



---

# Controllo PID di temperatura di una camera climatica tramite scheda NUCLEO con supervisione in LabVIEW

---

Anno Accademico 2019/2020

Progetto di **Sensors and Transducers**

**Prof. Ing. Francesco Adamo**

Membri del gruppo:

D'ALESSANDRO VITO IVANO

PACE NICOLA

PACIOLLA FRANCESCO

SOMMA EMANUELE

13 febbraio 2020

Corso di Laurea Magistrale in Ingegneria dell'Automazione

# Sommario

Progetto Camera Climatica.....	2
Strumenti utilizzati.....	2
Realizzazione dei collegamenti del sistema “camera climatica” .....	5
Resistori di potenza presenti nella camera climatica.....	6
Modulo programmabile .....	8
Partitore di tensione.....	9
Circuito di condizionamento del segnale di ingresso del dimmer elettronico .....	9
Circuito di condizionamento della ventola.....	11
Dimensionamento del filtro RC .....	13
Realizzazione del circuito realizzato su breadboard in Altium Designer.....	20
Codice Mbed.....	22
Main.cpp.....	22
PID_Int.h.....	26
PID_Def.h .....	27
Codice LabVIEW .....	28
Block Diagram.....	29
Event Structure.....	31
Front Panel.....	32
Teoria sul controllo di temperatura mediante regolatore PID .....	33
Proporzionale.....	34
Integrativo .....	34
Derivativo.....	35
Taratura del PID mediante metodo di Ziegler-Nichols .....	36
Teoria sulle termocoppie .....	37
Nozioni generali .....	37
Compensazione del “giunto freddo”.....	38
Legge dei metalli interposti.....	39
Compensazione Hardware.....	40
Compensazione software.....	41
NIST: Coefficienti polinomiale per la conversione Tensione-Temperatura .....	42
NIST: Coefficienti polinomiali per la conversione Temperatura-Tensione.....	42

# Progetto Camera Climatica

Il fine del seguente progetto è quello di controllare la temperatura di una camera climatica mediante controllo PID. La camera climatica è costituita da una scatola in legno contenente due resistori collegati in serie dal valore di  $100\ \Omega$  e due ventole, una sul lato inferiore della scatola per permettere il ricircolo dell'aria in modo da ottenere un valore di temperatura omogeneo nella scatola stessa ed una sul lato sinistro per il raffreddamento della camera climatica nel caso di superamento della soglia di temperatura imposta. Inoltre, mediante un foro presente sul lato della scatola è stato possibile inserire una termocoppia di tipo J per monitorare la temperatura all'interno della camera climatica.

Il controllo PID verrà implementato su una scheda Nucleo STM32-F401RE mediante l'utilizzo della libreria integrata nel compilatore Mbed.

## Strumenti utilizzati

Gli strumenti utilizzati per la realizzazione del progetto sono:

- Dimmer elettronico
- Scheda Nucleo **STM32-F401RE**
- Multimetro da banco digitale
- Oscilloscopio
- Alimentatore da banco
- Termocoppia di tipo J
- 2 resistenze di potenza di  $100\ \Omega$
- 1 MOSFET a canale N **FQP20N06**
- 1 integrato **LM358N**
- 1 resistenza di  $8.2\ k\Omega$
- 1 resistenza di  $1.2\ k\Omega$
- 1 resistenza di  $22\ k\Omega$
- 2 resistenze di  $1\ k\Omega$
- 1 resistenza di  $2.7\ k\Omega$
- Pinza amperometrica

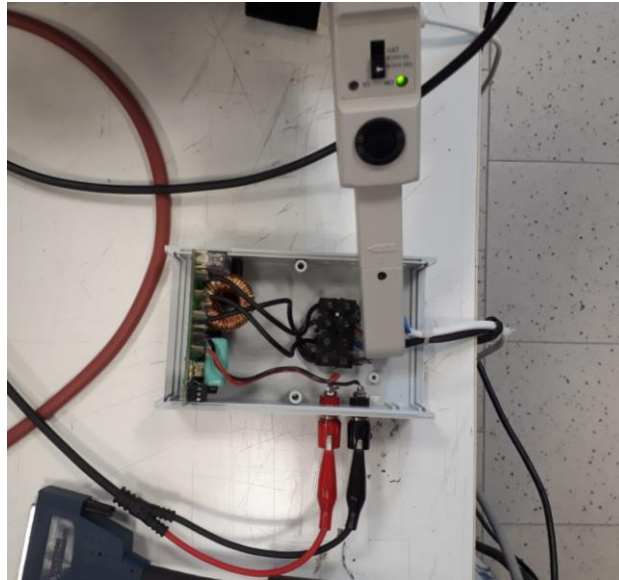
## Software utilizzati

I software utilizzati per la realizzazione del progetto sono:

- TeraTerm
- Mbed
- LabView
- MATLAB
- Altium Designer

## Fase preliminare di test dei componenti hardware: caratterizzazione del dimmer elettronico mediante MATLAB

Nella fase preliminare di test dei componenti hardware del sistema la camera climatica è stata collegata ad un dimmer elettronico, collegato a sua volta ad un generatore di funzione, utilizzato per fornire una tensione di alimentazione continua al dimmer nel range operativo 0 V – 10 V. Mediante una pinza amperometrica (settata nel range di misura 100 mV/A) connessa all'oscilloscopio mediante cavo BNC si è potuto osservare la forma d'onda della corrente fornita alle resistenze di potenza della camera climatica: partendo da una tensione di alimentazione pari a 0 V, si è osservata una forma d'onda della corrente di uscita del dimmer di tipo impulsiva; successivamente si è incrementata ciclicamente la tensione di ingresso con uno step di 1 V per verificare che all'aumentare della tensione si visualizza una forma d'onda sempre più simile ad una forma d'onda sinusoidale; infine imponendo una tensione di alimentazione di 10 V si è osservati sull'oscilloscopio una forma d'onda della corrente perfettamente sinusoidale, avendo raggiunto il limite superiore del range di tensione del dimmer. Tale procedimento è servito per verificare il corretto funzionamento del dimmer elettronico come potenziometro.



Per ottenere la curva caratteristica del dimmer si è pilotati da remoto sia il generatore di funzione AGILENT sia il multimetro da banco mediante il seguente script di MATLAB:

```
delete(instrfind);

FGEN = visa('ni', 'GPIB0::2::INSTR'); %definiamo il generatore di funzione
DMM = visa('ni', 'GPIB0::3::INSTR'); %definiamo il multimetro utilizzato per le misure di tensione

fopen(FGEN); %accendiamo il generatore di funzione
fopen(DMM); %accendiamo il multimetro
query(FGEN, '*idn?')
query(DMM, '*idn?')

N = 100; %numero di campioni di tensione di ingresso
%Definiamo il range di tensione di ingresso permesso dal dimmer elettronico
Vmin = 0; %limite inferiore del range di tensione
Vmax = 10; %limite superiore del range di tensione
Vinput = linspace(Vmin, Vmax, N); %definiamo la tensione di alimentazione in ingresso al dimmer
```

```

Vout = NaN*zeros(size(Vinput)); %inizializziamo il vettore che conterrà i valori misurati di
tensione in uscita dal dimmer mediante multimetro AGILENT

fprintf(FGEN, 'APPLY:DC DEF,DEF,0');
for k = 1:N
    comm = sprintf('APPLY:DC DEF,DEF,%6.2f',Vinput(k));
    fprintf(FGEN, comm);
    pause(0.1);
    Vout(k) = query(DMM, 'MEAS:VOLT:AC?', '%s', '%f');
    plot(Vinput, Vout, 'o-');
    hold on
end
title('Curva caratteristica del dimmer');

fprintf(FGEN, 'APPLY:DC DEF,DEF,0');

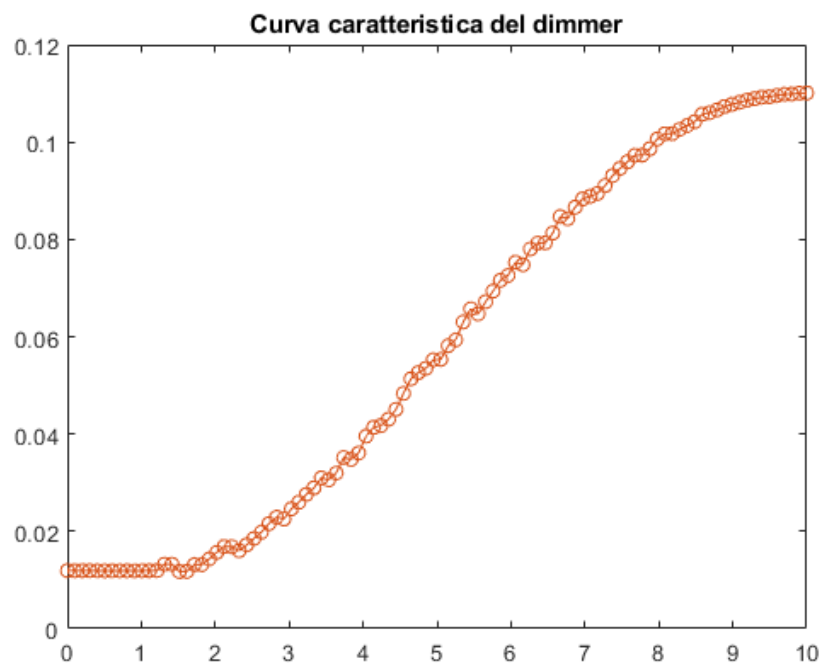
fclose(FGEN); %spegniamo il generatore di funzione
fclose(DMM); %spegniamo il multimetro

```

Il precedente script fornisce come output la curva caratteristica del dimmer (I - V) avendo in ingresso la tensione di ingresso del dimmer (nel range 0 V – 10 V).

Per ottenere una curva caratteristica del dimmer che descrivi in modo corretto il funzionamento del dispositivo elettronico sono stati considerati 100 valori di tensione nel range operativo di tensione del dimmer.

L'output ottenuto da tale script MATLAB è un grafico sul cui asse x sono presenti i valori di tensione dati in ingresso al dimmer tramite generatore di funzione e sul cui asse y sono definiti i valori di corrente alternata che fornisce in output il dimmer misurati tramite pinza amperometrica. Tale grafico è mostrato nella seguente figura:



Da tale grafico si osserva che il dimmer elettronico ha un comportamento lineare soltanto per valori di tensione centrati nel range operativo 0 V – 10 V mentre ha un comportamento non lineare (entra in saturazione) per valori di tensione prossimi ai limiti inferiore e superiore di funzionamento del dimmer stesso.

## Realizzazione dei collegamenti del sistema “camera climatica”

La misura di temperatura all'interno della camera climatica è effettuata mediante l'utilizzo di una termocoppia di tipo J inserita mediante un foro presente sul lato destro e fissata utilizzando due supporti in gomma. Successivamente la termocoppia è stata collegata al modulo di programmazione MCR-T-UI-2814090, configurabile per termocoppie e termoresistenze con tecnica di collegamento a 2,3 o 4 conduttori, alimentato dall'alimentatore DPP30-24 che fornisce una tensione di 24 V in DC. Uno dei più importanti vantaggi nell'utilizzare questo modulo di programmazione è quello di avere la possibilità di scegliere il range del segnale in uscita tra quelli presenti: nel caso in esame è stato utilizzato un range di 0 V – 10 V. Questo modulo è in grado di operare tra i -20°C e i 65°C ed il collegamento del sensore avviene mediante morsettiera a vite estraibile.

Inoltre, un ulteriore vantaggio del modulo programmabile è quello di poter applicare una compensazione via software della termocoppia utilizzato in modo da eliminare l'errore di misura della termocoppia nel caso in cui il giunto freddo non sia ad una temperatura pari a 0°C.

Poiché il modulo è stato settato per fornire in uscita tensioni nel range 0 V – 10 V mentre la scheda nucleo accetta tensioni in ingresso ad un pin analogico comprese tra 0 V – 3.3 V è stato necessario realizzare un partitore di tensione costituito da 3 resistenze con stesso valore nominale: in questo modo, fornendo in ingresso al pin analogico della scheda la tensione che sussiste su una delle 3 resistenze si ottiene una tensione compresa nel range di ingresso della scheda.

Il valore di tensione fornito in ingresso al pin analogico della scheda è convertito nel corrispondente valore di temperatura mediante un algoritmo implementato sulla piattaforma Mbed che utilizza le tabelle di conversione da tensione a temperatura riportate nel datasheet della termocoppia (“Nist Coefficient”). La funzione che asserisce a tale scopo è denominata `getTemperature_thermocouple` e sarà descritta successivamente.

Oltre l'algoritmo di conversione da tensione a temperatura, è stato implementato il regolatore PID che permette di controllare la temperatura misurata dalla termocoppia nella camera climatica, definendo i duty cycle dei segnali PWM ottenuti dai pin PWM della scheda nucleo (nel caso in esame i pin PWM utilizzati sono PB3, da cui otteniamo il segnale PWM da fornire in ingresso al dimmer e PB4, da cui otteniamo l'altro segnale PWM per l'accensione della ventola): in particolar modo, quando la temperatura misurata dalla termocoppia è inferiore alla soglia di temperatura imposta, il regolatore PID definisce il duty cycle della tensione PWM che verrà fornita in ingresso al dimmer, in modo da accendere i resistori di potenza e riscaldare la camera climatica; quando la temperatura misurata dalla termocoppia è maggiore del setpoint imposto, il PID definisce il duty cycle della tensione PWM che è portata in ingresso alla ventola, in modo da dissipare calore e ridurre la temperatura all'interno della camera.

Il circuito di condizionamento, realizzato su breadboard, dei segnali PWM ottenuti in uscita dai pin PWM della scheda nucleo utilizzata è alimentato mediante un alimentatore da banco (settato in modo da erogare una tensione di 12 V con una corrente di 0.5 A come limite massimo) utilizzando un cavo BNC-coccodrillo ed opportuno adattatore.

Il segnale PWM in uscita dalla scheda nucleo che rappresenta il segnale di ingresso del dimmer elettronico presenta un range di tensione compreso tra 0 V e 3.3 V quindi è necessario prima filtrare tale segnale mediante un filtro RC opportunamente dimensionato (in modo da eliminare da tale segnale tutte le componenti armoniche ad alta frequenza e mantenendo inalterata la componente continua) e successivamente amplificarlo in modo che il segnale abbia un range di tensione coincidente con quello del dimmer (0 V – 10 V). L'amplificatore operativo utilizzato in configurazione non invertente per amplificare tale segnale ha guadagno in tensione circa pari a 3.2. L'uscita di tale amplificatore verrà fornita ai morsetti del dimmer tramite due cavi coccodrillo-coccodrillo. Il dimmer è direttamente connesso alla rete elettrica quindi presenta in ingresso una tensione pari

alla tensione di rete ed è connessa alla camera climatica mediante una presa che fornisce fase e neutro alla camera climatica stessa.

Il secondo segnale PWM in uscita dalla scheda rappresenta l'ingresso della ventola. Poiché la ventola presenta una tensione nominale di lavoro di 12 V è necessario amplificare il segnale PWM della scheda utilizzando un secondo amplificatore operazione in configurazione non invertente con guadagno in tensione di circa 3.7 V. Infine, poiché la corrente di uscita dell'amplificatore operazionale è troppo bassa per far funzionare la ventola, è necessario porre in cascata all'amplificatore un buffer di corrente ottenuto utilizzando un MOSFET a canale N a source comune: in particolar modo, il carico (che nel caso in esame è la ventola) è posto tra la tensione di alimentazione, pari a 12 V, e il terminale di drain del MOSFET.

Nella seguente figura è mostrato il sistema complessivo realizzato:



## Resistori di potenza presenti nella camera climatica

Il modello dei resistori presenti nella camera climatica è l'RSSD (25 x 168) i quali possiedono un range di resistenza che varia tra 0.12  $\Omega$  e 220  $\Omega$  a seconda della temperatura di funzionamento dei resistori. Tali resistori dissipano una potenza di 200 W quindi nel caso in esame, essendo connessi in serie, la potenza complessiva dissipata è di 400 W.

Il modello di resistore utilizzato presenta un'elevata stabilità a lungo termine con una perdita di efficienza inferiore all'1.5% dopo 1000 ore di utilizzo ed una eccellente resistenza agli shock termici e ai sovraccarichi. Inoltre, presenta un'incertezza al più del 5 %.

Per evitare sovraccarichi la corrente massima che può circolare in ognuno di essi è pari a:

$$I_{max} = \sqrt{\frac{P_t}{R_n}}$$

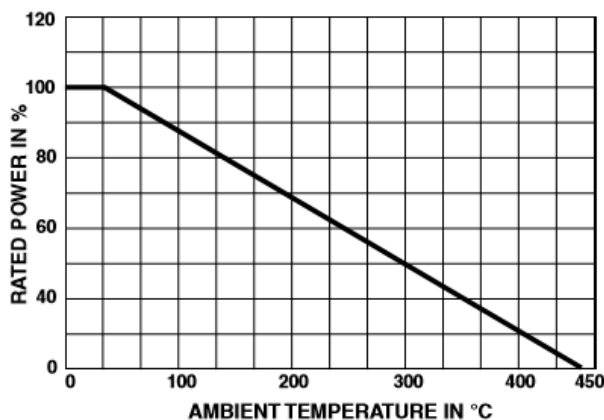
con  $P_t$  che rappresenta la potenza dissipata per effetto Joule e  $R_n$  che rappresenta il valore nominale di resistenza considerata.

Nel caso in esame, poiché la potenza massima dissipata è di 400 W e il valore nominale di resistenza è di 200  $\Omega$ , allora la corrente massima che può circolare nella serie delle due resistenze è pari a:

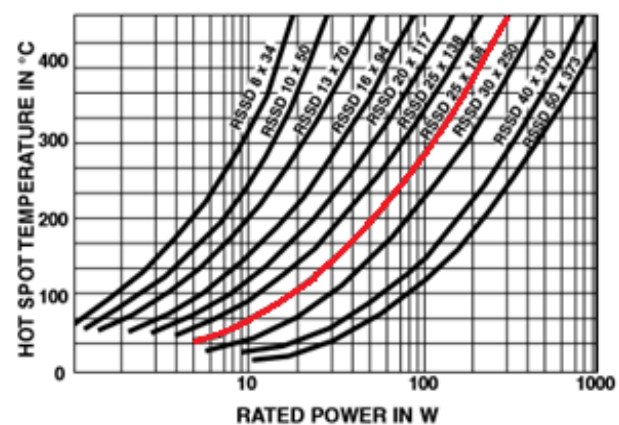
$$I_{max} = \sqrt{\frac{P_t}{R_n}} = \sqrt{\frac{400 \text{ W}}{200 \Omega}} = 2 \text{ A}$$

Nelle seguenti figure vengono mostrati i grafici del power rating in forma percentuale in funzione della temperatura e l'aumento di temperatura in funzione della potenza dissipata.

**POWER RATING**



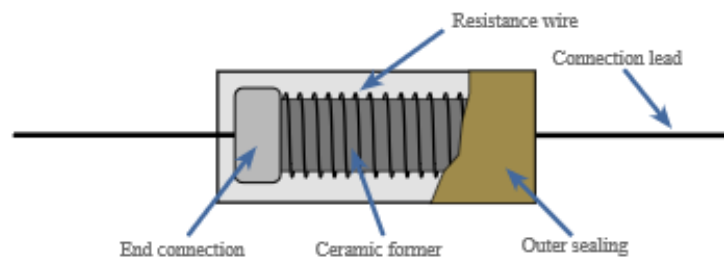
**TEMPERATURE RISE**



Il resistore “a filo avvolto” (wire wound) è costituito da un filo resistivo avvolto attorno a un corpo centrale solitamente realizzato in ceramica, materiale non conduttivo. Normalmente il filo resistivo è isolato, in modo che i fili adiacenti non cortocircuitino. Il materiale di cui è costituito il filo ha un'elevata resistività ed è solitamente costituito da una lega di nichel-cromo (Nichrome) o una lega di rame-nichel-manganese chiamata manganina.

La costruzione dei resistori a filo avvolto permette loro di tollerare temperature elevate e, di conseguenza, in molte applicazioni sono utilizzati come resistori ad alta potenza.

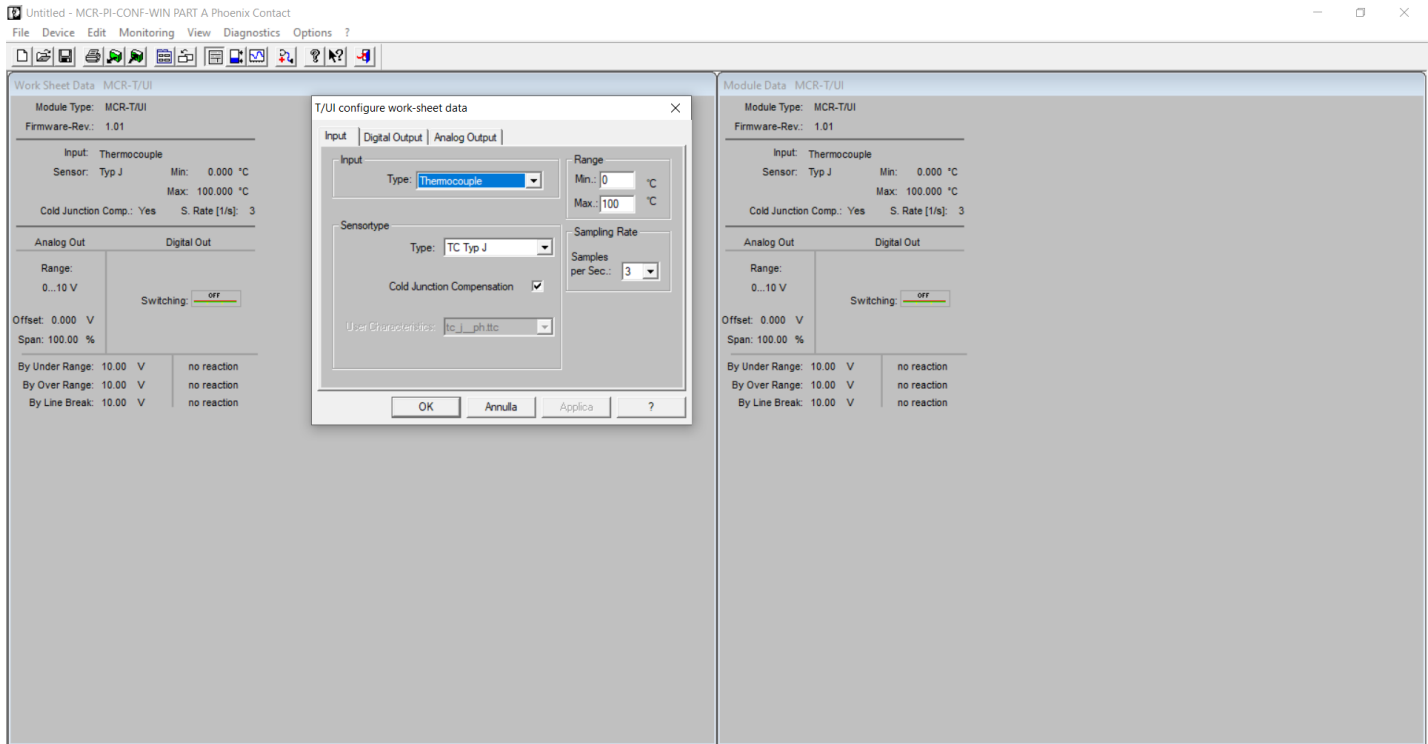
Nella seguente figura è mostrata la struttura di un resistore “wire wound”.





## Modulo programmabile

È stato utilizzato un modulo di programmazione per sensori di temperatura (ad esempio termocoppia di tipo J, NTC o PT100) il quale può essere programmato via software in base al sensore di temperatura che si sta utilizzando (mediante il software Phoenix). La schermata che appare per programmare il tipo di sensore di temperatura collegato al modulo è mostrata nella seguente figura:



La sezione di sinistra mostra le caratteristiche che sono state assegnate al modulo in precedenza mentre la sezione di destra mostra le caratteristiche assegnate al modulo dopo averlo riconfigurato in modo da fornire misure di temperatura attraverso una termocoppia di tipo J. In particolar modo sono stati definiti i seguenti parametri:

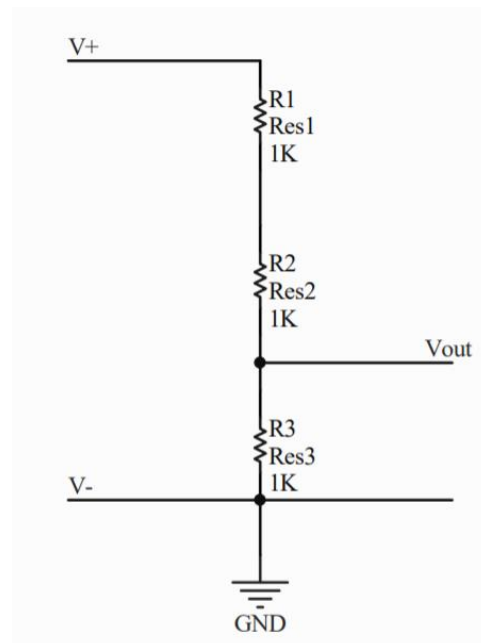
- Tipo di sensore di temperatura, che nel caso in esame è una termocoppia di tipo J;
- Compensazione via software del giunto freddo per una migliore misura della temperatura della camera climatica;
- Numero di campioni al secondo, impostato a 3;
- Range di temperatura pari a [0 °C 100 °C];
- Range di tensione in uscita dal modulo concorde con il range operativo del dimmer [0 V 10 V].

Mediante l'utilizzo di un multimetro da banco è stato verificato il corretto funzionamento della termocoppia di tipo J connessa al modulo precedentemente programmato. In particolar modo sono stati collegati i pin 1 e 2 (come mostrato nel datasheet) alla termocoppia di tipo J, un alimentatore da 24 V DC ai morsetti 8 e 9 del modulo programmabile e il multimetro da banco ai morsetti 11 e 12 del modulo. Sullo schermo del multimetro si misura la tensione di uscita della termocoppia che dipenderà dalla temperatura misurata dalla stessa: si osserva che, all'aumentare della temperatura mediante riscaldamento del terminale della termocoppia, si ha un incremento della tensione misurata dal multimetro da banco ai terminali della termocoppia.

## Partitore di tensione

Il modulo programmabile ha un range di funzionamento della temperatura nel range [0°C 100°C] e fornisce in output valori di tensioni proporzionali alla temperatura misurata, compresi nel range [0 V 10 V], mentre la scheda nucleo potrà leggere al più valori di tensione di 3.3 V. Per tale motivazione è stato necessario realizzare su breadboard un partitore di tensione che ripartisce in 3 la tensione di uscita del modulo programmabile in modo che il range di tensione in ingresso alla scheda NUCLEO sia [0 V 3.3 V].

Il circuito è costituito da 3 resistori con stesso valore nominale di resistenza (1 kΩ) ed è rappresentato nella seguente figura:



## Circuito di condizionamento del segnale di ingresso del dimmer elettronico

Il dimmer elettronico accetta una tensione di alimentazione massima in ingresso pari a 10V. Poiché la scheda NUCLEO F401RE utilizzata è in grado di fornire una tensione di alimentazione di 3.3 V, è stato necessario dimensionare e realizzare su breadboard un circuito di condizionamento del segnale in ingresso al dimmer. Il circuito utilizzato per amplificare la tensione di alimentazione fornita dalla scheda è costituito da un amplificatore operazionale in configurazione non invertente integrato nell'LM358N.

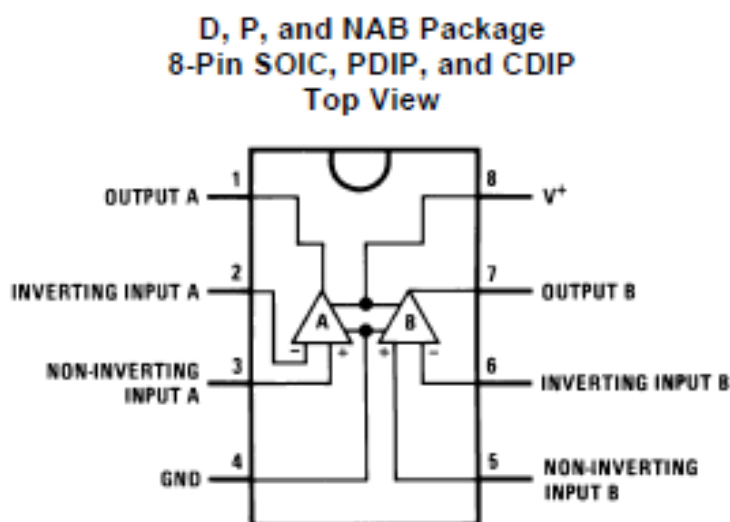
Nella configurazione non invertente dell'operazionale sono state utilizzate tre resistenze, due dal valore nominale  $R_1 = R_2 = 10 \text{ k}\Omega$  ed una dal valore nominale  $R_3 = 22 \text{ k}\Omega$  in modo che il guadagno dell'operazionale in configurazione non invertente sia circa 3.2.

Si ricordi che il guadagno in tensione di un operazionale in configurazione non invertente è pari a :

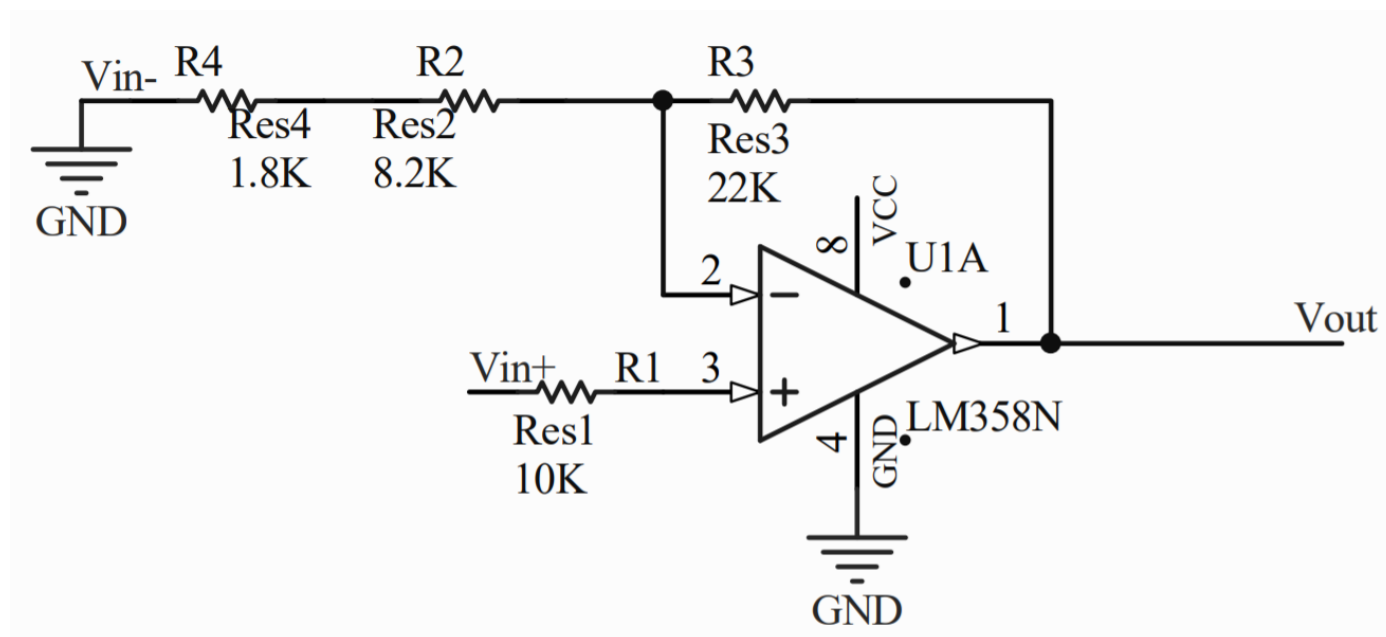
$$A_V = 1 + \frac{R_3}{R_1} .$$

L'LM358N utilizzato presenta due operazionali identici integrati e nel caso in esame sono utilizzati entrambi: il primo OPAMP, con guadagno in tensione di circa 3.2, è utilizzato per amplificare il segnale da dare in ingresso al dimmer mentre il secondo OPAMP, con guadagno in tensione di circa 3.7, è utilizzato per amplificare il segnale da dare in ingresso alla ventola quando la temperatura misurata dalla termocoppia supera il setpoint imposto.

I pin dell'integrato LM358N e le funzioni ad essi associate sono rappresentati nella seguente figura:

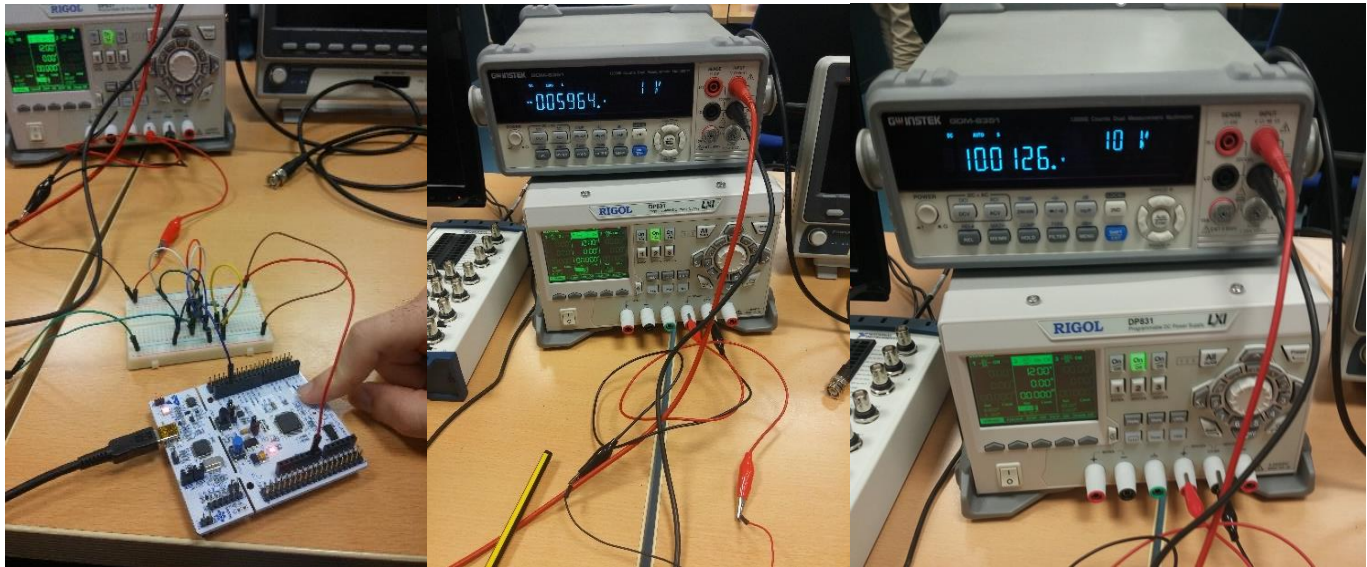


Di seguito viene riportato lo schematico del circuito di amplificazione del segnale da dare in ingresso al dimmer elettronico realizzato su ALTIUM con guadagno in tensione di circa 3.2:



Dopo aver realizzato su breadboard il circuito di condizionamento, si è collegati l'OPAMP in configurazione non invertente realizzato con guadagno pari a 3.2 al pin di alimentazione della scheda, che fornisce una tensione di 3.3 V. Successivamente è stata misurata la tensione di uscita dell'OPAMP collegando l'uscita dell'amplificatore al multimetro da banco mediante cavo BNC - coccodrillo e si è osservati che per una tensione di ingresso all'amplificatore pari a 0 V si ottiene una tensione in uscita pressoché nulla mentre per una tensione di ingresso all'amplificatore pari a 3.3 V (massima tensione erogabile dalla scheda) si ottiene un valore di tensione circa pari a 10 V.

Le seguenti figure mostrano il circuito di condizionamento del segnale realizzato e la fase di verifica del funzionamento del circuito:



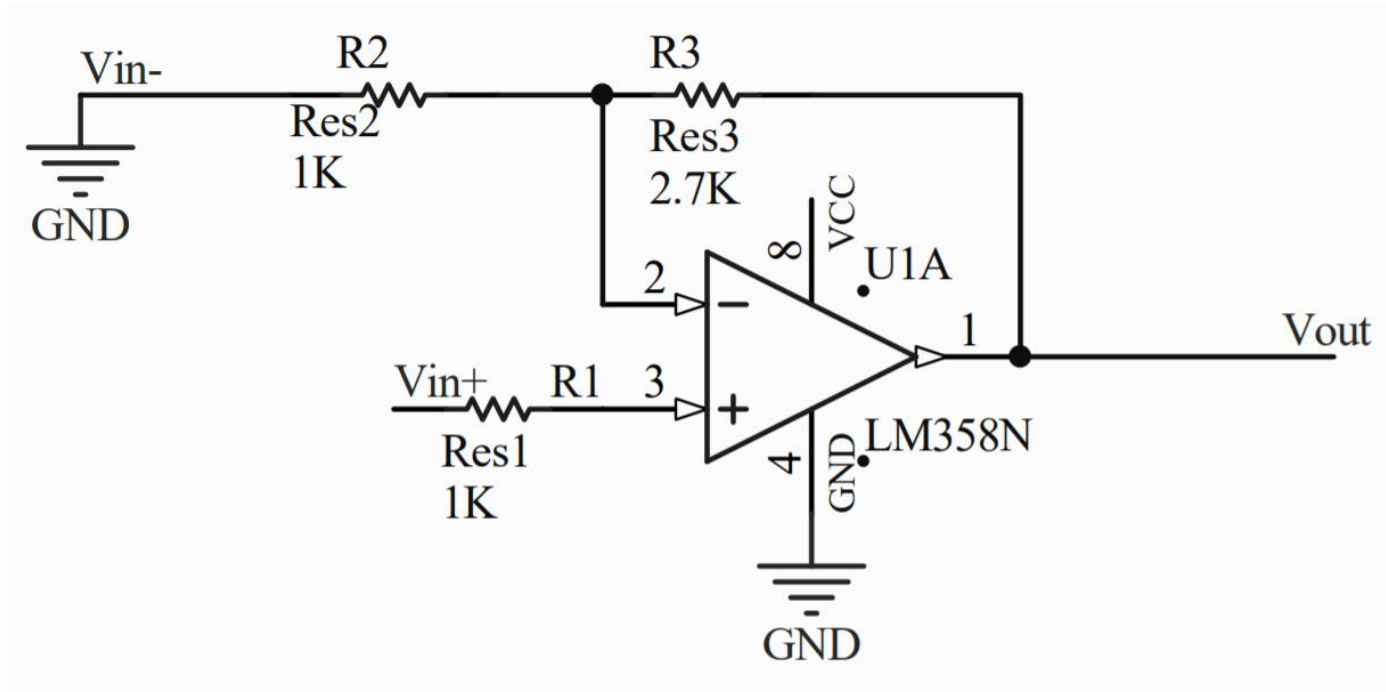
## Circuito di condizionamento della ventola

Il secondo operazionale presente nell'integrato LM358N è stato utilizzato come amplificatore in configurazione non invertente per amplificare la tensione fornita dalla scheda con un guadagno in tensione di circa 3.7.

L'amplificatore operazionale è costituito in tal caso da 2 resistenze di valore nominale di 1 k $\Omega$  ed una resistenza dal valore nominale di 2.7 k $\Omega$  (che definisce il ramo di retroazione).

Il motivo per cui è stato necessario utilizzare tale amplificatore dipende dal fatto che il segnale PWM ottenuto in uscita dalla scheda ha una tensione picco-picco di 3.3 V. La ventola, invece, si aziona quando la tensione in ingresso è contenuta nel range [6 V 16 V], preferibilmente quando assume valore pari a 12 V. Con un guadagno di circa 3.7 V otteniamo quindi un segnale PWM con ampiezza picco-picco di circa 12 V, come richiesto dalla ventola, mantenendo inalterato il duty cycle imposto dal regolatore PID implementato via software.

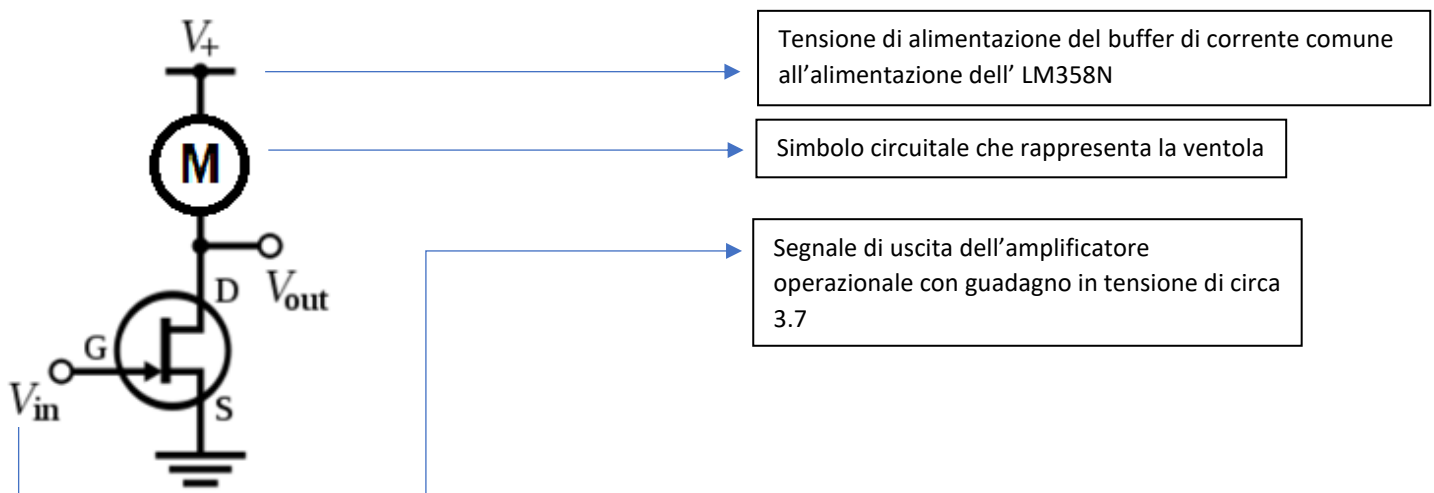
Di seguito viene riportato lo schematico del circuito di amplificazione del segnale da dare in ingresso alla ventola realizzato su ALTIUM con guadagno in tensione di circa 3.7:



In cascata all'amplificatore operazionale è stato posto un buffer di corrente per incrementare la corrente uscente dall'amplificatore e pilotare l'accensione della ventola, la quale necessita una tensione di ingresso di 12 V ed eroga una potenza di 2.4 Watt.

La resistenza di uscita dell'operazionale infatti, idealmente infinita, non permette di far erogare corrente verso la ventola, seppur alimentata con 12 V di tensione in ingresso. Inserendo il buffer di corrente la resistenza di uscita è stata notevolmente ridotta, permettendo alla ventola di assorbire la corrente necessaria per il suo corretto funzionamento.

Il buffer di corrente è stato realizzato utilizzando un MOSFET a canale N a source comune, la cui configurazione circuitale è mostrata nella seguente figura:



Come si osserva dal precedente circuito, l'uscita dell'amplificatore operazionale rappresenta il segnale fornito al terminale di Gate del MOSFET. La ventola, invece, è posta tra la tensione di alimentazione fornita dall'alimentatore da banco e comune all'alimentazione dell' LM358N, e il terminale di Drain del MOSFET.

## Dimensionamento del filtro RC

Il dimensionamento del filtro RC è necessario per poter ottenere un segnale di tensione continuo a partire da un segnale PWM, da dare in ingresso al dimmer elettronico. Per tale motivazione è stata definita una frequenza di taglio di alcuni Hz in modo da eliminare dallo spettro delle ampiezze del segnale la maggior parte delle armoniche, lasciando la sola componente continua.

Sono stati utilizzati i seguenti componenti:

- Condensatore elettrolitico di capacità pari a 10  $\mu\text{F}$
- Resistore di resistenza pari a 10  $\text{k}\Omega$

Si ricordi che il guadagno di un filtro RC, nel dominio di Laplace, è pari a  $A_v = \frac{1}{1+sRC}$ .

Si definisce costante di tempo del filtro RC:

$$\tau_p = RC = 10 \text{ k}\Omega * 10 \mu\text{F} = 0.1 \text{ s} = \frac{1}{w_p},$$

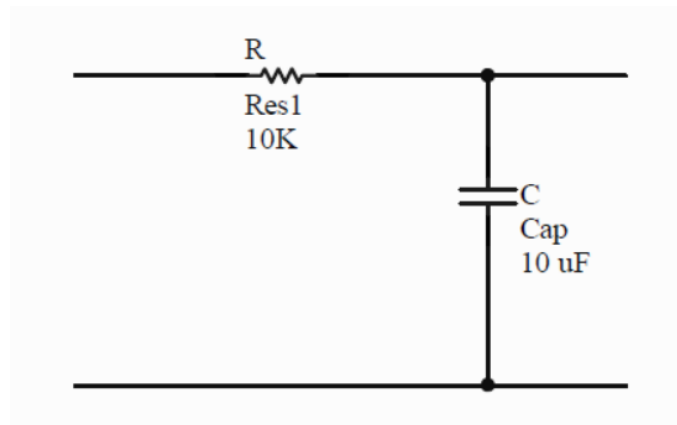
quindi si ottiene che la pulsazione di taglio è pari a:

$$w_p = \frac{1}{\tau_p} = 10 \frac{\text{rad}}{\text{s}}.$$

La pulsazione di taglio del filtro RC realizzato è pari a  $w_p = 2\pi f_H$  dove  $f_H$  rappresenta la frequenza di taglio del filtro. Dalla precedente relazione si ottiene che:

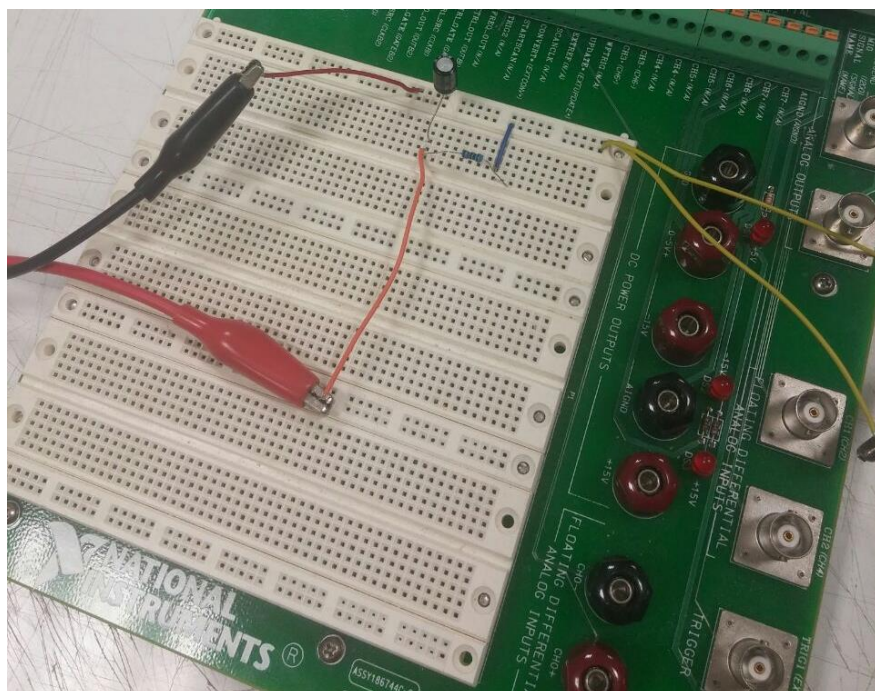
$$f_H = \frac{w_p}{2\pi} = \frac{10 \frac{\text{rad}}{\text{s}}}{2\pi} \cong 1.6 \text{ Hz}$$

Lo schema circuitale del filtro RC realizzato è mostrato nella seguente figura:





Il filtro RC realizzato su breadboard è mostrato nella seguente figura:

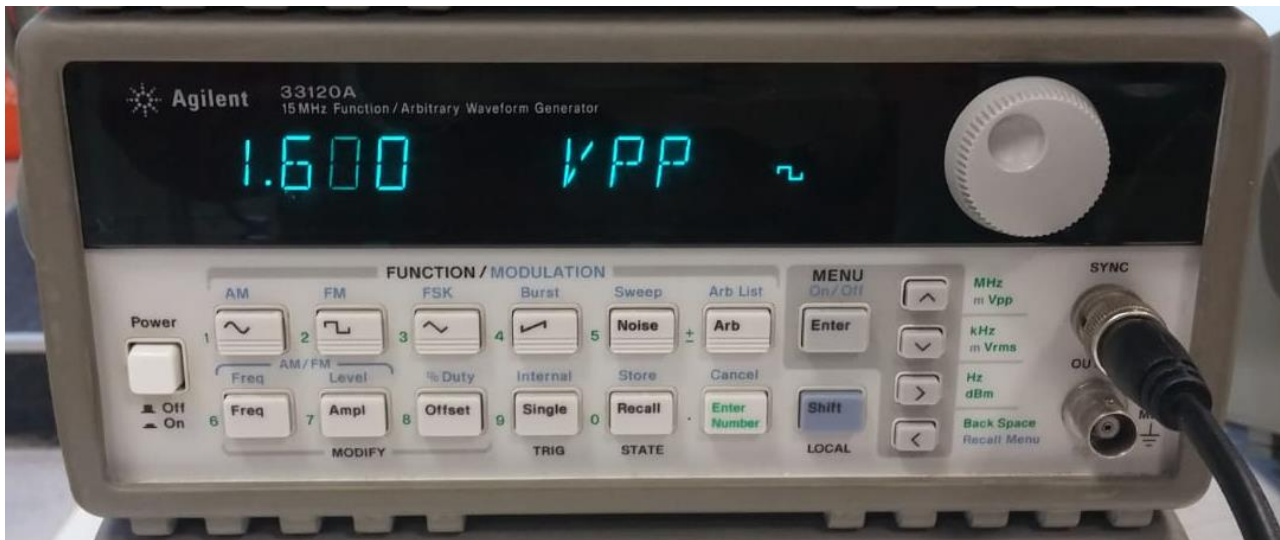


Dopo aver dimensionato il filtro RC, per verificarne il corretto funzionamento, si è misurata la tensione ai capi del condensatore collegando l'uscita del filtro e GND all'oscilloscopio mediante cavo BNC - coccodrillo e alimentando il filtro RC con un segnale onda quadra fornito dal generatore di funzioni con le seguenti proprietà:

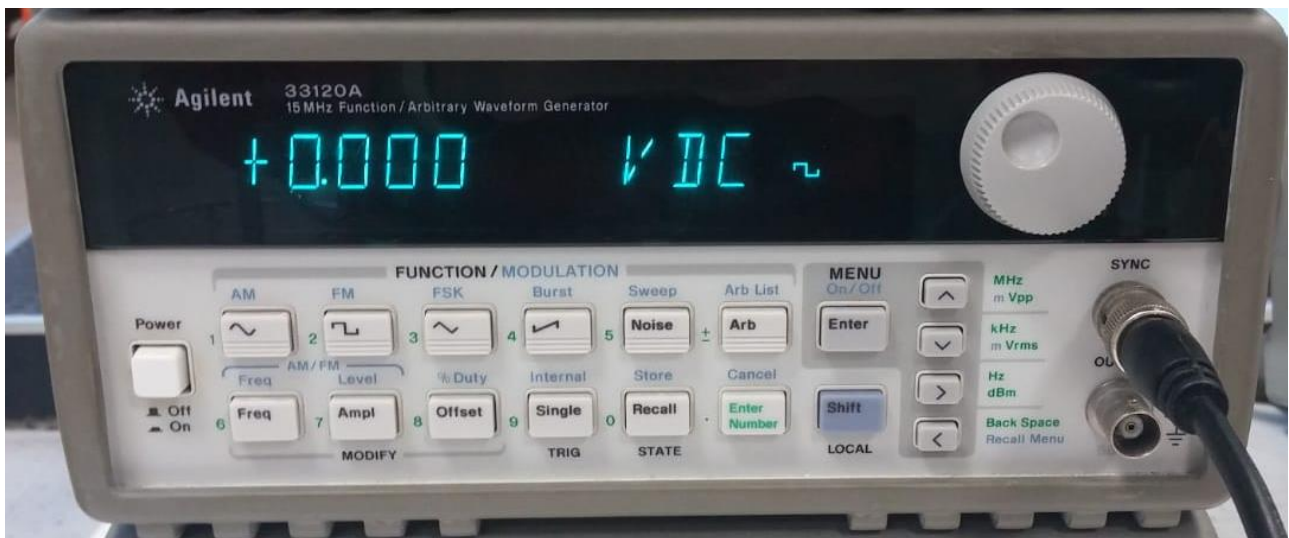
- Frequenza 1 kHz



- Ampiezza picco-picco 1.6 V

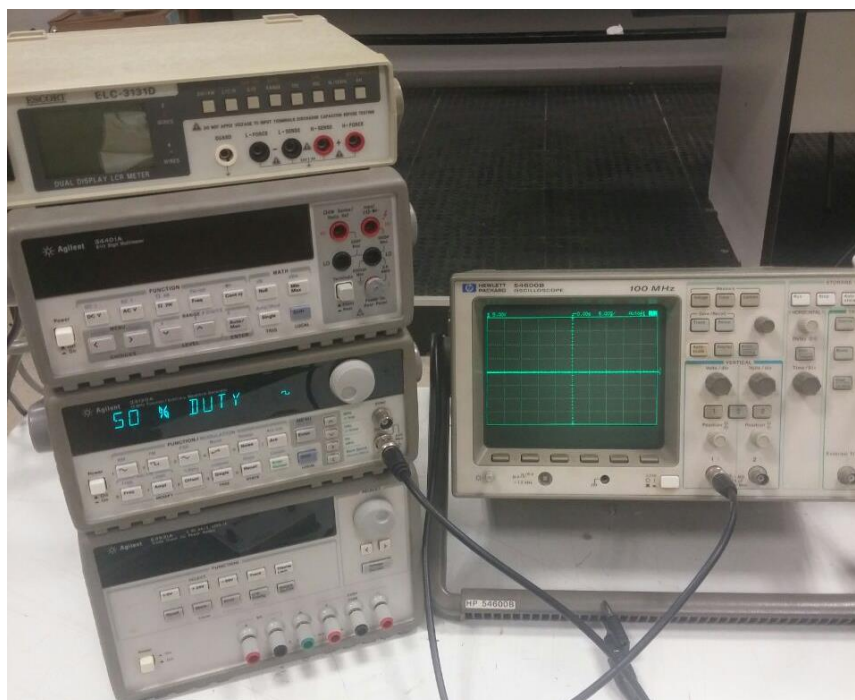


- Offset di 0 V

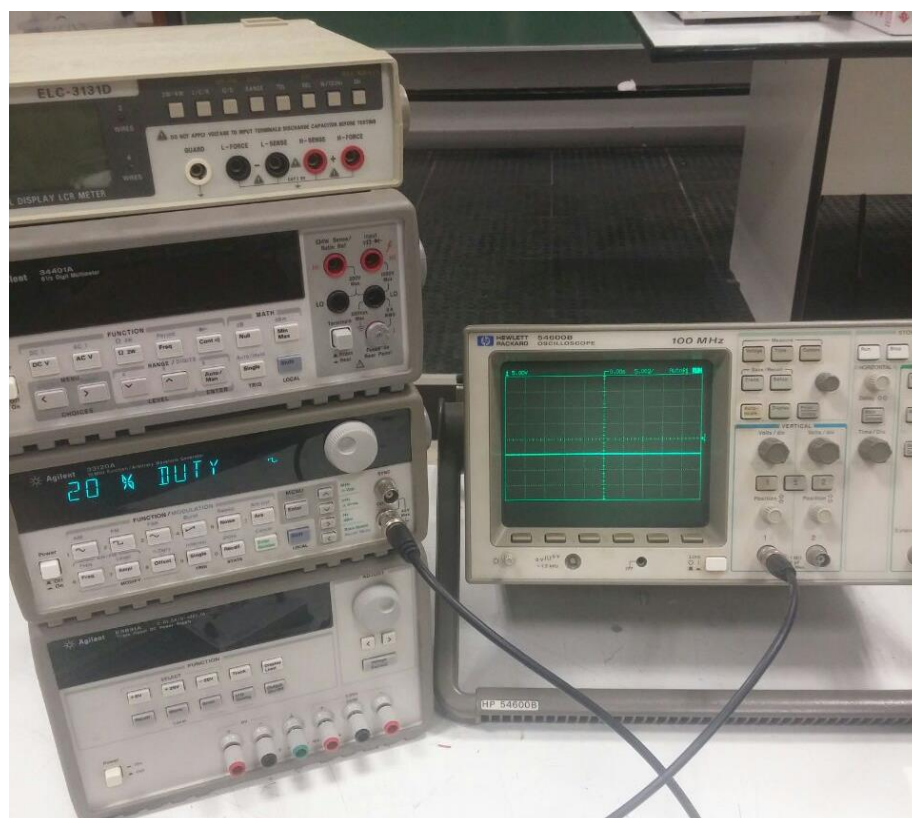




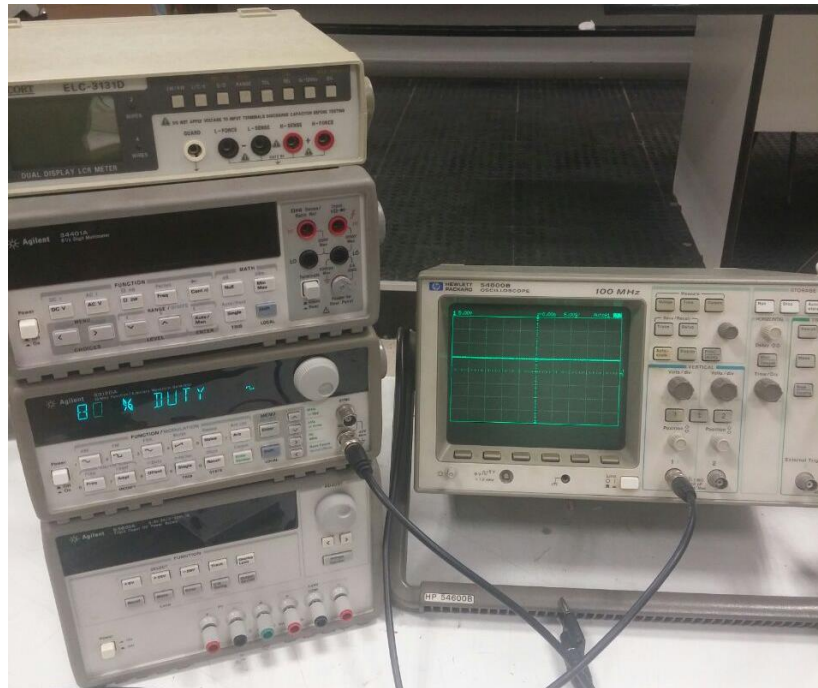
Impostando un duty cycle pari al 50% si è ottenuti il seguente segnale in uscita dal filtro RC:



Impostando un duty cycle del 20% ci si aspetta una tensione di uscita negativa, ed infatti è mostrata sull'oscilloscopio la seguente figura:



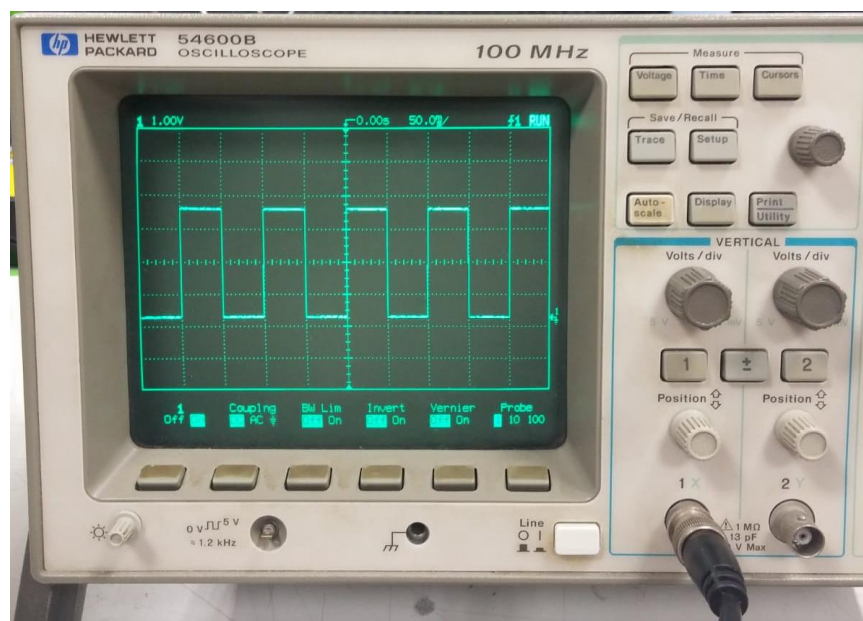
Impostando un duty cycle dell'80% ci si aspetta un valore di tensione positivo in uscita dal filtro ed infatti si visualizza sull'oscilloscopio la seguente figura:



Successivamente si è collegati l'ingresso del filtro RC ad un pin della scheda che genera un segnale PWM unipolare, come mostrato nella seguente figura. Per verificare il corretto funzionamento del circuito complessivo scheda - filtro RC si è considerati un caso particolare, impostando un periodo del segnale pari a:

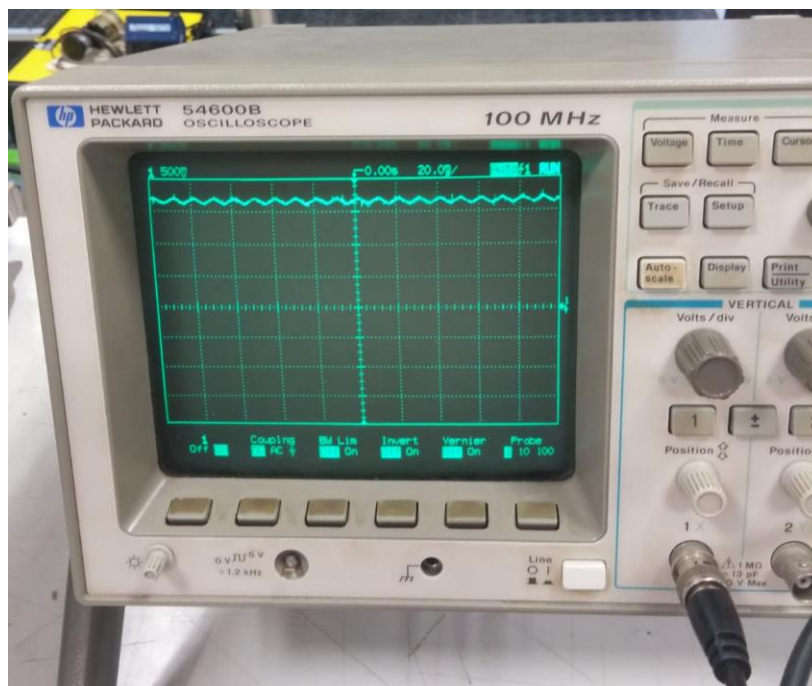
$$T_m = 10 \text{ ms.}$$

La forma d'onda ottenuta in uscita dal pin PWM della scheda è una forma d'onda quadra unipolare come mostrato sull'oscilloscopio:



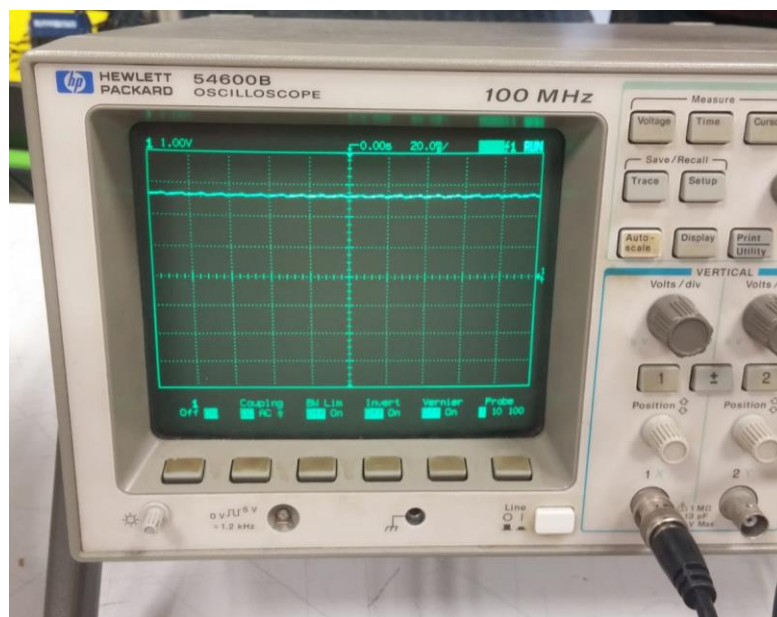
Si osservi che l'onda quadra è unipolare, infatti il riferimento dell'oscilloscopio è stato spostato verso il basso (definito dal cursore indicato con "1") in modo da visualizzare in modo ottimale la forma d'onda del segnale.

Impostando un duty cycle del 50 % per il segnale PWM ottenuto dal pin della scheda, viene visualizzata la seguente forma d'onda sull'oscilloscopio:



In tal caso si ottiene un valore nominale di tensione ai capi del condensatore del filtro RC pressoché costante attorno a 1.6 V (nonché il 50 % della tensione di alimentazione della scheda).

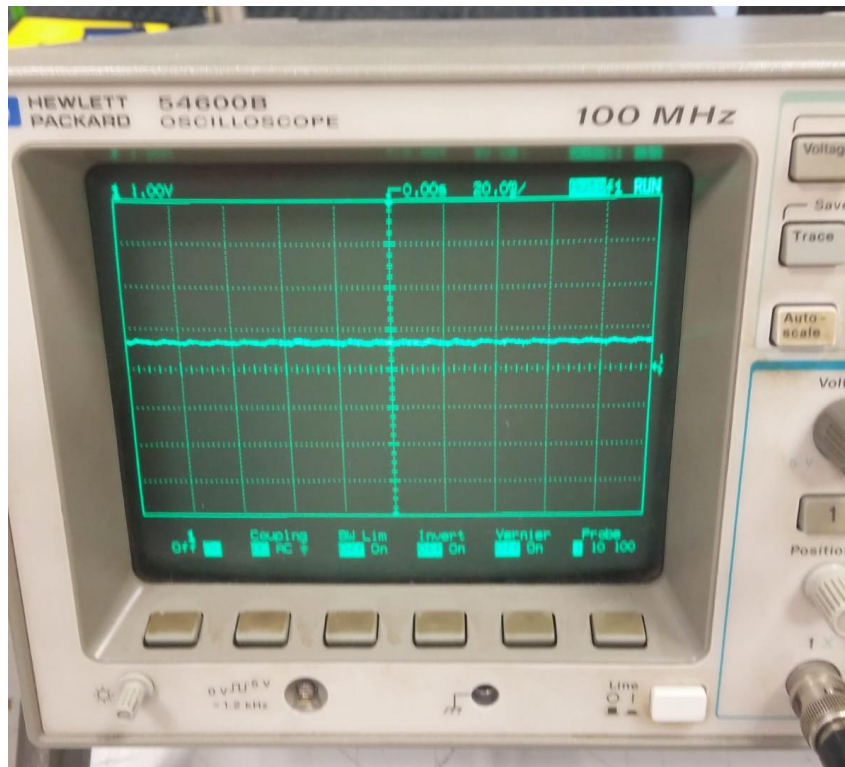
Impostando un duty cycle dell'80 % è visualizzata la seguente forma d'onda sull'oscilloscopio:





In tal caso si ottiene un valore nominale di tensione ai capi del condensatore del filtro RC di circa l'80% della tensione di alimentazione della scheda.

Impostando un duty cycle del 20% ed una tensione di alimentazione della scheda di 3.3 V si visualizza la seguente forma d'onda sull'oscilloscopio:

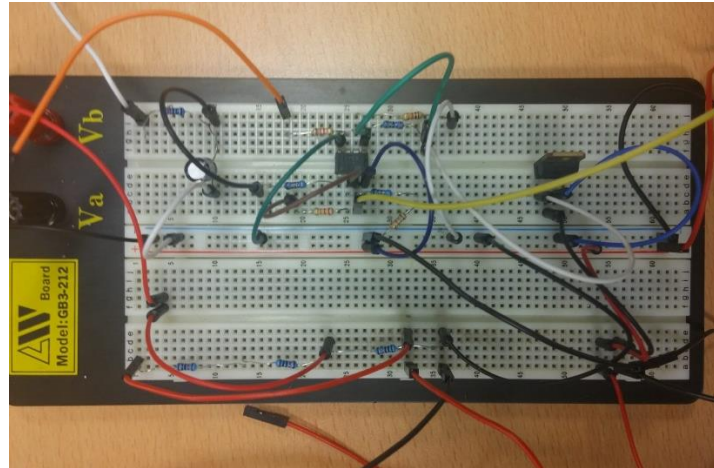


In tal caso il valore di tensione ai capi del condensatore del filtro RC è pari al 20% della tensione di alimentazione fornita dalla scheda.

In questo modo è stato verificato il corretto funzionamento del filtro RC che ha in ingresso il segnale PWM ottenuto in uscita dal pin PWM della scheda, e ha in uscita un valore di tensione pressoché costante, proporzionale alla tensione di alimentazione della scheda in base al duty cycle considerato.

## Realizzazione del circuito realizzato su breadboard in Altium Designer

Il circuito di condizionamento dei segnali PWM che devono essere inviati dalla scheda al dimmer elettronico ed alla ventola è stato realizzato su breadboard come mostrato nella seguente figura:

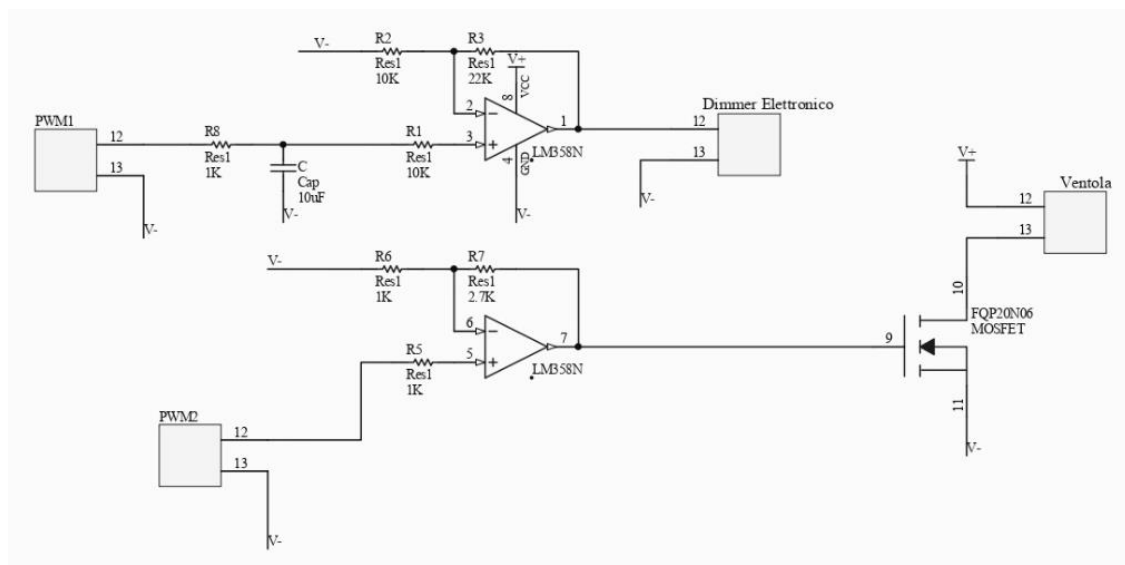


In seguito, di tale circuito è stato realizzato lo schematico e la PCB utilizzando il software Altium Designer.

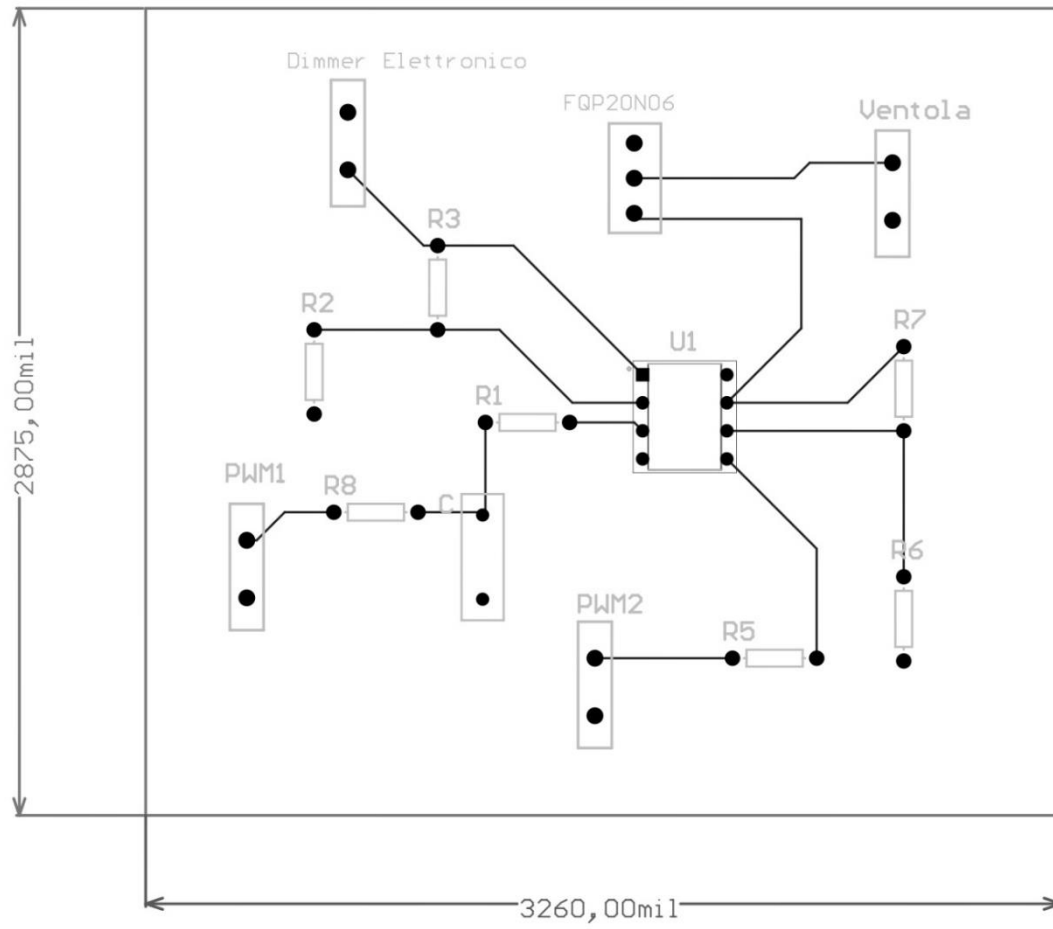
Innanzitutto, è stato realizzato lo schematico del circuito. Per fare ciò, è stata importata la libreria relativa all'integrato utilizzato contenente due amplificatori operazionali. Gli operazionali sono stati utilizzati in configurazione non invertente con guadagno in tensione di 3.7 e 3.2 (rispettivamente l'operazionale in basso e in alto).

L'ingresso dell'operazionale in alto rappresenta il segnale PWM relativo al dimmer elettronico mentre il secondo operazionale ha in ingresso il segnale PWM relativo alla ventola.

L'uscita del primo operazionale è connessa ad un Female Header a 2 pin al quale sarà connesso il dimmer elettronico, mentre il secondo operazionale ha in uscita un buffer di corrente (MOSFET-N a source comune) per amplificare la corrente di uscita. Tra la tensione di alimentazione (indicata con  $V_+$ ) e il terminale di drain del MOSFET è collegato un Female Header a 2 pin a cui sarà possibile connettere tramite jumper la ventola.



La seguente figura mostra la PCB dello schematico precedentemente descritto:



## Codice Mbed

All'interno del progetto è stata utilizzata una scheda NUCLEO F401RE per l'acquisizione di dati, in particolare la tensione in uscita dalla termocoppia di tipo J, e per il pilotaggio dei due componenti principali della camera climatica, le resistenze di potenza e la ventola per il raffreddamento dell'aria.

La scheda è stata programmata attraverso l'uso dell'IDE "Mbed.org", ambiente di sviluppo e compilatore Open Source che supporta numerosi prodotti della serie NUCLEO.

Il codice realizzato è stato diviso in tre file:

- **Main.cpp** – File principale contenente le funzioni realizzate.
- **PID\_Int.h** – File header contenente i prototipi delle funzioni e la definizione delle variabili di contesto;
- **PID\_Def.h** – File header contenente i valori costanti e le definizioni;

All'interno del file Main.cpp sono state realizzate due funzioni: **Main()** e **getTemperature\_thermocouple()**.

La funzione **Main()** implementa le seguenti funzionalità:

- Calcolo dei duty cycle dei segnali PWM da dare in ingresso al dimmer e alla ventola di raffreddamento. Questa operazione è stata realizzata attraverso l'utilizzo di una libreria che implementa un controllore PID.
- Comunicazione seriale, con il software Teraterm in fase di implementazione e debug ed infine con il software LABView per la modifica in tempo reale dei parametri del controllore PID;

La funzione **getTemperature\_thermocouple()** implementa la funzionalità di lettura della tensione in uscita dal modulo programmabile, connesso alla termocoppia, e di conversione della tensione misurata nel valore di temperatura corrispondente.

### Main.cpp

```
#include "PID_Int.h"

/*****
    Inizializzazione Variabili Globali
*****/

Serial pc(SERIAL_TX, SERIAL_RX);                /* Inizializza la comunicazione seriale. */

AnalogIn Thermocouple(THERMOCOUPLE);           /* Inizializza il pin A0 a cui è collegata la termocoppia. */

PwmOut Driver_res(DRIVER_RES);                  /* PWM per la regolazione dei resistori. */
PwmOut Driver_fan(DRIVER_FAN);                  /* PWM per la regolazione della ventola. */

static Ticker ticker;                           /* Ticker utilizzato dalla libreria millis.h. */

Context Contesto;                               /* Struttura che contiene le variabili di contesto. */

/* Oggetto della classe PID per il controllo della resistenza. */
PID controller(&Contesto.Input, &Contesto.Output, &Contesto.Setpoint, Contesto.Kp, Contesto.Ki, Contesto.Kd, DIRECT);
```

```

/* Oggetto della classe PID per il controllo della ventola. */
PID controller_back(&Contesto.Input, &Contesto.Output, &Contesto.Setpoint, Contesto.Kp, Contesto.Ki, Contesto.Kd, REVERSE);

/*****
    Funzioni
*****/

/*! \fn int main()
 *
 * \brief Funzione Main.
 *
 */
int main()
{
    /* Invia tramite seriale lo stato di ONLINE. */
    pc.printf("ONLINE\n");

    /* Inizializza il timer. */
    startMillis();

    /* Definisce il periodo in ms della PWM. */
    Driver_res.period_ms(100);
    Driver_fan.period_ms(100);

    /* Avvia i controllori PID. */
    controller.SetMode(AUTOMATIC);
    controller_back.SetMode(AUTOMATIC);

    /* Ottiene i valori iniziali dei parametri del PID. */
    pc scanf("%lf", &Contesto.Kp);
    pc scanf("%lf", &Contesto.Ki);
    pc scanf("%lf", &Contesto.Kd);
    pc scanf("%lf", &Contesto.Setpoint);

    /* Setta i parametri dei PID. */
    controller.SetTunings(Contesto.Kp, Contesto.Ki, Contesto.Kd);
    controller_back.SetTunings(Contesto.Kp, Contesto.Ki, Contesto.Kd);

    /* Loop infinito. */
    while(1) {

        /* Ottiene il valore che indica se un parametro è stato modificato e, in caso affermativo, indica quale dei parametri è stato modificato. */
        pc scanf("%d", &Contesto.Evento);

        switch(Contesto.Evento) {

            /* Modifica del parametro proporzionale del PID */
            case (Modifica_Kp): {
                pc scanf("%lf", &Contesto.Kp);
            }
            break;

            /* Modifica del parametro integrativo del PID */
            case (Modifica_Ki): {
                pc scanf("%lf", &Contesto.Ki);
            }
            break;

            /* Modifica del parametro derivativo del PID */

```



```

    case (Modifica_Kd): {
        pc.scanf("%lf",&Contesto.Kd);
    }
    break;

    /* Modifica del setpoint di temperatura. */
    case (Modifica_Setpoint): {
        pc.scanf("%lf",&Contesto.Setpoint);
    }
    break;

    /* Nessun parametro modificato. */
    default: {
    }
    break;
}

/* Setta i nuovi parametri dei PID. */
controller.SetTunings(Contesto.Kp, Contesto.Ki, Contesto.Kd);
controller_back.SetTunings(Contesto.Kp, Contesto.Ki, Contesto.Kd);

/* Temperatura attuale misurata dalla termocoppia. */
Contesto.Input = getTemperature_thermocouple();

/* Invia tramite seriale la temperatura misurata. */
pc.printf("%lf\n", Contesto.Input);

/* Verifica se la temperatura è già sopra o sotto soglia. */
if(Contesto.Input <= Contesto.Setpoint) {

    /* Processa il PID loop della resistenza. */
    bool isComputed = controller.Compute();

    /* Formatta il duty cycle per ottenere un valore tra [0 1.0]. */
    double duty_res = Contesto.Output/1000;

    /* Sovrascrive il duty cycle. */
    Driver_res.write(duty_res);
    Driver_fan.write(0);

    /* Invia tramite seriale i duty cycle. */
    pc.printf("%lf\n",0); /* Duty cycle della ventola. */
    pc.printf("%lf\n",duty_res); /* Duty cycle della resistenza. */

} else {

    /* Processa il PID loop della ventola. */
    bool isComputed_back = controller_back.Compute();

    /* Formatta il duty cycle per ottenere un valore tra [0 1.0]. */
    double duty_fan = Contesto.Output/1000;

    /* Sovrascrive il duty cycle. */
    Driver_fan.write(duty_fan);
    Driver_res.write(0);

    /* Invia tramite seriale i duty cycle. */
    pc.printf("%lf\n",duty_fan); /* Duty cycle della ventola. */
    pc.printf("%lf\n",0); /* Duty cycle della resistenza. */
}

/* Delay. */

```

```

        wait(0.1);
    }
}

/*! \fn double getTemperature_thermocouple()
 *
 * \brief Calcola la temperatura misurata dalla termocoppia utilizzando la
 *        la legge di trasformazione da tensione a temperatura usando i NIST
 *        Coefficients.
 */
double getTemperature_thermocouple()
{
    double temp;                                /* Temperatura misurata dalla termocoppia. */

    /* Legge e formatta il valore in V misurato dal pin A0. */
    double Lettura_termocoppia = Thermocouple.read() * (double)VOLT_MAX;

    /* Converte la tensione misurata dalla scheda al valore in mV che la termocoppia avrebbe in us
    cita a quella temperatura. */
    double b=(Lettura_termocoppia*mV_AT_100_C) / VOLT_MAX;

    /* NIST coefficients. */
    double a0=0.000000;
    double a1=1.978425*10 ;
    double a2=-2.001204*0.1;
    double a3= 1.036969*0.01;
    double a4=-2.549687*0.0001;
    double a5= 3.585153*0.000001;
    double a6=3.585153*0.00000001;
    double a7=5.099890*0.0000000001;

    /* Potenze. */
    double b2=b*b;
    double b3=b2*b;
    double b4=b3*b;
    double b5=b4*b;
    double b6=b5*b;
    double b7=b6*b;

    /* Converte da valore di tensione a valore di temperatura nel range [0 100] °C */
    temp=a0+a1*b+a2*b2+a3*b3+a4*b4+a5*b5+a6*b6+a7*b7;

    return temp;
}

```

## PID\_Int.h

```
#ifndef PID_INT_H
#define PID_INT_H
/*****
    Inclusioni
    *****/
#include "mbed.h"
#include "millis.h"
#include "PID.h"
#include "PID_Def.h"

/*****
    Strutture
    *****/

/*! \struct Context
 *
 * \brief Struttura che contiene le variabili di contesto.
 *
 */
struct Context {

    double Input;                /* Temperatura della camera climatica misurata
dalla termocoppia. */
    double Output;              /* Valore in uscita dal PID. */

    int Evento;                 /* Evento legato alla variazione dei parametri
Kp, Ki, Kd o setpoint. */

    double Setpoint;            /* Temperatura da raggiungere. */

    double Kp;                  /* Costante proporzionale del PID. */
    double Ki;                  /* Costante integrativa del PID. */
    double Kd;                  /* Costante derivativa del PID. */

};
/*****
    Enumerativi
    *****/

/*! \enum Event
 *
 * \brief Definisce gli eventi legati alla variazione dei parametri Kp, Ki, Kd o setpoint.
 *
 */
enum Event {

    Nessuna_modifica = 0,
    Modifica_Kp,
    Modifica_Ki,
    Modifica_Kd,
    Modifica_Setpoint

};

/*****
    Prototipi funzioni
    *****/
double getTemperature_thermocouple();

#endif
```

## PID\_Def.h

```
#ifndef PID_DEF_H
#define PID_DEF_H

#define THERMOCOUPLE      PA_0          /* NUCLEOF401RE Analog input pin A0 */
#define DRIVER_RES        PB_3          /* NUCLEOF401RE PWM output pin PB_3 */
#define DRIVER_FAN        PB_4          /* NUCLEO401RE PWM output pin PB_5 */
#define VOLT_MAX           3.3          /* Valore massimo di tensione [V] in uscita dal
partitore di tensione successivo al modulo programmabile. */

#define mV_AT_100_C        5.269        /* Valore di tensione [mV] misurato dalla termocop
pia a 100°C */
#define MANUAL             0            /* Modalità di funzionamento manuale del PID. */
#define AUTOMATIC          1            /* Modalità di funzionamento automatica del PID. */
/

#define DIRECT              0            /* Il PID calcola l'uscita in modo da passare da I
nput a SetPoint */
#define REVERSE             1            /* Il PID calcola l'uscita in modo da passare da I
nput a -SetPoint */
#endif
```

## Codice LabVIEW

Si è ritenuto necessario creare un'interfaccia grafica per il controllo manuale del sistema realizzato. È stato utilizzato come ambiente di sviluppo LabVIEW. L'interfaccia sviluppata permette di visualizzare a schermo la temperatura misurata dalla termocoppia, i duty cycle dei segnali PWM dati in ingresso alle resistenze di potenza e alla ventola. Sempre tramite l'interfaccia sviluppata è possibile impostare manualmente i parametri caratteristici del PID, ovvero la costante proporzionale  $K_p$ , la costante integrativa  $K_i$  e la costante derivativa  $K_d$  e il Setpoint di temperatura desiderato.

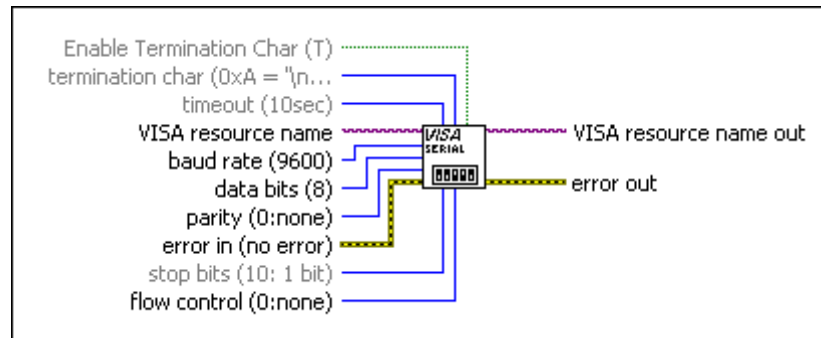
La comunicazione tra scheda e Interfaccia LabVIEW avviene tramite connessione seriale (utilizzando il protocollo UART). Infatti, nell'applicazione implementata sulla scheda sono effettuati dei *"printf"* e degli *"scanf"* attraverso cui rispettivamente si invia il dato a LabVIEW e si riceve su LabVIEW il dato dalla scheda.

La comunicazione seriale in LabVIEW è gestita usando le API di I/O fornite dalla libreria VISA.

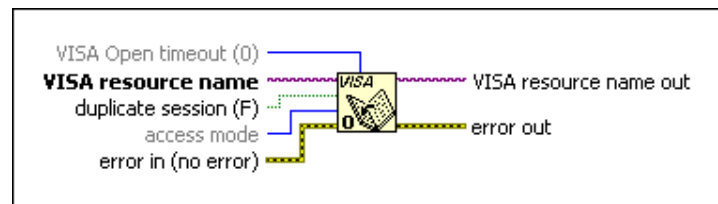
La *Virtual Instrument Software Architecture (VISA)* fornisce funzioni di basso livello che vengono utilizzate da LabVIEW per comunicare con i driver della scheda. Il livello VISA interagisce, tramite i driver di livello inferiore, con il bus di comunicazione (porta USB nel caso in oggetto).

Le funzioni della libreria VISA utilizzate sono le seguenti:

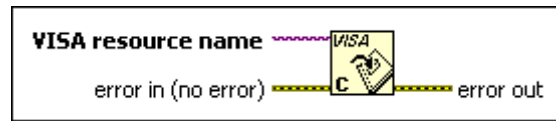
1. La funzione **VISA Configure Serial Port** inizializza la porta seriale specificata dal nome della risorsa VISA considerata. *Enable Termination Char* prepara il dispositivo seriale a riconoscere il carattere di terminazione. Se *TRUE* (come nella presente relazione), l'attributo *VI\_ATTR\_ASRL\_END\_IN* è impostato per riconoscere il carattere di terminazione.



2. La funzione **VISA Open** apre una sessione di comunicazione con il dispositivo specificato dal nome della risorsa VISA e restituisce un identificatore di sessione che può essere utilizzato per chiamare qualsiasi altra operazione di quel dispositivo.



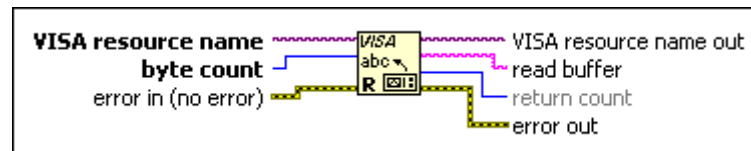
3. La funzione **VISA Close** chiude la sessione di comunicazione con il dispositivo o in taluni casi chiude la comunicazione del dispositivo relativo ad un particolare evento specificato dal nome della risorsa VISA.



4. La funzione **VISA Write** invia la stringa del buffer di scrittura al dispositivo specificato dalla VISA Resource Name. Il nome della risorsa VISA genera le informazioni sulla sessione, in modo che possano essere utilizzate dai VI in cascata. Il cluster di errore contiene le informazioni di errore.



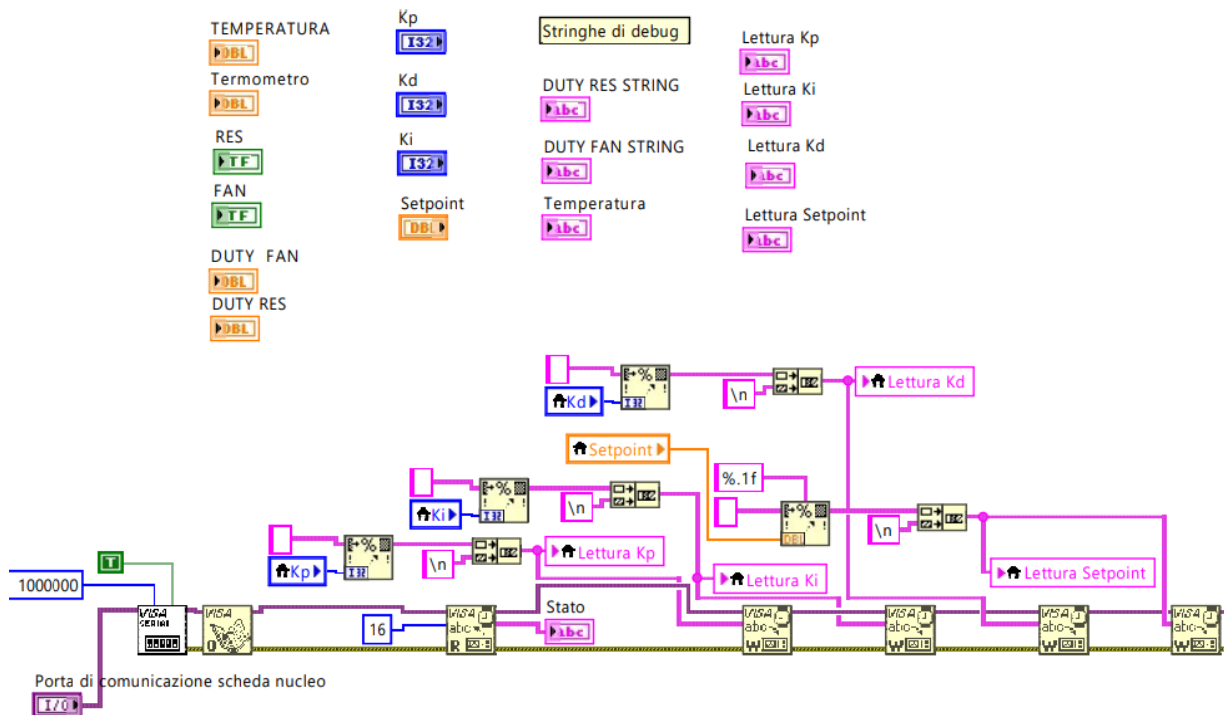
5. La funzione **VISA Read** legge i dati dal dispositivo specificato dal nome della risorsa VISA. Il *byte count* è il numero di byte da leggere nella stringa restituita dal *read buffer*. Questa funzione potrebbe restituire meno del numero di byte richiesti se la funzione raggiunge la fine del buffer (se ad esempio viene raggiunto un carattere di terminazione o se si verifica un timeout).



Il codice LabVIEW implementato è di seguito presentato.

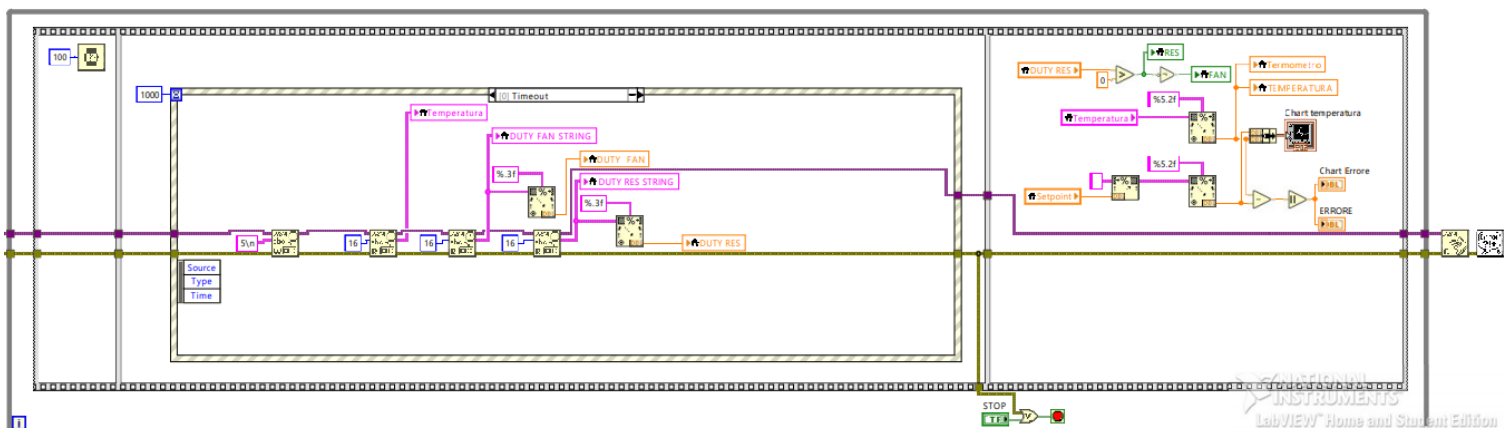
## Block Diagram

1. Si configura la comunicazione seriale tra scheda e LabVIEW e si apre tale comunicazione seriale utilizzando **VISA Configure Serial Port** e **Visa Open**, dando in ingresso la corretta risorsa VISA.
2. Tramite la funzione **VISA Read** viene ricevuta la stringa di debug *"ONLINE"* utile a capire se la comunicazione seriale è stata inizializzata correttamente.
3. Utilizzando quattro funzioni **VISA Write** in cascata sono inviati alla scheda i valori predefiniti di  $K_p$ ,  $K_i$ ,  $K_d$ , e Setpoint. Si noti che i valori devono essere inviati in formato di stringa, quindi viene prima effettuata una conversione dei dati ricevuti dalla scheda in formato stringa.



4. A questo punto si accede al **While Loop**, in cui è presente una **Flat Sequence** utile ad organizzare il codice sequenzialmente. La **Flat Sequence** si divide in 3 blocchi:

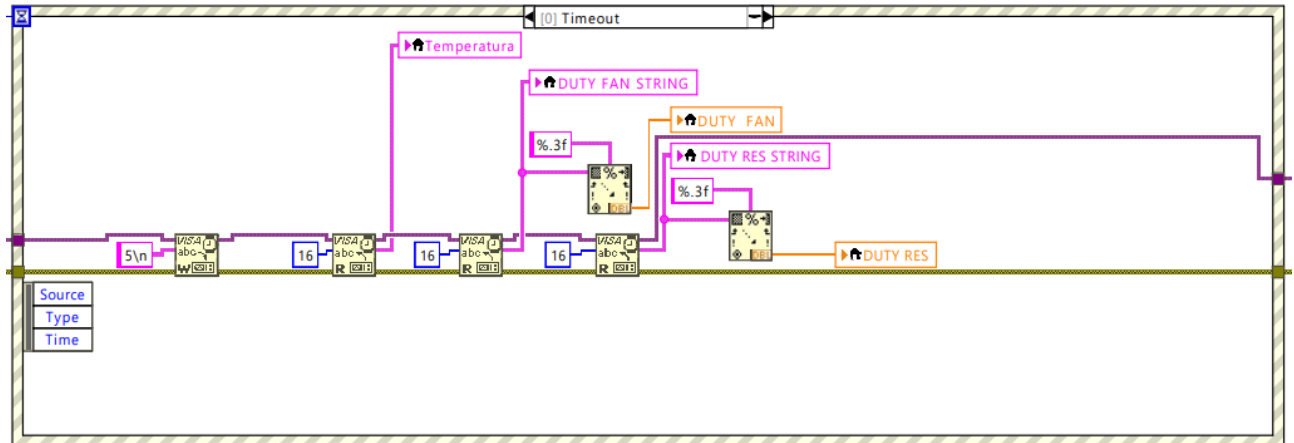
- I. Un blocco avente un *wait time* di 100 *ms* utile a sincronizzare il codice LabVIEW sviluppato con l'applicazione implementata su scheda.
- II. Un blocco dove è presente una **Event Structure**, che permette di ottimizzare il codice secondo il paradigma della programmazione ad eventi. Nel caso in esame sono presenti 5 eventi che verranno successivamente discussi.
- III. Un blocco finale in cui viene effettuata una conversione tra stringhe ricevute dalla scheda in valori float che rappresentano le grandezze di interesse, ovvero temperatura e duty cycle.



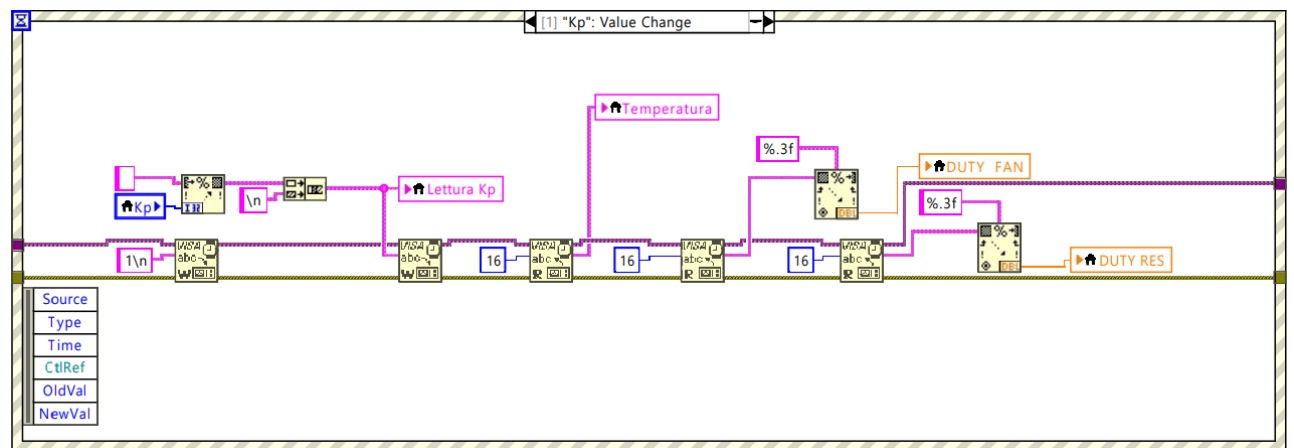
## Event Structure

Come già annunciato, nel caso in esame sono presenti 5 differenti eventi, distinguibili in 2 categorie:

- I. L'evento di **Timeout** (caso 5) a cui si accede dopo aver atteso 1 sec nell'ipotesi in cui non si verifichi alcuna condizione che porti in un altro caso.



- II. Gli eventi che avvengono quando è rilevato un "data change" (dal caso 1 al caso 4) su una delle variabili del PID e sul Setpoint (ovvero quando l'utente varia tali parametri mediante le manopole presenti nel Front Panel). A titolo di esempio è mostrata nella seguente figura l'evento relativo alla modifica del parametro  $K_p$ :



Come è possibile osservare dalla precedente figura, all'interno di ogni *event structure* vi è inizialmente un blocco **VISA Write** la cui funzione è quella di inviare alla scheda una stringa contenente un numero (1, 2, 3, 4, o 5) che indica quale evento si è verificato.

Successivamente, se è avvenuto un "data change", si invia alla scheda, mediante la funzione **VISA Write**, il nuovo valore della variabile modificata dall'utente nel front panel in LabVIEW.

Infine, in tutti i casi definiti nell'*event structure*, mediante 3 funzioni **VISA Read** in cascata si riportano sul front panel i valori di temperatura misurata dalla termocoppia e i duty cycle dei segnali PWM inviati dalla scheda alle resistenze e alla ventola. Questi valori sono salvati in delle variabili locali che verranno poi successivamente

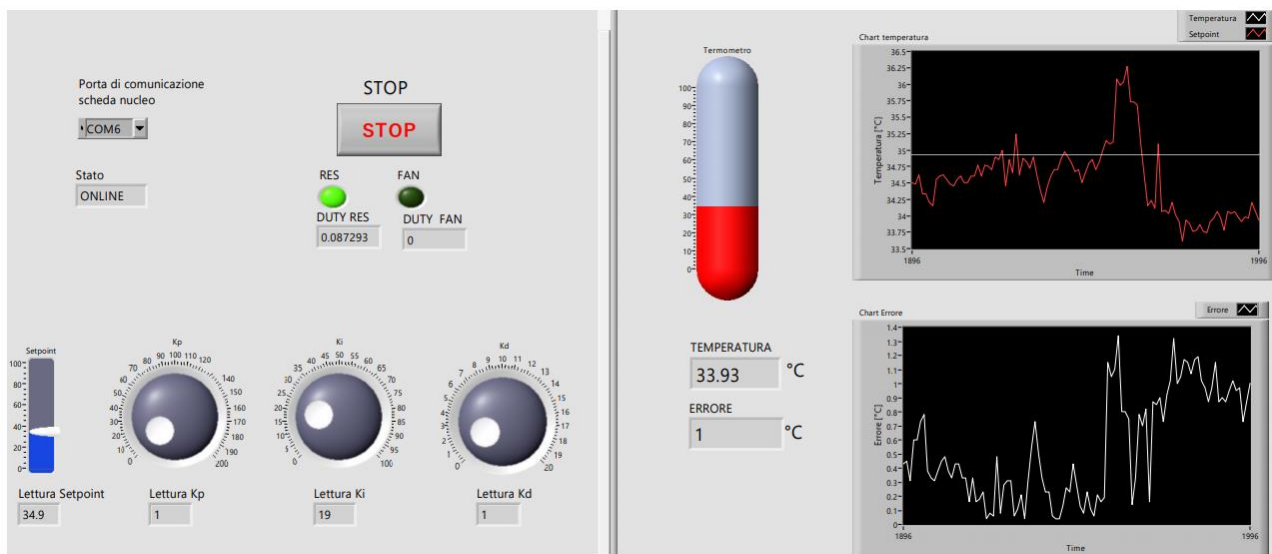


convertite in numeri interi o frazionari ed utilizzati per determinare i grafici che si possono osservare sul lato destro del front panel.

## Front Panel

Il *Front Panel* è organizzato in due sezioni divise da una *Vertical Split Bar* che dividono la parte di supervisione (sezione a destra) dalla parte di controllo (sezione a sinistra):

- Nella sezione di controllo vi è una *Vertical Bar* per il settaggio del setpoint e tre manopole per settare i parametri del controllore PID. Vi è inoltre il pannello di configurazione della porta seriale VISA e un output di stato che restituisce la stringa “*ONLINE*” se la connessione seriale tra scheda e LabVIEW è avvenuta correttamente. Inoltre, vi sono due led che indicano quale delle due azioni di controllo è attiva (Ventola o Resistenze) e il corrispettivo valore del Duty Cycle del segnale PWM che le controlla. Infine, vi è un pulsante di STOP per arrestare la comunicazione.
- Nella sezione di supervisione vi sono: un termometro con indicatore numerico associato per visualizzare la temperatura e l’errore tra la temperatura misurata dalla termocoppia e il Setpoint fissato. Infine, vi sono due *Waveform Chart* in cui si visualizzano le forme d’onda relative alla temperatura (grafico in alto) e all’errore di temperatura (grafico in basso).



## Teoria sul controllo di temperatura mediante regolatore PID

Con il termine controllo definiamo l'azione svolta per portare e mantenere ad un valore prefissato un parametro fisico di un processo. Se si indica con  $r(t)$  il valore che si vuole far assumere alla variabile controllata e con  $y(t)$  il valore misurato da un sensore di tale grandezza, possiamo definire la funzione errore come la differenza delle due grandezze, ovvero:

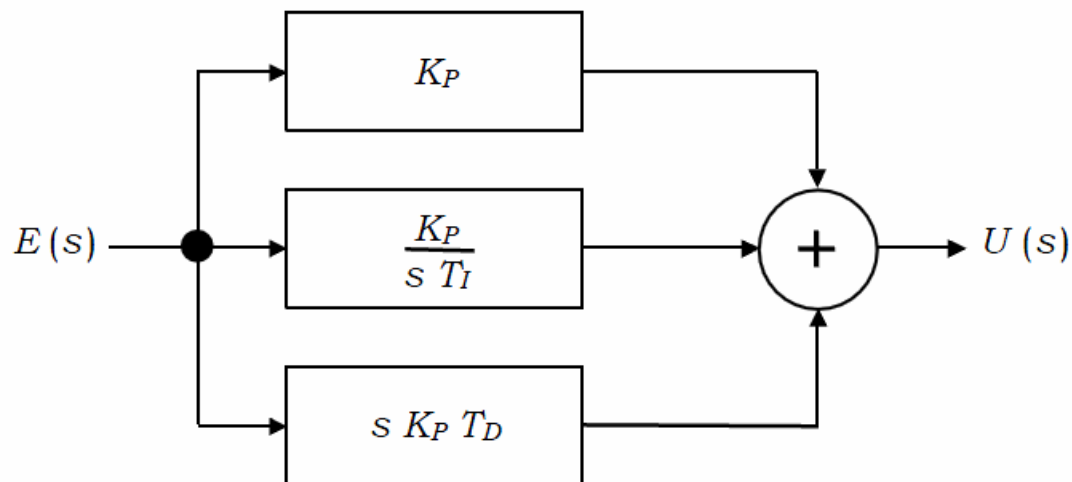
$$e(t) = r(t) - y(t).$$

Lo scopo dell'azione di controllo è quello di applicare un segnale di ingresso tale da rendere nullo l'errore a regime.

Nella realtà, il segnale di controllo non viene applicato direttamente al processo ma ad un attuatore, che interpreta l'istruzione fornitagli dal regolatore, per variare alcune condizioni.

Tra i controllori più diffusi ci sono i PID che, per controllare l'intero sistema si servono combinatamente di tre funzioni di controllo: proporzionale, integrativa e derivativa.

Lo schema generale di un controllore PID è il seguente:



Il legame uscita – ingresso è rappresentato, nel dominio di Laplace, dalla seguente equazione:

$$U(s) = K_p \left( 1 + \frac{1}{T_i s} + s T_d \right) E(s) = K_p + \frac{K_i}{s} + s K_d$$

Tale equazione è espressa nel dominio del tempo come segue:

$$u(t) = K_p \left( e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{de(t)}{dt} \right)$$

dove i parametri  $K_p$ ,  $T_d$  e  $T_i$  sono tre costanti positive modificabili a scelta dall'operatore esterno.

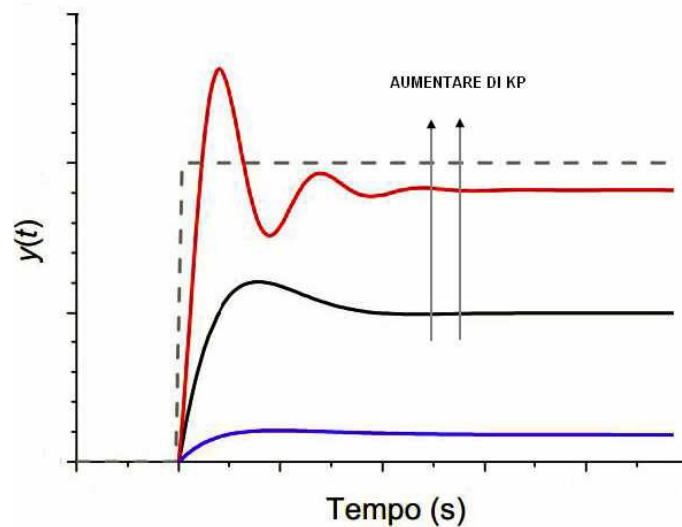
Di seguito vengono definite le azioni di controllo di ogni controllore nel processo di controllo considerato.

## Proporzionale

Il termine  $K_p$  è detto coefficiente proporzionale. Il valore della variabile di controllo risulta essere proporzionale all'errore, ovvero alla differenza tra segnale di riferimento e segnale misurato: maggiore è il valore di tale coefficiente, più elevata è la reazione del controllore anche per valori modesti dell'errore.

In termini statici, un  $K_p$  elevato riduce l'errore a regime mentre dal punto di vista dinamico produce un aumento della velocità di risposta del sistema e, quindi, una maggiore prontezza nell'inseguimento del riferimento. Allo stesso tempo, però, provoca una diminuzione dei margini di stabilità del sistema.

Di seguito è mostrata in figura la risposta di un sistema del secondo ordine in funzione del coefficiente proporzionale.

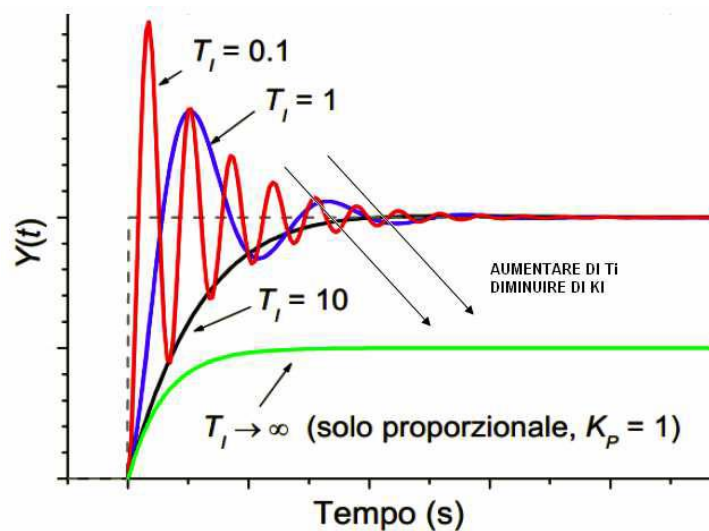


## Integrativo

La variabile di controllo è in tal caso proporzionale all'integrale dell'errore. Dal punto di vista statico, la funzione principale del termine integrale consiste nell'annullare l'errore a regime. In termini dinamici l'introduzione del termine integrale porta ad un aumento del ritardo di fase e, quindi, un peggioramento dei margini di fase ed ampiezza del sistema retro-azionato.

Il contributo integrale risulta essere la componente più adatta per tenere in considerazione l'evoluzione avuta dell'errore in passato. Il parametro con il quale l'operatore può variare il suo contributo è il termine  $T_i$ , inversamente proporzionale al coefficiente integrale  $K_i$ .

Di seguito è mostrata in figura la risposta di un sistema del secondo ordine in funzione del coefficiente integrale.

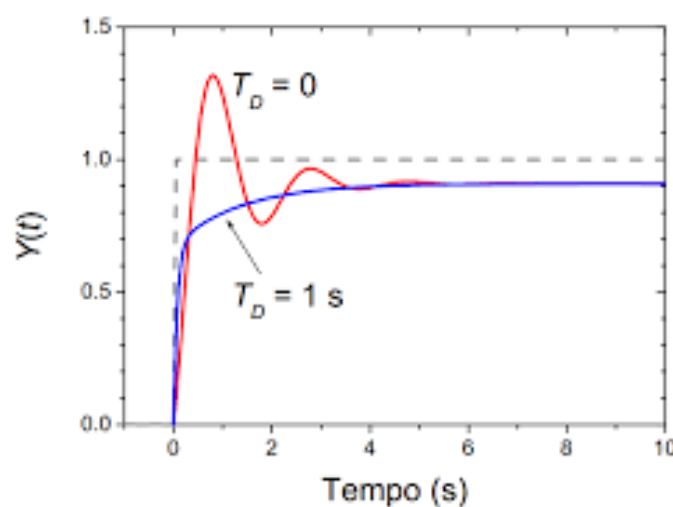


## Derivativo

La variabile di controllo è in questo caso proporzionale alla derivata dell'errore. L'azione derivativa ha lo scopo di tentare di anticipare l'andamento dell'errore negli istanti futuri. Più elevato è il valore del coefficiente derivativo, più si dà importanza a quello che si presume essere l'andamento futuro dell'errore, conferendo maggiore prontezza all'algoritmo, mentre valori più bassi trasferiscono sulla variabile di controllo variazioni più contenute.

Altra funzione del termine derivativo consiste nel migliorare i margini di stabilità del sistema fornendo un anticipo di fase all'anello di controllo in retroazione. Di contro, il termine derivativo ha la caratteristica di amplificare i segnali con contenuto armonico a frequenze elevate e, quindi, in caso di segnale di ingresso affetto da errore questa funzione può portare a un peggioramento del controllo.

Di seguito è mostrata in figura la risposta di un sistema del secondo ordine in funzione del coefficiente derivativo.



L'insieme delle azioni di questi regolatori permettono al controllore PID di svolgere efficacemente la sua azione di controllo.

### **Taratura del PID mediante metodo di Ziegler-Nichols**

La difficoltà nel realizzare un controllo PID sta nel trovare la miglior stima dei tre parametri, poiché tale stima dipende dalla struttura del sistema in esame. Esistono tuttavia dei metodi, denominati empirici, che in seguito ad alcune misurazioni di prova sul sistema permettono di ricavare i valori ottimali dei parametri del PID. Tra i più comuni vi è il metodo di Ziegler-Nichols.

Tale metodo prevede di effettuare una misura preliminare ponendo  $k_i = 0$  e  $k_d = 0$  e, dunque, gestendo la sola parte proporzionale del controllo. La misura consiste nell'aumentare gradualmente il valore della costante proporzionale finché il sistema non inizia ad oscillare stabilmente attorno al valore di aspettazione (ossia presenta oscillazioni che non si smorzano dopo un periodo transitorio).

A questo punto si misurano i valori della costante critica  $K_c$  e quello del periodo delle oscillazioni  $P_c$ . Da questi due valori si possono ricavare quelli delle tre costanti, secondo quanto indicato nella seguente tabella.

<b>Ziegler-Nichols</b>	$k_p$	$k_i$	$k_d$
P	$\frac{K_c}{2}$	/	/
PI	$\frac{K_c}{2,2}$	$\frac{P_c}{1,2}$	/
PID	$\frac{K_c}{1,7}$	$\frac{P_c}{2}$	$\frac{P_c}{8}$

# Teoria sulle termocoppie

## Nozioni generali

Una termocoppia è una giunzione di due metalli o leghe metalliche dissimili il cui funzionamento si basa sull'effetto Seebeck. L'applicazione di questo effetto è quello di effettuare una misura di temperatura attraverso una misura di tensione.

La variazione di tensione non è lineare rispetto alla variazione della temperatura: se  $dT$  è sufficientemente piccolo è possibile applicare una linearizzazione locale della funzione ottenendo  $dV = SdT$ , con  $S$  coefficiente di Seebeck [ $V/K$ ]. Questo coefficiente varia in base alla tipologia di metallo o lega utilizzata ma, in generale, è di poche decine di microvolt per grado Celsius. Ciò indica che i sensori che sfruttano l'effetto Seebeck possiedono un basso valore di sensibilità.

Uno dei più frequenti trasduttori utilizzati per la misura di temperatura è la termocoppia, che possiede il vantaggio di essere poco costosa e di operare in un ampio range di temperatura. I materiali maggiormente utilizzati sono quelli con elevata conducibilità elettrica e bassa conducibilità termica.

Esistono diverse tipologie di termocoppie, ad esempio la termocoppia di tipo  $J$ , utilizzata nel caso in esame, è costituita da un conduttore di ferro e da un conduttore costituito da una lega di rame e nichel detta costantana. Essa possiede un elevato potere termoelettrico ed è in grado di operare in modo efficiente nel range  $0^{\circ}\text{C} - 750^{\circ}\text{C}$ .

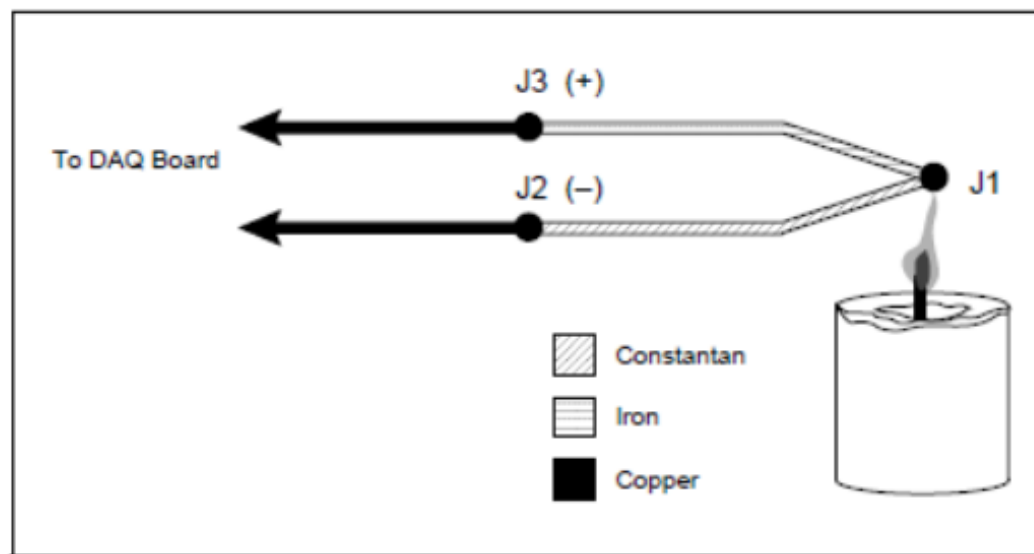


Figure 2. J-Type Thermocouple

Per misurare una tensione di Seebeck della termocoppia, non è possibile semplicemente collegare la termocoppia a un voltmetro o altro sistema di misurazione, perché il collegamento dei fili della termocoppia al sistema di misurazione crea circuiti termoelettrici parassiti che compromettono l'accuratezza delle misure effettuate.

Si consideri il circuito illustrato nella Figura 2, in cui si vuole utilizzare una termocoppia di tipo  $J$  per misurare la temperatura della fiamma di una candela. I due fili della termocoppia sono collegati ai cavi di rame di una scheda DAQ. Si noti che il circuito contiene tre diverse giunzioni metalliche:  $J1$ ,  $J2$  e  $J3$ .  $J1$ , la giunzione della

termocoppia, genera un effetto Seebeck proporzionale alla temperatura della fiamma della candela. J2 e J3 hanno ciascuno il proprio coefficiente di Seebeck e generano una tensione termoelettrica proporzionale alla temperatura sui terminali DAQ. Per determinare il contributo di tensione in J1, è necessario conoscere le temperature delle giunzioni J2 e J3 e la relazione tensione-temperatura di tali giunzioni. È quindi possibile sottrarre i contributi di termocoppie parassite generate da J2 e J3 dalla tensione misurata in J1.

### Compensazione del “giunto freddo”

Le termocoppie richiedono una qualche forma di riferimento di temperatura per compensare le termocoppie parassite indesiderate. Il termine “giunto freddo” deriva dalla pratica tradizionale di mantenere questa giunzione di riferimento a  $0^{\circ}\text{C}$  in un bagno di ghiaccio. Le tabelle di riferimento della termocoppia sono redatte dal National Institute of Standards and Technology (NIST) con questa configurazione, illustrata in Figura 3.

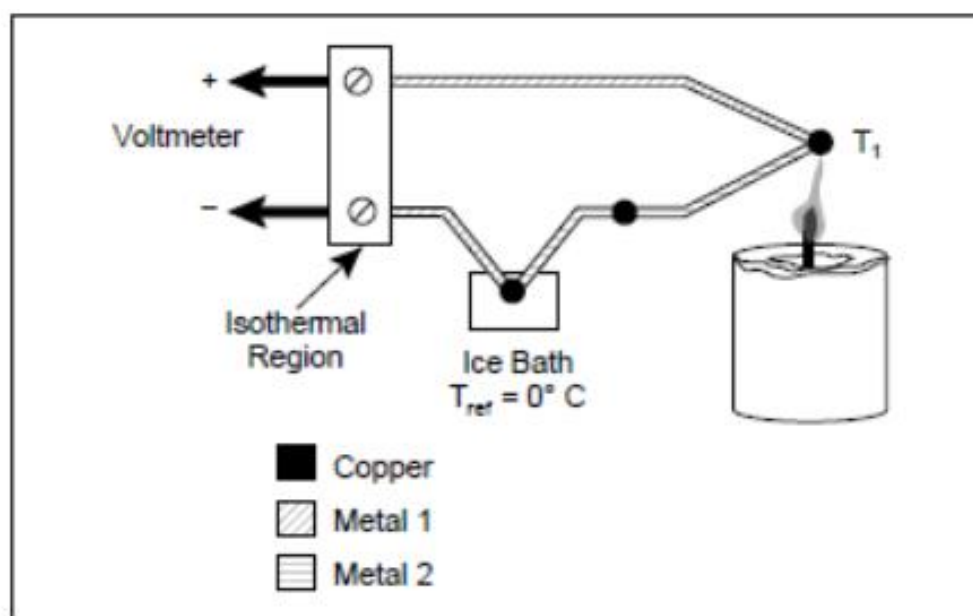


Figure 3. Traditional Temperature Measurement with Reference Junction Held at  $0^{\circ}\text{C}$

Nella precedente figura, la tensione misurata dipende dalla differenza di temperature  $T_1$  e  $T_{ref}$ ; in questo caso,  $T_{ref}$  è  $0^{\circ}\text{C}$ .

Si noti che poiché i collegamenti dei conduttori del voltmetro hanno la stessa temperatura (dato che si trovano in una regione isoterma in cui si ha pressoché lo stesso valore di temperatura), le tensioni termoelettriche parassite generate in questi due punti sono uguali e opposte. Pertanto, l'errore aggiunto da queste connessioni è nullo.

In queste condizioni, se la temperatura di misurazione è superiore a  $0^{\circ}\text{C}$ , una termocoppia ha un'uscita positiva; se sotto  $0^{\circ}\text{C}$ , l'uscita è negativa. Quando la giunzione di riferimento e la giunzione di misurazione hanno la stessa temperatura, la tensione netta è zero.

Sebbene un riferimento al bagno di ghiaccio sia accurato, non è sempre pratico. Un approccio più pratico è misurare la temperatura della giunzione di riferimento con un sensore di temperatura a lettura diretta e sottrarre i contributi di tensione termoelettrica parassite. Questo processo è chiamato compensazione del “giunto freddo”.

Utilizzando la legge dei metalli interposti e facendo alcune semplici assunzioni, è possibile notare come la tensione misurata dalla scheda DAQ nella Figura 2 dipenda solo dal tipo di termocoppia, dalla tensione della termocoppia J1 e dalla temperatura del “giunto freddo”. La tensione misurata è infatti indipendente dalle tensioni termoelettriche parassite generate dalle giunzioni J2 e J3.

### Legge dei metalli interposti

Secondo la legge dei metalli interposti illustrata in figura 4, inserendo qualsiasi tipo di filo nel circuito della termocoppia non si ha alcun effetto sull'uscita fintanto che entrambe le estremità di quel filo abbiano la stessa temperatura (condizione garantita da una regione isoterma).

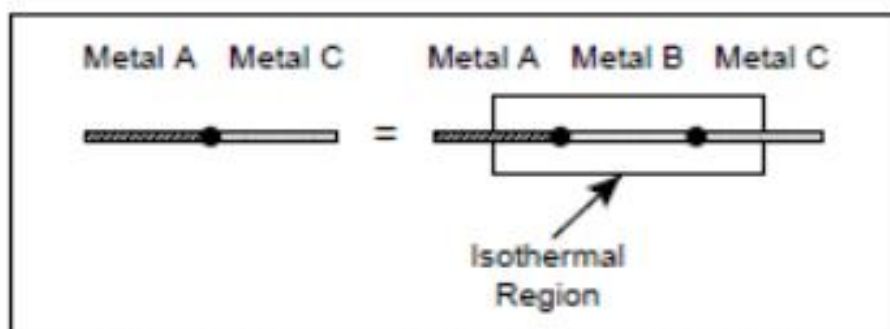


Figure 4. Thermocouple Law of Intermediate Metals

Si consideri il circuito in Figura 5. Questo circuito è simile al circuito precedentemente descritto in Figura 2, con la sola differenza che è inserito un breve tratto di filo di costantana dopo la giunzione J3 e si presume che le giunzioni siano mantenute alla stessa temperatura con un conduttore isoterma (ad esempio il rame). Supponendo che le giunzioni J3 e J4 abbiano la stessa temperatura, la legge dei metalli interposti ci dice che il circuito in Figura 5 è elettricamente equivalente al circuito in Figura 2. Di conseguenza, qualsiasi risultato preso dal circuito di Figura 5 si applica anche al circuito illustrato in Figura 2.

Nella Figura 5, le giunzioni J2 e J4 sono dello stesso tipo (rame-costantana) e poiché entrambi si trovano nella regione isoterma hanno anche la stessa temperatura. Le tensioni termoelettriche generate dalle due giunzioni sono uguali e opposte in segno quindi il loro contributo netto alla tensione misurata alla giunzione J1 è nullo. Le uniche giunzioni che hanno effetto sulla tensione totale misurata sono le giunzioni J1 e J3, giunzioni che sono entrambe di tipo ferro-costantana.



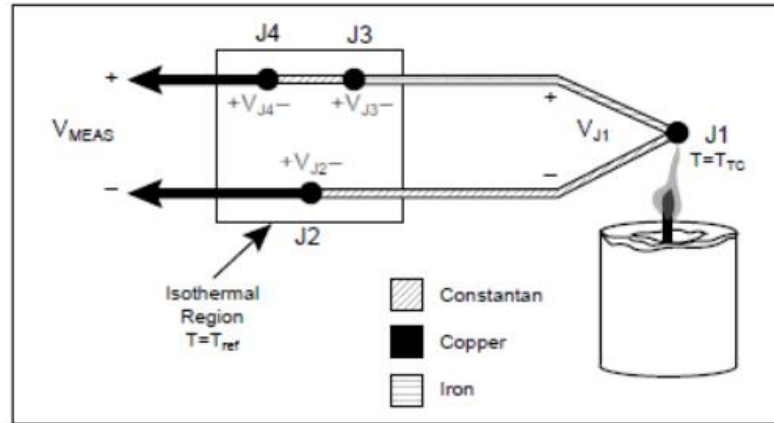


Figure 5. Inserting an Extra Lead in the Isothermal Region

Usando la notazione  $V_{JX}(T_y)$  per indicare la tensione generata dalla giunzione  $JX$  alla temperatura  $T_y$  il problema generale della termocoppia si riduce alla seguente equazione:

$$V_{meas} = V_{J1}(T_C) + V_{J3}(T_{ref})$$

dove  $V_{meas}$  è la tensione che la scheda DAQ misura,  $T_C$  è la temperatura della termocoppia in J1 e  $T_{ref}$  è la temperatura di riferimento della giunzione.

Esistono due tecniche per l'implementazione della compensazione del giunto freddo: compensazione hardware e compensazione software. Entrambe le tecniche richiedono che la temperatura sulla giunzione di riferimento sia rilevata con una lettura diretta da un sensore. Un sensore a lettura diretta ha un'uscita che dipende solo dalla temperatura del punto misurato.

I sensori a semiconduttore, i termistori o gli RTD vengono comunemente utilizzati per misurare la temperatura della giunzione di riferimento. Ad esempio, diverse morsettiere SCXI includono termistori che si trovano vicino ai morsetti a vite ai quali sono collegati i fili della termocoppia.

## Compensazione Hardware

Nel caso di compensazione hardware, è inserito nel circuito un generatore di tensione variabile (Cold Junction Compensation, CJC) utile per cancellare le tensioni termoelettriche parassite. Il generatore di tensione variabile genera una tensione di compensazione in accordo con la temperatura ambiente e aggiunge il corretto valore di tensione per eliminare gli effetti termoelettrici parassiti indesiderati. Con la compensazione hardware non è necessario considerare la temperatura in ingresso ai terminali del DAQ poiché tutte le termocoppie parassite sono state eliminate. L'unico svantaggio della compensazione hardware è che ogni tipo di termocoppia deve avere un circuito di compensazione separato che possa aggiungere il corretto valore di tensione di compensazione quindi ciò rende il circuito molto costoso. Per tale motivazione solitamente si preferisce la compensazione software.

## Compensazione software

In alternativa, è possibile usare il software per la compensazione del giunto freddo. Dopo aver misurato la temperatura della giunzione di riferimento con un sensore di temperatura a lettura diretta (come l'RTD), il software elabora il valore di tensione per eliminare l'effetto delle termocoppie parassite. Ci sono due modi per poter determinare la temperatura della termocoppia ad un dato valore di tensione  $V_{meas}$  e ad una temperatura di riferimento  $T_{ref}$  :

- 1) Ricordando che J1 e J3 sono dello stesso tipo ma in direzioni opposte:

$$V_{meas} = V_{J1}(T_C) + V_{J3}(T_{ref}) = V_{Tc}(T_{Tc}) - V_{Tc}(T_{ref})$$

Con la precedente relazione è possibile ricavare la tensione generata dalla termocoppia quando la temperatura del giunto freddo è pari a 0°C, ovvero  $V_{Tc}(T_{Tc}) = V_{Meas} + V_{Tc}(T_{ref})$ .

A questo punto è necessario:

- Misurare la temperatura del giunto freddo, indicata con  $T_{ref}$ .
- Convertire la temperatura misurata in una tensione equivalente  $V_{Tc}(T_{Ref})$  utilizzando la tabella NIST.
- Aggiungere la tensione equivalente determinata alla tensione misurata  $V_{meas}$  per ottenere la tensione che genererebbe la termocoppia quando il giunto freddo è ad una temperatura di 0°C.
- Convertire la tensione risultante in temperatura in modo da ottenere  $T_{Tc}$ .

- 2) Nel caso in cui ci siano piccole variazioni di temperatura la tensione prodotta in uscita dalla termocoppia è proporzionale alla temperatura misurata secondo il coefficiente di Seebeck, allora:

$$V_{Tc}(T_1) - V_{Tc}(T_2) \cong V_{Tc}(T_1 - T_2)$$

Questa assunzione è vera se  $T_1$  è molto vicina a  $T_2$ , ovvero nell'ipotesi di piccole variazioni di temperatura per le quali possiamo approssimare la curva caratteristica della termocoppia ad una retta lineare. Assumendo che la temperatura della termocoppia sia molto vicina alla temperatura di riferimento, è possibile scrivere che:

$$V_{meas} = V_{Tc}(T_{Tc} - T_{ref}).$$

A questo punto è necessario:

- Misurare la temperatura del giunto freddo  $T_{ref}$
- Convertire la tensione misurata  $V_{meas}$  in una temperatura utilizzando la relazione tra tensione e temperatura della termocoppia.
- Aggiungere la temperatura di riferimento della giunzione al valore precedentemente determinato in modo da ottenere la temperatura misurata dalla termocoppia.

## NIST: Coefficienti polinomiale per la conversione Tensione-Temperatura

	Thermocouple Type					
	E	J	K	R	S	T
<b>Range</b>	<b>0 to 1000 °C</b>	<b>0 to 760 °C</b>	<b>0 to 500 °C</b>	<b>-50 to 250 °C</b>	<b>-50 to 250 °C</b>	<b>0 to 400 °C</b>
a <sub>0</sub>	0.0	0.0	0.0	0.0	0.0	0.0
a <sub>1</sub>	1.7057035E-2	1.978425E-2	2.508355E-2	1.8891380E-1	1.84949460E-1	2.592800E-2
a <sub>2</sub>	-2.3301759E-7	-2.001204E-7	7.860106E-8	-9.3835290E-5	-8.00504062E-5	-7.602961E-7
a <sub>3</sub>	6.5435585E-12	1.036969E-11	-2.503131E-10	1.3068619E-7	1.02237430E-7	4.637791E-11
a <sub>4</sub>	-7.3562749E-17	-2.549687E-16	8.315270E-14	-2.2703580E-10	-1.52248592E-10	-2.165394E-15
a <sub>5</sub>	-1.7896001E-21	3.585153E-21	-1.228034E-17	3.5145659E-13	1.88821343E-13	6.048144E-20
a <sub>6</sub>	8.4036165E-26	-5.344285E-26	9.804036E-22	-3.8953900E-16	-1.59085941E-16	-7.293422E-25
a <sub>7</sub>	-1.3735879E-30	5.099890E-31	-4.413030E-26	2.8239471E-19	8.23027880E-20	
a <sub>8</sub>	1.0629823E-35		1.057734E-30	-1.2607281E-22	-2.34181944E-23	
a <sub>9</sub>	-3.2447087E-41		-1.052755E-35	3.1353611E-26	2.79786260E-27	
a <sub>10</sub>				-3.3187769E-30		
<b>Error</b>	<b>±0.02 °C</b>	<b>±0.05 °C</b>	<b>±0.05 °C</b>	<b>±0.02 °C</b>	<b>±0.02 °C</b>	<b>±0.03 °C</b>

## NIST: Coefficienti polinomiali per la conversione Temperatura-Tensione

	Thermocouple Type					
	E	J	K	R	S	T
Range	0 to 1000 °C	-210 to 760 °C	0 to 1372 °C	-50 to 1064 °C	-50 to 1064 °C	0 to 400 °C
c <sub>0</sub>	0.0	0.0	-17.600413686	0.0	0.0	0.0
c <sub>1</sub>	58.665508710	50.38118782	38.921204975	5.28961729765	5.40313308631	38.748106364
c <sub>2</sub>	4.503227558E-2	3.047583693E-2	1.85587700E-2	1.3916658978E-2	1.2593428974E-2	3.32922279E-2
c <sub>3</sub>	2.890840721E-5	-8.56810657E-5	-9.9457593E-5	-2.388556930E-5	-2.324779687E-5	2.06182434E-4
c <sub>4</sub>	-3.30568967E-7	1.322819530E-7	3.18409457E-7	3.5691600106E-8	3.2202882304E-8	-2.18822568E-6
c <sub>5</sub>	6.50244033E-10	-1.7052958E-10	-5.607284E-10	-4.62347666E-11	-3.314651964-11	1.09968809E-8
c <sub>6</sub>	-1.9197496E-13	2.09480907E-13	5.6075059E-13	5.007774410E-14	2.557442518E-14	-3.0815759E-11
c <sub>7</sub>	-1.2536600E-15	-1.2538395E-16	-3.202072E-16	-3.73105886E-17	-1.25068871E-17	4.54791353E-14
c <sub>8</sub>	2.14892176E-18	1.56317257E-20	9.7151147E-20	1.577164824E-20	2.714431761E-21	-2.7512902E-17
c <sub>9</sub>	-1.4388042E-21		-1.210472E-23	-2.81038625E-24		
c <sub>10</sub>	3.59608995E-25		NOTE A			

NOTE A: The equation for type K is  $v = c_0 + c_1T + c_2T^2 + \dots + c_9T^n + 118.5976e^{(-1.183432E-4)(T - 126.9686)^2}$