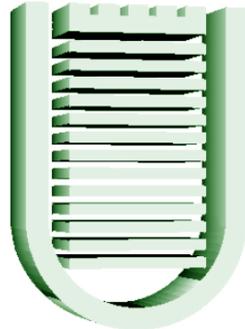


Università degli studi di Roma

Tor Vergata



FACOLTA' DI INGEGNERIA
Dipartimento di Ingegneria Elettronica

Tesi di laurea in Ingegneria Elettronica

**“Progettazione, realizzazione e verifica sperimentale di una
scheda per il controllo della temperatura di un filtro YIG”**

Candidato

Stefano Fasciani

Relatore

Prof. Paolo Colantonio

Correlatore

Walter Ciccognani

A.A. 2002/2003

Indice

Introduzione	pag. 1
Capitolo I – <i>Il sensore di temperatura</i>	pag. 3
Capitolo II – <i>La termoresistenza Pt-100</i>	pag. 8
Capitolo III – <i>Il circuito analogico</i>	pag. 16
Capitolo IV – <i>Il microcontrollore PIC 16F876</i>	pag. 29
Capitolo V – <i>Il display LCD</i>	pag. 39
Capitolo VI – <i>Calcolo della temperatura tramite PIC 16F876</i>	pag. 44
Capitolo VII – <i>Programma per il PIC 16F876</i>	pag. 53
Capitolo VIII – <i>Descrizione del programma</i>	pag. 95
Capitolo IX – <i>Realizzazione e verifica sperimentale</i>	pag. 105
Conclusioni	pag. 110
Bibliografia	pag. 111

Introduzione

Il lavoro presentato nelle pagine seguenti consiste nella progettazione, realizzazione e verifica sperimentale di una scheda per il controllo della temperatura di un filtro YIG.

Il dispositivo in questione è un filtro passa banda (o elimina banda) per segnali compresi tra frequenze di qualche GHz fino a 40-50 GHz. Il principio di funzionamento si basa su un cristallo YIG (Yttrium Iron Garnet), che, se immerso in un campo magnetico costante, si comporta come un risonatore per frequenze nella banda delle microonde.

Fissata la frequenza del filtro può anche accadere che si ottengano drift della stessa o risposte spurie del dispositivo. Ciò è dovuto alle variazioni della temperatura del filtro, che vanno a modificare la frequenza di risonanza del cristallo. Sarà quindi fondamentale controllare continuamente la temperatura del filtro, compresa nel range 40°C-60°C (con percezione anche delle variazioni di un decimo di grado), per assicurarsi il corretto funzionamento del filtro.

Obiettivo del lavoro che verrà presentato è quindi realizzare un “trasduttore elettronico” che trasformi l’informazione temperatura in una grandezza elettrica misurabile; successivamente discretizzarla e convertirla, ad esempio, in un formato binario ASCII, interpretabile da un display LCD, semplicemente percepibile da chi effettuerà la misura. Questo comporta: la scelta di un componente elettronico che abbia un parametro funzione della temperatura; la progettazione di un circuito analogico nel quale inserirlo; la ricerca di una legge che leghi la temperatura alla variabile d’uscita del circuito; la conversione in formato digitale dell’informazione

proveniente da tale componente; l'interfacciamento con un display su cui visualizzare la temperatura misurata. Quindi il circuito da progettare sarà in parte di natura analogica e in parte di natura digitale. Quest'ultima, oltre alla scelta di opportuni componenti, necessita della presenza di un microcontrollore, opportunamente programmato, che sia in grado di eseguire, in modo ciclico e di continuo, quanto sopra descritto.

I – Il sensore di temperatura

1.1 – Generalità

Il dispositivo che necessita del monitoraggio della temperatura è un filtro YIG. Il dispositivo menzionato è composto da due linee di trasmissione disposte lungo direzioni ortogonali e da una sferetta di materiale ferromagnetico posta all'intersezione. Come evidenziato in figura 2.1, il filtro ha 6 connettori: j1 e j2 sono rispettivamente la porta d'ingresso e la porta d'uscita del segnale da filtrare; E1 ed E2 sono i terminali del circuito di sintonia ed infine E3 ed E4 i morsetti dell'alimentazione del riscaldatore. Quest'ultimo elemento è presente in quanto la temperatura (come è stato precedentemente analizzato) è parametro funzionale del filtro YIG.

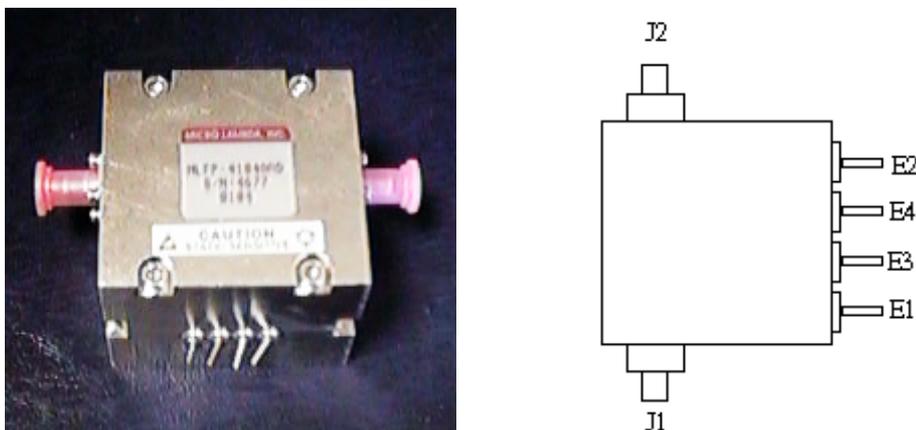


Figura 1.1: filtro YIG (foto e schematizzazione)

Il monitoraggio della temperatura verrà effettuata attraverso un componente in grado di seguirne le variazioni.

È possibile monitorare la temperatura con o senza contatto fisico. La misurazione senza contatto è effettuata tramite uso di pirometri ottici o altri dispositivi in grado di percepire le informazioni elettromagnetiche (nella zona dell'infrarosso) provenienti dall'oggetto in questione. I sistemi che prevedono contatto sfruttano le caratteristiche di componenti come le *termocoppie* e le *termoresistenze*. La via adottata in questo progetto è l'ultima menzionata, quindi la scelta cadrà sul componente che presenta maggiori vantaggi a tal fine.

1.2 – Termocoppie

Le *termocoppie* sono elementi sensibili alla temperatura che sfruttano il contatto tra due metalli diversi. Ponendo a contatto due fili metallici di natura diversa e mantenendo le due giunzioni a diversa temperatura, il sistema dà origine a una forza elettromotrice dell'ordine di alcuni millivolt che provoca, nel caso il circuito sia chiuso, un passaggio di corrente.

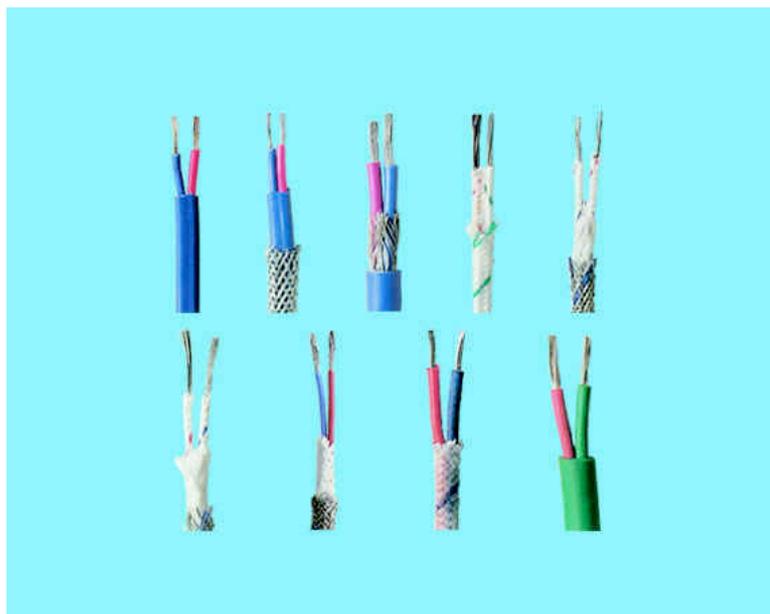


Figura 1.2: termocoppie

La generazione di tensione ai capi di un filo metallico, dovuta ad un gradiente di temperatura ai suoi estremi, è un fenomeno noto come *effetto Seebeck*. Stabilita la natura dei metalli di una termocoppia, il valore della forza elettromotrice è strettamente collegato alla differenza di temperatura esistente fra i due giunti.

Le termocoppie sono vantaggiose in quanto permettono di misurare anche temperature elevate (fino a 1750°C), non necessitano di nessuno stimolo elettrico, in quanto producono esse stesse una corrente; queste presentano però anche diversi svantaggi: sono poco sensibili (da 10µV/°C a 40µV/°C in base al tipo), poco precise (errore di misura da 0,5 a 5,0°C) e la maggior parte di esse ha caratteristiche di uscita non lineari. Inoltre il basso valore di f.e.m. prodotta li rende molto sensibili a disturbi elettromagnetici.

1.2 – Termoresistenze

Le *termoresistenze* sono delle particolari resistenze idonee alla misurazione della temperatura. Il valore di una resistenza è dato dalle relazioni

$$R = \rho \cdot \frac{l}{s} \qquad R_t = R_0 \cdot (1 + \alpha_0 T)$$

dove l e s rappresentano le dimensioni fisiche della resistenza (lunghezza e sezione) e ρ rappresenta la resistività, caratteristica del materiale, R_t è il valore della resistenza alla temperatura T , R_0 il valore alla temperatura di 0°C e α_0 il coefficiente di temperatura. Quindi il valore di R varia al variare della temperatura in quanto nei metalli si alterano le vibrazioni del reticolo cristallino che ostacolano il passaggio di elettroni.

La tabella sottostante riporta i valori di resistività e il coefficiente di temperatura di alcuni materiali conduttori (tabella 1.1).

Valori di resistività dei principali conduttori		
Materiali conduttori	Resistività ($\Omega\text{mm}^2/\text{m}$ a $0\text{ }^\circ\text{C}$) (ρ)	Coefficiente di temperatura (α)
Argento	0,015	0,0038
Rame elettrolitico	0,016	0,0039
Oro	0,023	0,0036
Alluminio	0,0265	0,004
Tungsteno	0,050	0,0042
Bronzo fosforoso	0,07	0,0039
Platino	0,1	0,0036
Ferro dolce	0,13	0,0048
Piombo	0,20	0,0042

Tabella 1.1

Ovviamente la misura della temperatura non viene effettuata con una comune resistenza, bensì con termoresistenze, che possono essere di vari tipi e materiali; la più comune è il modello chiamato Pt-100. La sigla indica il materiale con il quale sono realizzate queste resistenze (platino) e il valore della resistenza a 0°C (100Ω). Generalmente sono di due tipi: a terminazione di filo in platino e a resistenza a film sottile di platino.

La quasi linearità della relazione tra resistenza e temperatura, usando questi componenti, è uno dei principali vantaggi nella misurazione del parametro in questione. Le termoresistenze, inoltre, presentano un'alta precisione (da $0,1$ a $1,0^\circ\text{C}$) e una buona sensibilità ($0,4\ \Omega/^\circ\text{C}$). Tuttavia questi componenti hanno alcuni svantaggi: in primo luogo la necessita di uno stimolo elettrico (una resistenza non attraversata da corrente ha ai suoi

capi una tensione nulla o viceversa); l'auto riscaldamento per effetto joule,

$$W = R \cdot I^2 \quad \text{potenza dissipata da } R \text{ in calore}$$

che può influenzare la misura della temperatura e inoltre non si riescono a misurare temperature più alte di 650°C.



Figura 1.2: termoresistenza al platino

L'analisi precedente dimostra che la scelta di una termoresistenza per il controllo della temperatura del filtro YIG è più idonea in quanto una termocoppia risulterebbe disturbata da campi elettromagnetici presenti, risulterebbe meno precisa e comporterebbe una difficoltà maggiore nella trasduzione della temperatura a seguito del suo comportamento non lineare. Inoltre il sistema deve operare in un intervallo compreso tra i 40°C e i 60°C, quindi non necessita delle alte temperature rilevabili da una termocoppia. D'altro canto ci sarà bisogno della presenza di un generatore costante di tensione o corrente per stimolare elettricamente la termoresistenza.

II – La termoresistenza Pt-100

2.1 – Generalità

Al fine della realizzazione del misuratore in questione, il Pt-100, con le sue caratteristiche tipiche, si è rivelato essere il componente migliore nel raggiungimento dello scopo. Verranno mostrate ora le caratteristiche principali di tale componente.

Il componente in questione è un *Pt-100 Classe A 2x10mm a Film*.

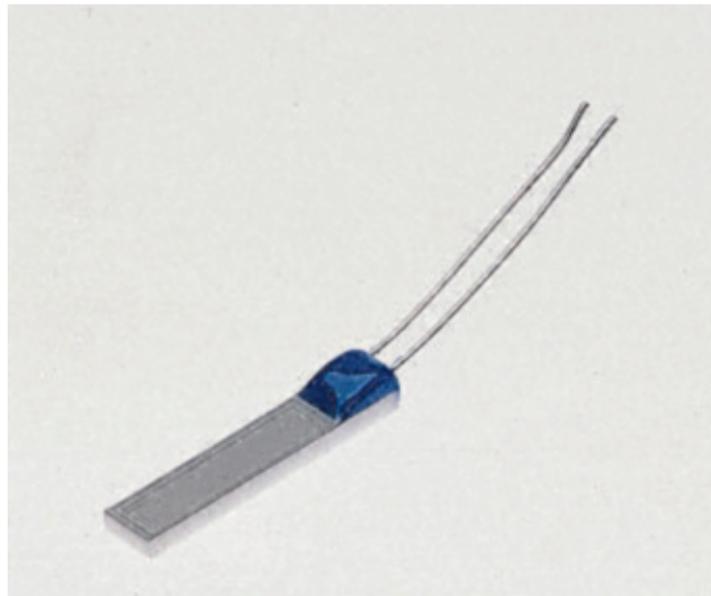


Figura 2.1: Pt-100 Classe A 2x10mm

La parte sensibile consiste in un elemento PT100 (pellicola) con rivestimento in vetro, ed è realizzato mediante deposizione sotto vuoto di platino su un substrato di ceramica che, successivamente, viene definito al laser. Le caratteristiche strutturali sopra riportate fanno sì che la velocità di risposta termica sia molto alta nell'intervallo $[-50^{\circ}\text{C} ; +500^{\circ}\text{C}]$.

Vi è inoltre la possibilità di installare due ulteriori connettori per effettuare una lettura a quattro poli (due per la tensione, due per la corrente) anziché a due. Tuttavia in questo caso adotteremo una lettura a due.



Figura 2.2: comando a 2 o 4 fili

La tabella sottostante riporta l'errore percentuale nella lettura della temperatura in presenza di campo magnetico, e come si può ben osservare, nell'intervallo d'interesse, l'errore è molto basso (tabella 2.1).

Typical Magnetic Field-Dependent Temperature Errors $\Delta T/T$ (%) at B (magnetic induction)					
T(K)	B (tesla)				
	2.5	5	8	14	19
20	20	–	100	250	–
40	0.5	1.5	3	6	8.8
87	0.04	0.14	0.4	1	1.7
300	-0.01	0.001	0.02	0.07	0.13

Tabella 1.1

La seguente tabella riporta i valori della resistenza al variare della temperatura (in gradi Celsius con step di 5°C) per la famiglia di componenti Pt-100 (tabella 2.2).

°C	Ω	°C	Ω	°C	Ω	°C	Ω
-200	18.52	20	107.79	240	190.47	460	267.56
-195	20.68	25	109.73	245	192.29	465	269.25
-190	22.83	30	111.67	250	194.10	470	270.93
-185	24.97	35	113.61	255	195.91	475	272.61
-180	27.10	40	115.54	260	197.71	480	274.29
-175	29.22	45	117.47	265	199.51	485	275.97
-170	31.33	50	119.40	270	201.31	490	277.64
-165	33.44	55	121.32	275	203.11	495	279.31
-160	35.54	60	123.24	280	204.90	500	280.98
-155	37.64	65	125.16	285	206.70	505	282.64
-150	39.72	70	127.08	290	208.48	510	284.30
-145	41.80	75	128.99	295	210.27	515	285.96
-140	43.88	80	130.90	300	212.05	520	287.62
-135	45.94	85	132.80	305	213.83	525	289.27
-130	48.00	90	134.71	310	215.61	530	290.92
-125	50.06	95	136.61	315	217.38	535	292.56
-120	52.11	100	138.51	320	219.15	540	294.21
-115	54.15	105	140.40	325	220.92	545	295.85
-110	56.19	110	142.29	330	222.69	550	297.49
-105	58.23	115	144.18	335	224.45	555	299.12
-100	60.26	120	146.07	340	226.21	560	300.75
-95	62.28	125	147.95	345	227.96	565	302.38
-90	64.30	130	149.83	350	229.72	570	304.01
-85	66.31	135	151.71	355	231.47	575	305.63
-80	68.33	140	153.58	360	233.21	580	307.25
-75	70.33	145	155.46	365	234.96	585	308.87
-70	72.33	150	157.33	370	236.70	590	310.49
-65	74.33	155	159.19	375	238.44	595	312.10
-60	76.33	160	161.05	380	240.18	600	313.71
-55	78.32	165	162.91	385	241.91	605	315.31
-50	80.31	170	164.77	390	243.64	610	316.92
-45	82.29	175	166.63	395	245.37	615	318.52
-40	84.27	180	168.48	400	247.09	620	320.12
-35	86.25	185	170.33	405	248.81	625	321.71
-30	88.22	190	172.17	410	250.53	630	323.30
-25	90.19	195	174.02	415	252.25	635	324.89
-20	92.16	200	175.86	420	253.96	640	326.48
-15	94.12	205	177.69	425	255.67	645	328.06
-10	96.09	210	179.53	430	257.38	650	329.64
-5	98.04	215	181.36	435	259.08	655	331.22
0	100.00	220	183.19	440	260.78	660	332.79
5	101.95	225	185.01	445	262.48		
10	103.90	230	186.84	450	264.18		
15	105.85	235	188.66	455	265.87		

Tabella 2.2

È possibile notare la caratteristica della quasi linearità del Pt-100 (visibile anche nella figura 2.3), ma, per ottenere una misura più precisa, sarà comunque utile avere una legge quadratica che leghi resistenza a temperatura.

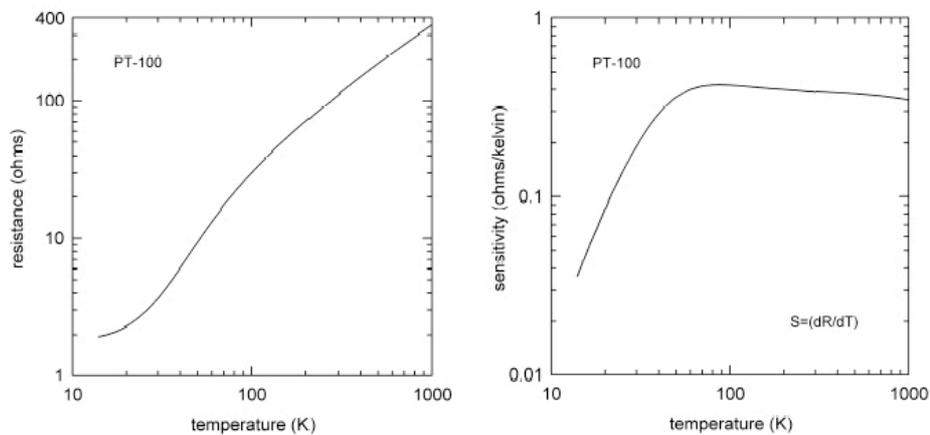


Figura 2.3: curve di risposta del Pt-100

Il sensore utilizzato è di classe A e, come mostrano i grafici sottostanti (tabella 2.3, figura 2.4), lo scostamento dal valore atteso di temperatura, in funzione della resistenza, è minimo specialmente nel range da prendere in considerazione.

Measuring Temp. °C	Permissible Deviations			
	Class A		Class B	
	Ω	°C	Ω	°C
-200	±0.24	±0.55	±0.56	±1.3
-100	±0.14	±0.35	±0.32	±0.8
0	±0.06	±0.15	±0.12	±0.3
100	±0.13	±0.35	±0.30	±0.8
200	±0.20	±0.55	±0.48	±1.3
300	±0.27	±0.75	±0.64	±1.8
400	±0.33	±0.95	±0.79	±2.3
500	±0.38	±1.15	±0.93	±2.8
600	±0.43	±1.35	±1.06	±3.3
650	±0.46	±1.45	±1.13	±3.6
700	—	—	±1.17	±3.8
800	—	—	±1.28	±4.3
850	—	—	±1.34	±4.6

Tabella 2.3

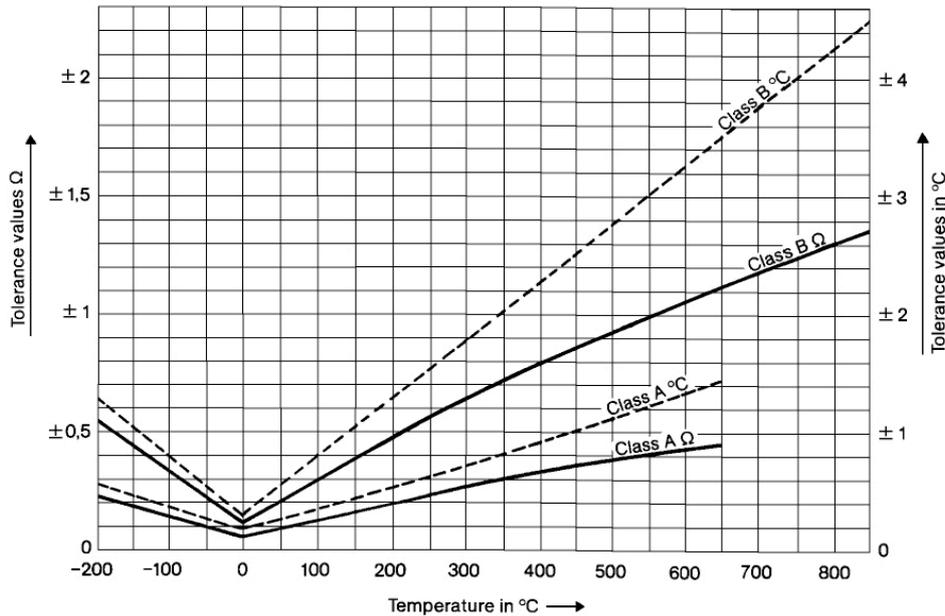


Figura 2.4: tolleranza in °C e in Ω

È necessario ricordare che stiamo considerando resistenze e quindi non bisogna trascurare l'effetto Joule e il rumore Johnson. Il passaggio di corrente elettrica dentro ogni conduttore provoca riscaldamento, conseguente alla produzione di una quantità di calore

$$Q = P \cdot \Delta t \quad P = R \cdot I^2$$

dove Δt è l'intervallo di tempo, il termine $R \cdot I^2$, la potenza dissipata dalla resistenza R. Nel caso del Pt-100 il valore di R è basso (dell'ordine di 10^2) quindi il valore della corrente che scorre al suo interno dovrà essere dell'ordine di 10^{-3} (con una potenza dissipata dell'ordine di 10^{-4}) per evitare un autoriscaldamento che falserebbe del tutto la misura da effettuare. Ad esempio a 0°C , con una corrente di 1mA, il Pt-100 dissipa una potenza di $100\mu\text{W}$. La casa costruttrice dichiara 1V, 3mA e 5mW come valori limite prima della rottura.

Anche il rumore potrebbe influenzare il valore letto, poiché ogni resistenza produce un valore efficace di rumore a valor medio nullo per radice di Hertz pari a

$$V_{noise} / \sqrt{Hz} = \sqrt{4 \cdot K_b \cdot T \cdot R}$$

che nel nostro caso è del tutto trascurabile perché il valore di R è molto basso. Possiamo inoltre trascurare anche la resistenza dei cavi che collegano il Pt-100 al resto del circuito in quanto sono talmente corti che è possibile fare questa approssimazione.

2.2 – Relazione temperatura-resistenza

Per ricavare dalla tabella una legge quadratica che leghi resistenza e temperatura come di seguito

$$T = a \cdot R^2 + b \cdot R + c$$

è possibile sfruttare le leggi per i modelli quadratici dai quali si ricavano i coefficienti a , b e c partendo da n coppie di numeri reali (x_i, y_i)

$$\Sigma_j = \sum_{i=1}^n x_i^j \qquad \Sigma'_j = \sum_{i=1}^n x_i^j \cdot y_i \qquad \text{con } j \in \mathbb{N}$$

$$\begin{cases} \Sigma_0 \cdot c + \Sigma_1 \cdot b + \Sigma_2 \cdot a = \Sigma'_0 \\ \Sigma_1 \cdot c + \Sigma_2 \cdot b + \Sigma_3 \cdot a = \Sigma'_1 \\ \Sigma_2 \cdot c + \Sigma_3 \cdot b + \Sigma_4 \cdot a = \Sigma'_2 \end{cases}$$

Il sistema di equazioni può essere scritto anche in forma matriciale

$$\begin{bmatrix} \Sigma_0 & \Sigma_1 & \Sigma_2 \\ \Sigma_1 & \Sigma_2 & \Sigma_3 \\ \Sigma_2 & \Sigma_3 & \Sigma_4 \end{bmatrix} \cdot \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} \Sigma'_0 \\ \Sigma'_1 \\ \Sigma'_2 \end{bmatrix}$$

e ricavando la matrice inversa, è semplice ottenere la terna di valori a, b, c .

$$\begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} \Sigma_0 & \Sigma_1 & \Sigma_2 \\ \Sigma_1 & \Sigma_2 & \Sigma_3 \\ \Sigma_2 & \Sigma_3 & \Sigma_4 \end{bmatrix}^{-1} \cdot \begin{bmatrix} \Sigma'_0 \\ \Sigma'_1 \\ \Sigma'_2 \end{bmatrix}$$

Nel caso considerato, le coppie di numeri da utilizzare sono

$(40^\circ\text{C}, 115.54\Omega)$;

$(45^\circ\text{C}, 117.47\Omega)$;

$(50^\circ\text{C}, 119.40\Omega)$;

$(55^\circ\text{C}, 121.32\Omega)$;

$(60^\circ\text{C}, 123.24\Omega)$.

e si ottengono

$$\begin{bmatrix} \Sigma_0 & \Sigma_1 & \Sigma_2 \\ \Sigma_1 & \Sigma_2 & \Sigma_3 \\ \Sigma_2 & \Sigma_3 & \Sigma_4 \end{bmatrix} = 1.0\text{e}+9 \cdot \begin{bmatrix} 0.00000000500000 & 0.00000059697000 & 0.00007131169250 \\ 0.00000059697000 & 0.00007131169250 & 0.00852303648538 \\ 0.00007131169250 & 0.00852303648538 & 1.01918447099435 \end{bmatrix}$$

$$\begin{bmatrix} \Sigma_0 & \Sigma_1 & \Sigma_2 \\ \Sigma_1 & \Sigma_2 & \Sigma_3 \\ \Sigma_2 & \Sigma_3 & \Sigma_4 \end{bmatrix}^{-1} = 1.0\text{e}+6 \cdot \begin{bmatrix} 1.05614297102729 & -0.01769836792723 & 0.00007410679317 \\ -0.01769836792723 & 0.00029660836597 & -0.00000124207481 \\ 0.00007410679317 & -0.00000124207481 & 0.00000000520178 \end{bmatrix}$$

$$\begin{bmatrix} \Sigma'_0 \\ \Sigma'_1 \\ \Sigma'_2 \end{bmatrix} = 1.0e+6 \cdot \begin{bmatrix} 0.0002500000000 \\ 0.0299447500000 \\ 3.5885673925000 \end{bmatrix}$$

conseguentemente

$$\begin{bmatrix} a \\ b \\ c \end{bmatrix} = 1.0e+2 \cdot \begin{bmatrix} -2.38715502053499 \\ 0.02238750914112 \\ 0.00001502003008 \end{bmatrix}$$

e la relazione che fornirà la temperatura in funzione della resistenza è

$$T = 0.0015020030 \cdot R^2 + 2.2387509141 \cdot R - 238.7155020535$$

con R conseguentemente variabile in [115.54 Ω;123.24 Ω].

III – Il circuito analogico

3.1 – Generalità

La metodologia più semplice per conoscere il valore di una resistenza consiste nel far scorrere al suo interno una corrente nota e costante, misurare la tensione ai suoi capi e sfruttare la legge di Ohm

$$R = \frac{V}{I_{\text{cost}}}$$

Altri metodi, come il ponte a resistenze, oltre ad essere di più difficile realizzazione, pur portando allo stesso risultato, non conservano le caratteristiche di linearità della legge di Ohm, condizione che rende molto più complicata la determinazione della relativa temperatura.

Quindi la base di partenza per progettare il circuito, nel quale deve essere inserito il Pt-100, è la presenza di una corrente costante di 1mA che deve scorrere in quest'ultimo. Come noto, tale valore è limitato in primo luogo dall'autoriscaldamento per effetto joule ed anche dalla condizione di rottura del componente. Questa corrente, produce ai capi della termoresistenza una differenza di potenziale pari a

$$V_{Pt-100} = R_{Pt-100} \cdot I_{\text{cost}}$$

Per poter elaborare in maniera digitale il segnale sarà necessario riportare tale potenziale all'ingresso di un convertitore analogico/digitale. Sostituendo i relativi valori nel range [40°C;60°C] si ottengono

$$V_{Pt-100}^{Min} = R_{Pt-100} \cdot I_{\text{cost}} = 115.54\Omega \cdot 10^{-3} A = 0,11554V$$

$$V_{Pt-100}MAX = R_{Pt-100} \cdot I_{cost} = 123.24\Omega \cdot 10^{-3} A = 0,12324V$$

È subito evidente che la tensione, sia minima che massima, è dell'ordine del centinaio di millivolt con una differenza di circa 8mV tra i due valori. I comuni microcontrollori dotati di ADC converter accettano valori da 0V a 5V, quindi la tensione uscente dal Pt-100 necessita di uno stadio di amplificazione, prima di essere riportata all'ingresso di un convertitore analogico/digitale.

3.2 – Il generatore di corrente costante controllato in tensione

In condizioni normali è possibile realizzare circuiti aventi un ramo attraversato da una corrente costante e nota; la questione si complica se i parametri del ramo (la resistenza in questo caso) variano e tale variazione non deve influenzare la corrente che vi scorre. È strettamente necessario svincolare tale corrente dai valori di resistenza che il Pt-100 può assumere. È presente una tensione all'ingresso del generatore, quindi questo risulta dipendente e controllato in tensione.

Lo schema adottato (figura 3.1) presenta la particolarità di fornire effettivamente corrente al carico quando questo è collegato ad una vera massa e non ad una massa virtuale come accade con altri generatori con simili caratteristiche.

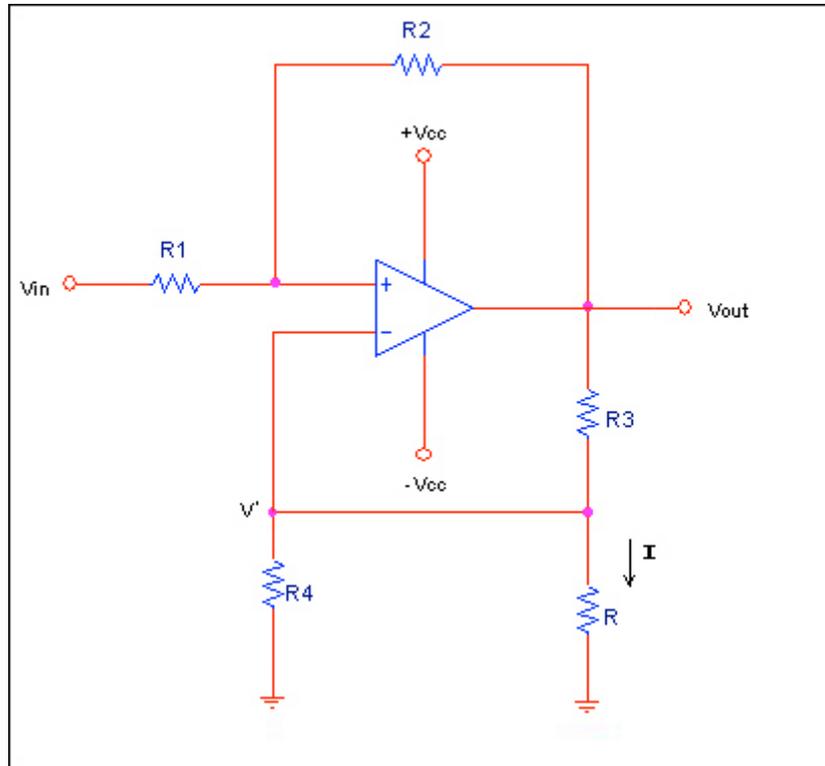


Figura 3.1: schema generatore di corrente

Considerando l'amplificatore operazionale con caratteristiche ideali (quindi con corrente in ingresso nulla e guadagno infinito) si ricava la seguente relazione al nodo del morsetto non invertente

$$\frac{V_{IN} - V'}{R_1} = \frac{V' - V_{OUT}}{R_2}$$

inoltre essendo

$$V' = \frac{V_{OUT}}{\frac{R_4 \cdot R}{R_4 + R} + R_3} \cdot \frac{R_4 \cdot R}{R_4 + R} = I \cdot R$$

è possibile ricavare le seguenti espressioni

$$I = V_{OUT} \cdot \frac{R_4}{R_4 \cdot R + R_3 \cdot R_4 + R_3 \cdot R}$$

$$V_{OUT} = I \cdot \frac{R_4 \cdot R + R_3 \cdot R_4 + R_3 \cdot R}{R_4}$$

$$V_{IN} \cdot \frac{R_2}{R_1} - V' \cdot \frac{R_2}{R_1} - V' = -V_{OUT}$$

$$V_{IN} \cdot \frac{R_2}{R_1} - V' \cdot \left(1 + \frac{R_2}{R_1}\right) = -V_{OUT}$$

sostituendo le espressioni ricavate

$$V_{IN} \cdot \frac{R_2}{R_1} - I \cdot R \cdot \left(1 + \frac{R_2}{R_1}\right) = -I \cdot \frac{R_4 \cdot R + R_3 \cdot R_4 + R_3 \cdot R}{R_4}$$

$$V_{IN} \cdot \frac{R_2}{R_1} = I \cdot \frac{R_1 \cdot R + R_2 \cdot R}{R_1} - I \cdot \frac{R_4 \cdot R + R_3 \cdot R_4 + R_3 \cdot R}{R_4}$$

$$V_{IN} \cdot \frac{R_2}{R_1} = I \cdot \frac{R_2 \cdot R_4 \cdot R - R_1 \cdot R_3 \cdot R - R_1 \cdot R_3 \cdot R_4}{R_1 \cdot R_4}$$

imponendo l'uguaglianza

$$R_2 \cdot R_4 \cdot R - R_1 \cdot R_3 \cdot R = 0 \quad \Rightarrow \quad R_1 \cdot R_3 = R_2 \cdot R_4$$

risulta essere

$$V_{IN} \cdot \frac{R_2}{R_1} = -I \cdot \frac{R_1 \cdot R_3 \cdot R_4}{R_1 \cdot R_4}$$

$$V_{IN} \cdot \frac{R_2}{R_1} = -I \cdot R_3$$

$$I = -V_{IN} \cdot \frac{R_2}{R_1 \cdot R_3} = -V_{IN} \cdot \frac{R_2}{R_2 \cdot R_4}$$

quindi risulta

$$I \cong -\frac{V_{IN}}{R_4}$$

quindi la corrente circolante nel carico R risulta essere indipendente dal carico stesso e dipendente solamente dai valori della tensione d'ingresso e della resistenza in parallelo col carico. Simulando il circuito con un operazionale ideale, la corrente risulta essere quella prevista ed inoltre svincolata dal carico. Ma considerando il caso reale, e, simulando il circuito al calcolatore, si riscontrano problemi dovuti alla non idealità dell'operazionale, ed essendo R in parallelo con R4, parte della I tende a confluire in quest'ultima. Il problema si risolve bilanciando opportunamente i valori di R3 ed R4 per ovviare a tali inconvenienti. La figura 3.2 mostra i risultati della simulazione, dove è possibile osservare il valore della corrente (1mA) che scorre nel Pt-100, schematizzato come una resistenza variabile. Ripetendo la simulazione con il valore massimo che assume il Pt-100, ovvero 123.24Ω (mentre nel caso precedente la simulazione era stata effettuata col valore minimo di 115.54Ω), il valore della corrente che lo attraversa rimane pressoché costante (ovvero 1.001mA).

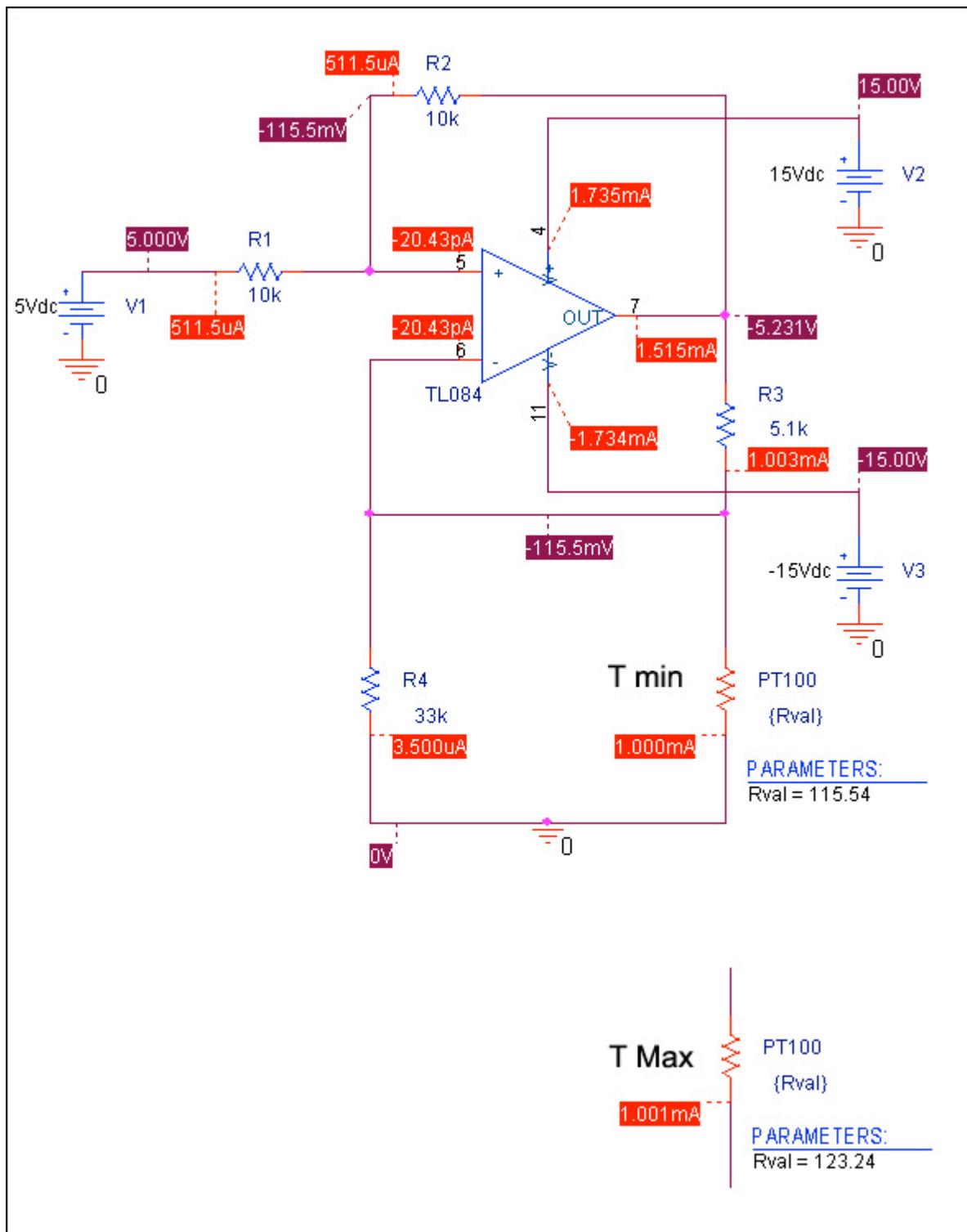


Figura 3.2: generatore di corrente costante

Le tensioni di alimentazione sono uguali a +15V e -15V per quanto riguarda l'amplificatore operazionale; la tensione V_{IN} è stata scelta pari a +5V in quanto risulterà utile per alimentare tutti i dispositivi digitali presenti nel circuito. Essendo tale tensione positiva, e ricordando la relazione ricavata

$$I \cong -\frac{V_{IN}}{R_4}$$

si genera una corrente negativa (di verso opposto) e quindi la tensione rilevata al capo non collegato a massa del Pt-100 ha valore negativo. Ciò non crea particolari problemi in quanto si provvederà a cambiare segno alla tensione nel successivo stadio di amplificazione.

3.3 – L'amplificatore di tensione

Come precedentemente affermato, si necessita di uno stadio amplificatore, tra l'uscita del Pt-100 e l'ingresso dell'ADC converter, per aumentare l'intervallo tra tensione minima e massima, mantenendo comunque tali valori centrali rispetto ai valori limite del convertitore. Inoltre tale stadio deve invertire il segno della tensione. Lo schema circuitale che soddisfa queste caratteristiche è mostrato in figura 3.3, dove un amplificatore operazionale in configurazione invertente presenta il morsetto non invertente collegato a massa. Esistendo un cortocircuito virtuale tra i due morsetti è possibile affermare che

$$I_{R1} = \frac{V_{IN}}{R_1} \quad \text{e} \quad I_{Rr} = \frac{V_{OUT}}{R_r}$$

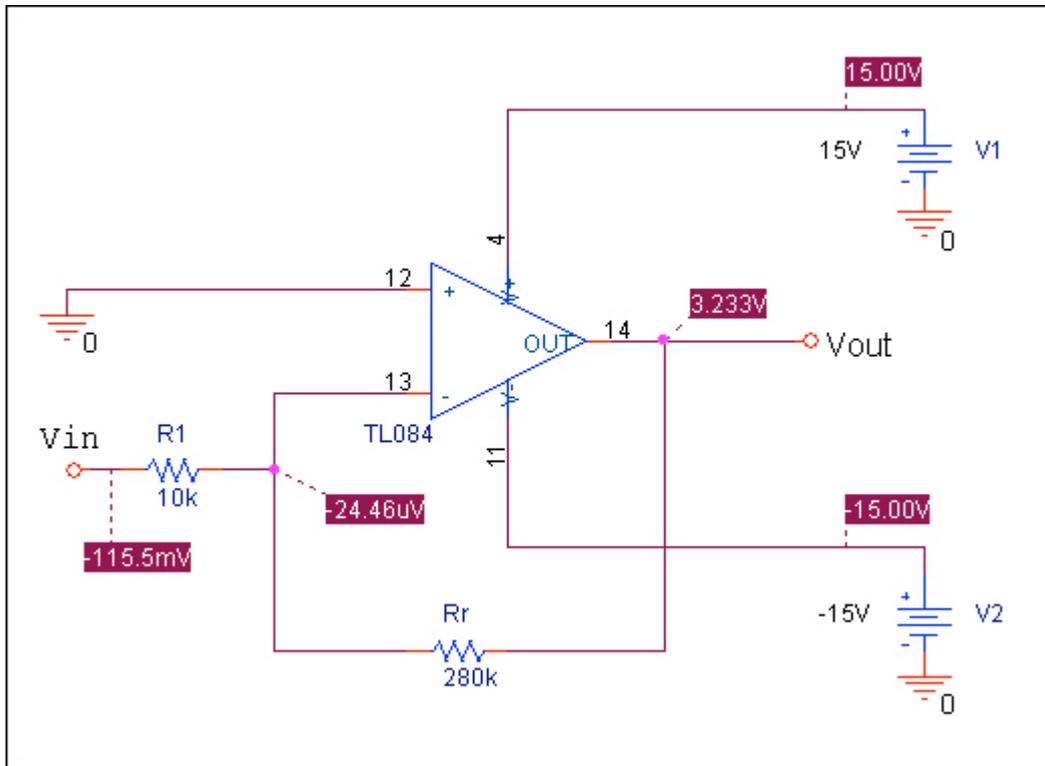


Figura 3.3: amplificatore invertente

ed essendo inoltre

$$I_{R1} = -I_{Rr}$$

il guadagno G dell'amplificatore è dato da

$$G = \frac{V_{OUT}}{V_{IN}} = -\frac{R_r}{R_1}$$

Dai calcoli effettuati, nel caso considerato il valore scelto per G è pari a -28; essendo infatti $R_1 = 10\text{k}\Omega$ ed $R_r = 280\text{k}\Omega$ (valore non esistente in commercio, quindi realizzabile con la serie di due resistenze da $270\text{k}\Omega$ e $10\text{k}\Omega$), il valore di G è verificato sia analiticamente che sperimentalmente tramite simulatore.

$$G = \frac{3.233V}{-0.1155V} \cong -\frac{280k\Omega}{10k\Omega} \cong -28$$

La relazione che è stata ottenuta conserva ancora le caratteristiche di linearità; un operazionale inizia a perdere queste caratteristiche quando si lavora con tensioni che si avvicinano a quelle di alimentazione (+15V e -15V in questo caso).

3.4 – L'inseguitore di tensione

I due subcircuiti fin ora descritti sono stati analizzati singolarmente, ma dovranno lavorare insieme, e dunque, il segnale di uscita del primo (tensione del Pt-100), sarà il segnale di ingresso per l'altro. Simulando il sistema completo, si nota subito una diminuzione della corrente sul Pt-100, in quanto, l'amplificatore di tensione, per quanto piccola, ha una corrente in ingresso che viene prelevata dal pin di V_{IN} che è connesso direttamente alla termoresistenza. Questo modifica la condizione iniziale dalla quale è partito il progetto del circuito, ovvero la necessità di una corrente costante di 1mA nel Pt-100. Per risolvere il problema, occorre minimizzare questo fenomeno al punto tale da renderlo più che trascurabile. Ciò è possibile ponendo tra out e in dei due subcircuiti un inseguitore di tensione (voltage follower), schematizzato in figura 3.4. Passando per il circuito equivalente dell'operazionale si ottiene un guadagno unitario, una resistenza d'ingresso molto elevata e una resistenza d'uscita praticamente nulla.

$$G = 1 \quad R_{IN} = \infty \quad R_{OUT} = 0$$

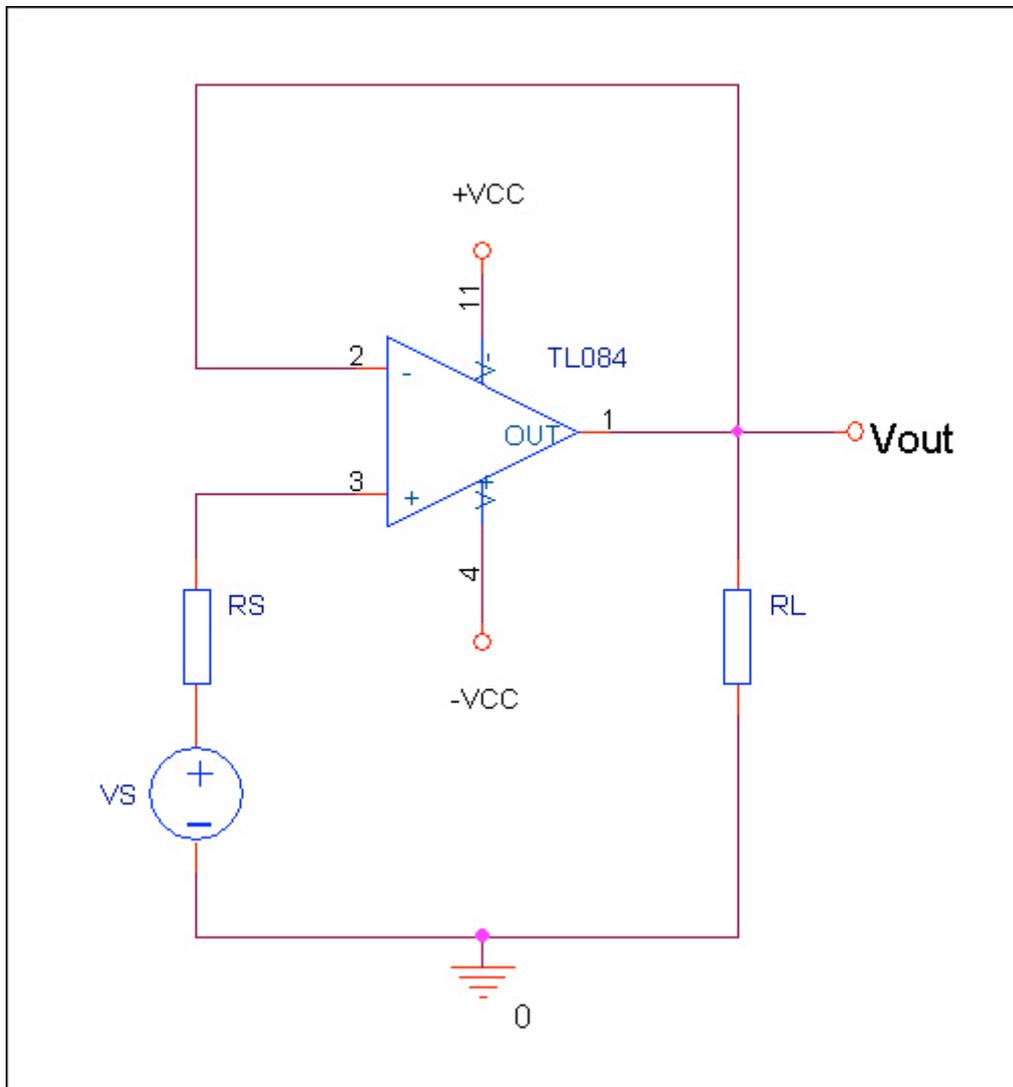


Figura 3.4: inseguitore di tensione

Questa configurazione fa sì che il generatore V_S sia totalmente disaccoppiato dal carico R_L , quindi, il segnale d'ingresso, viene trasferito in uscita inalterato, bloccando anche la corrente che veniva assorbita dal subcircuito che genera la corrente per il Pt-100.

Un inseguitore di tensione, per analoghe ragioni, verrà posto anche tra l'uscita del trasformatore che genera i 5V e il morsetto di V_{IN} del generatore di corrente costante.

3.5 – Simulazione completa del circuito

La tensione d'uscita, che d'altra parte rappresenta l'ingresso del convertitore analogico/digitale, è data dalla seguente espressione

$$V_{OUT} = G \cdot V_{Pt-100} = G \cdot I_{cost} \cdot R_{Pt-100}$$

che numericamente corrisponde a

$$V_{OUT} = (-28) \cdot (-1mA) \cdot R_{Pt-100} \quad \text{con } R_{Pt-100} \in [115.54\Omega; 123.24\Omega]$$

$$V_{OUT}Min = 3.233V \Leftrightarrow T = 40^{\circ}C$$

$$V_{OUT}Max = 3.453V \Leftrightarrow T = 60^{\circ}C$$

le seguenti simulazioni, riferite al circuito di figura 3.6, mostrano i valori di tensione del Pt-100 prima e dopo l'amplificazione, calcolati a 40°C, 50°C e 60°C.

Nella realizzazione pratica sarà usato un componente integrato, il TL084 (14 pin), il quale contiene al suo interno quattro amplificatori operazionali, per rendere più compatta la scheda.

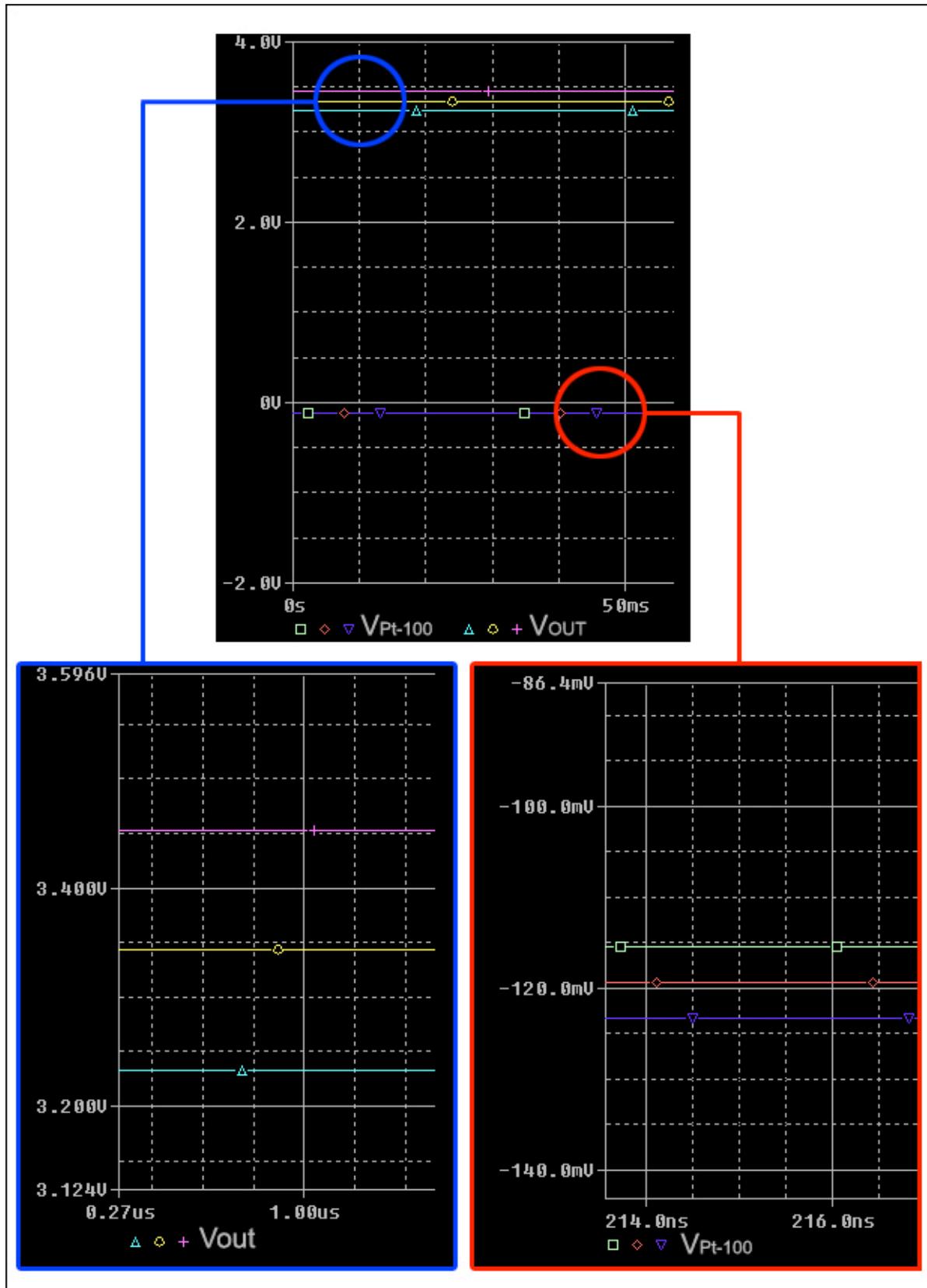


Figura 3.5: simulazione a diverse temperature

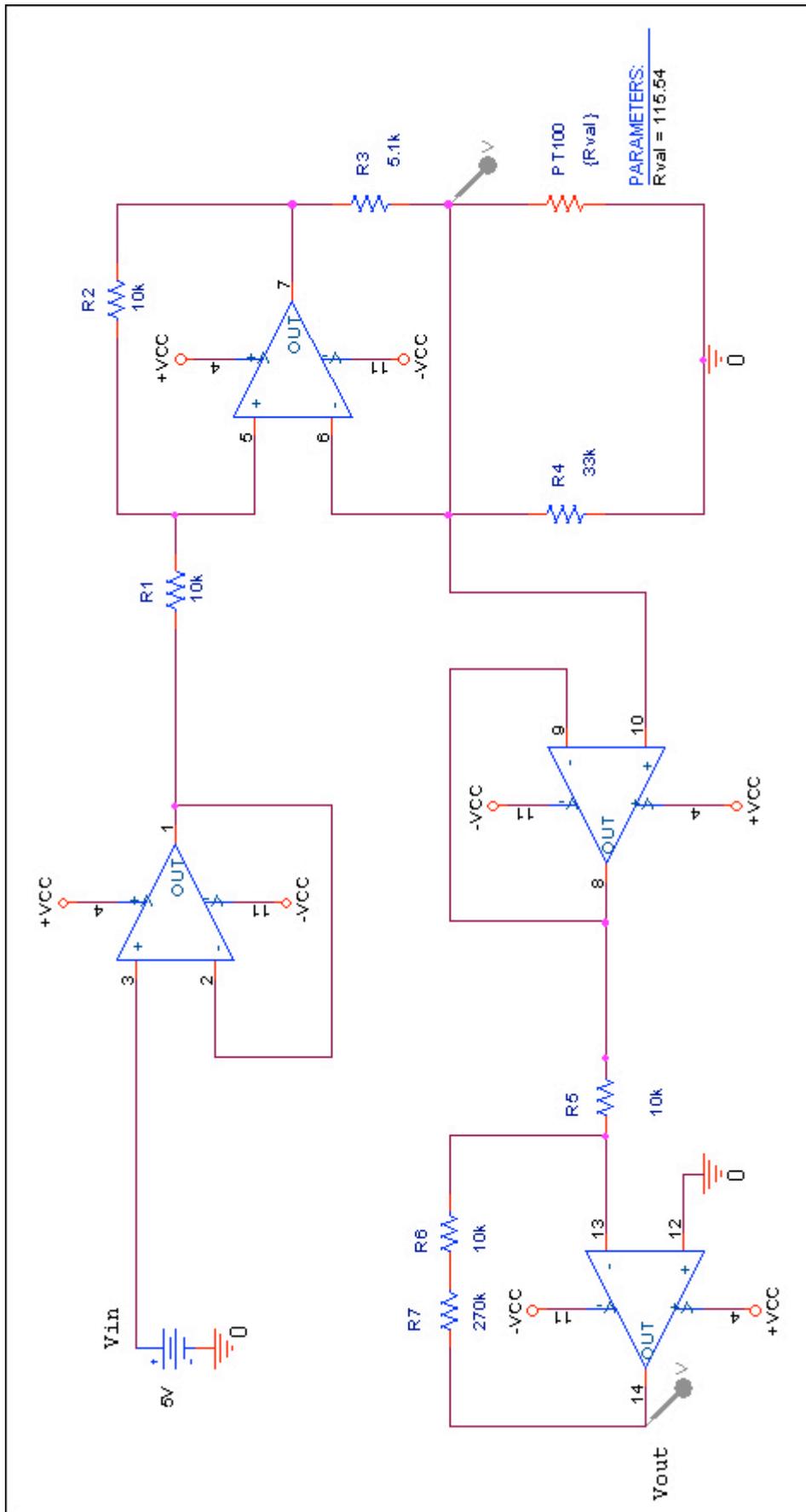


Fig. 3.6

IV – Il microcontrollore PIC16F876

4.1 – Caratteristiche generali

Il microcontrollore adottato per la realizzazione del controllo della temperatura è il *PIC16F876*, prodotto dalla *Microchip Technology Inc.*; questo è dotato delle periferiche necessarie: un ADC converter e una porta digitale per comandare il display LCD. Esso ha una frequenza operativa massima di 20MHz; ha 8Kbytes di memoria programma di tipo FLASH, con opcode di 14 bit; una memoria RAM di 386 bytes; una memoria dati EEPROM di 256 bytes; è dotato di 13 livelli di interrupt; ha tre porte di ingresso/uscita; tre diversi timer; due moduli per la cattura, comparazione o per la modulazione di segnali; è predisposto per comunicazioni seriali di tipo MSSP e USART; ha cinque canali d'ingresso per il modulo analogico/digitale a 10 bit; è programmabile in assembler con un set di trentacinque diverse istruzioni.

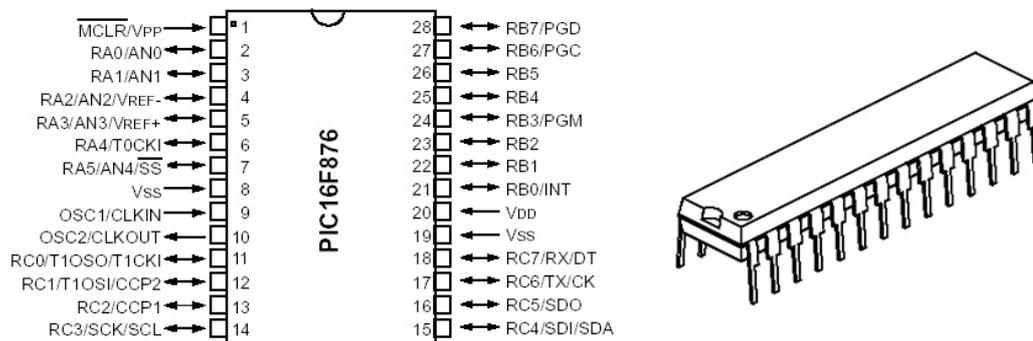


Figura 4.1: PIC 16F876 - pin diagram

La memoria interna è organizzata in quattro diversi banchi, all'interno dei quali ci sono locazioni di memoria riservate ai registri e locazioni di memoria libere; le prime servono per impostare il funzionamento del PIC e delle sue periferiche o come memoria ausiliaria di queste ultime; le seconde possono essere sfruttate dall'utente.

Il registro più importante è lo STATUS register (registro di stato).

STATUS REGISTER (ADDRESS 03h, 83h, 103h, 183h)

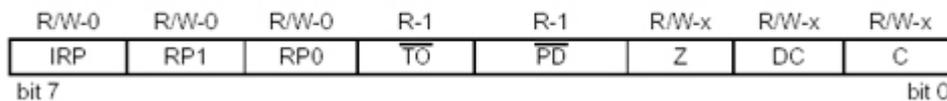


Figura 4.2: status register

È l'unico registro accessibile da qualsiasi banco, e tramite questo è possibile decidere su quale di essi avere accesso. Nel caso di uso di indirizzamento indiretto, il bit IRP seleziona il banco (IRP=0 → banco 0,1; IRP=1 → banco 2,3); le quattro combinazioni dei bit RP1 e RP0 consentono l'accesso a una delle quattro subdivisions della memoria nel caso si usi l'indirizzamento diretto. Gli altri bit sono il time out bit (TO), il power down bit (PD), lo zero bit (Z), il digit carry/borrow bit (DC) e il carry/borrow bit (C).

Altro registro di fondamentale importanza è il CONFIGURATION WORD che a differenza degli altri è più grande (14 bit contro 8 bit); esso viene caricato all'inizio del programma e normalmente non viene più modificato, in quanto riguarda parametri configurazionali del PIC e delle sue periferiche.

CONFIGURATION WORD (ADDRESS 2007h)



Figura 4.3: configuration register

Tra le impostazioni è possibile citare il settaggio della sorgente del segnale di clock, l'abilitazione o disabilitazione della scrittura sulla memoria e la definizione di quest'ultima come protetta o meno, l'abilitazione o disabilitazione dei due timer (power-up e watchdog).

Si possono inoltre menzionare altri registri che svolgono una funzione fondamentale per quanto riguarda l'interfacciamento del PIC con il "mondo esterno": TRISA, TRISB, TRISC, PORTA, PORTB, PORTC. Questi registri fanno riferimento alle tre porte A, B e C del PIC, ognuna delle quali ha fino ad otto pin. Nei registri TRIS- si setta la direzione di ogni piedino della porta (ad ogni bit ne corrisponde uno), cioè si imposta se si tratta di un pin di input o di output (bit 1=input; bit 0=output). I tre registri PORT- contengono il valore logico della tensione su ogni piedino. In condizione di input, mantenendo sempre l'associazione bit-pin, il valore della tensione sul piedino scrive il registro. In condizione di output, il valore dei bit del registro PORT- sarà riportato sul relativo pin in uscita, con livello logico di tensione basso o alto.

4.2 – Il convertitore analogico/digitale

La porta A del PIC16F876, costituita da cinque piedini, oltre a funzionare da ingresso/uscita digitale, può anche funzionare da ingresso per il convertitore secondo lo schema di figura 4.4. L'ingresso analogico carica una circuito di sample and hold, la cui uscita è corrispondentemente anche l'ingresso del convertitore, che partendo dal livello analogico, genera un risultato procedendo con una successione di operazioni di approssimazione. Il convertitore genera un valore digitale a 10 bit, e, di conseguenza, esso è in grado di riconoscere 1024 diversi livelli di tensione nel range $V_{DD} - V_{SS}$ (tensioni di alimentazione, solitamente 5V e 0V). E' possibile configurare i livelli di tensione di riferimento minimo e massimo sugli ingressi RA2 e

RA3. Il modulo A/D è dotato di quattro registri: ADCON0, ADCON1, ADRESH e ADRESL.

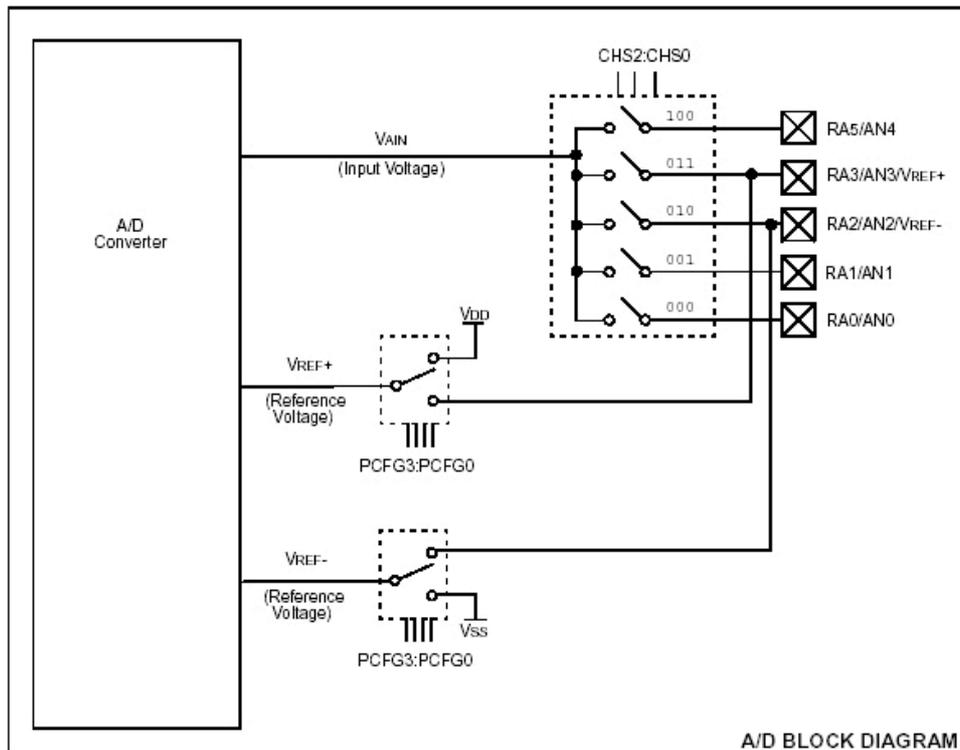


Figura 4.4: ADC block diagram

I registri ADCON0 e ADCON1 sono i registri in cui è possibile impostare tutte le opzioni di conversione, mentre, i registri ADRESH e ADRESL sono quelli in cui sarà memorizzato il risultato.

Il registro ADCON0 (figura 4.5) permette di impostare la frequenza del segnale di clock dell'ADC, tramite i bit ADCS1 e ADCS0; tale frequenza può essere un mezzo, un ottavo o un trentaduesimo della frequenza di lavoro del PIC, oppure può derivare dall'oscillatore RC interno. La scelta di uno di questi sottomultipli del segnale di clock generale è comunque limitata dalla frequenza di quest'ultimo. Le combinazioni dei bit CHS2, CHS1 e CHS0 servono a scegliere da quale dei cinque ingressi selezionare il segnale da

convertire in formato digitale. Il bit GO serve per dare inizio alla conversione, cioè, quando viene settato questo bit a 1 (con il modulo ADC operativo) la conversione parte, e il bit rimane a livello logico alto per tutta la conversione; una volta completata, questo bit torna automaticamente a 0, mentre, se si cancella il bit durante la conversione, quest'ultima viene interrotta. Il bit ADON attiva e disattiva il modulo (1=ON; 0=OFF), permettendo quindi, nelle fasi in cui non si necessita di conversione, di tenerlo in off al fine di consumare complessivamente meno corrente.

ADCON0 REGISTER (ADDRESS: 1Fh)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0
ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/DONE	—	ADON
bit 7						bit 0	

Figura 4.5: ADCON0 register

Il registro ADCON1 (figura 4.6) permette di impostare, tramite il bit ADFM, la giustificazione a destra o sinistra del risultato (1=destra; 0=sinistra). I bit PCFG3, PCFG2, PCFG1 e PCFG0 dello stesso registro,

ADCON1 REGISTER (ADDRESS 9Fh)

U-0	U-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
ADFM	—	—	—	PCFG3	PCFG2	PCFG1	PCFG0
bit 7				bit 0			

PCFG3: PCFG0	AN3 RA3	AN2 RA2	AN1 RA1	AN0 RA0	VREF+	VREF-
0000	A	A	A	A	VDD	VSS
0001	VREF+	A	A	A	RA3	VSS
0010	A	A	A	A	VDD	VSS
0011	VREF+	A	A	A	RA3	VSS
0100	A	D	A	A	VDD	VSS
0101	VREF+	D	A	A	RA3	VSS
011x	D	D	D	D	VDD	VSS
1000	VREF+	VREF-	A	A	RA3	RA2
1001	A	A	A	A	VDD	VSS
1010	VREF+	A	A	A	RA3	VSS
1011	VREF+	VREF-	A	A	RA3	RA2
1100	VREF+	VREF-	A	A	RA3	RA2
1101	VREF+	VREF-	A	A	RA3	RA2
1110	D	D	D	A	VDD	VSS
1111	VREF+	VREF-	D	A	RA3	RA2

A = Analog input D = Digital I/O

Figura 4.6: ADCON1 register

permettono di configurare i pin come ingressi analogici o digitali e di impostare conseguentemente i valori di tensione di riferimento per la conversione, come in tabella di figura 4.6. E' logico impostare ad 1 tutti i bit del registro TRISA, in modo che, tutti i pin della porta A abbiano funzione di input.

I registri ADRESH e ADRESL sono quelli in cui viene memorizzato il risultato. Ogni registro è dotato di 8 bit, quindi l'ADC dispone di un registro a 16 bit per il risultato, pur essendo in grado di calcolarne uno di solamente 10 bit. Questo eccesso di spazio rende flessibile la lettura del risultato, in quanto, con il bit ADFM del registro ADCON1, si sceglie come posizionare il risultato all'interno del registro 16 bit, ossia, per concludere, si imposta se giustificare a destra o a sinistra.

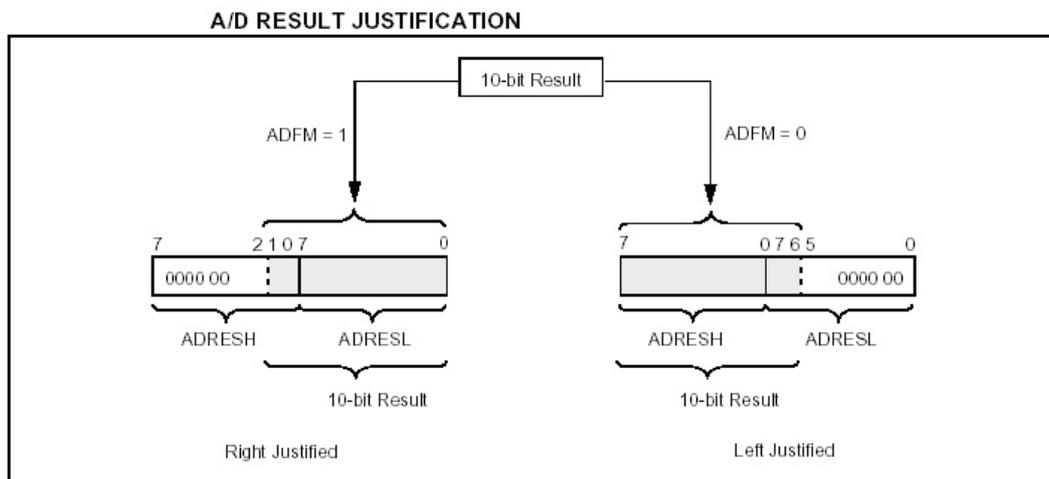


Figura 4.7: ADC result

Queste due possibilità fanno sì che uno dei due registri contenga una volta le otto cifre binarie più significative, un'altra volta le otto meno significative; è quindi possibile andare a fare delle operazioni con il risultato dell'ADC scegliendo una sola tra le due parti, a seconda delle esigenze.

Con il modulo ADC attivo, ogni qualvolta si fa partire la conversione (settando a 1 il bit GO del registro ADCON0), trascorre un certo intervallo di tempo prima che la conversione sia completa (bit GO a 0). Questo è determinato in primo luogo dal circuito analogico, che necessita di circa 100ns per staccare la capacità di mantenimento dall'ingresso analogico e, di altri 19.72µs (circa) che servono per assicurarsi che la capacità sia carica. Da questo momento la conversione può partire. Dopo un tempo pari a T_{AD} parte il calcolo dal bit 9, al bit 0. Ogni bit richiede un tempo di calcolo pari a T_{AD} , e una volta calcolato l'ultimo di essi, c'è bisogno di ulteriore tempo per caricare nel registro il risultato ottenuto, per connettere di nuovo la capacità all'ingresso analogico e infine per cancellare il bit GO, il quale, cancellato, non può essere rimesso a 1 prima che sia trascorso un tempo pari a $2T_{AD}$. Il tempo totale così stimato non supera nel peggiore dei casi i 300µs.

4.3 – Set di istruzioni e programmazione del PIC

Il set di istruzioni in assembler del PIC è estremamente ridotto ma in ogni caso potente; questo sfrutta un codice operativo a 14 bit, e la maggior parte di esse è eseguibile in un solo ciclo (due al massimo). Lo schema di figura 4.8 né da un visione generale. La scrittura di un programma in assembler non comprende solo istruzioni operative, ma anche direttive, variabili, tipi di dati, label e altro ancora. L'operatore in questi casi si occupa della generazione di un file.asm, nel quale vi è il listato del programma in assembler ed al più aggiunge anche parti nel file include (file.inc), che è prodotto dalla casa del microcontrollore e contiene gli indirizzi fisici in esadecimale di tutte le label di memoria. Il software chiamato "compilatore assembler" ha successivamente il compito di elaborare insieme il file.asm ed il file.inc generando un file.hex, un file.cod, un file.lst ed un file.err. Il file.hex contiene gli opcode interpretabili direttamente dal PIC codificati in

formato esadecimale, direttamente riconoscibile da qualsiasi software per programmare i microcontrollori. Il file.err contiene gli errori riscontrati dal compilatore nel file.asm, con descrizione dell'errore e numero di riga dell'errore. Il file.lst è un file di testo che contiene il source assembler con la corrispondente traduzione in opcode a fianco. Sarà successivamente compito di un software programmatore tradurre il file.hex in binario ed inviarlo al PIC, montato su di un apposito circuito, comandato tramite la porta seriale del calcolatore, su cui gira il software programmatore.

Mnemonic, Operands	Description	Cycles	14-Bit Opcode				Status Affected	
			MSb		LSb			
BYTE-ORIENTED FILE REGISTER OPERATIONS								
ADDWF	f, d	Add W and f	1	00	0111	dfff	ffff	C,DC,Z
ANDWF	f, d	AND W with f	1	00	0101	dfff	ffff	Z
CLRF	f	Clear f	1	00	0001	1fff	ffff	Z
CLRWF	-	Clear W	1	00	0001	0xxx	xxxx	Z
COMF	f, d	Complement f	1	00	1001	dfff	ffff	Z
DECf	f, d	Decrement f	1	00	0011	dfff	ffff	Z
DECFSZ	f, d	Decrement f, Skip if 0	1(2)	00	1011	dfff	ffff	
INCF	f, d	Increment f	1	00	1010	dfff	ffff	Z
INCFSZ	f, d	Increment f, Skip if 0	1(2)	00	1111	dfff	ffff	
IORWF	f, d	Inclusive OR W with f	1	00	0100	dfff	ffff	Z
MOVF	f, d	Move f	1	00	1000	dfff	ffff	Z
MOVWF	f	Move W to f	1	00	0000	1fff	ffff	
NOF	-	No Operation	1	00	0000	0xx0	0000	
RLF	f, d	Rotate Left f through Carry	1	00	1101	dfff	ffff	C
RRF	f, d	Rotate Right f through Carry	1	00	1100	dfff	ffff	C
SUBWF	f, d	Subtract W from f	1	00	0010	dfff	ffff	C,DC,Z
SWAPF	f, d	Swap nibbles in f	1	00	1110	dfff	ffff	
XORWF	f, d	Exclusive OR W with f	1	00	0110	dfff	ffff	Z
BIT-ORIENTED FILE REGISTER OPERATIONS								
BCF	f, b	Bit Clear f	1	01	00bb	bfff	ffff	
BSF	f, b	Bit Set f	1	01	01bb	bfff	ffff	
BTFSC	f, b	Bit Test f, Skip if Clear	1(2)	01	10bb	bfff	ffff	
BTFSS	f, b	Bit Test f, Skip if Set	1(2)	01	11bb	bfff	ffff	
LITERAL AND CONTROL OPERATIONS								
ADDLW	k	Add literal and W	1	11	111x	kkkk	kkkk	C,DC,Z
ANDLW	k	AND literal with W	1	11	1001	kkkk	kkkk	Z
CALL	k	Call subroutine	2	10	0kkk	kkkk	kkkk	
CLRWDI	-	Clear Watchdog Timer	1	00	0000	0110	0100	TO,PD
GOTO	k	Go to address	2	10	1kkk	kkkk	kkkk	
IORLW	k	Inclusive OR literal with W	1	11	1000	kkkk	kkkk	Z
MOVLW	k	Move literal to W	1	11	00xx	kkkk	kkkk	
RETFIE	-	Return from interrupt	2	00	0000	0000	1001	
RETLW	k	Return with literal in W	2	11	01xx	kkkk	kkkk	
RETURN	-	Return from Subroutine	2	00	0000	0000	1000	
SLEEP	-	Go into standby mode	1	00	0000	0110	0011	TO,PD
SUBLW	k	Subtract W from literal	1	11	110x	kkkk	kkkk	C,DC,Z
XORLW	k	Exclusive OR literal with W	1	11	1010	kkkk	kkkk	Z

Figura 4.8: instruction set

4.4 – Requisiti circuitali del PIC 16F876

La prima cosa di cui necessita il PIC è un'alimentazione di +5V al pin V_{DD} e 0V ai due pin V_{SS} . Altro elemento fondamentale è il segnale di clock, che è stato scelto proveniente dall'esterno anziché sfruttare quello interno, in quanto, quello esterno, è realizzabile anche con un cristallo al quarzo, molto più preciso dell'oscillatore RC interno. Quindi ai due pin del clock sono connesse le due estremità di un cristallo di quarzo che a loro volta saranno riportate a massa tramite due condensatori da 22pF. La frequenza del segnale di clock è selezionata in base al cristallo che si sceglie e, in questo caso, una velocità di 4MHz risulta più che sufficiente. Il pin MLCR, tramite cui si può eseguire il reset esterno, deve essere mantenuto a livello di tensione di alimentazione per far operare il PIC; questo si fa tipicamente ponendo una resistenza da 10K Ω tra il pin e V_{DD} , mentre un condensatore da 100nF connette lo stesso pin a massa. Per quanto riguarda il convertitore analogico/digitale ci sarà il pin AN0 connesso al segnale amplificato proveniente dal Pt-100, mentre ai pin AN2 e AN3 andranno portati i due livelli di tensione di riferimento. Questo verrà realizzato connettendo a V_{DD} ogni pin tramite la serie di due trimmer di valore 1M Ω e 1K Ω per avere a disposizione due regolazioni della tensione, una molto sensibile ma più approssimativa, l'altra meno sensibile ma molto più precisa.

Il tutto così come descritto è schematizzato in figura 4.9.

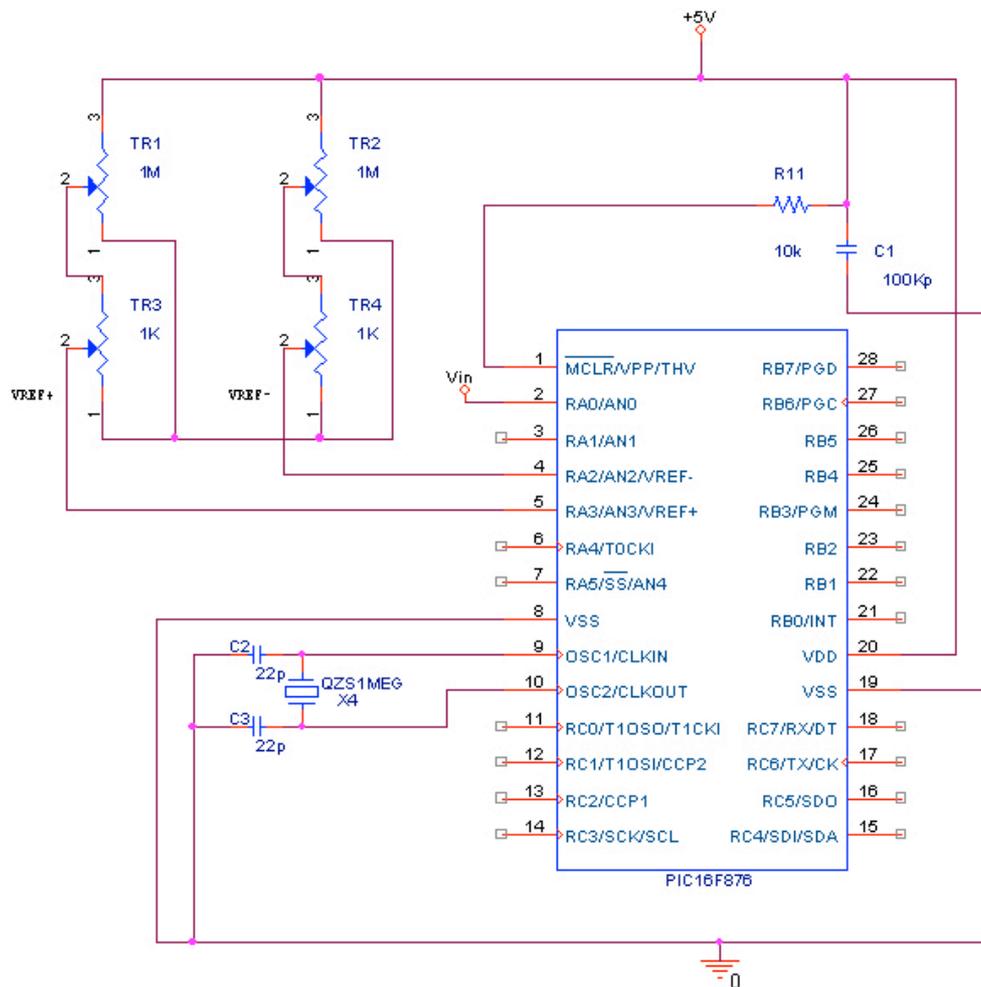


Figura 4.9: circuito per PIC16F876

V – Il display LCD

5.1 – Caratteristiche generali

Il PIC, oltre a gestire l'elaborazione della temperatura, dovrà gestire anche la visualizzazione di essa tramite un apposito dispositivo, che può essere pilotato sempre dallo stesso microcontrollore. Il display a cristalli liquidi scelto per questo tipo di applicazione è il modello *MDK52V-0*, che rispetta le caratteristiche standard della maggior parte dei display. È una struttura organizzata in forma matriciale composta da 2 righe e da 24 colonne per un totale di 48 caratteri. Nel display vi sono anche un microprocessore CMOS e un driver per lo stesso, e, grazie a questi, è compatibile ASCII. Questo codice, essendo anche interpretabile dal compilatore assembler del PIC, semplifica notevolmente la comunicazione fra i due dispositivi. La comunicazione dati con il display può essere eseguita a 4 bit o a 8 bit, ed è possibile visualizzare su di esso fino a 192 diversi caratteri.

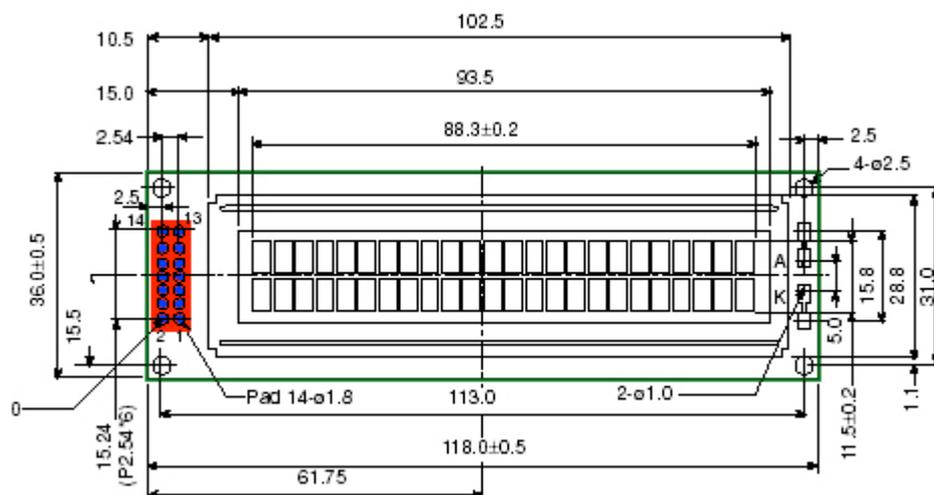


Figura 5.1: schematizzazione display LCD

Il display è dotato di 14 pin di cui ora ne verranno analizzate le caratteristiche peculiari. I primi tre pin sono ingressi per tre tensioni diverse: sul primo e sul secondo entrano le tensioni continue V_{SS} e V_{DD} , le quali corrispondono tipicamente a 0V e +5V (cioè le stesse del PIC); sul terzo invece entra una tensione continua V_0 che comanda il contrasto del display. È possibile realizzare un comando per il contrasto con un semplice trimmer da $1K\Omega$, avente i piedini esterni a contatto con V_{DD} e V_{SS} , mentre, quello centrale, verrà connesso direttamente a V_0 , così da essere in grado di fornire una tensione variabile tramite il trimmer. Il quarto, chiamato RS, è un pin esclusivamente di input. Il nome sta per register select, in quanto nel display sono presenti due registri: uno dati che può essere letto e scritto (ed è quello che riceve i caratteri ASCII da visualizzare), ed uno istruzioni nel quale si settano tutte le opzioni del display. Con il livello di RS alto è possibile leggere o scrivere nel registro dati; con RS basso è possibile scrivere nel registro istruzioni e leggere il busy flag e l'address counter. Il pin successivo è ovviamente R/W (read/write). È un pin esclusivamente di input e serve per selezionare la scrittura o la lettura dei registri. Con R/W basso si è in modalità di scrittura mentre con R/W alto si è in modalità di lettura. Il sesto pin è E (enable), che abilita o disabilita la comunicazione con il display nei due sensi, e in qualsiasi dei registri. Anche esso è un pin di input e, in particolare, è quello che scandisce il tempo delle operazioni che verranno eseguite dal display. I restanti ovvero DB0, DB1, DB2, DB3, DB4, DB5, DB6, DB7 sono pin bidirezionali tramite i quali si trasmettono o si leggono dati o comandi. Nella configurazione di comunicazione a 4 bit i pin DB0, DB1, DB2 e DB3 rimangono inutilizzati.

Il PIC che deve controllare il display è dotato di 8 pin per porta, quindi, volendo controllare il display a 8 bit, si necessiterebbe dell'uso di più porte e ciò potrebbe creare complicazioni a livello software e a livello hardware (necessità di più piste). Conviene quindi realizzare una connessione a 4 bit

dove il PIC dovrà gestire i quattro canali di trasmissione dati e i pin RS ed E. Gli altri pin del display, R/W compreso, possono anche essere direttamente connessi a massa, non essendoci mai necessita di leggere dati, e, per il comando di scrittura, il livello logico di R/W deve essere basso.

5.2 – Sequenza di inizializzazione

I data sheet del display dichiarano la necessita di un inizializzazione del tipo di quella riportata in figura 5.2.

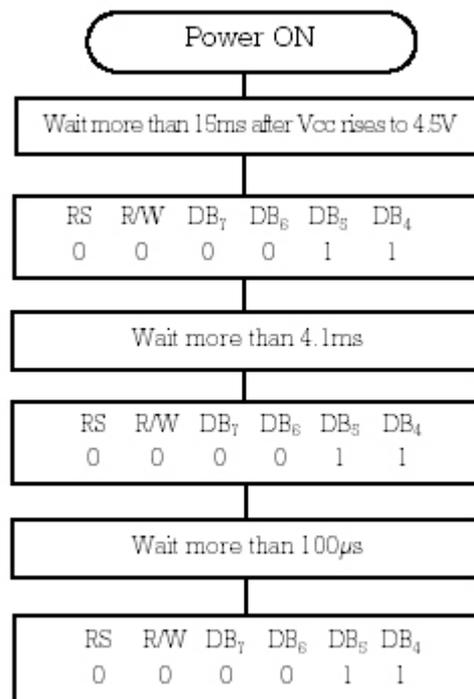


Figura 5.2: sequenza di inizializzazione del display (parte 1)

Per inviare ognuna di queste stringhe di bit è necessario abilitare la ricezione del display portando a livello logico alto il bit E. Come è possibile notare si sta andando a scrivere nel registro istruzioni essendo i bit R/W e RS a livello logico basso.

Dopo aver inviato questi primi tre comandi è possibile eseguire i successivi senza dover impostare un ritardo, in quanto, il tempo in cui il display ritorna disponibile a ricevere, è minore del tempo di esecuzione di un'istruzione del PIC. Gli altri comandi inviati completano l'inizializzazione impostandone le opzioni (compresa quella della comunicazione a 4 bit), relative al caso in cui ci troviamo.

RS	R/W	DB ₇	DB ₆	DB ₅	DB ₄
0	0	0	0	1	0
0	0	0	0	1	0
0	0	1	0	0	0
0	0	0	0	0	0
0	0	1	1	0	0
0	0	0	0	0	0
0	0	0	1	1	0

↓
Initialisation ends

Figura 5.3: sequenza di inizializzazione del display (parte 2)

Le tre stringhe di 8 bit contrassegnate in rosso in figura 5.3, vengono comunicate in due fasi differenti essendo la comunicazione effettuata con un bus a 4 bit. Queste contengono diversi bit scelti dall'utente del display per settarlo a suo piacimento. In questo caso, tramite la prima stringa, sono stati settati la comunicazione a 4 bit e il display a due righe; tramite la seconda viene attivato il display e disabilitati cursore e blink. Infine, tramite la terza stringa, viene impostata la modalità di incremento e disabilitato lo scorrimento del testo. Il display è quindi ora pronto per ricevere i dati relativi alla temperatura.

In figura 5.4 è mostrata la schematizzazione del circuito relativo al display con le connessioni alla porta B del PIC.

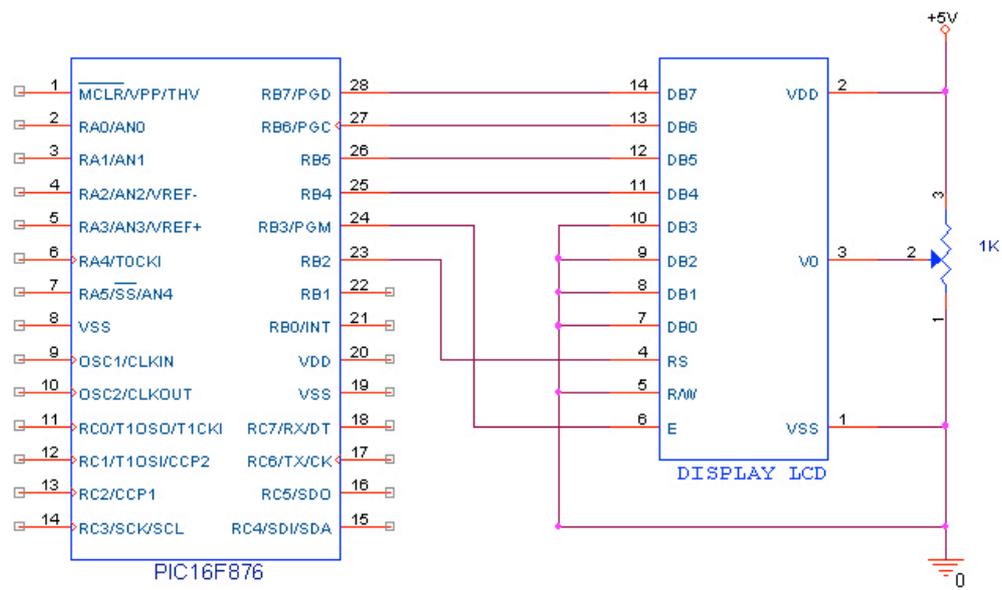


Figura 5.4: circuito per il display LCD

VI – Calcolo della temperatura tramite il PIC16F876

6.1 – Calcolo analitico

Dalla situazione fin ora analizzata emerge che al microcontrollore giunge una tensione continua, che è funzione praticamente lineare della temperatura. Volendo scrivere un'equazione che permette di estrapolare il valore della temperatura vanno considerate le seguenti relazioni

$$T = a \cdot R^2 + b \cdot R + c$$

$$V_{Pt-100} = R_{Pt-100} \cdot I_{\cos t}$$

$$V_{IN} = G \cdot V_{Pt-100}$$

considerando V_{IN} la tensione che entra sul pin dell'ADC, la corrente $I_{\cos t} = -1mA$ e il guadagno $G = -28$, è possibile, operando opportune sostituzioni, giungere fino alla seguente equazione

$$T = (1,9158201530 \cdot V_{IN}^2 + 79,955389789 \cdot V_{IN} - 238.7155020535)^\circ C$$

che è quella che lega la temperatura, ovvero l'informazione che deve elaborare il microcontrollore, con la V_{IN} , cioè l'unica informazione che ha a disposizione il PIC. Quest'ultimo è dotato di un'unità interna di calcolo ALU tramite la quale è possibile effettuare addizioni e sottrazioni. Tuttavia la gestione di numeri maggiori di 255 è problematica in quanto questo è il massimo valore decimale che può contenere una singola locazione di

memoria a 8 bit. È possibile anche implementare tramite codice assembler delle routine che svolgano delle moltiplicazioni, ma sempre rimanendo in una situazione relativamente semplice. In questo caso i prodotti da effettuare risultano assai complicati da implementare tramite routine perché, pur essendo relativamente piccoli, i numeri in questione hanno parecchie cifre dopo la virgola. Un'approssimazione di questi porterebbe sicuramente ad una perdita di precisione nel calcolo della temperatura pur non semplificando la scrittura della relativa routine. Altro nodo da sciogliere è quello legato al valore di V_{IN} , in quanto l'ADC non fornisce come risultato il valore della tensione al suo ingresso, ma un valore binario che indica su quale dei 1024 intervalli tra V_{ref-} e V_{ref+} si trova la tensione che è stata convertita. Ciò richiederebbe un'ulteriore operazione per ricavare V_{IN} oppure un'opportuna tabella interna da cui estrapolarlo. È necessario quindi cercare una strada più semplice e realizzabile che porti alla stessa meta.

6.2 – Soluzione alternativa

Considerando l'intervallo di temperatura [40°C; 60°C] quantizzato tramite degli step di 0.1°C, ci si accorge facilmente che vi sono 201 possibili livelli di temperatura. Tutte queste diverse temperature (40.0°C, 40.1°C, 40.2°C, 40.3°C, ... , 60.0°C) sono esattamente i 201 risultati possibili della nostra misura. L'ADC fornisce un risultato a 10 bit, cioè fornisce 1024 possibili risultati; tarando quindi V_{ref-} e V_{ref+} ai relativi valori minimo e massimo che può assumere l'ingresso del convertitore al variare della temperatura nel range considerato, avremo step di temperatura di 0.019°C. Tutto ciò è inutile in quanto nel caso in cui ci troviamo è sufficiente un livello di precisione più basso. È quindi necessario andare ad agglomerare insieme più livelli di temperatura fino a giungere ad un risultato più consono a questo caso.

È possibile risolvere tutti i problemi nel seguente modo: i 10 bit che fornisce il convertitore vengono scritti in due registri (ADRESH e ADRESL), in quanto ognuno di essi è di 8 bit ed il risultato è di 10 bit; di conseguenza, impostando a 0 il bit ADFM del registro ADCON1, si giustifica il risultato a sinistra, cioè il registro ADRESH, a fine conversione, contiene otto dei 10 bit del risultato, mentre gli altri due sono contenuti in ADRESL, come mostrato in figura 6.1

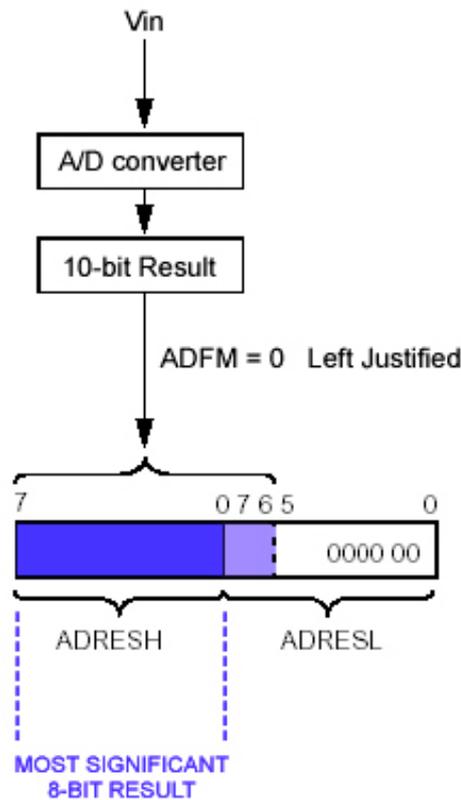


Figura 6.1: diagramma della conversione A/D

Il registro ADRESH contiene i bit più significativi, quindi, trascurando i due bit meno significativi in ADRESL, si ottiene un risultato a 8 bit che corrisponde ad una massimo di 255 in decimale. Ciò corrisponde a prendere un risultato ogni quattro della precedente conversione a 10 bit, e, nel caso

di calcolo della temperatura, vi saranno degli step più grandi, che meglio si adattano alla situazione in questione. Ora si procede facendo in modo che, ognuno dei possibili 256 risultati sia relativo all'incremento di 0.1°C, e ciò sarà possibile da realizzare con un opportuno settaggio delle tensioni di riferimento. V_{ref-} quindi non dovrà più essere riferita alla tensione al caso limite di 40.0°C, ma ad un livello opportunamente calcolato che risulterà essere più basso. Infatti se tra 40.0°C e 60.0°C ci sono 201 possibili livelli di temperatura diversi, scegliendo come riferimento inferiore la tensione che si avrebbe 55 livelli più in basso di 40.0, otterremo che il registro ADRESH incrementa di 1 bit quando la temperatura incrementa di un decimo di grado. Questo porta ad una precisione molto alta ed ad una probabilità di errore molto bassa, in quanto non va dimenticato che anche considerando un risultato solo ad 8 bit, il convertitore ne elabora sempre 10, e quindi, se fosse errato il meno significativo degli 8 bit, dovrebbero essere anche errati gli altri 2 bit non considerati, possibilità in questo caso molto remota.

Il valore che deve quindi essere impostato al pin V_{ref-} è quello relativo ai 34.5°C, che fanno comportare il Pt-100 come un resistore da 113.42Ω, che, a sua volta, attraversato da una corrente di 1mA, genera all'ingresso del convertitore una tensione di 3.1727V, quindi

$$V_{ref-} = 3.1727V$$

mentre il valore del riferimento alla temperatura massimo rimane invariato.

Nella tabella successiva sono stati riportati i possibili valori in binario del risultato a 8 bit dell'ADC, i relativi 256 step di resistenza simulati al calcolatore, i relativi valori di tensione, la temperatura ottenuta dalla relazione precedentemente ricavata e il valore approssimato di temperatura da mostrare sul display.

R PT100	V in	ADRESH	Temperatura	Temp. App.
113,42	3,1727=Vref-	B"00000000"=0	34,52553801	<40
.....
.....	B"00110110"=54	<40	<40
115,54	3,2332	B"00110111"=55	40,00075499	40
115,58	3,2343	B"00111000"	40,10419075	40,1
115,62	3,2354	B"00111001"	40,20763131	40,2
115,66	3,2365	B"00111010"	40,31107667	40,3
115,69	3,2376	B"00111011"	40,38866385	40,4
115,73	3,2387	B"00111100"	40,49211763	40,5
115,77	3,2398	B"00111101"	40,59557622	40,6
115,81	3,2409	B"00111110"	40,69903961	40,7
115,85	3,242	B"00111111"	40,8025078	40,8
115,89	3,2431	B"01000000"	40,90598081	40,9
115,93	3,2442	B"01000001"	41,00945862	41
115,96	3,2453	B"01000010"	41,08707013	41,1
116	3,2464	B"01000011"	41,19055635	41,2
116,04	3,2475	B"01000100"	41,29404738	41,3
116,08	3,2486	B"01000101"	41,39754321	41,4
116,12	3,2497	B"01000110"	41,50104385	41,5
116,16	3,2508	B"01000111"	41,6045493	41,6
116,19	3,2519	B"01001000"	41,68218154	41,7
116,23	3,253	B"01001001"	41,7856954	41,8
116,27	3,2541	B"01001010"	41,88921406	41,9
116,31	3,2552	B"01001011"	41,99273753	42
116,35	3,2563	B"01001100"	42,09626581	42,1
116,39	3,2574	B"01001101"	42,19979889	42,2
116,43	3,2585	B"01001110"	42,30333678	42,3
116,46	3,2596	B"01001111"	42,38099335	42,4
116,5	3,2607	B"01010000"	42,48453966	42,5
116,54	3,2618	B"01010001"	42,58809076	42,6
116,58	3,2629	B"01010010"	42,69164668	42,7
116,62	3,264	B"01010011"	42,7952074	42,8
116,66	3,2651	B"01010100"	42,89877293	42,9
116,7	3,2662	B"01010101"	43,00234326	43
116,73	3,2673	B"01010110"	43,08002416	43,1
116,77	3,2684	B"01010111"	43,18360291	43,2
116,81	3,2695	B"01011000"	43,28718646	43,3
116,85	3,2706	B"01011001"	43,39077482	43,4
116,89	3,2717	B"01011010"	43,49436798	43,5
116,93	3,2728	B"01011011"	43,59796595	43,6
116,96	3,2739	B"01011100"	43,67566758	43,7
117	3,275	B"01011101"	43,77927396	43,8
117,04	3,2761	B"01011110"	43,88288515	43,9
117,08	3,2772	B"01011111"	43,98650115	44
117,12	3,2783	B"01100000"	44,09012195	44,1
117,16	3,2794	B"01100001"	44,19374755	44,2
117,2	3,2805	B"01100010"	44,29737797	44,3
117,23	3,2816	B"01100011"	44,37510393	44,4

R PT100	V in	ADRESH	Temperatura	Temp. App.
117,27	3,2827	B"01100100"	44,47874276	44,5
117,31	3,2838	B"01100101"	44,58238639	44,6
117,35	3,2849	B"01100110"	44,68603482	44,7
117,39	3,286	B"01100111"	44,78968807	44,8
117,43	3,2871	B"01101000"	44,89334612	44,9
117,47	3,2882	B"01101001"	44,99700898	45
117,5	3,2893	B"01101010"	45,07475927	45,1
117,54	3,2904	B"01101011"	45,17843054	45,2
117,58	3,2915	B"01101100"	45,28210661	45,3
117,62	3,2926	B"01101101"	45,3857875	45,4
117,66	3,2937	B"01101110"	45,48947318	45,5
117,7	3,2948	B"01101111"	45,59316368	45,6
117,73	3,2959	B"01110000"	45,6709347	45,7
117,77	3,297	B"01110001"	45,77463361	45,8
117,81	3,2981	B"01110010"	45,87833732	45,9
117,85	3,2992	B"01110011"	45,98204583	46
117,89	3,3003	B"01110100"	46,08575916	46,1
117,93	3,3014	B"01110101"	46,18947729	46,2
117,97	3,3025	B"01110110"	46,29320023	46,3
118	3,3036	B"01110111"	46,37099558	46,4
118,04	3,3047	B"01111000"	46,47472693	46,5
118,08	3,3058	B"01111001"	46,57846308	46,6
118,12	3,3069	B"01111010"	46,68220405	46,7
118,16	3,308	B"01111011"	46,78594981	46,8
118,2	3,3091	B"01111100"	46,88970039	46,9
118,24	3,3102	B"01111101"	46,99345577	47
118,27	3,3113	B"01111110"	47,07127546	47,1
118,31	3,3124	B"01111111"	47,17503925	47,2
118,35	3,3135	B"10000000"	47,27880785	47,3
118,39	3,3146	B"10000001"	47,38258125	47,4
118,43	3,3157	B"10000010"	47,48635946	47,5
118,47	3,3168	B"10000011"	47,59014248	47,6
118,5	3,3179	B"10000100"	47,66798289	47,7
118,54	3,319	B"10000101"	47,77177432	47,8
118,58	3,3201	B"10000110"	47,87557056	47,9
118,62	3,3212	B"10000111"	47,9793716	48
118,66	3,3223	B"10001000"	48,08317745	48,1
118,7	3,3234	B"10001001"	48,1869881	48,2
118,74	3,3245	B"10001010"	48,29080356	48,3
118,77	3,3256	B"10001011"	48,36866831	48,4
118,81	3,3267	B"10001100"	48,47249218	48,5
118,85	3,3278	B"10001101"	48,57632086	48,6
118,89	3,3289	B"10001110"	48,68015434	48,7
118,93	3,33	B"10001111"	48,78399263	48,8
118,97	3,3311	B"10010000"	48,88783573	48,9
119,01	3,3322	B"10010001"	48,99168363	49
119,04	3,3333	B"10010010"	49,06957272	49,1
119,08	3,3344	B"10010011"	49,17342903	49,2
119,12	3,3355	B"10010100"	49,27729015	49,3
119,16	3,3366	B"10010101"	49,38115608	49,4

R PT100	V in	ADRESH	Temperatura	Temp. App.
119,19	3,3388	B"01100100"	49,45905868	49,5
119,23	3,3399	B"01100101"	49,56293302	49,6
119,27	3,341	B"10011000"	49,66681216	49,7
119,31	3,3421	B"10011001"	49,77069611	49,8
119,35	3,3432	B"10011010"	49,87458487	49,9
119,39	3,3443	B"10011011"	49,97847844	50
119,43	3,3454	B"10011100"	50,08237681	50,1
119,47	3,3465	B"10011101"	50,18627999	50,2
119,51	3,3476	B"10011110"	50,29018797	50,3
119,54	3,3487	B"10011111"	50,36812211	50,4
119,58	3,3498	B"10100000"	50,47203851	50,5
119,62	3,3509	B"10100001"	50,57595971	50,6
119,66	3,352	B"10100010"	50,67988571	50,7
119,7	3,3531	B"10100011"	50,78381653	50,8
119,74	3,3542	B"10100100"	50,88775215	50,9
119,78	3,3553	B"10100101"	50,99169258	51
119,81	3,3564	B"10100110"	51,06965105	51,1
119,85	3,3575	B"10100111"	51,17359989	51,2
119,89	3,3586	B"10101000"	51,27755353	51,3
119,93	3,3597	B"10101001"	51,38151198	51,4
119,97	3,3608	B"10101010"	51,48547524	51,5
120,01	3,3619	B"10101011"	51,58944331	51,6
120,04	3,363	B"10101100"	51,66742251	51,7
120,08	3,3641	B"10101101"	51,77139898	51,8
120,12	3,3652	B"10101110"	51,87538026	51,9
120,16	3,3663	B"10101111"	51,97936635	52
120,2	3,3674	B"10110000"	52,08335725	52,1
120,24	3,3685	B"10110001"	52,18735295	52,2
120,28	3,3696	B"10110010"	52,29135345	52,3
120,31	3,3707	B"10110011"	52,36935699	52,4
120,35	3,3718	B"10110100"	52,47336591	52,5
120,39	3,3729	B"10110101"	52,57737963	52,6
120,43	3,374	B"10110110"	52,68139816	52,7
120,47	3,3751	B"10110111"	52,7854215	52,8
120,51	3,3762	B"10111000"	52,88944964	52,9
120,55	3,3773	B"10111001"	52,99348259	53
120,58	3,3784	B"10111010"	53,07151046	53,1
120,62	3,3795	B"10111011"	53,17555182	53,2
120,66	3,3806	B"10111100"	53,27959799	53,3
120,7	3,3817	B"10111101"	53,38364896	53,4
120,74	3,3828	B"10111110"	53,48770474	53,5
120,78	3,3839	B"10111111"	53,59176533	53,6
120,81	3,385	B"11000000"	53,66981393	53,7
120,85	3,3861	B"11000001"	53,77388292	53,8
120,89	3,3872	B"11000010"	53,87795673	53,9
120,93	3,3883	B"11000011"	53,98203534	54
120,97	3,3894	B"11000100"	54,08611876	54,1
121,01	3,3905	B"11000101"	54,19020698	54,2
121,05	3,3916	B"11000110"	54,29430001	54,3
121,08	3,3927	B"11000111"	54,37237294	54,4

R PT100	V in	ADRESH	Temperatura	Temp. App.
121,12	3,3927	B"11001000"	54,47647438	54,5
121,16	3,3938	B"11001001"	54,58058063	54,6
121,2	3,3949	B"11001010"	54,68469168	54,7
121,24	3,396	B"11001011"	54,78880754	54,8
121,28	3,3971	B"11001100"	54,89292821	54,9
121,32	3,3982	B"11001101"	54,99705369	55
121,35	3,3993	B"11001110"	55,07515095	55,1
121,39	3,4004	B"11001111"	55,17928483	55,2
121,43	3,4015	B"11010000"	55,28342352	55,3
121,47	3,4026	B"11010001"	55,38756702	55,4
121,51	3,4037	B"11010010"	55,49171532	55,5
121,55	3,4048	B"11010011"	55,59586843	55,6
121,58	3,4059	B"11010100"	55,67398642	55,7
121,62	3,407	B"11010101"	55,77814794	55,8
121,66	3,4081	B"11010110"	55,88231427	55,9
121,7	3,4092	B"11010111"	55,98648541	56
121,74	3,4103	B"11011000"	56,09066135	56,1
121,78	3,4114	B"11011001"	56,19484209	56,2
121,82	3,4125	B"11011010"	56,29902765	56,3
121,85	3,4136	B"11011011"	56,37716997	56,4
121,89	3,4147	B"11011100"	56,48136393	56,5
121,93	3,4158	B"11011101"	56,5855627	56,6
121,97	3,4169	B"11011110"	56,68976628	56,7
122,01	3,418	B"11011111"	56,79397467	56,8
122,05	3,4191	B"11100000"	56,89818786	56,9
122,09	3,4202	B"11100001"	57,00240585	57
122,12	3,4213	B"11100010"	57,08057251	57,1
122,16	3,4224	B"11100011"	57,18479891	57,2
122,2	3,4235	B"11100100"	57,28903013	57,3
122,24	3,4246	B"11100101"	57,39326615	57,4
122,28	3,4257	B"11100110"	57,49750698	57,5
122,32	3,4268	B"11100111"	57,60175261	57,6
122,35	3,4279	B"11101000"	57,67993999	57,7
122,39	3,429	B"11101001"	57,78419404	57,8
122,43	3,4301	B"11101010"	57,88845289	57,9
122,47	3,4312	B"11101011"	57,99271654	58
122,51	3,4323	B"11101100"	58,09698501	58,1
122,55	3,4334	B"11101101"	58,20125828	58,2
122,59	3,4345	B"11101110"	58,30553636	58,3
122,62	3,4356	B"11101111"	58,38374807	58,4
122,66	3,4367	B"11110000"	58,48803456	58,5
122,7	3,4378	B"11110001"	58,59232585	58,6
122,74	3,4389	B"11110010"	58,69662195	58,7
122,78	3,44	B"11110011"	58,80092286	58,8
122,82	3,4411	B"11110100"	58,90522858	58,9
122,86	3,4422	B"11110101"	59,0095391	59
122,89	3,4433	B"11110110"	59,08777514	59,1
122,93	3,4444	B"11110111"	59,19209407	59,2
122,97	3,4455	B"11111000"	59,29641781	59,3

R PT100	V in	ADRESH	Temperatura	Temp. App.
123,01	3,4466	B"11111001"	59,40074635	59,4
123,05	3,4477	B"11111010"	59,50507971	59,5
123,09	3,4488	B"11111011"	59,60941786	59,6
123,12	3,4499	B"11111100"	59,68767463	59,7
123,16	3,451	B"11111101"	59,7920212	59,8
123,2	3,4521	B"11111110"	59,89637258	59,9
123,24	3,4532=Vref+	B"11111111"=255	60,00072876	60

Sarà ora necessario settare tramite i trimmer i valori delle tensioni di riferimento calcolati, impostare l'ADC in modo che operi tra le tensioni definite e implementare una routine che in base ai valori del registro ADRESH scelga i valori ASCII relativi alla temperatura calcolata. Empiricamente bisogna far sì che il PIC memorizzi al suo interno la terza e la quinta colonna (con le relative associazione di riga) della precedente tabella.

In questo modo non servono operazioni di prodotto, non serve una tabella che riporti dal risultato dell'ADC a quello della tensione d'ingresso, e non vi è bisogno di una routine che trasformi un numero decimale nei relativi caratteri ASCII. La situazione complessiva è stata quindi di gran lunga semplificata, non perdendo nulla in efficienza e precisione.

VII – Programma per il PIC16F876

Le seguenti istruzioni in assembler realizzano il file.asm, da assemblare successivamente, per il calcolo della temperatura.

```

;*****
;
;
; Programma per calcolo temperatura tramite PT100
;
;
; Realizzato da Fasciani Stefano
;
;
;*****
;

LIST                p=16F876
RADIX                DEC
INCLUDE              "p16f876.inc"
ERRORLEVEL          -302
__CONFIG             0x3D71

LCD_RS               equ 2                ;Register Select
LCD_E                equ 3                ;Enable
                    ;LCD data line bus
LCD_DB4              equ 4                ;LCD data line DB4
LCD_DB5              equ 5                ;LCD data line DB5
LCD_DB6              equ 6                ;LCD data line DB6
LCD_DB7              equ 7                ;LCD data line DB7

ORG 0CH

tmpLcdRegister       res 2
msDelayCounter       res 2
TEMPERATURA1         res 1
TEMPERATURA2         res 1
TEMPERATURA3         res 1
SIMBOLO              res 1

org 0x0000            ;reset
goto main

org 0x0004            ;interrupt
goto main

```

```

;*****
;
; Programma principale
;*****
;
main
    bcf     STATUS,RP0           ;accesso consentito al banco 0
    bcf     STATUS,RP1

    movlw   B'01000001'         ;setta il registro ADCON0
    movwf   ADCON0

    clrf    PORTA               ;cancella output delle due porte
    clrf    PORTB

    bsf     STATUS,RP0         ;accesso consentito al banco 1

    movlw   B'11111111'         ;setta tutte le linee della porta A in ingresso
    movwf   TRISA

    movlw   B'00001000'         ;setta il registro ADCON1
    movwf   ADCON1

    movlw   B'00000000'         ;setta tutte le linee della porta B in uscita
    movwf   TRISB

    bcf     PORTB,LCD_DB4       ;cancella le uscite della porta B relativa al display
    bcf     PORTB,LCD_DB5
    bcf     PORTB,LCD_DB6
    bcf     PORTB,LCD_DB7
    bcf     PORTB,LCD_E
    bcf     PORTB,LCD_RS

    bcf     STATUS,RP0         ;accesso consentito al banco 0

    call    LcdInit             ;inizializzazione del display lcd

adcgo
    bsf     ADCON0,GO           ;attiva l'ADC

    call    EsecuzioneRitardo   ;attende che l'ADC abbia eseguito l'acquisizione

    call    TestTemperatura     ;va dal risultato dell'ADC ai valori per display

    call    LcdClear

    movlw   0x06                ; primo carattere sulla sesta colonna
    call    LcdLocate

    movf    'T',W               ;carattere T
    call    LcdSendData

    movf    SIMBOLO,W           ;minore o uguale
    call    LcdSendData

    movf    TEMPERATURA1,W      ;decine

```

```

call    LcdSendData

movf    TEMPERATURA2,W      ;unità
call    LcdSendData

movf    ',',W               ;punto
call    LcdSendData

movf    TEMPERATURA3,W      ;decimi
call    LcdSendData

movf    '°',W               ;grado
call    LcdSendData

movf    'C',W               ;carattere C
call    LcdSendData

movlw   200                  ;attendi 200ms
call    msDelay

goto    adcgo

;*****
; Routine per l'attesa del completamento delle operazioni dell'ADC
;*****
EsecuzioneRitardo
movlw   B'00000001'         ;carica un ritardo di 1ms in W
goto    msDelay             ;esegue il ritardo caricato in W

btfsc   ADCON0,GO          ;controlla se l'ADC ha finito
goto    EsecuzioneRitardo   ;se ADC non ha fatto ripete il ritardo

return

;*****
; Routine per selezione di temperatura e caricamento valori ASCII
;*****
TestTemperatura
;cancella i quattro registri
clrf    TEMPERATURA1
clrf    TEMPERATURA2
clrf    TEMPERATURA3

movlw   '='                 ;carica il simbolo uguale
movwf   SIMBOLO

```

```

temp400
  movlw    '4'                ;carica i valori relativi alle decine e alle unita'
  movwf    TEMPERATURA1      ;decine
  movlw    '0'
  movwf    TEMPERATURA2      ;unita'

  movlw    0x37                ;carica il valore in W
  xorwf    ADRESH,W          ;esegue lo XOR con il risultato dell ADC
                                ;se uguali Z di STATUS va a 1
  btfss    STATUS,Z           ;controlla Z e salta un'operazione se uguali
  goto     temp401            ;salta al test successivo, perche' diversi
  goto     zero                ;uguali quindi carica il valore ASCII e termina

temp401
  movlw    0x38
  xorwf    ADRESH,W

  btfss    STATUS,Z
  goto     temp402
  goto     uno

temp402
  movlw    0x39
  xorwf    ADRESH,W

  btfss    STATUS,Z
  goto     temp403
  goto     due

temp403
  movlw    0x3A
  xorwf    ADRESH,W

  btfss    STATUS,Z
  goto     temp404
  goto     tre

temp404
  movlw    0x3B
  xorwf    ADRESH,W

  btfss    STATUS,Z
  goto     temp405
  goto     quattro

```

```

temp405
  movlw  0x3C
  xorwf  ADRESH,W

  btfss  STATUS,Z

  goto   temp406
  goto   cinque

temp406
  movlw  0x3D
  xorwf  ADRESH,W

  btfss  STATUS,Z
  goto   temp407
  goto   sei

temp407
  movlw  0x3E
  xorwf  ADRESH,W

  btfss  STATUS,Z
  goto   temp408
  goto   sette

temp408
  movlw  0x3F
  xorwf  ADRESH,W

  btfss  STATUS,Z
  goto   temp409
  goto   otto

temp409
  movlw  0x40
  xorwf  ADRESH,W

  btfss  STATUS,Z
  goto   temp410
  goto   nove

temp410
  movlw  '1'
  movwf  TEMPERATURA2           ;cambia il valore della cifra delle unità

  movlw  0x41
  xorwf  ADRESH,W

  btfss  STATUS,Z
  goto   temp411
  goto   zero

```

```
temp411
  movlw 0x42
  xorwf ADRESH,W

  btfss STATUS,Z
  goto temp412
  goto uno
```

```
temp412
  movlw 0x43
  xorwf ADRESH,W

  btfss STATUS,Z
  goto temp413
  goto due
```

```
temp413
  movlw 0x44
  xorwf ADRESH,W

  btfss STATUS,Z
  goto temp414
  goto tre
```

```
temp414
  movlw 0x45
  xorwf ADRESH,W

  btfss STATUS,Z
  goto temp415
  goto quattro
```

```
temp415
  movlw 0x46
  xorwf ADRESH,W

  btfss STATUS,Z
  goto temp416
  goto cinque
```

```
temp416
  movlw 0x47
  xorwf ADRESH,W

  btfss STATUS,Z
  goto temp417
  goto sei
```

```
temp417
  movlw 0x48
  xorwf ADRESH,W

  btfss STATUS,Z
  goto temp418
  goto sette
```

```
temp418
  movlw 0x49
  xorwf ADRESH,W

  btfss STATUS,Z
  goto temp419
  goto otto
```

```
temp419
  movlw 0x4A
  xorwf ADRESH,W

  btfss STATUS,Z
  goto temp420
  goto nove
```

```
temp420
  movlw '2'
  movwf TEMPERATURA2

  movlw 0x4B
  xorwf ADRESH,W

  btfss STATUS,Z
  goto temp421
  goto zero
```

```
temp421
  movlw 0x4C
  xorwf ADRESH,W

  btfss STATUS,Z
  goto temp422
  goto uno
```

```
temp422
  movlw 0x4D
  xorwf ADRESH,W

  btfss STATUS,Z
  goto temp423
  goto due
```

```
temp423
  movlw 0x4E
  xorwf ADRESH,W

  btfss STATUS,Z
  goto temp424
  goto tre
```

```
temp424
  movlw 0x4F
  xorwf ADRESH,W

  btfss STATUS,Z
  goto temp425
  goto quattro
```

```
temp425
  movlw 0x50
  xorwf ADRESH,W

  btfss STATUS,Z
  goto temp426
  goto cinque
```

```
temp426
  movlw 0x51
  xorwf ADRESH,W

  btfss STATUS,Z
  goto temp427
  goto sei
```

```
temp427
  movlw 0x52
  xorwf ADRESH,W

  btfss STATUS,Z
  goto temp428
  goto sette
```

```
temp428
  movlw 0x53
  xorwf ADRESH,W

  btfss STATUS,Z
  goto temp429
  goto otto
```

```

temp429
  movlw  0x54
  xorwff ADRESH,W

  btfss  STATUS,Z
  goto   temp430
  goto   nove

temp430
  movlw  '3'
  movwff TEMPERATURA2

  movlw  0x55
  xorwff ADRESH,W

  btfss  STATUS,Z
  goto   temp431
  goto   zero

temp431
  movlw  0x56
  xorwff ADRESH,W

  btfss  STATUS,Z
  goto   temp432
  goto   uno

temp432
  movlw  0x57
  xorwff ADRESH,W

  btfss  STATUS,Z
  goto   temp433
  goto   due

temp433
  movlw  0x58
  xorwff ADRESH,W

  btfss  STATUS,Z
  goto   temp434
  goto   tre

temp434
  movlw  0x59
  xorwff ADRESH,W

  btfss  STATUS,Z
  goto   temp435
  goto   quattro

```

```

temp435
  movlw 0x5A
  xorwf ADRESH,W

  btfss STATUS,Z
  goto  temp436
  goto  cinque

temp436
  movlw 0x5B
  xorwf ADRESH,W

  btfss STATUS,Z
  goto  temp437
  goto  sei

temp437
  movlw 0x5C
  xorwf ADRESH,W

  btfss STATUS,Z
  goto  temp438
  goto  sette

temp438
  movlw 0x5D
  xorwf ADRESH,W

  btfss STATUS,Z
  goto  temp439
  goto  otto

temp439
  movlw 0x5E
  xorwf ADRESH,W

  btfss STATUS,Z
  goto  temp440
  goto  nove

temp440
  movlw '4'
  movwf TEMPERATURA2

  movlw 0x5F
  xorwf ADRESH,W

  btfss STATUS,Z
  goto  temp441
  goto  zero

```

```
temp441
  movlw 0x60
  xorwf ADRESH,W

  btfss STATUS,Z
  goto temp442
  goto uno
```

```
temp442
  movlw 0x61
  xorwf ADRESH,W

  btfss STATUS,Z
  goto temp443
  goto due
```

```
temp443
  movlw 0x62
  xorwf ADRESH,W

  btfss STATUS,Z
  goto temp444
  goto tre
```

```
temp444
  movlw 0x63
  xorwf ADRESH,W

  btfss STATUS,Z
  goto temp445
  goto quattro
```

```
temp445
  movlw 0x64
  xorwf ADRESH,W

  btfss STATUS,Z
  goto temp446
  goto cinque
```

```
temp446
  movlw 0x65
  xorwf ADRESH,W

  btfss STATUS,Z
  goto temp447
  goto sei
```

```

temp447
  movlw 0x66
  xorwf ADRESH,W

  btfss STATUS,Z
  goto  temp448
  goto  sette

temp448
  movlw 0x67
  xorwf ADRESH,W

  btfss STATUS,Z
  goto  temp449
  goto  otto

temp449
  movlw 0x68
  xorwf ADRESH,W

  btfss STATUS,Z
  goto  temp450
  goto  nove

temp450
  movlw '5'
  movwf TEMPERATURA2

  movlw 0x69
  xorwf ADRESH,W

  btfss STATUS,Z
  goto  temp451
  goto  zero

temp451
  movlw 0x6A
  xorwf ADRESH,W

  btfss STATUS,Z
  goto  temp452
  goto  uno

temp452
  movlw 0x6B
  xorwf ADRESH,W

  btfss STATUS,Z
  goto  temp453
  goto  due

```

```
temp453
  movlw 0x6C
  xorwf ADRESH,W

  btfss STATUS,Z
  goto temp454
  goto tre
```

```
temp454
  movlw 0x6D
  xorwf ADRESH,W

  btfss STATUS,Z
  goto temp455
  goto quattro
```

```
temp455
  movlw 0x6E
  xorwf ADRESH,W

  btfss STATUS,Z
  goto temp456
  goto cinque
```

```
temp456
  movlw 0x6F
  xorwf ADRESH,W

  btfss STATUS,Z
  goto temp457
  goto sei
```

```
temp457
  movlw 0x70
  xorwf ADRESH,W

  btfss STATUS,Z
  goto temp458
  goto sette
```

```
temp458
  movlw 0x71
  xorwf ADRESH,W

  btfss STATUS,Z
  goto temp459
  goto otto
```

```
temp459
  movlw 0x72
  xorwf ADRESH,W

  btfss STATUS,Z
  goto temp460
  goto nove
```

```
temp460
  movlw '6'
  movwf TEMPERATURA2

  movlw 0x73
  xorwf ADRESH,W

  btfss STATUS,Z
  goto temp461
  goto zero
```

```
temp461
  movlw 0x74
  xorwf ADRESH,W

  btfss STATUS,Z
  goto temp462
  goto uno
```

```
temp462
  movlw 0x75
  xorwf ADRESH,W

  btfss STATUS,Z
  goto temp463
  goto due
```

```
temp463
  movlw 0x76
  xorwf ADRESH,W

  btfss STATUS,Z
  goto temp464
  goto tre
```

```
temp464
  movlw 0x77
  xorwf ADRESH,W

  btfss STATUS,Z
  goto temp465
  goto quattro
```

```

temp465
  movlw 0x78
  xorwf ADRESH,W

  btfss STATUS,Z
  goto temp466
  goto cinque

temp466
  movlw 0x79
  xorwf ADRESH,W

  btfss STATUS,Z
  goto temp467
  goto sei

temp467
  movlw 0x7A
  xorwf ADRESH,W

  btfss STATUS,Z
  goto temp468
  goto sette

temp468
  movlw 0x7B
  xorwf ADRESH,W

  btfss STATUS,Z
  goto temp469
  goto otto

temp469
  movlw 0x7C
  xorwf ADRESH,W

  btfss STATUS,Z
  goto temp470
  goto nove

temp470
  movlw '7'
  movwf TEMPERATURA2

  movlw 0x7D
  xorwf ADRESH,W

  btfss STATUS,Z
  goto temp471
  goto zero

```

```
temp471
  movlw  0x7E
  xorwf  ADRESH,W

  btfss  STATUS,Z
  goto   temp472
  goto   uno
```

```
temp472
  movlw  0x7F
  xorwf  ADRESH,W

  btfss  STATUS,Z
  goto   temp473
  goto   due
```

```
temp473
  movlw  0x80
  xorwf  ADRESH,W

  btfss  STATUS,Z
  goto   temp474
  goto   tre
```

```
temp474
  movlw  0x81
  xorwf  ADRESH,W

  btfss  STATUS,Z
  goto   temp475
  goto   quattro
```

```
temp475
  movlw  0x82
  xorwf  ADRESH,W

  btfss  STATUS,Z
  goto   temp476
  goto   cinque
```

```
temp476
  movlw  0x83
  xorwf  ADRESH,W

  btfss  STATUS,Z
  goto   temp477
  goto   sei
```

```
temp477
  movlw 0x84
  xorwf ADRESH,W

  btfss STATUS,Z
  goto temp478
  goto sette

temp478
  movlw 0x85
  xorwf ADRESH,W

  btfss STATUS,Z
  goto temp479
  goto otto

temp479
  movlw 0x86
  xorwf ADRESH,W

  btfss STATUS,Z
  goto temp480
  goto nove

temp480
  movlw '8'
  movwf TEMPERATURA2

  movlw 0x87
  xorwf ADRESH,W

  btfss STATUS,Z
  goto temp481
  goto zero

temp481
  movlw 0x88
  xorwf ADRESH,W

  btfss STATUS,Z
  goto temp482
  goto uno

temp482
  movlw 0x89
  xorwf ADRESH,W

  btfss STATUS,Z
  goto temp483
  goto due
```

```
temp483
  movlw 0x8A
  xorwf ADRESH,W

  btfss STATUS,Z
  goto  temp484
  goto  tre
```

```
temp484
  movlw 0x8B
  xorwf ADRESH,W

  btfss STATUS,Z
  goto  temp485
  goto  quattro
```

```
temp485
  movlw 0x8C
  xorwf ADRESH,W

  btfss STATUS,Z
  goto  temp486
  goto  cinque
```

```
temp486
  movlw 0x8D
  xorwf ADRESH,W

  btfss STATUS,Z
  goto  temp487
  goto  sei
```

```
temp487
  movlw 0x8E
  xorwf ADRESH,W

  btfss STATUS,Z
  goto  temp488
  goto  sette
```

```
temp488
  movlw 0x8F
  xorwf ADRESH,W

  btfss STATUS,Z
  goto  temp489
  goto  otto
```

```

temp489
  movlw 0x90
  xorwf ADRESH,W

  btfss STATUS,Z
  goto temp490
  goto nove

temp490
  movlw '9'
  movwf TEMPERATURA2

  movlw 0x91
  xorwf ADRESH,W

  btfss STATUS,Z
  goto temp491
  goto zero

temp491
  movlw 0x92
  xorwf ADRESH,W

  btfss STATUS,Z
  goto temp492
  goto uno

temp492
  movlw 0x93
  xorwf ADRESH,W

  btfss STATUS,Z
  goto temp493
  goto due

temp493
  movlw 0x94
  xorwf ADRESH,W

  btfss STATUS,Z
  goto temp494
  goto tre

temp494
  movlw 0x95
  xorwf ADRESH,W

  btfss STATUS,Z
  goto temp495
  goto quattro

```

```

temp495
  movlw 0x96
  xorwf ADRESH,W

  btfss STATUS,Z
  goto temp496
  goto cinque

temp496
  movlw 0x97
  xorwf ADRESH,W

  btfss STATUS,Z
  goto temp497
  goto sei

temp497
  movlw 0x98
  xorwf ADRESH,W

  btfss STATUS,Z
  goto temp498
  goto sette

temp498
  movlw 0x99
  xorwf ADRESH,W

  btfss STATUS,Z
  goto temp499
  goto otto

temp499
  movlw 0x9A
  xorwf ADRESH,W

  btfss STATUS,Z
  goto temp500
  goto nove

temp500
  movlw '5'
  movwf TEMPERATURA1 ;cambia il valore della cifra delle decine
  movlw '0'
  movwf TEMPERATURA2

  movlw 0x9B
  xorwf ADRESH,W

  btfss STATUS,Z
  goto temp501
  goto zero

```

```
temp501
  movlw 0x9C
  xorwf ADRESH,W

  btfss STATUS,Z
  goto temp502
  goto uno
```

```
temp502
  movlw 0x9D
  xorwf ADRESH,W

  btfss STATUS,Z
  goto temp503
  goto due
```

```
temp503
  movlw 0x9E
  xorwf ADRESH,W

  btfss STATUS,Z
  goto temp504
  goto tre
```

```
temp504
  movlw 0x9F
  xorwf ADRESH,W

  btfss STATUS,Z
  goto temp505
  goto quattro
```

```
temp505
  movlw 0xA0
  xorwf ADRESH,W

  btfss STATUS,Z
  goto temp506
  goto cinque
```

```
temp506
  movlw 0xA1
  xorwf ADRESH,W

  btfss STATUS,Z
  goto temp507
  goto sei
```

```

temp507
  movlw  0xA2
  xorwf  ADRESH,W

  btfss  STATUS,Z
  goto   temp508
  goto   sette

temp508
  movlw  0xA3
  xorwf  ADRESH,W

  btfss  STATUS,Z
  goto   temp509
  goto   otto

temp509
  movlw  0xA4
  xorwf  ADRESH,W

  btfss  STATUS,Z
  goto   temp510
  goto   nove

temp510
  movlw  '1'
  movwf  TEMPERATURA2

  movlw  0xA5
  xorwf  ADRESH,W

  btfss  STATUS,Z
  goto   temp511
  goto   zero

temp511
  movlw  0xA6
  xorwf  ADRESH,W

  btfss  STATUS,Z
  goto   temp512
  goto   uno

temp512
  movlw  0xA7
  xorwf  ADRESH,W

  btfss  STATUS,Z
  goto   temp513
  goto   due

```

```
temp513
  movlw  0xA8
  xorwf  ADRESH,W

  btfss  STATUS,Z
  goto   temp514
  goto   tre
```

```
temp514
  movlw  0xA9
  xorwf  ADRESH,W

  btfss  STATUS,Z
  goto   temp515
  goto   quattro
```

```
temp515
  movlw  0xAA
  xorwf  ADRESH,W

  btfss  STATUS,Z
  goto   temp516
  goto   cinque
```

```
temp516
  movlw  0xAB
  xorwf  ADRESH,W

  btfss  STATUS,Z
  goto   temp517
  goto   sei
```

```
temp517
  movlw  0xAC
  xorwf  ADRESH,W

  btfss  STATUS,Z
  goto   temp518
  goto   sette
```

```
temp518
  movlw  0xAD
  xorwf  ADRESH,W

  btfss  STATUS,Z
  goto   temp519
  goto   otto
```

```
temp519
  movlw 0xAE
  xorwf ADRESH,W

  btfss STATUS,Z
  goto temp520
  goto nove
```

```
temp520
  movlw '2'
  movwf TEMPERATURA2

  movlw 0xAF
  xorwf ADRESH,W

  btfss STATUS,Z
  goto temp521
  goto zero
```

```
temp521
  movlw 0xB0
  xorwf ADRESH,W

  btfss STATUS,Z
  goto temp522
  goto uno
```

```
temp522
  movlw 0xB1
  xorwf ADRESH,W

  btfss STATUS,Z
  goto temp523
  goto due
```

```
temp523
  movlw 0xB2
  xorwf ADRESH,W

  btfss STATUS,Z
  goto temp524
  goto tre
```

```
temp524
  movlw 0xB3
  xorwf ADRESH,W

  btfss STATUS,Z
  goto temp525
  goto quattro
```

```

temp525
  movlw  0xB4
  xorwf  ADRESH,W

  btfss  STATUS,Z
  goto   temp526
  goto   cinque

temp526
  movlw  0xB5
  xorwf  ADRESH,W

  btfss  STATUS,Z
  goto   temp527
  goto   sei

temp527
  movlw  0xB6
  xorwf  ADRESH,W

  btfss  STATUS,Z
  goto   temp528
  goto   sette

temp528
  movlw  0xB7
  xorwf  ADRESH,W

  btfss  STATUS,Z
  goto   temp529
  goto   otto

temp529
  movlw  0xB8
  xorwf  ADRESH,W

  btfss  STATUS,Z
  goto   temp530
  goto   nove

temp530
  movlw  '3'
  movwf  TEMPERATURA2

  movlw  0xB9
  xorwf  ADRESH,W

  btfss  STATUS,Z
  goto   temp531
  goto   zero

```

```
temp531
  movlw 0xBA
  xorwf ADRESH,W

  btfss STATUS,Z
  goto temp532
  goto uno
```

```
temp532
  movlw 0xBB
  xorwf ADRESH,W

  btfss STATUS,Z
  goto temp533
  goto due
```

```
temp533
  movlw 0xBC
  xorwf ADRESH,W

  btfss STATUS,Z
  goto temp534
  goto tre
```

```
temp534
  movlw 0xBD
  xorwf ADRESH,W

  btfss STATUS,Z
  goto temp535
  goto quattro
```

```
temp535
  movlw 0xBE
  xorwf ADRESH,W

  btfss STATUS,Z
  goto temp536
  goto cinque
```

```
temp536
  movlw 0xBF
  xorwf ADRESH,W

  btfss STATUS,Z
  goto temp537
  goto sei
```

```

temp537
  movlw  0xC0
  xorwf  ADRESH,W

  btfss  STATUS,Z
  goto   temp538
  goto   sette

temp538
  movlw  0xC1
  xorwf  ADRESH,W

  btfss  STATUS,Z
  goto   temp539
  goto   otto

temp539
  movlw  0xC2
  xorwf  ADRESH,W

  btfss  STATUS,Z
  goto   temp540
  goto   nove

temp540
  movlw  '4'
  movwf  TEMPERATURA2

  movlw  0xC3
  xorwf  ADRESH,W

  btfss  STATUS,Z
  goto   temp541
  goto   zero

temp541
  movlw  0xC4
  xorwf  ADRESH,W

  btfss  STATUS,Z
  goto   temp542
  goto   uno

temp542
  movlw  0xC5
  xorwf  ADRESH,W

  btfss  STATUS,Z
  goto   temp543
  goto   due

```

```
temp543
  movlw 0xC6
  xorwf ADRESH,W

  btfss STATUS,Z
  goto temp544
  goto tre
```

```
temp544
  movlw 0xC7
  xorwf ADRESH,W

  btfss STATUS,Z
  goto temp545
  goto quattro
```

```
temp545
  movlw 0xC8
  xorwf ADRESH,W

  btfss STATUS,Z
  goto temp546
  goto cinque
```

```
temp546
  movlw 0xC9
  xorwf ADRESH,W

  btfss STATUS,Z
  goto temp547
  goto sei
```

```
temp547
  movlw 0xCA
  xorwf ADRESH,W

  btfss STATUS,Z
  goto temp548
  goto sette
```

```
temp548
  movlw 0xCB
  xorwf ADRESH,W

  btfss STATUS,Z
  goto temp549
  goto otto
```

```

temp549
  movlw 0xCC
  xorwf ADRESH,W

  btfss STATUS,Z
  goto temp550
  goto nove

temp550
  movlw '5'
  movwf TEMPERATURA2

  movlw 0xCD
  xorwf ADRESH,W

  btfss STATUS,Z
  goto temp551
  goto zero

temp551
  movlw 0xCE
  xorwf ADRESH,W

  btfss STATUS,Z
  goto temp552
  goto uno

temp552
  movlw 0xCF
  xorwf ADRESH,W

  btfss STATUS,Z
  goto temp553
  goto due

temp553
  movlw 0xD0
  xorwf ADRESH,W

  btfss STATUS,Z
  goto temp554
  goto tre

temp554
  movlw 0xD1
  xorwf ADRESH,W

  btfss STATUS,Z
  goto temp555
  goto quattro

```

```

temp555
  movlw  0xD2
  xorwf  ADRESH,W

  btfss  STATUS,Z
  goto   temp556
  goto   cinque

temp556
  movlw  0xD3
  xorwf  ADRESH,W

  btfss  STATUS,Z
  goto   temp557
  goto   sei

temp557
  movlw  0xD4
  xorwf  ADRESH,W

  btfss  STATUS,Z
  goto   temp558
  goto   sette

temp558
  movlw  0xD5
  xorwf  ADRESH,W

  btfss  STATUS,Z
  goto   temp559
  goto   otto

temp559
  movlw  0xD6
  xorwf  ADRESH,W

  btfss  STATUS,Z
  goto   temp560
  goto   nove

temp560
  movlw  '6'
  movwf  TEMPERATURA2

  movlw  0xD7
  xorwf  ADRESH,W

  btfss  STATUS,Z
  goto   temp561
  goto   zero

```

```
temp561
  movlw  0xD8
  xorwf  ADRESH,W

  btfss  STATUS,Z
  goto   temp562
  goto   uno
```

```
temp562
  movlw  0xD9
  xorwf  ADRESH,W

  btfss  STATUS,Z
  goto   temp563
  goto   due
```

```
temp563
  movlw  0xDA
  xorwf  ADRESH,W

  btfss  STATUS,Z
  goto   temp564
  goto   tre
```

```
temp564
  movlw  0xDB
  xorwf  ADRESH,W

  btfss  STATUS,Z
  goto   temp565
  goto   quattro
```

```
temp565
  movlw  0xDC
  xorwf  ADRESH,W

  btfss  STATUS,Z
  goto   temp566
  goto   cinque
```

```
temp566
  movlw  0xDD
  xorwf  ADRESH,W

  btfss  STATUS,Z
  goto   temp567
  goto   sei
```

```

temp567
  movlw  0xDE
  xorwfb ADRESH,W

  btfss  STATUS,Z
  goto   temp568
  goto   sette

temp568
  movlw  0xDF
  xorwfb ADRESH,W

  btfss  STATUS,Z
  goto   temp569
  goto   otto

temp569
  movlw  0xE0
  xorwfb ADRESH,W

  btfss  STATUS,Z
  goto   temp570
  goto   nove

temp570
  movlw  '7'
  movwfb TEMPERATURA2

  movlw  0xE1
  xorwfb ADRESH,W

  btfss  STATUS,Z
  goto   temp571
  goto   zero

temp571
  movlw  0xE2
  xorwfb ADRESH,W

  btfss  STATUS,Z
  goto   temp572
  goto   uno

temp572
  movlw  0xE3
  xorwfb ADRESH,W

  btfss  STATUS,Z
  goto   temp573
  goto   due

```

```
temp573
  movlw  0xE4
  xorwf  ADRESH,W

  btfss  STATUS,Z
  goto   temp574
  goto   tre
```

```
temp574
  movlw  0xE5
  xorwf  ADRESH,W

  btfss  STATUS,Z
  goto   temp575
  goto   quattro
```

```
temp575
  movlw  0xE6
  xorwf  ADRESH,W

  btfss  STATUS,Z
  goto   temp576
  goto   cinque
```

```
temp576
  movlw  0xE7
  xorwf  ADRESH,W

  btfss  STATUS,Z
  goto   temp577
  goto   sei
```

```
temp577
  movlw  0xE8
  xorwf  ADRESH,W

  btfss  STATUS,Z
  goto   temp578
  goto   sette
```

```
temp578
  movlw  0xE9
  xorwf  ADRESH,W

  btfss  STATUS,Z
  goto   temp579
  goto   otto
```

```

temp579
  movlw  0xEA
  xorwf  ADRESH,W

  btfss  STATUS,Z
  goto   temp580
  goto   nove

temp580
  movlw  '8'
  movwf  TEMPERATURA2

  movlw  0xEB
  xorwf  ADRESH,W

  btfss  STATUS,Z
  goto   temp581
  goto   zero

temp581
  movlw  0xEC
  xorwf  ADRESH,W

  btfss  STATUS,Z
  goto   temp582
  goto   uno

temp582
  movlw  0xED
  xorwf  ADRESH,W

  btfss  STATUS,Z
  goto   temp583
  goto   due

temp583
  movlw  0xEE
  xorwf  ADRESH,W

  btfss  STATUS,Z
  goto   temp584
  goto   tre

temp584
  movlw  0xEF
  xorwf  ADRESH,W

  btfss  STATUS,Z
  goto   temp585
  goto   quattro

```

```

temp585
  movlw  0xF0
  xorwf  ADRESH,W

  btfss  STATUS,Z
  goto   temp586
  goto   cinque

temp586
  movlw  0xF1
  xorwf  ADRESH,W

  btfss  STATUS,Z
  goto   temp587
  goto   sei

temp587
  movlw  0xF2
  xorwf  ADRESH,W

  btfss  STATUS,Z
  goto   temp588
  goto   sette

temp588
  movlw  0xF3
  xorwf  ADRESH,W

  btfss  STATUS,Z
  goto   temp589
  goto   otto

temp589
  movlw  0xF4
  xorwf  ADRESH,W

  btfss  STATUS,Z
  goto   temp590
  goto   nove

temp590
  movlw  '8'
  movwf  TEMPERATURA2

  movlw  0xF5
  xorwf  ADRESH,W

  btfss  STATUS,Z
  goto   temp591
  goto   zero

```

```
temp591
  movlw  0xF6
  xorwf  ADRESH,W

  btfss  STATUS,Z
  goto   temp592
  goto   uno
```

```
temp592
  movlw  0xF7
  xorwf  ADRESH,W

  btfss  STATUS,Z
  goto   temp593
  goto   due
```

```
temp593
  movlw  0xF8
  xorwf  ADRESH,W

  btfss  STATUS,Z
  goto   temp594
  goto   tre
```

```
temp594
  movlw  0xF9
  xorwf  ADRESH,W

  btfss  STATUS,Z
  goto   temp595
  goto   quattro
```

```
temp595
  movlw  0xFA
  xorwf  ADRESH,W

  btfss  STATUS,Z
  goto   temp596
  goto   cinque
```

```
temp596
  movlw  0xFB
  xorwf  ADRESH,W

  btfss  STATUS,Z
  goto   temp597
  goto   sei
```

```

temp597
  movlw    0xFC
  xorwf    ADRESH,W

  btfss    STATUS,Z
  goto     temp598
  goto     sette

temp598
  movlw    0xFD
  xorwf    ADRESH,W

  btfss    STATUS,Z
  goto     temp599
  goto     otto

temp599
  movlw    0xFE
  xorwf    ADRESH,W

  btfss    STATUS,Z
  goto     temp600
  goto     nove

temp600
  movlw    0xFF
  xorwf    ADRESH,W

  btfss    STATUS,Z
  goto     minore

  movlw    '6'
  movwf    TEMPERATURA1
  movlw    '0'
  movwf    TEMPERATURA2
  goto     zero

minore                                     ;temperatura non tra 40 e 60
  movlw    '4'
  movwf    TEMPERATURA1
  movlw    '0'
  movwf    TEMPERATURA2
  movlw    '0'
  movwf    TEMPERATURA3
  movlw    '<'
  movwf    SIMBOLO                                     ;carattere minore di
  goto     done

```

```
zero
  movlw '0'
  movwf TEMPERATURA3
  goto  done
```

```
uno
  movlw '1'
  movwf TEMPERATURA3
  goto  done
```

```
due
  movlw '2'
  movwf TEMPERATURA3
  goto  done
```

```
tre
  movlw '3'
  movwf TEMPERATURA3
  goto  done
```

```
quattro
  movlw '4'
  movwf TEMPERATURA3
  goto  done
```

```
cinque
  movlw '5'
  movwf TEMPERATURA3
  goto  done
```

```
sei
  movlw '6'
  movwf TEMPERATURA3
  goto  done
```

```
sette
  movlw '7'
  movwf TEMPERATURA3
  goto  done
```

```
otto
  movlw '8'
  movwf TEMPERATURA3
  goto  done
```

```
nove
    movlw    '9'
    movwf   TEMPERATURA3
    goto    done
```

```
done
    nop

    return
```

```
.*****
;
; subroutine di ritardo
; W = tempo di ritardo espresso in ms (clock = 4MHz) con W decimale
.*****
```

```
msDelay
    movwf   msDelayCounter+1
    clrf   msDelayCounter+0

msDelayLoop
    nop
    decfsz msDelayCounter+0,F
    goto   msDelayLoop
    nop

    decfsz msDelayCounter+1,F
    goto   msDelayLoop

    return
```

```
.*****
;
; inizializzazione del display LCD
; questa subroutine va chiamata prima di ogni altra subroutine LCD
.*****
```

```
LcdInit
    movlw   30                                ;caricato ritardo di circa 30 ms
    call   msDelay

    .*****
    ; sequenza di reset
    .*****

    bcf    PORTB,LCD_RS                       ;imposta LCD in modalita comandi

    bsf    PORTB,LCD_DB4                       ;invio sequenza direset al display
    bsf    PORTB,LCD_DB5
    bcf    PORTB,LCD_DB6
    bcf    PORTB,LCD_DB7
```

```

bsf      PORTB,LCD_E      ;abilita LCD
movlw   5                  ;attendi 5 ms
call    msDelay
bcf      PORTB,LCD_E      ;disabilita LCD
movlw   1                  ;attendi 1ms
call    msDelay

bsf      PORTB,LCD_E      ;abilita LCD
movlw   1                  ;attendi 1ms
call    msDelay
bcf      PORTB,LCD_E      ;disabilita LCD
movlw   1                  ;attendi 1ms
call    msDelay

bsf      PORTB,LCD_E      ;abilita E
movlw   1                  ;attendi 1ms
call    msDelay
bcf      PORTB,LCD_E      ;disabilita E
movlw   1                  ;attendi 1ms
call    msDelay

bcf      PORTB,LCD_DB4
bcf      PORTB,LCD_DB5
bcf      PORTB,LCD_DB6
bcf      PORTB,LCD_DB7

bsf      PORTB,LCD_E      ;abilita LCD
movlw   1                  ;attendi 1ms
call    msDelay
bcf      PORTB,LCD_E      ;disabilita LCD
movlw   1                  ;attendi 1ms
call    msDelay

movlw   28H                ;lunghezza bus dati impostata a 4 bit
call    LcdSendCommand

movlw   06H                ;con incremento e senza scorrimento
call    LcdSendCommand

movlw   0CH                ;display ON, cursor OFF, blink OFF
call    LcdSendCommand

call    LcdClear           ;cancella il display

return

```

```

;*****
;
; Clear LCD
;*****
;

```

```

LcdClear
  movlw  01H                ;cancella il display
  call   LcdSendCommand

```

```

movlw 2                                ;attendi 2 ms
call msDelay

movlw 80H                               ;DD RAM address set 1st digit
call LcdSendCommand

return

```

```

;*****
;
; posizionamento del cursore sul display LCD
; W = D7-D4 riga, D3-D0 colonna
;*****

```

```

LcdLocate
    movwf    tmpLcdRegister+0

    movlw   80H
    movwf   tmpLcdRegister+1

    movf    tmpLcdRegister+0,W
    andlw   0FH
    iorwf   tmpLcdRegister+1,F

    btfsc   tmpLcdRegister+0,4
    bsf     tmpLcdRegister+1,6

    movf    tmpLcdRegister+1,W
    call    LcdSendCommand

    return

```

```

;*****
;
; invio dati al display LCD
;*****

```

```

LcdSendData
    bsf     PORTB,LCD_RS
    call    LcdSendByte

    return

```

```

;*****
;
; invio di un comando al display LCD
;*****

```

```

LcdSendCommand
    bcf     PORTB,LCD_RS
    call    LcdSendByte

    return

```

```

;*****
;
; invio dati al display LCD tramite bus dati a 4 bit
;*****
;

```

LcdSendByte

```

    movwf    tmpLcdRegister                ;salvataggio delvalore da inviare

    bcf     PORTB,LCD_DB4                ;invio dei 4 bit piu significativi
    bcf     PORTB,LCD_DB5
    bcf     PORTB,LCD_DB6
    bcf     PORTB,LCD_DB7

    btfsc   tmpLcdRegister,4
    bsf     PORTB,LCD_DB4
    btfsc   tmpLcdRegister,5
    bsf     PORTB,LCD_DB5
    btfsc   tmpLcdRegister,6
    bsf     PORTB,LCD_DB6
    btfsc   tmpLcdRegister,7
    bsf     PORTB,LCD_DB7

    bsf     PORTB,LCD_E                  ;abilita LCD
    movlw   1                            ;attndi 1ms
    call    msDelay
    bcf     PORTB,LCD_E                  ;disabilita LCD
    movlw   1                            ;attendi 1ms
    call    msDelay

    bcf     PORTB,LCD_DB4                ;invio dei 4 bit meno significativi
    bcf     PORTB,LCD_DB5
    bcf     PORTB,LCD_DB6
    bcf     PORTB,LCD_DB7

    btfsc   tmpLcdRegister,0
    bsf     PORTB,LCD_DB4
    btfsc   tmpLcdRegister,1
    bsf     PORTB,LCD_DB5
    btfsc   tmpLcdRegister,2
    bsf     PORTB,LCD_DB6
    btfsc   tmpLcdRegister,3
    bsf     PORTB,LCD_DB7

    bsf     PORTB,LCD_E                  ;abilita LCD
    movlw   1                            ;attendi 1ms
    call    msDelay
    bcf     PORTB,LCD_E                  ;disabilita LCD
    movlw   1                            ;attendi 1ms
    call    msDelay

    return

    END

```

VIII – Descrizione del programma

7.1 – Direttive generali

È importante premettere che nel file.asm tutto ciò che compare dopo il punto e virgola non viene interpretato dal compilatore e viene utilizzato a puro titolo descrittivo. I primi codici che si incontrano sono delle direttive, che non sono delle istruzioni mnemoniche che il compilatore traduce nel rispettivo opcode, ma delle semplici indicazioni rivolte al compilatore per determinarne il funzionamento durante la compilazione.

LIST p=16F876

è una direttiva del compilatore assembler che consente di definire per quale microcontrollore è stato scritto il source. In questo caso il compilatore viene informato che le istruzioni che seguono sono relative ad un PIC16F876.

RADIX DEC

serve ad informare il compilatore che i numeri riportati senza notazione, sono da intendersi come numeri decimali.

INCLUDE "p16f876.inc"

è un'altra direttiva per comunicare al compilatore di includere nel source un secondo file denominato p16f846.inc. Esso si limiterà a sostituire questa linea con tutto il contenuto del file indicato ed effettuerà la compilazione come se fosse anch'esso parte del source.

ERRORLEVEL -302

La direttiva *ERRORLEVEL* serve ad escludere la segnalazione di alcuni errori di compilazione. Qui viene utilizzata per evitare che il compilatore ci segnali l'errore di tipo 302.

__CONFIG 0x3D71

ci permette tramite il valore da associare ad esso di settare il registro CONFIGURATION WORD. In questo caso è stata disabilitata la protezione della memoria FLASH; è stata disabilitata la funzione "in-circuit debug"; la memoria programma non può essere scritta tramite controllo EECON; la memoria EEPROM non è protetta; è stata disabilitata la funzione "in-circuit serial programming"; sono attivi il brown-out reset e il power-up timer; il watchdog timer è disattivo e infine è stato scelto un segnale di clock proveniente dall'esterno.

LCD_RS equ 2

questa direttiva ci consente di definire delle costanti simboliche all'interno del programma. In particolare la parola *LCD_RS* da questo punto in poi sarà equivalente al valore 2, al fine di rendere più leggibile il source.

ORG 0CH

questa direttiva ci consente di definire l'indirizzo da cui si vuole che il compilatore inizi ad allocare i dati o le istruzioni che lo seguono. In questo caso si sta definendo un'area dati all'interno del PIC, ovvero, un'area in cui memorizzare variabili e contatori durante l'esecuzione del programma. Quest'area coincide con la RAM, definita come area dei FILE REGISTER, che altro non sono che locazioni disponibili per l'utente a partire dall'indirizzo 0x0C.

```
tmpLcdRegister res 2
```

tmpLcdRegister è una label, che tramite la direttiva *res* indica al compilatore che si intende riservare un certo numero di byte, o meglio di file register, all'interno dell'area dati (2 byte in questo caso). Il nome della label è scelto dall'utente ed è un marcatore che nel resto del programma assumerà il valore dell'indirizzo in cui è stato inserito.

```
ORG 00H
```

dato che con la direttiva *ORG* precedente avevamo definito l'indirizzo di partenza a 0x0C, ora, con questa nuova direttiva, si fa riferimento ad un indirizzo in area programma (nella FLASH) anziché in area dati. Da questo punto in poi saranno infatti inserite le istruzioni mnemoniche che il compilatore dovrà poi convertire negli opportuni opcode per il PIC.

```
org 0x0000  
goto main
```

```
org 0x0004  
goto main
```

il primo opcode eseguito dal PIC dopo il reset è quello memorizzato nella locazione 0x0000, mentre il primo opcode eseguito dopo un interrupt è quello memorizzato in 0x0004. Con queste quattro istruzioni si rimanda il program counter in entrambi i casi alla label *main*, che corrisponde all'inizio del programma principale. La routine principale verrà analizzata in seguito in quanto a sua volta si appoggia su altre subroutine che conviene precedentemente analizzare.

7.2 – Subroutine *msDelay*

Questa subroutine è utile in tutti i punti del programma in cui si necessita di un ritardo software, ed esegue delle operazioni di nessuna utilità al fine di far trascorrere un tempo che viene precedentemente caricato nel registro W.

È stato approssimativamente calcolato (se si opera a 4MHz) che il valore caricato in W, in formato decimale, esprime i millesimi di secondo nei quali il PIC sarà impegnato in questa subroutine. Questa si appoggia su di una parte di memoria a 16 bit (*msDelayCounter*), preventivamente dichiarata, in quanto gli sono state riservate due locazioni di memoria. Appoggiandosi sul registro F e su di un loop interno, svolge successive operazioni di decremento fino a terminare il valore caricato, quindi ritorna al punto in cui c'era stata la chiamata.

7.3 – Subroutine *EsecuzioneRitardo*

Questa parte di programma serve per attendere e testare che la conversione analogico/digitale sia completata. Infatti, essa carica in W il valore 1, chiama la subroutine *msDelay* (cioè attende inizialmente 1ms) e trascorso tale lasso di tempo, va a testare il bit GO del registro ADCON0, che nel caso di completamento della conversione torna a 0. Il tempo caricato dovrebbe essere più che sufficiente all'ADC, ma nel caso in cui questo non avesse concluso le operazioni, ritorna ad inizio subroutine ripetendo quindi il ritardo di 1ms; nel caso di conversione completa, al contrario, salta l'operazione di richiamo del ritardo ed esce tornando al programma principale.

7.4 – Subroutine *TestTemperatura*

E' tra tutte quella più importante, poiché calcola tramite operazioni di confronto la temperatura. Deve essere chiamata logicamente dopo che il convertitore abbia concluso le operazioni, quindi con il risultato caricato nel

registro ADRESH. Inizialmente cancella le tre locazioni di memoria nelle quali verranno caricati i valori ASCII da inviare al display e carica il carattere uguale nella locazione SIMBOLO. Da qui poi parte una struttura iterativa che va a confrontare in modo crescente tutte le possibili temperature comprese nell'intervallo tra 40.0°C e 60.0°C, andando a caricare inizialmente il valore 4 per le decine (TEMPERATURA1) e il valore 0 per le unità (TEMPERATURA2). Viene confrontato il valore di ADRESH con il valore preventivamente calcolato per ogni singola temperatura (nella tabella del capitolo VI) e, nel caso questi siano uguali, si procede con il caricamento del valore dei decimi di grado (TEMPERATURA3) e si esce dalla subroutine. Inizialmente caricato in W il valore 0x37 (relativo alla temperatura di 40.0°C) viene eseguita l'operazione

XORWF ADRESH,W

che nel caso i due valori siano uguali, porta il bit Z del registro STATUS a uno, altrimenti rimane a zero. L'operazione successiva

btfss STATUS,Z

testa lo stato del bit Z e salta l'operazione successiva se questo è alto (ovvero salta l'operazione che porta al confronto con la temperatura successiva 40.1°C) e procede con il caricamento del valore mancante ovvero quello dei decimi. Se invece il bit Z è basso non salta l'operazione seguente che lo manda direttamente al confronto successivo. Si procede così per tutte le temperature possibili, caricando preventivamente i valori di decine ed unità ad ogni cambiamento di queste, in quanto lasciando come unico valore da caricare quello dei decimi, nel caso di confronto positivo, si snellisce di molto la scrittura del programma. Inoltre il caricamento dell'ultima cifra non è immediato, ma è eseguito tramite l'invio alla parte finale della subroutine, in modo tale che vengano scritti solamente dieci diversi

caricamenti di TEMPERATURA3 (da zero a nove), anziché doverli scrivere per ogni possibile risultato (ovvero 201 volte).

Nel caso in cui nessuno dei valori compresi tra 0x37 (40.0°C) e 0xFF (60.0°C) venisse riconosciuto all'interno di ADRESH, si è al di sotto del range di temperatura considerato e quindi la subroutine provvede a caricare in SIMBOLO il carattere minore, e 40.0 per quanto riguarda decine, unità e decimi, per comunicare, tramite il display, che la temperatura è sotto a tale valore.

7.5 – Subroutine *LcdInit*

Questa subroutine realizza la sequenza di inizializzazione per il display LCD illustrata nel capitolo ad esso dedicato. Ciò necessita inizialmente di un'attesa di almeno 15ms che viene realizzata tramite la subroutine *msDelay*. Successivamente vanno inviate quattro stringhe di quattro bit, mantenendo a livello logico basso i bit R/W che è direttamente connesso a massa, ed RS che viene impostato tramite l'istruzione

```
bcf    PORTB,LCD_RS
```

con simili istruzioni (*bcf* *PORTB,linea* e *bsf* *PORTB,linea*) si imposta a 0 o ad 1 il livello di ogni singola linea della porta B connessa al display, e quando ciò è stato eseguito, alzando il livello del bit E con una simile istruzione, si comunica al display di acquisire dati. Abbassando poi il livello di E il display non accetta più dati e il PIC può nuovamente lavorare per cambiare (se necessario) i livelli logici della porta B. Il tutto deve rispettare la tempistica specificata nel capitolo V, quindi, quando ci sarà bisogno di un ritardo, non si farà altro che richiamare la subroutine *msDelay*.

Successivamente vi è l'invio di stringhe di 8 bit sfruttando un'altra subroutine (che verrà successivamente illustrata), tramite i quali vengono impostati: la lunghezza del bus dati a 4 bit, la modalità di incremento senza scorrimento e l'attivazione del display senza la presenza del cursore.

7.6 – Subroutine *LcdSendData-Command-Byte*

Le due subroutine *LcdSendData* e *LcdSendCommand* si basano entrambe sulle linee di programma della sezione *LcdSendByte*, con una sola differenza nell'impostazione del bit RS. Infatti, ogni volta che si vorrà trasmettere informazione (sia dati che sia comandi) al display, bisognerà preventivamente caricare questa nel registro W; successivamente bisognerà chiamare la subroutine per inviare un dato (ovvero un carattere) o un comando, le quali settano il bit RS a 0 per inviare un comando (*LcdSendCommand*), oppure a 1 per inviare un dato (*LcdSendData*). Completato ciò e non avendo modificato il contenuto del registro W, entrambe fanno chiamata a *LcdSendByte*, in quanto la modalità di trasmissione dei byte è la stessa.

Che si tratti di un dato o di un comando, il metodo di inviare dati al display usufruisce del registro ausiliario *tmpLcdRegister* nel quale viene copiato il contenuto di W, che essendo utilizzato anche per impostare dei ritardi nell'invio dei dati, perderebbe, in caso contrario, i bit da trasmettere al display. In seguito vengono cancellati tutti i bit relativi all'uscita della porta B, e si impostano i primi quattro bit da inviare (i più significativi) tramite le seguenti istruzioni

```
btfs    tmpLcdRegister,4  
bsf     PORTB,LCD_DB4  
btfs    tmpLcdRegister,5  
bsf     PORTB,LCD_DB5  
btfs    tmpLcdRegister,6  
bsf     PORTB,LCD_DB6  
btfs    tmpLcdRegister,7  
bsf     PORTB,LCD_DB7
```

Queste eseguono un test su ogni singolo bit del registro *tmpLcdRegister* e nel caso che questo sia a livello logico alto, porta a 1 anche il corrispondente bit del registro di uscita della porta B; se il bit è a zero l'operazione viene saltata, lasciando il bit basso e si passa al test del bit successivo. Abilitando la ricezione dati del display (bit E portato a 1), attendendo 1ms per assicurarsi che questo abbia completato l'acquisizione, e riportando il bit E a zero, si può passare con analoghe istruzioni, dopo aver cancellato nuovamente tutti i bit di PORTB, a inviare i quattro bit meno significativi.

7.7 – Subroutine *LcdClear* e *LcdLocate*

Queste due subroutine si usano ogni qual volta si necessita di ripartire con una nuova scrittura sul display. *LcdClear* permette, tramite l'invio di due comandi, di cancellare tutto ciò che è presente sullo schermo e di comunicare ad esso che tutte le comunicazioni dati che riceverà in seguito sono relative al primo carattere. *LcdLocate* consente, tramite il valore preventivamente caricato nel registro W, di selezionare la riga e la colonna da cui partire con la scrittura dei caratteri. I quattro bit più significativi selezionano la riga, gli altri la colonna. Il valore di W non è un'istruzione da inviare al display, e quindi, tramite opportune operazioni, viene convertito nel valore del relativo comando e, tramite *LcdSendCommand*, inviato al display.

7.8 – Main routine

Una volta descritte tutte le parti di programma utilizzate nella main, si può procedere con la descrizione di quest'ultima. Nella prima parte vengono caricati i registri che impostano il corretto funzionamento delle porte necessarie del PIC. Ciò avviene cambiando banco di memoria a seconda della posizione del registro che andrà settato. Nel registro ADCON0 viene caricato un valore che: imposta il segnale di clock del modulo ADC a 1/8 del clock con cui esegue le operazioni il PIC; seleziona RA0 come il canale da cui prelevare il segnale da convertire; attiva il convertitore ma non lo rende operativo. Si procede poi con l'azzeramento dei registri di ingresso/uscita della porta A e B. I pin della porta A vengono poi impostati tutti in input, portando a livello logico alto tutti i bit del registro TRISA; i bit della porta B, al contrario, sono impostati tutti in output, azzerando il registro TRISB. ADCON1 viene infine caricato per giustificare a sinistra il risultato dell'convertitore, e per selezionare RA2 ed RA3 come pin ai quali far riferimento per le tensioni minima e massima del convertitore. Eseguito ciò viene effettuata una chiamata all'inizializzazione del display e in seguito il programma entra nella sezione contraddistinta dalla label *adcgo*, contenente le operazioni che, eseguite ricorsivamente, permettono di monitorare continuamente la temperatura del filtro YIG. Queste comprendono: l'attivazione della conversione (tramite il bit GO del registro ADCON0); l'attesa di fine conversione, facendo chiamata a *EsecuzioneRitardo*; il calcolo della temperatura con il relativo caricamento dei valori ASCII nelle opportune locazioni di memoria; la cancellazione del display (chiamata a *LcdClear*); il posizionamento del cursore alla sesta colonna della prima riga (tramite *LcdLocate*); l'invio in sequenza dei caratteri T-SIMBOLO-DECINE-UNITA'-PUNTO-DECIMI-GRADO-C necessari per visualizzare la corretta temperatura (chiamando *LcdSendData* e avendo preventivamente caricato il relativo valore ASCII in W). Simbolo può essere "=" o "<", in quanto la temperatura può essere al di sotto dei

40.0°C. Infine si attendono 200ms prima di tornare alla label *adcgo*, in quanto si rischierebbe, in caso contrario, di cancellare il display pochi attimi dopo l'invio degli ultimi caratteri (operazione che renderebbe umanamente quasi invisibili questi ultimi). Il programma gira quindi all'infinito una volta alimentato il PIC, ripetendo tutte le istruzioni che partono da *adcgo* fino al termine della routine *main*.

IX – Realizzazione e verifica sperimentale

9.1 – Realizzazione pratica

Il circuito completo è quello visibile in figura 9.1. Questo comprende tutti i componenti necessari per realizzare concretamente il progetto. Sulla scheda da realizzare, al posto delle tensioni d'ingresso provenienti dagli alimentatori (+24V, +15V, -15V, GND), del Pt-100 e del display LCD sono stati posti tre appositi connettori, in quanto questi dispositivi non si trovano fisicamente sulla scheda. Per convertire la tensione a 5V è stato scelto un TMA2405, ossia un convertitore DC-DC, composto da una porta d'entrata e da una di uscita, in grado di fornire ampiamente la corrente di cui necessitano tutti i dispositivi. I quattro amplificatori operazionali sono tutti integrati dentro un unico componente, il TL084. Il PIC è montato su uno zocchetto, per rendere possibile una sua semplice rimozione in caso si necessiti di una nuova programmazione. I quattro trimmer relativi alle tensioni di riferimento dell'ADC sono stati scelti multigiri per consentire una migliore regolazione dei valori di resistenza e quindi delle relative tensioni, mentre, quello per il contrasto del display è sufficiente monogiro.

Il layout del circuito è visibile in figura 9.2; sono state poi stampate le piste (quelle in rosso) su carta trasparente, posizionate sopra la basetta di bachelite ricoperta di rame e di resist fotosensibile. Con l'ausilio del bromografo, della soda e dell'acido è stata ottenuta la basetta su cui saldare tutti componenti e i ponti (visibili in celeste in figura 9.2).

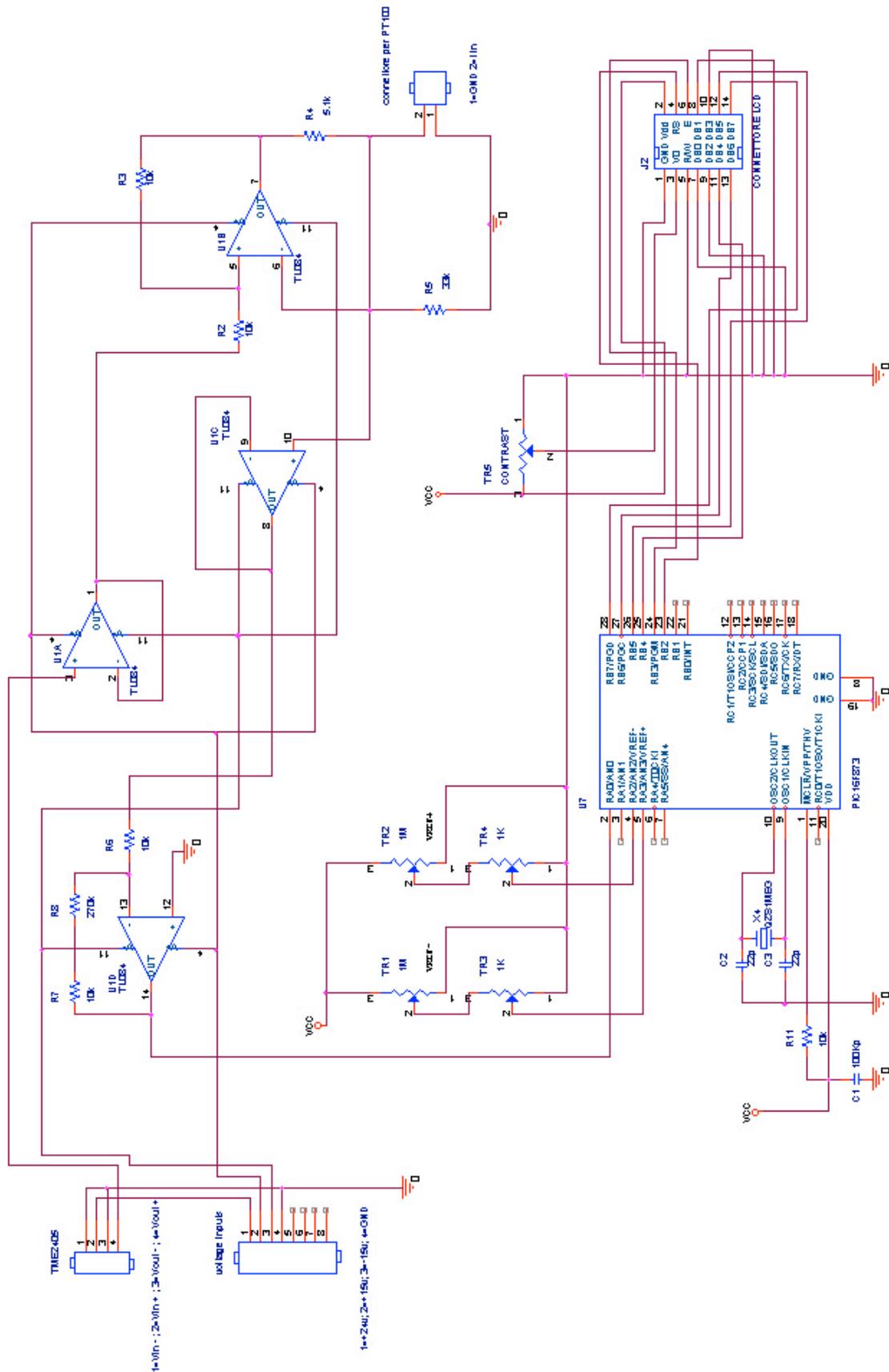


Figura 9.1: circuito completo

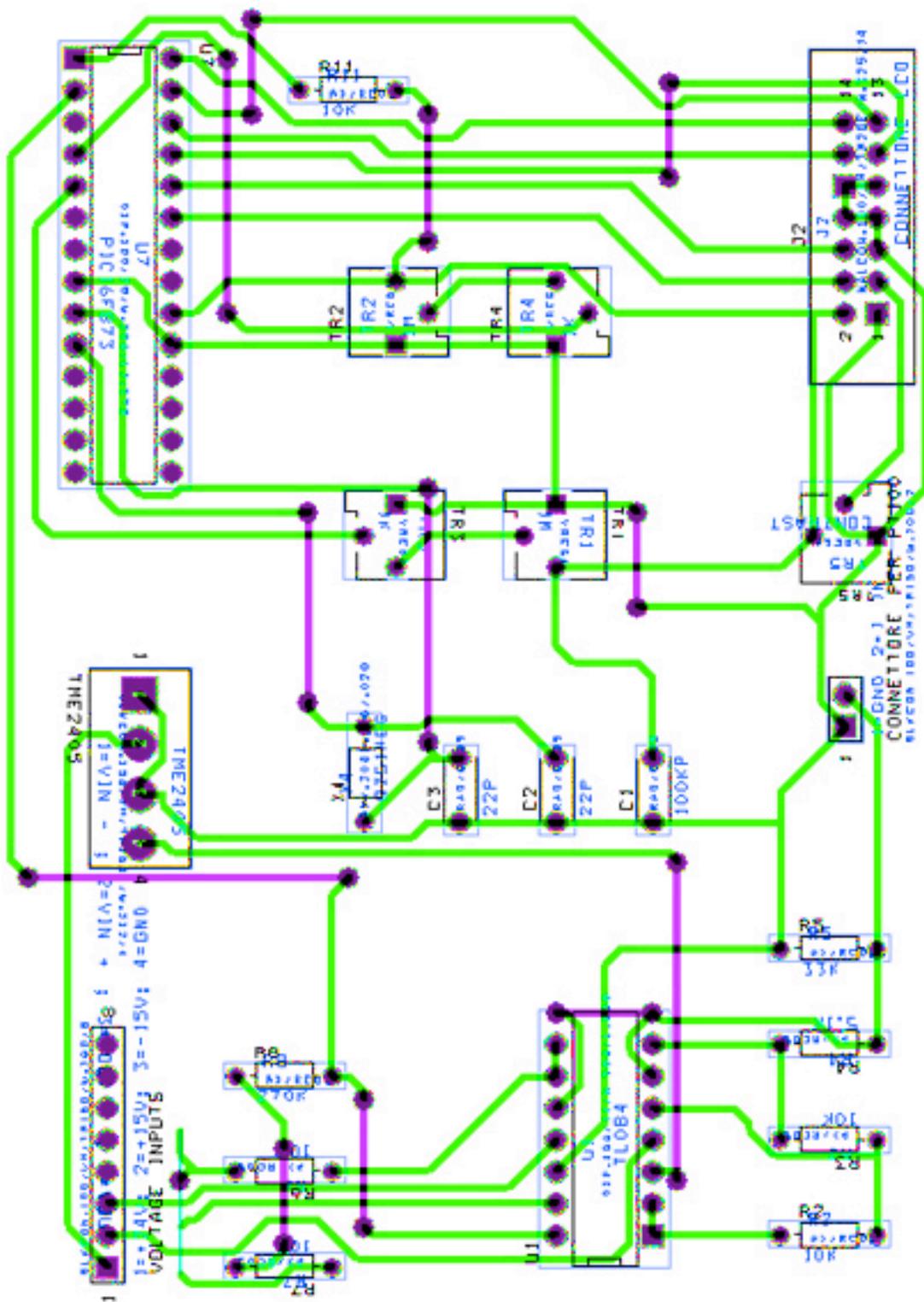


Figura 9.2: layout del circuito

9.2 – Test di laboratorio

Il circuito realizzato è quello mostrato in figura 9.3, e, nelle due figure successive (9.4 e 9.5), è possibile vedere il circuito collegato con gli alimentatori e il display mostrare la temperatura a cui si trova la termoresistenza (ovvero minore di 40°C in quanto posto a temperatura ambiente). È stata necessaria la taratura dei quattro trimmer, monitorando continuamente durante la regolazione le tensioni di riferimento con un tester ad altissima definizione. Procedendo poi con il test di tutte le singole tensioni in ogni punto del circuito, si riscontra che queste risultano essere (a meno di un fattore di errore trascurabile) quelle previste dalla simulazione. Il circuito risulta pienamente funzionante e pronto per essere utilizzato per lo scopo secondo cui è stato progettato.

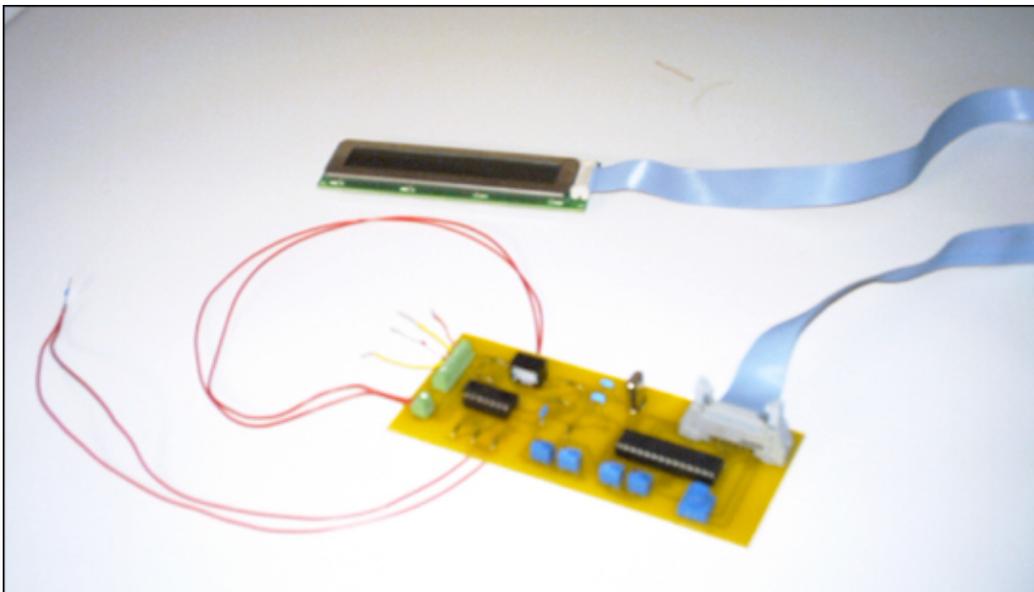


Figura 9.3: circuito realizzato

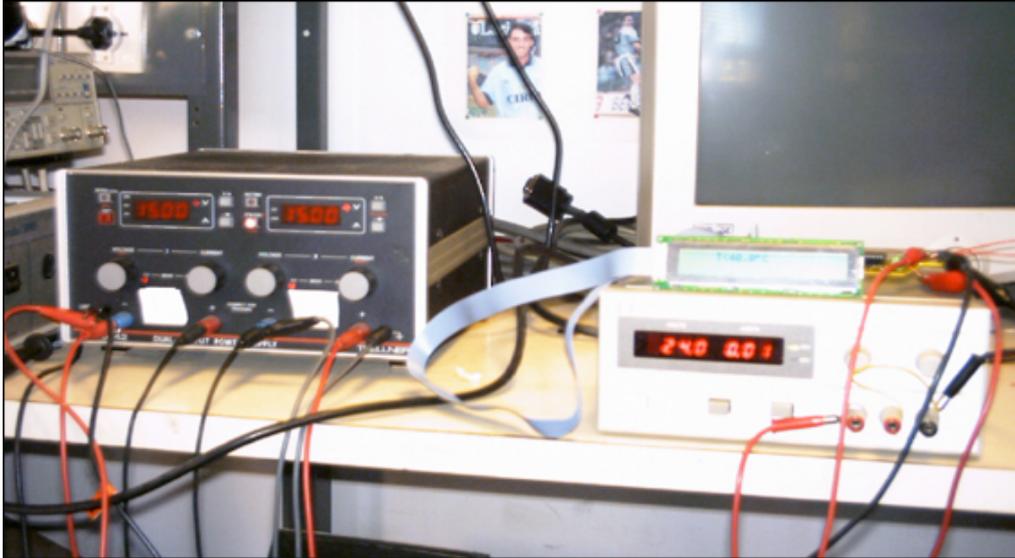


Figura 9.4: test di laboratorio

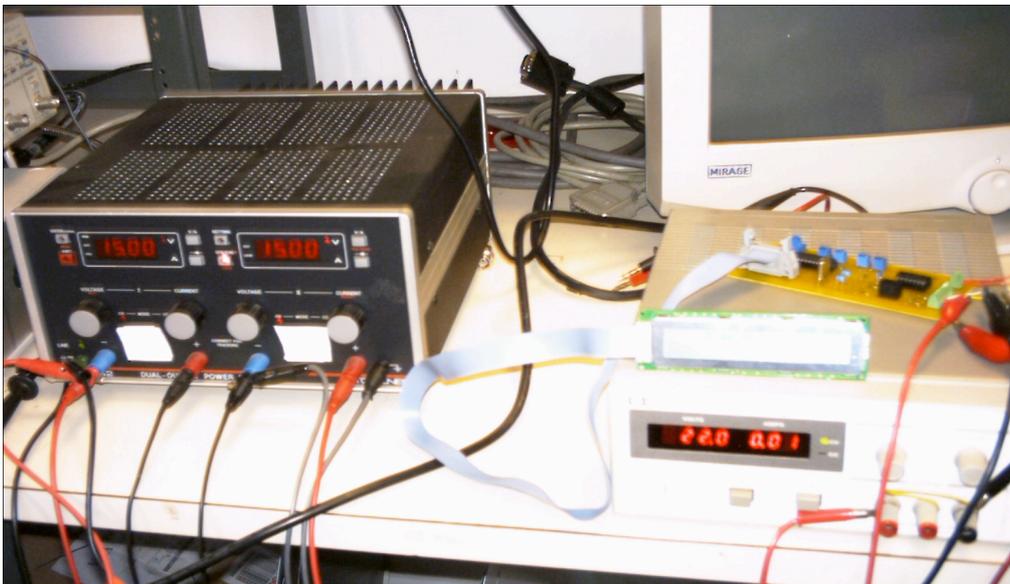


Figura 9.5: test di laboratorio

Conclusioni

Il progetto appena descritto ha portato alla realizzazione di un sistema digitale per la rilevazione della temperatura. Da diversi anni questi strumenti sono presenti in commercio e stanno ormai sostituendo quasi del tutto i classici termometri a mercurio. Sono innumerevoli le soluzioni che si possono adottare per realizzare tale misuratore. Nel caso appena analizzato è stata percorsa una strada dettata in parte da specifiche progettuali (ovvero dal tipo di misurazione da effettuare), in parte da considerazioni fatte sul caso, che hanno portato a semplificare le operazioni da svolgere nel trasformare la temperatura in codice binario ASCII, non perdendo in ogni modo elementi di affidabilità nella precisione della misura.

Il circuito realizzato, testato con successo, presenta un'alta versatilità: questo rileva temperature tra 40.0°C e 60.0°C, ma è possibile, con delle semplici modifiche sul programma, cambiare le sue specifiche per adattarlo a situazioni diverse. Si può traslare, in alto o in basso, il range di 20°C misurabili oppure si può espandere questo a piacimento. La minima variazione di temperatura rilevabile può essere più o meno grande rispetto al decimo di grado celsius considerato nel progetto. Questo è possibile anche grazie alla presenza dei quattro trimmer per regolare le tensioni di riferimento. Inoltre implementando delle semplici routine, che svolgono calcoli matematici di somma e sottrazione, è possibile visualizzare la temperatura in altre scale o mostrare la sua variazione in un arco di tempo prestabilito. In ultima analisi il Pt-100, essendo montato su un supporto rimovibile, può essere sostituito in caso di danneggiamento o in caso si voglia utilizzare una termoresistenza con caratteristiche diverse da quella attuale.

Bibliografia

- Lake Shore – Pt-100 Platinum RTDs
- Microchip – PIC16F87X data sheet
- RS data sheet - Powertip alphanumeric dot matrix liquid crystal displays