

### 1.2.3. Encender un LED utilizando un pulsador (Ref libro 1 pg 82 pero como 1.2.2 y pg 132 como 1.2.3)

La tercera actividad, es nuestra última actividad con el LED. En esta ocasión vamos a accionar el LED mediante otro dispositivo, es pulsador. El LED permanecerá activado mientras el botón esté pulsado y se desactivará al soltarlo.

Componentes necesarios para la actividad:

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

La conexión que vamos a realizar, tal y como muestra la Figura 1.2.3-1, es:

- PIN 11 con LED.
- PIN 7 con pulsador.

La elección de los pines puede ser distinta a la propuesta, en este caso no vamos a usar la salida PWM ya que solo queremos encender y apagar el LED y el pulsador puede conectarse en cualquier PIN. La única excepción es que el PIN 13 no puede usarse para el LED externo.

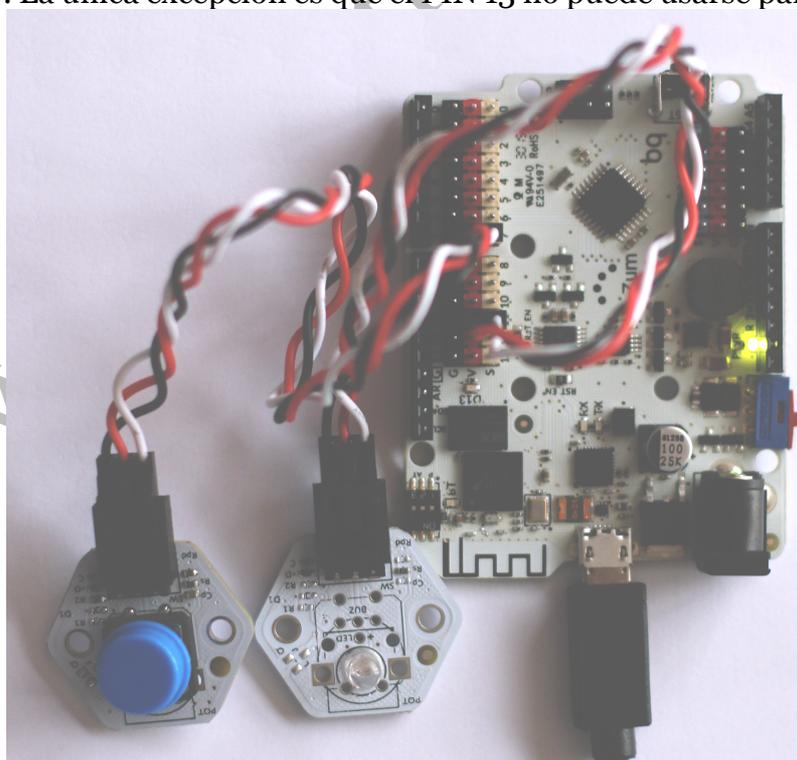


Figura 1.2.3-1. Conexión física del LED y el pulsador.

Una vez conectados los componentes a la placa comenzamos con nuestro programa en BitBloq. Conectamos nuestra placa al PC mediante USB y usamos la configuración hardware que se muestra

## Libro de Actividades de Robótica Educativa

en la Figura 1.2.3-2. Los nombres que se han asignado a los componentes son “led\_Ext\_1” para el LED y “boton\_1” para el botón.

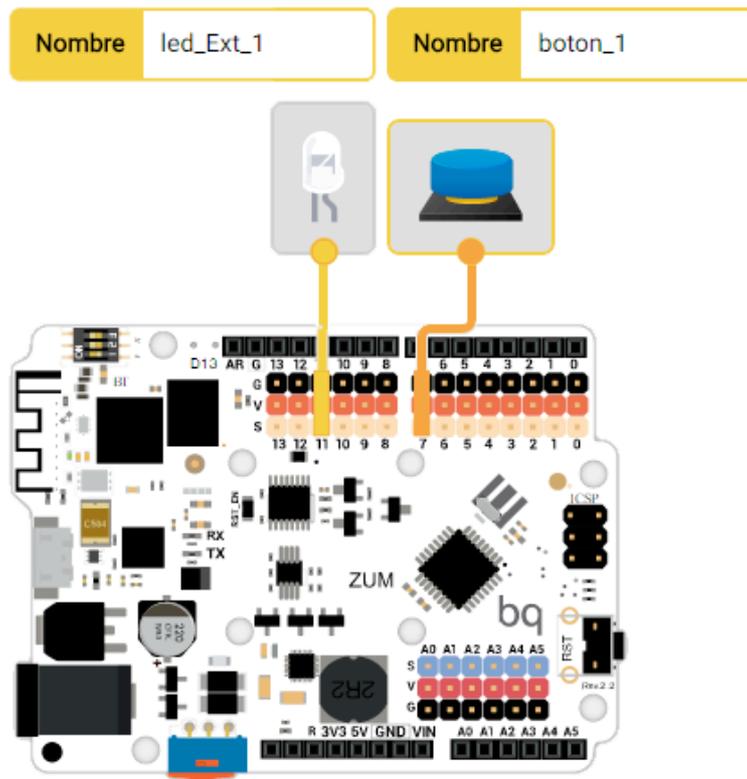
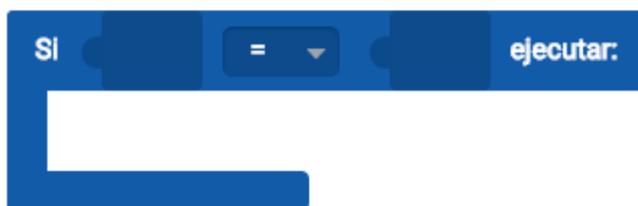


Figura 1.2.3-2. Conexión Hardware en BitBlok de encendido de un LED con un pulsador.

Pasamos a la pestaña de software para comenzar la programación. Para este programa vamos a introducir un concepto que no habíamos empleado en las actividades anteriores, la acción condicional. Estas acciones si ejecutan SI se cumple una condición, el primer tipo de condición que empleamos es la conocida como IF (SI en castellano).

De nuevo programaremos en el “Bucle principal”. Abrimos la pestaña “Control” y buscamos la sentencia “Si” que se muestra en la Figura 1.2.3-3.

### — Bucle principal (Loop)



La sentencia If tiene tres campos:

1. Variable o cte 1 que se compara.
2. Tipo de comparación (=, <, >, etc).
3. Variable o cte 2 que se compara.

Figura 1.2.3-3. Sentencia If en BitBlok.

El valor que se quiere comparar es el estado del botón (si está pulsado o no), por tanto arrastramos al primer campo del If “Leer boton\_1” que se encuentra en la pestaña de componentes tal y como muestra la Figura 1.2.3-4. Hasta ahora nunca habíamos usado un

componente botón, como puede verse, solo se permite la lectura ya que no tiene sentido mandar datos a un botón.

### – Bucle principal (Loop)

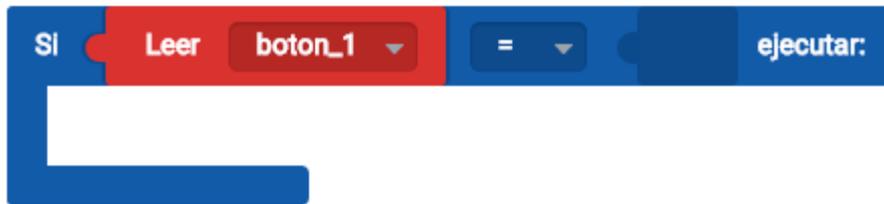


Figura 1.2.3-4. Primera condición If para leer el botón.

Queremos ver si el botón está pulsado o no. Cuando el botón está pulsado se envía Verdadero, es decir un 1 en binario, cuando no está pulsado se envía Falso, o lo que es lo mismo un 0. Por tanto la comparación que se realizará es comprobar si el estado leído del botón es verdadero o falso. En la Figura 1.2.3-5 vemos como queda la sentencia condicional.

### – Bucle principal (Loop)

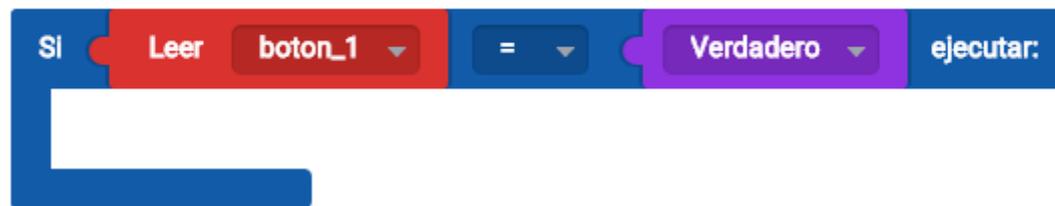


Figura 1.2.3-5. Sentencia IF para comprobar el estado del botón.

Con la sentencia condicional comprobamos si el botón está pulsado, pero no realizamos ninguna acción si se cumple la condición. Lo siguiente que debemos hacer es decir que acciones se realizan si se cumplen las condiciones impuestas. Para nuestra actividad queremos que el LED se encienda cuando el botón está pulsado, así que debemos ir a la pestaña de componentes y arrastrar el bloque de encendido del LED dentro de la sentencia condicional, de forma que se enciende cuando se cumple la condición de que el botón esté pulsado. Podemos ver como queda en la Figura 1.2.3-6.

### – Bucle principal (Loop)

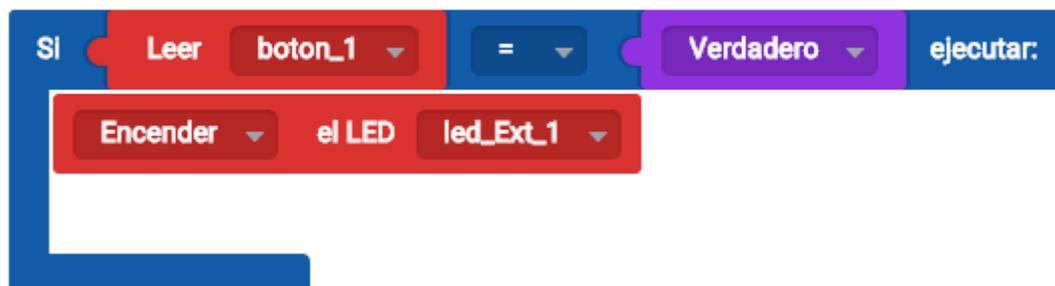


Figura 1.2.3-6. Encendido del LED si se pulsa el botón.

Con esto conseguimos que el LED se encienda al pulsar el botón, pero, no se apaga ya que no hemos puesto ninguna condición para el apagado. Para solucionar este problema vamos a usar la condición If-Else (Se podría traducir como Si-Si no), esta ampliación de la estructura condicional nos permite realizar otra acción si no se cumple la condición del If. En la pestaña “Control” encontramos el bloque “de lo contrario, ejecutar”, que nos permite hacer una acción si no se cumple el If (Si). Arrastramos este bloque bajo el Si y dentro añadimos un bloque para apagar el LED tal y como se muestra en la Figura 1.2.3-7.

– Bucle principal (Loop)



– Bucle principal (Loop)



Condición: si el botón esta pulsado

- Enciende el LED

Si no se cumple la condición

- Apagar el LED

Figura 1.2.3-7. Programa en BitBloq para encender un LED con un botón.

Si cargamos el programa en la placa podremos comprobar como al pulsar el botón el LED se enciende y al soltarlo se apaga.

## Libro de Actividades de Robótica Educativa

El código en Arduino es el siguiente:

```
/** Global variables and function definition */
const int led_Ext_1 = 11;
const int boton_1 = 7;

/** Setup */
void setup() {
  pinMode(led_Ext_1, OUTPUT);
  pinMode(boton_1, INPUT);
}

/** Loop */
void loop() {
  if (digitalRead(boton_1) == true) {
    digitalWrite(led_Ext_1, HIGH);
  } else {
    digitalWrite(led_Ext_1, LOW);
  }
}
```

El programa inicialmente define los nombres “led\_Ext\_1” y “boton\_1” y les asigna el valor de sus pines 11 y 7 respectivamente. Después define los pines que se van a usar y de que forma el PIN 11 que es el LED será de salida y el PIN 7 que es el botón será de entrada. En el loop se ejecuta la sentencia condicional, si la lectura del botón es 1 se pone el LED en HIGH y por tanto se enciende, en caso contrario el LED se pone en LOW y se apaga.

www.automatycayrobotics.com