

### Politecnico di Torino

Dipartimento di Ingegneria Meccanica e Aerospaziale - DIMEAS Laurea Magistrale in Ingegneria Meccanica

## Automazione di un banco elettropneumatico con controllo mediante PLC e HMI

Relatore: Prof. Terenziano Raparelli

Correlatori: Prof. Luigi Mazza Ing. Giuseppe Pepe Candidato: Michele Calò

## Indice

1	L'a	utomazione industriale e introduzione alle reti per l'automazione	1
	1.1	Introduzione all'automazione industriale	1
	1.2	Un cenno alla storia: l'evoluzione dell'automazione industriale	1
	1.3	Composizione di un sistema di automazione	3
	1.4	Architettura del CIM	5
	1.5	Reti di comunicazione per l'automazione ed esempi di reti di campo	6
	1.6	Il modello di riferimento di rete ISO/OSI	9
	1.7	Metodi di accesso per le reti standard e integrazione tra reti $\ldots \ldots \ldots \ldots$	10
<b>2</b>	Il P	LC: il controllore a logica programmabile	13
	2.1	L'evoluzione dei sistemi di controllo	13
	2.2	Definizione, vantaggi e classificazione dei PLC	14
	2.3	Il funzionamento dei PLC	16
	2.4	L'architettura hardware del PLC	17
		2.4.1 PLC: Modulo processore	18
		2.4.2 PLC: Memoria	20
		2.4.3 PLC: Modulo di alimentazione	21
		2.4.4 PLC: Moduli di Ingresso/Uscita	21
		2.4.5 PLC: Terminale di programmazione	22
		2.4.6 PLC: Moduli speciali	23
	2.5	L'architettura software del PLC	23
		2.5.1 PLC: Sistema operativo	24
		2.5.2 PLC: Ciclo real time	24
	2.6	PLC: linguaggi di programmazione	26
		2.6.1 Il linguaggio SFC	26
		2.6.2 Il linguaggio a contatti	27
		2.6.3 Il linguaggio a blocchi funzionali	29
		2.6.4 La lista di istruzioni	30
		2.6.5 Il testo strutturato	31
3	SIN	IATIC STEP 7: TIA Portal	33
	3.1	SIMATIC STEP 7	33
	3.2	Interfaccia utente TIA Portal V16	33
	3.3	Gestione dei progetti sul TIA Portal	35
	3.4	Configurazione di un nuovo dispositivo	37
		3.4.1 Collegamento e riconoscimento dei dispositivi fisici	42
		3.4.2 L'utilizzo del PLCSIM Standard	49
		3.4.3 PLCSIM Standard VS PLCSIM Advanced	49
	3.5	Scrivere un programma per PLC	52

		3.5.1 Blocchi Organizzativi	. 53
		3.5.2 Blocchi Funzionali	. 54
		3.5.3 Blocchi Funzione	. 55
		3.5.4 Blocchi Dati	. 55
	_		
4	Ban	co didattico Pneumax: componenti e struttura	57
	4.1	L'idea di un banco didattico	. 57
	4.2	Banco didattico Pneumax: parte fissa	. 59
		4.2.1 Parte fissa: Alimentatore	. 60
		4.2.2 Parte fissa: Morsettiera	. 61
		4.2.3 Parte fissa: PLC	. 61
		4.2.4 Parte fissa: HMI	. 64
		4.2.5 Parte fissa: Pulsantiera	. 65
		4.2.6 Parte fissa: Cassette per gli input/output	. 66
		4.2.7 Parte fissa: Schema elettrico	. 66
	4.3	Banco didattico Pneumax: parte mobile, PANNELLO 1	. 68
		4.3.1 Pannello 1: Elettrovalvole Optyma32-S	. 68
		4.3.2 Pannello 1: Cilindri a doppio effetto	. 75
		4.3.3 Pannello 1: Filtro regolatore e valvola di intercettazione	. 75
	4.4	Banco didattico Pneumax: parte mobile, PANNELLO 2	. 77
		4.4.1 Pannello 2: Elettrovalvole Optyma 32-S e Modulo I/O $\ \ldots \ldots \ldots \ldots$	. 77
		4.4.2 Pannello 2: Cilindro a doppio effetto	. 80
		4.4.3 Pannello 2: Trasduttori pneumo-elettrici	. 81
	4.5	Banco didattico Pneumax: parte mobile, PANNELLO 3	. 82
		4.5.1 Pannello 3: Elettrovalvole pneumatiche	. 83
		4.5.2 Pannello 3: Regolatore proporzionale a controllo elettronico $\ldots \ldots \ldots$	. 84
		4.5.3 Pannello 3: Trasduttore di pressione	. 86
-			00
5		mblaggio e programmazione della parte mobile	89
	5.1		00
	F 0	Introduzione alla parte sperimentale	. 89
	5.2	Introduzione alla parte sperimentale	. 89 . 90
	5.2	Introduzione alla parte sperimentale	. 89 . 90 . 91
	5.2	Introduzione alla parte sperimentale	. 89 . 90 . 91 . 93
	5.2	Introduzione alla parte sperimentale	. 89 . 90 . 91 . 93 . 96
	5.2 5.3	Introduzione alla parte sperimentale	. 89 . 90 . 91 . 93 . 96 . 112
	5.2 5.3	Introduzione alla parte sperimentale	. 89 . 90 . 91 . 93 . 96 . 112 . 113
	5.2 5.3	Introduzione alla parte sperimentale	. 89 . 90 . 91 . 93 . 96 . 112 . 113 . 115
	5.2 5.3	Introduzione alla parte sperimentale	. 89 . 90 . 91 . 93 . 96 . 112 . 113 . 115 . 119
6	5.2 5.3	Introduzione alla parte sperimentale	. 89 . 90 . 91 . 93 . 96 . 112 . 113 . 115 . 119
6	5.2 5.3 Cor 6.1	Introduzione alla parte sperimentale	. 89 . 90 . 91 . 93 . 96 . 112 . 113 . 115 . 119 <b>125</b> . 125
6	5.2 5.3 Cor 6.1 6.2	Introduzione alla parte sperimentale	$\begin{array}{cccccccccccccccccccccccccccccccccccc$
6	5.2 5.3 Con 6.1 6.2 6.3	Introduzione alla parte sperimentale	$\begin{array}{cccccccccccccccccccccccccccccccccccc$
6	5.2 5.3 Con 6.1 6.2 6.3 6.4	Introduzione alla parte sperimentale	$\begin{array}{cccccccccccccccccccccccccccccccccccc$
6	5.2 5.3 Cor 6.1 6.2 6.3 6.4 6.5	Introduzione alla parte sperimentale	$\begin{array}{cccccccccccccccccccccccccccccccccccc$
6	5.2 5.3 6.1 6.2 6.3 6.4 6.5 6.6	Introduzione alla parte sperimentale	$\begin{array}{cccccccccccccccccccccccccccccccccccc$
6	5.2 5.3 5.3 Cor 6.1 6.2 6.3 6.4 6.5 6.6 6.7	Introduzione alla parte sperimentale	$\begin{array}{cccccccccccccccccccccccccccccccccccc$
6	5.2 5.3 <b>Cor</b> 6.1 6.2 6.3 6.4 6.5 6.6 6.7 6.8	Introduzione alla parte sperimentale	$\begin{array}{cccccccccccccccccccccccccccccccccccc$
6	5.2 5.3 <b>Cor</b> 6.1 6.2 6.3 6.4 6.5 6.6 6.7 6.8 6.9	Introduzione alla parte sperimentale	$\begin{array}{cccccccccccccccccccccccccccccccccccc$
6	5.2 5.3 5.3 Con 6.1 6.2 6.3 6.4 6.5 6.6 6.7 6.8 6.9	Introduzione alla parte sperimentale	$\begin{array}{cccccccccccccccccccccccccccccccccccc$

	6.9.2	Confronto Feedback al variare del $K_p$	167
	6.9.3	Confronto Feedback al variare del $K_i$	172
	6.9.4	Confronto Feedback al variare del $K_d$	176
	6.9.5	Metodo di Ziegler-Nichols	180
	6.9.6	Ottimizzazione attraverso il blocco PID	185
	6.9.7	Risposta del sistema ad un disturbo esterno	189
6.10	Progra	mmazione del HMI del pannello 3	190

#### 7 Conclusioni e sviluppi futuri

1	9	3
---	---	---

# Elenco delle figure

1.1	Regolatore centrifugo di J. Watt [3]	2
1.2	Classificazione dei differenti sistemi di automazione [5]	4
1.3	Schema di un sistema Computer Integrated Manufacturing [6]	5
1.4	Schema a piramide del CIM [9]	5
1.5	Esempio di rete di campo con più unità Master che controllano le unità Slave [6]	$\overline{7}$
1.6	Esempio di rete di campo con utilizzo di dispositivi intelligenti [10]	8
1.7	Schema di rete con accesso Master/Slave [14]	10
1.8	Schema Profibus: metodo di accesso Token Bus [15]	11
1.9	Schema di un collegamento tra reti [16]	11
2.1	Evoluzione dei sistemi di controllo	14
2.2	Siemens: varie tipologie di PLC [18]	16
2.3	Schema semplificativo di come funziona un PLC [19]	16
2.4	Anello chiuso di controllo tra PLC e macchina	17
2.5	Architettura hardware di un PLC $[19]$	18
2.6	Disegno schematico della struttura di un PLC	18
2.7	Modulo processore Siemens [20]	19
2.8	Modulo di alimentazione Siemens [20]	21
2.9	Modalità di esecuzione del PLC a copia massiva di ingressi e uscite	25
2.10	Esempio di sequenze in parallelo mediante SFC [23]	27
2.11	Esempio di programmazione mediante il linguaggio SFC	28
2.12	Esempio di programmazione mediante il linguaggio SFC di un reattore chimico [10]	28
2.13	Elementi principali utilizzati nella programmazione mediante Ladder Diagram	29
2.14	Esempio di programmazione mediante linguaggio a schema funzionale	30
2.15	Esempio di programmazione mediante linguaggio AWL	30
2.16	Esempio di programmazione mediante testo strutturato	31
3.1	Vista portale presente nel TIA Portal	34
3.2	Vista progetto presente nel TIA Portal	34
3.3	Creazione di un nuovo progetto sul TIA Portal su vista portale	35
3.4	Creazione di un nuovo progetto sul TIA Portal su vista progetto	35
3.5	Salvataggio di un progetto sul TIA Portal su vista progetto	36
3.6	Archiviazione/Disarchiviazione di un progetto sul TIA Portal su vista progetto .	36
3.7	Apertura, chiusura e eliminazione di un progetto sul TIA Portal su vista progetto	36
3.8	Apertura di un progetto non editabile sul TIA Portal su vista progetto	37
3.9	Procedura per visualizzare la guida del TIA Portal su vista progetto	37
3.10	Primi passi da eseguire con il TIA Portal	38
3.11	Configurazione di un nuovo controllore	38
3.12	Vista del PLC dalla quale è possibile carpire informazioni utili alla configurazione	39
3.13	Selezione e inserimento della CPU nel progetto sul TIA Portal	39

3.14	Selezione e inserimento della "CPU non specificata" nel progetto sul TIA Portal .	40
3.15	Configurazione di un nuovo HMI	40
3.16	Retro del HMI dove è possibile carpire informazioni utili alla configurazione	41
3.17	Selezione e inserimento del HMI nel progetto sul TIA Portal	41
3.18	Personalizzazione del pannello operatore configurato sul TIA Portal	42
3.19	Visualizzazione degli indirizzi IP dei dispositivi configurati	42
3.20	Vista dispositivi e elenco proprietà dei dispositivi	43
3.21	Modifica dell'indirizzo IP di un dispositivo	43
3.22	Procedura per permettere il collegamento mediante cavo ETHERNET, fase 1	44
3.23	Procedura per permettere il collegamento mediante cavo ETHERNET, fase 2	44
3.24	Procedura per permettere il collegamento mediante cavo ETHERNET, fase 3	44
3.25	Procedura per permettere il collegamento mediante cavo ETHERNET, fase 4	45
3.26	Collegamento tra PC e PLC fisico	45
3.27	Compilazione del programma utente presente sul TIA Portal	46
3.28	Caricamento del progetto presente sul TIA Portal nel dispositivo. 1	47
3.29	Caricamento del progetto presente sul TIA Portal nel dispositivo, 2	47
3.30	Collegamento virtuale in rete sul TIA Portal	48
3.31	Collegamento virtuale HMI sul TIA Portal	48
3.32	Avviamento della simulazione del PLC sul TIA Portal 1	49
3 33	Avviamento della simulazione del PLC sul TIA Portal 2	49
3.34	Avviamento della simulazione del PLC sul TIA Portal 3	50
3 35	PLCSIM Standard tabelle di simulazione e sequenze di segnali	50
3.36	Schermata PLCSIM Standard	51
3.37	Simulazione e collegamenti mediante PLCSIM Advanced	51
3.38	Inserimento di nuovi blocchi di programma sul TIA Portal	53
3.39	Esempio di programmazione mediante i "preferiti" sul TIA Portal	53
3 40	Bappresentazione dei vari accessi ai blocchi dati sul TIA Portal	55
0.10		00
4.1	Tipologia di componenti forniti dalla Pneumax S.p.a. [30]	57
4.2	Banco didattico Pneumax al suo arrivo	58
4.3	Struttura base del banco didattico Pneumax	59
4.4	Banco didattico Pneumax: parte fissa	60
4.5	Banco didattico Pneumax, parte fissa: alimentatore	61
4.6	Banco didattico Pneumax, parte fissa: morsettiera	61
4.7	Banco didattico Pneumax, parte fissa: PLC, Scalance e Modulo Output analogici	62
4.8	Schema di un controllore S7-1200, Siemens AG	62
4.9	Schema elettrico della CPU 1214C DC/DC/DC, Siemens	63
4.10	Banco didattico Pneumax, parte fissa: HMI	64
4.11	Schema di un HMI, modello KTP700 Basic	65
4.12	Banco didattico Pneumax, parte fissa: Pulsantiera	65
4.13	Vista frontale della cassetta per gli input e schema 2D	66
4.14	Cablaggio interno presente nella cassetta degli input e schema 2D	66
4.15	Banco didattico Pneumax, parte fissa: Schema elettrico	67
4.16	Banco didattico Pneumax, parte mobile, pannello 1	68
4.17	Pannello 1: elettrovalvole Optyma32-S	69
4.18	Descrizione delle elettrovalvole Optyma32-S	70
4.19	Circuito integrato con terminale a connessione multipolare	71
4.20	Corrispondenza PIN per batteria di elettrovalvole montate tutte su doppie basi	
	per bistabile	71

4.21	Corrispondenza PIN per batteria di elettrovalvole monostabili montate tutte su	
	doppie basi per monostabile	72
4.22	Corrispondenza PIN per batteria di elettrovalvole montate su basi in configura-	
	zione mista	72
4.23	Connettore femmina SUB-D 25 poli	72
4.24	Elettrovalvola bistabile di tipo Solenoide-Solenoide [30]	73
4.25	Elettrovalvola bistabile di tipo Solenoide-Solenoide (5/3 centri chiusi) [30] $\ldots$	74
4.26	Elettrovalvola bistabile di tipo Solenoide-Molla [30]	74
4.27	Pannello 1: Cilindro a doppio effetto	75
4.28 4.29	Pannello 1: batteria composta da filtro regolatore e valvola di intercettazione Sulla sinistra vengono riportate le curve di portata del filtro regolatore, mentre	76
	sulla destra le caratteristiche di regolazione dello stesso	77
4.30	Banco didattico Pneumax, parte mobile, pannello 2	78
4.31	Pannello 2: elettrovalvole Optyma 32-S con l'aggiunta del modulo I/O	78
4.32	Schema del modulo I/O, con particolare del connettore M8-3 poli [30]	79
4.33	Corrispondenza PIN per batteria di elettroval vole con l'aggiunta di moduli $I/O$ .	80
4.34	Pannello 2: Cilindro a doppio effetto	81
4.35	Pannello 2: trasduttori pneumo-elettrici	82
4.36	Banco didattico Pneumax, parte mobile, pannello 3	82
4.37	Sulla sinistra l'elettroval vola pneumatica MATRIX $2/2~{\rm NC}$ e a destra un suo	
	spaceato $([32])$	83
4.38	Simbolo identificativo dell'elettroval vola $2/2$ NC	83
4.39	Sulla sinistra il regolatore proporzionale a controllo elettronico e sulla destra un	
	suo schema funzionale ([30]) $\ldots$ $\ldots$ $\ldots$ $\ldots$ $\ldots$ $\ldots$ $\ldots$ $\ldots$ $\ldots$	84
4.40	Loop chiuso del regolatore proporzionale a controllo elettronico [30]	85
4.41	Corrispondenza pin connettore con regolatore proporzionale [30]	85
4.42	Trasduttore di pressione utilizzato per il pannello 3	86
4.43	Caratteristica statica del trasduttore sulla sinistra e la tabella che riporta le	
	misurazioni fatte per ricavarla sulla destra	87
5.1	Grafcet	90
5.2	Grafcet contratto	90
5.3	Foto banco didattico Pneumax con pannello 1	91
5.4	Schema elettrico e pneumatico del banco con il pannello 1	92
5.5	Tabella delle variabili presente sul Tia Portal relativa al pannello 1	93
5.6	Inserimento del Blocco Automatismo all'interno del MAIN	96
5.7	Tabella delle variabili del DB Dati HMI relativa al pannello 1	96
5.8	Adattamento del programma nel blocco funzione per il pannello operatore	97
5.9	Adattamento del programma nel MAIN per HMI del pannello 1	97
5.10	Aggiunta della funzione monitoraggio nel MAIN per HMI del pannello 1	97
5.11	Procedura per aggiungere una nuova pagina all'HMI	100
5.12	Procedura per associare una nuova pagina ad un tasto dell'HMI	101
5.13	Anteprima sul TIA Portal della pagina dedicata al pannello 1 del banco didattico	101
5.14	Procedura per selezionare la voce "Biblioteche" dalla pagina principale	102
5.15	Inserimento del pulsante di emergenza e del tasto monostabile START	102
5.16	Vista globale dei tre pulsanti aggiunti sulla "Pagina Comandi"	103
5.17	Inserimento di una base rettangolare per i tre pulsanti	103
5.18	Procedura per associare un pulsante monostabile ad una variabile HMI	104
5.19	Procedura per associare un pulsante bistabile ad una variabile HMI	104
5.20	Vista globale dei led di monitoraggio del sistema sulla "Pagina Comandi"	105

5.21	Vista globale dell'animazione degli steli dei tre cilindri sulla "Pagina Comandi" .	105
5.22	Procedura per poter selezionare all'interno di una pagina HMI delle grafiche pre-	
	definite	106
5.23	Inserimento di uno stelo nella "Pagina Comandi"	106
5.24	Settaggio della posizione e delle dimensioni sulla "Pagina Comandi" del primo stelo	107
5.25	Impostazione dell'animazione e visibilità del primo stelo sulla "Pagina Comandi"	107
5.26	Inserimento di un secondo stelo nella "Pagina Comandi"	107
5.27	Settaggio della posizione e delle dimensioni sulla "Pagina Comandi" del secondo	
	stelo	108
5.28	Impostazione dell'animazione e visibilità del secondo stelo sulla "Pagina Comandi"	108
5.29	Inserimento di un terzo stelo nella "Pagina Comandi"	108
5.30	Impostazione dell'animazione e visibilità del terzo stelo sulla "Pagina Comandi".	109
5.31	Settaggio della variabile virtuale "moto 2 ritorno" per garantire il movimento	
	orizzontale dello stelo	109
5.32	Impostazione del movimento orizzontale sulla "Pagina Comandi" del terzo stelo.	110
5.33	Inserimento di un quarto stelo nella "Pagina Comandi"	110
5.34	Impostazione dell'animazione e visibilità del quarto stelo sulla "Pagina Comandi"	110
5.35	Settaggio della variabile virtuale "moto 2" per garantire il movimento orizzontale	
	dello stelo	111
5.36	Impostazione del movimento orizzontale sulla "Pagina Comandi" del quarto stelo	111
5.37	Inserimento del cilindro nella "Pagina Comandi"	112
5.38	Foto banco didattico Pneumax con pannello 2	112
5.39	Schema elettrico e pneumatico del banco con il pannello 2	114
5.40	Tabella delle variabili presente sul Tia Portal relativa al pannello 2	115
5.41	Aggiunta dei due blocchi funzione all'interno del MAIN per il pannello 2	115
5.42	Tabella delle variabili del DB Dati HMI relativa al pannello 2	119
5.43	Adattamento del programma nel blocco funzione Automatismo Temporizzato per	
	HMI	119
5.44	Adattamento del programma nel blocco funzione Automatismo Con Contatore	
	per HMI	120
5.45	Adattamento del programma nel MAIN per HMI del pannello 2	120
5.46	Aggiunta della funzione monitoraggio nel MAIN per HMI del pannello 2	120
5.47	Anteprima sul TIA Portal della pagina dedicata al pannello 2 del banco didattico	123
5.48	Attivazione del merker di clock di un PLC	123
5.49	Inserimento dell'elemento "Campo I/O" nella "Pagina dei comandi" $\ \ . \ . \ .$	124
6.1	Foto banco didattico Pneumax con pannello 3	125
6.2	Schema elettrico e pneumatico del banco con il pannello 3	127
6.3	Generazione di un'onda quadra, programma 1	128
6.4	Esempi di diverse onde quadre utilizzando il primo codice tentativo	128
6.5	Generazione di un'onda quadra, programma 2	129
6.6	Modulazione Onda quadra	130
6.7	Esempi di diverse onde quadre al variare del "Duty Cycle"	131
6.8	Esempi di diverse onde quadre al variare del "Duty Cycle", programma finale	131
6.9	Attivazione del generatore di impulsi "PTO1/PWM1"	135
6.10	Parametrizzazione del generatore di impulsi "PTO1/PWM1"	136
6.11	Uscite hardware del generatore di impulsi "PTO1/PWM1"	136
6.12	Indirizzi di I/O del generatore di impulsi "PTO1/PWM1"	137
6.13	Errore di protezione nel caricamento del programma nel PLC	137
6.14	Schema a blocchi del controllo pressione senza controllore	139

6.15	Variazione del "Duty Cycle" nel codice relativo al controllo pressione classico	146
6.16	Variazione del "Duty Cycle" nel codice relativo al controllo pressione rapido	146
6.17	Setpoint, Feedback e Modulazione della valvola di alimentazione con "Valore	
	Range" pari a 3	149
6.18	Setpoint, Feedback e Modulazione della valvola di alimentazione con "Valore	
	Range" pari a 1	150
6.19	Setpoint, Feedback e Modulazione della valvola di alimentazione con "Valore	
	Range" pari a 0.5	150
6.20	Confronto dei tre Feedback con "Valore Range" diversi, f=1 Hz $\ldots$	151
6.21	Confronto dei tre Feedback con "Valore Range" diversi, f=10 Hz $\ldots \ldots \ldots$	151
6.22	Confronto tra frequenze diverse a parità del "Valore Range"	152
6.23	Richiamo del PID_Compact nel Cyclic interrupt	154
6.24	Impostazioni base del PID_Compact nella Vista Funzionale	154
6.25	Selezione dei parametri di ingresso e uscita del PID_Compact	155
6.26	Selezione dei limiti del Feedback del PID_Compact	155
6.27	Selezione dei limiti dell'Output del PID_Compact	156
6.28	Selezione dei parametri dell'algoritmo di calcolo del PID_Compact	156
6.29	Setpoint, feedback, modulazione della valvola di alimentazione e scarico ad una	
	frequenza di lavoro pari a 1 Hz	162
6.30	Setpoint, feedback, modulazione della valvola di alimentazione e scarico ad una	
	frequenza di lavoro pari a 2 Hz	163
6.31	Setpoint, feedback, modulazione della valvola di alimentazione e scarico ad una	
	frequenza di lavoro pari a 10 Hz	163
6.32	Ingrandimento del Setpoint e feedback alla frequenza di lavoro pari a 10 Hz $\ .\ .\ .$	164
6.33	Setpoint, feedback, modulazione della valvola di alimentazione e scarico ad una	
	frequenza di lavoro pari a 20 Hz	164
6.34	Ingrandimento del Setpoint e feedback alla frequenza di lavoro pari a 20 Hz $\ldots$	165
6.35	Setpoint, feedback, modulazione della valvola di alimentazione e scaric o ad una	
	frequenza di lavoro pari a 50 Hz	165
6.36	Setpoint, feedback, modulazione della valvola di alimentazione e scarico ad una	
	frequenza di lavoro pari a 100 Hz	166
6.37	Ingrandimento del setpoint e feedback alla frequenza di lavoro pari a 50 Hz	
	$(sinistra) e 100 Hz (destra) \dots \dots$	166
6.38	Confronto tra feedback al variare della frequenza di lavoro	167
6.39	Setpoint e feedback, errore non compensato e errore compensato dal controllore	1.00
<u> </u>	$P \operatorname{con} K_p = 0.1 \dots \dots$	168
6.40	Setpoint e feedback, errore non compensato e errore compensato dal controllore	1.00
0.11	$P \operatorname{con} K_p = 1$	168
6.41	Setpoint e feedback, errore non compensato e errore compensato dal controllore	1.00
a 10	$P \operatorname{con} K_p = 2 \dots \dots$	169
6.42	Setpoint e feedback, errore non compensato e errore compensato dal controllore	1.00
<u> </u>	$P \operatorname{con} K_p = 5$	169
6.43	Zoom dell'errore non compensato e dell'errore compensato dal controllore P con	
o	$K_p=0$	170
6.44	Setpoint e feedback, errore non compensato e errore compensato dal controllore	
0.45	$ \begin{array}{c} P \operatorname{con} K_{p} = 10 \\ G \operatorname{con} K_{p} = 10 \end{array} $	170
<b>b</b> .45	Setpoint e feedback, errore non compensato e errore compensato dal controllore DID $z_{inv} K = 100$	1 /7 1
0.40	$PID \operatorname{con} K_p = 100 \dots \dots$	171
0.46	Confronto tra feedback al variare del guadagno proporzionale $K_p$	172

6.47	Setpoint e feedback, errore non compensato e errore compensato dal controllore	
	$PI \text{ con } K_p = 1 \text{ e } T_I = 0.01 \text{s} \dots \dots$	173
6.48	Setpoint e feedback, errore non compensato e errore compensato dal controllore	
0.40	$PI \operatorname{con} K_p = 1 e T_I = 0.1s \dots \dots$	173
6.49	Setpoint e feedback, errore non compensato e errore compensato dal controllore	
0 50	$PI \operatorname{con} K_p = 1 e I_I = 1s \dots $	174
6.50	Setpoint e feedback, errore non compensato e errore compensato dal controllore	1 17 4
6 51	P1 con $K_p = 1$ e $T_I = 10s$	174
0.51	Setpoint e feedback, errore non compensato e errore compensato dal controllore	175
6 59	PI con $K_p=1$ e $I_I=1008$	175
0.02	Confronto tra feedback al variare del coefficiente integrativo $K_i$	175
0.00	Setpoint e recuback, errore non compensato e errore compensato dal contronore $PL con K = 1.0 T_{\rm p} = 0.01c$	176
6 54	Satpoint a feedback errore non compensato a errore compensato del controllore	170
0.04	PL con K $-1 \text{ e } T_{\rm D} = 0.1\text{ s}$	177
6 55	Set point e feedback errore non compensato e errore compensato dal controllore	111
0.00	PI con $K_{\rm p}=1$ e $T_{\rm p}=1$ s	177
6.56	Setpoint e feedback, errore non compensato e errore compensato dal controllore	111
0.00	PI con $K_p=1$ e $T_D=10s$	178
6.57	Setpoint e feedback, errore non compensato e errore compensato dal controllore	
	PI con $K_p=1$ e $T_D=20s$	178
6.58	Confronto tra i feedback determinati da un controllore PD, con e senza filtro nella	
	parte derivativa	179
6.59	Confronto tra feedback al variare del coefficiente derivativo $K_d \ \ \ldots \ \ldots \ \ldots$	180
6.60	Ricerca del guadagno critico $K_{p0}$ e del suo periodo $T_0$	181
6.61	Identificazione del guadagno critico $K_{p0}$ e del suo periodo $T_0$	182
6.62	Risposta del sistema con i parametri P ottimizzati dal metodo di Ziegler-Nichols	182
6.63	Parametri PI ottimizzati dal metodo di Ziegler-Nichols	183
6.64	Risposta del sistema con i parametri PI ottimizzati dal metodo di Ziegler-Nichols	183
6.65	Parametri PID ottimizzati dal metodo di Ziegler-Nichols	184
6.66	Risposta del sistema con i parametri PID ottimizzati dal metodo di Ziegler-Nichols	184
6.67	Confronto tra le risposte ottenute ottimizzando con il metodo di Ziegler-Nichols .	185
6.68	Ottimizzazione dei parametri PID per un controllore di tipo PI	186
6.69	Ottimizzazione dei parametri PID per un controllore di tipo PI	186
6.70	Risposta del sistema con i parametri PI ottimizzati dal TIA Portal	187
6.71	Risposta del sistema con i parametri PID ottimizzati dal TIA Portal	187
6.72	Confronto tra le risposte ottenute ottimizzando con il TIA Portal	188
6.73	Confronto tra le risposte ottenute ottimizzando con il metodo di Zielger-Nichols e	
~ - /	con il TIA Portal	188
6.74	Regolatore di flusso sulla sinistra e valvola ad azionamento meccanico sulla destra	189
6.75	Risposta del sistema alla presenza di un disturbo esterno	189
6.76	Anteprima sul TIA Portal della pagina dedicata al pannello 3 del banco didattico	190
6.77	Inserimento e configurazione dell'elemento "barra grafica" dalla casella degli stru-	101
	menti	191
7.1	Confronto tra le risposte ottenute ottimizzando con il TIA Portal	194

## Abstract

L'elaborato qui presentato nasce con l'idea di progettare, strutturare e mettere a disposizione del Politecnico di Torino uno strumento utile allo studio e alla comprensione dell'automazione industriale. In particolare, è stato costruito un banco didattico, targato Pneumax, all'interno del quale sono stati integrati sistemi complessi, quali PLC e HMI, e componenti semplici di natura pneumatica ed elettropneumatica. Il banco didattico esposto nel corso della tesi sarà messo a disposizione degli studenti per le esercitazioni accademiche e consentirà loro di interfacciarsi con lo strumento comprendendone la logica di funzionamento.

Il banco è stato progettato con lo scopo di essere utilizzato per esperienze differenti, a questo fine è stato suddiviso in una parte fissa e in una mobile. La parte fissa, cuore del banco, è stata progettata per permettere agli studenti di apprendere in sicurezza e di effettuare in autonomia i più semplici collegamenti tramite morsettiere e connettori a banana. La sezione mobile, invece, è strutturata per essere flessibile e modulabile in base ai pannelli sperimentali che verranno via via analizzati.

Tutte le esperienze pensate e progettate per il presente banco didattico si focalizzano sul controllo dei differenti sistemi strutturabili mediante PLC e HMI. Tra le altre cose, sarà possibile studiare: il funzionamento di un sistema a tre cilindri comandati da elettrovalvole; l'utilizzo di trasduttori pneumo-elettrici e modulo I/O; il controllo in pressione di un serbatoio mediante la funzione PID prevista dal TIA Portal. All'interno dell'elaborato si analizzeranno nel dettaglio il banco e le esperienze sopracitate, descrivendone la genesi, il processo di progettazione e la conseguente realizzazione fisica. Particolare attenzione verrà posta alla strutturazione e programmazione del PLC e del Touch Panel in base alle esigenze emerse.

### Capitolo 1

## L'automazione industriale e introduzione alle reti per l'automazione

#### 1.1 Introduzione all'automazione industriale

Diverse sono le definizioni che esprimono il concetto di "Automazione". C'è chi l'interpreta come una disciplina, chi come un insieme di tecnologie e infine chi come una funzione. Con il termine automazione spesso si intende tutto ciò che è necessario per trasformare un processo o più semplicemente una macchina in modo automatico, ovvero senza l'intervento umano. Storicamente essa è nata con il principale obiettivo di sostituire i noiosi, nocivi e ripetitivi compiti dell'uomo con meccanismi in grado di operare in maniera autonoma o, in caso di estrema necessità, con minimi interventi da parte degli operatori. L'automazione industriale sfrutta le tecnologie meccaniche, informatiche ed elettroniche per il controllo di flussi di energia, di materiali e di informazioni necessari alla realizzazione di processi produttivi. I benefici derivanti da questa nuova tecnologia sono stati molteplici e tra questi si ha:

- riduzione dei costi, dei tempi e degli scarti di produzione;
- riduzione dell'impatto ambientale;
- miglioramento della qualità dei prodotti;
- risparmio energetico;
- flessibilità della produzione, ovvero ci potrebbe essere la possibilità di utilizzare lo stesso sistema produttivo per diversi prodotti.

Tra gli svantaggi si potrebbe pensare, dalla definizione di automazione, alla disoccupazione. Tuttavia, questo risulta un falso svantaggio in quanto l'automazione *elimina* lavori pericolosi e ripetitivi e *crea* lavori di profilo più alto come manutentori, sviluppatori, progettisti e installatori.

#### 1.2 Un cenno alla storia: l'evoluzione dell'automazione industriale

In questo paragrafo si vuole fare un excursus storico sulla storia dell'automazione industriale in modo da poterne comprendere quanto importante e rilevante sia questo fenomeno oggigiorno. Sin dai tempi più remoti, un processo produttivo può essere definito come un trasformazione fisica di materiali effettuata al fine di ottenere un prodotto desiderato [2]. Essenziali per un qualsiasi processo produttivo sono tre fattori: energia, informazione e controllo. Facendo riferimento a questa definizione si può affermare che sin dalla prestoria l'uomo ha sempre realizzato processi produttivi inconsapevolmente (ad esempio, partendo dalla pietra grezza, realizzava utensili da caccia usando come energia la propria forza e il suo cervello come controllo sull'intero processo).

Col passare dei secoli, l'uomo ha sempre più cercato di sottrarsi al lavoro fisico (se non totalmente almeno parzialmente) in quanto esso risultava anzitutto faticoso, noioso se tale processo richiedeva movimenti ripetitivi, ma anche pericoloso. Per cui la storia vede gradualmente un'eliminazione dell'intervento dell'uomo e l'introduzione di altri mezzi di energia capaci di garantire lo stesso risultato. Tra i primi tentativi ritroviamo l'utilizzo di energia animale (aratura dei campi o estrazione di acqua dai pozzi) o lo sfruttamento dell'energia idraulica o eolica nei mulini ad acqua o a vento. Si giunge così a parlare di *industrializzazione*, la quale raggiunge il suo apice nella rivoluzione industriale, quando i processi produttivi vengono caratterizzati dall'uso generalizzato di macchine azionate da energia meccanica e dall'utilizzo di nuove fonti energetiche inanimate (come ad esempio i combustibili fossili).

Intorno alla fine del Settecento, esattamente nel 1787, risiede forse il precursore dei sistemi di automazione realizzato da James Watt. Si tratta del *regolatore centrifugo* per controllare la velocità di una macchina a vapore. Questo regolatore, il cui scopo iniziale era di mantenere la velocità costante indipendentemente dal peso trainato o dalle pendenze della strada ferrata, faceva accelerare la macchina se questa rallentava per il troppo carico o viceversa la faceva rallentare in caso di una diminuzione del carico. Nella figura 1.1 viene riportato lo schema di questo meccanismo.



Figura 1.1: Regolatore centrifugo di J. Watt [3]

Come si evince dall'immagine, il suo funzionamento si basa, come dice il nome stesso, sugli effetti della forza centrifuga sulle due masse sferiche. In particolare se la velocità aumenta, le sfere si allargano e, tramite un sistema di leveraggio, fanno chiudere la valvola a farfalla; in questo modo la quantità di vapore diminuisce e la macchina rallenta. Al contrario, se la macchina rallenta, le masse si abbassano e fanno aprire la valvola, la quale fa passare molto più vapore accelerando la stessa. I sistemi automatici moderni nacquero alla fine del secolo successivo dalla necessità di avere macchine sempre più veloci e precise: erano, dunque, necessari meccanismi capaci di correggere in maniera autonoma i fattori di disturbo che alteravano il funzionamento delle macchine. Si è iniziato a parlare per la prima volta di *automazione* nel 1947 alla *Ford Motor Company*: tale termine fu utilizzato per indicare l'insieme delle movimentazioni automatiche che erano installate sulle loro linee di produzione. Fino a questo momento gli elaboratori erano dei

dispositivi di tipo meccanico o pneumatico che consentivano di eseguire algoritmi di elaborazione molto limitati e rappresentava un grosso problema anche la loro connessione a sensori e attuatori.

All'inizio degli anni Settanta, lo sviluppo dell'elettronica ha consentito l'utilizzo di schede a basso costo e di dimensioni ridotte per la regolazione e il controllo di valvole e attuatori. Il fattore vincente che rese possibile il progetto di sistemi di controllo con capacità (potenza di calcolo) superiori a quelli utilizzati ai tempi nelle industrie furono dei congegni creati in ambito informatico/elettronico: i calcolatori. Per il controllo dei segnali analogici e la regolazione di processi chimici e termici furono sviluppati i DCS, ovvero *Distributed Control System*. Per la gestione di dispositivi elettrici convenzionali nacquero dei congegni in grado di elaborare segnali digitali, con lo scopo di sostituire i quadri elettromeccanici composti da relè, temporizzatori o conta-impulsi: i PLC, ovvero *Programmable Logic Controller*.

La richiesta sempre maggiore di questi elaboratori ha portato ad eseguire funzioni di controllo sempre più avanzate. Lo sviluppo delle tecnologie informatiche e l'avvento di Internet hanno portato una vera e propria rivoluzione dei sistemi di automazione. Oggi sono parte essenziale del mondo dell'automazione e del controllo tecnologie avanzate, come quelle che sovrintendono a Scada, bus di campo, sensori wireless, internet of things e comunicazione autonoma tra le macchine (M2M), Cloud, strumentazione virtuale, sistemi cyberfisici, smart sensor e soluzioni per l'intelligence d'impianto [4].

#### 1.3 Composizione di un sistema di automazione

Col passare degli anni, nel campo dell'automazione industriale, la tecnologia pneumatica vede una maggiore integrazione con l'informatica e l'elettronica per via della crescente competizione nel mercato globale e una maggiore riduzione dei costi, soprattutto per i grandi volumi produttivi [8]. Lo sviluppo dei sistemi di automazione necessita una rete di comunicazione tra i diversi utenti a partire dai sistemi di controllo dei processi fisici fino ai sistemi di supervisione e di gestione di un'azienda. A seconda delle distanze, delle velocità di trasmissione dati o dal tipo di protezione, tali collegamenti possono differire [10]. Per sistemi di automazione più articolati, ovvero quelli composti da più PLC e altri dispositivi, possono insorgere problemi di trasmissione dati e collegamento, problemi che per semplici sistemi, invece, non esistono. In questo caso viene adoperata l'automazione della produzione integrata di fabbrica con una rete di comunicazione evoluta. Il modello di riferimento è il CIM, noto come *Computer Integrated Manufacturing*, il quale, appunto, è un sistema automatizzato che con la tecnologia informatica computerizzata unisce tutti i settori di progettazione, produzione e gestione di un'azienda con l'unico obiettivo di perfezionare la fabbricazione e la distribuzione di un prodotto.

In particolare, con il termine *manufacturing* si intende l'insieme dei processi fisici e/o chimici che devono essere applicati ai materiali grezzi/semi-lavorati al fine di ottenere il prodotto finale [2]. Esso è dunque un processo di trasformazione di materiali, solitamente di tipo sequenziale, che avviene attraverso l'uso di più fattori come macchinari, utensili e intervento umano.

Per ottimizzare il processo produttivo, alcune delle attività di produzione o di supporto risultano automatizzate. Negli impianti di produzione automatici vi sono spesso robot industriali per la lavorazione e l'assemblaggio delle parti, linee di trasporto la cui movimentazione risulta essere automatica oppure sistemi di controllo qualità anch'essi automatici. Tuttavia, è possibile fare una distinzione tra automazione rigida, programmabile e flessibile [2]:

• un sistema di *automazione rigida* prevede una sequenza di operazioni per la produzione fissa. Solitamente tali sistemi sono destinati a grandi produzioni in cui la frequenza di produzione è molto elevata, mentre la varietà di prodotto è limitata, se non assente;

- un sistema di *automazione programmabile* prevede una sequenza di operazioni per la produzione tale da poter variare in base al prodotto finale che si vuole ottenere. In genere, questo tipo di automazione è utilizzata per una produzione medio-bassa, caratterizzata da processi produttivi a lotti, dove ad ogni lotto corrisponde un prodotto diverso. Ovviamente, per questi tipi di impianti automatici c'è bisogno di un tempo morto per la riconfigurazione dell'impianto in base al prodotto desiderato, scaricando nell'elaboratore il nuovo programma di produzione;
- un sistema di *automazione flessibile* rappresenta un'estensione del sistema di automazione programmabile in cui, a differenza del precedente, è possibile diversificare la produzione senza avere tempi morti di conversione dell'impianto. Tale flessibilità è riferita all'ordine delle lavorazioni oppure all'assegnamento di una lavorazione a una stazione di lavoro.

Nella figura 1.2 è riportata la classificazione delle diverse tipologie di sistemi di automazione in funzione del volume di produzione e della varietà del prodotto.



Figura 1.2: Classificazione dei differenti sistemi di automazione [5]

Per quanto concerne invece l'automazione delle attività di supporto è importante mezionare il DSS, nonchè il *Decision Support System*. Si tratta di un software che garantisce all'utente una serie di funzionalità di supporto ai processi decisionali con l'obiettivo di elevare l'efficienza e l'efficacia del processo decisionale.

Un altro strumento è l'ERP, ovvero l'*Enterprise Resource Planning*, il quale rappresenta un insieme di funzioni informatiche utilizzate nel campo dell'automazione di attività di amministrazione e finanza, logistica gestione della produzione e delle risorse umane.

Infine vi sono anche altri sistemi di supporto per l'automazione come il CAD e il CAE che consentono la progettazione automatizzata di forme e funzionalità del prodotto finale, ma anche sistemi CAM e CAPP (*Computer Aided Process Planning*) che, invece, consentono di automatizzare prove di fattibilità, simulare processi di produzioni e generare il programma macchina per ogni operazione di lavorazione.

Nei sistemi di automazione moderni, l'automazione dell'impianto di produzione e l'automazione delle attività di supporto alla produzione vengono integrati tra di loro grazie ad un'unica infrastruttura informatica distribuita che coinvolge l'intero sistema di produzione (vedi figura 1.3). Questo fenomeno prende appunto il nome di *Computer Integrated Manufacturing* o meglio conosciuto come CIM. Tale integrazione risulta fondamentale al fine di conformarsi a leggi e norme di sicurezza del processo produttivo e qualità del prodotto e allo scopo di eliminare l'impatto energetico-ambientale del processo stesso.



Figura 1.3: Schema di un sistema Computer Integrated Manufacturing [6]

#### 1.4 Architettura del CIM

Come è stato già detto nel paragrafo precedente, il CIM si definisce dunque il sistema informativo dell'azienda che deve essere modellato attorno alle caratteristiche procedurali e funzionali di tutta l'organizzazione aziendale come il sistema nervoso nel corpo umano [12]. Il CIM è strutturato su sei livelli da un'adeguata norma ISO, che definisce l'architettura degli scambi informativi attraverso i vari livelli della singola macchina e trasversalmente tra le diverse macchine, impianti o siti produttivi [10]. Di seguito, viene riportata nella figura 1.4 uno schema del CIM semplificato su quattro livelli.



Figura 1.4: Schema a piramide del CIM [9]

Come è possibile notare dall'immagine, si possono distinguere i vari livelli:

- livello CAMPO, definito anche *livello di processo* o *sensor/actuator area*, permette l'interfacciamento tra i controllori ed i processi controllati tramite attuatori e sensori. L'architettura dei sistemi automatici su questo livello può avvenire secondo due modalità, ovvero o realizzato con connessioni elettriche dirette tra i diversi componenti oppure con una rete di comunicazione per il campo (*Fieldbus*).Per quanto riguarda la quantità di informazioni scambiate è dell'ordine dei bit, mentre la durata e la frequenza di trasmissione sono dell'ordine di micro e millisecondi;
- livello CONTROLLO, definito *livello del comando del processo* o *field area*, nel quale si trovano i comandi di gestione e sincronizzazione delle macchine, controllo dei processi parziali e macchine operatrici a controllo numerico attraverso CNC e PLC. Per quanto concerne la quantità di dati è dell'ordine dei byte, mentre la frequenza di trasmissione è dell'ordine dei 100 millisecondi;
- livello SUPERVISIONE, definito *livello di conduzione del processo* e *livello di direzione della produzione*, si occupa della gestione e controllo dei vari processi produttivi, come prefabbricazione, produzione e collaudo, mediante PC o sistemi di monitoraggio. Tra i principali compiti svolti da questo livello ci sono la *pianificazione*, la *preparazione* e la *distribuzione* degli ordini di produzione delle singole celle. In questo caso, la quantità di informazioni scambiata è dell'ordine dei kbyte, mentre la frequenza di trasmissione è circa dell'ordine dei minuti o ore;
- livello ENTERPRISE, definito anche *livello di direzione aziendale*, che permette i collegamenti tra i reparti interni dell'azienda, ma anche la comunicazione esterna alla stessa. Per cui si tratta di comunicazioni a stampo politico ed economico. Ovviamente la quantità di dati è elevata e il tempo di trasmissione è molto lungo, rivestendo un ruolo quasi secondario.

Il modello CIM può essere semplificato in soli due livelli, i quali possono essere denominati semplicemente livello basso e livello alto. Il livello basso racchiude i primi tre livelli del modello CIM completo, mentre il livello alto i restanti tre livelli. Il compito principale del livello basso è supervisionare tutti i processi tecnologici di produzione, basati su sistemi CAM (*Computer-Aided Manufacturing*), sistemi CAD/CAM integrati e sistemi di automazione distribuita con uso dei PC,PLC, attuatori e sensori. Il ruolo principale del livello alto, invece, si basa sulla progettazione computerizzata con l'uso dei sistemi CAD (*Computer Aided Design*) e CAE (*Computer Aided Engineering*).Per questo livello vengono anche utilizzati sistemi SCADA (*Supervisory Control And Data Acquisition*), che acquisiscono, elaborano e presentano i dati di produzione in tempo reale realizzando l'interfaccia uomo-macchina.

#### 1.5 Reti di comunicazione per l'automazione ed esempi di reti di campo

Per poter concretizzare un qualsiasi sistema CIM occorre l'utilizzo delle reti di comunicazione sia all'interno dello stesso livello sia tra i diversi livelli dell'intero sistema. Queste reti si possono distinguere in due grandi categorie: reti di automazione e reti di informazione, le quali corrispondono rispettivamente ai livelli CIM basso e alto. Le prime a loro volta si suddividono in reti per il controllo e reti di campo.

Le reti per le informazioni vengono utilizzate tra le apparecchiature riservate al controllo e gestione dello stabilimento e dell'azienda. Le reti per il controllo vengono utilizzate tra i dispositivi di controllo e di supervisione degli impianti e delle singole isole di produzione. Infine, le reti per il campo o bus di campo servono per effettuare le comunicazioni tra dispositivi di controllo con sensori e attuatori.

Le reti di campo si usano per il controllo in real-time dei sistemi industriali e sono realizzate per il collegamento tra apparecchiature di controllo (PLC), sensori e attuatori. A livello di standardizzazione tali reti rimandano alla IEC 61158, il cui compito è quello relativo alla gestione in tempo reale delle esigenze trasmissive dei processi afferenti al sistema di comunicazione a Bus di Campo. A questo punto, è possibile distinguere due diverse soluzioni a livello di campo: l' architettura tradizionale, la quale prevede dei collegamenti punto-punto in cui i sensori e gli attuatori sono collegati direttamente al PLC, e l'architettura a bus di campo (fieldbus). La topologia di rete più impiegata è del tipo ad albero o a Bus, dove sono presenti dei Master, unità intelligenti di comando e controllo (come PC o PLC), e Slave, collegate in parallelo. In campo pneumatico, il tipico componente che è presente sul campo che scambia informazioni con il livello inferiore del CIM è il pacchetto valvole: si tratta di un'unità slave per il comando degli attuatori, collegato al bus di campo mediante un nodo. Anche HMI (*Human Machine Interface*) rappresentano altre unità slave. Un esempio di rete Fieldbus è riportato nella figura 1.5.



Figura 1.5: Esempio di rete di campo con più unità Master che controllano le unità Slave [6]

Ciascuna unità è collegata alla rete mediante un doppino, fibra ottica o cavo coassiale. Sul Bus vengono trasportati tutti i dati in modo bidirezionale fra tutte le unità connesse alla rete. Per quest'ultima tipologia è possibile elencare i seguenti vantaggi:

- le risorse per il cablaggio ridotte grazie alla trasmissione dati seriale;
- la possibilità di creare un collegamento diretto tra dispositivi *intelligenti* (sensori ed attuatori);
- i tempi di guasto e manutenzione diminuiti grazie all'autodiagnosi del sistema;
- la scelta flessibile dei componenti di sistema: basata su standard aperta;
- maggiore sostenibilità: estensioni e modifiche semplici da effettuare.

Quando si parla di sensori e attuatori intelligenti si intende dispositivi dotati di un microprocessore programmato internamente che esegue diverse funzioni, tra cui quelle di calibrazione, diagnostica, conversione A/D e D/A dei dati. Nonostante tutti questi vantaggi, tali dispositivi non sono ancora diffusi e per i normali dispositivi, ovvero quelli non intelligenti per ovviare a problemi di connessione al bus di campo si utilizzano degli adattatori.



Figura 1.6: Esempio di rete di campo con utilizzo di dispositivi intelligenti [10]

Di seguito, invece, vengono riportati alcuni esempi di reti di campo e di controllo tra le più utilizzate nel campo dell'automazione.

- 1. AS-Interface, sviluppata da Siemens come sostituzione al cablaggio tradizionale. Tale rete è caratterizzata da un'installazione molto semplice e nasce con l'obiettivo di sviluppare un semplice collegamento in rete di sensori e attuatori e di trasportare attraverso un solo cavo (profilo piatto e di colore giallo) energia e dati. Mediante l'utilizzo di ripetitori è possibile estendere tale rete fino a 600 metri. La velocità raggiungibile sfiora i 167 kbit/s. Il collegamento dei cavi AS-i è veloce grazie alle prese con la tecnologia a piercing a perforazione di isolante. Infine, la topologia è libera e il sistema è aperto, per cui sono possibili tutte le architetture.
- 2. InterBus-S, sviluppata dalla società Phonix Contact per il campo dei sensori e attuatori. La topologia utilizzata da questa rete è quella ad anello gestita con il protocollo master/slave. Essa può gestire fino a circa 4000 ingressi e uscite digitali o combinazioni di ingressi e uscite digitali e analogiche. La velocità di trasmissione dati arriva fino a 500 kbit/s. Per quanto riguarda il cavo adoperato per questa rete può essere un doppino telefonica, fibra ottica oppure trasmissione ad infrarossi.
- 3. Profibus-DP, sviluppata dalla Siemens per estendere il campo Profibus al campo dei sensori e attuatori. Essa si basa su una topologia a bus e realizza sistemi mono/multi-master adoperando il protocollo del token-bus. La velocità raggiunta per la trasmissione dei dati risulta molto più elevata rispetto alle reti precedenti e può raggiungere i 12 Mbit/s. Il messo utilizzato di trasmissione è il doppino intrecciato e schermato oppure fibra ottica. Può raggiungere un estensione di 1200 metri mediante l'utilizzo di ripetitori.
- 4. DeviceNet, progettata da Allen-Bradley e consente il collegamento tra controllori a logica programmabile e calcolatori tradizionali, da un lato e sensori e attuatori industriali dall'altro. Il protocollo su cui è basato è il CAN-bus (*Controller Area Networking*), sviluppata dalla Bosch e applicata inizialmente in campo automobilistico, mentre oggi in quasi tutte le applicazioni industriali. La rete può essere utilizzata sia in modalità master-slave e sia in modalità peer-to-peer con un massimo di 64 nodi collegabili. Le velocità di trasmissioni possono raggiungere al massimo 500 kbit/s. Anche in questo caso il mezzo di trasmis-

sione può essere due doppini intrecciati schermati, uno utilizzato per l'alimentazione dei dispositivi e l'altro per la trasmissione dei segnali.

- 5. DataHighway+, si tratta di una rete per il controllo utilizzata per il collegamento tra PLC e macchine a controllo numerico. La velocità di trasmissione raggiunge un valore di 57.6 Kbit/s con un doppino intrecciato schermato lungo fino a 3000 metri. Il metodo di accesso è di tipo token-bus con 64 nodi collegabili
- 6. ControlNet, ultimo prodotto Allen-Bradley nel campo delle reti per il controllo. La velocità di trasmissione raggiunge i 5 Mbit/s con un massimo di 99 nodi collegabili. Il cavo utilizzato è un cavo coassiale schermato, la cui lunghezza massima arriva a 1000 metri, che con i ripetitori può raggiungere i 6000 metri. Il metodo di accesso è il CTDMA (Concurrent Time Domain Multiple Access, ovvero accesso multiplo concorrente nel dominio del tempo): per ogni intervallo temporale tutti i nodi abilitati possono comunicare con una data frequenza; i restanti nodi possono eventualmente comunicare dati se il tempo che rimane permette.

Le informazioni riportate delle diverse reti sono state estrapolate dai libri Automazione dei sistemi meccanici: corso base [10] e PLC e automazione industriale [13].

#### 1.6 Il modello di riferimento di rete ISO/OSI

Fino a questo momento si è parlato della rete di comunicazione, la quale serve a instradare le informazioni e trasmetterle da un'unità ad un'altra. Il protocollo di comunicazione è invece la modalità attraverso la quale il nodo invia e riceve dati. Ogni protocollo stabilisce il mezzo e la distanza di trasmissione, la sua velocità, il numero di nodi collegabili e le interfacce tra i livelli al fine di garantire un'adeguata comunicazione. Scelto il protocollo di comunicazione, tutti i nodi devono essere definiti per quello stesso protocollo. Essi sono standardizzati dalla ISO/OSI (*Open System interconnection*), il cui modello di rete è uno standard di comunicazione tra reti diverse permettendo la comunicazione tra componenti hardware e software di fornitori diversi.

Il modello di riferimento OSI è caratterizzato da sette livelli [10], ognuno dei quali svolge un particolare compito per garantire una buona comunicazione tra due entità di pari livello. I livelli sono divisi in base alla loro funzione e sono:

- livelli che definiscono il tipo di collegamento in rete con i protocolli orientati ai flussi di dati:
  - 1. Fisico;
  - 2. Linea;
  - 3. Rete.
- livelli che stabiliscono il protocollo da e verso i protocolli applicativi con funzioni di sicurezza:
  - 4. Trasporto;
  - 5. Sessione.
- livelli di applicazione con servizi orientati all'utente:
  - 6. Presentazione;
  - 7. Applicazione.

Può capitare, tuttavia, che nella pratica non tutti i livelli siano utilizzati, come appunto accade per la maggior parte delle reti di automazione, le quali si rifanno ad un modello OSI semplificato con tre livelli: fisico, linea e applicazione. Il primo livello è responsabile della trasmissione dati mediante un mezzo fisico di trasmissione, il secondo livello garantisce che la trasmissione avvenga da un nodo ad un altro senza aver alcun tipo di problema, mentre il terzo livello offre all'utente i servizi attraverso i quali può comunicare con la rete (posta elettronica, terminali, ecc.)

#### 1.7 Metodi di accesso per le reti standard e integrazione tra reti

Tra le funzioni principali del livello 2, *Linea*, del modello OSI bisogna ricordare il metodo di accesso, il quale deve definire due principali aspetti:

- 1. determinare i nodi tra i quali si deve creare un collegamento;
- 2. evitare i conflitti tra diversi nodi che simultaneamente richiedono l'utilizzo del canale di comunicazione.

Tra i più usati metodi di accesso nelle reti di campo vi sono:

• accesso master/slave: il nodo Master interroga ciclicamente gli slaves per inviare e ricevere rispettivamente comandi e dati da essi; il Master ha la possibilità di accedere alla rete in maniera indisturbata, mentre gli Slave solo se interrogati dal nodo principale. Tale metodo risulta essere preciso e affidabile, con l'unica pecca di non avere alte velocità di accesso e non dare differenti priorità di accesso.



Figura 1.7: Schema di rete con accesso Master/Slave [14]

- accesso multi-master: è una rete che può contenere due o più master che possono accedere in maniera indipendente ad essa. Questo è il modo più rapido per rispondere ad esigenze di tipo real-time. Il problema principale per questa tipologia di accesso sovviene nel momento in cui simultaneamente due Master cercano di accedere alla rete. Le soluzioni adottate per risolvere tale problema sono le seguenti:
  - 1. accesso di tipo random CSMA/CD;
  - 2. accesso con priorità CSMA/BA e CSMA/CA;
  - 3. accesso token-passing.

Il *Profibus* è un esempio di rete con il protocollo token-passing.



Figura 1.8: Schema Profibus: metodo di accesso Token Bus [15]

Una volta definiti i differenti modi di accesso per le reti standard, un altro problema ricorrente è il collegamento e l'integrazione tra reti [10], problema che si ripercuote sia se si tratta dello stesso tipo di rete e sia se si tratta di due reti completamente diverse. Ciò viene superato attraverso l'utilizzo di strumenti particolari che sono:

- il *ripetitore*, il quale è un dispositivo che permette di collegare le diverse tratte lunghe della stessa rete. Esso coinvolge solitamente il livello 1 dello standard ISO/OSI, lavorando appunto al livello fisico con segnali deboli, i quali vengono rigenerati e ritrasmessi;
- il *ponte*, il quale, invece, consente di collegare delle reti che lavorano con lo stesso protocollo e permette di collegare più segmenti in una stessa rete locale. Esso opera a livello software al livello 2 dello standard ISO/OSI ed è capace di indirizzare e filtrare i dati che si trovano ad un terminale del ponte;
- l'*instradatore*, il quale consente il collegamento delle reti al livello 3 dello standard ISO/OSI. L'instradatore o anche conosciuto come *router* instrada i dati per indirizzarli verso la rete di destinazione nel modo corretto, la cui rete potrebbe essere anche di tipologia diversa;
- la *porta*, la quale permette il collegamento di reti la cui struttura di protocollo risulta differente. Esattamente come l'instradatore lavora al livello 3 dello standard ISO/OSI.

Tra questi quattro dispositivi, mentre i primi due non generano un ritardo nella trasmissione dati, l'instradatore e la porta possono provocare un consistente ritardo delle informazioni scambiate (motivo per il quale tali congegni non si utilizzano per il controllo dei processi).



Figura 1.9: Schema di un collegamento tra reti [16]

### Capitolo 2

## Il PLC: il controllore a logica programmabile

#### 2.1 L'evoluzione dei sistemi di controllo

Uno dei principali scopi dell'uomo, sin dai tempi più antichi, è stato quello di far eseguire in maniera automatica processi di lavorazione senza il suo intervento. Prima della scoperta dell'elettricità, l'uomo si era cimentato nella costruzione di automatismi con controlli esclusivamente meccanici come ad esempio orologi a pendolo, orologi ad acqua oppure, come è stato già detto nel capitolo 1 di questa tesi, il regolatore di velocità di Watt.

Con l'invenzione dell'elettricità si iniziarono ad usare dispositivi elettromeccanici per poter costruire automatismi e tra questi nuovi congegni trovano luogo relè e temporizzatori, i quali venivano cablati tra di loro. Proprio per questo motivo tale sistema di controllo prese il nome di *logica cablata*. Con il passare degli anni, la tecnologia elettromeccanica continuava a fare passi da giganti permettendo così la realizzazione di sistemi di controllo sempre più complessi.

I relè, i quali vengono poco utilizzati oggigiorno, sono collegati fisicamente con cavi elettrici, secondo una determinata sequenza, con l'obiettivo di ottenere un comando particolare in risposta ad una certa situazione della macchina. Tale sistema di controllo è conosciuto anche come *Comando elettrico a relè*. Tuttavia, questi dispositivi presentavano alcuni svantaggi come una velocità di elaborazione dei segnali molto bassa, un costo molto elevato e una progettazione e una manutenzione lenta e laboriosa.

Con l'avvento dell'elettronica e, in particolare, con l'invenzione del transistor nel 1948, la situazione sembrava migliorare ottenendo caratteristiche notevolmente superiori rispetto ai classici relè. Tra i vantaggi principali vi erano:

- ingombri ridotti;
- velocità di elaborazione dati maggiore;
- minor costo;
- facilità di adattamento nelle diverse applicazioni del campo dell'automazione.

Con la comparsa dei circuiti integrati vennero migliorate ulteriormente le caratteristiche di questi dispositivi per lo scopo per cui erano nate, ma in ogni caso non permettevano di realizzare automatismi abbastanza flessibili tali da poterli adattare, in tempi brevi, a nuove situazioni. Tuttavia, il sistema di controllo che utilizza questi componenti rimane sempre del tipo a logica cablata, in quanto la circuiteria di controllo richiede la progettazione dello schema di collegamento di tutti gli elementi e il loro cablaggio sulle schede elettroniche. Questo sistema di controllo è anche conosciuto come *Comando elettronico*.

Il 1968 segna un anno rivoluzionario per il campo dell'automazione: è l'anno della nascita dei PLC (*Programmable Logic Controllers*) o anche noti come *Controllori a logica programmabile*, i quali furono sviluppati per la prima volta dalla *General Motors Corporation* negli Stati Uniti per ovviare al problema della rottamazione dei costosi sistemi di controllo a relè delle linee di assemblaggio, ogni volta che cambiava il modello di auto da produrre. La logica è detta *programmabile* perché è possibile modificare semplicemente le istruzioni del programma, senza dover ricablare gli elementi logici. Ovviamente, i primi PLC avevano una logica programmabile limitata alle funzioni logiche che erano realizzabili con i relè. Solo dopo cinque anni, nel 1973, furono realizzati i primi PLC *intelligenti*, ovvero capaci di svolgere operazioni aritmetiche e manipolare i dati delle memorie.

Nel 1975 è stata realizzata la funzione PID (Proporzionale, Integrativa, Derivativa), la quale mediante un algoritmo di regolazione di un anello di retroazione ha permesso l'implementazione nel PLC di controlli di regolazione mediante l'uso di sensori. Solo un anno dopo il PLC è stato inserito nel modello gerarchico del CIM.

I dati storici evidenziati in questo elaborato sono stati estrapolati dai libri *PLC e automazione* industriale [13] e Il *PLC: teoria, funzionamento e applicazioni* [12].



Figura 2.1: Evoluzione dei sistemi di controllo

#### 2.2 Definizione, vantaggi e classificazione dei PLC

Per poter definire efficacemente il PLC, noto anche come *Controllore a logica programmabile*, ci si avvale dello standard 1131 [17] del Comitato Elettrotecnico Internazionale (IEC), il quale lo intende come un:

sistema elettronico a funzionamento digitale, destinato all'uso in ambito industriale, che utilizza una memoria programmabile per l'archiviazione interna di istruzioni orientate all'utilizzatore per l'implementazione di funzioni specifiche, come quelle logiche, di sequenziamento, di temporizzazione, di conteggio e di calcolo aritmetico, e per controllare. mediante ingressi ed uscite sia digitali che analogici, vari tipi di macchine e processi. Nel gergo comune si utilizza la parola "PLC" per indicare l'insieme dei dispositivi hardware e il software relativo, i quali costituiscono un automatismo basato su un controllore a logica programmabile. Diversi sono i vantaggi che il PLC ha portato con sé e tra i più importati ritroviamo:

- l'essere facilmente riprogrammabile anche sul luogo in cui lavora con tempi morti brevi;
- l'essere realizzato con materiali consoni all'ambiente industriale, determinando una maggiore robustezza;
- l'avere ingombri e consumi energetici molto limitati;
- l'essere in grado di colloquiare in maniera semplice e istantanea con sensori e attuatori;
- l'avere costi molto bassi;
- l'avere una memoria interna espandibile, sia per programmi che per dati.

Dunque, ne viene fuori un dispositivo all'avanguardia che ha un costo relativamente basso, molto flessibile, affidabile con potenzialità mostruose, come ad esempio realizzare delle funzioni complesse di calcolo, memorizzare dati, mostrare grafici e comunicare con altri calcolatori attraverso reti di trasmissione dati. Ovviamente, in base alle esigenze, esistono diversi tipi di PLC. Essi si possono classificare essenzialmente in base al numero di I/O che sono in grado di gestire e alla capacità di memorizzazione. Partendo da quello più economico si hanno:

- i *nano PLC*, i quali gestiscono fino ad un massimo di 10 I/O e hanno una bassa capacità di memorizzazione;
- i micro PLC, i quali trattano fino a 64 ingressi ed uscite generalmente tutti digitali. Solitamente non hanno una struttura modulare ad armadio, garantendo però delle espansioni. Tra le varie funzioni a cui può avere accesso si ritrovano quelle di base, temporizzazione, conteggio e calcolo mediante operazioni matematiche;
- i *piccoli PLC*, i quali gestiscono dai 64 a 256 ingressi/uscite che possono essere sia digitali che analogiche. Hanno una struttura modulare ad armadio e possono avere capacità di connessione in rete e di gestione da remoto;
- i *medi PLC*, i quali arrivano fino a 2048 unità di ingressi e uscite. Anch'essi possono avere una struttura modulare ad armadio e a differenza dei precedenti questi possono essere arricchiti anche da moduli speciali. La capacità di comunicazione in rete è molto alta e il kit di istruzioni risulta essere abbastanza completo;
- i grandi PLC, i quali raggiungono le massime prestazioni per ogni caratteristica con la possibilità di raggiungere un numero di ingressi e uscite molto superiore a 2048.

I sistemi basati sui PLC trovano applicazione maggiormente nel campo dell'automazione industriale. In particolare, tra i principali settori di utilizzo ci sono il controllo di macchine industriali, la movimentazione del materiale nei cicli produttivi, l'assemblaggio dei prodotti, il controllo delle macchine utensili, il controllo da remoto, ecc. Tra le principali aziende produttive di controllori a logica programmabile si possono citare la *Siemens*, l'*Allen-Bradley* e l'*Omron*.



Figura 2.2: Siemens: varie tipologie di PLC [18]

#### 2.3 Il funzionamento dei PLC

Nel campo dell'automazione, il PLC è il *computer industriale* delle aziende, utilizzato principalmente per controllare processi [12]. Esso si discosta dai computer comuni in quanto è stato progettato esclusivamente per interfacciarsi con il mondo industriale, ovvero resistente ai luoghi polverosi che hanno una certa umidità e soggetti a numerose escursioni di temperatura. Inoltre, il PLC è protetto dalle impurità dei segnali elettrici e dalle interferenze RF (noti come *Noise* o *Rumori*). Il PLC si differenzia anche dalle macchine a controllo numerico (CNC) perché quest'ultime sono utilizzate per controllare la posizione, mentre i PLC per controllare sequenze.



Figura 2.3: Schema semplificativo di come funziona un PLC [19]

Altra funzione che rende il PLC unico nel suo genere è il fatto che esso sostituisce alla

logica cablata la logica programmabile, avvicinandosi quindi ad un minicalcolatore. Rispetto a quest'ultimo, il PLC ha come scopo principale quello di lavorare in tempo reale con le macchine industriali. Per poter adempiere a questa funzione in maniera efficace, è necessario che nel PLC siano presenti due caratteristiche:

- 1. l'acquisizione e la gestione dei segnali di Input e Output provenienti dai sensori e diretti agli attuatori, mediante moduli hardware capaci di tradurre i segnali elettrici in valori digitali e viceversa;
- 2. l'elaborazione in poco tempo della risposta ad una situazione prevista da un programma per garantire stabilità nell'anello chiuso di controllo che vede come protagonisti il PLC e la macchina industriale (figura 2.4).

Ultimo aspetto fondamentale del PLC è il suo linguaggio di programmazione, il quale si presenta comprensibile e accessibile agli stessi operatori che lavoravano sui dispositivi antecedenti al PLC: infatti gli stessi si sono trovati a lavorare con un linguaggio di programmazione che ricordava il tradizionale schema di circuiti a relè. Tale linguaggio viene definito *linguaggio a contatti*. Questo è sicuramente uno dei tanti pregi di questo congegno, il quale permette la sua programmazione senza ricorrere a tecnici esperti.



Figura 2.4: Anello chiuso di controllo tra PLC e macchina

#### 2.4 L'architettura hardware del PLC

Il controllore a logica programmabile a bus presenta un'architettura hardware che viene definita dalla norma IEC 61131-2 [21]. Nella configurazione minima di un sistema PLC si trovano una CPU che rappresenta l'unità di calcolo centrale, la quale è dotata di una memoria utilizzabile sia per memorizzare il programma sia per registrare i dati e che è collegata ad un alimentatore. Mediante la struttura a bus la CPU è collegata a moduli di interfaccia input/output, i quali sono a loro volta collegati ai sensori e gli attuatori presenti sul campo.

Inoltre, è presente anche un'interconnessione, sempre mediante bus, con dei moduli speciali dedicati a particolari funzioni e con un'unità di programmazione. Il tutto viene racchiuso all'interno dell'armadio, anche noto come rack: quest'ultimo contiene fisicamente tutti i moduli



Figura 2.5: Architettura hardware di un PLC [19]

e ne assicura la connessione meccanica e il collegamento elettrico; la sua forma generalmente è parallelepipeda e aperta su un lato per garantire l'inserimento dei moduli, i quali risultano elettricamente collegati tra loro grazie alla presenza di un circuito stampato con dei connettori. Tale scatola è di metallo e deve essere connessa elettricamente a terra sia per schermare i moduli che ci alloggiano sia per motivi di sicurezza. Nella figura 2.6 è riportato un disegno schematico di come si mostra la struttura di un PLC preso dal libro *PLC e automazione industriale* [13].



Figura 2.6: Disegno schematico della struttura di un PLC

Si voglia a questo punto spiegare più nel dettaglio cosa sono e a che cosa servono i componenti sopra citati. Le informazioni riportate su questa tesi in merito alla struttura del PLC sono state estrapolate da diversi libri (vedere bibliografia [2] [12] [13] ).

#### 2.4.1 PLC: Modulo processore

Il modulo processore rappresenta il cuore del sistema PLC ed è composto da una scheda con un microprocessore (o anche più di uno), mediante il quale esegue i programmi del sistema operativo o quelli sviluppati dall'utente, e dalla memoria nella quale sono memorizzati i programmi e i dati presenti nel PLC. In generale, un modulo processore potrebbe includere tre microprocessori, ognuno dei quali è specializzato in qualcosa: il primo per lavorare sul singolo bit, il secondo sulle istruzioni di tipo aritmetico/logico e l'ultimo per comunicare con dispositivi esterni al PLC o con i moduli I/O.

Un modulo processore è solito funzionare seguendo un ciclo di operazioni in loop, noto come ciclo a copia massiva degli ingressi e delle uscite, che prevede:

- 1. l'aggiornamento dell'area di memoria mediante le informazioni ricavate dagli input fisici;
- 2. l'esecuzione dei programmi utente utilizzando i dati presenti in memoria e salvandoci i risultati;
- 3. l'esecuzione di programmi di gestione del sistema (diagnostica);
- 4. la scrittura sulle uscite fisiche dei risultati ricavati.

Il fatto che il sistema operativo si occupi della lettura degli input e la scrittura degli output consente all'utilizzatore di occuparsi solamente del programma utente. Per particolari contesti è anche possibile accedere immediatamente ai punti di ingresso ed uscita anche se tale ciclo non risulta terminato. Tuttavia, tale modifica potrebbe rallentare i tempi di scansione e, dunque, bisogna sfruttare questa possibilità solo se strettamente necessario (sono solitamente riservate alla gestione delle emergenze).



Figura 2.7: Modulo processore Siemens [20]

Per quanto riguarda il *tempo di scansione*, esso rappresenta la velocità di elaborazione del modulo processore e si definisce come il tempo che passa tra due attivazioni consecutive della stessa porzione di programma, tenendo conto anche del tempo di aggiornamento degli ingressi e e delle uscite. Ne consegue che il tempo di scansione dipende esclusivamente dal numero di input/output e dalla complessità del programma (tale valore viene segnalato dal produttore e spazia da qualche unità a qualche decina di millisecondi per Kiloword di programma).

Il tempo di risposta, da non confondere con il tempo di scansione, è definito, invece, come il massimo intervallo di tempo che intercorre tra il rilevamento di un certo evento all'esecuzione dell'azione di risposta programmata, tenendo conto dei ritardi introdotti dai moduli I/O. Il sistema operativo di un PLC è formato da un insieme di programmi di supervisione che costituisco una parte fondamentale del PLC stesso e sono destinati al controllo delle attività, alla comunicazione, all'elaborazione dei programmi utente, ecc. Tra queste, bisogna ricordare la funzione di diagnostica interna: watchdog timer. Si tratta di un temporizzatore che controlla il tempo di esecuzione di una funzionalità e genera un errore nel caso ci sia qualcosa di strano. Così facendo è possibile verificare il corretto funzionamento di un sistema. Le modalità operative di un PLC si distinguono in:

- *modalità di esecuzione*, nelle quali avviene l'aggiornamento degli ingressi e delle uscite e l'esecuzione del programma;
- *modalità di validazione*, nelle quali avviene l'esecuzione del programma senza aggiornare le uscite; ciò definisce l'affidabilità del codice senza preoccuparsi delle conseguenze di errori di programmazione;
- *modalità di programmazione*, la quale risulta utile nel momento in cui bisogna caricare sul PLC il programma sviluppato dall'utente.

#### 2.4.2 PLC: Memoria

La memoria di un PLC è generalmente divisa per aree:

- area sistema operativo, destinata alla memorizzazione permanente dei programmi del sistema operativo. Si tratta di un memoria di sola lettura,nonché di tipo ROM (Read Only Memory);
- area di lavoro del sistema operativo, destinata alla memorizzazione dei dati intermedi dei programmi del sistema operativo e che ne acconsente la lettura e la scrittura, ovvero si tratta di una memoria di tipo RAM (Random Access Memory);
- *area ingressi/uscite*, destinata alla memorizzazione delle informazioni degli stati relativi agli ingressi e le uscite (RAM);
- area programmi utente, destinata alla memorizzazione dei programmi riportati dall'utente. Tale memoria deve essere di tipo RAM durante la stesura del codice del programma, ma può essere sostituita da una memoria programmabile a sola lettura (PROM) nel momento in cui il programma scritto sia sta scritto e verificato;
- area dati utente, destinata alla memorizzazione delle informazioni dei programmi utente (RAM);

Per evitare perdite di dati, le memorie RAM possono appoggiarsi a delle batterie tampone in caso emergenza ed assenza di alimentazione. La tecnologia adoperata per realizzare tali aree è la CMOS per limitarne il consumo. Per quanto concerne le dimensioni di memoria variano da mezzo Kiloword a qualche centinaio di Kiloword (1 Kiloword = 1024 parole) in base alla tipologia del PLC. Infine, c'è anche la possibilità di espandere la memoria mediante moduli ad hoc o schede standard come Secure Digital oppure Compact Flash.

#### 2.4.3 PLC: Modulo di alimentazione

Il modulo di alimentazione, mediante il rack, fornisce l'alimentazione necessaria al funzionamento di tutti gli altri moduli, garantendo una tensione priva di fluttuazioni e stabilizzata e una adeguata protezione rispetto a cortocircuiti.



Figura 2.8: Modulo di alimentazione Siemens [20]

Per erogare maggiore potenza è possibile anche mettere in parallelo alcuni moduli di alimentazione, garantendo anche una maggiore sicurezza in caso di guasti. Essendo molti i moduli da alimentare, il dimensionamento di tale modulo deve essere realizzato in maniera opportuna andando a considerare la somma delle potenze richieste da tutti i moduli del PLC, aumentata inoltre di una certa percentuale in caso di espansioni future. A livello strutturale, invece, il modulo alimentatore è composto da un trasformatore, da un circuito rettificatore, da un filtro, da un circuito stabilizzatore e da un circuito per la protezione da sovracorrenti o cortocircuiti.

#### 2.4.4 PLC: Moduli di Ingresso/Uscita

I moduli di ingresso e uscita, i quali possono essere sia digitali sia analogici, rappresentano l'interfaccia tra la logica interna del PLC e i segnali provenienti dall'ambiente esterno (ricezione dati dai sensori e comando azioni degli attuatori). Tali schede devono attuare, in chiave elettrica, l'adattamento e il condizionamento tra livelli di tensione e corrente del PLC e quelle utilizzate per trasmettere i segnali: all'interno del PLC generalmente si lavora in bassa potenza, diversamente dall'esterno dove si lavora a potenza nettamente superiori. Dunque, i moduli I/O permettono una connessione diretta tra PLC e dispositivi presenti sul campo (sensori e attuatori) senza preoccuparsi di problematiche relative al condizionamento e messa in scala dei segnali. Tali moduli sono anche dotati di uno stadio di isolamento galvanico composto da foto-accoppiatori e trasformatori, i quali si trovano tra il campo, il bus e i circuiti interni del PLC. Come è stato specificato in questo capitolo, i moduli di ingresso e di uscita sono alloggiati in punti ben precisi all'interno dell'armadio, il quale ne può contenere solo un determinato numero. La posizione destinata a tali moduli non può essere confusa grazie a particolari accorgimenti meccanici. Inoltre, in base alla posizione assunta dal modulo I/O è possibile determinare il loro indirizzamento a livello software, consentendo al softwarista di conoscere gli indirizzi di memoria ai quali sono correlati i dati di un determinato sensore. I moduli di ingresso ed uscita si possono distinguere in base al

numero di I/O che contengono, in base alla tipologia (se si tratta di segnali digitali o analogici) e infine anche in base alla tipologia di attuatore o sensore al quale devono essere collegati. Sul PLC sono presenti anche dei LED i quali mostrano la presenza o meno del segnale dal campo. A questo punto si vada a caratterizzare le due tipologie di segnali principali:

- i moduli di I/O digitali sono tipicamente realizzati con circuiti di filtraggio per rumori o rimbalzi e con indicatori di stato sia prima che dopo l'isolamento. I Datasheet di questi moduli devono contenere informazioni riguardo il numero di input e output presenti, tensioni di riferimento, e ritardo dei segnali: le tensioni di riferimento generalmente per gli stati ON/OFF sono 0-24 Volt oppure 0-220 Volt in corrente continua o alternata, 0-5 Volt in corrente continua oppure 0-50 Volt in corrente continua. Per quanto concerne le uscite, esse sono create con transistori (corrente continua), TRIAC o SCR (corrente alternata) o relè (corrente continua e alternata). Per la prima tipologia di uscite, ovvero quella dei transistori, deve essere specificato se la corrente è fornita dal carico o assorbita da esso per un corretto funzionamento. Questi segnali possono assumere solamente due valori corrispondenti a due stati logici contrapposti che possono essere ad esempio acceso/spento, chiuso/aperto, ecc.;
- i moduli di ingresso/uscita analogici sono dei moduli che compiono conversioni digitali/analogiche o viceversa con lo scopo di interfacciare i segnali esterni analogici con quelli digitali del PLC. Questi moduli analizzano molti segnali, hanno un loro intervallo di lavoro e caratteristiche filtrati selezionabili via software; inoltre, sono caratterizzati da valori di tensione, corrente e dalla tipologia di segnali, come single-ended o differenziali, ma anche dalla risoluzione, dalla velocità di conversione e dalla rappresentazione dei dati forniti. Per ragioni economiche, sono presenti all'interno dei moduli un solo A/D o D/A con l'aggiunta di multiplexer o demultiplexer in modo da poter usare un solo convertitore per più segnali. Esattamente come i segnali digitali, anche essi possiedono degli indicatori di stato.

Con il passare degli anni, sono stati realizzati dei moduli speciali di input che si interfacciano direttamente con sensori particolari come termocoppie, celle di carico o estensimetri, i quali anni fa richiedevano circuiti elettronici particolari per la rilevazione delle informazioni.

#### 2.4.5 PLC: Terminale di programmazione

Il PLC non disponendo di tastiere e schermi per la comunicazione con il softwarista per poter essere da lui programmato, necessita di ulteriori dispositivi che consentono di fare ciò. Per i vecchi controllori a logica programmabile si era pensato un ulteriore modulo, definito terminale di programmazione o programming unit, capace di interfacciare il PLC con l'operatore in modo da garantirne la sua programmazione mediante una semplice tastiera e un display per il controllo visivo del programma (la programmazione, in questo caso, avviene all'interno della memoria del PLC). I moderni PLC sono dotati di sistemi di sviluppo per il software che garantiscono la loro esecuzione sui classici personal computer. La programmazione quindi può avvenire, anche offline, sui PC tramite questi software dedicati con la possibilità di poter simulare ciò che si è programmato. La connessione al PLC avviene attraverso una porta dedicata e tale connessione non permette solo il trasferimento del programma, ma anche funzionalità di monitoraggio durante la lettura del programma da parte del PLC per poter correggere live eventuali errori. Mediante appositi moduli, come verrà spiegato nel paragrafo successivo, è anche possibile la programmazione remota connettendo il PLC ad una rete informatica. Oggigiorno, tale funzione sta riscontrando tanto successo nel campo dell'automazione industriale grazie alla possibilità di ridurre i costi, debug e riprogrammazioni in caso di spostamenti necessari effettuati dal progettista in una fase post-programmazione.
#### 2.4.6 PLC: Moduli speciali

Nel mondo PLC esiste una vasta gamma di moduli, ognuno dei quali garantisce una certa funzione e rende il sistema PLC più flessibile e adeguato nella risposta alle diverse situazioni. Tra questi moduli speciali è possibile citare:

- il modulo di I/O remoto, il quale viene utilizzato nel caso il numero di ingressi ed uscite risulta essere molto elevato e sono disposti su una superficie molto vasta. Sfruttando questo modulo è possibile collegare gli armadi di quel impianto al PLC e inviare gli stati degli ingressi e le uscite nei rack mediante una linea seriale;
- il modulo per la connessione in rete, il quale gestisce diversi protocolli di comunicazione per le diverse tipologie di reti informatiche che possono comprendere un sistema PLC (si pensi al bus di campo oppure all' ETHERNET);
- il modulo coprocessore, il quale è un modulo che contiene al suo interno un calcolatore convenzionale che può accedere ai dati esistenti nella memoria del PLC. La nascita di questi moduli è stata dettata dalla necessità di eseguire operazioni complesse attraverso programmi scritti con linguaggi di programmazione un po' più classici, come il BASIC o il C, con lo scopo di poter comunicare direttamente con l'esterno attraverso interfacce come la seriale RS-232, la parallela Centronics o lo standard PCMCIA;
- il modulo PID, attraverso il quale è possibile avere a disposizione regolazioni PID (proporzionaleintegrativa-derivativa) a cui il PLC deve limitarsi a fornire i riferimenti. Tali moduli possono essre auto-sintonizzati e permettere un passaggio neutro tra modalità di regolazione manuale e automatica. Senza questo modulo tale regolazione non può essere eseguita in quanto il ciclo di scansione risulta essere molto lungo;
- il *modulo di servo*, il quale crea direttamente l'asservimento di uno o più motori a passo, idraulici o in corrente continua con un encoder incrementale;
- il *modulo encoder*, il quale crea tutte le funzionalità necessarie per usare uno o più encoder incrementali o assoluti. Si tratta di moduli con contatori ad alta velocità;
- il modulo interfaccia operatore, noto come Man Machine Interface (MMI), il quale rende possibile l'interfacciamento dell'operatore con il sistema PLC mediante l'utilizzo di tastiere e display. Solitamente questa funzionalità viene realizzata con calcolatori classici connessi in rete con i PLC;
- il modulo di backup, il quale viene inserito negli armadi di due PLC collegati tra di loro e che permette la funzione di backup. Il modulo processore di riserva, mediante il modulo di backup, viene aggiornato di pari passo dal modulo processore principale, eseguendo lo stesso programma e in caso di errori o malfunzionamenti, può anche sostituirlo per brevi istanti nella gestione di ingressi ed uscite. questa modalità prende il nome di sistema a massima disponibilità. Ne esiste tuttavia un'altra che prende il nome di sistema a sicurezza intrinseca, la quale prevede che i due moduli processore siano concordi nel decidere lo stato che deve assumere un'uscita prima che ciò sia eseguito

## 2.5 L'architettura software del PLC

Nel paragrafo 2.4 si è descritto tutta l'architettura hardware di un PLC. In questo paragrafo, invece, si punterà a descrivere quella che è l'architettura software, descrivendo inizialmente il

sistema operativo, introducendo anche le sue principali modalità operative, passando a descrivere il ciclo di funzionamento in real-time del PLC, giungendo, infine, ai diversi linguaggi di programmazione utilizzabili. Anche in questo paragrafo le informazioni riportate su questa tesi sono state estrapolate da diversi libri (vedere bibliografia [2] [12] [13] ).

### 2.5.1 PLC: Sistema operativo

Il sistema operativo di un PLC è costituito da un insieme di programmi i quali garantiscono il corretto funzionamento del sistema. Tali programmi sono salvati nella memoria del PLC e vengono eseguiti ogniqualvolta il dispositivo PLC viene avviato. Il sistema operativo deve supervisionare le attività del sistema complessivo, eseguire i programmi utente e gestire tutti i protocolli di comunicazione con i moduli presenti nel PLC. Altra funzione affidata al sistema operativo è quella della diagnostica che può essere sia esterna che interna: ad esempio grazie a questa funzione è possibile individuare guasti nel sistema attivando il sistema di emergenza, anticipare eventuali malfunzionamenti oppure controllare e gestire lo stato di carica delle batterie tampone per l'alimentazione delle memorie non volatili o la tensione di alimentazione. Infine, ultima funzione da ricordare, dal punto di vista software, è quella dedicata all'interfaccia con l'operatore sia in condizioni normali e sia in caso di situazioni particolari con l'attivazione di indicatori dedicati. Per quanto riguarda le modalità operative che il sistema operativo può attivare sono:

- la modalità di programmazione, mediante la quale è possibile caricare i programmi utenti nel PLC: nel caso vecchi PLC dotati di terminali di programmazione è possibile programmare direttamente sulla memoria del PLC, altrimenti nel caso di moderni PLC, mediante una porta presente sul PLC è possibile scrivere il programma su un qualsiasi PC e caricarlo sul PLC mediante la porta citata. Durante il funzionamento di questa modalità, il sistema operativo attiva la memorizzazione del codice utente nella memoria impedendo ogni altro tipo di azione;
- la *modalità di convalida e debug*, la quale si attiva immediatamente dopo la modalità di programmazione e in cui il sistema operativo consente l'esecuzione di tale programma senza attivare le uscite, nonché gli attuatori dell'impianto. In questa maniera, il sistema effettua una verifica del codice e controlla se ci sono eventuali debug;
- la *modalità di esecuzione*, la quale rappresenta la modalità di funzionamento nominale: in questa modalità il PLC esegue il programma abilitando in questo caso sensori e attuatori aggiornando ingressi ed uscite. Tale fase si attiva quando le prime due modalità sono terminate.

Per ragioni di sicurezza, per evitare conflitti , il passaggio da una modalità ad un'altra è spesso attivabile mediante una chiave hardware: in alcuni PLC infatti non è possibile eseguire in maniera software il passaggio dalla modalità di programmazione a quella di esecuzione.

## 2.5.2 PLC: Ciclo real time

Il funzionamento real time è garantito dalla modalità di funzionamento nominale ciclica, anche conosciuta come *Copia massiva di ingressi ed uscite*. Il ciclo di funzionamento a copia massiva di ingressi ed uscite è generalmente composto da quattro step principali:

• aggiornamento dell'area di memoria riservata con i valori provenienti dagli ingressi fisici (sensori);

- esecuzione dei programmi utente operando sui valori della memoria e conservando i risultati sempre in memoria;
- esecuzione di programmi di gestione del sistema (come ad esempio la diagnostica);
- scrittura sulle uscite fisiche dei loro valori conservati nell'area di memoria ad essi riservata.

Nella figura 2.9 è riportato uno schema a blocchi, ripreso dal libro *Sistemi di automazione indu*striale: architetture e controllo [2], che permette di capire in che modo avviene tale modalita di funzionamento.



Figura 2.9: Modalità di esecuzione del PLC a copia massiva di ingressi e uscite

Si può osservare che prima ancora che siano abilitate le uscite, tutto ciò viene anticipato da una fase di inizializzazione e alcune operazioni specifiche. Per quanto concerne il periodo di scansione del ciclo real time, esso rappresenta un fattore molto importante di un PLC e la sua velocità dipende dal numero di istruzioni che possono essere eseguite in tale intervallo. Data la complessità dei programmi utente, il numero di input/output e la presenza di diverse istruzioni, le quali richiedono tempi di esecuzione diversi da parte della CPU, i produttori di PLC indicano un valore medio di tempo di scansione indicato in millisecondi per kiloword di programma. Il tempo di risposta del PLC, il quale nel caso peggiore può essere al massimo due volte il tempo di scansione, deve essere sommato ai ritardi introdotti dai moduli di ingresso e uscita. Tuttavia, il tempo di scansione viene mantenuto costante dal sistema operativo grazie ad un timer noto come *watchdog timer*, il quale controlla l'esecuzione dei programmi utente, interrompendoli qualora superino il tempo di scansione massimo e generando un errore Infine, si vuole mettere in risalto che la modalità ciclica finora esposta non è l'unica funzionalità dei PLC: in casi in cui il ritardo di esecuzione non è tollerabile è possibile accedere direttamente alle aree di memoria I/O. Alcuni sistemi operativi infatti sono dotati di una gestione event driven di particolari interrupt, ciò significa che il SO può interrompere in qualsiasi momento ciò che sta elaborando il PLC per eseguire altre procedure: questa modalità solitamente viene utilizzata per la gestione delle emergenze.

## 2.6 PLC: linguaggi di programmazione

Per poter programmare un PLC, gli strumenti primari di cui ci si serve sono i linguaggi di programmazione, i quali sono stati normati nello standard IEC 61131-3 [22]. L'obiettivo di tale standard è quello di progredire verso una normalizzazione della sintassi dei linguaggi di programmazione affinché si giunga a metodi moderni per lo sviluppo di applicazioni e semplifichi la validazione dei codici. I cinque principali linguaggi di programmazione sono:

- 1. Diagramma funzionale sequenziale o anche noto come SFC (Sequential Functional Chart);
- 2. Linguaggio a contatti o anche noto come Ladder Diagram;
- 3. Diagramma a blocchi funzionali o anche noto come FBD (Function Block Diagram);
- 4. Lista di istruzioni;
- 5. Testo strutturato.

#### 2.6.1 Il linguaggio SFC

Il Diagramma funzionale sequenziale è un linguaggio grafico molto innovativo e utile soprattutto per la progettazione di sistemi di controllo. Tale linguaggio nacque in Francia da uno studio intenso fatto da una commissione la quale aveva come obiettivo quello di ricercare uno strumento per spiegare i diversi sistemi di automazioni industriale complessi in un modo semplificato. Da questo studio, ne usci vincente il *GRAPHe de Coordination Etapes Transitions* (GRAFCET), il quale nel 1987 fu standardizzato dal Comitato Elettrotecnico Internazionale e successivamente integrato tra i linguaggi di programmazione nello standard IEC 61131-3 con il nome di SFC. Il linguaggio SFC è composto da tre semplici elementi:

- la *fase*, la quale rappresenta l'azione che l'automatismo deve compiere in quell'istante;
- la transizione, la quale rappresenta la condizione da eseguire per passare alla fase successiva;
- l'arco orientato, il quale connette due fasi successive e sul quale è segnata la transizione.

Per costruire un SFC bisogna seguire due regole ben precise:

- 1. tra due fasi collegate da un arco orientato vi deve essere sempre una transizione;
- 2. tra due transizioni vi deve sempre essere almeno una fase;

Generalmente ogni fase è contraddistinta da un numero e in particolare la fase iniziale viene indicata con una doppia cornice e dal numero "0". Come è stato già detto, ad ogni fase è associata un'azione e ciò può accadere anche per le fasi iniziali. Per distinguere invece una fase attiva da una non attiva, si inserisce all'interno del quadrato un pallino: se così fosse quella fase risulta attiva e vengono compiute le azioni ad essa associate, mentre tutte le altre azioni non vengono eseguite. Inoltre, se più fasi convergono in una stessa transizione o se a una transizione succedono più fasi si inserisce una doppia linea orizzontale in modo da evidenziare la fine e l'inizio di sequenze che devono evolvere in parallelo come nella figura riportata di seguito.



Figura 2.10: Esempio di sequenze in parallelo mediante SFC [23]

Nella figura 2.11 viene riportato un esempio di automatismo semplice scritto in linguaggio SFC. Si tratta di un sistema automatico in cui è previsto un ciclo automatico e uno manuale. Sono presenti sei fasi (è possibile tuttavia passare a tre fasi contratte) e lo scopo di tale sistema è basato sulla fuoriuscita e il rientro degli steli di tre cilindri "A", "B" e "C" (il segno "+" si vuole indicare la fuoriuscita dello stelo, mentre il segno "-" il suo rientro). Tra una fase e l'altra sono presenti le transizioni, le quali in questo caso coincidono con i finecorsa dei diversi cilindri (il pedice "0" definisce lo stato di stelo rientrato, mentre il pedice "1" dello stelo fuoriuscito e che ha raggiunto la sua corsa massima). Ovviamente si tratta di un esempio abbastanza semplice, ma nella realtà si ha a che fare con sistemi di automazione industriale nettamente più complessi, come quello riportato nella figura 2.12, preso dal libro Automazione dei sistemi meccanici: corso di base (vedi [10]). In questo caso si è programmato un sistema automatico che vede come protagonista un reattore chimico. In questo esempio emerge uno dei principali vantaggi del linguaggio SFC: la possibilità di prevedere sequenze che si verificano in parallelo o anche di prevedere sequenze retro-azionate che modificano lo sviluppo delle sequenze normali.

#### 2.6.2 Il linguaggio a contatti

Si tratta del linguaggio di programmazione sicuramente più diffuso utilizzato dai controllori a logica programmabile. È anche noto con il nome inglese *Ladder Diagram*, che letteralmente si traduce come "Diagramma a scala" per la forma assunta dal programma o anche come KOP. Tale linguaggio risulta essere molto intuitivo e per questo molto usato nel campo dell'automazione, in quanto si avvicina al linguaggio utilizzato dai tecnici dell'epoca pur non avendo quasi alcuna conoscenza in campo informatico. La forma che assume il programma deriva dalla logica a relè: ci sono due linee verticali, definiti *montanti* della scala, che rappresentano l'alimentazione e delle linee orizzontali, denominati anche *pioli* della scala o *rung*, che alimentano delle bobine se la combinazione di contatti lo permette. I contatti possono essere associati a ingressi digitali provenienti dal campo o a condizioni interne al dispositivo; la bobina è solitamente associata a una uscita digitale o segnala una condizione interna. La programmazione mediante ladder diagram definisce una rappresentazione grafica simile allo schema elettrico in cui i simboli utilizzati sono riportati nell'esempio che segue estrapolato dalla guida presente sul Tia Portal (vedi [24] [25])

La corrente viene trasmessa da sinistra verso destra e i contatti sono utili a stabilire o interrompere un collegamento tra due elementi. In un programma KOP sono disponibili i seguenti contatti:

• contatto normalmente aperto, come quello associato all'ingresso "I8.3", il quale fa passare corrente se lo stato del segnale di un operando binario indicato è "1";



Figura 2.11: Esempio di programmazione mediante il linguaggio SFC



Figura 2.12: Esempio di programmazione mediante il linguaggio SFC di un reattore chimico [10]



Figura 2.13: Elementi principali utilizzati nella programmazione mediante Ladder Diagram

- contatto normalmente chiuso, come quello associato all'ingresso "I8.4", il quale fa passare corrente se lo stato del segnale di un operando binario indicato è "0";
- contatto con funzione aggiuntiva, il quale fa passare corrente se soddisfa una certa condizione. Essi permettono di eseguire una funzione aggiuntiva come una valutazione dei fronti o un confronto.

Le bobine invece comandano gli operandi binari e possono settare o resettare un operando binario a seconda del segnale di stato del risultato logico combinatorio. Esistono le seguenti tipologie di bobine:

- *bobine semplici*, come quelle associate alle uscite "Q16.0" e "Q16.1", le quali impostano un operando binario se la corrente la percorre;
- bobine con funzione aggiuntiva, come quella associata all'uscita "Q16.2", le quali oltre alla valutazione del risultato logico combinatorio presentano ulteriori funzionalità come quelle per la valutazione dei fronti o controllo;

Infine possono esserci anche elementi come i box, i quali possono essere innumerevoli capaci di eseguire diverse funzioni. Tra i principali box bisogna ricordare quelli che soddisfano le funzioni di temporizzazione (TON,TOF, TONR), di conteggio (CTU,CTD,CTUD), di controllo del programma(JMP,LBL,JSR,SBR), di manipolazione dati (MOVE) e realizzazione di funzioni speciali (PID,SEND,GET).

#### 2.6.3 Il linguaggio a blocchi funzionali

Il linguaggio a blocchi funzionali, noto anche con l'acronimo tedesco FUP, usa una rappresentazione delle istruzioni nella quale i compiti dell'automazione sono espressi in forma grafica a blocchi con i simboli conformi alle norme DIN 40700 e DIN 40719. Tale linguaggio viene solitamente utilizzato quando la logica combinatoria raggiunge una certa complessità e in particolare si serve, per semplificare la programmazione, di "schemi funzionali". Esso tratta, esattamente come altri linguaggi di programmazione, la logica binaria e la sua simbologia deriva dalla progettazione di circuiti elettronici. Analogamente al linguaggio a contatti, lo schema funzionale è ruotato di 90 gradi e lo schema viene letto sempre da sinistra verso destra. Di seguito viene riportato un esempio di programmazione in linguaggio FUP preso dal manuale di riferimento della Siemens Schema logico (FUP) per S7-300/400 [26].



Figura 2.14: Esempio di programmazione mediante linguaggio a schema funzionale

Lo stato di segnale della bobina "A4.0" viene impostato sullo stato logico "1" quando:

- le uscite" E0.0" AND "E0.1" hanno lo stato di segnale "1";
- OR lo stato di segnale dell'uscita "E0.2" è pari a "0".

#### 2.6.4 La lista di istruzioni

Si tratta di un linguaggio di programmazione strettamente testuale molto simile a quei linguaggi assemblativi dei tradizionali calcolatori. Esso è anche noto con l'acronimo tedesco AWL e, pur assomigliando a un linguaggio mnemonico molto vicino al codice macchina, mostra istruzioni e macroistruzioni potentissimi che permettono di fare della programmazione strutturata sfruttando architetture di programmazione caratteristiche di linguaggi di alto livello. Viene spesso utilizzato questo tipo di linguaggio in quanto risolve problemi di automazione in maniera modulare, ottimizzando i programmi e semplificando lo sviluppo, i debug e la documentazione del software. Inoltre, c'è la possibilità di utilizzare delle memorie ausiliarie, definite *Merker*, per registrare al loro interno dei risultati parziali che volendo posso essere utilizzati in più posti della logica. Nella figura 2.15 viene riportato un esempio di linguaggio AWL estrapolato dal manuale di riferimento della Siemens *Schema logico (AWL) per S7-300/400* [27]

Program	ma AWL	Schema logico di relè
		Sbarra collettrice
300×	E 0.0 M 10.0	E 0.0 M 10.0
¥00)	E 0.2 M 10.3	E 0.2 M 10.3
U	M 10.1	M 10.1
=	A4.0	A 4.0 Bobina

Figura 2.15: Esempio di programmazione mediante linguaggio AWL

#### 2.6.5 Il testo strutturato

Questo linguaggio è il meno utilizzato nel campo dell'automazione industriale. Si tratta di un linguaggio di programmazione testuale ad alto livello, simile al PASCAL o al C, con adattamenti all'utilizzo nei sistemi di controllo. Nella figura 2.16 viene riportato un esempio di programmazione mediante l'uso di questo linguaggio, il quale è stato preso dal libro PLC e automazione industriale [13].

```
if (livello < livello_max)
then
    valvola1 = true;
else
    allarme = true; valvola1 = false;
end_if;
aux = i25*10 - 4;</pre>
```

Figura 2.16: Esempio di programmazione mediante testo strutturato

## Capitolo 3

# SIMATIC STEP 7: TIA Portal

## 3.1 SIMATIC STEP 7

Come ultimo step, prima di poter entrare nel vivo di questa tesi, si è pensato di introdurre il pacchetto software per lo sviluppo dei progetti di automazione noto come SIMATIC STEP 7. Tale pacchetto utilizza al suo interno tutti i prodotti Siemens della serie SIMATIC e l'ultima versione a cui si ha accesso è la SIMATIC STEP 7 V16, ovvero quella utilizzata per la programmazione di semplici automatismi in questa tesi. Con il termine *TIA Portal* si vuole intendere l'ambiente di progettazione centralizzato in cui lavora STEP 7. TIA Portal è l'acronimo inglese di *Totally Integrated Automation Portal* ed è anche definito come una *piattaforma di engineering* per tutti i campi che riguardano l'automazione. Esso è caratterizzato da un'interfaccia utente uguale per tutti i campi dell'automazione che mette a disposizione diversi servizi condivisi (come quelli di configurazione, comunicazione o diagnostica) e un solo database al quale accedono altri pacchetti software come ad esempio SIMATIC WinCC V16, SINAMICS Startdrive o SIMATIC STEP 7 PLCSIM V16. Tuttavia, del SIMATIC STEP 7 V16 sono presenti due versioni:

- *Basic*, adatta per progettare con il modello Siemens S7-1200;
- Professional, adatta per progettare con il modello Siemens S7-300, S7-400 e S7-1500;

## 3.2 Interfaccia utente TIA Portal V16

Nell'interfaccia utente del TIA Portal è possibile evidenziare due viste principali denominate vista portale e vista progetto. La vista portale, indicata nella figura 3.1, è quella che si apre in maniera automatica quando si cerca di aprire il TIA Portal e che dà la possibilità all'utente di scegliere quali operazioni vuole svolgere con l'ambiente TIA Portal. È principalmente composta da due finestre e un tasto posto in basso a sinistra: il menu posto a sinistra (indicato con il numeretto "1") consente di scegliere quale attività si vuole svolgere sul TIA Portal, la finestra di destra (indicata con il numeretto "2") si apre in base all'operazione scelta nel menu precedente e rappresenta una finestra di selezione e infine, il tasto indicato con il numeretto "3" consente il passaggio alla vista progetto.

La vista progetto, riportata nella figura 3.2, rappresenta la finestra di lavoro del TIA Portal. Essa consente lo svolgimento di qualsiasi operazione all'interno del progetto e si compone di diverse sotto-finestre:

• la finestra contrassegnata con il numero "4" consente di visualizzare e interagire con tutti i componenti presenti nel progetto;







Figura 3.2: Vista progetto presente nel TIA Portal

- la finestra contrassegnata con il numero "5" permette di visualizzare i dettagli di un particolare componente se selezionato nella finestra precedente;
- la finestra contrassegnata con il numero "6" rappresenta la vera finestra di lavoro dell'utente, all'interno della quale può apportare modifiche al progetto, programmare un PLC o creare nuove pagine per il touch panel con animazioni o tasti interattivi;
- la finestra contrassegnata con il numero "7" consente di visualizzare le proprietà, le informazioni o la diagnostica del componente selezionato modificandone le impostazioni;
- la finestra contrassegnata con il numero "8" cambia in base all'editor che viene riportato nella finestra di lavoro 6, consentendo di visualizzare il menu *Biblioteche* del TIA Portal.

## 3.3 Gestione dei progetti sul TIA Portal

Nell'ambiente TIA Portal è possibile sia creare un nuovo progetto che utilizzare progetti già creati anche con ambienti di programmazione Siemens non nuovi. Inoltre c'è la possibilità di salvare progetti, archiviarli in un file di tipo .zap16 e aprirli senza editarli. Dalla vista portale è possibile creare un nuovo progetto selezionando inizialmente il tasto Avvia e poi Crea nuovo progetto come mostrato nella figura 3.3. A questo punto si apre una nuova finestra dove va inserito il nome del progetto che si vuole creare, il percorso file dove si crea il nuovo progetto, l'autore del progetto e eventualmente un commento.

₩a Si	emens					_ # X
						Totally Integrated Automation PORTAL
Ċ	vvia			Crea nuovo progetto		
			Apri progetto esistente	Nome del progetto:	Esempio di nuovo progetto	
		- A.	Apri progetto esistente	Percorso:	C:\Users\michele\Desktop\Tesi Magistrale	
			Crea nuovo progetto	Versione:	V16	
			Migrazione progetto	Autore:	michele	
				Commento:		<u>^</u>
		-105	Chiudi progetto			
					<u>.</u>	Ľ
			Welcome Tour			Crea
	Online & Diagnostica	10				

Figura 3.3: Creazione di un nuovo progetto sul TIA Portal su vista portale

È possibile creare anche un nuovo progetto dalla vista progetto selezionando *Progetto* dalla barra dei menu in alto e selezionare la voce *Nuovo*.

3	🏠 Sieme	ns					_				
	Progetto	Modifica	Visualizza	Inserisci	Online	Strumenti	Tool	Fin	estr	а	?
Γ	脊 <u>N</u> uovo.				2	) ± ( <sup>24</sup> ± 1	ih 10	16			6
	📑 <u>A</u> pri			Ct	trl+O				-		
	Migrazi	ione proget	tto								
	<u>C</u> hiudi			Ct	trl+W						
l	Elimina	a progetto		C	trl+E						

Figura 3.4: Creazione di un nuovo progetto sul TIA Portal su vista progetto

Per salvare un progetto si deve selezionare dalla vista progetto sempre la voce *Progetto* dalla barra dei menu in alto e selezionare la voce *Salva* o *Salva con nome*.

1	Siemer	ns - C:\Us	ers\mikel\D	esktop\Pro	getto Te	esi Magistral	e PID\P	ID\Progetto Tes ool Finestra II II II II II II istrale PID >	
	Progetto	Modifica	Visualizza	Inserisci	Online	Strumenti	Tool	Finest	ra
	Muovo.					<mark>ર (</mark> #± િ		<b>近</b> 및	BT
1	<u>A</u> pri			Ct	rl+0	netto Tesi M	baietr		
	Migrazi	one proget	to		<u> </u>	getto resi w	agisti		
	Chiudi			Ct	rl+W				
l	Elimina	progetto	• 2	C	trl+E	.X =5 =5			- 6
	📘 <u>S</u> alva			C	trl+S	for =, =,	-9 j =		
	Salva c	on no <u>m</u> e		Ctrl+Maiu:	sc+S				
	Archivia	a			-	connicitto			

Figura 3.5: Salvataggio di un progetto sul TIA Portal su vista progetto

Per quanto riguarda l'archiviazione di un progetto, si deve selezionare sempre la voce *Progetto* dalla barra dei menu in alto a sinistra e subito dopo la voce *Archivia*: in questa maniera verrà creato un file con un estensione .zap16 il quale contiene gli stessi dati del progetto riorganizzati in modo tale da compattare il progetto stesso. Al contrario, se si vuole aprire un progetto precedentemente archiviato si deve selezionare dal menu degli strumenti in alto a sinistra la voce *Progetto* e poi *Disarchivia*.

Vîà	Siemer	ns - C:\Us	ers\mikel\D	esktop\Pro	getto Tes	i Magistral	e PID\F	Prog	etto	Tes
Pro	ogetto	Modifica	Visualizza	Inserisci	Online	Strumenti	Tool	Fir	estr	а
1 ar	Nuovo.				5	) ± ( <sup>24</sup> ± 1	1	16		
- 🖻	Apri			Ct	rl+0					
	Migrazi	one proget	to							
1	Chiudi			Ct	rl+W					
	Elimina	progetto	•	C	trl+E					
	<u>S</u> alva			C	trl+S					
	Salva c	on no <u>m</u> e		Ctrl+Maius	c+S					
	Archivia	a								

Figura 3.6: Archiviazione/Disarchiviazione di un progetto sul TIA Portal su vista progetto

Inoltre, dallo stesso menu a tendina, ovvero dalla voce *Progetto*, è possibile aprire un progetto presente sul PC e modificarlo mediante la voce *Apri*, chiuderlo mediante la voce *Chiudi* oppure eliminarlo mediante la voce *Elimina*.

	VA Siemen	is - C:\Us	ers\mikel\D	esktop\Pro	getto Te	si Magistrale	e PID\F	rogetto	Tes
	Progetto	Modifica	Visualizza	Inserisci	Online	Strumenti	Tool	Finest	ra
	Muovo				iic.	) ± (* ± 1	h ID	旧里	B
9	<u>A</u> pri			Ct	rl+0				
	Migrazio	one proget	tto						
	<u>C</u> hiudi			Ct	rl+W				
	Elimina	progetto		C	trl+E				

Figura 3.7: Apertura, chiusura e eliminazione di un progetto sul TIA Portal su vista progetto

Se invece si vuole aprire un progetto esistente in modalità non editabile senza chiudere il progetto su cui si sta lavorando, bisogna selezionare la voce *Visualizza* dalla barra dei menu e poi *Progetti di riferimento*, i quali vengono riportati su sfondo grigio perché non possono essere appunto editati. Facendo "drag&drop" è possibile copiare oggetti e trasferirli nel progetto che sta scrivendo da quello non editabile: in questa maniera si è sicuri di non modificare il vecchio programma.

VA Siemens - C:\Use	ers\mikel\Desktop\Progetto Te	si Magistrale PID\Progetto Tes
Progetto Modifica	Visualizza Inserisci Online	Strumenti Tool Finestra
📑 📑 🖬 Salva prog	Vai a vista <u>p</u> ortale	Alt+F7 🗟 🛄 🖬 🖳
Navigazione de	Navigazione del progetto	Ctrl+1
	Vista generale	Ctrl+2
Dispositivi	Task <u>c</u> ard	Ctrl+3
<b>P</b>	Vista <u>d</u> ettagli	Ctrl+4
	<u>Finestra di ispezione</u>	Ctrl+5
T Progetto Ter	🥅 Progetti di <u>r</u> iferimento	Ctrl+9
Aggiungi	Tastiera su sc <u>h</u> ermo Ctrl+N	Maiusc+K

Figura 3.8: Apertura di un progetto non editabile sul TIA Portal su vista progetto

Infine, se ci dovessero problemi di qualsiasi tipo, è consigliato visionare la guida offerta dal TIA Portal [24]. Per poterla vedere bisogna andare dalla barra dei menu in alto e selezionare la voce "?" come indicato nella figura 3.9 e a quel punto selezionare la voce Visualizza guida.

Vi Siemens - C	:\Users\mikel\Des	ktop\Tesi M	lagistrale\Progette	006 ge	stione HMI	\Proge	etto06 gestione HMI		
Progetto Modifi	ca Visualizza I progetto 进 🕽	nserisci O	nline Strumenti X 🔊 ± (य ±	Tool	Finestra	? <b>?</b> ⊻is	ualizza Guida		e
Navigazione	e del progetto					Se	rvice & Support in Internet rvice & Support in <u>T</u> IA Portal	Ctrl+Maiusc+7	
Dispositiv						<u>S</u> o	ftware installato		
遊									

Figura 3.9: Procedura per visualizzare la guida del TIA Portal su vista progetto

## 3.4 Configurazione di un nuovo dispositivo

Si passi a questo punto a definire i primi passi che bisogna effettuare quando si vuole creare un nuovo progetto. Come prima cosa si deve creare il progetto e lo si fa seguendo gli step descritti nel paragrafo precedente definendone il nome e il percorso file. A questo punto ci si troverà in una finestra simile a quella riportata nella figura 3.10 in cui c'è la possibilità di:

- *Configurare un dispositivo*, mediante il quale è possibile inserire all'interno del progetto tutti i dispositivi presenti sul campo, come ad esempio PLC, HMI, SCALANCE e altri prodotti Siemens;
- *Scrivere programma per PLC*, dove appunto si vuole scrivere il programma che il PLC deve eseguire;
- *Configurare oggetto tecnologico*, nel quale è possibile inserire un qualsiasi oggetto tecnologico all'interno del progetto, come ad esempio il PID o SINAMICS;

• *Progettare una pagina HMI*, dove è possibile inserire e personalizzare le pagine che verranno in seguito caricate sul touch panel (mezzo con cui si interfaccia l'operatore);

Avvio			
<b>→</b>	\$ <sup>\$</sup>	Configura un dispositivo	
→	٢	Scrivi programma per PLC	
→	-00-	Configura oggetto tecnologico	
→	Ø	Progetta una pagina HMI	
$\rightarrow$		Apri vista progetto	

Figura 3.10: Primi passi da eseguire con il TIA Portal

Si parta dalla selezione dei dispositivi hardware presenti sul campo da configurare. Quasi sicuramente bisognerà configurare un PLC, il quale risulta vitale per l'automatismo che si vuole creare.

Aggiungi nuovo dispo	sitivo		
Controllori HM Sistemi PC	<ul> <li>Controllori</li> <li>SIMATIC 57-1200</li> <li>CPU</li> <li>CPU</li> <li>CPU 1211C ACIDC/Rly</li> <li>CPU 1211C DCIDCIDC</li> <li>CPU 1211C DCIDC/Rly</li> <li>CPU 1212C DCIDC/RLY</li> <li>CPU 1212C DCIDC/RLY</li> <li>CPU 1212C DCIDC/RLY</li> <li>CPU 1212C DCIDC/RLY</li> <li>CPU 1214C DCIDC/RLY</li> <li>CPU 1215C DCIDC/RLY</li> <li>CPU 1215C DCIDC/RLY</li> <li>CPU 1215C DCIDC/RLY</li> <li>CPU 1215C DCIDC/RLY</li> <li>CPU 1212C DCIDC/DC</li> <li>CPU 1212C DCIDC/DC</li> <li>CPU 1212C DCIDC/RLY</li> <li>CPU 1212C DCIDC/RLY</li> <li>CPU 1212C DCIDC/RLY</li> <li>CPU 1214C DCIDC/DC</li> <li>CPU 1214C DCIDC/DC</li> <li>CPU 1214C DCIDC/DC</li> <li>CPU 1214C DCIDC/RLY</li> <li>CPU 1214C DCIDC/DC</li> <li>CPU 1214C DCIDC/DC</li> <li>CPU 1214C DCIDC/RLY</li> <li>CPU 1214C DCIDC/RLY</li> <li>CPU 1214C DCIDC/RLY</li> <li>CPU 1214C DCIDC/DC</li> </ul>		Dispositivo: N° di articolo: Versione: Descrizione: SIMATIC S7-1200
	CPU 1215FC DC/DC/Rly      The CPU SIPLUS      The CPU 1200 non specificata		
	CPU 1200 SIPLUS non sp     Im SIMATIC S7-1500	~	

Figura 3.11: Configurazione di un nuovo controllore

Per cui, cliccando sul sezione *Configura un dispositivo*, si procede con la selezione del dispositivo che si intende configurare, che in questo caso coincide con un controllore, e infine bisogna selezionare la tipologia di CPU e la sua versione. In particolare, sulla scatola del PLC sono fornite tutte le informazioni utili per poterlo configurare nell'ambiente TIA Portal sul suo lato destro come mostrato nella figura 3.12.



Figura 3.12: Vista del PLC dalla quale è possibile carpire informazioni utili alla configurazione

Ad esempio, in questo caso si tratta di un controllore SIMATIC S7-1200, la cui CPU è la 1214C DC/DC/DC con il numero di identificazione "6ES7 214-1AG40-0XB0". A questo punto, dalla finestra relativa alla configurazione del controllore, si va a selezionare quella CPU come mostrato nella figura 3.13.



Figura 3.13: Selezione e inserimento della CPU nel progetto sul TIA Portal

È possibile notare che una volta selezionato il PLC che si vuole utilizzare nel progetto, sulla parte a destra della selezione sono presenti tutti i dettagli della CPU scelta. Tuttavia, potrebbe capitare che al momento di creazione del progetto non si abbia a disposizione il PLC fisico o il codice di riferimento oppure che il dispositivo configurato sul TIA Portal, nonostante coincida con il PLC fisico, durante il collegamento tramite cavo PROFINET, non lo riconosca e dia degli errori (Firmware o di altro genere). Se ci si dovesse trovare in una delle due condizioni si consiglia di configurare all'interno del TIA Portal, supponendo di voler installare una 1200, una *CPU 1200 non specificata* come è visibile nella figura 3.14. Si può osservare che in questo caso il codice identificativo della CPU è il seguente: "6ES7 2XX-XXXX-XXXX". Questo perché nel momento in cui si andrà a fare la configurazione di un controllore fisico, al posto delle "X", grazie ad una funzione interna di riconoscimento del TIA Portal e allo scambio di informazioni attraverso cavo PROFINET, verranno inserite cifre alfa-numeriche che andranno a comporre il codice del PLC collegato (il collegamento di un dispositivo fisico verrà trattato nel paragrafo successivo).



Figura 3.14: Selezione e inserimento della "CPU non specificata" nel progetto sul TIA Portal

Nella stessa maniera è possibile configurare un pannello operatore (HMI) o anche noto come touch panel. In questo caso bisogna andare a selezionare la sezione HMI come mostrato nella figura 3.15 e selezionare l'HMI la tipologia e il codice identificativo dello stesso.

	🗕 🧮 HMI	Dispositivo:	
	➡ Image: SIMATIC Basic Panel		
	🕨 🔚 3'' Display		
Controllori	🕨 🛅 4'' Display		
controllon	6" Display		
	🕨 🕞 7'' Display		
	🕨 🛅 9'' Display		
	🕨 🔚 10'' Display		
	🕨 🛅 12'' Display	N° di articolo:	
нм	🕨 🛅 15'' Display	Versione:	
	SIMATIC Comfort Panel		
	SIMATIC Mobile Panel	Descrizione:	
	HMLSIPLUS	7" Display	

Figura 3.15: Configurazione di un nuovo HMI

Per conoscere il codice del pannello operatore, anche in questo caso, esso si trova sul retro dello stesso come mostrato nella figura 3.16.



Figura 3.16: Retro del HMI dove è possibile carpire informazioni utili alla configurazione

Ad esempio, il touch panel che si vuole configurare in questo caso è un SIMATIC Basic Panel, 7" Display, il cui codice identificativo è "6AV2 123-2GB03-0AX0" come riportato nella figura 3.17. Anche in questo caso sono presenti ulteriori dettagli sulla parte destra del HMI.

	<b>▼ 🛅 HM</b>	Dispositivo:	
	🕶 🛅 SIMATIC Basic Panel		
	🕨 🛅 3'' Display	and a second second	
Controllori	🕨 🛅 4'' Display		
	🕨 🛅 6'' Display		
	🕶 🛅 7'' Display	KTP700 Basic PN	
	🕶 🛅 KTP700 Basic	KII YOO BUSICI IN	
	6AV2 123-2GA03-0AX0	NR disertestes CAVO 100 0000 0000	-
	6AV2 123-2GB03-0AX0	Nº di articolo: 6AV2 123-2GB03-0AX0	
HMI	🕨 🫅 KTP700 Basic Portrait	Versione: 16.0.0.0	Tr
	🕨 🛅 9'' Display		
	🕨 🛅 10'' Display	Descrizione:	
	🕨 🛅 12'' Display	Display da 7", 800 x 480 pixel, Colori 64K;	
	🕨 🛅 15'' Display	PROFINET, 1 x USB	
	SIMATIC Comfort Panel		
Sistemi PC	SIMATIC Mobile Panel		-

Figura 3.17: Selezione e inserimento del HMI nel progetto sul TIA Portal

Una volta inserito il pannello operatore si aprirà una nuova finestra, come quella riportata nella figura 3.18, nella quale si potrà personalizzare a proprio piacimento il pannello operatore, selezionando prima a quale PLC bisogna collegarlo e poi definire come rappresentare le pagine dell'HMI, come indicare le segnalazioni, come impostare il numero di pagine da visualizzare e le pagine di sistema e infine come selezionare i pulsanti.

È possibile anche configurare un sistema PC selezionando l'ultima voce della sezione Aggiungi nuovo dispositivo, ma nel caso in esame, dovendo lavorare per la costruzione di un banco didattico, non servirà installare nessun sistema PC.

Assistente per il pannello o	peratore: KTP700 Basic PN Collegamenti PLC Confi	gurare i collegamenti FLC	×
Collegamenti PLC Rappresentazione pagine Segnalazioni Pagine Pagine di sistema Pulsanti	HML2 KTP700 Basic+ PN	Driver di comunicazione : -PLC> Interfaccia : V	Seleziona PLC Sfoglia
🛃 Salva impostazioni		≪Indietro Avanti≫	<u>⊆</u> ompleta A <u>n</u> nulla

Figura 3.18: Personalizzazione del pannello operatore configurato sul TIA Portal

Dunque, per automatismi semplici didattici, Controllori e HMI rappresentano i due principali dispositivi che bisogna configurare senza i quali non è possibile far funzionare il sistema.

## 3.4.1 Collegamento e riconoscimento dei dispositivi fisici

Dopo aver configurato i dispositivi nel modo in cui è stato spiegato nel paragrafo precedente, nel caso si hanno a disposizione i dispositivi fisicamente, è possibile collegare e riconoscere tali dispositivi. Tuttavia, prima di procedere al collegamento diretto con i dispositivi fisici, bisogna assegnare ad ogni dispositivo un indirizzo IP. Solitamente, quando si configura un dispositivo, il sistema TIA Portal procede all'assegnazione automatica dell'indirizzo IP, il quale è possibile visionarlo cliccando su *Dispositivi & Reti* dal menu a tendina presente sulla sinistra del TIA Portal: infatti così facendo si aprirà una finestra che mostra la *vista di rete* con tutti i dispositivi configurati e cliccando sul tasto evidenziato nella figura 3.19 sarà possibile vedere gli indirizzi di tutti i dispositivi. Gli indirizzi IP serviranno al collegamento della parte hardware presente sul TIA Portal con la parte hardware presente sul campo.



Figura 3.19: Visualizzazione degli indirizzi IP dei dispositivi configurati

Andando, invece, su *vista dispositivi* e selezionando un qualsiasi dispositivo è possibile visualizzare tutte le proprietà che lo contraddistinguono.

Navigazione del progetto 🛛 🔳 📢	Progetto06 gest	tione HN	¶ → PL	C_1 [C	PU 1214C	DC/DC/DC]								_		×
Dispositivi	dt PLC_1 (CPU 1	1214C]		•	E 4	11 Q ±	Vi	sta top	ologica	1 ato 1	/ista di	rete (1	Vist	ta dispo	sitivi	ז
de Dispositivi 8 Reti	Telaio di montagg	103	102	101	80403      	1 ant row Find	2	3	4	5	6	7	8	9		anisodsia isa nan
Gackup online	< II								> 1	00%			-		ě	
Figure Traces	PLC_1 [CPU 121	4C DC/D	C/DC]			(	🔍 Prop	rietà	1 Inf	ormazi	oni 追	😵 Dia	gnostic	a		-
Comunicazione ore ox     Sati proxy dei dispositivi	Generale	Variabi	le IO	Cos	tanti di si	stema Testi		6								
<ul> <li>Informazioni sul progra</li> <li>Elenchi di testi di segnal</li> <li>Moduli locali</li> <li>HML 1 (KTP200 Basis PM)</li> </ul>	Generale Interfaccia PROF DI 14/DQ 10 Al 2	INET (X1)		Ger Info	nerale rma <mark>zi</mark> oni s	ul progetto		G	2							* II
Growith non aggruppat     Growith non aggruppat     Growith non aggruppat     Growith is security     Growith is immitial PLC     Growith aggruppation is a security     Growith aggruppation is a security     Growith aggruppation is a security	<ul> <li>Contatori veloci</li> <li>Generatori di im Avviamento Ciclo Carico di comun Merker di clock e</li> <li>Senver web</li> </ul>	(HSC) pulsi (PT icazione e di sisten				Nome: Autore: Commento:	PLC_1 miche	le								

Figura 3.20: Vista dispositivi e elenco proprietà dei dispositivi

In particolare, dalla lista proprietà è possibile modificare l'indirizzo IP del dispositivo scelto: si seleziona Interfaccia PROFINET [X1] e di seguito Indirizzi Ethernet. Nella sezione Protocollo IP è possibile cambiare l'indirizzo IP ricordando che in fase di collegamento deve coincidere con il dispositivo fisico.



Figura 3.21: Modifica dell'indirizzo IP di un dispositivo

Ultimo accorgimento prima di effettuare il riconoscimento del dispositivo è una procedura da eseguire sul proprio PC. In particolare, bisogna accedere al *Pannello di controllo* del PC, andare su *Reti e Internet* e infine su *Connessioni di rete*.



Figura 3.22: Procedura per permettere il collegamento mediante cavo ETHERNET, fase 1

A questo punto, cliccando su *Ethernet* e premendo il tasto destro bisogna selezionare *Proprietà*.

- > × <b>↑</b>	🔮 > Pannello di controllo >	Rete e Internet > Connessioni di rete		
Drganizza 🔻	Disabilita dispositivo di rete	Esegui diagnosi della connessione	Rinomina connessione	Cambia impostazioni connession
Cavo R	et 2 di rete scollegato Disabilita Stato	Wi-Fi polito 2 ntel(R) Dual Band Wireless-A	C 31	
•	Connessioni con bridging			
	Elimina Rinomina			
	Proprietà			

Figura 3.23: Procedura per permettere il collegamento mediante cavo ETHERNET, fase 2

Dopodichè, bisogna selezionare nella finestra che si apre la voce Protocollo Internet versione 4 (TCP/IPv4) e cliccare su Proprietà.

Rete	Condivisione			
~				
Conn	etti tramite:			
7	Realtek PCIe GE	BE Family Controller #2		
			Configura	
Laco	nnessione utilizza	ali elementi seguenti:		
	Client per reti l	Microsoft		^
	Condivisione f	ile e stampanti per reti Mir	crosoft	
	Utilità di pianifi	cazione pacchetti QoS	brobbit	
	Protocollo Inte	met versione 4 (TCP/IP)	(4)	
	Protocollo Mic	rosoft Network Adapter N	Aultiplexor	
	PROFINET IC	protocol (DCP/LLDP)		
	Driver protoco	llo LLDP Microsoft		~
<			>	0
	Installa	Disinstalla	Proprietà	
De	scrizione			
т	P/IP. Protocollo p	redefinito per le WAN ch	e permette la	
CO	municazione tra di	verse reti interconnesse.		

Figura 3.24: Procedura per permettere il collegamento mediante cavo ETHERNET, fase 3

Infine, bisogna inserire, se non è già selezionato, il nuovo indirizzo IP, che nel caso in esame,"192.168.0.99" e il subnet mask "255.255.255.0".

Dopo questa procedura è possibile andare avanti collegando i dispositivi fisici all'hardware configurato nel TIA Portal mediante un cavo PROFINET.

Rete	Condivisione		
Co	Proprietà - Protocollo Internet version	ne 4 (TCP/IPv4)	
g	Generale		
La	È possibile ottenere l'assegnazione au rete supporta tale caratteristica. In ca richiedere all'amministratore di rete le i	comatica delle impostazioni IP so contrario, sarà necessario mpostazioni IP corrette.	se la
	Ottieni automaticamente un indiri	zzo IP	
	• Utilizza il seguente indirizzo IP:		
	Indirizzo IP:	192.168.0.99	
6	Subnet mask:	255.255.255.0	
<	Gateway predefinito:		
- C	Ottieni indirizzo server DNS autor	naticamente	
	Utilizza i seguenti indirizzi server l	ONS:	
	Server DNS preferito:		
	Server DNS alternativo:	· · · ·	
_	Convalida impostazioni all'uscita	Avanzat	e

Figura 3.25: Procedura per permettere il collegamento mediante cavo ETHERNET, fase 4

Si supponga di scrivere il programma utente e di volerlo caricare all'interno del PLC. In primis, bisogna collegare il PC alla CPU mediante un cavo PROFINET, selezionare il PLC nell'ambiente TIA Portal che corrisponde alla CPU fisica e in seguito effettuare in ordine prima una compilazione e poi un caricamento del dispositivo rispettivamente con i tasti in alto *compila* e *carica nel dispositivo*, come mostrato nella figura 3.26.

JA Siemens - C:\Users\mikel\Desktop\Tesi M	Agistrale\Progetto06 gestione HMI\Progetto06 gestione HMI
Progetto Modifica Visualizza Inserisci Or	Online Strumenti Tool Finestra ?
📑 📑 🔚 Salva progetto 🚦 🐰 🟥 🗊 🕽	🗙 🏷 🛨 (🏝 🔛 🛐 🖳 🗖 Collega online 🖉 Interrompi collegamento online
Navigazione del progetto	Progetto06 😥 😰 M 🕨 Dispositivi & Reti
Dispositivi	🚽 Vista topol
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	Collega in rete 👖 Collegamenti Collegamento HM 💌 🕎 🖫 🗐 🔅
Progetto06 gestione HMI     Aggiungi nuovo dispositivo     Dispositivi & Reti     Dispositivi & Reti     Dispositivi Reti     Dispositivi non raggruppati     Simpostazioni Security     Mul_1 (KTP700 Basic PN)     Simpostazioni Security     Dispositivi non raggruppati     Dispositi	PLC_1 CPU 1214C HML_1 KTP700 Basic PN PN/IE_1
G Lingue & Risorse     G Lingue & Risorse     G Version Control Interface     G Accessi online     G Card Reader/memoria USB	•

Figura 3.26: Collegamento tra PC e PLC fisico

Quando viene premuto il tasto *compila* (indicato con il numeretto 2), il TIA Portal esegue un'operazione di compilazione del programma, la quale ha come obiettivo:

- ricercare gli errori di sintassi nel programma utente;
- eliminare le istruzioni non necessarie dal programma utente;

- verificare tutti i richiami dei blocchi all'interno dei blocchi compilati;
- verificare l'univocità nel programma utente dei numeri dei blocchi. Pertanto, qualora vi siano più blocchi con gli stessi numeri, durante la compilazione vengono rinumerati automaticamente i blocchi che presentano un conflitto di numeri. I conflitti di numeri che non possono essere risolti modificando automaticamente il numero devono essere corretti manualmente. In ogni caso si consiglia di osservare i relativi messaggi nella finestra di ispezione per poter successivamente correggere gli errori.

La compilazione dei blocchi mediante la navigazione del progetto offre ulteriori opzioni:

- Software (soltanto modifiche), in cui vengono compilate tutte le modifiche del programma dei blocchi selezionati oppure se è selezionata una cartella di blocchi, vengono compilate tutte le modifiche del programma dei blocchi contenuti nella cartella.
- Software (compila completamente i blocchi), dove tutti i blocchi vengono completamente compilati e questa opzione è consigliata per la prima compilazione o nel caso di modifiche consistenti.
- Software (resetta riserva di memoria), in cui tutte le variabili dichiarate nell'area di riserva delle interfacce dei blocchi selezionati, vengono spostate nel campo standard dell'interfaccia. La riserva di memoria acquisisce capacità libere per ulteriori ampliamenti delle interfacce.

Viene inoltre eseguita anche una verifica di coerenza tra le interfacce di blocchi richiamati o di tipi di dati PLC utilizzati: ad esempio potrebbero verificarsi incoerenze tra i blocchi richiamanti e quelli richiamati, oppure tra i tipi di dati PLC e i blocchi dati globali che utilizzano questi tipi di dati.



Figura 3.27: Compilazione del programma utente presente sul TIA Portal

La compilazione del programma deve essere effettuata ogni qualvolta viene modificato il programma utente e ciò che si osserva sul TIA Portal è un l'elenco di operazioni che fa il sistema per verificare che sia tutto compilato bene. Al termine vengono visualizzati il numero di errori o avvisi, ammesso che ci siano. Cliccando, invece, sul tasto *carica nel dispositivo* si apre una nuova finestra, denominata *Caricamento avanzato*, nella quale si dovranno compilare i seguenti campi:

- *Tipo di interfaccia PG/PC*, che nel caso in esame è il PN/IE;
- Interfaccia PG/PC, nella quale si inserisce la scheda di rete che si vuole utilizzare per il collegamento;
- Collegamento con l'interfaccia/la sottorete, nella quale si inserisce in che modo avviene il collegamento (se diretto o tramite un posto connettore come uno scalance);

	Dispositivo	Tino di dispositivo	Porto c	Tipo di inter	a Indirizzo	Sottore	to
	PLC_1	CPU 1214C DC/D	1 X1	PN/IE	192.168.0.1	PN/IE_	1
		Tipo di interfaccia f Interfaccia f	PG/PC: [	PN/IE	GBE Family Controller	• •	
	Collegamen	to con l'interfaccia/la sotto 1º gati	eway:	Direttamente	nel posto connettore	1 XI	
	Selezionare il siste	ema di destinazione:			Visualizza tutti i nor	di compatibili	
	Selezionare il siste Dispositivo	ema di destinazione: Tipo di dispositivo	Tipo di i	interfaccia Ir	Visualizza tutti i noi ndirizzo	di compatibili Dispositivo	di de
no. — ]	Selezionare il siste Dispositivo PLC_1	ema di destinazione: Tipo di dispositivo CPU 1214C DC/D	Tipo di i	interfaccia Ir 1	Visualizza tutti i nor Idirizzo 92.168.0.1	di compatibili Dispositivo PLC_1	di de
τ	Selezionare il siste Dispositivo PLC_1 	ema di destinazione: Tipo di dispositivo CPU 1214C DC/D —	Tipo di i PN/IE PN/IE	interfaccia Ir 1 Ii	Visualizza tutti i non ndirizzo 92.168.0.1 ndirizzo di accesso	di compatibili Dispositivo PLC_1 -	di de
LED lampeggia	Selezionare il siste Dispositivo PLC_1 -	ema di destinazione: Tipo di dispositivo CPU 1214C DC/D. —	PN/IE	interfaccia Ir I Ii	Visualizza tutti i nor Idirizzo 92.168.0.1 Idirizzo di accesso	di compatibili Dispositivo PLC_1 —	di des
LED lampeggia	Selezionare il siste Dispositivo PLC_1 -	ema di destinazione: Tipo di dispositivo CPU 1214C DCD. —	<ul> <li>Tipo di i</li> <li>PN/IE</li> <li>PN/IE</li> </ul>	interfaccia Ir 1 Ir	Visualizza tutti i nor Idirizzo 92.168.0.1 Indirizzo di accesso	di compatibili Dispositive PLC_1 -	di de
LED lampeggia	Selezionare il siste Dispositivo PLC_1 -	ema di destinazione: Tipo di dispositivo CPU 1214C DC/D —	Tipo di i	interfaccia Ir 1 II	Visualizza tutti i nov idirizzo 22.168.0.1 ndirizzo di accesso	di compatibili Dispositivo PLC_1 - - Avvia	di de: ricerc
LED lampeggia	Selezionare il siste Dispositivo PLC_1 - - - - - - - - - - - - -	ema di destinazione: Tipo di dispositivo CPU 1214C DCID. - - dirizzo 192.168.0.1 è riusc	Tipo di i PN/IE PN/IE	interfaccia Ir 1 II	Visualizza tutti i nor dirizzo 92.168.0.1 dirizzo di accesso	di compatibili Dispositivo PLC_1 — — 	di de: ricerc
LED lampeggia	Selezionare il siste Dispositivo PLC_1 - - - - - - - - - - - - -	ema di destinazione: Tipo di dispositivo CPU 1214C DC/D. - - - - - - - - - - - - -	Tipo di i PN/IE PN/IE	interfaccia Ir 1 1	Visualizza tutti i nor dirizzo 92.168.0.1 Idirizzo di accesso	di compatibili Dispositivo PLC_1 — — 	di de: ricero

Figura 3.28: Caricamento del progetto presente sul TIA Portal nel dispositivo, 1

Una volta completati tutti i campi, bisogna cliccare su Avvia ricerca e aspettare che il TIA Portal trovi il PLC ricercato. Come è possibile vedere dalla figura 3.28, la ricerca ha trovato un nodo sull'indirizzo "192.168.0.1" il quale corrisponde esattamente all'indirizzo del PLC sull'hardware dell'ambiente TIA Portal. Visivamente è anche possibile verificare che il PLC fisico a cui ci si vuole collegare sia quello: lo si fa mediante la funzione *LED lampeggia*, dove una volta spuntata la casella fa lampeggiare il sistema PLC. Verificato che il controllore sia quello esatto è possibile caricare nel dispositivo il progetto mediante il tasto *Carica*.

tato	1	Desti	nazione	Messaggio	Operazione
40	0	▼ PI	LC_1	Pronto per l'operazione di caricamento.	Carica 'PLC_1'
	0	•	Online è attuale.	La Configurazione non viene caricata perché lo stato online è attu.	
	0	•	Funzione di test e	Le funzioni di test e messa in servizio attive possono essere annul	Applica tutto
1					

Figura 3.29: Caricamento del progetto presente sul TIA Portal nel dispositivo, 2

Si aprirà di seguito un'ulteriore finestra, definita Anteprima di caricamento, come quella riportata nella figura 3.29, nella quale bisogna verificare o modificare le operazioni che richiede di fare. Ottenute tutte le spunte blu sulla sinistra, è possibile caricare in maniera definitiva il progetto nel dispositivo con il tasto *Carica* e una volta terminato il caricamento bisogna chiudere la finestra mediante il tasto *Fine*. In questa maniera il PLC fisico avrà al suo interno il progetto

appena caricato e sarà pronto ad elaborare il programma utente.

La medesima cosa è possibile farla con un HMI (pannello operatore), cliccando innanzitutto sul HMI presente sul menu a sinistra, per poi fare prima una compilazione e successivamente una caricamento nel dispositivo con i tasti rispettivamente *compila* e *carica nel dispositivo*. Tuttavia, prima di effettuare questa operazione bisogna collegare virtualmente e fisicamente il PLC con l'HMI mediante un cavo PROFINET. Virtualmente è necessario andare su *Dispositivi & Reti* dal menu di sinistra, selezionare la *Vista di rete* e una volta cliccato su *collega in rete*, partendo dal quadratino verde presente sulla CPU, bisogna tenere premuto fino al raggiungimento del quadratino verde dell'HMI formando nel seguente modo il collegamento PN/IE come nell'immagine che segue.



Figura 3.30: Collegamento virtuale in rete sul TIA Portal

A questo punto, rimanendo sempre nella stessa finestra, si deve cliccare sul voce *Collegamenti* e bisogna effettuare la stessa procedura del caso precedente collegando i due dispositivi, ma ottenendo in questo caso una linea tratteggiata come quella riportata nella figura 3.31. Il collegamento, in questo caso, verrà denominato HMI Collegamento.

VA	Siemens - C:\Users\mike\\Desktop\Tesi Magistrale\Progetto06 gestione HMI\Progetto06 gestione HMI		
Pr	rogetto Modifica Visualizza Inserisci Online Strumenti Tool Finestra ?		1
	🛉 💁 🖬 Salva progetto 📕 🐰 🗉 🗈 🗙 🏷 ± 🥂 🗄 🛄 🗳 🔛 🖉 🖉 🖉 Collega online 🖉 Interrompi collegamento online 🏭 🖪 🖬 🛠 🖃 🗋 🔪		
	Navigazione del progetto 🛛 🗧 🕈 Progetto06 gestione HMI > Dispositivi & Reti	7	×
	Dispositivi 🖉 Vista di rete 👔 Vista di sposi	itivi	
	🔞 🔲 🗃 💦 Collega in rete 🔢 Collegamenti Collegamento HM 🔍 🐷 🖏 冒 🖽 🛄 🔍 🛓 🏹	4	
	3 Evidenziato: Collegemento	^	
	Aggiungi nuovo dispositivo	=	
	Dispositivi & Reti     Dispositivi & Reti     CPU 1214C DC/DC/     PLC_1 (CPU 1214C DC/DC/     PLC_1 (CPU 1214C DC/DC/	-	
	v Olimina o Diagnosita		
	Inserisci nuovo blocco HMI_Collegamento_2		
	America Main [OB1]		
	Dati HM [D81]		
	Gggetti tecnologici		Dat
	Sorgenti esterne		₫.
	Variabili PLC		rete
	▶ Le Tipi di dati PLC	1	
	Figure Tabella di controllo e di f		

Figura 3.31: Collegamento virtuale HMI sul TIA Portal

Fatti questi passaggi il sistema PLC più HMI è pronto per il funzionamento.

#### 3.4.2 L'utilizzo del PLCSIM Standard

Come è stato già accennato nei paragrafi precedenti, potrebbe capitare di non avere la CPU fisica per poter caricare il progetto al suo interno oppure un HMI. Tuttavia, Siemens è riuscito a risolvere questo problema introducendo un pacchetto, il quale è integrato nell'ambiente TIA Portal, che permette di simulare la CPU SIMATIC S7-1200 e la S7-1500. Per poter utilizzare questa modalità bisogna cliccare sul dispositivo che si vuole simulare e successivamente cliccare sul tasto nel menu in alto Avvia Simulazione, indicato nella figura 3.32.

7 Siemens - C:\Users\mikel\Desktop\Tesi M	agistrale\Progetto06 ge	stione	HMI\Pro	ogetto0	6 gestion	e HMI						
Progetto Modifica Visualizza Inserisci O 📑 隆 📮 Salva progetto 📑 💥 🗐 👔	nline Strumenti Tool K 🎦 ± (य ± 🗟 🗓	Finest	tra?	Colleg	ja online 🏢	🕅 Interrompi collega	mento o	nline	?	<b>.</b> ×		] •
Navigazione del progetto	Progetto06 gestione	HMI≯	PLC_1	[CPU 1	214C DC	/DC/DC]						
Dispositivi						2	Vista t	opologi	ca	- Vista	di rete	ľ
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	H PLC_1 [CPU 1214C]		-			🔲 🔍 ±						
Topostiol (1)         Image: Second	Telaio di montagg	103	102	101		1 Rest in all Rest in all Rest Rest Rest Rest Rest Rest Rest Rest	2	3	4	5	6	7

Figura 3.32: Avviamento della simulazione del PLC sul TIA Portal, 1

Di seguito comparirà un messaggio come quello riportato nella figura 3.33, al quale bisognerà cliccare OK.



Figura 3.33: Avviamento della simulazione del PLC sul TIA Portal, 2

A questo punto si apriranno due finestre come riportato nella figura 3.34. In particolare, in alto a sinistra è presente l'interfaccia del PLCSIM, mentre a destra, con dimensioni nettamente più grandi, la stessa finestra che compariva quando si andava a caricare il programma nel dispositivo con l'unica differenza che nel campo *Interfaccia PG/PC* è presente *PLCSIM*. Da questo punto in poi sarà come connettersi online (bisogna cliccare prima su carica, poi sulla nuova finestra di nuovo carica e di seguito fine), con la differenza che in questo caso si ha un PLC simulato e non fisico. Lo stesso si può fare con un pannello operatore.

### 3.4.3 PLCSIM Standard VS PLCSIM Advanced

Il PLCSIM Standard, come è stato già specificato nel paragrafo precedente, è integrato nel TIA PORTAL e serve per simulare S7-1200 e S7-1500, mentre PLCSIM Advanced è un'opzione

₩a s	iemens - C:\Users\mikel\Desktop\Tesi Mag	jistrale\Progetto06 gesti	one HMI\Progetto0	6 gestione HMI				
Pro	🖌 Siemens 🖉 🗕 🗙 🗓 👓	Caricamento avanzato						×
2	PLC non configurato [SIM-1200] 🔤 值 🗡		Nodi di accesso cor	figurati di "PLC_1"				50 50
			Dispositivo	Tipo di dispositivo	Posto c	Tipo di interfa	Indirizzo	Sottorete
			PLC_1	CPU 1214C DC/D	1 X1	PN/IE	192.168.0.1	PN/IE_1
ivi & Ret	ERROR MAINT PAUSE							
posit	MRES			Interfaccia PG/PG	н <mark>с</mark> е с. <b>Ю</b> Р			
Dis	X1		Collegamento co	n l'interfaccia/la sottorete	PN/IE	1		
	<nessun progetto=""></nessun>			1° gateway	v [	-		
	2 mpostazioni securty							
	Qer Funzioni oltre i limiti dei PLC     Dati comuni		Selezionare il sister	na di destinazione:			visualizza i disposit	ivi con la stessa indiri 💌
	Informazioni sul documento	_	Diseasitive		The end in	L.		Disessitive di des
	🕨 🐻 Lingue & Risorse	the second se	CPUsamman	CPU 1200 Simula	PN/IC	102	169.0.1	CPU common
	Version Control Interface		Crocommon	CF0-1200 Simula	. FINIE	192	. 166.U. 1	Crucommon
	Accessi online	·	-	-	FINIL	man	1220 UI accesso	_
•	📴 Card Reader/memoria USB	<u> </u>						
		E LED lampeggia						
	Vista dettagli							
1	Moduli							<u>Avvia ricerca</u>
		Informazioni sullo stato o	nline:			E	Visualizza solo m	essaggi di errore
	Nome	🚯 Ricerca terminata. So	ono stati trovati 1 nod	i compatibili su 1 nodi ac	cessibili.			^
	Configurazione dispositivi	🐴 Richiamo informazior	ni sui dispositivi in cor	50				
9	Online & Diagnostica 📃	Scansione e interrog	azione terminate.					
	Blocchi di programma							~
	Oggetti tecnologici							[and
10	Sorgenti esterne						Cari	an l ( Annulla )
							Cari	ca <u>A</u> nnulla

Figura 3.34: Avviamento della simulazione del PLC sul TIA Portal, 3

aggiuntiva che deve essere acquistata che simula esclusivamente S7-1500 e ha funzionalità in più rispetto a quella classica. Il PLCSIM Standard dà la possibilità di variare manualmente il valore delle variabili forzandole e osservando sulla colonna "Valore di controllo" il valore di una certa variabile. Come si evince dalla figura 3.35, è anche possibile creare una sequenza, ovvero è possibile creare una serie di eventi successivi che possono avvenire sugli ingressi sfasati nel tempo. In questo modo è possibile vedere come lavora la macchina in esame a fronte di questi eventi.

	SIM_1							- 61
9 <sub>1</sub> 9	9 3 3 3 4 4							
	Nome In	dirizzo Fo	mato di visual	Valore d	i controllo/supervisi	Bit	Modifica coe	rente 🗲
-0	"Abilita":P %	10.0:P Bo	ol	FALSE			FALSE	M 1
-0	"Acceso" %	Q0.0 Bo	l	FALSE			FALSE	
-0	*DatiHM*.Comm	Bo	le	TRUE			SALSE .	
	"DatiHM".Comm	Nu	mero in virgol	0			O.	
	(14)						*	
		-	•		_	_		0
eque	nza_1 양 중 문, 및 1		Intervallo pre	ed	50 ms 🕒 🕯	<ul> <li>Te</li> </ul>	mpo di esec	00:00:00.00
eque	nza_1 84 8 8 12 14	Nome	Intervallo pre	ed	50 ms 🕒	• Te Azione	mpo di esec Parametro az	00:00:00.00
eque ₽ ≝?  ++	nza_1	Nome	Intervallo pre	ed	50 ms 🕑 🕯 Formato di visual	<ul> <li>Te</li> <li>Azione</li> <li>Avvia immediata</li> </ul>	mpo di esec Parametro az	00:00:00.00
eque ₽ ⋛	nza_1	Nome *Abilita*:	Intervalio pre	ed ndirizzo 610.0:P	50 ms 🕚 🕏 Formato di visual Bool	Te Azione Avvia immediata Imposta al valore	mpo di esec Parametro az  : TRUE	00:00:00.00
eque ₽ ≝₽	nza_1 Br & Bs III I Tempo 00:00:00.00 00:00:00.00 00:00:10.00	Nome *Abilita*: *DatiHM	Intervalio pre	ed ndirizzo 610.0:P	50 ms 🕑 🕯 Formato di visual Bool Bool	<ul> <li>Te</li> <li>Azione</li> <li>Avvia immediata</li> <li>Imposta el valore</li> <li>Imposta el valore</li> </ul>	mpo di esec Parametro az	00:00:00.00
eque ₽ ₫	nza_1 Br	Nome *Abilita*: *DatiHM *Abilita*:	Intervalio pre in comm.Av 9 a	ed ndirizzo 610.0:P	50 ms 🕒 🕯 Formato di visual Bool Bool Bool	<ul> <li>Te</li> <li>Azione</li> <li>Avvia immediata</li> <li>Imposta al valore</li> <li>Imposta al valore</li> <li>Imposta al valore</li> </ul>	mpo di esec Parametro az TRUE TRUE FALSE	00:00:00.00
eque ₽ ₽₽	nza_1 Br & B, III II Tempo 00:00:00.00 00:00:10.00 00:00:15.00 00:00:15.00	Nome *Abilita*: *OatiHM *Abilita*: *OatiHM	Intervalio pre	ed ndirizzo 610.0:P 610.0:P	50 ms  S  S Formato di visual Bool Bool Bool	<ul> <li>Te</li> <li>Azione</li> <li>Avvia immediata</li> <li>Imposta al valore</li> <li>Imposta al valore</li> <li>Imposta el valore</li> <li>Imposta el valore</li> </ul>	mpo di esec Parametro az TRUE TRUE FALSE FALSE	00:00:00.00

Figura 3.35: PLCSIM Standard, tabelle di simulazione e sequenze di segnali

Tutto ciò è però possibile farlo solo se si passa al quadro completo della finestra PLCSIM, cliccando il tasto *Passa alla vista progetto* in alto a destra. Una volta aperta la nuova finestra, bisognerà creare il nuovo progetto. Tale progetto definisce il PLC Simulato all'interno del quale verrà scaricato il programma scritto dal softwarista.



Figura 3.36: Schermata PLCSIM Standard

Per quanto il PLCSIM Standard possa essere un potente mezzo di simulazione, anch'esso possiede dei limiti come:

- la possibilità di lanciare massimo due istanze sullo stesso PC e farle comunicare tra di loro;
- l'impossibilità di comunicare con l'esterno, come ad esempio comunicare su base TCP e che non può testare più PLC insieme, il Webserver o un HMI di terze parti;
- l'inefficacia di alcune sequenze, laddove esse dipendono da parametri che non si possono controllare;
- la non retroattività, ovvero che in caso di problemi con il codice, se modificato, bisogna rifare tutte le prove e quelle precedenti fatte sono risultate inutili.

Molti di questi limiti vengono risolti dal PLCSIM Advanced, il quale, rispetto al PLCSIM Standard non funziona per gli S7-1200 ma solo per gli S7-1500. Esso è un controllore virtuale che non simula il tempo ciclo reale del PLC, ma il suo scopo è quello di simulare le comunicazioni (ha un indirizzo IP che può vivere all'interno del PC). Ciò significa che il PLCSIM Advanced può far comunicare più istanze del PLCSIM e questo avviene grazie allo *PLCSIM Virtual Switch* e a sua volta questo lo mette in collegamento con l'esterno, in quanto si può interfacciare con la scheda di rete fisica del PC per poi parlare con altri PC. Tuttavia, esso non simula il PROFINET.



Figura 3.37: Simulazione e collegamenti mediante PLCSIM Advanced

Un'altra funzionalità importantissima è la possibilità di mettere a disposizione delle API, ovvero interfacce pubbliche o librerie di istruzioni che possono essere messe a disposizione di altri software che possono utilizzare per interagire con il PLCSIM Advanced. Queste possono essere utilizzate anche con codice C++ o C# (Utili per leggere le uscite o scrivere gli ingressi). Per cui si potrebbe creare un codice C# (C Sharp) che va a leggere l'uscita, come un'attivazione di una valvola o un pistone, e successivamente sfruttare l'altra funzione di scrittura degli ingressi, scrivere il comando di ingresso al PLC il comando, ad esempio di valvola aperta.

## 3.5 Scrivere un programma per PLC

Dopo aver spiegato come si configura e si riconosce un dispositivo presente sul campo, è necessario scrivere il programma utente che vede come protagonisti gli stessi. Per poter programmare un PLC, come è stato già anticipato nel capitolo 2, ci si avvale di diversi linguaggi di programmazione, definiti dallo standard IEC 61131-3, e di diversi blocchi di programmazione, i quali non sono stati ancora discussi. Di seguito sarà mostrato in che modo tali blocchi possono essere aggiunti nel progetto e a cosa servono, sfruttando la guida presente sul TIA Portal [24]. Innanzitutto, è possibile classificare tali blocchi in quattro diverse categorie:

- *Blocchi Organizzativi*, noti anche come *OB di ciclo*, i quali costituiscono l'interfaccia tra sistema operativo e programma utente;
- *Blocchi Funzionali*, noti anche come *FB*, i quali sono blocchi di codice che memorizzano i propri valori in modo permanente nei blocchi dati di istanza in modo da poterne disporre anche dopo l'elaborazione del blocco;
- *Blocchi Funzione*, noti anche come *FC*, i quali sono simili ai precedenti con l'unica differenza che non sono dotati di memoria;
- Blocchi Dati, noti anche come DB, che servono a memorizzare i dati del programma.

Per aggiungere i diversi blocchi, bisogna andare sul menu di sinistra, selezionare il PLC all'interno del quale si desidera scrivere il programma utente, cliccare sulla sezione *Blocchi di Programma* e infine selezionare la voce *Inserisce nuovo blocco*. Così facendo si aprirà una nuova finestra, come quella riportata nella figura 3.38, all'interno della quale si può scegliere quale blocco inserire e quale linguaggio si intende usare.



Figura 3.38: Inserimento di nuovi blocchi di programma sul TIA Portal

Una volta selezionato il blocco di programma che si intende utilizzare, è possibile programmare mediante i *Preferiti* (contatto NC, contatto NA, bobina, box, ecc) presenti nell'editor in alto, come mostrato nella figura 3.39.

Progetto Tesi Magist	trale 🕨 PLC_1 [CPL	J 1214C DC/DC/D	)C] → Blocchi di j	programma 🕨	Blocco_Pulsant	tiera [FC2]	_ <b>@</b>
<sub>K</sub> ä <sub>K</sub> a ∌ ⊉ ∎ <sub>a</sub>	e 2 = 9 4:	: @ : 별 : 🖃	😥 to to ta	a 4≱ ¢= I= `	<u>।</u> (। (। 🖓		
		-	Interfaccia del bloci	0			
	<b>ь _т</b> ₹ы⊧ к	BF H					
<ul> <li>Titolo del blocco:</li> </ul>							^
Commento							
▼ Segmento 1:							
Commento							=
0. Ob?	%IO_O	%10.7		%M1.0			
"START"	"START"	"FC3-IN"		"M1"			
				(s )			
"Dati_HMI".start	%0.1			%M1.1			
	"SELETTORE"			"M2"			
0/884 4				(R)			
*M2*	"Dati_HMI".						
	selettore						
	"Dati_HMI".start						
	$\square$						

Figura 3.39: Esempio di programmazione mediante i "preferiti" sul TIA Portal

## 3.5.1 Blocchi Organizzativi

I blocchi organizzativi (OB) definiscono l'interfaccia tra il sistema operativo e il programma utente. Tali blocchi vengono ripresi dal sistema operativo e comandano diverse operazioni, come ad esempio:

- l'elaborazione ciclica del programma;
- la verifica iniziale del comportamento del sistema di automazione;
- il trattamento degli errori;
- l'elaborazione del programma comandata da un allarme.

Tuttavia, esistono diversi tipi di OB e tra i più importanti si possono citare:

- gli *OB di avviamento*, che vengono elaborati una sola volta al passaggio dello stato di funzionamento della CPU da STOP a RUN. Al termine dell'elaborazione di tale OB, inizia quella dell'OB di ciclo;
- gli *OB di ciclo*, i quali vengono elaborati ciclicamente. Sono blocchi di codice di livello superiore nel programma nei quali avviene la programmazione di istruzioni da parte dell'utente o la creazione di ulteriori blocchi;
- gli *OB di schedulazione orologio*, che permettono di avviare programmi in maniera indipendente dall'elaborazione ciclica del programma in intervalli definiti (l'intervallo di schedulazione può essere modificato dalle proprietà dell'OB);
- gli *OB dell'allarme di ritardo*, i quali interrompono l'elaborazione ciclica del programma trascorso un tempo definito (il tempo di ritardo viene impostato nel parametro STR\_DINT);
- gli *OB di interrupt di processo*, che interrompono l'elaborazione ciclica del programma per via di un evento hardware definito nelle proprietà dell'OB;
- gli *OB di allarme di diagnostica*, i quali interrompono l'elaborazione ciclica del programma se l'unità con funzioni di diagnostica per la quale è stato abilitato l'allarme individua un errore.

## 3.5.2 Blocchi Funzionali

I blocchi funzionali sono blocchi di codice che memorizzano i parametri di ingresso, uscita e transito in maniera permanente nei blocchi dati di istanza così da poterne disporre anche dopo l'elaborazione del blocco. Per questo sono definiti blocchi con "memoria". Tuttavia, essi lavorano anche con le variabili temporanee, le quali non vengono salvate nel DB di istanza ma restano a disposizione solo per un ciclo. I FB contengono sottoprogrammi che vengono eseguiti quando vengono richiamati da un altro blocco di codice. Inoltre, può essere richiamato anche più volte in diversi punti all'interno di un programma permettendo in questa maniera di semplificare la programmazione di funzioni complesse a ricorrenza frequente. Il richiamo di un blocco funzionale viene definito istanza. Ogni istanza di un blocco funzionale necessita di un blocco dati di istanza nel quale vengono salvati valori specifici dell'istanza per i parametri formali dichiarati nell'FB. Esisto due diverse possibilità di accesso ai blocchi dati di istanza assegnati ad un FB al momento del richiamo:

- Blocchi dati con accesso ottimizzato, i quali non presentano una struttura definita della memoria. Gli elementi all'interno del blocco dati contengono solo un nome simbolico, ma non un indirizzo interno al blocco;
- *Blocchi dati con accesso standard*, i quali hanno una struttura definita dalla memoria. In questo caso gli elementi oltre ad avere un nome simbolico, presentano anche un indirizzo fisso.

#### 3.5.3 Blocchi Funzione

I blocchi funzioni sono blocchi di codice senza memoria e dunque i dati sviluppati al suo interno non vengono salvati. Per questo quando la funzione viene richiamata i parametri formali devono essere sostituiti con parametri attuali. Per il salvataggio permanente dei dati, le funzioni dispongono di blocchi dati globali (esistono blocchi dati globali, ai quali è possibile accedere da tutti i blocchi di codice, e blocchi dati di istanza, assegnati a un determinato richiamo di FB). Una funzione contiene un programma che viene eseguito quando la funzione viene richiamata da un altro blocco di codice. Le funzioni possono essere impiegate per:

- la restituzione dei valori della funzione per il blocco richiamante, come per funzioni matematiche
- l'esecuzione di funzioni tecnologiche, come controlli singoli con collegamenti binari

Anche una funzione può essere richiamata anche più volte in diversi punti all'interno di un programma permettendo di semplificare la programmazione di funzioni complesse a ricorrenza frequente.

#### 3.5.4 Blocchi Dati

I blocchi dati sono destinati alla memorizzazione dei dati del programma. Essi contengono quindi dati variabili che vengono utilizzati dal programma utente. I blocchi dati globali acquisiscono dati che possono essere utilizzati da tutti gli altri blocchi. La loro dimensione massima varia in funzione della CPU e la struttura dei blocchi dati globali può essere definita a piacere.

Come è stato già affrontato nei paragrafi precedenti, ogni blocco funzionale, ogni funzione o ogni blocco organizzativo può leggere o scrivere i dati da/in un blocco dati globale. I dati vengono mantenuti nel blocco dati anche quando si esce dal blocco. Un blocco dati globale e un blocco dati di istanza possono essere aperti contemporaneamente. Di seguito viene riportata la figura 3.40 estrapolata dalla guida del TIA Portal [24], la quale mostra i vari accessi possibili ai blocchi dati.



Figura 3.40: Rappresentazione dei vari accessi ai blocchi dati sul TIA Portal

## Capitolo 4

# Banco didattico Pneumax: componenti e struttura

## 4.1 L'idea di un banco didattico

La tesi in questione nasce con l'idea di poter portare all'interno del Politecnico di Torino un banco didattico capace di rappresentare l'automazione industriale anche con semplici componenti pneumatici ed elettropneumatici. Si tratta di un banco didattico acquistato dall'azienda *Pneumax S.p.a.*, la quale dal 1976, anno in cui è stata fondata, ha incominciato a ricoprire un ruolo primario nel panorama mondiale per l'automazione pneumatica. I suoi due punti di forza sono: grande capacità di proporre prodotti nuovi e innovativi e grossa capacità produttiva. La sua vasta gamma di prodotti copre diversi campi che vanno dal trattamento dell'aria, alla sua distribuzione e il suo utilizzo.



Figura 4.1: Tipologia di componenti forniti dalla Pneumax S.p.a. [30]

Molti di questi componenti sono stati acquistati con lo scopo di poter costruire il nuovo

banco didattico da utilizzare all'interno del Politecnico di Torino durante le ore di esercitazione di *Automazione a fluido*. Con esso gli studenti dovranno interfacciarsi e comprenderne la logica di funzionamento. Direttamente dalla Pneumax S.p.a., il banco didattico è giunto al Politecnico di Torino, presso i laboratori del DIMEAS, dentro a dei cartoni intorno al mese di luglio, come mostrato in figura.



Figura 4.2: Banco didattico Pneumax al suo arrivo

Partendo da qui e osservando tutto il materiale fornito, si è iniziato a pensare che tipo di banco poteva venirne fuori. Osservando la struttura base, si è giunti alla conclusione di realizzare una parte fissa al banco e una parte mobile, in modo di poter realizzare con lo stesso banco didattico più esperienze di laboratorio: in particolare, una prima esperienza focalizzata sull'e-lettropneumatica con l'utilizzo di elettrovalvole e cilindri a doppio effetto, una seconda basata sull'utilizzo di trasduttori pneumo-elettrici e una terza basata sul controllo pressione mediante elettrovalvole pneumatiche 2/2. Si parta con la descrizione della struttura base del banco, la quale viene riportata nella figura 4.3. Essa si sviluppa in altezza ed in base a come è stato concepito il banco, è possibile suddividere la struttura in due parti: la parte alta (A), che rappresenta la zona di lavoro sulla quale, grazie alla presenza di barre in acciaio, è possibile agganciare e sganciare dei talai mobili, e la parte bassa (B), la quale viene utilizzata per conservare i restanti telai mobili.


Figura 4.3: Struttura base del banco didattico Pneumax

Sulla parte alta, oltre ad avere un telaio mobile, si avrà anche la parte fissa di cui si parlava prima, la quale è composta da un alimentatore (24 Volt), una morsettiera, un PLC, una pulsantiera, un HMI e, infine, delle cassette pensate per gli input e gli output.

# 4.2 Banco didattico Pneumax: parte fissa

Come è stato già anticipato, sulla struttura base del banco didattico è presente una parte fissa, denominata così in quanto, indipendentemente dall'esperienza che si vuole fare, rimane sempre identica. Nella figura 4.4 viene riportata tale parte fissa e in questo paragrafo verranno definiti tutti gli elementi che la compongono, compreso il cablaggio eseguito tra i vari elementi.



Figura 4.4: Banco didattico Pneumax: parte fissa

#### 4.2.1 Parte fissa: Alimentatore

Si parta dall'elemento fondamentale di questo banco, senza il quale non sarebbe possibile il funzionamento dell'intero banco. Si tratta di un alimentatore stabilizzante a tensione di uscita fissa del *GRUPPO K.E.R.T: Sistemi di energia e alimentazione*, il cui codice è "KAT 5V". Esso è riportato nella figura 4.5, mentre nella tabella sono riportate tutte le specifiche dell'alimentatore, estrapolate dal suo datasheet.

Scheda tecnica Alimentatore	
Tensione uscita	24 Vdc
Corrente uscita	4 A
Tensione ingresso	$230\pm10\%$
Frequenza ingresso	$50 \mathrm{~Hz}$
Ripple	20  mVpp
Classe di protezione IP	IP20
Temperatura di funzionamento	$0 \div 50 \ ^{\circ}\mathrm{C}$
Umidità relativa non condensata	$0\div95~\%$



Figura 4.5: Banco didattico Pneumax, parte fissa: alimentatore

#### 4.2.2 Parte fissa: Morsettiera

Una volta collegato l'alimentatore alla presa di corrente, si è utilizzata una morsettiera, fornita dall'azienda Pneumax, per portare il "+24 Volt" e lo "0 Volt" in tutta la parte fissa. In particolare, essi serviranno per alimentare il PLC, lo Scalance, l'HMI e le cassette per gli input e gli output. Nella figura 4.6 viene riportata la morsettiera con i vari collegamenti.



Figura 4.6: Banco didattico Pneumax, parte fissa: morsettiera

#### 4.2.3 Parte fissa: PLC

Il cuore dell'intero automatismo è invece rappresentato dal PLC. Si tratta della "CPU 1214C DC/DC/DC" (codice "6ES7214-1HG40-0XB0"), il quale fa parte della famiglia S7-1200, ed è prodotto e venduto dalla *Siemens AG*. Inoltre, oltre al PLC, è stato inserito anche un modulo

aggiuntivo per ampliare gli output analogici e uno scalance, entrambi sempre prodotti e venduti dalla Siemens AG. Nella figura 4.7 vengono riportati tutti gli elementi appena citati del banco didattico.



Figura 4.7: Banco didattico Pneumax, parte fissa: PLC, Scalance e Modulo Output analogici

La CPU riunisce in un'unica apparecchiatura compatta un microprocessore, un alimentatore integrato, circuiti di ingresso e di uscita, PROFINET integrato, I/O veloci che supportano la funzione di controllo del movimento e ingressi analogici onboard creando così un potente controllore. Di seguito viene riportato uno schema del controllore preso dal manuale dell'S7-1200 della Siemens AG [28].



Figura 4.8: Schema di un controllore S7-1200, Siemens AG

La CPU dispone di una porta PROFINET per la comunicazione tramite rete PROFINET

e, come è stato già anticipato, comprende svariati moduli, i quali consentono di ampliare le funzionalità della CPU con I/O aggiuntivi o altri protocolli di comunicazione. Nel caso in esame è stato aggiunto un modulo per gli output analogici (utilizzato per il leggere i valori di input al regolatore proporzionale, il quale verrà presentato nei paragrafi successivi), che riporta il codice "6ES7 232-4HA30-0XB0" e uno scalance affinché ci sia comunicazione tra PLC e HMI (anche PC quando si vuole caricare un programma nel PLC). Di seguito viene riportata una tabella riassuntiva con tutte le caratteristiche principali della CPU in esame e uno schema elettrico semplificato della CPU 1214C DC/DC/DC estrapolato sempre dal manuale di sistema della S7-1200 [28].

Caratteristiche principali CPU 1214 C $\rm DC/DC/DC$	
Alimentazione	24 Vdc
Ingressi digitali	14
Uscite digitali	10
Ingressi analogici	2
Uscite analogiche	0
Memoria utente (Lavoro)	100 Kbyte
Memoria utente (Carico)	4 Mbyte
Memoria utente (Ritenzione)	10 Kbyte



Figura 4.9: Schema elettrico della CPU 1214C DC/DC/DC, Siemens

Nella figura 4.9 dello schema elettrico, il numeretto "1" indica l'uscita di alimentazione per sensori 24 Volt DC (per aumentare l'immunità al rumore si consiglia di collegare "M" alla massa del telaio anche se non si utilizza l'alimentazione per i sensori). Per gli ingressi ad assorbimento di corrente bisogna collegare "-" a "M" (come viene indicato nello schema con il numeretto "2"). Per gli ingressi ad emissione di corrente, il "+" a "M".

#### 4.2.4 Parte fissa: HMI

Il banco didattico Pneumax presenta nella parte fissa anche un pannello operatore, anche noto come HMI, utilizzato sempre più nel campo dell'automazione per svolgere funzioni di comando e visualizzazione anche nelle macchine e negli impianti di piccole dimensioni. Si tratta di un "Basic Panel 2nd Generation Siemens", il cui modello è il "KTP700 Basic" ed etichettato con il codice "6AV2 123-2GB03-0AX0". Nella figura 4.10 viene riportata la foto dell'HMI del banco didattico e una tabella riassuntiva delle caratteristiche principali del pannello operatore del banco didattico.



Figura 4.10: Banco didattico Pneumax, parte fissa: HMI

Caratteristiche principali KTP700 Basic	
Alimentazione	24 Vdc
Area del display attiva	154,1 x 85,9 mm (7")
Risoluzione	$800 \ge 480$ pixel
Colori rappresentabili	16 bit (65536 colori)
Memoria di dati	256  MB
Memoria di programma	512  MB

Nella figura 4.11 viene riportato uno schema generale del touch panel preso dalle istruzioni operative della *Siemens* [29].



Figura 4.11: Schema di un HMI, modello KTP700 Basic

#### 4.2.5 Parte fissa: Pulsantiera

Altro elemento posizionato nella parte fissa, in quanto comune a tutte le varie esercitazioni, è la pulsantiera, come quella riportata nella figura 4.12, la quale presenta tre diversi pulsanti: uno tasto bistabile pensato per azionare l'emergenza (quello rosso), un tasto monostabile pensato per dare l'avvio ad un ciclo manuale (quello verde) e un selettore pensato per avviare un ciclo automatico.



Figura 4.12: Banco didattico Pneumax, parte fissa: Pulsantiera

#### 4.2.6 Parte fissa: Cassette per gli input/output

Per completare la parte fissa del banco didattico, sono state inserite delle cassette pensate per facilitare il collegamento degli input e gli output con il PLC mediante l'uso di prese da pannello e connettori a banana. Tali cassette, riportate nella figura 4.13 come esempio, sono state realizzate artigianalmente utilizzando sia del materiale messo a disposizione dal dipartimento DIMEAS del Politecnico di Torino e sia del materiale acquistato presso rivenditori specializzati come BI-ESSE, GBC e SIRO.

È stata adottata questa soluzione per permettere agli studenti futuri di lavorare in piena sicurezza, senza mai toccare i collegamenti con il PLC. Inoltre, per fissarle sulla parte alta del banco didattico, sono stati utilizzati dei magneti gentilmente offerti dall'azienda Pneumax. Nella figura 4.14 è riportato il retro di una cassetta tipo, per mostrare i collegamenti tra i cavi provenienti da PLC alle prese da pannello che sono presenti sulle stesse.



Figura 4.13: Vista frontale della cassetta per gli input e schema 2D



Figura 4.14: Cablaggio interno presente nella cassetta degli input e schema 2D

#### 4.2.7 Parte fissa: Schema elettrico

In questo sottoparagrafo viene riportato lo schema elettrico relativo alla parte fissa del banco didattico Pneumax. Rispetto alla figura 4.4, la quale ritrae un'immagine reale del banco, nello schema elettrico è stata spostata la pulsantiera per una migliore lettura del disegno.

N.B. nello schema elettrico i collegamenti tra il PLC e le cassette degli input/output sono visibili fino al raggiungimento della rispettiva cassetta; in realtà, come mostrato nella figura 4.15, quei cavi terminano sulle prese da pannello corrispondenti. Ad esempio, si consideri l'ingresso

"I0.0". Da esso parte un cavo blu, il quale entra all'interno della cassetta degli input e si congiunge alla presa da pannello corrispondente all'ingresso "I0.0", come indicato sulla stessa.



Figura 4.15: Banco didattico Pneumax, parte fissa: Schema elettrico

### 4.3 Banco didattico Pneumax: parte mobile, PANNELLO 1

Dopo aver descritto in maniera accurata la parte fissa (componenti e cablaggio), si passi alla descrizione, in questo paragrafo, della parte mobile e, in particolare, il primo pannello pensato per le esercitazioni di *Automazione a fluido*. Il pannello 1 prevede l'utilizzo di un pacchetto di elettrovalvole Optyma32-S, tre cilindri a doppio effetto, un filtro regolatore, una valvola di intercettazione e dei finecorsa magnetici. Quest'ultimi, i quali sono da montare sui cilindri, sono utilizzati per rilevare la posizione dello stelo: quando sono investiti dal campo magnetico generato dal magnete permanente del pistone, il finecorsa emette un segnale elettrico utilizzabile per il comando diretto di elettrovalvole. Nella figura 4.16 viene riportato il pannello 1 con tutti i suoi componenti, i quali verranno descritti nei sottoparagrafi seguenti. Le informazioni riguardanti tali componenti sono stati estrapolati dal catalogo Pneumax [30] fornito direttamente dall'azienda stessa.



Figura 4.16: Banco didattico Pneumax, parte mobile, pannello 1

#### 4.3.1 Pannello 1: Elettrovalvole Optyma32-S

Optyma32-S rappresenta un ampliamento della gamma Optyma spingendosi verso taglie più piccole, basate sempre su montaggio modulare con integrata all'interno la gestione dei segnali elettrici di comando. Questa linea di elettrovalvole ha una portata nominale massimizzata fino a 550 [Nl/min] e gli elettropiloti montati su di esse risultano a basso consumo energetico entrambi posizionati sullo stesso lato dell'elettrovalvola. Ogni elettrovalvola, bistabile o monostabile che sia (occupano lo stesso spazio), può essere assemblata rapidamente sulla sottobase grazie alla presenza di una vite ed è possibile sostituirle senza preoccuparsi di disconnettere i collegamenti

pneumatici. Nella figura 4.17 è riportato il pacchetto di elettrovalvole utilizzato per il pannello 1 del banco didattico, mentre nella tabella che segue la figura sono riportate tutte le caratteristiche funzionali delle elettrovalvole.



Figura 4.17: Pannello 1: elettrovalvole Optyma32-S

Caratteristiche funzionali delle elettrovalvole Optyma32-5	5
Tensione di alimentazione	24 Vdc
Assorbimento elettropiloti	0.5 Watt
Pressione di lavoro condotti valvola	da vuoto fino a 10 bar $\max$
Pressione di alimentazione elettropiloti	da 2,5 fino a 7 bar max
Temperatura di impiego	$-5^{\circ}C a + 50^{\circ}C$
Grado di protezione	IP65
Fluido	Aria filtrata e lubrificata o non

Con tali elettrovalvole è possibile gestire fino a 22 segnali elettrici (11 bistabili, 22 monostabili o la combinazione dei due in modo tale che la somma dei segnali sia massimo 22) e il collegamento elettrico avviene tramite un connettore a vaschetta a 25 poli, il quale attraverso un cavo multipolare riporta tali segnali al PLC.

Secondo l'articolo di Roberto Mazzei, *Pneumatica a isole* [31], si è passati a questa nuova tecnologia in quanto essa ha permesso non solo di ridurre il numero di elementi di dettaglio che l'utente deve curare in fase di progettazione, ma anche i costi di manutenzione, montaggio, installazione e collaudo. Per cui i componenti destinati all'installazione a bordo macchina devono essere raggruppati e precablati, come le elettrovalvole Optyma32-S, permettendo la nascita della *Pneumatica a isole*, ovvero i "terminali di valvole". Mediante un terminale di valvole a connessione multipolare, l'operatore dovrà solamente connettere il terminale al quadro di controllo utilizzando un cavo multipolare collegato da un lato alla morsettiera e dall'altro al terminale a mezzo di presa multipolare connessa alla corrispondente spina. Di seguito viene riportata una



Figura 4.18: Descrizione delle elettrovalvole Optyma32-S

figura estrapolata sempre dallo stesso articolo di Mazzei che esprime il concetto di connessione multipolare.

Ritornando alle elettrovalvole Optyma32-S, la distribuzione dei segnali elettrici tra i singoli moduli avviene mediante una scheda elettrica posta nella sottobase che riceve i segnali dal modulo precedente, preleva i segnali necessari per gestire gli elettropiloti dell'elettrovalvola in questione e trasmette i rimanenti a valle. Le sottobasi modulari sono a doppio posto valvola e disponibili in due diverse tipologie:

- Base modulare a due posizioni bistabili, dove vengono occupati 2 segnali elettrici dalla posizione 1 e altrettanti dalla posizione 2, occupando in definitiva 4 segnali elettrici per ogni sottobase;
- Base modulare a due posizioni monostabili, dove vengono occupati 1 segnale elettrico dalla posizione 1 e 1 dalla posizione 2, occupando solamente 2 segnali per ogni sotto-base.

Per cui, nella base a due posizioni bistabili, il primo segnale viene connesso con l'elettropilota lato 14 della prima posizione e il secondo con l'elettropilota lato 12 sempre nella stessa posizione. Lo stesso si ripete con il terzo e quarto segnale nella posizione 2 dell'elettrovalvola. I rimanenti segnali vengono trasferiti a valle. Tuttavia, su una base bistabile, è possibile montare anche elettrovalvole monostabili, ma in questo caso si perde un segnale per ciascuna elettrovalvola. Proprio per questo motivo è possibile montare sia elettrovalvole bistabili che monostabili su



Figura 4.19: Circuito integrato con terminale a connessione multipolare

tutto il blocco. Su una base monostabile non è invece possibile montare elettrovalvole bistabili, in quanto non è predisposta per azionare l'elettropilota lato 12. Di seguito, vengono riportati alcuni esempi (figure 4.20, 4.21 e 4.22), presi dal catalogo della Pneumax [30], su come avviene la corrispondenza segnale/pin di riferimento considerando un connettore a 25 poli.



Figura 4.20: Corrispondenza PIN per batteria di elettrovalvole montate tutte su doppie basi per bistabile

Come è stato già anticipato, tutti questi segnali elettrici, mediante un cavo multipolare, vengono riportati sulle morsettiere degli input e gli output e collegati in base al programma utente caricato nel PLC. In particolare, ad ogni segnale elettrico proveniente dalle elettrovalvole corrisponderà un cavo di un dato colore e seguendo la tabella sotto riportata è possibile effettuare i giusti collegamenti con le morsettiere degli input e output per realizzare l'automatismo che più si preferisce. La corrispondenza segnale/colore è riferita ad un connettore a 25 poli, di cui i pin dal 1 al 22 sono riferiti ai segnali, il 23 e il 24 si rifanno al comune e il 25 alla linea passante. Di seguito, viene riportato uno schema del connettore femmina a vaschetta 25 poli presente sul blocco valvole.



**Figura 4.21:** Corrispondenza PIN per batteria di elettrovalvole monostabili montate tutte su doppie basi per monostabile



Figura 4.22: Corrispondenza PIN per batteria di elettrovalvole montate su basi in configurazione mista



Figura 4.23: Connettore femmina SUB-D 25 poli

Corrispondenza segnale/colore del cavo	
1	bianco
2	marrone
3	verde
4	giallo
5	grigio
6	rosa
7	blu
8	rosso
9	nero
10	viola
11	m grigio/rosa
12	$\mathrm{rosso/blu}$
13	bianco/verde
14	$\operatorname{marrone}/\operatorname{verde}$
15	$\mathrm{bianco/giallo}$
16	giallo/marrone
17	$\mathrm{bianco/grigio}$
18	$\operatorname{grigio}/\operatorname{marrone}$
19	$\mathrm{bianco/rosa}$
20	rosa/marrone
21	$\mathrm{bianco/blu}$
22	blu/marrone
23	$\mathrm{bianco/rosso}$
24	rosso/marrone
25	bianco/nero

Dopo aver definito in che modo vanno fatti collegamenti mediante cavo multipolare, si visualizzi all'interno del blocco di elettrovalvole in questione le diverse tipologie di valvole presenti. Nelle prime tre posizioni sono presenti delle elettrovalvole 5/2 del tipo "Solenoide-Solenoide", rappresentate dalla lettera "C". Queste elettrovalvole riportano il codice "2241.52.00.35.02", dove "02" indica "24 VDC PNP", ovvero che il carico viene collegato con l'uscita commutata e il comune (-).



Figura 4.24: Elettrovalvola bistabile di tipo Solenoide-Solenoide [30]

Nella quarta posizione è presente una elettrovalvola 5/3 centri chiusi del tipo "Solenoide-Solenoide", rappresentata dalla lettera "E". Queste elettrovalvole riportano il codice "2241.52.31.35.02", dove "02" indica "24 VDC PNP", ovvero che il carico viene collegato con l'uscita commutata e il comune (-).



Figura 4.25: Elettrovalvola bistabile di tipo Solenoide-Solenoide (5/3 centri chiusi) [30]

Dalla quinta all'ottava posizione sono presenti delle elettrovalvole 5/2 del tipo "Solenoide-Molla", rappresentata dalla lettera "A". Queste elettrovalvole riportano il codice "2241.52.00.39.02", dove "02" indica "24 VDC PNP", ovvero che il carico viene collegato con l'uscita commutata e il comune (-).



Figura 4.26: Elettrovalvola bistabile di tipo Solenoide-Molla [30]

Infine, a chiudere il blocco di elettrovalvole Optyma32-S, sono presenti un terminale sinistroautoalimentato (connessione elettrica mediante un connettore a 25 poli PNP) e un terminale destro.

#### 4.3.2 Pannello 1: Cilindri a doppio effetto

Per il pannello 1 sono stati utilizzati tre cilindri a doppio effetto targati Pneumax. In particolare, si tratta di microcilindri a norme ISO 6432 "MIR" con teste cianfrinate, con una corsa di 100 mm e un alesaggio di 20 mm.

Nella figura 4.27, di seguito riportata, è mostrato uno dei tre cilindri presenti sul pannello 1 del banco didattico Pneumax e nella tabella sottostante le caratteristiche di funzionamento e costruttive principali.



Figura 4.27: Pannello 1: Cilindro a doppio effetto

Caratteristiche principali dei cilindri a doppio effetto, serie 1200	
Pressione max di esercizio	10  bar
Fluido	aria filtrata e lubrificata
Temperatura di esercizio	da -5°C a 70°C
Alesaggio	$20\mathrm{mm}$
Corsa	$100\mathrm{mm}$
Stelo	inox
Pistone	alluminio
camicia	inox AISI 304
Testate	alluminio anodizzato

#### 4.3.3 Pannello 1: Filtro regolatore e valvola di intercettazione

Per questo pannello, si è pensato anche di inserire una batteria a 2 posti occupati da un filtro regolatore e una valvola di intercettazione, fornita sempre dall'azienda Pneumax. Nella figura successiva viene riportata la batteria presente nel pannello 1 del banco didattico.



Figura 4.28: Pannello 1: batteria composta da filtro regolatore e valvola di intercettazione

Il filtro regolatore in questione è del tipo "E" e presenta il codice "T171BEWBB". Si tratta di un filtro regolatore taglia 1, con filetti in Tecnopolimero, connessioni G1/4", la cui soglia di filtrazione è pari a 20  $\mu$ m, una gamma di regolazione da 0 a 8 bar e la direzione del flusso che va da destra verso sinistra. Di seguito vengono riportate le curve di portata, le caratteristiche di regolazione e in una tabella le caratteristiche tecniche dello stesso, presi dal catalogo della Pneumax [30].

Caratteristiche tecniche del filtro regolatore	
Pressione max di alimentazione	13  bar
Temperatura di impiego	da -5°C a +50°C
Campo di regolazione della pressione	0-4 bar
Soglia di filtrazione	$20 \ \mu m$
Connessioni	G1/4"
Attacchi manometro	G 1/8"

Per quanto concerne la valvola di intercettazione, essa è del tipo "VL" e presenta il codice "T171BVL", ovvero si tratta di una valvola di intercettazione manuale taglia 1, con filetti in



Figura 4.29: Sulla sinistra vengono riportate le curve di portata del filtro regolatore, mentre sulla destra le caratteristiche di regolazione dello stesso

Tecnopolimero e connessioni G1/4". Si tratta di una valvola a 3 vie ad otturatore ad azionamento manuale. L'apertura della stessa avviene mediante una doppia azione della manopola, ovvero spinta e rotazione in senso orario. Per quanto riguarda la chiusura con messa in scarico dell'impianto a valle, avviene mediante rotazione della manopola in senso antiorario. Di seguito viene riportata un tabella con riassunte tutte le caratteristiche tecniche dalla valvola, prese dal catalogo della Pneumax [30].

Caratteristiche funzionali della valvola di intercettazione	
Pressione max di alimentazione	13 bar
Temperatura di impiego	da -5°C a +50°C
Angolo di apertura/chiusura manopola	90°
Portata nominale a 6 bar con $\Delta$ P=1	1400  Nl/min
Portata nominale in scarico a 6 bar con $\Delta$ P=1	$550 \ \mathrm{Nl/min}$
Connessioni	G1/4"
Connessioni scarico	G 1/4"

## 4.4 Banco didattico Pneumax: parte mobile, PANNELLO 2

Si passi a questo punto alla descrizione del pannello 2. Esso prevede l'utilizzo di un pacchetto di elettrovalvole Optyma32-S (con l'aggiunta di Modulo I/O), un cilindro a doppio effetto, due trasduttori pneumo-elettrici, un filtro regolatore, una valvola di intercettazione e finecorsa pneumatici. L'idea di questo banco è quella di trasformare innanzitutto un segnale pneumatico in uno elettrico, per poi portare tali segnali al modulo di I/O della batteria Optyma32-S. Questo è utile laddove il PLC si trovasse troppo distante dal campo di lavoro: infatti così facendo, invece di portare ogni singolo segnale al PLC, si usa un unico cavo multipolare per trasmettere tutti i segnali utilizzati. Nella figura seguente viene riportato il pannello 2 con tutti i suoi componenti, i quali verranno descritti nei sottoparagrafi seguenti se non è stato già fatto in precedenza. Le informazioni riguardanti tali componenti sono stati estrapolati dal catalogo Pneumax [30] fornito direttamente dall'azienda stessa.

#### 4.4.1 Pannello 2: Elettrovalvole Optyma32-S e Modulo I/O

Anche per questo pannello sono state pensate delle elettrovalvole OPtyma32-S, ma, rispetto al pannello 1, con l'aggiunta di un modulo di I/O. Nella figura 4.31 è riportato il pacchetto di elettrovalvole utilizzato per il pannello 2 del banco didattico.



Figura 4.30: Banco didattico Pneumax, parte mobile, pannello 2



Figura 4.31: Pannello 2: elettrovalvole Optyma<br/>32-S con l'aggiunta del modulo I/O $\,$ 

Le batterie di valvole in esame offrono la possibilità di prelevare i segnali elettrici che non sono stati utilizzati sulla batteria e di renderli disponibili su un connettore a vaschetta da 25 poli femmina posizionato sul terminale destro. A questo connettore è possibile collegare o un multipolare che verrà a sua volta collegato a un'altra batteria di elettrovalvole oppure è possibile collegare uno o massimo 2 moduli di I/O sul quale è possibile portare dei segnali di ingresso o di

uscita. Ogni modulo I/O, il cui codice identificativo è "2240.08S", possiede 8 connettori femmina da M8-3 poli. È possibile, come dice il nome stesso, utilizzare questo modulo sia per input che output. Per meglio monitorare questi segnali di input/output, il modulo I/O contiene 8 LED di diagnostica, i quali si attivano nel momento in cui è presente il segnale.



Figura 4.32: Schema del modulo I/O, con particolare del connettore M8-3 poli [30]

Il connettore è stato pensato sia per collegare ingressi a 2 fili, come interruttori, finecorsa magnetici o pressostati, e sia ingressi a 3 fili, come proximity, fotocellule o finecorsa magnetici elettronici. Per quanto riguarda le uscite, invece, non sono protette dai cortocircuiti e per questo motivo bisogna fare attenzione al collegamento elettrico. Di seguito viene riportata una tabella nella quale sono riassunte tutte le caratteristiche generali di tale modulo I/O.

Caratteristiche funzionali delle elettrovalvole Optyma32-S	
Tensione Uscite	fornita dall'utente
Tensione Ingressi	dipende dall'utilizzo
Connettori di I/O	M8-3poli femmina
N. max di I/O	8
Collegamento alla batteria	diretto a vaschetta 25 poli
Temperatura ambiente	da 0°C a $+50$ °C
Grado di protezione	IP65

Per capire, invece, come vengono distribuiti i segnali raccolti dal multipolare in presenza di uno o più moduli I/O, si dia uno sguardo agli esempi riportati successivamente. Tali esempi sono stati presi sempre dal catalogo generale fornito dalla Pneumax [30].



Figura 4.33: Corrispondenza PIN per batteria di elettrovalvole con l'aggiunta di moduli I/O

Nel secondo esempio è stato adoperato un cavo multipolare a 37 poli, differentemente dal primo caso che è da 25. Per quanto riguarda la corrispondenza pin/colore del cavo multipolare, essa rimane la stessa descritta nel sottoparagrafo 4.3.1.

#### 4.4.2 Pannello 2: Cilindro a doppio effetto

Il cilindro a doppio effetto presente nel pannello 2 è anch'esso targato Pneumax e fa parte della serie 1300. In particolare, si tratta di un cilindro a norme ISO 15552 - ECOLIGHT, il cui codice identificativo è "1390.32.100.01". Il cilindro, magnetico con stelo inox, presenta un alesaggio di 32mm e una corsa pari a 100 mm.

Nella figura 4.34, di seguito riportata, è mostrato il cilindro in esame presente sul pannello 2 del banco didattico Pneumax e nella tabella sottostante le caratteristiche di funzionamento e costruttive principali.



Figura 4.34: Pannello 2: Cilindro a doppio effetto

Caratteristiche principali dei cilindri a doppio effetto, serie 1300	
Pressione max di esercizio	10 bar
Fluido	aria filtrata e lubrificata
Temperatura di esercizio	da -5°C a 70°C
Alesaggio	$32\mathrm{mm}$
Corsa	$100\mathrm{mm}$
Stelo	inox
Pistone	resina acetalica
camicia	lega allumini, anodizzato
Testate	alluminio pressofuso

#### 4.4.3 Pannello 2: Trasduttori pneumo-elettrici

Nella realizzazione di impianti di automazione pneumatica è ricorrente la necessità di condizionare determinati segnali ad un uso particolare. I due trasduttori pneumo-elettrici presenti su questo pannello, ad esempio, trasformano un segnale pneumatico in uno elettrico in quanto si ha la necessità di trattare con segnali elettrici. Tali trasduttori, commercializzati dalla Pneumax S.p.a., fanno parte della serie 900 e in particolare sono identificati dal codice "900.18.1-4". Di seguito viene riportata una figura che ritrae i due trasduttori sul pannello 2, mentre nella tabella sono riportate le caratteristiche di funzionamento.

Caratteristiche di funzionamento dei trasduttori pneumo-elettrici	
Pressione di funzionamento max	10 bar
Pressione di commutazione	4 bar
Temperatura d'impiego	da -5°C a +70°C
Connessioni di pilotaggio	G1/8"
Fluido	aria filtrata e lubrificata



Figura 4.35: Pannello 2: trasduttori pneumo-elettrici

# 4.5 Banco didattico Pneumax: parte mobile, PANNELLO 3

L'esperienza che contraddistingue il pannello numero 3 è basata sul controllo pressione di un serbatoio, targato sempre Pneumax. Il pannello<br/>3 è composto da due elettrovalvole del tipo 2/2 NC, un regolatore propor<br/>zionale, un serbatoio, un trasduttore di pressione e infine un manometro per poter controllare la pressione presente nel serbatoio. Di seguito viene riportata una foto che ritra<br/>e la terza esperienza e di seguito, nei vari sottoparagrafi, verranno spiegati singolarmente i componenti utilizzati.



Figura 4.36: Banco didattico Pneumax, parte mobile, pannello 3

#### 4.5.1 Pannello 3: Elettrovalvole pneumatiche

Gli elementi peculiari per questo pannello sono le elettrovalvole pneumatiche MATRIX del tipo 2/2 NC, in quanto grazie a queste è possibile raggiungere lo scopo di questa esercitazione. Queste elettrovalvole fanno parte della serie 820 e alimentate a 24 Volt a corrente continua presentano un tempo di riposta pari a 5 ms in apertura e 2 ms in chiusura, operando a una frequenza massima di 100 Hz. Oltretutto hanno caratteristiche dinamiche molto elevate, una portata massima di 100 l/min (ANR) e hanno un campo variabile di pressione pari a 0-8 bar. Mediante il controllo di queste elettrovalvole è possibile applicare la tecnica del PWM (Pulse Width Modulation) o la tecnica del PFM (Pulse Frequency Modulation). Di seguito viene riportata il particolare dell'elettrovalvola adoperata e di un suo spaccato per comprenderne meglio il suo funzionamento.



Figura 4.37: Sulla sinistra l'elettrovalvola pneumatica MATRIX 2/2 NC e a destra un suo spaccato ([32])

Il codice identificativo di tale elettrovalvola MATRIX è "MX-821.104C224" e nella tabella sottostante sono riassunte le sue caratteristiche principali.

Caratteristiche generali delle elettroval vole pneumatiche $2/2~{\rm NC}$	
Tempo di risposta in apertura	$5 \mathrm{ms}$
Tempo di risposta in chiusura	$2 \mathrm{ms}$
Frequenza massima	200 Hz
Soglia di filtrazione	$40 \ \mu m$
Temperatura d'impiego	da -5°C a $+50$ °C
Grado di protezione	IP 62
Fluido	aria filtrata e non lubrificata

Di seguito è invece riportato il simbolo identificativo dell'elettrovalvola.



Figura 4.38: Simbolo identificativo dell'elettrovalvola 2/2 NC

#### 4.5.2 Pannello 3: Regolatore proporzionale a controllo elettronico

Il regolatore proporzionale a controllo elettronico nasce dalla necessità di variare dinamicamente durante il funzionamento di una macchina parametri come la forza generata e la velocità con cui si muove lo stelo. Mediante un regolatore proporzionale è possibile variare nel tempo il valore di pressione controllando cosi la velocità o la forza di un attuatore. Nella figura 4.39 viene riportata la foto di un regolatore proporzionale (171E2N.T.D.0005) e di un suo schema funzionale per comprendere meglio il suo funzionamento, mentre nella tabella le sue caratteristiche pneumatiche, elettriche e funzionali.



**Figura 4.39:** Sulla sinistra il regolatore proporzionale a controllo elettronico e sulla destra un suo schema funzionale ([30])

Caratteristiche generali del regolatore proporzionale	
Pressione minima ingresso	pressione desidarata+1 bar
Pressione massima d'uscita	10  bar
Taglia	1
Portata nominale da 1 a $2$	1100  Nl/min
Portata in scarico	1300  Nl/min
Tensione di alimentazione	24  VDC
Segnale di riferimento	0-10 V
Uscita analogica Tensione	0-10 V
Uscita analogica Corrente	0-20 mA
Linearità	$<\pm$ 0,3% F.S.
Isteresi	${<}0,3\%$ F.S.
Ripetibilità	$<\pm$ 0,3% F.S.
Sensibilità	$<\pm$ 0,3% F.S.
Grado di protezione	IP 65
Temperatura di funzionamento	da -5°C a +50°C

Le connessioni pneumatiche del regolatore (G1/4") prevedono il foro di alimentazione e di scarico su un lato ed il foro per l'utilizzatore sul lato opposto, come è bene evidente nella figura 4.39. Nella parte superiore dei regolatori sono posizionate le elettrovalvole di controllo, il sensore di pressione e l'elettronica di gestione. Il regolatore proporzionale è definito *LOOP CHIUSO*, il quale è rappresentato nella figura 4.40, in quanto un trasduttore di pressione presente nel circuito trasmette un segnale analogico continuo al microprocessore il quale compara il valore di riferimento con quello rilevato e si comporta di conseguenza le elettrovalvole di controllo.



Figura 4.40: Loop chiuso del regolatore proporzionale a controllo elettronico [30]

Il collegamento elettrico avviene mediante un connettore SUB-D 15 poli femmina e tali collegamenti elettrici devono essere effettuati seguendo lo schema riportato nella figura 4.41.

...

Versione stand	ard	
PIN CONNETTORE		
1 (GRIGIO/ROSA)	= INGR. DIGITALE 1	
2 (ROSSO/BLU)	= INGR. DIGITALE 2	
3(BIANCO/VERDE)	= INGR. DIGITALE 3	$\bigcirc$
4(MARRONE/VERDE)	= INGR. DIGITALE 4	9 0 0 0 0 0 0 0 15
5(BIANCO/GIALLO)	= INGR. DIGITALE 5	
6 (MARRONE/GIALLO	) = INGR. DIGITALE 6	
7 (BLU)	= INGR. DIGITALE 7	
8(VERDE)	= INGRESSO ANALOGICO /	
	INGRESSO DIGITALE 8	
9(ROSSO)	= ALIMENTAZIONE (24 VDC)	
10(ROSA)	= USCITA DIGITALE (24 VDC PNP)	- Children
11(BIANCO)	= USCITA ANALOGICA (CORRENTE)	
12 (MARRONE)	= USCITA ANALOGICA (TENSIONE)	
13 (GIALLO)	= Rx RS-232	
14(VIOLA)	= Tx RS-232	
15(NERO+GRIGIO)	= GND	

Figura 4.41: Corrispondenza pin connettore con regolatore proporzionale [30]

N.B. Collegamenti errati possono danneggiare il dispositivo. Se l'alimentazione elettrica viene interrotta, la pressione in uscita viene mantenuta sul valore selezionato. Per scaricare il circuito a valle bisogna prima azzerare il riferimento, accertandosi dal display che il regolatore l'abbia fatto e che l'indicatore dia zero, e poi è possibile sospendere l'alimentazione elettrica.

#### 4.5.3 Pannello 3: Trasduttore di pressione

Per poter fare un controllo di pressione all'interno del serbatoio attraverso delle elettrovalvole pneumatiche a due vie e due posizioni, c'è bisogno di un dispositivo che legga la pressione attuale del serbatoio di volta in volta. Il dispositivo chiamato a svolgere questo compito è il trasduttore di pressione. Il modello in questione è distribuito dalla *Honeywell Sensing and Control* con il codice "40PC150G2A" e presenta le seguenti caratteristiche: tensione di alimentazione 5 VDC (ottenuta mediante un alimentatore a 5 Volt), non linearità pari a  $\pm 0.2\%$  F.S., isteresi e ripetibilità pari a  $\pm 0.15\%$  F.S. e un'uscita in tensione tra 0.5 e 4.5 Volt. Di seguito viene riportata la figura del trasduttore di pressione installata sul banco didattico e una tabella riassuntiva delle caratteristiche principali del trasduttore.



Figura 4.42: Trasduttore di pressione utilizzato per il pannello 3

Caratteristiche generali del trasduttore di pressione	
Tensione di alimentazione	5 VDC
Range di pressione	0-15  psi
Offset tensione	$0,5 \mathrm{V}$
Fondo scala	$4,5 \mathrm{V}$
Non-linearità	$\pm$ 0,2% F.S.
Isteresi e ripetibilità	$\pm$ 0,15% F.S.
Risposta del sistema	$1 \mathrm{ms}$

Inoltre, è stata misurata anche la sua caratteristica statica direttamente sul campo, mediante l'uso di un manometro per leggere la pressione e un multimetro per leggere la tensione corrispondente. Viene riportata, di seguito, una tabella con tutti i valori misurati di pressione e tensione e la caratteristica statica.



Figura 4.43: Caratteristica statica del trasduttore sulla sinistra e la tabella che riporta le misurazioni fatte per ricavarla sulla destra

L'equazione della caratteristica statica del trasduttore di pressione é:

$$P_{\rm V} - P_{\rm V0} = G_{\rm p} \left( P - P_0 \right) \tag{4.1}$$

dove:

- P<sub>V</sub> indica il valore di tensione corrispondente alla pressione generica P;
- $P_{V0}$  rappresenta il valore minimo di tensione corrispondente alla pressione minima (0 bar) e vale 0.5 Volt ;
- P è il valore generico di pressione in bar;
- $P_0$  è il valore di pressione minimo ottenibile ed è pari a 0 bar;
- $G_p$  rappresenta il guadagno statico del trasduttore e si misura in  $\left[\frac{V}{bar}\right]$ ;

Per ricavare il valore del guadagno, si sfrutta la tensione massima in quanto:

$$G_{\rm p} = \frac{P_{\rm VMAX} - P_{\rm V0}}{P_{\rm MAX}} \tag{4.2}$$

 $\operatorname{con}$ 

$$P_{\rm VMAX} = 2.80 V$$
$$P_{\rm V0} = 0.50 V$$
$$P_{\rm MAX} = 6 bar$$

Di conseguenza si ottiene che il guadagno statico del trasduttore è pari a:

$$G_{\rm p} = 0.38 \, \frac{V}{bar}$$

Tale valore, insieme al valore dell'offset ( $P_{V0} = 0.50 \text{ V}$ ), sarà poi utilizzato nel codice pensato per il controllo pressione.

# Capitolo 5

# Assemblaggio e programmazione della parte mobile

#### 5.1 Introduzione alla parte sperimentale

In questo capitolo si analizzeranno nel dettaglio le prime due esperienze accademiche realizzate per questo banco, descrivendo prima come è stato realizzato il programma utente da caricare nel PLC per far funzionare l'automatismo in base al progetto sviluppato e dopo la corrispondente programmazione della pagina HMI, con la quale gli studenti si interfacceranno durante le esercitazioni. Dunque, i prossimi sottoparagrafi saranno dedicati rispettivamente ai due pannelli mobili programmati, fruibili dai docenti per poter insegnare sempre qualcosa di nuovo ai futuri studenti iscritti all'indirizzo di specializzazione *Automazione* o che avranno come materia a scelta *Automazione a fluido*. Per questa prima esperienza, si è pensato di creare una sequenza di comandi che movimenti gli steli dei tre cilindri secondo un ordine ben preciso. Ad esempio, per questo pannello si è realizzato il circuito definito dal seguente diagramma movimento-fasi.



il cui Grafcet e Grafcet contratto sono riportati rispettivamente nella figura 5.1 e 5.2.



Figura 5.1: Grafcet



Figura 5.2: Grafcet contratto

# 5.2 Banco Didattico Pneumax con pannello 1

Il pannello 1, come è stato già mostrato nel capitolo precedente, è principalmente composto da un pacchetto di elettrovalvole pneumatiche di vario tipo, tre cilindri a doppio effetto, sei finecorsa magnetici, un filtro regolatore e una valvola di intercettazione. Di seguito si intende riportare una figura che presenta il banco didattico reale.



Figura 5.3: Foto banco didattico Pneumax con pannello 1

#### 5.2.1 Schema elettrico e pneumatico del banco

Come primo step, si intende riportare uno schema elettrico e pneumatico del banco didattico con il pannello 1 in modo da poter realizzare in maniera esatta i collegamenti senza sbagliarsi. A livello pneumatico, si consiglia di non alimentare l'aria compressa fino a quando non si è certi di aver inserito i tubi all'interno delle boccole delle elettrovalvole o attuatori e che tirandoli non vengano via fuori. Se così non fosse, potrebbero urtare e lesionare gli studenti o chiunque sia intorno al banco didattico. A livello elettrico, invece, bisogna far attenzione a collegare nel posto giusto i cavi perché, se così non fosse, si potrebbero danneggiare i diversi dispositivi elettrici. Dunque, di seguito, viene riportata la figura 5.4 che definisce lo schema elettrico e pneumatico del banco didattico con il pannello 1. Per una migliore comprensione dello schema elettrico e pneumatico, si consideri che:

- le linee più marcate in blu definiscono i collegamenti pneumatici, mentre quelli meno marcati di differenti colori rappresentano quelli elettrici;
- i differenti colori dei cavi elettrici si riferiscono ai colori dei cavi presenti all'interno del multipolare, in modo da non poter sbagliare i collegamenti se si guarda lo schema;
- nelle cassette degli input ed output il pallino pieno (riempito con lo stesso colore del filo) indica che è stato collegato un cavo elettrico (per risalire al cavo bisogna seguire la linea che giunge nel pallino in esame), mentre se è vuoto significa che non è stato collegato alcun connettore a banana alla presa da pannello.
- i cavi del multipolare collegati alla cassetta degli output sono:
  - cavo MARRONE-ROSSO alla presa da pannello corrispondente allo "0V";
  - cavo BIANCO alla presa da pannello corrispondente all'ingresso "Q0.0";
  - cavo MARRONE alla presa da pannello corrispondente all'ingresso "Q0.1";
  - cavo VERDE alla presa da pannello corrispondente all'ingresso "Q0.2";
  - cavo GIALLO alla presa da pannello corrispondente all'ingresso "Q0.3";
  - cavo GRIGIO alla presa da pannello corrispondente all'ingresso "Q0.4";
  - cavo ROSA alla presa da pannello corrispondente all'ingresso "Q0.5".



Figura 5.4: Schema elettrico e pneumatico del banco con il pannello 1

#### 5.2.2 Programmazione del pannello 1

Terminati i vari collegamenti, pneumatici o elettrici che siano, si passa a questo punto alla stesura del programma utente sul Tia Portal. Per prima cosa bisogna definire le variabili del PLC, le quali ovviamente, affinché l'automatismo funzioni, devono corrispondere ai collegamenti fatti fisicamente. Nella figura 5.5 viene riportata la tabella delle variabili del Tia Portal per questo pannello con i relativi indirizzi associati.

	Variab	oili PLC				
	Nome		Tabella delle variabili	Tipo di dati	Indirizzo	
1	-	START	Tabella delle variabi.	Bool	%10.0	
2	-	SELETTORE	Tabella delle variabi.	Bool	%I0.1	
З	-	EMERGENZA	Tabella delle variabi.	Bool	%10.2	
4	-	FC1-IN	Tabella delle variabi.	Bool	%10.3	
5	-	FC1-OUT	Tabella delle variabi.	Bool	%10.4	
6	-00	FC2-IN	Tabella delle variabi.	Bool	%10.5	
7	-	FC2-OUT	Tabella delle variabi.	Bool	%I0.6	
8	-	FC3-IN	Tabella delle variabi.	Bool	%10.7	
9	-	FC3-OUT	Tabella delle variabi.	Bool	%11.0	
10	-	CILINDRO 1+	Tabella delle variabi.	Bool	%Q0.0	
11	-	CILINDRO 1-	Tabella delle variabi.	Bool	%Q0.1	
12	-	CILINDRO 2+	Tabella delle variabi.	Bool	%Q0.2	
13	-	CILINDRO 2-	Tabella delle variabi.	Bool	%Q0.3	
14	-	CILINDRO 3+	Tabella delle variabi.	Bool	%Q0.4	
15	-	CILINDRO 3-	Tabella delle variabi.	Bool	%Q0.5	
16	-	M1	Tabella delle variabi.	Bool	%M1.0	
17	-00	M2	Tabella delle variabi.	Bool	%M1.1	

Figura 5.5: Tabella delle variabili presente sul Tia Portal relativa al pannello 1

Una volta definite le variabili, è possibile passare alla programmazione vera e propria. La tecnica utilizzata per scrivere il programma utente è la cosiddetta tecnica *batch*, la quale è una tecnica di programmazione Ladder che consiste nell'associare ad ogni fase un bit di memoria. In questo modo, quando la fase è attiva questo bit è positivo, altrimenti è nullo. Il diagramma Ladder viene suddiviso così in tre parti:

- il modulo per la gestione del ciclo automatico;
- il modulo dei comandi;
- il modulo di Reset.

Per questo caso specific, o i bit di memoria relativi alle 2 fasi contratte sono stati denominati M1 e M2 (rispettivamente associati alle memorie M1.0 e M1.1). Nel modulo per la gestione del ciclo automatico viene implementato in linguaggio Ladder la logica di attivazione delle memorie appena nominate. Nel modulo dei comandi, in base alla memoria attiva, viene abilitata l'attivazione delle uscite corrispondenti. Il modulo di Reset, come dice la parola stessa, si occupa del reset di tutte le memorie. Solitamente questo modulo viene adoperato per tagliare la linea all'intero impianto quando c'è qualche anomalia attraverso il pulsante di emergenza. Per questo automatismo è stato creato un blocco funzione FC, il cui programma viene riportato successivamente. Tale blocco, denominato *Blocco\_Automatismo*, permette la movimentazione degli steli dei cilindri secondo la sequenza ipotizzata.

# Blocco\_Automatismo [FC2]

Blocco_Automatismo Proprietà								
Generale								
Nome	Blocco_Automatismo		Numero		2		Тіро	FC
Linguaggio	Linguaggio KOP		Numerazione		Automatico			
Informazioni								
Titolo	Titolo		Autore				Commento	
Famiglia			Versione		0.1		ID definito dall'utente	
Nome Tipo di		dati	Valore o	li default	Comme	ento		
Input								
Output								
InOut								
Temp								
Constant								
▼ Return								
Blocco_Automatismo Void								

#### Segmento 1: Modulo per la gestione del ciclo manuale/automatico: Attivazione fase 1/ Disattivazione fase 2

Premendo il pulsante START è possibile avviare il ciclo manuale con l'attivazione della fase 1 e la disattivazione della fase 2. Se dopo aver premuto il pulsante START viene attivato anche il SELETTORE, allora si avvia il ciclo automatico.



Segmento 2: Modulo per la gestione del ciclo manuale/automatico: Attivazione fase 2 / Disattivazione fase 1



#### Segmento 3: Modulo dei comandi

Viene abilitata l'attivazione delle uscite corrispondenti alla memoria M1


### Segmento 4: Modulo dei comandi

Viene abilitata l'attivazione delle uscite corrispondenti alla memoria M2



A questo punto bisogna inserire il blocco funzionale nel MAIN del Tia Portal per poterlo caricare nel PLC; se non fosse inserito al suo interno, il programma utente risulterebbe vuoto. In particolare, prima del blocco bisogna inserire un contatto aperto associato al pulsante dell'emergenza. Così facendo, in qualunque punto del ciclo in cui il sistema si trova, andando a premere il pulsante di emergenza la movimentazione degli steli dei cilindri si blocca. Il sistema rimane fermo fino a quando non si disattiva il pulsante di emergenza. Disabilitato il tasto, il sistema riprende il ciclo dal punto in cui si era arrestato.

Segmento 1: Attivazione del ciclo manuale/automatico



Figura 5.6: Inserimento del Blocco\_Automatismo all'interno del MAIN

Dopo aver completato il MAIN, bisogna compilare e caricare il programma sul PLC mediante la procedura descritta nel capitolo 3 e a questo punto non bisogna far altro che premere il pulsante START per avviare il ciclo.

#### 5.2.3 Programmazione del HMI del pannello 1

Si supponga a questo punto di voler creare un'interfaccia uomo-macchina mediante un touch panel senza utilizzare i testi presenti sul campo. Innanzitutto, bisogna modificare il programma utente e riadattarlo anche all'HMI. Per cui prima di cominciare a programmare il pannello operatore, si crei un nuovo DB globale, denominato ad esempio *Dati\_HMI*, all'interno del quale è possibile definire tutte le variabili del pannello operatore. Di seguito viene riportato l'elenco delle variabili presenti all'interno del DB.

	Da	ti_	HMI		
		Nome		Tipo di dati	Valore di avvio
1		•	Static		
2			start	Bool	false
З			selettore	Bool	false
4	-		emergenza	Bool	false
5	-		sistema avviato	Bool	false
6	-		sistema non avviato	Bool	false
7	-		sistema in emergenza	Bool	false

Figura 5.7: Tabella delle variabili del DB Dati HMI relativa al pannello 1

Si può osservare che principalmente vengono riportate le variabili relative alla pulsantiera e questo perché si vuole ricreare sull'HMI la stessa situazione che potrei avere sul campo. Le altre tre variabili (sistema avviato, sistema non avviato, sistema in emergenza) sono state inserite per monitorare sul touch panel anche lo stato del sistema. Prima di passare alla programmazione del HMI, come è stato già anticipato, affinché il ciclo possa attivarsi anche dai tasti virtuali presenti sull'HMI, bisogna fare una piccola modifica al programma. Tale modifica consisterà nel mettere in parallelo i contatti normalmente aperti relativi ai tasti virtuali start e selettore presenti nel DB appena creato con i contatti normalmente aperti associati ai tasti fisici START e SELETTORE presenti nelle variabili del PLC. In sostanza, si andrà a modificare solamente il primo segmento del *Blocco Automatismo*, come mostrato qui di seguito.

#### Segmento 1: Modulo per la gestione del ciclo manuale/automatico: Attivazione fase 1/ Disattivazione fase 2

Premendo il pulsante START è possibile avviare il ciclo manuale con l'attivazione della fase 1 e la disattivazione della fase 2. Se dopo aver premuto il pulsante START viene attivato anche il SELETTORE, allora si avvia il ciclo automatico.



Figura 5.8: Adattamento del programma nel blocco funzione per il pannello operatore

Infine, bisogna andare ad inserire il contatto del pulsante di emergenza dell'HMI e lo si inserisce, questa volta, direttamente nel MAIN. Esso verrà riportato in serie al contatto NA dell'emergenza prima del *Blocco\_Automatismo*, così come è mostrato nella figura 5.9.

#### Segmento 1: Attivazione del ciclo manuale/automatico



Figura 5.9: Adattamento del programma nel MAIN per HMI del pannello 1

Come è visibile dal DB *Dati\_HMI*, al suo interno sono presenti altre tre variabili (Sistema avviato, Sistema non avviato e Sistema in emergenza), le quali sono state pensate per tenere sotto controllo il processo direttamente dal pannello operatore (ad esempio è possibile capire se il sistema è in movimento o se è stata attivata l'emergenza). Per questo motivo è stato creato un nuovo blocco funzione FC, denominato *Stato del sistema*. Di seguito viene riportato il modo in cui tale blocco è stato inserito nel MAIN e in che modo è stato programmato per adempiere alla sua funzione.



Figura 5.10: Aggiunta della funzione monitoraggio nel MAIN per HMI del pannello 1

# Stato del sistema [FC1]

Stato del sister	ma Proprietà							
Generale								
Nome	Stato del sistema		Numer	ю	1		Тіро	FC
Linguaggio	КОР		Numer	razione	Automatico			
Informazioni								
Titolo			Autore	!			Commento	
Famiglia			Versio	ne	0.1		ID definito dall'utente	
Nome		Tipo di	dati	Valore o	li default	Comme	ento	
Input								
Output								
InOut								
Temp								
Constant								
➡ Return								
Stato del	sistema	Void						

#### Segmento 1: Set/Reset della variabile "Sistema non avviato"

Se il sistema non è avviato, allora si setta la variabile "Sistema non avviato". Cliccando il pulsante "START" (fisicamente o dall'HMI), e quindi avviando il sistema, si resetta la variabile "Sistema non avviato".



#### Segmento 2: Attivazione della variabile "Sistema in emergenza"

Se durante il ciclo si preme il pulsante di emergenza (fisicamente o dall'HMI), si attiva la variabile "Sistema in emergenza".



#### Segmento 3: Settaggio della variabile "Sistema avviato"

Premendo il tasto START (fisicamente o dall'HMI) o prima lo START e poi il SELETTORE (fisicamente o dall'HMI) si attiva la variabile "Sistema avviato".



#### Segmento 4: Reset della variabile "Sistema avviato"

Quando il sistema arriva a fine ciclo, ovvero quando è attiva la memoria M2 e contemporaneamente lo stelo del cilindro 3 rientra, si resetta la variabile "Sistema avviato".



Dopo aver caricato il pannello operatore, come spiegato nel capitolo 3, bisogna creare la pagina all'interno della quale si andrà ad inserire tutto quello che si preferisce: nel caso in esame si andranno ad aggiungere i pulsanti virtuali, mediante i quali sarà possibile attivare/disattivare il ciclo manuale/automatico, dei cilindri animati, i cui steli saranno sincronizzati con il movimento degli steli reali e, infine, dei led per visualizzare lo stato del sistema. Per cui, dopo aver aperto il menu di sinistra, nel quale sono elencati tutti i dispositivi caricati sul progetto, e dopo aver selezionato l'HMI (in questo caso è il KTP700 Basic PN), bisogna selezionare la sottocartella *Pagine* e selezionare la voce *Aggiungere nuova pagina* come è visibile dalla figura qui sotto riportata.



Figura 5.11: Procedura per aggiungere una nuova pagina all'HMI

A questo punto, alla lista delle pagine sotto elencate nella cartella *Pagine*, si aggiungerà una nuova voce denominata *Pagina\_1*, la quale è stata rinominata successivamente *Pagina Comandi*. Andando a cliccare su questa pagina, il TIA Portal aprirà una nuova finestra che sarà quella che verrà proiettata sul pennello operatore quando questo sarà alimentato.

Innanzitutto, prima ancora di iniziare a programmare il touch panel, è stata associata la *Pagina Comandi* al secondo pulsante dell'HMI: per poterlo fare, dallo stesso menu di sinistra, si deve trascinare la nuova pagina sul secondo pulsante del pannello operatore. Se su quest'ultimo comparirà un triangolo giallo, allora il procedimento appena descritto è andato a buon fine (si guardi la figura 5.12).



Figura 5.12: Procedura per associare una nuova pagina ad un tasto dell'HMI

Ora è possibile passare alla programmazione vera e propria. Nella figura 5.13 è stata riportata la *Pagina Comandi* pensata per il pannello 1 del banco didattico Pneumax.



Figura 5.13: Anteprima sul TIA Portal della pagina dedicata al pannello 1 del banco didattico

Si parta con l'inserimento dei tre pulsanti virtuali, i quali sono disponibili nella sezione *Biblioteche* nel menu presente a destra alla finestra di lavoro. È possibile che non sia visibile tale voce: se così fosse, per visualizzarlo, bisogna cliccare sulla freccetta in basso come indicato nella figura 5.14 nel cerchietto rosso.



Figura 5.14: Procedura per selezionare la voce "Biblioteche" dalla pagina principale

Dalla sezione *Biblioteche* è possibile inserire qualsiasi oggetti all'interno della pagina, come pulsanti monostabili, pulsanti bistabili, led e altre tipologie di tasti. Nel caso in esame, bisogna andare nelle *Biblioteche Globali*, selezionare prima la voce *Buttons-and-Switches* e poi *Copie master* e infine premere su *PushbuttonSwitches*. Da questa cartella, facendo drag & drop, è possibile riportare sulla *Pagina Comandi* il tasto *Pushbutton\_Emergency* e il pulsante monostabile *Pushbutton\_Round\_G* come è indicato nell'immagine 5.15.



Figura 5.15: Inserimento del pulsante di emergenza e del tasto monostabile START

Per quanto concerne il selettore, bisogna estrapolarlo dalla cartella *RotarySwitches* nella stessa maniera. Nella figura 5.16 è riportato ciò che si dovrebbe avere a questo punto della programmazione.



Figura 5.16: Vista globale dei tre pulsanti aggiunti sulla "Pagina Comandi"

Si noti che nella figura, al di sotto dei tre pulsanti, è presente un rettangolo: esso è stato creato ancor prima di inserire i pulsanti dell'HMI dalla sezione *Casella degli strumenti*, sotto la voce *Oggetti semplici* (si guardi la figura successiva per capire come inserire un base rettangolare). Se tale sezione non è visibile, così come si è fatto per la sezione *Biblioteche*, bisogna questa volta cliccare la freccetta verso l'alto, in quanto è la prima sezione del menu del HMI.



Figura 5.17: Inserimento di una base rettangolare per i tre pulsanti

Cosi come in molti programmi di grafica, anche sul TIA Portal è possibile inserire il livello di ogni oggetto. Quando si parla di livelli, si intende la possibilità di sovrapporre un oggetto su un altro, tenendo a mente che il "Livello 0" fa da base a tutti gli altri livelli: ad esempio, considerando due oggetti, un rettangolo e un cerchio, e avendo associato ad ognuno rispettivamente il livello 0 e 1, il risultato finale sarà un cerchio sovrapposto su un rettangolo. Per poter selezionare i livelli dei diversi oggetti, bisogna recarsi sulle proprietà dell'oggetto in esame e selezionare la voce Varie.

Il prossimo step è associare ad ogni oggetto la variabile HMI corrispondente. Innanzitutto, esistono procedure diverse in base alla tipologia di pulsante aggiunto, ovvero in base al fatto che sia un pulsante monostabile, come lo START, o bistabile, come l'EMERGENZA o il SELETTORE.

Si parta abbinando il pulsante START alla variabile *start* del DB *Dati\_HMI*. Dopo aver selezionato il pulsante verde sulla *Pagina Comandi* dell'HMI, bisogna andare sulle sue proprietà e andare nella sezione *Eventi*. A questo punto, bisogna cliccare sulla voce *Premi* e aggiungere

nella tabella la funzione ImpostaBitConTastoAttivato che è una funzione presente nella sezione Elaborazione di bit. Dopo aver selezionato questa funzione, bisogna inserire la variabile a cui si vuole associare il pulsante, che in questo caso sarà lo start del DB Dati\_HMI. Nella figura 5.18 viene riportata la procedura numerata.



Figura 5.18: Procedura per associare un pulsante monostabile ad una variabile HMI

Per i pulsanti bistabili, invece, questa procedura è più semplice in quanto bisogna direttamente inserire la variabile corrispondente. Per cui, dopo aver selezionato il pulsante, ad esempio, di EMERGENZA, bisogna andare sulle sue proprietà e andare nella sezione *Proprietà*. A questo punto, dalla voce generale, bisogna inserire direttamente la variabile nel campo dedicato, come mostrato nella figura 5.19. In questa maniera, quando si andrà a premere il pulsante di emergenza, esso rimarrà attivo fino a quando non lo si ripreme per disattivarlo. Lo stesso procedimento bisogna effettuare con il selettore.

Pushbutton_Emergency	[Interruttore]	🖳 Proprietà	🛄 Informazioni	追 🗓 Di	agnostica	18-
Proprietà Agimazi	oni Eventi	Testi				
📑 Elenco delle proprietà	Generale					
Generale 3 Conformazione	Processo			M	odo	
Motivo di riempimento	Variabile:	Dati_HMI_emergenza	1		Formato:	Interruttore con
Layout	Variabile PLC:	Dati HMI emergenza				
Rappresentazione	4	Dud_millemengenzu		Gr	afica	
Formato del testo	Indirizzo:		Bool		unica	
Limiti	Valore per ON:	1			ON:	Pushbutton_E
Stili/Design					OFF:	Pushbutton E
Varie						- Losson
Sicurezza						
	<					>

Figura 5.19: Procedura per associare un pulsante bistabile ad una variabile HMI

Un procedimento del tutto simile a quello fatto per i pulsanti bistabili è possibile farlo per monitorare il sistema. Per cui ad ogni led è stata associata la variabile corrispondente aprendo direttamente le proprietà del led e inserendo la variabile correlata. Il risultato finale è stato riportato nella figura 5.20.



Figura 5.20: Vista globale dei led di monitoraggio del sistema sulla "Pagina Comandi"

In particolare:

- si accederà il led bianco quando il sistema è fermo;
- si accenderà il led verde quando il sistema è in movimento;
- si accenderà il led rosso quando il sistema è stato bloccato dall'attivazione dell'emergenza.

Per completare la *Pagina Comandi*, si è pensato di aggiungere l'animazione degli steli dei tre cilindri sincronizzati in real-time con quelli presenti sul campo, come riportato nella figura 5.21.



Figura 5.21: Vista globale dell'animazione degli steli dei tre cilindri sulla "Pagina Comandi"

Per quanto riguarda le immagini dei cilindri e degli steli, sono state prese dalle grafiche presenti sul TIA Portal. Per poter accedere a queste grafiche bisogna selezionare, sempre dal menu di destra, prima la voce *Casella degli strumenti* e poi *Grafiche*, come indicato nella figura 5.22. In questa sezione si troveranno un sacco di grafiche predefinite per ogni campo ingegneristico da poter utilizzare sull'HMI.



Figura 5.22: Procedura per poter selezionare all'interno di una pagina HMI delle grafiche predefinite

D'ora in poi verrà descritto il procedimento per poter effettuare tale animazione. Si consideri che il procedimento utilizzato per il cilindro 2 potrà essere replicato anche sugli altri due cilindri, utilizzando però le giuste variabili. Di seguito, mediante elenco puntato, verranno definiti i diversi step da eseguire per poter ottenere lo stesso risultato.

1. Dalla sezione *Grafiche* inserire l'immagine di uno stelo nella *Pagina Comandi* come rappresentato nella figura 5.23



Figura 5.23: Inserimento di uno stelo nella "Pagina Comandi"

2. Cliccare sullo stelo appena posizionato e settare nelle sue proprietà, sulla voce Rappresentazione, la posizione e le dimensioni a seconda delle esigenze richieste. Nel caso in esame, come indicato nella figura 5.24, lo stelo è stato fissato alla coordinata X=113 e Y=315.

Vista grafica_12	[Vista grafica]				Rroprietà	1 Informazioni	<b>i</b>	Diagnostica	
Proprietà	Animazioni	Eventi	Testi						
📑 Elenco delle pro	oprietà 🛛 R	appresentazi	one						
Generale Conformazione Rappresentazion Varie		Posizione 8 X: 1: Y: 3:	dimensio	oni ₩₩ 189 I 21	¢	Adatta dimensior Non adattare au Adatta grafica a Adatta oggetto a Sincronizza dime	<b>i</b> toma dime Ila gi nsioi	ticamente nsioni oggetto rafica più grande ni oggetto con grafica	

Figura 5.24: Settaggio della posizione e delle dimensioni sulla "Pagina Comandi" del primo stelo

3. Dalle stesse proprietà, cliccare sulla sezione Animazioni e aggiungere dalla voce Vista una nuova animazione cliccando su Visibilità. Nella sezione Processo inserire come variabile "CILINDRO 2+" e spuntare la voce Area facendola variare tra 0 e 1. Infine, spuntare nel campo Visibilità la voce Visibile, come è stato fatto nella figura 5.25. In questa maniera lo stelo sarà visibile quando la variabile "CILINDRO 2+" varierà tra 0 e 1, ovvero sarà sempre visibile.

Vista grafica_12 [Vista grafic	ca]	💁 Proprietà	🗓 Informazioni 🚺 🗓 🛙	Diagnostica 🛛 🗐 🗉 🤜
Proprietà Animazioni	Eventi Testi			
Vista generale  Vista  Aggiungi nuova ani  Visibilitä  Movimenti	Visibilità Processo Variabile: CILINDRO 2+ • Area Da: 0 A: 1 • Bit singolo 0		Visibilità Visibile Non visibile	

Figura 5.25: Impostazione dell'animazione e visibilità del primo stelo sulla "Pagina Comandi"

4. Dalla sezione *Grafiche* inserire l'immagine un secondo stelo delle stesse dimensioni nella *Pagina Comandi* come rappresentato nella figura 5.26.



Figura 5.26: Inserimento di un secondo stelo nella "Pagina Comandi"

5. Cliccare sullo stelo appena posizionato e settare nelle sue proprietà, sulla voce *Rappresentazione*, la posizione e le dimensioni a seconda delle esigenze richieste. Nel caso in esame,

come indicato nella figura 5.27, avendo supposto una corsa di 100 mm, lo stelo è stato fissato alla coordinata X=213 e Y=315.

Vista grafica_14 [Vista gra	ifica]	<b>Proprietà</b>	🗓 Informazioni 追 🗓 Diagnostica	<b>-</b>
Proprietà Animazio	ni Eventi Testi			
📑 Elenco delle proprietà	Rappresentazione			
Generale	Posizione & dimension		Adatta dimensioni	1
Conformazione	rosizione a uniension		Adatta dimensioni	
Rappresentazione	X: 213 🗢	₩₩ 189	Non adattare automaticamente	
Varie	Y: 315 🗘	21	Adatta grafica a dimensioni oggetto	
	<b>4</b>		O Adatta oggetto alla grafica più grande	
	•		O Sincronizza dimensioni oggetto con grafica	

Figura 5.27: Settaggio della posizione e delle dimensioni sulla "Pagina Comandi" del secondo stelo

6. Dalle stesse proprietà, cliccare sulla sezione Animazioni e aggiungere dalla voce Vista una nuova animazione cliccando su Visibilità. Nella sezione Processo inserire come variabile "FC2-OUT" (finecorsa magnetico che si attiva quando lo stelo è fuoriuscito dal cilindro) e spuntare la voce Area facendola variare tra 1 e 1. Infine, spuntare nel campo Visibilità la voce Visibile, come è stato fatto nella figura 5.28. In questa maniera lo stelo sarà visibile quando la variabile "FC2-OUT" sarà 1, ovvero quando lo stelo del cilindro sarà fuoriuscito dal cilindro 2.

Vista grafica_14 [Vista grafica]		🖳 Proprietà	🗓 Informazioni 🔒 🗓 Diagnostica	
Proprietà Animazioni	Eventi Testi			
Vista generale ▼ 🕆 Vista interpretational series animazione Interpretational series animazione Interpretatione Interpretationale series animazione Interpretationali se	Processo Variabile: FC2-OUT		Visibilità Visibile Non visibile	
▶ 🦨 Movimenti	Area     Bit singolo	Da: 1 •		

Figura 5.28: Impostazione dell'animazione e visibilità del secondo stelo sulla "Pagina Comandi"

7. Dalla sezione *Grafiche* inserire l'immagine un terzo stelo delle stesse dimensioni nella *Pagina Comandi* come rappresentato nella figura 5.29.



Figura 5.29: Inserimento di un terzo stelo nella "Pagina Comandi"

8. Cliccare sullo stelo appena posizionato e settare nelle sue proprietà, sulla voce Rappresentazione, la posizione e le dimensioni a seconda delle esigenze richieste. Esattamente come il secondo stelo, anch'esso è stato fissato alla coordinata X=213 e Y=315. Dopodiché cliccare la sezione Animazioni e aggiungere dalla voce Vista una nuova animazione cliccando su Visibilità. Nella sezione Processo inserire come variabile "CILINDRO 2-" e spuntare la voce Area facendola variare tra 1 e 1. Infine, spuntare nel campo Visibilità la voce Visibile, come è stato fatto nella figura 5.30. In questa maniera lo stelo sarà visibile quando la variabile "CILINDRO 2-" sarà 1, ovvero da quando lo stelo del cilindro 2 sta rientrando nel cilindro fino al reset della memoria "M2".

Vista grafica_6 [Vista grafica]		🖳 Proprietà	🗓 Informazioni 🔒 🗓 Diagnostica	
Proprietà Animazioni	Eventi Testi			
	Visibilità			
Vista generale ▼ ∰ Vista ∰ Aggiungi nuova animazione ♥ Visibilită ♥ Visibilită	Processo Variabile: CILINDRO 2-		Visibilità Visibile Non visibile	
r 🖉 Movimenu	Bit singolo	Da: 1		

Figura 5.30: Impostazione dell'animazione e visibilità del terzo stelo sulla "Pagina Comandi"

9. A differenza dei precedenti steli, il terzo e il quarto stelo dovranno simulare lo spostamento dello stelo: in particolare, questo stelo dovrà simulare il rientro dello stelo nel cilindro. Prima di fare ciò, dal menu di sinistra cliccare prima su Variabili HMI e poi Visualizza tutte le variabili. Si aprirà una tabella all'interno della quale saranno presenti tutte le variabili HMI create e a queste bisognerà aggiungere la nuova variabile "moto 2 ritorno". Dopo aver creato questa nuova variabile, bisogna selezionare la variabile HMI "CILINDRO 2-" e andare sulle sue proprietà. A questo punto bisogna cliccare sulla sezione Eventi e aggiungere la funzione DefinisciVariabile e selezionare come variabile quella creata precedentemente, ovvero "moto 2 ritorno". Nella figura 5.31 viene riportato come è stata inserita la funzione "DefinisciVariabile".

CILINDRO 2- [Variabile_HN	AI]	🔍 Proprietà	🗓 Informazioni 🔒 児 Diagnostica	∎∎ ▼
Proprietà Eventi	Testi			
	±∓⊟≣×			
Cambio valore				
	Variabile (Uscita)		moto 2 ritorno	
	Valore		0	
	<aggiungi funzione=""></aggiungi>			
	•			

Figura 5.31: Settaggio della variabile virtuale "moto 2 ritorno" per garantire il movimento orizzontale dello stelo

10. A questo scopo, cliccare sempre sulla sezione Animazioni e aggiungere dalla voce Movimenti una nuova animazione cliccando su Movimento orizzontale. A questo punto si aggiunge la variabile che in questo caso è stata chiamata "moto 2 ritorno", la quale verrà creata

successivamente nelle variabili dell'HMI, con un'area compresa tra 0 e 5 e un delta tra la posizione finale e quella iniziale pari a 100 (esattamente come il valore della corsa dello stelo), come è stato fatto nella figura 5.32.

Vista grafica_6 [\	/ista grafica]				🔍 Proprietà	1 Informazi	oni 追 🗓	Diagnostica	
Proprietà /	Animazioni	Eventi	Testi						
		Movir	mento orizz	ontale					
Vista generale		Va	riabilo						
🕶 🖀 Vista		Va	nabile						
📑 Aggiungi nu	uova animazione	6	N	ome: mo	to 2 ritorno				
👁 Visibilità			Are	a da: 0	•				
🛨 🎤 Movimenti		4		A: 5					
📑 Aggiungi nu	uova animazione								
I+II Movimento	orizzontale	Pos	sizione iniz	tiale		Posizion	e di destina	zione	
			X: 213			x.	105		
			Y: 315	<b></b>		Y:	315 📮		

Figura 5.32: Impostazione del movimento orizzontale sulla "Pagina Comandi" del terzo stelo

11. Dalla sezione *Grafiche* inserire l'immagine un quarto stelo delle stesse dimensioni nella *Pagina Comandi* come rappresentato nella figura 5.33.



Figura 5.33: Inserimento di un quarto stelo nella "Pagina Comandi"

12. Cliccare sullo stelo appena posizionato e settare nelle sue proprietà, sulla voce Rappresentazione, la posizione e le dimensioni a seconda delle esigenze richieste. Esattamente come il secondo stelo, anch'esso è stato fissato alla coordinata X=113 e Y=315. Dopodiché cliccare la sezione Animazioni e aggiungere dalla voce Vista una nuova animazione cliccando su Visibilità. Nella sezione Processo inserire come variabile "CILINDRO 2+" e spuntare la voce Area facendola variare tra 1 e 1. Infine, spuntare nel campo Visibilità la voce Visibile, come è stato fatto nella figura 5.34. In questa maniera lo stelo sarà visibile quando la variabile "CILINDRO 2+" sarà 1, ovvero da quando lo stelo del cilindro 2 sta fuoriuscendo dal cilindro fino al reset della memoria "M1".

Vista grafica_5	Vista grafica_5 [Vista grafica]						🗓 Informazioni 🔋 🗓 Diagno:	stica 📑 🗖 🗖 🤝
Proprietà	Animazioni	Eventi	Testi					
		Visibi	lità					
Vista gener ▼ 🐏 Vista 💣 Aggiungi	Pro	ocesso ariabile:				Visibilità  Visibile		
👁 Visibilità		- C	CILINDRO 2+				🔘 Non visibile	
Movimenti			Area	Da:	1	\$		
		•		A:	1	٩		
			Bit singolo		0			

Figura 5.34: Impostazione dell'animazione e visibilità del quarto stelo sulla "Pagina Comandi"

13. Anche il quarto stelo dovrà simulare lo spostamento dello stelo: in questo caso, lo stelo dovrà simulare la fuoriuscita dello stelo dal cilindro. Prima di fare ciò, dal menu di sinistra cliccare prima su Variabili HMI e poi Visualizza tutte le variabili. Si aprirà una tabella all'interno della quale saranno presenti tutte le variabili HMI create e a queste bisognerà aggiungere la nuova variabile "moto 2". Dopo aver creato questa nuova variabile, bisogna selezionare la variabile HMI "CILINDRO 2+" e andare sulle sue proprietà. A questo punto bisogna cliccare sulla sezione Eventi e aggiungere la funzione DefinisciVariabile e selezionare come variabile quella creata precedentemente, ovvero "moto 2". Nella figura 5.35 viene riportata la schermata in cui è presente la tabella delle variabili HMI e come è stata inserita la funzione "DefinisciVariabile".

Navigazione del progetto	Progetto Tesi Magistrale	→ HMI_1 [KTP700 Basic PN] → Va	ariabili HMI		_ # =×
Dispositivi					
	2				
	Variabili HMI				
🕨 🔀 Traces	Nome	Tabella delle variabili	Tipo di dati	Collegamento Nome PLC	
🕨 🔯 Comunicazione OPC UA	CILINDRO 1-	Tabella delle variabili star	ndard Bool	HMI_Collegame PLC_1	^
Dati proxy dei dispositivi	CILINDRO 1+	Tabella delle variabili star	ndard Bool	HMI_Collegame PLC_1	=
Informazioni sul programma	CILINDRO 2-	Tabella delle variabili star	ndard Bool	HMI_Collegame PLC_1	
Elenchi di testi di segnalazion.	CILINDRO 2+	Tabella delle variabili star	ndard 🔽 Bool	HMI Collega PLC 1	
🕨 🧊 Moduli locali	CILINDRO 3-	Tabella delle variabili star	ndard Bool	HMI_Collegame PLC_1	
HMI_1 [KTP700 Basic PN]	CILINDRO 3+	Tabella delle variabili star	ndard Bool	HMI_Collegame PLC_1	
Configurazione dispositivi	Dati_HM_emerge	anza Tabella delle variabili star	ndard Bool	HMI_Collegame PLC_1	~
V. Online & Diagnostica	1				>
🍸 Impostazioni di runtime					
🕨 🛅 Pagine	Segnalazioni a bit	Segnalazioni analogiche Var	iabili di archivio		
Gestione pagine	CILINDRO 2+ [Variabile ]	нмі	Descriptà 21 Inf		
🕶 🔚 Variabili HMI	CIENTO 21 [Valiable_]			ormazioni 🚺 🖸 Diagnostica	
a Visualizza tutte le variabili	Proprietà Eventi	Testi			
📑 Aggiungi nuova tabella de					
💥 Tabella delle variabili stan					
2 Collegamenti	Cambio valore				
🖂 Segnalazioni HM		<ul> <li>DefinisciVariabile</li> </ul>			
📑 Ricette		Variabile (Uscita)	moto 2		
Archivi		Valore	0		
5 Schedulazione		<aggiungi funzione=""></aggiungi>			
Elenchi di grafiche e testi					
👔 Gestione utenti		- Fl			
Dispositivi non raggruppati					
🕨 🔚 Impostazioni Security					
A State and the limit dates of	<u> </u>				
	-		11		
Vista dettagli					

Figura 5.35: Settaggio della variabile virtuale "moto 2" per garantire il movimento orizzontale dello stelo

14. A questo scopo, cliccare sempre sulla sezione Animazioni e aggiungere dalla voce Movimenti una nuova animazione cliccando su Movimento orizzontale. A questo punto si aggiunge la variabile che in questo caso è stata chiamata "moto 2", la quale verrà creata successivamente nelle variabili dell'HMI, con un'area compresa tra 0 e 5 e un delta tra la posizione finale e quella iniziale pari a 100 (esattamente come il valore della corsa dello stelo), come è stato fatto nella figura 5.36.

Vista grafica_5 [Vista grafica]		🖳 Proprietà	🗓 Informazioni 🔋 🗓 Diagnostica	18
Proprietà Animazioni	Eventi Testi			
	Movimento orizzontale	\ <u></u>		
Vista generale	Variabilo			
🕨 🖀 Vista	valiablie			
🕶 🎤 Movimenti	Nome:	moto 2		
💣 Aggiungi nuova animazione	Area da:	0		
HI Movimento orizzontale	- A:	5		
	Posizione iniziale		Posizione di destinazione	
	X: 114 🖨		X: 222	
	Y: 315		Y: 315	

Figura 5.36: Impostazione del movimento orizzontale sulla "Pagina Comandi" del quarto stelo

15. Infine, per completare l'animazione, dalla sezione *Grafiche* inserire l'immagine un cilindro nella *Pagina Comandi* come rappresentato nella figura 5.37. Differentemente dai quattro

steli, il cilindro, dovendo coprire gli steli, avrà un livello superiore rispetto a quest'ultimi. Per cui se agli steli si è imposto un livello pari a 1, al cilindro bisogna imporre almeno il livello 2.



Figura 5.37: Inserimento del cilindro nella "Pagina Comandi"

Svolti questi step per il cilindro 2 e ripetendo la stessa procedura anche per gli altri due cilindri, il risultato finale che si ottiene è una replicazione dei movimenti degli steli presenti sul campo. Con l'aggiunta di questa animazione, la *Pagina Comandi* risulta completa e pronta per essere caricata sul pannello operatore mediante il procedimento descritto nel capitolo 3. Così facendo, si potrà comandare l'intero sistema direttamente dall'HMI senza fare uso dei pulsanti presenti sul campo e monitorare il circuito anche a distanza visualizzando sia lo stato del sistema mediante i led, ma anche attraverso l'animazione presente sullo schermo del touch panel, la quale replica in maniera efficace ciò che succede sul pannello 1.

## 5.3 Banco Didattico Pneumax con pannello 2

Il pannello 2, come è stato già mostrato nel capitolo precedente, è principalmente composto da un pacchetto di elettrovalvole pneumatiche di vario tipo con l'aggiunta di un modulo I/O, un cilindro a doppio effetto, due finecorsa pneumatici, due trasduttori pneumo-elettrici, un filtro regolatore e una valvola di intercettazione. Di seguito si intende riportare una figura che presenta il banco didattico reale.



Figura 5.38: Foto banco didattico Pneumax con pannello 2

#### 5.3.1 Schema elettrico e pneumatico del banco

Come primo passo, si intende riportare uno schema elettrico e pneumatico del banco didattico con il pannello 2 in modo da poter fare in maniera esatta i collegamenti senza sbagliarsi. A livello pneumatico, si consiglia di non alimentare l'aria compressa fino a quando non si è certi di aver inserito i tubi all'interno delle boccole delle elettrovalvole o attuatori e che tirandoli non vengano via fuori. Se così non fosse, potrebbero urtare e lesionare gli studenti o chiunque sia intorno al banco didattico. Per questo particolare pannello, affinché i trasduttori pneumo-elettrici funzionino bene, la pressione di alimentazione dell'intero sistema deve essere fissato almeno a 4 bar. A livello elettrico, invece, bisogna far attenzione a collegare nel posto giusto i cavi perché, se così non fosse, si potrebbero danneggiare i diversi dispositivi elettrici. Dunque, di seguito, viene riportata la figura 5.39 che definisce lo schema elettrico e pneumatico del banco didattico con il pannello 2. Per una migliore comprensione dello schema elettrico e pneumatico, si consideri che:

- le linee più marcate in blu definiscono i collegamenti pneumatici, mentre quelli meno marcati di differenti colori rappresentano quelli elettrici;
- i differenti colori dei cavi elettrici si riferiscono ai colori dei cavi presenti all'interno del multipolare, in modo da non poter sbagliare i collegamenti se si guarda lo schema;
- nelle cassette degli input ed output il pallino pieno (riempito con lo stesso colore del filo) indica che è stato collegato un cavo elettrico (per risalire al cavo bisogna seguire la linea che giunge nel pallino in esame), mentre se è vuoto significa che non è stato collegato alcun connettore a banana alla presa da pannello.
- i cavi del multipolare collegati alla cassetta degli input sono:
  - cavo BIANCO-NERO alla presa da pannello corrispondente al "24V";
  - cavo BIANCO-VERDE alla presa da pannello corrispondente all'ingresso "I1.0";
  - cavo MARRONE-VERDE alla presa da pannello corrispondente all'ingresso "I1.1".
- i cavi del multipolare collegati alla cassetta degli output sono:
  - cavo MARRONE-ROSSO alla presa da pannello corrispondente allo "0V";
  - cavo BIANCO alla presa da pannello corrispondente all'ingresso "Q0.0";
  - cavo MARRONE alla presa da pannello corrispondente all'ingresso "Q0.1".

Inoltre, si può osservare dallo schema, differentemente dal pannello precedente, che i collegamenti elettrici vengono effettuati tutti mediante cavo multipolare. Questo è possibile grazie alla presenza del modulo I/O, il quale utilizza i segnali liberi messi a disposizione dal pacchetto di elettrovalvole Optyma32-S. Tutti questi segnali, raccolti in un unico cavo, vengono poi portati alla parte fissa del banco didattico Pneumax.



Figura 5.39: Schema elettrico e pneumatico del banco con il pannello 2

#### 5.3.2 Programmazione del pannello 2

Terminati i vari collegamenti, pneumatici o elettrici che siano, si passa a questo punto alla stesura del programma utente sul Tia Portal. Per prima cosa bisogna definire le variabili del PLC, le quali ovviamente, affinché l'automatismo funzioni, devono corrispondere ai collegamenti fatti fisicamente. Nella figura 5.40 viene riportata la tabella delle variabili del Tia Portal per questo pannello con i relativi indirizzi associati.

1	Varia	bili PLC			
		Nome	Tabella delle variabili	Tipo di dati	Indirizzo
1	-	START	Tabella delle variabi.	Bool	%10.0
2	-	SELETTORE	Tabella delle variabi.	Bool	%I0.1
3	-	EMERGENZA	Tabella delle variabi.	Bool	%10.2
4	-	FC1-IN	Tabella delle variabi.	Bool	%11.0
5	-	FC1-OUT	Tabella delle variabi.	Bool	%11.1
6	-	CILINDRO 1+	Tabella delle variabi.	Bool	%Q0.0
7	-	CILINDRO 1-	Tabella delle variabi.	Bool	%Q0.1
8	-	M1	Tabella delle variabi.	Bool	%M1.0
9	-	M2	Tabella delle variabi.	Bool	%M2.0
10	-	MB	Tabella delle variabi.	Bool	%M3.0
11	-	Clock_2.5Hz	Tabella delle variabi.	Bool	%M0.2

Figura 5.40: Tabella delle variabili presente sul Tia Portal relativa al pannello 2

Dopo aver definito le variabili del PLC, è possibile passare alla stesura del programma utente. Per il pannello 2, sono stati definiti due blocchi funzione: *Automatismo\_Temporizzato* e *Automatismo\_Con\_Contatore*.

Il blocco Automatismo\_Temporizzato è stato studiato per inserire all'interno del ciclo di lavoro un temporizzatore (TON) e, in particolare, utilizzato per ritardare il rientro dello stelo del cilindro 1. Il ciclo temporizzato viene attivato solo e soltanto se viene premuto il pulsante START.

Il blocco Automatismo\_Con\_Contatore, invece, presenta all'interno del programma utente un contatore e tale ciclo viene attivato mediante un SELETTORE. Di seguito viene riportata la modalità in cui questi blocchi sono stati aggiunti nel MAIN e in modo sono stati programmati.



Figura 5.41: Aggiunta dei due blocchi funzione all'interno del MAIN per il pannello 2

# Automatismo\_Temporizzato [FC3]

Automatismo_	Temporizzato Pro	prietà						
Generale								
Nome	Automatismo_Ten zato	nporiz-	Nume	ro	3		Тіро	FC
Linguaggio	КОР		Nume	razione	Automatico			
Informazioni								
Titolo			Autore	9			Commento	
Famiglia			Versio	ne	0.1			
Nome		Tipo di	dati	Valore	di default	Comm	ento	
Input								
Output								
InOut								
Temp								
Constant								
➡ Return								
Automat to	tismo_Temporizza-	Void						

#### Segmento 1: Avvio del ciclo temporizzato

Mediante il pulsante START è possibile attivare il ciclo temporizzato



#### Segmento 2: Fuoriuscita dello stelo del cilindro 1



### Segmento 3: Rientro dello stelo del cilindro 1

Lo stelo del cilindro 1 rientra dopo che il temporizzatore raggiunge il valore impostato dal PT



# Automatismo\_Con\_Contatore [FC4]

Automatismo_	Con_Contatore Pr	oprietà						
Generale								
Nome	Automatismo_Cor tatore	_Con-	Nume	ro	4		Тіро	FC
Linguaggio	КОР		Nume	razione	Automatico			
Informazioni								
Titolo			Autore	9			Commento	
Famiglia			Versio	ne	0.1		ID definito dall'utente	
Nome		Tipo di	dati	Valore o	di default	Comme	ento	
Input								
Output								
InOut								
Temp								
Constant								
🔻 Return								
Automat tore	ismo_Con_Conta-	Void						

#### Segmento 1: Avvio ciclo con contatore

Mediante il SELETTORE è possibile attivare il ciclo che contiene il contatore.



#### Segmento 2: Fuoriuscita dello stelo del cilindro 1



#### Segmento 3: Rientro dello stelo del cilindro 1

Ogni volta che lo stelo rientra nel cilindro, il contatore incrementa di un'unità. Quando raggiunge il valore del PV, si attiva la bobina M3, la quale va a bloccare l'intero ciclo. Per resettare il contatore bisogna disattivare il SELET-TORE.



Si può osservare che nel blocco Automatismo\_Temporizzato è stato inserito un temporizzatore con un valore di PT (durata temporizzatore) pari a 5000 ms, ovvero 5 secondi. Nel blocco Automatismo\_Con\_Contatore, invece, è stato inserito nel contatore un valore di PV (preset value) pari a 3: questo significa che dopo tre volte che lo stelo del cilindro rientra, viene attivata la bobina M3, come indicato nel programma utente. A questo punto, è possibile caricare il programma nel PLC e testare il funzionamento del sistema.

#### 5.3.3 Programmazione del HMI del pannello 2

Si supponga a questo punto di voler creare un'interfaccia uomo-macchina mediante un touch panel senza utilizzare i testi presenti sul campo. Innanzitutto, bisogna modificare il programma utente e riadattarlo anche all'HMI. Per cui prima di cominciare a programmare, si crei un nuovo blocco dati DB globale, denominato ad esempio *Dati\_HMI*, all'interno del quale è possibile definire tutte le variabili del pannello operatore. Di seguito viene riportato l'elenco delle variabili presenti all'interno del DB.

	Da	ti_	HMI		
		No	me	Tipo di dati	Valore di avvio
1		•	Static		
2			start	Bool	false
3			selettore	Bool	false
4	-		emergenza	Bool	false
5			sistema avviato	Bool	false
6			sistema non avviato	Bool	false
7	-		sistema in emergenza	Bool	false
8	-00		attesa temporizzatore	Bool	false

Figura 5.42: Tabella delle variabili del DB Dati\_HMI relativa al pannello 2

Si può osservare che principalmente vengono riportate le variabili relative alla pulsantiera e questo perché si vuole ricreare sull'HMI la stessa situazione che potrei avere sul campo. Le altre quattro variabili (sistema avviato, sistema non avviato, sistema in emergenza, attesa temporizzatore) sono state inserite per monitorare sul touch panel anche lo stato del sistema. Prima di passare alla programmazione del HMI, come è stato già anticipato, affinché il ciclo possa attivarsi anche dai tasti virtuali presenti sull'HMI, bisogna fare una piccola modifica al programma. Tale modifica consisterà nel mettere in parallelo i contatti normalmente aperti relativi ai tasti virtuali start e selettore presenti nel DB appena creato con i contatti normalmente aperti associati ai tasti fisici START e SELETTORE presenti nelle variabili del PLC. In sostanza, si andrà a modificare solamente il primo segmento del blocco Automatismo\_Con\_Contatore, come mostrato nella figura 5.43, e il primo segmento del blocco Automatismo\_Con\_Contatore, come mostrato nella figura 5.44.



Figura 5.43: Adattamento del programma nel blocco funzione Automatismo Temporizzato per HMI



Figura 5.44: Adattamento del programma nel blocco funzione Automatismo Con Contatore per HMI

Infine, bisogna andare ad inserire il contatto del pulsante di emergenza dell'HMI e lo si inserisce, questa volta, direttamente nel MAIN. Esso verrà riportato in in serie al contatto NA dell'emergenza prima dei due blocchi Automatismo\_Temporizzato e Automatismo\_Con\_Contatore, così come è mostrato nella figura 5.45.





Figura 5.45: Adattamento del programma nel MAIN per HMI del pannello 2

Come è visibile dal DB *Dati\_HMI*, al suo interno sono presenti altre quattro variabili (Sistema avviato, Sistema non avviato, Sistema in emergenza e attesa temporizzatore), le quali sono state pensate per tenere sotto controllo il processo direttamente dal pannello operatore (ad esempio è possibile capire se il sistema è in movimento o se è stata attivata l'emergenza oppure se il sistema è attivo anche se apparentemente è fermo per via del temporizzatore). Per questo motivo è stato creato un nuovo blocco funzione FC, denominato *Stato del sistema*. Di seguito viene riportato il modo in cui tale blocco è stato inserito nel MAIN e in che modo è stato programmato per adempiere alla sua funzione.



Figura 5.46: Aggiunta della funzione monitoraggio nel MAIN per HMI del pannello 2

# Stato del sistema [FC1]

Constant **Return** 

Stato del sistema

Stato del siste	ma Proprietà							
Generale								
Nome	Stato del sistema		Nume	ro	1		Тіро	FC
Linguaggio	КОР	КОР		razione	Automatico	Automatico		
Informazioni								
Titolo			Autore	3			Commento	
Famiglia			Versio	ne	0.1		ID definito dall'utente	
Nome		Tipo di	dati	Valore o	di default	Comm	ento	
Input								
Output								
InOut								
Temp								

## Segmento 1: Set/Reset della variabile "Sistema non avviato"

Void

Se il sistema non è avviato, allora si setta la variabile "Sistema non avviato". Cliccando il pulsante "START" (fisicamente o dall'HMI), e quindi avviando il sistema, si resetta la variabile "Sistema non avviato".



#### Segmento 2: Attivazione della variabile "Sistema in emergenza"

Se durante il ciclo si preme il pulsante di emergenza (fisicamente o dall'HMI), si attiva la variabile "Sistema in emergenza".

"D er	ati_HMI". %10.2 nergenza "EMERGENZA"	"Dati_HMI". "sistema in emergenza"

#### Segmento 3: Settaggio della variabile "Sistema avviato"

Premendo il tasto START (fisicamente o dall'HMI) o prima lo START e poi il SELETTORE (fisicamente o dall'HMI) si attiva la variabile "Sistema avviato".



#### Segmento 4: Reset della variabile "Sistema avviato"

Quando il sistema arriva a fine ciclo, ovvero quando è attiva la memoria M2 e contemporaneamente lo stelo del cilindro 3 rientra, si resetta la variabile "Sistema avviato".



#### Segmento 5: Attivazione variabile "attesa temporizzatore"

Quando si attiva il temporizzatore, il led collegato con la variabile "attesa temporizzatore" inizia a lampeggiare.



Dopo aver caricato il programma modificato per il pannello operatore, come spiegato nel capitolo 3, bisogna creare la pagina all'interno che verrà strutturata come meglio si crede per il progetto appena realizzato: nel caso in esame si andranno ad aggiungere i pulsanti virtuali, mediante i quali sarà possibile attivare/disattivare i due differenti cicli, dei led per visualizzare lo stato del sistema e, infine, due campi I/O per visualizzare in real-time il tempo del temporizzatore e il conteggio portato avanti dal contatore. Di seguito viene mostrata l'anteprima della *Pagina Comandi* dell'HMI pensata per il pannello 2.



Figura 5.47: Anteprima sul TIA Portal della pagina dedicata al pannello 2 del banco didattico

Per la creazione della pagina HMI, della pulsantiera virtuale e dei led, si guardi la spiegazione esposta nel paragrafo 5.2.3. Per quanto riguarda quest'ultimi, però, rispetto al pannello 1, ce n'è uno che funziona ad intermittenza. Per poter creare un led di questo tipo, è stato inserito nel blocco funzione *Stato di Sistema* un contatto NA associato a un *Merker di Clock*, che in questo caso corrisponde alla memoria "M0.2", denominato *Clock 2.5 Hz*. Per poter attivare i merker di clock, bisogna andare sulle proprietà del PLC e spuntare la casella "Attiva l'utilizzo del byte del merker di clock", come mostrato nella figura 5.48.

Generale	Variabile	10	Costanti di sistema	Testi		
Generale Interfaccia PR DI 14/DQ 10	OFINET [X1]	~	Sempre 1 () Sempre 0			
Al 2 Signal board A Contatori velo Generatori di	AQ1 ici (HSC)		Bit del merker di clock		🖌 Attiva l'utilizzo del byte del merker di clock	
Avviamento Ciclo Carico di com	unicazione		Indirizzo del byte del mer clock ( Clock 1	kerdi MBx): 0 Hz:	0 %M0.0	
Merker di cloc Server web Lingue dispor	k e di sistema nibili per la		Clock 2 Clock	5 Hz: 5 Hz: 2 Hz:	96M0.2 (Clock_2.5Hz) 96M0.3	
Protezione & S Controllo di co	Security onfigurazione	~	Clock 1.2 Clock Clock 0.62	25 Hz: 1 Hz:	96M0.4 96M0.5 95M0.6	

Figura 5.48: Attivazione del merker di clock di un PLC

In questa maniera, la bobina associata alla variabile "attesa temporizzatore" verrà attivata

in modo intermittente con una frequenza di 2.5 Hz. Ovviamente per poter associare il led alla variabile appena discussa si segue sempre lo stesso iter fatto per gli altri led, come esposto nel paragrafo 5.2.3.

Per inserire l'oggetto Campo I/O, che consente l'immissione e la visualizzazione dei valori di processo, bisogna andare sulla sezione Casella degli strumenti e selezionare la voce Elementi. In questo caso specifico, sono stati programmati due diversi campi I/O: il primo per visualizzare il tempo del temporizzatore e il secondo il conteggio aggiornato del contatore.

Dopo aver trascinato sulla *Pagina Comandi* i due campi I/O, bisogna associare loro le variabili che contengano al loro interno rispettivamente il tempo attuale del temporizzatore e il conto aggiornato del contatore. Aprendo le proprietà dei due campi I/O, per il temporizzatore, bisogna associare la variabile "Timer 1.ET" e come formato di rappresentazione "9999", mentre, per il contatore, la variabile "Contatore 1.CV" e come formato di rappresentazione "9", come mostrato nella figura 5.49.

Eventi	Testi				
enerale					
Characterization 200					
Processo				Formato	
Flocesso				Tornato	
Variabile:	Contatore 1_C	V		Formato visualizzazione:	Decimale
Variabile PLC:	"Contatore 1"	CV.		Numeri decimali:	0
vanabne ree.	contortore r .			Humen deeman.	
Indirizzo:		Int		Lunghezza del campo:	1
				Zeri iniziali:	
Tipo				Formato rappresentaz.:	9
Modo:	Ingresso/Usci	Jscita 💌			
	Processo Variabile: Variabile PLC: Indirizzo: Tipo Modo:	Processo Variabile: Contatore 1_C Variabile PLC: Contatore 1* Indirizzo: Tipo Modo: Ingresso/Usci	Processo Variabile: Contatore 1_CV Variabile PLC: *Contatore 1*.CV Indirizzo: Int Tipo Modo: Ingresso/Uscita	Processo Variabile: Contatore 1_CV Variabile PLC: *Contatore 1*.CV * Indirizzo: Int Tipo Modo: Ingresso/Uscita •	Processo     Formato       Variabile:     Contatore 1_CV        Variabile PLC:     *Contatore 1*.CV     Mumeri decimali:       Indirizzo:     Int     Lunghezza del campo:       Tipo     Zeri iniziali:     Formato rappresentaz.:       Modo:     Ingresso/Uscita     Variabile

Figura 5.49: Inserimento dell'elemento "Campo I/O" nella "Pagina dei comandi"

In questo modo sarà possibile visualizzare in real-time il tempo e il computo dei cicli fatti dallo stelo del cilindro.

# Capitolo 6

# Controllo pressione di un serbatoio

## 6.1 Banco Didattico Pneumax con pannello 3

Fino a questo momento sono stati sviluppati programmi utili al controllo di semplici sistemi automatici utilizzando ingressi e uscite digitali del PLC. In questo capitolo, invece, viene effettuato il controllo pressione all'interno un serbatoio mediante l'utilizzo di ingressi analogici e uscite digitali. Per quanto concerne la tipologia di controllo realizzato per il pannello 3 è di tipo PWM (*Pulse Width Modulation*) e fisicamente viene effettuato mediante due valvole digitali 2/2 NC della ditta fornitrice *Matrix*.

La struttura fisica realizzata per l'esperienza 3 non si sviluppa verticalmente come le precedenti, ma è stata costruita sulla piattaforma orizzontale del banco didattico come mostrato nella figura 6.1. Questo è dovuto al fatto che i componenti principali di tale sistema sono sprovvisti di particolari agganci, come magneti, che ne permettano il fissaggio sui telai mobili. Il sistema è costituito da un alimentatore di tensione regolabile, un serbatoio Pneumax, un manometro, un sensore di pressione e due valvole digitali 2/2 NC.



Figura 6.1: Foto banco didattico Pneumax con pannello 3

#### 6.2 Schema elettrico e pneumatico del banco

Così come è stato fatto per i precedenti pannelli, anche per il pannello 3 si intende riportare uno schema elettrico e pneumatico del banco didattico in modo da non poter sbagliare i vari collegamenti. A livello pneumatico, si consiglia di non alimentare l'aria compressa fino a quando non si è certi di aver inserito i tubi all'interno delle boccole delle elettrovalvole o attuatori e che tirandoli non vengano via fuori. Se così non fosse, potrebbero urtare e lesionare gli studenti o chiunque sia intorno al banco didattico. A livello elettrico, invece, bisogna far attenzione a collegare nel posto giusto i cavi perché, se così non fosse, si potrebbero danneggiare i diversi dispositivi elettrici. Nella pagina successiva, viene riportata la figura 6.2 che definisce lo schema elettrico e pneumatico del banco didattico con il pannello 3. Per una migliore comprensione dello schema elettrico e pneumatico, si consideri che le linee più marcate in blu definiscono i collegamenti pneumatici, mentre quelli meno marcati di differenti colori rappresentano quelli elettrici.

## 6.3 La modulazione PWM

Un segnale PWM, acronimo inglese di *Pulse Width Modulation* che significa "*Modulazione a variazione della larghezza d'impulso*", è un'onda quadra generata da valvole modulanti, le quali vengono fatte funzionare a una frequenza fissa e un periodo di ciclo T.

$$f = \frac{1}{T} \tag{6.1}$$

 $\cos$ 

$$f = Frequenza \ e \ T = Periodo$$

Ciò che viene variato nel ciclo è il tempo di apertura " $t_{ON}$ " e il tempo di chiusura " $t_{OFF}$ " delle valvole. In particolare, viene modificato il *Duty Cycle*, che si definisce come il rapporto tra il tempo di apertura e il periodo T. Questo rapporto può variare tra 0 (valvola sempre chiusa) e 1 (valvola sempre aperta).

$$dc = \frac{t_{\rm ON}}{T} \tag{6.2}$$

$$t_{\rm ON} = T - t_{\rm OFF} \tag{6.3}$$

 $\operatorname{con}$ 

$$dc = Duty Cycle, t_{ON} = Tempo di apertura e t_{OFF} = Tempo di chiusura$$

Ad esempio:

- se il "Duty Cycle" è pari al 50%, si avrà in uscita un'onda quadra simmetrica che assumerà per metà periodo valore 1 e la restante metà 0;
- se il "Duty Cycle" è pari al 25%, si avrà in uscita un'onda quadra che assumerà per un quarto di periodo valore 1 e per il restante periodo 0;
- se il "Duty Cycle" è pari al 100%, si avrà in uscita un segnale sempre alto, ovvero con valore sempre pari a 1;
- se il "Duty Cycle" è pari al 0%, si avrà in uscita un segnale sempre basso, ovvero con valore sempre pari a 0.



Figura 6.2: Schema elettrico e pneumatico del banco con il pannello 3

# 6.4 Generazione di un'onda quadra

Per poter applicare la tecnica PWM, bisogna, innanzitutto, scrivere un programma che generi un'onda quadra da modulare in seguito. Due sono state le soluzioni trovate per risolvere questo tipo di problema e la prima delle due é riportata nella figura 6.3.



Figura 6.3: Generazione di un'onda quadra, programma 1

Come è possibile osservare, il codice presenta tre rung. Nel primo e nel secondo rung sono presenti due diversi TON (*Ritardo all'inserzione*), i quali richiedono come input uno specifico PT (*Preset Time*), ovvero il tempo di attivazione dell'uscita "Q" di quello stesso temporizzatore. In particolare, il TON del primo rung, " $ton_3$ ", è anticipato da un contatto normalmente chiuso associato all'uscita del TON del secondo rung, " $ton_4$ "; il TON del secondo rung, " $ton_4$ ", invece, è anticipato da un contatto normalmente aperto associato all'uscita del TON del primo rung, " $ton_4$ "; il terzo rung è invece utilizzato per visualizzare l'onda quadra appena realizzata sull'uscita "Q0.3".



Figura 6.4: Esempi di diverse onde quadre utilizzando il primo codice tentativo

Nella figura 6.4 sono riportati rispettivamente i seguenti esempi:

- se i entrambi PT fossero pari a 1000 ms, si avrebbe sull'uscita "Q0.3" un'onda quadra simmetrica di periodo pari a 2000 ms;
- se il primo PT fosse pari a 500 ms e e il secondo pari a 1500 ms, in uscita si avrebbe sempre un'onda quadra di periodo pari a 2000 ms, ma in questo caso non simmetrica: essa assumerà valore "1" per i primi 1500 ms e valore "0" per i successivi 500 ms;
- invertendo i valori dei due PT del caso precedente si avrà un'onda complementare a quella appena descritta, ovvero l'onda assumerà valore "1" per i primi 500 ms e valore "0" per i successivi 1500 ms.

La seconda soluzione ricavata per generare un'onda quadra è, invece, riportata nella figura 6.5.



Figura 6.5: Generazione di un'onda quadra, programma 2

Si può osservare che, a differenza del primo codice, questo presenta due rung. Anche qui sono presenti due diversi TON, i quali sono stati posizionati in parallelo e anticipati entrambi da un contatto normalmente chiuso associato all'uscita del primo TON, " $ton_5$ ". Il secondo rung è utilizzato per visualizzare l'onda quadra appena realizzata sull'uscita "Q0.4". La bobina "Q0.4" è anticipata da un contatto normalmente chiuso associato all'uscita del secondo TON, " $ton_6$ ". A differenza del primo metodo, affinché in uscita si abbia un'onda quadra, il PT del secondo TON deve essere minore o massimo uguale al PT del primo TON, mentre il PT del primo TON deve essere pari al periodo dell'onda quadra. Per cui volendo realizzare nello stesso ordine le onde quadre generate mediante il primo codice, bisogna assegnare:

- al PT del primo TON 2000 ms e al secondo 1000 ms;
- al PT del primo TON 2000 ms e al secondo 1500 ms;
- al PT del primo TON 2000 ms e al secondo 500 ms;

#### 6.5 Modulazione di un'onda quadra

Per poter effettuare il controllo pressione mediante la tecnica PWM, lo step successivo è modulare l'onda quadra. A questo scopo, è stata introdotta all'interno del programma una

variabile di tipo "Real", denominata "*Duty Cycle*", che potrà assumere valori compresi tra 0 e 1. Moltiplicando quest'ultima per il periodo T dell'onda quadra, si otterrà il valore del tempo di apertura della valvola, " $t_{ON}$ ", il quale verrà fatto coincidere con il valore del PT del secondo TON dei programmi utili alla generazione dell'onda quadra (ton\_4 e ton\_6). Nella figura 6.6 viene riportata la parte di programmazione in cui sono presenti queste nuove variabili.



Figura 6.6: Modulazione Onda quadra

Dalla figura 6.6 è possibile notare che prima e dopo la moltiplicazione, avvenuta grazie al box "MUL", è presente il blocco "T\_CONV", il quale permette la conversione di un dato di tipo "Time" in uno di tipo "Int" e viceversa. Questi sono stati inseriti in quanto non è possibile moltiplicare direttamente un dato di tipo "Time" con uno di tipo "Real", mentre è consentito farlo tra un dato "Int" e uno "Real". In questa maniera, dunque, andando ad applicare questa procedura ai due codici precedentemente descritti e variando opportunamente il valore della variabile "*Duty Cycle*", è possibile modulare le due onde quadre generate. In particolare, supponendo di fissare un periodo T di 2 s:

- se la variabile "*Duty Cycle*" vale "0", si avrà in uscita un segnale sempre basso, ovvero con valore sempre pari a 0;
- se la variabile "*Duty Cycle*" vale "0.25", si avrà in uscita un'onda quadra non simmetrica che assumerà per un quarto di periodo T valore "1" e per il restante tempo "0";
- se la variabile "*Duty Cycle*" vale "0.50", si avrà in uscita un'onda quadra simmetrica che assumerà per mezzo periodo valore "1" e per il restante mezzo "0";
- se la variabile "*Duty Cycle*" vale "0.75", si avrà in uscita un'onda quadra simmetrica non simmetrica che assumerà per i primi tre quarti del periodo T valore "1" e per il restante tempo "0";
- se la variabile "*Duty Cycle*" vale "1", si avrà in uscita un segnale sempre alto, ovvero con valore sempre pari a 1.
Gli stessi cinque esempi, vengono riportati graficamente nel medesimo ordine, nella figura 6.7.



Figura 6.7: Esempi di diverse onde quadre al variare del "Duty Cycle"

Ponendo l'attenzione sull'ultimo grafico, ovvero quello in cui il "*Duty Cycle*" è posto pari a 100%, ci si può accorgere che ad ogni periodo T la curva mostra degli impulsi. Per quello che si vuole realizzare, nonché un controllo PWM, questo andamento non potrebbe andare bene in quanto a Duty Cycle pari a 100% la valvola dovrebbe essere sempre aperta. Per ovviare a questo problema si è pensato di generare prima un'onda quadra per estrapolarne il periodo e con quello creare un treno di impulsi. La modulazione, invece, verrà effettuata prolungando gli impulsi appena generati in base alle esigenze richieste dal sistema. Per realizzare un programma di questo tipo, si è modificato il primo codice utilizzato per la generazione dell'onda quadra riportato nel paragrafo precedente. Di seguito, vengono riportati nella figura 6.8 i nuovi grafici dettati dal nuovo programma e nella pagina successiva il nuovo codice che tiene conto anche della nuova variabile "*Duty Cycle*".



Figura 6.8: Esempi di diverse onde quadre al variare del "Duty Cycle", programma finale

# 1\_onda\_quadra [FC1]

1_onda_quadra Proprietà								
Generale								
Nome	1_onda_quadra		Numer	0	1		Тіро	FC
Linguaggio	КОР		Numer	razione	Automatico			
Informazioni								
Titolo			Autore	9			Commento	
Famiglia			Versio	ne	0.1		ID definito dall'utente	
Nome		Tipo di	dati	Valore o	li default	Comme	ento	
Input								
Output								
InOut								
Temp								
Constant								
🔻 Return								
1_onda_	quadra	Void						

# Segmento 1: Definizione del t\_ON

Per poter calcolare il t\_ON, bisogna moltiplicare un valore compreso tra 0 e 1 per il periodo T dell'onda quadra generata.

Prima di questa moltiplicazione, bisogna convertire il periodo T da dato di tipo "Time" in "Int". Dopo la moltiplicazione

invece bisogna riconvertirlo in un dato di tipo "Time" per poterlo inserire nel TON.

### Segmento 1: Definizione del t\_ON



### Segmento 2: Calcolo del mezzo periodo

Dovendo lavorare con un onda simmetrica di periodo T, si è pensato di dividere il periodo in 2 parti uguali e impostare per entrambi i TON

i mezzi periodi.



### Segmento 3: Generazione Onda quadra simmetrica con 2 TON e generazione treno di impulsi

Avendo impostato ai due TON lo stesso PT, verrà generata un'onda quadra simmetrica. Viene attivato prima il TON\_1 e allo scadere del tempo impostato viene attivato il TON\_2. Quando anche quest'ultimo termina di contare, si apre il contatto "TON\_2.Q" e viene riattivato il ciclo.

Attraverso il secondo rung è possibile creare un treno di impulsi a frequenza costante. Ciò è reso possibile dall'attivazione della bobina "Q0.0" e dall'apertura del contatto NC "Q0.0". La bobina viene attivata quando l'uscita del TON\_1 si attiva.

ATTENZIONE ALLA DISPOSIZIONE DEI RUNG (il PLC legge dall'alto verso il basso e da sinistra verso destra).



## Segmento 4: Modulazione dell'onda quadra

Mediante il TOF in questione è possibile modulare l'onda quadra sfruttando gli impulsi generati nel segmento 2. Ad esempio, in questo caso, avendo messo un PT pari a1 secondo, si è generata un'onda simmetrica di periodo 2 secondi.



Analizzando nel complesso l'intero programma, è possibile notare che i PT dei diversi TON sono sempre fissi contrariamente a quello che accadeva nei primi due codici; questo perchè nel programma completo costruito per il controllo pressione, il periodo T dell'onda quadra non rimaneva mai costante e, in particolare, diminuiva con il passare del tempo. Come seconda grossa modifica, si può osservare che nel segmento 3, oltre i soliti TON, sono presenti altre due diramazioni parallele al secondo TON. Ciò è stato fatto per generare un treno di impulsi avente lo stesso periodo dell'onda quadra definita dai due temporizzatori. L'impulso è dettato dal contatto normalmente chiuso legato all'uscita "Q0.0", il quale si apre subito dopo l'attivazione dell'uscita del "TON 1": infatti, così facendo, nel primo ciclo di scansione subito dopo l'attivazione dell'uscita del primo TON, l'uscita "Q0.1" viene attivata, mentre nel secondo ciclo viene disattivata grazie all'attivazione dell'uscita "Q0.0" che va ad aprire il contatto in serie all'uscita "Q0.1". Infine, nell'ultimo segmento, è possibile notare un blocco TOF (Ritardo alla disinserzione), il quale ritarda il reset dell'uscita Q della durata PT parametrizzata e che nel caso in esame si occupa della modulazione dell'onda quadra. L'attivazione impulsiva dell'uscita "Q0.1", fa funzionare il TOF e attiva l'uscita "Q0.2", sulla quale si potrà visualizzare l'onda modulata. Il PT del TOF è dettato dalla variabile "Duty Cycle" che permette la modulazione dell'onda in base alle esigenze del sistema.

# 6.6 Modulazione PWM mediante il blocco "CTRL PWM"

Un ulteriore modo per generare e modulare un'onda quadra sul TIA Portal è adottando il blocco denominato "CTRL\_PWM". L'istruzione "CTRL\_PWM" consente di attivare e disattivare tramite il software un generatore di impulsi supportato dalla CPU. L'identificativo HW del generatore di impulsi che si intende controllare con questa istruzione deve essere specificato sull'ingresso PWM. Il generatore di impulsi viene attivato quando è impostato il bit dell'ingresso ENABLE dell'istruzione. Se ENABLE presenta il valore TRUE, il generatore crea impulsi caratterizzati dalle proprietà definite nella Configurazione dispositivi, mentre se l'ENABLE viene resettato o se la CPU commuta in STOP, il generatore di impulsi viene disattivato e non vengono più generati impulsi.

Generale	Variabile IO	Costanti di sistema Testi	
Generale Interfaccia PR DI 14/DQ 10 AI 2 Signal board Contatori velo Generatori di PTO 1/PWM Parame Uscite H Indirizzi PTO 2/PWMC PTO 3/PWME PTO 3/PWME PTO 3/PWME	OFINET [X1] AQ1 AQ1 impulsi (PTO/PWM) trizzazione hardware di I/O 2 4	Generale     Attiva     Attiva     Informazioni sul progetto     Nome: Pulse_1     Commento:	

Figura 6.9: Attivazione del generatore di impulsi "PTO1/PWM1"

Per poter attivare un generatore di impulsi bisogna selezionare dalle proprietà della CPU, come mostrato nella figura 6.9, la voce *Generale* della sezione PTO1/PWM1 e premere sulla casella "attiva questo generatore di impulsi".

Dalla voce *Parametrizzazione* della stessa sezione è possibile caratterizzare il generatore a proprio piacimento, selezionando la tipologia del segnale (PWM), il periodo di ciclo, la durata del periodo iniziale e le loro rispettive unità di misura, come mostrato nella figura 6.10.

Generale	Variabile IO	C	ostanti di sistema	Testi			
Generale Interfaccia PRO	DFINET [X1]		<ul> <li>Parametrizzazione</li> <li>Opzioni di impulso</li> </ul>				
AI 2			opzioni di impuist	<b>`</b>			
Signal board /	AQ1		Тіро	di segnale:	PWM		-
Contatori velo	ci (HSC)		Bas	e di tempo:	Millisecondi		-
Generatori di ▼ PTO1/PWM1	impulsi (PTO/PWM)		Formato dura	ta impulso:	Centesimo		-
General	e		Ten	po di ciclo:	1000	ms 🖨	
Parame	trizzazione	•	Durata impu	lso iniziale:	0	Centesimo 🖨	
Uscite h	ardware				Consenti m	odifica del tempo di ciclo durante	
Indirizzi	di I/O				l'esecuzion	e	
PTO2/PWM2							
► PTO3/PWM3							
PTO4/PWM4							
Avviamento							

Figura 6.10: Parametrizzazione del generatore di impulsi "PTO1/PWM1"

Dalla voce *Uscite hardware* è, invece, possibile selezionare l'uscita fisica dalla quale leggere il segnale dell'onda quadra modulata. In questo caso, come mostrato nella figura 6.11, è stata scelta l'uscita "Q0.0" in quanto è una delle quattro uscite veloci che può lavorare fino a una frequenza di 100 kHz. Le altre tre uscite veloci sono la "Q0.1", "Q0.2" e la "Q0.3".

Generale	Variabile IO	Costanti di sistema Testi
Generale	-	
Interfaccia PR	OFINET [X1]	Scite hardware
DI 14/DQ 10		
AI 2		Uscita impulsi: %Q0.0 Uscita onboard 100 kHz
Signal board	AQ1	
Contatori velo	eci (HSC)	
Generatori di	impulsi (PTO/PWM)	
➡ PTO1/PWM1	l.	Attiva ussita di direzione
General	e	
Parame	trizzazione	
Uscite h	ardware	Uscita di direzione: %Q0.1 Uscita onboard 100 kHz
Indirizzi	di I/O	
► PTO2/PWM2	2	
► PTO 3/PWM3	3	
▶ PTO4/PWM4	4	
Avviamento	~	
:		

Figura 6.11: Uscite hardware del generatore di impulsi "PTO1/PWM1"

Infine, dalla voce Ingressi di I/O, è possibile visualizzare l'indirizzo iniziale e finale di uscita ("QW1000" nel caso in esame) e utilizzare lo stesso per modulare l'onda quadra.

Generale	Variabile IO	Costanti di sistem	na Testi			
Generale		> Indirizzi di I/C	)			
DI 14/DQ 10		Indirizzi di us	cita			
AI 2						
Signal board	AQ1	=	Indirizzo iniziale:	1000	.0	
Contatori velo	ci (HSC)		Indirizzo finale:	1001	.7	
Generatori di	impulsi (PTO/PWM)	Blog	co organizzativo:	(Aggiornament	o automatico)	
▼ PTO1/PWM1	I	4		( ) ggionnaire		
General	e	Imma	gine di processo:	Aggiornamento a	utomatico	•••
Parame	trizzazione	-				
Uscite h	ardware					
Indirizzi						
PTO2/PWM2	2					
PTO 4/PMM	1					
Avviamento						

Figura 6.12: Indirizzi di I/O del generatore di impulsi "PTO1/PWM1"

Dopo aver attivato il generatore di impulsi dalle proprietà della CPU, si è scritto, mediante il blocco "CTRL\_PWM", un programma capace di generare e modulare manualmente un'onda quadra mediante una variabile definita "*Duty Cycle*". Tale programma è riportato nella pagina successiva con i relativi commenti. Osservando il codice, nel primo rung è presente proprio il blocco "CTRL\_PWM", il quale richiede in ingresso il PWM, che nel caso in esame è il "Local\_Pulse\_1" (si tratta di un tipo di dato "HW\_PWM" e rappresenta una costante di sistema associata al numero 265), e un ENABLE, associato all'ingresso "I0.0". Se quest'ultimo dovesse essere disattivo, di conseguenza lo sarà anche il generatore di impulsi.

Nel secondo segmento sono presenti invece due blocchi: il primo è uno "SCALE\_X" il quale scala la variabile "*Duty Cycle*", compresa tra 0 e 1, tra 0 e 100 e un "MOVE" che sposta tale valore nella variabile di uscita "QW1000" per permettere al generatore di impulsi di modificare l'onda quadra. La variabile"*Duty Cycle*" è stata scalata tra 0 e 100 perché durante la caratterizzazione del PWM, nella sezione *Parametrizzazione*, la durata dell'impulso è stata impostata in centesimi.

Dopo aver scritto il programma, bisogna caricarlo nel PLC e verificare che sull'uscita "Q0.0" esca effettivamente un'onda quadra modulata al variare del "*Duty Cycle*". Solitamente quando si attiva un generatore di impulsi o lo si modifica, durante il caricamento nel PLC, compare un avviso relativo alla mancanza di un livello di protezione, come mostrato nella figura 6.13. Se così non fosse, allora il generatore rimarrebbe disattivato e sull'uscita "Q0.0" non si visualizzerà nulla.

		🔍 Proprietà	i, Informazi	oni	Diagnostica
Generale 😧 Riferimenti i	ncrociati Compila Sintassi				
🚹 🕦 Visualizza tutti i messa	ıgi 🔽				
Percorso -	Descrizione	Vaia	7 Errore	Avorisi	Ora
▼ PLC 1	besenzione	7	0	1	17:04:50
<ul> <li>Configurazione hardware</li> </ul>		7	0	1	17:04:50
<ul> <li>\$7-1200 station_1</li> </ul>		7	0	1	17:04:50
<ul> <li>Telaio di montagg</li> </ul>	io_0	~	0	1	17:04:50
▼ PLC_1		~	0	1	17:04:50
▼ PLC_1			0	1	17:04:50
	PLC_1 non contiene un livello di protezione conf	igurato 🗡			17:04:50
<ul> <li>Blocchi di programma</li> </ul>		7	0	0	17:04:50
	Non è stato compilato alcun blocco. Tutti i blocc	hi sono attuali.			17:04:50
	Compilazione terminata (errori: 0; avvisi: 1)				17:04:50

Figura 6.13: Errore di protezione nel caricamento del programma nel PLC

### PWM [FC6]

PWM Proprietà								
Generale								
Nome	PWM		Numer	o	6		Тіро	FC
Linguaggio	КОР		Numer	azione	Automatico			
Informazioni								
Titolo			Autore				Commento	
Famiglia			Versione		0.1		ID definito dall'utente	
Nome		Tipo di	dati	Valore o	di default	Comme	ento	
Input								
Output								
InOut								
Temp								
Constant								
▼ Return								
PWM		Void						

### Segmento 1: Creazione dell'onda quadra

Il blocco "CTRL\_PWM" permette di generare un'onda quadra di periodo T costante sul canale "Local~Pulse\_1" o "265" (tipologia dati "HW\_PWM"). Tale blocco, in questo caso, viene attivato dall'ingresso "I0.0".



#### Segmento 2: Modulazione dell'onda quadra

In ingresso al blocco "SCALE\_X" vi è la variabile "Duty Cycle" (numero reale tra 0 e 1), la quale viene scalata, appunto, tra 0 e 100 e il suo risultato, tramite il blocco "MOVE", viene spostato nella variabile "QW1000". Questa operazione consentirà la modulazione dell'onda quadra.



#### Segmento 3: Visualizzazione Onda Quadra Modulata

Il segmento 3 consente la visualizzazione sull'uscita fisica "Q0.0", come impostato dalle proprietà della CPU.



# 6.7 Controllo pressione con blocco "CTRL PWM"

Dopo aver esaminato le diverse soluzioni capaci di riprodurre un'onda quadra modulata, il prossimo step è utilizzare una di esse per controllare la pressione di un serbatoio, fine ultimo di questo capitolo.

Per poter effettuare un controllo ad anello chiuso, come quello eseguito su questo pannello, un tassello importante è rappresentato dal sensore di pressione, il quale fornisce un feedback che ci permette di capire quanto è grande l'errore rispetto al setpoint richiesto. Nella figura 6.14 è riportato lo schema a blocchi del controllo realizzato in questo paragrafo.



Figura 6.14: Schema a blocchi del controllo pressione senza controllore

Questo errore sarà proporzionale al tempo di apertura delle valvole 2/2 presenti all'interno del sistema e la variazione del t<sub>ON</sub> garantirà la modulazione delle due diverse onde quadre. Dovendo controllare una valvola per l'alimentazione e una per lo scarico, sono stati utilizzati due blocchi "CTRL PWM", in quanto ogni blocco prevede la modulazione di un unico segnale.

Osservando il codice realizzato per questo tipo di controllo, riportato dalla pagina successiva, si nota che il programma tiene conto sia del passaggio da una pressione più bassa a una più alta e viceversa; dunque, in base al fatto che il SETPOINT sia maggiore o inferiore al FEEDBACK, si attiva o il segmento 4 o il 5, facendo lavorare di conseguenza o la valvola di alimentazione o quella di scarico.

In particolare, per poter compiere una modulazione completa, nel calcolo del "Duty Cycle", bisogna tener conto da che pressione si parte e a che pressione si vuole arrivare. Se la pressione relativa all'interno del serbatoio è di 0 bar e al sistema si chiede di raggiungere un qualsiasi altro valore di pressione, allora, affiché il "Duty Cycle" sia inizialmente pari a 1, l'errore calcolato tra SETPOINT e FEEDBACK deve essere diviso per il SETPOINT stesso. Se, invece, la pressione relativa all'interno del serbatoio diversa da 0 bar e si chiede al sistema di raggiungere un valore di pressione maggiore di quello presente, allora l'errore calcolato tra SETPOINT e FEEDBACK deve essere diviso per l'errore stesso, per avere inizialmente "Duty Cycle" pari a 1.

La parte centrale del codice è quella già nota e spiegata nei paragrafi precedenti, ovvero legata alla generazione e modulazione delle due diverse onde quadre utilizzando i blocchi "CTRL\_PWM" messi a disposizione dal TIA Portal. Per questo controllo pressione, sono stati utilizzati rispettivamente il "Local\_Pulse\_2" (266) e il "Local\_Pulse\_3" (267) per la valvola di alimentazione e quella di scarico, mentre è stato associato all'ENABLE lo stesso indirizzo "I0.0" corrispondente al selettore.

In fondo al programma, invece, sono presenti tre rung il cui compito è quello di attivare la valvola di alimentazione, attivare quella di scarico e disattivare entrambe se l'errore presente nel sistema è inferiore a un certo range fissato nei blocchi "IN RANGE".

# PWM [FC6]

PWM Proprietà	L							
Generale								
Nome	PWM		Numer	ro	6		Тіро	FC
Linguaggio	КОР		Numer	razione	Automatico			
Informazioni								
Titolo			Autore				Commento	
Famiglia			Versio	ne	0.1		ID definito dall'utente	
Nome		Tipo di	dati	Valore o	di default	Comme	ento	
Input								
Output								
InOut								
Temp								
Constant								
🔻 Return								
PWM		Void						

### Segmento 1: Lettura del sensore di pressione

Tale segmento legge un segnale proveniente dal sensore di pressione (segnale analogico) e lo trasforma in un segnale in tensione mediante i due blocchi "NORM\_X" e "SCALE\_X". I successivi due blocchi, ovvero "SUB" e "DIV", dopo aver studiato la caratteristica del sensore, trasformano il valore in tensione in pressione.





### Segmento 2: Parametri utili alla determinazione del denominatore del SEGMENTO 4

In tale segmento si generano una variabile di confronto e i due valori di range min e max. Queste variabile serviranno nel segmento successivo per determinare il denominatore da utilizzare nel segmento 4 per la valutazione del Duty Cycle.



Segmento 2: Parametri utili alla determinazione del denominatore del SEGMENTO 4

#### Segmento 3: Scelta del denominatore adatto nel caso aumenti la pressione nel serbatoio

Se il SETPOINT è maggiore del FEEDBACK, si attiva il segmento 3. In particolare, per avere un Duty Cycle compreso tra 0 e 1 nel segmento 4, è possibile distinguere due diversi casi:

- FEEDBACK è uguale a 0. Se così fosse, il SETPOINT è anche il denominatore del blocco "DIV" del segmento 4;

- FEEDBACK è diverso da 0. Se così fosse, invece, la differenza tra il SETPOINT e il FEEDBACK è il denominatore del blocco "DIV" del segmento 4.

Le variabili "M1.0" e "M2.0" servono solamente a non far variare il denominatore appena calcolato ad ogni ciclo.



### Segmento 4: Valutazione del Duty Cycle nel caso in cui la pressione aumenti nel serbatoio

Se nel serbatoio viene richiesta una pressione maggiore di quella presente, si attiva questo segmento in quanto il SETPOINT è maggiore del FEEDBACK. Tale segmento effettua una sottrazione tra SETPOINT e FEEDBACK, definendo l'errore, e il risultato lo divide in modo da ottenere un valore compreso tra 0 e 1. Il valore ricavato è il Duty Cycle POSITIVO.



### Segmento 5: Valutazione del Duty Cycle nel caso in cui la pressione diminuisca nel serbatoio

Se nel serbatoio viene richiesta una pressione minore di quella presente, si attiva questo segmento in quanto il SETPOINT è minore del FEEDBACK. Tale segmento effettua una sottrazione tra FEEDBACK e SETPOINT, definendo un errorepositivo, e il risultato lo divide in modo da ottenere un valore compreso tra 0 e 1. Il valore ricavato è il Duty Cycle NEGATIVO. Al rung primario sono stati aggiunti in parallelo altri tre rung. Questo perchè se al denominatore ci fosse lo "0", l'intero sistema va in tilt.



### Segmento 6: Modulazione dell'onda quadra nel caso in cui la pressione aumenti nel serbatoio

In ingresso al blocco "SCALE\_X" vi è la variabile "Duty Cycle POSITIVO" (numero reale tra 0 e 1), la quale viene scalata, appunto, tra 0 e 100 e il suo risultato, tramite il blocco "MOVE", viene spostato nella variabile "QW1002". Questa operazione consentirà la modulazione dell'onda quadra.



### Segmento 7: Generazione dell'onda quadra nel caso in cui la pressione aumenti nel serbatoio

Il blocco "CTRL\_PWM" permette di generare un'onda quadra di periodo T costante sul canale "Local~Pulse\_2" o "266" (tipologia dati "HW\_PWM"). Tale blocco, in questo caso, viene attivato dall'ingresso "I0.0".



### Segmento 8: Modulazione dell'onda quadra nel caso in cui la pressione diminuisca nel serbatoio

In ingresso al blocco "SCALE\_X" vi è la variabile "Duty Cycle NEGATIVO" (numero reale tra 0 e 1), la quale viene scalata, appunto, tra 0 e 100 e il suo risultato, tramite il blocco "MOVE", viene spostato nella variabile "QW1004". Questa operazione consentirà la modulazione dell'onda quadra.



Segmento 9: Generazione dell'onda quadra nel caso in cui la pressione diminuisca nel serbatoio

Il blocco "CTRL\_PWM" permette di generare un'onda quadra di periodo T costante sul canale "Local~Pulse\_3" o "267" (tipologia dati "HW\_PWM"). Tale blocco, in questo caso, viene attivato dall'ingresso "I0.0".



# Segmento 10: Attivazione della valvola di alimentazione

Se il SETPOINT è maggiore del FEEDBACK, allora si setta l'uscita associata alla valvola di alimentazione e si resetta la valvola di scarico. Mediante il blocco "MOVE" si sposta il valore "0" nella variabile "QW1004" per essere sicuri che la valvola di scarico sia sempre chiusa.



### Segmento 11: Attivazione della valvola di scarico

Se il SETPOINT è minore del FEEDBACK, allora si setta l'uscita associata alla valvola di scarico e si resetta la valvola di alimentazione. Mediante il blocco "MOVE" si sposta il valore "0" nella variabile "QW1002" per essere sicuri che la valvola di alimentazione sia sempre chiusa.



# Segmento 12: Disattivazione delle valvole

Se l'errore della valvola di scarico è compreso tra i valori "0" e "0.05", allora si resetta la valvola di scarico.



# 6.8 Controllo pressione rapido con blocco "CTRL PWM"

Il codice appena mostrato presenta però un piccolo problema, ovvero risulta essere molto lento nel raggiungimento della pressione desiderata. Per ovviare a questo problema, si è introdotta nel codice una piccola modifica, la quale restringe il range di modulazione del segnale mandato alla valvola.

Ad esempio, si supponga di avere nel serbatoio una pressione relativa pari a 0 bar e che il SETPOINT fornito al sistema sia di 3 bar. Con il precedente programma, il sistema avrebbe risposto alla richiesta dell'utente con una modulazione del segnale realizzata sin da subito, impiegando un sacco di tempo per raggiungere il valore del SETPOINT. Questo perché il "Duty Cycle" calcolato è spalmato sull'intero range dell'errore tra SETPOINT e FEEDBACK, come mostrato nella figura 6.15.



Figura 6.15: Variazione del "Duty Cycle" nel codice relativo al controllo pressione classico

Con il nuovo programma, invece, la modulazione del segnale mandato alla valvola avviene solo nell'intorno del valore del SETPOINT, come mostrato nella figura 6.16. Nel caso in esame il range all'interno del quale viene modulato il segnale è pari a  $\pm 0.5$  bar rispetto il valore del SETPOINT.



Figura 6.16: Variazione del "Duty Cycle" nel codice relativo al controllo pressione rapido

Le modifiche effettuate per ottenere un controllo pressione rapido sono riportate di seguito, rispettivamente per il caso in cui la pressione aumenti o diminuisca nel serbatoio. Tali segmenti andranno a sostituire i segmenti 2, 3, 4 e 5 del precedente codice.

### Segmento 2: Valutazione del Duty Cycle nel caso in cui la pressione aumenti nel serbatoio

Se nel serbatoio viene richiesta una pressione maggiore di quella presente, si attiva questo segmento in quanto il SETPOINT è maggiore del FEEDBACK. Tale segmento effettua una sottrazione tra SETPOINT e FEEDBACK, definendo l'errore, e il risultato se è molto grande associa al "Duty Cycle POSITIVO" un valore pari a 1, mentre quando l'errore entra nel range stabilito dal blocco "IN\_RANGE" lo divide per il "Valore Range" massimo indicato, in modo da ottenere un "Duty Cycle POSITIVO" variabile tra 0 e 1.



Segmento 2: Valutazione del Duty Cycle nel caso in cui la pressione aumenti nel serbatoio

### Segmento 3: Valutazione del Duty Cycle nel caso in cui la pressione diminuisca nel serbatoio

Se nel serbatoio viene richiesta una pressione minore di quella presente, si attiva questo segmento in quanto il SETPOINT è minore del FEEDBACK. Tale segmento effettua una sottrazione tra FEEDBACK e SETPOINT, definendo un valore del'errore positivo, e il risultato se è molto grande associa al "Duty Cycle NEGATIVO" un valore pari a 1, mentre quando l'errore entra nel range stabilito dal blocco "IN\_RANGE" lo divide per il "Valore Range" massimo indicato, in modo da ottenere un "Duty Cycle NEGATIVO" variabile tra 0 e 1.



#### Segmento 3: Valutazione del Duty Cycle nel caso in cui la pressione diminuisca nel serbatoio

Nelle figure 6.17, 6.18 e 6.19 sono state riportate tre prove sperimentali al variare del range di modulazione, avendo supposto per le valvole una frequenza di lavoro pari a 1 Hz, ovvero di periodo T=1000ms. In particolare, la prima prova è stata ricavata assegnando alla variabile "Valore Range" il valore "3" e quindi come se la modifica appena descritta nella pagina precedente non ci fosse. La seconda e la terza prova, invece, sono state ricavate assegnando alla variabile "Valore Range" rispettivamente i valori "1" e "0.5".



Figura 6.17: Setpoint, Feedback e Modulazione della valvola di alimentazione con "Valore Range" pari a 3

Osservando nel dettaglio la figura 6.17, è evidente che l'inizio della modulazione della valvola di alimentazione avviene quasi istantaneamente alla generazione dell'errore presente nel sistema: man mano che ci si avvicina al valore del setpoint, il tempo di apertura della valvola diminuisce fino a diventare nullo, garantendo la chiusura della valvola in maniera permanente. Nelle figure 6.18 e 6.19 la modulazione, invece, parte in prossimità del valore del Setpoint. Infatti, nei primi secondi, in entrambi i grafici, la valvola di alimentazione è lasciata sempre aperta (Duty Cycle=1) fino a quando l'errore tra setpoint e feedback scende rispettivamente sotto a 1 bar e a 0.5 bar. Differentemente dalla prima prova, dove la modulazione avviene gradualmente e il "Duty Cycle" diminuisce a passo costante, nella seconda e terza prova, sia ha un drastico cambiamento del "Duty Cycle", il quale passa dal 100% al 25% circa. Tuttavia, l'errore statico presente nel sistema è di circa 0.03 nel caso in cui la variabile "Valore Range" sia uguale a 3 e 0.01 negli altri due casi. L'errore statico presente nella prima prova è maggiore rispetto a quello ricavato dalle altre due perché, essendo il range di modulazione molto più grande, quando il Feedback è molto vicino al valore del setpoint, il valore del "Duty Cycle" è prossimo allo zero e quindi il tempo di apertura della valvola di alimentazione risulta essere talmente piccolo da non riuscire ad alimentare il serbatoio, mantenendo la pressione costante al suo interno. Infatti, quando il feedback raggiunge il valore 2.97 bar, l'errore è 0.03 dal setpoint, che ha valore 3 bar, e il "Duty Cycle" è 0.01. Quest'ultimo moltiplicato per il periodo T pari a 1000 ms, fornisce un  $t_{ON}$  pari a 10 ms, tempo troppo breve per la commutazione della valvola.



**Figura 6.18:** Setpoint, Feedback e Modulazione della valvola di alimentazione con "Valore Range" pari a 1



**Figura 6.19:** Setpoint, Feedback e Modulazione della valvola di alimentazione con "Valore Range" pari a 0.5

Nella figura 6.20, invece, vengono riportati sullo stesso grafico i tre diversi feedback associati ai tre "Valori Range" precedentemente citati. Ciò che si vuole mettere in risalto in questa figura è il tempo impiegato dal feedback per raggiungere il setpoint: minore è il "Valore Range" impostato, minore sarà il tempo di raggiungimento del feedback con il setpoint.



Figura 6.20: Confronto dei tre Feedback con "Valore Range" diversi, f=1 Hz

Le stesse prove sono state eseguite con una frequenza di lavoro a 10Hz della valvola di alimentazione. Di seguito, nella figura 6.21, sono riportate sullo stesso grafico i tre feedback associati a tre "Valori Range" differenti. Anche in questo caso, al diminuire del "Valore Range", il tempo impiegato dal feedback per raggiungere il setpoint diminuisce.



Figura 6.21: Confronto dei tre Feedback con "Valore Range" diversi, f=10 Hz

Per quanto riguarda l'errore statico, anch'esso diminuisce al diminuire del "Valore Range",

ma a parità di quest'ultimo, lo scostamento dal setpoint è maggiore rispetto alla configurazione precedente, le cui valvole lavoravano a 1Hz. Questo è giustificato dal fatto che lo stesso errore viene moltiplicato per periodi differenti e minore è il periodo T, minore è il tempo di apertura della valvola. In particolare, se, come nel caso precedente, l'errore fosse pari a 0.03 bar, il "Duty Cycle" sarebbe pari a 0.01, che moltiplicato per il periodo T=100ms, fornirebbe un t<sub>ON</sub> pari a 1ms, tempo assolutamente troppo basso per l'apertura della valvola. Dal datasheet della valvola MATRIX, infatti, il tempo di risposta in apertura è minore di 5ms, mentre quello in chiusura è minore di 2ms. Ritornando al tempo impiegato dal feedback per raggiungere il setpoint e impostando come margine di errore  $\pm 0.05$  da esso, come è evidente dalla figura 6.22, si osserva che maggiore è la frequenza di lavoro, maggiore è il tempo impiegato per raggiungere la pressione richiesta.



Figura 6.22: Confronto tra frequenze diverse a parità del "Valore Range"

# 6.9 Controllo pressione con PID

Il controllo pressione effettuato fino a questo momento è stato corretto mediante l'errore non compensato, ovvero quello stabilito dalla differenza reale tra setpoint e feedback. In questo paragrafo, invece, si utilizzerà per il controllo pressione l'errore compensato da un controllore PID, nonché un controllore Proporzionale-Integrativo-Derivativo. Il TIA Portal dispone anch'esso di un proprio controllore PID, denominato PID\_Compact, e questo verrà inserito nel programma utente con l'obiettivo di migliorare il controllo pressione nel serbatoio.

Il PID\_Compact è un regolatore PIDT1 con anti-windup e ponderazione del componente P e D e il suo algoritmo di calcolo si basa sulla seguente formula:

$$\mathbf{y} = \mathbf{K}_{p} \cdot \left[ (\mathbf{b} \cdot \mathbf{w} - \mathbf{x}) + \frac{\mathbf{1}}{\mathbf{T}_{I} \cdot \mathbf{s}} \cdot (\mathbf{w} - \mathbf{x}) + \frac{\mathbf{T}_{D} \cdot \mathbf{s}}{\mathbf{a} \cdot \mathbf{T}_{D} \cdot \mathbf{s} + \mathbf{1}} \cdot (\mathbf{c} \cdot \mathbf{w} - \mathbf{x}) \right]$$
(6.4)

dove:

- y = Valore di uscita dell'algoritmo PID
- $K_p = Guadagno proporzionale$
- s = Operatore di Laplace
- b = Ponderazione del componente P
- $\mathbf{w} = \mathbf{Setpoint}$
- $\mathbf{x} = \text{Feedback}$
- $T_I =$ Tempo di Integrazione
- $T_D = Tempo derivativo$
- a = Coefficiente per il ritardo derivativo
- c = Ponderazione del componente D

Osservando l'algoritmo di calcolo, esso si rivela molto simile a quello di un classico controllore PID che utilizza le costanti di tempo  $T_I \in T_D$ , ma diversamente da esso vi sono anche dei coefficienti di ponderazione del componente proporzionale e derivativo e che nella formula sono indicate con  $b \in c$ . Mediante questi coefficienti è possibile limitare l'effetto dei due componenti o addirittura annullarlo se posti uguale a 0. Per cui, per rendere tale controllore simile a uno classico, le variabili associate alla *Ponderazione del componente* P e alla *Ponderazione del componente* D sono state imposte uguale a 1. Così facendo, l'algoritmo di calcolo semplificato sarà:

$$\mathbf{y} = \mathbf{K}_{\mathrm{p}} \cdot \left[ (\mathbf{w} - \mathbf{x}) + \frac{\mathbf{1}}{\mathbf{T}_{\mathrm{I}} \cdot \mathbf{s}} \cdot (\mathbf{w} - \mathbf{x}) + \frac{\mathbf{T}_{\mathrm{D}} \cdot \mathbf{s}}{\mathbf{a} \cdot \mathbf{T}_{\mathrm{D}} \cdot \mathbf{s} + \mathbf{1}} \cdot (\mathbf{w} - \mathbf{x}) \right]$$
(6.5)

Tuttavia, nella maggior parte dei casi, al posto della  $T_I \in T_D$  si utilizzano dei coefficienti denominati coefficiente integrativo,  $K_i$ , e coefficiente derivativo,  $K_d$ , e che si definiscono rispettivamente come:

$$\mathbf{K}_{i} = \frac{\mathbf{K}_{p}}{\mathbf{T}_{I}} \tag{6.6}$$

$$\mathbf{K}_{\mathrm{d}} = \frac{\mathbf{T}_{\mathrm{D}}}{\mathbf{K}_{\mathrm{p}}} \tag{6.7}$$

Dopo aver mostrato l'algoritmo di calcolo del PID presente nel TIA Portal, quest'ultimo bisogna richiamarlo nel programma utente e in particolare, bisogna richiamarlo in un OB di schedulazione orologio, il quale lo richiama a intervalli costanti nel ciclo. In esso è presente un solo rung, il quale richiama appunto il *PID\_Compact*, come riportato nella figura 6.23.

Il blocco del *PID\_Compact* richiede in ingresso due variabili, che sono il *Setpoint* e il *Feedback*, mentre in uscita l'*Outoput* del PID, il quale coincide esattamente con la variabile y, precedentemente descritta. Di queste variabili, escluso il setpoint che viene inserito manualmente ogni qualvolta si effettui il controllo pressione, le altre due vengono definite nel MAIN. Prima di descrivere il codice presente nel MAIN, bisogna configurare il *PID\_Compact* in base al controllo che si intende effettuare. Per poterlo fare, bisogna cliccare o sulla "Cassetta blu degli attrezzi"



Figura 6.23: Richiamo del PID\_Compact nel Cyclic interrupt

presente sul blocco del *PID\_Compact* in alto a destra oppure cliccare su *Oggetti Tecnologici* nel menu di sinistra del TIA Portal, cliccare sull'oggetto tecnologico "PID\_Compact" di interesse e selezionare la voce *Configurazione*.

A questo punto si aprirà una nuova finestra , come quella mostrata nella figura 6.24, nella quale si potranno settare tutti i parametri del PID. Innanzitutto, il TIA Portal richiede nelle impostazioni base la modalità di regolazione e quale grandezza si intende regolare. In questo caso, come indicato nella figura 6.24, dovendo effettuare un controllo pressione, la grandezza da inserire è la pressione, misurata in bar (in particolare, si lavorerà con le pressione relative). Sulla stessa finestra, bisogna selezionare anche il "Mode" che si intende eseguire, che nel caso in esame, è un *Automatic Mode*.

🔻 Impostazioni di base 🛛 😪	
Modo di regolazione 🛛 🚽	Modo di regolazione
Parametri di ingressolus 😪	
🕶 Impostazione del valore ist 😪	Pressione 💌 bar 💌
Limiti del valore istant 😪	Inverti senso di regolazione
Scala del valore istant 😪	
🕶 Impostazioni avanzate 🛛 🚽	Attiva Mode dopo il riavvio della CPU
Controllo del valore ist <	Imposta Mode su: 🛛 Funzionamento automati 💌
Limitazioni PWM 🧹	
Limiti valore di uscita 🛛 🚽	
Parametri PID	

Figura 6.24: Impostazioni base del PID\_Compact nella Vista Funzionale

Sempre dalla stessa sezione, bisogna selezionare i parametri di ingresso e uscita, che, come è mostrato nella figura 6.25, sono rispettivamente "INPUT" e "OUTPUT". Oltre questi due parametri, è possibile inserire anche i parametri "INPUT\_PER" per gli ingressi e "OUT-PUT\_PER" e "OUTPUT\_PWM": l'unico degli ingressi e il primo delle uscite sono utilizzati quando si hanno segnali analogici, mentre il secondo delle uscite è usato per applicare la tecnica PWM direttamente con il PID.

😤 🛍 🔛		
🝷 Impostazioni di base 🛛 🧧		
Modo di regolazione 🛛 😪	Parametri di ingresso/uscita	
Parametri di ingressolus <		
💌 Impostazione del valore ist 😪	Setpoint:	
Limiti del valore istant 😪		
Scala del valore istant 😪		
🕶 Impostazioni avanzate 🛛 😔	Input:	Output:
Controllo del valore ist <	Input 🔹 🗖	Output 👻
Limitazioni PWM 🥪		
Limiti valore di uscita 🛛 😪		
Parametri PID 🧧		

Figura 6.25: Selezione dei parametri di ingresso e uscita del PID\_Compact

Nella seconda sezione, *Impostazioni del valore istantaneo*, è possibile inserire i limiti che il Feedback può raggiungere durante il controllo pressione. In questo caso è stato scelto, come è visibile nella figura 6.26, un limite superiore pari a 7 bar e un limite inferiore pari a 0 bar. Si tratta di tutte pressioni relative.

😤 🖬 🖽		
🕶 Impostazioni di base 🛛 🥪		
Modo di regolazione 🛛 🥪	Limiti del valore istantaneo	
Parametri di ingressolus 🥪		
👻 Impostazione del valore ist 🥪		bar
Limiti del valore istant 📀		
Scala del valore istant		T
🕶 Impostazioni avanzate 🛛 🥪	Limite superiore valore istantaneo: 7.0 bar	
Controllo del valore ist		
Limitazioni PWM 🥏		
Limiti valore di uscita 🛛 🥑		
Parametri PID 🥏		
	Limite inferiore valore istantaneo: 0.0 bar	
	•	

Figura 6.26: Selezione dei limiti del Feedback del PID Compact

Nella sezione *Impostazioni avanzate* è possibile indicare i limiti dell'output del PID, che in questo caso sono rispettivamente 10 e -10 per il limite superiore e inferiore (vedi figura 6.27) e i parametri PID, i quali possono essere cambiati anche manualmente se si spunta la casella *Attiva immissione manuale*, come mostrato nella figura 6.28. Nella stessa sezione è possibile scegliere anche la modalità di ottimizzazione del regolatore che si intende eseguire, scegliendo tra un'ottimizzazione PID e una PI.

😤 🖬 🙂	
🕶 Impostazioni di base 🛛 📀	
Modo di regolazione 🥏	Limiti valore di uscita
Parametri di ingresso/us 🥪	
🝷 Impostazione del valore ist 🤣	Limiti valore di uscita
Limiti del valore istant 📀	70 A
Scala del valore istant 🣀	T
🕶 Impostazioni avanzate 🛛 🧿	Limite superiore valore di uscita: 10.0 %
Controllo del valore ist 🧭	
Limitazioni PWM 🥏	
Limiti valore di uscita 📀	
Parametri PID 🥝	
	Limite inferiore valore di uscita: -10.0 %
	Comportamento in caso di errori
	Imposta Output su: Inattivo
	Valore di uscita sostitutivo: 0.0 %

Figura 6.27: Selezione dei limiti dell'Output del PID\_Compact

Impostazioni di base	0			
Modo di regolazione	0	Parametri PID		
Parametri di ingressolus.	🥑			
Impostazione del valore ist		🖌 Attiva immissione manuale		
Scala del valore istant	õ	Guadagno proporzionale:	1.0	
Impostazioni avanzate	0	Tempo di integrazione:	0.0	s
Controllo del valore ist	0	Tempo derivativo:	0.0	s
Limitazioni PWM	0	Coefficiente ritardo derivativo:	0.0	
Limiti valore di uscita		Ponderazione del componente Pi	1.0	
Parametri PID	~	Fonderazione dei componente F:	1.0	
		Ponderazione del componente D:	1.0	
	4	Tempo di campionamento dell'algoritmo PID:	1.0	s
	ŀ	Regola per l'ottimizazione		
		Struttura del regolatore:	PID	-

Figura 6.28: Selezione dei parametri dell'algoritmo di calcolo del PID\_Compact

La sezione *Parametri PID* delle *Impostazioni avanzate* verrà utilizzata successivamente per valutare l'influenza del guadagno proporzionale, integrativo e derivativo e di tutti gli altri parametri presenti all'interno dell'algoritmo di calcolo del PID.

Dopo aver richiamato e configurato nel Cyclic Interrupt il PID\_Compact, il prossimo step è scrivere il programma utente che sta alla base del controllo pressione. In realtà, questo codice risulta essere uguale a quello scritto nei paragrafi precedenti, con la principale differenza che è stato riadattato alle variabili del PID. Di seguito viene riportato l'intero codice presente nell'OB1 (MAIN) pensato per questa tipologia di controllo. Dal sotto paragrafo successivo verrà effettuata la *Messa in servizio*, la quale consente di attivare il PID\_Compact premendo prima di tutto lo "Start" del campionamento del segnale e poi lo "Start PID\_Compact" in basso a destra nella finestra di lavoro.

# Main [OB1]

Main Propriet	à							
Generale								
Nome	Main		Numero		1		Тіро	OB
Linguaggio	КОР		Numerazione		Automatico			
Informazioni								
Titolo	"Main Program Sv (Cycle)"	ram Sweep		3			Commento	
Famiglia			Versione		0.1		ID definito dall'utente	
Nome		Tipo di	i dati	Valore	di default	Comm	ento	
🛨 Input								
Initial_Call		Bool				Initial c	iitial call of this OB	
Remanence		Bool				=True,	=True, if remanent data are available	
Temp								
Constant								

### Segmento 1: Calcolo del Valore attuale (Feedback) presente nel serbatoio

In questo segmento viene prelevato mediante un sensore di pressione il valore di pressione all'interno del serbatoio che si intende controllare. Il segnale analogico, derivante dal sensore, viene normalizzato e scalato tra 0 e 10 per ottenere il valore di tensione corrispondente. Il valore in Volt viene successivamente convertito in pressione mediante l'utilizza del guadagno e l'offset del trasduttore. Il risultato di questa catena di calcoli è il valore attuale o feedback del valore di pressione nel serbatoio (anche INPUT del PID\_Compact).





## Segmento 2: Calcolo del Duty Cycle

Attraverso il segmento 2 è possibile calcolare il valore del Duty Cycle. In ingresso al blocco "DIV" vi è la variabile "Output" del PID, la quale viene divisa per 10 per scalarla tra 0 e 1. Nel caso la divisione dia un valore negativo, il blocco "ABS" provvederà a farne il valore assoluto.



# Segmento 3: Calcolo dell'errore non compensato

L'errore di pressione è definito come la differenza tra il "Setpoint" e il "Feedback".

	SUB Auto (Real)	
"Blocco_PID". Setpoint — "Blocco_PID". Feedback —	EN - ENO	"Blocco_dati_ sistema". "Errore di pressione"

# Segmento 4: Valutazione del T\_ON (tempo di apertura della valvola)

Il segmento in esame permette la valutazione del tempo di apertura della valvola di alimentazione e di scarico. Questo è possibile farlo perchè il valore del Duty Cycle viene moltiplicato per il periodo di lavore delle valvole e mediante i blocchi "T\_CONV" è possibile convertire innanzitutto il periodo T della valvola in un numero intero e poi il valore intero del t\_ON in un tempo.



## Segmento 4: Valutazione del T\_ON (tempo di apertura della valvola)

### Segmento 5: Calcolo del mezzo periodo

Dovendo lavorare con un onda simmetrica di periodo T, si è pensato di dividere il periodo in 2 parti uguali e impostare per entrambi i TON i mezzi periodi.



### Segmento 6: Generazione Onda quadra simmetrica con 2 TON

Avendo impostato ai due TON lo stesso PT, verrà generata un'onda quadra simmetrica. Viene attivato prima il TON\_1 e allo scadere del tempo impostato viene attivato il TON\_2. Quando anche quest'ultimo termina di contare, si apre il contatto "TON\_2.Q" e viene riattivato il ciclo.



### Segmento 7: Generazione treno di impulsi

Attraverso il secondo rung è possibile creare un treno di impulsi a frequenza costante visualizzabile sull'uscita "Q1.0". Ciò è reso possibile dall'attivazione della bobina "Q0.7" e dall'apertura del contatto NC "Q0.7". La bobina viene attivata quando l'uscita del TON\_1 si attiva.

ATTENZIONE ALLA DISPOSIZIONE DEI RUNG (il PLC legge dall'alto verso il basso e da sinistra verso destra).



# Segmento 8: Modulazione dell'onda quadra

Mediante il TOF in questione è possibile modulare l'onda quadra sfruttando gli impulsi generati nel segmento 7.



### Segmento 9: Attivazione della valvola di scarico

Se l'Output del PID risulta minore di zero, significa che la valvola che deve funzionare e modulare in base all'errore compensato esistente è quella di SCARICO. In questo modo, viene settata la variabile "M20.0" e resettata la "M25.0"



### Segmento 10: Attivazione della valvola di alimentazione

Se l'Output del PID risulta maggiore di zero, significa che la valvola che deve funzionare e modulare in base all'errore compensato esistente è quella di ALIMENTAZIONE. In questo modo, viene settata la variabile "M25.0" e resettata la "M20.0"



# Segmento 11: Modulazione della valvola di scarico



### Segmento 12: Modulazione della valvola di alimentazione

%M25.0 "Tempo "Attivazione Apertura".Q Alimentazione"	%Q0.1 "Alimentazione" 	

### 6.9.1 Confronto Feedback al variare della frequenza di lavoro

A questo punto, definito il programma utente capace di controllare la pressione nel serbatoio Pneumax, bisogna individuare a quale frequenza possono lavorare le valvole di alimentazione e scarico, per poi successivamente studiare come il sistema reagisce al variare dei parametri che caratterizzano il PID ( $K_p$ ,  $K_i \in K_d$ ).

Si parta dunque dallo studio delle frequenze di lavoro delle elettrovalvole digitali. In particolare, è stata effettuata la stessa prova a sei diverse frequenze di lavoro e con un  $K_p$  uguale a 1, ponendo tutto il resto uguale a 0 (quindi come se fosse valutato un errore non compensato), con l'obiettivo di innalzare la pressione del serbatoio fino a valore di 3 bar. Le frequenze studiate per osservare il comportamento del sistema sono le seguenti: 1 Hz, 2 Hz, 10 Hz, 20 Hz, 50 Hz e 100 Hz. Di seguito, verranno plottati i diversi grafici eseguiti a frequenze diverse per estrapolare da essi conclusioni utili alla scelta della frequenza di lavoro alla quale le valvole di alimentazione e scarico dovranno lavorare durante il controllo pressione.

Nella figura 6.29 sono riportati 3 grafici sincronizzati tra loro che mostrano, dall'alto verso il basso, come il feedback raggiunge il valore del setpoint, la modulazione della valvola di alimentazione e la modulazione della valvola di scarico.



Figura 6.29: Setpoint, feedback, modulazione della valvola di alimentazione e scarico ad una frequenza di lavoro pari a 1 Hz

Supponendo che ci sia un range di tolleranza del  $\pm 1\%$  dal valore del setpoint, è possibile affermare che il feedback raggiunge il setpoint in un tempo pari a 72s con un errore pari a 0.03 bar. Tuttavia, con il passare del tempo, il feedback si stabilizza sul valore 3 bar. In questa prova, essendo il segno dell'errore sempre positivo, non è richiesto il funzionamento della valvola di scarico, ma solo di quella di alimentazione, la quale lavorerà con un periodo T pari a 1000ms, ovvero ad una frequenza di 1Hz.

Nella figura 6.30, invece, sono stati riportati gli stessi grafici sincronizzati, con la sola differenza che le valvole di alimentazione e scarico lavorano ad una frequenza pari a 2Hz.



**Figura 6.30:** Setpoint, feedback, modulazione della valvola di alimentazione e scarico ad una frequenza di lavoro pari a 2 Hz

In questo caso, il tempo che impiega il feedback a raggiungere il setpoint con un errore minimo di 0.03 bar è 60s, nonché 12 secondi in meno rispetto al caso precedente. Anche qui, con il passare del tempo, il feedback raggiunge completamente il setpoint ed essendo l'errore sempre positivo la valvola di scarico non è fatta funzionare. Nella figura 6.31, la frequenza utilizzata è pari a 10 Hz, ovvero il periodo di lavoro imposto è di 100ms.



**Figura 6.31:** Setpoint, feedback, modulazione della valvola di alimentazione e scarico ad una frequenza di lavoro pari a 10 Hz

Osservando il grafico, il tempo di raggiungimento del setpoint è nettamente minore rispetto ai primi due casi e in particolare, considerando sempre un margine di errore dal setpoint pari all'1%, tale tempo è pari a 34s circa. Già a partire da questa frequenza, essendo il periodo pari a 100ms, la modulazione delle valvole di alimentazione e scarico non è più visibile come nei precedenti casi. Inoltre, da questa prova, entra in gioco anche la valvola di scarico e ciò è dovuto al fatto che il feedback inizia ad oscillare intorno al valore del setpoint. Nonostante ciò, come è possibile osservare nella figura 6.32, tale oscillazione rientra ancora nel margine di errore impostato inizialmente e dunque la frequenza di tale prova può essere definita ancora buona per il fine ultimo del progetto.



Figura 6.32: Ingrandimento del Setpoint e feedback alla frequenza di lavoro pari a 10 Hz

Nella figura 6.33 vengono sempre riportati i 3 grafici sincronizzati tra loro caratterizzati da una frequenza di lavoro pari a 20Hz.



**Figura 6.33:** Setpoint, feedback, modulazione della valvola di alimentazione e scarico ad una frequenza di lavoro pari a 20 Hz

Per questa prova, il tempo impiegato dal feedback per raggiungere il setpoint è pari a 26s, sempre se si considera il range di tolleranza pari a  $\pm 1\%$  dal valore del setpoint. Anche qui la modulazione delle due valvole non è visibile in quanto il periodo T con cui lavorano è pari a 50ms. La valvola di scarico funziona con una maggiore frequenza rispetto al caso precedente e ciò è dovuto al fatto che il feedback oscilla con una ampiezza maggiore intorno al setpoint (vedi figura 6.34). Tuttavia, anche questa frequenza sembra rispettare tutti i limiti impostati per garantire un errore minore dell'1%.



Figura 6.34: Ingrandimento del Setpoint e feedback alla frequenza di lavoro pari a 20 Hz

Le ultime due prove, invece, riportate nelle figure 6.35 e 6.36, rispettivamente fatte a una frequenza di 50Hz e 100Hz, presentano un tempo di raggiungimento del setpoint molto piccolo, ma presentano errori molto grossi nell'intorno del setpoint.



**Figura 6.35:** Setpoint, feedback, modulazione della valvola di alimentazione e scaric o ad una frequenza di lavoro pari a 50 Hz



**Figura 6.36:** Setpoint, feedback, modulazione della valvola di alimentazione e scarico ad una frequenza di lavoro pari a 100 Hz

Infatti, per quanto riguarda la quinta prova, ovvero quella eseguita a 50Hz, il tempo impiegato dal feedback per raggiungere il setpoint è pari a 17s, mentre per la sesta prova, ovvero quella eseguita a 100Hz, è circa pari a 10s. Tuttavia, come già anticipato precedentemente, gli errori presenti in queste prove fuoriescono dai limiti imposti come è evidente nella figura 6.37. L'errore calcolato per la quinta prova si aggira intorno al 2%, mentre quello della sesta intorno al 5%. Ciò significa che per un controllo solo proporzionale con  $K_p=1$ , nel caso in cui la frequenza di lavoro delle valvole sia maggiore di 20 Hz, il sistema inizia a diventare instabile.



Figura 6.37: Ingrandimento del setpoint e feedback alla frequenza di lavoro pari a 50 Hz (sinistra) e 100 Hz (destra)

In conclusione, osservando anche il grafico riportato nella figura 6.38, effettuando un controllo solo proporzionale con  $K_p=1$ , si può dedurre che all'aumentare della frequenza di lavoro, il feedback impiega meno tempo per raggiungere il valore del setpoint, ma con un errore sempre più grande.


Figura 6.38: Confronto tra feedback al variare della frequenza di lavoro

Dunque, per la scelta della frequenza, bisogna trovare il giusto compromesso tra tempo di raggiungimento ed errore stabilito tra setpoint e feedback, e bisogna tener conto soprattutto del fatto che le valvole MATRIX, come fornito dal loro datasheet, presentano un tempo di risposta all'apertura massimo di 5 ms e uno alla chiusura di 2 ms. Per cui, teoricamente, si potrebbe lavorare ad una frequenza pari a 50Hz, ma, per quanto visto nelle prove sperimentali, le elettrovalvole digitali anche a 50 Hz iniziano a rispondere male, non riusciendo in alcuni momenti a commutare a causa della dinamica della valvola. Per questo motivo, si è preferito lavorare ad una frequenza di 20Hz, in modo da ottenere buoni risultati anche con un semplice controllo pressione esclusivamente proporzionale  $K_p$  uguale a 1.

#### 6.9.2 Confronto Feedback al variare del K<sub>p</sub>

Dopo aver scelto la frequenza di lavoro alla quale le valvole dovranno lavorare, ovvero 20Hz (T=50ms), è possibile studiare il controllo pressione al variare dei parametri PID. In questo sotto paragrafo si analizzerà nel dettaglio l'influenza del guadagno proporzionale,  $K_p$ , ponendo zero i restanti parametri, ottenendo nella seguente maniera un controllo esclusivamente proporzionale. Tuttavia, prima di eseguire queste prove sperimentali, bisogna fare la messa in servizio del PID\_Compact. Si parta osservando la figura 6.39, nella quale vengono riportati tre grafici sincronizzati tra loro che indicano rispettivamente, leggendo dall'alto verso il basso, il feedback che insegue il setpoint, l'errore non compensato e infine l'errore compensato, definito nella figura "Output del PID".



Figura 6.39: Setpoint e feedback, errore non compensato e errore compensato dal controllore P con  $K_p=0.1$ 

Essendo in questo caso  $K_p$  inferiore a 1, il valore dell'errore compensato viene decimato rispetto a quello non compensato e per questo motivo, come è ben visibile dal grafico sopra riportato, il feedback impiega molto tempo per raggiungere il valore del setpoint: in maniera più precisa, il tempo stimato per raggiungere la pressione di 3 bar è 73 secondi, tenendo conto della solita tolleranza pari all'1%. Per cui nel momento in cui si richiede al sistema una pressione di 3 bar, mentre l'errore iniziale non compensato è pari a 3 bar, quello compensato iniziale è uguale a 0.3 bar. Questo determina un tempo di apertura della valvola molto piccolo e quindi un lento raggiungimento del valore del setpoint. Se il  $K_p$  è, invece, uguale a 1, si ottiene la stessa regolazione che si era fatta nei paragrafi precedenti senza PID, in quanto l'errore non compensato è uguale a quello compensato. Nella figura 6.40 vengono riportati gli stessi grafici precedenti, però con un  $K_p$  uguale a 1.



Figura 6.40: Setpoint e feedback, errore non compensato e errore compensato dal controllore P con  $K_p=1$ 

È evidente che il tempo impiegato dal feedback per raggiungere il setpoint è nettamente inferiore rispetto al caso precedente ed è pari a 26 secondi. Nella figura 6.41 il guadagno proporzionale è stato posto uguale a 2. Questo significa che l'errore compensato in uscita dal controllore PID, in questo caso, è il doppio di quello non compensato. Infatti, se si dà un gradino di 3 bar, l'errore iniziale tra setpoint e feedback è pari a 3, mentre quello l'errore iniziale compensato risulta pari a 6.



Figura 6.41: Setpoint e feedback, errore non compensato e errore compensato dal controllore P con  $K_p=2$ 

Per quanto riguarda il tempo impiegato dal sistema per raggiungere il valore del setpoint, esso è ovviamente più basso della prova con  $K_p$  uguale a 1 e vale circa 17 secondi. Per cui raddoppiando il guadagno proporzionale si è riusciti a risparmiare ben 10 secondi.



Figura 6.42: Setpoint e feedback, errore non compensato e errore compensato dal controllore P con  $K_p=5$ 

Ovviamente bisogna sempre tener sotto controllo il feedback e fare in modo che esso non

oscilli troppo intorno al valore del setpoint. Nella figura 6.42, in cui i grafici sono stati ricavati imponendo il  $K_p$  pari a 5, si osserva un nuovo fenomeno che nelle altre prove non si è mai verificato, meglio conosciuto come fenomeno di saturazione. Ciò succede quando l'errore compensato eccede il limite del valore dell'output del PID imposto e, fino a quando questo valore non scende sotto tale limite, l'uscita dal controllore assume sempre il valore massimo (10 in questo caso).



Figura 6.43: Zoom dell'errore non compensato e dell'errore compensato dal controllore P con  $K_p=5$ 

La valvola di alimentazione, che fornisce portata al serbatoio aumentando la pressione al suo interno, lavora con un Duty Cycle pari al 100% fino a quando l'errore compensato in uscita dal PID è maggiore o uguale a 10. Infatti, come mostrato nella figura 6.43, quando l'errore non compensato scende al di sotto dei 2 bar, anche quello compensato inizia a diminuire. Osservando, invece, la pendenza della curva relativa al feedback, se paragonata al caso precedente, risulta essere maggiore e ciò si traduce in un tempo minore per il raggiungimento del setpoint, che in questo caso vale 10 secondi. Aumentando il  $K_p$  fino a 10, si può iniziare ad osservare una piccola sovraelongazione del feedback rispetto al valore del setpoint, come riportato nella figura 6.44.



Figura 6.44: Setpoint e feedback, errore non compensato e errore compensato dal controllore P con ${\rm K_p}{=}10$ 

Per quanto concerne il tempo impiegato dal sistema per raggiungere il valore del setpoint, se si considera il range di tolleranza pari a  $\pm 1\%$  dello scalino imposto, anche in questo caso, è pari

a 10 secondi.



Figura 6.45: Setpoint e feedback, errore non compensato e errore compensato dal controllore PID con  $K_p=100$ 

N.B.: il tempo di raggiungimento non corrisponde a quello associato alla prima intersezione del feedback con il setpoint, ma alla sua stabilità intorno a quel valore con un certo range di tolleranza. Per cui, già da questo valore del  $K_p$ , iniziano a non esserci più vantaggi legati alla velocità del sistema per raggiungere la pressione richiesta dall'utente. Infatti, aumentando il guadagno proporzionale, si renderà il sistema sempre più instabile, come avviene nella prova sperimentale fatta con  $K_p$  uguale a 100 e riportata nella figura 6.45.

Anche in questa prova è ovviamente presente una sovraelongazione, che si spinge fino a 3.3 bar per poi dare inizio a un'oscillazione permanente intono al valore del setpoint. Nella prova in esame, è difficile determinare il tempo di raggiungimento del setpoint se non si considera un errore almeno del 5%: se così fosse il tempo stimato è di 30 secondi circa. Questo errore grossolano si verifica perché, in molti istanti di tempo, si ha la saturazione dell'errore compensato, soprattutto quando il feedback è maggiore del setpoint. Questo comportamento definisce il sistema instabile. Per cui, in definitiva, andando a comparare tutte le curve ricavate al variare del  $K_p$  (vedi figura 6.46), si è osservato che usando un controllore esclusivamente proporzionale, all'aumentare del K<sub>p</sub>, il sistema è più gagliardo, ma meno stabile. Infatti, per K<sub>p</sub> uguale 100 il feedback, nonostante raggiunga il valore del setpoint in meno di 10 secondi, presenta una sovraelongazione iniziale susseguita da oscillazioni permanenti nel tempo senza ma assestarsi al valore dei 3 bar. Contrariamente, per un valore basso del K<sub>p</sub>, il feedback raggiunge più lentamente il setpoint, rimanendo, tuttavia, stabile. All'aumentare del guadagno proporzionale, l'errore compensato tende a saturare più frequentemente permettendo un aumento più rapido di pressione nel serbatoio e facendo corrispondere ad errori non compensati minimi, errori compensati molto alti determinando l'instabilità del segnale di controllo.



Figura 6.46: Confronto tra feedback al variare del guadagno proporzionale  $K_p$ 

#### 6.9.3 Confronto Feedback al variare del K<sub>i</sub>

In questo sotto paragrafo si andrà ad analizzare l'azione integrale del controllore PID, la quale rappresenta un fattore molto importante in quanto garantisce l'assenza di un errore nullo a regime. Mediante il guadagno integrativo è possibile realizzare un controllore PI, ovvero proporzionale-integrativo, il quale garantisce una migliore precisione senza rendere instabile l'intero sistema se si scelgono i giusti valori delle costanti. A tutto questo si aggiunge anche una maggiore velocità del feedback nel raggiungere il valore del setpoint. Come sarà possibile osservare dalle prove sperimentali realizzate al diminuire del  $K_i$  e ponendo il  $K_p$  pari a 1, un minimo errore positivo porterà sempre ad un incremento del segnale di controllo, mentre uno negativo il suo decremento indipendentemente da quanto piccolo sia l'errore.

Per quanto riguarda il TIA Portal, la configurazione del blocco PID\_Compact porta l'utente a variare non il  $K_i$ , ma la costante di tempo integrale  $T_I$ , la quale più piccola è tanto più grande sarà il suo effetto. Si parta con l'osservare la figura 6.47, nella quale è sono stati riportati due grafici incolonnati e sincronizzati tra loro. Guardando dall'alto verso il basso, il primo grafico mette in luce l'inseguimento del setpoint da parte del feedback, mentre il secondo mostra l'errore non compensato istante per istante.



Figura 6.47: Setpoint e feedback, errore non compensato e errore compensato dal controllore PI con  $K_p=1$  e  $T_I=0.01s$ 

Ricordando che il  $T_I$  è inversamente proporzionale al  $K_i$  e che per queste prove il  $K_p$  è stato posto uguale a 1, la prima prova è stata eseguita con un  $K_i$  uguale a 100. Si può immediatamente osservare che per questo valore di  $K_i$ , il sistema è altamente instabile, presentando una sovraelongazione con conseguenti oscillazioni permanenti. Questo è dovuto principalmente al fatto che l'errore compensato in uscita dal controllore PI, anche in caso di piccoli errori positivi o negativi, è sempre saturo e assume il valore massimo. Infatti, abbassando il valore del  $K_i$  a 10 (oppure alzando il valore della costante di tempo integrale  $T_I$  a 0.1s), come è visibile dalla figura 6.48, l'errore compensato inizia ad oscillare più frequentemente intorno allo 0 dell'output del PID con minori saturazioni.



Figura 6.48: Setpoint e feedback, errore non compensato e errore compensato dal controllore PI con  $K_p=1$  e  $T_I=0.1s$ 

Anche in questa prova, è presente una sovraelongazione del feedback rispetto al valore del

setpoint, ma, in questo caso, le oscillazioni si attenuano fino a stabilizzarsi al valore del setpoint. Abbassando il valore del  $K_i$  a 1 ( $T_I=1s$ ), la risposta del sistema sembra migliorare. Infatti, osservando la figura 6.49, nonostante la sovraelongazione iniziale fino a 5 bar, il feedback sembra stabilizzarsi in un tempo pari a 90 secondi circa.



Figura 6.49: Setpoint e feedback, errore non compensato e errore compensato dal controllore PI con  $K_p=1$  e  $T_I=1s$ 

Nonostante ciò, è possibile migliorare ancora il sistema se il  $K_i$  lo si diminuisce ulteriormente fino a 0.1 ( $T_I=10s$ ), come è stato fatto nella figura 6.50. In questa maniera, per errori non compensati piccoli corrispondono errori compensati non saturati e quindi un migliore controllo del sistema sulla pressione. Il feedback sembra raggiungere quasi subito il valore del setpoint senza grosse oscillazioni e con errori prossimi allo zero.



Figura 6.50: Setpoint e feedback, errore non compensato e errore compensato dal controllore PI con  $K_p=1$  e  $T_I=10s$ 

Se, invece, si continua a diminuire il valore del K<sub>i</sub>, come mostrato nella figura 6.51, fino a 0.01

 $(T_I=100s)$ , il controllo si avvicina sempre più a quello esclusivamente proporzionale in quanto per valori molto alti del  $T_I$ , l'effetto dell'azione integrativa viene annullato. Infatti, se il  $T_I$  tende all'infinito, il termine integrale dell'algoritmo di calcolo del PID sparisce, facendo emergere solo l'effetto del proporzionale.



Figura 6.51: Setpoint e feedback, errore non compensato e errore compensato dal controllore PI con  $K_p=1$  e  $T_I=100s$ 



Figura 6.52: Confronto tra feedback al variare del coefficiente integrativo K<sub>i</sub>

Per cui, in definitiva, si è osservato che all'aumentare del  $K_i$  (vedi figura 6.52), nonché al diminuire del  $T_I$ , il sistema risulta essere più rapido raggiungendo il valore del setpoint più velocemente, ma presenta alcune ripercussioni sulla sua stabilità mostrando forti oscillazioni. Solitamente il termine integrativo viene utilizzato per azzerare l'errore a regime e scegliendo i giusti parametri per un controllore proporzionale-integrativo è possibile ottenere una regolazione precisa senza rendere instabile l'intero sistema. Il termine integrale introduce, a differenza del termine proporzionale, anche uno sfasamento di 90° in ritardo, il quale porta ad una anticipata instabilità del sistema ad anello chiuso.

#### 6.9.4 Confronto Feedback al variare del K<sub>d</sub>

In ultima analisi, resta da analizzare la parte derivativa presente nei controllori PID. In particolare, quello che si è realizzato con queste ultime prove sperimentali è un controllo pressione mediante un controllore PD, ovvero proporzionale-derivativo, fissando il  $K_p$  uguale a 1 e facendo variare il  $K_d$ . Un controllore derivativo è definito anche anticipatore o controllore di velocità, in quanto l'output derivante da esso non dipende dall'errore attuale nel sistema, ma dalla velocità con cui varia. Diversamente dall'azione integrativa, quella derivativa consente di avere un anticipo di fase di 90°, introducendo uno zero nell'origine, e questa caratteristica viene sfruttata solitamente da tutti quei controlli definiti ad anello aperto. Il parametro che assume un ruolo fondamentale in questi controlli è la costante di tempo derivativa  $T_D$ , il cui valore determina la rapidità di salita del feedback durante il controllo. Tuttavia, lo svantaggio principale che caratterizza il termine derivativo è legato alla stabilità, in quanto rende il segnale di controllo instabile. Per studiare il contributo che questo termine derivativo dà al sistema, si è fatto variare il valore della costante di tempo  $T_D$  e quindi anche del  $K_d$ , essendo le due grandezze direttamente proporzionali tra di loro. Nella figura 6.53, vengono riportati i soliti tre grafici incolonnati e sincronizzati tra di loro, che mostrano, leggendo dall'alto verso il basso, l'inseguimento del setpoint dal segnale di controllo (feedback), l'errore non compensato presente nel sistema e quello compensato dal controllore. Nel dettaglio, questa prova sperimentale è caratterizzata da un  $T_D$ pari a 0.01s (K<sub>d</sub>=0.01).



Figura 6.53: Setpoint e feedback, errore non compensato e errore compensato dal controllore PI con  $K_p=1$  e  $T_D=0.01s$ 

Osservando la risposta del sistema, è evidente come l'andamento del feedback sia molto simile a quello determinato da un controllore esclusivamente proporzionale. Ciò è spiegato dal fatto che per questa prova, essendo il valore del  $T_D$  estremamente piccolo, l'effetto dell'azione derivativa è poco rilevante rispetto a quella proporzionale. Tuttavia, è possibile osservare una leggera oscillazione del feedback intorno al setpoint, che nel caso solo proporzionale era inesistente: questo perché l'azione derivativa rende instabile il segnale di controllo. Nella figura 6.54 sono riportati i grafici relativi ad un controllore caratterizzato da una costante di tempo derivativa pari a 0.1s ( $K_d=0.1$ ). Anche in questo caso il segnale oscilla intorno al valore del setpoint quando viene raggiunto.



Figura 6.54: Setpoint e feedback, errore non compensato e errore compensato dal controllore PI con  $K_p=1$  e  $T_D=0.1s$ 



Figura 6.55: Setpoint e feedback, errore non compensato e errore compensato dal controllore PI con  $K_p=1$  e  $T_D=1s$ 

Tuttavia, oltre l'instabilità, bisogna osservare anche la rapidità con cui il feedback cerca di

raggiungere il setpoint: in particolare, all'aumentare della costante di tempo derivativa, aumenta anche il tempo di risposta del loop. Ciò è ancora più evidente nella prova successiva, in cui il valore della  $T_D$  è pari a 1s (vedi figura 6.55). Infatti, a parità di errore massimo, la pendenza del feedback è maggiore rispetto a quella dei casi precedenti. Nonostante questo, il tempo impiegato dal feedback per raggiungere il setpoint aumenta e questo è dovuto al fatto che man mano l'errore diminuisce, diminuisce anche la velocità con cui esso varia. Nella figura 6.56, la costante di tempo  $T_D$  è stata posta uguale a 10. determinando un  $K_d$  dello stesso valore.



Figura 6.56: Setpoint e feedback, errore non compensato e errore compensato dal controllore PI con  $K_p=1$  e  $T_D=10s$ 



Figura 6.57: Setpoint e feedback, errore non compensato e errore compensato dal controllore PI con  $K_p=1$  e  $T_D=20s$ 

Per questo valore della costante di tempo derivativa, il feedback raggiunge e supera il valore del setpoint, per poi stabilizzarsi con un certo offset da esso. Questo comportamento è ancora più eclatante nella prova sperimentale con  $T_D$  pari a 20, dove questo scostamento dal setpoint è maggiore. I risultati di tale prova sono stati riportati nella figura 6.57.

In queste due ultime prove è evidente come le curve relative ai feedback siano più instabili delle prime, presentando anche delle oscillazioni continue. Questo inconveniente è correlato all'azione derivativa del controllore, il quale, a frequenze elevate, amplifica i segnali. Per limitare l'amplificazione dei segnali, è possibile introdurre nel blocco PID un filtro del primo ordine, ponendo il parametro a, noto come *coefficiente per il ritardo derivativo*, pari a 1, che fino a questo momento è stato sempre nullo. Nella figura 6.58, sono riportati due grafici che mostrano come il feedback, a due diversi valori di  $T_D$ , cerca di raggiungere il setpoint nel caso in cui ci sia o meno questo filtro.

Sulla sinistra è riportato il caso in cui il valore della costante di tempo derivativa è pari a 0.1s, mentre sulla destra il caso in cui è pari a 10s. È possibile osservare che per valori di  $T_D$  molto bassi, la presenza o meno del filtro è quasi irrilevante, mentre per valori più alti, l'andamento del feedback risulta migliore se è presente il filtro, eliminando quell'errore statico a regime.



Figura 6.58: Confronto tra i feedback determinati da un controllore PD, con e senza filtro nella parte derivativa

In conclusione, andando a comparare i risultati ricavati al variare del coefficiente derivativo  $K_d$ , come riportato nella figura 6.59, la parte derivativa è utile per avere un miglior tempo di risposta del loop quando la costante di tempo  $T_D$  aumenta, ma, in compenso, la stabilità del sistema peggiora. Tuttavia, soprattutto ad alte frequenze, la presenza di un filtro del primo ordine nella parte derivativa del controllore potrebbe migliorare il controllo eseguito.



Figura 6.59: Confronto tra feedback al variare del coefficiente derivativo  $K_d$ 

#### 6.9.5 Metodo di Ziegler-Nichols

Dopo aver analizzato il comportamento del sistema al variare dei principali parametri PID  $(K_p, K_i, K_d)$ , bisogna trovare la giusta combinazione di valori che rendano il sistema ottimale. Esistono diverse tecniche di taratura dei parametri PID che consentono di individuare questa terna e per questo controllo pressione è stato scelto il *metodo di Ziegler-Nichols in anello chiuso*. Tuttavia, questo metodo non è possibile applicarlo su tutti i sistemi di controllo, ma solo a quelli che, se portati alla soglia dell'instabilità, non provocano danni. Si tratta di un metodo di taratura prettamente sperimentale e per poterlo applicare bisogna eseguire i seguenti step nell'ordine in cui vengono elencati:

- 1. annullare nel sistema di controllo ad anello chiuso tutti i parametri che caratterizzano un controllore PID ( $K_p$ ,  $K_i$  o  $T_I$ ,  $K_d$  o  $T_D$ );
- 2. aumentare progressivamente il valore del coefficiente proporzionale  $(K_p)$  fino a quando, introducendo nel sistema una piccola perturbazione a gradino dell'ingresso, si genera un'oscillazione permanente dell'uscita. Il valore del  $K_p$  ricavato, verrà denominato guadagno critico o  $K_{p0}$ , mentre il periodo di oscillazione generato  $T_0$  (se il sistema non entra mai in oscillazione, il metodo non è applicabile);
- 3. inserire nella tabella sotto riportata (estrapolata dal libro Automazione dei sistemi meccanici: corso di base[10]) i valori di  $K_{p0}$  e  $T_0$  e, in base alla tipologia di controllore che si intende utilizzare, estrapolare i valori dei parametri PID calcolati.

Tipo di controllo	$\mathbf{K}_{\mathbf{p}}$	$\mathbf{T}_{\mathbf{I}}$	$\mathbf{T}_{\mathbf{D}}$
Р	$0.50\cdot K_{\rm p0}$	-	-
PI	$0.45 \cdot K_{\rm p0}$	$\frac{T_0}{1.2}$	-
PID	$0.60 \cdot K_{\mathrm{p}0}$	$\frac{T_0}{2}$	$\frac{T_0}{8}$

Tale procedura è stata applicata a questo controllo pressione con lo scopo di ottimizzarlo. Di seguito verranno riportate le prove sperimentali fatte al variare del  $K_p$  per individuare il valore del guadagno critico  $K_{p0}$  e del suo periodo  $T_0$ , utili al calcolo dei parametri PID (vedi figura 6.60).



Figura 6.60: Ricerca del guadagno critico  $K_{p0}$  e del suo periodo  $T_0$ 

Da questi sei grafici è possibile notare che all'aumentare del  $K_p$ , il feedback inizia ad oscillare intorno al valore del setpoint (curva di colore nero), ma non presenta, almeno per questi valori del guadagno proporzionale, ancora un'oscillazione permanente con un periodo  $T_0$ . Continuando ad aumentare tale valore, quando si è giunti al valore 200 del  $K_p$ , il segnale ha iniziato a presentare le caratteristiche richieste dal metodo di taratura sopra descritto, come è evidente nella figura 6.61.



Figura 6.61: Identificazione del guadagno critico  $K_{p0}$  e del suo periodo  $T_0$ 

Per cui, andando ad associare il valore 200 al guadagno critico  $K_{p0}$  e il tempo 7 secondi al periodo critico  $T_0$ , è possibile identificare i valori dei parametri PID in base alla tipologia di controllore che si vuole adottare. Andando a sostituire nella tabella, si ricavano i seguenti valori:

Tipo di controllo	$\mathbf{K}_{\mathbf{p}}$	$\mathbf{T}_{\mathbf{I}}$	$\mathrm{T}_{\mathrm{D}}$
Р	100	-	-
PI	90	5.833	-
PID	120	3.5	0.875

Dopo aver calcolato i diversi parametri, è possibile inserirli nella sezione "Parametri PID" del blocco PID\_Compact per visualizzare quanto ottimale siano rispettivamente le soluzioni trovate per i tre diversi tipi di controllori da poter utilizzare.

Si parta dal controllo pressione mediante un controllore esclusivamente proporzionale, il cui valore del guadagno proporzionale è posto uguale a 100 e i restanti parametri posti nulli. Nella figura 6.62 è riportato il grafico che descrive il comportamento del sistema con tali parametri in risposta ad un valore di setpoint imposto dall'utente.



Figura 6.62: Risposta del sistema con i parametri P ottimizzati dal metodo di Ziegler-Nichols

Osservando il feedback di questa prima ottimizzazione, è evidente che il segnale si presenta abbastanza instabile: infatti, è presente una sovraelongazione iniziale seguita da oscillazioni perenni attorno al valore del setpoint che vanno via via attenuandosi. Nelle figure successive, 6.63 e 6.64, sono riportati rispettivamente i parametri PID e la relativa risposta del sistema ad un segnale a gradino se si considerasse un controllore proporzionale-integrativo.



Figura 6.63: Parametri PI ottimizzati dal metodo di Ziegler-Nichols



Figura 6.64: Risposta del sistema con i parametri PI ottimizzati dal metodo di Ziegler-Nichols

In questo caso, il feedback é più reattivo nel raggiungimento del setpoint, presenta una maggiore precisione e infine, anche una migliore stabilità rispetto alla regolazione mediante controllore esclusivamente proporzionale. Infine, nelle figure 6.65 e 6.66 sono riportati i parametri PID e il comportamento del sistema se questo è regolato da un controllore proporzionale-integrativo-derivativo. Anche per questi parametri, il feedback sembra seguire molto bene il valore del setpoint e il tempo di raggiungimento risulta essere molto breve. Tuttavia, è presente una leggera sovraelongazione del feedback, la quale è dettata principalmente dalla parte derivativa del controllore.



Figura 6.65: Parametri PID ottimizzati dal metodo di Ziegler-Nichols



Figura 6.66: Risposta del sistema con i parametri PID ottimizzati dal metodo di Ziegler-Nichols

Effettuando un paragone tra le tre risposte ricavate mediante il metodo di Ziegler-Nichols (vedi figura 6.67), è possibile affermare che tra le tre, sicuramente quella derivante dalla regolazione con controllore esclusivamente proporzionale non è idonea per il sistema studiato in quanto risulta principalmente instabile. Per quanto riguarda i feedback derivanti dai controllori PI e PID, sono caratterizzate da un tempo di raggiungimento del setpoint breve, definendo così un sistema pronto, mantenendo il sistema stabile. In particolare, è possibile notare che, a differenza della risposta derivante dal controllore PI, quella derivante dal controllore PID presenta una piccola sovraelongazione per la presenza del termine derivativo. Nonostante questo difetto, entrambe le curve possono ritenersi idonee al controllo pressione.



Figura 6.67: Confronto tra le risposte ottenute ottimizzando con il metodo di Ziegler-Nichols

In conclusione, mediante i parametri PID ricavati dal metodo di taratura di Ziegler-Nichols, si è effettuato un buon controllo pressione del serbatoio solamente mediante un controllore PI o PID. Tali valori potrebbero essere i parametri definitivi da configurare nel blocco PID, oppure, se lavorati e maneggiati accuratamente, rappresentare un buon punto di partenza per ottenere risultati migliori.

#### 6.9.6 Ottimizzazione attraverso il blocco PID

L'ambiente di lavoro TIA Portal offre la possibilità di ottimizzare i parametri PID mediante un algoritmo interno. Tale ottimizzazione è eseguibile durante la messa in servizio dell'oggetto tecnologico PID\_Compact dopo aver deciso il tempo di campionamento con il quale si intende registrare i dati. Esistono due tipi di ottimizzazione:

- Ottimizzazione iniziale, la quale rileva la risposta del processo a un gradino del valore di uscita e cerca il punto di flesso. Dalla pendenza massima e dal tempo morto del sistema regolato vengono calcolati i parametri PID;
- Ottimizzazione fine, la quale genera un'oscillazione costante e limitata del valore istantaneo. I parametri PID per il punto di lavoro vengono ottimizzati partendo dall'ampiezza e dalla frequenza di oscillazione e partendo dai risultati vengono ricalcolati tutti i parametri PID.

I parametri PID risultanti dall'ottimizzazione fine mostrano un comportamento pilota e di disturbo migliore rispetto ai parametri PID dell'ottimizzazione iniziale. Tuttavia, i parametri PID migliori si ottengono eseguendo di seguito l'ottimizzazione iniziale e l'ottimizzazione fine. Per eseguire le due ottimizzazioni bisogna prestare attenzione anche al campo "Stato" in basso a sinistra, il quale indica le eventuali operazioni in corso e se vi sono errori nel sistema: se esistenti potrebbe capitare che l'ottimizzazione non si avvii. La barra di avanzamento indica appunto lo stato di avanzamento dell'operazione in corso. Se durante l'ottimizzazione fine non si sono verificati errori, allora i parametri PID sono stati ottimizzati; viceversa, se si sono verificati errori, bisogna individuare dalla guida del TIA Portal l'errore corrispondente e risolverlo. I nuovi parametri ottimizzati si troveranno nella voce "Parametri PID" della sezione "Configurazione" dell'oggetto tecnologico PID\_Compact. Inoltre, il TIA Portal permette l'ottimizzazione sia di controllori PID e sia PI. Per questo tipo di controllo, è stata eseguita prima un'ottimizzazione per un controllore PI, i cui parametri ricavati sono stati riportati nella figura 6.68, e dopo un'ottimizzazione per un controllore PID, i cui parametri sono stati riportato nella figura 6.69.



Figura 6.68: Ottimizzazione dei parametri PID per un controllore di tipo PI

Attiva immissione manuale				
Guadagno proporzionale:	97.02143		0	±
Tempo di integrazione:	2.390443E-1	s	•	±
Tempo derivativo:	4.183275E-2	s	•	±
Coefficiente ritardo derivativo:	0.1		0	±
Ponderazione del componente P:	0.8		0	±
Ponderazione del componente D:	0.0		0	±
Tempo di campionamento dell'algoritmo PID:	0.003998	s	•	±
Regola per l'ottimizazione				
Struttura del regolatore:	PID	-	0	+

Figura 6.69: Ottimizzazione dei parametri PID per un controllore di tipo PI

Nella figura 6.70 sono riportati, invece, due grafici relativi alla prima ottimizzazione che

mostrano rispettivamente in uno setpoint e feedback e nell'altro l'errore non compensato presente nel sistema istante per istante



Figura 6.70: Risposta del sistema con i parametri PI ottimizzati dal TIA Portal

Di seguito, invece, sono riportati i medesimi grafici riportati per un controllore PI, ma in questo caso con i parametri PID ottimizzati.



Figura 6.71: Risposta del sistema con i parametri PID ottimizzati dal TIA Portal

Volendo esprimere un giudizio in merito ai risultati ricavati, è possibile dire che effettivamen-

te con entrambe le terne il sistema è stato ampiamente ottimizzato (si guardi la figura 6.72). Infatti, entrambi i feedback rispondono velocemente alla richiesta dell'utente, impiegando solo 4 secondi a raggiungere il valore del setpoint. Tra le due curve non vi sono nette differenze, ma è possibile osservare che il feedback determinato dal controllore PID, rispetto al semplice PI, presenta delle leggere oscillazioni attorno al setpoint e questo è dovuto alla parte derivativa presente nel controllore, il quale amplifica i segnali.



Figura 6.72: Confronto tra le risposte ottenute ottimizzando con il TIA Portal

Mentre, volendo fare un paragone con la miglior risposta ricavata dal metodo di taratura ad anello chiuso di Ziegler-Nichols, come è visibile dalla figura 6.73, è facile comprendere che la migliore curva è quella ricavata dall'ottimizzazione fatta dal TIA Portal.



**Figura 6.73:** Confronto tra le risposte ottenute ottimizzando con il metodo di Zielger-Nichols e con il TIA Portal

#### 6.9.7 Risposta del sistema ad un disturbo esterno

Ciò che si vuole mostrare in questo sotto paragrafo è la reazione del sistema all'introduzione di un disturbo esterno, che in questo caso è simulato da un regolatore di flusso, fornita della ditta *Metal Work*, posizionata a valle del serbatoio preceduta da una valvola ad azionamento meccanico a pulsante incassato della *SMC* (vedi figura 6.74).



Figura 6.74: Regolatore di flusso sulla sinistra e valvola ad azionamento meccanico sulla destra

Per cui, dopo aver ottimizzato con diversi metodi il controllo pressione, si è scelta l'ottimizzazione migliore tra quelle proposte, ovvero quella fatta dal TIA Portal con un controllore proporzionale-integrativo. Una volta impostati i valori nella sezione "Parametri PID" dell'oggetto tecnologico PID\_Compact, si è eseguita la "Messa in servizio" di tale blocco e si è imposto il valore del setpoint pari a 3 bar. Dopo alcuni secondi dal raggiungimento del feedback al valore impostato, si è premuto il pulsante della valvola meccanica per un periodo circa pari a 10 secondi in modo da garantire la fuoriuscita del flusso d'aria dalla resistenza posta a valle dell'intero sistema. Nella figura 6.75 sono riportati due grafici i quali mostrano, il primo, il setpoint e il feedback del sistema ottimizzato e, il secondo, l'errore non compensato presente istante per istante.



Figura 6.75: Risposta del sistema alla presenza di un disturbo esterno

Si osserva che, nel momento in cui viene premuto il pulsante, il feedback ha un cedimento di circa 0.02 bar e che nel giro di 1 secondo riesce subito a compensare riportandosi a 3 bar, per poi rimanere inalterato. Lo stesso accade quando il pulsante viene rilasciato, dove vi è una contropressione che rende momentaneamente il sistema instabile (2 secondi) per poi raggiungere nuovamente il valore del setpoint. Per cui, il sistema così ottimizzato è da considerarsi pronto, stabile e reattivo alla presenza di disturbi esterni.

### 6.10 Programmazione del HMI del pannello 3

La programmazione dell'HMI del pannello 3 non richiede alcuna modifica del programma già scritto, in quanto le possibili variabili visualizzabili sul touch panel sono già delle variabili presenti nei DB. Per cui, per questo pannello è possibile passare direttamente alla creazione della pagina dei comandi dell'HMI.

Per questa esperienza, si è pensato di strutturare la pagina con alcuni Campi I/O, già introdotti nel capitolo 5, una barra grafica e una Vista delle curve. Nella figura 6.76 è statas riportata un'anteprima della pagina progettata per il pannello .



Figura 6.76: Anteprima sul TIA Portal della pagina dedicata al pannello 3 del banco didattico

Come è possibile osservare dall'anteprima della pagina dei comandi dedicata al pannello 3, sono stati importati tre campi I/O utilizzati rispettivamente per visualizzare ed eventualmente inserire il periodo T dell'onda quadra, il valore del setpoint e il valore del feedback. Inoltre, accanto ad ogni campo I/O è presente una scritta che descrive quello che bisogna inserire negli appositi campi. Per associare la giusta variabile, si esegue sempre lo stesso percorso spiegato nel capitolo 5, ovvero cliccare sull'oggetto in questione, aprire le sue proprietà e inserire nella sezione generale la variabile da far corrispondere al campo I/O.

Per inserire l'oggetto *Barra grafica*, la quale permette la visualizzazione dei valori di processo su una barra grafica appunto, bisogna andare sulla sezione *Casella degli strumenti* e selezionare la voce *Elementi*. Attraverso "drag & drop" è possibile importare tale oggetto sulla pagina dei comandi e dalle sue proprietà è possibile caratterizzarlo: in particolare, mediante questa barra, si è pensato di mostrare visivamente il valore del feedback. Per poter caratterizzare tale oggetto, dalla sezione generale, come mostrato nella figura 6.77, bisogna inserire, oltre che la variabile da associare, anche il valore massimo e minimo in cui essa può variare.

Barra grafica_1 [Barra gra	fica]	🧟 Propri	età 🚺 Informazioni	追 🗓 Diagnostica	
Proprietà Animazio	ni Eventi Testi				
📑 Elenco delle proprietà	Generale				
Generale					
Conformazione	Processo				
Tipo di bordo	Valore massimo 5				
Scale	della scala:				
Etichetta		Variabile di processo:	Blocco_PID_Feedback		
Rappresentazione	-	Variabile PLC:	Blocco PID Feedback		
Formato del testo	F	venebie rec.	biocco_ribirecobuck		
Limiti/campi	F E	Indirizzo:		Real	
Stili/Design	Valore minimo				
Varie	della scala:				

Figura 6.77: Inserimento e configurazione dell'elemento "barra grafica" dalla casella degli strumenti

Infine, sempre dalla sezione *Casella degli strumenti*, selezionando la voce *Control*, è possibile inserire l'oggetto *Vista delle curve*, il quale consente di rappresentare valori di variabili dell'attuale processo o di archivio in forma di curva. Questo elemento è stato inserito per poter osservare live l'andamento del setpoint e del feedback e per poterlo configurare, si consiglia di consultare la guida presente sul TIA Portal.

Per completare la pagina, è stata aggiunta anche un'immagina di un serbatoio per richiamare l'idea che tramite tale programma è possibile effettuare un controllo pressione in un serbatoio. Tale grafica, esattamente come i cilindri e gli steli riportati sulla pagina dei comandi del pannello 1, è stata selezionata dalla voce *Grafiche* della sezione *Casella degli strumenti*.

# Capitolo 7 Conclusioni e sviluppi futuri

Il presente lavoro di tesi nasce con l'idea di costruire un nuovo banco didattico all'interno del Politecnico di Torino che, messo a disposizione ai futuri studenti specializzandi nel settore dell'automazione, li aiuti a comprendere meglio la progettazione e la programmazione di sistemi meccanici automatici. In particolare, il banco è stato costruito e pensato per essere compatto e, allo stesso tempo, flessibile e, dunque, strutturato in due principali parti: una parte fissa, indivisibile dal banco e composta da dispositivi indispensabili per qualsiasi tipo di automatismo, e una parte mobile, invece, intercambiabile, in quanto è rappresentata da pennelli rimovibili, ognuno dei quali può essere adoperato a seconda di ciò che si vuole inculcare nelle future menti accademiche. In questo elaborato sono state realizzate tre diverse esperienze di laboratorio, ognuna delle quali mette in luce diverse tipologie di programmazione del PLC e del HMI.

La prima esperienza, realizzata mediante il pannello 1, permette agli studenti di imparare a programmare un semplice banco elettropneumatico e fa scoprire loro l'importanza del cavo multipolare, soluzione adoperata nel campo industriale, utile al trasporto di più segnali dal pacchetto di elettrovalvole alle morsettiere degli input e output associate al PLC.

La seconda esperienza realizzata con il pannello 2, invece, vuole evidenziare il funzionamento del Modulo I/O, il quale consente di raccogliere tutti i segnali presenti sul campo di lavoro ed inviarli, tramite cavo multipolare, agli ingressi e le uscite del PLC, e quello dei trasduttori pneumo-elettrici, i quali permettono la trasformazione di un segnale pneumatico in uno elettrico. Per entrambe queste due esperienze si è programmato ad hoc anche l'HMI, il quale dà la possibilità all'utente di controllare il sistema anche dal touch panel.

La terza e ultima esperienza è stata focalizzata sul controllo pressione di un serbatoio Pneumax, effettuato mediante il blocco PID\_Compact presente all'interno del TIA Portal. Tale controllo è stato basato sulla tecnica del PWM, mediante la quale si è riusciti a regolare la pressione all'interno del serbatoio, modulando due valvole 2/2 NC, rispettivamente una per l'alimentazione e l'altra per lo scarico. Tramite un trasduttore di pressione, posto a valle del serbatoio, è stato possibile leggere il valore di pressione al suo interno e definire il feedback del sistema. Dopo aver studiato l'influenza della frequenza di lavoro delle elettrovalvole e dei parametri PID sul sistema, si è proceduto con l'ottimizzazione dello stesso e, tra le diverse ottimizzazione eseguite, la migliore è stata quella effettuata dal TIA Portal. Di seguito, viene riportata la tabella riassuntiva con i valori ottimizzati dal TIA Portal, mentre nella figura 7.1 sono riportati i feedback generati, utilizzando quei parametri, in risposta ad un segnale a gradino.

Tipo di controllo	$\mathbf{K}_{\mathbf{p}}$	$\mathbf{T}_{\mathbf{I}}$	$\mathbf{T}_{\mathbf{D}}$
PI	82.64268	0.27908	-
PID	97.02143	0.23904	0.04183



Figura 7.1: Confronto tra le risposte ottenute ottimizzando con il TIA Portal

È possibile concludere, quindi, dicendo che la migliore regolazione effettuata sul sistema è quella gestita da un controllore proporzionale-integrativo-derivativo, i cui parametri sono stati ottimizzati utilizzando lo strumento messo a disposizione dal TIA Portal. Pensando ad uno sviluppo futuro, invece, sarebbe bello continuare a ricercare tali parametri, effettuando ulteriori prove sperimentali, per capire se effettivamente l'ottimizzazione realizzata dal TIA Portal sia la migliore.

Inoltre, si potrebbe utilizzare anche un PLC con prestazioni migliori, come l'S7-1500, con lo scopo di aumentare la velocità di controllo o, in caso, utilizzare dei moduli di ingresso più veloci.

Infine, sarebbe interessante effettuare anche un controllo pressione attraverso il regolatore proporzionale descritto nel capitolo 4, con lo scopo di realizzare una regolazione continua di pressione nel serbatoio, utilizzando come setpoint un segnale sinusoidale o un'onda quadra.

## Bibliografia

- [1] Guido Belforte (2005), Manuale di Pneumatica, II Edizione, Tecniche Nuove, Milano.
- [2] Bonivento Claudio, Gentili Luca, Paoli Andrea (2011), Sistemi di automazione industriale: Architetture e controllo, McGraw-Hill, Milano.
- [3] http://automazione-plus.it/che-cose-lautomazione 71032
- [4] Martin Armando, Canna Franco (13 maggio 2014), Rivista: Automazione e Strumentazione, Che cos'è l'Automazione.
- [5] https://amslaurea.unibo.it/17830/1/sofia\_gianluca\_tesi.pdf
- [6] http://www.diag.uniroma1.it/ deluca/automation/Automazione\_CIM\_Architetture.pdf
- [7] Ulisse Belladonna, Angelo Mombelli (2001), Pneumatica, Hoepli, Milano.
- [8] Raparelli Terenziano, Antonelli Michele, Beomonte Zobel Pierluigi, Bucci Giovanni (2007), Rivista: Trasmissioni di potenza oleodinamica pneumatica lubrificazione, La pneumatica e le reti di comunicazione industriale: un'esercitazione didattica.
- [9] https://equitaliani.com/2018/03/26/automazione-e-telecontrollo-nozioni-base-proveconcorso
- [10] Viktorov Vladimir, Colombo Federico (2013), Automazione dei sistemi meccanici: corso di base, CLUT, Torino.
- [11] Guido Belforte, Andrea Manuello Bertetto, Luigi Mazza (2000), *Pneumatica: corso completo*, Tecniche Nuove, Milano.
- [12] Mucherino Antonio, Antuono Walter (1993), Il PLC: teoria, funzionamento e applicazoni, Tecniche Nuove, Torino.
- [13] Chiacchio Pasquale (1996), PLC e automazione industriale, McGraw-Hill, Milano.
- [14] http://www-2.unipv.it/industriale/images/documenti/didattica/SCA2012-13/Dispensa\_Bus\_Isocroni.pdf
- [15] http://docenti.etec.polimi.it/IND32/Didattica/AzionamentixAutomazione/files/Profibus%20Corso.pdf
- [16] http://homes.di.unimi.it/labinfoXchimica/Lezione5 pg22-37.pdf
- [17] Standard IEC 61131-1, Programmable controllers Part 1: General information, IEC, Ginevra, 2003.
- [18] http://www.aumatec.it

- [19] http://www.peduto.it/PLC/Corso%20PLC%20con%20esercitazioni.pdf
- [20] https://new.siemens.com/it/it/prodotti.html
- [21] Standard IEC 61131-2, Programmable controllers Part 2: Equipment requirements and tests, IEC, Ginevra, 2003.
- [22] Standard IEC 61131-3, Programmable controllers Part 3: Programming languages, IEC, Ginevra, 2003.
- [23] http://www.schoolofnerd.it/sites/default/files/Linguaggio%20SFC.pdf
- [24] Guida su SIMATIC STEP 7: TIA PORTAL V16.
- [25] Manuale di riferimento SIMATIC, Schema a contatti (KOP) per S7-300/400, Siemens AG, 2010.
- [26] Manuale di riferimento SIMATIC, Schema logico (FUP) per S7-300/400, Siemens AG, 2010.
- [27] Manuale di riferimento SIMATIC, Lista istruzioni (AWL) per S7-300/400, Siemens AG, 2010.
- [28] Manuale di sistema SIMATIC V4.2, S7, Sistema di automazione S7-1200, Siemens AG, 2016.
- [29] Istruzioni operative SIMATIC HMI, Pannelli operatore Basic Panel 2nd Generation, Siemens AG, 2019.
- [30] CATALOGO GENERALE PNEUMAX, Componenti per l'automazione pneumatica, Pneumax S.p.a., 2018.
- [31] Roberto Mazzei (2007), Rivista: Pneumatica ad isole, Dal componente al sistema integrato: un nuovo concetto di automazione.
- [32] http://www.matrix.to.it/brochures/820%20Series.pdf