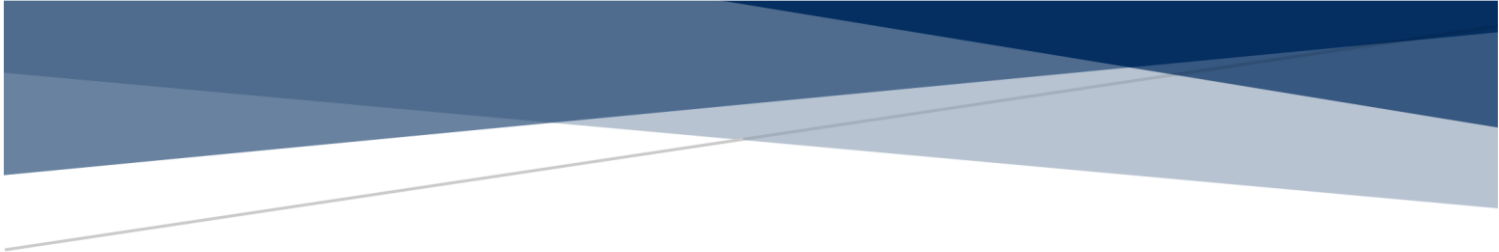




Rapporti Tecnici INAF INAF Technical Reports

Number	270
Publication Year	2023
Acceptance in OA@INAF	2023-03-10T13:27:08Z
Title	COLD HEAD MONITOR: Monitoraggio delle teste fredde
Authors	CABRAS, Alessandro, Roberto Caocci, Salvatore Garau, MAXIA, PAOLO
Affiliation of first author	O.A. Cagliari
Handle	http://hdl.handle.net/20.500.12386/34019 , https://doi.org/10.20371/INAF/TechRep/270



COLD HEAD MONITOR

MONITORAGGIO DELLE TESTE FREDDE

Alessandro Cabras¹, Roberto Caocci², Salvatore Garau³, Paolo Maxia¹

¹ INAF (Osservatorio Astronomico di Cagliari)

² Università di Pisa

³ Università di Cagliari

Revisione: Pierluigi Ortu, Tonino Pisanu

Sommario

Indice abbreviazioni.....	1
1 Introduzione	2
1.1 Funzionamento della testa fredda.....	4
2 Hardware	5
2.1 Architettura.....	5
2.2 Funzionamento generale	6
2.2.1 Board HALL V1	9
2.2.2 Board HALL CONDITION V1.....	9
2.3 Layout e PCB.....	10
3 Firmware.....	13
3.1 Setup	13
3.1.1 Inizializzazione dell'ADC	13
3.2 Loop.....	14
3.3 Diagramma di Flusso	16
3.4 Test del firmware	17
4 Software.....	18
4.1 Pannello connection.....	19
4.2 Pannello monitor info	19
4.3 Pannello plot.....	20
5 Conclusioni e sviluppi futuri.....	21
6 Bibliografia	22

Indice abbreviazioni

ADC (Analog to digital converter)

ARP (Address Resolution Protocol)

EEPROM (Electrically Erasable Programmable Read-Only Memory)

GAIA (Gain Attenuation Intelligent Amplifier)

LNA (Low noise amplifier)

I²C (Inter Integrated Circuit)

ICMP (Internet Control Message Protocol)

IGMP (Internet Group Management Protocol)

IPv4 (Internet Protocol version 4)

ISP (In-System Programming)

JTAG (Joint Test Action Group)

MCU (MicroController Unit)

PCB (Printed circuit board)

PFP (Primary Focus Position)

PPPoE (Point-to-Point Protocol over Ethernet)

PWM (Pulse-width modulation)

PSU (Power supply unit)

RFI (Radio Frequency Interference)

RISC (Reduced instruction set computer)

SPI (Serial Peripheral Interface)

TCP (Transmission Control Protocol)

UDP (User Datagram Protocol)

USART (Universal Synchronous-Asynchronous Receiver/Transmitter)

1 Introduzione

Un radiotelescopio è una struttura complessa progettata per la ricezione e la misura della potenza RF emessa dalle sorgenti radioastronomiche. Il segnale ricevuto è essenzialmente un rumore a larga banda le cui proprietà statistiche sono del tutto simili a quelle del rumore di fondo e del rumore generato dal sistema ricevente del radiotelescopio stesso.

I livelli di potenza dei segnali radioastronomici sono generalmente dell'ordine di $10^{-20} \div 10^{-26}$ W, questo comporta che uno dei requisiti fondamentali dei ricevitori installati in un radiotelescopio è la sensibilità.

La potenza totale disponibile dal "sistema radiotelescopio" può essere espressa in termini di potenza di rumore in base alla seguente espressione:

$$P_{sys} = P_A + P_{rec} \quad (1)$$

In virtù dell'equivalenza potenza-temperatura, sintetizzabile nell'espressione $P = kTB$ (dove k rappresenta la costante di Boltzmann, T la temperatura e B la banda del segnale ricevuto), l'equazione precedente può essere scritta come:

$$T_{sys} = T_A + T_{rec} \quad (2)$$

Nella (2) T_A rappresenta la temperatura d'antenna e dipende dalla direzione di puntamento (in elevazione) dell'antenna stessa: varia tipicamente da 25K per angoli di elevazione di 45° fino a valori di 90K per angoli di elevazione di 5° o prossimi all'orizzonte terrestre; T_{rec} indica la temperatura del ricevitore ed è determinata dalla temperatura dei vari componenti che fanno parte della catena ricevente. La T_{sys} indicata nella (2) è collegata alla sensibilità attraverso l'equazione del radiometro, che fornisce un'indicazione del rapporto segnale-rumore:

$$\frac{S}{N} = \frac{T_{src} \sqrt{\Delta_f} \tau}{T_{sys}} \quad (3)$$

dove T_{src} rappresenta la temperatura della radiosorgente osservata e τ il tempo di integrazione (o di osservazione). Poiché generalmente $T_{sys} \gg T_{src}$, nella (3) risulta di vitale importanza la riduzione dei contributi dovuti a T_A e T_{rec} . Tralasciando in questa sede la descrizione delle tecniche utilizzate per la riduzione di T_A , è importante notare che la temperatura di rumore T_{rec} del ricevitore, dipende dal numero di componenti presenti sulla catena ricevente: il suo valore è quantificabile tramite la formula di *Friis*, la quale afferma che la temperatura equivalente di rumore di un generico sistema composto da N dispositivi è uguale a:

$$T_{tot} = T_1 + T_2 G_1 + T_3 G_1 G_2 + \dots + T_N G_1 G_2 \cdot \dots \cdot G_{N-1}$$

Dove T_i, G_i (con $i = 1, \dots, N$) sono rispettivamente le temperature equivalenti di rumore ed i guadagni associati ai singoli componenti, che, nel nostro caso rappresentano i vari dispositivi facenti parte della catena ricevente. Poiché la temperatura di rumore di questi ultimi dipende anche dalla loro temperatura fisica, è di fondamentale importanza per il corretto funzionamento di un ricevitore radioastronomico provvedere al raffreddamento della catena stessa.

Per il raggiungimento delle temperature richieste vengono utilizzati dei criostati (Dewar) composti generalmente da due stadi: il primo lavora a temperature prossime a $10 \div 20\text{K}$ mentre il secondo, utilizzato per il raggiungimento delle temperature più basse, opera intorno ai 70K .

All'interno del dewar vengono posizionati i componenti della catena ricevente: la caratteristica più importante di un criostato è la sua capacità di refrigerazione e di minimizzazione del carico termico, quest'ultimo legato ai meccanismi di conduzione del calore (irraggiamento, conduzione e convezione). Il contenimento dell'irraggiamento viene ottenuto anche tramite l'utilizzo di materiali particolari come le super-insulation, mentre la conduzione viene ridotta limitando il più possibile il collegamento termico tra le parti del criostato raffreddate a 70 K e a 20 K . Per quanto riguarda invece il contributo dovuto alla convezione, esso viene sensibilmente ridotto tramite l'ottenimento (ed il mantenimento) del vuoto all'interno del dewar: l'eliminazione dell'aria impedisce che avvenga lo scambio termico dovuto alle correnti d'aria che si generano quando superfici adiacenti presentano temperature differenti.

In generale, un dewar criogenico può essere composto da: un sistema di monitoraggio della temperatura e del livello di vuoto raggiunto (vacuometro), un sistema preposto all'ottenimento del vuoto (il cui componente principale è la pompa da vuoto), un circuito di refrigerazione comprendente una testa fredda e il compressore del gas utilizzato per il raggiungimento delle basse temperature (generalmente elio) più la componentistica necessaria all'alimentazione elettrica.

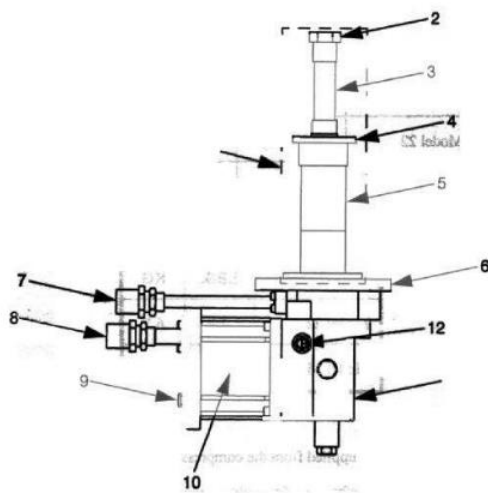
Per le motivazioni sopra esposte, la corretta manutenzione delle criopompe e dei compressori è fondamentale per il mantenimento delle prestazioni richieste nel tempo.

Lo scopo di questo progetto è quello di monitorare le correnti di alimentazione del motore della testa fredda: le parti meccaniche che lo compongono sono soggette ad usura e possono compromettere il funzionamento dell'intero sistema. Eventuali anomalie nelle correnti possono infatti permettere di prevedere in anticipo eventuali malfunzionamenti riducendo i tempi di inattività non programmati.

1.1 Funzionamento della testa fredda

La funzione di una testa fredda è quella di produrre una refrigerazione continua a ciclo chiuso a temperature che, a seconda del carico termico imposto, sono generalmente comprese tra 40K e 100K per il primo stadio e tra 4K e 20K per il secondo.

Il processo di raffreddamento (ciclo) delle teste fredde è noto come ciclo Gifford-McMahon ed è analogo a quello dei comuni frigoriferi domestici. In quest'ultimo, un fluido di lavoro (gas freon) viene compresso ed espanso per produrre un raffreddamento al di sotto della temperatura ambiente.



Legend

1. Cylinder
2. Second Stage Cold Station
3. Second Stage Cylinder
4. First Stage Cold Station
5. First Stage Cylinder
6. Top Flange
7. Helium Gas Return Connector (with dust cap)
8. Helium Gas Supply Connector (with dust cap)
9. Electrical Power Connector
10. Drive Motor
11. Crankcase (houses drive mechanism)
12. Pressure Relief Valve

Più nel dettaglio, la testa fredda è composta da (Figura 1): l'unità di azionamento, il cilindro e il gruppo dislocatore-rigeneratore, che si trova all'interno del cilindro. L'unità di azionamento è costituita dai seguenti sottogruppi: il motore di azionamento; il basamento e il meccanismo di azionamento, che si trova all'interno del basamento. L'unità di azionamento aziona il gruppo dislocatore-rigeneratore e controlla il flusso di elio in entrata e in uscita dalla testa fredda. Il motore impiegato è un motore ad azionamento diretto, a velocità costante, che può funzionare generalmente a 50 o 60 Hz. Ogni alloggiamento del motore ha tre connettori: uno è il connettore di alimentazione elettrica, attraverso il quale viene fornita l'alimentazione, mentre gli altri due sono rispettivamente il raccordo di mandata e di ritorno del gas elio. Funzionalmente, il gas elio ad alta pressione in arrivo dal compressore entra nella testa fredda attraverso il connettore di mandata del gas elio. Il gas passa quindi nel gruppo dislocatore-rigeneratore, defluisce nel carter, attraverso l'alloggiamento del motore e infine attraverso il connettore di ritorno del gas elio torna al compressore.

L'espansione del gas elio nel gruppo dislocatore-rigeneratore fornisce il raffreddamento nel primo e nel secondo stadio, ciascuno a temperature diverse.

Figura 1. Componenti principali di una Testa Fredda [1]

2 Hardware

2.1 Architettura

Il sistema di monitoraggio è stato studiato in modo tale che un suo eventuale malfunzionamento non comprometta l'operatività del sistema e non pregiudichi il suo *MTBF* (*Mean Time Between Failure*). Questo è garantito dal fatto che le tre alimentazioni necessarie per il motore trifase entrino nella Box e vengano intercettate senza subire variazioni o alterazioni dall'elettronica di elaborazione del segnale, pertanto il prototipo realizzato si tratta di un sistema Pass Through.

In Figura 2 viene mostrato l'inserimento della *Cold Head Monitoring* all'interno del sistema.

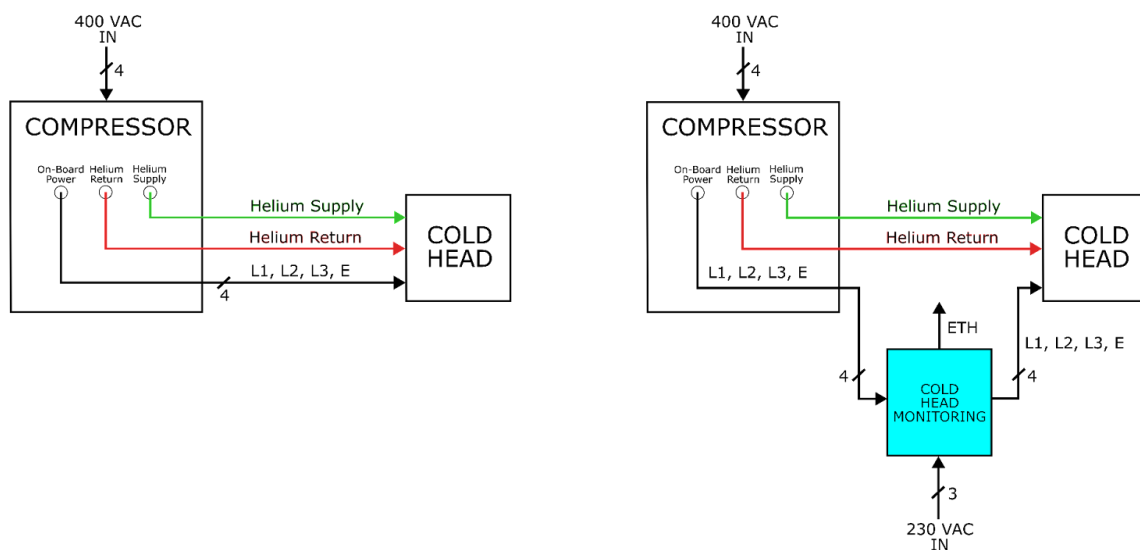


Figura 2. Sistema privo di CHM (sinistra), sistema con CHM (destra).

2.2 Funzionamento generale

Analizzando nel dettaglio il sistema di controllo di Figura 3, è possibile individuare diversi macro-blocchi:

- Una prima Board chiamata *HALL V1* nella quale si trovano i sensori ad effetto Hall.
- Una seconda Board chiamata *HALL MONITORING V1*, nella quale è presente lo stadio di amplificazione e l'acquisizione delle tensioni con l'ADC *ADS1015*.
- Un *Arduino LEONARDO* per la logica e per un'ulteriore acquisizione mediante i pin digitali.
- Alimentatore da 230 VAC a 5 VDC per l'alimentazione dei sensori, ADC e dell'Arduino stesso.

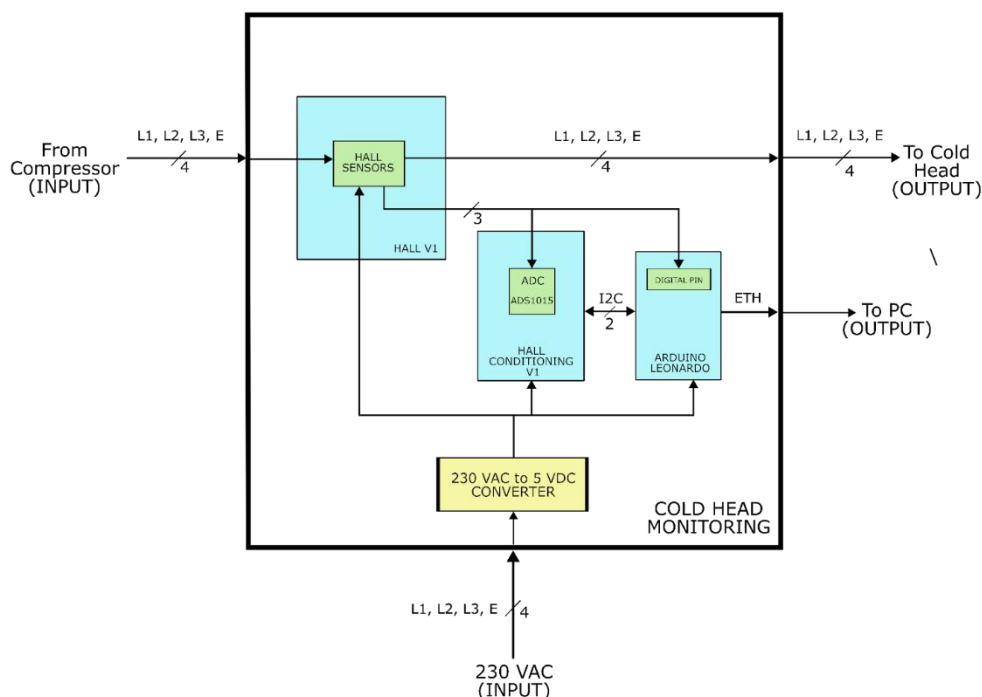


Figura 3. Schema a blocchi Cold Head Monitoring

Dal Compressore, oltre i tubi dell'Elio che vanno direttamente alla testa fredda (non riportati in Figura 3) arrivano le alimentazioni (L1, L2, L3) più la Terra (E); esse vengono intercettate dalla board *HALL V1* nella quale sono presenti i sensori ad effetto Hall, per poi essere mandate in uscita senza essere alterate.

Le uscite in tensione di questi sensori vengono prelevate e mandate sulla board *HALL CONDITIONING V1* nella quale è presente un primo ADC da 12 Bit della Adafruit [2] e parallelamente anche all'*Arduino Leonardo* [3], in particolare sui Pin Digitali 0, 1 e 7.

La motivazione dell'acquisizione in parallelo degli stessi dati deriva dalla necessità di poter campionare con un primo ADC le ampiezze delle tensioni proporzionali alle 3 correnti, mentre con i Pin Digitali il loro rispettivo sfasamento.

Queste tre schede (Sensori Hall, ADC, Arduino) necessitano di essere alimentate, ecco il perché della presenza di un Convertitore AC-DC.

Infine la visualizzazione, salvataggio ed eventuale post-processing del segnale avviene mediante un PC con il quale si acquisiscono i dati mediante protocollo Ethernet. Di seguito vengono mostrate alcune foto relative alla *Cold Head Monitoring* rispettivamente in Figura 4, Figura 5 e Figura 6.

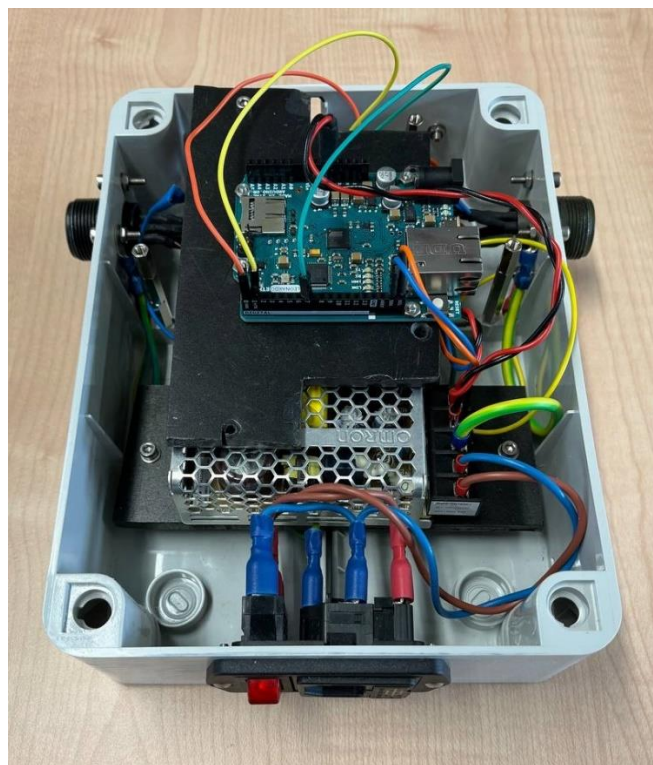


Figura 4. Interno della Box CHM.



Figura 5. Connettore alimentazione 220 V e connettore Ethernet, sopra e sotto rispettivamente.



Figura 6. Lato sinistro e lato destro box; connettori militari a 6 poli Maschio e Femmina rispettivamente.

2.2.1 Board HALL V1

In questa scheda, sono presenti i sensori e la rispettiva elettronica di filtraggio e di contorno per garantirne la corretta polarizzazione e quindi il corretto funzionamento.

Analizzando il funzionamento del sensore, si può comprendere come sia possibile avere un prototipo pass-through.

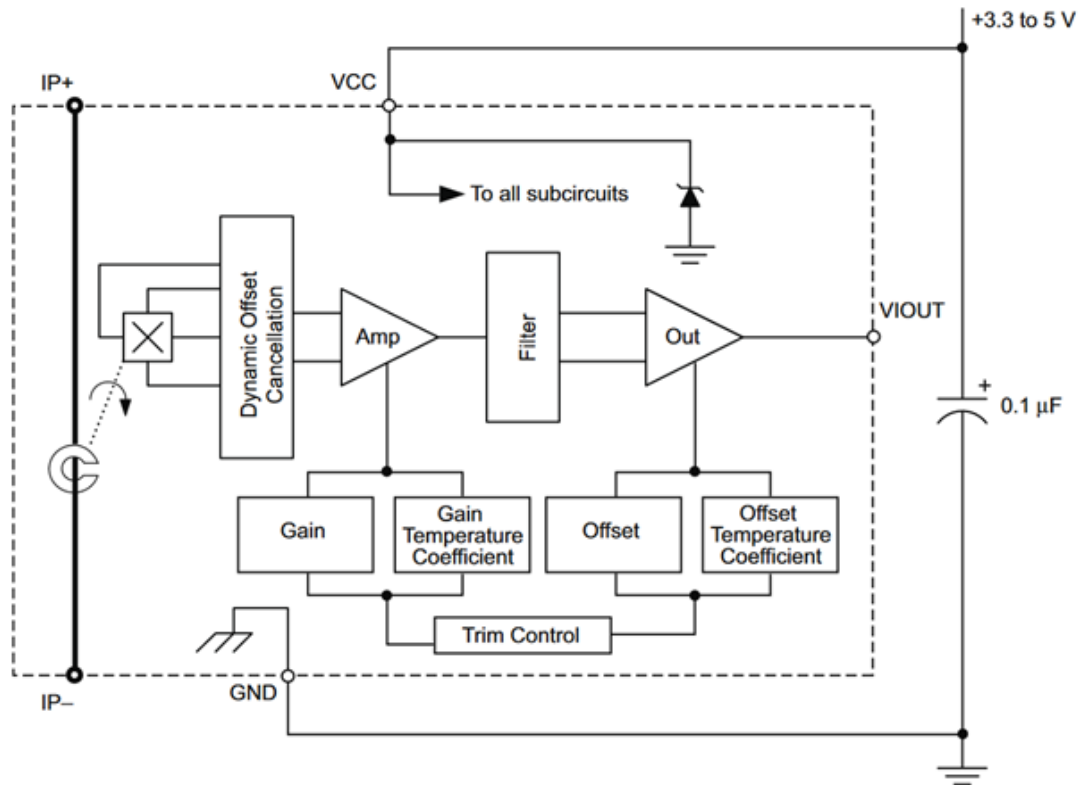


Figura 7. Schema a blocchi del sensore ACS758.

Dalla Figura 7 si nota come i terminali $IP+$ ed $IP-$, rispettivamente, ingresso ed uscita della generica alimentazione, siano fra loro collegati senza incontrare nulla lungo il cammino. Si nota che tale linea risulta essere una semplice pista di collegamento, permettendo di evitare effetti caricanti sulle alimentazioni della Testa Fredda.

2.2.2 Board HALL CONDITION V1

A seguito dell'intercettazione delle alimentazioni L1, L2 ed L3, i sensori ad effetto Hall forniscono in uscita una tensione proporzionale alla loro rispettiva corrente. Le tensioni ottenute vengono inviate sia all'ADS1015, sia all'Arduino. In questo modo, si ottengono informazioni differenti: mentre la *HALL CONDITIONING V1* viene utilizzata per trarre informazioni sulle ampiezze delle 3 fasi mediante l'ADC, l'Arduino invece, sfrutta i Pin Digitali per ottenere informazioni sulle fasi. Ciò è reso possibile sfruttando il fronte di salita del Pin Digitale e confrontando l'intervallo di tempo fra le tre sinusoidi in modo da determinarne lo sfasamento.

2.3 Layout e PCB

Dopo una prima fase di progetto, sono stati sviluppati gli schematici e successivamente si è passati alla fase di sbroglio del layout delle due schede.

Le schede sono state realizzate su 2 Layer e in Figura 8, Figura 9 e Figura 10 vengono riportati i rispettivi layout.

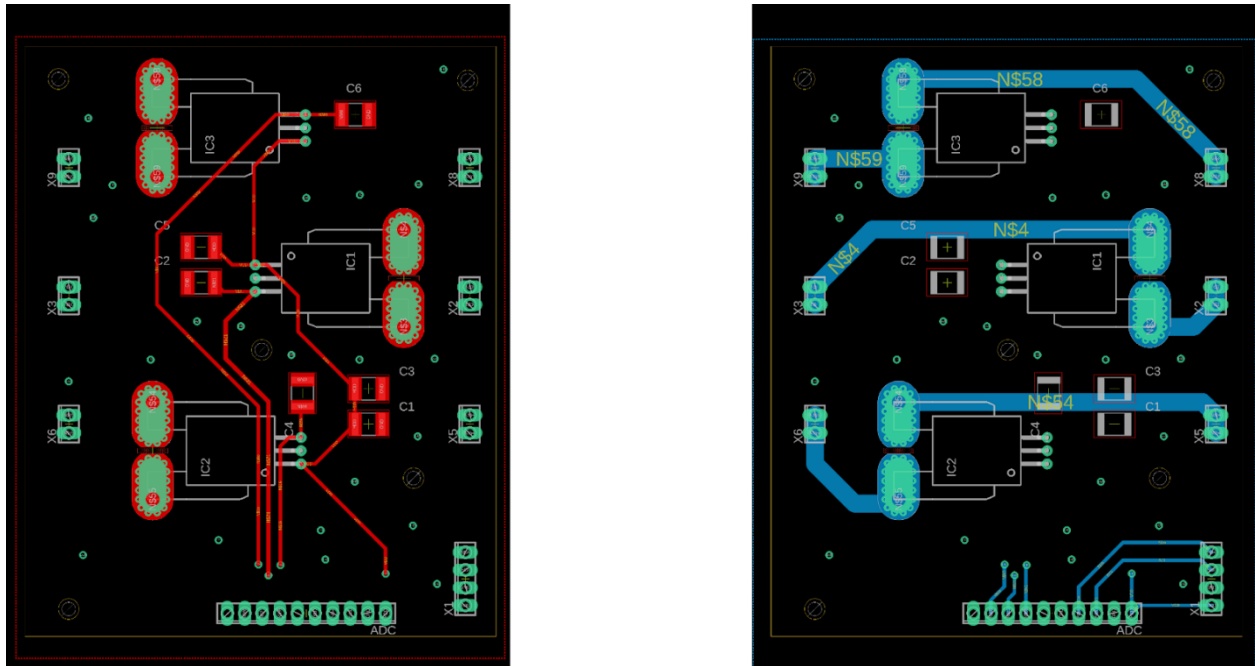


Figura 8. Layout board dei Sensori ad effetto Hall; Top e Bottom da sinistra a destra rispettivamente.

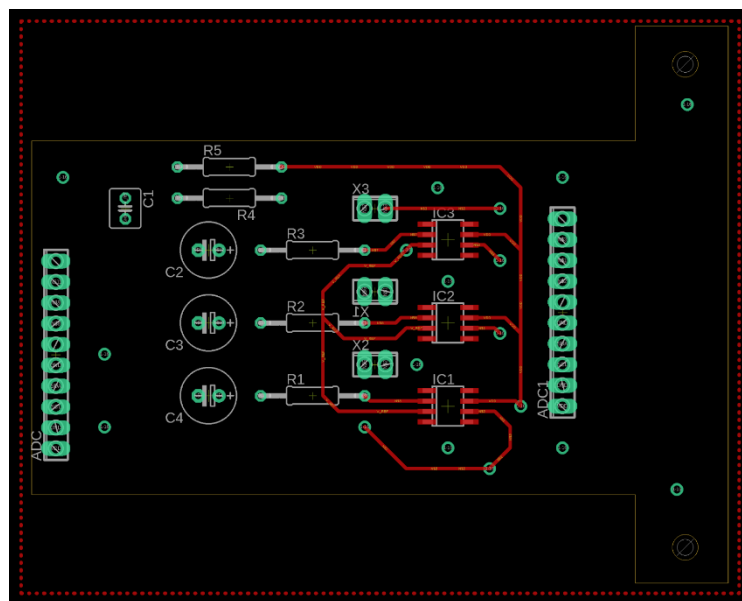


Figura 9. Layout scheda amplificazione e campionamento – Top View.

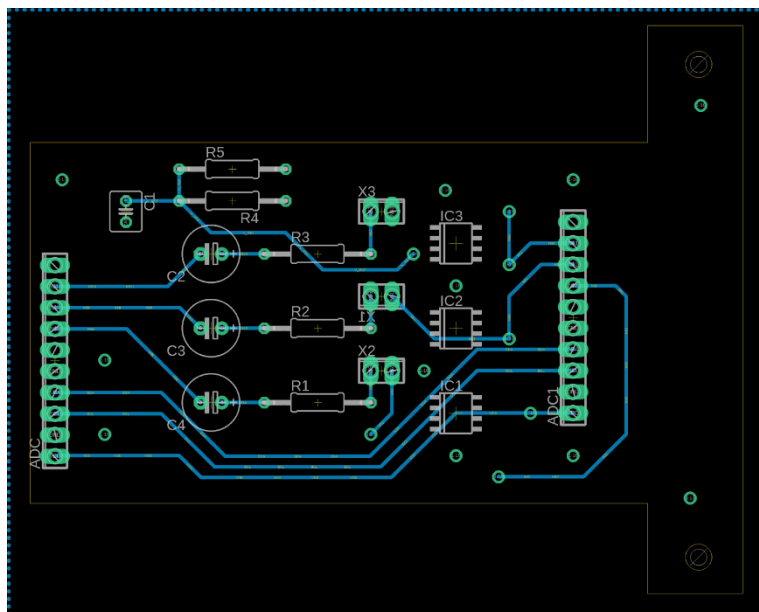


Figura 10. Layout scheda amplificazione e campionamento – Bottom View.

Dopo la fase di progettazione e design, si è passati alla fase di realizzazione del prototipo mediante LPKF e successivo montaggio dei componenti. In Figura 11e Figura 12, vengono mostrate le due board una volta ultimate.

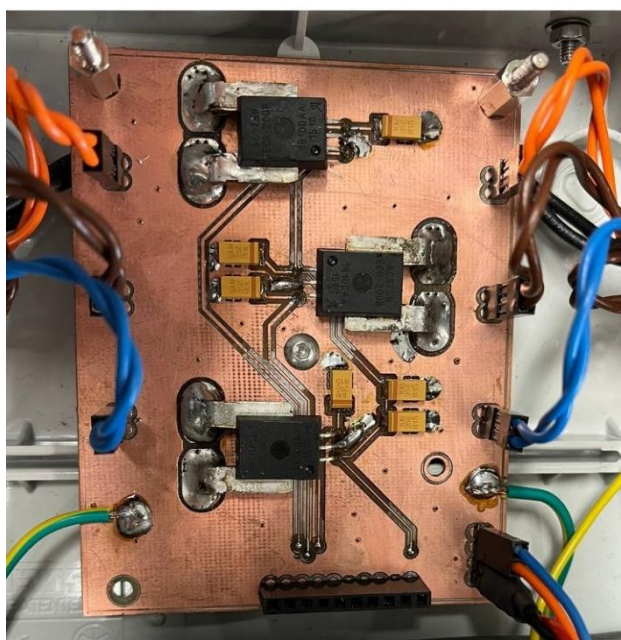


Figura 11. Board con sensori ad effetto Hall.

Inoltre, in Figura 11 si possono notare, sul lato sinistro le 3 Alimentazioni in ingresso mentre sul lato destro le 3 Alimentazioni in uscita.

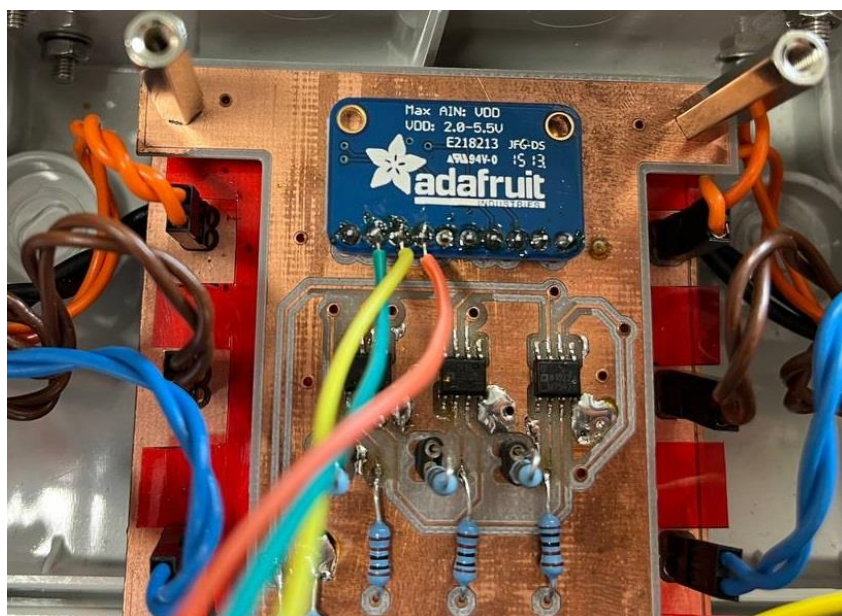


Figura 12. Board Amplificazione e Acquisizione.

3 Firmware

Il firmware è stato progettato e sviluppato per il microcontrollore Arduino Leonardo Ethernet in linguaggio C. I file sono suddivisi in un blocco principale, di cui viene riportato il diagramma di flusso Figura 13, e da tre file che gestiscono le funzioni di libreria che risultano essere, il controllo dell'ADC, il riconoscimento delle API e quello delle funzioni.

Nei paragrafi seguenti viene riportato il dettaglio dei blocchi che compongono il diagramma di flusso.

3.1 Setup

Il setup è suddiviso in due blocchi di istruzioni:

- **Setup Ethernet:** In questo blocco vengono inizializzati: il Mac Address del modulo Ethernet, l'IP statico e il numero di porta che dovrà essere interrogato dal client per instaurare la connessione con il dispositivo.
- **Inizializzazione:** Questo blocco contiene l'inizializzazione delle variabili globali, in particolare quelle dell'ADC (ad esempio `myADC_init()`), dei canali di ingresso, e l'inizializzazione degli interrupt usati per il calcolo delle differenze di fase tra i tre segnali.

Tale inizializzazione è stata implementata utilizzando la funzione:

`attachInterrupt(digitalPinToInterrupt(<pin>), rising_ch, RISING))`

la quale collega la funzione **interrupt service routine (ISR)** al pin specificato, indicando la tipologia dell'evento scatenante. Nel nostro caso è stato utilizzato il fronte di salita del segnale (RISING).

3.1.1 Inizializzazione dell'ADC

Per la gestione del convertitore AD viene utilizzata la libreria proprietaria Adafruit_ADS1X15 [4], che mette a disposizione tutte le funzioni necessarie alla corretta comunicazione con il modulo. L'oggetto ADC viene istanziato utilizzando la funzione `begin()`.

L'intervallo di ingresso all'ADC (o guadagno) dev'essere modificato attraverso la funzione `setGain(<gain>)`. Nel caso specifico, i valori in ingresso ricadono nell'intervallo 0÷5V e di conseguenza il guadagno è stato settato al valore `GAIN_TWOTHIRDS` che equivale a:

$$\frac{2}{3} \cdot Gain \pm 6.144 V$$

Dove:

$$1 \text{ bit} = 3 \text{ mV}$$

Il settaggio di un guadagno errato, con intervallo minore di tensione rispetto a quello di ingresso, ha come conseguenza la potenziale rottura del modulo ADC stesso.

Per ottenere la massima risoluzione nel segnale campionato e convertito, è stata impostata la velocità massima di campionamento disponibile per questo modello di convertitore. Utilizzando la funzione `setDataRate(<rate>)` è stato quindi settato un rate di 3300 SPS.

Il modulo ADC comunica con la MCU attraverso il protocollo I2C. Per poter raggiungere la velocità di campionamento desiderata è stata settata la frequenza massima di comunicazione, ovvero 400 KHz, utilizzando la funzione `Wire.setClock(<frequency>)`.

3.2 Loop

La funzione **loop** contiene il cuore del software, si ripetono ciclicamente le seguenti funzioni: ai fini della corretta comunicazione in rete dello strumento/prototipo, viene verificato che la connessione ethernet sia attiva, che il link fisico sia collegato, e che un client si sia connesso. Quando tali condizioni si verificano contemporaneamente, l'algoritmo ricerca nel pacchetto ricevuto la stringa di avvio lettura (API_START_CHAR) e successivamente esegue i tre blocchi seguenti:

- **Misurazione della differenza di fase:** Tale blocco legge i valori prodotti dalle ISR al verificarsi dell'evento di interrupt. Quando il segnale nei pin 0, 1 e 7 transita per il fronte di salita, il *timestamp* in microsecondi viene salvato in una variabile globale, attraverso la quale viene poi calcolato lo shift temporale tra i segnali. I valori temporali ottenuti vengono riportati in gradi, tenendo conto che la frequenza dei segnali è 50 Hz, dai quali si calcola un periodo di 20 ms si ottiene che la fase tra il primo e il secondo segnale è:

$$Phase\ Shift = 360^\circ \cdot \frac{(timestamp_1 - timestamp_2)}{\frac{50\ Hz}{10^6}}$$

- **Lettura delle tensioni attraverso l'ADC:** Per l'acquisizione dei dati viene utilizzata la libreria `Adafruit_ADS1X15` ed in particolare le funzioni:
 - `readADC_SingleEnded(pin)`: Tale funzione svolge tutte le operazioni necessarie per stabilire la comunicazione tra il master (MCU) e lo slave (ADS1015) sfruttando il protocollo I2C. La funzione riceve in ingresso il valore di uno dei 4 pin del modulo ADC. Il valore ottenuto si trova nell'intervallo compreso tra 0 e 4095 count per un ADC a 12 bit.
 - `computeVolts(adc)`: Tale funzione restituisce un valore di tensione proporzionale al valore dei count restituiti dalla funzione precedente. Tale conversione dipende dal valore dei guadagni impostati per i canali dell'ADC. Nel caso specifico, nei canali 2 e 3 è stato impostato un guadagno pari a 2/3x, pertanto risulta che 1 bit equivale ad una risoluzione minima di 3 mV. Il guadagno del canale 1 invece, poiché è stato ridotto via hardware attenuando di un mezzo il guadagno della cella front-end, viene riportato in scala via firmware moltiplicando per 2.
- **Invio della stringa:** Il firmware mette a disposizione un set di API mostrato in Tabella 1 per la comunicazione con il software di controllo. Nella tabella sottostante sono elencate le funzioni offerte dal firmware e la corrispondente risposta, si noti che ogni messaggio inizia con il carattere '#' e termina con il carattere '*'.

Codice API	Risposta
#IDN?*	Un stringa che rappresenta la versione del firmware usata (e.g. 1.0)
#ACQUIRE*	OK*
#GETADC adc_ch* Note: adc_ch è un valore tra 0 e 2	Una lista di lunghezza BUF_LEN contenente la sequenza dei valori grezzi letti dall'ADC per il canale scelto, separati da spazio. (e.g. 120 125 130 150 ... 257 312*)
#GETGAIN adc_ch* Note: adc_ch è un valore tra 0 e 2	Una stringa che rappresenta il guadagno di amplificazione per il canale specificato. (e.g. 2/3x*)
#GETSHIFT shift_code* Note: shift_code è un valore enumerato	Una stringa che rappresenta la differenza di fase tra i canali specificati, espressa in gradi. (e.g. 127*)

Tabella 1. API messe a disposizione dal firmware.

3.3 Diagramma di Flusso

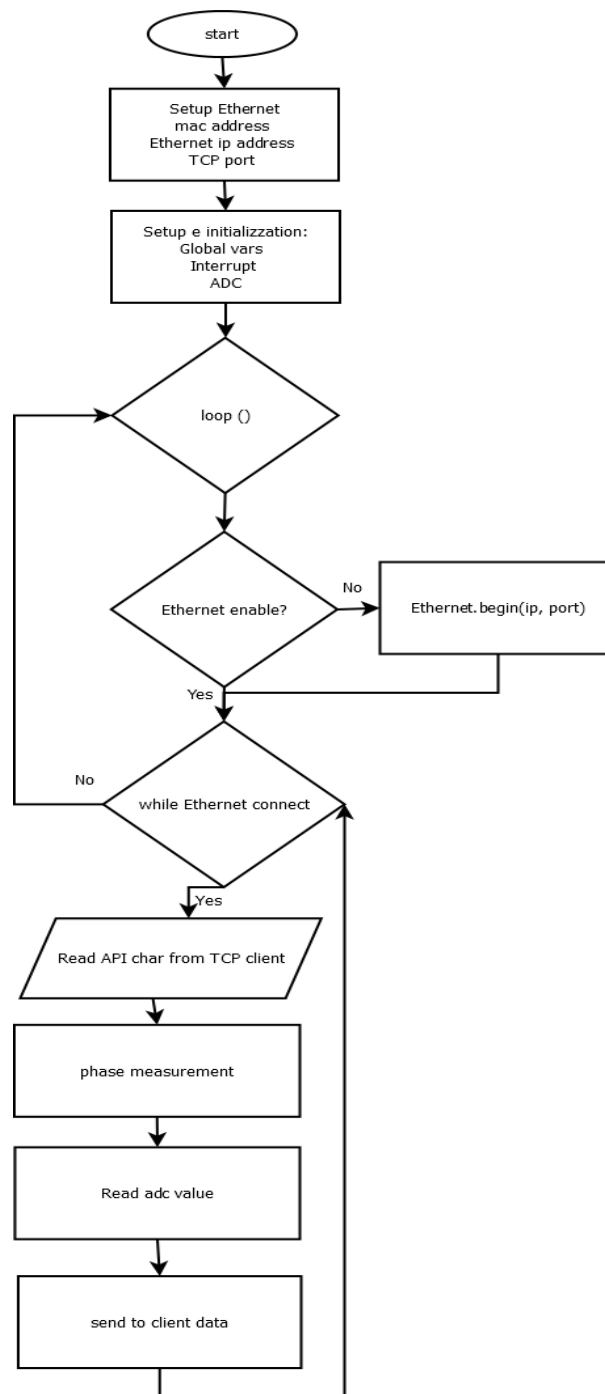


Figura 13. Diagramma di flusso del firmware.

3.4 Test del firmware

Durante lo sviluppo del codice, le funzioni sono state testate utilizzando inizialmente dei segnali stabili prodotti utilizzando il generatore di forme d'onda Siglent SDG6032X (Figura 14) [5]. Nel caso specifico sono state generate due sinusoidi, entrambe alla frequenza di 50 Hz, con ampiezza 5 Vpp, con un offset di 2.5 VDC. I due segnali sono stati sfasati di 120° .

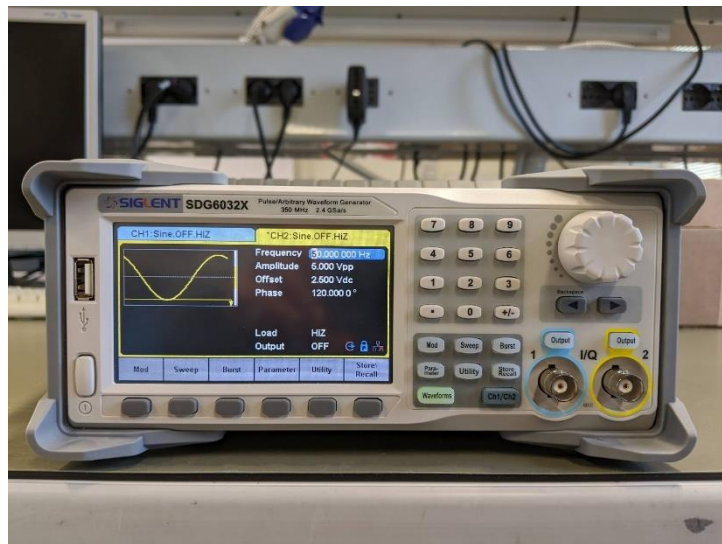


Figura 14. Generatore di funzioni Siglent SDG6032X

Dopo la fase di sviluppo il sistema è stato collegato al motore della testa fredda per valutare il funzionamento con i segnali reali. In parallelo, i segnali dei 3 canali, sono stati acquisiti utilizzando l'oscilloscopio Rohde & Schwarz RTO1044 (Figura 15) [6] al fine di valutare il corretto campionamento delle sinusoidi, gli sfasamenti e la stabilità della frequenza della linea. I valori di sfasamento tra i segnali presentano numerosi outlier, sia nelle letture eseguite dal firmware sia in quelle eseguite con l'oscilloscopio benché il valor medio sia rimasto molto vicino a quello reale.

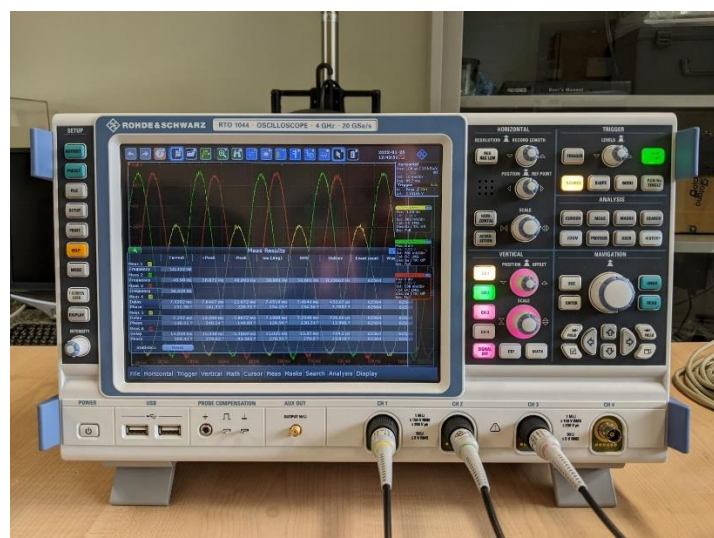


Figura 15. Oscilloscopio Rohde & Schwarz RTO1044

4 Software

Il monitoraggio e il controllo del sistema vengono effettuati tramite un software per PC che riceve i dati tramite scheda di rete, i pacchetti TCP vengono decodificati e viene estratto il codice di risposta inviato dal firmware, così come definito nella sezione precedente.

Il software è stato scritto usando il linguaggio Python e le librerie grafiche Qt ed è costituito da 3 moduli mostrati in Figura 16:

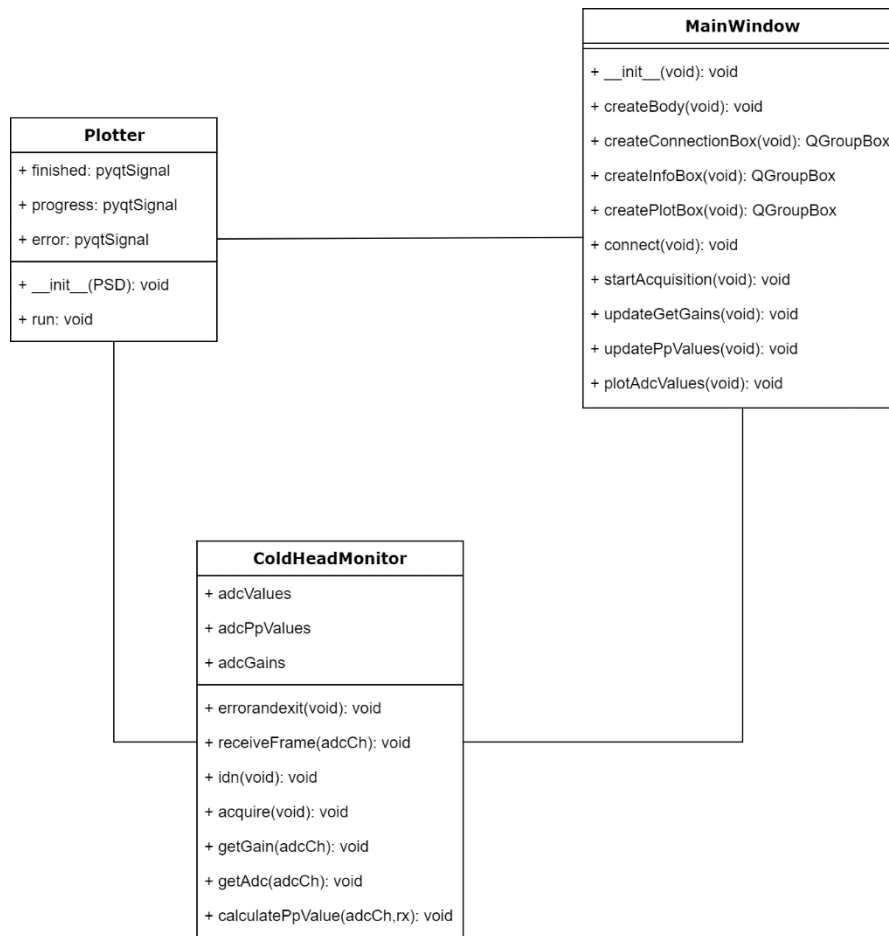


Figura 16. Diagramma UML del software.

La classe **ColdHeadMonitor** gestisce tutte le funzioni di comunicazione con il firmware, la decodifica dei pacchetti e si occupa del calcolo del valore picco-picco in ampere.

La classe **MainWindow** istanzia un oggetto di tipo **ColdHeadMonitor** e gestisce tutte le funzioni grafiche, il plot dei dati e l'aggiornamento delle statistiche.

La classe **Plotter** a sua volta, utilizzando la stessa istanza della classe **ColdHeadMonitor**, gira su un thread separato e si occupa dell'acquisizione continua dei dati dall'ADC.

L'interfaccia grafica è organizzata in 3 pannelli distinti, ciascuno di essi istanziato a livello software come membro pubblico della classe **MainWindow**.

4.1 Pannello connection

Attraverso il pannello di connessione mostrato in Figura 17 ci si può connettere al dispositivo specificando negli appositi campi, l'indirizzo IP e la porta definiti a livello firmware per il microcontrollore. In caso di connessione avvenuta, il bottone diventa di colore verde, mentre in caso contrario rimane di colore rosso e viene notificato un messaggio d'errore.

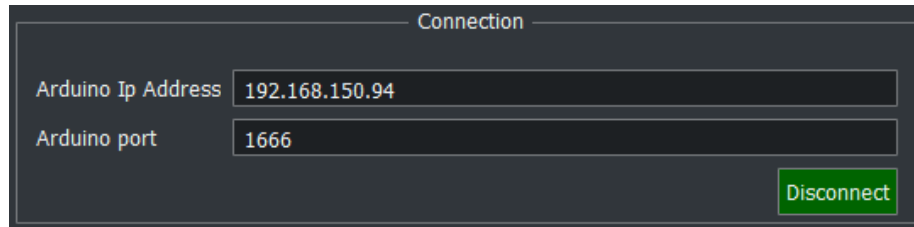
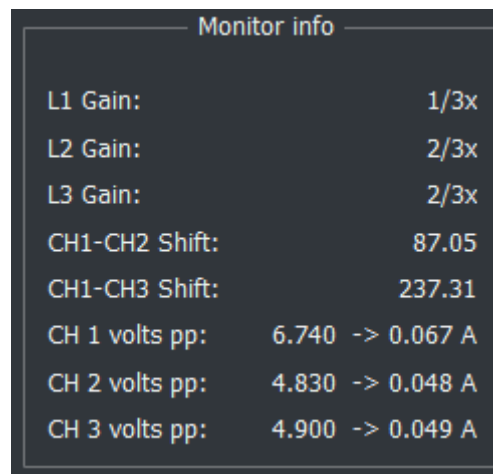


Figura 17. Interfaccia del pannello di connessione

4.2 Pannello monitor info

Nel pannello delle statistiche in Figura 18 vengono visualizzati:



Monitor info	
L1 Gain:	1/3x
L2 Gain:	2/3x
L3 Gain:	2/3x
CH1-CH2 Shift:	87.05
CH1-CH3 Shift:	237.31
CH 1 volts pp:	6.740 -> 0.067 A
CH 2 volts pp:	4.830 -> 0.048 A
CH 3 volts pp:	4.900 -> 0.049 A

Figura 18. Interfaccia del pannello informazioni

- I guadagni per i tre canali del convertitore AD;
- Lo sfasamento in gradi tra i segnali nei canali 1-2 e quello tra i canali 1-3;
- I valori convertiti dall'ADC per i 3 canali, espressi in Volt picco-picco.

I valori vengono aggiornati con cadenza dell'ordine dei secondi, in particolare i valori di tensione restituiti, vengono bufferizzati dal firmware ed inviati ogni volta che il buffer si riempie.

4.3 Pannello plot

Il pannello di visualizzazione contiene il grafico all'interno del quale vengono visualizzati in tempo reale i 3 segnali campionati dal convertitore. Ogni frame di visualizzazione, grafica l'intero buffer ricevuto dal microcontrollore, è perciò possibile che lo sfasamento tra le sinusoidi, non sia coerente con il valore indicato nel pannello delle statistiche.

L'interfaccia completa è visualizzata in Figura 19.

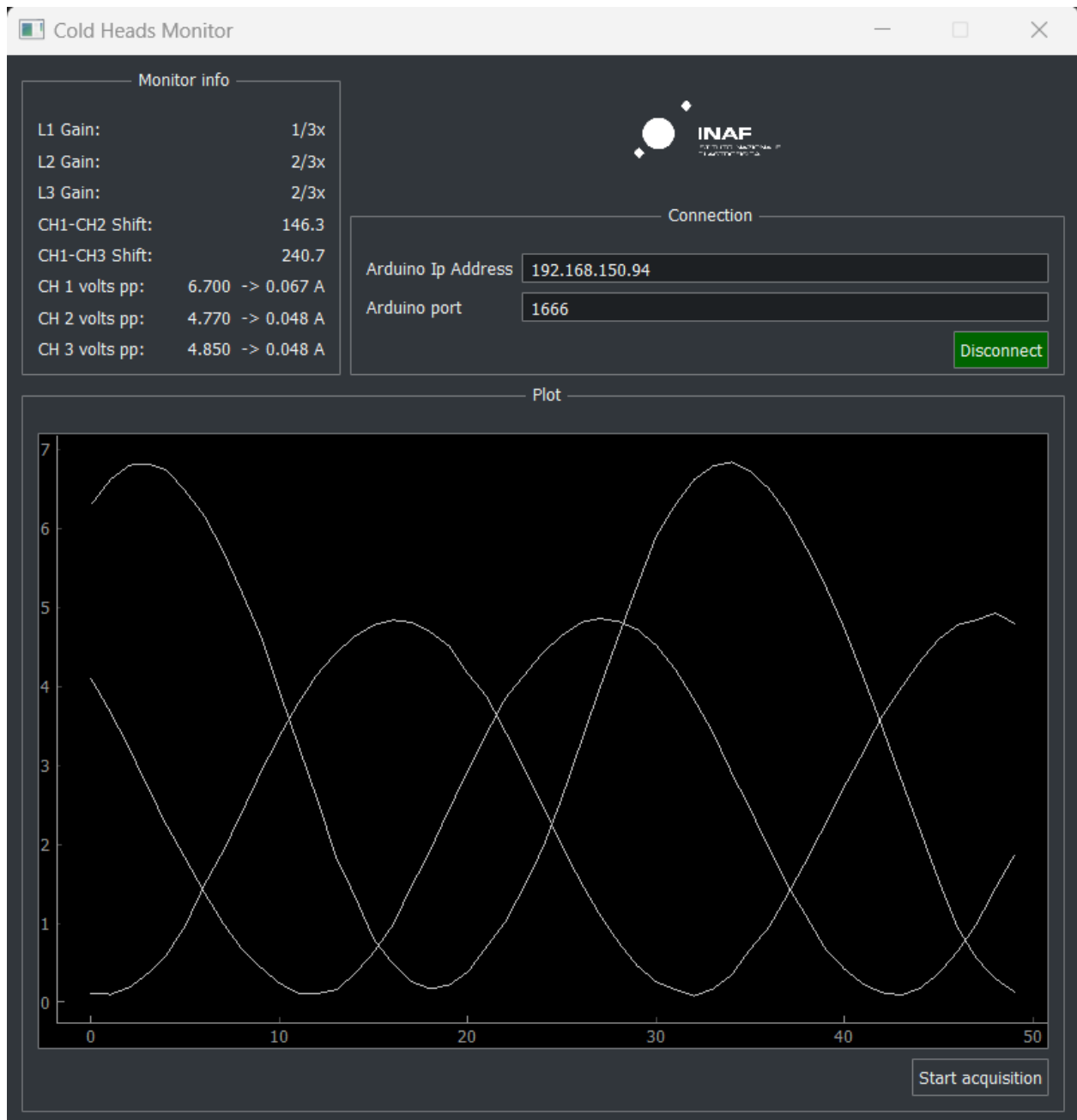


Figura 19. Interfaccia complessiva del software.

5 Conclusioni e sviluppi futuri

Per quanto riguarda la parte Hardware, essendo una prima versione del prototipo, risulta scarsamente ottimizzata in termini di ingombro, in quanto l'idea iniziale era creare un'unica board contenente sia i sensori che il primo ADC. La seconda versione verrà ottimizzata sotto tale aspetto, unendo tutte le board in un prodotto custom, il che permetterà di ridurre notevolmente le dimensioni dell'intera box, ottenendo un prodotto molto più compatto e funzionale.

In questa versione del prototipo è fondamentale che un operatore controlli regolarmente i dati ottenuti poiché al momento il sistema non notifica eventuali malfunzionamenti.

Per questo motivo, nelle successive versioni verrà implementato un sistema di manutenzione predittiva. La manutenzione predittiva è una tecnologia avanzata che utilizza algoritmi di apprendimento automatico per analizzare i dati raccolti dal sistema e prevedere eventuali problemi prima che si verifichino. Questo significa che il sistema può essere controllato e mantenuto in modo proattivo, prima che si verifichino problemi, in modo da garantire che questo funzioni correttamente.

L'importanza della manutenzione predittiva non può essere sottovalutata. Questa tecnologia consente di ridurre i tempi di inattività del sistema riducendo il MTBF, migliorare la disponibilità dei dati e ridurre i costi di manutenzione.

6 Bibliografia

[1] Multiple uses of model 22C/350C Cryodine refrigerators, 1995 CTI-Cryogenics - Helix Technology Corporation

[2] <https://www.adafruit.com/product/1083>

[3] <https://docs.arduino.cc/retired/boards/arduino-leonardo-eth>

[4] https://github.com/adafruit/Adafruit_ADS1X15

[5] <https://www.siglent.eu/product/1138544/siglent-sdg6032x-350mhz-function-arbitrary-waveform-generator>

[6] https://scdn.rohde-schwarz.com/ur/pws/dl_downloads/dl_common_library/dl_brochures_and_datasheets/pdf_1/RTO_bro_en_5214-2327-12_v2201.pdf