

PLC e Structured Text parte 1

Convenzioni usate negli esempi:

␣ per indicare uno spazio; se ␣ non compare, non va inserito uno spazio

↵ per indicare che bisogna andare a capo e continuare sulla riga successiva.

Una nota sul simbolo ␣

Il simbolo ␣ si chiama blank symbol.

È un simbolo storico dell'informatica usato per rappresentare graficamente lo spazio, che normalmente non è visibile.

Il termine SPACE per indicare il carattere spazio era già utilizzato nei primi standard informatici degli anni Sessanta. Il simbolo ␣ venne usato nella documentazione tecnica proprio per rendere visibile quello spazio. Oggi è meno comune, ma indica ancora la stessa cosa.

I nomi delle variabili

Il nome di una variabile può essere scelto liberamente, rispettando le regole del linguaggio Structured Text e del software PLC utilizzato.

Per esempio:

variabile1

temperatura

pompa1

statoAllarme

Non è obbligatorio iniziare con una lettera maiuscola né utilizzare un nome particolare. È comunque consigliabile scegliere nomi chiari e descrittivi, così da rendere il programma più facile da leggere e mantenere.

Assegnare un valore

Scriviamo:

variabile1␣:=␣TRUE;

Significa assegnare alla variabile di nome **variabile1** il valore **TRUE**, cioè VERO.

variabile1 nome della variabile

:= operatore di assegnazione

TRUE valore vero

; fine dell'istruzione

In un programma PLC, l'assegnazione viene eseguita durante l'elaborazione del codice Structured Text.

Assegnare il valore falso

Scriviamo:

```
variabile1 := FALSE;
```

Significa assegnare alla variabile **variabile1** il valore **FALSE**.

variabile1 nome della variabile

:= operatore di assegnazione

FALSE valore falso

; fine dell'istruzione

Usare una condizione

Scriviamo:

```
IF variabile1 = TRUE THEN ↵
```

```
variabile2 := TRUE; ↵
```

```
END_IF;
```

Significa: **SE variabile1** è uguale a **TRUE**, allora assegna a **variabile2** il valore **TRUE**.

IF introduce la condizione.

THEN indica l'azione da eseguire quando la condizione è vera.

END_IF; chiude la struttura condizionale.

In Structured Text, il confronto viene eseguito con l'operatore **=**, mentre l'assegnazione viene eseguita con l'operatore **:=**

Aggiungere un'alternativa

Scriviamo:

```
IF variabile1 = TRUE THEN ↵
```

```
variabile2 := TRUE; ↵
```

```
ELSE ↵
```

```
variabile2 := FALSE; ↵
```

```
END_IF;
```

Significa: **SE** *variabile1* è **uguale** a **TRUE**, **ALLORA** assegna a *variabile2* il valore **TRUE**; **ELSE** (ALTRIMENTI), assegna a *variabile2* il valore **FALSE**.

END_IF; chiude la condizione.

La parte introdotta da **ELSE** viene eseguita quando la condizione dell'**IF** è falsa.

Controllare due condizioni insieme

Scriviamo:

```
IF variabile1 = TRUE AND variabile2 = TRUE THEN
    variabile3 := TRUE;
END_IF;
```

Significa: **SE** *variabile1* è **uguale** a **TRUE** e anche *variabile2* è **uguale** a **TRUE**, allora assegna a *variabile3* il valore **TRUE**.

AND è un operatore logico e significa che entrambe le condizioni devono essere vere.

In questo caso, l'assegnazione a *variabile3* viene eseguita solo quando tutte e due le condizioni risultano vere.

Controllare se almeno una condizione è vera

Scriviamo:

```
IF variabile1 = TRUE OR variabile2 = TRUE THEN
    variabile3 := TRUE;
END_IF;
```

Significa: **SE** *variabile1* è **uguale** a **TRUE** oppure *variabile2* è **uguale** a **TRUE**, allora assegna a *variabile3* il valore **TRUE**.

OR è un operatore logico e significa che è sufficiente che almeno una delle condizioni sia vera.

In questo caso, l'assegnazione a *variabile3* viene eseguita se *variabile1*, *variabile2* oppure entrambe risultano vere.

Invertire una condizione

Scriviamo:

```
IF NOT variabile1 THEN
    variabile2 := TRUE;
END_IF;
```

Significa: **SE** **variabile1** non è **TRUE**, **allora** assegna a **variabile2** il valore **TRUE**.

NOT è un operatore logico e serve a invertire il valore di una condizione.

Se **variabile1** è **TRUE**, con **NOT** la condizione diventa falsa.

Se **variabile1** è **FALSE**, con **NOT** la condizione diventa vera.

Confrontare due valori

Scriviamo:

```
IFvariabile1>variabile2THEN
```

```
variabile3:=TRUE;
```

```
END_IF;
```

Significa: **SE** **variabile1** è **maggiore** di **variabile2**, **allora** assegna a **variabile3** il valore **TRUE**.

> è un operatore di confronto e significa **maggiore di**.

In questo caso, l'assegnazione a **variabile3** viene eseguita solo se il valore di **variabile1** è **maggiore** del valore di **variabile2**.

Confrontare un valore minore

Scriviamo:

```
IFvariabile1<variabile2THEN
```

```
variabile3:=TRUE;
```

```
END_IF;
```

Significa: **SE** **variabile1** è **minore** di **variabile2**, **allora** assegna a **variabile3** il valore **TRUE**.

< è un operatore di confronto e significa **minore di**.

In questo caso, l'assegnazione a **variabile3** viene eseguita solo se il valore di **variabile1** è **minore** del valore di **variabile2**.

Confrontare un valore maggiore o uguale

Scriviamo:

```
IFvariabile1>=variabile2THEN
```

```
variabile3:=TRUE;
```

```
END_IF;
```

Significa: **SE variabile1 è maggiore o uguale a variabile2, allora** assegna a **variabile3** il valore **TRUE**.

>= è un operatore di confronto e significa **maggiore o uguale a**.

In questo caso, l'assegnazione a **variabile3** viene eseguita se il valore di **variabile1** è **maggiore o uguale** al valore di **variabile2**.

Confrontare un valore minore o uguale

Scriviamo:

```
IF variabile1 <= variabile2 THEN
  variabile3 := TRUE;
END_IF;
```

Significa: **SE variabile1 è minore o uguale a variabile2, allora** assegna a **variabile3** il valore **TRUE**.

<= è un operatore di confronto e significa **minore o uguale a**.

In questo caso, l'assegnazione a **variabile3** viene eseguita se il valore di **variabile1** è **minore o uguale** al valore di **variabile2**.

Verificare se due valori sono uguali

Scriviamo:

```
IF variabile1 = variabile2 THEN
  variabile3 := TRUE;
END_IF;
```

Significa: **SE variabile1 è uguale a variabile2, allora** assegna a **variabile3** il valore **TRUE**.

= è un operatore di confronto e significa **uguale a**.

In questo caso, l'assegnazione a **variabile3** viene eseguita solo se il valore di **variabile1** è **uguale** al valore di **variabile2**.

Verificare se due valori sono diversi

Scriviamo:

```
IF variabile1 <> variabile2 THEN
  variabile3 := TRUE;
END_IF;
```

Significa: **SE variabile1 è diversa da variabile2, allora** assegna a **variabile3** il valore **TRUE**.

<> è un operatore di confronto e significa **diverso da**.

In questo caso, l'assegnazione a **variabile3** viene eseguita solo se il valore di **variabile1** è **diverso** dal valore di **variabile2**.

Aggiungere una seconda condizione

Scriviamo:

```
IF variabile1 = TRUE THEN
    variabile3 := TRUE;
ELSIF variabile2 = TRUE THEN
    variabile3 := FALSE;
END_IF;
```

Significa: **SE variabile1 è uguale a TRUE, allora** assegna a **variabile3** il valore **TRUE**; **ALTRIMENTI, SE variabile2 è uguale a TRUE**, assegna a **variabile3** il valore **FALSE**.

ELSIF permette di controllare una seconda condizione quando la prima condizione dell'**IF** non è vera.

La condizione introdotta da **ELSIF** viene quindi controllata solo se la condizione precedente è risultata falsa.

Aggiungere un'alternativa finale

Scriviamo:

```
IF variabile1 = TRUE THEN
    variabile3 := TRUE;
ELSIF variabile2 = TRUE THEN
    variabile3 := FALSE;
ELSE
    variabile3 := TRUE;
END_IF;
```

Significa: **SE variabile1 è uguale a TRUE, allora** assegna a **variabile3** il valore **TRUE**; **ALTRIMENTI, SE variabile2 è uguale a TRUE**, assegna a **variabile3** il valore **FALSE**; **ALTRIMENTI, con ELSE**, assegna a **variabile3** il valore **TRUE**.

La parte introdotta da **ELSE** viene eseguita solo quando tutte le condizioni precedenti sono risultate false.

Assegnare un valore numerico

Scriviamo:

```
variabile1b:=b10;
```

Significa assegnare alla **variabile1** il valore numerico **10**.

10 è un valore numerico.

A differenza di **TRUE** e **FALSE**, che rappresentano valori logici, un numero può essere utilizzato per rappresentare per esempio una temperatura, una quantità, un conteggio o una misura.

Sommare due valori

Scriviamo:

```
variabile3b:=bvariabile1b+bvariabile2;
```

Significa assegnare a **variabile3** il risultato della somma tra **variabile1** e **variabile2**.

+ è l'operatore di addizione.

Se **variabile1** vale **10** e **variabile2** vale **5**, a **variabile3** viene assegnato il valore **15**.

Sottrarre due valori

Scriviamo:

```
variabile3b:=bvariabile1b-bvariabile2;
```

Significa assegnare a **variabile3** il risultato della sottrazione tra **variabile1** e **variabile2**.

- è l'operatore di sottrazione.

Se **variabile1** vale **10** e **variabile2** vale **5**, a **variabile3** viene assegnato il valore **5**.

Moltiplicare due valori

Scriviamo:

```
variabile3b:=bvariabile1b*bvariabile2;
```

Significa assegnare a **variabile3** il risultato della **moltiplicazione** tra **variabile1** e **variabile2**.

***** è l'operatore di moltiplicazione.

Se **variabile1** vale **10** e **variabile2** vale **5**, a **variabile3** viene assegnato il valore **50**.

Dividere due valori

Scriviamo:

```
variabile3 = variabile1 / variabile2;
```

Significa assegnare a **variabile3** il risultato della **divisione** tra **variabile1** e **variabile2**.

/ è l'operatore di divisione.

Se **variabile1** vale **10** e **variabile2** vale **5**, a **variabile3** viene assegnato il valore **2**.