

Sensore Termico con Arduino

Nao Pross

November 28, 2015

1 Elettronica

L'intero circuito é alimentato dalla tensione di $+5\text{ V}$ che esce dal pin dell'Arduino. Le misurazioni della temperatura avvengono su V_t attraverso il pin di misura Pin_{A0} e a dipendenza del risultato viene illuminato il LED RGB. Il bottone S_1 permette di eseguire misurazioni di variazioni di temperatura più piccole.

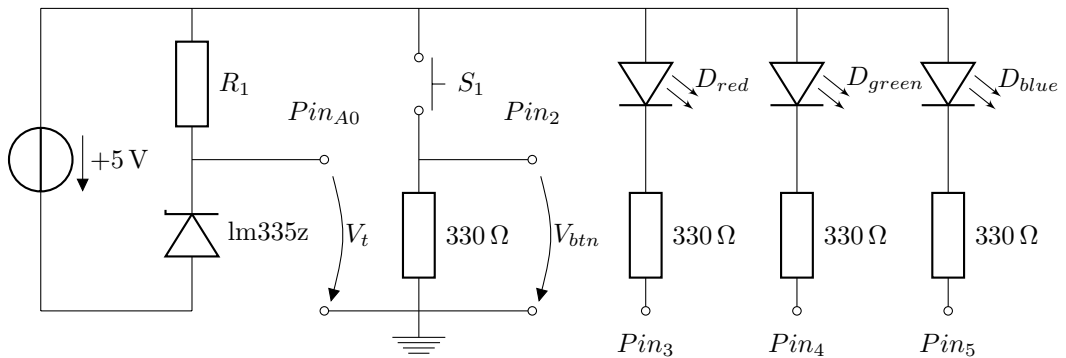


Figure 1: Circuito Elettronico

Nota: Il LED RGB é rappresentato con 3 diodi LED dei tre colori.

1.1 Controllo del LED RGB

Il LED RGB, rappresentato nello schema come tre led D_{red} , D_{green} e D_{blue} é combinato in tale maniera per richiedere meno potenza dalla sorgente di tensione. Questo sistema viene applicato spesso in elettronica ma implica che il controllo dei PIN sia inverso. Quindi per avere uno stato in cui D_n é spento si deve generare una tensione di -5 V dal Pin_n , mentre normalmente al Pin_n dovrebbero esserci 0 V .

Avere il componente collegato in tale modo implicherà delle modifiche nel software del circuito.

2 Sensore

La tensione misurata in V_t é la tensione che cade sul diodo Zener LM335z, questo sensore é configurato per funzionare da -40°C a 100°C . Nel datasheet é dato che a 25°C ha una tensione di 2.98 V , da questo possiamo trovare attraverso una proporzione, come convertire la tensione in temperatura.

(Nota: Nel calcolo i 25°C vengono convertiti in Kelvin)

$$\begin{aligned}\frac{V_t}{2.98\text{ V}} &= \frac{T_K}{25^\circ\text{C}} \\ T_K &= \frac{298\text{ K}}{2.98\text{ V}} * V_t \\ \implies T_k &= 10 * V_t\end{aligned}$$

L'Arduino però legge un valore analogico come un numero intero da 0 a 1023, quindi per convertire il valore di nuovo in Volts é necessario sapere a quanto corrisponde un unità della misura del pin di Arduino. Dalle formule precedenti possiamo trarre una proporzione tra le due grandezze e quindi estrarre V_t .

$$\frac{M_{A0}}{1024} = \frac{V_t}{V_{max}} \implies V_t = \frac{M_{A0} * V_{max}}{1024} \text{ con } 0 \leq M_{A0} \leq 1023$$

V_{max} può essere calcolato semplicemente attraverso la proporzione precedente, quindi

$$V_{max} = \frac{1}{10} * T_K \implies V_{max} = \frac{373\text{ K}}{10} = 37.3\text{ K}$$

Utilizzando questo metodo creiamo una scala, e dato che la una unità in gradi Kelvin equivale a una nei gradi Celsius con il calcolo possiamo ottenere direttamente una temperatura in $^\circ\text{C}$.

$$\begin{aligned}V_t &= \frac{37.3\text{ K}}{1024} * M_{A0} \\ T_{^\circ\text{C}} &= 10 * V_t\end{aligned}$$

3 Software

3.1 Rappresentazione

Per rappresentare la temperatura ambiente con il circuito ho deciso di utilizzare il LED RGB, avendo la formula per convertire l'input del sensore in $^\circ\text{C}$ posso anche andare nella direzione opposta. Quindi la luminosità del LED sarà proporzionale alla temperatura. Inoltre potendo utilizzare i colori ho pensato di rappresentare in rosso le temperature superiori a 0°C , in blu le temperature inferiori a 0°C e in verde la temperatura di esattamente 0°C .

Purtroppo però in tale modo sarebbe difficile dimostrare il funzionamento del circuito, quindi ho aggiunto un bottone S_1 che permette di spostare la referenza. Nel momento in cui S_1 viene premuto la temperatura letta viene impostata come referenza e utilizzata al posto di 0°C . Di nuovo quindi il LED sarà rosso se la temperatura é maggiore della referenza, blu se inferiore e verde se uguale. In questo modo la misurazione sarà molto sensibile e il led cambierà colore molto frequentemente poiché in una stanza la temperatura ambiente non é costante ma cambia in continuazione ($\pm 0.06^\circ\text{C}$).

3.2 Codice Sorgente

Il codice sorgente in C++ si trova sul mio GitHub in un repository chiamato *2samb.hw.and.sw* nella cartella *_00.thermal_sensor* o più semplicemente al seguente link:

https://github.com/NaoPross/samb2_hw_and_sw/tree/master/_00_thermal_sensor

3.3 Struttogrammi

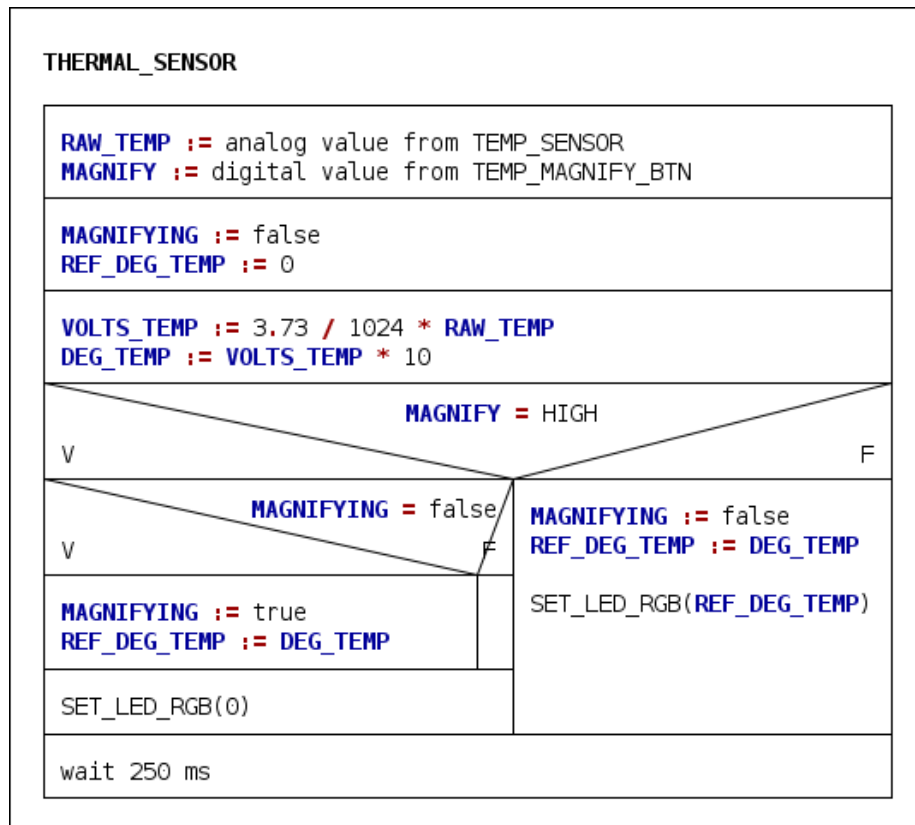


Figure 2: Diagramma Nassi-Schneiderman del programma

Nel programma é utilizzata una funzione SET_LED_RGB descritta nello schema successivo.

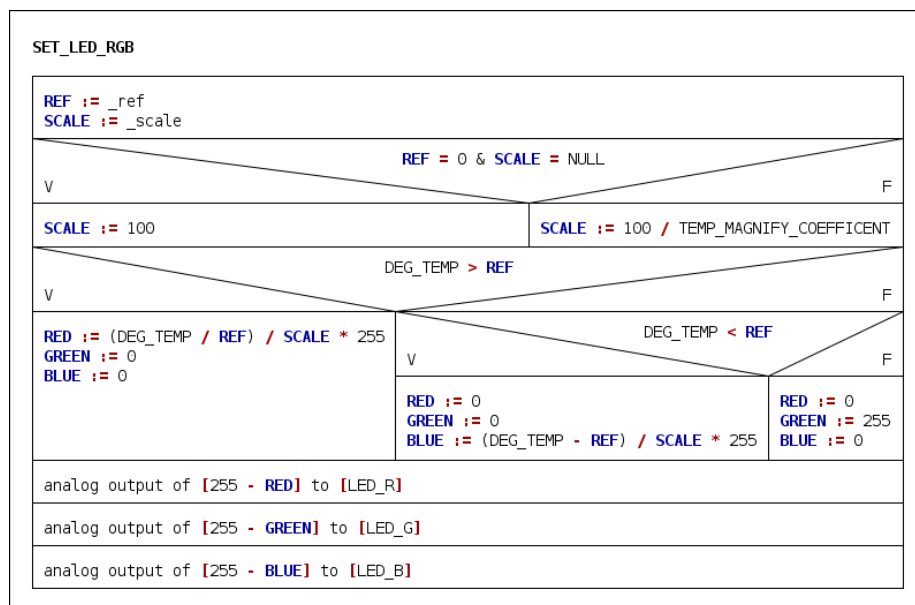


Figure 3: Diagramma Nassi-Schneiderman della funzione SET_LED_RGB