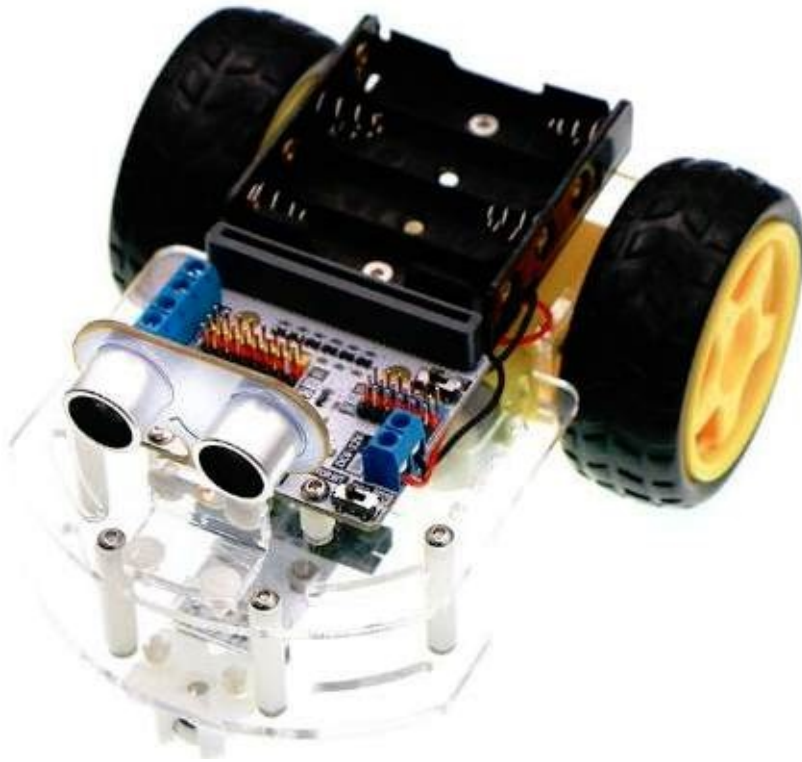

Tabla de contenido

| | |
|-----------------------------|-------|
| 1 Introduccion | 1.1 |
| 1.1 Micro:BIT | 1.1.1 |
| 1.2 Kit de préstamo CATEDU | 1.1.2 |
| 1.3 Montaje | 1.1.3 |
| 1.4 Conexión con Makecode | 1.1.4 |
| Reto 1 Música | 1.1.5 |
| 2 Sensor distancia | 1.2 |
| Reto 2 Me quedo a 5cm | 1.2.1 |
| Reto 3 Evitar obstáculos | 1.2.2 |
| 3 Sigue-lineas | 1.3 |
| Reto 4 Seguir línea | 1.3.1 |
| 4 Si tuviera 2 micro:BIT | 1.4 |
| Reto 5 Mando a distancia | 1.4.1 |
| 5 Muro micro:BIT | 1.5 |
| 6 Muro microCar | 1.6 |
| 7 Grupo robótica | 1.7 |
| 8 Pensamiento computacional | 1.8 |
| 9 Créditos | 1.9 |

Introduccion SmartCar



Micro:BIT se queda un poco triste si no se "mueve", pero hay que tener en cuenta que en la robótica el movimiento se paga:

- Adición de motores, chasis, ruedas...
- Necesitas una electrónica añadida de potencia y control de los motores.
- Necesitas unos sensores de lo contrario el movimiento se queda insípido sin reacción al mundo exterior.

La placa de micro:BIT no incorpora movimiento, por eso cuesta tan poco, pero si quieres este extra tienes que pagar ([unos 40€ casi el doble de lo que cuesta micro:BIT, ver este enlace](#)). Aún así el conjunto sale más barato que [mBot](#).

Otra opción de dotar ruedas a nuestro microbit es usar un mbot, sólo comprando una placa, [aquí tienes cómo hacerlo](#)

Más opciones: Están saliendo kits de ruedas más baratos, [este cuesta 25€-30€](#)

Este kit viene con sensor de ultrasonidos, sensor de sigue-lineas y la placa **motorbit** que se inserta la micro:BIT además de tener altavóz y salidas a puertos por si se quiere añadir servos por ejemplo. Para más información de esta placa consultar [esta web](#).



Micro:BIT

Antes de empezar con este tutorial, tienes que haber comprendido la placa micro:BIT

<https://catedu.gitbooks.io/micro-bit/content/>



1.2 Kit de préstamo CATEDU

Si te matriculas en nuestros cursos, tienes derecho a un kit de préstamo ([ver condiciones](#)) Pero para este curso sería conveniente conseguir **otro micro:BIT** para hacer el reto **MANDO A DISTANCIA** (puedes matricularte tú en el curso micro:BIT o intentar hacer el curso con otro compañero)

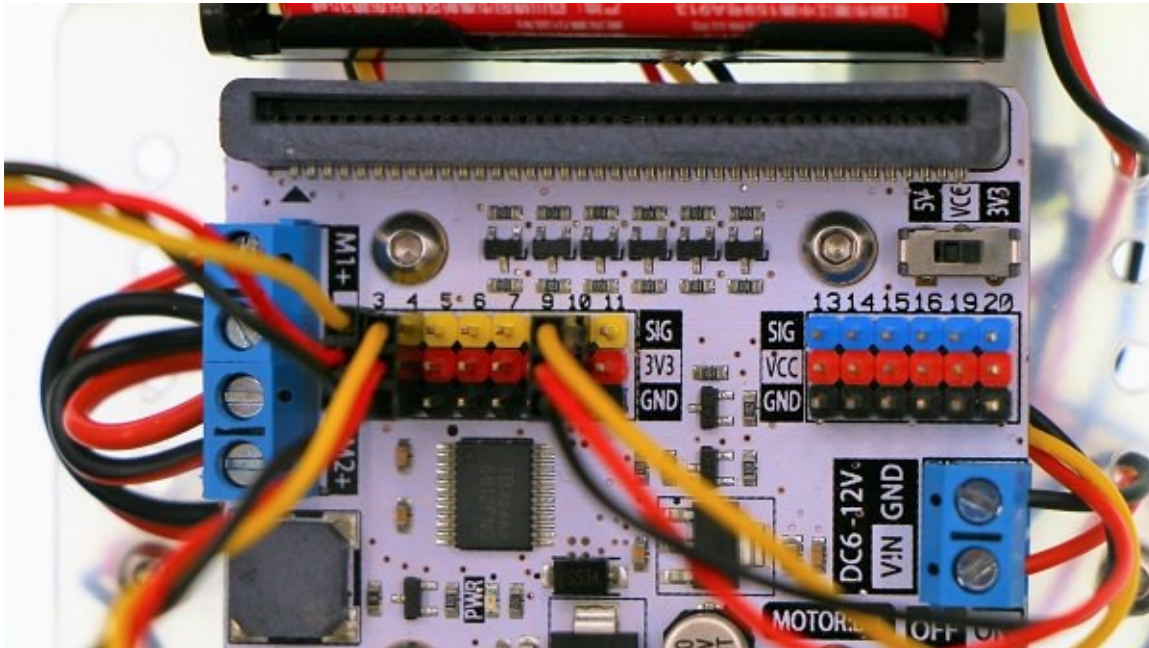


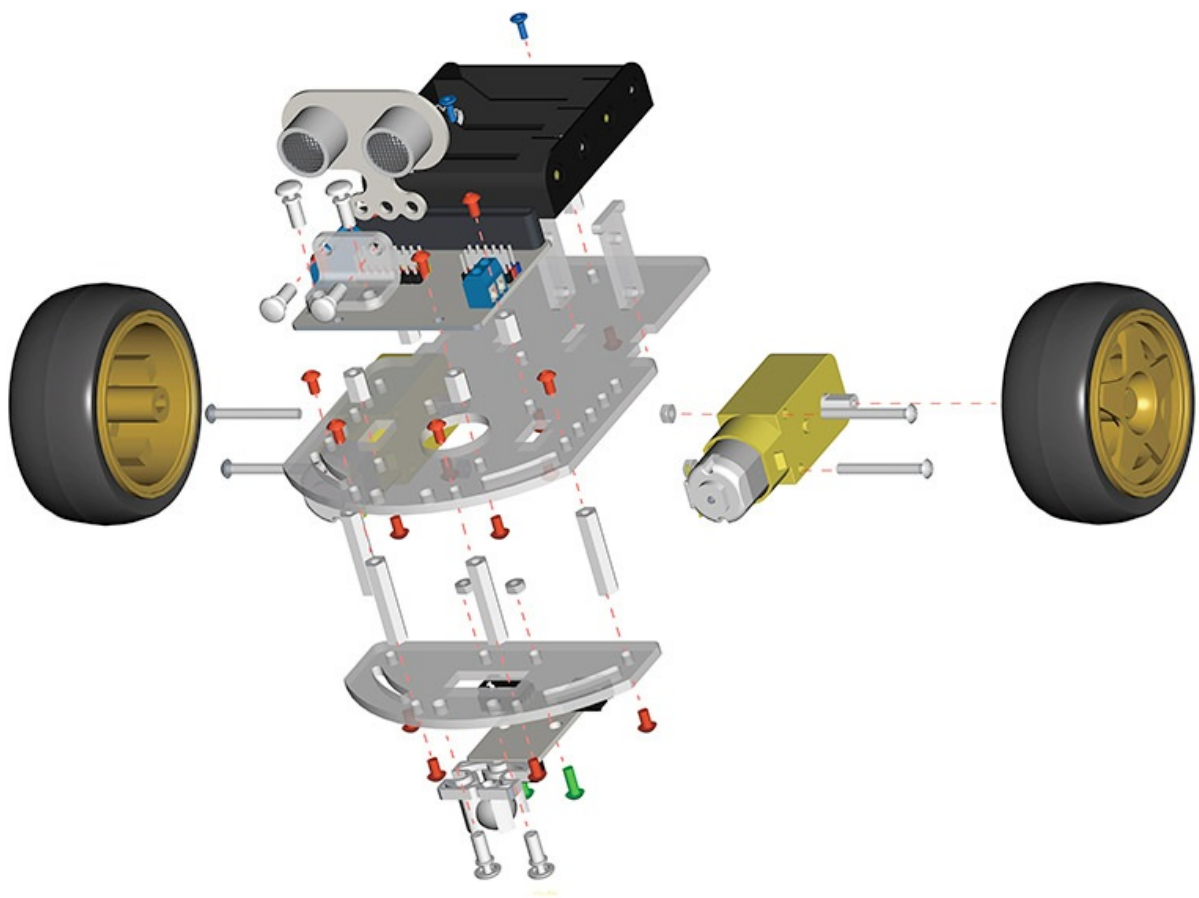
1.1 Montaje

El paquete viene con unas piezas listas para montar que no requieren especial destreza DYC, lo único "*difícil*" es que hay que **soldar** los cables a los motores (4 pequeñas soldaduras).

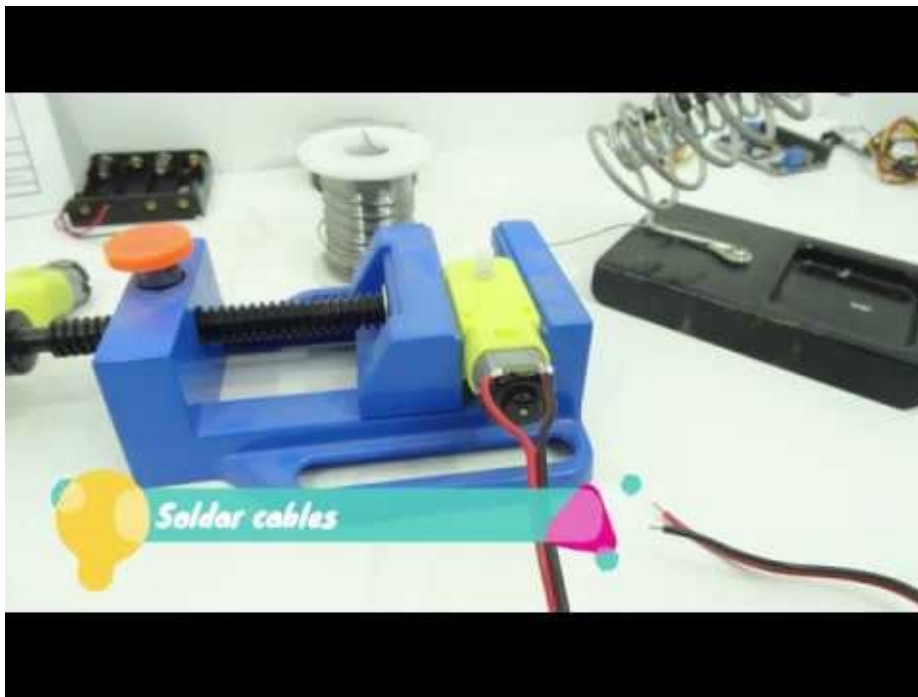
El sensor sigue-líneas y el sensor ultrasonidos se puede poner en cualquier pin, pero en este curso lo fijaremos en:

- Sensor ultrasonidos en el PIN 10
- Sensor siguelíneas izquierda en el PIN3
- Sensor siguelíneas derecha en el PIN4





Con un poco de paciencia se hace muy bien



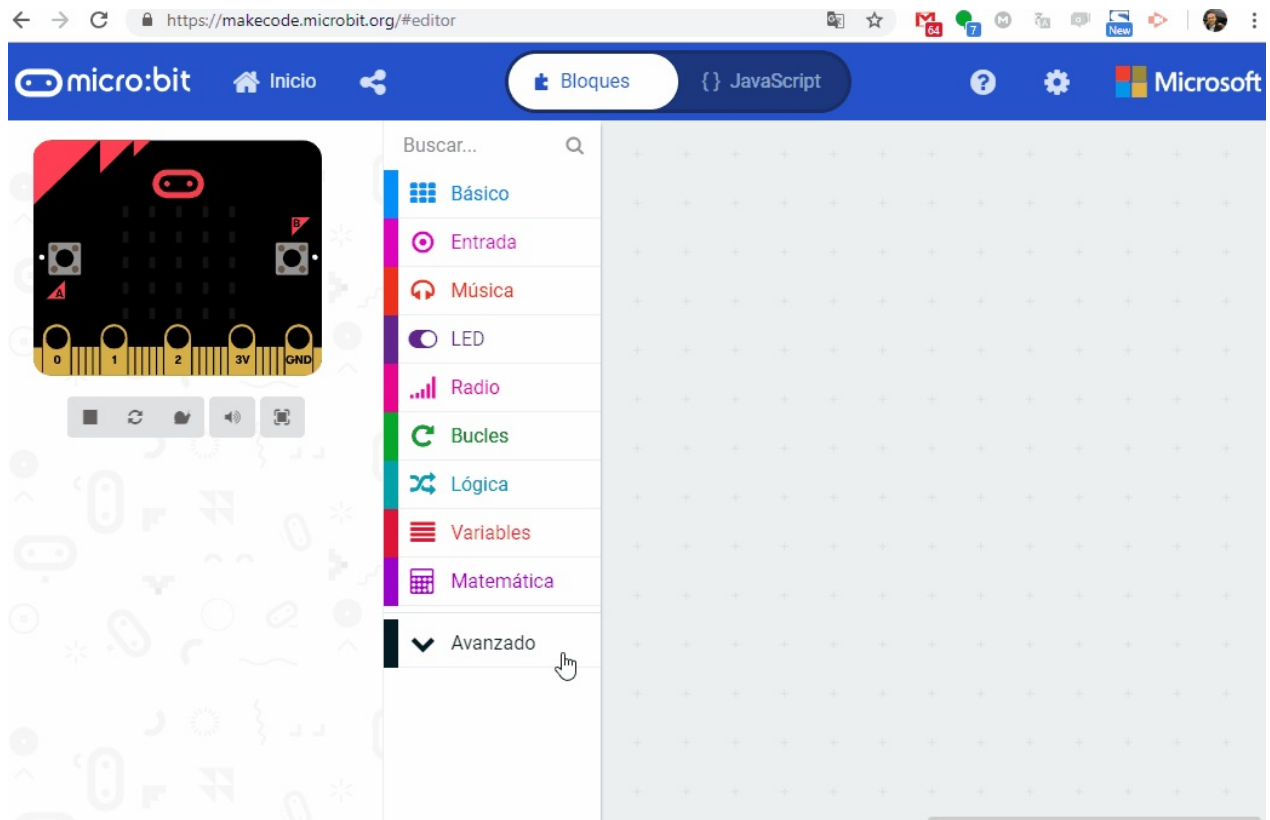
[Video link](#)

Nosotros no somos comerciales, luego ésto mejor que no :



Conexión con Makecode

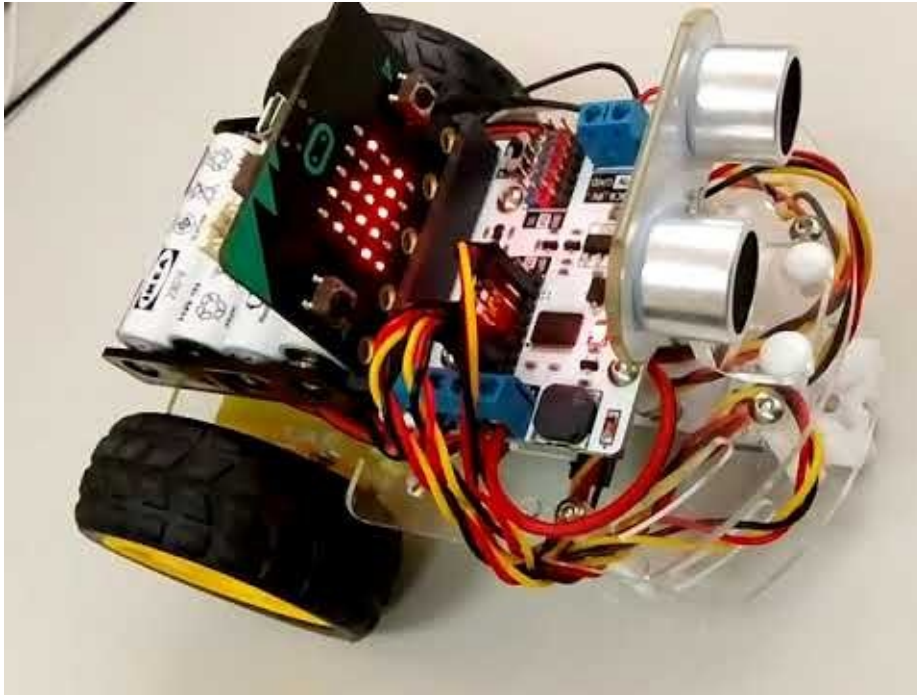
Para tener las instrucciones específicas, hay que entrar en Makecode <https://makecode.microbit.org/#editor> entrar en **AVANZADO** en **EXTENSIONES** y buscar la extensión **MOTORBIT** instalar, y ya está tenemos instalado las funciones específicas de este coche:



Reto 1 Música

SmartCar tiene incorporado un altavoz ya en el puerto P0 por lo tanto no es necesario hacer cocodrilos o papeles de aluminio [como hemos visto anteriormente](#).

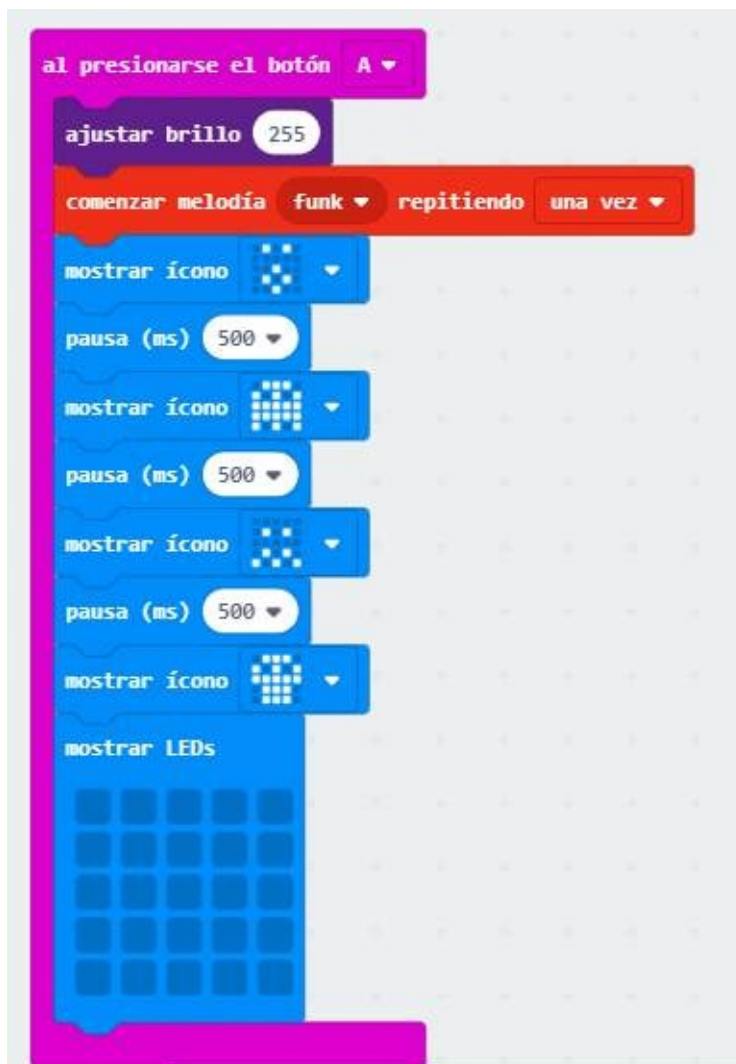
Vamos a hacer un programa que al pulsar el botón A suene música etc...



[Video link](#)

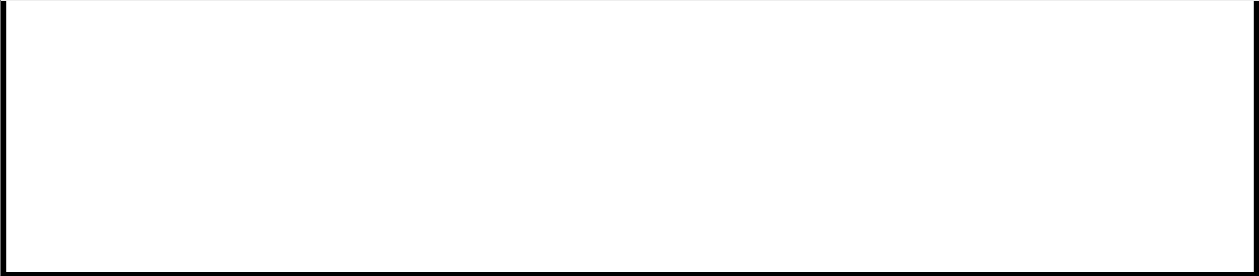
Descripción del programa

Es simplemente que si presiono el botón A pues que suene y salgan iconos por la pantalla:



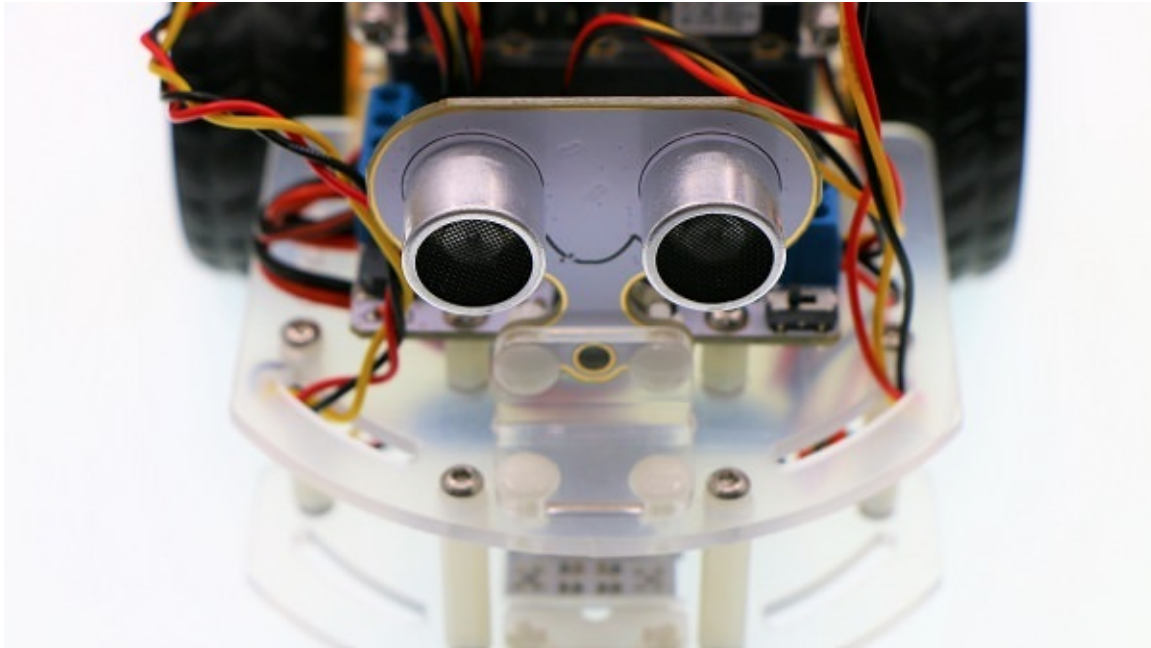
Puedes probarlo en este simulador

loading...



Y [aquí](#) lo tienes para descargar:

2 Sensor distancia



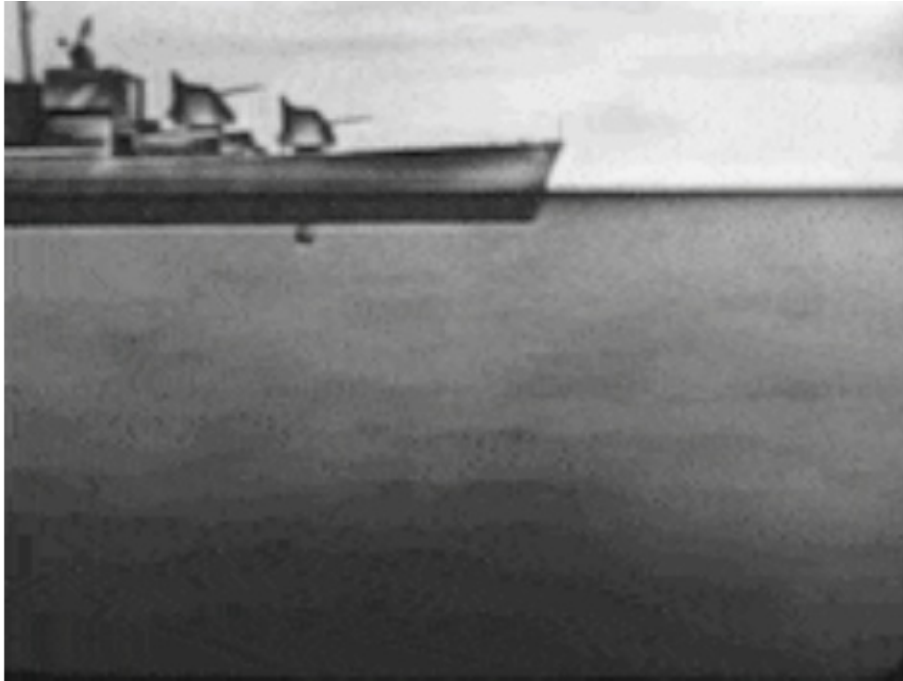
Al poner la extensión micromotor se añade la instrucción de sonar que está en otro apartado *Sonarbit*:



Este sensor, en nuestro kit lo conectaremos en **PIN 10** aunque puede estar conectado en cualquier otro. Además recomendamos crear una variable (por ejemplo "distancia") y poner las unidades de medida en cm que son más intuitivas. Esta instrucción se pone al principio del bucle y sólo hay que utilizar la variable **distancia**:



El sensor de distancia funciona igual que los sónares: Uno "ojo" es realmente un altavóz que emite un sonido con frecuencia alta (*ultrasonido, por eso no lo oímos*) y el otro "ojo" es un micrófono que recibe ese pulso. El circuito electrónico calcula la distancia con la diferencia de tiempo en emisión y la recepción del eco, igual que los radares, sónares...



via [GIPHY](#)

O incluso los murciélagos, delfines...

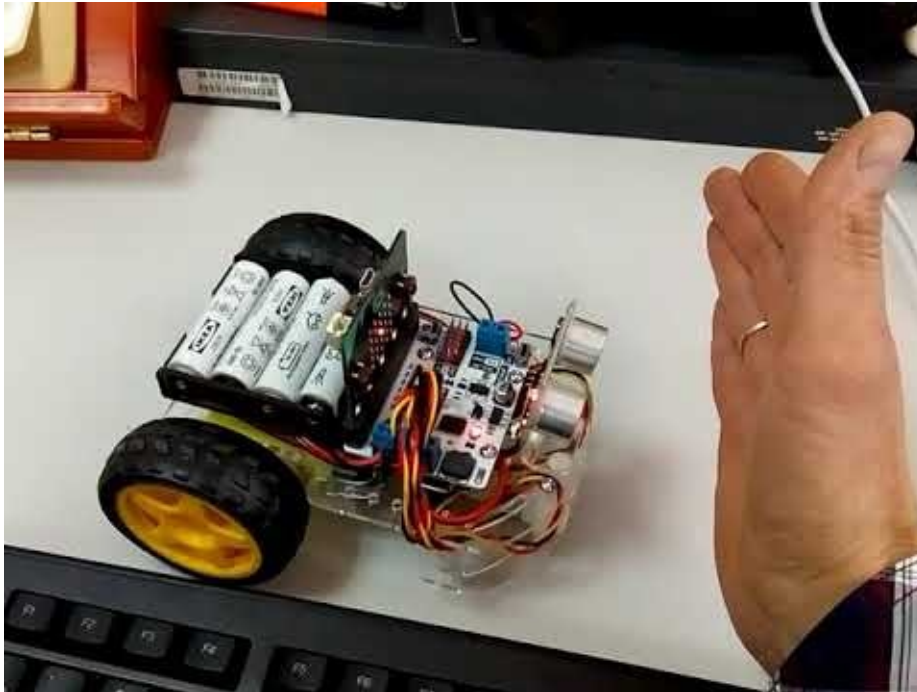


via [GIPHY](#)

Reto 2 Me quedo a 5cm

El siguiente reto es que el microCar se quede a 5cm

ATENCIÓN vamos a poner el sensor de ultasonidos en el PIN 10



[Video link](#)

Solución

La solución se encuentra en esta página https://www.electronicsforu.com/learn-en/motor_bit_smart_car_case_02/

El proyecto te lo puedes descargar [aquí](#)



This site uses cookies for analytics, personalized content and ads. By continuing to browse this site, you agree to this use.

[Learn more](#)



Reto

Que empiece parado y que se aleje si ponemos la mano, o sea, [intenta cogermelo](#)

ee to this

[Learn more](#)

Reto 3 Evitar obstáculos

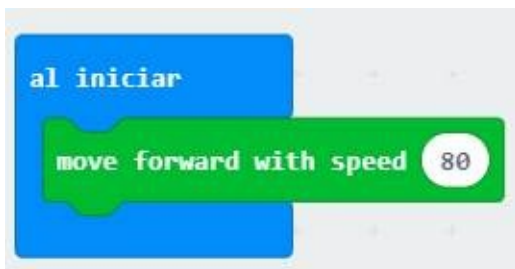
¿Cómo no? lo pide a gritos ! vamos a hacer un [roomba](#)



[Video link](#)

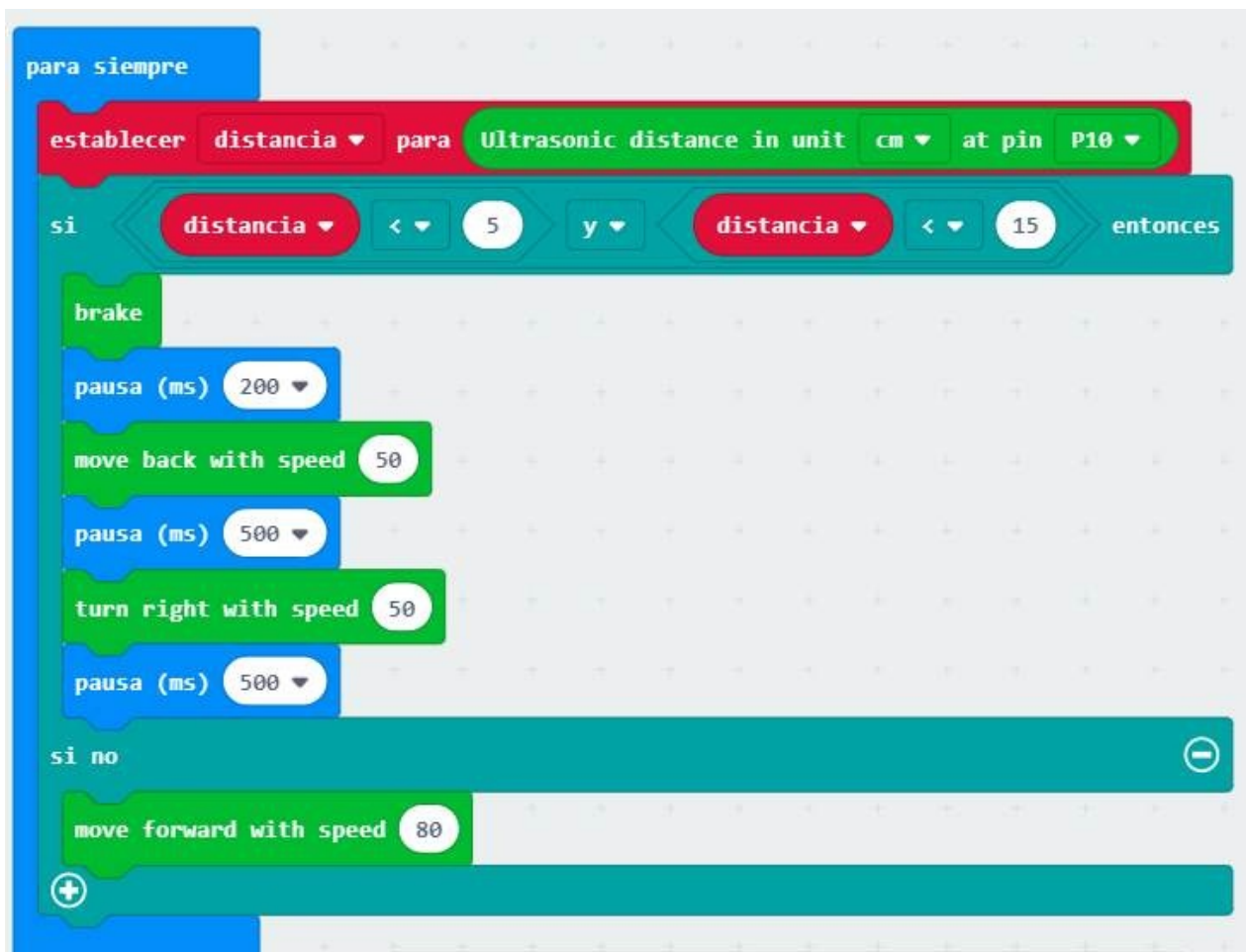
Descripción del proyecto

Empezamos el programa que al iniciar vaya recto:



Luego hacemos el bucle "para siempre":

- El sensor no funciona muy fino, hay veces que da *0 falsos* por lo tanto el bucle **si** ponemos una condición **si es mayor de 5 y es menor de 10** para quitarnos estos falsos positivos de obstáculos.
- si encuentra obstáculo, que pare un poco, que *recule* y que gire
 - cada instrucción anterior con una pequeña pausa, cuanto más grande sea la pausa más *regulará*, girará etc..
- si no encuentra obstáculo que siga recto



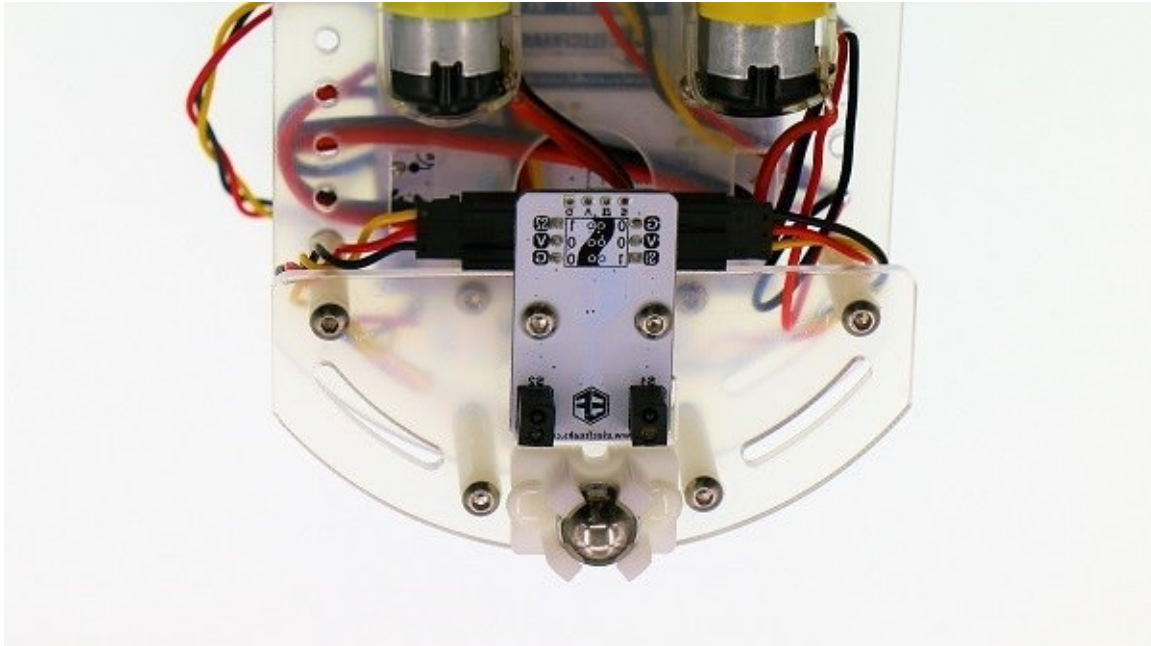
Aquí lo tienes en editor

Reto

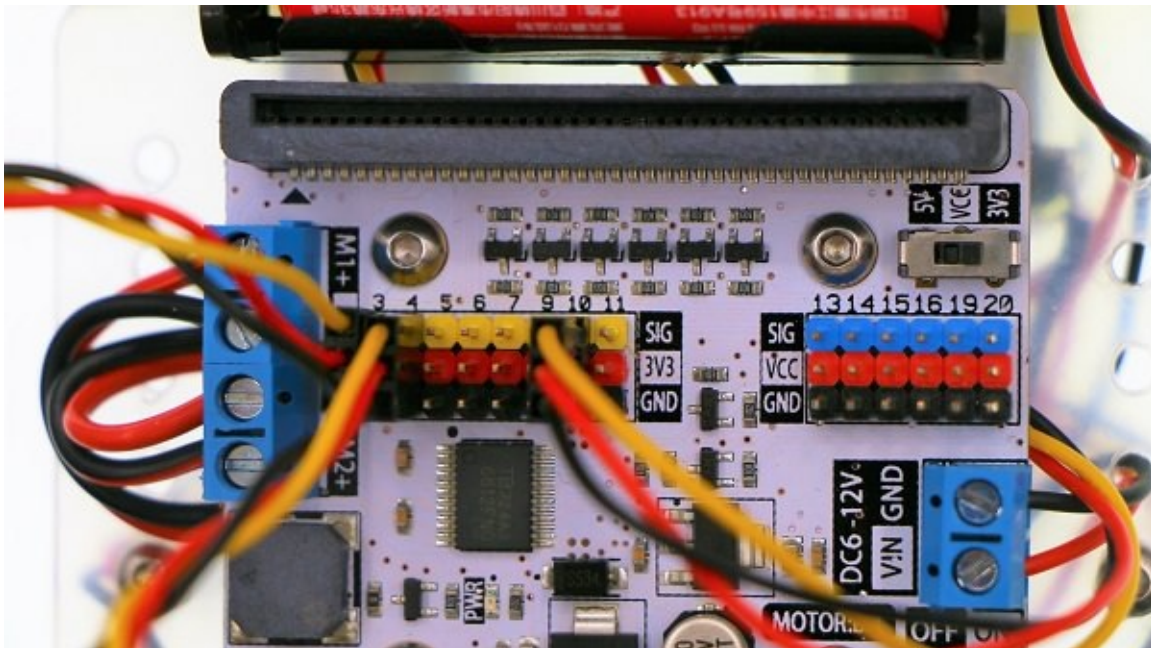
Si te fijas sólo gira a la derecha. Modifica el anterior programa para que gira a la derecha o a la izquierda de forma aleatoria.

Sigue líneas

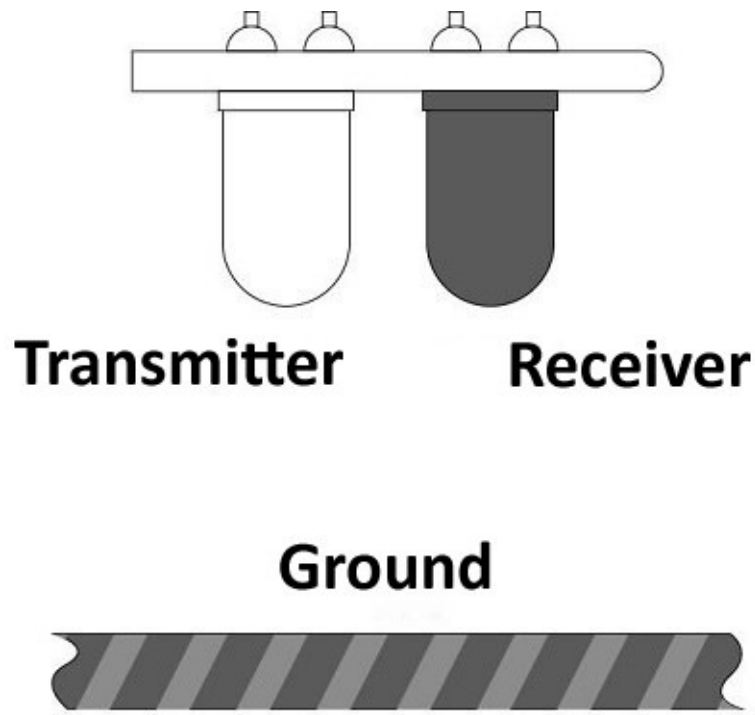
El sigue-lineas son dos sensores que están colocados debajo del robot



Por unificar criterios, los conectaremos en P3 y P4



Cada sensor tiene dos leds, uno emisor y otro receptor. El receptor recoge la luz reflejada, si hay debajo algo que no refleja la luz (por ejemplo una línea negra) entonces manda OFF en caso contrario ON



OJO VA AL REVÉS es decir

- cuando hay linea negra es OFF
- cuando no hay linea es ON

por lo tanto queremos:

- Que cuando sea OFF sea un 1 lógico (línea)
- Que cuando sea ON sea un 0 lógico (no hay línea)

Esto se llama **CONFIGURACIÓN PULL-UP** (pincha [aquí](#) para saber más) luego lo primero que tenemos que hacer es configurar estos sensores como PULL-UP con estas instrucciones (han traducido **UP** como *subir*):



Están un poco escondidas:



Y luego crear unas variables por ejemplo **izquierda** y **derecha** dentro del bucle que lean esos sensores. El resto del código sólo utilizaremos estas variables:



Reto 4 Seguir la línea

Otro reto que pide a gritos este sensor:



[Video link](#)

Consejo: en clase, antes de atacar con este reto, aconsejamos otro más sencillo como que el robot se mueva y si encuentra línea que se pare, este reto lo puedes ver [aquí](#)

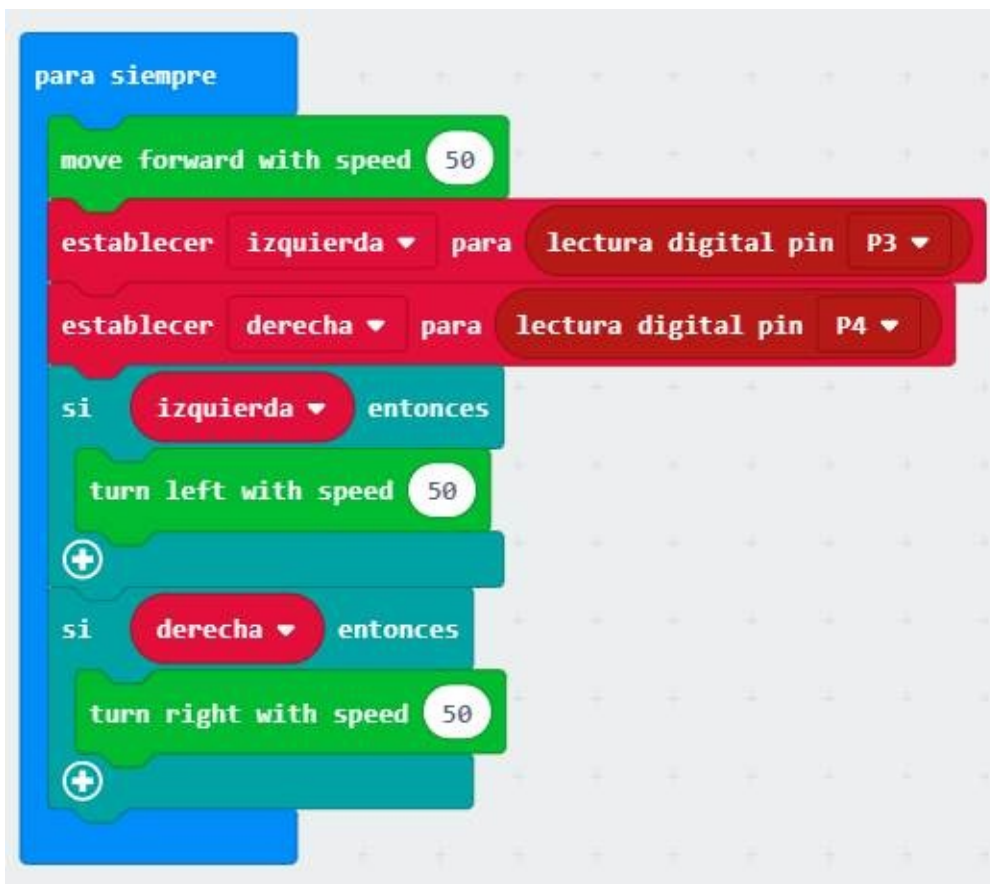
Descripción del programa

Al empezar configuraremos los sensores en PULL-UP tal y [como hemos explicado](#):



Luego entramos en el bucle donde simplemente dice:

- Sigue hacia **delante**
- Lee los **sigue-líneas**
- Si te desvías hacia la **derecha** gira a la **izquierda**
- Si te desvías hacia la **izquierda** gira a la **derecha**



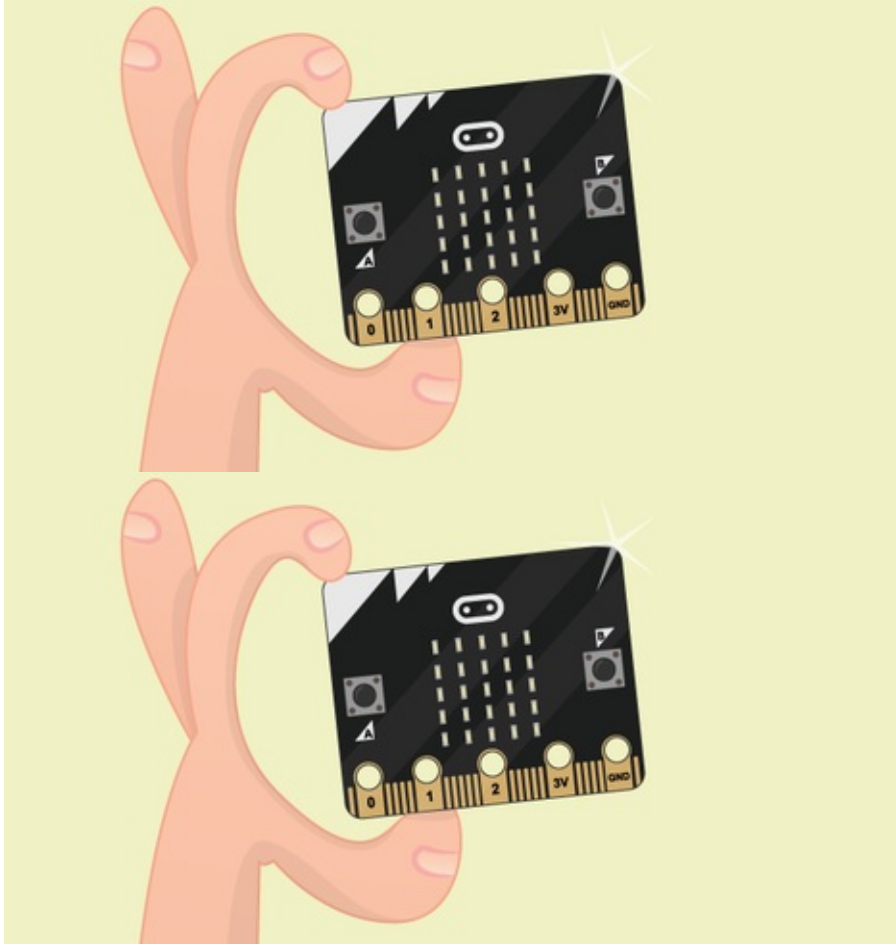
El programa tiene la pega

que si aumentamos la velocidad, pierde la línea.

Evidentemente hay muchas versiones y mejoras, por ejemplo [aquí](#) pero ésta https://makecode.microbit.org/_U3VP8JhVTXaJ es desde luego la versión más sencilla.

4 Si tuviera 2 micro:BIT

Si existe la posibilidad de tener un segundo micro:BIT uno de ellos puede hacer funcionar el Smartcar y el otro de **MANDO A DISTANCIA** gracias a su función de **radio**



By Raspberry Pi Foundation - <https://www.raspberrypi.org> [CC BY-SA 4.0], via Wikimedia Commons

Rento 5 Mando a distancia

Esta vez **vamos a utilizar dos micro:BITs**



[Video link](#)

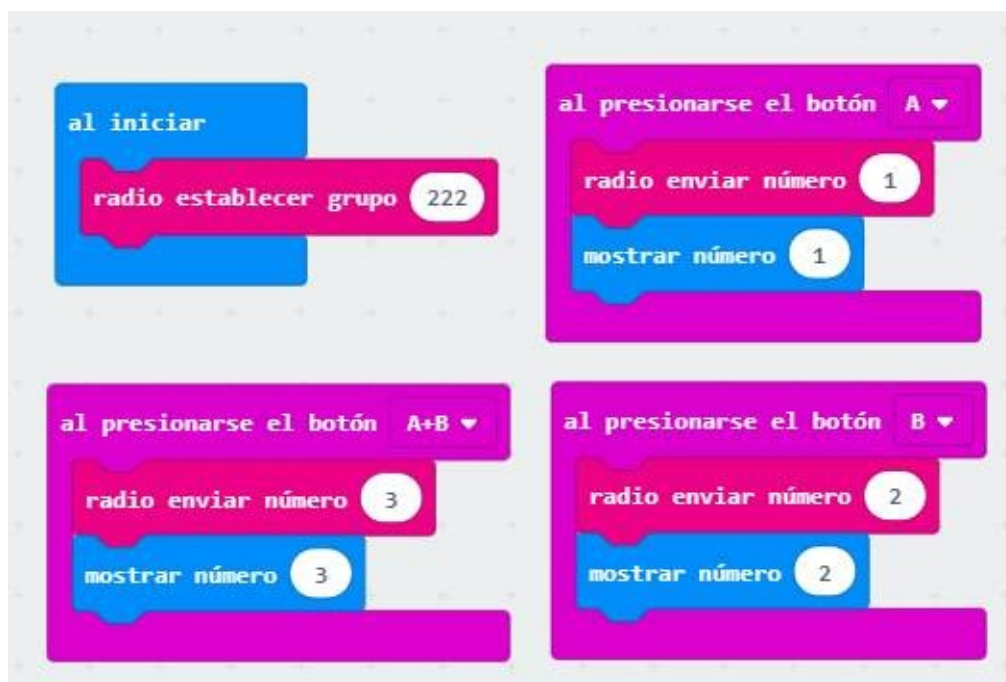
Descripción del programa

Microbit que hace de mando

Este microBIT hay que alimentarlo con pilas o utilizando una batería típica de móvil.

El mando se inicia en un grupo (en este caso el 222) y simplemente realiza lo siguiente:

- Si se pulsa A manda un 1 y lo visualizo
- Si se pulsa B manda un 2 y lo visualizo
- Si se pulsa A+B manda un 3 y lo visualizo



El programa te lo puedes descargar [aquí](#):

Microbit que está en SmartCar

Al iniciar el programa asigna este microbit al mismo grupo de radio que el mando y además asigna una nueva variable con valor 0

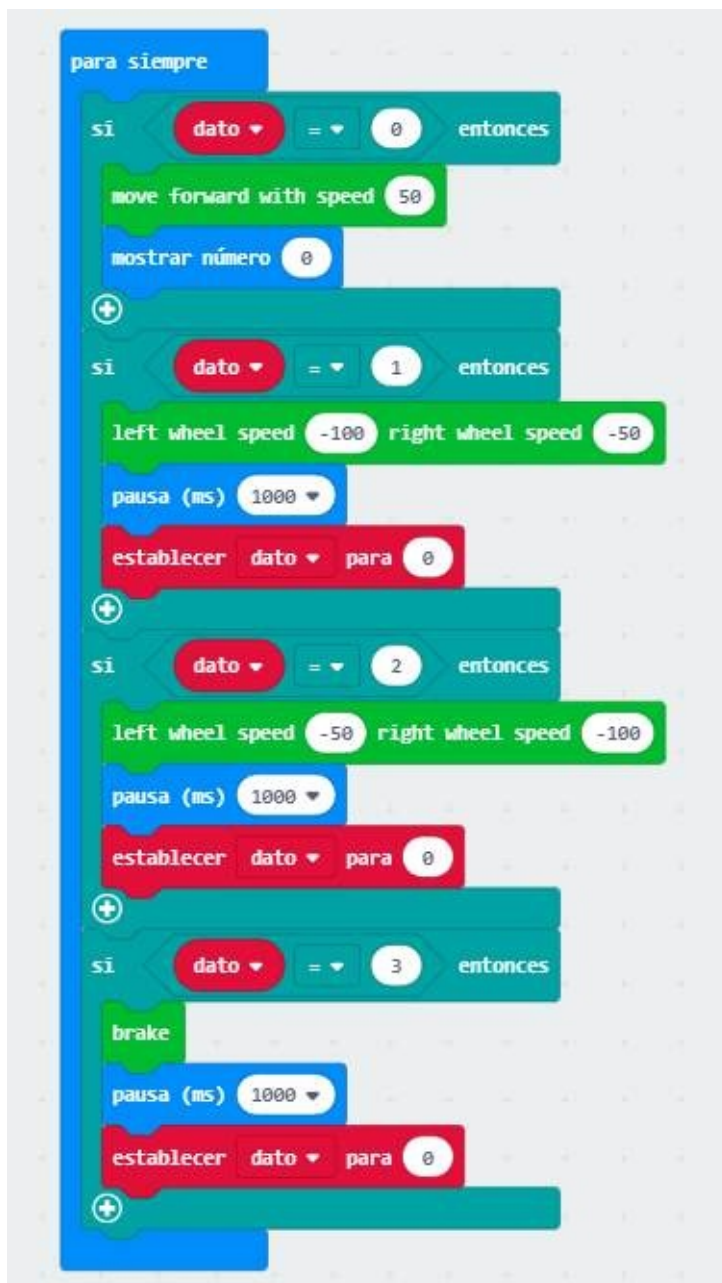


Al recibir un número lo asigna a esa variable y además lo muestra:



Y establecemos un bucle por siempre que :

- Si no ha recibido nada, es por lo tanto dato=0 luego que siga hacia delante
- Si recibe 1 que gire hacia la derecha y hacia atrás
- Si recibe 2 igualmente pero al otro lado
- Si recibe 3 que pare
- En los tres casos anteriores damos un tiempo para que ejecute la instrucción con una pausa y luego reseteamos dato para que siga el robot su camino



El proyecto te lo puedes descargar [aquí](#) :



via GIPHY

Nota:

Intenté también hacer un mando a distancia que funcionase con la brújula interna de microbit, es decir:

- Si pulsaba A entonces se ponía en marcha
- Si giraba el mando a distancia, el robot giraba en el mismo sentido
- Si pulsaba B entonces se paraba

No conseguí que funcionase fino, por lo que he dedicado quitarlo del curso, pero si alguien lo quiere probar, [aquí](#) está el programa del mando y [aquí](#) el del coche.

MURO proyectos MICROBIT

Colabora en este muro <https://padlet.com/CATEDU/microbit> añadiendo proyectos que veas interesantes:



Made with padlet

Muro MicroCar

Este [muro](#) es sólo exclusivo para ejemplos tuyos o de otros interesantes sobre microCAR



Made with padlet

Grupo ROBOTICA EDUCATIVA EN ARAGÓN

En <http://robaragon.chatbro.com> puedes ver un chat sincronizado con Telegram que que te puedes unir con [este enlace](#) para:

- Compartir dudas y resolverlas
- Propuestas, eventos e información
- Ver lo que hacen otros compañeros



Pensamiento computacional

¿Dónde se encaja este robot? ¿se puede comparar este robot con otros robots de otros cursos que hacemos desde CATEDU?

Esta [es la hoja de ruta](#), no se tiene que tomar al pie de la letra, pero intenta ayudar al profesorado que tenga una visión global de tanta oferta:

Como se puede ver MICRO:BIT y smartCAR es una alternativa muy buena frente a mBot y si ya se trabaja con micro:BIT es un buen complemento y ampliación.

RoboTICa

Oferta de formación en Pensamiento computacional del Centro Aragonés de Tecnologías para la Educación.



Créditos

Autoría

- **Colaboradores:**
-

Cualquier observación o detección de error por favor aquí sosporte.catedu.es

Los contenidos se distribuye bajo licencia Creative Commons tipo BY-NC-SA.



GOBIERNO DE ARAGON

Departamento de Educación,
Cultura y Deporte

CATEDU



CENTRO ARAGONÉS de TECNOLOGÍAS para la EDUCACIÓN

