

Nome e Cognome:

 LUN MAR GIO

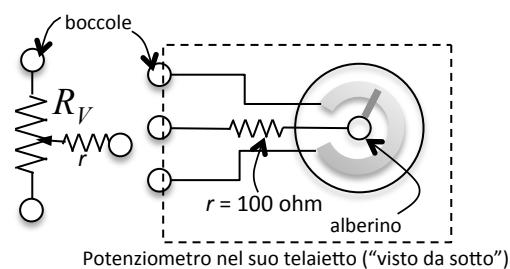
Data:

3

Acquisizione di un campione di segnali continui con Arduino

L'esperienza consente di costruire campioni di misure di d.d.p. continue tramite presa dati automatizzata via Arduino e ha lo scopo principale di calibrarne il digitalizzatore (per confronto con il multimetro digitale). Parte preliminare è la realizzazione e caratterizzazione di un partitore di tensione con potenziometro.

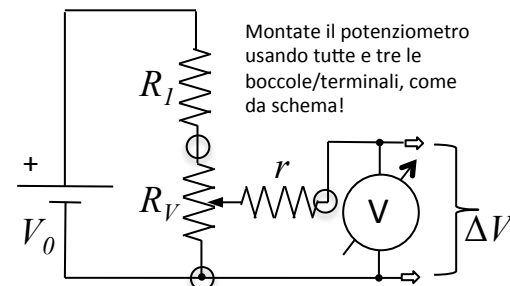
La figura riporta il simbolo del potenziometro assieme a una rappresentazione schematica della sua costruzione e del montaggio del componente sul telaio (nel telaio possono esserci altre boccole, da non usare). Normalmente, $R_V = 4.7$ kohm (nominali); il telaio alloggia anche un resistore (normalmente $r = 100$ ohm, nominali) saldato al terminale centrale del potenziometro come in figura.



- Misurate le resistenze R_I , R_V , e V_0 (a "circuito aperto"): si consiglia di scegliere $R_I = 330$ ohm oppure 680 ohm.

R_I []	R_V []	V_0 [V]

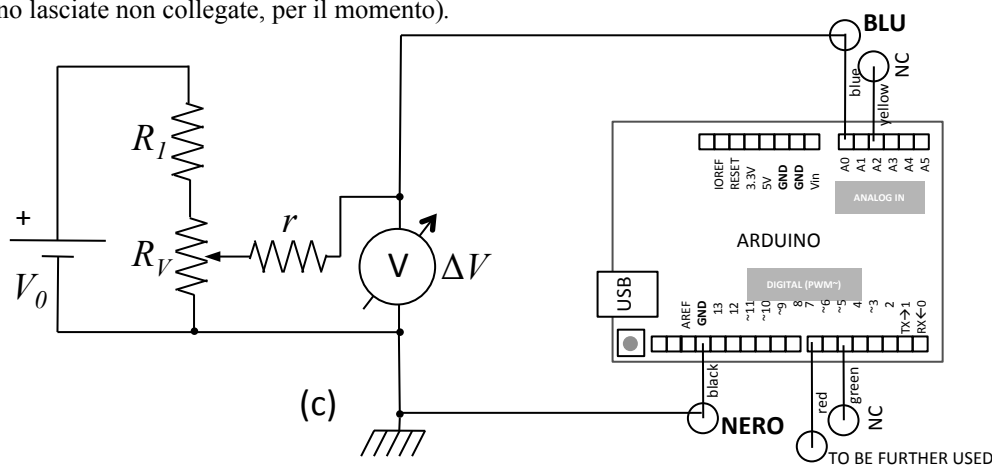
- Determinate dalla conoscenza dello schema (dunque senza fare misure) la resistenza di Thévenin R_{Th}^* del generatore costituito dalla sorgente V_0 (da considerare qui come un generatore ideale) e dal collegamento di R_I , R_V , r . Dovete eseguire il calcolo considerando la regolazione del potenziometro che permette di avere il massimo valore ΔV_{MAX} della d.d.p. in uscita; prendete le resistenze in valore nominale e lasciate perdere l'indicazione dell'incertezza, o tolleranza. Notate che il generatore di cui dovete misurare la resistenza di Thévenin è quello che fornisce la d.d.p. ΔV .



Montate il potenziometro usando tutte e tre le boccole/terminali, come da schema!

- Realizzate il partitore collegando in modo corretto il potenziometro (tre boccole devono essere impiegate!) e misurate i valori minimo e massimo della d.d.p. ΔV in uscita.
- Preliminarmente all'uso di Arduino, dovete eseguire l'upload dello sketch `ardu2016.ino` nella sua memoria utilizzando il programma Arduino (o Arduino IDE) nel computer di laboratorio (lo sketch si trova nella directory `/Arduini/`). Ricordate di collegare Arduino alla presa USB del computer! Trovate altre informazioni generali sull'uso di Arduino in un foglio che si trova sul vostro banco (lasciatecelo!).
- A questo punto potete collegare il partitore alla scheda Arduino come rappresentato in figura, facendo attenzione a non commettere errori (dovete in sostanza collegare le boccole volanti nera e blu al partitore rispettando le polarità, le altre boccole vanno lasciate non collegate, per il momento).

	espressione
R_{Th}^*	
R_{Th}^*	valore nominale []
ΔV_{MIN}	[]
ΔV_{MAX}	[]



- Quindi dovete modificare a vostro piacimento lo script di Python (nome `ardu2016.py`, posto nella stessa directory) che serve per gestire la comunicazione seriale via USB: in particolare dovete decidere il numero di acquisizioni (variabile `nacqs`, di default pari a 1), ognuna costituita da 256 coppie di misure (tempo di digitalizzazione in μs e valore digitalizzato della d.d.p. in digit), e il nome del file. Lo script contiene anche l'indicazione dell'intervallo di campionamento nominale Δt_{nom} , di default pari a $500 \mu s$ (non è necessario modificarlo in questa esperienza).
- A questo punto siete pronti per partire con le misure per la calibrazione del digitalizzatore: dunque selezionate, ruotando l'alberino del potenziometro, un certo valore di ΔV (controllate, osservando il display del tester, che esso si mantenga costante durante le misure).

3

8. Per la calibrazione di Arduino è sufficiente impiegare un singolo blocco di 256 misure ($n_{acqs}=1$) e prendere nota del valore medio e della deviazione standard sperimentale indicate in console. Costruite una tabella con i valori ΔV_j letti dal multimetro digitale per j -diverse regolazioni del potenziometro e con i corrispondenti valori digitalizzati X_j , determinandone la barra di errore rispettivamente come da manuale e come il massimo tra la deviazione standard sperimentale e il valore convenzionale, $\Delta X = \pm 1$ digit.

j	ΔV_j [V]	X_j [digit]
1		
2		
3		
4		
5		
6		

j	ΔV_j [V]	X_j [digit]
7		
8		
9		
10		
11		
12		

9. Fate un grafico di ΔV_j in funzione di X_j e un best-fit lineare (si consiglia numerico) secondo la funzione $\Delta V = \alpha + \beta X$. Riportate qui i valori dei parametri del fit e scrivete tutte le altre informazioni necessarie nel foglio stampato di grafico e best-fit.
10. Eseguite una “calibrazione alternativa”: a questo scopo misurate la d.d.p. ΔV_{pin7} presente tra il pin 7 di Arduino (boccola rossa) e la linea di massa, o terra. Infatti lo sketch istruisce Arduino a usare questo pin come porta digitale di uscita e a porla a “livello alto” (accesa), corrispondente, secondo le specifiche, al massimo valore di d.d.p. che Arduino può digitalizzare. Determinate con una singola misura il fattore di conversione ξ che lega la lettura digitalizzata X alla d.d.p. ΔV supponendo la relazione di proporzionalità diretta $\Delta V = \xi X$ e ricordando che Arduino ha una “dinamica” di 10 bit.

α	[mV]
β	[mV/digit]
ΔV_{pin7}	[V]
ξ	[mV/digit]

11. Commentate meglio che potete (in modo quantitativo!) le eventuali discrepanze tra i risultati delle due calibrazioni. Lo scopo dei commenti dovrebbe essere quello di identificare in maniera quantitativa l'errore che si compie utilizzando la calibrazione alternativa rispetto all'altra (state attenti: questo errore non è costante per tutte le letture!).

Commenti (per bene; se non basta il riquadro, usate un altro foglio!):

12. È auspicabile che troviate il tempo, magari portandovi i dati a casa, per svolgere almeno qualcuna delle ulteriori analisi possibili con la configurazione montata. In ordine di importanza decrescente, potreste ad esempio: (i) analizzare il campione di misure digitalizzate allo scopo di studiarne la distribuzione (istogramma); (ii) ripetere le operazioni impostando Arduino in modo che esso impieghi il riferimento interno a 1.1 V (nominali): allo scopo, fate l'upload dello sketch `ardu_1V1_2016.ino` e ponete attenzione a usare $\Delta V \leq 1.1$ V; (iii) costruire il campione degli intervalli di campionamento Δt_j (gli intervalli di tempo tra una digitalizzazione e la successiva) e analizzarlo, confrontando i risultati con l'impostazione nominale Δt_{nom} e determinando l'incertezza da attribuire alla misura dei tempi eseguita da Arduino; (iv) eseguire delle acquisizioni mantenendo scollegata la porta di ingresso, A0. Qualsiasi cosa facciate, ora o in futuro, documentatela adeguatamente usando altri fogli.