Controllo di un processo e lettura della temperatura ambiente tramite un personal computer

Studenti:
Simone De Carolis e *Simone Dionisi*

Tutori:
Ubaldo Denni e *M. Antonietta Frani*

Premessa

Il tema dello stage è di realizzare un circuito composto di un sistema retroazionato, che regola una temperatura, e da sensori che leggono la temperatura ambiente e di processo: entrambe sono scritte in un file e visualizzate su un grafico.

Scopo principale dello stage è di introdurre gli studenti in un ambiente di lavoro tipico della ricerca in fisica delle particelle elementari presso i Laboratori Nazionali di Frascati dell'INFN.

Il tema è stato scelto in maniera che fosse il più didattico possibile e anche perché attualmente le problematiche affrontate dal gruppo SISTEMI sono molto specifiche al di fuori del programma scolastico.

In ogni caso il tema scelto è alla base dei sistemi automatici di controllo e lettura di parametri ambientali "slow control".

È stata affrontata la retroazione a controllo proporzionale e l'algoritmo utilizzato è stato sviluppato sperimentalmente.

Il circuito è stato sviluppato con componentistica esistente in laboratorio ed il microcontrollore utilizzato è comandato tramite caratteri ASCII evitando così di "appesantire" il progetto con programmazione in assembler o C.

Il programma di gestione è LabVIEW tipico degli ambienti di ricerca, in ogni caso di facile apprendimento e utile alla realizzazione di prototipi.

Gli studenti sono stati inseriti in un gruppo di lavoro ed hanno partecipato alla progettazione, realizzazione, programmazione fino all'acquisizione dei dati del processo di controllo delle temperature.

Che cosa abbiamo imparato:

LabVIEW

LabVIEW è un programma applicativo che usa il linguaggio grafico G.

Il linguaggio di programmazione G può essere considerato l'equivalente del codice sorgente nella programmazione tradizionale, ma differisce da quest'ultima perché crea programmi sottoforma di diagramma a blocchi con flusso di dati, mentre i linguaggi tradizionali creano programmi testuali con flusso di codice.

I programmi realizzati con LabVIEW sono chiamati strumenti virtuali (**VI**) perché la loro apparenza e il loro funzionamento può imitare quello di strumenti reali.

Un **VI** consiste in un interfaccia utente interattiva, **pannello frontale**, un diagramma di flusso, **diagramma a blocchi**, un'**icona** e dei **connettori**.

L'interfaccia utente di un VI è simile al **pannello frontale** di uno strumento fisico; possiamo introdurre dei valori in ingresso e visualizzare i risultati generati dal diagramma a blocchi. Per analogia con il pannello frontale di uno strumento vero, gli ingressi sono chiamati **controlli** e le uscite **indicatori**. Si possono utilizzare diversi tipi di controlli e indicatori come manopole, interruttori, bottoni, istogrammi, grafici, array, cluster e così via per rendere il pannello frontale più chiaro all'utilizzatore.

Il **diagramma a blocchi** è il luogo in cui è creato il programma è composto di **nodi**, **terminali** e **collegamenti**.

I **nodi** sono gli elementi che consentono l'esecuzione del programma, operazioni matematiche I/O su file, formattazione di stringhe, sub VI, cicli etc.

I **terminali** sono collegamenti che permettono il passaggio dei dati tra il diagramma a blocchi e il pannello frontale e tra i nodi del diagramma a blocchi stesso.

I **collegamenti** sono i percorsi dei dati che vanno da un terminale all'altro e corrispondono alle variabili dei linguaggi tradizionali.

I dati fluiscono in una sola direzione, da un terminale sorgente ad uno o più terminali destinazione, questo è il principio che regola l'esecuzione di un programma in LabVIEW, ed è chiamato **Data Flow**.

Un nodo è eseguito solo quando i dati sono disponibili a tutti i suoi terminali di ingresso, finita l'esecuzione, il nodo fornisce i dati a tutti i suoi terminali di uscita.

L'**icona** è la rappresentazione grafica di una VI, serve ad identificarla all'interno di una VI di alto livello, ed in questo caso è chiamata **subVI**.

Ai **Connettori** sono collegati i terminali di ingresso e uscita corrispondenti ai controlli ed indicatori di quel VI o subVI.

La potenza di LabVIEW sta nella natura gerarchica del VI. Dopo aver creato un VI, possiamo utilizzarlo nel diagramma a blocchi di un VI ad alto livello. Non c'é limite al numero di livelli nella gerarchia.

Una volta creato un VI è possibile utilizzarlo come sottoprogramma (**subVI**) nel diagramma a blocchi di uno strumento a più alto livello, oppure se un diagramma a blocchi ha un gran numero di nodi o una parte di programma è ripetuta più volte, conviene creare uno strumento virtuale di livello inferiore, per mantenere una leggibilità e una maggiore comprensione del programma ed una maggiore facilità di ricerca degli errori.

Comunicazione Seriale

Nella metà degli anni '60 quasi tutti i collegamenti dei computers per accessi remoti ad un unità centrale avvenivano attraverso linee telefoniche.

L'RS-232 aveva originariamente la funzione di standardizzare le interconnessioni tra terminali ed host computers. I modems, collegati ai computers tramite RS-232 erano utilizzati per tradurre segnali digitali in segnali analogici, i quali erano trasmessi su linee telefoniche.

In questo periodo ciascun produttore usava una configurazione differente per interfacciare un DTE (Data Terminal Equipment) con una DCE (Data Communications Equipment). Cavi, connettori e livelli di tensione erano differenti e incompatibili, così l'interconnessione di due apparecchi elettronici costruiti da ditte diverse avevano bisogno di convertitori, di speciali cavi e connettori.

Nel 1969, EIA con Bell Laboratories stabilirono una serie di standard per interfacciare computers e comunicare dati tra apparecchi elettronici costruiti da ditte diverse. Questo standard definisce in poche parole le caratteristiche elettriche, meccaniche e funzionali della porta seriale. Le caratteristiche elettriche includono parametri come i livelli di tensione e di impedenza; quelle meccaniche descrivono i pin e la loro connessione mentre quelle funzionali definiscono le funzioni dei differenti segnali per essere utilizzati.

Questo standard è chiamato brevemente RS-232 ed è largamente adottato dai produttori di computers e terminali. Dagli anni '80 tutte le varie industrie di microcomputers usano anche l'RS-232 come standard di comunicazione.

La definizione "Interfaccia Seriale" nasce dal modo in cui i dati sono trasmessi, cioè i bit che costituiscono l'informazione sono trasmessi uno per volta su di un solo filo.

Nella trasmissione seriale esistono due metodologie di funzionamento: la trasmissione sincrona e la trasmissione asincrona.

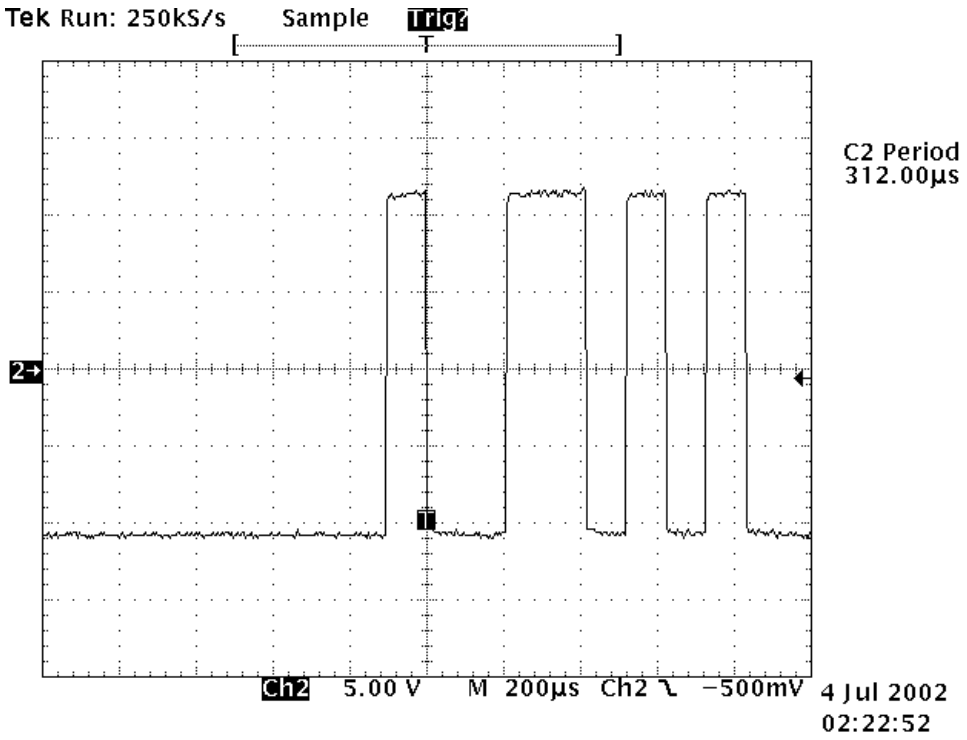
Nel caso della trasmissione **sincrona,** innanzitutto, occorrono minimo due fili che sono rispettivamente utilizzati per la trasmissione/ricezione dei dati e per la trasmissione/ricezione del *clock*. Quando una stazione sincrona trasmette il proprio dato seriale verso una stazione ricevente provvede a delimitare la sua validità trasmettendo un impulso di clock affinché la stazione ricevente sia informata di prendere in considerazione solo il valore del dato trasmesso nell'intervallo di tempo dell'impulso di clock ricevuto.

La trasmissione seriale **asincrona**, invece, pur essendo molto più limitata nella velocità, risulta più utilizzata nelle interfacce RS-232 dei personal computers. Il principio di funzionamento si basa sulla trasmissione dei dati (8 bit) su di un filo e sulla ricezione degli stessi su di un altro. Il trucco consiste nell'inviare il nostro dato compreso tra un bit di start (bit di sincronismo) e un bit di stop (bit di fine trasmissione dato). In alcuni casi, se il dato da inviare viene settato a 7 bit, possiamo accodarci un bit di parità che sarà utilizzato dalla stazione ricevente per effettuare un semplice controllo sulla correttezza del dato trasmesso.

Le unità di misura della velocità di trasmissione sono essenzialmente due: il *baud* ed il *bit per secondo* (bps).

Il baud indica il numero di transizioni al secondo che avvengono sulla linea, il bit per secondo indica quanti bit al secondo sono trasmessi lungo la linea. Nel caso di trasmissione binaria (un livello alto ed uno basso) le due unità di misura coincidono.

*In figura è visualizzato un segnale RS-232 ad 8 bit, no parity, 9600 bps, del carattere ascii "**s**".*



L'ampiezza del segnale va da un +12V chiamato space (livello logico basso) ad un segnale basso –12V chiamato mark (livello logico alto). Ogni transizione è di 104µS (pari ad 1/9600 bps).
Il segnale all'inizio si trova basso (nessun dato in transito); la prima transizione da basso ad alto indica l'inizio della trasmissione (inizia il bit di start con una lunghezza di 104µS). Dopo altri 104µs segue il bit meno significativo (LSB) e così via fino al bit più significativo (MSB). Infine c'é un periodo di riposo che è di almeno di 208µS prima che inizi un nuovo dato.
Le varianti possibili sono le seguenti:

- La velocità con cui può lavorare una porta seriale con un personal computer varia da un minimo di 110 ad un massimo di 115200 bps
- Invece di 8 bit si può lavorare anche con 6 o 7 aggiungendo alla fine un bit di parità in grado di verificare la correttezza del dato ricevuto. Esistono cinque tipi di parità:
  1. None: nessun tipo di parità
  2. Pari: il numero di mark è sempre pari
  3. Dispari: il numero di mark è sempre dispari
  4. Mark: il bit di parità vale sempre mark
  5. Space: il bit di parità vale sempre space
- Alla fine la linea rimane nello stato di riposo per almeno 1 o 1.5 o 2 bit.

I parametri elettrici RS-232 sono i seguenti: la tensione di uscita deve essere compresa per il LL0 tra 5V e 25V e il LL1 tra –5V e –25V. Il ricevitore deve funzionare con tensioni di ingresso comprese tra 3V e 25V per il LL0 e tra –3V e –25V per il LL1.

La corrente di uscita deve essere di almeno 1.6mA.Lo slew-rate deve essere minore di 30V/uS per evitare eccessive emissioni di EMC.

Tabella 1: Comparazione tra interfacce tipiche.

Interface	**Format**	**Number of Devices (maximum)**	**Length (maximum, feet)**	**Speed (maximum, bits/sec.)**
RS-232 (EIA/TIA-232)	asynchronous serial	2	50-100	20k (115k with some drivers)
RS-485 (TIA/EIA-485)	asynchronous serial	32 unit loads	4000	10M
IrDA	asynchronous serial infrared	2	6	115k
Microwire	synchronous serial	8	10	2M
SPI	synchronous serial	8	10	2.1M
I^2C	synchronous serial	40	18	400k
USB	asynchronous serial	127	16	12M
Firewire	serial	64	15	400M
IEEE-488 (GPIB)	parallel	15	60	1M
Ethernet	serial	1024	1600	10M
MIDI	serial current loop	2	15	31.5k
Parallel Printer Port	parallel	2, or 8 with daisy-chain support	10-30	1M

Che cosa abbiamo utilizzato:

ITC-232

Il microcontrollore ITC232 è stato realizzato dalla ditta RMV canadese (http://www.rmv.com) verso la meta degli anni '90, ma sempre attuale per la sua facilità d'uso e versatilità.

È un circuito integrato multifunzione per acquisizioni dati e controllo con un'interfaccia seriale intelligente, si comanda facilmente da un terminale tramite una porta seriale di un computer (via RS-232).

L'ITC-232 ha 32 linee di input/output organizzate in 5 porte, che possono essere lette e/o scritte con semplici comandi ASCII, non sono richiesti complessi protocolli di comunicazione, perciò è facile scrivere il proprio programma, in qualsiasi linguaggio, ed il fatto di mandare comandi tramite un terminale facilità il debugging del sistema prima o durante la scrittura del proprio programma.

Caratteristiche tecniche:

- I comandi ASCII e dati sono inviati sulla porta seriale RS232, con velocità che va da 300 a 115.200 baud selezionabile via hardware e software, in formato decimale, esadecimale o binario
- 3 porte a 8 bit I/O che possono essere configurate come input o output
- Una porta seriale SPI
- Misura diretta della capacità o resistenza
- I bits più significativi delle tre porte digitali possono essere configurate per comandare step-motors (in modo monofase, bifase e mezzo-passo con una frequenza che va da 10 a 4000 steps/sec)
- Un'uscita in PWM da 10 a 10.000 Hz, 0-100 % duty cycle
- Due ingressi di interrupt.

Esempio di comandi:

Configuriamo i 3 bit più significativi della porta A come Input e i 5 bit meno significativi come Output:

Pca 224

Alziamo il bit 2 della porta A:

Pwa2

Abilitiamo la porta SPI:

Pcsa128

Abilitiamo i 4 bit più significativi della porta B per comandare uno step-motor con half-step 1000 steps\sec e 100 stop delay:

Sebh 1000;100

Facciamo ruotare a sinistra il motore di 10000 steps:

Sbl 10000

Configuriamo il PWM con frequenza 5000 ed un duty cycle del 80%:

W5000;80

Sensori temperatura

Attualmente ci sono molte tecnologie di costruzione per i sensori di temperatura, quelle più comuni sono RTDs, termocoppie, termistori e sensori ICs.

RTDs (Resistance Temperature Detector)

Consistono in una resistenza di platino, depositata su un substrato o un filo avvolto su una bobina, la temperatura varia la resistenza in maniera lineare su una gamma tra 0°C e 400 °C. Approssimativamente il loro range di lavoro è tra –250°C e 800 °C, la loro ripetibilità è elevata, richiedono un'elettronica accurata per il loro condizionamento e non sono economiche.

TERMISTORI

È un altro tipo di sensore resistivo, ma a basso costo; debbono essere selezionati secondo la temperatura di lavoro desiderata ed in ogni caso non si può usare un termistore singolo per un range di temperatura grande a causa della forte non linearità. Possono essere utilizzati per semplici applicazioni come termostati e controllori di temperatura economici, per realizzare queste applicazioni si ha bisogno di pochi componenti: il termoresistore, comparatore e qualche resistenza.

In commercio n'esistono con coefficiente di temperatura sia positivo sia negativo.

TERMOCOPPIE

Una termocoppia consiste in una giunzione "intima" tra due fili di differente materiale, per esempio in una termocoppia di tipo J la giunzione è tra ferro e costantana, la quale genera una tensione molto piccola di decine di microVolt per grado, ed esige un'elettronica di condizionamento con un basso offset.

La loro non linearità richiede un circuito di linearizzazione; quando si connettono i cavi di rame, per portare il segnale all'amplificatore, si creano altre giunzioni termiche (giunzione fredda) che contribuiscono a variare il segnale generato dalla termocoppia, si debbono utilizzare varie tecniche e trucchi per migliorare la precisione tenendo conto della giunzione fredda.

Anche se bisogna avere particolari accortezze per il loro funzionamento, le termocoppie hanno una gran popolarità, in parte dovuta alla loro bassa capacità termica e il loro vasto range di operatività, che può arrivare fino a 1700 °C.

SENSORI ICs

I sensori di temperatura a circuito integrato differiscono significamene dagli altri tipi almeno per due motivi.
Il primo è il range di temperatura operativa (da -55°C a 150°C) molto più piccolo rispetto agli altri sensori e il secondo è il funzionamento.

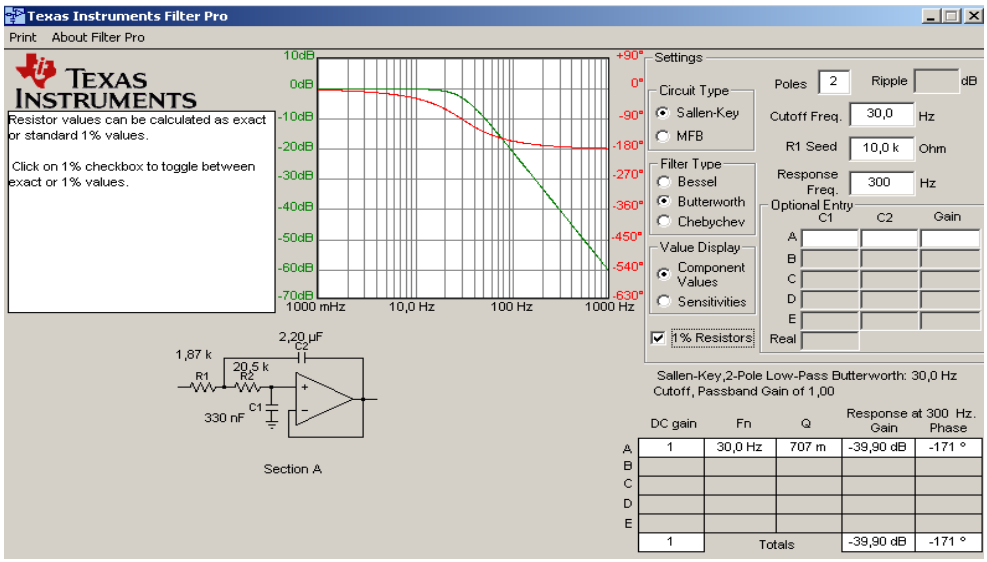
Questi sensori sono circuiti integrati e possono includere nel loro interno circuiti più o meno complessi che processano il segnale.

Noi abbiamo usato il tipo LM35DZ che può essere alimentato da 4 a 20 Volt, con un segnale in uscita di 10mV/°C, il range di temperatura è tra 0°C e 100°C.

Filtri

Per impedire che il rumore elettrico influenzi il segnale che proviene dai sensori di temperatura si usa interporre, tra sensori e ingressi dell'ADC, un filtro.

Per la progettazione del filtro attivo passa basso abbiamo usato un programma chiamato *FilterPro*.

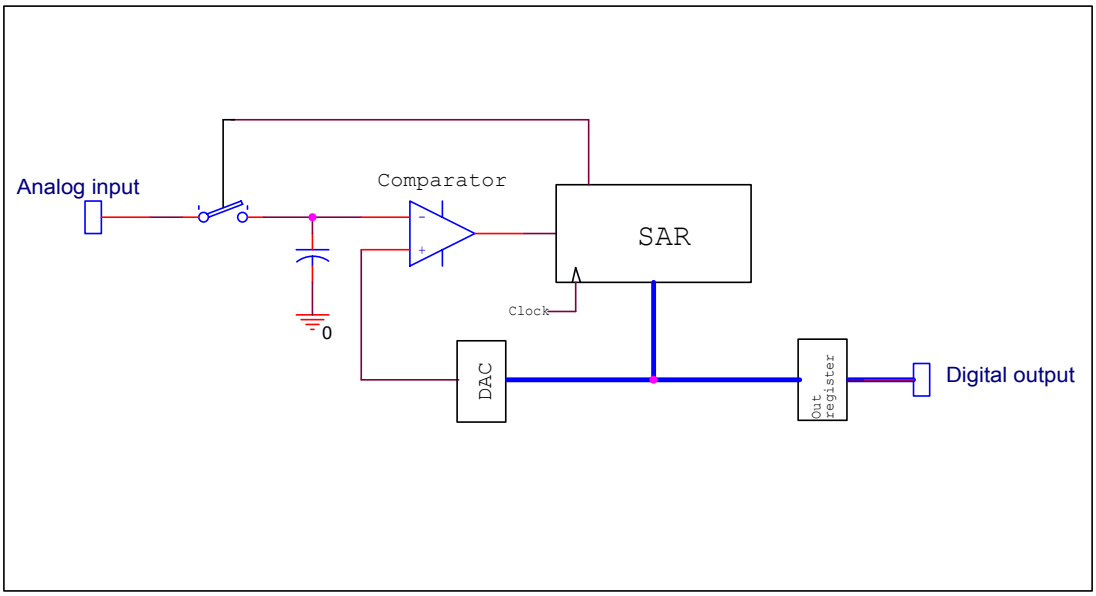


È stato sufficiente scegliere il tipo di circuito (Salen-Key), il numero di poli (2) e la frequenza di cut-off (30 Hz). Dalla simulazione vediamo che il filtro lavora egregiamente per il nostro scopo, infatti, scegliendo la frequenza di lavoro del PWM a 1KHz si ha un'attenuazione, della frequenza fondamentale, di 60 dB.

ADC (Analog digital conversion)

Il convertitore A/D usato ha 11 ingressi con una risoluzione di 12 bit, il C.I. si chiama TLC2543 è del tipo ad approssimazioni successive, per interfacciarlo è necessaria una porta seriale sincrona SPI (Serial Peripheral Interface).

Il convertitore con architettura ad approssimazioni successive è un ADC a media velocità e a basso costo per applicazioni "general purpose".



La figura mostra lo schema a blocchi semplificato del convertitore, il controllo del sistema è il circuito SAR (The successive-approximation estimator), prima di iniziare la conversione isola il condensatore dall'ingresso analogico, tramite un convertitore DAC (digital analog conversion) mette la soglia del comparatore a ½ della tensione di riferimento V_{ref}.

Il comparatore immette nel circuito SAR il risultato della comparazione, se la soglia del comparatore non è uguale alla tensione d'ingresso il circuito SAR seguiterà a sommare o sottrarre alla soglia del comparatore "pesi" della V_{ref}, che ad ogni ciclo diminuisce di $^1/_4$ ... $^1/_8$ ... $^1/_{16}$...

La conversione finisce quando la tensione della soglia, generata all'ultimo ciclo dal DAC, è uguale alla tensione d'ingresso, l'uscita digitale è pronta per essere letta, il condensatore è collegato di nuovo all'ingresso analogico ed il sistema è pronto per iniziare un nuovo ciclo.

L'ADC ha un ingresso di chip-select che è usato per l'abilitazione alla scrittura e lettura; con la transizione da LL1 a LL0 inizializza tutti i registri interni.

Con una porta seriale SPI si può leggere e/o scrivere più di un dispositivo, per questo si ha bisogno del chip-select che comandiamo tramite una porta digitale dell'ITC-232.

La porta SPI è uno shift register circolare formato da 8 bit; per leggere l'ADC a 12 bit dobbiamo configurare la lunghezza dei dati in uscita ad 8 bit con i primi bit ad uscire più significativi, in modo che con due letture in rapida successione, sommando e traslando opportunamente le due parole possiamo ricostruire il dato a 12 bit.

Da tener presente che ogni volta che cambiamo il numero del canale la prima lettura dei dati deve essere eliminata.

Cosa abbiamo realizzato:

Il circuito

Cuore del circuito è naturalmente il microcontrollore ITC-232, dal punto di vista hardware è sufficiente dargli 5 Volt ed adattargli, tramite un C.I. MAX-232, i livelli di tensione tra lo standard RS-232 e la TTL.

Abbiamo dei sensori da leggere quindi usiamo un ADC nel nostro caso un C.I. della Texas TLC2543 con 11 ingressi a 12 bits che si può leggere e scrivere tramite porta seriale SPI; utilizziamo soltanto i primi due mettendo gli altri ingressi non utilizzati riferiti a massa.

Mettiamo sull'ingresso del REF+ 1,235 Volts, tramite il C.I. LT1004, e l'altro ingresso REF- a massa in questo modo abbiamo 1235 mV "affettati" dai 12 bits di conversione.

La nostra precisione sarà di +/- 1bit quindi +/- 0,3 mV che corrispondono a +/- 0,03 °C; possiamo dire che la precisione dell'elettronica è superiore alle nostre esigenze.

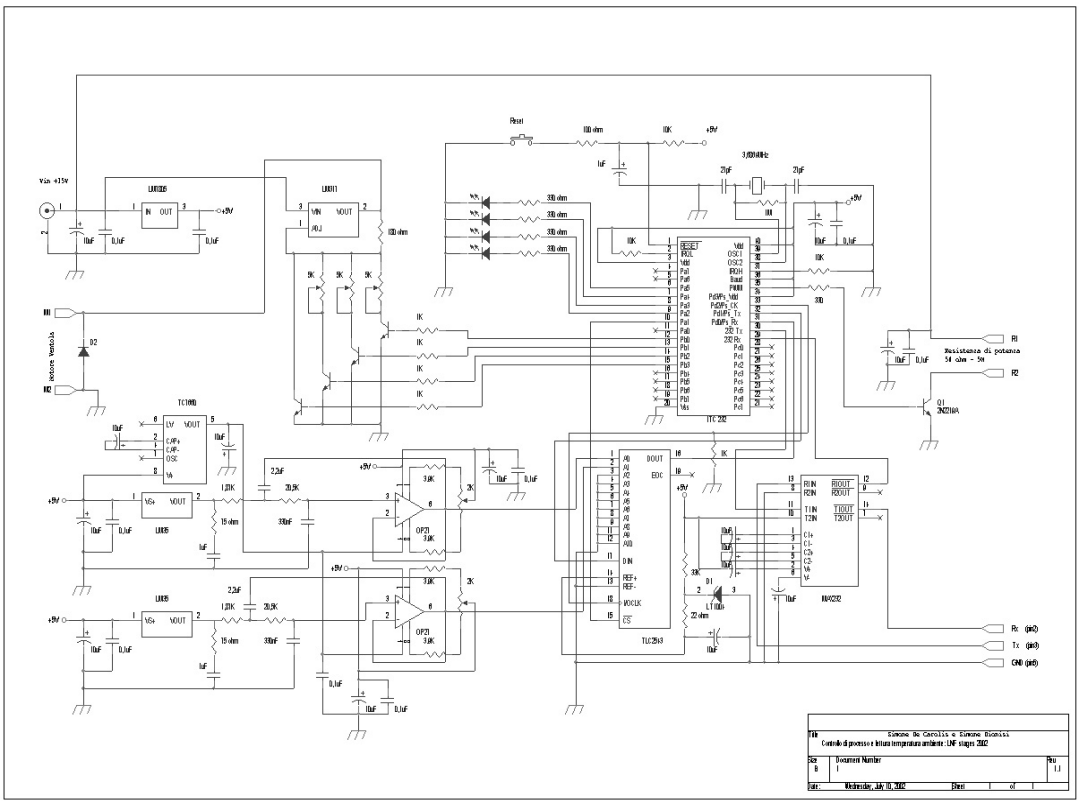
Tra i sensori di temperatura e ADC mettiamo dei filtri attivi passa basso per eliminare il più possibile rumore dal mondo esterno e soprattutto dalla nostro circuito perché siamo noi stessi generatori di rumore con il riscaldamento della resistenza.

Possiamo riscaldare più o meno la resistenza variando il duty-cycle della frequenza che esce dal pin n. 35 dell'ITC-232 e va, tramite una resistenza di adattamento, alla base del transistor, facendolo lavorare in saturazione ed interdizione, in modo da far passare corrente o no nella resistenza di potenza.

Per la riduzione del rumore abbiamo collegato a "stella" l'alimentazione della resistenza di potenza e l'alimentazione degli integrati, mettendo vicino tra pins di alimentazione dei C.I. e la massa condensatori appropriati (di solito è una coppia formata da un condensatore ceramico da 0,1 uF e da un condensatore al tantalio da 1 o 10 uF per "assorbire" sia variazioni rapide sia lente sulle piste di alimentazione), ulteriori manovre per la riduzione del rumore vengono fatte anche via software mediando dei campionamenti.

Sulla resistenza di potenza abbiamo incollato (con una colla speciale che fa passare il calore) il sensore di temperatura LM35DZ che viene letto, dopo essere stato filtrato, tramite l'ingresso A0 dell'ADC.

Sui C.I. OP26 con i quali vengono costruiti i filtri attivi vengono messi anche dei trimmers per regolare, eventualmente, l'offset dell'integrati stessi; naturalmente per non avere un alimentatore duale si è preferito fare localmente la tensione negativa tramite il C.I. TLC7660. Per "raffreddare" la resistenza si è pensato di utilizzare una piccola ventola usata normalmente sulle CPU dei PC; con quattro uscite digitali dell'ITC-232 si saturano o interdicono quattro transistor facendo così variare la tensione di uscita del C.I. LM317 che alimenta la ventola facendola girare più o meno veloce, in pratica si è costruito un rudimentale DAC.



Schema Elettrico

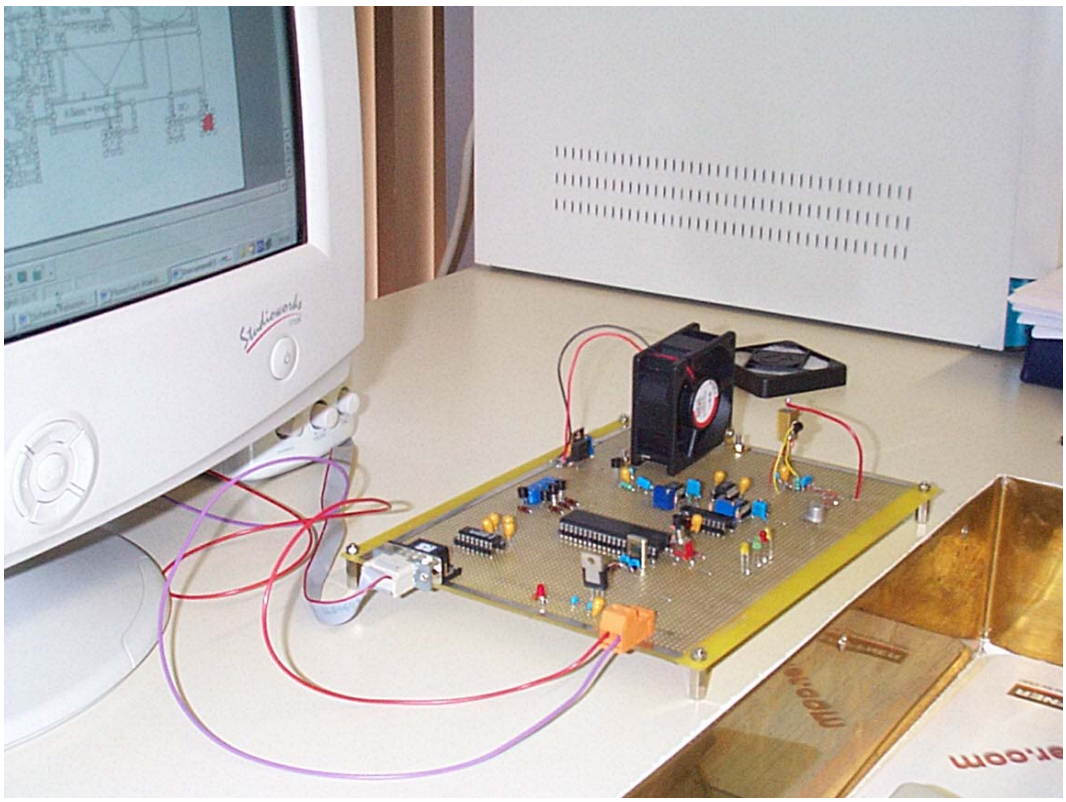


Foto Circuito

Il Software

Per la gestione del processo di retroazione e lettura temperatura ambiente utilizziamo LabVIEW.

Per primo cosa configuriamo la porta seriale del PC (COM1: 9600, 8, n, 1) e il microcontrollore ITC232 (porte digitali A e B come output e abilitato la porta SPI).

Inizializziamo il nostro sistema (CS alto e ventola spenta), azzeriamo tutte le variabili e costruiamo una path per aprire un file dove memorizziamo i dati .

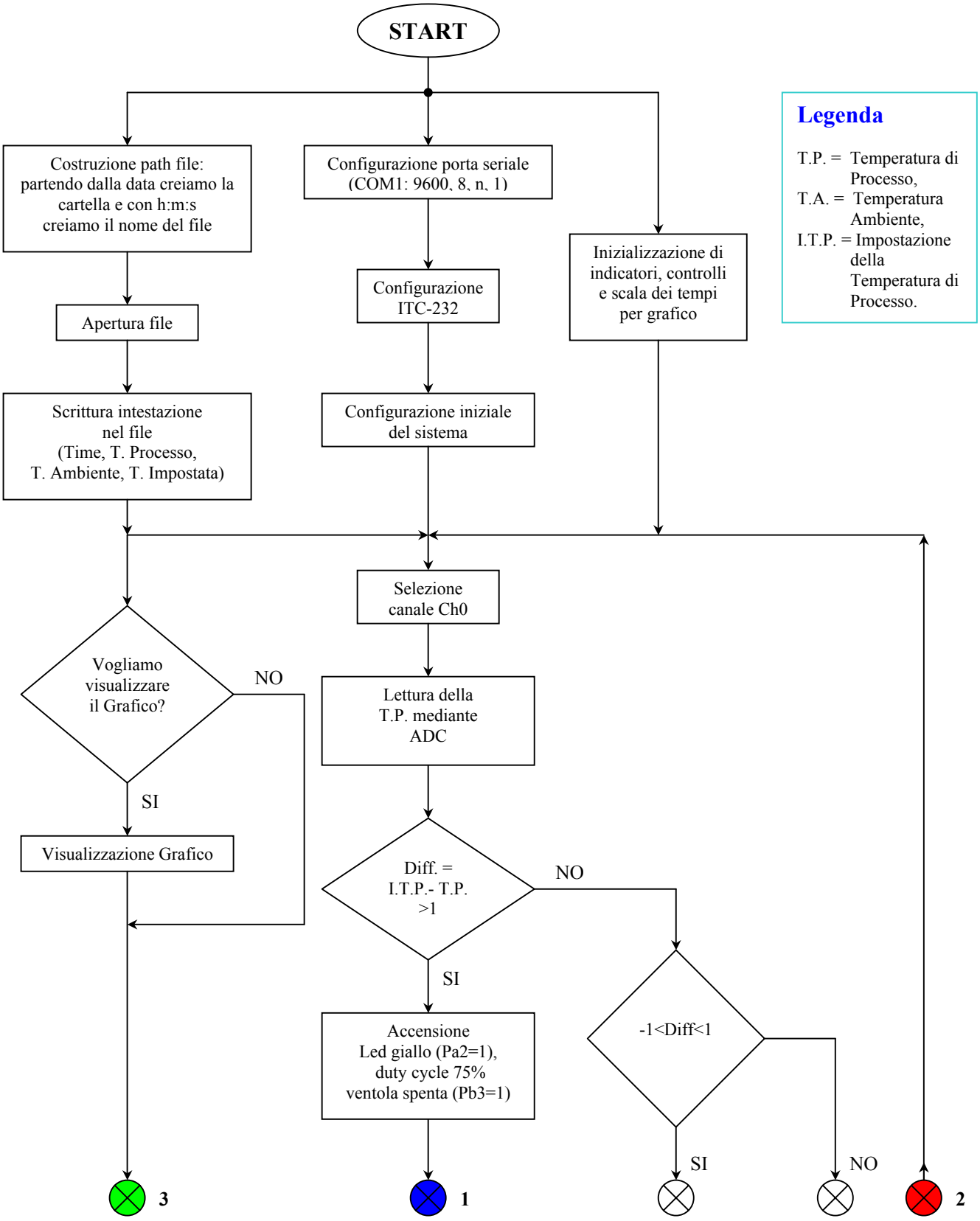
Fatto questo entriamo nel ciclo di lavoro, selezioniamo il canale (CH0 dell'ADC) e leggiamo la temperatura di processo, la confrontiamo con la temperatura impostata ed agiamo sul processo come descritto dalla seguente tabella:

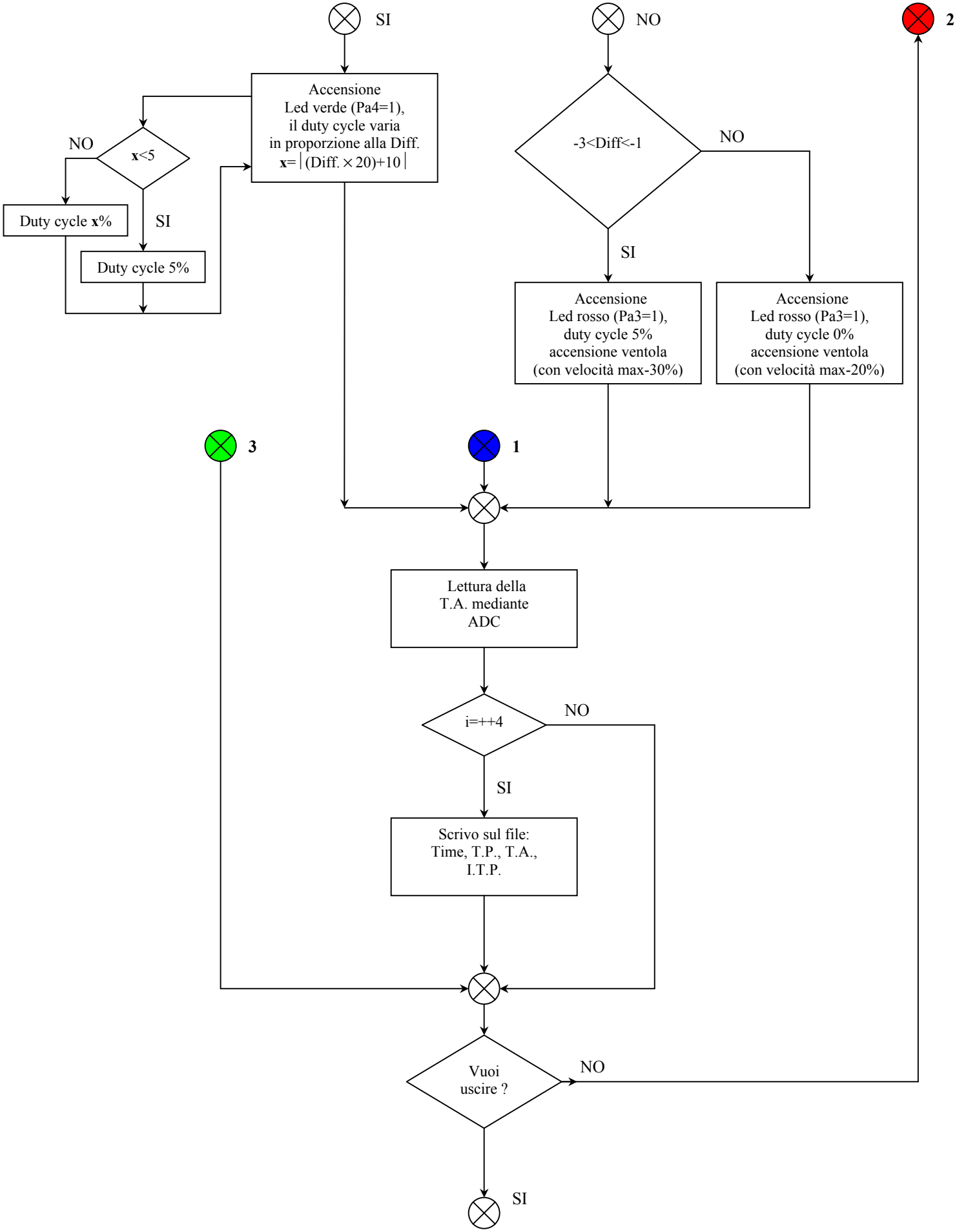
(T.I.-T.P.) > 1	LED giallo ON Ventola OFF PWM 75%
1 < (T.I.-T.P.) > -1	LED verde ON Ventola OFF PWM proporzionale a (T.I.-T.P.)
-3 < (T.I.-T.P.) > -1	LED rosso ON Ventola ON (Vel_{max}- 30%) PWM 5%
(T.I.-T.P.) < -3	LED rosso ON Ventola ON (Vel_{max}-20%) PWM 0%

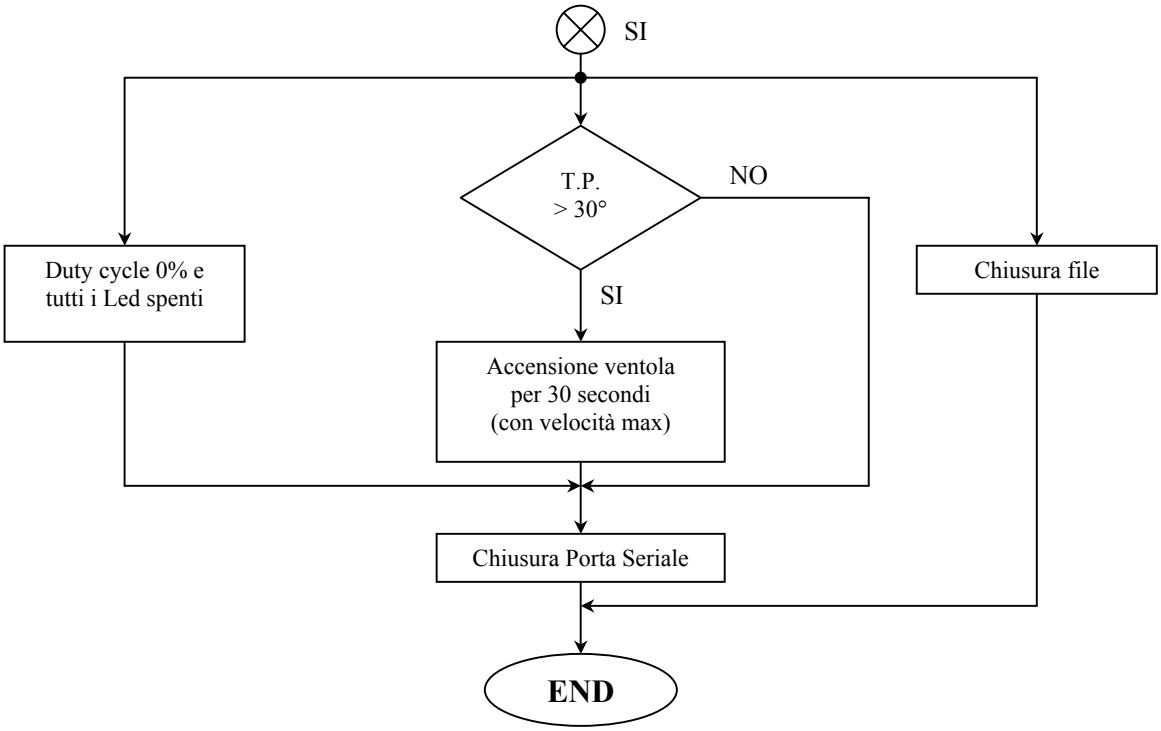
Leggiamo la temperatura ambiente (CH1 dell'ADC) e ogni 5 cicli di lavoro andiamo a memorizzare i dati su un file (Ora, T. di processo, T. ambiente e T. impostata). Durante il ciclo di lavoro possiamo scegliere, tramite un tasto, la visualizzazione dei dati su un grafico. Quando decidiamo di fermare il programma, tramite un tasto di stop, controlliamo se la temperatura di processo è maggiore di 30 gradi se si accendiamo la ventola a Vel_{max} per 30 secondi in modo da raffreddare la resistenza.

Le Flowcharts

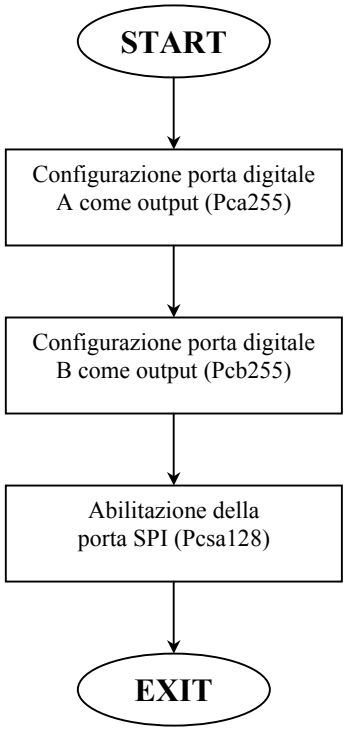
Controllo Temperatura.vi



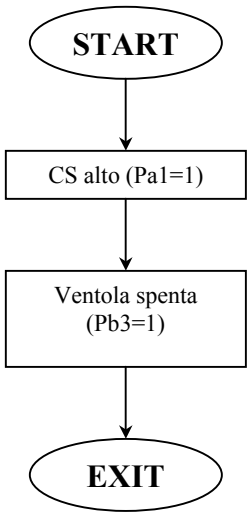

SI
NO
2
Accensione
Led verde (Pa4=1),
il duty cycle varia
in proporzione alla Diff.
x= | (Diff. × 20)+10 |
NO
x<5
Duty cycle x%
SI
Duty cycle 5%
-3<Diff<-1
NO
SI
Accensione
Led rosso (Pa3=1),
duty cycle 5%
accensione ventola
(con velocità max-30%)
Accensione
Led rosso (Pa3=1),
duty cycle 0%
accensione ventola
(con velocità max-20%)
3
1
Lettura della
T.A. mediante
ADC
i=++4
NO
SI
Scrivo sul file:
Time, T.P., T.A.,
I.T.P.
Vuoi
uscire ?
NO
SI



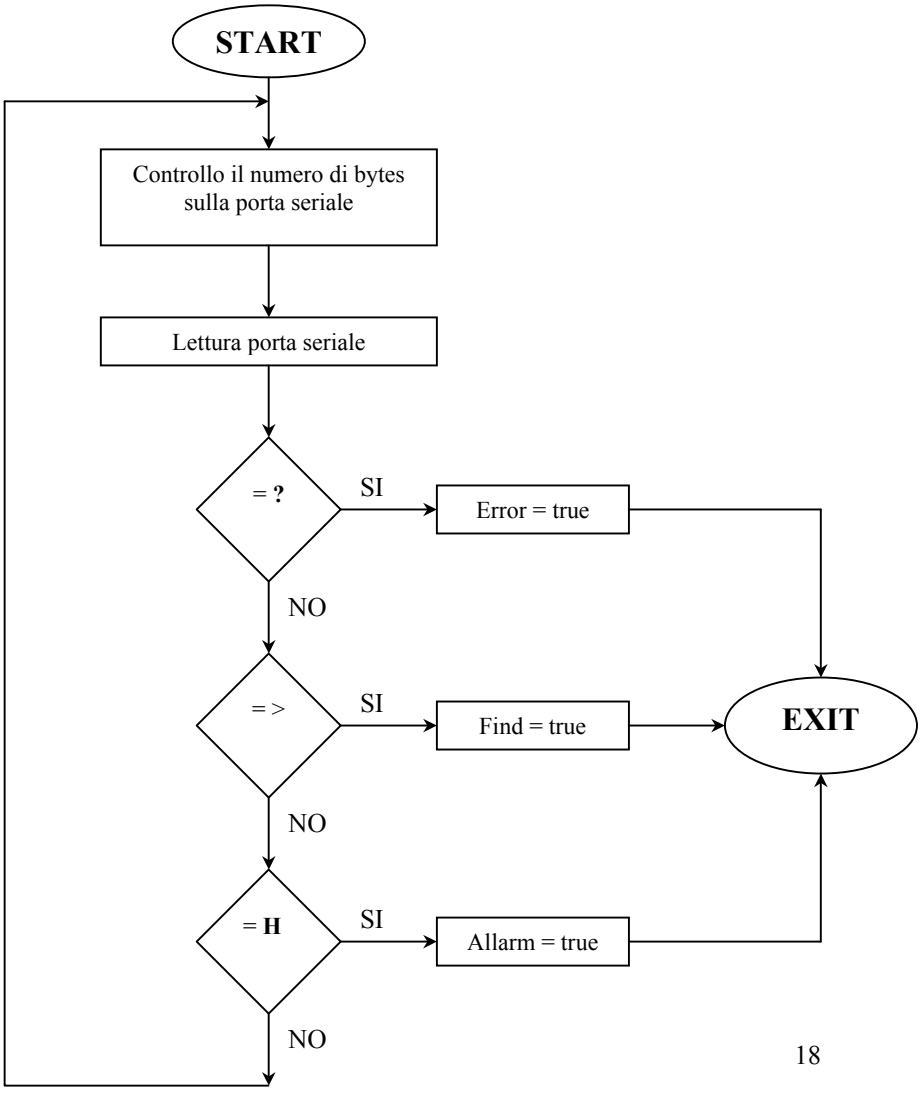
Configurazione ITC-232.vi



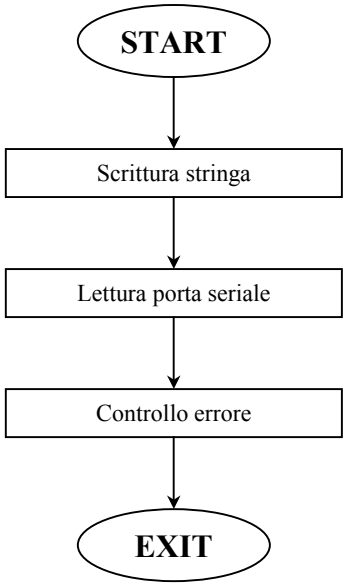
Configurazione Ventola e CS ADC .vi



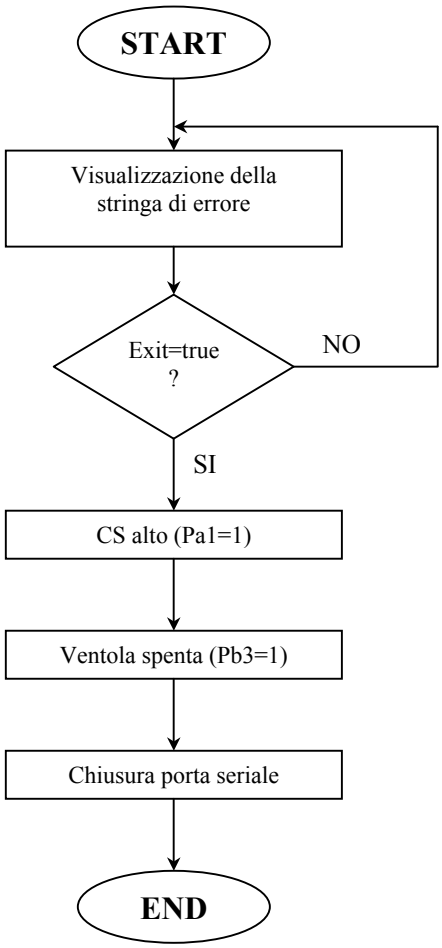
Read Serial.vi



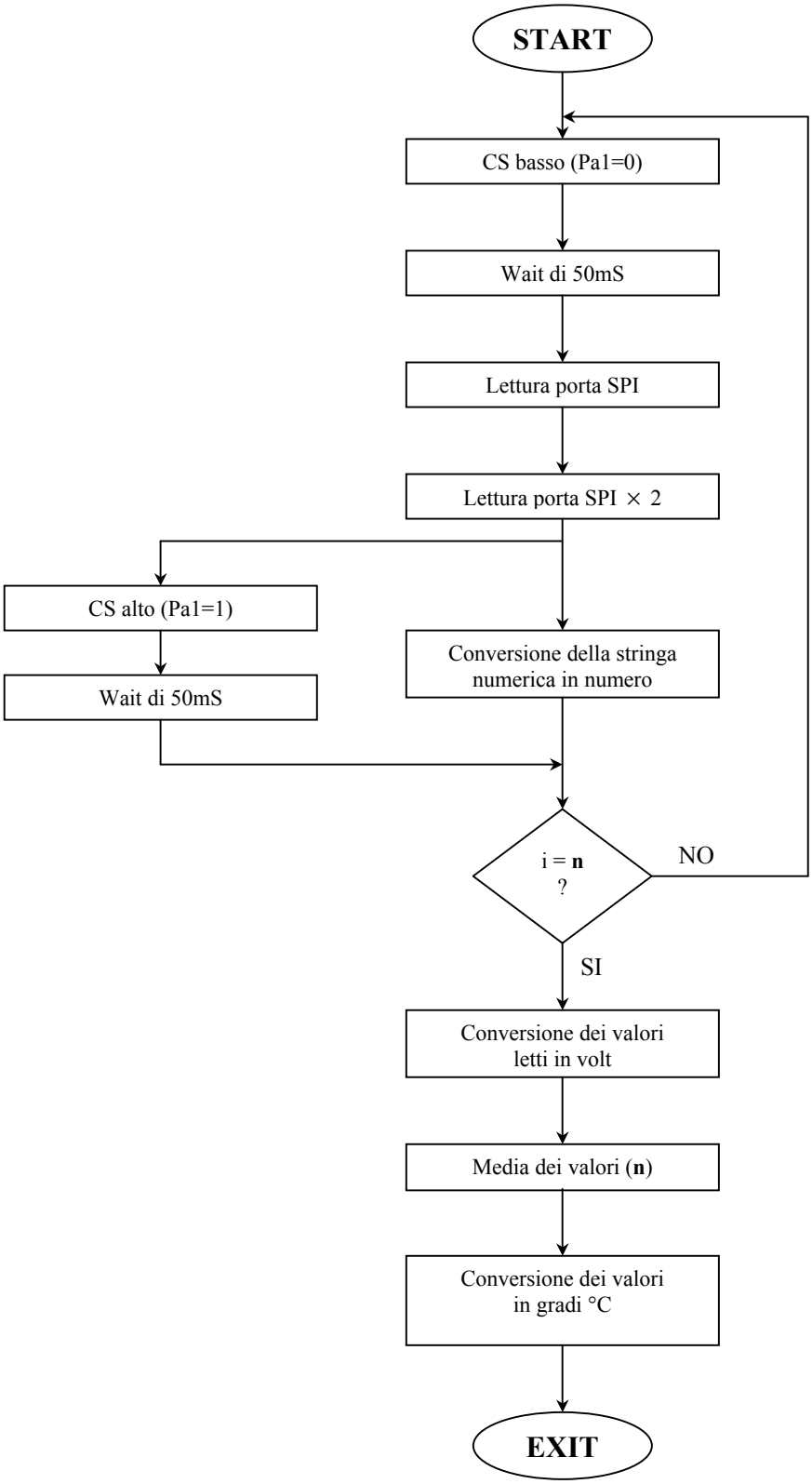
Write Serial.vi



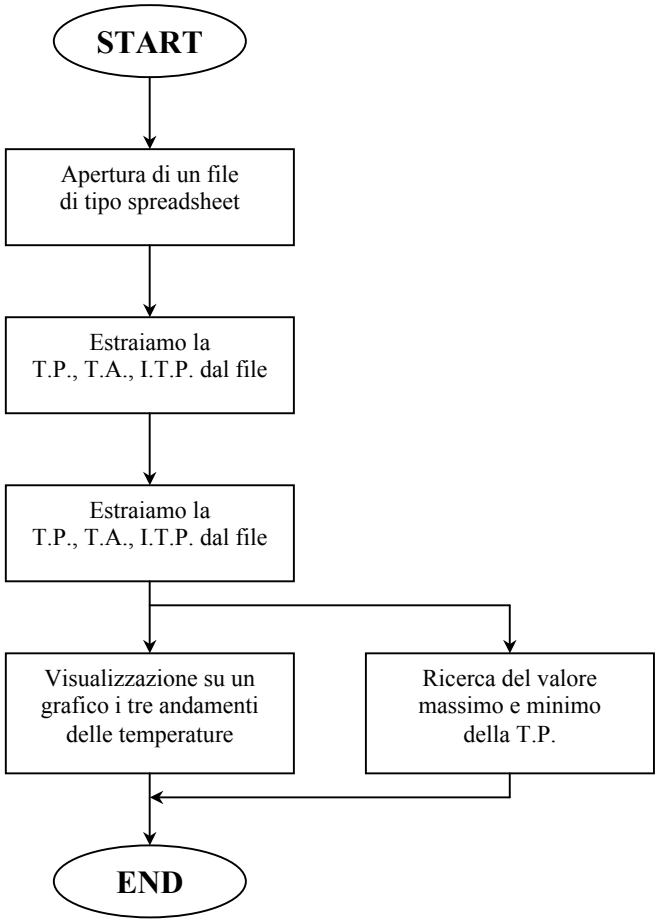
Controllo Errore.vi



Lettura ADC.vi



Lettura Dati.vi



Controllo Temperatura.vi

Pannello Frontale :

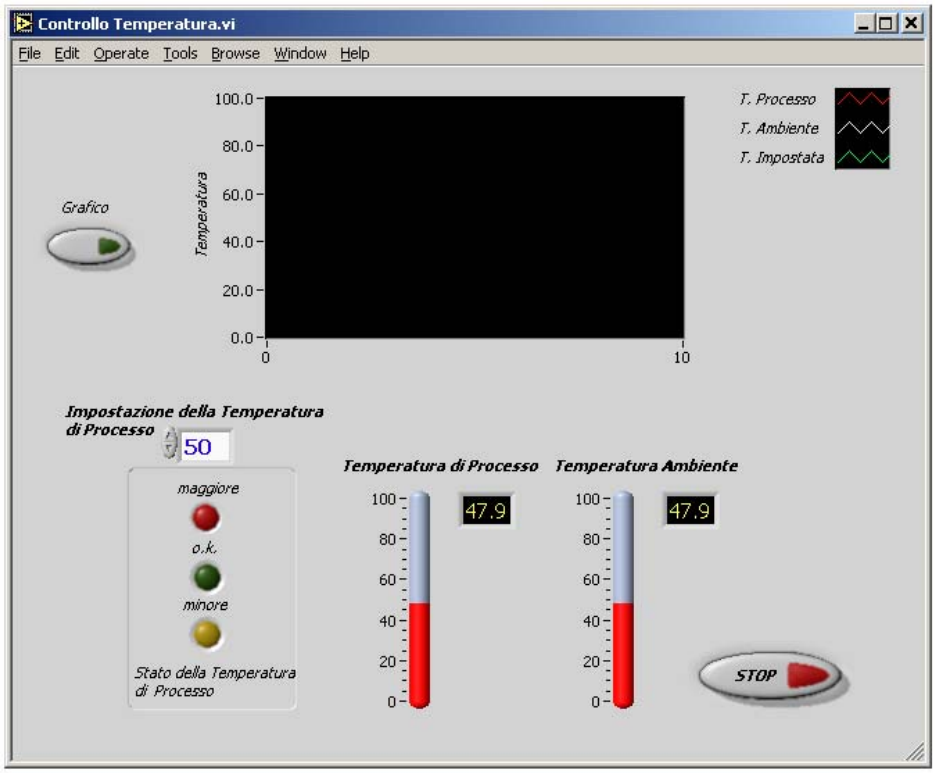
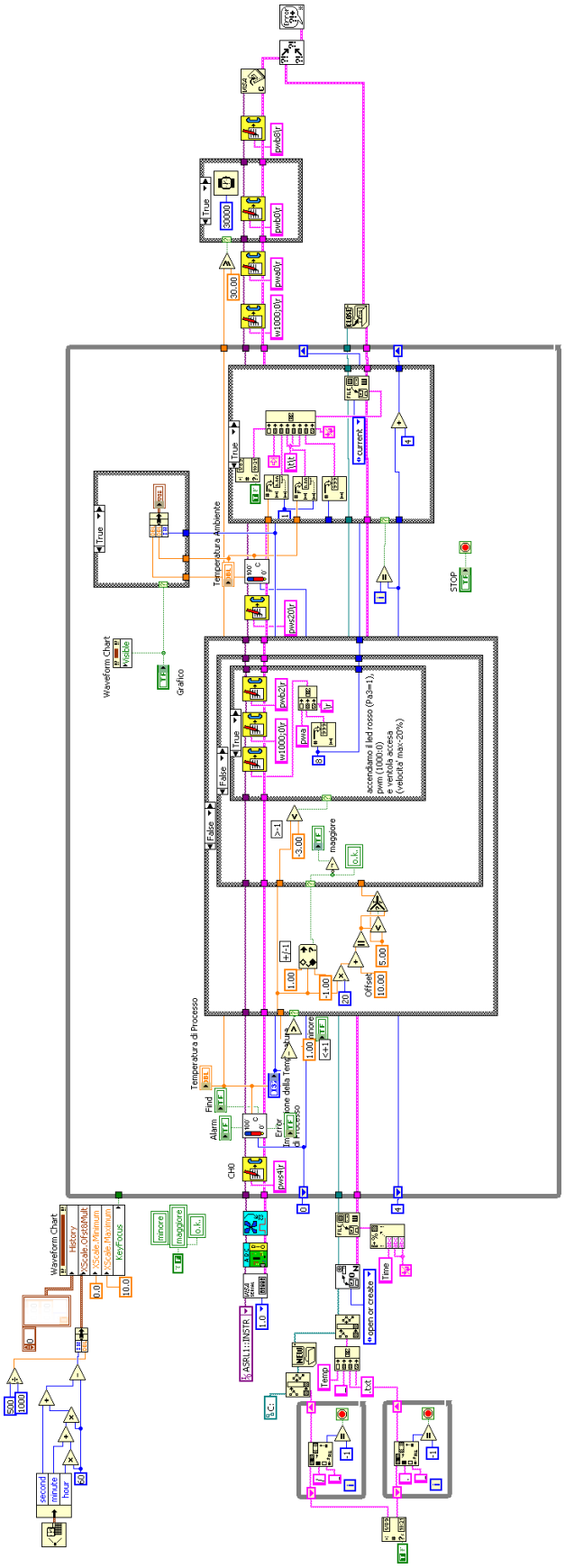


Diagramma a Blocchi



Lettura dati.vi

Pannello Frontale:

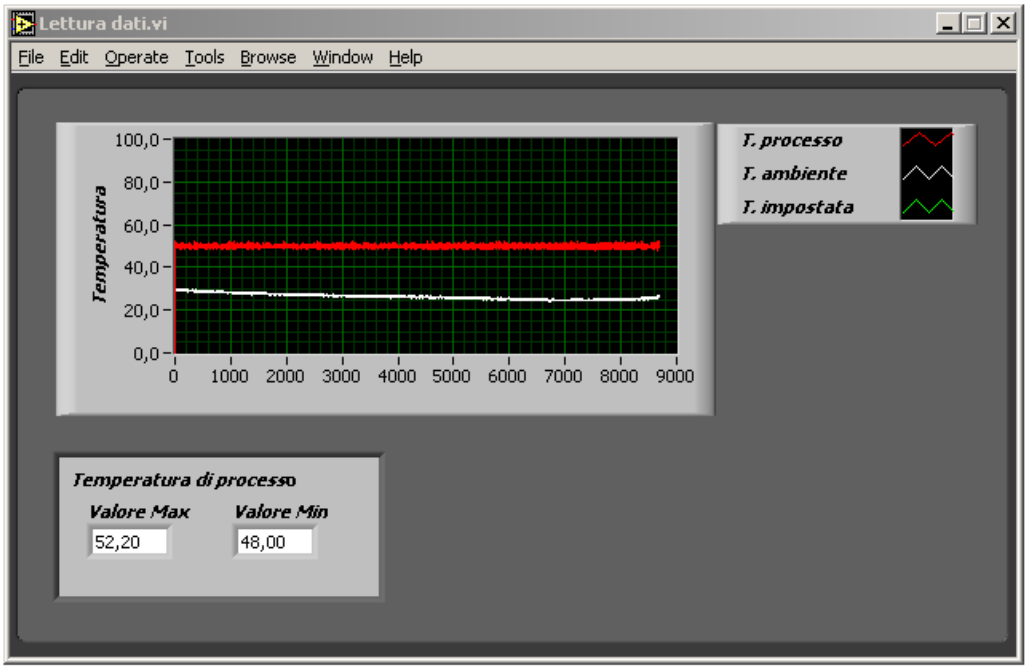
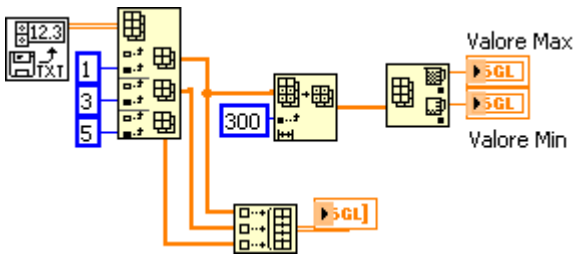
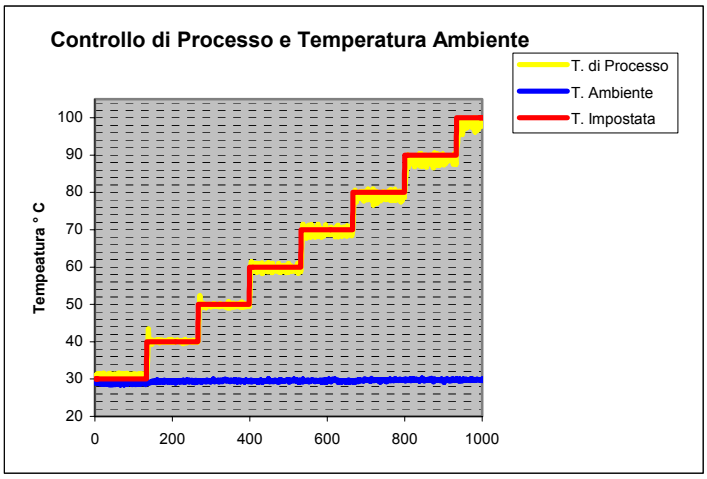


Diagramma a Blocchi:



Che cosa abbiamo capito:

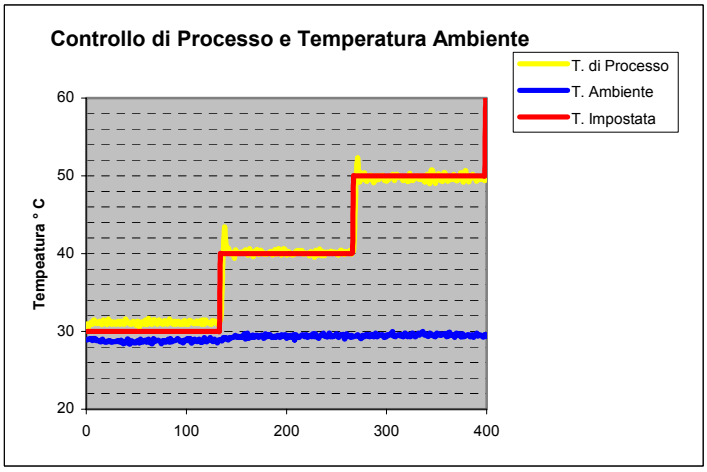
Analisi dati



Quello che notiamo è che il sistema lavora egregiamente per temperature di processo intermedie, diminuendo di precisione verso temperature di processo più alte rispetto alla temperatura ambiente.

Causa di quest'imprecisione è la bassa capacità termica della resistenza di potenza insieme ad una resistività termica verso l'ambiente molto bassa.

Per migliorare avremmo dovuto agire sul sistema retroazionato, oltre che con un'azione proporzionale, anche con azioni integrata e derivata, come viene effettuata da un controllo PID.



Notiamo, anche, una certa imprecisione quando la temperatura di processo è simile a quella ambiente perché la ventola che raffredda la resistenza prende aria nell'ambiente circostante diminuendo l'efficienza di raffreddamento.