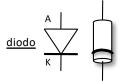
Nome e Cognome:	□мак Data:	□MER	□GIO	8

## Diodo e curva caratteristica *I-V* (con Arduino)

L'esercitazione prevede di impiegare e analizzare un diodo al silicio a giunzione bipolare p-n. Simbolo circuitale e indicazione di come è fatto il diodo li trovate nella figurina qui a fianco



1. Prima di inizare con l'esercitazione "vera e propria" testate la funzionalità del diodo usando il multimetro digitale configurato come ohmetro <u>usando diverse portate</u>. Commentate qui sotto le vostre osservazioni <u>interpretandole sulla base del modello del multimetro digitale usato come ohmetro e di quanto sapete del funzionamento del diodo</u>: spiegate anche e soprattutto <u>l'andamento della resistenza misurata con il fondo scala!</u>

l	Commenti sul "test" del diodo con l'ohmetro (a diverse portate) e <u>spiegazioni</u> di quanto si osserva (ricordate il modello dell'ohmetro digitale!):		
l			
l			
l			
l			
l			

L'esercitazione "vera e propria" consiste nella registrazione della curva caratteristica del diodo e nella sua analisi. Userete Arduino in due modalità diverse: la prima è obbligatoria, la seconda è riservata a chi vuole. Osservazione importante: Arduino ha bisogno di uno sketch specifico per operare nell'una e nell'altra modalità. I nomi degli sketch sono, rispettivamente: ardp2.ino (è lo sketch che avete già usato tante volte!) e diodo2016.ino (uno sketch specifico per la seconda modalità dell'esperimento). I due sketch vanno impiegati con gli script di Python che hanno lo stesso nome (ardp2.py e diodo2016.py), altrimenti, ovviamente, la comunicazione non funziona.

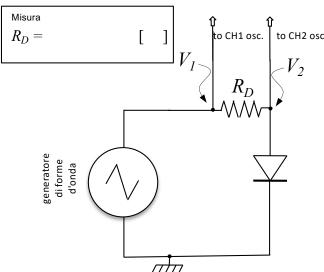
In genere avete sempre trovato lo sketch ardp2.ino già caricato nel vostro Arduino. Per questa esercitazione <u>può essere che sia necessario caricarlo</u>, ovvero caricare lo sketch diodo2016.ino nel caso desideriate fare l'esperimento anche con la seconda modalità. Le informazioni qui sotto servono per gestire la procedura di upload dello sketch su Arduino (chiedete in caso di dubbi!):

- a. Collegare Arduino al PC;
- b. Lanciare la app Arduino IDE (icona con un infinito): l'interfaccia dovrebbe già essere configurata in modo corretto (Arduino board, serial port, etc.), caso mai chiedete;
- c. Aprire il file dello sketch desiderato da dentro Arduino IDE (il doppio click non funziona quasi mai!): gli sketch si trovano nella directory Arduini della root, ma tenete presente che essi sono contenuti all'interno di sotto-directory che hanno lo stesso nome degli sketch stessi;
- d. Eseguire l'upload premendo sul tasto dedicato;
- e. Se non funziona, come spesso succede, chiedete aiuto!!!

## PRIMA MODALITÀ (OBBLIGATORIA PER TUTTE E TUTTI!)

La ricostruzione della curva caratteristica richiede di inviare al diodo una d.d.p. che varia, di misurare questa d.d.p. e, simultaneamente (se possibile!), misurare una d.d.p. proporzionale alla corrente che circola nel diodo. In questa modalità la d.d.p. che varia è prodotta dal generatore di forme d'onda configurato per creare onde triangolari.

- Regolate il generatore di forme d'onda in modo che l'onda triangolare prodotta sia compresa tra 0 e circa 3.2 V (mi raccomando: non vuol dire solo che l'ampiezza della forma d'onda è minore di 3.2 V, ormai lo sapete!).
- Come operazione preliminare, dovete acquisire il segnale prodotto dal generatore con Arduino, usando il solito script ardp2.py (è ovviamente necessario che lo sketch di uguale nome sia stato caricato su Arduino, altrimenti non funziona...). Spero che tutti ricordiate come si fa e che non abbiate bisogno di uno schema, dopo un semestre di Lab2! In sostanza dovete collegare la boccola nera di Arduino alla massa, o terra, del generatore, la boccola blu alla boccola rossa del generatore, la boccola verde al segnale che proviene dalla uscita TTL/CMOS Output del generatore (il segnale, non la terra, capito?).
- L'acquisizione deve essere sincrona e si tratta di un'operazione preliminare poiché serve solo per aggiustare frequenza del generatore e tempo di campionamento di Arduino allo scopo di registrare solamente, e in modo completo, un unico tratto (possibilmente quello di salita, dipende dalla scelta della slope) della forma d'onda triangolare. In sostanza, dovete aggiustare i parametri liberi (frequenza del generatore e tempo di campionamento) in modo che la traccia registrata, contenente un certo numero di punti (non ha senso andare oltre 512 punti), parta da un livello di circa 0 digit e arrivi a un po' meno di 4095 digit, disegnando una rampa, cioè un unico segmento.
- A questo punto realizzate il complicatissimo circuito di figura, nel quale si impiega una resistenza  $R_D$  il cui valore va scelto cum grano salis: io consiglio qualcosa di compreso tra 0.33 e 3.3 kohm, valori nominali. I due segnali indicati con  $V_1$  e  $V_2$  vi serviranno al punto successivo, ma intanto, magari, guardateli all'oscilloscopio. Ricordate di riportare nel riquadro la misura di  $R_D$ .



- Ora dovete registrare in due acquisizioni diverse (purtroppo lo sketch non consente l'acquisizione simultanea di più canali) i segnali indicati con V<sub>1</sub> e V<sub>2</sub> : questi segnali vanno registrati su file, quindi mettete dei nomi ragionevoli che vi permettano di distinguerli tra loro! Per l'acquisizione potete scegliere se mediare su diverse sweep, oppure no: come sapete, la differenza sta nell'errore, che nel caso non mediato dovete mettere voi (per esempio ± 1 digit) e nel caso mediato viene determinato dal software (errore sulla media) e scritto su una colonna del file prodotto. Mi raccomando: fra un'acquisizione e la successiva non cambiate nulla se non il collegamento alla boccola blu di Arduino (una volta per il segnale  $V_I$  e la volta successiva per  $V_2$ , o viceversa). Ovviamente l'acquisizione si fa usando i collegamenti di cui al punto 3: le boccola nera e verde dovrebbero essere già collegate, per cui vi basta agire sulla blu.
- Poiché nell'analisi dei dati dovrete impiegare dati registrati in unità fisiche (V), è necessario che calibriate Arduino. Si consiglia di impiegare la "calibrazione alternativa": allo scopo, come sapete, è sufficiente che misuriate la d.d.p.  $V_{pin7}$ presente tra boccola rossa (collegata alla porta 7 di Arduino) e linea di massa, o terra, con il multimetro digitale. Supponendo linearità e sapendo che il digitalizzatore ha una dinamica di 12 bit (massimo livello 4095 digit), potete determinare il coefficiente di calibrazione ξ. Scrivete i dati rilevanti nei riquadri qui sotto e ricordate che, come sempre, la calibrazione deve essere affetta da una debita incertezza.

$V_{pin7} =$	[ ] ξ=	[mV/digit]
--------------	--------	------------

- Siete finalmente pronti per l'analisi dei dati. Avendo a disposizione  $V_1$  e  $V_2$ , e conoscendo  $R_D$ , potete ricostruire l'intensità della corrente I che fluisce nel diodo. Quindi potete realizzare il grafico della curva caratteristica I vs V, che dovete rappresentare nelle debite unità fisiche (A e V , o loro sottomultipli) tenendo conto della calibrazione. È anche fondamentale che determiniate in maniera corretta le barre di errore: occhio, nessuna barra di errore può essere nulla, altrimenti il fit che dovrete eseguire convergerà molto difficilmente! Magari, stavolta, usate tutte le vostre conoscenze per realizzare una propagazione dell'errore sensata!
- Eseguite infine un best-fit dei dati usando, come funzione modello a due parametri, l'equazione di S. (se non la ricordate, la scriveremo da qualche parte), riportando nei commenti tutti i risultati e le informazioni rilevanti.
- 10. Bonus track : magari potreste chiederci un qualche metodo per riscaldare il diodo e rifare le misure con la giunzione riscaldata (giusto se qualcuno ha voglia di farlo, non si può fare per tutti!)

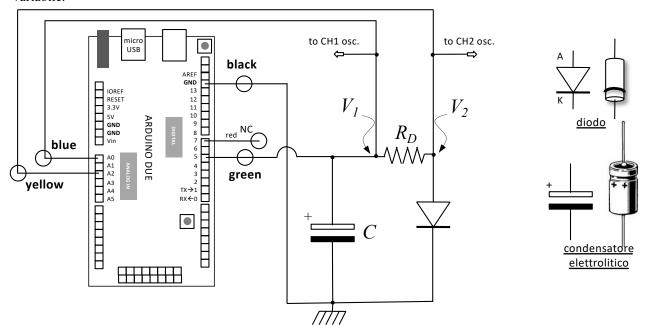
Commenti sul best-fit e risultati ( <u>tutti</u> ):	Funzione modello $I(V)=% {\displaystyle\int\limits_{V}^{V}} {\displaystyle\int\limits$	
	Page 2 of 3	

Nome e Cognome:	□MAR □MER □GIO  Data:	8')
Segue commenti sul best-fit:		

## SECONDA MODALITÀ (SOLO PER CHI VUOLE)

Nella seconda modalità la d.d.p. che varia è prodotta direttamente da Arduino, che si occupa anche di misurare entrambi i segnali  $V_I$  e  $V_2$ , stavolta in modo simultaneo. Poiché Arduino non dispone di uscite analogiche adeguate per l'esperimento, ma ha la possibilità di operare in modalità PWM (sostanzialmente producendo un'onda quadra di frequenza fissa e <u>duty-cycle variabile</u> in maniera digitale su 256 livelli diversi) la d.d.p. che varia è ottenuta livellando con un condensatore tle oind quadra. Potrebbe essere opportuno impiegare un condensatore <u>elettrolitico</u> (per esempio nel range 47-470  $\mu$ F, capacità maggiori di quelle disponibili per i condensatori ordinari a vostra disposizione). <u>Occhio</u>: il condensatore elettrolitico è <u>polarizzato</u>, cioè i suoi due fili <u>non</u> sono invarianti per scambio e il collegamento deve essere eseguito come da schema: vedete che c'è anche una figurina che, grosso modo, fa capire come individuare i due poli del condensatore elettrolitico.

11. Montate lo schema di figura, in cui si impiega la porta pin 5 di Arduino per creare l'onda quadra con duty-cycle variabile.



- 12. Per operare, dovete preliminarmente caricare su Arduino lo sketch diodo2016.ino e utilizzare lo script diodo2016.py Questo script, che <u>non prevede alcuna uscita grafica</u>, permette di inserire il nome del file in cui vengono registrati i dati sotto forma di due colonne, rispettivamente  $V_1$  e  $V_2$ , entrambi da 256 punti. Dovete quindi trattare i dati acquisiti esattamente come nei punti 8 e 9 precedenti (in linea di principio, almeno se non cambiate  $R_D$ , i risultati di questa analisi e della precedente dovrebbero essere molto simili tra loro, vediamo se è vero!).
- 13. Da ultimo, provate a vedere con l'oscilloscopio come sono fatti i segnali  $V_1$  e  $V_2$ , sia quando il circuito è tutto montato, che scollegando il condensatore e, infine, scollegando anche il diodo. Cercate di capirci qualcosa e scrivete i commenti a pagina seguente, assieme a quelli relativi al nuovo best-fit.