zoom in

Una delle caratteristiche specifiche dei microcontrollori è la capacità di acquisizione di grandezze fisiche tramite trasduttori. Se a questa funzione aggiungiamo la capacità di gestire azionamenti otteniamo un sistema di controllo completo. Nel nostro caso ci limitiamo alla semplice acquisizione di una temperatura ma nello stesso tempo vogliamo dare la più ampia visibilità a questi dati. È pertanto scontata la combinazione microcontrollore-PC

1010 e trasmissin



saminiamo dapprima l'applicazione. Il microcontrollore trasmette il valore acquisito tramite la porta seriale al PC che si occupa della visualizzazione, anche in formato grafico, della memorizzazione (come datalogger) e della pubblicazione dei valori su una pagina web. Se al sistema aggiungiamo una webcam, possiamo abbinare la contemporanea pubblicazione dell'immagine. Attraverso il software gratuito (disponibile alla pagina http://www.lundie.ca/fwink/) è possibile infatti configurare la webcam in modo che registri su disco (sulla stessa cartella del file.exe) a intervalli prefissati l'immagine, con il nome webcam.jpg, o anche invii direttamente l'immagine su una paaina web).

Nel nostro caso sarà il programma Delphi a occuparsi di inviare direttamente l'immagine alla pagina web. La scelta del

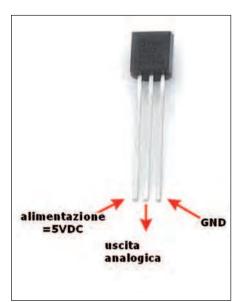


Figura 1: pinout del sensore LM35.

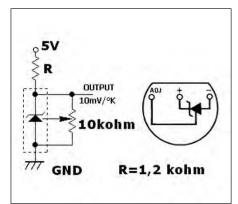


Figura 2: pinout del sensore LM335.

microcontrollore è ricaduta su Arduino perché questo permette l'hardware più semplice ed economico per interfacciare i più comuni trasduttori di temperatura. Inoltre è disponibile per Arduino il software per interfacciare il trasduttore DS18B20. Nulla impedisce, rispettando il formato dei dati stabilito, di sostituire Arduino con altro microcontrollore.

I SENSORI DI TEMPERATURA

Per consentire la più estesa possibilità di scelta abbiamo implementato i seguenti trasduttori:

Lm35. Possiede un'uscita analogica di 10 mv/°C e un'accuratezza di $\pm 1/4$ °C a temperature ambiente. A 0 °C l'uscita è 0 Volt. Il componente può essere connesso direttamente all'alimentazione 5 V di Arduino. L'uscita viene connessa a un suo qualsiasi ingresso analogico.

LM335. È simile al modello LM35, ma l'uscita è 0 V a -273 °C. Il potenziometro va tarato per avere un'uscita di riferimento: ad esempio 2,93 V a 20 °C.

DS18B20. Questo componente è un sensore digitale e non analogico. Ha un campo di misura da -55 °C a +125 °C e un'accuratezza di ±0,5 °C nel campo -10 / +85 °C.

La connessione è effettuata anche in questo caso prelevando l'alimentazione da Arduino mentre l'uscita va a un pin digitale e non analogico. Il software di lettura, reperibile su Internet, si avvale della libreria OneWire.h di Arduino.

NTC 10K. Questo componente è in sostanza una resistenza variabile con la temperatura. La relativa connessione è riportata in **figura 3**.

Termocoppia. Una termocoppia è un sen-

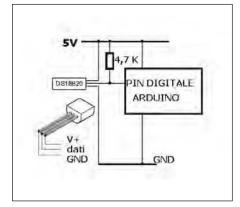


Figura 3: pinout del sensore DS18B20, resistenza è di 4,7 kohm.



sore di temperatura basato sull'effetto termoelettrico. La giunzione T fra i due fili di materiale metallico diverso di cui è costituita la termocoppia genera ai suoi capi una differenza di tensione proporzionale alla differenza fra la temperatura del giunto Tp, che costituisce il punto di misura, e quella dell'altro capo Ta (tipicamente l'ambiente nel quale è inserito lo strumento associato alla termocoppia). La relazione fra tensione generata e la differenza Tp – Ta è, con buona approssimazione, lineare.

Poiché questa tensione è molto piccola deve essere amplificata. Supponiamo

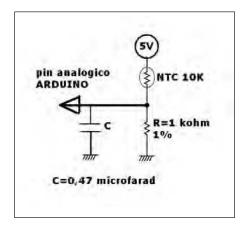


Figura 4: collegamento del NTC.

LISTATO 1

```
//11 febbraio 2012
//arduinoweb
//autore:taraschi_nicola@libero.it
#include <OneWire.h>
int tipo = 1;
void setup(void) {
... .
void getTemp() {
void loop(void) {
analogReference(INTERNAL);
switch (tipo) {
case 1:
sensorValue1 =2000+analogRead(analogInPin1);
temp1=sensorValue1+100000;
Serial.print(temp1);
break;
case 2:
sensorValue1 =4000+analogRead(analogInPin1);
temp1=sensorValue1+100000;
Serial.print(temp1);
break;
case 3:
getTemp(); Serial.print(temp);
break;
case 4:
sensorValue1 =6000+analogRead(analogInPin1);
temp1=sensorValue1+100000;
Serial.print(temp1);
break;
sensorValue1 =8000+analogRead(analogInPin1);
temp1=sensorValue1+100000;
Serial.print(temp1);
break;
delay(1000);
```

ARDUINO, LA PERIFERICA OPEN SOURCE DAL CUORE COMPLETAMENTE PROGRAMMABILE.







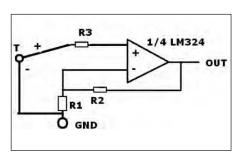


Figura 5: collegamento della termocoppia R1 = R3 = 10 kohm, R2 = 2 Mohm, amplificatore operazione = LM324.

che anche la tensione amplificata sia lineare. Pertanto Tp – Ta= $K \cdot \Delta \cdot V$, dove ΔV rappresenta l'uscita OUT dell'amplificatore che va connessa a un ingresso analogico di Arduino.

IL SOFTWARE SU ARDUINO

Il programma su Arduino legge la temperatura a seconda del tipo di sonda impostata, tramite l'assegnazione della variabile TIPO, secondo la corrispondenza, cui faremo riferimento nel seguito:

- 1. LM35:
- 2. LM335;
- 3. NTC 10K;
- 4. DS18B20:
- 5. termocoppia.

Successivamente esso invia il dato sulla porta seriale. Nel caso della sonda 4, il dato è direttamente il numero reale rappresentante il valore di temperatura ma, per stabilire un formato dell'uscita valido per tutte le sonde utilizzabili, viene aggiunto il numero 800 a questo valore. In questo modo, qualunque sia la temperatura, il dato di uscita è a 6 byte, comprensivo del punto e dei decima-

Negli altri casi il dato è analogico e il convertitore ADC trasforma il valore di tensione in un numero intero fra 0 e 1023. Se è inserita l'istruzione analogReference(valore) il valore 1023 è fatto corrispondere a:

• valore=DEFAULT: il riferimento di tensione della scheda, cioè 5 volt o 3,3 volt;

- valore=INTERNAL: 1,1 volt su ATmega168 o ATmega328 e 2,56 volt su ATmega8;
- valore=EXTERNAL: la tensione applicata al pin AREF.

Per uniformare il formato del dato di uscita, al valore letto viene aggiunto:

- 102000 nel caso 1;
- 104000 nel caso 2;
- 106000 nel caso 4;
- 108000 nel caso 5.

Nel **listato 1** si riportano i punti più significativi del programma su Arduino, mentre il listato completo è disponibile sul sito di *Fare Elettronica*.

IL SOFTWARE SU DELPHI 6

Il programma in Delphi 6 consente:

- la connessione seriale alla scheda Arduino (specificando nelle impostazioni la porta seriale). Questo avviene generalmente attraverso connessione USB. Con l'installazione di Arduino viene creata una porta seriale (nell'ambiente Arduino, voce menu: TOOLS -> SERIAL PORT);
- la memorizzazione su file Excel dei dati completi della data e ora di acquisizione (max 1.000 valori);
- l'accesso alle impostazioni. Per scrivere i dati sulla pagina web occorre avere uno spazio web, generalmente assegnato gratuitamente con una casella di posta e accessibile tramite protocollo FTP. Le prime cinque righe delle impostazioni sono relative ai dati per l'accesso a questo spazio;
- la visualizzazione grafica (con limiti del grafico personalizzabili come da impostazioni).





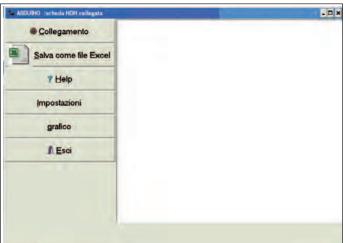


Figura 6: videata del programma su PC.

Figura 7: le impostazioni.

[1]hostname	ftp.termoinrete.com
[2]porta	ftp
[3]username	******
[4]password	*****
[5]hostdimame	/termoinrete.com/webcam
[6]porta seriale default	6
[7]minimo valore temp[°C]	-10.0
[8]max valore temp[°C]	50.0
[9]max valore tempo nel grafico[s]	600
[10]Tensione di riferimento [V]	1.100
[11]	
[12]salto temp termocoppia a V=VRIFERIM	ENTO 300.000
[13]temp riferimento termocoppia[°C]	20.0
[14]intervallo campionamento [s]	1
[15]beta NTC	3450
[16]R0 NTC[kohm]	10
[17]	

Il programma legge dalla porta seriale specificata l'unico dato e distingue a seconda del valore il tipo di sonda, quindi è in grado di determinare il valore trasferito nella trasmissione secondo la seguente tipologia (sia N il valore letto).

Sonda LM35: essendo l'uscita del LM35 10 mv / °C, a partire da 0 °C la temperatura t è = VRIF \cdot N \cdot 100 / 1023, dove N è depurato della costante 102000. Se la tensione di riferimento è 1,1 V, quando N = 1023 allora t = 110 °C.

Sonda LM335: l'uscita del LM35 è di 10 mv / °C. Regolando l'uscita a 2,93 V a 20 °C e assumendo VREF = 5 V, si ha che la temperatura t è = VRIF \cdot N \cdot 100 / 1023-293, dove N è depurato della costante. NTC 10K: in questo caso il NTC è inseri-

NTC 10K: in questo caso il NTC è inserito in un partitore di tensione ad alimentazione 5 V stabilizzata.

La tensione di uscita è Vout = 5 / (1+R), dove R è la resistenza in kohm del componente che vale quindi: R = (5 - Vout) / Vout. La tensione di uscita Vout è = N * VRIF / 1023. Nota che sia la tensione d'uscita, il software determina la temperatura tenendo conto che per lo NTC vale la relazione:

$R = R_0 e b (1/T - 1/T_0)$

dove R_0 è la resistenza alla temperatura di riferimento di 25 °C (per un NTC 10 K R_0 = 10, b è una costante caratteristica del materiale del termistore (tipicamente 3450). La temperatura t[°C] viene ricavata dall'espressione:

 $t = 1 / [(1 / b) LN (R / R_0) + 1 / T_0]$ -273,15

DS18B20: in questo caso la temperatura è: N – 800.

Termocoppia. Il salto di temperatura = $Tp - Ta = \Delta t = VRIF \cdot K \cdot N / 1023$. Se il salto di temperatura è Tp - Ta = 300 °C a VRIF = 1,1 V, la costante K è uguale a 300 / 1,1 = 273. Se VRIF = 5 V, K = 300 / 5 = 60

La lettura e visualizzazione vengono effettuate secondo l'intervallo di tempo di campionamento impostato. Alternativamente ogni 24 valori avviene l'invio del file dei dati o dell'immagine (se esiste il file webcam.jpg nella stessa cartella dell'eseguibile) sulla pagina web.





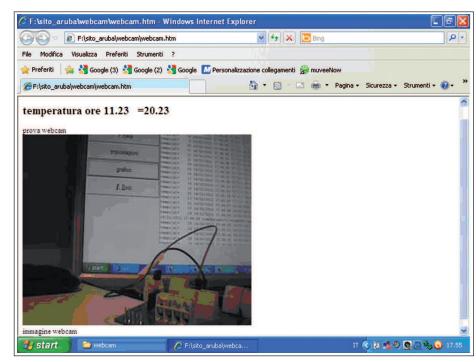
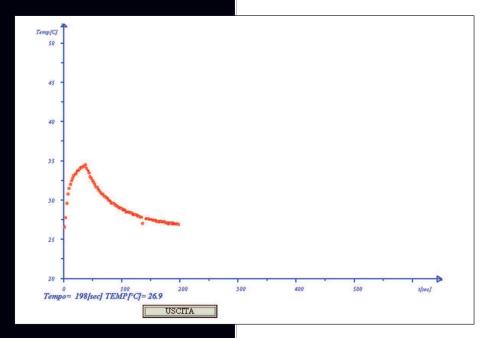


Figura 8: il grafico.

Figura 9: la pagina web.



Nella pagina web (nel nostro caso) è inserito un controllo di Explorer che legge il file di testo data1.txt e visualizza i dati corrispondenti.

Si riporta parte della pagina, comunque visibile all'indirizzo: http://www.termoinrete.com/webcam/webcam.htm

<PARAM NAME="DataURL" VA-

LUE="data1.txt"> <PARAM NAME="UseHeader" VALUE="TRUE"> <PARAM NAME="TextOualifier" VALUE="~"> <PARAM NAME="FieldDelim" VALUE=" | "> </OBJECT> <h2> DATASRC="#data1" TAFLD="temp"> </h2><P>prova webcam
 <IMG src="webcam.jpg" width="404"</pre> height="337" border="0">
 immagine webcam

CONCLUSIONI

L'abbinamento del PC al microcontrollore consente la massima visibilità nell'acquisizione dei dati. Con un hardware minimo e con un software opportuno è possibile realizzare un sistema, tutto sommato economico, di pubblicazione di dati fisici. E mai come in questo inverno, particolarmente rigido, la temperatura ha svolto un ruolo così di primo piano!

CODICE MIP 2819873