupposto è che vi sia un collegamento online con la CPU e la CPU si trovi nello stato di funzionamento "STOP".

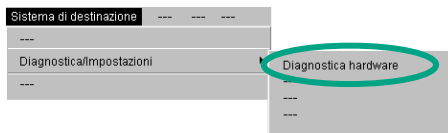


Commutare su "STOP" il selettore dei modi operativi della CPU.

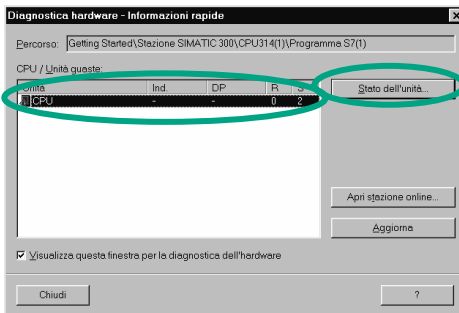


Punto di partenza è di nuovo il SIMATIC Manager con il progetto "Getting Started Offline" aperto.

Selezionare la cartella **Blocchi**.



Se vi sono più CPU nel progetto, verificare prima quale CPU è passata allo stato STOP.



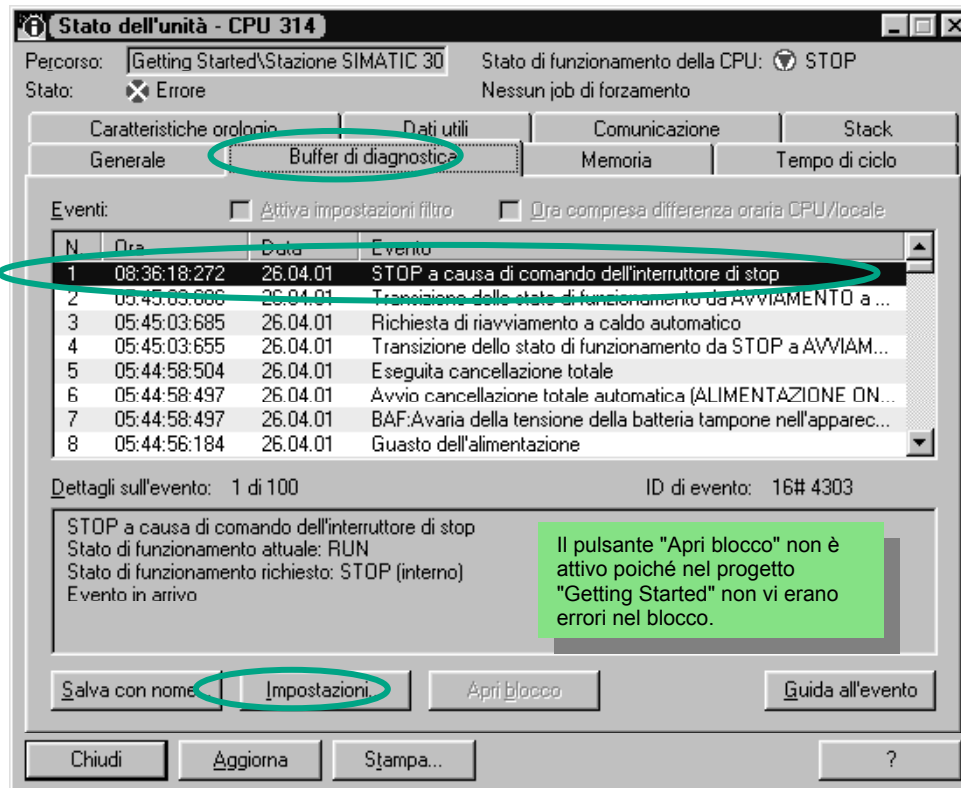
Nella finestra di dialogo "Diagnostica hardware" vengono elencate tutte le CPU accessibili. E' selezionata la CPU con lo stato di funzionamento STOP.

Il progetto "Getting Started" dispone solo di una CPU, che viene visualizzata.

Fare clic su **Stato dell'unità**, per valutare il buffer di diagnostica di questa CPU.

Se è collegata solo una CPU, è possibile interrogare anche direttamente lo stato dell'unità di questa CPU mediante il comando di menu **Sistema di destinazione > Diagnostica/Impostazioni > Stato dell'unità**.

Nella finestra "Stato dell'unità" sono visualizzate le proprietà ed i parametri della Vostra CPU. Scegliere ora la scheda **Buffer di diagnostica** per determinare la causa della commutazione in STOP.



L'evento più recente (n° 1) è il primo in alto. La causa dello STOP viene visualizzata. Chiudere tutte le finestre eccetto quella del SIMATIC Manager.

Se è stato un errore di programmazione a causare lo STOP, selezionare l'evento e fare clic sul pulsante **Apri blocco**.

Il blocco viene aperto nella finestra del programma KOP/AWL/FUP e viene selezionato il segmento errato.

Con questo capitolo abbiamo concluso il progetto di esempio "Getting Started" dalla creazione di un progetto fino al test del programma completo. Nei capitoli seguenti approfondiremo ulteriormente le conoscenze acquisite con esercitazioni selezionate.

Per maggiori informazioni vedere ? > **Argomenti della Guida** "Diagnostica" e "Funzioni di informazione dello stato dell'unità".

8 Programmazione di una funzione (FC)

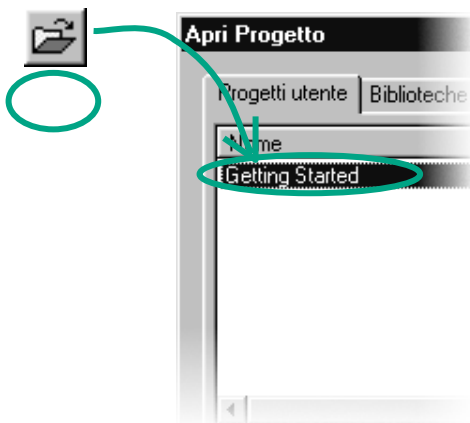
8.1 Come creare ed aprire una funzione

La funzione, così come il blocco funzionale, è subordinata al blocco organizzativo. Affinché la funzione possa essere elaborata dalla CPU, anch'essa deve essere richiamata nel blocco sovraordinato. A differenza del blocco funzionale non è qui necessario alcun blocco dati.

Anche in una funzione i parametri vengono riportati nella tabella di dichiarazione delle variabili, tuttavia non sono ammessi dati locali statici.

La funzione viene programmata analogamente al blocco funzionale nella finestra del programma KOP/FUP/AWL.

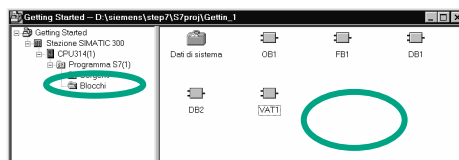
Si dovrebbe avere già dimestichezza con la programmazione in KOP, FUP o AWL (vedere capitoli 4 e 5) nonché con la programmazione con nomi simbolici (vedere capitolo 3).



Se avete eseguito il progetto esemplificativo "Getting Started" capitoli 1–7, aprirlo ora.

Se non lo avete ancora eseguito, creare un nuovo progetto nel SIMATIC Manager con **File > Assistente "Nuovo progetto"**. Procedere come descritto al capitolo 2.1 e nominare il progetto "Funzione Getting Started".

Seguiremo lo svolgimento del progetto "Getting Started". E' tuttavia possibile seguire lo svolgimento del programma passo per passo anche sulla base di un nuovo progetto creato.

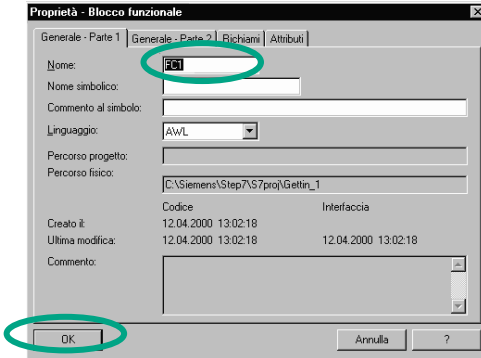


Navigare alla cartella **Blocchi** e aprirla.

Fare clic con il tasto destro del mouse nella parte destra della finestra.

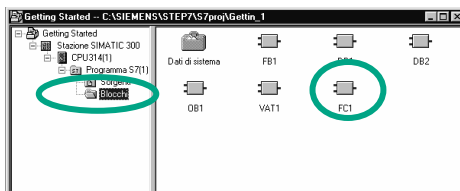


Inserire una **funzione** (FC) tramite il menu di scelta rapida.



Nella finestra di dialogo "Proprietà - Funzione" memorizzare il nome FC1, e selezionare il linguaggio di programmazione desiderato.

Confermare le rimanenti preimpostazioni con **OK**.



La funzione FC1 è stata aggiunta alla cartella Blocchi. Aprire **FC1** con doppio clic.

All'interno di una funzione, a differenza del blocco funzionale, non possono essere definiti dati statici nella tabella di dichiarazione delle variabili.

I dati statici definiti in un blocco funzionale rimangono immutati anche dopo l'elaborazione del blocco. I dati statici sono per esempio i merker utilizzati per i valori limite "Numero di giri" (vedere capitolo 5).

Per la programmazione della funzione si può ricorrere come sempre ai nomi simbolici della tabella dei simboli.

Per maggiori informazioni vedere ? > **Argomenti della Guida** "Elaborazione del progetto di automazione", "Nozioni per lo sviluppo di strutture di programma" e "Blocchi nel programma utente".

8.2 Come programmare la funzione

Nel nostro esempio programmeremo nel seguito una funzione di temporizzazione. La funzione di temporizzazione fa sì che all'accensione di un motore (vedere capitolo 5) viene contemporaneamente inserito anche un ventilatore che continua a girare ancora 4 secondi dopo lo spegnimento del motore (ritardo alla disinserzione).

Come già accennato, i parametri di ingresso e di uscita della funzione (dichiarazione "in" e "out") devono essere indicati nella tabella di dichiarazione delle variabili.

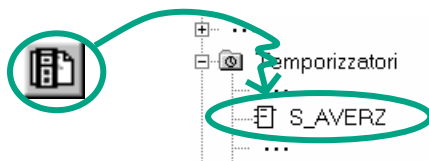
La finestra del programma KOP/AWL/FUP è stata aperta. Lavorare con questa visualizzazione analitica delle variabili come con la visualizzazione analitica delle variabili del blocco funzionale (vedere capitolo 5).

Introdurre le seguenti dichiarazioni.

Contenuto di: 'Ambiente\Interfaccia\IN'			
Nome	Tipo di dati	Commento	
Engine_On	Bool	Segnale per l'inserzione del motore	
Timer_Function	Timer	Funzione di temporizzazione per il ritardo alla disinserzione	

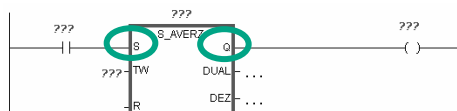
Contenuto di: 'Ambiente\Interfaccia\OUT'			
Nome	Tipo di dati	Commento	
Fan_On	Bool	Segnale per l'inserzione del ventilatore	

Come programmare la funzione di temporizzazione in KOP



Selezionare il montante per l'introduzione dell'istruzione KOP.

Nel catalogo degli elementi del programma navigare all'elemento **S_AVERZ** (avvia temporizzatore come ritardo alla disinserzione), ed inserire l'elemento.



Inserire un contatto normalmente aperto prima dell'ingresso **S**.

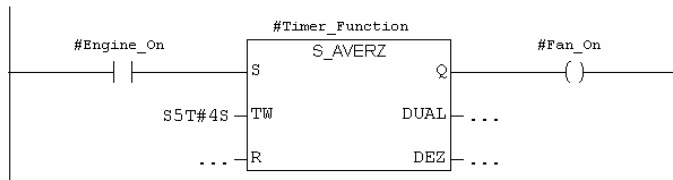
Inserire una bobina dopo l'uscita **Q**.



Selezionare i punti interrogativi, introdurre un "#" e scegliere i nomi corrispondenti.

All'ingresso TW di S_AVERZ impostare la durata del ritardo. S5T#4s significa che una costante con il tipo di dati S5Time#(S5T#) viene definita con una durata di quattro secondi (4s).

Memorizzare infine la funzione e chiudere la finestra.



Con il parametro di ingresso "#Motore_On" viene avviata la "#Funzione di temporizzazione". Ad essa vengono più tardi assegnati da un lato i parametri per il motore a benzina, dall'altro i parametri per il motore diesel (p.es. T1 per "Tempo di inerzia_MB") durante il richiamo nell'OB1. I nomi simbolici di questi parametri vengono più tardi introdotti nella tabella dei simboli.

Come programmare la funzione di temporizzazione in AWL

```

U   #Engine_On
L   S5T#4S
SA  #Timer_Function
U   #Timer_Function
=   #Fan_On
    
```

Nel caso si programmi in AWL, selezionare l'area di introduzione sotto il segmento ed introdurre l'istruzione raffigurata.

Salvare la funzione e chiudere la finestra.

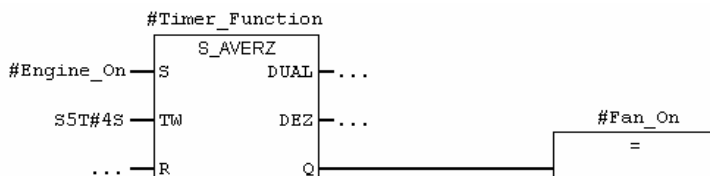




Come programmare la funzione di temporizzazione in FUP

Nel caso si programmi in FUP, selezionare il campo di editazione che si trova sotto il segmento ed introdurre il programma FUP sottostante della funzione di temporizzazione.

Salvare la funzione e chiudere la finestra.



Affinché venga elaborata la funzione di temporizzazione, è necessario un richiamo della funzione in un blocco sovraordinato (nel nostro esempio nell' OB1).

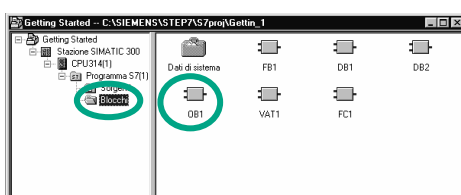
Per maggiori informazioni vedere ? > **Argomenti della Guida** "Richiamo delle Guide di riferimento", "Descrizione del linguaggio KOP/FUP/AWL" e "Operazioni di temporizzazione".

8.3 Richiamo della funzione nell'OB1

Il richiamo della funzione FC1 avviene in modo analogo al richiamo del blocco funzionale nell'OB1. A tutti i parametri della funzione vengono assegnati nell'OB1 gli operandi corrispondenti del motore a benzina o motore diesel.

Poiché questi operandi non sono ancora definiti nella tabella dei simboli, i nomi simbolici degli operandi vengono introdotti in tale tabella.

Un operando è una parte di una istruzione STEP 7 che definisce con quali mezzi deve operare il processore. Esso può essere indirizzato in modo assoluto o simbolico.



Il SIMATIC Manager è aperto con il progetto "Getting Started" o il nuovo progetto creato.

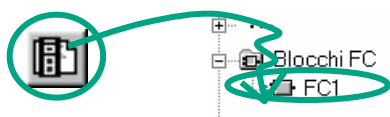
Navigare alla cartella **Blocchi** ed aprire l' **OB1**.

Si apre la finestra del programma KOP/AWL/FUP.

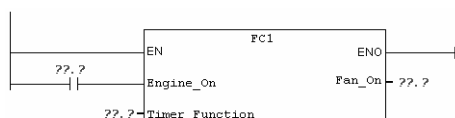
Come programmare il richiamo in KOP



Ci si trova nella visualizzazione **KOP**. Selezionare il segmento 5 e inserire un nuovo segmento (6).



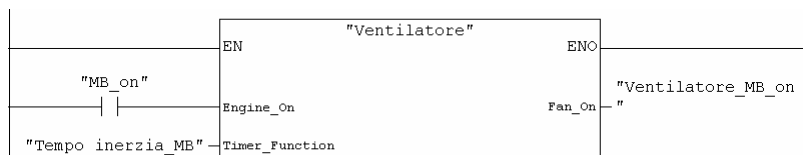
Navigare a FC1 nel catalogo degli elementi del programma e inserire la funzione.



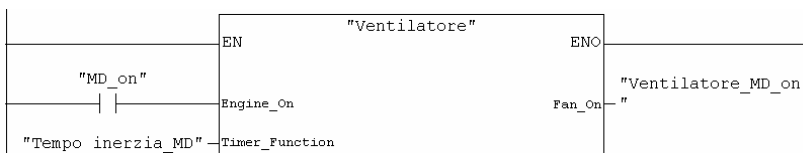
Prima di "Motore_On" inserire un contatto normalmente aperto.

Utilizzando il menu **Visualizza > Finestra con > Rappresentazione simbolica** è possibile commutare tra la rappresentazione simbolica e assoluta.

Fare clic sui punti interrogativi del richiamo FC1 ed introdurre i nomi simbolici.



Programmare nel segmento 7 il richiamo della funzione FC1 con gli operandi del motore diesel. Procedere come per il segmento precedente (gli operandi per il motore diesel sono già stati acquisiti nella tabella dei simboli).



Salvare il blocco e chiudere la finestra.

Attivare **Visualizza > Finestra con > Informazioni sul simbolo**, per ottenere in ogni segmento informazioni dei singoli indirizzi.

Per rappresentare diversi segmenti sullo schermo disattivare **Visualizza > Finestra con > Commento** e eventualmente **Visualizza > Finestra con > Informazioni sul simbolo**.

Con **Visualizza > Fattore di zoom** è possibile regolare la grandezza di rappresentazione dei segmenti.





Come programmare il richiamo in AWL

Segmento 6: Comando ventilatore per motore a benzina

```
CALL "Ventilatore"
Engine_On := "MB_On"
Timer_Function := "Tempo inerzia_MB"
Fan_On := "Ventilatore_MB_on"
```

Nel caso si programmi in AWL, selezionare l'area di introduzione sotto ogni nuovo segmento ed introdurre le istruzioni AWL accanto.

Dopo di che salvare il richiamo e chiudere la finestra.

Segmento 7: Comando ventilatore per motore diesel

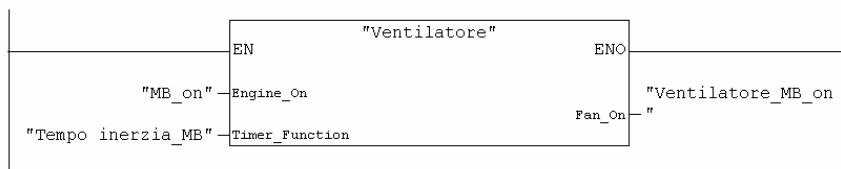
```
CALL "Ventilatore"
Engine_On := "MD_On"
Timer_Function := "Tempo inerzia_MD"
Fan_On := "Ventilatore_MD_on"
```

Come programmare il richiamo in FUP

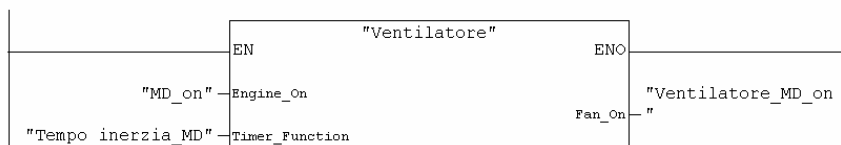
Nel caso programmate in FUP, selezionare l'area di introduzione sotto ogni nuovo segmento ed introdurre le istruzioni FUP sottostanti.

Dopo di che salvare il richiamo e chiudere la finestra.

Segmento 6: Comando ventilatore per motore a benzina



Segmento 7: Comando ventilatore per motore diesel



Il richiamo delle funzioni è stato programmato nel nostro esempio come richiamo assoluto, ovvero la funzione viene sempre elaborata.

A seconda delle esigenze del Vostro compito di automazione è possibile combinare il richiamo di FC o FB anche con certe condizioni: p.es. ad un ingresso o ad un'interconnessione precedente. Per la programmazione delle condizioni sono disponibili l'ingresso EN oppure l'uscita ENO del box.

Per maggiori informazioni ? > **Argomenti della Guida**
 "Richiamo delle Guide di riferimento", "Descrizione del linguaggio KOP/FUP/AWL".

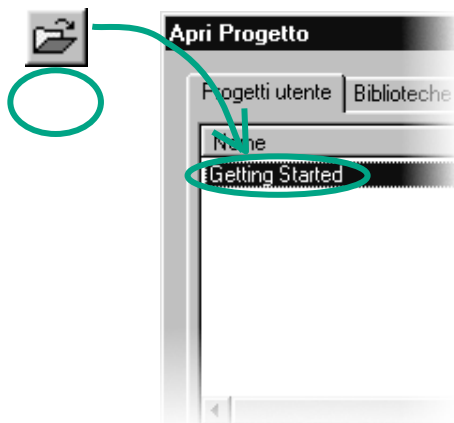
9 Programmazione di un blocco dati globali

9.1 Come creare ed aprire un blocco dati globali

Nel caso il numero di merker interni di una CPU (celle di memoria) non è più sufficiente ad accogliere la quantità di dati, i dati cercati possono essere depositati in un blocco dati globali.

I dati del blocco dati globali sono a disposizione di ogni altro blocco. Un blocco dati di istanza è invece assegnato ad un determinato blocco funzionale, i suoi dati sono disponibili solo localmente in questo blocco funzionale (vedere capitolo 5.5).

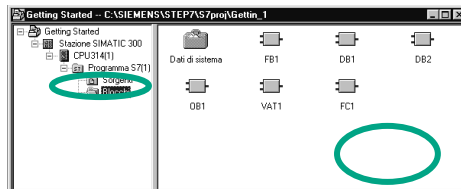
Si dovrebbe già avere dimestichezza con la programmazione in KOP, FUP o AWL (vedere capitoli 4 e 5), nonché con la programmazione con i nomi simbolici (vedere capitolo 3).



Nel caso abbiate eseguito il progetto esemplificativo "Getting Started" capitoli 1–7, apritelo adesso.

Se così non fosse, creare un nuovo progetto nel SIMATIC Manager con **File > Assistente "Nuovo progetto"**. Procedere come descritto al capitolo 2.1 e nominare il progetto "Getting Started DB globali".

Seguiremo lo svolgimento del progetto "Getting Started". E' tuttavia possibile seguire lo svolgimento del programma passo per passo anche sulla base di un nuovo progetto creato.

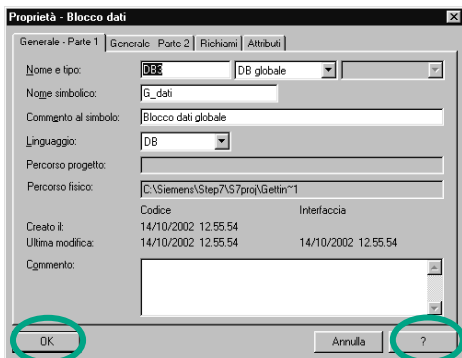


Navigare alla cartella **Blocchi** e aprirla.

Fare clic con il tasto destro del mouse nella parte destra della finestra.



Tramite il menu di scelta rapida inserire un **blocco dati**.



Nella finestra di dialogo "Proprietà – Blocco dati" acquisire tutte le preimpostazioni con **OK**.

Utilizzare la **Guida** per ulteriori informazioni.

Il blocco dati DB3 è stato aggiunto alla cartella **Blocchi**.

Aprire il **DB3** con doppio clic.

Ricordate: Al capitolo 5.5 è stato creato un blocco dati di istanza attivando "Blocco dati associato ad un blocco funzionale". Tramite "Blocco dati" viene invece creato un blocco dati globali.

Come programmare le variabili nel blocco dati

Indirizzo	Nome	Tipo	Valore iniziale	Commento
0.0				
+0.0	INT VAR	INT	0	Variabile jolly provvisoria
-2.0				

Nella colonna Nome introdurre "MB_numero giri_attuale".

Come Tipo selezionare **Dati semplici > INT** con il tasto destro del mouse tramite il menu di scelta rapida.

A titolo di esempio sono definiti tre dati globali nel DB3. Introdurre in modo corrispondente i dati nella tabella di dichiarazione delle variabili.

Indirizzo	Nome	Tipo	Valore iniziale	Commento
0.0		STRUCT		
+0.0	ME_Actual_Speed	INT	0	Numero di giri attuale per il motore a benzina
+2.0	DE_Actual_Speed	INT	0	Numero di giri attuale per il motore diesel
+4.0	Preset_Speed_Reached	BOOL	FALSE	Entrambi i motori hanno raggiunto il numero di giri prefissato
-6.0		END_STRUCT		

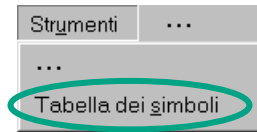
Le variabili per i numeri di giri attuali nel blocco dati "MB_numero giri_attuale" e "MD_numero giri_attuale" vengono trattate come le parole di merker MW2 (MB_numero giri_attuale) e MW4 (MD_numero giri_attuale). Questo viene mostrato al capitolo seguente.



Salvare il blocco dati globali.



Come assegnare i simboli



Anche ad un blocco dati può essere assegnato un nome simbolico.

Aprire la **tabella dei simboli** ed introdurre il simbolo "G_Dati" per il blocco dati DB3.

Nel caso abbiate copiato, al capitolo 4, la tabella dei simboli da un progetto di esempio (ZIt01_05_STEP7__KOP_1-9, ZIt01_01_STEP7__AWL_1-9 o ZIt01_03_STEP7__FUP_1-9) nel progetto "Getting Started", non deve essere introdotto alcun simbolo.

Simbolo	Indirizzo	Tipo di dati	Commento
...
G_dati	DB 3	DB 3	Blocco dati globale ramento di inerzia del ventilatore del motore diesel



Salvare la tabella dei simboli e chiudere la finestra "Editor di simboli".

Chiudere inoltre il blocco dati globale.



DB globali nella tabella di dichiarazione delle variabili:

Con **Visualizza > Dati** è possibile modificare i valori attuali del tipo di dati INT nella tabella del blocco dati globali (vedere capitolo 5.5).

DB globali nella tabella dei simboli:

A differenza del blocco dati di istanza, il tipo di dati per il blocco dati globali è nella tabella dei simboli sempre l'indirizzo assoluto. Nel nostro esempio si tratta del tipo di dati "DB3". Per il blocco dati di istanza viene sempre indicato il relativo FB come tipo di dati.

Per maggiori informazioni vedere ? > **Argomenti della Guida** "Programmazione di blocchi" e "Creazione di blocchi dati".

10 Programmazione di una multiistanza

10.1 Come creare ed aprire un blocco funzionale sovraordinato

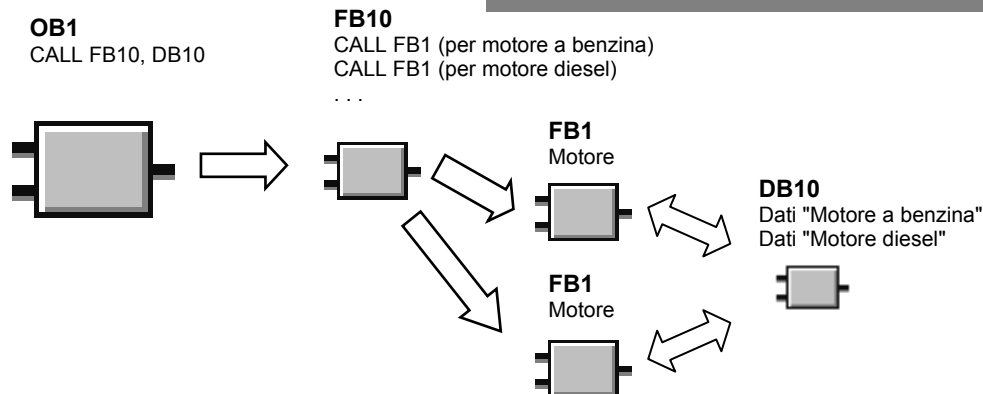
Al capitolo 5 avete programmato un comando del motore con il blocco funzionale "Motore" (FB1). Al richiamo del blocco funzionale FB1 nel blocco organizzativo OB1, l'FB1 ha qui utilizzato i blocchi dati "Benzina" (DB1) e "Diesel" (DB2). Ciascuno dei blocchi dati conteneva i diversi dati (p.es. #Numero giri_prefissato) relativi ai motori.

Immaginate di aver bisogno di altri comandi del motore per i Vostri compiti di automazione, p.es. per il comando di un motore a olio di colza, un motore all'idrogeno, ecc.

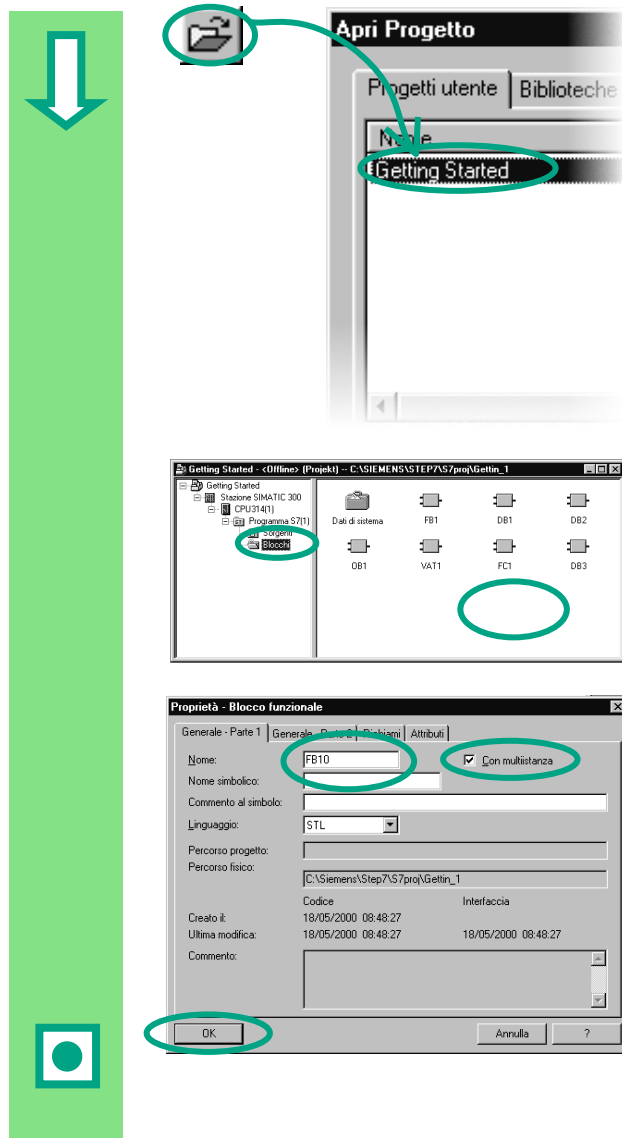
Nel procedimento finora appreso utilizzereste l'FB1 per ogni comando del motore addizionale e ad esso assegnereste un nuovo DB con i relativi nuovi dati del motore, ovvero per il comando del motore a olio di colza l'FB1 con DB3 e per il motore a idrogeno l'FB1 con DB4 ecc. Il numero dei blocchi aumenterebbe notevolmente rispetto ai comandi del motore.

E' possibile ridurre il numero di blocchi lavorando con multiistanze. Creare a tal fine un nuovo FB sovraordinato (nel nostro esempio l'FB10) e richiamare in esso l'FB1 immutato come "istanza locale". Per ogni richiamo l'FB1 subordinato deposita i suoi dati nel blocco DB10 dell' FB10 sovraordinato. All' FB1 non deve così essere assegnato alcun DB. Tutti gli FB fanno riferimento ad un singolo blocco dati (qui il DB10).

Nel DB10 vengono integrati i blocchi dati DB1 e DB2. A tal fine è necessario dichiarare l' FB1 nei dati locali statici dell' FB10.



Si dovrebbe avere già dimestichezza con la programmazione in KOP, FUP o AWL (vedere capitoli 4 e 5) nonché con la programmazione simbolica (vedere capitolo 3).



Nel caso abbiate eseguito l'esempio "Getting Started" capitoli 1–7, aprire il progetto "Getting Started".

Se così non fosse, tramite il SIMATIC Manager aprire il progetto

Zlt01_05_STEP7_KOP_1-9,
Zlt01_01_STEP7_AWL_1-9,
Zlt01_03_STEP7_FUP_1-9.

Navigare alla cartella **Blocchi**, e aprirla.

Fare clic con il tasto destro del mouse nella parte destra della finestra, ed introdurre un blocco funzionale tramite il menu di scelta rapida.

Modificare il nome del blocco in "FB10", e selezionare il linguaggio di programmazione desiderato.

Se necessario, attivare la **proprietà di multiistanza**, e acquisire le preimpostazioni rimanenti con **OK**.

L'FB10 è stato aggiunto alla cartella Blocchi. Aprire l'FB10 con doppio clic.

Potete creare multiistanze per blocchi funzionali a piacere, p.es. anche per comandi di valvole. Se desiderate lavorare con multiistanze tenere presente che sia i blocchi funzionali richiamanti, sia i blocchi funzionali richiamati hanno proprietà di multiistanza.

Per maggiori informazioni vedere ? > **Argomenti della Guida** "Programmazione di blocchi" e "Creazione di blocchi e biblioteche"

10.2 Come programmare l'FB10

Per richiamare l' FB1 come "Istanza locale" dell' FB10, nella visualizzazione analitica delle variabili viene dichiarata per ogni richiamo programmato dell'FB1 una variabile statica e viene predefinito un altro nome. Il tipo di dati è qui l'FB1 ("Motore").



Dichiarazione / definizione delle variabili

L'FB 10 è aperto nella finestra del programma KOP/AWL/FUP. Riportare nella visualizzazione analitica delle variabili le dichiarazioni illustrate nel seguito. Selezionare in sequenza i tipi di dichiarazione "OUT", "STAT" e "TEMP" nella visualizzazione sintetica delle variabili e completare la visualizzazione analitica. Per il tipo di dichiarazione "STAT" scegliere nella casella di riepilogo il tipo di dati "FB <n.>" e sostituire "<n.>" mediante il numero "1".

Contenuto di: 'Ambiente\Interfaccia\OUT'					
Nome	Tipo di dati	Indirizzo	Valore iniziale	Commento	
Preset_Speed_Reached	Bool	0.0	FALSE	Entrambi i motori hanno raggiunto il numero di giri prefissat	

Contenuto di: 'Ambiente\Interfaccia\STAT'					
Nome	Tipo di dati	Indirizzo	Valore iniziale	Commento	
Petrol_Engine	Motore	2.0		Prima istanza locale dell' FB 1 "Motore"	
Diesel_engine	Motore	2.0		Seconda istanza locale dell' FB 1 "Motore"	

Contenuto di: 'Ambiente\Interfaccia\TEMP'					
Nome	Tipo di dati	Indirizzo	Valore iniziale	Commento	
PE_Preset_Speed_Reached	Bool	0.0		Numero di giri prefissato raggiunto (motore a benzina)	
DE_Preset_Speed_Reached	Bool	0.1		Numero di giri prefissato raggiunto (motore diesel)	

Le istanze locali dichiarate compaiono infine nel catalogo degli elementi del programma sotto "Multiistanze".



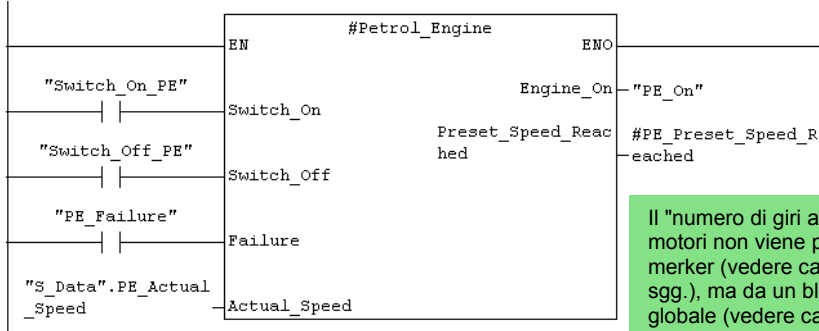


Come programmare FB10 in KOP



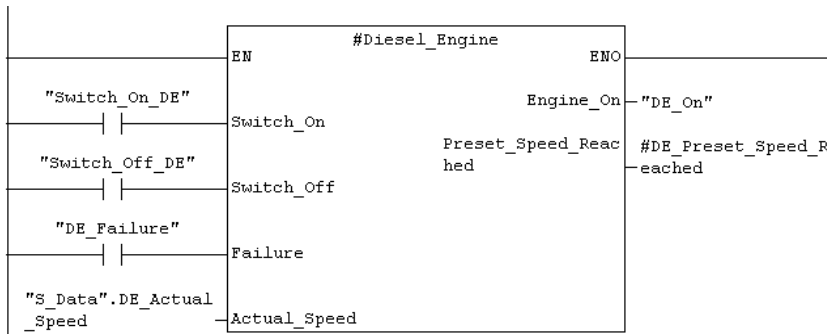
Inserire il richiamo "Motore a benzina" come blocco di multiistanza "Motore a benzina" nel segmento 1.

Infine inserire i necessari contatti normalmente aperti e completare il richiamo con i nomi simbolici.



Il "numero di giri attuale" per i motori non viene prelevato da un merker (vedere capitolo 5.6 sgg.), ma da un blocco dati globale (vedere capitolo 9.1). L'indirizzamento generale si chiama "Blocco dati".operando, p.es. "G_Dati".MB_numero_giri_attuale.

Inserire un nuovo segmento e programmare il richiamo del motore diesel. Procedere come per il segmento 1.



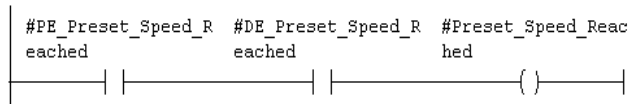


Inserire un nuovo segmento e programmare un circuito in serie con l'indirizzamento corrispondente. Salvare poi il programma e chiudere il blocco.



Utilizzare le variabili temporanee, contrassegnate nella casella di riepilogo dal simbolo illustrato a sinistra.

Salvare infine il programma e chiudere il blocco.



Le variabili temporanee ("MB_prefissato_raggiunto" e "MD_prefissato_raggiunto") vengono trasferite al parametro di uscita "Prefissato_raggiunto" che viene poi elaborato ulteriormente nell'OB1.

Come programmare FB10 con AWL

```
CALL #Petrol_Engine
  Switch_On      := "Switch On PE"
  Switch_Off     := "Switch Off PE"
  Failure        := "PE Failure"
  Actual_Speed   := "S_Data".PE_Actual_Speed
  Engine_On      := "PE On"
  Preset_Speed_Reached := #PE_Preset_Speed_Reached
```

```
CALL #Diesel_Engine
  Switch_On      := "Switch On PE"
  Switch_Off     := "Switch Off PE"
  Failure        := "PE Failure"
  Actual_Speed   := "S_Data".PE_Actual_Speed
  Engine_On      := "PE On"
  Preset_Speed_Reached := #PE_Preset_Speed_Reached
```

```
U #PE_Preset_Speed_Reached
U #DE_Preset_Speed_Reached
= #Preset_Speed_Reached
```

Nel caso si programma in AWL, selezionare l'area di introduzione sotto il nuovo segmento ed introdurre le istruzioni AWL vicine.

Salvare infine il programma e chiudere il blocco.

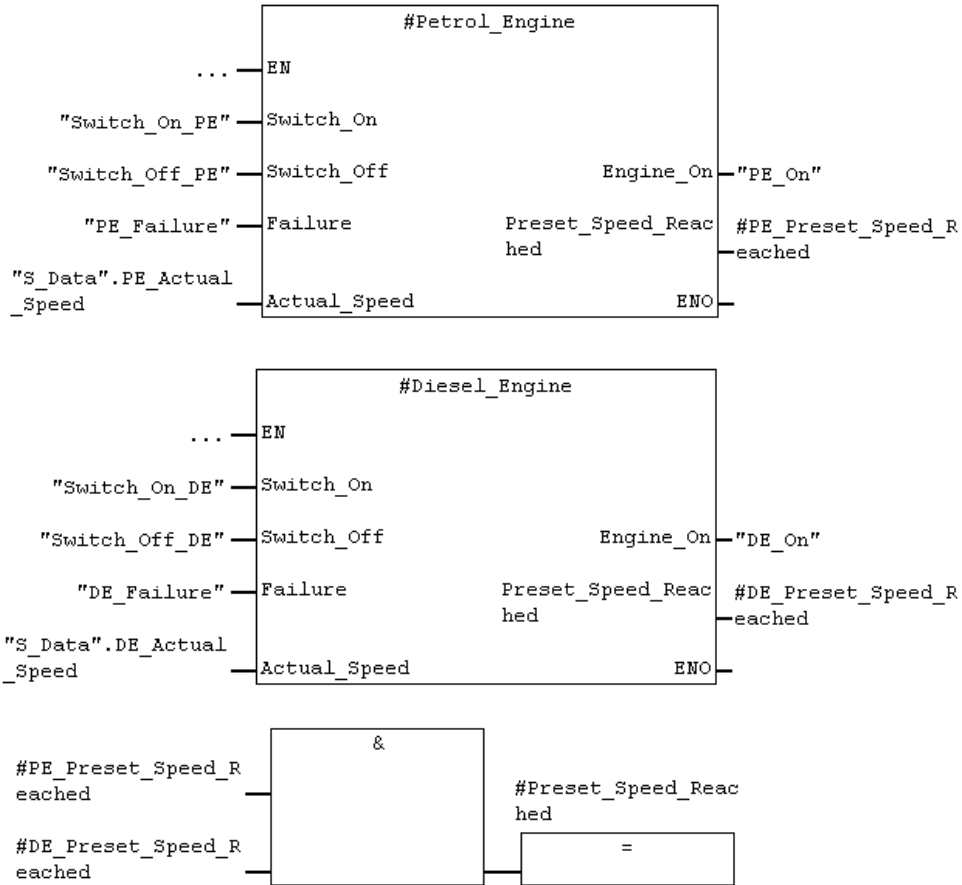




Come programmare FB10 in FUP

Nel caso programmate in FUP, selezionare l'area di introduzione sotto ogni nuovo segmento ed introdurre le istruzioni FUP sottostanti.

Salvare infine il programma e chiudere il blocco.

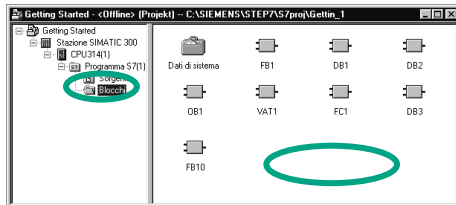


Per elaborare i due richiami dell'FB10 nell' FB10, deve essere richiamato l'FB10 stesso. Le multiistanze possono essere programmate esclusivamente per i blocchi funzionali. La creazione di multiistanze per le funzioni (FC) non è possibile.

Per maggiori informazioni vedere ? > **Argomenti della Guida** "Programmazione di blocchi", "Creazione di blocchi di codice" e "Multiistanze nella dichiarazione delle variabili".

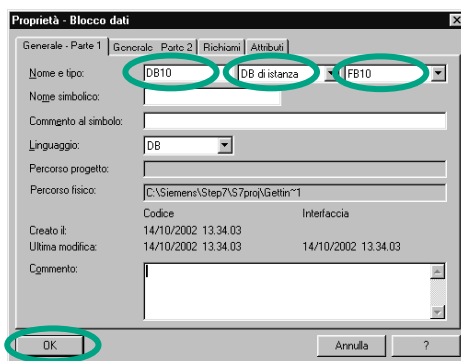
10.3 Come creare il DB10 ed adattare il valore attuale

Il nuovo blocco dati DB10 sostituirà i blocchi dati DB1 e DB2. Nel DB10 vengono depositati i dati del motore a benzina o diesel che saranno necessari più tardi durante il richiamo dell'FB10 nell'OB1 (vedere il richiamo dell'FB1 nell'OB1 al capitolo 5.6 e sgg.).



Nel SIMATIC Manager, progetto "Getting Started", cartella **Blocchi** generare il blocco dati DB10 con il menu di scelta rapida del tasto destro del mouse.

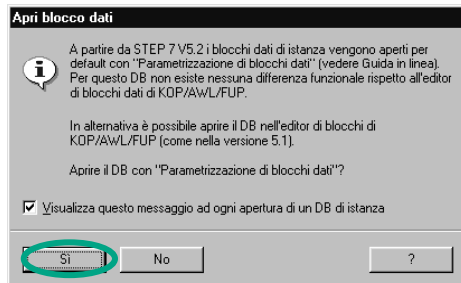
A tal fine modificare nella finestra di dialogo il nome del blocco dati in DB10, e confermare le rimanenti impostazioni con **OK**.



Nella finestra di dialogo "Proprietà – Blocco dati" modificare in DB10 il nome del blocco dati, scegliere nella casella di riepilogo la voce "DB di istanza". Scegliere il nome del blocco funzionale "FB10" assegnato e confermare tutte le impostazioni con **OK**.

Il blocco dati DB10 viene aggiunto al progetto "Getting Started".

Aprire il **DB10** con un doppio clic.



Per parametrizzare il blocco dati di istanza, confermare con **Si** la finestra di dialogo visualizzata.

Attivare la vista di **Dati**.

Dati (vista di-) mostra ora ogni singola variabile nel DB10, anche le variabili "interne" dei due richiami dell'FB1 ("istanze locali").

Dichiarazione mostra le variabili come sono dichiarate nell'FB10.



Modificare il valore attuale del motore diesel in "1300", salvare il blocco dati e infine chiuderlo.

	Indirizzo	Dichiarazion	Nome	Tipo	Valore iniziale	Valore attuale	Commento
1	0.0	out	Preset_Speed_Reached	BOOL	FALSE	FALSE	Entrambi i motori hanno raggiunto il numero di giri prefissato
2	2.0	stat:in	Petrol_Engine.Switch_On	BOOL	FALSE	FALSE	Inserisci motore
3	2.1	stat:in	Petrol_Engine.Switch_Off	BOOL	FALSE	FALSE	Disinserisci motore
4	2.2	stat:in	Petrol_Engine.Failure	BOOL	FALSE	FALSE	Disturbo del motore, causa la disinserzione
5	4.0	stat:in	Petrol_Engine.Actual_Speed	INT	0	0	Numero di giri effettivo del motore
6	6.0	stat:out	Petrol_Engine.Engine_On	BOOL	FALSE	FALSE	Il motore viene inserito
7	6.1	stat:out	Petrol_Engine.Preset_Speed_Reached	BOOL	FALSE	FALSE	Numero di giri prefissato raggiunto
8	8.0	stat	Petrol_Engine.Preset_Speed	INT	1500	1500	Numero di giri del motore richiesto
9	10.0	stat:in	Diesel_engine.Switch_On	BOOL	FALSE	FALSE	Inserisci motore
10	10.1	stat:in	Diesel_engine.Switch_Off	BOOL	FALSE	FALSE	Disinserisci motore
11	10.2	stat:in	Diesel_engine.Failure	BOOL	FALSE	FALSE	Disturbo del motore, causa la disinserzione
12	12.0	stat:in	Diesel_engine.Actual_Speed	INT	0	0	Numero di giri effettivo del motore
13	14.0	stat:out	Diesel_engine.Engine_On	BOOL	FALSE	FALSE	Il motore viene inserito
14	14.1	stat:out	Diesel_engine.Preset_Speed_Reached	BOOL	FALSE	FALSE	Numero di giri prefissato raggiunto
15	16.0	stat	Diesel_engine.Preset_Speed	INT	1500	1300	Numero di giri del motore richiesto

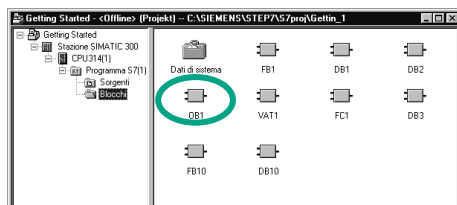
Nella tabella di dichiarazione delle variabili del DB10 sono ora contenute tutte le variabili. Nella prima parte della tabella si vedono le variabili per il richiamo del blocco funzionale "Motore a benzina" e nella parte inferiore il richiamo per il blocco funzionale "Motore diesel" (vedere capitolo 5.5).

Le variabili "interne" dell'FB1 contengono i nomi simbolici, p.es. "Inserisci". Questi vengono ora preceduti dal nome dell'istanza locale, p.es "Motorebenzina.Inserisci".

Per maggiori informazioni vedere ? > **Argomenti della Guida**
 "Programmazione di blocchi", "Creazione di blocchi dati".

10.4 Richiamo dell'FB10 nell'OB1

Il richiamo dell'FB10 avviene nel nostro esempio nell'OB1. Questo richiamo ha la stessa funzionalità di come abbiamo imparato durante la programmazione ed il richiamo dell'FB1 nell'OB1 (vedere capitolo 5.6 sgg.). Utilizzando la multiistanza i segmenti 4 e 5 programmati al capitolo 5.6 e sgg. vengono sostituiti.



Aprire l'OB1 nel progetto in cui avete appena programmato l'FB10.

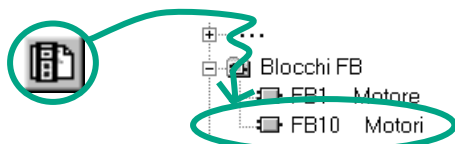
Come definire i nomi simbolici

La finestra del programma KOP/AWL/FUP è aperta. Aprire la tabella dei simboli con **Strumenti > Tabella dei simboli**, ed introdurre nella tabella dei simboli i nomi simbolici per il blocco funzionale FB10 ed il blocco dati DB10.

Salvare infine la tabella dei simboli e chiudere la finestra.

Simbolo	Indirizzo	Tipo di dati	Commento
...
Motori	FB 10	FB 10	Esempio per le multi-istanze
Dati_motore	DB 10	FB 10	Blocco dati di istanza per FB 10
...

Come programmare il richiamo in KOP



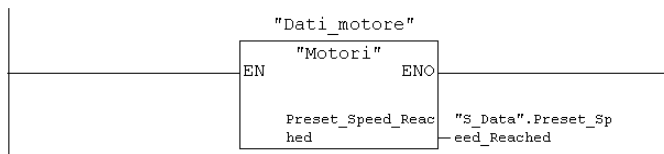
Alla fine dell'OB1 inserire un nuovo segmento e completarlo con il richiamo dell'FB10 ("Motori").



Completare il richiamo sottostante con i nomi simbolici corrispondenti.

Cancellare il richiamo dell' FB1 nell'OB1 (segmenti 4 e 5 del capitolo 5.6 e sgg), poiché ora l'FB1 viene richiamato centralmente tramite l'FB10.

Salvare infine il programma e chiudere il blocco.



Il segnale di uscita "Prefissato_raggiunto" dell' FB10 ("Motori") viene trasferito alla variabile nel blocco dati globale.

Richiamo in AWL

Nel caso si programmi in AWL, selezionare l'area di introduzione sotto il nuovo segmento ed introdurre le istruzioni AWL sottostanti. Utilizzare a tal fine **Blocchi FB > FB10 Motori** nel catalogo degli elementi del programma.

Cancellare il richiamo dell'FB1 nell'OB1 (segmenti 4 e 5 del capitolo 5.6 e sgg), poiché ora l'FB1 viene richiamato centralmente tramite l'FB10.

Salvare infine il programma e chiudere il blocco.

```

CALL "Motori" , "Dati_motore"
Preset_Speed_Reached:="S_Data".Preset_Speed_Reached
  
```



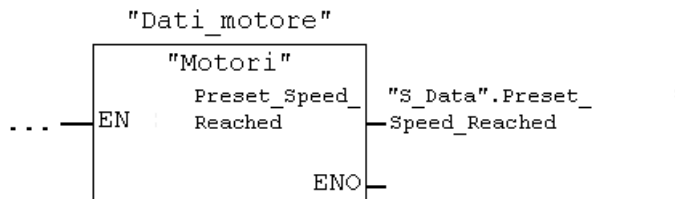


Richiamo in FUP

Nel caso si programmi in FUP, selezionare l'area di introduzione del nuovo segmento ed introdurre le istruzioni FUP sottostanti. Utilizzare a tal fine **blocchi FB > FB10 Motori** nel catalogo degli elementi del programma.

Cancellare il richiamo dell'FB1 nell' OB1 (segmenti 4 e 5 al capitolo 5.6 sgg.), poiché ora l'FB1 viene richiamato centralmente tramite l'FB10.

Salvare infine il programma e chiudere il blocco.



Se per la soluzione del Vostro compito di automazione sono necessari altri comandi di motore, quali motori a metano, motori a biogas ecc., anche questi devono essere programmati come multiistanza e devono essere richiamati tramite l'FB10.

A tal fine dichiarare, come illustrato, gli altri motori nella tabella di dichiarazione delle variabili dell'FB10 ("Motori") e programmare nell'FB10 il richiamo dell'FB1 (Multiistanza nel catalogo degli elementi del programma). Per la programmazione con nomi simbolici devono infine essere definiti nella tabella dei simboli i nuovi nomi simbolici p.es. per le operazioni di inserzione e disinserzione.

Per maggiori informazioni vedere ? > **Argomenti della Guida** "Richiamo delle Guide di riferimento", "Rimandi alla descrizione dei linguaggi e Guida a blocchi e attributi di sistema".

11 Configurazione della periferia decentrata

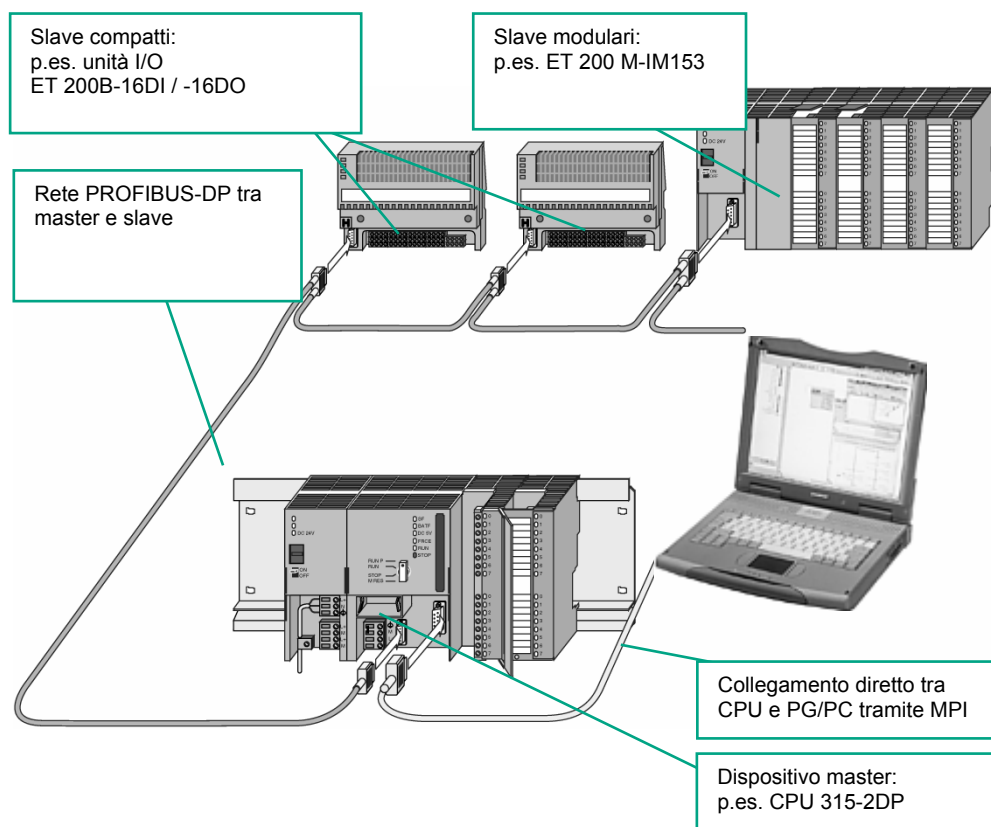
11.1 Come configurare la periferia decentrata con PROFIBUS-DP

Nella configurazione tradizionale di impianti di automazione i cavi di collegamento dei sensori e degli attuatori vengono inseriti direttamente nelle unità di ingresso/uscita del sistema di automazione della configurazione centrale. Ciò dà spesso luogo a lunghi tempi di cablaggio.

Una configurazione decentrata consente di ridurre notevolmente i tempi di cablaggio posizionando le unità di ingresso/uscita in prossimità dei sensori e degli attuatori. Il collegamento tra il sistema di automazione, le unità della periferia e gli apparecchi di campo viene stabilito mediante il bus di campo PROFIBUS-DP.

Abbiamo appreso già nel capitolo 6 la programmazione della configurazione tradizionale. Nella configurazione decentrata non vi sono differenze rispetto alla configurazione centrale. Selezionare dal catalogo hardware le unità, ordinarle e adattare le loro proprietà alle esigenze specifiche.

Sarebbe utile sapere già come creare un progetto e come eseguire una configurazione centrale (vedere capitoli 2.1 e 6).

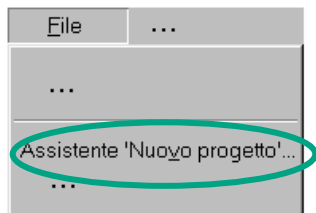




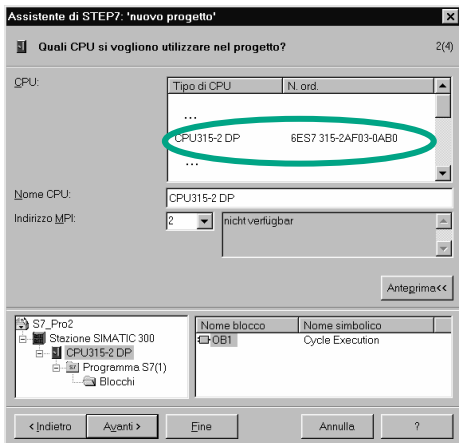
Come creare un nuovo progetto



Punto di partenza è il SIMATIC Manager. Per una maggiore chiarezza chiudere eventuali progetti ancora aperti.



Creare un **nuovo progetto**.

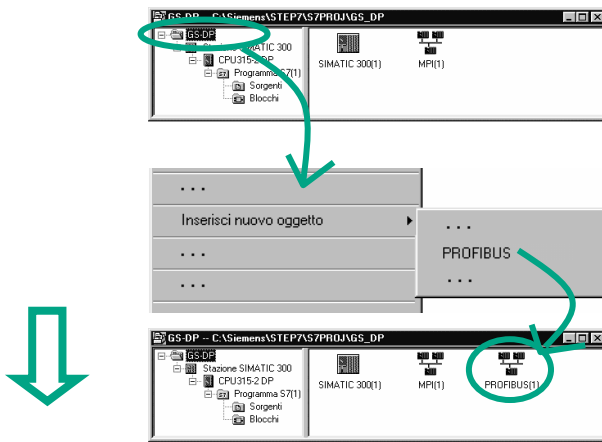


Nella finestra di dialogo corrispondente selezionare la **CPU 315-2DP** (CPU con la rete PROFIBUS-DP).

Altrimenti procedere come descritto al capitolo 2.1 ed assegnare al progetto il nome "GS-DP" (Getting Started – Periferia decentrata).

Nel caso vogliate creare subito la configurazione, indicare ora la CPU impiegata. Fare attenzione che questa sia dotata di proprietà DP.

Come inserire la rete PROFIBUS

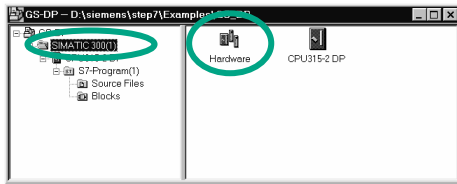


Selezionare la cartella **GS-DP**.

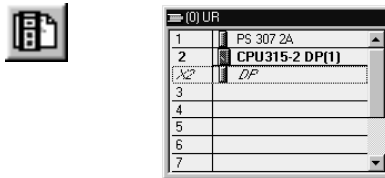
Inserire la **Rete PROFIBUS** tramite il menu di scelta rapida del tasto destro del mouse.



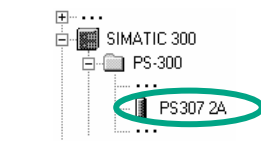
Come configurare la stazione



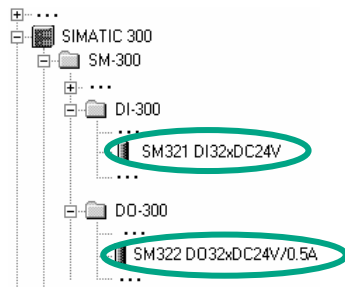
Selezionare la cartella **Stazione SIMATIC300**, e fare doppio clic su **Hardware**. Viene aperta la finestra "Configurazione HW"- (vedere il capitolo 6.1).



La CPU 315-2DP è già presente nel telaio di montaggio. Se necessario, attivare il catalogo hardware tramite **Visualizza > Catalogo hardware** o il pulsante.



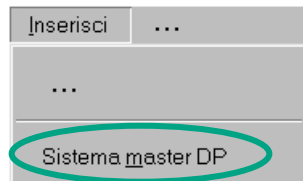
Inserire l'alimentatore **PS307 2A** sul posto connettore 1 mediante Drag & Drop.



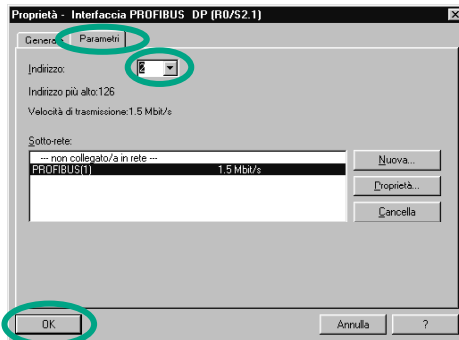
Inserire analogamente le unità di ingresso/uscita **DI32xDC24V** e **DO32xDC24V/0.5A** sui posti connettore 4 e 5.

Oltre alla CPU con proprietà DP possono essere collocate sullo stesso telaio di montaggio anche unità centrali (questa operazione non viene qui effettuata).

Come configurare il sistema master DP

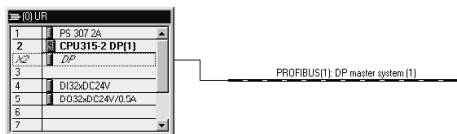


Selezionare un **sistema master DP** ed inserire il master DP sul posto connettore 2.1.

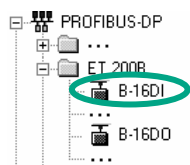


Nella finestra di dialogo visualizzata acquisire l'indirizzo proposto.

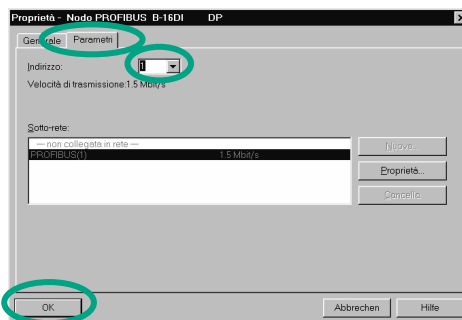
Nel campo "Sottorete" selezionare "PROFIBUS(1)" e confermare le impostazioni con OK.



Tutti gli oggetti nel sistema master possono essere spostati selezionandoli e tenendo premuto il tasto del mouse.



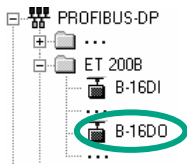
Navigare all'unità **B-16DI** nel catalogo hardware ed inserirla nel sistema master (con Drag & Drop trascinare direttamente sul sistema master, il puntatore del mouse modifica il proprio aspetto, rilasciarlo).



Nella finestra di dialogo "Proprietà" è possibile ora modificare sotto **Parametri** l'indirizzo di nodo dell'unità inserita.

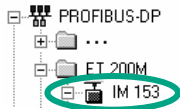
Confermare con **OK** l'indirizzo proposto.





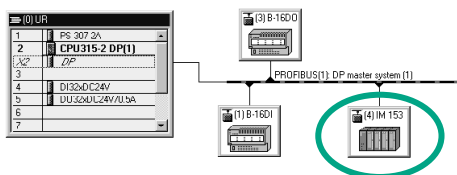
Trascinare analogamente l'unità **B-16DO** sul sistema master.

Nella finestra di dialogo l'indirizzo di nodo viene adattato automaticamente. Confermarlo con **OK**.



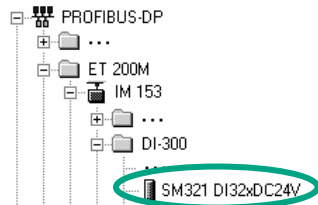
Trascinare l'unità di interfaccia **IM 153** sul sistema master, e confermare di nuovo l'indirizzo del nodo con **OK**.

Nel nostro esempio acquisiamo gli indirizzi di nodi preimpostati. Questi indirizzi possono tuttavia essere modificati in qualsiasi momento e adattati alle esigenze specifiche.



Selezionare l'**ET 200M** nella rete. Nella tabella di configurazione vengono visualizzati ora i posti connettore liberi dell'ET 200M. Selezionare lì il posto connettore **4**.

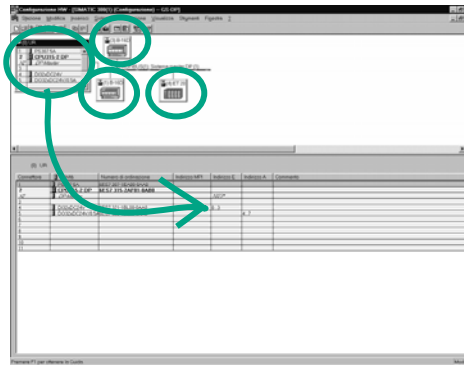
Posto connettore	Unità	Numero di ordinazione
5		
6		
7		
8		
9		
10		
11		



L'ET 200M può accogliere altre unità di ingresso/uscita. Selezionare per esempio l'unità **DI32xDC24V** per il posto connettore 4 ed inserire l'unità con doppio clic.

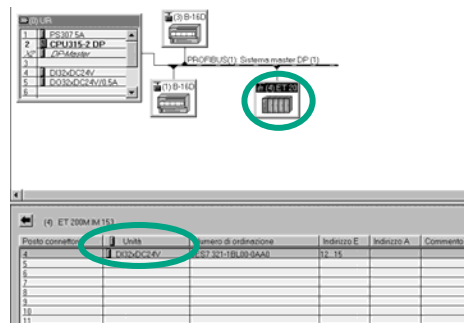
Durante la selezione delle unità, verificare che si stia lavorando nella cartella corretta del catalogo hardware (p.es. per la selezione delle unità per l'ET 200M nella cartella ET 200M).

Come modificare l'indirizzo del nodo



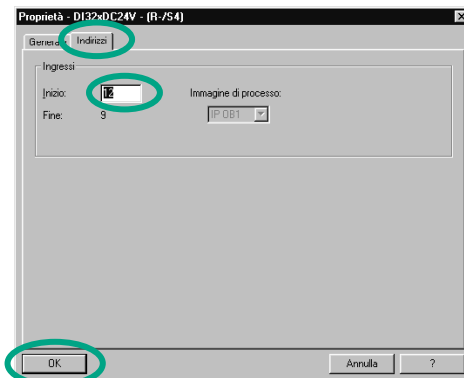
Nel nostro esempio non è richiesta la modifica dell'indirizzo del nodo. Nella pratica, tuttavia, questa è spesso necessaria.

Selezionare gli altri nodi in successione e controllare gli indirizzi di ingresso e di uscita. La configurazione hardware ha adattato tutti gli indirizzi, non vi sono assegnazioni doppie.

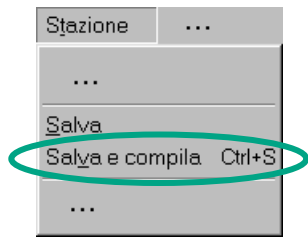


Supponiamo che vogliate modificare l'indirizzo dell'ET 200M:

Selezionare l'**ET 200M** e fare doppio clic su **DI32xDC24V** (posto connettore 4).



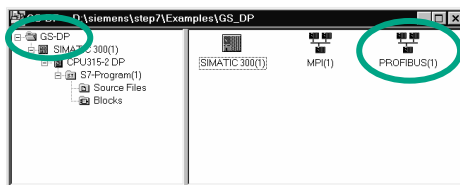
Nella finestra di dialogo "Proprietà" modificare sotto **Indirizzi** ora gli indirizzi di ingresso da 6 a 12. Chiudere la finestra di dialogo con **OK**.



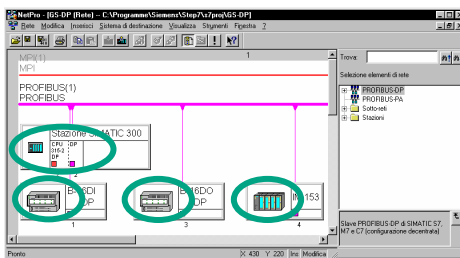
Infine **salvare e compilare** la configurazione della periferia decentrata.
Chiudere la finestra.

Con Salva e compila la configurazione viene automaticamente sottoposta ad una verifica della coerenza. Se la configurazione è priva di errori vengono allora creati i dati di sistema che possono essere caricati nel sistema di destinazione.
Con Salva la configurazione può essere salvata anche se è errata. In questo caso non è possibile caricare la configurazione nel sistema di destinazione.

Alternativa: Progettazione della rete



La configurazione della periferia decentrata può essere eseguita anche con la progettazione della rete.
Fare doppio clic nel SIMATIC Manager sulla rete **PROFIBUS (1)**.



Viene aperta la finestra "NetPro".
Dal catalogo degli oggetti della rete è possibile trascinare altri slave DP sul PROFIBUS-DP.
Fare doppio clic su un elemento a piacere per configurarlo. Viene aperta la finestra "Configurazione hardware".

Con **Stazione > Verifica coerenza** (finestra Configurazione hardware) e **Rete > Verifica coerenza** (finestra Progettazione della rete) è possibile verificare se vi sono errori nella configurazione. In caso di eventuali errori, questi vengono visualizzati da STEP 7 e vengono offerte diverse soluzioni.

Per maggiori informazioni vedere ? > **Argomenti della Guida** "Configurazione dell'hardware" e "Configurazione della periferia decentrata".

Congratulazioni! Avete eseguito "Getting Started" ed avete appreso i concetti importanti, i procedimenti e le funzionalità di STEP 7. In tal modo potete già "lanciarVi" con il Vostro primo progetto.

Nel caso in cui in progetti futuri cerchiate determinate funzioni o abbiate dimenticato alcune sequenze di comando di STEP 7, potete utilizzare la nostra ricca Guida a STEP 7.

Seminari specifici vengono inoltre offerti se volete approfondire le Vostre conoscenze di STEP 7: i consulenti di vendita Siemens saranno lieti di darVi tutti i consigli di cui avete bisogno.

Buon lavoro

Siemens AG

A. Appendice A

A.1 Panoramica dei progetti di esempio per Getting Started

- **ZIt01_01_STEP7__AWL_1-9:**
I capitoli di programmazione 1 - 9 inclusa la tabella dei simboli nel linguaggio di programmazione AWL.
- **ZIt01_02_STEP7__AWL_1-10:**
I capitoli di programmazione 1 - 10 inclusa la tabella dei simboli nel linguaggio di programmazione AWL.
- **ZIt01_03_STEP7__FUP_1-9:**
I capitoli di programmazione 1 - 9 inclusa la tabella dei simboli nel linguaggio di programmazione FUP.
- **ZIt01_04_STEP7__FUP_1-10:**
I capitoli di programmazione 1 - 10 inclusa la tabella dei simboli nel linguaggio di programmazione FUP.
- **ZIt01_05_STEP7__KOP_1-9:**
I capitoli di programmazione 1 - 9 inclusa la tabella dei simboli nel linguaggio di programmazione KOP.
- **ZIt01_06_STEP7__KOP_1-10:**
I capitoli di programmazione 1 - 10 inclusa la tabella dei simboli nel linguaggio di programmazione KOP.
- **ZIt01_07_STEP7__DezP_11:**
Il capitolo di programmazione 11 con la periferia decentrata.

Indice analitico

A

- Accendere la CPU 67
- AND istruzione in AWL 30
- Applicare la tensione 65
- Aprire OB1 24
- Aprire un blocco dati globali 85
- Aprire una funzione 77
- Avviare il SIMATIC Manager 13
- AWL
 - AND istruzione 30
 - istruzione di memoria 32
 - istruzione OR 31
 - programmare la funzione di temporizzazione 80
 - richiamo di blocco 55
 - Testare 68

C

- Cancellazione totale della CPU e commutazione in RUN 65
- Caricamento del programma nel sistema di destinazione 65
- Circuito in parallelo in KOP 28
- Circuito in serie in KOP 26
- Collegare online la tabella delle variabili 71
- Comandare le variabili 72
- Come procedere con STEP 7 10
- Configurare il sistema master DP 105
- Configurare l'hardware 61
- Configurare la periferia decentrata PROFIBUS-DP 101
- Configurazione della periferia decentrata 101
- Configurazione delle unità centrali 61
- Configurazione dell'hardware 61
- Controllare le variabili 72
- Controllare lo stato di funzionamento 67
- Copiare la tabella dei simboli 24
- Creare i blocchi dati di istanza 50
- Creare il progetto 13
- Creare la tabella delle variabili 70
- Creare un blocco dati globali 85
- Creare un blocco funzionale 37
- Creare una funzione 77
- Creazione di un programma nell'OB1 23

D

- DB globali nella tabella dei simboli 87
- DB globali nella tabella di dichiarazione delle variabili 87
- Definire l'hardware 63
- Dichiarazione delle variabili
 - AWL 43
 - FUP 46
 - KOP 39

F

- Finestra del programma KOP/AWL/FUP 25
- Funzione AND in FUP 33
- Funzione di memoria in KOP 29
- Funzione OR in FUP 35
- FUP
 - funzione AND 33
 - funzione OR 35
 - programmare la funzione di temporizzazione 81
 - richiamo di blocco 57
 - Testare 68

I

- in AWL 31
- Indirizzo assoluto 19
- Inserisci > Simbolo 27, 31, 34
- Inserisci simbolo
 - AWL 31
 - FUP 34
- Inserisci Simbolo
 - KOP 27
- Installazione 11
- Introduzione a STEP 7 7
- Istruzione di memoria in AWL 32

K

- KOP
 - circuito in parallelo 28
 - circuito in serie 26
 - funzione di memoria 29
 - programmare la funzione di temporizzazione 79
 - richiamo di blocco 52
 - Testare 68

M

- Materia di apprendimento 7
- Modificare i valori attuali 50
- Modificare l'indirizzo del nodo 107

N

- Navigare all'interno della struttura del progetto 18

P

- Progettazione della rete 108
- Programma con FB e DB 37
- Programmazione con nomi simbolici 20
- Programmazione la funzione di temporizzazione in AWL 80
- Programmazione la funzione di temporizzazione in FUP 81

Programmare la funzione di temporizzazione in KOP 79
Programmazione dell'FB in AWL 43
Programmazione dell'FB in FUP 46
Programmazione dell'FB in KOP 39
Programmazione dell'FB1 in AWL 43
Programmazione dell'FB1 in FUP 46
Programmazione dell'FB1 in KOP 39
Programmazione di un blocco dati globali 85
Programmazione di una funzione (FC) 77
Programmazione di una multiistanza 89

R

Rappresentazione simbolica
 AWL 32
 FUP 36
Rete > Verifica coerenza 108
Richiamo della funzione 82
Richiamo della Guida 17
Richiamo di blocco in AWL 55
Richiamo di blocco in FUP 57
Richiamo di blocco in KOP 52

S

Scegliere tra KOP, AWL o FUP 23
Stabilire il collegamento online 63
Stazione > Verifica coerenza 108
STEP7
 Assistent
 Neues Projekt 13
Struttura del progetto nel SIMATIC Manager 16

T

Testare con AWL 68
Testare con FUP 68
Testare con KOP 68
Tipo di dati 21

V

Valutare il buffer di diagnostica 74
Verifica coerenza
 Rete 108
 Stazione 108
Vista di Dati 95
vista di dichiarazione 95