Campionamento e digitalizzazione: una breve introduzione

fuso@df.unipi.it; http://www.df.unipi.it/~fuso/dida

(Dated: version 2 - FF, 30 Dicembre 2014)

È linguaggio comune distinguere le grandezze misurate e i metodi di misura tra *analogici* e *digitali*. Ormai da alcuni decenni il mondo viaggia veloce verso la diffusione capillare di metodi digitali, pressoché per tutte le necessità pratiche, in particolare quelle legate allo scambio e al trattamento delle informazioni. Nel nostro piccolissimo, anche noi possiamo toccare con mano vantaggi (molti) e svantaggi (pochi, legati soprattutto alla scelta dell'hardware) dell'approccio digitale, grazie in particolare all'uso di Arduino. Questa nota intende fornire un minimo di background concettuale utile per affrontare questo approccio.

I. ANALOGICO VS DIGITALE

Le grandezze di interesse per l'elettricità e l'elettronica hanno generalmente valori che "appaiono" continui (grandezze reali), cioè non discreti, né quantizzati[1]. La misura di grandezze di questo tipo si dice talvolta *analogica*. Un buon esempio di misura analogica è quella che si fa con uno strumento a lancetta, dove idealmente la lancetta può muoversi in modo continuo assumendo con continuità qualsiasi posizione angolare.

Ci sono poi delle grandezze la cui misura implica un conteggio, cioè l'ottenimento di un valore rappresentato da un *numero intero* (detto anche *parola digitale*), per esempio quando si contano gli atomi contenuti in un campione, i fotoni emessi per unità di tempo da una sorgente, gli elettroni presenti in un pezzo di materiale, e così via. La misura di queste grandezze si dice talvolta *digitale*. Un buon esempio di misura digitale è il conteggio degli impulsi di un oscillatore (non necessariamente elettronico: potete considerare per esempio come conteggio il passaggio per una determinata posizione di una pendola).

È ovvio che dal punto di vista concettuale esistono notevoli differenze tra le due tipologie di misura: l'analogica consente di ottenere un valore idealmente accurato, cioè con tante cifre significative quante sono consentite dalla sensibilità, o accuratezza, dello strumento: cambiando strumento, cioè migliorandone la sua sensibilità, è possibile abbattere l'incertezza fino a valori virtualmente trascurabili. La misura digitale fornisce un valore intero, normalmente con un'incertezza pari all'unità: idealmente questa incertezza non dipende dallo strumento di misura (purché idoneo per la misura che si intende eseguire).

Dal punto di vista pratico, anche una qualsiasi misura analogica può dare luogo a un risultato intero (opportunamente dimensionato), per esempio quando essa, codificata in modo opportuno (per esempio scegliendo l'unità di misura in modo che non ci siano cifre significative decimali) viene trascritta su un foglio, o inserita in un computer per il successivo trattamento. La misura digitale, però, ha tali caratteristiche "in natura", e non richiede ulteriori passaggi per essere ricondotta a un numero intero.

A. Digitalizzazione e campionamento

I meccanismi di digitalizzazione, di cui daremo un breve esempio concettuale in seguito, sono inerentemente accompagnati da processi di *campionamento*. Campionare significa prendere un pezzettino, in senso generalmente temporale, della grandezza da misurare, che quindi viene analizzata solo per questo pezzettino. In pratica, la digitalizzazione avviene in un intervallo temporale finito e non nullo, cioè essa richiede del tempo per essere portata a termine.

Come ben sapete, nella descrizione convenzionale di un qualsiasi strumento di misura si fa riferimento alla "prontezza" come alla capacità di seguire e registrare le variazioni temporali della grandezza misurata. Qualsiasi strumento ha una prontezza finita: in un digitalizzatore l'analogo del limite di prontezza è dettato proprio dal campionamento.

Il legame tra digitalizzazione e campionamento è evidente anche in settori apparentemente diversi dalla misura di grandezze elettriche. Facciamo un solo esempio, valido anche per intuire i progressi permessi dai metodi digitali: la riproduzione di musica è notoriamente passata da tecniche analogiche (fonografo e poi giradischi, dove, per intenderci, il movimento della puntina sul solco del disco, opportunamente amplificato con metodi meccanici e poi elettronici, provocava un analogo spostamento della membrana dell'altoparlante, e viceversa per la registrazione) a metodi digitali a tutti voi ben noti. Sono altrettanto noti i vantaggi che se ne sono ricavati, essenzialmente costituiti da un miglioramento della qualità audio (estensione dinamica e in frequenza nettamente più ampie) e dalla drastica riduzione dei disturbi, ovvero rumori, dovuti alla soppressione dei segnali "spurii" generati in fase di registrazione e riproduzione. I più appassionati di musica tra voi sapranno che la capacità dinamica, intesa come *profondità di digitalizzazione* (espressa in bits, come vedremo in seguito) e la rapidità (ovvero *rate o frequenza di campionamento*) sono figure di merito rilevanti nella riproduzione musicale. Esse sono anche caratteristiche distintive nella misura digitale di segnali elettrici qualsiasi.

II. FUNZIONAMENTO DI BASE DI UN DIGITALIZZATORE MODELLO

Grazie alla vasta diffusione della digitalizzazione, sono state messe a punto diverse tecnologie specifiche, molto efficaci e spesso complicate da descrivere. Tuttavia è possibile identificare un meccanismo base, a cui abbiamo già fatto cenno in una precedente nota, in grado di spiegare come sia possibile digitalizzare un segnale analogico. In sostanza, quello che qui descriviamo è il principio di funzionamento di un *convertitore analogico-digitale* (ADC - Analog to Digital Converter, o convertitore A/D), che può essere considerato il cuore di ogni dispositivo che permetta di convertire in numero intero una grandezza elettrica, per esempio una d.d.p., analogica in natura.

Gli ingredienti fondamentali del metodo sono:

1. un *clock*, ovvero un dispositivo in grado di generare impulsi, di durata virtualmente trascurabile, equispaziati nel tempo, così come fa un orologio;

2. un *contatore*, cioè un dispositivo in grado di contare gli impulsi di cui sopra, dotato di un circuito di reset (di azzeramento) opportunamente comandabile;

3. un'*interfaccia digitale*, cioè, per intenderci, un processore in grado di trattare (registrare, visualizzare, etc.) il conteggio prodotto dal contatore creando la parola digitale, anch'esso dotato della possibilità di registrare, visualizzare, etc. in istanti comandabili dall'esterno;

4. un *generatore di rampa*, cioè un dispositivo, opportunamente triggerabile, in grado di fornire una d.d.p. crescente linearmente con il tempo (come il generatore di sweep dell'oscilloscopio, per intenderci);

5. un *comparatore*, cioè un dispositivo in grado di comparare i livelli (le d.d.p., riferite ovviamente alla stessa linea di massa) di due ingressi e fornire in uscita un segnale dipendente dall'esito della comparazione (per intenderci, zero o uno a seconda che prevalga uno o l'altro dei due segnali in ingresso).

La Fig. 1 illustra lo schema a blocchi, semplificato, del sistema [pannello (a)] e mostra uno schemino della temporizzazione (timing) del circuito, ovvero della sequenza temporale delle operazioni da esso compiute.

Per capire il funzionamento di questo digitalizzatore modello, cominciamo subito con il notare che il conteggio degli impulsi di clock è un'operazione inerentemente digitale, dato che il suo esito è sicuramente un numero intero. Agli ingressi del comparatore sono inviati il segnale (d.d.p.) da misurare e l'uscita (d.d.p.) del generatore di rampa. Supponiamo poi che la partenza della rampa, cioè il trigger del generatore che produce la rampa, scatti contemporaneamente al reset del contatore: la parola digitale è dunque zero quando il valore della rampa è pure

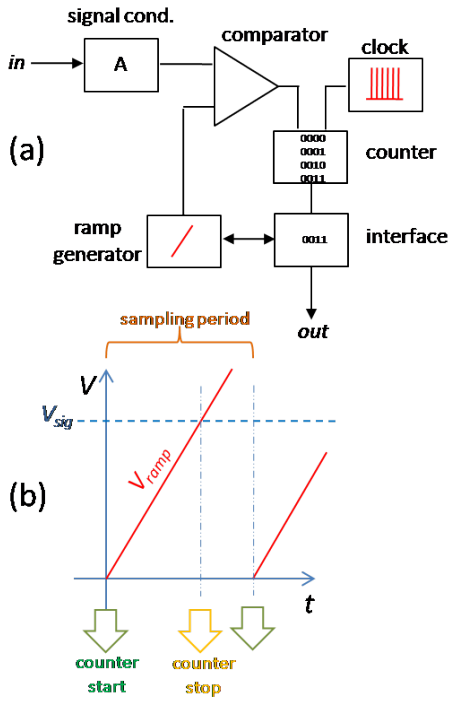

Figura 1. Schema a blocchi semplificato (a) e schema del funzionamento (b) di un digitalizzatore modello; V_{sig} e V_{ramp} indicano rispettivamente la d.d.p. del segnale (incognita da misurare) e la d.d.p. prodotta dal generatore di rampa.

zero (o altro livello noto). Infine, supponiamo che l'uscita del comparatore vada a comandare l'interfaccia digitale: il contatore conta (1, 2, 3,..) fin quando il segnale incognito in ingresso è inferiore alla rampa. Quando invece la rampa supera il segnale incognito, il processo si interrompe e l'interfaccia digitale acquisisce il conteggio. Grazie alla linearità della rampa con il tempo, questo conteggio è proporzionale al valore del segnale in ingresso, che quindi risulta digitalizzato, cioè convertito in un numero intero. Un'opportuna calibrazione consente di ricondurre il conteggio al valore di d.d.p. del segnale incognito entro una determinata incertezza di calibrazione. Inoltre è ovvio che nelle implementazioni pratiche il segnale incognito, prima di arrivare all'ingresso del comparatore, viene "condizionato", cioè opportunamente amplificato o attenuato, eventualmente cambiato di segno e traslato rispetto allo zero. In assenza di questo condizionamento non sarebbe possibile, per esempio, misurare segnali negativi. Anche il circuito di condizionamento soffre di incertezze di calibrazione, delle quali tenere conto nella determinazione dell'errore complessivo di misura. Nelle esperienze pratiche con Arduino non faremo uso di alcun sistema di condizionamento "esterno", e infatti, ad esempio, misureremo sempre delle d.d.p. positive.

La descrizione di questo semplice meccanismo mette in evidenza immediatamente le due prinicipali figure di merito di un digitalizzatore:

- il rate di campionamento, che è evidentemente di-

pendente dalla rapidità di risposta del comparatore e dalla ripidità della rampa prodotta; esso è normalmente espresso in Sa/s (samples per secondo) e qualche volta in Hz (eventi di campionamento per secondo);

- la capacità dinamica, o profondità di digitalizzazione, cioè il massimo numero di conteggi che può essere registrato (ovviamente nel tempo necessario alla misura), normalmente espresso in *bit*, dato che nel mondo digitale vige la codifica binaria.

Entrambe queste figure di merito hanno ripercussioni sulla qualità della misura. Il rate di campionamento influenza direttamente la possibilità di seguire le variazioni di segnali dipendenti dal tempo. La capacità dinamica, o profondità di digitalizzazione, ha a che fare con la portata e la sensibilità della misura. Inoltre è evidente che, a prescindere da ogni altra eventuale causa di errore legata alla calibrazione, esiste un'*incertezza di digitalizzazione* che vale un bit. Infatti la capacità dinamica finita implica che il comparatore scatti quando il segnale si trova all'interno di un intervallo la cui ampiezza è pari a quella di un singolo bit, per cui il valore del segnale può essere determinato solo all'interno di questo intervallo.

Facciamo qualche esempio, mantenendoci nell'ambito degli strumenti di misura. A partire dalla loro prima comparsa nel mercato, intorno a metà degli anni '80, gli oscilloscopi digitali hanno progressivamente soppiantato gli oscilloscopi analogici, come quelli da voi impiegati. Oggi un oscilloscopio digitale economico (più economico dei rari e preziosi strumenti analogici ancora a catalogo) può avere un rate di campionamento (*sampling rate*) di diverse centinaia di MSa/s. Oscilloscopi di classe più elevata campionano a un rate di diversi GSa/s su tutti i canali (tipicamente quattro) di cui sono dotati. Un elevato sampling rate è necessario per seguire l'andamento temporale di segnali che variano rapidamente nel tempo. Per esempio, nel caso di segnali periodici una ricostruzione fedele della forma d'onda richiede che venga acquisito il segnale corrispondente a diversi istanti nell'ambito del periodo. Dunque un sampling rate di 1 GSa/s garantisce un'agevole ricostruzione di segnali periodici che hanno una frequenza dell'ordine di alcune centinaia di MHz (sempre che la *banda passante* dello stadio di ingresso dello strumento lo consenta).

Normalmente la profondità di digitalizzazione per un oscilloscopio è di 8 bit, che significa che sono accessibili $2^8 = 256$ livelli distinti, ovvero che il massimo conteggio può andare da zero a 255 unità. Esistono anche strumenti in cui tale profondità arriva a 12 bit (ovvero $2^{12} = 4096$ livelli) e certamente alcuni dispositivi (per esempio, schede di acquisizione da inserire negli slot dei computer e strumenti simili) permettono di arrivare a 16 bit ($2^{16} = 65536$ livelli), comprensibilmente a spese del rate di campionamento, limitato, in questo caso, alle centinaia di kHz. La risoluzione, ovvero la sensibilità, di un digitalizzatore a 8 bit può non essere particolarmente esaltante. Supponendo di avere un segnale la cui ampiezza massima (se volete, picco-picco) è 10 V, allora la misura fatta a 8 bit avrà un'incertezza di digitalizzazione di 10 V/256 $\simeq$ 40 mV, che sarà dunque la minima variazione che potrà essere letta nella misura (se preferite, tutto questo vuol dire che l'incertezza di digitalizzazione ha un valore relativo dello 0.4%, che non è un dato eccellente per le misure che si possono fare in un laboratorio, specie considerando che ad essa si devono generalmente aggiungere gli errori sistematici di calibrazione).

A. "Simulazione" (tentativa)

I processi di campionamento possono facilmente condurre a degli artefatti, cioè alla comparsa di comportamenti non presenti nel segnale originale. Questo argomento è piuttosto complesso, quindi in questa sede ci limitiamo a fornire un'illustrazione del problema, basandoci su una modellizzazione semplificata e su una sorta di "simulazione" eseguita con Python.

Supponiamo allora di avere un segnale periodico alternato, come quello rappresentato nel pannello superiore di Fig. 2. Per rendere realistica la simulazione, scegliamo per questo segnale una frequenza $f = 1$ kHz e lo mostriamo nell'intervallo temporale $\Delta t = 10$ ms (si osservano 10 periodi di oscillazione). Nella simulazione questo segnale è rappresentato da un array costituito da molti punti, in numero $n = 10^4$ in questo esempio, uniti tra loro da una linea blu. Naturalmente sullo schermo, o sul foglio stampato, tutti questi punti non compaiono, a causa della risoluzione limitata dello schermo, o della stampa. In ogni caso possiamo attribuire a questo segnale (che definiamo "originale") un sampling rate pari a $S = n/\Delta t = 1$ MSa/s, cioè due punti consecutivi dell'array si riferiscono a istanti temporali distanti tra loro per un intervallo $1/S = 1$ μs.

Il processo di campionamento produce in primo luogo una *decimazione* della rappresentazione del segnale, cioè l'array campionato (quello che deriva dal processo di campionamento) ha un numero di punti inferiore rispetto all'originale e due punti consecutivi corrispondono a istanti temporali distanti di più che non nell'originale. Per rendere realistica la simulazione, poniamo come valori dell'array campionato quelli che si ottengono mediando il segnale originale sulla durata dell'intervallo di campionamento, cioè si fa una *media mobile* sul segnale originale seguita dalla decimazione. Notiamo che questa operazione non è sempre necessariamente giustificata, dato che i dettagli dell'(inevitabile) operazione di media temporale condotta nell'intervallo di campionamento dipendono dallo specifico circuito di digitalizzazione impiegato.

I pannelli inferiori di Fig. 2 mostrano il risultato dell'operazione per alcune scelte di sampling rates (indicate in figura). I markers rappresentano i valori degli array ottenuti per campionamento e le linee rosse sono solo una guida per l'occhio. Si vedono chiaramente i deleteri effetti del *sotto-campionamento*: a 2 kSa/s l'array campionato è costituito solo da 20 punti, come si può facilmente veri-

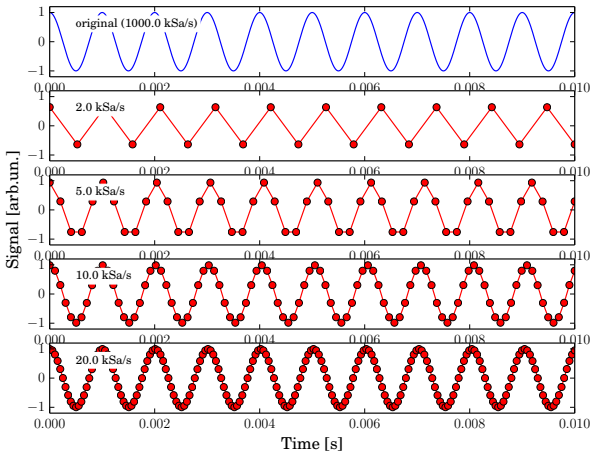

Figura 2. "Simulazione" del processo di campionamento come descritto nel testo. Il pannello superiore mostra il segnale "originale", quelli inferiori le onde ricostruite per campionamento con il rate indicato in figura (le linee continue rosse sono solo una guida per gli occhi). Grafici ottenuti con Python.

ficare, e l'onda originariamente sinusoidale appare triangolare. Artefatti (distorsioni spurie della forma d'onda) si osservano anche a 5 kSa/s e, in maniera meno evidente, a 10 kSa/s. Finalmente a 20 kSa/s la forma d'onda ottenuta appare ragionevolmente simile all'originale, almeno per l'accuratezza della stampa o della visione sul display del computer. Dunque campionare con un rate troppo basso rispetto alla frequenza dell'onda originale può condurre a seri problemi di interpretazione.

Notate che questo esempio si riferisce a una situazione particolarmente semplice, quella in cui la forma d'onda originale è periodica. Potete facilmente immaginare la tipologia di problemi che possono verificarsi nel caso di segnali non periodici. Tutto questo fa capire che occorre cautela nell'impiegare sistemi di acquisizione digitali, cautela che in genere non è richiesta con strumenti analogici, come gli oscilloscopi che usate in laboratorio didattico.

III. ARDUINO

Nelle nostre esperienze di laboratorio abbiamo a disposizione strumenti analogici e digitali. Per esempio, avete tutti usato il multimetro digitale, che, di fatto, è un digitalizzatore unito a diversi componenti che servono per rendere possibili le varie misure di cui è capace, condizionando opportunamente il segnale, o la grandezza, in ingresso. Le caratteristiche intrinseche del digitalizzatore non sono note (non sono riportate nel manuale). Poichè la lettura del display è di 3 digits e mezzo, cioè i "conteggi" della lettura possono andare da zero a 1999, si può azzardare che la profondità di digitalizzazione sia di 11 bit ($2^{11} = 2048$ livelli). Per quanto riguarda il rate di campionamento, si può notare che la prontezza dello strumento è molto bassa, dato che il display viene rinfrescato con un rate dell'ordine della frazione di Hz (circa un paio di letture al secondo). Tuttavia è presumibile che questo rate sia dovuto essenzialmente alla parte di circuito che gestisce il display (un rate di refresh più rapido renderebbe difficile fare letture) e che il rate di campionamento del digitalizzatore sia più elevato, probabilmente dell'ordine del kHz o delle decine di kHz. Le tante digitalizzazioni eseguite nel tempo necessario per rinfrescare il display dovrebbero essere mediate in modo da aumentare la precisione dello strumento, che alla fine risulta limitata essenzialmente dall'incertezza di calibrazione.

Altri strumenti digitali che avete usato sono i frequenzimetri (quello del generatore di forme d'onda e quello dell'oscilloscopio). In questi casi il carattere digitale è scontato, dato che la misura di frequenza di una forma d'onda presuppone naturalmente un'operazione di conteggio (quanti picchi dell'onda si contano in un determinato intervallo di tempo). Il frequenzimetro, infatti, è essenzialmente un contatore, così come è stato definito prima, che conta per un determinato intervallo di tempo (*tempo di gate*).

Il multimetro a lancetta è invece uno strumento evidentemente analogico, così come analogico è l'oscilloscopio da voi impiegato. Dal punto di vista "didattico" (e non solo) gli oscilloscopi analogici sono nettamente preferibili a quelli digitali, però essi hanno l'evidente limite di non poter acquisire in modo automatico delle forme d'onda, o dei segnali comunque dipendenti dal tempo. Questo è invece di estrema utilità quando si intende studiare nei dettagli un fenomeno transiente, per esempio la carica/scarica di un condensatore. Inoltre, più in generale, l'abilità di acquisire segnali è un preziosissimo aiuto quando l'analisi di un fenomeno richiede di eseguire molte misure, cosa che, fatta manualmente, richiederebbe tempi biblici.

Un modo per risolvere il problema, di amplissima diffusione nei laboratori, è quello di dotarsi di una *scheda di acquisizione*, cioè di un dispositivo di input-output collegato a un computer. Nelle nostre esperienze, questo dispositivo è una scheda Arduino Uno collegata via porta seriale (USB) al computer. Arduino non è certamente la scelta migliore per campionare e digitalizzare un segnale analogico: infatti la scheda nasce per scopi diversi, soprattutto didattici e ludici. Però, pur con delle limitazioni abbastanza evidenti, essa può essere convenientemente impiegata per i nostri scopi.

Senza entrare troppo nei dettagli, il cuore di una scheda Arduino è un microcontroller che ha la possibilità di gestire, attraverso istruzioni di alto livello (cioè scritte in un linguaggio facilmente comprensibile), un certo numero di ingressi analogici e digitali, e un certo numero di uscite digitali (possono valere uno zero o un uno digitale, cioè normalmente 0 o 5 V). Per i nostri scopi di base, questo significa che, inviando un segnale analogico a uno dei piedini, o porte, della scheda (ovviamente il segnale deve essere riferito alla massa della scheda stessa), è possibile

istruire il microcontroller in modo da campionare e digitalizzare il segnale, rendendolo disponibile al computer sotto forma di numero intero, ovvero parola digitale.

L'istruzione del microcontroller avviene via software: poche righe di programma, normalmente scritte in un ambiente specifico e denominate *sketch*, vengono preliminarmente trasferite alla memoria interna del microcontroller (una memoria flash da 32 kB) attraverso il collegamento seriale (USB). Qui vengono interpretate trasformandosi nei dovuti ordini impartiti al microcontroller. Inoltre lo sketch istruisce il microcontroller a trasferire il risultato della digitalizzazione, cioè il numero intero corrispondente alla misura, alla piccolissima memoria RAM interna al microcontroller (si parla di una memoria da 2 kB) e da qui, tramite la connessione seriale, al computer. Alla fine, nel computer si ritrovano delle variabili (numeri interi) che possono essere salvate su files contenenti le misure eseguite, che possono essere facilmente analizzate e trattate via software, per esempio con Python.

Le caratteristiche specifiche delle misure e delle istruzioni necessarie a eseguirle saranno descritte a corredo delle varie esperienze. In questa nota ci limitiamo e esaminare alcune proprietà generali della scheda Arduino Uno.

A. Caratteristiche principali di Arduino

Cominciamo con il notare che il microcontroller a bordo della scheda Arduino Uno lavora a una *frequenza di clock* di 16 MHz. Questo vuol dire che, (molto) in linea di principio, esso può compiere fino a $(16/2) \times 10^6$ operazioni digitali (binarie) al secondo e non ha nulla a che vedere con il rate del digitalizzatore, il cui funzionamento non corrisponde certamente a una singola operazione binaria. In termini assoluti, la frequenza di clock di Arduino non dice molto, in termini relativi tutti sapete che i processori dei vostri computer, o dei vostri telefonini, girano a frequenze di clock anche due ordini di grandezza superiori, ma il processore installato nell'Apollo 11 aveva un clock di soli 2 MHz.

Leggendo la scheda tecnica di Arduino Uno si vede che sono disponibili 6 distinti ingressi analogici (marcate come porte `A0-A5`) in grado di accettare d.d.p. positive da zero a un valore massimo di riferimento V_{ref}. Questo valore di riferimento dipende, nelle nostre applicazioni, dalla tensione di alimentazione della scheda che, sempre nelle nostre applicazioni, proviene dal computer tramite la porta USB. Quello che si ottiene è normalmente $V_{ref} \sim 5$ V. La dinamica di digitalizzazione è 10 bit, per cui sono porissibili $2^{10} = 1024$ livelli distinti; dunque la risoluzione, o incertezza di misura, è tipicamente di circa 5 V/1024 $\simeq$ 5 mV. Notate che l'errore di calibrazione, che dipende dal valore effettivo della tensione di riferimento usata nella rampa del comparatore, non è noto: quando non sarà strettamente necessario per le esigenze della misura, trascureremo del tutto l'errore di calibrazione, rappresentando i dati in *unità arbitrarie di digitalizzazione*. Questo significa che il risultato della digitalizzazione sarà espresso in numeri interi, da 0 a 1023. La possibilità di eseguire in automatico tante misure può consentire, in linea di principio, di stimare direttamente l'incertezza dallo scarto quadratico medio del campione di misure.

Il rate di campionamento è, invece, una caratteristica non facilmente individuabile nella scheda tecnica. Infatti Arduino non è nato come digitalizzatore, per cui questa informazione non compare in maniera immediata. Purtroppo, poi, il rate di campionamento dipende anche dalla rapidità con cui le parole digitali già formate (le misure già effettuate) sono trasferite al computer. Nelle normali schede di acquisizione digitale, questa fase è resa più rapida grazie alla presenza di un grande spazio di memoria RAM che può accogliere momentaneamente le parole che poi vengono trasferite al computer (si dice che la memoria funziona da *buffer*). In Arduino Uno, la memoria destinata a questo scopo è davvero poca, e quindi si può formare un collo di bottiglia che "intasa" la digitalizzazione.

Un valore ragionevole per il *massimo* rate di campionamento (per la lettura di un singolo ingresso) è dell'ordine di 10 kSa/s[2]. L'intervallo di campionamento minimo, che dunque risulta dell'ordine di 100 μs, o un po' minore, è adeguato solo per seguire l'andamento temporale di segnali che variano piuttosto lentamente, sulla scala dei ms, o superiore.

Oltre agli ingressi analogici, Arduino ha la possibilità di controllare 14 porte digitali. Queste porte possono assumere o leggere un valore binario (zero o uno) a seconda che siano a livello "basso" (per esempio, 0 V) o "alto" (tipicamente oltre 3.5 V). Infatti il significato dell'aggettivo digitale ad esse attribuito significa proprio che esse hanno un comportamento binario.

Sei tra queste porte digitali possono operare come uscite in una modalità detta `PWM` (Pulse-Width Modulation). In questa modalità le porte generano treni di impulsi a frequenza fissa (e piuttosto bassa, cioè minore di 1 kHz) con duty cycle regolabile via software (può essere aggiustato su 8 bit, cioè 256 distinti valori). Supponendo di collegare a queste uscite un circuito integratore con frequenza di taglio opportuna, la regolazione del duty cycle del treno di impulsi permette di "simulare" un'uscita analogica (regolabile tra qualcosa di prossimo a 0 V e qualcosa di prossimo a V_{ref}), essendo la d.d.p. ai capi del condensatore proporzionale alla frazione di tempo in cui l'impulso rimane al suo livello "alto": potete facilmente farvene una ragione pensando al comportamento di un integratore RC.

Ci sono poi vari altri aspetti rilevanti per il funzionamento e l'uso di Arduino, di cui ci occuperemo in sede di presentazione e discussione delle singole esperienze.

[1] Naturalmente ci sono importantissimi controesempi a questa affermazione, dovuti in particolare alla natura discreta della carica elettrica. Provate per esempio a trovare il legame tra d.d.p. e carica accumulata in un condensatore di capacità piccolissima, dell'ordine dell'aF. Tuttavia le grandezze di interesse pratico nel mondo macroscopico hanno sicuramente un carattere non discreto.

[2] Sampling rates più alti sono riportati in rete per alcune applicazioni specifiche, basate su sketch ottimizzati. In ogni caso, il microcontroller di cui è dotato Arduino Uno impiega almeno 16 μs per eseguire una singola digitalizzazione, valore che pone un limite ultimo nel rate di campionamento.