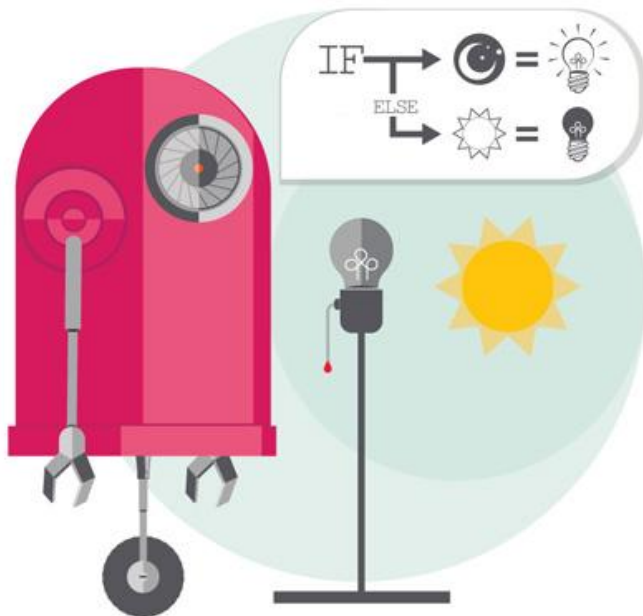


Programa la luz inteligente de tu bici



En este proyecto vas a descubrir cómo programar tu propia luz inteligente para ponerle tu bicicleta, con el fin de que ésta se encienda cuando detecte que es de noche.

Nivel de dificultad: Bajo

Tiempo estimado: 15 minutos

Materiales:

- Bicicleta de cartulina.
- 1 ordenador.
- Bitbloq.
- Placa controladora.
- 1 LED RGB.
- 1 sensor de luz o LDR.
- Cinta adhesiva.

¡A programar!

Antes de empezar, es importante que conozcamos cómo funciona Bitbloq. Para ello, vamos a crear un programa que encienda un LED.

¿Cómo conectar el LED a la placa?, ¿y en Bitbloq?

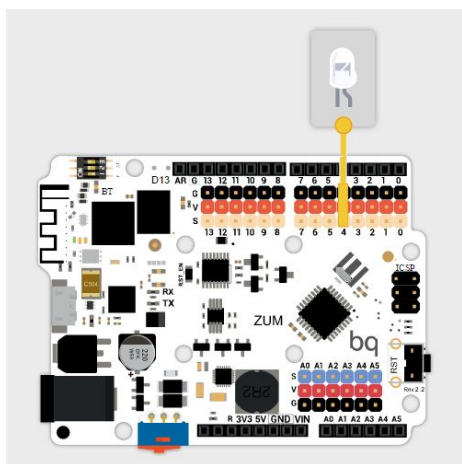
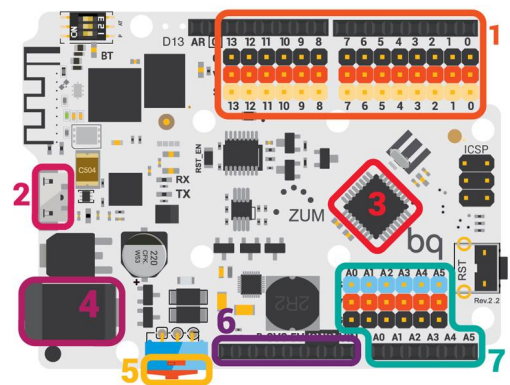
Antes de conectar un componente a la placa, deberemos pensar si es un componente **analógico**, aquellos que pueden tener más de dos valores, o **digital**, aquellos con solo dos valores (0 o 1).

En nuestro caso, el LED es un componente digital ya que solo tiene dos estados, o encendido (1) o apagado (0). Por ello, deberemos conectarlos a los pines digitales de nuestra placa controladora.

¿Cuáles son los pines digitales de la placa controladora?

Son los que corresponden con el número 1 de la siguiente imagen.

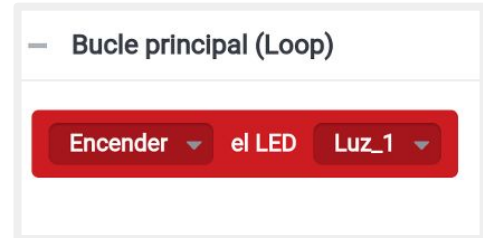
* Debemos intentar no utilizar los pines digitales 0 y 1 de la placa, ya que a través de estos dos pines, la placa se comunica con el ordenador y otros aparatos por lo que en ocasiones pueden dar problemas.



Una vez que tenemos el LED conectado a la placa, deberemos ir a Bitbloq y en la pestaña *Hardware*, arrastrar nuestra placa y el LED, conectándolo al mismo pin que utilizamos en la placa física.

¿Qué bloques necesito para encender un LED?

Una vez que has conectado tu LED, hay que programarlo para que se encienda. Para ello, solo debes seleccionar el bloque del LED en **Componentes** y **arrastrarlo dentro del bloque Loop()**.



Conecta tu placa al ordenador y clicas sobre el botón cargar. Tras cargarse el programa en tu placa, tu LED se encenderá.

Si tienes problemas para encender tu LED, consulta el siguiente enlace:

<http://diwo.bq.com/antes-de-empezar-con-bitbloq-2/>.

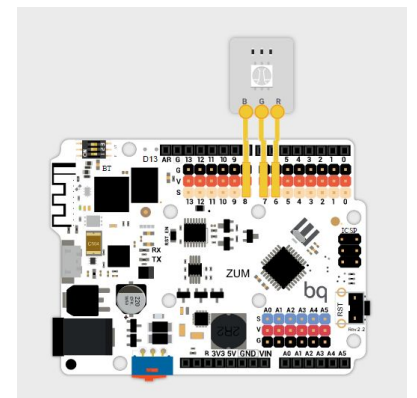
Con este programa hemos aprendido cómo encender un LED, y por tanto, cómo encender la luz de la bicicleta. Sin embargo, como la luz delantera y trasera deben tener diferentes colores, tendremos que utilizar el LED RGB.

¿Cómo programo un LED RGB?

Para programar nuestro LED RGB, lo que necesitamos conocer es lo siguiente:

- **¿Cómo lo conecto?**

Este componente tiene 4 cables. Debemos conectar el cable negro o GND a cualquiera de los pines digitales de color negro, correspondientes a tierra, y los otros cables a los pines digitales de señal, los mismos en los que los hayamos conectado en Bitbloq.



- **¿Cómo lo enciendo?**

Para encender el LED RGB podemos utilizar dos bloques:

- Uno más sencillo, para indicar en qué color de los que aparecen en el desplegable del bloque se tiene que encender el LED RGB.



- Otro más complejo, para indicar un color específico. De esta manera, pondremos en el siguiente bloque los valores de los colores primarios de luz (rojo, azul y verde) que forman dicho color. Para consultar estos valores podemos utilizar este enlace http://rapidtables.com/web/color/RGB_Color.htm

Encender el led RGB **led_RGB_0** con un valor de rojo de **0** , un valor de verde de **0** y un valor de azul de **0**

- **¿Cómo lo apago?**

Para que permanezca apagado, deberemos utilizar el siguiente bloque y poner todos los valores a 0.

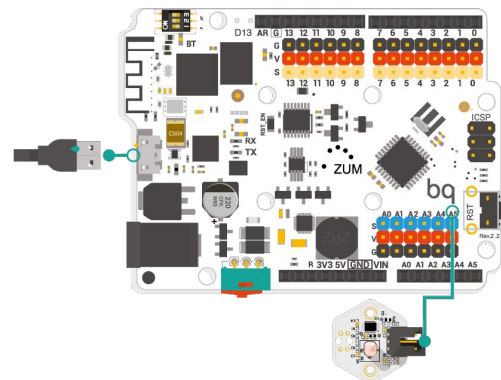
Encender el led RGB **led_RGB_0** con un valor de rojo de **0** , un valor de verde de **0** y un valor de azul de **0**

Ahora ya sabemos encender los LED, sin embargo, queremos hacer que ésta luz se encienda sola cuando detecte que es de noche. Para ello, deberemos utilizar el sensor de luz o ldr que nos dirá cuánta luz hay.

¿Cuánta luz hay?

Para conocer cómo funciona el ldr o sensor de luz y los valores que devuelve, vamos a comprobar la cantidad de luz que hay en nuestra aula.

En primer lugar, conectamos a nuestra placa un sensor de luz y el puerto serie.



Una vez que has conectados ambos componentes, hay que programarlos. Para ello, utilizaremos el apartado *Bucle principal (loop)* al que añadimos el bloque mostrar por *puerto serie*, al que incluiremos el bloque *Leer sensor de luz* y finalmente una *Espera* de unos pocos segundos:

— **Bucle principal (Loop)**

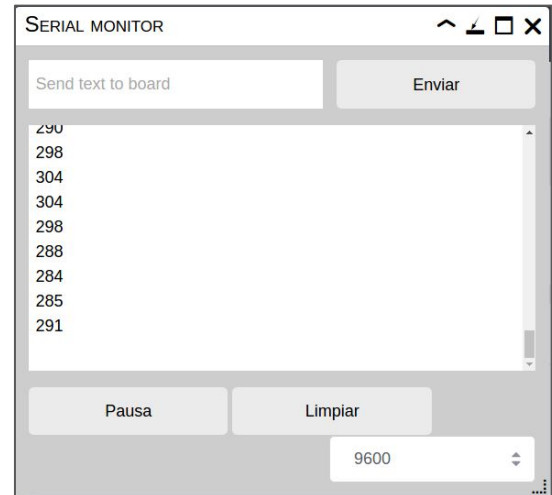
puerto_serie_0 Enviar Leer sensor_de_luz_0 Con salto de línea

Esperar 2000 ms

Conecta tu placa al ordenador, clicas sobre el botón cargar y selecciona “Mostar Serial Monitor” en la opción “Ver”. De esta forma, aparecerán los valores de luz detectadas por el sensor de luz en el aula.

Para más información, consulta los siguientes post:

- <http://diwo.bq.com/ve-hacia-la-luz-robotin-programando-el-sensor-de-luz/> - Sensor de luz
- <http://diwo.bq.com/comunicandose-con-puerto-serie-bitbloq2/> - Puerto serie



Creando nuestra luz inteligente

Ahora que ya sabemos cómo encender un LED y cómo funciona el sensor de luz (LDR), vamos a crear nuestra luz inteligente para la bici.

Como ya sabemos, es obligatorio que las bicicletas lleven una luz delantera y una trasera, para mejorar su visibilidad. Por ello, deberemos crear dos programas: uno para la luz delantera y otro para la luz trasera.

Ambos programas serán iguales, pero cambiando el color que debe mostrar el LED RGB. Para programar, deberemos seguir los siguientes pasos:

- Utilizamos un bloque *si* para indicar qué debe hacer el LED RGB cuando reciba mucha luz (por ejemplo, más de 200).
- Utilizamos el condicional *de lo contrario* para indicar qué debe hacer el LED RGB y de qué color tiene que ser cuando reciba poca luz.

Para comprobar que los programas que hemos creado funcionan, deberemos cargar el programa en nuestra placa y comprobar que cuando hay luz el LED no se enciende, y cuando no hay luz si se enciende. Podemos encender y apagar las luces de nuestra clase para simular el día y la noche.

Tras esto, incorporamos nuestra luz inteligente a la bicicleta que hemos construido.