

SISTEMI ELETTRONICI PER L'AUTOMAZIONE E L'INDUSTRIA (SEAI)

A.A. 2023-2024 INTRODUZIONE AL PLC



Negli anni 60-70 c'erano i PDP-11

• 16 bit, della DEC, che governavano i processi industriali ad alto livello

Alla fine degli anni 70 l'elettronica diventa a basso costo

- Prime console e home computer
- Primi sistemi per l'industria "a scatola chiusa" (centraline)
- Primi "PLC" (1968, Dick Morley, Modicon –oggi Schneider Electric-).
- Il termine PLC è stato inventato da Allen Bradley
- Per approfondire https://it.qaz.wiki/wiki/Programmable_logic_controller



PDP-11

Primi PLC



I PLC sono nati con la funzione di sostituire le operazioni logiche che venivano effettuate collegando in vario modo i relè.

Il primo linguaggio di programmazione dei PLC (ladder diagram) ricalca gli schemi elettrici a relè (linguaggio a contatti)

Problemi dei primi PLC (anni 70-80)

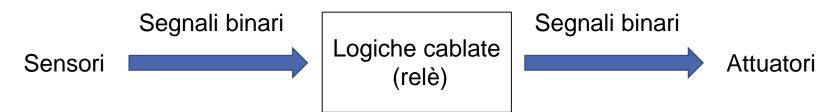
- Costo, Scarsa affidabilità
- Diffidenza degli addetti ai lavori (manutentori,...) verso un sistema "black box"

Rispetto alle logiche a relais:

- Riduzione dei cablaggi e degli ingombri (1 bit ⇔ 2 cm³) vs (1M bit ⇔ 2 cm³)
- Riduzione della potenza (1 bit ⇔ 0,1 W) vs (10 M bit ⇔ 10 W)
- Aumento della velocità di elaborazione (1 bit ⇔ ms) vs (1 bit ⇔ ns)
- Elevata versatilità verso "upgrade" (non solo bit)
- Consente nuovi campi d'impiego (regolazione e controllo, calcolo numerico,...)



Le logiche a relè sono dei "processori" di segnali binari



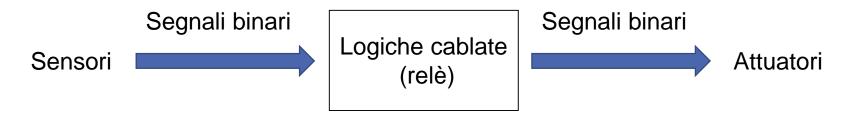
Perché il PLC? Perché non PC, centraline, oppure prendo una scheda con segnali di I/O (es. Raspi, Arduino,...) e la programmo?

II PC:

- Non acquisisce segnali (solo segnale audio –microfono-)
- Non genera segnali (solo segnale audio –altoparlanti-)
- Devo connettere a Ethernet/USB/WiFi/BLE un'interfaccia per segnali di I/O (Input/Output)
- E' complesso quindi considerato poco affidabile (aggiornamenti, O.S.,...)
- Soffre di obsolescenza digitale (lo devo cambiare ogni 5-6 anni, mentre gli impianti durano 20-30 anni)



Le logiche a relè sono dei "processori" di segnali binari



Perché il PLC? Perché non PC, centraline, oppure prendo una scheda con segnali di I/O (es. Raspi, Arduino,...) e la programmo?

Le centraline

• Non sono programmabili (modifico le istruzioni) ma solo configurabili (modifico i dati). Ad es. nel relè monitoraggio variabili analogiche posso configurare le soglie

"Prendo una scheda con segnali di I/O e la programmo da zero"

• Richiede un'elevata professionalità. Elevato costo del software

PLC: semplice, progettato come controllore industriale di segnali



https://new.siemens.com/it/it/prodotti/automazione.html https://new.siemens.com/it/it/prodotti/automazione/systems/industrial.html



PLC ad architettura modulare (es. Siemens S7-1500) CPU e moduli per segnali di ingresso o uscita, digitali o analogici Architettura molto flessibile che salva gli investimenti

- Aggiungo moduli all'esistente
- Cambio CPU, mantengo moduli
- Cambio moduli, mantengo il resto

Approfondimenti sui PLC

https://www.siemens.com/global/en/home/products/automation/systems/industrial/plc.html



PLC ad architettura compatta (es. Siemens S7-1200)
Posso aggiungere moduli opzione



- CONTROLLORI DISTRIBUITI
- PERIFERIA INTELLIGENTE PER LA FABBRICA

Controllore Distribuito

- E' una struttura modulare tipo PLC ma più compatto e fortemente connesso per lavorare con altri PLC in architetture master-slave

Periferia

- E' una struttura modulare tipo PLC ma più compatto e fortemente connesso per lavorare da slave verso un PLC in architetture master-slave
- -> L'hardware è lo stesso, cambia la CPU e il SW

Esempio: Siemens ET 200 SP

E' una struttura modulare per controllo distribuito o periferia Può fare elaborazione asservita ad un PLC (o PC) https://www.youtube.com/watch?v=o18WQfAocm4

https://cache.industry.siemens.com/dl/files/293/58649293/att_88880/v1/et200sp_system_manual_it-IT_it-IT.pdf

Facile utilizzo

- Moduli compatti, cablaggio permanente con connettore a uno o più fili
- Risparmio di tempo grazie alla tecnica di collegamento senza strumenti con morsetti push-in
- Adattamento della configurazione per ampliamenti futuri tramite controllo di configurazione integrato

- Dimensioni ridotte ed elevata variabilità grazie alla possibilità di cambio di scala
- Massima chiarezza nel minimo spazio grazie all'innovativo sistema di etichette
- Alimentazione della corrente di carico integrata nel sistema

Safety Integrated

- Semplice integrazione dei moduli fail-safe
- Impostazione di tutti i parametri F tramite software



- PROFINET IO
- PROFIBUS DP
- ET-Connection
- AS-Interface
- IQ-I ink
- Punto a punto (RS232, RS485)

Elevata performance

 PROFINET IO con sincronismo di clock con i profili PROFIsafe e PROFIenergy

Tecnologia avanzata

 Moduli per le funzioni di conteggio, posizionamento, pesatura e misura delle grandezze caratteristiche elettriche

Efficienza energetica

PROFlenergy come funzione integrata

CPL

- Interfaccia PROFINET con 3 porte
- IO Controller
- I Device
- Modulo CM DP opzionale per il collegamento a PROFIBUS DP





- PERIFERIA INTELLIGENTE PER IL CONTROLLO DI PROCESSO

Di facile applicazione

- Cablaggio semplice grazie all'assegnazione di colori alle connessioni
- Siglatura di tutte le interfacce secondo CAx
- Configurazione e messa in servizio pratiche e veloci con STEP 7 (TIA Portal)
- Adeguamento della configurazione per ampliamenti futuri tramite controllo diconfigurazione integrato

- Numero elevato di IO con larghezze dei moduli di 30 o 45 mm e fino a 32 moduli per ogni ET 200AL
- Grado di protezione elevato IP65/IP67
- Adatto per temperature da -30 °C/-25 °C fino a +55 °C e accelerazione fino a 10 g

Montaggio flessibile

- Facilità di montaggio in tutte le posizioni, anche negli spazi più ristretti
- Montaggio separato dei moduli in punti diversi attraverso un bus backplane realizzato come cavo(= ET Connection)
- Collegamento flessibile a PROFINET, PROFIBUS o integrazione in SIMATIC ET 200SP
- Integrazione di sensori e attuatori tramite tecnica di collegamento M8 e M12



Standard di comunicazione

- PROFINET IO
- PROFIBUS DP
- IO-Link
- PROFIsafe



Efficienza energetica

PROFlenergy come funzione integrata

Potente tecnologia

- Modulo con funzione di conteggio
- · Moduli fail-safe

https://www.youtube.com/watch?v=-eE6lja7ctY

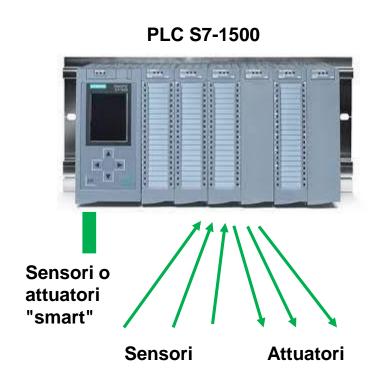


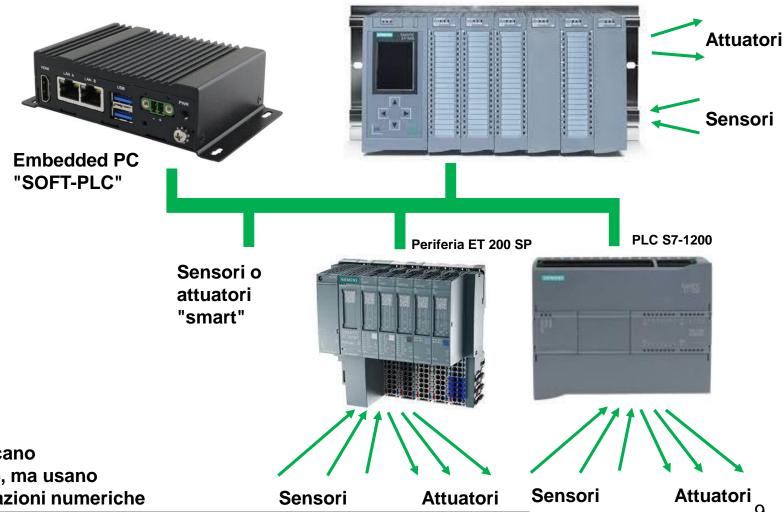
PLC: controllo centralizzato e controllo distribuito

CONTROLLO CENTRALIZZATO

CONTROLLO DISTRIBUITO

PLC S7-1500

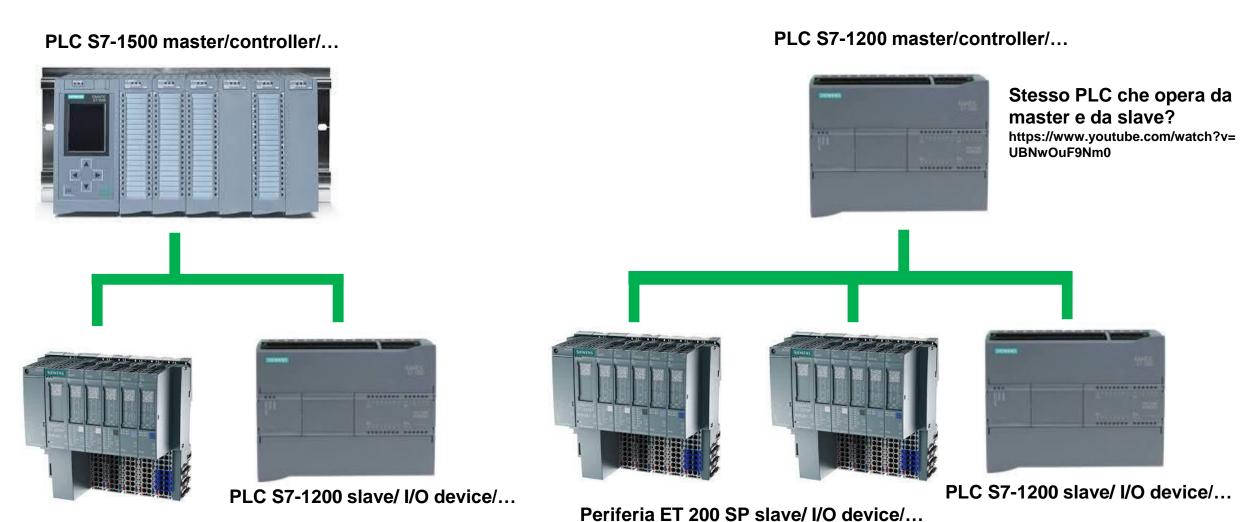




NOTA: i sensori e gli attuatori "smart" non codificano l'informazione nei segnali analogici (es. tensione), ma usano interfacce di comunicazione e scambiano informazioni numeriche



PLC: esempi di controllo distribuito



Periferia ET 200 SP slave/ I/O device/...



PLC: La comunicazione a livello di campo

Sistemi di comunicazione (es. USB, Ethernet, BLE, WIFI,...)

- Molto diffusi, ottimo rapporto costo/prestazioni
- Pensati per trasferire tanta informazione in poco tempo (es. file). Domina il concetto di "banda" ossia di numero di informazioni trasferite mediamente nell'unità di tempo (es. Gbit Ethernet = Ethernet che mediamente trasferisce un miliardo di bit in un secondo)
- L'utente dell'informazione è una persona o un data repository (se anche l'informazione arriva 100 ms in ritardo non è un problema)

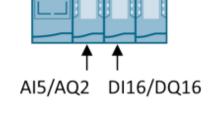
Sistemi di comunicazione tra PLC e periferia (es. Profibus, Profinet, I/O Link,...)

- Poco diffusi (il mercato dell'automazione è piccolo rispetto al mercato consumer), si poggiano sulle tecnologie consumer (es. Industrial Ethernet) per migliorare il rapporto costo/prestazioni
- Pensati per trasferire poca informazione con poco ritardo e spesso (es. Controllo PID: misuro, calcolo "errore", calcolo uscita, attuo e ricomincio). Domina il concetto di "latenza" ossia di tempo di ritardo tra la misura e l'attuazione (es. ritardo tra voce in microfono e ascolto su Whatsapp, tip. ~ 10-100 ms)
- L'utente dell'informazione è un controllore che deve agire in modo tempestivo (se l'informazione arriva 100 ms in ritardo l'attuazione potrebbe essere controproducente)

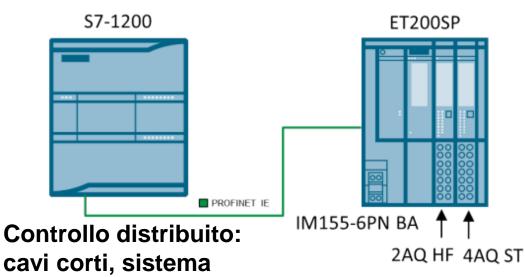
NOTA: il primo centro di competenza italiano accreditato su Profibus e Profinet è a Brescia https://www.csmt.it/it/profibus-profinet

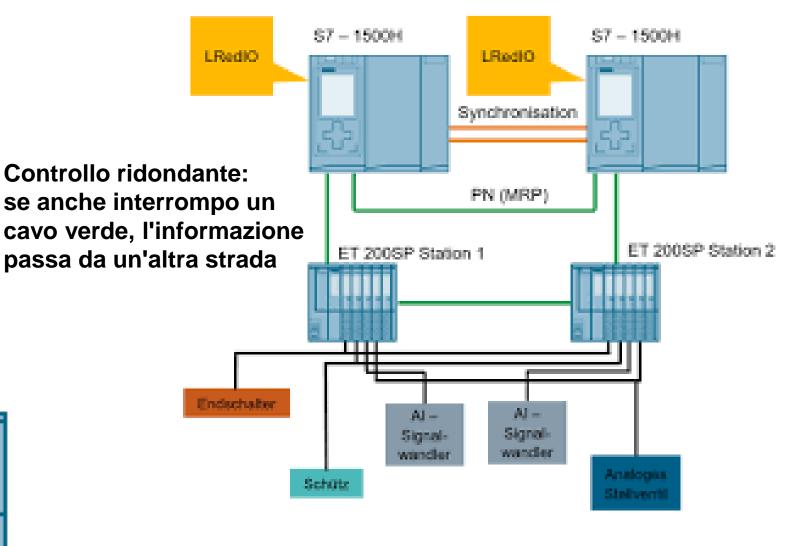






sezionabile





PLC e periferie, le nuove architetture https://www.youtube.com/watch?v=K25vXdNCKs0

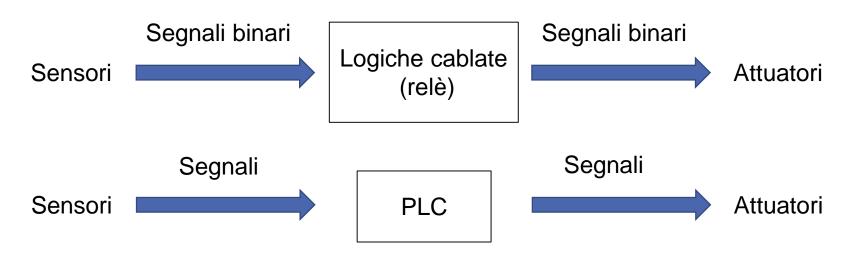


PLC: Software

Il PLC si basa su un microprocessore e un software di base che può essere assimilato ad uno schedulatore di processi

E' programmabile con linguaggi grafici o comunque orientati allo svolgimenti di sequenze temporizzate del tipo

- acquisisci ingressi (binari e numerici)
- elabora
- produci uscite (binarie e numeriche) e ricomincia





PLC: Software, la matrice interfaccia

Collegare un segnale d'ingresso (es. da pulsante) ad un relè era facile

· Individuo il relè, individuo il morsetto di quel relè, cablo i contatti

Se collego un segnale d'ingresso al settimo morsetto del terzo modulo come lo individuo nel programma?

 C'è una corrispondenza tra ogni morsetto, la tipologia di segnale e il nome della variabile (indirizzo "fisso" o "posizionale" o "geografico")

Tabella dei simboli o Tabella di I/O o matrice interfaccia (PLC = "Black box")

Modulo	# morsetto	Modo segnale	Tipo segnale	Indirizzo	Simbolico
CPU	1	Ingresso	Booleano	10.0	Finecorsa_FC1
CPU	2	Ingresso	Booleano	10.1	Finecorsa_FC2
CPU	3	Uscita	Booleano	Q0.0	Lampada
CPU	4	Uscita	Booleano	Q0.1	Motore
Modulo 1	1	Ingresso	Analogico	AW0	Temperatura_S1
Modulo 1	2	Ingresso	Analogico	AW2	Temperatura_S2



PLC: Software, le memorie dati

Oltre ad ingressi che vengono dai sensori e uscite che vanno agli attuatori, le logiche a relè utilizzano relè di appoggio (es. i relè "set"

e "reset" nelle logiche set-reset)

Nel PLC si utilizzano le memorie Merker

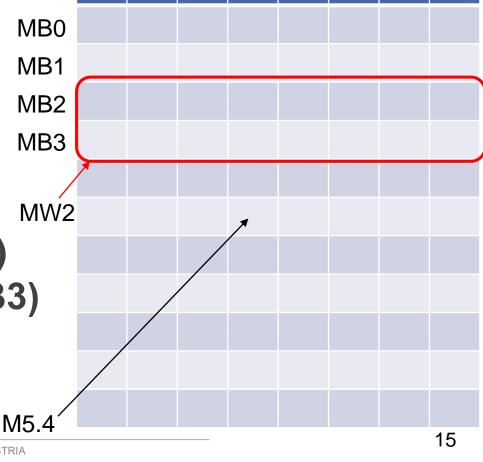
I merker sono organizzati (indirizzi) a byte

1 byte è fatto da 8 bit

1 word è fatta da 2 byte (MW0=MB0 e MB1)

1 double è fatta da 2 word (MD0=MB0...MB3)

I merker sono dichiarati nella tabella di I/O (Un PLC come il nostro S7-1215C ha 8kbyte)

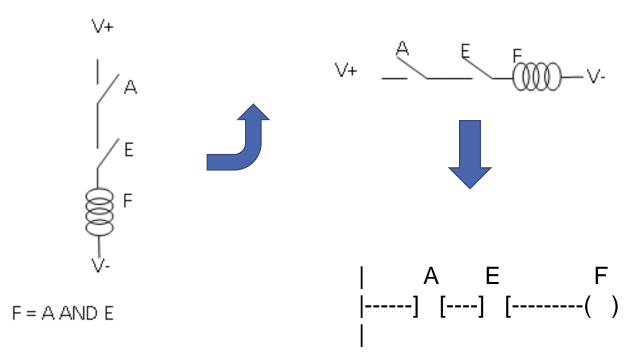


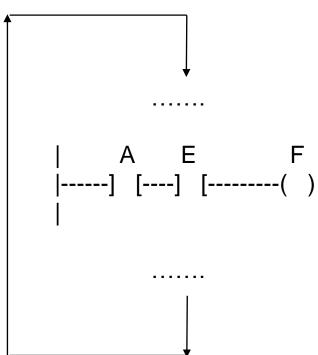


PLC: Software, le funzionalità

Funzionalità equivalenti alle reti di interruttori realizzate a relè

Prendo gli schemi a relè e li ruoto a sinistra di 90° -> ladder diagram





Ad un "filare" a relè corrisponde un segmento in ladder

- Barra verticale a sinistra implicitamente connessa a V+
- Bobine a destra implicitamente connesse a V- (corrente da sx a dx)

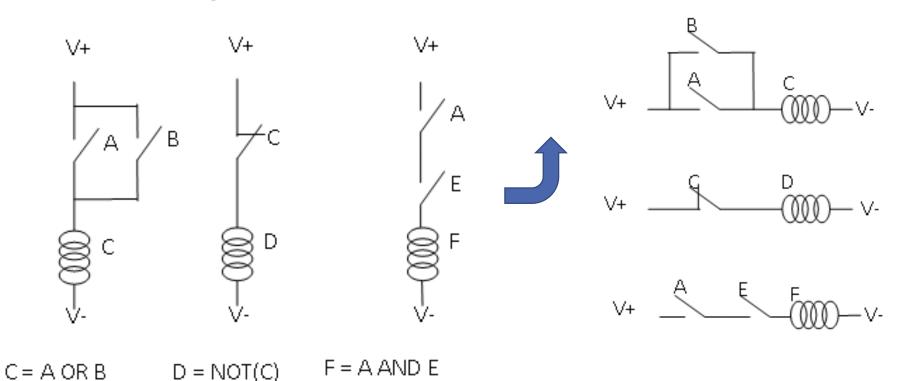
Il segmento in ladder è in esecuzione "continua", come accade per il relè, ossia la sequenza dei segmenti ladder si ripete



PLC: Software, le funzionalità

Funzionalità equivalenti alle reti di interruttori realizzate a relè

Prendo gli schemi a relè e li ruoto a sinistra di 90° -> ladder diagram



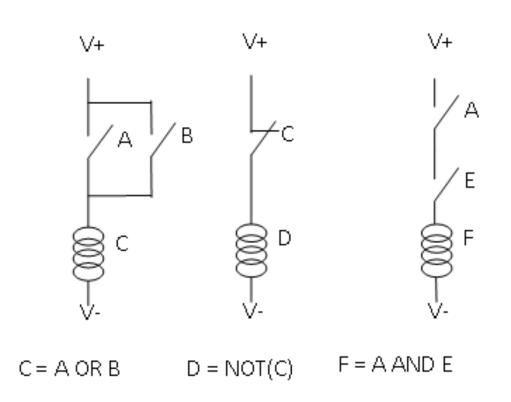
Programma

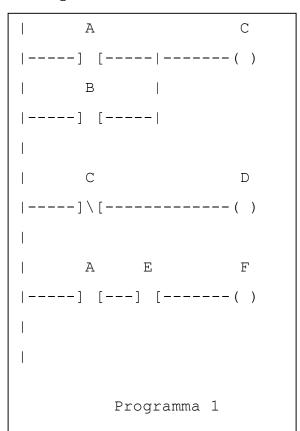
- L'esecuzione a relè è "lenta" (ritardo di ms tra bobina e contatti) ma parallela
- L'esecuzione del programma è veloce (ritardo di <ps tra un'istruzione e l'altra), ma è sequenziale, quindi l'ordine delle istruzioni può modificare le funzionalità

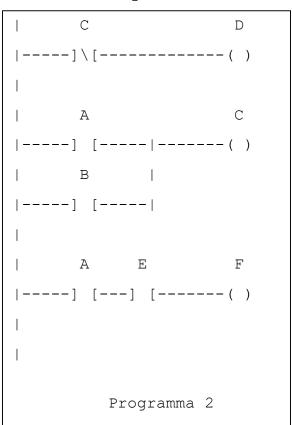


PLC: Software, le funzionalità

I segmenti a relè sono eseguiti in parallelo, i ladder in sequenza







Quale tra i programmi 1 e 2 genera segnali simili a quelli generati dagli schemi a relè? Si faccia il diagramma nel tempo dei segnali di ingresso e di uscita



PLC: Il ciclo di scansione

Ciascun ladder, come il "filare" a relè, realizza un'istruzione del tipo "se.. allora..." ossia "se f(ingressi) allora uscita=1 altrimenti uscita=0

La sequenza di ladder deve essere finita e l'esecuzione di tale sequenza finita si ripete all'infinito

Il tempo tra l'esecuzione di una sequenza e l'esecuzione della successiva deve essere "quasi nullo" per emulare il funzionamento in parallelo dei "filari" a relè (<10 ms)

Deve esserci un software di base che ciclicamente manda in esecuzione la sequenza di ladder



Ciclo di scansione



PLC: Il ciclo di scansione

Il ciclo di scansione controlla l'esecuzione sequenziale e ciclica del programma scritto in Ladder Diagram

Ciclo di scansione

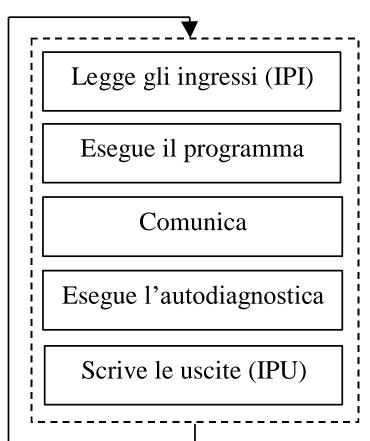
- T_ciclo = Tc ~ ms
- Tc dipende dalle istruzioni
- Tc cambia da ciclo a ciclo (asincrono)

Fase "comunica"

- diagnostica via PC
- fase di background a T controllato

Fase "autodiagnosi"

- diagnostica locale (CPU)
- diagnostica dei moduli di I/O





PLC: Le immagini di processo

Immagini di processo degli ingressi (IPI):

- variabili nelle quali viene memorizzato il valore degli ingressi logici all'inizio del ciclo (campionamento del ciclo k, una sorta di fotografia degli ingressi)
- il programma applicativo (ciclo k) si svolge a ingressi congelati
- Un test su un ingresso darà la stessa risposta ovunque nel programma

Immagini di processo delle uscite (IPU):

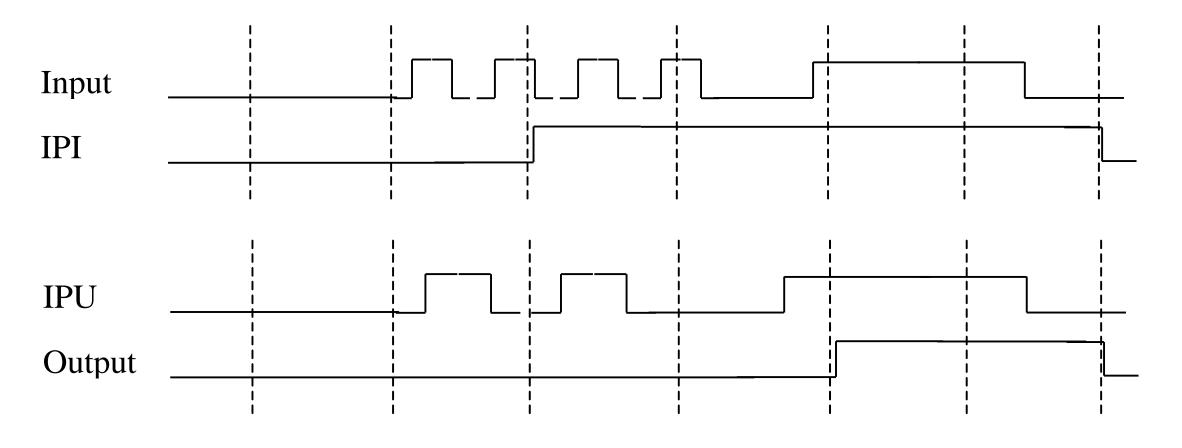
- variabili nelle quali il programma applicativo scrive come se fossero le uscite logiche ma che vengono effettivamente scaricate sulle uscite alla fine del ciclo
- l'ultima scrittura di un'uscita è la sola che ha effetto
- sincronizzazione delle uscite fisiche





PLC: Immagini di processo

Alcune variazioni degli ingressi possono andare perse (dipende da Tc) È consigliabile accedere alle uscite in un unico punto del programma





PLC: Modalità di funzionamento

Modalità STOP (/PROG):

- non esegue il programma applicativo di sua iniziativa
- esegue funzioni di dialogo con il sistema di programmazione (tip. PC)
- esegue funzioni di:
 - diagnostica
 - configurazione
 - programmazione del programma utente (download / upload)

Modalità RUN:

- esegue il programma applicativo sotto il controllo del ciclo di scansione
- effettua autodiagnostica
- se connesso al sistema di programmazione può:
 - visualizzare all'operatore lo stato di variabili
 - eseguire il programma applicativo solo per un certo numero di cicli
 - effettuare leggere modifiche al programma



PLC: Linguaggio Ladder Diagram

Dati

- Il Ladder supporta tutti i tipi di dati semplici (bit, byte, word, integer, real,...)
- Supporto dei dati relativi al tempo (Date&Time, Time)
- Array difficilmente supportati (solo indirizzamento immediato o diretto)

Istruzioni

- Codice = sequenza di istruzioni con lo stesso costrutto (tranne la bobina) if "condition" then "operation" (altrimenti no operation).
- Per la bobina invece vale
 if "condition" then bobina=1 else bobina=0
- "condition": confronti/interrogazioni (se var=4, se var1>var2, se A vero,...)
 "operation": set, reset, somma, assegna, inizia a contare il tempo, esegui un sottoprogramma,...



PLC: Linguaggio Ladder Diagram, dati

I dati come array di bit (operazioni logiche e ~aritmetiche)

- Byte: 8 bit (0x00-0xFF in notazione esadecimale)
- Word: 16 bit, concatenazione di 2 Byte (0x0000-0xFFFF)
- Double: 32 bit, concatenazione di 2 Word (0x00000000-0xFFFFFFFF)
- Long: 64 bit, concatenazione di 2 Double

I dati come Integer (operazioni aritmetiche)

- USint: Unsigned short integer, 8 bit (0...255 in notazione decimale)
- Uint: Unsigned integer, 16 bit (0...2¹⁶-1)
- UDint: Unsigned integer, 32 bit (0...2³²-1)
- Sint: Short integer, 8 bit (-128...+127) ossia (-2⁷...2⁷-1)
- Int: Integer, 16 bit (-2¹⁵...2¹⁵-1)
- Dint: Double Integer, 32 bit (-2³¹...2³¹-1)

I dati come Real (operazioni aritmetiche)

• Real: IEEE754 32 bit (-1)s-2e-1,m (esponente e USInt, mantissa m 23 bit) –Lreal a 64 bit



PLC: Linguaggio Ladder Diagram

Un programma scritto in ladder è una sequenza di istruzioni del tipo If "condition" then "operation" (else no operation)

- Se "operation" è uguale in due segmenti, allora le rispettive "conditions" sono in OR
 - If VAR1 then set(VAR5)
 - If VAR2 then set(VAR5)

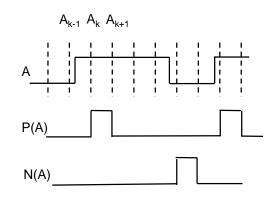
Equivale a

- If VAR1 or VAR2 then set(VAR5)
- Possono coesistere diverse operations sulla stessa uscita ma sarà solo l'ultima ad avere effetto in caso di conflitto
- L'operatore bobina non può coesistere con nessun altro operatore perché in ogni caso scrive sull'Object cancellando qualsiasi effetto di memoria (manca "else no operation")



PLC: Ladder diagram e operatori booleani (in)

- ---| --- Interrogazione vera, $A_k=1$? Se si passa corrente da sx a dx
- ---|\|--- Interrogazione negata, A_k=0? Se si passa corrente da sx a dx
- ---|P|--- Rilevatore di fronte positivo (A_k=1)&(A_{k-1}=0)?



NOTA: l'esecuzione del programma è ciclica e quindi avremo il ciclo k, il ciclo k+1, e così via, quindi A_k A_{k+1} NOTA: Mem serve per memorizzare il valore vecchio di $A = A_{k-1}$ e viene aggiornato con il valore di A_k alla fine dell'esecuzione di quel ladder



PLC: Ladder diagram e operatori booleani (out)

- ---()--- Assegna ad A "1" se arriva corrente alla bobina altrimenti assegna ad A "0"
- ---(s)--- Modifica A ad "1" se arriva corrente (altrimenti lascia A com'è)
- ---(r)--- Modifica A ad "0" se arriva corrente (altrimenti lascia A com'è)

NOTA: le logiche di Set-Reset sono molto facili

NOTA: ci sono altri operatori booleani che vedremo in laboratorio



PLC: Timer e Counter, il Timer TON

Timer

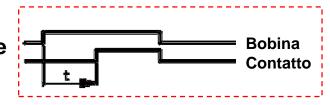
 Il timer non è un oggetto fisico, come i relè temporizzatori, ma un'area di memoria gestita dal software (come vedremo sono "blocchi software")

Timer TON (ritardo all'inserzione)

Equivalente al relè temporizzatore

Ritardo all'inserzione (Funzione 11 dei relè temporizzatori)
In corrispondenza del fronte di salita del segnale di ingresso (bobina), il temporizzatore inizia a contare a passato il tempo T attiva il contatto di uscita il contatto di uscita di

inizia a contare e, passato il tempo T, attiva il contatto di uscita. Il contatto di uscita si disattiva quando l'ingresso (bobina) va a zero



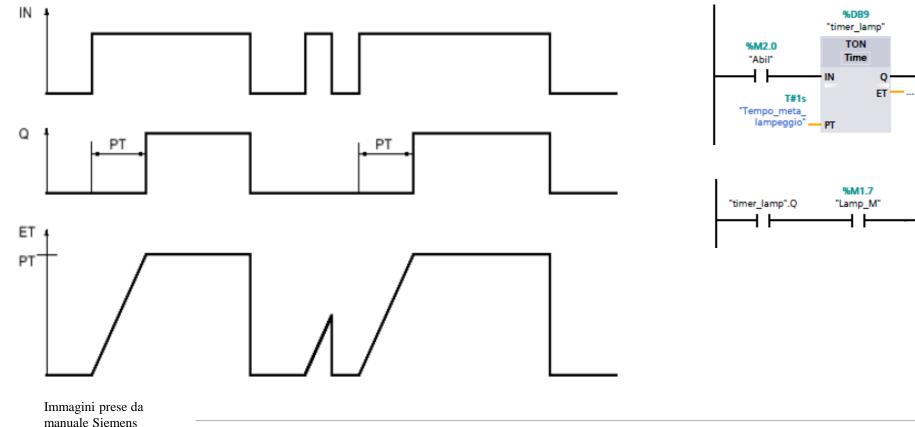
Area di memoria di un TON

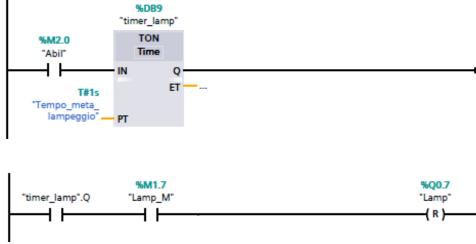
- IN, "Abilitazione" Ingresso Booleano (equivalente a Bobina)
- PT, "Costante di tempo" Ingresso di tipo Time (es. T#3s120ms pari a 3,12s)
- ET, "Valore corrente" Uscita di tipo Time
- Q, "Uscita" Booleana (equivalente a Contatto)



PLC: Timer e Counter, il Timer TON

Se IN ha un fronte di salita il TON inizia a contare e quando ET>PT allora Q si attiva. Se IN ha un fronte di discesa il TON si azzera (Esiste l'istruzione Reset Timer –RT-)

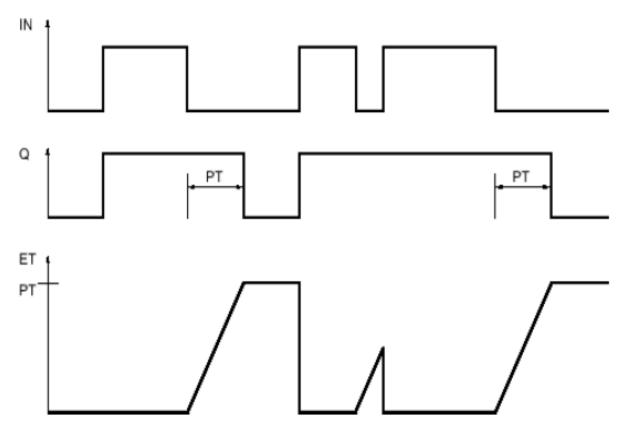




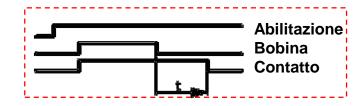


PLC: Timer e Counter, il Timer TOF

Se IN ha un fronte di salita allora Q va a 1; al fronte di discesa di IN il timer inizia a contare e quando ET>PT allora Q va a 0.



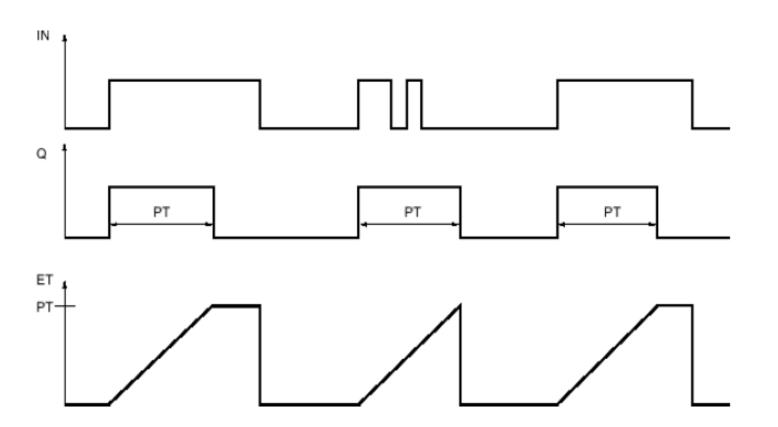
Ritardo alla disinserzione (Funzione 12)
In corrispondenza del fronte di salita del segnale di ingresso (bobina) si attiva il contatto di uscita; in corrispondenza del fronte di discesa di bobina, il temporizzatore inizia a contare e, passato il tempo T, disattiva il contatto di uscita. Si noti che l'uscita "estende" l'attività del segnale di ingresso e quindi esiste un ulteriore ingresso di abilitazione che permette tale estensione (Nel caso di Funzione 11 l'uscita "limita" l'attività del segnale)





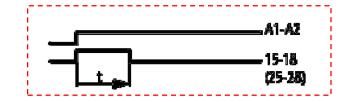
PLC: Timer e Counter, il Timer TP

Se IN ha un fronte di salita allora Q va a 1 e ci rimane per un tempo PT indipendentemente da IN, poi quando ET>PT allora Q va a 0.



Passante all'inserzione (Funzione 21) In corrispondenza del fronte di salita del segnale di ingresso (bobina), il contatto di uscita si attiva e il temporizzatore inizia a contare e, passato il tempo T, disattiva il contatto di uscita.

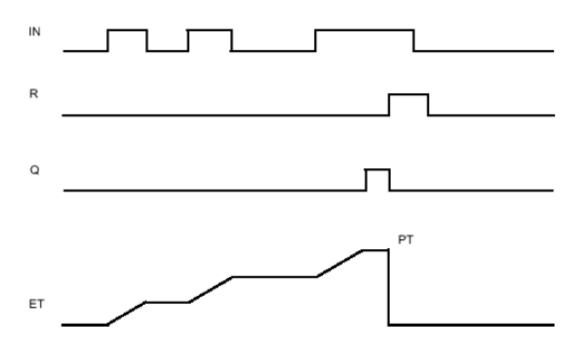
NOTA: TP non è uguale al relè temporizzatore che non è una funzione estensiva





PLC: Timer e Counter, il Timer TONR

Funziona come TON ma se IN ha un fronte di discesa il TON non si azzera ma si "congela" (Fondamentale l'istruzione Reset Timer –RT-)



Letteralmente, TONR sta per TON con "ritenuta" ossia con memoria, nel senso che quando viene a mancare l'abilitazione IN il Timer mantiene memorizzato il valore al quale è arrivato



PLC: Timer e Counter, il tipo Time

Una variabile di tipo Time equivale ad una Double (MD dichiarata in area merker) il cui contenuto viene interpretato come numero di millisecondi.

- La rappresentazione comprende le indicazioni di giorni (d), ore (h), minuti (m), secondi (s) e millisecondi (ms)
- TIME viene espresso con 32 bit (integer, unità ms) nel formato con segno da T#-24d20h31m23s648ms a T#+24d20h31m23s647ms
- Esempi di rappresentazione:
 - T#10d20h30m20s630ms,
 - TIME#10d20h30m20s630ms
 - 10d20h30m20s630ms
 - T#5h10s (Non è necessario indicare tutte le unità di tempo)
 - Non si devono eccedere i limiti (23 h, 59 m, 59 s o 999 ms)



PLC: Timer e Counter, Osservazioni sui Timer

I timer includono le funzionalità dei relè temporizzatori e permettono un uso "booleano" o un uso numerico (interrogando ET)

La costante di tempo PT può essere una variabile (Merker)

Con i timer TON e TONR si può fare tutto

Tipo	Il conteggio	Il conteggio	Il conteggio si	Out è a 1 se	Il timer si
Timer	inizia	si ferma	azzera (*)	timer attivo e	disattiva
TON	Fronte		Fronte discesa	Se ET >= PT	Fronte discesa
	salita In		In		In
TOF	Fronte		Fronte salita	Se ET < PT	Solo al reset
	discesa In		In		
TP	Fronte		Fronte discesa	Se ET < PT	Fronte discesa
	salita In		In & ET>=PT		In & ET>=PT
TONR	Fronte	Fronte	Solo al reset	Se ET >= PT	Solo al reset
	salita In	discesa In			



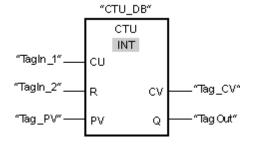
PLC: Timer e Counter, il Contatore

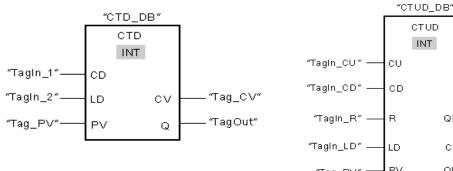
Counter

- Il timer "conta" il trascorrere del tempo, il counter conta eventi
- Il counter, come il timer, è un'area di memoria (sono "blocchi software")

• Esistono contatori a incremento (CTU), a decremento (CTD) e bidirezionali

(CTUD)





Counter CTU

Area di memoria di un CTU

- CU, "Count Up" Ingresso Booleano (il fronte di salita incrementa di 1 CV)
- R, "Reset" Ingresso Booleano (il valore 1 azzera CV inibendo CU)
- PV, "Costante di tempo" Ingresso di tipo Integer
- CV, "Valore corrente" Uscita di tipo Integer
- Q, "Uscita" Booleana (indica se CV>=PV)

"TagOut_QD"

"TagOut QU"

"Tag_CV"



PLC: Ambienti di sviluppo

Un ambiente di sviluppo permette di creare un progetto che comprende dispositivi (non solo PLC) e programmi (es. TIA Portal)

Se ci si limita ad un solo dispositivo (es. PLC S7-1215C DC/DC/DC) l'ambiente permette di:

- Configurare il dispositivo, ossia personalizzare l'hardware
 - Es. Modificare i filtri (che rimuovono i disturbi ma rallentano i segnali)
 - Es. comunicare al programma che c'è un modulo opzione connesso)
- Compilare la tabella dei simboli (v. slide "Matrice interfaccia")
- Scrivere il programma che sarà eseguito nel ciclo di scansione (OB1)
- Simulare il programma, impostando una serie di eventi
- Caricare il progetto nel PLC e mandarlo in esecuzione