Esercitazione di Laboratorio:

Arduino

Coa Giulio (s236723) Licastro Dario (s234421)
Montano Alessandra (s238160)

19 gennaio 2020

1 Scopo dell'esperienza

Lo scopo di questa esercitazione è sviluppare un termometro digitale tramite l'uso di un sensore di temperatura e di una scheda Arduino.

2 Strumentazione utilizzata

La strumentazione usata durante l'esercitazione è:

Strumento	**Marca e Modello**	**Caratteristiche**
Multimetro	Agilent 34401A	
Oscilloscopio	Rigol DS1054Z	4 canali, $B = 50\,\mathrm{MHz}$, $f_\mathrm{c} = 1\,\mathrm{G}\frac{\mathrm{Sa}}{\mathrm{s}}$, $R_\mathrm{i} = 1\,\mathrm{M}\Omega$, $C_\mathrm{i} = 13\,\mathrm{pF}$, 12 Mbps di profondità di memoria
Generatore di segnali	Rigol DG1022	2 canali, $f_\mathrm{uscita} = 20\,\mathrm{MHz}$, $Z_\mathrm{uscita} = 50\,\Omega$
Scheda Arduino	UNO	$f_\mathrm{c,max} = 76.9\,\mathrm{k}\frac{\mathrm{Sa}}{\mathrm{s}}$
Sensore di temperatura	LM335	$S = 10\,\mathrm{m}\frac{\mathrm{V}}{\mathrm{K}}$, $V_\mathrm{out} = 0\,\mathrm{V}$ @ $0\,\mathrm{K}$ $\delta T = \pm 2\,^\circ\mathrm{C}$ Campo di temperatura pari a $-40 \div 100\,^\circ\mathrm{C}$ Resistenza termica pari a $165\,\frac{^\circ\mathrm{C}}{\mathrm{W}}$
Cavi coassiali		Capacità dell'ordine dei $80 \div 100\,\mathrm{p}\frac{\mathrm{F}}{\mathrm{m}}$
Connettori		

3 Premesse teoriche

3.1 Incertezza sulla misura dell'oscilloscopio

La misura del valore di un segnale tramite l'oscilloscopio (sia esso l'ampiezza, la frequenza, il periodo, etc.) presenta un'incertezza che dipende, principalmente, da due fattori:

- l'incertezza strumentale introdotta dall'oscilloscopio (ricavabile dal manuale).
- l'incertezza di lettura dovuta all'errore del posizionamento dei cursori.

Quest'ultima incertezza deriva dal fatto che il segnale visualizzato non ha uno spessore nullo sullo schermo.

3.2 Arduino

Arduino è una piattaforma elettronica open surce basata su un hardware di facile utilizzo e programmabie in un ambiente software dedicato.

3.2.1 Arduino UNO

Arduino UNO è una scheda composta da un convertitore ADC a 10 bit (8 bit se la frequenza d'utilizzo è maggiore di $15\,\mathrm{k}\frac{\mathrm{Sa}}{\mathrm{s}}$) che può essere alimentato da due distinte sorgenti, una interna alla scheda da $1.1 \pm 0.1\,\mathrm{V}$ e una esterna da $5 \pm 0.25\,\mathrm{V}$ ($4.85 \pm 0.4\,\mathrm{V}$ se si usa la porta USB 3.0 anzichè la porta USB 2.0).

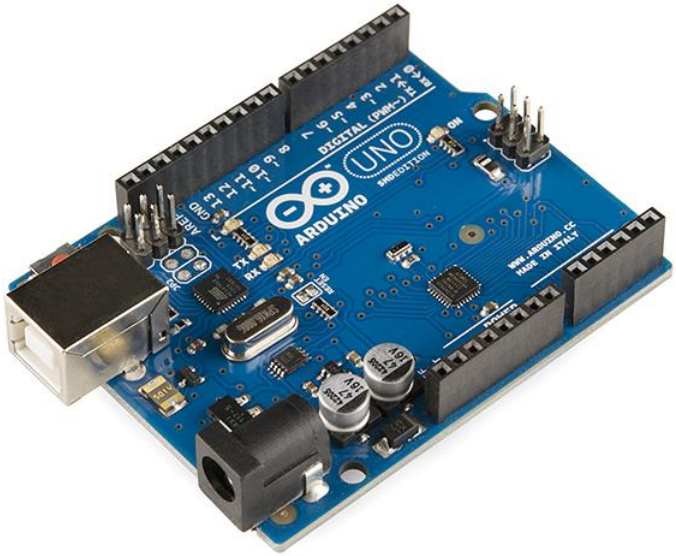

Figura 1: Arduino UNO.

3.3 Sensore LM335

Il sensore LM335 è un sensore di temperatura prodotto dalla National Semiconductor; esso permette di avere in uscita una tensione proporzionale alla temperatura rilevata ($V_{\mathrm{out}} = S \cdot T_{\mathrm{K}}$). Il suo comportamento è assimilabile a quello di un diodo di Zener la cui corrente I_{D} deve essere compresa nell'intervallo $0.4\,\mathrm{mA} \div 5\,\mathrm{mA}$.

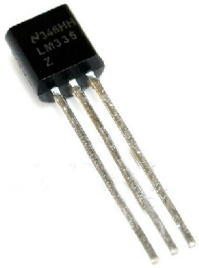

Figura 2: Sensore LM335.

3.4 Other

.

4 Esperienza in laboratorio

4.1 Circuito di condizionamento

Per i nostri scopi, il sensore LM335 deve lavorare lavorare in regione di polarizzazione inversa, percui esso verrà utilizzato con il catodo collegato alla massa e con l'anodo collegato alla sorgente di tensione.
Al fine di garantire un corretto funzionamento del sensore, dobbiamo garantire che la corrente che vi circola all'interno sia nel range di funzionamento; a tale scopo applichiamo una resistenza R_1 a monte del diodo, di modo che la corrente che scorra nel diodo rientri nel suddetto range.

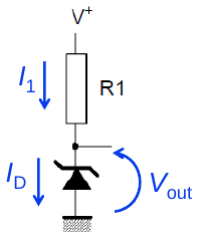

Figura 3: Circuito di condizionamento.

4.2 Misura della temperatura

```
#include <string.h>

//#define INT

int pin, time, D_out, N_B;
float S, T_C, T_K, V_out, V_FR;

void setup() {
 Serial.begin(9600); // setup Serial
 pin = A0; // set the pin to read data
 time = 1000; // set how much time the sensor must execute the measurement of the temperature
 S = 10 * pow(10, -3);
 N_B = 10;
 V_FR = 4.9;
 #ifdef INT
  analogReference(INTERNAL); // set the type of alimentation
 #endif
}
void loop(){
 D_out = analogRead(pin); // read data from the pin
 T_K = D_out * V_FR / pow(2, N_B) * 1 / S;
 T_C = T_K - 273.15;
 V_out = D_out * V_FR / pow(2, N_B);
 Serial.println("D_out: " + String(D_out) + " -- V_out: " + String(V_out) + " V -- T: " +
String(T_K) + " K -- T: " + String(T_C) + " °C");
 delay(time);
}

```

4.3 Stima dell'incertezza

.

4.4 Modifica del circuito di condizionamento

.

4.5 Firmware del micro-controllore

.

4.6 Caratterizzazione del sistema

.

5 Risultati

5.1 Circuito di condizionamento

In base ai dati fornitici, abbiamo proceduto al calcolo della tensione massima e della tensione minima a cui il sensore LM335 deve lavorare.

$$\begin{aligned} V_{\text{out,max}} &= S \cdot T_{\text{K,max}} = \\ &= S \cdot (T_{\text{max}} + 273.15) = \\ &= 10m \cdot (50 + 273.15) = \\ &= 3.23\,\text{V} \end{aligned}$$

$$\begin{aligned} V_{\text{out,min}} &= S \cdot T_{\text{K,min}} = \\ &= S \cdot (T_{\text{min}} + 273.15) = \\ &= 10m \cdot (5 + 273.15) = \\ &= 2.78\,\text{V} \end{aligned}$$

Essendo che la resistenza R_1 è posta in serie al sensore, la corrente che scorre nel diodo è uguale a quella che scorre nella resistenza (in realtà è approssimabile, dato che si va a misurare la tensione ai capi del diodo e si ha un minimo di "perdite").

$$I_{\text{D}} \approx I_1 = \frac{V_{\text{s}} - V_{\text{out}}}{R_1}$$

Da ciò, si ha che le limitazioni sulla corrente che attraversa il diodo diventano

$$\frac{V_{\text{s}} - V_{\text{out,min}}}{R_1} > 0.4\,\text{mA}$$

$$\frac{V_{\text{s}} - V_{\text{out,min}}}{R_1} < 5\,\text{mA}$$

Da cui si ha che

$$\frac{V_{\text{s}} - V_{\text{out,min}}}{0.4m} > R_1 > \frac{V_{\text{s}} - V_{\text{out,min}}}{5m}$$

$$4425\,\Omega > R_1 > 444\,\Omega$$

Successivamente, abbiamo stimato la temperatura dell'ambiente in cui il diodo avrebbe lavorato, tenendo conto dell'autoriscaldamento del sensore, e, di conseguenza, la tensione in uscita che esso avrebbe prodotto.

$$\begin{aligned} V_{\text{out}} &= S \cdot T_{\text{K}} = \\ &= S \cdot (T + 273.15) = \\ &= 10m \cdot (25 + 273.15) = \\ &= 2.98 = \\ &\approx 3\,\text{V} \end{aligned}$$

Ed abbiamo imposto che I_{D} fosse di 2 mA, da cui

$$\begin{aligned} R_1 &= \frac{V_{\text{s}} - V_{\text{out}}}{I_1} = \\ &= \frac{5-3}{2m} = \\ &= 1\,\text{k}\Omega \end{aligned}$$

5.2 Misura della temperatura

```
#include <string.h>

//#define INT

int pin, time, D_out, N_B;
float S, T_C, T_K, V_out, V_FR;

void setup() {
Serial.begin(9600); // setup Serial
pin = A0; // set the pin to read data
time = 1000; // set how much time the sensor must execute the measurement of the temperature
S = 10 * pow(10, -3);
N_B = 10;
V_FR = 4.9;
#ifdef INT
analogReference(INTERNAL); // set the type of alimentation
#endif
}
void loop(){
D_out = analogRead(pin); // read data from the pin
T_K = D_out * V_FR / pow(2, N_B) * 1 / S;
T_C = T_K - 273.15;
V_out = D_out * V_FR / pow(2, N_B);
Serial.println("D_out: " + String(D_out) + " -- V_out: " + String(V_out) + " V -- T: " +
 String(T_K) + " K -- T: " + String(T_C) + " °C");
delay(time);
}

```

5.3 Stima dell'incertezza

.

5.4 Modifica del circuito di condizionamento

.

5.5 Firmware del micro-controllore

.

5.6 Caratterizzazione del sistema

.