

Università degli Studi di Ferrara  
Facoltà di Scienze Matematiche, Fisiche e Naturali  
Corso di Laurea in Informatica



Sistemi di controllo e raccolta dati  
nell'industria dei laterizi

*Relatore:*

Dott. MIRCO ANDREOTTI

*Laureando:*

SIMONI ALFRED

*Anno accademico 2005/2006*



*Ai miei genitori*



# Indice

<b>Introduzione</b>	<b>ix</b>
<b>1 Introduzione al processo di produzione dei laterizi</b>	<b>1</b>
1.1 I laterizi . . . . .	1
1.2 Fase 1: La materia prima . . . . .	3
1.3 Fase 2: Prelavorazione della materia prima . . . . .	4
1.3.1 Finalità della prelavorazione . . . . .	4
1.3.2 Il trasporto del materiale . . . . .	6
1.4 Fase 3 : Formatura del laterizio . . . . .	6
1.4.1 Formatura per estrusione . . . . .	6
1.4.2 Formatura per pressatura . . . . .	8
1.5 Fase 4: Essiccazione . . . . .	9
1.5.1 Riscaldamento diretto . . . . .	9
1.5.2 Riscaldamento indiretto . . . . .	10
1.6 Tipologie di essiccatoi . . . . .	11
1.6.1 Essiccatoi continui . . . . .	11
1.6.2 Essiccatoi statici . . . . .	11
1.6.3 Essiccatoi semicontinui . . . . .	12
1.7 Fase 5: Cottura . . . . .	12
1.8 Fase 6: Confezionamento . . . . .	13
1.9 Sistema di movimentazione dei castelli o carrelli . . . . .	14
<b>2 Introduzione al PLC</b>	<b>19</b>
2.1 Funzionamento del PLC . . . . .	21
2.2 I/O del PLC . . . . .	23
2.2.1 Tipi di dati . . . . .	24
2.2.2 I/O Digitali . . . . .	25
2.2.3 I/O Analogiche . . . . .	26
2.3 Memoria del PLC . . . . .	28
2.3.1 La memoria di lavoro RAM . . . . .	29

2.3.2	La memoria ROM . . . . .	29
2.3.3	La memoria di massa . . . . .	29
2.4	Protocolli di comunicazione tra PLC . . . . .	30
2.5	Il protocollo PROFIBUS . . . . .	33
2.5.1	Profibus Fieldbus Message Specification . . . . .	35
2.5.2	Profibus Device Peripheral . . . . .	35
2.5.3	Profibus Process Automation . . . . .	37
2.5.4	Struttura dei frame ProFibus . . . . .	37
2.6	Programmazione dei PLC . . . . .	40
2.6.1	Il software . . . . .	40
2.6.2	Linguaggi di programmazione . . . . .	41
<b>3</b>	<b>Sistemi di supervisione</b>	<b>49</b>
3.1	Tipologie di HMI . . . . .	50
3.2	Progettazione dell'interfaccia utente i vincoli . . . . .	52
3.3	Requisiti progettuali dell'interfaccia utente . . . . .	53
3.4	Model,View,Controller . . . . .	54
3.5	Comunicazione tra PLC e HMI . . . . .	55
3.6	Caratteristiche del software SCADA . . . . .	56
<b>4</b>	<b>Sistema di supervisione del forno</b>	<b>61</b>
4.1	Fasi del processo di sviluppo . . . . .	61
4.2	La programmazione del PLC . . . . .	63
4.2.1	Blocchi organizzativi OB . . . . .	65
4.2.2	Blocchi Funzione FC . . . . .	68
4.2.3	Blocchi Funzione FB . . . . .	70
4.2.4	Blocchi Dati DB . . . . .	72
4.2.5	Blocchi Funzione di Sistema SFC e SFB . . . . .	74
4.3	Test del software del PLC . . . . .	74
4.4	Modellazione del software SCADA . . . . .	76
4.4.1	La strutture delle pagine . . . . .	77
4.4.2	Funzionalità del progetto . . . . .	79
.1	Appendice . . . . .	95
.2	Appendice A . . . . .	96
.2.1	La codifica 4B5B . . . . .	96
.2.2	La codifica NRZ . . . . .	97
.2.3	La codifica NRZI . . . . .	98
.2.4	La codifica Manchester . . . . .	99
.2.5	La codifica MLT-3 . . . . .	99

---

.3	Appendice B . . . . .	101
.3.1	Modalità Sincrona . . . . .	101
.3.2	Modalità Asincrona . . . . .	102
.4	Appendice C . . . . .	105
.4.1	Lo standard RS232 . . . . .	105
.4.2	Lo standard RS422 . . . . .	107
.4.3	Lo standard RS485 . . . . .	108
.5	Appendice C . . . . .	111
.6	Appendice E . . . . .	113

<b>Bibliografia</b>	<b>114</b>
---------------------	------------



# Introduzione

La necessità di avere macchine sempre più veloci e precise è che non richiedano la presenza dal uomo per la regolazione, ha portato le persone a cercare e costruire nuovi sistemi di automazione. All'inizio l'automazione veniva fatta con macchine a vapore, poi con l'applicazione dell'elettricità in questo campo, vennero inventate macchine elettromeccaniche. Con la nascita dell'elettronica prima quella analogica e successivamente quella digitale ci fu una vera rivoluzione in questo campo. Vennero creati dispositivi elettronici programmati. Al inizio si trattava di apparecchiature molto limitate e con molti problemi di funzionamento. Sul finire degli anni 70 le aziende produttrici di sistemi elettronici industriali iniziarono a intravedere, nel mondo dei microprocessori, un nuovo modo di realizzare i sistemi di controllo dei processi e delle macchine. Il motivo era molto semplice : si poteva sviluppare un unico prodotto per tutte le applicazioni, ma personalizzabile per ogni cliente mediante la semplice modifica del software al suo interno. Nella maggior parte dei casi l'apparecchiatura sarebbe stata fornita senza alcun software, è sarebbe stato il cliente a svilupparlo su misura per la propria applicazione. In quel periodo i mondi tecnologici che attendevano questa nuova rivoluzione erano essenzialmente due : 1. Il mondo dei segnali analogici, per il controllo e la regolazione dei processi chimici e termici. 2. Il mondo dei segnali elettrici on/off, per il controllo di macchinari e dispositivi elettrici convenzionali.

Dato che le due tecnologie erano molto diverse tra loro, così come lo erano i clienti che le richiedevano, la strada si è naturalmente divisa : Da un lato sono nati i dispositivi programmabili per il trattamento di segnali analogici per effettuare regolazione di processi, successivamente sfociati nell'acronimo DCS (Distributed Control System).

Dall'altro lato sono nati i dispositivi per l'elaborazione dei segnali digitali, con lo scopo di sostituire i vecchi quadri composti da relè, temporizzatori, contaimpulsori ecc, conosciuti poi con l'acronimo PLC (Programmable Logic Controller). An-

che i produttori di questi sistemi si sono divisi in due gruppi, ognuno con il suo bagaglio di esperienze e di specialisti, ma il secondo gruppo, quello dei PLC, ha ottenuto una fetta di mercato notevolmente più ampia. I sistemi di regolazione hanno infatti un comportamento che è analogo a quello delle azioni umane ed è per questo che hanno avuto un così grande successo. Da un certo punto di vista questo ragionamento è anche ovvio, in quanto lo scopo dei sistemi per l'automazione ed il controllo delle macchine è quello di sostituirsi all'uomo. Il PLC con il progredire della tecnologia elettronica e informatica sta diventando un sistema multifunzionale adatto a qualsiasi impiego industriale, è negli ultimi anni anche in quello civile. È da quest'ultimo tipo di impiego che si è sviluppato un ramo nuovo dell'automazione chiamato DOMOTICA. La domotica ha lo scopo di riunire sotto il controllo di un unico dispositivo (PLC) tutti i dispositivi tecnologici che ci possono essere in una abitazione. L'interfaciamento con l'utente è reso possibile grazie a dispositivi HMI con interfaccia grafica molto intuitiva. Con la questa tesi si è cercato di dare una breve descrizione di questo tipo di tecnologia. La tesi tratta un sistema di controllo e di raccolta dati applicato in un impianto utilizzato per la produzione di laterizi. Nel primo capitolo si dà una breve descrizione di quale sia il processo produttivo dei laterizi, nel senso di quale sia il processo da automatizzare. Si fornisce inoltre una breve descrizione dei vari componenti di questo impianto, del loro funzionamento. Nel secondo capitolo si fornisce una sintesi dei sistemi di controllo PLC, di come questi sistemi siano costruiti e del loro funzionamento. Si fornisce una descrizione dei loro componenti principali quali: memoria, dispositivi di input-output, CPU ect. Sempre in questo capitolo si descrive la comunicazione via rete di questi dispositivi, dei loro protocolli e delle architetture di rete che si possono formare. Il terzo capitolo tratta le interfacce grafiche HMI distinguendo tra questi gli HMI standard e i sistemi SCADA inoltre descrive quali siano le tecnologie utilizzate per queste interfacce, le regole di progettazione ed infine la comunicazione tra HMI-PLC. In fine il capitolo quattro riporta un progetto realizzato con un PLC e un sistema SCADA per la conduzione di una parte del impianto di produzione dei laterizi, più precisamente per la conduzione di un forno. Essendo un argomento lungo e complesso si è cercato di dare una descrizione quanto più concisa e chiara.

# Capitolo 1

## Introduzione al processo di produzione dei laterizi

### 1.1 I laterizi

I laterizio , il mattone in particolare ha origini antiche, le prime testimonianze risalgono al 4000 a.C. Nel corso dei secoli questo prodotto si è perfezionato sempre di più fino ad arrivare ai giorni nostri dove è diventato il componente di base per la costruzione di abitazioni . La tecnologia per la produzione dei laterizi nel corso degli anni è cambiata ma il processo produttivo [1] fondamentale e rimasto invariato. Nella produzione industriale di grande serie, quella del laterizio presenta una tipologia di elementi modulari estremamente estesa. Una prima suddivisione di tali elementi si può scegliere in base alla loro funzione specifica. È noto generalmente che il laterizio nelle costruzioni si impiega nelle seguenti forme:

- Mattoni e blocchi per murature portanti e non;
- Tegole e coppi per coperture;
- Blocchi leggeri per solai, o per tutte quelle opere edili a superfici orizzontali o in alcuni casi inclinate.

In questa prima inquadratura si può precisare che, nei termini su indicati (mattoni e blocchi, tegole e coppi, blocchi leggeri), sono compresi moltissimi tipi, aventi ognuno particolari caratteristiche di forma, atte a soddisfare non soltanto requisiti estetici, ma anche di resistenza meccanica a vari tipi di sforzi impressi, di protezione contro gli agenti atmosferici, di protezione termica

ecc. Tutti queste tipologie di laterizi hanno in comune il processo produttivo. Nell'immagine seguente Fig:1.1 viene mostrata tramite uno schema a blocchi il processo produttivo dei laterizi.



Figura 1.1: Schema generale del processo produttivo dei laterizi

Di seguito viene data una breve descrizione di ognuna di queste fasi.

## 1.2 Fase 1: La materia prima

Le argille sono responsabili delle caratteristiche tipiche dei materiali per laterizi, quali la coesione, la plasticità, la lavorabilità e la resistenza meccanica del secco e del cotto. I tipi di argilla sono numerosi ed hanno molti punti in comune e comportamenti analoghi in presenza di acqua. Tuttavia ognuno di essi allo stato puro si differenzia dagli altri principalmente per l'associazione di elementi chimici di varia natura e per la ripartizione granulometrica ecc. Dopo la scoperta dei giacimenti di argilla si procede allo scavo. Esistono diversi tipi di macchinari che svolgono questo compito, vengono usate in base alla conformazione del terreno. Dopo lo scavo l'argilla viene raccolta in grossi cumuli all'interno dello stabilimento in attesa per la lavorazione. L'argilla accumulata viene prelevata mediante un escavatore a tazze e immessa nei nastri trasportatori che alimentano la linea di produzione. Come si vede dall'immagine fig:1.2 l'escavatore a tazze preleva l'argilla dal cumolo e la deposita su un nastro trasportatore.

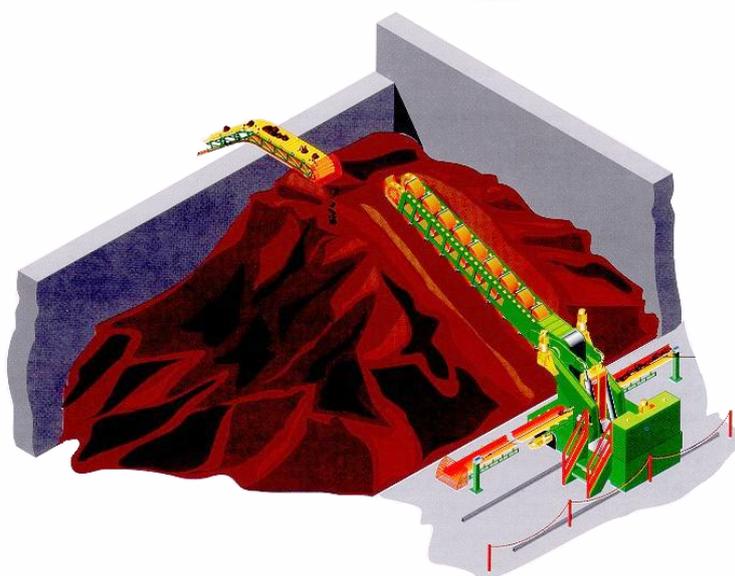


Figura 1.2: Escavatore a tazze

Dopo la fase di stoccaggio, l'argilla viene immessa su una linea di trans-

port (nastri trasportatori) che trasportano questo materiale nella sala macchine di prelaborazione. Questo ambiente è composto da diverse tipologie di macchine che svolgono singoli compiti. Nella parte seguente verranno descritte alcune di queste macchine, macchine che non devono mancare mai in una linea di prelaborazione.

## **1.3 Fase 2: Prelavorazione della materia prima**

### **1.3.1 Finalità della prelaborazione**

La prelaborazione, chiamata anche preparazione, consiste in una serie di operazioni, molto diverse tra loro, atte a condurre la materia prima, o più materie prime, dallo stato in cui si trovano dopo lo scavo ad una condizione finale, nella quale si siano contemporaneamente raggiunte le seguenti caratteristiche:

- Uniformità e costanza di composizione degli elementi mineralogici e chimici, contenuti nella materia prima a disposizione.
- Suddivisione dei vari componenti in parti minute, affinché nella miscela l'omogeneità sia estesa anche alle frazioni granulometriche più piccole.
- Umidificazione uniforme ed in quantità sufficiente alla più conveniente formazione del prodotto.

Il ciclo produttivo di queste macchine generalmente è fatto in modo sequenziale e continuo. Sequenziale perché, il materiale lavorato da una macchina viene trasportato alla macchina successiva, cioè si ha una sequenza di lavorazioni ben definite. Continuo perché non ci sono tempi di attesa tra il passaggio da una macchina all'altra. Di seguito viene data una breve descrizione di tre fasi importanti della prelaborazione.

#### **Alimentazione e dosaggio dei materiali**

In questa fase ha luogo la miscelazione di materiali argillosi diversi. Oltre alla miscelazione delle argille, se si ritiene opportuno si può integrare il materiale argilloso di base con aggiunte di altri elementi chimici. Questa aggiunta di altri elementi chimici serve a modificare le caratteristiche del prodotto in base alle esigenze di produzione. Oltre alla miscelazione delle argille con sostanze chimiche questa macchina ha il compito di alimentare le altre macchine a valle della catena produttiva, di conseguenza è posta all'inizio dello stabilimento.

### La macinazione

Dopo l'integrazione con sostanze chimiche si passa alla frantumazione dell'argilla. Le macchine per la frantumazione dei materiali argillosi hanno forme ed funzioni molto diverse. Ad alcune di queste macchine viene affidata la funzione secondaria, ma importante, della eliminazione delle impurità eventualmente presenti nel materiale scavato (pietre, ghiaia, legno, radici). A partire dalle zolle più grosse, scavabili mediante scavatori a cucchiaio, fino a quelle più piccole ottenibili dall'escavatore a tazze. Si possono identificare varie azioni di riduzione o macinazione. Tale azione avviene per gradi e si distingue la riduzione in base alla dimensione del prodotto in uscita:

- Grossolana, nell'intervallo tra 30 - 60 mm;
- Media, fino a 4 - 6 mm;
- Fina, dell'ordine di 1 mm.

Il laminatoio è una macchina adibita a questo scopo, inoltre non manca mai nelle linee di prelavorazione. Tale macchina è costituita sostanzialmente da due cilindri lisci a rotazione contraria con velocità di rotazione differenziate. Uno dei due alberi è fisso, l'altro può essere spostato. Ciò serve per la regolazione della dimensione di macinazione o luce di passaggio fra i due cilindri (chiamata anche finezza di laminazione), inoltre serve anche per la sicurezza nel caso di introduzione di impurità molto dure. In quest'ultima eventualità lo spostamento avviene per rottura di ponticelli a reazione calibrata, o a mezzo di braccia spostabili a reazione di tipo pneumatico. Le azioni che il laminatoio esercita sull'argilla sono: schiacciamento, urto e stiramento. In funzione della luce di passaggio tra i cilindri la laminazione si distingue in sgrossatura (3 - 4 mm) e finitura (1 - 3 mm); in presenza di calcinelli<sup>1</sup> la luce si riduce fino a 0,8 mm. La produzione di questa macchina dipende dalla dimensione della luce di passaggio, dal diametro dei cilindri, dal numero di giri medio e dalla regolare distribuzione del materiale sui cilindri. Nell'laminatoio le sue masse rotanti sono dotate di grande inerzia, l'avviamento o la fermata presenta quindi qualche difficoltà.

---

<sup>1</sup>Con il termine calcinelli viene indicata una formazione di granuli di ossido di calcio, originati da calcare presenti nel terreno.

### **Miscelazione e bagnatura**

Mettere insieme elementi di differenti caratteristiche per formare una unica miscela, che sia omogenea nel tempo, è l'obiettivo della miscelazione. Le condizioni di omogeneità diventano migliori in corrispondenza della più minuta suddivisione dell'argilla, ossia della più intensa macinazione. Durante la miscelazione, quando l'umidità complessiva dei componenti non è sufficiente per ottenere la voluta plasticità, si aggiunge per differenza l'acqua necessaria. Questo processo richiede tempo in quanto l'acqua aggiunta deve essere assorbita dall'argilla.

### **1.3.2 Il trasporto del materiale**

La ripresa del materiale da una macchina e l'alimentazione della successiva sono affidate a nastri trasportatori, con i quali si superano le distanze in orizzontale e le distanze verticali. L'impiego dei nastri consente la massima libertà nella progettazione delle linee di macchine, come scelta delle posizioni in relazione alla disponibilità di spazio e all'agibilità nei montaggi, nella manutenzione e nella sostituzione. Con questo si conclude la parte riguardante la prelavatura, nella parte successiva verranno descritte le tecniche più utilizzate per la formatura del laterizio.

## **1.4 Fase 3 : Formatura del laterizio**

Per formatura del laterizio si intende quel processo per cui la materia prima assume una forma ben determinata. Di sistemi di formatura del laterizio ve ne sono diversi, ma in pratica si dividono in due metodi, principali, che sono:

- Formatura per estrusione.
- Formatura per pressatura.

Di seguito si dà una breve descrizione dei due metodi.

### **1.4.1 Formatura per estrusione**

La formatura per estrusione è un sistema usato per materiali umidi di plasticità sufficiente a permettere il loro passaggio attraverso un diaframma forato (filiera). La filiera è rimovibile, nel senso che quando varia il tipo di prodotto si sostituisce la filiera con quella del prodotto che si vuole realizzare. La forza

necessaria è impartita da una elica rotante al interno di un involucro anulare che spinge l'impasto attraverso la filiera. La fig:1.3 mostra questo tipo di macchina.

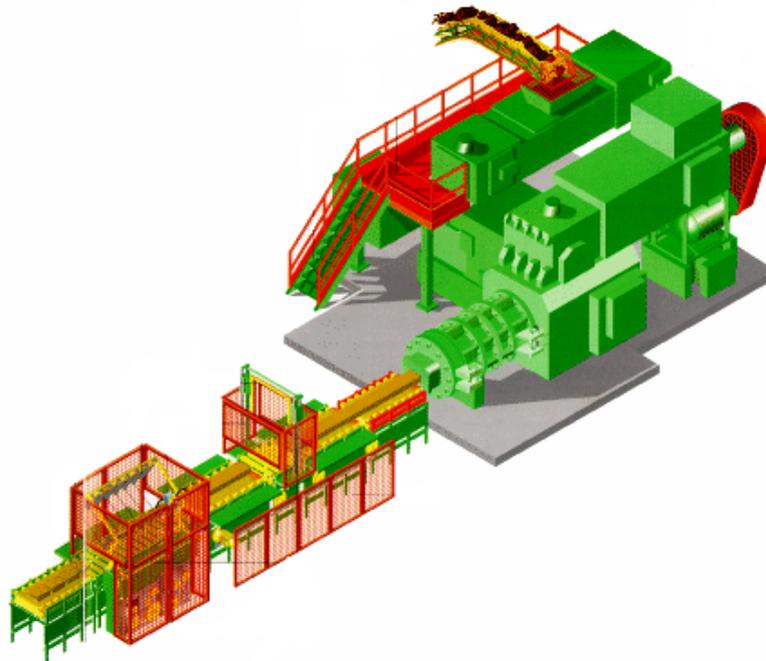


Figura 1.3: Estrusore con taglierina

Dopo questo processo il filone di materiale pressato che esce dalla filiera passa attraverso una taglierina che taglia il filone nelle dimensioni volute . Questo tipo di lavorazione per estrusione è indicato per la produzione di laterizi per muratura ,ad esempio per mattoni .Può essere usato anche per la produzione di laterizi di copertura ,che sono descritti in seguito.

### 1.4.2 Formatura per pressatura

Un altro metodo usato per la produzione di laterizi, è quello di formatura del laterizio per pressatura. L'argilla precedentemente trattata viene pressata secondo un profilo o forma mediante una pressa. Questo sistema non prevede l'uso della taglierina in quanto il laterizio è completamente formato. In generale è un sistema usato per argille che hanno una bassa percentuale di umidità intorno al 3% o 4%. Nella fig:1.4 viene riportata una pressa di questo tipo.

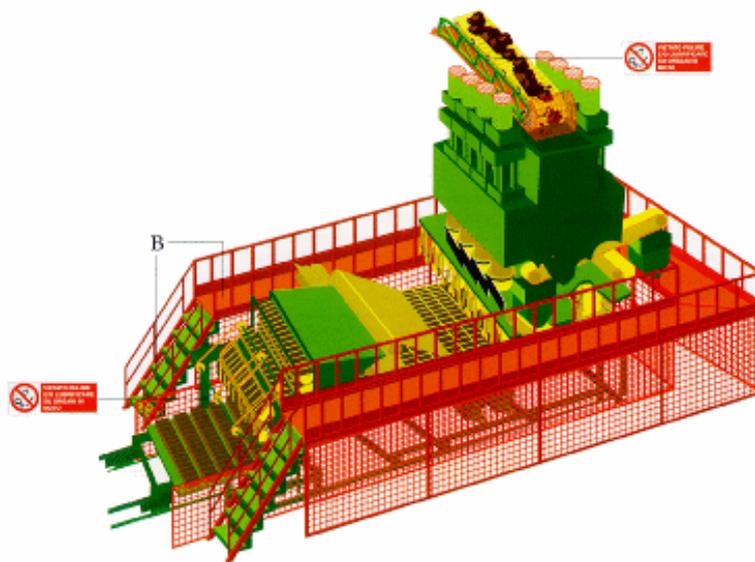


Figura 1.4: Mattoniera a pressa

Un altro tipo di lavorazione dei laterizi in cui viene usata la pressa, è quello per la produzione dei laterizi di copertura, esempio le tegole. In questo tipo di lavorazione l'argilla passa attraverso un estrusore il quale crea un filone con un determinato profilo. Il filone, come nel caso dei mattoni, passa attraverso una taglierina in cui viene tagliato in pezzi uguali. Questi pezzi prendono il nome di gallette e, a differenza dei mattoni non è un prodotto completamente formato, cioè necessita della ulteriore lavorazione che di pressatura al fine di ottenere la forma definitiva. A seguito di questi processi questa il prodotto è pronto per l'essiccazione.

## 1.5 Fase 4: Essiccazione

A seguito della formatura del laterizio si rende necessaria l'essiccazione come processo preparatorio a quello successivo della cottura. Con l'essiccazione le quantità d'acqua impiegate nella formatura non vanno però completamente eliminate, comunque il processo di essiccazione si ritiene concluso anche se nella massa del prodotto rimangono modeste percentuali di umidità. Per convenzione le percentuali di umidità si riferiscono al peso del prodotto essiccato ad una determinata temperatura. Il laterizio si definisce secco quando permane a peso costante per un determinato tempo ad una data temperatura. L'essiccazione si effettua per evaporazione dei liquidi contenuti al interno del prodotto. Il passaggio di stato, acqua-vapore, in qualunque condizione avvenga, è un fenomeno fisico, che si accompagna ad un forte assorbimento di calore. Ogni kg di acqua evaporata alle temperature medie degli essiccatoi industriali richiede da 580 - 600 Kcal. Una certa quantità di energia è necessaria per vincere le forze di legame dell'acqua con l'argilla, forze che divengono sensibilmente grandi solo in presenza di bassi frazioni di umidità. Esistono diverse tecniche e mezzi utilizzate per l'essiccazione dei laterizi ed in linea generale sono:

- Riscaldamento diretto del materiale e quindi del liquido contenuto.
- Riscaldamento indiretto o ventilazione di aria calda.

Per l'essiccazione è necessario un ambiente nel quale possano venire superate le condizioni di equilibrio di modo che il vapore generato dall'essiccazione venga assorbito da tale ambiente. Quindi come si può immaginare per il processo di essiccazione sono richieste notevoli quantità di energia. Di seguito vengono descritti i due metodi utilizzati per l'essiccazione.

### 1.5.1 Riscaldamento diretto

Negli essiccatoi si possono distinguere tre modi di riscaldamento diretto, che sono:

- Raggi infrarossi, coi quali per irraggiamento diretto ed in ambiente protetto, si trasmette calore alla superficie esterna del prodotto, per conduzione e quindi con ritardo, verso l'interno del materiale;
- Microonde, sono onde elettromagnetiche ad alta frequenza che inducono il riscaldamento nel prodotto con temperature maggiori all'interno.

- In autoclave, o ambiente chiuso, ove sia possibile un riscaldamento del prodotto, argilla ed umidità, a temperatura superiore a quella di ebollizione dell'acqua. La differenza rispetto al primo sistema, risiede nel fatto che, durante il riscaldamento, l'evaporazione rimane frenata dalla pressione che si forma all'interno dell'ambiente chiuso. Ciò rende possibile il raggiungimento della temperatura di ebollizione. L'apertura controllata di una valvola di comunicazione con l'esterno provoca nell'autoclave una diminuzione di pressione che mette in moto un'essiccazione molto rapida.

### 1.5.2 Riscaldamento indiretto

Con il metodo di riscaldamento indiretto, si utilizza la ventilazione ad aria calda che lambisce il prodotto, si sfrutta sia il fenomeno fisico della trasmissione di calore per convezione, sia la capacità dell'aria di assorbire il vapore. L'aria in questo processo, mentre cede calore per l'evaporazione ed il riscaldamento eventuale del prodotto, contemporaneamente si raffredda. Il processo ha un limite quando il raffreddamento dell'aria raggiunge la temperatura di saturazione umidità relativa del 100%, al di sotto della quale l'aria non può più assorbire il vapore. L'essiccazione con aria può avvenire in due modi:

- In ambiente atmosferico e quindi con lo sfruttamento della ventilazione e delle temperature naturali, nel qual caso il calore necessario all'evaporazione lo fornisce l'aria esterna.
- In ambiente chiuso a temperatura controllata mediante alimentazione di aria preriscaldata. L'essiccazione ad aria calda sfrutta facilmente tutte le possibilità di cessione di calore da parte del forno è rappresenta oggi il metodo industrialmente più usato per l'essiccazione del laterizio.

Questa fase del processo produttivo richiede una quantità di tempo notevole di permanenza del prodotto al interno dell'essiccatoio. In alcuni stabilimenti per essiccatoi di questo tipo l'aria calda usata per l'essiccazione viene prelevata dalla zona di raffreddamento del forno di cottura. Nel caso in cui l'aria calda proveniente dal forno non sia sufficientemente calda viene riscaldata ulteriormente mediante bruciatori, così da garantire una determinata temperatura.

## 1.6 Tipologie di essiccatoi

Gli essiccatoi per laterizi, secondo una classificazione generale, che distingue soltanto le caratteristiche fondamentali di ogni tipo, si dividono in:

- Continui
- Statici
- Semi continui

### 1.6.1 Essiccatoi continui

Gli essiccatoi continui sono costituiti da gallerie fig:1.5, contenenti un determinato numero di carrelli, nelle quali ad ogni introduzione dei carrelli effettuata da una estremità, deve corrispondere dall'altra un'estrazione. L'introduzione è l'estrazione avvengono in modo regolare e continuo, di giorno e di notte. Sotto l'aspetto funzionale interno gli essiccatoi continui sono caratterizzati da una distribuzione fissa, nel tempo e nello spazio, delle condizioni termoigrometriche ed è il prodotto stesso, nel suo cammino tra l'una e l'altra estremità, che può incontrare variazioni graduali di condizioni di essiccazione. Nella denominazione usuale di continuo, viene inteso un essiccatoio caratterizzato da un tempo di essiccazione compreso tra le 15 e le 40 ore. Questo tipo di essiccatoio necessita di scorte essendo un impianto che è in funzione 24 ore su 24.

### 1.6.2 Essiccatoi statici

Gli essiccatoi statici sono costituiti da gruppi di locali separati, chiamati normalmente camere od anche celle. Durante le ore di lavoro delle macchine le camere vengono riempite una dopo l'altra, in numero corrispondente al volume di produzione. A riempimento completato ogni camera si chiude e può iniziare il suo processo di essiccazione, caratterizzato dal fatto che il prodotto rimane fermo e si fanno cambiare nel tempo le condizioni termoigrometriche interne. Per ogni camera è possibile teoricamente l'effettuazione di un processo particolare di essiccazione essendo ciascuna camera indipendente dalle altre. La possibilità, di condurre regolazioni differenti per diversi tipi di prodotto, viene ritenuta un'importante prerogativa degli essiccatoi statici. Gli essiccatoi statici non hanno necessità di scorte, come gli essiccatoi continui, tuttavia essi si trovano legati strettamente all'orario di produzione delle macchine per le operazioni di carico e scarico.

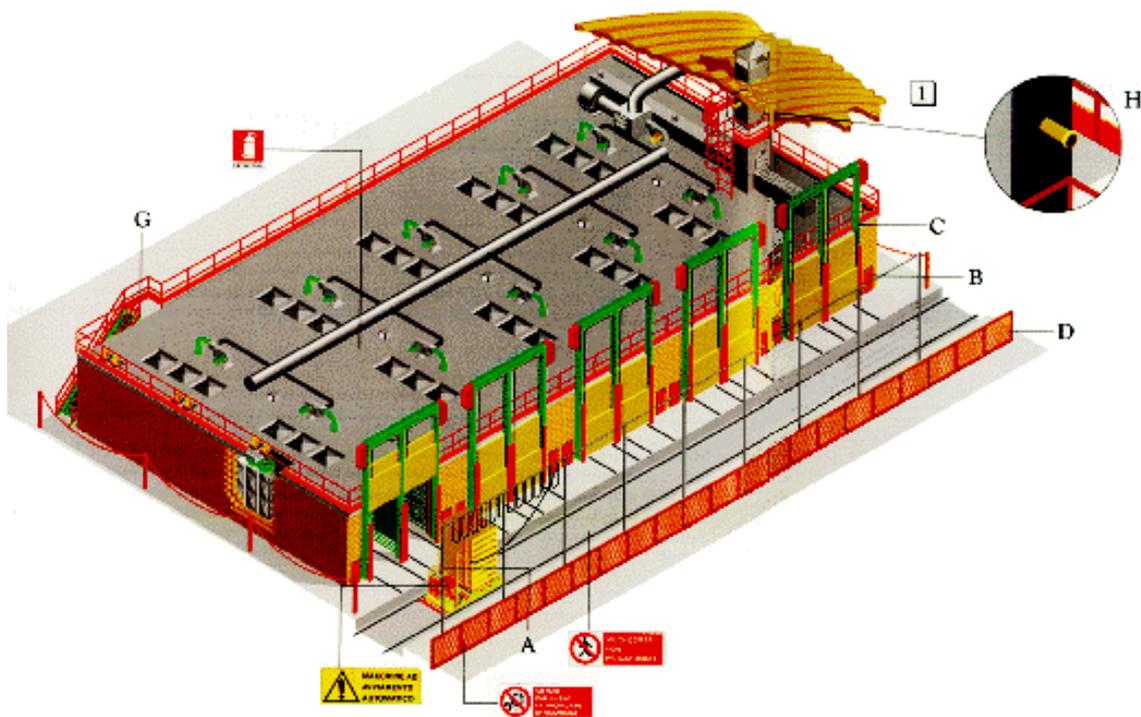


Figura 1.5: Essiccatoio continuo a riscaldamento indiretto

### 1.6.3 Essiccatoi semicontinui

Questo tipo di essiccatoio è simile al continuo, per quanto riguarda il carico del prodotto su carrelli, e il percorso degli stessi in galleria. L'essiccatoio semicontinuo si differenzia dal continuo, per il fatto che tutti i carrelli prodotti devono essere introdotti completamente nella galleria, senza deposito temporanei o riserva. Pertanto il ritmo di avanzamento dei carrelli è uguale a quello della produzione, mentre durante le ore di riposo delle macchine il funzionamento dell'essiccatoio semi continuo è simile a quello di un essiccatoio statico.

## 1.7 Fase 5: Cottura

Dopo l'essiccazione il prodotto è pronto per la cottura. Il prodotto viene sottoposto ad elevate temperature 800-1000°C, molto superiori a quelle dell'essiccatoio, per un determinato periodo di tempo in modo da fornire al prodotto la resistenza meccanica necessaria all'uso. Fino ad una decina di anni fa esistevano due tipo-

gie di forni: forni a tunnel 1.6é forni Hoffman. Al giorno d'oggi i forni a tunnel sono diventati lo standard segnando la fine dei forni Hoffman. Sempre nel corso degli anni anche il tipo di combustibile utilizzato per l'alimentazione dei forni è cambiato ,da olio combustibile e polvere di carbone si é passati ad alimentazione a gas metano meno inquinante e più economico. In generale in tutti i forni sia tunnel che Hoffmann si distinguono tre zone:

- Zona di preriscaldamento, antecedente la zona di cottura . Il prodotto inizia il suo riscaldamento a partire dalla temperatura ambiente e gradualmente si porta a circa 500-700C. Il calore è fornito al prodotto dalla combustione del gas che esce dalla zona di cottura ad alta temperatura, marcia in direzione del camino, e viene espulso a circa 50-100C.
- Zona centrale nella quale sono posizionati i bruciatori , si formano le più elevate temperature circa 800-1000C , è la zona dove il materiale viene effettivamente cotto.
- Zona di raffreddamento in cui il prodotto perde gradualmente temperatura e si porta fino a 50-100C. Aria fredda viene immessa alla fine della zona, in direzione di quella di cottura.L'aria riscaldata viene riciclata per l'essiccatoio

Il prodotto all'uscita dal forno possiede una temperatura tra 50 - 100 C .A questo punto il ciclo di lavorazione dal punto di vista della trasformazione chimico/fisica è terminato . Il prodotto viene indirizzato verso il confezionamento.

## 1.8 Fsaè 6: Confezionamento

Dopo la fase di cottura il prodotto cotto è pronto per essere confezionato. Il processo di manipolazione del laterizio per il confezionamento è in massima parte meccanizzato . Le macchine impiegate si possono suddividere in base al tipo di operazione svolta.

- Macchine per lo scarico dei pacchi di prodotto cotto, le quali provvedono a liberare i carrelli ed avviare il prodotto cotto verso i piazzali di deposito o le linee di confezionamento.
- Macchine per l'imballo dei pacchi di laterizio,

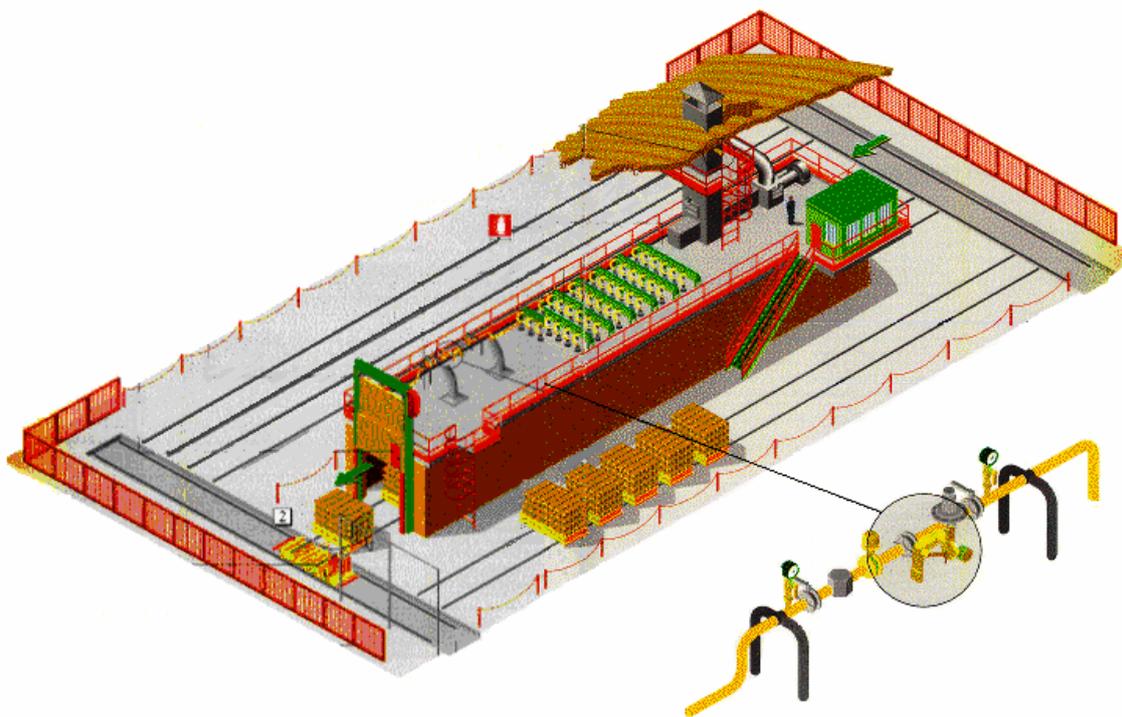


Figura 1.6: Forno a tunnel

Tutte le macchine di ultima generazione sono controllate elettronicamente da dispositivi programmabili, che condizionano tutti i movimenti in sequenze logiche e in successione, provvedendo alla sicurezza ed alla segnalazione dei guasti. Nella fig:1.8 vengono riportati due esempi di macchine utilizzate per il confezionamento.

Questa è l'ultima fase del processo di produzione dopo di che il prodotto confezionato viene stoccato in magazzino in attesa di essere spedito.

## 1.9 Sistema di movimentazione dei castelli o carrelli

Fino ad ora si sono descritti le singole fasi senza specificare come il prodotto viene spostato da una fase all'altra. La movimentazione del prodotto che alimenta ogni singola fase del processo produttivo è affidata a macchinari totalmente automatizzati. Il loro funzionamento è affidato a dispositivi elettronici che controllano i movimenti della macchina (PLC). Come già accennato prima la movimentazione della materia prima nella fase di prelaborazione è affidata a nastri

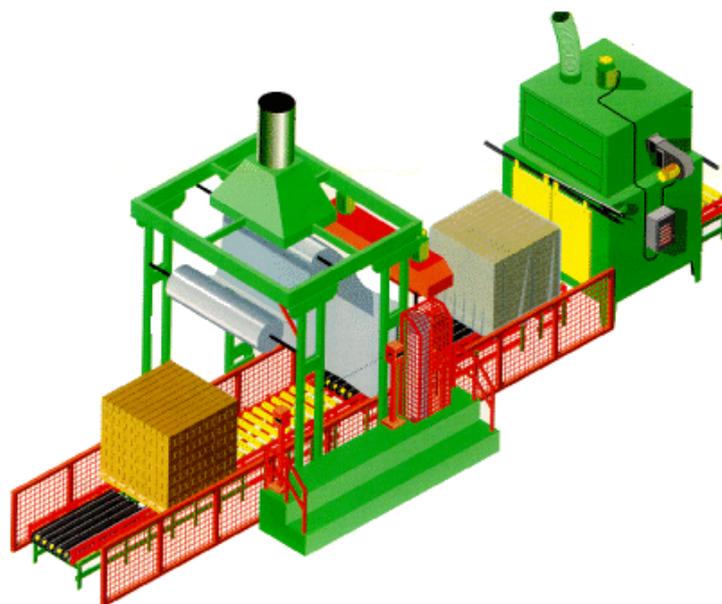


Figura 1.7: Macchina per il confezionamento con materiale termoretraibile



Figura 1.8: Due tipologie di macchine utilizzate per il confezionamento a) Reggiatrice verticale b) Reggiatrice orizzontale

trasportatori. Il funzionamento di questi nastri è legato al funzionamento della linea di prelaborazione, questo fino alla fase di taglio/pressatura dopo la quale la movimentazione è affidata a un sistema meccanizzato di carrelli e macchine automatiche. Di seguito è riportato un elenco di tali macchine:

- a) Macchine per lo spostamento del prodotto dai nastri trasportatori ai pianali. Vengono utilizzate per posizionare il prodotto appena formato sopra ai pianali del carrello. Questa macchina deve essere programmata in modo tale che, al variare delle dimensioni del prodotto (cambio modello) la macchina debba variare di conseguenza in modo da avere un posizionamento corretto. Esiste una macchina analoga che svolge il lavoro opposto posizionata all'uscita dell'essiccatoio (descritta in seguito) in quanto i pianali vengono caricati all'ingresso dell'essiccatoio e scaricati all'uscita dell'essiccatoio.
- b) Macchine per lo spostamento dei pianali sui carrelli. Viene usata per spostare i pianali caricati precedentemente sopra al carrello. È posizionata prima dell'essiccatoio di modo che prepari i carrelli pronti per l'essiccazione. Nella fig:1.9 viene visualizzata questa tipologia di macchina.

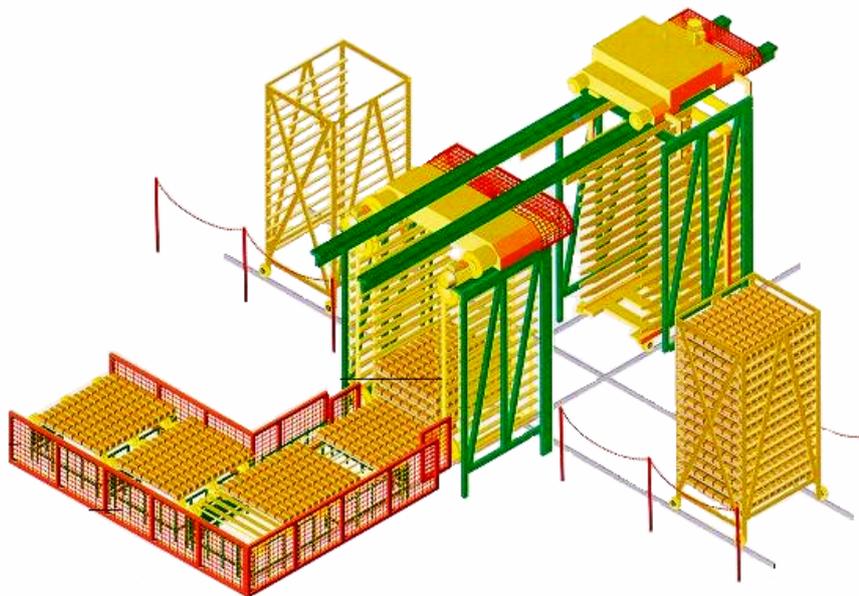


Figura 1.9: Macchine per lo spostamento dei pianali sui carrelli

- c) Macchine per la formazione dei pacchi di prodotto secco pronti per la cottura. Sono comunemente denominate impilatrici e sono impiegate per la sistemazione dei prodotti in pacchi prelevandoli dai pianali dei carrelli all'uscita dell'essiccatoio, posizionandoli direttamente sopra i carrelli del forno, ovviamente tale macchina é posta prima del forno. Deve esistere una macchina analoga posta all'uscita del forno che svolge il lavoro opposto a quello della impilatrice. Questa macchina é riportata nella fig:1.10

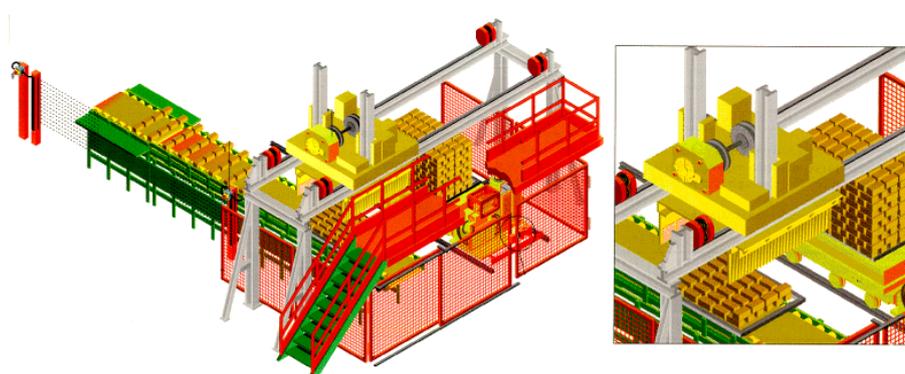


Figura 1.10: Macchine per la formazione dei pacchi di prodotto secco

- d) Macchine per la riformazione dei pacchi, preparazione al confezionamento. Viene usata per formare i pacchi di prodotto finito, pronto per il confezionamento. Fondamentalmente non é diversa dalla impilatrice come funzionamento, in piú questa macchina fa anché il confezionamento.

Il sistema di movimento dei carrelli é basato su rotaie e su un sistema di ganci con i quali i carrelli vengono ancorati. In base alle esigenze di produzione e organizzazione del lavoro é previsto anche un sistema di parcheggio per l'accumulo o scorte. Questo accumulo puó servire per alimentare l'impianto anche dopo la fine dell'orario di lavoro, in particolare per l'essiccatoio é forno nei quali si esegue un processo che richiede molto tempo, sono impianti che hanno bisogno di una alimentazione continua. Con questa ultima argomento si conclude la descrizione del processo di produzione dei laterizi. É da sottolineare che in base al budget delle aziende produttrici di laterizio alcuni dei macchinari possono non essere presenti nella linea produttiva sopra tutto nella parte di prelaborazione. Nei prossimo capitolo seguirá una descrizione dei componenti elettronici fondamentali utilizzati nell'automazione industriale.



# Capitolo 2

## Introduzione al PLC

All'inizio l'automazione industriale si realizzava con relé 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 relé eccitato o meno. Con il progredire del mondo industriale, soprattutto nel campo delle macchine automatiche, si è avuto il conseguente aumento dell'utilizzo dei circuiti elettrici di comando, per i quali, si sono iniziati ad intravedere i limiti. Le logiche cablate hanno infatti alcuni svantaggi, ed in particolare sono:

- La complessità di un quadro a logica cablata aumenta in modo esponenziale con il numero di utilizzatori da esso servite.
- La logica presenta un alto numero di cablaggi interni e di relé ausiliari che ne diminuiscono di fatto l'affidabilità.
- In caso di guasto, la sua ricerca è lunga e laboriosa, e questo si ripercuote sull'impianto controllato dal quadro stesso.
- Ogni relé o temporizzatore aggiunto necessita di maggiore spazio nel quadro, quindi automazioni complesse hanno necessita di quadri di grandi dimensioni.
- Nel caso siano necessarie modifiche o ampliamenti, si presentano lunghi tempi di fermo dell'impianto sia per studiare il sistema, sia per realizzare le modifiche;

Con l'applicazione dell'elettronica analogica seguita da quella digitale ,nel campo del automazione industriale il modo di pensare é fare automazione e cambiato, soprattutto con l'utilizzo di quest'ultima. Sono stati creati dispositivi elettronici per rimediare ai problemi su indicati. Questi dispositivi sono i PLC (Programmable logic controller ) e DCS (Distribuited Controller System ). Con questi nuovi componenti è cambiato il dimensionamento elettrico, ma i concetto base della logica a due stati e comunque rimasto invariato. In pratica il relé è stato sostituito con un bit di memoria in quanto il suo funzionamento è a due stati, acceso o spento. La tabella 2.1 mostra la transizione dall'elettrotecnica all'elettronica digitale. Con questo non si intende dire che i relé non vengano più utilizzati al interno di quadri elettrici,ma vengono utilizzati in misura minore ,questo vale per tutti quei componenti che possono essere simulati dal PLC o DCS.

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

Figura 2.1: Transizione dal elettrotecnica al elettronica digitale

I PLC hanno dimostrato da subito grande versatilità e affidabilità tanto da diventare il dispositivo più usato per l'automazione industriale superando di fatto i DCS. La loro caratteristica fondamentale sta nel fatto che pur essendo dispositivi elettronici, e quindi funzionanti a bassa tensione non presentano difficoltà di funzionamento in ambienti industriali con notevoli disturbi. I vantaggi di avere un sistema di controllo governato da un programma , anziché vincolato da un cablaggio fatto di relé , temporizzatori , ed altri circuiti ausiliari , sono notevoli. In primo luogo tutti i segnali fanno capo al PLC ed é quindi immediata la ricerca di guasti o malfunzionamenti dovuti , ad esempio , a contatti incerti o collegamenti interrotti. In secondo luogo è possibile aggiornare il programma con una operazione che richiedono pochi secondi, mentre con il quadro elettrico cablato si dovrebbe mettere fuori servizio l'impianto per giorni interi, che nel caso di stabilimenti si traduce in perdite economiche per mancata produzione. In conclusione , la storia dimostra che i concetti per rendere automatiche le macchine non sono cambiati, ma sono cambiati i mezzi per metterle in pratica. Al giorno

d'oggi l'automazione è incentrata su dispositivi elettronici a microprocessore, per cui il buon funzionamento di qualunque macchina o processo è tutto basato sul software eseguito dal processore stesso. In questa tesi vengono trattati solo sistemi di controllo ad anello chiuso (vedere appendice E) in quanto esistono sistemi di controllo ad anello aperto. La differenza tra i due è che il primo sistema riceve un *feedback* dai vari sensori, mentre il secondo non ha questo segnale di ritorno.

## 2.1 Funzionamento del PLC

Indipendentemente dal produttore il principio di funzionamento di un PLC è generalmente lo stesso per tutti i modelli. Fondamentalmente il PLC è un calcolatore basato su microprocessore con architettura *RISC*, possiede capacità di calcolo ridotte, in compenso ha una grande affidabilità. In linea generale tra le CPU utilizzate nei PLC si possono individuare due grandi famiglie:

- PLC dotati di microprocessori *generalpurpose*, e quindi liberamente reperibili sul mercato, non espressamente progettati per una specifica funzionalità.
- PLC dotati di microprocessori *specialpurpose*, 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, le quali possono permettersi di impiegare risorse per progettare un microprocessore adatto alle esigenze della loro clientela. Il PLC non possiede un sistema operativo come quello dei PC, in quanto non deve interfacciarsi direttamente con l'uomo. Il suo sistema operativo è integrato con il *bios* (basic input output system) e insieme formano un sistema che viene indicato anche come sistema *embedded*. Questo software viene chiamato solitamente *firmware* e risiede nella ROM (read only memory). In pratica è un sistema *real time*, la maggior parte di essi sono a singolo processo anche se ultimamente sono stati messi in commercio dei modelli che adottano il multi processo. Al momento del passaggio dallo stato di *stop* a quello di *run*, il PLC inizia l'esecuzione di un ciclo (loop) che viene

ripetuto a intervalli di tempo regolari. All'avvio del processo come primo passo vengono controllati i dati di sistema poi viene caricato il programma utente, dopo di che vengono acquisiti i segnali degli ingressi ed elaborati in base al programma utente, infine si inviano i segnali in uscita. L'immagine in fig:2.2 riporta lo schema a blocchi del funzionamento di un PLC.

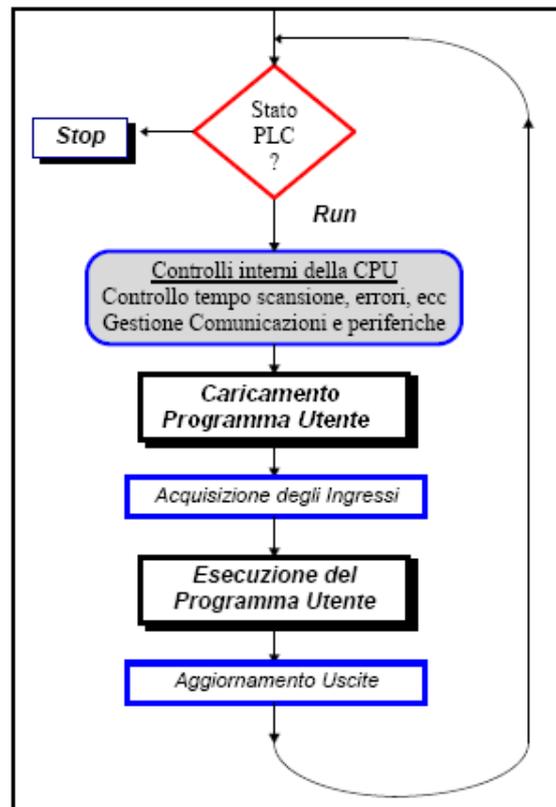


Figura 2.2: Lo schema descrive il funzionamento del PLC

Come già accennato, quando il PLC è in funzione (*run*), il programma utente viene eseguito a intervalli di tempo regolari, questo fa sì che il programma utente sia sempre in esecuzione, deve essere vero sempre in quanto il PLC controlla di fatto la macchina o l'impianto al quale è collegato. Naturalmente quando è in funzione il PLC non si occupa solo dell'esecuzione del programma utente, ma si occupa anche di altre funzioni di seguito elencate:

- Controllo di errori generati nell'esecuzione del programma.
- Controllo nella durata dell'esecuzione del programma.

- Controllo dello stato del hardware, ossia di tutte le periferiche che lo compongono.
- Aggiornamento dello stato di memoria riferito agli ingressi e alle uscite.
- Comunicazione con altri PLC o altri dispositivi connessi alle sue porte di comunicazione

Quando il PLC passa dallo stato di *run* a quello di *stop* e dovuto principalmente ad errori irreversibili nel programma o nei componenti elettronici interni di conseguenza il PLC deve interrompere l'esecuzione normale del programma. In questi casi è richiesto l'intervento del utente per rimettere in funzione la macchina ed accertare le cause del errore. Una causa classica di errore (*fault*) di un PLC è l'esecuzione di una operazione aritmetica che tenti una divisione per zero . Il funzionamento dei PLC è basato su *interrupt*, al verificare di una determinato evento sospende l'esecuzione normale del ciclo e avvia la *routine* associata la tipo di interrupt. L'organizzazione del codice/programma utente varia da produttore a produttore , alcuni organizzano il codice e i dati in blocchi , altri riservano questa sistema solo per il codice lasciando i dati in un unica area di memoria. Negli ultimi anni è stata introdotta la tecnica di programmazione *object oriented* in alcuni modelli di PLC con notevoli vantaggi sia nella programmazione che nella qualità del programma.

## 2.2 I/O del PLC

Un'altra caratteristica interessante dei PLC sta nel fatto che sono sistemi modulari, nel senso che in base a esigenze di progettazione si può aumentare o diminuire il numero degli ingressi o delle uscite semplicemente aggiungendo o rimuovendo moduli di ingressi-uscite. Sempre in base a esigenze di progettazione si deve scegliere il modello di PLC che supporti un numero di I/O richiesti dal progetto. Il numero massimo di I/O che un PLC può gestire , spesso chiamata anche *taglia* del PLC. Oggi in commercio si trovano diverse tipologie di PLC che vanno dai micro PLC i quali gestiscono un numero ridotto di I/O fino ad arrivare a PLC che gestiscono alcune migliaia di I/O .È ovvio che anche la capacità di calcolo è la dimensione ed il prezzo variano in base alla loro capacità. Esistono due tipologie di moduli ingressi-uscite che sono:

- Ingressi - uscite digitali.

- Ingressi - uscite analogiche.

Prima di discutere dei tipi di ingressi e uscite, bisogna descrivere i tipi di dato utilizzati dai PLC. Gli ingressi o uscite devono essere identificate da un tipo di dato di modo da riservare un'area di memoria adeguata alla memorizzazione del valore.

### 2.2.1 Tipi di dati

Nella norma IEC 1131-3 [3] si trovano oltre 20 tipi di dati che possono essere gestiti da un PLC, soprattutto per abbracciare un po' tutte le variabili usate dai vari produttori. Ogni ingresso e uscita viene identificato all'interno del PLC tramite il nome assegnatogli ed il tipo di dato, ad esempio per rappresentare un I/O digitale viene usato il tipo di dato *Booleano* indicato con *BOOL* in quanto serve un solo bit per memorizzare lo stato dell'ingresso o uscita. Per gli I/O analogici solitamente vengono usate le *Word* (valore rappresentabile in 16 bit) in quanto bisogna rappresentare un valore numerico che varia all'interno di un determinato range. La tab:2.3 mostra un elenco dei tipi di dati più utilizzati nella programmazione dei PLC.

## TIPI PRIMITIVI

	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	$2^{64}$
SINT	Short integer 8 bit	-128 + +127
USINT	Unsigned Short integer 8 bit	0 + 255
INT	Integer 16 bit	-32768 + +32767
UINT	Unsigned Integer 16 bit	0 + 65535
DINT	Double Integer 32 bit	$-2^{31} + 2^{31} (-1)$
UDINT	Unsigned Double Integer 32 bit	0 + $2^{32} (-1)$
LINT	Long Integer 64 bit	$-2^{63} + 2^{63} (-1)$
ULINT	Unsigned Long Integer 64 bit	0 + $2^{64} (-1)$
REAL	Floating Point Number 64 bit	$-10^{-38} + 10^{38}$
LREAL	Long real Number 128 bit	.....

## ALTRI TIPI DI DATI USATI DAL PLC

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

Figura 2.3: La tabella include i tipi di dati più utilizzati dai PLC.

## 2.2.2 I/O Digitali

Per ingressi digitali di un PLC si intendono quegli ingressi ai quali è possibile inviare segnali che assumono solo due valori ,come ON / OFF .Esempi di apparecchiature che possono inviare questo tipo di segnale sono il termostato , pressostato, fincorsa, pulsante ecc. Normalmente per gli ingressi digitali si utilizza la tensione 24V in corrente continua ,quindi nel quadro elettrico insieme al PLC si rende necessario anche l'installazione di un alimentatore dedicato. Per separare i circuiti interni della CPU con la tensione proveniente dall'impianto, ogni ingresso digitale è dotata di appositi optoisolatori (detti anche fotoaccoppiatori) fig:2.4, che resistono a differenze di potenziale anche di 1500 V (tensione di

isolamento).

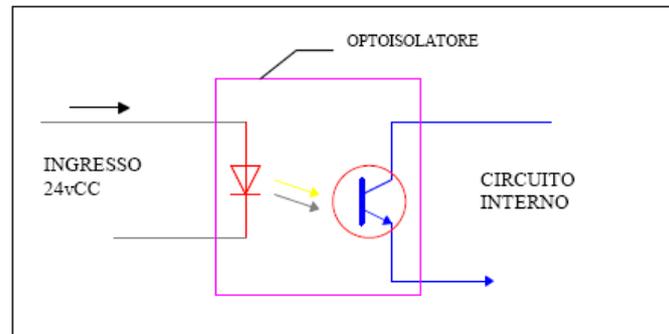


Figura 2.4: Schema di in ingresso digitale con optoisolatore

Quando un segnale arriva ad un ingresso digitale il diodo led emette luce , dal altra parte del optoisolatore e presente un elemento fotosensibile il quale in presenza di luce genera i livelli di tensione o segnale che viene interpretato dalla CPU. Per quanto riguarda le uscite digitali il segnale viene generato internamente e deve essere portato ai dispositivi esterni . Viene considerato alto o *TRUE* un segnale che varia tra 24 e 28 v ,viene considerato basso o *FALSE* un segnale che si aggira intorno ai 5 e 12 v. Gli ingressi digitali non possono acquisire segnali che variano troppo velocemente nel tempo , generalmente un ingresso perché venga letto dal programma deve permanere almeno 0.5 secondi.

### 2.2.3 I/O Analogiche

Nelle automazioni industriali si presenta spesso il problema di misurare e di gestire temperature, pressioni, pesi, portate ed altre variabili legate a fenomeni fisici. Queste misure vengono eseguite da appositi strumenti detti trasduttori i quali trasformano i valori rilevati in segnali elettrici. Successivamente questi segnali elettrici vengono acquisiti dal sistema di controllo convertiti a valori di tensione interpretabili dal PLC in modo corretto. È in base all'elaborazione di questi segnali il PLC comanda degli attuatori che possono essere motori elettrici, sotto inverter ect. Quindi per la misura di queste grandezze ce bisogno di due componenti i trasduttori e i convertitori. Negli strumenti di misura generalmente il trasduttore converte la grandezza fisica in un segnale elettrico, ed il convertitore che adatta il segnale elettrico del trasduttore agli standard dei segnali analogici. Talvolta , per alcune misure di temperatura si usa installare nell'impianto

solamente il trasduttore, mentre la conversione e linearizzazione del segnale é realizzata all'interno del sistema di controllo come mostrato dalla fig:2.5.

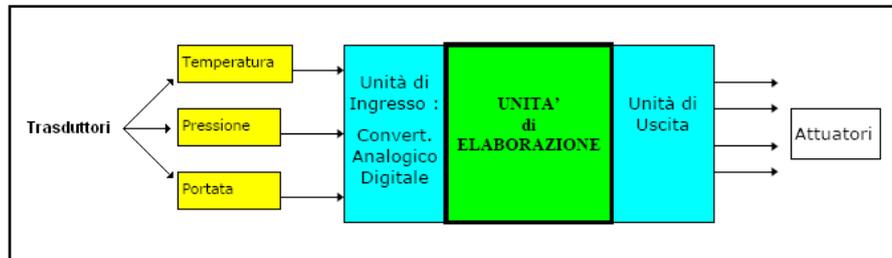


Figura 2.5: Schema delle misure analogiche

I segnali analogici a differenza di quelli digitali in quanto assumono qualunque valore elettrico nel intervallo (range) di lavoro. Un segnale digitale per definizione assume solo due valori : alto che rappresenta lo stato 1 (ad esempio 24 V) e basso che rappresenta lo stato 0 (ad esempio 0 V). Un segnale analogico invece può assumere tutti i valori compresi tra i due estremi del campo di lavoro , cosicchè può rappresentare il valore di una grandezza fisica così com'è nella realtà. Esempi di trasduttori di o misuratori di grandezze fisiche sono :

- Sonde di Temperatura.
- Misuratori di Pressione e di Portata.
- Carico (misure di peso).
- Sonde di umidità.
- Sensori di torsione e sforzo meccanico in generale
- Sensori di Ossigeno, Ossido di Carbonio, ed altri rivelatori di chimiche.

I segnali analogici sono sempre in corrente continua ed i più utilizzati sono sostanzialmente questi due :

- In Tensione 0 - 10 V.
- In Corrente 4 - 20 mA.

Il segnali in tensione sono quelli più semplici da utilizzare e con minore costo nei dispositivi che li devono gestire. Possono percorrere pochi metri e sono facilmente soggetti a disturbi da campi elettromagnetici. Vengono usati per rilevare le temperature negli impianti di riscaldamento e condizionamento, dove i sensori sono posti a pochi metri dalle centraline di controllo, e quindi la lunghezza dei cavi non supera i 15-20 metri. I segnali più utilizzati per le misure analogiche è il segnale in corrente da 4-20 mA. Questo tipo di trasmissione presenta elevata immunità a disturbi elettromagnetici, una buona tolleranza alle fluttuazioni nella tensione di alimentazione, la possibilità di compiere tratte molto lunghe (anche 100-200 metri) senza bisogno di alcun particolare accorgimento, in pratica è un tipo di segnale più stabile. Un I/O analogico viene identificato all'interno del PLC dal suo nome e dal tipo di dato. Il nome può essere il nome dell'indirizzo dell'I/O o un nome *mnemonico* assegnato dal progettista ad un area dati riferita a quell'ingresso. Nel caso degli ingressi digitali si usa il tipo di dato *BOOL*, nel caso di un ingresso analogico solitamente viene usato il tipo di dato *WORD* (16 bit) nella quale viene memorizzato il valore letto dalla sonda. Nei sistemi più evoluti e nel caso in cui sia richiesta una maggiore precisione possono essere usati anche variabili *REAL* (64 bit) per la memorizzazione del valore. Il PLC, normalmente per ogni ingresso analogico, assegna 16 bit una Word ma di questi 16 bit ne destina solo alcuni alla rappresentazione del segnale in ingresso. Ad esempio un bit viene utilizzato per il segno +/-, mentre un 1 bit serve per indicare eventuali errori o fuori scala. Questo comporta delle approssimazioni nella conversione da segnale analogico a digitale, ma più bit vengono utilizzati migliore è la risoluzione (precisione della misure). Un ingresso analogico con convertitore a 14 bit ha una risoluzione discreta, quindi assicura una buona qualità della conversione.

## 2.3 Memoria del PLC

Come tutti i sistemi di calcolo anche i PLC sono dotati di memoria. L'organizzazione della memoria al interno dei PLC in generale è simile a quella dei PC, nel senso che anche i PLC possiedono una memoria RAM (random access memory), una memoria ROM (read only memory) ed in fine una memoria di massa che in genere è un memoria di tipo EPROM o EEPROM. Diseguito si da breve descrizione delle caratteristiche principali di queste memorie.

### 2.3.1 La memoria di lavoro RAM

Questa memoria è integrata nel PLC e non è ampliabile la sua dimensione generalmente varia in base al modello di PLC. Questa memoria prende il nome di RAM (random access memory) è la memoria di lavoro in cui viene caricato il programma utente. La RAM deve essere una memoria a ritenzione, nel senso che i valori in essa contenuti devono essere mantenuti anche dopo una interruzione dell'alimentazione. Questa memoria viene usata per memorizzare i valori delle variabili quali: i temporizzatori, contatori, valori di processo, immagine degli I/O, unità parametrizzabili ect. La ritenzione viene effettuata mediante una batteria chiamata anche batteria tampone, i parametri e il contenuto della memoria RAM del PLC vengono mantenuti finché la tensione della batteria rientra nella tolleranza. Alcuni modelli di PLC realizzano la ritenzione mediante l'utilizzo di un condensatore ad alta capacità, ovviamente questo sistema ha un'autonomia molto inferiore al uso della batteria ed in alcuni casi non tutta la RAM è a ritenzione.

### 2.3.2 La memoria ROM

Nella memoria ROM è contenuto il *firmware*, cioè il programma base che permette alla CPU di funzionare ed eseguire il programma utente. Questa memoria non può essere scritta o cancellata dall'utilizzatore del PLC. È in tale memoria che sono memorizzati i valori di default (ossia di fabbrica) che stabilisce il costruttore per le variabili interne del PLC. In alcuni PLC di più moderna concezione, le CPU non vengono equipaggiate con memorie ROM, quindi il Sistema Operativo deve essere caricato in RAM alla prima accensione.

### 2.3.3 La memoria di massa

I programmi per i PLC vengono generalmente elaborati tramite un personal computer, quindi compilati e trasferiti nella memoria del PLC. Tale memoria è normalmente una EEPROM (Electrically Erasable Programmable Read Only Memory), ossia una memoria che non permette la cancellazione del suo contenuto se non tramite impulsi elettrici generati dal PC in fase di trasferimento chiamato programmatore di EEPROM. Questo garantisce il mantenimento del software in qualunque condizione di funzionamento, sia che stia funzionando sia che sia spento: tale tipo di memoria viene detta non volatile a differenza della RAM. Le EEPROM hanno soppiantato le più obsolete EPROM (dette anche UVPRM)

che non permettevano la cancellazione del programma se non esponendole ad una lampada a raggi ultravioletti. Dato che normalmente l'accesso in scrittura o in lettura di una memoria EEPROM richiede un tempo superiore a quello della memoria RAM. Alcuni costruttori per velocizzare l'esecuzione del programma hanno fatto in modo che il processore, all'accensione, copi il programma utente e dalla EEPROM alla RAM. Altri costruttori, per economizzare i costi, utilizzano solamente la memoria RAM, nella quale il software utente permane grazie alla batteria tampone. Negli ultimi tempi si è cominciato ad utilizzare come unità di memoria le memory card (MMC). Utilizzando la MMC la batteria viene eliminata, i dati vengono salvati direttamente in MMC, la ritenzione viene fatta solo per una parte della memoria RAM mediante un condensatore.

## 2.4 Protocolli di comunicazione tra PLC

Essendo gli impianti di produzione dei laterizi suddivisi per zone o reparti (vedi capitolo 1) di conseguenza anche il sistema di controllo è suddiviso per zone, nel senso che non viene utilizzato un sistema di controllo centralizzato ma bensì un sistema distribuito comunicante via rete. Nel caso il reparto non sia particolarmente esteso può essere sufficiente un solo sistema di automazione: nel caso invece il reparto sia di dimensioni estese possono essere presenti più sistemi di automazione: per esempio un PLC per ogni macchina come mostrato in fig:2.6. I vari sistemi sono interconnessi da reti di diversa tipologia a seconda delle funzionalità richieste. Adottando una suddivisione di questo tipo (un PLC per ogni reparto/macchina) è possibile intervenire sulla singola macchina per manutenzione ordinaria o straordinaria o per la messa in servizio senza bloccare completamente gli altri reparti. Gli svantaggi principali di un sistema di controllo centralizzato sono la distanza, la difficoltà nella gestione e manutenzione, difficoltà nelle modifiche dovute alla maggiore complessità ed inoltre richiede un tempo maggiore nel ciclo di esecuzione ecc. La distanza che il segnale deve percorrere da e verso il PLC in sistemi centralizzati può essere molto grande di conseguenza il segnale viene degradato dai disturbi elettromagnetici che ci possono essere inoltre anche dalla resistenza del mezzo trasmissivo. Quindi si preferisce usare PLC locali che controllano una parte del impianto. Per quanto riguarda il tempo di esecuzione, in alcuni modelli di PLC questo tempo varia in base al numero di istruzioni che la CPU deve eseguire. In pratica questo tempo dipende dalle dimensioni del programma, e ovvio che anche in questo caso esiste un limite oltre al quale non è possibile andare, questo per ragioni di efficienza. Tutti control-

lori locali (chiamati anche slaves) vengono messi in comunicazione con uno o più PLC centrali (concentratori di dati chiamati anche Master), in pratica viene formata una rete a due livelli. Al primo livello si trova il master e al livello sottostante ci sono tutti gli slave.

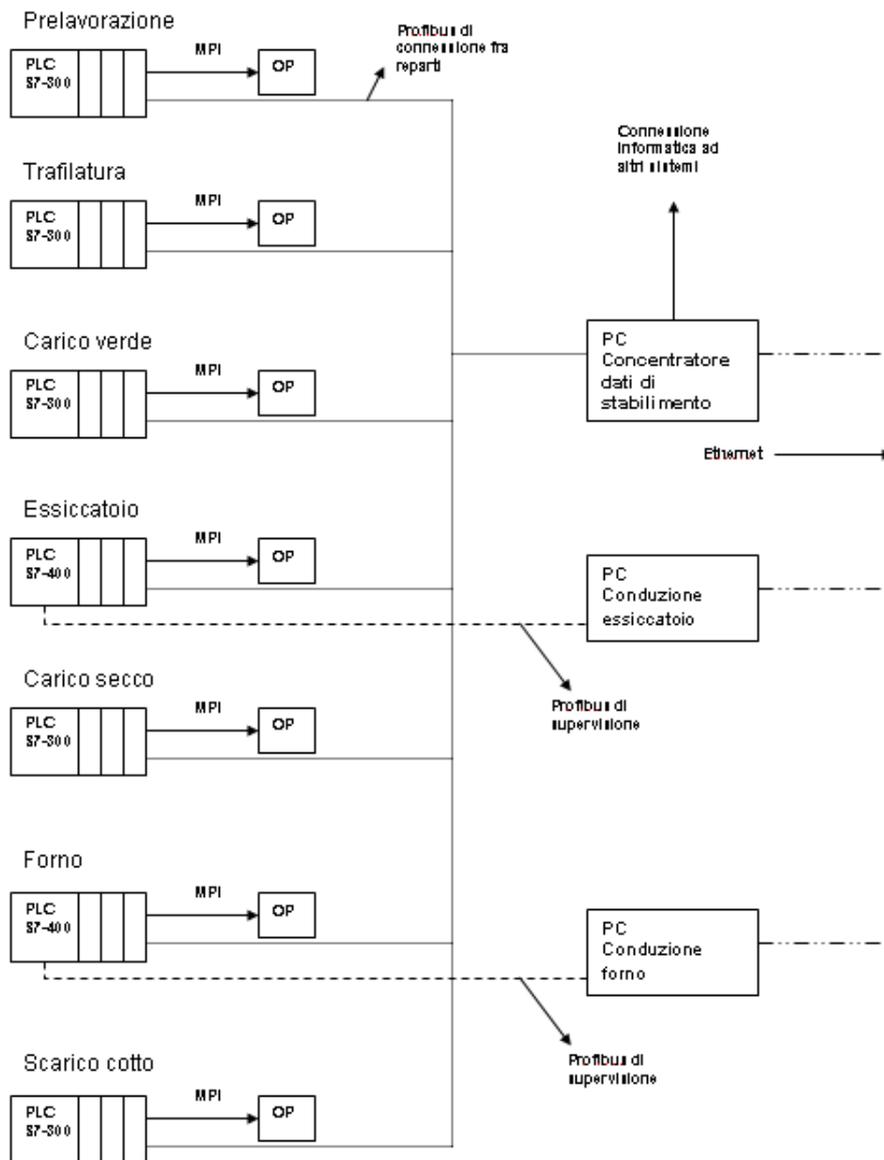


Figura 2.6: Schema di una rete di PLC utilizzate per il controllo di un impianto per la produzione dei laterizi.

La trasmissione dei dati da parte degli slave avviene solo su interrogazione del

master. Le interrogazioni consistono in lettura dei dati locali così da avere un'immagine globale di tutto il processo produttivo, o scrittura nel senso che il master può inviare dei comandi agli slave. Le tipologie di rete di PLC che si possono formare sono le seguenti.

- Configurazione *Point to Point*: la configurazione punto a punto è la più semplice che si può utilizzare, connette due PLC direttamente.
- Configurazione ad *Anello*: i PLC sono connessi in modo tale da formare un anello, in pratica è una sequenza di connessioni punto a punto. Con questa configurazione si raggiungono grandi distanze, ma ha lo svantaggio che nel caso di guasto di un nodo si interrompe la comunicazione in tutta la rete.
- Configurazione a *Stella*: nella configurazione a stella tutti i nodi sono connessi ad un accoppiatore centrale che gestisce l'intera comunicazione. Il guasto dell'accoppiatore mette fuori uso tutta la rete, mentre il guasto di un nodo non ha effetti rilevanti sul resto della rete.
- Configurazione a *Bus* condiviso: in questa configurazione il mezzo trasmissivo viene condiviso da più nodi, la connessione viene effettuata con appositi connettori. Il guasto di un nodo non ha effetti rilevanti sul resto della rete in quanto non interferisce con il resto della comunicazione.

Oltre alle reti master-slave esistono le reti multi-master. In questo tipo di rete i master possono essere connessi con una delle configurazioni su indicate. L'accesso al canale di comunicazione da parte dei master viene realizzato con il protocollo *token passing*, in pratica un master è abilitato a interrogare gli slave solo quando riceve il token. La procedura *token passing* garantisce che l'accesso al *bus* viene riservato o concesso solo al master che in quel momento possiede il token. È ovvio che l'intervallo di tempo di ritenzione del token è costante; quindi il tempo che ogni master dovrà attendere per trasmettere dipenderà dal numero di dispositivi attivi presenti nella rete in più dal massimo tempo di utilizzo del bus permesso (*token hold time*). Durante la fase di inizializzazione del sistema, il compito del sottoprogramma MAC (Medium Access Control, sottolivello del livello 2) di ogni dispositivo master è di definire il proprio indirizzo all'interno dell'anello logico costituito da tutti i dispositivi attivi. Il token viene passato da un master all'altro secondo l'ordine prefissato dell'anello logico. Uno dei canali di comunicazione o *layer fisico* nei PLC più usati negli anni scorsi era lo standard RS-232 connettore serial

(vedi appendice C). Data la sua limitazione in fatto di distanza tra due nodi (max 15 m) , è del fatto che questo standard permette solo connessioni punto a punto, sono stati sviluppati dei nuovi protocolli seriali. Attualmente vengono usati gli standard RS422,RS485 (vedi appendice C) con trasferimento dei dati in modalità asincrona ( vedi appendice B). Anche questi standard utilizzano un'interfaccia seriale con connettori a 9 poli .I dati trasmessi vengono codificati solitamente con la codifica NRZ ( Non return to zero ) o *Manchester* (vedi appendice A).Ogni azienda produttrice di sistemi di controllo ha sviluppato dei protocolli dedicati per il proprio prodotto ,di conseguenza attualmente esistono svariati protocolli di comunicazione che sono incompatibili tra prodotti di differenti marche.La tabella:??protocolli) mostra un elenco dei protocolli di comunicazione utilizzati da ogni produttore.

Per ovviare al problema di comunicazione dovuto all'incompatibilità dei protocolli di comunicazione di diversi produttori si è cercato di standardizzare alcuni di questi protocolli . Il protocollo standardizzato attualmente per la comunicazione di rete tra PLC sono il *PROFIBUS*(Processed Field Bus) seguito da *MODBUS*. Nella prossima sezione viene data una breve descrizione del protocollo *PROFIBUS* in quanto attualmente è il protocollo standard più utilizzato.

## 2.5 Il protocollo PROFIBUS

Profibus è un *bus* di campo fortemente voluto da Siemens, il cui sviluppo ed amministrazione è stato ufficialmente affidato ad un'organizzazione denominata *Profibus Trade Organization (PTO)*. La tecnologia Profibus è nata come standard tedesco secondo la normativa *DIN 19245* nel 1991, successivamente nel 1996 è stata riconosciuta conforme allo standard europeo *EN 50170*. Profibus è riconosciuto come protocollo di tipo 3 dalla normativa *IEC*(International Electrotechnical Commission) 61158 (Digital data communication for measurement and control Fieldbus for use in industrial control Systems). La norma *IEC 61158* descrive il bus di campo secondo i livelli ISO/OSI, mentre la norma *IEC 61784* (Profile sets for continuous and discrete manufacturing relative to fieldbus use in industrial control systems) descrive i profili che risiedono sopra il livello 7; Profibus occupa il *CPF* (Communication Profile Family) 3 e in particolare *CPF3/1* si riferisce alla versione *Profibus-DP*, *CPF3/2* si riferisce a *Profibus-PA* mentre *CPF3/3* è dedicato a *Profinet* un'evoluzione del *Industrial Ethernet* il quale non viene trattato in questa tesi. La struttura a strati del protocollo PROFIBUS è mostrata nella fig:2.8.

Produttore		Protocollo
SIEMENS	SIMATIC S7	<ul style="list-style-type: none"> <li>• PPI</li> <li>• MPI</li> <li>• PROFIBUS</li> <li>• (Ethernet) TCP/IP</li> </ul>
	SIMATIC S5	<ul style="list-style-type: none"> <li>• AS 511</li> <li>• PROFIBUS</li> </ul>
	SIMATIC 500/505	<ul style="list-style-type: none"> <li>• NTP</li> <li>• PROFIBUS</li> </ul>
	Protocollo SIMATIC HMI HTTP	<ul style="list-style-type: none"> <li>• (Ethernet) HTTP/HTTPS</li> </ul>
	SIMOTION	<ul style="list-style-type: none"> <li>• PROFIBUS</li> </ul>
OPC		<ul style="list-style-type: none"> <li>• DCOM</li> </ul>
Allen-Bradley		Serie PLC SLC500, SLC501, SLC502, SLC503, SLC504, SLC505, MicroLogix e PLC5/11, PLC5/20, PLC5/30, PLC5/40, PLC5/60, PLC5/80 <ul style="list-style-type: none"> <li>• DF1 <sup>2)</sup></li> <li>• DH+ tramite DF1 <sup>3)</sup></li> <li>• DH485 tramite DF1 <sup>4)</sup></li> <li>• DH485 <sup>4)</sup></li> </ul>
GE Fanuc Automation		Serie PLC 90-, 9090 <ul style="list-style-type: none"> <li>• SNP</li> </ul>
LG Industrial Systems (Lucky Goldstar) / IMO		Serie PLC GLOFA GM (GM4, GM6 e GM7) / Serie G4, G6 e G7 <ul style="list-style-type: none"> <li>• Dedicated communication</li> </ul>
Mitsubishi Electric		Serie PLC MELSEC FX e MELSEC FX0 <ul style="list-style-type: none"> <li>• FX</li> </ul>
Mitsubishi Electric		Serie PLC MELSEC FX0, FX1n, FX2n, AnA, AnN, AnS, AnU, QnA e QnAS <ul style="list-style-type: none"> <li>• Protocollo 4</li> </ul>
OMRON		Serie PLC SYSMAC C, SYSMAC CV, SYSMAC CS1, SYSMAC alpha e CP <ul style="list-style-type: none"> <li>• Hostlink/Multilink (SYSMAC Way)</li> </ul>
Schneider Automation (Modicon)		Serie PLC Modicon 984, TSX Quantum e TSX Compact <ul style="list-style-type: none"> <li>• MODBUS RTU</li> </ul>

Figura 2.7: Elenco di protocolli di comunicazione tra PLC

La famiglia Profibus consiste di tre protocolli fondamentalmente compatibili tra loro i quali sono descritti di seguito.

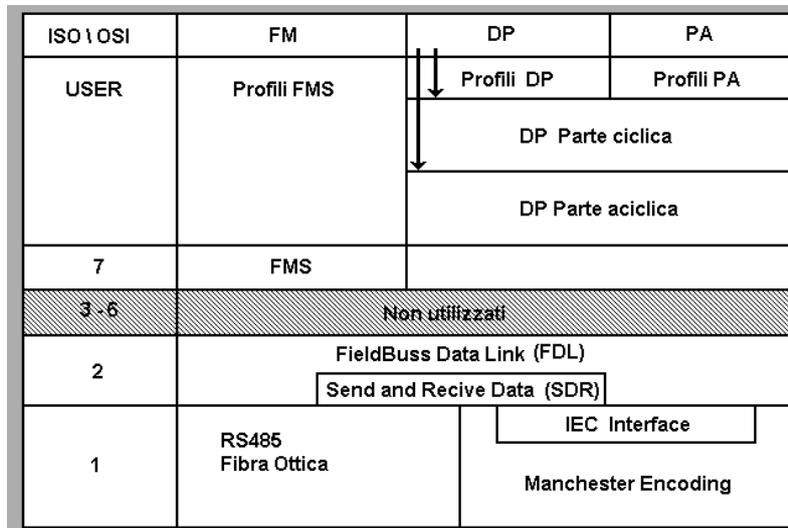


Figura 2.8: I layer del protocollo Profibus

### 2.5.1 Profibus Fieldbus Message Specification

Profibus FMS è la prima versione di Profibus nata nel 1991, oggi in disuso inoltre non è previsto dalle norme IEC, rappresenta la soluzione *general-purpose* per la comunicazione anche a livello di cella<sup>1</sup>. Il livello di applicazione è costituito dal *Fieldbus Message Specification* (FMS) e dal *Lower Layer Interface* (LLI). FMS contiene il protocollo verso l'applicazione e le fornisce una vasta gamma di servizi. LLI comunica sopra con FMS e sotto con lo strato di collegamento livello 2 ed implementa relazioni di comunicazione che permettono a FMS di accedere al livello 2 indipendentemente dal tipo di dispositivo 2.8. Il livello 2, FDL, implementa il controllo per l'accesso al bus e garantisce la sicurezza dei dati. Per la trasmissione fisica, il protocollo FMS permette di utilizzare sia la tecnologia basata su RS 485 con codifica *Manchester* che le fibre ottiche.

### 2.5.2 Profibus Device Peripheral

Profibus DP nato nel 1994 (DIN 19245 parte 3) è ottimizzato per collegamenti economici ad alte prestazioni. Questa versione di Profibus, che è la più utilizzata, è rivolta soprattutto alla comunicazione tra sistemi di controllo e dispositivi distribuiti di ingresso/uscita, sensori/attuatori. Profibus DP utilizza gli strati 1 (PHY) e 2 (FDL) del modello a 7 strati ISO/OSI 2.8. Gli strati dal 3 al 7 non sono

<sup>1</sup>Con il termine cella si indica un insieme di stazioni PLC collegati in rete

definiti per motivi di efficienza e funzionalità di comunicazione tra *layer*, previste dal protocollo è messe a disposizione dell'utente attraverso l'interfaccia utente, vengono mappate sul livello 2 da un applicativo denominato *DDL*M (Direct Data Link Mapper). Questa architettura razionalizzata assicura una trasmissione veloce ed efficiente. Per la trasmissione fisica, vale lo stesso discorso fatto con FMS. Profibus DP e FMS, utilizzando la stessa tecnologia trasmissiva e lo stesso protocollo di accesso al bus (il *token passing*), possono operare simultaneamente sullo stesso cavo, anche se con le prestazioni del protocollo più lento (FMS). La maggior parte dello scambio di dati in questo contesto avviene in modo ciclico, tuttavia per permettere lo svolgimento di procedure di configurazione, di diagnostica o di gestione degli allarmi, il protocollo supporta anche funzioni di comunicazione aciclica. L'aumento significativo di velocità rispetto al protocollo FMS deriva sostanzialmente dell'utilizzo del servizio *SRD* (Send and Receive Data) del secondo livello del protocollo, che consente la trasmissione di dati di ingresso e di uscita in un singolo ciclo di messaggio. Ogni sistema Profibus DP riconosce tre differenti tipi di dispositivi:

- *DP Master Class 1 (DPM1)*: è un controllore centrale che scambia informazioni con le stazioni decentralizzate a lui assegnate (DP slave) all'interno di un ben definito ciclo di messaggio. Gestisce la trasmissione dei dati utente e può comunicare con i dispositivi DP Master (classe 2). Tipicamente appartengono a questa classe PLC slave, PC slave o altri sistemi slave.
- *DP Master Class 2 (DPM2)*: vengono utilizzati nella fase di configurazione del sistema DP o nelle necessarie operazioni di monitoraggio e diagnostiche. Appartengono a questa classe i dispositivi di programmazione PC, di configurazione e pannelli di controllo OP.
- *DP Slave*: sono stazioni periferiche (dispositivi di I/O, trasmettitori, moduli di comunicazione ecc.) che ricevono dati in input (richieste) e/o spediscono in output informazioni al controllore centrale (risposte). Sono indirizzabili da entrambi i tipi di master.

I master sono spesso denominati come stazioni attive, mentre gli slave sono detti stazioni passive. Vi possono essere più DPM1, ciascuno con il suo gruppo di slaves anche se tra loro i DPM1 non comunicano, è possibile invece la comunicazione tra un DPM1 e un DPM2. La velocità elevata di trasmissione non è l'unico elemento che ha determinato il successo della versione DP del protocol-

lo Profibus. Procedure di installazione semplici e trasmissioni senza errori sono caratteristiche altrettanto importanti per gli utilizzatori.

### 2.5.3 Profibus Process Automation

Profibus PA nato nel 1995 (DIN 19245 parte 4) è progettato principalmente per l'automazione di processo, consente di collegare sensori ed attuatori su una linea di comunicazione comune in sicurezza. Con il protocollo PA è possibile trasmettere dati e alimentazione su un bus a due conduttori in accordo con lo standard internazionale IEC 61158-2, che consente di operare in condizioni di sicurezza intrinseca permettendo ai dispositivi di essere alimentati direttamente dal bus. A livello fisico Profibus PA può essere interfacciato con DP per mezzo di un *bridge* chiamato anche accoppiatore. Profibus PA utilizza per la trasmissione dei dati un protocollo DP esteso oltre ad un profilo PA nel quale viene definito il comportamento dei dispositivi di campo. Tutte le versioni di Profibus (DP, FMS e PA) utilizzano un protocollo uniforme di accesso al bus implementato al livello 2 del modello ISO/OSI. Profibus-PA viene utilizzato in campi a rischio di esplosione.

### 2.5.4 Struttura dei frame Profibus

Il protocollo Profibus-DP prevede 2 tipi di servizi di trasmissione che sono.

- SRD invio e richiesta dati con conferma.
- SDN invio di dati senza conferma, utilizzato per la comunicazione *broadcast* o *multicast*.

Durante il servizio SRD il master invia in uscita i dati allo slave e riceve in risposta i dati in ingresso entro un periodo di tempo specificato (ACK + Dati). Un messaggio o telegramma, chiamato anche *frame*, è costituito da più *bytes* l'uno adiacente all'altro preceduti da uno stato di inattività *Idle* durante il quale la linea è a 1 (non sono permessi *idle bit* tra un frame e l'altro), per maggiori informazioni si veda l'appendice B. In questo protocollo esistono 4 tipi di *frame*, distinguibili attraverso il primo byte (SD) 2.9 di seguito riportati:

- 1) Frame con il campo dati di lunghezza variabile fig:2.9.

SYN: periodo di sincronizzazione (33 bit (idle bit) ciascuno di valore 1).

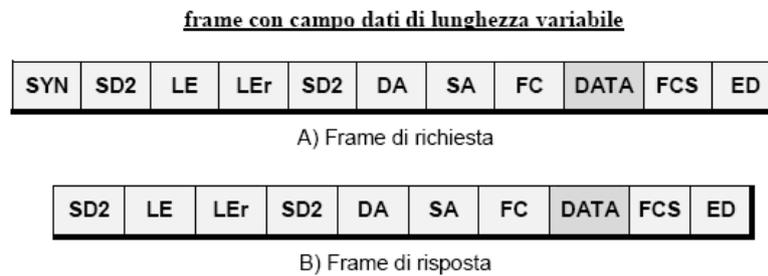


Figura 2.9: Frame con campo dati di lunghezza variabile

SD2: *Start Delimiter* delimitatore iniziale, di valore esadecimale pari a 68H.

LE: numero di byte da DA incluso fino a FCS escluso; valori permessi [4249].

LEr: numero di byte da DA incluso fino a FCS escluso.

DA: *Destination Address* indirizzo del destinatario [0127].

SA: *Source Address* indirizzo del mittente [0126]. Il bit b8 di SA e DA individua la presenza di un'estensione del campo di indirizzamento dei dispositivi.

FC: *Frame Control*. FC realizza un controllo sul tipo di frame, su eventuali perdite o moltiplicazioni di messaggi e sul tipo di stazione. Inoltre contiene informazioni circa il tipo di funzione implementata nel frame (SDN, SDR, ecc.).

DATA: campo dati caratterizzato da un numero massimo di byte pari a 246.

FCS: *Frame Check Sequence*, frame di controllo della sequenza dei bit.

ED: *End Delimiter*, delimitatore finale di valore esadecimale pari a 16H.

2) Frame con campo dati di lunghezza fissa fig:2.10.

SD3: *Start Delimiter* delimitatore iniziale, di valore esadecimale pari a A2H.

DATA-UNIT: campo dati caratterizzato da 8 byte

3) Frame di lunghezza fissa senza campo dati fig:2.11.

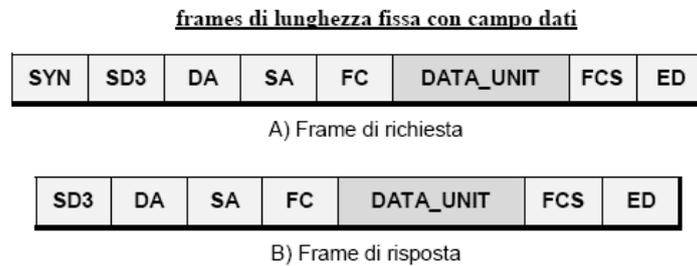


Figura 2.10: Frame con campo dati di lunghezza fissa

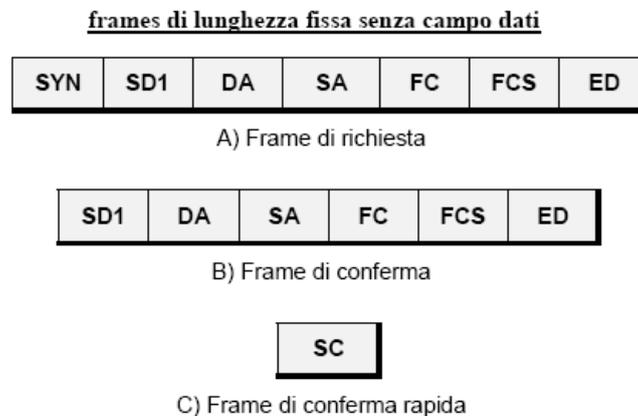


Figura 2.11: Frame di lunghezza fissa senza campo dati

SD1: *Start Delimiter* delimitatore iniziale, di valore esadecimale pari a 10H.

SC: *Single Character* carattere singolo di valore esadecimale pari a E5H.

4) Token frame fig:2.12.

SD4: *Start Delimiter* delimitatore iniziale, di valore esadecimale pari a DCH.

Il protocollo Profibus definisce come vengono scambiati i dati tra le stazioni presenti sul bus. In aggiunta è possibile realizzare dei profili che permettono di stabilire il significato dei dati scambiati sul bus, soprattutto nelle parti definite dall'utente è relative a una specifica applicazione, come il dispositivo profibus-DP debba essere utilizzato in un certo ambito di applicazione. Per assicurare una

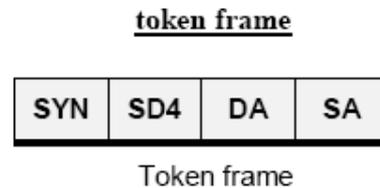


Figura 2.12: Token frame

certa interoperabilità ,Profibus prevede quindi dei profili che descrivono la struttura, il tipo di dati e le assegnazioni dei dispositivi raggruppati per tipologia (esempio sensori/attuatori, azionamenti a velocità variabile, apparecchi di manovra a bassa tensione, automazione degli edifici ect).

## 2.6 Programmazione dei PLC

### 2.6.1 Il software

I sistemi di controllo si basano sul software per controllare il processo produttiva .Questo tipo di attività si svolge per fasi, dopo la fase di raccolta delle informazioni si passa quella di progettazione , segue quella di implementazione. Come si può immaginare tale software deve soddisfare dei requisiti di qualità . Per qualità del software si intende la misura in cui un prodotto software soddisfa un certo numero di aspettative rispetto sia al suo funzionamento sia alla sua struttura interna. Gran parte della ricerca nel campo dell'ingegneria del software è dedicata direttamente o indirettamente , al tema della qualità del software. I fattori rispetto a cui misurare o definire la qualità sono le seguenti:

- **Correttezza:** Un programma o sistema si dice corretto se si comporta esattamente secondo quanto previsto dalla sua specifica dei requisiti ,se il comportamento del software è quello previsto dai progettisti.
- **Affidabilità:** Un sistema è tanto più affidabile quanto più raramente si manifestano malfunzionamenti durante l'utilizzo del sistema , tenendo conto che , nella valutazione dell'affidabilità ,errori gravi si considerano solita-

mente più influenti rispetto a errori non gravi , indipendentemente dalla frequenza con cui si manifestano.

- **Robustezza:** È la misura di come il sistema si comporta in situazioni impreviste non contemplate dalle specifiche.
- **Efficienza:** Un sistema è efficiente se usa le risorse ,CPU,memoria , in modo efficiente ovvero in modo proporzionale al servizio che deve svolgere . Un sistema software non efficiente pregiudica le prestazioni di tutto il sistema di controllo.
- **Manutenibilità:** Facilità di apportare modifiche al sistema realizzato( è maggiore se il sistema è ben progettato) , non corrisponde solo alla correzione degli errori ma comprende anche l'evoluzione del software.

Come in tutti i processi di sviluppo del software ,arrivati alla fine della implementazione si passa alla fase di test e collaudo. Si collauda il software per verificare il comportamento è rilevare errori di programmazione , dopo di che si passa al collaudo del impianto di produzione (chiamata messa in servizio) testando il comportamento di tutto il sistema hardware / software. Anche questo processo di sviluppo produce la propria documentazione.

## 2.6.2 Linguaggi di programmazione

I linguaggi utilizzati per la programmazione dei PLC sono generalmente linguaggi a basso livello . Come già accennato prima alcuni produttori hanno scelto di dividere il codice e i dati in blocchi così da avere una separazione tra blocchi funzione FB e blocchi dati DB. Questo tipo di suddivisione del programma è molto efficace in quanto il programma è più comprensibile, una qualsiasi modifica viene fatta a livello di blocco sia esso di dati che di funzione. Dopo la stesura del codice il programma deve essere compilato prima di caricarlo nel PLC . Quindi in generale i linguaggi per la programmazione di PLC sono linguaggi compilati. Le norme IEC 1131-3 hanno avuto il merito di cercare di unificare i linguaggi di programmazione più diffusi nel mondo dei PLC, cercando di evitare il proliferare dei vari dialetti proposti dai diversi costruttori. Questi linguaggi si dividono in due gruppi:

### a) Linguaggi Testuali

- ST: Structured text

- IL: Instruction list

#### b) Linguaggi Grafici

- SFC: Sequential function chart
- FBD: Function block diagram
- LD: Ladder diagram

Di seguito viene data una breve descrizione di questi linguaggi.

#### **Il linguaggio *Structured Text***

Il linguaggio testuale *Structured Text* (ST) rappresenta il passaggio alla programmazione strutturata è di alto livello per i controllori industriali. Mantenendo la semplicità e la chiarezza sintattica necessarie per essere utilizzato con successo nello sviluppo di applicazioni anche complesse, può essere utilizzato su dispositivi dalle prestazioni computazionali limitate. Il linguaggio ST si presta per la risoluzione in modo rapido ed intuitivo di qualsiasi tipo di compito di programmazione. Tuttavia, dato il campo di applicazione ed il contesto nel quale è definito, considerando cioè anche gli altri linguaggi grafici o testuali della Norma IEC 61131-3, risulta particolarmente adatto per eseguire complesse elaborazioni matematiche, realizzabili con poche righe di codice, oppure nel caso in cui occorra eseguire test condizionali con molteplici alternative, i quali richiederebbero con la logica a contatti o con il linguaggio IL un grande numero di operazioni di salto con la conseguente riduzione di leggibilità del software sviluppato. Strutturalmente assomiglia al linguaggio Pascal. L'immagine in fig:2.13 riporta un esempio di questo linguaggio.

```
VAR
    Sel : INT;
END_VAR

(* Selezione con IF *)
IF Sel = 0 THEN
    istruzioni0;
ELSIF Sel = 1 THEN
    istruzioni1;
ELSIF Sel = 2 THEN
    istruzioni2;
ELSE istruzionielse;
END_IF;

(*Selezione con CASE *)
CASE Sel OF
    0: istruzioni0;
    1: istruzioni1;
```

Figura 2.13: Esempio di codice scritto in ST

### Il linguaggio *Instruction List*

La tipica applicazione dei PLC industriali consiste nello svolgere compiti di controllo logico con operazione prevalentemente booleane e l'architettura hardware delle loro CPU é ottimizzata per svolgere questo tipo di operazioni in modo semplice e rapido. I linguaggi di livello assemblativo di tali dispositivi sono pertanto basati storicamente su una sintassi del tipo 1 operatore : 1 operando. Con questa sintassi, occorre un particolare tipo di registro interno del processore, detto accumulatore, nel quale memorizzare il risultato dell'operazione, mentre leventuale secondo operando necessario all'operazione (es. somma logica tra due valori booleani) consiste implicitamente nel contenuto dell'accumulatore stesso. Anche lo standard IEC 61131-3, essendo derivato dall'analisi dettagliata del panorama corrente del mercato PLC, segue questo tipo di sintassi per la definizione del linguaggio *Instruction List* (IL), il linguaggio testuale di livello inferiore nello standard. Essendo un linguaggio assemblativo, tutti gli altri linguaggi possono avere un'equivalente in IL, mentre non é sempre possibile convertire codice IL in altri linguaggi. Infatti, l'IL si presta per sua natura alla generazione di codice ottimiz-

zato, il quale tuttavia potrebbe perdere l'equivalenza con codice di alto livello o diagrammi grafici in LD o FBD (trattati in seguito). L'immagine in fig:il riporta un esempio di questo tipo di linguaggio.

```

U      "M2.0"          M2.0          -- marcia comandi # marche commandes
U{
UN     "M25.0"         M25.0         -- Puls. JP3 Degasatore 0=remoto 1=locale # Boite a b. JP3 malaxeur 0=DIST 1=LOCAL
U{
U      "M2.1"          M2.1          -- manuale # manuel
UN     "YV02"          A34.7          -- FRIZIONE DEGASATORE # EMBRAYAGE MALAXEUR
U      "SB30"          E21.2          -- MARCIA DEGASATORE # MARCHE MALAXEUR
U      "M4.2"          M4.2          -- avviamenti esauriti # démarrages épuisés
O      "KM13S_K"      A34.6          -- START AVV. DIRETTO DEGASATORE # START DEMARRAGE DIRECT MALAXEUR
}
U      "SB31"          E21.3          -- ARRESTO DEGASATORE # ARRÊT MALAXEUR
O
U      "M25.0"         M25.0         -- Puls. JP3 Degasatore 0=remoto 1=locale # Boite a b. JP3 malaxeur 0=DIST 1=LOCAL
U{
U      "M2.1"          M2.1          -- manuale # manuel
UN     "YV02"          A34.7          -- FRIZIONE DEGASATORE # EMBRAYAGE MALAXEUR
U      "SB36"          E22.2          -- MARCIA DEGASATORE (JP3) # MARCHE MALAXEUR (JP3)
U      "M4.2"          M4.2          -- avviamenti esauriti # démarrages épuisés
O      "KM13S_K"      A34.6          -- START AVV. DIRETTO DEGASATORE # START DEMARRAGE DIRECT MALAXEUR
}
U      "SB37"          E22.3          -- ARRESTO DEGASATORE (JP3) # ARRÊT MALAXEUR (JP3)
}
U
U      "M10.0"         M10.0         -- l=marcia diretta motore degasatore # l = marche direct moteur malaxeur
UN     "OP7".ALL17    DB100.DBX75.0 -- degasatore: sezionatore
UN     "OP7".ALL19    DB100.DBX75.2 -- degasatore: sovratemperatura motore
UN     "OP7".ALL20    DB100.DBX75.3 -- degasatore: selezione tipo di avviamento non possibile
UN     "OP7".ALL21    DB100.DBX75.4 -- degasatore: termico motore principale
UN     "G13_STA"     A34.3          -- START INVERTER DEGASATORE # START VARIATEUR MALAXEUR
=      "KM13S_K"     A34.6          -- START AVV. DIRETTO DEGASATORE # START DEMARRAGE DIRECT MALAXEUR

```

Figura 2.14: Esempio di codice scritto in IL

### Il linguaggio *Sequential Functional Chart*

Negli anni 70 venne sviluppato un linguaggio grafico ,inizialmente di specifica , poi anche di programmazione per il controllo logico.All'inizio venne chiamato Grafcet ,é stato incluso nella norma IEC 1131-3 con il nome di *Sequential Functional Chart* (SFC).Gli elementi fondamentali di SFC sono i seguenti:

- fasi o tappe.
- transizioni.
- archi orientati che connettono fasi a transizioni o viceversa.
- regole di evoluzione, che definiscono senza ambiguitá il comportamento del programma.
- azioni associate alle fasi.
- condizioni logiche associate alle transizioni.

Lo stato di attivazione delle fasi rappresenta lo stato del sistema. Esso viene modificato dall'occorrenza di eventi, che mediante le transizioni portano il sistema in una nuova condizione (con altre fasi attive). In generale il funzionamento é simile ad una macchina a stati finiti in cui il verificarsi di un evento é seguito da una transizione di stato.L'immagine in fig:2.15 e riportato un esempio di questo linguaggio.

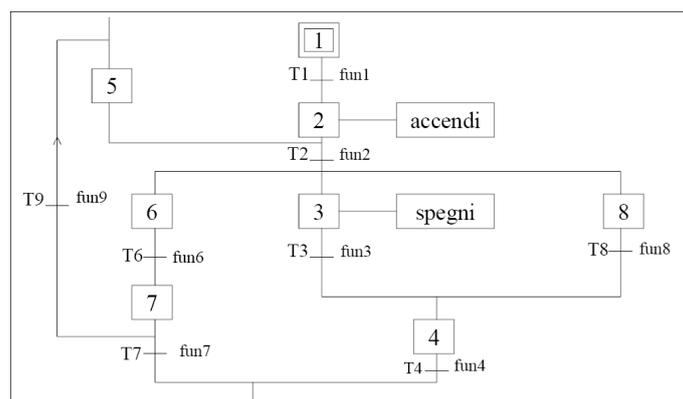


Figura 2.15: Esempio di codice scritto in SFC



### Il linguaggio *Ladder Diagram*

Considerazioni analoghe a quelle fatte sulla definizione dell'linguaggio Instruction List possono essere fatte sul linguaggio grafico *Ladder Diagram* (LD). Infatti, anche questo linguaggio si basa sull'osservazione delle caratteristiche comuni tra le varie implementazioni della logica di programmazione maggiormente diffusa tra gli utenti di PLC: la logica a contatti o *relay ladder logic*. Tali caratteristiche comuni sono pertanto unificate nella definizione del LD, in modo da permettere al programmatore di avere un approccio piú immediato ai diversi editor grafici di differenti produttori. Il concetto alla base del LD é quello di rappresentare graficamente un flusso virtuale di corrente elettrica tra due barre di potenziale, regolato da interruttori e bobine, in modo da implementare in modo molto intuitivo una logica booleana: passaggio di corrente = TRUE, assenza di corrente =FALSE. Una riga di codice corrisponde perciò ad una rete logica di contatti, Functions, Function Blocks e bobine, connesse da linee (eventualmente estese da etichette testuali dette connettori, in caso di suddivisione su piú righe o pagine) attraversate da un flusso di corrente. La norma stabilisce che il flusso di corrente si intende diretto da sinistra a destra e che le reti vanno valutate (eseguite) dall'alto verso il basso, a meno che non siano presenti istruzioni di salto o di termine forzato dell'esecuzione. L'immagine in fig:2.17 e riportato un esempio di questo linguaggio.

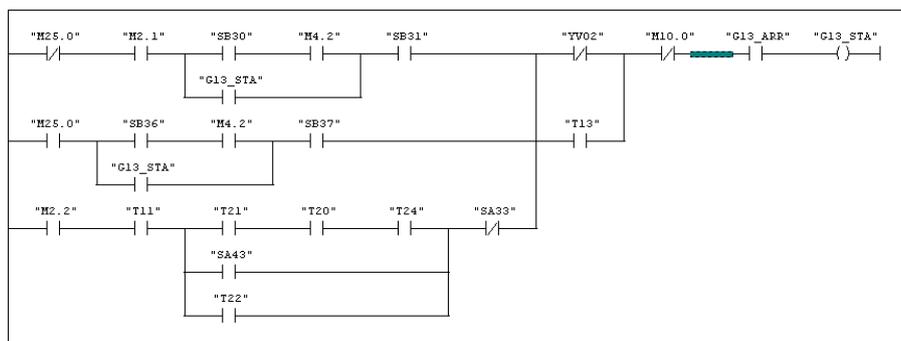


Figura 2.17: Esempio di codice scritto in LD

Con quest'ultimo argomento si conclude la trattazione riguardante i sistemi di controllo. Per informazioni piú esaurienti si possono consultare le documentazione rilasciate dai vari produttori di sistemi di controllo.



# Capitolo 3

## Sistemi di supervisione

Un sistema di supervisione viene generalmente impiegato per controllare una realtà complessa, dove l'acquisizione di una grande quantità di dati in tempo reale, la sua elaborazione, interpretazione e presentazione sono essenziali per il funzionamento di impianto, dal momento che le decisioni riguardanti la conduzione vengono prese proprio in base a tali dati, la loro raccolta non può essere effettuata manualmente. Sostanzialmente i sistemi di supervisione si dividono in due categorie che sono le seguenti:

- a) Sistemi HMI (Human Machine Interface)
- b) Sistemi SCADA (Supervisory Control And Data Acquisition)

I sistemi HMI sono generalmente impiegati a bordo macchina, cioè nelle immediate vicinanze della macchina che si vuole controllare. I sistemi SCADA sono anche essi fondamentalmente degli HMI e per molti aspetti più evoluti rispetto agli HMI standard. La differenza più grande rispetto agli HMI consiste nel fatto che i sistemi SCADA possono memorizzare i valori di processo rilevati anche per lunghi periodi di tempo. Questi sistemi hanno una sorta di database per la memorizzazione di tali dati. Tutti i sistemi SCADA vengono integrati o appoggiati su sistemi di database. Solitamente i database sono software commerciali sviluppati da terzi, come esempio Microsoft SQL, Oracle, DB2... ect. Oltre ai DB commerciali vengono impiegati anche DB *open source* come PostgreSQL, MySQL... ect. Un'altra differenza tra SCADA e gli HMI standard risiede nel fatto che i sistemi SCADA non vengono impiegati a bordo macchina. Questi sistemi effettuano un controllo da postazioni remote via rete, per questo motivo vengono anche chiamati sistemi di supervisione e raccolta dati. In seguito verranno illustrati i componenti che cos-

tituiscono un sistema di supervisione come avviene la comunicazione tra essi , il processo di sviluppo è quali requisiti devono essere soddisfatti.

### 3.1 Tipologie di HMI

In un sistema di supervisione è importante il modo in cui avviene l'interazione uomo-macchina. I dispositivi HMI stanno assumendo un ruolo sempre più decisivo nelle applicazioni di automazione sia industriale che civile, infatti la possibilità di visualizzare in tempo reale messaggi diagnostici , allarmi o istruzioni per l'operatore è al contempo modificare i parametri operativi i modo semplice e diretto è diventata un'esigenza essenziale nella maggiorparte delle applicazioni . Nel caso dell'attività di controllo di un impianto di produzione l'utente dovrà essere in grado di visualizzare lo stato degli elementi che compongono questo impianto ad esempio sonde di temperatura, di umidità ,posizionamenti di parti mobili del impianto ecc ,inoltre l'utente dovrà essere in grado di impartire comandi al impianto ed effettuare regolazioni. L'interazione uomo-macchina avviene attraverso la riproduzione di tutti i pannelli di controllo dei regolatori di temperatura , allarmi, ed altri attuatori . È il supervisore a farsi carico del controllo del regolare funzionamento degli apparati , della segnalazione di anomalie e dell'esecuzione o suggerimento di azioni correttive. Si avverte sempre più l'esigenza di visualizzare, controllare e interagire con i sistemi da postazioni remote.La tendenza del mercato HMI riflette le linee di evoluzione introdotte dal fenomeno della convergenza digitale : le applicazioni e i dispositivi si sono fatti sempre più flessibili e potenti,consentendo remotizzazione ,multicanalità, mobilità,personalizzazione e adattamento a diverse tipologie di rete, mentre le interfacce utente e le architetture applicative si sono uniformate ad alcuni tra i principali standard tecnologici emersi sul mercato, architetture basate su indirizzamento IP,ISO/OSI,architetture a 3 livelli ecc. Sostanzialmente oggi sul mercato esistono diverse tipologie di HMI, che vanno da semplici display con linee di testo nei quali si possono visualizzare informazioni testuali, a display grafici nei quali assieme al testo sono presenti anche informazione grafiche. Alcuni di questi sistemi integrano le funzionalità dei PC desktop e vengono chiamati *panel PC* ,sostanzialmente è un sistema di calcolo integrato in un display grafico dotato di sistema operativo ,generalmente Microsoft Windows ,Windows CE .Questi sistemi possiedono tutte le funzionalità sia grafiche che di comunicazione dei PC.Tuttavia una limitazione di questi dispositivi è data dalla dimensione dello schermo chè nei modelli più economici è ridotta di conseguenza si ha una limi-

tazione nelle applicazioni grafiche. Alcuni di questi dispositivi sono visualizzati nelle immagini in fig:3.1.



Figura 3.1: Pannello operatore testuale



Figura 3.2: Pannello operatore grafico con touch screen



Figura 3.3: Panel PC con sistema operativo Microsoft

## 3.2 Progettazione dell'interfaccia utente i vincoli

Con l'avvento dei servizi web come punti centrali per la condivisione di informazioni e di dati, gli utenti hanno sentito la necessità di accedere e gestire le informazioni di lavoro, personali, da qualsiasi luogo in qualsiasi momento e con qualsiasi dispositivo (computer, palmare, cellulare, televisione ecc), di conseguenza la progettazione di applicazioni per dispositivi si è considerevolmente evoluta. Quando si progetta l'interfaccia di una applicazione è necessario considerare molti elementi relativi al dispositivo di destinazione, come la risoluzione, la dimensione e gli angoli di visualizzazione dello schermo, i colori, i caratteri, in quanto le applicazioni vengono visualizzate diversamente a seconda del dispositivo. L'obiettivo consiste nello sviluppare applicazioni più rapide, produttive semplici ed esteticamente gradevoli e di facile comprensione ed utilizzo da parte del utente finale. I principali vincoli di progettazione di una interfaccia utente che bisogna osservare sono i seguenti:

**Dimensioni:** La maggior parte dei dispositivi ha dimensioni molto ridotte rispetto a quelle PC desktop, questo comporta delle limitazioni per alcune applicazioni.

**La visualizzazione:** Mentre la risoluzione dei monitor tradizionali dei PC desktop è di almeno 800x600 pixel, per i dispositivi HMI l'interfaccia grafica (ad esempio per un pannello operatore con touch screen di 12) dispone di una risoluzione inferiore. Anche la scala di grigio e il numero di colori sono limitati per alcuni di questi dispositivi. Questi sono fattori da tenere presente quando si progetta un interfaccia grafica.

**I colori:** Utilizzare dei colori espressivi che esprimono in modo chiaro lo stato di funzionamento di un componente del impianto. Convenzionalmente il colore rosso è associato ad uno stato di pericolo di conseguenza viene utilizzato per segnalare lo stato allarme, malfunzionamento, del componente. Il colore verde viene utilizzato per rappresentare un funzionamento corretto del impianto. Il colore arancio viene utilizzato per segnalare ritardi ed altri tipi di anomalie lievi. Tutti gli altri colori sono a discrezione del progettista, tenendo sempre presente il dispositivo di destinazione.

**Le segnalazioni:** Quando si verifica un evento di malfunzionamento o allarme per cui è richiesto l'intervento del operatore. Il messaggio di notifica del evento occorso deve essere un messaggio chiaro in modo che possa guidare l'operatore alla fonte del problema. Quindi si evitano messaggi troppo corti o tecnici in quanto non sono di aiuto al operatore per risolvere i problemi.

**Metodi di input e di output:** I dispositivi consentono di utilizzare diversi metodi di interazione, quali pulsanti hardware, pulsanti software (nel caso dei dispositivi touch screen), stilo, mouse e tastiera. Bisogna quindi tenere presente le dimensioni dello schermo e di conseguenza dimensionare correttamente le componenti grafiche, quali i pulsanti, menu, *check box* ect.

Per migliorare l'accessibilità dell'applicazione è consigliabile tenere in considerazione tutti questi vincoli durante la progettazione dell'interfaccia. Un altro aspetto da valutare è l'analisi del tipo di attività che può essere svolta dal utente, dalla rapidità con cui tali attività vengono completate, ed alle prestazioni di dispositivi e applicazioni. È pertanto necessario progettare l'interfaccia dell'applicazione in modo da semplificare il completamento delle attività eseguite dall'utente, anziché pensare all'applicazione in termini estetici. A questo scopo, è possibile riprodurre un'interfaccia simile o correlata alle attività compiute dagli utenti con il dispositivo, rendere l'applicazione il più possibile gradevole esteticamente sapendo quali elementi è possibile visualizzare sullo schermo e quali no, tenendo presente quanto detto sopra.

### 3.3 Requisiti progettuali dell'interfaccia utente

I requisiti principali che deve possedere l'interfaccia utente sono utilizzabilità e accessibilità. Con il termine utilizzabilità si intende il grado in cui un prodotto può essere usato da particolari utenti per raggiungere certi obiettivi con efficacia, efficienza e soddisfazione in uno specifico contesto d'uso. Il problema dell'usabilità è emerso dapprima negli anni 80 con la diffusione delle tecnologie informatiche a livello di ufficio e di famiglia, ed è definitivamente esploso negli anni 90 con la diffusione del personal computer. Mentre prima i principali utilizzatori del prodotto software finivano per essere gli stessi progettisti o persone esperte con una formazione simile ai progettisti, ora gli utenti finali del software non sono necessariamente esperti di informatica. L'utilizzabilità nasce dunque soprattutto come ausilio alla progettazione: l'obiettivo è fare in modo che il modello mentale di chi ha progettato il software (design model), da cui deriva il suo reale funzionamento, corrisponda il più possibile al modello mentale del funzionamento del software così come se lo costruisce l'utente finale (user model). Le tecniche di utilizzabilità tentano dunque di porre al centro dell'attenzione progettuale proprio l'utente: ad ogni sua azione l'interfaccia proporrà un risultato, un cambiamento di stato, ma ai fini dell'utilizzabilità non importa come l'interfaccia sia giunta

a quello stato cioè quali meccanismi di programmazione siano stati usati. L'accessibilità consiste nella possibilità di rendere fruibili i contenuti delle pagine e dei dati in essa contenuti agli utenti finali. Il compito del progettista è quello di produrre del software, che consenta di rendere la struttura della pagina il più possibile flessibile, in modo che i suoi contenuti possano essere utilizzati senza perdita d'informazioni.

### 3.4 Model,View,Controller

Mediante il paradigma MVC *Model, View, Controller* vengono identificati i tre componenti fondamentali di un'applicazione interattiva. L'intento della tecnica *Model, View, Controller* applicato agli HMI è di disaccoppiare il più possibile tra loro le parti dell'applicazione adibite al controllo, all'accesso ai dati, è alla loro presentazione. Questo approccio porta a diversi vantaggi che sono:

- Indipendenza strutturale tra *model*, logica di presentazione *view* e la logica di controllo *controller*.
- Viste diverse per lo stesso *model*.
- Separazione dei compiti dei ruoli è relative interfacce.

L'immagine in fig:3.4 mostra schematicamente questo tipo di interazione tra i vari componenti.

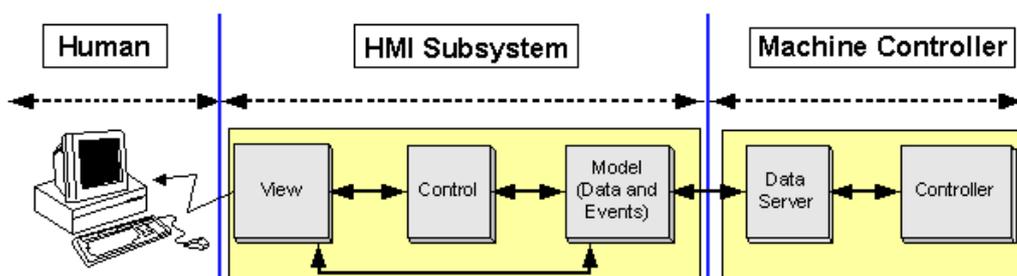


Figura 3.4: Design Patern MVC

**MODEL:** Il *core* dell'applicazione viene implementato dal Model, che incapsulando lo stato dell'applicazione definisce i dati e le operazioni che possono essere

eseguite su questi. Quindi definisce le regole per l'interazione con i dati, esponendo alla view ed al controller rispettivamente le funzionalità per l'accesso e l'aggiornamento. Il model può inoltre avere la responsabilità di notificare ai componenti della view eventuali aggiornamenti verificatisi in seguito a richieste del controller, al fine di permettere alle view di presentare agli occhi degli utenti dati sempre aggiornati.

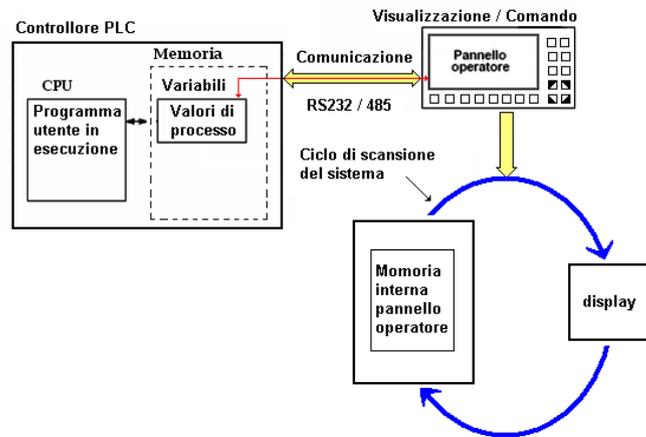
**VIEW:** La logica di presentazione dei dati viene gestita solo e solamente dalla view, questa deve fondamentalmente gestire la costruzione dell'interfaccia grafica *GUI* che rappresenta il mezzo mediante il quale gli utenti interagiranno con il sistema. Ogni *GUI* può essere costituita da schermate diverse che presentano più modi di interagire con i dati dell'applicazione. Inoltre la view delega al controller l'esecuzione dei processi richiesti dall'utente dopo averne catturato gli input e la scelta delle eventuali schermate da presentare.

**CONTROLLER:** Questo componente ha la responsabilità di trasformare le interazioni dell'utente della view in azioni eseguite dal model. Ma il controller non rappresenta un semplice ponte tra view e model, infatti realizza la mappatura tra input dell'utente e processi eseguiti dal model, è seleziona la schermate della view richieste, il controller inoltre implementa la logica di controllo dell'applicazione.

### 3.5 Comunicazione tra PLC e HMI

Come illustrato precedentemente la comunicazione tra HMI-SCADA e PLC avviene principalmente mediante connessione seriale o ethernet. I protocolli di livello superiore possono essere differenti a seconda del tipo di applicazione e non è garantita l'interoperabilità tra i differenti standard. Le più diffuse implementazioni di questi protocolli sono MODBUS/RS 232, MODBUS TCP-IP, PROFIBUS RS232-RS485 (vedi capitolo 2), PROFINET TCP-IP, INDUSTRIAL ETHERNET, DEVICENET, MPI, ed infine CONTROLNET, protocolli che non sono trattati in quest tesi.

Come si può vedere dall'immagine fig:3.5 la connessione tra HMI e PLC generalmente una connessione seriale punto-punto, anche se è possibile fare comunicare un OP (Pannello Operatore) con più PLC. Inoltre si può stabilire che la logica funzionale sia situata all'interno del PLC, cioè che questo sia l'unico strumento di elaborazione delle informazioni. Quindi il compito del OP è quello di visualizzare e se necessario modificare le informazioni contenute al interno del PLC. Infatti nella memoria del OP vengono memorizzate variabili che fanno riferimento



a indirizzi di memoria contenuti nel PLC. Come descritto in capitolo 2 i dispositivi HMI-SCADA fanno parte dei dispositivi master. In una connessione OP-PLC il ruolo del master è riservato al OP ,in quanto è proprio quest'ultimo a prendere l'iniziativa alla comunicazione. Il sistema operativo del OP scansiona la memoria interna in modo ciclico dopodichè genera le richieste che successivamente verranno inviate al PLC il quale a sua volta invierà all'OP i dati richiesti. Questo procedimento serve ad avere una visualizzazione dei dati sempre aggiornata. Oltre a visualizzare i valori di processo, lo HMI deve essere progettato in modo da offrire gli strumenti affinché l'utente possa intervenire sui valori di processo: per esempio modificandoli secondo esigenze di produzione.

### 3.6 Caratteristiche del software SCADA

Come già accennato i software di supervisione sono denominati SCADA, questi programmi rappresentano in forma grafica lo stato di un segnale proveniente da una impianto. La forma può essere planimetrica nel senso che è legata al disegno della macchina, oppure in forma tabellare tipo allarme oppure ancora in forma di grafico temporale. Lavorare con tali software non costituisce una vera e propria programmazione anche se mettono a disposizione strumenti di sviluppo ad esempio compilatori C, Visual Basic, Java ect, ma si tratta di una modellazione del software fino a fargli assumere la forma voluta. Clienti diversi hanno esigenze diverse sia in fatto di visualizzazione che di progetto , i software SCADA mettono a disposizione del progettista un insieme di strumenti per lo sviluppo del progetto. Tali strumenti a volte possono risultare insufficienti per il completamen-

to del progetto ,ed è compito del progettista realizzare le funzionalita mancanti mediate l'utilizzo di linguaggi di programmazione. Il sistema di supervisione SCADA è il complesso delle apparecchiature che interfacciano i sistemi di regolazione e controllo con gli operatori. Al sistema di supervisione è demandato il compito di rappresentare tramite pagine grafiche il funzionamento dell'impianto, nonchè di accettare comandi dall'operatore per la conduzione dell'impianto stesso. Da un punto di vista strutturale il software SCADA si può descrivere secondo lo schema raffigurato nella immagine fig:3.5,come si può vedere questo software ha una strutture intera complessa. Inoltre questi software richiedono molte risorse hardware per funzionare è di conseguenza per far'girare l'applicazione viene utilizzato un PC desktop a loro dedicato, opportunamente configurato.

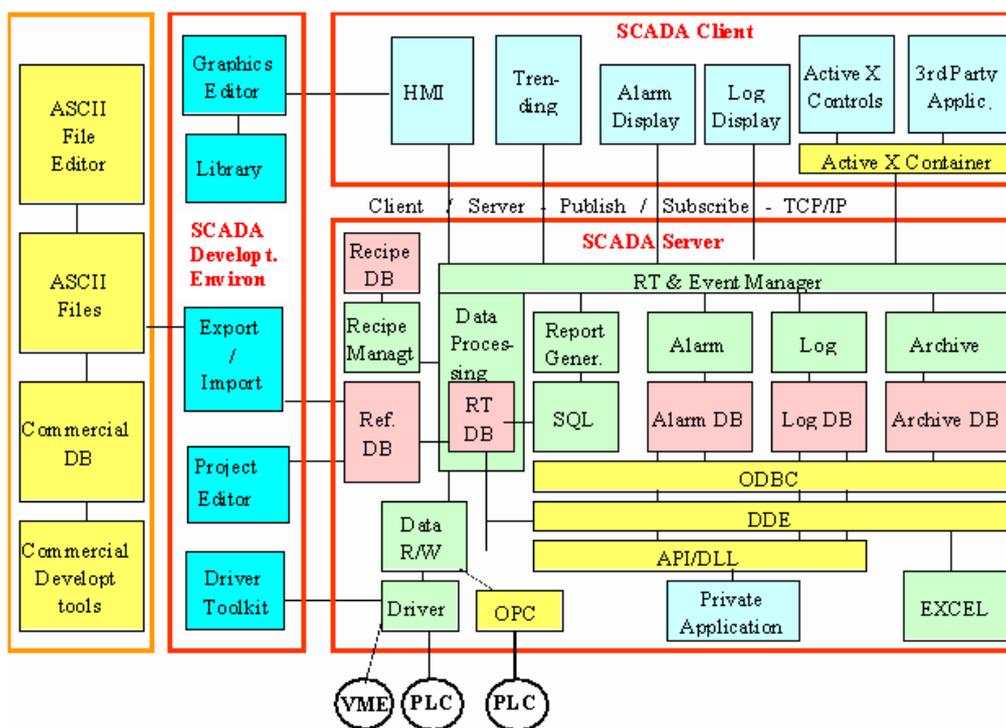


Figura 3.5: Struttura interna di un software SCADA

Un sistema di supervisione e raccolta dati si intende costituito da elaboratori standard PC equipaggiati con software di tipo SCADA è connesso con i controllori (PLC) tramite una rete. Lo SCADA ha una architettura di tipo *client server*, nello stesso applicativo gira sia la parte client che quella server .Al momento del-

la configurazione della rete vengono decisi i ruoli delle varie stazioni. Le stazioni client curano solamente l'interfaccia grafica uomo/macchina (HMI), non sono connesse ad alcun controllore PLC come mostrato in fig:3.6, sono solamente delle repliche della stazione server.

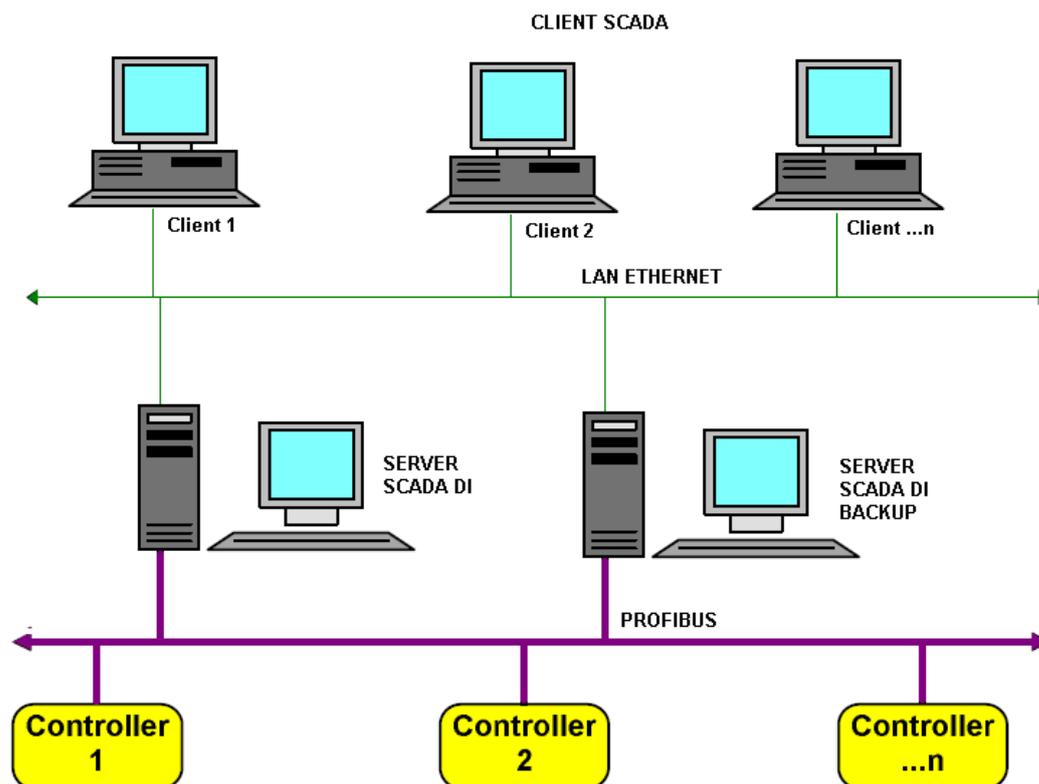


Figura 3.6: Struttura interna di un software SCADA

Le stazioni server assolvono alla funzione di scambio dati tra i sistemi di regolazione PLC e il database interno nel quale sono archiviati in tempo reale tutti i parametri dell'impianto. Generalmente in un sistema di supervisione non ci sono limiti al numero di stazioni server o client che possono essere installate, se non i limiti fisici di memoria dei PC server o di velocità della rete.

I dati richiesti dalle stazioni client sono solo quelli in corso di visualizzazione, in ogni caso qualunque client avrà a disposizione i dati di tutti i server connessi alla rete. Nei client non vi sono duplicazioni del database di un sistema di supervisione, le uniche stazioni a cui è consentito un duplicato del database sono le

stazioni dette *server di backup* fig:3.6, che intervengono automaticamente in sostituzione delle stazioni principali in caso di guasto. Inoltre essendo lo scada un software commerciale, viene commercializzato in due versioni che sono: versione *runtime*, che si occupa di eseguire un applicativo già sviluppata, e la versione di sviluppo che mette a disposizione del progettista tutti gli strumenti previsti. In base alla licenza d'uso acquistata si determina il tipo di utilizzo del software. Nel prossimo capitolo verrà mostrato come esempio un progetto che utilizza un sistema SCADA ed un PLC di fabbricazione SIEMENS.



# Capitolo 4

## Sistema di supervisione del forno

In questo capitolo viene presentato un progetto costituito da un sistema SCADA-PLC impiegato per il controllo e la raccolta dati di un forno. I software di sviluppo utilizzati sono WinCC, ProTool Pro, Simatic Step7. Tali software fanno parte dei strumenti di sviluppo messi a disposizione da Siemens per la programmazione dei propri prodotti quali, PLC, OP ... ect. WinCC è lo SCADA di Siemens, è basato su un database SQL [7] di produzione Microsoft, mette a disposizione del progettista un compilatore VB e C, per la creazione di procedure personalizzate. ProTool Pro è un software utilizzato per la modellazione di interfacce grafiche per pannelli operatore HMI, mette a disposizione un editor VB per creare componenti personalizzati. Infine lo Step7, viene utilizzato per la programmazione di PLC e per la configurazione di rete grazie a NetPro che è incluso nel pacchetto Step7. L'editor di Step7 permette di programmare in tre differenti linguaggi, quali il ladder diagram, il function block diagram ed infine l'istruzione list (vedi capitolo 2). I dispositivi di programmazione utilizzati per la programmazione dei PLC vengono chiamati *Field PG*, sono forniti sempre dalla Siemens. Sono *laptop* equipaggiati di tutte le interfacce necessarie per la comunicazione con i PLC, OP ect. Questi sono i strumenti software e hardware principalmente utilizzati dal progettista per questo lavoro (si veda l'appendice E). Prima di descrivere il progetto si fornisce una breve descrizione di come viene realizzata il progetto, i passi fondamentali che costituiscono il processo di sviluppo del software.

### 4.1 Fasi del processo di sviluppo

Come ogni altra attività di sviluppo software [10], anche questa attività si divide in fasi, che sono: raccolta delle informazioni iniziali, pianificazione del progetto

da realizzare, analisi dei requisiti, implementazione, test e collaudo, ed in fine la messa in servizio. Il processo di sviluppo software è schematizzato in fig:4.1.

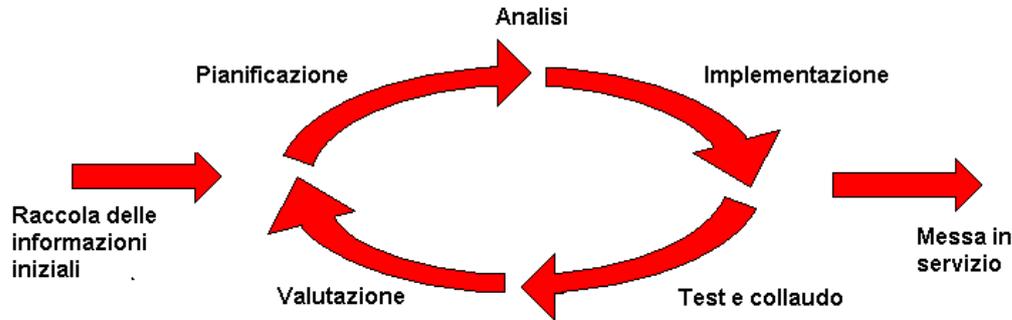


Figura 4.1: Schema di un processo di sviluppo software

La raccolta delle informazione consiste nello stabilire, requisiti del sistema, vincoli, esigenze del cliente, schemi elettrici, componenti hardware, layout del impianto da realizzare, scadenze e tempi di consegna...ect. Per quanto riguarda l'attività di pianificazione ed analisi, si tratta in prevalenza di decidere quali soluzioni adottare per lo sviluppo del progetto, tenendo presente quali siano le esigenze derivanti dalla prima fase. Dopo la conclusione della seconda fase si passa a quella di implementazione, questa fase deve essere necessariamente preceduta dalle prime due, se così non fosse il risultato ottenuto al termine di questa fase potrebbe non essere corretto, potrebbe non rispettare alcuni vincolo o esigenze, di conseguenza ci sarebbe un prolungamento dei tempi di sviluppo. Nella fase di test e collaudo il software viene testato per verificare la presenza di errori o anomalie di funzionamento ed successivamente la loro eliminazione. La fase di valutazione consiste nel valutare se il software creato si comporta secondo le specifiche ,è sodisfa tutte le richieste derivanti dalla prima fase.L'ultima fase è quella di messa in servizio, si tratta in pratica di mettere in funzione l'impianto e testare il comportamento. Durante questa fase si possono presentare dei problemi di funzionamento, modifiche fatte al progetto in fase successive alla progettazione o anche che il cliente richieda delle funzionalità supplementari non previste nella fase uno. Con quest'ultima fase si conclude anche il processo di sviluppo, ognuna di queste fasi produce la propria documentazione tecnica. Nella parte seguente verrà descritto il progetto di un sistema di controllo e raccolta dati di un forno utilizzato per la produzione di laterizi. Come mostrato

in fig:2.6 il sistema di controllo di un impianto per produzione di laterizi è formato da più PLC connessi in rete, ovviamente il numero dei controllori varia in base all'estensione dell'impianto. In generale ognuno dei PLC deve possedere la sua interfaccia OP come si vede anche dall'immagine. Inoltre sempre in questa immagine ce un sistemi di supervisione e acquisizione dati SCADA connesso direttamente con i PLC del forno è. Il progetto che verra descritto nella seguente parte tratto il sistema PLC-SCADA utilizzati per la conduzione del forno tralasciando il resto della rete in quanto i concetti di questo sistema sono applicabili a tutti gli altri elementi della rete.

## 4.2 La programmazione del PLC

Prima di passare alla costruzione del software del PLC si deve configurare l'hardware (PLC) ed successivamente la rete. Sono i primi due passi che bisogna obbligatoriamente fare prima di iniziare la programmazione. Con la configurazione del hardware si intende, selezionare il PLC corretto è tutti i suoi moduli di I/O più altri moduli di comunicazione con I/O remoti. Con la configurazione della rete si intende configurare il PLC in modo che sia in grado di comunicare con vari partner della rete. Questa fase consiste nello scegliere il tipo di protocollo di comunicazione, le assegnazione degli indirizzi di rete ed infine la velocità di comunicazione *bitrate*. Alla fine della configurazione quando tutti gli elementi sono stati individuati il file deve essere salvato e compilato. In generale senza questa configurazione preliminare in Step 7 non è possibile iniziare una qualsiasi attività di programmazione. Al termine di questa fase si passa alla fase di programmazione. Il linguaggio utilizzato per la programmazione del PLC il ladder diagram LD (vedi capitolo2) ,senza dubbio è il linguaggio di programmazione più utilizzato. Come già descritto nel capitolo 2, in un PLC vengono eseguiti due programmi distinti che sono:

- a) Il sistema operativo.
- b) Il programma utente.

Il sistema operativo contenuto in ogni PLC di produzione Siemens organizza tutte le procedure della CPU che non sono legate a un compito di controllo specifico. I compiti del sistema operativo comprendono:

- La gestione del nuovo avviamento chiamato avviamento a caldo, inoltre gestione del riavviamento.

- L'aggiornamento dell'immagine di processo (degli ingressi) ed emissione dell'immagine di processo (delle uscite).
- Il richiamo del programma utente.
- rilevamento di allarmi e richiamo degli OB(descritte di seguito) di allarme
- riconoscimento e gestione degli errori
- gestione delle aree di memoria

Per quanto riguarda il programma utente, questo deve essere creato e caricato nel PLC. Il programma contiene tutte le funzioni necessarie per l'elaborazione dei vari compiti di automazione. I compiti del programma utente comprendono la definizione delle condizioni per il nuovo avviamento chiamato avviamento a caldo o per il riavviamento della PLC, l'elaborazione dei dati di processo per esempio combinazione di segnali digitali, lettura di valori analogici, determinazione di segnali digitali di uscita, emissione di valori analogici. Comprende inoltre il controllo e la reazione agli allarmi e la gestione di anomalie intervenute durante la normale esecuzione del programma. Il software STEP 7 offre la possibilità di strutturare il programma utente, ovvero di suddividerlo in singole sezioni indipendenti, in un certo senso viene adottato il paradigma di programmazione *object oriented*, in questo modo si ottengono i seguenti vantaggi:

- Programmi di grandi dimensioni essendo suddivisi in blocchi risultano più chiari.
- Le singole parti del programma possono essere riutilizzate in altri progetti.
- L'organizzazione del programma viene semplificata.
- Le modifiche del programma si possono eseguire più facilmente in quanto possono essere fatte a livello di blocco.
- Il test del programma viene semplificato, poiché può essere eseguito per blocchi o sezioni

I progettisti della Siemens hanno sviluppato Step 7 in modo che il programmatore possa creare programmi strutturati, mettendo a disposizione diversi tipi di blocchi che si differenziano tra loro per il tipo di servizio che offrono. La tabella in fig:4.2 include l'elenco di tutte le tipologie di blocchi disponibili in Step7.

Blocco	Descrizione della funzione	Vedere anche
Blocchi organizzativi (OB)	Gli OB determinano la struttura del programma utente.	"Blocchi organizzativi e struttura del programma"
Blocchi funzionali di sistema (SFB) e funzioni di sistema (SFC)	Gli SFB e le SFCs sono integrati nella CPU S7, e rendono accessibili alcune importanti funzioni di sistema.	"Blocchi funzionali di sistema (SFB) e funzioni di sistema (SFC)"
Blocchi funzionali (FB)	Gli FB sono blocchi con "memoria", programmabili dall'utente.	"Blocchi funzionali (FB)"
Funzioni (FC)	Le FC contengono routine di programma per le funzioni più utilizzate.	"Funzioni (FC)"
Blocchi dati di istanza (DB di istanza)	I blocchi dati di istanza vengono assegnati al blocco quando viene richiamato un FB/SFB. Essi vengono generati automaticamente nella compilazione.	"Blocchi dati di istanza"
Blocchi dati (DB)	I DB sono aree di dati per la memorizzazione dei dati utente. Oltre ai dati rispettivamente assegnati a un blocco funzionale, possono essere definiti dati globali utilizzabili da blocchi qualsiasi.	"Blocchi dati globali (DB)"

Figura 4.2: Step7, vari componenti usati per la programmazione

Gli OB, FB, SFB, FC e SFC contengono parti del programma, e vengono pertanto definiti come blocchi di codice. Il numero massimo di blocchi e la loro lunghezza ammesse, questo per ogni tipo di blocco dipende dal PLC. Oltre a questi esistono anche i blocchi dati DB dove vengono memorizzate tutte le variabili del programma compreso la loro descrizioni. Nella parte seguente verranno descritti tutti questi blocchi con riferimento il progetto di controllo del forno.

### 4.2.1 Blocchi organizzativi OB

I blocchi organizzativi OB rappresentano l'interfaccia tra il sistema operativo e il programma utente. Vengono richiamati dal sistema operativo e comandano l'elaborazione ciclica del programma, la loro esecuzione si basa su *interrupt software* ed dipende dal tipo di OB. Programmando questi blocchi organizzativi è possibile determinare il comportamento della PLC. I blocchi organizzativi determinano la sequenza di eventi all'avvio in cui verrà elaborato programma utente. L'elaborazione di un OB può essere interrotta dal richiamo di un altro OB in base alla priorità, nel senso che è la priorità a stabilire quale OB andrà in esecuzione. Ovviamente gli OB con priorità più alta interrompono quelli con priorità più bassa. Gli eventi di avvio che provocano il richiamo di un determinato OB vengono

definiti anche allarmi. La tabella in fig:4.3 riporta tutti i tipi di interrupt o allarmi di STEP 7 ed anche la loro priorità . I blocchi organizzativi indicati è le loro classi di priorità non sono necessariamente presenti in tutti modelli PLC di S7.

In alcuni modelli di PLC Siemens, allarmi o interrupt sono parametrizzabili. La parametrizzazione permette per esempio di modificare le classi di priorità nei blocchi di parametri. Come si può vedere dalla tabella in fig:4.3, l'OB1 ha la priorità più alta .È da questo OB che parte l'esecuzione del programma utente, in un certo senso assomiglia alla funzione *main* del linguaggio C. Da questo OB partono le chiamate alle funzioni e blocchi funzione. Gli altri OB interrompono l'esecuzione del OB1 per un determinato periodo di tempo. L'interrupt generato da un OB può essere fatto in modo che dipenda da una variabile ó da un evento legato al funzionamento della macchina.

Ogni blocco organizzativo dispone di informazioni di avvio di 20 byte di dati locali, che il sistema operativo trasferisce all'avvio dell'OB. I dati trasferiti all'avvio contengono informazioni su eventi di avvio dell'OB, data e ora di avvio, errori verificatisi ed eventi di diagnostica.

Nel progetto, sono stati usati l'OB1 come blocco principale del programma, l'OB35, l'OB100 che viene utilizzato per l'avviamento a caldo. L'OB1 è il blocco principale, di conseguenza è sempre incluso in qualsiasi progetto. Per quanto riguarda l'OB35, questo tipo di OB viene usato per eseguire istruzioni a intervalli di tempo regolari. Step 7 mette a disposizione 9 OB di questo tipo, dall'OB30 fino all'OB38, le differenze tra un OB e l'altro sono proprio gli intervalli di tempo di attesa tra una esecuzione e l'altra, alle differenze di tempo hanno anche le differenze di priorità. L'OB35 ha un intervallo di tempo di *default* di 100ms è una classe di priorità 12. Nel progetto, questo OB è stato usato come contattore del consumo di energia, in modo da avere una misura in termini di gas che di elettricità . Per quanto riguarda l'OB100 ,questo OB viene eseguito in caso di avviamento a caldo. Nei PLC si distingue tra tre differenti operazione di avvio chiamata *start-up*, che sono:

- Riavviamento, il processo deve essere completamente reinizializzato.
- Avviamento caldo , il processo riparte dopo una breve interruzione.In questo caso è possibile riprendere il funzionamento dopo una procedure di inzializzazione ridotta.
- Avviamento a freddo, il processo deve essere completamente inzializzato dopo un periodo di fermo.

Tipo di allarme	Blocchi organizzativi	Classe di priorità (predefinita)	Vedere anche
Ciclo libero	OB 1	1	"Blocco organizzativo per l'elaborazione ciclica del programma (OB 1)"
Allarmi dall'orologio	da OB 10 a OB 17	2	"Blocchi organizzativi di allarme dall'orologio (da OB 10 a OB 17)"
Allarme di ritardo	OB 20 OB 21 OB 22 OB 23	3 4 5 6	"Blocchi organizzativi per l'allarme di ritardo (da OB 20 a OB 23)"
Schedulazione orologio	OB 30 OB 31 OB 32 OB 33 OB 34 OB 35 OB 36 OB 37 OB 38	7 8 9 10 11 12 13 14 15	"Blocchi organizzativi di schedulazione orologio (da OB 30 a OB 38)"
Interrupt di processo	OB 40 OB 41 OB 42 OB 43 OB 44 OB 45 OB 46 OB 47	16 17 18 19 20 21 22 23	"Blocchi organizzativi per interrupt di processo (da OB 40 a OB 47)"
Allarmi DPV1	OB 55 OB 56 OB 57	2 2 2	Programmazione di apparecchiature DPV1
Allarme di multicomputing	OB 60 Multicomputing	25	Funzionamento sincrono in multicomputing di diverse CPU
Allarmi di sincronismo di clock	OB 61 OB 62 OB 63 OB 64	25	Progettazione di tempi di reazione del processo brevi e della stessa durata nel PROFIBUS DP
Errori di ridondanza	OB 70 Errore di periferia ridondata (solo nei sistemi H) OB 72 Errore di ridondanza CPU (solo nei sistemi H)	25 28	"Blocchi organizzativi per l'elaborazione degli errori (da OB 70 a OB 87 / da OB 121 a OB 122)"
Errori di asincronismo	OB 80 Errore di tempo OB 81 Errore di alimentazione OB 82 Allarme di diagnostica OB 83 Allarme di inserimento/estrazione OB 84 Errore di guasto hardware della CPU OB 85 Errore di esecuzione programma OB 86 Guasto del telaio di montaggio OB 87 Errore di comunicazione	25 (o 28, quando l'OB di errore di asincronismo si verifica nel programma di avviamento)	"Blocchi organizzativi per l'elaborazione degli errori (da OB 70 a OB 87 / da OB 121 a OB 122)"
Ciclo con priorità bassa	OB 90	29 <sup>1)</sup>	"Blocco organizzativo di priorità bassa (OB 90)"
Avviamento	OB 100 Nuovo avviamento (avviamento a caldo) OB 101 Riavviamento OB 102 Avviamento a freddo	27 27 27	"Blocchi organizzativi per l'avviamento (OB 100/OB 101/OB 102)"
Errori sincroni	OB 121 Errore di programmazione OB 122 Errore di accesso	Priorità dell'OB che causa l'errore	"Blocchi organizzativi per l'elaborazione degli errori (da OB 70 a OB 87 / da OB 121 a OB 122)"

1) Alla classe di priorità 29 corrisponde la priorità 0.29. L'OB 90 ha quindi priorità inferiore a quella del ciclo libero.

Figura 4.3: Tutti i blocchi OB utilizzati in Step7

Rispettivamente gli OB eseguiti per ognuna di queste fasi sono OB101,OB100,OB102. Come già menzionato sopra l'OB100 viene eseguito al avviamento del PLC. Questo OB è stato impiegato nel PLC del forno per *resettare* delle variabili che vengono utilizzate come indicatori di allarme nel pannello operatore in caso di riavviamento. In pratica all'avvio del PLC le variabili utilizzate come indicatori di allarme vengono azzerate. Come si vede dalla tabella in fig:4.3 Step7 mette a disposizione del programmatore diversi OB, l'utilizzo di queste OB dipende dal progettista e da come è stato pensato il progetto.

### 4.2.2 Blocchi Funzione FC

Le funzioni FC fanno parte dei blocchi programmati dall'utente. La funzione FC è un blocco di codice privo di memoria, nel senso che non può avere variabili statiche. Le variabili temporanee dell'FC vengono memorizzate nello *stack* dei dati locali della funzione. Dopo l'elaborazione dell'FC, cioè all'uscita dalla funzione, questi dati vanno persi. Per la memorizzazione dei dati, le funzioni possono utilizzare blocchi dati globali DB (descritte di seguito). Dato che ad una FC non è abbinata alcuna memoria, si rende necessario indicarne sempre i parametri attuali. Inoltre ai dati locali di una FC non può essere assegnato alcun valore iniziale. Una funzione FC può essere chiamata all'interno di un'altra FC (FC annidate) o di una FB (descritta di seguito), la stessa FC può essere chiamata più volte durante uno stesso ciclo di esecuzione. Una FC contiene una sequenza di istruzioni che vengono eseguite ogni qualvolta l'FC viene richiamata da un altro blocco. Come in un qualsiasi linguaggio di programmazione le funzioni possono essere utilizzate per i seguenti motivi:

- Restituire un valore al blocco chiamante.
- Eseguire una funzione tecnologica del PLC esempio:FC1.
- Suddividere il programma in blocchi più piccoli.

Nell'immagine in fig:4.4 viene visualizzata la schermata di Step7 della FC130 in cui viene effettuata la chiamata alla FC131. Per quanto riguarda i parametri di una funzione FC, si distingue tra parametri attuali e parametri formali.

I parametri formali sono solo della definizione del tipo di dati che verranno passati alla funzione, ed i parametri attuali sono i dati che utilizza la funzione al momento della sua esecuzione. In pratica i parametri attuali sostituiscono

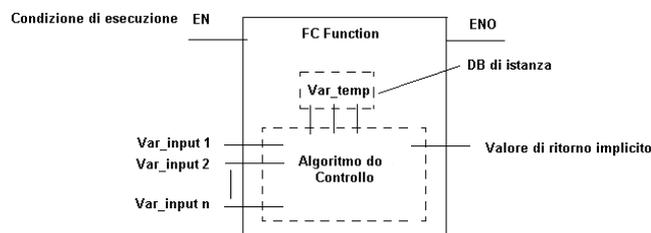
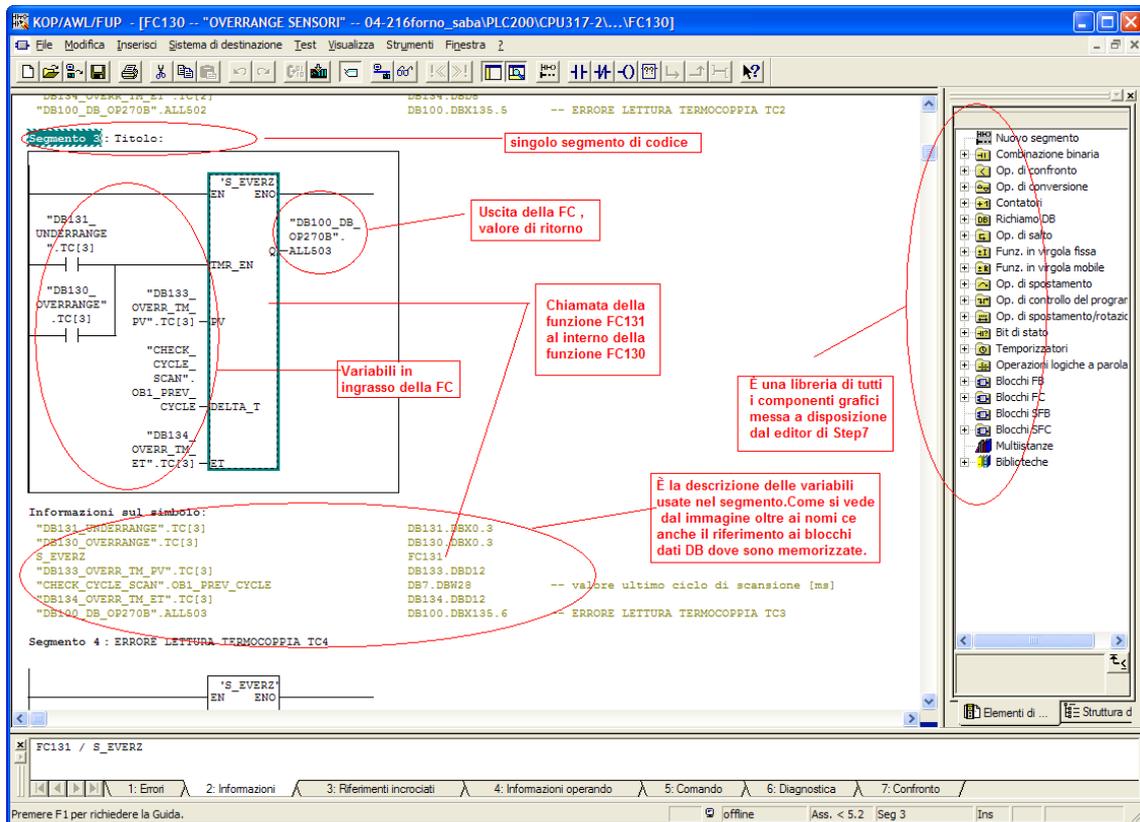


Figura 4.4: Blocco funzione FC in Step7

i parametri formali all'avvio della FC. I parametri che riguardano ingressi e uscite utilizzate da un FC vengono passati per riferimento e non per valore. Nel PLC del forno sono stati impiegati 55 blocchi FC, che vanno dal FC1 a FC790. L'FC1 è una funzione tecnologica messa a disposizione da Step7, non contiene istruzioni definite dal progettista. La sua funzione è quello di modificare il tempo in base a due parametri passati dal utente, in pratica fa la somma dei due tempi e restituisce questa somma. Il primo parametro è un *DT date and time* mentre il secondo è solo *time T*. È stato utilizzato per il calcolo di data e ora, che verrà visualizzata

nell'interfaccia grafica. Tutte le altre funzioni sono a discrezione del progettista sia per la scelta del nome che per il loro numero, infatti nel PLC del forno ne sono stati utilizzati diversi blocchi FC per vari scopi. Una descrizione dei singoli blocchi FC sarebbe eccessivamente lunga, in generale i compiti da svolgere vengono divisi per singole FC in modo da rendere il progetto più chiaro e facilmente modificabile. Per esempio l'FC8 è stata usata per gestire gli allarmi, le FC70/71/74 sono state usate per il controllo dei bruciatori e la fornitura (la portata) di gas; le FC88,89,90 sono state usate per la lettura delle pressioni, delle temperature e delle posizioni delle serrande..ect. Comunque lo schema seguente fig:4.5 descrive la struttura del programma, di come siano legati i vari blocchi l'uno con l'altro.

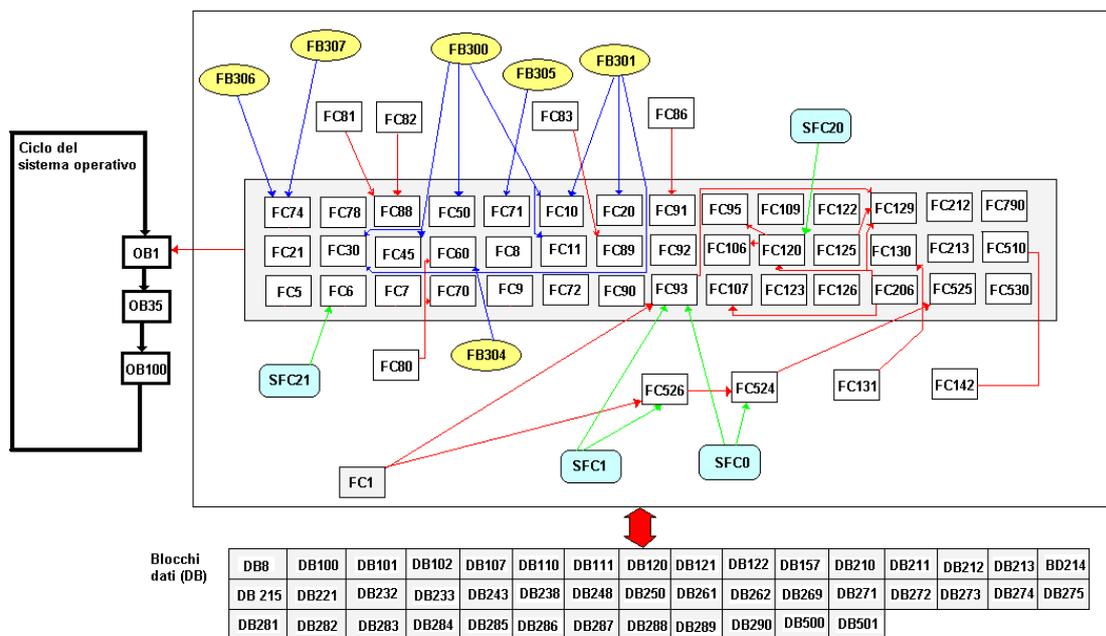


Figura 4.5: Struttura a blocchi del software del PLC che controlla il forno

### 4.2.3 Blocchi Funzione FB

I blocchi funzionali FB fanno parte dei blocchi programmabili dall'utente. Un blocco funzione è un blocco che ha memoria, tale memoria è realizzata mediante l'utilizzo di variabili statiche. Esso dispone di un blocco dati correlato come

memoria chiamato blocco dati di istanza o DB di istanza, che verra descritta in seguito. Sia i parametri che vengono trasmessi all'FB, sia le variabili statiche vengono memorizzati nel blocco dati di istanza. Le variabili temporanee, quelle che non sono persistenti, vengono memorizzate nello *stack* dei dati locali. Al termine dell'elaborazione del FB, i dati memorizzati nel DB di istanza non vanno persi. Per quanto riguarda le variabili memorizzate nello *stack* essendo queste ultime dati locali all'uscite della FB questi vengono persi. Come nel caso delle FC anche le FB contengono un insieme di righe di codice che vengono eseguito ogni qualvolta l'FB viene richiamata. Nell'immagine in fig:4.6 vine visualizzata la schermata di Step7 della FB300 in cui viene effettuata la chiamata alla FC131.

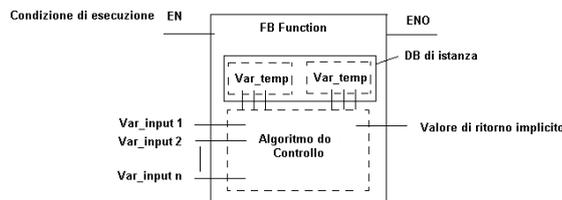
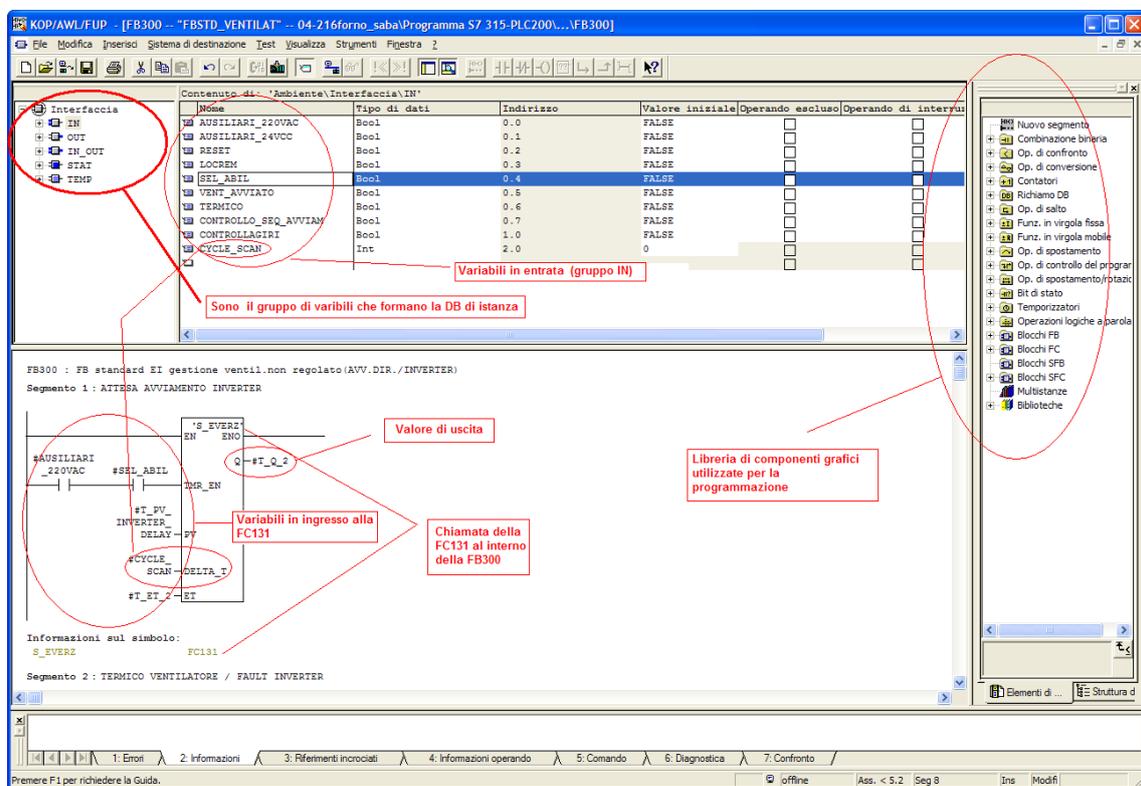


Figura 4.6: Blocco funzione FB in Step7

Come si vede dall'immagine fig:4.6, nella parte superiore della finestra si vedono le variabili che formano la DB di istanza. Come si può vedere nella parte destra di questa schermata le variabili sono divise per gruppi. Il gruppo di *input*(IN) definisce le variabili di ingresso che vengono passate alla FB per essere elaborate. Nel gruppo 2 ci sono le variabili di *output* (OUT), sono le variabili che vengono trasferite in uscita, disponibili al chiamante del blocco come valore di ritorno della FB. Oltre alle variabili di IN e OUT ci sono anche le variabili di INOUT. Queste variabili vengono date in ingresso alla FB, dopo l'elaborazione il valore di ritorno viene memorizzato nella stessa variabile. Le variabili statiche (STAT) sono le variabili che vengono usate all'interno del FB ed il loro valore non viene perso all'uscita dalla FB. Nell'ultimo gruppo ci sono le variabili temporanee (TEMP), sono le variabili utilizzate all'interno della FB, il loro valore non viene mantenuto alla fine della esecuzione. Nel PLC del forno sono stati usati i blocchi funzione :FB300,FB301,FB304,FB305,FB305,FB307,FB350,FB500. Sono stati impiegati per il controllo della ventilazione, per il controllo dei bruciatori, per il controllo della pressione, per la gestione da locale a remoto del impianto. Si faccia riferimento all'immagine in fig:4.5 per avere una visione di come sono interconnessi i blocchi FB con i blocchi FC.

### Differenze tra FC e FB

Quando viene chiamata una FB, l'accesso ai parametri viene fatto su una copia dei parametri attuali nella DB di istanza. Se nel richiamare un FB non viene assegnato un parametro di ingresso o se il risultato non viene scritto in un parametro di uscita nel blocco, allora la FB utilizza i valori meno recenti presenti nel DB di istanza. Come già menzionato le funzioni FC non hanno una memoria quindi l'assegnazione dei parametri formali non è opzionale ma obbligatoria. L'accesso ai parametri di una FC si esegue tramite indirizzi e puntatori chiamati puntatori multiarea. Per l'assegnazione dei parametri viene memorizzata temporaneamente una copia del parametro attuale nei dati locali del blocco chiamante. Sono queste essenzialmente le differenze tra i blocchi funzione FB e le funzioni FC.

### 4.2.4 Blocchi Dati DB

A differenza dei blocchi di codice tipo FB , FC , i blocchi dati DB non hanno istruzioni, ma servono alla memorizzazione di valori. Nei blocchi dati sono quindi contenuti i dati con cui opera il programma utente. I blocchi dati globali DB ,memorizzano i dati utente , questi dati sono accessibili o visibili a tutti gli altri

blocchi sia essi FB,FC,OB. Nell'immagine in fig:4.7 viene visualizzata la schermata di Step7 della DB274 in cui viene effettuata la chiamata alla FC131.

Indirizzo	Dichiarazione	Nome	Tipo	Valore iniziale	Valore attuale	Descrizione
1	0.0 in	AUSILL...	BOOL	FALSE	TRUE	PRESENZA AUSILIARI 220VAC
2	0.1 in	AUSILL...	BOOL	FALSE	TRUE	PRESENZA AUSILIARI 24VCC
3	0.2 in	RESET	BOOL	FALSE	FALSE	PULSANTE RESET ALLARMI
4	0.3 in	LOCREM	BOOL	FALSE	TRUE	REGOLAZIONE SELETTORE
5	0.4 in	SEL_A	BOOL	FALSE	TRUE	SELETTORE ABILITAZIONE
6	0.5 in	SEL_B	BOOL	FALSE	TRUE	SELETTORE ABILITAZIONE
7	0.6 in	SEL_C	BOOL	FALSE	TRUE	SELETTORE ABILITAZIONE
8	0.7 in	AL_G...	BOOL	FALSE	FALSE	ALLARME GRUPPO IN BLOCCO
9	1.0 in	BRUC...	BOOL	FALSE	TRUE	BLOCCO BRUCIATORE 1
10	1.1 in	BRUC...	BOOL	FALSE	TRUE	BLOCCO BRUCIATORE 2
11	1.2 in	BRUC...	BOOL	FALSE	FALSE	BLOCCO BRUCIATORE 3
12	1.3 in	BRUC...	BOOL	FALSE	FALSE	BLOCCO BRUCIATORE 4
13	1.4 in	AL_TC	BOOL	FALSE	FALSE	ALLARME ERRORE LETTURA TERMOCOPPIA DI REGOLAZIONE
14	1.5 in	CONS...	BOOL	FALSE	TRUE	CONSENSO PER FIAMMA ALTA
15	1.6 in	BRUC...	BOOL	FALSE	TRUE	Bruciatore in funzione
16	2.0 in	CYCL...	INT	0	9	CICLO SCANSIONE PLC
17	4.0 in	MISURA	REAL	0.000000e...	5.460000e...	MISURA TEMPERATURA
18	8.0 in	SET	REAL	0.000000e...	6.750000e...	SET TEMPERATURA
19	12.0 in	BM	REAL	0.000000e...	5.000000e...	BANDA MORTA
20	16.0 out	MSG...	BOOL	FALSE	FALSE	ALLARME BRUCIATORE IN BLOCCO 1
21	16.1 out	MSG...	BOOL	FALSE	FALSE	ALLARME BRUCIATORE IN BLOCCO 2
22	16.2 out	MSG...	BOOL	FALSE	FALSE	ALLARME BRUCIATORE IN BLOCCO 3
23	16.3 out	MSG...	BOOL	FALSE	FALSE	ALLARME BRUCIATORE IN BLOCCO 4
24	16.4 out	MSG...	BOOL	FALSE	FALSE	ALLARME GRUPPO
25	16.5 out	MSG...	BOOL	FALSE	FALSE	ALLARME BRUCIATORE NON AVVIATO (INTERVENTO TERMOSTATO DI SICUR)
26	16.6 out	MSG...	BOOL	FALSE	FALSE	MSG DISCORDANZA ABILITAZIONI TRA SELETTORE E PC
27	16.7 out	ABILITA	BOOL	FALSE	TRUE	USCITA DI COMANDO ABILITA BRUCIATORE

Messaggi  
 E' stata caricata la vista standard in quanto l'attributo di sistema adeguato non è stato impostato o manca.

Ripristina la finestra alla sua dimensione normale. offline

Figura 4.7: Blocco dati DB in Step7

Le dimensioni delle DB sono variabili, le dimensioni massime variano in base al modello di PLC e alla quantità di memoria disponibile. La struttura interna delle DB può essere stabilita liberamente. Quando un blocco di codice FC, FB, OB viene chiamato, esso può temporaneamente occupare spazio in memoria nell'area dati locale, in pratica crea una DB di istanza. La DB di istanza nel caso delle FC e OB è un insieme di dati temporaneo, al termine dell'esecuzione di questi blocchi l'area di memoria locale viene liberata. Al contrario della DB di istanza, i dati contenuti nelle DB globali sono persistenti. Ogni FB, FC o OB può accedere in lettura o scrittura in una DB globale ed inoltre è possibile aprire contemporaneamente un DB globale e un DB di istanza. Il vantaggio di dividere i dati in blocchi risiede nel fatto che si possono creare dei blocchi dati che interessano parti specifiche del impianto. Nel senso che variabili uguali si inseriscono

nello stesso gruppo in questo modo si rende piú facile la consultazione dei dati nonché la loro modifica ed è possibile il loro riutilizzo in altri progetti. Nel progetto del forno i blocchi di dati globali DB sono divisi per il loro contenuto, per esempio: le DB che contengono le variabili delle sonde di temperature, valore minimo ,massimo ,misura attuale: contengono le variabili delle sonde di umidità anche in questo caso minimo ,massimo,misura attuale. Vale lo stesso per le variabili delle serrande, dei bruciatori, dei motori ect. Nel progetto è inserita anche una DB che include tutte le variabili utilizzate per gli allarmi e la loro descrizione. Questo raggruppamento dei dati fa sì che il programma finale risulti piú chiaro e comprensibile.

#### 4.2.5 Blocchi Funzione di Sistema SFC e SFB

Step 7 mette a disposizione del progettista delle funzioni di sistema o funzioni predefinite. Questi blocchi funzione non devono essere programmati e possono essere definiti liberamente. Questi blocchi vengono chiamati SFC e SFB, si differenziano dagli altri blocchi per la lettera S posta al inizio del nome .In genere sono blocchi protetti dal sistema quindi non è possibile modificarli. Come per i blocchi funzione FB, FC, anche gli SFB e SFC si differenziano per il tipo di servizio. Come le FC anche le SFC non hanno variabili statiche di conseguenza all'uscita dalla funzione l'area di dati locali stack viene liberata. Al contrario le SFB, simili alle FB, possiedono delle variabili statiche DB di istanza, i valori vengono mantenuti all'uscita dalla funzione . Le funzioni di sistema sono numerose e offrono diversi tipi di servizi, nel progetto del forno sono state usate le SFC0,SFC1,SFC120,SFC121.La prima serve a impostare l'orologio di sistema, la seconda legge l'ore e restituisce il valore letto, la terza sposta o copia un blocco dati da un DB in un'altra, l'ultima è utilizzata per inizializzare un blocco dati con il contenuto di un altro blocco dati. Alcune di queste funzioni di sistema sono utilizzate anche per la comunicazione tra PLC.

### 4.3 Test del software del PLC

Dopo la fase di programmazione si passa alla fase di collaudo. Per collaudare il software si hanno due possibilità: la prima possibilità è quella di usare un PLC dotato delle sue unità di ingressi e uscite. La seconda possibilità è quella di usare un PLC software. Il pacchetto Step 7 include un software che svolge questa funzione che si chiama PLC Simulator fig:4.8.

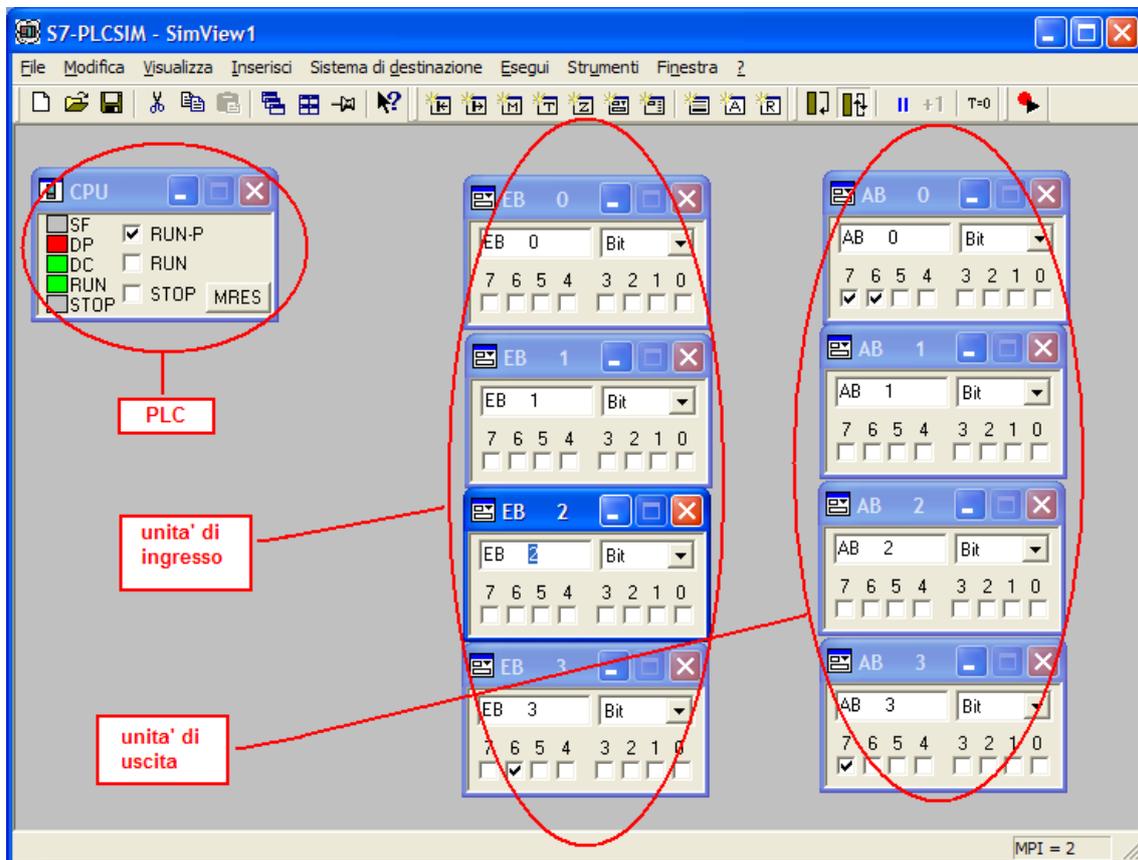


Figura 4.8: Step 7: PLC Simulator

Questo software oltre al PLC simula anche gli I/O sia digitali che analogiche, nonché temporizzatori, permette inoltre di impostare i singoli bit, sia in ingresso che in uscita in modo da testare il software. Ovviamente essendo un PLC software i tempi di ciclo e di esecuzione non sono esattamente uguali a quelli di un PLC hardware. Se i risultati sono corretti, se non si presentano errori o anomalie di alcun tipo allora si passa alla fase successiva che come descritto sopra è quella della messa in servizio. Essendo questa attività un lavoro di gruppo si cerca di portare avanti in parallelo sia lo sviluppo software del PLC, che quello di modellazione dello SCADA. Nella prossima parte del capitolo verrà mostrato il sistema di raccolta dati realizzato con WinCC [2] riguardante il progetto del forno.

## 4.4 Modellazione del software SCADA

Nel capitolo 3 è stato descritto lo SCADA in generale, del fatto che rappresenta in forma grafica lo stato di un segnale proveniente dall'impianto. La forma grafica di questo segnale può essere planimetrica legata al disegno della macchina, oppure in forma tabellare tipo allarmi, oppure ancora in forma di grafico temporale. Come già menzionato lavorare con tali software non costituisce una vera e propria programmazione anche se in genere gli SCADA offrono la possibilità di creare delle procedure personalizzate tramite linguaggi quali; VB, C, java . Si tratta di una modellazione del software fino a fargli assumere la forma voluta. Nella fig:4.9 viene rappresentato lo SCADA di Siemens, viene riportata la schermata principale di tale software.

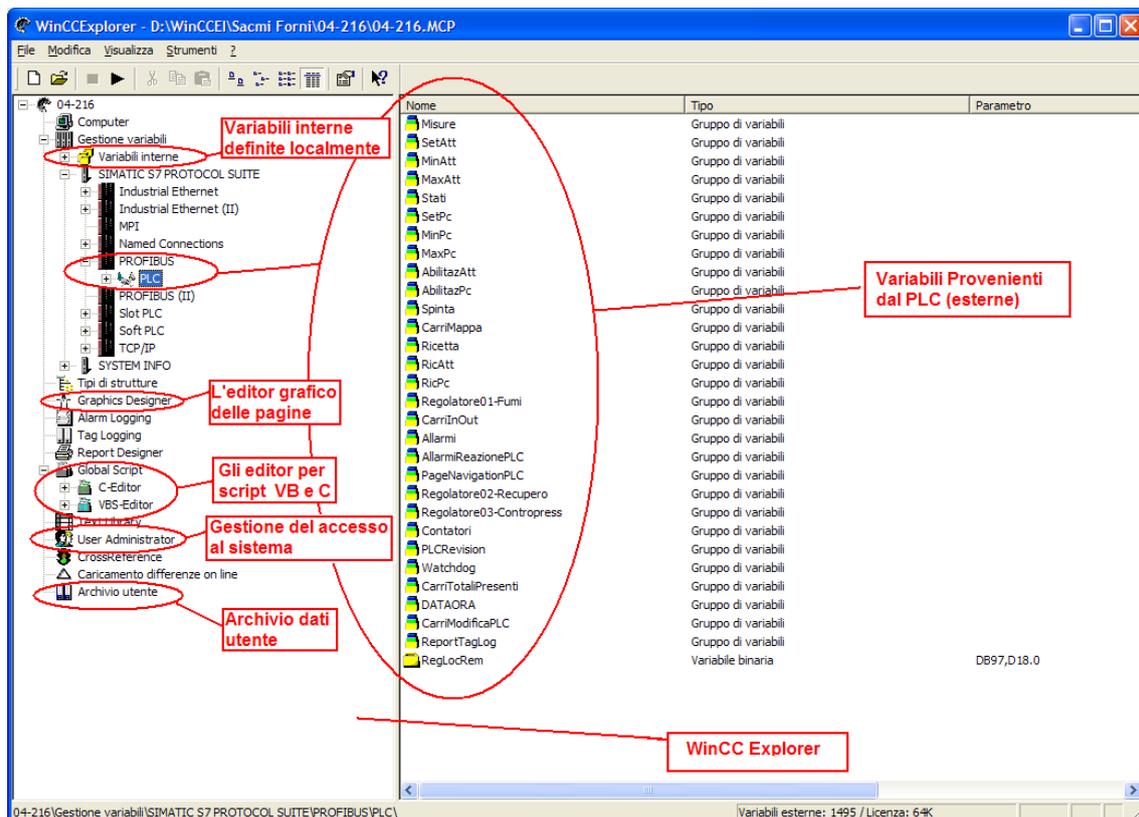


Figura 4.9: Simatic WinCC : schermata principale l'explorer.

Come si vede dall'immagine la schermata principale WinCC Explorer si presenta divisa in due aree, sulla parte di sinistra ci sono tutti i componenti software del pacchetto WinCC, quali: *Global Script*, componente che include C-

editor e VBS-editor , *Graphics Designer* ,componente utilizzato per creare le pagine grafiche, *User Archive*, é l'archivio utente dove vengono memorizzati tutti i dati di processo. Per quanto riguarda le variabili ne esistono di due tipi , variabili interne e variabili esterne. Le variabili interne sono variabili definite localmente, a livello di progetto, cioè non provenienti dal PLC , sono visibili e utilizzabili in tutto il progetto.

Le variabili esterne sono le variabili provenienti dal PLC ed in questo caso dal PLC del forno. Anche queste sono variabili definite localmente, má ogni singola variabile fa riferimento ad un indirizzo di un area di memoria del PLC, che sono poi le DB di dati del PLC. È compito del sotto sistema di comunicazione di WinCC gestire la comunicazione con il PLC facendo in modo che il valore delle variabili sia sempre aggiornato. WinCC gestisce diversi tipi di protocolli di comunicazione, quindi prima bisogna selezionare il tipo di protocollo di comunicazione con il PLC è successivamente si creano le variabili. Come si vede dal immagine fig:4.9 il protocollo di comunicazione selezionato è il PROFIBUS in quanto tutta la rete è basata su questo protocollo. È buona norma creare tutte le variabili che verranno utilizzate nel progetto prima di iniziare il lavoro. Essendo queste variabili locali che fanno riferimento a aree di memoria del PLC, solitamente si assegna a tali variabili è lo stesso nome che viene usato all'interno del PLC questo per motivi di coerenza dei nomi, è soprattutto per chiarezza. Inoltre WinCC supporta la tecnologia OLE *Object Linking and Embedding* che permette di inserire in un documento, parte di un altro documento, ad esempio un grafico, un'immagine, una tabella...ect, creato con un altro programma, questa funzionalità aiuta molto nel creare alcuni oggetti grafici nel graphics designer.

#### 4.4.1 La strutture delle pagine

Le pagine grafiche nel progetto del forno sono suddivise in tre zone che sono:

- a) intestazione
- b) area di lavoro
- c) area comandi

Come mostrato dall'immagine seguente fig:4.10.

Nella parte alta della pagine dove è posizionata l'intestazione fig:4.11, vengono visualizzate alcune informazioni, quali: ricetta attiva che indica la ricetta che per ultima è stata attivata sull'impianto. Inoltre viene visualizzato il campo *carri/giorno*, che indica la cadenza di entrata dei carri nel forno, espressa in carri giorno. Sempre nella intestazione viene indicato se stiamo conducendo il forno



Figura 4.10: Struttura delle pagine grafiche

dal PC di supervisione, è cioè da remoto (REM), o dal quadro comandi con il pannello operatore locale (LOC). Il campo *spinta*, indica in minuti ogni quanto tempo si ha una spinta, avanzamento dei carri al interno del forno.

 elettrotecnica imolese	06/05/2005 16.05.12		<b>P030-CURVA COTTURA</b>		FORNO TEGOLE						
	[Alt+F2] LOGIN:										
Ricetta attiva Tegole	Carrig	40	Reg	REM	Spinta	3 : 00 Fase 10	SPINTA IN CORSO	Pross.spinta	2 : 10	Usc.Carro	15 : 30 : 10
Ricetta stor. Tegole	Carrig	40	06/05 05:00								

Figura 4.11: Struttura delle pagine grafiche

In questa area sono contenute anche altre informazioni quali: la lingua del testo visualizzato, l'intestazione e il numero della pagina, nonché i vari marchi di fabbrica. L'area di lavoro è riservata alla visualizzazione delle informazioni provenienti dal PLC e alla rappresentazione grafica di alcune di queste informazioni, per la visualizzazione di grafici, di layout del impianto, di report ed infine degli allarmi. Per quanto riguarda la parte di comando, questa area è stata organizzata in forma di barra dei pulsanti e menu a scelta multipla. In questa barra sono stati inseriti i pulsanti per la navigazione delle pagine (cambiare le pagine), pulsanti per la visualizzazione degli allarmi, dei grafici di cottura, dei layout dell'impianto. È stato inserito un menu a tendina per la selezione di altre pagine non raggiungibili direttamente dal area di comando, ovviamente la stes-

sa barre di pulsanti è stata replicata su ogni pagine del progetto. Qui di seguito vengono descritte le funzionalità principali del progetto.

#### 4.4.2 Funzionalità del progetto

Il progetto è stato realizzato su un Sistema operativo Microsoft e con ambiente di sviluppo WinCC V6.0. Il sistema acquisisce i dati dal PLC di controllo del forno e con esso interagisce per la regolazione dei parametri di cottura. Di seguito vengono descritte le caratteristiche del progetto e delle sue funzionalità, che sono:

- Archivio della Curva di cottura, lo storico: Archiviazione dei valor ogni 10 minuti della curva di cottura e dei dati di regolazione, con la possibilità di richiamarli specificando data e l'ora della curva richiesta. La durata massima dei dati archiviati è di un anno.
- Curva di cottura: Visualizza il profilo dell'andamento delle temperature lungo l'intero forno e visualizzazione dei parametri delle regolazioni principali.
- Gestione dei materiali: Da la possibilità di visualizzare il materiale presente su ogni carro nella linea, inoltre da la possibilità di visualizzare le caratteristiche del materiale presente sul carro e del codice associato a ciascun carro.
- Registratore: Offre la possibilità di archiviazione ad ogni minuto dei dati dell'impianto, è possibile visualizzare l'andamento temporale *real-time* mediante grafico temporale. Anche in questo caso la permanenza dei dati sul disco é di circa un anno.
- Gestione delle Ricette: Gestione di un database di ricette archiviate sul disco, identificate da un nome e dal numero di carri al giorno, con possibilità attivare o copiare i dati attivi sul PLC.
- Gestione Allarmi: Pagina di visualizzazione e riconoscimento degli allarmi, inoltre archivia di tutti i messaggi d'allarme, è offre la possibilità al operatore di richiamare la lista degli allarmi archiviati.
- Sinottico: Visualizza il layout del forno con la rappresentazione dello stato dei suoi componenti e dell'esatta posizione dei carri nel forno, offre la possibilità di visualizzare i valori di temperatura e modificarli.

- Gestione database dei materiali: La pagina di gestione del database dei materiali in produzione, con impostazione delle caratteristiche di ogni materiale per esempio numero pezzi sul carro, peso, descrizione ect.
- Gestione dei dati di produzione: Registrazione dei consumi di produzione, report di produzione con totali parziali in base al materiale e totale globale dell'intera produzione, analisi dei dati in base ad intervallo tra due date.

Queste sono alcune delle caratteristiche principali del progetto, possiamo dire che sono caratteristiche comuni alla maggior parte dei progetti realizzati in azienda. Di seguito verranno descritte queste funzionalità e la loro realizzazione. L'immagine in fig:4.12 mostra come si presenta la pagina principale del progetto all'avvio del *runtime*.

	05/05/2005 14.38.07		<b>P001- MENU</b>				LOGO.....						
	Ricetta attiva Tegole	Carrig	40	Reg	REM	Spinta	3 : 00	Fase	10	SPINTA IN CORSO	Pross.spinta	2 : 10	Usc.Carro



ATTENDERE: AVVIAMENTO IN CORSO ...

REALIZZAZIONE  
ELETTRONICA IMOLESE s.r.l.  
Via Pagnina 4 - Bubano di Mordano (BO)

ALT+F1 STAMPA	F2 MENU	F3 ALLARMI	F4 SCELTA PAGINA	F5 CURVA DI COTTURA		SHIFT+F2 EXIT	SHIFT+F3 MANUTENZ E.I.	SHIFT+F4 ESP/IMP DATI	SHIFT+F5 CAMBIA PASSWORD
------------------	------------	---------------	------------------------	---------------------------	--	------------------	------------------------------	-----------------------------	--------------------------------

Figura 4.12: Pagina principale del progetto.

La pagina principale chiamata Menu appare all'avvio del runtime, è stato realizzato il modo che questa pagina sia raggiungibile da qualsiasi altra pagine del progetto tramite il tasto funzione F2.

### Curva di cottura

Nella pagina fig:4.13 viene visualizzata la curva di cottura e le relative impostazioni. Il grafico rappresenta fino a 5 curve diverse contemporaneamente, utilizzando colorazioni diverse per ogni curva. Per il disegno delle curve sono stati creati degli script in linguaggio C che permettono di disegnare tali curve.

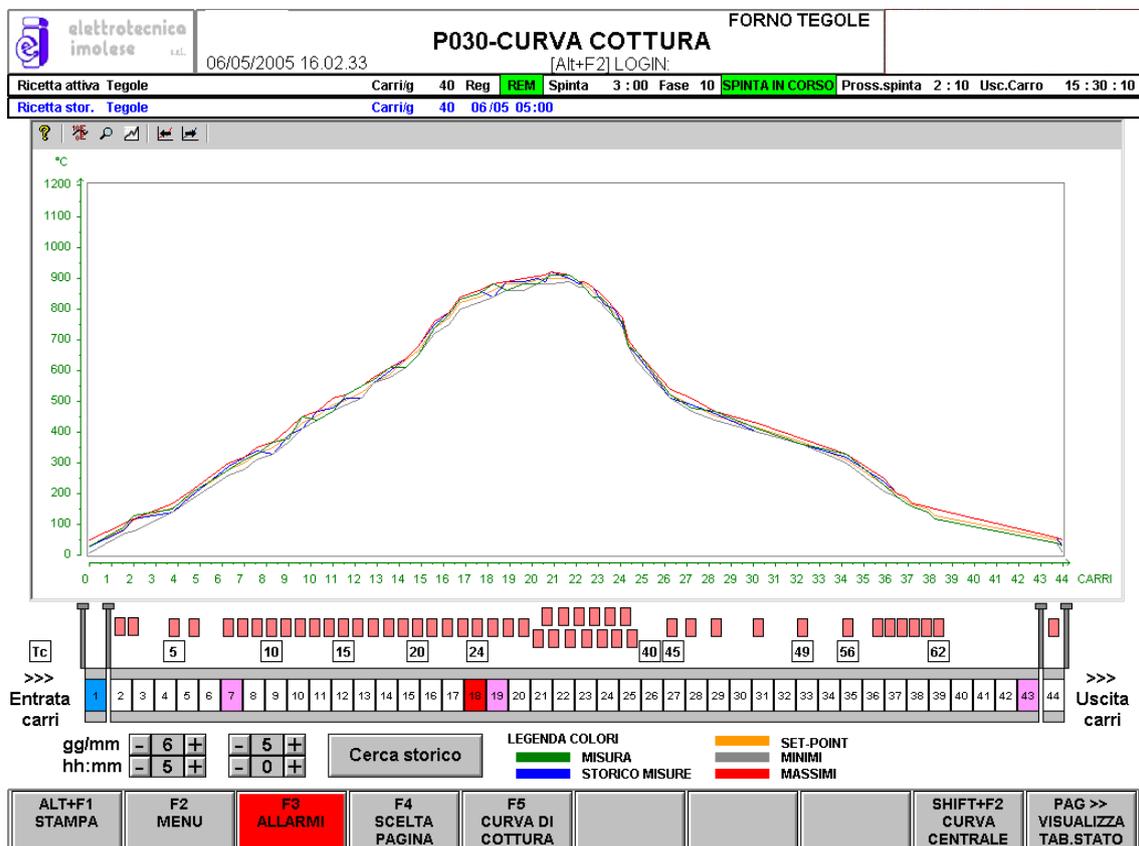


Figura 4.13: Pagina della curva di cottura.

La curva verde visualizza le misure attuali delle termocoppie. La curva blu visualizza misure storiche di cottura. La curva dei set-point delle termocoppie è disegnata in arancione. La curva grigia visualizza i minimi di temperatura impostati per le termocoppie. In fine la curva rossa visualizza le temperature massime

impostate sempre per le termocoppie. Sulle valori di minimo e di massimo viene fatto un controllo con il valore attuale, se tale valore supera questi limiti si genera un allarme. Da questa pagina è possibile cambiare le impostazioni di massimi e minimi di ciascuna termocoppia. Utilizzando le funzionalità messe a disposizione da WinCC sono stati inseriti oggetti grafici che fanno riferimento a queste termocoppie, infatti cliccando direttamente sul quadratino rosa che identifica una termocoppia si apre la relativa finestra 4.14, un messaggio di *tool-tips* che identifica il sensore. In WinCC ad un oggetto grafico è possibile associarli una variabile o anche una funzione C o VBS. Un oggetto grafico, in questo caso un quadrato, che identifica una termocoppia gli è stato associata una funzione C, all'evento del click del mouse su questo quadrato viene eseguita la funzione associata aprendo una finestra di notifica delle stato della termocoppia. Nella finestra relativa alla termocoppia è possibile modificare il valori di processo.

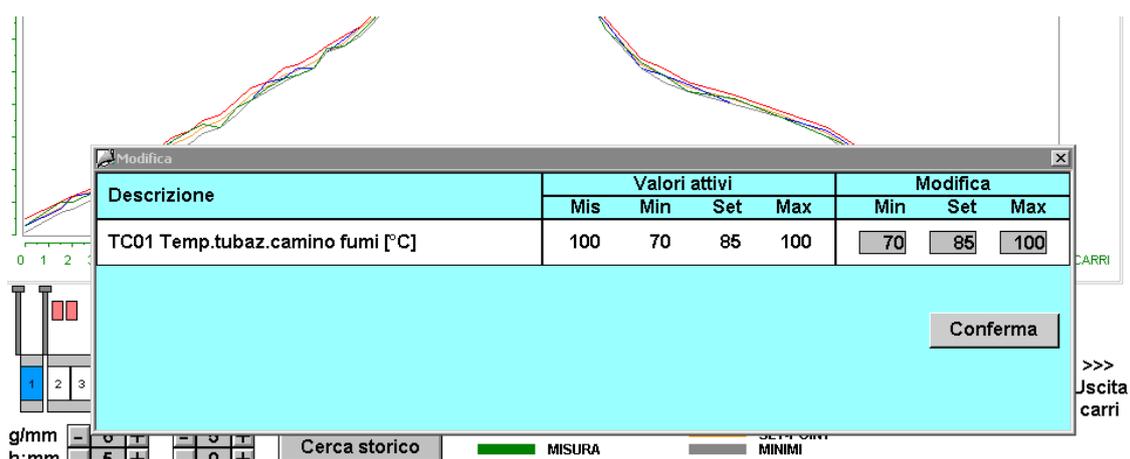


Figura 4.14: Finestra grafica riferita ad una sonda di temperatura.

È possibile visualizzare i dati storici di cottura, inserendo la data e l'ora desiderata negli appositi campi e premendo il tasto *Cerca Storico*. L'intestazione delle pagine che visualizzano dati storici riporterà la data e l'ora ricercata. Ovviamente anche questa funzione è stata creata con delle azioni C che interrogano il database SQL in quanto WinCC non dispone di tali funzioni.

### Tabella di stato

Nella pagina della curva di cottura al fig:4.14 tasto *Tab.stato* è stato associato una azione che quando viene premuto richiama la pagina della tabella di stato in cui

sono visualizzati i valori attuali più importanti e i valori storici misurati, questo per tutti gli attuatori, sensori, ventilatori ..ect. Per la ricerca storica si deve impostare la data e l'ora desiderata nella pagina curva di cottura.L'immagine in fig:4.15 mostra la strutture della pagina di stato.

elettrotecnica imolese		FORNO TEGOLE					
10/05/2005 12.28.56		P031-VISUALIZZA TABELLA STATO 1/12					
[Alt+F2] LOGIN: Elettrotecnica							
Ricetta attiva Tegole	Carrig	30 Reg	REM Spinta 4:00 Fase 10 SPINTA IN CORSO				
Pross.spinta 1:56 Usc.Carro		12:34:49					
Ricetta stor. Tegole	Carrig	30	10/05 10:40				
Descrizione	Storico		Valori attivi				
	Stato	Mis	Stato	Mis	Min	Set	Max
TC01 Temp.tubaz.camino fumi [°C]		80		80	70	85	100
TC02S Temp.volta zona camino fumi sx [°C]		80		120	80	115	120
TC02D Temp.volta zona camino fumi dx [°C]		111		81	81	116	121
TC03 Temp.tubaz.riciclo sx fumi [°C]		110		130	100	170	140
TC04 Temp.tubaz.riciclo dx fumi [°C]		130		130	130	140	160
TC05 Temp.volta zona preriscaldamento [°C]		140		140	140	155	170
TC06 Temp.volta zona preriscaldamento [°C]		190		190	180	195	210
TC07 Temp.volta bruc.later.gr.03 [°C]	DIS	260	DIS	280	260	280	300
TC08 Temp.volta bruc.later.gr.04 [°C]	DIS	290	DIS	310	280	300	320
TC09 Temp.volta bruc.later.gr.05 [°C]	DIS	310	DIS	330	310	330	350
TC10 Temp.volta bruc.later.gr.06 [°C]	DIS	350	DIS	370	330	350	370
TC11 Temp.volta bruc.later.gr.07 [°C]	DIS	410	DIS	380	370	390	410
TC12 Temp.volta bruc.later.gr.08 [°C]	DIS	430	DIS	450	410	430	450
TC13 Temp.volta bruc.later.gr.09 [°C]	DIS	470	DIS	470	440	455	470
TC14 Temp.volta bruc.later.gr.10 [°C]	DIS	500	DIS	470	470	490	510
TC15 Temp.volta bruc.later.gr.11 [°C]	DIS	510	DIS	510	490	505	520
TC16 Temp.volta bruc.later.gr.12 [°C]	DIS	530	DIS	550	510	530	550
TC17 Temp.volta bruc.later.gr.13 [°C]	DIS	580	DIS	560	560	560	580
TC18 Temp.volta bruc.later.gr.14 [°C]	DIS	600	DIS	600	580	590	610
TC19 Temp.volta bruc.later.gr.15 [°C]	DIS	640	DIS	640	610	635	640
TC20 Temp.volta bruc.later.gr.16 [°C]	DIS	680	DIS	680	650	665	680
TC21 Temp.volta bruc.later.gr.17 [°C]	DIS	720	DIS	740	720	740	760
TC22 Temp.volta bruc.later.gr.18 [°C]	DIS	760	DIS	780	750	770	790
TC23 Temp.volta bruc.later.gr.19 [°C]	DIS	810	DIS	830	800	820	840
TC24 Temp.volta bruc.later.gr.20 [°C]	DIS	830	DIS	850	820	840	860
TC24S Temp.volta bruc.later.gr.20 sx [°C]		821		841	821	841	861
TC24D Temp.volta bruc.later.gr.20 dx [°C]		822		842	822	842	862

ALT+F1 STAMPA	F2 MENU	F3 ALLARMI	F4 SCELTA PAGINA	F5 CURVA DI COTTURA				<< PAG INDIETRO	PAG >> AVANTI
------------------	------------	---------------	------------------------	---------------------------	--	--	--	--------------------	------------------

Figura 4.15: Pagina riferita allo stato dei sensori.

In questa immagine viene mostrata la prima delle 12 pagine che permettono di visualizzare oltre allo stato degli attuatori anche il confronto di dati attuali con dati storici. È possibile scorrere tutte le 12 pagine premendo con il mouse su indietro-avanti, o direttamente con i tasti della tastiera PageUp e PageDown. Nei tasti PAG è stato associato l'evento del mouse che richiama la pagina precedente o successiva.

## Ricette è la loro gestione

Viene chiamato ricetta un insieme di impostazione che permettono di impostare un impianto per un determinato tipo di produzione. Alla pagina delle ricette fig:4.16 si accede premendo da tutte le pagine la sequenza di tasti funzione [F4 F8] oppure con il mouse selezionare il tasto F4 Scelta Pagina, è poi F8 Ricette. Le pagine di gestione delle ricette sono suddivise in due sezioni:

- 1a)la gestione della ricetta attiva
- 2a)la gestione del database delle ricette su disco.

L'immagini fig:4.16 mostra come si presenta questa pagina.

elettrotecnica imolese		10/05/2005 12.03.06		P101-RICETTE 1/12		FORNO TEGOLE					
Ricetta attiva Tegole		Carrig	30	Reg	REM	Spinta	4 : 00 Fase 10 SPINTA IN CORSO	Pross.spinta	1 : 56	Usc.Carro	12 : 09 : 01
Descrizione	Valori attivi					Valori di ricetta in editazione					
	Stato	Mis	Min	Set	Max	Abilitazione	Min	Set	Max		
TC01 Temp.tubaz.camino fumi [°C]		90	70	85	100		70	85	100		
TC02S Temp.volta zona camino fumi sx [°C]		110	80	115	120		80	115	120		
TC02D Temp.volta zona camino fumi dx [°C]		81	81	116	121		81	116	121		
TC03 Temp.tubaz.riciclo sx fumi [°C]		130	100	170	140		100	170	140		
TC04 Temp.tubaz.riciclo dx fumi [°C]		140	130	140	160		130	140	160		
TC05 Temp.volta zona preriscaldamento [°C]		150	140	155	170		140	155	170		
TC06 Temp.volta zona preriscaldamento [°C]		200	180	195	210		180	195	121		
TC07 Temp.volta bruc.later.gr.03 [°C]	DIS	280	260	280	300	0	260	280	300		
TC08 Temp.volta bruc.later.gr.04 [°C]	DIS	310	280	300	320	0	280	300	320		
TC09 Temp.volta bruc.later.gr.05 [°C]	DIS	330	310	330	350	0	310	330	350		
TC10 Temp.volta bruc.later.gr.06 [°C]	DIS	370	330	350	370	0	330	350	370		
TC11 Temp.volta bruc.later.gr.07 [°C]	DIS	380	370	390	410	0	370	390	410		
TC12 Temp.volta bruc.later.gr.08 [°C]	DIS	450	410	430	450	0	410	430	450		
TC13 Temp.volta bruc.later.gr.09 [°C]	DIS	440	440	455	470	0	440	455	470		
TC14 Temp.volta bruc.later.gr.10 [°C]	DIS	470	470	490	510	0	470	490	370		
TC15 Temp.volta bruc.later.gr.11 [°C]	DIS	520	490	505	520	0	490	505	520		
TC16 Temp.volta bruc.later.gr.12 [°C]	DIS	550	510	530	550	0	510	530	550		
TC17 Temp.volta bruc.later.gr.13 [°C]	DIS	580	560	560	580	0	560	560	580		
TC18 Temp.volta bruc.later.gr.14 [°C]	DIS	610	580	590	610	0	580	590	610		
TC19 Temp.volta bruc.later.gr.15 [°C]	DIS	610	610	635	640	0	610	635	640		
TC20 Temp.volta bruc.later.gr.16 [°C]	DIS	650	650	665	680	0	650	665	680		
TC21 Temp.volta bruc.later.gr.17 [°C]	DIS	740	720	740	760	0	720	740	760		
TC22 Temp.volta bruc.later.gr.18 [°C]	DIS	780	750	770	790	0	750	770	790		
TC23 Temp.volta bruc.later.gr.19 [°C]	DIS	830	800	820	840	1	800	820	840		
TC24 Temp.volta bruc.later.gr.20 [°C]	DIS	850	820	840	860	0	820	840	860		
TC24S Temp.volta bruc.later.gr.20 sx [°C]		841	821	841	861		821	841	861		
TC24D Temp.volta bruc.later.gr.20 dx [°C]		842	822	842	862		999	842	862		

Salva modifica N.Carrig 30

ALT+F1 STAMPA	F2 MENU	F3 ALLARMI	F4 SCELTA PAGINA	F5 CURVA DI COTTURA	SHIFT+F2 GESTIONE RICETTE			<< PAG INDIETRO	PAG >> AVANTI
------------------	------------	---------------	------------------------	---------------------------	---------------------------------	--	--	--------------------	------------------

Figura 4.16: Pagine dedicate alla gestione delle ricette.

In figura è mostrata la prima delle 12 pagine totali dedicate alle ricette, servono ad editare tutti i parametri più importanti per la conduzione del forno. Sulla

parte sinistra della pagina sono mostrati i valori attivi nell'impianto, mentre sulla parte destra vengono visualizzati i campi editabili in cui possono essere variati i valori.

In particolare si possono impostare i massimi, minimi e set degli attuatori e dei sensori, il nome della ricetta e la produzione espressa in carri giorno. Inoltre dalla pagina d'impostazioni ricette premendo la sequenza di tasti [Shift+F2] si accede alla pagina di gestione ricette fig:4.17 in cui è possibile impostare tali ricette. Come menzionato sopra per scorrere tutte le pagine si utilizzano i pulsanti di avanti o indietro, o direttamente con i tasti della tastiera PageUp e PageDown.

The screenshot displays the 'Gestione ricette' window with the following data:

ID	RicEdit_NomeRic	RicEdit_SpintaNcg
1	Tegole 1	30
2	Tegole 2	30

ax	Abilitazione	Min	Set	Max
00		70	85	100
20		80	115	120
21		81	116	121
40		100	170	140
60		130	140	160
70		140	155	170
10		180	195	121
00	0	260	290	300
20	0	290	300	320
50	0	310	330	350
70	0	330	350	370
10	0	370	390	410
50	0	410	430	450
70	0	440	455	470
10	0	470	490	370
20	0	490	505	520
50	0	510	530	550
80	0	560	560	580
10	0	580	590	610
40	0	610	635	640
80	0	650	665	680
60	0	720	740	760
90	0	750	770	790
40	1	800	820	840
60	0	820	840	860
61		821	841	861
		822	842	862

Figura 4.17: Pagina impostazioni ricette.

In questa pagina sono state messe a disposizione dell'operatore le seguenti operazioni:

- Tabella ricette: Sono presenti i nomi di tutte le ricette memorizzate nel sistema di supervisione. Con il mouse è possibile selezionare una ricetta alla volta.

- Pulsante Carica: I valori di ricetta in editazione, vengono riempiti con i valori salvati nella ricetta selezionata nella Tabella ricette.
- Pulsante Elimina: Utilizzato per eliminata in modo definitivo la ricetta selezionata nella Tabella ricette.
- Pulsante Nuova: viene creata una nuova ricetta non vuota. I valori con cui la ricetta viene inizializzata sono gli stessi della ricetta selezionata nella Tabella ricette al momento della pressione del tasto nuova.
- Pulsante Legge da impianto: i valori di ricetta in editazione, vengono riempiti con i valori attivi nell'impianto, cioè con i valori con cui l'impianto sta funzionando. Questo permette di ottimizzare i valori dell'impianto è solo successivamente salvarli in una ricetta.
- Pulsante Attiva ricetta: nell'impianto vengono immediatamente attivati i valori della ricetta selezionata nella Tabella ricette.
- Pulsante Salva modifica: per ragioni di comodità dell'operatore, il pulsante salva modifica ricetta, è posizionato su ogni pagina ricetta. I valori impostati nei campi valori di ricetta in editazione saranno salvati nella posizione correntemente selezionata nella Tabella ricette.

### **Sinottico**

Alla pagina del sinottico si accede premendo da tutte le pagine, la sequenza di tasti funzione [F4 F7] oppure con il mouse selezionare il tasto F4 Scelta Pagina, e poi F7 Sinottico. L'immagine fig:4.18 mostra il layout del forno in modo grafico.

In questa pagina è possibile visualizzare lo stato dei principali attuatori dell'impianto quali: motori, ventole, serrande, bruciatori ect. Il funzionamento viene rappresentato mediante un'associazione dello stato ad un colore i quali sono.

- Grigio: Disabilitato
- Blu: Spento
- Rosso: Allarme
- Verde: Acceso

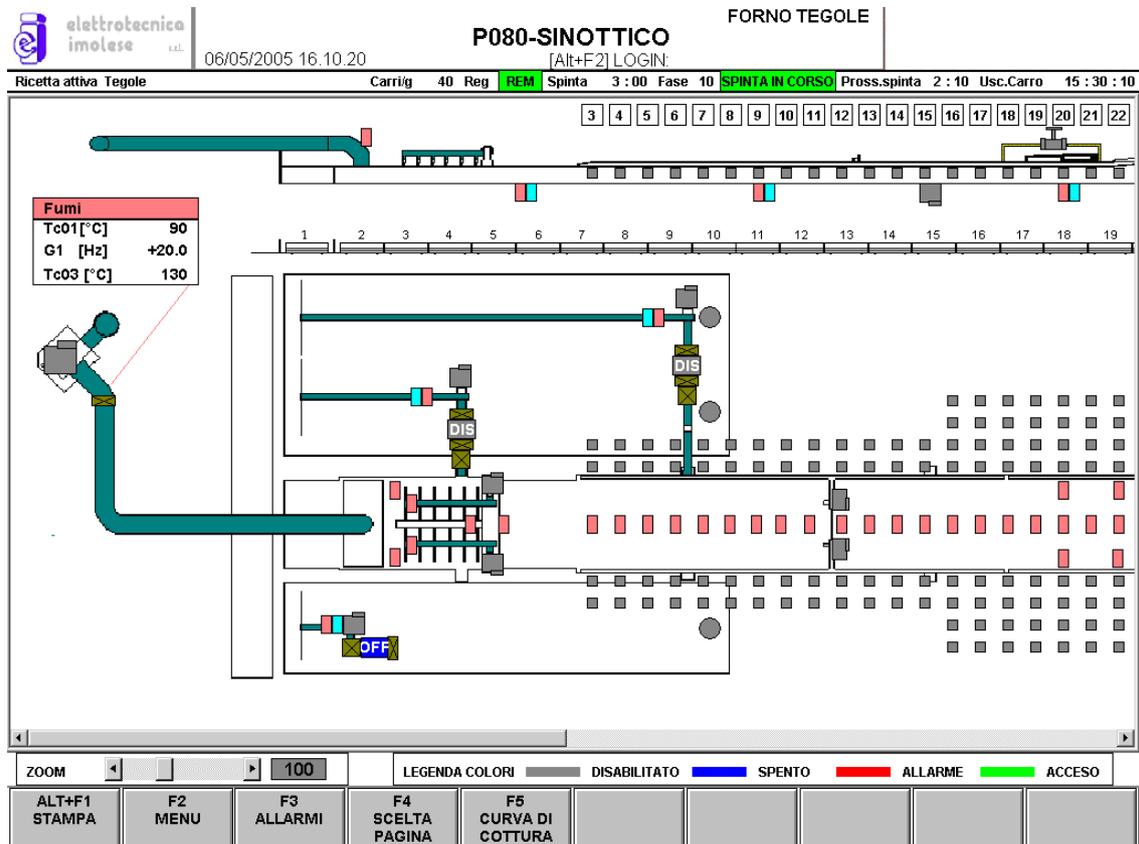


Figura 4.18: Layout del forno; il sinottico

Questa funzione viene realizzata in base al valore assunto da una variabile, nel senso che in base al valore assunta dalla variabile il componente grafico ad essa associato cambia colorazione. La scelta dei colori è dettata dalle regole descritte in capitolo 3 sugli vincoli della progettazione del interfaccia grafica.

Cliccando con il mouse sopra i simboli delle termocoppie, bruciatori, ventilatori, ecc; appare l'indicazione dei valori attivi del dispositivo. Nella parte destra della finestra apparsa, si possono modificare i valori attivi nell'impianto come mostrato in fig:4.19.

È stato inserito anche la funzione di zoom per ridimensionare il sinottico in modo da avere diversi livelli di visualizzazione.

Descrizione	Valori attivi				Modifica		
	Mis	Min	Set	Max	Min	Set	Max
TC03 Temp.tubaz.riciclo sx fumi [°C]	100	100	170	140	100	170	140
TC04 Temp.tubaz.riciclo dx fumi [°C]	150	130	140	160	130	140	160
G1 Ventil. camino fumi [Hz]	+20.0	+25.0		+30.0	+25.0		+30.0
M1 Ventilatore fumi	DIS						
Tempo regolazione ferma [s]			60			60	
Tempo trigger regolazione [s]			60			60	
RP2 Pos.serr.camino fumi [%]	+0.0						

MAN AUTO +35.0 [Hz] START POSIZ.MAN Conferma

Figura 4.19: Layout del forno; i valori attivi del sinottico

## Mappa Carri

Questa pagina rappresenta uno schema della movimentazione dei carri. Alla pagina mappa carri si accede premendo da tutte le pagine la sequenza di tasti funzione [F4 F11] oppure con il mouse selezionare il tasto F4 Scelta Pagina, è poi F11 Mappa Carri. L'immagine in fig:4.20 mostra come si presenta questa pagina.

In questa pagina sono visualizzati i carri dell'impianto con il loro contenuto. Nella parte inferiore dello schermo viene raffigurato un riassuntivo dei materiali presenti sulla linea, in cui ad ogni colore carro viene associato il codice materiale, il codice colore, il numero dei carri e il descrittivo del materiale. Materiali diversi hanno codici diversi e colori diversi. Cliccando con il mouse su un carro, si apre una finestra che permette di visionare immediatamente il contenuto del carro. Tutte queste azioni sono state realizzate con degli script C.

## Editor dei materiali

Alla pagina editor materiali si accede premendo da tutte le pagine la sequenza di tasti funzione [F4 F9] oppure con il mouse selezionare il tasto F4 scelta pagina, è poi F9 Editor Materiali. Questa pagine e riportate nell'immagine in fig:4.21.

Ogni materiale gestito dal sistema di supervisione, deve essere configurato all'interno del database dei materiali. Per ogni materiale devono essere specificare le seguenti caratteristiche, che sono:

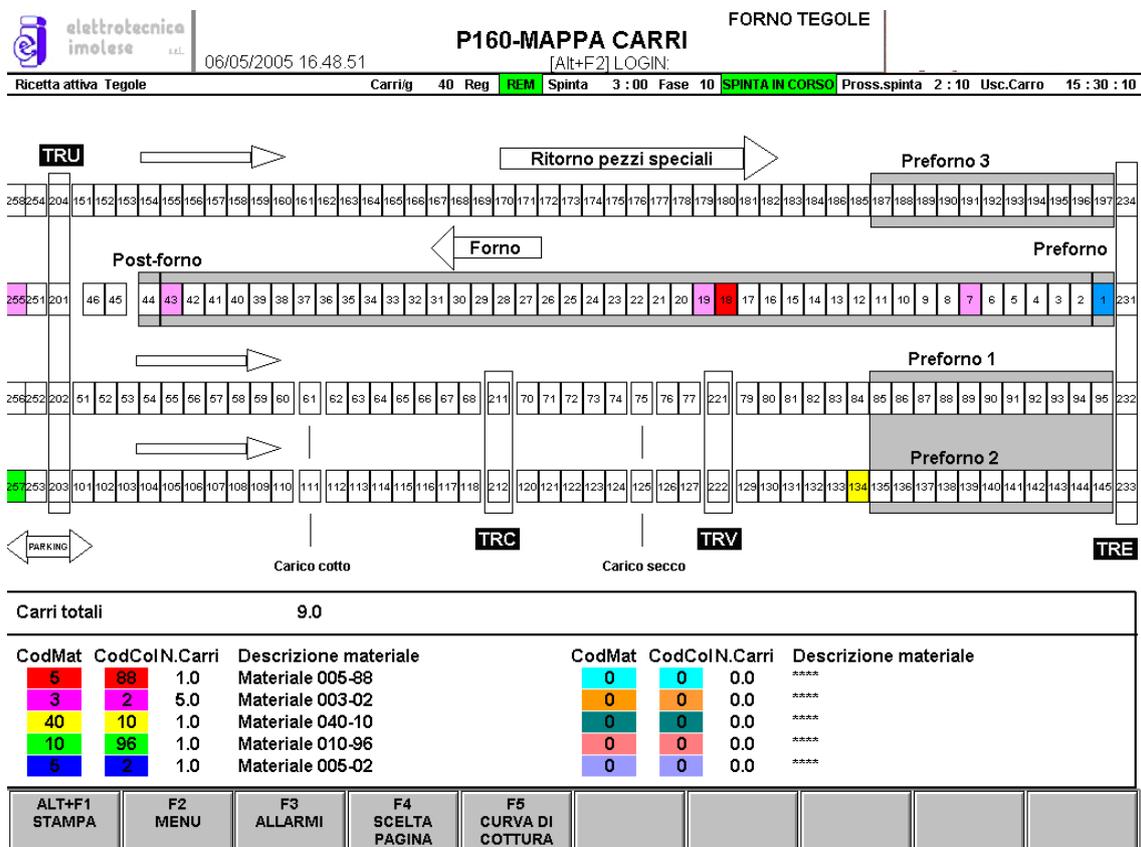


Figura 4.20: Pagina mappa carri

- Descrittore materiale: Codice alfanumerico utile ad identificare un materiale.
- ACGNum: Acronimo con significato di codice commerciale numerico del materiale.
- ACGCod: Acronimo con significato di codice commerciale alfanumerico del materiale.
- Codice materiale: codice numerico di produzione del materiale.
- Codice colore: codice numerico del colore del materiale.
- Peso num.pezzo cotto: peso del pezzo cotto in kg, utile al fini statistici.
- Nun.pezzi per carro: numero di pezzi presenti su un carro, utile a fini statistici.


FORNO TEGOLE  
**P120-EDITOR MATERIALI**  
 06/05/2005 16.58.05 [Alt+F2] LOGIN:

Ricetta attiva pluto2 Carrigi 0 Reg **REM** Spinta 0 : 00 Fase 10 **SPINTA IN CORSO** Pross.spinta 0 : \*\*\* Usc.Carro 15 : 30 : 10

EditMat_Descrittore	EditMat_CodMat	EditMat_CodCol
Materiale 003-01	3	1
Materiale 003-02	3	2
Materiale 005-02	5	2
Materiale 005-88	5	88
Materiale 010-96	10	96
Materiale 040-10	40	10

Descrittore materiale   
 ACGNum   
 Codice ACG   
 Codice materiale   
 Codice colore   
 Peso nom.pezzo cotto [kg]   
 N.pezzi per carro

Completato Rec 1/6 Row 1 Col 1 EditorMateriali



Figura 4.21: Pagina dell'editor materiali

- Tabella materiali: mostra i materiali presenti all'interno del database del sistema di supervisione. Cliccando con il mouse sopra ad un materiale lo si seleziona .

Come si vede dall'immagine premendo il pulsante Nuova viene creata una nuova riga nel database dei materiali crea una nuova riga del database materiali. Tramite il pulsante Elimina viene eliminato il materiale selezionato, infine premendo il pulsante Salva Modifica, vengono salvate le modifiche fatte sul materiale selezionato. Sono state create delle procedure in C per la raccolta delle informazione dal database. Alcuni di questi codici vengono utilizzate nella pagine della mappa carri per visualizzare lo stato della produzione attuale.

### Allarmi e la loro gestione

Infine la pagina degli allarmi, in questa pagina vengono visualizzati tutti i messaggi verso l'operatore e tutti gli allarmi attivi è non riconosciuti. L'immagine in fig:4.22 mostra una porzione della finestra grafica in cui si vedono gli allarmi. Sono stati usati diversi colori per i diversi tipi di allarme.

...	Num	Testo segnalaz.	Data	Ora	Durata	St
1	2	A0002:MAX TEMPERATURA TC2S	05/05/05	01.37.09 PH	0.00.02	-
2	2	A0002:MAX TEMPERATURA TC2S	05/05/05	01.37.19 PH	0.00.12	Ri
3	2	A0002:MAX TEMPERATURA TC2S	05/05/05	01.37.19 PH	0.00.00	+
4	2	A0002:MAX TEMPERATURA TC2S	05/05/05	01.37.21 PH	0.00.02	-
5	2	A0002:MAX TEMPERATURA TC2S	05/05/05	01.37.31 PH	0.00.12	Ri
6	2	A0002:MAX TEMPERATURA TC2S	05/05/05	01.37.31 PH	0.00.00	+
7	2	A0002:MAX TEMPERATURA TC2S	05/05/05	01.37.33 PH	0.00.02	-
8	2	A0002:MAX TEMPERATURA TC2S	05/05/05	01.37.43 PH	0.00.12	Ri
9	2	A0002:MAX TEMPERATURA TC2S	05/05/05	01.37.43 PH	0.00.00	+

Figura 4.22: Pagina degli allarmi

Le segnalazioni di allarme appaiono secondo la seguente codifica:

- ROSSO: Allarme/messaggio attivo e non riconosciuto.
- VERDE: Allarme/messaggio non attivo e non riconosciuto.
- BLU: Allarme/messaggio attivo e riconosciuto.

### Gestione dei dati di produzione

Alla pagina gestione dei dati di produzione si accede premendo da tutte le pagine la sequenza di tasti funzione [F4 F10] oppure con il mouse selezionare il tasto F4 Scelta Pagina, è poi F10 Produzione. Questa pagina è riportata in fig:4.23.

È possibile impostare la cadenza di produzione espressa in carri/giorno, anche nella pagina ricette è possibile impostare la cadenza di produzione, ma dalla ricetta non è possibile attivare un singolo valore alla volta. In questa pagina sono visualizzati i contatori dell'energia, del gas, il contatore di carri usciti, totale e parziale. Alcuni di questi contatori sono resettabili, serve per cancellare il conteggio parziale. Nella tabella a fondo pagina è possibile visualizzare i carri prodotti, ricercati impostando una data di inizio ricerca e una data di fine ricerca.



# Conclusioni

L'ambito di lavoro documentato da questa tesi è *l'automazione*, ovvero l'automazione industriale. In pratica si tratta di automatizzare una parte del processo produttivo dei laterizi.

Con questa tesi si è trattato il sistema di controllo e raccolta dati di un forno utilizzato per la cottura del laterizio. In generale sono impianti di grandi dimensioni che utilizzano migliaia di variabili in cui il loro controllo deve essere fatto in tempo reale. Il funzionamento di tali impianti è molto complesso, nei capitoli 1 sono state descritte le caratteristiche dei forni, il loro principio di funzionamento, ma senza entrare nei dettagli di come effettivamente esso funzioni. Al giorno d'oggi è impensabile il funzionamento di un impianto di questo tipo senza l'utilizzo di dispositivi elettronici (PLC) e sistemi software per il controllo.

La tesi è stata tratta da un progetto realizzato in azienda. È ovvio che esistono diversi punti di vista per la realizzazione di un progetto di questo tipo, quello riportato nell'ultimo capitolo fa parte di una delle possibili soluzioni del problema. Questa attività richiede impegno grande esperienza e responsabilità ed una grande capacità organizzativa da parte del capo progetto per la suddivisione dei compiti. Per la progettazione di questo tipo di sistema sono fondamentali una perfetta integrazione e interoperabilità tra i processi di sviluppo che implementano le specifiche relative alla gestione dei seguenti sottosistemi:

- Impianti tecnologici, come generazione e distribuzione dell'energia, gas, luce, impianti di illuminazione.
- Impianti di produzione quali, presse, estrusori, essiccatoi ed inoltre tutto il sistema di movimentazione e confezionamento.
- Impianti di comunicazione tra sistema di amministrazione o eventualmente tra altre linee produttive.

Il passo successivo necessario a realizzare un sistema di automazione è l'integrazione di tutti questi sottosistemi rendendo possibile una vera e propria

gestione centralizzata del processo produttivo. È una attività che richiede un continuo aggiornamento sui nuovi prodotti hardware-software in modo da essere sempre al passo coi tempi.

L'ambito di applicazione di questo tipo di tecnologia è molto vasto lo dimostra il fatto che i sistemi di controllo PLC oramai sono presenti ovunque, nel campo industriale,civile, persino in quello medico.

## .1 Appendice

## .2 Appendice A

La codifica dei bit nelle reti di telecomunicazione viene utilizzata per ridurre la frequenza di trasmissione. In quanto frequenze troppo alte potrebbero non essere supportate dal canale di comunicazione. Tipiche codifiche usate nell'ambito delle reti di calcolatori sono.

- 1) 4B5B
- 2) NRZ
- 3) NRZI
- 4) Manchester
- 5) MLT-3

In ogni codifica si definisce un intervallo di switch o di campionamento, corrispondente all'intervallo temporale in cui è definito un bit, cioè il tempo necessario a trasmettere un bit. La frequenza di bit è l'inverso di questo intervallo di tempo. Si definisce inoltre una frequenza fondamentale, che è la massima frequenza di switch necessaria al tipo di codifica. A seconda del tipo la frequenza fondamentale può essere la frequenza di bit o un sottomultiplo di essa ed influenza la ampiezza della banda necessaria alla trasmissione di un segnale con un certo bit rate. Queste codifiche si realizzano utilizzando le varie modulazioni di frequenza. La modulazione modifica un parametro della portante in modo che al bit 1 e al bit 0 corrispondano rispettivamente due diverse ampiezze, frequenze o fase della portante come mostrato nell'immagine in fig:24 seguente.

Di seguito si dà una breve descrizione di queste codifiche.

### .2.1 La codifica 4B5B

Questa codifica utilizza una tabella di conversione, fissa, che trasforma ogni sequenza di 4 bit in una 5 bit con la caratteristica di avere al massimo un bit 0 come prefisso e due bit 0 come suffisso così da garantire almeno due transizioni. In questo modo si eliminano i problemi dovuti alle lunghe sequenze di questi bit 0 (al massimo si hanno sequenze di 3 bit 0) con una efficienza dell'80% però rimangono i problemi dovuti alle lunghe sequenze di 1. La tab:25 mostra la conversione dei bit.

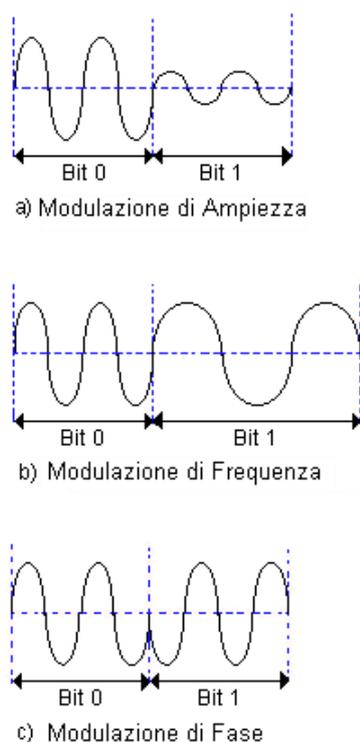


Figura 24: La varie modulazioni della portante

## .2.2 La codifica NRZ

In questo tipo di codifica il livello del segnale é mantenuto costante durante tutto il periodo di switch, in presenza di una sequenza di valori uguali. La codifica *Non Return to Zero* fig:26 prevede due livelli di voltaggio, basso ed alto, corrispondenti ai bit 0 ed 1. Ogni bit occupa un intervallo di switch. Le transizioni avvengono ai confini tra intervalli di switch. Per NRZ esiste il problema della sincronizzazione dovuto alle lunghe sequenze di bit uguali , che viene risolto allungando le sequenze di bit da trasmettere ed inserendo bit aggiuntivi che garantiscono un numero minimo di transizioni regolarmente, tramite l' utilizzo della codifica 4B5B. La codifica NRZ con 5B6B é prevista dallo standard IEEE 802.12 è raggiunge velocità di 100 Mb/s su doppini di rame e fibre ottiche.

Codifica 4B5B

Dato	Codice 4B	Codice 5B
0	0000	11110
1	0001	01001
2	0010	10100
3	0011	10101
4	0100	01010
5	0101	01011
6	0110	01110
7	0111	01111
8	1000	10010
9	1001	10011
A	1010	10110
B	1011	10111
C	1100	11010
D	1101	11011
E	1110	11100
F	1111	11101

Figura 25: Conversione dei bit nella codifica 4B5B

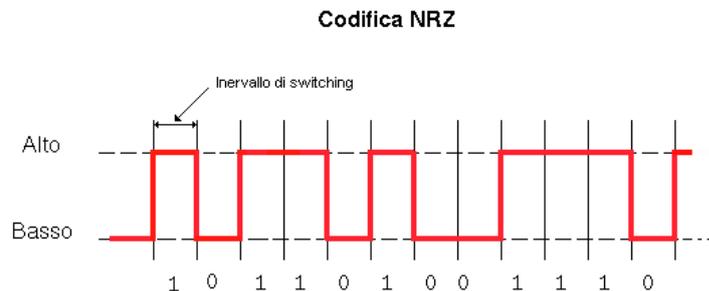


Figura 26: Schema di bit codificati con NRZ

### .2.3 La codifica NRZI

La codifica NRZI ( *Non Return to Zero Inverted on one* )risolve il problema delle lunghe sequenze di bit 1. Il bit 0 é codificato mantenendo costante, per tutto il periodo di *switching*, il livello del segnale al valore dell'ultimo bit trasmesso , in pratica non si ha nessuna transizione per i bit 0. Mentre il bit 1 é codificato come una transizione del livello del segnale a meta' dell'intervallo di switch (da alto a basso oppure viceversa) rispetto al valore dell'ultimo bit trasmesso. Questa codifica utilizzata assieme alla 4B5B permette di risolvere completamente il problema della sincronizzazione dovuto alle lunghe sequenze di bit uguali. É

utilizzata, congiuntamente alla codifica 4B5B (in modo da risolvere entrambi i problemi legati alle sequenze di bit 0 e 1). La codifica NRZI é impiegata per FDDI su fibra ottica.

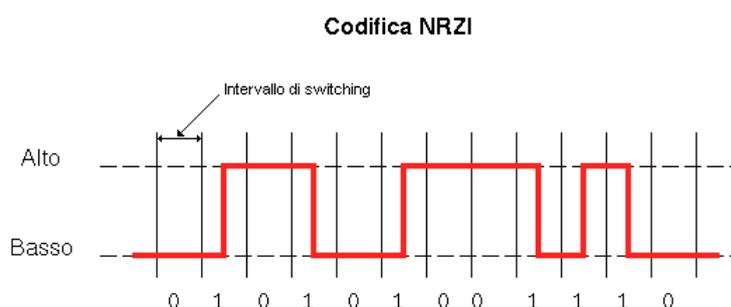


Figura 27: Schema di bit codificati con NRZI

## .2.4 La codifica Manchester

L'idea alla base della codifica *Manchester* é di dividere in due frazioni di uguale durata il periodo di *switch* a trasmettere un bit. In ogni intervallo di switch avviene almeno una transizione, a metà dell'intervallo stesso. La transizione é , da basso ad alto in presenza di bit 0 è da alto a basso in presenza di bit 1 com mostrato in fig:28. In caso di due bit consecutivi differenti vi é una sola transizione per intervallo, se i bit consecutivi sono uguali avviene una ulteriore transizione di ritorno all'inizio dell'intervallo. La codifica di Manchester si chiama anche codifica differenziale. Purtroppo con questo codice si ha una efficienza del 50%, cioè per ogni bit da trasmettere né vengono trasferiti due. In pratica richiede una banda doppia rispetto a quella strettamente necessaria. È utilizzata nelle reti locali *Ethernet* a 10 Mb/s e *Token Ring*.

## .2.5 La codifica MLT-3

La codifica MLT-3 opera su tre livelli anziché su due è prevede una transizione per i bit a 1, a metà dell'intervallo di switch, e nessuna transizione per i bit a 0. Denominando i livelli con 0, 1 e 2 dal basso verso l'alto fig:29 le transizioni successive sono nell'ordine: da 0 a 1, da 1 a 2, da 2 a 1, da 1 a 0. Nel caso di numero massimo di transizioni, con tutti i bit a 1. Come si può immaginare la

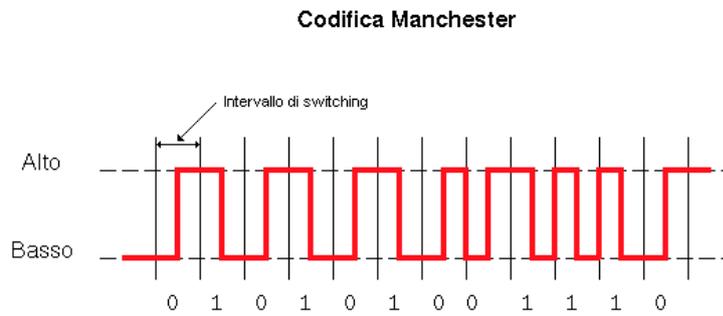


Figura 28: Schema di bit con la codifica Manchester

frequenza fondamentale è un quarto della frequenza di bit. In caso vi fosse una presenza continua di bit a 0, anche in questo caso si impone una ulteriore codifica, tipicamente 4B5B. MLT-3 viene usata da FDDI TP-PMD a 125 Mb/s è da IEEE 802.3 100 BaseTX, su cavi di rame.

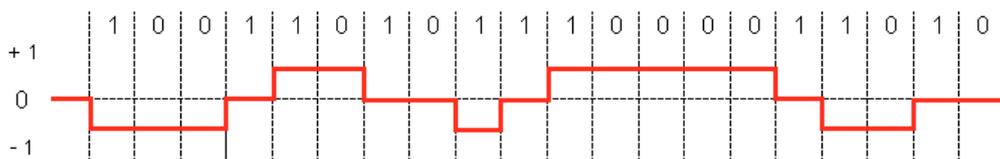


Figura 29: Schema di bit con la codifica MTL-3

## .3 Appendice B

Le modalità di comunicazione o invio dei dati tra due nodi della rete sono due:

- 1) Sincrona
- 2) Assincrona

### .3.1 Modalità Sincrona

Nella trasmissione sincrona i caratteri costituenti un messaggio vengono trasmessi *back-to-back*, con cadenza continua ed in sincronismo con un clock condiviso dal trasmettitore (master) e dal ricevitore (slave). Ciascun blocco di caratteri inizia con una serie di caratteri di sincronizzazione per permettere al ricevitore di sincronizzarsi e mettersi al passo con il trasmettitore. In pratica oltre ai dati viene trasmesso anche un segnale di *clock* esterno. Quindi per effettuare una connessione sincrona sono sufficienti solo tre fili, uno per il *clock* (CK) uno per dati (D) ed uno per la massa (GND). Anche se il filo che porta il segnali di massa può essere eliminato, comunque rimane una soluzione adatta per brevi distanze. La soluzione di portare una linea di clock per lunghe distanze è scarsamente utilizzata poiché non è realizzabile su normali linee telefoniche, presenta inoltre un'alta probabilità di perdere la sincronizzazione. Per ovviare al problema si ricava il clock direttamente dal segnale in arrivo mediante un circuito di aggancio di fase fig:30. Quest'ultima è la soluzione più comunemente utilizzata dato che richiede del hardware piuttosto semplice e offre un buon grado di affidabilità. Una delle tecniche utilizzate per ottenere l'aggancio è il codice bipolare alternato (*Alternate Mark Inversion* AMI). In pratica ricorre alla codifica del bit 1 nel segnale trasmesso alternativamente con impulsi uguali in modulo, ma di polarità opposta. Il codice alternato presenta due inconvenienti: 1) Tutto il sistema va progettato con questo tipo di codifica; 2) Questo codice lascia inalterati gli zeri, cosicché una lunga sequenza di zeri in linea può far perdere la sincronizzazione.

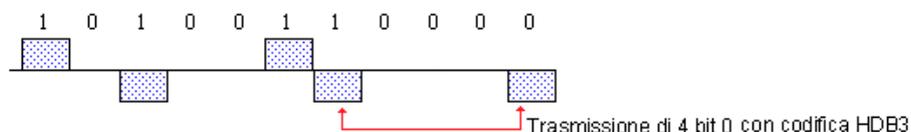


Figura 30: Trasmissione sincrona

Per ovviare a questi inconvenienti si sono introdotti altri codici che si possono classificare in: codici a lunghezza fissa (codici MB-NT) e codici a lunghezza variabile. Fra questi ultimi si collocano il sistema europeo HDB3 e l'americano B8ZS. Il primo deriva dal bipolare alternato con la variante di sostituire le sequenze con quattro bit 0 consecutivi con la sequenza B00V dove V è 0,1 e B è -,0,+ secondo le regole: a) V deve cambiare segno ogni volta che si incontra una nuova sequenza di quattro bit zeri; b) B deve assumere valore tale che V (e solo V) violi la regola dell'alternanza di polarità. Il secondo invece trasmette una sequenza di otto bit 0 consecutivi come 000+-0-+, sequenza che contiene una violazione dell'alternanza (...-0-...) per poter essere identificata con certezza. Con il sistema sincrono si ha una minore ridondanza, e quindi una migliore efficienza di trasmissione, a spese di una maggiore complessità circuitale, che però trova piena giustificazione per le alte velocità di trasmissione consentite.

### 3.2 Modalità Asincrona

Nella modalità asincrona il trasmettitore e il ricevitore si sincronizzano usando i dati stessi. In questa modalità i caratteri vengono trasmessi con cadenza non necessariamente regolare: durante i tempi di inattività la linea viene mantenuta in stato di riposo o *idle*. Il trasmettitore, prima di inviare il primo bit di ciascun byte, deve portare la linea in stato di lavoro o attiva questa commutazione costituisce lo start-bit e ha l'obiettivo di mettere in fase ricevitore e trasmettitore. Il trasmettitore invia inizialmente un bit di *start* fig:31, poi il dato vero e proprio (tipicamente cinque o otto bits, con il bit meno significativo per primo), un bit opzionale di parità, è infine un tempo di *stop* che può avere diverse lunghezze minime (uno o due tempi di switch). Il bit di partenza è di polarità opposta alla condizione normale. Il bit di stop è di polarità normale, e crea uno spazio prima della partenza del byte successivo. La modalità asincrona usa i segnali in modo molto meno efficiente della modalità sincrona, in cui tutti i bit trasportano dati. Il ricevitore, all'arrivo del bit di start, si predispose alla ricezione della sequenza di

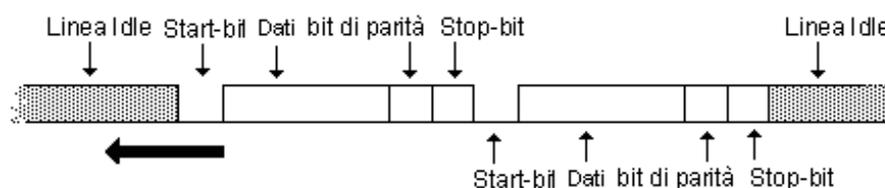


Figura 31: Trasmissione asincrona

bit associato ad un carattere attivando un suo clock interno (che necessariamente deve avere una frequenza di oscillazione nominale uguale a quella del segnale trasmesso). E' sufficiente che il clock del ricevitore mantenga il sincronismo solo per la durata del byte. Trasmesso l'ultimo bit, il trasmettitore riporta la linea in stato di riposo e così la mantiene per un tempo pari alla durata di uno o due bit (stop-bit), a seconda della convenzione adottata. Dopo la segnalazione di stop, se non vi sono altri caratteri da trasmettere, la linea rimane in stato di riposo, altrimenti viene emesso un nuovo start-bit e così di seguito.

La trasmissione asincrona é conveniente quando l'emissione dei caratteri é irregolare come ad esempio nel caso dei terminali a tastiera; lo svantaggio principale sta nello scarso rendimento della trasmissione a causa della ridondanza dovuta all'aggiunta dei bit di start e stop, che incide per un 30% sui bit realmente trasmessi.



## .4 Appendice C

Nell'automazione è in tutti i campi ad essa collegati, il metodo di comunicazione piú utilizzato per acquisire dati ed eseguire azioni è sicuramente rappresentato dalle cosiddette *linee seriali*. Una comunicazione di tipo seriale permette il collegamento di due o piú dispositivi utilizzando un numero limitato di conduttori. Alcuni standard inoltre consentono di raggiungere distanze dell'ordine delle migliaia di metri. Gli standard di comunicazione seriale piú diffusi ed economici sono i seguenti:

- RS232,
- RS422
- RS485

Di seguito viene data una descrizione di questi standard.

### .4.1 Lo standard RS232

Il primo di questi standard di comunicazione è l'RS232, è fig:32 sicuramente il piú diffuso, tanto che è normalmente disponibile di serie su tutti i computer desktop. L'RS 232 venne introdotto nel 1962 dalla American Electronics Industries Association per standardizzare le interfacce fra i terminali dati e i computer. Il metodo di trasferimento dati puó essere sincrono o asincrono, anche se quest'ultimo è di gran lunga il piú usato. L'RS232 consente di effettuare esclusivamente connessioni punto-punto, che quindi mettono in comunicazione due soli dispositivi. Inoltre la massima lunghezza raccomandata del cavo è di 15 metri.

Secondo lo standard EIA fig:33, che descrive le specifiche RS232, i valori di tensione presenti in uscita ed in ingresso hanno un intervallo che va da un minimo di -15V ad un massimo di +15V. Questi livelli di tensione sono incompatibili con i circuiti realizzati con tecnologia TTL è la maggior parte di quelli in

Pin 9 poli	Simbolo EIA	Verso dei segnali	Nome
3	BA	DTE >> DCE	Dati trasmessi
2	BB	DTE << DCE	Dati ricevuti
7	CA	DTE >> DCE	Richiesta trasmissione
8	CB	DTE << DCE	Pronto a trasmettere
6	CC	DTE << DCE	Modem pronto
1	AB	-----	Massa dei segnali
9	CF	DTE << DCE	Rivelazione portante
4	CD	DTE >> DCE	Terminale dati pronto
5	CE	DTE << DCE	Indicatore di chiamata

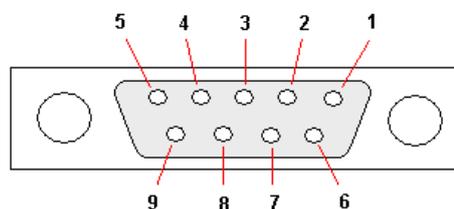


Figura 32: Connettore seriale DB9F e relativi collegamenti sia DTE che DCE

CMOS. Per poter far quindi comunicare un dispositivo RS232 (EIA) con un circuito TTL bisogna adattare i valori di tensione mediante l'utilizzo di adattatori che riportano i livelli di tensione secondo gli standard TTL e CMOS .

Le versioni piú aggiornate denominate RS-422 e RS-485 consentono di raggiungere distanze di migliaia di metri e inoltre consentono un tipo di connessione *multidrop*, quindi con un dispositivo principale che dialoga con un certo numero di dispositivi periferici.



Figura 33: Conversione del segnale nelle diverse famiglie logiche TTL , CMOS... per lo standard RS232

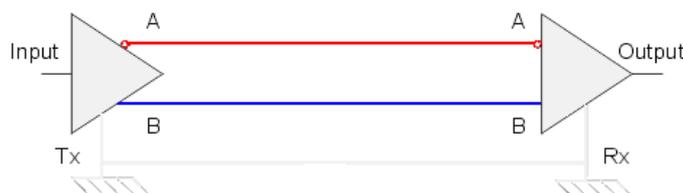


Figura 34: Standard di comunicazione RS422

## .4.2 Lo standard RS422

Questo standard è stato originariamente proposto per la trasmissione di segnali digitali fino a 10 Mbit/s su distanze di circa 1200 m. Usando integrati moderni è inoltre possibile superare i limiti imposti dallo standard sia in termini di velocità che di distanza. Lo standard RS422 prevede che ciascuna linea differenziale sia pilotata da un driver. I ricevitori possono essere fino a 10. L'utilizzo più comune di questo standard è nelle comunicazioni punto-punto, cioè per collegare un singolo trasmettitore ad un singolo ricevitore, come rappresentato nello schema in fig:34.

I due stati di ciascuna linea sono definiti nel seguente modo:

1) Quando il terminale A è negativo rispetto a B, la linea rappresenta un 1 binario. Tale stato rappresenta anche l'assenza di segnale (stato idle)  
 2) Quando il terminale A è positivo rispetto a B, la linea rappresenta uno 0 binario.

Nella immagine in fig:35 viene mostrato l'andamento idealizzato dei segnali sui due fili A (rosso) e B (blu), come si vede si tratta di due segnali tra loro in opposizione di fase. Nell'immagine sono mostrati come variabili tra zero ed una tensione positiva (come del resto avviene il più delle volte anche nei sistemi reali) anche se questo non è richiesto dallo standard. La tensione differenziale è quella che effettivamente trasmette l'informazione ed è positiva o negativa in funzione del livello logico trasmesso.

Da notare che il terminale A ha sempre valori logici opposti al terminale B. All'uscita del trasmettitore la differenza di potenziale tra le linee A e B deve essere di almeno 4 V e la tensione di modo comune deve essere minore di 7 V (normalmente una linea vale 0 V e l'altra circa 5 V). Il ricevitore deve essere in grado di interpretare correttamente lo stato della linea quando la differenza di potenziale è superiore in modulo a 200 mV. Nel caso di realizzazioni industriali, la

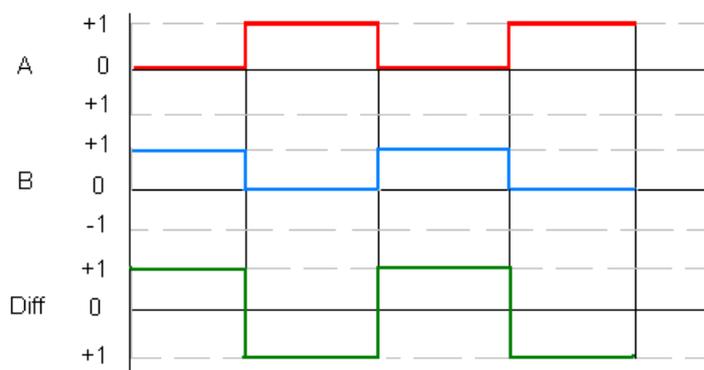


Figura 35: Trasmissione su tensione differenziale nel RS422

topologia più frequente prevede due dispositivi collegati tra loro attraverso due coppie di cavi (oltre la massa), una per ciascun verso di trasmissione del segnale. In questo caso è possibile realizzare una comunicazione *full-duplex*, ciascuno dei due dispositivi può contemporaneamente ricevere e trasmettere dati.

### .4.3 Lo standard RS485

È la tecnologia trasmissiva più utilizzata nelle applicazioni Profibus è il suo impiego include tutti quei casi in cui si rende necessaria una trasmissione ad alta velocità ed implementabile in maniera semplice. Utilizza come mezzo trasmissivo una coppia di conduttori attorcigliati, è necessario schermati. Si tratta quindi di una modalità di trasmissione molto economica e adatta a operare anche in ambienti sfavorevoli. Lo standard RS485 è dal punto di vista elettrico molto simile all'RS422. Anche esso infatti utilizza una trasmissione del segnale in corrente su tensione differenziale di 5V. La differenza sostanziale sta nel supporto delle linee *multi-drop*, cioè linee con la coesistenza di più ricevitori e trasmettitori sulla stessa coppia di fili. Al fine di evitare conflitti è ovviamente necessario che un solo trasmettitore alla volta sia attivo. Questo implica l'uso di trasmettitori che, oltre alle uscite corrispondenti allo zero e uno, possano gestire anche un terzo stato in cui appare come fisicamente non collegata alla linea, chiamato anche stato ad alta impedenza. I ricevitori possono invece essere tutti attivi contemporaneamente ed in genere lo sono effettivamente. La topologia più usata con questo protocollo è quella costituita da una coppia di fili più la massa fig:36 (il classico doppino con schermatura, la quale fa anche da terzo filo per la massa). Questa connessione

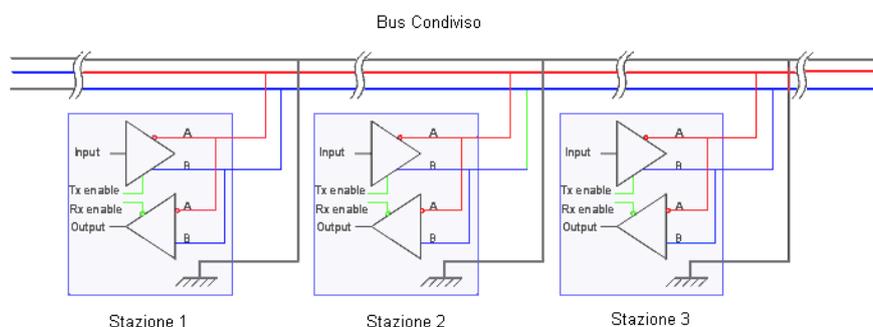


Figura 36: Standard di comunicazione RS485

permette la trasmissione bidirezionale non contemporaneamente *Half Duplex* tra due o più nodi, chè dal punto di vista elettrico, sono tra loro equivalenti.

La sezione rice-trasmittente di ciascuno nodo è evidenziato in figura da un rettangolo blu. Le connessioni verso la linea sono costituite semplicemente dai due terminali A e B comuni sia alla sezione di ricezione che a quella di trasmissione e dalla massa. Ciascun modulo trasmettitore deve possedere, oltre un ingresso dati, anche un ingresso di abilitazione alla trasmissione, pilotato localmente, che permette di disabilitare il trasmettitore quando non serve: al fine di evitare conflitti è necessario prevedere un qualche meccanismo che impedisca l'attivazione contemporanea di più trasmettitori oppure sia in grado rilevare tali conflitti ed intervenire opportunamente. I driver RS485 sono comunque progettati per non riportare danni anche in caso di corto circuito permanente, limitando la corrente massima a 250 mA. Lo standard originario permette la connessione di massimo 32 ricevitori ed il bus deve essere terminato all'inizio e alla fine di ogni segmento da terminazioni (stazioni) attive. L'RS-485, non specifica il connettore da utilizzare, ma si limita a descrivere i poli da usare, è consuetudine l'impiego di connettori 9-pin (DB9) la cui piedinatura è riportata in tabella 37.

Nella comunicazione seriale asincrona (appendice B) i dati viaggiano sulla stessa linea, uno dopo l'altro e non esiste un riferimento temporale comune al trasmettitore e al ricevitore, cosicchè questi devono sincronizzarsi di tanto in tanto, modalità di trasferimento asincrona.

N° pin	Nome segnale	Significato
1*	SHIELD	Protezione EMC
2*	M24V	Tensione di uscita - 24V
3	RxD/TxD-P	Ricezione/trasmissione dati
4*	CNTR-P	Segnale di controllo per ripetitori
5	DGND	Massa di Vp
6**	VP	Tensione positiva +5V
7*	P24V	Tensione di uscita + 24V
8	RxD/TxD-N	Ricezione/trasmissione dati
9*	CNTR-N	Segnale di controllo per ripetitori

\*) segnali opzionali

\*\*\*) segnale necessario solo per le stazioni di fine bus

Figura 37: Connettore RS485

## .5 Appendice C

Fino ad ora si è parlato di sistemi di controllo in generale senza fare alcuna distinzione, ebbene i sistemi di controllo di un qualunque processo produttivo possono in generale essere divisi in due categorie che sono:

- sistemi ad anello chiuso
- sistemi ad anello aperto

I sistemi ad anello chiuso sono sistemi di controllo completi, ossia dotati di capacità di correggere una azione in corso di svolgimento, emulando il comportamento umano. Un sistema ad anello chiuso viene indicato come sistema dotato di retroazione, ossia il segnale che dalla macchina torna al sistema di controllo tramite un sensore. In particolare si compongono in questo modo :

- da un segnale di start o set-point che avvia il processo
- da un organo di potenza che lo esegue
- da un sensore che percepisce l'andamento dell'azione
- da una unità di controllo P.I.D che decide come correggere l'azione in corso sulla base del segnale di ritorno proveniente dal sensore

Lo scheme seguente rappresenta il funzionamento di un sistema di controllo ad anello chiuso.

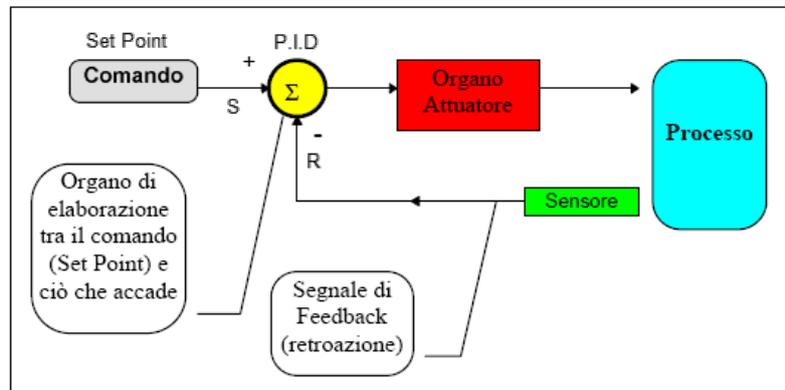


Figura 38: Schema di un sistema di controllo ad anello chiuso

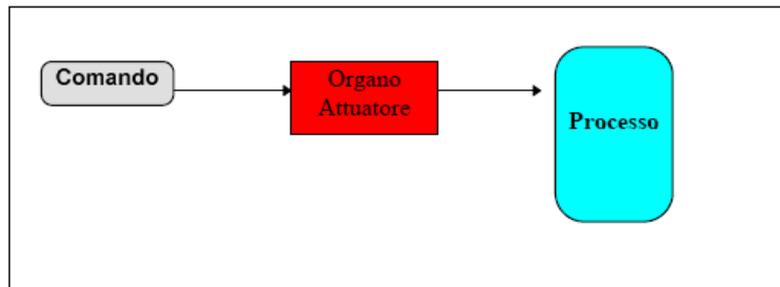


Figura 39: Schema di un sistemi di controllo ad anello aperto

Nei sistemi ad anello aperto il sistema di controllo non è a conoscenza dell'esito della sua azione di comando, in quanto manca il sensore sul processo con il relativo segnale di retroazione. In particolare si compone in questo modo :

1. da un segnale di start o set-point che avvia il processo
2. da un organo di potenza che lo esegue

Mancando il segnale di retroazione non è quindi un sistema di regolazione ma solo di comando.

Il sistema di controllo trattato in questa tesi fa parte della prima categoria in quanto sono previsti dei sensori di temperatura ,pressione ,umidità, posizionamento, contatori ect, che rilevano il valori di processo.

## **.6 Appendice E**

Il seguente elenco si riferisce gli strumenti software impiegati in azienda:

- 1) Lo scada della Seimens - WINCC V6.0
  - C-Editor -( Compilatore linguaggio C)
  - VBS-Ediror -(Editor Visual Basic Script)
  - Graphics Designer -( Editor grafico )
  - Microsoft SQL server 2000 -(DBMS della Microsoft)
  - User Archive - (Software per la gestione del archivio dati)
- 2) Editor per interfacce grafiche HMI - ProTool Pro CS/RT
- 3) Simatic Step 7 - Software Siemens usato per la programmazione di PLC
  - Simatic S7 PLCSIM - (Simulatore di PLC)
  - Simatic NetPro -(Software della Siemens usato per la configurazione di rete ,Profibus DP/PA/ ,MPI ,Profinet)

Strumenti hardware impiegati in azienda:

- 1) PLC Siemens serie Simatic 300
  - CPU 317-2
  - CPU 315-2 DP
- 2) OP -( Pannello operatore OP17)
- 3) Stazione di lavoro PG/PC di produzione-Siemens
- 4) Cavi di connessione Profibus e MPI



# Bibliografia

- [1 ] I laterizi E.Facincani Faenza (Gruppo Editoriale)
- [2 ] SIEMENS SIMATIC Manual Collection  
Url: [http://www.siemens.com/automation/service & support](http://www.siemens.com/automation/service%20&%20support)
- [3 ] IEC 61131-3 (1993-03) Programmable controllers - Part 3: Programming languages, International Electrotechnical Commission (1993).
- [4 ] Sistemi di Basi di Dati 4<sup>a</sup> edizione - Ramez A. Elmasri , Shamkant B.Navathe Addison Pearson 2004
- [5 ] Il linguaggio C 2<sup>a</sup> edizione Kernighan & Ritchie, Milano, Pearson 2004
- [6 ] C La Guida completa 3<sup>a</sup> edizione Herbert Schildt, McGraw-Hill 2003
- [7 ] Visual Basic 6.0 - Guida alla programmazione Julia Case Bradley, Anita C. Millspaugh McGraw-hill
- [10 ] Principi di ingegneria del software 4<sup>a</sup> edizione R.S.Pressman McGraw-Hill 2004



# *Ringraziamenti*

*Giunto al termine di questo lavoro e del mio percorso universitario , desidero ringraziare chi in qualsiasi modo vi abbia contribuito. Desidero innanzitutto ringraziare l'Ing. Massimo Foresti per avermi dato l'opportunità di svolgere il mio stage formativo presso la ditta Elettrotecnica Imolese , consentendomi di apprendere le basi per lo sviluppo di software a livello professionale è del automazione industriale, inoltre di avermi dato la possibilità di sviluppare questa tesi. Desidero ringraziare il Dott. Mirco Andreotti per la sua disponibilità , per avermi consigliate e seguito nelle realizzazione di questa tesi. Un ringraziamento speciale va ai miei familiari per aver creduto in me, per avermi sostenuto moralmente ed economicamente dandomi la possibilità di arrivare alla fine di questo percorso.*

*Desidero ringraziare i miei colleghi di lavoro per avermi consigliato, è viceversa ho chiesto consiglio. Ringrazio inoltre tutti i miei compagni di corso è di studio che mi sono stati di grande aiuto con i quali ho trascorso dei bei momenti. Inoltre ringrazio gli amici di sempre per il loro aiuto è sostegno.*

*Alfred Simoni*

*Ferrara , novembre , 2006*