

## **1.2.2. Cambiar la luminosidad de un LED (Ref libro 1 pg 131)**

La segunda actividad que vamos a realizar está muy vinculada con la primera, consiste en hacer que un LED varíe su luminosidad. Para ello vamos a utilizar el LED que incluye el ZUM KIT, por tanto vamos a conectar nuestro primer dispositivo externo a la placa.

Componentes necesarios para la actividad:

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

Vamos a conectar el LED a la placa, es importante destacar algo que ya se comentó, el PIN 13 no se puede usar para conectar un LED externo por tanto se debe elegir alguno de los doce anteriores. Además, se nos presenta otra limitación, para realizar esta actividad necesitamos una salida PWM de la placa. Más adelante se explicará que es el PWM. Por el momento solo debemos saber que estas son salidas digitales que trabajan de forma similar a una analógica, es decir los valores no se limitan a encendido (1) y apagado (0).

Las placas Arduino y compatibles, de forma general, llevan los PINES 11, 10, 9, 6, 5 y 3 habilitados para el uso de PWM, por tanto debemos elegir uno de estos para conectar nuestro LED.

Por tanto la conexión que vamos a realizar es:

- PIN 11 con LED.

En las placas ZUM BT con el LED del ZUM KIT es muy sencillo, solo hay que asegurarse de que los colores de los cables coinciden con los de la placa. Para otras placas hay que conectar el PIN 11 al cable de datos del LED y se deben conectar alimentación y tierra a los pines con este fin. En la Figura 1.2.2-1 puede verse el LED conectado a la placa ZUM BT.

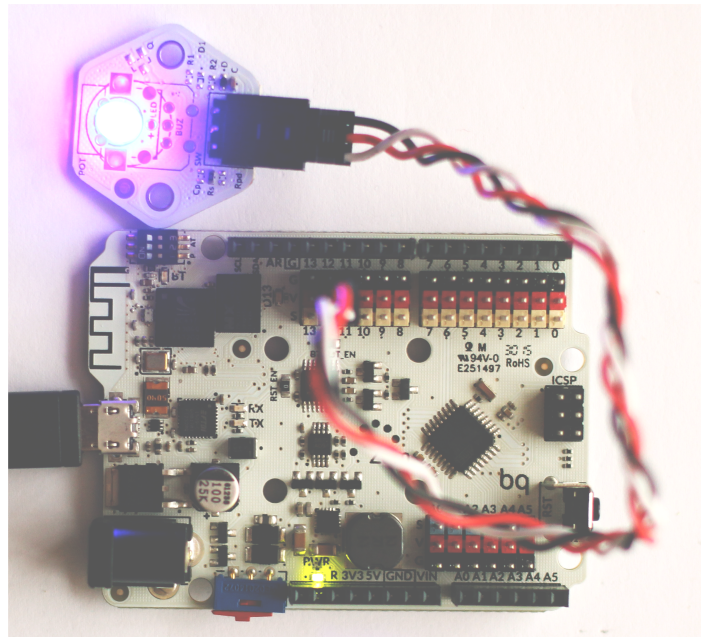


Figura 1.2.2-1. Conexión del LED a la placa ZUM BT

Una vez conectado el LED comenzamos con nuestro programa en BitBloq. Conectamos nuestra placa al PC mediante USB y usamos el conexionado que se muestra en la Figura 1.2.2-2 en la pestaña de Hardware.

Nombre led\_Ext\_1

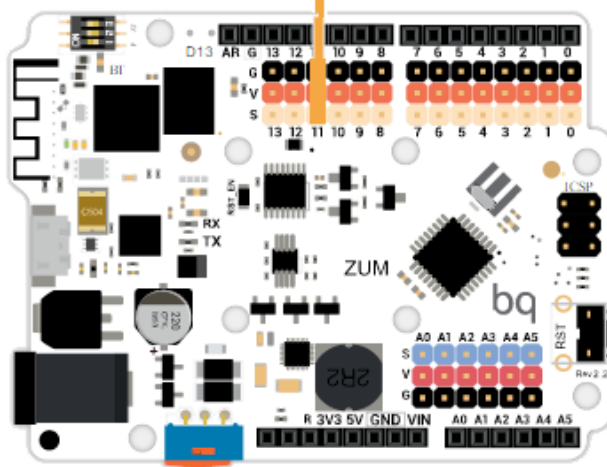


Figura 1.2.2-2. Conexión del LED externo en BitBloq.

## Libro de Actividades de Robótica Educativa

Se conecta el Bloque de LED, como en el ejercicio anterior, pero esta vez al PIN 11. El nombre que hemos usado para nuestro componente ha sido “led\_Ext\_1”.

Ahora vamos a la pestaña de “Software” y comenzamos nuestro programa. Como en la actividad anterior, vamos a trabajar sobre el “Bucle principal (Loop)”. Vamos a la pestaña “Componentes” y en la parte inferior pinchamos en “Avanzados”. Arrastramos “Escribir en pin analógico”. En el primer campo ponemos “Variable (componentes)” y en el segundo vamos a la pestaña “Matemáticas” y cogemos el bloque de variable numérica.



Figura 1.2.2-3. Programa en BitBloq para cambiar la intensidad de un LED.

Ponemos el valor que queramos entre 0 y 255 en la variable numérica, 0 quiere decir que el LED está apagado y 255 que brilla con su mayor intensidad. Le damos a cargar y nuestro LED externo lucirá con una intensidad proporcional al valor de la variable numérica. SI queremos otra intensidad, variamos el valor y volvemos a cargar.

En esta actividad, como dijimos al principio, se pretende utilizar una salida digital como analógica, para ello se usa la codificación PWM. En las opciones avanzadas de componentes nos permiten forzar este tipo de salidas con el bloque “Escribir en el pin analógico”, el primer campo que se ha usado es donde se quiere escribir, en nuestro caso en el componente “led\_Ext\_1” y el último campo es una variable numérica que indica que campo vamos a escribir, no tiene por que ser una variable numérica, se podría haber usado cualquier otro dato numérico, por ejemplo la lectura de otro dispositivo.

El código en Arduino es el siguiente:

```
/** Global variables and function definition */  
const int led_Ext_1 = 11;  
  
/** Setup */  
void setup() {  
  pinMode(led_Ext_1, OUTPUT);  
}  
  
/** Loop */  
void loop() {  
  analogWrite(led_Ext_1, 0);  
}
```

El código en Arduino es muy sencillo. Se asigna el valor 11 a la variable “led\_Ext\_1”, se define el PIN 11 como salida por último en el bucle de ejecución se escribe en el PIN 11 el valor deseado, en este caso 0.