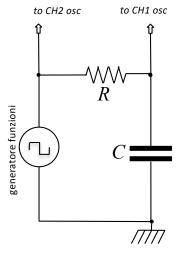
Ε	2400	5.0 -	aa	24	/2!
---	------	-------	----	----	-----

			_	
Nomi e Cognomi:	□мак Data:	□MER	□GIO	5

Carica/scarica del condensatore e Arduino

Questa esercitazione ha come finalità principale lo studio del processo di carica/scarica di un condensatore e come finalità accessoria l'impiego di Arduino Due come campionatore e digitalizzatore, sia per la carica/scarica del condensatore che per l'acquisizione di forme d'onda periodiche (dati che userete in futuro, se non li avete già acquisiti nella scorsa esercitazione!).

1. Come primo esperimento, dimensionate (per la lettura della capacità nominale dei condensatori vedi nota a fine scheda) e realizzate il circuito di figura, in cui una forma d'onda quadra, che inizialmente sceglierete alternata e di ampiezza come vi pare, viene usata per caricare e scaricare periodicamente il condensatore C attraverso la resistenza R. Eseguite delle osservazioni sul circuito, magari in funzione della frequenza f del generatore: con un minimo di skills, dovreste riuscire a stimare il tempo caratteristico τ di carica e scarica. Per "stima" si intende qui una misura senza incertezza, che potrete fare in maniera brutale oppure un po' più raffinata (per questo scopo può essere utile impiegare in modo sensato le linee orizzontali 0, 10%, 100% che trovate in graticola e ricordare che $ln(10) \sim 2.3$). Condizione preliminare è che triggeriate per bene l'oscilloscopio (ricordate quanto avete fatto nell'esercitazione precedente!). Riportate le vostre osservazioni nei commenti, cercando di capire per bene cosa succede e cosa si vede ai due canali dell'oscilloscopio, dunque spiegando anche cosa si vede su CH2. Può essere carino anche fare delle osservazioni in modalità X-Y (o Y-X) dell'oscilloscopio, che si attiva premendo un tastino sul pannello, e commentarle brevemente (per questa osservazione vi conviene usare una forma d'onda sinusoidale...).



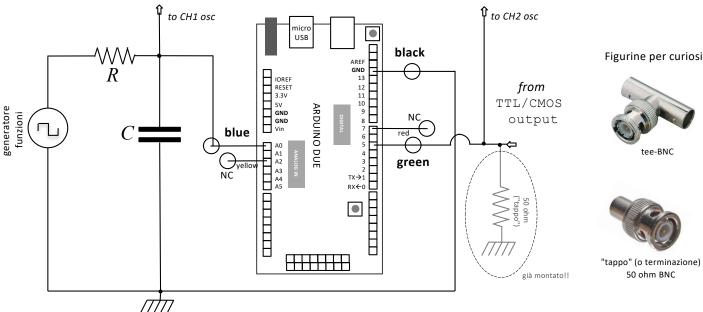
sservazioni, compreso (nza errore!) di vi e come i	a avete fatta, comfonto	con le aspettative,
	1 115		1. 1	1 11 11

2. Questa parte è propedeutica all'impiego di Arduino e davvero necessita di triggerare per bene l'oscilloscopio (magari potete usare il segnale dell'uscita TTL/CMOS output!): giocate con ampiezza e offset del generatore di forme d'onda (agite sulle manopole AMPL e OFFSET, quest'ultima anche da estrarre) e osservate come cambia il segnale in uscita, cioè su CH2. L'obiettivo finale è quello di fare in modo che questo segnale, da misurare in accoppiamento DC, sia compreso tra circa 0 e circa 3.2 V rispetto alla linea di massa, o terra (dunque sempre positivo). Se non siete sicuri di quello che state facendo, chiamateci!

3. Dopo esservi accertati cento volte che il segnale sia come si deve, potete passare alle operazioni preliminari che riguardano Arduino. Per prima cosa accendete il PC e collegateci Arduino tramite cavo USB. Quindi lanciate dall'interno di Pyzo script ardp2.py, che potete trovare nella directory directory /home/studentelab2/Arduini/. Tutto è uguale a quanto avete fatto in una precedente esercitazione!



4. Solo dopo esservi accertati un altro centinaio di volte che il segnale sia ampio come si deve (cioè compreso tra 0 e 3.2 V e sempre positivo), potete collegare Arduino al circuito, secondo lo schema a pagina seguente. Guardate che dovete collegare anche il segnale dell'uscita TTL/CMOS output proveniente dal generatore alla boccola verde di Arduino: per evitare che Arduino Due si bruci (il segnale TTL è troppo ampio), subito all'uscita del generatore troverete una tee-BNC e un "tappo" da 50 ohm, usato per portare il livello TTL a valori compatibili con Arduino Due: tutto questo è già stato fatto e voi non dovete fare nulla (ma magari guardare come è fatto il tutto può essere carino e utile). Notate che potete usare il CH2 dell'oscilloscopio per visualizzare il segnale TTL/CMOS output e usarlo convenientemente come trigger per l'oscilloscopio. Arduino ha cinque boccole volanti colorate, tre delle quali vanno collegate come in figura.



- 5. A questo punto siete pronti per l'acquisizione dati: una volta lanciato, lo script ardp2.py presenta un pannello in cui possono essere scelti: (i) il nome del file che raccoglierà i dati acquisiti (occhio che riscrive senza avvisare!); (ii) l'intervallo nominale Δt_{nom} tra un campionamento e il successivo (regolabile in passi da 10 μs fino a 9990 μs); (iii) il numero totale di punti campionati (ovvero la lunghezza del record, che può essere scelta in potenze di 2, da 128 a 8192). Inoltre il pannello consente di selezionare tra acquisizioni asincrone e sincrone (il sincronismo può avvenire a scelta con la slope positiva o negativa del segnale inviato alla porta digitale) e, solo nel caso di acquisizioni sincrone e, con qualche limitazione di lunghezza del record, di eseguire medie su un certo numero di eventi (da 8 a 128, nella implementazione attuale); il calcolo della media è accompagnato dalla stima della sua incertezza (standard deviation della media), che viene anche acquisita.
- 6. Potete scegliere liberamente se usare dati mediati su diversi eventi, o no (magari potete esplorare tutte e due le opzioni!). Ricordate che il file prodotto, di default registrato nella directory /home/studentelab2/dati_arduino/, ha un numero di colonne che dipende dalla configurazione prescelta. In particolare, per acquisizioni singole ("single sweep"), le colonne sono due e contengono: tempo in μs, d.d.p. in digit (da 0 a 4095); per quelle mediate le colonne sono quattro e contengono: tempo in μs, sua incertezza in μs, d.d.p. in digit (da 0 a 4095), sua incertezza in digit. Osservate che, grazie al processo di media, i dati digitalizzati sono espressi come numeri reali (con due cifre decimali) e non come interi. Notate anche che il pannello di controllo ha un check box che, quando cliccato preventivamente, permette di visualizzare immediatamente dati con barre di errore. Tenete conto che, nel caso di acquisizioni singole, dovete scegliere voi le incertezze "convenzionali" da attribuire (fate riferimento all'esercitazione 3).
- 7. Poiché vi servirete dei dati acquisiti per fare dei grafici e best-fit, dovete fare in modo che le acquisizioni siano "sensate" e "belle" (insomma, il meglio che si può!). Qualora il dimensionamento di R e C che avete fatto in precedenza non vi soddisfi, potete, anzi dovete, modificarlo. L'obiettivo sarebbe quello di analizzare sia carica che scarica: potete farlo con un solo record o (è certamente più semplice) con due record distinti. Poiché siete interessati a determinare i tempi caratteristici, non è necessario, anzi è concettualmente sbagliato, convertire i dati digitalizzati in unità fisiche di d.d.p.: tenete tuttavia conto della possibile presenza di un piccolo offset nel digitalizzatore di Arduino, quindi scrivete in modo opportuno la funzione modello di scarica, aggiungendo un termine costante (per la carica l'offset è già "compreso"!). Tenete anche conto che, per gli scopi del best-fit, sarebbe sufficiente acquisire dati su un "breve" intervallo temporale, comprendente solo la fase di scarica o carica. Però per l'estetica prendete anche un pochino della parte asintotica. Scrivete tutti i risultati del best-fit (stile Lab2!!) e tutte le informazioni necessarie nel riquadro dei commenti.Page 2 of 4

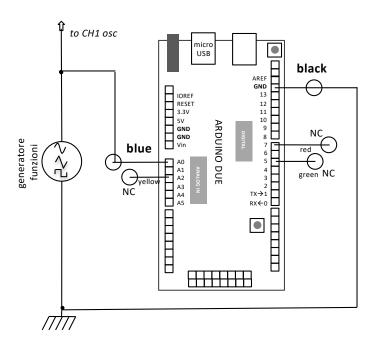
E 23005.1 - aa 23/24

	Nomi e Cognomi:	□mar Data:	MER	□GIO	5'	
C	Commenti, risultati e info del best-fit (tutti, in stile Lab2!):					

- 8. <u>Facoltativo</u>: dato che l'obiettivo del best-fit potrebbe essere quello di avere una misura indiretta della capacità C e che per questo scopo è utile avere una misura accurata di r_G (resistenza interna del generatore, che il manuale dice essere 50 ohm con tolleranza \pm 10%), siete consigliati di eseguire una misura a la Thévenin di r_G , in modo da poter aumentare l'accuratezza nella misura indiretta di C.
- 9. Sempre per lo stesso scopo, potrete anche utilizzare un multimetro un po' più accurato (ce ne è uno solo e lo monteremo su una cattedra) per determinare *R* con buona accuratezza.
- 10. <u>Facoltativo</u> (se è il caso): è possibile che, nonostante l'impiego di acquisizioni mediate, gli effetti di spikes o altre fluttuazioni raccolte da Arduino determinino la presenza di outliers (cioè dati che palesemente sono fuori dalle aspettative e la cui presenza può ragionevolmente essere attribuita a eventi casuali sporadici). A casa e con calma, potete provare a vedere se ci sono outliers evidenti ed eventualmente scrivere un pezzetto di codice che li individua, secondo un qualche criterio arbitrario, e li elimina dal best-fit (naturalmente nel grafico devono rimanere e essere indicati con chiarezza, dato che i dati sono sacri!).
- 11. <u>Facoltativo (ma molto rapido!</u>): potreste anche fare delle acquisizioni della d.d.p. a monte della resistenza *R*, cioè del segnale che avete osservato con l'oscilloscopio su CH2 in fase preliminare. A casa potreste cercare di fare un best-fit anche di questo segnale, sempre che, per il dimensionamento che avete scelto, esso risulti rilevante. Nel caso, prima di collegare l'ingresso di Arduino (boccola blu) al segnale in questione, <u>controllate attentamente</u> che esso rispetti i requisiti di ampiezza necessari per Arduino Due (compreso tra 0 e 3.2 V, circa), altrimenti aggiustate il generatore di conseguenza. <u>Facoltativo (ma molto rapido e potenzialmente interessante)</u>: potreste provare a guardare all'oscilloscopio e acquisire con Arduino la d.d.p. ai capi del condensatore (il circuito è sempre quello dello schema precedente) selezionando forme d'onda del generatore triangolari e sinusoidali. Se vi sentite tanto massicci con la matematica, trovare una funzione modello adatta e fare best-fit (naturalmente a casa) potrebbe essere carino!



12. La parte che segue non è facoltativa, anche se magari l'avete già svolta nella precedente esercitazione. Tenete conto che i dati che acquisite ora potranno farvi comodo per un lavoro (obbligatorio!) da svolgere nel secondo semestre: prima acquisite 'sti dati e meglio è! Mi raccomando: quando lo fate, mettete i dati al sicuro in una vostra chiavetta, dato che i dischi dei PC vengono periodicamente brasati. Per questa parte dell'esercitazione dovete semplicemente collegare il generatore di forme d'onda all'ingresso di Arduino, come nello schema di pagina seguente, facendo ovviamente in modo che l'ampiezza della forma d'onda prescelta sia sempre compresa tra 0 e circa 3.2 V . Poiché le acquisizioni non devono essere necessariamente sincronizzate, potete scollegare da Arduino l'uscita TTL/CMOS output del generatore (ricordate anche di agire di conseguenza nel panello dello script). Prima di prendere i dati che servono, siete invitati a giocare con frequenza della forma d'onda, intervallo di campionamento, lunghezza del record, per vedere l'effetto che fa, cercando anche di riprodurre condizioni che corrispondono ad acquisizioni volutamente fatte male (sovra- o sotto-campionate, in gergo).



Schema minimale: nulla toglie che continuiate a usare il segnale di sincronismo come nello schema precedente!

13. Tutto questo vi dà la possibilità di acquisire dei record fatti per bene di varie forme d'onda (sinusoidale, quadra, triangolare, comprese versioni rese asimmetriche agendo sulla manopola del duty-cycle del generatore). Acquisite alcune di queste forme d'onda, possibilmente scegliendo record lunghi (almeno 4096 punti) e facendo in modo di registrare parecchi periodi (almeno una decina, meglio una ventina). Come già accennato, questi files saranno vitali per svolgere un lavoro obbligatorio previsto per il secondo semestre, dunque acquisite meglio che potete e conservate con cura i dati!

Nota: lettura della capacità nominale dei condensatori (a carta, o poliestere) disponibili in Lab Did

Ci sono due tipi di condensatori, colorati (gialli e verdi) e grigi. Per i primi la lettura di capacità e tolleranza è immediata, essendo tutto scritto esplicitamente. Per I condensatori grigi, la capacità in μ F è espressa dal numero che precede la lettera "M" riportata sul corpo dei condensatori. Occhio: M non sta per MegaFarad, ma indica piuttosto la tolleranza (la lettera M corrisponde a \pm 20%)!