

Curva caratteristica di collettore del transistor con Arduino

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

(Dated: version 5 - Lara Palla e Francesco Fuso, 23 febbraio 2017)

Questa nota discute alcuni aspetti di interesse per l'esperienza di registrazione della curva caratteristica di collettore del transistor bipolare, condotta in laboratorio usando Arduino.

I. INTRODUZIONE

L'obiettivo dell'esperienza qui discussa è la ricostruzione della *famiglia di curve caratteristiche di uscita, o di collettore*, del transistor *npn* 2N1711 disponibile in laboratorio. L'acquisizione delle curve, cioè delle coppie di punti I_C, V_{CE} per una data intensità di corrente di base I_B , è eseguita in maniera automatica usando Arduino. L'obiettivo è simile a quello che ci eravamo posti con l'acquisizione della curva caratteristica I vs V del diodo a giunzione, ma questa volta, per scopi prevalentemente didattici, useremo un approccio diverso, che fa uso del generatore di funzioni.

Lo scopo principale dell'esperienza è infatti la realizzazione di un sistema *sincronizzato* di acquisizione dati, una necessità molto frequente quando si richiede l'analisi di segnali generalmente dipendenti dal tempo. Il concetto e le modalità pratiche della sincronizzazione (con Arduino) erano già state introdotte all'epoca dell'acquisizione di segnali periodici, in particolare per la costruzione di record contenenti un gran numero di misure (coppie tempo, d.d.p. digitalizzata). Qui essa viene applicata a un esperimento che ha finalità specifiche. D'altra parte, anche il contenuto "fisico" dell'esperienza ha un suo valore: le tecnologie di costruzione dei transistor prevedono grosse tolleranze nei parametri di funzionamento e conoscere la caratteristica di uscita dello specifico componente che si ha a disposizione può essere molto utile. Infatti, attraverso il metodo della *retta di carico* tale conoscenza permette di stabilire il punto di lavoro effettivo del componente nelle condizioni di operazione, come verificheremo in una prossima esperienza. Inoltre qui potremo valutare il guadagno in corrente β_F (o h_{FE}) del transistor e anche stimare il guadagno β_f (o h_{fe}) per piccoli segnali oscillanti, che sono anche parametri di interesse per l'interpretazione di una futura esperienza.

II. CURVA CARATTERISTICA DI USCITA

La curva caratteristica di uscita, o di collettore, del transistor rappresenta la dipendenza della corrente di collettore I_C in funzione della tensione V_{CE} fra collettore e emettitore, misurata a una certa corrente di base, o di *polarizzazione* di base, I_B . Se I_B viene variato discretamente in un certo intervallo, allora si ricostruisce una *famiglia di curve*, ognuna per una certa I_B .

L'interesse pratico generale, e nostro in particolare, è quello di tracciare curve caratteristiche che rappresentino

il funzionamento del transistor nel *regime attivo*, o nei suoi "pressi". Questo vuol dire che

- la giunzione base-emettitore deve essere polarizzabile in modo diretto, cioè occorre che la base si trovi a potenziale positivo rispetto all'emettitore (per un transistor *npn* come il 2N1711) e che questo potenziale possa arrivare al valore di soglia V_{thr} , o superarlo [1];
- è opportuno esplorare valori di I_B che siano sufficientemente piccoli (nel nostro caso fino a *poche decine di μA*) per evitare che il transistor lavori prevalentemente in *regime di saturazione*;
- di conseguenza la d.d.p. V_{BE} tra base ed emettitore dovrà essere esplorata in un range che al massimo raggiungerà, o andrà poco oltre, la tensione di soglia V_{thr} ; dal punto di vista pratico, questo si può tradurre nella richiesta $0 < V_{BE} \lesssim 0.55 - 0.75 \text{ V}$.

Il transistor disponibile in laboratorio è montato su una basetta, alloggiata in un telaio con un certo numero di boccole. Oltre al transistor, che è contattato con uno zoccolo (il montaggio è molto delicato), la basetta ospita un resistore $R_B = 560 \text{ ohm}$ nominali collegato in serie alla base, la cui presenza è praticamente irrilevante per i nostri scopi. Inoltre il collettore è collegato a due resistori $R_C = 1 \text{ kohm}$ e $R_C = 2.2 \text{ kohm}$ (nominali), che possono essere selezionati alternativamente, a seconda della boccia impiegata, e giocano il ruolo di *resistenza di collettore*. La basetta, che va osservata con attenzione per capirne i collegamenti, ha in definitiva lo schema rappresentato nel box tratteggiato di Fig. 1(a).

Per eseguire la misura, impieghiamo una configurazione *a emettitore comune*, in cui l'emettitore è collegato alla linea di riferimento dei potenziali, ovvero la linea di massa, o di terra. Per eseguire la misura c'è poi bisogno di prevedere due distinti sotto-circuiti, uno dedicato alla polarizzazione di base, cioè a fornire la d.d.p. V_{BE} necessaria per far lavorare il transistor attorno al regime attivo, e l'altro dedicato a fornire la d.d.p. V_{CE} (oltre che permettere la misura di I_C).

Dal punto di vista concettuale, il circuito mostrato in Fig. 1(a) soddisfa queste richieste. La parte di sinistra del circuito ipotizza un generatore di d.d.p. variabile V_{BE} e la presenza di un amperometro per monitorare il valore di I_B . Quella di destra ipotizza un generatore di d.d.p. variabile, qui chiamata V_{CC} , che controlla la d.d.p. V_{CE} e dunque la polarizzazione della giunzione

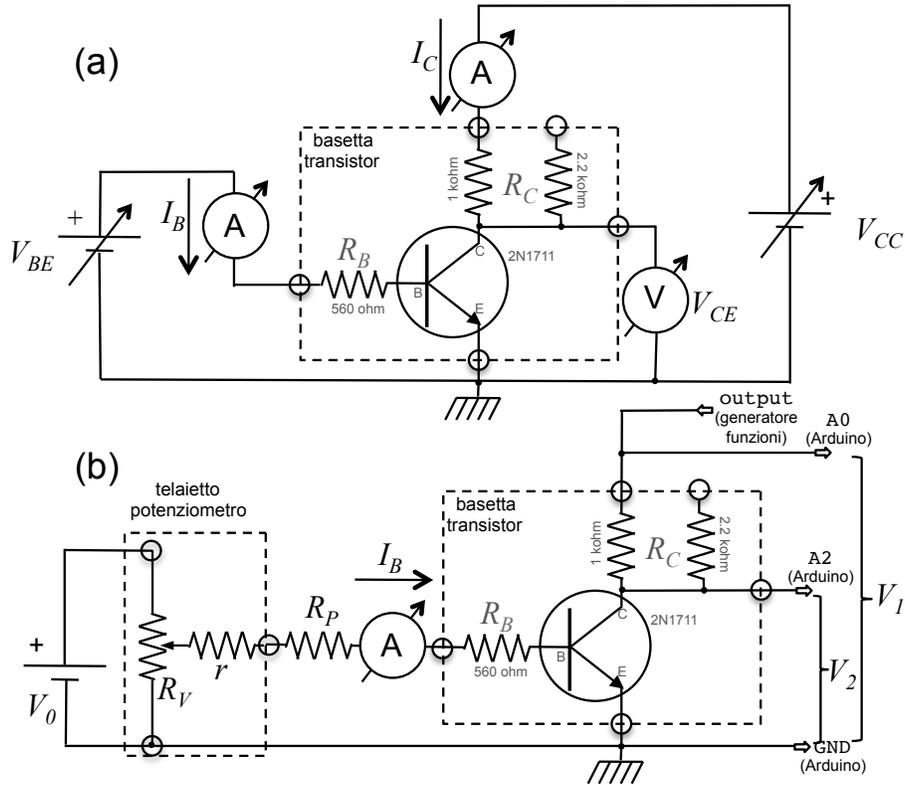


Figura 1. Schema concettuale di un'esperienza di ricostruzione della curva caratteristica I_C vs V_{CE} di un transistor *npn*, del tipo di quelli disponibili in laboratorio (a); implementazione dell'esperienza di ricostruzione automatizzata della curva caratteristica mediante Arduino (b).

collettore-base. Come chiariremo in seguito, la scelta delle polarità e dello schema impiegato garantiscono che la giunzione collettore-base (o base-collettore) sia polarizzabile inversamente, come richiesto per il funzionamento del transistor nel regime attivo. Per la costruzione della curva caratteristica, si può supporre che V_{CC} venga variata, e che per ogni valore prescelto vengano misurate V_{CE} con un voltmetro e I_C con un amperometro, in modo da poter ricostruire la curva caratteristica per punti [2].

A. Polarizzazione della base

Nell'esperienza pratica si consiglia di acquisire alcune curve I_C vs V_{CE} per diversi valori di I_B fissati nel range approssimato $1 - 40 \mu\text{A}$, che corrispondono grossolanamente a valori di V_{BE} che vanno da poco sotto a poco sopra V_{thr} . In laboratorio non abbiamo disponibile un generatore di d.d.p. con le caratteristiche necessarie: esso va dunque costruito.

Lo schema rappresentato nella parte di sinistra di Fig. 1(b) rappresenta una possibile soluzione. In esso l'alimentatore $V_0 \sim 5 \text{ V}$, impiegato in molte altre esperienze di laboratorio, è seguito da un partitore di tensione variabile realizzato con un potenziometro (resistore variabile) R_V , che è anche stato già impiegato in una

precedente esperienza pratica. La posizione angolare dell'alberino del potenziometro determina la resistenza tra il terminale centrale (in serie a questo si trova una piccola resistenza $r = 100 \text{ ohm}$ nominali, che è praticamente irrilevante per i nostri scopi) e i terminali estremi. Dunque in questo modo si realizza un partitore di tensione che ha un rapporto di partizione dipendente dalla posizione angolare dell'alberino: all'"uscita" di questo partitore si ottiene una d.d.p. variabile, idealmente con continuità [3], da circa 0 a circa V_0 . All'"uscita" del partitore si trova un'ulteriore resistenza, denominata R_P (il simbolo sta per resistenza di polarizzazione), che è collegata alla base (si trascura la presenza del resistore R_B). La resistenza R_P serve per limitare la corrente che fluisce verso la base ai valori di nostro interesse (al massimo poche decine di μA): si consiglia di usare $R_P = 68 \text{ kohm}$ (oppure $R_P = 330 \text{ kohm}$) nominali (non c'è bisogno di misurare, e potete facilmente dedurre quali siano i corretti parametri di dimensionamento sulla base del range di valori desiderati per I_B). Un (micro)amperometro collegato in serie con la base permette di monitorare I_B , che dunque varierà *non linearmente* in funzione della posizione angolare dell'alberino del potenziometro. Nell'esperienza pratica si consiglia di monitorare anche V_{BE} , collegando un voltmetro ad alta impedenza di ingresso in parallelo: tuttavia questo non è strettamente necessario, visto che le condizioni di polarizzazione di base del transistor sono

completamente determinate da I_B .

La parte di destra dello schema di Fig. 1(b) mostra la strategia adottata per la variazione di V_{CC} , qui indicato come V_1 , e la misura di I_C .

Partiamo da quest'ultima: dato che vogliamo usare Arduino e che Arduino digitalizza delle tensioni analogiche in ingresso, dobbiamo trovare il modo di convertire I_C in una d.d.p.. Questo può essere facilmente ottenuto misurando la caduta di potenziale su una resistenza in serie al collettore, in pratica usando la stessa tecnica che abbiamo per esempio impiegato per ricostruire la curva caratteristica del diodo a giunzione. Questa resistenza è R_C , che è già presente nella basetta del transistor: il valore consigliato è $R_C = 1 \text{ kohm}$ nominali (questa resistenza va misurata con il multimetro); tenendo conto del tipico fattore di calibrazione di Arduino ($\xi \simeq 5 \text{ mV/digit}$), la sensibilità nella misura di I_C risulta dell'ordine di alcuni μA , che è sicuramente sufficiente per i nostri scopi. Ovviamente la misura della caduta di potenziale su R_C richiede di usare due ingressi analogici di Arduino (i pin A0 e A2) e misurare due d.d.p. rispetto a terra, indicate con V_1 e V_2 nello schema; V_2 coincide proprio con V_{CE} , e quindi, come già realizzato nell'esperienza della curva caratteristica del diodo, la misura delle due tensioni permette di conoscere sia I_C che V_{CE} . Si ha infatti

$$V_{CE} \equiv V_2 \quad (1)$$

$$I_C \equiv \frac{V_1 - V_2}{R_C} . \quad (2)$$

Anticipiamo che, come già notato nell'esperienza della curva caratteristica del diodo, questa configurazione sperimentale ha dei limiti nei valori di V_{CE} che possono essere raggiunti. Infatti, per alte intensità di corrente I_C la caduta di potenziale su R_C può diventare così grande da impedire a V_{CE} di crescere pur in presenza di un aumento di V_1 . Dunque il metodo qui presentato non è adatto per misure di I_C superiori a un certo valore (pochi mA) e alcune delle curve acquisite potranno risultare "monche".

L'ultimo ingrediente che dobbiamo reperire per la nostra ricetta è una sorgente variabile di d.d.p. (V_1) che giochi il ruolo della V_{CC} indicata in Fig. 1(a). Nell'esperienza della curva caratteristica del diodo abbiamo impiegato a questo scopo un'uscita PWM di Arduino seguita da un integratore RC. Potremmo implementare la stessa strategia anche in questo caso. Però, per scopi prevalentemente didattici, cioè per discutere alcuni aspetti generali e per semplificare il montaggio, in questa esperienza usiamo un altro approccio: infatti la d.d.p. variabile V_1 è qui ottenuta dal generatore di funzioni, regolato in termini di forma d'onda, ampiezza, offset e frequenza per creare un'onda triangolare di caratteristiche opportune.

B. Regolazione generatore funzioni e sincronizzazione

La filosofia della misura è a questo punto chiara: aggiustata, tramite il potenziometro, la corrente I_B a un dato

valore, il generatore di funzioni invia una d.d.p. variabile linearmente nel tempo; a intervalli regolari, separati tra di loro da una quantità Δt definibile via software, Arduino campiona le d.d.p. V_1 e V_2 presenti rispettivamente alle porte A0 e A2. Una volta registrate in un file al termine dell'acquisizione, esse sono trattate secondo Eq. 1 in modo da poter ricostruire per punti la curva caratteristica di collettore alla data corrente di base. La misura può essere poi ripetuta per altri valori di I_B per ottenere una famiglia di curve.

Affinché l'operazione abbia l'esito progettato, occorre rispettare un certo numero di requisiti nella forma d'onda prodotta dal generatore di funzioni. Questi requisiti, elencati qui di seguito, vanno controllati accuratamente *prima* di avviare l'acquisizione, usando l'oscilloscopio e, preferibilmente, avendo scollegato gli ingressi di Arduino.

1. Arduino può leggere e digitalizzare solo tensioni positive (o nulle) rispetto alla linea di terra. Di default il generatore di funzioni produce forme d'onda alternate, che cioè corrispondono a d.d.p. positive e negative. Pertanto è necessario aggiungere un offset continuo al segnale alternato. Questo si ottiene facilmente agendo sulla manopola **OFFSET** del generatore di funzioni, che deve essere estratta e regolata per aggiungere l'offset desiderato.
2. Arduino, poi, digitalizza segnali che valgano al massimo la propria V_{ref} , cioè, nelle nostre condizioni di operazione, circa 5 V. Pertanto è necessario che l'ampiezza massima della forma d'onda triangolare prodotta sia circa 5 V, o un po' meno per sicurezza. Questo si ottiene lavorando con la manopola **AMPL** del generatore di funzioni, iterando, se necessario, la regolazione dell'offset. Alla fine, sull'oscilloscopio usato per monitorare la forma d'onda essa deve apparire come un triangolo con ampiezza compresa tra circa 0 e circa 5 V.
3. La frequenza dell'onda triangolare deve anche essere aggiustata in maniera opportuna. Come chiariremo nel seguito, al massimo il campione acquisito consta di 256 punti. L'intervallo di campionamento Δt , nella configurazione usata per l'esperienza, può essere variato in passi discreti tra 1 e 9 ms (al solito, la selezione si esegue via software nello script di Python che controlla l'operazione), per cui l'intera acquisizione, trascurando gli eventuali ritardi e latenze di Arduino, dura al massimo un tempo compreso tra circa 250 ms e circa 2.3 s. Questo tempo deve corrispondere a quello necessario perché la forma d'onda triangolare passi dal suo valore minimo a quello massimo, cioè a *metà* del periodo T dell'onda stessa. Si sottolinea che le frequenze necessarie per la corretta esecuzione dell'esperienza sono così basse da rendere difficoltosa la visualizzazione di tracce continue sullo schermo dell'oscilloscopio, per cui si consiglia vivamente di lasciare la regolazione della frequenza "per ultima", in modo da eseguire più agevolmente il monitoraggio dell'ampiezza.

Soddisfare questi requisiti, in maniera più o meno rigorosa a seconda delle proprie capacità e della propria pazienza, è condizione essenziale per eseguire la misura, ma non sufficiente. Infatti occorre anche fare in modo che l’acquisizione dei dati da parte di Arduino parta *in sincrono* con l’“inizio” della forma d’onda triangolare. In altre parole, Arduino deve essere “triggerato” in modo opportuno dal generatore di funzioni.

C. Trigger

Di come sia possibile triggerare Arduino (per i nostri scopi) ci siamo già occupati in precedenza. Ripetiamo qui alcune considerazioni generali e le ricette pratiche impiegate, partendo dalla constatazione che, in linea di principio, esistono vari modi per eseguire la sincronizzazione richiesta. Per esempio, potremmo fare come si fa nei circuiti di trigger degli oscilloscopi, cioè monitorare continuamente il segnale V_1 e far partire l’acquisizione solo se questo è nella sua fase crescente. L’implementazione pratica di questo trigger è però abbastanza poco affidabile: essa è relativamente lenta (occorrono almeno due campionamenti consecutivi per capire la pendenza della forma d’onda) e, soprattutto, sensibile a fluttuazioni di lettura dovute a rumore, che potrebbero facilmente ingannare il trigger. La strategia di sincronizzazione qui impiegata si basa invece sulla disponibilità di un segnale specifico, progettato proprio per scopi di questo tipo e prodotto dal generatore di funzioni in sincrono con le forme d’onda. Questo segnale si trova su un connettore BNC marcato TTL/CMOS OUTPUT sul retro della scatola dell’apparecchio o, per alcuni modelli, direttamente sul frontale. L’indicazione TTL [4] significa che questo segnale segue le prescrizioni dello standard TTL [5] in termini di livelli. Questo standard stabilisce che il livello è basso (“low”, corrispondente a uno 0 binario) in corrispondenza a una d.d.p. praticamente nulla (ovvero minore di 0.8 V), e che il livello è alto (“high”, corrispondente a un 1 binario) per una d.d.p. maggiore di 2 V. Nella pratica (e quasi sempre), livello basso e livello alto di un segnale TTL corrispondono rispettivamente a circa 0 e circa 5 V (informazione molto utile, da ricordare sempre).

Questo segnale può essere facilmente letto da Arduino in forma *digitale*, cioè usando come ingresso una delle sue porte digitali (nella pratica è la porta 5). L’operazione è vantaggiosa rispetto alla lettura di un segnale analogico, poiché i livelli sono codificati, e pertanto meno sensibili a rumore e fluttuazioni, e non c’è bisogno di introdurre delle istruzioni software di confronto per stabilire se ci si trova al livello basso o alto [6].

Come già accennato, questo segnale TTL è sincrono con la forma d’onda [7]: il timing è infatti quello rappresentato in Fig. 2, dove si osserva come il segnale presente all’uscita del generatore di funzioni, opportunamente regolato secondo i requisiti prima elencati, inizi la sua fase di salita quando il segnale TTL passa dal livello alto a

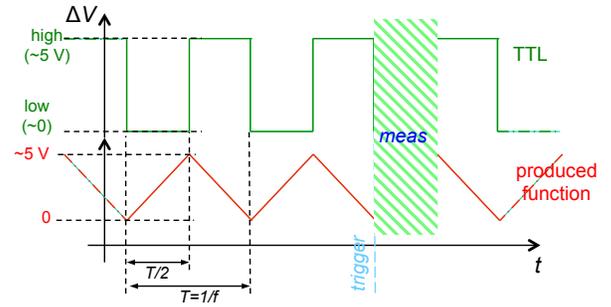


Figura 2. Schema del timing dei segnali: la forma d’onda triangolare prodotta dal generatore e usata come V_1 nell’esperienza è rappresentata in rosso, il segnale TTL di sincronizzazione in verde. Nella figura si suppone che l’istante di trigger sia quello indicato e che la misura avvenga all’interno del riquadro ombreggiato.

quello basso, cioè da circa 5 V a circa 0. Allora nel loop di acquisizione scritto nello sketch di Arduino dovremo fare in modo che le misure abbiano inizio quando il livello letto dalla porta digitale 5 passa da alto a basso. Notate, infatti, che porre una semplice condizione sul livello della porta digitale stessa, stabilendo per intenderci che l’acquisizione inizi quando esso è basso, potrebbe far partire l’acquisizione in un *qualsiasi* istante in cui il segnale TTL è basso, cioè non necessariamente quando esso passa da alto a basso, come invece è richiesto qui.

La Fig. 3 mostra lo schema delle connessioni che devono essere fatte alla scheda di Arduino. In sostanza è necessario impiegare quattro boccole a banana di colore diverso:

1. la boccola nera deve essere collegata alla linea di terra, o massa, del circuito di Fig. 1(b);
2. la verde deve ricevere il segnale di sincronismo proveniente dall’uscita (connettore BNC) TTL/CMOS OUTPUT sul retro, o sul frontale, del generatore di funzioni;
3. la gialla deve essere collegata al collettore del transistor [C in Fig. 1(b)] allo scopo di leggere la d.d.p. V_2 ;
4. la blu deve essere collegata a valle del resistore R_C di Fig. 1(b) allo scopo di leggere V_1 e anche all’uscita del generatore di funzioni che fornisce V_1 .

La boccola volante rossa è non collegata: tuttavia essa può essere utile, dato che corrisponde alla porta digitale 7 di Arduino, configurata, per default, come un’uscita. Un’opportuna istruzione dello sketch la pone costantemente a livello alto: dunque la misura della d.d.p. presente su questa porta (rispetto alla linea di massa, o terra) consente di determinare il valore della tensione di riferimento V_{ref} usata da Arduino, che può essere utile qualora si voglia stabilire il fattore di calibrazione ξ che serve per convertire la lettura digitalizzata in unità di differenza

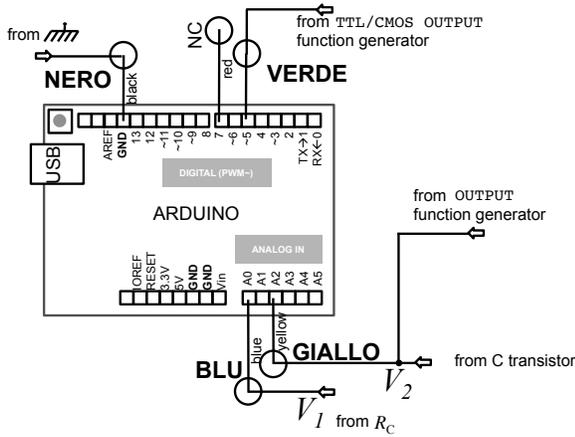


Figura 3. Schema delle connessioni da effettuare alla scheda Arduino con indicazione del colore delle boccole volanti. L'eventuale lettura di V_{ref} , utile per la calibrazione, può essere effettuata eseguendo una misura di tensione tra porta 7 (boccola rossa) e massa; infatti la porta 7 viene posta a livello alto, in prima approssimazione corrispondente a V_{ref} , da un'opportuna istruzione nello sketch.

di potenziale. In questa esperienza non è strettamente necessario individuare le grandezze richieste (V_{CE} e I_C) con grande accuratezza, per cui l'approccio di calibrazione "alternativo" basato sulla misura di V_{ref} (cioè della d.d.p. presente tra porta digitale 7 e linea di massa) è ampiamente sufficiente. Addirittura, vi mancasse il tempo, per evitare di prolungare troppo la vostra permanenza in laboratorio potreste anche accontentarvi di una sorta di calibrazione nominale, ponendo $\xi = 5 \text{ V} / 1023 \approx 5 \text{ mV}$.

D. Numero di punti acquisiti

Un altro aspetto di interesse per la gestione dell'esperienza riguarda il numero di punti sperimentali registrati in ogni acquisizione, cioè per ogni regolazione della corrente di base I_B . Come abbiamo già avuto modo di discutere in altre occasioni, la ridotta disponibilità di memoria del microcontroller di Arduino impone dei limiti nel massimo numero di dati che possono essere registrati, che qui è fissato a 256 punti [cioè vengono registrati su file due colonne con al massimo 256 punti ognuna, corrispondenti alla lettura digitalizzata rispettivamente delle porte analogiche A2 e A0, ovvero dei segnali V_2 e V_1 di Fig. 1(b)]. Il nostro desiderio sarebbe ovviamente che questi 256 punti fossero relativi all'intera fase di salita dell'onda triangolare usata come V_1 , in modo da poter esplorare l'intero intervallo di d.d.p. a nostra disposizione. Ottenere o meno questo risultato dipende dalla scelta dell'intervallo di campionamento Δt e della frequenza f del generatore di funzioni.

Infatti l'intervallo di campionamento Δt determina la durata temporale complessiva (massima) dell'acquisizione, $T_{acq} \approx 256 \times \Delta t$ (diventa un'uguaglianza se si trascu-

rano latenze, ritardi e fluttuazioni negli intervalli temporali), che deve essere confrontata con il semiperiodo $T/2 = 1/(2f)$ della forma d'onda prodotta dal generatore di funzioni. Possono allora verificarsi queste situazioni:

1. per T_{acq} (nettamente) minore di $T/2$, la misura viene effettuata in corrispondenza a un intervallo di variazione della d.d.p. V_1 (nettamente) minore del massimo possibile, cioè, in altre parole, V_1 è limitata in alto a qualcosa di (nettamente) minore del massimo possibile (circa 5 V);
2. viceversa, se T_{acq} è (nettamente) maggiore di $T/2$, le misure vengono eseguite non solo per la fase in cui V_1 cresce, ma anche per quella in cui esso cala; di conseguenza i dati registrati non sono più ordinati in maniera monotona. Per evitare gli eventuali problemi connessi a questa situazione [8], l'acquisizione di Arduino viene interrotta automaticamente non appena il livello del segnale TTL di sincronismo passa per la prima volta dal livello alto a quello basso, come rappresentato in Fig. 2: dunque in ogni caso i dati vengono registrati solo nella fase crescente di V_1 .

L'ultima affermazione che abbiamo riportato implica che, nel caso considerato, il campione di misure sia costituito da un numero minore di 256 punti. Quindi rimane importante che l'intervallo di campionamento e la frequenza del generatore di funzioni vengano scelte in modo opportuno. Come illustreremo nel seguito, per aiutare nella verifica lo script di Python che gestisce l'acquisizione riporta sulla console delle indicazioni utili, cioè il numero di punti acquisito, che idealmente dovrebbe essere prossimo a, 256, e anche i valori minimi e massimi digitalizzati per V_1 , che dovrebbero essere prossimi rispettivamente a 0 e 1023.

III. SKETCH E SCRIPT

Lo sketch di Arduino scritto per l'esperienza, disponibile in rete sotto il nome `curv.ino` e riportato in Appendice, è sviluppato sulla base di quello, discusso in precedenza, usato per la curva caratteristica del diodo a giunzione. Le modifiche principali riguardano:

1. la definizione della porta 5 come ingresso digitale (l'istruzione, posta nella parte di inizializzazione della scheda, recita `pinMode(sincPin, INPUT);`, dove `sincPin` è una costante intera che punta alla porta in questione;
2. l'implementazione del trigger come descritto sopra, che in sostanza è eseguito attraverso due loop successivi di attesa realizzati con le istruzioni `while (sinc==LOW)` e `while (sinc==HIGH)`, dove la lettura della porta di ingresso digitale viene eseguita con le istruzioni `sinc = digitalRead(sincPin);` inserite nei loop;

3. la presenza dell'istruzione `if (sinc==HIGH) break;` all'interno del ciclo di acquisizione, che serve per interromperlo se il segnale di sincronismo passa a livello alto, secondo quanto discusso prima;
4. l'eventuale aggiunta della parola '9999' al termine dell'array che contiene i dati nel caso in cui l'acquisizione sia eseguita su meno di 256 punti, che serve per segnalare allo script di Python la fine dei dati.

Anche lo script di Python, disponibile in rete sotto il nome `curv_v1.py`, è molto simile a quello già impiegato per l'esperienza della curva caratteristica del diodo. Si ricorda che, al termine dell'acquisizione, viene creato un file di due colonne (e un numero di linee pari a quello delle misure effettuate, fino a un massimo di 256) che contiene i valori digitalizzati di, nell'ordine, V_1 e V_2 [vedi Fig. 1(b)]. Questi dati sono disponibili per ulteriori analisi e in particolare per tracciare per punti la curva caratteristica di interesse. Inoltre lo script, oltre al nome del file e all'eventuale directory di archiviazione, permette di impostare l'intervallo di campionamento Δt in unità di ms (da 1 a 9 ms, con passi discreti di 1 ms). Un'opportuna serie di istruzioni consente di gestire e registrare files contenenti meno di 256 punti: a questo scopo i dati trasmessi da Arduino via porta seriale vengono monitorati via via che arrivano, e il loop di lettura e registrazione su file di questi dati viene interrotto o quando sono stati registrati 256 coppie di dati, oppure quando il dato è rappresentato dalla parola '9999'. Il numero di punti acquisiti, assieme al valore minimo e massimo digitalizzato di V_1 , viene scritto sulla console per praticità.

IV. ESEMPIO DI MISURE

La Fig. 4 riporta come esempio una famiglia di curve caratteristiche di uscita, o di collettore, per il transistor 2N1711, registrate come descritto in questa nota per le intensità di corrente di base I_B riportate in legenda (nell'intervallo 1.2–35 μA , che è sufficiente per i nostri scopi). Le acquisizioni sono state effettuate con $\Delta t = 1$ ms (nominale); i campioni sono costituiti da 218 punti (dunque un numero abbastanza prossimo, ma non troppo, a quello "ideale" di 256), con V_1 variabile tipicamente tra 1 e 988 digit (dunque un intervallo abbastanza prossimo a quello "ideale" compreso tra 0 e 1023).

Per la conversione delle letture digitalizzate in unità di d.d.p. (V) si è usato il fattore di conversione, $\xi = (4.98 \pm 0.03)$ mV/digit, stabilito leggendo la V_{ref} di Arduino presente sulla porta 7 e tenendo conto che il convertitore analogico/digitale di Arduino ha una "dinamica" di 10 bit. Per la determinazione dell'intensità di corrente I_C secondo Eq. 1 si è impiegato il valore, misurato con tester digitale, $R_C = (1028 \pm 8)$ ohm. Le barre di errore del grafico su sono state deliberatamente e arbitrariamente scelte come rappresentative della sola incertezza di digitalizzazione (± 1 digit, convertito in unità fisiche per la misura di V_{CE} , per l'asse orizzontale e

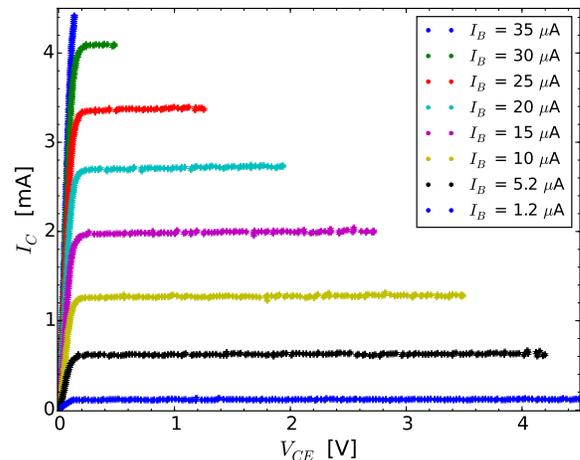


Figura 4. Esempio di famiglia di curve caratteristiche di uscita del transistor 2N1711 ricostruite nell'esperienza pratica. Le barre di errore tengono conto dell'incertezza di digitalizzazione; l'incertezza di calibrazione è attesa contribuire per meno dell'1% della lettura per le barre di errore orizzontali, e per meno del 2% della lettura per le barre di errore verticali.

± 2 digit, convertito in unità fisiche per la misura di I_C , per l'asse verticale, dove si è tenuto conto che la misura di corrente è proporzionale alla differenza di due distinte letture di d.d.p. e si è usata l'incertezza massima risultante). Questa scelta è sostanzialmente dovuta al fatto che i dati acquisiti non vengono qui impiegati per trarre conclusioni quantitative, per esempio per determinare una grandezza fisica tramite best-fit, per cui non si ritiene utile tenere conto degli errori di calibrazione. La scelta è debitamente segnalata nella caption della figura, dove si riporta anche una stima dell'errore di calibrazione.

Le curve ottenute sono in accordo qualitativo con le aspettative. In particolare, per $I_B = 1.2 \mu\text{A}$ la corrente che fluisce nel collettore è al massimo dell'ordine di poche decine di μA , suggerendo come in queste condizioni il transistor si trovi in condizioni di lavoro prossime a quelle di *interdizione*. Infatti la debole corrente di base misurata corrisponde a V_{BE} inferiore, o perlomeno non decisamente superiore, alla soglia. All'aumentare di I_B , invece, la giunzione BE diventa sicuramente polarizzata direttamente. In queste condizioni, si verifica che il transistor può accedere al regime attivo, in cui la giunzione BC è polarizzata inversamente, cioè, con ovvio significato dei simboli, $V_{BC} < V_{thr}$. Poiché, con altrettanto ovvio significato dei simboli, $V_{CE} = V_{BE} + V_{CB} = V_{BE} - V_{BC}$, si può porre $V_{BC} = V_{BE} - V_{CE}$, per cui la giunzione BC risulta polarizzata inversamente per $V_{CE} > V_{BE} - V_{thr}$. D'altra parte, la polarizzazione diretta della giunzione BE implica che V_{BE} sia "leggermente" al di sopra di V_{thr} , per cui la condizione che stiamo cercando si traduce in $V_{CE} > V'$, tipicamente con $0 < V' < V_{thr}$ (e V' ovviamente dipendente da V_{BE} , cioè da I_B). La Fig. 4 mostra che per il transistor esaminato, e per il range di I_B esploro-

rato, si ha $V' \sim 0.2$ V; infatti al di sotto di questo valore I_C tende rapidamente a zero, al di sopra di questo valore I_C diventa sensibilmente maggiore di zero, mostrando un'evidente transizione dal regime di *saturazione* a quello attivo.

È rilevante osservare che le curve acquisite con i valori più alti di I_B esplorati in questa esperienza non coprono lo stesso range di variazione di V_{CE} di quelle registrate a valori di I_B minori. La spiegazione è ovvia e immediata: quando la giunzione BE è “fortemente” polarizzata in modo diretto, allora una rilevante intensità di corrente scorre attraverso il collettore. Questa corrente provoca una altrettanto rilevante caduta di potenziale sulla resistenza di collettore R_C , e quindi limita il massimo valore possibile per V_{CE} . In altre parole, il generatore che produce V_{CE} , visto in termini di Thévenin, ha una resistenza interna elevata, che è data dalla somma della resistenza interna del generatore di funzioni (50 ohm, quindi trascurabile) e di R_C (vale circa 1 kohm, tutt'altro che trascurabile).

Per il transistor esaminato, a $I_B = 35 \mu\text{A}$ la corrente I_C diventa ben presto così alta da rendere la caduta di potenziale su R_C simile a V_1 , per cui, di fatto, può essere ricostruita solo una piccola porzione della curva caratteristica. Infatti, tenendo conto che $R_C = 1$ kohm nominali, per $I_C \rightarrow 5$ mA, $V_{CE} \rightarrow 0$ anche per $V_1 \sim 5$ V. La limitazione potrebbe essere rimossa impiegando una resistenza di collettore più bassa, ma questo non è necessario per i nostri scopi.

Il brusco passaggio da piccole a grandi correnti di collettore può ovviamente essere realizzato anche mantenendo costante V_1 e agendo su I_B : nelle condizioni dell'esperienza, per $V_1 \sim 5$ V, per esempio, quando I_B viene variata da circa $1 \mu\text{A}$ a $35 \mu\text{A}$ la corrente di collettore passa da un valore trascurabile (poche decine di μA) a valori dell'ordine di 5 mA. Questo comportamento è alla base dell'impiego del transistor come *switch* e illustra bene come la variazione di una “piccola” corrente (quella di base, e la variazione è piccola in termini assoluti) possa controllare la variazione di una corrente ben più “grande” (quella di collettore, e la variazione è grande in termini assoluti).

Il modello di Ebers-Moll, che, ricordiamo, rappresenta una descrizione approssimativa del funzionamento del transistor, prevede

$$I_C = -I_{0C} \left[\exp\left(\frac{V_{BC}}{\eta V_T}\right) - 1 \right] + \alpha_F I_{0E} \left[\exp\left(\frac{V_{BE}}{\eta V_T}\right) - 1 \right], \quad (3)$$

dove I_{0C} e I_{0E} sono le correnti di saturazione inversa delle due giunzioni BC e BE, e $\alpha_F \simeq 1$ è il coefficiente dell'“effetto transistor”. Nel regime di cui ci stiamo occupando, in cui BE è polarizzata direttamente, il secondo termine della somma fornisce un contributo pressoché costante e numericamente molto rilevante, essendo $V_{BE} \gg \eta V_T$. Il primo termine della somma diventa sicuramente trascurabile quando V_{BC} è piccolo, o addirittura negativo, come si verifica per $V_{CE} > V_{BE}$, cioè in pieno regime attivo. Nel regime di saturazione, invece, V_{BC} diventa positiva e approssima V_{thr} per $V_{CE} \rightarrow 0$: in

queste condizioni il primo termine della somma in Eq. 3 contribuisce con un termine negativo, che diventa tanto più grande quanto più V_{BC} si avvicina a V_{thr} . Di conseguenza I_C tende ad annullarsi seguendo un brusco andamento, come trovato in Fig. 4 per $V_{CE} < V'$. In altre parole, l'Eq. 3 può giustificare analiticamente l'andamento delle curve caratteristiche nella transizione da regime attivo a regime di saturazione, ma il suo impiego, per esempio attraverso un best-fit, *non è richiesto* nell'esperienza pratica. Esso, infatti, non è particolarmente significativo, poiché i parametri di fit eventualmente determinati, che sono legati in maniera non completamente ovvia con i parametri fisici del modello, possono essere trovati in maniera generalmente più immediata [9].

A. Guadagno in corrente

Una considerazione immediata che si può fare esaminando le curve caratteristiche di uscita di un transistor è la differenza di ordini di grandezza tra le intensità di corrente di base e di collettore che si verifica quando si opera in regime attivo. Questa è in effetti una delle caratteristiche più importanti, o la più importante, del transistor a giunzione bipolare (*BJT*). Infatti l'effetto transistor, specifico del regime attivo, dà luogo a un'amplificazione delle correnti (continue) secondo la legge $I_C = \beta_F I_B$. Il termine β_F , definito spesso anche h_{FE} , è legato a quello del coefficiente α_F dell'effetto transistor secondo la $\beta_F = \alpha_F / (1 - \alpha_F)$; poiché $\alpha_F \sim 1$, β_F è tipicamente grande, dell'ordine delle decine o centinaia.

In prima approssimazione, β_F può essere ritenuto indipendente da I_C . Tuttavia, per grandi correnti di collettore l'effetto transistor diventa più marcato, poiché la probabilità che gli elettroni che superano l'interfaccia emettitore-base si ricombinino nella base diminuisce a causa della densità finita di lacune, conseguenza del drogaggio a bassa densità di questa regione (drogata di tipo *p* in un transistor *npn*). Pertanto, α_F si avvicina sempre di più all'unità, e β_F tende ad aumentare con I_C . Sfortunatamente, poi, il valore di α_F , oltre a dipendere dalle condizioni di operazione (temperatura inclusa), ha un'estrema variabilità legata alle fluttuazioni di fabbricazione. Per intenderci, il datasheet del transistor 2N1711 indica, per $I_C = 10$ mA e $V_{CE} = 10$ V (condizioni non raggiunte nella presente esperienza), un valore tipico $\beta_F = 80$, ma, allo stesso tempo, avverte che la tolleranza di fabbricazione rende possibili valori compresi tra $\beta_F = 35$ e $\beta_F = 300$ (uno spread di quasi un ordine di grandezza).

La misura della curva caratteristica di uscita permette di determinare β_F nelle varie condizioni di operazione. Allo scopo, è sufficiente prendere i dati relativi a un certo valore di I_B (misurato con il multimetro), scegliere un dato valore di V_{CE} (la scelta è “libera”, qui si è preso $V_{CE} = 1.0$ V nominali), vedere nel file o, per praticità, direttamente dal grafico, quanto vale I_C e infine calcolare il rapporto $\beta_F = I_C / I_B$. Come esempio, si riporta in

I_B [μA]	I_C [mA]	β_F
5.2 ± 0.1	0.63 ± 0.02	121 ± 5
10.1 ± 0.2	1.32 ± 0.04	131 ± 8
14.9 ± 0.2	1.99 ± 0.06	133 ± 10
19.8 ± 0.3	2.75 ± 0.08	139 ± 12

Tabella I. Tabella I. Valori di I_B e I_C desunti da alcune delle curve caratteristiche di Fig. 4 e corrispondente valore del guadagno in corrente (continua) $\beta_F = I_C/I_B$. I dati si riferiscono tutti al valore $V_{CE} = 1.0$ V nominali. Le incertezze sulla misura di I_B sono quelle dovute all'impiego del multimetro digitale, quelle su I_C tengono conto dell'errore di digitalizzazione, di quello di calibrazione di Arduino e dell'errore nella misura di R_C .

Tab. I il calcolo eseguito per alcuni valori, naturalmente tutti corrispondenti all'operazione nel regime attivo (il guadagno in corrente non è definito per gli altri regimi di operazione del transistor). Per la determinazione dell'incertezza su I_C , e quindi su β_F , sono state considerate tutte le cause di errore (calibrazione di V_1 e V_2 , loro incertezza di digitalizzazione, incertezza sulla misura di R_C , il tutto opportunamente propagato con somme in quadratura). Si osserva come i valori di β_F tendano a crescere, appena più delle barre di errore, con I_C .

Come è noto, il comportamento da amplificatore in corrente del transistor in regime attivo può essere leggermente differente a seconda che le correnti di base e collettore siano continue o alternate (in quest'ultimo caso, ci si riferisce a *deboli segnali* alternati). Infatti il guadagno in corrente corrispondente, indicato con β_f (o h_{fe} , dove l'uso dei pedici minuscoli rimanda alla presenza dei "deboli" segnali), è riportato nel datasheet avere un valore tipico di 135 (tolleranza da 70 a 300, dunque sempre gigantesca). La sua misura diretta richiede di realizzare un opportuno circuito amplificatore e l'uso di un debito modello, secondo quanto sarà svolto in una prossima esperienza pratica. Qui possiamo limitarci a darne una stima. A questo scopo possiamo registrare curve caratteristiche I_C vs V_{CE} in corrispondenza a valori fissati di I_B che siano "poco distanti" l'uno dall'altro. In questo esempio, sono state impiegate le curve corrispondenti a $I_B = (10.1 \pm 0.2) \mu\text{A}$ e $I_B = (14.9 \pm 0.2) \mu\text{A}$ [10], di cui sono stati ricavati i valori di I_C per $V_{CE} = 1.0$ V (nominali) riportati in Tab. I. Quindi si è definito $\beta_f \simeq \Delta I_C / \Delta I_B$, dove ΔI_B e ΔI_C sono le *variazioni* delle due intensità di corrente corrispondenti alle due condizioni (esse si determinano facilmente eseguendo la differenza). Si è ottenuto $\beta_f = 140 \pm 12$, un valore leggermente superiore rispetto ai corrispondenti β_F . Dunque il transistor esaminato, debitamente polarizzato, si comporta da amplificatore con un guadagno grossolanamente compreso tra 130 e 140 sia per correnti continue che per (deboli) correnti alternate.

B. Effetto Early

Una delle affermazioni approssimate che si usa fare per descrivere il comportamento di un transistor nel regime attivo è che la corrente I_C dipende *esclusivamente* da I_B . Questo è anche il comportamento grossolanamente previsto dal modello di Ebers-Moll: se nell'Eq. 3 poniamo condizioni tipiche del regime attivo, cioè $V_{BE} \gtrsim V_{thr}$ e $V_{BC} < 0$, I_C risulta pressoché indipendente da V_{CE} (la dipendenza da V_{CE} è implicita, aumentando V_{CE} si rende sempre più negativa V_{BC} , ma poiché V_{BC} compare come argomento di un esponenziale, la variazione del termine che la contiene è poco rilevante).

Che questa approssimazione sia non completamente verificata si può dimostrare con i nostri dati. Infatti risulta abbastanza evidente che, anche in pieno regime attivo, I_C non è costante, ma tende a crescere con V_{CE} . L'effetto che interpreta questo andamento è chiamato *effetto Early*. All'aumentare di V_{CE} , la giunzione base-collettore risulta inversamente polarizzata in modo sempre più marcato e il campo elettrico nella regione di collettore (orientato "verso la base" per un transistor *npn*) aumenta la sua intensità. Dal punto di vista qualitativo, possiamo supporre che:

- la regione di svuotamento della giunzione BC aumenti la sua estensione longitudinale, ovvero il suo spessore, rendendo meno probabile la ricombinazione della corrente di elettroni che proviene dall'emettitore e passa nella base;
- l'efficacia della raccolta di elettroni da parte del campo presente nel collettore aumenti, e una densità maggiore di carica, cioè una corrente di intensità maggiore, fluisca nel collettore.

Entrambi questi effetti portano α_F più prossimo all'unità, cioè aumentano β_F e di conseguenza l'intensità di corrente I_C , al crescere di V_{CE} .

L'effetto Early è modellato dalla seguente espressione:

$$I_C \simeq \beta_F I_B \left(1 + \frac{V_{CE}}{V_{Early}} \right), \quad (4)$$

dove V_{Early} , che ha le dimensioni di una d.d.p., rappresenta un parametro caratteristico del transistor impiegato (naturalmente dipende anche dalle condizioni di operazione) e ha valori tipici compresi tra circa 50 e circa 200 V. Supponendo costante il termine $\beta_F I_B$, l'Eq. 4 stabilisce un andamento lineare crescente, dove $-V_{Early}$ rappresenta l'intercetta con l'asse orizzontale del grafico della curva caratteristica di uscita.

La Fig. 5 mostra gli stessi dati di Fig. 4 per $I_B = 10 \mu\text{A}$, graficati su una diversa scala per evidenziare l'andamento crescente. I dati per $V_{CE} > 0.2$ V (scelto arbitrariamente) sono stati fittati numericamente a una retta del tipo $y = a + bx$; supponendo (arbitrariamente) che le incertezze ΔI_C fossero correttamente rappresentative dell'errore statistico (esse corrispondono all'errore

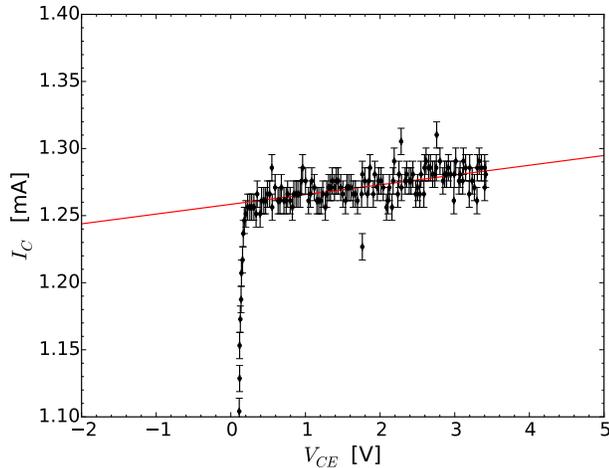


Figura 5. Stessi dati di Fig. 4 per $I_B = 10 \mu\text{A}$ graficati su una scala diversa per evidenziare l'andamento crescente attribuito all'effetto Early. La linea continua rappresenta un best-fit lineare ai dati corrispondenti a $V_{CE} > 0.2 \text{ V}$. Si notano alcuni punti sperimentali apparentemente fuori dalle aspettative, che potrebbero essere dovuti a fluttuazioni nella digitalizzazione o contributo di rumori esterni.

di digitalizzazione), nella procedura di best-fit è stata scelta l'opzione `absolute_sigma=True`. Il fit ottenuto è sovrapposto ai dati in Fig 5. I risultati sono

$$a = (1.258 \pm 0.002) \text{ mA} \quad (5)$$

$$b = (7.3 \pm 9) \mu\text{A/V} \quad (6)$$

$$\chi^2/\text{ndof} = 114/120 \quad (7)$$

$$R^2 = 0.999 \quad (8)$$

$$\text{norm. cov.} = -0.89. \quad (9)$$

I parametri trovati conducono a $V_{Early} = a/b = (173 \pm 22) \text{ V}$, dove nell'incertezza abbiamo tenuto conto della calibrazione di Arduino e della barra di errore nella misura di R_C .

APPENDICE: SKETCH

```
//Dichiarazione
const unsigned int sincPin = 5; //pin 5 ingresso digitale per la sincronizzazione con il generatore
const unsigned int digitalPin = 7; //pin 7 uscita digitale (serve per eventuale calibrazione convertitore)
const unsigned int analogPin_uno=0; //pin A0 per lettura V1
const unsigned int analogPin_due=2; //pin A2 per lettura V2
unsigned int i=0; //variabile di conteggio per i cicli
int V1[256]; //array per memorizzare V1 (d.d.p, letta da analogPin_uno)
int V2[256]; //array per memorizzare V2 (d.d.p, letta da analogPin_due)
int delays; //variabile che contiene il ritardo tra due step successivi (in unita' di 1 ms, fino a 9 ms)
int start=0; //flag per dare inizio alla misura
int sinc; //variabile di sincronizzazione
int nmis=0; //variabile con il numero di punti acquisiti
//Inizializzazione
void setup()
{
  pinMode(sincPin, INPUT); //pin sincPin configurato come ingresso digitale
  Serial.begin(9600); //inizializzazione della porta seriale
  Serial.flush(); // svuota il buffer della porta seriale
}
//Ciclo di istruzioni del programma
void loop()
{
  if (Serial.available() >0) // Controlla se il buffer seriale ha qualcosa
  {
    delays = (Serial.read()-'0')*1; // Legge il byte e lo interpreta come ritardo (unita' 1 ms)
    Serial.flush(); // Svuota il buffer della seriale
    start=1; // Pone il flag start a uno
  }
  if(!start) return // solo se il flag e' a uno parte l'acquisizione
```

```

delay(2000); // attende 2s per evitare casini
digitalWrite(digitalPin,HIGH); // pone digitalPin a livello alto (per eventuale calibrazione)
sinc = digitalRead(sincPin); //legge sincPin
while (sinc==LOW) // ciclo di attesa iniziale per sincronizzazione, attende che sincPin vada alto
  {sinc = digitalRead(sincPin);} //legge sincPin
while (sinc==HIGH) // ciclo di attesa iniziale per sincronizzazione, attende che sincPin vada basso
  {sinc = digitalRead(sincPin);} //legge sincPin
for(i=0;i<256;i++) //ciclo di acquisizione temporizzata delle misure (fino a 256 punti)
  {
  sinc = digitalRead(sincPin); //legge sincPin
  if (sinc==HIGH) break; //esce dal ciclo se sincPin torna alto
  delay(delayms); //aspetta il tempo impostato
  V1[i]=analogRead(analogPin_uno); //legge il pin analogPin_uno
  V2[i]=analogRead(analogPin_due); //legge il pin analogPin_due
  }
nmis = i; //scrive in nmis il numero effettivo di punti acquisiti
for(i=0;i<nmis;i++) //nuovo ciclo che scorre gli array di dati e li scrive sulla seriale
{
  Serial.print(V1[i]);
  Serial.print(" ");
  Serial.println(V2[i]);
}
if (nmis<256) //se il numero di punti e' minore di 256 scrive un carattere convenzionale di fine file
{
  Serial.println(9999); //il carattere convenzionale e' 9999
}
start=0; // Annulla il flag
Serial.flush(); // svuota il buffer della porta seriale
}

```

-
- [1] La tensione di soglia per un'ordinaria giunzione in silicio vale normalmente $V_{thr} = 0.45 - 0.65$ V. Notate che il suo valore non è completamente definito: da un lato, le tolleranze costruttive e i parametri di disegno possono influenzare V_{thr} , dall'altro, come abbiamo già osservato per il diodo, la grandezza V_{thr} non è individuabile in modo univoco. Dunque considerate sempre come approssimativo il valore di V_{thr} ogni volta che ci si fa riferimento.
- [2] A differenza di V_{BE} , non ci sono limiti di principio nel range di variazione di V_{CE} . Infatti è sufficiente disporre di un generatore di d.d.p. positiva che vari da zero (convenzionalmente le curve caratteristiche partono da $V_{CE} = 0$) fino a un valore massimo che è idealmente limitato dalla massima tensione, o corrente, sopportabile dal transistor. La disponibilità di strumentazione che abbiamo in laboratorio è sicuramente adeguata per evitare di lavorare in condizioni che possano danneggiare il transistor 2N1711.
- [3] Sapete per esperienza che il potenziometro è un dispositivo "delicato": la rotazione dell'alberino può comportare brusche variazioni della d.d.p. in uscita dal partitore, e questa d.d.p. può fluttuare anche in maniera sensibile. È compito del "bravo sperimentale" evitare che queste difficoltà tecniche diventino troppo importanti e anche che esse siano causa di eccessive (e ovvie) lamentazioni.
- [4] In realtà il segnale può anche seguire lo standard cosiddetto CMOS, che è caratterizzato da livelli ed eventualmente polarità diverse. Per commutare da TTL a CMOS occorre agire su una specifica manopola del frontale del generatore di funzioni, estraendola e quindi ruotandola per aggiustare i livelli.
- [5] TTL sta per *Transistor-Transistor-Logic*, uno standard di impulsi logici tra i primi a essere codificato, già agli albori dell'elettronica digitale.
- [6] La lettura di segnali "digitali" è quasi sempre preferibile rispetto a quella di segnali analogici nel caso in cui si sia interessati solo a conoscere uno "stato" (alto o basso, in questo caso). Tuttavia vale la pena di ricordare che possono esserci delle eccezioni nel caso di segnali che passano da un livello all'altro in modo molto rapido o che hanno una durata molto breve: essi potrebbero non essere letti in modo corretto da Arduino o dare luogo a "rimbalzi" del segnale. Al momento, questi problemi non sembrano essere presenti nell'implementazione pratica qui proposta, grazie anche alla relativa lentezza del campionamento, ma non se ne può escludere a priori la presenza.
- [7] L'eventuale ritardo interno al generatore di funzioni tra le forme d'onda di interesse è sicuramente trascurabile per gli scopi di questa esperienza. I ritardi interni delle apparecchiature sperimentali potranno diventare rilevanti quando eseguirete misure su scale temporali brevi, per

esempio inferiori al μs (cosa che non si realizza nel nostro corso di laboratorio).

- [8] Quello qui menzionato è un requisito di carattere poco più che “estetico”. Sarebbe infatti possibile ordinare a posteriori i dati in modo da renderli crescenti, qualora questo fosse necessario. Tuttavia, operare nelle condizioni descritte comporta inevitabilmente una riduzione della “densità” dei punti acquisiti, cioè, per intenderci, la distanza in d.d.p. tra due acquisizioni successive di V_1 sarebbe maggiore di quanto ottenibile con una regolazione ottimale dei tempi.
- [9] Se ne avete voglia, potete comunque provare un best-fit per una curva che mostri la transizione tra regime di saturazione e regime attivo. Potete verificare facilmente che l’Eq. 3, opportunamente rimanipolata per V_{BE} co-

stante, conduce a $I_C = A - B \exp(-V_{CE}/(\eta V_T))$, con $A \simeq B \simeq \beta_F I_B$, dove β_F è il guadagno in corrente (per correnti continue) del transistor, qui supposto costante. Questa funzione descrive i dati in maniera non completa, ma tuttavia ragionevole, considerando le varie approssimazioni del modello di Ebers-Moll.

- [10] La regolazione della corrente di base tramite potenziometro rende difficile fare aggiustamenti fini. In una serie di dati non riportata in questa nota sono state eseguite misure per valori di I_B distanti tra loro di $1 \mu A$ (sempre nel range $10 - 15 \mu A$) e i valori di I_C corrispondenti a $V_{CE} = 1.0 V$ sono stati fittati a una retta passante per l’origine: la pendenza di questa retta corrisponde a β_f . Il valore ottenuto dal best-fit è in accordo con quello determinato nel modo, molto più immediato e grossolano, discusso nel testo.