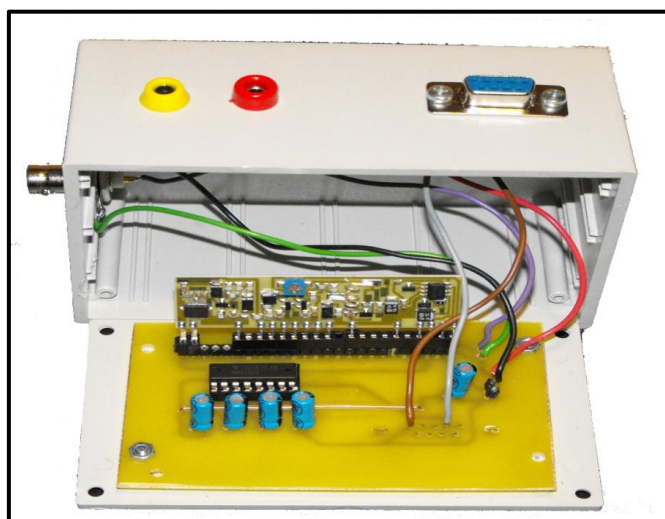
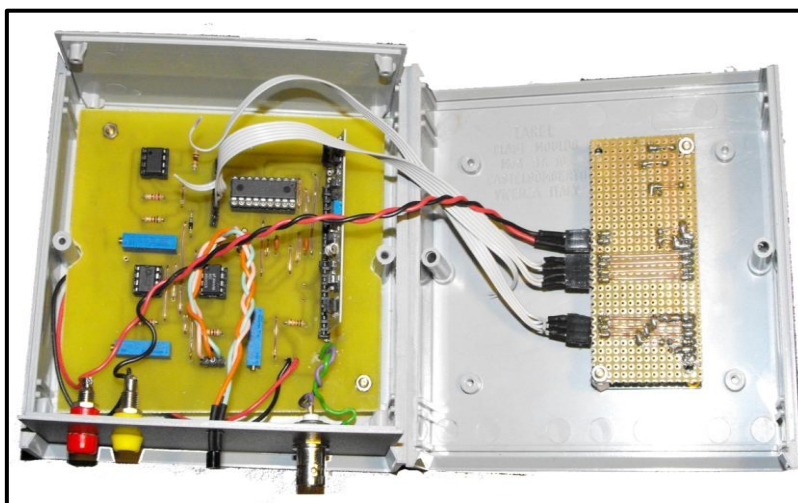


I.T.I.S. "C. ROSATELLI" DI RIETI
Classe V sez. A Ind. Elettronica e Telecomunicazioni
Esami di Maturità
A.S. 2008/2009

Candidato: Bufalino Andrea

**Sistema di acquisizione a distanza
che rileva la temperatura e la
visualizza su un PC tramite
un'applicazione in Visual Basic.**



Quest'anno, per l'area di progetto, si è scelto di realizzare un sistema che acquisisce una temperatura a distanza e la visualizza su un Personal Computer.

La scelta è ricaduta su questo tipo di progetto in quanto racchiude gran parte del programma svolto nelle materie tecniche del quinto anno di corso dell'ITIS.

Il progetto è stato svolto in gran parte nelle ore di T.D.P., visto che è la materia che ha più ore di laboratorio, mentre nelle altre materie si sono apprese le conoscenze teoriche di alcuni dei componenti utilizzati.

SPECIFICHE DI PROGETTO

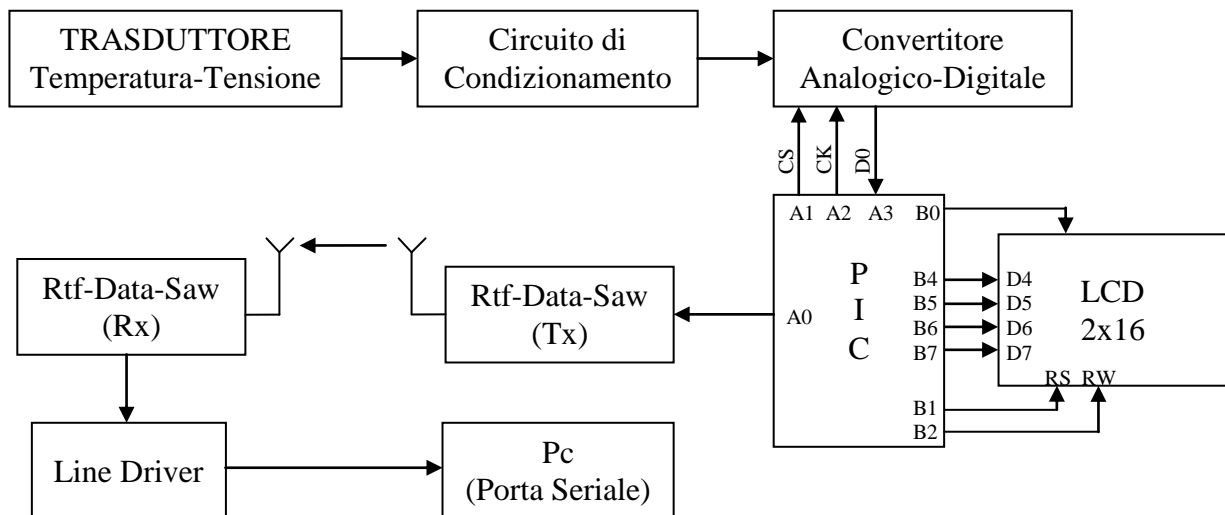
- Range di ingresso: $0 \div 100^{\circ}\text{C}$
- Trasmissione via etere tramite modulatori OOK a 433MHz
- Acquisizione della temperatura di tipo monocanale
- Acquisizione su PC di tipo seriale

SCHEMA A BLOCCHI DEL SISTEMA:

Nello schema in figura la temperatura è acquisita tramite un trasduttore (LM335) che la converte in tensione; tale valore non ha però il livello tale da poter essere acquisito da un P.C. quindi è necessario condizionarlo (amplificato) e convertirlo in digitale.

I dati digitali ottenuti dalla conversione sono inviati ad un microcontrollore che li elabora e li invia ad un display LCD e ad un trasmettitore con temporizzazione che rispetta lo standard RS232.

I dati che invia il trasmettitore sono captati dall'antenna di un ricevitore che li invia ad un adattatore di livello che trasforma i livelli di tensione TTL in livelli di tensione RS232, cioè formato elettrico acquisibile da un PC.



DETTAGLIO DEI BLOCCHI E SCELTA DEI COMPONENTI

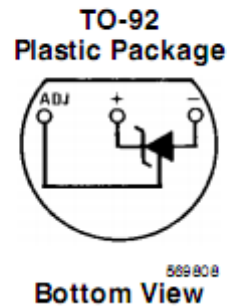
Trasduttore:

Si è deciso di utilizzare il trasduttore LM335 della National Semiconductor, i cui dati principali estratti dal datasheet sono i seguenti:

Features

- Directly calibrated in °Kelvin
- 1°C initial accuracy available
- Operates from 400 μ A to 5 mA
- Less than 1 Ω dynamic impedance
- Easily calibrated
- Wide operating temperature range
- 200°C overrange
- Low cost

Connection Diagrams



Absolute Maximum Ratings (Note 4)

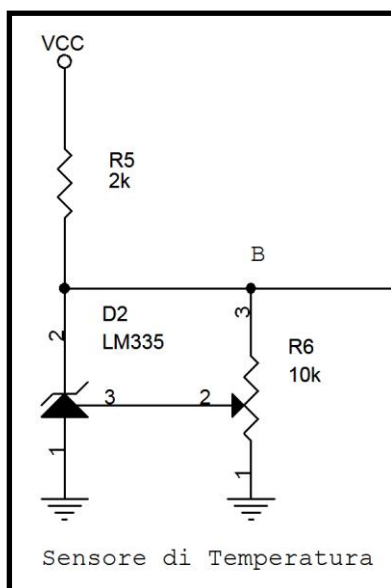
If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Reverse Current	15 mA
Forward Current	10 mA
Storage Temperature	
8-Pin SOIC Package	-65°C to 150°C
TO-92 Package	-60°C to 150°C
TO-46 Package	-60°C to 180°C

Specified Operating Temp. Range

	Continuous	Intermittent (Note 2)
LM135, LM135A	-55°C to 150°C	150°C to 200°C
LM235, LM235A	-40°C to 125°C	125°C to 150°C
LM335, LM335A	-40°C to 100°C	100°C to 125°C
Lead Temp. (Soldering, 10 seconds)		
8-Pin SOIC Package:		300°C
Vapor Phase (60 seconds):		215°C
Infrared (15 seconds):		220°C
TO-92 Package:		260°C
TO-46 Package:		300°C

Sappiamo che il trasduttore scelto fornisce una tensione d'uscita pari a 10mV per ogni variazione di grado Kelvin. In serie al trasduttore viene posta una resistenza in modo da limitarne la corrente infatti il funzionamento è garantito solo con una corrente compresa tra 400 μ A e 5mA, noi decidiamo di usare una corrente di 1mA. Il potenziometro R6 viene inserito per consentire la calibrazione (il valore di 10K è suggerito dal datasheet).



$$0^{\circ}\text{C} \Leftrightarrow 273^{\circ}\text{K} \Leftrightarrow 2,73 \text{ V}$$

$$100^{\circ}\text{C} \Leftrightarrow 373^{\circ}\text{K} \Leftrightarrow 3,73 \text{ V}$$

La R5 si è scelta pari a 2k perché come detto in precedenza si è decisa una corrente pari a 1mA.

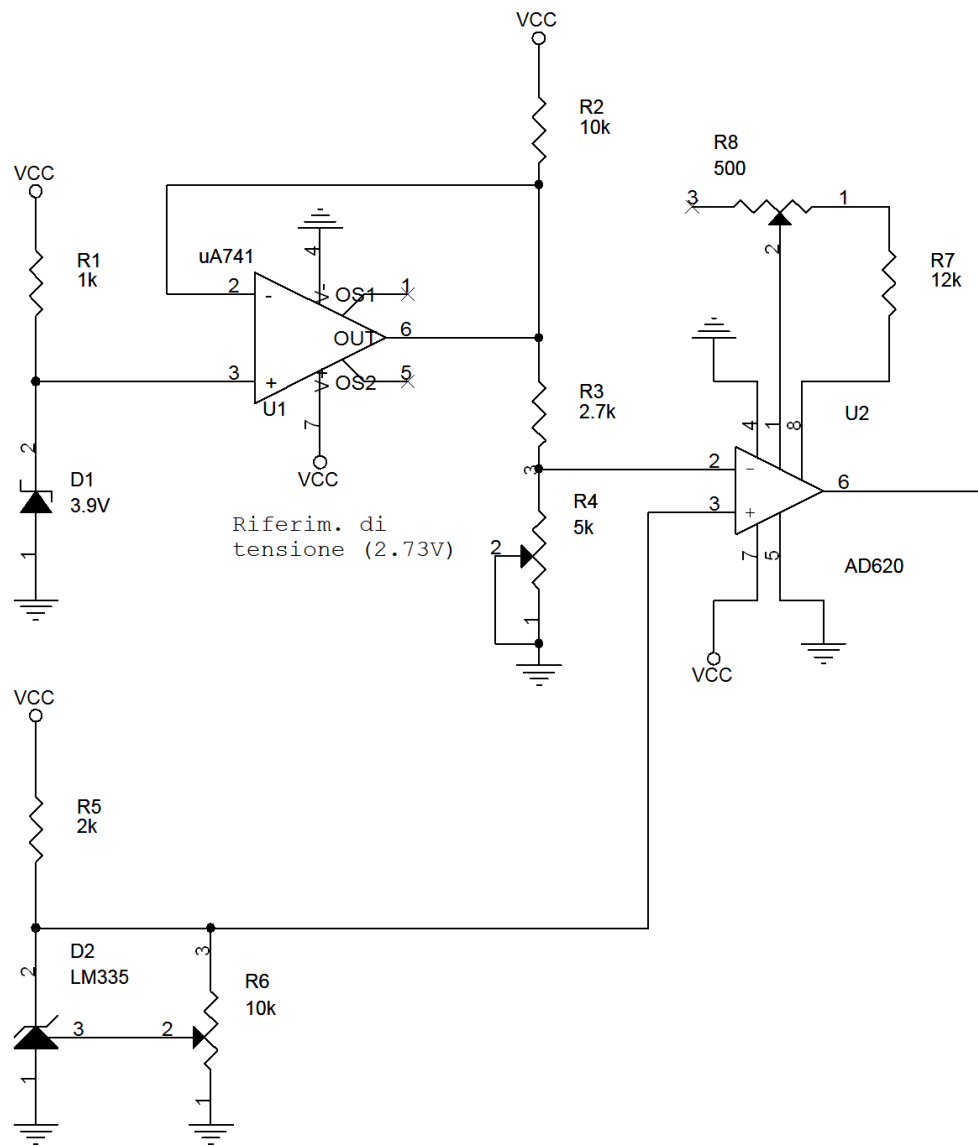
Dato che il sensore può essere visto come un diodo zener V_{LM335} la tensione che fornisce a temperatura ambiente è $20^{\circ}\text{C} = 293^{\circ}\text{K}$ $\Rightarrow 2,93\text{V}$.

$$R_3 = \frac{V}{I} = \frac{V_{CC} - V_{LM335}}{I} = \frac{5 - 2,93}{1 \cdot 10^{-3}} \cong 2\text{K}\Omega$$

Circuito di condizionamento:

Questo circuito deve essere progettato in modo tale che sia in grado di fornire una tensione d'uscita di 0V con in ingresso un valore di temperatura di 0°C e di +5V con in ingresso una temperatura di 100°C. Questo circuito viene inserito perché il convertitore analogico-digitale (ADC0831) a disposizione supporta in ingresso tensioni comprese tra 0V e +5V.

Teoricamente basterebbe un unico amplificatore operazionale in configurazione differenziale ma si è preferito utilizzare un amplificatore da strumentazione per ottenere una misura più precisa e stabile. Il circuito realizzato è il seguente:



Sensore di Temperatura

Il $\mu A741$ ha la funzione di buffer ed esso consente di avere un'uscita stabile di 3,9 V (valore dello zener). Attraverso il partitore composto da R3 ed il trimmer R4, opportunamente regolato, si ottiene una tensione di 2,73V (valore di tensione che avrebbe il sensore a $T=0^\circ\text{C}$). La R2 è una resistenza di pull-up. La R1 serve solo per limitare la corrente che attraversa il diodo D1.

L'amplificatore AD620 è un amplificatore per strumentazione, si è scelto di utilizzare questo tipo di amplificatore rispetto ad altri perché questo presenta una resistenza d'ingresso pressoché infinita (circa $10^{12} \Omega$) in più l'amplificazione è determinata da una singola resistenza e questo evita errori dovuti alla tolleranza (infatti se avessimo usato un $\mu A741$ in configurazione sottrattore sarebbe stato impossibile trovare resistenze uguali due a due). L'uscita del circuito quindi è data dalla differenza

amplificata tra la tensione in uscita dal trasduttore e la tensione presente sul terminale invertente (2,73V).

Con una temperatura di 0°C dobbiamo avere una tensione d'uscita di 0V e questo avviene perché il trasduttore a 0°C fornisce una tensione di 2,73 quindi $2,73(\text{tensione trasduttore}) - 2,73(\text{tensione terminale invertente}) = 0$ possiamo applicare un qualunque valore di amplificazione e sempre 0V rimane.

Per quanto riguarda l'altro estremo del range di temperatura con 100°C dobbiamo avere una tensione di uscita pari a 5V.

$$100^{\circ}\text{C} \rightarrow 3,73\text{V}$$

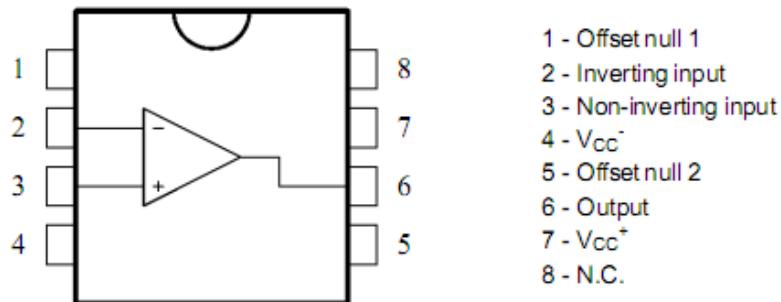
All'ingresso dell'AD620 abbiamo $3,73 - 2,73 = 1$ quindi per avere 5V di uscita dobbiamo amplificare di 5.

Il coefficiente di amplificazione dell'amplificatore per strumentazione è dato dal rapporto tra le sue resistenze interne e la resistenza R_g (resistenza vista tra i punti R_{G1} e R_{G2}) + 1. Dal datasheet si ha la seguente relazione: $\text{Gain} = \frac{49,4K}{R_g} + 1$.

Dovendo avere un gain pari a 5 quindi si ha che $R_g = 12350\Omega$.

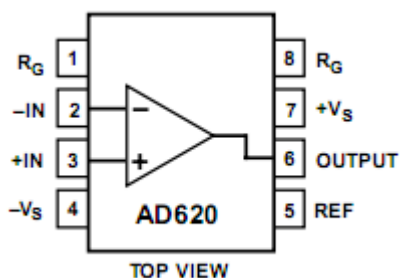
Per l'AD620 si consiglia di utilizzare resistenze con tolleranza dell'1%. In assenza di esse si può ricorrere all'uso di resistenze normali con tolleranza del 20% "corrette" con un potenziometro (metodo da noi utilizzato).

µA741:



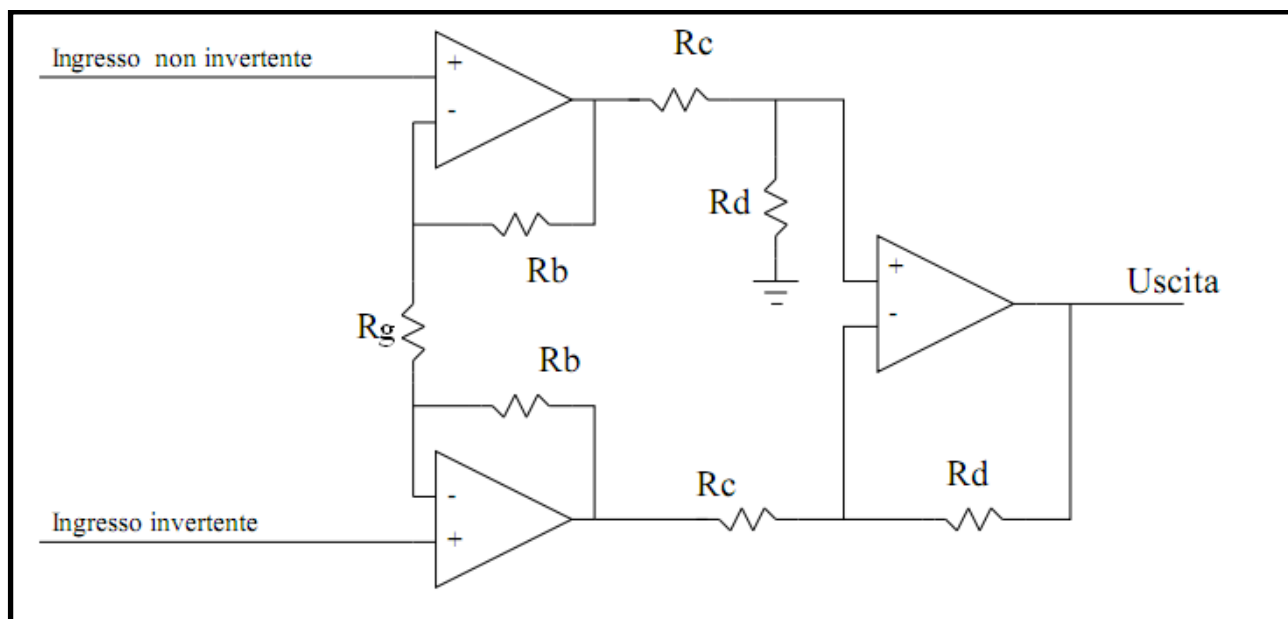
ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	UA741M	UA741I	UA741C	Unit
V_{CC}	Supply Voltage	± 22			V
V_{id}	Differential Input Voltage	± 30			V
V_i	Input Voltage	± 15			V
P_{tot}	Power Dissipation	500			mW
	Output Short-circuit Duration	Infinite			
T_{oper}	Operating Free Air Temperature Range	-55 to +125	-40 to +105	0 to +70	$^{\circ}\text{C}$
T_{stg}	Storage Temperature Range	-65 to +150	-65 to +150	-65 to +150	$^{\circ}\text{C}$

AD620:**ABSOLUTE MAXIMUM RATINGS¹**

Supply Voltage	± 18 V
Internal Power Dissipation ²	650 mW
Input Voltage (Common Mode)	$\pm V_S$
Differential Input Voltage	± 25 V
Output Short Circuit Duration	Indefinite
Storage Temperature Range (Q)	-65°C to $+150^{\circ}\text{C}$
Storage Temperature Range (N, R)	-65°C to $+125^{\circ}\text{C}$
Operating Temperature Range	
AD620 (A, B)	-40°C to $+85^{\circ}\text{C}$
AD620 (S)	-55°C to $+125^{\circ}\text{C}$
Lead Temperature Range	
(Soldering 10 seconds)	$+300^{\circ}\text{C}$

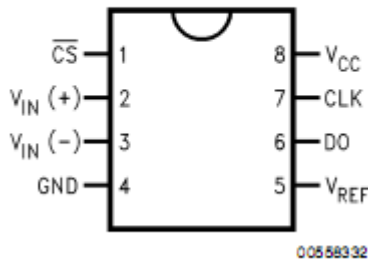
Configurazione base di un amplificatore per strumentazione:



Convertitore Analogico-Digitale:

Dovendo interfacciare il convertitore con un Pic si è deciso di utilizzare il convertitore seriale a 8 bit ADC0831 della National Semiconductor che ha le caratteristiche di seguito riportate.

ADC0831 Single Differential Input Dual-In-Line Package (N)



Key Specifications

■ Resolution	8 Bits
■ Total Unadjusted Error	$\pm \frac{1}{2}$ LSB and ± 1 LSB
■ Single Supply	5 V _{DC}
■ Low Power	15 mW
■ Conversion Time	32 μ s

Top View

Absolute Maximum Ratings (Notes 1, 2)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Current into V ⁺ (Note 3)	15 mA
Supply Voltage, V _{CC} (Note 3)	6.5V
Voltage	
Logic Inputs	-0.3V to V _{CC} + 0.3V
Analog Inputs	-0.3V to V _{CC} + 0.3V
Input Current per Pin (Note 4)	± 5 mA
Package	± 20 mA
Storage Temperature	-65°C to +150°C
Package Dissipation	
at T _A =25°C (Board Mount)	0.8W
Lead Temperature (Soldering 10 sec.)	

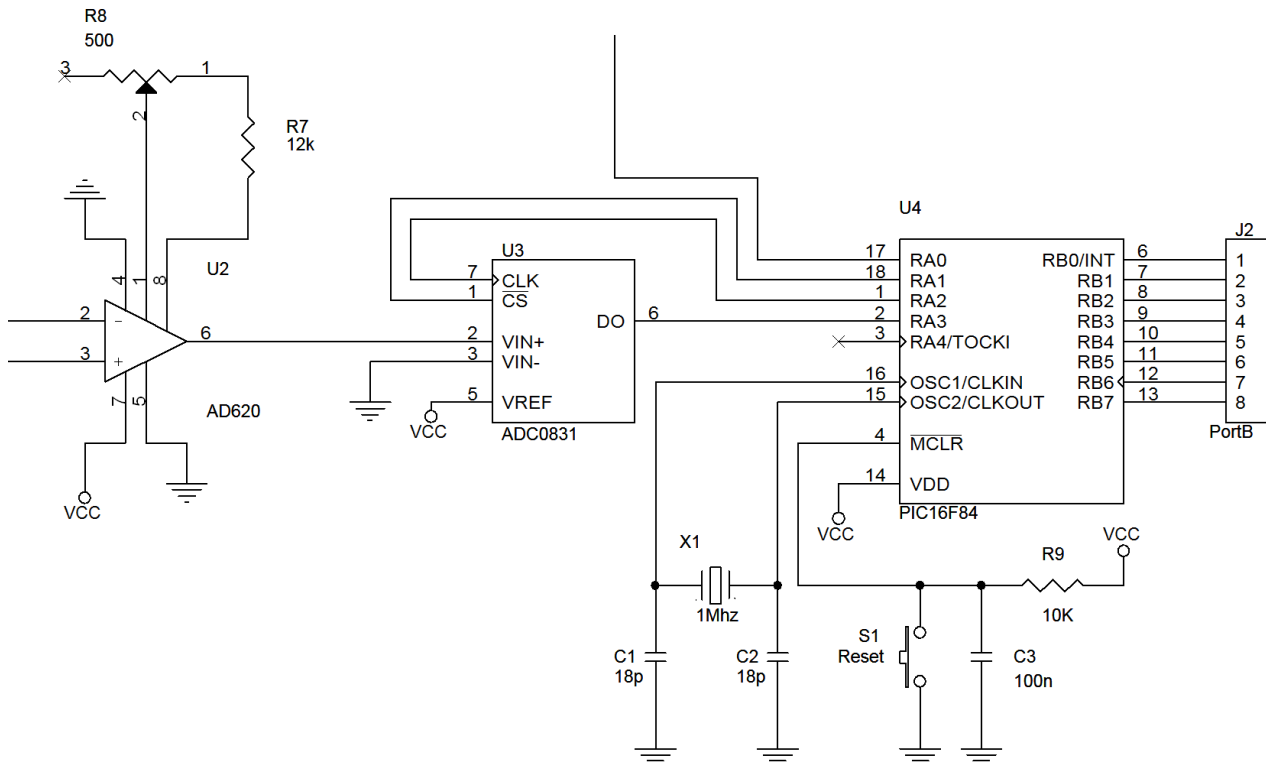
Dual-In-Line Package (Plastic)	260°C
Molded Chip Carrier Package	
Vapor Phase (60 sec.)	215°C
Infrared (15 sec.)	220°C
ESD Susceptibility (Note 5)	2000V

Operating Ratings (Notes 1, 2)

Supply Voltage, V _{CC}	4.5 V _{DC} to 6.3 V _{DC}
Temperature Range	T _{MIN} ≤ T _A ≤ T _{MAX}
ADC0832/8CIWM	-40°C to +85°C
ADC0834BCN,	
ADC0838BCV,	
ADC0831/2/4/8CCN,	
ADC0838CCV,	
ADC0831/2/4/8CCWM	0°C to +70°C

La scelta è caduta su questo tipo di convertitore, in quanto il microcontrollore studiato presenta un numero di pin di I/O abbastanza esiguo: il PIC16F84 presenta 5 pin di I/O sulla PORTA e 8 pin di I/O sulla PORTB. Per l'interfacciamento dell'AD0831 si richiedono esclusivamente tre pin per il suo interfacciamento con un controllore, contro un maggior numero di pin richiesti per l'interfacciamento di un convertitore AD parallelo che avrebbe occupato quasi tutti i pin di I/O disponibili sul microcontrollore.

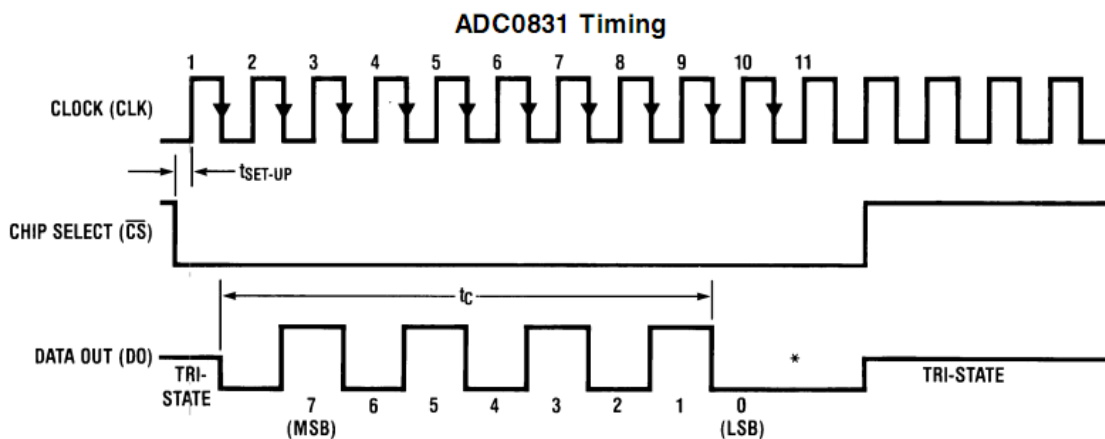
Il collegamento tra convertitore AD e PIC16F84 è mostrato nell'immagine seguente:



V_{IN+} e V_{IN-} sono ingressi differenziali dell'AD0831, per questo V_{IN-} viene posto a massa. Essendo questo convertitore a 8 bit può apprezzare $2^8=256$ diversi livelli di segnale.

$$LSB = \frac{\text{Valore Fondo Scala}}{2^{nbit}} = \frac{5}{256} = 0,0195V$$

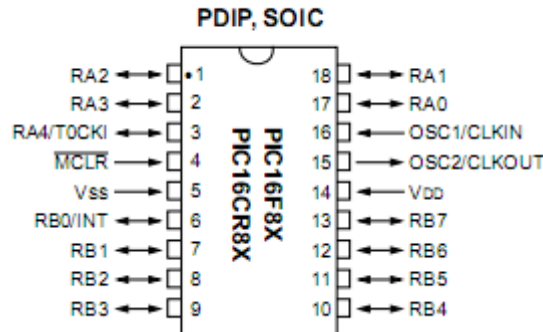
Il μC fornisce all'ADC la seguente temporizzazione (tratta dal datasheet), pilotando opportunamente i pin CLK, \overline{CS} , DO:



I bit prelevati all'uscita del convertitore sono inviati al μC che utilizzando un protocollo seriale RS232, l'invia tramite i moduli RTF-DATA-SAW al PC.

PIC 16F84A:

E' un microcontrollore a 8 bit della Microchip che si presenta in un contenitore a 18 piedini. Di seguito si riporta la piedinatura dell'integrato.

Pin Diagrams

L'integrato richiede una tensione di alimentazione compresa tra 2V e 6V (valore tipico di 5V) da applicare ai terminali indicati con V_{DD} (terminale positivo) e V_{SS} (terminale di massa GND). I pin OSC1/CLKIN e OSC2/CLKOUT devono essere collegati ad un quarzo o un a rete RC, per generare il clock di sistema. La frequenza massima di tale segnale è di 10 MHz (20 MHz per il modello 16F84A).

L'unità centrale di elaborazione CPU è di tipo RISC (Reduced Instruction Set Computer) con un set di istruzioni ridotto a sole 35 istruzioni in grado di elaborare dati a 8 bit con già inclusi i metodi di indirizzamento immediato, diretto ed indiretto. Il modo di operare sfrutta la tecnica denominata pipeline a 2 stati per cui mentre viene eseguita un'istruzione, contemporaneamente viene caricata l'istruzione successiva nel registro delle istruzioni per la decodifica. Ciò si traduce in una riduzione del tempo di esecuzione di ciascuna istruzione.

Il RESET del PIC si realizza portando la linea MCLR (pin 4) al livello basso. Tale linea deve essere tenuta, normalmente, al livello alto V_{DD} .

L'integrato dispone di 13 linee bidirezionali di I/O; 5 indicate con RA0...RA4 costituiscono la PORTA A e 8 con RB0....RB7 la PORTA B. Ciascuna linea può essere programmata come Input o come Output ed è in grado di assorbire (sink current) fino a 25mA ed erogarne (source current) fino a 20mA.

Si è detto che il PIC 16F84 dispone di 13 linee di I/O programmabili individualmente come Input o come Output denominate Porta A (5 linee) e PORTA B (8 linee). Su ciascun pin della Porta B è possibile inserire a software una resistenza di pull-up.

Per questo progetto si è utilizzato un quarzo con una frequenza pari a 1Mhz.

Il programma di gestione del sistema è stato scritto in CCSC (C per PIC) attraverso il compilatore "PIC C Compiler". Essendo il C un linguaggio ad alto livello la programmazione risulta molto più semplice rispetto all'Assembly ed in più già di suo il compilatore implementa delle librerie per la trasmissione seriale e per la gestione dell'LCD.

I Microcontrollori, infatti, sono programmati utilizzando il linguaggio assembler, questo è un linguaggio costituito da diverse istruzioni mnemoniche ed è specifico per ogni microcontrollore; l'assembler per un tipo di microcontrollore non può essere utilizzato per un altro microcontrollore.

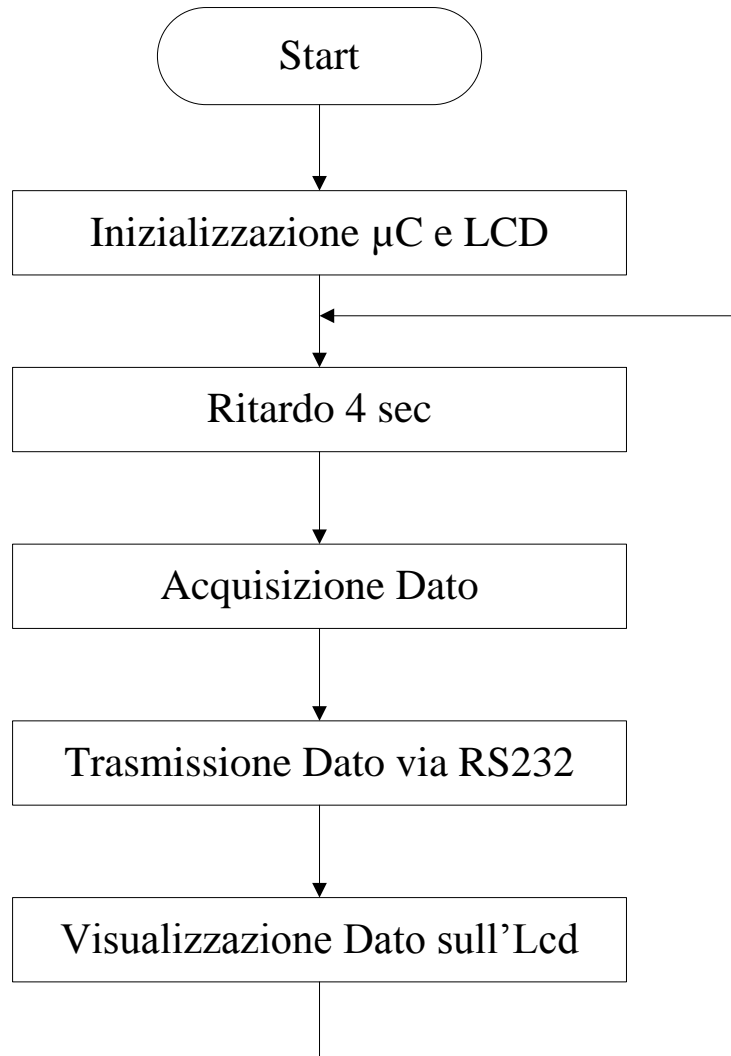
In genere i programmi assembler producono codice che occupa meno risorse rispetto al codice generato con un altro linguaggio e spesso i tempi di esecuzione sono inferiori rispetto ai tempi di esecuzione di un codice generato con altro linguaggio.

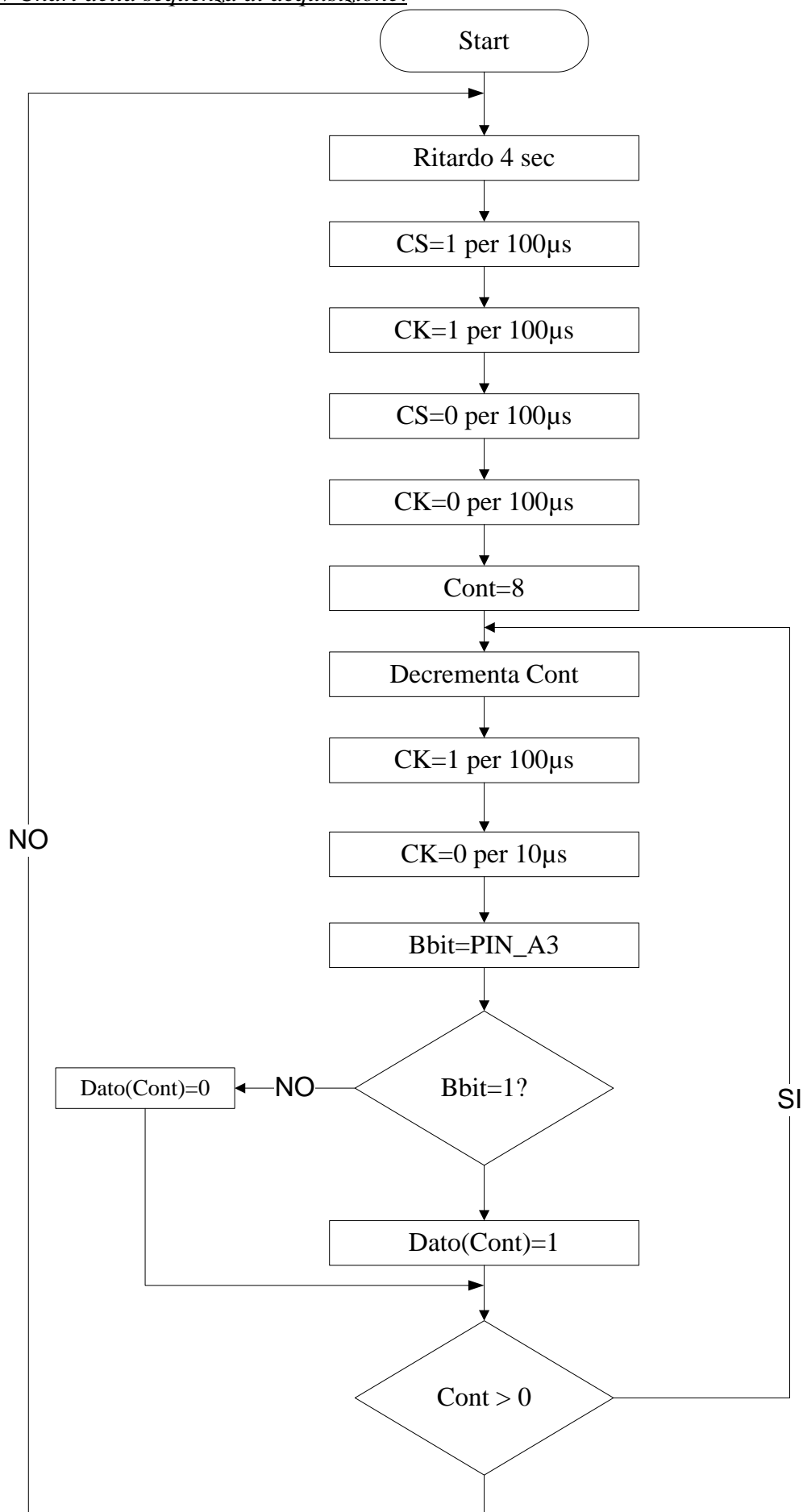
I microcontrollori possono essere programmati anche utilizzando un **linguaggio ad alto livello**, ad esempio è possibile utilizzare compilatori in **BASIC**, in **PASCAL**, in **JAVA** o in linguaggio **C**, la maggior parte di questi compilatori generano codice macchina nativo che può direttamente essere caricato nella memoria del microcontrollore mediante opportuno "programmatore".

In genere questi linguaggi hanno delle istruzioni molto più vicine al nostro normale linguaggio e quindi sono più facilmente "ricordabili" (da questo vengono anche definiti linguaggio ad alto

livello). **In questa ottica il linguaggio C è uno dei più utilizzati nella programmazione dei microcontrollori.**

Vediamo adesso qual è l'algoritmo che dovrà utilizzare il nostro PIC16F84. Di seguito è riportato il diagramma di flusso che lo descrive:



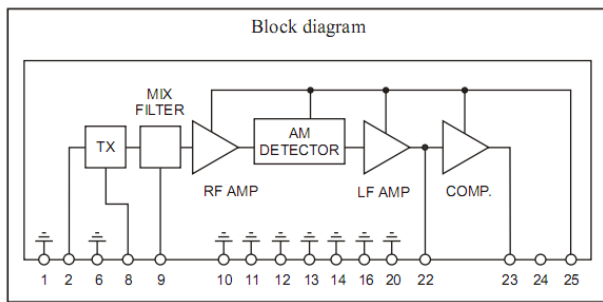
Flow Chart della sequenza di acquisizione:

RTF-DATA-SAW:

Per poter trasmettere un segnale attraverso l'etere è necessario eseguire una modulazione in quanto l'etere rappresenta un canale di tipo passa-banda. Il motivo di ciò risiede nel fatto che per poter realizzare le antenne di trasmissione è necessario costruirle con altezze opportune che diminuiscono all'aumentare della frequenza del segnale da trasmettere.

Il nostro ricetrasmittitore RTF-DATA-SAW esegue, infatti, una modulazione di tipo on-off, cioè un modulazione digitale di ampiezza tipo OOK. Una modulazione digitale si ottiene modulando i bit generati da una sorgente (nel nostro caso i bit che fuoriescono dalla PortA del PIC) con una portante sinusoidale. Dal mixaggio (moltiplicazione dei due segnali) ne deriva un segnale che presenta la portante in corrispondenza dell'1 logico della sorgente e nel non trasmettere nulla in corrispondenza di uno 0 logico.

L'RTF-DATA-SAW è un modulo integrato dell'Aurel che ha il vantaggio poter funzionare da trasmettitore oppure da ricevitore a seconda della configurazione che si dà ai suoi pin.. Nel nostro caso un modulo è configurato per la ricezione mentre l'altro per la ricezione. Questo modulo può tranquillamente trasmettere dati seriali fino ad una velocità massima di 4800 baud/s in questo caso i dati sul ricevitore vanno prelevati dal pin 23 Rx Digital Out. Nel nostro caso la velocità scelta è di 1200 baud/s.

**Pin-out**

1) Ground	12) Ground
2) TX data input	13) Ground
0V=Tx Off	14) Ground
5V= Tx continuous On	16) Ground
6) Ground	20) Ground
8) Tx +5V supply	22) RX analog out
9) Antenna	23) RX digital out
10) Ground	24) N.U.
11) Ground	25) RX +5V supply

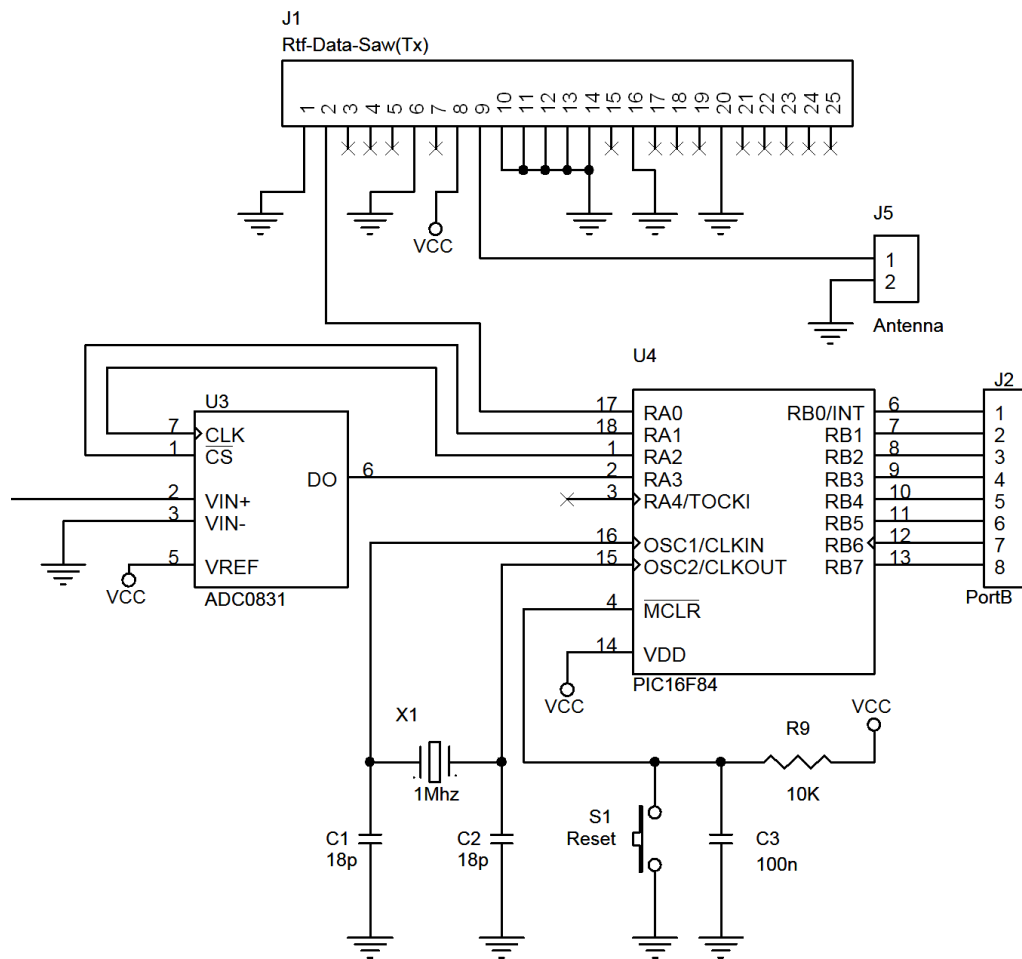
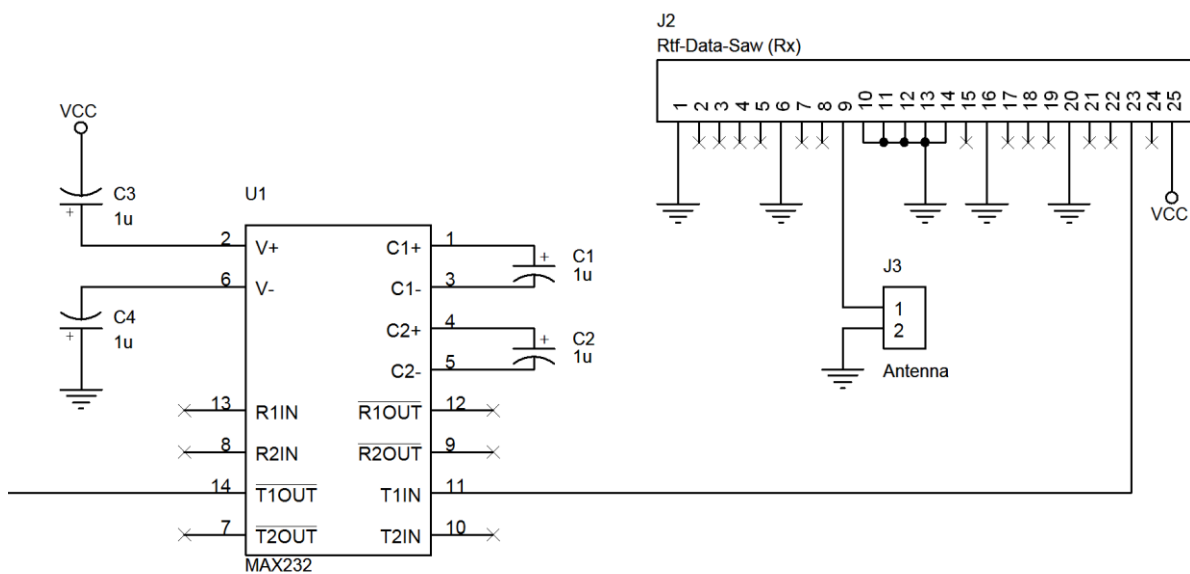
Di seguito riportiamo le caratteristiche dei moduli di trasmissione e ricezione AUREL utilizzati nel progetto:

Caratteristiche Tecniche:

- Realizzazione in circuito ibrido su allumina ad elevata miniaturizzazione ;
- Frequenza disponibile : 433,92 MHz ;
- Sezione Tx con risuonatore SAW ;
- Potenza TX : 8 mW (9 dBm \pm 2 dB) su carico da 50 Ω
- Banda passante BF : onda quadra 5 KHz (Mod. RTF)
onda quadra 3 KHz (Mod. RTL)
- Sezione Rx di tipo supereattivo ;
- Sensibilità RF misurata con segnale On-Off in ingresso :
migliore di 7 μ V (-90 dBm) per il mod. **RTF-DATA-SAW**,
migliore di 2,24 μ V (-100 dBm) per il mod. **RTL-DATA-SAW** ;
- Tempo di commutazione Tx-Rx : migliore di 100 ms, con Rx sempre On ;
- Formato "in line" con dimensioni: 63,5 x 17,9 x 5 mm. Pin passo 2,54 mm ;

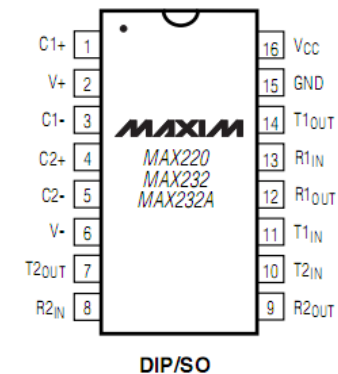
Assorbimento @ +5V

- Sezione Tx 4.5 mA con onda quadra in ingresso ;
- Sezione Rx 2.5 mA ;
- In stand-by : consumo nullo (sia Tx che Rx) ;

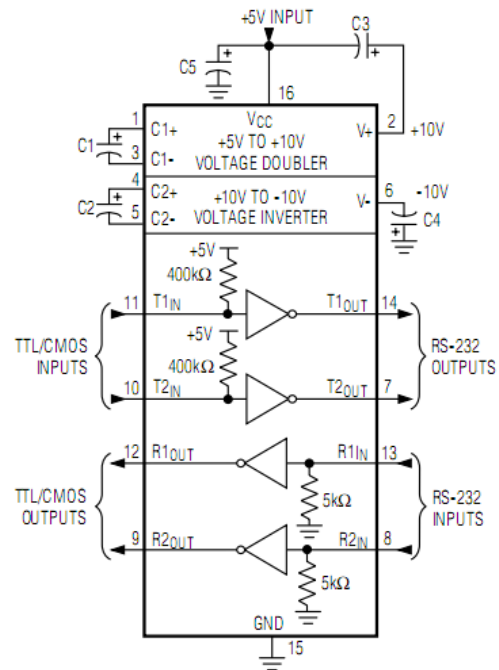
Configurazione Trasmettitore:Configurazione Ricevitore:

Line Driver (MAX232):

TOP VIEW



CAPACITANCE (μF)					
DEVICE	C1	C2	C3	C4	C5
MAX220	4.7	4.7	10	10	4.7
MAX232	1.0	1.0	1.0	1.0	1.0
MAX232A	0.1	0.1	0.1	0.1	0.1



La porta seriale al contrario di quella parallela non accetta livelli logici di tipo TTL in quanto lavora con il principio della logica negata: l'uno logico corrisponde ad una tensione inferiore a -3V mentre lo zero logico corrisponde ad una tensione maggiore di +3V. I valori -3 e +3 possono estendersi rispettivamente fino ad arrivare a -15V e +15V. Per adattare questi livelli esistono due tipi di circuiti Line Driver e Line Receiver. I Line Driver adattano i livelli logici TTL a livelli compatibili con l'interfaccia seriale mentre i Line Receiver compiono l'operazione contraria.

Nel nostro caso viene utilizzato l'integrato MAX232 della Maxim, questo integrato internamente contiene circuiti che si occupano di elevare la tensione da +5V a $\pm 10\text{V}$, due line driver e due line receiver. Per il corretto funzionamento di esso bisogna collegare opportunamente 5 condensatori elettrolitici da $1\mu\text{F}$.

Porta seriale (RS232):

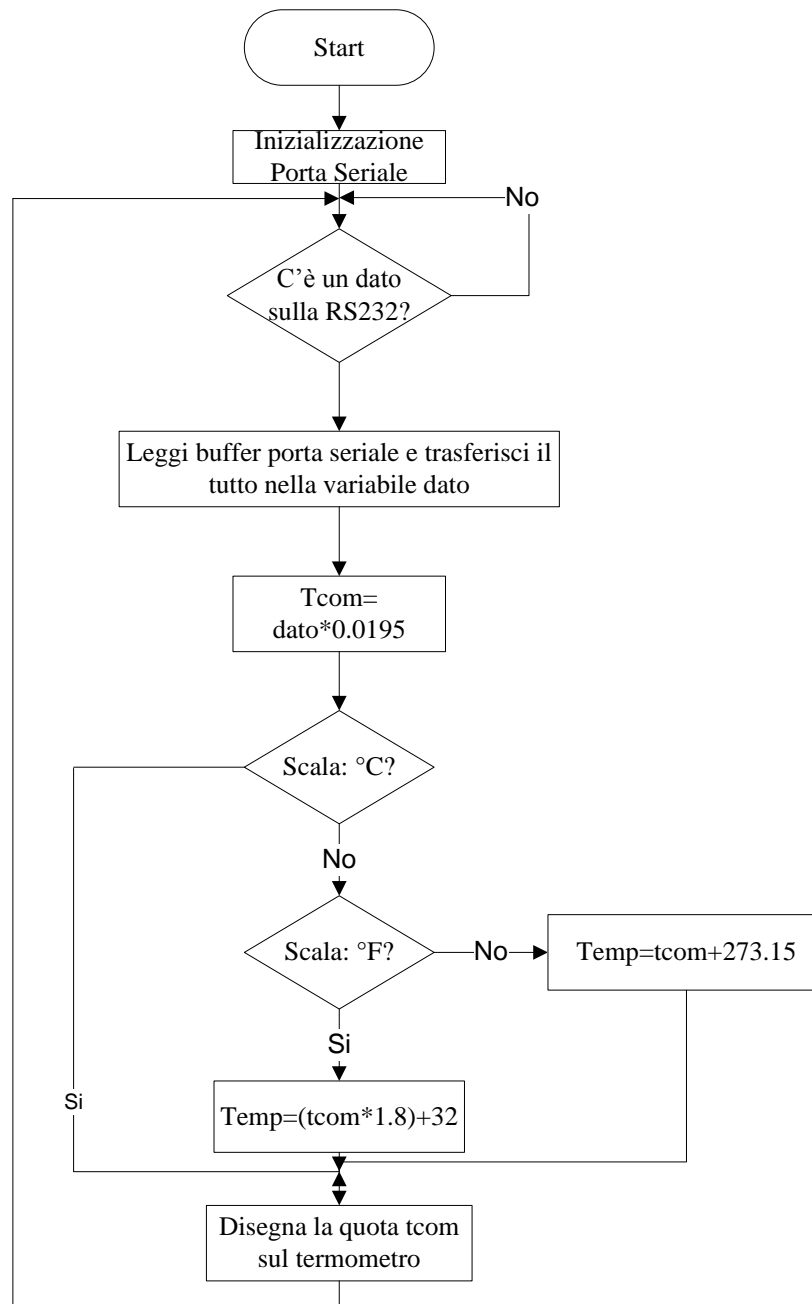
La porta seriale di un PC è riconoscibile dal connettore maschio a 9 poli (DB9) sul retro del case. Tale porta soddisfa lo standard RS232 che prevede una velocità di trasmissione fino a 20kbit/sec su distanze fino a 15m.

Pin Out porta seriale a 9 pin:

Pin	Sigla	Segnale
1	CD	Data Carrier Detect
2	RXD	Received Data
3	TXD	Transmitted Data
4	DTR	Data Terminal Ready
5	GND	Signal Ground
6	DSR	Data Set Ready
7	RTS	Request To Send
8	CTS	Clear To Send
9	RI	Ring Indicator

Il tipo di trasmissione che è possibile eseguire con la porta seriale può essere di tipo sincrono e asincrono.

Il programma per l'acquisizione sul PC della temperatura proveniente dalla scheda posta a distanza rispetto al computer dovrà eseguire i seguenti passi:



Modulo LCD:

Si è scelto di utilizzare un LCD 2x16 HD44780 compatibile, per il quale esistono una miriade di librerie in C quindi interfacciarlo diventa semplicissimo.

Questo dispositivo elettronico ha il nome derivante dal fatto che l'informazione è affidata allo stato di numerosi piccoli punti (pixel), organizzati su righe e colonne (Matrice) successive.

Il visualizzatore può essere:

- **Alfanumerico:** in questo caso i punti sono predisposti in gruppi di 5*10 (5*8 o anche 5*7) pixel; accendendo opportunamente i puntini ogni gruppo può assumere l'aspetto di un carattere; di solito questi visualizzatori mostrano una o più righe di caratteri, fino a 40 caratteri per riga.
- **Grafico:** su questi display non si distinguono le righe di caratteri tipiche degli alfanumerici: l'area dei pixel è organizzata in modo omogeneo e continuo, prestandosi in modo egregio a realizzare un'immagine più o meno grande, a seconda della risoluzione.

I metodi di pilotaggio utilizzati per l'accensione dei pixel sono sostanzialmente 2:

- **a matrice passiva:** è il più diffuso e il più economico si basa sull'applicazione di campi elettrici da applicare su un reticolo di elettrodi di riga e di colonna, trasparenti, posti ai lati del cristallo liquido; quasi tutti i dispositivi elettronici (orologi digitali, calcolatrici, i primi portatili...) usano questa tecnica per accendere i pixel.
- **a matrice attiva:** ogni singolo pixel è associato ad un transistor (Thin Film Transistor), per cui si parla di display TFT-LCD o più sinteticamente di display TFT; la visualizzazione su di esso è splendida (nel vero senso della parola...) e il controllo dei pixel è veloce e accurato; naturalmente gli elettrodi del primo tipo non sono necessari ma la presenza di un componente attivo per ogni pixel porta i costi a livelli piuttosto alti ancora oggi.

I display a cristalli liquidi (LCD, Liquid Cristal Display) sono visualizzatori che, a differenza dei tubi a raggi catodici (CRT, Cathode Ray Tubes) che sono una sorgente luminosa, riflettono o assorbono la luce prodotta da una sorgente luminosa esterna o da un sistema di retroilluminazione.

Certamente il più importante tra questi è il display con controller HD44780, un componente programmabile chiamato ad un duplice compito:

- dispone della struttura necessaria per gestire fino a 50 (10*5) pixel per ciascuno dei possibili caratteri alfanumerici (16, 20, 24, 32, 40, ...) del display, organizzati su una o più righe
- assicura l'interfaccia intelligente tra il visualizzatore e il microcontrollore chiamato a gestirlo.

Piedinatura del modulo LCD

Pin Lcd	Sigla	Descrizione
1	Vss	Ground
2	Vdd	Supply Voltage for logic
3	Vo	Operating voltage for LCD
4	RS	H: DATA, L: Instruction code
5	R/W	H:Read(MPU→Module) L: Write(MPU→Module)
6	E	Chip enable signal
7	DB0	Data bit 0
8	DB1	Data bit 1
9	DB2	Data bit 2
10	DB3	Data bit 3
11	DB4	Data bit 4
12	DB5	Data bit 5
13	DB6	Data bit 6
14	DB7	Data bit 7

I pin 1 (Vss) e 2 (Vcc) servono per fornire alimentazione (5 volt); di solito l'assorbimento è contenuto, dell'ordine di pochi mA.

Il pin 3 (Vo) è un ingresso analogico al quale si deve fornire una tensione compresa tra 0 e 5V al fine di regolare il contrasto del display, cioè la capacità di rendere più o meno scuri i pixel attivi, per rendere migliore la visualizzazione, in funzione della luminosità dell'ambiente, del display; di solito la regolazione si ottiene con un potenziometro da 10k.

Il pin 4 (RS), detto Register Selector, indica la natura dell'informazione presente sul bus dati: se viene forzato a 0 il byte in ingresso sarà interpretato come un comando da eseguire (istruzione), mentre con un 1 sarà ritenuto dato da interpretare.

Il pin 5 (R/W) specifica la direzione dei bytes sul bus: se vale 0 si sta scrivendo nella memoria interna LCD, mentre con un 1 questa memoria è sottoposta a lettura.

Il pin 6 (E) è il segnale che abilita il dispositivo: quando è a livello alto sincronizza la lettura del dato o del comando predisposto sul bus dati. L'Enable è il vero e proprio segnale di sincronismo, da forzare a 1 quando il dispositivo è pronto a gestire un dato o un comando predisposto sul bus dati (in funzione del valore del busy flag: finché il suo valore è 1 il controller è occupato a gestire i trasferimenti interni e non è in grado di accettare l'istruzione successiva).

I pin dal numero 7 al numero 14 (D0-D7) rappresentano il Data Bus su cui avviene la trasmissione o ricezione dati dal display, in base all'operazione che si sta svolgendo.

Talvolta sono presenti 2 pin aggiuntivi (pin 15 e 16), mediante i quali è possibile alimentare l'eventuale luce di retroilluminazione.

Di seguito si riporta la tabella dei collegamenti per collegare il display al PIC

Pin Lcd	Sigla	Descrizione	Pin Pic	Sigla
1	Vss	Ground	-	-
2	Vdd	Supply Voltage for logic	-	-
3	Vo	Operating voltage for LCD	-	-
4	RS	H: DATA, L: Instruction code	7	B1
5	R/W	H:Read(MPU→Module) L: Write(MPU→Module)	8	B2
6	E	Chip enable signal	6	B0
7	DB0	Data bit 0	-	-
8	DB1	Data bit 1	-	-
9	DB2	Data bit 2	-	-
10	DB3	Data bit 3	-	-
11	DB4	Data bit 4	10	RB4
12	DB5	Data bit 5	11	RB5
13	DB6	Data bit 6	12	RB6
14	DB7	Data bit 7	13	RB7
15	A	Anodo Retroilluminazione	-	-
16	K	Catodo Retroilluminazione	-	-

REALIZZAZIONE PRATICA**LISTA COMPONENTI:**Scheda Trasmettitore:

- 2 Condensatori da 18pF
- 1 Condensatore da 100 pF
- 1 Diodo Zener 3,9 V
- 1 LM335
- 1 Resistenza da 1 K Ω
- 1 Resistenza da 2,7 K Ω
- 1 Resistenza da 2 K Ω
- 1 Resistenza da 12 K Ω
- 1 Trimmer da 500 Ω
- 1 Trimmer da 10 K Ω
- 1 Trimmer da 5 K Ω
- 1 Quarzo da 1Mhz
- 1 Pulsante
- 1 μ A741
- 1 AD620
- 1 ADC0831
- 1 PIC16F84A
- 1 Scheda Rft-Data-Saw
- 1 Antenna per i 433 Mhz
- 1 Display LCD 2x16 HD44780 compatibile

Scheda Trasmettitore:

- 5 Condensatori Elettrolitici da 1 μ F
- 1 MAX232
- 1 Scheda Rft-Data-Saw
- 1 Connettore Seriale a 9 pin femmina
- 1 Antenna per i 433 Mhz

STRUMENTI UTILIZZATI:

- Oscilloscopio
- Multimetro
- Alimentatore
- Generatore di funzioni
- Termometro digitale

MONTAGGIO E COLLAUDO:

Si consiglia di eseguire un montaggio “blocco per blocco” in modo da facilitare l’individuazione e l’opportuna correzione di eventuali errori

Si è iniziato con il montaggio dei blocchi a monte costituiti dal sensore e dal circuito di condizionamento. Questa parte è stata collaudata col solo uso di un multimetro digitale. Si è verificato innanzitutto che al pin invertente dell’AD620 ci fosse una tensione fissa di 2,73V, in caso contrario si agisce sul trimmer R₄ per fissarne il valore.

Al pin non invertente dell'AD620 si verifica che il valore di tensione fornito dal sensore sia quello corrispondente alla temperatura in °K misurata da un termometro ambientale moltiplicata per 10mV, in caso contrario si agisce sul trimmer R₆ di calibrazione dell'LM335.

Verificate le tensioni all'ingresso dell'AD620 si verifica che in uscita cvi sia il corrispondente valore calcolato pari a $V_{out} = \left(\frac{49,4K}{R_g} + 1\right) * V_{in} = \left(\frac{49,4K}{12350} + 1\right) * (V_{LM335} - 2,73)$;

nel caso in cui il valore calcolato non venga ritrovato all'uscita dell'AD620 si agisce sul trimmer R₈.

Una volta verificata questa prima parte si procede con il montaggio dell'ADC al circuito precedente e si verifica mediante un oscilloscopio digitale utilizzando come trigger esterno il segnale CS dell'AD0831 che effettivamente il dispositivo trasmette la sequenza di bit digitali corrispondenti al valore misurato.

Successivamente si passa al montaggio del PIC, dell'LCD, e del MAX232.

Si esegue allora una prima verifica dell'avvenuta trasmissione del dato al PC mediante l'utilizzo di Hyperterminal (applicazione di Windows) e utilizzando esclusivamente l'ADC e il PIC scollegati dal sensore.

Per l'utilizzo di HyperTerminal si è proceduto nel seguente modo:

- Si è aperta l'applicazione HyperTerminal che si trova sotto Accessori→Comunicazioni e si è seguita la procedura sotto riportata:
 - Si dà un nome alla connessione e si preme ok
 - Si seleziona la Com a cui è stata connessa la scheda rx (in genere è com1) e si preme ok.
 - Si imposta la porta come segue:
 - Bit per secondo: 1200
 - Bit di dati: 8
 - Parità: Nessuno
 - Bit di stop: 1
 - Controllo di flusso: Nessuno

Una volta caricato il programma di gestione del termometro nel PIC, e fissato l'ingresso del convertitore AD a 5V, nel display LCD si dovrebbe leggere una temperatura di circa 100 °C .

Nella videata di HyperTerminal si dovrebbero visualizzare dei simboli, per verificare l'esattezza di questi aprire la Mappa Caratteri (Accessori→Utilità di sistema), in questa selezionare il carattere Terminal e selezionare il carattere ricevuto da HyperTerminal. Una volta selezionato il carattere in basso a sinistra viene visualizzato il codice carattere in esadecimale, per verificare la corrispondenza tra esso e la temperatura, convertirlo in decimale e moltiplicarlo per 0.39. Il risultato dovrebbe venire intorno a 100.

Se fino a questo punto il tutto funziona si lancia il programma in visual basic che se corretto dovrebbe visualizzare una temperatura di circa 100 ° C.

Infine, collegando all'ingresso dell'AD0831 il circuito costituito dal sensore e dal condizionamento si visualizza sul display LCD e sul programma in visual basic in esecuzione sul PC la temperatura ambiente.

Una volta verificato che la trasmissione con la porta seriale funziona bisogna verificare il funzionamento dei moduli Aurel.

Una prima verifica è stata effettuata collegando all'ingresso del modulo collegato come trasmettitore un'onda quadra (tramite generatore di funzioni) e verificando con l'oscilloscopio che in uscita al modulo ricevitore ci fosse la stessa onda quadra.

Quando la verifica precedente ha dato buon esito si è provveduto a finire il montaggio dell'intero schema e verificare le temperature visualizzate sul display LCD e sul monitor del PC.

DIFFICOLTA' INCONTRATE NELLA REALIZZAZIONE DEL PROGETTO:

Numerosi sono stati i problemi che si sono verificati durante la realizzazione di questo progetto. Inizialmente l'Adc veniva pilotato da una rete logica e il clock veniva fornito da un generatore di onda quadra formato da un 74HC14. Fornendo il clock tramite la not triggerata non si è riuscito a far funzionare la trasmissione in quanto non c'era sincronismo tra il clock della seriale e quello fornito all'adc.

Riguardo il programma in Visual Basic inizialmente la temperatura veniva acquisita sfruttando la proprietà RThreshold che scatena l'evento OnComm, successivamente si è notato che c'erano problemi di acquisizione via radio sempre per il motivo che riguarda i sincronismi. Per risolvere questo problema si è modificato il programma facendo acquisire la temperatura tramite un timer impostato con un intervallo di 1 sec.

SCHEMI ELETTRICI E FOTO DEL PROGETTO

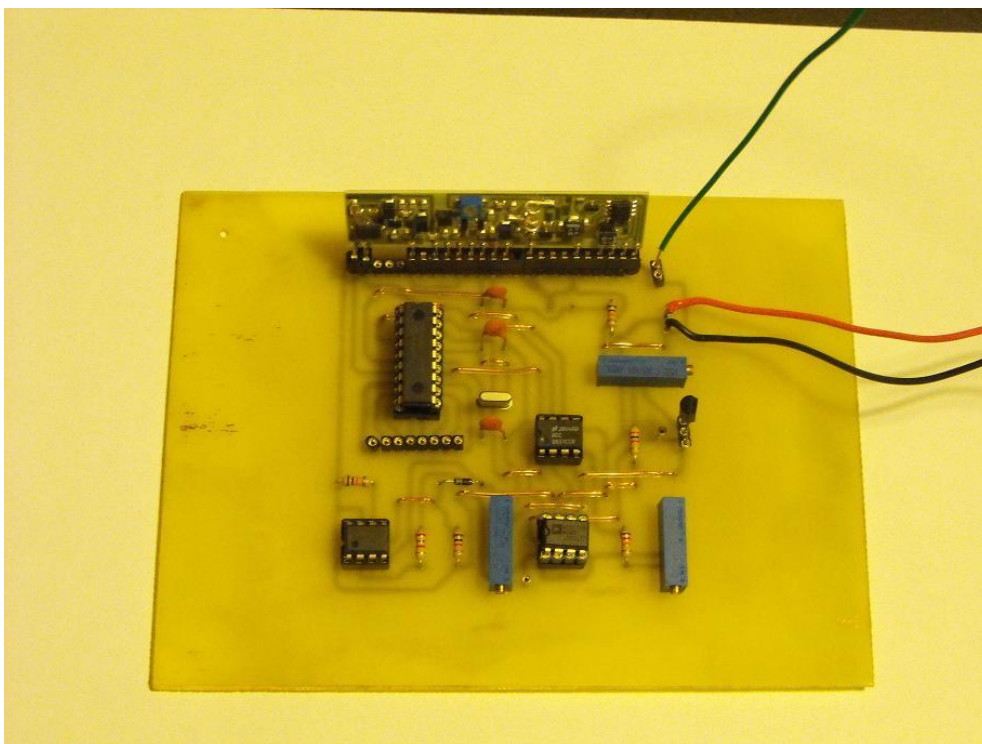


Figura 1 Lato Top della scheda di trasmissione

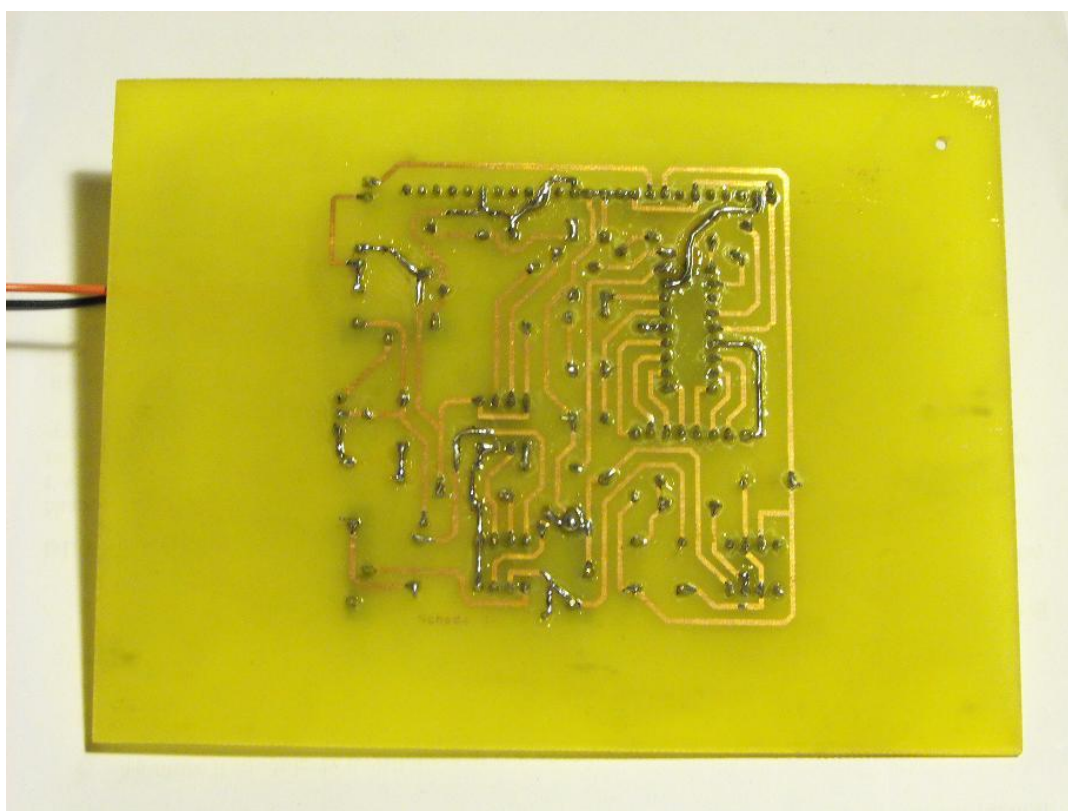


Figura 2 Lato Bottom della scheda di trasmissione

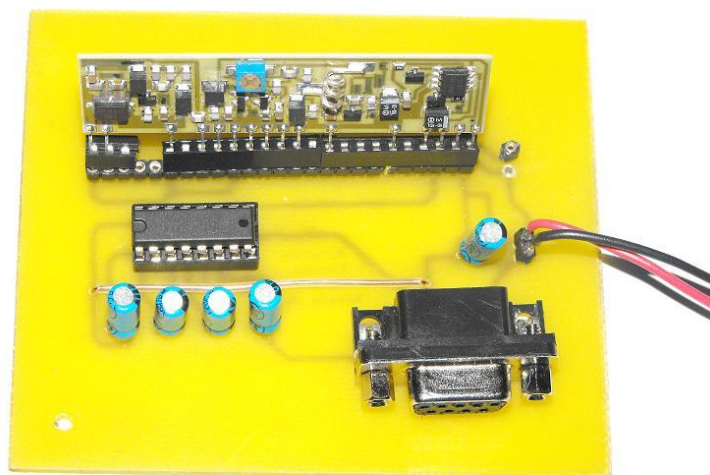


Figura 3 Lato Top della scheda di ricezione

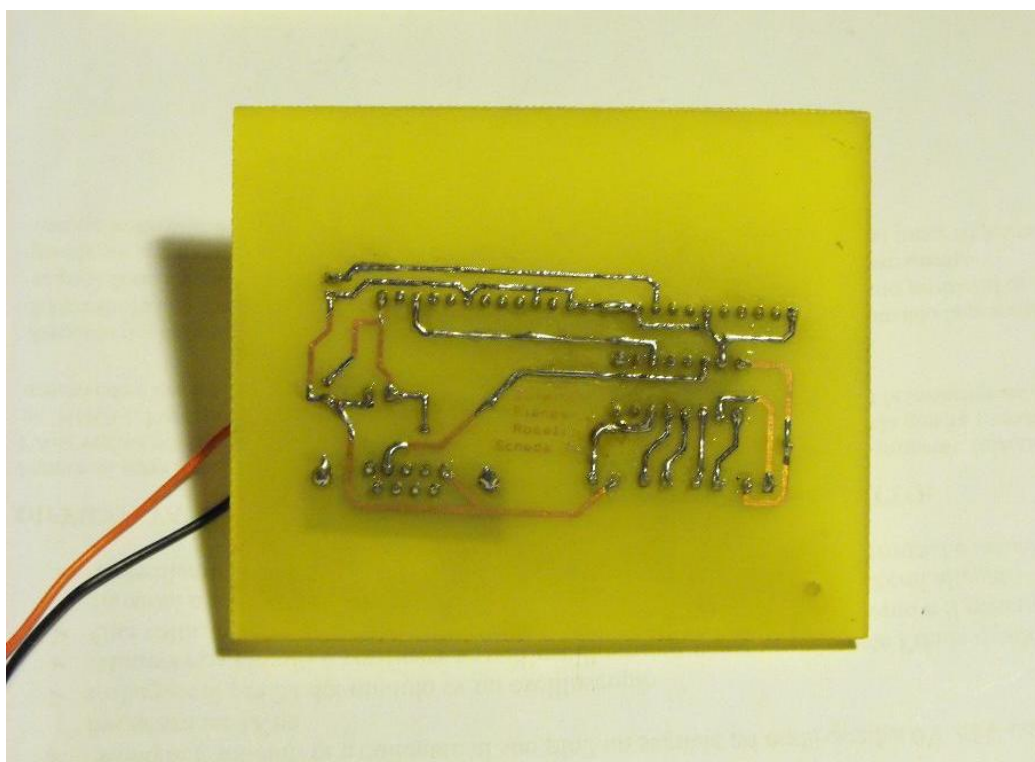
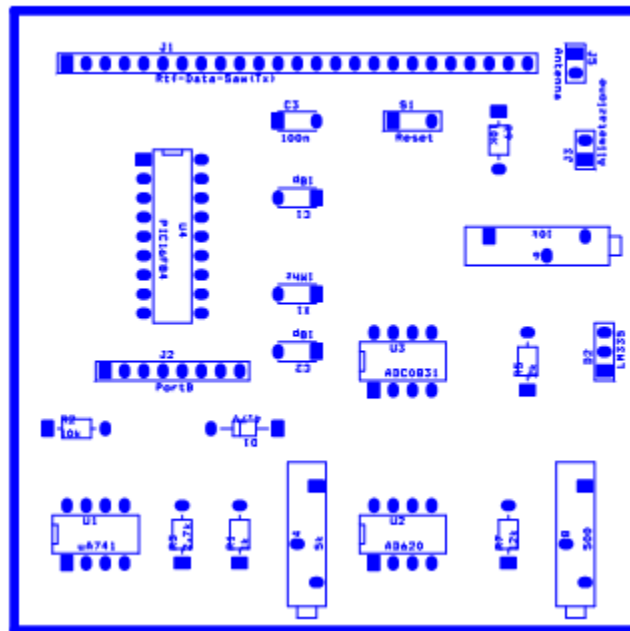
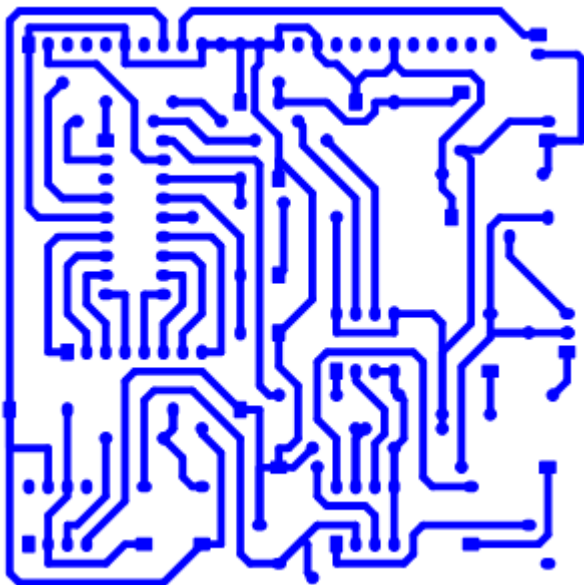


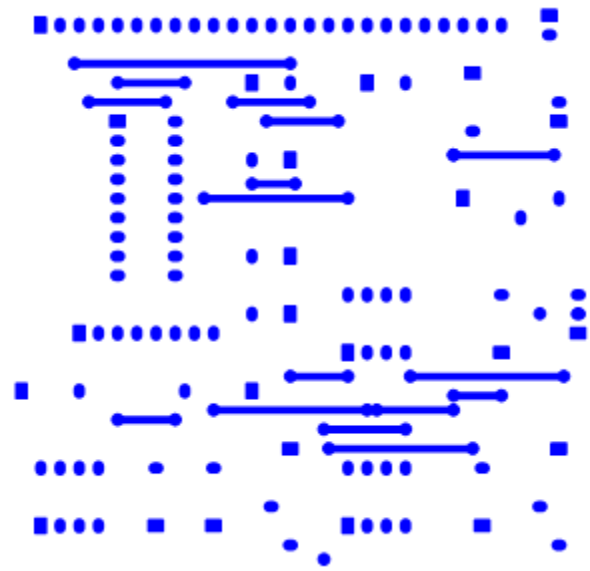
Figura 4 Lato Bottom della scheda di ricezione



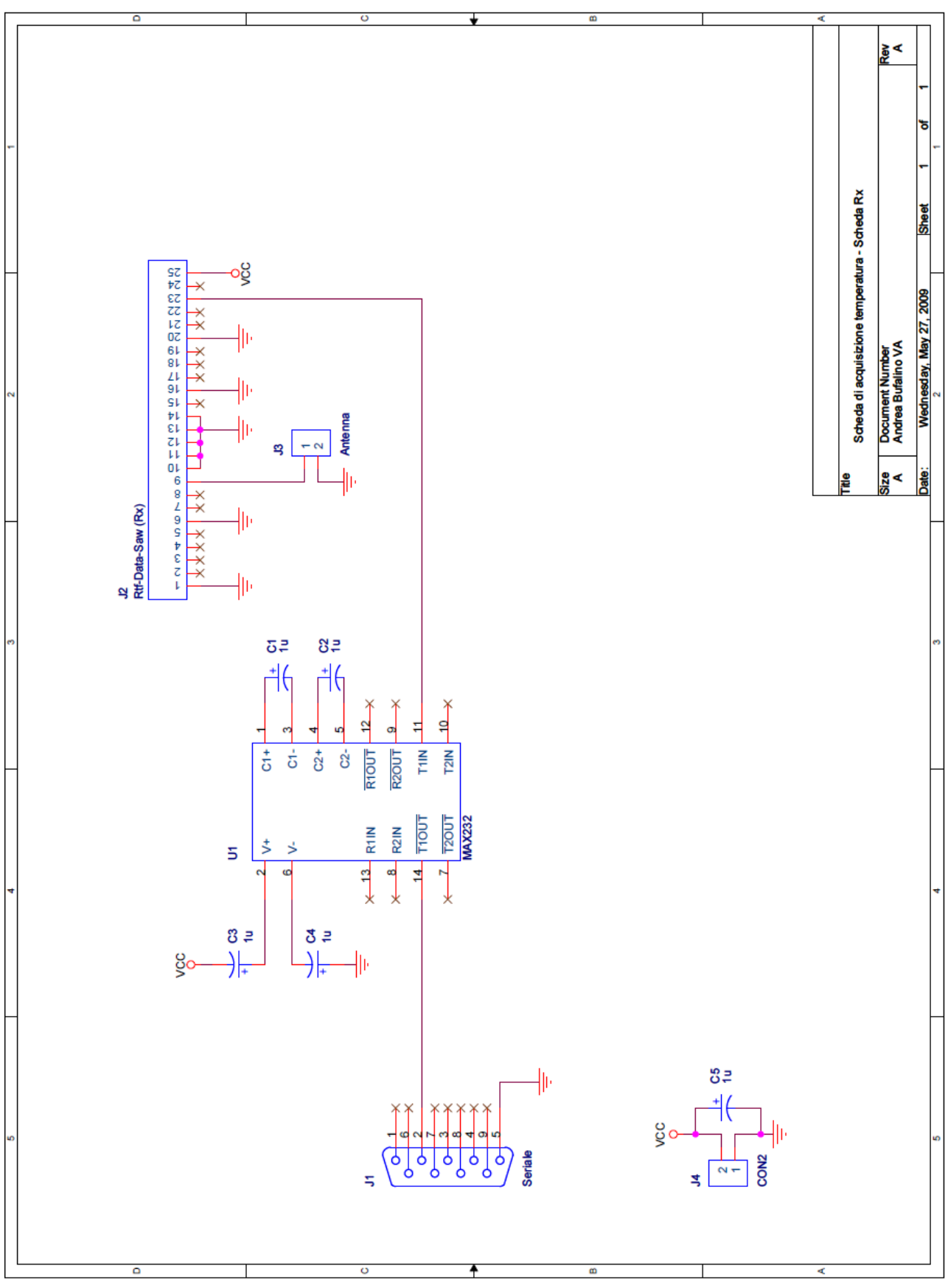
Layer Assembly Top

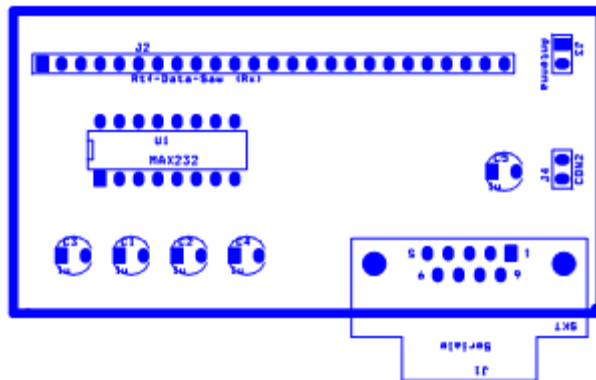


Bottom Layer

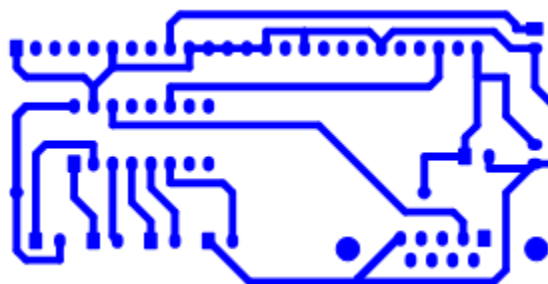


Jumper Layer





Layer Assembly Top



Bottom Layer



Jumper Layer

APPENDICE

SOFTWARE DEL PROGETTO

Programma in C per il PIC:

File: rxtx.c

```
#include "rxtx.h"
#include "LCD420.C"
// Tx A0
#define cs pin_A1
#define ck pin_A2
#define d pin_A3
void main()
{
    char cont;
    char dato;
    float temp;
    int8 k;
    short bbit;
    dato=0;
    setup_timer_0(RTCC_INTERNAL|RTCC_DIV_1);
    lcd_init();
    lcd_putc ("Inizializzazione");
    while(1)
    {
        delay_ms(4000);
        lcd_putc ("f");
        for (k=20;k>0;--k)
        {
            output_high(cs);
            delay_us(100);
            output_high(ck);
            delay_us(100);
            output_low(cs);
            delay_us(100);
            output_low(ck); // bit start
            delay_us(100);
            for(cont=8;cont>0;--cont)
            {
                output_high(ck);
                delay_us(100);
                output_low(ck);
                delay_us(10);
                bbit=input(PIN_A3);
                if (bbit==1)
                    bit_set(dato,(cont));
                else
                    bit_clear(dato,(cont));
            }
            putc(dato);
        }
        temp = ((dato*0.0195)*20);
        lcd_putc ("Temp. Ambiente: ");
        lcd_gotoxy(1,2);
```

```
printf(lcd_putc,"%3.1f",temp);  
lcd_putc(0b11011111);  
lcd_putc("C");  
}  
  
}
```

File: rxtx.h

```
#include <16F84A.h>  
  
#FUSES NOWDT           //No Watch Dog Timer  
#FUSES XT              //Crystal osc <= 4mhz  
#FUSES NOPUT          //No Power Up Timer  
#FUSES NOPROTECT      //Code not protected from reading  
  
#use delay(clock=1000000)  
#use rs232(baud=1200,parity=N,xmit=PIN_A0,rcv=PIN_A4,bits=8)
```

Programma Visual Basic:

```
Dim tcom As Single 'da seriale
Dim temp As Single 'converita
Dim cont As Integer 'Contatore
Dim tref As Single
Dim vtemp(1 To 16) As Single
Dim app As Single
Dim scala As String
Dim f As Integer
Dim Rx$
```

```
Private Sub Form_Load()
Picture1.Scale (0, 100)-(2, 0)
MSComm1.CommPort = 1
MSComm1.Settings = "1200,n,8,1"
MSComm1.PortOpen = True
cont = 1
End Sub
```

```
Private Sub optc_Click()
Picture1.Scale (0, 100)-(2, 0)
scala = " °C"
f = 0
temp = tcom
For k = f To temp Step 0.2
Picture1.Line (0, k)-(2, k)
Next k
Lbl1.Caption = temp & scala
End Sub
```

```
Private Sub optf_Click()
Picture1.Scale (0, 212)-(2, 32)
scala = " °F"
f = 32
temp = (tcom * 1.8) + 32
For k = f To temp Step 0.2
Picture1.Line (0, k)-(2, k)
Next k
Lbl1.Caption = temp & scala
End Sub
```

```
Private Sub optk_Click()
Picture1.Scale (0, 373.15)-(2, 273.15)
scala = " °K"
f = 273.15
temp = tcom + 273.15
For k = f To temp Step 0.2
Picture1.Line (0, k)-(2, k)
Next k
Lbl1.Caption = temp & scala
End Sub
```

```
Private Sub Timer1_Timer()  
Rx$ = MSComm1.Input ' Leggo il contenuto del buffer di ricezione (e svuoto .Input)  
If Len(Rx$) Then  
    dato = Mid(Rx$, 3, 1)  
    If dato = "" Then GoTo fine  
    dato = Val(Asc(dato))  
    tcom = dato * 0.0195 * 20  
    tcom = Round(tcom, 1)  
    If cont < 17 Then  
        vtemp(cont) = tcom  
        cont = cont + 1  
    End If  
    If cont = 17 Then  
        For k = 1 To 15  
            For j = (k + 1) To 16  
                If vtemp(k) > vtemp(j) Then  
                    app = vtemp(k)  
                    vtemp(k) = vtemp(j)  
                    vtemp(j) = app  
                End If  
            Next j  
        Next k  
        cont = 18  
    End If  
    tref = vtemp(8)  
    If cont = 18 Then  
        If tcom >= (tref + 2) Or tcom <= (tref - 2) Then  
            GoTo fine  
        Else  
            tref = tcom  
        End If  
    End If  
    Picture1.Cls  
    If optc.Value = True Then  
        scala = " °C"  
        f = 0  
        temp = tcom  
    ElseIf optf.Value = True Then  
        scala = " °F"  
        f = 32  
        temp = (tcom * 1.8) + 32  
    Else  
        scala = " °K"  
        f = 273.15  
        temp = tcom + 273.15  
    End If  
    Lbl1.Caption = temp & scala  
    lblora.Caption = "Ultima acquisizione ore: " & Time  
    For k = f To temp Step 0.2  
        Picture1.Line (0, k)-(2, k)  
    Next k  
End If  
fine: End Sub
```