

## Práctica de laboratorio 3

### El Transistor BJT

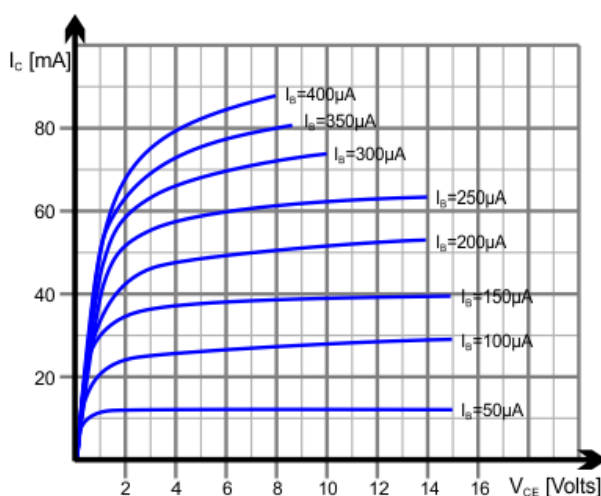
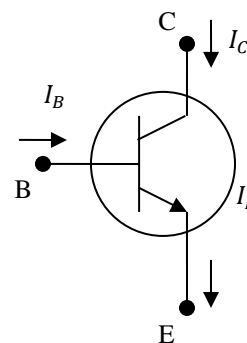
#### 1. Objetivos de la práctica.

- Estudio del transistor BJT.
- Profundización en el uso de Arduino.
- Realización de un sistema de control de un motor de corriente continua.

#### 2. Introducción teórica.

El transistor bipolar o BJT está formado por un semiconductor dividido en tres zonas de dopaje: una zona denominada Emisor, otra Base y la última Colector. Según el dopaje de cada zona (Tipo n o p) un transistor puede ser del tipo NPN o PNP. Podemos observar el símbolo de un transistor del tipo NPN en la siguiente figura adjunta.

Cuando un transistor no está polarizado es similar a dos diodos contrapuestos, diodo Base-Emisor y diodo Base-Colector, con una tensión de codo de 0.6 a 0.7 V en el caso de que sean de silicio (0.2 a 0.3 V en caso del germanio). En todo transistor se cumple siempre que  $I_E = I_B + I_C$ . Según como lo polaricemos éste puede encontrarse en tres zonas de trabajo. La curva característica del transistor BC547 puede apreciarse en la siguiente gráfica:



Regiones de trabajo:

**Corte:** Producida cuando la intensidad de emisor ( $I_E$ ) es cero. Las uniones B-C y B-E están polarizadas en inversa, de manera que  $I_B = -I_{CBO} \approx 0$ .

**Activa:** Se produce cuando la intensidad de base ( $I_B$ ) es distinta de cero, y la intensidad que circula por el colector ( $I_C$ ) se mantiene proporcional a ésta cumpliendo  $I_C = \beta I_B$  (donde  $\beta$  es la

ganancia del transistor<sup>1</sup>). La unión B-C está polarizada en inversa y la unión B-E en directa. Podemos suponer  $V_{BE} \approx 0.7 \text{ V}$  para Si y  $V_{BE} \approx 0.2 \text{ V}$  para Ge.

**Saturación:** Se produce cuando al aumentar la intensidad de base ( $I_B$ ), la intensidad que circula por el colector ( $I_C$ ) alcanza un valor máximo (dado por el propio circuito), a partir del cual se mantiene constante. Si seguimos aumentando más la  $I_B$  se cumple que  $I_C < \beta I_B$ . Podemos suponer en esta zona la tensión  $V_{CE} \approx 0.2 \text{ V}$  y la tensión  $V_{BE} \approx 0.8 \text{ V}$ . La unión B-C y la unión B-E están polarizadas en directa.

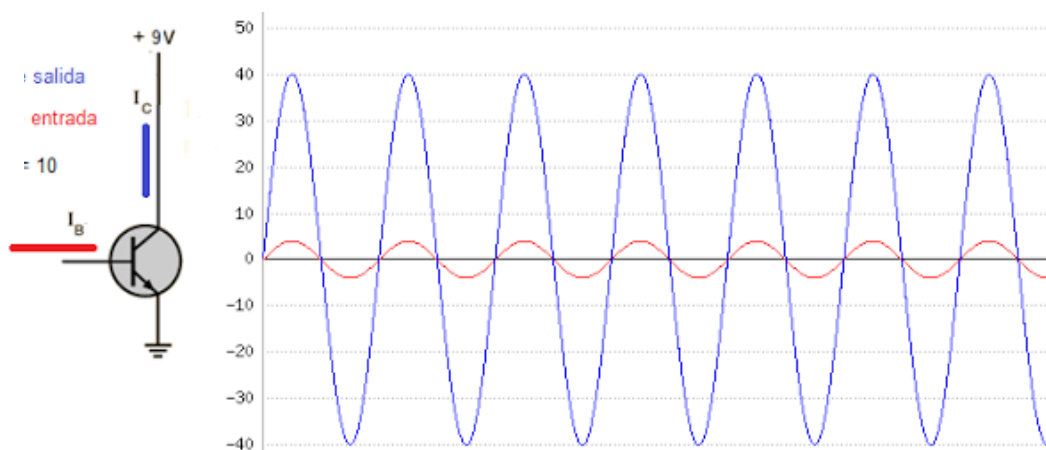
Entre las diferentes aplicaciones del transistor BJT se pueden destacar las siguientes:

## Inversor

El funcionamiento del transistor BJT es muy útil tanto para funciones digitales (puesto que permite distinguir entre dos estados, que es la base de la electrónica digital), como para controlar, accionando y parando, algún dispositivo (que será nuestro caso en el montaje correspondiente al control del motor). Para ello, se obtendrá en la salida una tensión próxima a la de alimentación (valor alto de tensión) así como una próxima a cero (valor bajo de tensión), trabajando el transistor en las zonas de corte (para el primer caso) y saturación (para el segundo caso).

## Amplificador

La necesidad de amplificar las señales es casi una necesidad constante en la mayoría de los sistemas electrónicos. En este proceso, los transistores desarrollan un papel fundamental, pues bajo ciertas condiciones, pueden entregar a una determinada carga una potencia de señal mayor de la que absorben. Para ello, el transistor o transistores de dicho circuito trabajan en la zona de activa directa.



<sup>1</sup> Voluntariamente se ha querido simplificar el funcionamiento del transistor despreciando la influencia de  $I_{CB0}$ .

### 3. Instrumentación y Componentes.

- Placa Arduino UNO, junto con placa de prototipado.
- Transistor BJT BC547.
- 2 resistencias de 1 k $\Omega$ .
- Motor de continua.
- Fuente de alimentación (5 V).
- Pulsador.
- Potenciómetro.
- Multímetro.
- Cables de conexionado.
- Cable USB.

### 4. Estudio Teórico.

Dado el esquemático de la siguiente figura, y teniendo en cuenta que la  $\beta$  del transistor tiene un valor de 100, analice qué valores de V1 hacen conmutar el transistor entre las diferentes regiones.  $V_{CC} = 5V$ ,  $R1 = 1k\Omega$ ,  $R2 = 100\Omega$ ,  $V_{BE}(ZAD) = 0.7V$ ,  $V_{BE}(SAT) = 0.8V$  y  $V_{CE}(SAT) = 0.2V$ .

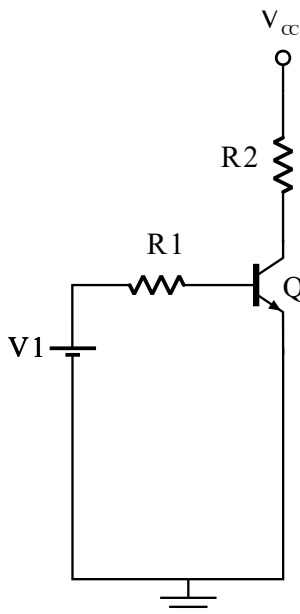
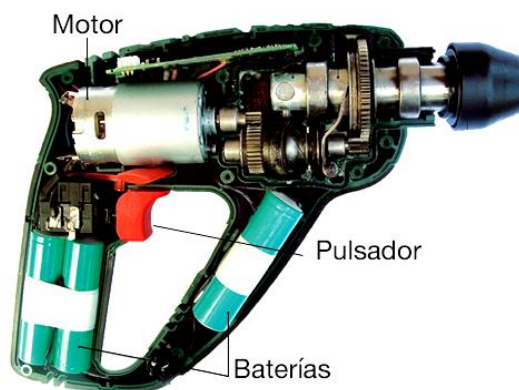


Figura 1.- Circuito para estudio teórico.

## 5. Desarrollo de la práctica.

### Montaje 1.

Este montaje se propone para la realización de un sistema dedicado a controlar el accionamiento de un motor de corriente continua, como el empleado en algunos tipos de los denominados coches eléctricos (ver siguiente figura, en la cual se muestra un taladro como ejemplo de uso de motor DC). Para ello únicamente se utilizará la placa Arduino UNO y el motor de continua. El control será de tipo todo o nada, es decir, se pondrá en funcionamiento el motor (marcha) o se detendrá (paro). Para ello se dispondrá de un pulsador que permitirá la entrada de eventos por parte del usuario.



Realizar los siguientes pasos:

- 1 Realice el montaje de la siguiente figura utilizando como alimentación del motor DC una salida digital. En este montaje se utilizará un diodo con el objetivo de proteger el puerto digital de la plataforma Arduino frente a las tensiones negativas originadas por el carácter inductivo del motor.
- 2 Considere la entrada de un pulsador como evento para el accionamiento marcha/paro en el motor.
- 3 Programe el microcontrolador de tal forma que al accionar el pulsador cambie el estado del motor: si estaba en ON debe pasar a OFF y viceversa. Puede basarse en el siguiente código:

```
const int motorPin = 9; // motor conectado al PIN 9
const int pulsadorPin = 2; // entrada de usuario para marcha/paro
int marchaParo = LOW; // por defecto hacemos el paro del motor
int eventoPulsacion = LOW; // pulsación == HIGH, no pulsación == LOW
void setup() {
    pinMode(motorPin, OUTPUT);
    pinMode(pulsadorPin, INPUT);
    Serial.begin(9600); // monitor serie
    Serial.println("CONTROL MOTOR CC");
    Serial.print("Inicialmente, marchaParo = ");
    Serial.println(marchaParo);
}

void loop() {
    eventoPulsacion = digitalRead(pulsadorPin); // leemos la pulsación
    if (eventoPulsacion == HIGH) { // pulsación
        if (marchaParo == HIGH) { // si estaba en marcha
            marchaParo = LOW; // paramos
        }
        else { // si estaba en paro
            marchaParo = HIGH; // hacemos la marcha
        } // si marchaParo == true, motor ON, si marchaParo == false, motor OFF
        digitalWrite(motorPin, marchaParo);
        Serial.print("marchaParo = ");
        Serial.println(marchaParo);
    }

    delay(300); // espera 30 ms para que el efecto sea visible
}
```

- 4 ¿Qué observa al accionar el pulsador? ¿Cómo se podría incrementar la velocidad de giro del motor? Justifique su respuesta.

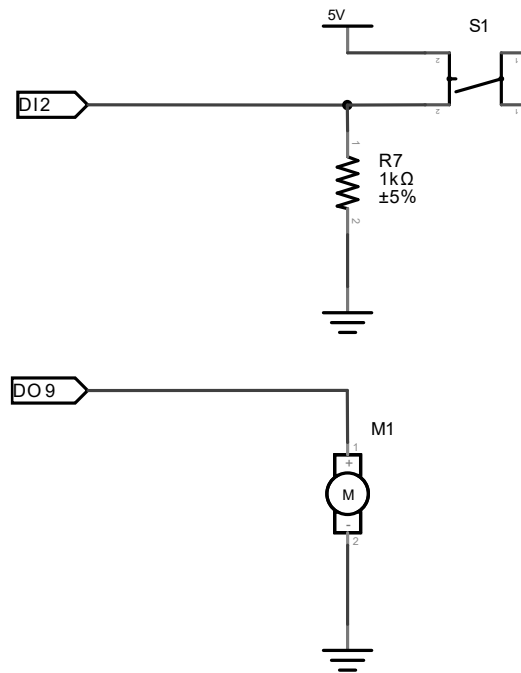


Figura 2.- Esquemático 1.

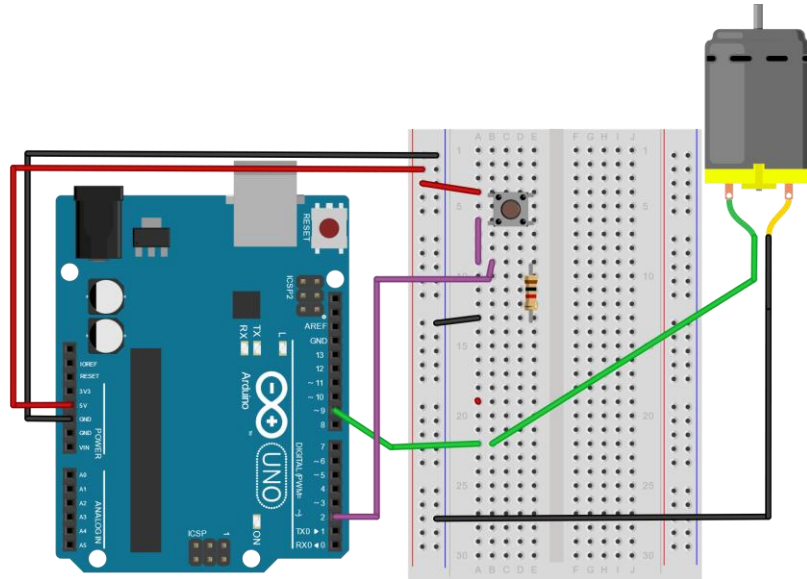


Figura 3.- Montaje 1.

## Montaje 2.

Este montaje tiene como objetivo el control de la velocidad del motor DC estudiado en el montaje anterior. En este caso, se pretende mejorar las prestaciones obtenidas, además de habilitar un control más flexible de la velocidad (no sólo marcha-paro, sino distintos puntos de velocidad). Para ello, se utilizará un transistor BJT y un potenciómetro, además de los elementos utilizados en el montaje anterior, y otros elementos adicionales necesarios para cumplir las especificaciones asociadas a la funcionalidad buscada.

Realizar los siguientes pasos:

- 1 Utilice la plataforma Arduino UNO para realizar el montaje de la figura 5. El control de la velocidad de rotación del eje del motor se realizará mediante el uso de un potenciómetro. El pulsador se utilizará para generar el marcha-paro.
- 2 Realice el programa de control de la plataforma Arduino UNO con el propósito de controlar el accionamiento del motor con el pulsador.
- 3 Pruebe el correcto funcionamiento del motor.
- 4 Dibuje en una gráfica diversos pares de valores de tensión  $V_i$ , eje de abscisas, y entre colector y emisor,  $V_o$ , eje de ordenadas, (ver figura 5). ¿Qué forma tiene la curva obtenida? ¿En qué zonas está trabajando el transistor en cada momento cuando se actúa sobre el potenciómetro? Justifique su respuesta.
- 5 Compruebe el consumo de potencia que requiere el motor a máxima velocidad. ¿Cuándo es mínimo?

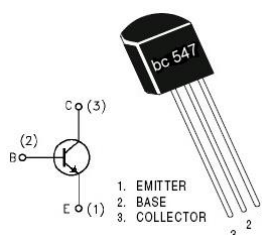


Figura 4.- Transistor BJT.

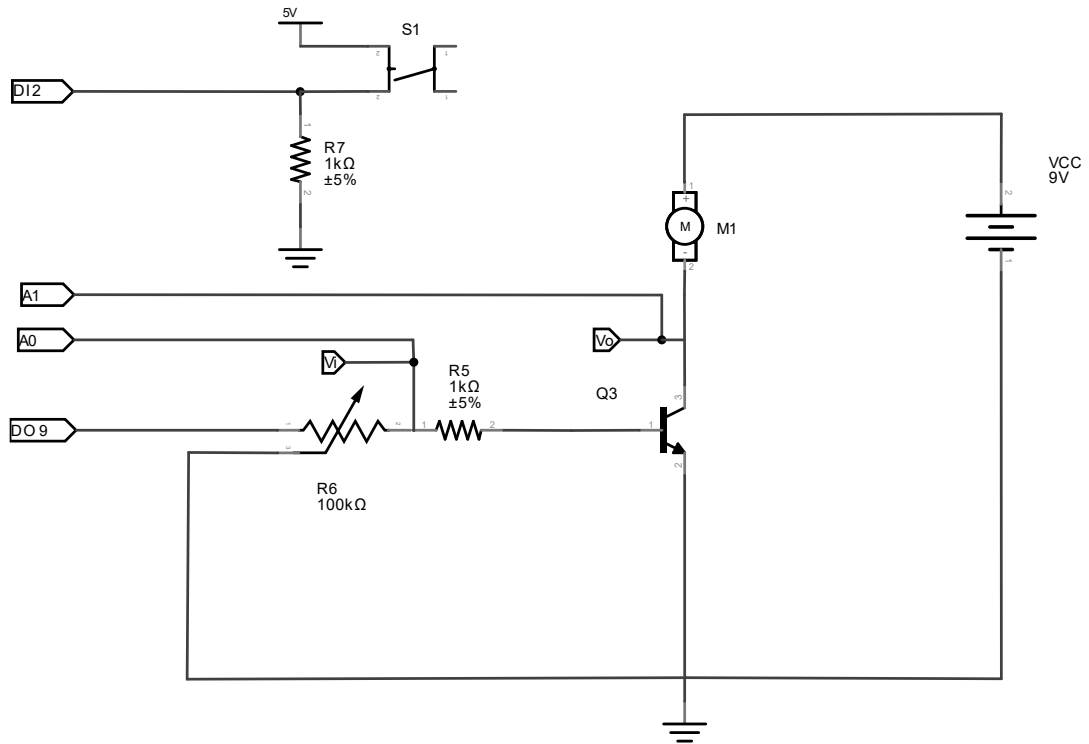


Figura 5.- Esquemático 2