

# Capitolo 1

## Programmable Logic Controller

In questo capitolo verranno descritti i PLC dal punto di vista hardware. Si prenderanno in considerazione le parti fondamentali che li compongono e i principali vantaggi e svantaggi derivanti dal loro impiego [1] [2] [3] [4].

### 1.1 Architettura dei PLC

Un controllore a logica programmabile è costituito da un'unità centrale e da un insieme di moduli periferici. Tali moduli sono montati su guide profilate e collegati attraverso appositi cavi o bus<sup>1</sup>. Generalmente un sistema PLC si compone di:

- **Modulo processore**
- **Moduli di input/output**
- **Modulo alimentatore**
- **Terminale di programmazione**

---

<sup>1</sup>Un bus è un insieme di linee elettriche, raggruppate per funzioni (linee indirizzo, linee dati e linee di controllo), e dei protocolli associati, attraverso i quali un dispositivo può comunicare con gli altri ai quali è collegato.

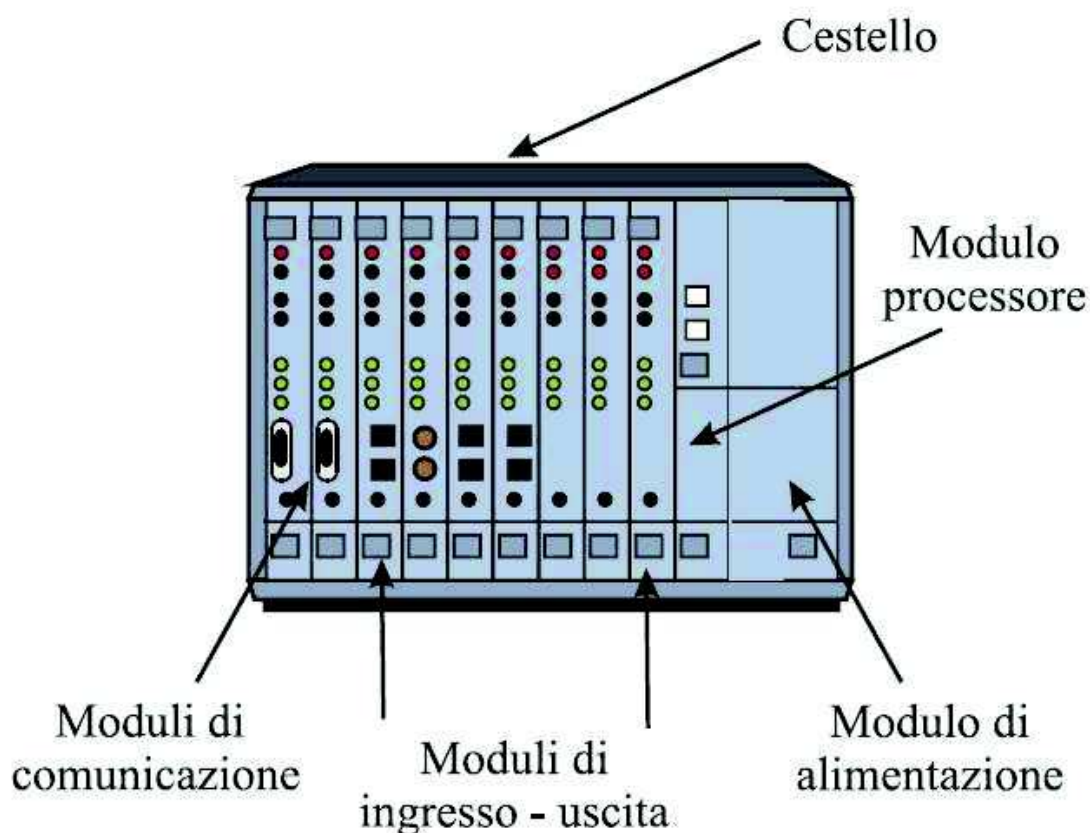


Figura 1.1: Architettura di un sistema PLC modulare[5].

A seconda dell'applicazione e delle specifiche richieste possono essere aggiunti al controllore altri moduli più specifici ( es. moduli di comunicazione, moduli di I/O remoto, moduli di interfaccia ecc.) che rendono più flessibile il controllore e ne facilitano l'impiego nelle applicazioni più svariate.

### 1.1.1 Modulo processore

La CPU (Central Processing Unit) è il cuore del PLC: racchiude una scheda con uno o più microprocessori che eseguono il sistema operativo e i programmi sviluppati dall'utente e una memoria dove questi programmi sono salvati. Il modulo dispone anche di una memoria RAM per i dati e di una batteria di backup, data la sua volatilità. La particolarità che distingue il funzionamento del PLC da quello

degli altri dispositivi a microprocessore è la ciclicità con cui il programma viene eseguito. Durante ogni ciclo il sistema esegue una sequenza di azioni:

- legge gli ingressi;
- esegue il programma utente;
- elabora le richieste di comunicazione;
- esegue l'autodiagnosi interna;
- scrive i valori sulle uscite;

L'unica eccezione a tale ciclo è rappresentata dalla gestione degli interrupt, di cui si parlerà in seguito.

Il modulo processore e tutti gli altri moduli che compongono il controllore sono predisposti per essere montati negli slot di apposite guide profilate che, oltre alla funzione di supporto, fungono anche da bus di comunicazione fra le varie schede. La CPU presenta inoltre molti connettori ai quali, a seconda dell'applicazione, possono essere collegati: moduli di memoria aggiuntivi, moduli di comunicazione con altri dispositivi e altri moduli speciali.



Figura 1.2: CPU Omron serie CS[6].

### 1.1.2 Moduli di input/output

In alcuni PLC di piccole dimensioni i terminali di ingresso e di uscita sono presenti nel modulo principale, mentre nella maggior parte dei casi essi si trovano su moduli esterni. I moduli di input/output sono i mezzi attraverso cui il PLC comunica con il processo fisico da controllare, rilevando dati dai sensori e comandando azioni agli attuatori. Le schede di I/O presenti sul mercato sono molteplici e con caratteristiche differenti a seconda del costruttore. Generalmente ci sono quattro, otto, sedici o trentadue terminali per ogni modulo con uno o più segnali di riferimento a seconda delle dimensioni e della tipologia del prodotto. Tuttavia, date le caratteristiche dei segnali presenti in un impianto industriale, tutti i costruttori forniscono moduli di I/O sia digitali che analogici con range di corrente e tensione standard.

Le schede di I/O digitali trattano segnali in tensione continua o alternata i cui range di tensione tipici sono 0-24V, 0-48V, 0-110V, 0-240V e sono inoltre dotate di circuiti di filtraggio contro i rumori. Oltre ai classici moduli di input che rilevano il livello del segnale in ingresso esistono anche moduli alimentati internamente che riconoscono la presenza di un contatto aperto o chiuso ai loro capi.

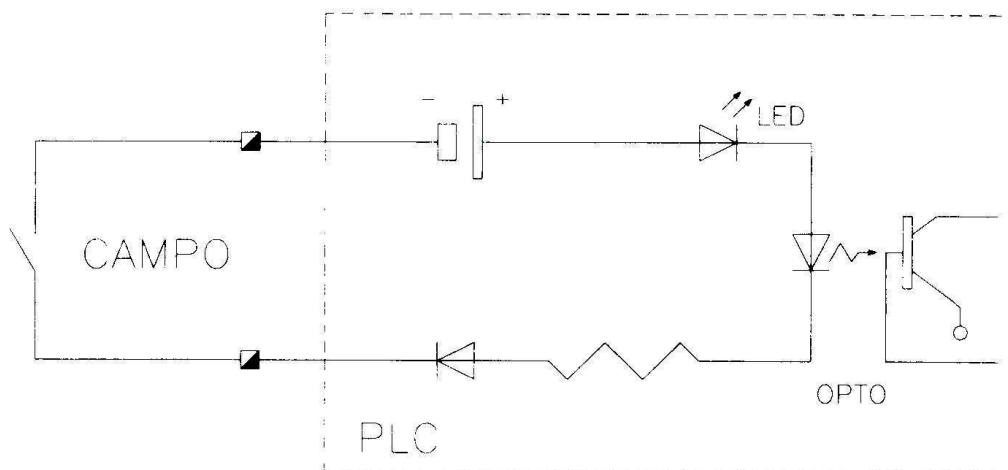


Figura 1.3: Schema elettrico di un modulo di input digitale optoisolato[1].

I moduli di I/O analogici trattano segnali in tensione o in corrente i cui range tipici sono  $\pm 10V$ ,  $4 \div 20mA$ ,  $\pm 10mA$ ,  $0 \div 20mA$ . Tali schede effettuano una conversione analogico/digitale del segnale e lo inviano alla CPU. Nella maggior parte dei casi i dispositivi di questo tipo effettuano l'acquisizione di più segnali analogici con un unico convertitore A/D utilizzando un multiplexer interno. Questa soluzione è vantaggiosa in termini di costo della scheda ma riduce necessariamente la velocità di acquisizione dei dati. Allo stesso modo nei moduli di uscita i valori numerici ricevuti dalla CPU vengono elaborati da un convertitore D/A ed eventualmente distribuiti da un multiplexer ai terminali.



Figura 1.4: Moduli di I/O digitali Omron serie CS[6].



Figura 1.5: Moduli di I/O analogici Omron serie CS[6].

Per applicazioni nelle quali i classici moduli di I/O non soddisfano le specifiche richieste esistono speciali moduli dotati di microprocessore che permettono di svolgere funzioni gravose per la CPU del PLC. Queste schede vengono per esempio usate per gestire cicli PID, per gestire contatori veloci oppure per applicazioni che necessitano di utilizzare algoritmi di conversione sui segnali provenienti dal campo.

Tutti i terminali di ingresso/uscita di questi dispositivi sono isolati galvanicamente<sup>2</sup> dall'elettronica interna del sistema, per evitare che tensioni o correnti troppo elevate la possano danneggiare. Nei moduli di I/O digitali per realizzare il disac-

<sup>2</sup>Con questo termine si definisce un collegamento tra due punti realizzato evitando una effettiva connessione elettrica.

coppiamento vengono utilizzati degli optoaccoppiatori mentre in quelli analogici l'isolamento galvanico è più complicato; esistono due possibilità: isolare direttamente il segnale analogico proveniente dal campo tramite trasformatori oppure agire sui segnali digitali dopo la conversione mediante optoaccoppiatori.

### 1.1.3 Modulo alimentatore

Il modulo alimentatore serve a fornire l'alimentazione elettrica stabilizzata necessaria al funzionamento di tutti gli altri moduli. Esso è composto da un trasformatore, da un circuito rettificatore, da un filtro, da un circuito stabilizzatore e da una protezione contro sovracorrenti o cortocircuiti. Tra le sue caratteristiche c'è la possibilità di inviare al PLC un segnale di shutdown se il livello dell'alimentazione fornita scende al di sotto di certi limiti. Il modulo processore in presenza di tale segnale può così avviare una procedura speciale prima dell'effettivo spegnimento in modo tale da mettere in sicurezza l'impianto. Se l'interruzione della tensione di rete è temporanea e non supera la decina di millisecondi il circuito di protezione non entra in funzione e l'alimentazione ai vari moduli è garantita dalla presenza di condensatori interni all'alimentatore.



Figura 1.6: Modulo alimentatore e rack Omron serie CS[6].

### 1.1.4 Terminale di programmazione

Il PLC non prevede tastiere e schermi quindi la sua programmazione avviene tramite dispositivi esterni. Esistono dei terminali dalle funzionalità limitate che, tramite un piccolo display e una tastiera, permettono di programmare direttamente la memoria del PLC. Generalmente, però, la programmazione avviene tramite un personal computer con software forniti dal costruttore del PLC che permette funzionalità di programmazione molto più complesse. Il PC può rimanere connesso ai PLC tramite una rete informatica permettendo all'utente di eseguire il monitoraggio del sistema, oltre alla possibilità di sviluppare e caricare direttamente i programmi.

### 1.1.5 Moduli speciali

Oltre ai moduli principali sopra citati, a seconda dell'applicazione per cui il PLC è impiegato, esistono una vasta gamma di moduli speciali che rendono ancora più flessibile il sistema. I principali sono:

- **Moduli di I/O remoto.** Vengono utilizzati quando i dispositivi controllati dal PLC sono molto lontani dalla CPU e con una connessione classica i segnali subirebbero un degrado troppo elevato. Ogni dispositivo controllato è dotato di un modulo remoto ed è connesso, tramite un bus, agli altri moduli e alla CPU. I dati trasmessi dal processore o dai moduli di I/O vengono codificati in modo da rispettare un determinato protocollo di trasmissione e poi decodificati dal modulo remoto del destinatario. Con questa soluzione, oltre al vantaggio di avere una maggiore qualità dei segnali, si ottiene anche una notevole riduzione del cablaggio fra il quadro principale della CPU e i vari dispositivi di controllo del processo. Un altro vantaggio non trascurabile è la riduzione dei tempi e dei costi di installazione e manutenzione data la maggior semplicità dei collegamenti che realizzano la rete.
- **Moduli per l'interfacciamento di termocoppie o di termoresistenze.** Questi moduli sono realizzati in modo da poter ricevere segnali direttamente da termocoppie o da termoresistenze i cui livelli di segnale sono molto

bassi. Il modulo realizza le necessarie operazioni di filtraggio, linearizzazione e amplificazione del segnale oltre che la sua conversione in digitale.

- **Moduli per la connessione in rete.** Sono moduli che gestiscono i protocolli di comunicazione per le varie tipologie di reti informatiche a cui il PLC può essere collegato (bus di campo, ethernet ecc)
- **Moduli PID.** Se la regolazione proporzionale-integrale-derivativa non è già presente nel modulo processore, con questi moduli è possibile avere a disposizione alcuni anelli PID a cui la CPU si limita a dare i riferimenti.
- **Moduli di posizionamento assi.** Questi dispositivi permettono di realizzare sistemi integrati per il controllo della posizione di più assi sincronizzati pilotando motori in AC, DC o passo-passo. L'azionamento dei vari motori è gestito da un programma interno al modulo al quale la CPU del PLC deve limitarsi a specificare posizione finale, accelerazione, decelerazione e velocità di regime. Generalmente questi moduli consentono di controllare in tempo reale la retroazione del motore asservito tramite la lettura di un trasduttore di posizione. Queste schede, di solito, sono dotate di memoria interna nella quale possono essere memorizzati parametri relativi alla movimentazione tramite la procedura dell'autoapprendimento<sup>3</sup>. Dispongono, inoltre, anche di contatori che consentono di effettuare sequenze ripetitive del programma.
- **Moduli di conteggio veloce.** Sono moduli che vengono utilizzati quando la frequenza dei segnali di ingresso assume valori rilevanti e gli ingressi normali non riescono a soddisfare le esigenze del caso poiché la scansione da parte del PLC risulta troppo lenta. Sono equipaggiati con un proprio processore e sono in grado di leggere segnali a frequenze che arrivano a centinaia di KHz. Le applicazioni tipiche per questi moduli sono: conteggio pezzi, misure di lunghezza o di posizione, misure di velocità, misure di durata.

---

<sup>3</sup>Ciascun asse viene movimentato manualmente fino al raggiungimento della posizione prefissata. A questo punto i parametri letti dal modulo tramite encoder vengono salvati in memoria per poter essere poi utilizzati nel programma finale.



- **Moduli interfaccia operatore.** Sono terminali costituiti essenzialmente da un display e da una tastiera che, collegati al modulo processore, consentono all'operatore di visualizzare messaggi, allarmi e di controllare o modificare le variabili del processo controllato. Il collegamento con la CPU avviene di solito per mezzo di una linea seriale (RS232, RS422, RS485). Durante la comunicazione il terminale si comporta da master mentre il modulo processore da slave; in questa configurazione il master interroga ciclicamente il PLC rispettando una temporizzazione fissa impostata dall'operatore (tempo di aggiornamento) andando a leggere e scrivere alcune locazioni di memoria ognuna delle quali ha un significato particolare. Attraverso questo scambio di dati il PLC può impostare i messaggi che devono essere letti dall'utente mentre il terminale visualizza e consente la modifica delle variabili di processo.

## 1.2 La memoria

La memoria del PLC è tipicamente costituita da dispositivi a semiconduttore di diverso genere a seconda della funzione che svolgono. In particolare essa può essere suddivisa in:

- **Memoria di sistema.** La memoria di sistema contiene tutte le informazioni che servono per la gestione ed il controllo del funzionamento della CPU, esse costituiscono il sistema operativo (firmware) del PLC. Data la primaria importanza di tale contenuto, i costruttori utilizzano generalmente memorie di sistema di tipo ROM in modo da evitarne l'accidentale cancellazione. Nulla vieta l'impiego di memorie PROM o EPROM purché queste rimangano inaccessibili all'utente.
- **Memoria di programma.** In questa memoria è contenuto il programma utente che il controllore deve eseguire. Ha una capacità variabile a seconda delle dimensioni e della tipologia del dispositivo ed è necessariamente di tipo riscrivibile. Come supporto fisico nei dispositivi più recenti si usa la

tecnologia FLASH in sostituzione delle memorie RAM, data la sua maggior affidabilità e non volatilità.

- **Memoria dati.** In questa sezione della memoria vengono memorizzati i risultati temporanei delle elaborazioni, i contenuti dei contatori, dei temporizzatori, dei bit interni che rappresentano lo stato logico degli ingressi e delle uscite. Tale memoria è necessariamente realizzata con memorie RAM o FLASH e, durante il funzionamento normale del controllore, è accessibile solo dal processore. Per favorire la messa a punto del software di controllo molti produttori danno la possibilità di attivare una modalità di lavoro nella quale all'utente è consentito di accedere alla memoria dati per simulare lo stato del sistema e verificare il corretto funzionamento del programma.

### 1.3 Confronto logica cablata/PLC

Alcuni fra i principali vantaggi nell'uso dei PLC sono riassunti nei seguenti punti.

- *Flessibilità:* prima dell'introduzione di questi dispositivi le macchine automatiche richiedevano l'utilizzo di sistemi di controllo realizzati con componenti elettromeccanici sulla base di uno schema elettrico specifico per ogni tipo di macchina. Con l'introduzione dei controllori programmabili il cablaggio dei sistemi di controllo risulta molto più semplice e lo stesso tipo di PLC, seppur con programmi differenti, può essere montato su macchine per mansioni molto differenti.
- *Facile gestione dei cambiamenti e correzione degli errori:* nella logica cablata per modificare il sistema di controllo è necessario ricablare tutto il circuito e cambiare i componenti che lo compongono. Quando viene effettuata una modifica ad un PLC generalmente non è necessario modificare lo schema elettrico, ma è sufficiente modificare il programma con un costo in termini di tempo assai minore.
- *Bassi costi:* con la diffusione di questi dispositivi il loro costo è sceso notevolmente. Questa diffusione, combinata allo sviluppo tecnologico che li

rende sempre più potenti e versatili, fa diventare i PLC molto vantaggiosi anche in termini economici.

- *Prova di esecuzione:* a differenza dei sistemi in logica cablata quelli programmabili possono essere facilmente testati prima di essere montati sul campo.
- *Velocità nelle operazioni.*
- *Facilità di riprogrammazione:* il PLC può rimanere collegato al terminale di programmazione ed essere riprogrammato velocemente a seconda delle necessità di produzione.

In alcuni casi particolari, però, l'introduzione della logica programmata risulta svantaggiosa.

- *Applicazioni fisse:* in alcune applicazioni che hanno un'unica funzione non è conveniente usare le numerose capacità di un PLC se queste non vengono sfruttate. Se le operazioni svolte dalla macchina non vengono modificate può risultare più economico un controllo elettromeccanico.
- *Condizioni ambientali:* alcune condizioni tipiche degli ambienti industriali quali le alte temperature, le vibrazioni, i disturbi elettromagnetici limitano l'utilizzo dei PLC perché più sensibili a questi fattori rispetto alla logica cablata.
- *Affidabilità e sicurezza:* mentre nei sistemi elettromeccanici un guasto porta generalmente all'interruzione del funzionamento, nelle apparecchiature programmabili esso può causare un funzionamento incontrollato. Per questo motivo le norme, ed in particolare la Norma EN 60204-1 all'art. 9.2.5.4, impongono che i comandi di arresto di emergenza siano realizzati solamente con dispositivi elettromeccanici cablati mediante contatti di interruzione.

## 1.4 Classificazione e mercato dei PLC

Visto il crescente impiego di questi dispositivi in applicazioni di automazione e visto lo sviluppo che quest'ultima sta avendo negli ultimi decenni, il mercato

dei PLC è in continua espansione e mette a disposizione dell'utente una gamma sempre più vasta di prodotti. La classificazione di questi controllori può essere fatta in base a diversi criteri.

Il primo criterio è quello che distingue i PLC secondo le loro dimensioni, cioè secondo il numero massimo di I/O e la quantità di memoria messi a disposizione dell'utente.

- **Micro PLC:** sono controllori monoblocco che gestiscono al massimo 64 ingressi/uscite, generalmente tutti di tipo digitale. La memoria messa a disposizione del programmatore va da 1Kbyte a 2Kbyte e il set di istruzioni disponibili comprende: quelle di base, quelle di temporizzazione e conteggio, oltre a quelle necessarie per operazioni aritmetiche.
- **Piccoli PLC:** hanno una struttura modulare e trattano da 64 a 512 punti di ingresso/uscita in prevalenza digitali. Per avere a disposizione una sezione di ingresso/uscita analogica è di solito necessario espandere il sistema con dei moduli analogici appositi. Possiedono una memoria che arriva generalmente fino a 4K e la capacità di connettersi in rete ad altri dispositivi o a moduli per la gestione di I/O remoto. Il set di istruzioni a disposizione del programmatore è più completo rispetto ai micro PLC.
- **Medi PLC:** possono arrivare a trattare da 256 a 2048 terminali di I/O e ad avere una memoria di qualche decina di Kbyte. Hanno una struttura modulare e sono espandibili, oltre che con i moduli base, con moduli speciali e moduli per la comunicazione su bus.
- **Grandi PLC:** sono caratterizzati dalla possibilità di poter gestire un numero molto elevato di ingressi/uscite, di solito qualche migliaio, e dalla disponibilità di centinaia di Kbyte di memoria. Hanno una potenza di calcolo molto superiore rispetto a quelli visti in precedenza e per questo vengono utilizzati come supervisori di impianti automatizzati e come interfacciamento tra PLC di minori dimensioni e calcolatori di gestione (esempio: PC che realizzano l'interfaccia utente per il controllo del processo).

I controllori programmabili si possono anche distinguere in base al criterio costruttivo adottato.

- **Monoblocco (o compatti):** sono quei dispositivi che vengono offerti in una configurazione rigida e compatta che non può essere modificata. In alcuni casi esiste la possibilità di aumentare il numero di ingressi/uscite collegando il PLC ad un altro blocco specifico uguale, sia nella dimensione che nelle prestazioni, all'unità di base. I dispositivi di questo tipo sono di solito di gamma bassa.
- **Modulari:** sono quei PLC configurabili dall'utente in base alle esigenze specifiche dell'applicazione. I vari moduli che compongono il sistema vengono montati su un rack e collegati tramite cavi appositi o bus. Tipicamente tutti i PLC di taglia media o alta sono modulari.



Figura 1.7: PLC monoblocco Siemens Logo[7].



Figura 1.8: Sistema PLC modulare Siemens S7-300[7].

Un'ultima classificazione dei controllori può essere fatta in base all'impiego per cui sono progettati.

- **Sequenziali:** sono i PLC di prima generazione, nati per sostituire i quadri elettromeccanici e sono impiegati nella realizzazione di automatismi che funzionano secondo una logica sequenziale. Ovviamente, rispetto ai primi, gli attuali PLC sequenziali mettono a disposizione altre funzionalità oltre a

quelle classiche della logica sequenziale: calcoli matematici, elaborazione di segnali digitalizzati, conteggio veloce<sup>4</sup>, ecc.

Solitamente si presentano come PLC compatti o modulari di taglia media o piccola.

- **Multifunzione:** sono dei dispositivi con una potenza di calcolo maggiore e mettono a disposizione dell'utente un maggior numero di funzionalità. Vengono utilizzati nelle applicazioni in cui sono richieste, oltre alle funzioni caratteristiche della logica sequenziale, le seguenti prestazioni: misura, regolazione PID, controllo assi, dialogo tra PLC e periferiche, comunicazione tra PLC.

### 1.4.1 Mercato

Le principali aziende produttrici di controllori a logica programmabile presenti sul mercato italiano sono:

- Siemens
- Allen-Bradley
- Omron
- Telemécanique
- GE Fanuc
- Toshiba
- Mitsubishi
- Matsushita
- Bosch
- Hitachi

---

<sup>4</sup>La funzione conteggio veloce è quella che consente al PLC di ricevere segnali che hanno un'elevata frequenza come, ad esempio, quelli provenienti da un rilevatore di posizione angolare (encoder)

# Capitolo 2

## Funzionamento

### 2.1 Ciclo di funzionamento

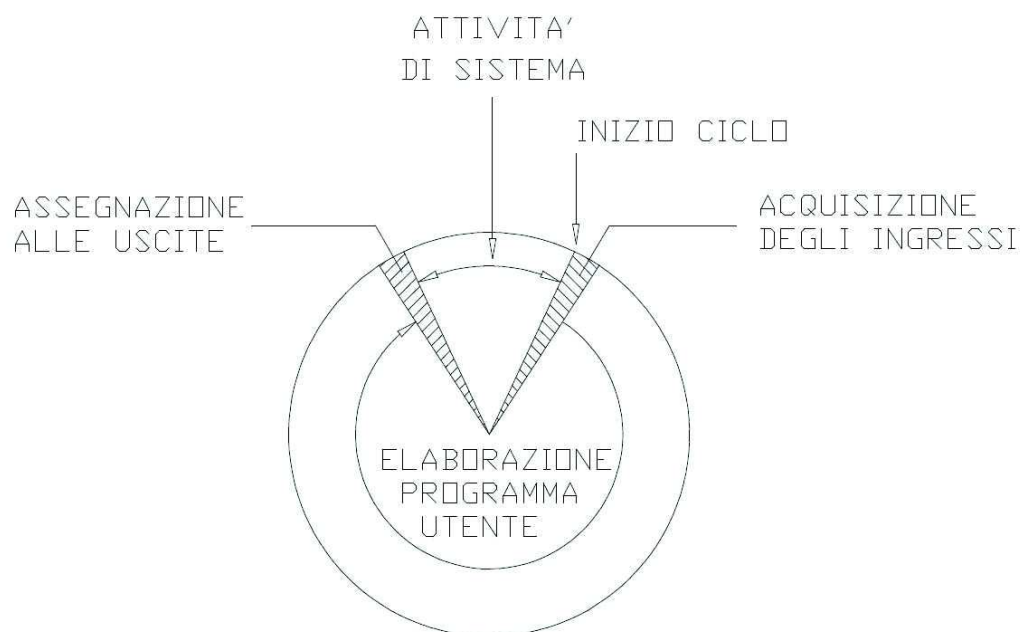


Figura 2.1: Fasi del ciclo di funzionamento di un PLC.

Come detto in precedenza, la caratteristica fondamentale dei PLC è il loro funzionamento ciclico. Le modalità con le quali vengono gestiti gli ingressi e le uscite possono essere diverse a seconda del particolare prodotto o applicazione e identificano i differenti tipi di cicli macchina[2].

- **Ciclo sincrono in ingresso e sincrono in uscita:** gli ingressi vengono acquisiti tutti all'inizio del ciclo e vengono memorizzati in un'area di memoria denominata immagine degli ingressi. Tale immagine rimane invariata per tutta la durata del ciclo e aggiornata solamente al ciclo successivo. Allo stesso modo, durante l'esecuzione del programma, le uscite vengono memorizzate nell'immagine delle uscite e solo alla fine vengono effettivamente inviate sul campo. Questo tipo di ciclo ha lo svantaggio di introdurre dei ritardi nell'acquisizione degli ingressi e nel comando delle uscite.



Figura 2.2: Fasi di un ciclo sincrono in ingresso e sincrono in uscita.



- **Ciclo asincrono in ingresso e sincrono in uscita:** a differenza del caso precedente non esiste l'immagine degli ingressi. Durante l'esecuzione del programma, quando si fa riferimento ad un ingresso questo viene letto direttamente dal campo. Per questo motivo più letture dello stesso ingresso, durante lo stesso ciclo di programma, possono dare valori diversi a seconda dell'istante in cui vengono acquisiti.
- **Ciclo sincrono in ingresso e asincrono in uscita:** durante l'esecuzione del programma le uscite non vengono memorizzate in un'immagine ma vengono inviate direttamente agli attuatori. La stessa uscita può quindi cambiare valore all'interno di un ciclo macchina se comandata più volte.

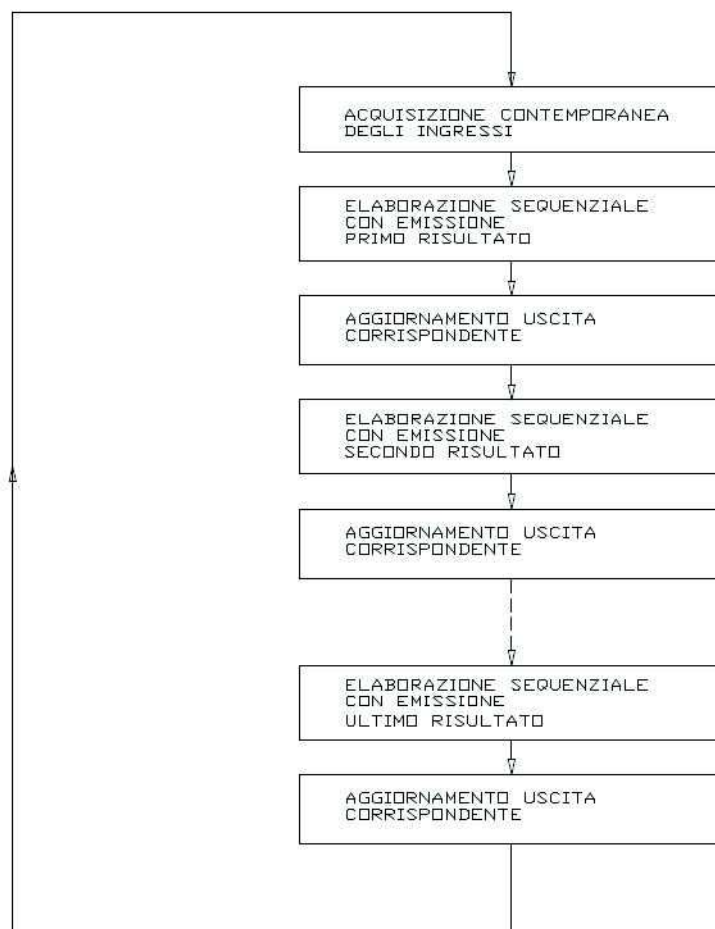


Figura 2.3: Fasi di un ciclo sincrono in ingresso e asincrono in uscita.

- **Ciclo asincrono in ingresso e asincrono in uscita:** non esistono immagini di ingresso né di uscita quindi i tempi di esecuzione sono più bassi rispetto a tutti gli altri tipi di ciclo.

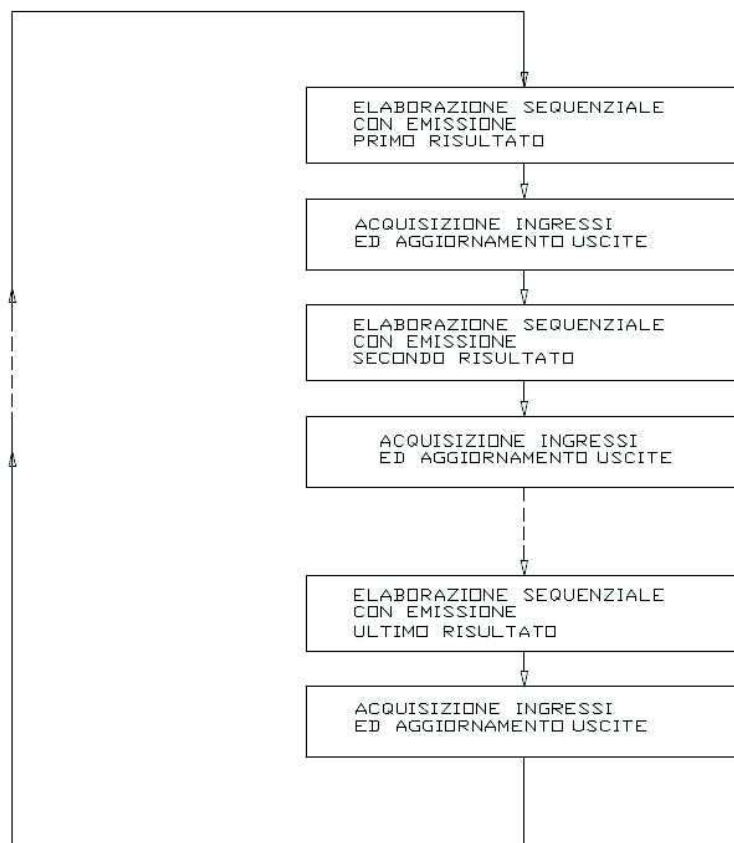


Figura 2.4: Fasi di un ciclo asincrono in ingresso e asincrono in uscita.

Gli ingressi e le uscite dei PLC sono memory mapped<sup>1</sup>, quindi il tempo impiegato per la lettura di un ingresso o del relativo valore salvato nell'immagine degli ingressi è lo stesso. Ciò che differenzia i tempi di esecuzione fra cicli sincroni e asincroni rendendo più veloci i secondi è la creazione ed elaborazione dell'immagine degli ingressi e delle uscite.

---

<sup>1</sup>Il dispositivo usa lo stesso bus per indirizzare sia la memoria che i dispositivi di I/O. Inoltre le stesse istruzioni della CPU utilizzate per leggere e scrivere la memoria sono utilizzate anche per accedere ai dispositivi di ingresso/uscita.

### 2.1.1 Tempo di ciclo

Il tempo che intercorre tra due esecuzioni successive della stessa parte di programma è chiamato tempo di scansione o tempo di ciclo e costituisce un parametro molto importante nel funzionamento del PLC. Il tempo di scansione dà un'indicazione dei tempi necessari per effettuare il ciclo sopra descritto ed è quindi legato al numero di ingressi/uscite da aggiornare, dalla lunghezza e complessità del programma da eseguire e per questo non assume un valore costante. A seconda del valore assunto da determinate variabili il programma può procedere in alcuni rami piuttosto che in altri, accorciando o allungando il ciclo rispetto alle iterazioni precedenti. Per questo motivo i costruttori danno come parametro di confronto fra PLC il tempo di esecuzione di un numero fisso di istruzioni (1Kword, 10Kword), ma anche in questo caso il paragone è difficile perché dipende dal tipo di istruzioni. Tempi di ciclo tipici per programmi di queste lunghezze sono comunque compresi tra qualche millisecondo e qualche decina di millisecondi a seconda della complessità delle istruzioni da eseguire. Il tempo di ciclo dà inoltre un'indicazione sulla banda passante massima dei segnali gestiti dal PLC poiché nel caso peggiore (funzionamento sincrono) i segnali di ingresso vengono acquisiti una volta per ogni ciclo macchina. Si hanno quindi bande passanti di qualche decina di Hertz che possono sembrare molto piccole, ma che generalmente sono sufficienti in ambito industriale. Per applicazioni dove il PLC deve essere in grado di gestire sensori più veloci si impiegano moduli speciali dotati di processore proprio.

### 2.1.2 Tempo di risposta

Il tempo di risposta del controllore programmabile è definito come l'intervallo di tempo massimo che intercorre tra il verificarsi di un evento in ingresso al dispositivo e l'istante in cui avviene la reazione. Nel caso peggiore l'evento si verifica subito dopo che il controllore ha costruito l'immagine degli ingressi. In questo caso saranno necessari un ciclo di programma per aggiornare l'immagine e un altro ciclo perché l'effetto venga inviato sul campo. Si può quindi dire che il tempo di reazione del dispositivo nel caso peggiore è di due cicli macchina.

## 2.2 Modi di funzionamento

Tutte le CPU hanno degli switch, diversi a seconda del costruttore, che permettono di selezionare la modalità di funzionamento del dispositivo. Tipici switch sono:

- Off: il sistema non può essere avviato o programmato.
- Run: il sistema può lavorare ma il programma non può essere modificato.
- Run/Program: il programma può lavorare e può essere modificato in esecuzione.
- Program/Stop: in questa modalità è possibile modificare il programma ma il sistema non può lavorare.
- Reset: tenendo per alcuni istanti lo switch in questa posizione è possibile resettare il programma utente, cioè farlo ripartire azzerando tutti i dati elaborati in precedenza. In alcune CPU, tenendo lo switch nella stessa posizione per un tempo ancora maggiore, è possibile resettare la memoria del dispositivi cancellando il programma utente.

Se il PLC è collegato in rete ad un PC è possibile modificare il modo di funzionamento del dispositivo direttamente via software senza agire sugli switch. Questo dà all'operatore un pieno controllo sul PLC senza la necessità di agire sul campo.

## 2.3 Sistema operativo

Il sistema operativo (SO) del PLC è costituito da un insieme di programmi di base per la gestione del programma utente, l'acquisizione degli ingressi e l'attuazione delle uscite, la diagnostica del sistema[2]. La sua particolarità, che lo contraddistingue dai sistemi operativi comuni, è quella di funzionare in tempo reale: dà cioè la possibilità di specificare a priori dei valori massimi (sufficientemente brevi) dei tempi di reazione del controllore programmabile. Tale operazione di controllo del tempo di ciclo è detta *Watchdog timer*. Nel caso in cui il timer, precaricato ad

ogni ciclo al valore massimo stabilito, si annulli, il dispositivo attiva un opportuno modulo di risposta all'errore di programma. Per fare ciò il programma utente, che contiene il modulo di emergenza, riceve dal sistema operativo la segnalazione di errore tramite dei bit riservati in memoria. Tale funzione è necessaria perché lo scadere del *Watchdog timer* indica un'anomalia<sup>2</sup> che, se ignorata, pregiudicherebbe il corretto funzionamento del sistema di controllo.

Il sistema operativo di un PLC presenta caratteristiche che variano a seconda del dispositivo e quindi del costruttore, ma alcune di queste devono essere presenti in tutti i controllori programmabili.

- Controllo del ciclo di funzionamento.
- Gestione della sezione di ingresso/uscita.
- Gestione degli interrupt.
- Funzioni diagnostiche.
- Funzioni ausiliarie.

### 2.3.1 Controllo del ciclo di funzionamento

Questa funzione, come dice il nome stesso, ha il compito di garantire il corretto funzionamento del ciclo macchina del PLC. Il programma utente è normalmente costituito da un insieme di blocchi (routine) che vengono eseguiti su chiamata a seconda dell'evoluzione del programma stesso. Tra questi blocchi ne esiste uno principale che viene eseguito all'inizio di ogni ciclo macchina. All'interno di questo blocco principale sono presenti chiamate ad altri blocchi i quali, a loro volta, ne possono chiamare degli altri. Quando questi blocchi vengono trasferiti dal modulo di programmazione al PLC è compito del SO collocarli nella memoria in un'area specifica e costruire una tabella di corrispondenza all'interno della quale ad ogni blocco è associato l'indirizzo a cui si trova in memoria. Se durante l'esecuzione del codice si verifica una chiamata ad uno di questi blocchi il sistema operativo

---

<sup>2</sup>Generalmente si tratta di un loop infinito che blocca il programma utente per una condizione di uscita dal ciclo errata.

esegue automaticamente una sua specifica routine la quale accede alla tabella e scrive nel program counter (PC) <sup>3</sup> l'indirizzo della prima istruzione del blocco da eseguire. Tale routine ha inoltre il compito di salvare l'indirizzo dell'istruzione successiva a quella di salto per permettere, terminata l'esecuzione del blocco, la corretta ripresa del flusso di programma.

La funzione di controllo del ciclo di funzionamento ha anche il compito all'inizio ed alla fine di ogni ciclo di caricare nel PC l'istruzione successiva che il microprocessore deve eseguire.

Come ultima mansione questa funzione monitora continuamente il tempo di ciclo bloccando il programma utente (si dice che il PLC entra in STOP) se tale tempo supera le soglie prestabilite.

### 2.3.2 Gestione della sezione di ingresso/uscita

I riferimenti agli ingressi/uscite del PLC nel programma utente utilizzano una notazione simbolica che non specifica il loro indirizzo assoluto in memoria (E3.0 rappresenta, ad esempio, il bit 0 del byte dell'ingresso 3, mentre, sempre a titolo di esempio, la notazione simbolica *pump-on* per attivare un'uscita è ancora più astratta)[2] e che deve essere interpretata in modo da consentire l'accesso alle locazioni di memoria nelle quali l'I/O è mappato. Questo compito è svolto dal SO del dispositivo ed in particolare da due delle sue routine. Indipendentemente dal tipo di ciclo, quando è necessaria una lettura degli ingressi il sistema operativo chiama una routine denominata *i\_read*, mentre per la scrittura delle uscite ne utilizza un'altra denominata *q\_write*. Queste routine, invocate automaticamente quando nel programma utente si fa riferimento agli ingressi/uscite, hanno la funzione di interpreti che, tramite l'utilizzo di una tabella di corrispondenza, associano ad ogni byte dell'I/O il suo indirizzo nella memoria del dispositivo.

Generalmente la mappatura nei relativi indirizzi di memoria è fissa, ma non sempre tutti gli ingressi/uscite sono presenti nella configurazione del sistema. Per verificare tale configurazione esiste una terza routine denominata *config* che viene eseguita automaticamente ad ogni avvio del PLC e che ha il compito di verificare

---

<sup>3</sup>Registro presente nel microprocessore che contiene sempre l'indirizzo dell'istruzione successiva da eseguire.

gli ingressi e le uscite presenti e di completare la tabella di corrispondenza con un'indicazione sulla loro presenza/assenza.

Come ultima funzione di gestione dell'I/O il sistema operativo crea le immagini di ingresso e di uscita nei cicli di tipo sincrono. Per fare ciò sfrutta una routine denominata *image* che, tramite le chiamate ad *i\_read* e a *q\_write*, salva in una specifica zona di memoria gli ingressi e le uscite ad ogni ciclo come spiegato nel funzionamento dei cicli di tipo sincrono.

### 2.3.3 Gestione degli interrupt

La gestione degli interrupt da parte dei PLC è molto simile a quella dei comuni dispositivi a microprocessore[2]. Quando si verifica un interrupt l'esecuzione del programma principale viene interrotta, viene salvato l'indirizzo dell'istruzione successiva e viene eseguita la routine di interrupt associata alla particolare causa che lo ha provocato. Al termine dell'esecuzione della routine il processore ritorna ad eseguire il programma dal punto in cui lo aveva lasciato in sospeso. Nei PLC queste particolari routine sono dei blocchi di programma associati in modo statico dal SO alle diverse cause di interrupt: quando una di esse si verifica il ciclo di programma viene sospeso e il blocco associato, adeguatamente programmato dall'utente, viene eseguito riprendendo successivamente l'esecuzione del programma al termine dello stesso. Nel caso in cui si verifichi una situazione di interrupt il cui blocco associato non è stato programmato dall'utente il PLC continua l'esecuzione del ciclo con conseguenti rischi per il corretto funzionamento del sistema. Generalmente il costruttore dà la possibilità di inserire all'interno del codice delle istruzioni per abilitare e disabilitare le funzioni di interrupt nel corso del ciclo macchina. Queste istruzioni vengono utilizzate principalmente nel caso in cui la mansione svolta in quel momento dal controllore non possa essere interrotta per un determinato intervallo di tempo.

Generalmente un PLC prevede tre diversi tipi di interrupt:

- *Interrupt hardware*. Provengono direttamente dall'impianto e coincidono con il verificarsi di situazioni di allarme. Per gestire tali interruzioni è sufficiente collegare i segnali di allarme agli appositi ingressi messi a disposizio-

ne nei moduli di I/O e programmare i corrispondenti moduli associati. Al verificarsi della situazione di allarme il sistema operativo effettuerà automaticamente la chiamata alla routine in questione e il ritorno all'esecuzione del programma principale.

- *Interrupt software.* Provengono da elaborazioni interne del programma utente quando si verificano determinate situazioni. Perché sia lanciata la routine di interrupt è necessario che il programma utente effettui delle apposite operazioni di scrittura su dei registri di memoria riservati dal sistema operativo proprio a tale scopo.
- *Interrupt temporizzati.* Questo tipo di interruzioni sono specifiche dei controllori programmabili e non si riscontrano comunemente negli altri dispositivi a microprocessore. Si tratta di particolari routine che vengono eseguite in modo periodico all'interno del ciclo macchina senza cause dirette. Generalmente il periodo di attivazione di tali blocchi non può essere deciso dall'utente, ma è il costruttore che mette a disposizione blocchi che, se programmati, si attivano ad intervalli prefissati (valori tipici sono 10ms 100ms 1s ecc.) Nei PLC di ultima generazione è disponibile un'ulteriore tipologia di interrupt temporizzati: si tratta di specifiche routine che vengono attivate ad un istante specificata con data e ora, o con un ritardo prestabilito rispetto all'istante specificato<sup>4</sup>.

### 2.3.4 Funzioni diagnostiche

Durante il funzionamento del sistema si possono verificare delle anomalie generalmente dovute ad errori di programmazione o a problemi hardware. Tipici esempi di tali malfunzionamenti sono: il richiamo di un blocco di programma non presente in memoria, il tentativo accesso a variabili inesistenti, un guasto all'alimentazione, il cortocircuito in un modulo di uscita ecc. Il sistema operativo, come nella gestione degli interrupt, è in grado di far fronte a questo tipo di situazioni se viene adeguatamente programmato dall'utente. Una mancanza del programma in

---

<sup>4</sup>Per rendere disponibile questo tipo di routine di interrupt il sistema operativo del dispositivo deve essere dotato di una funzione orologio.



tal senso può portare a conseguenze anche gravi per il processo controllato o a rischi per il personale dell'impianto.

Al verificarsi dell'anomalia il sistema operativo ha il compito di capire di quale si tratta e chiamare il corrispondente blocco di programma come avviene nelle chiamate ad una routine di interrupt.

Nel caso in cui si verifichi un'anomalia il cui blocco corrispondente non è presente, il sistema operativo forza il controllore nello stato di Stop che generalmente consiste in una situazione di sicurezza per l'impianto. Questo però non è sempre vero quindi è sempre preferibile che le azioni da intraprendere in caso di anomalia siano correttamente inserite nel programma utente.

### 2.3.5 Funzioni ausiliarie

Il sistema operativo dei controllori programmabili presenta di solito delle funzioni ausiliarie di vario tipo che risultano molto utili per il programmatore e che dipendono da costruttore e modello del dispositivo. Fra le principali c'è sicuramente la funzione orologio che permette di fissare e mantenere aggiornati data e ora all'interno del controllore. Questi dati sono contenuti in una particolare area di memoria del PLC e sono accessibili al programma utente per applicazioni di vario tipo (operazioni di logging<sup>5</sup> per esempio).

Una seconda funzione ausiliaria del sistema operativo molto utile consiste nella misurazione continua del tempo di ciclo del PLC e la sua memorizzazione in uno specifico registro del sistema. In questo modo, se necessario, il programma utente o l'utente stesso possono monitorare il tempo di ciclo.

Ulteriori funzioni ausiliarie del sistema operativo permettono, per esempio, di specificare le modalità con le quali vengono gestiti alcuni registri di memoria oppure la generazione e/o cancellazione automatica di blocchi contenenti dati.

---

<sup>5</sup>Per logging si intende un'operazione nella quale si associa un evento alla data e all'ora a cui esso è avvenuto



# Capitolo 3

## Programmazione

In questo capitolo verranno descritti i PLC dal punto di vista software introducendo i vari tipi di linguaggi di programmazione e spiegando le funzionalità che questi mettono a disposizione per il controllo dei processi industriali[5][1].

### 3.1 Linguaggi di programmazione

La progettazione e lo sviluppo del software di controllo per PLC presenta alcune problematiche dovute principalmente alla sua dipendenza dall'hardware del controllore da programmare. Il mercato dei controllori programmabili per l'industria è costituito da un gran numero di produttori, ognuno dei quali mette a disposizione dei propri clienti un ambiente di sviluppo proprietario. Sebbene questi software siano per certi versi simili, le differenze tra le architetture hardware dei vari dispositivi e le funzionalità per le quali questi sono progettati possono rendere molto differenti i set di istruzioni disponibili al programmatore nei vari ambienti di sviluppo.

Per favorire una convergenza fra i diversi costruttori nello sviluppo di software per la programmazione dei PLC è stato introdotto dall'organismo internazionale IEC (International Electrotechnical Commission) uno standard che si propone di definire gli aspetti descrittivi e di programmazione dei dispositivi di controllo per l'Automazione Industriale. Tale norma è denominata IEC 1131 e risale al 1993 [8]. In particolare nella sua terza parte definisce un gruppo di linguaggi di pro-

grammazione e per ciascuno di essi indica i principali campi di applicazione e le regole sintattiche e semantiche da utilizzare. La normativa definisce in particolare i seguenti cinque tipi di linguaggi:

- Sequential Function Charts (SFC)
- Ladder Diagram (LD)
- Function Block Diagram (FBD)
- Instruction List (IL)
- Structured Text (ST)

La norma consente inoltre, di sviluppare applicazioni per PLC combinando più tipi di linguaggio nello stesso programma.

Generalmente i costruttori non forniscono tutti e cinque i linguaggi per ciascun PLC in quanto non necessario se il linguaggio fornito è conforme allo standard.

Per quanto riguarda la trasportabilità del codice tra un costruttore ed un altro lo standard non prevede nessuna regola e non sono nemmeno previste novità in questo senso. Lo scopo principale della Norma IEC 1131-3 è quello di rendere disponibili dai diversi costruttori linguaggi di programmazione sempre più simili per ridurre i costi e gli sforzi nell'addestramento dei programmatori.

### 3.1.1 Sequential Function Chart

Gli schemi funzionali sequenziali (SFC) costituiscono una tecnica di programmazione basata sull'organizzazione del flusso del programma in una sequenza di passi (o fasi) e transizioni. Le fasi rappresentano le azioni da compiere per il controllore, mentre le transizioni rappresentano le condizioni da soddisfare affinché si possa passare alla fase successiva. Gli SFC danno la possibilità di realizzare delle sequenze di passi in parallelo o eventualmente con delle priorità e di inserire delle sequenze di retroazione che modificano lo svilupparsi delle sequenze normali. Questa tecnica consente di organizzare bene il software di controllo e costituisce il punto di partenza per risolvere problemi di automazione. Il programma di controllo ed in particolare le sue fasi e condizioni di transizione devono essere

successivamente sviluppati in uno degli altri linguaggi di programmazione poiché esso, analogamente ad uno schema funzionale, serve solo a dare una vista generale del sistema.

### 3.1.2 Ladder Diagram

Lo schema a contatti (LD) è un linguaggio grafico di programmazione a contatti inventato e ancora largamente diffuso negli USA. In questo caso la programmazione si realizza tramite i classici diagrammi a relè dei circuiti elettromeccanici ampiamente diffusi nell'automazione già prima dell'introduzione dei PLC. I LD mettono a disposizione del programmatore elementi tipici della logica cablata come contatti normalmente chiusi/aperti e bobine uniti a blocchi per realizzare funzioni di temporizzazione e conteggio.

Simbolo	Descrizione
*** --   --	<b>Contatto normalmente aperto</b> La corrente passa verso destra solamente se il valore della variabile booleana indicata da *** è vero.
*** -- / --	<b>Contatto normalmente chiuso</b> La corrente passa verso destra solamente se il valore della variabile booleana indicata da *** è falso.
*** -- P --	<b>Contatto rilevatore di fronti di salita</b> La corrente passa verso destra solamente se il valore della variabile booleana indicata da *** è passato da un valore falso nella scansione precedente del programma ad un valore vero in quella attuale.
*** -- N --	<b>Contatto rilevatore di fronti di discesa</b> La corrente passa verso destra solamente se il valore della variabile booleana indicata da *** è passato da un valore vero nella scansione precedente del programma ad un valore falso in quella attuale.
*** --( )--	<b>Bobina momentanea</b> Se l'espressione booleana a sinistra della bobina ha valore vero la variabile booleana *** assume valore vero per la durata della scansione corrente del programma.
*** --( )--	<b>Bobina momentanea negata</b> Se l'espressione booleana a sinistra della bobina ha valore vero la variabile booleana *** assume valore falso per la durata della scansione corrente del programma.
*** --(S)--	<b>Bobina con memoria</b> Se l'espressione booleana a sinistra della bobina ha valore vero la variabile booleana *** assume valore vero e lo mantiene fino ad un esplicito <i>reset</i> da parte di una bobina con memoria negativa.
*** --(R)--	<b>Bobina con memoria negativa</b> Se l'espressione booleana a sinistra della bobina ha valore vero la variabile booleana *** assume valore falso e lo mantiene fino ad un esplicito <i>set</i> da parte di una bobina con memoria.
*** --(P)--	<b>Bobina per fronti di salita</b> Se l'espressione booleana a sinistra della bobina passa da un valore falso nella precedente scansione del programma ad un valore vero in quella attuale la variabile booleana *** assume valore vero (solo per la durata della scansione corrente).
*** --(N)--	<b>Bobina per fronti di discesa</b> Se l'espressione booleana a sinistra della bobina passa da un valore vero nella precedente scansione del programma ad un valore falso in quella attuale la variabile booleana *** assume valore vero (solo per la durata della scansione corrente).

Figura 3.1: Principali elementi circuitali messi a disposizione dai Ladder Diagram [5]

Il concetto fondamentale alla base di questo tipo di programmazione è di rappresentare graficamente un flusso virtuale di corrente elettrica fra due barre di potenziale, regolato da interruttori bobine ecc. , in modo da implementare in modo intuitivo una logica booleana; passaggio di corrente = TRUE, assenza di corrente = FALSE. Una riga di codice corrisponde quindi ad una rete di elementi connessi da linee attraversate da un ideale flusso di corrente. La normativa IEC 1131 prevede che questo flusso si intende diretto da sinistra verso destra e che le linee di programma (network) vengono eseguite dall'alto al basso.

### 3.1.3 Function Block Diagram

I diagrammi a blocchi funzionali (FBD) costituiscono un linguaggio di programmazione grafico molto popolare in Europa. I blocchi funzionali si presentano come piccoli tasselli che, collegati gli uni agli altri come in uno schema elettrico, compongono le funzione di controllo del dispositivo. Questi blocchi grafici elaborano i segnali collegati ai loro parametri di ingresso e trasmettono i risultati dell'elaborazione attraverso i connettori connessi ai loro parametri di uscita. I principali blocchi utilizzabili per la realizzazione dell'elaborazione corrispondono alla rappresentazione grafica delle Functions e dei Function Blocks descritti dallo standard IEC 1131:

- Funzioni di conversione
- Funzioni aritmetiche
- Funzioni booleane e per stringhe di bit
- Funzioni di selezione
- Funzioni di comparazione
- Temporizzatori
- Contatori

Oltre a questi, già presenti nelle librerie fornite dal produttore, possono essere inseriti blocchi realizzati dall'utente. Gli aspetti sintattici di maggiore interesse

del linguaggio FBD riguardano l'orientamento del flusso di segnale e le regole per la valutazione complessiva della rete. Analogamente a quanto accade nei Ladder Diagram, il flusso di segnale si considera da sinistra verso destra mentre per quanto riguarda la valutazione della rete lo Standard definisce tre principi fondamentali:

- Nessun elemento della rete deve essere valutato prima che siano stati valutati i valori di tutti i propri ingressi.
- La valutazione di un elemento della rete non è completa finché non sono stati valutati i valori di tutte le sue uscite.
- La valutazione della rete termina quando tutte le uscite di tutti i suoi elementi sono state valutate.

Nonostante queste regole generali, alcuni particolari rimangono dipendenti dall'implementazione. Ad esempio, mentre per il linguaggio LD è espressamente richiesto che le reti(network) separate siano valutate dall'alto verso il basso, questo non è specificato per gli FBD. Tuttavia, è prassi comune da parte dei realizzatori di ambienti di programmazione aderenti allo Standard seguire la stessa convenzione del Ladder Diagram.

### 3.1.4 Instruction List

La lista di istruzioni è un linguaggio testuale simile all'Assembler ed è anche quello di più basso livello tra quelli definiti dallo Standard. Essendo un linguaggio assemblativo, tutti gli altri linguaggi possono avere un'equivalente in IL, mentre non è sempre possibile convertire il codice IL in altri linguaggi. Vista la sua maggior complessità di programmazione viene utilizzato principalmente per piccole applicazioni o per ottimizzare parti di un'applicazione. I linguaggi di tipo assemblativo come questo sono basati su una sintassi del tipo *1 operatore : 1 operando*. Per fare ciò occorre un particolare registro all'interno del processore detto accumulatore. In questo registro è memorizzato l'eventuale secondo operando necessario all'operazione e viene salvato il risultato al termine della stessa. Secondo le regole sintattiche della Norma IEC 1131-3 una riga di codice IL si compone di una



eventuale *label* (etichetta) seguita da *:*, un *operatore*, un'eventuale *modificatore* ed un *operando*. Si possono aggiungere eventuali commenti che non modificano il funzionamento del programma, ma che servono solo per una comprensibilità del codice. Un'istruzione IL termina con il termine della riga testuale, quindi i commenti devono concludersi entro il termine della stessa. Inoltre i commenti non possono precedere l'istruzione e si devono quindi trovare nella parte destra della riga.

Operatore	Modificatore ammesso	Significato
LD	N	Carica il valore dell'operando nell'accumulatore.
ST	N	Assegna il valore dell'accumulatore alla variabile operando.
S	Nessuno	Pone l'operando al valore booleano vero (1) se l'accumulatore ha un valore booleano vero, altrimenti lo lascia invariato. L'operando deve essere di tipo BOOL.
R	Nessuno	Pone l'operando al valore booleano falso (0) se l'accumulatore ha un valore booleano vero, altrimenti lo lascia invariato. L'operando deve essere di tipo BOOL.
AND o &	N, (	Esegue il prodotto logico fra l'operando e l'accumulatore.
OR	N, (	Esegue la somma logica fra l'operando e l'accumulatore.
XOR	N, (	Effettua l'operazione di OR esclusivo tra l'operando e l'accumulatore.
NOT	Nessuno	Effettua la negazione logica (complemento a uno) dell'accumulatore. Non richiede operando.
ADD	(	Somma matematica tra operando e accumulatore.
SUB	(	Sottrazione matematica dell'operando dall'accumulatore.
MUL	(	Moltiplica l'accumulatore per l'operando.
DIV	(	Divide l'accumulatore per l'operando.
MOD	(	Effettua il modulo, cioè calcola il resto della divisione tra accumulatore e operando.
GT	(	Confronta se l'accumulatore è maggiore dell'operando.
GE	(	Confronta se l'accumulatore è maggiore o uguale all'operando.
EQ	(	Confronta se l'accumulatore è uguale all'operando.
NE	(	Confronta se l'accumulatore è diverso dall'operando.
LE	(	Confronta se l'accumulatore è minore o uguale all'operando.
LT	(	Confronta se l'accumulatore è minore dell'operando.
JMP	C, N	Salta alla label indicata nell'operando. Se modificato da C il salto è condizionato dal valore booleano vero dell'accumulatore, se modificato con C ed N è condizionato dal valore booleano falso dell'accumulatore.
CAL	C, N	Chiama il Function Block indicato nell'operando. I modificatori C ed N agiscono come indicato in precedenza.
RET	C, N	Forza l'uscita dalla Function, dal Function Block o dal Program attuale. I modificatori C ed N agiscono come indicato in precedenza. Non richiede operando.
)	Nessuno	Termina una sequenza iniziata con una istruzione modificata da ( e fa eseguire l'operazione differita.

Figura 3.2: Principali operatori del linguaggio Instruction List [5]



### 3.1.5 Structured Text

Il quinto linguaggio, il testo strutturato (ST), è un linguaggio di alto livello introdotto in quanto non si voleva adottare uno di quelli già preesistenti (per esempio il C) perché troppo rischiosi in applicazioni di automazione industriale. Questo tipo di linguaggio assomiglia molto al *Basic* perché dà al programmatore la possibilità di combinare gli elementi degli schemi funzionali in formato testuale e scrivere il codice di controllo nello stesso linguaggio. Esso è particolarmente adatto per eseguire complesse elaborazioni matematiche in quanto permette di realizzarle con poche righe di codice oppure nel caso in cui occorra eseguire test condizionali con molteplici alternative perché rende molto più semplice e leggibile il programma rispetto a IL o LD.

Simbolo	Operazione	Priorità
( ) fun(args) - NOT	Valutazione di funzione Negazione (matematica) Negazione (booleana)	<b>MAGGIORE</b>
**	Elevamento a potenza	
* / MOD	Moltiplicazione Divisione Modulo	
+ -	Somma Sottrazione	
<, <=, >, >=	Test comparativi	
= <>	Uguaglianza (test) Disuguaglianza	
& o AND	AND booleano	
XOR OR	OR booleano esclusivo OR booleano	<b>MINORE</b>

Figura 3.3: Principali operatori del linguaggio Structured Text [5]

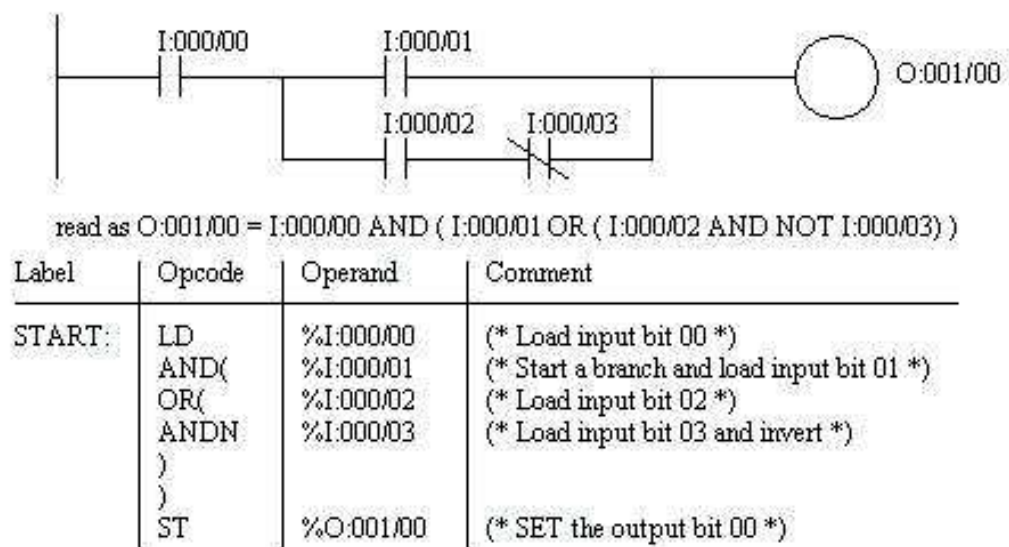


Figura 3.4: Esempio di frammento di codice implementato tramite LD (sopra) e IL (sotto).

## 3.2 Registri

Tutti i PLC hanno dei “cassetti” interni alla memoria dedicati a contenere dati o istruzioni[1]. Questi cassetti prendono il nome di registri e, a seconda del loro tipo, permettono al programma di accedere al dato contenuto e/o di modificarlo. A seconda delle dimensioni del dispositivo il numero di registri può variare da poche unità per i PLC più piccoli a diverse centinaia per quelli di dimensioni maggiori. Generalmente l’accesso ai dati contenuti in questi cassetti può essere fatto per singoli bit, per byte (8 bit alla volta), per parola o doppia parola<sup>1</sup>. I registri di un controllore programmabile vengono suddivisi in:

- **Registri speciali di sistema.** Sono dei registri interni alla CPU e hanno lo scopo di mantenere traccia dei processi interni al processore senza rendere disponibili i dati in essi contenuti ai moduli di input/output.

<sup>1</sup>Il numero di bit accessibili in questi due ultimi casi dipende dalla dimensione della parola del microprocessore interno alla CPU.

- **Registri di input.** Questi registri hanno in genere le stesse caratteristiche dei registri di sistema, eccetto per il fatto che sono accessibili dai moduli di input. Ciascun bit del registro è direttamente associato ad un terminale di ingresso: se tale ingresso è attivo (ON) il bit relativo si trova a uno, se non è attivo (OFF) il corrispondente bit si trova a zero.
- **Registri di output.** Un registro di output si differenzia dagli altri poiché è accessibile dal corrispondente modulo di output. Ogni modulo di output è associato ad un registro dal quale legge i valori da trasferire sul campo. Quando un bit del registro si trova a uno il corrispondente segnale di uscita sarà nello stato ON mentre quando il bit si trova a zero il terminale di uscita sarà nello stato OFF.

### 3.3 Timer

I dispositivi più utilizzati nello sviluppo della logica per il controllo dei processi, oltre alle bobine ed ai contatti, sono sicuramente i timer[1]. Esistono molti tipi di timer, ma il principale è quello con ritardo all'inserzione cioè quello che abilita il contatto d'uscita dopo un determinato tempo. Da questo sono derivate tutte le altre funzioni di temporizzazione quali timer con ritardo alla disinserzione, timer impulsivi e timer multipli.

I timer dei controllori programmabili sono molto più flessibili rispetto a quelli digitali in quanto permettono, collegando un potenziometro o un pannello di interfaccia<sup>2</sup> alla CPU, di settare intervalli di tempo variabili a piacere.

Generalmente in un PLC di piccole o medie dimensioni esistono soltanto uno o al massimo due tipi di timer che vengono combinati, se necessario, per ottenere le più svariate funzioni di temporizzazione. Due fra i più diffusi tipi di temporizzatori sono:

- **Timer con ritardo all'inserzione.** Il timer più diffuso nei controllori programmabili e quello che permette di realizzare tutti gli altri in modo

---

<sup>2</sup>Si tratta di un modulo con il quale l'utente può andare a leggere o modificare dei parametri di controllo. Può trattarsi di un display touch screen o di un modulo dotato di display a cristalli liquidi e di un tastierino numerico

semplice è, come detto in precedenza, il timer con ritardo all'inserzione. Un esempio è mostrato in figura. Attivando il contatto di ingresso si avvia

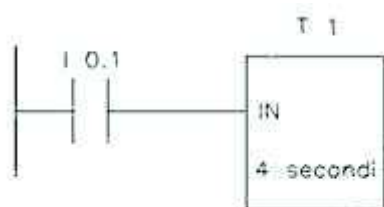


Figura 3.5: Funzione di temporizzazione nel linguaggio LD [1]

il timer che inizia il suo conteggio. Allo scadere del tempo i contatti normalmente aperti di uscita si chiudono e il timer si resetta. Successivamente quando l'ingresso viene azzerato anche l'uscita si riporta nello stato di OFF e il timer è pronto per essere attivato nuovamente. Se il contatto di ingresso si apre durante il conteggio le uscite del timer non vengono abilitate e il conteggio viene resettato.

- **Timer con memoria.** Esso si differenzia dal precedente perché presenta due ingressi, IN e Reset. Il primo ingresso IN funziona in modo del tutto simile all'ingresso del timer con ritardo all'inserzione con la differenza che se il contatto si apre prima della fine del conteggio questo non viene resettato ma tiene memoria del tempo trascorso. La funzione dell'altro ingresso (Reset) è quella di azzerare il conteggio del timer quando l'ingresso ad esso associato viene abilitato.

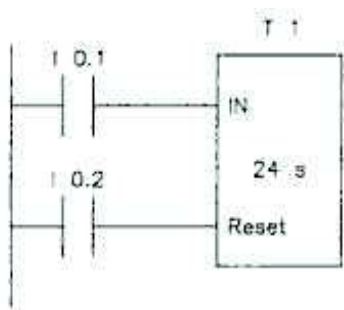


Figura 3.6: Funzione di temporizzazione con memoria nel linguaggio LD [1]

Ricorrendo al solo timer con ritardo all'inserzione è comunque possibile realizzare tutte le altre funzioni di temporizzazione quali, per esempio, il ritardo alla disinserzione, i cicli pausa-lavoro, i cicli sequenziali, i timer impulsivi ecc.

## 3.4 Contatori

I contatori nei PLC vengono utilizzati in modo simile ai timer visti nella sezione precedente: quando il conteggio raggiunge il valore prefissato il programma provvede ad eseguire un'azione[1]. La differenza sta nel fatto che mentre nel timer il procedere nel conteggio era dovuto al tempo, in questo caso il conteggio avanza al verificarsi di un evento monitorato dal controllore tramite un sensore. Un'altra proprietà che differenzia i contatori è la presenza di un ingresso di reset che se portato a livello logico alto azzerà il conteggio fino ad allora raggiunto. Generalmente ogni controllore mette a disposizione più tipi di contatori al programmatore che rendono possibile un conteggio solo progressivo (in avanti o indietro) oppure un conteggio in avanti e all'indietro con lo stesso contatore. Il funzionamento è molto semplice, il contatore in avanti incrementa di una unità il conteggio ad ogni impulso che arriva dall'ingresso da monitorare. Quando il numero di impulsi di ingresso raggiunge il valore prefissato il contatore si abilita, ovvero il bit di conteggio passa dallo stato OFF allo stato ON dando inizio all'azione che il controllore deve compiere. A questo punto anche se al contatore arrivano ulteriori impulsi il bit di conteggio rimane nello stato ON mentre il valore di conteggio continua a crescere fino ad un impulso di reset oppure al raggiungimento del valore massimo dopo il quale il dispositivo si azzerà automaticamente. Il contatore all'indietro funziona allo stesso modo con la differenza che il conteggio parte dal valore impostato per arrivare a zero.

## 3.5 Sottoprogrammi

Nella descrizione di un algoritmo risolutore per un problema di controllo si individuano molto spesso dei sottoproblemi la cui soluzione permette di risolvere più facilmente il problema di partenza[1]. A loro volta, questi sottocasi, posso-

no essere non banali e quindi scomponibili in casi ancora più semplici. Quando il problema diventa sufficientemente semplice la scrittura del programma che lo risolve diventa molto semplice per l'operatore. Spesso inoltre, l'algoritmo di risoluzione richiede di risolvere più volte lo stesso sottocaso seppur in momenti e con dati differenti. Proprio per questo motivi i costruttori di PLC mettono a disposizione dell'utente la possibilità di scrivere dei sottoprogrammi come strumento per facilitare la risoluzione dei problemi di controllo. Dal punto di vista del programma principale questi sottoprogrammi si comportano come "scatole nere": non importa il procedimento con il quale arrivano al risultato ma solo la funzione che svolgono. Alla chiamata da parte del programma principale, o di un altro sottoprogramma, questi blocchi ricevono dei dati in ingresso che verranno elaborati per restituire i risultati al momento di ritorno al programma chiamante. Un sottoprogramma, detto anche subroutine o blocco, è identico ad un programma tranne per l'intestazione e per la chiusura. L'intestazione contiene il nome del sottoprogramma e in alcuni casi la dichiarazione dei parametri globali o locali utilizzati nel frammento di codice. La chiusura si differenzia da quella di un programma in quanto, nelle subroutine, l'ultima istruzione è una istruzione di ritorno: il processore carica nel registro PC l'indirizzo dell'istruzione successiva a quella della chiamata al sottoprogramma e ricomincia l'esecuzione del programma chiamante.

Il blocco, o programma, principale è denominato generalmente OB1 e deve essere sempre presente nella memoria del PLC perché questo possa funzionare[2]. Gli eventuali altri blocchi che andranno a comporre il programma di controllo possono essere di vario tipo anche a seconda del costruttore. Fra quelli messi a disposizione dai principali ambienti di sviluppo ci sono:

- **Blocchi Organizzativi (OB).** Sono blocchi messi a disposizione dal sistema operativo che generalmente non vengono utilizzati per realizzare logiche di controllo per l'impianto, ma per caratterizzare il funzionamento del dispositivo. Con questo tipo di blocchi si possono per esempio decidere le modalità di avviamento del PLC, scegliere le azioni da compiere in caso di errori di programmazione, fissare il tempo di ciclo massimo, programmare routine di interrupt temporizzate, ecc. Il numero di blocchi organizzativi

a disposizione dipende dal costruttore e dalle dimensioni del dispositivo. Ogni OB è numerato in ordine di importanza: OB1 è il blocco principale poiché è il primo che viene eseguito all'accensione e quello dal quale vengono chiamati tutti gli altri blocchi.

- **Funzioni (FC).** Sono blocchi programmabili adatti a realizzare funzioni logiche di controllo dell'impianto. Nella loro realizzazione il programmatore ha a disposizione l'intero set di istruzioni fra cui, tra le più importanti, quelle per realizzare cicli iterativi e strutture decisionali (strutture del tipo IF-THEN-ELSE). Queste funzioni sono individuate da una sigla e da un numero (per esempio FC1 sarà la funzione numero 1). Il numero totale di funzioni disponibili dipende ovviamente dal modello e dal costruttore del PLC. Le funzioni possono avere delle variabili di ingresso e/o di uscita le quali devono essere dichiarate formalmente alla scrittura della funzione stessa e che, quando la funzione viene chiamata all'interno di un blocco, vengono sostituite con degli operandi reali.

Alle subroutine viene assegnata una zona di memoria di lavoro detta stack che ha il compito di contenere le variabili locali, cioè quelle utilizzate esclusivamente durante l'esecuzione della funzione stessa.

Per richiamare una funzione da un blocco organizzativo esistono istruzioni particolari le quali permettono chiamate condizionate o incondizionate a seconda delle esigenze (esempio: CALL o CC per il linguaggio Step7 di Siemens). Quando il processore incontra una di queste istruzioni sospende l'esecuzione del programma principale per passare all'esecuzione della subroutine trasferendole i parametri specificati nella sintassi della chiamata.

- **Blocchi Funzionali (FB).** Questi blocchi sono analoghi ai blocchi funzione visti in precedenza con la differenza che danno la possibilità di definire anche variabili statiche. Ad ognuno di questi blocchi deve essere associato un *blocco dati*, cioè un insieme di variabili raggruppate in un'area specifica di memoria identificata dal numero del blocco in questione. In questo blocco dati il FB memorizza i valori dei parametri e delle variabili statiche necessari all'elaborazione. Sarà compito del sistema operativo del PLC mantenere

aggiornati i valori delle variabili all'interno del blocco dati durante l'esecuzione del blocco funzionale.

Come per le funzioni, anche per la chiamata dei blocchi funzionali esiste una sintassi specifica che permette, oltre alla chiamata del FB, anche di scegliere il blocco dati da associargli durante l'esecuzione.



## Capitolo 4

# Utilizzo dei PLC in applicazioni di automazione industriale

I sistemi di controllo programmabili, contrariamente a quelli *custom*<sup>1</sup>, non sono progettati per il controllo di un particolare sistema, ma hanno caratteristiche generali che consentono di adattarli alle diverse problematiche di controllo che si possono presentare in campo industriale[5] [1]. La modularità dei PLC consente di adattare il controllore alle specifiche richieste dal processo e di modificarlo a piacere se tali specifiche dovessero cambiare. Il progettista si deve quindi limitare a scegliere ed assemblare i componenti che meglio soddisfano le necessità del caso, mentre l'unico sforzo progettuale si riduce alla scrittura del software di controllo. Proprio per questi motivi e per la facilità con la quale è possibile programmarlo, il PLC è diventato uno dei componenti fondamentali nell'automazione industriale. Allo stato attuale la quasi totalità degli impianti automatizzati di dimensioni medio-grandi sfrutta i controllori programmabili per la gestione dei processi, mentre per quelli di taglia più piccola si preferisce spesso sfruttare ancora controllori elettromeccanici.

A seconda delle dimensioni del processo, esistono varie possibilità di configurazione del PLC. Nei processi di piccole dimensioni, dove tutti i dispositivi da controllare sono nelle vicinanze della CPU, l'intero controllore viene messo al-

---

<sup>1</sup>Si tratta di sistemi di controllo integrati progettati ad hoc per gestire un particolare processo in base alle specifiche richieste.

l'interno di un quadro elettrico principale dal quale partono e arrivano tutte le linee che lo collegano ad attuatori e sensori. In questo caso vengono utilizzati sia PLC monoblocco che PLC modulari di piccole dimensioni. Nei processi di dimensioni medio-grandi, dove i dispositivi controllati dal PLC possono essere anche molto lontani dalla CPU, si rende necessaria la realizzazione di sistemi di controllo distribuiti (DCS) nei quali i vari moduli che compongono il controllore sono connessi tra loro tramite reti di comunicazione.

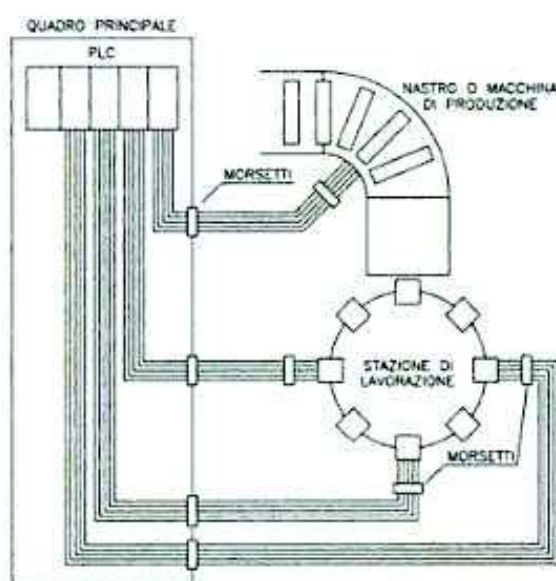


Figura 4.1: Collegamento convenzionale [1].

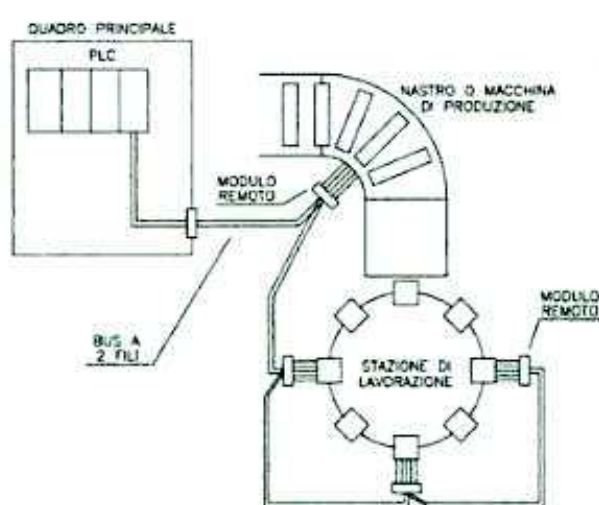


Figura 4.2: Collegamento remoto tramite bus di campo [1].

## 4.1 Sistemi di controllo distribuiti

I sistemi di controllo distribuiti (Distributed Control System, DCS) sono i sistemi di controllo più diffusi in ambito industriale e sono utilizzati per gestire processi che hanno la caratteristica di essere distribuiti geograficamente (es. un impianto di raffinazione petrolchimico)[5]. Un DCS è quindi caratterizzato dalla possibilità di distribuire in diversi punti dell'impianto i moduli per l'acquisizione, l'elaborazione ed il controllo dei dati e di connetterli in rete. I controllori programmabili realizzati in questo modo hanno il vantaggio di non avere nessun vincolo geogra-

fico sulla disposizione dei moduli nel senso che il programmatore si deve curare soltanto di quali moduli sono utilizzati nel controllore e non di come questi sono disposti.

A seconda della dimensione del processo da controllare il sistema di controllo può avere più livelli di reti di comunicazione.

- **Reti informatiche:** sono reti che collegano i sistemi di alto livello per la supervisione del processo con altri sistemi informativi aziendali. Su queste reti vengono scambiate, per esempio, informazioni relative alla produzione dell'impianto, alle scorte di magazzino ecc. La trasmissione di questo tipo di dati non richiede specifiche di real time quindi possono essere utilizzati protocolli di comunicazione standard come TCP/IP, DECNET ecc.
- **Reti per il controllo:** sono utilizzate per la comunicazione tra i dispositivi dedicati al controllo e alla supervisione delle varie parti dell'impianto (PLC, controllori di tipo custom, PC). Queste reti devono garantire la sincronizzazione delle operazioni tra sottoparti del processo controllato e richiedono quindi una comunicazione real time. Le reti per il controllo sono generalmente di tipo proprietario, cioè sviluppato dallo stesso produttore dei PLC di controllo. La rete può quindi essere composta solamente da PLC o controllori custom del costruttore in questione e da PC che siano dotati di schede e programmi appositi per l'interfacciamento.
- **Reti di campo (fieldbus):** sono le reti che rendono possibile la comunicazione tra i sistemi di controllo (ad es. PLC) e i sensori ed attuatori dotati di un apposita interfaccia. In questo caso vengono forniti moduli di ingresso/uscita dotati di microprocessore interno in modo da poter comunicare in rete rispettando il protocollo vigente. Con l'utilizzo di un controllore basato su bus di campo si hanno numerosi benefici: riduzione del cablaggio, possibilità di configurare sensori ed attuatori direttamente sul campo, maggior robustezza delle trasmissioni, possibilità di espandere e riconfigurare facilmente il controllore.

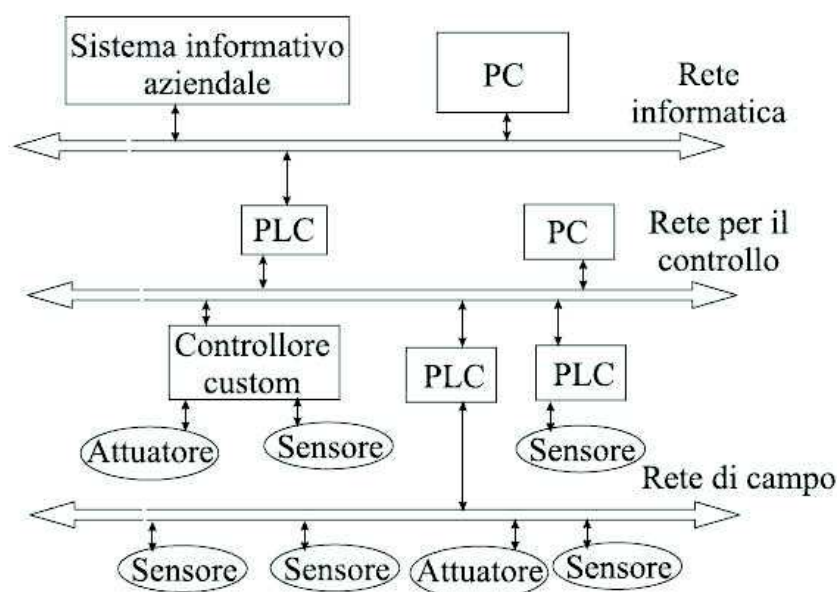


Figura 4.3: Reti per l'automazione di un processo di grandi dimensioni[5].

#### 4.1.1 Gestione delle emergenze

Mentre un guasto nei sistemi di controllo elettromeccanici porta generalmente ad una interruzione del funzionamento dell'impianto, nelle apparecchiature a logica programmata un guasto può portare a funzionamenti incontrollati e a situazioni pericolose per l'utente[1]. Per questo motivo esistono normative in ambito di sicurezza dei macchinari, ed in particolare la Norma EN 60204-1 [9] all'articolo 9.2.5.4, che prescrivono come si devono gestire situazioni di emergenza negli impianti automatizzati. Tale norma prevede che tutti i comandi di arresto e di emergenza devono essere realizzati esclusivamente con dispositivi elettromeccanici cablati e mediante contatti di interruzione. I PLC non possono dunque essere utilizzati per gestire queste situazioni, ma possono essere impiegati soltanto per rilevare ed elaborare dati relativi all'arresto. Quando il circuito manda in stato di emergenza il sistema, il PLC provvede comunque ad arrestare tutti gli attuatori interessati, segnala all'operatore il dispositivo che ha causato l'arresto e dà il consenso a riavviare l'automatismo quando l'emergenza rientra. Tipicamente per gestire queste situazioni si utilizzano degli appositi moduli di emergenza completamente svincolati dal PLC.

Un esempio di circuito di emergenza è schematizzato in figura.

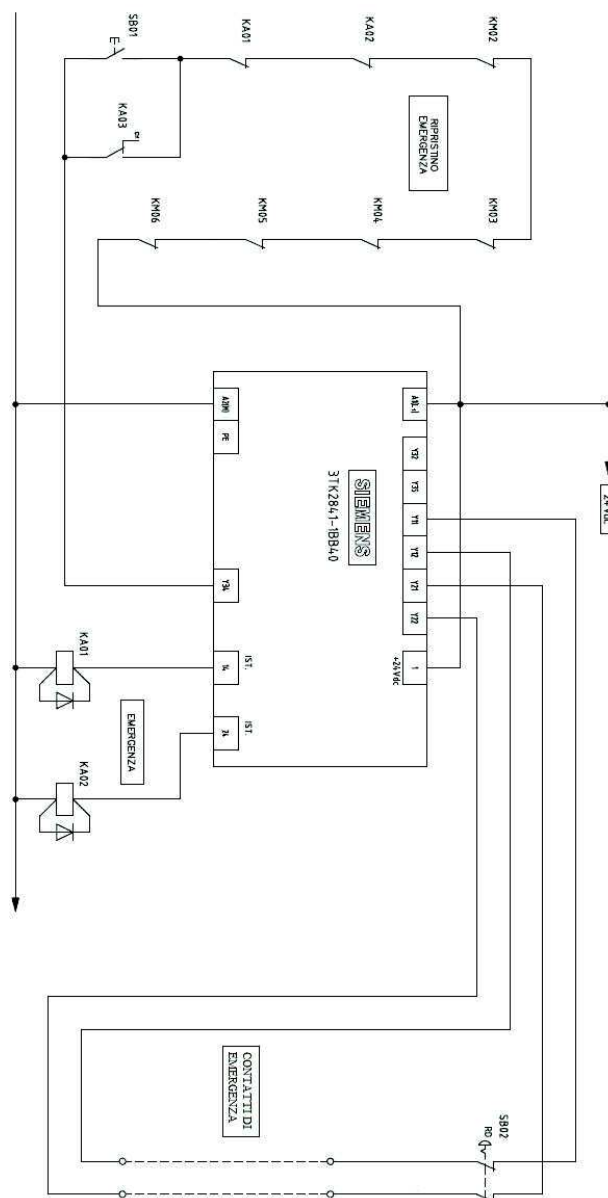


Figura 4.4: Esempio di circuito di emergenza gestito da centralina Siemens.

Il funzionamento di tale modulo è molto semplice: appena uno dei contatti di emergenza viene azionato il dispositivo disattiva i relè (KA01 e KA02 in figura) che forniscono l'alimentazione agli attuatori da arrestare e ne impedisce la messa in marcia finché il ripristino manuale non lo consente. Generalmente i pulsanti di emergenza sono dotati di doppio contatto di sicurezza NC (normalmente chiuso)

e di un terzo contatto NC o NA collegato al PLC in modo da poter individuare quale sia stata la causa dell'arresto. Il controllore può inoltre gestire tramite un terminale d'uscita uno dei contatti di ripristino in modo da consentire o meno la nuova messa in marcia del sistema.

In alcuni moduli di emergenza sono inoltre disponibili delle uscite ritardate che, a differenza di quelle istantanee del circuito in figura, disabilitano l'alimentazione agli attuatori dopo un tempo prefissato. Questa soluzione è molto utile se l'arresto immediato può provocare danni al sistema: tramite l'uscita di arresto istantaneo viene attivata una procedura di arresto controllato e solamente in seguito si toglie l'alimentazione agli attuatori evitando di danneggiare l'impianto.

#### 4.1.2 Guasti nei sistemi di controllo

In un impianto automatizzato i guasti o malfunzionamenti al sistema di controllo si distribuiscono mediamente come in figura.

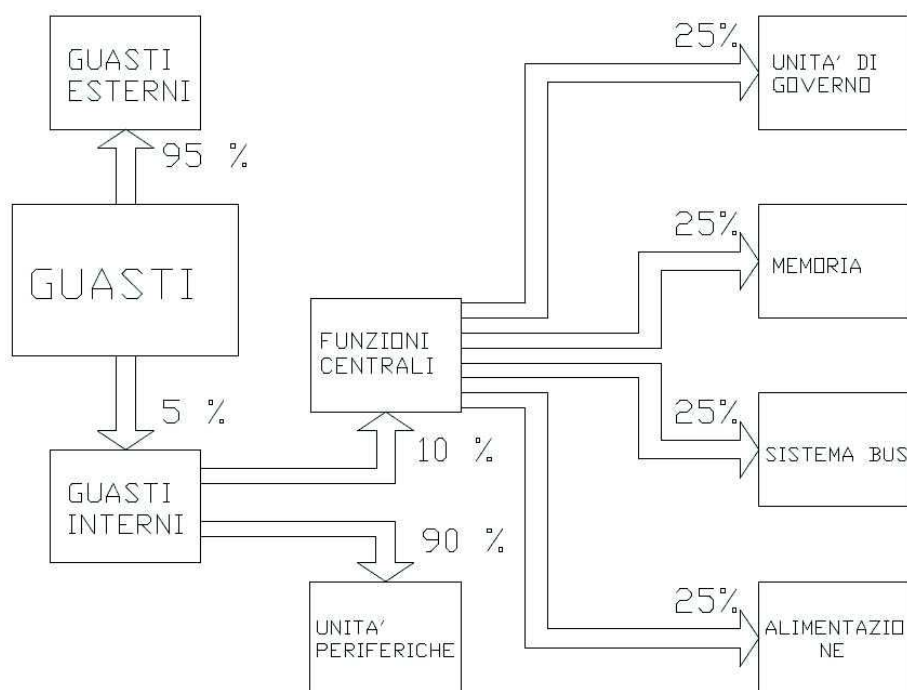


Figura 4.5: Distribuzione dei guasti in un sistema di controllo gestito da PLC[1].

Come si può vedere la maggioranza dei guasti che si verificano non riguardano il controllore programmabile (circa il 95%), ma dispositivi sul campo quali trasduttori, attuatori e i cablaggi che li collegano. Soltanto una piccola parte (circa il 5%) dei guasti avviene nei circuiti interni del sistema PLC e questi si suddividono a loro volta in:

- guasti dell'unità centrale (circa lo 0.5% del totale). Questi si suddividono poi equamente fra le varie parti che costituiscono il modulo processore.
- guasti nelle unità periferiche (circa il 4.5% del totale). Comprendono tutti i guasti che si possono verificare negli altri moduli che compongono il controllore.

Si può quindi dedurre come la maggior parte delle situazioni di emergenza provocate da guasti al sistema di controllo non siano imputabili al PLC ma agli altri dispositivi che compongono il controllore.