**LABORATORIO 5**

**Determinazione della linearità di una scheda Arduino**

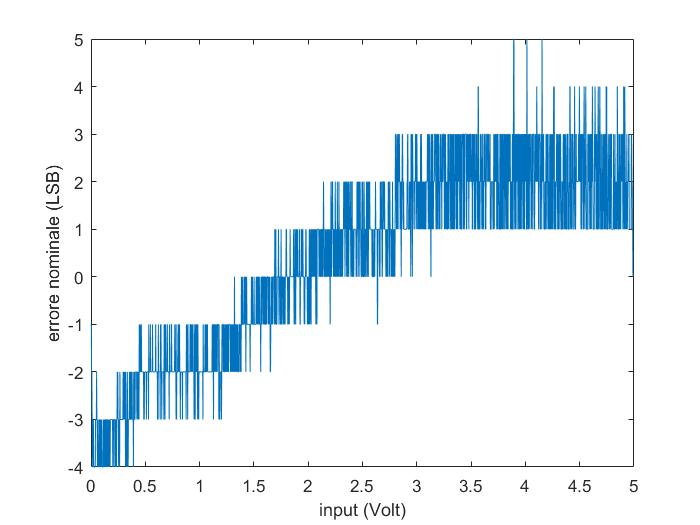
L’obiettivo dell’esperienza è quello di caratterizzare la caratteristiche statica del convertitore ADC di Arduino. Per far questo, ci serviremo di una scheda di acquisizione NI 6030E e di un multimetro HP 34401A: usiamo la scheda di acquisizione per generare le tensioni in ingresso al convertitore ADC di Arduino; il multimetro misura le tensioni in uscita dalla scheda. Il confronto tra le tensioni misurate dal multimetro e quelle lette dall’ADC consente di eseguire un controllo di taratura sul comportamento del convertitore. Dai dati ottenuti calcoliamo i parametri di un modello che porti le letture dell’ADC vicino a quelle attese. Possiamo mettere a punto il convertitore ADC con tre diversi modelli:

* caratteristica nominale: modellizziamo la risposta dell’ADC con una retta del tipo V=k\*N;
* caratteristica end point: modellizziamo la risposta dell’ADC con una retta del tipo V=VN=Nmax/Nmax\*N + Voff;
* caratteristica ottima: modellizziamo la risposta dell’ADC con una retta del tipo V= Voff’ + k’\*N ottenuta da una stima ai minimi quadrati.

Scriviamo degli appositi programmi Arduino e C#: il codice C# gestisce la generazione delle tensioni in ingresso ad Arduino tramite la scheda d’acquisizione, la lettura di quest’ultime tramite multimetro e Arduino; il programma Arduino si occupa a comando di leggere la tensione in ingresso al convertitore ADC e inviare il valore acquisito al PC tramite linea seriale. Il codice C# stampa poi, su due file separati, i valori del multimetro e del convertitore.

La tensione di riferimento del convertitore digitale è circa 5V ed è ricavato dall’USB: ha un’incertezza del 10% e non è stabile nel tempo; l’instabilità causerà problemi alla nostra verifica di taratura. L a scheda genera una tensione compresa tra 0V e 5V. Raccogliamo 2047 campioni, al fine di ottenere una migliore verifica di taratura: generiamo 2047 tensioni a partire a 0V fino a 5V con step di 5V/2047.

*Di seguito viene riportata la verifica di taratura e messa a punto con prescaler impostato uguale a 128.*

La scelta della tensione di riferimento ricavata dall’USB si è rivelata una scelta infelice. Dalla figura 1, che mostra di quanto si discosti la caratteristica da noi misurata del convertitore ADC da quella nominale dichiarata dal costruttore in corrispondenza dei diversi valori di tensione in ingresso al convertitore generati dalla scheda e misurati col multimetro, evidenzia un qualche fenomeno di deriva. Visto il lungo tempo impiegato nel raccogliere i campioni e l’instabilità dei campioni, possiamo ipotizzare una qualche variazione della tensione di riferimento del convertitore durante l’esperienza.

Possiamo adesso provare a modellizzare il comportamento del convertitore con una caratteristica ‘endpoint’. Per ricavarne i coefficienti, utilizziamo il primo e l’ultimo dei campioni raccolti: il primo sarà il nostro Voff; l’ultimo sarà VN=Nmax .

Figura . Errore nominale

La figura 2 mostra la differenza in Volt tra i valori del convertitore e quelli della caratteristica con cui è stato appena modelizzato.

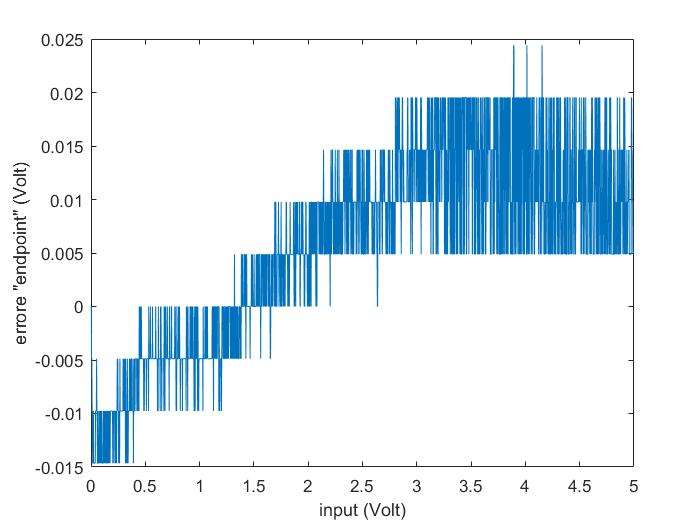


Figura . Errore "endpoint"

Il modello “endpoint” approssima molto meglio il comportamento del convertitore ADC.

Possiamo provare a mettere a punto il convertitore ADC di Arduino con una caratteristica ottima, basata su una stima ai minimi quadrati. Abbiamo utilizzato le formule della regressione lineare per trovare i coefficienti del modello:

dove Xi e Yi sono rispettivamente i valori misurati dell’ADC e del multimetro.

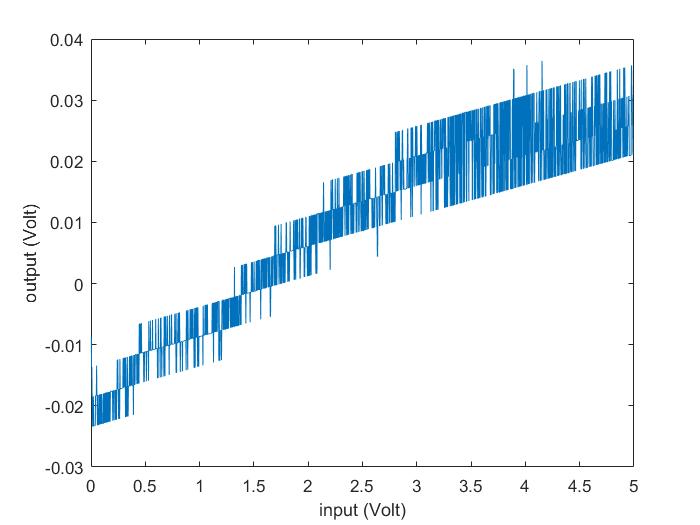
La figura 3 mostra quanto dista il modello dal comportamento reale del dispositivo. Come possiamo ben vedere, proprio a causa del fenomeno di deriva di cui parlavamo prima, la caratteristica “ottima” approssima peggio della caratteristica nominale il comportamento dell’ADC.

Figura . Errore "caartteristica endpoint"

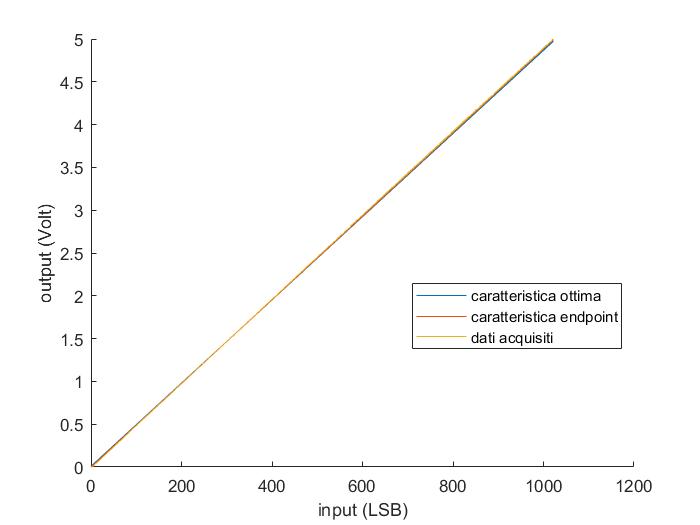
Possiamo infine riportare (figura 4), le 3 caratteristiche con cui abbiamo descritto il comportamento del convertitore in un unico grafico, in modo da evidenziare il perchè la caratteristica endpoint sia la migliore per la messa a punto dello strumento.

Figura . Modelli approssimanti il comportamento dell'ADC

Cambiando il valore del prescaler, avremmo potuto constatare un peggioramento dell’ordine di qualche LSB dell’uscita del convertitore. Portando il prescaler a un valore molto basso, il convertitore avrebbe smesso di funzionare.

Nella nostra trattazione abbiamo ipotizzato che il fenomeno di deriva sia da imputare all’instabilità della tensione di riferimento. Per confermare la nostra ipotesi si potrebbe ripetere l’esperienza. Raccogliere meno punti potrebbe essere un altro mezzo per agevolarne il buon esito. Qualora si accerti l’ipotesi della non stabile tensione di riferimento, la verifica di taratura e la messa a punto sopra descritte non sarebbero corrette, ma da rifare. Si potrebbe poi pensare di ottimizzare il programma C# e quello di Arduino per poter raccogliere il medesimo numero di campioni in minor tempo. Altra possibile correzione sarebbe quella di cambiare la tensione di riferimento: si potrebbe usare una tensione di riferimento esterna, oppure il riferimento interno (1.1 V circa).