

UNIVERSITÀ DEGLI STUDI DI PADOVA

FACOLTÀ DI INGEGNERIA
CORSO DI LAUREA MAGISTRALE IN INGEGNERIA DELL'AUTOMAZIONE

TESI DI LAUREA MAGISTRALE

Progetto e realizzazione di nuovi sistemi di
controllo per rubinetti mecatronici basati su
piattaforma Arduino

Relatore: Alessandro Sona

Laureanda: Claudia Zanin
626618-IAM

ANNO ACCADEMICO: 2011-12

SOMMARIO

In questo elaborato sono presentati il progetto e la realizzazione di un sistema di controllo per la regolazione automatica della temperatura e della portata dell'acqua in rubinetti per uso sanitario. L'attività proposta fa parte di uno studio effettuato presso il Dipartimento di Tecnica e Gestione dei Sistemi Industriali di Vicenza, su richiesta di Consorzio Ruvaris (Ricerca e Sviluppo di Rubinetti e Valvole). Lo scopo di questo lavoro è individuare nuove soluzioni meccatroniche applicabili ai rubinetti per ottenere un prodotto innovativo, economico, compatto, efficiente e semplice da maneggiare da parte di un qualsiasi utente.

Il prototipo realizzato è basato su hardware stand-alone molto compatto garantito dalla piattaforma open-source Arduino e prevede l'utilizzo di sensori di portata e temperatura e di elettrovalvole proporzionali. L'apparato sperimentale così ottenuto è stato sottoposto a differenti test al fine di analizzarne le prestazioni e valutarne l'effettiva applicabilità. In una seconda fase di progetto le elettrovalvole proporzionali sono state sostituite da vitoni comandati da motori passo-passo. Il nuovo sistema così individuato è stato modellizzato e implementato in ambiente Matlab/Simulink e sottoposto a simulazioni atte a confermare la validità della nuova soluzione e il miglioramento di prestazioni ottenibile, in vista di una futura realizzazione fisica del modello.

Infine si è posta l'attenzione su un modulo XBee, per la comunicazione wireless a basso costo tra diverse schede Arduino, in prospettiva di un inserimento del sistema all'interno della domotica della casa, con conseguente possibilità di monitoraggio dei consumi.

RINGRAZIAMENTI

Ai miei genitori che mi hanno accompagnato e sostenuto con pazienza e costanza in tutti questi anni di studi.

A mia sorella, ai miei amici, ad Alessia e tutta la squadra Bissuola Futsal.

Ai miei colleghi automatici che hanno condiviso con me questa esperienza, in particolare gli amici Giuseppe, Andrea e Dario.

Infine un ringraziamento particolare al mio relatore Prof. Alessandro Sona, al Dott. Aldo Dalla Via e allo staff di tutti i tecnici e collaboratori che hanno lavorato a questo progetto rendendolo realizzabile.

INDICE

Introduzione	1
1 PRESENTAZIONE DEL PROGETTO	3
1.1 Idea principale	3
1.2 Progetto dell'apparato sperimentale	4
2 VALVOLE E DISPOSITIVI PER LA MISURA	7
2.1 Elettrovalvole proporzionali	7
2.2 Sensori di portata	8
2.3 Sensori di temperatura	9
2.4 Vitoni 1/2 pollice	11
3 ELETTRONICA DI BASE	15
3.1 Arduino	15
3.1.1 Cos'è	15
3.1.2 Schede Arduino	16
3.1.3 Arduino Mega2560	18
3.2 Scheda di gestione integrata	21
3.3 Dispositivi per l'interfacciamento	22
4 REALIZZAZIONE PRIMO PROTOTIPO	23
4.1 Realizzazione apparato sperimentale	24
4.1.1 Case interfaccia utente	24
4.1.2 Case idraulico	25
4.1.3 Sistema globale	25
4.2 Controllo valvole e dispositivi per la misura	27
4.2.1 Sensori di temperatura	27
4.2.2 Sensori di portata	28
4.2.3 Elettrovalvole proporzionali	30
4.3 Controllo generale	31
4.3.1 Gestione casi eccezionali	34
5 REALIZZAZIONE SECONDO PROTOTIPO	37
5.1 Modello blocco idraulico	37
5.1.1 Descrizione del modello	37
5.1.2 Implementazione del modello in ambiente Matlab/Simulink	40
5.1.3 Analisi bontà del modello	43
5.2 Modello dell'intero apparato	45
6 RISULTATI SPERIMENTALI	49
6.1 Test primo prototipo	49
6.1.1 Variazione setpoint di temperatura	51
6.1.2 Variazione setpoint di portata	53
6.1.3 Strozzature acqua calda o fredda	55
6.1.4 Calo di pressione	57
6.1.5 Richiesta di setpoint inammissibili di temperatura	57

6.2	Test secondo prototipo	58	
6.2.1	Variazione setpoint di temperatura	59	
6.2.2	Variazione setpoint di portata	61	
6.2.3	Calo di pressione	62	
6.2.4	Strozzature acqua calda o fredda	64	
6.3	Confronto prestazioni	66	
6.3.1	Variazione setpoint di temperatura	66	
6.3.2	Variazione setpoint di portata	68	
7	INTRODUZIONE ALLA COMUNICAZIONE WIRELESS		71
7.1	Protocollo 802.15.4	71	
7.1.1	Topologie di rete	72	
7.1.2	ZigBee e 802.15.4	73	
7.2	I moduli XBee	75	
7.2.1	Cosa sono e cosa fanno	75	
7.2.2	Vantaggi di utilizzo	76	
7.2.3	Caratteristiche elettriche e meccaniche	76	
7.2.4	Adattamento dei livelli logici	78	
7.2.5	Tipi di antenna	79	
7.2.6	Serie 1 e Serie 2: le differenze	80	
7.3	Apparato sperimentale	81	
7.3.1	XBee Serie1	81	
7.3.2	XBee Shield	83	
7.3.3	XBee USB Board	84	
7.4	Motivazione e presentazione prove sperimentali		85
7.4.1	Comunicazione wireless tra due Arduino		87
7.4.2	Programmazione wireless di Arduino		89
	Conclusioni	91	
	APPENDIX	93	
A	APPENDICE	95	
A.1	Datasheet Elettrovalvole Proporzionali bürkert		95
A.2	Datasheet Vitoni Nuova Galatron	98	
A.3	Schema elettrico scheda di gestione integrata		102
	ELENCO DELLE FIGURE	103	
	ELENCO DELLE TABELLE	106	
	BIBLIOGRAFIA	107	

INTRODUZIONE

La rubinetteria elettronica sta riscuotendo sempre più consensi nell'ambito della totalità dell'offerta commerciale. Numerosi vantaggi distinguono i rubinetti elettronici dai rubinetti tradizionali, tra i quali spiccano sicuramente igiene e risparmio di acqua e di energia. Originariamente progettato per essere collocato in bagni pubblici, è un prodotto che si sta rivelando indicato anche per bagni privati. La miscelazione elettronica dell'acqua in rubinetti per uso sanitario è, infatti, un'applicazione che può diventare di uso abbastanza comune data la semplicità con cui la tecnologia è ormai entrata a far parte di qualsiasi ambito della nostra casa. Avere a disposizione un sistema automatico di regolazione della temperatura e del flusso dell'acqua è una comodità, ma anche un modo per poter risparmiare. Una comodità, o meglio un comfort, che i più diffusi rubinetti a cartucce monocomando, non riescono a garantire: basti pensare alla difficoltà nel mantenere costante la temperatura dell'acqua in uscita, ad esempio quando si ha una variazione di portata di acqua calda o fredda in corrispondenza all'apertura di un altro rubinetto collegato allo stesso impianto. Ma anche un modo per poter risparmiare, dal momento che, con una regolazione servocontrollata, si è certi della temperatura in uscita, evitando ad esempio di lasciar scorrere l'acqua per alcuni minuti nell'attesa che sia calda quanto desiderato. Inoltre, automatizzando il processo e integrandolo in un sistema elettronico, è possibile usufruire di varie funzionalità, tra cui un monitoraggio costante dell'utilizzo dell'acqua, la possibilità di avere una notifica visiva o sonora al raggiungimento della condizione di regime oppure avere un'indicazione del tempo di utilizzo, un'interruzione al verificarsi di determinate condizioni, fino ad arrivare ad un'integrazione dell'apparato in un sistema domotico di monitoraggio, che raccolga centralmente i dati inviati dal rubinetto.

Il mercato mondiale dei rubinetti elettronici offre però, al momento, prodotti ancora molto costosi (dell'ordine delle centinaia di Euro) e con funzionalità limitate dal punto di vista del controllo della miscelazione. In commercio sono presenti miscelatori termostatici, dotati di valvole termosensibili che si regolano al variare della temperatura desiderata. Questi dispositivi, tuttavia, hanno una dinamica lenta e si possono usurare perdendo efficacia e precisione nella regolazione della temperatura dell'acqua.

Su richiesta del Consorzio Ruvaris, è stato effettuato presso il Dipartimento di Tecnica e Gestione dei Sistemi Industriali di Vicenza, uno studio per l'individuazione di nuove applicazioni mecatroniche da impiegare in rubinetti per uso sanitario, al fine di realizzare un pro-

dotta innovativo e più efficiente rispetto a quelli che si trovano attualmente in commercio.

L'attività eseguita è consistita nell'analizzare due possibili soluzioni e verificarne la loro efficacia e industrializzabilità. La prima soluzione è stata realizzata fisicamente e il suo prototipo prevede l'utilizzo di elettrovalvole proporzionali che, adeguatamente comandate da un microcontrollore Arduino, assicurano un passaggio di acqua calda e di acqua fredda tale da garantire valori di portata totale e temperatura in uscita dal rubinetto pari a quelli desiderati e impostati dall'utente. Nella seconda soluzione le elettrovalvole proporzionali sono state rimpiazzate da una coppia di vitoni comandati in modo diretto da motori passo-passo. Questo nuovo assetto, di cui si analizza per il momento un efficiente modello, ha l'obiettivo di ridurre ulteriormente costi, dimensioni e pesi della componente idraulica.

Dopo una prima introduzione alla realizzazione del sistema generale (Capitolo 1) e un'analisi dei vari componenti utilizzati, quali dispositivi di misura (Capitolo 2) e componenti elettronici (Capitolo 3), sono presentati i due prototipi (Capitoli 4 e 5). Il primo prototipo e il modello del secondo, implementato in ambiente Matlab/Simulink, sono stati sottoposti quindi a diversi test simulativi (sotto ben precise condizioni lavorative) e confrontati tra loro per valutarne le prestazioni e l'effettiva realizzabilità fisica (Capitolo 6).

Infine, nel Capitolo 7, si introduce l'utilizzo di una comunicazione wireless a basso costo tra schede Arduino, basata sul protocollo IEEE 802.15.4 e realizzata con moduli XBee Serie 1. Una comunicazione di questo tipo, applicata allo studio in esame, renderebbe il sistema ancora più maneggevole e permetterebbe l'eventuale datalog per il monitoraggio dei consumi.

In Appendice sono riportati alcuni datasheet dei componenti utilizzati nell'apparato sperimentale, al fine di dare al lettore una visione tecnica completa del progetto dei sistemi di controllo realizzati, e qualche schema elettrico rilevante.

PRESENTAZIONE DEL PROGETTO

1.1 IDEA PRINCIPALE

I rubinetti ad oggi più diffusi sono caratterizzati dalla presenza di un miscelatore monocomando, in cui il flusso e la temperatura dell'acqua in uscita sono regolati da un'unica manopola. Esiste, infatti, una sola leva che, ruotata da destra a sinistra, regola la temperatura e, azionata in senso verticale, agisce sulla portata (Figura 1.1).

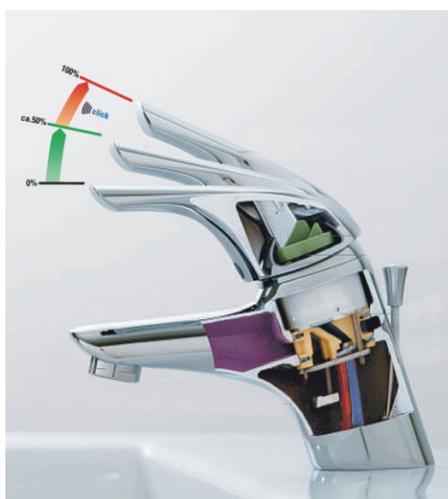


Figura 1.1: Esempio di miscelatore monocomando

In una prima fase di automatizzazione del sistema si è cercato di riprodurre il movimento effettuato da un individuo sulla leva di un classico miscelatore monocomando, controllandone in modo automatico la sola rotazione. Tale strategia, pur garantendo la temperatura dell'acqua in uscita dal rubinetto pari a un valore prefissato, si è rivelata, sotto diversi aspetti, poco efficiente, a causa dei seguenti motivi:

- impossibilità di controllare in modo automatico la portata in uscita, regolata quindi manualmente
- dinamica lenta, dovuta all'utilizzo di sensori termocoppie per il monitoraggio della temperatura dell'acqua in uscita
- dimensioni elevate, per il tipo di controllore stand-alone scelto, NI CompactRIO

Questo lavoro si pone l'obiettivo di individuare un nuovo approccio al sistema: tralasciare la dinamica del miscelatore monocomando e

agire direttamente sulle linee di acqua calda e acqua fredda, con il vantaggio di garantire un accurato controllo sia sulla portata che sulla temperatura dell'acqua miscelata. A questo scopo sono stati ideati due differenti prototipi, qui di seguito descritti.

Primo prototipo: sistema dotato di una coppia di *elettrovalvole proporzionali* posizionate lungo i tubi dell'acqua calda e fredda. L'invio di opportuni segnali garantisce il corretto grado di apertura e, di conseguenza, il desiderato flusso per ogni singola linea.

Secondo prototipo: le elettrovalvole proporzionali sono sostituite da *vitoni da 1/2 pollice*, comandati da motori passo-passo. Questa modifica apporta una riduzione degli ingombri, dei pesi e dei costi rispetto al primo apparato sperimentale.

Un aspetto di fondamentale importanza, in entrambe le soluzioni, è l'utilizzo del microcontrollore *Arduino*. Tale dispositivo costituisce l'elemento chiave del sistema di controllo: uno dei suoi compiti principali è la generazione degli opportuni segnali di comando da inviare alle elettrovalvole o ai motori che azionano i vitoni.

Per affinare ulteriormente il controllo e ottenere una dinamica molto veloce, si utilizzano sensori di portata molto efficienti. In questi progetti, infatti, il valore della temperatura dell'acqua in uscita viene ottenuto da una stima basata sulla misura delle portate e delle temperature lungo i canali di acqua calda e fredda. I sensori termocoppia vengono utilizzati in questi sistemi solo per rilevare tali temperature, generalmente sottoposte a variazioni di dinamica molto lenta.

La prima idea proposta presenta dinamiche molto buone anche a fronte di variazioni improvvise dei riferimenti di temperatura e portata. Tuttavia vi sono ancora dei margini di miglioramento, soprattutto per quanto riguarda costi e dimensioni, ottenibili con la realizzazione fisica del secondo prototipo ideato.

1.2 PROGETTO DELL'APPARATO SPERIMENTALE

Il progetto dell'apparato sperimentale di seguito descritto ha consentito di verificare la realizzabilità e l'efficacia delle soluzioni pensate, dando un'idea di massima sui componenti necessari allo sviluppo di questo prodotto, affinché esso fosse innovativo e industrializzabile. Lo scopo prefissato è la realizzazione di un prototipo di laboratorio funzionante, tralasciando per il momento le fasi di industrializzazione che rappresentano solamente un problema secondario.

In Figura 1.2 è riportata una schematizzazione dell'apparato sperimentale relativo al primo prototipo, ricordando che il secondo è molto simile e si differenzia principalmente per la sostituzione delle elet-

trovalvole con vitoni comandati da motori stepper.

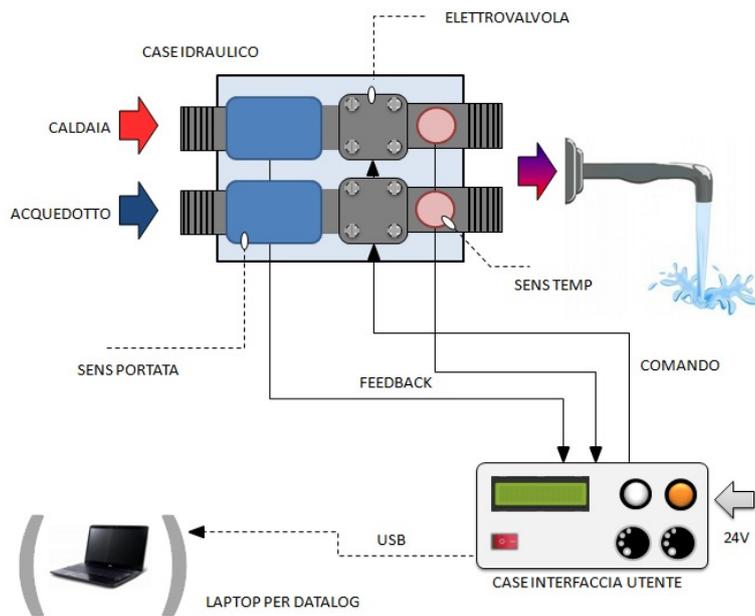


Figura 1.2: Rappresentazione schematizzata dell'apparato sperimentale

Il sistema risulta essenzialmente costituito da due sezioni:

- *case idraulico*, costituito da sensori di temperatura, sensori di portata ed elettrovalvole proporzionali
- *case interfaccia*, costituito da un microcontrollore Arduino, una scheda integrata e altri accessori per l'interfacciamento quali display, potenziometri e pulsanti monostabili

Queste due componenti interagiscono tra loro per assicurare il raggiungimento dei valori di portata e temperatura dell'acqua miscelata richiesti dall'utente che sta utilizzando il prodotto. L'acqua calda e fredda, proveniente rispettivamente dalla caldaia e dall'acquedotto, passa attraverso i due canali, lungo i quali sono disposti, in cascata, i sensori di portata, le elettrovalvole proporzionali e i sensori di temperatura. I dati restituiti dai sensori di portata, per le due linee, sono ricevuti dal case interfaccia, all'interno del quale un microcontrollore Arduino li analizza e comanda, in base ai riferimenti di portata e temperatura richiesti, la corretta apertura delle valvole corrispondenti. I sensori di temperatura, invece, monitorano le dinamiche (lente) relative ai due tubi. Infine, il case interfaccia, grazie a pulsanti, potenziometri e display LCD, permette all'utente di interagire con il sistema impostando le specifiche desiderate.

Nei prossimi capitoli si analizzano nel dettaglio le componenti che costituiscono i due case in entrambi i progetti studiati, dapprima rivolgendo l'attenzione alle valvole e ai dispositivi di misura utilizzati e, successivamente, all'elettronica presente.

VALVOLE E DISPOSITIVI PER LA MISURA

Si analizza di seguito il funzionamento delle componenti fisiche utilizzate o utilizzabili per la realizzazione dei prototipi visti. Solo in un secondo momento si porrà l'attenzione sulla loro effettiva introduzione all'interno dell'apparato sperimentale, esaminando il loro collegamento reciproco e il controllo su di esse applicato.

2.1 ELETTRIVALVOLE PROPORZIONALI

Nel primo prototipo sono state utilizzate elettrovalvole proporzionali a due vie della casa *bürkert* (Figura 2.1, [1]), alimentate con una corrente di 0.6 A.



Figura 2.1: Elettrovalvola proporzionale a due vie *bürkert*

Il disegno di Figura 2.2 rappresenta il principio di funzionamento della valvola. Quando è chiusa, il fluido in ingresso ha una pressione pari a p_1 , l'otturatore (3) è fuoriuscito e quindi preme sulla sede del pilota (4). Di conseguenza a questo e anche grazie alla forza della molla che agisce sul pistone (2), la sede principale (5) si chiude. Uno strozzatore (6) permette al fluido di entrare nella camera di controllo (1) e spinge la membrana dall'alto con una pressione p_x . Se lo strozzatore, la sede del pilota e il rapporto tra le aree sulla sede principale sono dimensionate correttamente, le forze di compressione sul pistone raggiungono un equilibrio quando la sede è aperta in una certa posizione. Con il controllo proporzionale del pilota, il pistone, idealmente, segue il movimento assiale continuo dell'otturatore, precisamente alla distanza che realizza questo equilibrio.

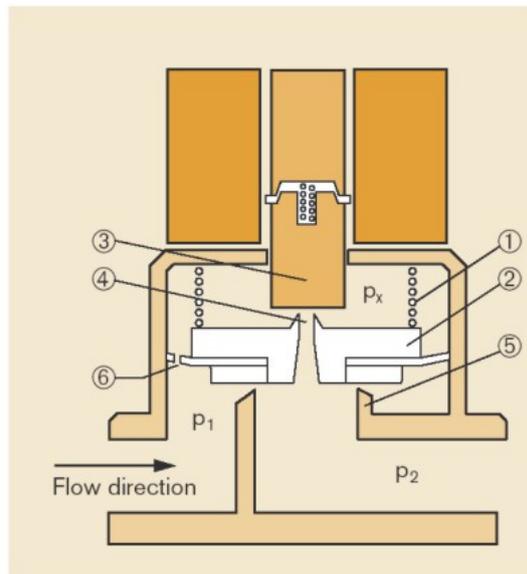


Figura 2.2: Diagramma di una valvola proporzionale servo-assistita

Questo controllo proporzionale viene effettuato dal microcontrollore di cui è dotato Arduino, che provvede a inviare a queste valvole i segnali opportuni per il loro corretto funzionamento. I comandi inviati, come si vedrà anche nel seguito, consistono in PWM a frequenza fissa e duty cycle variabile, il cui valore è calcolato e riaggiornato ad ogni istante, tramite il controllo.

2.2 SENSORI DI PORTATA

I dati inviati alle due valvole, che comandano la loro apertura, sono basati sulle misure provenienti dai *sensori di portata* che restituiscono, ad ogni ciclo di lavoro, la quantità di acqua che sta scorrendo in ognuna delle due linee. I sensori *OVAL - Eggs DELTA Pulse, LIQUI-VIEW* (Figura 2.3, [2]) sono dei misuratori di portata a vortice per liquidi.



Figura 2.3: Sensore di portata *Liqui-View*

Osservando Figura 2.4 è possibile comprenderne il loro funzionamento. L'ostruzione (*corpo di Bluff*) presente lungo il flusso del liquido

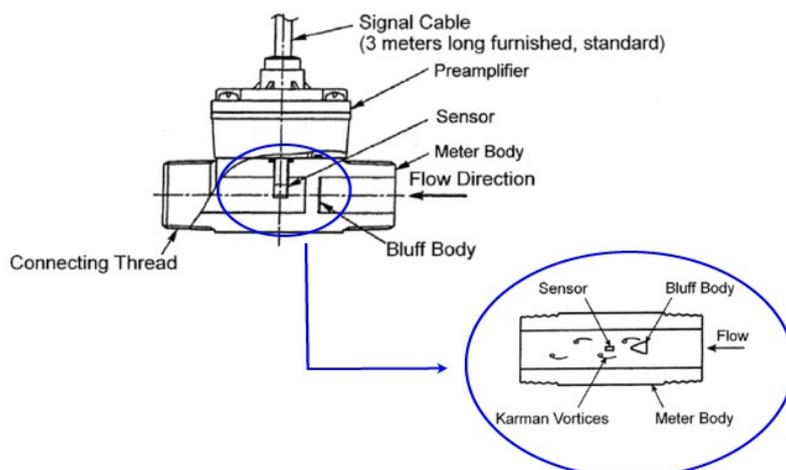


Figura 2.4: Analisi componenti del sensore di portata e loro funzioni

crea, a valle, dei vortici (*vortici di Von Karman*) di frequenza proporzionale alla velocità del fluido. Un sensore piezoelettrico individua questi vortici e genera degli impulsi elettrici proporzionali alla portata del liquido stesso. Questi sensori restituiscono, infatti, un'onda quadra a frequenza variabile e duty-cycle fisso al 50%, di ampiezza 5V. La lettura di questi dati, e quindi la portata delle due linee ad ogni istante, è resa possibile da un programma appositamente creato (analizzato più avanti) che prevede il campionamento dei fronti di salita delle onde quadre (o salita e discesa, nel caso si desideri una migliore risoluzione), poi opportunamente conteggiati ed elaborati. In Tabella 2.1 si riportano le principali specifiche di questi sensori:

Water Flow Range [L/min]	1.1 - 15
Operating Temperature Range	-20 to +60°C
Max Operating Pressure	0.98 MPa
Accuracy	± 3% of full scale
Repeatability	± 0.5%
Water Pressure Loss [kPa]	0.12 - 34.3
Nom. Output Pulse Unit [mL/P]	0.08900

Tabella 2.1: Tabella riassuntiva caratteristiche principali sensori portata

2.3 SENSORI DI TEMPERATURA

I *sensori di temperatura*, sonde MAXIM DS18S20 (Figura 2.5, [3]), sono dei termometri digitali che utilizzano il protocollo di comunicazione 1-Wire: su un unico filo viaggiano tutti i segnali.

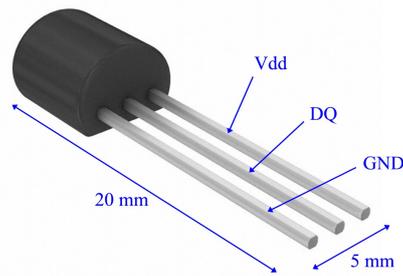


Figura 2.5: Sonda MAXIM DS18S20 con modalità di funzionamento attiva

Tale sonda ha due differenti modalità di funzionamento: *attiva* e *parassita*. Nel primo caso (sonda del tipo di Figura 2.5) il dispositivo è dotato di tre fili: *bus*, *ground* e *power*. Il bus è indipendente dal trasferimento dei dati e il microcontrollore può continuamente trasmettere lo stato del sensore. La modalità parassita, invece, permette alla sonda di essere alimentata dallo stesso filo su cui viaggiano i segnali, riducendo così solo a due (segnale + massa) i fili necessari per il collegamento. In questo progetto è stata utilizzata una sonda funzionante in modalità attiva.

Il funzionamento del protocollo *1-Wire* concettualmente è abbastanza semplice: la linea dati è collegata ad I/O open-drain, per cui c'è una resistenza di pull-up, comune a tutti i dispositivi, che tiene la linea dati normalmente a livello logico alto. In una linea di comunicazione *1-Wire* viene identificato un dispositivo Master (che gestisce le comunicazioni) e uno o più dispositivi Slave (che rispondono solo se interrogati). I dispositivi, non essendo dotati di clock, dovranno comunicare in tempi prestabiliti. La restrizione su tali tempi diviene più forte nel caso in cui si utilizzi la modalità di alimentazione parassita (questo uno dei motivi che hanno portato all'utilizzo della sonda in modalità attiva). Prima di cominciare una trasmissione, il dispositivo Master emette un segnale di reset: la linea dati viene mantenuta a livello logico basso per almeno $480\mu\text{s}$. I dispositivi presenti sul bus, quando devono effettuare la lettura, campionano la linea dati ogni $30\mu\text{s}$ a partire dai fronti di discesa del segnale. I dati sono inviati in gruppi di 8 bit. I dispositivi Slave sono identificati da un indirizzo univoco a 64 bit (8 bytes), per cui il Master, quando vorrà comunicare, dovrà specificare *con chi*. A tal proposito il Master dovrà conoscere a priori gli indirizzi di ogni dispositivo sulla linea. Nel protocollo *1-Wire* sono definiti alcuni comandi, chiamati ROM-commands, che servono appunto al riconoscimento/rilevazione degli indirizzi dei dispositivi presenti sul bus.

La sonda, come tutti gli altri dispositivi *1-Wire*, ha uno scratchpad, che è in pratica una memoria RAM con funzione di *cache*: i dati pro-

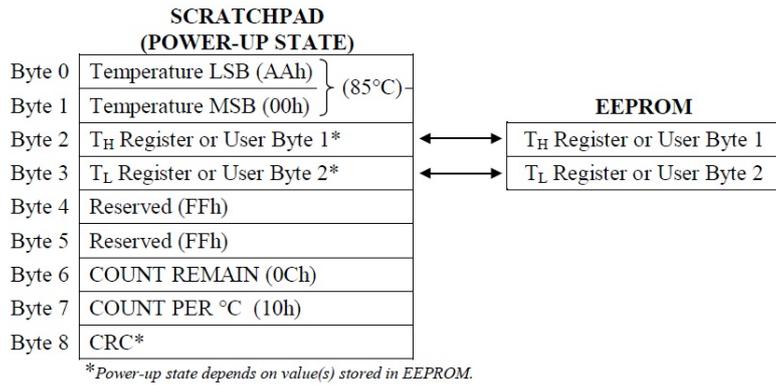


Figura 2.6: DS18S20 Memory Map

venienti dai sensori vengono trasferiti attraverso questa memoria. Lo scratchpad della sonda DS18S20 (Figura 2.6) è formato da 9 registri a 8 bit dedicati a funzioni particolari. Una corretta analisi ed elaborazione di questi registri permette di ricevere le informazioni di temperatura desiderate.

2.4 VITONI 1/2 POLLICE

La seconda soluzione prevede la sostituzione delle elettrovalvole proporzionali con vitoni da 1/2 pollice (Figura 2.7, [4]), della casa *Nuona Galatron - Solid Plast*, comandati da motori stepper.



Figura 2.7: Vitoni ceramici, GN90 e GN180, da 1/2 pollice

Il vitone ceramico è una sorta di valvola, la cui apertura/chiusura è garantita dalla rotazione di un perno zigrignato presente nella sua parte superiore (si veda Figura 2.8). Il range di apertura massima può essere di 90° (GN90) o altrimenti, come nel caso in esame, di 180° (GN180). Nella realizzazione del secondo prototipo si vorrebbe applicare, in presa diretta ai perni dei vitoni presenti nelle due linee, una coppia di motori passo-passo, in modo da permettere l'automatizzazione di apertura e chiusura del vitone. Il risultato dovrebbe essere



Figura 2.8: Schema funzionamento vitone

assolutamente analogo al precedente, ottenuto con le elettrovalvole proporzionali, con sostanziali differenze di costi e dimensioni, che ne risultano, così, estremamente ridotti. Come si può osservare in Figura 2.8, il flusso dell'acqua proveniente dalla caldaia, o dall'acquedotto, a seconda della linea considerata, entra dal basso e viene quindi gettata fuori lateralmente in quantità uguale o minore, in base all'angolo di apertura della valvola ($0^\circ - 180^\circ$).

La quantità di acqua che passa attraverso il singolo vitone è definita non solo dal grado di apertura della valvola, ma anche dalla pressione presente a monte di essa. Di seguito, in Figura 2.9 e in Tabella 2.2, sono riportati il diagramma della portata (L/min) al variare della pressione da 0 a 5 bar e i loro corrispondenti valori più significativi, in relazione al vitone scelto per il progetto in esame.

180°										
Bar	0.5	1.0	1.5	2.0	2.5	3.0	3.5	4.0	4.5	5.0
L/min	20	24	28.5	33.5	37	41.5	43	47	49.5	52

Tabella 2.2: Tabella della portata al variare della pressione da 0 a 5 Bar

Come già accennato in precedenza, il secondo prototipo non è stato realizzato fisicamente ma ne è stato individuato un opportuno modello, per l'analisi delle prestazioni, in vista della creazione di un nuovo apparato sperimentale. Nella creazione di suddetto modello si è cercato di ricreare l'andamento di Figura 2.9 in modo il più accurato

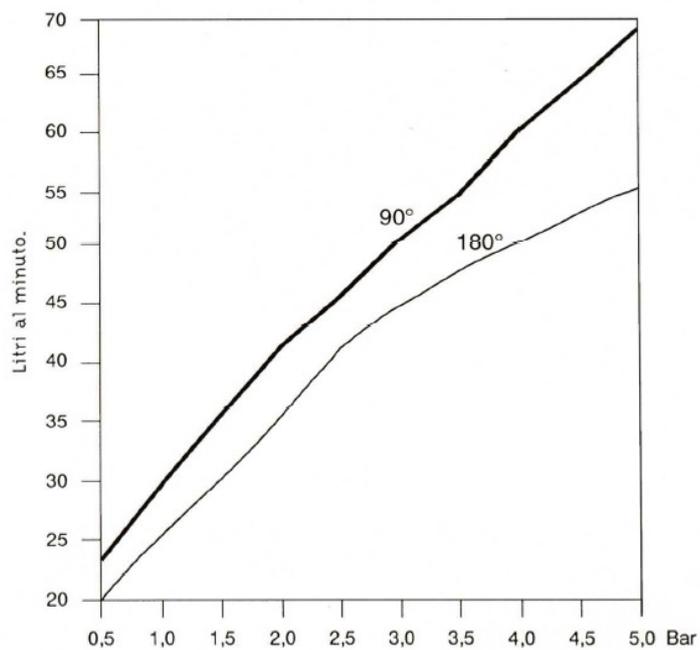


Figura 2.9: Diagramma della portata al variare della pressione da 0 a 5 bar

possibile, al fine di ottenere simulazioni attendibili e quasi pari alla realtà.

In questo capitolo si introducono le componenti elettroniche di base utilizzate per la realizzazione dell'apparato sperimentale. L'elemento chiave è sicuramente *Arduino*, sul quale è applicata una scheda integrata appositamente realizzata. Inoltre vi sono altri dispositivi elettronici per l'interfacciamento, che, insieme ai precedenti, costituiscono il case interfaccia. Di seguito si analizzano nel dettaglio le differenti componenti che costituiscono il case e se ne studia il loro funzionamento.

3.1 ARDUINO

3.1.1 Cos'è

Arduino ([5]) è una piattaforma open-source di prototipazione elettronica basata su hardware e software flessibili e semplici da utilizzare. E' composto da una piattaforma hardware per il physical computing sviluppata presso l'*Interaction Design Institute*, fondato da *Olivetti* e *Telecom Italia*.

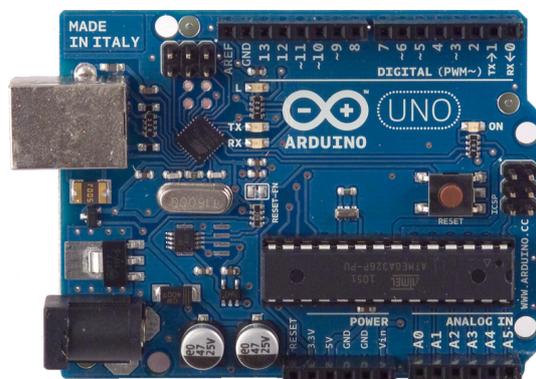


Figura 3.1: Esempio di scheda Arduino (Arduino UNO)

Questa si basa su un circuito stampato che integra un microcontrollore con PIN connessi alle porte I/O, un regolatore di tensione e un'interfaccia USB che permette la comunicazione con il computer. A questo hardware viene affiancato un ambiente di sviluppo integrato (*IDE*) multiplatforma (Linux, Apple Macintosh, Windows). Questo software permette di scrivere programmi con un linguaggio semplice e intuitivo derivato da C e da C++, chiamato *Wiring*, liberamente scaricabile e modificabile.

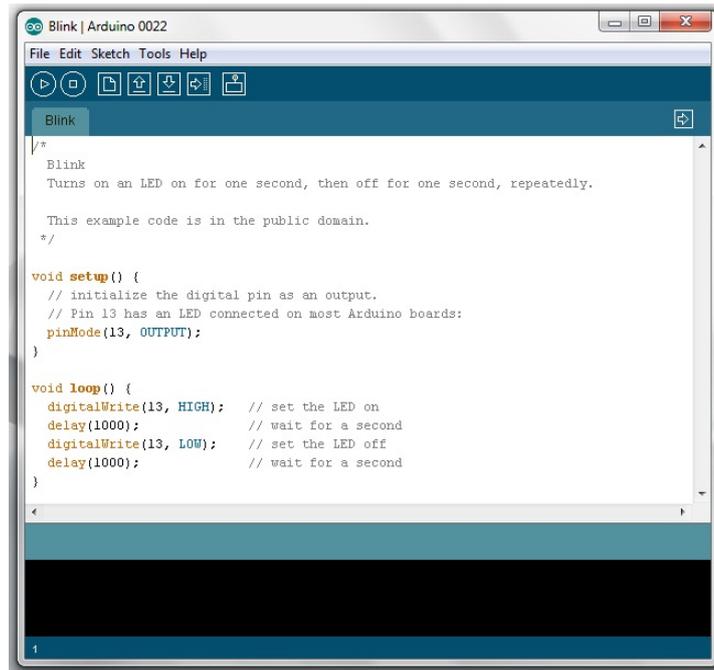


Figura 3.2: Esempio di screenshot dell'IDE di Arduino

Arduino può essere utilizzato per lo sviluppo di oggetti interattivi *stand-alone* ma può anche interagire, tramite collegamento, con software presenti su computer, come ad esempio Adobe Flash, Processing, Pure Data, ecc..

La piattaforma hardware Arduino è spesso distribuita in versione pre-assemblata, acquistabile in internet o in negozi specializzati. La particolarità del progetto è che le informazioni sull'hardware e soprattutto i progetti sono disponibili per chiunque: hardware open source. In questo modo, chi lo desidera, può legalmente auto-costruirsi un clone di Arduino o derivarne una versione modificata, scaricando gratuitamente lo schema elettrico e l'elenco dei componenti elettronici necessari. Alla base, tuttavia, vi è un unico codice sorgente per l'Ambiente di sviluppo integrato e un'unica libreria residente, che sono resi disponibili, e concessi in uso, secondo i termini legali di una licenza libera, GPLv2. Grazie alla base software comune, ideata dai creatori del progetto, per la comunità Arduino è stato possibile sviluppare programmi per connettere a questo hardware più o meno qualsiasi oggetto elettronico, computer, sensore, display o attuatore. Dopo anni di sperimentazione, è oggi possibile fruire di un database di informazioni vastissimo.

3.1.2 Schede Arduino

Una scheda Arduino tipica consiste in un microcontrollore a 8-bit AVR prodotto dalla *Atmel*, con l'aggiunta di componenti complemen-

tari per facilitarne l'incorporazione in altri circuiti. In queste schede sono usati chip della serie megaAVR, nello specifico i modelli ATmega8, ATmega168, ATmega328, ATmega1280 e ATmega2560. Fino ad oggi sono state commercializzate tredici versioni dell'hardware Arduino, tra cui Arduino Mega2560, utilizzata nel progetto analizzato, che fa uso di un ATmega2560 e che è un'evoluzione dell'Arduino Mega già esistente in precedenza.

Il microcontrollore presente nelle schede, in alcuni casi, è preprogrammato con un *bootloader* che semplifica il caricamento dei programmi sulla memoria flash incorporata nel chip. A livello concettuale, tutte le schede sono programmate attraverso una porta seriale RS-232, ma il modo in cui questa funzionalità è implementata nell'hardware varia da versione a versione. Le schede seriali Arduino contengono un semplice circuito inverter che permette la conversione tra il livello della RS-232 e il livello dei segnali TTL. Le versioni attuali di Arduino sono gestite via USB.

Per implementare il comportamento interattivo, Arduino è dotato di funzionalità Input/Output, grazie alle quali essa riceve i segnali raccolti da sensori esterni. In base a tali valori, il comportamento della scheda è gestito dal microcontrollore, secondo le decisioni determinate dal particolare programma in esecuzione in quel momento sulla scheda. L'interazione con l'esterno avviene attraverso attuatori pilotati dal programma attraverso i canali di output in dotazione (ad esempio i comandi inviati alle elettrovalvole proporzionali). A tale scopo, Arduino è dotata di molti connettori Input/Output per microcontrollore in uso su altri circuiti. Tutti i pin di I/O sono collocati sulla parte superiore delle schede, mediante connettori femmina da 0.1". Inoltre, sono disponibili commercialmente molte schede applicative plug-in, note come *shields*.

Nelle schede più recenti vi sono connettori per l'I/O digitale, la cui direzione di funzionamento, input o output, è decisa dallo *sketch* programmato sull'IDE. Vi sono poi dei canali I/O adibiti alla produzione di segnali Pulse Width Modulation (PWM), attraverso i quali è possibile, ad esempio, regolare l'intensità di luminosità di un LED o la velocità di rotazione di un motorino elettrico. Esiste un'apposita funzione, *analogWrite()*, che permette di controllare la PWM del segnale in uscita in maniera efficiente, senza dover eseguire linee di codice appositamente predisposte. La funzione citata accetta due parametri, il primo dei quali è il pin pilotato, mentre il secondo rappresenta l'intensità della modulazione (espressa su una scala da 0 a 255).

Infine vi sono connettori dedicati a ingressi di segnali analogici, cioè i valori di tensione letti da sensori esterni. Fino a un massimo di 5 Volt, questi valori sono convertiti in 1024 livelli discreti (da 0 a 1023). Questi connettori possono essere riprogrammati (sempre dal codice dello sketch sull'IDE) per funzionare come normali entrate/uscite digitali. L'alimentazione della scheda può avvenire attraverso la porta USB

del computer, o attraverso la maggior parte degli alimentatori USB oppure, infine, attraverso un adattatore in corrente continua a 9 Volt, con connettore cilindrico. In quest'ultimo caso, la scheda commuta automaticamente sull'alimentazione esterna quando il connettore dell'alimentatore esterno è inserito, mentre commuta autonomamente sull'alimentazione USB in caso di disconnessione del connettore.

3.1.3 Arduino Mega2560

La scheda utilizzata in questo progetto è il modello Arduino Mega2560, di cui se ne riporta, in Figura 3.3, la faccia superiore, evidenziandone le parti principali, di cui alcune già accennate nel paragrafo precedente.

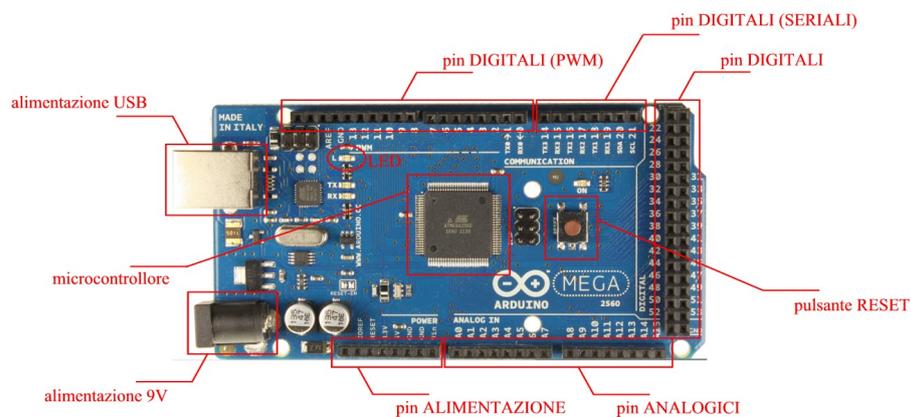


Figura 3.3: Arduino Mega2560 Fronte

Una veloce panoramica delle specifiche tecniche della scheda è riportata in Tabella 3.1.

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	54 (of which 14 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	40mA
DC Current for 3.3V Pin	50mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

Tabella 3.1: Tabella riassuntiva caratteristiche principali Arduino Mega2560

Si analizzano di seguito le principali componenti evidenziate in Figura 3.3.

- Microcontrollore ATmega2560:
 - *Programmazione*:
 - * programmato attraverso un convertitore USB-to-serial
 - * un bootloader permette di caricare un nuovo codice senza l'utilizzo di un programmatore hardware esterno. Il bootloader può essere anche bypassato, programmando il microcontrollore attraverso l'ICSP (In-Circuit Serial Programming)
 - * comunica utilizzando il protocollo originale STK500
 - *Memoria*:
 - * 256KB di flash memory per la memorizzazione del codice (di cui 8KB utilizzati per il bootloader)
 - * 8KB di SRAM
 - * 4KB di EEPROM (che può essere letta e scritta attraverso l'apposita EEPROM Library)
 - *Comunicazione*:
 - * 4 UARTs (hardware serial ports) per la comunicazione seriale TTL (5V), una delle quali è incanalata tramite USB e fornisce una porta COM virtuale al software del computer
 - * Quando i dati vengono trasmessi al microcontrollore o quando vi è una connessione al computer via USB lampeggiano i LED RX e TX presenti sulla scheda.
 - * Supporta comunicazioni del tipo TWI e SPI
- Alimentazione: Arduino Mega può essere alimentato tramite connessione USB o attraverso una fonte esterna. L'utilizzo di una o dell'altra sorgente è selezionato automaticamente. L'alimentazione esterna (non USB) può essere fornita da un adattatore AD/DC (wall-wart), inserendo lo spinotto da 2.1 mm nel jack presente sulla scheda, o da una semplice batteria, collegando i due poli ai pin Gnd e Vin. La scheda può lavorare con un'alimentazione esterna che va dai 6 ai 20 volt. Tuttavia, con una tensione inferiore ai 7V, i pin che lavorano a 5V forniscono una tensione inferiore ai 5V e la scheda può diventare instabile. Utilizzando, invece, una tensione superiore ai 12V, il regolatore di tensione può surriscaldarsi e danneggiare la scheda. Di conseguenza il range di tensione da preferire è quello compreso tra 7 e 12V.
I pin relativi all'alimentazione sono:

- V_{in} , tensione in ingresso alla scheda Arduino quando si usa una fonte di alimentazione esterna. E' possibile fornire in ingresso la tensione desiderata, attraverso questo pin.
 - $5V$, regolamentare alimentazione utilizzata per il microcontrollore e gli altri componenti presenti sulla scheda. Questa può essere fornita sia dal V_{in} , attraverso un regolatore on-board, sia da USB o da un qualsiasi alimentatore a $5V$.
 - $3V_3$, tensione di $3.3V$ generata da un regolatore on-board. Massimo assorbimento di corrente: $50mA$.
 - GND , pin di terra (ground).
- Input/Output: ognuno dei 54 pin digitali presenti sulla scheda può essere utilizzato sia come ingresso che come uscita, utilizzando le apposite funzioni *pinMode()*, *digitalWrite()* e *digitalRead()*. Essi operano alla tensione di $5V$. Ognuno di questi pin può fornire o ricevere una corrente massima di $40mA$ e ha al suo interno una resistenza di pull-up di $20-50k\Omega$. Di seguito si analizzano i pin più importanti tra quelli adibiti all'I/O, sia analogici che digitali:
 - *Serial*, vi sono ben quattro differenti linee seriali in Arduino Mega2560: *Serial* (pin 0 RX, pin 1 TX), *Serial1* (pin 19 RX, pin 18 TX), *Serial2* (pin 17 RX, pin 16 TX) e *Serial3* (pin 15 RX, pin 14 TX). Dove i pin RX e TX sono adibiti rispettivamente alla ricezione e alla trasmissione di dati seriali TTL.
 - *External Interrupts*, pin 2 (interrupt 0), pin 3 (interrupt 1), pin 18 (interrupt 5), pin 19 (interrupt 4), pin 20 (interrupt 3) e pin 21 (interrupt 2). Questi pin possono essere configurati per far scattare un interrupt in corrispondenza all'arrivo di un valore logico basso, di un fronte di salita o di discesa, o, infine, in corrispondenza di un cambio di valore (da alto a basso e viceversa). La funzione interessata è *attachInterrupt()*.
 - *PWM*, i pin dallo 0 al 13 sono in grado di fornire in uscita una PWM a 8-bit attraverso la funzione *analogWrite()*.
 - *LED*, è un LED del tipo built-in connesso al pin digitale 13. Quando il pin ha valore logico alto (HIGH), il LED è acceso, mentre quando il pin ha valore logico basso (LOW) il LED risulta spento.
 - *Analog Input*, ingressi analogici con risoluzione 10 bit (1024 valori differenti). Di default coprono un range che va da 0 (ground) a 5 volt, tuttavia è possibile modificare il limite superiore di questo range utilizzando il pin *AREF* e la funzione *analogReference()*.

- *AREF*, tensione di riferimento per gli ingressi analogici. Utilizzato con la funzione *analogReference()*.
- *Reset*, per resettare il microcontrollore è sufficiente portare a livello logico basso (LOW) questa linea. Generalmente viene utilizzato per aggiungere un ulteriore pulsante di reset sulle shield che rendono il pulsante di reset di Arduino inaccessibile.

3.2 SCHEDA DI GESTIONE INTEGRATA

Una scheda di gestione integrata è stata creata appositamente per essere impilata sulla scheda Arduino Mega 2560. Questa scheda ha il compito di assicurare i collegamenti dei vari dispositivi ad Arduino e proteggerli da eventuali sovratensioni. In Figura 3.4 se ne riporta una rappresentazione schematica (per lo schema elettrico si veda Appendice A.3), in cui vengono evidenziati i componenti principali, di seguito analizzati:

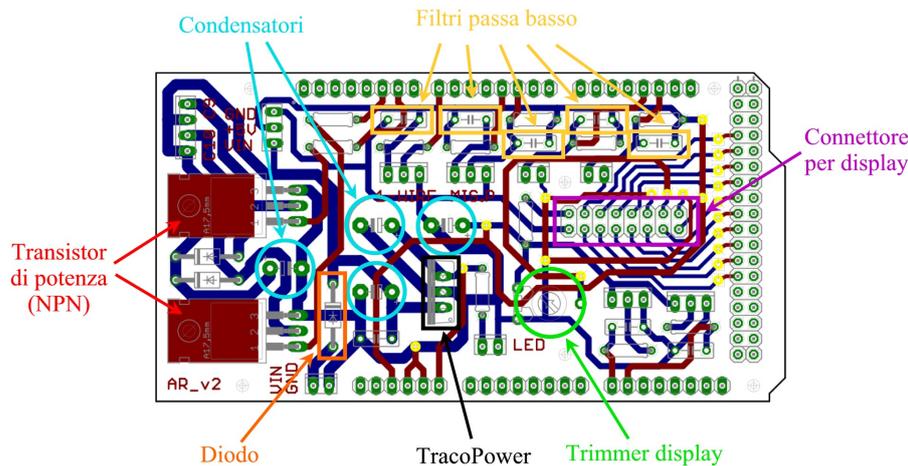


Figura 3.4: Schema scheda di gestione integrata

- *Transistor di potenza (NPN)*, per regolare l'alimentazione delle elettrovalvole
- *TracoPower*, convertitore switching, dotato di condensatori prima e dopo secondo la specifica richiesta. Trasforma i 24V che arrivano dall'alimentatore in 5V, ammessi da Arduino.
- *Condensatori*, elettrolitici di circa 100 μF , posizionati vicino ai transistor, per limitare i disturbi in zone in cui passa corrente impulsiva (che potrebbero arrivare alle elettrovalvole), e vicino al convertitore switching come richiesto da specifica.
- *Connettore per display*, per alimentare e comandare il display presente nel case interfaccia.

- *Trimmer display*, per la regolazione dell'intensità/luminosità del display.
- *Filtri passa basso*, per tutti gli ingressi, analogici e digitali, per contrastare gli eventuali disturbi ad alta frequenza, provenienti dalla PWM, in ingresso ai chip.
- *Diode*, protezione contro l'inversione di polarità nell'alimentazione
- *LED*, si accende quando c'è alimentazione sulla scheda

3.3 DISPOSITIVI PER L'INTERFACCIAMENTO

All'interno del case interfaccia, sede delle componenti elettroniche utilizzate, si trovano, oltre ad Arduino e alla scheda di gestione integrata, dei dispositivi addetti all'interfacciamento dell'utente con il sistema realizzato, quali:

- Display LCD alfanumerico
- N.2 potenziometri
- N.2 pulsanti monostabili

Il loro funzionamento è controllato, come si vedrà nel prossimo capitolo, da Arduino tramite la scheda di gestione integrata applicata su di esso.

REALIZZAZIONE PRIMO PROTOTIPO

Come già accennato, il progetto prevede l'utilizzo di due unità, idraulica e di controllo, che comunicano tra loro per garantire il corretto funzionamento del sistema rubinetto (come illustrato in Figura 4.1, figura già vista nel Capitolo 1 ma qui riportata per comodità). Nel presente capitolo viene analizzato il primo prototipo, realizzato sperimentalmente, caratterizzato dalla presenza di elettrovalvole proporzionali (sostituite da vitoni comandati da motori passo-passo nel secondo prototipo, analizzato in dettaglio nel Capitolo 5).

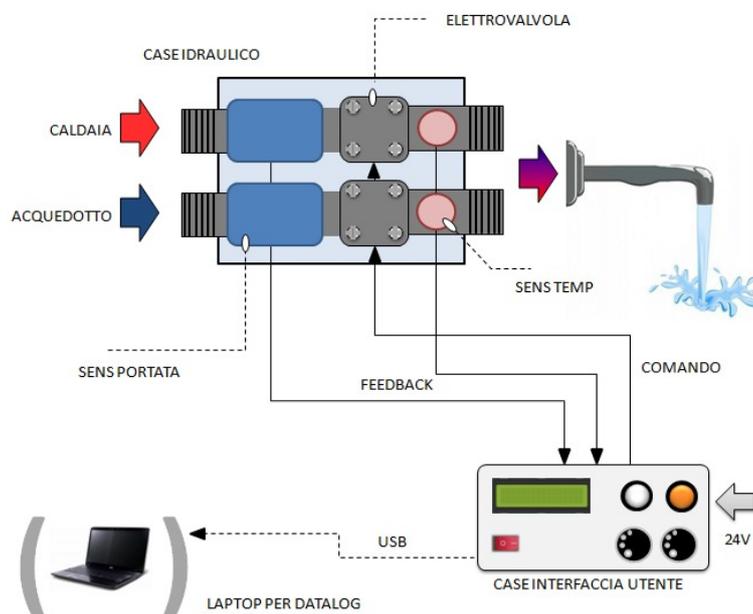


Figura 4.1: Rappresentazione schematizzata dell'apparato sperimentale

Il **controllo** dell'intero sistema è totalmente affidato ad **Arduino** (situato all'interno del case interfaccia), il quale, grazie al suo microcontrollore, è in grado di ricevere ed elaborare i dati provenienti dai sensori di temperatura e portata (visti nel Capitolo 2) e, in base a questi e alle specifiche richieste dall'utente che sta utilizzando il prodotto, inviare i corretti comandi alle elettrovalvole proporzionali (anch'esse analizzate nel Capitolo 2).

Di seguito si riporta una panoramica generale sull'apparato sperimentale realizzato, dopodiché si rivolge l'attenzione al controllo. Si descrivono, innanzitutto, le modalità con cui Arduino è in grado di ricevere ed elaborare i dati provenienti dai sensori utilizzati e di inviare gli opportuni segnali alle elettrovalvole, e, in un secondo momento,

si analizza nel dettaglio il controllo generale, utilizzato per l'intero sistema.

4.1 REALIZZAZIONE APPARATO SPERIMENTALE

4.1.1 Case interfaccia utente

In Figura 4.2 è riportata un'immagine dell'unità di controllo, in cui vengono evidenziati i vari componenti utilizzati:

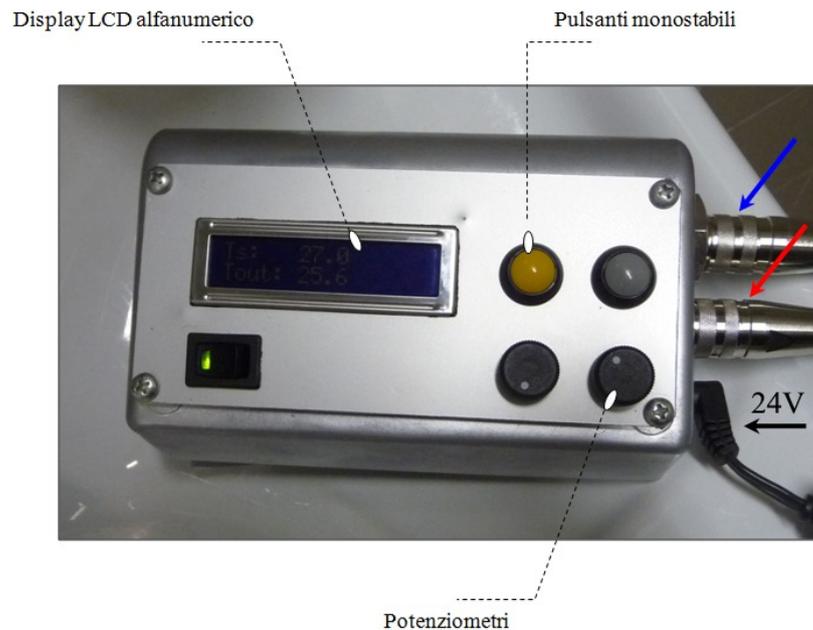


Figura 4.2: Case interfaccia utente

- *Interruttore di accensione*: per accendere e spegnere il dispositivo
- *2 Potenziometri*: dedicati alla regolazione della portata e della temperatura da parte dell'utente
- *Display LCD alfanumerico*: interfaccia grafica per la visualizzazione delle informazioni utili (portata e temperatura impostate, dati derivanti dai sensori, ecc.)
- *2 Pulsanti monostabili*: per scorrere, sul display, i diversi parametri con i relativi valori

Al case è collegato un alimentatore a 24V, che alimenta la scheda Arduino presente al suo interno (che lavora in modalità stand-alone) e le elettrovalvole proporzionali (tramite i due cavi indicati dalle frecce rossa e blu in Figura 4.2). All'interno del case sono contenute una scheda Arduino Mega 2560 e una scheda di gestione integrata direttamente applicata su di essa, già analizzate in dettaglio nel Capitolo 2.

4.1.2 Case idraulico

L'unità idraulica, come si può osservare dalle Figure 4.3 e 4.4, è costituita da

- 2 *Elettrovalvole proporzionali*, il cui grado di apertura è comandato da Arduino in modo tale da garantire acqua miscelata che soddisfi le specifiche (di portata e temperatura) richieste dall'utente
- 2 *Sensori di portata*, che restituiscono un feedback molto efficiente per le portate delle due linee
- 2 *Sensori di temperatura*, per il monitoraggio delle temperature dell'acqua proveniente rispettivamente dall'acquedotto e dalla caldaia

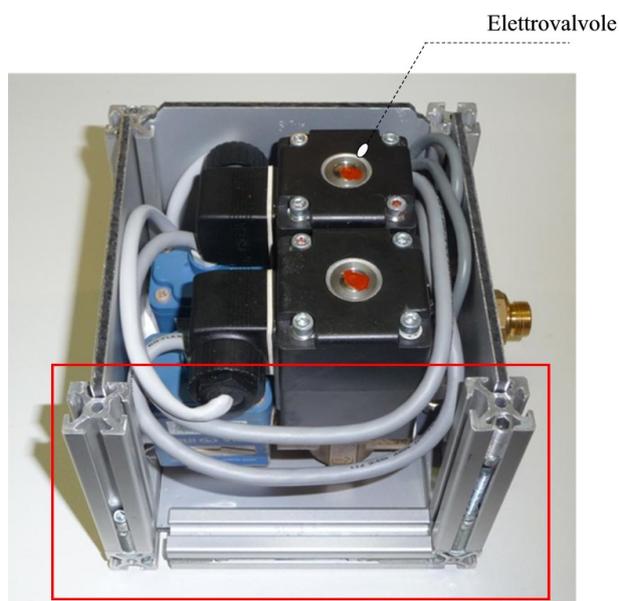


Figura 4.3: Case idraulico, vista dall'alto

In ingresso al case si hanno i due tubi di acqua derivante rispettivamente dall'acquedotto e dalla caldaia, mentre in uscita si hanno i due tubi che portano l'acqua al rubinetto, in quantità tale, per ognuna delle due linee, da garantire una corretta portata totale dell'acqua miscelata.

4.1.3 Sistema globale

Le unità così costituite vengono opportunamente collegate, tra loro e al rubinetto, per verificarne il corretto funzionamento. In Figura 4.5 è riportata un'immagine del sistema globale così ottenuto, in cui sono state evidenziate le due unità viste.

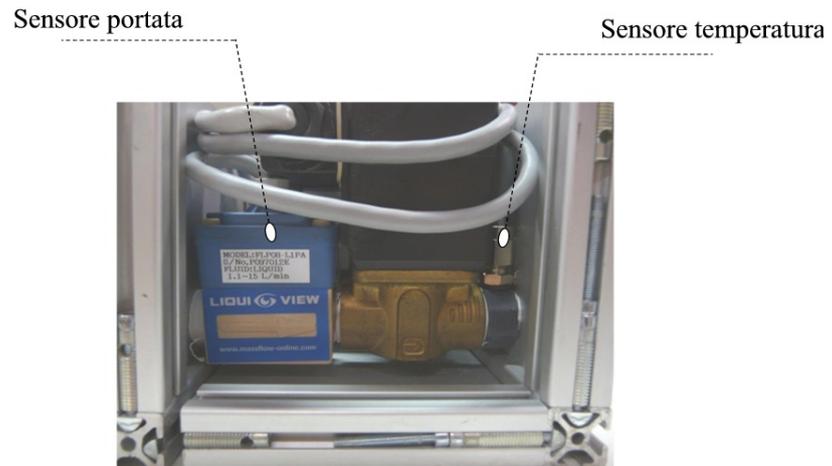


Figura 4.4: Case idraulico, vista frontale



Figura 4.5: Foto del sistema fisico realizzato per il primo prototipo

4.2 CONTROLLO VALVOLE E DISPOSITIVI PER LA MISURA

Il controllo e la gestione dei vari dispositivi utilizzati sono governati, come già detto, dal microcontrollore di cui è dotato Arduino. Sulla scheda, infatti, viene caricato un programma, appositamente creato, che fornisce ad essa le corrette istruzioni per coordinare il funzionamento dell'intero sistema. Di seguito si analizzano le modalità con cui essa opera con i differenti dispositivi collegati.

4.2.1 Sensori di temperatura

Si ricorda, come visto nel Paragrafo 2.3, che i dati provenienti dai sensori di temperatura utilizzati (sonde MAXIM DS18S20) sono memorizzati in uno scratchpad (di cui se ne riporta un'immagine in Figura 4.6). Il programma creato in Arduino, per la lettura dei dati

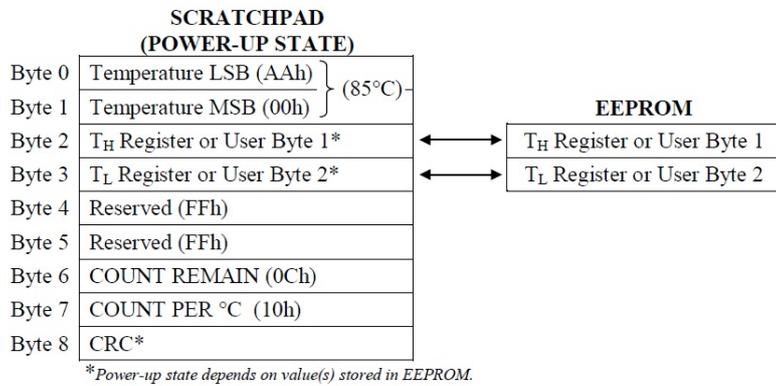


Figura 4.6: DS18S20 Memory Map

provenienti da questi sensori, estrapola le informazioni utili dai relativi scratchpad seguendo una specifica routine, che procede come segue:

- Rilevazione presenza di dispositivi e individuazione dei relativi indirizzi
- Richiamo della funzione *convertT*, che esegue, per ogni dispositivo individuato, la conversione della temperatura rilevata e salva i dati nei primi due bytes dello scratchpad
- Verifica CRC (Cyclic Redundancy Check): una volta ottenuto il dato dal sensore e riempito adeguatamente lo scratchpad, si esegue un confronto tra il CRC ottenuto con quello effettivo del sensore e, in caso di incongruenza (quindi in caso di guasto, manomissione del sensore o errore di trasmissione del dato), si mantiene in memoria il valore di temperatura precedente, fintantoché non viene ripristinato il corretto funzionamento del sensore stesso.

- Lettura dati scratchpad: il valore di temperatura è memorizzato nei byte 0 e 1 (TEMP_READ) come numero in complemento a due a 16 bit. Il calcolo della temperatura effettiva in °C si ottiene con la (4.1), che utilizza le variabili salvate nei bytes 6 e 7 in Figura 4.6, presente nel datasheet del sensore e qui riportata

$$\text{TEMP} = \text{TEMP_READ} - 0.25 + \frac{\text{COUNT PER } ^\circ\text{C} - \text{COUNT REMAIN}}{\text{COUNT PER } ^\circ\text{C}} \quad (4.1)$$

4.2.2 Sensori di portata

Come visto nel Paragrafo 2.2, questi dispositivi sono dotati di un sensore piezoelettrico che individua i vortici causati dal corpo di Bluff e che genera, di conseguenza, degli impulsi elettrici proporzionali alla portata del liquido stesso. Il segnale di uscita che ne deriva è un'onda quadra a frequenza variabile e duty-cycle fisso al 50%, di ampiezza 5V. La lettura di questi dati, e quindi la portata delle due linee ad ogni istante, è resa possibile da un programma appositamente creato che prevede il campionamento dei fronti di salita delle onde quadre (o salita e discesa, nel caso si desideri una migliore risoluzione), conteggiati ed elaborati nel seguente modo:

- Si definisce il *tempo ciclo* desiderato, ossia la finestra temporale all'interno della quale si contano gli impulsi in ingresso (fronti di salita e di discesa della PWM proveniente dal sensore). Questa operazione di conteggio viene effettuata tramite *interrupts*: ad ogni fronte (di salita e/o di discesa) viene richiamata una routine (precedentemente definita) che incrementa un contatore di impulsi. In ambiente Arduino, il comando utilizzato diventa:

attachInterrupt(0, increaseH, CHANGE);

dove

- 0, è il numero di interrupt che viene richiamato (in Arduino ve ne sono differenti da poter utilizzare, ognuno dei quali richiama una particolare routine sotto particolari condizioni fissate di volta in volta)
- *increaseH*, è la routine che viene richiamata quando si verifica l'interrupt (H sta ad indicare che si sta analizzando la linea calda)
- *CHANGE*, indica che viene generato l'interrupt ogni volta che il pin esaminato cambia valore (da 0 a 1 o viceversa, analizzando i fronti di salita e di discesa)

Nel caso in esame è stato scelto un tempo ciclo $T_C = 80\text{ms}$.

- Il numero di impulsi ottenuto viene convertito in una misura di portata, ossia L/min. Per fare ciò innanzitutto si usa un prescaler di valore 64, per ridurre la frequenza con cui il contatore del timer viene incrementato (originariamente 16kHz). Dopodiché si moltiplica il valore così ottenuto per un coefficiente di portata pari a $k_q = 0.0264\text{L/min}$ (nel caso di conteggio dei soli fronti di salita, mentre $k_q = 0.0132\text{L/min}$ se si considerano anche i fronti di discesa). I coefficienti k_q si ottengono a partire delle specifiche del sensore di portata utilizzato: si ha un fattore di conversione unitario degli impulsi in uscita pari a 0.4408mL/P . Il segnale che arriva dal sensore è di 1Hz, quindi, se si considera 1 impulso in 1s, il fattore di conversione diventa 0.4408mL/s , ossia $0.4408 \cdot 60\text{mL/min}$ e dunque 0.02644L/min . Questo valore deve essere dimezzato nel caso in cui si considerino due impulsi (fronti di salita e di discesa) in 1s. Dal momento che si sta lavorando con una finestra di 80ms, si deve adattare il parametro di conseguenza, dividendolo per il range di tempo della finestra che si sceglie.
- Dal momento che la portata subisce variazioni che non sono istantanee e vi è un'inerzia termica nel rubinetto, per poter avere un dato in uscita più preciso, si utilizza un filtro a media mobile che memorizza, oltre al dato ottenuto dall'ultima acquisizione ($flow1$), i due dati precedenti ($flow2$ e $flow3$), pesandoli in maniera differente, come riportato in Figura 4.7.

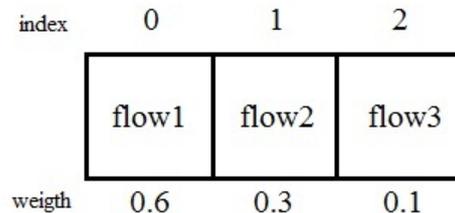


Figura 4.7: Struttura filtro a media mobile per lettura portate

Questo filtro evita soprattutto il manifestarsi di *ripple* nella lettura delle portate, dovuti all'effetto del campionamento. La finestra di analisi, infatti, è discreta e tra la lettura di un fronte e il successivo si ha una carenza di informazioni. In Figura 4.8 si riporta un semplice esempio esplicativo. Si suppone di avere una finestra di osservazione di 1s e un tempo ciclo $T_C = 100\text{ms}$: ogni 100ms si contano gli impulsi in arrivo dal sensore. Nelle finestre temporali di 100ms in cui non si hanno fronti di salita o di discesa, e quindi non si hanno impulsi da contare, secondo il sensore la portata è nulla. In questo modo c'è un continuo alternarsi di valori di portata diversi da zero e uguali a zero nelle fasi di transizione da una finestra alla successiva. Il filtro a

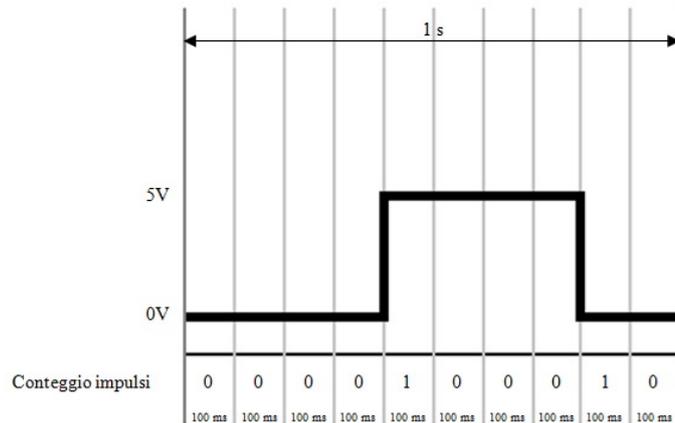


Figura 4.8: Conteggio impulsi su finestra temporale di 1s

media mobile realizzato tende a mitigare questo effetto. Si noti che, utilizzando $k_q = 0.0264\text{L}/\text{min}$ si riscontra un errore del 10 %, che può essere ridotto al 5 % con $k_q = 0.0132\text{L}/\text{min}$, al quale va aggiunto un 3 % di errore intrinseco del sensore di portata stesso.

4.2.3 Elettrovalvole proporzionali

Per poter studiare accuratamente il ruolo di Arduino nel comando delle elettrovalvole, è utile analizzare, innanzitutto, lo schema elettrico relativo al loro reciproco collegamento elettrico (Figura 4.9). L'in-

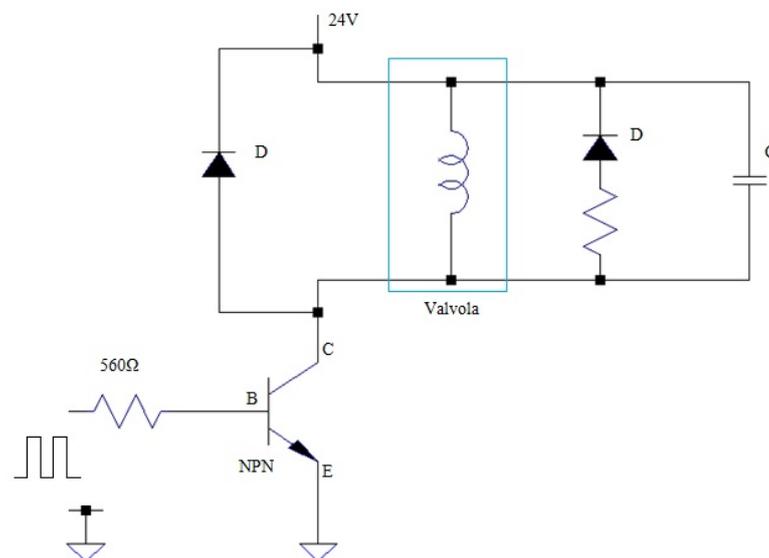


Figura 4.9: Schema elettrico collegamento valvola

gresso è costituito da una PWM, derivante da Arduino, di ampiezza 5V, che aziona il transistor, il quale, in presenza di un fronte di salita

(ON) dell'onda quadra, chiude il relè statico, la valvola viene alimentata da un trasformatore con una tensione di 24V e al suo interno comincia a scorrere una corrente. All'arrivo del fronte di discesa della PWM (OFF), il relè statico si apre e la corrente presente all'interno del circuito viene dissipata, in larga misura, dal condensatore C e dal diodo D (posti in parallelo alla valvola). La presenza di questi componenti riduce in modo importante gli spikes che si hanno in corrispondenza al fronte di discesa della PWM, dovuti a un picco di corrente che si ha sul transistor in fase di carica.

L'onda quadra generata da Arduino è a frequenza fissa ma duty-cycle variabile, il cui valore viene opportunamente calcolato in base a temperatura e portata richieste e quindi all'apertura desiderata per la valvola in esame. Il comando utilizzato, ad esempio per la linea calda, è

analogWrite(VALVEH_PIN, dutyHot);

che "scrive" sul Pin analogico, collegato alla corrispondente valvola, il valore calcolato di duty cycle per la linea, rappresentato da un numero che va da 0 (valvola totalmente chiusa) a 255 (valvola totalmente aperta).

4.3 CONTROLLO GENERALE

Il programma caricato in Arduino rileva, ad ogni ciclo, le informazioni necessarie dai sensori di temperatura e portata e, sulla base di questi dati, individua il corretto grado di apertura delle elettrovalvole e invia loro un opportuno segnale. In Figura 4.10 è riportato uno schema a blocchi rappresentante il controllo ideato a questo scopo.

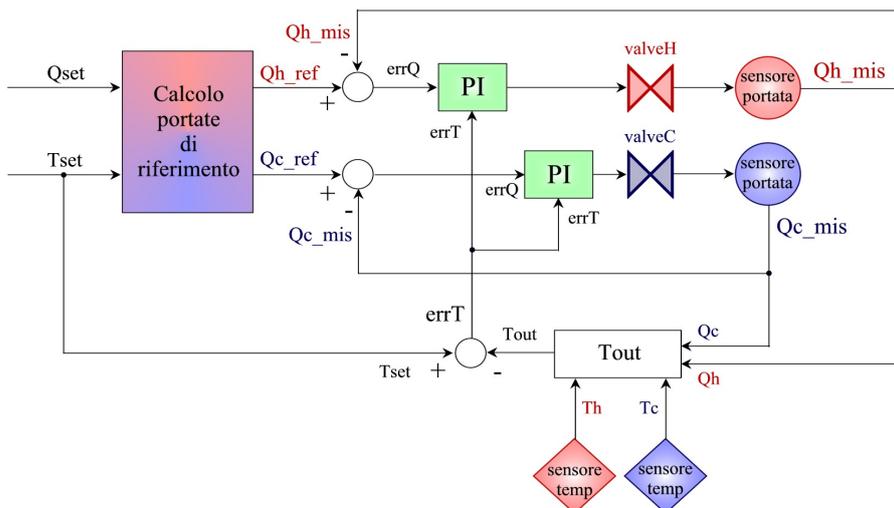


Figura 4.10: Schema a blocchi del controllo relativo al primo prototipo

Come si può vedere, si tratta di un semplice controllo in retroazione, di tipo **PI (Proporzionale-Integrativo)**, il cui scopo è calcolare, sulla base delle specifiche di temperatura (T_{set}) e di portata totale (Q_{set}) desiderate dall'utente, il grado di apertura delle elettrovalvole. Una volta note le specifiche richieste, infatti, si procede al calcolo delle portate di riferimento (Q_{h_ref} e Q_{c_ref}), ossia le portate che devono essere ottenute rispettivamente nelle due linee al fine di ottenere, in uscita, acqua miscelata tale da soddisfare i parametri desiderati. Per ottenere i due valori cercati è sufficiente risolvere il sistema di due equazioni in due incognite:

$$\begin{cases} T_{set} = \frac{T_h \cdot Q_{h_ref} + T_c \cdot Q_{c_ref}}{Q_{h_ref} + Q_{c_ref}} \\ Q_{set} = Q_{h_ref} + Q_{c_ref} \end{cases} \quad (4.2)$$

in cui T_{set} viene espressa come media pesata basata sulle portate di riferimento e sulle temperature relative ai tubi dell'acqua calda e fredda (monitorate dai relativi sensori), mentre Q_{set} viene imposta uguale alla somma delle due portate di riferimento, in modo da assicurare la corretta portata totale.

Una volta definiti i setpoint da inseguire, si procede al controllo PI. Per ottenere migliori prestazioni, soprattutto nel caso di strozzature di una delle due portate, si è deciso di fare un doppio controllo PI, che agisca in maniera incrociata sulla **portata** e sulla **temperatura**. In ingresso ai blocchi "PI", infatti, arrivano l'errore relativo alla portata, per le due linee ($errQ_h$ e $errQ_c$), e quello relativo alla temperatura ($errT$), calcolati come segue

$$errQ_h = Q_{h_ref} - Q_{h_mis}, \quad (4.3)$$

$$errQ_c = Q_{c_ref} - Q_{c_mis}, \quad (4.4)$$

$$errT = T_{set} - T_{out}, \quad (4.5)$$

con lo scopo di annullarli.

Nota: Q_{h_mis} e Q_{c_mis} derivano da un'opportuna lettura dei sensori di portata, mentre T_{out} è la temperatura stimata dell'acqua in uscita dal rubinetto, ottenuta come media pesata, riferita alle portate e alle temperature (T_h e T_c , rilevate dai sensori) per le due linee, come espresso dalla seguente

$$T_{out} = \frac{T_h \cdot Q_{h_ref} + T_c \cdot Q_{c_ref}}{Q_{h_ref} + Q_{c_ref}}. \quad (4.6)$$

Se $Q_{h_ref} + Q_{c_ref} = 0$, T_{out} viene posta uguale alla media aritmetica delle temperature T_h e T_c .

L'introduzione del controllo sulla temperatura, in aggiunta a quello sulla portata, ha lo scopo di velocizzare la dinamica di correzione nel transitorio. Si riportano, a titolo esemplificativo, alcune righe di codice estrapolate dallo sketch realizzato per Arduino relative al calcolo della parte proporzionale e integrale per la linea calda ([0]) e per quella fredda ([1])

```
errProp[0] = kp_q*errQ[0] + k_t*qR[0]*errT;
errInt[0] = errIntOld[0] + ki_q*errQ[0] + k_t*qR[0]*errT;

errProp[1] = kp_q*errQ[1] - k_t*qR[1]*errT;
errInt[1] = errIntOld[1] + ki_q*errQ[1] + k_t*qR[1]*errT;

dutyHot = round(errInt[0] + errProp[0]);
dutyCold = round(errInt[1] + errProp[1]);
```

dove

- I parametri **kp_q**, **ki_q**, **k_t** sono quelli relativi rispettivamente all'azione proporzionale e integrale per la portata e per la temperatura (**k_t**), i cui valori sono stati modificati tramite metodo *trial-and-error*, fino ad ottenere le prestazioni desiderate.
- I valori **errQ[0]**, **errQ[1]** e **errT** corrispondono, rispettivamente, ai valori calcolati con (4.3)-(4.5).
- Il contributo dato dall'errore di temperatura, sia nell'azione proporzionale che in quella integrale ($k_t*qR[0]*errT$), serve a velocizzare la dinamica, che sarebbe altrimenti rallentata dalla componente legata alla portata, e a poter individuare e gestire anche eventuali situazioni di emergenza, quali strozzature, cali di pressione, ecc.. Si analizzano, ad esempio, le azioni proporzionali per le due linee, per spiegare in modo più efficace il loro ruolo.

```
errProp[0] = kp_q*errQ[0] + k_t*qR[0]*errT;
```

- $kp_q*errQ[0] > 0$, se $Q_{h_{ref}} - Q_{h_{mis}} > 0$ e quindi se $Q_{h_{ref}} > Q_{h_{mis}}$. In questo caso la portata calda di riferimento è maggiore rispetto a quella effettivamente misurata e ottenuta in uscita al sistema e quindi la componente proporzionale deve essere incrementata, con conseguente aumento dell'ampiezza dell'apertura della valvola relativa alla linea calda (in maniera duale, se $Q_{h_{ref}} < Q_{h_{mis}}$ si assiste a un decremento)
- $k_t*qR[0]*errT > 0$, se $T_{set} - T_{out} > 0$ e quindi se $T_{set} > T_{out}$. In questo caso la temperatura fissata dall'utente risulta maggiore rispetto a quella che si ha in uscita al sistema

(si ricordi che si dispone di una temperatura in uscita solo stimata, ottenuta a partire dalle informazioni relative alle portate in uscita, effettivamente monitorate). La componente proporzionale subirà quindi un ulteriore incremento in questo caso, comportando una maggiore apertura della valvola corrispondente. In maniera duale, se $T_{set} < T_{out}$ si assisterà a un decremento dell'angolo.

Per quanto riguarda il canale dell'acqua fredda ([1]) le cose sono molto simili. Ciò che effettivamente cambia è la presenza di un segno 'meno' tra le due componenti proporzionali considerate:

$$\text{errProp}[1] = k_p \cdot \text{errQ}[1] - k_t \cdot \text{qR}[1] \cdot \text{errT};$$

Questo fatto può essere spiegato in maniera molto semplice: $k_t \cdot \text{qR}[1] \cdot \text{errT} > 0$ se $T_{set} - T_{out} > 0$ e quindi se $T_{set} > T_{out}$. Dal momento che si tratta della linea relativa all'acqua fredda, questa condizione implica che deve essere garantita una diminuzione dell'apertura della valvola corrispondente, perché è richiesta una temperatura maggiore rispetto a quella effettivamente ottenuta dall'acqua erogata. In modo duale, il caso $T_{set} < T_{out}$ (si desidera acqua in uscita più fredda), restituisce la seconda componente negativa e un conseguente incremento della relativa componente proporzionale.

Nota: nel caso in cui si ottengano valori di $\text{errInt}[0]$ o di $\text{errInt}[1]$ non compresi nell'intervallo 0-255, ammesso dalla funzione $\text{analogWrite}()$ utilizzata per inviare il comando di apertura alle valvole, vengono opportunamente forzati all'uno o all'altro estremo, eseguendo una saturazione

- **errIntOld[0]** e **errIntOld[1]** sono i valori dell'integrale ottenuti, per le due linee, al passo precedente, ai quali vanno aggiunti i nuovi opportuni incrementi/decrementi.
- **dutyHot** e **dutyCold** sono i valori di duty cycle corretti da inviare alle due elettrovalvole, calcolati semplicemente sommando i due contributi, proporzionale e integrale, ottenuti. Viene effettuata un'operazione di arrotondamento (*round*) perchè i valori ammessi dalla funzione $\text{analogWrite}()$, utilizzata per inviare i comandi alle valvole, sono numeri interi compresi tra 0 e 255.

Nota: nel caso in cui si ottengano valori di dutyHot o di dutyCold non compresi nell'intervallo 0-255, questi vengono opportunamente forzati all'uno o all'altro estremo, eseguendo una saturazione

4.3.1 Gestione casi eccezionali

Nella realtà si incorre, molto spesso, in fenomeni "indesiderati", quali strozzature, cali di pressione, specifiche richieste inammissibili, ecc..

Il controllo realizzato, al verificarsi di questi fenomeni, esegue particolari routine, atte a garantire ugualmente all'utente, se possibile, le prestazioni attese o, in caso contrario, a difenderlo da eventuali danni. Di seguito si percorre una panoramica dei possibili casi eccezionali, analizzandone, per ognuno, l'azione intrapresa dal controllore.

- $T_h - T_c \leq 2$

Le temperature registrate lungo i due tubi sono molto simili (ad esempio quando il tubo dell'acqua fredda passa esternamente all'edificio ed è colpito dal sole estivo, oppure quando, caso più probabile, è stata utilizzata tutta l'acqua calda contenuta nel boiler)

Azione intrapresa: se $T_{set} < T_h$, si comanda l'apertura della sola valvola fredda, assicurando la portata totale desiderata dall'utente, Q_{set} ; altrimenti, in caso contrario ($T_{set} \geq T_h$), si comanda, con lo stesso criterio, l'apertura della sola valvola dell'acqua calda.

- $T_{set} < T_c$

La temperatura richiesta per l'acqua in uscita dal rubinetto è più bassa della temperatura dell'acqua proveniente dal tubo della linea fredda.

Azione intrapresa: si comanda l'apertura della sola valvola relativa all'acqua fredda, con portata pari a quella totale richiesta dall'utente, Q_{set} . Inoltre si pone "flag_tempSat = true". Quando il controllo deve procedere al calcolo dell'errore di temperatura, se questa flag risulta alzata, lo pone direttamente pari a 0. Lo scopo di questa operazione è evitare comportamenti anomali causati dal continuo manifestarsi di uno sfasamento tra le richieste da parte dell'utente e le temperature realmente ottenibili.

- $T_{set} > T_h$

La temperatura richiesta per l'acqua in uscita dal rubinetto è più alta della temperatura dell'acqua proveniente dal tubo della linea calda.

Azione intrapresa: si comanda l'apertura della sola valvola relativa all'acqua calda, con portata pari a quella totale richiesta dall'utente, Q_{set} . Inoltre si pone "flag_tempSat = true". Quando il controllo deve procedere al calcolo dell'errore di temperatura, se questa flag risulta alzata, lo pone direttamente pari a 0. Lo scopo di questa operazione è evitare comportamenti anomali causati dal continuo manifestarsi di uno sfasamento tra le richieste da parte dell'utente e le temperature realmente ottenibili.

Se nessuna di queste eccezioni è verificata, ci si trova nelle condizioni normali di funzionamento e il controllo procede al calcolo delle portate di riferimento, seguendo la procedura vista sopra.

Il sistema così ottenuto è stato sottoposto a differenti prove sperimentali (Capitolo 6), per analizzarne le prestazioni sia in caso di funzionamento normale che anomalo.

REALIZZAZIONE SECONDO PROTOTIPO

La nuova soluzione ipotizzata per il complesso *rubinetto - case idraulico - case interfaccia* prevede un'importante modifica: la sostituzione delle elettrovalvole proporzionali con **vitoni da 1/2 pollice comandati da motori passo-passo**. Lo scopo di questa operazione è quello di ridurre costi, pesi e dimensioni dell'unità idraulica, pur mantenendo la stessa efficienza dell'intero sistema. Prima di realizzare questo secondo progetto fisicamente (fase di lavoro futura), si è deciso di implementarlo su base simulativa. A questo scopo si è rivelato necessario l'utilizzo dell'ambiente Matlab/Simulink, con il quale è stato implementato un modello matematico dell'intero sistema, rivolgendo un'attenzione particolare ai vitoni. È stato, infatti, studiato nel dettaglio il comportamento del nuovo blocco idraulico (tralasciando momentaneamente la dinamica dei sensori presenti al suo interno) e si è cercato di riprodurre un modello, il più fedele possibile. Una volta ultimata questa fase di lavoro, il modello parziale così realizzato è stato introdotto all'interno del sistema generale di controllo e sono state effettuate diverse simulazioni, atte a verificarne la validità e la realizzabilità (i risultati sono riportati nel Capitolo 6).

5.1 MODELLO BLOCCO IDRAULICO

5.1.1 Descrizione del modello

Si analizza, innanzitutto, una rappresentazione schematizzata (Figura 5.1) del blocco idraulico, la quale si rivela utile principalmente per l'individuazione e l'analisi delle perdite idrauliche ([6]) che si sviluppano nei diversi punti del sistema. Una volta ultimato questo passaggio, è possibile realizzare una corretta riproduzione del sistema in ambiente Matlab/Simulink, con un modello che riporti le stesse caratteristiche e gli stessi I/O del sistema reale.

In figura si riconoscono i due tubi dell'acqua, calda (in alto) e fredda (in basso), caratterizzati entrambi da un restringimento, dato dalla presenza del vitone, che confluiscono poi in un unico tubo attraverso il quale passa l'acqua miscelata, in uscita dal rubinetto, la cui temperatura dovrebbe corrispondere a quella richiesta dall'utente. L'ultima parte del tubo, dopo la confluenza, prevede la presenza di un *aeratore*, presente generalmente nei rubinetti, e che provoca, anch'esso, delle perdite di energia.

Per analizzare correttamente l'assetto energetico dello schema riportato si suggerisce di guardarlo a partire dal suo lato destro, procedendo

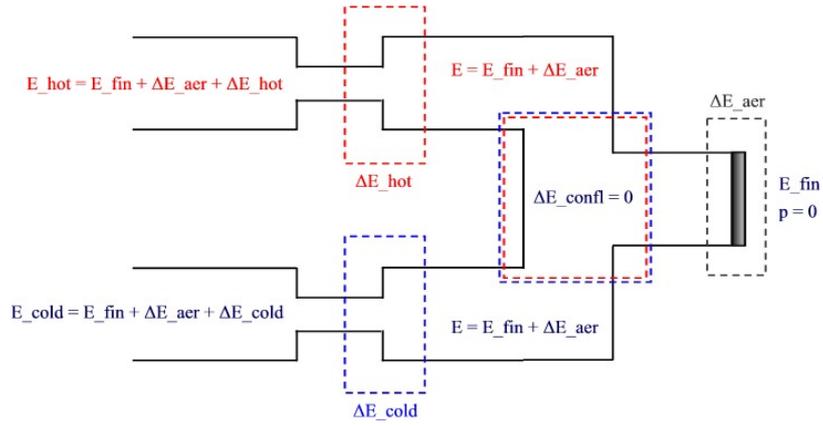


Figura 5.1: Schema blocco rubinetto con analisi perdite idrauliche

poi a ritroso verso il lato sinistro. Ragionando in questo modo, infatti, si individuano tutte le perdite localizzate di energia presenti nei tratti di tubo percorsi dall'acqua, calda, fredda e miscelata, e si risale all'energia totale che si ha in ingresso ai due tubi, governata dalla generica relazione:

$$E_{\text{tot}} = \frac{p}{\gamma} + \frac{v^2}{2g}, \quad (5.1)$$

con

- p , pressione fluido [Pa]
- γ , peso specifico [N/m^3]
- v , velocità media fluido all'ingresso del tratto di tubo [m/s]
- g , accelerazione di gravità [m/s^2]

Si consideri che, in generale, le perdite di energia localizzate sono definite come

$$\Delta E_{\text{tot}} = \xi \cdot \frac{v^2}{2g}, \quad (5.2)$$

dove ξ è un valore, stimato in modo analitico o determinato sperimentalmente, che esprime l'entità della perdita localizzata e v è la velocità media del fluido all'ingresso del tratto del tubo in cui si verifica la perdita localizzata.

Procedendo a ritroso, noto che all'uscita dell'aeratore la pressione dell'acqua è nulla ($p = 0$), si che che l'energia totale finale (all'uscita del rubinetto) è

$$E_{\text{fin}} = \frac{v^2}{2g} = \frac{(Q_h + Q_c)^2}{2g \cdot \frac{A^2}{3^2}}, \quad (5.3)$$

dove si è utilizzata, per la velocità, la relazione

$$v = \frac{Q}{A}, \quad (5.4)$$

valida nel caso in cui siano note le portate (in questo caso dell'acqua calda e fredda in uscita) e l'area A della sezione attraverso la quale scorre il fluido in questione che, per un tubo a sezione circolare, risulta:

$$A_t = \frac{\pi \cdot d_t^2}{4}. \quad (5.5)$$

In questo caso si ha $A/3$, ipotizzando che la sezione del tubo risulti ridotta di un fattore 3 per effetto dell'aeratore.

Le diverse perdite localizzate provocate dalla presenza di aeratore e vitoni (su tubi acqua calda e acqua fredda) sono date, invece, rispettivamente da:

$$\Delta E_{aer} = \xi \frac{v^2}{2g} = \xi \frac{(Q_h + Q_c)^2}{2g \cdot A^2}, \quad (5.6)$$

$$\Delta E_{hot} = \xi \frac{v^2}{2g} = \xi \frac{Q_h^2}{2g \cdot A_h^2}, \quad (5.7)$$

$$\Delta E_{cold} = \xi \frac{v^2}{2g} = \xi \frac{Q_c^2}{2g \cdot A_c^2}, \quad (5.8)$$

dove A_h e A_c costituiscono le sezioni di tubo, attraversate rispettivamente dall'acqua calda e dall'acqua fredda, che variano al variare dell'angolo θ di apertura dei corrispondenti vitoni, secondo la relazione

$$A_v = k_v \frac{\pi \cdot d_v^2}{8} \cdot \frac{\theta}{180}, \quad (5.9)$$

Q_h e Q_c sono le portate dei due tubi, mentre ξ , in questo caso, viene definito

$$\xi = \left(\frac{A_v}{A_t} - 1 \right)^2 = \left(\frac{k_v \frac{\pi \cdot d_v^2}{8} \cdot \frac{\theta}{180}}{A_t} - 1 \right)^2. \quad (5.10)$$

Si noti che è stata supposta nulla la perdita localizzata in corrispondenza della confluenza ($\Delta E_{confl} = 0$), trascurabile nelle condizioni di lavoro considerate.

5.1.2 Implementazione del modello in ambiente Matlab/Simulink

Il modello creato viene riproposto, in modo riadattato, in ambiente Matlab/Simulink. L'idea di base è quella di creare un sistema in cui, date le pressioni a monte delle valvole e dato il grado di apertura di ciascuna valvola, si calcola la portata su ciascuna linea. In Figura 5.2 si osserva il sistema così realizzato.

In ingresso al blocco simulatore si hanno le pressioni relative ai due tubi dell'acqua calda e dell'acqua fredda, convertite in Pascal (conveniente per poter utilizzare le relazioni di variazione di energia e perdite di carico localizzate viste in precedenza), e l'apertura dei due relativi vitoni (per il momento questi parametri vengono impostati manualmente dall'utente). Queste informazioni, insieme alle portate calcolate al passo precedente, costituiscono gli ingressi della funzione Matlab *pressure.m*, che procede al calcolo delle portate stesse, individuate imponendo che le pressioni che si hanno in uscita dalla funzione siano le stesse che si hanno in ingresso. Il risultato desiderato, infatti, viene ottenuto ragionando a ritroso: rifacendosi anche allo schema di Figura 5.1, una volta calcolate E_{fin} , ΔE_{aer} e A_v e ξ per le due linee calda e fredda, e quindi le rispettive ΔE_{hot} e ΔE_{cold} , si procede al calcolo della pressione che si ha all'ingresso dei due tubi in queste condizioni, utilizzando la (5.1), qui riadattata come segue:

$$p_{h_{mis}} = \left((E_{fin} + \Delta E_{aer} + \Delta E_{hot} + \Delta E_{tubo-hot}) - \frac{\left(\frac{Q_h}{A_t}\right)^2}{2g} \right) \cdot \gamma, \quad (5.11)$$

cercando quindi di far combaciare il risultato così ottenuto con le pressioni effettive misurate a monte e ricevute in ingresso al blocco. Questo risultato viene raggiunto grazie all'uso di un semplice integratore (k_i + blocco integratore), che riceve in ingresso l'errore di pressione della linea considerata (pressione dell'acquedotto, qui fissata dall'utente, e pressione calcolata come in (5.11)) cercando via via di annullarlo e restituendo la rispettiva portata, che raggiunge il suo valore corretto in corrispondenza al raggiungimento dell'errore nullo sulle pressioni.

Note:

- *Nell'energia totale a monte è stata considerata un'ulteriore perdita di energia presente in genere nel tubo, che riassume eventuali perdite non considerate per semplificare i conti*
- *D'ora in avanti si considererà, per semplicità, il solo caso della linea calda, ricordando che, per analogia, si ricavano anche le informazioni relative a quella fredda*

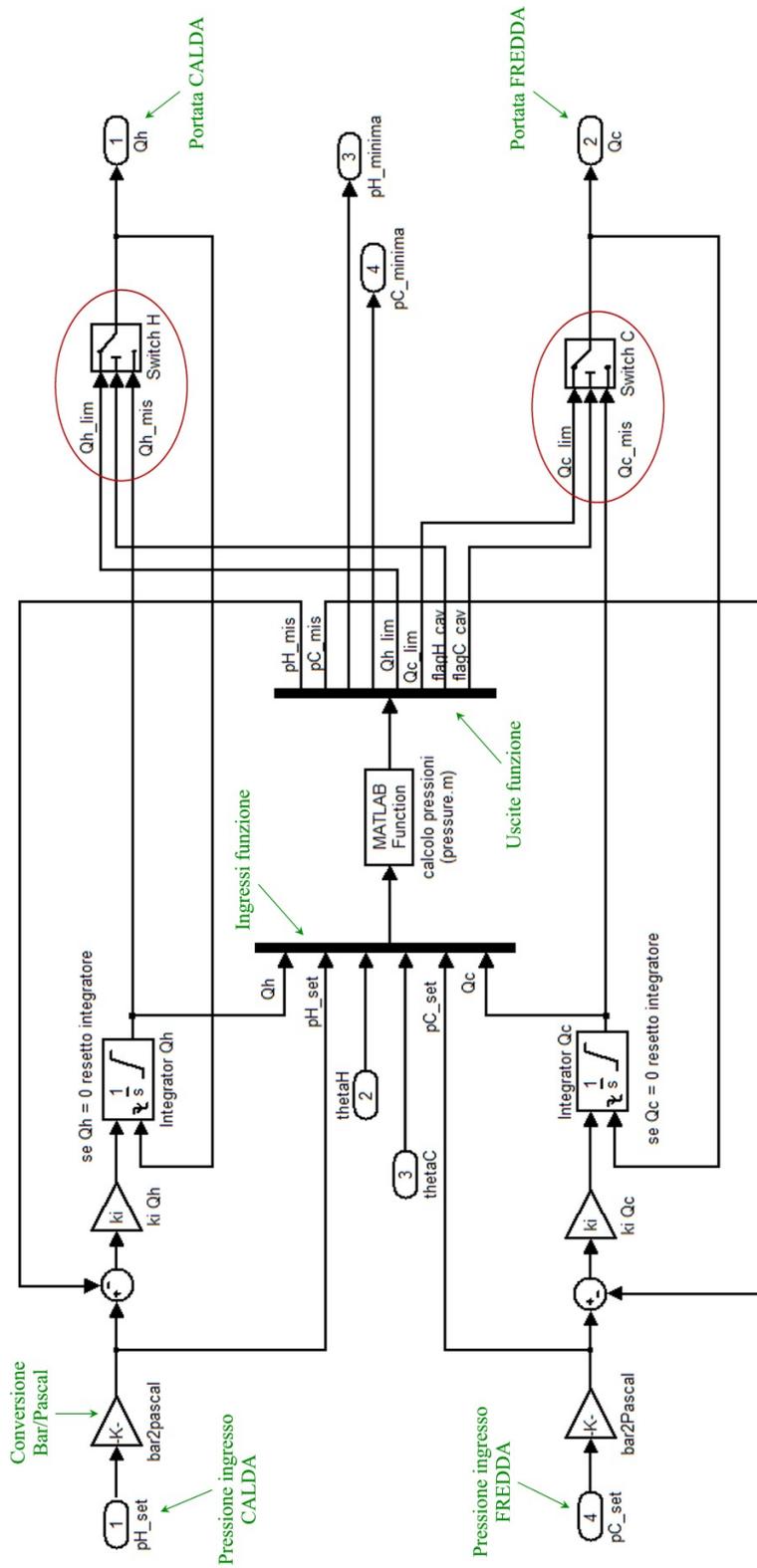


Figura 5.2: Modello Simulink realizzato per simulare il blocco rubinetto

Osservando attentamente lo schema di Figura 5.2, si nota che in uscita dal blocco "MATLAB Function - *pressure.m*", vi sono anche altri parametri che sono legati al fenomeno della cavitazione. La cavitazione si presenta ogni volta che la pressione statica locale scende al di sotto della tensione di vapore del liquido, e consiste nella *vaporizzazione locale* del fluido stesso. Generalmente questo fenomeno si presenta quando si ha un restringimento della condotta, infatti, quando il fluido è costretto a percorrere passaggi molto stretti, la pressione raggiunge livelli *inferiori* a quelli della pressione atmosferica. In questi casi si assiste alla formazione di una o più bolle all'interno del tubo, causate proprio dalle forti depressioni che si determinano nelle zone di "strangolamento" del fluido, con conseguente diminuzione della velocità dello stesso e aumento di perdite di energia. Le bollicine generalmente implodono quando il flusso ridiventa normale, producendo un rumore caratteristico, onde di pressione molto elevata e nuove bolle. Il liquido colpisce con violenza le pareti della condotta e genera aumenti localizzati di pressione e di temperatura, e le sollecitazioni derivanti creano sulla condotta erosioni delle superfici e/o rotture.

Nel caso in esame, questo restringimento è causato dalla presenza dei vitoni ed è quindi molto importante tenere conto di questo fenomeno per poter realizzare un modello che ben approssimi la realtà. La funzione Matlab creata, infatti, pone attenzione su questo aspetto, verificando, ad ogni passo, se la pressione minima che si manifesta all'interno del vitone, dove si creano le forti depressioni (e non quella all'ingresso delle linee), assume valori ammissibili, ossia superiori alla pressione del vapore del fluido considerato, detta anche *pressione di cavitazione* (qui considerata, per l'acqua, pari a -9.5Bar).

La pressione minima è la pressione che si ha subito dopo il vitone:

$$\frac{p_{h_{\min}}}{\gamma} = \frac{p_{h_{\text{mis}}}}{\gamma} + \frac{\left(\frac{Q_h}{A_t}\right)^2}{2g} - \frac{\left(\frac{Q_h}{A_{v_h}}\right)^2}{2g}. \quad (5.12)$$

Se il valore ottenuto è inferiore a quello della pressione di cavitazione, essa viene forzata ad assumere questo valore e viene alzata una *flag*, il cui ruolo verrà spiegato più avanti. Una volta eseguita questa operazione, viene calcolata la portata limite, ossia la massima portata che può essere ottenuta dalla linea considerata, nel caso in cui si sia raggiunta o superata la pressione di cavitazione. Rifacendosi alla (5.12) risulta dunque semplice ricavare

$$Q_{h_{\text{lim}}} = \sqrt{\frac{\left(p_{\text{cav}} - \frac{p_{h_{\text{mis}}}}{\gamma}\right)}{\left(\frac{1}{A_t^2} - \frac{1}{A_{v_h}^2}\right)}} \cdot 2g \quad (5.13)$$

Solo se la *flag* trattata sopra ha valore "1", e quindi viene oltrepassato il valore minimo ammissibile di pressione lungo la linea, la portata che viene restituita in uscita all'utente è quella calcolata in (5.13). Questa operazione viene effettuata, su entrambe le linee, dal blocco "Switch", cerchiato in rosso in Figura 5.2.

Si noti, infine, che il blocco integratore è dotato di un *external reset*, di tipo *falling*: il valore dell'integrale viene azzerato nel caso in cui il segnale di reset (in questo caso la *portata* della linea corrispondente) passa da un valore positivo a un valore nullo.

5.1.3 Analisi bontà del modello

Una volta ultimato il modello Matlab/Simulink, sono state effettuate delle simulazioni mirate atte a verificare la validità del sistema creato, apportando, eventualmente, delle modifiche ai vari parametri interessati, per avvicinarsi il più possibile alla realtà. E' stata infatti analizzata la caratteristica reale del vitone utilizzato e si è cercato di ottenere, in simulazione, una forma d'onda il più simile possibile a questa. A questo scopo, si è rivelato necessario analizzare i soli tratti di tubo relativi ai vitoni nelle due linee, azzerando le perdite di carico dovute al tubo e all'aeratore, e comandando l'apertura dei due vitoni uno al 10% (circa 10°) e l'altro al 100% (180°).

In Tabella 5.1 si riportano i valori di pressione in una scala da 0.5 a 5.0 Bar e le corrispondenti portate ottenute dal vitone aperto al massimo (180°), nel caso reale (in rosso) e nel caso simulato (in blu), ottenuto a partire dal modello creato.

Bar	0.5	1.0	1.5	2.0	2.5	3.0	3.5	4.0	4.5	5.0
L/min	20	24	28.5	33.5	37	41.5	43	47	49.5	52
L/min	-	23.3	28.7	33.2	37.2	40.8	43.4	45.8	48.1	50.3

Tabella 5.1: Confronto caratteristiche vitone reale (rosso) e suo modello (blu)

Come si evince dai dati riportati in Tabella 5.1, i risultati ottenuti sono più che soddisfacenti. Si dimostra un leggero discostamento del modello rispetto al caso reale solo alle pressioni più elevate.

In Figura 5.3 e Figura 5.4 si riportano i grafici utilizzati per la realizzazione della Tabella 5.1. A partire dal primo grafico si è ricavata una corrispondenza pressione/tempo (es.: il valore di pressione 1 Bar si ha all'istante di tempo 1.112 s) e, nel secondo grafico si è verificato, in corrispondenza degli istanti temporali evidenziati nel primo (associati ai desiderati valori di pressione), il relativo valore di portata ottenuto.

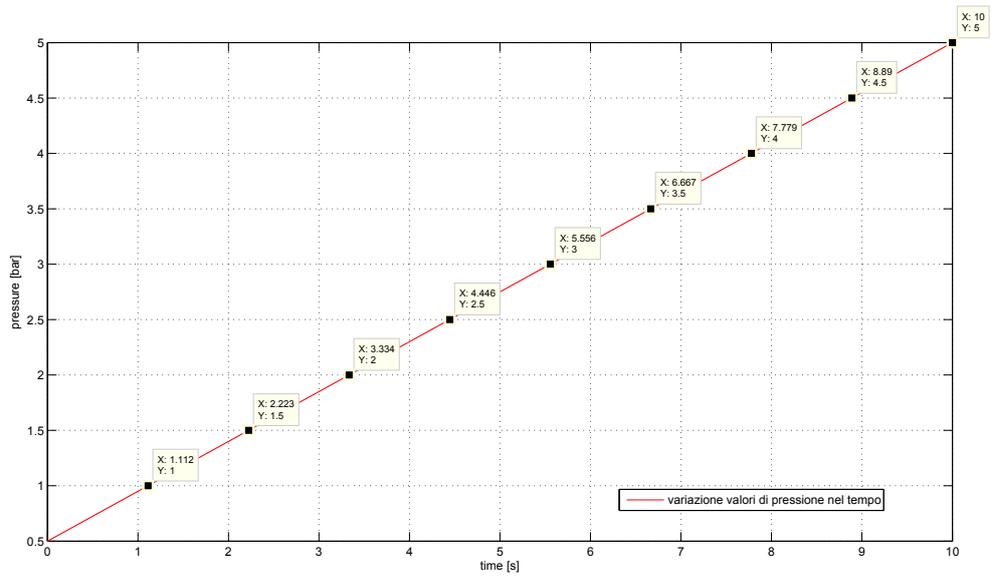


Figura 5.3: Variazione di pressione da 0.5 a 5.0 Bar in un tempo di 10 s

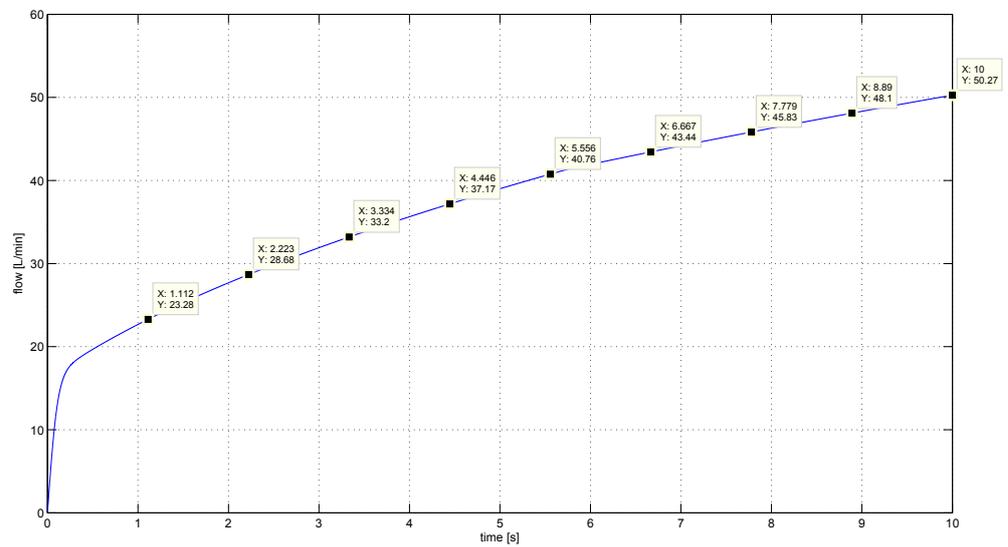


Figura 5.4: Diagramma della portata al variare della pressione da 0.5 a 5.0 Bar

5.2 MODELLO DELL'INTERO APPARATO

Il blocco di simulazione del case idraulico, realizzato seguendo la procedura vista nel precedente paragrafo, viene inserito all'interno di un modello che rappresenti l'intero sistema e che consideri anche i sensori, di temperatura e portata, e il blocco di controllo.

Il modello, implementato ancora una volta in ambiente Simulink, riproduce il più fedelmente possibile il sistema realizzato fisicamente nel primo prototipo del progetto in esame (controllato con Arduino), pur riportando la sostanziale modifica vista per l'unità idraulica (elettrovalvole sostituite da vitoni comandati da motori passo-passo).

In Figura 5.7 è riportato lo schema a blocchi realizzato in Simulink. Tuttavia, prima di analizzare il modello nel suo complesso e studiarne in dettaglio il suo funzionamento, si pone l'attenzione su dei particolari blocchi presenti all'interno dello schema:

- Sensori di temperatura, sono sostanzialmente dei sistemi del primo ordine $\frac{1}{1+s\tau}$, che introducono un ritardo $\tau = 5s$. In Figura 5.5 è riportato lo schema Simulink li riproduce. Nel caso in esame, i sensori ricevono in ingresso un valore costante di temperatura, fissato rispettivamente a $15^{\circ}C$ per il tubo dell'acqua fredda e a $60^{\circ}C$ per il tubo dell'acqua calda.

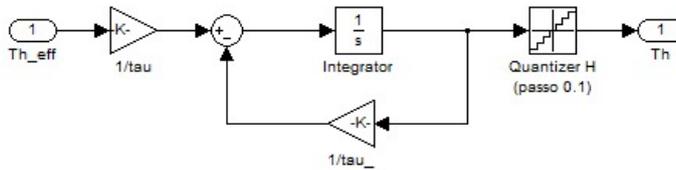


Figura 5.5: Modello Simulink per il sensore di temperatura

Da notare la presenza del blocco "Quantizer", di passo 0.1, che approssima l'incertezza che si ha sulla misura.

- Sensori di portata, sono qui trattati come semplici ritardi di 120 ms (per questo motivo viene utilizzato il blocco "Transport Delay" di valore 0.12). In Figura 5.6 è riportato il blocco corrispondente.

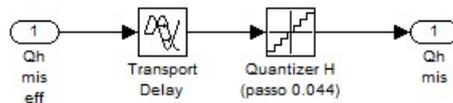


Figura 5.6: Modello Simulink per il sensore di portata

Anche in questo caso è stato introdotto il blocco "Quantizer" di passo 0.044, che indica l'incertezza che si ha sulla misura.

Questo valore è ottenuto a partire dal fatto che si sa di avere un'incertezza sulla frequenza di misura della portata di 3.33 Hz (1 impulso ogni 0.3 s) che, riportato in L/min diventa:

$$\frac{15\text{L/min}}{570\text{Hz}} \cdot 3.33\text{Hz} = 0.088\text{L/min},$$

valore che si dimezza (0.044) se si considerano sia i fronti di salita che di discesa dell'onda quadra che arriva al sensore di portata.

Si osservi dunque la Figura 5.7. Si nota come questa possa essere suddivisa in tre parti (come evidenziato in figura):

- Ingressi e loro elaborazione
- Controllo
- Blocco simulatore rubinetto

Procedendo da sinistra verso destra, si analizzano le varie fasi.

Ingressi: gli ingressi del modello sono portata totale e temperatura desiderati dall'utente e i dati derivanti dai sensori di temperatura. A partire da questi, il blocco "MATLAB Function" che esegue la funzione *flows.m* (cerchiato in azzurro in Figura 5.7), calcola le portate di riferimento corrette (tramite i calcoli visti nel paragrafo ..) desiderate per le due linee, da dover inseguire per soddisfare le richieste dell'utente (portata totale e temperatura dell'acqua in uscita dal rubinetto).

Controllo: la seconda parte dello schema si occupa del controllo, la quale viene curata essenzialmente dal blocco "MATLAB Function" che esegue la funzione *onecontrol.m* (in Figura 5.7 cerchiato in verde). Questa funzione ha il compito di calcolare, in base alle esigenze dell'utente, il grado di apertura delle valvole (vitoni) e quindi la quantità di acqua che dovrà scorrere nelle due linee al fine di ottenere la corretta portata totale e la corretta temperatura dell'acqua miscelata. I parametri ricevuti in ingresso sono:

- Portate di riferimento Qh_{ref} e Qc_{ref} calcolate dalla funzione *flows.m*
- Portate effettive ottenute in uscita al sistema Qh_{mis} e Qc_{mis}
- Temperatura desiderata dall'utente $Tset$
- Stima della temperatura effettivamente ottenuta in uscita $Tout$ (ottenuta con la (4.6)).

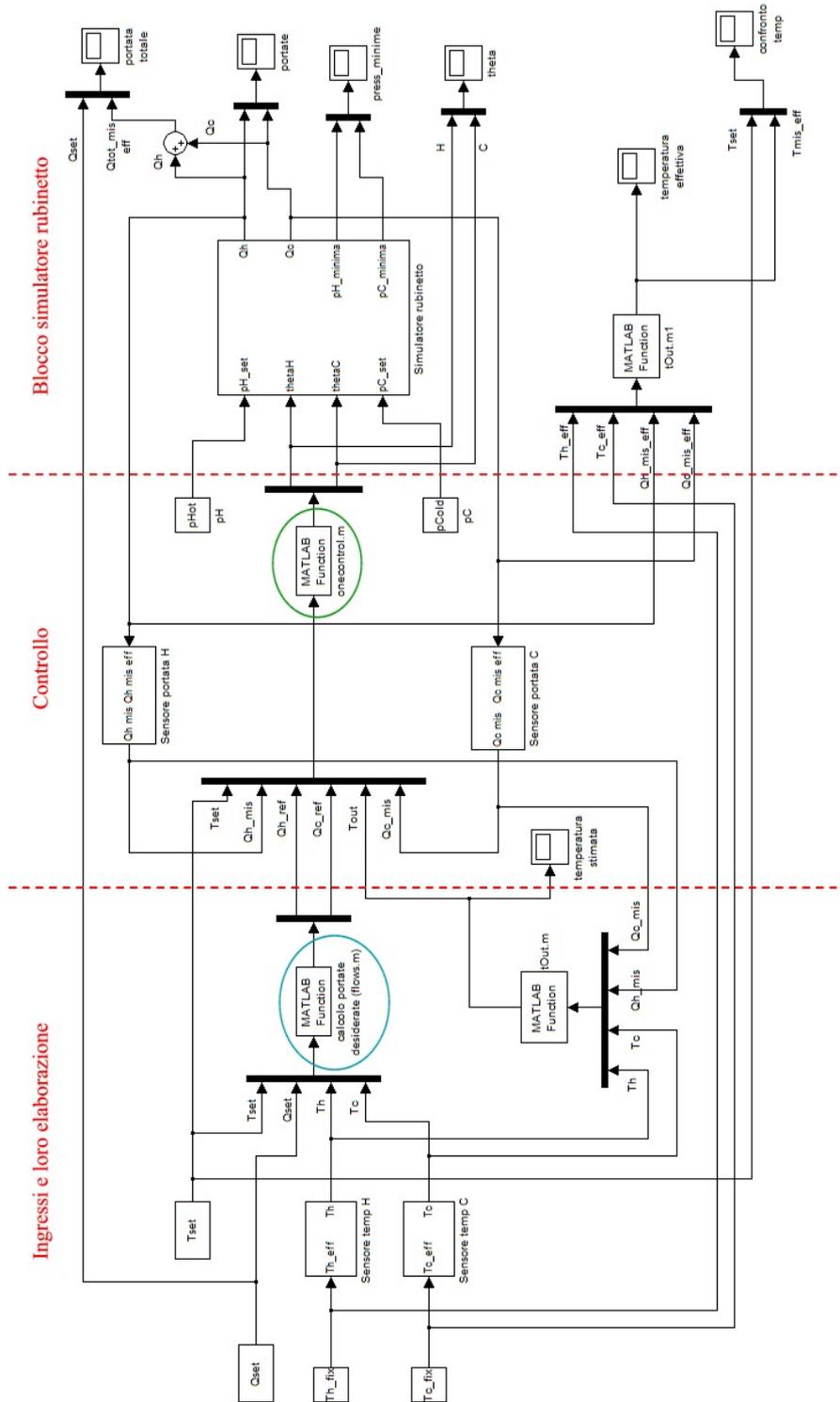


Figura 5.7: Schema Simulink relativo all'intero modello del sistema rubinetto

Mentre l'uscita è, per entrambe le linee, l'angolo θ , grado di rotazione che deve essere garantito dai motori passo-passo e quindi grado di apertura dei vitoni ad essi collegati in modo diretto (si avranno quindi θ_h e θ_c). La procedura utilizzata è del tutto analoga a quella vista per il primo prototipo (per questo si rimanda il lettore al Paragrafo 4.3).

Blocco simulatore rubinetto: La terza parte dello schema Simulink di Figura 5.7, infine, riguarda il blocco contenente il modello del rubinetto, dai vitoni fino all'aeratore, già analizzato nel precedente paragrafo. Questo blocco riceve in ingresso le pressioni a monte del rubinetto (pressioni dell'acqua proveniente dall'acquedotto) e gli angoli θ_h e θ_c che vanno a comandare i motori passo-passo e quindi l'apertura dei vitoni. In uscita dal blocco si hanno le portate ottenute dalle due linee che, ad ogni passo, vengono confrontate con quelle di riferimento. *Si noti che all'interno del blocco è stato introdotto anche un "Quantizer" di passo 0.9, che va a simulare la dinamica del motore passo-passo, che raggiunge la posizione di riferimento desiderata eseguendo passi di 0.9°.*

Il modello così realizzato è stato sottoposto a diverse simulazioni, analizzate in dettaglio nel Capitolo 6.

RISULTATI SPERIMENTALI

In questa sezione vengono illustrate le differenti modalità di test realizzate per il sistema rubinetto al fine di valutarne le prestazioni e quindi l'applicabilità. Per il primo e il secondo prototipo sono stati realizzati i seguenti tipi di test:

- *Variazione setpoint di portata*: al variare del setpoint di portata, si verifica se la temperatura dell'acqua in uscita rimane inalterata, pur rispettando il vincolo di portata totale fissato
- *Variazione setpoint di temperatura*: al variare del setpoint di temperatura, si verifica se la portata totale dell'acqua in uscita rimane inalterata pur inseguendo la temperatura desiderata dall'utente
- *Calo di pressione*: in presenza di un calo di pressione e quindi un calo di portata, dovuto, ad esempio, all'apertura di una fonte di acqua posta in prossimità di quella in uso, si deve assicurare un mantenimento della temperatura richiesta, controllando in modo il più pronto possibile le portate nelle due differenti linee, per garantire il massimo comfort all'utente
- *Strozzature acqua calda o fredda*: in caso di caduta di pressione istantanea lungo uno dei due tubi dell'acqua, si deve assicurare (secondo precise norme idrauliche) la chiusura dell'altra linea entro pochi secondi.
- *Richiesta di setpoint inammissibili di temperatura*: in questo test si considera il caso in cui il valore di setpoint di temperatura richiesto eccede i limiti disponibili: temperatura richiesta (T_{set}) maggiore del valore registrato dal sensore relativo alla linea calda o minore di quello registrato dal sensore relativo alla linea fredda.

6.1 TEST PRIMO PROTOTIPO

I valori di portata e temperatura relativi all'acqua calda e quella fredda possono essere monitorati in real-time, grazie alla creazione di un'interfaccia grafica realizzata con Labview, riportata in Figura 6.1. Questa interfaccia è stata utilizzata per eseguire i test sul primo prototipo di rubinetto, realizzato come riportato nel Capitolo 4.

Considerando il quadrante dei primi quattro grafici (Figura 6.1), nel primo grafico (in alto a sinistra) vengono riportate le temperature, in °C, percepite dai due sensori (acqua calda e acqua fredda), rappresentate rispettivamente dalla linea rossa e dalla linea bianca. La scala

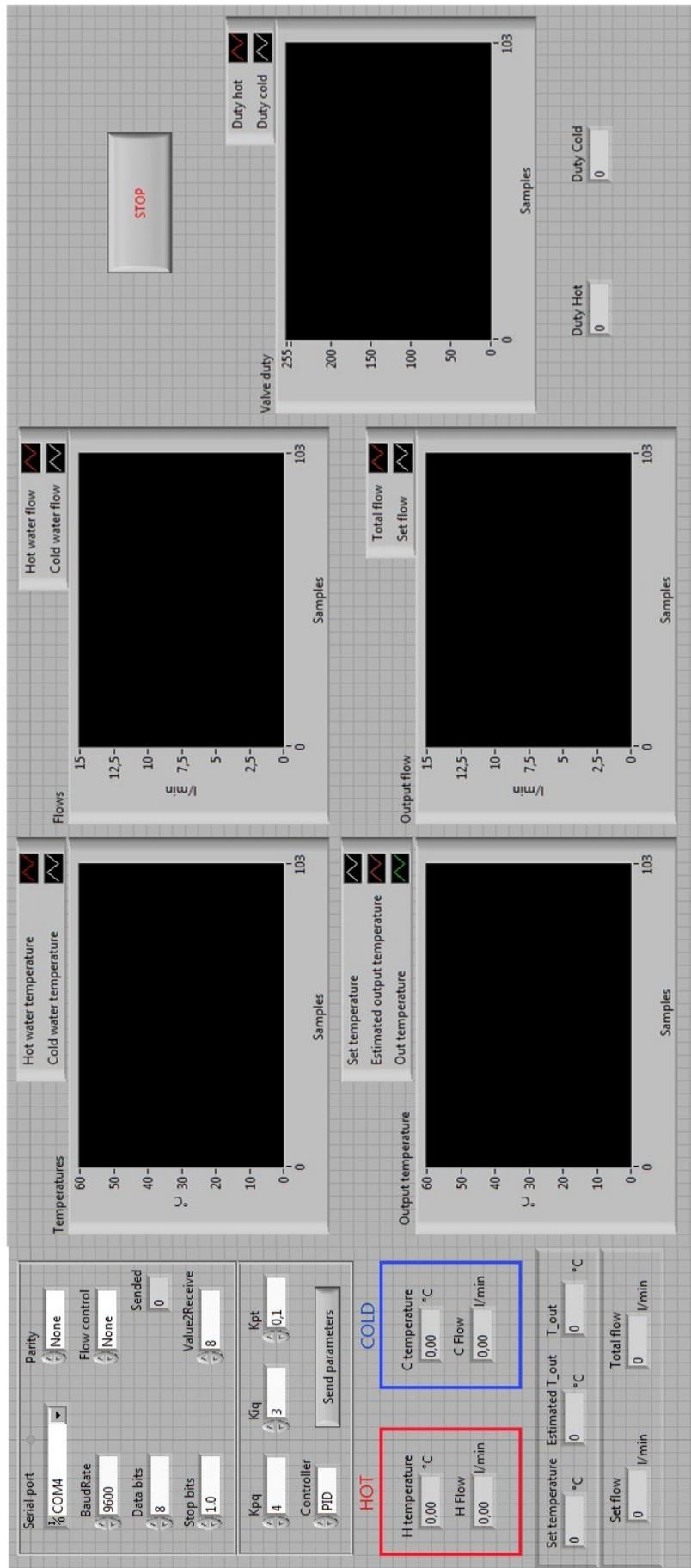


Figura 6.1: Interfaccia Labview utilizzata per i test relativi al primo prototipo

utilizzata analizza valori compresi tra 0 e 60°C, range di temperature compatibili con la situazione reale.

Il secondo grafico (in alto a destra nel quadrante) riporta, invece, l'andamento della portata delle valvole. Il comportamento della curva è dato dalla variazione della frequenza dell'onda quadra in ingresso alla valvola e, di conseguenza, della sua portata (maggiore o minore apertura). La scala utilizzata prevede un range che va dagli 0 L/min ai 15 L/min, portata nominale, ottenuta alla massima frequenza possibile, 570 Hz.

Il terzo e il quarto grafico monitorano le uscite relative alla temperatura e alla portata. Nel terzo grafico (in basso a sinistra), infatti, si riportano gli andamenti della temperatura stimata (ottenuta dalla media pesata con le portate), quella desiderata dall'utente (T_{set}) e quella in uscita dal rubinetto (nel caso si inserisca un terzo sensore di temperatura, in aggiunta a quelli posti lungo le due linee di acqua calda e fredda). Nel quarto e ultimo grafico del quadrante (in basso a destra) si riportano il flusso totale ottenuto dalle valvole (somma portate relative all'acqua calda e fredda, riportate nel secondo grafico) e il flusso totale desiderato e impostato dall'utente (Q_{set}).

Nel restante grafico vengono riportati i *duty cycle* relativi alle due valvole, ossia i comandi di apertura/chiusura che vengono ad esse inviati (riportati anche in forma numerica sotto al grafico).

Nei riquadri presenti in sinistra, infine, si hanno i valori numerici dei parametri analizzati (riquadri rosso, HOT, e blu, COLD), le cui variazioni sono percepibili sempre in real-time. Mentre nei due sottostanti si riportano i vari parametri relativi a temperatura e portata considerati nei grafici analizzati.

Nota: Si ricordi che T_{set} e Q_{set} vengono ricevuti come ingressi analogici, variati con potenziometri sul case interfaccia, visto nel dettaglio nel paragrafo 4.1.1.

Prima di procedere ai vari test, è stata individuata la corretta combinazione dei parametri relativi al controllo PI, visti nel paragrafo 4.3:

Guadagno proporzionale portata	kp_q	2.5
Guadagno integrale portata	ki_q	2.5
Guadagno proporzionale/integrale temperatura	k_t	0.03

Tabella 6.1: Parametri PI scelti

6.1.1 Variazione setpoint di temperatura

Al variare del setpoint di temperatura, il sistema di controllo realizzato deve assicurarne il corretto inseguimento e, contemporaneamente,

garantire il mantenimento della portata totale richiesta. Nei grafici riportati in Figura 6.2 e Figura 6.3, si analizzano, rispettivamente, gli andamenti relativi alla temperatura (quella impostata dall'utente e quella stimata ottenuta dalla media pesata con le portate) e quelli relativi alle portate nelle due linee e il flusso totale ottenuto.

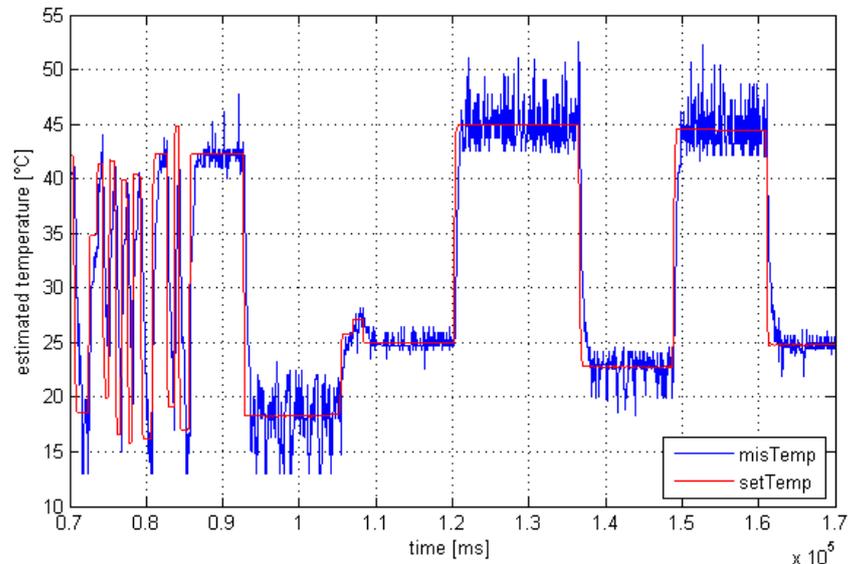


Figura 6.2: Variazione setpoint di temperatura con relativo inseguimento da parte della temperatura stimata

Osservazioni: Si nota un buon inseguimento del setpoint di temperatura, anche a fronte di variazioni repentine del riferimento. La prontezza ottenuta è dovuta all'efficienza dei sensori di portata, dai quali derivano i dati utilizzati per la stima della temperatura in uscita (linea blu in Figura 6.2). Si nota, tuttavia, un andamento oscillatorio in corrispondenza a valori abbastanza elevati o abbastanza ridotti di temperatura, dovuto all'instabilità registrata dai sensori di portata in quegli istanti temporali. Osservando Figura 6.3, infatti, si vede che le oscillazioni si presentano in corrispondenza a questi intervalli e sono causate dal fatto che una delle due portate risulta molto ridotta (inferiore a 1 L/min). Come visto nel paragrafo 2.2, il range di lavoro ottimale per i sensori utilizzati è compreso tra 1.1 e 15 L/min: al di fuori di questi valori le prestazioni risultano sicuramente meno efficienti, restituendo misure sfalsate. Tuttavia, le oscillazioni, pur essendo di modesta ampiezza, risultano molto frequenti e attorno al riferimento richiesto: sotto queste condizioni, la mano umana non è in grado di percepire variazioni importanti, e quindi il risultato ottenuto è da considerarsi molto positivo. La portata totale messa a disposizione, inoltre, rimane sempre molto prossima ai 4.1 L/min, valore di Q_{set} desiderato.

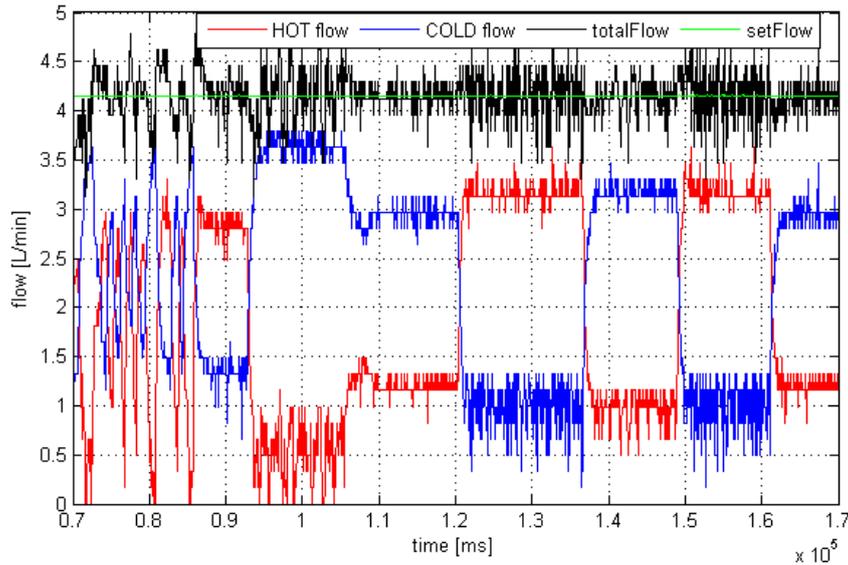


Figura 6.3: Comportamento portate nelle due linee in corrispondenza di variazione setpoint temperatura e monitoraggio del relativo flusso totale ottenuto

6.1.2 Variazione setpoint di portata

Al variare del setpoint di portata, il sistema di controllo realizzato deve assicurarne il corretto inseguimento, modificando dinamicamente le portate nelle due linee al fine di garantire, oltre alla desiderata portata totale, il mantenimento della temperatura in uscita richiesta. Nelle Figure 6.4 e 6.5 sono riportati gli andamenti della portata e della temperatura reali posti a confronto con quelli fissati dall'utente.

Osservazioni: Si osserva un buon comportamento del sistema per quanto riguarda l'inseguimento del setpoint di portata: i tempi di salita e di discesa, a fronte di variazioni repentine, sono sempre inferiori al mezzo secondo. Tuttavia, se si osserva il comportamento della temperatura stimata (Figura 6.5), si riscontrano importanti oscillazioni, di ampiezza di anche 6-7 °C, dovute, ancora una volta, all'imprecisione dei sensori di portata per valori inferiori a 1.1 L/min. In Figura 6.6, si può verificare, infatti, che, in corrispondenza agli intervalli temporali in cui si presenta questo fenomeno, si ha la portata fredda che raggiunge valori inferiori a 1 L/min. Questo risultato, tuttavia, non deve spaventare, perché quella riportata è una stima (anche troppo accurata) della temperatura in uscita, e le oscillazioni, pur essendo di ampiezza elevata, sono molto frequenti. Questo fatto garantisce un valore medio in uscita molto prossimo a quello impostato (35°C): sperimentalmente è possibile verificare che, in realtà, la temperatura dell'acqua in uscita percepita dalla mano umana rimane pressoché costante attorno al valore fissato.

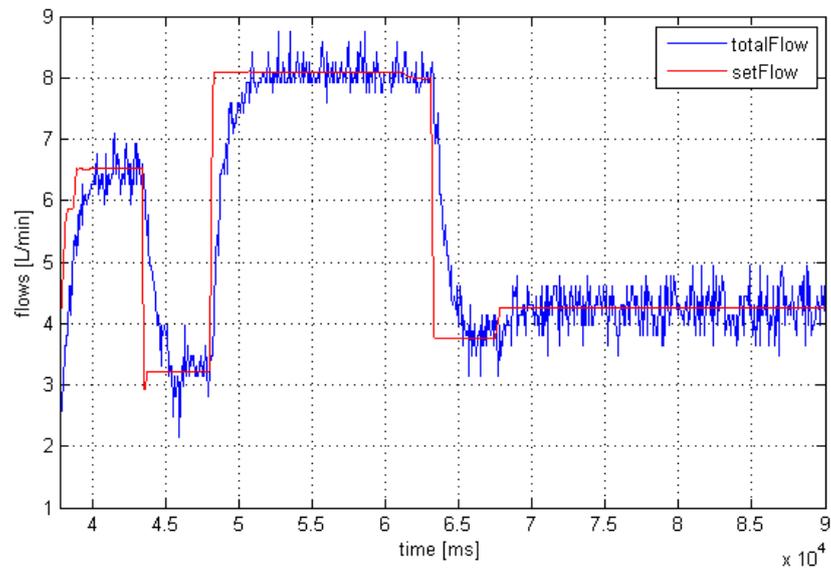


Figura 6.4: Variazione setpoint di portata con relativo inseguimento da parte della portata totale realmente ottenuta

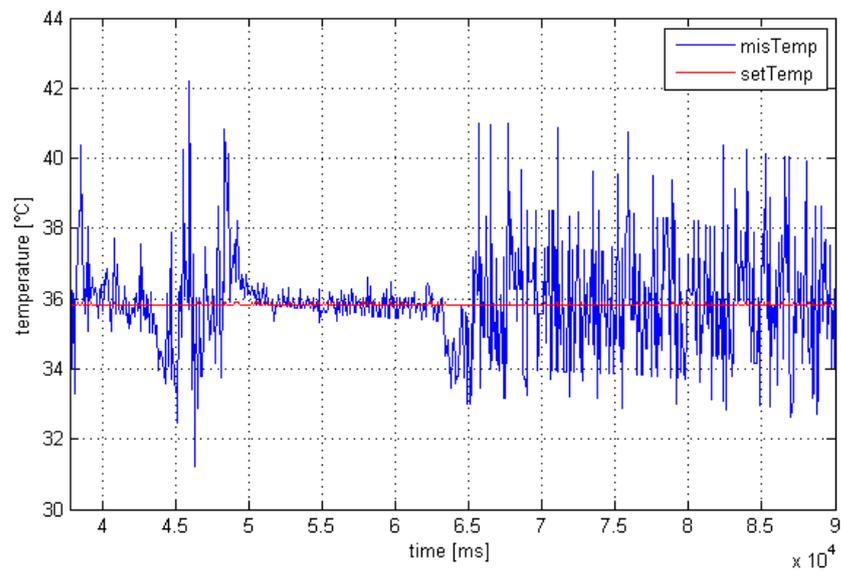


Figura 6.5: Andamento della temperatura in uscita stimata, al variare del setpoint di portata

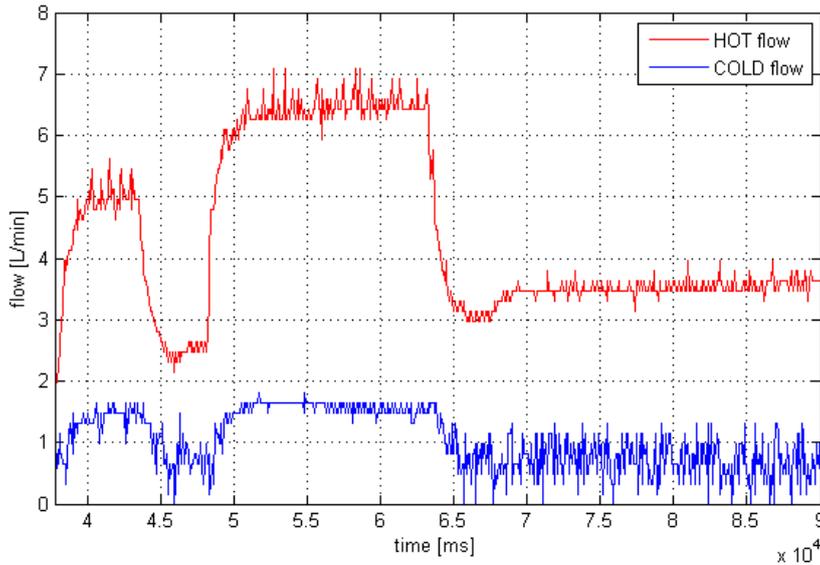


Figura 6.6: Andamento delle portate nelle due linee (calda e fredda), a fronte della variazione del setpoint di portata, ottenuta come riportato in Figura 6.4

6.1.3 Strozzature acqua calda o fredda

Per questioni di sicurezza, è necessario controllare i tempi di risposta del sistema rubinetto in presenza di una caduta istantanea di pressione dell'acqua calda o fredda. Secondo la norma EN 1111:2000, infatti, se si verifica una rottura, un guasto o un qualsiasi evento che causi un'interruzione dell'erogazione dell'acqua lungo una delle due linee, si deve assicurare, entro tempi molto ristretti, il blocco della linea opposta, onde evitare danneggiamenti all'utente che sta utilizzando il dispositivo.

In Figura 6.7 e Figura 6.8 si riportano i comportamenti ottenuti nelle due linee del sistema realizzato a fronte di una chiusura immediata di una delle due portate (rispettivamente calda e fredda).

Osservazioni: Come si evince dai grafici, il sistema è molto pronto, soprattutto nel caso di strozzatura di acqua calda. In corrispondenza dell'interruzione di questa portata, infatti, in 1 secondo si interrompe anche l'altra. Osservando, invece, Figura 6.8, si nota una prestazione leggermente meno efficiente: alla chiusura dell'acqua fredda, il sistema impiega circa 4 secondi a chiudere totalmente anche la calda. Questo è dovuto, probabilmente al fatto che non vi è molta differenza tra i valori di temperatura dell'acqua calda e quella fredda, registrate dai due relativi sensori (acqua calda del boiler in esaurimento).

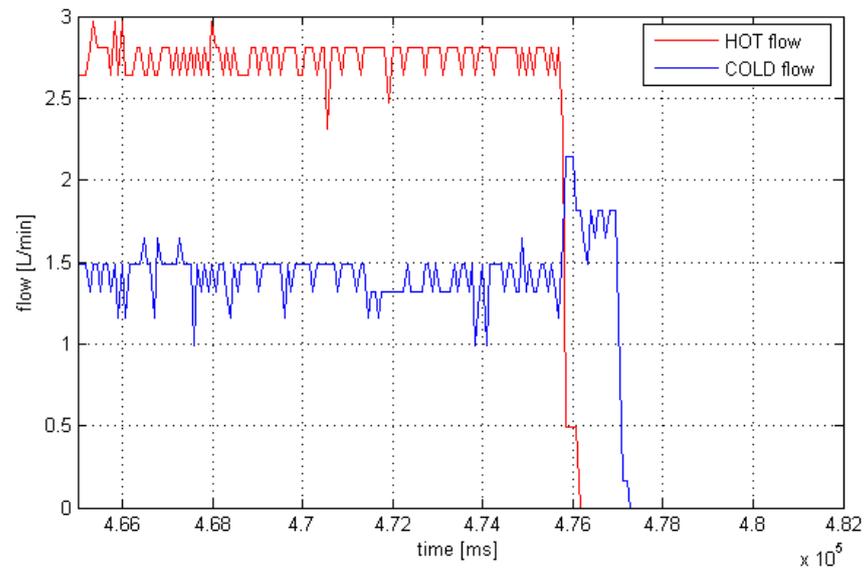


Figura 6.7: Andamento delle portate nelle due linee (calda e fredda), a fronte di una caduta di pressione istantanea dell'acqua calda

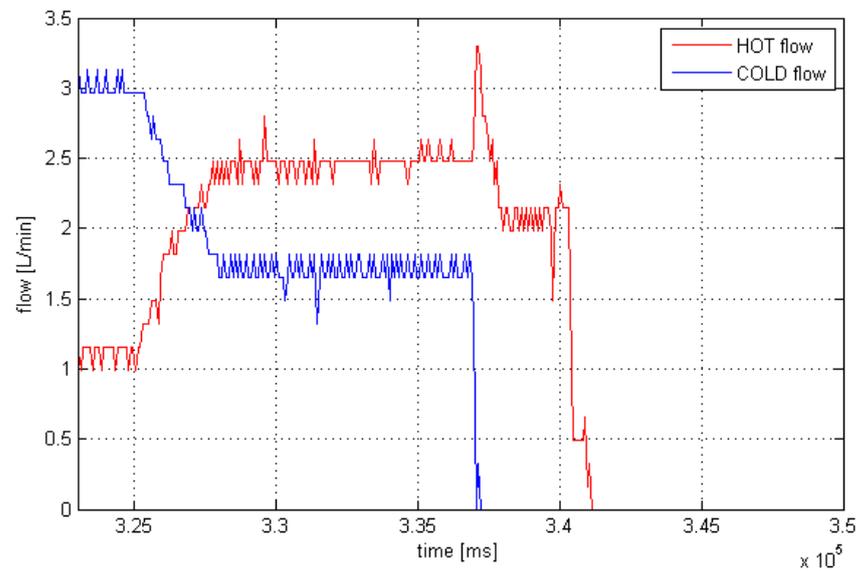


Figura 6.8: Andamento delle portate nelle due linee (calda e fredda), a fronte di una caduta di pressione istantanea dell'acqua fredda

6.1.4 Calo di pressione

In questo tipo di test si verifica la prontezza di risposta del sistema a fronte di destabilizzazioni della portata dovute all'apertura o alla chiusura improvvisa di una fonte di acqua posta in prossimità di quella utilizzata. In Figura 6.9 si riporta il caso in cui è aperta la sola linea calda (è richiesta una temperatura pari o superiore a quella registrata dal sensore in questa linea) e viene aperta/chiusa/aperta acqua calda di un rubinetto adiacente.

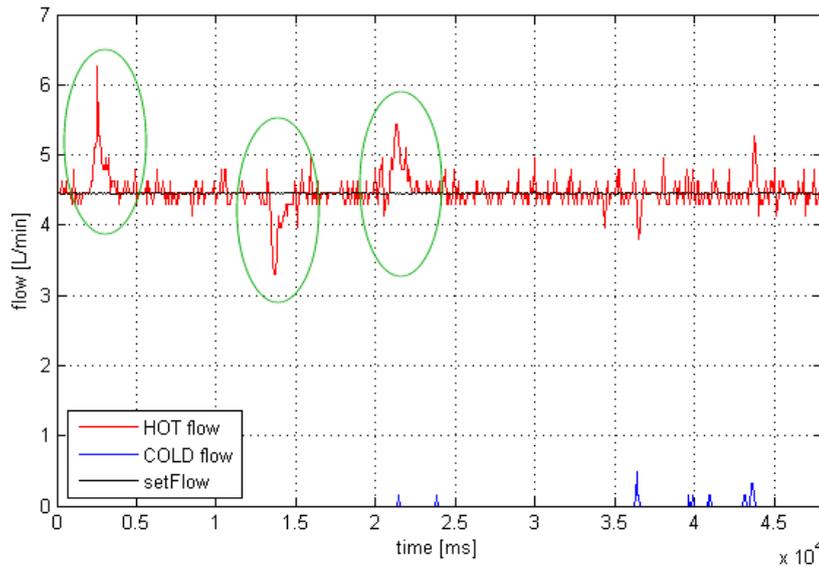


Figura 6.9: Comportamento della portata calda in presenza di sbalzi di pressione dovuti all'apertura o chiusura di acqua calda in una fonte posta in prossimità a quella utilizzata

Osservazioni: Negli istanti temporali in cui vengono eseguite le operazioni di apertura/chiusura/apertura, si manifestano dei picchi nella portata (cerchiati in verde in Figura 6.9). Questi picchi sono rivolti verso l'alto in corrispondenza dell'apertura, dal momento che questa operazione comporta una riduzione di pressione lungo la linea e una conseguente richiesta di maggior quantità di acqua. Analogamente, i picchi verso il basso si presentano quando si ha la chiusura. Tuttavia l'efficiente controllo sulla portata la riporta, in tutti i casi, entro i valori desiderati in circa 1 secondo. Si è verificato sperimentalmente che la mano umana praticamente non percepisce queste variazioni.

6.1.5 Richiesta di setpoint inammissibili di temperatura

In caso di richiesta di acqua a temperature fuori dai range ammissibili, ossia temperature inferiori a quella monitorata dal sensore relativo alla linea fredda o superiori a quella monitorata lungo la linea calda,

si assicura la disponibilità limitata ai valori minimi/massimi possibili. In Figura 6.10 si riportano due esempi.

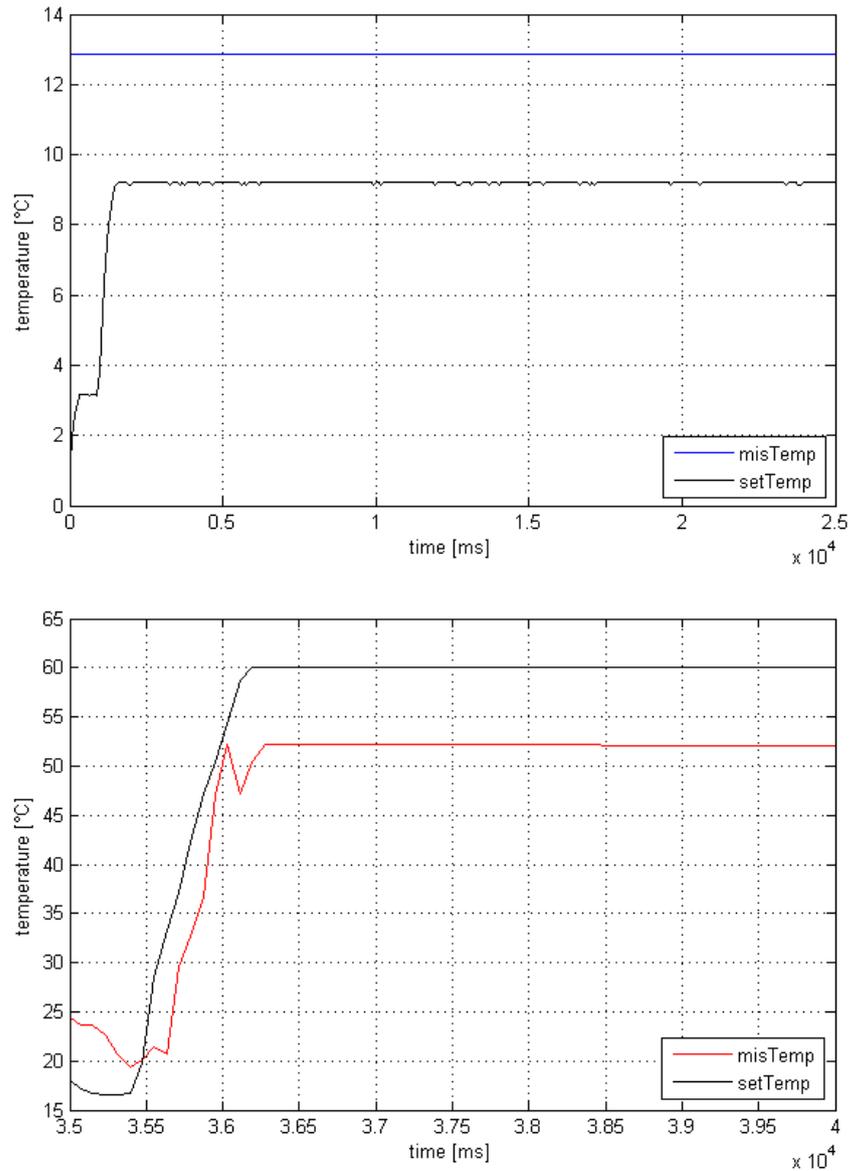


Figura 6.10: Andamento della temperatura in corrispondenza di richiesta di acqua a temperatura inferiore/superiore al limite minimo/massimo possibile (T_c/T_h)

6.2 TEST SECONDO PROTOTIPO

Il secondo prototipo, come visto nel Capitolo 5, è stato modellizzato in ambiente Matlab/Simulink, senza esser realizzato fisicamente. Per questo motivo i test effettuati sono in realtà delle simulazioni, atte a valutare le prestazioni del modello creato, sotto particolari condizioni

lavorative. Le prove effettuate, come si vedrà di seguito, sono molto simili a quelle già viste per il primo prototipo.

6.2.1 Variazione setpoint di temperatura

In questa sezione viene analizzato il comportamento del modello al variare del setpoint di temperatura impostato dall'utente che sta utilizzando il dispositivo. Le prove sono state effettuate con due diversi valori di Q_{set} (portata totale desiderata) per analizzare eventuali diversità di reazione del modello nei due casi.

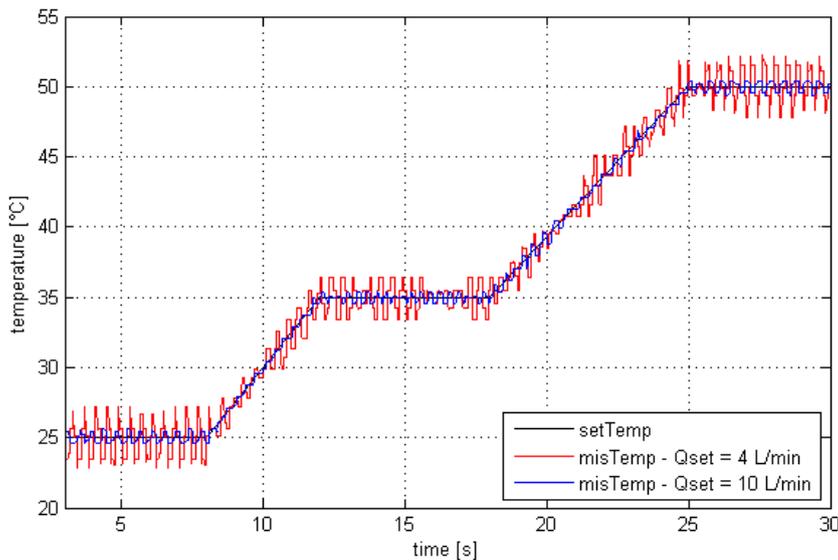


Figura 6.11: Andamento della temperatura in uscita stimata, al variare del setpoint di temperatura, con $Q_{set} = 4, 10$ L/min

Osservazioni: Come si evince dal grafico di Figura 6.11, il sistema è molto pronto e insegue in modo più che soddisfacente il riferimento imposto. Tuttavia si può notare una diversità di comportamento al variare del riferimento di portata totale (4 o 10 L/min). Si vede infatti che quando la portata totale richiesta è più ridotta (4 L/min) si hanno delle oscillazioni, di ampiezza anche di qualche grado, attorno al valore medio desiderato. Questo fatto è dovuto, come visto per il primo prototipo, dal range di efficienza garantito dai sensori di portata: a portate inferiori a 1 L/min si ha instabilità nella misura (si veda il grafico in alto di Figura 6.12), che si ripercuote quindi anche nella stima della temperatura, basata su di essa. Nel caso di $Q_{set} = 10$ L/min, invece, come si evince dal grafico in basso di Figura 6.12, i comportamenti sono molto più lineari e stabili.

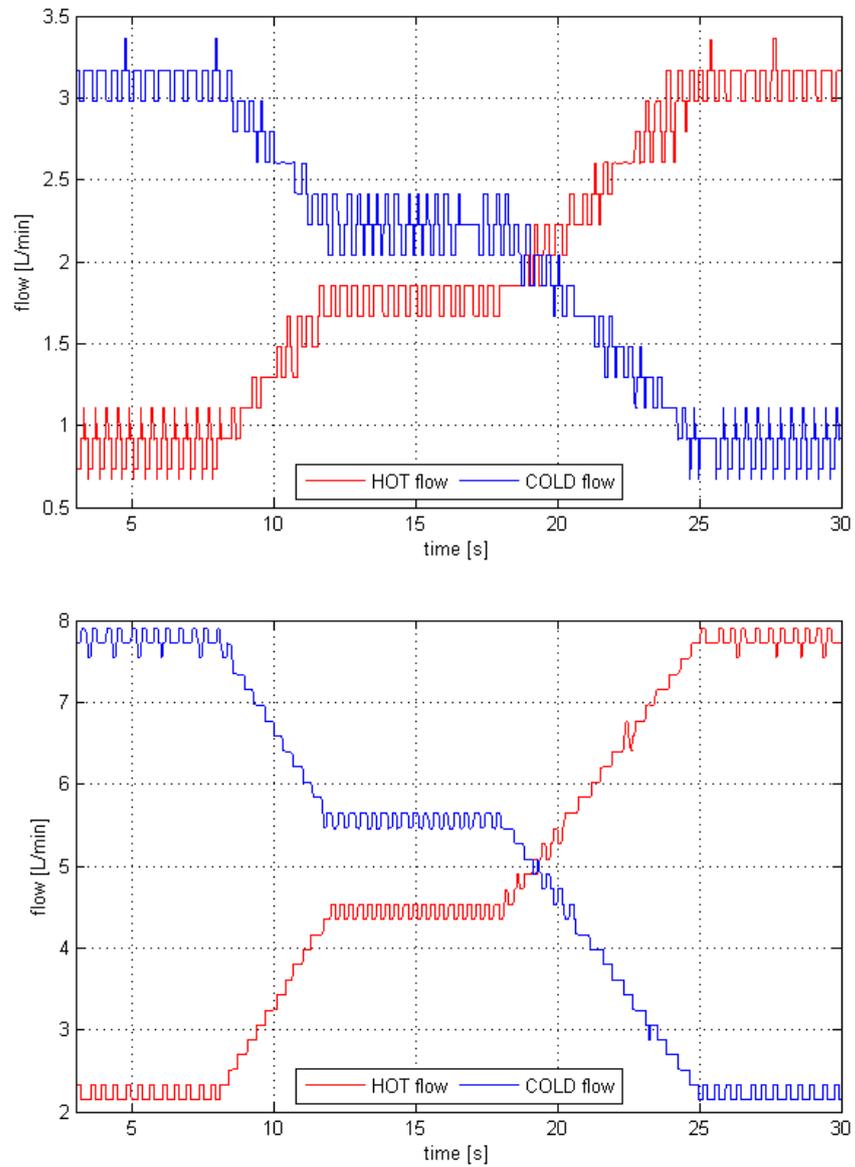


Figura 6.12: Comportamento portate nelle due linee in corrispondenza di variazione setpoint temperatura, ottenuto, rispettivamente, con $Q_{set} = 4, 10$ L/min

6.2.2 *Variazione setpoint di portata*

Al variare del setpoint di portata ci si aspetta che il sistema inseguia prontamente il riferimento desiderato, in modo tale da garantire il mantenimento del corretto valore di temperatura richiesto. In Figura 6.13 e Figura 6.14 sono riportati, rispettivamente, i grafici relativi alle variazioni delle portate delle linee calda e fredda e totale, al variare del setpoint di portata totale desiderato, e alla temperatura stimata in uscita.

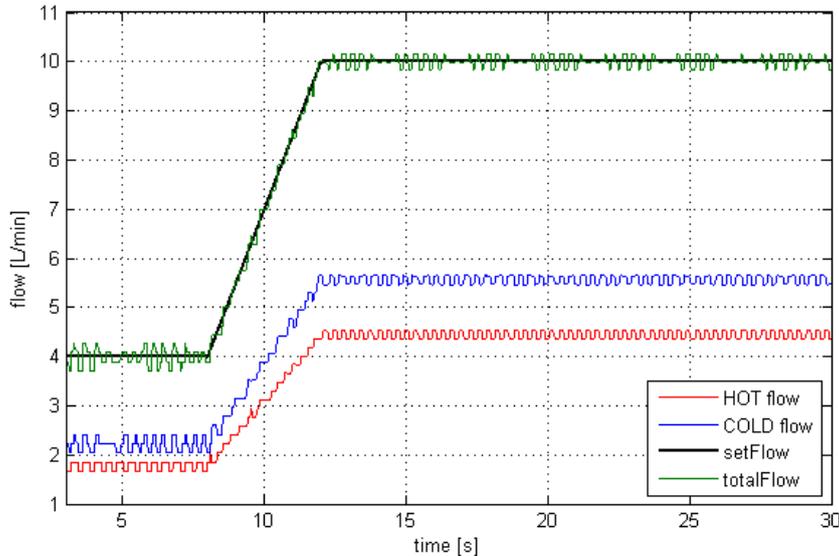


Figura 6.13: Comportamento delle portate nelle due linee e totale in corrispondenza di variazione del setpoint di portata da $Q_{set} = 4$ L/min a $Q_{set} = 10$ L/min

Osservazioni: Il sistema risulta essere molto pronto per quanto riguarda l'inseguimento delle portate. Osservando l'andamento della temperatura stimata (Figura 6.14), invece, si nota un comportamento migliore in corrispondenza all'aumento del setpoint di portata: fatto atteso, considerando il range di efficienza dei sensori di portata, visto anche nei precedenti test. In presenza di portate ridotte (Figura 6.13) si rileva qualche oscillazione di ampiezza un po' più elevata, che si ripercuote anche nella stima della temperatura. Per valori di Q_{set} più elevati (attorno ai 10 L/min) si notano ancora delle oscillazioni attorno al riferimento di temperatura imposto, ma queste sono talmente frequenti e di ampiezza ridotta (massimo 1 °C) che la mano umana non è in grado di percepirle.

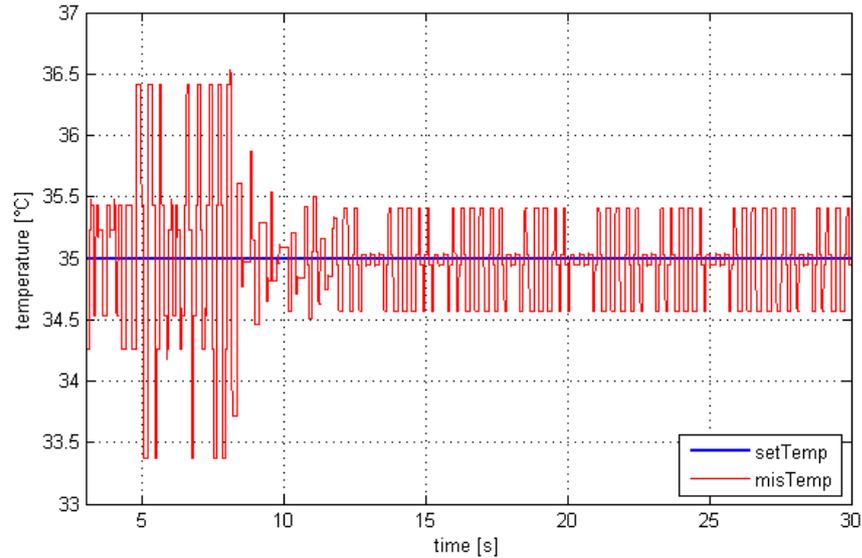


Figura 6.14: Andamento della temperatura in uscita stimata, al variare del setpoint di portata da $Q_{set} = 4$ L/min a $Q_{set} = 10$ L/min

6.2.3 Calo di pressione

Il modello realizzato per il sistema rubinetto dotato di vitoni ceramici è stato poi sottoposto al test relativo al calo di pressione. Per rendere possibile questa operazione è stato fatto variare il riferimento di pressione (lungo la linea calda e lungo quella fredda, in due differenti test), utilizzando come ingresso (p_{Set} nel modello Simulink di Figura 5.7) il segnale riportato in Figura 6.15, che prevede un calo di pressione da 3 a 1 bar in 7 secondi.

Di seguito si analizza, a titolo esemplificativo, il caso in cui si ha

- Calo di pressione lungo la linea fredda (da 3 a 1 bar)
- $T_{set} = 35$ °C
- $Q_{set} = 10$ L/min

Si osservino le Figure 6.16 e 6.17 per analizzare le reazioni del sistema a questo evento.

Osservazioni: Nell'intervallo compreso tra i 5 e i 12 secondi, in corrispondenza del quale si ha la diminuzione lineare della pressione da 3 a 1 bar, si nota, sia nell'andamento della temperatura (Figura 6.16) sia in quello della portata totale (Figura 6.17), una destabilizzazione. Tuttavia, osservando attentamente i due grafici si rileva che le variazioni sono caratterizzate da oscillazioni molto frequenti e di ampiezza ridotta, che si discostano di poco rispetto al valore di riferimento desiderato. Destabilizzazioni di tale entità risultano praticamente im-

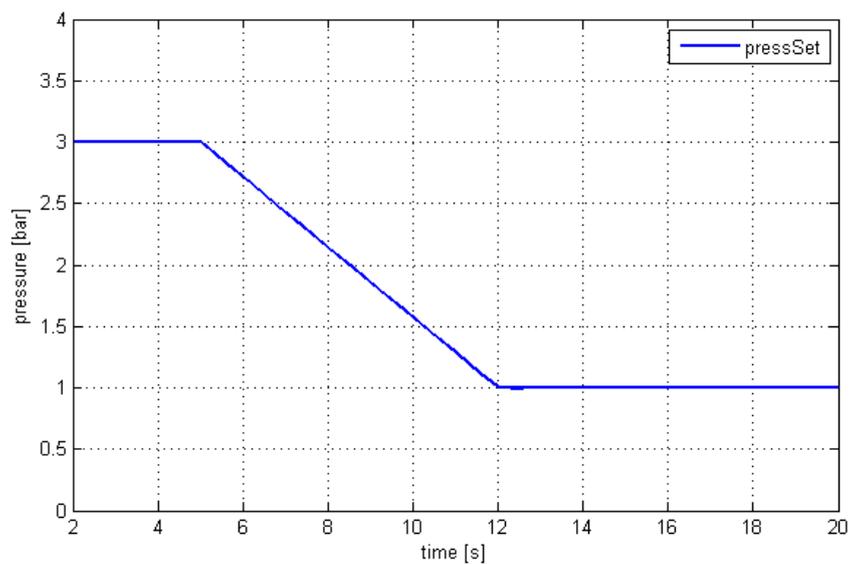


Figura 6.15: Variazione setpoint di pressione da 3 a 1 bar

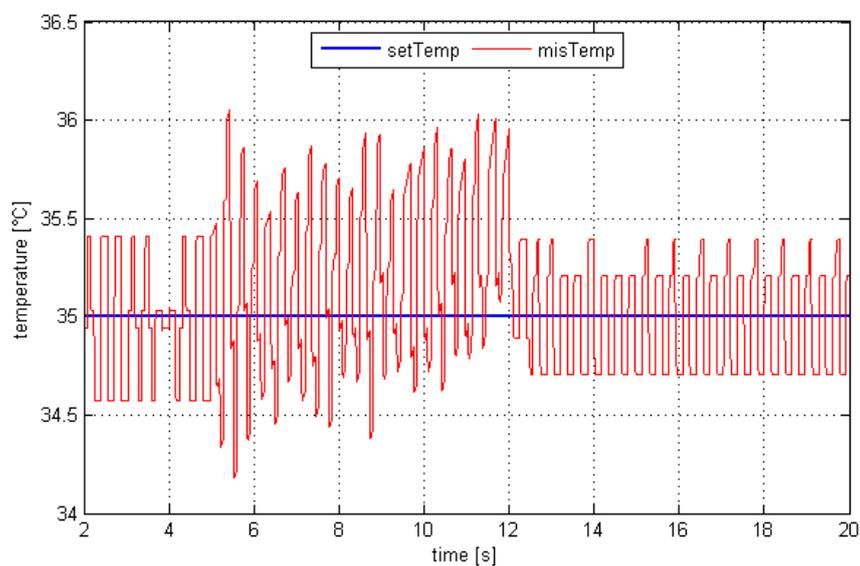


Figura 6.16: Andamento della temperatura stimata in presenza di uno sbalzo di pressione del tipo di Figura 6.15

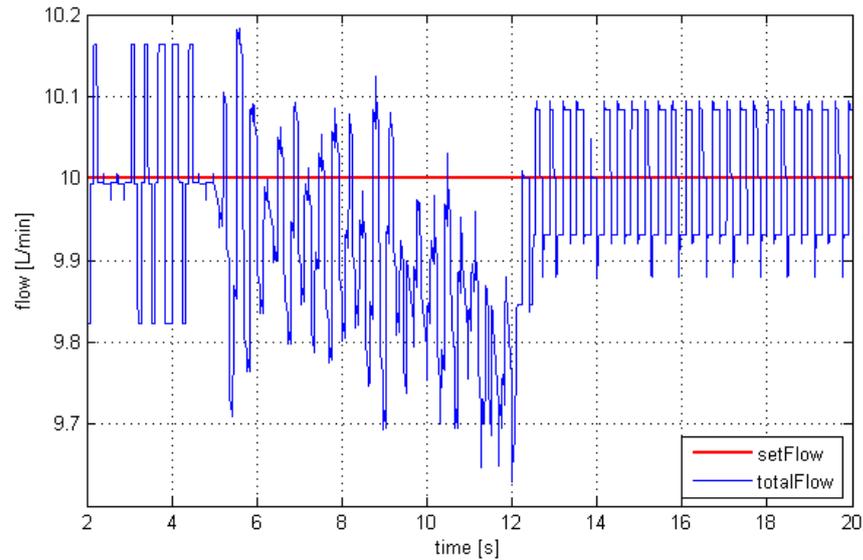


Figura 6.17: Comportamento della portata totale in presenza di uno sbalzo di pressione del tipo di Figura 6.15

perceptibili alla mano umana. Il risultato ottenuto è da considerarsi perciò molto soddisfacente.

6.2.4 Strozature acqua calda o fredda

L'ultimo test effettuato sul modello è quello relativo alla strozzatura di una delle due linee, dovuta alla mancata alimentazione di acqua calda o acqua fredda. Nei grafici di Figura 6.18 e Figura 6.19 si riportano gli andamenti delle due portate (calda e fredda) in caso di caduta di pressione istantanea (all'interno del modello Simulink di Figura 5.7 viene fornita come pressione di riferimento al blocco simulatore del rubinetto, per la linea calda o quella fredda, un segnale simile a quello di Figura 6.15, con la differenza che la pressione finale risulta pari a 0 bar: condotta chiusa).

Osservazioni: I comportamenti ottenuti dal sistema in caso di chiusura istantanea di acqua calda o acqua fredda sono analoghi. In entrambe le situazioni, infatti, i tempi di risposta sono molto ridotti: all'interruzione di una delle due portate, l'altra viene chiusa in meno di mezzo secondo. I risultati ottenuti dal modello realizzato per il secondo prototipo sembrano essere nettamente migliori rispetto a quelli visti in precedenza con il primo prototipo (paragrafo 6.1.3).

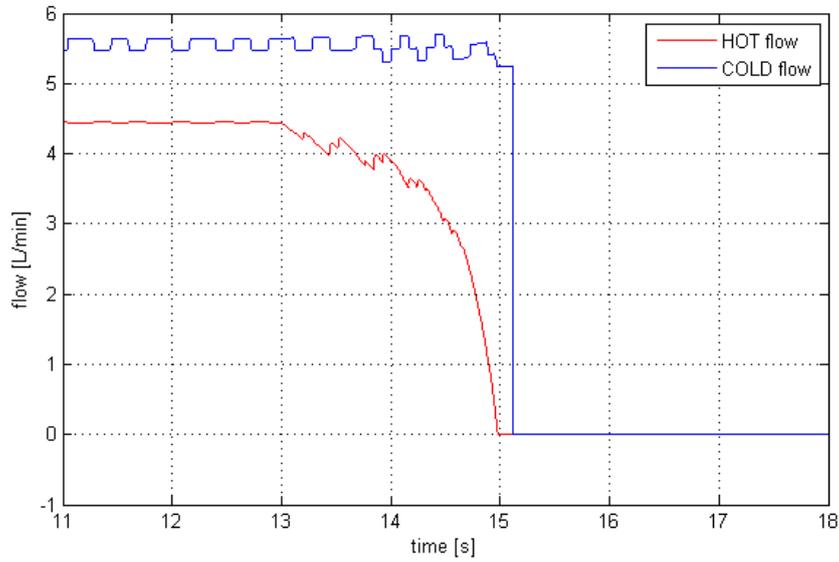


Figura 6.18: Andamento delle portate nelle due linee (calda e fredda), a fronte di una caduta di pressione istantanea dell'acqua calda

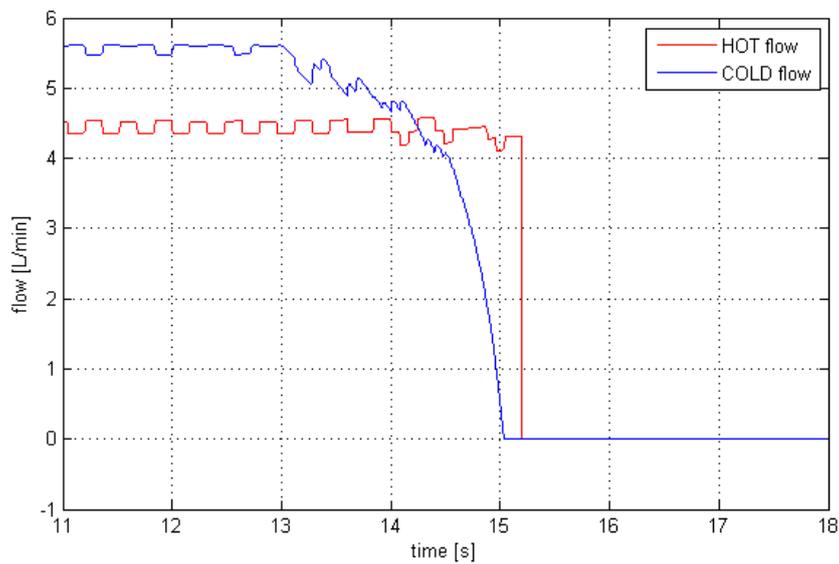


Figura 6.19: Andamento delle portate nelle due linee (calda e fredda), a fronte di una caduta di pressione istantanea dell'acqua fredda

6.3 CONFRONTO PRESTAZIONI

Nella presente sezione si vogliono confrontare le prestazioni ottenute con i due prototipi (quello fisicamente realizzato e quello modellizzato in ambiente Simulink) con lo scopo di valutarne l'efficienza e la realizzabilità del secondo. L'idea alla base di questi test è, infatti, quella di promuovere l'implementazione fisica del modello simulato per ottenere, con prestazioni pari o superiori, una netta riduzione di costi e dimensioni (vitoni comandati da motori passo-passo molto più economici e maneggevoli delle elettrovalvole proporzionali).

Le prove più significative da effettuare a questo scopo si sono rivelate quelle relative alle variazioni di temperatura e di portata: applicando in ingresso ai due sistemi lo stesso segnale si analizzano le loro differenti risposte.

6.3.1 *Variazione setpoint di temperatura*

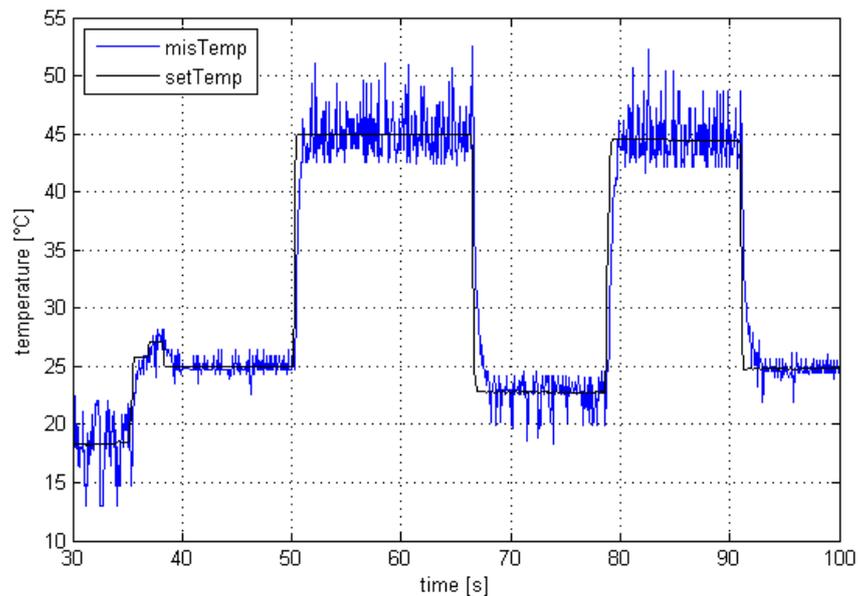


Figura 6.20: Andamento della temperatura in uscita stimata, al variare del setpoint di temperatura, nel primo prototipo

Osservazioni: Come si evince dalle Figure 6.23 e 6.24 il secondo prototipo sembra comportarsi in maniera nettamente migliore. Questo infatti presenta una risposta più pronta e con oscillazioni ridotte al minimo. In Figura 6.22 si analizza un confronto più dettagliato delle risposte (sovrapposte) dei due prototipi, ottenuto eseguendo uno "zoom" dei grafici precedenti (esattamente nell'intervallo di tempo compreso tra i 70 e i 90 s) per evidenziare ulteriormente le considerazioni fatte.

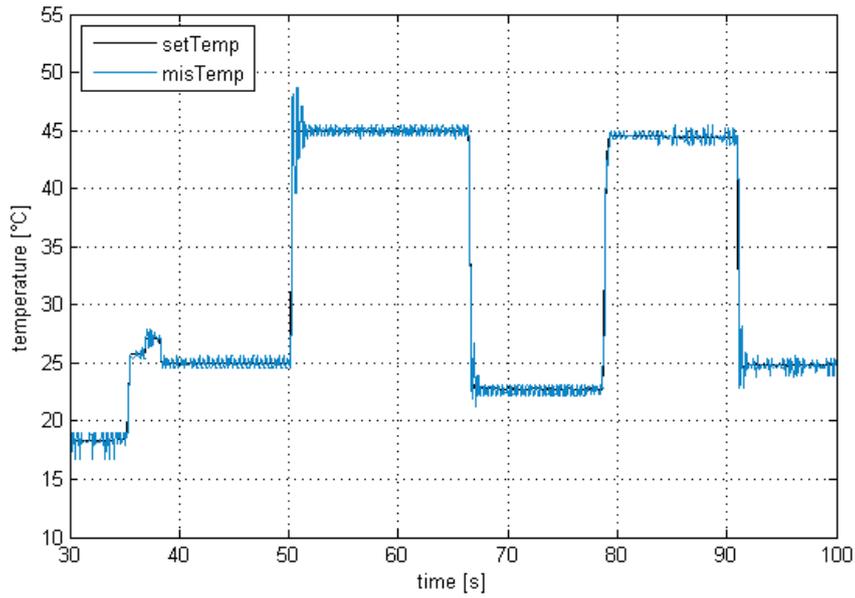


Figura 6.21: Andamento della temperatura in uscita stimata, al variare del setpoint di temperatura, nel secondo prototipo (modello)

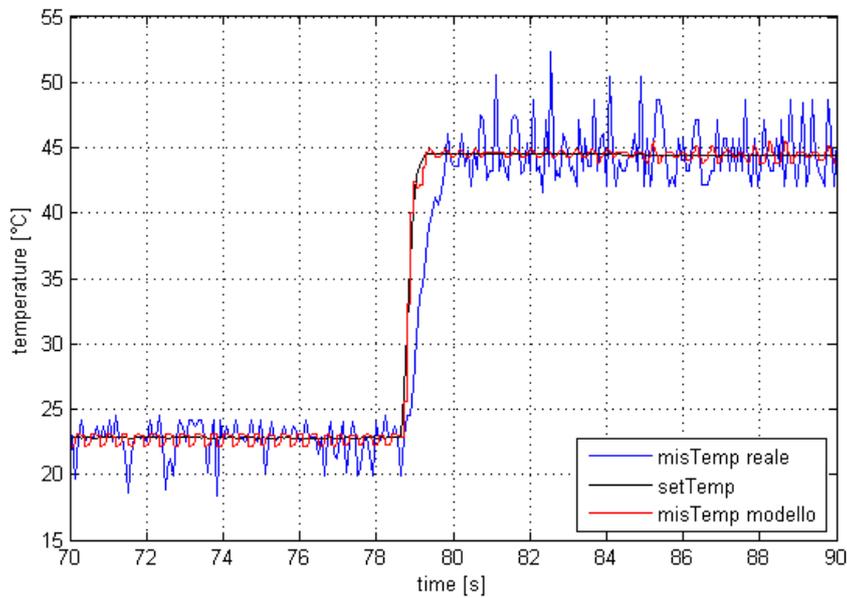


Figura 6.22: Confronto andamento della temperatura in uscita stimata, al variare del setpoint di temperatura, nei due prototipi

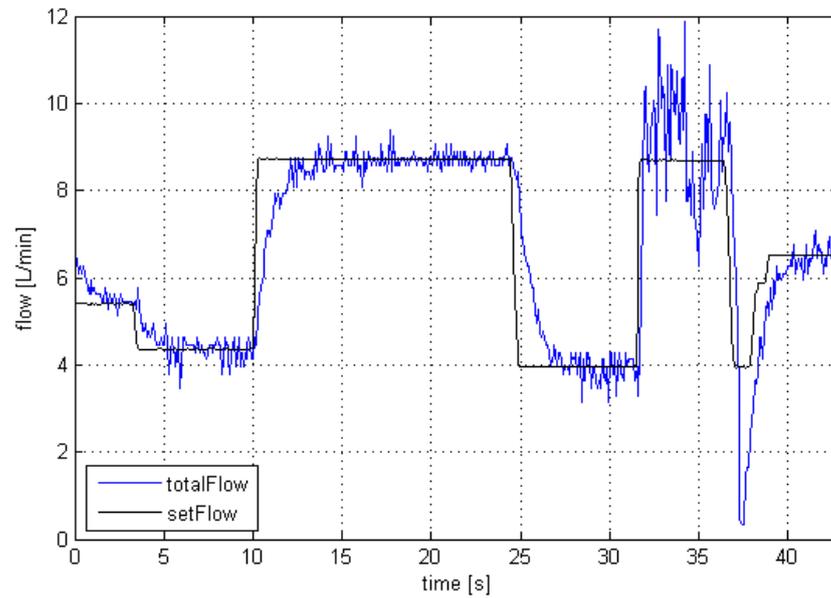
6.3.2 *Variazione setpoint di portata*

Figura 6.23: Variazione setpoint di portata con relativo inseguimento da parte della portata totale realmente ottenuta con il primo prototipo

Osservazioni: Come nel caso dell'inseguimento della temperatura, si evidenzia anche qui la bontà del secondo prototipo rispetto al primo, con un comportamento pronto e quasi privo di instabilità (oscillazioni attorno al riferimento), se non di ridottissima entità.

Ovviamente, in entrambi i test, si esegue un confronto tra un sistema fisico reale e un modello della realtà che, pur essendo molto preciso, rimane un modello. Tuttavia i risultati ottenuti sono più che soddisfacenti e interessanti in vista di una futura implementazione fisica del prototipo implementato in ambiente Matlab/Simulink.

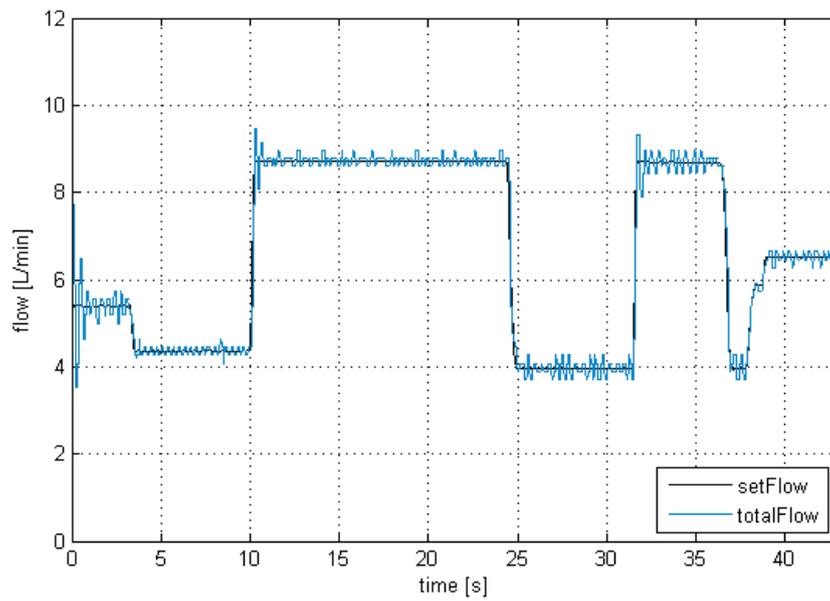


Figura 6.24: Variazione setpoint di portata con relativo inseguimento da parte della portata totale ottenuta con il secondo prototipo

INTRODUZIONE ALLA COMUNICAZIONE WIRELESS

Nell'ultima fase di lavoro è stato realizzato uno studio riguardo la comunicazione wireless in ambiente Arduino. Gli scopi principali di questa analisi sono:

- Comunicazione wireless tra due o più schede Arduino
- Programmazione Arduino via wireless

Ad oggi i tipi di comunicazione più adatti in contesti come questi, sia dal punto di vista economico che di efficienza, sono ZigBee e il protocollo 802.15.4. Questi protocolli, infatti, sono stati sviluppati per reti wireless di controllo e sensori a bassa velocità di dati. Reti di questo tipo sono largamente diffuse in automazione industriale, sistemi di sicurezza per la casa, misurazioni remote e periferiche per PC. In vista, dunque, di un inserimento nella domotica del progetto rubinetto realizzato, questo tipo di comunicazione si è presupposto più che valido.

Nei paragrafi che seguono si riporta un'analisi semplificata di questi protocolli, in particolare dell'IEEE 802.15.4 (utilizzato nel progetto), si presentano le componenti elettroniche necessarie per garantire questa comunicazione all'interno del progetto e, infine, si studiano due applicazioni pratiche esemplificative.

7.1 PROTOCOLLO 802.15.4

Il protocollo *IEEE 802.15.4* è uno standard per le comunicazioni wireless definito dall'IEEE ([7]). La presenza di uno standard in un apparato di comunicazione garantisce l'interoperabilità anche tra moduli realizzati da diversi produttori. L'IEEE 802.15.4, approvato nell'estate del 2003, definisce il protocollo di trasmissione a basso livello, tramite comunicazione radio, tra diversi dispositivi rientranti in una Personal Area Network (PAN). Le WPAN (Wireless PAN) vengono utilizzate per distribuire informazione su distanze relativamente brevi e senza cavi di collegamento; le connessioni effettuate attraverso le WPAN, infatti, riguardano piccoli ambienti ed infrastrutture, il che favorisce lo sviluppo di soluzioni poco costose ed energeticamente efficienti per un'ampia gamma di applicazioni.

Lo standard definisce, più in particolare, le specifiche del livello fisico (*PHY*) e datalink (*MAC*), al fine di garantire una modalità di connessione wireless a basso data-rate tra dispositivi fissi, portatili o mobili che necessitano di un basso consumo di potenza, ovvero lunga durata

delle batterie a bordo e che, tipicamente, lavorano in uno spazio operativo (POS: Personal Operating Space) dell'ordine di qualche decina di metri. Il data-rate deve essere sufficientemente elevato in modo da consentire la connettività di periferiche interattive largamente diffuse, come ad esempio quelle per PC, ma nel contempo vi deve essere anche la possibilità di ridurlo fino ai livelli tipici richiesti da sensori e applicazioni orientate al controllo e all'automazione delle infrastrutture. Si perviene quindi a una *LR-WPAN* (*Low Rate WPAN*), ovvero una rete di comunicazione semplice e a basso costo selettivamente orientata verso applicazioni a basso consumo e a throughput (qualità di informazione trasmessa) non elevato.

7.1.1 Topologie di rete

Una LR-WPAN può includere due diversi tipi di dispositivi: *FFD* (*Full Function Device*) e *RFD* (*Reduced Function Device*).

- *FFD*, dispositivo che può operare all'interno della rete secondo tre differenti modalità: coordinatore della rete, coordinatore semplice o terminale di comunicazione; può dialogare con altri dispositivi di entrambe le categorie (*FFD* e *RFD*).
- *RFD*, è un *EndDevice*, che può comunicare direttamente solo con un dispositivo *FFD*. L'inclusione di terminali di questo tipo all'interno di una rete è orientata per applicazioni estremamente semplici, come interruttori di luce o sensori ad infrarossi, che non necessitano l'invio di grosse quantità di dati e possono quindi essere supportate attraverso minime risorse energetiche e limitate capacità di memoria.

Una WPAN è costituita da un minimo di due dispositivi operanti in una stessa POS; in ciascuna rete, uno solo di essi può configurarsi come "PAN Coordinator" che si occupa di iniziare, gestire e terminare la comunicazione tra le diverse periferiche.

A seconda della particolare applicazione, una LR-WPAN può essere configurata in varie topologie, illustrate in Figura 7.1: comunicazione *point-to-point*, *point-to-multipoint* o *peer-to-peer*.

Point-to-point: Questa topologia di rete prevede che due dispositivi A e B comunichino unicamente tra loro. È una versione semplificata della topologia *point-to-multipoint*, costituita da due soli nodi.

Point-to-multipoint: Nelle applicazioni *point-to-multipoint*, dette anche "a stella", è richiesto un coordinatore (nodo centrale) che può ricevere e trasmettere da/verso tutti e si occupa in qualche modo di avviare e gestire la rete, mentre i nodi periferici (*end devices*) possono soltanto comunicare con il coordinatore. Il coordinatore è tipicamente collegato ad alimentazione fissa, mentre gli altri dispositivi sono

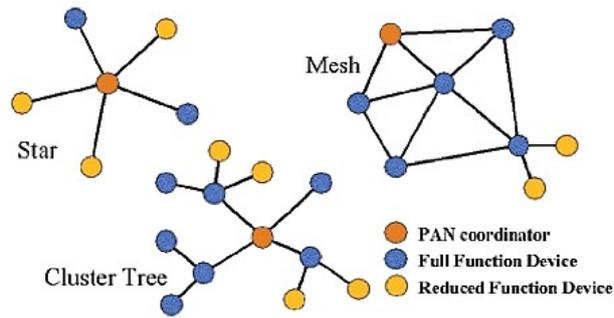


Figura 7.1: Principali topologie di rete previste dallo standard IEEE 802.15.4

dotati di batteria a bordo oppure alimentati da fonti energetiche innovative (mini pannelli solari, sistemi piezoelettrici, ecc.).

Peer-to-peer: Questo tipo di rete differisce dalla precedente in quanto ciascun dispositivo può comunicare direttamente con un altro interno alla rete, a patto che questo rientri nella sua area operativa di copertura, senza ricorrere alla mediazione del coordinatore. Una tale topologia si presta, quindi, alla formazione di reti di comunicazione decisamente complesse che coinvolgono potenzialmente un elevato numero di dispositivi: essa è tipicamente rivolta ad applicazioni come il controllo e il monitoraggio industriale e ambientale, attraverso reti di sensori wireless, in aggiunta all'utilizzo in domotica. Un esempio di rete complessa fondata su una topologia "peer-to-peer" è il "cluster-tree".

In ogni caso, ciascun dispositivo interno alla rete possiede un indirizzo esteso a 64bit. Tale indirizzo può essere direttamente utilizzato per la comunicazione oppure, in alternativa, si può ricorrere a un indirizzo ridotto, attribuito dal PAN Coordinator ogni volta che il dispositivo viene da esso rilevato.

7.1.2 ZigBee e 802.15.4

Il protocollo ZigBee è stato definito dalla *ZigBee Alliance* ([8]), un'associazione no-profit di circa 150 aziende (Mitsubishi, Samsung, Siemens, Philips, ...) che ha definito e sviluppato un protocollo semplice da utilizzare, a basso consumo e che soddisfacesse varie esigenze, sia commerciali che industriali. Proprio per questo, la ZigBee Alliance ha scelto come base di partenza lo standard 802.15.4.

ZigBee opera nelle frequenze radio assegnate per scopi industriali, scientifici e medici (ISM); 868 MHz in Europa, 915 MHz negli Stati Uniti e 2.4 GHz nella maggior parte del resto del mondo. Questa tecnologia ha lo scopo di essere più semplice e più economica di altre

WPAN come, ad esempio, *Bluetooth*. Il nodo ZigBee del tipo più complesso si dice che richieda solamente il 10% del codice necessario per un tipico nodo Bluetooth o Wi-Fi, mentre il più semplice dovrebbe richiedere intorno al 2%.

I protocolli ZigBee sono progettati per l'uso in applicazioni embedded che richiedano un basso *transfer rate* e bassi consumi. L'obiettivo attuale di ZigBee è di definire una *Wireless mesh network* non mirata, economica e autogestita che possa essere utilizzata per scopi quali il controllo industriale, le reti di sensori, la domotica, le telecomunicazioni. La rete risultante avrà un consumo energetico talmente basso da poter funzionare per uno o due anni sfruttando la batteria incorporata nei singoli nodi.

La differenza principale tra lo standard 802.15.4 e ZigBee è il tipo di collegamento tra le varie unità interessate. Il collegamento di tipo *mesh* (Figura 7.2), è utile quando due moduli (A e B, guardando Figura 7.2) devono comunicare tra loro ma la distanza che li separa è eccessiva: in questo caso altri moduli presenti nella rete, operanti come *router* (C e D), possono effettuare il rilancio del messaggio.

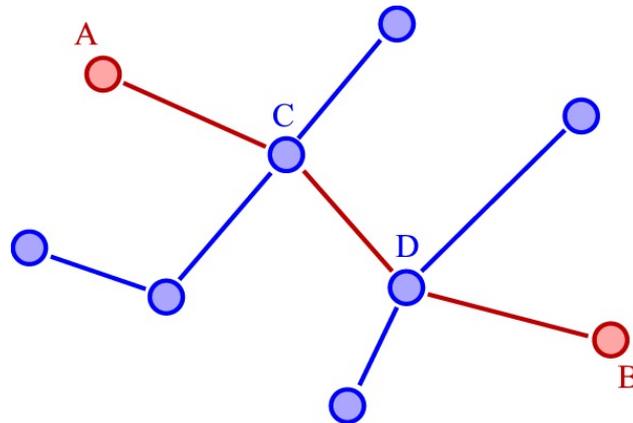


Figura 7.2: Esempio di comunicazione di tipo *mesh*

Il protocollo ZigBee è strutturato in maniera tale che il trasferimento da A a B attraverso i router intermedi avvenga in automatico, senza interventi esterni: i nodi costituenti la rete capiscono l'esigenza e si organizzano affinché il messaggio possa giungere a destinazione. Se per caso un nodo viene rimosso, gli altri moduli si accorgono dell'evento e riorganizzano la rete sempre in maniera autonoma. Infine, in caso di errore di trasmissione, il protocollo assicura che il messaggio venga inviato nuovamente.

Questo tipo di rete ha bisogno di un coordinatore per poter funzionare, il quale mette in opera la rete, dopodiché agisce come un normale router. Nella rete vi possono essere altri dispositivi operanti come semplici router e, inoltre, vi sono sicuramente degli end devices.

7.2 I MODULI XBEE

7.2.1 Cosa sono e cosa fanno

La tecnologia Arduino che utilizza i protocolli 802.15.4 e ZigBee è basata sui dispositivi XBee ([9]) moduli prodotti dalla DIGI (ex MaxStream) che consentono il trasferimento dati via RF (Radio Frequenza), sfruttando una normale comunicazione seriale. Utilizzando questa strategia, infatti, per trasferire dati in modalità wireless è sufficiente saper utilizzare la periferica UART inclusa nel microcontrollore o la porta seriale del computer: una coppia di XBee può essere vista come un sostituto di un cavo seriale.



Figura 7.3: Esempio di modulo XBee

In Figura 7.4, è riportata una schematizzazione del funzionamento della comunicazione tra due moduli XBee: tutto ciò che il microcontrollore invia sul pin RX di un modulo XBee, giunge ad un altro XBee sul pin TX e può quindi essere ricevuto dal dispositivo remoto. Questo particolare modo di operare degli XBee viene anche detto *Modalità trasparente (Transparent Mode)* e, anche se non è l'unica modalità operativa di questi moduli, è tuttavia quella più semplice e forse più utilizzata.

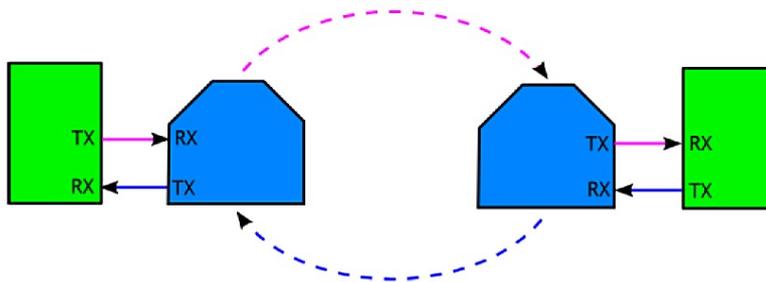


Figura 7.4: Schematizzazione della comunicazione tra due moduli XBee

Gli XBee più diffusi operano alla frequenza di 2.4 GHz, ma esistono anche XBee operanti a 900 MHz e a 868 MHz, ma non tutti sono utilizzabili in Italia. Si noti che, sebbene anche il WiFi operi a 2.4 GHz,

i due tipi di comunicazione non interferiscono tra loro, in quanto utilizzano protocolli differenti.

7.2.2 *Vantaggi di utilizzo*

Quando ci si trova nella necessità di dover realizzare un'applicazione che comunichi senza fili, ci si imbatte in una grande vastità di moduli wireless e spesso si fanno dei confronti con moduli RF a basso costo che operano sui 433 MHz. Tali moduli sicuramente costano meno e sono utilizzabili per semplici applicazioni, tuttavia le differenze con gli XBee sono notevoli e alcune caratteristiche spesso vengono trascurate o sottovalutate. Tra i principali vantaggi derivanti dall'utilizzo di un modulo XBee, si evidenziano i seguenti:

- Un modulo XBee è bidirezionale: un singolo modulo è in grado sia di ricevere che di trasmettere. Molti moduli economici possono operare in una sola delle due direzioni.
- Un modulo XBee può essere indirizzato in maniera univoca: ogni XBee ha un numero seriale che lo distingue da tutti gli altri XBee. Tramite una semplice utility (X-CTU) è infatti possibile fare in modo che due moduli comunichino solo tra loro senza eccessivi sforzi di programmazione software. Inoltre, questo consente di avere molti moduli all'interno di una stessa area, senza che essi interferiscano tra loro.
- Nei moduli XBee sono già implementati, nell'hardware, tutti i controlli tipici di una comunicazione wireless, gestiti in maniera automatica senza l'intervento dell'utente. Inoltre, in caso di errori di trasmissione, i dati vengono ritrasmessi finché non giungono a destinazione correttamente. Questi controlli, sui moduli economici, devono essere realizzati via software.
- Sugli XBee è possibile assegnare diversi canali ai moduli presenti all'interno di una stessa area, al fine di minimizzare ulteriormente le interferenze.

7.2.3 *Caratteristiche elettriche e meccaniche*

E' molto facile riconoscere un modulo XBee: la loro forma è particolare ed è stata copiata da molti produttori nella realizzazione di altri moduli, soprattutto nei confronti della piedinatura. I moduli XBee possiedono, infatti, una doppia fila di piedini con una spaziatura (pitch) di 2mm, quindi leggermente più stretta di quella "classica" da 2.54mm a cui molti utenti sono abituati con millefori e breadboard. Generalmente gli XBee vengono utilizzati su schede di adattamento che prevedono anche una regolazione di tensione, dal momento che lavorano a 3.3V.

Vi sono differenti famiglie di XBee, che si suddividono quasi sempre in due varianti: il modello *Low Power* e il modello *PRO*. Di cui, i primi offrono consumi più ridotti, distanza di comunicazione inferiore, a fronte di un costo decisamente più basso. I modelli *PRO*, invece, sono anche identificati con la dicitura *Extended Range*, perchè ricoprono distanze superiori rispetto al corrispettivo *Low Power*, pur avendo, per contro, un consumo e un costo più elevati. Esteticamente i due modelli si differenziano dal fatto che i *PRO* sono leggermente più lunghi e presentano la scritta "PRO" chiaramente stampata sulla parte superiore del corpo (Figura 7.5 e Figura 7.6).

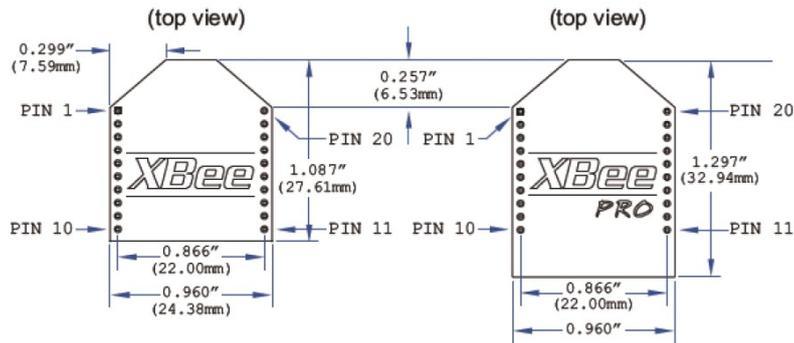


Figura 7.5: XBee e XBee-PRO, front

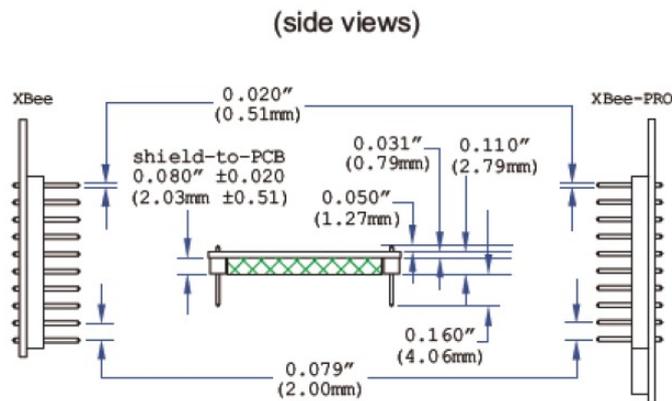


Figura 7.6: XBee e XBee-PRO, side

I moduli *Low Power* esistono in una sola variante: sono uguali in tutte le parti del mondo. I moduli *PRO*, invece, possono esistere in due varianti, a seconda del paese a cui sono destinati, a causa delle diverse legislazioni relative alle trasmissioni RF: negli Stati Uniti, infatti, sono concesse potenze di trasmissione superiori a quelle previste in altre parti del mondo. I modelli *PRO* destinati all'Europa e ad altre parti del mondo riportano la lettera J nella parte finale della sigla.

7.2.4 Adattamento dei livelli logici

Come accennato in precedenza, i moduli XBee lavorano a una tensione di 3.3V. Si rivela quindi necessario prestare gli opportuni accorgimenti nel momento in cui si vuole interfacciare questi moduli con microcontrollori operanti a 5V. Sulla linea TX dell'XBee (collegata alla linea RX del dispositivo utilizzato), non è necessario eseguire alcun adattamento di tensione: le logiche a 5V riconoscono 3.3V come livello alto. Infatti, le logiche TTL a 5V hanno una soglia indeterminata del range 1.5-2V, cioè, le tensioni con valori compresi in questo intervallo potrebbero essere riconosciute sia come livello alto che come livello basso. Al di sopra di 2V vengono sicuramente riconosciute come livello alto. I problemi riguardano, invece, la linea RX dell'XBee, che non può assolutamente ricevere una tensione di 5V, pena il danneggiamento. In questo caso basta un semplice *partitore di tensione* per fare in modo che i 5V (che identificano il livello logico alto) forniti dal dispositivo utilizzato sulla linea TX, giungano all'XBee come 3.3V, senza recare danni (Figura 7.7).

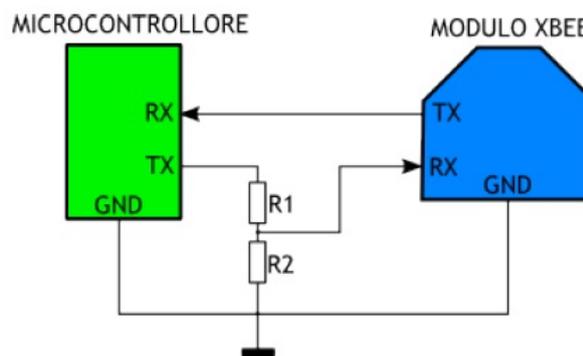


Figura 7.7: Utilizzo di un partitore di tensione per l'adattamento dei livelli logici

L'utilizzo di un partitore di tensione è sicuramente la soluzione più semplice ed efficace, anche se induce un certo *slew rate* che, tuttavia, non reca problemi con le velocità di comunicazione massime consentite dagli XBee. Si potrebbe, però, incorrere in errori di trasmissione nei casi in cui la lunghezza dei collegamenti tra dispositivo e XBee fosse eccessiva, anche se, generalmente, questa situazione non si presenta.

Il partitore di tensione, tuttavia, non è l'unico sistema per adattare i livelli logici. Sulle schede di adattamento che prevedono un collegamento su USB o su RS232, è sempre presente l'opzione di *level shifting*, che può essere realizzata secondo differenti modalità.

7.2.5 Tipi di antenna

I vari modelli di XBee appartenenti a una famiglia (sia le varianti Low Power che le varianti PRO) si distinguono tra loro anche in base al tipo di antenna. Antenne più performanti consentono di avere una maggiore potenza di trasmissione e, di conseguenza, una maggiore distanza coperta. I costi (e i consumi) ovviamente aumentano man mano che il tipo di antenna consente una copertura maggiore.

I modelli con minore potenza di trasmissione, ma più economici e compatti, hanno un'antenna stampata su PCB o un'antenna a chip o a filo (Figura 7.8).

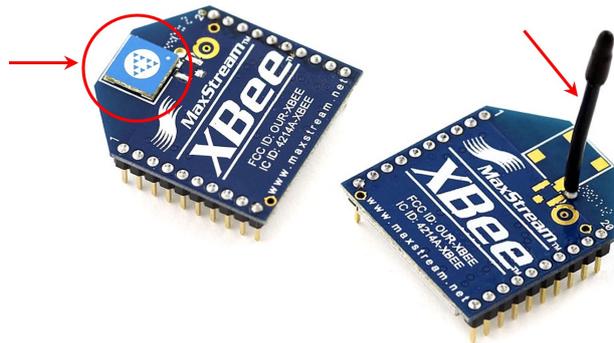


Figura 7.8: XBee con antenna a chip e a filo

Questi due modelli, con antenna integrata (*embedded antenna*), tuttavia, per garantire buone prestazioni, non possono essere rinchiusi all'interno di contenitori metallici. Nel caso di utilizzo di contenitori metallici, è necessario ricorrere all'uso di modelli dotati di antenna esterna, del tipo di Figura 7.9. Tali modelli consentono di raggiunge-

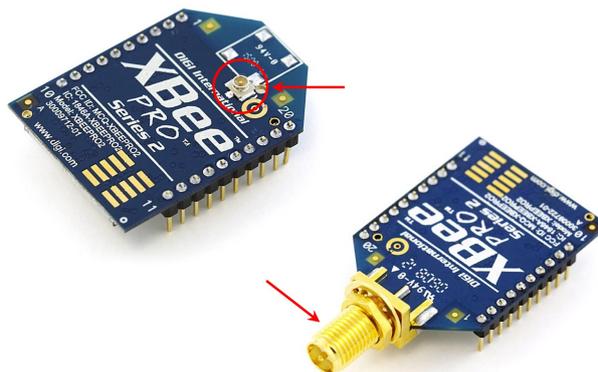


Figura 7.9: XBee con antenna esterna

re il massimo della potenza di trasmissione, a fronte di un'opportuna scelta dell'antenna da collegare. Essi prevedono un connettore U.FL o un connettore RP-SMA. Il connettore U.FL (in Figura 7.9 a sx) è coassiale, miniaturizzato, con innesto a pressione. I connettori di questo

tipo non sono progettati per essere rimossi ed innestati di continuo in quanto sono molto piccoli e abbastanza delicati. Generalmente, un'antenna di questo tipo, una volta inserita, non viene più rimossa e, inoltre, necessita di essere agganciata ad un pannello in quanto non può essere collegata direttamente sull'XBee. Il connettore RP-SMA (a dx in Figura 7.9), invece, pur essendo anch'esso coassiale, ha dimensioni decisamente maggiori e un fissaggio a vite. Tale connettore è molto robusto e consente di avvitare l'antenna anche direttamente al corpo dell'XBee, oltre che ad un pannello esterno, utilizzando un cavo di prolunga.

7.2.6 Serie 1 e Serie 2: le differenze

Gli XBee vengono comunemente identificati come *Serie 1* o *Serie 2*. Questa denominazione, in realtà, non è indicativa delle varie famiglie di XBee, ma è piuttosto un riferimento all'hardware montato a bordo, che permette loro di funzionare in un modo piuttosto che in un altro (Figura 7.10).

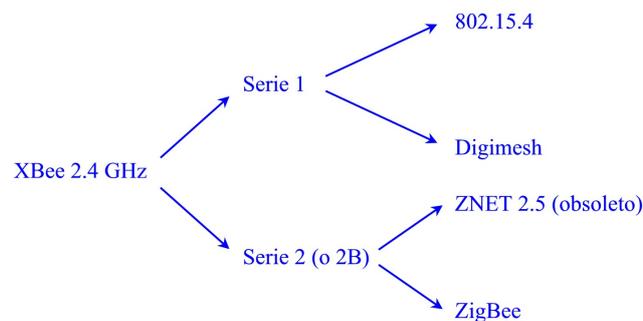


Figura 7.10: Suddivisione *hardware* schematica delle diverse famiglie XBee

Tuttavia, la DIGI (loro casa di produzione) fa una distinzione basata sul protocollo utilizzato piuttosto che sull'hardware (si veda Figura 7.11), dividendo gli XBee in due grandi famiglie: **Point-to-multipoint** e **Mesh** (suddivisa ancora, a sua volta, nelle due sottofamiglie ZigBee e Digimesh).

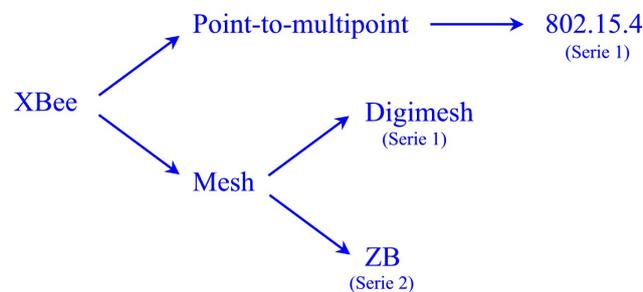


Figura 7.11: Suddivisione *software* schematica delle diverse famiglie XBee

Nei casi in cui si desidera realizzare una comunicazione *point-to-*

point (comunicazione tra due soli dispositivi) è sicuramente conveniente scegliere i moduli appartenenti alla famiglia XBee 802.15.4 OEM RF modules, ovvero la Serie 1. Questa scelta è consigliata anche dalla stessa DIGI, in quanto questi moduli di default sono già impostati per la comunicazione in modalità trasparente e non necessitano di particolari settaggi (per una comunicazione punto-punto potrebbero essere utilizzati anche gli XBee Serie 2, ma sono più difficili da utilizzare in quanto andrebbero opportunamente configurati dato che essi nascono per lavorare in una rete Mesh secondo il protocollo ZigBee). Generalmente si sceglie un modulo di Serie 2 nei casi in cui si desidera implementare una topologia di rete Mesh secondo il protocollo ZigBee, mentre si sceglie un modulo Serie 1 Digimesh quando si desidera implementare una rete Mesh senza coordinatori. Molte volte, erroneamente, si crede che la Serie 2 sia la sostituta della Serie 1 e che quindi i primi moduli siano solo più potenti o più performanti dei secondi. Ciò che realmente distingue le due serie, invece, è che la Serie 1 permette la realizzazione di alcune topologie di rete, mentre i modelli Serie 2 permettono la comunicazione secondo lo standard ZigBee. La scelta di una o dell'altra serie, dunque, deve essere guidata solamente dal tipo di comunicazione che si vuole ottenere.

7.3 APPARATO SPERIMENTALE

La comunicazione wireless tra due o più schede Arduino e la possibilità di programmare una di queste schede a distanza, senza la necessità di un collegamento via cavo, sono i due aspetti che hanno reso importante lo studio sulla possibile applicazione del wireless nel progetto rubinetto. Una comunicazione di questo tipo, tuttavia, richiede, come sempre, un'adeguata strumentazione, che assicuri il corretto collegamento tra le varie componenti interessate e quindi il corretto funzionamento dell'intero sistema. Di seguito sono illustrati i diversi dispositivi necessari all'esecuzione delle prove sperimentali.

7.3.1 XBee Serie1

Tra le differenti versioni di moduli XBee, il modulo XBee Serie 1 con antenna a chip (Figura 7.12), si è rivelato essere la scelta migliore in questo contesto. Questa tipologia di XBee, infatti, come si è visto nel precedente paragrafo, è conforme allo standard 802.15.4 ed è in grado di eseguire comunicazioni di tipo *point-to-point* o *point-to-multipoint* (adatte per questo progetto), in modo molto semplice e immediato, senza la necessità di ricorrere a particolari accorgimenti. In Tab. 7.1 si riportano alcune delle sue specifiche più importanti.

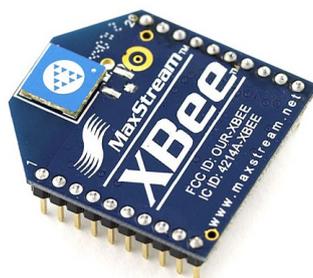


Figura 7.12: XBee Serie 1, Low Power

Specification	Values
Indoor/Urban Range	up to 100 ft. (30 m)
Outdoor RF line-of-sight Range	up to 300 ft. (100 m)
Transmit Power Output	1 mW
RF Data Rate	250,000 bps
Serial Interface Data Rate	1200-115200 bps
Supply Voltage	2.8-3.4 V
Transmit Current	45 mA
Receive Current	50 mA
Operating Frequency	ISM 2.4 GHz
Operating Temperature	-40 to 85°C

Tabella 7.1: Tabella riassuntiva caratteristiche principali XBee Serie 1

7.3.2 XBee Shield

Le Shield sono delle schede che possono essere applicate direttamente su Arduino, aumentando ulteriormente le sue prestazioni. Queste schede sono semplici da applicare ed economiche nella realizzazione. *XBee Shield* (Figura 7.13), in particolare, permette ad Arduino di comunicare via wireless utilizzando il protocollo ZigBee. Creata in collaborazione con *Libelium*, la sua struttura è basata sul modulo XBee di casa DIGI (ex MaxStream), il quale si applica perfettamente, in

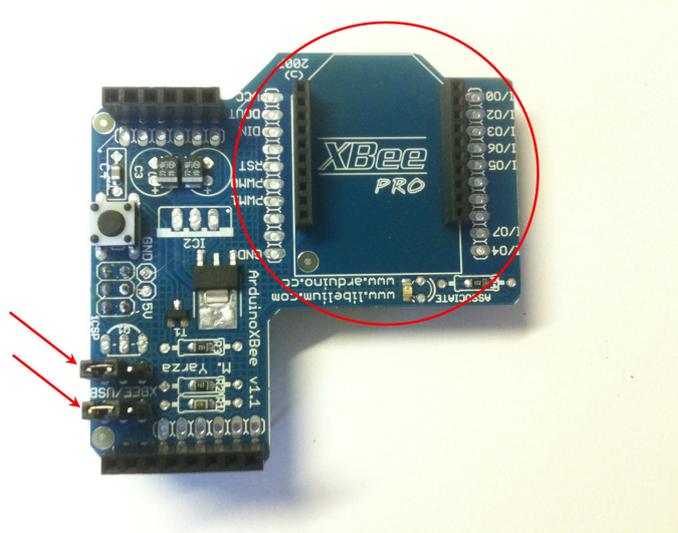


Figura 7.13: Arduino XBee Shield

corrispondenza della parte cerchiata in rosso. Ogni pin maschio del modulo XBee si collega con un pin femmina della Shield. Inoltre sono presenti sulla scheda delle testate di pin femmina per l'accesso ai pin digitali dal 2 al 7 e agli ingressi analogici di Arduino, coperti dalla shield (i pin digitali dall'8 al 13 non sono coperti dalla shield e quindi sono accessibili direttamente da Arduino).

La Shield è dotata di due jumper: due piccoli ponticelli di plastica removibili, indicati con le frecce rosse, che raggruppano due dei tre pin presenti in corrispondenza dell'etichetta XBEE/USB. Questi determinano come la comunicazione seriale dei moduli XBee si connette alla comunicazione seriale presente tra il microcontrollore e il chip FDTI USB-to-serial della scheda Arduino.

Con i jumper in posizione **XBEE**, il pin DOUT del modulo XBee è connesso al pin RX del microcontrollore, e DIN è connesso al pin TX (si ricordi che i pin RX e TX del microcontrollore sono connessi rispettivamente ai pin TX e RX del chip FDTI): i dati inviati dal microcontrollore sono trasmessi al computer via USB oltre ad essere inviati via radio dal modulo XBee. Il microcontrollore, tuttavia, è abilitato a ricevere dati dal modulo XBee, ma non da computer (via USB).

Con i jumper in posizione **USB**, invece, il pin DOUT del modulo XBee

è collegato al pin RX del chip FDTI, e il pin DIN del modulo XBee è collegato al pin TX del chip FDTI. Questi collegamenti permettono al modulo XBee di comunicare direttamente con il computer (questo è possibile solo se viene rimosso il microcontrollore dalla scheda Arduino). Se il microcontrollore viene esportato dalla scheda Arduino, essa è in grado di comunicare normalmente con il computer via USB, ma né il computer né il microcontrollore riescono a parlare al modulo XBee. Generalmente questa posizione dei jumper viene mantenuta quando si vuole caricare un programma sulla scheda Arduino (da computer via USB). Prima di eseguire questa operazione, però, è necessario asportare il modulo XBee dalla shield.

7.3.3 XBee USB Board

La scheda XBee - USB Board (Figura 7.14) è stata progettata per consentire una connessione semplice e affidabile tra i moduli XBee e il PC. Tutti i tipi di moduli XBee sono supportati (XBee, XBee PRO, XBee Serie 2). La solida costruzione, la circuiteria Soft Start e il poten-

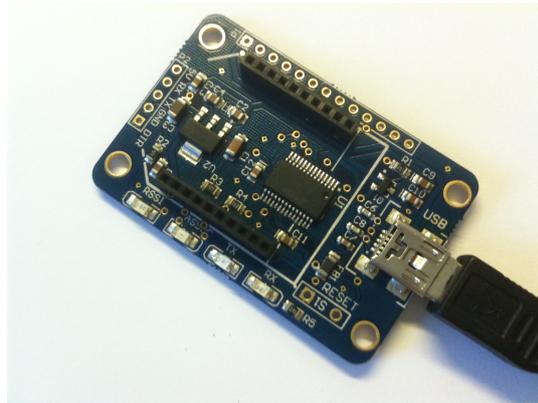


Figura 7.14: 990.002 XBee USB Board

te regolatore di tensione assicurano la massima affidabilità e le massime prestazioni dei moduli XBee. In Figura 7.15 si evidenzia il layout della XBee USB Board. Quattro utili LED consentono il costante monitoraggio dell'attività della scheda, semplificando e velocizzando la ricerca di problematiche. Un connettore permette l'accesso alle più importanti connessioni con il modulo XBee e con il chip USB, aggiungendo grande flessibilità a questa scheda. Il connettore USB Mini B, infine, assicura dimensioni ridotte e una scheda semplice e comoda da utilizzare.

Specifiche tecniche

- *Moduli supportati:* XBee, XBeePRO, XBee Series 2
- *Porta USB:* è richiesta una porta USB 1.0/2.0 alimentata (500 mA)

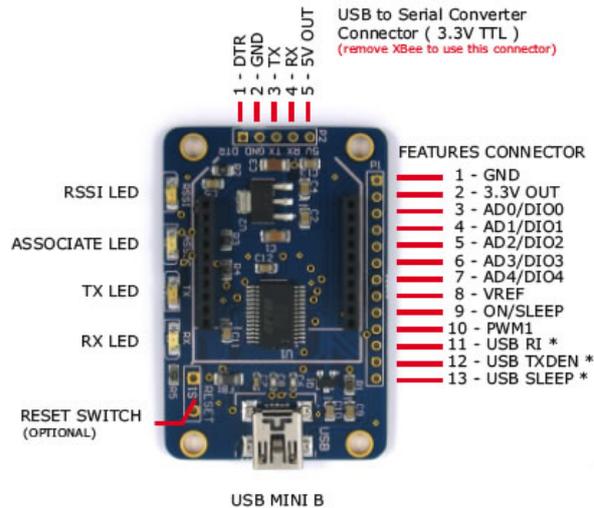


Figura 7.15: 990.002 XBee USB Board - Layout

- *3.3V OUT*: La potenza di uscita dipende dal modulo utilizzato. Circa 300 mA disponibili con modulo XBee, e circa 140 mA con modulo XBeePRO
- *Dimensioni*: 51,5 x 34 mm, altezza massima 7 mm senza modulo XBee

Caratteristiche tecniche

- LED RSSI e ASSOCIATE per monitorare l'attività del modulo XBee
- LED TX e RX per monitorare l'attività seriale tra il modulo XBee e il chip USB
- 3.3V OUT: uscita ausiliaria per alimentare dispositivi esterni (microcontrollori, schede, sensori, etc.)
- Circuito Soft Start per massimizzare l'affidabilità
- Connettore laterale per accedere direttamente alle connessioni più importanti del modulo XBee e del chip USB
- Connettore USB to Serial: è possibile utilizzare la scheda come un convertitore USB-to-Serial

7.4 MOTIVAZIONE E PRESENTAZIONE PROVE SPERIMENTALI

Lo scopo principale di questo studio è l'utilizzo di una comunicazione di tipo wireless nel sistema rubinetto. L'idea di base è quella di eliminare i cavi di collegamento tra il case idraulico e il case interfaccia (visti nel Capitolo 4). Per fare ciò si è pensato di inserire all'interno

di ognuno dei due case una scheda Arduino e far comunicare le due schede via radio, tramite due moduli XBee, caricando al loro interno opportuni programmi. In questo modo i dati provenienti dai sensori di temperatura e portata, letti dalla scheda Arduino presente nell'unità idraulica, vengono inviati via wireless all'unità di controllo. La scheda Arduino presente all'interno di questa, a sua volta, elabora i dati ricevuti e individua, di conseguenza, il corretto comando da inviare (tramite la prima scheda, sempre via wireless) alle elettrovalvole proporzionali.

Aggiungendo, eventualmente, un terzo modulo XBee collegato al computer tramite XBee USB Board, è possibile effettuare il monitoraggio dei consumi, ricevendo sul pc i dati dalla scheda Arduino presente nell'unità di controllo, che in questo caso prenderebbe il ruolo di coordinatore in una rete a stella composta da tre elementi.

Un secondo scopo, di importanza relativamente minore, ma altrettanto interessante, è la programmazione a distanza delle schede Arduino presenti nei due case, che eliminerebbe la necessità di collegamenti di tipo USB con il computer. In Figura 7.16 si trova una rappresentazione schematizzata dell'idea sopra esposta.

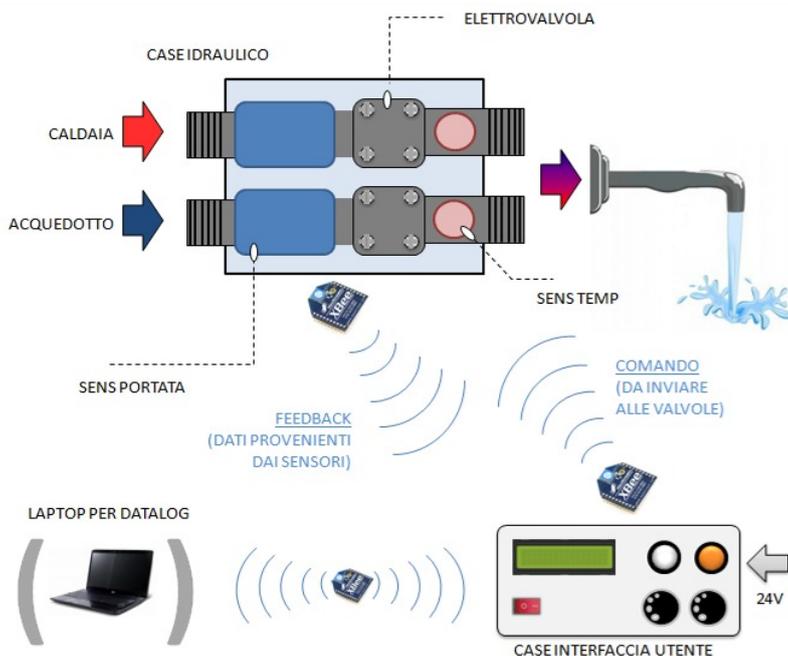


Figura 7.16: Rappresentazione schematizzata inserimento wireless nel progetto rubinetto

Di seguito si analizzano due possibili applicazioni del wireless, in via del tutto generale, in modo da poterne studiare la realizzabilità e quindi l'applicabilità nel contesto di interesse, sopra esposto.

7.4.1 Comunicazione wireless tra due Arduino

E' possibile far comunicare due schede Arduino tramite XBee Shield e moduli XBee, senza alcun tipo di configurazione, utilizzando semplicemente i comandi seriali standard di Arduino. E' sufficiente, infatti, applicare su entrambe le schede una XBee Shield, che accoglierà il modulo XBee adatto all'esperienza che si vuole eseguire (in questo caso si ha una comunicazione point-to-point, quindi si opta per il modulo XBee Serie 1), e caricare sulle due schede il programma corretto. Di seguito si analizza un semplice esempio di comunicazione wireless tra due Arduino, atto a dimostrarne la realizzabilità, in cui una delle due schede invia un comando logico alto o basso (H o L) al pin 13 (led) dell'altra scheda, la quale a seconda del comando ricevuto accende o spegne il led. Si riportano i due programmi caricati sulle due schede, ricordando che, per poter caricare uno sketch su una scheda, quando la XBee Shield è applicata, ci si deve assicurare che i jumper presenti sulla shield siano sulla posizione "USB" e che il modulo XBee sia stato esportato.

Listing 1: Sketch scheda n°1

```
const int ledPin = 13; // the pin that the LED is attached to
int incomingByte;      // variable to read incoming serial data

void setup()
{
  // initialize serial communication:
  Serial.begin(9600);
  // initialize the LED pin as an output:
  pinMode(ledPin, OUTPUT);
}
void loop()
{
  // see if there's incoming serial data:
  if (Serial.available() > 0)
  {
    // read the oldest byte in the serial buffer:
    incomingByte = Serial.read();
    // if it's a capital H (ASCII 72), turn ON the LED:
    if (incomingByte == 'H')
    {
      digitalWrite(ledPin, HIGH);
    }
    // if it's an L (ASCII 76), turn OFF the LED:
    if (incomingByte == 'L')
    {
      digitalWrite(ledPin, LOW);
    }
  }
}
```

Come accennato in precedenza, questo sketch comanda alla scheda su cui viene caricato di accendere il led associato al pin 13 quando riceve un 'H' lungo la linea seriale, e di spegnerlo quando riceve 'L'. Una volta caricato questo programma sulla prima scheda, la si disconnette dal computer e si posizionano i suoi jumper in posizione 'XBEE'. Dopodiché si prende l'altra scheda, sulla quale è già stata posizionata la XBee Shield con i jumper in posizione 'USB', la si connette al pc via USB e gli si carica il seguente programma.

Listing 2: Sketch scheda n°2

```
void setup()
{
  Serial.begin(9600);
}

void loop()
{
  Serial.print('H');
  delay(1000);
  Serial.print('L');
  delay(1000);
}
```

Una volta eseguito l'upload di questo codice sulla seconda scheda, aprendo la serial monitor, si vedono comparire uno dopo l'altro i caratteri 'H' e 'L' in tempi distanziati di 1 secondo. Verificato il corretto funzionamento del secondo programma, si procede a disconnettere anche la seconda scheda dal computer e a posizionare i jumper della sua XBee Shield sull'opzione 'XBEE'.

A questo punto, dopo aver posizionato i moduli XBee Serie 1 in corrispondenza degli appositi slot presenti sulle XBee Shield delle due schede, e dopo aver collegato le due schede a una fonte di alimentazione (non necessariamente al computer), è possibile vedere il led della prima scheda accendersi e spegnersi a distanza di 1 secondo, sotto comando della seconda scheda. Le due schede comunicano via wireless. Si noti che anche sulle XBee Shield lampeggia un led, ma la sua funzione è semplicemente quella di fornire informazioni riguardo lo stato del modulo XBee.

Al di là del semplice esempio sopra riportato, è utile sapere che con XBee Shield è possibile utilizzare qualsiasi comando standard relativo alla comunicazione seriale di Arduino. Quando i jumper delle shield sono posizionati su 'XBEE', i comandi sulla seriale *print* e *println* inviano dati tramite XBee Shield e la connessione USB (quindi alle altre shield e al computer contemporaneamente). In questa configurazione, la scheda, invece, può ricevere dati solamente dalla shield e non dalla connessione USB (sarebbe necessario applicare uno switch ai jumper per poter ricevere dati dal computer). Il modulo XBee presente sulla

shield di default lavora a 9600 baud.

Per permettere al computer di comunicare direttamente con la XBee Shield, è necessario collegare la shield a una scheda Arduino privata di microcontrollore e posizionare i suoi jumper in posizione 'USB'. In questo modo è possibile inviare e ricevere dati al/dal modulo XBee da qualsiasi terminale. Questa opzione viene utilizzata, ad esempio, quando si vogliono monitorare i dati che il modulo sta ricevendo dalle diverse XBee Shield.

7.4.2 Programmazione wireless di Arduino

Quando si lavora a un progetto con Arduino e la scheda si trova, ad esempio, all'interno di un case o comunque in una posizione inaccessibile con semplicità, una programmazione di tipo wireless porta un risparmio di tempo e di energie notevole. Per rendere possibile questa operazione è necessario disporre dei seguenti dispositivi:

- N.1 XBee USB Board
- N.2 XBee Serie 1
- N.1 XBee Shield
- N.1 Arduino Board
- N.1 cavo mini USB
- N.1 cavo alimentazione Arduino
- N.1 0.1 μ F ceramic capacitor
- N.1 10KOhm resistor
- N.1 NPN transistor

I passaggi da seguire per la programmazione di Arduino via wireless sono riportati in maniera dettagliata in [10]. Di seguito si accenna brevemente la procedura effettuata:

1. Innanzitutto si devono configurare i due moduli XBee, in modo tale che uno dei due sia il 'reset transmitter' e l'altro sia il 'receiver'. Per fare ciò è necessario connettere, prima uno e poi l'altro modulo, al computer tramite la scheda XBee USB Board (analizzata in dettaglio al paragrafo 7.3.3), collegata al pc tramite il cavo mini USB. La configurazione, sia per transmitter che per receiver, prevede l'utilizzo del software di casa DIGI, X-CTU, secondo i passaggi riportati in [10].
2. Il 'reset transmitter' ha il compito di inviare via wireless il programma alla scheda Arduino sulla quale è posizionato il modulo XBee scelto come 'receiver'. Una volta seguite le istruzioni di

configurazione, è necessario effettuare una particolare saldatura sulla XBee USB Board (come riportato, in dettaglio, in [10]), che permette di effettuare il 'reset' di Arduino. Questa operazione si rivela necessaria, in quanto il 'reset', che generalmente viene eseguito in maniera automatica via software quando si programma Arduino via cavo, in questo modo non sarebbe garantito, se non seguendo opportuni accorgimenti.

3. Allo stesso modo, dopo aver configurato l'XBee 'receiver', lo si collega ad Arduino, utilizzando la XBee Shield e collegando opportunamente tra loro le linee di reset di XBee e Arduino, secondo lo schema elettrico riportato in Figura 7.17.

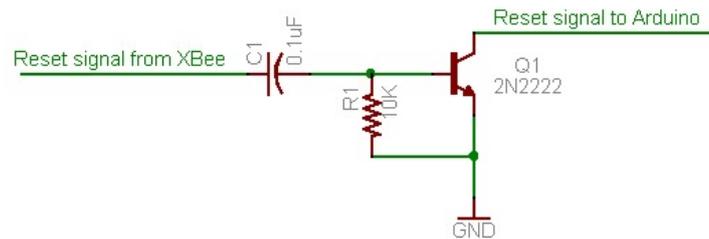


Figura 7.17: Schema elettrico collegamento linee di reset XBee e Arduino

4. Una volta sistemata la linea di reset, si può collegare Arduino a una fonte di alimentazione e programmarlo a distanza e osservare la sua serial monitor, anche a 30 m di distanza.

CONCLUSIONI

Nel presente lavoro sono stati analizzati due differenti prototipi di miscelatore automatizzato per rubinetti di uso sanitario. La peculiarità delle due soluzioni individuate è il controllo, implementato su piattaforma Arduino, che garantisce il monitoraggio della portata e della temperatura dell'acqua in uscita dal dispositivo, con ingombri e costi molto ridotti. Elettrovalvole proporzionali (nel primo prototipo) e vitoni ceramici (nel secondo), opportunamente comandati, stabiliscono, sulla base delle misure ricevute dai sensori di portata e temperatura, il corretto grado di apertura delle linee di acqua calda e acqua fredda al fine di soddisfare le specifiche richieste dall'utente.

L'apparato fisico è costituito di due unità: unità di controllo e unità idraulica. La prima permette all'utente di interfacciarsi con il sistema rubinetto mentre la seconda riceve le informazioni dalla prima e, in base a queste, comanda i dispositivi (sensori e valvole) contenuti al suo interno.

L'attività eseguita ha previsto la realizzazione fisica solo del primo prototipo, mentre il secondo è stato modellizzato e simulato in ambiente Matlab/Simulink, per studiarne la realizzabilità e l'efficienza, in vista di una sua futura applicazione fisica.

Tra i vantaggi del sistema realizzato si evidenzia in primo luogo l'efficienza dell'hardware: la piattaforma Arduino garantisce

- estrema facilità di programmazione, basata su linguaggio C++
- elevata modularità, che permette di collegare rapidamente segnali provenienti da sensori o attuatori di vario tipo
- possibilità di gestire emergenze e features di utilizzo, quali reazioni a guasti o rotture di tubi, limitazioni del tempo di utilizzo, limitazioni temperatura massima acqua miscelata, monitoraggio consumi
- costi, dimensioni e consumi estremamente ridotti

Inoltre si sottolinea l'efficienza dei sensori piezoelettrici di portata, dotati di una dinamica molto veloce, che permette di avere prestazioni molto spinte, con l'utilizzo di un opportuno controllo, e una stima molto efficace della temperatura dell'acqua in uscita dal rubinetto, il cui andamento non potrebbe essere ugualmente monitorato da termocoppie, sensori dalla dinamica troppo lenta.

Tra gli svantaggi si ricordano, invece, l'ingombro e i costi non molto contenuti delle elettrovalvole proporzionali, che, proprio per questo motivo, si vorrebbero rimpiazzare, in un futuro progetto, con vitoni ceramici comandati da motori passo-passo, le cui prestazioni, per il

momento, sono state analizzate solo in via teorica.

Le prove sperimentali effettuate hanno permesso di valutare l'efficacia ma anche i limiti dei sistemi creati, tra cui spicca la scelta dei sensori di portata, molto efficienti, ma limitati a un range di funzionamento non sempre del tutto opportuno. Nel caso in cui, per garantire il soddisfacimento delle esigenze dell'utente, una delle due portate sia impostata a valori inferiori a 1 L/min, si incorre in oscillazioni talvolta anche importanti. Questo aspetto, probabilmente, può essere almeno in parte risolto rimpiazzando le molto sensibili elettrovalvole proporzionali, con più stabili vitoni ceramici. I test di confronto realizzati, infatti, ne evidenziano questa peculiarità.

Tra gli sviluppi futuri si prevede la realizzazione fisica del secondo prototipo, dotato di vitoni ceramici, che restituirebbe un apparato sperimentale meno costoso, più compatto e, probabilmente, stando ai risultati sperimentali ottenuti, più efficiente del primo. Inoltre si ipotizza l'eventuale utilizzo di un sensore di portata con range di funzionamento ottimale più ampio, pur assicurandosi di mantenere i costi ridotti.

Nell'ultima fase di lavoro, infine, si è posta l'attenzione sulla comunicazione wireless in ambiente Arduino, basata sugli standard IEEE 802.15.4 e ZigBee. La comunicazione wireless tra due o più schede Arduino e la loro programmazione senza fili, rese possibili dall'utilizzo di moduli XBee e opportune shield, permetterebbe di limitare al massimo gli ingombri e le difficoltà di collegamento delle differenti unità e, inoltre, di inserire il rubinetto elettronico all'interno di un sistema di monitoraggio dei consumi.

APPENDIX

APPENDICE

A.1 DATASHEET ELETTROVALVOLE PROPORZIONALI BÜRKERT

2 8 3 5

bürkert
FLUID CONTROL SYSTEMS

2/2-way proportional valve



- High sensitivity
- 0 to 25 bar
- DN 2 to 8 mm
- G 3/8 and G 1/2

Type 2835 can be combined with...



Type 8605
Digital control electronics
Cable plug version



Type 8605
Digital control electronics
DIN-rail version



Type 2508
Cable plug



Type 8611
Universal controller

The direct-acting proportional valve Type 2835 can be used as a control valve for process control and is suitable for technical vacuum. Low hysteresis, high repeatability and high sensitivity ensure superior regulation behavior. Thanks to an elastomeric sealing, the valve closes tightly and securely.

Circuit function A



Direct acting 2-way proportional valve, normally closed

Valve control takes place through the control electronics of Type 8605, which converts an analogue input signal into a PWM signal¹⁾.

Further functional features of the Type 8605 electronic control unit:

- Temperature compensation for coil heating by internal current regulation
- Simple zero and span settings
- Ramp function to dampen fast status changes

Technical Data - valve	
Body material	Brass, Stainless steel
Seal material	FKM, EPDM on request
Media	Neutral gases, liquids
Medium temperature	-10 ... +90 °C
Ambient temperature	max. +55 °C
Viscosity	max. 21 mm ² /s
Operating voltage	24 V DC
Power consumption	16 W
Duty cycle	100 % continuously rated
Port connection	G 3/8, G 1/2, NPT 3/8, NPT 1/2
Electric connection	Cable plug (DIN EN 175301-803 Form A)
Installation	As required, preferably with actuator in upright position
Typical control data ²⁾	
Hysteresis	< 5 %
Repeatability	< 0.25 % v. F.S.
Sensitivity	< 0.25 % v. F.S.
Turn-down ratio	1:100
Protection class - valve	IP65

¹⁾ PWM pulse-width modulation

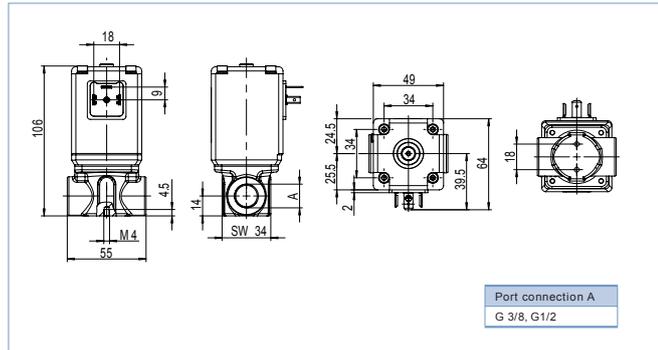
²⁾ Characteristic data of control behaviour depends on process conditions

DTS 100003965 EN Version: B Status: RL (released / freigegeben / valid) | printed: 30.07.2007



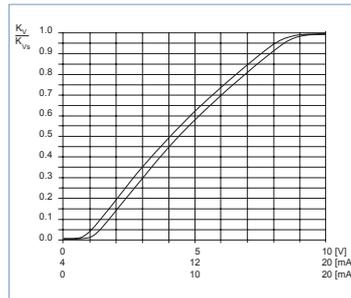
2 8 3 5

Dimensions [mm]



DTS 1000063985 EN Version: B Status: RL (released | freigegeben | valid) printed: 30.07.2007

Characteristics of a proportional valve



Advice for valve sizing

In continuous flow applications, the choice of appropriate valve size is much more important than with on/off valves. The optimum size should be selected such that the resulting flow in the system is not unnecessarily reduced by the valve. However, a sufficient part of the pressure drop should be taken across the valve even when it is fully opened.

recommended value: $\Delta p_{\text{valve}} > 30\%$ of total pressure drop within the system

For that reason take advantage of Bürkert competent engineering services during the planning phase!

Determination of the kv value

Pressure drop	kv value for liquids [m ³ /h]	kv value for gases [m ³ /h]
Subcritical $p_2 > \frac{p_1}{2}$	$= Q \sqrt{\frac{p}{1000 \Delta p}}$	$= \frac{Q_0}{514} \sqrt{\frac{T_1 p_2}{p_2 \Delta p}}$
Supercritical $p_2 < \frac{p_1}{2}$	$= Q \sqrt{\frac{p}{1000 \Delta p}}$	$= \frac{Q_0}{257 p_1} \sqrt{T_1 \rho_{N_2}}$

- k_v Flow coefficient [m³/h]¹⁾
 - Q Standard flow rate [m³/h]²⁾
 - p₁ Inlet pressure [bar]³⁾
 - p₂ Outlet pressure [bar]³⁾
 - Δp Differential pressure p₁-p₂ [bar]
 - ρ Density [kg/m³]
 - ρ_{N₂} Standard density [kg/m³]
 - T₁ Temperature of fluid medium [(273+t)⁴⁾K]
- ¹⁾ measured for water, Δp = 1 bar, via the device
²⁾ Standard conditions at 1.013 bar³⁾ and 0 °C (273K)
⁴⁾ Absolute pressure

28 35



Ordering chart for valves

All valves with FKM sealing

Circuit function	Orifice [mm]	Port connection	K _v value water [m ³ /h] ¹⁾	Q _N value [l/min] ²⁾	Maximum pressure [bar] ³⁾	Coil power consumption [W]	Maximum coil current [mA]	Item no. Brass body	Item no. Stainless steel body
A 2/2-way normally closed (NC) 	2 ⁴⁾	G 3/8	0.12	129	25	16	750	175 980	175 996
		NPT 3/8	0.12	129	25	16	750	175 997	175 998
	3	G 3/8	0.25	270	10	16	750	175 999	176 000
		NPT 3/8	0.25	270	10	16	750	176 001	176 002
	4	G 3/8	0.45	485	8	16	750	176 003	176 004
		NPT 3/8	0.45	485	8	16	750	175 995	175 984
		G 1/2	0.45	485	8	16	750	176 005	176 006
		NPT 1/2	0.45	485	8	16	750	175 985	175 986
	6	G 1/2	0.80	862	4	16	750	175 989	175 990
		NPT 1/2	0.80	862	4	16	750	175 993	175 994
	8	G 1/2	1.10	1186	2	16	750	178 794	179 412
		NPT 1/2	1.10	1186	2	16	750	179 305	179 306

¹⁾ K_v value: Flow rate value for water, measured at +20 °C and 1 bar pressure differential over a fully opened valve.
²⁾ Q_N value: Flow rate value for air with inlet pressure of 6 bar¹⁾, 1 bar pressure differential and +20 °C.
³⁾ Pressure data [bar]: Overpressure with respect to atmospheric pressure.
⁴⁾ For Δp>10bar it is possible to get discontinuities in the characteristic curve because of flow conditions in the application.
 Please note that the valves are delivered without control electronics unit and cable plug (see accessories below).

Further versions on request

- Materials
Seal: FFKM (resistant to aggressive media), EPDM
- Analytical
Oxygen version
Part oil-, fat- and silicon free

- Electrical connection
12 V coil
- Approvals
UL recognised, CSA, Ex version - II 2G EEx m IIC T4, PTB No. 02 ATEX 2094X with or without terminal box

Ordering chart for accessories

Cable plug Type 2508 according to DIN EN 175301-803 Form A
 The delivery of a cable plug includes the flat seal and fixing screw

Circuitry	Voltage / frequency	Item no.
None	0 - 250 V AC/DC	008 376
None, with 3 m cable	0 - 250 V AC/DC	783 573

Electronic Control Type 8605
 Please see Datasheet

DTS 100005985 EN Version: B Status: RL (released / freigegeben / valid) printed: 30.07.2007

A.2 DATASHEET VITONI NUOVA GALATRON

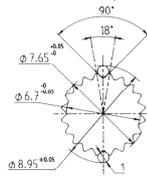


GN90^{1/2}-GN180^{1/2}

VITONE A DISCHI CERAMICI - CERAMIC DISC HEADWORK
TÊTE A DISQUES CERAMIQUES - OBERTEIL MIT KERAMIKSCHEIBEN
MONTURA CON DISCOS DE CERAMICA

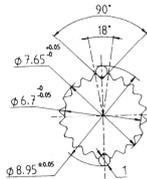
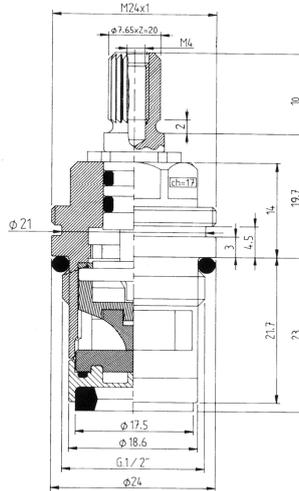


NUOVA GALATRON



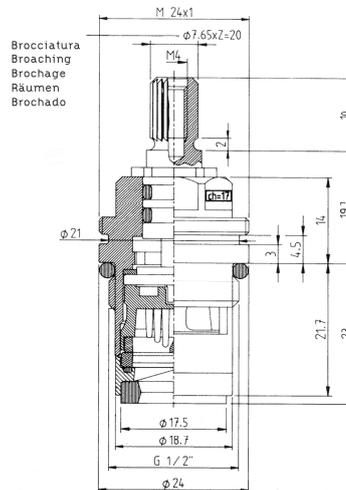
GN90DX
Codice ordinazione 216A01D00.00

GN90SX
Codice ordinazione 216A01S00.00



GN180DX
Codice ordinazione 216B01D00.00

GN180SX
Codice ordinazione 216B01S00.00



ATTENZIONE!
LAVORAZIONE DELLE SEDI
 Le migliori condizioni di utilizzo del vitone ceramico GN 90°/180° si ottengono con la sede n. 1. È consigliabile inoltre, se si vogliono ottenere buone portate, realizzare uno scarico Ø22 dopo il filetto come evidenziato nel disegno. È possibile l'impiego del vitone ceramico GN 90°/180° anche con la sede con scarico (vedi dis. n.2).

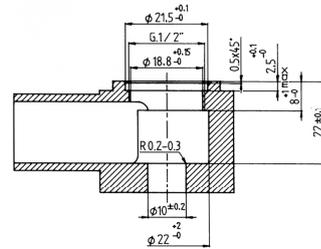
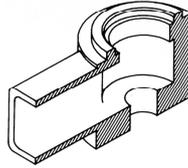
ATTENTION!
SEAT WORKING
 It is possible to get the best conditions for the use of GN 90°/180° headwork with seat n. 1. To obtain good deliveries we also recommend to make a Ø22 relief after the thread, as shown in the drawing. It is also possible to use GN 90°/180° ceramic headwork in seats worked in the same way as for headworks with waste (see drawing 2).

ATTENTION!
USINAGE DES SIEGES.
 On obtient les meilleures conditions d'utilisation de la tête ceramique GN 90°/180° avec la siége n. 1. Pour obtenir de bons debits, nous conseillons de pratiquer un degagement Ø22 apres le filet, comme l'on peut voir du dessin. L'emploi de la tête ceramique GN 90°/180° est aussi possible avec la siége usine pour des têtes avec vidage (voir dessin 2).

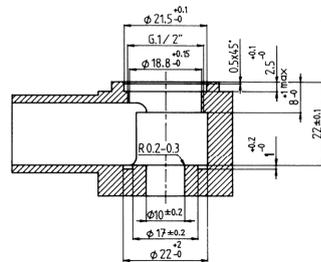
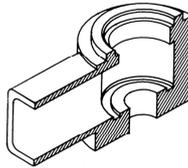
ACHTUNG!
VERARBEITUNG DER SITZE
 Die beste bedingungen für die ausnutzung des keramikoberteils GN 90°/180° werden mit dem sitz nr. 1 erzielt. Um gute leistungen zu erhalten, waere es ausserdem angebracht, eine unterschneidung Ø22 nach dem gewindegang anzubringen - wie in der zeichnung dargestellt. Die verwendung des keramikoberteils GN 90°/180° ist auch im sitz fuer obertheile mit abfluß moeglich (siehe zeichnung 2).

ATENCION!
ELABORACION DE LAS SEDES
 Las mejores condiciones para utilizar la montura ceramica GN 90°/180° se obtiene con la sede n. 1. Ademas se aconseja, si se desea obtener buenos caudales, realizar una descarga Ø22 despues de la rosca, como se indica en el diseño. Es posible el uso de la montura ceramica GN 90°/180° tambien con la sede trabajada para monturas fin descarga (Ver dis. n. 2).

1
 Sede piatta.
 Flat seat.
 Siege plat.
 Flacher sitz.
 Sede plana.



2
 Sede con scarico.
 Seat with waste.
 Siege avec vidage.
 Sitz mit Abfluß.
 Sede fin descarga.



SOLID PLAST e NUOVA GALATRON si riservano il diritto di riportare modifiche e miglioramenti tecnici in qualsiasi momento.	SOLID PLAST and NUOVA GALATRON reserve the right to change and improve on specifications at any time.	SOLID PLAST et NUOVA GALATRON se réservent le droit de reporter des modifications et des améliorations techniques à n'importe quel moment.	SOLID PLAST und NUOVA GALATRON behalten sich das Recht vor, jederzeit Änderungen und technische Besserungen durchzuführen.	SOLID PLAST y NUOVA GALATRON se reservan el derecho de modificar o adoptar mejoras técnicas en cualquier momento.
---	---	--	--	---

Diagramma della portata al variare della pressione da 0 a 5 Bar senza resistenza.

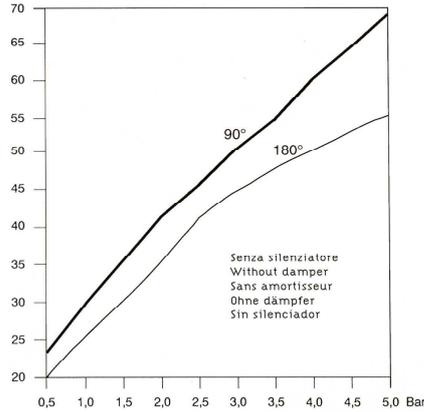
Diagram of flow rate at pressure varying from 0 to 5 Bars without resistance.

Diagramme de débit à la variation de la pression de 0 à 5 Bars sans résistance.

Durchflussdiagramme bei schwenkendem Druck von 0 bis 5 Bar ohne Widerstand.

Diagrama del caudal con variación de la presión de 0 a 5 Bar sin resistencia.

Litri al minuto.
Litres per minute.
Litres à la minute.
Liter / minute.
Litros por minuto.




Guarnizione normale di fondo.
Normal seal on the bottom.
Joint normal sur le fond.
Normale Dichtung auf dem Boden.
Junta normal de fondo.

Tabella della portata al variare della pressione da 0 a 5 Bar senza resistenza e senza silenziatore.

Table of flow rate at pressure varying from 0 to 5 Bars without resistance and without damper.

Tableau de débit à la variation de la pression de 0 à 5 Bars sans résistance et sans amortisseur.

Durchflusstabelle bei schwenkendem Druck von 0 bis 5 Bar ohne Widerstand ohne dämpfer.

Tabla del caudal con variación de la presión de 0 a 5 Bar sin resistencia y sin silenciador.

90°

Bar	0,5	1,0	1,5	2,0	2,5	3,0	3,5	4,0	4,5	5,0
lt/min	23	29,5	36	41,5	45,5	50,5	54,5	60	64	68,5

Tabella della portata al variare della pressione da 0 a 5 Bar senza resistenza e senza silenziatore.

Table of flow rate at pressure varying from 0 to 5 Bars without resistance and without damper.

Tableau de débit à la variation de la pression de 0 à 5 Bars sans résistance et sans amortisseur.

Durchflusstabelle bei schwenkendem Druck von 0 bis 5 Bar ohne Widerstand ohne dämpfer.

Tabla del caudal con variación de la presión de 0 a 5 Bar sin resistencia y sin silenciador.

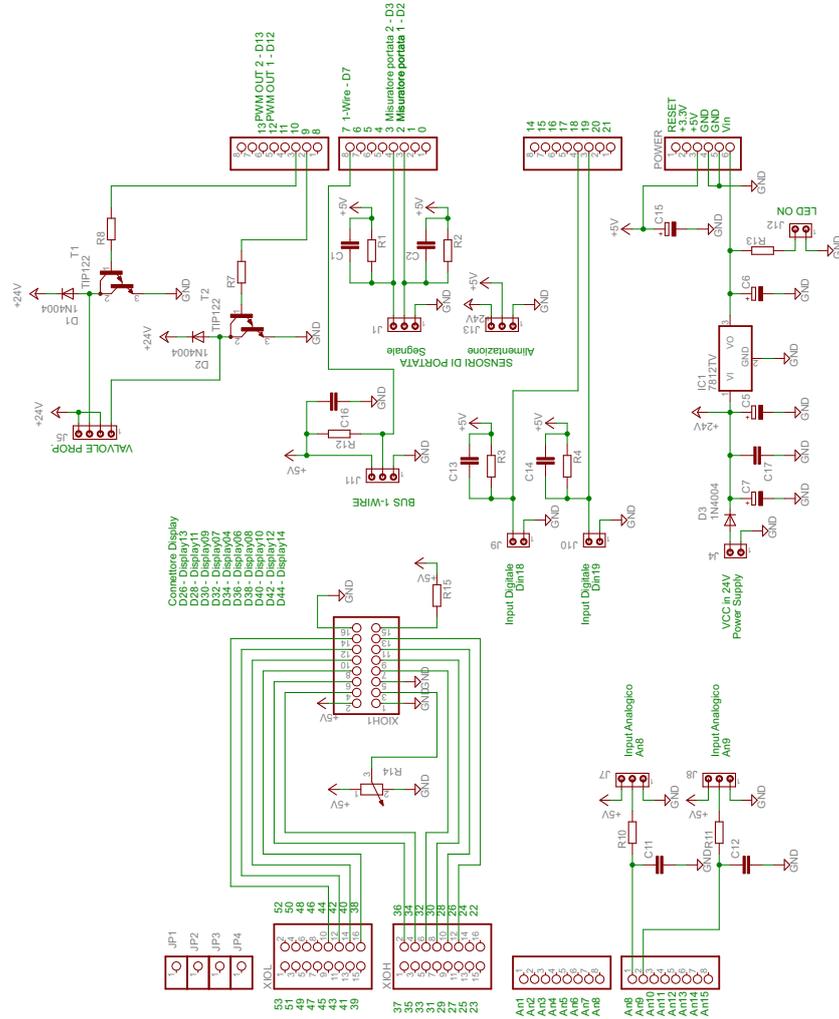
180°

Bar	0,5	1,0	1,5	2,0	2,5	3,0	3,5	4,0	4,5	5,0
lt/min	20	24	28,5	33,5	37	41,5	43	47	49,5	52

 **NUOVA GALATRON S.R.L.**
Via dell'Artigianato, 22
I - 46043 CASTIGLIONE DELLE STIVIERE (MN)
Tel. (0376) 630778 - Fax (0376) 632333
E-mail: nuovagalatron@dsrnet.it

 **SOLID PLAST S.p.A.**
UNDER GALATRON LICENSE
Via 8 Marzo, 3 - zona industriale Cornali
25013 CARPENEDOLO (BS) Italy
Tel. (030) 9965721 - Fax (030) 9998368
E-mail: solidplast@dsrnet.it

A.3 SCHEMA ELETTRICO SCHEDA DI GESTIONE INTEGRATA



ELENCO DELLE FIGURE

Figura 1.1	Esempio di miscelatore monocomando	3
Figura 1.2	Rappresentazione schematizzata dell'apparato sperimentale	5
Figura 2.1	Elettrovalvola proporzionale a due vie <i>bürkert</i>	7
Figura 2.2	Diagramma di una valvola proporzionale servoassistita	8
Figura 2.3	Sensore di portata <i>Liqui-View</i>	8
Figura 2.4	Analisi componenti del sensore di portata e loro funzioni	9
Figura 2.5	Sonda <i>MAXIM DS18S20</i> con modalità di funzionamento <i>attiva</i>	10
Figura 2.6	<i>DS18S20</i> Memory Map	11
Figura 2.7	Vitoni ceramici, GN90 e GN180, da 1/2 pollice	11
Figura 2.8	Schema funzionamento vitone	12
Figura 2.9	Diagramma della portata al variare della pressione da 0 a 5 bar	13
Figura 3.1	Esempio di scheda Arduino (Arduino UNO)	15
Figura 3.2	Esempio di screenshot dell'IDE di Arduino	16
Figura 3.3	Arduino Mega2560 Fronte	18
Figura 3.4	Schema scheda di gestione integrata	21
Figura 4.1	Rappresentazione schematizzata dell'apparato sperimentale	23
Figura 4.2	Case interfaccia utente	24
Figura 4.3	Case idraulico, vista dall'alto	25
Figura 4.4	Case idraulico, vista frontale	26
Figura 4.5	Foto del sistema fisico realizzato per il primo prototipo	26
Figura 4.6	<i>DS18S20</i> Memory Map	27
Figura 4.7	Struttura filtro a media mobile per lettura portate	29
Figura 4.8	Conteggio impulsi su finestra temporale di 1s	30
Figura 4.9	Schema elettrico collegamento valvola	30
Figura 4.10	Schema a blocchi del controllo relativo al primo prototipo	31
Figura 5.1	Schema blocco rubinetto con analisi perdite idrauliche	38
Figura 5.2	Modello Simulink realizzato per simulare il blocco rubinetto	41
Figura 5.3	Variazione di pressione da 0.5 a 5.0 Bar in un tempo di 10 s	44

- Figura 5.4 Diagramma della portata al variare della pressione da 0.5 a 5.0 Bar 44
- Figura 5.5 Modello Simulink per il sensore di temperatura 45
- Figura 5.6 Modello Simulink per il sensore di portata 45
- Figura 5.7 Schema Simulink relativo all'intero modello del sistema rubinetto 47
- Figura 6.1 Interfaccia Labview utilizzata per i test relativi al primo prototipo 50
- Figura 6.2 Variazione setpoint di temperatura con relativo inseguimento da parte della temperatura stimata 52
- Figura 6.3 Comportamento portate nelle due linee in corrispondenza di variazione setpoint temperatura e monitoraggio del relativo flusso totale ottenuto 53
- Figura 6.4 Variazione setpoint di portata con relativo inseguimento da parte della portata totale realmente ottenuta 54
- Figura 6.5 Andamento della temperatura in uscita stimata, al variare del setpoint di portata 54
- Figura 6.6 Andamento delle portate nelle due linee (calda e fredda), a fronte della variazione del setpoint di portata, ottenuta come riportato in Figura 6.4 55
- Figura 6.7 Andamento delle portate nelle due linee (calda e fredda), a fronte di una caduta di pressione istantanea dell'acqua calda 56
- Figura 6.8 Andamento delle portate nelle due linee (calda e fredda), a fronte di una caduta di pressione istantanea dell'acqua fredda 56
- Figura 6.9 Comportamento della portata calda in presenza di sbalzi di pressione dovuti all'apertura o chiusura di acqua calda in una fonte posta in prossimità a quella utilizzata 57
- Figura 6.10 Andamento della temperatura in corrispondenza di richiesta di acqua a temperatura inferiore/superiore al limite minimo/massimo possibile (T_c/T_h) 58
- Figura 6.11 Andamento della temperatura in uscita stimata, al variare del setpoint di temperatura, con $Q_{set} = 4, 10$ L/min 59
- Figura 6.12 Comportamento portate nelle due linee in corrispondenza di variazione setpoint temperatura, ottenuto, rispettivamente, con $Q_{set} = 4, 10$ L/min 60

Figura 6.13	Comportamento delle portate nelle due linee e totale in corrispondenza di variazione del setpoint di portata da $Q_{set} = 4$ L/min a $Q_{set} = 10$ L/min	61
Figura 6.14	Andamento della temperatura in uscita stimata, al variare del setpoint di portata da $Q_{set} = 4$ L/min a $Q_{set} = 10$ L/min	62
Figura 6.15	Variazione setpoint di pressione da 3 a 1 bar	63
Figura 6.16	Andamento della temperatura stimata in presenza di uno sbalzo di pressione del tipo di Figura 6.15	63
Figura 6.17	Comportamento della portata totale in presenza di uno sbalzo di pressione del tipo di Figura 6.15	64
Figura 6.18	Andamento delle portate nelle due linee (calda e fredda), a fronte di una caduta di pressione istantanea dell'acqua calda	65
Figura 6.19	Andamento delle portate nelle due linee (calda e fredda), a fronte di una caduta di pressione istantanea dell'acqua fredda	65
Figura 6.20	Andamento della temperatura in uscita stimata, al variare del setpoint di temperatura, nel primo prototipo	66
Figura 6.21	Andamento della temperatura in uscita stimata, al variare del setpoint di temperatura, nel secondo prototipo (modello)	67
Figura 6.22	Confronto andamento della temperatura in uscita stimata, al variare del setpoint di temperatura, nei due prototipi	67
Figura 6.23	Variazione setpoint di portata con relativo inseguimento da parte della portata totale realmente ottenuta con il primo prototipo	68
Figura 6.24	Variazione setpoint di portata con relativo inseguimento da parte della portata totale ottenuta con il secondo prototipo	69
Figura 7.1	Principali topologie di rete previste dallo standard IEEE 802.15.4	73
Figura 7.2	Esempio di comunicazione di tipo <i>mesh</i>	74
Figura 7.3	Esempio di modulo XBee	75
Figura 7.4	Schematizzazione della comunicazione tra due moduli XBee	75
Figura 7.5	XBee e XBee-PRO, front	77
Figura 7.6	XBee e XBee-PRO, side	77
Figura 7.7	Utilizzo di un partitore di tensione per l'adattamento dei livelli logici	78
Figura 7.8	XBee con antenna a chip e a filo	79

Figura 7.9	XBee con antenna esterna	79
Figura 7.10	Suddivisione <i>hardware</i> schematica delle diverse famiglie XBee	80
Figura 7.11	Suddivisione <i>software</i> schematica delle diverse famiglie XBee	80
Figura 7.12	XBee Serie 1, Low Power	82
Figura 7.13	Arduino XBee Shield	83
Figura 7.14	990.002 XBee USB Board	84
Figura 7.15	990.002 XBee USB Board - Layout	85
Figura 7.16	Rappresentazione schematizzata inserimento wireless nel progetto rubinetto	86
Figura 7.17	Schema elettrico collegamento linee di reset XBee e Arduino	90

ELENCO DELLE TABELLE

Tabella 2.1	Tabella riassuntiva caratteristiche principali sensori portata	9
Tabella 2.2	Tabella della portata al variare della pressione da 0 a 5 Bar	12
Tabella 3.1	Tabella riassuntiva caratteristiche principali Arduino Mega2560	18
Tabella 5.1	Confronto caratteristiche vitone reale (rosso) e suo modello (blu)	43
Tabella 6.1	Parametri PI scelti	51
Tabella 7.1	Tabella riassuntiva caratteristiche principali XBee Serie 1	82

BIBLIOGRAFIA

- [1] Bürkert Fluid Control Systems. *2/2-way proportional valve*, 2007.
- [2] OVAL Comporation. *Vortex Flow Monitor, Eggs DELTA Pulse From OVAL Liqui-View*, 2009.
- [3] Maxim Integrated Products. *DS18S20, High-Precision 1-Wire Digital Thermometer*, 2010.
- [4] Solid Plast s.p.a. Nuova Galatron S.R.L. *Vitoni ceramici GN90 1/2 - GN180 1/2*, 2010.
- [5] Arduino. *Arduino*, 2011.
- [6] *Idraulica* n.28, giugno 2005.
- [7] IEEE. *Advancing technology for humanity*, 2012.
- [8] ZigBee Alliance. *Zigbee alliance*, 2012.
- [9] MaxStream Digi International. *XBee/XBee-PRO OEM RF Modules*, 2005.
- [10] Ladyada. *Xbee radios, wireless arduino programming/serial link*, 2011.

