



PC Factory
Área de profesionales y especialistas
Equipo de computación y electrónica

Manual de usuario

Aprendiendo con Boson

Proyectos de entrada basados en Micro:bit



Índice

1	Introducción	2
1.1	Plug Play	2
1.2	Agarre múltiple.....	2
1.3	Comienza sin código	3
1.4	Es programable.....	3
1.5	Integración con Micro:bit	3
2	Alarma manual	4
2.1	Requerimientos	4
2.2	Diagrama de conexiones.....	4
2.3	Programa en bloques	5
3	Puerta automática	6
3.1	Requerimientos	6
3.2	Diagrama de conexiones.....	6
3.3	Programa en bloques	7
4	Reproductor de música activado por sonido.....	7
4.1	Requerimientos	7
4.2	Diagrama de conexiones.....	8
4.3	Programa en bloques	8
5	Instrumento musical por proximidad	9
5.1	Requerimientos	9
5.2	Diagrama de conexiones.....	9
5.3	Programa en bloques	10
6	Luces sacudidas	10
6.1	Requerimientos	11
6.2	Diagrama de conexiones.....	11
6.3	Programa en bloques	11
7	Coche seguidor de línea	12
7.1	Requerimientos	12
7.2	Diagrama de conexiones.....	13
7.3	Programa en bloques	13

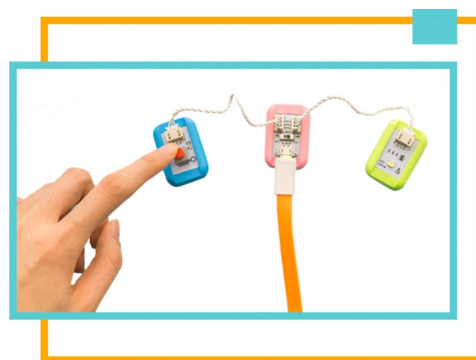
1. Introducción

Boson es una plataforma de módulos electrónicos pensados en el aprendizaje y enseñanza de los más pequeños del hogar. Dentro de su diseño se destacan elementos como un código de colores para diferenciarlos según su función, flexibilidad de armado y compatibilidad con LEGO.

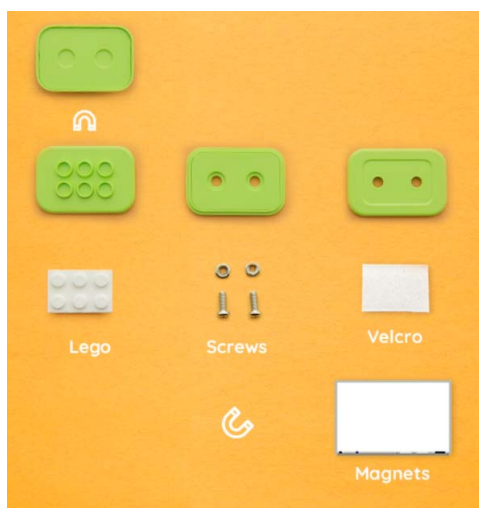
En esta guía encontrarás diferentes mini-proyectos que puedes armar para comenzar tu aprendizaje con los módulos Boson. Antes de esto hemos de familiarizarnos con el funcionamiento de estos módulos y su uso.

1.1. Plug Play

Todos los módulos Boson tienen conectores en común, que hace que poder utilizar los módulos sea tan sencillo como llegar y conectar un cable entre dos módulos diferentes. De esta manera no se requiere soldar y se asegura una estabilidad en la señal tanto de datos como de poder, además de evitar problemas por conexiones en inversa.



1.2. Agarre múltiple

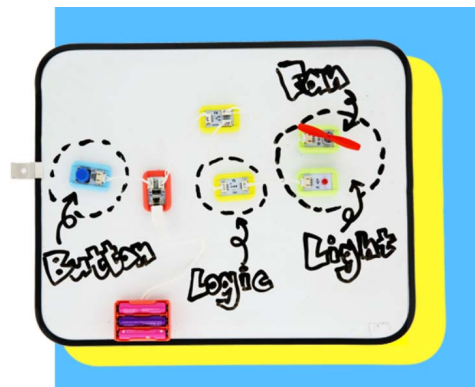
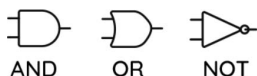


Los módulos Boson incluyen imanes que te permitirán pegarlos donde sea: una pizarra, en el refrigerador o cualquier otra superficie en donde podrías pegar un imán (¡como otro módulo Boson!). Además, cada módulo viene con bases compatibles con bloques LEGO y tornillos para afirmarlos.

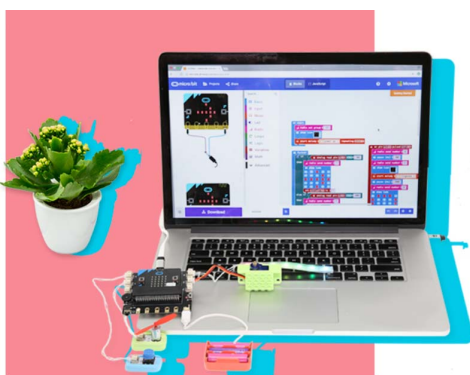


1.3. Comienza sin código

Dentro de la plataforma Boson hay módulos que representan compuertas lógicas: AND, OR, NOT. Estos te permitirán representar programas de manera física y conseguir un entendimiento instantáneo del flujo lógico que tengan tus invenciones.



1.4. Es programable



Puedes llevar las plataformas Boson a un nivel superior y ocuparlas con plataformas como **micro:bit** y Arduino, con estas tendrás acceso a programar los módulos via Scratch, MakeCode, JavaScript, Python y C.

 micro:bit  Arduino

1.5. Integración con Micro:bit

Puedes aprovechar todas las virtudes de micro:bit a través de la placa de expansión de Boson (ver Figura 1). Con ella podrás conectar cualquier módulo que desees y verás como se comportarán como si siempre hubieran sido uno.

En las próximas secciones se utilizarán nombres en inglés de algunas componentes para evitar ambigüedades

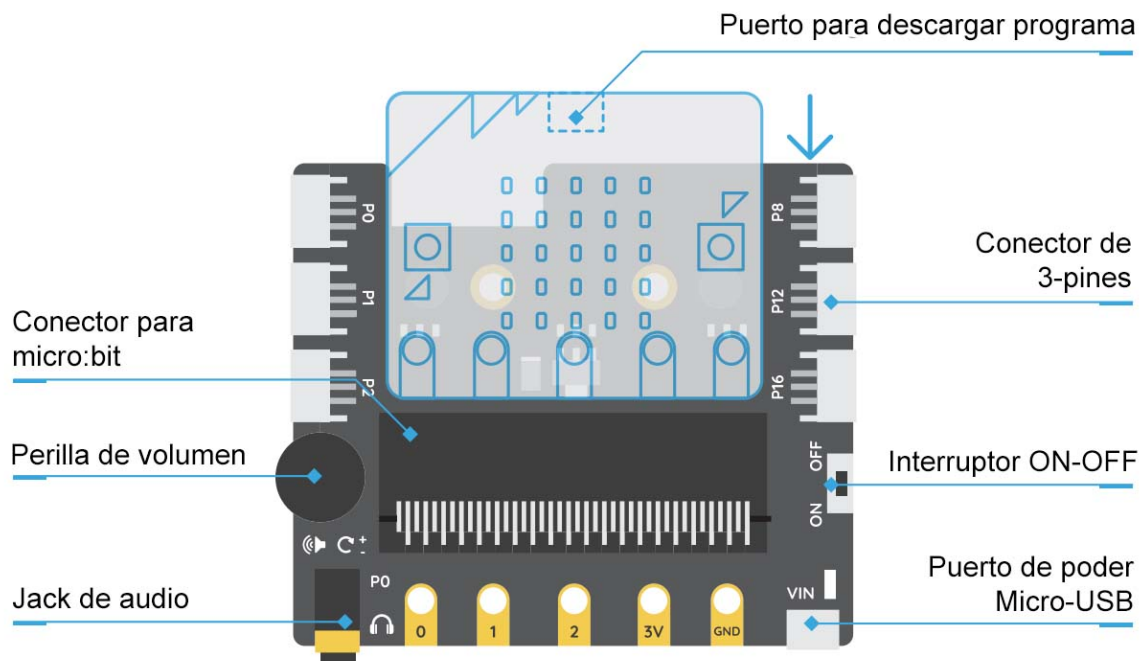


Figura 1: Placa de expansión micro:bit para Boson

2. Alarma manual

Este mini proyecto consiste en armar una alarma que sonará 10 veces una vez sea presionado un botón.

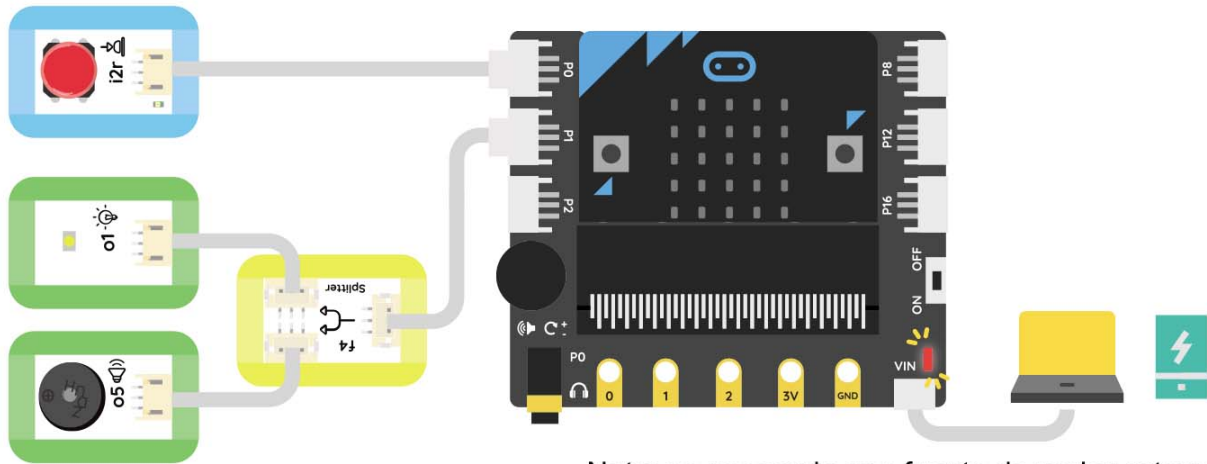
2.1. Requerimientos

Cantidad	Componentes
1	micro:bit
1	Módulo buzzer (o5)
1	Módulo LED (o1)
1	Módulo botón (i2r)
1	Módulo splitter (f4)
4	Cables de 3-pines
1	Cable micro-USB

2.2. Diagrama de conexiones

- P0: Botón
- P1: Splitter

- Splitter0: LED
- Splitter1: Buzzer



Nota: es necesaria una fuente de poder externa

Figura 2: Diagrama de conexiones

2.3. Programa en bloques

Una forma sencilla de programar estos módulos es a través del editor oficial de micro:bit, [MakeCode](#). Ya sea ocupes este o alguna otra interfaz de programación por bloques, basta con que copies una secuencia como la presentada en la Figura 3 y tendrás tu alarma funcionando.

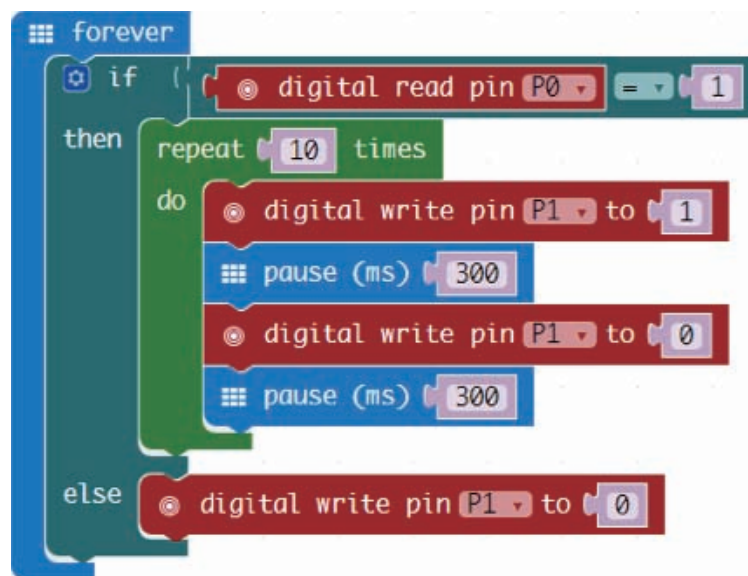


Figura 3: Programa en bloques

Este algoritmo tiene consiste en estar revisando constantemente el puerto digital P0 y si este tiene un valor de 1 (se presionó el botón), se ejecuta el enviar un 1 a P1. El splitter se encarga de enviar la señal del micro:bit al módulo led y al buzzer al mismo tiempo, por lo que estos actuarán una vez haya cambios en el puerto P1.

3. Puerta automática

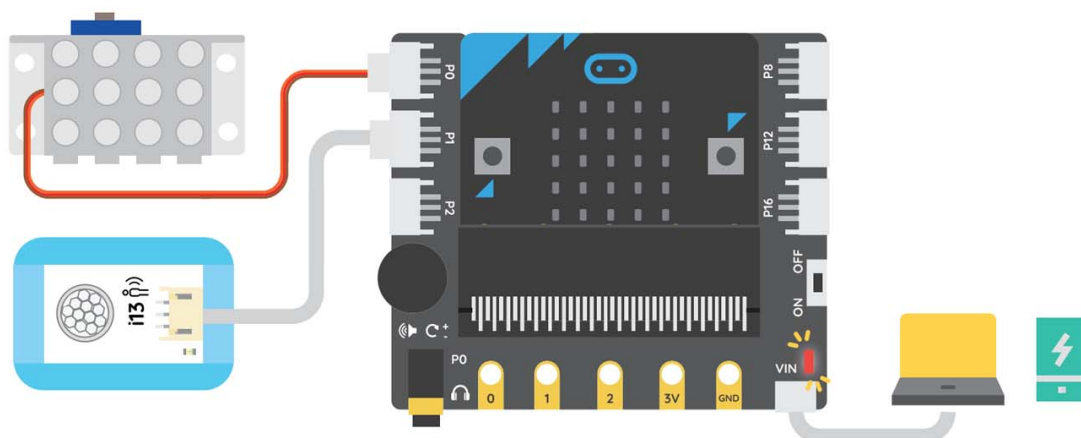
Este mini proyecto te permitirá tener la base de una puerta que se manera de abra de manera automática cuando te acerques a ella. Para hacerla funcionar basta con acercar tu mano al sensor de movimiento y el servomotor comenzará a girar simulando ser la cerradura de la puerta.

3.1. Requerimientos

Cantidad	Componentes
1	micro:bit
1	Módulo servomotor
1	Módulo sensor de movimiento
1	Cables de 3-pines
1	Cable micro-USB

3.2. Diagrama de conexiones

- P0: Servomotor
- P1: Sensor de movimiento



Nota: es necesaria una fuente de poder externa

Figura 4: Diagrama de conexiones

3.3. Programa en bloques

Una forma sencilla de programar estos módulos es a través del editor oficial de micro:bit, [MakeCode](#). Ya sea ocupes este o alguna otra interfaz de programación por bloques, basta con que copies una secuencia como la presentada en la Figura 5 y tendrás tu puerta funcionando.

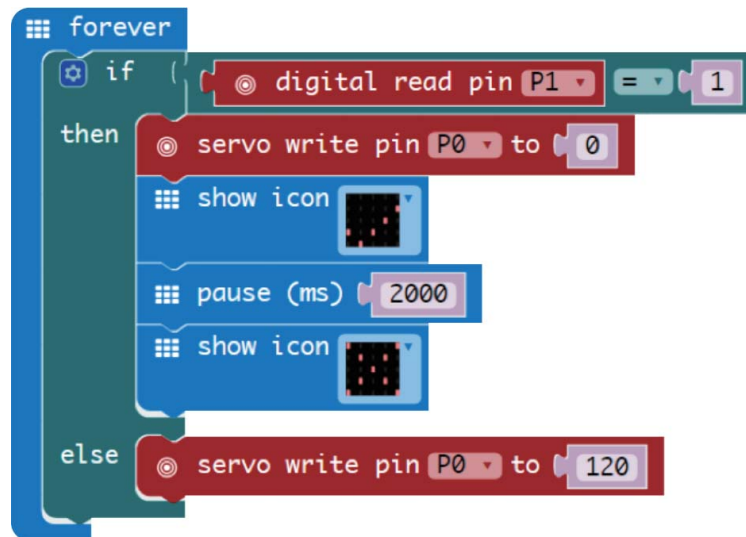


Figura 5: Programa en bloques

En este programa, se revisa constantemente el valor del puerto P1 y si este vale 1 (alguien o algo se cruzó frente del sensor), se envía una señal al servomotor para que este comience a girar hasta su posición de apertura, en caso contrario se devuelve el servomotor a la posición inicial.

4. Reproductor de música activado por sonido

Este mini proyecto hará que puedas reproducir tu música con un simple aplauso, grito o cualquier ruido lo suficientemente fuerte.

4.1. Requerimientos

Cantidad	Componentes
1	micro:bit
1	Módulo micrófono
1	Audífonos
1	Cables de 3-pines
1	Cable micro-USB

4.2. Diagrama de conexiones

- P1: Micrófono

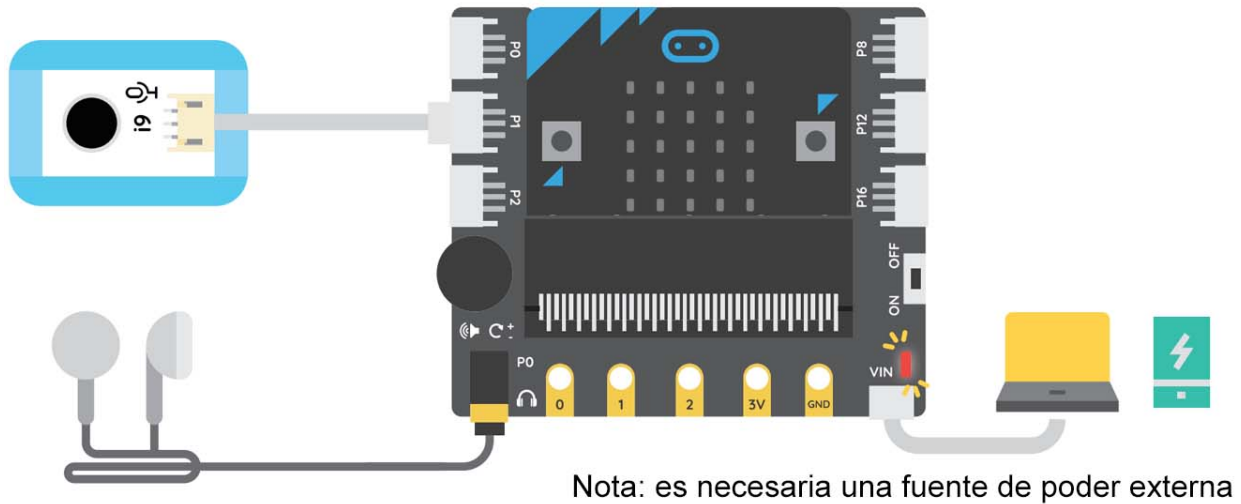


Figura 6: Diagrama de conexiones

4.3. Programa en bloques

Una forma sencilla de programar estos módulos es a través del editor oficial de micro:bit, [MakeCode](#). Ya sea ocupes este o alguna otra interfaz de programación por bloques, basta con que copies una secuencia como la presentada en la Figura 7 y tendrás tu reproductor funcionando.



Figura 7: Programa en bloques

El funcionamiento de este programa se basa en revisar de manera constante el estado del puerto P1 y una vez ese supere un determinado umbral, cuando el sonido sea lo suficientemente fuerte, se comenzará a reproducir la melodía seleccionada (ringtone). El parámetro destacado en la Figura 7 de valor 200 corresponde al umbral de activación, cambiar su valor alterará que tan fácil será hacer que tu reproductor comience a sonar.

5. Instrumento musical por proximidad

Este mini proyecto hará que puedas reproducir ciertas notas musicales dependiendo de que tan cerca te encuentras de tu Boson. ¡Con esto podrás ocupar el kit como si fuera un instrumento musical que puedes tocar a la distancia!

5.1. Requerimientos

Cantidad	Componentes
1	micro:bit
1	Módulo sensor ultrasónico
1	Audífonos
1	Cables de 3-pines
1	Cable micro-USB

5.2. Diagrama de conexiones

- P1: Sensor ultrasónico

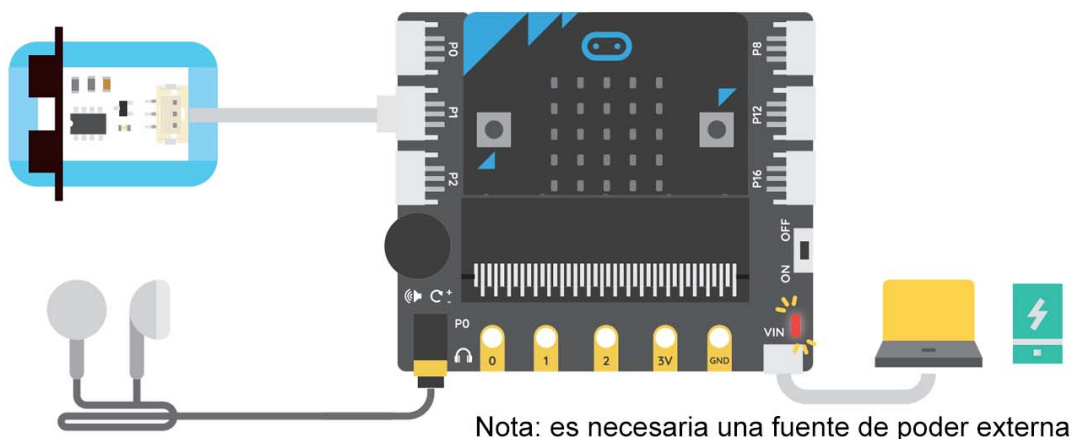


Figura 8: Diagrama de conexiones

5.3. Programa en bloques

Una forma sencilla de programar estos módulos es a través del editor oficial de micro:bit, [MakeCode](#). Ya sea ocupes este o alguna otra interfaz de programación por bloques, basta con que copies una secuencia como la presentada en la Figura 9 y tendrás tu instrumento funcionando.

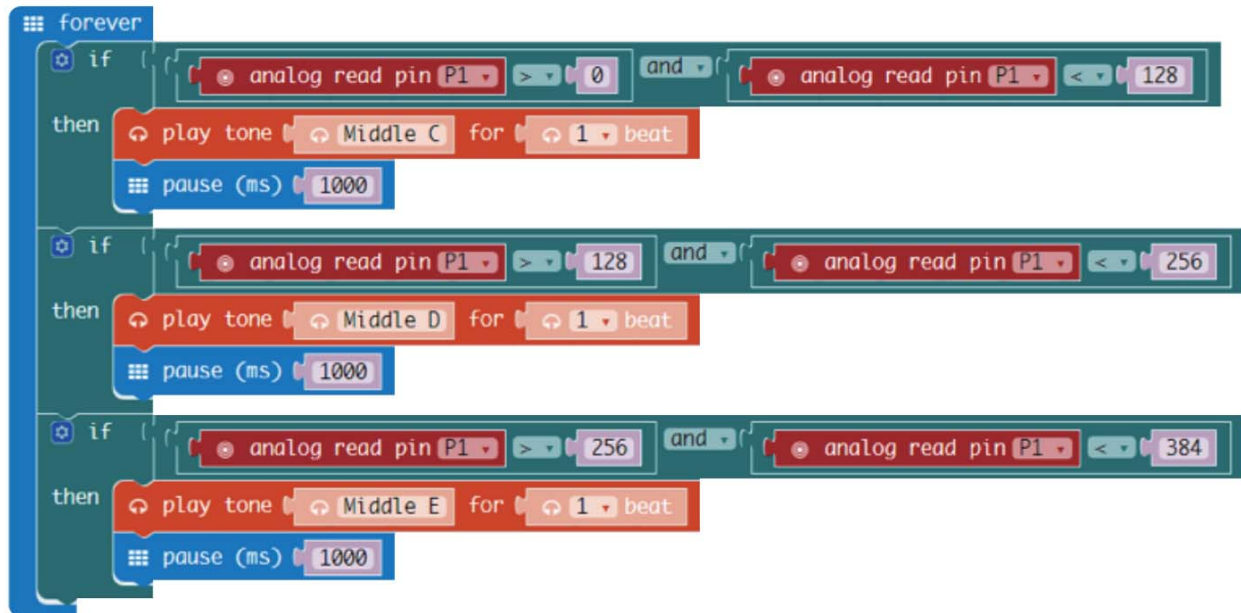


Figura 9: Programa en bloques

Este programa base viene configurado para que el valor del puerto P1 se revise constantemente, y dependiendo de su valor se tocará una nota musical diferente. El valor del puerto P1 está a ligado a la cercanía del objeto más cercano al ultrasónico. En este ejemplo, se ocupan 3 rangos diferentes para determinar cuando tocar cada nota musical, en específico:

- $0 < P1 < 128$: Nota musical C
- $128 < P1 < 256$: Nota musical D
- $256 < P1 < 384$: Nota musical E

6. Luces sacudidas

Este mini proyecto consiste en configurar tu micro:bit para que ilumine un conjunto de luces LED cuando lo sacudas. Además, aprenderás cual es el formato para ocupar una función creada por ti en el programa por bloques.

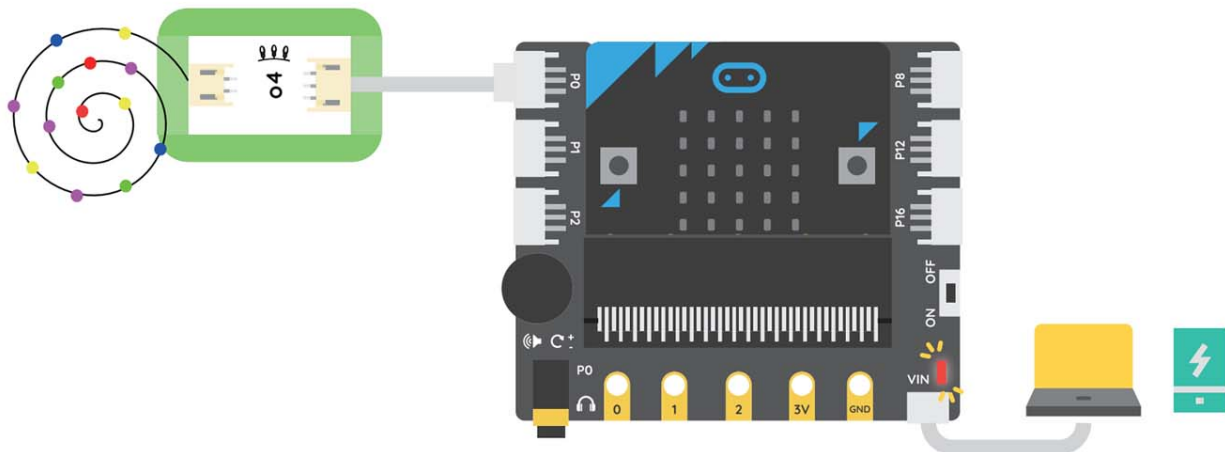
6.1. Requerimientos

Cantidad	Componentes
1	micro:bit
1	Módulo tira de luces multicolor
1	Audífonos
1	Cables de 3-pines
1	Cable micro-USB

6.2. Diagrama de conexiones

- P0: Módulo tira de luces multicolor
 - Módulo0: Conjunto de luces LED

Los LEDs, en general, deben ir conectados con resistencias en serie para protegerlos de sobre-corriente y evitar que estos se sobrecarguen. Puedes conectar más LEDs en paralelo para tener toda una tira iluminada, con esto debes tener cuidado se no superar el consumo de corriente permitido por el micro:bit. El micro:bit tiene un límite máximo de 90mA a distribuir entre todos sus puertos, siempre y cuando la fuente de poder a la que está conectado se lo permita.



Nota: es necesaria una fuente de poder externa

Figura 10: Diagrama de conexiones

6.3. Programa en bloques

Una forma sencilla de programar estos módulos es a través del editor oficial de micro:bit, [MakeCode](#). Ya sea ocupes este o alguna otra interfaz de programación por bloques, basta con que

copies una secuencia como la presentada en la Figura 11 y tendrás tu instrumento funcionando.



Figura 11: Programa en bloques

El bloque **púrpura** inicial del programa corresponde a una interrupción o evento, este funciona de tal manera que cuando ocurra la acción seleccionada **on shake** (sacudida) se ejecutarán los bloques internos. En este caso, dentro del evento se ejecuta el llamar a una función propia **Breathing Light** definida de manera independiente. La función **Breathing Light** configura el puerto analógico **P0** para que realice un barrido del brillo de las luces LED, partiendo por todas apagadas hasta prenderlas completamente y luego devuelta.

7. Coche seguidor de línea

Este mini proyecto consiste en configurar tu micro:bit para que controle un automóvil de 2 ruedas de tracción para seguir una línea.

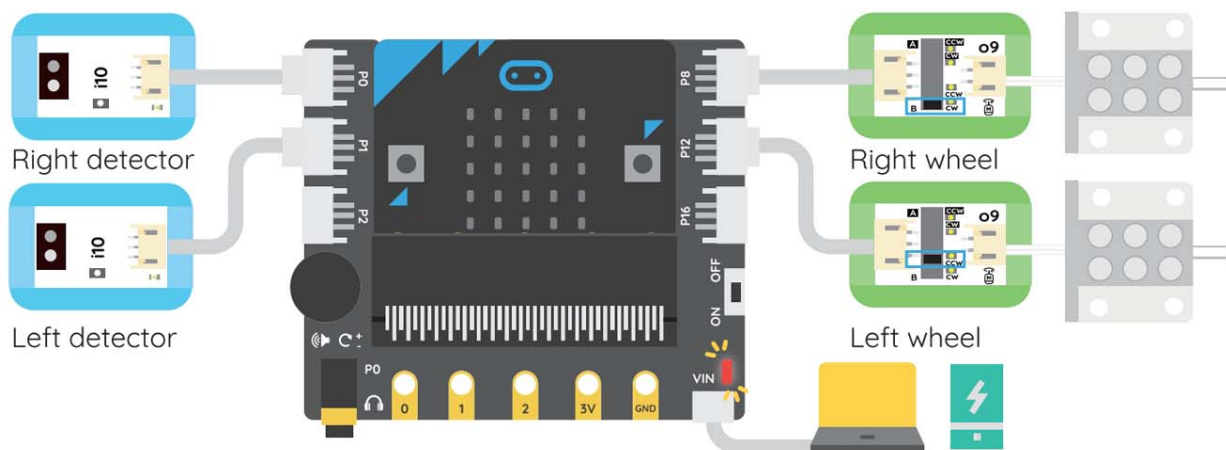
7.1. Requerimientos

Cantidad	Componentes
1	micro:bit
2	Módulo detector de línea
2	Módulo controlador de motores
4	Cables de 3-pines
1	Cable micro-USB

7.2. Diagrama de conexiones

- P0: Módulo detector de línea (i10)
- P1: Módulo detector de línea (i10)
- P8: Módulo controlador de motores (o9)
 - Controlador0: Motor DC
- P12: Módulo controlador de motores (o9)
 - Controlador0: Motor DC

Los LEDs, en general, deben ir conectados con resistencias en serie para protegerlos de sobrecorriente y evitar que estos se sobrecarguen. Puedes conectar más LEDs en paralelo para tener toda una tira iluminada, con esto debes tener cuidado se no superar el consumo de corriente permitido por el micro:bit. El micro:bit tiene un límite máximo de 90mA a distribuir entre todos sus puertos, siempre y cuando la fuente de poder a la que está conectado se lo permita.



Nota: es necesaria una fuente de poder externa

Figura 12: Diagrama de conexiones

7.3. Programa en bloques

Una forma sencilla de programar estos módulos es a través del editor oficial de micro:bit, [MakeCode](#). Ya sea ocupes este o alguna otra interfaz de programación por bloques, basta con que copies una secuencia como la presentada en la Figura 13 y tendrás el cerebro de tu coche funcionando.

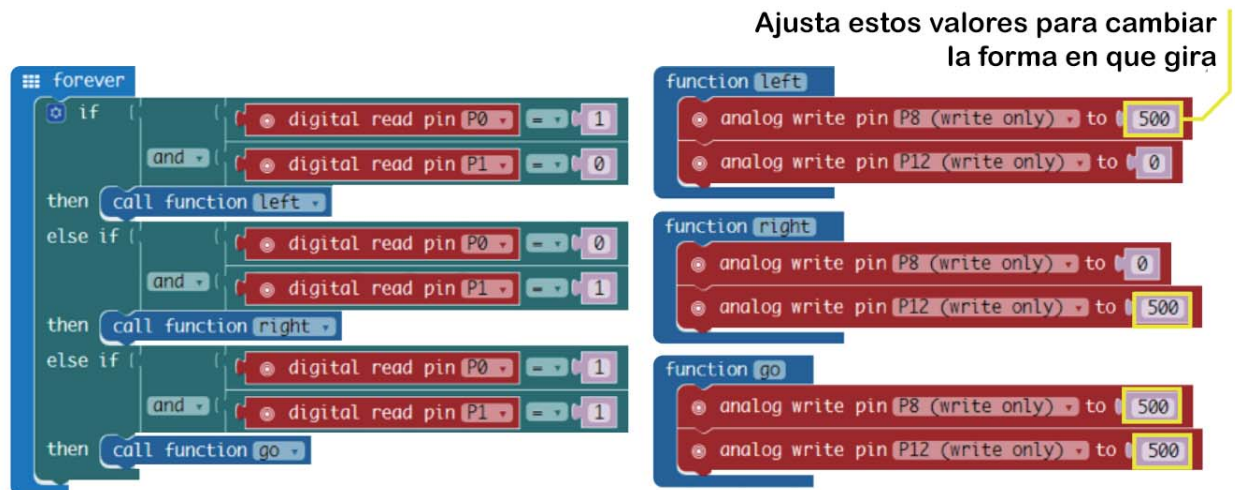


Figura 13: Programa en bloques

El programa presentado revisa de manera constante las mediciones provenientes de los sensores de línea, de acuerdo a cuáles de ellos se encuentren en el camino se ajusta el giro de los motores para que el coche siga en su camino. En este ejemplo, para lograr cada giro del auto se detiene una rueda mientras se deja avanzar la otra.