PLC

Controllori a logica Programmabile

MANUALE

di Programmazione

a cura di Marco Dal Prà

Edizione

16 Novembre 2005

Novatekno S.r.l. - Via Volturno, 47 - 30173 Mestre (Venezia)

www.novatekno.it

Per commenti e suggerimenti : marco.dalpra@novatekno.it

Indice generale degli argomenti

Cap.		Pagine
	Presentazione	3
	Obbiettvi	4
	Cosa non fare	4
	Prerequisiti	5
	Normative	6
1	**Qualche cenno storico**	8
2	**Concetti base nell'hardware dei PLC**	10
3	**Modi di Programmazione**	15
4	**Modi di funzionamento e Scansione**	18
5	**Numeri binari e Variabili nei PLC**	21
6	**Caratteristiche Software del PLC**	26
7	**Obbiettivi principali & Criteri di Sicurezza**	32
8	**La Progettazione del Programma**	38
9	**Componenti Elettromeccanici Virtuali**	43
10	**La Programmazione di Base**	54
11	**Cambiamenti di stato dei segnali**	57
12	**Criteri generali di programmazione**	61
	Bibliografia	64
	Copyright	65

PRESENTAZIONE

Questo manualetto è indirizzato a coloro che si approcciano alla realizzazione di automazioni mediante PLC, e contiene indicazioni pratiche per la programmazione.
Si può leggere a parte, ma è il completamento di altri manuali orientati all'installazione dei PLC ed al cablaggio.
La programmazione rimane un argomento molto ostico, in quanto è difficile parlarne senza riferirsi ad una marca precisa (i PLC sono tutti simili, ma sono anche tutti diversi).
Cercherò comunque di evitare di indicare marche o modelli.
Personalmente, dopo aver "assaggiato" varie marche di provenienza Europea ed Orientale, mi sono orientato verso un produttore Statiunitense.
Tutto sommato l'elettronica e l'informatica sono stati inventati da quelle parti, e ciò si riflette ancora oggi nei prodotti figli di queste tecnologie.
Vorrei comunque puntualizzare che un PLC per essere un buon prodotto, oltre che affidabile e completo nella gamma, deve essere semplice e pratico da programmare.
La programmazione, in particolare, deve sfruttare gli strumenti e soprattutto i concetti che si è già abituati ad utilizzare nel PC dell'ufficio e negli impianti elettrici.
Per quanto riguarda il linguaggio di programmazione si è scelto lo "Schema a contatti", dato la sua comprensibilità praticamente immediata, anche senza nessuna cognizione di programmazione, e che paradossalmente dovrebbe essere chiara anche per un matematico che conosce la logica di Boole.
Questo manualetto è un riepilogo di nozioni tecniche generali e consigli pratici; per gli obblighi di legge o imposti dalle norme tecniche è necessario consultare le stesse.
Ringrazio fin d'ora chi vorrà mandarmi suggerimenti per migliorare e rendere più completo questo documento.
Mi scuso infine se non riesco a trattenermi dal fare qua e là cenni sulla storia dell'elettrotecnica o del transistor.

Marco Dal Prà
Agosto 2004

OBBIETTIVI

Scopo di questo documento è la programmazione di PLC orientati al controllo di ***Macchine Industriali***, che è il settore d'uso primario di questo dispositivo.
Tra l'altro, data la varietà di macchine ed impianti automatizzati esistenti nel mondo industriale, questo settore è quello che ne fa l'uso più completo e severo.
Gli altri settori, come le piccole automazioni, ed il Building Automation, seguono a ruota, in quanto tipicamente hanno applicazioni più semplici.

Il cammino, in ogni caso, sarà lungo, in quanto si vorrà anche far conoscere al lettore i rischi in cui si incorre scrivendo un programma in modo approssimativo e senza una attenta "progettazione"
Il tutto allo scopo di realizzare software affidabili, sicuri e comprensibili.

COSA NON FARE !

Tipicamente appena ci si approccia ad un PLC si installa subito il software di programmazione nel PC e si comincia per tentativi a scrivere il programma.
Poi, dopo qualche ora, ci si rende conto che non solo non si conosce bene il software, che però se si è fortunati è stile Windows e si riesce quantomeno a farlo funzionare, ma che non si sa assolutamente nulla del PLC.
Ecco che per dipanare la matassa si comincia a guardare il manuale di programmazione del PLC per capire come si chiamano le istruzioni, come è organizzata la memoria, ecc.
Dopodiche' si inizia di nuovo a smanettare con il PC per tentare di sviluppare il programma.
Tutto ciò comporta il risultato di realizzare un programma confuso, senza capo ne' coda, che poi si è costretti a rivedere da zero.

Questo manuale affronta il problema a monte, cercando di impostare un metodo di lavoro rigoroso e lineare.

L'uso del PC e la trascrizione del programma è solo l'ultimo passo !

PREREQUISITI

Questo testo non è alla portata di tutti, e dà per scontate molte nozioni tecniche.
In primo luogo è necessario avere dimestichezza con gli Schemi Funzionali, condizione inderogabile per approcciarsi alla programmazione dei PLC.
Ne consegue che bisogna essere a conoscenza dei concetti generali di Elettrotecnica : che cosè un relè, un motore, un avviamento stella-triangolo, una elettrovalvola, ecc.

Per utilizzare un PLC è anche indispensabile saper utilizzare un Personal Computer con i relativi software, visto che la programmazione dei PLC si esegue proprio in tale ambiente (generalmente con sistema operativo MS - Windows).
Non è invece necessario essere degli esperti di elettronica, anche se le nozioni di base non guastano.

Non nascondo il fatto che i tecnici informatici tipicamente non sono adatti per predisporre questi programmi, in quanto nella loro cultura mancano le basi di elettrotecnica e di meccania.

Per finire, non è possibile approcciare al mondo industriale e dei PLC con questi preconcetti :

- I motori a gabbia di scoiattolo sono usati negli Zoo-Safari,
- La fotocellula è quella che accende la luce delle scale,
- kVA significa ChiloWattAmper,
- Un comando pneumatico è utile per sterzare con l'automobile,
- Un inverter serve per usare la TV quando si è in camper.

Del resto il programma in un PLC, come già detto, non è altro che una "rappresentazione su PC " dello schema funzionale.

NORMATIVE

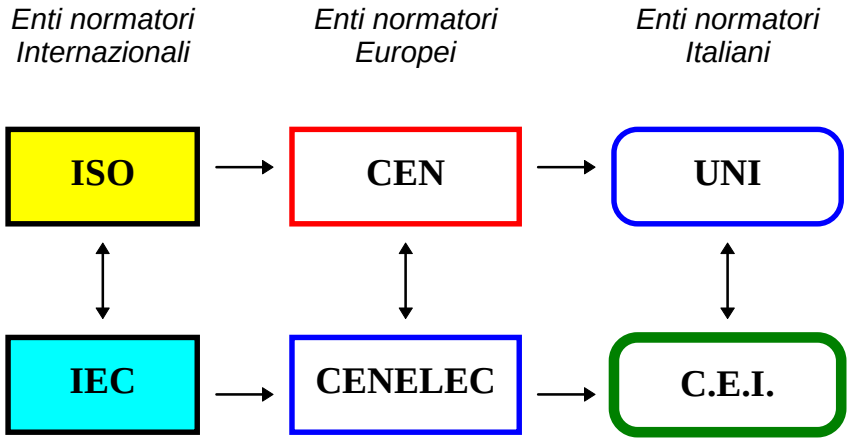
Le norme sono documenti di standardizzazione emessi da comitati tecnici nazionali, europei o internazionali, nei campi più svariati della tecnologia.
Non sono da confondere con le Direttive, che sono praticamente le "leggi" emesse dal parlamento europeo.
Le normative nel mondo sono emesse in due "filoni" :

- le norme norme di orientamento generale in quasi tutti i campi, dal comitato **ISO**
- le norme di orientamento elettrico ed elettronico, dalla commissione **IEC**

In Italia la gestione delle norme tecniche è curata da questi due enti :

- l' **UNI**, che emette norme norme di orientamento generale
- il **CEI**, che emette le norme nei settori Elettrico, Elettronico e Telecomunicazioni

L'iter delle norme tecniche (detti Standard in inglese) è quello qui rappresentato, che a partire dai comitati internazionali, passando per i comitati europei, arriva agli enti italiani.



Le norme internazionali, tra l'altro, vengono scritte "a quattro mani" tra i comitati Europei e quelli mondiali, per evitare tempi morti delle ratifiche e raccogliere i commenti dei comitati nazionali.

> ***Curiosità***
> Quando una norma IEC viene approvata "in toto" dal CENELEC, al numero IEC si somma 60.000 e viene aggiunto il suffisso EN.

Chi si approccia al mondo dei PLC dovrebbe quantomeno conoscere le normative tecniche che vi si applicano.

Le norme “elettriche” che incidono nella progettazione e programmazione di un PLC sono essenzialmente due :

Descrizione	Denominazione IEC	Denominazione Cenelec	Numerazione CEI
Impianti Elettrici Utilizzatori	364-1	(HD 384)	64-8
Equipaggiamento elettrico d. macchine	204-1	EN 60204-1	44-5

Dato che l’applicazione prevalente dei PLC è sugli impianti “a bordo macchina” la norma EN 6024-1 è quella più importante da conoscere.
Di riflesso un impianto elettrico con PLC fatto nel rispetto di tale norma, data la radice comune di entrambe, rispetta anche l’altra.

> ***Nota***
> Nozioni inerenti questa norma, e relativi consigli pratici ed esempi applicativi si trovano anche su diversi libri.
> Se ne possono trovare alcuni nella bibliografia in fondo al presente documento.

Le norme “di prodotto” relative ai PLC sono invece :

Descrizione	IEC	Cenelec	CEI
Controllori Programmabili - Informazioni Generali	1131-1	EN 61131-1	65-23
Controllori Programmabili - Linguaggi di Programmazione	**1131-3**	EN 61131-3	65-40

CAPITOLO 1.

QUALCHE CENNO STORICO

1.1 Introduzione

All'inizio c'era l'Elettrotecnica, e l'automazione si realizzava con relè e relativi contatti collegati in serie o in parallelo.
In questo modo per far funzionare un qualunque macchinario, semplice o complesso, si cablavano i vari componenti elettrici nella sequenza esatta di come veniva richiesto, realizzando quindi la *Logica di Funzionamento*.
Ne conseguì che il disegno che rappresenta questi componenti nel gergo venne appunto chiamato "schema funzionale".

La logica di ogni componente era ovviamente a due stati, che si identificava con il relè eccitato o meno.
Gli elettrotecnici, senza volerlo, avevano messo in pratica la logica binaria di *George Boole* (matematico inglese, 1815-1864), utilizzando relè e circuiti elettrici !

Poi un giorno arrivarono le "valvole termoioniche", cosicché si cominciò realmente ad intravedere l'elettronica digitale.
Questi circuiti a valvole, infatti, potevano eseguire delle operazioni logiche che si facevano con i relè, ma molto più velocemente (e nel silenzio).
Con le valvole si passò inoltre alla costruzione di un elaboratore che potesse compiere operazioni "logiche" a piacimento, e quindi eseguire un programma, mentre con i relè, una volta cablati, si poteva solamente ripetere sempre la stessa operazione.

> Il primo computer fu realizzato nel 1943 dal governo americano per scopi militari : si chiamava ENIAC, ed era composto da 18.000 valvole, che causavano un consumo elettrico per 150 kW.
> Mentre funzionava aveva schiere di tecnici che continuavano a girare tra i suoi pannelli con carrelli zeppi di valvole di ricambio.
> Dato il consumo notevole per qui tempi, si racconta che alla prima accensione il sovraccarico causò il black-out di un intero quartiere di Philadelphia.

1.2 L'Elettronica Digitale

Il transistor, componente inventato nel 1945 nei laboratori Bell della compagnia telefonica americana AT&T, è la vera rivoluzione nel ramo della tecnologia elettrica.
Con esso praticamente si scinde dall'elettrotecnica una nuova disciplina tecnica basata sull'elaborazione di segnali elettrici a bassissima energia, ossia l'*Elettronica*.
L'elettronica aveva quindi come scopo non di inviare elettricità come forma di energia, ma come segnale che esprime un preciso significato, come si faceva con il telefono o con il telegrafo.
Inizialmente l'elettronica era orientata alla gestione dei segnali elettrici (*elettronica analogica)*, e quindi con lo scopo di manipolarne il valore, amplificarlo, filtrarlo, ecc. come richiesto dagli impianti telefonici, ma gli sviluppi furono presto in tutti i settori.

Dato che il transistor aveva soppiantato le valvole, anche per chi si occupava di logica binaria con le valvole era nata una nuova era : l' ***Elettronica Digitale***.
In questa disciplina sono cambiati i componenti ed il dimensionamento "elettrico", ma i concetti base dell'algebra di Boole sono comunque rimasti invariati.

In pratica il relè viene virtualemente sostituito da una cella di memoria del circuito elettronico, denominata bit, dove gli stati possono essere 1 e 0.

Riassumiamo i paragoni tra i due mondi

Elettrotecnica	***Elettronica***
Collegamento in serie di contatti	Logica AND
Collegamento in parallelo	Logica OR
Relè Eccitato	Bit = 1 (VERO)
Relè Diseccitato	Bit = 0 (FALSO)

Nota : Per chi voglia divertirsi ad approfondire l'Elettronica Digitale, si rimanda a testi specifici (quelli di Quarta ITIS vanno bene).
Nota 2 : per chi ha tempo nel dilettarsi potrebbe studiare l'algebra di Boole, ma è molto più bello usare un PLC in quanto la si *vede* realmente funzionare !

> Anche se può sembrare impossibile, ma teoricamente un PC potrebbe essere costruito con relè anziché con transistor.
> Resterebbe "solo" il problema di dove mettere i milioni di relè che sono necessari per realizzarlo !

CAPITOLO 2.

Concetti Base nell'Hardware dei PLC

2.1 I Microprocessori

Il microprocessore è il dispositivo massima espressione dell'elettronica digitale : è molto piccolo, fa moltissime cose e consuma pochissima energia.
Questo *gadget* compie operazioni logiche (e matematiche) molto velocemente, e soprattutto le esegue secondo una "scaletta" predefinita a nostro piacimento, chiamata semplicemente *programma*.

> L'elenco di tutte le operazioni che un microprocessore può eseguire è detto Set di Istruzioni, in inglese *Instruction Set*.

Il programma quindi non è altro che un elenco di operazioni che deve compiere il microprocessore.

Tra le tante istruzioni che un microprocessore può elaborare vi sono anche quelle della logica binaria (vedi ad esempio AND e OR citati in precedenza), ecco quindi che può essere utilizzato per elaborare dei segnali elettrici e comandare delle macchine, proprio come si fa in elettrotecnica con i circuiti di comando.
La trasposizione del microprocessore dal mondo dell'elettronica digitale al mondo elettrotecnico è stata quindi abbastanza semplice, viste le affinità dal punto di vista della logica.

Queste affinità comportano, in generale, una straordinaria conseguenza :

ogni schema funzionale può essere tradotto in una sequenza di istruzioni per un microprocessore.

E' quindi possibile trasformare uno schema elettrico funzionale che comanda una macchina in un programma che praticamente svolge la stessa funzione all'interno del microprocessore.
Un PLC è appunto un apparecchio dotato di microprocessore e destinato al comando di dispositivi elettrici, ma con l'enorme vantaggio che una modifica del programma è molto semplice e veloce, rispetto alla modifica del circuito elettrico che deve essere ricablato.

2.2 Differenza PC - PLC

Vediamo nella tabella che segue una comparazione esemplificativa che confronta i mondi PC e PLC :

Caratteristica	***PC***	***PLC***
Spostamento di dati	> 500.000 kByte/sec	< 10 kByte/s
Dimensione dei programmi	> 10.000 kByte	< 10 kByte
Operazioni binarie tipiche	spostamento di 32 bit	operazioni su 1 bit singolo
Frequenza Microprocessore	> 1 GHz	< 100 MHz
Funzionamento tipico	8 ore al giorno	24 ore su 24
Immunità a disturbi elettrici	scarsa	elevata
Condizioni ambientali	interno climatizzato	da 0 a 55 °C
Programmazione	Compilata con linguaggi ad alto livello	diretta, praticamente in "linguaggio macchina"

Il programma di un PLC, se confrontato con i software che abbiamo nel Personal Computer posto sulla nostra scrivania, è veramente piccolo, anzi microscopico.
Quando si usa un PC il microprocessore al suo interno sposta centinaia di MegaByte al secondo attraverso i vari bus, mentre nel PLC si spostano pochi Byte, se non addirittura bit singoli.

Paradossalmente si potrebbe costruire un PLC anche con un processore per PC dell'ultimissima generazione, ma sarebbe *un tantino* sovradimensionato.

Tipicamente un microprocessore, di qualunque tipo, può eseguire un programma :

a) una volta sola, come ad esempio il calcolo di un'espressione matematica, oppure
b) all'infinito come succede per il Sistema Operativo, che praticamente è sempre in esecuzione nel microprocessore.

Nel PLC regolarmente in funzione, il programma è praticamente sempre in esecuzione, in quanto devono sempre essere controllati i circuiti che vi sono collegati.

2.3 Caratteristiche Hardware

Come già accennato, un PLC è un piccolo computer, ma a differenza del PC non deve interfacciarsi con l'uomo, ma con circuiti elettrici.
Il suo microprocessore non si aspetta quindi di ricevere un segnale da un mouse o da una tastiera, ma da dispositivi elettrici quali un singolo pulsante, un contatto di allarme, un segnale di livello, ecc.
A sua volta il PLC, una volta elaborati i segnali di ingresso tramite il programma, produrrà in uscita un risultato atto, non a essere visualizzato su un monitor, ma a comandare un motore, elettrovalvole, spie luminose, ecc.

Ingressi	**Uscite**
Gli ingressi costituiscono gli "occhi" del PLC, in quanto è con essi che il programma si rende conto di quello che sta succedendo nel macchinario controllato. Gli ingressi digitali sono quei morsetti del PLC ai quali vengono collegati componenti che al loro interno hanno un semplice contatto. Tipici esempi di questi componenti sono: • Pulsanti, • Finecorsa, • Relè Termici, • Fotocellule, • Sensori di Prossimità, • ecc.	Le Uscite costituiscono le "mani" del PLC, in quanto è con esse che il programma comanda gli Attuatori, o organi di segnalazione. Le usicte digitali sono sono quei morsetti del PLC ai quali vengono collegati componenti attivi che tipicamente effettuano una azione "meccanica". Tipici esempi di questi componenti sono: • Relè, • Contattori • Lampade di segnalazione • Elettrovalvole • ecc.
Nelle schede di ingresso il PLC ha dei circuiti che "adattano" i segnali elettrici provenienti da macchinari e/o impianti, alle tensioni e correnti interne al PLC; Tipicamente i circuiti di ingresso sono costitiuti da componenti elettronici detti fotoaccoppiatori che acquisiscono seganli a 24 Vcc.	Nelle schede di uscita il PLC ha dei circuiti che "adattano" i segnali interni a quelli dicomando. Generalmente il microprocessore pilota delle bobine di microrelè o dei transistor di tipo "open collector".

Tra gli accessori più usati da non dimenticare i dispositivi di interfaccia uomo-macchina, indispensabili per segnalare errori, anomalie, e dare la possibilità all'operatore di inserire set-point e scegliere modalità diverse di funzionamento.

2.4 Tipi di Segnali elettrici Gestiti

Tipicamente un PLC, nell'interfacciarsi con il mondo esterno, può gestire :

Segnali Digitali	
Digital Inputs (DI)	Gli Ingressi digitali sono segnali proveneinti da contatti, pulsanti, termostati, ecc. e che tipicamente hanno tensione 0 quanto OFF e tensione +24 Vcc quando ON.
Digital Outputs (DO)	Le Uscite Digitali sono i segnali con i quali il PLC comanda (tramite relè ausiliari e/o contattori) gli attuatori, quali motori, elettrovalvole, segnalazioni, ed altri circuiti.
Segnali Analogici	
Analog Inputs (AI)	Gli Ingressi Analogici sono segnali provenienti da trasduttori di pressione, portata, o termometri, igrometri, analizzatori chimici, analizzatori di Energia Elettrica e altri strumenti che trasducono la grandezza fisica analizzata in un segnale elettrico proporzionale (tipicamente 4-20 mA o anche 0-10 V);
Analog Outputs (AO)	Le Uscite Analogiche sono segnali atti a pilotare valvole proporzionali, strumenti indicatori, registratori, Regolatori di Velocità per motori (Drives o Inverter), e altre apparecchiature regolatrici.

In particolare la gestione di queste ultime due categorie di segnali è di grande utilità e consente un enorme passo avanti ai sistemi di controllo programmati rispetto a quelli cablati, con i quali non sono possibili simili operazioni.

Ad esempio è possibile collegare un PLC ad una sonda di temperatura in modo che acquisica il suo valore istantaneo, per poi effeture un comando.
Con i circuiti cablati l'unico dispositivo similare è il termostato, ma il programma non può sapere se manca tanto o poco a raggiungere la temperatura preimpostata.

Nota
Per un approfondimento di tutti questi aspetti si rimanda alle altre sezioni del "Manuale Pratico del PLC" (edizione 2002).

2.5 Dummy, il nostro PLC ideale

Dato che nel presente manuale si farà spesso ricorso ad esempi, configuriamo ora un PLC virtuale con il quale fare tutte le nostre successive sperimentazioni, e che chiameremo “Dummy”.
Dummy è un ipotetico PLC di media potenza e con caratteristiche Hardware e Software abbastanza aggiornate, in modo da poter sviluppare programmi con le concezioni più avanzate (senza esagerare).
In poche parole Dummy ci metterà a disposizone gli strumenti più moderni e contemporaneamente più pratici che si hanno a disposizione

Come è fatto Dummy

Il PLC sarà composto da un telaio, detto rack, con 7 posti scheda così utilizzati :

- Slot 0 - CPU
- Slot 1 - Modulo con 16 segnali di Ingresso Digitali 16 DI
- Slot 2 - Modulo con 8 segnali di Ingresso Digitali 8 DI
- Slot 3 - Modulo con 16 segnali di Uscita Digitali 16 DO
- Slot 4 - Modulo con 8 segnali di Uscita Digitali 8 DO
- Slot 5 - Modulo con 8 segnali Analogici di Ingresso 8 AI
- Slot 6 - Modulo con 4 segnali Analogici di Uscita 4 AO

Da ciò risulta il seguente disegno :

nr.	***0***	***1***	***2***	***3***	***4***	***5***	***6***
Modulo Tipo In/Out	**CPU**	16 in Digitali	8 in Digitali	16 Out Digitali	8 Out Digitali	8 In Analog.	4 Out Analog.

Nota : tipicamente a lato della CPU vi è il modulo Alimentatore, ma non ha numerazione.

CAPITOLO 3.

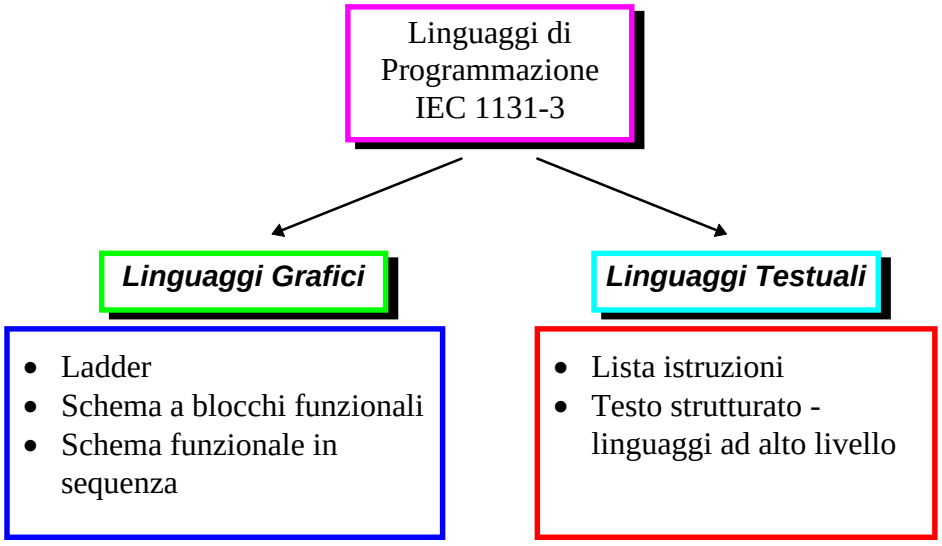
Modi di Programmazione

3.1 Generalità

Il presente capitolo affronta brevemente le varie modalità di programmazione dei PLC soffermandosi poi sulla più usata, che tra l'altro è quella che ricalca in forma grafica gli schemi elettrici.

3.2 Modi di Programmazione

La norma di riferimento sulla programmazione dei PLC, come già indicato nella premessa dedicata alle normative, è la IEC 1131-3, che riassume un po' tutte le modalità di programmazione.
Di tutti i linguaggi di programmazione si può fare una prima macro-distinzione in due grandi categorie :



3.3 Linguaggi di programmazione testuali

Questi modi di programmazione sono quelli più "ostici" per chi si approccia al mondo PLC dal settore elettrotecnico.
Tali sistemi di programmazione sono di provenienza prettamente elettronica e/o informatica e quindi costituiscono dei veri linguaggi di progammazione con enormi possibilitò di sviluppare le necessità del progetto.

Lista Istruzioni - I*L* - *Instruction List*

Questo modo di programmazione è praticamente il linguaggio macchina, ossia un linguaggio che usa direttamente le istruzioni del microprocessore.
Tale sistema oltre che poco intuitivo è poco pratico, in quanto no ha alcuna rappresentazione grafica e richiede molto tempo al programmatore per la ricerca di errori nel programma o guasti all'impianto controllato.
Alcuni PLC non permettono l'uso di questo sistema, mentre con altri è indispensabile per sfruttare appieno le potenzialità del PLC.

Testo Strutturato (ST)

Questo è un linguaggio di programmazione ad alto livello, come lo sono il Pascal, il Basic, il linguaggio C++, ecc.
Talvolta è indispensabile determinate applicazioni e/o reti di comunicazione

3.4 Linguaggi di programmazione Grafici

Tuttaltro mondo è quello dei linguaggi di programmazione grafici, che si presentano al programmatore come veri e propri schemi elettrici o schemi a blocchi.

Schema a contatti - *LD - Ladder Diagram*

Questo è il linguaggio di programmazione più usato, in quanto è analogo ad uno schema elettrico funzionale.
Ladder significa letteralmente "scala a pioli", dato che esteticamente lo schema ricorda appunto una scala; nel mondo anglosassone ogni ramo orizzontale viene chiamato *rung*, ossia piolo.
Nel paragrafo seguente verrà ampiamente approfondito.

Schema Blocchi di Funzione - *FBD - Funcion Block Diagram*

E' un linguaggio a "porte logiche" che permette di disegnare uno schema classico dell'elettronica digitale
E' molto usato nei sistemi di controllo dei grossi impianti di processo (centrali termoelettriche, impianti chimici, ecc).

Schema Funzionale in Seguenza - *SFC - Sequential Funcion Chart*

E' un linguaggio sviluppato in Francia con il nome di "Linguaggio Grafcet".
Rappresenta il funzionamento per passi di un processo automatico in modo del tutto similare ad un Flow-chart, ma dove ogni blocco rappresenta uno stato del processo di lavorazione della macchina.

3.5 Lo Schema Funzionale Europeo

Nelle norme del gruppo EN 61082 riguardanti la documentazione degli impianti elettrici, il termine "schema funzionale" non esiste.
Il termine schema funzionale è comunque ampiamente consolidato, e consente di distinguere chiaramente lo schema di comando (tipicamente un circuito FELV), dallo schema di potenza (tipicamente a 230/400Vac).
In esso si trovano i contatti e tutti i relè che compongolo la parte elettrica, cosicchè talvolta nel gergo viene chiamato anche "*Schema degli ausiliari*".

> **Precisazione**
>
> Il più simile al termine nella norma EN 61082-2/5 è il termine *Circuit Diagram*, che indica un disegno nel quale può essere trascritto sia lo schema di comando, sia lo schema di potenza.
> Per completezza, nella norma si trova anche il termine *logic-function diagram*, ma esso indica uno schema a porte logiche, ad uso abituale nell'elettronica digitale, che corrisponde appunto al linguaggio FBD.

Lo schema funzionale, in ogni caso, è un disegno che si sviluppa con i rami disposti in senso verticale, e nel quale si trova nella parte alta la sorgente di alimentazione dei circuiti ausiliari, e nella parte bassa il conduttore "comune" (detto anche di ritorno o anche neutro nel caso di circuiti a 230V).

3.6 Lo Schema Funzionale Americano

Negli Stati Uniti lo schema di comando di una macchina viene tracciato in per linee orizzontali, e da sinistra verso destra come nella scrittura ordinaria.
In pratica è come se si ruotasse lo schema Europeo di 90°.
In tale schema si parte da sinistra con l'alimentazione e passando per i vari contatti si arriva alla bobina del relè nella parte destra.

> Per fare questi disegni, anzichè fogli, si usavano talvolta dei lunghi rotoli da consultare in senso longitudinale, come era uso farsi nel medioevo.

Questo modo di fare gli schemi di comando è stato di grandissimo aiuto per l'invenzione del PLC, infatti i software nei PC non fanno altro che riproporre sullo schermo lo schema elettrico, mentre al loro interno in realtà lo traducono in una sequenza di istruzioni per il PLC.
Per il tecnico "elettrico" statunitense, quindi, questa nuova tecnologia è stata praticamente indolore, in quanto si è ritrovato ad usare i soliti simboli, i soliti schemi, ma nello schermo del computer, con l'ovvio vantaggio di poterli modificare a piacimento senza spostare alcun filo.

CAPITOLO 4.

Modi di Funzionamento e Scansione

4.1 Introduzione

La programmazione dei PLC, eccetto che per modelli molto piccoli non trattati nel presente manuale, tipicamente si esegue mediante un software dal Personal Computer.
Questa fase di scrittura del programma si svolge Off-Line, cioè scollegati dal PLC, che può non essere ancora nelle mani del programmatore.
Una volta preparato il programma, si collega il PC al PLC con un cavetto seriale e si "scarica" il programma nella memoria del PLC stesso.
Questa operazione è chiamata ***Download***.

In questi frangenti si dice che il PC è On-Line, in quanto dallo schermo del PC si può vedere il programma in esecuzione e se ne possono vedere le "evoluzioni" durante il funzionamento dell'impianto che il PLC controlla.

> ***Nota di invidia*** : gli informatici che scrivono i programmi ordinari per PC non possono fare una cosa del genere, ci vorrebbero due PC, uno che "guarda" dentro l'altro !!

Nei PLC di maggior potenza è anche possibile, attraverso comandi dal PC collegato On-Line, modificare il programma nel PLC mentre è in Run (!).
In tutti gli altri è necessario modificare il programma nel PC, salvarlo, e quindi scaricarlo completamente nel PLC a programma fermo.

4.2 Modi di Funzionamento

Quasi tutti i PLC, all'esterno, sono dotati di un piccolo selettore con il quale si può selezionare il modo di funzionamento.

Tipicamente questo selettore ha due posizioni :

- ***Modalità Program*** (Programmazione)
  In questa modalità il PLC è "fermo" e attende che il programmatore effettui modifiche al programma stesso o alla configurazione della memoria.
  Alcuni costruttori chiamano questa modalità "STOP".
  Praticamente rimane in uno stato di stand-by dal quale non esce fino a nuovo ordine.
  In ogni caso, quando il PLC è in Stop le uscite sono tutte disabilitate e quindi tutti gli apparecchi da esso controllati sono spenti.
  Questo particolare non è da trascurare per i PLC utilizzati nel settore del Building Automation, dove i PLC comandano circuiti di illuminazione : in tali casi gli interventi di modifica al software dovranno essere attentamente pianificati.

- ***Modalità Run*** (Esecuzione)
  In questa modalità il PLC esegue il programma interno ed aggiorna le uscite sulla base dello stato degli ingressi e delle condizioni imposte dal programma.
  Quando il PLC è in RUN e accade un errore irreversibile nel programma o nei componenti elettronici interni, il PLC passa automaticamente in Program (o anche in STOP) attendendo che il programmatore si connetta con il PC per verificare tipo e origine dell'errore.
  Una causa classica di fault di un PLC è l'esecuzione di una operazione aritmetica che tenti una divisione per zero.

In altri modelli di PLC, privi del suddetto selettore, la commutazione del modo di funzionamento viene operata attraverso il software di configurazione.

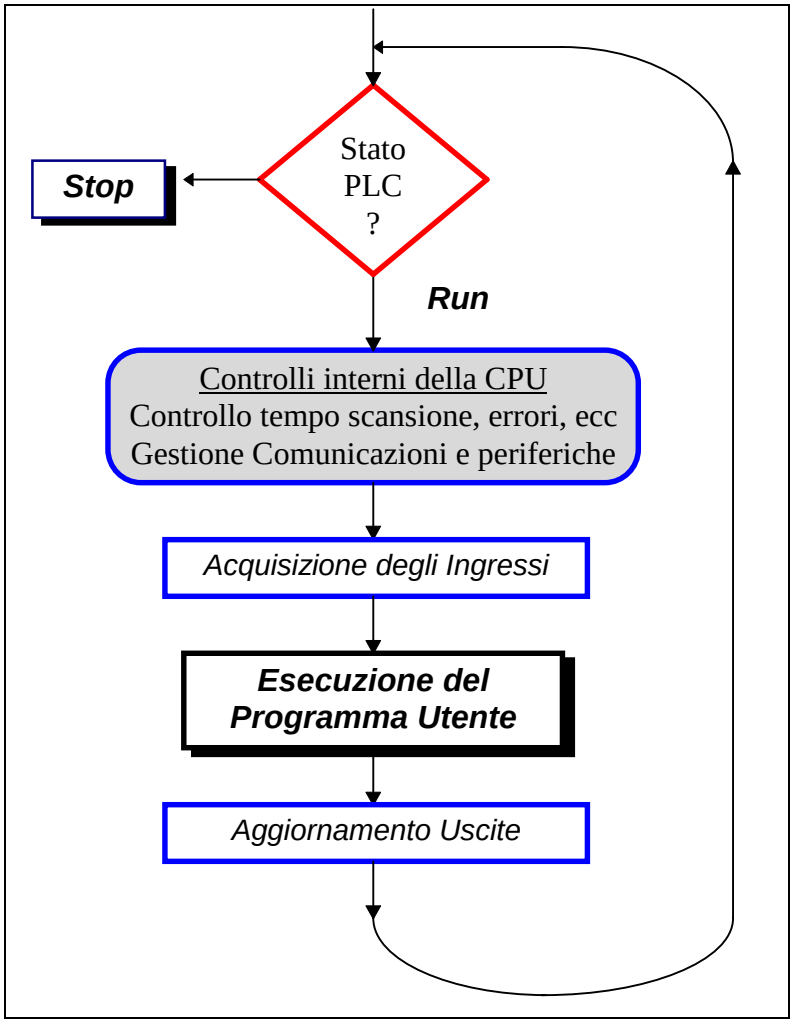
4.3 Scansione

Come già accennato, quando il PLC è in Run, il programma è praticamente sempre in esecuzione per tenere sotto controllo la macchina o l'impianto al quale è collegato.
Dato che il microprocessore esegue ciclicamente il programma, tipicamente si usa dire che il PLC esegue la *Scansione del programma*, in quanto viene eseguito dall'inizio alla fine ad intervalli di tempo piuttosto regolari.

Naturalmente quando è in Run il PLC non si cura solo dell'esecuzione del programma, ma si occupa anche di :

- controllare se il programma ha causato errori
- controllare se l'esecuzione (scansione) del programma è durata troppo
- Controllare lo stato dell' Hardware, ossia di tutte le perifiriche che lo compongono;
- aggiornare lo stato dei file degli ingressi e delle uscite,
- Comunicare con altri PLC o altri dispositivi connessi via porte di comunicazione

Il tempo di scansione con i PLC attualemente in commercio, comprensivo di tutti i controlli, tipicamente si attesta intorno ai 2-3 millisecondi.



In realtà per il programmatore questa informazione della scansione ha un aspetto marginale che interessa solo nei casi in cui si controllano macchine ad elevata velocità.

In tali casi infatti l'aggiornamento degli ingressi e uscite non può essere rimandato ad altri momenti, ma deve essere fatto in sicronia con alcuni calcoli o elaborazioni del programma.

In taluni casi il programmatore può inserire nel mezzo del programma una istruzione di REFRESH immediato degli Intup/Output che non attende altri tempi morti per aggiornare il suddetti file di I/O.

Figura 4.3 - Scansione del programma nel PLC

In alcuni casi la scansione può tornare veramente utile per far eseguire operazioni in sequenza, ma su questo aspetto ci si soffermerà negli esempi dei prossimi capitoli.

E' comunque importante ricordare che l'aggiornamento degli ingressi avviene in un momento distinto dall'esecuzione del programma, quindi mentre questo è in esecuzione anche se fisicamente gli ingressi cambiano, ***il programma non se ne rende conto !***
Questo finchè non avviene una nuova scansione, che comunque avviene ogni manciata di millisecondi.

CAPITOLO 5.

Numeri Binari e Variabili nei PLC

5.1 Introduzione

In questo capitolo si daranno alcuni accenni sul sistema di numrazione binaria e soprattutto di come è stato utilizzato all'interno dei PLC.
In particolare si farà riferimento alla norma quadro sulla programmazione dei PLC, ossia la IEC 1131-3.
Non è possibile infatti proseguire senza avere chiaro in testa il concetto di rappresentazione dei numeri decimali con un numero binario.
I PLC, tra l'altro, hanno vari modi di rappresentare i numeri al loro interno, che in fase di programmazione si dovranno scegliere sulla base delle proprie esigenze.

5.2 Bit, Byte e Word

L'informazione digitale di base è il bit, una cella di memoria che fisicamente è costituita da un transistor e che si considera che possa valere 0 (quando non attivo) o 1 (quando attivo).

Nota

La tensione di funzionamento del transistor è ininfluente ai fini della logica binaria, e non è di interesse per il programmatore.
Al fine di erudire i più curiosi diremo che ovviamente quando si dice che il transistor è nello stato logico "0" anche la tensione del transistor sarà a zero.
Quando invece lo stato logico è "1", la tensione dipenderà dal microprocessore.
Tipicamente nei circuiti dell'elettronica digitale di tipo TTL lo stato logico 1 corrisponde alla tensione di alimentazione di 5 Volt.
Nei processori dei PC più avanzati, comunque, al fine di diminuire la dissipazione termica dei Chip si sono scelte tensioni molto più basse, persino inferiori ad 1 Volt.

I bit a loro volta vengono raggruppati in blocchi come dalla seguente tabella :

Sigla	***Descrizione***	***Combinazioni possibili***	***Numero Decimale rappresentato***
BYTE	Gruppo di 8 bit	2^8 = 256	da 0 a 255
WORD	Gruppo di 16 bit	2^{16} = 65536	da 0 a 65535
Double WORD	Gruppo di 32 bit	2^{32} = 4.294.967.296	da 0 a 4.294.967.295
Long WORD	Gruppo di 64 bit	2^{64} = ...	....

Questo però non è l'uso consueto che si fà dei dati contenuti all'interno di un microprocessore, sia che sia di un PC che sia in un PLC.

Al fine di chiarire questo concetto fondamentale, si danno degli esempi ***per far capire la differenza che si ha nei dati tra la rappresentazione binaria e quella realmente utile***.

Esempio 1

Nei PLC le Word sono tipicamente organizzate nel seguente modo

bit 15	bits 14 , 13 , 12 , 11 , 10 , 9 , 8 , 7 , 6 , 5 , 4 , 3 , 2 , 1 , 0
segno	valore

Ne consegue che una word viene a rappresentare un numero decimale intero nel range :

-32768 ... +32767

Questa variabile, anche se "fisicamente" si tratta di una Word, tipicamente è chiamata *Numero Intero* (o *Integer*).

Esempio 2

Una DoubleWord è un elemento nella memoria di un microprocessore costituito da 2 word, cioè 32 bit, e permette 2^{32} combinazioni, ossia da 0 a 4,29 x 10^9.
Nei PLC è tipicamente utilizzata per visualizzare un numero a virgola mobile, detto anche Floating, secondo lo standard americano IEEE-754.
In tal modo si rappresenta un numero nel range :

$$- 3{,}4028 \times 10^{-38} \quad ... \quad + 3{,}4028 \times 10^{38}$$

detto anche *Numero Reale* (o *Real*).

Esempio 3

Il Byte molto spesso è utilizzato per rappresentare dei caratteri alfanumerici.
A tal fine, agli albori dell'era informatica, è stata creata una ***tabella dei caratteri ASCII***, la quale esprime l'equivalenza di un byte con una lettera o un simbolo della tastiera.
Ad esempio se un byte vale 65 significa che rappresenta la lettera A.

Byte (8 bit)	0 1 0 0 0 0 0 1
Significato Decimale	65
Carattere rappresentato	"A"

Dato che i microprocessori dei PLC tipicamente funzionano a 16 bit si tratta *sempre* di informazioni organizzate come Word.

5.3 I tipi di dati standardizzati

Nella norma IEC 1131-3 si trovano oltre 20 tipi di tipologie di dati che possno essere gestite da un PLC, soprattutto per abbracciare un po' tutte le variabili usate dai vari produttori.
Possiamo grossomodo dividere le tipologie dei dati nelle seguenti tre categorie

1. Variabili Binarie

Data Type	***Descrizione***	***Range***
BOOL	un singolo bit del sistema binario	0 (False) , 1 (True)
BYTE	Gruppo binario di 8 bit	da 0 a 255
WORD	Gruppo binario di 16 bit	da 0 a 65535
Double WORD	Gruppo binario di 32 bit	da 0 a 4.294.967.295
Long WORD	Gruppo binario di 64 bit	.....

2. Variabili Numeriche

Data Type	***Descrizione***	***Range***
SINT	Short integer	-128 ÷ +127
USINT	Unsigned Short integer	0 ÷ 255
INT	Integer	-32768 ÷ +32767
UINT	Unsigned Integer	0 ÷ 65535
DINT	Double Integer	-2^{31} ÷ $+2^{31}$ (-1)
UDINT	Unsigned Double Integer	0 ÷ 2^{32} (-1)
LINT	Long Integer	-2^{63} ÷ $+2^{63}$ (-1)
ULINT	Unsigned Long Integer	0 ÷ 2^{64} (-1)
REAL	Floating Point Number	-10^{-38} ÷ $+10^{38}$
LREAL	Long real Number	-10^{-308} ÷ $+10^{308}$

3. Altri tipi di Variabili

Data Type	***Descrizione***	***Range***
DATE DATE_AND_TIME	Data del calendario Data e Ora	
TIME TIME_OF_DAY	Time (giorno e ora) Ora del giorno o anche detta Real Clock Time	
STRING	gruppo di caratteri (testo)	
R_EDGE F_EDGE	Fronte di Salita (rising) Fronte di discesa (falling)	0 (False) , 1 (True) idem

Questa panoramica è stata fatta per far vedere come esista una grande varietà di dati che i PLC possono gestire.
In seguito comunque si useranno quelli più siginificativi, a partire dal quelli necessari per realizzare i programmi di controllo di circuiti elettrici.

CAPITOLO 6.

Caratteristiche Software del PLC

6.1 Introduzione

Come già accennato il PLC è un piccolo computer, del quale eredita le caratteristiche principali.
E' quindi è dotato di un microprocessore, una memoria interna, un "sistema operativo", e di dispositivi di Input / Output.
Tipicamente i microprocessori dei PLC sono a 16 bit, e questo si riflette sulla struttura dei dati interna, come si potrà vedere in seguito.
Il set di istruzioni a disposizione di questi micoprocessori è generalmente molto vasto e potente, consentendo moltissime funzioni rispetto a quelle basilari usate per il controllo di un impianto elettrico.

6.2 Tipi di Processori

In linea generale tra i PLC si possono individuare due grandi famiglie :

a) i PLC dotati di microprocessori *multipurpose*, e quindi liberamente reperibili sul mercato, non espressamente progettati per questa funzionalità;
b) i PLC dotati di microprocessori *custom*, ossia progettati espressamente per essere integrati in un PLC e quindi per la gestione di un sistema elettrico di automazione.

La differenza tra i due mondi, in realtà è molto sottile, e tipicamente agli effetti pratici nel funzionamento non sortisce alcun effetto.
La prima soluzione comporta tipicamente un costo minore.
Nel secondo caso a fronte di un costo maggiore si hanno vantaggi notevoli in fase di programmazione, con istruzioni ottimizzate che fanno risparmiare tempo al programmatore e risorse di memoria nel PLC.

Questa seconda opzione è tipicamente seguita solo dalle grandi case produttrici, che possono permettersi di impiegare risprse per progettare un microprocessore adatto alle esigenze della loro clientela.

6.3 Set di Istruzioni

Come già accennato i microprocessori dei PLC sono dotati di un ampio Set di istruzioni. A prescindere dal fatto che tra marche diverse si trovano istruzioni diverse, si è cercato di fare un elenco che riproponesse le istruzioni in modo generalizzato al fine di fare una prima distinzione esemplificativa.

Diciamo che il nostro PLC Dummy è dotato del seguente set di istruzioni, suddiviso per gruppi, in modo da separare le istruzioni per funzionalià omogenee o per categorie. All'interno di ogni gruppo, poi, si trovano altre sottocategorie.

Gruppo	***Tipi di istruzioni***
A	Istruzioni BASE per il controllo di circuiti elettrici • Istruzioni in logica di Boole come AND/OR su singoli bit • Istruzioni supplementari come Temporizzatori e Contaimpulsi
B	Istruzioni per la Gestione del Programma • Gestione di Subroutine • Istruzioni di salto, Interrupt, ecc.
C	Istruzioni per la Manipolazione di Dati • Istruzioni per lo spostamento dei dati nella memoria, • Calcoli Booleani su Byte, word, • Operazioni di Confronto tra dati • ecc.
D	Istruzioni per la Manipolazione di Dati Numerici • Calcoli Matematici di Base, • Scalatura di Variabili e Unità di Misura, • Calcoli Scientifici • ecc.
E	Istruzioni per la Manipolazione di Dati Alfanumerici (stringhe)
F	Istruzioni per la gestione delle comunicazioni
G	Istruzioni per il controllo di variabili nei processi (PID)

Tipicamente le istruzioni del gruppo "A" sono sempre utilizzate, mentre per le istruzioni degli altri gruppi dipende dalla complessità dell'applicazione.

6.4 La memoria del PLC

6.4.1 Memoria Statica

Nei PLC di prima concezione la memoria era di tipo Statico, quindi all'utente erano assegnate delle aree "preconfezionate" con quantitativi fissi di dati memorizzabili all'interno, e che non potevano essere cambiati in alcun modo.
La memoria era rigidamente divisa tra dati e programmi ed anche tra tipi diversi di dati.
Un sistema di questo tipo non permette di personalizzare la memoria a seconda dell'applicazione, e soprattutto accadeva spesso che molte aree dati restavano inutilizzate a scapito di altre aree che avevano la necessità di ampliamento.
Anche il programma, sebbene potesse venir suddiviso in subroutine, in realtà appariva come un unica sequenza di istruzioni, creando non poca confusione.

Per fare un paragone con l'ambiente PC, è un po' come se nel nostro PC tutti i documenti Word fossero concentrati in un unico file.
Un sistema del genere, anche se non influisce nel funzionamento del PLC, non permette una gestione efficente e chiara dell'intero progetto.

Questo sistema è ancora ampiamente usato per piccoli PLC con applicazioni semplici e pochi ingressi e uscite da gestire, dove la gestione della memoria in un modo piuttosto che in un altro non influisce nel risultato.

6.4.2 Memoria Dinamica

Nei PLC più moderni la memoria interna è di tipo *Dinamico*, e può essere considerata come un Hard Disk di un Personal Computer, con file di dati e file di programmi.
In fase di caricamento del software applicativo il PLC alloca dinamicamente la memoria a seconda delle risorse che sono richieste.
A parte certe aree di sistema, ed alcune aree preconfezionate al fine di facilitare la programmazione ai meno esperti, la memoria per il resto è liberamente configurabile.
Per le aree di memoria già pronte c'è da dire che comunque sono quelle maggiormente usate per la stesura dei programmi.
Per il resto si possono configurare nuove aree dati, anche dello stesso tipo, ma usate da sottoprogrammi diversi.
Ogni area dati può avere un nome, denominato Nome File, e tipicamente ogni area al suo interno può avere fino a 256 elementi.
Per i programmi se ne possono creare a piacere (tipicamente fino a 256 subroutine), con l'unico limite se non quello dovuto alla dimensione massima della memoria fisica.
Il file di programma già pronto viene di solito denominato *Main Program*.

> ***Nota :*** *consideramo quindi che il nostro PLC "Dummy" sia dotato di memoria Dinamica.*

6.5 Organizzazione interna dei Dati

I dati memorizzabili nei PLC possono essere di vario tipo, ma in realtà al loro interno si tratta sempre di informazioni Binarie.
Tale caratteristica è comunque invisibile al programmatore, che vede solo i dati nel formato da lui prescelto.

6.5.1 File di Input / Output

Dato che il PLC è un dispositivo che dialoga con l'esterno, vi saranno anche due file appositamente destinati a tale attività, relativa ai moduli di Ingresso e Uscita.
I dati di ingesso ed uscita sono infatti acquisiti dal microprocessore in momenti distinti dall'esecuzione del programma e conservati in questi due file detti anche *File di Immagine degli Ingressi* e *File di Immagine delle Uscite*.

nr. File	***Sigla File***	***Tipo di Dati***	***Composizone***	*Altre Silgle*
0	**O**	Uscite	1 bit per ogni segnale digitale 1 word per ogni segnale analogico	**Q , Y**
1	**I**	ingressi	(idem)	**X**

6.5.2 File di Sistema

In quest'area di memoria, di tipo statico e non facente parte delle aree programmabili, vi sono memorizzate la configurazione del PLC ed altri parametri necessari per il suo funzionamento.

nr. File	***Sigla File***	***Tipo di Dati***	***Composizone***
2	**S**	Registro di Sistema	Dati di configurazione del PLC

La conoscenza di quest'area di memoria al programmatore è indispensabile in quanto in essa si trovano i bit di errore, le configurazione delle porte seriali e di programmazione, i bit di allarme e di overflow nei calcoli matematici.

Come già per altri casi si rimanda alla specifica documentazione del PLC.

6.5.3 File di Dati Interni

All'interno di ogni PLC vi sono dei file dati molto utili nella programmazione e che svolgono funzioni virtuali dei componenti usati nell'elettrotecnica, quali Temporizzatori, Contaimpulsi e Relè.
Naturalmente questi elementi cambiano a seconda del PLC utilizzato, ma sono quasi tutti presenti in tutti i modelli, anche se con nomi diversi.

Ipotizziamo quindi che il nostro PLC Dummy sia dotato dei seguenti File Dati :

nr. File	***Sigla File***	***Tipo di Dati***	***Composizone***
3	**B**	Bit , detti anche *Relè Interni*	1 File con 256 word = 4096 bit
4	**T**	Timer (temporizzatori)	1 File = 256 timer
5	**C**	Counter (Contatrori)	1 File = 256 Counter
7	**N**	Numeri Interi (anche *Integer*)	1 File = 256 word (dati numerici)
8	**F**	Floating - numero in virgola mobile (o anche *Real*)	1 File con 256 dati F
9	**ST**	Stringhe ASCII (String)	1 File = 256 Stringhe di caratteri

L'uso pratico di questi files verrà affrontato nei capitoli successivi.

File Speciali

Per alcune attività sono necessari dei file di dati strutturati in modo particolare, come ad esempio i file per gestire le comunicazioni o per la manipolazione di stringhe ASCII.
Dato che tali formati variano a seconda della marca, in fase di programmazione sarà necessario approfondire se le istruzioni utilizzate fanno ricorso a file speciali.

6.6 Le aree di Memoria Retentive

Nei PLC tipicamente vi sono delle zone della memoria nelle quali i dati restano memorizzati anche al mancare della tensione di alimentazione.
In tal caso, per distinguerle dalle altre, queste memorie vengono dette *Retentive*.

> ***Nota :*** *al fine di evitare confusione, e sfruttando le caratteristiche dei PLC più moderni, consideramo che tutta la memoria del PLC "Dummy" sia retentiva.*

6.7 Indirizzamento

Una delle questioni di fondamentale importanza quando si approccia ad un PLC, anche per gli esperti programmatori, è quella che bisogna conoscere il ***modo di idirizzamento***. Con questo termine si intende la sintassi con la quale il programma accedere ai dati nella memoria o nei dispositivi di Ingresso e Uscita.

Per fare un esempio utilizziamo il nostro PLC Dummy già preparato in precedenza :

0	***1***	***2***	***3***	***4***	***5***	***6***
CPU	16 in Digitali	8 in Digitali	16 Out Digitali	8 Out Digitali	8 In Analog.	4 Out Analog.

Segnali Digitali

Per i segnali di tipo digitale si trova la seguente sintassi :

- I:1/3 segnale di ingresso 3 (bit) nel modulo 1
- I:2.0 word di ingresso di tutti i bit del modulo 2
- O:3/12 segnale di uscita 12 del modulo 3
- O:4.0 word di uscita di tutti i bit del modulo 4

Segnali Analogici

I segnali analogici, che praticamente sono delle word, la sintassi potrebbe essere:

- I:5.0 word dell'ingresso Analogico 0 nel modulo 5
- I:5.7 word dell'ingresso Analogico 7 nel modulo 5
- O:6.1 word dell'uscita Analogica 1 del Modulo 6

L'indirizzamento vale per tutti i file di dati (Ingressi, Uscite, e per i file interni elencati nei paragrafi precedenti); Ad esempio :

- T4:1 Timer 1 del File di temporizzatori nr. 4
- C5:3 Counter 3 del File di contatori (contaimpulsi) nr. 5
- N7:9 Numero intero posto nella Word 9 del file di interi 7

Nota : L'indirizzamento usato dalla norma IEC ha una sintassi similare a quella esposta, ma più complessa (ad esempio I:1/3 diventa I%1.0.0.3), e poco usata.

CAPITOLO 7.

Obbiettivi Principali e Criteri di Sicurezza

7.1 Introduzione

Prima di affrontare i temi della stesura vera e propria di un programma è opportuno fare una pausa di riflessione.
Dato che il sistema di controllo causerà l'azionamento di macchinari e meccanismi è necessario rendersi conto dei pericoli e delle responsabilità alle quali si và incontro.

> *Nota*
> Le nozioni "ufficiali" dovranno comunque essere consultate sulle relative norme. A tal proposito si consiglia di consultare la norma CEI EN 60204-1, oppure i libri indicati nella bibliografia.

Questo capitolo affronta il primo passo da compiere prima di iniziare veramente la programmazione di un PLC : la sicurezza e le responsabilità.

7.2 Obbiettivi

Il sistema che controlla una macchina deve essere attentamente progettato in quanto si devono perseguire i seguenti obbiettivi primari :

- ***la sicurezza delle persone,***
- ***la sicurezza contro danni alle cose,***
- ***il buon funzionamento della macchina al fine di dare il risultato richiesto,***
- ***l'affidabilità e l'immunità agli errori,***
- ***la comprensibilità del programma.***

Al fine di raggiungere tali obbiettivi il programmatore del PLC non può essere all'oscuro dalla parte elettrica della macchina, anzi deve conoscerla al 110% e possibilmente deve partecipare alla progettazione per suggerire parti che potrebbero poi tornargli utili in fase di programmazione.

7.3 Sicurezza

Come indicato negli obbiettivi, il tema Sicurezza deve essere al primo posto quando si progetta il sistema di controllo di una macchina.
Anche se la sicurezza, ai sensi della norma EN 60204-1, deve essere gestita obbligatoriamente da logiche cablate, e quindi non dal PLC, in ogni caso il programma non può essere concepito senza rispettare le regole dettate dal buon senso.
In particolare sarà necessario fare le dovute riflessioni sui rischi che la macchina può provocare nel caso di manomissione dei dispositivi di sicurezza, o comunque nel caso vengano usati in modo "distorto".

I Sistemi di comando a due mani

Alcune macchine per lavorazione meccaniche, prive di ripari, per l'avviamento sono dotate di due pulsanti distinti e distanti tra loro in modo da costringere l'operatore ad avviarle con entrambe le mani.
In questo modo le mani durante il funzionamento sono lontane dall'utensile.
Per evitare che l'operatore faccia il furbo, ad esempio incastrando uno dei due pulsanti (ad esempio con un peso o con una zeppa di legno !), basta un piccolo accorgimento nel software del PLC :
La macchina si avvia solo se i due pulsanti vengono premuti in un determinato istante, ed entro un tempo massimo (ad esempio 0,5 secondi).
Questo impedisce all'operatore di aggirare le sicurezze e di rischiare la propria salute.

Il programma nel PLC, quindi, dovrà essere concepito con qualche piccolo accorgimento in modo che sia immune da tali manomissioni.
Dato che il programmatore è il responsabile del funzionamento della macchina, e non può prescindere dalla parte elettrica alla quale è collegato il PLC.
La stesura di software intrinsecamente sicuro comporta un grande vantaggio per il programmatore : nessuno potrà scaricare la colpa sulla parte software se la macchina si rende pericolosa in determinate circostanze.

7.4 Sicurezza contro i danni alle cose

In questa categoria rientrano i danni dovuti a movimenti errati del macchinario che potrebbero provocare un urto tra parti meccaniche, un incastro, o un cortocircuito.
Il programma deve quindi essere sempre dotato di accorgimenti interni che impediscano queste situazioni, a prescindere da come è concepita la parte elettrica a valle del PLC.
Il caso tipico è sul cablaggio di un motore con il comando avanti / indietro, che deve essere provvisto di un interblocco elettrico che impedisca la contemporanea attivazione di entrambi i comandi (cosa che provocherebbe un bel cortocircuito).
Questa sicurezza è tipicamente realizzata in sede di cablaggio, ma che dire se sbatatamente l'elettricista l'ha collegata in modo sbagliato ?
E' quindi erroneo demandare ad altri questioni che possono essere risolte semplicemente all'interno del software e che ne aumentano l'affidabilità.

Un breve riassunto di alcuni accorgimenti da attuare :

- interblocchi che non accettano il comando di più pulsanti contemporaneamente con funzioni diverse o addirittura opposte (ad esempio Sollevamento + Discesa).
- interblocchi che escludano i comandi manuali quando una macchina sta funzionando secondo un ciclo automatico
- interblocchi che impediscano al software stesso di attivare una sequenza mentre è già in atto quella contraria, o comunque ne sta svolgendo una diversa e meccanicamente non eseguibile in contemporanea ;
- ecc.

Da questi accorgimenti si deduce che il programmatore del PLC dovrà essere edotto su tutte le caratteristiche meccaniche della macchina, su cosa può fare e soprattutto cosa non deve fare.

7.6 Ridondanza degli ingressi

Vi sono casi particolari dove ai sistemi di controllo sono richieste prestazioni di sicurezza molto spinte ed una una elevata continuità di servizio, e nei quali viene richiesta la *Ridondanza* degli ingressi.
Sono casi che si trovano ad esempio negli impianti chimici, navali, piattaforme petrolifere, centrali termoelettriche, ecc.
La ridondanza si realizza collegando al PLC i due contatti di ogni componente, ovvero sia quello normalmente aperto (NA) che quello normalmente chiuso (NC).
In questo modo il PLC può capire, oltre alla tradizionale posizione del contatto, se esso ha il cavo interrotto (entrambi i contatti aperti) o se il cavo è in cortocircuito (entrambi i contatti chiusi).
Questo sistema implica una programmazione molto più sofisticata, dato che il programma deve prendere una decisione "speciale" (che non esiste nei normali circuiti cablati), quando riscontra un guasto su un componente.

7.5 Regole di cablaggio degli Ingressi Digitali

Quando si collega un componente elettromeccanico ad un circuito elettrico, ad esempio un finecorsa, questo tipicamente ha solo uno o al massimo due contatti a disposizione.
Per i componenti di tipo "elettronico" come ad esempio fotocellule e sensori di prossimità, tipicamente non esiste questa possibilità, in quanto hanno un unica uscita.
In ogni caso anche se un componente è dotato di più contatti, tipicamente se ne collega al PLC solamente uno.
Tra l'altro, come si vedrà nei capitoli relativi alla programmazione, all'interno del programma è come se il finecorsa avesse un numero di contatti infinito, dandoci la possibilità di utilizzarlo per più funzioni.

In ogni caso quando si progetta l'impianto si devono fare delle considerazioni di sicurezza, cosicchè l'uso dei contatti per i componenti in ingresso al PLC potrebbe essere ad esempio :

- Per i pulsanti "Marcia" si utilizza il contatto Normalmente Aperto
- Per i pulsanti "Arresto" si utilizza il contatto Normalmente Chiuso
- Per i finecorsa di arresto si utilizza il contatto Normalmente Chiuso

Questo vale per i componenti "ordinari" della macchina.
Per i componenti di sicurezza quali relè termici, finecorsa dei ripari, pulsanti di emergenza ed altri dispositivi, la situazione è invece diversa.
Per questi componenti il collegamento al PLC è opzionale, e può essere utile solo per la ricerca delle cause di fermo macchina, non per il funzionamento vero e proprio.

Da ciò si desume la seguente regola :

- *Quando un relè, un finecorsa o un qualunque altro dispositivo è utilizzato per funzioni di sicurezza, per portare il segnale al PLC si utilizza il contatto Normalmente Aperto, in quanto quello Chiuso è cablato con il circuito di sicurezza.*

7.7 Comprensibilità della Documentazione

L'aspetto documentale della programmazione dei PLC è spesso trascurato e da esso si comprende il grado qualitativo del progetto eseguito a monte.
Mi è capitato spesso di vedere programmi funzionanti, ma senza un minimo di commenti o note, e comunque strutturati secondo una logica avulsa, senza possibilità alcuna di comprensione.

Un programma organizzato in tal modo tra qualche anno metterà in crisi lo stesso sviluppatore, in quanto non si ricorderà più nulla di come lo ha concepito.

Un terzo che dovesse necessariamente apportare delle modifiche al quel programma sarebbe praticamente impossibilitato, a meno che non abbia a disposizione giorni e giorni per interpretarlo e comprendere dove introdurre la modifica.

Tipicamente in questi casi il tempo di comprensione del programma supera il tempo che sarebbe necessario per riscriverlo.

Questo aspetto è maggiormente importante in tutti quei casi in cui il macchinario dotato di PLC viene venduto all'estero o comunque trasportato in luoghi molto lontani da dove è stato costruito.

Se il programma viene poi preso in consegna da una terza persona, questa sarà facilitata nell'apprendimento se il programma è organizzato in modo schematico.

	Gli elementi principali per documentare un programma sono :
1.	Assegnare una sigla e/o un nome ad ogni segnale di ingresso
2.	Assegnare una sigla e/o un nome ad ogni segnale di uscita
3.	Commentare le righe del programma, almeno ogni qualvolta si inizia una nuova funzione o si comanda un diverso dispositivo.
4.	Commentare ciascuna Subroutine (Sottoprogramma) all'inizio, con una sintetica descrizione dei compiti che esegue.

Per i punti 1 & 2 si raccomanda che le sigle o le descrizioni inserite siano eguali a quanto indicato per quel segnale negli schemi elettrici

7.8 Programmi e Programmatori

I PLC contengono del software, e come tale questo dovrebbe essere soggetto alla legge dei diritti di autore (argomento comunque da approfondire !).

A protezione del proprio software si possono usare vari sistemi, come la password nella CPU che impedisce qualunque accesso (in questo modo ci si mette al sicuro da manomissioni e da copie non autorizzate).

A protezione della propria responsabilità invece sarebbe meglio consegnare la stampa del programma firmata al cliente al termine del collaudo : in questo modo ci si mette al sicuro da future modifiche eseguite da terzi.

Un programmatore dovrebbe riflettere attentamente prima di modificare un programma eseguito da altri : fondamentalmente si diventerebbe responsabili di tutto il comportamento di un software che non conosce perfettamente.

Meglio quindi pensarci due volte, a meno che non si tratti di modifiche veramente minime, come ad esempio per correggere errori di "battitura" che aveva il programma o i cosiddetti "cambio formato" nelle linee di produzione.
In altri casi meglio riscrivere tutto il software da zero .

CAPITOLO 8.

La Progettazione del Programma

8.1 Introduzione

Dopo aver trattato alcuni cenni storici ed aver presentato brevemente le caratteristiche hardware del PLC, ***si presentano ora due argomenti ben distinti :***

1. La progettazione del programma, una attività che, in linea generale, è tale e quale alla progettazione dello *Schema Funzionale* di un circuito di comando.
2. La configurazione e programmazione del PLC dal punto di vista dell'uso del software di programmazione, e quindi l'uso della sua memoria e delle istruzioni;

In questo capitolo si affronterà l'argomento progettazione, mentre il secondo aspetto verrà analizzato nel capitoli successivi (che comunque deve essere sempre approfondito con i manuali forniti dal costruttore del PLC).

8.2 Principi generali di Progettazione

Come visto finora, la stesura del software per un PLC ha bisogno di un quantitativo di carta al quadrato rispetto al listato finale del programma.

Sulla base di quanto detto fino ad ora, buona parte di questa carta è costituita da :

1. documentazione che è indispensabile consultare (Norme CEI, Direttiva Macchine, libri, ecc),
2. Disegni meccanici o lo schema di processo (P&ID)
3. Schemi elettrici
4. Caratteristiche della macchina, prestazioni richieste, organi di sicurezza, elenco motori ed utenze, ecc
5. Documento con elencate le modalità di funzionamento richieste e le caratteristiche di sicurezza che dovranno essere rispettate.

Giunti a questo punto si può cominciare a tracciare lo schema a blocchi del sistema di controllo, ossia del programma.

Non si tratta di un flowchart, ma piuttosto di un layout che dà l'impostazione generale del programma, ossia la struttura.
Il layout sarà graficamente composto da più blocchi ognuno dedicato ad una appropriata funzione, e che nella realtà finale del progetto saranno dei sottoprogrammi (subroutine).

Quando si trascrive un programma per un piccolo impianto, infatti, dato l'esiguo numero di segnali, è possibile avere sott'occhio (e a mente), tutto il sistema controllato.
Al contrario, nel caso di grosse macchine o impianti estesi ciò è praticamente impossibile, ***quindi è necessario suddividerli in unità elementari***, che riconducano al caso dell'impianto semplice.
Ogni subroutine avrà quindi al proprio interno delle logiche autonome inerenti la funzione che deve svolgere.
Questa attività viene chiamata *Scomposizione Funzionale*.

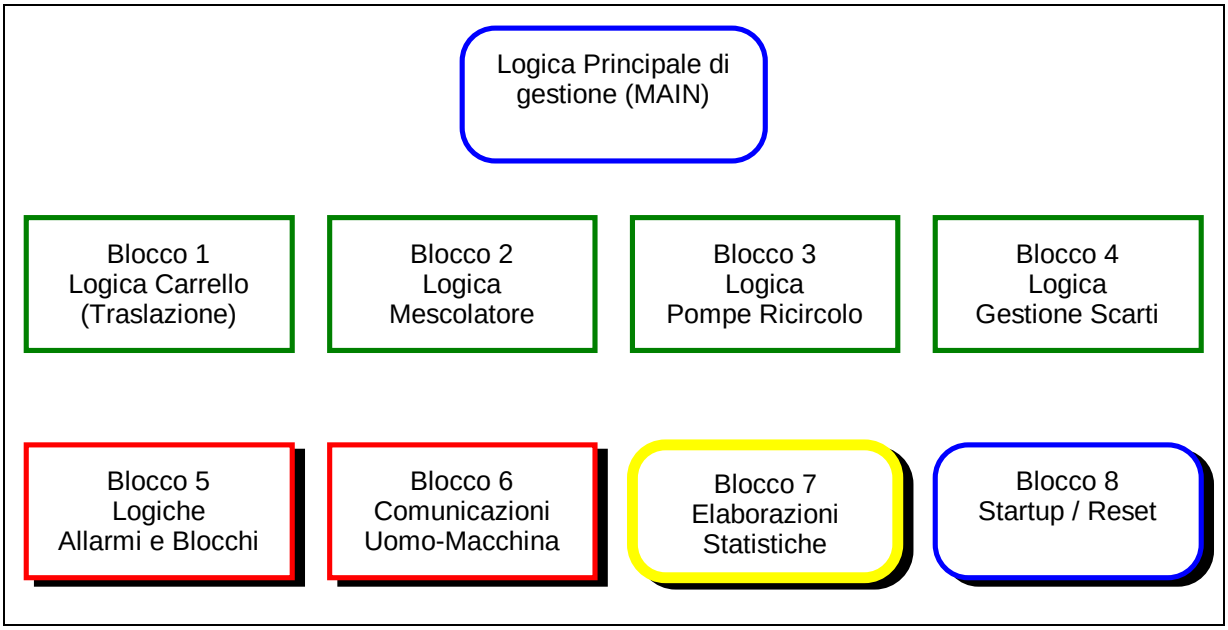


Figura 1 - Esempio di organizzazione del sistema di controllo di una ipotetica macchina.

Tale suddivisione potrebbe risultare esagerata per un PLC con 10 ingressi e 10 uscite, ma non è detto.
La strutturazione del progetto consente infatti una più agevole ricerca guasti e comprensibilità dello stesso ad una terza persona, come affrontato nel paragrafo "Comprensibilità e Documentazione".
La suddivisione del programma i più blocchi "logici" consente inoltre di aumentarne l'affidabilità, dato che durante la stesura del programma sarà più semplice evitare interferenze o errori di battitura tra righe destinate a compiti diversi.

8.3 Analisi della macchina

Come è visto nell'esempio precedente, per costruire il Layout di un sistema di controllo (ossia del programma), è necessario avere ben chiari tutti i componenti dell'impianto che si deve a controllare, e tutte le funzionalità che si dovranno realizzare.
Ecco che quindi è necessario analizzare ciascuna macchina con il relativo "processista" che l'ha concepita per comprenderne tutti gli aspetti che influiscono il modo di funzionamento e quini la programmazione.

Una volta tracciato il Layout, questo potrebbe essere usato come documento ufficiale.
Ad esempio se viene tralasciato di informare il programmatore che e' necessario gestire con il PLC gli scarti di produzione in una determinata maniera, il blocco 4 non figurerebbe nel Layout, ed il progettista meccanico, osservando tale diagramma, si accorgerebbe della incompletezza.
Il Layout poi non deve essere obbligatoriamente fatto in modo grafico, ma può essere trascritto anche sotto la forma di un elenco delle funzioni da realizzare.

8.4 Modalità di Funzionamento

In fase di progettazione dovranno essere ben chiarite quali saranno le modalità di funzionamento della macchina o dell'impianto.
Ad esempio si dovranno descrivere i modi di funzionamento manuale e automatico, che tra l'altro tipicamente non sono compatibili tra loro.
Dovrà quindi essere documentato come funziona la macchina a seconda della posizione del *Selettore Modale di Funzionamento* (EN 60204-1, articolo 9.2.3).
Questo selettore, gerarchicamente posizionato al di sotto del famigerato *Pulsante di Emergenza*, in realtà prevale su tutti gli altri comandi.
Per contro il cambiamento della sua posizione non deve provocare nessuna azione nella macchina, ma solo la preselezione della logica nel programma.

8.5 La Logica Principale

In questa logica, detta anche Main e talvolta anche Blocco Organizzativo, si gestiscono tutte le altre subroutine e le funzioni di particolare importanza per tutto l'impianto controllato.
Per il resto il programma principale dovrebbe essere composto da chiamate alle subroutine, condizionate o meno dalla logica del macchinario.

8.5.1 Consenso al Funzionamento

Per una macchina, nella logica principale le prime operazioni da affrontare saranno quelle inerenti i controlli sulla sicurezza :

- Controllo se tutti gli interruttori di alimentazione ed i relè termici sono OK.
- Controllo se tutti i parametri di funzionamento sono entro i valori prestabiliti (temperature, pressioni, tara, ecc).
- Controllo se tutte le sicurezze sono in ordine (ripari chiusi, barriere posizionate, ecc), compreso il Pulante di Emergenza.

Solo con il benestare di tutte queste logiche si potrà procedere all'avviamento della macchina.
Il relè che riassume il benestare a procedere può essere chiamato KF, in quanto è da considerarsi il *Consenso al Funzionamento*.
Il relè che riassume la presenza regolare di tutte le alimentazioni, invece, tipicamente è chiamato *relè di Linea* o KL.

8.5.2 Funzioni al Power-on

Si potrebbe cominciare a ragionare sul programma con considerare il momento dell'accensione di una macchina e del relativo PLC.
In questa fase può essere necessario lanciare procedure di pre-avviamento indispensabili per il buon funzionamento della macchina.

Si cita ad esempio :
- il riscaldamento dell'olio nelle centraline oleodinamiche,
- il ricambio dell'aria nel vano lavorazione,
- l'apertura delle elettrovalvole dei circuiti dell'aria
- l'accensione della pompa di circolazione dell'acqua di raffreddamento,
- ecc.

Anche per questi casi è necessario discutere con il progettista meccanico della macchina o con il processista come si deve procedere

8.5.3 Settaggi al Power-On

All'accensione di un PLC tipicamente sono necessarie delle procedure di set-up, per cui è meglio prevedere un sottoprogramma apposito.
Tale subroutine verrà richiamata dalla logica principale solo all'accensione.
Tipicamente si resettano tutti i temporizzatori, i contatori, ecc. , semprechè la macchina non abbia bisogno di una funzione di "ripartenza a freddo".
In questa fase può essere utile ricaricare in memoria i set-point di funzionamento, quali misure, conteggio pezzi, ecc. che potrebbero essere memorizzati invece nel terminale operatore (interfaccia uomo-macchina), esterno al PLC.

8.6 Black-Out

Questo è un'altro argomento che spesso viene tralasciato: lo spegnimento improvviso.
Forse dipende dal fatto che si considera la mancanza di energia elettrica un fatto raro, e che comunque non incide nella programmazione dato che "in quei momenti" il programma è praticamente morto con tutta la macchina che deve controllare.
Al contrario anche il black-out è un episodio che andrebbe considerato in fase di programmazione.

Le tipiche fasi di black-out di un macchinario

1. In una piccola industria l'operatore addetto ad un macchinario lo sta facendo funzionare in serenità e pace e questo gli corrisponde producendo i prodotti che deve.
2. A migliaia di chilometri di distanza, in Svizzera, all'improvviso un albero cade su di una linea aerea, mandandola fuori servizio;
3. Inspiegabilmente, come conseguenza del guasto, nella piccola industria manca l'energia elettrica;
4. Si spengono tutte le macchine, i computer perdono tutti i dati e i telefoni si zittiscono all'improvviso.
5. Dall'ufficio del capo si sente arrivare un ululato bestiale;
6. L'operatore della macchina, dopo un momento di sbigottimento, si rende conto che e' entrato in ferie anche senza volerlo.
7. Dopo alcuni minuti, l'energia elettrica ritorna come se niente fosse
8. A quel punto il nostro operatore alla macchina preme START, ma ...non succede nulla!
9. Dopo vari tentativi ed imprecazioni non resta che eliminare il prodotto semilavorato e ricominciarne uno da zero, con 2 ore di lavoro perse.
10. Quando la notizia arriva nell'ufficio del capo, si sente un secondo ululato ancora più bestiale del primo.

Dato che oggi praticamente tutti i PLC sono dotati di memorie "a rimanenza", nelle quali i dati restano memorizzati anche in mancanza dell'energia elettrica, utilizzando queste aree di memoria, talvolta è possibile "fotografare" la situazione del macchinario nell'ultimo instante prima della mancanza di energia elettrica, cosicchè al ritorno dell'energia è possibile riprendere il processo dal punto che era rimasto.

Deve sempre e comunque essere possibile cancellare queste memorie, laddove la situazione sia da ricominciare da zero, cosa attuabile con un apposito pulsante di reset, che tipicamente viene previsto nella consolle di comando della macchina.

Talvolta è utile discutere di un possibile black-out, in quanto gli ideatori della macchina si accorgono se è o meno necessario dotare la macchina di gruppi di continuità, per garantire la sicurezza in queste fasi (ad esempio per la frenatura, ecc).

CAPITOLO 9.

Componenti Elettromeccanici Virtuali

9.1 Generalità

All'interno di un PLC, in pratica, con il programma è come se si creassero dei componenti elettromeccanici virtuali.
In fase di programmazione si possno collegare, al pari dei circuiti cablati, relè, contatti aperti, contatti chiusi, temporizzatori, contaimpulsi ed altri componenti.
Tali componenti nella realtà sono elementi logici nella del PLC, ma il programmatore nello schermo del suo PC di configurazione continua a vedere uno schema elettrico.
E' proprio tale software che effettua la conversione da grafico a lista di istruzioni per il microprocessore (programma), ma questa operazione è comunque invisibile ed automatica.

In questo capitolo si affronteranno i componenti di maggior uso nella programmazione dei PLC, e che di fatto sono quelli di maggior uso nei circuiti elettrici.

9.2 Contatti illimitati

Prima di iniziare facciamo una breve precisazione, molto utile.
Tra i vari vantaggi che vi sono nell'usare un PLC, vi è la possibilità pressochè illimitata di usare il contatto di un relè o di un qualunque altro elemento in memoria.
Per spiegare meglio questo concetto prendiamo come esempio un relè vero e proprio : questo tipicamente ha un numero ben definito di contatti :

- Ad esempio vi sono i relè a 2 scambi ed i relè a 3 scambi.

In questo modo quando si sono collegati tutti i contatti è necessario aggiungere un secondo relè in parallelo al primo, e così via.

Nei PLC invece ***non esiste questa limitazione***, ed è possibile utilizzare ogni elemento come se avesse un numero infinito di contatti, dandoci la possibilità di utilizzare lo stesso segnale per parecchie funzioni.

Questa caratteristica è dovuta dal fatto che il microprocessore può interrogare un numero infinito di volte la sua memoria per vedere se un bit è nello stato 1 (ossia eccitato) o nello stato 0 (ossia diseccitato).

Non vi sono quindi limiti nell'utilizzo dei contatti aperti o chiusi di una qualunque "entità" presente nel PLC.

Nota : per entità si intende un qualunque bit di un qualunque file di dati.

9.3 INGRESSI DIGITALI

Come già detto gli ingressi costituiscono gli "occhi" del PLC, in quanto è con essi che il programma si rende conto di quello che sta succedendo nel macchinario controllato.
Gli ingressi sono quei morsetti del PLC ai quali vengono collegati componenti che al loro interno hanno un semplice contatto.
In realtà sarebbe più corretto chiamarli Ingressi Digitali, o DI, dall'inglese *Digital Inputs*, anche per distingueri dagli ingressi analogici.
Invece nel gergo, vuoi per semplicità o vuoi per il loro uso prevalente, quando ci si riferisce ad essi nei PLC li si indica semplicemente come "ingressi".

Durante la prorgammazione ***gli ingressi digitali vengono sempre visualizzati come contatti***, anche se nella realtà rappresentano componenti molto diversi tra loco, come Pulsanti, Finecorsa, Relè Termici, Fotocellule, Sensori di Prossimità, ecc.

9.3.1 Indirizzamento

Come già visto, nella fase di stesura del programma per individuare un qualunque segnale è necessario utilizzare il suo indirizzo.
Ipotizziamo ora di collegare quattro segnali in ingresso al nostro PLC "Dummy", che avranno i seguenti indirizzi :

Indirizzo	*Descizione*	*Note*
I:1/10	Pulsante Start	Collegato all'ingresso 0 nel modulo 1
I:1/11	Pulsante Stop	Collegato all'ingresso 1 nel modulo 1
I:2/6	Finecorsa Fase 1 Avanti	Collegato all'ingresso 6 nel modulo 2
I:2/7	Finecorsa Fase 1 Indietro	Collegato all'ingresso 7 nel modulo 2

Questo esempio sarà utile in seguito per costruire un piccolo programma che gestira' una sequenza automatica.

9.4 RELE' INTERNI

Tra le varie aree dati all'interno di un PLC, come abbiamo visto nel paragrafo "6.5.3 - File di Dati Interni", vi è quella dei cosiddetti Relè interni.
Dato che in inglese il relè interno si dice *Internal Relay* , in alcuni PLC questi elementi vengono indicati con la sigla "IR".

> *Nota*
> Nella Norma IEC invece è stato scelto di chiamarli *Bool* (booleani), concetto un po' troppo matematico per usarlo nell'automazione !

Come già anticipato, i relè interni al PLC sono bit posti su delle Word, quindi un file di relè è composto da elementi con 16 "relè" ciascuno.

9.4.1 Indirizzamento

Esempi di indirizzamento dei relè interni del nostro PLC "Dummy" sono :

- **B3:0/5** bit 5 della Word 0 del file B3
- **B3:5/14** bit 14 della Word 5 del file B3
- **B11:2/9** bit 9 della Word 2 del file B11

In alcuni casi è possibile configurare il software di programmazione in modo che a video la numerazione dei relè interni sia da 0 a 4095.
Per chiarire questo aspetto si sono utilizzati gli stessi bit l'esempio precedente, che però, come si vede, hanno un sistema di identificazione più semplice :

- B3/5 (= Word 0 * 16 + 5)
- B3/94 (= Word 5 * 16 + 14)
- B11/41 (= Word 2 * 16 + 9)

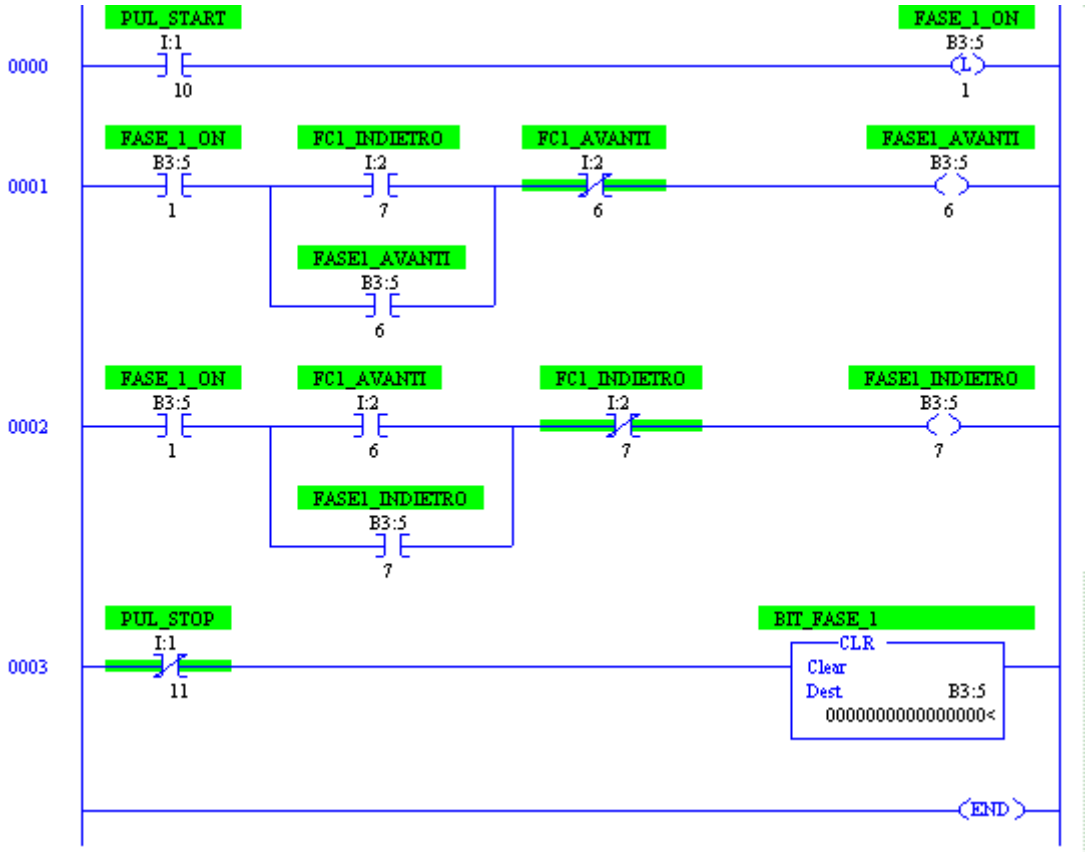
In pratica il PC nel rappresentare il relè interno calcola il suo numero sommando anche tutti i bi delle word precedenti, alla word dove è posizionato.

9.4.2 Pianificazione

In realtà il sapere che i Relè interni nel PLC sono organizzati per Word può essere molto utile in fase di programmazione.
Dato che si è stabilito di controllare ciascuna fase di funzionamento dell'impianto tramite subroutine (sottoprogrammi), ***sarà utile assegnare ad ogniuna di esse una o più Word di relè interni***.
Questo costituisce un modo di procedere ordinato nella stesura del programma, ma tali word saranno molto utili per l'arresto di quella fase o nel reset della memoria.

ESEMPIO

L'esempio che segue è di un programma che gestisce la "Fase 1" di una ipotetica macchina.
La fase di lavoro viene avviata dall'operatore tramite il pulsante Start, e viene fermata tramite il pulsante Stop (che come da premesse è normalmente chiuso).
Le varie sequenze di lavoro sono gestite da relè interni al PLC (la gestione delle uscite si vedrà più avanti).



Nota
Per l'arresto, invece di resettare ogni bit singolramente, si è usata una istruzione speciale, chiamata CLEAR, che azzera completamente l'intera word, e che quindi "diseccita" tutti i suoi relè in un colpo solo.

9.5 TIMER

Nei circuiti di controllo il temporizzatore è il secondo elemento più usato dopo il relè.
Tale componente è presente anche nei PLC con il nome di TIMER, e dà la possibilità a chi scrive il programma di scegliere il range di funzionamento con molta elasticità.
Ad esempio è possibile impostare come tempo di lavoro 3 secondi, ma anche 30 millisecondi.

Nella memoria dati del PLC un Timer occupa tre Word :

- Preset è la word che contiene il valore prestabilito (anche detto Set-Point)
- Acc è la word che il PLC incrementa col passare del tempo (Accumulatore).
- Bit di stato danno lo stato di funzionamento del timer

Base dei Tempi

Il range di lavoro di un Timer tipicamente è da 0 a 32767, ma questo valore ha un significato diverso a seconda della base dei tempi che si sceglie.
Al fine di determinarne la precisione quindi, in fase di programmazione è possibile, e anche indispensabile, scegliere la base dei tempi con cui lavora il timer stesso :

Base dei Tempi	***Range di lavoro***	***Precisione***
1 secondo	0 - 32767 secondi	1 secondo
0,1 secondi	0 - 32767 decimi di sec.	1/10 secondo
0,01 secondi	0 - 32767 centesimi di sec.	1/100 secondo
0,001 secondi	0 - 32767 millesimi di sec.	1 ms

Nota : alcuni PLC lavorano contando alla rovescia dal Preset a zero.

Tipologie

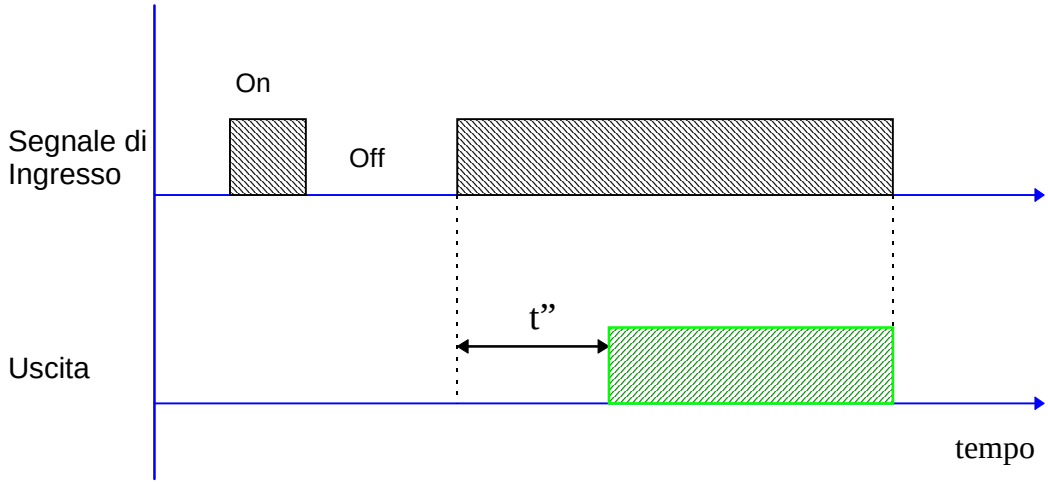
Nel PLC sono disponibili due “istruzioni timer”, che funzionano in modo opposto :

- Timer ritardato all’eccitazione sigla **TON**
- Timer ritardato alla diseccitazione sigla **TOF**

Nel paragrafo seguente si analizzeranno questi due timer e ne verrà spiegato il funzionamento.

9.5.1 Timer ON Delay

Questo temporizzatore eccita la sua uscita solo se il contatto di ingresso rimane attivato per almeno un certo tempo, dopodichè l'uscita del timer si attiva fino a quando non viene rilasciato il contatto di ingresso.



Come si vede nel grafico al primo segnale di "start" proveniente dal contatto di ingresso il temporizzatore non risponde, in quanto la durata del segnale è inferiore al tempo di Preset.
Nel secondo caso trascorso il tempo "t", l'uscita del PLC si attiva, finchè il segnala di ingresso viene a cessare.

Indirizzamento

Anche per "localizzare" un timer, in fase di programmazione è necessario conoscerne l'indirizzamento.
Esempi di indirizzamento di un Timer del nostro PLC "Dummy" :

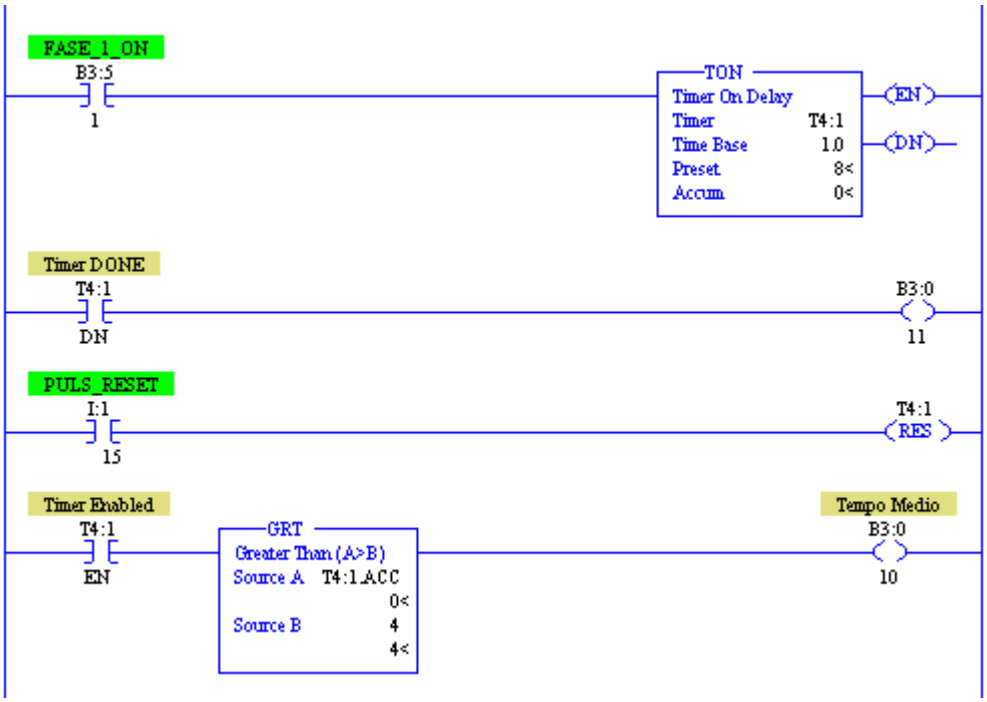
- **T4:1** rappresenta il Timer nr. 1 (del file nr. 4 di Timers)

Il timer a sua volta è composto dai seguenti elementi :

- **T4:1.PRE** Contiene il valore di preset (ad esempio 5 secondi)
- T4:1.ACC E' la locazione di memoria dove il PLC incrementa il tempo
- T4:1 / EN Bit di ENABLE : si attiva quando l'ingresso del Timer è a "1"
- **T4:1 / DN** Bit di DONE : si attiva quando il tempo è trascorso*

*Questo ultimo bit rappresenta l'uscita vera e propria del Timer, infatti la parola DONE si traduce con "Fatto!", intendendo che il timer ha finito di contare.

Graficamente un timer si presenta nel PLC in questo modo :



Quando il relè interno denominato “Fase_1_On” si attiva, il timer a sua volta attiva il bit Enable (EN) ed inizia il conteggio.
Dopo 5 secondi il conteggio termina, e viene posto a 1 anche il bit di Done (DN).
Se durante il conteggio qualcuno preme il pulsante reset (ingresso I:1/15) il timer si ripristina e riparte da zero (come se l’ingresso fosse venuto a mancare per un istante).

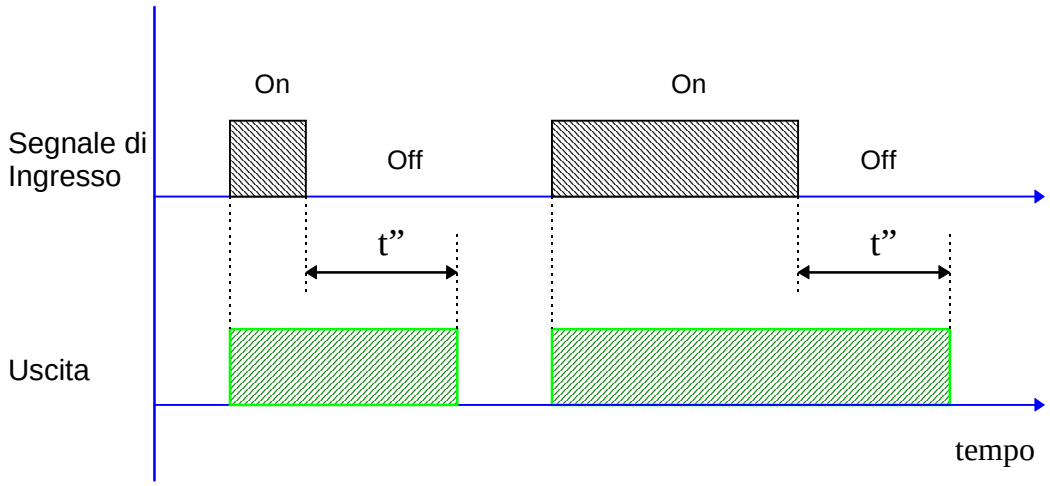
L’ultima istruzione è una piccola chicca che funziona solo quando il Timer è abilitato: confronta il valore del registro accumulatore con il numero 4, e solo quando il valore è maggiore attiva il relè interno denominato “Tempo medio”.

Nota Importante

Attenzione a non confondere il Timer con l’Istruzione Timer !
Un Timer è rappresentato dalla locazione nella *memoria dati* che contiene tutti i suoi elementi (tempo trascorso, set-point, ecc), ed è individuato da una sigla univoca, come ad esempio T4:1.
Le istruzioni invece TON e TOF possono essere utilizzate a piacimento, ma ovviamente con timer diversi : è quindi necessario fare la dovuta attenzione a non usare più volte lo stesso timer !

9.5.2 Timer OFF Delay

Questo temporizzatore eccita la sua uscita appena viene chiuso il contatto di ingresso, e rimane attivo fino a che il tempo non è trascorso.
Praticamente questo timer funziona come i “Temporizzatori Luci Scale” che si usano in tutti i palazzi.
Il diagramma di funzionamento è il seguente :



Come si vede nel grafico appena giunge il segnale di “start” proveniente dal contatto di ingresso l’uscita del Timer viene subito eccitata.
Il conteggio invece inizia solamete quando il segnale di ingresso torna Off.
Trascorso il tempo “t”, l’uscita del temporizzatore si diseccita.

I due esempi nel diagramma fanno capire come il timer venga attivato a prescindere dalla durata dell’impulso proveniente dall’ingresso.

ATTENZIONE !

Questo timer, una volta avviato, non può essere resettato (ossia fermato).
Sarà quindi necessario porre dei contatti di “sicurezza” in serie alle uscite comandate da questi tipi di temporizzatori.

9.6 CONTATORI

Ultimo componente discretamente utilizzato nei circuiti di controllo è il contaimpulsi.
Tale elmento è presente nei PLC con il nome di Counter , e viene detto anche contatore.

Questo dispositivo conteggia quante volte un certo ingresso ha compiuto una transazione da "0" a "1", indipendentemente dal tempo; in pratica conteggia quante volte si è chiuso un contatto.
Quando il valore di conteggio (accumulatore) raggiunge il numero predeterminato, il contatore eccita la propria uscita.

Nella memoria del PLC anche il Counter occupa due Word :

- Preset è la word che contiene il valore prestabilito (il Set-Point)
- Acc è la word che il PLC incrementa (o decrementa) ad ogni impulso.
- Bit di Stato danno lo stato di funzionamento del counter

Range di Lavoro

Tipico	da -32768 a 32767
in alcuni PLC più semplici	da 0 a 32767

Indirizzamento

Esempi di indirizzamento di un Counter del nostro PLC "Dummy" :

- **C5:1** rappresenta il Counter nr. 1 (del file di contatori nr. 5)

Il contatore a sua volta è composto dai seguenti elementi :

- **C5:1.PRE** Contiene il valore di preset (ad esempio 10 conteggi)
- C5:1.ACC E' la locazione di memoria che contiene il valore da conteggiare
- C5:1 / OV Bit di OVerflow - il conteggio ha superato il massimo (32767)
- C5:1 / UN Bit di UNderflow - il conteggio ha superato il minimo (-32768)
- **C5:1 / DN** Bit di DONE : si attiva quando il preset è stato raggiunto (uscita)

Tipologie

I counter sono disponibili in due modalità di funzionamento :

- Counter UP sigla CTU (conteggio a salire)
- Cunter Down sigla CTD (conteggio a decrescere)

Nota Importante

Attenzione a non confondere il Counter con le istruzioni che lo gestiscono !
Un Timer è rappresentato dalla locazione nella *memoria dati* che contiene tutti i suoi elementi, ed è individuato da una sigla univoca, come ad esempio C5:1.
Le istruzioni CTU e CTD invece, possono essere utilizzate a piacimento.

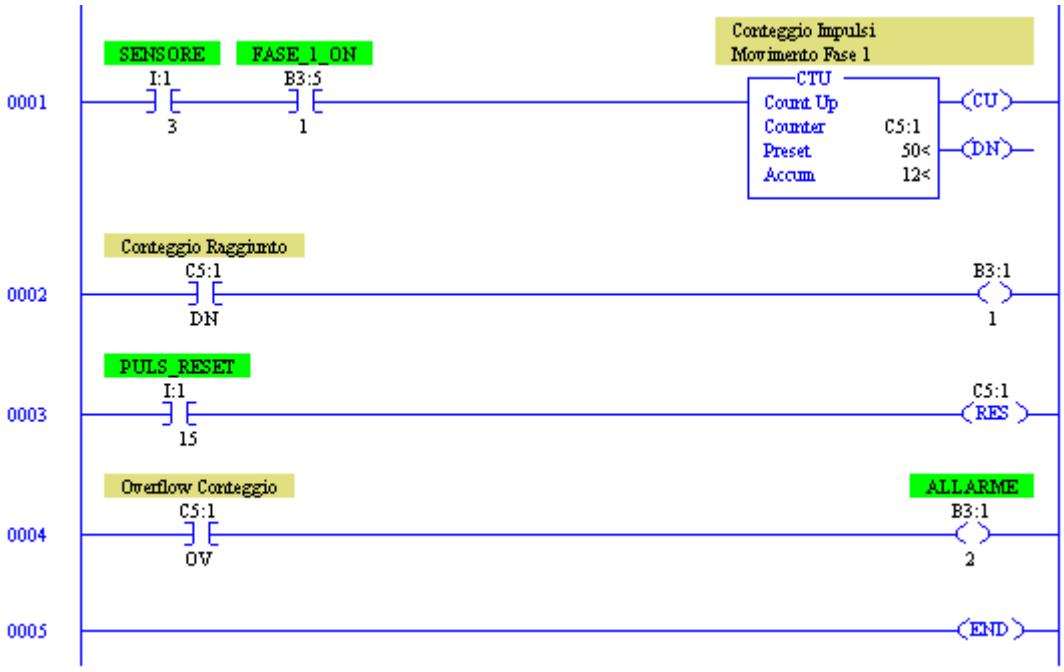
9.6.1 Counter Up - "CTU"

Questa istruzione elabora il contatore in modo che ogni volta che l'ingresso passa da 0 a 1 il conteggio venga incrementato.
Quando il valore di conteggio raggiunge il Preset, l'uscita viene attivata.

Per disattivare l'uscita ci sono due possibilità :

1. Si decrementa il conteggio in mondo da farlo diventare inferiore al Preset, oppure
2. Si esegue una istruzione RESET (indirizzata a quel contatore).

In quest'ultimo caso il valore di conteggio viene riportato a zero.



Nell'esempio riportato qui sopra quando è attiva la "fase 1" ad ogni impulso proveniente dal Sensore (I:1/3) il contatore C5:1 incrementa il conteggio (nel caso è arrivato a 12), e quando arriva al Preset (nell'esempio 50) si attiva l'uscita C5:1/DN.
Il pulsante Reset in questo caso lancia l'istruzione che resetta il contatore, mente se il contatore raggiunge il valore di overflow (32767) si attiva un bit di Allarme.

9.6.2 Counter Down - "CTD"

In questa istruzione ogni volta che l'ingresso passa da 0 a 1 il contatore viene decrementato.
Quando il valore di conteggio raggiunge il Preset, l'uscita viene attivata.

Per disattivare l'uscita ci sono due possibilità (simili al CTU) :

1. Si incrementa il conteggio in mondo da farlo diventare superiore al Preset, oppure
2. Si esegue una istruzione RESET.

9.7 RESET

Questa istruzione, come già visto negli esempi, serve per resettare timer e counter.
Quando la riga del programma esegue questa istruzione, essa resetta il singolo timer o singolo il counter che è stato indicato.
Il reset consiste essenzialmente nel portare a zero il valore dell'Accumulatore e nel resettare tutti i bit di stato del contatore o temporizzatore.

Nota : non è possibile resettare con un unica istruzione di reset tutti i timer e/o counter nel programma !

CAPITOLO 10.

La Programmazione di Base

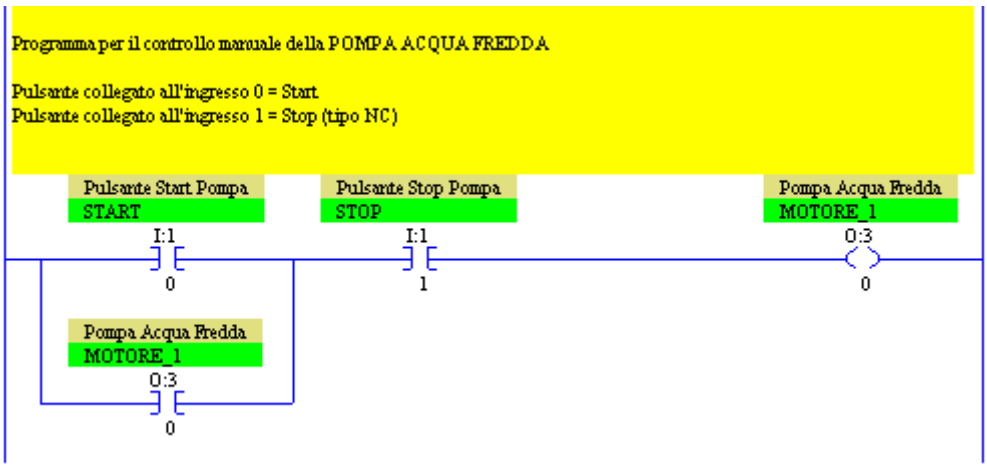
10.1 Generalità

Ecco il tanto agognato capitolo dedicato alla programmazione.
Ripensando ai tanti concetti già esposti, ora ci si renderà conto che l'argomento non era così semplice ed ha risvolti molto lontani dall'uso della tastiera e del mouse per inserire una sequenza di istruzioni.
Come già detto dietro ad un programma di un macchinario complesso o un impianto esteso c'è moltissima carta che ci dovrà essere fornita, ma anche che produrremo con i nostri appunti ed i nostri schemi a blocchi.
Per quanto riguarda la programmazione di base si rimanda ai manuali che forniscono le case costruttrici dei PLC dove vi sono sempre esempi pratici.

10.2 Alcuni esempi pratici

10.2.1 Marcia - Arresto Motore

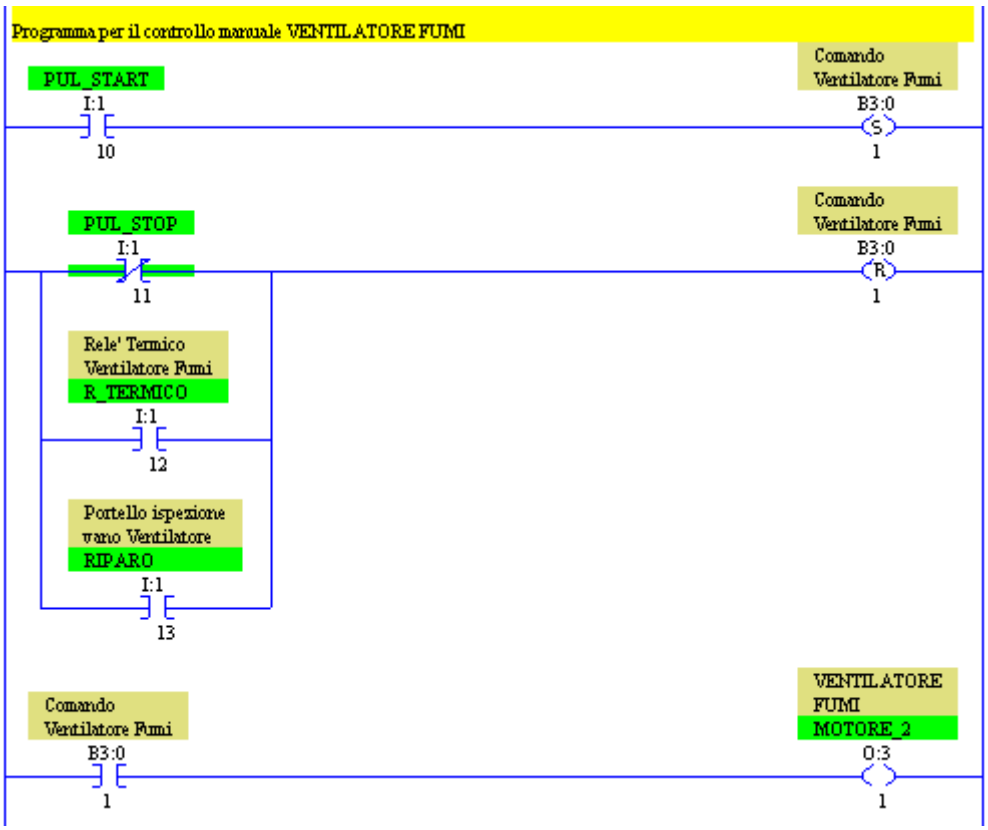
Un esempio classico di un circuito di comando industriale è quello dell'avviamento di un motore tramite due pulsanti Marcia-Arresto.



In pratica premendo il pulsante Start si causerà l'avviamento della pompa, che rimarrà in moto fino a che non verrà premuto il pulsante Stop, o non viene spento il PLC.
Il Pulsante Stop, ai fini della sicurezza, dovrà essere di tipo Normalmente Chiuso.

10.2.2 Marcia - Arresto con Memoria

Questo secondo esempio serve per eseguire la stessa funzione, ma è stato utilizzato un relè interno per fare in modo che alla mancanza di energia elettrica non venga "cancellato" il comando di avviamento.
Sono stati anche inseriti alcuni consensi di sicurezza (che comunque agiscono anche nella logica cablata del quadro elettrico).
In particolare è stato inserito il contatto relativo al relè termico di protezione del motore, e un contatto di sicurezza posto in un portello apribile.
Con questo programma il relè interno viene settato (eccitato) dal pulsante Start, e viene resettato dal pulsante Stop o dalle sicurezze.



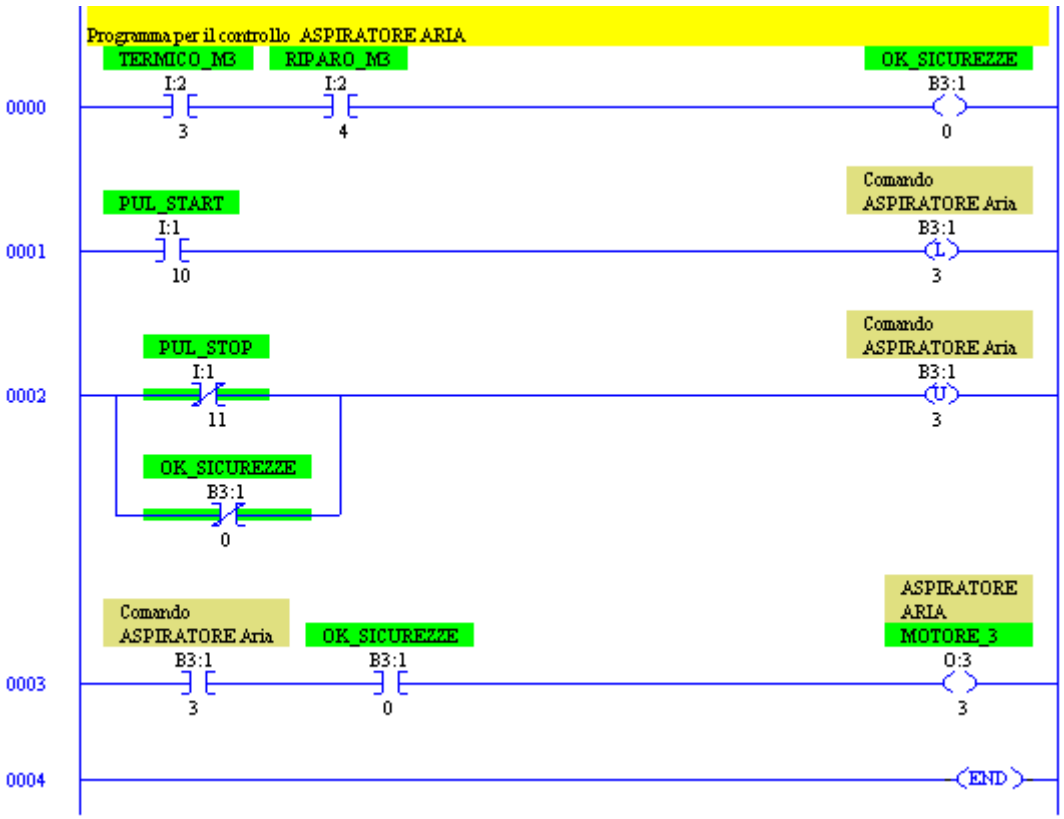
Il contatto del relè termico è quello NO in quanto quello NC è stato cablato in serie con la bobina del contattore che avvia il motore.
Lo stesso discorso vale per il contatto proveniente dal riparo.

Nota

Le istruzioni “Set” e “Reset”, da adesso si troveranno con le sigle “L” ed “U”, che significano, per analogia, “Latch” e “Unlatch”.
Ciò per non confondersi con il comando Reset, che serve per resettare timer e contatori.

10.2.3 Errore

Il programma appena presentato potrebbe funzionare in modo pericoloso se si inseriscono le righe di start e stop nell’ordine inverso.
In tal caso infatti succederebbe che il comando Start avrebbe la prevalenza sullo Stop.
Al fine di evitare questi problemi, ed al fine di rendere più evidente se una macchina è abilitata a funzionare, si è creato un apposito relè interno denominato “OK Sicurezze” e posto all’inizio del programma.
Tutte le uscite devono sempre avere in serie con la logica che le comando il contatto di consenso generale derivato dalle sicurezze.



CAPITOLO 11.

Cambiamenti di Stato dei Segnali

11.1 Generalità

Capita spesso negli impianti di automazione che sia necessario gestire un segnale nel momento in cui cambia di stato, piuttosto che quando è permanentemte in uno stato definito.

La gestione del cambiamento di stato di un segnale può servire spesso nella stesura di un programma di controllo di un qualsiasi macchinario.

"Impulsi di Comando"

Molto spesso con i pulsanti di comando è necessario che la logica di gestistione acquisisca solo il cambiamento di stato del pulsante.
Questo per fare in modo che l'azione mantenuta sul pulsante dalla mano dell'operatore non provochi ripetute azioni all'automatismo.

Si prenda ad esempio il tasto di Reset (o ripristino), che deve essere premuto ogni qualvolta avviene un allarme, ma non deve sortire alcun effetto se viene mantenuto costantemente premuto.

E' quindi necessario che nei PLC esista una procedura, in fase di programmazione, che sia in grado di gestire queste situazioni.

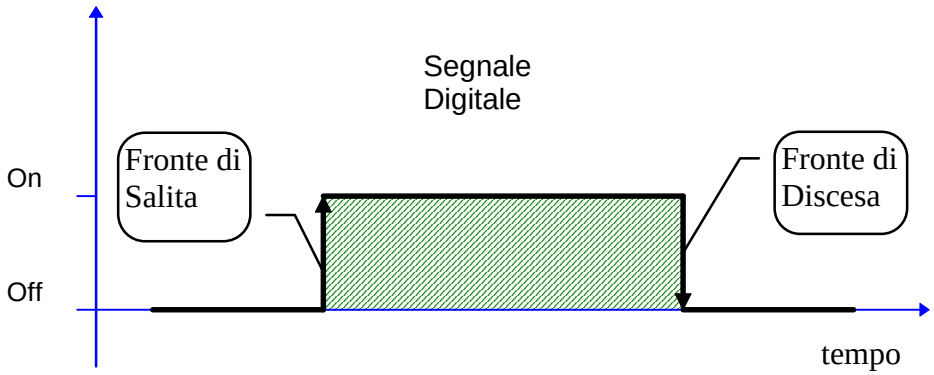
Prima di affrontare i temi della programmazione facciamo un breve escursus sul tema.

11.2 Fronte di un segnale

Il cambiamento di stato di un segnale, detto anche transazione, e può essere di due tipi :

Tipo di transazione			***Denominazione***	*Inglese*
da Falso a Vero	$0 \rightarrow 1$	Off $\rightarrow$ On	Fronte di Salita	Rising-Edge
da Vero a Falso	$1 \rightarrow 0$	On $\rightarrow$ Off	Fronte di Discesa	Falling-Edge

Per meglio spiegare questo concetto basta osservare il grafico sotto che riporta l'andamento nel tempo di un segnale digitale qualunque.



Come si vede, il fornte di salità è momento in cui avviene il cambiamento di stato di un segnale digitale da zero ad uno, mentre il fronte di discesa rappresenta la situazione opposta.

Si ricorda ora quanto detto nel paragrafo 4.3 (Cap. 4 - Scansione), ossia che un programma si "accorge" che un segnale è cambiato solo quando ricomincia una nuova scansione.

E' quindi possibile, tramite delle apposite istruzioni, programmare il PLC in modo che attivi una procedura particolare quando un certo segnale cambia di stato.
Tale procedura, naturalmente, verrà eseguita solo per il periodo di una singola scansione, che è il tempo minimo gestibile da un PLC.

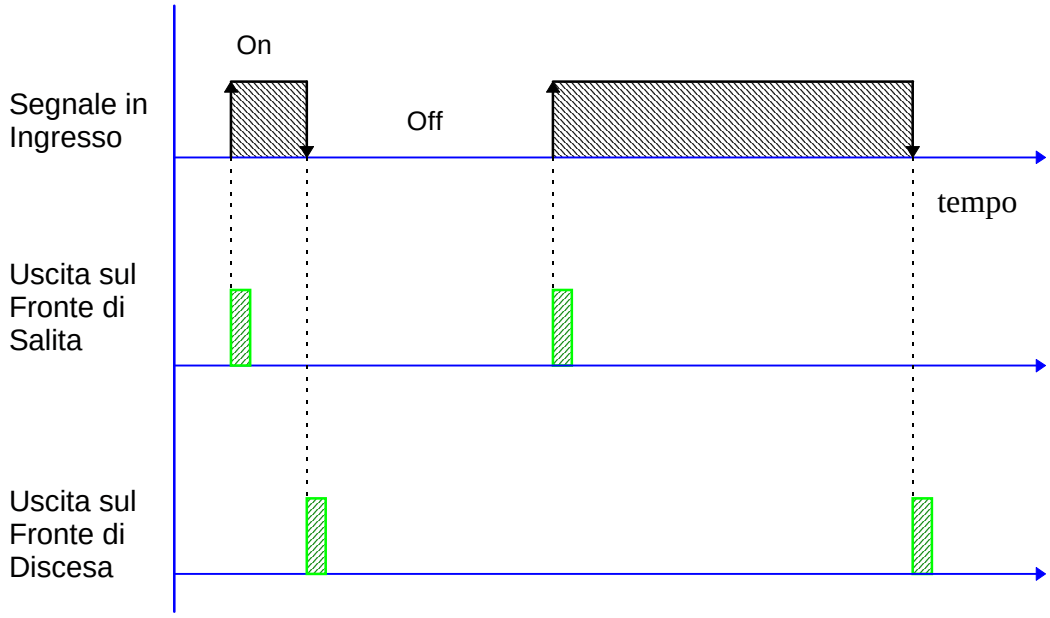
11.3 Istruzioni nel PLC

I costruttori di PLC per venire incontro a questa esigenza, hanno creato delle apposite istruzioni che acquisiscono un qualsiasi ingresso, ed in uscita danno un impulso che segnala il cambiamento di stato dell'ingresso.

In realtà questa funzionalità è possibile crearsela con le istruzioni base, ma diciamo comuque che che il nostro PLC Dummy abbiamo le seguenti istruzioni :

Nome Istruzione	***Sigla***	***Note***	*altri nomi*
One Shot Rising	**OSR**	Fronte di Salita	Differential Up
One Shot Falling	**OSF**	Fronte di Discesa	Differential Down

Prendendo come esempio un segnale digitale che cambia in modo casuale nel tempo e ponendolo in ingresso alle due istruzioni, il risultato è il seguente :

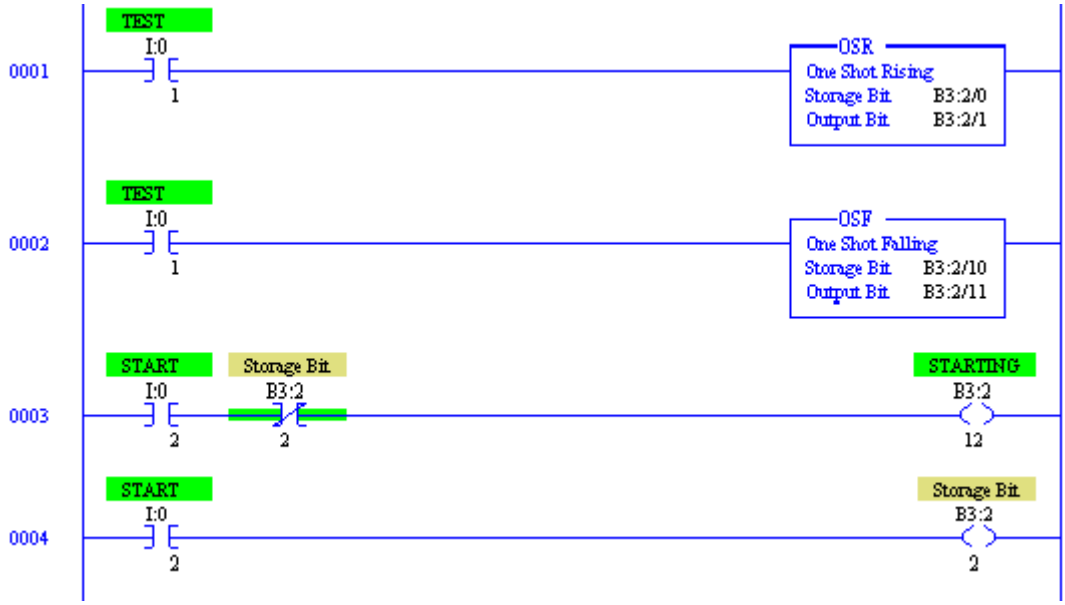


Come si vede la gestione del "Fronte di Salita" riconosce i cambiamenti da 0 a 1 e li segnala al programma, mentre non emette alcuna risposta al fronte di disensa.
La gestione del "Fronte di discesa" funziona invece in modo contrario.

NOTA : A prescindere dalla durata del segnale in ingresso, l'impulso ha una durata pari ad una sola scansione del programma.

11.4 Esempio

Nell'esempio che segue si sono usate le istruzioni speciali, e successivamente si è usata la programmazione "manuale" per dare una maggiore comprensione.
Da notare che per gestire di un cambiamento di stato c'è sempre bisogno di un bit di appoggio, che è stato qui denominato "Storage bit".



Nelle prime due righe dell'esempio, l'ingresso TEST provoca due distinti eventi :

- quando passa da Off a On l'istruzione OSR attiva per una scansione il bit B3:2/1,
- quando passa da On a Off l'istruzione OSF attiva per una scansione il bit B3:2/11.

Le righe 3/4 invece servono per emulare l'istruzione OSR, ma con la logica tradizionale dei contatti del PLC, un trucchetto si usava quando non esistevano le istruzioni avanzate, e che sfrutta proprio il fatto che il programma viene eseguito riga per riga.

Vediamo in pratica come funziona :

- quando l'ingresso START è a zero, lo sono anche il bit "Starting", e lo "Storage bit".
- quando l'ingresso START passa da zero a uno viene attivato il bit "Starting", e successivamente lo "Storage bit".
- Nella scansione successiva, a causa della presenza dello "Storage bit",nella riga 3 il bit di "starting" ritorna a zero, ***che quindi è stato attivo per una sola scansione***.

Nella realtà il PLC con le istruzioni avanzate OSF/OSR compie la stessa funzionalità del "vecchio trucchetto", ma occupa solo una riga di programma e la lettura dello stesso risulta molto più chiara.

CAPITOLO 12.

Impostiamo un Metodo di Lavoro

Generalità

Capita spesso che tanti tecnici, non di meno il sottoscritto, tentino di risolvere tutti i propri problemi stando davanti al PC.
Questa tecnica comporta risultati pessimi ed enormi perdite di tempo, discorso che riguarda qualunque programmazione o progettazione.

Prima di cominciare a programmare un PLC quindi lasciate perdere il personal computer, ed invece procuratevi matita, gomma e un bel pò di carta.
Va bene anche quella che è già stata stampata da una parte.

Come cominciare

Il metodo di lavoro che qui si propone è suddiviso in vari "fogli", ognuno di essi destinato ad una funzione ben specifica.
L'elenco che segue è redatto in ordine cronologico, come si svolgerà il lavoro, ma in realtà dovrete ritornare sui vostri passi più volte mentre lo svolgerete.

FOGLIO 1 - Planimetria Generale

Il programma del vostro PLC sicuramente andrà ad agire un impianto o su di una macchina.
Su un primo foglio quindi esegui uno schizzo dell'impianto o del macchinario tipo una planimetria ed inserisci tutti gli elementi che lo compongono :
- struttura generale
- motori, pistoni , parti in movimento
- oggetti in lavorazione (scatoloni, assi di legno, lavatrici, ecc..)

L'ideale è servirsi di un software CAD per fare una prima bozza di lavoro, cioè un layout generale, sulla quale poi a matita inserire tutti gli elementi collegati veramente all'automazione.

NOTA
Non è indispensabile poi riportarli nel disegno a PC, a meno che non sia richiesto da chi poi dovrà utilizzare o manutenere l'impianto e/o il macchinario.
Non si può negare che consegnare con la documentazione finale del vostro lavoro questo disegno completo di tutti gli elementi sia un fatto di pregio non indifferente.

FOGLI 2 & 3 - Elenco Ingressi e Uscite

Prepara un paio di fogli nei quali indicherai tutti gli elementi che saranno collegati al PLC, utilizzando una tabella di questo tipo :

Indirizzo	*Sigla*	*Descrizione*
I 0.1	FTC1	Fotocellula fine nastro trasportatore
I 0.2	RT2	Relè termico motore elevatore...

Numera o meglio assegna un nome a tutti i dispositivi che saranno collegati al PLC e mettili anche nella planimetria.
Il nome viene chiamato “Mnemonico” in molti software di programmazione, che è la colonne “sigla” della tabella mostrata.
In un foglio indicherai gli ingressi, ossia i segnali provenienti da sensori, pulsanti, e contatti in genere, mentre nell’altro indica le uscite, ossia i comandi che impartisce il PLC.

FOGLIO 4 - SEQUENZA GENERALE

Schizza una sorta di Flowchart della sequenza che esegue la macchina.
Attento che questo non e' il programma, ma lo schema di funzionamento "meccanico" della macchina.
Una sorta di elenco di tutto quello che succede.
Può essere anche molto sintetico, ma e' utile per schiarirsi le idee.

FOGLIO 5 - ELENCO SUBROUTINE

Guardando il flowchart generale cerca vedere se è possibile smembrare questo “processo” in più fasi distinte ed autonome.
In questo modo si potrà ottimizzare la programmazione creando dei sottoprogrammi ognuno dei quali svolgerà una funzione ben precisa,.
Creare un software “modulare” comporterà enormi vantaggi nella fase di correzione e messa a punto del sistema, soprattutto se ognuno dei moduli si gestirà sostanzialmente in modo autonomo rispetto agli altri.

Di questo aspetto si è già parlato anche nel Capitolo 8.

FOGLI 6 e seguenti - FUNZIONI dei PROGRAMMI

Abbozza ora un elenco delle cose che ogni subroutine deve fare e controlla se hai tutti i segnali di ingresso per eseguire correttamente quel compito.
Ogni subroutine un foglio, con il titolo bene in alto.
Nota che in questa fase, se ben fatta, ci si accorge se manca il segnale di "start" per eseguire quella certa movimentazione, o al contrario quello di "fine", o se serve un "timer" per attendere un evento.
In pratica questo è il momento in cui si elaborano davvero le fasi che ogni programma dovrà eseguire, e che farà da base per la stesura del software nel PLC.

CONCLUSIONE

Ora con tutta la documentazione che hai preparato puoi accendere il PC ed iniziare a scrivere il programma per il tuo PLC.
Auguri e buon lavoro.

Altri esempi di programmazione per futuri capitoli possono essere

- Tecniche di gestione delle sicurezze
- Gestione dei consensi all'avviamento e delle modalità Automatico/Manuale
- Tecniche di gestione degli Allarmi,
- Tecniche di gestione di una sequenza automatica con Set-Reset progressivi,
- Tecniche di gestione degli ingressi Analogici
- Tecniche per il reset del posizionamento di elementi meccanici mobili.

Bibliografia

	Libri - Automazione	
Enrico Grassani	L'equipaggiamento elettrico delle macchine	Editoriale Delfino
Enrico Grassani	La Direttiva Macchine (DPR 459/96)	Editoriale Delfino
	La Direttiva Macchine - Problemi e Soluzioni	Edizioni TNE - Torino
Carlo Torresan	Controllo dei processi chimici e termici	1999 - Editoriale Delfino
	Elettrotecnica & Normative	
Vito Carrescia	Fondamenti di Sicurezza Elettrica	Edizioni TNE - Torino
Francesco Dal Mas	Manuale di Disegno Elettrotecnico	Edizione CEI - Milano
	Volume "Norma CEI 64-8"	Edizione CEI - Milano
	Norma CEI 44-5 / EN 60204-1	CEI - Milano
	Riviste mensili	
	ELETTRIFICAZIONE	Editoriale Delfino
	AUTOMAZIONE Oggi	Editore VNU
	TUTTONORMEL	Edizioni TNE - Torino
	WATT	Tecniche Nuove
	Testi in lingua Inglese	
W. Bolton	Programmable Logic Controllers	Newnes
Clements-Jeffcoat	The PLC Workbook	Prentice Hall
Alan J Crispin	Programmable Logic Controllers 2.ed.	McGraw - Hill

COPYRIGHT - Proprietà del Documento

Questo documento è stato redatto da Marco Dal Prà, perito industriale iscritto all'albo della Provincia di Venezia, e responsabile tecnico della ditta Novatekno Srl di Mestre (VE). Tutti i diritti sono riservati.

Cosa si può fare

Il documento può essere liberamente utilizzato e distribuito per scopi didattici sia da parte di studenti che di docenti di scuole pubbliche di ogni grado, e di corsi di specializzazione pubblici.
Può essere liberamente stampato per uso personale da chiunque sia interessato ad approfondire l'argomento in proprio.

Cosa non si può fare

Il documento non può essere replicato, nemmeno parzialmente, su altri siti internet, mailing list, pubblicazioni cartacee (riviste) e cd-rom, ciò indipendentemente dalle finalità di lucro.
E' proibito utilizzarlo a scopo di lucro, come ad esempio da parte di società private che a qualsiasi titolo tengano corsi di aggiornamento e/o di specializzazione.
Per tali finalità è possibile prendere accordi con il responsabile di Novatekno Srl, Marco Dal Prà, accordi che dovranno essere formulati in forma scritta da entrambe le parti.

Esclusione di Responsabilità

I contenuti del presente documento sono utilizzabili così come sono.
Nonostanti i controlli fatti prima di renderlo di pubblico dominio nel sito internet della ditta Novatekno Srl, **non** è possibile assicurare che il documento sia esente da errori e/o omissioni.
Nessuna responsabilità può essere attribuita all'autore del documento per l'utilizzo dello stesso.
E' quindi responsabilità del lettore controllare molto attentamente il buon funzionamento dei programmi da lui realizzati seguendo gli esempi contenuti nel presente testo.

Note

Le schermate di esempio sono state ricavate dal software RSLogix500 della Rockwell Automation, in ambiente Microsoft Windows..
I marchi citati nel presente documento sono di proprietà dei relativi produttori.

Aggiornamenti

I presente documento può essere aggiornato dall'autore a sua discrezione e senza alcun preavviso. Ad esempio l'autore può decidere di effettuare un aggiornamento sulla base di libere segnalazioni fatte dai lettori.
In ogni caso, ciò non avviene a cadenza periodica.
Gli eventuali aggiornamenti saranno messi a disposizione di tutti nel sito internet della Novatekno Srl, alla pagina *download*.
L'edizione del documento è costituita dalla data, visibile nella prima pagina.