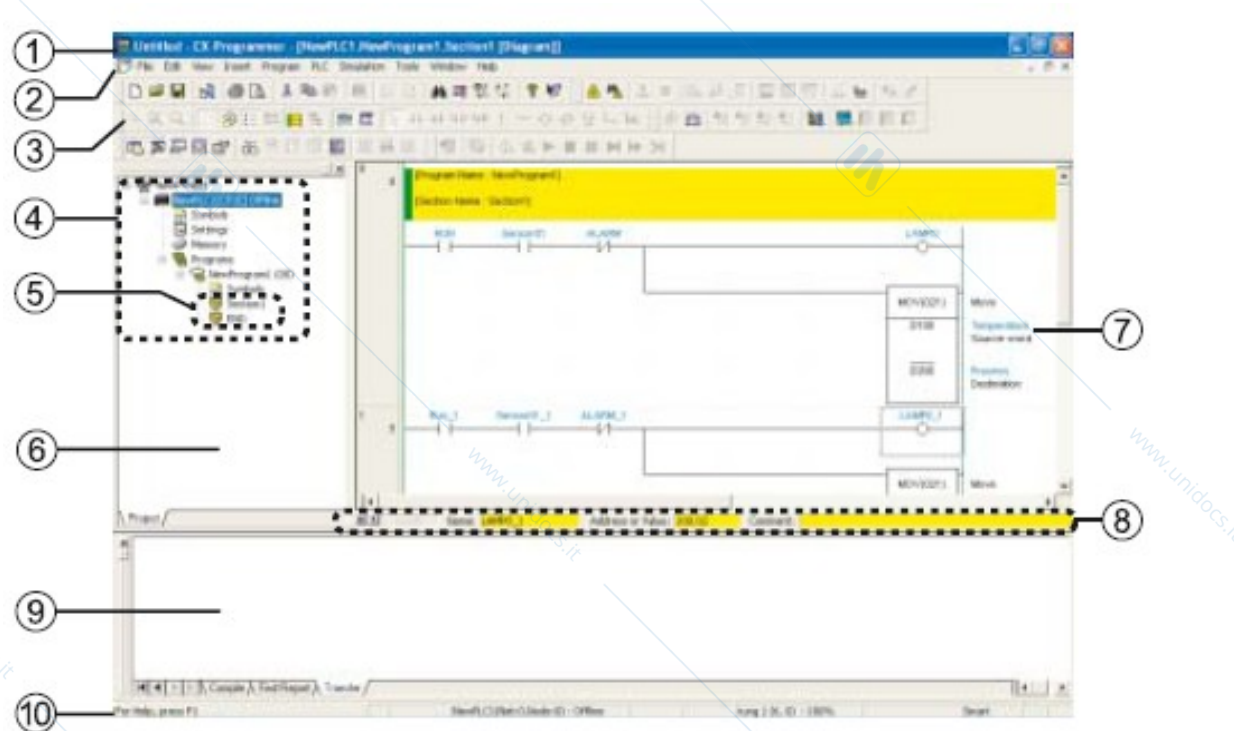
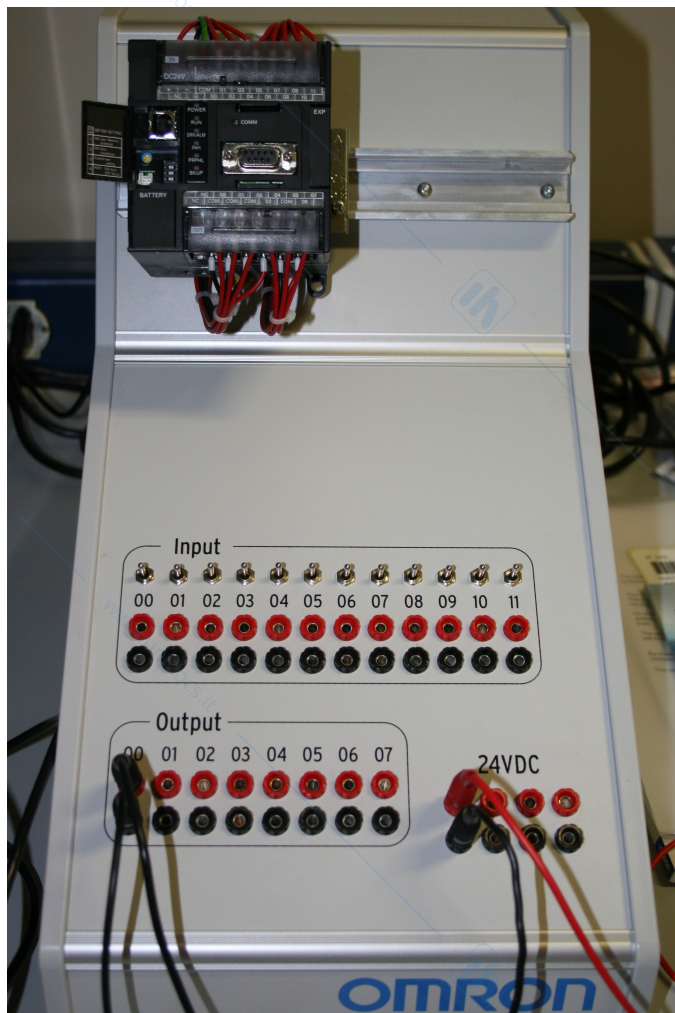


# Programmazione dei PLC in linguaggio Ladder

## Prima Parte



ITI "Alessandro Volta" - Sassuolo  
Anno Scolastico 2009/2010

# Linguaggio LADDER (linguaggio “a contatti”)

## Introduzione e concetti fondamentali

- Il *LADDER* (detto anche “*linguaggio a contatti*” o “*linguaggio degli elettricisti*”) è un *linguaggio di tipo grafico* ed è storicamente il primo che è stato adottato per programmare il PLC, in quanto consentiva una traduzione quasi immediata tra schemi funzionali di potenza e/o comando e programma per lo stesso;
- Nelle attuali versioni, molto evolute rispetto a quelle iniziali, gli elementi principali del linguaggio sono:

### Elementi di tipo “elettrico”:

- Binari (o linee) di potenza (power rail);
- Contatti elettrici NA o NC;
- Bobine di uscita (chiamate anche “relè” o “coil”);
- Relè “interni” e “speciali”.

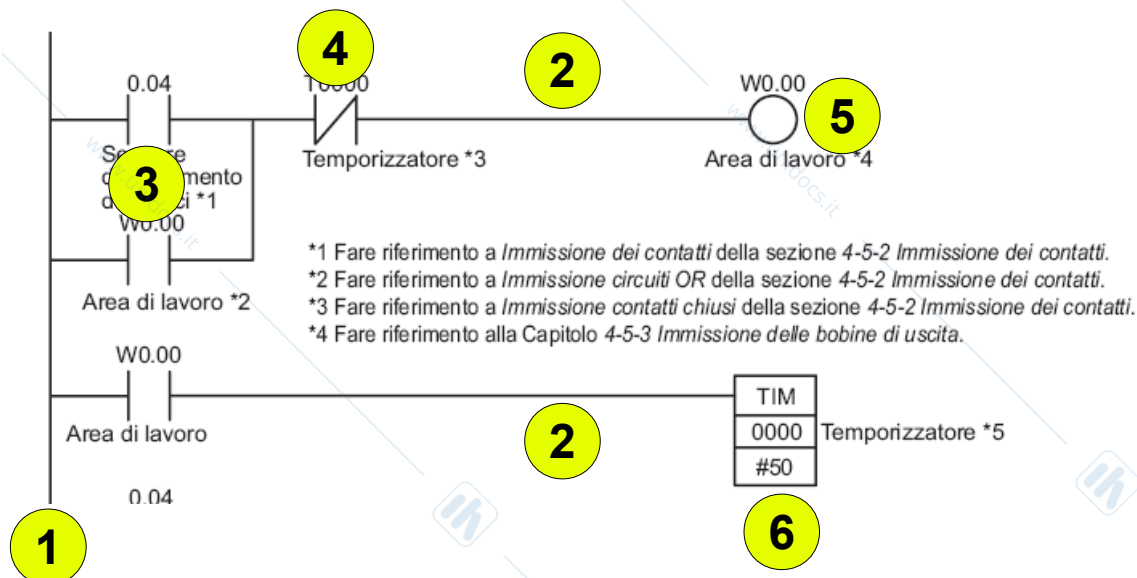
### Elementi di tipo “blocco funzione”:

- Contatori;
- Temporizzatori;
- Funzioni matematiche, logiche, di trattamento dati ecc. (le tipologie dipendono largamente dal modello di PLC utilizzato).

# Struttura base di un programma LADDER

- Il programma si articola in linee orizzontali dette “rung”, che si sviluppano tra i due binari di potenza sinistro e destro;
- Ogni rung inizia obbligatoriamente dal binario di potenza sinistro, mentre il collegamento col binario di potenza di destra è spesso facoltativo (generalmente i software di programmazione effettuano di default questa connessione laddove è richiesta);
- Ciascun rung può contenere contatti, bobine di uscita, blocchi funzione, relè di uscita interni o speciali ecc..

## ESEMPIO



- **(1)** - Binario di potenza sinistro;
- **(2)** - Rung (in questo esempio è presente il collegamento con il binario di potenza sinistro, mentre è sottintesa la connessione con il destro);
- **(3)** - Contatti NA in parallelo (OR);
- **(4)** - Contatto NC in serie (AND);
- **(5)** - Relè di uscita interno;
- **(6)** - Blocco temporizzatore.

# Contatti



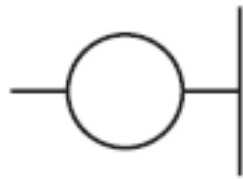
Contatto NA – Normalmente Aperto



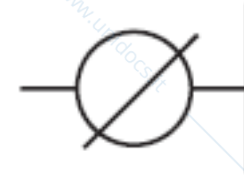
Contatto NC – Normalmente Chiuso

- I contatti sono in genere associati a dispositivi di ingresso fisicamente connessi al PLC (pulsanti, interruttori, finecorsa, sensori di prossimità ecc), ma possono anche essere riferiti ad altri elementi come bobine di uscita, relè interni, locazioni di memoria, blocchi funzione (contatori, temporizzatori ecc.);
- Ad ogni contatto viene associata una variabile binaria interna alla memoria del PLC che rappresenta lo stato elettrico del contatto stesso (ON/OFF);
- I contatti possono essere di due tipi, rappresentati dai simboli grafici illustrati nelle figure:
  - Contatto NA: la corrente circola quando il contatto viene azionato (a riposo essa non circola);
  - Contatto NC: la corrente circola con il contatto in posizione di riposo (a contatto azionato essa non circola);

# Bobine di uscita



Bobina NA



Bobina NC

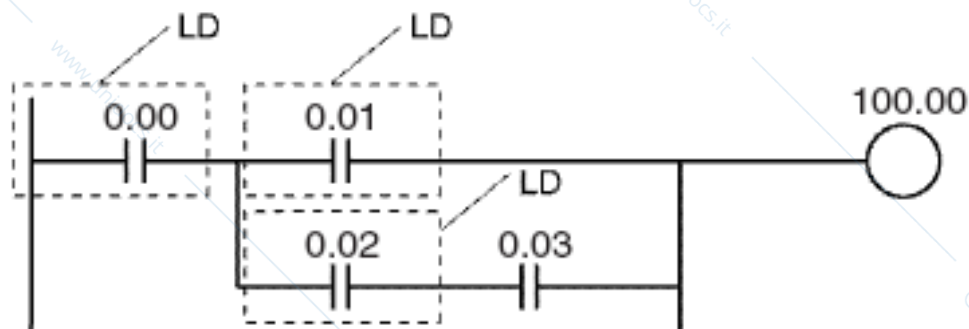
- Le bobine (o relè) di uscita sono il modo più comune con il quale si termina un rung di linguaggio ladder: come si vede infatti dalle figure esse hanno uno dei terminali collegato al binario di potenza destro (a volte tale collegamento è sottinteso oppure viene effettuato in automatico dai software di programmazione);
- La bobina NA (normalmente aperta) si attiva quando nel rung corrispondente circola corrente, mentre la bobina NC (normalmente chiusa) si attiva in caso contrario;
- Generalmente le bobine vengono riferite a:
  - uscite fisiche del PLC: in questo caso determinano lo stato del dispositivo di uscita corrispondente, attraverso il quale viene comandato il relativo attuatore;
  - Relè o aree di memoria interne al PLC: in questo caso hanno la funzione di memorizzare il risultato dell'elaborazione di un determinato rung di programma da utilizzare successivamente.

# Uso di contatti e bobine

- I contatti e le bobine di uscita sono sicuramente le istruzioni ladder più utilizzate nella programmazione dei PLC, in quanto fanno riferimento, rispettivamente, a dispositivi di input (pulsanti, finecorsa, sensori di prossimità, fotocellule ecc.) o di output (lampade di segnalazione, elettrovalvole, teleruttori ecc,) praticamente sempre presenti in qualunque tipo di impianto;
- Le regole che normalmente si utilizzano per la scrittura di rung con contatti e bobine sono le seguenti:
  - In ogni rung (che, ricordiamo, *inizia sempre dal binario di potenza sinistro*) può essere inserito un numero teoricamente illimitato di contatti, che possono essere connessi tra loro in modo da ottenere le funzioni logiche AND e OR:
    - Due o più contatti connessi in serie realizzano la funzione logica AND;
    - Due o più contatti connessi in parallelo realizzano la funzione logica OR.
  - Ogni rung termina generalmente con una sola bobina di uscita connessa al binario di potenza destro; le bobine non possono essere collegate tra loro in serie, ma si possono inserire più bobine di uscita in parallelo in modo che il rung le possa comandare contemporaneamente.

# Esempio

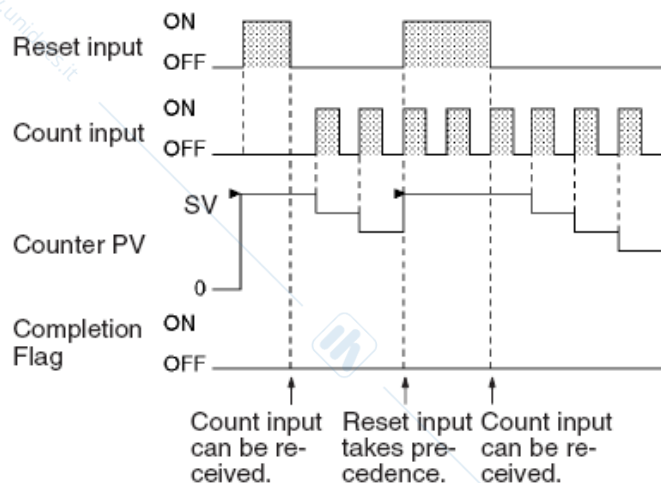
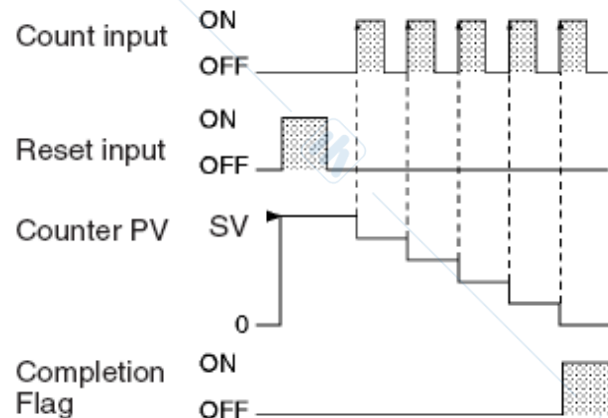
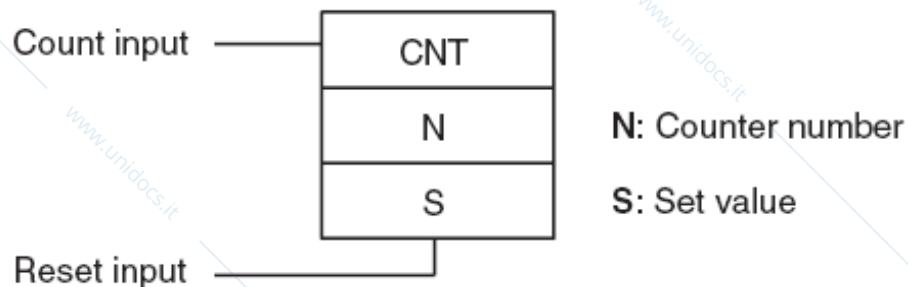
## (Dal manuale di programmazione del PLC Omron CP1L)



- La figura mostra un rung nel quale è presente una combinazione di 4 contatti di ingresso che comandano una bobina di uscita (è sottintesa la connessione con il binario di potenza destro);
- I contatti e la bobina vengono contrassegnati da un indirizzo che li identifica univocamente all'interno del PLC CP1L : p. es. 0.00 sta ad indicare il punto di ingresso 00 del modulo di input 0; 100.00 sta ad indicare il punto di uscita 00 del modulo di output 100;
- Tenendo presente che i contatti in serie realizzano la funzione logica AND, quelli in parallelo la funzione OR e che tutti i contatti sono di tipo NA, lo stato della bobina di uscita è dato dalla tabella di verità riportata a fianco.

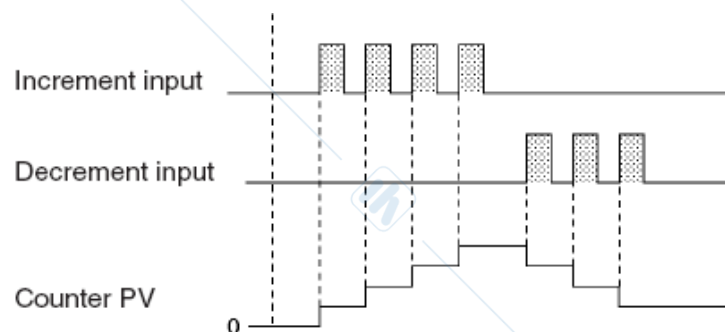
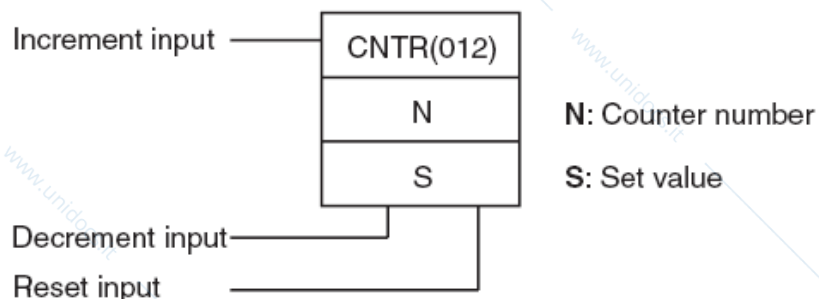
Stato contatti				Stato bobina
<b>0.00 (NA)</b>	<b>0.01 (NA)</b>	<b>0.02 (NA)</b>	<b>0.03 (NA)</b>	<b>100.00 (NA)</b>
OFF	OFF	OFF	OFF	OFF
OFF	OFF	OFF	ON	OFF
OFF	OFF	ON	OFF	OFF
OFF	OFF	ON	ON	OFF
OFF	ON	OFF	OFF	OFF
OFF	ON	OFF	ON	OFF
OFF	ON	ON	OFF	OFF
OFF	ON	ON	ON	OFF
ON	OFF	OFF	OFF	OFF
ON	OFF	OFF	ON	OFF
ON	OFF	ON	OFF	OFF
ON	OFF	ON	ON	ON
ON	ON	OFF	OFF	ON
ON	ON	OFF	ON	ON
ON	ON	ON	OFF	ON
ON	ON	ON	ON	ON

# Contatori a decremento

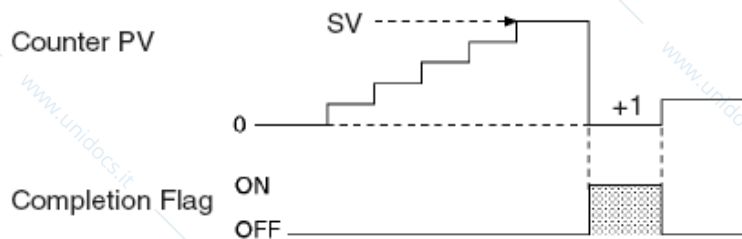


- Il blocco funzione **CNT (Counter)** permette di inserire nel programma un contatore a decremento con settaggio del valore iniziale;
- Il contatore presenta due ingressi:
  - Count input: viene connesso al dispositivo che manda gli impulsi al contatore (p. es. la fotocellula che rileva il passaggio di una persona attraverso una porta); ad ogni impulso presente sull'ingresso il contatore decrementa di 1 il valore corrente;
  - Reset input: viene connesso al comando che permette di resettare il contatore al valore iniziale (p. es. un pulsante).
- Quando si inserisce un contatore in un programma si forniscono:
  - Il numero del contatore al quale si fa riferimento (da 0 a 4095 per il CP1L);
  - Il valore iniziale di settaggio del contatore (da 0 a 9999).
- Il funzionamento del contatore è descritto dai grafici riportati a fianco:
  - Un impulso sull'ingresso reset input pone  $PV = S$  (PV: valore corrente del contatore);
  - Ad ogni impulso presente su count input PV viene decrementato di 1;
  - Quando  $PV = 0$  il contatore si attiva, eccitando in questo modo la bobina ad esso collegata;
  - Il contatore può essere resettato in qualunque momento portando alto l'ingresso reset input.;
  - Se reset input viene mantenuto alto il conteggio è bloccato.

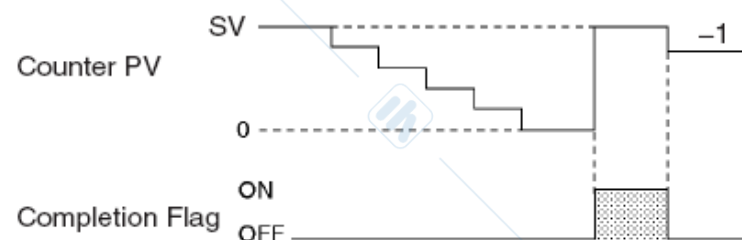
# Contatori reversibili (Up/Down)



1



2



3

- I contatori reversibili (Up/Down) permettono di ottenere sia il conteggio a incremento (UP) che quello a decremento (DOWN);
- Come risulta dalla figura a fianco sono del tutto simili ai precedenti, ad eccezione del fatto che sono dotati di 3 terminali di ingresso:
  - Increment input: conteggio in avanti (UP);
  - Decrement input: conteggio all'indietro (DOWN);
  - Reset input: reset del contatore al valore iniziale (SV);
- Il funzionamento del contatore è illustrato dai grafici a fianco:
  - **(1)** - Il valore corrente del contatore (PV) può "fluttuare" tra 0 ed il valore di settaggio (SV);
  - **(2)** - Nel conteggio UP il contatore incrementa di 1 ad ogni impulso: raggiunto il valore di settaggio si ha l'attivazione del contatore al successivo impulso che ne determina l'azzeramento;
  - **(3)** - Nel conteggio DOWN il funzionamento è del tutto simile a quanto detto sopra, con la differenza che il contatore decrementa di 1 ad ogni impulso.

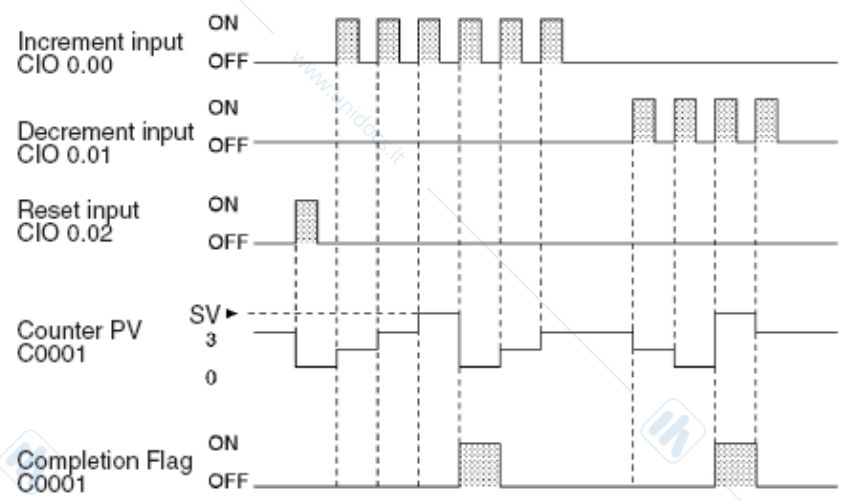
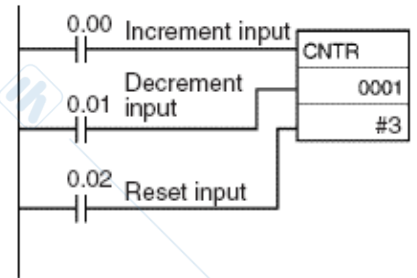
# Esempio di utilizzo di un contatore Up/Down

## Dal manuale di programmazione del PLC CP1L

### Basic Operation of CNTR(012)/CNTRX(548)

The counter PV is reset to 0 by turning the reset input (CIO 0.02) ON and OFF. The PV is incremented by 1 each time that the increment input (CIO 0.00) goes from OFF to ON. When the PV is incremented from the SV (3), it is automatically reset to 0 and the Completion Flag is turned ON.

Likewise, the PV is decremented by 1 each time that the decrement input (CIO 0.01) goes from OFF to ON. When the PV is decremented from 0, it is automatically set to the SV (3) and the Completion Flag is turned ON.



# Temporizzatori (Timer)

PV refresh method	Symbol	Operands
BCD		N: 0000 to 4095 (decimal) S: #0000 to #9999 (BCD)
Binary		N: 00000 to 4095 (decimal) S: &0 to &65535 (decimal) #0000 to #FFFF (hex)

I **timer** vengono utilizzati per creare un ritardo all'eccitazione con precisione di 0.1s rispetto all'attivazione di un segnale di comando;

Il timer ha un solo ingresso, al quale va connesso il contatto che corrisponde al segnale di attivazione del timer stesso;

Quando si inserisce un timer in un programma occorre specificare:

- Il numero del timer N (da 0 a 4095);
- Il valore di settaggio S, che determina il ritardo all'eccitazione.

Ci sono due tipi di timer, che differiscono tra loro per il valore max del ritardo all'eccitazione:

$$TIM - 0.1s \times 9999 = 999.9s;$$

$$TIMX - 0.1s \times 65535 = 6553.5s;$$

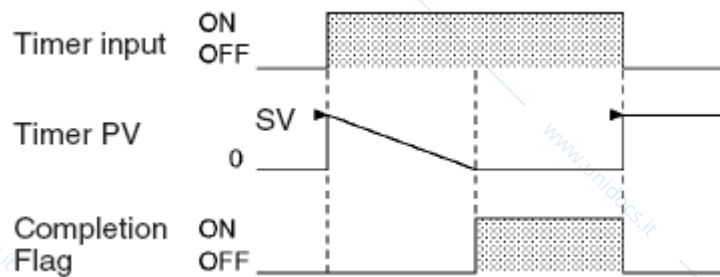
Il ritardo all'eccitazione (in secondi) si ottiene moltiplicando S per 0.1; ad esempio con S = 100 si ottiene un ritardo di  $0.1s \times 100 = 10s$ ;

Il funzionamento dei timer è illustrato nei grafici riportati a fianco:

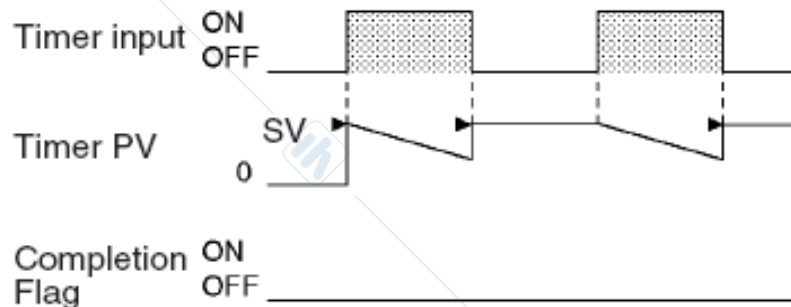
(1) - Quando il segnale di input viene portato alto il valore corrente del timer inizia a decrementare partendo da S: quando PV raggiunge il valore 0 il timer si attiva e rimane in questo stato fino a che l'ingresso ritorna basso;

(2) - Se l'ingresso viene riportato basso prima dell'azzeramento di PV il timer viene resettato al valore iniziale senza che si abbia alcuna eccitazione.

1



2



# Temporizzatori veloci (High Speed Timer)

PV refresh method	Symbol	Operands			
BCD	<table border="1"> <tr><td>TIMH(015)</td></tr> <tr><td>N</td></tr> <tr><td>S</td></tr> </table> <p>N: Timer number S: Set value</p>	TIMH(015)	N	S	N: 0000 to 4095 (decimal) S: #0000 to #9999 (BCD)
TIMH(015)					
N					
S					
Binary	<table border="1"> <tr><td>TIMHX(551)</td></tr> <tr><td>N</td></tr> <tr><td>S</td></tr> </table> <p>N: Timer number S: Set value</p>	TIMHX(551)	N	S	N: 00000 to 4095 (decimal) S: &0 to &65535 (decimal) #0000 to #FFFF (hex)
TIMHX(551)					
N					
S					

PV refresh method	Symbol	Operands			
BCD	<table border="1"> <tr><td>TMHH(540)</td></tr> <tr><td>N</td></tr> <tr><td>S</td></tr> </table> <p>N: Timer number S: Set value</p>	TMHH(540)	N	S	N: 0000 to 15 (decimal) S: #0000 to #9999 (BCD)
TMHH(540)					
N					
S					
Binary	<table border="1"> <tr><td>TMHHX(552)</td></tr> <tr><td>N</td></tr> <tr><td>S</td></tr> </table> <p>N: Timer number S: Set value</p>	TMHHX(552)	N	S	N: 00000 to 15 (decimal) S: &0 to &65535 (decimal) #0000 to #FFFF (hex)
TMHHX(552)					
N					
S					

- I timer veloci funzionano con le stesse modalità dei precedenti ma con valori di precisione più bassi: questo significa che essi sono in grado di realizzare ritardi all'eccitazione con maggiore accuratezza rispetto ai precedenti;
- I timer TIMH e TIMHX hanno una precisione di 10ms;
- I timer TIMHH e TIMHHX hanno una precisione di 1ms; inoltre, come risulta dalla figura a fianco, per essi N (numero del timer) varia tra 0 e 15.

# Bibliografia - Sitografia

- Massimo Barezzi  
*PLC – Controllori Logici Programmabili*  
*Programmazione, applicazioni ed esercitazioni*  
Editrice San Marco
- Manuali tecnici OMRON:
  - CPU CP1L  
*Guida introduttiva*
  - CPU CP1L  
*Programming manual*
  - CPU CP1L  
*Guida di avvio rapido*
- I manuali elencati sopra possono essere scaricati gratuitamente in formato PDF dal sito di OMRON ITALIA al seguente indirizzo:

**<http://industrial.omron.it/it/home/default.html>**

dal quale è possibile reperire ulteriore documentazione tecnica relativa ai PLC CP1L e altri prodotti OMRON.