

Nome e Cognome:

 LUN MAR GIO

Data:

10

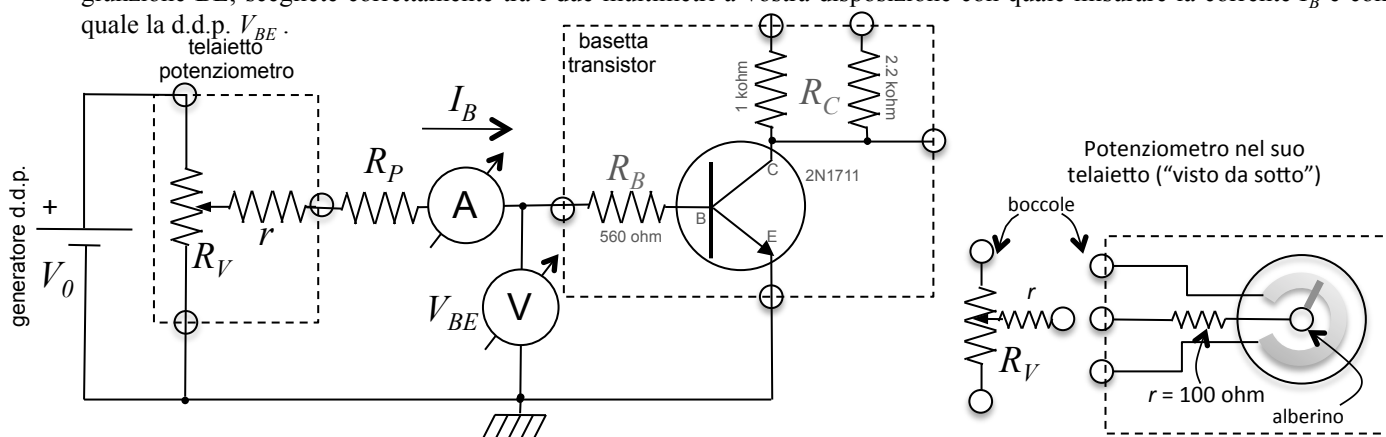
Curve caratteristiche di collettore del transistor (con Arduino)

Obiettivo dell'esperienza è realizzare un esperimento con presa dati automatizzata via Arduino finalizzato a ricostruire per punti una famiglia di curve caratteristiche di uscita, o di collettore, I_C vs V_{CE} , del transistor *npn* modello 2N1711. La famiglia si riferisce a valori di I_B discreti nell'intervallo consigliato da circa 1 a circa 35 μA .

Il transistor è montato su una basetta alloggiata in un telaietto: è necessario che preliminarmente individuiate le connessioni alle varie boccole, che sono quelle rappresentate qui a fianco.

Quindi, visto che il setup sperimentale è abbastanza complicato, procedete per passi successivi, come qui di seguito illustrato.

1. Montate il circuito sotto indicato (si consiglia $R_p = 68 \text{ kohm}$ oppure 330 kohm nominali), che serve per polarizzare la giunzione BE; scegliete correttamente tra i due multimetri a vostra disposizione con quale misurare la corrente I_B e con quale la d.d.p. V_{BE} .



2. Verificate che ruotando l'alberino del potenziometro venga variata la polarizzazione della giunzione BE: potete farlo qualitativamente oppure costruendo un grossolano grafico I_B vs V_{BE} . Commentate qui sotto se l'andamento soddisfa le aspettative e anche se, e perché, la misura del voltmetro è veramente rappresentativa della d.d.p. tra base ed emettitore.

Commenti sull'andamento I_B vs V_{BE} :

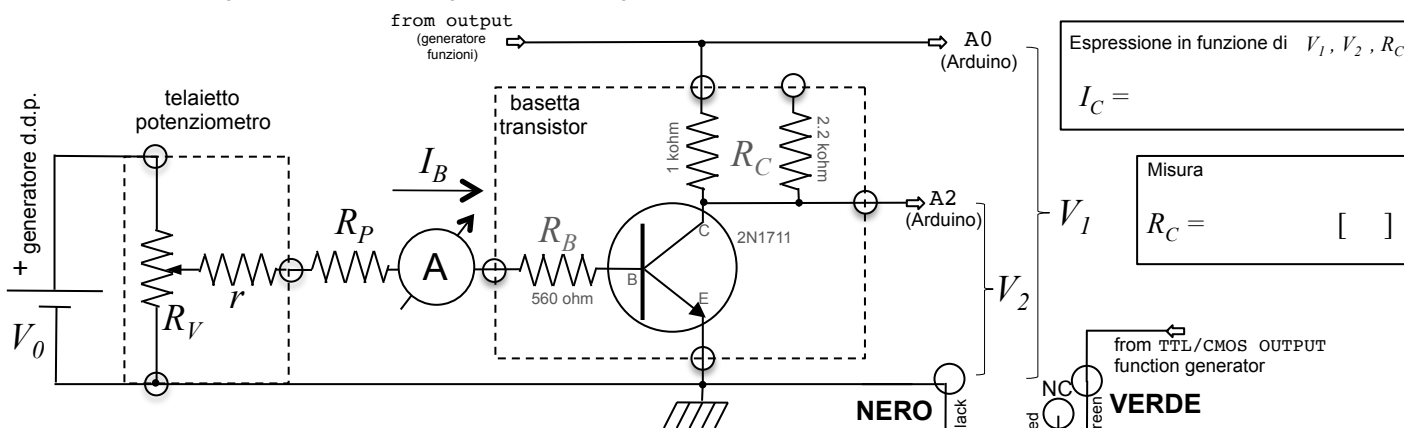
3. Predisponete il generatore di funzioni per gli scopi dell'esperienza: a questo scopo, esso deve produrre una forma d'onda triangolare di d.d.p. (rispetto terra) rigorosamente compresa tra circa 0 e circa 5 V. Verificate poi che il segnale all'uscita TTL/CMOS OUTPUT (lo trovate riportato su un cavo coassiale già collegato, il nero è la terra), che userete come segnale di sincronismo, sia "come ve lo aspettate" (fate un disegno nei commenti per dimostrarlo).

Commenti e/o disegni per mostrare le forme d'onda rilevanti:

4. Preliminarmente ai cicli di acquisizione, dovete eseguire l'upload dello sketch `curv.ino` nella memoria di Arduino utilizzando il programma Arduino (o Arduino IDE) nel computer di laboratorio.
5. Quindi dovete modificare lo script di Python (nome `curv_v1.py`) che serve per gestire la comunicazione seriale via USB, scegliendo il nome (eventualmente la directory) del file generato. Lo script prevede anche l'impostazione dell'intervallo di tempo Δt fra due campionamenti successivi, impostabile da 1 a 9 ms in passi discreti unitari.
6. Prima di eseguire i cicli di acquisizione, dovete aggiustare la frequenza f del generatore di funzioni sulla base del Δt prescelto, allo scopo di permettere un'"ottimale" registrazione dei dati: tenete presente che l'acquisizione avviene su un massimo di 256 punti, e ragionate di conseguenza.

7. A questo punto potete assemblare l'intero circuito, che prevede di usare i componenti montati finora per la polarizzazione della giunzione BE (si può omettere il voltmetro), il generatore di funzioni per produrre il segnale V_1 (vedi lo schema), la scheda Arduino per la digitalizzazione e misura di $V_2 = V_{CE}$ e di V_1 , necessaria per dedurre I_C tramite misura di R_C (si consiglia $R_C = 1 \text{ kohm}$ nominale).

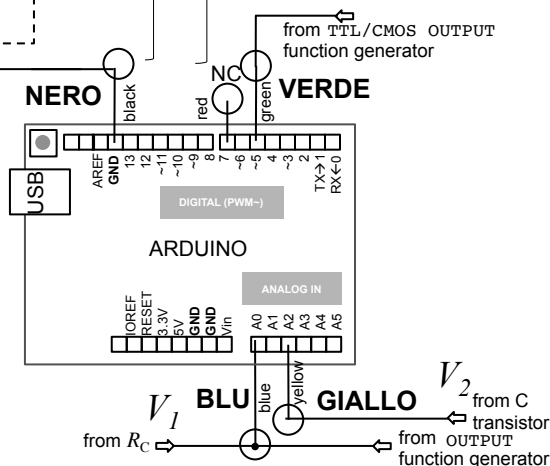
10



8. Fissate un certo valore di I_B ed eseguite l'acquisizione. Per vostra comodità, lo script di Python scrive su console il numero di punti acquisito e i valori min e max di V_1 : se non siete soddisfatti, aggiustate quello che dovete aggiustare e ripetete l'acquisizione. Al termine si ottiene un file di due colonne (e fino a 256 righe) che riportano nell'ordine i valori digitalizzati di V_1 e V_2 .

9. Ripetete la procedura per qualche valore di I_B e graficate le curve I_C vs V_{CE} in un unico grafico.

10. Per la conversione delle tensioni digitalizzate in unità fisiche, potete misurare la d.d.p. tra porta 7 di Arduino (boccola rossa) e terra, che, in prima approssimazione, si trova al valore V_{ref} , consentendo di determinare il fattore di conversione ξ . Se siete in ritardo, accontentatevi del valore "nominale" $\xi = 5 \text{ mV/digit}$.



11. Determinate (dalle curve) il valore del guadagno in corrente continua $\beta_F = I_C/I_B$ per diversi valori di I_B e un certo valore di V_{CE} (si suggerisce $V_{CE} \sim 1 \text{ V}$); riportatelo in tabella, valutando in maniera coscienziosa l'incertezza associata.

12. Facoltativo: stimare il valore del guadagno in corrente per "deboli segnali" a una certa corrente (si suggerisce $I_B \sim 10\text{-}15 \mu\text{A}$), definito qui come $\beta_f \approx \Delta I_C/\Delta I_B$ (Δ indica la differenza tra valori acquisiti per valori di I_B "prossimi" tra loro).

13. Facoltativo, ma consigliato: scegliete un certo valore di I_B (da dichiarare) e selezionate il tratto della curva I_C vs V_{CE} che vi aspettate corrispondere sicuramente al "regime attivo"; fatene un grafico e un best-fit secondo una retta non passante per l'origine, con lo scopo di determinare il parametro V_{Early} definito dalla:

Riportate il valore di V_{Early} e tutte le informazioni per la comprensione del best-fit, oltre a ogni eventuale altro commento, nel riquadro qui sotto o sui fogli dei grafici.

$$I_C \approx \beta_F I_B \left(1 + \frac{V_{CE}}{V_{Early}} \right)$$

$V_{porta\ 7} = V_{ref} \text{ [V]}$	$\xi \text{ [mV/digit]}$

$V_{CE} \sim \text{ [V]}$		
$I_B \text{ [\mu A]}$	$I_C \text{ [mA]}$	$\beta_F = I_C/I_B$

$I_B \sim \text{ [\mu A]} ; V_{CE} \sim \text{ [V]}$		
$\Delta I_B \text{ [\mu A]}$	$\Delta I_C \text{ [mA]}$	$\beta_f = \Delta I_C/\Delta I_B$

Commenti sull'eventuale best-fit e su ogni altro aspetto secondo voi rilevante::