



MarcoD

ACQUISIZIONE E REGISTRAZIONE TEMPERATURE

12 July 2015

Descrivo un sistema sperimentale di misura e registrazione che ho realizzato per **monitorare le temperature di un termosifone** per prepararmi a verificare il funzionamento del teleriscaldamento, dei termostati e dei contabilizzatori del calore. Mi interessava misurare **tre temperature**: la temperatura della superficie del termosifone, la temperatura in vicinanza della manopola dove verrà montata la testa termostatica e una temperatura a circa 1 cm dal termosifone. Ho utilizzato componenti economici e il tutto mi è costato poche ore di sviluppo, è facile adattarlo ad acquisire altre grandezze elettriche che variano lentamente, lo descrivo sperando sia di spunto per altre applicazioni. Il sistema si compone di **sensori di temperatura**, di una **scheda concentratore dati**, del **programma di acquisizione e trasmissione**, di una **connessione USB**, di un **PC supervisore** con un programma di visualizzazione e memorizzazione dei **dati in formato compatibile Excel**.

Sensori temperatura

Ho adoperato dei sensori LM35DZ con involucro TO-92 a tre piedini a saldare, già disponibili da più di 10 anni. Si alimentano da 4 a 40 V, il campo di funzionamento è 2 - 100 C°. L'uscita è una tensione riferita a massa che vale 10 mV/C°, quindi alla temperatura di 20 C° forniscono una tensione di 200 mV. La precisione dichiarata è migliore di +/- 1 C°. Costano circa 2 Euro l'uno. Connessi a un voltmetro si può già leggere la temperatura in gradi. Per misurare anche una temperatura inferiore allo 0 C°, avrei potuto adoperare dei sensori LM60, con un campo da -25 C° a 125 C°, tensione in uscita di 6,25 mV/C° + 424 mV, precisione +/- 2 C°.

Recentemente sono disponibili i sensori Maxim DS18B20 che forniscono la temperatura già in digitale su una linea seriale. Se ne possono connettere diversi in parallelo sulla stessa linea, ognuno ha il proprio indirizzo e si possono interfacciare con un unico ingresso/uscita logico. Sarebbero i migliori da adoperare per una produzione di serie, ma in piccole quantità costano tre volte di più. Poi la scheda Arduino del concentratore possiede già sei ingressi analogici.

Avrei anche potuto adoperare dei termistori da 2,2 kohm a 25 C°, alimentandoli con una R in serie di valore prossimo che crea un partitore di tensione. Vanno bene per temperature ambiente; costano la metà, ma la caratteristica resistenza temperatura non è dichiarata, la variazione della tensione di partizione con la temperatura non è lineare e avrei dovuto caratterizzarli immersi in un bicchiere d'acqua calda insieme a un termometro di riferimento. Come cavi di connessione ho

usato un cavo telefonico a tre fili, e un cavo con calza schermata a quattro conduttori per linea seriale che avevo disponibili.

Concentratore

Si adoperava una scheda Arduino Uno o 2009. L'alimentazione, la programmazione e la trasmissione dati avvengono tramite il cavo USB. Nello schema elettrico insieme con le connessioni ai sensori sono riportati i nomi dei piedini che compaiono sulla scheda. Come connettori per sezionare la scheda e i sensori dal cablaggio ho usato delle strisce di pin. I condensatori (10 uF 16 V) eliminano i disturbi indotti dalla 50 Hz nei cavi che provocavano piccole fluttuazioni nelle misure.

[schema elettrico monitoraggio temperature jpg.jpg](#)

schema elettrico monitoraggio temperature jpg.jpg

[concentratore e sensore.JPG](#)

concentratore e sensore.JPG

Il programma del concentratore misura tutti i sei ingressi ogni 120 secondi, acquisendo per 30 volte di seguito ogni ingresso e sommando i valori. La tensione di riferimento del convertitore analogico digitale (ADC) è selezionata a 1,1 Volt. Dato che il numero acquisito varia fra 0 e 1023 (1023 per 1,1 V) al massimo la somma vale $1023 * 30 = 30690$ è minore di 32767, quindi è contenuta in una variabile intera della CPU.

La funzione di libreria “Serial.print (To,1);” converte la variabile floating T1 e la scrive sulla linea seriale in caratteri ASCII con una cifra decimale separata da un punto. Il programma trasmette le tre temperature al PC come caratteri ASCII nel seguente formato: “ \$xx.x;xx.x;xx.x” seguito da cr . Il carattere \$ serve come sincronismo per individuare l'inizio del pacchetto.

Il listato del programma sorgente da caricare nell'ambiente Arduino è il seguente:

```
//Acquisizione temperatura da sensori di temperatura LM35DZ e trasmissione
// valori acquisiti a un PC
int nu_ana[6];
int tms, memt;
float T0, T1, T2; //temperature in C°
int Dtempo ; //secondi periodicità esecuzione calcoli
int Dms, N;

void setup()
{
  Serial.begin(19200);
  Dtempo = 10;  Dms = Dtempo * 1000;
  N= 12; //trasmette valori subitito
```

```

analogReference(INTERNAL); // riferimento interno 1,1 V
memt = (millis() & 0xFFFFF) -Dms; //inizializza per trasmettere subito
}

void loop()
{
  tms = millis() & 0xFFFFF; //serve solo una variabile int
  if ((tms - memt) >= Dms ) // periodicità 10 s
  {
    memt = tms;
    N=N+1;
    if (N >= 12) //una volta ogni 12, ossia ogni 2 minuti
    {
      N=0;
      for (int j=0;j<6;j++)
      {
        nu_ana[j] = 0; for (int i=0;i<30;i++)
        {nu_ana[j]= nu_ana[j]+ analogRead(j);}}//legge input
        //sensore 10 mV/C° , da cui: 1 C° ogni 10 mV ossia
        // 1 C° ogni centesimo di volt
        // il max è 1023 per 5V.
        // Si sommano 30 valori; 30690 = 500 centesimi di V
        // 500/30690 = 0,0162919
        //il max è 1023 per 1,1V rif. interno.
        // Si sommano 30 valori; 30690 per 110 centesimi
        // V 110/30690 = 0,0035823
        const float kconv = 0.0035823;//const float kconv = 0.0162919 ;
        T0= (kconv * nu_ana[0]);
        T1= (kconv * nu_ana[1]);
        T2= (kconv * nu_ana[2]);
        Serial.print ("$$");
        Serial.print (T0,1); Serial.print (";");
        Serial.print (T1,1); Serial.print (";");
        Serial.print (T2,1);
        Serial.println ("");
      }
    }
  }
}

```

Programma supervisore

Alla partenza, il programma legge la data e il tempo dall'orologio interno del PC e crea un file nominandolo "aa_mm_gg_hh_mm.txt". Alla ricezione di un pacchetto di dati dal concentratore, attacca ai dati la data e il tempo HHMM presi dall'orologio interno e scrive il tutto sul file in un

formato importabile direttamente in Excel come ad esempio: “ 13/2/2012 11.05;25.3;42,8;46,6 “. Il programma cambia i punti in virgole per adattarsi alla versione italiana di Excel.

Il programma, oltre a memorizzare, per comodità di verifica del funzionamento, visualizza sullo schermo in modo numerico gli ultimi valori di temperatura ricevuti e il tempo di acquisizione, inoltre visualizza il grafico degli ultimi 200 valori.

[Schermata prog registr temperature](#)

Schermata prog registr temperature

Quando l'ora passa per le 12.00 o le 00.00, il file viene chiuso e ne viene aperto un'altro, in questo modo se il PC viene spento accidentalmente non si perdono tutti i dati. Premendo il tasto U il programma chiude il file in corso e termina.

Per visualizzare i valori sullo schermo, ho dovuto riconvertire i caratteri ASCII nel numero rappresentato; per farlo ho utilizzato una struttura logica chiamata “macchina a stati” eseguita a ogni ricezione di un carattere; la descrivo in modo semplificato per la decodifica della sola variabile To:

```

Si definisce una variabile Stato, alla partenza del programma la si pone= 0.
Se Stato = 0, attende il carattere "$" per porre Stato = 1.
Se Stato = 1, quando legge il primo carattere,
    lo si converte nella cifra decimale corrispondente,
    si mette la cifra in una variabile Accumula e si pone Stato=2 .
Se Stato = 2, quando legge un nuovo carattere (é il secondo),
    si moltiplica Accumula per 10,
    si somma ad Accumula la cifra decimale corrispondente a si pone Stato=3.
Se Stato = 3, quando legge un nuovo carattere (é il punto),
    lo si trascura ponendo Stato=4.
Se Stato = 4, quando legge un nuovo carattere (é il terzo e ultimo),
    si moltiplica Accumula per 10, si somma ad Accumula la cifra decimale
    corrispondente, trasferisce il contenuto di Accumula in "T0" e riparte
    ponendo Stato=0.

```

Il programma è scritto nel linguaggio simil C++ dell'ambiente open www.processing.org, renderò disponibile il sorgente a chi me lo chiede.

In alternativa al programma di supervisione, si può adoperare il monitor dell'ambiente Arduino o il programma Hyper Terminal della dotazione di Windows Xp pro (configurare la porta COMxx adoperata a 19200, 8N1).

Elaborazione successive dei dati memorizzati

Riporto come esempio una elaborazione ottenuta in Excel visualizzando i dati importati:

[grafico temperature](#)

grafico temperature

Elaborando ulteriormente i dati memorizzati, si può calcolare una stima del calore prodotto e dei relativi costi.

Nota autore: La qualità dell'impaginazione delle immagini è insoddisfacente, è quanto sono riuscito a fare. Spero che almeno il contenuto dell'articolo sia interessante.

23/2/2012 Rev del 17/7/2015 autore Marco Ducco

Estratto da "<http://www.electroyou.it/mediawiki/index.php?title=UsersPages:Marcod:acquisizione-e-registrazione-temperature>"