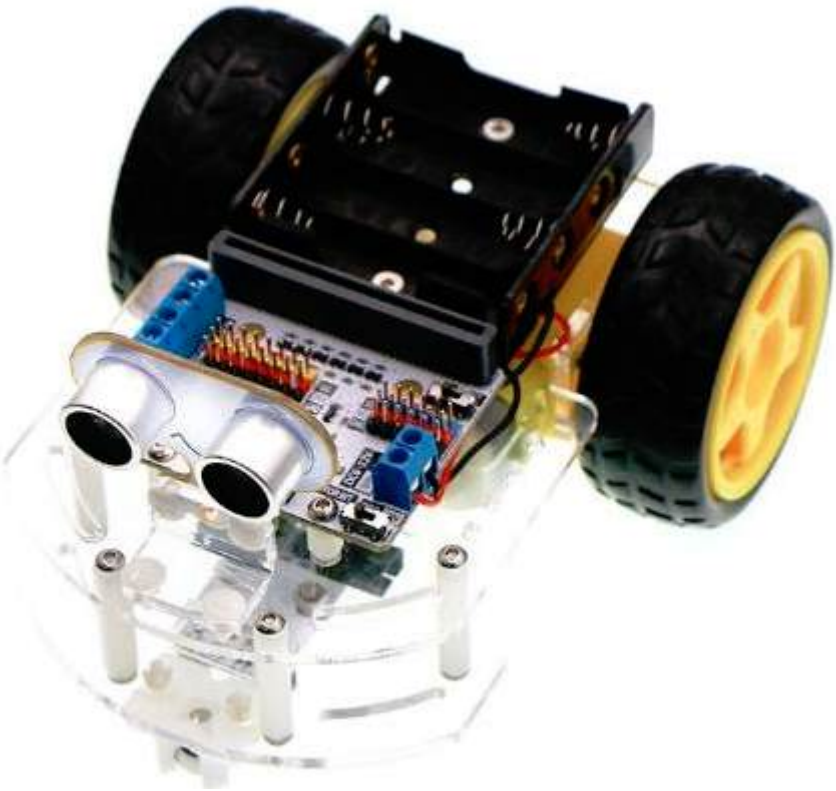
Smartcar

- Smartcar (desfasado)
- Sensor distancia
- Sigue-lineas
- Si tuviera DOS micro:bits
- Muro micro:BIT

Smartcar (desfasado)

> **ATENCIÓN: ESTE ROBOT YA NO SE ENCUENTRA EN EL MERCADO. SE MANTIENEN LOS TUTORIALES POR SI ALGÚN CENTRO TIENE ESTE EQUIPO**

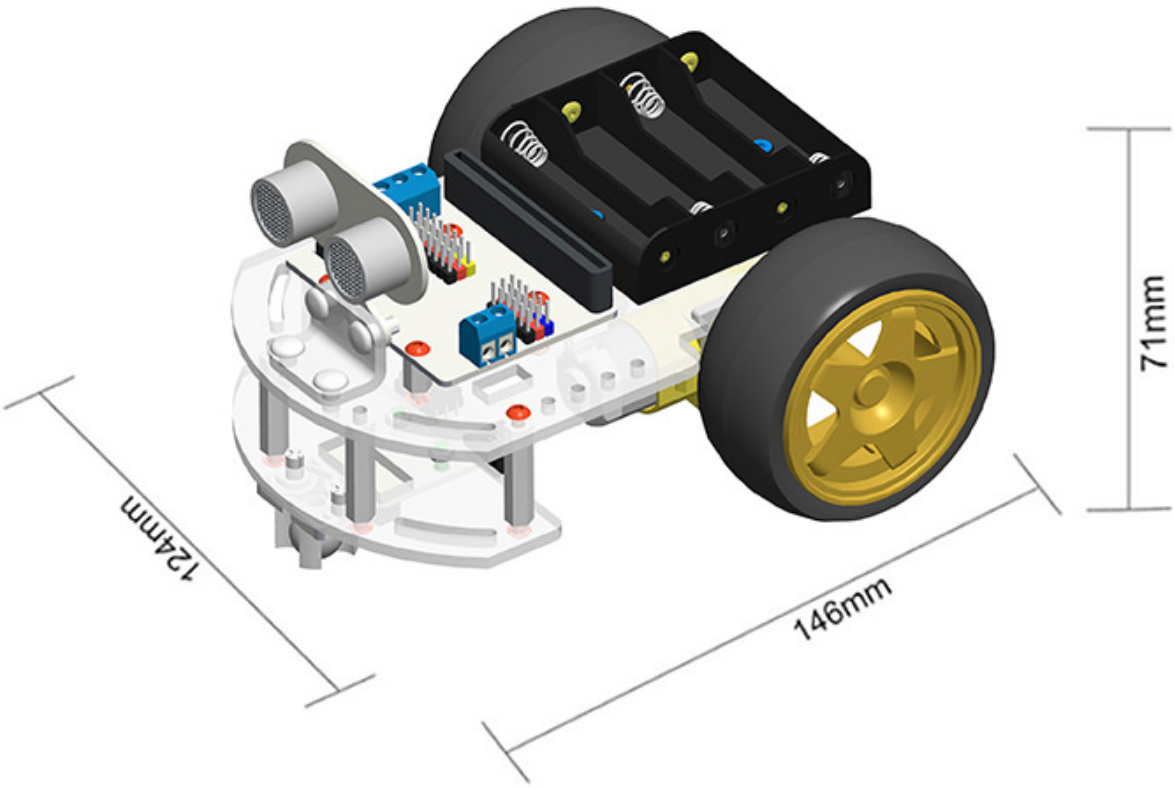


Micro:BIT se queda un poco triste si no se "mueve", pero hay que tener en cuenta que en la robótica el movimiento se paga:

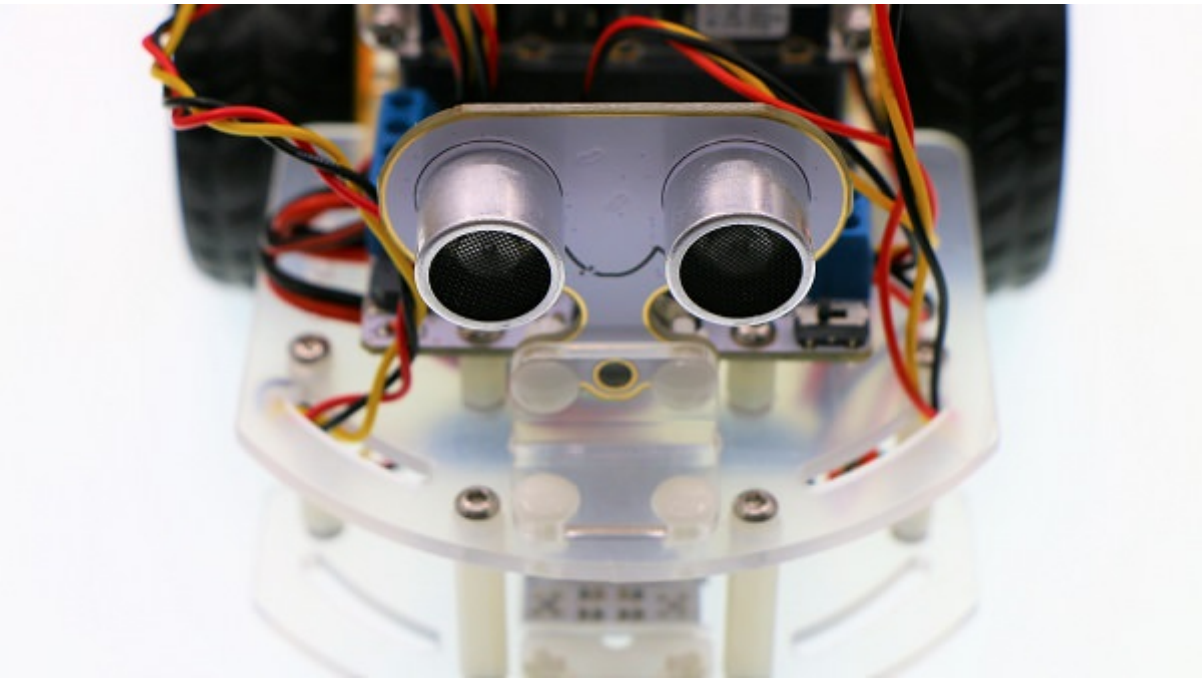
- Adición de motores, chasis, ruedas...
- Necesitas una electrónica añadida de potencia y control de los motores.
- Necesitas unos sensores de lo contrario el movimiento se queda insípido sin reacción al mundo exterior.

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.

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.

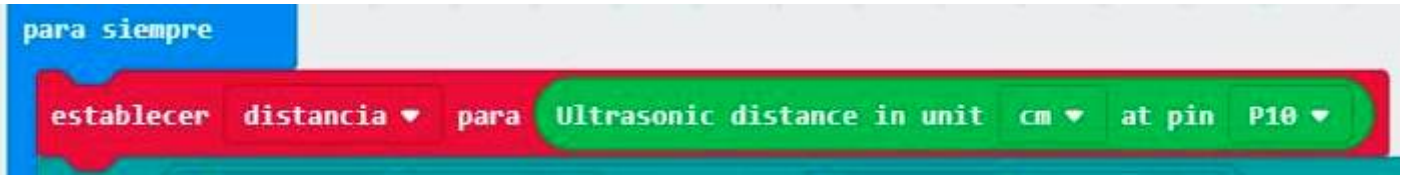

71mm
124mm
146mm

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 rádares, sónares...

https://giphy.com/embed/4xGCaTMCO59le

via GIPHY

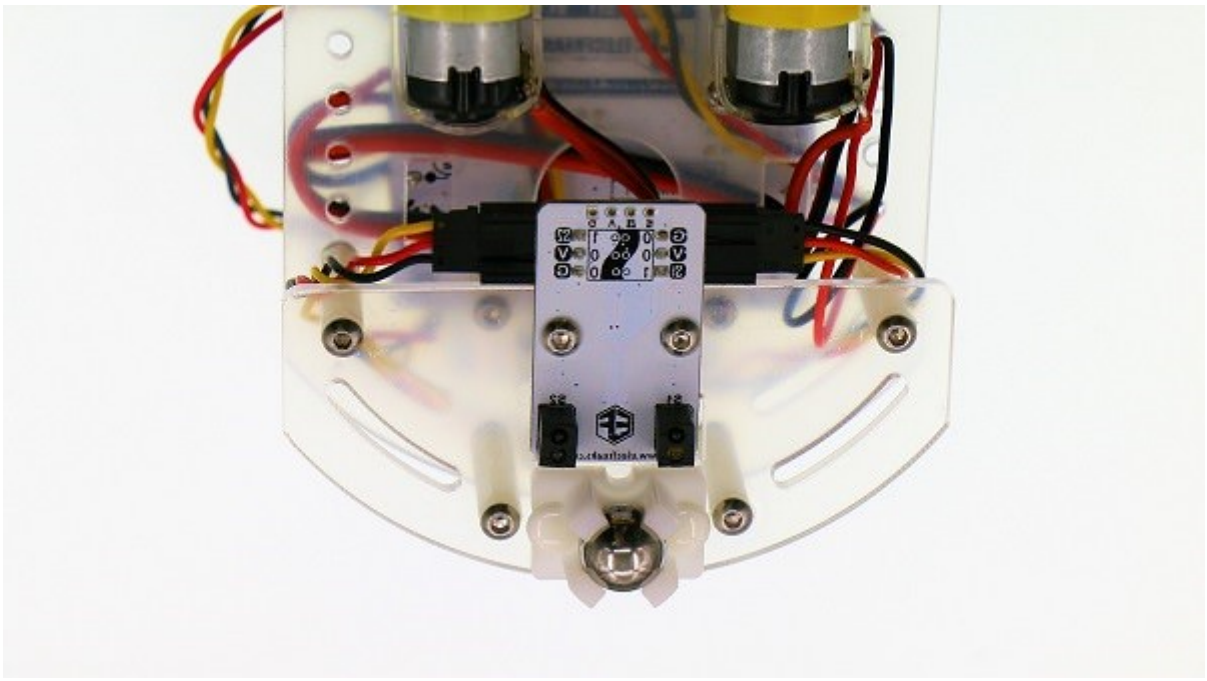
O incluso los murciélagos, delfines...

https://giphy.com/embed/3o7TKu5aIDY4tU3SXm

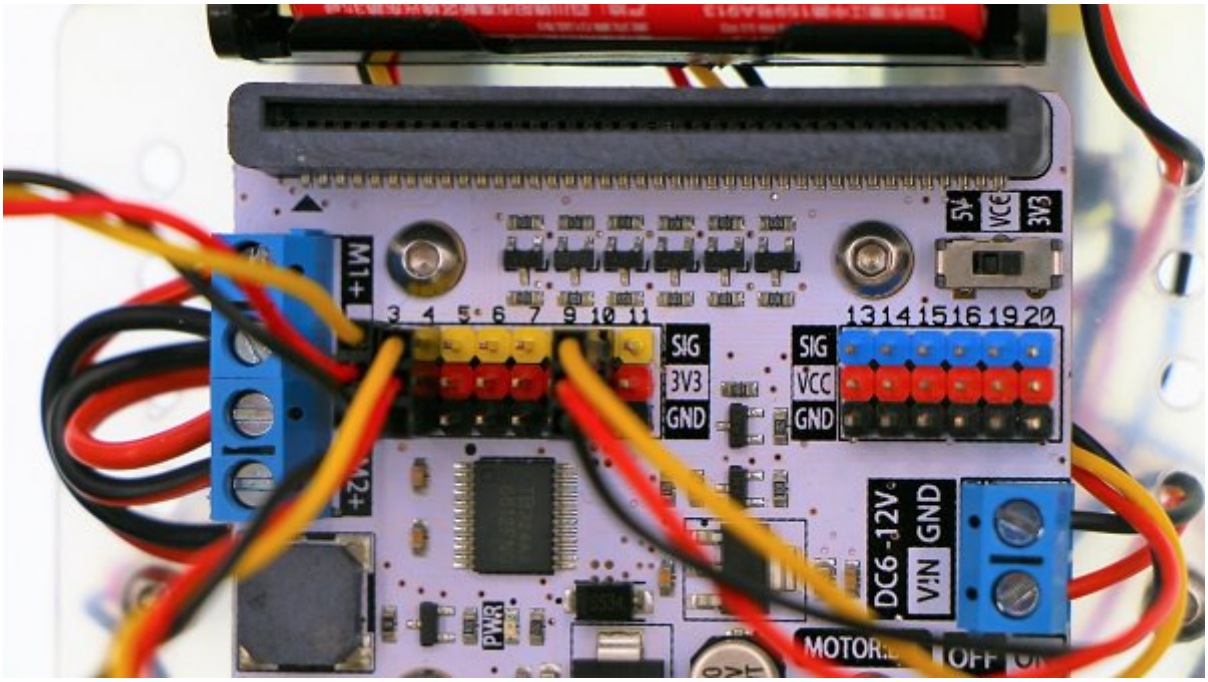
via GIPHY

Sigue-lineas

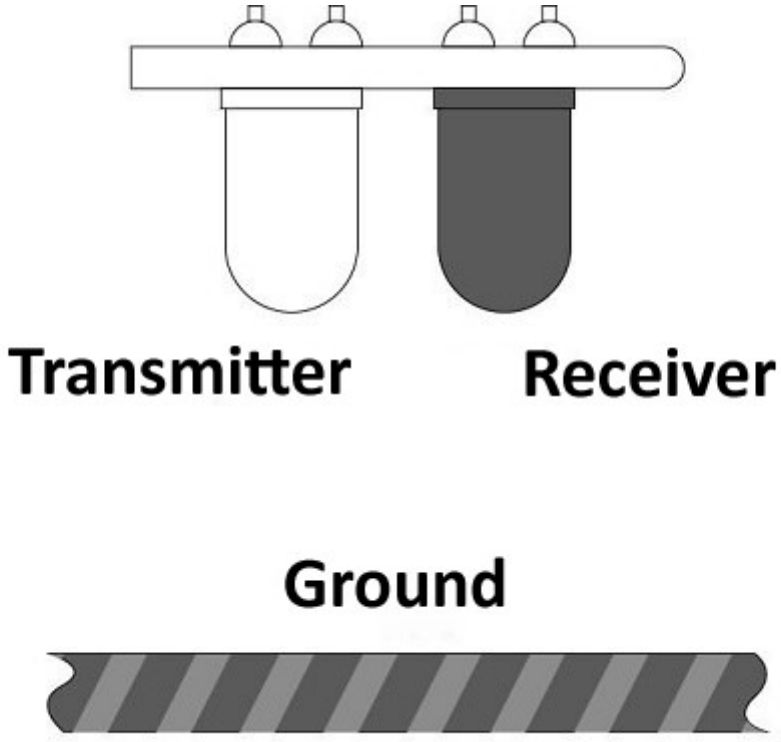
El sigue-lineas son dos sensores que están colocados debajo del robot



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 ejemlo 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:

Si tuviera DOS micro:bits

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**

RPi learning Getting Started with the micro-bit RPi learning Getting Started with the micro-bit

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

https://www.youtube.com/embed/oxumk9GYoVU

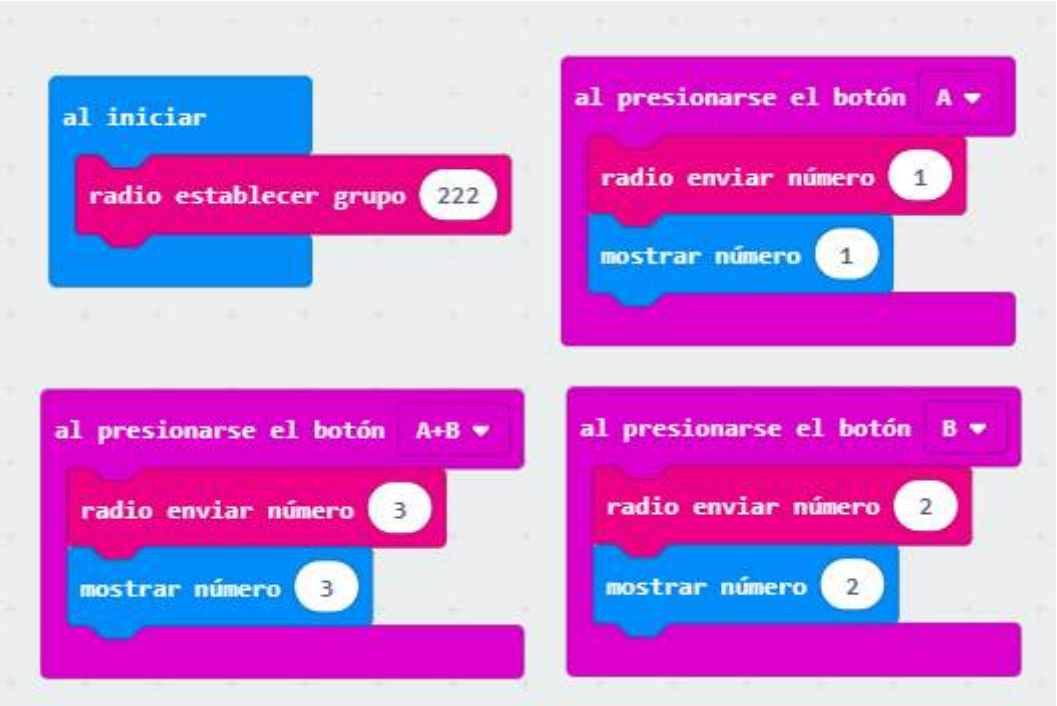
Descripción del programa

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:

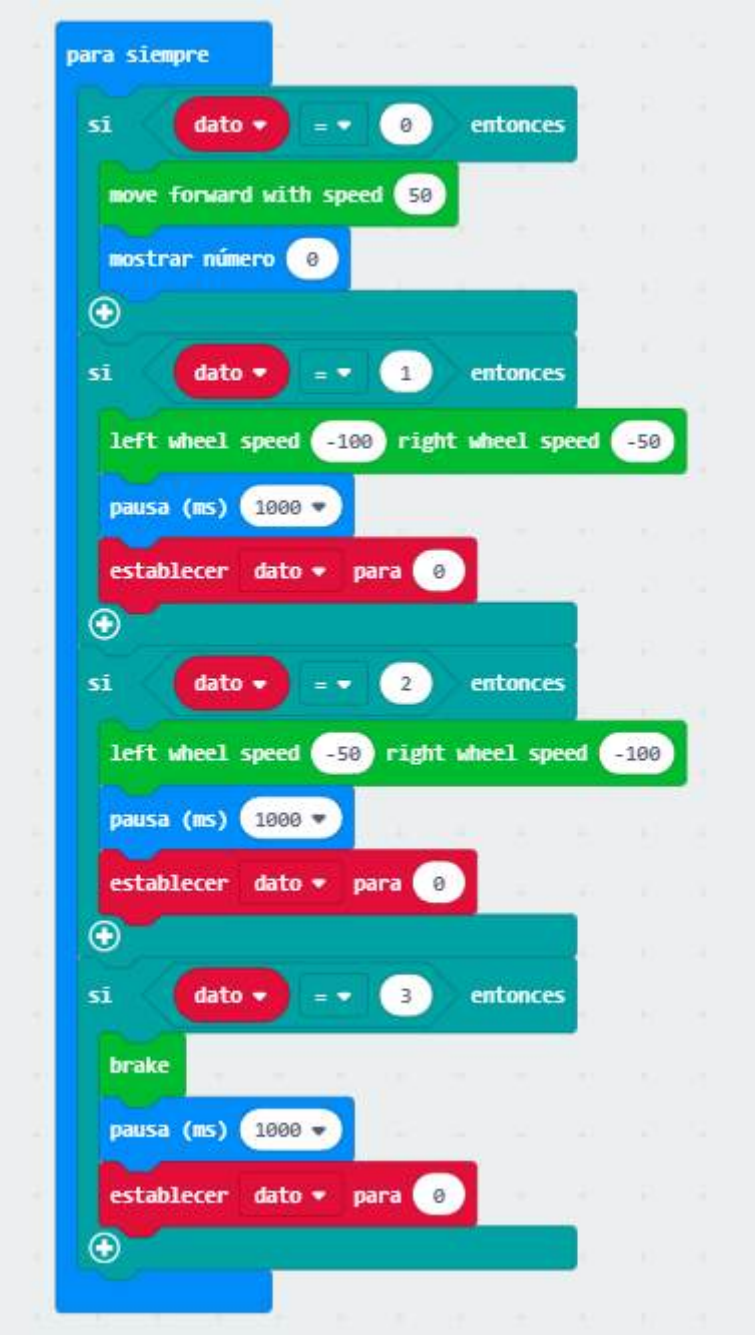
Microbit que está en SmartCar

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 :

https://giphy.com/embed/AiTagxd6jrmiUqq8D2

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 dedicido quitarlo del curso, pero si alguien lo quiere probar, aquí está el programa del mando y aquí el del coche.

Muro micro:BIT

MURO proyectos MICROBIT

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

https://padlet.com/embed/euk7b9yvu3pe