



Misura di temperatura con sensore LM335 e scheda Arduino Uno



Obiettivo dell'esperienza

- Sviluppare un termometro digitale usando
 - ↳ un sensore elettronico di temperatura
 - ✓ LM335 (costo \approx 1 EUR)
 - ↳ una scheda Arduino
 - ✓ Arduino UNO (costo \approx 20 EUR)
- Progettare il circuito di condizionamento del sensore
- Stimare l'incertezza attesa

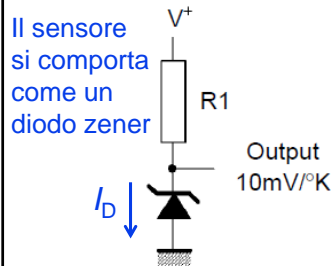
Il sensore

➤ National Semiconductor modello LM335

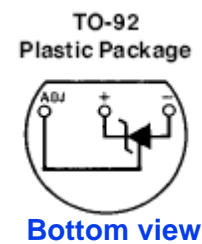
➤ Sensibilità nominale $S = 10 \text{ mV/K}$

➤ Uscita riferita allo zero assoluto

$$\hookrightarrow V_{\text{out}} = 0 \text{ V @ } 0 \text{ K} \equiv -273.15 \text{ °C}$$



La corrente I_D deve essere inclusa nel campo $(0.4 \div 5) \text{ mA}$



Circuito di condizionamento

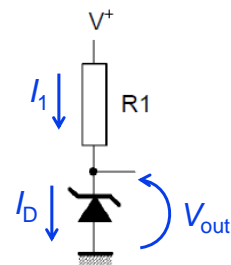
➤ Tensione di alimentazione fornita dalla scheda Arduino (uscita 5 V)

\hookrightarrow Verificare se l'uscita 5 V è in grado di alimentare il sensore

➤ Stimare il valore di R_1

$$I_D \approx I_1 = \frac{V^+ - V_{\text{out}}}{R_1}$$

I_D dipende da V_{out} , che a sua volta dipende dalla temperatura in misura





Circuito di condizionamento

- Stimare il valore di R_1

↳ $V^+ = 5\text{ V}$

↳ Campo di temperatura: $(5 \div 75)^\circ\text{C}$

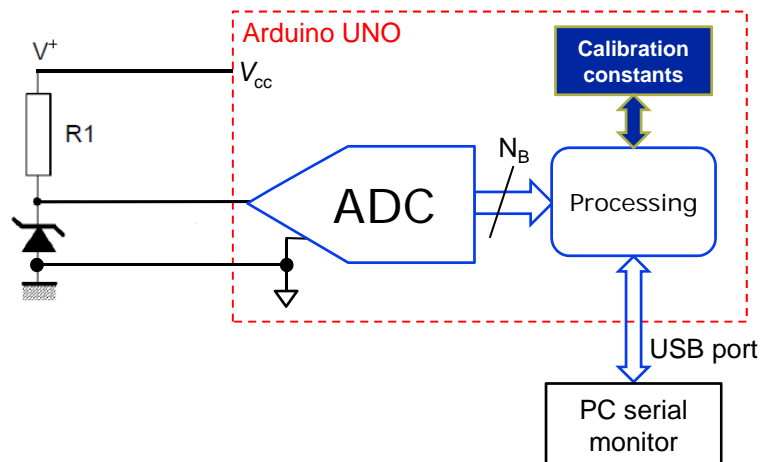
$$I_{D,\min} = \frac{V^+ - V_{\text{out,max}}}{R_1} > 0.4\text{ mA}$$

$$I_{D,\max} = \frac{V^+ - V_{\text{out,min}}}{R_1} < 5\text{ mA}$$

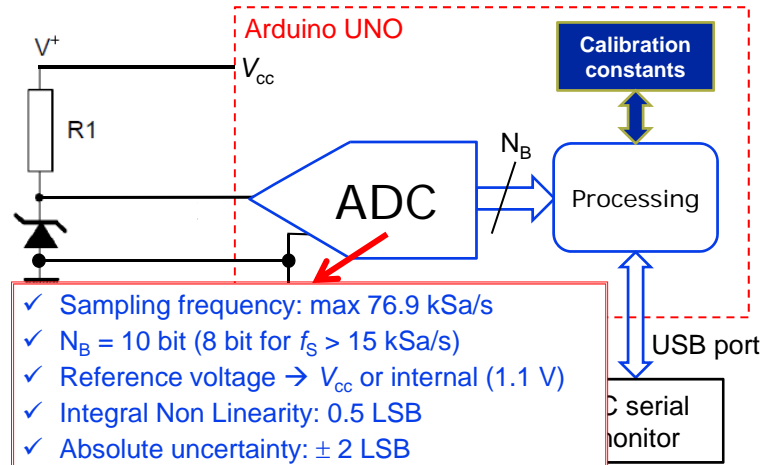
➤ ...



Schema di principio



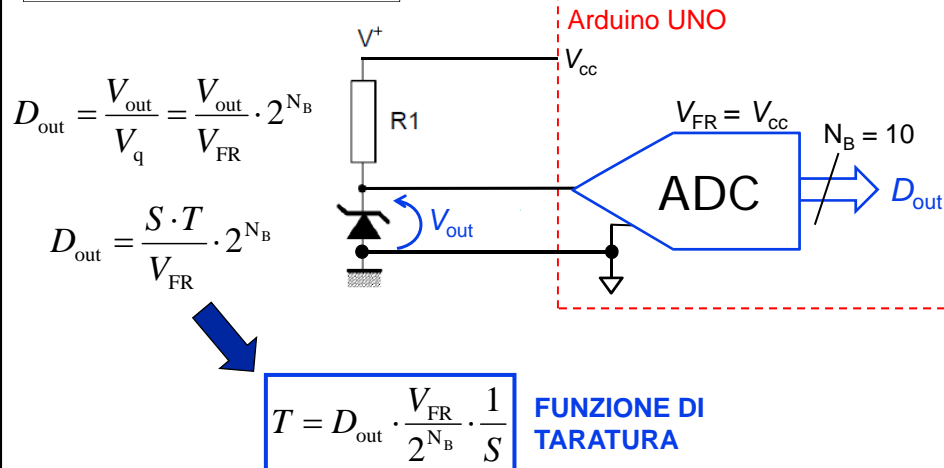
Schema di principio



Schema di principio

- Campo di temperatura: $(5 \div 75) ^\circ\text{C}$
 - $\approx (278 \div 348) \text{ K}$
 - ↪ $V_{out}: (2.78 \div 3.48) \text{ V}$
- Tensione di riferimento: V_{cc}
 - V_{cc} : dalla porta USB
 - ↪ USB 2.0 $\rightarrow V_{cc} = (5 \pm 0.25) \text{ V}$
 - ↪ USB 3.0 $\rightarrow V_{cc}$ tra 4.45 V e 5.25 V

Catena di misura



Catena di misura

$$T = D_{out} \cdot \frac{V_{FR}}{2^{N_B}} \cdot \frac{1}{S}$$

➤ Stima dell'incertezza (modello deterministico)

$$\delta T = \left| \frac{\partial T}{\partial D_{out}} \right| \cdot \delta D_{out} + \left| \frac{\partial T}{\partial V_{FR}} \right| \cdot \delta V_{FR} + \left| \frac{\partial T}{\partial S} \right| \cdot \delta S$$

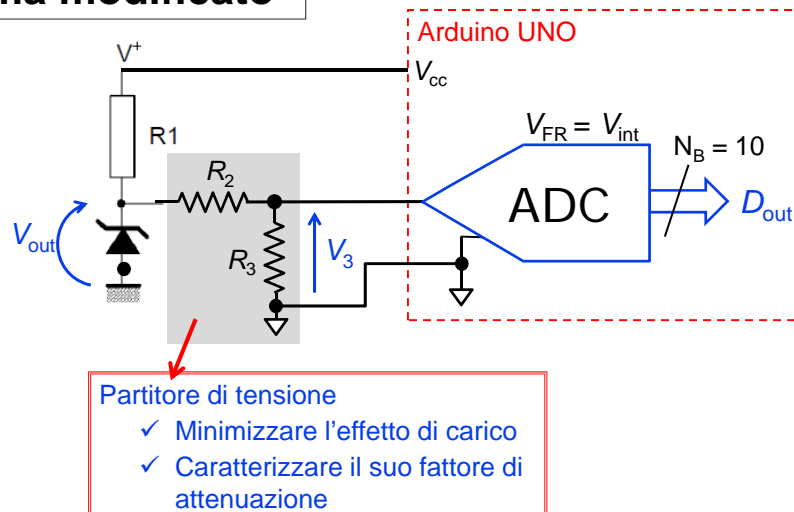
$$\delta T = \frac{V_{FR}}{S \cdot 2^{N_B}} \cdot \delta D_{out} + \frac{D_{out}}{S \cdot 2^{N_B}} \cdot \delta V_{FR} + \delta T^{\text{sensor}}$$

$\delta T^{V_{FR}} \rightarrow 17.4 \text{ K !!! (per } D_{out, \max} = 713)$

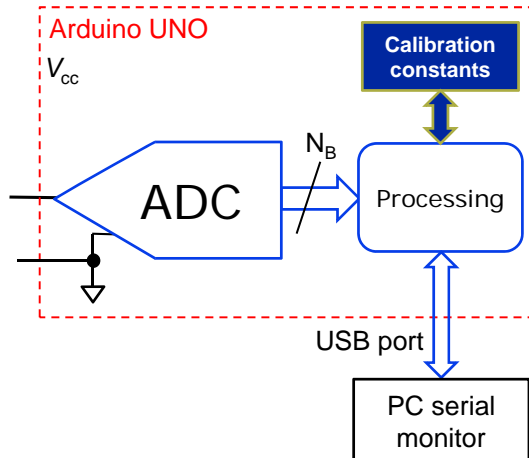
Catena di misura

- Possibili interventi per ridurre l'incertezza
 - ✓ Usare il riferimento interno dell'ADC (1.1 V)
 - ↳ Necessario attenuare il segnale di uscita del sensore
 - ↳ $V_{int} = (1.1 \pm 0.1) \text{ V}$ ☹☹☹
 - ✓ Misurare il riferimento di tensione dell'ADC
 - ↳ V_{cc} oppure V_{int} ?
 - ↳ La prima scelta non richiede un partitore di tensione, **ma la costante di taratura è legata al PC usato**

Schema modificato



Firmware del micro-controllore



- Gestire la comunicazione seriale con il PC
- Configurare l'ADC
- Acquisire il segnale di tensione
- Implementare la funzione di taratura
- Fornire le misure di temperatura

Firmware del micro-controllore

- Gestire la comunicazione seriale
 - Fissare il *baud rate*, usare la *built-in function* `serialEvent()`, ...
 - ↳ Vedere capitoli 3, 4 e 5 della guida *Using Arduino boards in Measurements for dummies*
- Configurare l'ADC
- Acquisire il segnale di tensione
 - *built-in functions* `analogReference()` e `analogRead()`
 - ↳ Vedere capitolo 6 della guida



Firmware del micro-controllore

- Implementare la funzione di taratura
 - Convertire il codice di uscita dell'ADC in misura di temperatura
 - ↳ Usare le opportune costanti di taratura (V_{cc} o V_{int} , fattore di attenuazione)
 - ↳ Stimare l'incertezza di misura
- Fornire le misure di temperatura
 - Inviare i risultati al *serial monitor* via USB



Caratterizzazione del sistema

- Specificare:
 - ✓ Campo di temperatura
 - ✓ Risoluzione
 - ✓ Incertezza
 - ↳ *unadjusted*
 - ↳ *adjusted* (ADC *offset* e *gain error*)
 - ✓ Dimensione del *firmware*