

1.2.1. Hacer que un LED parpadee (Ref libro 1 pg 130)

El primer programa que realizaremos consiste en hacer que un LED parpadee. Para ello vamos a utilizar el LED que incluye la placa ZUM BT, casi todas las placas Arduino y compatibles incluyen un LED integrado. Este LED, salvo excepciones, se controla desde el PIN 13 de la placa.

Componentes necesarios para la actividad:

- Placa ZUM BT, Arduino o compatible.
- Cable USB para conectar la placa al PC

En este caso, como ya se ha comentado, no necesitamos conectar nada a la placa ya que el LED está integrado.

Para crear este programa en BitBloq conectamos nuestra placa al PC mediante USB y usamos el conexionado que se muestra en la Figura 1.2.1-1 en la pestaña de Hardware.

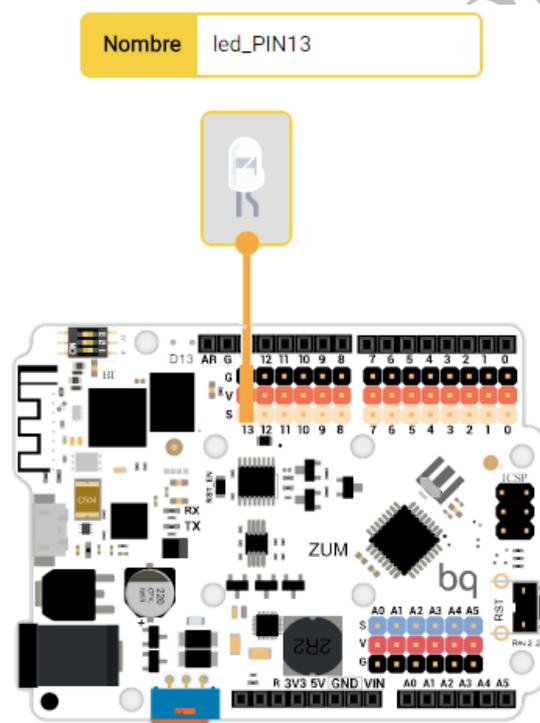


Figura 1.2.1-1. Conexionado Hardware del LED en BitBloq.

Se debe conectar un bloque LED al PIN 13 de la placa. En nuestro caso hemos llamado a este componente “led_PIN13”.

Una vez definidos todos los componentes y su conexionado, pasamos a la pestaña de Software de BitBloq. Para nuestro primer programa tan solo vamos a usar el “Bucle principal (Loop)”. Recordamos que en el Bucle Principal se repiten los comandos de forma cíclica.

Libro de Actividades de Robótica Educativa

Primero vamos a hacer que el LED se encienda, pero no que se apague. En la pestaña “Componentes” encontramos los controles para el LED, no aparece nada más ya que solo definimos el LED en la parte Hardware. Arrastramos el componente al “Bucle Principal (Loop)”.

– Bucle principal (Loop)



El bloque LED tiene dos elementos configurables.

1. Permite decir si queremos Encender o Apagar el LED.
2. Indica el LED que estamos controlando, en este caso solo tenemos el “led_PIN13”.

Figura 1.2.1-2. Programa en BitBloq para encender un LED.

Con esto termina el programa, ahora solo hay que descargarlo en la placa. Pulsamos el botón “Cargar” y el programa se descarga en la placa y comienza su ejecución. Al terminar el LED integrado en la placa debe encenderse como se muestra en la Figura 1.2.1-3.

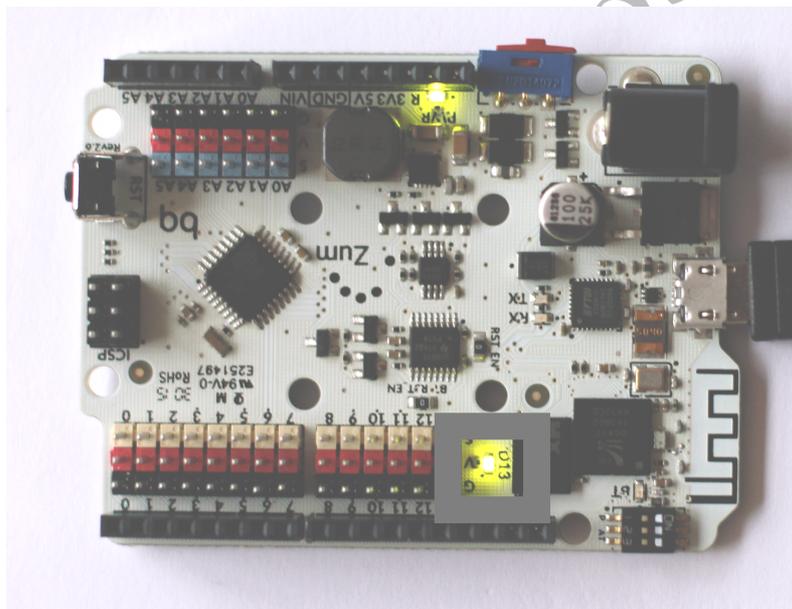


Figura 1.2.1-3. Resultado del programa para encender un LED.

La finalidad de esta actividad es que el LED del PIN 13 parpadee, por ahora solo lo hemos encendido. Para que un LED parpadee hace falta encenderlo y apagarlo cada un tiempo determinado.

Volvemos a la pestaña de Software de Bitbloq y vamos a añadir nuevos bloques a nuestro programa. En esta ocasión vamos a la pestaña “Control”, en ella se encuentra el bloque “Esperar”, lo arrastramos y lo ponemos debajo de nuestro código. Añadimos bloque hasta que obtenemos un código como en que se muestra en la Figura 1.2.1-4.

– Bucle principal (Loop)

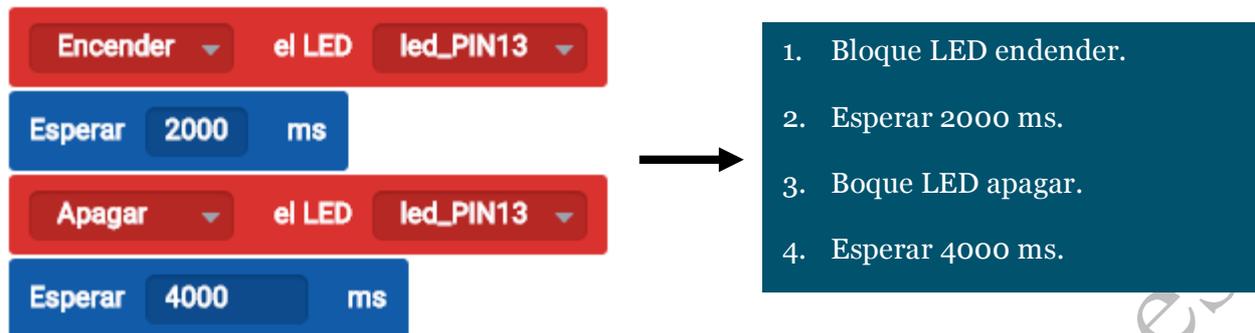


Figura 1.2.1-4. Programa en Bitbloq para parpadeo de un LED.

Una vez terminado se descarga en la placa y el LED de la placa debe comenzar a parpadear permaneciendo 2 segundos encendida y 4 segundos apagada. El programa es muy sencillo:

1. En la primera línea usamos el bloque de LED y le decimos que se encienda. Si ya está encendido no hará nada
2. Con el bloque Esperar le decimos cuanto tiempo debe esperar para leer la siguiente línea de código, en nuestro caso 2000 ms, así aseguramos que el LED permanece encendido 2 s.
3. Cuando termina la espera se lee la tercera línea, en este caso un bloque LED que nos dice que se debe apagar.
4. La cuarta línea contiene un nuevo bloque Esperar, en este caso con 4000 ms, por tanto, el LED permanece apagado 4s.
5. Como estamos en el Bucle Principal, una vez se termina de leer la cuarta línea se vuelve a la primera.

Los valores de los bloques Esperar se pueden modificar para modificar la velocidad del parpadeo.

A continuación, podemos ver el código en Arduino para este programa:

```
/** Global variables and function definition */  
const int led_PIN13 = 13;  
  
/** Setup */  
void setup() {  
  pinMode(led_PIN13, OUTPUT);  
}  
/** Loop */  
void loop() {  
  digitalWrite(led_PIN13, HIGH);  
  delay(2000);  
  digitalWrite(led_PIN13, LOW);  
  delay(5000);  
}
```