

CPS1

Corso Base PLC

Argomenti del Corso

1. Introduzione
2. I segnali
3. Il PLC
4. Sistemi di codifica
5. I PLC compatti
6. Introduzione alla programmazione
7. Le funzioni
8. I PLC modulari
9. Istruzioni aritmetiche
10. Gestione dei segnali analogici
11. CX-Simulator

***Introduzione
all'automazione
industriale***

Di cosa si tratta?

L'automazione industriale è una disciplina che studia metodi e tecnologie per il controllo di flussi di energia, di materiali e informazioni necessari alla realizzazione dei processi produttivi.

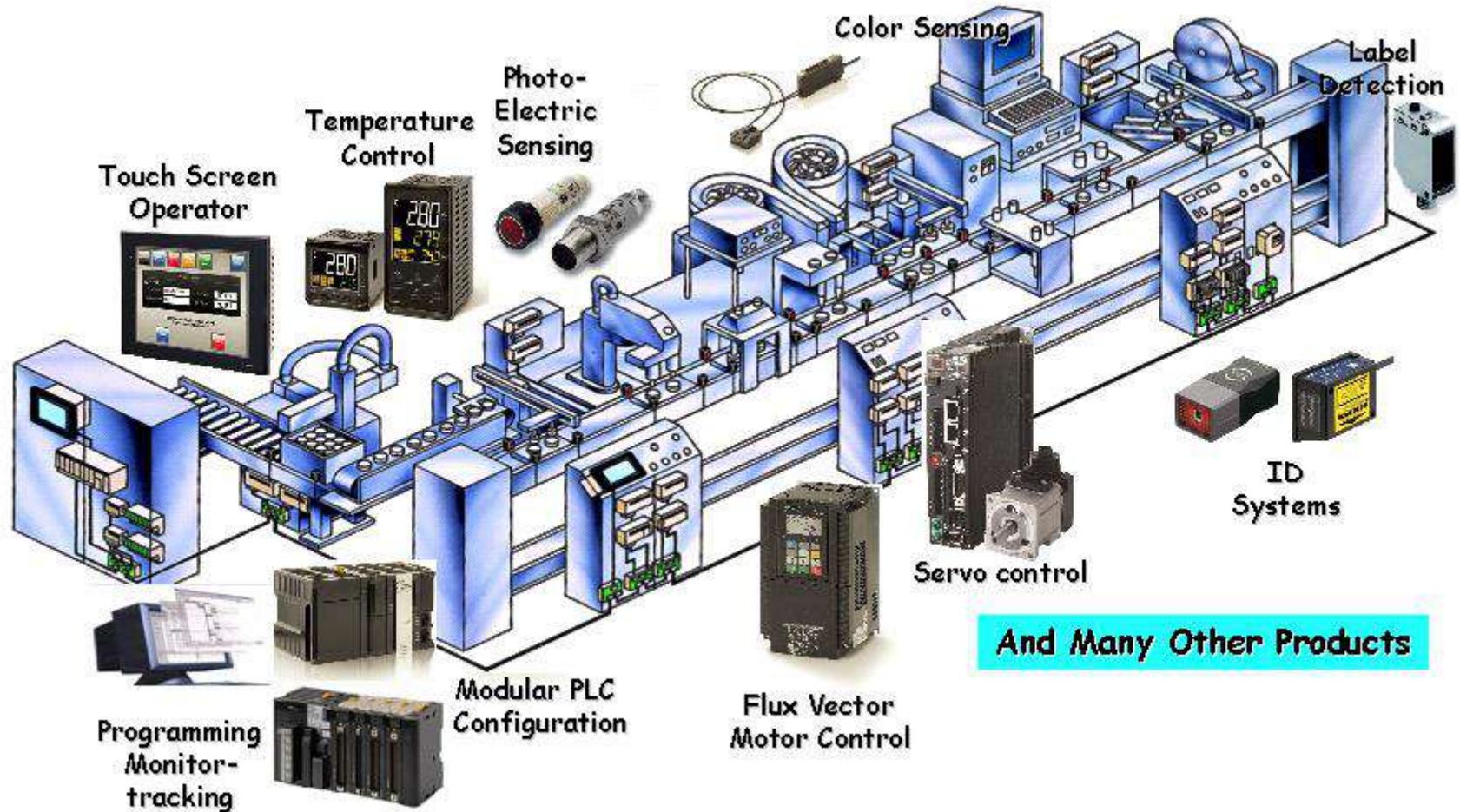
L'automazione nei processi produttivi porta una serie di vantaggi:

- Miglioramento della qualità dei prodotti
- Flessibilità degli impianti (possibilità di utilizzare uno stesso impianto per realizzare più tipologie di prodotti)
- Riduzione dei tempi di produzione
- Abbattimento del costo di produzione
- Riduzione degli scarti di lavorazione
- Risparmio energetico

Il sistema di controllo

- Un sistema di controllo di un processo industriale può essere descritto come un insieme di dispositivi interconnessi e comunicanti tra loro attraverso una o più reti di comunicazione
- Un dispositivo è caratterizzato da componenti hardware o software che permettono di elaborare i dati e comunicare con l'esterno
- Il sistema di controllo legge le informazioni dai sensori, elabora le informazioni e agisce sul processo attraverso gli attuatori
- Il sistema di controllo permette di registrare dati, creare report, statistiche, allarmi...

I prodotti per l'automazione industriale



And Many Other Products

Elementi fondamentali di un sistema di controllo moderno

- PLC: Programmable Logic Controller (controllore logico programmabile)
- Sensori
- Attuatori
- HMI: Human Machine Interface (interfaccia uomo macchina)

Il PLC

- Acquisisce i segnali dei sensori
- Elabora il programma contenuto nella sua memoria
- Fornisce i segnali di comando agli attuatori



Sensori

- Sensori: convertono una grandezza fisica in un segnale elettrico
 - Movimento, rotazione (encoder)
 - Distanza
 - Livello
 - Temperatura
 - Confronto immagini (visione)
 - Identificazione
 - Presenza (fotocellula, induttivo, capacitivo)



Attuatori

- Attuatori: convertono un segnale elettrico in una grandezza fisica
 - Inverter per il controllo della velocità dei motori
 - Servo azionamenti (per il controllo della posizione e della coppia)
 - Elementi di riscaldamento
 - Comandi ON/OFF



Interfacce operatore

- Interfacce operatore: vengono utilizzate per comandare una macchina e visualizzare eventuali stati funzionali
- HMI: Interfaccia uomo macchina
- IPC: PC industriale
- SCADA: software di supervisione



Il concetto di I/O

- Il termine I/O (input/output) indica l'associazione tra il segnale elettrico e il segnale digitale utilizzato dal PLC
- Tale associazione avviene sia per gli ingressi sia per le uscite
- In un sistema di controllo è importante definire prima ingressi e uscite per dimensionare correttamente il PLC

I Segnali

Ingressi/Uscite Digitali,
Ingressi/Uscite Analogici e Segnali ad
Alta Frequenza

I segnali rilevabili dal PLC

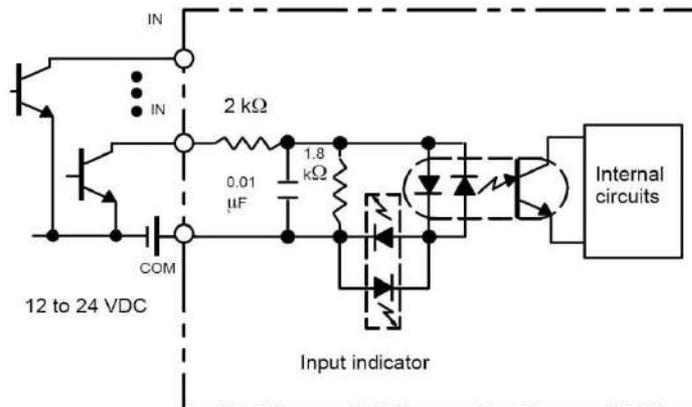
- I segnali elettrici possono essere:
 - Digitali
 - Standard
 - Alta Frequenza
 - Analogici
- Avremo quindi ingressi/uscite digitali e ingressi/uscite analogici
- Gli ingressi per segnali ad alta frequenza (conteggio veloce) vengono utilizzati ad esempio per lettura segnali provenienti da un encoder
- Le uscite a treno d'impulsi vengono utilizzate per pilotare servo azionamenti o inverter

Ingressi digitali

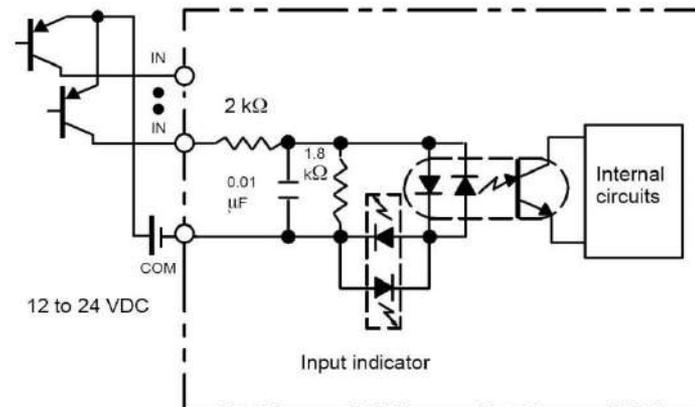
- Il segnale digitale fornisce un'informazione di tipo ON/OFF e viene interpretato dal PLC come la modifica di un bit
- Il livello fisico del segnale digitale è un segnale in tensione
- L'interfaccia fisica di ingresso del PLC deve essere in grado di rilevare il segnale
- Normalmente i segnali digitali rilevabili sono segnali in tensione a 12 o 24 Vc.c.
- È necessario avere informazione sulla logica del segnale ovvero se il segnale è di tipo PNP o di tipo NPN

Ingressi digitali: NPN e PNP

- Ingressi NPN



- Ingressi PNP



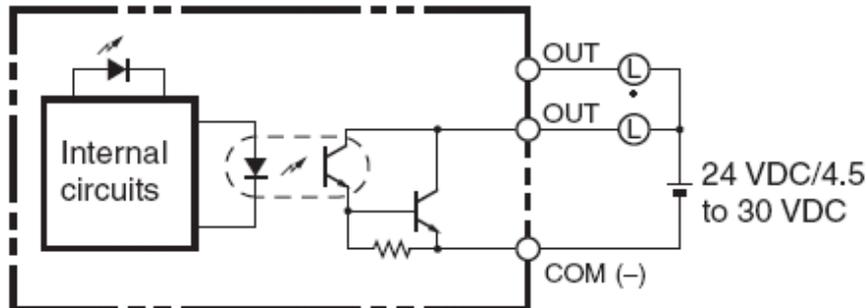
Uscite a transistor

- Il PLC comanda un transistor di uscita. L'entità della corrente sopportabile è dell'ordine delle centinaia di mA
- Il transistor può sopportare solo tensioni continue polarizzate
- La scelta del tipo di uscita a transistor (NPN o PNP) avviene in base al tipo di dispositivo da comandare
- Non ci sono limiti fisici al numero di commutazioni
- La frequenza di commutazione è dell'ordine di grandezza dei kHz (superiore rispetto all'uscita a relè)

Uscite a transistor: NPN e PNP

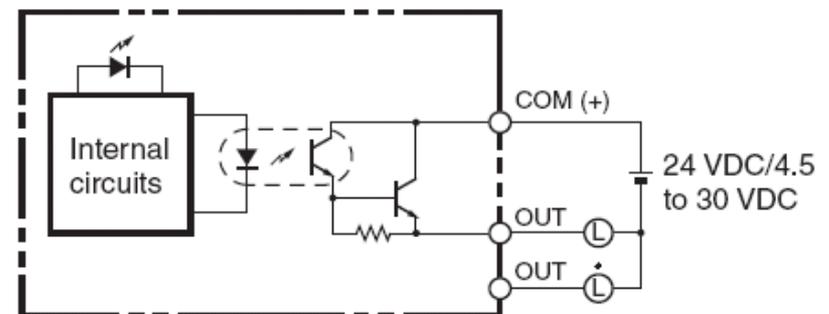
- NPN**

Quando l'uscita si attiva, il transistor conduce: l'OUT e COM vengono portati allo stesso potenziale (**0V**)



- PNP**

Quando l'uscita si attiva, il transistor conduce: l'OUT e COM vengono portati allo stesso potenziale (**24V**)

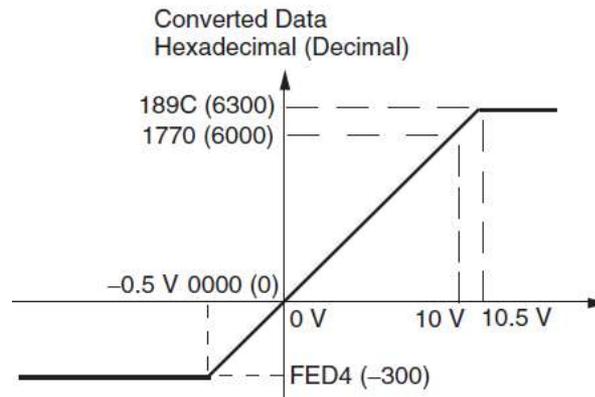


Uscita a relè

- Il PLC comanda un contatto di uscita che può sopportare correnti di una certa entità (qualche Ampere) e segnali in tensione sia continua sia alternata
- Un'uscita a relè ha un numero massimo di operazioni possibili dovuto allo spostamento fisico (frequenza di commutazione) e all'usura del contatto

Segnale analogico

- Il segnale analogico fornisce un'informazione di *tipo complesso*
- Il livello in ingresso, viene convertito in un valore digitale in funzione della risoluzione della scheda di acquisizione e copiato in un'area di memoria del PLC
- Allo stesso modo un valore digitale contenuto in un'area di memoria del PLC viene convertito in un valore di tensione di uscita
- Di seguito è riportato un esempio di conversione su un segnale di ingresso 0-10V (CP1W-MAD11)



Segnale analogico - Principali caratteristiche

- Risoluzione:
 - Indica la sensibilità nel rilevare variazioni del segnale analogico
- Tempo di campionamento:
 - indica la capacità di inseguire una variazione del segnale di ingresso.
- Tipi di segnali:
 - segnali in tensione 0..10V, -10..+10V, 0..5V o 1..5V
 - segnali in corrente 0..20 mA o 4..20 mA
- Precisione:
 - Percentuale massima di errore sul fondo scala e linearità

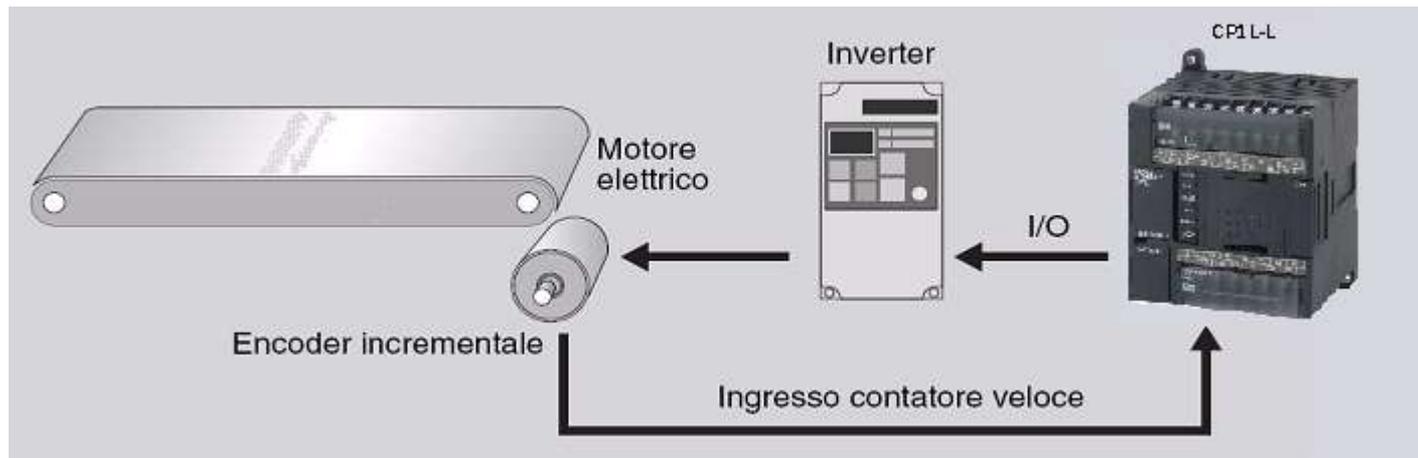
Segnali ad alta frequenza

- Ingresso contatore veloce

A bordo di tutte le CPU CP1L sono disponibili due contatori veloci fino a 100 kHz (in modalità bidirezionale impulso + direzione) o quattro da 100 kHz (monodirezionali)

- Uscita a treno d'impulsi

A bordo di tutte le CPU CP1L è possibile gestire due uscite a treno di impulsi (fino a 100 kHz)



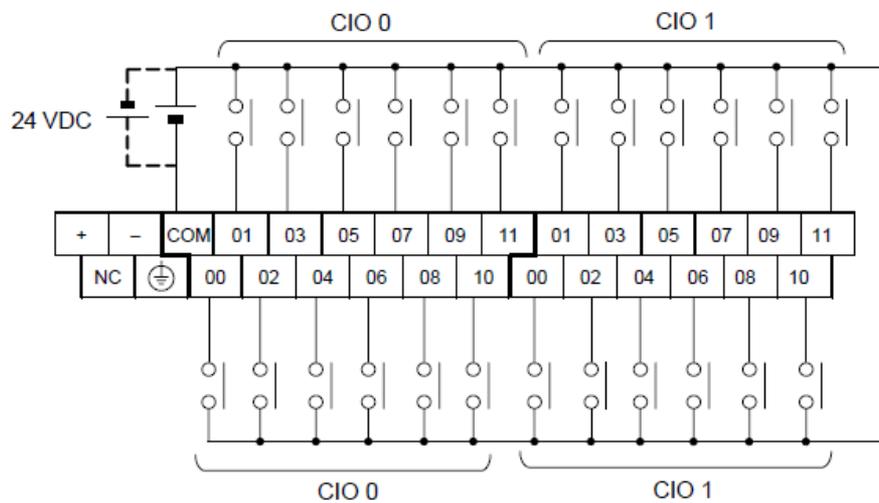
Segnali ad alta frequenza – CP1E

		CP1E-E			CP1E-N		CP1E-NA
		10	14 / 20	30 / 40 / 60	14 / 20	30 / 40 / 60	20
Contatori veloci	Mono direzionale incrementale	5x10 kHz	6x10 kHz	6x10 kHz	4x10 kHz, 2x100 kHz	4x10 kHz, 2x100 kHz	4x10 kHz, 2x100 kHz
	Impulso + direzione	2x10 kHz	2x10 kHz	2x10 kHz	2x100 kHz	2x100 kHz	2x100 kHz
	CW / CCW	2x10 kHz	2x10 kHz	2x10 kHz	1x10 kHz, 1x100 kHz	1x10 kHz, 1x100 kHz	1x10 kHz, 1x100 kHz
	Bidirezionale: fasi differenziali (x4) (A, B, Z)	2x5 kHz	2x5 kHz	2x5 kHz	1x5 kHz, 1x50 kHz	1x5 kHz, 1x50 kHz	1x5 kHz, 1x50 kHz
Uscite veloci	Impulso + direzione	---	---	---	2x100 kHz	2x100 kHz	2x100 kHz

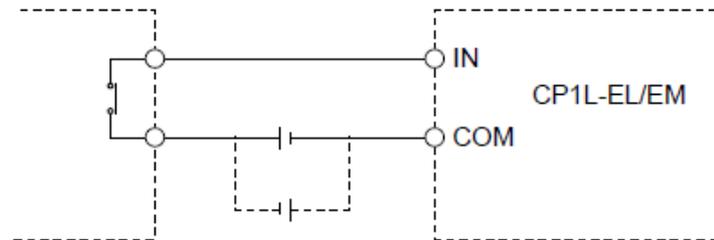
Segnali ad alta frequenza – CP1L e CP1H

		CP1L-L / M / EL / EM	CP1H-X / XA
Contatori veloci	Mono direzionale incrementale	4x100 kHz	4x100 kHz
	Impulso + direzione	2x100 kHz	4x100 kHz
	CW / CCW	2x100 kHz	4x100 kHz
	Bidirezionale: fasi differenziali (x4) (A, B, Z)	2x50 kHz	4x50 kHz
Uscite veloci	Impulso + direzione	2x100 kHz	4x100 kHz

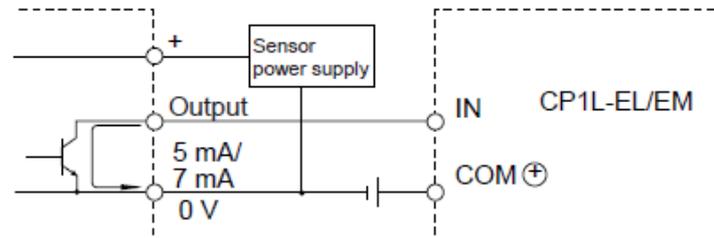
Ingressi digitali – Esempio di cablaggio



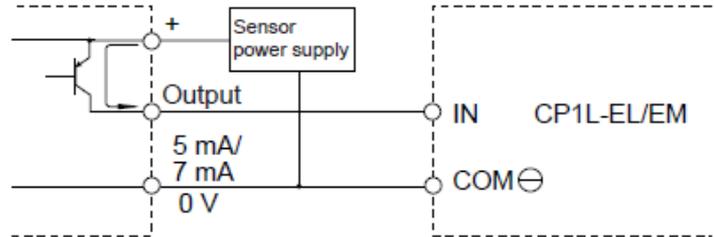
Contact output



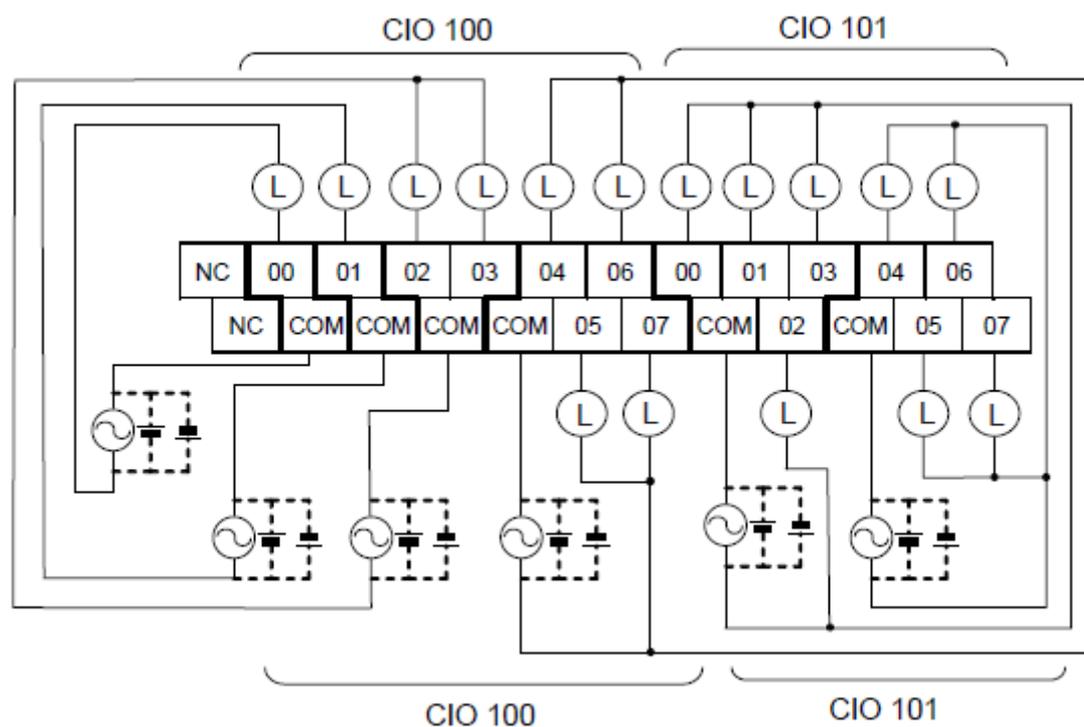
NPN open-collector output



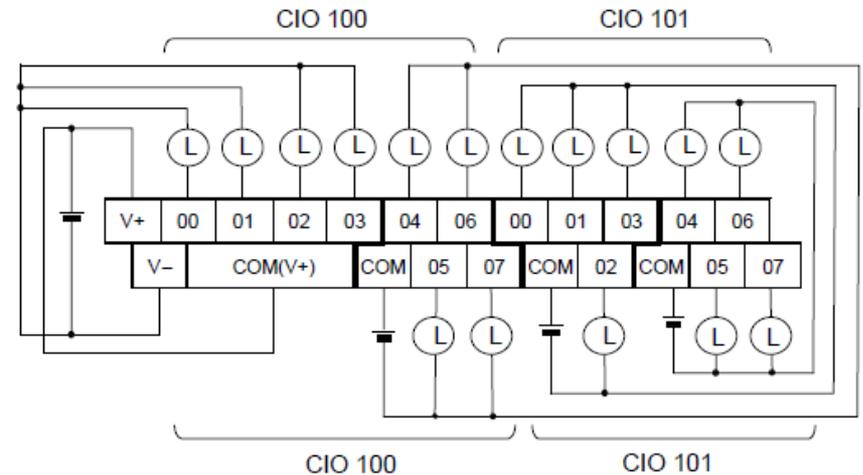
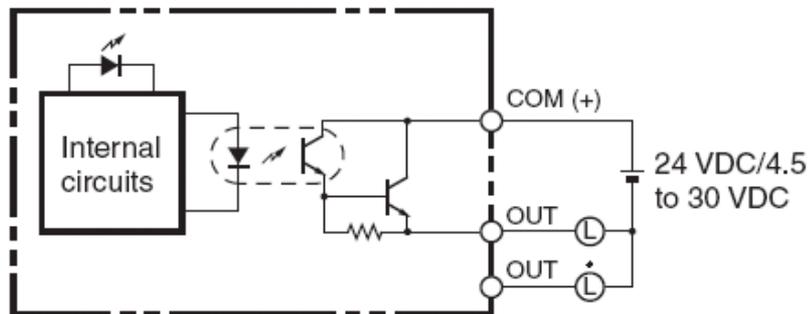
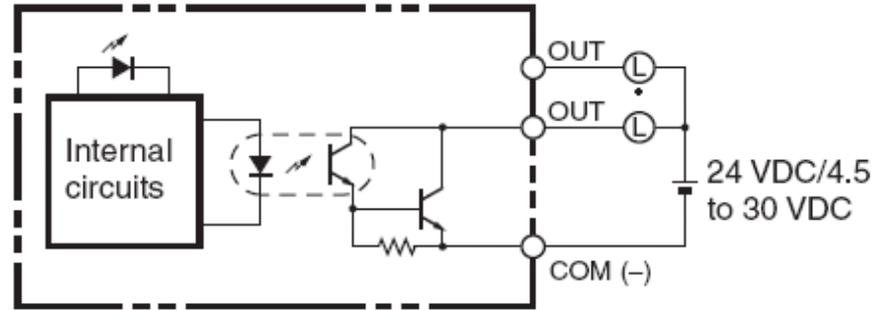
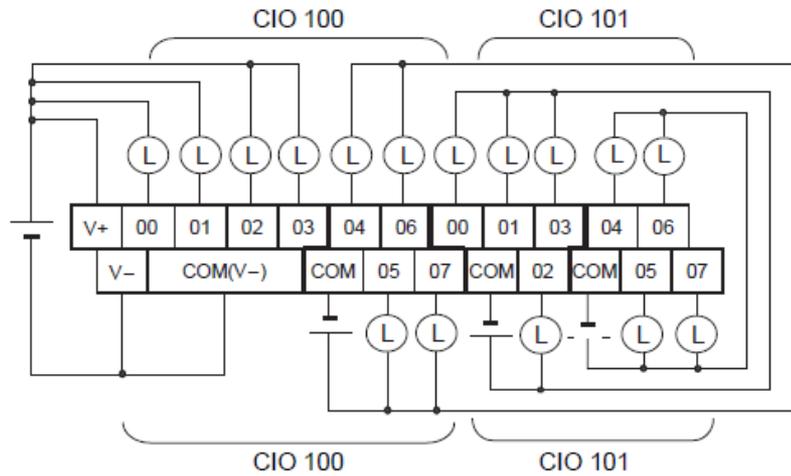
PNP current output



Uscite a relè – Esempio di cablaggio



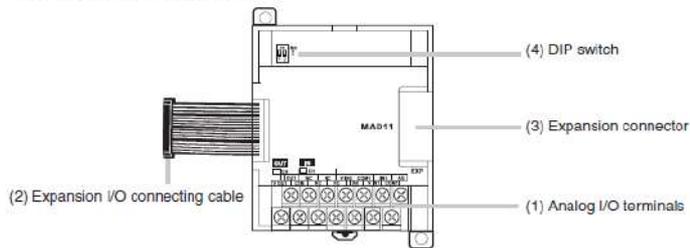
Uscite NPN e PNP – Esempio di cablaggio



Ingressi e uscite analogiche

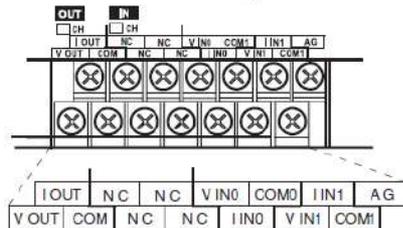
Esempio di cablaggio e specifiche (CP1W-MAD11)

CP1W-MAD11/CPM1A-MAD11



- (1) Analog I/O Terminals
Connected to analog I/O devices.

CPM1A-MAD11 Terminal Arrangements



Note For current inputs, short V IN0 to I IN0 and V IN1 to I IN1.

V OUT	Voltage output
I OUT	Current output
COM	Output common
V IN0	Voltage input 0
I IN0	Current input 0
COM0	Input common 0
V IN1	Voltage input 1
I IN1	Current input 1
COM1	Input common 1

Item		Voltage I/O	Current I/O	
Analog Input Section	Number of inputs	2 inputs (2 words allocated)		
	Input signal range	0 to 5 VDC, 1 to 5 VDC, 0 to 10 VDC, or -10 to 10 VDC	0 to 20 mA or 4 to 20 mA	
	Max. rated input	±15 V	±30 mA	
	External input impedance	1 MΩ min.	Approx. 250 Ω	
	Resolution	1/6000 (full scale)		
	Overall accuracy	25°C	0.3% full scale	0.4% full scale
		0 to 55°C	0.6% full scale	0.8% full scale
	A/D conversion data	16-bit binary (4-digit hexadecimal) Full scale for -10 to 10 V: F448 to 0BB8 hex Full scale for other ranges: 0000 to 1770 hex		
Averaging function	Supported (Settable for individual inputs via DIP switch)			
Open-circuit detection function	Supported			
Analog Output Section	Number of outputs	1 output (1 word allocated)		
	Output signal range	1 to 5 VDC, 0 to 10 VDC, or -10 to 10 VDC,	0 to 20 mA or 4 to 20 mA	
	Allowable external output load resistance	1 kΩ min.	600 Ω max.	
	External output impedance	0.5 Ω max.		
	Resolution	1/6000 (full scale)		
	Overall accuracy y	25°C	0.4% full scale	
		0 to 55°C	0.8% full scale	
	Set data (D/A conversion)	16-bit binary (4-digit hexadecimal) Full scale for -10 to 10 V: F448 to 0BB8 hex Full scale for other ranges: 0000 to 1770 hex		
Conversion time	2 ms/point (6 ms/all points)			
Isolation method	Photocoupler isolation between analog I/O terminals and internal circuits. No isolation between analog I/O signals.			
Current consumption	5 VDC: 83 mA max., 24 VDC: 110 mA max.			

II PLC

Struttura e logica di
funzionamento

PLC: definizione

- Programmable Logic Controller: Controllore a Logica Programmabile
- APPARECCHIATURA ELETTRONICA PROGRAMMABILE PER IL CONTROLLO DI MACCHINE E PROCESSI INDUSTRIALI
- Nasce come elemento sostitutivo della logica cablata e dei quadri di controllo a relè
- Si qualifica in breve tempo come elemento insostituibile nell'automazione di fabbrica

La logica CABLATA

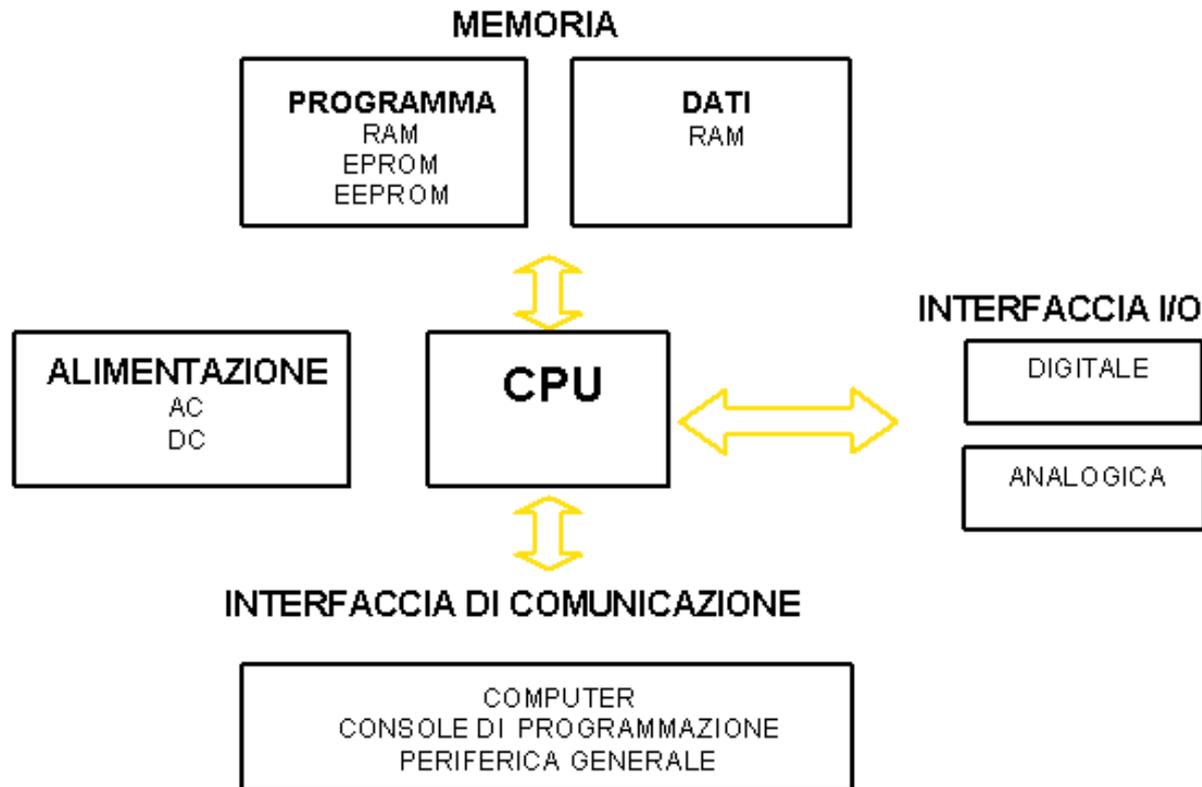
- La logica con cui vengono controllate le uscite è realizzata tramite dispositivi DISCRETI (Relè, Temporizzatori, Contatori,...)
 - E' difficilmente modificabile e ampliabile
 - E' di difficile diagnostica
 - E' difficile da interfacciare con computer stampanti, terminali...
 - Non è idonea per sistemi di controllo complessi

La logica PROGRAMMATA

- La logica di controllo è realizzata tramite una serie di ISTRUZIONI caricate nella Memoria di Programma
 - E' facile da ampliare e modificare
 - E' di facile diagnostica
 - E' facilmente interfacciabile con dispositivi quali PC, stampanti, terminali,
 - E' pensata per il controllo di sistemi complessi

PLC: la struttura

- In un PLC possiamo distinguere i seguenti blocchi funzionali



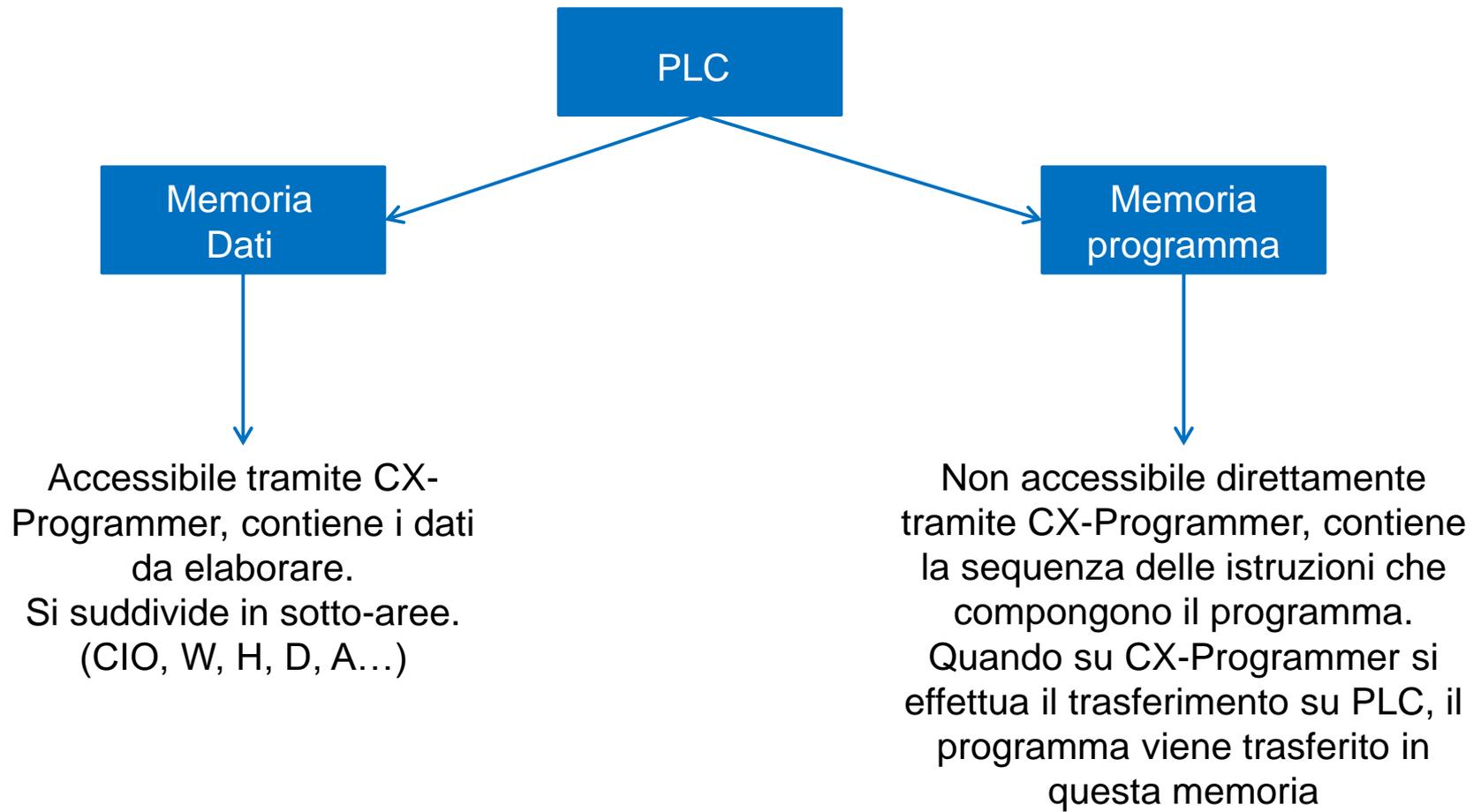
La CPU

- E' il "cervello" del sistema dove avvengono le elaborazioni delle informazioni e la gestione del PLC stesso
- Le sue caratteristiche sono:
 - la velocità di elaborazione
 - la memoria di programma
 - la memoria dati
 - il numero e il tipo di funzioni integrate disponibili
 - il numero di interfacce gestibili
 - I/O, Comunicazione e Memoria

La memoria

- Si divide in memoria dati e memoria programma
- La memoria **Programma** contiene la sequenza delle istruzioni di programma. È normalmente una memoria ritentiva (Flash memory o RAM tamponata da batteria o condensatore). Alcuni PLC dispongono di una memoria di backup EPROM/EEPROM o un supporto opzionale Compact flash standard
- La memoria **Dati** contiene i dati da elaborare: la memoria dati è suddivisa in sotto aree secondo la loro funzione specifica (Aree di memoria)

La memoria



Le interfacce e l'alimentatore

- **Interfaccia di I/O** (analogica o digitale): è l'hardware dedicato al collegamento di sensori e attuatori. Questa interfaccia permette di convertire il segnale fisico (elettrico) in un segnale logico (bit o word) dell'area di memoria
- **Interfaccia di comunicazione**: permette la comunicazione del PLC con altri dispositivi che hanno una porta seriale (PC, PLC, Inverter, MMI, lettori di codici a barre, termoregolatori, RFID...)
- **Alimentatore**: fornisce l'alimentazione a tutto il sistema PLC. Deve essere dimensionato in funzione della potenza assorbita dai moduli collegati

PLC compatti e modulari

- I PLC compatti sono composti da un unico “blocco” contenente alimentatore, CPU e I/O
- Tale configurazione non sempre consente di adattare il PLC alle esigenze del sistema da controllare



PLC compatti e modulari

- I PLC modulari sono composti da una serie di moduli collegati tra loro tramite bus di comunicazione interno
- Alimentatore, CPU e moduli di I/O sono quindi unità indipendenti e ordinabili separatamente



- Il PLC è liberamente configurabile in base alle necessità del sistema da controllare

PLC: logica di funzionamento

- L'elaborazione dei dati di un PLC avviene di norma in modo sequenziale secondo l'ordine delle istruzioni all'interno del programma
- La differenza con un normale computer dipende dal fatto che l'attuazione (segnale fisico sull'uscita) e l'acquisizione (rilevamento dell'ingresso) avvengono di norma alla fine dell'elaborazione (a meno di specifiche funzioni)

PLC: logica di funzionamento

- Possiamo considerare la logica di controllo di un PLC suddivisa in tre momenti:
 1. Acquisizione dei dati dai sensori (fotografia degli ingressi)
 2. Elaborazione dei dati + processi comuni
 3. Attuazione delle uscite

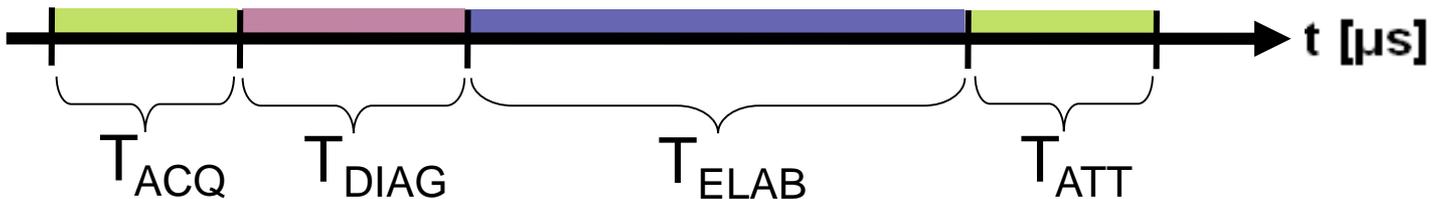


PLC: logica di funzionamento

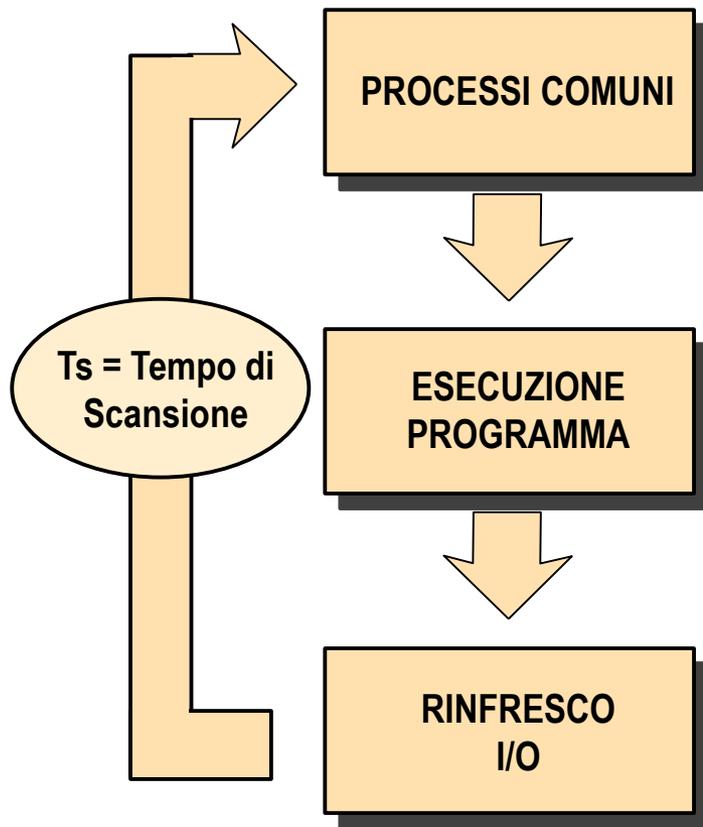
- Questo modo di funzionamento implica l'impossibilità di rilevare (se non con speciali funzioni o moduli) la variazione di un segnale che avviene durante l'elaborazione del programma
- **Nella prima fase** viene acquisita in un preciso istante la condizione degli ingressi (fotografia degli ingressi fisici) e fatta una "copia" nell'area di memoria di I/O
- **Nella seconda fase** vengono elaborati gli ingressi e modificate le uscite (solo all'interno dell'area di memoria del PLC) in modo sequenziale secondo l'esecuzione del programma
- **Nella terza fase** avviene l'attuazione fisica delle uscite secondo lo stato logico determinato durante l'elaborazione del programma
- La prima e la terza fase vengono comunemente indicate come fase di "I/O Refresh" (rinfresco di Ingressi e Uscite)

Il tempo di scansione

- Il tempo di scansione è il tempo che intercorre tra un I/O refresh ed il successivo
- E' composto dalla somma di:
 - tempo di elaborazione
 - tempo di I/O refresh
 - tempo di gestione dei processi comuni



Il tempo di scansione



- Affinché un segnale sia sicuramente rilevabile dagli ingressi del PLC deve avere una durata minima maggiore del tempo di scansione



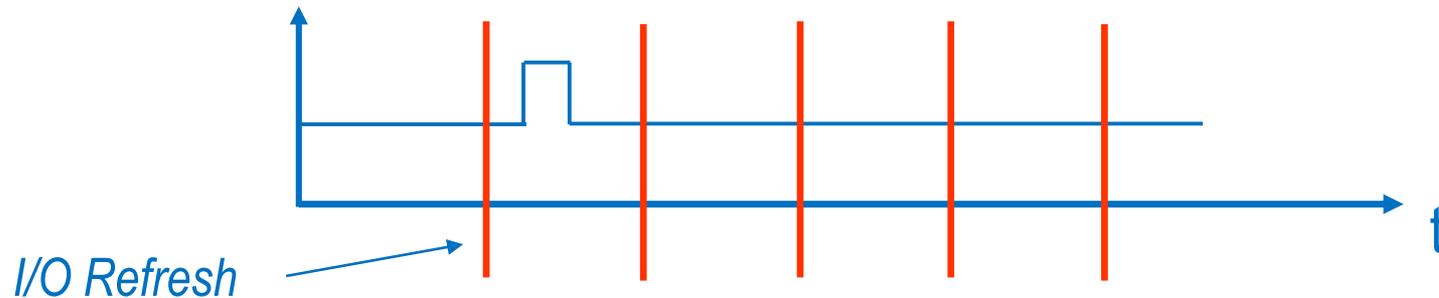
- L'attivazione fisica di una uscita non avviene immediatamente all'esecuzione dell'istruzione, bensì a fine scansione durante il rinfresco degli I/O

Minimo segnale rilevabile

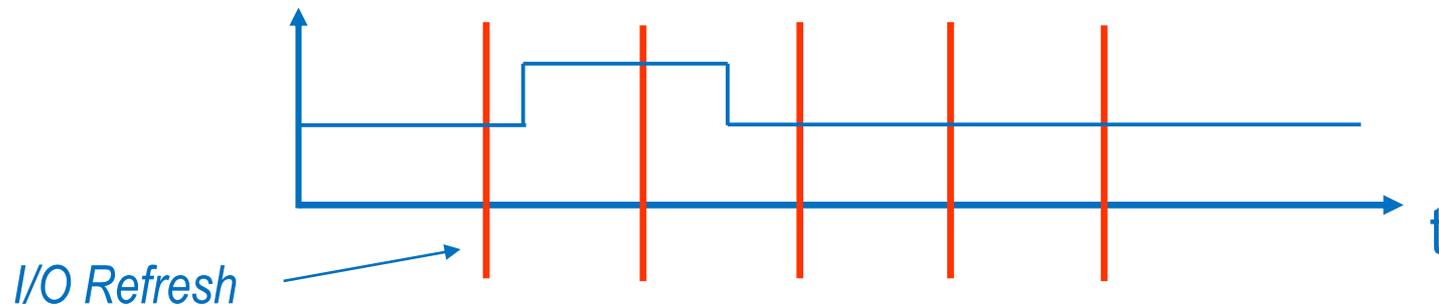
- Il segnale elettrico interpretato dall'interfaccia di I/O del PLC viene letto ad ogni scansione
- Ciò implica che se un segnale ha una variazione più breve del tempo di scansione potrebbe non essere rilevato
- Per avere la certezza che un segnale fisico sia sicuramente rilevato dal PLC è necessario quindi che la sua durata sia superiore al tempo di scansione

Minimo segnale rilevabile

- Segnale di ingresso non sempre rilevabile:

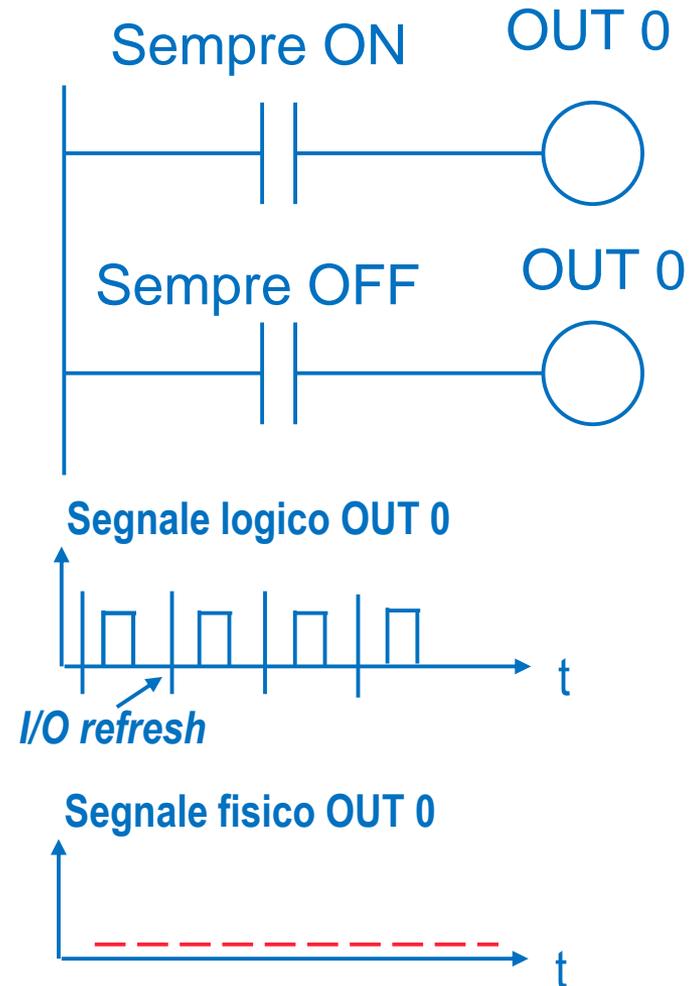


- Segnale di ingresso sicuramente rilevabile:



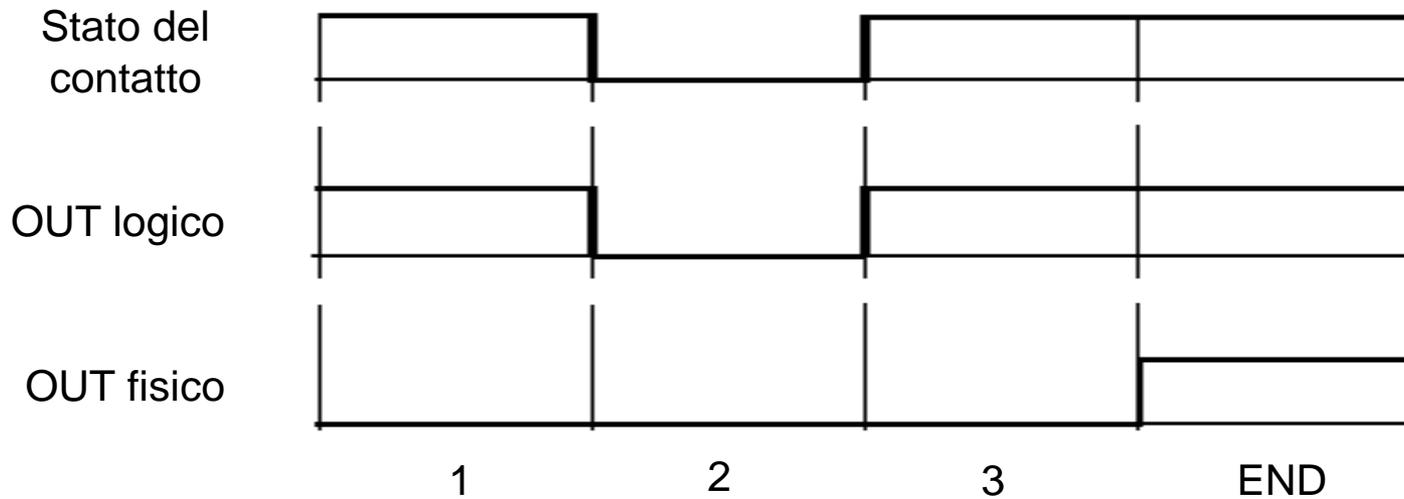
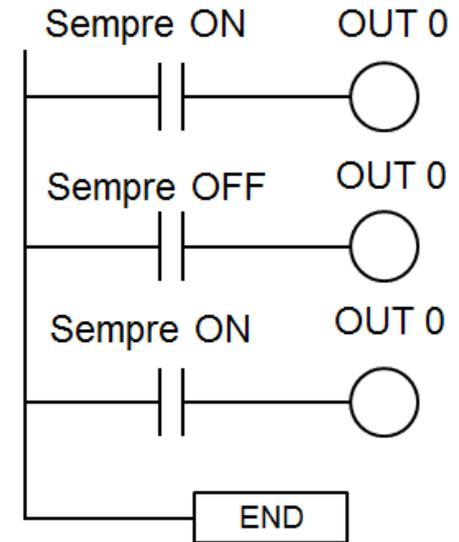
Stato logico e stato fisico

- La modifica dello stato logico di un'uscita durante l'elaborazione del programma non implica la modifica del suo stato fisico
- Lo stato fisico è determinato solo dall'ultima modifica dello stato logico presente nell'Immagine di Processo



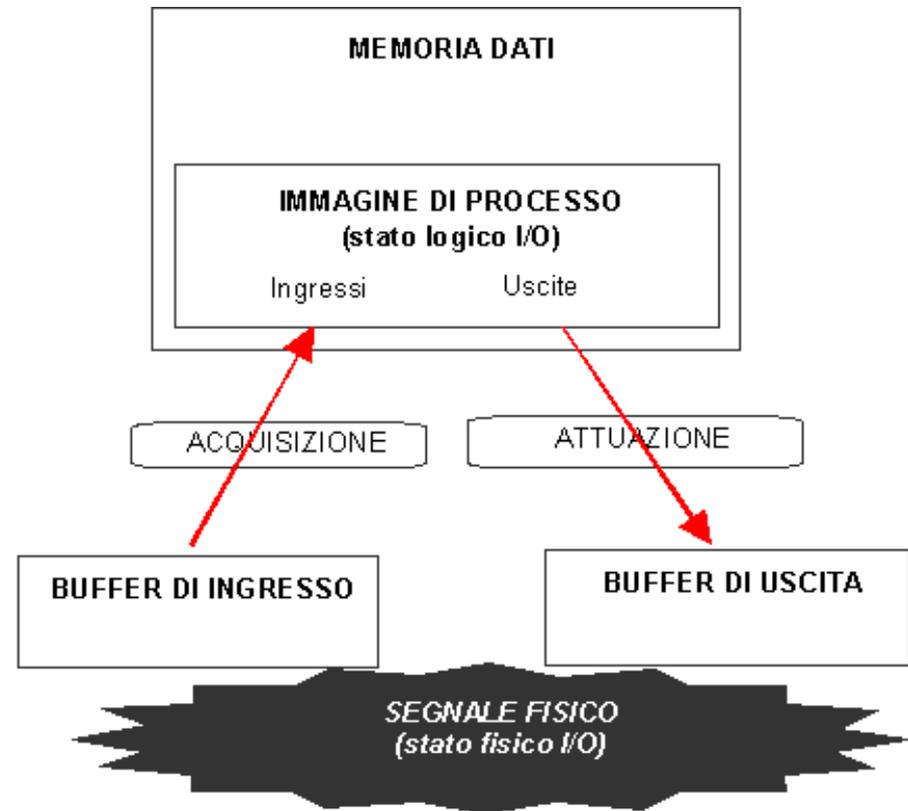
Stato logico e stato fisico

N. rung	Stato del contatto	OUT logico	OUT fisico
1	ON	ON	OFF
2	OFF	OFF	OFF
3	ON	ON	OFF
END		ON	ON



PLC e l'Immagine di Processo

- L'Immagine di Processo contiene lo stato logico degli ingressi relativo all'ultima acquisizione e lo stato logico delle uscite risultante dall'elaborazione del programma
- Le uscite verranno fisicamente aggiornate al corrispondente stato logico solo all'attuazione successiva



La modalità operativa del PLC

- Il PLC ha 3 modalità operative:
 - **Stop/Program**: il programma non viene eseguito; è possibile scrivere le aree di memoria e scaricare il programma e le impostazioni del PLC
 - **Monitor**: il programma viene eseguito in modo ciclico; è possibile scrivere le aree di memoria ma non è possibile scaricare il programma e le impostazioni del PLC (modalità tipica di funzionamento)
 - **Run**: il programma viene eseguito in modo ciclico; non è possibile scrivere le aree di memoria e non è possibile scaricare il programma e le impostazioni del PLC (è una modalità di funzionamento “sicura” poiché non possono essere modificate le aree del PLC dall’esterno)

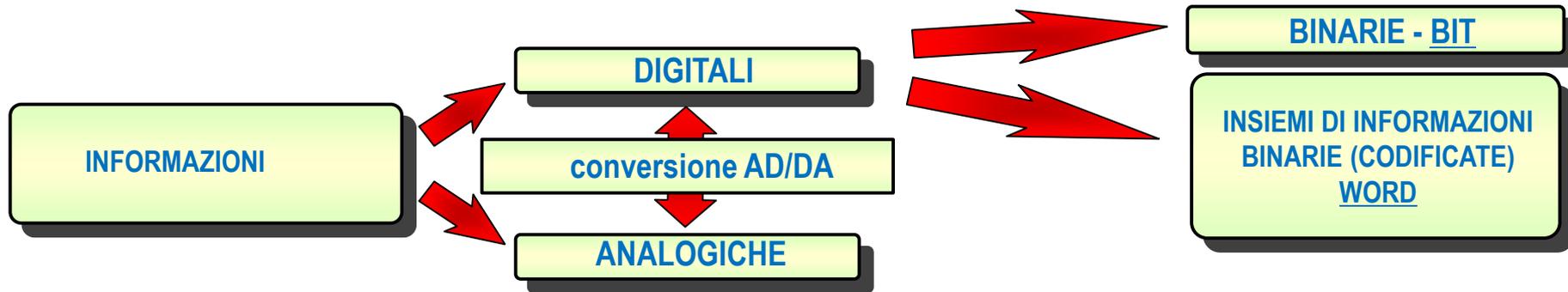
Are di memoria e sistemi di codifica

Rappresentazione dell'informazione
digitale

L'area di memoria del PLC

- L'area di memoria di un PLC è un insieme di celle composte da 16 elementi consecutivi di memoria elementare che possono assumere il valore 0 o 1
- L'elemento di memoria elementare si chiama bit
- Un canale (o word) è una sequenza di 16 bit consecutivi
- Tutte le informazioni vengono memorizzate nel PLC in aree di memoria formate da 16 bit

Le informazioni: BIT... WORD



BIT → 1/0 Unità elementare di memoria

DIGIT → 1/0 1/0 1/0 1/0 4 bit

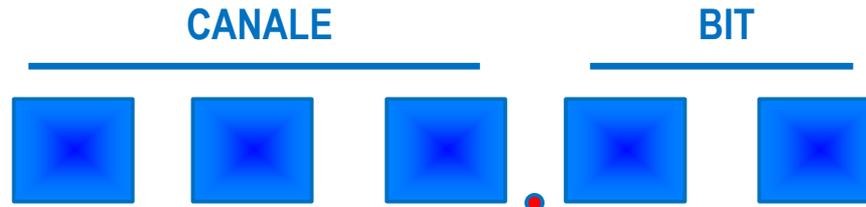
BYTE → 1/0 1/0 1/0 1/0 1/0 1/0 1/0 1/0 8 bit

WORD → 1/0 1/0 1/0 1/0 1/0 1/0 1/0 1/0 1/0 1/0 1/0 1/0 1/0 1/0 1/0 1/0 16 bit

1 byte = 2 digit
1 word = 4 digit

Indirizzamento delle aree di memoria

- L'area di memoria è una serie di celle consecutive, ognuna della quale è composta da 16 bit
- Gli indirizzamenti sulla memoria dei PLC possono essere fatti a canale o a bit



Esempi di indirizzi

- Esempi di indirizzamento a word (canale):

D00100	Canale 100 dell'area D
W056	Canale 56 dell'area W
1234	Canale 1234 dell'area CIO

- Esempi di indirizzamento a bit:

0022.13	Bit 13 del canale 22 dell'area CIO
W056.00	Bit 0 del canale 56 dell'area W
H234.11	Bit 11 del canale 234 dell'area H

Il valore di un numero e le sue rappresentazioni

- Un numero è composto da una serie di cifre che rappresenta un valore
- Nella codifica decimale esistono 10 cifre che rappresentano un valore in funzione della loro posizione
- Ad esempio il valore del numero 1234 si calcola come:
$$1 \times 10^3 + 2 \times 10^2 + 3 \times 10^1 + 4 \times 10^0 = 1234$$
- Il fattore moltiplicativo si compone di una base (nel sistema decimale 10) e di un esponente che varia da 0 a n a seconda della posizione della cifra
- Le cifre più a destra sono quindi cifre meno significative

La codifica in base n

- Per convertire un numero di n cifre in una qualunque base B in un valore decimale si deve operare come segue
- $$N_{\text{dec}} = B^{n-1} \times 1^{\text{cifra}} + B^{n-2} \times 2^{\text{cifra}} + \dots + B^0 \times n^{\text{cifra}}$$
- Le cifre a sinistra sono indicate come le cifre “più significative”

Codifica BCD

- In passato il modo più naturale di rappresentare un numero decimale in una word di dimensione 16 bit era utilizzando la codifica BCD.
- Ogni singola cifra decimale (da 0 a 9) era rappresentata con 4 bit (1 digit) in codifica binaria. Con una word era quindi possibile rappresentare 4 cifre, corrispondenti a un intervallo da 0 a 9999.
- Il peso di ogni cifra è dato dalla posizione del singolo digit come avviene nella codifica decimale

Esempio:

Rappresentazione BCD del numero decimale 4897

0 1 0 0	1 0 0 0	1 0 0 1	0 1 1 1
---------	---------	---------	---------

Esempi di codifica BCD

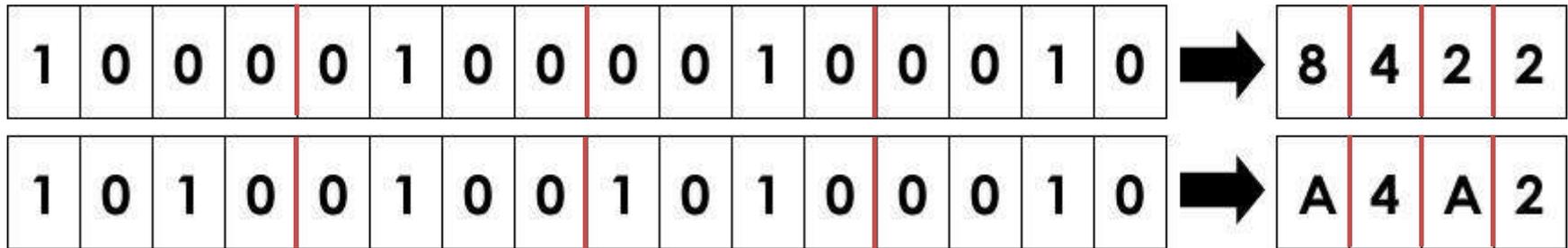
1234	→	0001 "1"	0010 "2"	0011 "3"	0100 "4"
5678	→	0101 "5"	0110 "6"	0111 "7"	1000 "8"
3790	→	0011 "3"	0111 "7"	1001 "9"	0000 "0"
2A39	→	ERRORE			

La codifica esadecimale (binaria)

- L'utilizzo di 4 bit per implementare le cifre da 0 a 9 rappresenta però un sotto utilizzo della macchina.
- Sfruttando tutte le combinazioni di 4 bit i valori rappresentabili sono infatti 16 e non 10.
- La codifica esadecimale aggiunge le cifre A, B, C, D, E, ed F alle cifre da 0 a 9.
- E' possibile rappresentare in una word un valore esadecimale da 0 a FFFF, che corrisponde ad un valore decimale da 0 a 65535.
- La codifica nativa del PLC è l'esadecimale.

BCD e Esadecimale

- Rappresentiamo due esempi: codifica in BCD (prima word) e esadecimale (seconda word)



- Il valor massimo raggiungibile dalla codifica BCD è 9999
- Il valore massimo raggiungibile dalla codifica esadecimale è FFFF (che in valore decimale senza segno equivale a 65535)

Esempi di codifica dei numeri

Base 10

4	3	2	1
---	---	---	---

- Codifica decimale: cifre da 0 a 9
 $4321 = 10^3 \times 4 + 10^2 \times 3 + 10^1 \times 2 + 10^0 \times 1$
 $=$
 $= 4.321$ unità

Base 2

1	0	1	1
---	---	---	---

- Codifica binaria: cifre 0 e 1
 $1011 = 2^3 \times 1 + 2^2 \times 0 + 2^1 \times 1 + 2^0 \times 1 =$
 $= 11$ unità

Base 16

1	A	3	2
---	---	---	---

- Codifica esadecimale: cifre da 0 a 9 + A...F
 $1A32 = 16^3 \times 1 + 16^2 \times 10 + 16^1 \times 3 + 16^0 \times$
 $2 = 6.706$ unità

Tabella di codifica

DECIMALE	ESADECIMALE	BINARIO	BCD
0	0	0000	0000 0000
1	1	0001	0000 0001
2	2	0010	0000 0010
3	3	0011	0000 0011
4	4	0100	0000 0100
5	5	0101	0000 0101
6	6	0110	0000 0110
7	7	0111	0000 0111
8	8	1000	0000 1000
9	9	1001	0000 1001
10	A	1010	0001 0000
11	B	1011	0001 0001
12	C	1100	0001 0010
13	D	1101	0001 0011
14	E	1110	0001 0100
15	F	1111	0001 0101

I PLC compatti

Gamma, prestazioni, espandibilità,
aree di memoria e allocazione I/O

La gamma

- CPM1A
- CPM2A
- CPM2C



**Prodotto obsoleto → Modello CPM1A
Modello CPM2A**

**Sostituzione raccomandata → Modello CP1E-N
Modello CP1L**

- CP1E
- CP1L
- CP1H



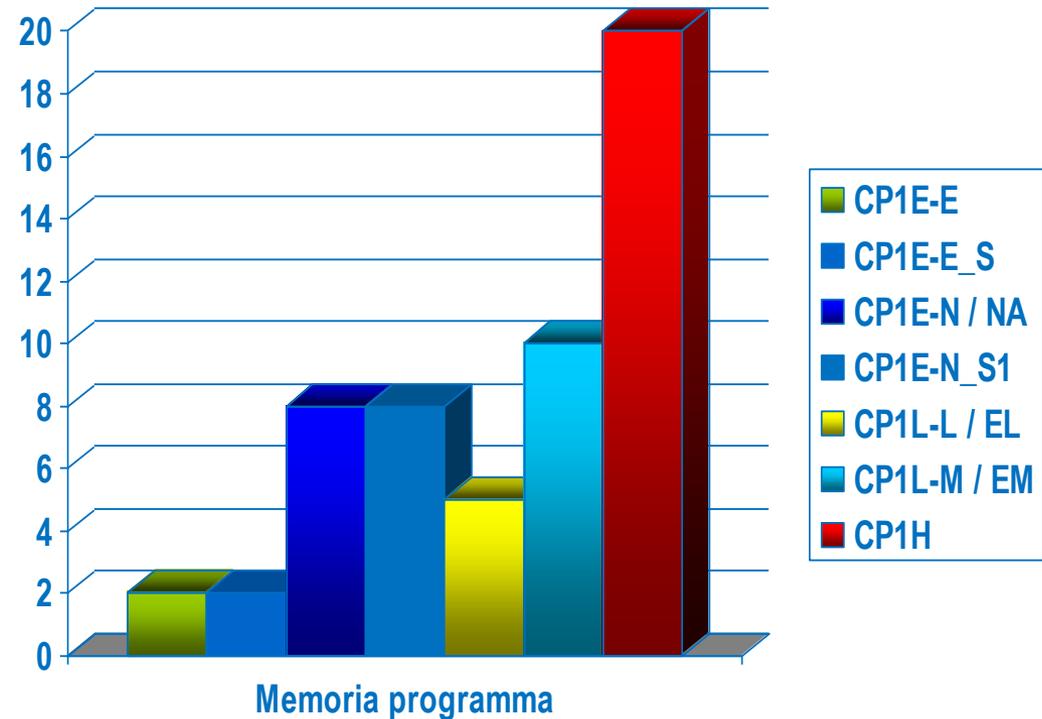
Classificazione CPU – PLC compatti (famiglia CP1)

Le caratteristiche principali che permettono di classificare una CPU sono le seguenti:

- Memoria programma
- Memoria dati
- Numero di I/O massimi gestibili
- Numero di moduli di espansione collegabili
- Tempo di esecuzione delle istruzioni
- Linguaggi di programmazione supportati

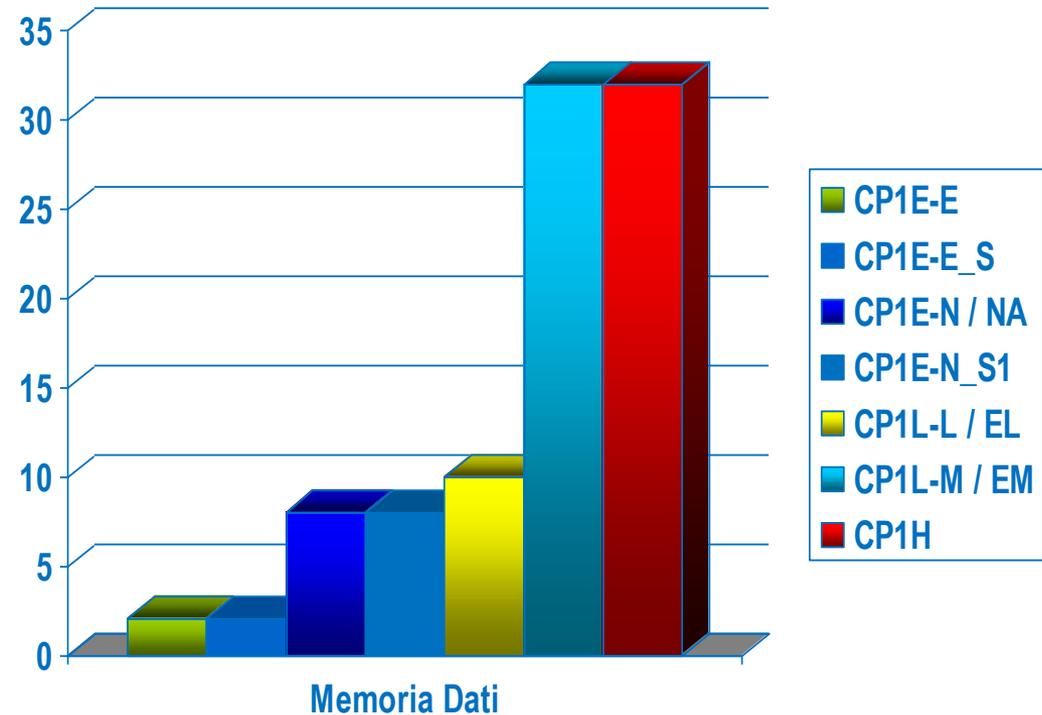
Classificazione CPU – Memoria Programma

<i>PLC</i>	<i>Memoria Programma</i>
CP1E-E CP1E-E_S	2 Kstep
CP1E-N / NA CP1E-N_S1	8 Kstep
CP1L-L / EL	5 Kstep
CP1L-M / EM	10 Kstep
CP1H	20 Kstep



Classificazione CPU – Memoria Dati

PLC	Memoria Dati (DM)
CP1E-E CP1E-E_S	2 Kword
CP1E-N / NA CP1E-N_S1	8 Kword
CP1L-L / EL	10 Kword
CP1L-M / EM	32 Kword
CP1H	32 Kword



Classificazione CPU

- Per tempo di esecuzione delle istruzioni:
 - CP1L:
 - Istruzione LD: 0,61 μ s
 - Istruzione MOV: 4,1 μ s
 - CP1H:
 - Istruzione LD: 0,10 μ s
 - Istruzione MOV: 0,15 μ s
 - CP1E:
 - Istruzione LD: 1,19 μ s
 - Istruzione MOV: 8 μ s

Linguaggi supportati

- CP1L e CP1H:
 - Ladder
 - Lista istruzioni (mnemonico)
 - Testo strutturato (solo per i Function Block)
- CP1E:
 - Ladder
- I FB non possono essere utilizzati nelle CPU CP1E

Espandibilità e configurazione

- La configurazione è limitata all'aggiunta di moduli di espansione (numero variabile in base alla CPU)
- Esistono moduli di espansione:
 - Ingressi/uscite relè/transistor
 - Ingressi/uscite analogici
 - Ingressi per termocoppie o PT100
 - Moduli slave Compobus S, Device Net, Profibus DP

Espandibilità e configurazione

- **Per numero di I/O:**

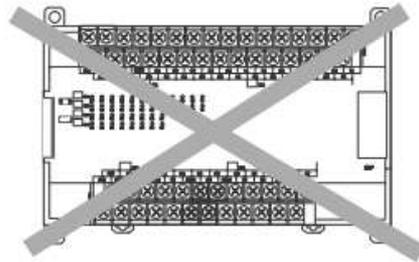
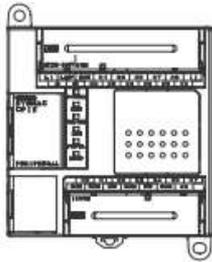
– CP1E-E10, CP1L-L10:	10 punti di I/O max
– CP1E-E14(S), -N14:	14 punti di I/O max
– CP1E-E20(S), -N20:	20 punti di I/O max
– CP1L-L14:	54 punti di I/O max (14 + 40 x 1)
– CP1L-L20, -EL20:	60 punti di I/O max (20 + 40 x 1)
– CP1E-NA20:	140 punti di I/O max (20 + 40 x 3)
– CP1E-E30(S), -N30(S1), CP1L-M30, -EM30:	150 punti di I/O max (30 + 40 x 3)
– CP1E-E40(S), -N40(S1), CP1L-M40, -EM40:	160 punti di I/O max (40 + 40 x 3)
– CP1E-E60S, -N60(S1), CP1L-M60:	180 punti di I/O max (60 + 40 x 3)
– CP1H:	320 punti di I/O max (40 + 40 x 7)

- **Per sistemi di espansione collegabili:**

– CP1E-E10/14/20, -E14/20S, -N14/20	NON espandibile
– CP1L-L10	NON espandibile
– CP1L-L14/20, -EL	max 1 modulo CP1W/CPM1A
– CP1E-E30/40, -E30/40/60S	max 3 moduli CP1W/CPM1A
– CP1E-N30/40/60, -N30/40/60S1, -NA	max 3 moduli CP1W/CPM1A
– CP1L-M, -EM	max 3 moduli CP1W/CPM1A
– CP1H	max 7 moduli CP1W/CPM1A

Configurazione per CP1E

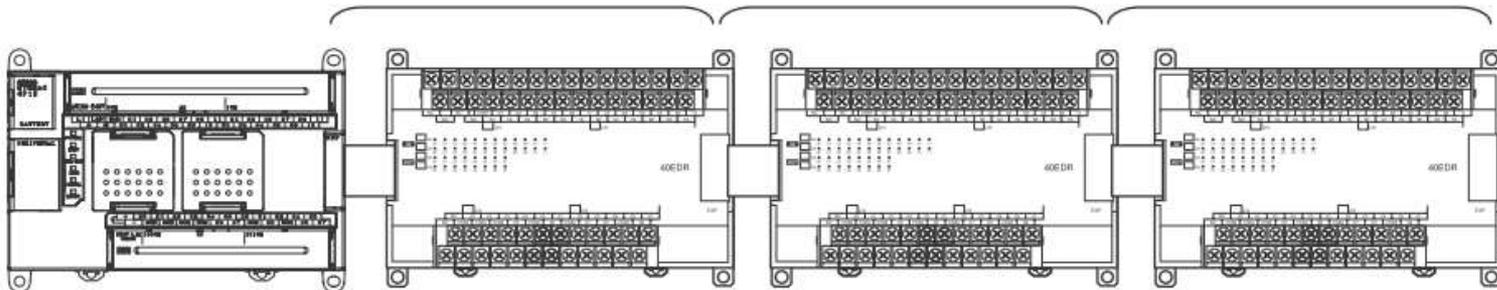
- CP1E E10/14/20(S) or N14/20 CPU Unit



CP-series Expansion Units and Expansion I/O Units cannot be connected.

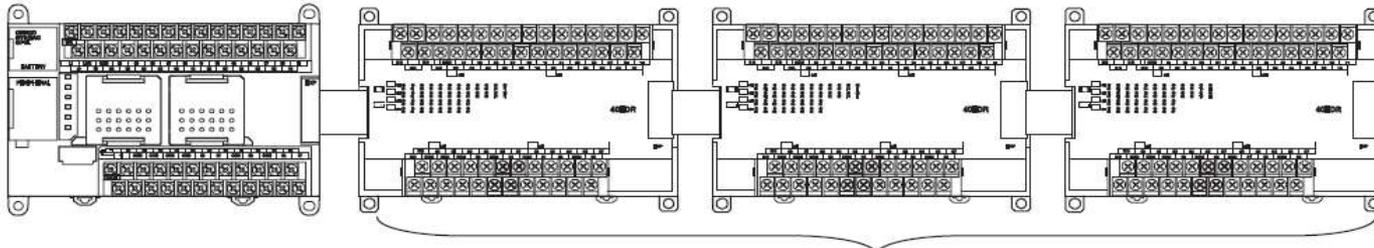
- CP1E E30/40/60(S), N30/40/60(S□) or NA20 CPU Unit

A total of up to three CP-series Expansion I/O Units and Expansion Units can be connected.



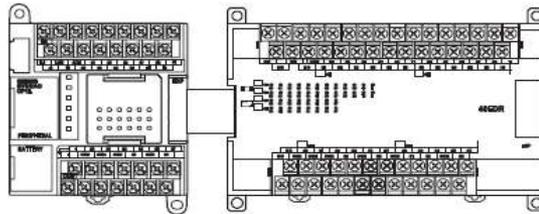
Configurazione per CP1L

CP1L CPU Unit with 30, 40 or 60 I/O Points



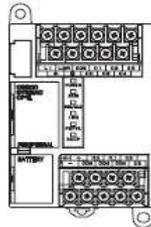
A maximum of three CP-series Expansion I/O Units or Expansion Units can be added.

CP1L CPU Unit with 20 or 14 I/O Points



One CP-series Expansion I/O Unit or Expansion Unit can be added.

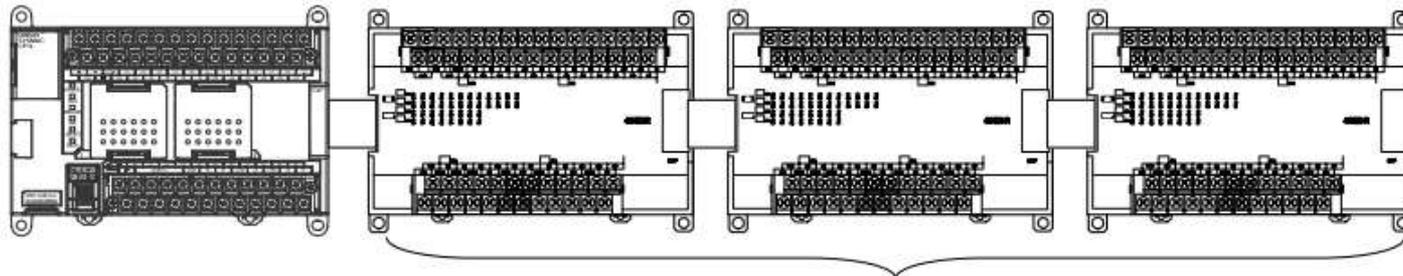
CP1L CPU Unit with 10 I/O Points



No CP-series Expansion I/O Unit or Expansion Unit can be added.

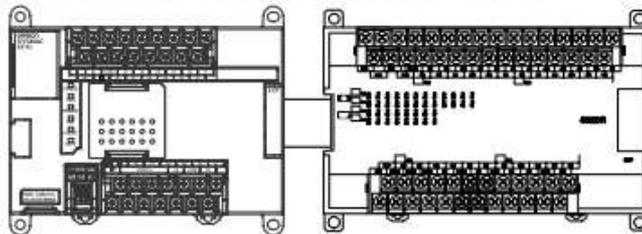
Configurazione per CP1L-E

CP1L-EM CPU Unit with 30 or 40 I/O Points



A maximum of three CP-series Expansion I/O Units or Expansion Units can be added.

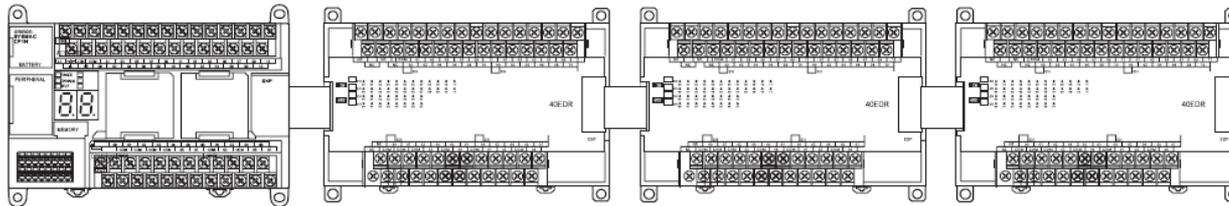
CP1L-EL CPU Unit with 20 I/O Points



One CP-series Expansion I/O Unit or Expansion Unit can be added.

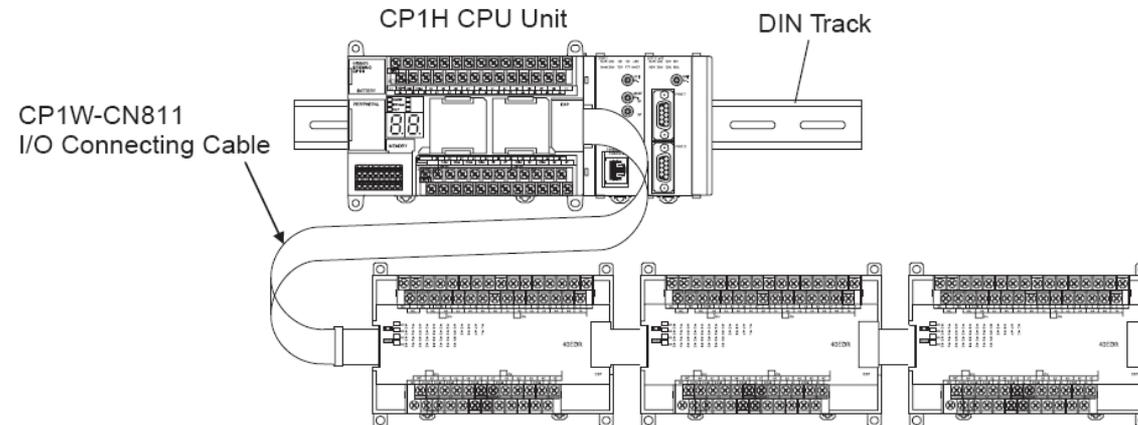
Configurazione per CP1H

CP1H CPU Unit



A maximum of 7 CPM1A-series Expansion I/O Units or Expansion Units can be added.

When CP1W-CN811 I/O Connecting Cable is used, the cable length can be extended by up to 80 cm, enabling installing the Units in two rows.



Schede opzionali

- Per tutte le CPU da 10 punti di I/O, per le CP1E-E_S e le CP1E-N_S1 non è possibile installare nessuna scheda opzionale
- I modelli da 14 o 20 punti di I/O sono disponibili con un solo slot dedicato per l'installazione di una scheda opzionale
- I restanti modelli di CPU da 30 punti di I/O in su, hanno due slot opzionali



Schede opzionali – Gamma

- CP1W-CIF01: porta RS-232C. Max distanza di trasmissione 15 m
- CP1W-CIF11: porta RS-422A/485. Max dist. di trasmissione 50 m
- CP1W-CIF12: porta RS-422A/485 (isolato). Max dist. di trasmissione 500 m
- CP1W-CIF41: porta Ethernet 
- CP1W-DAM01: display LCD 4 righe, 12 caratteri visualizzati 
- CP1W-ADB21: 2 ingressi analogici, 0-10 V, 0-20 mA 
- CP1W-DAB21V: 2 uscite analogiche, 0-10 V 
- CP1W-MAB221: 2 ingressi analogici 0-10 V, 0-20 mA e 2 uscite 0-10 V 



Schede opzionali – Modelli supportati

Codice scheda	CP1E					CP1L					CP1H
	-E(S)	-N14/20	-N30/40/60	-N(S1)	-NA20	-L10	-L14/20	-M30/40/60	-EL20	-EM30/40	
Porta RS232 int.	○	●	●	●	●	○	○	○	○	○	○
Porta RS485 int.	○	○	○	●	○	○	○	○	○	○	○
CP1W-CIF01	○	○	●	○	●	○	●	●●	●	●●	●●
CP1W-CIF11	○	○	●	○	●	○	●	●●	●	●●	●●
CP1W-CIF12	○	○	●	○	●	○	●	●●	●	●●	●●
CP1W-CIF41 (v. 2)	○	○	●	○	●	○	●	●●	○	○	●●
CP1W-DAM01	○	○	○	○	○	○	●	●	●	●	●
CP1W-ADB21	○	○	● *	○	● *	○	○	○	●	●●	○
CP1W-DAB21V	○	○	● *	○	● *	○	○	○	●	●●	○
CP1W-MAB221	○	○	● *	○	● *	○	○	○	●	●●	○

○ = Non supportata

● = Supportata (max 1 collegabile)

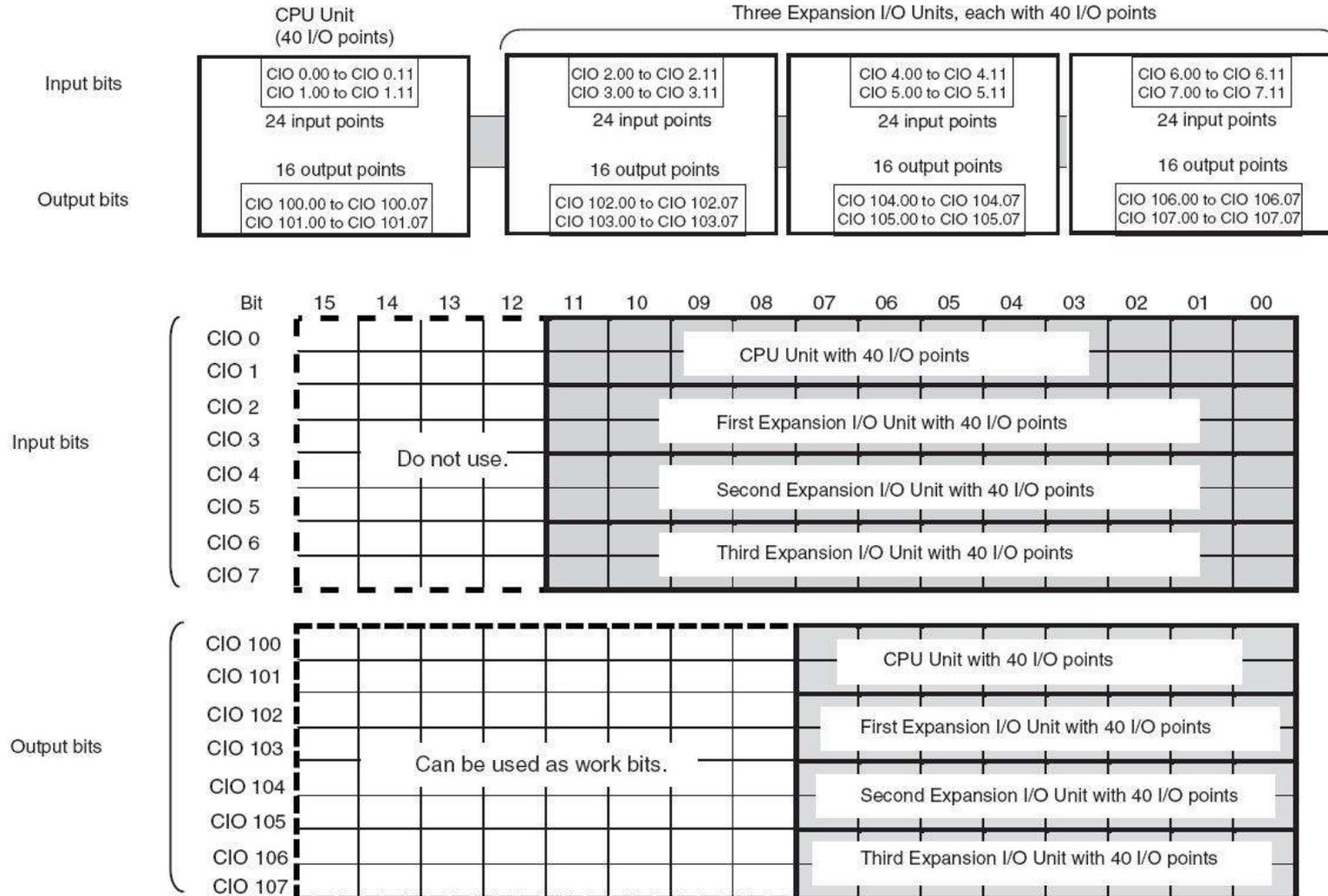
●● = Supportata (max 2 collegabili)

* A partire dalla versione firmware 1.2

Allocazione I/O per tutte le CPU CP1

- Il canale di partenza per gli ingressi è CIO 0
- Il canale di partenza per le uscite è CIO 100
- Gli ingressi sono utilizzati a gruppi di 12 bit
- Le uscite sono utilizzate a gruppi di 8 bit
- I canali vengono riempiti in maniera sequenziale

Allocazione I/O per tutte le CPU CP1



Adattatore per moduli speciali

- Solo per il CP1H: è possibile utilizzare al massimo due moduli CJ (speciali e/o bus) da allocare nel seguente modo:
 - CPU Bus: 1500 + 25 x n (da CIO 1500 a 1899) [n da 0...9 e A..F]
 - I/O Speciali: 2000 + 10 x n (da CIO 2000 a 2959) [n da 0...9 e A..F]

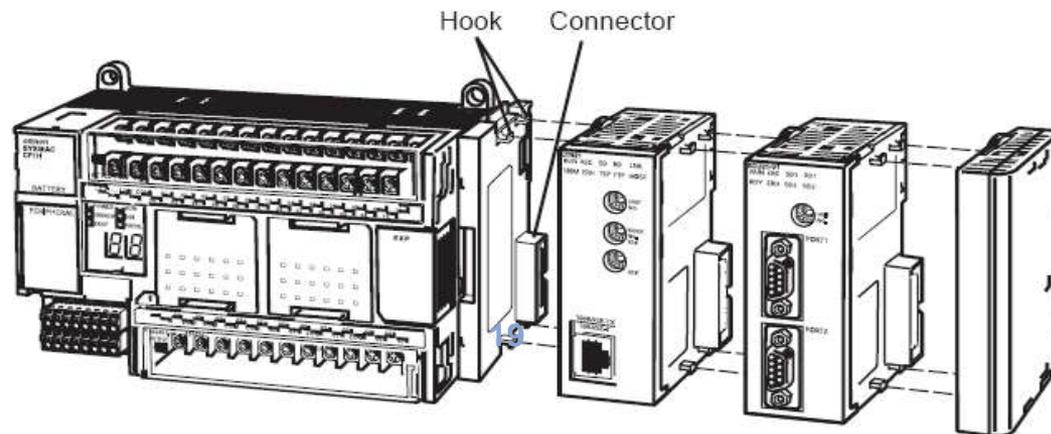
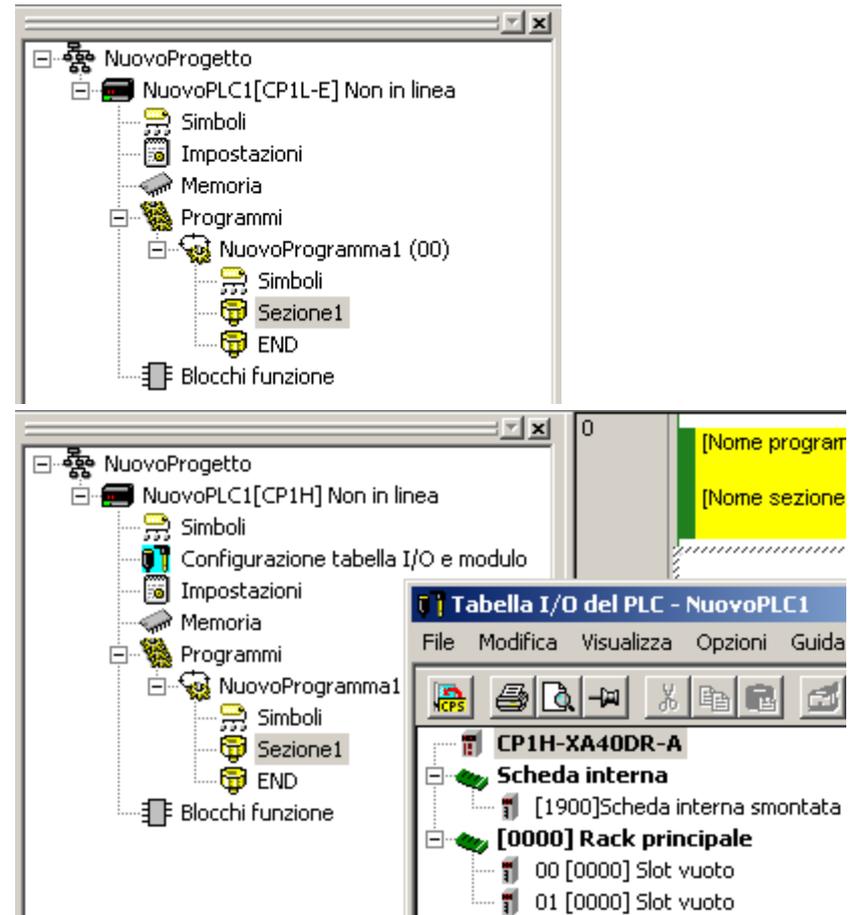


Tabella I/O

- La tabella I/O per i CP1E e CP1L non esiste poiché la configurazione degli I/O è *fissa*
- E' presente invece nel CP1H poiché occorre definire le impostazioni degli altri 2 slot aggiuntivi CJ1 (con adattatore)



Aree di memoria

- Le aree di memoria sono suddivise in:
 - RITENTIVE: sono quelle aree che permettono di mantenere i dati salvati anche dopo lo spegnimento dell'unità o al cambio della modalità operativa
 - NON RITENTIVE: sono le aree che si azzerano dopo lo spegnimento dell'unità o al cambio della modalità operativa

Aree di memoria

- Le principali aree di memoria sono:
 - Area CIO (Core I/O area): nessun prefisso. Area di memoria di I/O, moduli speciali, schede di comunicazione. Area non ritentiva, viene azzerata ad ogni avvio del PLC. L'accesso può avvenire a Bit o a Canale
 - Area HR (Holding Area): prefisso H. Area ritentiva gestibile a Bit e a Canale. Memorizzazione permanente di dati di programma
 - Area W (Work Area): prefisso W; per l'utilizzo come registri temporanei o Bit di appoggio; viene usata tipicamente per l'allocazione automatica. Accesso a Bit e a canale. Non ritentiva

Aree di memoria

- Area A (Auxiliary Area): prefisso A. Contiene informazioni per la diagnostica, lo stato e la configurazione del PLC
- Area Temporizzatori (Timer Area): prefisso T. Viene simulato il funzionamento di temporizzatori
- Area Contatori (Counter Area): prefisso C. Viene simulato il funzionamento di contatori
- Area DM (Data Memory): prefisso D. Area di memoria ritentiva per la memorizzazione dei dati. Viene utilizzata anche per la memorizzazione della configurazione dei moduli speciali, indirizzabile unicamente a canale.

Gestione memoria

Area code	CP1L		CP1H	CP1E	
CIO	IN: 0 to 99 OUT: 100 to 199		IN: 0 to 16 OUT: 100 to 116	IN: 0 to 99 OUT: 100 to 199	
W	W000 to W511			W0 to W99	
H	H000 to H511			H0 to H49	
A	Read Only: A000 to A477 Read-write: A448 to A959			Read Only: A0 to A447 Read-Write: A448 to A753	
D	<i>CPU 10, 14, 20 I/O</i>	<i>CPU 30, 40, 60 I/O</i>	D00000 to D19999: free	CP1E-E	CP1E-N
	D0 to D9999 D32000 to D32767	D0 to D32767	D20000 to D29599: special I/O unit DM area	D0 to D2047	D0 to D8191
			D29600 to D29999: free		
			D30000 to D31599: CPU BUS unit DM area		
		D31600 to D32767: free			
T	T0 to T4095			T0 to T255	
C	C0 to C4095			C0 to C255	

Introduzione alla programmazione

Linguaggi di programmazione

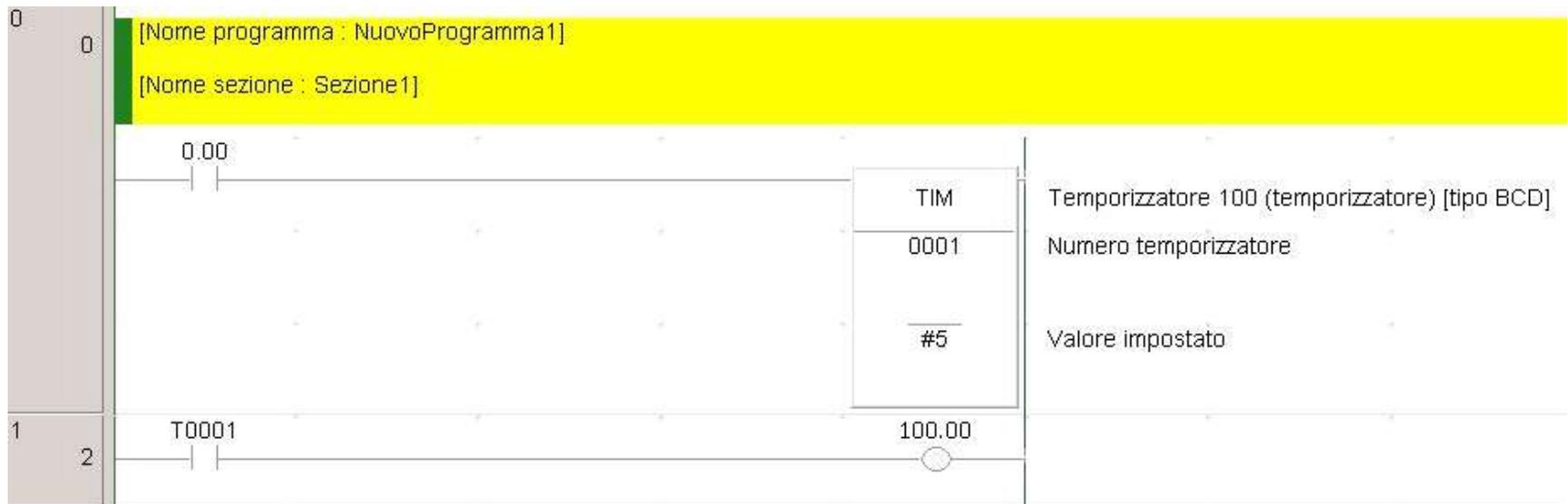
- I principali linguaggi di programmazione sono:
 - Lista istruzioni (mnemonico)
 - Ladder
 - Testo strutturato
 - SFC

Lista istruzioni (mnemonico)

- Il linguaggio in lista istruzioni è un linguaggio mnemonico molto compatto e consente di programmare il PLC con una vera e propria sequenza di istruzioni
- Le istruzioni sono codificate nel seguente formato:
 - Indirizzo della memoria in cui l'istruzione è memorizzata
 - Istruzione stessa
 - Il codice di identificazione dell'elemento funzionale
 - Il valore da settare (temporizzatori, contatori, ...)
- Esempio: LD 0.00

Ladder

- Il Ladder è un linguaggio di programmazione grafico dove troviamo contatti, bobine e funzioni
- E' il linguaggio di programmazione più utilizzato per la sua semplicità d'interpretazione ed è supportato da tutti i PLC Omron.



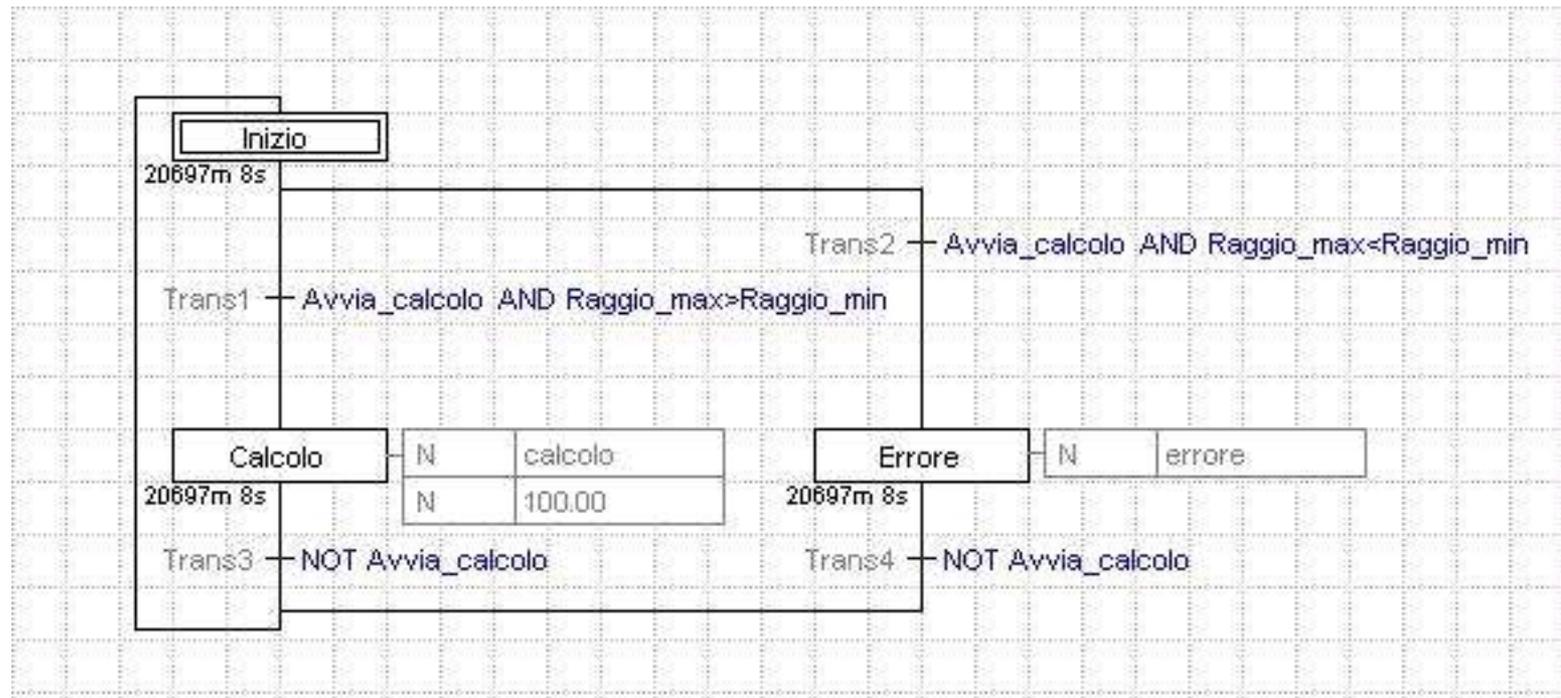
Testo strutturato

- Il testo strutturato è un linguaggio di programmazione simile al Pascal
- E' particolarmente indicato per l'elaborazioni di algoritmi e calcoli

```
IF raggio_max > raggio_min THEN
    cost:=4000,0/(raggio_max-raggio_min);
    raggio:=INT_TO_REAL(v_in)/cost+raggio_min;
    circonferenza:=raggio*2,0*3,14;
ELSE
    ENO:=false;
    circonferenza:=0,0;
END_IF;
```

SFC

- E' un linguaggio di programmazione grafico
- L'SFC è utilizzato soprattutto per il controllo di sequenze



CX-ONE

- CX-ONE è un collezione di software per la programmazione e la configurazione dei dispositivi Omron
- CX-ONE comprende i seguenti software principali
 - CX-Programmer Programmazione PLC
 - CX-Integrator Configuratore DeviceNet, integrazione dispositivi
 - CX-Designer Programmazione terminali NS
 - CX-Drive Configurazione Inverter e Servo Drive
 - CX-Motion Programmazione schede motion control PLC
 - CX-Motion-MCH Programmazione motion control Mechatrolink
 - CX-Position Configurazione schede controllo assi PLC
 - CX-Process Programmazione unità PLC per il controllo di processo
 - CX-ConfiguratorFDT Configurazione rete Profibus DP
 - CX-Protocol
 seriale su PLC Programmazione protocolli di comunicazione
 - CX-Simulator Software per la simulazione dei PLC CJ1 e CS1
 - CX-Thermo Configurazione termoregolatori
 - CX-Server Driver di comunicazione di CX-ONE

Requisiti di sistema

- Configurazione minima richiesta per il corretto funzionamento di CX-One e CX-Programmer:
 - Hard Disk con almeno 560 MB di spazio libero (4 GB per l'installazione completa)
 - Minimo 256 MB di RAM (consigliato > 512 MB)
 - XVGA 1024x768 o superiore
 - Windows 2000, XP, Vista, 7 e 8 (Sistemi a 32 e 64 bit)



CX-Programmer

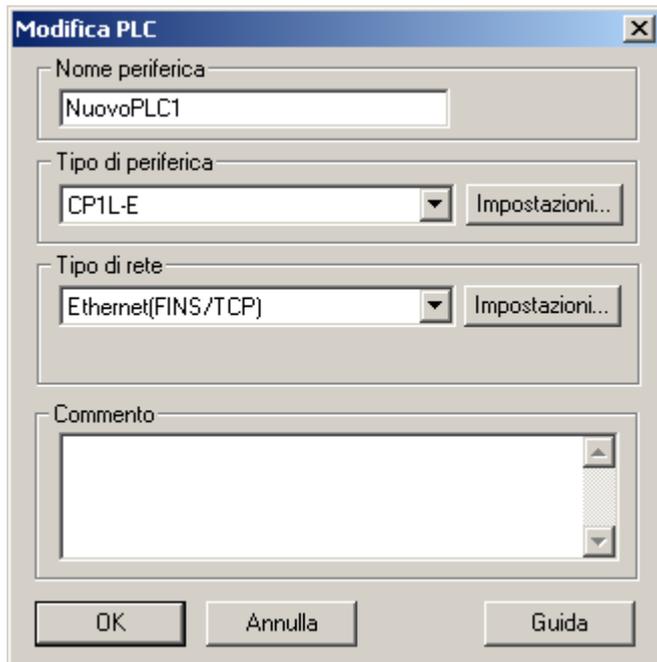
- CX-Programmer è un software per ambiente Windows, creato per la programmazione dei PLC Omron per mezzo di un Personal Computer
- CX-Programmer consente di:
 - Creare e modificare programmi
 - Archiviare i programmi su disco
 - Verificare la correttezza del programma tramite operazioni di debug
 - Verificare la funzionalità del programma mediante connessione seriale con il PLC (On-line) e monitoraggio dei dati
 - Commentare il programma in tutte le sue parti
 - Stampare i programmi e la documentazione di riferimento

CX-Programmer: Installazione

- Il programma viene fornito esclusivamente all'interno di CX-One
- Per effettuare l'installazione inserire il DVD e avviare da Gestione Risorse il programma SETUP presente sul DVD
- L'installazione è quasi completamente automatica: seguire le istruzioni indicate a video

Creazione di un progetto

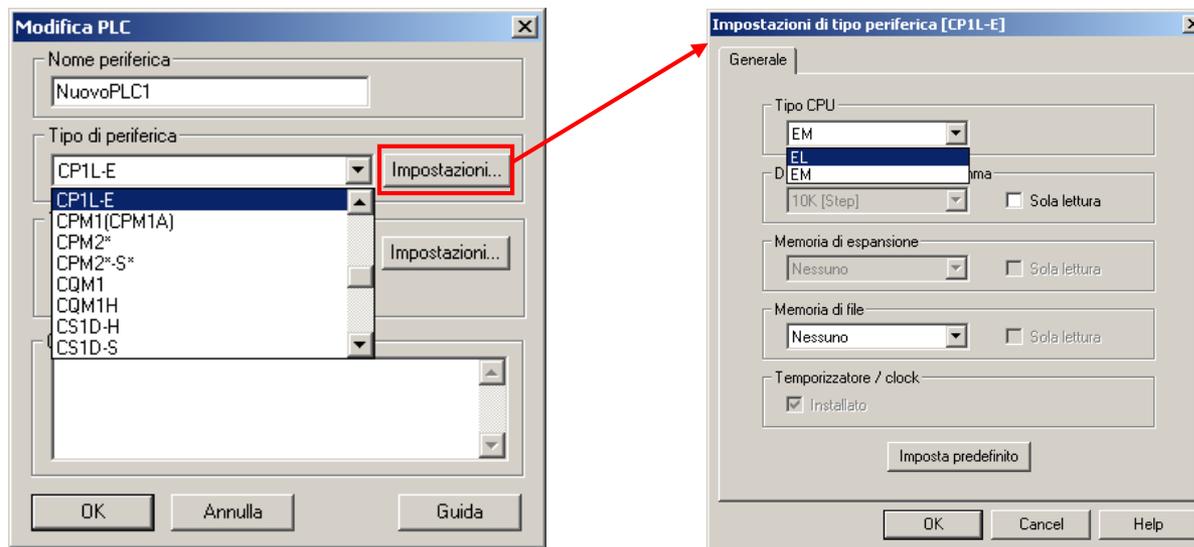
- Dopo aver lanciato CX-Programmer dal menù di avvio di Windows, selezionare “File” → “Nuovo”



- Tipo di periferica: permette di selezionare il tipo di PLC; il pulsante impostazioni viene utilizzato per selezionare il tipo di CPU
- Tipo di rete: permette di selezionare il tipo di collegamento (Ethernet FINS/TCP, Sysmac Way, Toolbus ed altri tipi di rete per i PLC che le supportano).
Il pulsante Impostazioni visualizza la finestra per la configurazione dei parametri di comunicazione.

Selezione del PLC

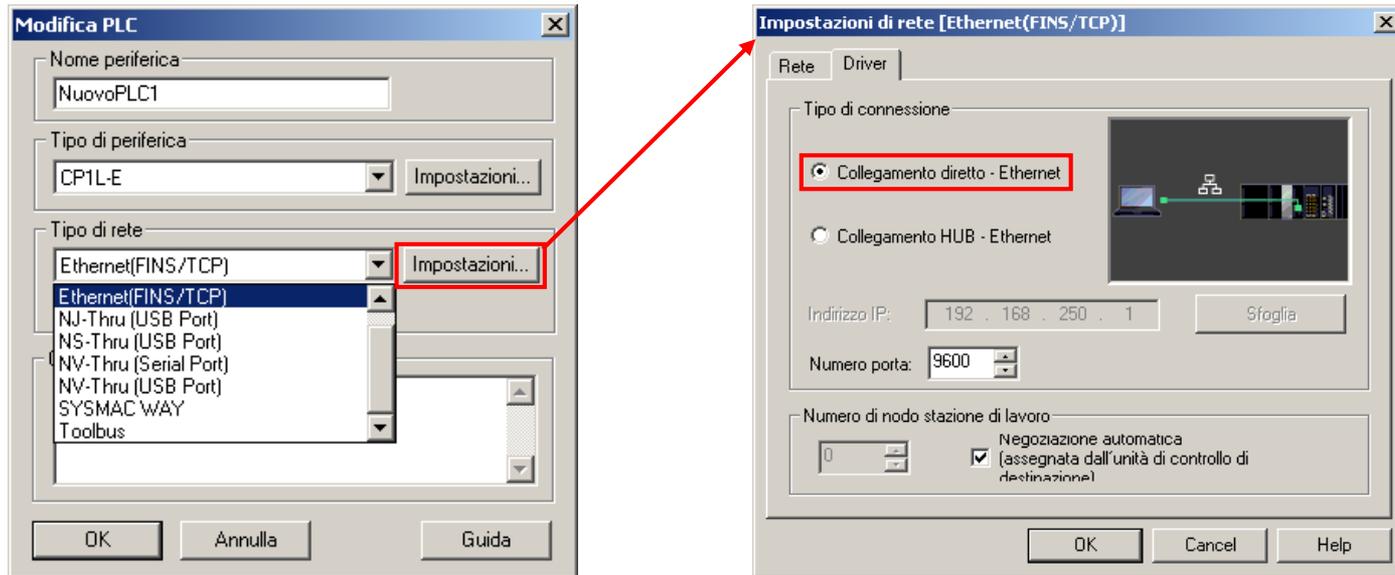
- Selezionare il tipo di PLC dall'elenco a discesa e cliccare sull'opzione 'Impostazioni' per inserire il tipo esatto di CPU



- Se le impostazioni di CX-Programmer non corrispondono esattamente al PLC collegato, non sarà possibile comunicare con il PLC

Impostazioni della comunicazione – metodo Ethernet

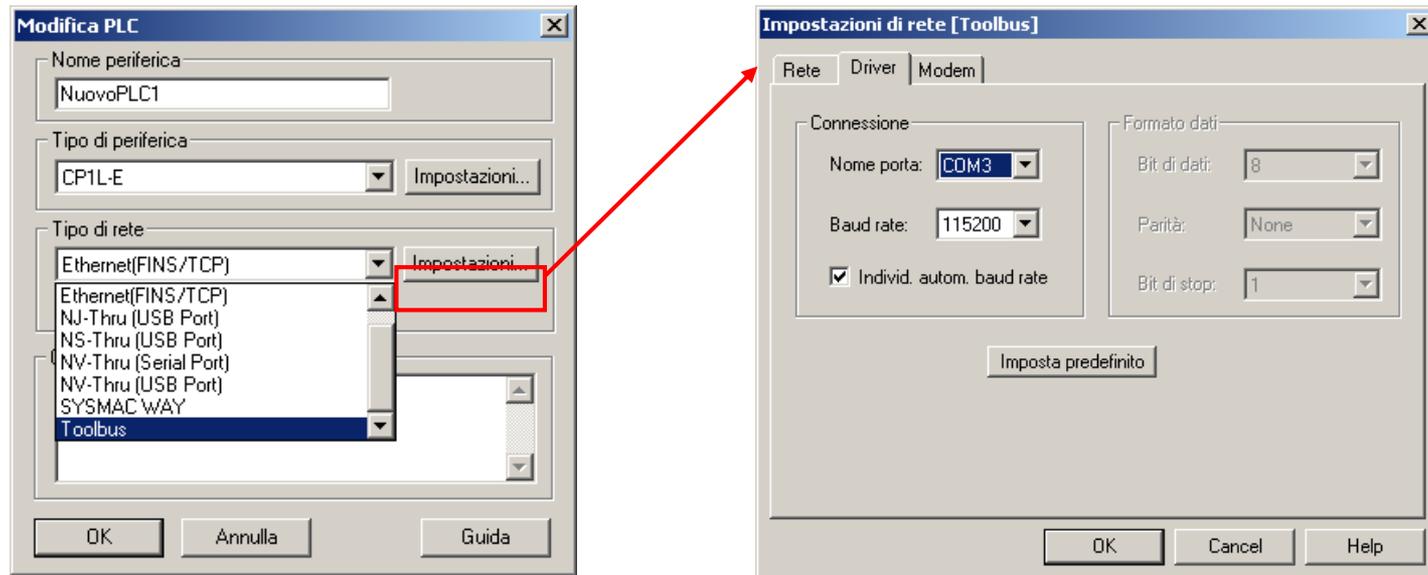
- Selezionare il protocollo Ethernet(FINS/TCP), quindi cliccare su ‘Impostazioni’ per determinare se il tipo di connessione tra PC e PLC avviene direttamente o tramite un HUB



- NB: verificare che non sia disabilitata la connessione Ethernet diretta nel software ‘Communications Middleware Utilities’

Impostazioni della comunicazione – metodo seriale

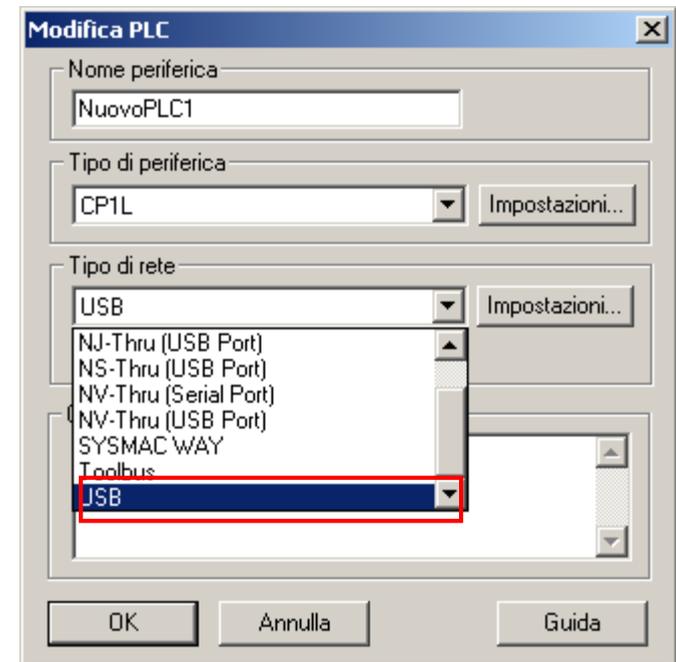
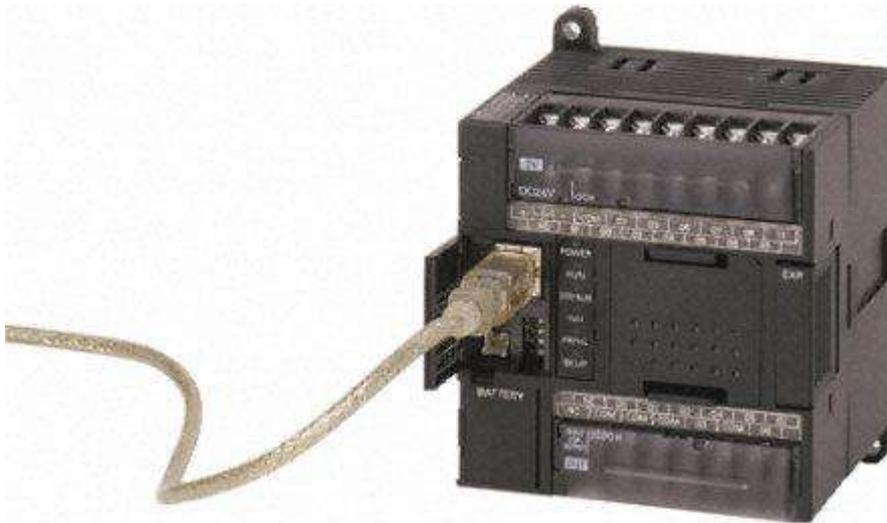
- Effettuare la selezione del protocollo (Toolbus, Sysmac Way,...) cliccando su ‘Impostazioni’ per specificare il numero della porta, la velocità e il formato dei dati



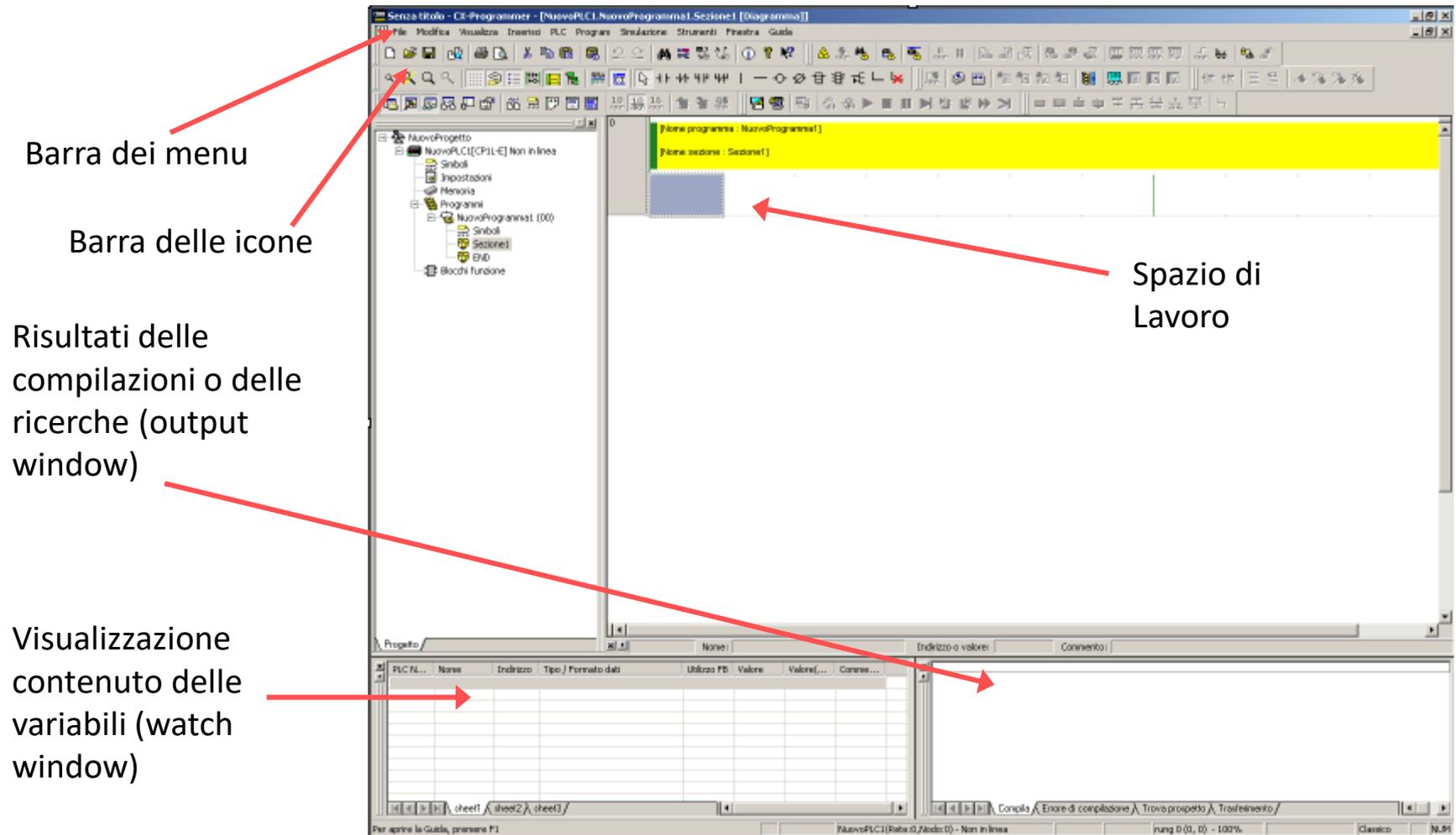
- Riassumendo quindi, è possibile collegarsi al PLC tramite porta Ethernet, porta seriale oppure via Modem

Impostazioni della comunicazione – metodo USB

- In alcuni modelli di PLC (CJ2, CP1E, CP1L-L, CP1L-M e CP1H) è anche possibile collegarsi al software CX-Programmer tramite protocollo USB



La finestra principale

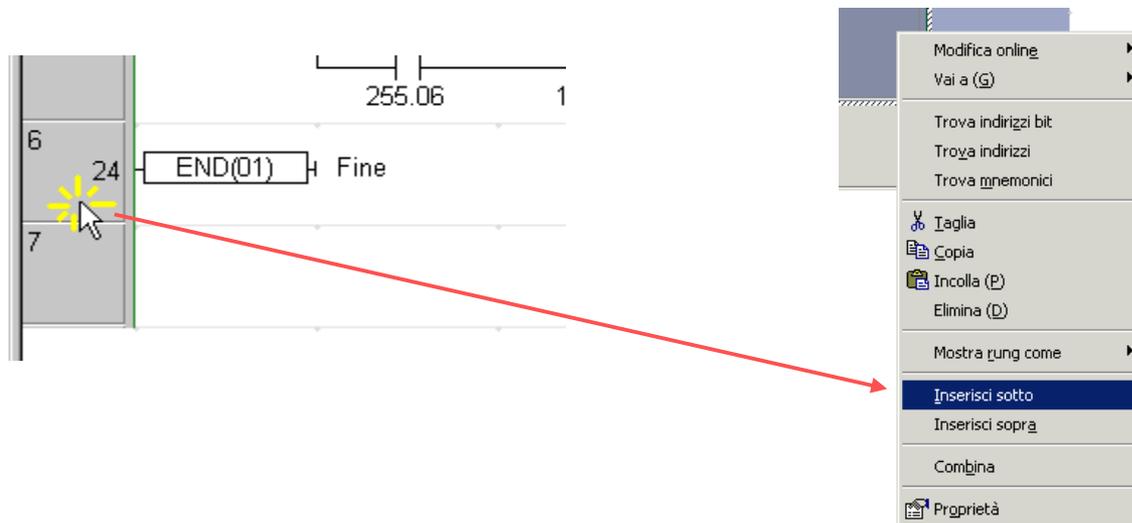


Le Rung o 'Network'

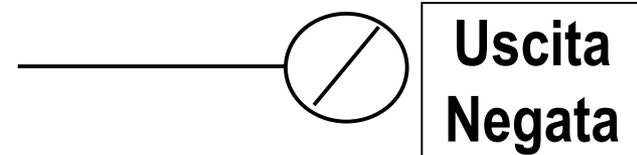
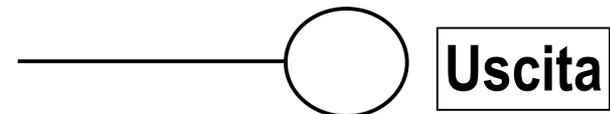
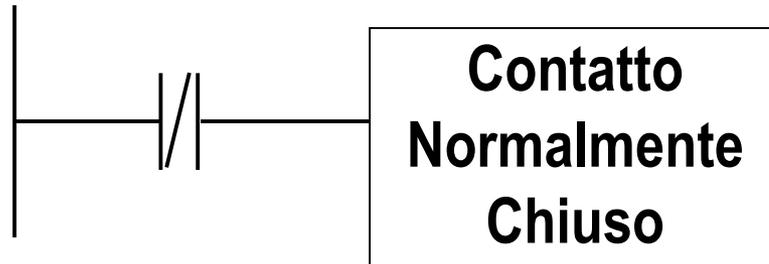
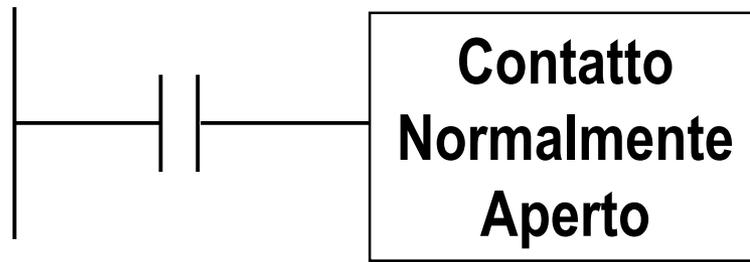
- Durante la programmazione, per agevolare l'editazione, CX-Programmer crea automaticamente una Rung vuota
- Lo schema mnemonico viene tradotto automaticamente, e ne viene verificata la correttezza
- Le Rung errate o incomplete vengono indicate con una linea verticale rossa sulla barra delle network
- E' anche possibile lasciare network incomplete, lasciare contatti privi di indirizzo e/o nome, lasciare le istruzioni prive di operandi
- Tutto può essere completato in un secondo momento...

Le Rung o 'Network'

- Quando si apre un nuovo progetto il cursore è già posizionato sulla prima Rung
- Per inserire una nuova Rung tra due già esistenti è sufficiente selezionare con il tasto destro del mouse il bordo sinistro del ladder e scegliere se inserire la nuova Rung al di sotto o al di sopra di quella correntemente selezionata

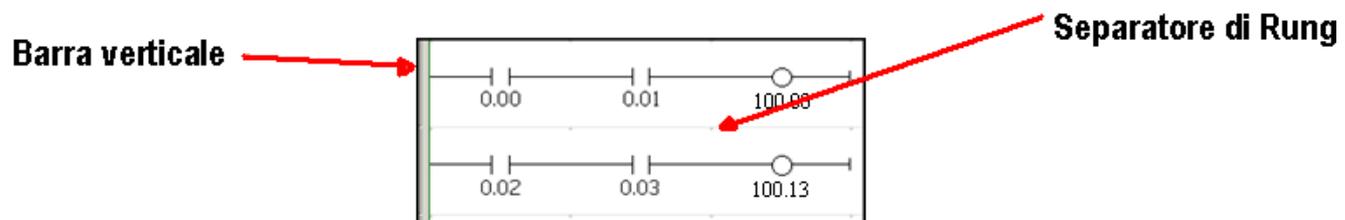


Ladder diagram



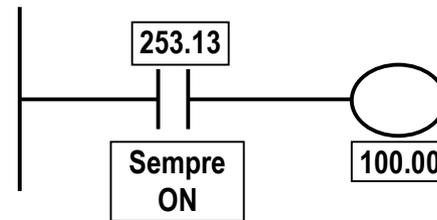
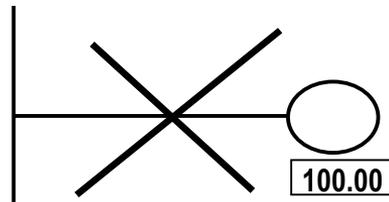
Concetti base

- Un programma in diagramma a relè (Ladder Diagram), è formato da una serie di righe circuitali
- Una riga circuitale (Network o Rung) è composta da:
 - una serie di contatti collegati tra loro in serie e/o parallelo
 - eventuali diramazioni
 - bobine o funzioni
- Ogni Rung (o Network) ha origine da una barra verticale posta a sinistra del diagramma
- Il flusso dei contatti viene letto da sinistra a destra e dall'alto verso il basso



Concetti base

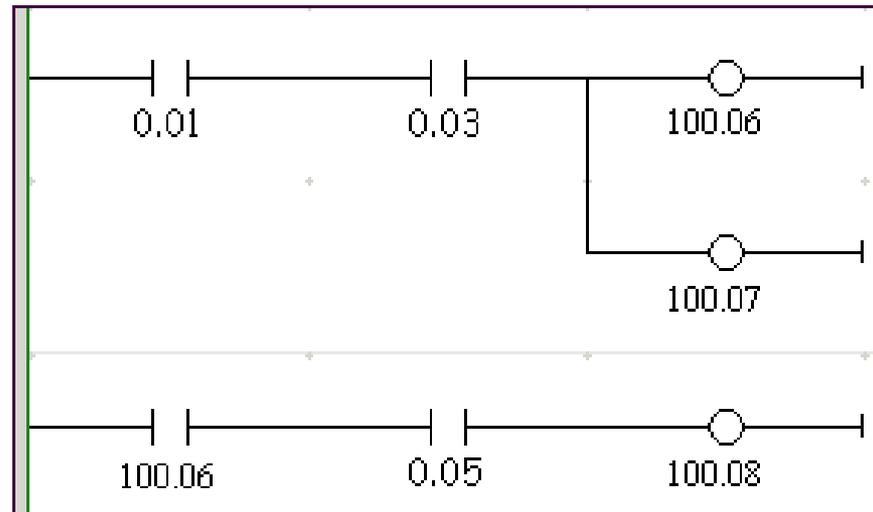
- Una bobina non può essere collegata direttamente alla barra di sinistra; in tal caso è necessario interporre un contatto 'Sempre ON' (sempre chiuso)



- A destra di una bobina di uscita o di una funzione non è possibile programmare alcun contatto
- Il numero di contatti collegabili in serie o parallelo è limitato solo dalla memoria di programma del PLC

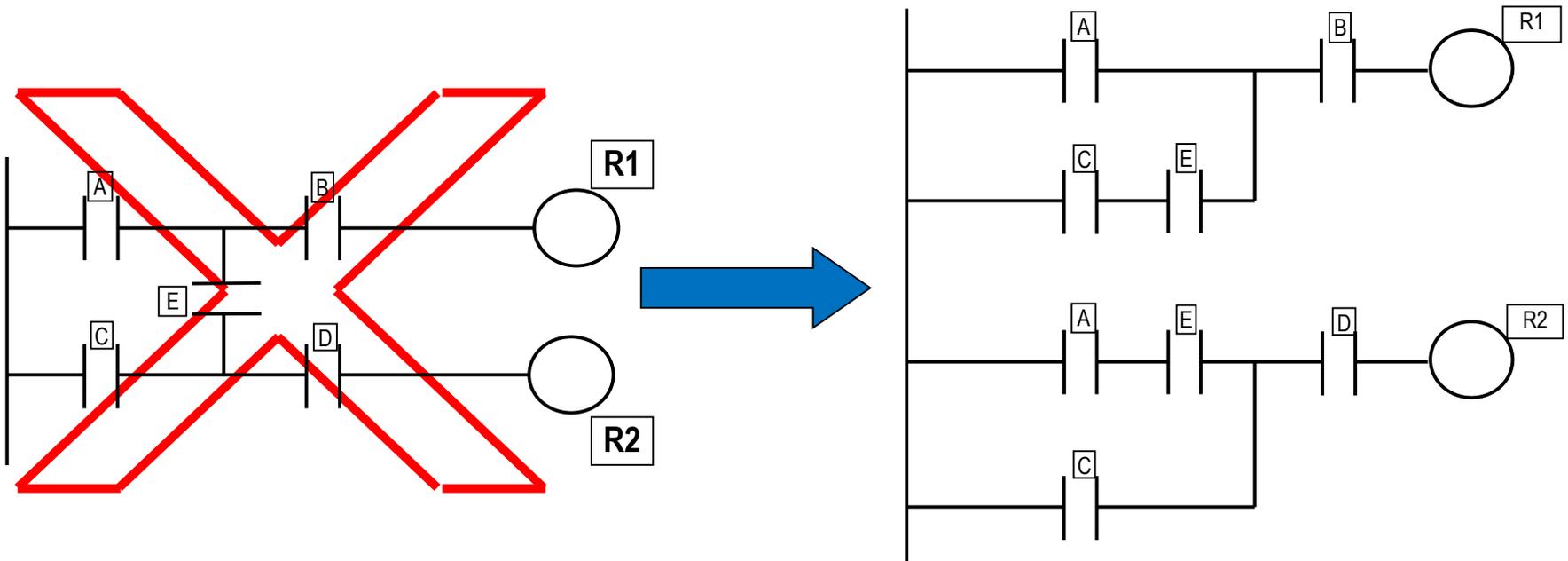
Concetti base

- E' consigliabile non programmare un contatto di uscita come bobina più di una volta
- E' possibile utilizzare come input ausiliario un contatto associato allo stato dell'uscita
- E' possibile collegare in parallelo due o più bobine



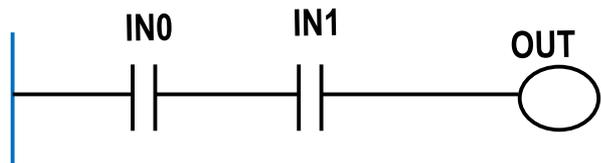
Concetti base

- Non è possibile programmare dei contatti sui rami di collegamento verticale

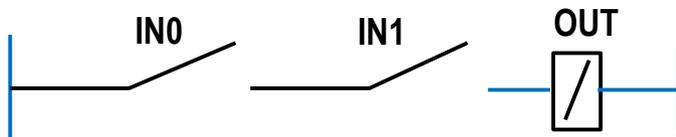


L'istruzione AND

DIAGRAMMA A RELE'

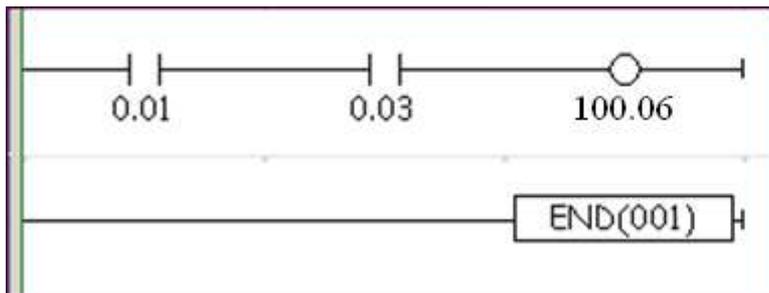


SCHEMA ELETTRICO



IN0	IN1	OUT
0	0	0
0	1	0
1	0	0
1	1	1

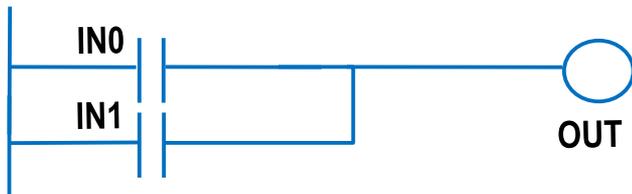
- Esempio



Rung	Passo	Istruzione	Operando
0	0	LD	0.01
	1	AND	0.03
	2	OUT	100.06
1	3	END(001)	

L'istruzione OR

DIAGRAMMA A RELE'

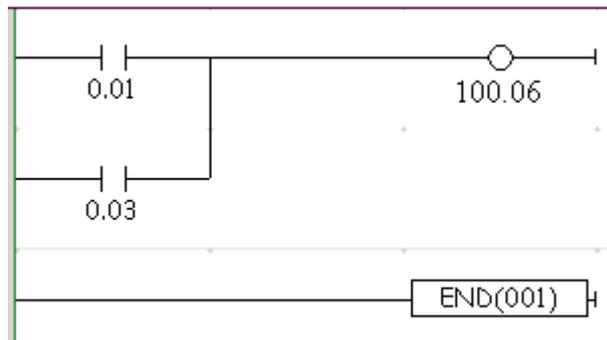


SCHEMA ELETTRICO



IN0	IN1	OUT
0	0	0
0	1	1
1	0	1
1	1	1

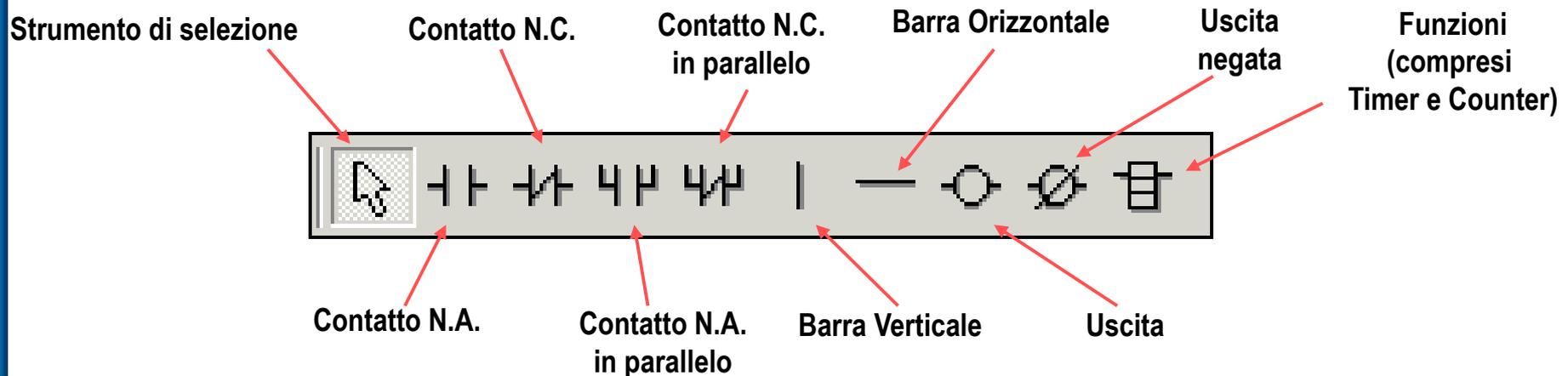
- Esempio



Runq	Passo	Istruzione	Operando
0	0	LD	0.01
	1	OR	0.03
	2	OUT	100.06
1	3	END(001)	

Simboli del programma Ladder

- Per disegnare il diagramma a relè, contatto per contatto, è sufficiente:
 - Posizionare il cursore
 - Fare clic sull'icona relativa al simbolo ladder desiderato
 - Inserire i parametri relativi e premere Invio



Inserimento di contatti e bobine

- Nella finestra di inserimento di un nuovo contatto è possibile specificare nome, indirizzo o valore, commento e tipo di simbolo (locale o globale) associato al contatto selezionato

- Per i PLC serie CP1, CS1 e CJ è anche possibile selezionare la modalità di differenziazione e il refresh immediato

Inserimento di commenti e simboli

- Per aggiungere un simbolo è necessario essere Off-Line

|- Nuovo contatto

Modifica OK Annulla

Informazioni sui simboli

Nome:

Indiriz./valore:

Commento:

Globale Colleg. al file CX-Server

Differenziazione

Nessuno Su Giù Aggiorn. immediato

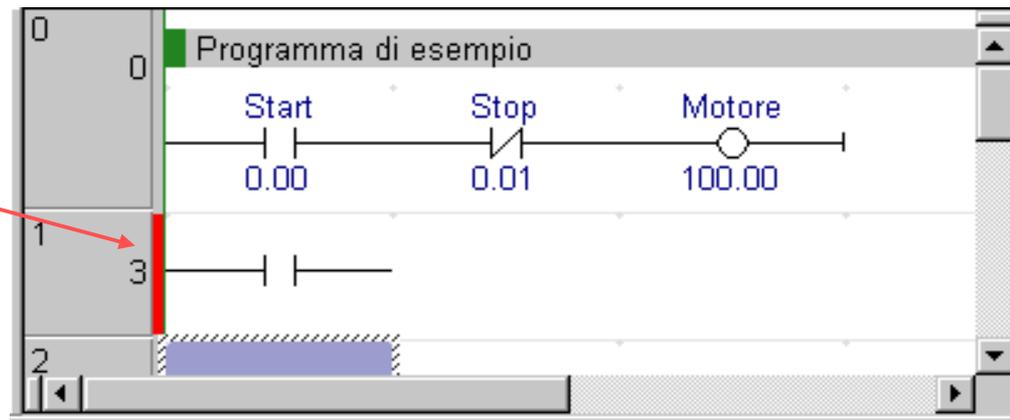
Si inserisce il nome, l'indirizzo o il valore, il commento ed infine il tipo di dato.

Il tipo di dato permette di associare ad una variabile un tipo, rendendo così possibile un controllo sugli operandi delle istruzioni (es.: su un dato definito BCD sarà possibile effettuare solo operazioni BCD, in caso contrario durante la compilazione verrà generato un messaggio di Warning)

Inserimento del programma

- Inserimento degli altri contatti:
 - Per aggiungere un nuovo contatto usare l'icona  oppure il tasto 'C'
 - Per aggiungere un uscita utilizzare l'icona  oppure il tasto 'O'
 - Per collegare due circuiti usare l'icona  oppure il tasto 'V' (per collegare dall'alto in basso) o il tasto 'U' (per collegare dal basso verso l'alto)

La zona rossa indica che la network è incompleta



Editor dei simboli

- Esiste un editor per i simboli globali e un editor per i simboli locali di ciascun singolo task
- Nell'editor dei simboli globali sono già presenti i principali flag del PLC

Nome della variabile

Tipo della variabile

Indirizzo variabile o contenuto della costante

Utilizzo (in/out/work)

Name	Type	Address / Value	Rack Location	Usage	Comment
· P_NE	BOOL	CF001		Work	Not Equals (NE) Flag
· P_OF	BOOL	CF009		Work	Overflow (OF) Flag
· P_Off	BOOL	CF114		Work	Always OFF Flag
· P_On	BOOL	CF113		Work	Always ON Flag
· P_Output_Off_Bit	BOOL	A500.15		Work	Output OFF Bit
· P_Step	BOOL	A200.12		Work	Step Flag
· P_UF	BOOL	CF010		Work	Underflow (UF) Flag
· START	BOOL	0.01	Main Rack : Slot 01	In	Contatto globale di avvio

Posizione nel rack (Bit I/O)

Commento

Tipi di dati

Tipo	Dimensione	Segno	Formato	Range	Note
BOOL	1 Bit	-	Binary	0/1	Indirizzo di un Bit di ingresso o uscita, stati possibili 0 OFF e 1 ON.
INT	1 word	Sì	Binary	-32768 / +32767	Indirizzo di una word binaria con segno.
DINT	2 word	Sì	Binary	-2147483648 +2147483647	Indirizzo di una doppia word binaria con segno.
LINT	4 word	Sì	Binary	-9,223372036855x10 ¹⁸ +9,223372036855x10 ¹⁸	Indirizzo di un intero con segno quadruplo (4 word binarie).
UINT	1 word	No	Binary	0 / 65535	Indirizzo di una word binaria senza segno.
UDINT	2 word	No	Binary	0 / 429467295	Indirizzo di una doppia word binaria senza segno.
ULINT	4 word	No	Binary	0 / 1,844674407371x 10 ¹⁹	Indirizzo di un intero senza segno quadruplo (4 word binarie).
UINT_BCD	1 word	No	BCD	0 / 9999	Indirizzo di una word in formato BCD (4 digit)
UDINT_BCD	2 word	No	BCD	0 / 99999999	Indirizzo di una doppia word in formato BCD (8 digit)
ULINT_BCD	4 word	No	BCD	0 / 9999999999999999	Indirizzo di un intero in formato BCD a 16 digit.
REAL	2 word	Sì	IEEE	-3.402823 x 10 ³⁸ -1.175494 x 10 ⁻³⁸ 0 1.175494 x 10 ⁻³⁸ 3.402823 x 10 ³⁸	Indirizzo di una doppia word (32Bit) gestita in formato floating point IEEE. Questo formato viene usato per le operazioni floating point dei CVM1-V2 e dei CS1.
CHANNEL	1 o più word	-	-	-	Indirizzo di una o più word. Su questo tipo non vengono effettuati controlli ne conversioni.
NUMBER	-	No	Decimal	-	Nome simbolico utilizzabile al posto di una costante numerica.

Simboli Locali e Globali

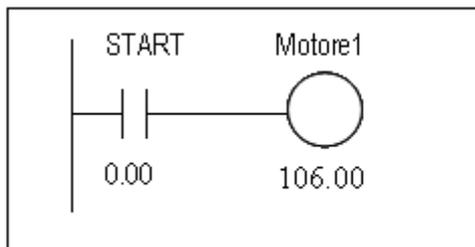
- CX-Programmer è in grado di gestire più PLC o programmi all'interno dello stesso progetto
- I simboli Globali sono comuni a tutti i programmi di un PLC
- I simboli Locali si riferiscono solo al programma in cui sono state definite
- Simboli locali e globali possono avere lo stesso nome anche se si riferiscono a indirizzi diversi
- Se esistono due simboli, uno locale e uno globale con lo stesso nome e indirizzo diverso, il simbolo locale ha la priorità su quello globale

Visibilità dei simboli: esempio

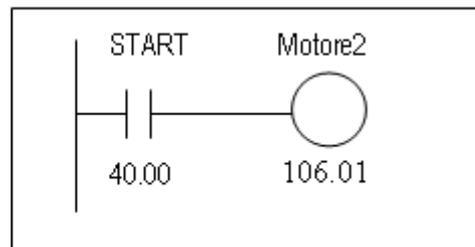
Nome	Tipo dati	Indirizzo / Valore	Posizione rack	Utilizzo	Co
▾ P_Off	BOOL	CF114		Lavoro	
▾ START	BOOL	0.00		Lavoro	
▾ Motore1	BOOL	106.00		Lavoro	
▾ Motore2	BOOL	106.01		Lavoro	

Nome	Tipo dati	Indirizzo / Valore	Posizione rack	Utilizzo	Co
▾ START	BOOL	40.00		Lavoro	

Task 1



Task 2



Simboli globali per il PLC1

Al simbolo globale START è stato assegnato il Bit 0.00,

Simboli locali per il TASK2

Al simbolo locale START è stato assegnato il Bit 40.00

Nel task 1 al simbolo START corrisponde la definizione globale. Nel task 2 la ridefinizione locale del simbolo START copre il simbolo globale

Creare un elenco di simboli con Excel

E' possibile creare con Excel una tabella a tre colonne (nome, indirizzo o valore, commento) e importarla nel CX-Programmer come elenco di simboli con un semplice copia-incolla.

The screenshot shows a Microsoft Excel window titled 'Microsoft Excel - Cartel1'. The spreadsheet contains a table with the following data:

	A	B	C	D	E	F	G
1	I/O Symbol	Address	Comment				
2	Start	0.00	Start Button				
3	Stop	0.01	Stop Button				
4	OverLoad	0.02	Overload Input				
5	Motor	100.00	Motor Run Output				
6	Fan_Delay	T0001	Fan Delay				
7							

Commento dei Programmi

I commenti sul diagramma ladder possono essere inseriti come commenti alle network o ai singoli elementi del diagramma ladder (ad esempio contatti e bobine).



Per commentare un elemento del ladder, è sufficiente selezionarlo con il tasto destro e scegliere l'opzione "Proprietà". Lo stesso meccanismo si applica anche ai Rung.



L' archiviazione

Una volta completato il progetto è possibile salvarlo su disco sotto forma di file richiamando la voce "*Salva con nome...*" dal menù "*File*".

Tutti i file salvati possono essere richiamati selezionando la voce "*Apri*" dal menù "*File*".

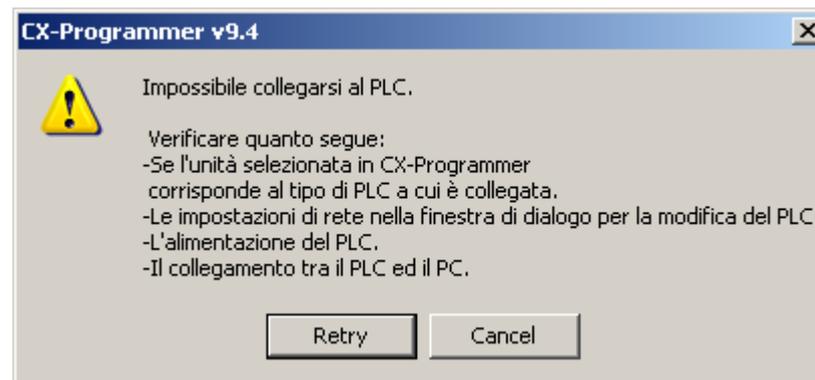


Lavorare 'On-Line'

- Per attivare la comunicazione con il PLC, cliccare sull'icona  oppure selezionare la voce 'Funzionamento in Linea' del menù PLC:



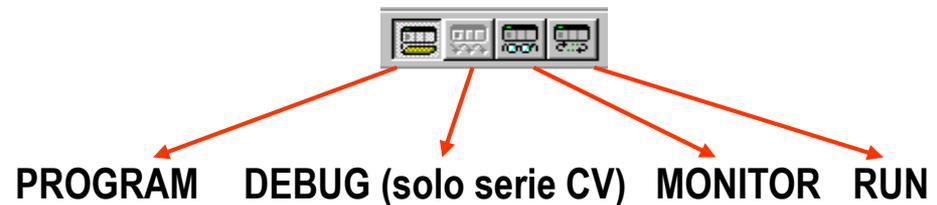
- Se il PLC è collegato correttamente, lo sfondo dell'editor del ladder varierà il proprio colore (da bianco a grigio); in caso contrario comparirà la seguente finestra di errore:



Modalità Operativa

- Quando si è in collegamento con il PLC, la modalità operativa viene indicata:

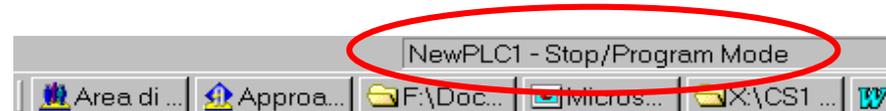
- Sulla barra delle icone



- Nella finestra di progetto



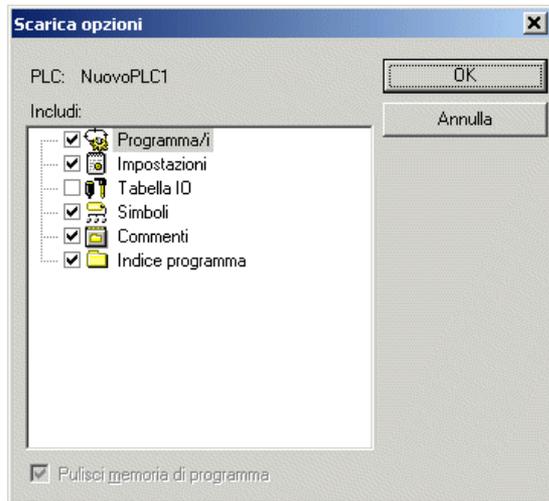
- Sulla barra di stato



- Dalla barra delle icone è anche possibile cambiare la modalità del PLC selezionando l'icona corrispondente alla modalità desiderata

Trasferire il programma

- Collegarsi al PLC e cliccare sull'icona  oppure selezionare l'opzione "Trasferisci... Al PLC" dal menù PLC



Comparirà una finestra per la selezione delle opzioni di download, in modo da stabilire quali elementi debbano essere trasferiti insieme al programma. Per scaricare il programma il PLC NON può essere ne in modalità MONITOR ne in modalità RUN

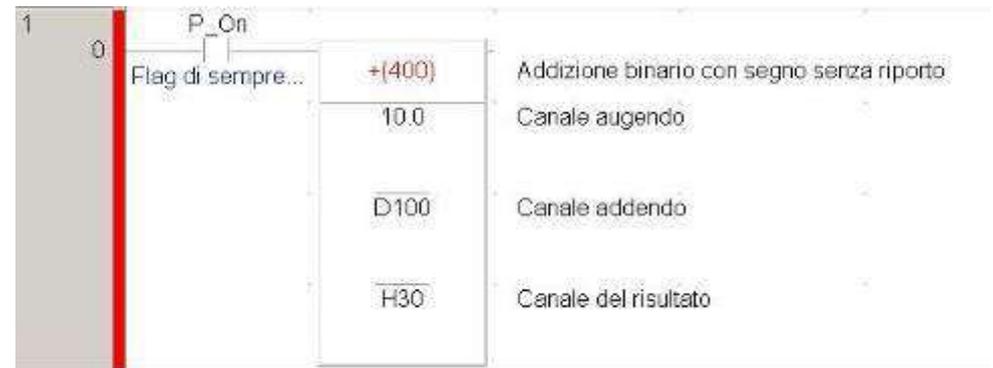
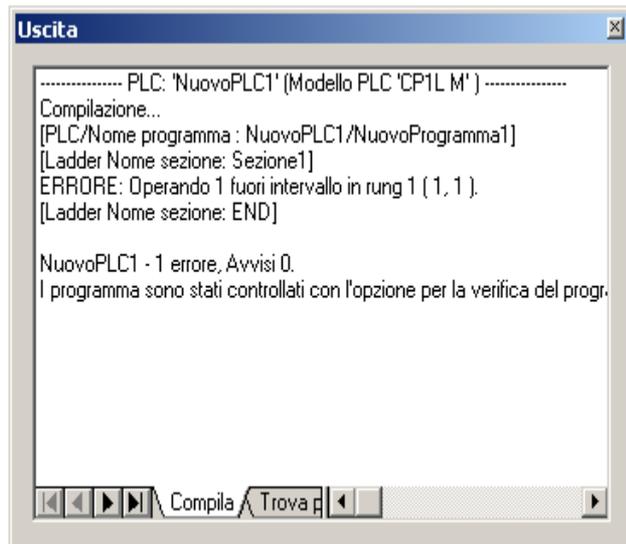
- Per trasferire il programma dal PLC al PC utilizzare l'icona  o selezionare l'opzione "Trasferisci... Dal PLC" dal menù PLC
- Il caricamento del programma è possibile in qualunque modalità

Verifica del Programma

- Il controllo di programma viene effettuato durante la compilazione
- Gli eventuali errori o avvisi sono mostrati nella finestra di Output
- Facendo un doppio click sull'errore, il cursore verrà posizionato sull'istruzione errata
- Il controllo può essere avviato agendo sull'apposita icona o selezionando l'opzione *Compila* del menu *Program*

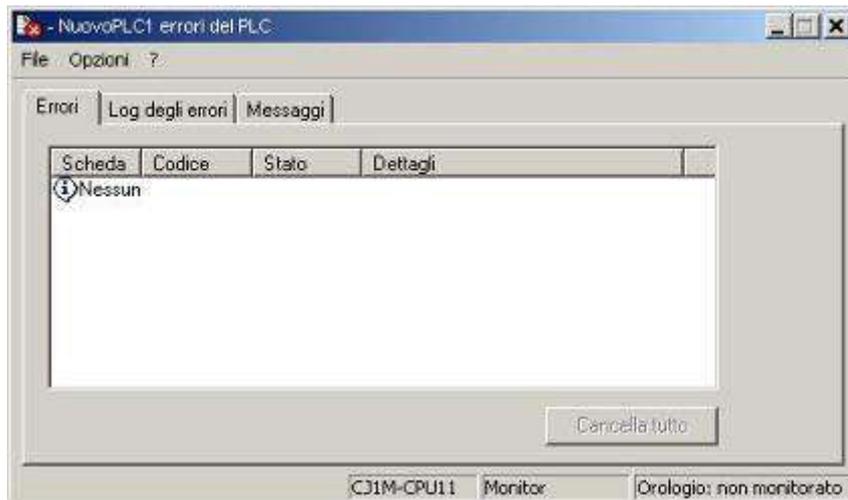
Verifica del Programma

- Nella finestra output vengono visualizzati i risultati della compilazione
- AVVISO: errori non fatali, è possibile il download
- ERRORE: errori fatali, non è possibile il download



Cancellazione Errori

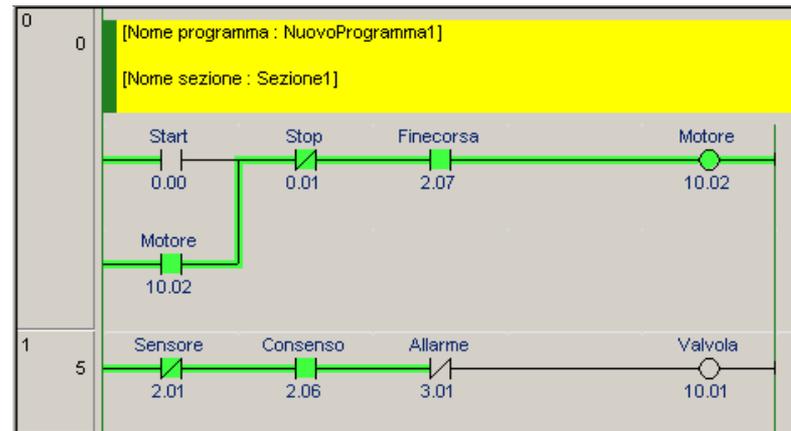
- Selezionando Registro degli Errori nell'Area di progetto compare una finestra nella quale vengono mostrati gli errori registrati nella memoria del PLC



Gli errori compaiono in ordine di importanza, dal codice di errore si risale al problema riscontrato dal PLC.

Monitoraggio On-Line

- Grazie a CX-Programmer è possibile monitorare l'esecuzione del programma
- Le linee di programma 'attive' sono visualizzate da una linea evidenziata con un colore configurabile dall'utente



- Per attivare/disattivare il monitoraggio cliccare sull'icona



Impostazione Bit

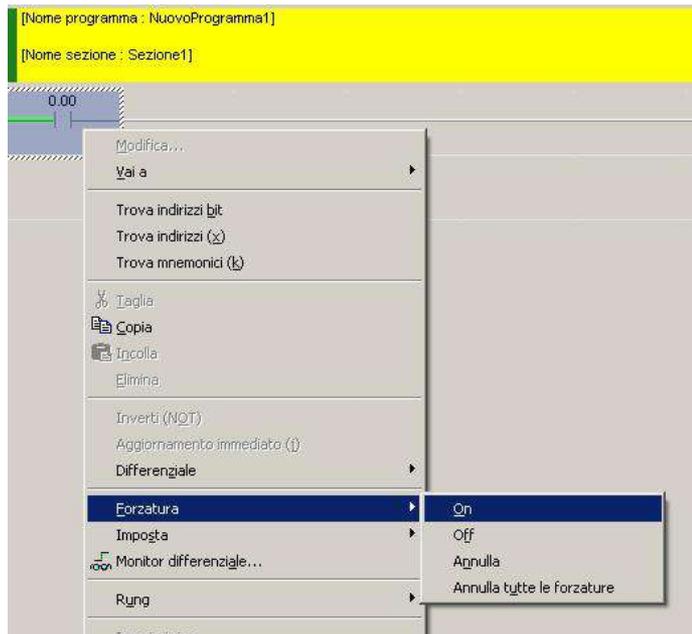
- Per impostare lo stato di un bit è sufficiente portarsi col cursore sul contatto o la variabile e premere il tasto destro



In questo modo è possibile modificare lo stato di un Bit.

Forzatura dei Bit

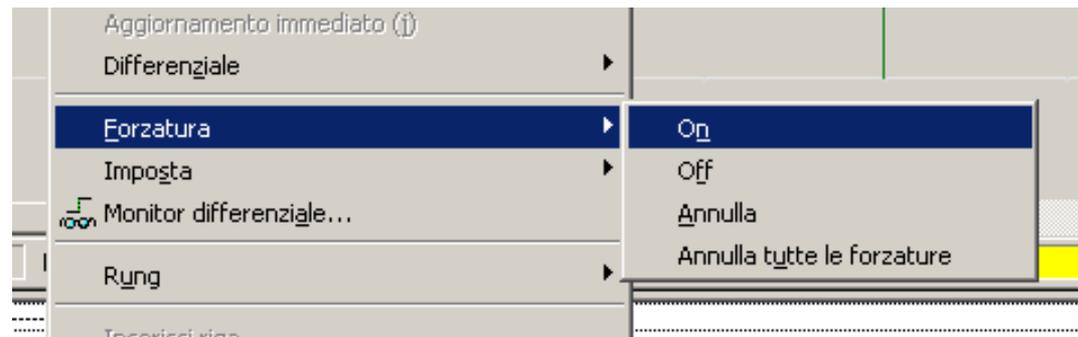
- Un Bit forzato mantiene il proprio stato On oppure Off indipendentemente dalle operazioni del PLC.



Per impostare lo stato forzato di un bit è sufficiente portarsi col cursore sul contatto premere il tasto destro e selezionare “Forzatura”.

“Annulla” rimuove la condizione di forzatura.

E’ anche possibile annullare la forzatura di tutti i Bit del PLC



Impostazione dei Dati

- Per impostare un bit o il valore di una variabile è sufficiente portarsi col cursore sul contatto o la variabile e premere il tasto destro

1) Selezionare il valore con il tasto destro

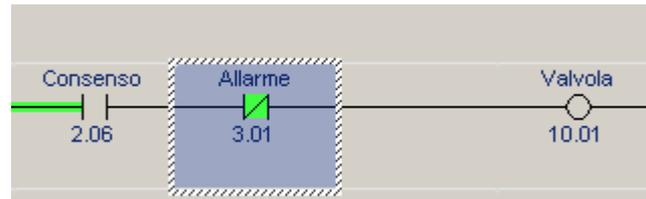
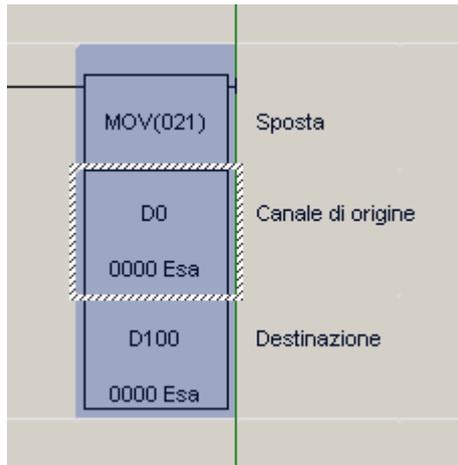
2) Selezionare "Imposta" poi "valore"

3) Digitare il nuovo valore

La finestra di immissione mostra anche il range ammissibile per quel tipo di dato.

Impostazione Bit e dati (alternativa)

- In alternativa, per impostare un bit o il valore di una variabile è sufficiente portarsi col cursore sul contatto e premere “invio”



1) Selezionare il valore o il contatto e premere invio

2) Digitare il nuovo valore

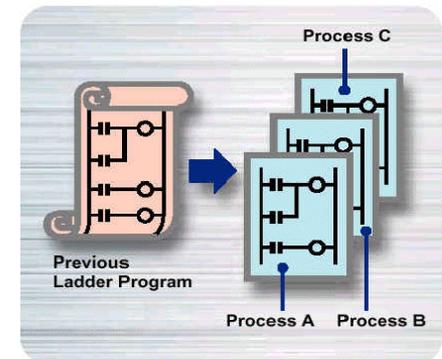
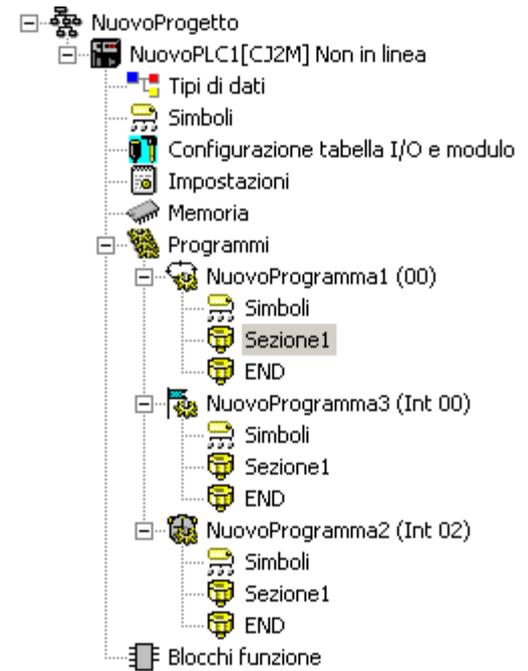
La finestra di immissione mostra anche il range ammissibile per quel tipo di dato. Per i Bit sono ammessi solo i valori “0” e “1”

La modifica On-Line

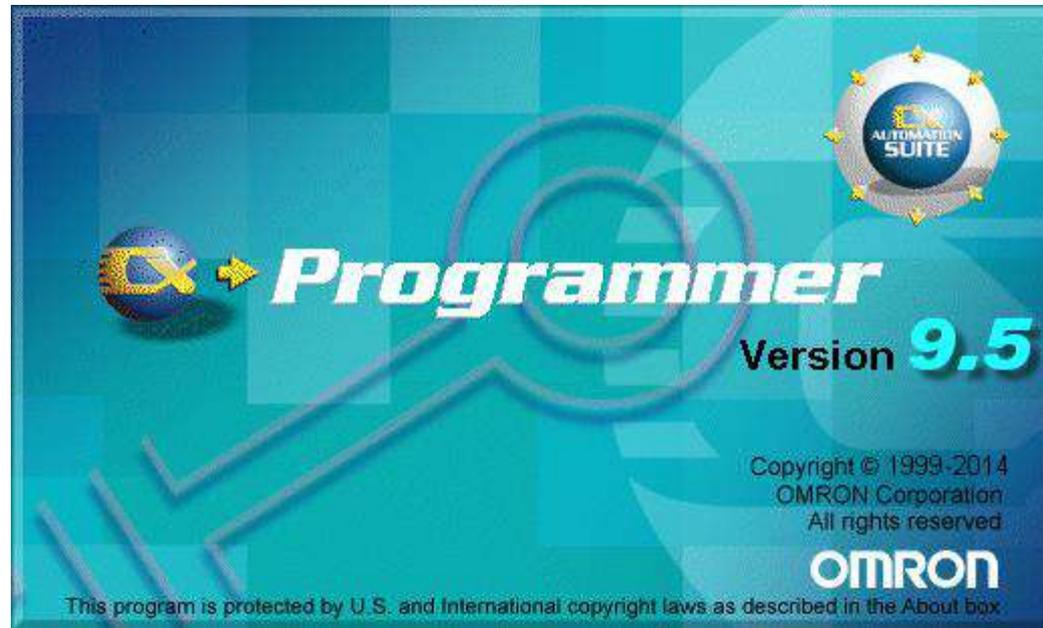
- Per attivare le modifiche online
 - Selezionare una o più network
 - Selezionare l'icona 
 - CX-Programmer effettuerà il controllo di corrispondenza della network selezionata.
 - L'area attiva per la modifica diventerà bianca
- All'interno dell'area attiva è possibile effettuare qualsiasi modifica. E' anche possibile aggiungere network allargando l'area selezionata
- Per confermare le modifiche selezionare l'icona 
- Per annullare le modifiche utilizzare l'icona 

Suddivisione dei programmi

- Nelle famiglie CP1 e CJ1:
 - Il programma può essere suddiviso in 32 Task ciclici (128 nel CJ2) e 256 Task ad interrupt
 - I Task ciclici possono essere attivati o disattivati dall'interno del programma
 - All'interno dei task possono essere utilizzate subroutine (1.024 in totale)
 - Gli Interrupt, siano essi hardware o software, sono gestiti da Task dedicati
- L'esecuzione dei Task è sequenziale
- La memoria del PLC è comune a tutti i Task



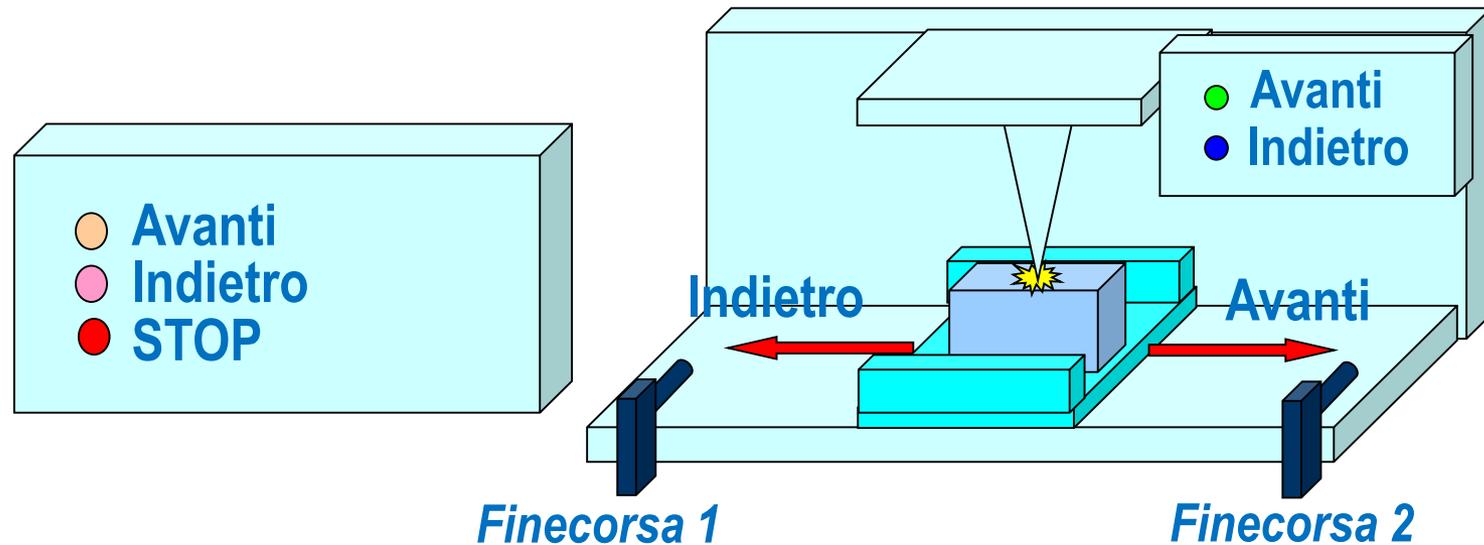
Il Mio Primo Programma



Flusso di lavoro

- Per la costruzione di un programma è consigliabile seguire i seguenti passaggi:
 - Risoluzione schematica del problema (schema a blocchi)
 - Allocazione degli I/O
 - Scelta del PLC e configurazione
 - Scrittura dei programmi
 - Test
 - Produzione della documentazione (fase di rilascio)

Esercizio 1: Controllo di una macchina

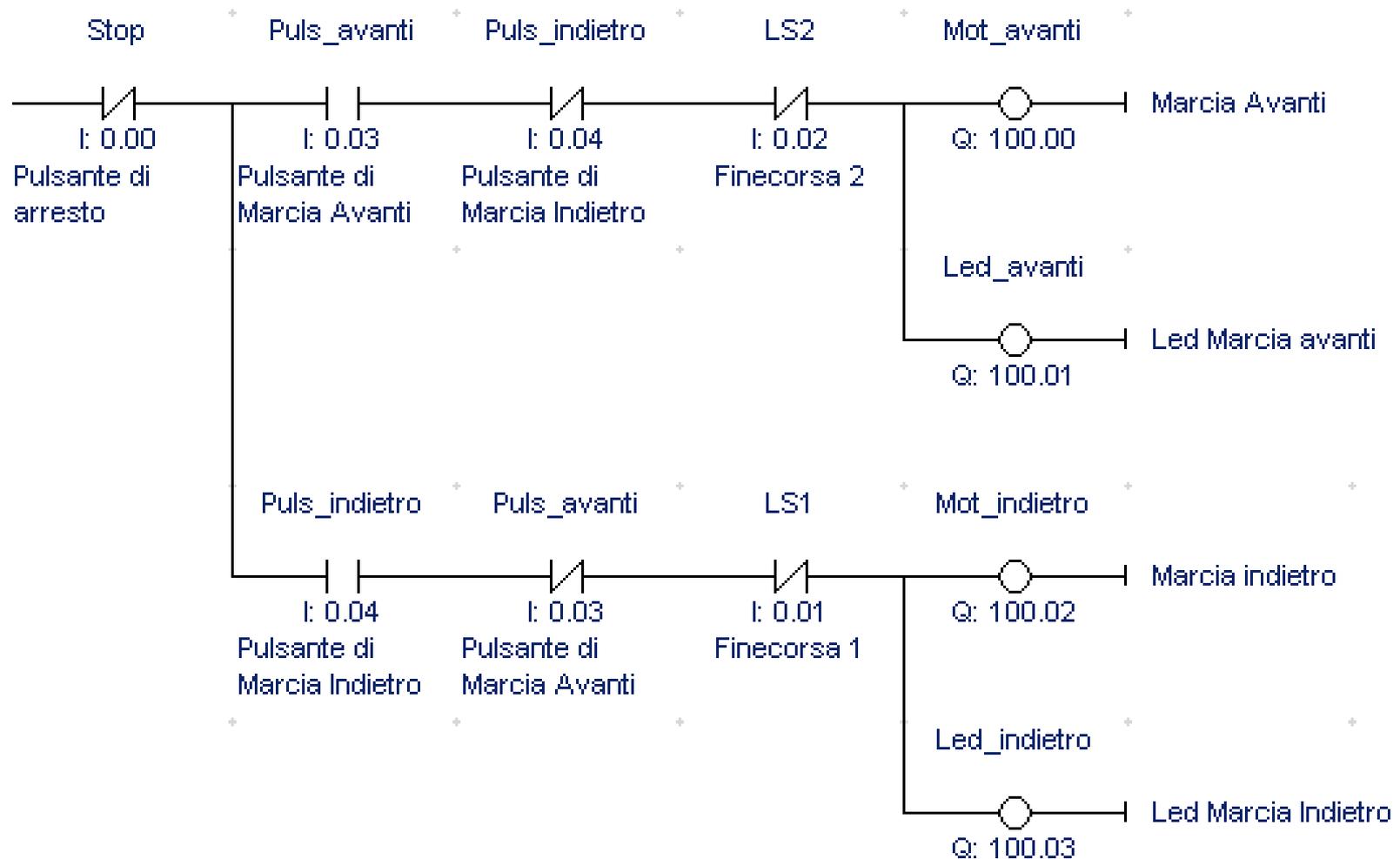


Nome	Tipo	Indirizzo / Valore	Posizione rack	Usò	Commento
^ Mot_avanti	BOOL	100.00			Marcia Avanti
^ Mot_indietro	BOOL	100.02			Marcia indietro
^ Led_avanti	BOOL	100.01			Led Marcia avanti
^ Led_indietro	BOOL	100.03			Led Marcia Indietro
^ LS1	BOOL	0.01			Finecorsa 1
^ LS2	BOOL	0.02			Finecorsa 2
^ Puls_avanti	BOOL	0.03			Pulsante di Marcia Avanti
^ Puls_indietro	BOOL	0.04			Pulsante di Marcia Indietro
^ Stop	BOOL	0.00			Pulsante di arresto

Esercizio 1: Controllo di una macchina

- Il pulsante 'Puls_avanti' finchè premuto determina la marcia in avanti del carrello fino al raggiungimento del finecorsa LS2 o alla pressione del comando di 'Stop'
- Il pulsante 'Puls_indietro' finchè premuto determina la marcia indietro del carrello, fino al raggiungimento del finecorsa LS1 o alla pressione del comando di 'Stop'
- Se i due pulsanti di 'Puls_avanti' e di 'Puls_indietro' vengono premuti simultaneamente il carrello non deve muoversi in alcuna direzione

Esercizio 1: Soluzione



Le Funzioni

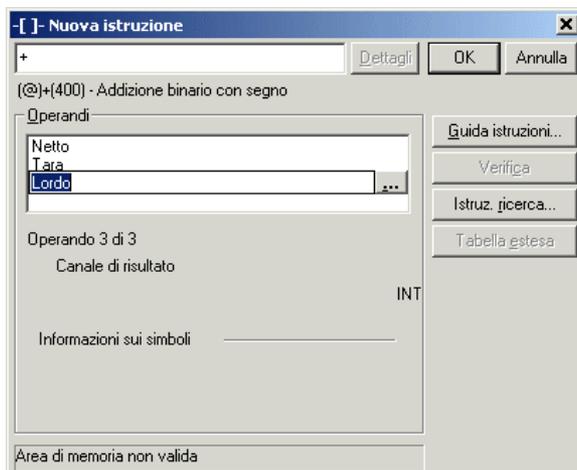
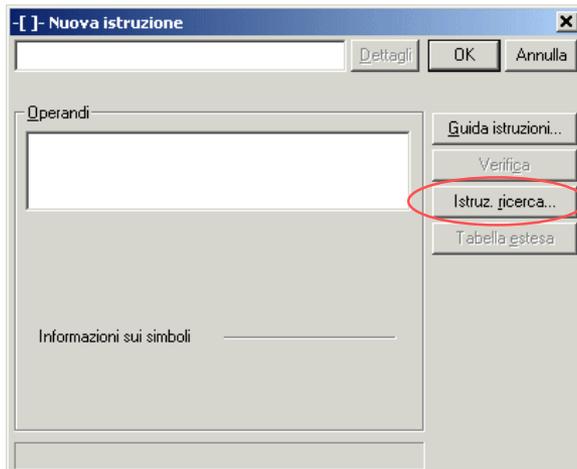
Istruzioni su Bit, TIM, CNT e
movimento dati



Le Funzioni

- Rappresentano quella serie di istruzioni per le quali non è previsto sulla console di programmazione il corrispondente tasto dedicato
- Ad ogni funzione è associato, in modo univoco, un codice numerico di tre cifre. E' possibile richiamare una funzione anche inserendo direttamente il codice
- Nel diagramma a relè le funzioni vengono rappresentate con un blocco funzionale in cui compare il nome dell'istruzione, insieme ai relativi parametri
- La serie CJ/CS e CP1 dispone di un set di oltre 400 istruzioni, così suddiviso:
 - 14 Istruzioni Base (schema a contatti)
 - circa 400 Istruzioni Avanzate (funzioni)

Le Funzioni



- Selezionando tra le icone di progetto l'icona  si apre la finestra di immissione di una funzione
- E' possibile richiamare direttamente la funzione, digitando il nome corrispondente, oppure è possibile selezionarla da una lista disponibile facendo clic sul tasto di ricerca
- Una volta selezionata la funzione è possibile inserirne i parametri (se richiesti) e decidere se utilizzare o meno la forma differenziale (se consentito)

Uso delle costanti numeriche

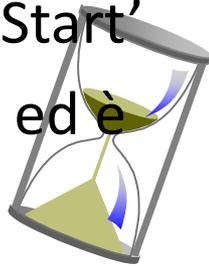
- Occorre prestare attenzione durante l'inserimento di costanti come operandi di istruzioni

Method	Applicable operands	Data format	Code	Range
Constant (16-bit data)	All binary data and binary data within a range	Unsigned binary	#	#0000 to #FFFF
		Signed decimal	+ -	-2,147,483,648 to +2,147,483,647
		Unsigned decimal	& (Note)	&0 to &65,535
	All BCD data and BCD data within a range	BCD	#	#0000 to #9999

- Dalla tabella si nota che per operandi di istruzioni binarie, sono possibili tre modi di inserimento di costanti (#, &, + e -)
- &, + e - si usano per inserire valori decimali mentre # denota un valore esadecimale o BCD

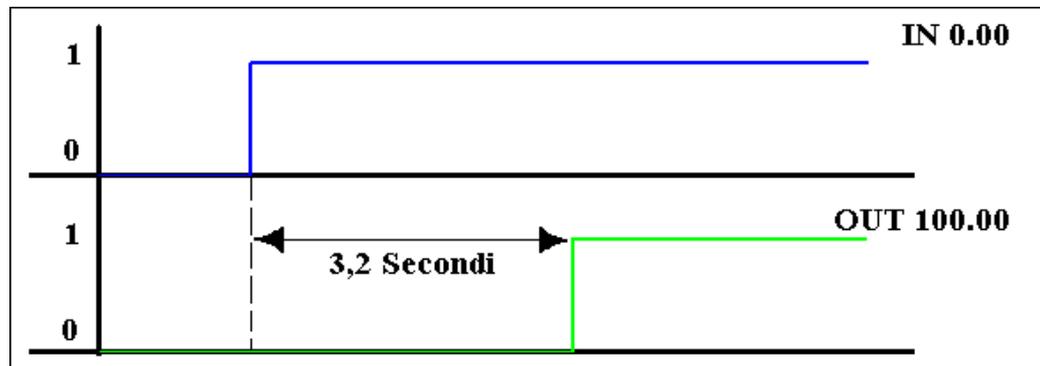
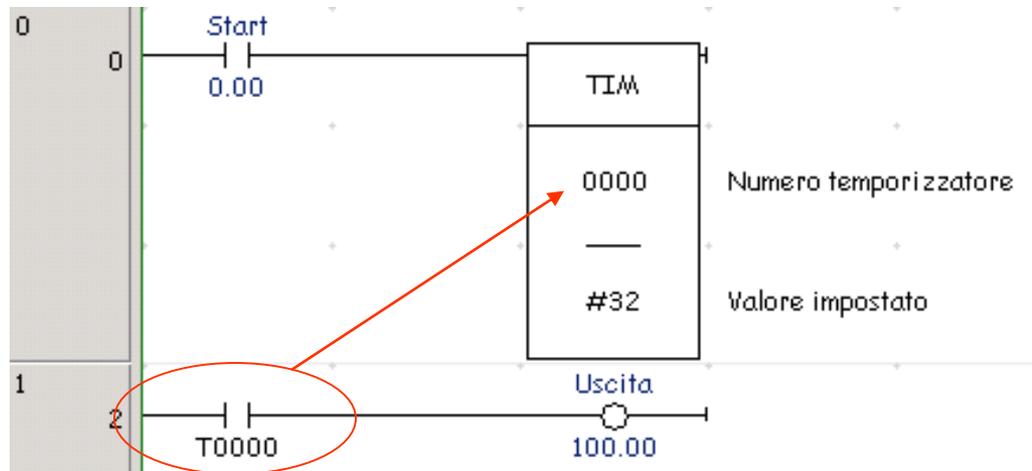
Timer

- L'istruzione TIM (Timer) viene utilizzata per generare un ritardo all'eccitazione rispetto al segnale di abilitazione 'Start'
- Il ritardo SV (Set Value) può variare da 0 a 999,9 secondi ed è programmabile in unità di 0,1 secondi
- Quando il segnale di Start va ad ON il valore attuale PV (Present Value) del timer (inizialmente pari all'SV) inizia a decrementare
- Quando il PV (non ritentivo) diventa uguale a zero, lo stato del timer va ad uno
- Le famiglie di PLC CJ/CS e CP1 dispongono di 4.096 timer (da 0 a 4.095) identificati univocamente dal proprio numero



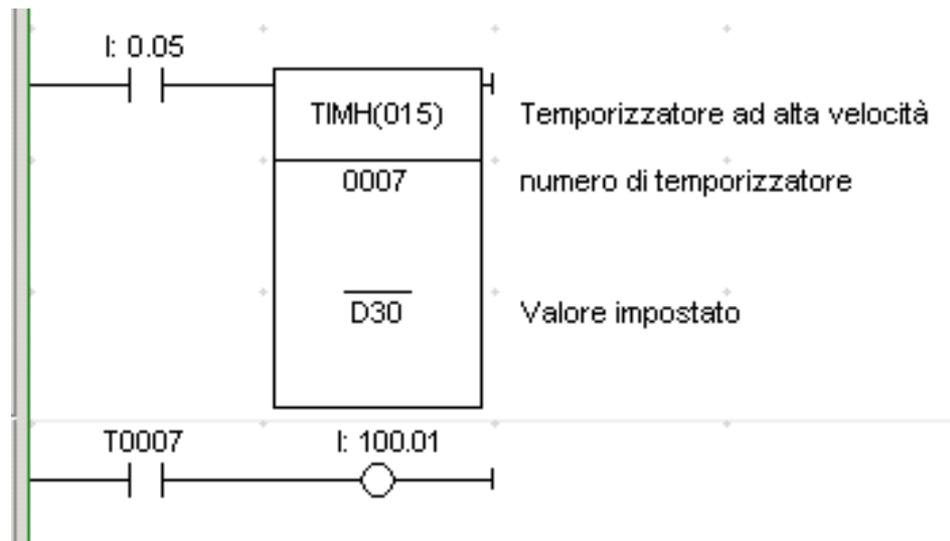
Esempio di utilizzo di un timer

- Il segnale di Start 0.00 eccita l'uscita 100.00 dopo 3,2 secondi



La funzione TIMH (015)

- La funzione TIMH (timer ad alta velocità) serve per generare ritardi all'eccitazione con una precisione maggiore rispetto alla funzione TIM
- Il ritardo (SV) può variare tra 0,00 e 99,99 secondi ed è programmabile in unità di centesimi di secondo

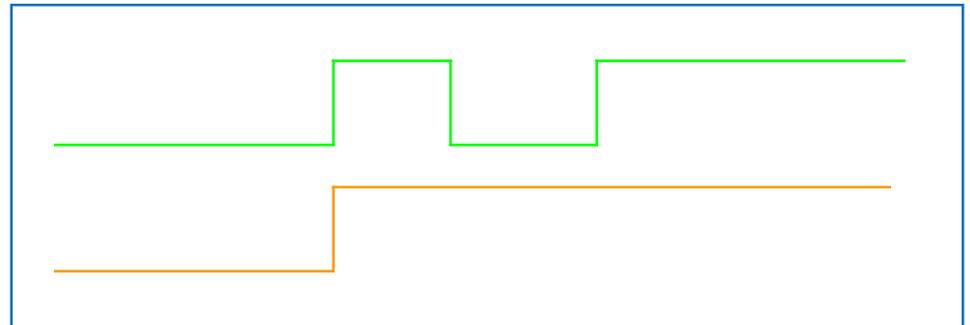
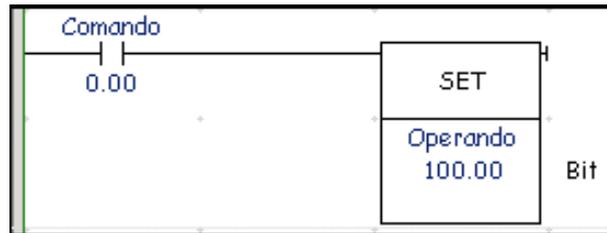


La funzione TIMH (015)

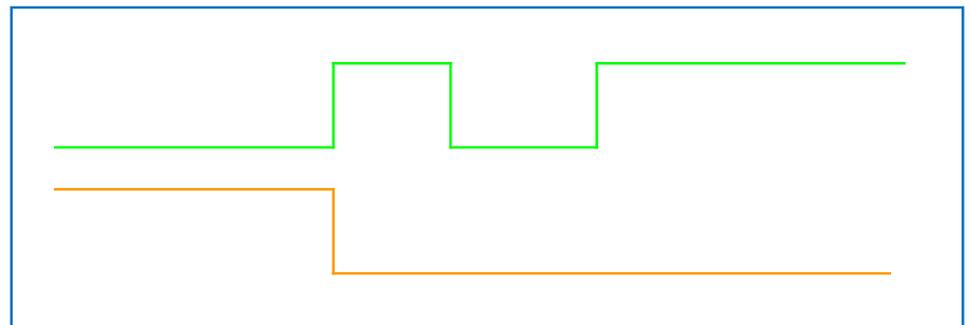
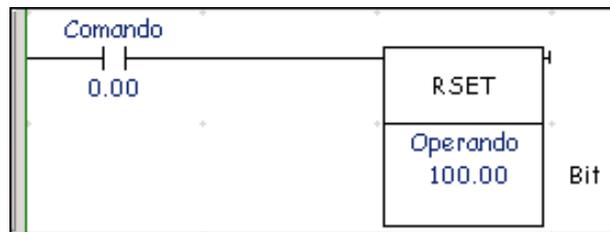
- Se il tempo di scansione supera i 10 ms l'accuratezza per i TIMH diminuisce a meno che non si utilizzi un numero di timer compreso tra 0 e 255
- Nelle famiglie di PLC CS e CJ il PV dei primi 256 timer è infatti rinfrescato 'ad interrupt' ogni 10 ms e non all'esecuzione dell'istruzione come invece avviene per i timer successivi
- Tutti i timer della famiglia CP1 lavorano invece con un tempo di refresh pari a 100 ms

Le istruzioni di SET e RESET

- L'istruzione SET imposta ad 1 il bit indicato come operando



- L'istruzione RSET imposta ad 0 il bit indicato come operando



Fronte di salita e funzioni differenziate

- Con fronte di salita (discesa) di un segnale, si intende la condizione per cui, nella scansione precedente quel segnale era stato letto ad OFF (ON) dal PLC e nella scansione attuale viene letto ad ON (OFF)
- Quasi tutte le funzioni integrate nel set di istruzioni del PLC possono essere eseguite in due modalità:
 - ad ogni scansione se è presente il segnale di abilitazione della funzione
 - solo per una scansione sul fronte di salita (o di discesa) del segnale di abilitazione della funzione (funzione differenziata)

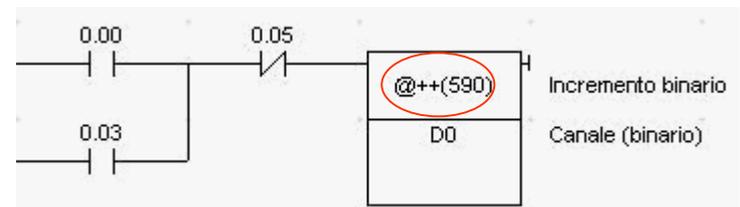
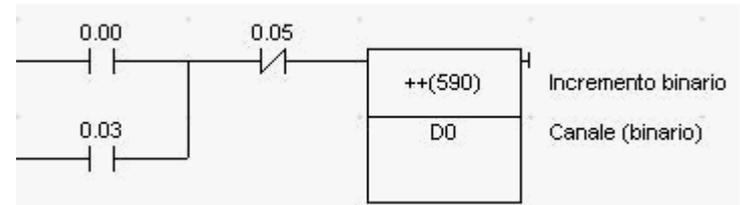
Fronte di salita e funzioni differenziate

- Per differenziare una funzione è sufficiente inserire il carattere '@' prima di digitarne il nome

- Esempio:

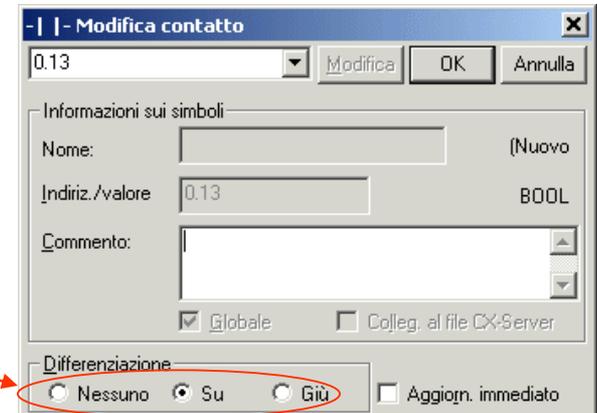
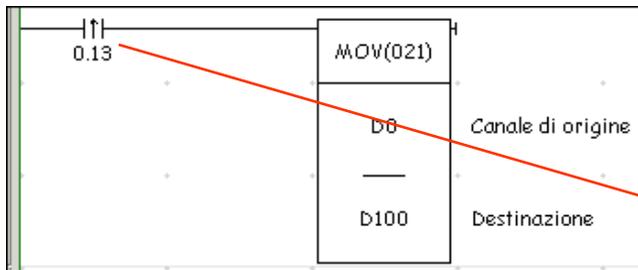


- L'istruzione '++' viene eseguita ad ogni scansione se il segnale di abilitazione è ad ON
- L'istruzione '@++' viene eseguita per una sola scansione sul fronte di salita del segnale di abilitazione se è ad ON



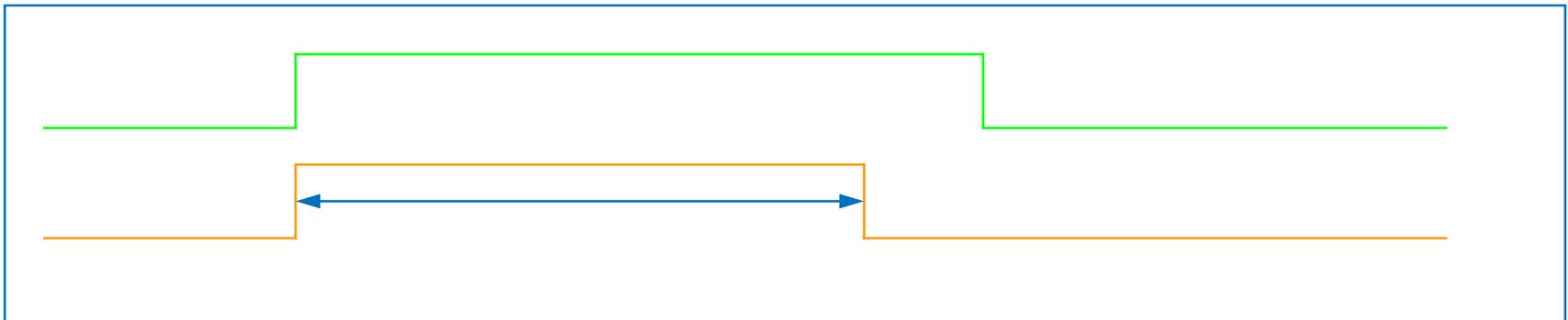
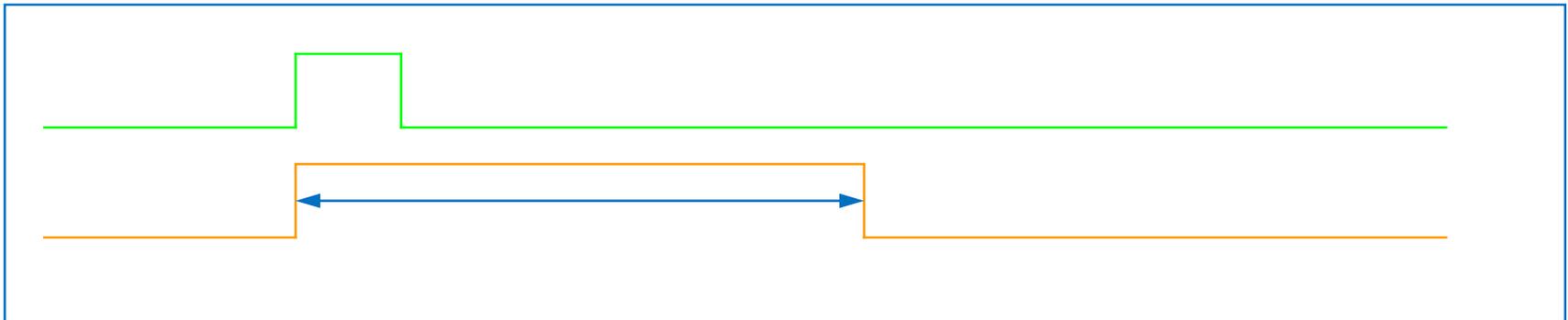
Fronte di salita e funzioni differenziate

- Originariamente nel set di istruzioni del PLC erano previste due funzioni utili per intercettare i fronti di salita (DIFU) o di discesa (DIFD) dei segnali, in modo da poter controllare con un bit l'esecuzione di tutte le istruzioni non differenziabili
- A partire dalle famiglie CS/CJ e CP1 è invece possibile differenziare i contatti direttamente durante il loro inserimento, o utilizzare delle istruzioni intermedie (UP e DOWN) consentendo di risparmiare memoria di programma

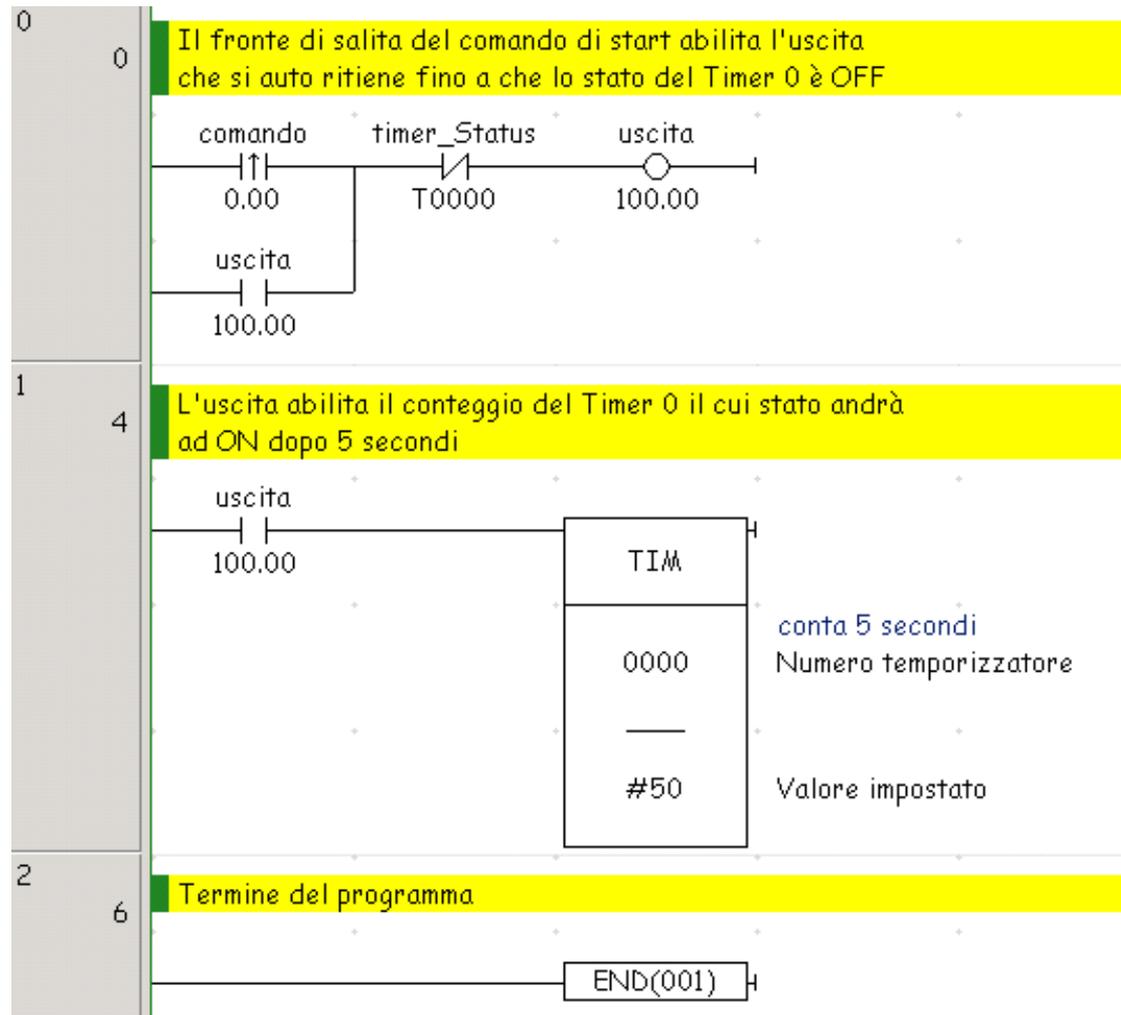


Esercizio 2: Timer monostabile

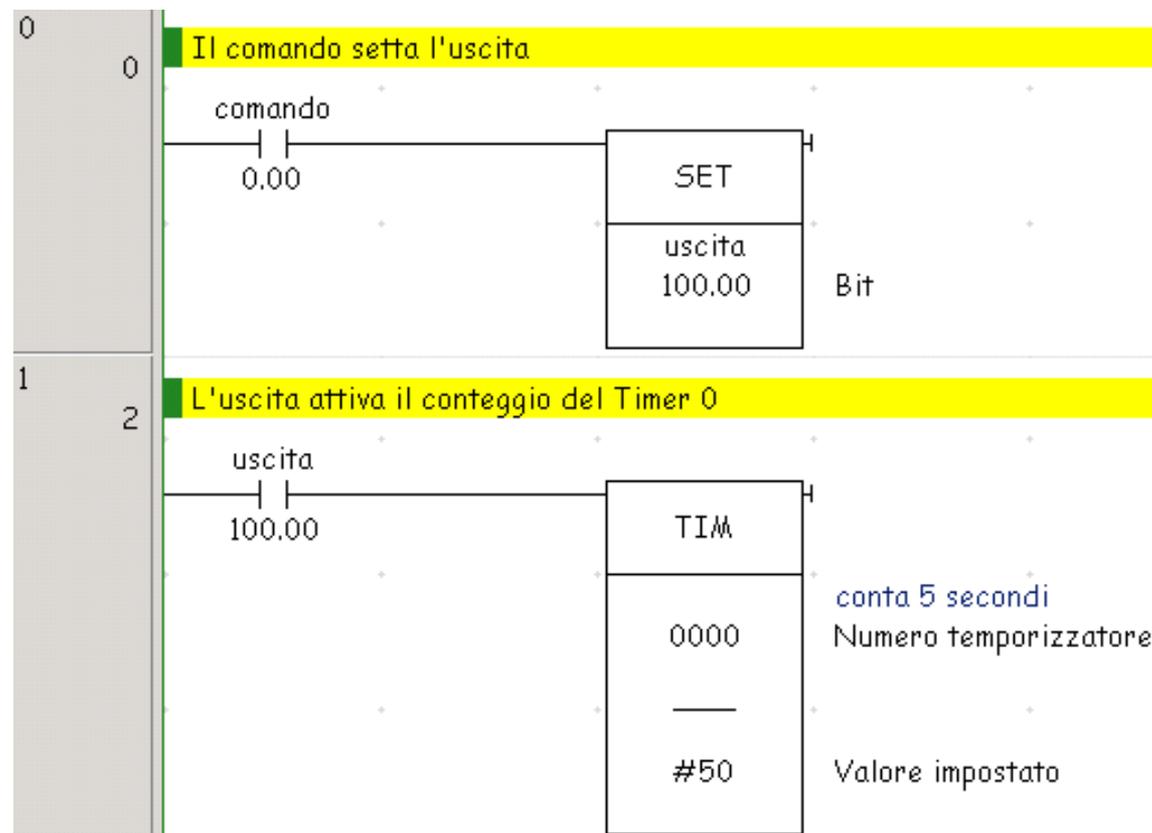
- Realizzare un timer che fissi la durata dell'attivazione di un'uscita a 5 secondi, indipendentemente dalla durata del segnale di Start



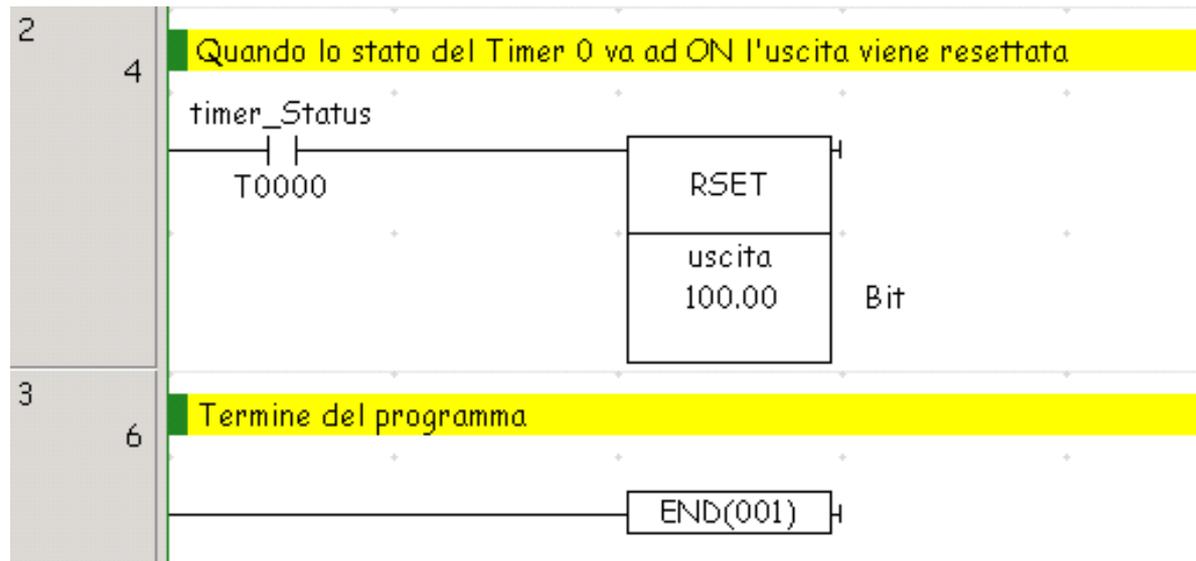
Esercizio 2/A: Soluzione con autoritenuta



Esercizio 2/B: Soluzione con SET e RESET

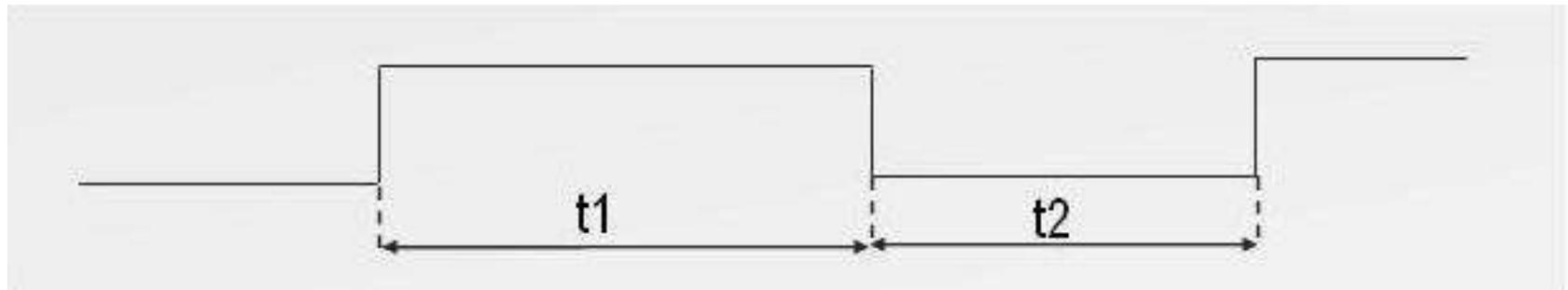


Esercizio 2/B: Soluzione con SET e RESET

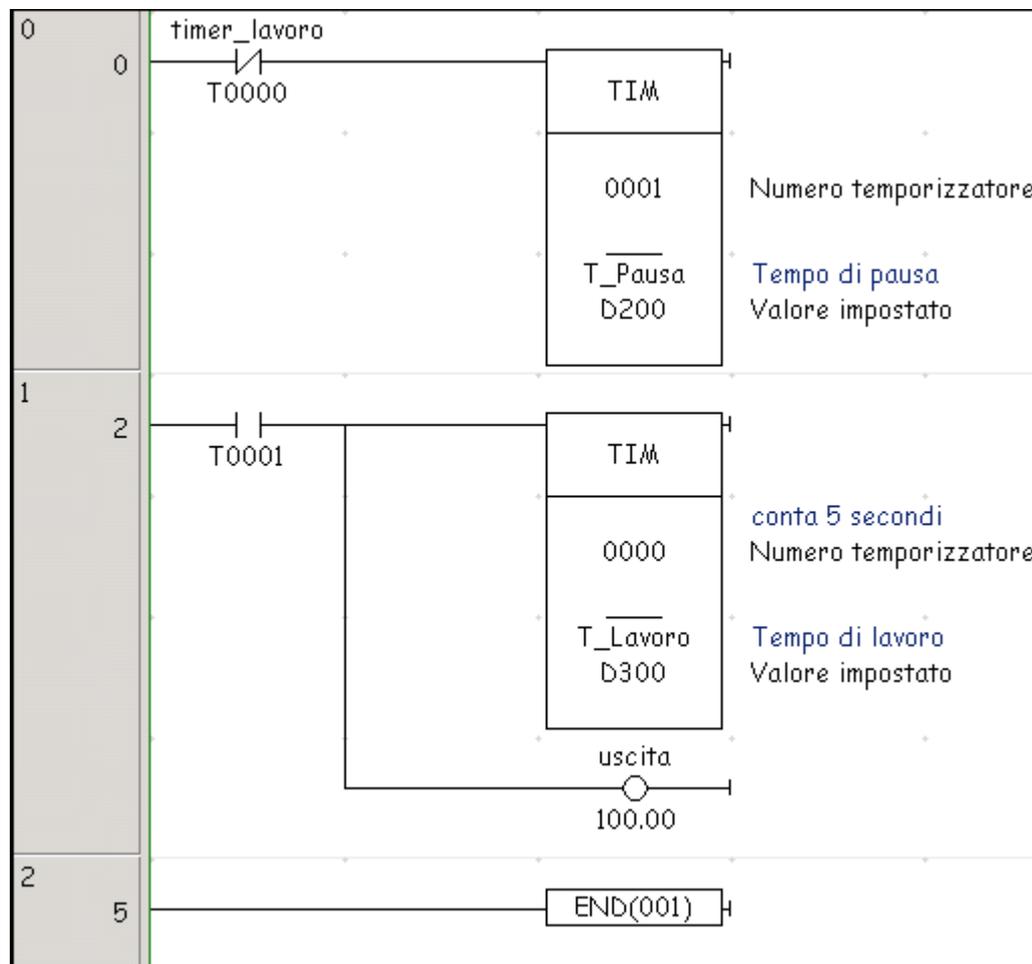


Esercizio 2/C: Timer Pausa/Lavoro

- Utilizzando due timer in combinazione tra loro, realizzare un'onda con Duty Cycle variabile



Esercizio 2/C: Soluzione



I nuovi temporizzatori

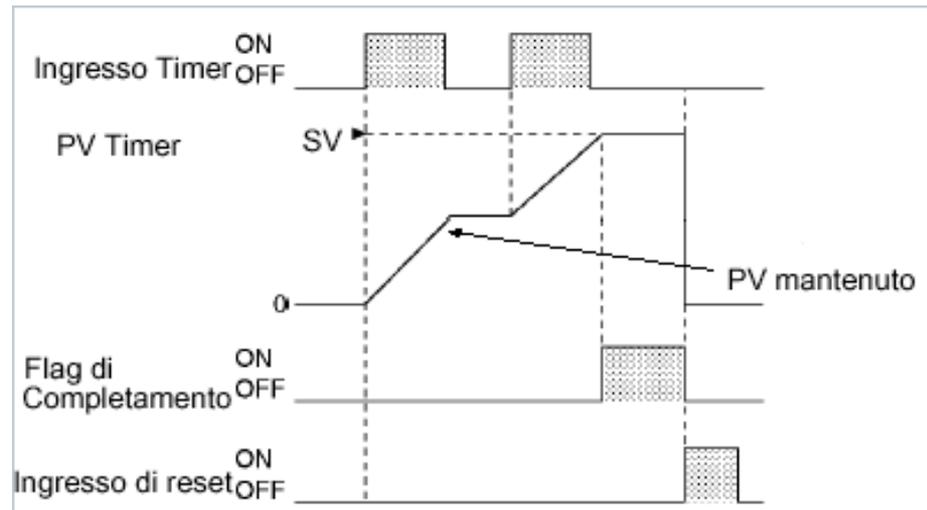
- Le più recenti CPU delle famiglie di PLC CS/CJ e CP1 supportano una nuova gamma di temporizzatori non presente nelle serie di PLC precedenti
- La nuova gamma di temporizzatori comprende:
 - Funzione di temporizzatore ritentivo TTIM(087)
 - Funzione di temporizzatore lungo TIML(542)
 - Funzione di temporizzatore multiplo MTIM(543)

Timer totalizzatore (TTIM)

- Temporizzatore TTIM(087): Simula un temporizzatore ad incremento con ritardo programmabile in unità di decimi di secondo. L'SV può variare tra 0 e 999,9 secondi ed il PV viene mantenuto in caso di interruzione del segnale di abilitazione

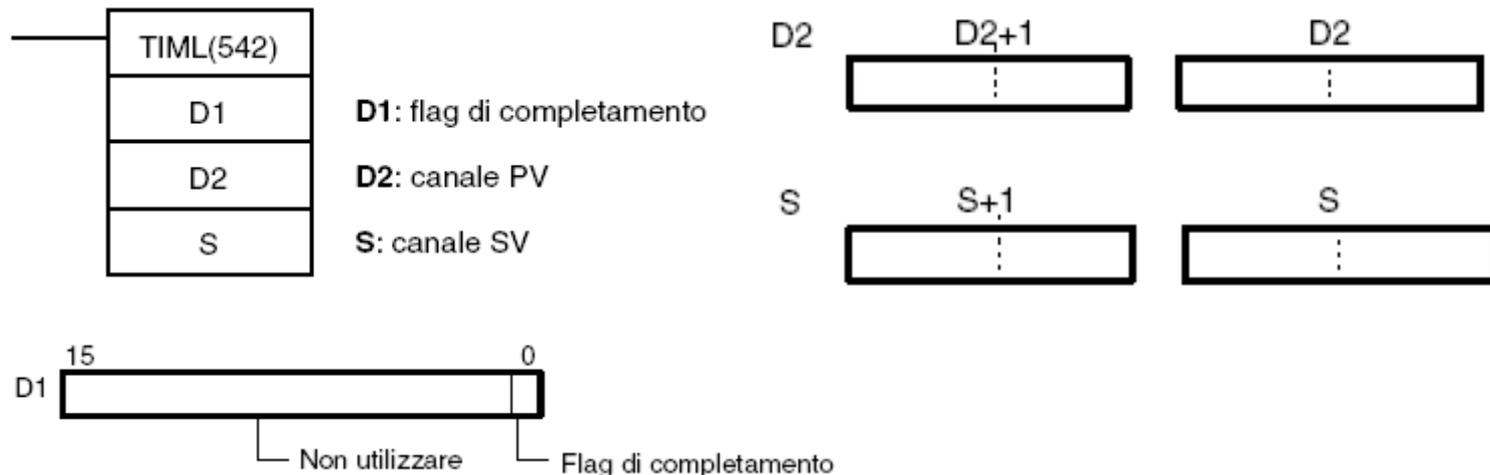


N: Numero Timer
S: Set value



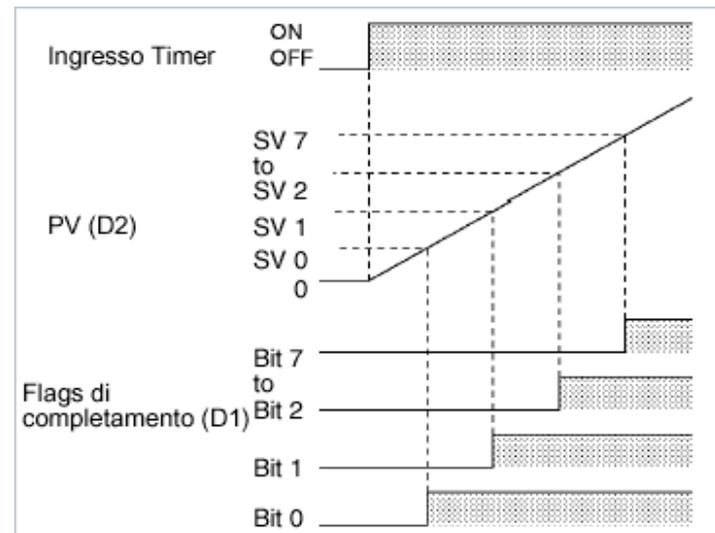
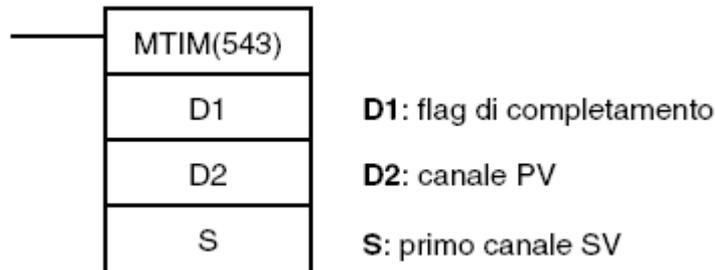
Long Timer (TIML)

- Temporizzatore lungo TIML(542): Simula un temporizzatore a decremento, con ritardo programmabile in unità di decimi di secondo, il cui SV può variare tra 0 e 9999999,9 secondi (115 giorni circa) in modalità BCD e tra 0 e 429496729,4 secondi (49.710 giorni) in modalità Hex (TIMLX)



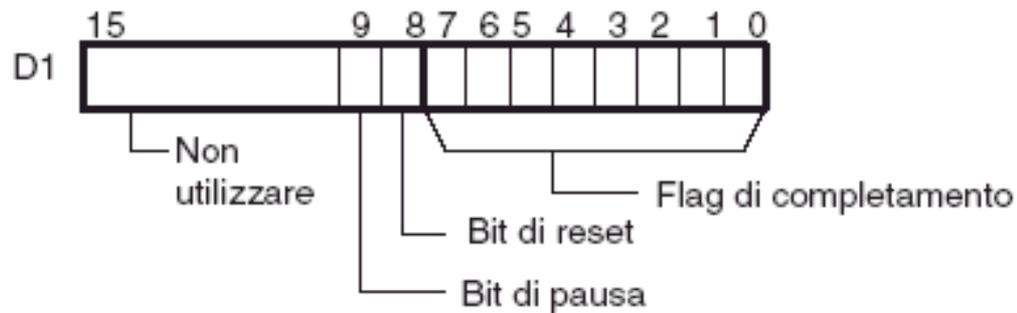
Timer a impostazione multipla

- Temporizzatore multi uscita MTIM(543): Simula un temporizzatore a decremento con 8 set value ed 8 flag di completamento indipendenti. Il ritardo è programmabile in unità di decimi di secondo (da 0 a 999,9 secondi)



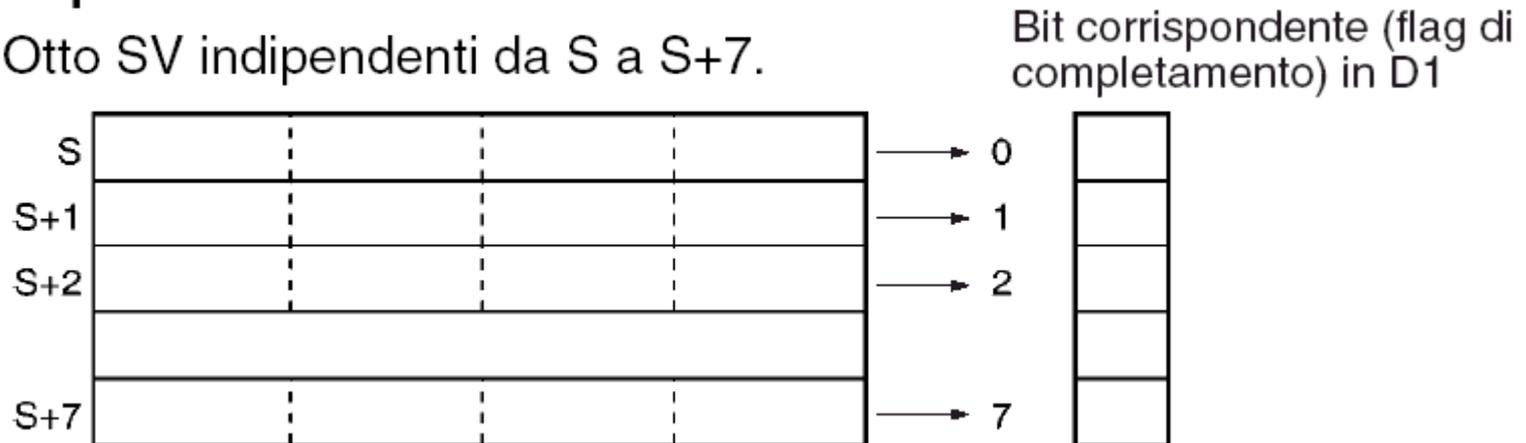
Timer a impostazione multipla

D1 contiene gli otto flag di completamento nonché i bit di pausa e di reset.



S: primo canale SV

Otto SV indipendenti da S a S+7.

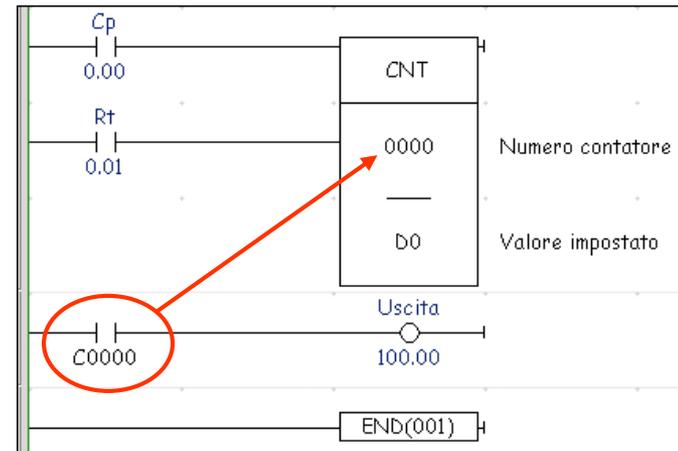


La funzione Contatore

- La funzione contatore (CNT) simula un contatore sottraente a predisposizione. Il valore di predisposizione (SV) può variare tra 0 e 9999

Il contatore ha due ingressi:

- Ingresso di conteggio impulsi
- Ingresso di reset



- Se Rt è OFF, ogni fronte di salita di Cp determina il decremento del PV. Quando il PV diventa uguale a zero lo stato del contatore va ad ON

La funzione Contatore

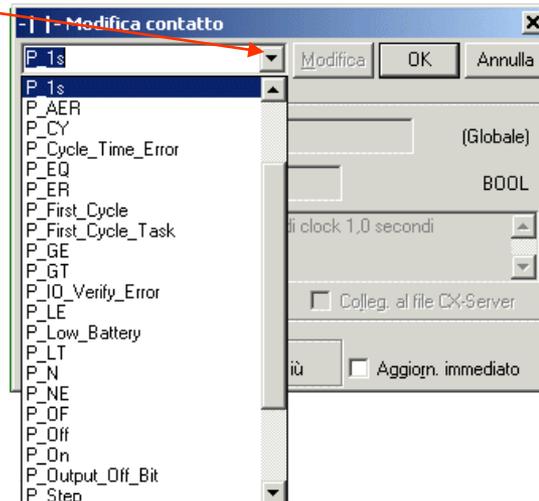
- I contatori sono ritentivi, ovvero il loro stato (bit di uscita e PV) viene mantenuto sia in caso di Power Off che di cambio della modalità operativa del PLC
- Quando il PV è uguale a 0 tutti gli eventuali impulsi di ingresso vengono ignorati
- Se Rt va ad ON il contatore si resetta (PV=SV, stato=OFF)
- Accoppiato ad un clock di sistema, un contatore può essere utilizzato per realizzare un timer ritentivo
- Accoppiando due contatori in cascata tra loro, si ottiene un conteggio pari al prodotto di SV1 e SV2

Timer e Counter

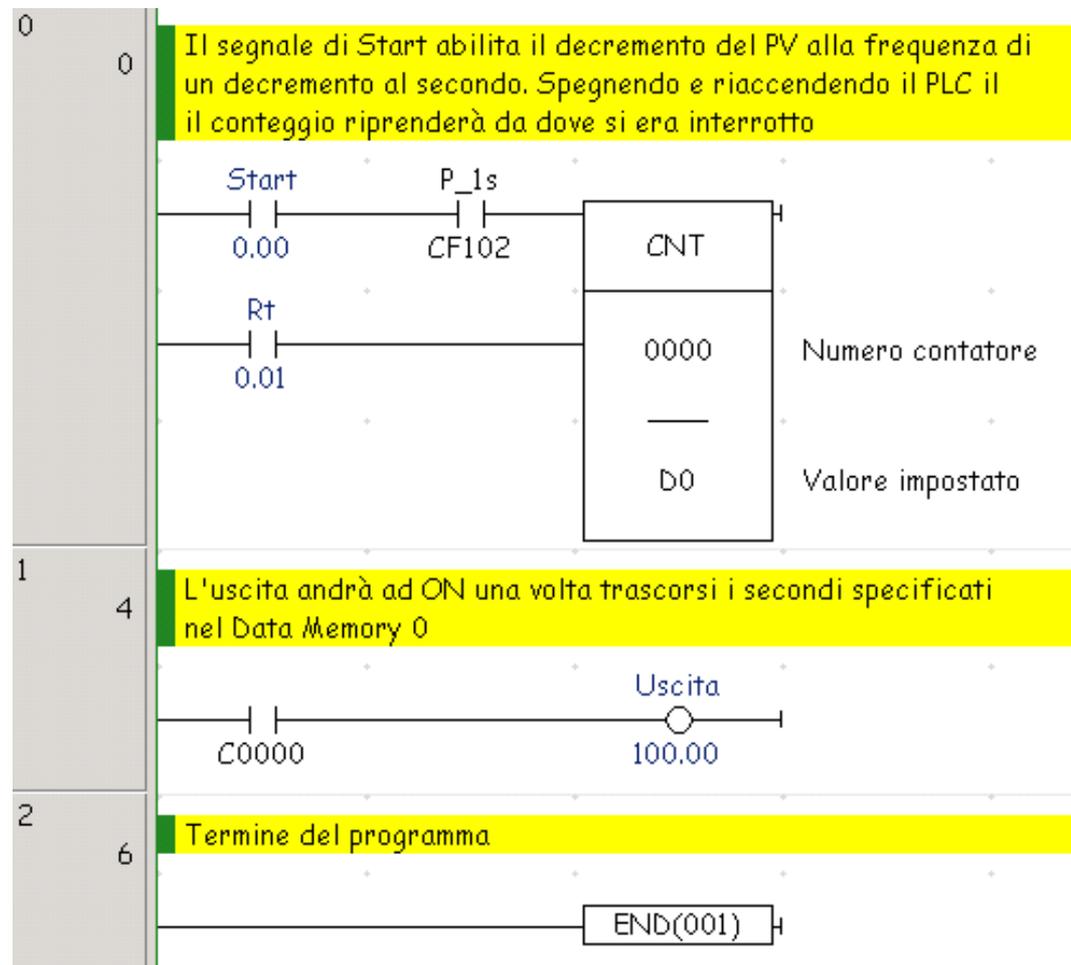
- Le famiglie di PLC CJ/CS e CP1 dispongono di 4.096 contatori (da 0 a 4.095) identificati univocamente dal proprio numero e separati dall'area riservata ai 4.096 temporizzatori
- Nel caso di Timer e Counter è opportuno sottolineare che:
 - Il PV (valore attuale che decrementa) e il bit relativo alla condizione di ON o di OFF del Timer (o Counter) si trovano nell'area dati
 - L'SV o valore di impostazione (valore iniziale di conteggio) si trova nell'area di programma

Esercizio 3: Realizzare un Timer ritentivo

- Utilizzare un contatore abbinato ad un clock di sistema di un secondo (P_1s)
- Nel PLC sono presenti molti flag di sistema (clock a varie frequenze, flag di errore, bit sempre ad ON o ad OFF, ecc)
- L'elenco completo è disponibile nella maschera di inserimento di un nuovo contatto, cliccando sulla freccia di estensione della tendina a scorrimento



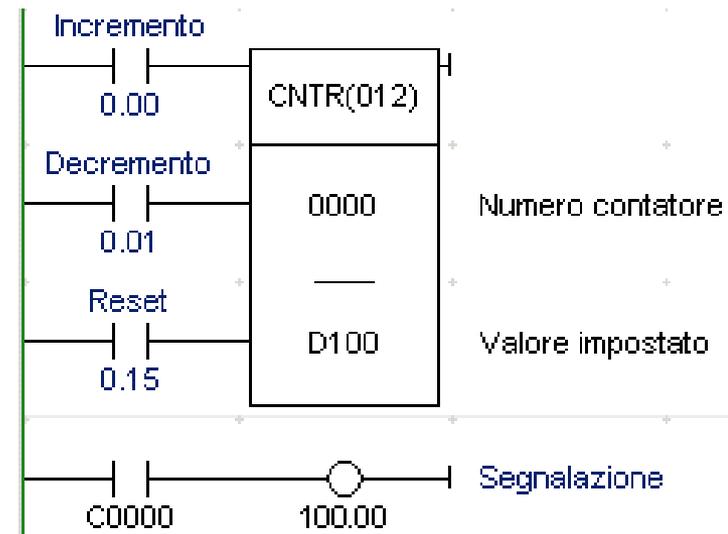
Esercizio 3: Soluzione



La funzione CNTR

- Permette di realizzare un contatore reversibile 'ad anello'
- Questo tipo di contatore necessita di tre segnali di ingresso:
 - Ingresso di incremento
 - Ingresso di decremento
 - Ingresso di reset

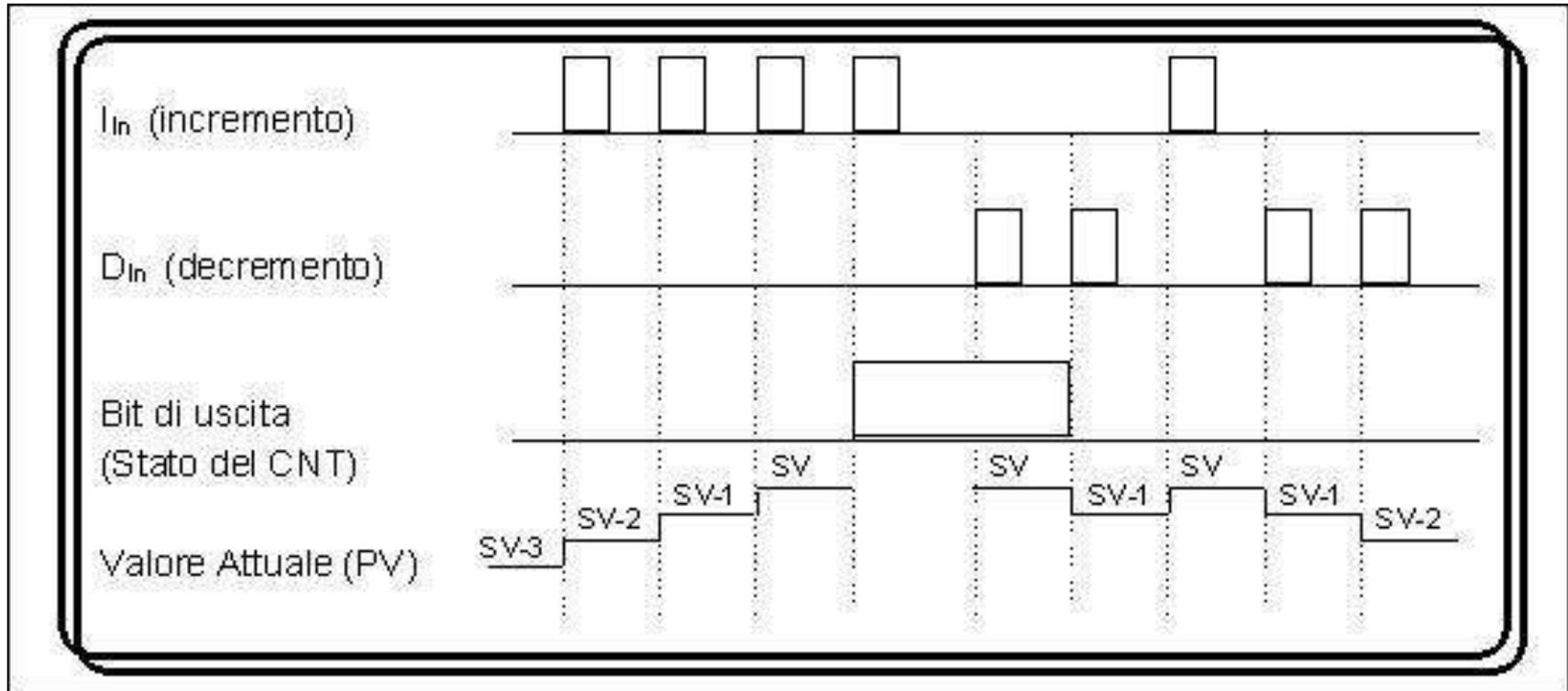
1..2..3..4..5..4..3..2..1..0



La funzione CNTR

- In fase di conteggio addizionale, quando il valore attuale di conteggio (PV) ha raggiunto il valore di preselezione (SV) al successivo impulso il contatore va ad ON ed il PV diventa uguale a zero
- In fase di conteggio sottraente, quando il valore attuale di conteggio (PV) è uguale a zero, al successivo impulso di decremento il contatore va ad ON e il PV diventa uguale all'SV
- Se l'ingresso di Reset è ad ON il PV diventa uguale a zero, lo stato del contatore va ad OFF e gli impulsi di ingresso vengono ignorati

La funzione CNTR

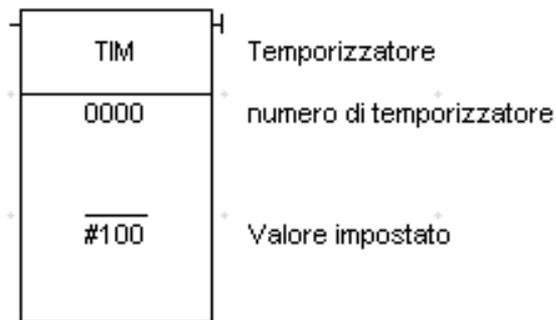


Timer e Counter con impostazione Esadecimale

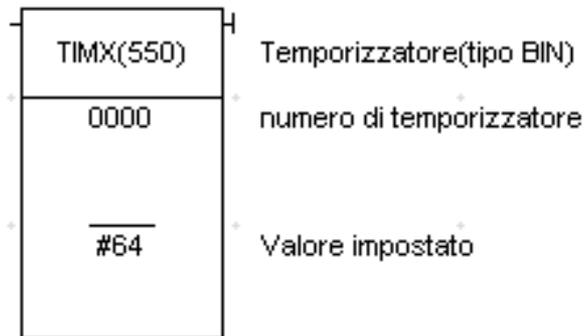
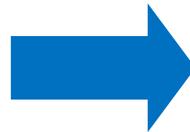
- Nei PLC serie CJ/CS e CP1 è possibile impostare timer e counter con valori binari esadecimali
- Questa possibilità estende il campo di conteggio/temporizzazione delle istruzioni timer counter abituali
- Si passa da un range di impostazione da 0 a 9999 per il BCD a un range da 0 a 65535 per la modalità binaria
- Nei PLC precedenti l'impostazione poteva avvenire solo in BCD

Timer e Counter con impostazione Esadecimale

- Le due modalità di impostazione BCD e Binaria esadecimale non possono coesistere all'interno dello stesso programma; esistono perciò delle istruzioni Timer e Counter dedicate alla modalità esadecimale
- Solo nel caso dei PLC della serie CJ2 e CP1E le due modalità possono convivere



Modalità BCD 10 sec.
SV=#100



Modalità Hex 10 sec.
SV=#64
Oppure SV=&100

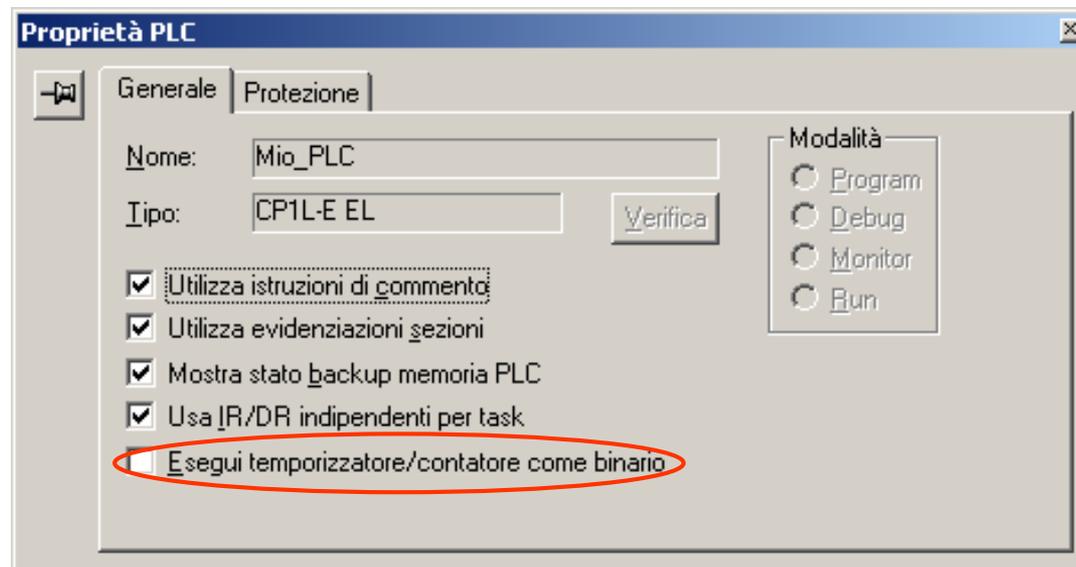
Timer e Counter con impostazione Esadecimale

- Durante la stesura del progetto è possibile definire se la modalità di lavoro dei temporizzatori e dei contatori debba essere BCD o Esadecimale (binaria)
- CX-Programmer compilerà correttamente solo il set di istruzioni per temporizzatori e contatori relativo alla modalità specificata

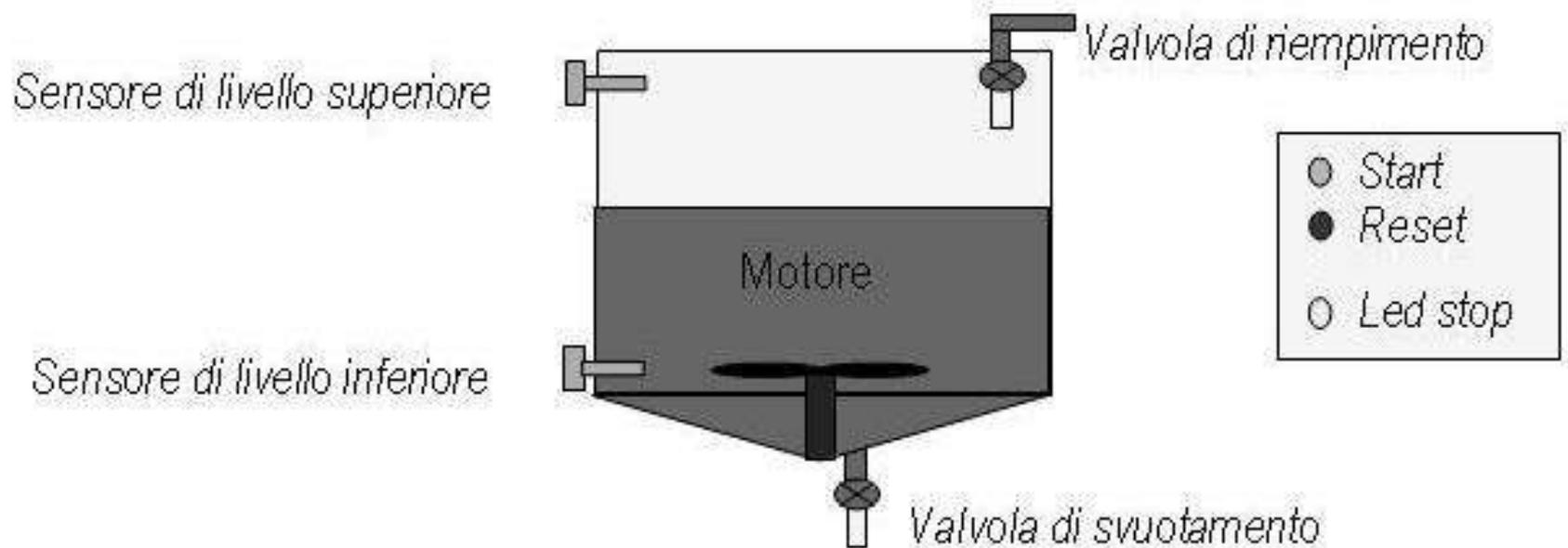
Nome	Mnemonico	
	Modalità BCD	Modalità Binaria
TIMER (100 ms)	TIM	TIMX(550)
HIGH-SPEED TIMER (10 ms)	TIMH(015)	TIMHX(551)
ONE-MS TIMER (1 ms)	TMHH(540)	TMHHX(552)
ACCUMULATIVE TIMER (100 ms)	TTIM(087)	TTIMX(555)
LONG TIMER (100 ms)	TIML(542)	TIMLX(553)
MULTI-OUTPUT TIMER (100 ms)	MTIM(543)	MTIMX(554)
COUNTER	CNT	CNTX(546)
REVERSIBLE COUNTER	CNTR(012)	CNTRX(548)
RESET TIMER/ COUNTER	CNR(545)	CNRX(547)

Timer e Counter con impostazione Esadecimale

- Per visualizzare la finestra di impostazione della modalità di lavoro dei timer e counter è necessario cliccare con il tasto destro sull'icona del PLC nell'area di progetto (🖱️ Mio_PLC[CP1L-E]) e selezionare la voce 'Proprietà'



Esercizio 4: Ciclo di riempimento-svuotamento

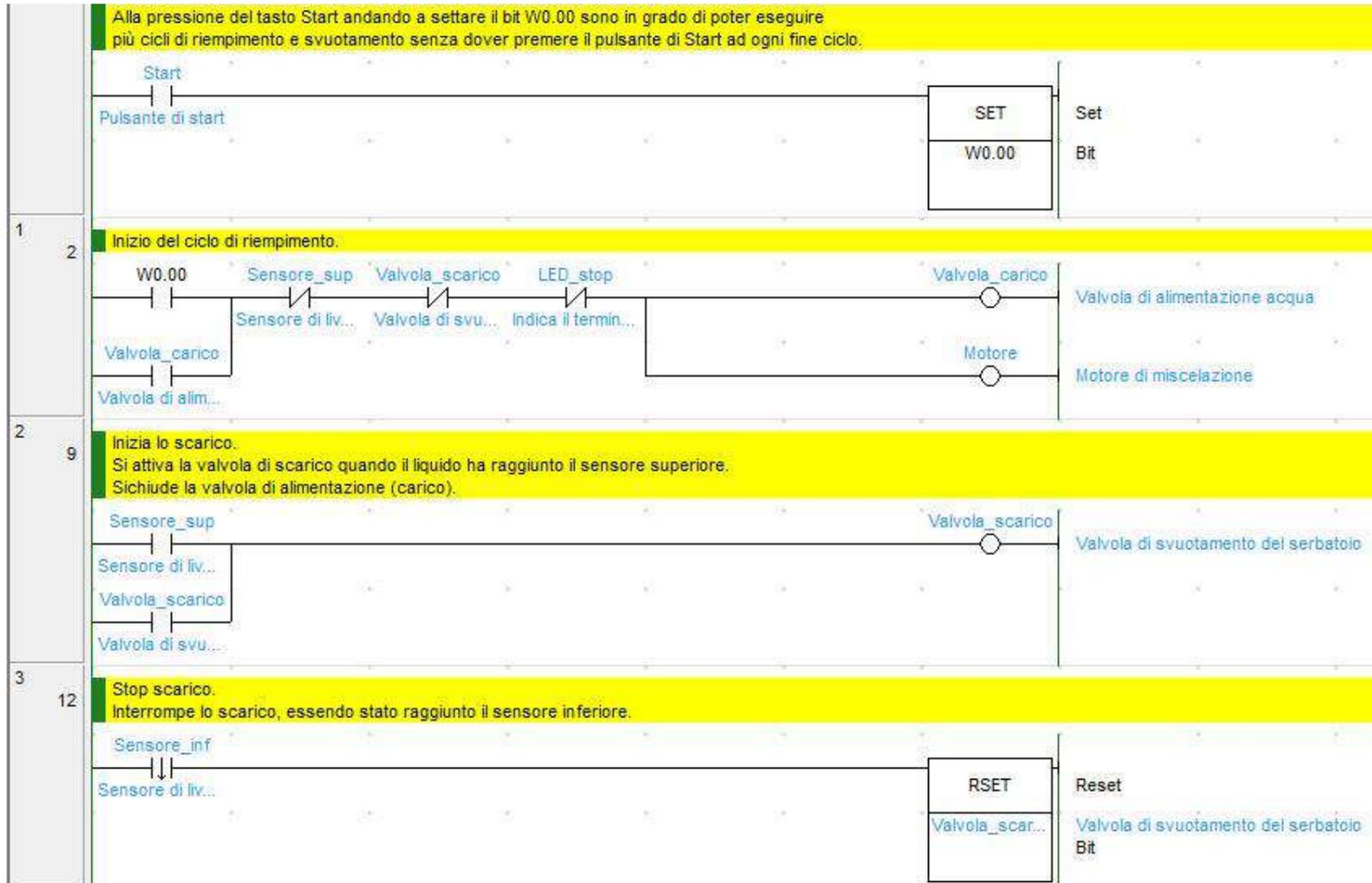


^ LED_stop	BOOL	100.03	Lavoro	Indica il termine del ciclo
^ Motore	BOOL	100.02	Lavoro	Motore di miscelazione
^ Reset	BOOL	0.01	Lavoro	Comando di reset del ciclo
^ Sensore_inf	BOOL	0.03	Lavoro	Sensore di livello inferiore
^ Sensore_sup	BOOL	0.02	Lavoro	Sensore di livello superiore
^ Start	BOOL	0.00	Lavoro	Pulsante di start
^ Valvola_carico	BOOL	100.00	Lavoro	Valvola di alimentazione acqua
^ Valvola_scarico	BOOL	100.01	Lavoro	Valvola di svuotamento del serbatoio

Esercizio 4: Ciclo riempimento-svuotamento/1

- Alla pressione del pulsante di Start (0.00), si apre la valvola di riempimento (100.00) e si mette in moto il motore di miscelazione (100.02).
- Quando il livello raggiunge il sensore superiore (0.02), la valvola di riempimento si chiude, il motore si ferma e si apre la valvola di svuotamento (100.01).
- Quando il livello raggiunge il sensore inferiore (0.03) la valvola di svuotamento si richiude e può iniziare un altro riempimento.

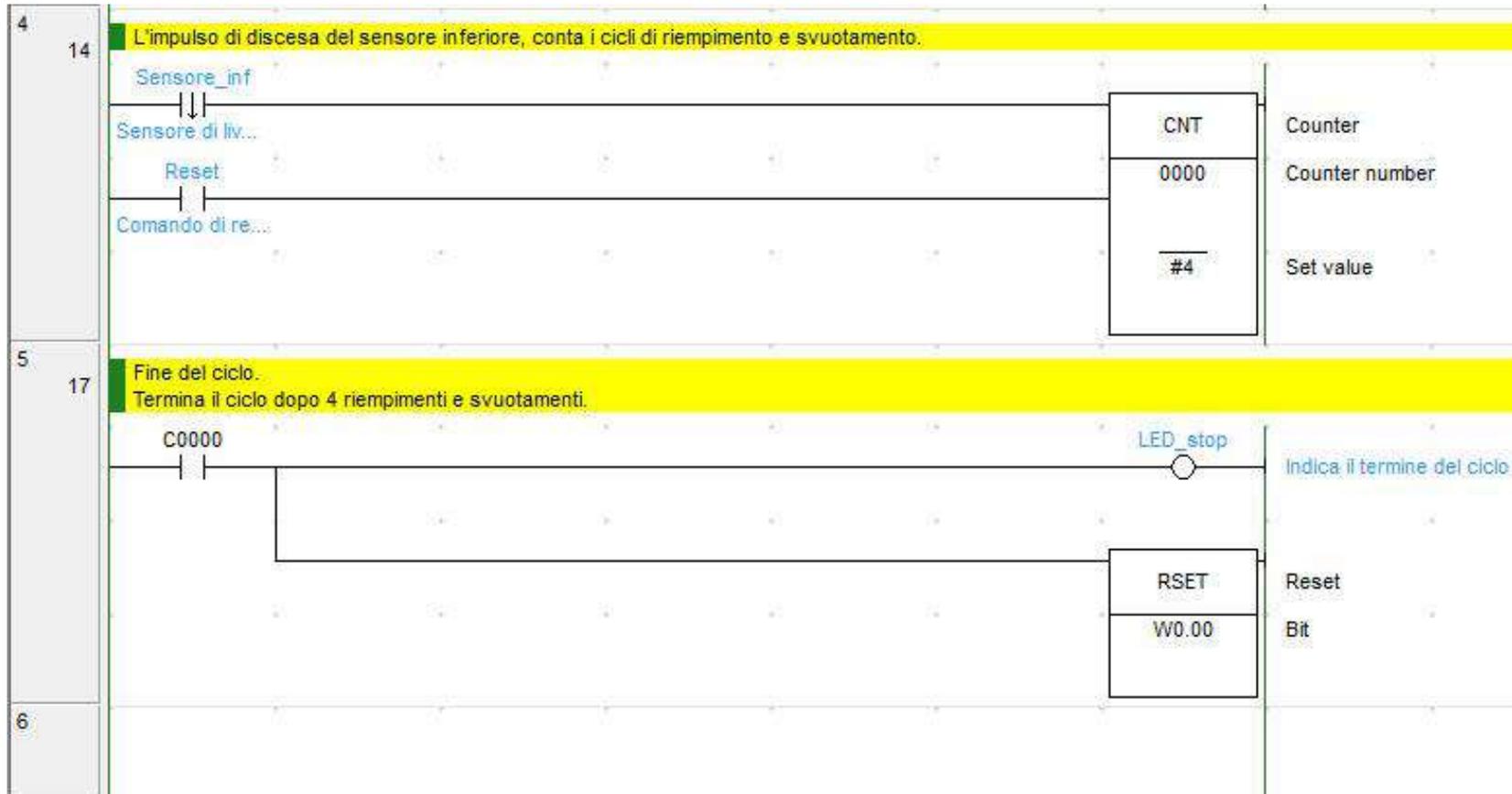
Esercizio 4: Soluzione



Esercizio 4: Ciclo riempimento-svuotamento/2

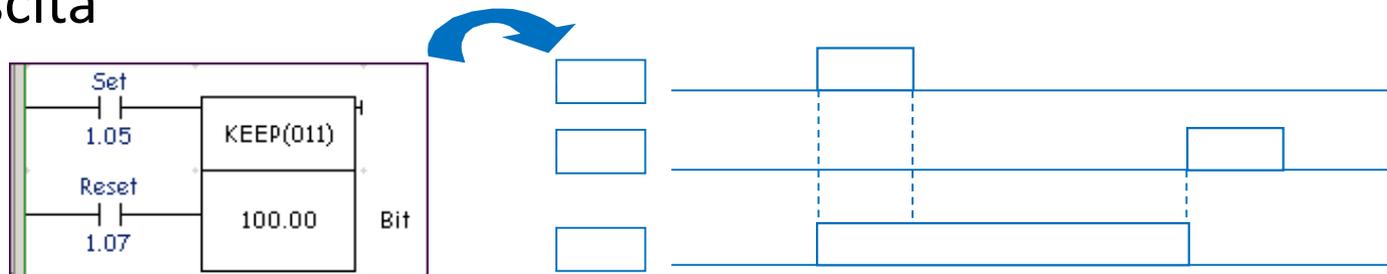
- Quando il ciclo è stato ripetuto quattro volte (con una sola pressione di Start), il led di stop (100.03) si accende e un ulteriore ciclo non può essere iniziato se non dopo aver premuto il pulsante di Reset (0.01)

Esercizio 4: Soluzione



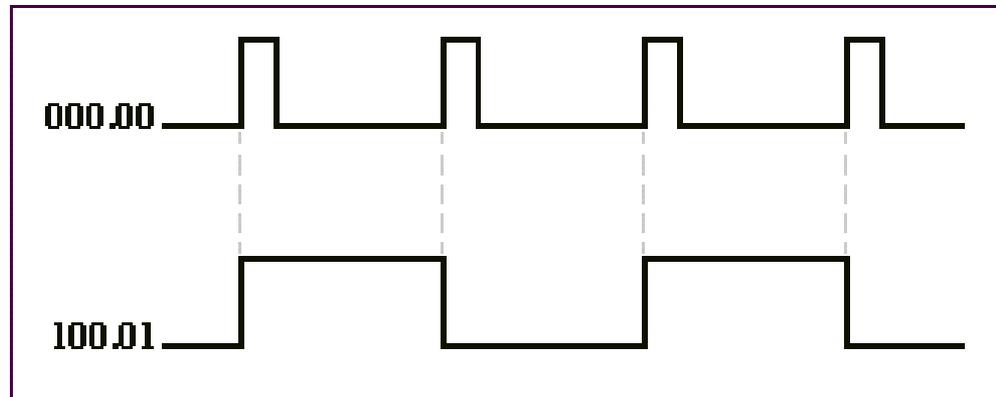
La funzione KEEP

- L'istruzione KEEP permette di realizzare un relè a ritenuta
- Si hanno due ingressi:
 - L'ingresso di Set
 - L'ingresso di Reset
- Se l'ingresso di Reset è ad OFF un impulso su S attiva in modo permanente il bit programmato come operando della funzione
- Un impulso di Reset determina la disattivazione del bit di uscita

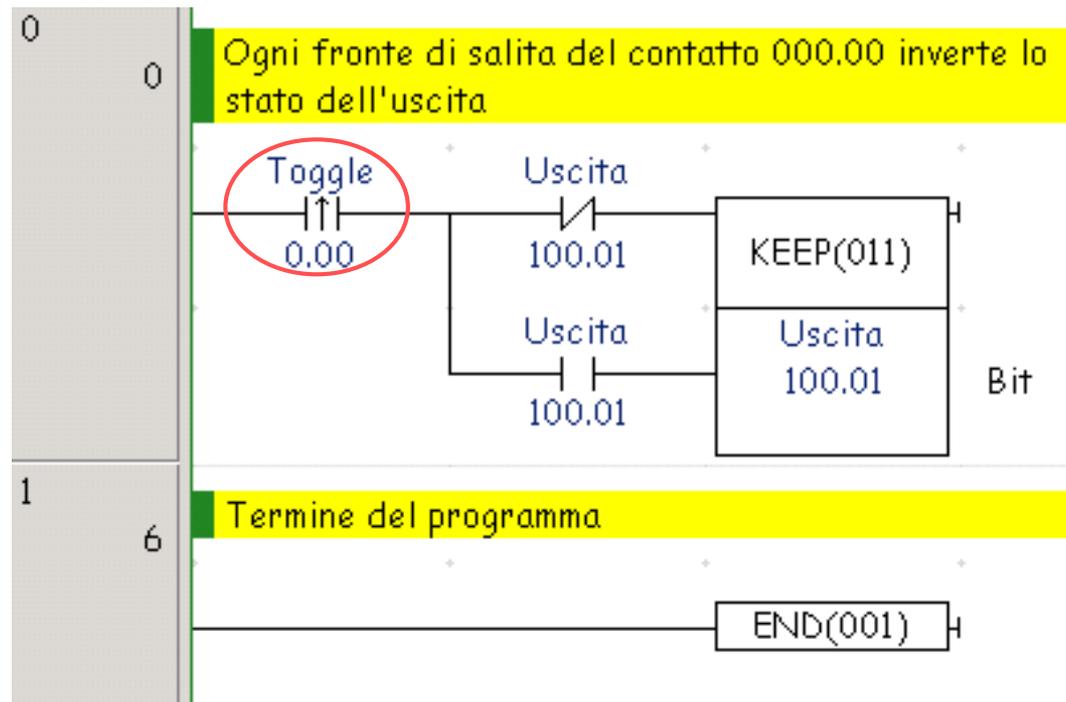


Esercizio 5: Toggle

- Fare in modo che l'uscita 100.01 del PLC si attivi e disattivi alternativamente ad ogni fronte di salita del segnale di ingresso 000.00



Esercizio 5: Soluzione



Finestra di controllo

- Per richiamare la finestra di controllo, selezionare l'icona 
- Oppure selezionare "Visualizza" dal menu principale, e poi scegliere la voce "Controlla" del menu "Finestre"
- Oppure premere insieme i tasti "ALT" e "3"

La finestra di controllo

Nome PLC	Nome	Indirizzo	Tipo / Formato dati	Utiliz...	Valore	Valore(binario)
PLC1	NuovoProgramma1.Start	0.00	BOOL (On/Off,Contatto)		0	
PLC1	NuovoProgramma1.Stop	0.01	BOOL (On/Off,Contatto)		1	
PLC1	NuovoProgramma1.Valvola	10.01	BOOL (On/Off,Contatto)		0	
PLC1		D0	CHANNEL (Esa,Canale)		04D2 Esa	0000 0100 1101 0010
PLC1		D100	CHANNEL (Esa,Canale)		1234 Esa	0001 0010 0011 0100

sheet1 / sheet2 / sheet3

Nome del PLC

Foglio attivo

Nome della variabile

Indirizzo PLC

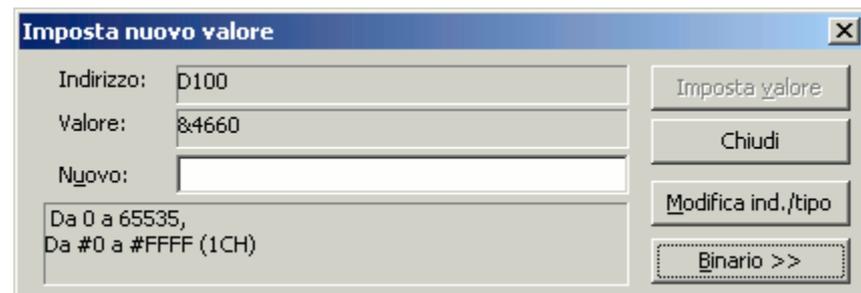
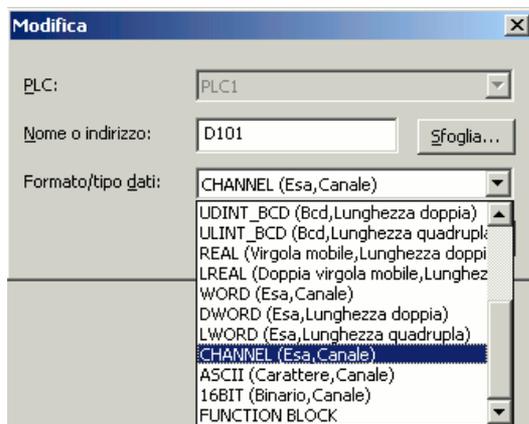
Tipo della variabile e
formato di visualizzazione

Rappresentazione binaria del valore

Valore

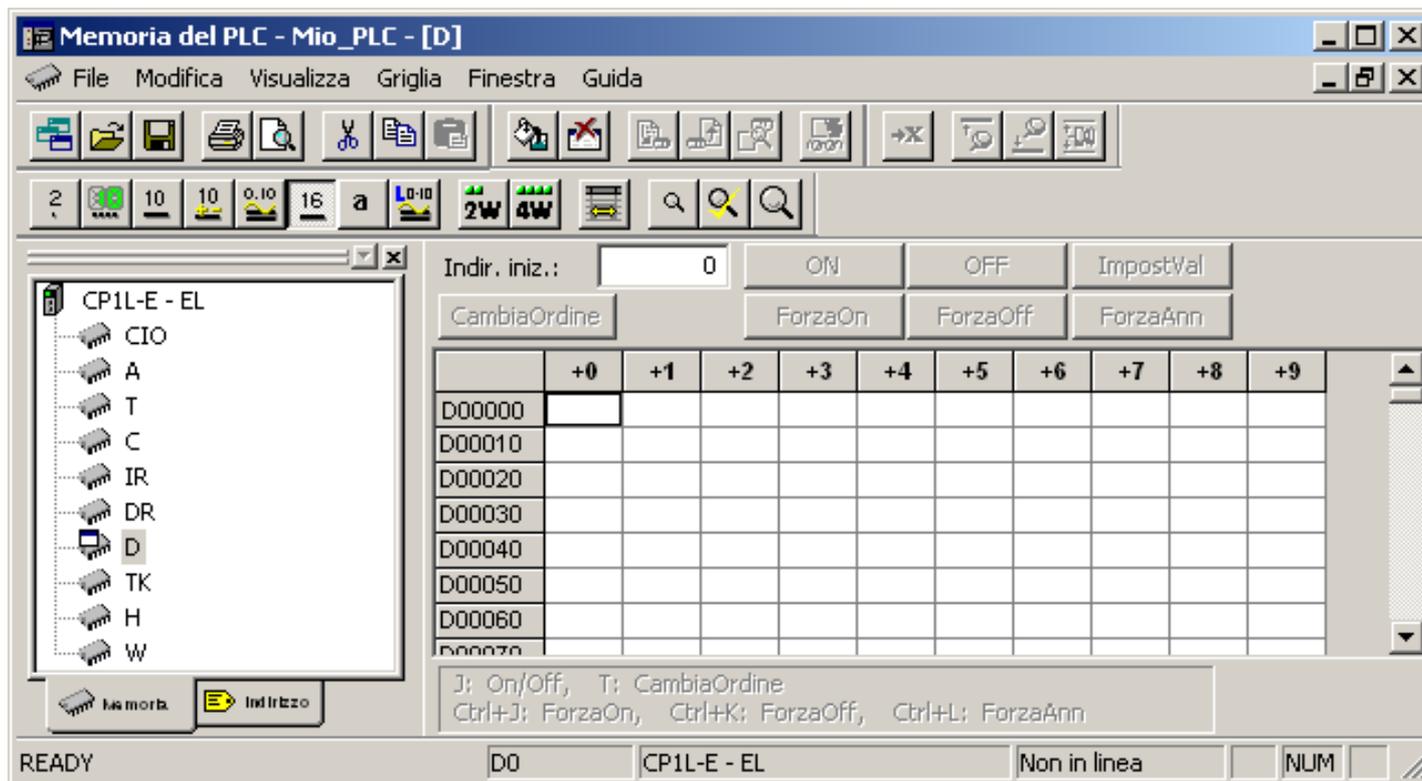
La finestra di controllo

- Per inserire un indirizzo da monitorare, premere “Ins” oppure selezionare una riga vuota e fare doppio click o premere “invio”; verrà mostrata la finestra per l’immissione di un nuovo elemento
- Per modificare un valore online, selezionare la riga desiderata e fare doppio click o premere “invio”; verrà mostrata una finestra per l’immissione del nuovo valore



Editor di memoria

- Per modificare/monitorare una maggior quantità di dati si può ricorrere all'editor di memoria.



Editor di memoria

Operazioni Online

The screenshot shows the 'Memoria del PLC - Mio_PLC' window. The interface includes a menu bar (File, Modifica, Visualizza, Griglia, In linea, Finestra, Guida), a toolbar with various icons, and a main workspace. The workspace is divided into a tree view on the left and a data grid on the right. The tree view shows a hierarchy for 'CP1L-E - EL' with sub-items: CIO, A, T, C, IR, DR, D, TK, H, and W. The data grid has columns labeled '+0' through '+9' and rows labeled from CIO0000 to CIO0170. The 'Indir. iniz.' field is set to '0'. The status bar at the bottom shows 'READY', 'CIO0', 'CP1L-E - EL', 'Program', and 'NUM'.

Seleziona del formato (indicated by an arrow pointing to the toolbar icons)

Seleziona dell'indirizzo (indicated by an arrow pointing to the 'Indir. iniz.' field)

Visualizzazione del contenuto dell'area dati (indicated by an arrow pointing to the data grid)

Seleziona dell'area dati (indicated by an arrow pointing to the tree view)

L'informazione nell'area di memoria



CIO																	
Indir. iniz.:	0			ON			OFF			ImpostVal							
CambiaOrdine	ForzaOn			ForzaOff			ForzaAnn										
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Hex
CIO0000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000
CIO0001	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000
CIO0002	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000
CIO0003	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000
CIO0004	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000
CIO0005	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000

J: On/Off, T: CambiaOrdine
 Ctrl+J: ForzaOn, Ctrl+K: ForzaOff, Ctrl+L: ForzaAnn

- Dall'apposita barra si può cambiare la base di codifica della visualizzazione degli indirizzi di memoria
- Se si preme il tasto  della barra si può visualizzare lo stato di ogni singolo bit di ogni canale

L'informazione nell'area di memoria



- Gli altri tasti della barra permettono la visualizzazione in:



– Codifica BCD



– Codifica decimale



– Codifica decimale con segno



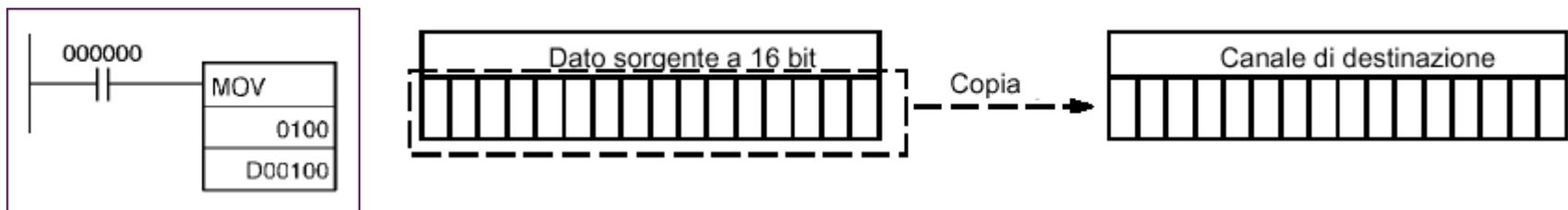
– Notazione Floating Point (virgola mobile)



– Codifica esadecimale

La funzione MOV

- L'istruzione MOV realizza la copia di un dato a 16 bit da un canale sorgente S ad un canale destinazione D
- Le aree utilizzabili per il trasferimento sono:
 - CIO, W, HR, A, D, T, C
- Se S=0 il flag di uguale P_EQ va ad ON subito dopo l'esecuzione dell'istruzione

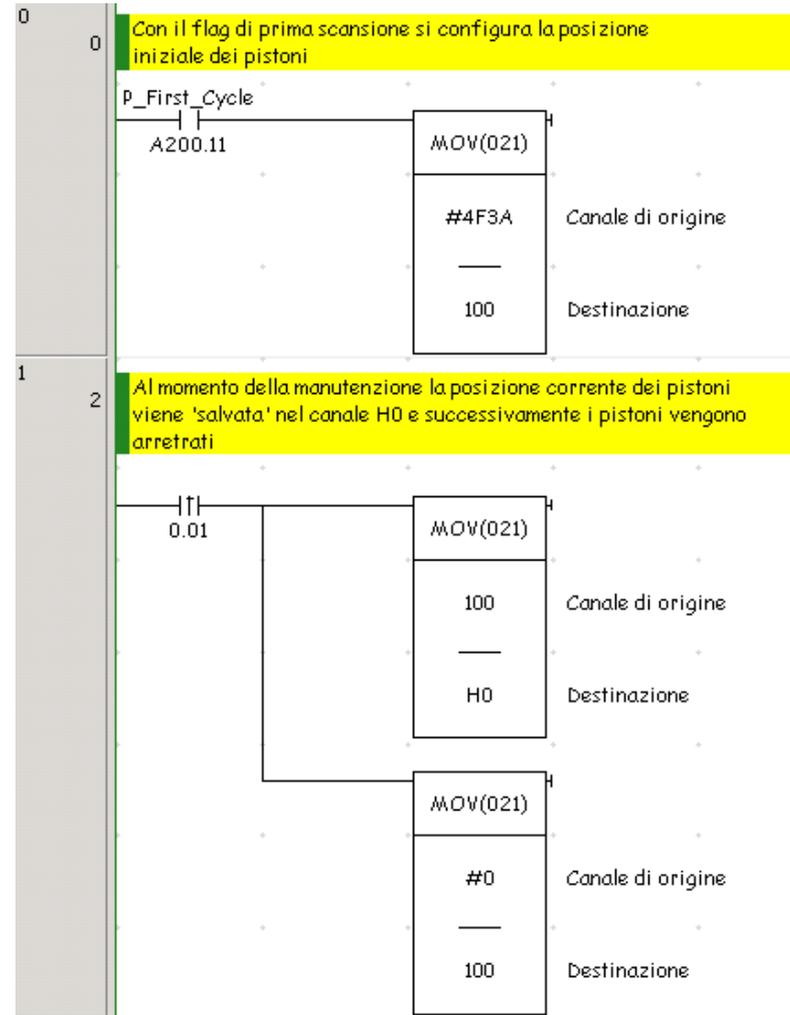


- Esiste anche la funzione MVN che effettua uno spostamento dati NEGANDO però il dato stesso

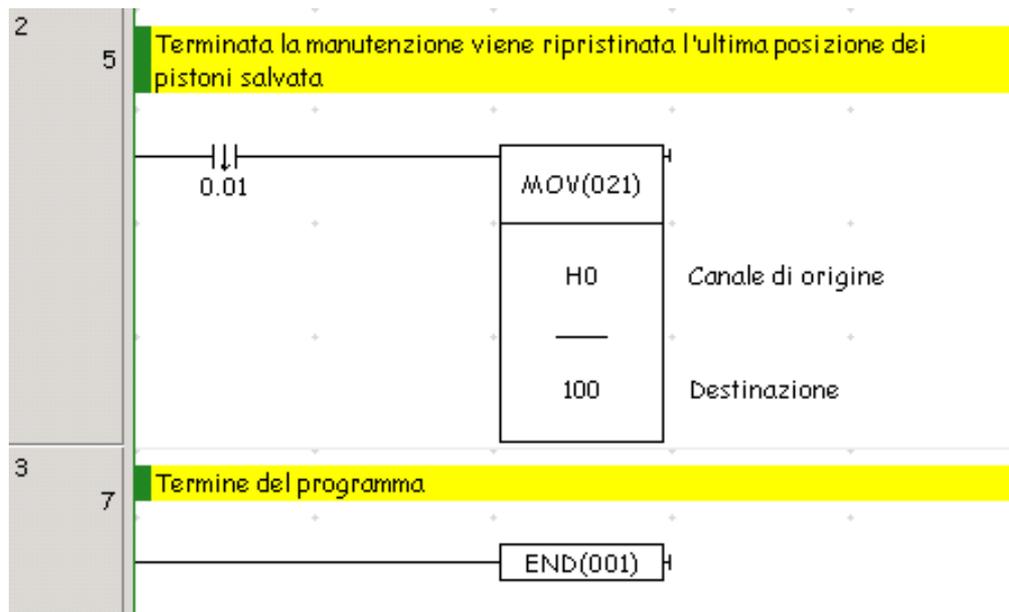
Esercizio 6: Zero macchina

- Fare in modo che all'accensione il canale 100 di uscita del PLC abbia il seguente aspetto: **0 1 0 0 1 1 1 1 0 0 1 1 1 0 1 0** sapendo che ad ogni bit di uscita corrisponde la posizione avanzata (1) o arretrata (0) di un pistone
- Per poter mantenere la macchina si deve dare la possibilità, chiudendo ad esempio il contatto 000.01 di portare tutti i pistoni in posizione arretrata
- E' necessario memorizzare la posizione dei pistoni prima della manutenzione (utilizzare il canale H0 come 'memoria della posizione') per poterla ripristinare al termine della manutenzione (contatto 000.01 aperto)

Esercizio 6: Soluzione



Esercizio 6: Soluzione

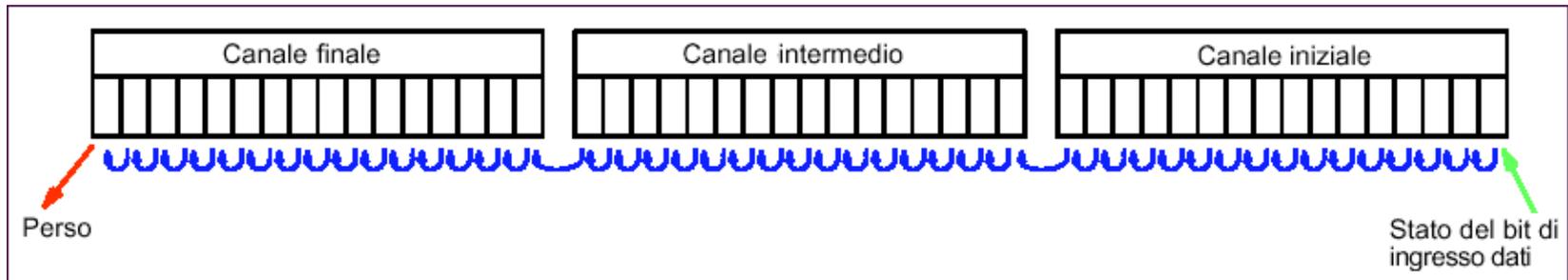
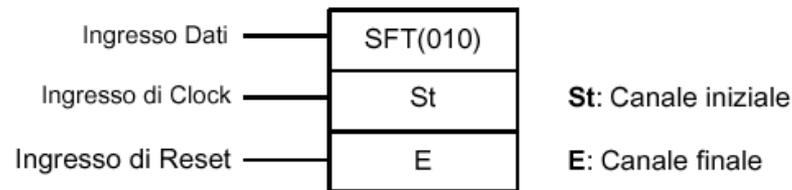


La funzione SFT

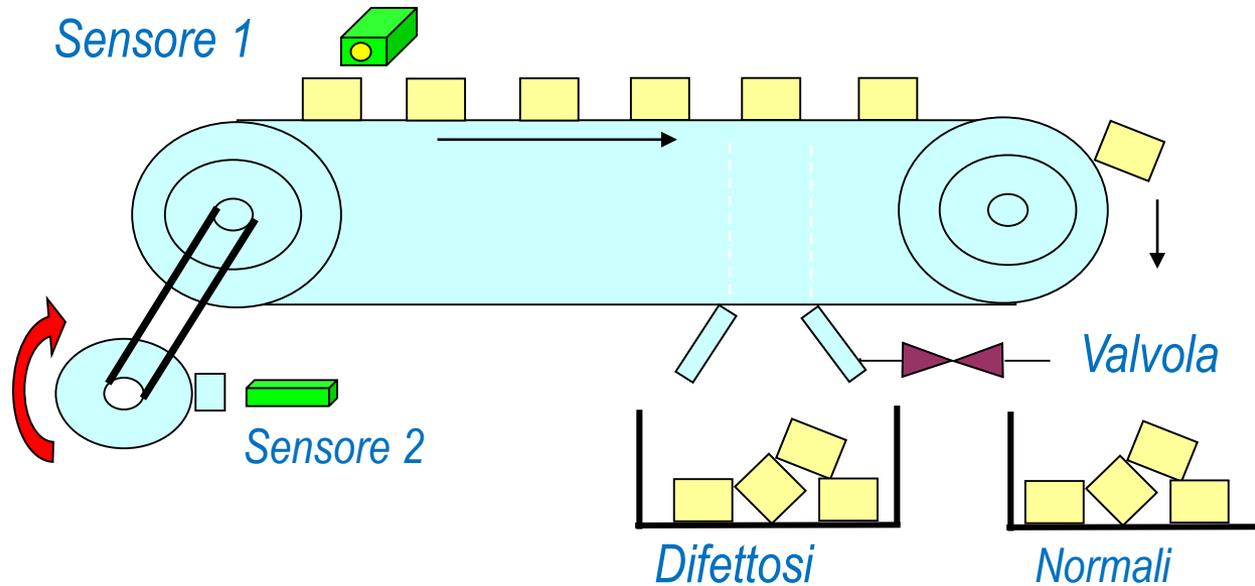
- L'istruzione SFT realizza la funzione di registro a scorrimento con un ingresso seriale
- Lo scorrimento avviene a bit, partendo dal meno significativo al più significativo
- La funzione SFT richiede tre ingressi:
 - L'ingresso Dati: lo stato di questo ingresso viene trasferito nel registro a scorrimento sul fronte di salita dell'ingresso di clock
 - L'ingresso di Clock: il fronte di salita di questo ingresso determina lo scorrimento dei dati e il trasferimento dell'ingresso dati
 - L'ingresso di Reset: determina il reset del canale (o dei canali) su cui avviene lo scorrimento
- E' necessario specificare come operandi della funzione il primo e l'ultimo canale del registro

La funzione SFT

- Il primo e l'ultimo canale del registro devono appartenere alla stessa area
- Le aree utilizzabili sono: HR, W e A (da A448 ad A959)
- Se si utilizzano canali di scorrimento appartenenti all'area HR (ritentiva) i dati verranno mantenuti in caso di spegnimento



Esercizio 7: Controllo di qualità

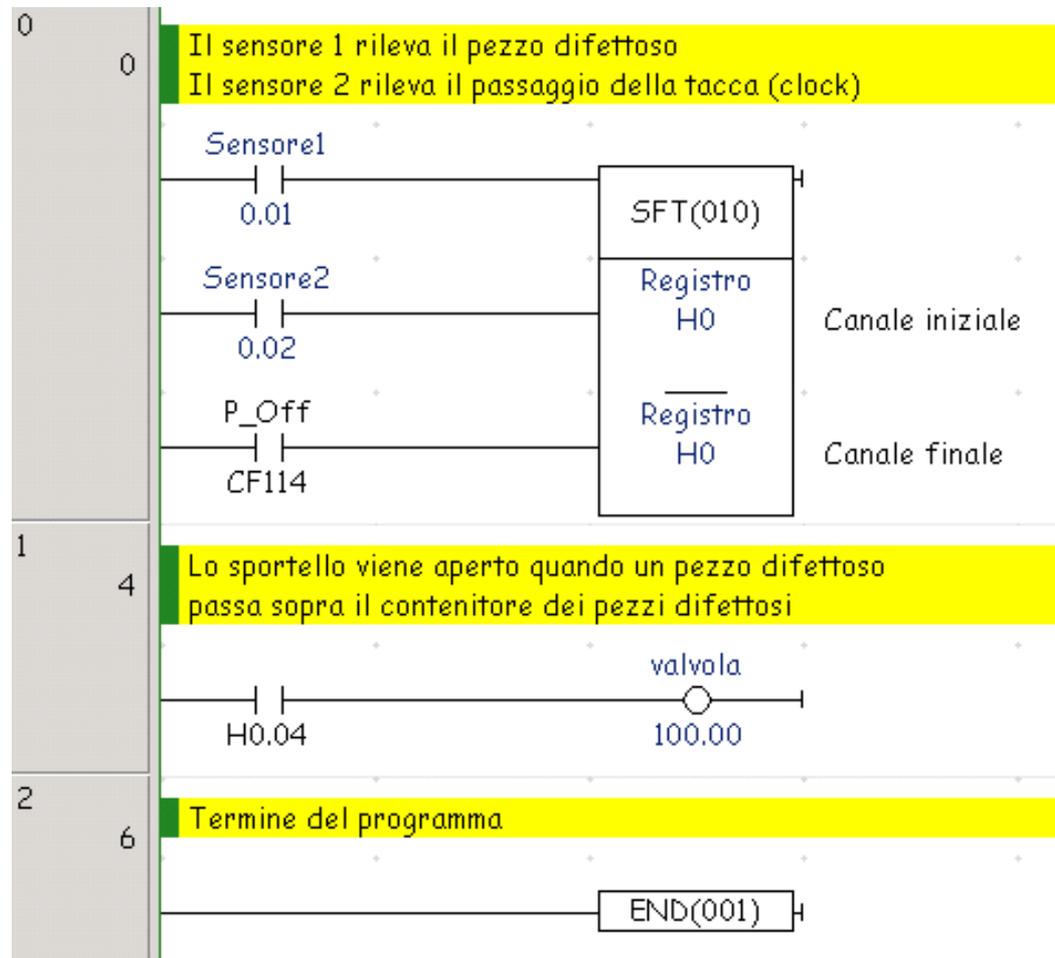


Registro	CHANNEL	H0	Lavoro	Registro a scorrimento
^ Sensore1	BOOL	0.01	Lavoro	rileva un pezzo difettoso
^ Sensore2	BOOL	0.02	Lavoro	Rileva tacca
^ valvola	BOOL	100.00	Lavoro	apre lo sportello

Esercizio 7: Controllo di qualità

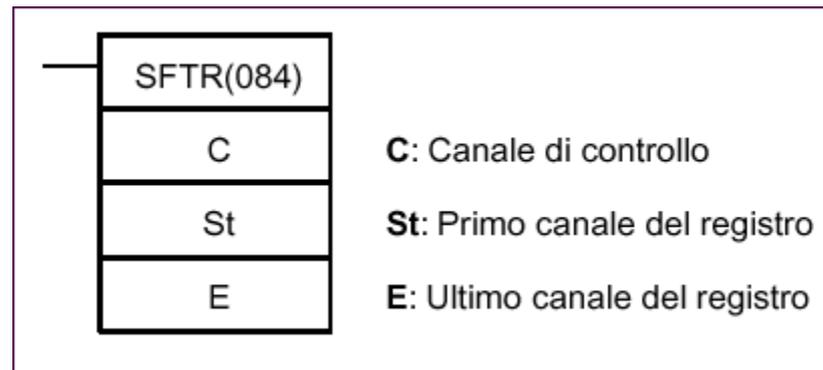
- Il sensore 1 (000.01) rileva i pezzi difettosi in transito su un nastro trasportatore, mentre il sensore 2 (000.02) rileva il passaggio di una tacca che misura lo scorrere dei pezzi (posti ad eguale distanza)
- Quando un pezzo difettoso passa sopra lo sportello (posto ad una distanza corrispondente a 4 pezzi dal sensore 1) una valvola (100.00) ne permette l'apertura ed il pezzo viene scartato
- Suggerimento: si utilizzi il canale H0 come registro a scorrimento per 'mappare' la posizione dei pezzi difettosi sul nastro trasportatore

Esercizio 7: Soluzione



La funzione SFTR

- L'istruzione SFTR permette di realizzare la funzione di registro a scorrimento in modo *bidirezionale*
- Lo scorrimento può avvenire su uno o più canali (appartenenti alla stessa area dati)
- Nella funzione devono essere specificati tre parametri:



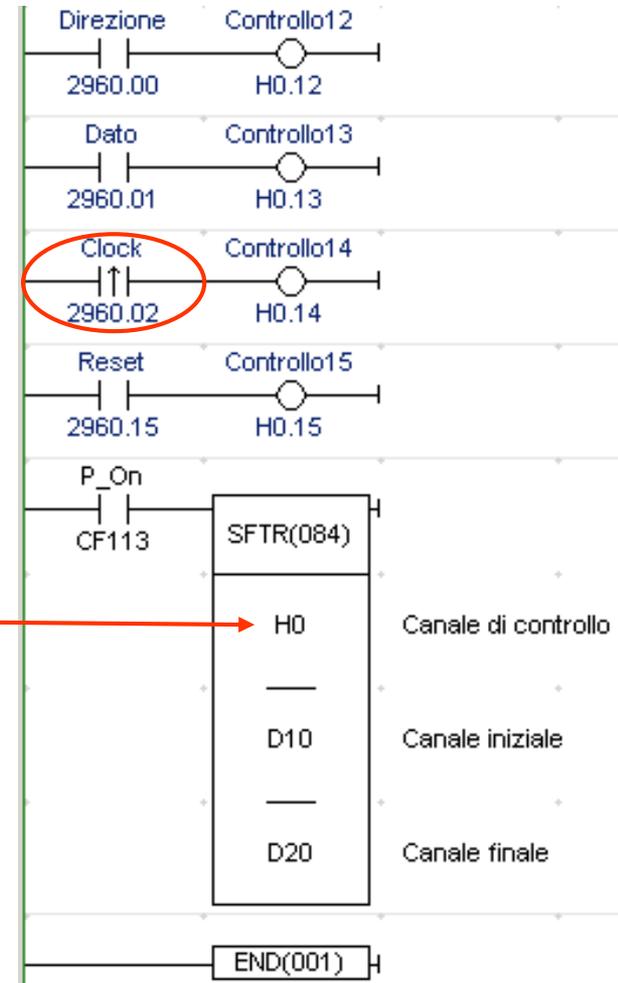
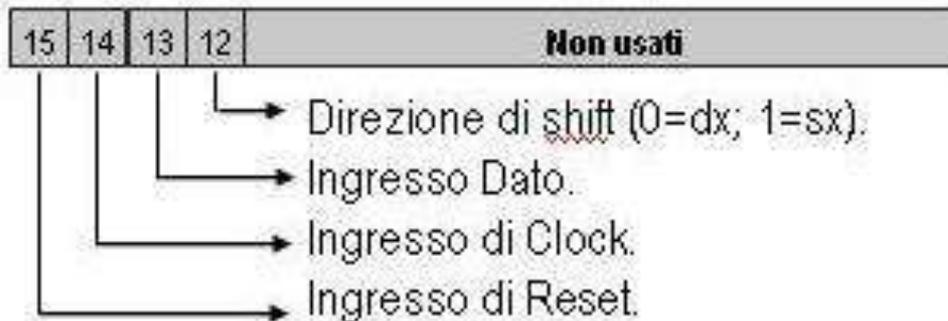
La funzione SFTR

- Lo scorrimento dei dati avviene nelle seguenti condizioni:
 - L'istruzione di SFTR è abilitata
 - Il bit 14 del canale di controllo (clock) fornisce un fronte di salita
- Il flag P_CY (Carry) riceve lo stato del bit 'in uscita' dal



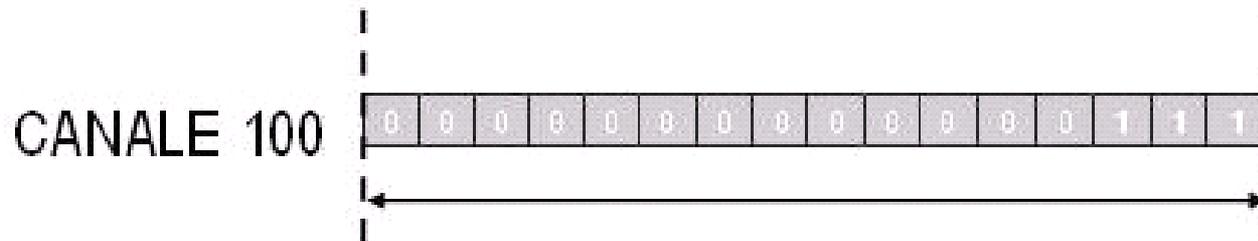
La funzione SFTR

- I vari ingressi (Direzione, Dato, Clock e Reset) vengono gestiti tramite un canale di controllo
- I quattro bit più significativi del canale definito come canale di controllo assumono il seguente significato:



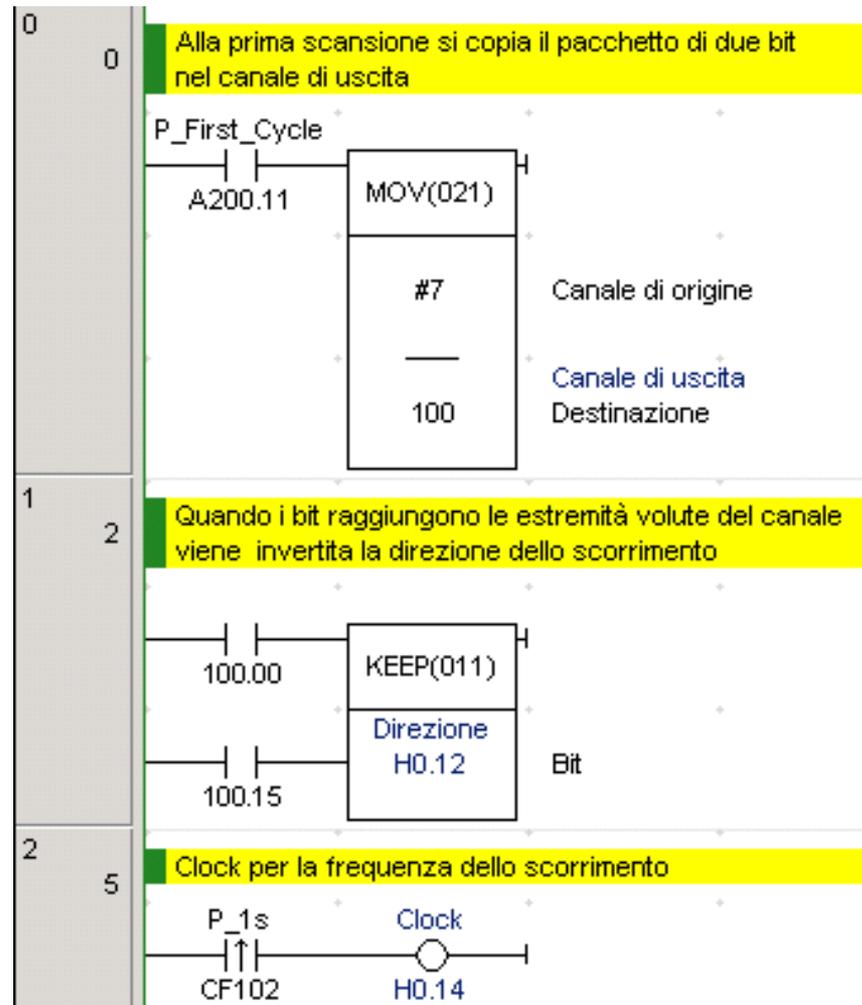
Esercizio 8: Scorrimento Up/Down

- Si realizzi un programma che faccia scorrere in modo up/down un pacchetto di tre bit nel canale di uscita (100) con il ritmo di uno shift al secondo

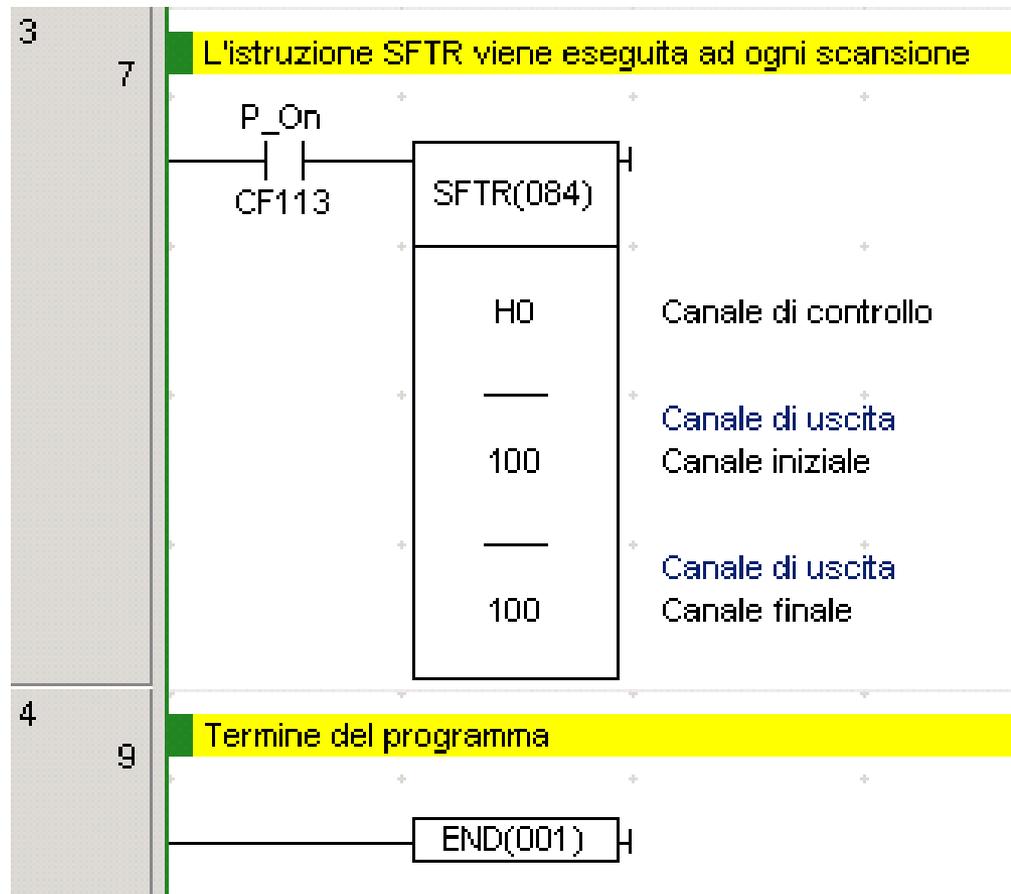


- Utilizzare le istruzioni SFTR(084), MOV(021), KEEP(011) e i flag di sistema P_First_Cycle, P_On e P_1s

Esercizio 8: Soluzione



Esercizio 8: Soluzione

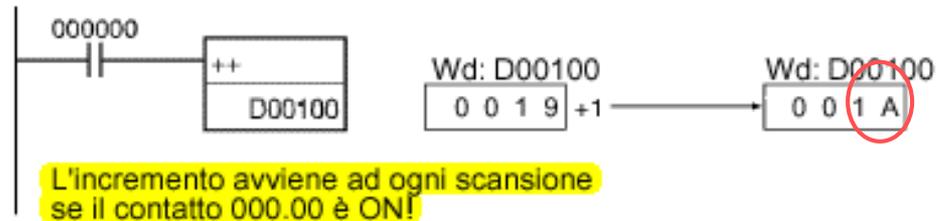


Istruzioni di Incremento e Decremento

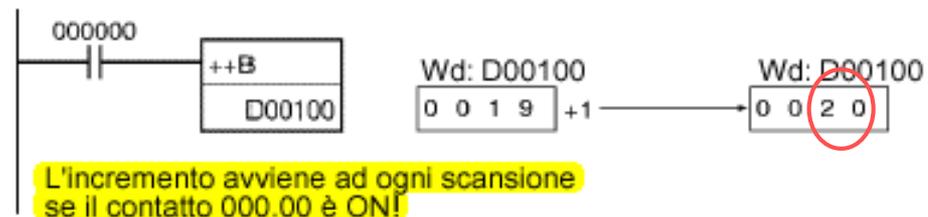
- Istruzioni di incremento e decremento in esadecimale:
 - ++** : Incrementa in Esadecimale il contenuto di una word
 - : Decrementa in Esadecimale il contenuto di una word
 - ++L** : Incrementa in Esadecimale il contenuto di due word
 - L** : Decrementa in Esadecimale il contenuto di due word
- Istruzioni di incremento e decremento in BCD:
 - ++B** : Incrementa in BCD il contenuto di una word
 - B** : Decrementa in BCD il contenuto di una word
 - ++BL** : Incrementa in BCD il contenuto di due word
 - BL** : Decrementa in BCD il contenuto di due word

Istruzioni di Incremento e Decremento

- Esempio Incremento Binario (++):



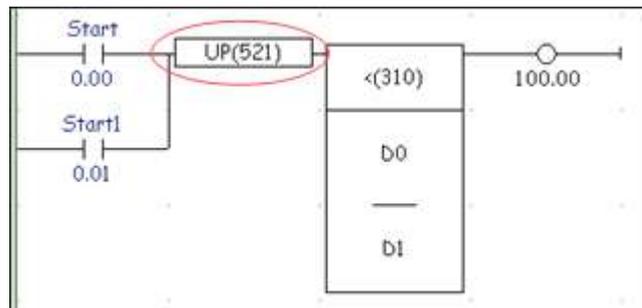
- Esempio Incremento BCD (++B):



Le istruzioni intermedie

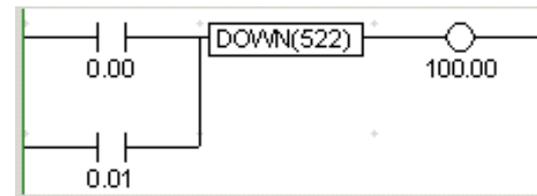
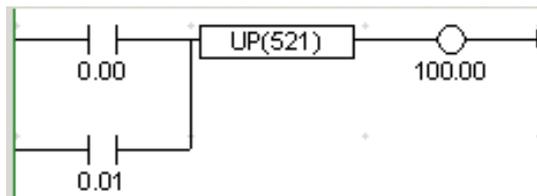
- Come già sottolineato, le nuove famiglie di PLC CJ/CS e CP1 supportano un insieme di nuove istruzioni
- Tra queste, sono state introdotte anche alcune funzioni cosiddette ‘intermedie’ che non si trovano al termine della Rung come tutte le altre, bensì all’interno di essa, e richiedono quindi anche un segnale di uscita
- L’utilizzo di queste istruzioni in alcuni casi semplifica notevolmente la stesura del programma

- Esempio:

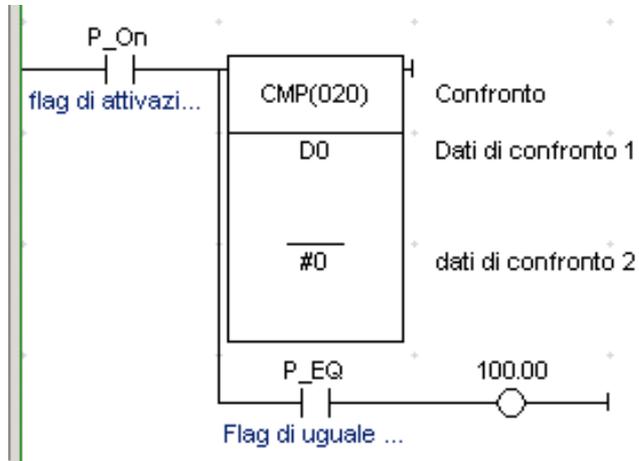


Le istruzioni intermedie

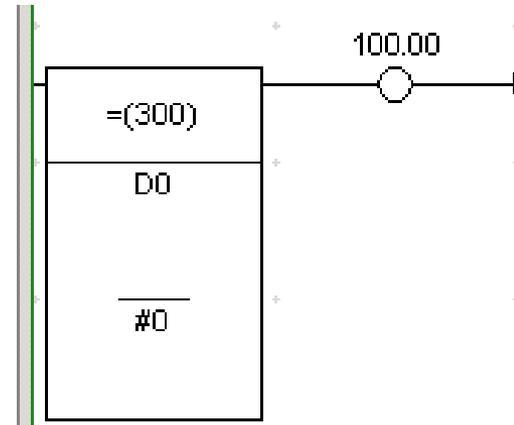
- Istruzioni di ingresso differenziate:
 - Fronte di salita (@)
 - Fronte di discesa (%)
- Istruzioni di refresh immediato:
 - Refresh immediato di un ingresso o di un'uscita (!)
 - Possibile combinazione con la differenziazione
- Istruzioni UP e DOWN
 - Come DIFU e DIFD
 - Sono istruzioni intermedie che non richiedono di un bit di appoggio



Istruzioni di comparazione intermedie



Comparazione standard



Comparazione intermedia

- Sui PLC della serie CJ/CS e CP1 è possibile utilizzare le comparazioni come istruzioni intermedie
- L'istruzione intermedia fornisce direttamente in uscita il risultato della comparazione

Istruzioni di comparazione intermedie

- Istruzioni di comparazione:
 - Sono istruzioni intermedie (=, <>, <, <=, >, >=)
 - Suffisso S (p.es: >S): Confronto di dati binari con segno
 - Suffisso L (p.es: >L): Confronto di dati su doppia word
 - Suffisso F (p.es: >F): Confronto di dati in virgola mobile
- Istruzioni di Set e Reset multiplo:
 - SETA(530) : Imposta ad 1 il numero specificato di bit consecutivi a partire da un determinato bit del canale iniziale stabilito
 - RSTA(531) : Imposta a 0 il numero specificato di bit consecutivi a partire da un determinato bit del canale iniziale stabilito

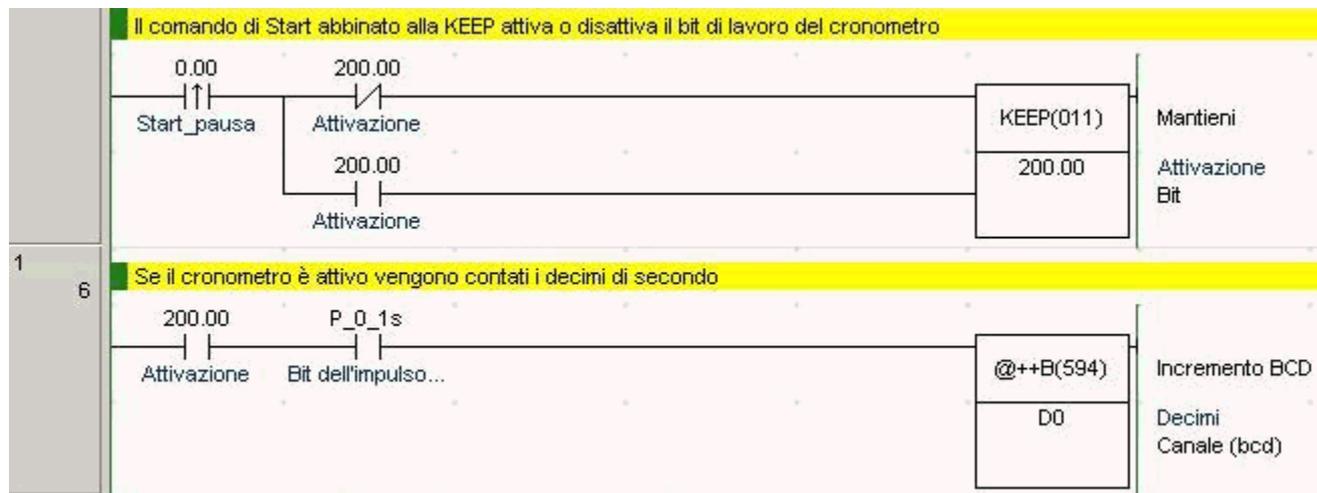
Esercizio 9: Cronometro



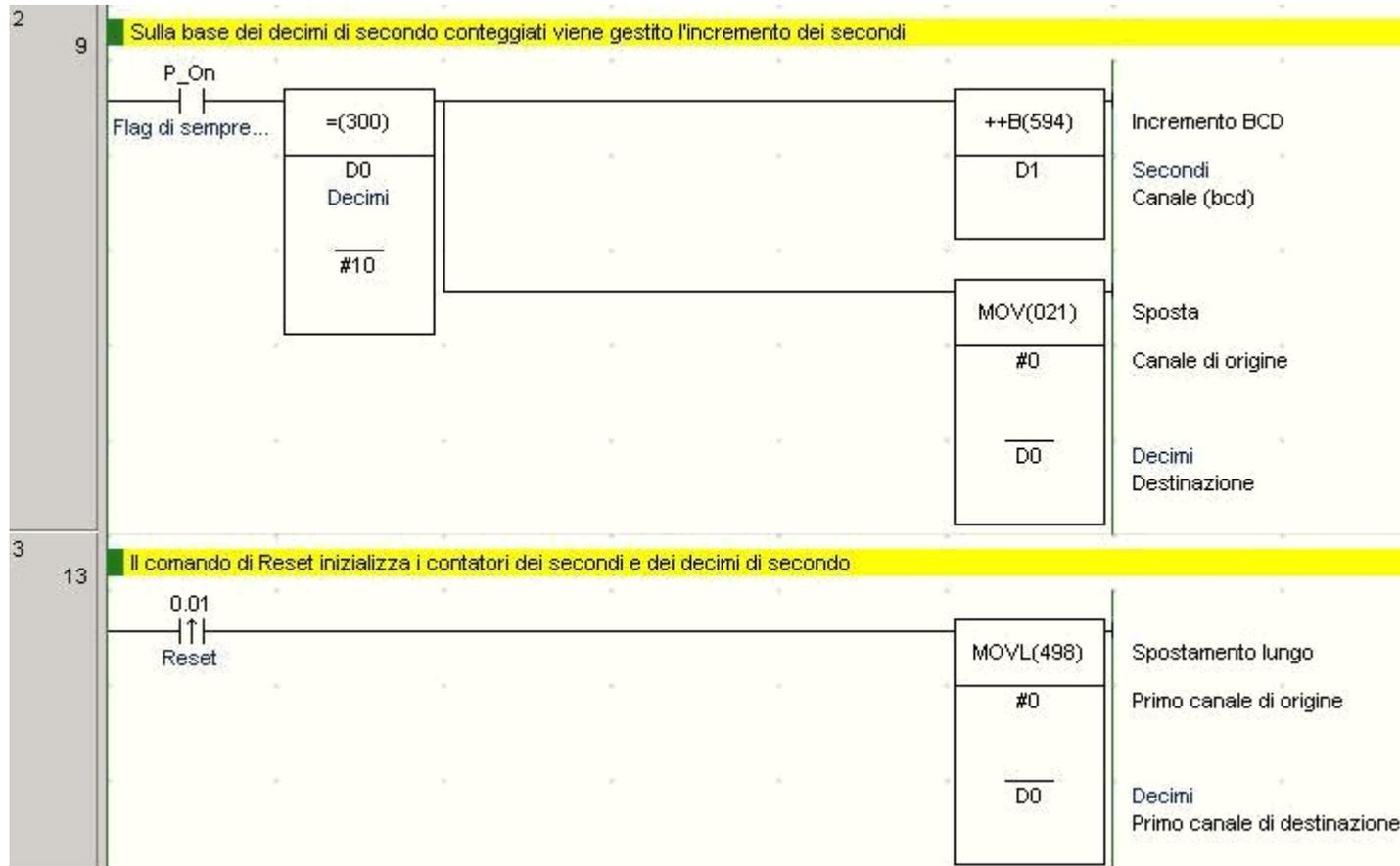
- Realizzare un cronometro che gestisca i secondi e i decimi di secondo
- Un unico pulsante dovrà comandare lo Start e lo Stop del cronometro
- Prevedere anche un pulsante di Reset

^ Attivazione	BOOL	200.00		Lavoro	cronometro attivo
▣ Decimi	CHANNEL	D0		Lavoro	decimi di secondo
^ Reset	BOOL	0.01		Lavoro	Pulsante di Reset
▣ Secondi	CHANNEL	D1		Lavoro	secondi
^ Start_pausa	BOOL	0.00		Lavoro	Pulsante di start / pausa

Esercizio 9: Soluzione

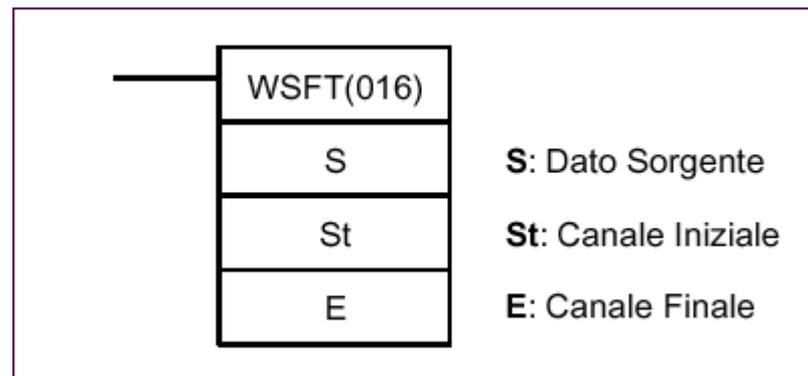


Esercizio 9: Soluzione



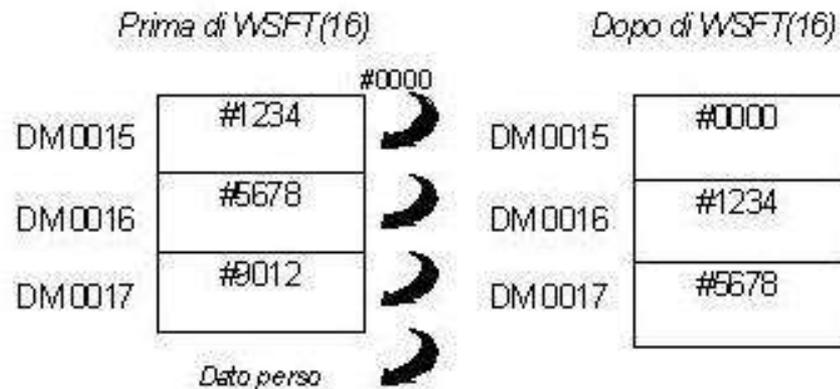
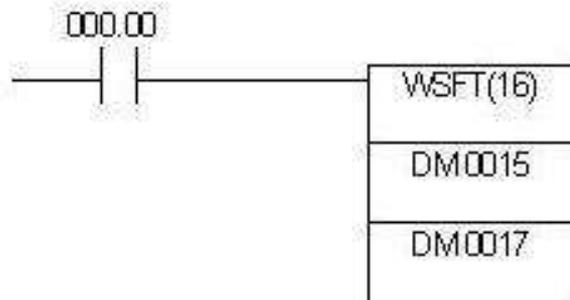
La funzione WSFT

- L'istruzione WSFT permette di realizzare uno scorrimento dati a livello di canale
- Lo scorrimento può avvenire su uno o più canali appartenenti alla stessa area dati
- Ad ogni esecuzione dell'istruzione si ha uno scorrimento dati pari ad un intero canale



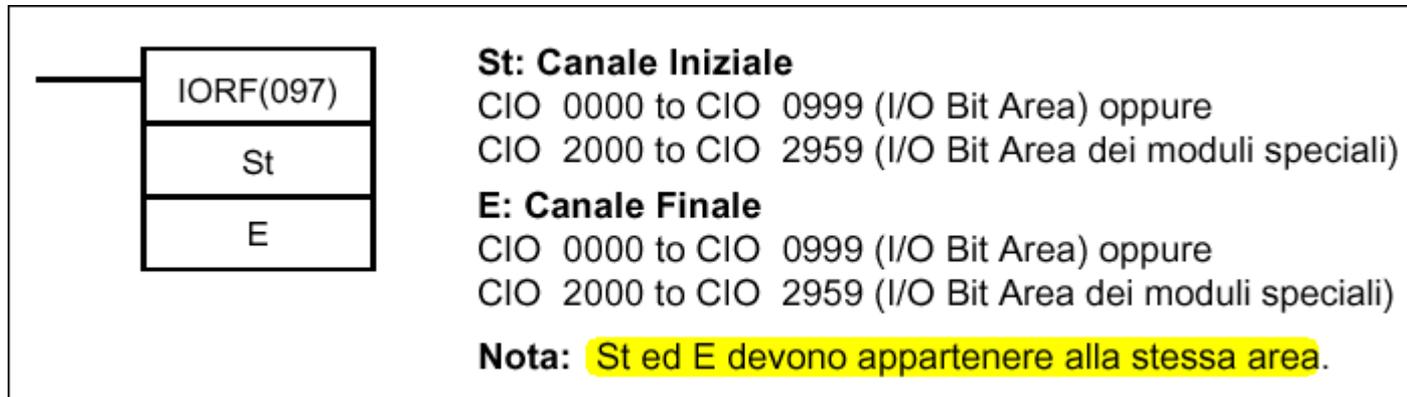
La funzione WSFT

- Dopo lo scorrimento, nel canale di partenza viene scritto il valore 0 mentre il contenuto dell'ultimo canale viene perso
- Esempio:



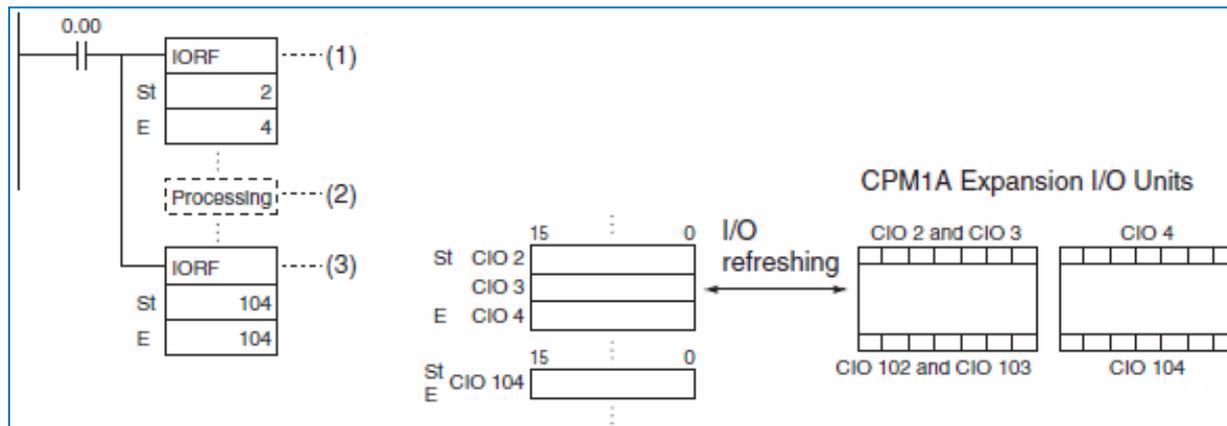
La funzione IORF

- L'immagine di processo (IP) degli I/O viene normalmente aggiornata a fine scansione
- L'istruzione IORF (aggiornamento degli I/O) provoca, nell'istante in cui viene eseguita, l'aggiornamento reale dei canali allocati ai moduli di I/O standard o ai moduli di I/O speciali



La funzione IORF

- Questa istruzione può essere utilizzata per ovviare a problemi legati al campionamento a scansione della IP, nel caso in cui si abbiano tempi di scansione elevati
- L'IORF può essere utilizzata per campionare quindi il segnale con una frequenza più elevata



L'istruzione BIN

- L'istruzione BIN converte un numero decimale (BCD) di 4 cifre in un numero esadecimale (BIN) a 16 bit e trasferisce il dato convertito in un altro canale
- Se il valore convertito è 0000 allora il flag di uguale (P_EQ) viene posto ad ON
- Se il valore da convertire non è BCD, il flag di errore (P_ER) viene posto ad ON

L'istruzione BCD

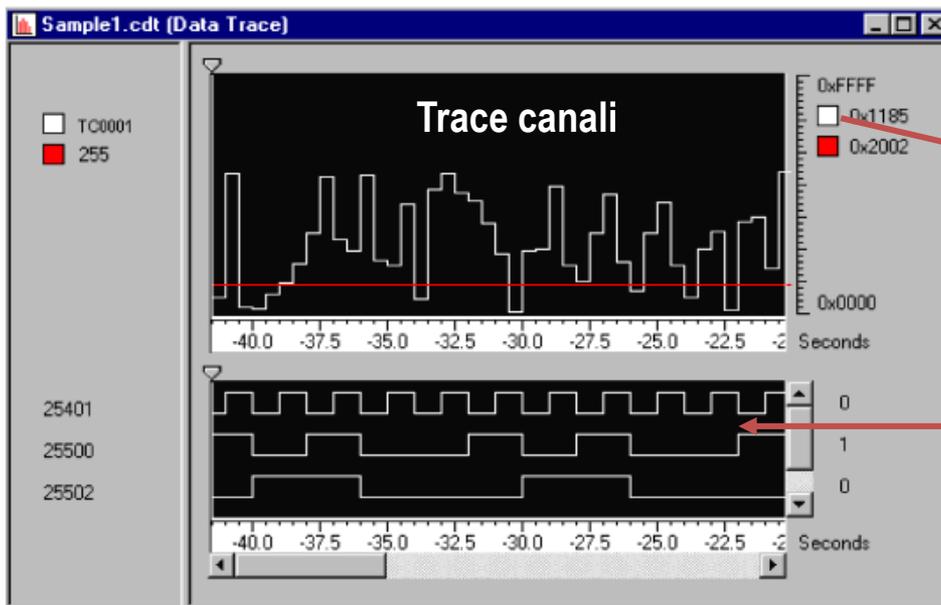
- L'istruzione BCD converte un numero esadecimale (BIN) a 16 bit in un numero decimale (BCD) di 4 cifre e trasferisce il dato convertito in un altro canale
- Se il valore convertito è 0 allora si alza il flag P_EQ
- Il massimo valore convertibile è pari a 270F (HEX) che equivale al valore 9999 (BCD)
- Se il valore da convertire è maggiore di 270F il flag P_ER viene posto ad ON

Data Trace

- Dal Cx-Programmer con un click sul pulsante  si attiva il Data Trace :
 - La modalità Data Trace effettua il campionamento dei canali e dei bit specificati, partendo dall'istante in cui si avvera la condizione di Trigger. Quando il Buffer si riempie o il campionamento viene fermato, i dati campionati possono essere caricati e visionati.
 - La modalità Monitor diagramma temporale effettua il campionamento di canali e bit On-Line
- Una volta scelto Modalità dal menù Operazione attraverso il pulsante  si procede alla configurazione del Data-Trace

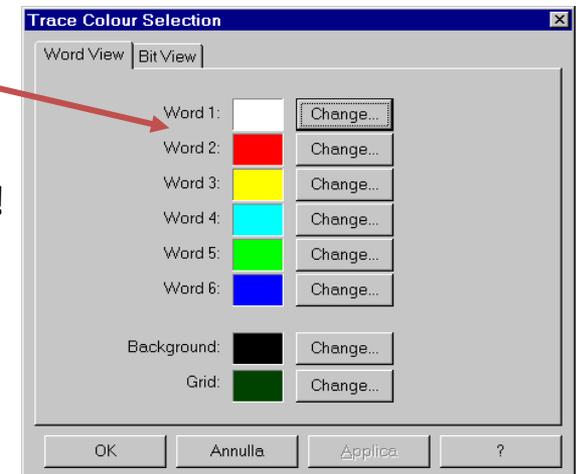
Data Trace

- E' possibile configurare il trace simultaneo di 6 canali e 12 bit
- I valori possono essere visualizzati in stile "Trend graph"
- La velocità di aggiornamento minima è di 10 millisecondi sfruttando la trace memory del PLC
- I grafici possono essere stampati o salvati su disco



Il colore delle tracce è selezionabile!

Trace stato dei Bit



Liste Incrociate

- CX-Programmer dispone di una gestione delle liste incrociate molto efficace
- E' possibile avere liste globali al programma, oppure limitate al contatto corrente
- Si può scegliere tra la vista di una specifica area dati e la lista incrociata con tutte le aree
- Dopo avere generato un report è possibile stamparlo

Tipo di lista

- Completa
- Vista indirizzi utilizzati
- Vista completa degli indirizzi

Report type: Memory area:

Free UM:

Total UM:

Address	Program	Step	Instruction	Start Addr...	Symbol
0	NewProgram1	0	LD [1]	0.00	Marcia
0	NewProgram1	2	ANDNOT [1]	0.01	Arresto
0.00	NewProgram1	0	LD [1]	0.00	Marcia
0.01	NewProgram1	2	ANDNOT [1]	0.01	Arresto
6	NewProgram1	3	OUT [1]	6.00	Motore
6	NewProgram1	1	OR [1]	6.00	Motore

Selezione dell'area dati

Selezionare All per tutte le aree dati

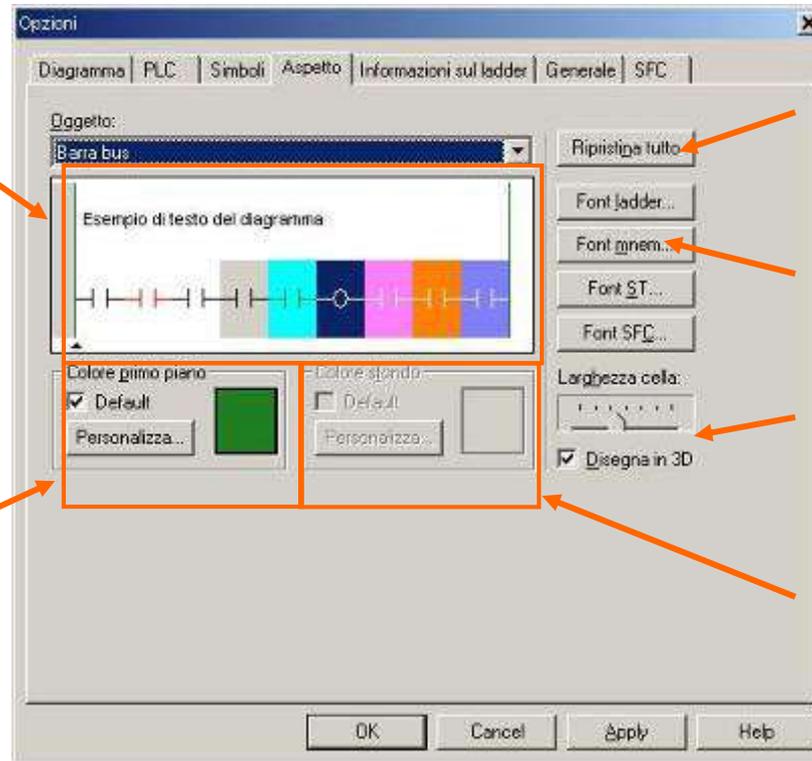
Facendo doppio click sulla riga, l'editor ladder salta all'istruzione corrispondente

Personalizzazione

- E' possibile modificare font e colori degli elementi del diagramma ladder

Aspetto complessivo del ladder

Modifica del colore dell'elemento



Applica le impostazioni di default

Selezione del Font

Disegna il ladder con effetti 3D

Modifica del colore di sfondo dell'elemento

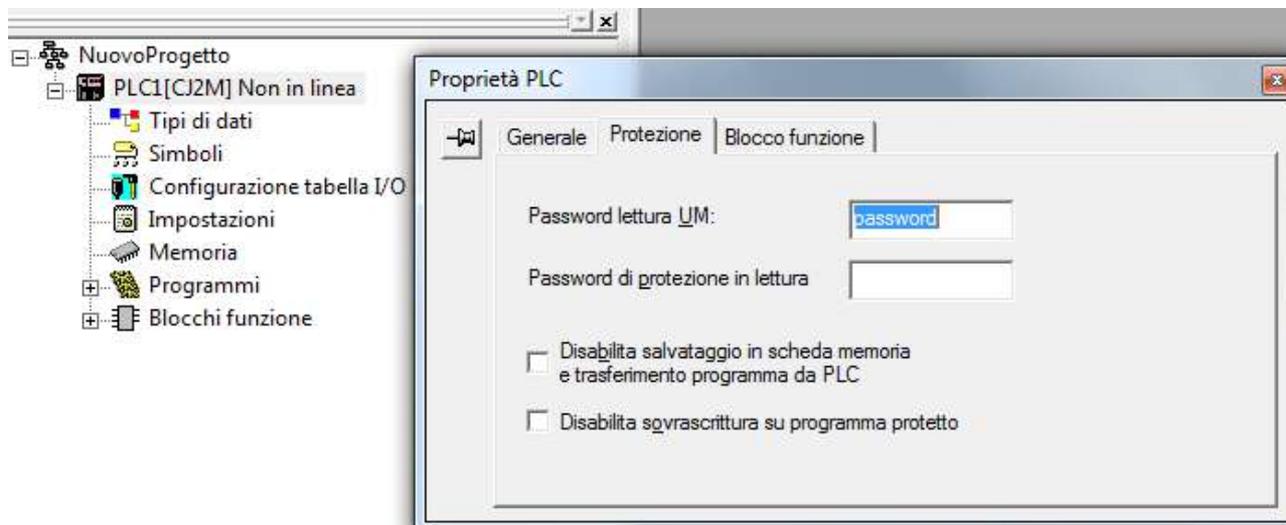
Impostazione di una password di protezione

E' possibile proteggere il progetto con password su tre diversi livelli:

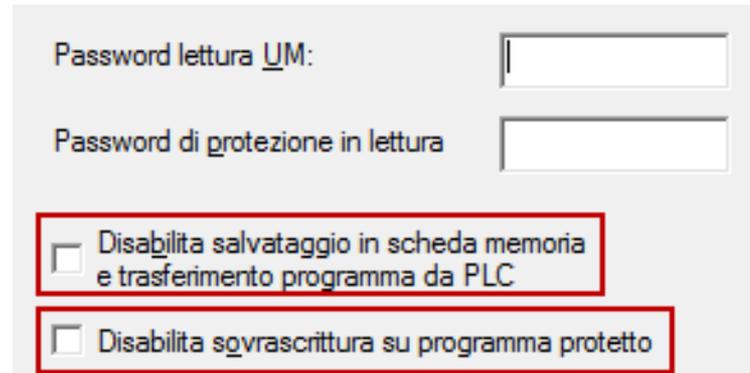
- Password di lettura UM, ovvero una protezione completa sul progetto che rende impossibile effettuare l'upload dal PLC.
- Password di protezione in lettura. Questo livello di protezione permette di rendere irraggiungibili solamente alcuni task all'interno del progetto.
- Password di protezione dei Function Block. Questa opzione permette di impostare una password di protezione indipendente per ogni Function Block presente nel progetto.

Impostazione password di lettura UM

- Click con il tasto destro sul nome del PLC → proprietà
- Nella finestra inserire la password desiderata nel campo Password di lettura UM.
- Impostare la password da: PLC → Protezione → Imposta password
- Andare online con il PLC e trasferire il progetto.



Impostazione password di lettura UM



Password lettura UM:

Password di protezione in lettura

Disabilita salvataggio in scheda memoria e trasferimento programma da PLC

Disabilita sovrascrittura su programma protetto

Nella finestra di impostazione della password, spuntando i rispettivi checkbox, è possibile:

- Disabilitare la sovrascrittura del progetto protetto
- Disabilitare l'upload del progetto ed il salvataggio del programma su scheda di memoria CF

Impostazione password di protezione in lettura

- Click con il tasto destro sul nome del PLC → proprietà
- Nella finestra inserire la password desiderata nel campo Password di protezione in lettura.
- Click con il tasto destro sul task da proteggere con password → Proprietà → Protezione
- Selezionare il campo 'Protezione in lettura di task'
- Impostare la password da: PLC → Protezione → Imposta password
- Collegarsi al PLC e trasferire il progetto.

Impostazione password di protezione dei Function Block

- Click con il tasto destro sul Function Block da proteggere con password → proprietà → Protezione → Impostazione
- Inserire la password nei due campi 'Password' e confermare con 'Impostazione'
- Collegarsi al PLC e trasferire il progetto.

Ladder Printing Utility

- Dal menu File, seguire Imposta Pagina / Impostazione Print Utility. Compare la finestra indicata, dalla quale è possibile
- Impostare l'aspetto della stampa



Varie

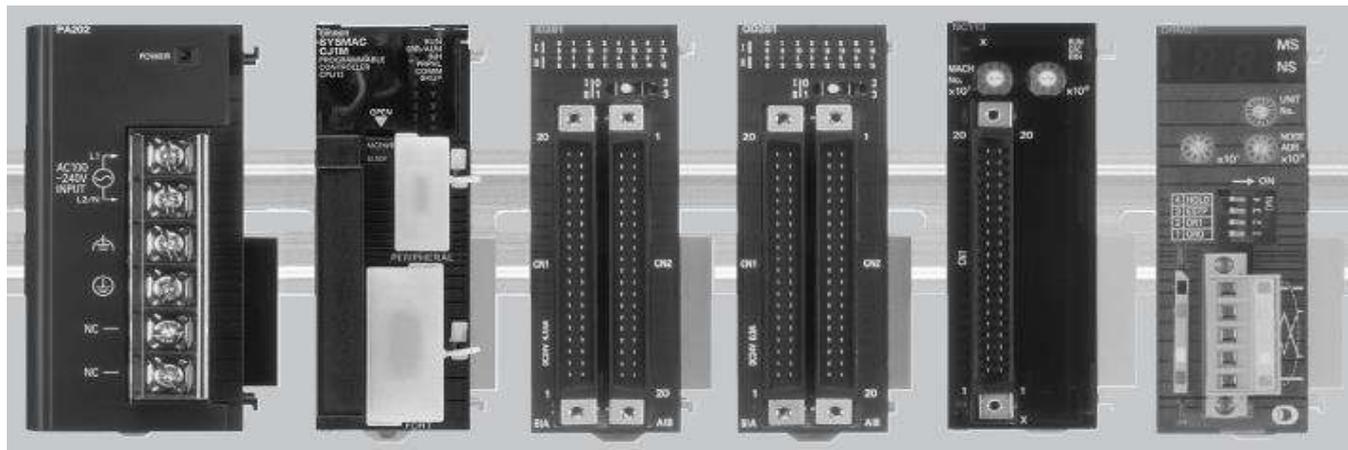
- Prospetto riferimento incrociato: utilizzata per fare un controllo sui vari indirizzi utilizzati all'interno dei programmi permettendo anche una ottimizzazione delle aree di memoria. Per generarla attivare il tasto 
- Inserisci PLC: all'interno dello stesso progetto è possibile inserire un nuovo PLC  Inserisci PLC...
- Inserisci sezione: permette di aggiungere una nuova sezione al programma attuale  Inserisci sezione
- Le sezioni consentono la suddivisione del programma in blocchi logici. L'esecuzione del programma avviene nel modo consueto

I PLC modulari

Gamma, prestazioni, espandibilità,
aree di memoria e allocazione I/O

PLC modulare

- Il sistema PLC (Alimentatore, Cpu, Interfacce...) è realizzato in base al sistema da controllare.
- Il PLC modulare risulta essere più flessibile come configurazione ma necessita l'assemblaggio di più moduli.



La gamma

- PLC modulari senza rack
 - CJ1-M/G/H
 - CJ2-M/H



- PLC modulari con rack
 - CS1G/H
 - CS1D



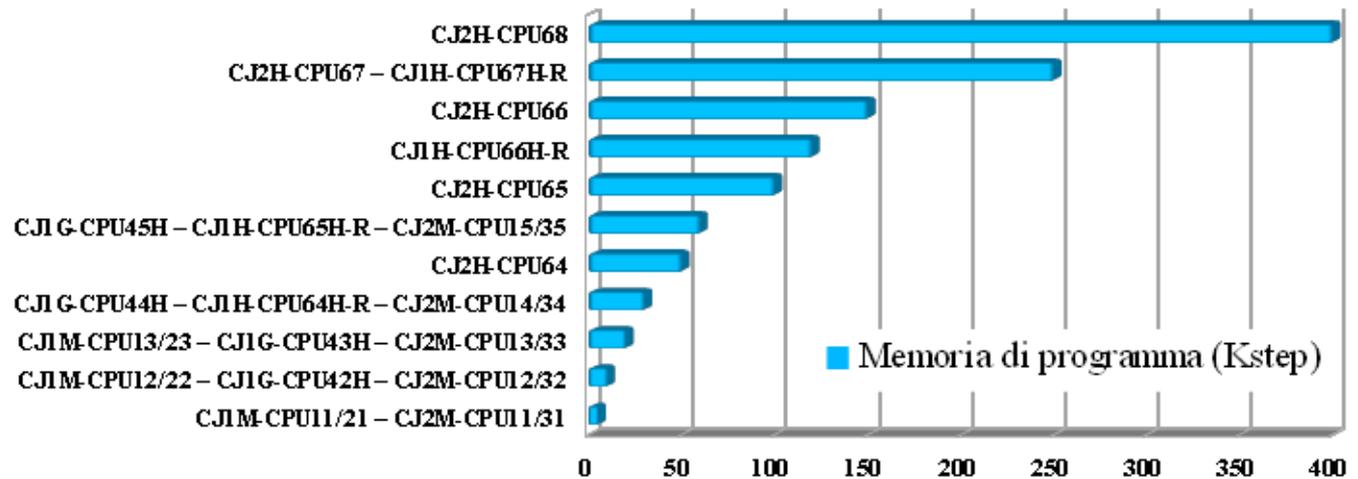
PLC serie CJ2



Classificazione delle CPU

- Per memoria di programma

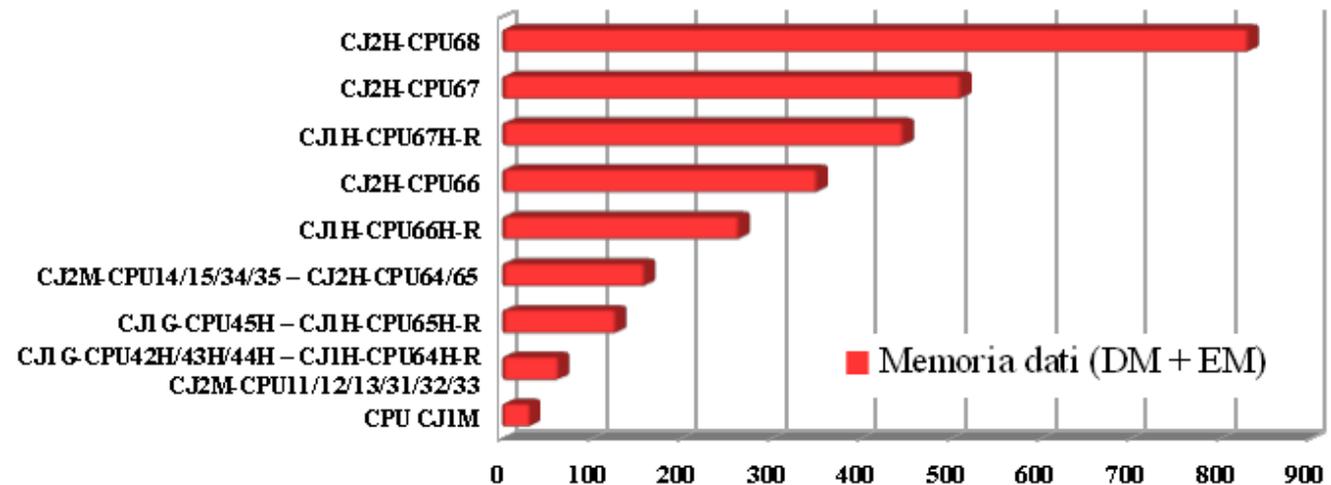
PLC	Memoria
CJ1M-CPU11/21	5 Kstep
CJ1M-CPU12/22	10 Kstep
CJ1M-CPU13/23	20 Kstep
CJ1G-CPU42H	10 Kstep
CJ1G-CPU43H	20 Kstep
CJ1G-CPU44H	30 Kstep
CJ1G-CPU45H	60 Kstep
CJ1H-CPU64H-R	30 Kstep
CJ1H-CPU65H-R	60 Kstep
CJ1H-CPU66H-R	120 Kstep
CJ1H-CPU67H-R	250 Kstep
CJ2M-CPU11/31	5 Kstep
CJ2M-CPU12/32	10 Kstep
CJ2M-CPU13/33	20 Kstep
CJ2M-CPU14/34	30 Kstep
CJ2M-CPU15/35	60 Kstep
CJ2H-CPU64	50 Kstep
CJ2H-CPU65	100 Kstep
CJ2H-CPU66	150 Kstep
CJ2H-CPU67	250 Kstep
CJ2H-CPU68	400 Kstep



Classificazione delle CPU

- Per memoria dati (DM + EM)

PLC	Memoria
CPU CJ1M	32 Kword
CJ1G-CPU42H	64 Kword
CJ1G-CPU43H	64 Kword
CJ1G-CPU44H	64 Kword
CJ1G-CPU45H	128 Kword
CJ1H-CPU64H-R	64 Kword
CJ1H-CPU65H-R	128 Kword
CJ1H-CPU66H-R	265 Kword
CJ1H-CPU67H-R	448 Kword
CJ2M-CPU11/31	64 Kword
CJ2M-CPU12/32	64 Kword
CJ2M-CPU13/33	64 Kword
CJ2M-CPU14/34	160 Kword
CJ2M-CPU15/35	160 Kword
CJ2H-CPU64	160 Kword
CJ2H-CPU65	160 Kword
CJ2H-CPU66	352 Kword
CJ2H-CPU67	512 Kword
CJ2H-CPU68	832 Kword



N.B.: Tutte le CPU hanno 32 Kword di Data Memory

Classificazione delle CPU

- Per numero di I/O:
 - 160 Bit: CJ1M-CPU11/21
 - 320 Bit: CJ1M-CPU12/22
 - 640 Bit: CJ1M-CPU13/23
 - 960 Bit: CJ1G-CPU42H/43H
 - 1.280 Bit: CJ1G-CPU44H
 - 2.560 Bit: CJ1G-CPU45H – CJ1H-CPU64/65/66/67H-R – CPU CJ2
- Per sistemi di espansione collegabili:
 - Nessuna espansione per i PLC CJ1M-CPU11, CPU12, CPU21 e CPU22
 - 1 espansione per i PLC CJ1M-CPU13 e CPU23
 - 2 espansione per i PLC CJ1G-CPU42H e CPU43H
 - 3 espansioni per i PLC CJ1G-CPU44H, CPU45H, PLC CJ1H e CJ2

Classificazione delle CPU

- Per velocità di elaborazione:
 - Serie CJ1M
 - Istruzione LD: 0,10 μ s.
 - Istruzione MOV: 0,30 μ s.
 - Serie CJ1G
 - Istruzione LD: 0,04 μ s.
 - Istruzione MOV: 0,20 μ s.
 - Serie CJ1H
 - Istruzione LD: 0,02 μ s.
 - Istruzione MOV: 0,18 μ s.
 - Serie CJ2M
 - Istruzione LD: 0,04 μ s.
 - Istruzione MOV: 0,12 μ s.
 - Serie CJ2H
 - Istruzione LD: 0,016 μ s.
 - Istruzione MOV: 0,05 μ s.

Linguaggi supportati

- Il linguaggio mnemonico e il linguaggio ladder sono supportati da tutti i PLC per qualsiasi CPU.
- Il testo strutturato e i function block sono supportati per le serie CS/CJ; CS1-H, CJ1-H e CJ1M CPU Units con firmware versione 3 o superiore.
- L'SFC è supportato da alcune CPU come mostrato nella seguente tabella:

PLC model	CPU model
CJ2 unit version 1.0 or later	CJ2 CPU
CS1G-H unit version 4.0	CS1G-CPU45H/44H/43H/42H
CS1H-H unit version 4.0	CS1H-CPU67H/66H/65H/64H/63H
CJ1G-H unit version 4.0	CJ1G-CPU45H/44H/43H/42H
CJ1H-H unit version 4.0	CJ1H-CPU67H/66H/65H/67H-R/66H-R/65H-R/64H-R
CJ1M unit version 4.0	CJ1M-CPU23/22/21/13/12/11

Modulo CPU

Dispositivo di bloccaggio a slitta

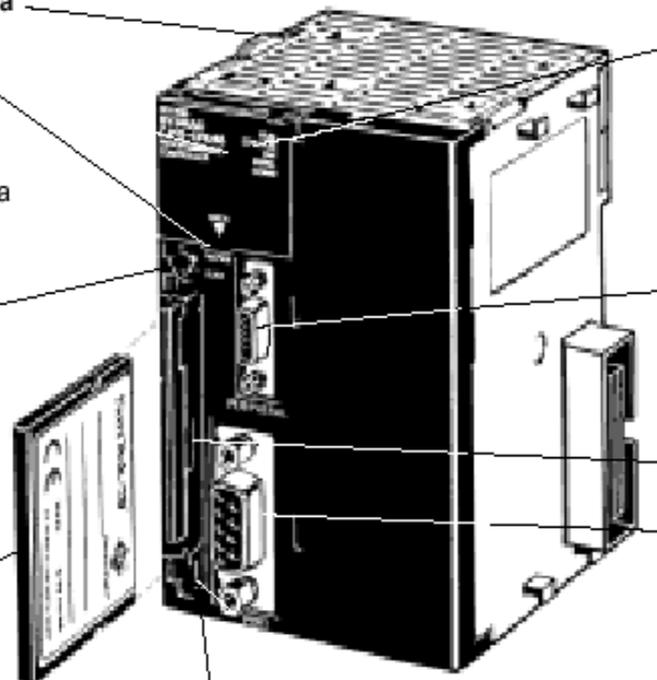
Led Memory Card

Quando l'alimentazione è attiva, lampeggia il led MCPWR (di colore verde). Quando si accede alla memory card lampeggia il led BUSY (di colore arancione).

Interruttore alimentazione Memory Card

L'interruttore dell'alimentazione della cartuccia di memoria viene premuto per disattivare l'alimentazione prima di rimuovere la cartuccia di memoria.

Memory Card



Led e Dip Switches

Porta periferica

La porta periferica viene collegata ai dispositivi di programmazione quali console di programmazione ed host computer o come interfaccia per adattatori RS232-C.

Connettore Memory Card

Porta RS232-C

La porta RS232-C viene collegata ai dispositivi periferici diversi dalle console di programmazione, come host computer, dispositivi esterni di impiego generale e terminali programmabili.

Pulsante di espulsione della Memory Card

Dip Switches

- Su tutte le CPU sono presenti 8 dip-switch per il setup hardware:

Dip 1 - Protezione della memoria programma

Dip 2 - Trasferimento automatico del programma da Memory Card

Dip 3 - Non utilizzato

Dip 4 - ON: Porta periferica funzionante come da Setup Software

OFF: Porta periferica funzionante in Auto-Detect da Cx-Programmer

Dip 5 - ON: Porta RS232 funzionante in Auto-Detect da Cx-Programmer

OFF: Porta RS232 funzionante come da Setup software

Dip 6 - Alza il bit A395.12

Dip 7 - ON: Permette di salvare programma e setup su Memory Card

Dip 8 - Non utilizzato

NOTA: Per la serie CJ2, gli switch 3, 4 e 8 non sono utilizzati.

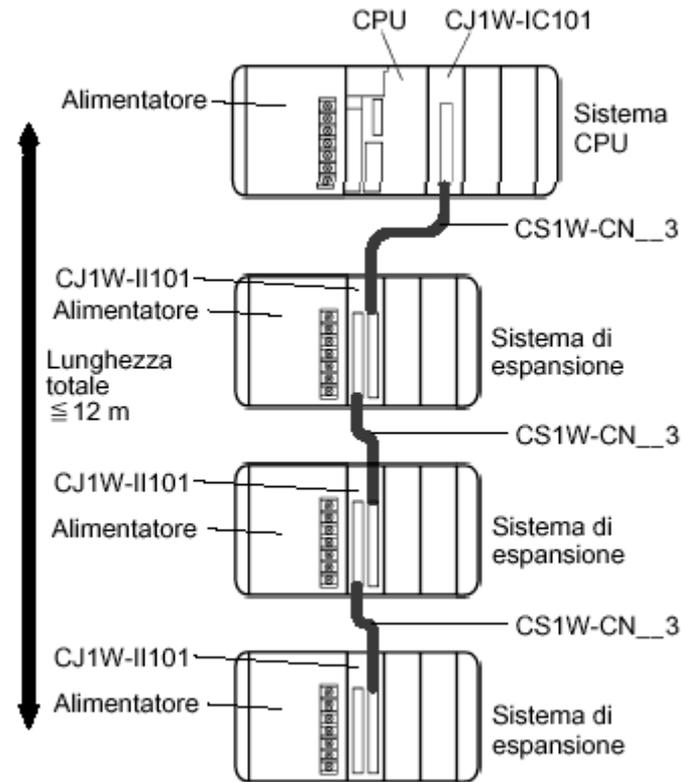
L'impostazione predefinita per tutti gli switch è OFF.

Setup della CPU

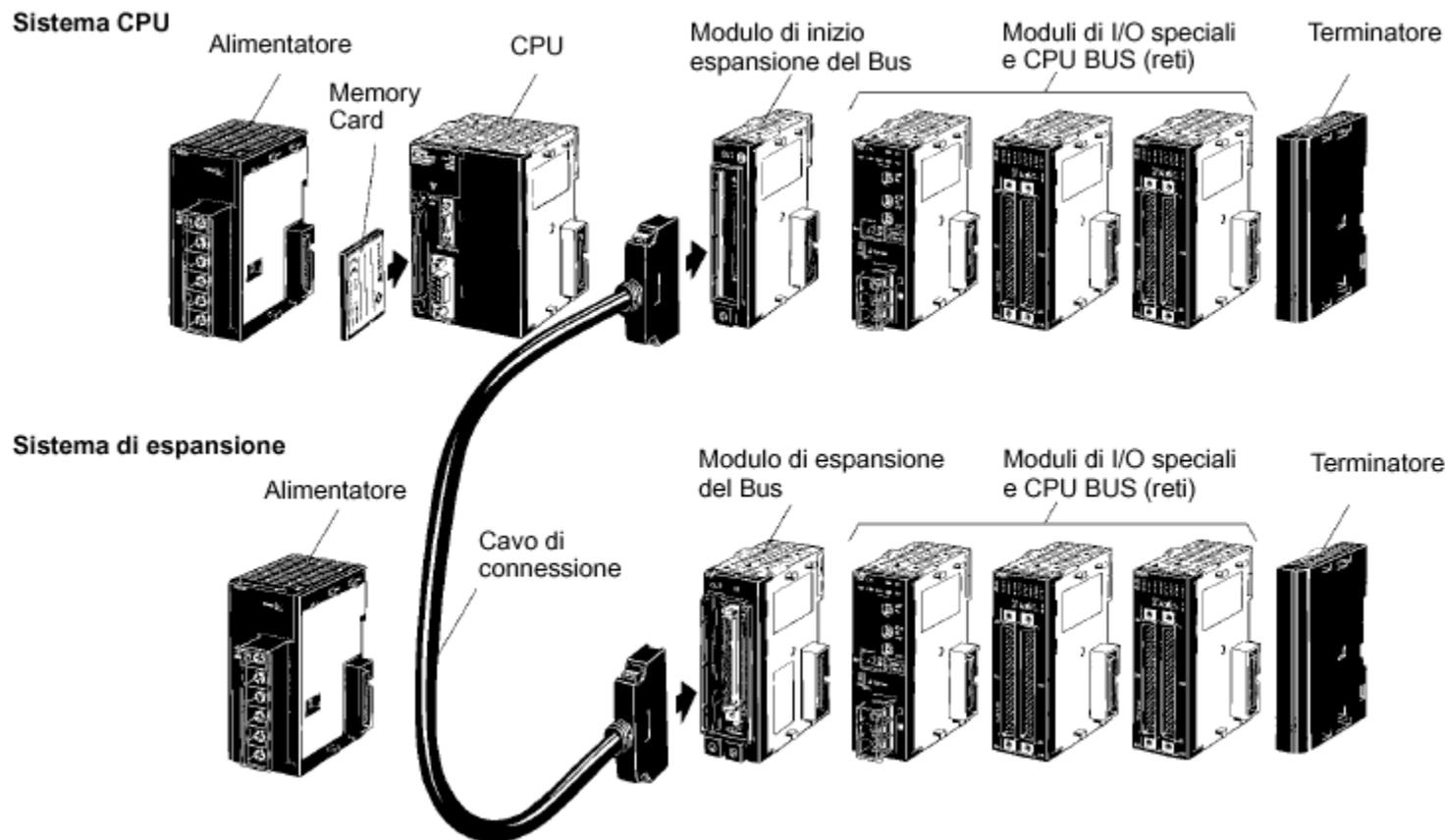
- La finestra Impostazioni PLC di CX-Programmer rappresenta uno strumento completo per la configurazione del PLC.
- Per accedere alla finestra di impostazioni è sufficiente richiamare la voce Modifica - Impostazioni dal menù PLC, o cliccare due volte sull'icona relativa, nell'area di progetto
( Impostazioni)
- Il menù Opzioni della finestra permette di leggere, trasferire, verificare o proteggere in scrittura la configurazione corrente, oppure riportare la configurazione alle impostazioni di default.
- La configurazione può essere eventualmente anche stampata (File - Stampa)

Configurazione ed espandibilità

- Max. 10 moduli installabili a fianco della CPU
- 0/1/2/3 unità di espansione (10 moduli per unità) per un totale di 10/20/30/40 moduli collegabili
- Ogni unità di espansione è composta da alimentatore, modulo di interfaccia e moduli di I/O
- Distanza massima totale: 12 m.

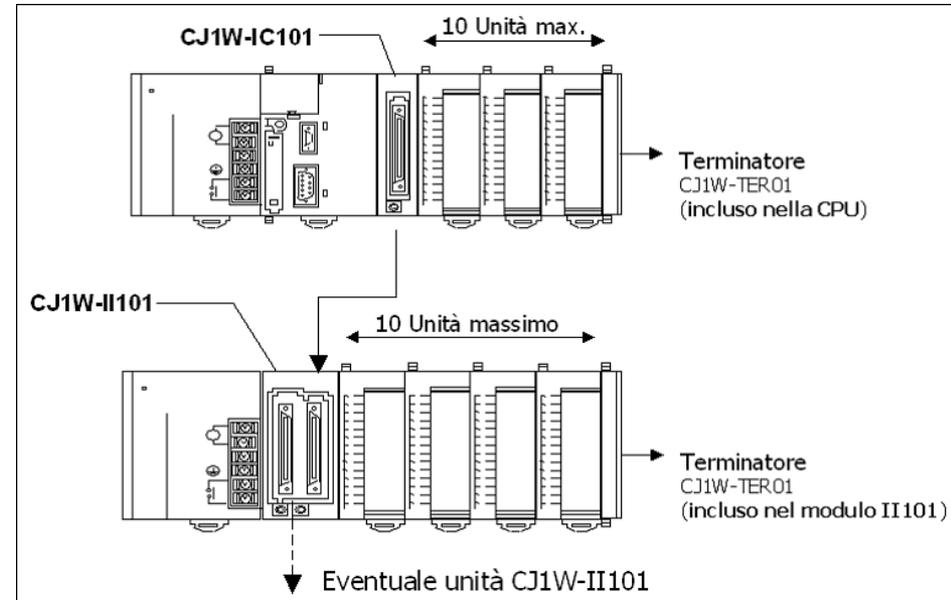


Configurazione ed espandibilità



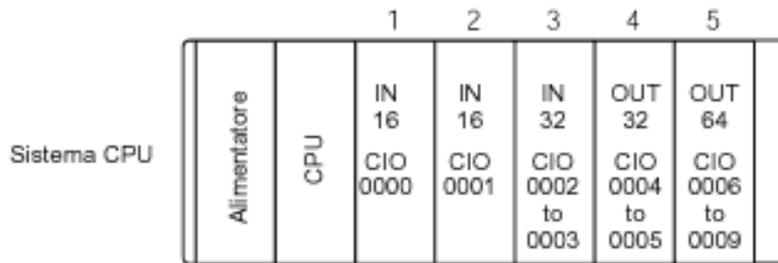
Configurazione ed espandibilità

- La configurazione massima prevede un totale di 40 moduli installabili.
- E' necessario rispettare le seguenti specifiche:
 - Moduli I/O Standard: massimo 40
 - Moduli I/O Speciali: massimo 40 (indirizzabili da 0 a 95)
 - Moduli Bus CPU: massimo 16

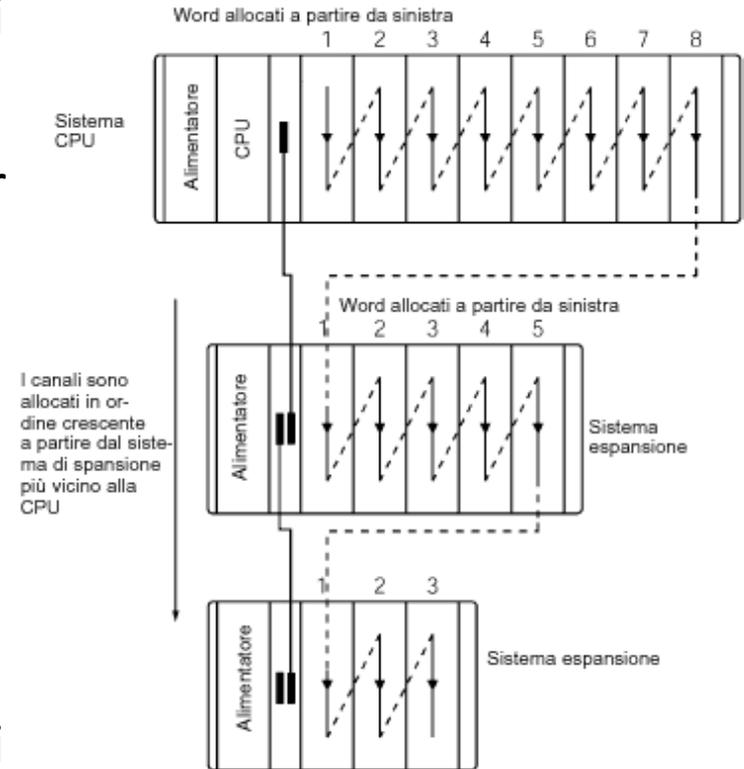


Allocazione I/O (moduli standard)

- L'allocazione dei canali CIO per i moduli di I/O standard procede in modo consecutivo seguendo l'ordine di installazione, sia per gli ingressi che per le uscite, a partire dal canale CIO 0000.



NB: L'eventuale modulo di espansione va posizionato subito a destra della CPU (o subito a destra dell'alimentatore, nei sistemi di espansione)



Allocazione I/O (moduli speciali)

- 10 canali CIO per modulo, allocati secondo il numero di unità N.
 - Calcolo indirizzo base area CIO:
 $ch = 2000 + 10 \times N$

- 100 canali D per modulo, allocati secondo il numero di unità N.
 - Calcolo indirizzo base area D:
 $ch = 20000 + 100 \times N$

Numero unità	Canali assegnati
0	CIO 2000... CIO 2009
1	CIO 2010... CIO 2019
2	CIO 2020... CIO 2029
⋮	⋮
15	CIO 2150... CIO 2159
⋮	⋮
95	CIO 2950... CIO 2959

Numero Unità	Canali Assegnati
0	D20000 to D20099
1	D20100 to D20199
⋮	⋮
10	D21000 to D21099
⋮	⋮
95	D29500 to D 29599

Allocazione I/O (moduli BUS)

- 25 canali CIO per modulo, allocati secondo il numero di unità N.
 - Calcolo indirizzo base area CIO:
 $ch = 1500 + 25 \times N$

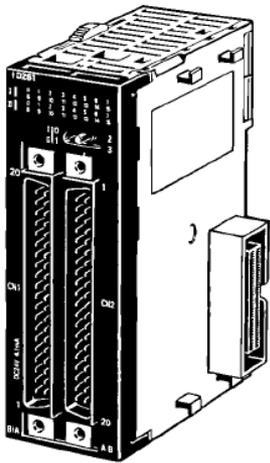
- 100 canali D per modulo, allocati secondo il numero di unità N.
 - Calcolo indirizzo base area D:
 $ch = 30000 + 100 \times N$

Numero unità	Canali assegnati
0	CIO 1500... CIO 1524
1	CIO 1525... CIO 1549
2	CIO 1550... CIO 1574
15	CIO 1875... CIO 1899

Numero Unità	Canali assegnati
0	D30000 to D30099
1	D30100 to D30199
2	D30200 to D30299
15	D31500 to D31599

Allocazione I/O - Riepilogo

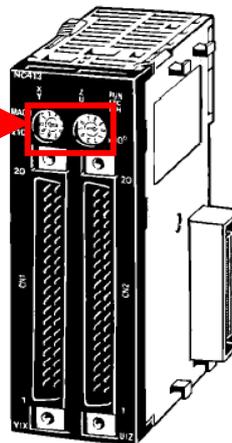
Moduli I/O standard



Da CIO 0000 a CIO 0079
(8 , 16 , 32 or 64 punti)
(Max. 40 moduli)

Numero di modulo I/O speciale
(Max. 40 moduli)

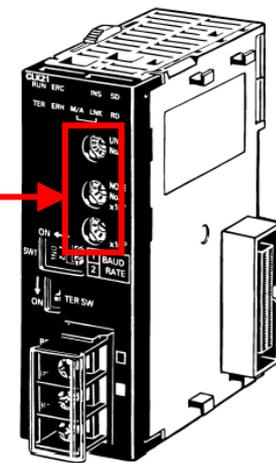
Moduli I/O speciali



Da CIO 2000 a CIO 2959
(Refresh Area: 10 word/unità)

Da D20000 a D29599
(Setup Area : 100 word/unità)

Moduli Bus CPU



Da CIO 1500 a CIO 1899
(Refresh Area : 25 canali/unità)

Da D30000 a D31599
(Setup Area : 100 canali/unità)

Numero di modulo Bus CPU
(Max. 16 moduli)

Allocazione I/O - Esempio

	0	1	2	3	4
Alimentatore					
CPU	IN 16 pt CIO 0000	Modulo I/O Specia CIO 2000 a 2009	Modulo Bus CPU CIO 1500 a 1524	OUT 16 pt CIO 0001	Modulo Bus CPU CIO 1525 a 1549

Slot	Modulo	Canali richiesti	Canali allocati	Numero modulo	Gruppo
0	CJ1W-ID211 Modulo Ingresso 16-punti DC	1	CIO 0000	---	Modulo I/O Standard
1	CJ1W-AD081 Modulo Ingresso Analogico	10	Da CIO 2000 a CIO 2009	0	Modulo I/O Speciale
2	CJ1W-SCU41 Modulo Comunicazione Seriale	25	Da CIO 1500 a CIO 1524	0	Modulo Bus CPU
3	CJ1W-OD211 Modulo Uscita a 16 punti NPN	1	CIO 0001	---	Modulo I/O Standard
4	CJ1W-CLK21 Modulo Controller Link	25	Da CIO 1525 a CIO 1549	1	Modulo BUS CPU

La tabella I/O

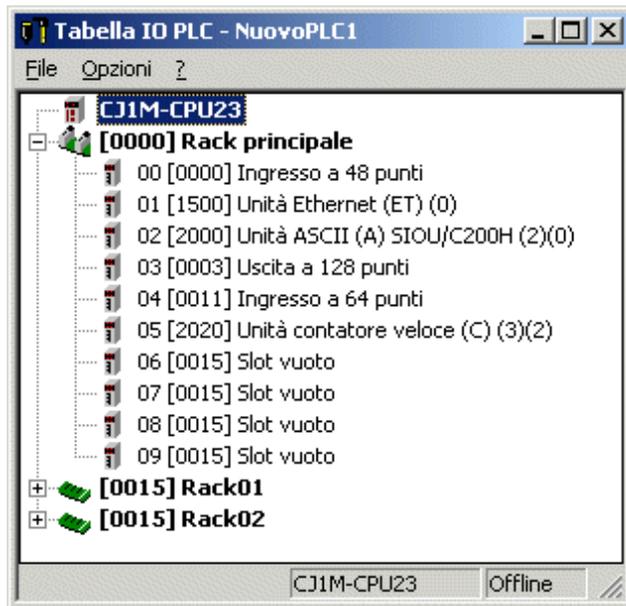
- La configurazione hardware dei moduli del CJ1 può essere memorizzata in una tabella detta 'Tabella di I/O'
- All'accensione, se la tabella è stata registrata, il sistema verifica la configurazione attuale con quella registrata; in caso di mancata corrispondenza viene generato un errore:
 - Se è stato tolto, aggiunto o spostato un modulo, il sistema fornisce la segnalazione di allarme non fatale 'I/O VERIFY ERROR'
 - Se la variazione di configurazione è costituita invece dallo scambio di posizione tra un modulo di ingresso ed uno di uscita, viene fornita la segnalazione di allarme fatale 'I/O SET ERROR' e l'esecuzione del programma si arresta

La tabella I/O

- La tabella di I/O può essere:
Creata, Letta, Verificata e Cancellata
- Se la tabella è stata cancellata il sistema all'accensione non esegue alcun controllo
- In CX-Programmer è stato previsto un intero menù dedicato alla gestione della tabella di I/O, cui si accede cliccando su Tabella I/O dal menù PLC - Modifica - oppure facendo doppio click sull'icona relativa, nell'area di progetto ( Tabella IO)
- Il canale A261 del PLC contiene i dettagli di eventuali errori riscontrati durante la creazione della tabella di I/O

La tabella I/O – CX-Programmer

Rappresentazione grafica della tabella di I/O



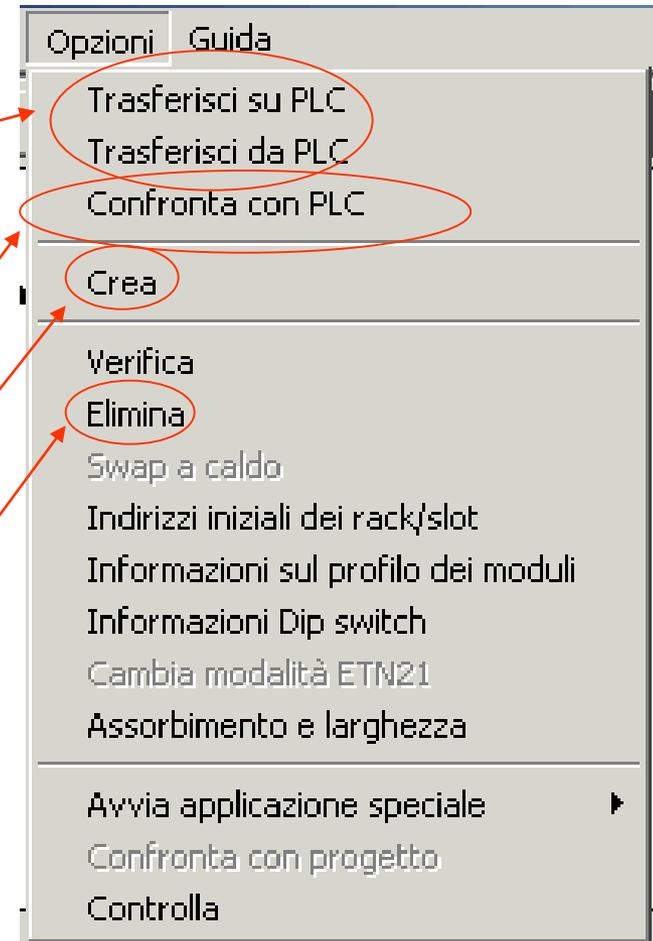
Legge la tabella di I/O dal PLC collegato

Scrive la tabella di I/O sul PLC.

Confronta la tabella di I/O del PLC con quella memorizzata nel progetto

Genera la tabella di I/O

Elimina dal PLC la tabella di I/O registrata



Tipi di memoria

- Le aree di memoria sono suddivise in:
 - RITENTIVE: sono quelle aree che permettono di mantenere i dati salvati anche dopo lo spegnimento dell'unità o al cambio della modalità operativa
 - NON RITENTIVE: sono le aree che si azzerano dopo lo spegnimento dell'unità o al cambio della modalità operativa

Aree di memoria

- Le principali aree di memoria sono:
 - Area CIO (Core I/O area): 6.144 word, nessun prefisso. Area di memoria di I/O, moduli speciali, schede di comunicazione. Area non ritentiva, viene azzerata ad ogni avvio del PLC. L'accesso può avvenire a Bit o a Canale
 - Area HR (Holding Area): 512 word, prefisso H. Area ritentiva gestibile a Bit e a Canale. Memorizzazione permanente di dati di programma
 - Area W (Work Area): 512 word, prefisso W; per l'utilizzo come registri temporanei o Bit di appoggio; viene usata tipicamente per l'allocazione automatica. Accesso a Bit e a canale. Non ritentiva

Aree di memoria

- Area A (Auxiliary Area): 960 word di cui 448 a sola lettura per la serie CJ1; per la serie CJ2, 3.008 word di cui 1.984 di sola lettura, prefisso A. Contiene informazioni per la diagnostica, lo stato e la configurazione del PLC
- Area Temporizzatori (Timer Area): 4.096 word, prefisso T. Viene simulato il funzionamento di temporizzatori
- Area Contatori (Counter Area): 4.096 word, prefisso C. Viene simulato il funzionamento di contatori
- Area DM (Data Memory): 32.768 word, prefisso D. Area di memoria ritentiva per la memorizzazione dei dati. Viene utilizzata anche per la memorizzazione della configurazione dei moduli speciali. Per la serie CJ2, allocabile anche a bit

Aree di memoria

- Area EM (Expanded Memory): 32.768 word per banco, prefisso da E0 a En. Area ritentiva indirizzabile a canale.

Il numero di banchi a disposizione dipende dal modello di CPU.

I banchi EM possono essere trasformati in unica area di memoria di massa detta File Memory. Quest'area può essere usata allo stesso modo della Memory Card.

Le istruzioni aritmetiche

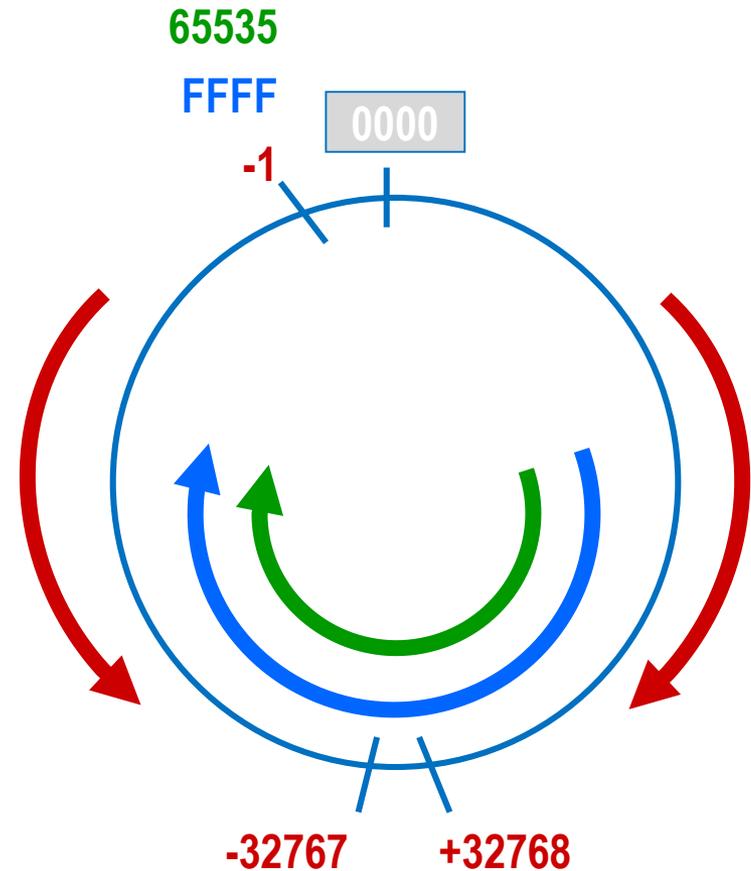
Rappresentazione dei dati

- Per interpretare correttamente un valore numerico è necessario conoscere quale tipo di notazione viene utilizzata:

HEX

Decimale

Decimale con segno



I dati nelle istruzioni

- Ogni istruzione ha, per i propri operandi, dei Tipi di dati predefiniti
- Questo rende possibile un potente controllo sui dati in quanto, alla compilazione, il CX-Programmer effettua un confronto tra il tipo dell'operando inserito e il tipo dell'operando accettabile dall'istruzione
- Esempio: inserendo un dato definito UINT come operando di una +B(400) (somma BCD senza carry), alla compilazione viene generato un segnale di Warning
- Sarà comunque possibile trasferire il programma sul PLC

Istruzioni di somma e sottrazione

- Storicamente le istruzioni aritmetiche più comunemente utilizzate (ADD, SUB,.. ecc) lavoravano in formato BCD ed erano quindi soggette a tutte le limitazioni derivanti da quel tipo di codifica
- Inoltre il flag di Carry (o riporto) CY veniva sempre sommato o sottratto agli operandi, e doveva quindi essere trattato con particolari istruzioni di Set Carry (STC) e Clear Carry (CLC) in modo da gestire correttamente Overflow e risultati negativi
- Nella nuova generazione di PLC le istruzioni aritmetiche lavorano invece di default in esadecimale (se non specificato diversamente) ed il flag di Carry non influisce il risultato

Istruzioni di somma e sottrazione

- Istruzioni di somma e sottrazione binaria:

+ (-) Somma (sottrazione) binaria con segno senza CY

+L (-L) Somma (sottrazione) binaria con segno senza CY (2
word)

+C (-C) Somma (sottrazione) binaria con segno con CY

+CL(-CL) Somma (sottrazione) binaria con segno con CY (2
word)

- Istruzioni di somma e sottrazione BCD:

+B (-B) Somma (sottrazione) BCD senza CY

+BL (-BL) Somma (sottrazione) BCD senza CY (2 word)

+BC (-BC) Somma (sottrazione) BCD con CY

+BCL(-BCL) Somma (sottrazione) BCD con CY (2 word)

Istruzioni di moltiplicazione e divisione

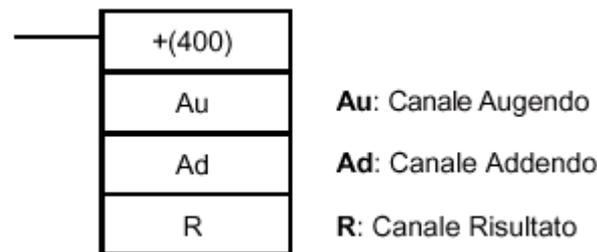
- Istruzioni di moltiplicazione e divisione binaria:
 - * (/) Moltiplicazione (divisione) binaria con segno
 - *L (/L) Moltiplicazione (divisione) binaria con segno (2 word)
 - *U (/U) Moltiplicazione (divisione) binaria senza segno
 - *UL(/UL) Moltiplicazione (divisione) binaria senza segno (2 word)
- Istruzioni di moltiplicazione e divisione BCD:
 - *B (/B) Moltiplicazione (divisione) BCD
 - *BL (/BL) Moltiplicazione (divisione) BCD (2 word)

Tabella di comparazione

Tipo di operazione	PLC serie <i>CPM1/CPM2, CQM1, C200HA</i> Alpha	PLC serie <i>CJ/CS/CP1</i>
Somma (Sottrazione) BCD senza riporto	--	+B (-B)
Somma (Sottrazione) BCD con riporto	ADD (SUB)	+BC (-BC)
Somma (Sottrazione) Binaria senza riporto	--	+ (-)
Somma (Sottrazione) Binaria con riporto	ADB (SBB)	+C (-C)
Somma (Sottrazione) BCD senza riporto (2 word)	--	+BL
Somma (Sottrazione) BCD con riporto (2 word)	ADDL (SUBL)	+BCL
Somma (Sottrazione) Binaria senza riporto (2 word)	--	+L (-L)
Somma (Sottrazione) Binaria con riporto (2 word)	ADBL (SBBL)	+CL (-CL)
Moltiplicazione (Divisione) BCD	MUL (DIV)	*B (/B)
Moltiplicazione (Divisione) Binaria con segno	MBS (DBS)	* (/)
Moltiplicazione (Divisione) Binaria senza segno	MLB (DVB)	*U (/U)
Moltiplicazione (Divisione) BCD (2 word)	MULL (DIVL)	*BL (/BL)
Moltiplicazione (Divisione) Binaria con segno (2 word)	MBSL (DBSL)	*L (/L)
Moltiplicazione (Divisione) Binaria senza segno (2 word)	--	*UL (/UL)

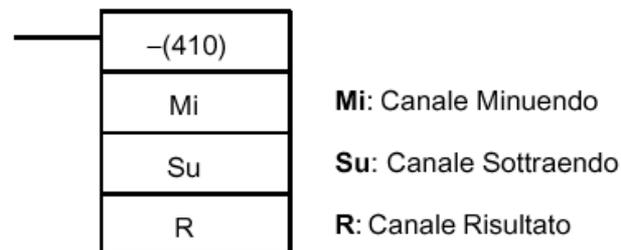
L'istruzione di somma (+)

- L'istruzione di somma + (400) esegue la somma binaria con segno tra due dati a 16 bit (canali e/o costanti)
- Il bit di Carry NON viene sommato al risultato
- Gli operandi dell'istruzione sono tre:
 - Au e Ad: Addendi. Are consentite: CIO, W, H, A (da A0 ad A959), T, C, D, E, #, IR (da IR0 a IR15)
 - R: Risultato. Are consentite: CIO, W, H, A (da A448 ad A959), T, C, D, E, #, IR (da IR0 a IR15)



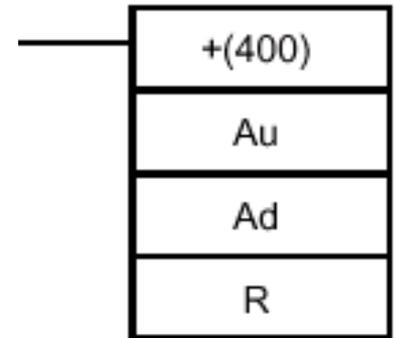
L'istruzione di sottrazione (-)

- L'istruzione di sottrazione - (410) esegue la sottrazione binaria con segno tra due dati a 16 bit (canali e/o costanti)
- Il bit di Carry NON viene sottratto al risultato
- Gli operandi dell'istruzione sono tre:
 - Mi e Su: Minuendo e sottraendo. Arete consentite: CIO, W, H, A (da A0 ad A959), T, C, D, E, #, IR (da IR0 a IR15)
 - R: Risultato. Arete consentite: CIO, W, H, A (da A448 ad A959), T, C, D, E, #, IR (da IR0 a IR15)



Uso delle costanti numeriche - esempi

SOMMA BINARIA				RISULTATO
Nome istruzione	Canale augendo	Canale addendo	Canale risultato	
+(400)	&10	&10	D0	D0 = 20 (14 hex)
+(400)	&10	#10	D1	D1 = 26 (1A hex)
+(400)	&10	+10	D2	D2 = 20 (14 hex)
+(400)	#10	#10	D3	D3 = 32 (20 hex)
+(400)	#10	+10	D4	D4 = 26 (1A hex)
+(400)	+10	+10	D5	D5 = 20 (14 hex)
+(400)	-10	+10	D6	D6 = 0 (0 hex)
+(400)	-10	-10	D7	D7 = -20 (FFEC hex)



Au: Canale Augendo

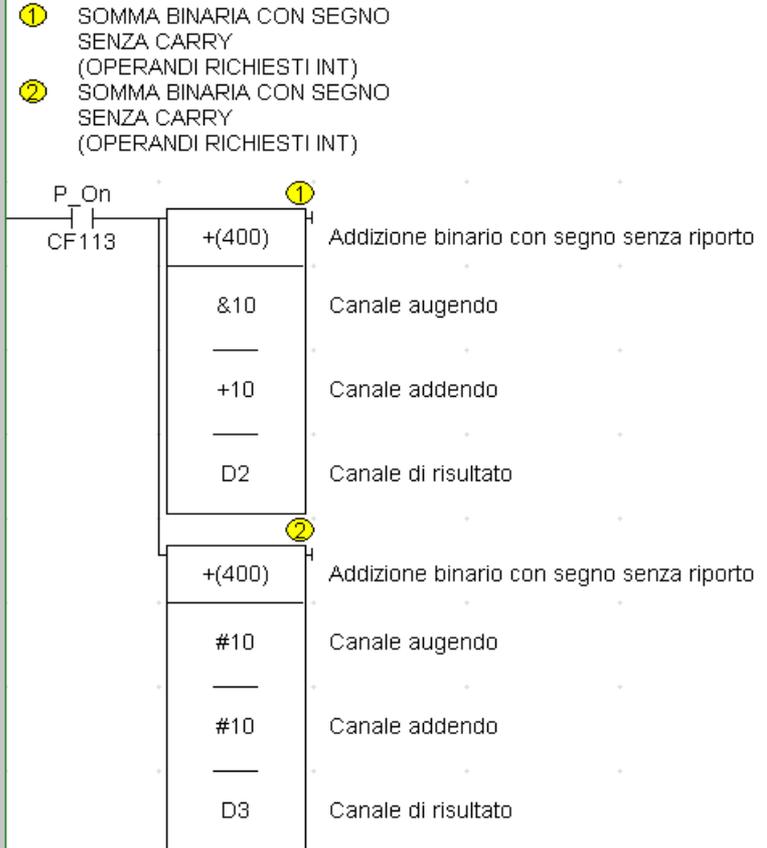
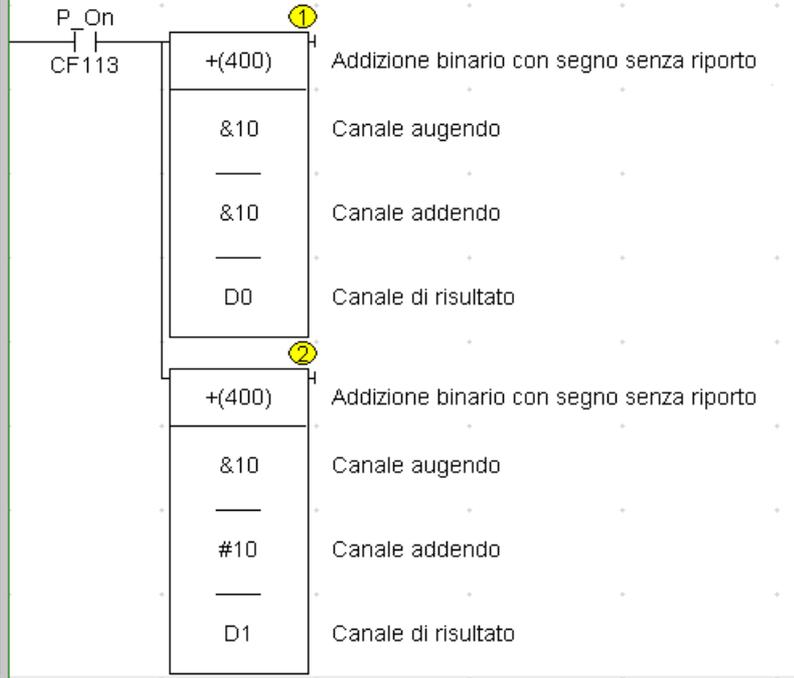
Ad: Canale Addendo

R: Canale Risultato

Costanti numeriche: esempio

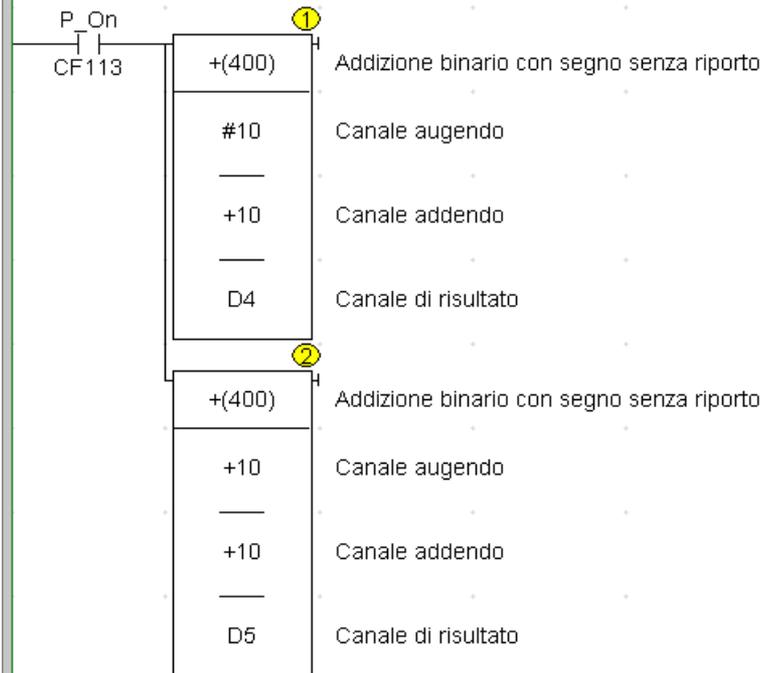
**PROVE DI INSERIMENTO DI COSTANTI CON & E #
NOTARE LE DIFFERENTI INTERPRETAZIONI IN BASE
AL TIPO DI ISTRUZIONE CONSIDERATA**

- ① SOMMA BINARIA CON SEGNO SENZA CARRY (OPERANDI RICHIESTI INT)
- ② SOMMA BINARIA CON SEGNO SENZA CARRY (OPERANDI RICHIESTI INT)



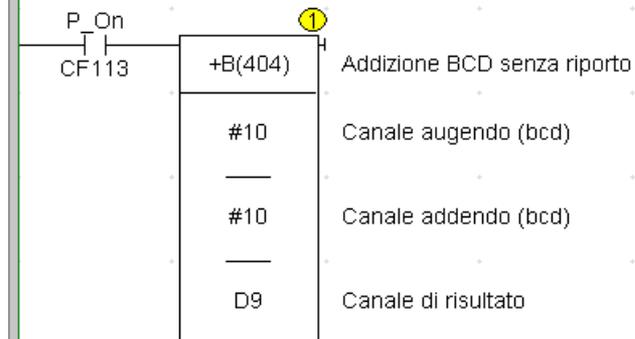
Costanti numeriche: esempio

- ① SOMMA BINARIA CON SEGNO SENZA CARRY (OPERANDI RICHIESTI INT)
- ② SOMMA BINARIA CON SEGNO SENZA CARRY (OPERANDI RICHIESTI INT)



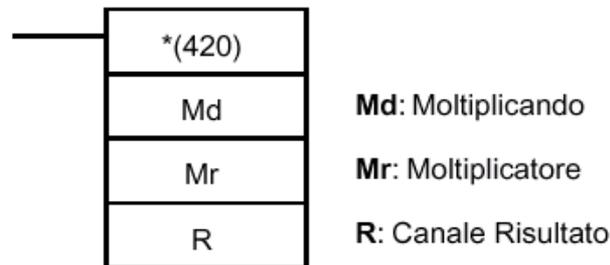
Somma di numeri BCD

- ① SOMMA BCD SENZA CARRY (OPERANDI RICHIESTI UINT_BCD)



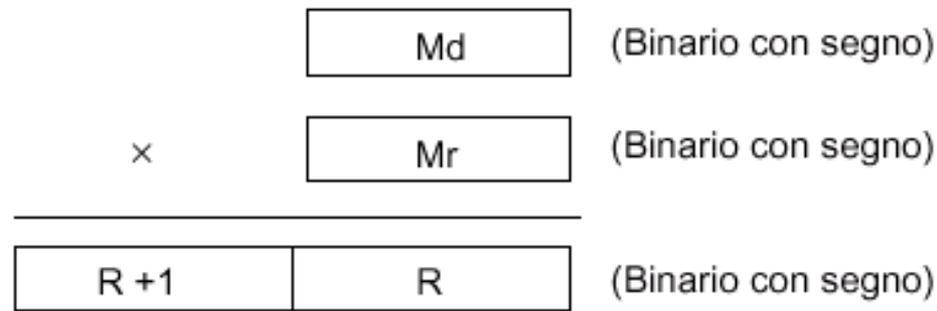
L'istruzione di moltiplicazione (*)

- L'istruzione di moltiplicazione *(420) esegue la moltiplicazione tra due dati binari con segno a 16 bit
- Gli operandi dell'istruzione sono tre:
 - Md e Mr: Moltiplicando e Moltiplicatore. Aree consentite: CIO, W, H, A (da A0 ad A959), T, C, D, E, #, IR (da IR0 a IR15)
 - R: Risultato. Aree consentite: CIO, W, H, A (da A448 ad A959), T, C, D, E, #, IR (da IR0 a IR15)



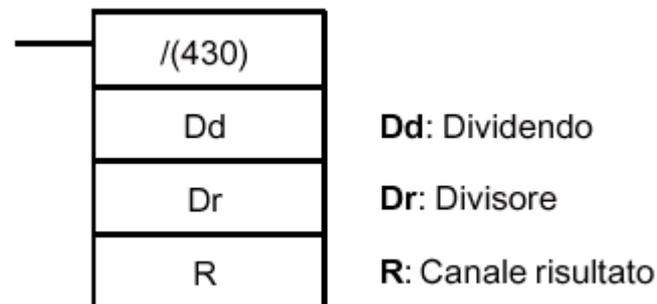
L'istruzione di moltiplicazione (*)

- Il risultato della moltiplicazione verrà posto in due canali consecutivi
- Il canale specificato nell'istruzione come terzo parametro indica il canale meno significativo in cui verrà trasferito il risultato
- Se il risultato della moltiplicazione è pari a 0 si alza il flag di uguale P_EQ



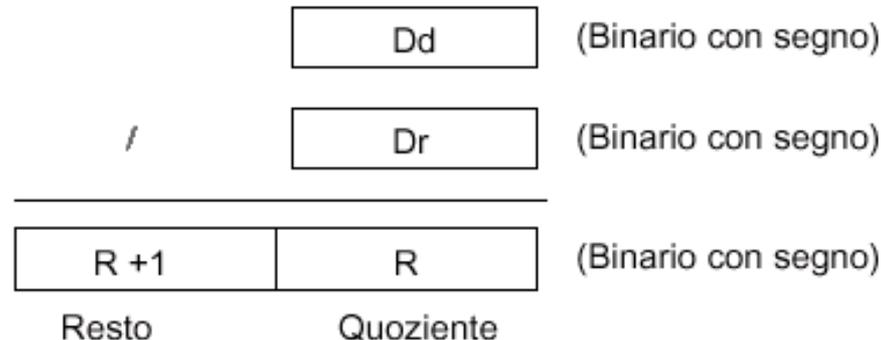
L'istruzione di divisione (/)

- L'istruzione di divisione /(430) esegue la divisione tra due dati binari con segno a 16 bit
- Gli operandi dell'istruzione sono tre:
 - Dd e Dr: Dividendo e Divisore. Aree consentite: CIO, W, H, A (da A0 ad A959), T, C, D, E, #, IR (da IR0 a IR15)
 - R: Risultato. Aree consentite: CIO, W, H, A (da A448 ad A959), T, C, D, E, #, IR (da IR0 a IR15)



L'istruzione di divisione (/)

- Il risultato della divisione verrà posto in due canali consecutivi e suddiviso in quoziente e resto
- Il canale successivo a quello specificato nell'istruzione come terzo parametro viene riservato per il resto
- Se il risultato della divisione è pari a 0 si alza il flag di uguale P_EQ
- La divisione per zero non viene eseguita e si alza il flag di errore P_ER



Esercizio 10: linee di produzione

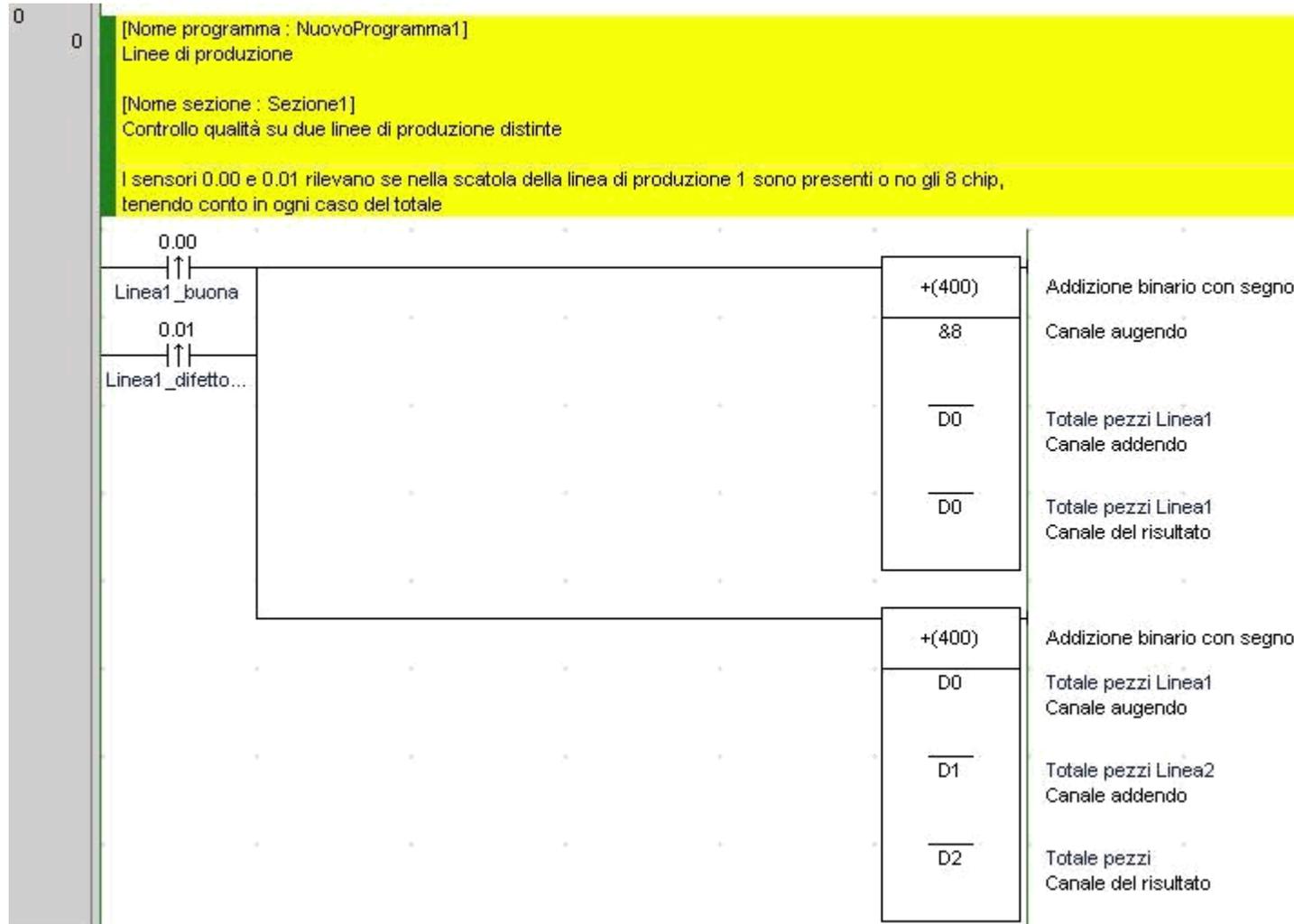
- Due sensori rilevano scatole contenenti chip difettosi provenienti da due linee di produzione
- Sul primo nastro ogni scatola contiene 8 chip; sul secondo nastro ogni scatola contiene 15 chip
- Il controllo qualità dovrà essere effettuato considerando l'intera scatola.
- Nel caso venga rilevata una scatola difettosa, l'intera confezione verrà eliminata.

Esercizio 10: linee di produzione

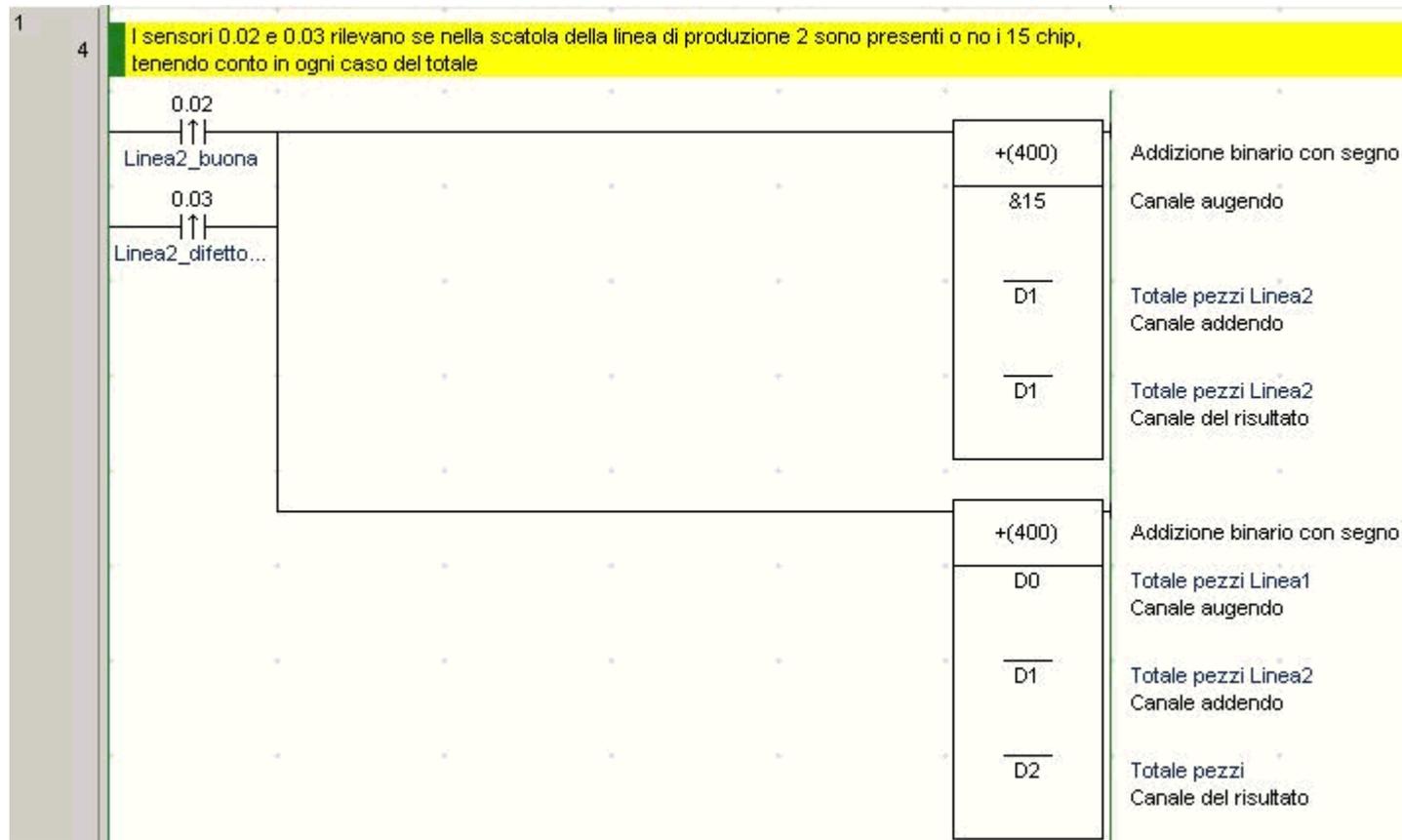
- Calcolare la percentuale dei pezzi difettosi sul totale dei pezzi prodotti e prevedere un tasto di reset che azzeri tutti i conteggi

	CHANNEL	D7	Lavoro	Percentuale pezzi difettosi
	CHANNEL	D5	Lavoro	Totale difettosi
	CHANNEL	D4	Lavoro	Totale difettosi Linea2
	CHANNEL	D3	Lavoro	Totale difettosi Linea1
	CHANNEL	D2	Lavoro	Totale pezzi
	CHANNEL	D1	Lavoro	Totale pezzi Linea2
	CHANNEL	D0	Lavoro	Totale pezzi Linea1
	BOOL	0.04	Lavoro	Reset
	BOOL	0.03	Lavoro	Linea2_difettosa
	BOOL	0.02	Lavoro	Linea2_buona
	BOOL	0.01	Lavoro	Linea1_difettosa
	BOOL	0.00	Lavoro	Linea1_buona

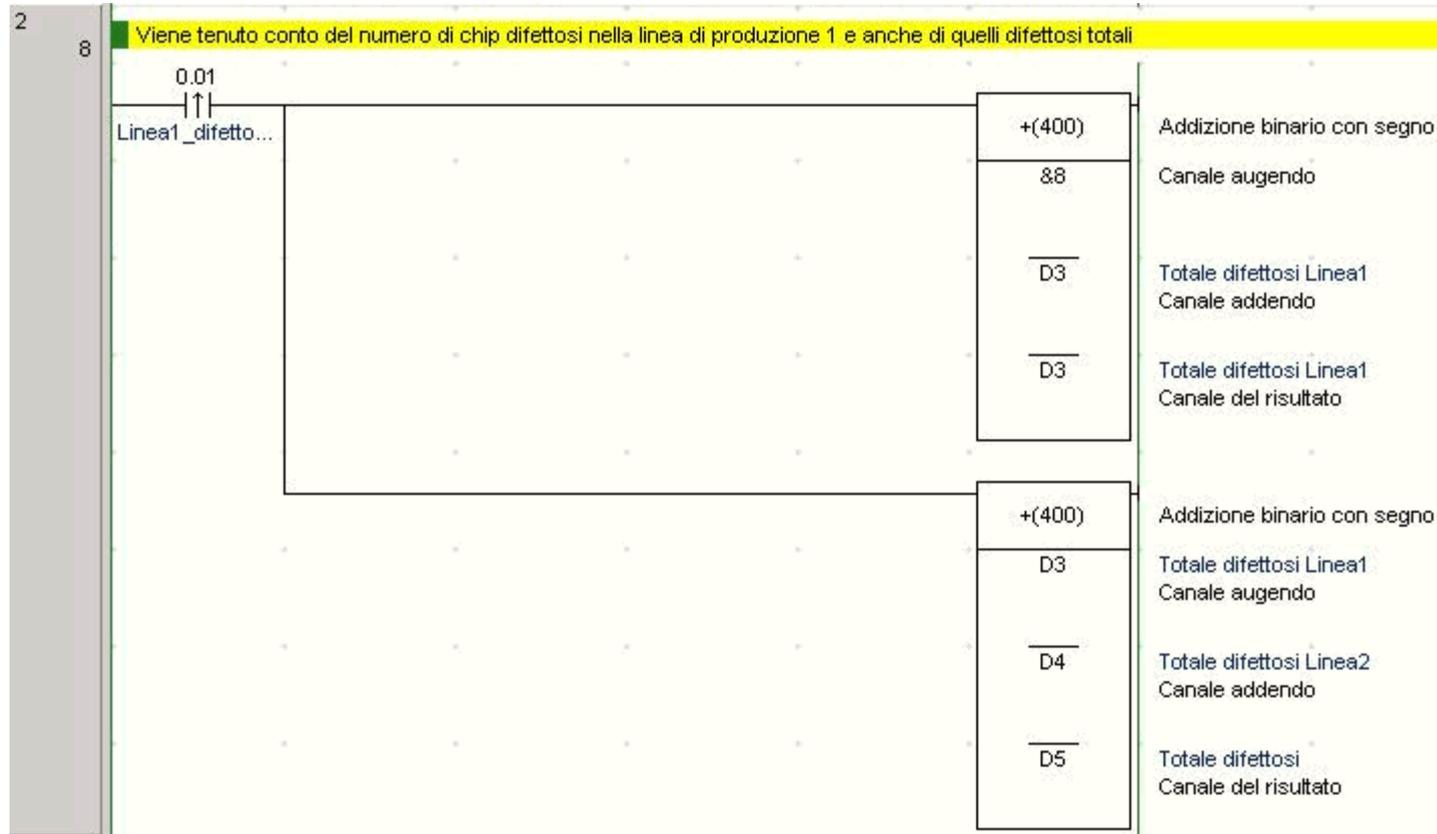
Esercizio 10: soluzione



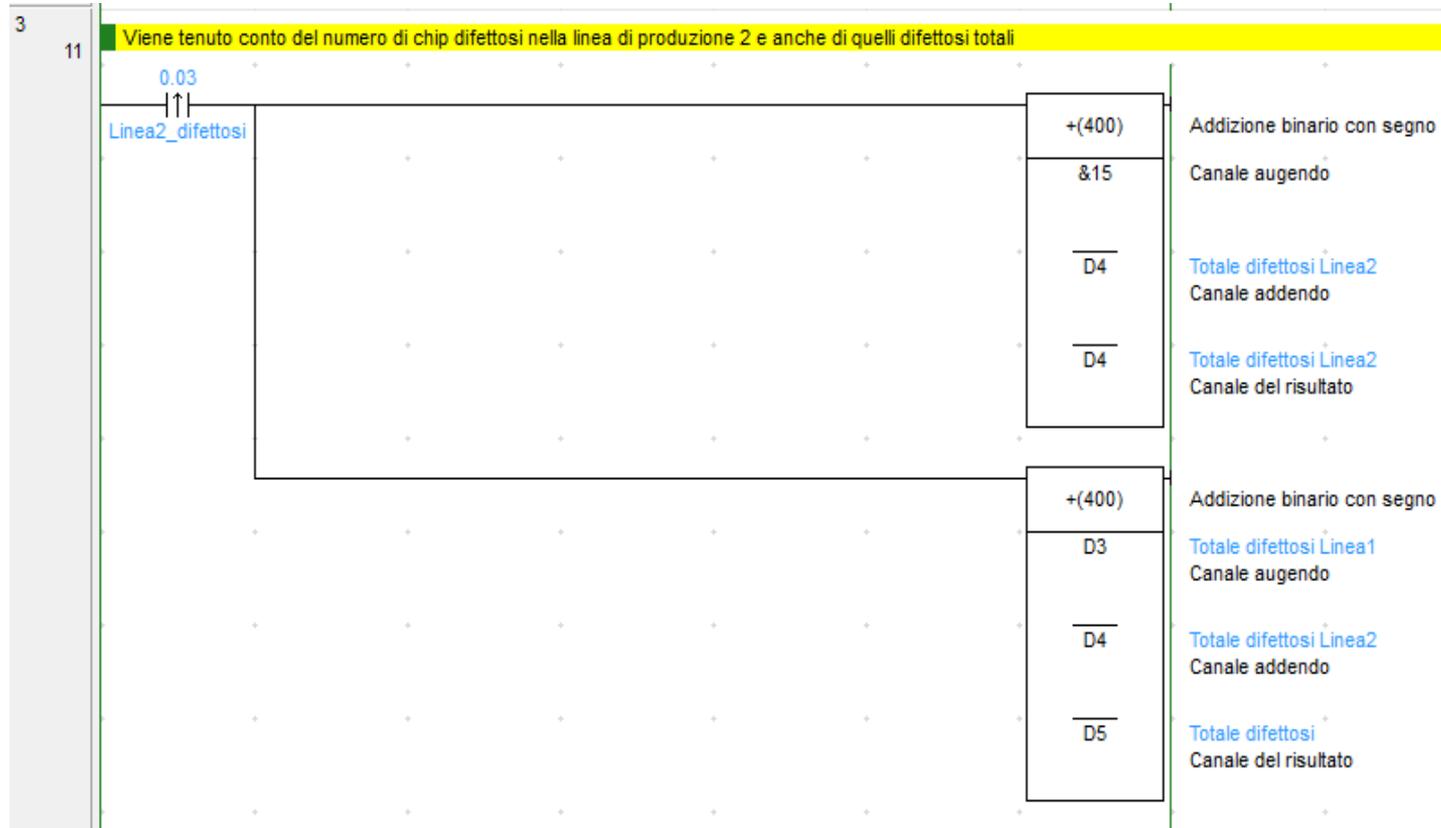
Esercizio 10: soluzione



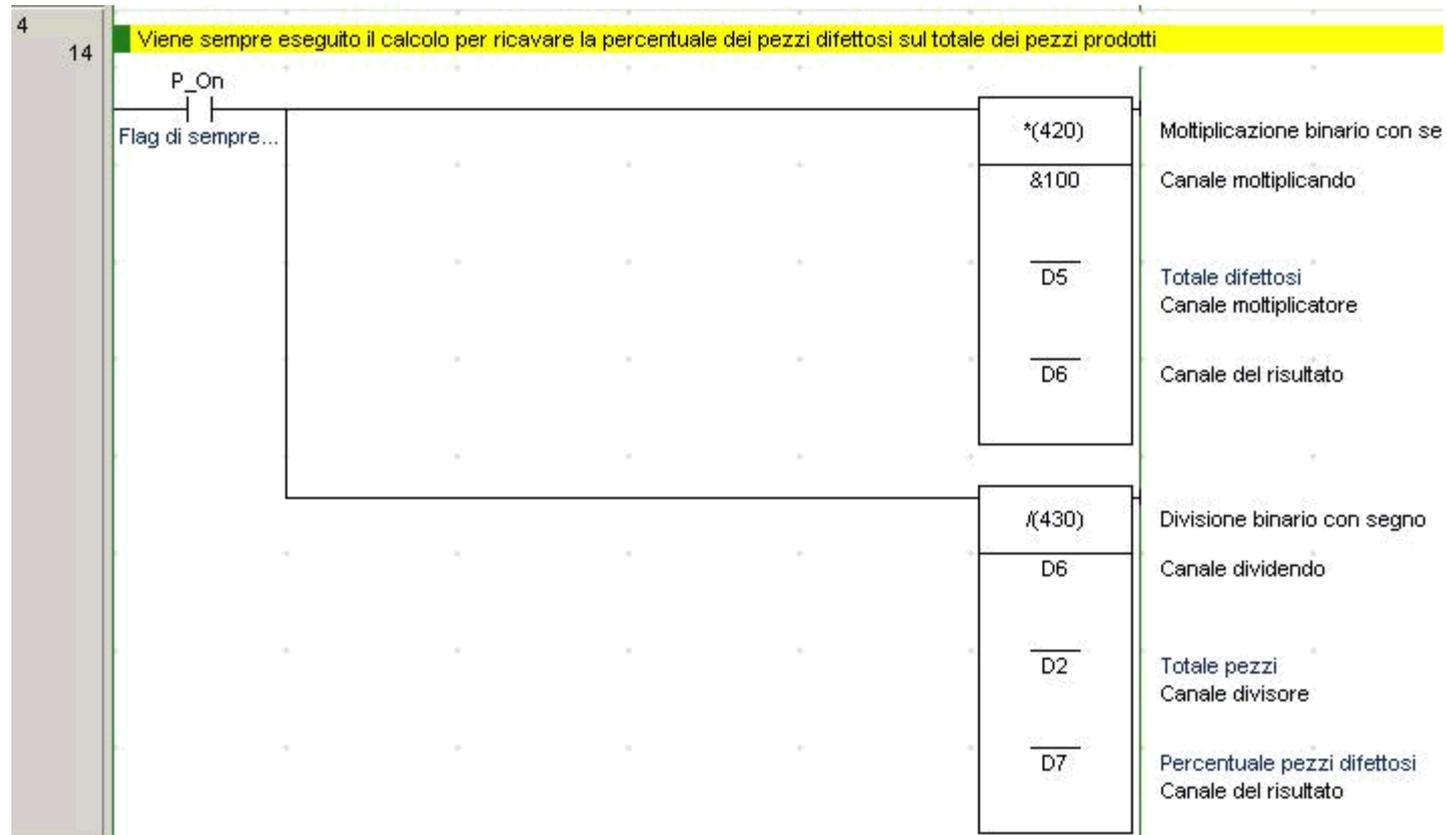
Esercizio 10: soluzione



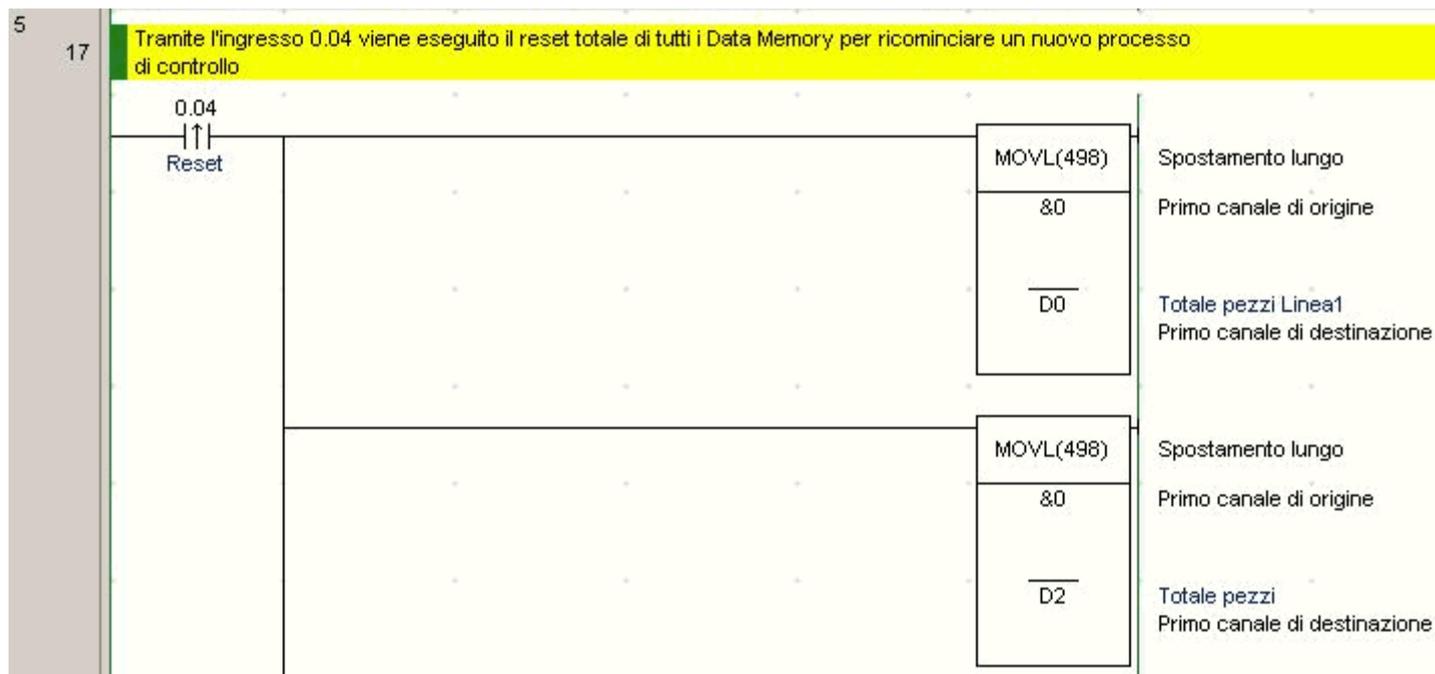
Esercizio 10: soluzione



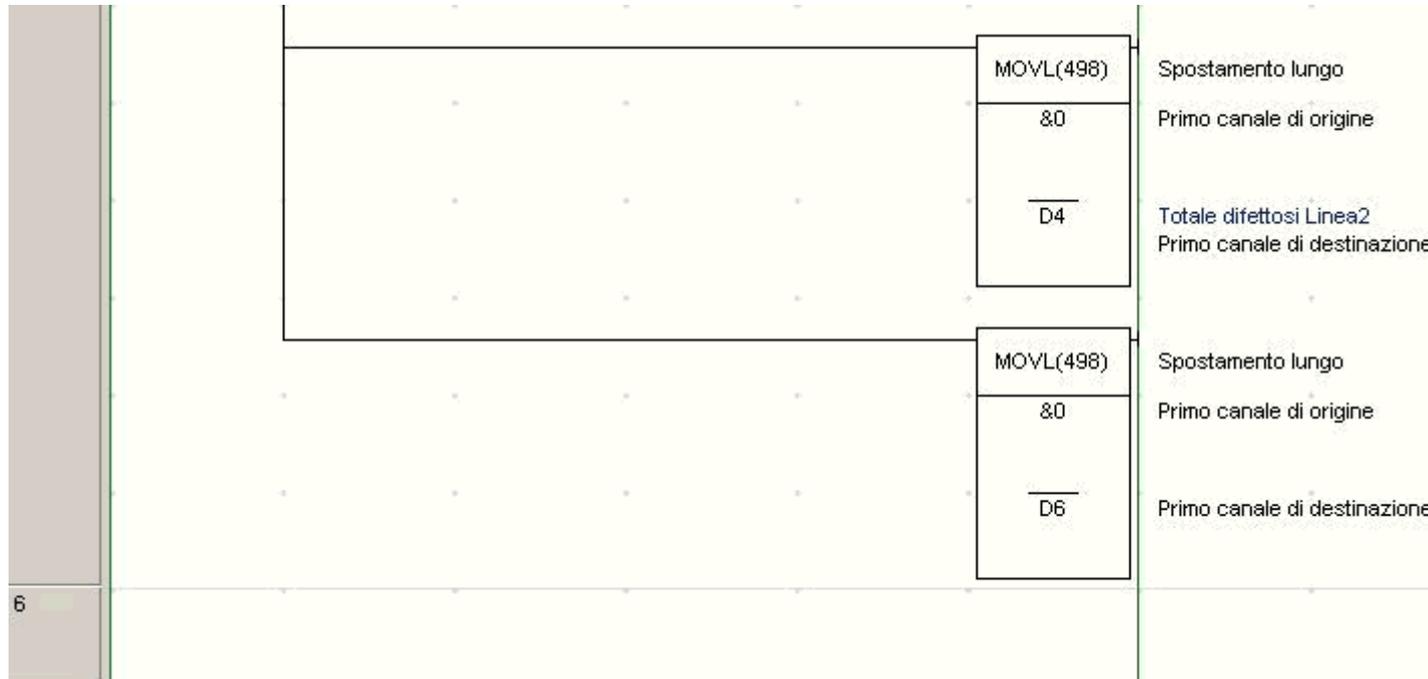
Esercizio 10: soluzione



Esercizio 10: soluzione



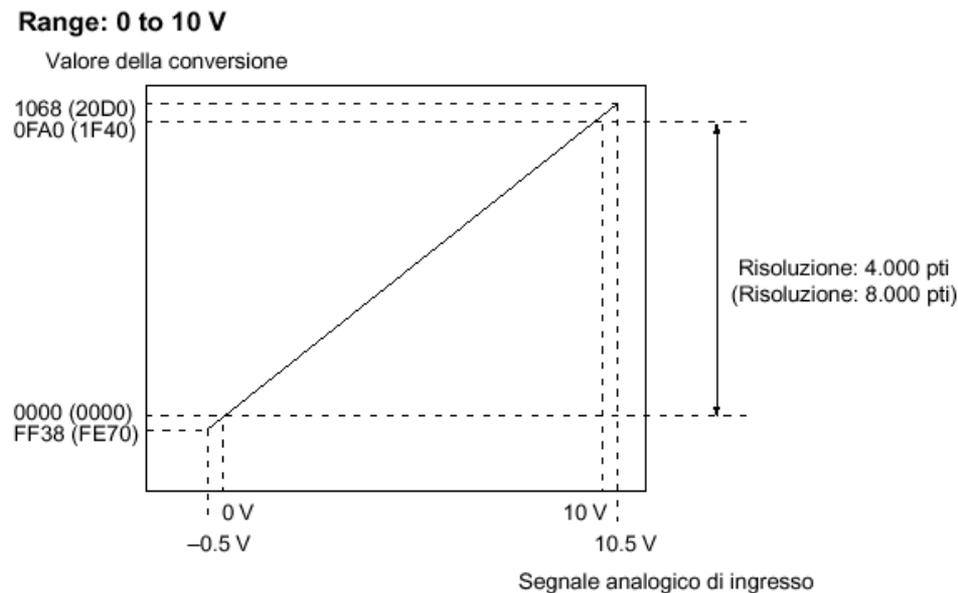
Esercizio 10: soluzione



La gestione dei segnali analogici

Utilizzo dei segnali analogici

- Nei PLC i segnali analogici vengono gestiti tramite schede di conversione A/D (ingressi) e D/A (uscite) che li traducono in valori numerici esadecimali secondo precise caratteristiche di conversione, proprie del tipo di scheda utilizzata
- Esempio:



Utilizzo dei segnali analogici

- Alcuni PLC compatti della famiglia CP1, gestiscono questa tipologia di segnali tramite l'utilizzo di analogiche integrate e/o di moduli opzionali (ad esempio: CP1W-MAD11 / -MAB221)



Utilizzo dei segnali analogici

- Modelli supportati:

Codice scheda	CP1E				CP1L					CP1H	
	-E	-N	-E_S / -N_S1	-NA20	-L10	-L14/20	-M30/40/60	-EL20	-EM30/40	-X	-XA
<i>Trimmer di regolazione</i>	2	2	○	2	1	1	1	○	○	1	1
<i>Analogiche integrate</i>	○	○	○	2IA / 1UA (6000 ris.)	1 IA (8 bit)	1 IA (8 bit)	1 IA (8 bit)	2 IA (10 bit)	2 IA (10 bit)	1 IA (8 bit)	4IA / 2UA (12000 ris.)
<i>CP1W-ADB21</i>	○	●**	○	●*	○	○	○	●	●●	○	○
<i>CP1W-DAB21V</i>	○	●**	○	●*	○	○	○	●	●●	○	○
<i>CP1W-MAB221</i>	○	●**	○	●*	○	○	○	●	●●	○	○

○ = Non supportata

● = Supportata (max 1 collegabile)

●● = Supportata (max 2 collegabili)

* A partire dalla versione firmware 1.2

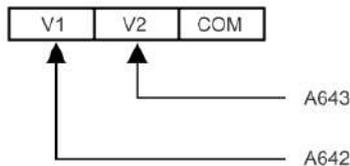
** A partire dalla versione firmware 1.2 solo per CPU da 30/40/60 punti

IA = Ingresso analogico

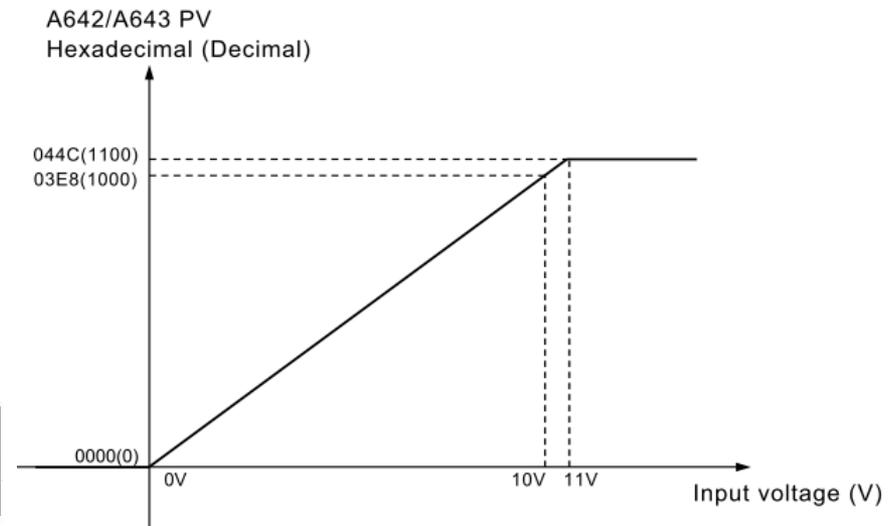
UA = Uscita analogica

CP1L-E: ingressi analogici integrati

- I modelli CP1L-EL / -EM sono equipaggiati di 2 ingressi analogici integrati. Questi ingressi hanno una risoluzione di 1000 punti ed un indirizzamento fisso (canali: A642 per l'ingresso 1 e A643 per l'ingresso 2)
- Il segnale accettato in ingresso è un valore in tensione compreso tra 0 e 10 V convertito quindi in un valore lineare compreso tra 0000 e 03E8 Hex (0...1000 decimale)



V1	Voltage Input 1
V2	Voltage Input 2
COM	Input Common



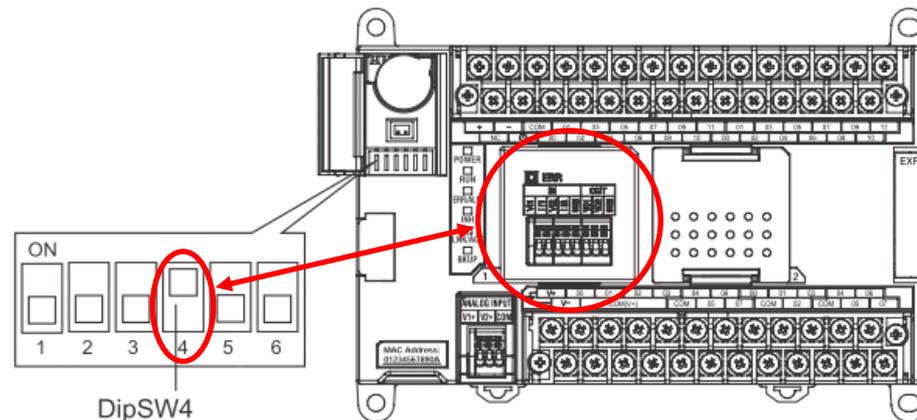
CP1L-E: schede analogiche opzionali

- Essendo privo di uscite analogiche integrate, è possibile montare delle schede analogiche opzionali. Attualmente sono disponibili i seguenti moduli:
 - CP1W-DAB21V: 2 uscite analogiche, range 0-10 V
 - CP1W-MAB221: 2 ingressi analogici range 0-10 V, 0-20 mA e 2 uscite 0-10 V)

Analog Option Board		Voltage Input 0V~10V (Resolution:1/4000)	Current Input 0mA~20mA (Resolution:1/2000)	Voltage Output 0V~10V (Resolution:1/4000)
Analog I/O Option Board	CP1W-MAB221	2CH		2CH
Analog Input Option Board	CP1W-ADB21	2CH		---
Analog Output Option Board	CP1W-DAB21V	---		2CH

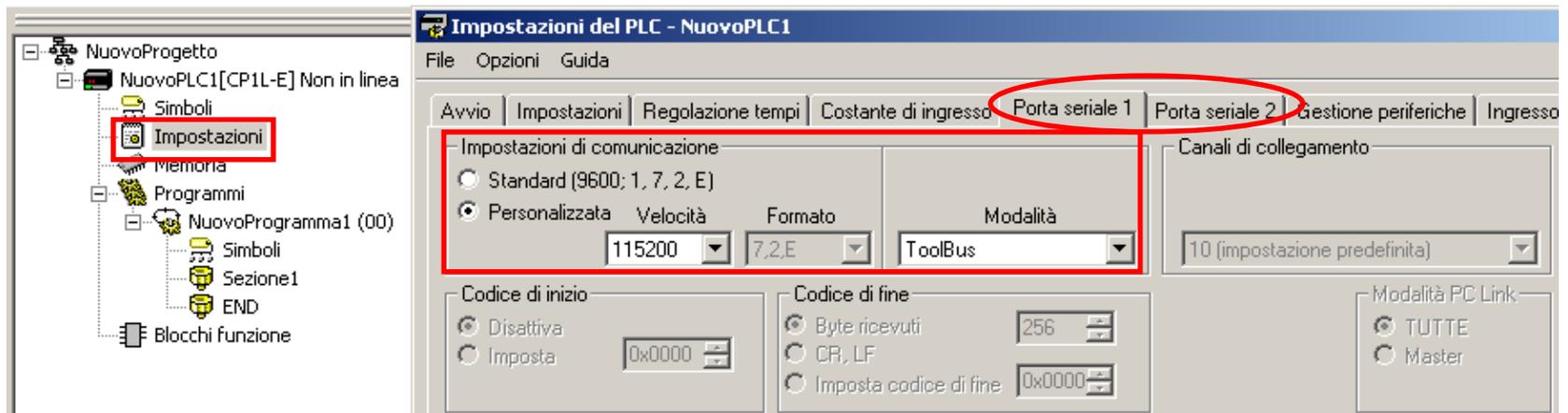
CP1L-E: schede analogiche opzionali - configurazione

- Prima di utilizzare queste schede opzionali è necessario abilitarle tramite **uno** di questi due metodi:
 1. Dip-switch della CPU
 2. Settaggio tramite impostazioni del PLC
- Metodo 1:
 - A seconda della posizione di montaggio della scheda opzionale, sarà necessario abilitare il dip-switch 4 (slot 1) e/o il dip-switch 5 (slot 2, ove possibile) a PLC spento.



CP1L-E: schede analogiche opzionali - configurazione

- Metodo 2:
 - Dalla finestra “spazio di lavoro” del CX-Programmer, selezionare la voce “Impostazioni”. Qui sarà possibile abilitare le schede opzionali selezionando, in base alla posizione di montaggio, l’etichetta “Porta seriale 1” (slot 1) e/o “Porta seriale 2” (slot 2)
 - Inserire quindi la seguente configurazione:



- Trasferire quindi al PLC, infine spegnere e riaccendere.
- Per CP1E-N

CP1L-E: schede analogiche opzionali - indirizzamenti

- Le schede analogiche opzionali sono provviste della seguente locazione fissa dipendente dalla posizione di montaggio:

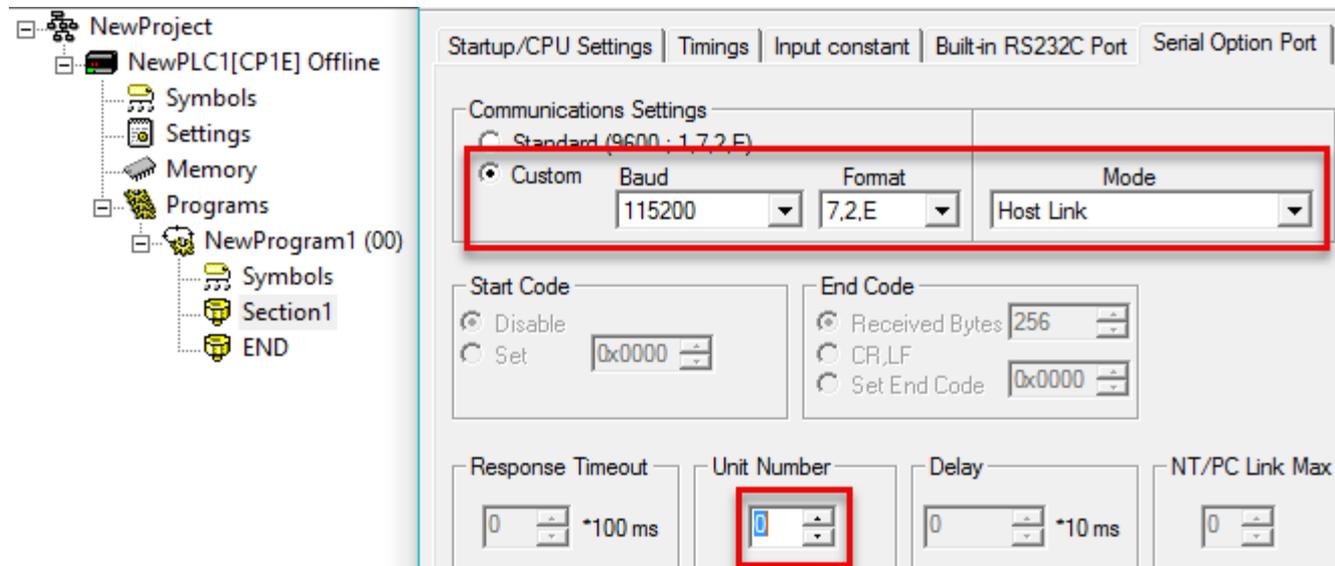
CP1L-EL/EM series PLC

I/O Capacity	Option Port	Beginning Channel (m)	Channel Range
20	Port 1	CIO2990	CIO2990 ~ CIO2999
30/40	Port 1 (Left)	CIO2980	CIO2980 ~ CIO2989
	Port 2 (Right)	CIO2990	CIO2990 ~ CIO2999

Channel	Contents		
	CP1W-ADB21	CP1W-DAB21V	CP1W-MAB221
m	Analog Input 1	---	Analog Input 1
m+1	Analog input 2	---	Analog input 2
m+2 to m+4	---	---	---
m+5	---	Analog Output 1	Analog Output 1
m+6	---	Analog Output 2	Analog Output 2
m+7 to m+9	---	---	---

CP1E-N30/40/60: schede analogiche opzionali - configurazione

- Metodo 2:
 - Dalla finestra “spazio di lavoro” del CX-Programmer, selezionare la voce “Impostazioni”. Qui sarà possibile abilitare le schede opzionali selezionando Porta Seriale Opzionale.
 - Inserire quindi la seguente configurazione:



- Trasferire quindi al PLC, infine spegnere e riaccendere.

CP1E-N30/40/60: schede analogiche opzionali - indirizzamenti

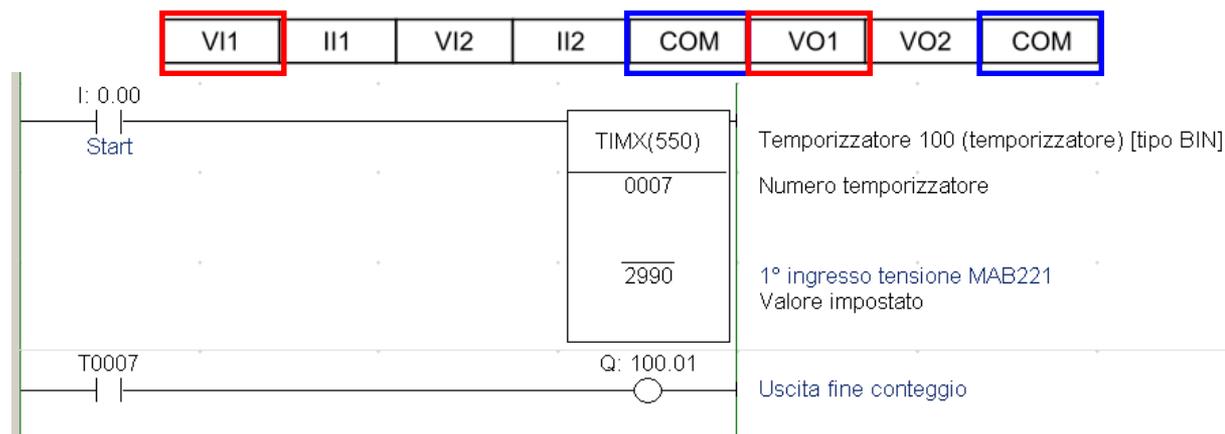
- Le schede analogiche opzionali sono provviste della seguente locazione fissa:

Channel	Contents		
	CP1W-ADB21	CP1W-DAB21V	CP1W-MAB221
CIO80	Analog Input 1	---	Analog Input 1
CIO81	Analog input 2	---	Analog input 2
CIO82 to CIO84	---	---	---
CIO85	---	Analog Output 1	Analog Output 1
CIO86	---	Analog Output 2	Analog Output 2
CIO87 to CIO89	---	---	---

CP1L-E: schede analogiche opzionali - esempio

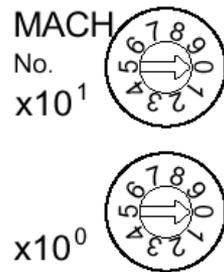
- Tramite, ad esempio, la scheda opzionale CP1W-MAB221 montata nello slot 1, è possibile impostare un valore di set point al Timer (TIMX) con un range variabile da 0 s a 400.0 s (0...4000)
- Utilizzare l'ingresso analogico 1 e l'uscita analogica 1

Terminal Arrangement for CP1W-MAB221



Configurazione di un modulo analogico modulare

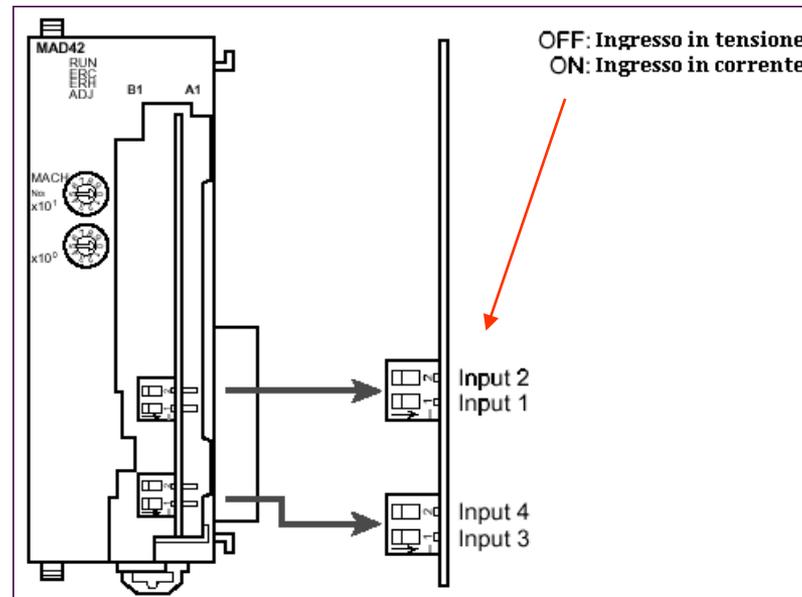
- I moduli analogici, essendo moduli speciali, sono provvisti di selettori rotativi per stabilire la locazione dei canali occupati:



Impostaz. selettore	Numero di modulo	Indirizzamento area CIO	Indirizzamento area DM
0	Unit #0	da CIO 2000 a CIO 2009	da D20000 a D20099
1	Unit #1	da CIO 2010 a CIO 2019	da D20100 a D20199
2	Unit #2	da CIO 2020 a CIO 2029	da D20200 a D20299
3	Unit #3	da CIO 2030 a CIO 2039	da D20300 a D20399
4	Unit #4	da CIO 2040 a CIO 2049	da D20400 a D20499
5	Unit #5	da CIO 2050 a CIO 2059	da D20500 a D20599
6	Unit #6	da CIO 2060 a CIO 2069	da D20600 a D20699
7	Unit #7	da CIO 2070 a CIO 2079	da D20700 a D20799
8	Unit #8	da CIO 2080 a CIO 2089	da D20800 a D20899
9	Unit #9	da CIO 2090 a CIO 2099	da D20900 a D20999
10	Unit #10	da CIO 2100 a CIO 2109	da D21000 a D21099
~	~	~	~
n	Unit #n	da CIO 2000 + (n x 10) a CIO 2000 + (n x 10) + 9	da D20000 + (n x 100) a D20000 + (n x 100) + 99
~	~	~	~
95	Unit #95	da CIO 2950 a CIO 2959	D29500 a D29599

Configurazione di un modulo analogico

- La configurazione del modulo CJ1W-MAD42 avviene sia via hardware, tramite appositi dip-switches, che via software nei relativi canali dell'area DM
- Configurazione dei Dip-Switches:



Configurazione di un modulo analogico

- Impostazioni principali della scheda (area Data Memory)

DM word	Bits																			
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
D(m)	Ratio conversion use setting								Input use setting				Output use setting							
	Not used.				Not used.				Loop 2		Loop 1		Input 4	Input 3	Input 2	Input 1	Not used.		Not used.	Out-put 2
D(m+1)	Input signal range setting								Output signal range setting											
	Input 4				Input 3				Input 2		Input 1		Not used.		Not used.		Output 2		Output 1	

Input signal range

00:	-10 to 10 V
01:	0 to 10 V
10:	1 to 5 V/4 to 20 mA
11:	0 to 5 V

Use setting

0:	Not used.
1:	Used.

Output signal range

00:	-10 to 10 V
01:	0 to 10 V
10:	1 to 5 V
11:	0 to 5 V

DM word	Bits															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
D(m+18)	Conversion time/resolution setting								Operation mode setting							
	00: 8000 punti C1: 4000 punti								00: Normal mode C1: Adjustment mode							

Configurazione di un modulo analogico

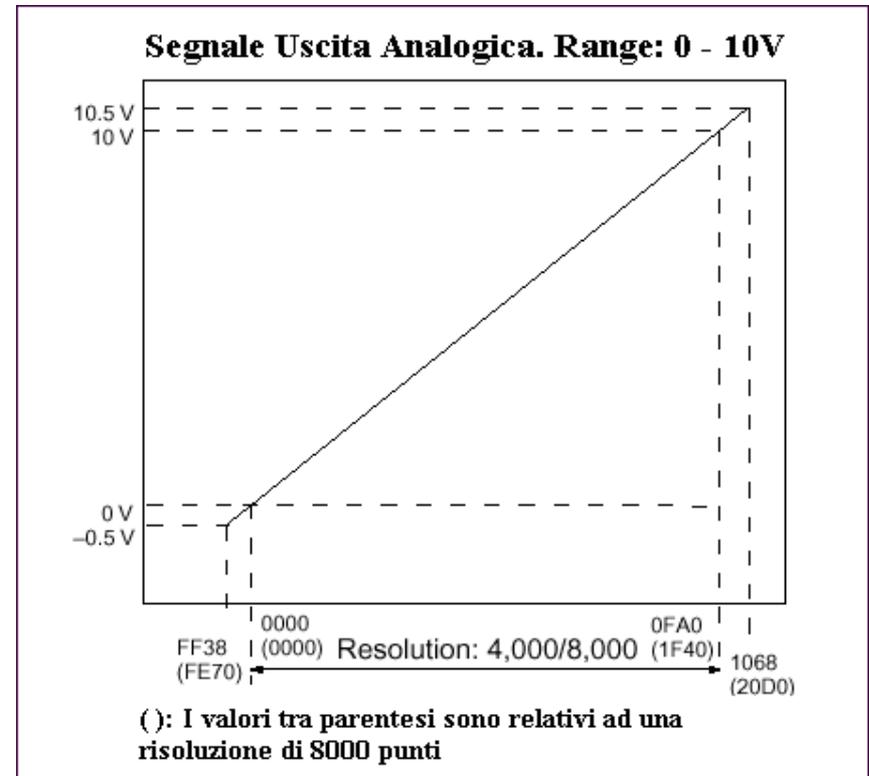
- Area CIO e cablaggio

I/O	Word	Bits																						
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0							
Output (CPU to Unit)	n	Not used.								Peak value hold				Not used.		Conversion enable								
						Input 4				Input 3				Input 2				Input 1		Out-put 2		Out-put 1		
	n + 1	Output 1 set value																						
		16 ³				16 ²				16 ¹				16 ⁰										
	n + 2	Output 2 set value																						
n + 3	Not used.																							
n + 4	Not used.																							
Input (Unit to CPU)	n + 5	Input 1 conversion value / Loop 1 calculation result																						
		16 ³				16 ²				16 ¹				16 ⁰										
	n + 6	Input 2 conversion value / Loop 2 calculation result																						
	n + 7	Input 3 conversion value																						
	n + 8	Input 4 conversion value																						
n + 9	Alarm Flags								Disconnection detection								Output setting error							
					Input 4				Input 3				Input 2				Input 1				Out-put 2		Out-put 1	

Voltage output 2 (+)	B1	A1	Voltage output 1 (+)
Output 2 (-)	B2	A2	Output 1 (-)
Current output 2 (+)	B3	A3	Current output 1 (+)
N.C.	B4	A4	N.C.
Input 2 (+)	B5	A5	Input 1 (+)
Input 2 (-)	B6	A6	Input 1 (-)
AG	B7	A7	AG
Input 4 (+)	B8	A8	Input 3 (+)
Input 4 (-)	B9	A9	Input 3 (-)

Comando uscita analogica

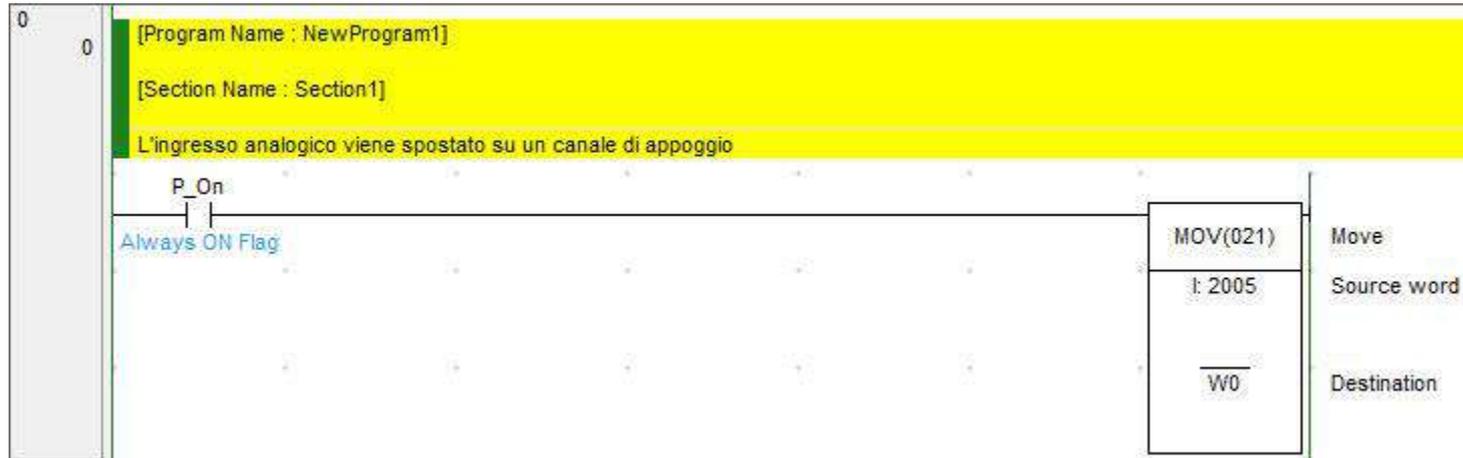
- L'uscita analogica della scheda CJ1W-MAD42 converte in modo lineare un valore esadecimale compreso tra 0000 e 0FA0 (0...4000 decimale) in un valore di tensione compreso tra 0 e 10 V.



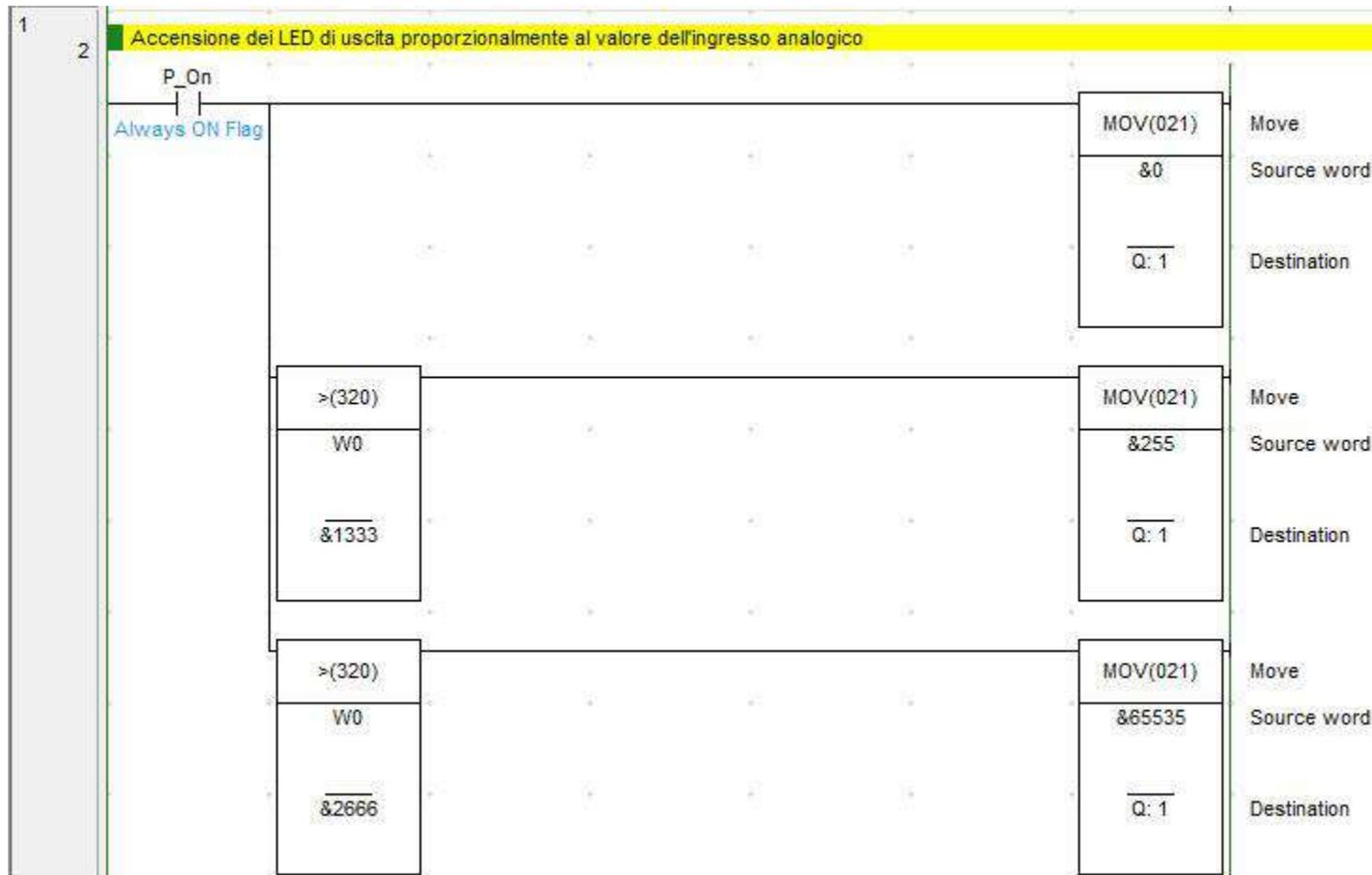
Esercizio 11: IN/OUT analogici – Parte 1

- Fare in modo che si accendano i LED relativi alle uscite digitali proporzionalmente al valore di un ingresso analogico dato da potenziometro.
- Suddividere il range di valori dell'ingresso analogico in tre intervalli; se il valore attuale dell'ingresso analogico è nel primo intervallo tutti i LED dovranno essere spenti, se è nel secondo la prima metà dei LED dovrà essere accesa, se è nel terzo tutti i LED dovranno essere tutti accesi.
- Ad esempio se, impostato l'ingresso analogico come 0-10V, acquisisco un segnale da 6V (convertito nel valore 2400) avrò i primi 8 LED di uscita accesi.

Esercizio 11: soluzione – Parte 1



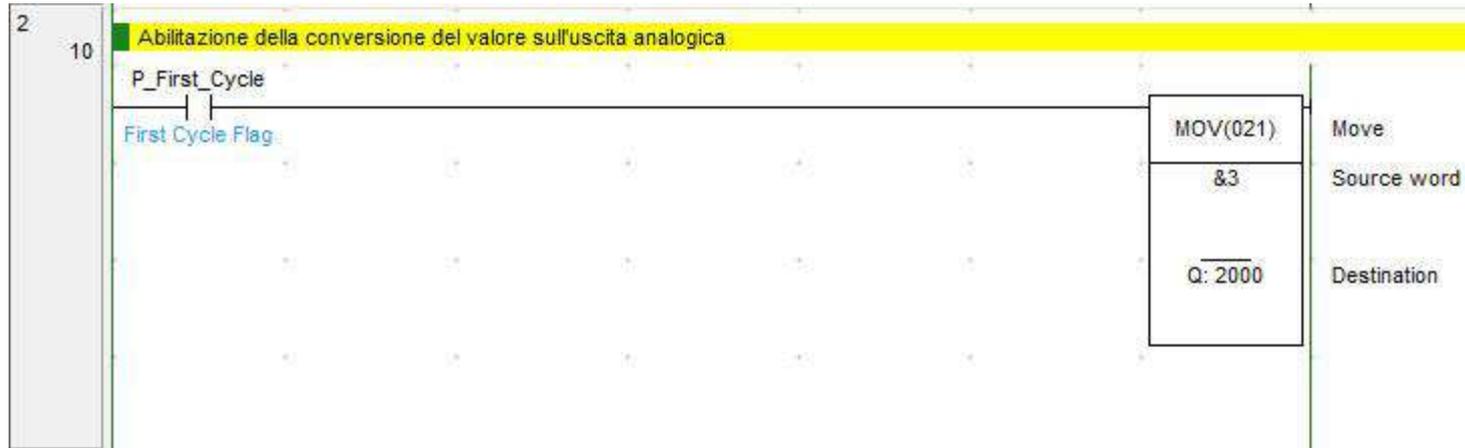
Esercizio 11: soluzione – Parte 1



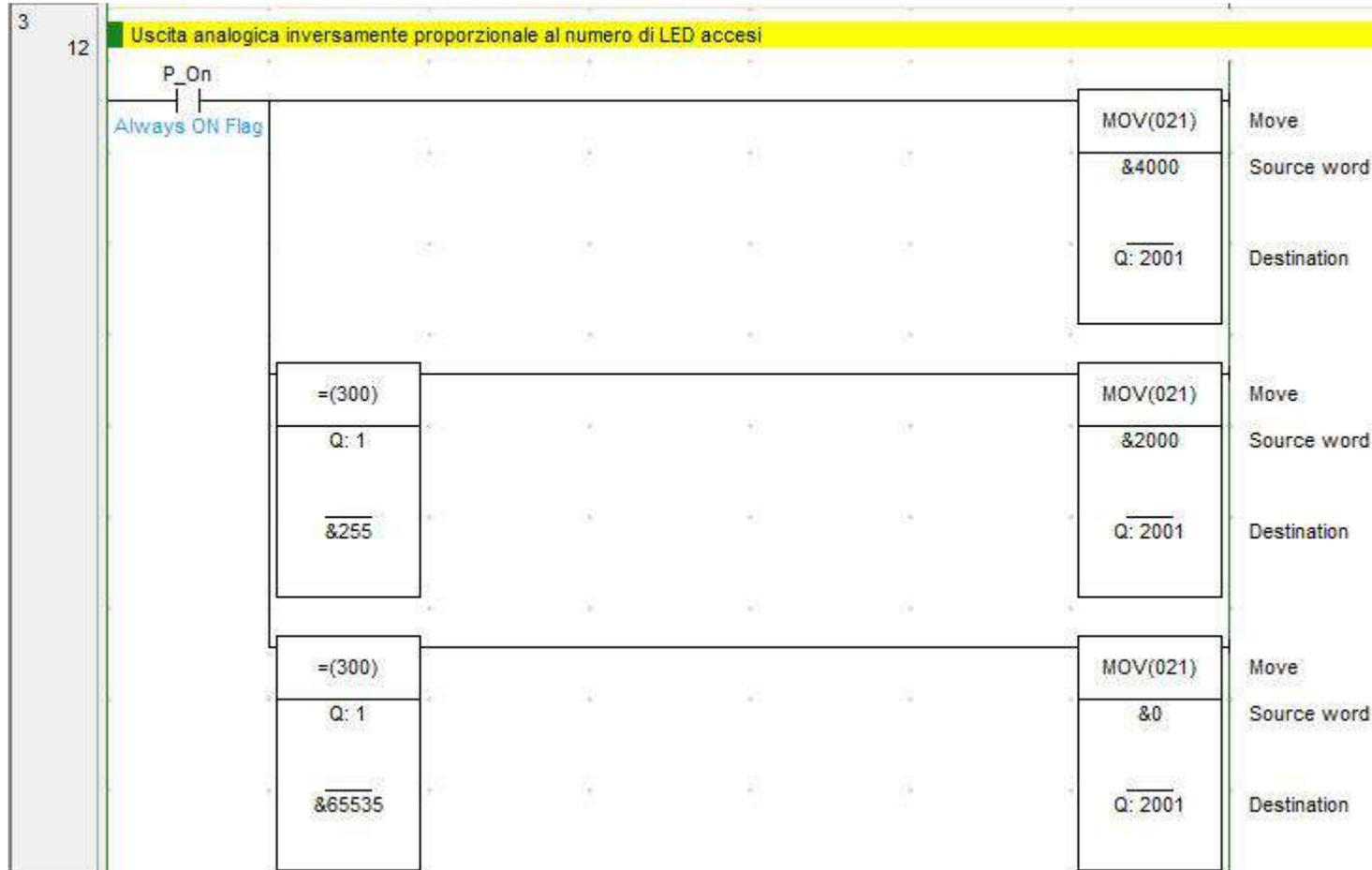
Esercizio 11: IN/OUT analogici – Parte 2

- Fare in modo che l'uscita analogica sia inversamente proporzionale al numero di LED accesi.
- Se i LED di uscita sono tutti spenti il valore dell'uscita analogica dovrà essere il massimo, se solo la prima metà dei LED è accesa il valore dell'uscita analogica dovrà essere la metà del valore massimo, se i LED sono tutti accesi il valore dell'uscita analogica dovrà essere il minimo.
- Ad esempio se, impostata l'uscita analogica come 0-10V, tutti i LED di uscita sono spenti si avrà il valore 4000 sul canale relativo all'uscita analogica che verrà convertito in 5V.

Esercizio 11: soluzione – Parte 2



Esercizio 11: soluzione – Parte 2



CX-Simulator



Introduzione

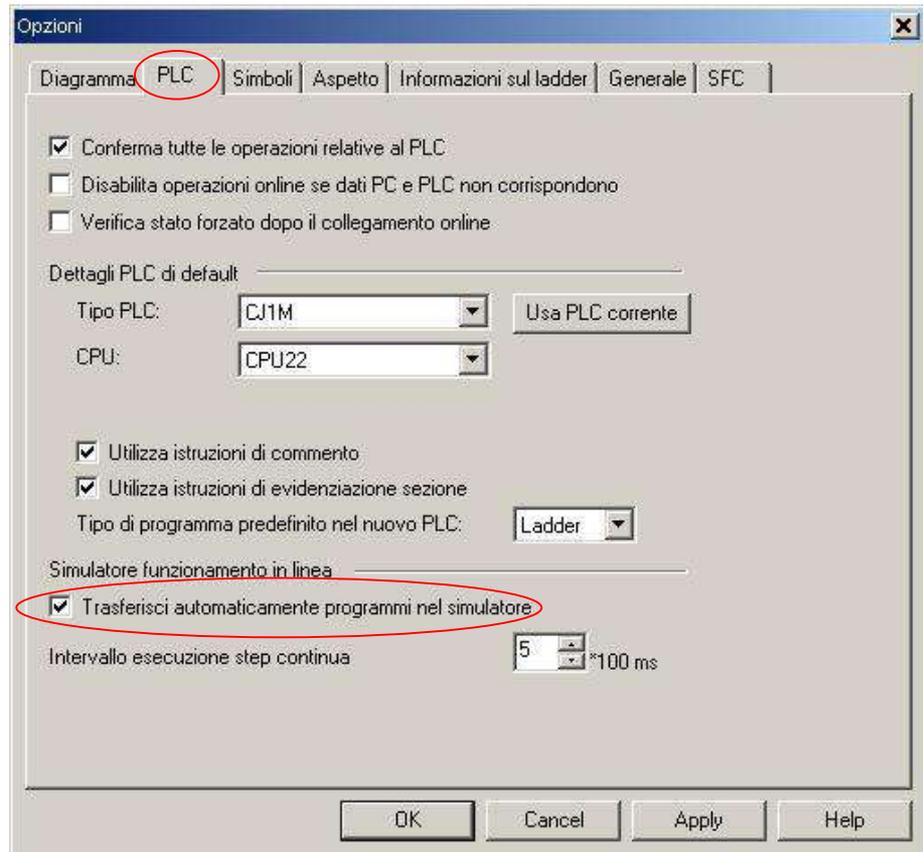
- Il CX-Simulator simula il funzionamento delle CPU serie CS/CJ e CP1
- Comprende strumenti e funzioni di debug che non sono disponibili nel PLC reale consentendo così una riduzione dei tempi e dei costi di sviluppo
- I programmi possono essere trasferiti tra CX-Simulator e CX-Programmer esattamente come avviene per un PLC fisico
- CX-Simulator può essere eseguito su Windows 2000, XP, Vista e 7

Avvio

- Esistono 2 metodi differenti per avviare CX-Simulator:
 - dal CX-Programmer selezionando dalla barra degli strumenti:
Simulazione → Simulatore funzionamento in linea
 - avviando separatamente CX-Simulator dal menù. Le impostazioni sono effettuate grazie al Setup guidato del PLC
- Quando si utilizza CX-Programmer v3.0 o superiore, CX-Simulator può essere lanciato da CX-Programmer.

1° metodo

- Selezionare Strumenti → Opzioni dalla barra degli strumenti di CX-Programmer
- Cliccare sull'etichetta PLC
- Selezionare il checkbox 'trasferisci automaticamente programmi nel simulatore' e cliccare OK
- Avviare il simulatore come descritto precedentemente o tramite l'icona 



1° metodo

- Dopo aver cliccato l'icona per la simulazione, apparirà una finestra di debug, indicando anche il n° dei conteggi di ciclo:



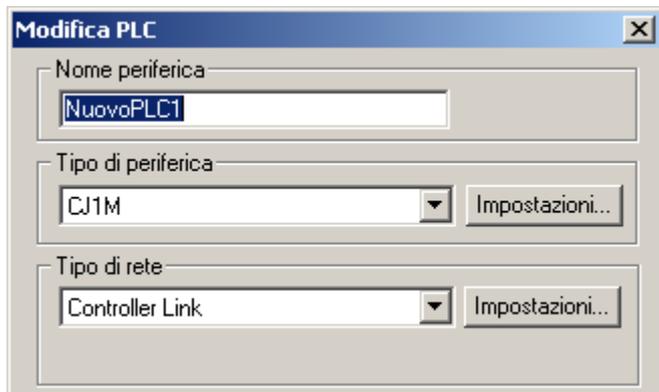
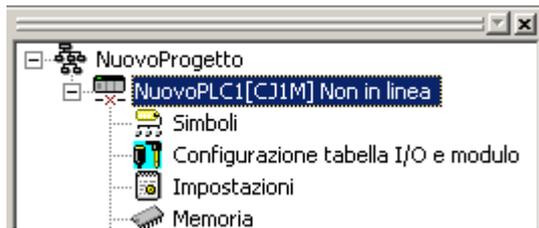
Nome	Funzione
Run	Esegue la scansione continua
Stop	Arresta l'esecuzione. La modalità di funzionamento diventerà PROGRAM automaticamente
Pause	Mette in pausa l'esecuzione. L'esecuzione può essere ripresa in una qualsiasi modalità di esecuzione
Step Run	Elabora ogni passo del codice mnemonico
Continuous Step Run	Esegue continuamente lo Step Run in un determinato intervallo
Scan Run	Esegue il ladder diagram per una sola scansione. Run viene eseguito durante lo Step Run, il programma verrà eseguito alla fine di quest'ultimo
Continuous Scan Run	Esegue continuamente lo Scan Run in un determinato intervallo
Scan Replay	Tornare allo stato della memoria I/O della scansione eseguita prima dello Step Run. Il conteggio del programma viene cancellato
Reset	Resetta l'hardware ed esegue la procedura di avvio
Show Step Run	Visualizza la finestra dello Step Run
Task Manage	Visualizza la finestra del Task Manage
I/O Break Condition Setting	Visualizza la finestra dell'I/O Break Condition Setting

1° metodo

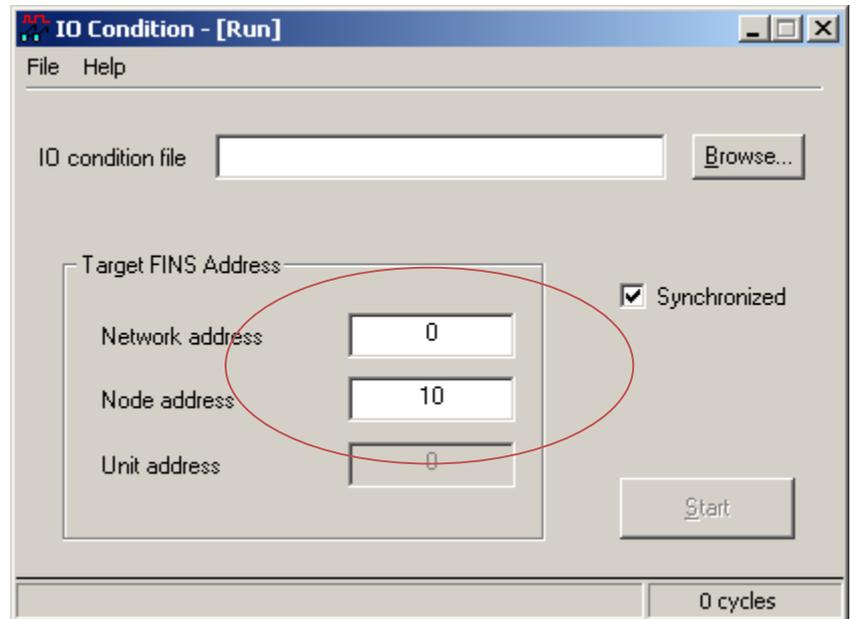
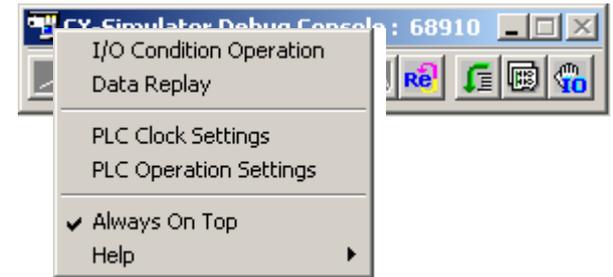
- Le impostazioni PLC definite in CX-Programmer vengono trasferite sul PLC virtuale e sono osservabili cliccando col tasto destro sulla barra di intestazione della debug console di CX-Simulator e selezionando I/O Condition Operation
- Nella finestra che apparirà si potrà controllare se effettivamente il network address e il node address di rete corrispondono a quello definito nel progetto

1° metodo

Impostazioni CX-Programmer

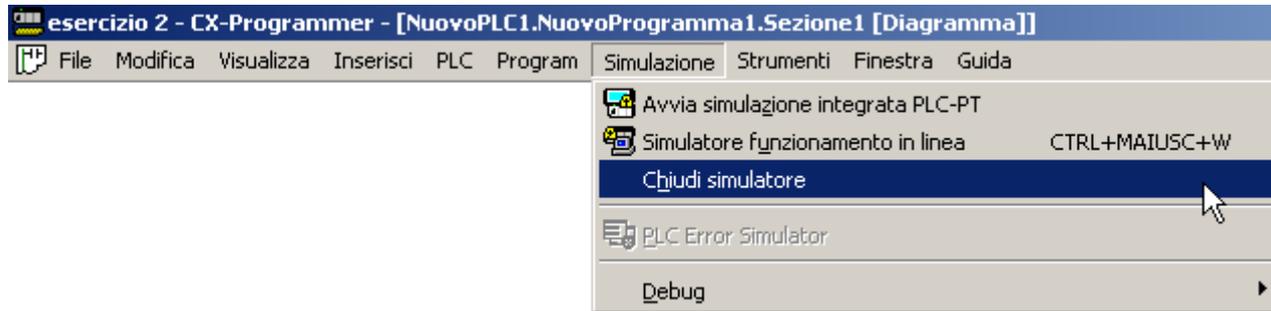


Impostazioni CX-Simulator

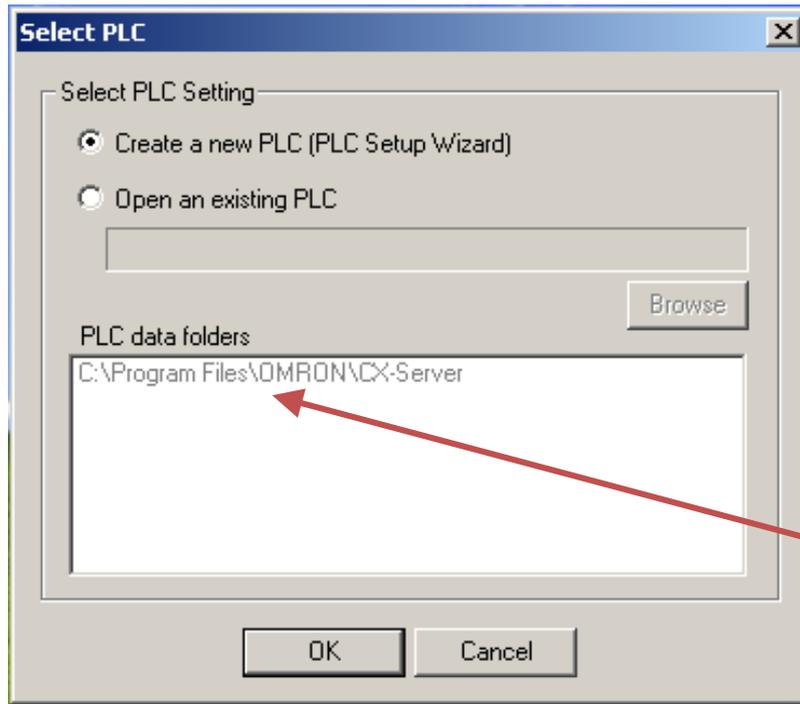


1° metodo

- Per chiudere la simulazione tornare in modalità off-line premendo di nuovo il tasto  e andare nel menù a tendina su “Simulazione” → “Chiudi simulatore”

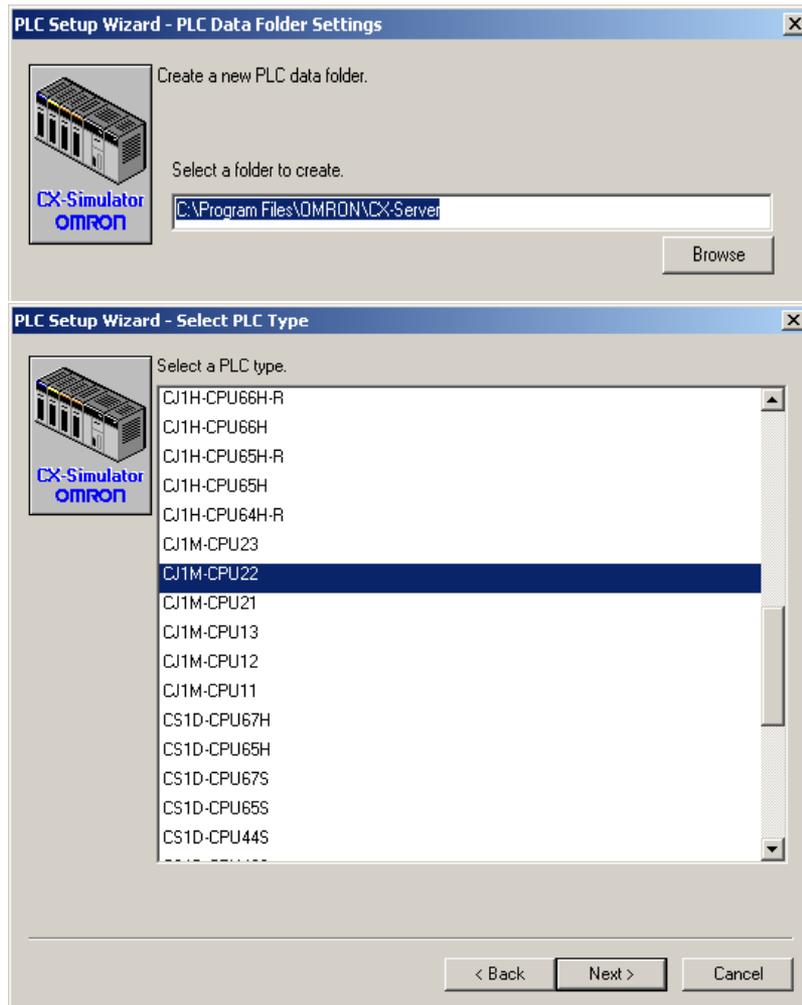


2° metodo: creare un PLC virtuale



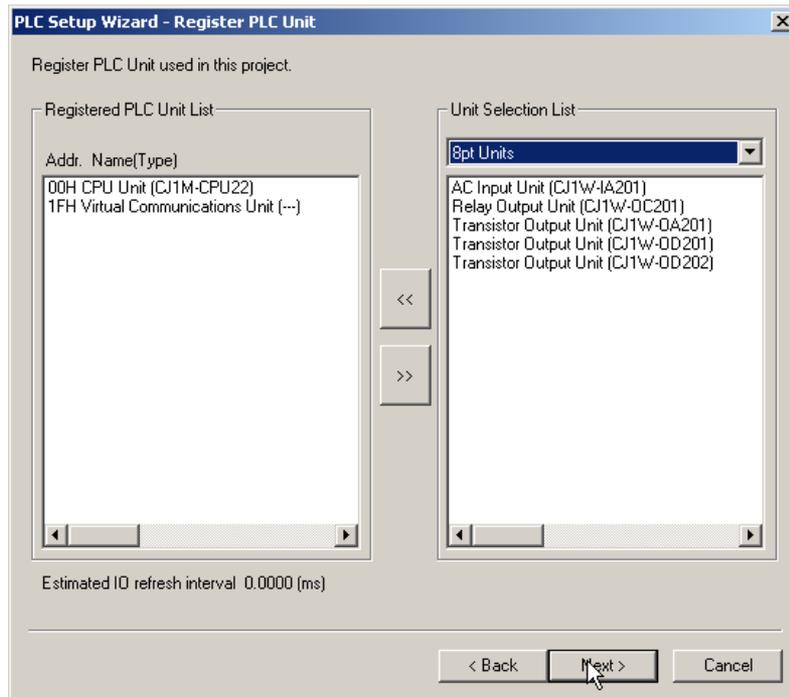
- Aprire CX-Simulator dal menu Start → Programmi → Omron → CX-Simulator
- Specificare se creare un nuovo PLC virtuale o aprirne uno già esistente specificando il percorso esatto

2° metodo: creare un PLC virtuale



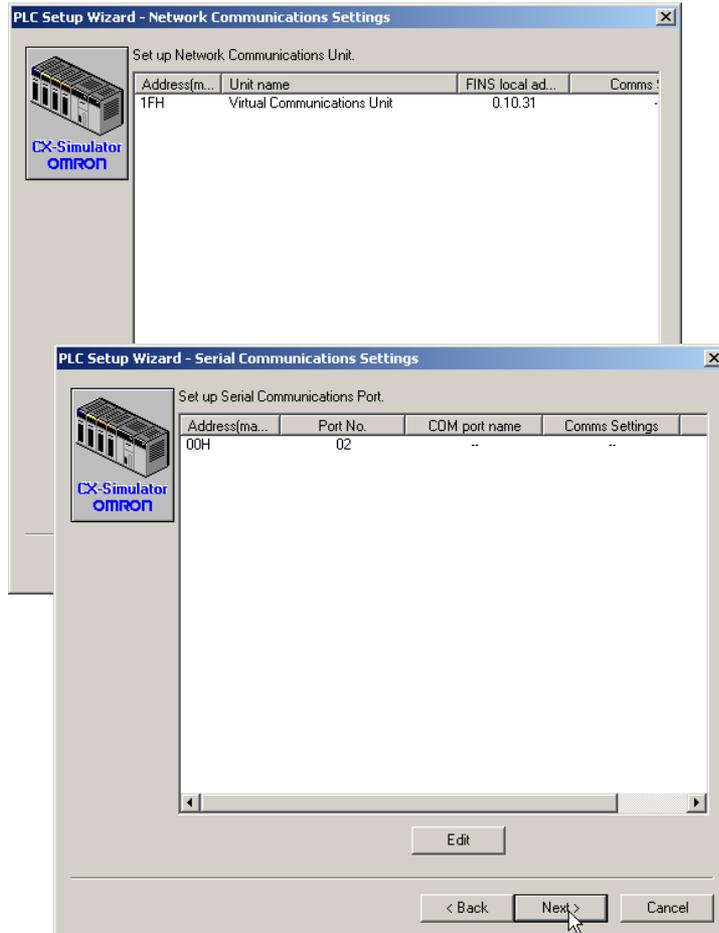
- Se si sta creando un nuovo PLC specificare prima la cartella in cui salvarlo e dopo il modello di PLC che si vuole simulare con relativa CPU

2° metodo: creare un PLC virtuale



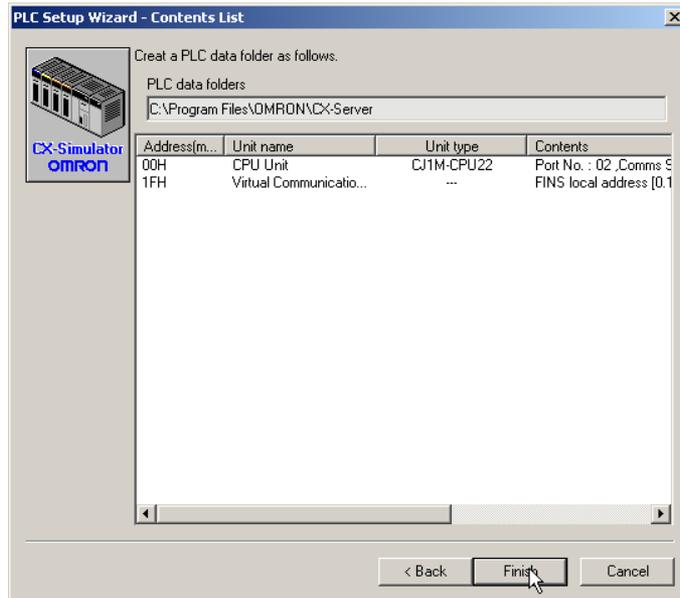
- In questa finestra è possibile comporre la configurazione di periferiche del PLC virtuale selezionando il componente desiderato dall'elenco sulla destra e premendo il tasto  per trasferirlo nell'elenco dei componenti aggiunti a sinistra

2° metodo: creare un PLC virtuale



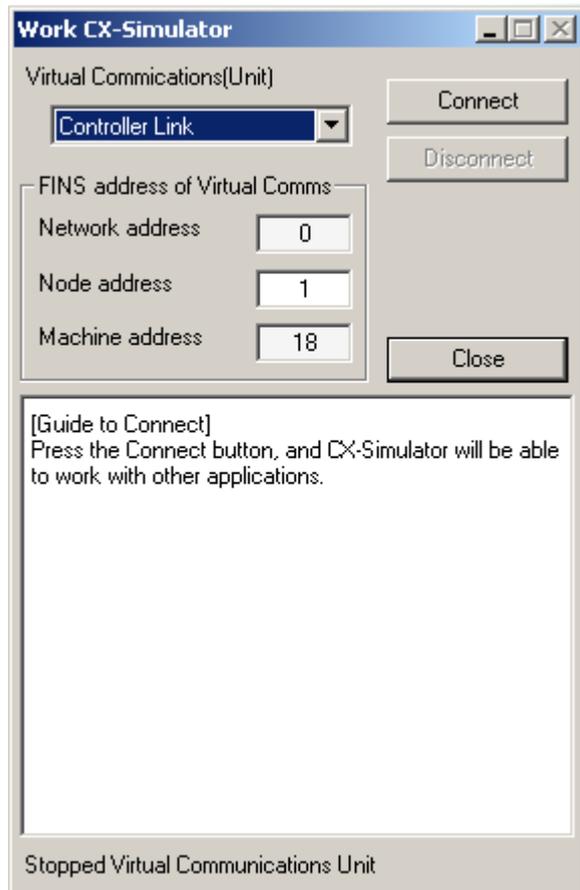
- Mandare avanti le successive due finestre cliccando su “Next”

2° metodo: creare un PLC virtuale



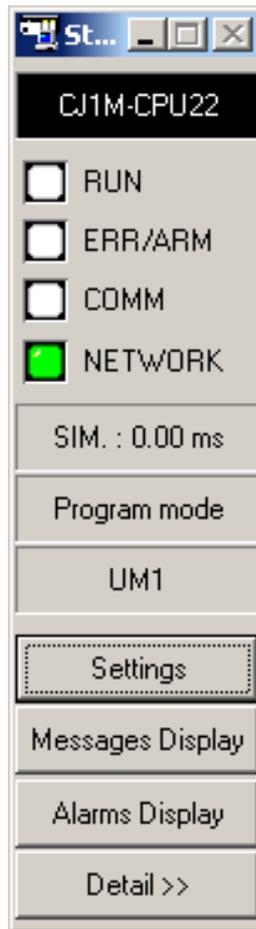
- L'ultima finestra è un riepilogo del PLC creato e di tutte le schede ad esso collegate
- Se la configurazione è quella voluta, cliccare su "Finish"

2° metodo: connessione al PLC



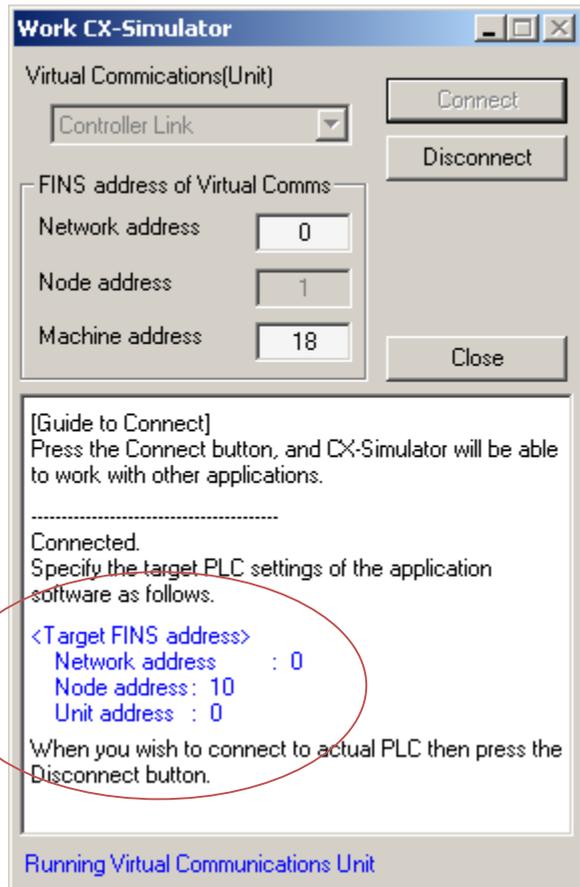
- Creato il nuovo PLC non rimane altro che connetterlo:
 - Selezionare come tipo di rete Controller Link
 - Premere il pulsante Connect

2° metodo: connessione al PLC



- Se si connette correttamente il led “Network” del PLC virtuale si accenderà
- La finestra rappresentante il PLC fornisce anche informazioni sul modello del PLC e sul suo stato di funzionamento tramite il led “Run”

2° metodo: connessione al PLC



- La finestra mostrerà un messaggio di avvenuta connessione
- N.B. Annotarsi il Network e il Node Address utile per la configurazione del CX-Programmer
- Apparirà così la Simulator Debug Console, vista precedentemente, che permette di gestire l'elaborazione del programma mentre viene eseguito dal PLC virtuale



PLC serie CJ2M

Corso intermedio CPS2

Contenuti

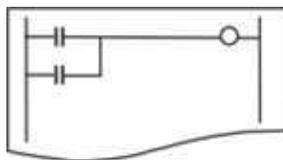
- Introduzione alla serie CJ2M
- Configurazione del sistema
- Funzioni ladder avanzate
- Task e Subroutine
- Indirizzamento indiretto e Cicli For-Next
- Moduli CJ2M-MD21_
- Gestione degli Interrupt
- Uso dei contatori veloci
- Introduzione alle uscite a treno di impulsi
- Funzione di Backup/Restore
- Scambio dati

Serie CJ2M

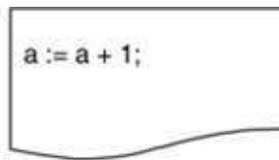


CJ2M: a prima vista...

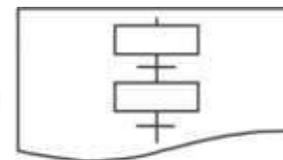
- **SCALABILITA'**: cinque taglie di memoria programma (da 5 a 60 Kstep!).
- **SEMPLICITA'**: porta USB di programmazione su tutte le CPU. 
- **CONNETTIVITA'**: modelli disponibili con Ethernet integrata e funzione di EtherNet/IP Data Link. 
- **PROGRAMMAZIONE**: possibilità di utilizzare tre diversi linguaggi di programmazione: Ladder, ST e SFC. 



Ladder diagram
Control of equipment
and external devices



ST
Math processing



SFC
Step progression control

CJ2M: a prima vista...

- **MODULARITA'**: controllo di 2 o 4 assi con moduli opzionali a treno di impulsi.



- **PORTA SERIALE:**

- Integrata su modelli senza EtherNet/IP;
- Modulo RS-232C (o RS-422A/485) opzionale per i modelli con Ethernet.



- **ARCHIVIAZIONE:** salvataggio dati su Compact Flash (128–256–512 MB) ancora presente su tutte le CPU.



La serie CJ2M

- La serie di CPU CJ2M si divide in due gruppi:
 - **CPU11, 12, 13, 14 e 15:** CPU standard con porta seriale integrata;
 - **CPU31, 32, 33, 34 e 35:** CPU con Ethernet integrata e porta seriale su slot opzionale. La porta Ethernet supporta anche il protocollo industriale EtherNet/IP.



Specifiche tecniche

Item		CJ2M-				
		CPU 11/31	CPU 12/32	CPU 13/33	CPU 14/34	CPU 15/35
User Memory		5K steps	10K steps	20K steps	30K steps	60K steps
I/O Bits		2,560 bits				
Processing Speed	Overhead Processing Time *1	Normal Mode: CJ2M-CPU3□:			270 μs	
		CJ2M-CPU1□:			160 μs	
	Execution Time	Basic Instructions: 0.04 μs min.; Special Instructions: 0.06 μs min.				
Interrupts	I/O Interrupts and External Interrupts		Interrupt task startup time: 31 μs Return times to cyclic tasks: 10 μs			
DM Area		32k words				
EM Area		32k words/bank × 4 banks max.: E00_00000 to E3_32767 max. :				
		32K words × 1 bank			32K words × 4 banks	
Maximum Number of Connectable Units		Total per CPU Rack or Expansion Rack: 10 Units max. Total per PLC: 40 Units max.				

Aree di memoria

- **Area CIO** (Core I/O Area): 6.144 word, nessun prefisso. Area di memoria di I/O, moduli speciali, schede di comunicazione. Area non ritentiva, viene azzerata ad ogni avvio del PLC. L'accesso può avvenire a bit o a canale.
- **Area HR** (Holding Area): 1.536 word, prefisso H. Area ritentiva gestibile a bit e a canale. Memorizzazione permanente di dati di programma.
- **Area W** (Work Area): 512 word, prefisso W; per l'utilizzo come registri temporanei o Bit di appoggio; viene usata tipicamente per l'allocazione automatica. Accesso a bit e a canale. Non ritentiva.

Aree di memoria

- **Area A** (Auxiliary Area): 3.008 word di cui 1.984 a sola lettura, prefisso A. Contiene informazioni per la diagnostica, lo stato e la configurazione del PLC.
- **Area DM** (Data Memory): 32.768 word, prefisso D. Area di memoria ritentiva accessibile a canale per la memorizzazione dei dati. Viene utilizzata anche per la memorizzazione della configurazione dei moduli speciali. Per la serie CJ2 l'accesso può avvenire a bit o a canale.

Aree di memoria

- **Area EM** (Expanded Memory): 32.768 word per banco, prefisso da E0_ a E3_. Area ritentiva indirizzabile a canale.
 - 1 banco per le CPU 11,12, 13, 31, 32 e 33.
 - 4 banchi per CPU 14, 15, 34 e 35.

I banchi EM possono essere trasformati in unica area di memoria di massa detta File Memory. Quest'area può essere usata allo stesso modo della Memory Card.

Aree di memoria

- **Area Temporizzatori (Timer Area):** 4.096 word, prefisso T. Viene simulato il funzionamento di temporizzatori. L'area è condivisa per tutte le istruzioni di temporizzazione tranne quelle di TIML(542), TIMLX(553), MTIM e MTIMX(554) che non necessitano di un numero di timer.
- **Area Contatori (Counter Area):** 4.096 word, prefisso C. Viene simulato il funzionamento di contatori. L'area è condivisa per tutte le istruzioni di conteggio: CNT e CNTR(012).

Aree di memoria aggiuntive

- Sulle CPU CJ2M sono presenti delle ulteriori aree di memoria per il salvataggio di simboli e commenti.
- Queste aree vengono trasferite insieme al programma e permettono di caricare il programma dal PLC conservando i simboli e i commenti:
 - **Area simboli:** contiene il simbolico delle variabili;
 - **Area commenti:** contiene i commenti contenuti nel programma ladder;
 - **Area indice:** contiene i nomi dei programmi e delle sezioni di programma.

Aree dedicate ai Function Block

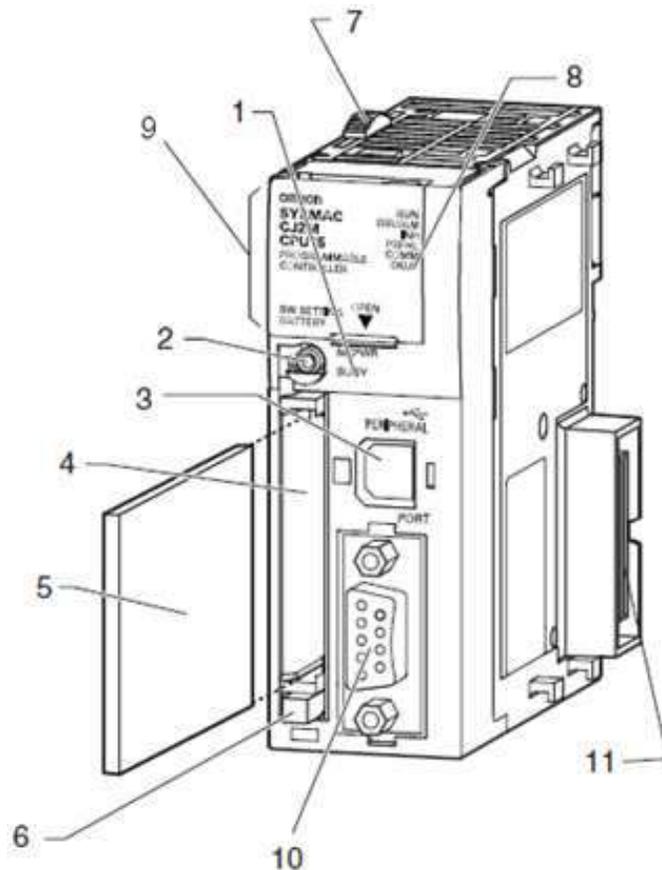
- I PLC Omron che supportano i Function Block hanno delle aree di memoria dedicate per tale scopo.
- **Memoria Sorgente FB:** in questa area trova posto il programma sorgente dei Blocchi Funzione, è utile alle funzioni di Debug.
- **Memoria programma FB aggiuntiva:** tutte le CPU CJ2M hanno a disposizione 20 kstep di memoria utente extra che può essere utilizzata dai blocchi funzione.



Configurazione del sistema

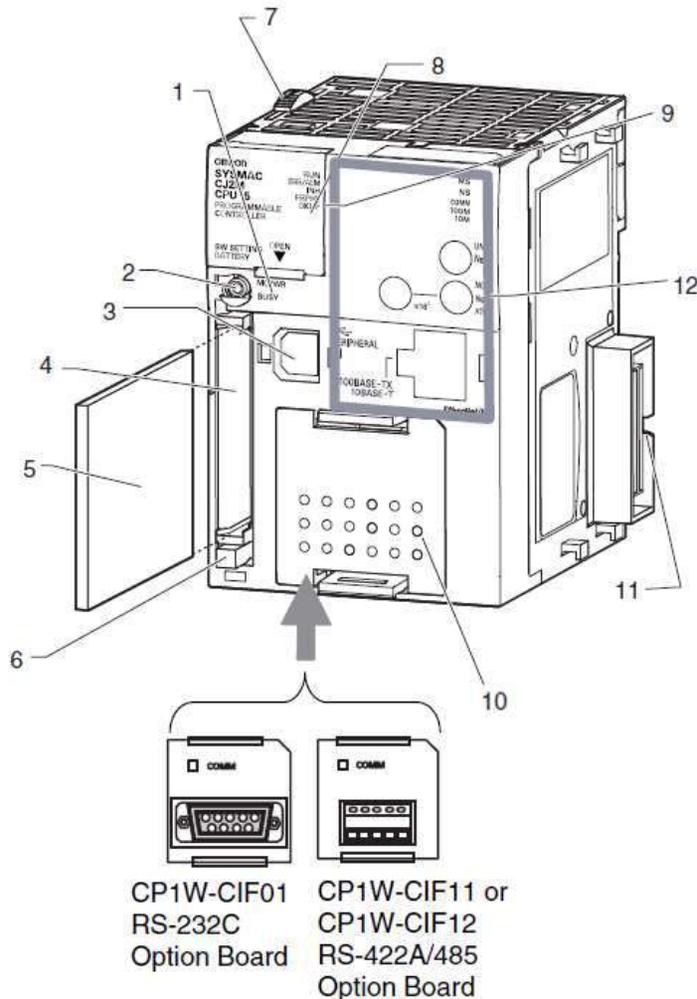
Corso intermedio CPS2

Moduli CPU 1_



1. Indicatori di accesso e di alimentazione della Memory Card.
2. Interruttore di alimentazione Memory Card, utilizzato anche per eseguire il backup dei dati.
3. Porta USB per collegare dispositivi di programmazione, come il CX-Programmer.
4. Connettore Memory Card.
5. Scheda di memoria. Tutti i dati del Modulo CPU possono essere copiati su una scheda di memoria per eseguire il backup dei dati.
6. Pulsante di espulsione per rimuovere la Memory Card dalla CPU.
7. Leva di blocco tra la CPU e il Modulo successivo.
8. Indicatori LED dello stato della CPU.
9. DIP Switch.
10. RS-232C porta seriale.
11. Bus Connector, collegato all'unità successiva.

Moduli CPU 3_



1. Indicatori di accesso e di alimentazione della Memory Card.
2. Interruttore di alimentazione Memory Card, utilizzato anche per eseguire il backup dei dati.
3. Porta USB per collegare dispositivi di programmazione, come il CX-Programmer.
4. Connettore Memory Card.
5. Scheda di memoria. Tutti i dati del Modulo CPU possono essere copiati su una scheda di memoria per eseguire il backup dei dati.
6. Pulsante di espulsione per rimuovere la Memory Card dalla CPU.
7. Leva di blocco tra la CPU e il Modulo successivo.
8. Indicatori LED dello stato della CPU.
9. DIP Switch.
10. Schede opzionali seriali collegabili allo slot:
 - Scheda opzionale RS-232C: CP1W-CIF01
 - Scheda opzionale RS-422A/485: CP1W-CIF11/12
11. Bus Connector, collegato all'unità successiva.
12. EtherNet/IP integrata.

Funzione dei DIP Switch

Dip Switch	Funzione descrizione
1	ON: memoria programma e impostazioni protetta in scrittura.
	OFF: memoria programma e impostazioni accessibili.
2	ON: caricamento del programma e delle impostazioni dalla scheda di memoria.
	OFF: avvio normale della CPU.
5	ON: la porta RS-232 è forzata al protocollo Toolbus.
	OFF: la porta RS-232 prende la configurazione dalla memoria del PLC.
7	ON: abilita la funzione "Easy Backup".
	OFF: nessuna operazione.

 = Valore predefinito

Gli switch 3, 4, 6 e 8 non sono utilizzati. L'impostazione predefinita per tutti gli switch è OFF.

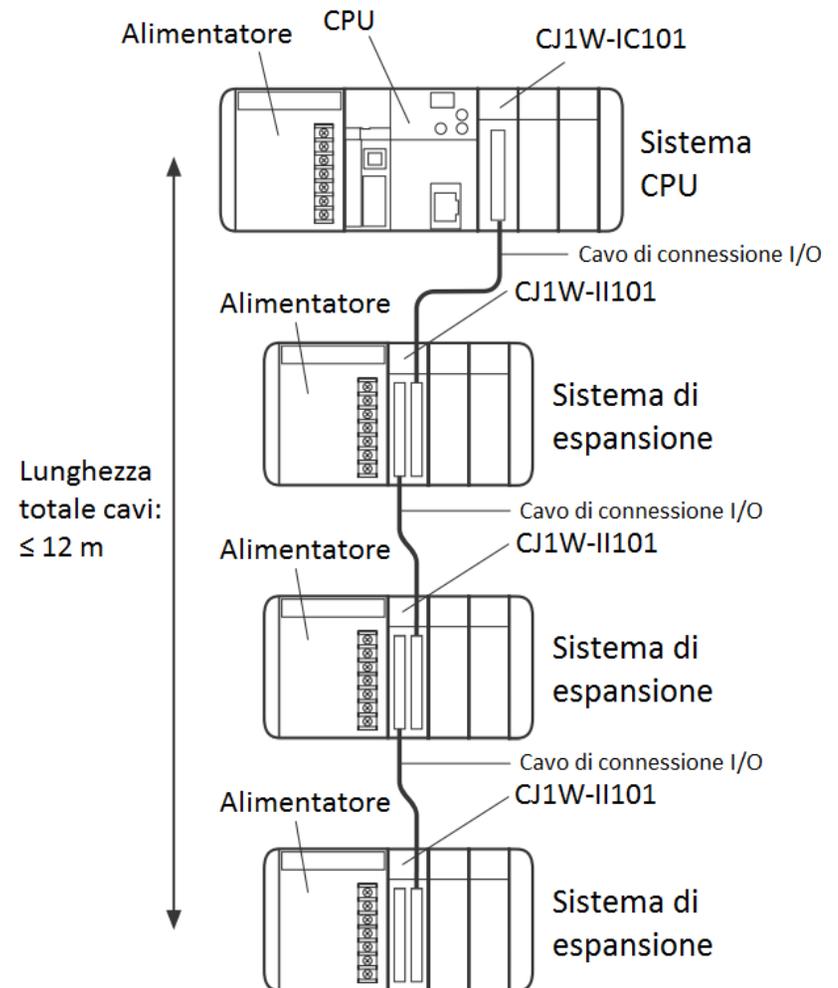
Memory Card

- La Memory Card ha la stessa funzione della cartuccia di Backup dei PLC della serie CP1L.
- Tutti i PLC della serie CJ dispongono di uno slot per Memory Card di tipo Compact Flash.
- Attualmente, per i PLC della serie CJ sono disponibili memorie da 128, 256 e 512 MB.
- La Compact Flash è formattata in standard FAT16, i file contenuti sono accessibili anche da PC utilizzando un lettore/adattatore per schede CF.



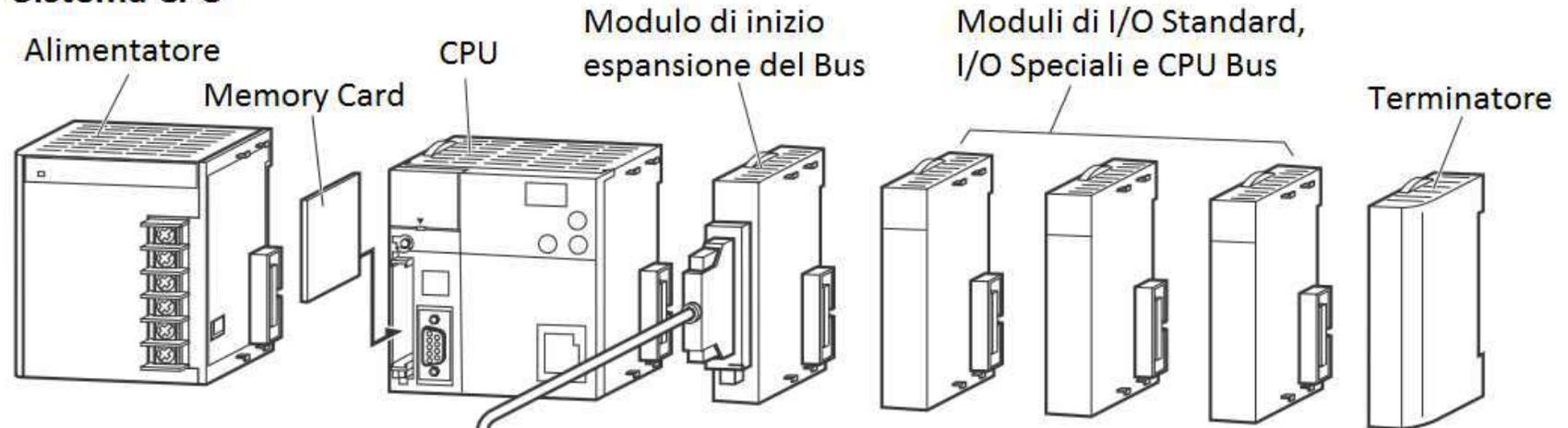
Configurazione del sistema

- La configurazione di sistema consente la massima flessibilità grazie all'assenza del rack.
- Max. 10 moduli installabili a fianco della CPU.
- 0/1/2/3 unità di espansione (**10 moduli per unità**) per un totale di 10/20/30/40 moduli collegabili.
- Ogni unità di espansione è composta da alimentatore, modulo di interfaccia e moduli di I/O.
- Distanza massima totale: **12 m**.

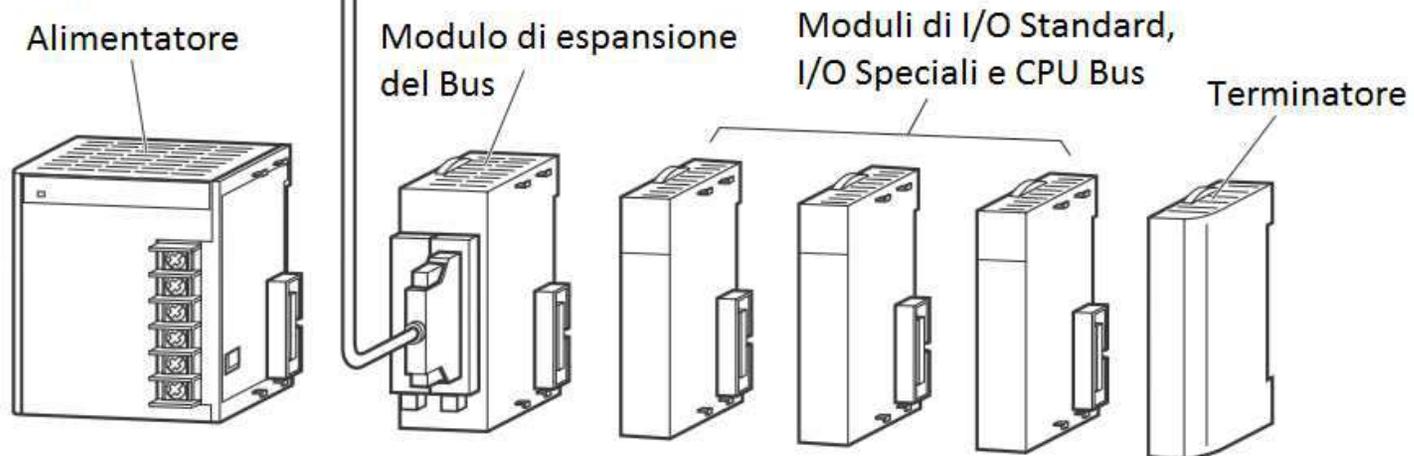


Configurazione del sistema

Sistema CPU



Sistema di espansione

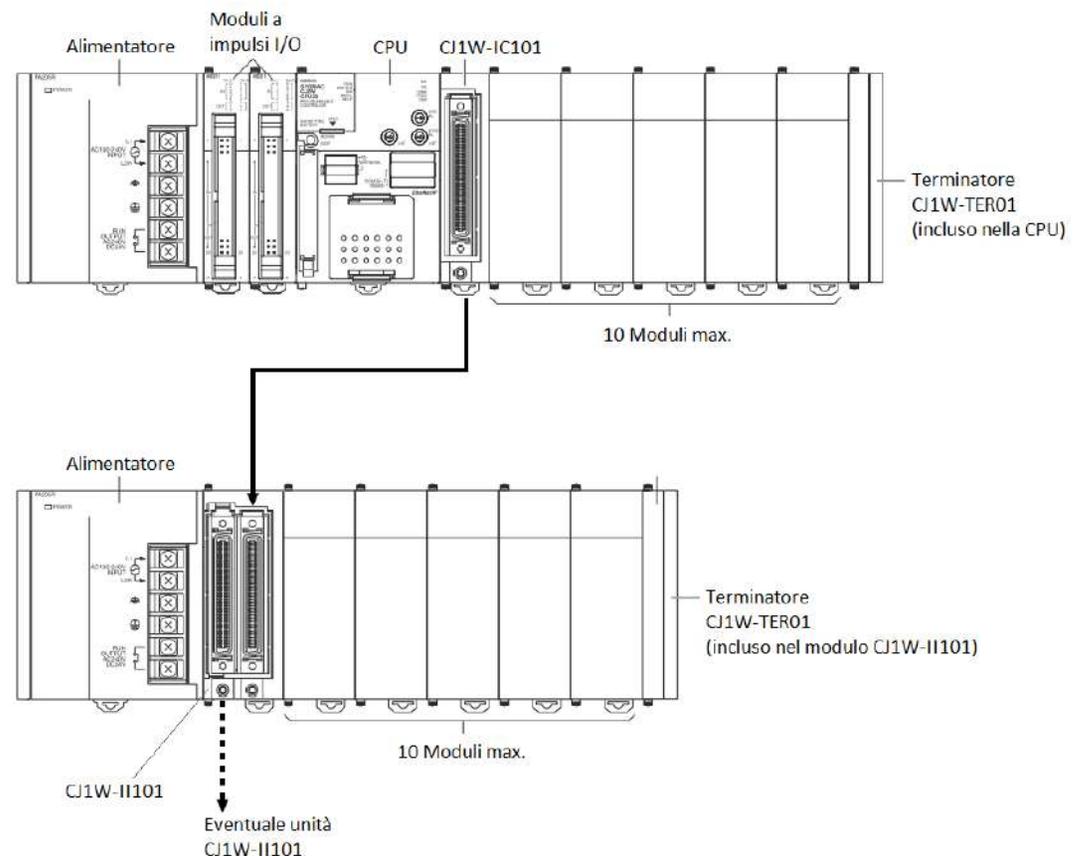


Configurazione del sistema

La configurazione massima prevede un totale di 40 moduli installabili.

E' necessario rispettare le seguenti specifiche:

- Moduli I/O Standard: **massimo 40.**
- Moduli I/O Speciali: **massimo 40.**
(indirizzabili da 0 a 96)
- Moduli Bus CPU: **massimo 16.**





Configurazione della CPU

Corso intermedio CPS2

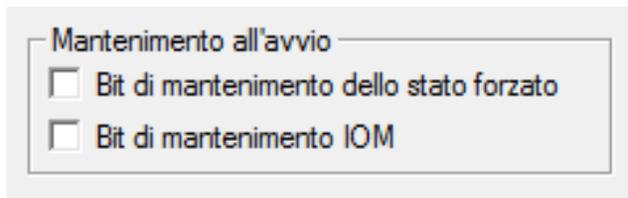
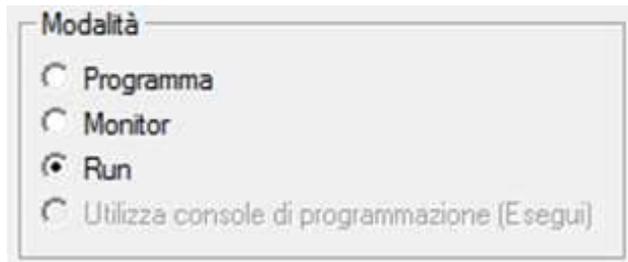
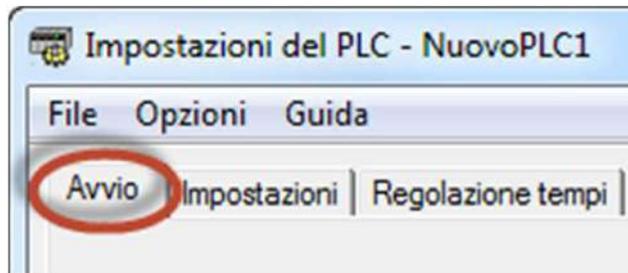
Setup del PLC

- La finestra Impostazioni PLC di CX-Programmer rappresenta uno strumento completo per la configurazione del PLC.
- Per accedere alla finestra di impostazioni è sufficiente richiamare la voce “Modifica” → “Impostazioni” dal menù “PLC”, o cliccare due volte sull'icona relativa, nell'area di progetto ( Impostazioni).
- Il menù “Opzioni” della finestra permette di leggere, trasferire, verificare o proteggere in scrittura la configurazione corrente, oppure riportare la configurazione alle impostazioni di default.
- La configurazione può essere eventualmente anche stampata (“File” → “Stampa”).

Impostazioni Comuni

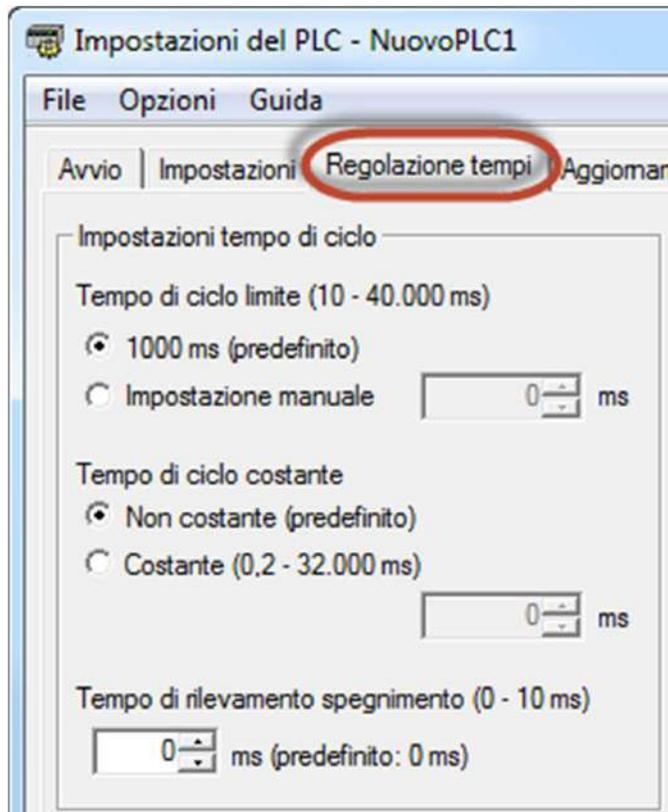
- Per tutte le CPU della serie CJ2M, CX-Programmer permette di configurare:
 - Modalità di avvio del PLC;
 - Le impostazioni relative alla CPU (gestione errori, esecuzione in background,...);
 - Tempi di ciclo, monitoraggio, identificazione spegnimento...;
 - Tempi di risposta degli ingressi;
 - Impostazioni relative alle porte di comunicazione;
 - Impostazioni servizi periferica.

Impostazioni di avvio



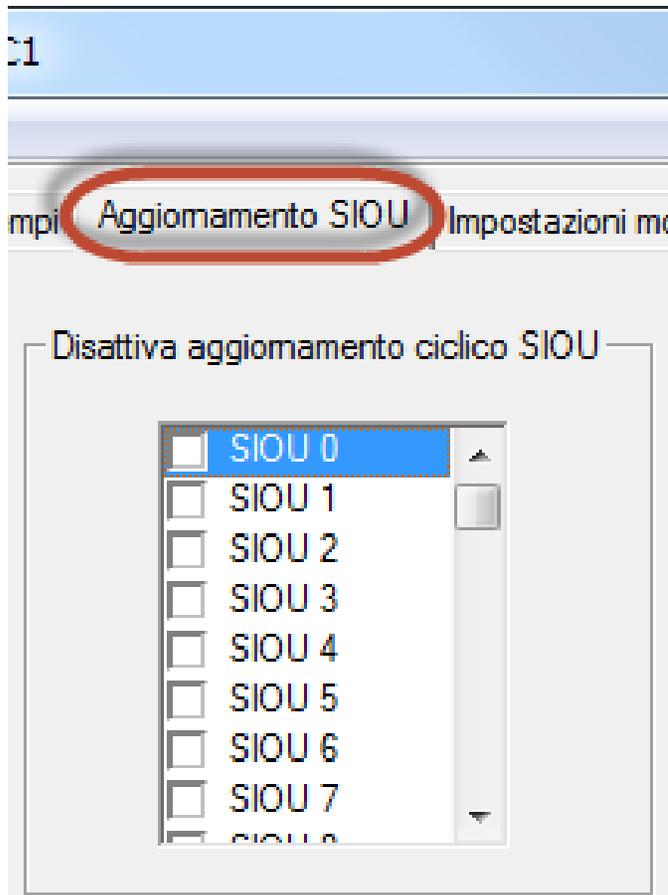
- **Modalità:** viene stabilito se la modalità di avvio del PLC deve corrispondere a quella selezionata nella console di programmazione o a quella indicata nel setup del PLC.
- **Mantenimento all'avvio:** i bit A500.12 e A500.13, che rendono ritenitive l'area di memoria di I/O e le forzature, non vengono più ignorati.

Impostazione del tempo di ciclo



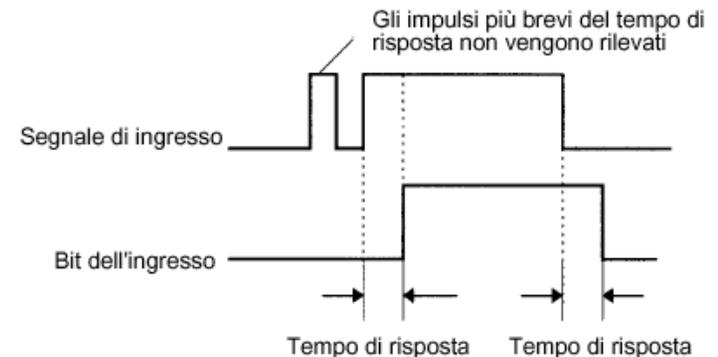
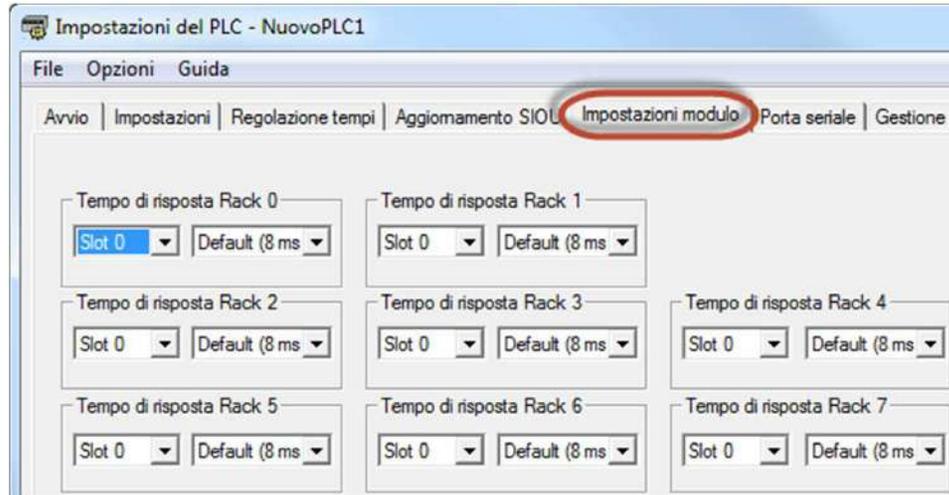
- Queste impostazioni hanno effetto sul **tempo di ciclo** della CPU.
- Se il tempo di ciclo supera il valore di ciclo limite si ha un errore fatale della CPU. Il valore max del ciclo può essere impostato da 10 ms a 40 s.
- E' possibile rendere il ciclo costante impostando una durata minima del ciclo. Se il tempo di ciclo corrente è inferiore al minimo, la CPU si mette in attesa fino al completamento del tempo previsto. Il ciclo minimo può essere impostato da 0.2 ms a 32 s.

Aggiornamento ciclico SIOU



- Disattiva aggiornamento ciclico SIOU:** se un modulo di I/O speciale verrà rinfrescato all'interno di un task ad interrupt con l'istruzione IORF(097) è necessario disabilitare il refresh ciclico per quel modulo. In caso contrario il flag di errore della duplicazione del rinfresco (A402.13) andrà ad ON ed il rinfresco non verrà eseguito nel modo richiesto.

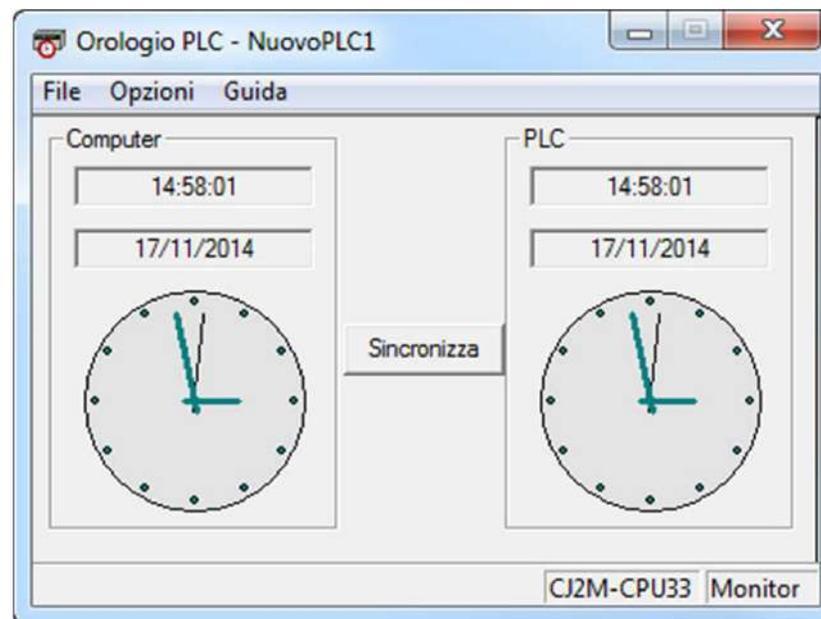
Impostazioni dei tempi di risposta



- Per tutti i moduli I/O Standard è possibile specificare il tempo di risposta degli ingressi (da 0 a 32 ms) in modo da ridurre l'effetto dei disturbi.
- Specificando 0 ms, il tempo di risposta dipenderà esclusivamente dalle caratteristiche elettroniche del modulo.

Orologio/Calendario

- La funzione Orologio/Calendario è implementata su tutti i PLC della serie CJ.
- E' possibile sincronizzare data e ora direttamente da CX-Programmer.



Orologio/Calendario

- Nei quattro canali dell'area A (da A351 ad A354) risiedono, secondo lo schema riportato nella tabella sottostante, le informazioni relative all'orologio calendario:

Name	Address	Description
Clock Data	A351.00 to A351.07	Seconds: 00 to 59 (BCD)
	A351.08 to A351.15	Minutes: 00 to 59 (BCD)
	A352.00 to A352.07	Hour: 00 to 23 (BCD)
	A352.08 to A352.15	Day of the month: 01 to 31 (BCD)
	A353.00 to A353.07	Month: 01 to 12 (BCD)
	A353.08 to A353.15	Year: 00 to 99 (BCD)
	A354.00 to A354.07	Day of the week (00 to 06 BCD): 00: Sunday, 01: Monday, 02: Tuesday, 03: Wednesday, 04: Thursday, 05: Friday, 06: Saturday

Orologio/Calendario

- Esiste inoltre un intero set di istruzioni dedicato alla gestione di tali dati:

Instruction	Mnemonic	Function
CLOCK ADJUSTMENT	DATE(735)	Changes the internal clock setting to the setting in the specified source words.
HOURS TO SECONDS	SEC(065)	Converts time data in hours/minutes/seconds format to an equivalent time in seconds only.
SECONDS TO HOURS	HMS(066)	Converts seconds data to an equivalent time in hours/minutes/seconds format.
CALENDAR ADD	CADD(730)	Adds time to the calendar data in the specified words.
CALENDAR SUBTRACT	CSUB(731)	Subtracts time from the calendar data in the specified words.
Time Comparison Instructions	=DT(341) <>DT(342) <DT(343) <=DT(344) >DT(345) >=DT(346)	Time comparison instructions compare two BCD time values and create an ON execution condition when the comparison condition is true.

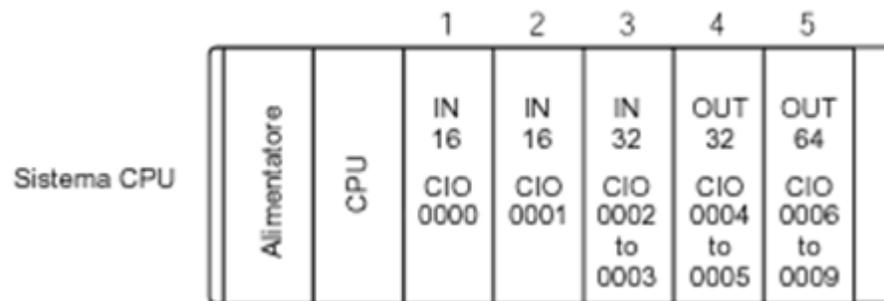


CJ2: allocazione dei moduli e Tabella I/O

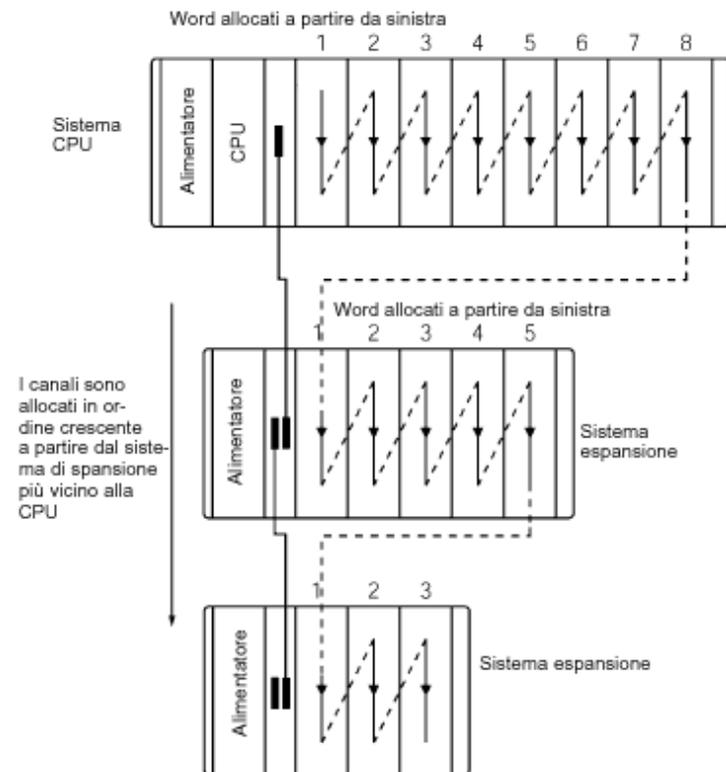
Corso intermedio CPS2

Allocazione I/O (Moduli I/O Standard)

L'allocazione dei canali CIO per i moduli di I/O Standard procede in modo consecutivo seguendo l'ordine di installazione, sia per gli ingressi che per le uscite, a partire dal canale **CIO 0**.



NB: L'eventuale modulo di espansione va posizionato subito a destra della CPU (o subito a destra dell'alimentatore, nei sistemi di espansione).



Allocazione I/O (Moduli I/O Speciali)

Area I/O:

Per lo scambio dati con la CPU utilizzano 10 canali CIO per modulo, allocati secondo il numero di unità N.

Calcolo indirizzo base area CIO:

$$ch = 2000 + 10 \times N$$

Area DM:

Per la configurazione utilizzano 100 canali D per modulo, allocati secondo il numero di unità N.

Calcolo indirizzo base area D:

$$ch = 20000 + 100 \times N$$

Numero unità	Word assegnate
0	CIO 2000... CIO 2009
1	CIO 2010... CIO 2019
2	CIO 2020... CIO 2029
15	CIO 2150... CIO 2159
95	CIO 2950... CIO 2959

Numero unità	Word assegnate
0	D20000... D20099
1	D20100... D20199
2	D20200... D20299
15	D21500... D21599
95	D29500... D29599

Allocazione I/O (Moduli Bus CPU)

Area I/O:

Per lo scambio di informazioni con la CPU utilizzano 25 canali CIO per modulo, allocati secondo il numero di unità N.

Calcolo indirizzo base area CIO:

$$ch = 1500 + 25 \times N$$

Numero unità	Word assegnate
0	CIO 1500... CIO 1524
1	CIO 1525... CIO 1549
2	CIO 1550... CIO 1574
15	CIO 1875... CIO 1899

Area DM:

Per la configurazione utilizzano 100 canali D per modulo, allocati secondo il numero di unità N.

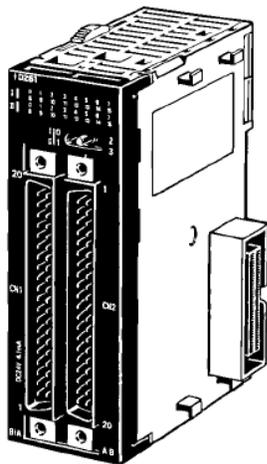
Calcolo indirizzo base area D:

$$ch = 30000 + 100 \times N$$

Numero unità	Word assegnate
0	D30000... D30099
1	D30100... D30199
2	D30200... D30299
15	D31500... D31599

Allocazione I/O – Riepilogo

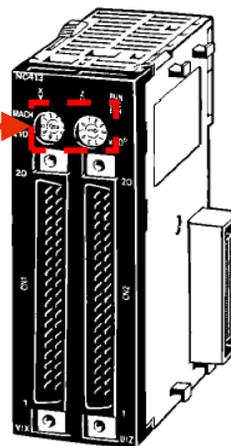
Moduli I/O Standard



Da CIO 0000 a CIO 0159
(8, 16, 32 o 64 punti)
(Max. 40 moduli)

Numero di modulo
I/O Speciale (Max. 40 moduli)

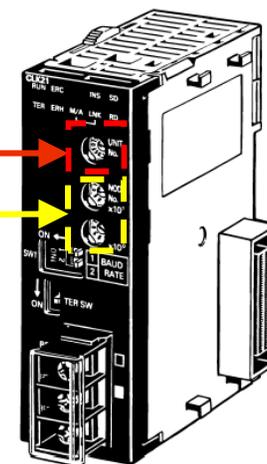
Moduli I/O Speciali



Da CIO 2000 a CIO 2959
(Refresh Area: 10 word/unità)
Da D20000 a D29599
(Setup Area: 100 word/unità)

Numero di modulo
Bus CPU (Max. 16 moduli)

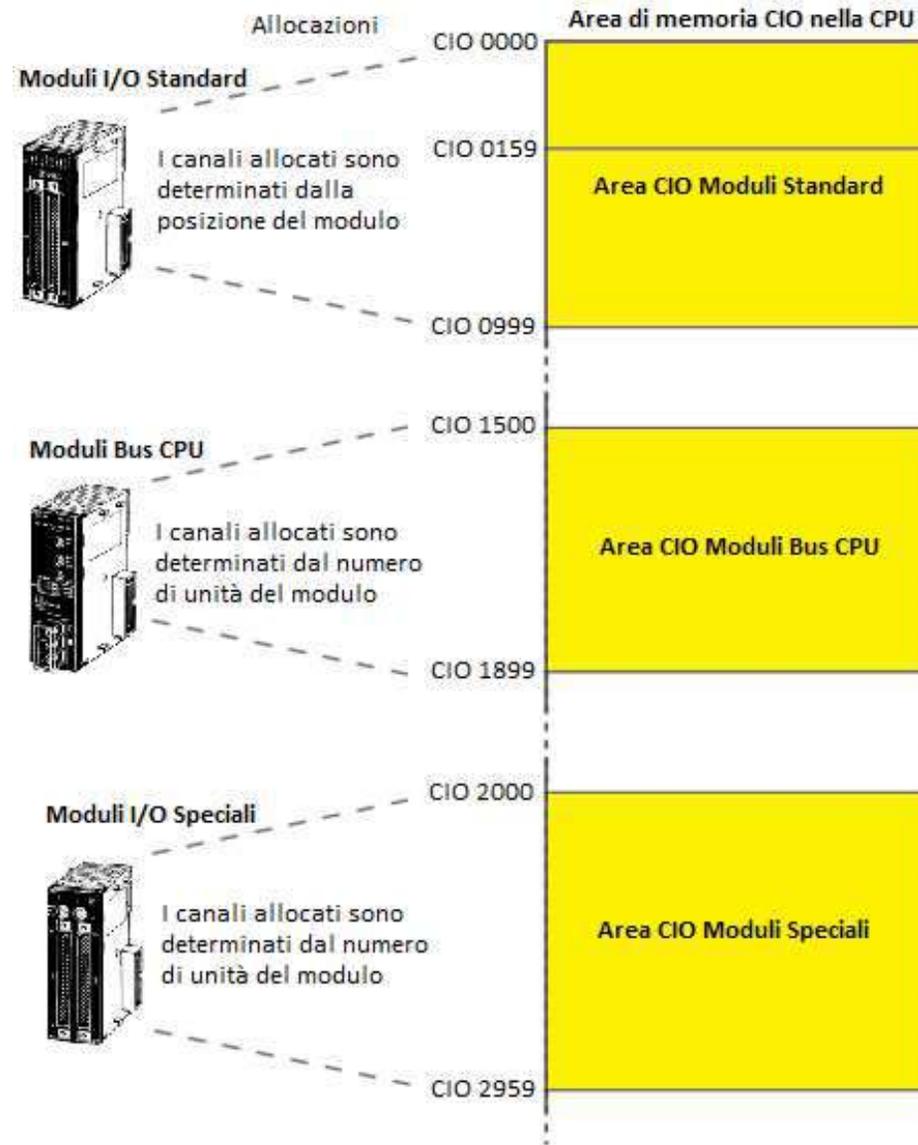
Moduli Bus CPU



Da CIO 1500 a CIO 1899
(Refresh Area : 25 word/unità)
Da D30000 a D31599
(Setup Area: 100 word/unità)

Indirizzo del nodo del
modulo Bus CPU

Allocazione I/O – Riepilogo



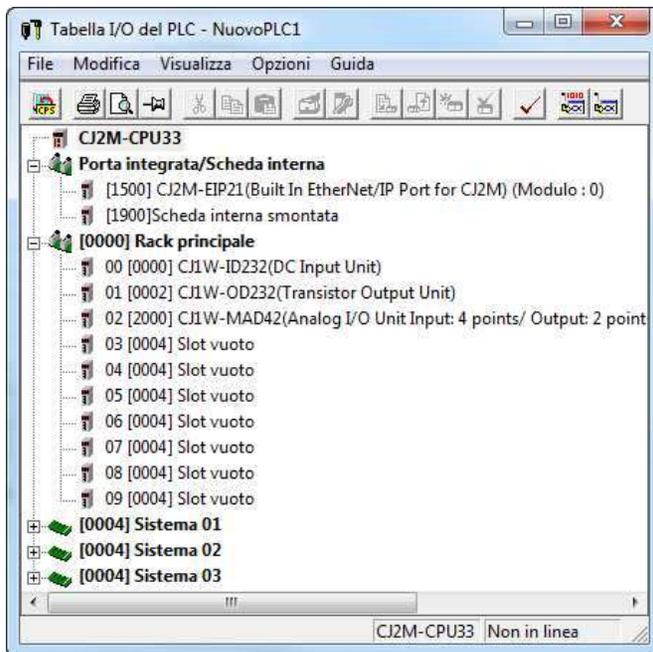
La Tabella di I/O

- La configurazione hardware dei moduli del CJ può essere memorizzata in una tabella detta 'Tabella di I/O'.
- All'accensione, se la tabella è stata registrata, il sistema verifica la configurazione attuale con quella registrata; in caso di mancata corrispondenza viene generato un errore:
 - Se è stato tolto, aggiunto o spostato un modulo, il sistema fornisce la segnalazione di allarme **non fatale** 'I/O VERIFY ERROR';
 - Se la variazione di configurazione è costituita invece dallo scambio di posizione tra un modulo di ingresso ed uno di uscita, viene fornita la segnalazione di allarme **fatale** 'I/O SET ERROR' e l'esecuzione del programma si arresta.

La Tabella di I/O

- La Tabella di I/O può essere:
 - Creata, Letta, Verificata e Cancellata.
- Se la tabella è stata cancellata il sistema all'accensione non esegue alcun controllo.
- In CX-Programmer è stato previsto un intero menù dedicato alla gestione della Tabella di I/O, cui si accede cliccando su “Tabella I/O impostazione modulo” dal menù “PLC” → “Modifica”, oppure facendo doppio click sull'icona relativa:  Configurazione tabella I/O e modulo).
- Il canale A261 del PLC contiene i dettagli di eventuali errori riscontrati durante la creazione della Tabella di I/O.

La Tabella di I/O – CX-Programmer



Rappresentazione grafica della Tabella di I/O

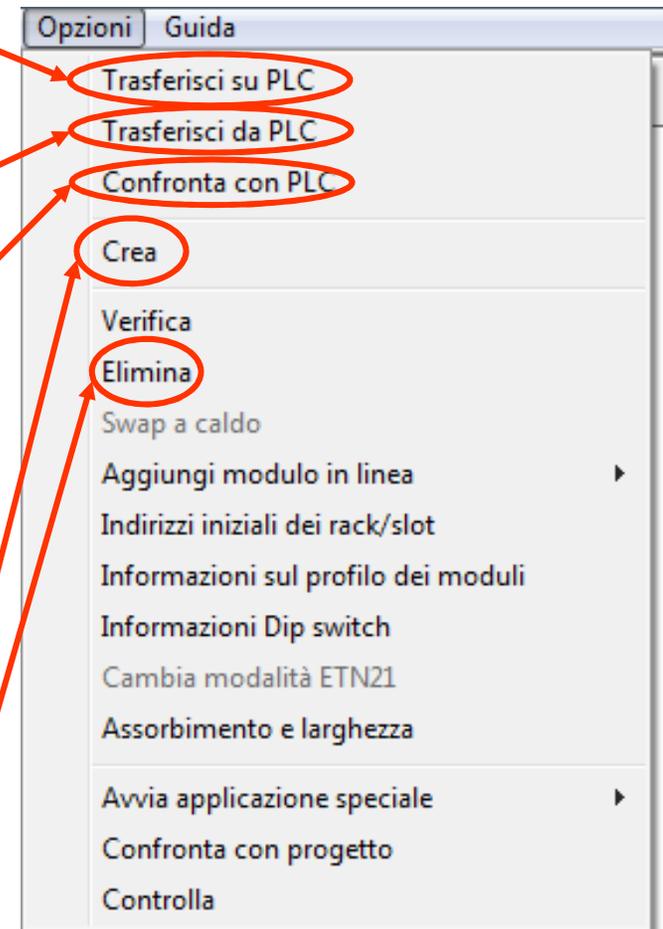
- **Scrive** la Tabella di I/O sul PLC.

- **Legge** la Tabella di I/O dal PLC collegato.

- **Confronta** la Tabella di I/O del PLC con quella memorizzata nel progetto.

- **Genera** la Tabella di I/O.

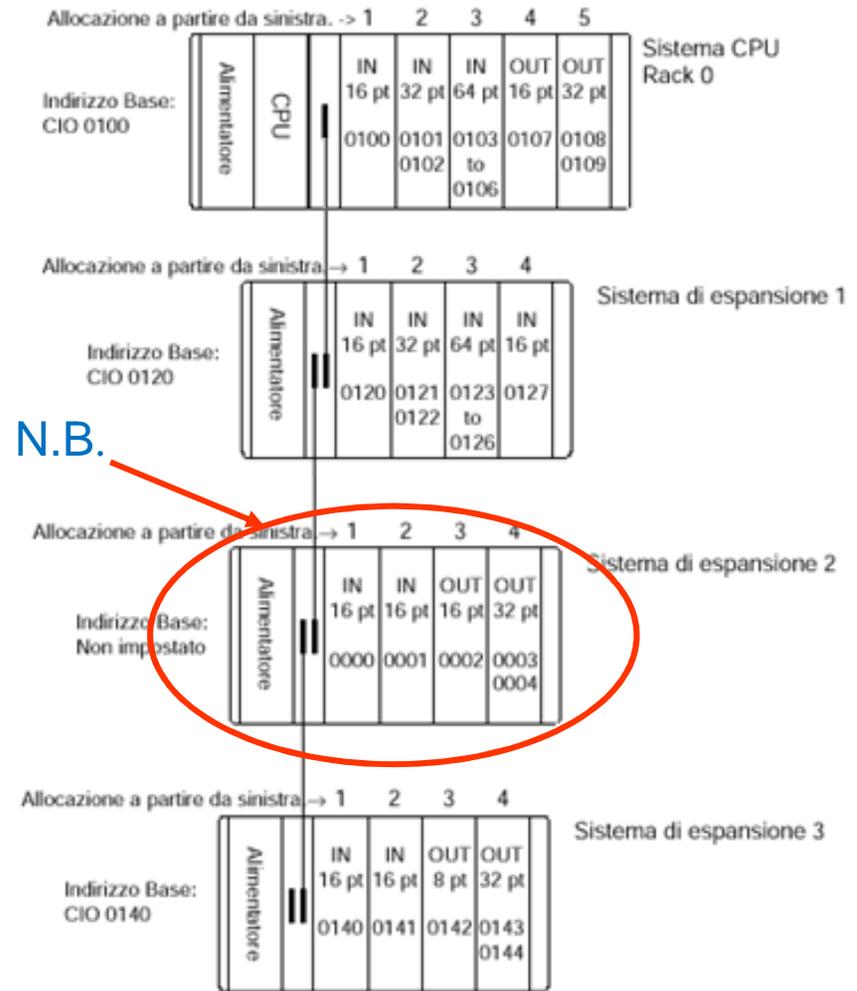
- **Elimina** dal PLC la Tabella di I/O registrata.



Allocazione I/O – Indirizzo iniziale rack

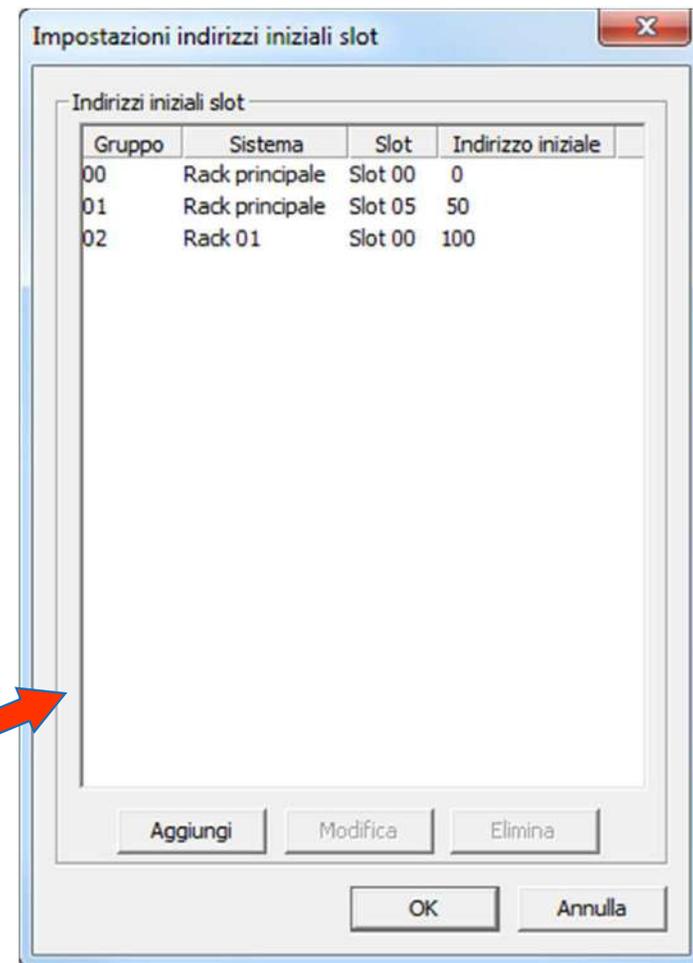
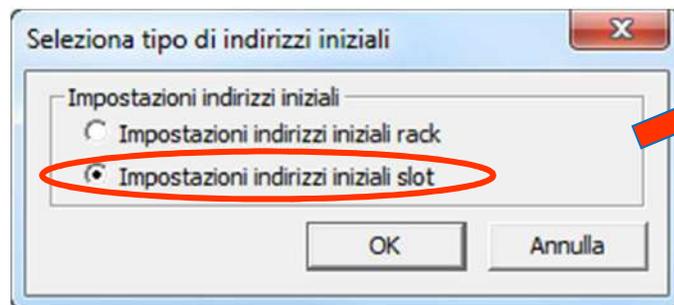
Dall'editor della Tabella di I/O di CX-Programmer è possibile impostare un indirizzo di base per il sistema CPU e/o per ciascun sistema di espansione (tra CIO 0 e CIO 900).

- Su ciascun sistema il primo indirizzo disponibile non dipenderà più dal sistema precedente e dai moduli montati su di esso.
- Non sono ammesse sovrapposizioni di indirizzi: l'indirizzo base di un sistema non può essere utilizzato da un altro sistema.



Allocazione I/O – Indirizzo iniziale slot

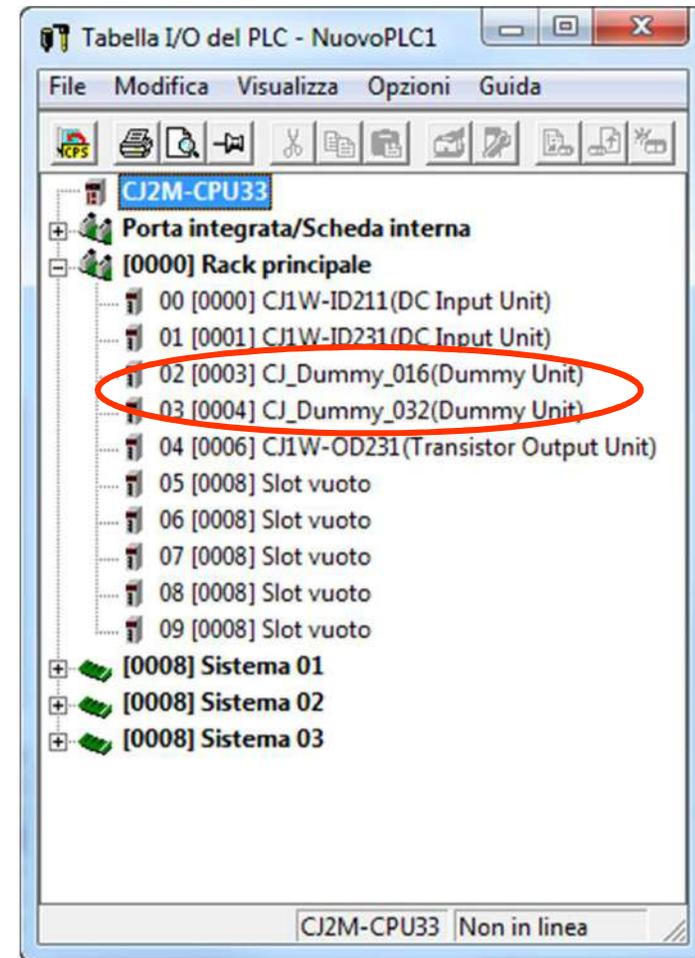
- Dall'editor della Tabella di I/O è possibile assegnare un indirizzo qualsiasi a ciascuno slot del PLC.
- In questo caso la corrispondenza tra indirizzo e posizione del modulo è completamente arbitraria.



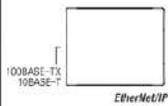
Allocazione I/O – Moduli fittizi

- CX-Programmer permette di riservare un certo numero di canali da associare a moduli fisicamente non presenti.
- Anche questa operazione viene svolta nell'editor della Tabella di I/O in CX-Programmer.

	1	2	3	4	5
Alimentatore					
CPU	IN 16 pt	IN 32 pt	Riser. 16 pt	Riser. 32 pt	OUT 32 pt
	0000	0001 0002	0003	0004 0005	0006 0007



Allocazione I/O – Esempio

		0	1	2	3	4
Alimentatore	CPU 	IN 16 pt	Modulo I/O Speciale	Modulo BUS CPU	OUT 16 pt	Modulo BUS CPU
		CIO 0000	CIO 2000 a 2009	CIO 1500 a 1524	CIO 0001	CIO 1525 a 1549

Slot	Modulo	Modello	Canali richiesti	Canali allocati	Numero di modulo	Gruppo
---	Porta EtherNet/IP integrata nella CPU	CJ2M-CPU33	25	CIO 1500 to CIO 1524	0	Modulo CPU (la porta è trattata come un modulo Bus CPU)
0	Modulo IN 16 punti in c.c.	CJ1W-ID211	1	CIO 0000	---	Modulo I/O Standard
1	Modulo 8 IN analogici	CJ1W-AD081	10	CIO 2000 to CIO 2009	0	Modulo I/O Speciale
2	Modulo comunicazione seriale	CJ1W-SCU41	25	CIO 1525 to CIO 1549	1	Modulo Bus CPU
3	Modulo OUT a transistor NPN, 16 punti	CJ1W-OD211	1	CIO 0001	---	Modulo I/O Standard
4	Modulo Controller Link	CJ1W-CLK21	25	CIO 1550 to CIO 1574	2	Modulo Bus CPU



Istruzioni di programmazione avanzate

Corso intermedio CPS2

Contenuti

- Istruzioni intermedie
- Impostazione dei Timer in Hex
- Funzioni Timer evolute
- Virgola mobile

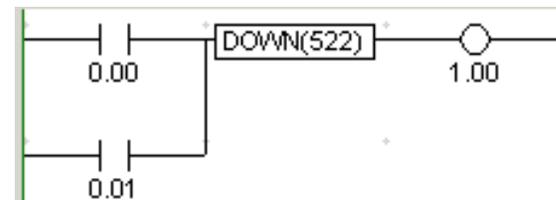
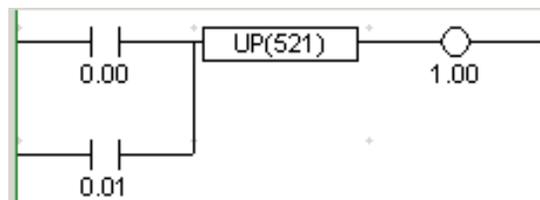


Istruzioni ladder avanzate

- I PLC serie CJ, CS e CP supportano un insieme di istruzioni avanzate.
- **Istruzioni Intermedie:** esistono alcune funzioni cosiddette 'intermedie' che non si trovano al termine della Rung come tutte le altre, bensì all'interno di essa, e richiedono quindi anche un segnale di uscita.
- **Timer avanzati:** oltre alle funzioni Timer base (TIM, TIMH...) esistono istruzioni Timer avanzate, Timer multi-target e Timer totalizzatori.
- **Virgola mobile:** esiste un insieme completo di istruzioni per la gestione dei dati in virgola mobile.

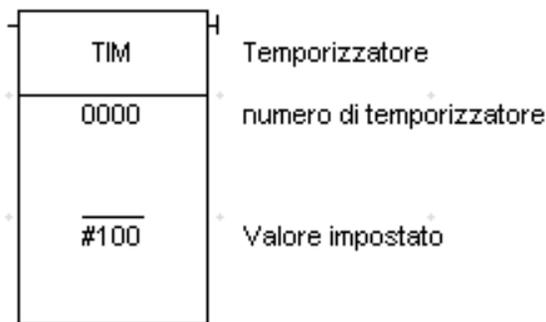
Altre istruzioni ladder avanzate

- Istruzioni di ingresso differenziate:
 - Fronte di salita (@);
 - Fronte di discesa (%).
- Istruzioni di refresh immediato:
 - Refresh immediato di un ingresso, di un'uscita o di un'istruzione (!);
 - Possibile combinazione con la differenziazione;
 - Funzione IORF(097).
- Istruzioni UP e DOWN:
 - Come DIFU e DIFD;
 - Sono istruzioni intermedie che non richiedono di un bit di appoggio.

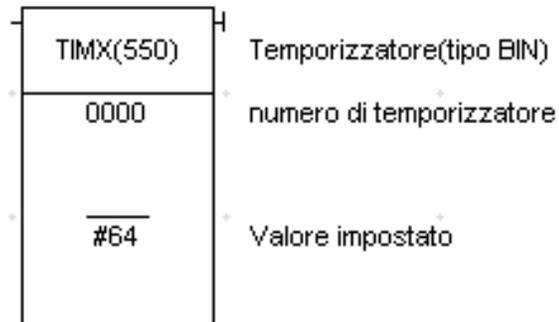


Timer e Counter con impostazione Esadecimale

- Nei PLC serie CJ, CS e CP esistono due gruppi di istruzioni Timer e Counter:
 - Timer e Counter a impostazione BCD;
 - Timer e Counter a impostazione Esadecimale.



Modalità BCD 10 sec.
SV=#100



Modalità Hex 10 sec.
SV=#64 oppure SV=&100

Timer e Counter con impostazione Esadecimale

- I PLC CJ2 e CP1E consentono l'uso contemporaneo di entrambe le modalità di impostazione dei Timer e Counter.
- Nei PLC CJ1, CS1, CP1L e CP1H è indispensabile scegliere quale tipo di impostazione si vuole utilizzare. Questi PLC non permettono l'uso contemporaneo dei Timer/Counter BCD e dei Timer/Counter Hex.

Le istruzioni Timer e Counter

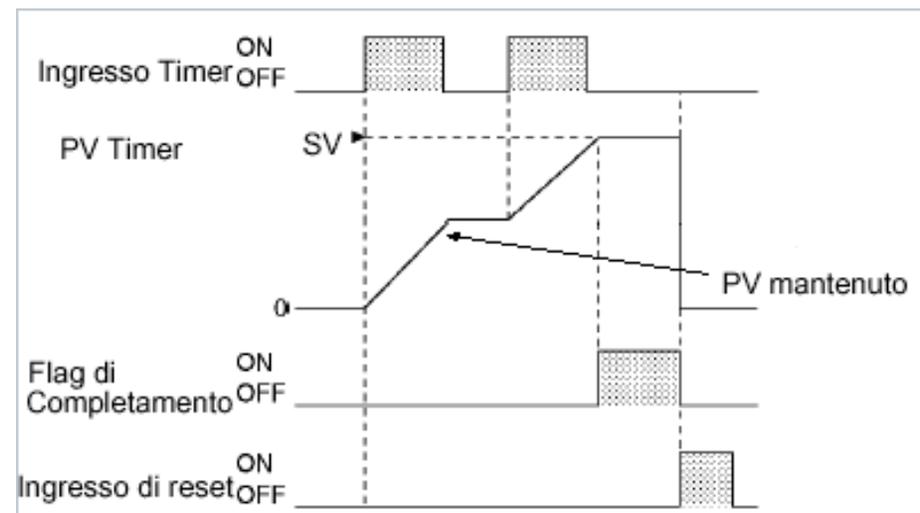
Nome	Mnemonico	
	Modalità BCD	Modalità Binaria
TIMER (100 ms)	TIM	TIMX(550)
HIGH-SPEED TIMER (10 ms)	TIMH(015)	TIMHX(551)
ONE-MS TIMER (1 ms)	TMHH(540)	TMHHX(552)
ACCUMULATIVE TIMER (100 ms)	TTIM(087)	TTIMX(555)
LONG TIMER (100 ms)	TIML(542)	TIMLX(553)
MULTI-OUTPUT TIMER (100 ms)	MTIM(543)	MTIMX(554)
COUNTER	CNT	CNTX(546)
REVERSIBLE COUNTER	CNTR(012)	CNTRX(548)
RESET TIMER/COUNTER	CNR(545)	CNRX(547)

Timer totalizzatore TTIM/TTIMX

- L'istruzione TTIM (TTIMX) implementa un temporizzatore ad incremento con ritardo programmabile in unità di decimi di secondo. L'SV può variare tra 0 e 999,9 secondi (0 e 6553,5 secondi per TTIMX).
- PV viene mantenuto in caso di interruzione del segnale di abilitazione.

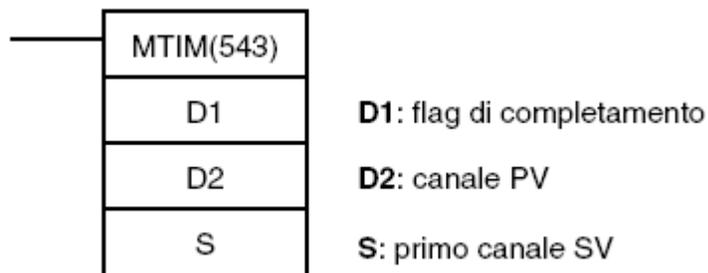


N: Numero Timer
S: Set value

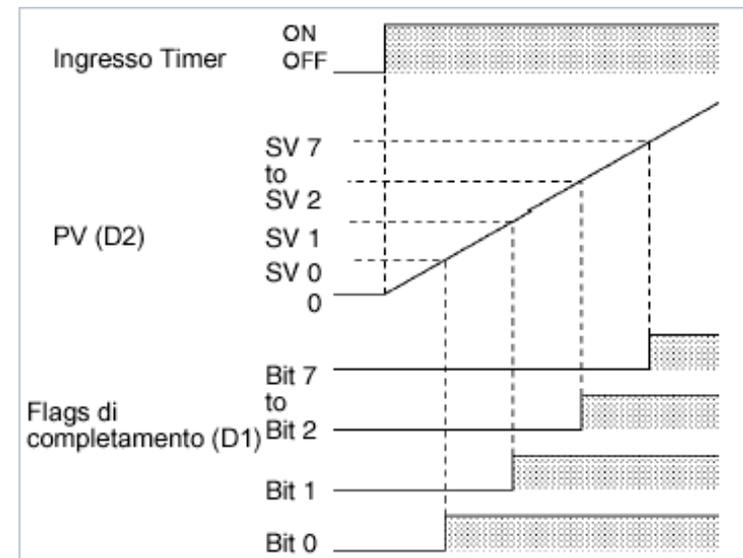


Timer a impostazione multipla MTIM/MTIMX

- Il temporizzatore multi uscita MTIM (MTIMX) implementa un temporizzatore a incremento con 8 valori di impostazione e 8 flag di completamento indipendenti. Ogni ritardo è programmabile in unità di decimi di secondo (da 0 a 999,9 s).

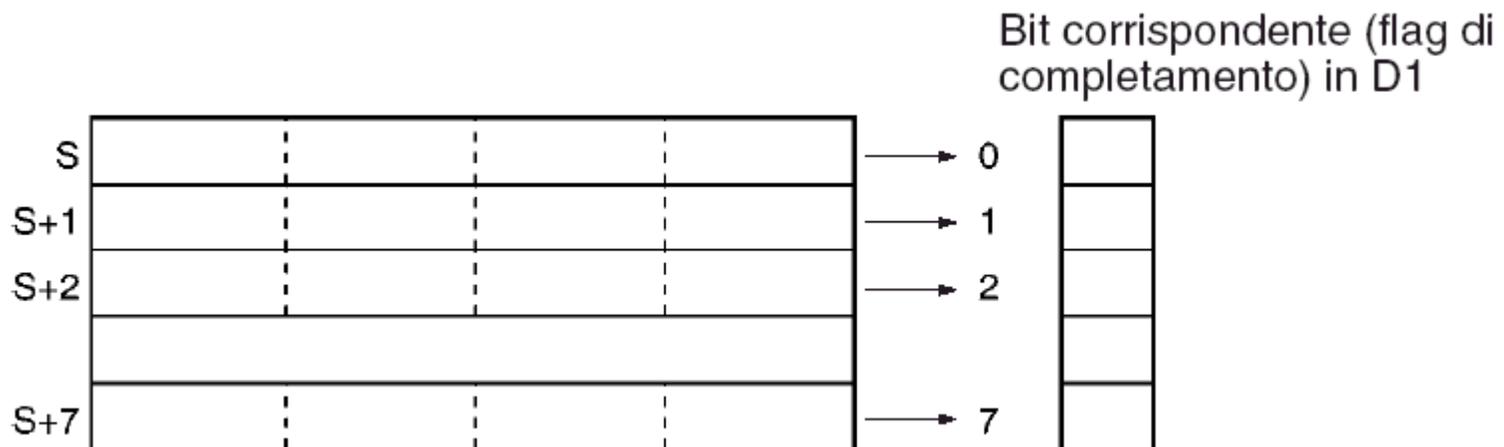
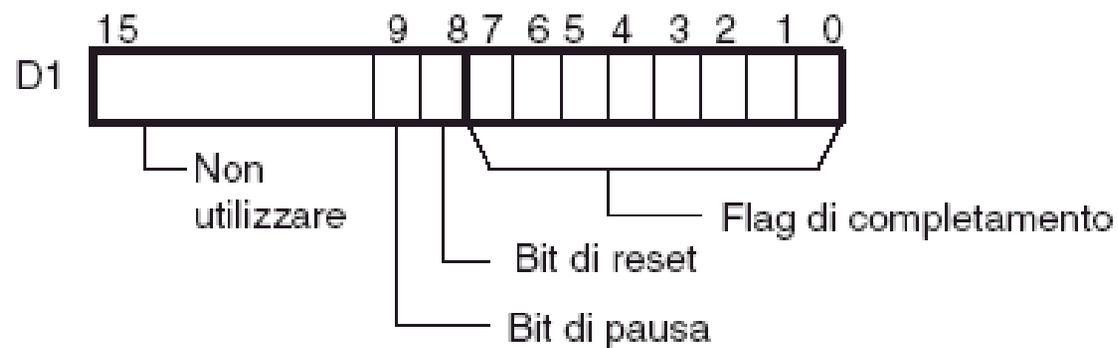


- I Flag sono attivati al raggiungimento del relativo setpoint e restano attivi fino al reset.
- Il timer prosegue il conteggio dopo il superamento dell'ultimo setpoint.



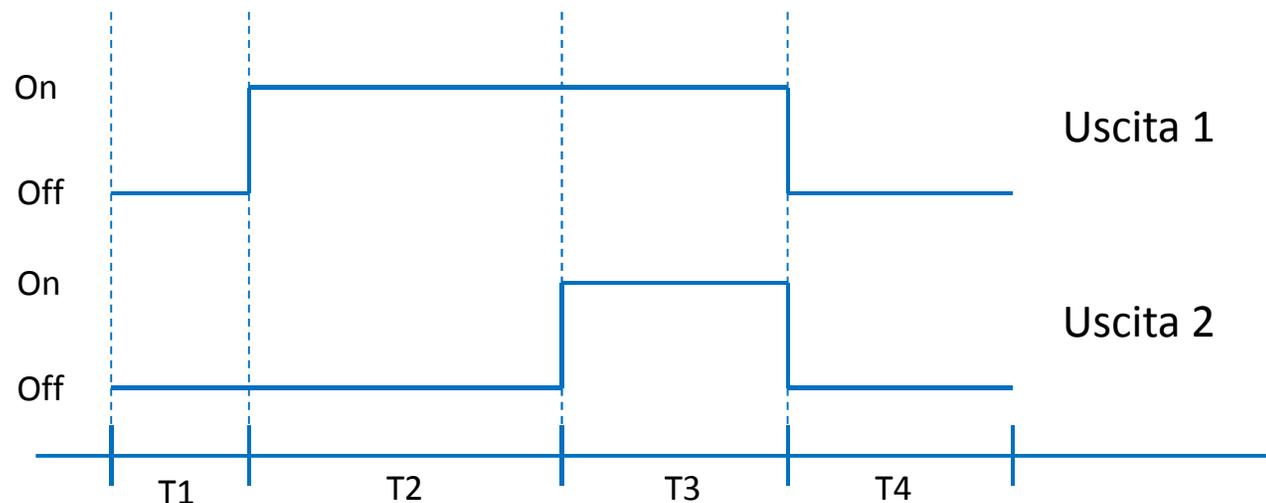
Timer a impostazione multipla MTIM/MTIMX

D1 contiene gli otto flag di completamento nonché i bit di pausa e di reset.



Esempio: utilizzo del temporizzatore multiplo

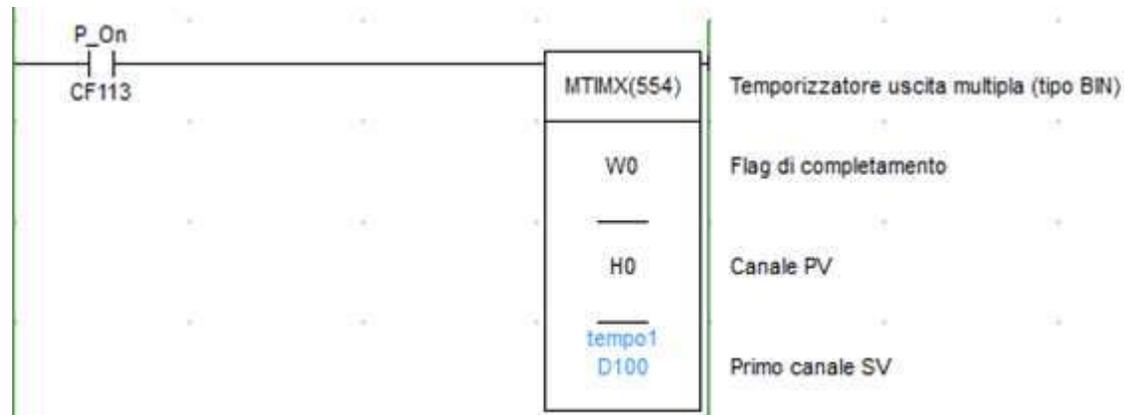
- L'utilizzo tipico per questa istruzione è quello di realizzare l'attivazione sequenziale di un certo numero di uscite attivate a tempi definiti. In questo esempio verranno attivate due uscite secondo il diagramma temporale illustrato.



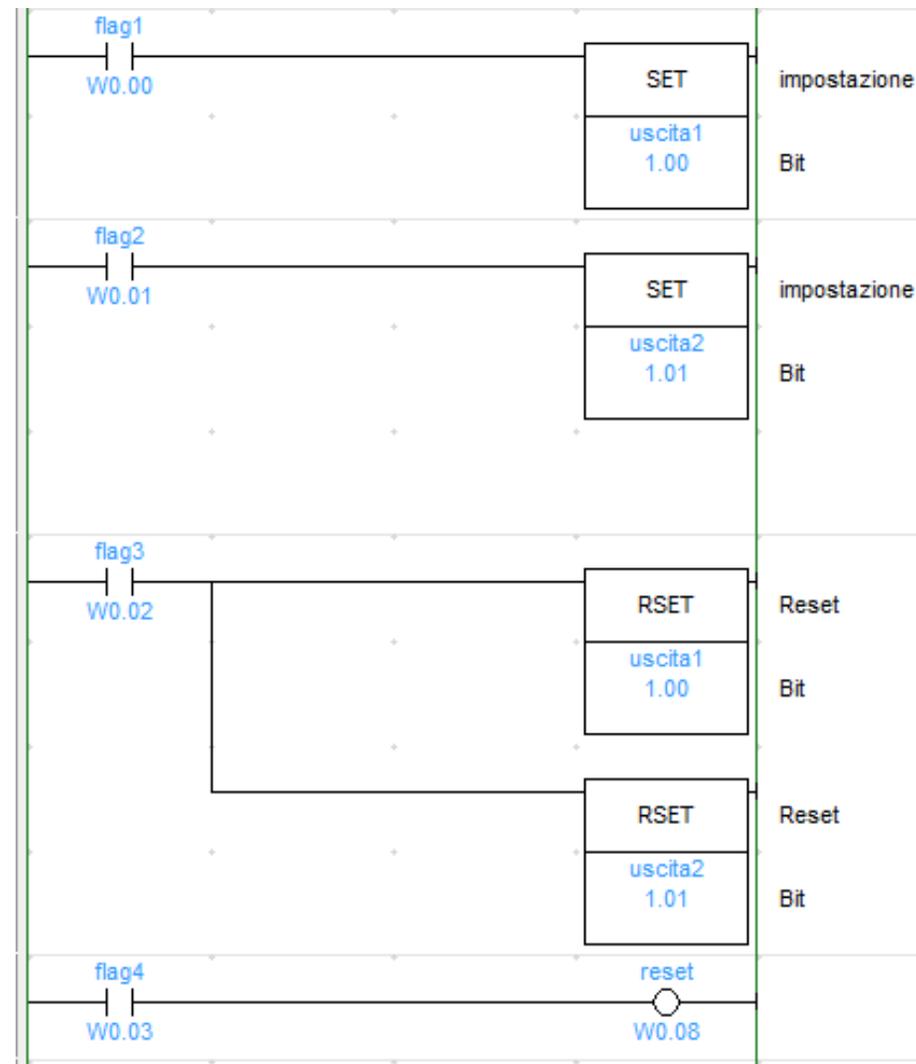
- Il ciclo inizia all'avvio del PLC.
- Dopo il tempo T4, il ciclo riprende dall'inizio.

Soluzione: impostazione del Multitimer

- Sarà necessario impostare 4 valori di setpoint memorizzandoli nei Data Memory D100, D101, D102 e D103.
- $T1 = 1s \rightarrow D100 = 10$.
- $T2 = 3s \rightarrow D101 = 40 (T1+T2)$.
- $T3 = 2s \rightarrow D102 = 60 (T1+T2+T3)$.
- $T4 = 2s \rightarrow D103 = 80 (T1+T2+T3+T4)$.

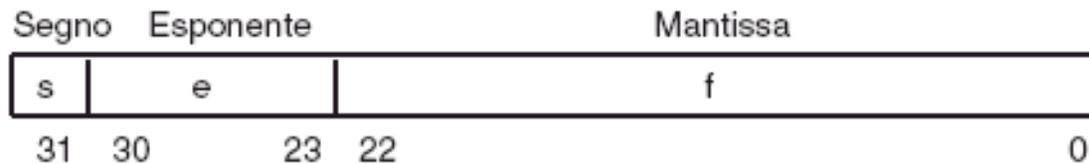


Soluzione: controllo delle uscite



Numeri in virgola mobile

Formato IEEE754 a singola precisione 32Bit

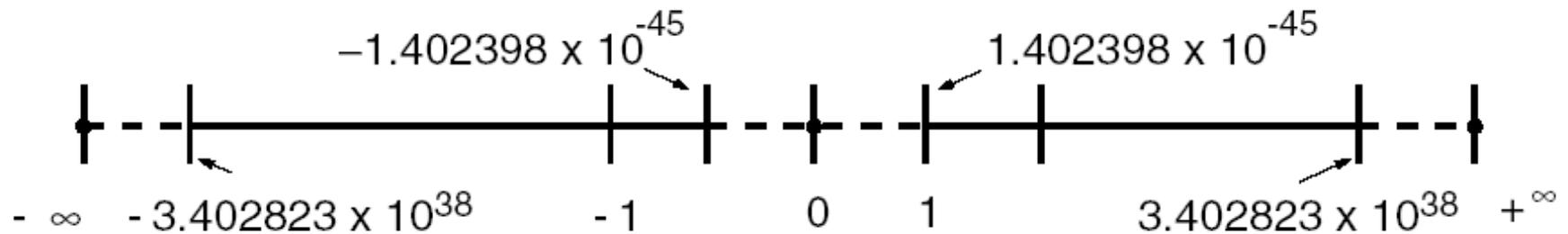


Dati	N. di bit	Contenuto
s: segno	1	0: positivo, 1: negativo
e: esponente	8	Il valore dell'esponente (e) è compreso tra 0 e 255. L'esponente effettivo è il valore che resta dopo la sottrazione di 127 da e; tale valore è compreso tra -127 e 128. "e=0" ed "e=255" esprimono numeri speciali.
f: mantissa	23	La parte di mantissa dei dati binari a virgola mobile corrisponde al formale $2.0 > 1.f \geq 1.0$.

$$\text{Valore} = (-1)^{\text{segno}} \times 1.(\text{Mantissa}) \times 2^{\text{Esponente}}$$

Numeri in virgola mobile

- Il numero delle cifre effettive per i dati a virgola mobile è 24 bit per il formato binario (approssimativamente sette cifre per il formato decimale).



Intervallo numerico rappresentabile in virgola mobile

Operazioni in virgola mobile

Istruzione	Mnemonico	Descrizione
FLOATING TO 16-BIT	FIX	Converte un valore float in un intero a 16 Bit (1 word).
FLOATING TO 32-BIT	FIXL	Converte un valore float in un intero a 32 Bit (2 word).
16-BIT TO FLOATING	FLT	Converte un valore intero a 16 Bit (1 word) in un float.
32-BIT TO FLOATING	FLTL	Converte un valore intero a 32 Bit (2 word) in un float.
FLOATING-POINT ADD	+F	Somma valori in virgola mobile.
FLOATING-POINT SUBTRACT	-F	Sottrae valori in virgola mobile.
FLOATING-POINT MULTIPLY	*F	Moltiplicazione in virgola mobile.
FLOATING-POINT DIVIDE	/F	Divisione in virgola mobile.

Operazioni in virgola mobile

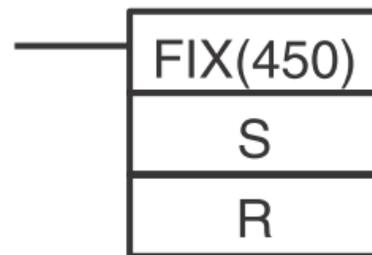
Istruzione	Mnemonico	Descrizione
DEGREES TO RADIANS	RAD	Converte in Radianti il valore dell'angolo espresso in Gradi.
RADIANS-TO-DEGREES	DEG	Converte in Gradi il valore dell'angolo espresso in Radianti.
SINE	SIN	Restituisce il Seno dell'angolo espresso in Radianti.
COSINE	COS	Restituisce il Coseno dell'angolo espresso in Radianti.
TANGENT	TAN	Restituisce il Coseno dell'angolo espresso in Radianti.
ARC SINE	ASIN	Restituisce l'angolo in radianti corrispondente al valore di seno impostato.
ARC COSINE	ACOS	Restituisce l'angolo in radianti corrispondente al coseno impostato.
ARC TANGENT	ATAN	Restituisce l'angolo in radianti corrispondente alla tangente impostata.

Operazioni in virgola mobile

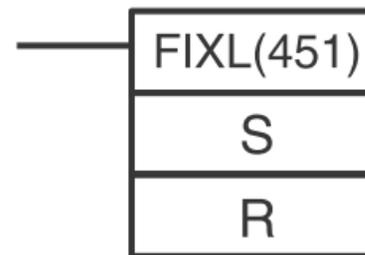
Istruzione	Mnemonico	Descrizione
SQUARE ROOT	SQRT	Estrae la radice quadrata.
EXPONENT	EXP	Eleva la base e (base dei logaritmi naturali) all'esponente specificato.
LOGARITHM	LOG	Calcola il Logaritmo naturale.
EXPONENTIAL POWER	PWR	Eleva la base X all'esponente Y.

Istruzioni di conversione FIX e FIXL

- FIX: converte un numero floating-point in un numero binario con segno (1 word).
- FIXL: converte un numero floating-point in un doppio binario con segno (2 word).
- Solo la parte intera viene convertita.



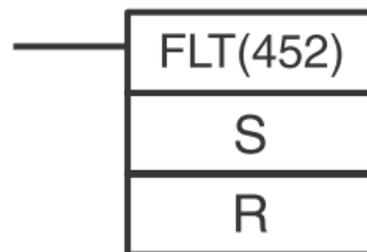
S: 1st source word
R: Result word



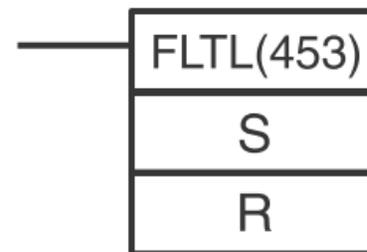
S: 1st source word
R: 1st result word

Istruzioni di conversione FLT e FLTL

- FLT: converte un numero binario con segno (1 word) in un numero floating-point.
- FLTL: converte un numero binario con segno (2 word) in un floating-point.
- Il risultato viene approssimato se il numero da convertire supera 16.777.215 (24 bit significativi).



S: Source word
R: 1st result word



S: 1st source word
R: 1st result word

Istruzioni in virgola mobile

Addizione
+F

Sottrazione
-F

Moltiplicazione
***F**

Divisione
/F

Name	Label	Operation
Error Flag	ER	ON if the dividend or divisor data is not recognized as floating-point data. ON if the dividend or divisor is not a number (NaN). ON if the dividend and divisor are both 0. ON if the dividend and divisor are both $+\infty / -\infty$. OFF in all other cases.
Equals Flag	=	ON if both the exponent and mantissa of the result are 0. OFF in all other cases.
Overflow Flag	OF	ON if the absolute value of the result is too large to be expressed as a 32-bit floating-point value.
Underflow Flag	UF	ON if the absolute value of the result is too small to be expressed as a 32-bit floating-point value.
Negative Flag	N	ON if the result is negative. OFF in all other cases.

Funzioni trigonometriche

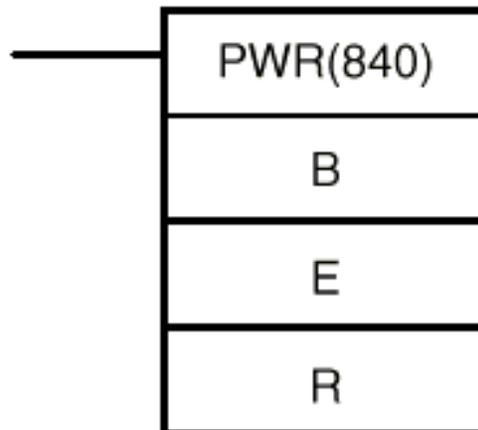
- RAD, DEG: conversione Gradi \leftrightarrow radianti.
- SIN, ASIN: seno e arcoseno (in radianti).
- COS, ACOS: coseno e arcocoseno.
- TAN, ATAN: tangente e arcotangente.

Funzioni esponenziali

- SQRT: radice quadrata (solo numeri positivi e floating-point).
- ROTB: radice quadrata in binario (solo parte intera del risultato).
- EXP: esponenziale $y = e^x$ (la costante e è pari a: 2,718282).
- LOG: logaritmo naturale (in base e).

Elevamento a potenza: PWR(840)

- Eleva un numero B a una potenza E --> B^E .
- B ed E sono valori floating-point.



B: First base word

E: First exponent word

R: First result word

Esempio: conversione di valore analogico

- In questo esempio si ipotizza di collegare un sensore di temperatura ambiente a un ingresso analogico del PLC.
- Il range di temperatura $-20.0 \div +60.0$ gradi centigradi viene tradotto in una tensione da 0 a 10 V.
- La scheda analogica convertirà la tensione in un valore numerico da 0 a 4000.

$$val_{conv} = \left(\frac{val_{ingresso}}{range_{ingresso}} \times range_{uscita} \right) - offset$$

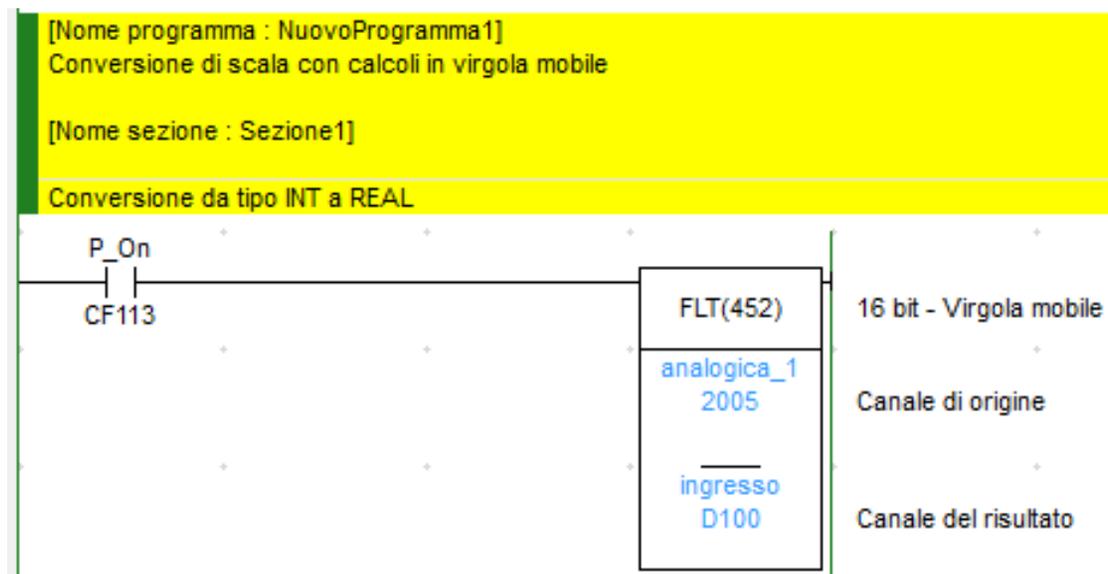
- Nel nostro caso, il range d'uscita è l'escursione massima della temperatura cioè 80 gradi, mentre l'offset vale -20. Il range d'ingresso è quello dell'analogica, 4000.

$$t = \left(\frac{input}{4000.0} \times 80.0 \right) - 20.0$$

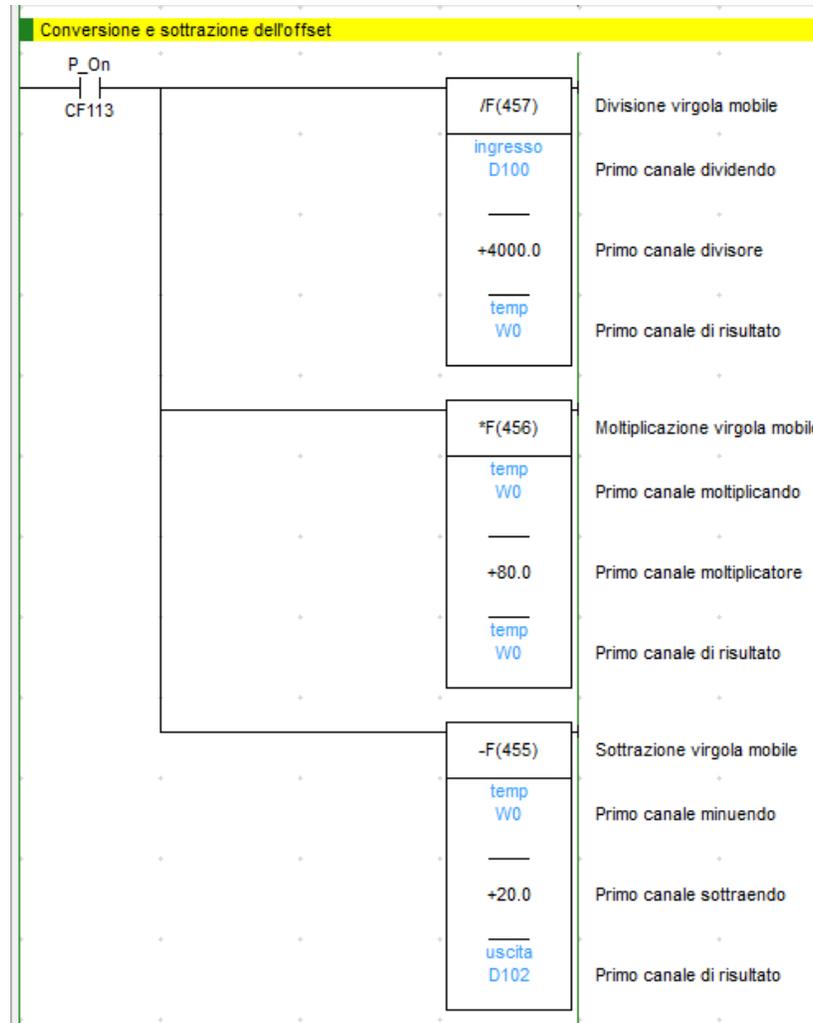
Traduzione in programma ladder

- Per valutare una formula in ladder è necessario distribuire le singole operazioni secondo l'ordine di priorità degli operatori matematici e memorizzare i risultati intermedi delle operazioni.
- La nostra formula sarà svolta secondo l'ordine:
 1. Dividere il valore in ingresso per 4000.0 e copiare il risultato in una variabile temporanea.
 2. Moltiplicare il valore temporaneo per 80.0 e copiare il risultato in una variabile temporanea.
 3. Sottrarre 20.0 dal valore temporaneo e copiare il risultato nella variabile di destinazione che conterrà il valore convertito.
- La prima operazione da compiere sarà la conversione dell'ingresso analogico da intero a virgola mobile (istruzione FLT).

Soluzione: conversione di valore analogico



Soluzione: conversione di valore analogico



Comparazioni in virgola mobile

- Sui PLC CS1 (non CPU CS1H), CJ2, CJ1 (non CPU CS1G/CJ1G) e su tutti i CP1 è possibile effettuare comparazioni tra valori in virgola mobile.
- Si utilizzano i soliti operatori di confronto (<, >, =, ecc...) seguiti dal suffisso F.
- Non è consigliabile utilizzare l'operazione di '=' per valori diversi da 0.
- E' consigliabile realizzare delle comparazioni 'a finestra', cioè verificare se il valore è compreso in un intervallo prefissato.

Doppia precisione

- Sui PLC CS1 (non CPU CS1H), CJ2, CJ1 (non CPU CS1G/CJ1G) e CP1 (non CPU CP1E) è disponibile anche il formato virgola mobile a doppia precisione.
- Per ogni valore vengono utilizzati 64 bit (4 word) invece dei 32 bit della singola precisione.
- Un valore in doppia precisione è composto da 1 bit di segno 52 bit di mantissa e 11 bit di esponente.
- Il tipo di dato in CX-Programmer è LREAL.
- Per ogni istruzione in virgola mobile esiste la variante in doppia precisione.

Doppia precisione

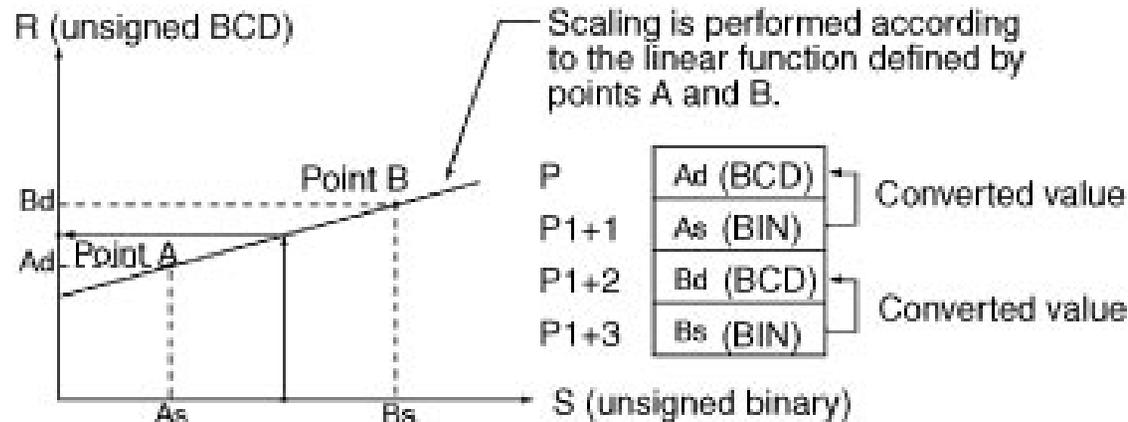
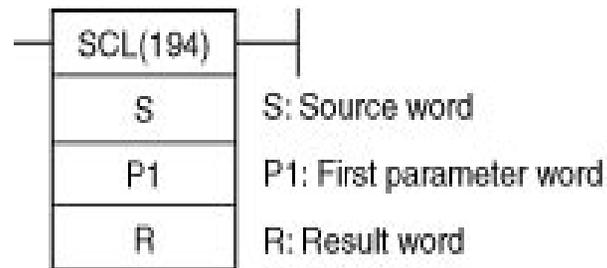
- Istruzioni in doppia precisione:
 - Operatori matematici (suffisso D invece di F):
`+D, -D, *D e /D;`
 - Funzioni trigonometriche ed esponenziali (aggiunta del suffisso D al nome della funzione):
`COSD, PWRD, EXPD, ecc...;`
 - Conversione da intero a doppia precisione:
`DBL e DBLL;`
 - Conversione da doppia precisione a intero:
`FIXD e FIXLD;`
 - Comparazioni (suffisso D invece di F):
`=D, >D, <D, ecc...;`

Altre istruzioni di conversione: SCL(194)

- Tipicamente i dati trattati dalle schede analogiche sono di tipo esadecimale.
- L'istruzione di 'scaling' SCL(194) permette di eseguire conversioni lineari da dati esadecimali (BIN) a dati decimali (BCD).
- La conversione avviene linearmente secondo una retta di conversione definita da due punti.
- Le quattro coordinate che definiscono i due punti della retta di conversione devono essere specificate in una tabella a partire dal canale 'P1'.

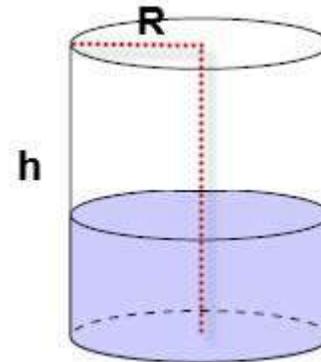
Altre istruzioni di conversione: SCL(194)

- Le quattro coordinate che definiscono i due punti della retta di conversione devono essere specificate in una tabella a partire dal canale 'P1'.



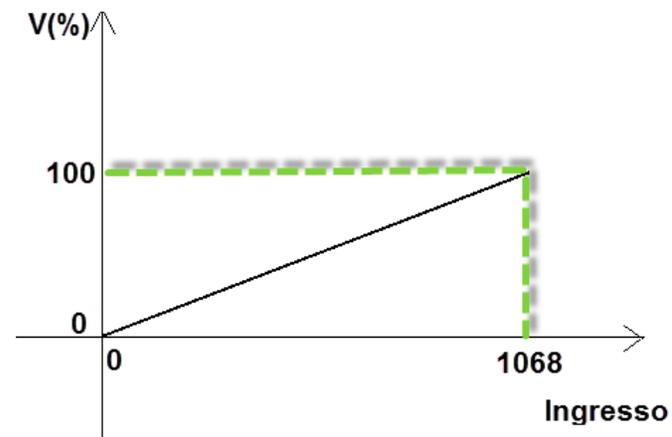
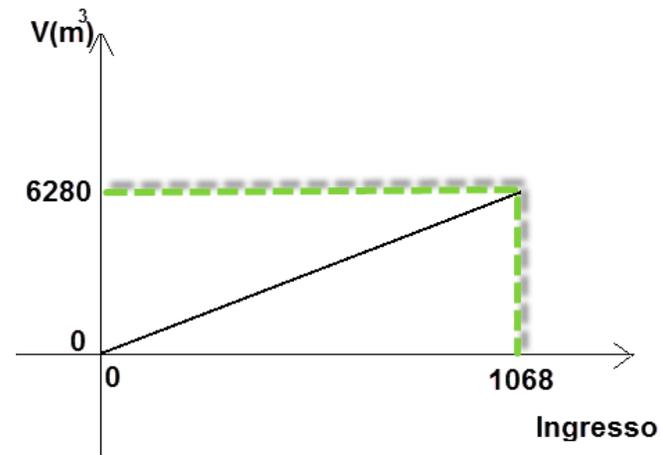
Esempio: conversione lineare SCL

- Visualizzare in due canali del PLC (per esempio H0 e H1), la quantità volumetrica e percentuale di un liquido all'interno di un serbatoio di forma cilindrica, in base al segnale proveniente da un sensore di livello collegato ad un ingresso analogico del PLC (CIO 2005).
- $R = 5 \text{ m}$ (raggio).
- $h = 20 \text{ m}$ (altezza).
- Volume cilindro = $h * \pi R^2 = 1570 \text{ m}^3$.

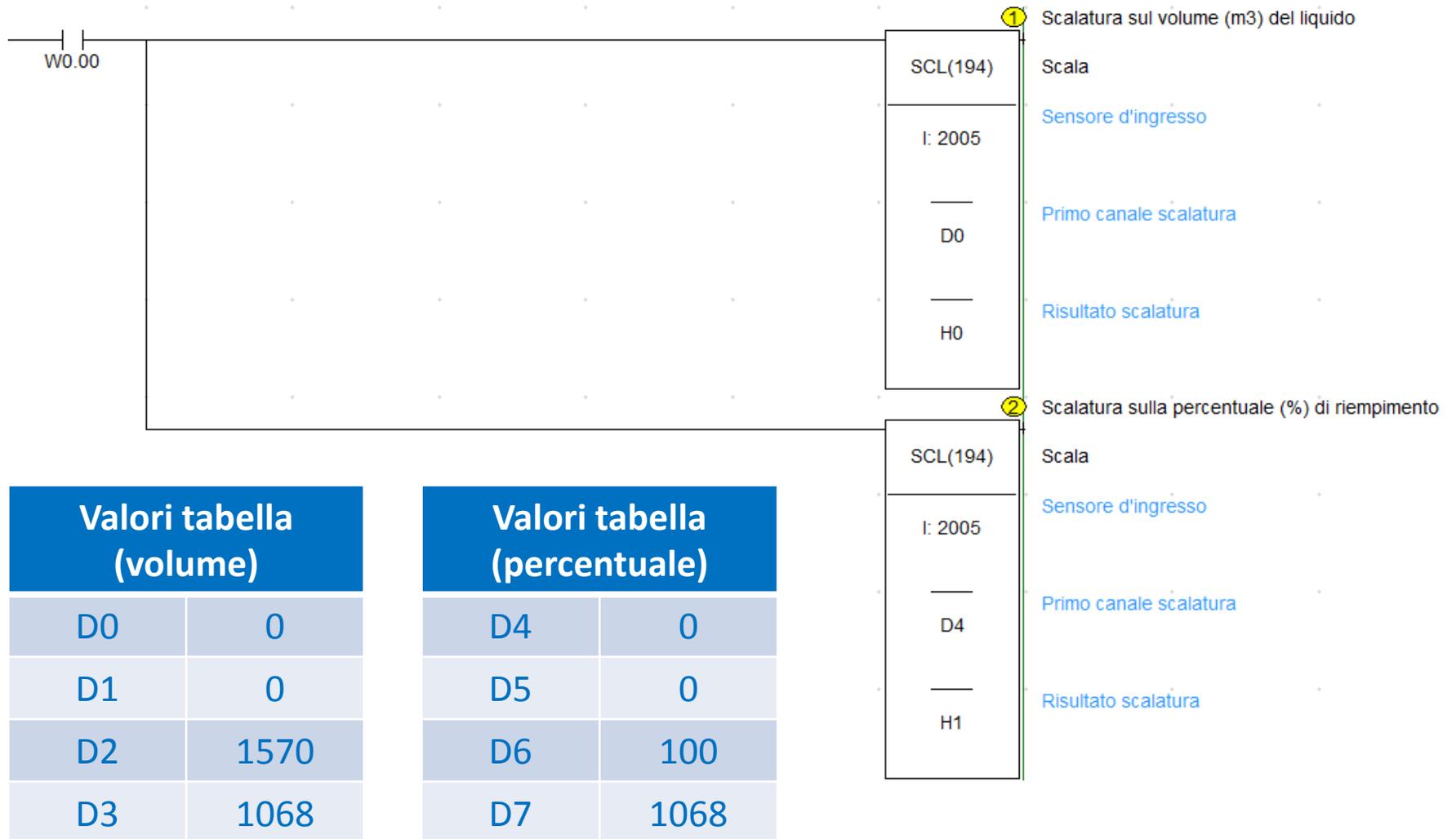


Esempio: conversione lineare SCL

- Il volume del liquido è direttamente proporzionale al valore in ingresso proveniente dal sensore.
- Lo stesso vale per la percentuale di riempimento del cilindro.

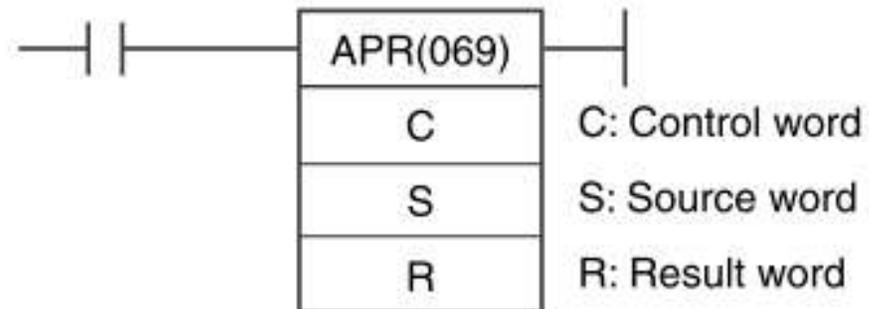
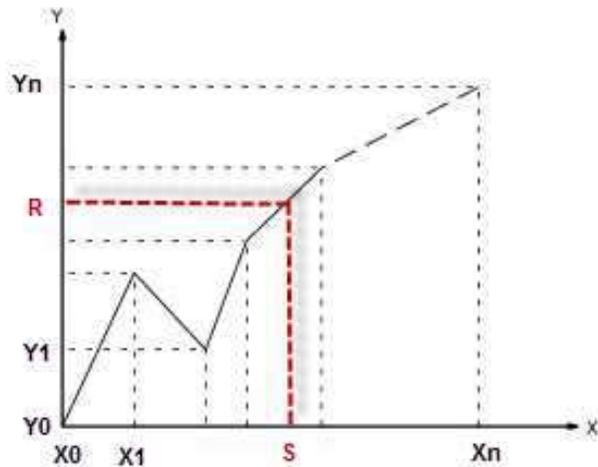


Esempio: conversione lineare SCL



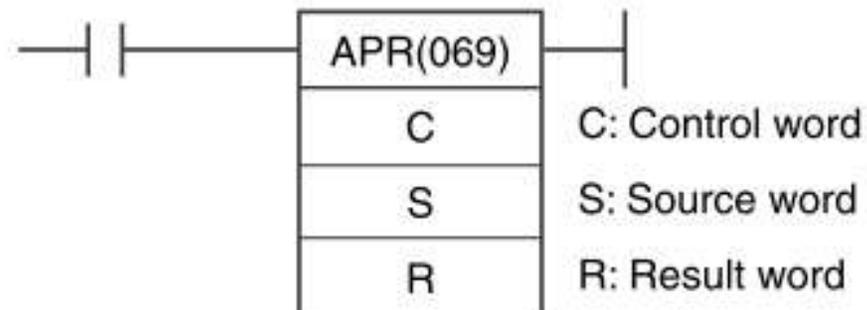
Altre istruzioni di conversione: APR(069)

- L'istruzione 'arithmetic process' APR(069) consente di calcolare il seno, il coseno e l'estrapolazione lineare di un dato in ingresso.



APR(069): canale di controllo

- Il primo operando dell'istruzione APR(069) identifica il canale di controllo 'C'; in base alla sua definizione è possibile eseguire le seguenti funzioni:
 - Se C = #0000 → Funzione seno;
 - Se C = #0001 → Funzione coseno.
 - Se C è un'area di memoria → Estrapolazione lineare.



APR(069): canale di controllo

- Scegliendo, ad esempio, un'area di memoria per il canale di controllo, è possibile utilizzare le seguenti tipologie di dato per la definizione delle coordinate:
 - Intero senza segno (BIN o BCD) a 16 bit;
 - Intero con segno (BIN) a 16 bit;
 - Intero con segno (BIN) a 32 bit;
 - Floating-point a singola precisione (REAL) a 32 bit.
- Le ultime tre tipologie di dato sono supportate dai PLC della serie CJ2, CS1-H, CJ1-H, CJ1M e CS1D.

APR(069): canale di controllo

- Intero senza segno (BIN o BCD) a 16 bit e intero con segno (BIN) a 16 bit:

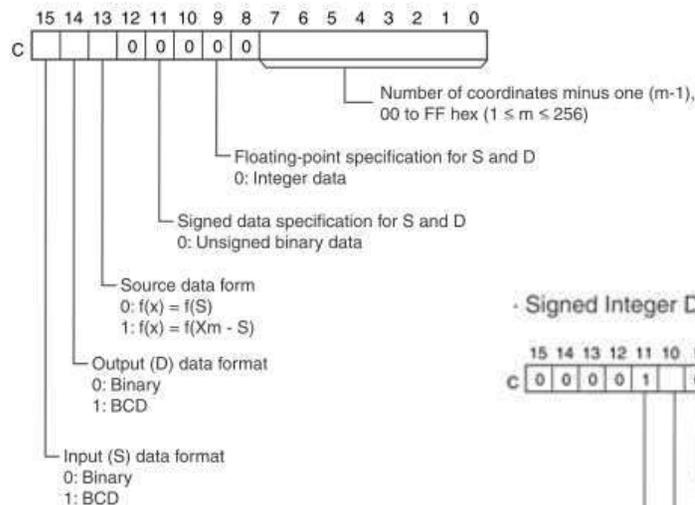
16-bit BCD/16-bit binary (signed or unsigned) or 16-bit BCD data

C+1	X0 (*1)
C+2	Y0
C+3	X1
C+4	Y1
C+5	X2
C+6	Y2
	Xn
	Yn
C+ (2m+1)	Xm
C+ (2m+2)	Ym

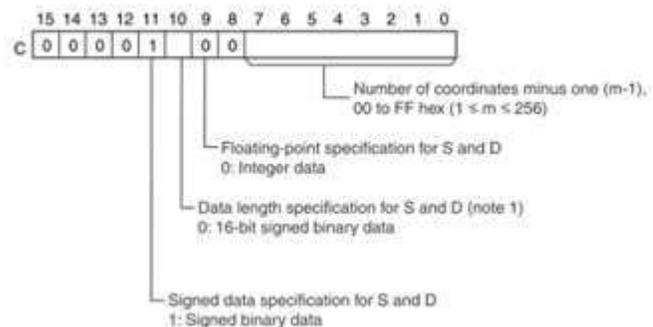
Note: Write X_m (max. X value in the table) in word C+1 when the I/O data in S and D contain unsigned data (bit 11 of C = 0).

Note: The X coordinates must be in ascending order: X₁ < X₂ < ... < X_m. Input all values of (X_n, Y_n) as binary data, regardless of the data format specified in control word C.

· Unsigned Integer Data (Binary or BCD)



· Signed Integer Data (Binary)

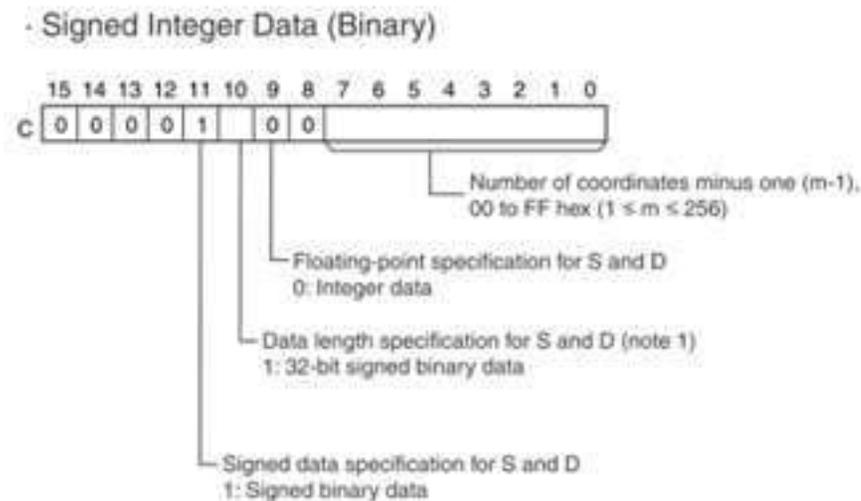


APR(069): canale di controllo

- Intero con segno (BIN) a 32 bit:

32-bit signed binary data

C+1	X0 (rightmost 16 bits)
C+2	X0 (leftmost 16 bits)
C+3	Y0 (rightmost 16 bits)
C+4	Y0 (leftmost 16 bits)
C+5	X1 (rightmost 16 bits)
C+6	X1 (leftmost 16 bits)
C+7	Y1 (rightmost 16 bits)
C+8	Y1 (leftmost 16 bits)
to	to
C+ (4n+1)	Xn (rightmost 16 bits)
C+ (4n+2)	Xn (leftmost 16 bits)
C+ (4n+3)	Yn (rightmost 16 bits)
C+ (4n+4)	Yn (leftmost 16 bits)
to	to
C+ (4m+1)	Xm (rightmost 16 bits)
C+ (4m+2)	Xm (leftmost 16 bits)
C+ (4m+3)	Ym (rightmost 16 bits)
C+ (4m+4)	Ym (leftmost 16 bits)



Note: The X coordinates must be in ascending order: $X_1 < X_2 < \dots < X_m$. Input all values of (X_n, Y_n) as binary data, regardless of the data format specified in control word C.

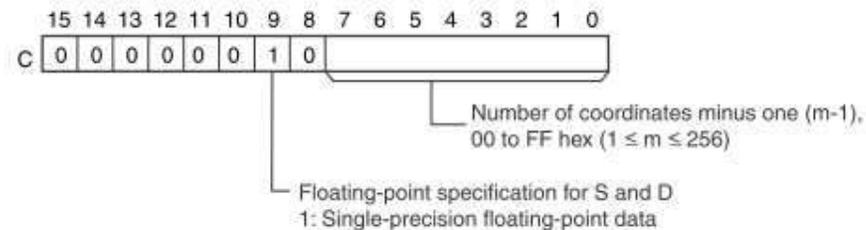
APR(069): canale di controllo

- Floating-point a singola precisione (REAL) a 32 bit:

Floating-point data

C+1	X0 (rightmost 16 bits)
C+2	X0 (leftmost 16 bits)
C+3	Y0 (rightmost 16 bits)
C+4	Y0 (leftmost 16 bits)
C+5	X1 (rightmost 16 bits)
C+6	X1 (leftmost 16 bits)
C+7	Y1 (rightmost 16 bits)
C+8	Y1 (leftmost 16 bits)
to	to
C+ (4n+1)	Xn (rightmost 16 bits)
C+ (4n+2)	Xn (leftmost 16 bits)
C+ (4n+3)	Yn (rightmost 16 bits)
C+ (4n+4)	Yn (leftmost 16 bits)
to	to
C+ (4m+1)	Xm (rightmost 16 bits)
C+ (4m+2)	Xm (leftmost 16 bits)
C+ (4m+3)	Ym (rightmost 16 bits)
C+ (4m+4)	Ym (leftmost 16 bits)

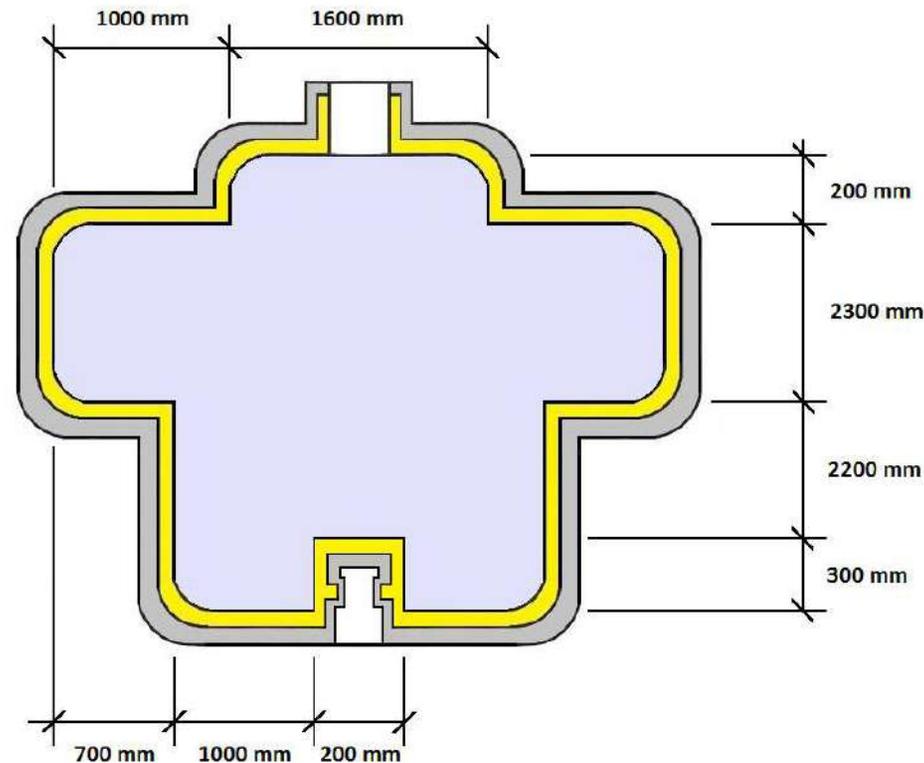
Single-precision Floating-point Data



Note: The X coordinates must be in ascending order: $X_1 < X_2 < \dots < X_m$. Input all values of (X_n, Y_n) as binary data, regardless of the data format specified in control word C.

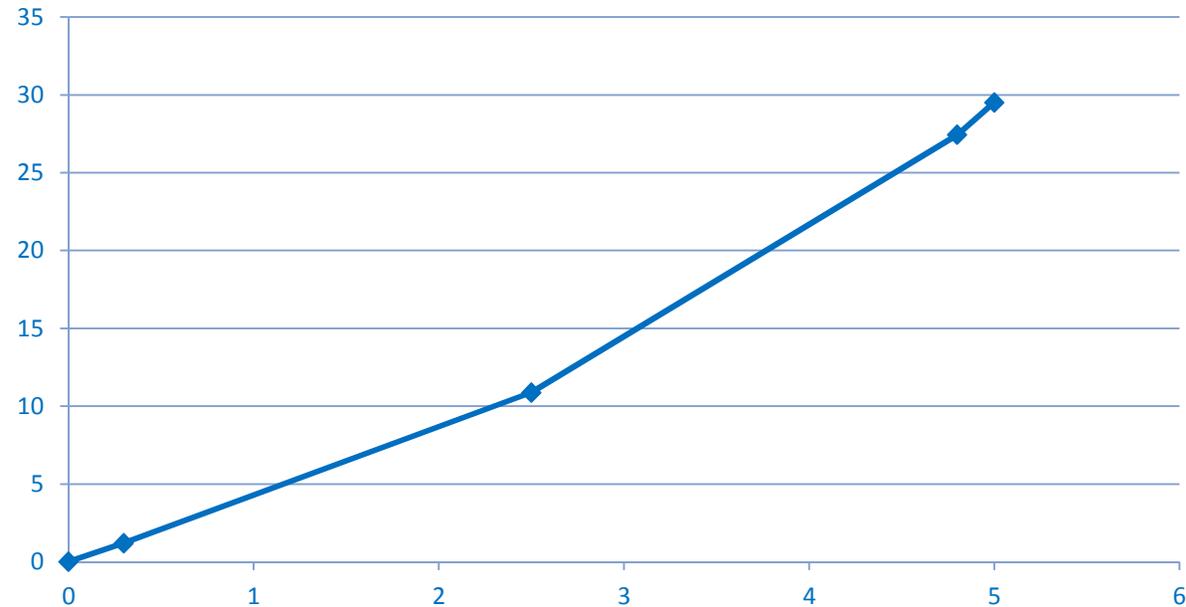
APR(069): Esempio

- Utilizzando l'istruzione APR(069) convertire l'altezza del fluido nel volume di acqua presente nel serbatoio della figura sottostante. La profondità del serbatoio è di 2 metri.



APR(069): Esempio

Altezza h	Volume V
0	0
0.3	1.2
2.5	10.88
4.8	27.44
5	29.52



PLC Name	Nome	Indirizzo	Tipo / Formato dati	Utilizzo...	Valore	Valore(binario)
NuovoPLC1	CONTROL WORD	H10	CHANNEL (Esa,Canale)		8&516	0000 0010 0000 0100
NuovoPLC1		H11	REAL (Virgola mobile,Lunghezza dop...		+0.0000000 Float	+0.0000000 Float
NuovoPLC1		H13	REAL (Virgola mobile,Lunghezza dop...		+0.0000000 Float	+0.0000000 Float
NuovoPLC1		H15	REAL (Virgola mobile,Lunghezza dop...		+0.3000000 Float	+0.3000000 Float
NuovoPLC1		H17	REAL (Virgola mobile,Lunghezza dop...		+1.2000000 Float	+1.2000000 Float
NuovoPLC1		H19	REAL (Virgola mobile,Lunghezza dop...		+2.5000000 Float	+2.5000000 Float
NuovoPLC1		H21	REAL (Virgola mobile,Lunghezza dop...		+10.8800000 Float	+10.8800000 Float
NuovoPLC1		H23	REAL (Virgola mobile,Lunghezza dop...		+4.8000000 Float	+4.8000000 Float
NuovoPLC1		H25	REAL (Virgola mobile,Lunghezza dop...		+27.4400000 Float	+27.4400000 Float
NuovoPLC1		H27	REAL (Virgola mobile,Lunghezza dop...		+5.0000000 Float	+5.0000000 Float
NuovoPLC1		H29	REAL (Virgola mobile,Lunghezza dop...		+29.5200000 Float	+29.5200000 Float
NuovoPLC1	SOURCE WORD	D200	REAL (Virgola mobile,Lunghezza dop...		+4.9000000 Float	+4.9000000 Float
NuovoPLC1	RESULT WORD	D300	REAL (Virgola mobile,Lunghezza dop...		+28.4800000 Float	+28.4800000 Float

APR(069): Esempio



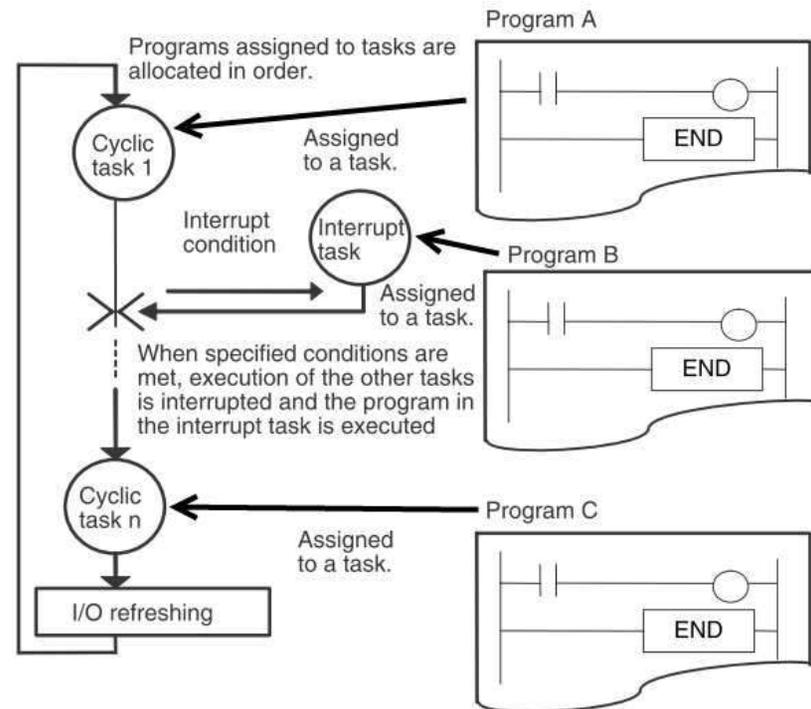


Task e Subroutine

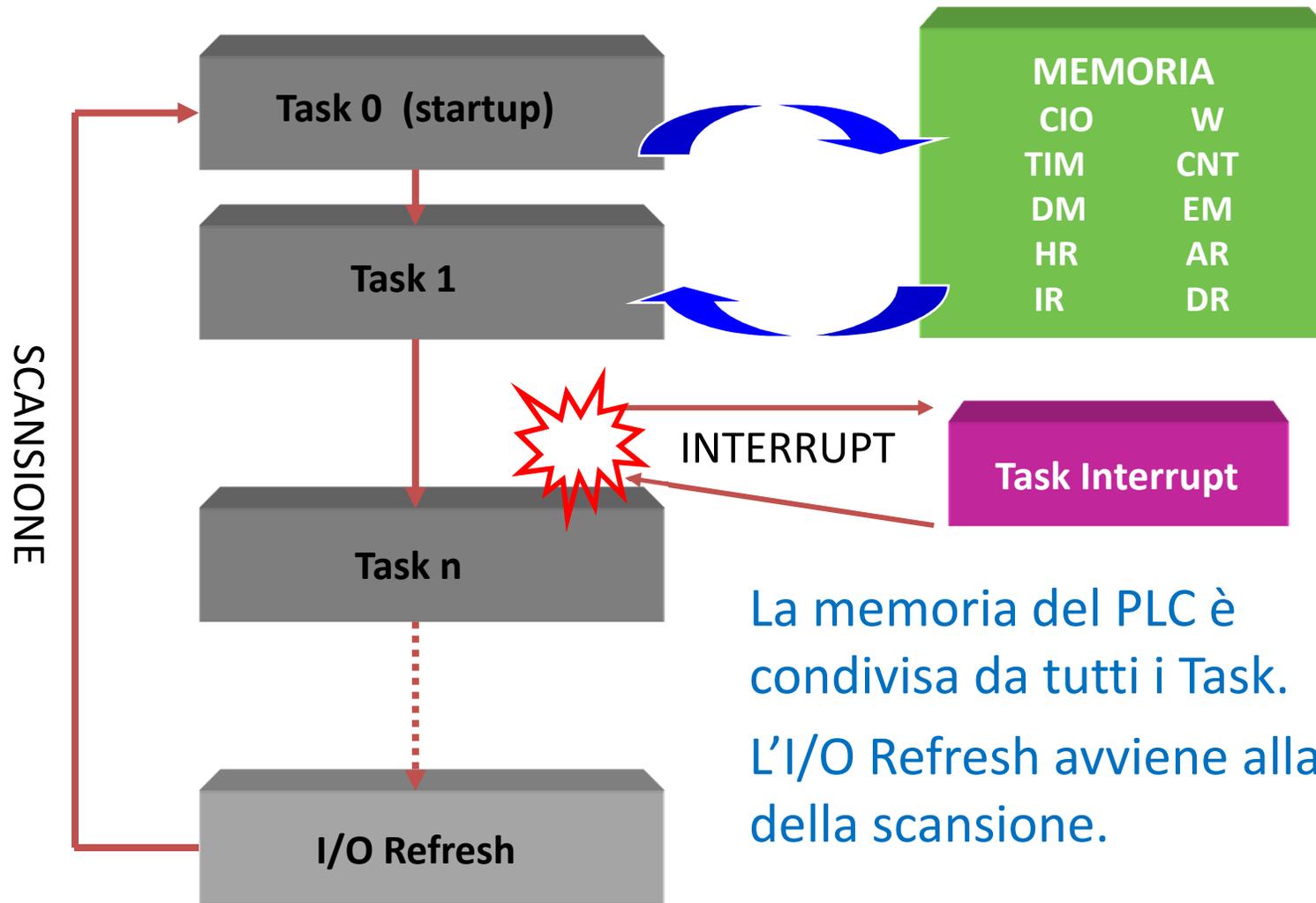
Corso intermedio CPS2

Programmazione a Task

- Il programma del PLC può essere suddiviso in Task.
- Ogni singolo Task è un programma a sé stante.
- I Task si suddividono in due categorie:
 - Task Ciclici, controllabili da programma e legati a una scansione ciclica.
 - Task a Interrupt, invocati dal verificarsi di determinate condizioni **HW/SW**.



Esecuzione dei Task



La memoria del PLC è condivisa da tutti i Task.
L'I/O Refresh avviene alla fine della scansione.

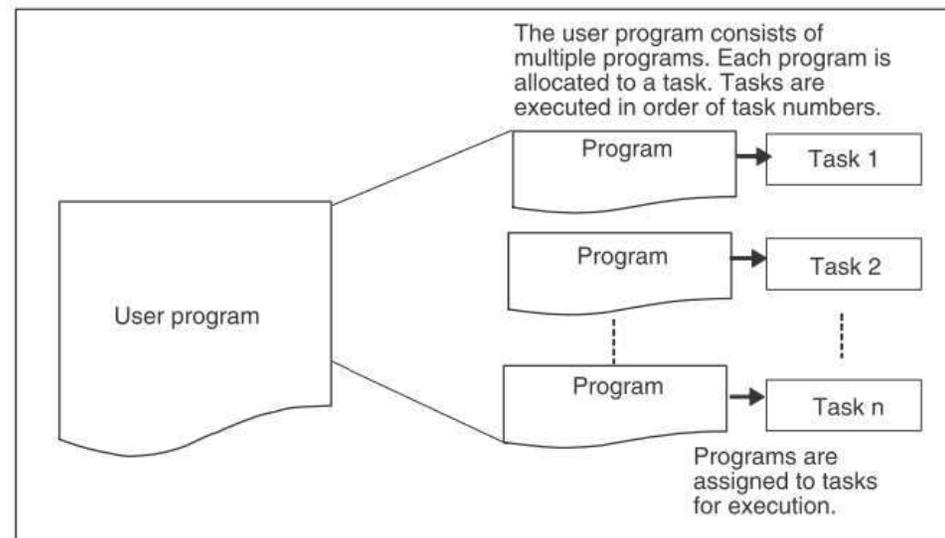
Vantaggi della divisione in Task

- **Suddivisione logica del programma.**
 - Suddividendo il programma in unità funzionali indipendenti è poi possibile assegnare un Task a ciascuna unità.
- **Ripartizione del lavoro tra gruppi.**
 - Le unità di programma possono essere assegnate a programmatori diversi.
- **Creazione di applicazioni multi-linguaggio.**
 - Ciascun Task può essere scritto in uno dei tre linguaggi supportati dal PLC: ladder, adatto alle funzioni logiche; Testo Strutturato (ST) adatto alla manipolazione dati e SFC, orientato ai programmi sequenziali.

Task Ciclici

- I Task Ciclici vengono eseguiti consecutivamente secondo il loro ordine numerico (0 -> 1 -> 2 ecc...).
- Ogni Task può essere attivato con l'istruzione TKON (task ON) e disattivato con l'istruzione TKOF (task OFF).
- Si può scegliere quali Task devono essere attivi all'avvio del PLC.
- Su tutte le CPU CJ2 si ha a disposizione 128 Task Ciclici numerati da 0 a 127.
- Si possono anche utilizzare i Task ad interrupt liberi.
- Sui PLC CJ1 e CP1 sono disponibili 32 Task Ciclici.

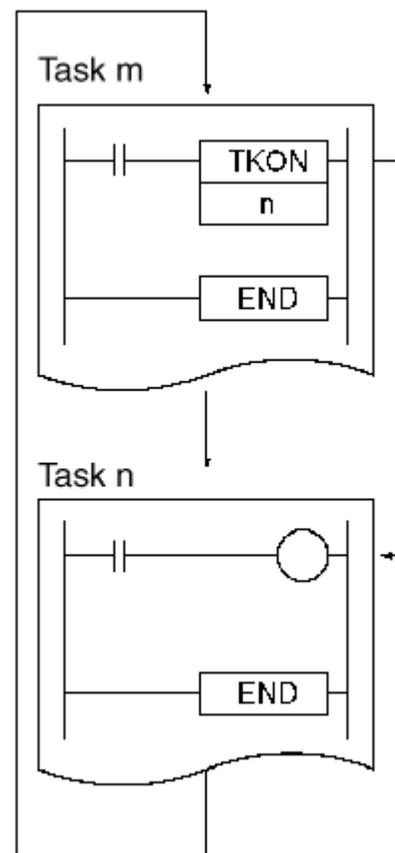
CPU Unit



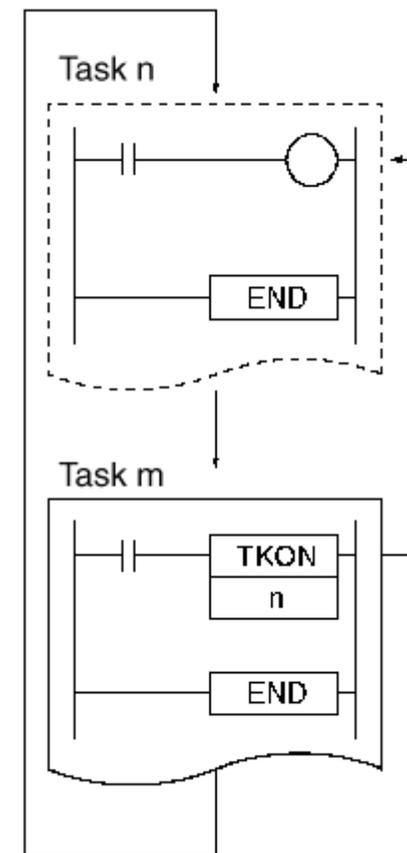
Attivazione task: TKON

- Il parametro indica qual è il Task da attivare.
- Rende eseguibile il Task indicato, e aggiorna il flag corrispondente.
- L'inizio esecuzione dipende dal numero dei Task.

$M < N$



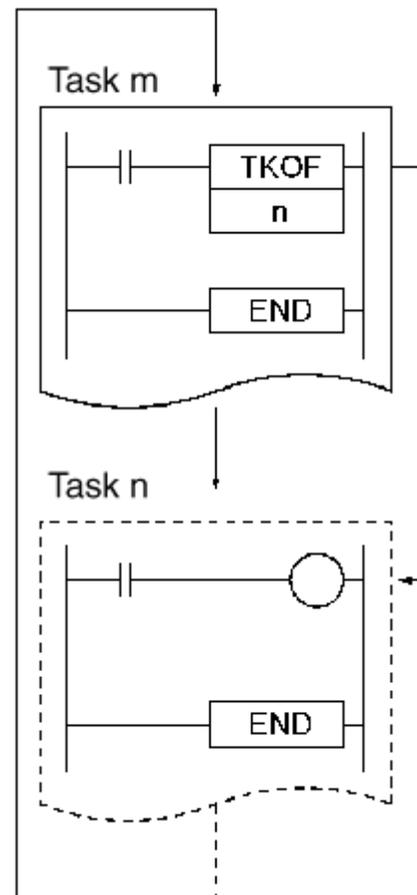
$M > N$



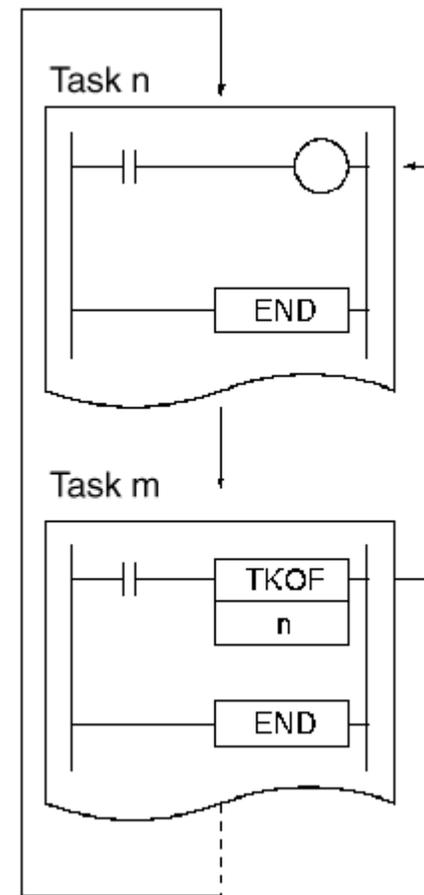
Disattivazione task: TKOF

- Mette in stand-by il Task indicato, e aggiorna il flag corrispondente.
- Il Task rimane inattivo anche nei cicli seguenti.

$M < N$



$M > N$



Subroutine

- Quando un insieme di istruzioni viene ripetuto più volte all'interno del programma, conviene definirlo come 'sottoprogramma' o Subroutine.
- Utilizzando le Subroutine si hanno due vantaggi importanti:
 - Risparmio di memoria di programma (si eliminano ripetizioni inutili);
 - Organizzazione modulare (facilità di lettura e manutenzione).
- Si hanno a disposizione 1.024(*) Subroutine, divisibili in:
 - Subroutine Locali: richiamabili solo dal task in cui sono state create;
 - Subroutine Globali: richiamabili da tutti i task del programma.

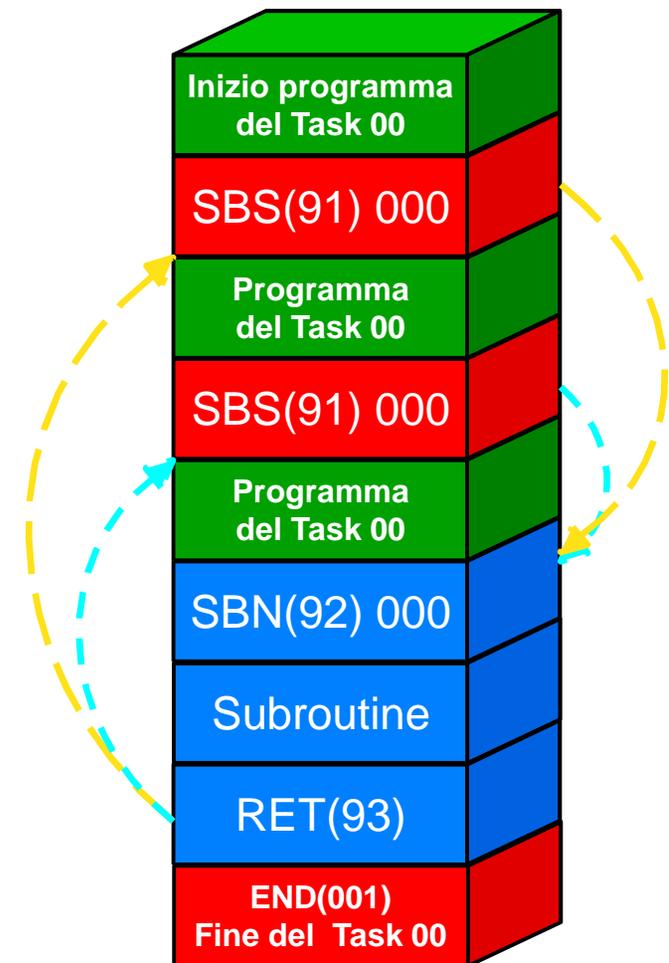
(*): Su alcuni modelli di PLC CJ1 e CP1, il numero è inferiore.

Subroutine Locali

- Tutte le Subroutine Locali devono essere inserite dopo l'ultima istruzione del Task e prima dell'istruzione END(001).
- Ciascuna Subroutine deve essere compresa tra le istruzioni SBN(092) e RET(093), e può essere richiamata tramite l'istruzione SBS(091):
 - **SBN(092)** Subroutine Number: definisce l'inizio di una Subroutine e ne specifica il numero (da 0 a 1.023).
 - **RET(093)** Return: definisce la fine di una Subroutine.
 - **SBS(091)** Subroutine Start: definisce la chiamata alla Subroutine specificata.

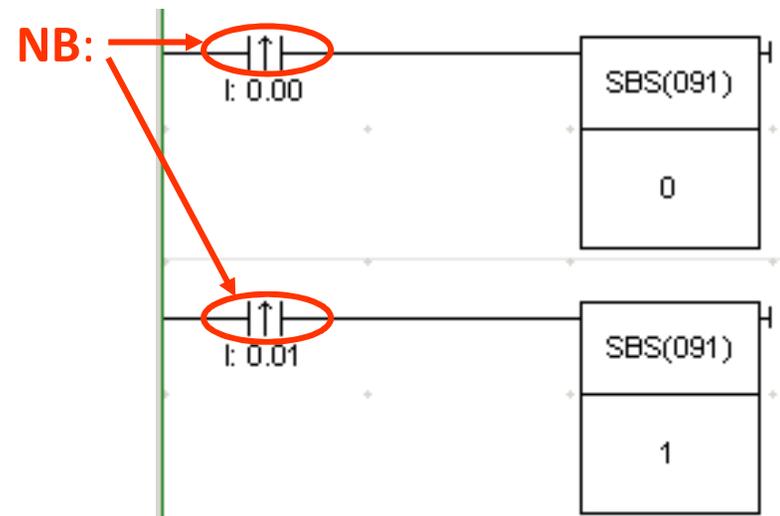
Subroutine Locali

- Se necessario una Subroutine può essere richiamata più volte, utilizzando l'istruzione SBS in più punti del programma all'interno del Task.
- **NB:** una Subroutine non può richiamare sé stessa.

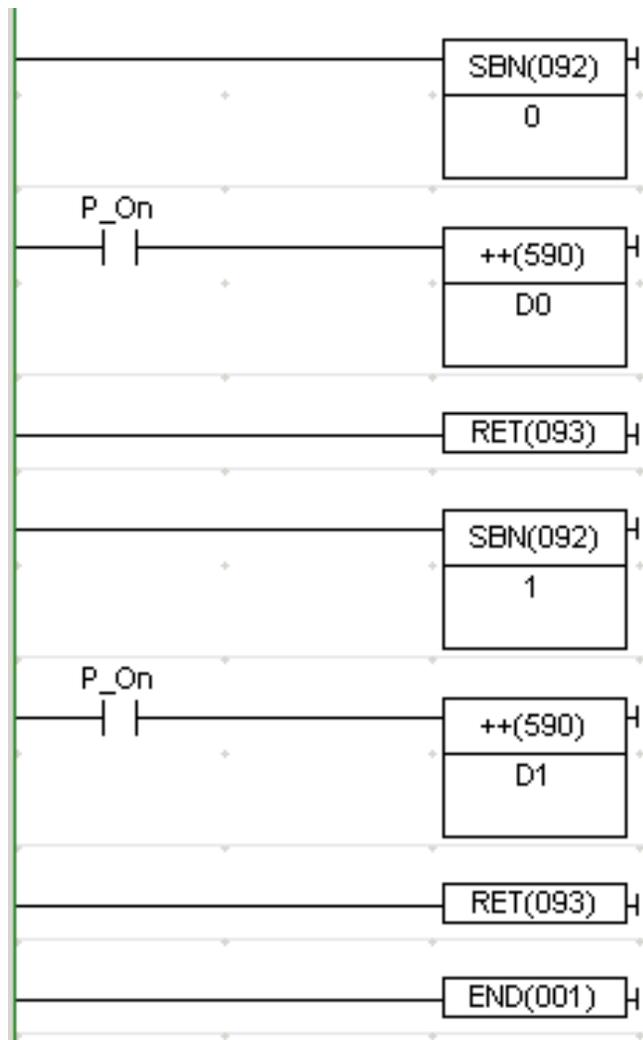


Subroutine Locali: esempio

- All'interno per esempio del Task Ciclico 00, inserisco le condizioni di chiamata alle Subroutine Locali 0 e 1.
- Successivamente, prima dell'istruzione END(001), definisco le due Subroutine.
- L'ingresso 0.00 chiama la Subroutine n° 0
- L'ingresso 0.01 chiama la Subroutine n° 1



Subroutine Locali: esempio



Definizione della Subroutine 0.

Chiamando la Subroutine 0 viene incrementato il DM0.

Fine della Subroutine 0.

Definizione della Subroutine 1.

Chiamando la Subroutine 1 viene incrementato il DM1.

Fine della Subroutine 1.

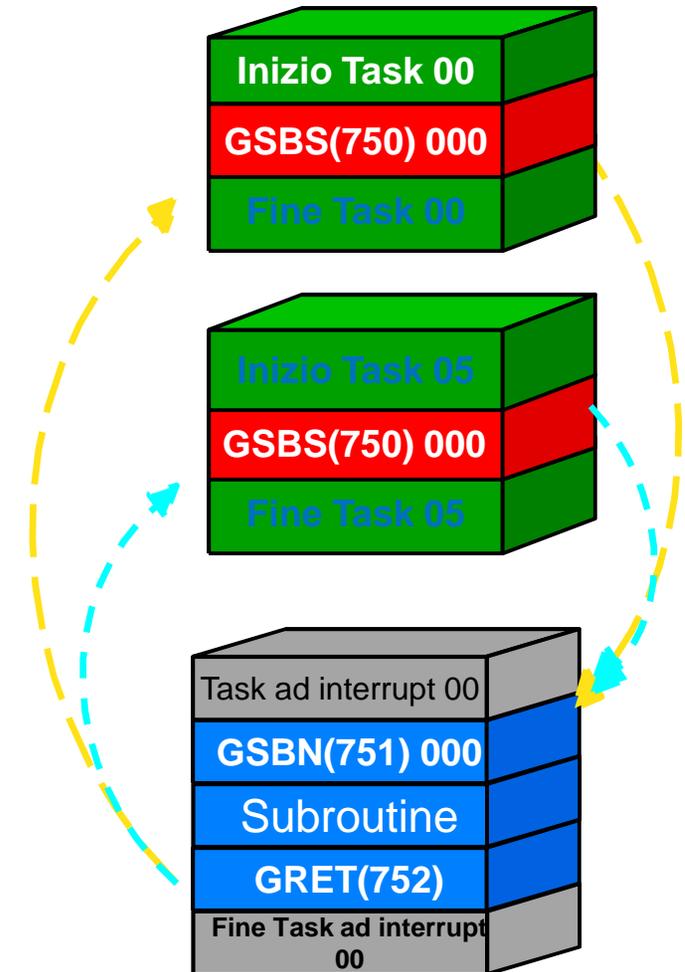
Fine del Task 00.

Subroutine Globali

- Ciascuna subroutine globale deve essere compresa tra le istruzioni GSBN(751) e GRET(752) e può essere richiamata da qualsiasi altro task tramite l'istruzione GSBS(750):
 - **GSBN(092)** Global Subroutine Number: definisce l'inizio di una Subroutine Globale e ne specifica il numero (da 0 a 1.023).
 - **GRET(093)** Global Subroutine Return: definisce la fine di una Subroutine Globale.
 - **GSBS(091)** Global Subroutine Start: definisce la chiamata alla Subroutine Globale specificata.

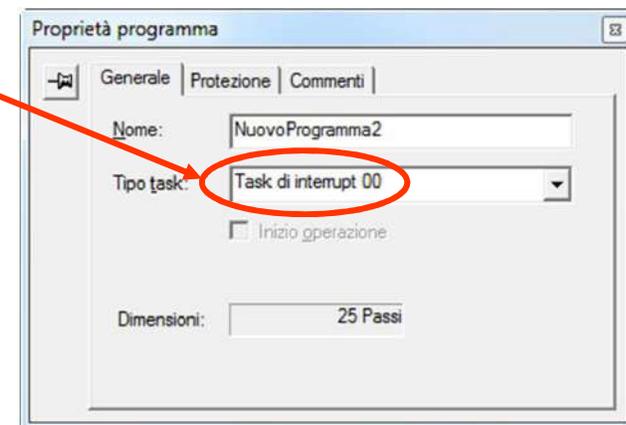
Subroutine Globali

- La stessa Subroutine Globale può essere richiamata da Task differenti utilizzando l'istruzione GSBS(750) relativa a quella Subroutine.
- **NB:** Occorre prestare attenzione a non assegnare ad una Subroutine locale un numero già riservato ad una Subroutine Globale e viceversa. In caso contrario CX-Programmer genererà un errore in fase di compilazione e non sarà possibile scaricare il programma nel PLC.



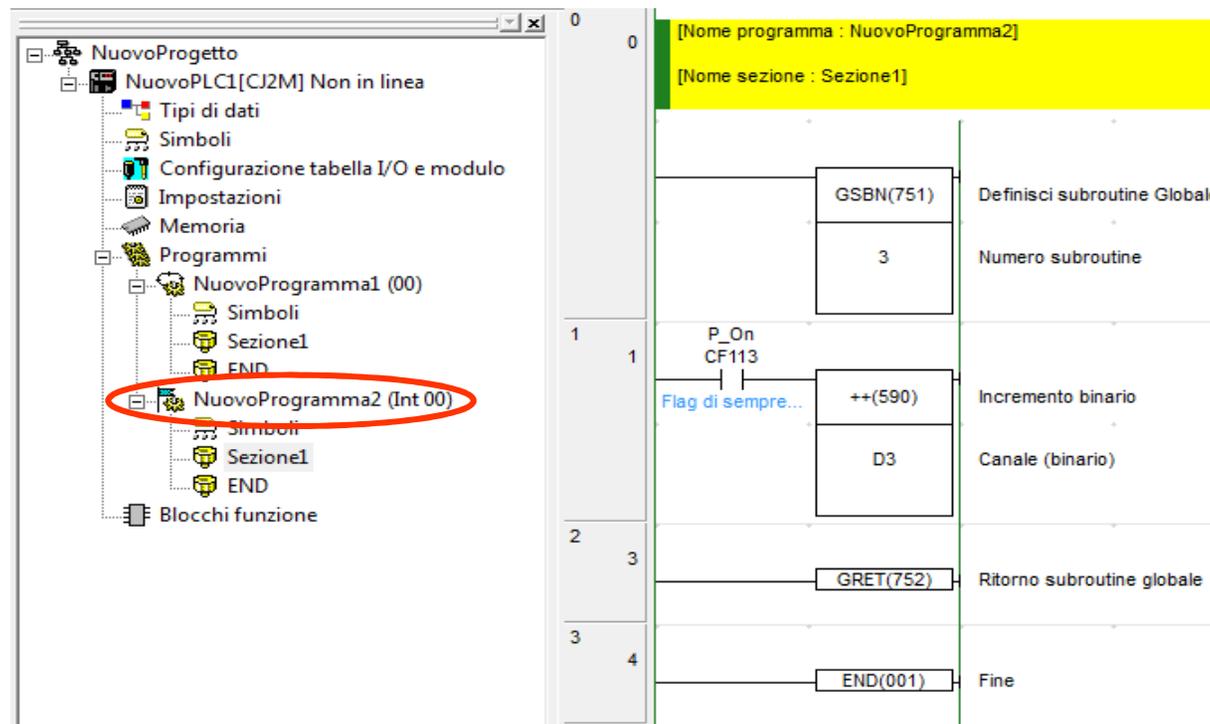
Subroutine Globali

- Tutte le Subroutine Globali devono essere definite all'interno del Task ad Interrupt '00'.
- Per creare il Task ad interrupt 00 occorre:
 - Cliccare sulla barra degli strumenti l'icona  per aggiungere un nuovo Task al programma.
 - Cliccare con il tasto destro sul nuovo Task e selezionare 'Proprietà' per accedere alla finestra di impostazioni del Task.
 - Selezionare come tipo di Task "Task di interrupt 00" e premere INVIO per confermare.



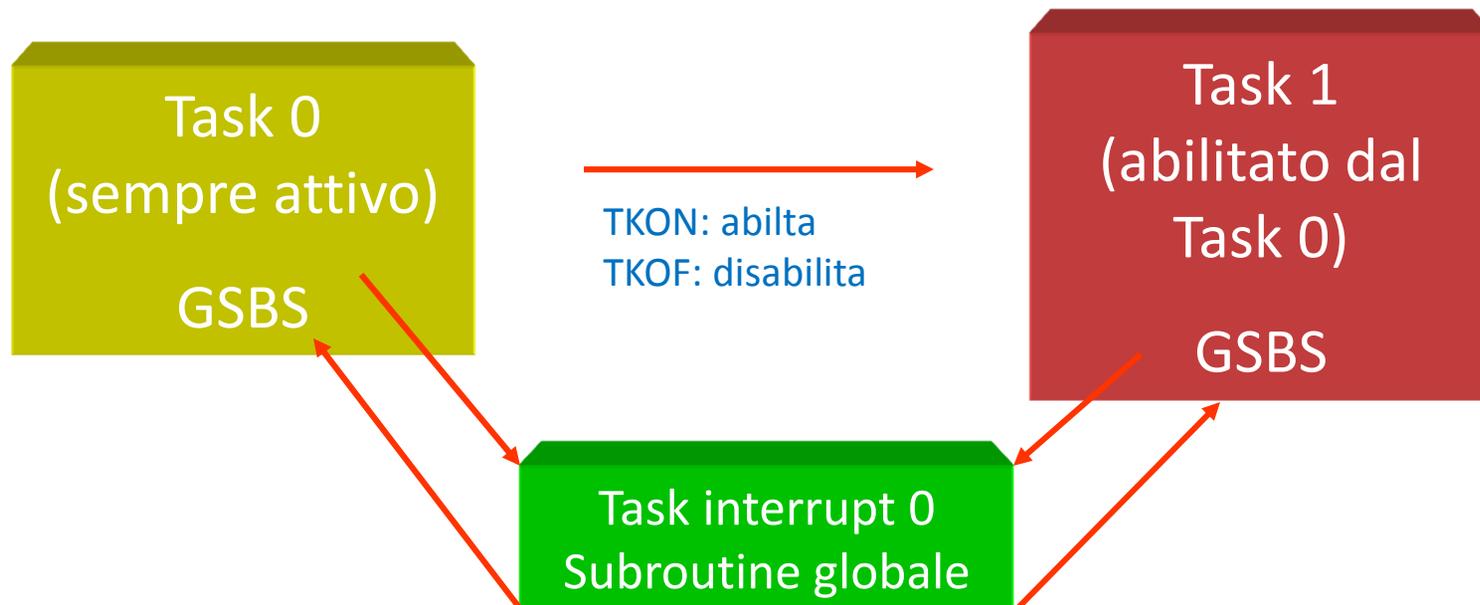
Subroutine Globali: esempio

- All'interno del Task ad interrupt 00 definisco la Subroutine Globale 3. Con l'istruzione G SBS(750) sarà possibile richiamare questa Subroutine da tutti gli altri Task (compresi quelli ad Interrupt).



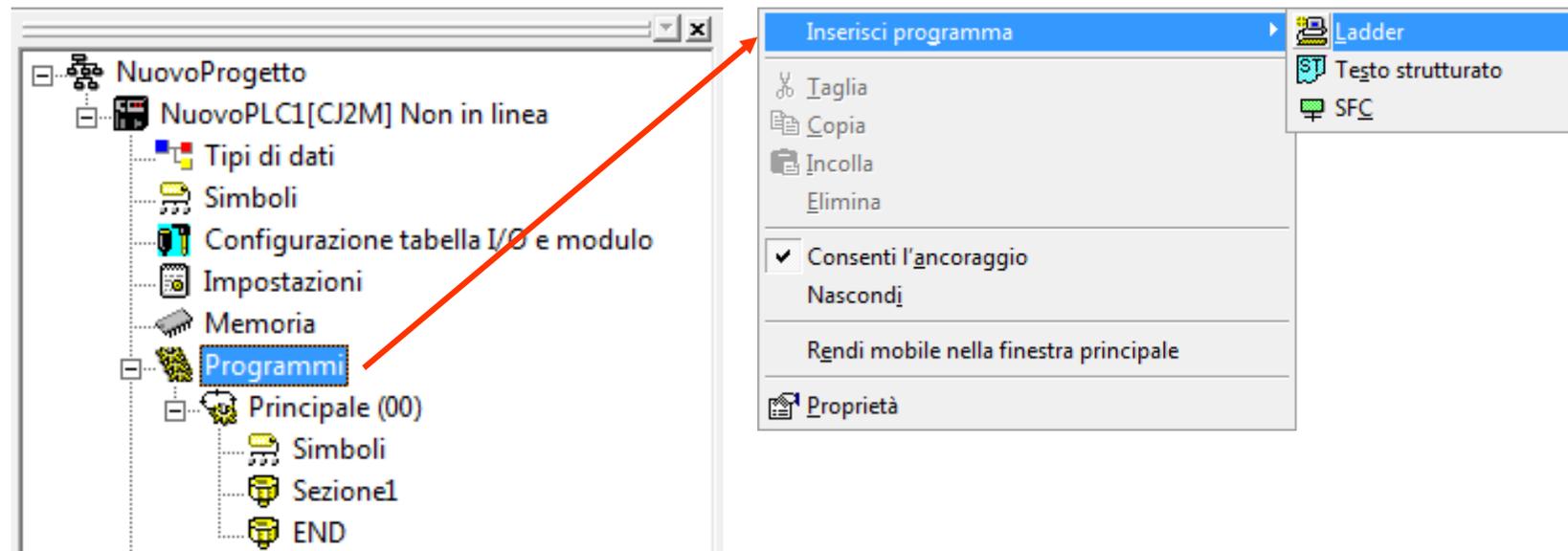
Task e Subroutine: esempio

- In questo esempio due Task Ciclici diversi richiamano la stessa Subroutine Globale.
- Mentre uno dei Task Ciclici è sempre attivo (Task 0), l'altro può essere attivato/disattivato con le istruzioni TKON e TKOF.



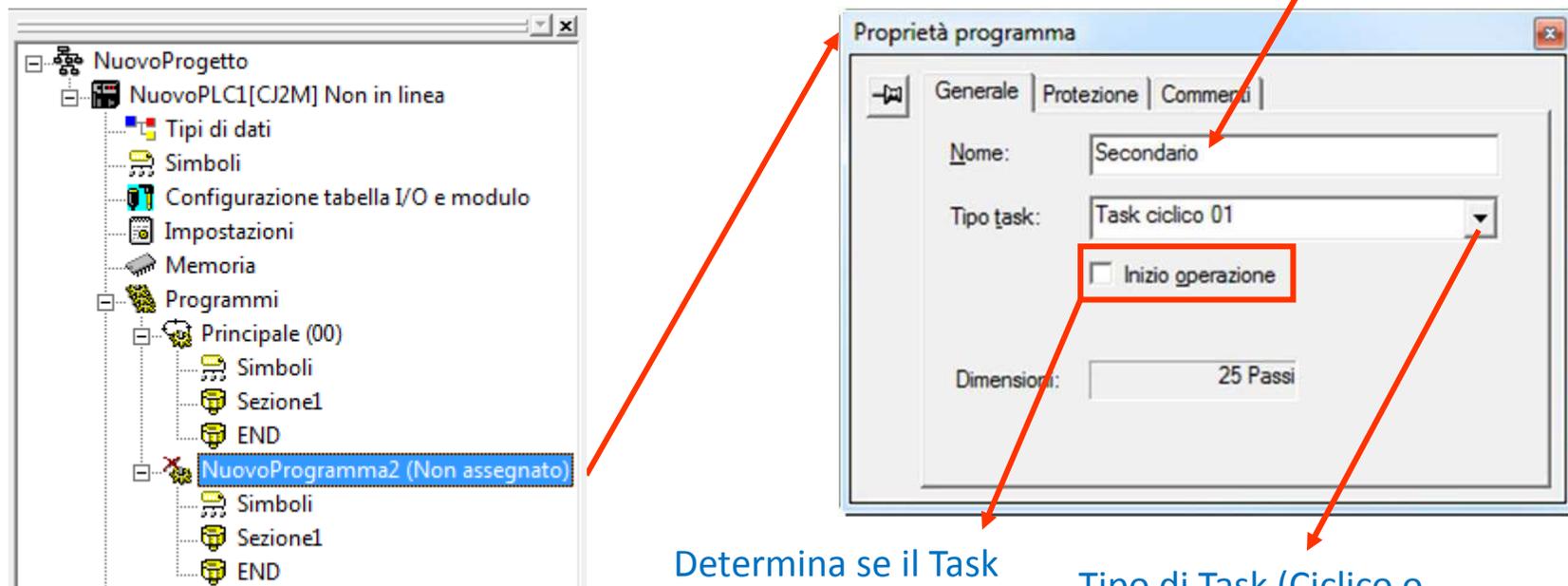
Esempio: creazione Task

- Selezionare l'icona "Programmi" con il tasto destro.
- Dal menu selezionare "Inserisci programma" → "Ladder".



Esempio: creazione Task Ciclico 1

- Per assegnare un Task al nuovo programma:
 - Selezionare l'icona del nuovo programma con il tasto destro e selezionare "Proprietà".



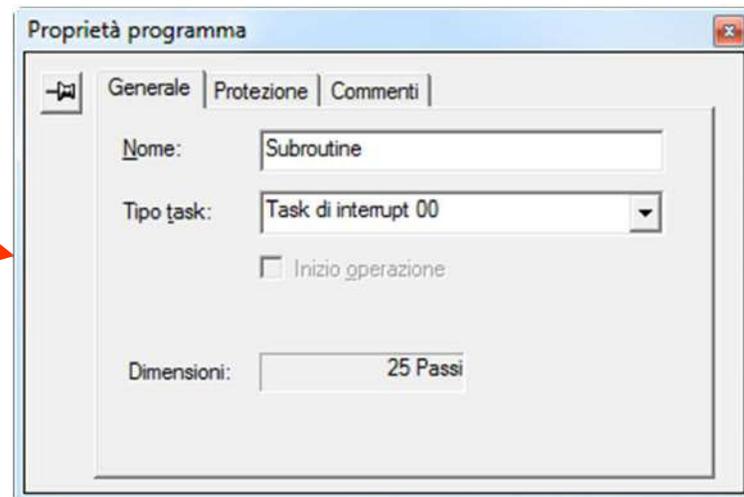
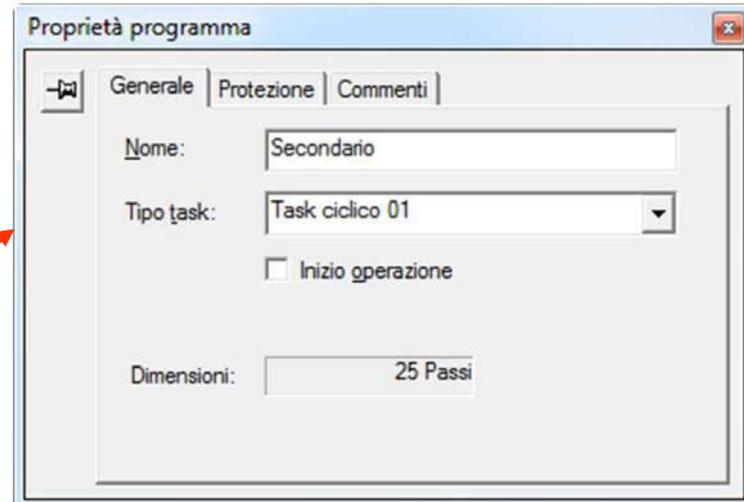
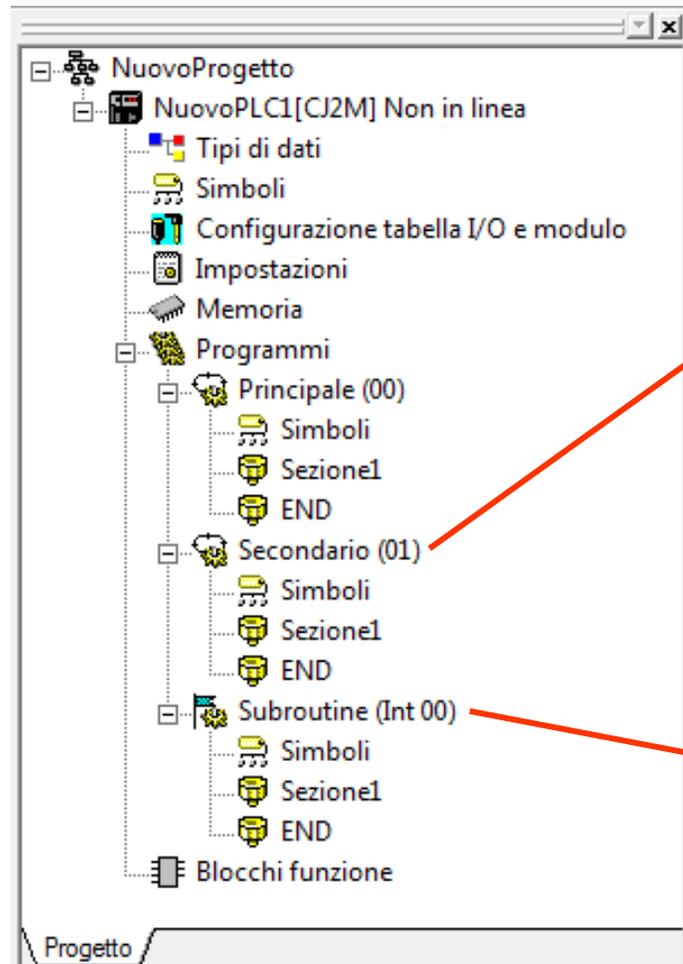
Nome del task.

Determina se il Task deve essere attivo all'accensione.

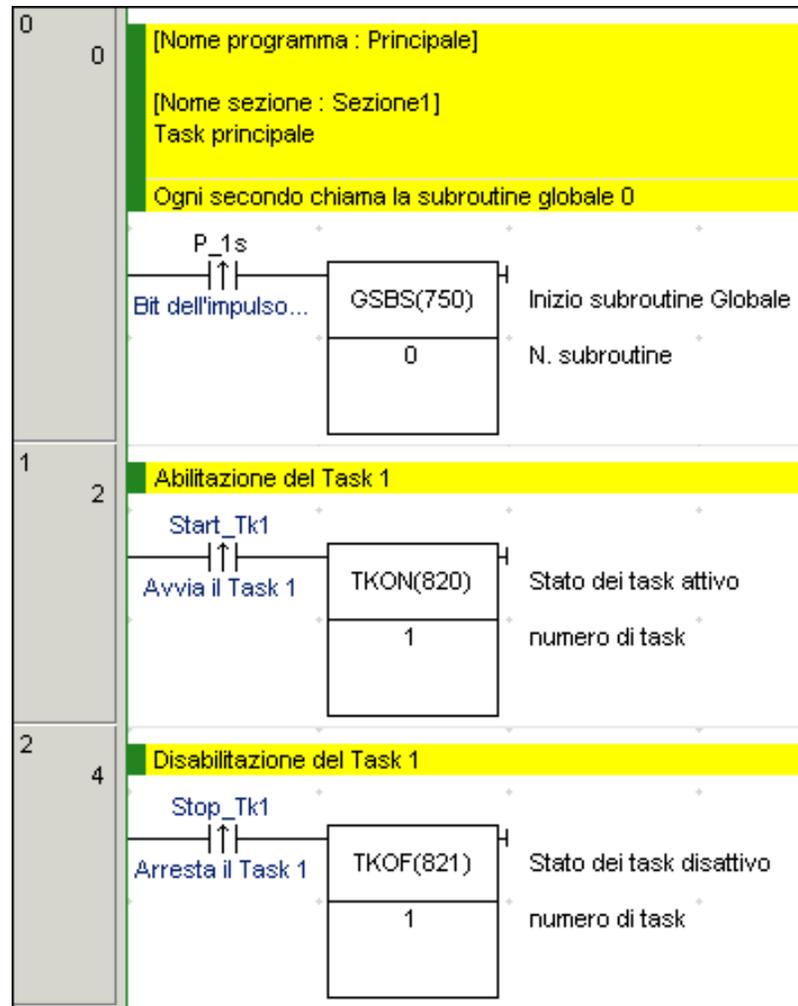
Tipo di Task (Ciclico o Interrupt) e numero del Task.

Esempio: definizione delle proprietà dei Task

Disabilitare la casella di "Inizio operazione".



Esempio: programma Task ciclico 0



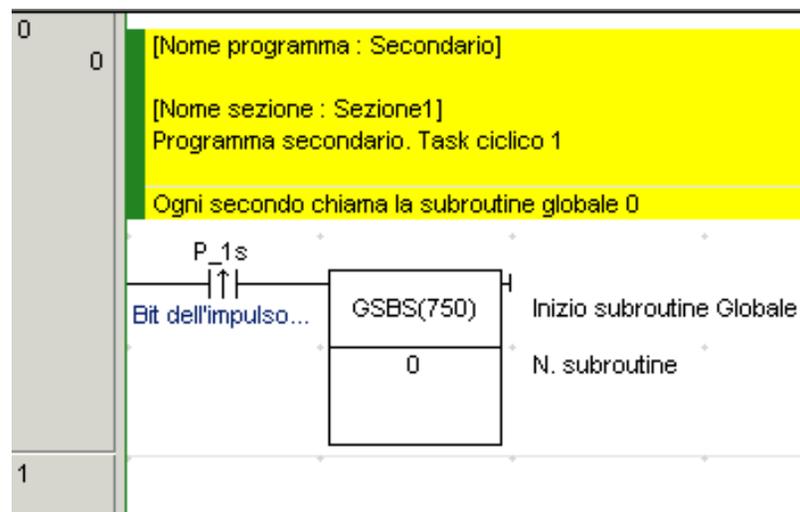
Chiamata della Subroutine Globale 0 sul fronte di salita del flag a 1 sec.

Sul fronte di salita dell'ingresso 0.00 attiva il Task Ciclico 1.

Sul fronte di salita dell'ingresso 0.01 disattiva il Task Ciclico 1.

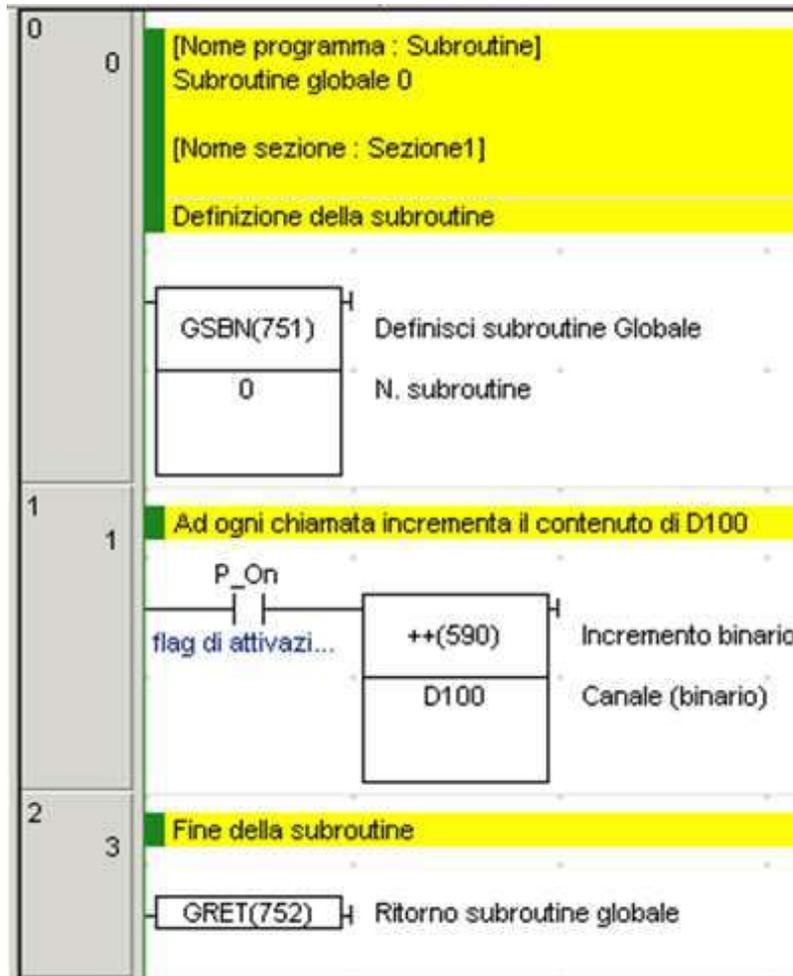
Esempio: programma Task ciclico 1

Programma secondario assegnato al Task Ciclico 1.



Chiamata della Subroutine Globale 0 sul fronte di salita del flag a 1 sec.

Esempio: Subroutine Globale 0



Subroutine Globale 0 definita all'interno del Task a interrupt 0.

Definizione della Subroutine. L'istruzione GSBN delimita l'inizio e assegna il numero.

Il bit 'sempre a On' fa in modo che il valore di D100 venga incrementato ad ogni chiamata della Subroutine.

Determina la fine della Subroutine e il ritorno al programma chiamante.

Indirizzamento indiretto e cicli FOR-NEXT

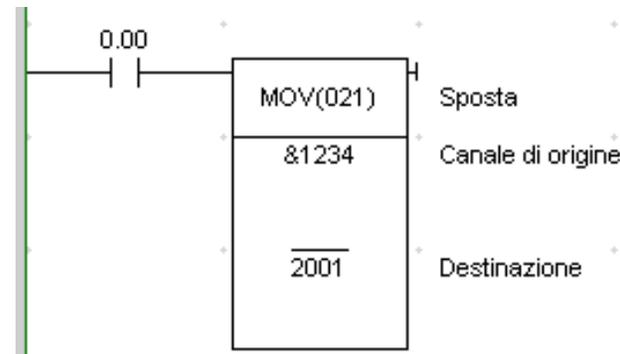
Corso intermedio CPS2

Indirizzamento indiretto

- Parametri di un'istruzione possono contenere:

- Una costante:

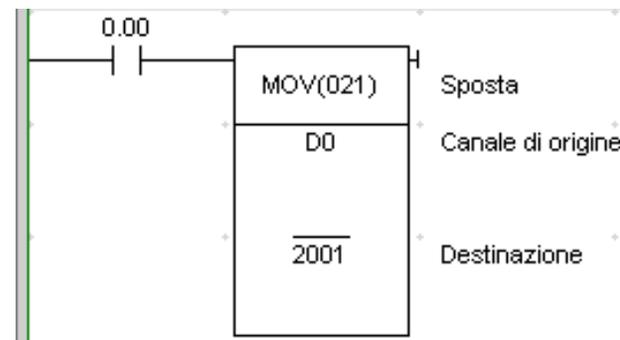
Assegnazione di un valore costante.



Alla chiusura del contatto 0.00 il contenuto della word 2001 sarà uguale alla costante specificata nel primo operando dell'istruzione MOV(021).

- Un indirizzo diretto ad una variabile:

Assegnazione di una variabile.

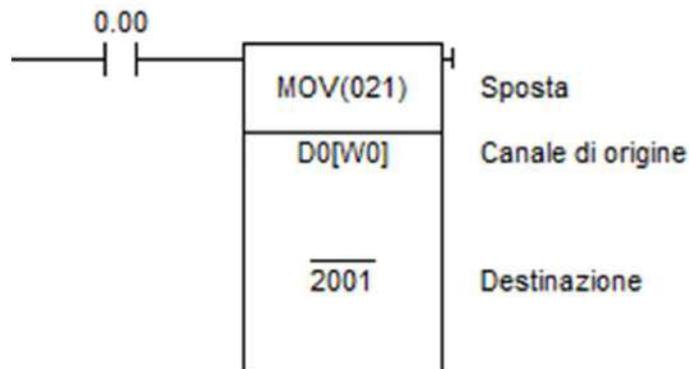


Alla chiusura del contatto 0.00 il contenuto della word 2001 sarà uguale al valore presente nella word D0.

Se D0=1234 allora 2001=1234.

Indirizzamento indiretto

- E' possibile specificare direttamente degli **offset** sugli indirizzi specificando l'indice tra parentesi quadre. L'offset può essere anche una variabile, cioè un'altra word, in modo da poter indicizzare dinamicamente l'accesso alla memoria.

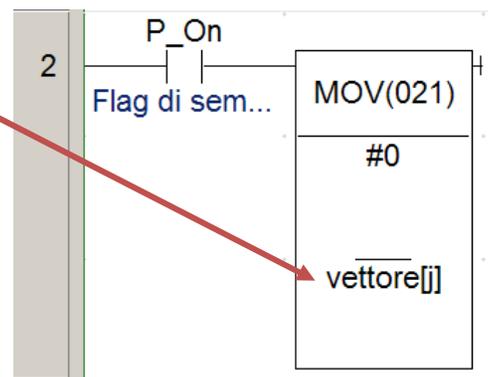


W0 = #0050 (50 Hex = 80 dec.).

D80 = 1234 → 2001 = 1234.

Variabili Array

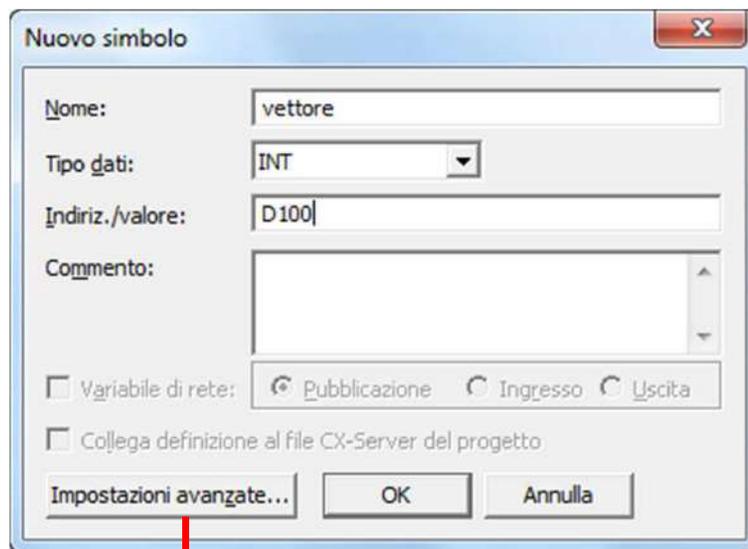
- La possibilità di specificare direttamente gli offset sugli indirizzi permette l'uso di variabili Array con **indice dinamico** (sui PLC CJ1 sono possibili unicamente array con indice statico).
- Esempio:
vettore[j]: il valore della variabile j specifica l'indice dell'array 'vettore'.



vettore	INT[20]	D1000
j	UINT	H10

Definizione di un Array

- Nella fase di creazione di un nuovo simbolo, selezionare l'opzione "Impostazioni avanzate..." e inserire il numero degli elementi.
- Gli Array sono monodimensionali (indice singolo) e la dimensione massima è di 32000 elementi.



Nuovo simbolo

Nome: vettore

Tipo dati: INT

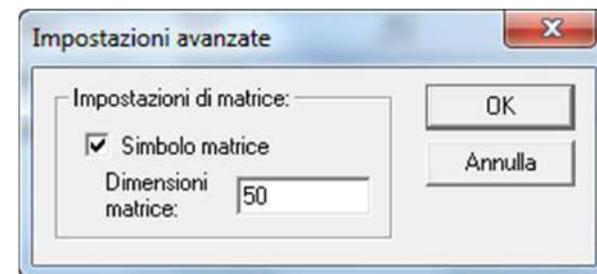
Indiriz./valore: D100

Commento:

Variabile di rete: Pubblicazione Ingresso Uscita

Collega definizione al file CX-Server del progetto

Impostazioni avanzate... OK Annulla



Impostazioni avanzate

Impostazioni di matrice:

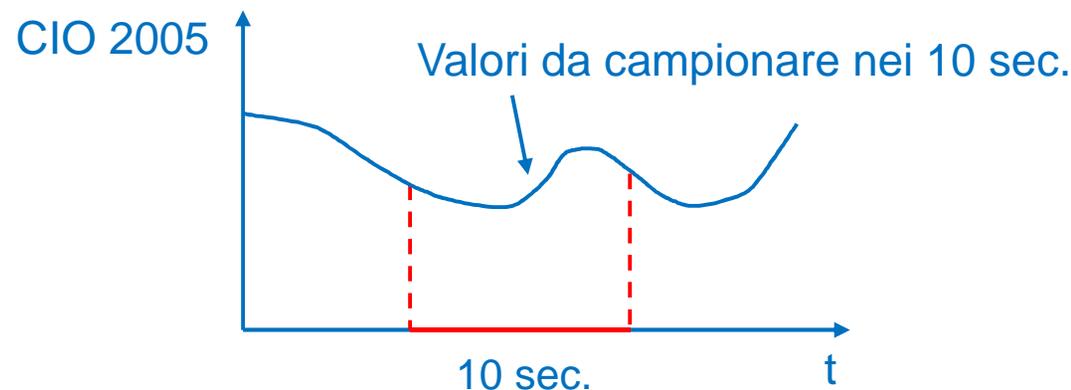
Simbolo matrice

Dimensioni matrice: 50

OK Annulla

Esempio: creazione di una tabella di campioni

- E' necessario memorizzare in una tabella i valori provenienti dal campionamento dell'ingresso analogico 0 (canale 2005).
- Il campionamento deve iniziare quando viene attivato il comando esterno 0.00.
- Si vogliono registrare solo 10 campioni, con periodo di campionamento di 1 secondo, a partire dal canale D100.



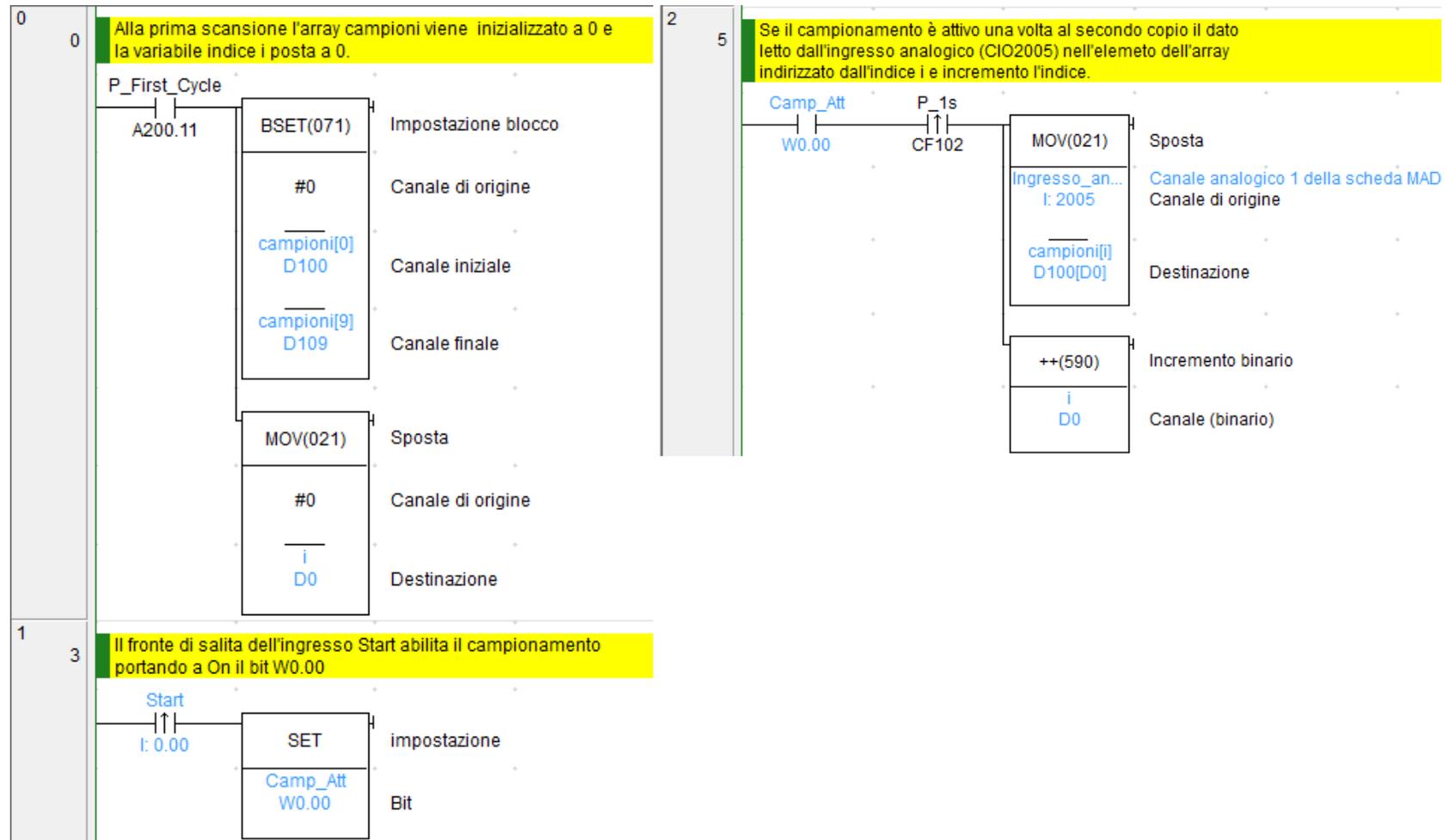
Esempio: creazione di una tabella di campioni

- La tabella memorizzata dopo i dieci campionamenti potrebbe essere ad esempio come quella che segue:

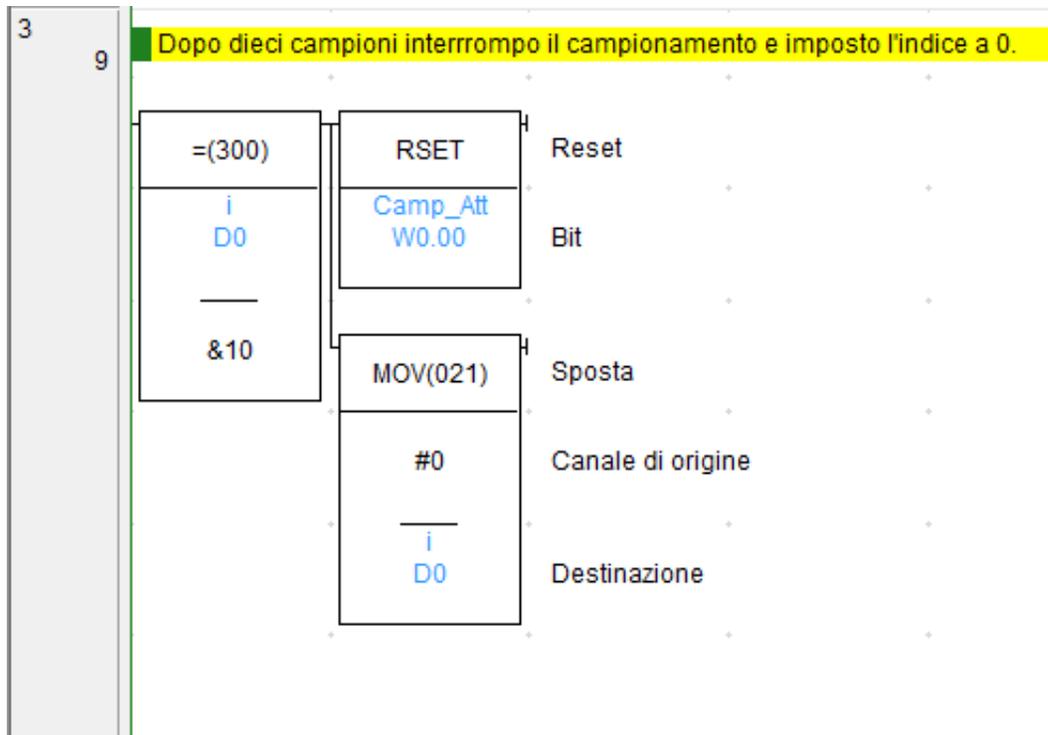
D00100=050F	D00105=05D5
D00101=04C2	D00106=05F2
D00102=0485	D00107=0529
D00103=04B0	D00108=052F
D00104=04B0	D00109=051C

- Sarà necessario definire un Array di dimensione 10 elementi a partire da D100.
- Verrà creato un simbolo (locale o globale) di nome campioni di tipo INT. Nelle impostazioni avanzate verrà dichiarata la dimensione di 10 elementi.

Soluzione: creazione di una tabella di campioni



Soluzione: creazione di una tabella di campioni



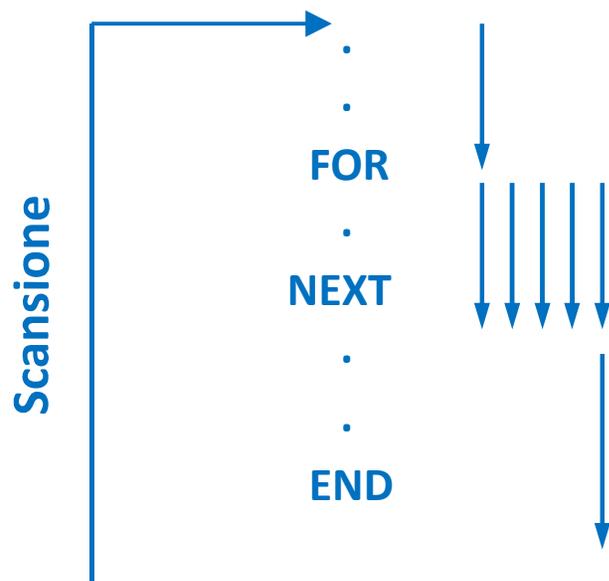


Ciclo FOR-NEXT

Corso intermedio CPS2

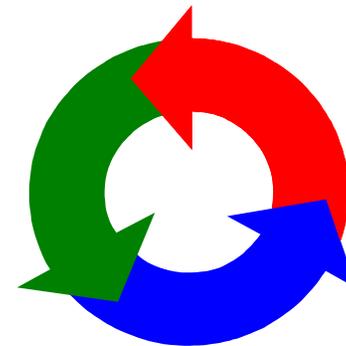
Ciclo FOR-NEXT

- L'istruzione FOR accetta un solo operando che indica il numero di ripetizioni delle istruzioni comprese tra i FOR e NEXT.
- NEXT indica la fine del ciclo.
- BREAK impone, se attivata, l'uscita dal ciclo stesso.



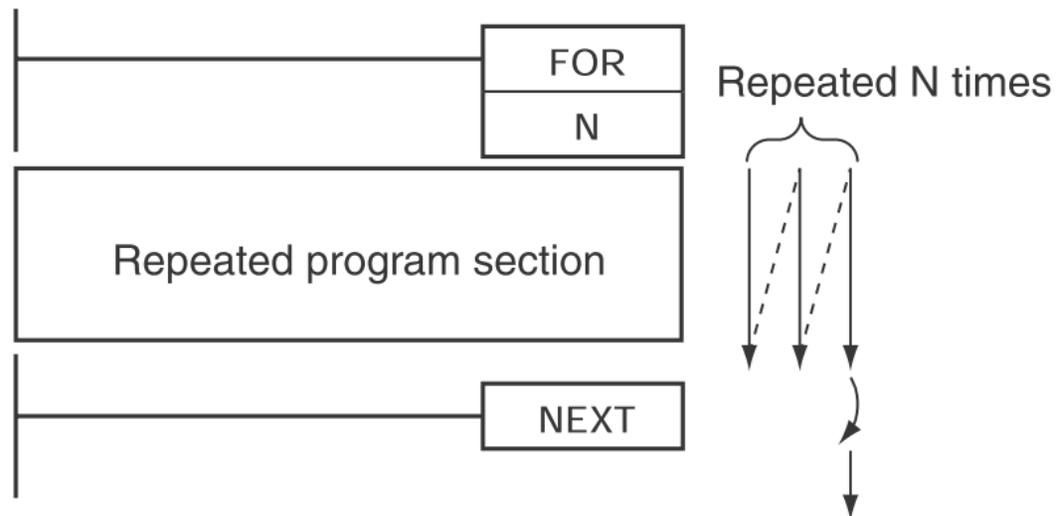
A lato è rappresentato lo schema di esecuzione di un ciclo FOR-NEXT.

N cicli



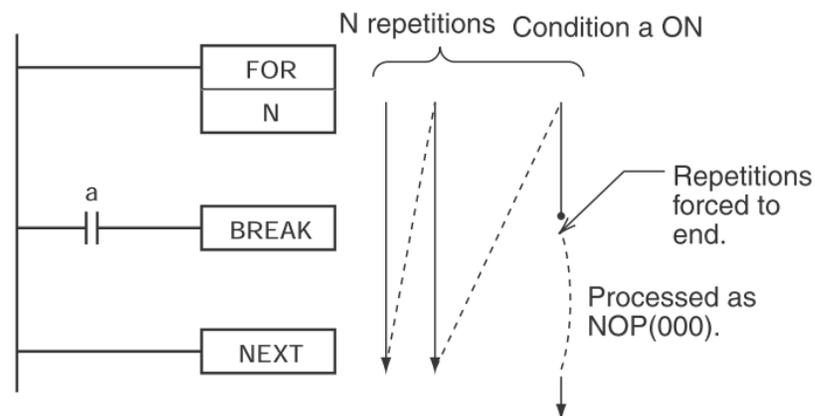
Istruzioni di ciclo FOR e NEXT

- Utili per elaborare tabelle.
- Fino a 15 livelli di annidamento.
- Il ciclo è interrompibile con l'istruzione BREAK.



Istruzione BREAK

- Interrompe il ciclo e le istruzioni rimanenti vengono eseguite come NOP.
- Per uscire da cicli innestati sono necessari più BREAK.

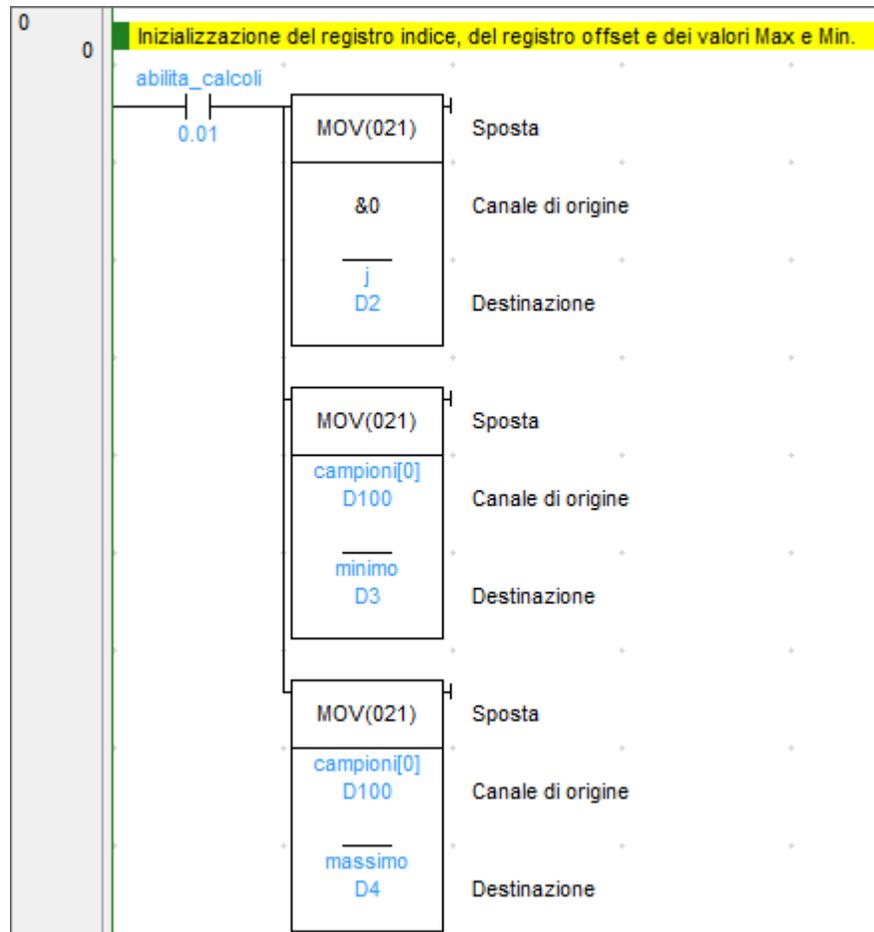


- All'interno di un ciclo FOR-NEXT è conveniente valutare subito la condizione di uscita (contatto che attiva la BREAK) subito dopo l'istruzione FOR, onde evitare inutili perdite di tempo all'interno del ciclo stesso.

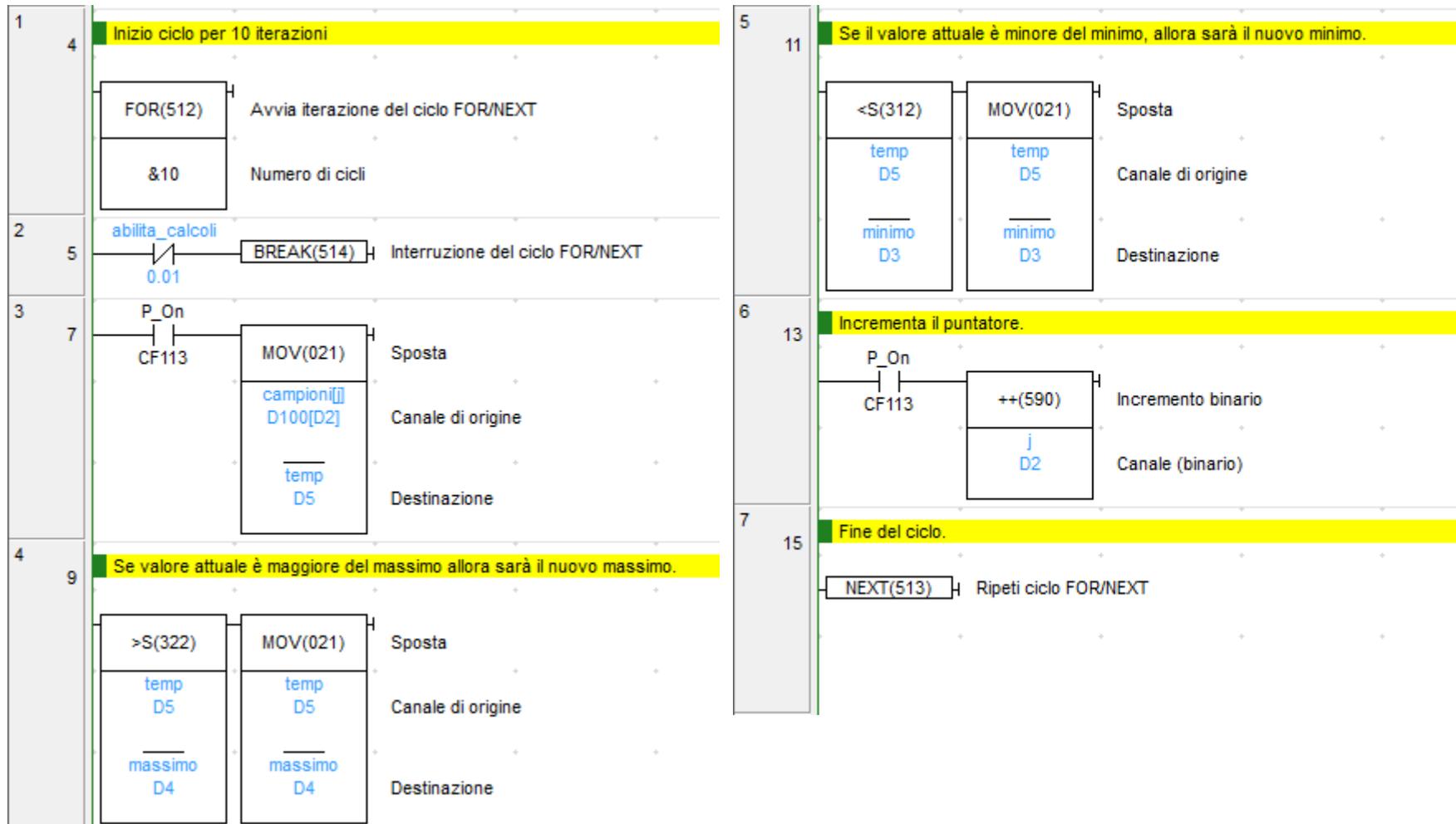
Esempio: ricerca del minimo e del massimo

- In questo esempio viene utilizzato un ciclo FOR-NEXT per la scansione di una tabella per calcolare il valore massimo e il valore minimo.
- Come tabella può essere utilizzata la tabella di campioni ottenuta con l'esercizio precedente.
- Tutte le volte che si chiude il contatto 0.1 viene lanciato il ciclo per ricavare il Min e il Max.
- Per scansionare la tabella verrà fatto variare l'indice 'j' per accedere agli elementi dell'Array **'campioni'**.

Soluzione: ricerca del minimo e del massimo



Soluzione: ricerca del minimo e del massimo



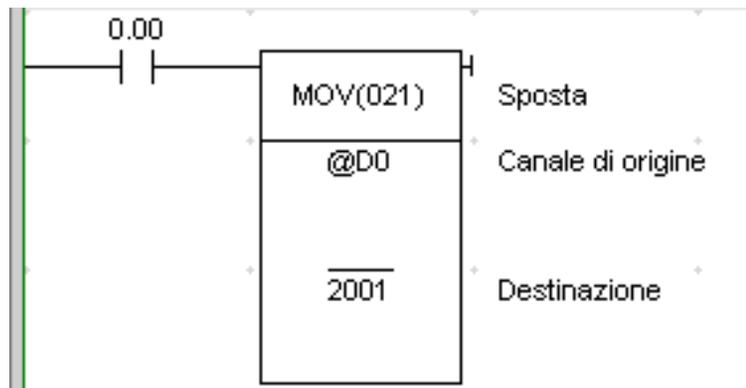


Altri metodi di Indirizzamento indiretto

Corso intermedio CPS2

Altri Indirizzamenti indiretti

- E' possibile specificare un puntatore ad una word che contiene l'indirizzo in cui si trova la variabile (Indirizzamento indiretto):
 - Con il prefisso '@', il contenuto della word verrà interpretato come un indirizzo espresso in esadecimale.

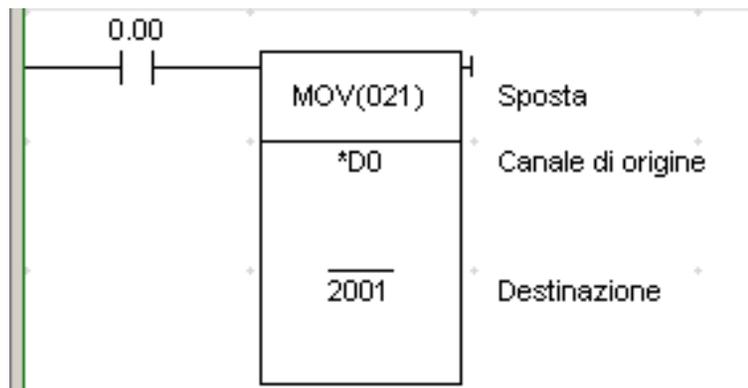


D0 = #0050 (50 Hex = 80 dec.).

D80 = 1234 → 2001 = 1234.

Altri Indirizzamenti indiretti

- Per compatibilità con i PLC delle serie precedenti (ad esempio: C200H e CQM1), è possibile specificare l'indirizzo in BCD.
- Su PLC CS1/CJ1 sarà possibile accedere solo ai DM da D0 a D9999.
 - Con il prefisso '*' il contenuto della word verrà interpretato come un numero in formato BCD che rappresenta l'indirizzo di una word.

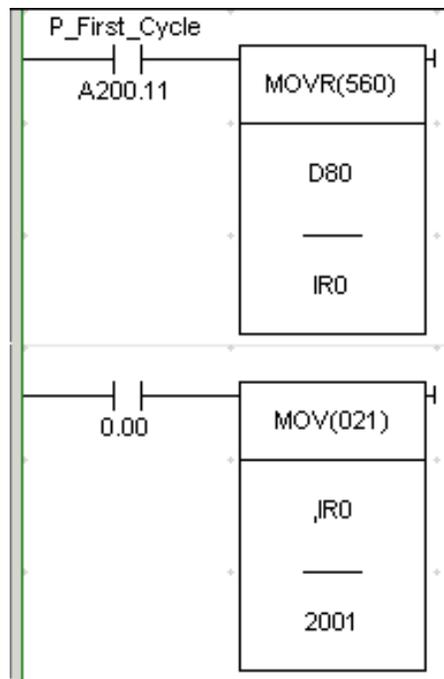


D0 = #0050.

D50 = 1234 → 2001 = 1234.

Altri Indirizzamenti indiretti

- E' inoltre possibile utilizzare un Registro Indice (IR) come puntatore per l'Indirizzamento indiretto.



IR0 è un registro indice, al suo interno viene posto l'indirizzo per avere l'accesso ad un'altra memoria.

,IR0 consente l'accesso alla memoria indirizzata da IR0.

Dopo la MOV(R) IR0 punta a D80 e quindi ,IR0 = D80.

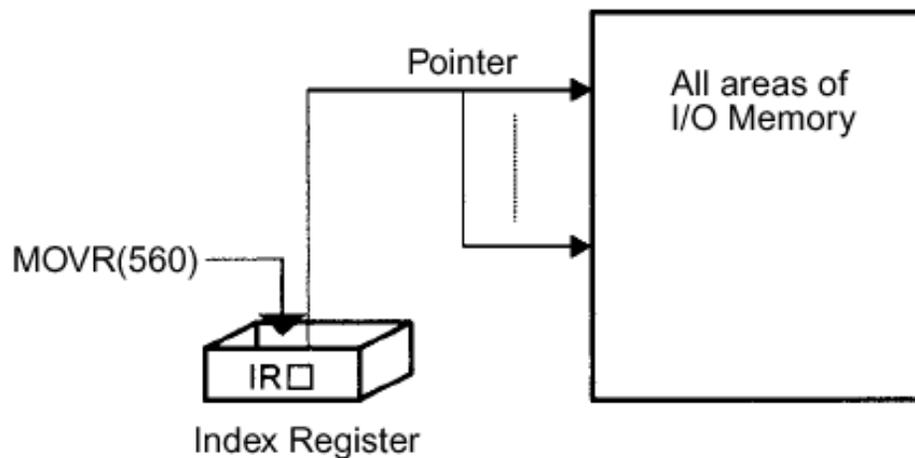
Se D80 = 1234 → 2001 = 1234.

Altri Indirizzamenti indiretti

- L'Indirizzamento indiretto è estremamente potente e semplice da sfruttare grazie ai registri IR e DR e delle funzioni ad essi associate.
- Registri indice IR: puntatore alla memoria del PLC.
- Registri DR: offset relativo alla posizione del puntatore.
- Esistono **16 registri IR** (IR00 a IR15).
- Esistono **16 registri DR** (DR00 a DR15).

Altri Indirizzamenti indiretti

- Il registro IR contiene l'indirizzo fisico della word o del bit all'interno della memoria del PLC.
- Per accedere al valore puntato dal registro IR, si usa la sintassi **'IR'**.
- L'Indirizzamento indiretto può essere utilizzato anche per i bit.

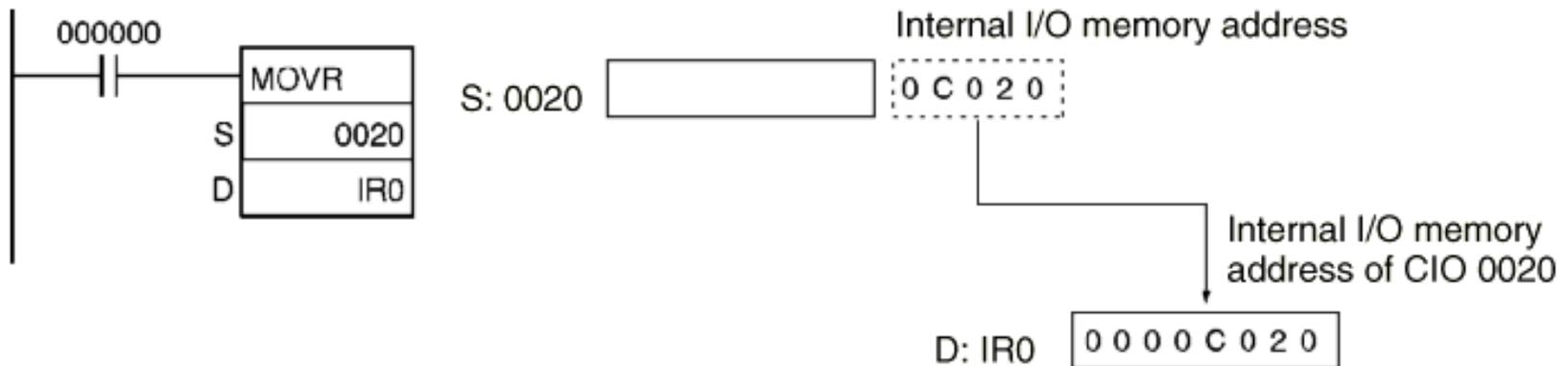


MOVR | D10 | IR0

Carica l'indirizzo fisico di DM10 in IR0.

Inizializzazione dei Registri Indice

- MOVR: imposta un Registro Indice con l'indirizzo di una word, di un bit, di un flag di timer/contatore.
- MOVRW: imposta un Registro Indice con l'indirizzo di un PV di timer/contatore.



Accesso ai dati utilizzando i Registri Indice

- Per accedere al contenuto della word (o del bit) puntato dal registro indice:

,IR00

- Per utilizzare un registro DR come offset rispetto all'indirizzo puntato dal registro IR:

DR0,IR0

- Per impostare un offset fisso rispetto all'indirizzo puntato dal registro IR:

+x,IR00

-x,IR00

(con x compreso tra 0 e 1024).

Accesso ai dati utilizzando i Registri Indice

- Per accedere alla memoria puntata dal Registro Indice e spostarsi in avanti è possibile usare gli operatori:

,IR00+ (incremento di 1 word)

,IR00++ (incremento di 2 word)

L'incremento del registro avviene dopo l'accesso.

- Per accedere alla memoria puntata dal Registro Indice e spostarsi all'indietro è possibile usare gli operatori:

,-IR00 (decremento di 1 word)

,-IR00 (decremento di 2 word)

Il decremento del registro avviene prima dell'accesso.

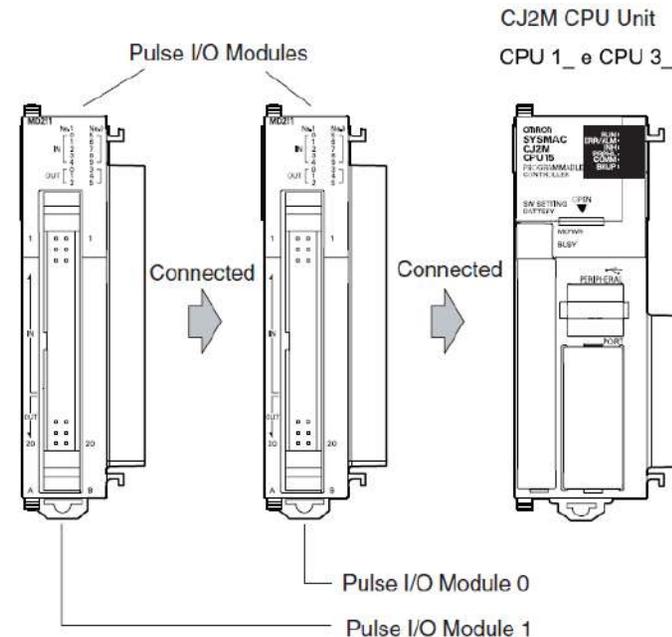


I Moduli CJ2M-MD21_

Corso intermedio CPS2

Moduli di espansione MD211 e MD212

- Alle CPU CJ2M è possibile collegare fino a un max. di **2 moduli MD21_**.
- Ogni modulo MD21_ dispone delle seguenti funzionalità:
 - Due ingressi di conteggio veloce per collegamento di 2 encoder;
 - Due uscite a treno di impulsi con controllo della direzione;
 - Quattro ingressi a Interrupt;
 - Due uscite a modulazione di impulso PWM.



I moduli MD211 e MD212 differiscono unicamente per le uscite.

CJ2M-MD211 → Uscite NPN (pin compatibili con CJ1M-CPU21, 22 e 23).

CJ2M-MD212 → Uscite PNP.

Ingressi ad Interrupt

- Quattro dei dieci ingressi dei moduli CJ2M-MD21_ (da IN 0 a IN 3) possono essere configurati per gestire segnali ad Interrupt.
- E' possibile impostare tre differenti modalità operative:
 - **Modo diretto:** un Task ad Interrupt viene eseguito su ogni fronte di salita (o di discesa) dell'ingresso.
 - **Modo contatore:** un Task ad Interrupt viene eseguito quando il conteggio del segnale sull'ingresso raggiunge un valore preimpostato.
 - **Risposta rapida:** il PLC è in grado di rilevare, bufferizzandoli, i segnali di durata minima di 30 μ sec, indipendentemente dal tempo di scansione attuale.

Ingressi a conteggio veloce

- I moduli CJ2M-MD21_ supportano due ingressi a conteggio veloce (50-100 kHz) per la gestione dei segnali provenienti da sensori e da encoder reali o simulati.
- Caratteristiche principali:
 - Modalità di conteggio lineare o circolare;
 - Reset solo software o software più fase Z;
 - Impostazione ingressi a fasi differenziali, impulso più direzione, bidirezionali o modo incrementale;
 - Generazione dell'Interrupt di conteggio a Target o a Range.

Uscite impulsive

- I moduli CJ2M-MD21_ possono generare due uscite a treno di impulsi per il controllo di posizione.
- Caratteristiche principali:
 - Modalità CW/CCW o impulso più direzione;
 - Selezione automatica della direzione per il posizionamento in coordinate assolute;
 - Controllo di tipo triangolare;
 - Modifica del target durante il posizionamento;
 - Possibilità di utilizzare due uscite con Duty Cycle variabile (PWM).



Gestione degli Interrupt

Corso intermedio CPS2

Segnali di Interrupt

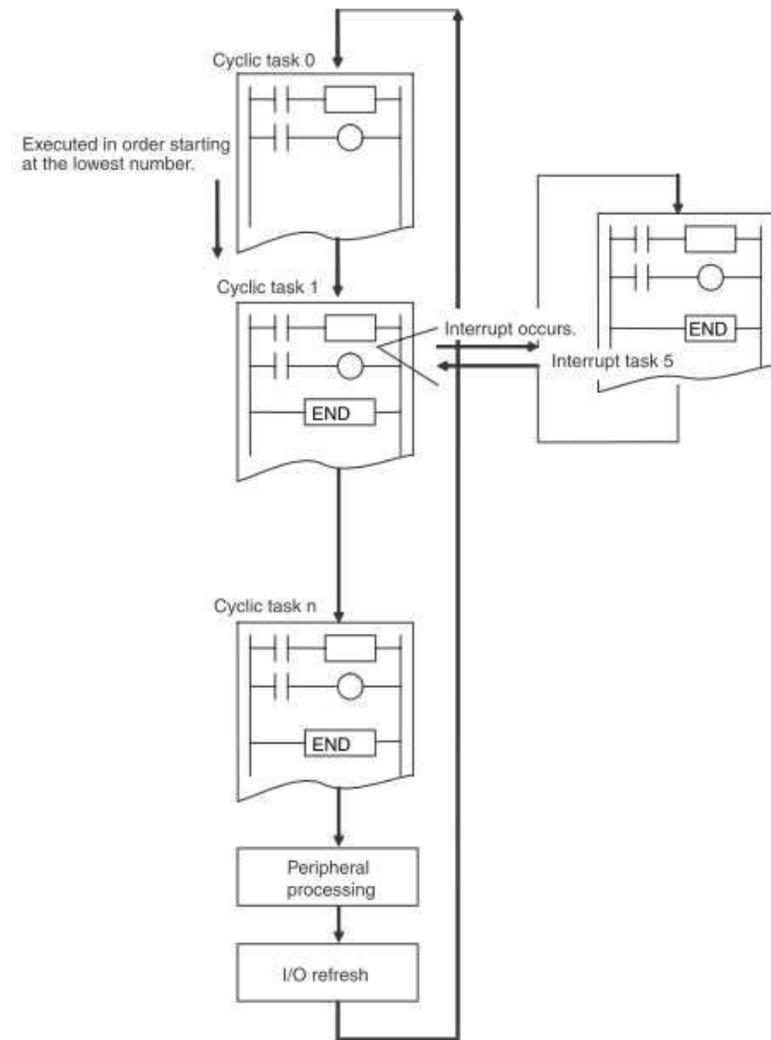
- Di norma la gamma dei segnali rilevabili dal PLC è funzione del tempo di scansione.
- E' possibile definire determinati ingressi come 'ingressi ad Interrupt' gestiti dalla CPU in modo indipendente dalla durata del tempo di scansione.
- I segnali di Interrupt possono essere anche generati direttamente dalla CPU, dai moduli Speciali e dai moduli Bus CPU.
- Al presentarsi di un Interrupt, l'esecuzione del Task Ciclico corrente viene temporaneamente interrotta, e viene eseguito il Task ad Interrupt relativo all'Interrupt stesso.

I Task ad Interrupt

- Esistono quattro tipologie di Task ad Interrupt:
 - **1 Task di spegnimento:** eseguito quando si spegne la CPU (Task ad Interrupt 01). Questo Task deve essere attivato nella finestra di impostazioni del PLC.
 - **2 Task schedulati:** eseguiti ad intervalli di tempo prefissati (Task ad Interrupt 02 e 03). L'intervallo di tempo può essere impostato nel setup del PLC in unità di decimi di millisecondo, millisecondi o decine di millisecondi.
 - **32 Task di I/O:** eseguiti quando va ad ON un ingresso di un modulo di ingressi ad Interrupt (Task ad Interrupt da 100 a 131).
 - **221 Task esterni:** eseguiti quando richiesto da un modulo di I/O speciale o da un modulo Bus CPU.
 - **4/8 Task hardware:** associati a 4/8 ingressi, quando collegato/i modulo/i CJ2M-MD21_ alla CPU (Task ad Interrupt da 140 a 147).

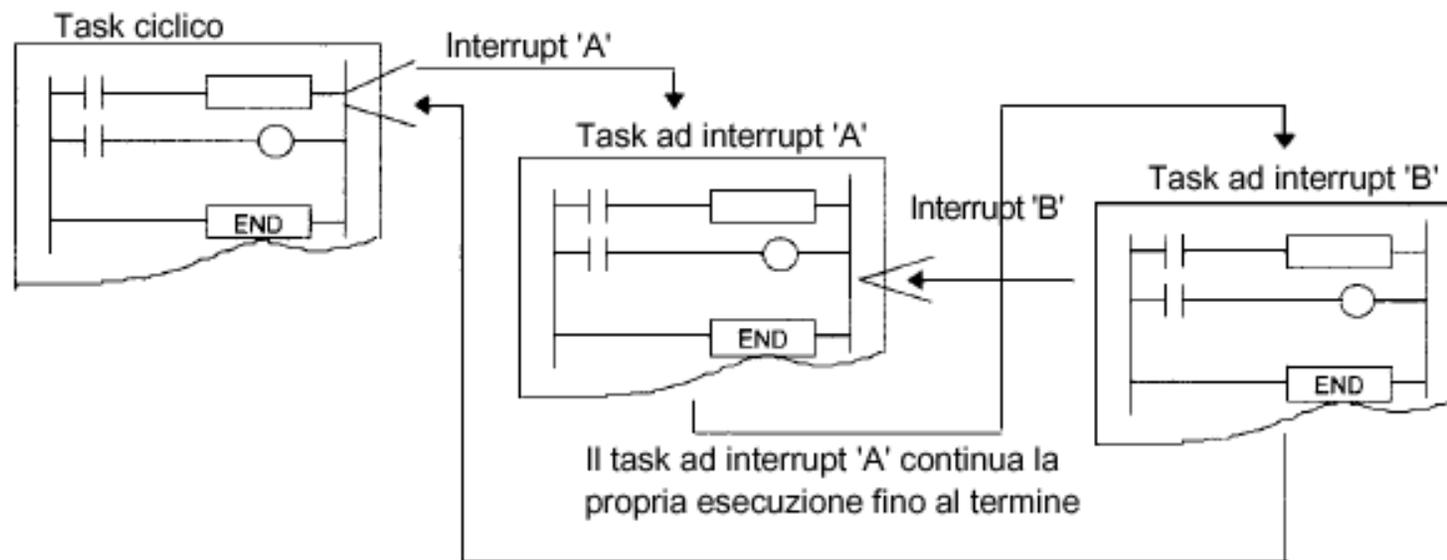
I Task ad Interrupt

- Esempio:
 - L'Interrupt associato al Task ad Interrupt 5 viene rilevato durante l'esecuzione del Task ciclico 1.
 - L'esecuzione del Task ciclico 1 viene sospesa e la CPU esegue il Task ad Interrupt 5.
 - Terminata l'esecuzione del Task ad Interrupt 5 l'esecuzione del Task ciclico 1 riprende esattamente da dove era stata interrotta.



I Task ad Interrupt

- Se si presenta un Interrupt mentre è in esecuzione un Task ad Interrupt, il Task relativo al nuovo Interrupt verrà eseguito solo al termine dell'esecuzione del Task ad Interrupt in corso.



I Task ad Interrupt

- I Task ad Interrupt hanno diversa priorità:

Interrupt da I/O > Interrupt esterni > Interrupt schedulati.

- In caso di contemporaneità verrà eseguito per primo l'Interrupt a priorità più alta; tutti gli altri verranno accodati.
- In ogni caso il Task ad Interrupt di spegnimento interrompe l'esecuzione di tutti gli altri Task ad Interrupt

Gli Interrupt del CJ2M

- Le CPU CJ2M dispongono di:
 - 8 Interrupt hardware originati dalle schede CJ2M-MD21_;
 - 2 Interrupt schedulati;
 - 4 Interrupt legati ai contatori veloci delle schede CJ2M-MD21_.
- Sono inoltre gestibili:
 - Interrupt provenienti da moduli di ingresso ad Interrupt CJ1W-INT01(*);
 - Interrupt provenienti da schede di conteggio veloce CJ1W-CT021(*);
 - Interrupt provenienti da moduli Bus CPU.

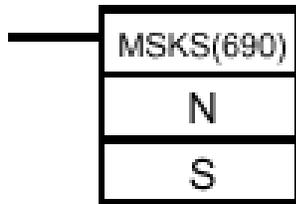
(*) N.B.: gli Interrupt vengono gestiti solo sui primi 5 slot a fianco della CPU.

Gli Interrupt hardware del modulo CJ2M-MD21_

- I primi quattro ingressi di ciascun modulo CJ2M-MD21_ possono essere definiti come ingressi ad Interrupt, dalla maschera 'Impostazioni' del PLC → 'Modulo di I/O', nel CX-Programmer.
- Una volta definiti 'Ingresso Interrupt' otto ingressi sono associati a otto Task ad Interrupt (dal n° 140 al n° 147).
- Gli Interrupt hardware possono lavorare sul fronte di salita o sul fronte di discesa in modalità diretta o in modalità contatore a seconda dei parametri specificati nell'istruzione MSKS(690).

Gli Interrupt hardware del modulo CJ2M-MD21_

- MSKS(690): parametri per la rilevazione degli Interrupt hardware sul fronte di salita o sul fronte di discesa.



Operando	Significato
N	Numero dell'Interrupt: 110: input Interrupt 0 (Block 0) (Interrupt Task 140). 111: input Interrupt 1 (Block 0) (Interrupt Task 141). 112: input Interrupt 2 (Block 0) (Interrupt Task 142). 113: input Interrupt 3 (Block 0) (Interrupt Task 143). 114: input Interrupt 4 (Block 1) (Interrupt Task 144). 115: input Interrupt 5 (Block 1) (Interrupt Task 145). 116: input Interrupt 6 (Block 1) (Interrupt Task 146). 117: input Interrupt 7 (Block 1) (Interrupt Task 147).
S	0000 hex: Interrupt attivo sul fronte di salita. 0001 hex: Interrupt attivo sul fronta di discesa.

Nota: per il primo modulo MD21_ è possibile utilizzare gli stessi operandi del CJ1M (N: 10, 11, 12 e 13 per gli Interrupt 0, 1, 2 e 3).

Gli Interrupt hardware del modulo CJ2M-MD21_

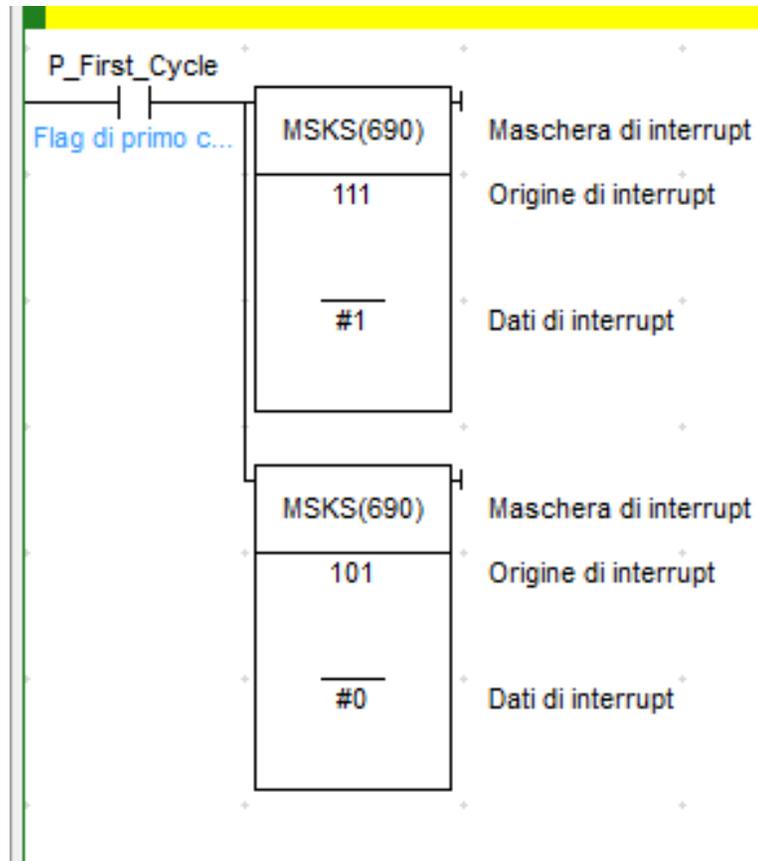
- MSKS(690): parametri per la mascheratura/smascheratura degli Interrupt hardware e per l'impostazione della modalità.



Operando	Significato
N	Numero dell'Interrupt: 100: input Interrupt 0 (Block 0) (Interrupt Task 140). 101: input Interrupt 1 (Block 0) (Interrupt Task 141). 102: input Interrupt 2 (Block 0) (Interrupt Task 142). 103: input Interrupt 3 (Block 0) (Interrupt Task 143). 104: input Interrupt 4 (Block 1) (Interrupt Task 144). 105: input Interrupt 5 (Block 1) (Interrupt Task 145). 106: input Interrupt 6 (Block 1) (Interrupt Task 146). 107: input Interrupt 7 (Block 1) (Interrupt Task 147).
S	0000 hex: abilita (unmask) l'Interrupt (direct mode). 0001 hex: disabilita (mask) l'Interrupt (direct mode). 0002 hex: avvia il contatore a decr. e abilita l'Interrupt (counter mode). 0003 hex: avvia il contatore a incr. e abilita l'Interrupt (counter mode).

Nota: per il primo modulo MD21_ è possibile utilizzare gli stessi operandi del CJ1M (N: 6, 7, 8 e 9 per gli Interrupt 0, 1, 2 e 3).

CJ2M: esempio Interrupt hardware



La rilevazione dell'ingresso ad Interrupt 1 avverrà sul fronte di discesa del segnale.

N = 111: scelta del fronte sull'ingresso a Interrupt 1.

S = #1: attivo sul fronte di discesa.

L'ingresso ad Interrupt 1 viene abilitato (smascherato) e gestito in modalità diretta.

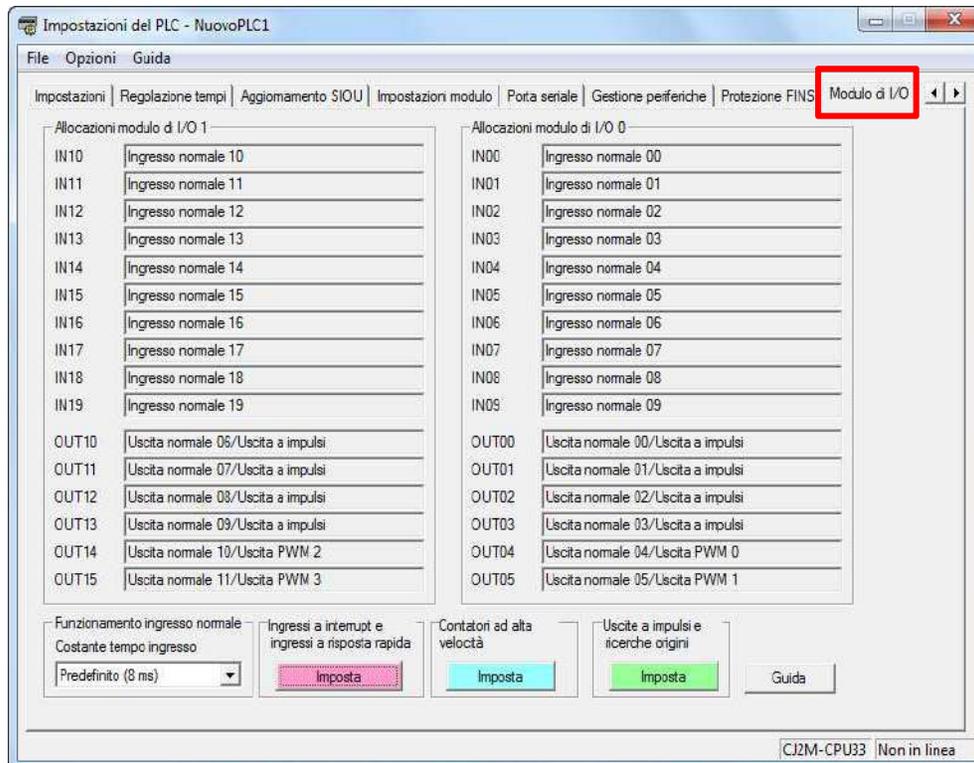
N = 101: abilitazione dell'ingresso a Interrupt 01.

S = #0: Interrupt attivo.

Esempio: configurazione dell'Interrupt 0

- Sequenza delle operazioni da eseguire:
 1. Configurare l'ingresso per la modalità Interrupt.
 2. Inserire nel programma principale la sezione di configurazione dell'Interrupt con le istruzioni MSKS(690).
 3. Aggiungere un nuovo programma e assegnarlo al Task a Interrupt che corrisponde all'ingresso (ad esempio 140).
 4. Scrivere nel Task 140, del codice ladder per verificare il corretto funzionamento dell'Interrupt. Per esempio l'incremento di un Data Memory.

Esempio: menu di impostazione degli Interrupt

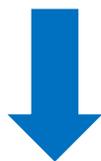


- Per la configurazione si utilizza il tab 'Modulo di I/O' nell'editor Impostazioni.
- La schermata mostra lo stato complessivo degli ingressi e delle uscite dei due moduli MD21_. Il Modulo I/O 0 è quello vicino alla CPU.
- Per configurare gli Interrupt selezionare il pulsante 'Imposta' del gruppo 'Ingresso a Interrupt e ingresso a risposta rapida'.

Esempio: configurazione ingressi a Interrupt

Ingressi a interrupt e ingressi a risposta rapida

Imposta



IN00

Ingresso normale

Ingresso normale

Ingresso a risposta rapida

Ingresso interrupt



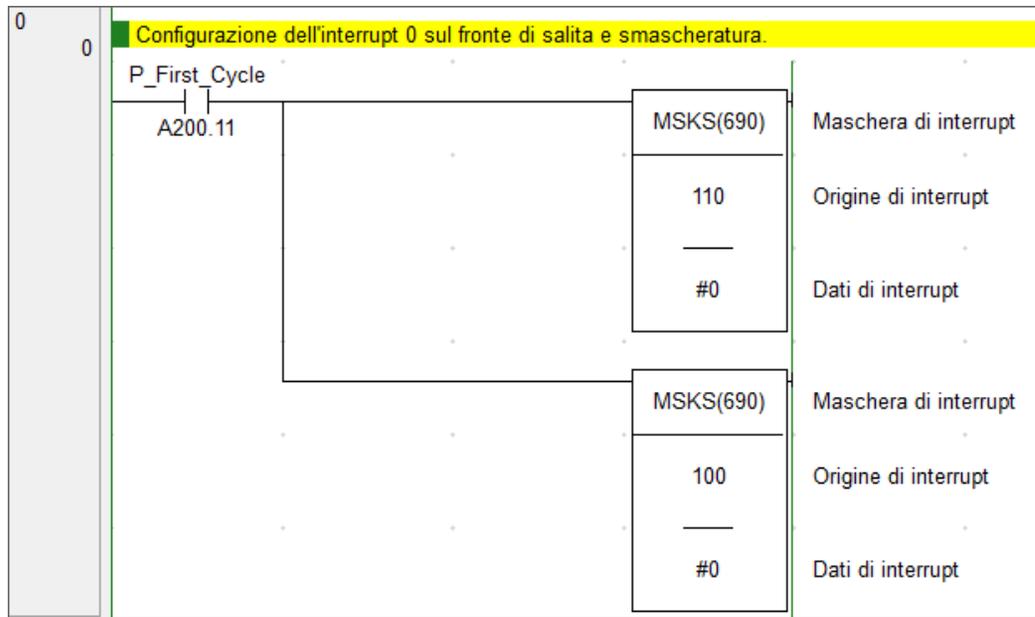
Impostazioni dettagliate ingressi a interrupt e ingressi a risposta rapida

Elemento	IN00	IN01	IN02	IN03	IN10	
Funzioname	Ingresso interrupt	Ingresso normale	Ingresso normale	Ingresso normale	Ingresso no	In
Contorno	In salita	In salita	In salita	In salita	In salita	In
Linguetta di	Non utilizzare	Non utilizzare	Non utilizzare	Non utilizzare	Non utilizza	No

Definizione OK Annulla

- Per ogni ingresso a Interrupt è possibile selezionare la modalità tra:
 - Ingresso normale, cioè nessun Interrupt;
 - Ingresso a risposta rapida;
 - Ingresso Interrupt.

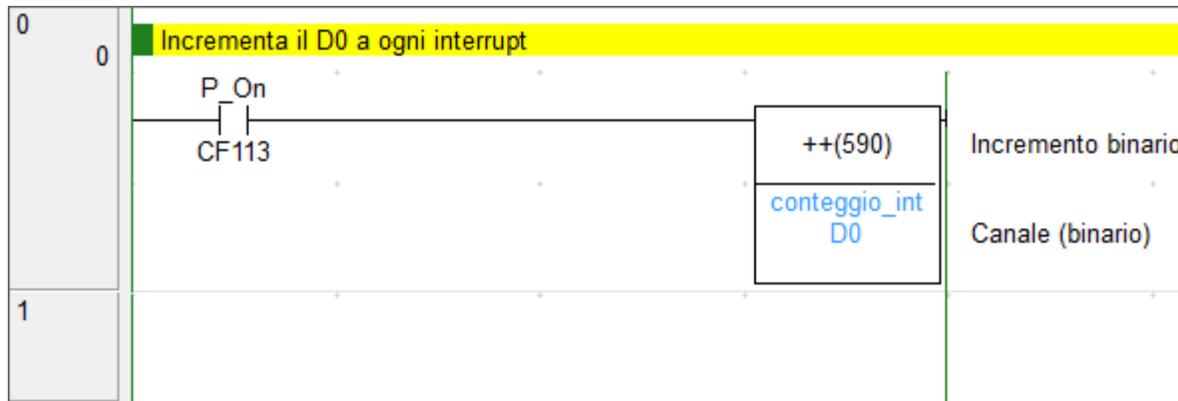
Esempio: Task ciclico principale



Selezione del fronte di salita per l'Interrupt 0.

Smaschera l'Interrupt 0.

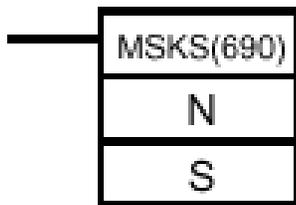
Esempio: Task a Interrupt 140



Il programma contenuto nel Task 140 viene eseguito solo al verificarsi dell'Interrupt.

Gli Interrupt Schedulati

- Due Interrupt schedulati, programmabili in unità di 0,1 ms(*), 1 ms, o 10 ms, permettono di eseguire Task ad Interrupt ad intervalli prefissati.
- Anche questi Interrupt sono gestiti dall'istruzione MSKS(690):

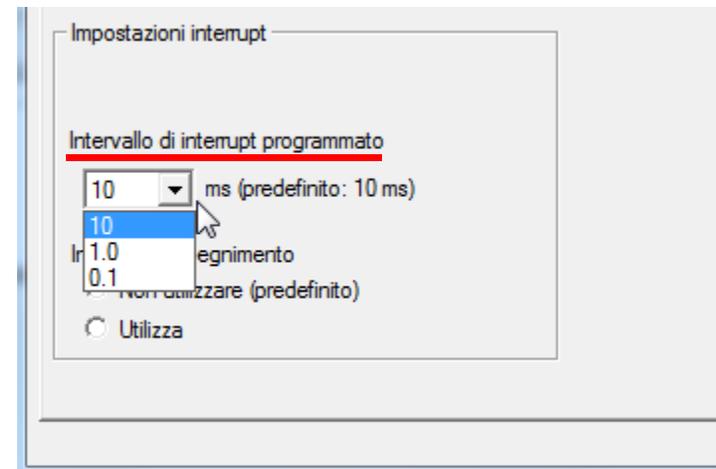
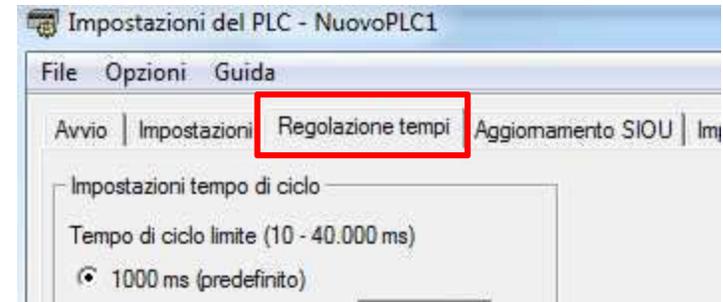


Operando	Significato
N	Specifica il numero di Interrupt schedulato. 14: Interrupt schedulato 0 (Task ad Interrupt 2). 15: Interrupt schedulato 1 (Task ad Interrupt 3).
S	0000: disabilita l'Interrupt schedulato. Da 0001 a 270F Hex (da 1 a 9999 dec.): periodo dell'Interrupt schedulato. Da 0.01 s a 99.99 s con base impostata a 10 ms. Da 1 ms a 9.999s con base impostata a 1 ms. Da 0.4 ms a 999.9 ms con base impostata a 0.1 ms.

(*) Nota: per le CPU CJ2M, il periodo minimo è 0,4 ms.

Configurazione degli Interrupt schedulati

- La base dei tempi degli Interrupt schedulati viene configurata nell'editor delle impostazioni della CPU.
- Fare doppio click sull'icona 'Impostazioni' e selezionare il tab 'Regolazione tempi'.
- Per rendere effettive le nuove impostazioni è necessario spegnere e riaccendere il PLC.



Esercizio: Interrupt schedulato a 100 ms

- Esecuzione del Task a Interrupt 2 (primo Interrupt schedulato) con cadenza periodica di 100 ms.
- Sequenza delle operazioni da eseguire:
 1. Configurare la base dei tempi per gli Interrupt a 1 ms.
 2. Inserire nel programma principale la sezione di configurazione dell'Interrupt con l'istruzione MSKS(690).
 3. Aggiungere un nuovo programma e assegnarlo al Task a Interrupt 2.
 4. Scrivere nel Task 2, delle righe di codice ladder per verificare il corretto funzionamento dell'Interrupt; ad esempio l'incremento di un Data Memory.

Contatori veloci

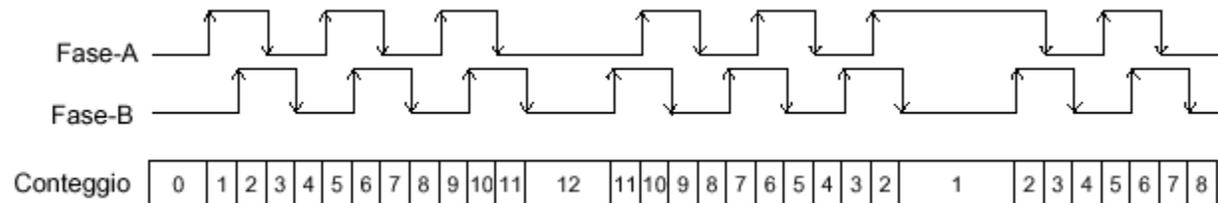
Corso intermedio CPS2

CJ2M-MD21_ : contatore veloce

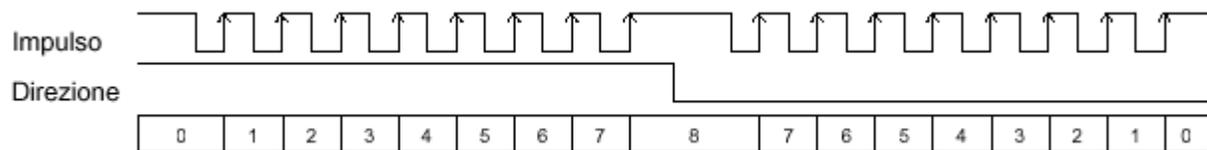
- Ogni modulo MD21_ dispone di 2 ingressi di conteggio veloce configurabili individualmente.
- Entrambi i contatori possono lavorare su ingresso:
 - **Differenziale** (30 kHz 24V - 50 kHz Line Driver);
 - **Impulso più direzione** (60 kHz 24V - 100 kHz Line Driver);
 - **Bidirezionale (Up/Down)** (60 kHz 24V - 100 kHz Line Driver);
 - **Incrementale** (60 kHz 24V - 100 kHz Line Driver).
- Vengono gestite le modalità di conteggio **lineare** e **circolare**.
- Il valore di conteggio è direttamente disponibile in una coppia di word (formato DINT):
 - Contatore 0: A270+A271;
 - Contatore 1: A272+A273;
 - Contatore 2: A316+A317;
 - Contatore 3: A318+A319.

CJ2M-MD21_ : contatore veloce

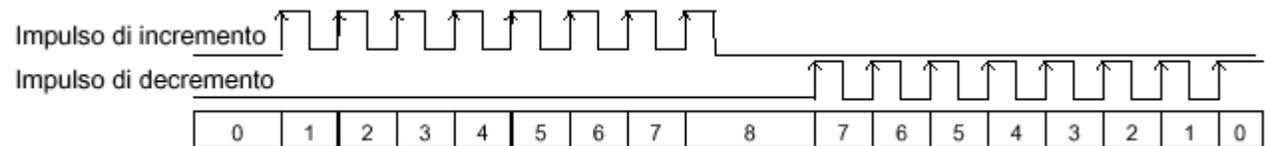
Fasi differenziali:



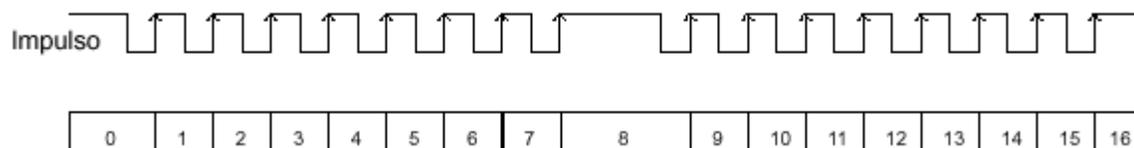
Impulso più direzione:



Bidirezionale:

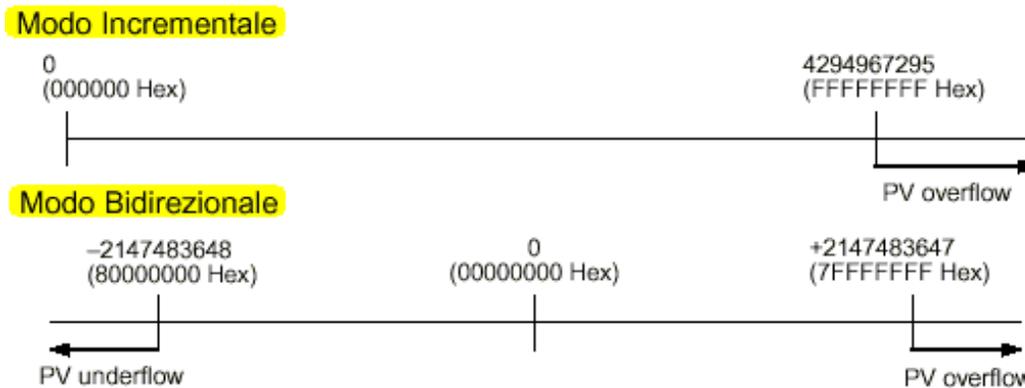


Incrementale:



CJ2M-MD21_ : conteggio lineare

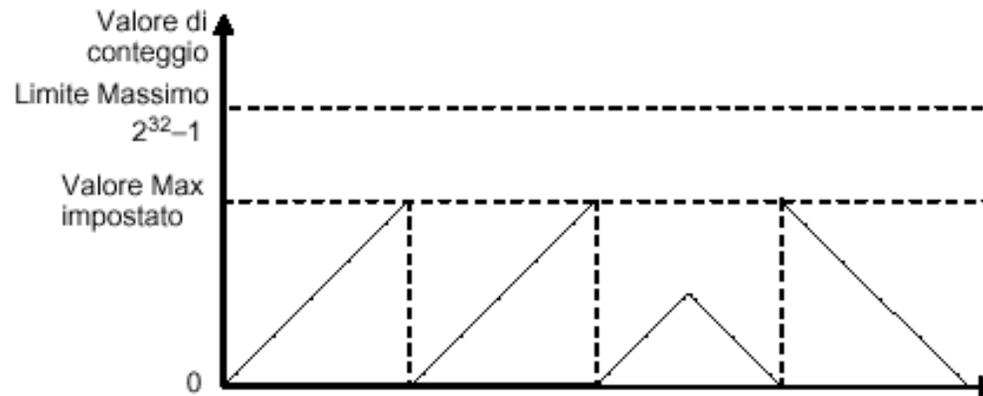
- Modalità di conteggio lineare:



- Gli impulsi vengono conteggiati tra il limite inferiore ed il limite superiore propri del contatore veloce (**32 bit**).
- Se il valore di conteggio fuoriesce dai limiti si genera un errore di Overflow/Underflow ed il conteggio si interrompe.

CJ2M-MD21_: conteggio circolare

- Modalità di conteggio circolare:



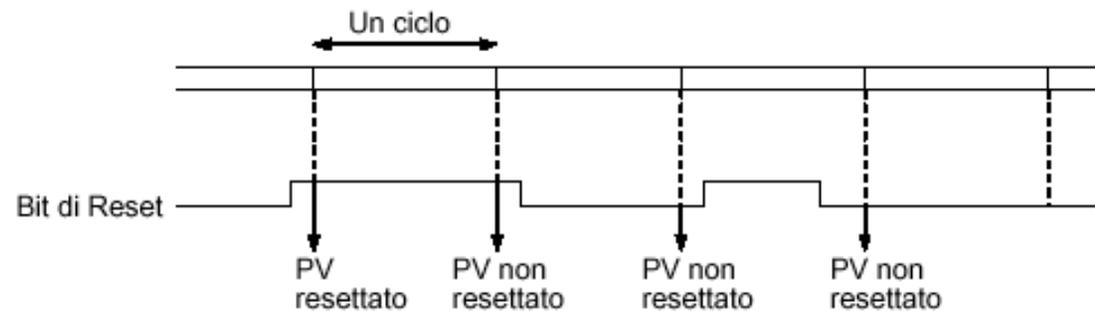
- Gli impulsi vengono conteggiati all'interno di un Range ad anello il cui limite massimo è fissato nel setup del PLC.
- Conseguentemente errori di Overflow/Underflow non possono verificarsi.

CJ2M-MD21_: metodi di reset

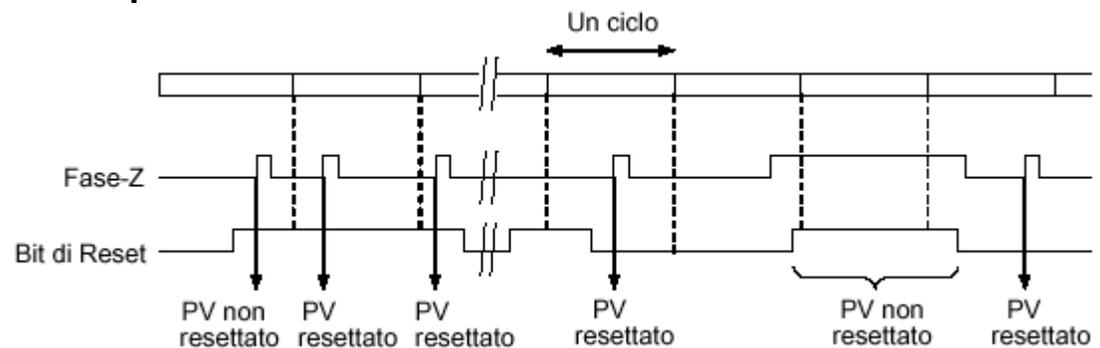
- Metodo di reset software:

- Bit di Reset:

- A531.00 (HSC 0);
 - A531.01 (HSC 1);
 - A531.02 (HSC 2);
 - A531.03 (HSC 3).



- Metodo di reset software più Fase-Z:



CJ2M-MD21_: funzione gate

- La funzione di gate permette di interrompere temporaneamente il conteggio anche se vengono ricevuti degli impulsi.
- Per inibire il conteggio è sufficiente mettere ad ON il bit di gate relativo a ciascun contatore:
 - Gate bit contatore 0: A531.08;
 - Gate bit contatore 1: A531.09;
 - Gate bit contatore 2: A351.10;
 - Gate bit contatore 3: A351.11.
- Se il tipo di reset impostato è 'Reset software più fase Z', nel caso in cui il bit di reset sia ad ON (attesa della fase Z per il reset) il bit di gate viene ignorato.

Impostazione del contatore veloce

Nell'editor delle impostazioni, selezionare il gruppo 'Moduli di I/O' e cliccare il pulsante 'Imposta' nella sezione 'Contatori ad alta velocità'.

Abilitazione del contatore.

Selezione della modalità di conteggio: lineare/circolare.

Valore limite del conteggio circolare.

Selezione del tipo di reset: solo con bit di reset o bit di reset + fronte della fase Z.

Comportamento del confronto CTBL(882) in caso di reset del contatore.

Selezione della modalità di conteggio.

Impostazioni dettagliate contatori ad alta velocità

Elemento	Contatore ad alta velocità 0	Contatore ad alta veloc	Contatore ad :
Impostazione contatore	*Non utilizzare	*Non utilizzare	*Non utilizzare
Modalità di conteggio	*Modalità lineare	*Modalità lineare	*Modalità lineare
Valore max contatore circolare	0	0	0
Metodo di reset	*Fase Z + reset software	*Fase Z + reset softw	*Fase Z + reset
Confronto dopo reset contator	*Arresta	*Arresta	*Arresta
Modalità ingresso a impulsi	*Fase differenziale	*Fase differenziale	*Fase differenz

Le impostazioni predefinite sono indicate da asterischi.

Copia impostazioni contatori ad alta velocità

Copia

Definizione

OK

Guida

Annulla

Copia la configurazione di un contatore su un altro/i contatori.

Riporta la configurazione di tutti i contatori al default.

Esempio: configurazione del contatore veloce

- Configurazione del contatore 0 con la seguente modalità:
 - Reset software;
 - Modalità a fasi differenziali.
- Dopo il trasferimento delle impostazioni sulla CPU, è necessario spegnere e riaccendere il PLC.
- Il valore di conteggio può essere letto visualizzando la word A270 in formato DINT nella finestra di controllo di CX-Programmer.

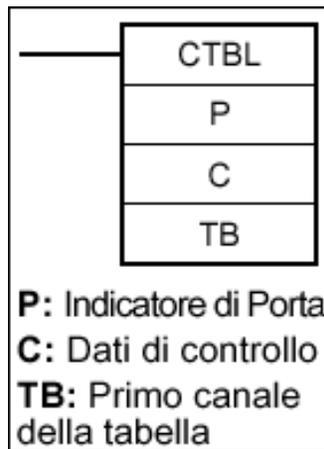
Elemento	Contatore ad alta velocità 0
Impostazione contatore	Frequenza impulso ingresso (100 kHz max.)
Modalità di conteggio	*Modalità lineare
Valore max contatore circolare	0
Metodo di reset	Reset software
Confronto dopo reset contator	*Arresta
Modalità ingresso a impulsi	*Fase differenziale 

La frequenza massima degli impulsi in ingresso può essere impostata a 60 o 100 kHz.

Il valore più basso può limitare i disturbi ad alta frequenza.

La funzione CTBL(882)

- E' possibile eseguire dei confronti tra il valore di conteggio attuale (PV) e valori Target o Range di valori, mandando in esecuzione i Task ad Interrupt associati.
- La registrazione della tabella di comparazione avviene per mezzo dell'istruzione CTBL(882):



P	Porta
0000 hex	Contatore veloce 0.
0001 hex	Contatore veloce 1.
0002 hex	Contatore veloce 2.
0003 hex	Contatore veloce 3.

C	Significato
0000 hex	Registra una tabella di valori Target e avvia la comparazione.
0001 hex	Registra una tabella con 8 Range e avvia la comparazione.
0002 hex	Registra una tabella di valori Target ma <u>non avvia</u> la comparazione.
0003 hex	Registra una tabella con 8 Range ma <u>non avvia</u> la comparazione.
0004 hex	Registra una tabella con max 32 Range e avvia la comparazione.
0005 hex	Registra una tabella con max. 32 Range ma <u>non avvia</u> la comparazione.

Comparazione a Target

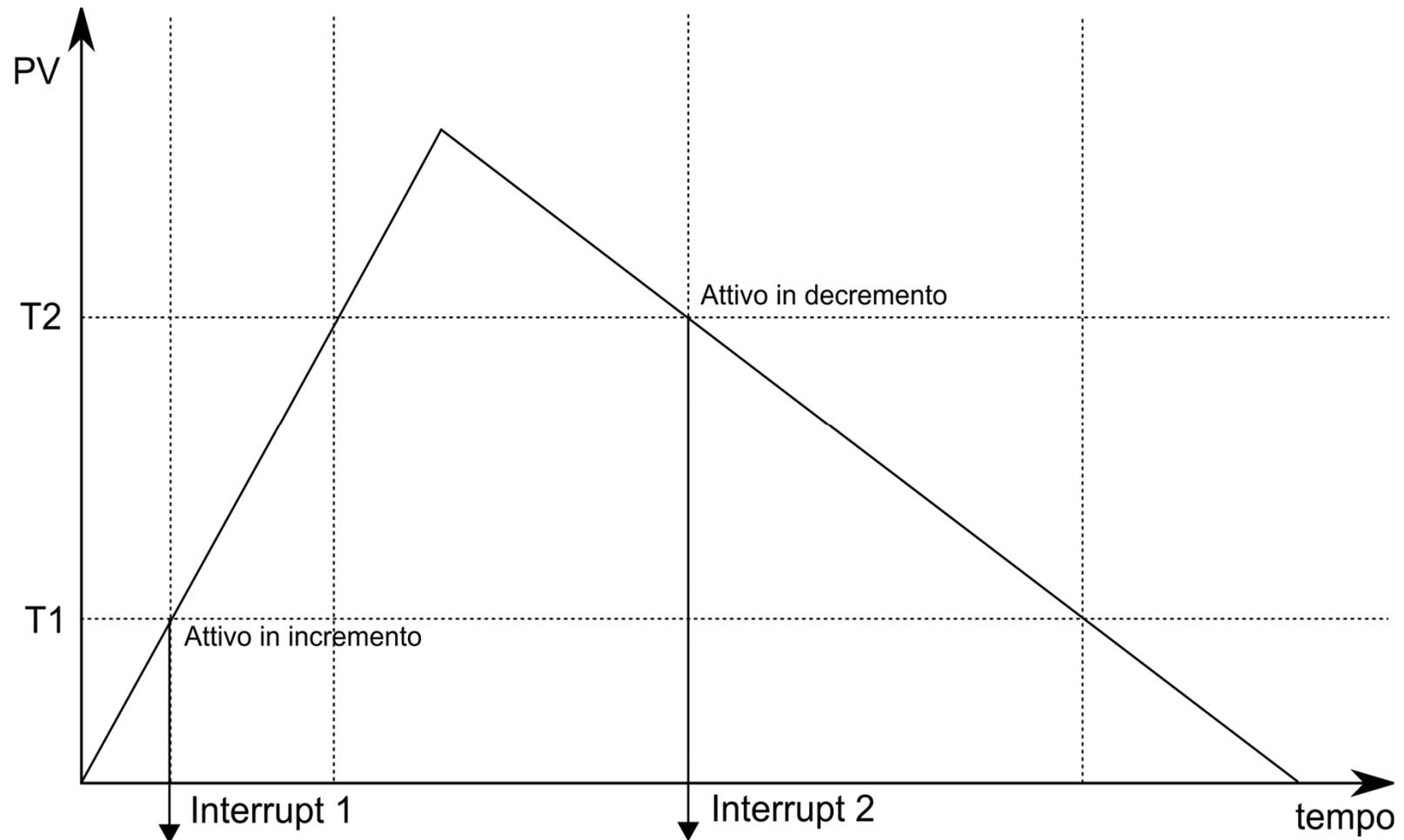
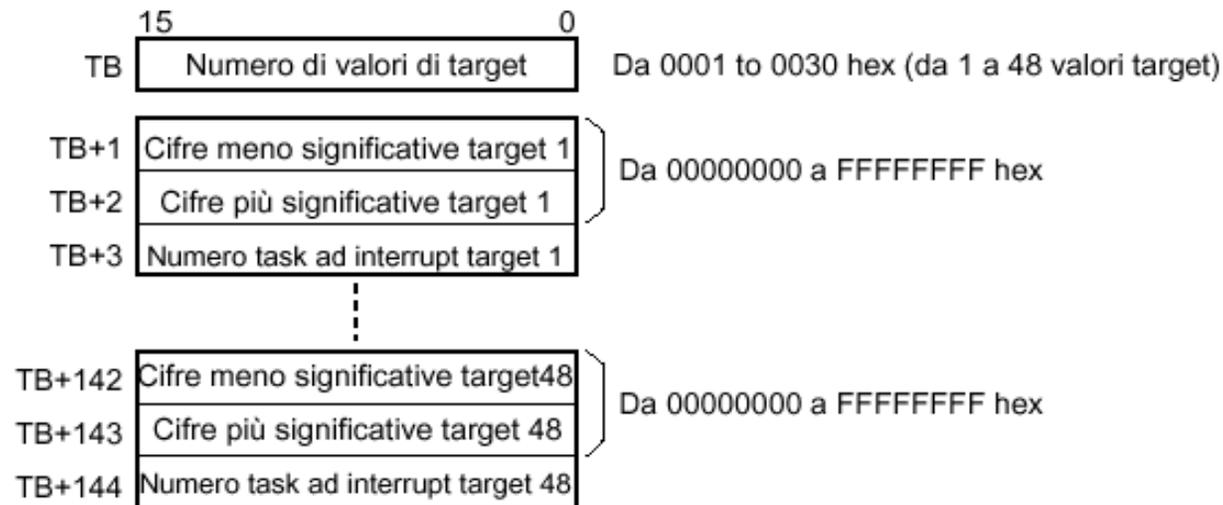
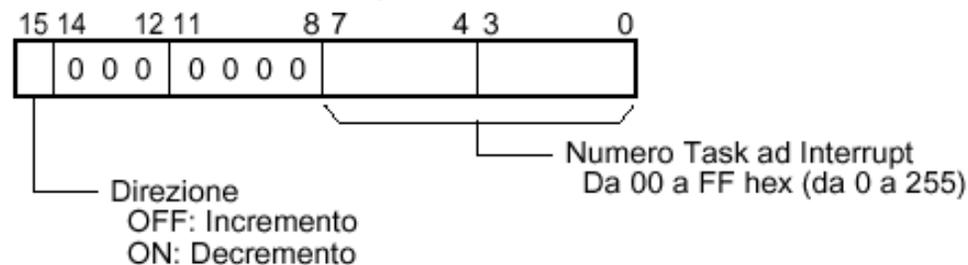


Tabella di comparazione a Target

- Formato della tabella per valori di comparazione a Target:



Numero Task ad Interrupt



Comparazione a Range

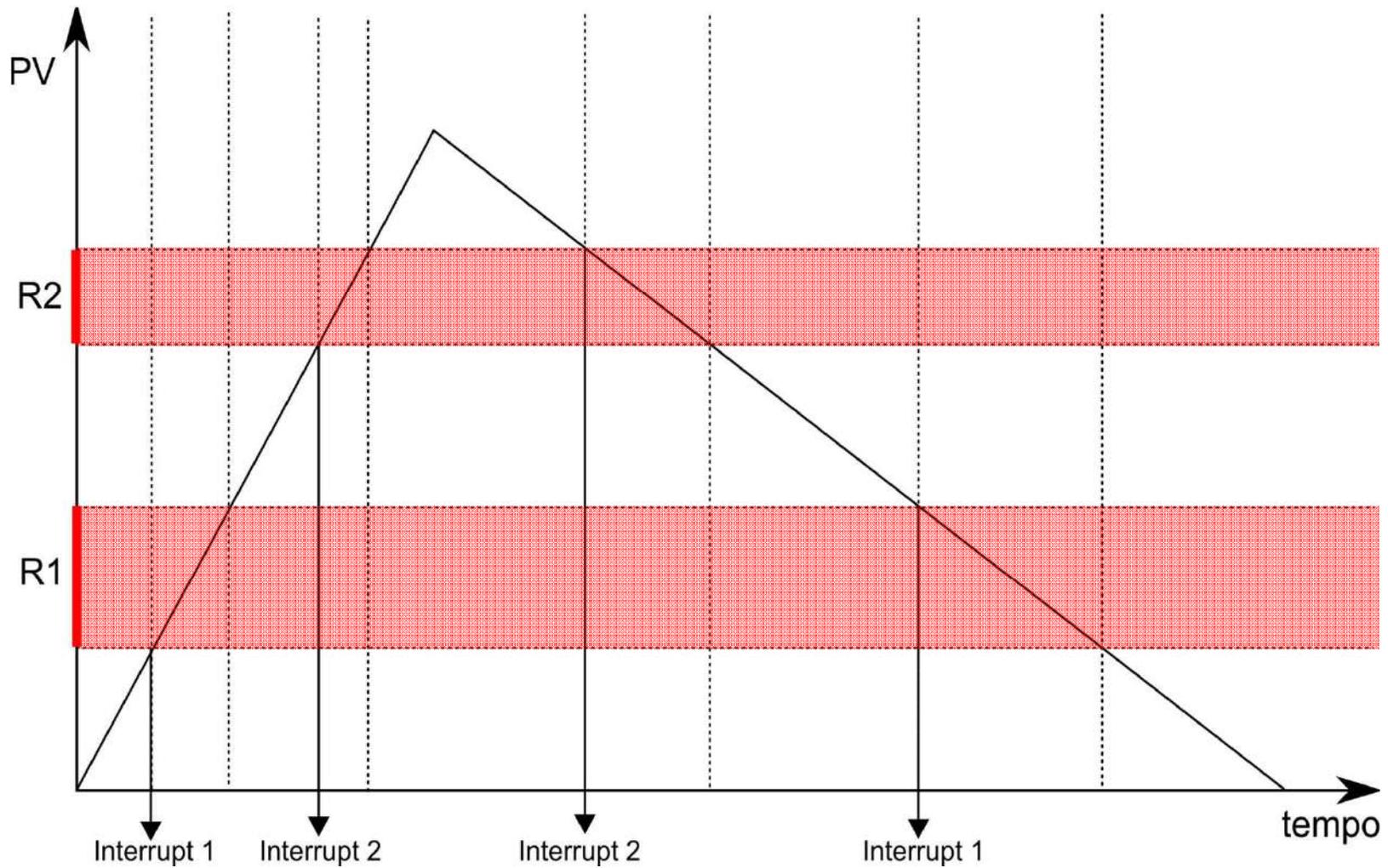
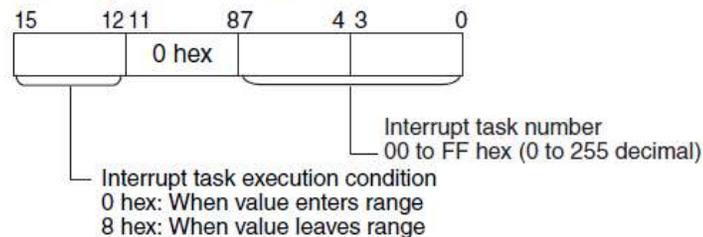


Tabella di comparazione a Range

- Tabella per le modalità CTBL(882) 4 e 5 (32 Range max.):

TB	Number of ranges	0001 to 0020 hex (1 to 32 decimal)
TB+1	Lower word of range 1 lower limit	0000 0000 to FFFF FFFF hex
TB+2	Upper word of range 1 lower limit	
TB+3	Lower word of range 1 upper limit	0000 0000 to FFFF FFFF hex
TB+4	Upper word of range 1 upper limit	
TB+5	Range 1 interrupt task number	
⋮		
TB+156	Lower word of range 32 lower limit	0000 0000 to FFFF FFFF hex
TB+157	Upper word of range 32 lower limit	
TB+158	Lower word of range 32 upper limit	0000 0000 to FFFF FFFF hex
TB+159	Upper word of range 32 upper limit	
TB+160	Range 32 interrupt task number	

Interrupt task number



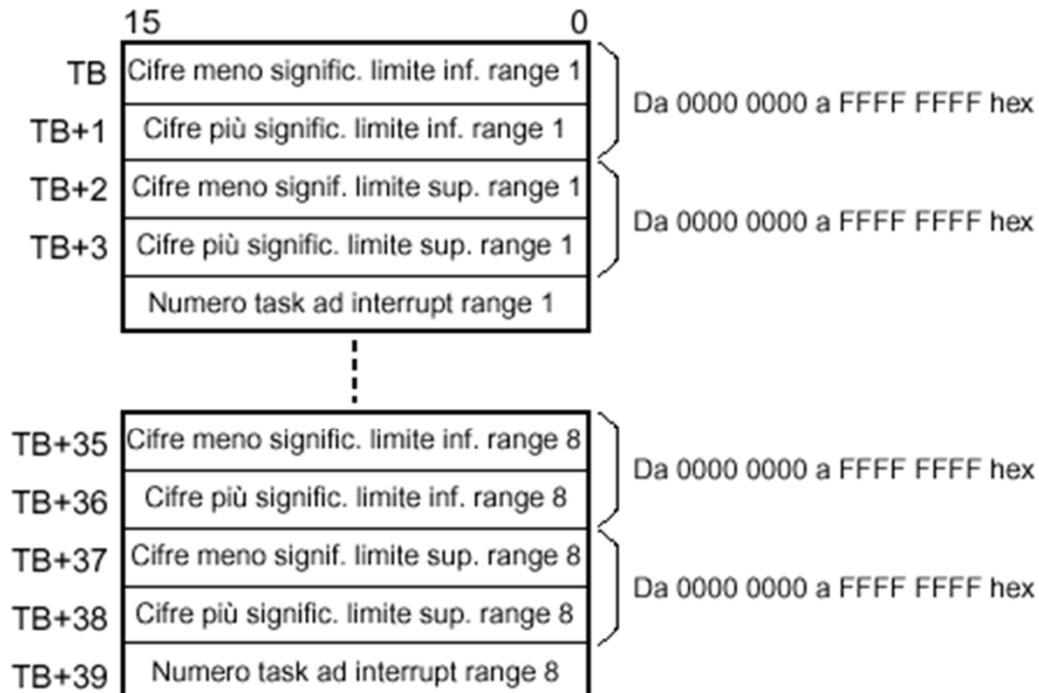
AAAA hex: Do not execute interrupt task. (Flag output only)
 FFFF hex: Ignore the settings for this range.

Informazioni sulla CTBL(882)

- Se più di una condizione è soddisfatta all'interno di un ciclo, verrà eseguito solo il primo dei Task ad Interrupt relativi, presenti in tabella.
- Il secondo Interrupt verrà eseguito nel ciclo successivo, e così via.
- I flag che indicano in quale dei Range si trova il PV (Range Comparison Condition Met Flags) sono:
 - Per il contatore veloce 0: da A274.00 a A274.07.
 - Per il contatore veloce 1: da A275.00 a A275.07.
- Il bit 8 dei canali di controllo indica se la comparazione è attiva (ON) o meno (OFF), il bit 10 la direzione.

Modalità compatibile con PLC CJ1M-CPU2_

- Impostando il parametro C della CTBL(882) a 1 o 3, si utilizza la modalità compatibile CJ1M:

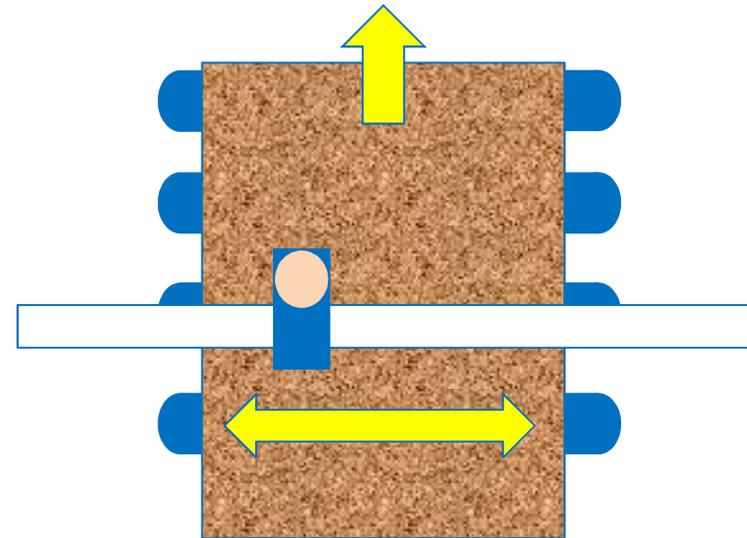


Numero Task ad interrupt
 Da 0000 a 00FF hex: Task ad interrupt da 0 a 255
 AAAA hex: Non esegue il task ad interrupt.
 FFFF hex: Ignora le impostazioni per questo range

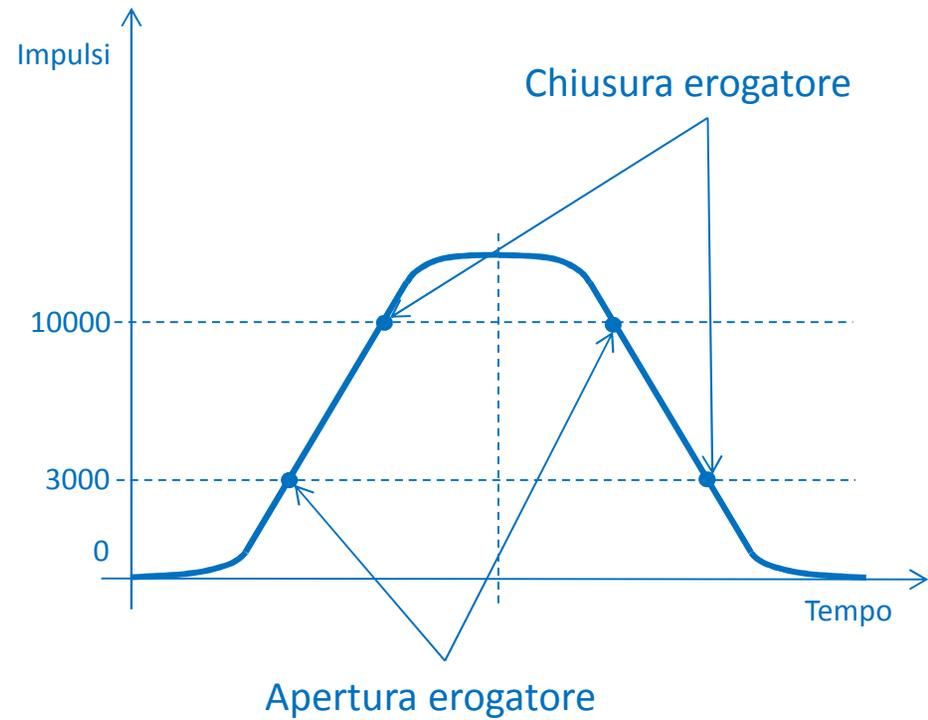
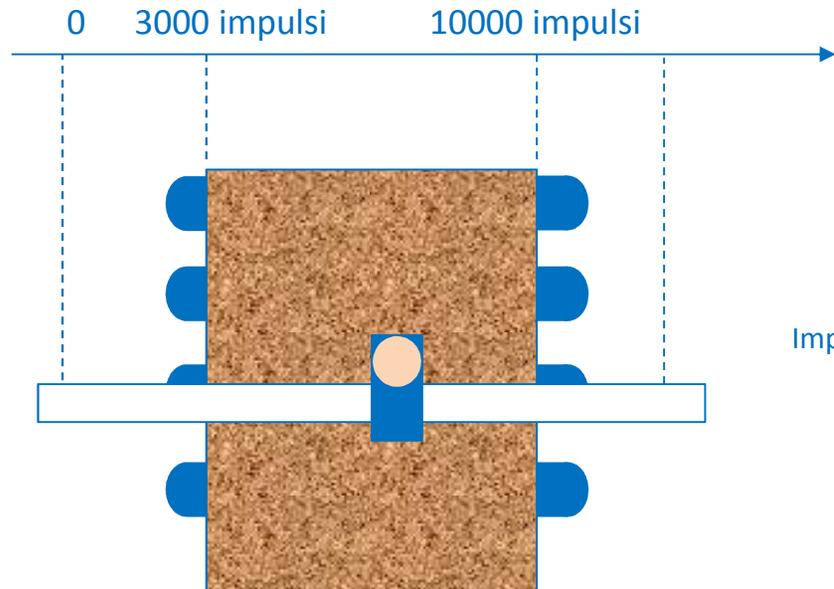
- In questa modalità è possibile definire un massimo di 8 Range.
- La dimensione della tabella è fissa a 40 word anche se non tutti i Range sono utilizzati.
- Non è previsto l'Interrupt in uscita dal Range.

Esempio: applicatore colla

- L'applicazione della colla sui pannelli avviene con un asse che trasla orizzontalmente.
- Dopo ogni passata il pannello viene fatto avanzare (nell'esempio questa parte non verrà considerata).
- Poiché la corsa totale dell'applicatore è maggiore della larghezza del pannello, è necessario stabilire dei punti (quote) di inizio e di fine per l'erogazione.
- L'applicazione della colla avviene in entrambe le direzioni; al ritorno, la quota di fine applicazione diventerà quella di inizio e quella di inizio applicazione utilizzata all'andata diventerà il punto di chiusura per il ritorno.



Esempio: funzionamento dell'applicatore

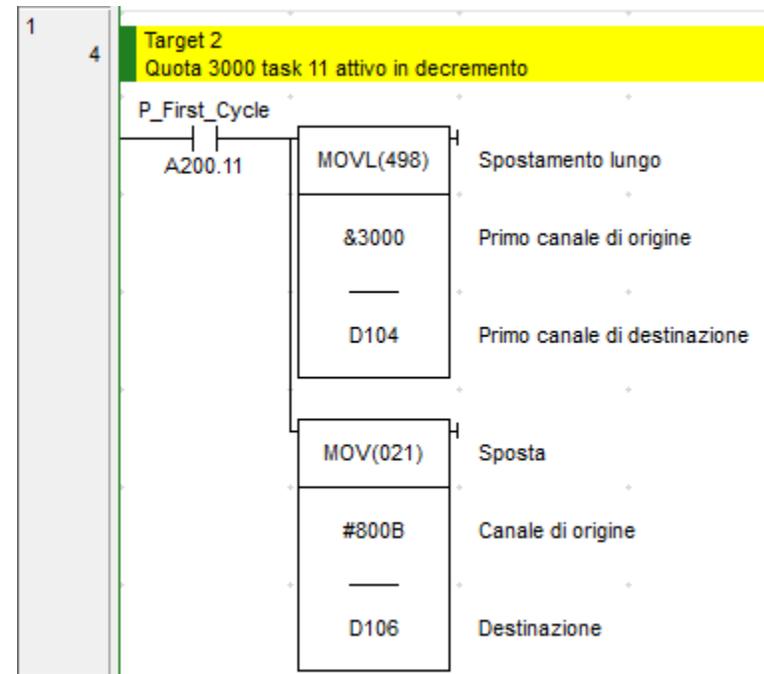
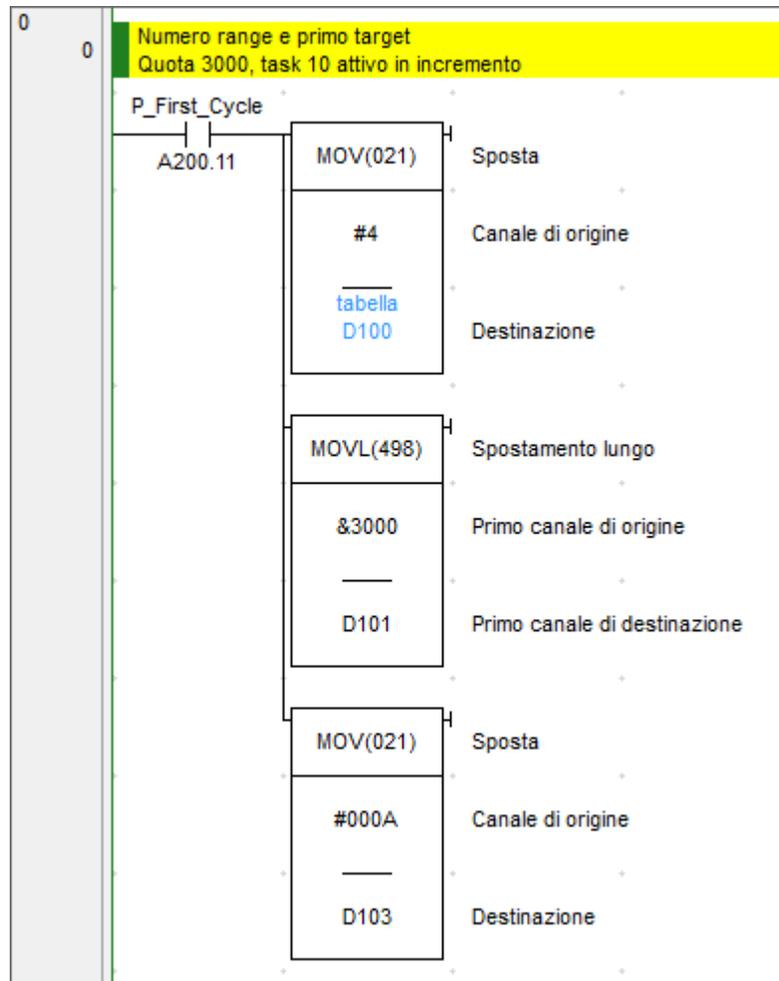


Esempio: definizione tabella Target

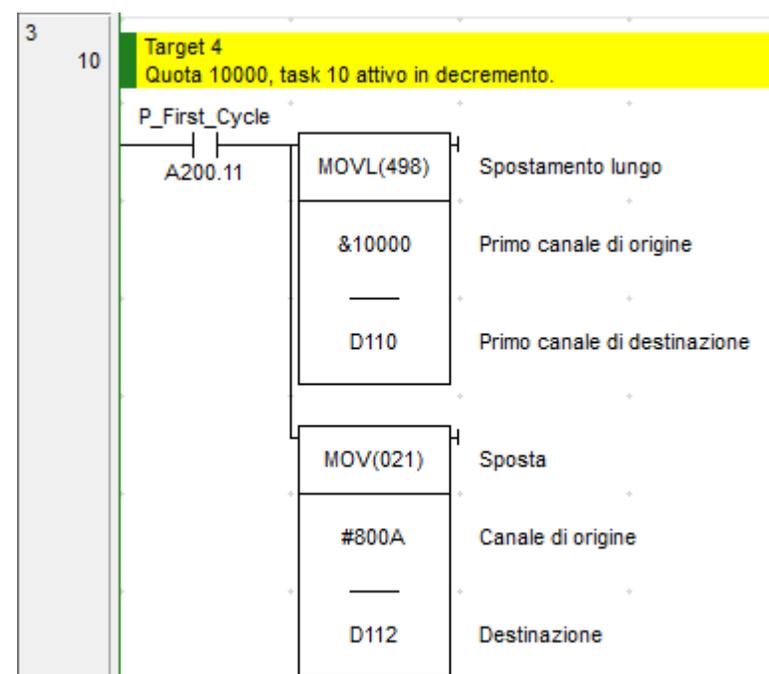
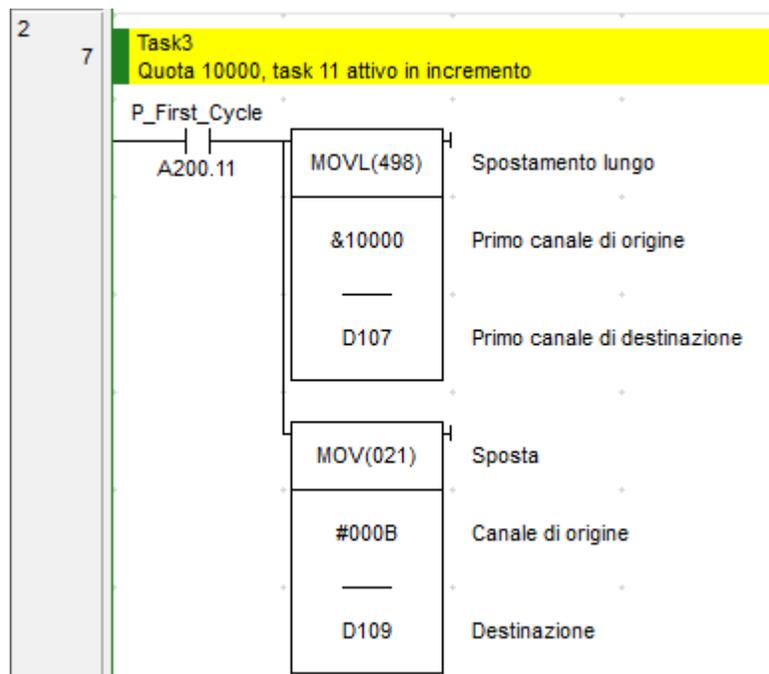
- Si utilizzerà la funzione CTBL(882) per la comparazione a Target.
- Sono necessari 4 Target, due Target ripetuti due volte cambiando la direzione di intervento.
- La tabella viene posta a partire dal D100.
- Per risparmiare tempo è possibile scrivere i valori nei DM con CX-Programmer senza usare le istruzioni MOV(021).

Indirizzo	Valore	Descrizione
D100	#0004	Numero di Target.
D101	#0BB8	LSW Target (3000).
D102	#0000	MSW Target.
D103	#000A	Task 10 attivato in incremento.
D104	#0BB8	LSW Target (3000).
D105	#0000	MSW Target.
D106	#800B	Task 11 attivato in decremento.
D107	#2710	LSW Target (10000).
D108	#0000	MSW Target.
D109	#000B	Task 11 attivato in incremento.
D110	#2710	LSW Target (10000).
D111	#0000	MSW Target.
D112	#800A	Task 10 attivato in decremento.

Esempio: preparazione della tabella dei Target

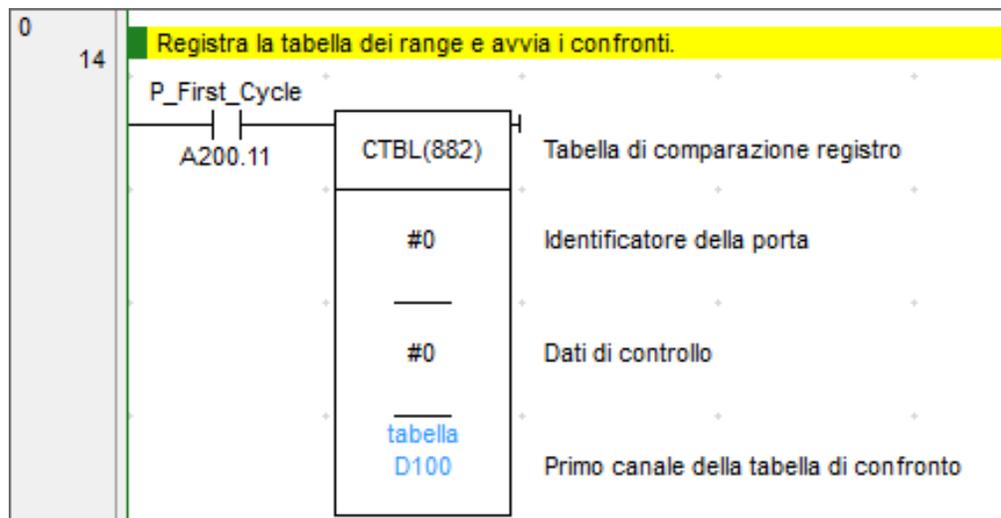


Esempio: preparazione della tabella dei Target

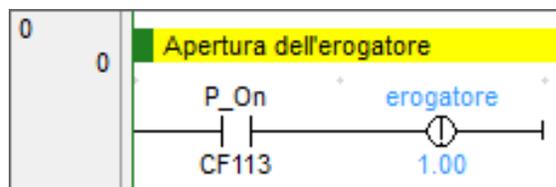


Esempio: Task principale e Task a Interrupt

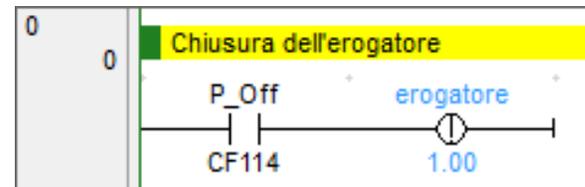
Programma principale, dopo la definizione della tabella, l'istruzione CTBL(882) attiva i confronti.



Task a Interrupt 10 apertura erogatore.



Task a Interrupt 11 chiusura erogatore.





Uscite a treno d'implusi

Corso intermedio CPS2

Le uscite a treno di impulsi

- I moduli MD211 e MD212 del CJ2M possono generare **2 treni di impulsi** sfruttando le uscite dedicate.
- Un treno di impulsi può essere per esempio associato ad un motore passo-passo o ad un servoazionamento, per comandare il posizionamento di un asse.
- Il CJ2M permette di gestire sia un controllo di **velocità** (dove gli impulsi vengono generati in modo continuo, a specificate frequenze) che un controllo di **posizione** (generazione di un numero di impulsi ben determinato).

Le uscite a treno di impulsi

- Caratteristiche principali:
 - Modalità CW/CCW o impulso più direzione: il metodo è selezionabile tramite l'operando di un'istruzione e deve essere comune ad entrambe le uscite;
 - Selezione automatica della direzione per il posizionamento in coordinante assolute (quando cioè l'origine è stata determinata);
 - Controllo di tipo triangolare in automatico nel caso in cui il numero di impulsi specificato sia insufficiente per raggiungere la frequenza di target;
 - Modifica del target durante il posizionamento (Multiple Start Function);
 - Possibilità di passare, durante il posizionamento, da controllo di velocità a controllo di posizione.

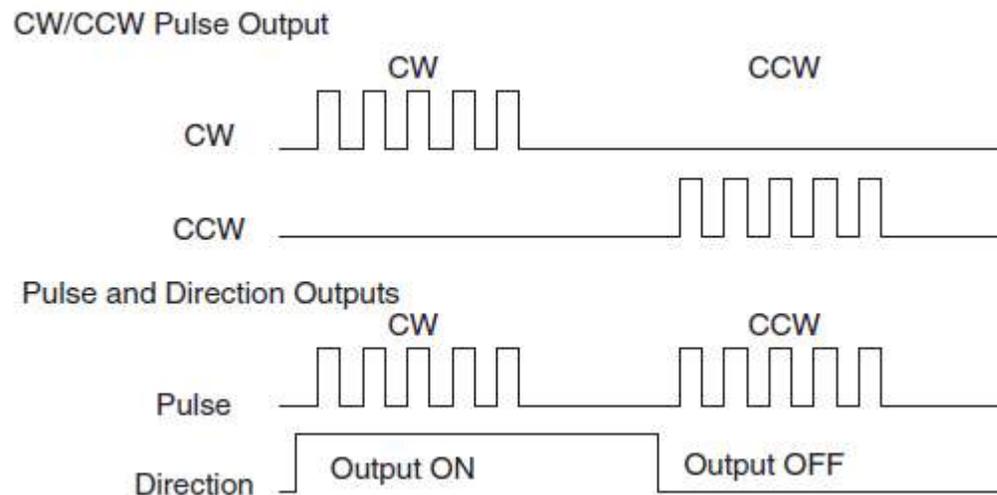
Istruzioni per la generazioni di impulsi

- Le caratteristiche e le modalità di esecuzione delle uscite a treno di impulsi vengono definite tramite l'esecuzione di istruzioni dedicate, direttamente da programma ladder.

Istruzione	Mnemonico	Codice funzione	Utilizzo principale
SET PULSES	PULS	886	Impostazione del numero di impulsi dell'uscita impulsiva
SPEED OUTPUT	SPED	885	Uscita impulsiva senza accelerazione o decelerazione.
ACCELERATION CONTROL	ACC	888	Uscita impulsiva con accelerazione e decelerazione.
PULSE OUTPUT	PLS2	887	Controllo trapezoidale
ORIGIN SEARCH	ORG	889	Ricerca dell'origine e ritorno
MODE CONTROL	INI	880	Interruzione dell'uscita impulsiva o modifica dei PV
HIGH-SPEED COUNTER PV READ	PRV	881	Lettura dei PV

Specifiche delle uscite a treno di impulsi

- Modalità di uscita a treno di impulsi:



- Specifiche:
 - La frequenza massima delle uscite è 100 kHz;
 - Il numero massimo di impulsi generabili è 2.147.483.647 in modalità relativa. In modalità assoluta il valore di target può variare tra -2.147.483.648 e 2.147.483.647.

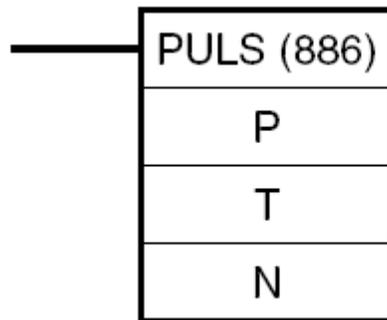
Uso dell'uscita a treno d'impulsi

- Le uscite a treno di impulsi possono operare in due modalità, indipendente e continua.
 - **Modalità indipendente:** la generazione di impulsi si arresta al raggiungimento del numero di impulsi impostato dall'istruzione PULS(886). L'istruzione SPED(885) avvia l'uscita e imposta la velocità;
 - **Modalità continua:** non c'è nessuna condizione di arresto degli impulsi, si utilizza la sola istruzione SPED(885).

Istruzione PULS(886)

- L'istruzione **PULS(886)** (set pulses) determina il numero totale di impulsi da generare attraverso un'uscita a treno di impulsi.
- Con l'istruzione PULS(886) è possibile lavorare in modalità relativa o assoluta.
 - **Modalità relativa:** numero di impulsi conteggiato dalla posizione attuale;
 - **Modalità assoluta:** il numero di impulsi rappresenta la posizione rispetto all'origine.

Istruzione PULS(886)



P: identificatore della porta
T: tipo di impulsi
N: numero di impulsi

P	Porta
000	Uscita a treno di impulsi 0.
001	Uscita a treno di impulsi 1.
002	Uscita a treno di impulsi 2.
003	Uscita a treno di impulsi 3.

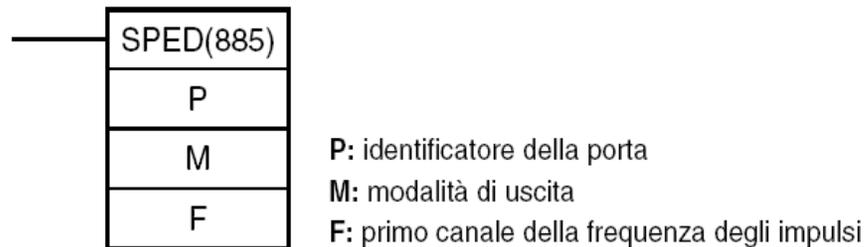
T	Tipo di impulsi
000	Modalità relativa.
001	Modalità assoluta.

N	Numero di impulsi
	Numero di impulsi espresso su 2 word in complemento a 2. La prima word contiene la parte meno significativa.

Istruzione SPED(885)

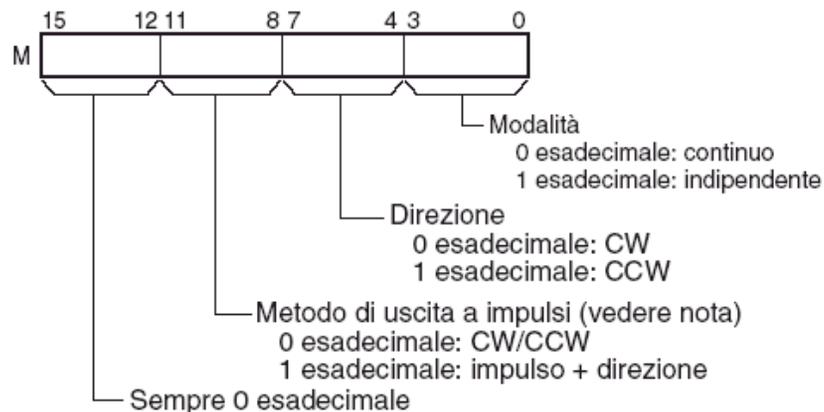
- L'istruzione **SPED(885)** (speed output) determina:
 - la frequenza di uscita degli impulsi espressa in Hertz;
 - il modo di funzionamento (continuo/indipendente);
 - la direzione di movimento (orario/antiorario) per la modalità relativa;
 - il tipo di uscita a impulsi (due fasi o impulso + direzione).
- Sui PLC CJ la frequenza è impostabile da 0 a 100 kHz.

Istruzione SPED(885)



M: modalità di uscita

Il valore di M determina la modalità di uscita.



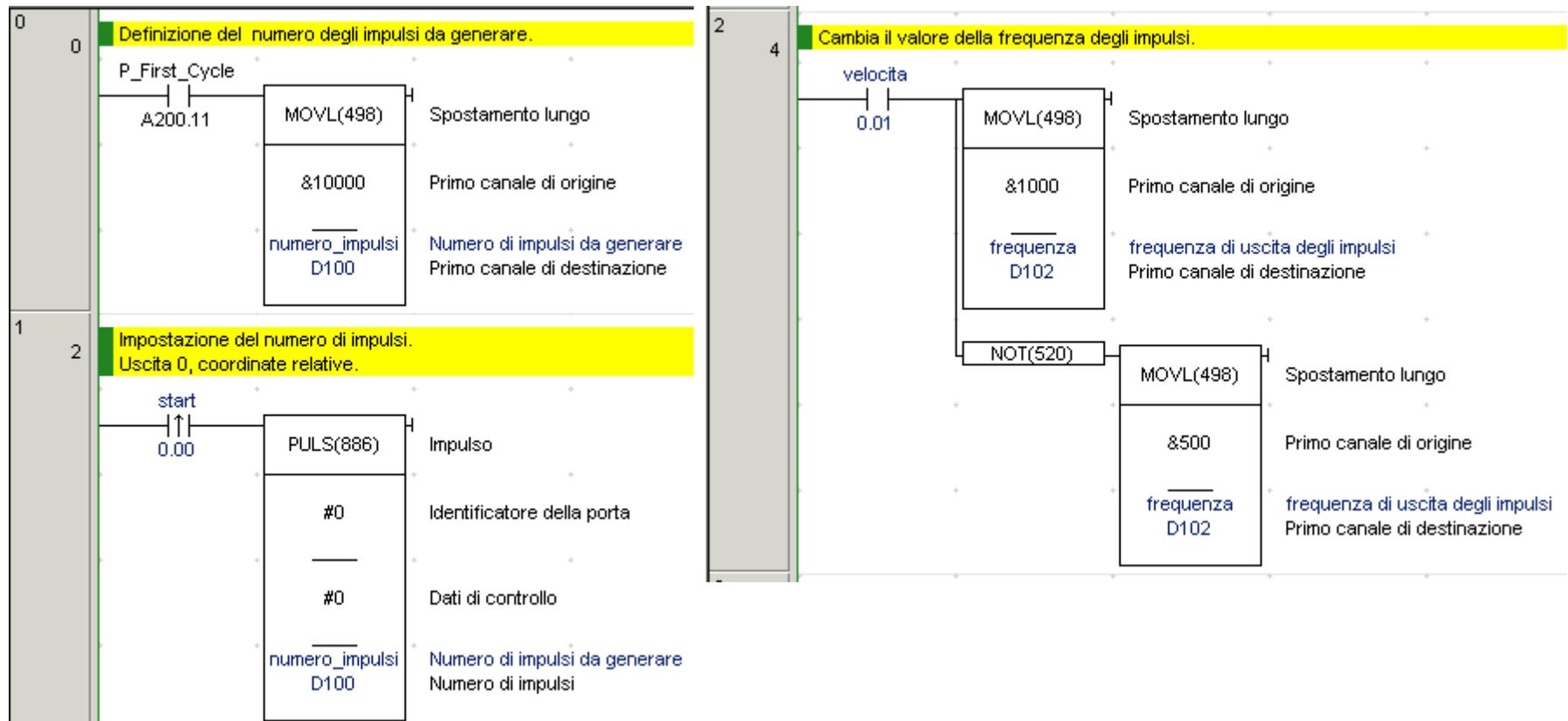
P	Porta
000	Uscita a treno di impulsi 0.
001	Uscita a treno di impulsi 1.
002	Uscita a treno di impulsi 2.
003	Uscita a treno di impulsi 3.

Numero di impulsi	
F	F e la word seguente contengono il valore della frequenza di uscita degli impulsi. Il valore è espresso su due word e il range è 0-100.000 (0-186A0 hex).

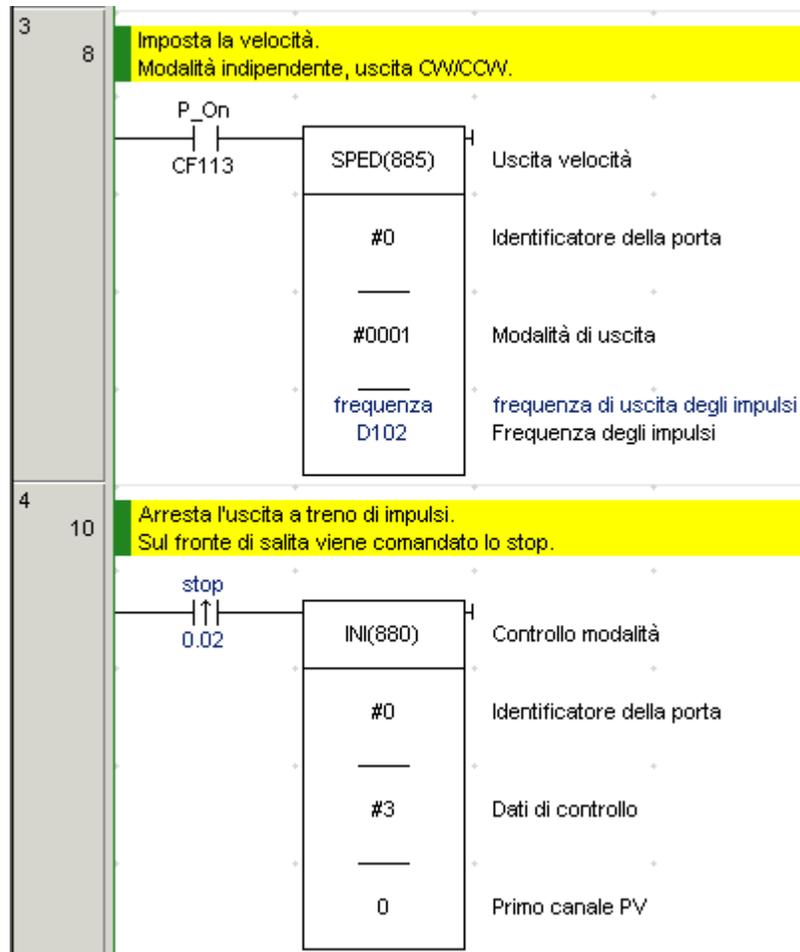
Esempio: uso dell'uscita a treno d'impulsi

- Lo scopo dell'esercizio è il seguente:
 - Alla chiusura del contatto 0.0, viene avviata l'uscita a impulsi 0 con una frequenza iniziale di 500 Hz;
 - La frequenza di uscita dipenderà dallo stato dell'ingresso 0.1; se il bit è a "0" la frequenza sarà 500 Hz, se il bit è "1", la frequenza sarà 1 kHz;
 - L'uscita si arresterà dopo aver generato 10.000 impulsi o alla chiusura del contatto 0.2 (utilizzando l'istruzione INI).

Soluzione: uso dell'uscita a treno d'impulsi



Soluzione: uso dell'uscita a treno d'impulsi



Backup dei programmi

Corso intermedio CPS2

La funzione di “Easy Backup”

- È possibile eseguire il backup di **tutti** i dati (programmi utente, parametri e memoria I/O) sulla memory card premendo un pulsante sulla CPU.
- In questo modo, è possibile eseguire immediatamente il backup e il ripristino di tutti i dati presenti nella CPU senza dover utilizzare un dispositivo di programmazione.
- Per abilitare l'operazione occorre mettere a ON lo switch 7 sulla CPU.
- La procedura si divide in 2 fasi “**Backup e Restore**”.

Easy Backup: individuazione dei componenti

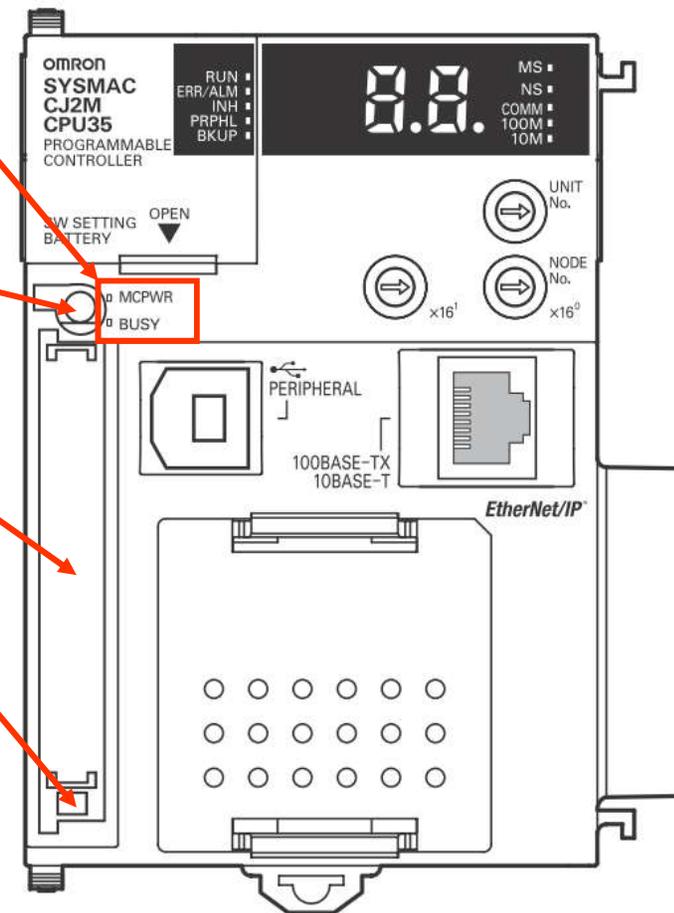
Led di stato della scheda di memoria:
 MCPWR: scheda presente e alimentata.
 BUSY: accesso alla scheda in corso.

Pulsante di alimentazione della scheda di memoria "MC Pwr".

Slot della scheda di memoria (formato CF).

Tasto di espulsione della scheda.

Nota: prima di rimuovere la scheda assicurarsi che entrambi i Led siano spenti.



Backup

- L'operazione di backup può avvenire anche durante il normale funzionamento del PLC.
- Per effettuare il “**Backup**”:
 - Inserire la memory card nello slot di memoria;
 - Portare a ON il dip switch 7 sulla CPU;
 - Premere per almeno 3 secondi il pulsante “MC Power”;
 - Attendere lo spegnimento di entrambi i led relativi alla CF;
 - Portare a OFF il dip switch 7;
 - Estrarre la memory card.

Restore

- L'operazione di restore può avvenire solo all'accensione del PLC.
- Per effettuare il **“Restore”**:
 - Spegnere il PLC;
 - Inserire la memory card nello slot di memoria;
 - Portare a ON il dip switch 7 sulla CPU;
 - Accendere il PLC;
 - Attendere lo spegnimento di entrambi i led relativi alla CF;
 - Spegnere il PLC;
 - Portare a OFF il dip switch 7;
 - Estrarre la memory card;
 - Accendere il PLC.

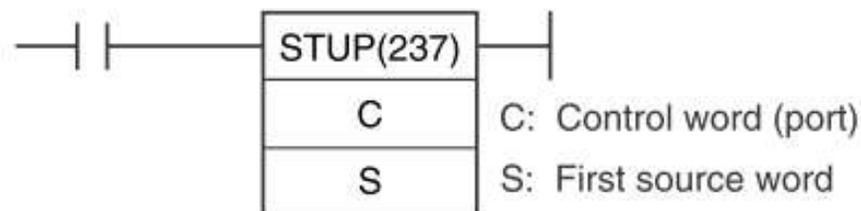


Scambio dati

Corso intermedio CPS2

Impostazione porte di comunicazione

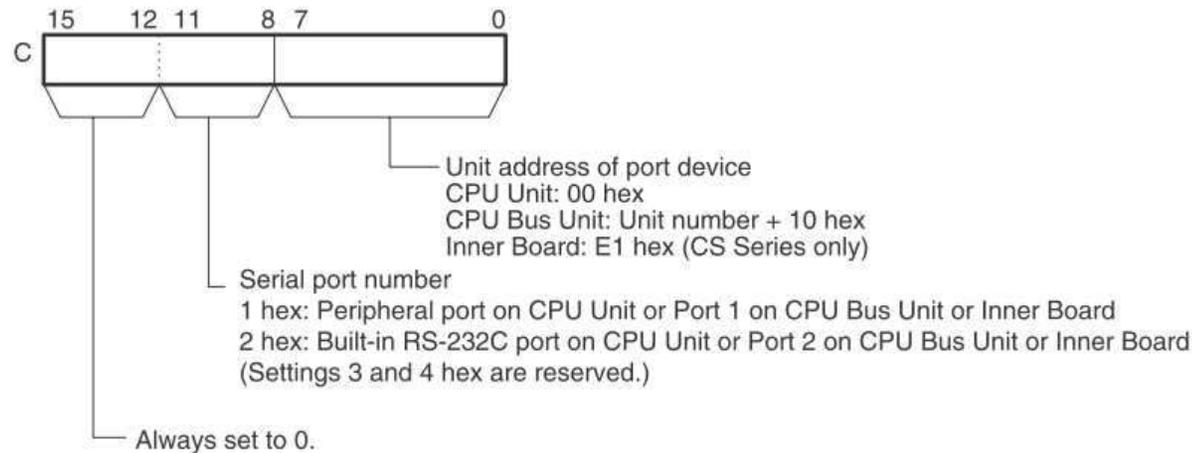
- La modalità di utilizzo delle porte di comunicazione è specificata nell'area di setup del PLC a loro destinata.
- Per esempio, per impostare il protocollo di comunicazione della porta seriale occorre intervenire nel setup del PLC tramite CX-Programmer.
- In alternativa è possibile cambiare le impostazioni direttamente da programma, utilizzando l'istruzione **STUP(237)**.



Impostazione porte di comunicazione

- L'istruzione STUP(237) scriverà **10 canali** (da S a S+9) nell'area di setup dei parametri di comunicazione relativi alla porta indicata in 'C'.
- 'S' = #0000 riporta la porta alle impostazioni di default.

C: Control word (port)

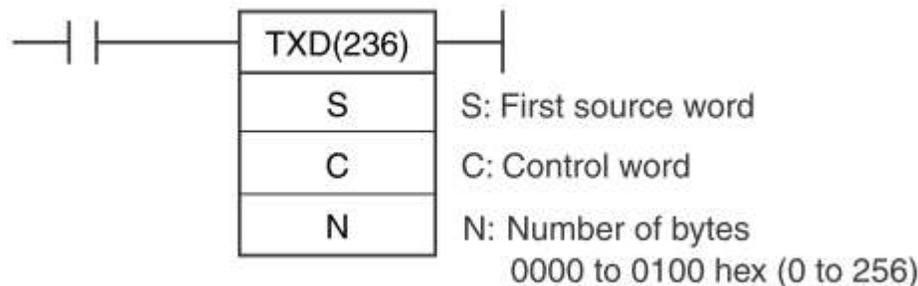


Le istruzioni TXD e RXD

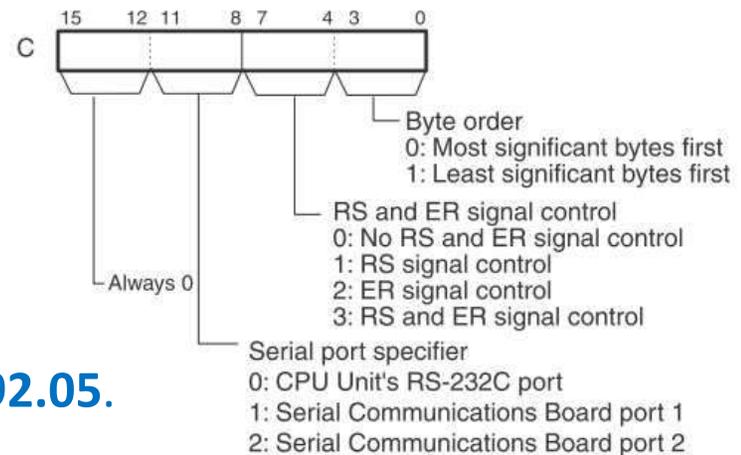
- Le istruzioni **TXD(236)** e **RXD(235)** inviano e ricevono dati nelle comunicazioni in protocollo libero (custom) tra il PLC e dispositivi esterni quali ad esempio:
 - PC e stampanti;
 - Altri PLC;
 - Sensori intelligenti e termoregolatori;
 - Servoazionamenti;
 - Bilance;
 - Lettori di carte magnetiche;
 - Qualunque dispositivo dotato di porta seriale RS-232C.

L'istruzione TXD

- L'istruzione **TXD(236)** viene programmata con tre parametri:
- La TXD(236) invierà in uscita sulla porta seriale N Byte (caratteri) a partire da **S** fino a **S+(N/2)-1**.
- Verranno aggiunti i codici di inizio e di fine specificati per il protocollo libero nelle impostazioni del PLC.



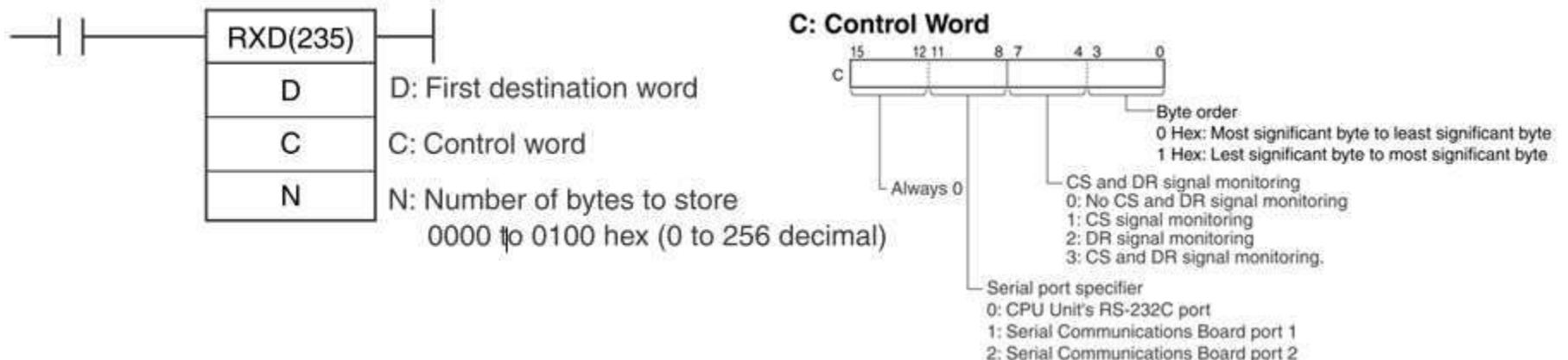
C: Control word



Il flag di abilitazione alla trasmissione è: **A392.05**.

L'istruzione RXD

- L'istruzione **RXD(235)** viene programmata con tre parametri:
- La RXD(235) leggerà i dati ricevuti dalla porta seriale e salverà N byte di dati nei canali da **D** a **D+(N/2)-1**.
- In protocollo libero verranno ricevuti anche i codici di inizio e di fine specificati nelle impostazioni del PLC.

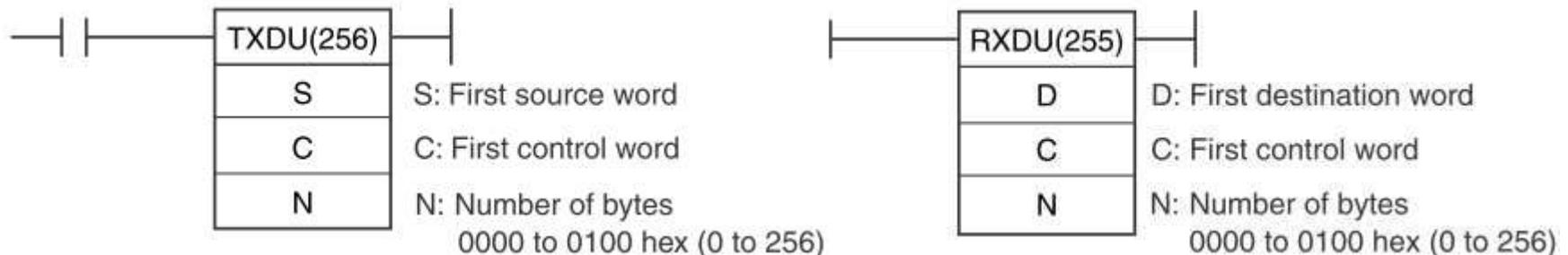


L'istruzione RXD

- L'istruzione RXD(235) viene utilizzata controllando i flag e i canali del buffer di ricezione, che sono:
 - **A392.06: flag di fine ricezione** (ad ON quando la ricezione in protocollo libero è stata completata, oppure quando è stato ricevuto il numero di byte specificato, o quando è stato incontrato il codice di fine stringa specificato);
 - **A392.07: flag di overflow** (ad ON se sono stati ricevuti più byte di quelli previsti, oppure se non è stato ricevuto niente al termine della ricezione o dopo il codice di fine stringa o il 256mo byte);
 - **A393: n° di byte ricevuti** (in esadecimale).

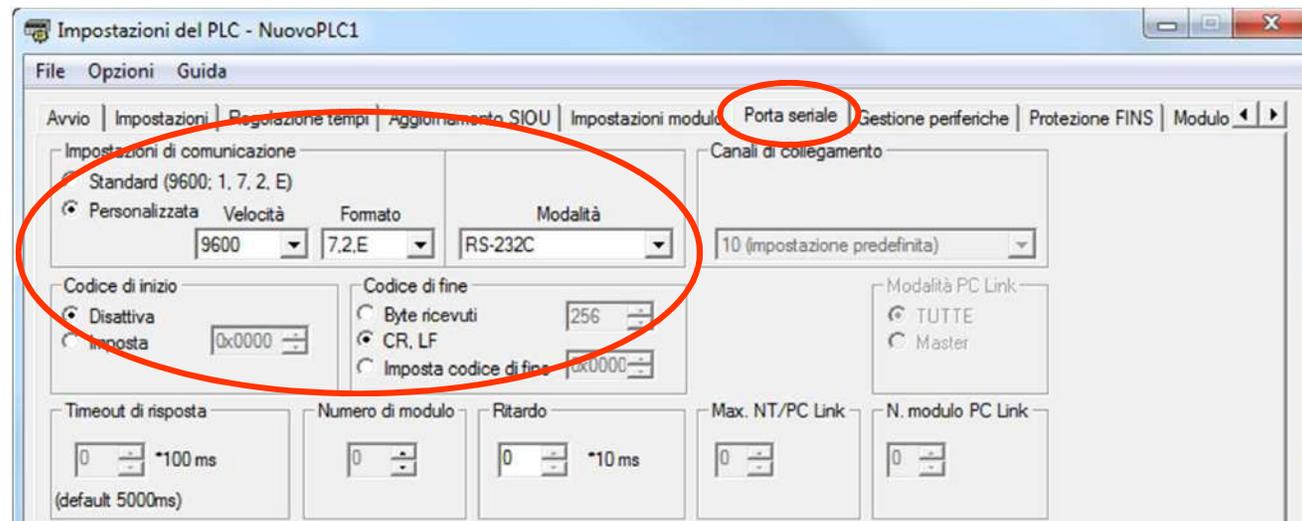
Le istruzioni TXDU e RXDU

- Le istruzioni TXDU(256) e RXDU(255) sono equivalenti alle istruzioni TXD(236) e RXD(235).
- Permettono di utilizzare le porte seriali delle schede SCU21/31/41 del CJ e CS1 e delle schede SCB21/41 del CS1.
- Il funzionamento è analogo alle istruzioni TXD/RXD.



Esempio: utilizzo di TXD e RXD

- Impostare la porta seriale del PLC in questo modo:

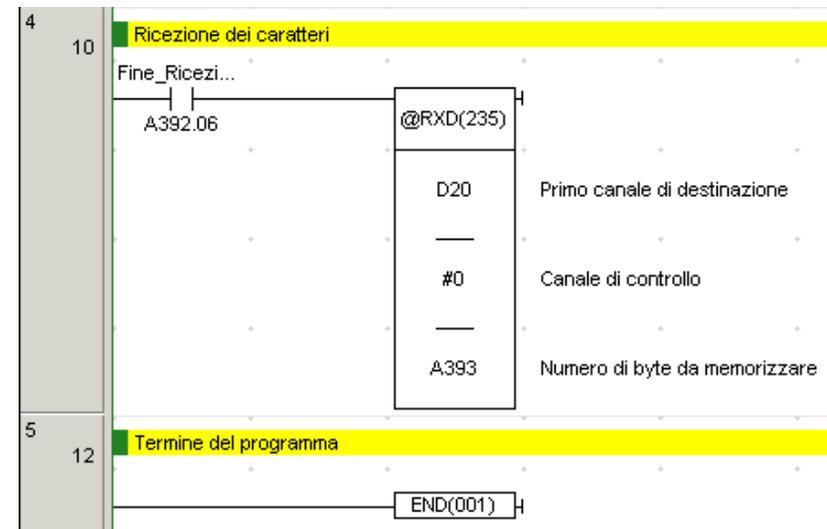
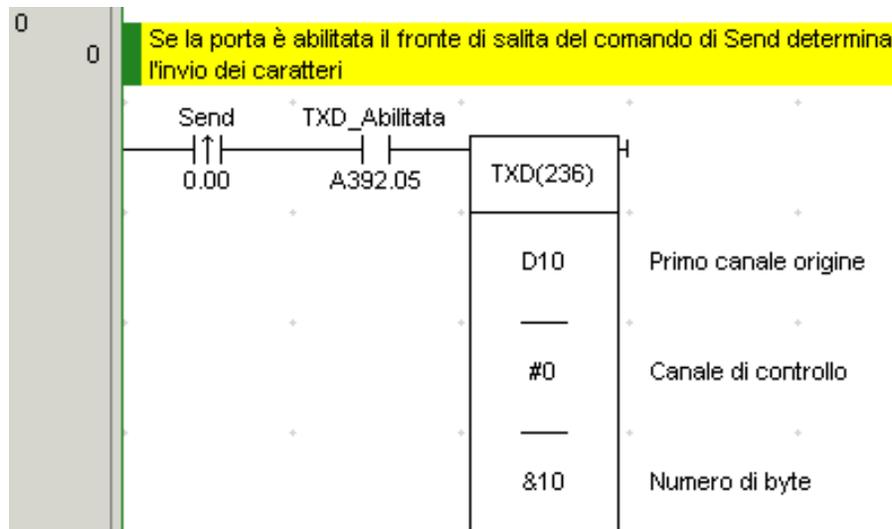


- ...e realizzare un programma che:
 - A fronte di un comando sull'ingresso 0.00 trasmetta una stringa di 10 caratteri i cui codici ASCII sono memorizzati a partire dal canale D10.
 - Riceva, autonomamente, a partire dal canale D20 tutte le stringhe di caratteri delimitate da un CR LF.

Soluzione: utilizzo di TXD e RXD

Caratteri da inviare:

	+0	+1	+2	+3	+4	+5
D00010	4F4D	524F	4E20	434A	314D	



Il 'Terminal' di Multiway

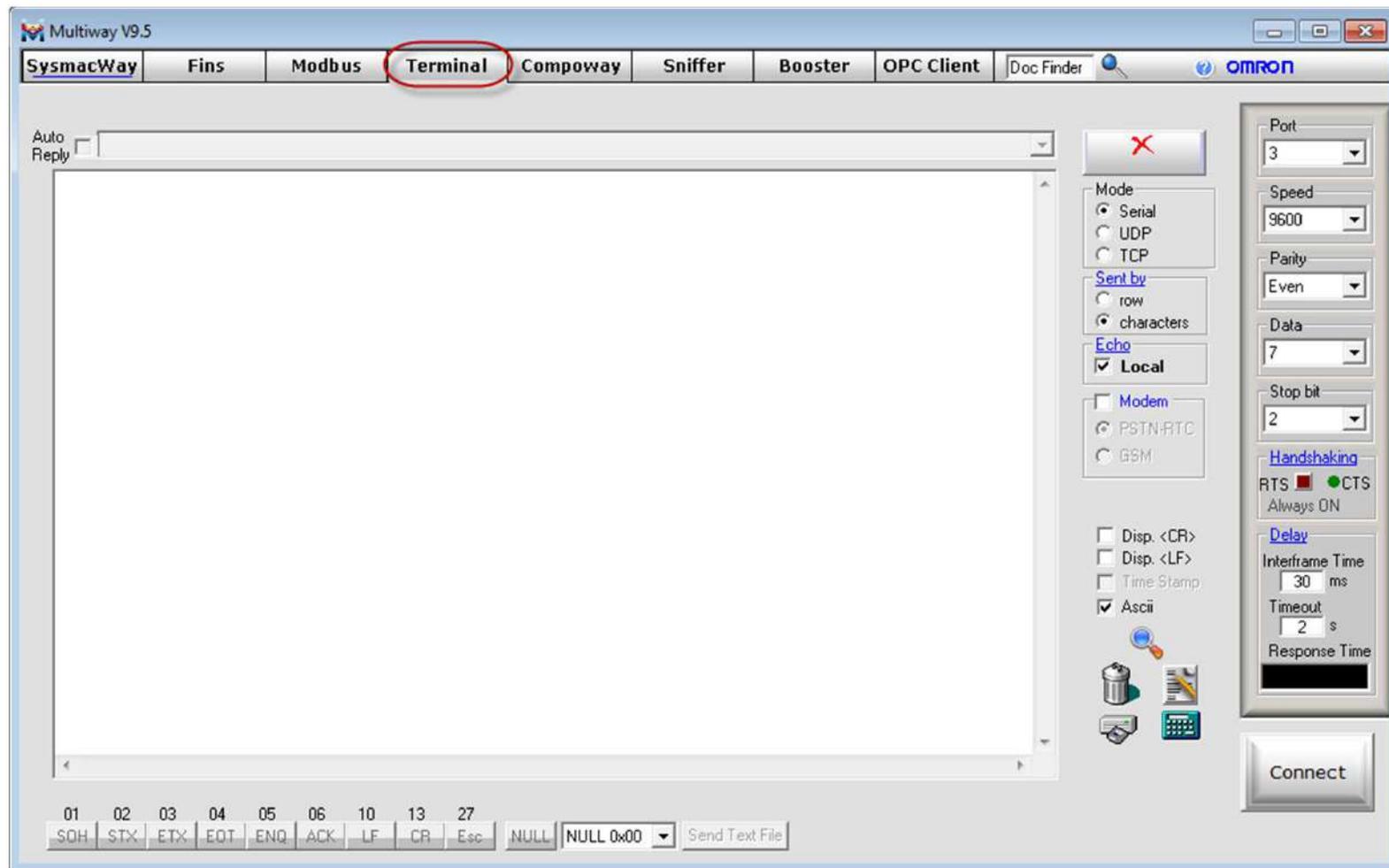
- Un utile strumento per verificare la corretta transizione dei dati sulla porta seriale del PLC, è costituito dall'**Hyper Terminal** presente in ambiente Windows (da Win'95 in poi).
- Esistono anche software 'free' per testare la comunicazione dei dispositivi, uno di questi per esempio è Multiway.



Il 'Terminal' di Multiway

- Sono supportate le seguenti funzionalità e protocolli:
 - ✓ Sysmac-Way serial;
 - ✓ FINS serial, Toolbus, Ethernet UDP and TCP;
 - ✓ Modbus RTU Master/Slave, TCP Client/Server;
 - ✓ ASCII Terminal and Modem functions;
 - ✓ Terminal for ASC02/11/21/31 Unit (up to V8.4);
 - ✓ Terminal for ASC02/11/21/31 Unit (up to V8.4);
 - ✓ Terminal for ASC02/11/21/31 Unit (up to V8.4);
 - ✓ Frame Booster (RS232C/Ethernet UDP);
 - ✓ OPC Client.

Il 'Terminal' di Multiway



Il 'Terminal' di Multiway

- E' possibile utilizzare il terminale per leggere i caratteri in ricezione sul PC (in trasmissione dal PLC) e scrivere le stringhe di caratteri da trasmettere sulla seriale (in ricezione sul PLC).

