ACTIVIDADES CON BITBLOQ

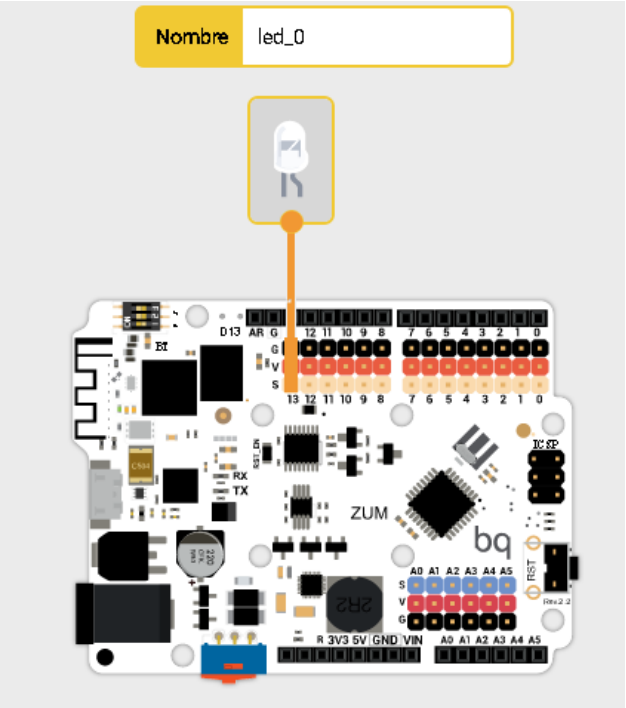
Bitbloq es una plataforma que permite integrar de forma virtual las tarjetas microcontroladoras, los componentes electrónicos que se van a controlar y el lenguaje de programación, tanto en forma de bloques, como con el entorno de desarrollo Arduino. Así mismo, permite realizar la exportación de los programas a la placa para que ésta los ejecute mediante las conexiones a los componentes necesarios.

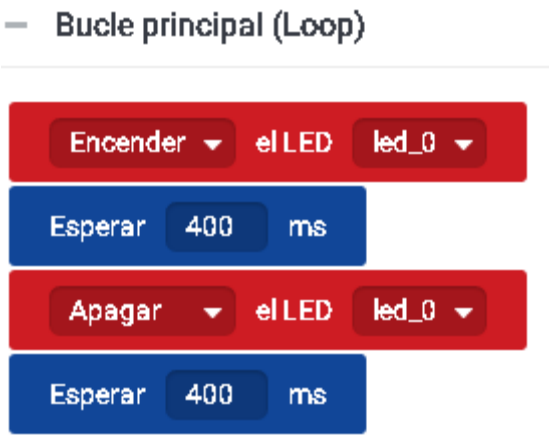
Vamos a partir de los componentes contenidos en una caja de Bitbloq llamado Zum-Kit. Contiene los componentes electrónicos más fundamentales así como la placa microcontroladora Zum.

Para programar con Bitbloq es necesario registrarse en la plataforma con un correo electrónico e introducir una contraseña (es fundamental anotar o recordar claramente la contraseña para no tener problemas). También es posible descargarse el programa de Bitbloq para trabajar off-line, es decir, sin conexión a Internet. Bitbloq requiere de un programa accesorio necesario para exportar la programación a la placa. Se denomina Web to Board y sirve para reconocer las características propias de la placa que se conecte y poder hacer la exportación correctamente.

En clase nos vamos a familiarizar con el entorno de trabajo e iremos introduciendo nociones básicas de programación en el transcurso de la realización de las prácticas.

01.- LED INTERMITENTE

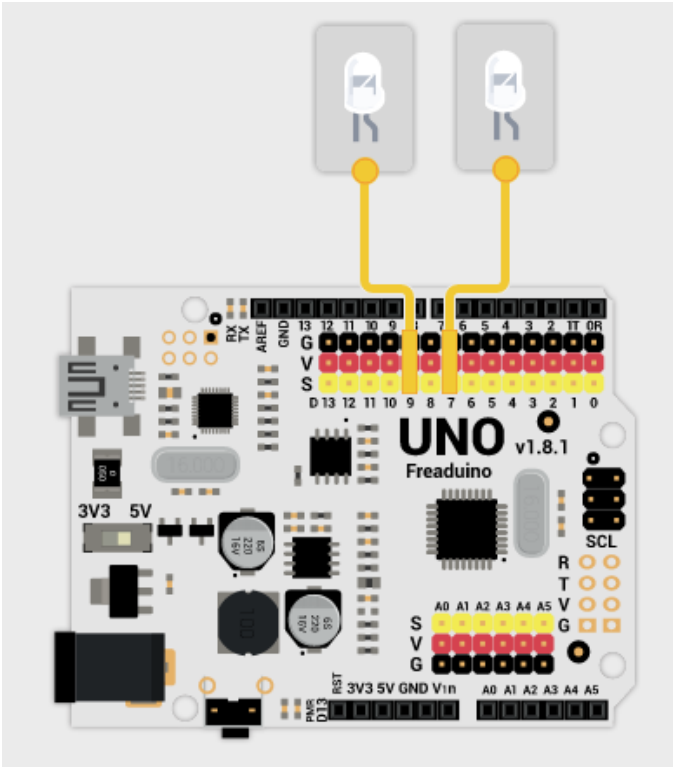




Prueba a cambiar el nombre del LED y mira qué sucede. Haz la conexión en otro pin de salida y comprueba cómo ha afectado.

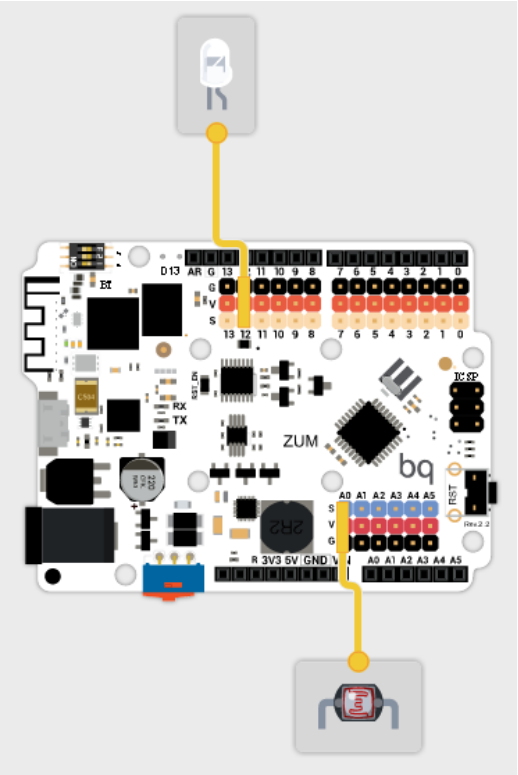
Como podéis apreciar los tiempos en Bitbloq se expresan en milisegundos.

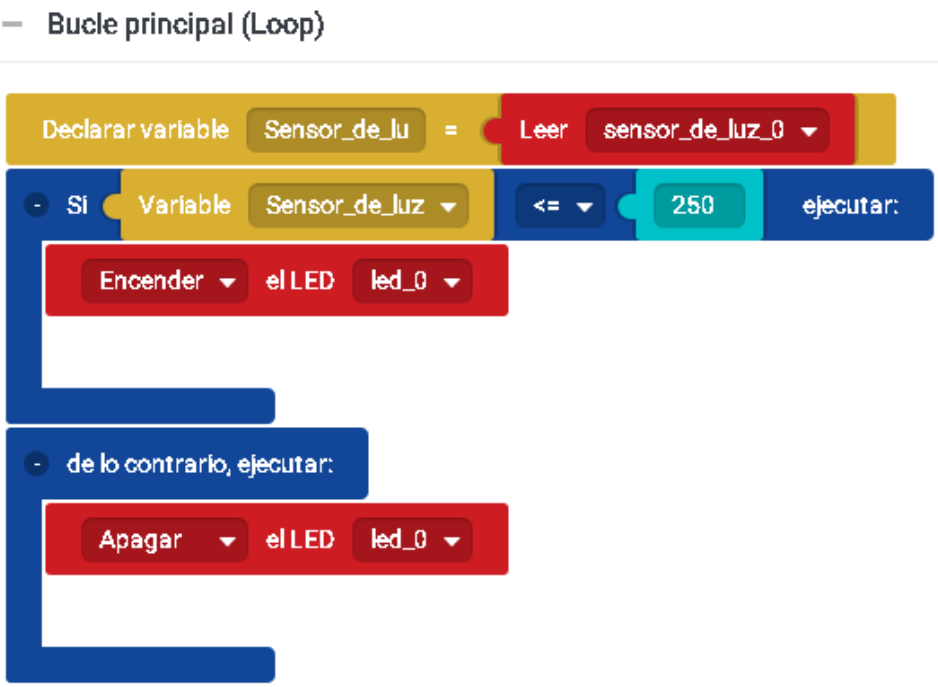
02.- DOS LED QUE LUCEN DE FORMA ALTERNATIVA



Experimenta realizando modificaciones en los tiempos de encendido y apagado.

03.- LED CONTROLADO POR LDR (CREPUSCULAR).

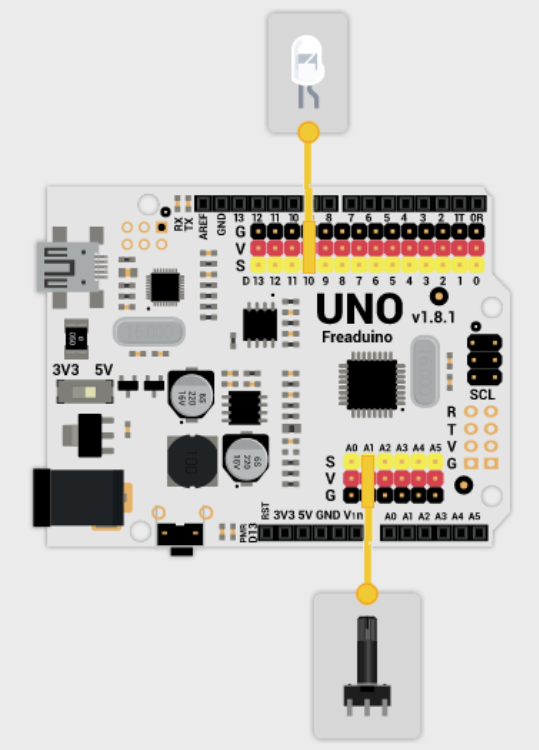




En este ejercicio hemos introducido el concepto de variable. De esta forma lo podremos incluir en ejercicios más complicados. En el presente ejercicio es posible hacerlo igualmente sin emplear una variable. Trata de hacerlo por tu cuenta y comprueba si funciona de la misma forma. Cambia la orden “de lo contrario, ejecutar:” por “en cambio, si…ejecutar”. Prueba a variar el valor del umbral de la intensidad lumínica para experimentar qué sucede.

Incorpora las imágenes de tus programaciones a la memoria que estás realizando.

04.- CONTROL DE UN LED CON POTENCIÓMETRO



Bucle principal (Loop)

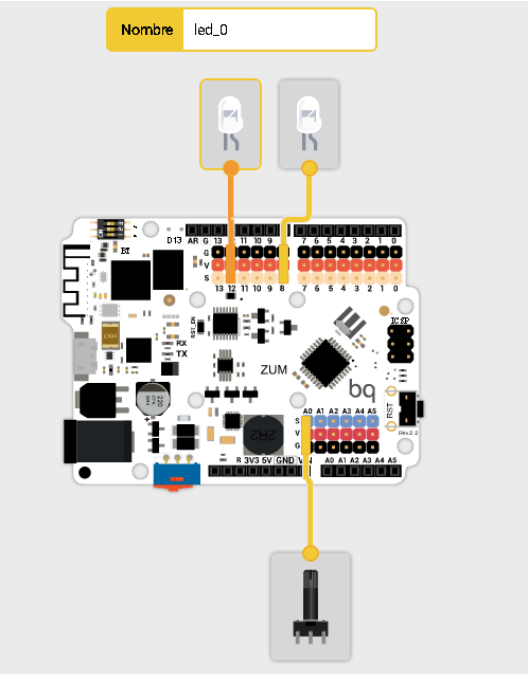
En este ejercicio se incorpora lo que se denomina mapeo. Lo que se consigue es asignar el valor numérico que deseemos a una variable determinada. En este caso, el valor de la resistencia del potenciómetro cambia según se gira el dial. Si suponemos que puede dar una vuelta completa, asignamos 360, que serían los grados que gira. Si gira ¾ partes, asignamos 270, y así sucesivamente.

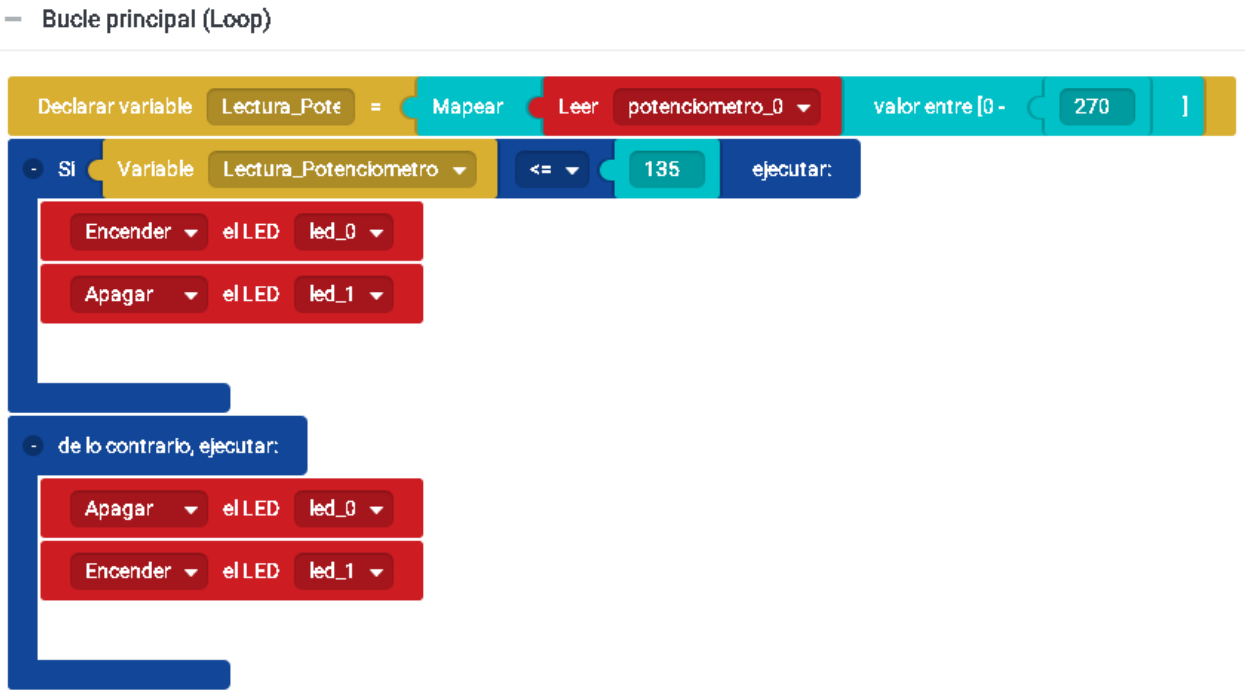
5.- CONTROL DE DOS LED CON UN POTENCIÓMETRO

Encender un LED cuando el potenciómetro esté a menos de la mitad de su recorrido y encender otro cuando supere la mitad del recorrido.

Hazlo por tu cuenta e intenta corregir los fallos si no te funciona.

Cuando te funcione intenta encontrar la mitad del recorrido del potenciómetro. Sucede en el tránsito del encendido de un LED al otro.

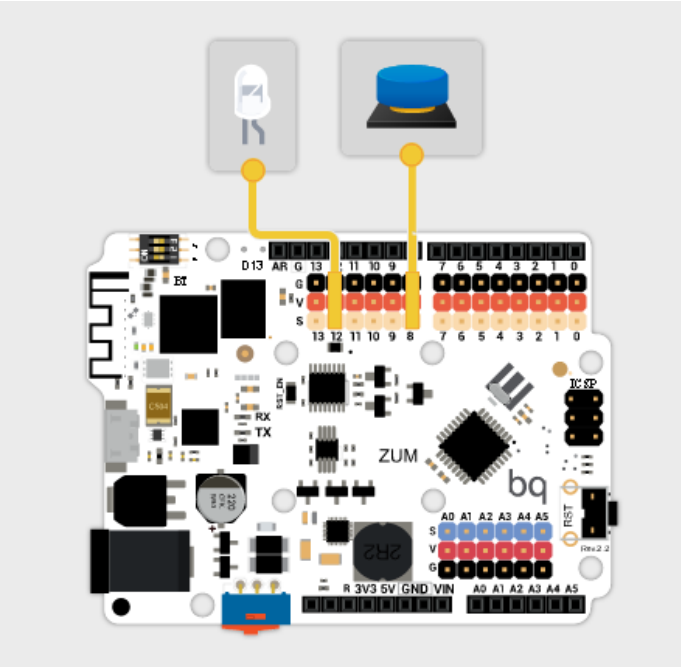


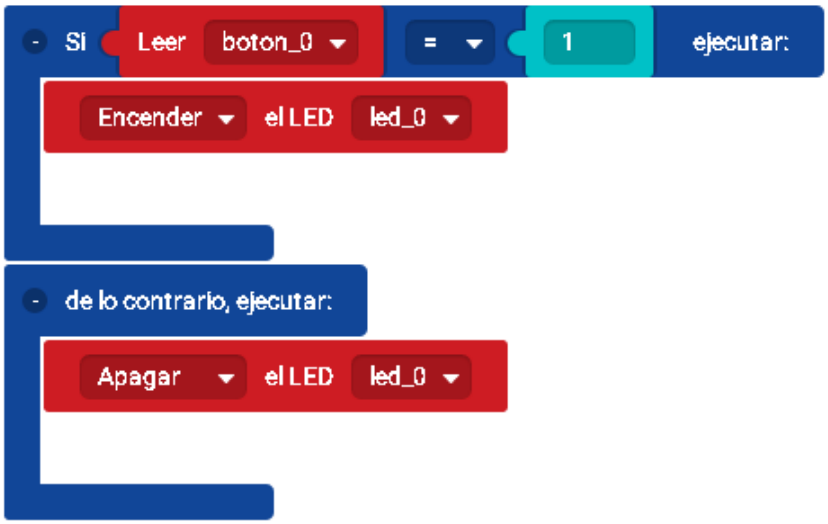


06.- PULSADOR Y LED

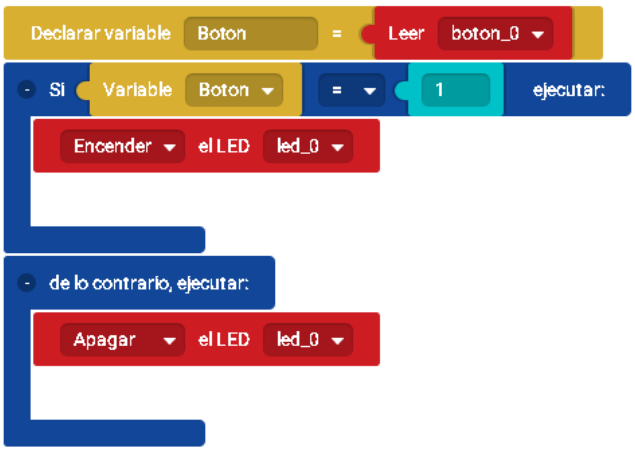
Un pulsador es un componente digital (0 o 1, activado o desactivado), Por lo tanto su conexión es a un pin digital.

Haz que cuando se pulse el botón, se encienda el LED.





Prueba a ver si funcionaría declarando una variable para el botón.



07.- PULSADOR Y DOS LED. LUZ DE FRENO.

Imagina que vas en un coche con las luces encendidas. Por tanto, un LED debe lucir siempre.

Por otra parte tenemos un freno, que será el botón.

Diseña el programa para que un LED luzca siempre y el otro “al pisar el freno”.

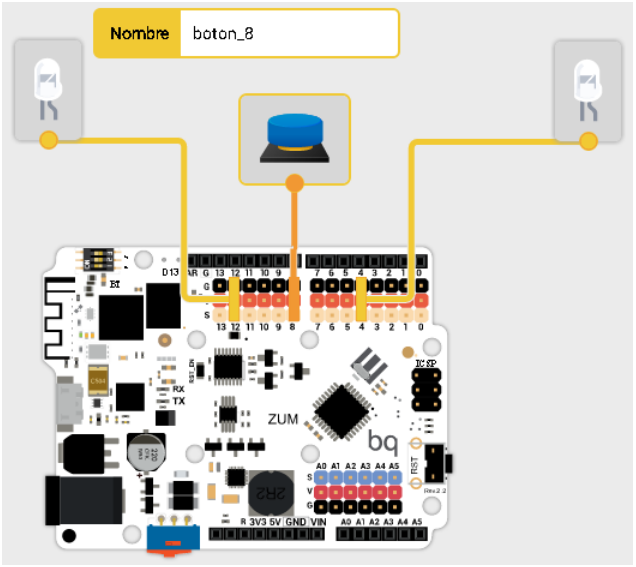
En este ejercicio cambia el nombre de los tres componentes para que coincidan con el número de pin y así lo tenemos a la vista cuando programemos. Se trata de una simple ayuda.

Por ejemplo:

Luz_12: en el pin 12

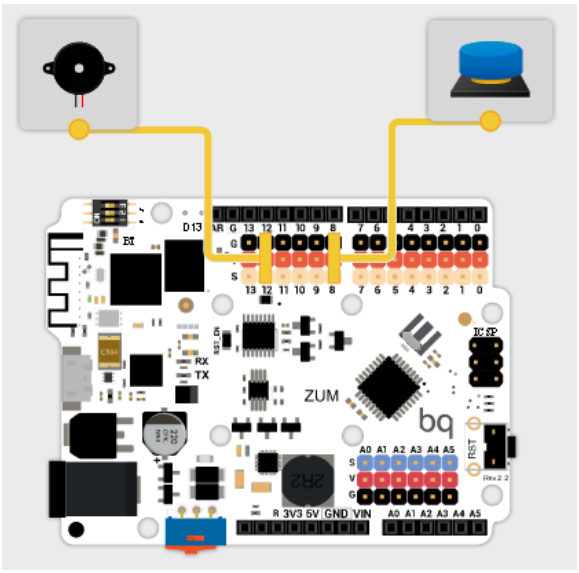
Botón_8: en el pin 8

Luz_4: en el pin 4



08.- HACER SONAR UN ZUMBADOR CON UN BOTÓN.

Haz que cuando pulse el botón, el zumbador suene en la nota Do durante dos segundos.



09.- ESCALA DE NOTAS

Haz sonar la escala de notas en un zumbador al pulsar un botón.

10.- MARCHA IMPERIAL

Bucle principal (Loop)

Sonar el zumbador	Variable (componentes)	zumbador_0	con la nota	880	durante	500 ms
Sonar el zumbador	Variable (componentes)	zumbador_0	con la nota	440	durante	350 ms
Sonar el zumbador	Variable (componentes)	zumbador_0	con la nota	440	durante	150 ms
Sonar el zumbador	Variable (componentes)	zumbador_0	con la nota	880	durante	500 ms
Sonar el zumbador	Variable (componentes)	zumbador_0	con la nota	830	durante	250 ms
Sonar el zumbador	Variable (componentes)	zumbador_0	con la nota	784	durante	250 ms
Sonar el zumbador	Variable (componentes)	zumbador_0	con la nota	740	durante	125 ms
Sonar el zumbador	Variable (componentes)	zumbador_0	con la nota	698	durante	125 ms
Sonar el zumbador	Variable (componentes)	zumbador_0	con la nota	740	durante	250 ms

Esperar 250 ms

Sonar el zumbador	Variable (componentes)	zumbador_0	con la nota	455	durante	250 ms
Sonar el zumbador	Variable (componentes)	zumbador_0	con la nota	622	durante	500 ms
Sonar el zumbador	Variable (componentes)	zumbador_0	con la nota	587	durante	250 ms
Sonar el zumbador	Variable (componentes)	zumbador_0	con la nota	554	durante	250 ms
Sonar el zumbador	Variable (componentes)	zumbador_0	con la nota	523	durante	125 ms
Sonar el zumbador	Variable (componentes)	zumbador_0	con la nota	466	durante	125 ms
Sonar el zumbador	Variable (componentes)	zumbador_0	con la nota	523	durante	250 ms

Esperar 250 ms

Sonar el zumbador	Variable (componentes)	zumbador_0	con la nota	125	durante	125 ms
Sonar el zumbador	Variable (componentes)	zumbador_0	con la nota	415	durante	500 ms
Sonar el zumbador	Variable (componentes)	zumbador_0	con la nota	349	durante	375 ms
Sonar el zumbador	Variable (componentes)	zumbador_0	con la nota	523	durante	125 ms
Sonar el zumbador	Variable (componentes)	zumbador_0	con la nota	659	durante	1000 ms
Sonar el zumbador	Variable (componentes)	zumbador_0	con la nota	880	durante	500 ms
Sonar el zumbador	Variable (componentes)	zumbador_0	con la nota	440	durante	350 ms
Sonar el zumbador	Variable (componentes)	zumbador_0	con la nota	440	durante	150 ms
Sonar el zumbador	Variable (componentes)	zumbador_0	con la nota	880	durante	500 ms
Sonar el zumbador	Variable (componentes)	zumbador_0	con la nota	830	durante	250 ms
Sonar el zumbador	Variable (componentes)	zumbador_0	con la nota	784	durante	250 ms
Sonar el zumbador	Variable (componentes)	zumbador_0	con la nota	740	durante	125 ms
Sonar el zumbador	Variable (componentes)	zumbador_0	con la nota	698	durante	125 ms
Sonar el zumbador	Variable (componentes)	zumbador_0	con la nota	740	durante	250 ms

Esperar 250 ms

Seguro que todos habréis reconocido este pasaje de la BSO de Star Wars compuesto por John Williams. Esta marcha siempre sonaba cuando aparecía Darth Vader.

En este ejercicio se fijan los tiempos indicándolos en milisegundos para cada nota.

Habréis observado que en algún caso se han introducido *delays* de 10 ms, prácticamente imperceptible al oído. Se han introducido porque, sin ellos algunas notas sonaban mal. Puedes probar tú a no ponerlos, a lo mejor a ti no te son necesarios.

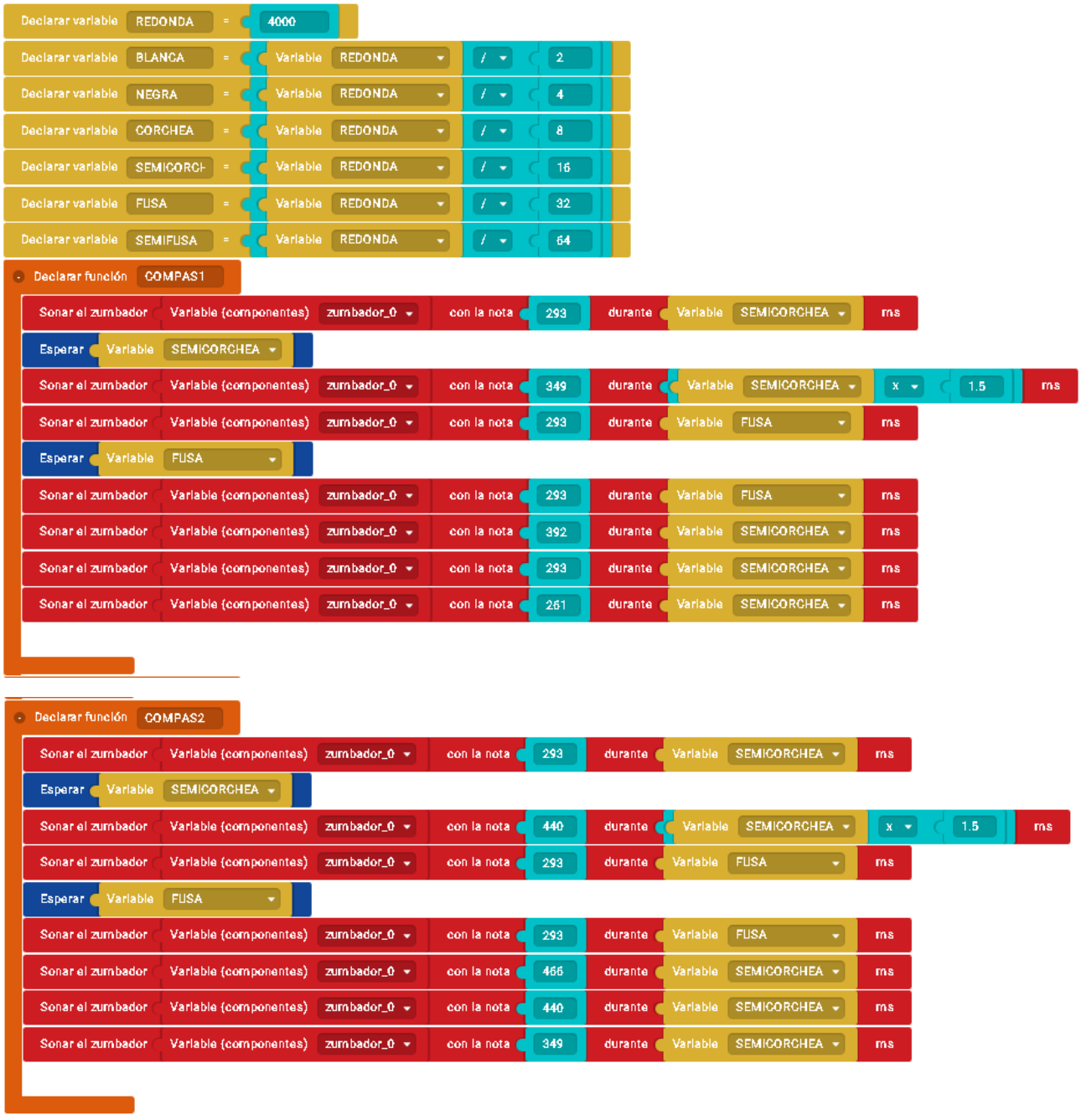
Otra forma más sofisticada de recrear música con Bitbloq es definir como **variables globales** los tipos de notas. Establecemos la duración de una redonda como cuatro segundos y, las demás, son fracciones de esta: blanca, la mitad; negra la cuarta parte; y, corchea, la octava parte… Después a cada nota se le asigna su variable ajustándose a la partitura.

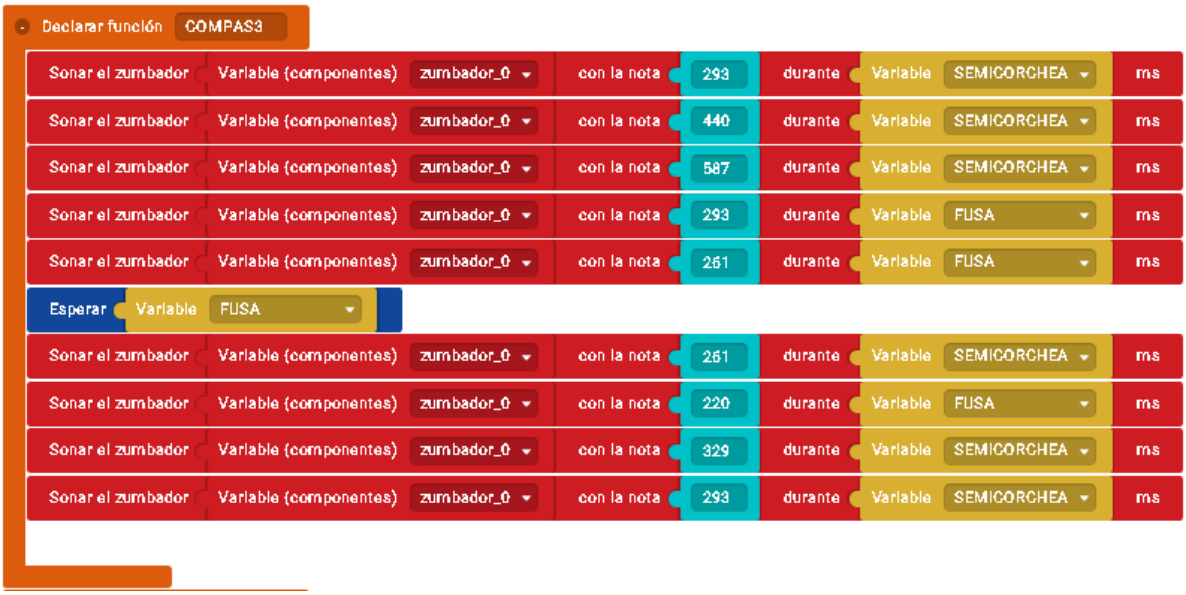
Por otra parte, definimos los compases como **funciones**. Una vez hecho esto, organizamos la ejecución de las funciones (los compases) en la secuencia que deseemos.

El próximo ejercicio se realiza de esta manera y, con él, conseguiréis hacer sonar la melodía principal de la banda sonora de la famosa película de 1984 “Beberly Hills Cop”, compuesta por Harold Faltermeyer y Danny Elfman.

11.- SUPER-DETECTIVE EN HOLLYWOOD

– Variables globales, funciones y clases





— Instrucciones iniciales (Setup)

Arrastra un bloque aquí para empezar tu programa

— Bucle principal (Loop)

12.- SENSOR DE ULTRASONIDOS

¿Qué es un sensor de ultrasonidos?

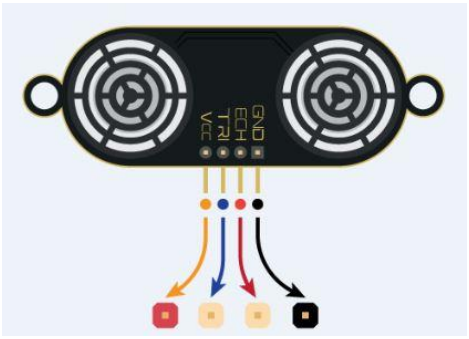
Un sensor de ultrasonidos es un componente que utiliza ondas de alta frecuencia para saber la distancia a un objeto. Este tipo de sensores tienen dos partes, una es el **emisor** que emite la señal y la otra el **receptor** que recibe la señal si ésta rebota sobre algún obstáculo cercano.

Este sistema es el mismo que utilizan los murciélagos para orientarse y se aplica de forma similar en robótica para detectar obstáculos y medir distancias.

Vaya, cuántos cables... ¿Cómo se conecta el sensor de ultrasonidos?

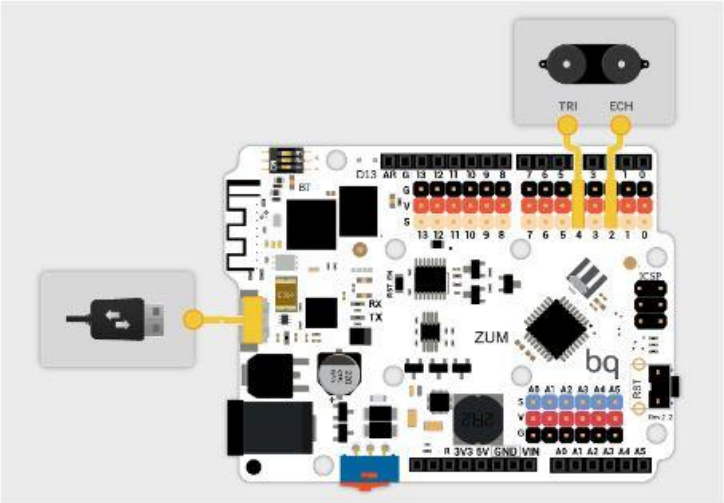
El sensor de ultrasonidos tiene cuatro pines marcados como: **GND**, **ECH**, **TRI**, **VCC**.

El **TRI** (Trigger) es el encargado de ordenar al sensor que emita la onda y el **ECH** (Echo) la recibe. En función del tiempo que transcurre entre la emisión y recepción de la señal, nuestra placa calcula la distancia. Estos dos cables deben ir conectados a un pin de señal, que en tu placa ZUM son de color blanco. Los otros dos cables, **GND** y **VCC**, se conectan a cualquier pin negro y rojo respectivamente.

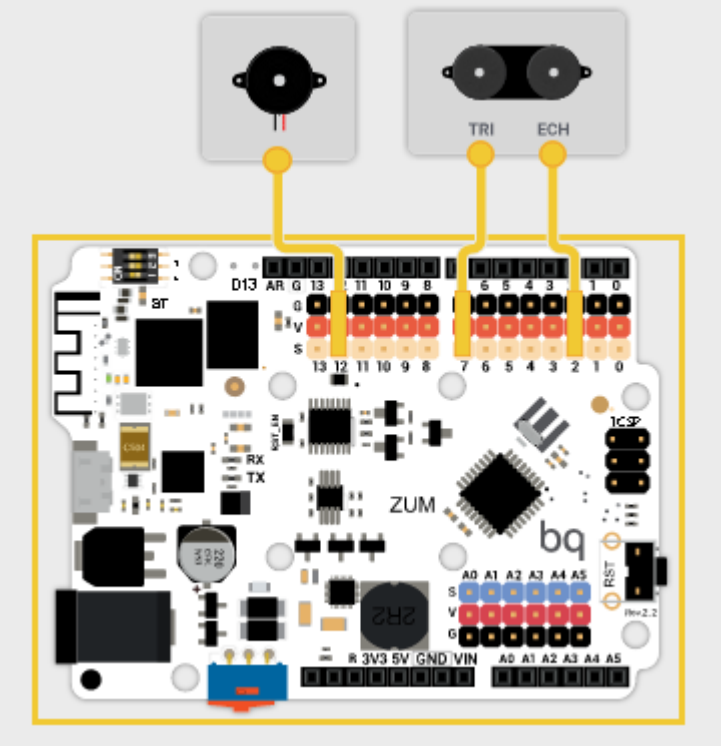


Comprobemos cómo funciona

Podemos visualizar la distancia a un objeto utilizando el puerto serie. El sensor de ultrasonidos siempre nos devolverá la distancia en centímetros.



13.- FRENO DE APARCAMIENTO CON ULTRASONIDOS



Bucle principal (Loop)

14.- CALIBRADO DE UN MOTOR DE ROTACIÓN CONTÍNUA

Algunos motores o servomotores disponen de un potenciómetro incorporado para calibrar bien su funcionamiento. En el lado opuesto a la entrada del cable se puede ver una ranura en la que, introduciendo un destornillador, calibra el motor.

IMPORTANTE: esta es una operación delicada y se puede estropear el motor sin no se hace adecuadamente. Debe girarse despacio y, cuando se llega a un extremo, no se puede forzar.

La cuestión consiste en que el motor responda a la programación que se detalla en las fotos de abajo. En los cinco segundos de espera se gira levemente el potenciómetro y se observa qué sucede cuando vuelva a ponerse en marcha. Si todo funciona correctamente, no debe tocarse nada. Una vez finalizada esta parte, se programa lo mismo cambiando el sentido de giro a anti-horario y repetir la operación.