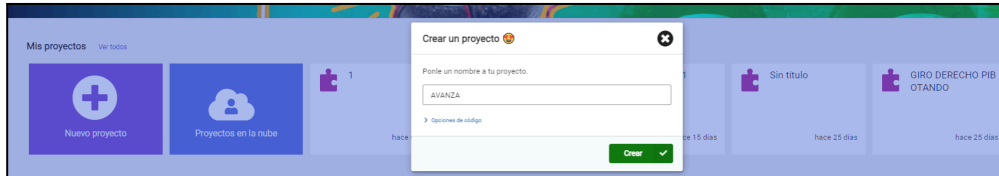


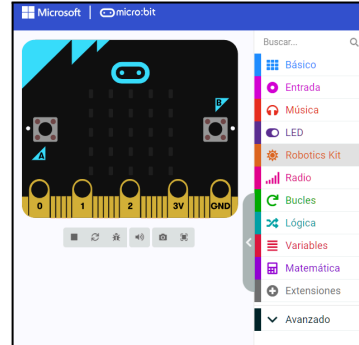
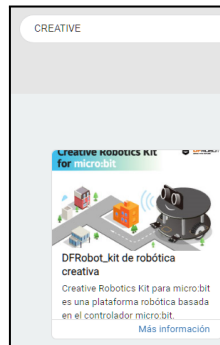
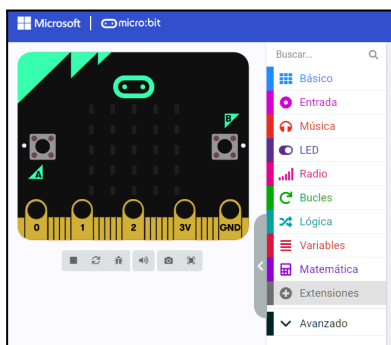
2ª SESIÓN: PROGRAMACIÓN MOVIMIENTOS BÁSICOS DEL KIT CREATIVO DE ROBÓTICA

Antes de empezar a programar, te recomiendo que compruebes que el robot tenga correctas las etiquetas identificativas; M1 (motor izquierdo), M2 (motor derecho). Esto ayudará a entender el movimiento de los robots. Se pueden dar diferentes movimientos a un robot: (las imágenes del robot están mirándolo como si lo siguiéramos por detrás. Por otro lado, la velocidad de los motores se gradúa entre los valores 0 y 255.

1. Abrir la aplicación de programación: [Microsoft MakeCode for micro:bit](#) y selecciona **nuevo proyecto**, pon como nombre **"AVANZA"**.

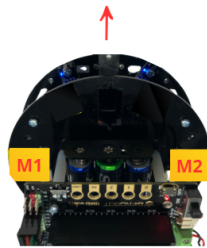


2. Seleccionar **"+ Extensiones"** escribir en el buscador **"Creative"** y seleccionar el Kit: **"Creative Robotics Kit para micro:bit"**, (se amplían así los recursos para la programación), se habrá añadido **Robotics Kit**.



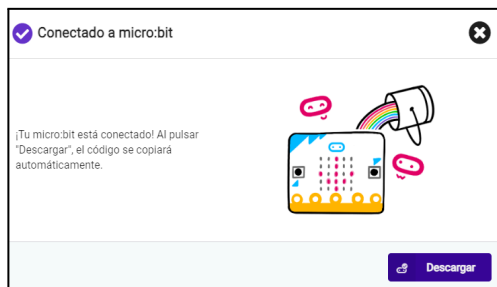
3. Empezamos a programar diferentes movimientos que llamaremos RETOS, mira los colores de los bloques.

RETO 1: Avanza 500 ms y paro:



4. Para pasar la programación de MakeCode al robot, procederemos de la siguiente forma:
 - a. Conectar el cable de la micro:bit, del ordenador a la placa micro:bit.
 - b. En la parte inferior de la aplicación, dar a **Guardar** (no a Descargar) / **Emparejar ahora** / seleccionar la placa que aparece y dar a **conectar** / **Listo** / y por último **Descargar** (tal y como aparece en las imágenes de abajo).



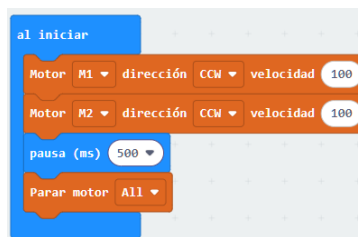
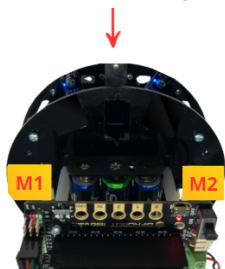


- En descargas / copiar y pegar (o arrastrar) el archivo descargado en el puerto de la placa micro:bit (el led de la micro:bit parpadea hasta que se descargue totalmente el programa).
- IMPORTANTE: desconectar el cable de la placa micro:bit y activar el interruptor de la Placa de Expansión.

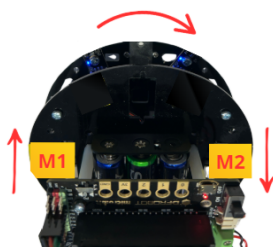
SEGUIMOS PROGRAMANDO MÁS MOVIMIENTOS BÁSICOS DE UN ROBOT

Después de descargar cada programa, puedes cambiar el nombre y modificar la programación, una vez terminado, dar nuevamente a descargar, así tendrás todos los retos descargados. En vez de empezar de nuevo cada programa, puedes modificar los anteriores o ir ampliando la programación debajo para que el robot haga varios movimientos seguidos, en este caso tendrás que meter un pausa de 500 ms al parar los motores

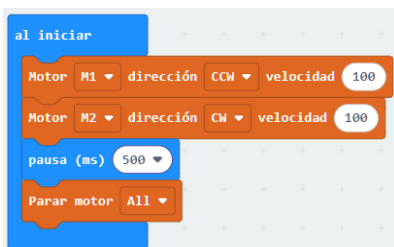
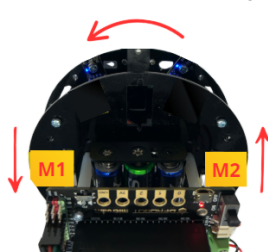
RETO 2: Retrocede 500 ms y paro:



RETO 3: Giro Derecha 500 ms y paro:



RETO 4: Giro Izquierda 500 ms y paro:



RETO 5: El robot gire 90° y se pare, tendrás que ajustarlo con el pausa (ms), dependerá de lo gastado que esté la batería.

RETO 6: El robot se mueva haciendo un cuadrado y pare. (Pruébalo en el suelo para evitar que se caiga el robot)

RETO 7: El robot se mueva haciendo un 8 y pare. (Pruébalo en el suelo para evitar que se caiga el robot)

3ª SESIÓN: PROGRAMANDO SIGUE LÍNEAS

Para programar un sigue-líneas hay que tener en cuenta el número de sensores que tiene el robot, pudiendo ser de 2-5 sensores, cuanto más sensores mayor precisión.

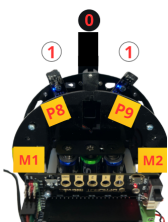
En nuestro caso, tenemos dos sensores infrarrojos, así que el robot deberá mantener la línea negra siempre entre estos dos sensores, o que es lo mismo, los **dos sensores estén sobre blanco** para poder **avanzar**.

Estos sensores infrarrojos devuelven un **1 si detectan una superficie blanca** (*alta reflexión de luz infrarroja*) y un **0 si la superficie es negra** (*baja reflexión de luz infrarroja*). Esto se debe a que el **negro absorbe la luz** infrarroja, mientras que el **blanco la refleja**, (aunque a veces depende de la configuración del software del infrarrojo, pudiendo tener una lógica inversa). Si el **sensor izquierdo detecta negro, girará a la derecha**, y si es el **sensor derecho**, el que detecta **negro, girará a la izquierda**, para reafirmar y poder seguir la línea.

Antes de empezar te recomiendo que los alumnos una vez hayan montado el robot pongan unas etiquetas identificando los motores; M1 (izquierdo), M2 (derecho) y los sensores infrarrojos P8 (izquierdo) y P9 (derecho), según seguimos al robot en su movimiento. Esto ayudará a entender el movimiento de los robots.

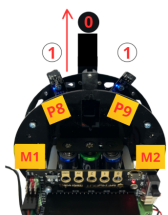
Se pueden dar diferentes situaciones de los sensores con respecto a la línea negra:

1. Sensor P8 (Izquierdo) y sensor P9 (derecho) sobre blanco

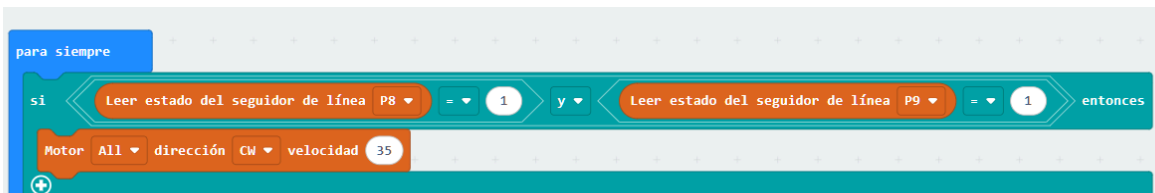


dirección del robot

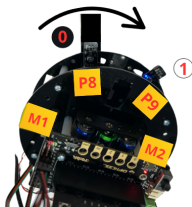
Condición: Si P8 Y P9 reciben un 1 (línea blanca)



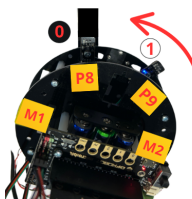
Entonces: el robot avanzará. Los dos motores deben avanzar.
Avanza CW
Velocidad 35 (los motores tienen una velocidad entre 0 y 225)



2. Sensor P8 (Izquierdo) invade la línea negra y sensor P9 (derecho) sobre blanco



Condición: Si P8 recibe un 0 (línea negra) y el P9 recibe un 1 (línea blanca), quiere decir que el robot se ha girado hacia su derecha

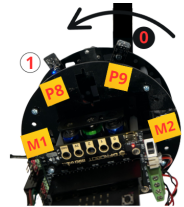


Entonces: el robot tendrá que girar a la izquierda para volver a su estado de avance o línea negra entre sensores.
M1 CCW retrocede más lento
M2 CW avanza

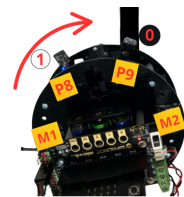
```

si Leer estado del seguidor de línea P8 = 0 y Leer estado del seguidor de línea P9 = 1 entonces
  Motor M1 dirección CCW velocidad 10
  Motor M2 dirección CW velocidad 35
  
```

3. Sensor P8 (izquierdo) sobre blanco y sensor P9 (derecho) invade la línea negra



Condición: Si P8 recibe un 1 (línea blanco) y el P9 recibe un 0 (línea negra), quiere decir que el robot se ha girado hacia su izquierda.



Entonces: el robot tendrá que girar a la derecha para volver a su estado de avance o línea negra entre sensores.
M1 CW avanza 35
M2 CCW retrocede más lento 10

```

si Leer estado del seguidor de línea P8 = 1 y Leer estado del seguidor de línea P9 = 0 entonces
  Motor M1 dirección CW velocidad 35
  Motor M2 dirección CCW velocidad 10
  
```

PROGRAMACIÓN FINAL

```

para siempre
  si Leer estado del seguidor de línea P8 = 1 y Leer estado del seguidor de línea P9 = 1 entonces
    Motor All dirección CW velocidad 35
  si Leer estado del seguidor de línea P8 = 0 y Leer estado del seguidor de línea P9 = 1 entonces
    Motor M1 dirección CCW velocidad 10
    Motor M2 dirección CW velocidad 35
  si Leer estado del seguidor de línea P8 = 1 y Leer estado del seguidor de línea P9 = 0 entonces
    Motor M1 dirección CW velocidad 35
    Motor M2 dirección CCW velocidad 10
  
```

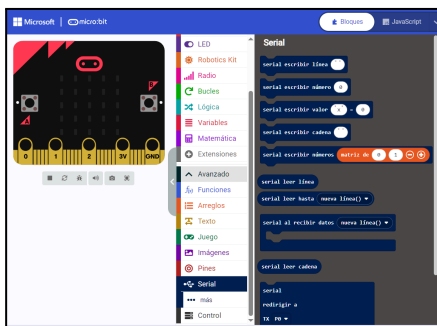
4ª SESIÓN: PROGRAMANDO EVITA OBSTÁCULOS

Para programar un evita obstáculos, hay que conocer el funcionamiento del sensor ultrasónicos que se basa en la propagación de las ondas sonoras.

El sensor HC-SR04, por un lado tiene un elemento que **emite** ondas de alta frecuencia continuamente, (que no puede percibir el oído humano, alrededor de 40 KHz) y un **receptor** que detecta cuándo esas ondas vuelven, al chocar con un objeto, de modo que este sensor calcula el tiempo que tarda la onda en ir y volver y lo convierte en la distancia que hay al objeto.

[Vídeo funcionamiento del ultrasonido más detallado](#)

RETO 1: Obtener **datos por el sensor ultrasónico** y envíe los datos del sensor a través del puerto serie cada 500 milisegundos.



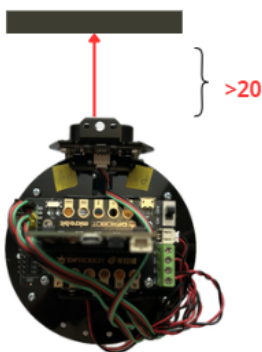
1. Ir a **AVANZADO / Serial / serial escribir línea**
2. **Robotics Kit / Obtener distancia del sensor ultrasónico del pin P0**
3. **Pausa (ms) 500**

para siempre

serial escribir línea Obtener distancia del sensor ultrasónico del pin P0 en (cm).
 pausa (ms) 500

RETO 2: Programación EVITA OBSTÁCULOS

1ª CONDICIÓN: Si.....entonces.....



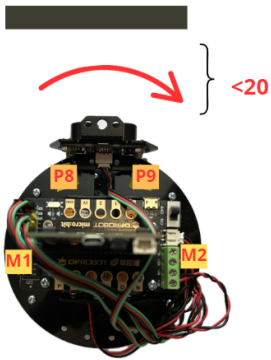
Condición: si la distancia es mayor de 20 cm

Entonces: el robot avanza
 M1 y M2 CW avanza 70 o todos los motores CW avanza 70

para siempre

si Obtener distancia del sensor ultrasónico del pin P0 en (cm). > 20 entonces
 Motor All dirección CW velocidad 70

2ª CONDICIÓN: Si no.....



Condición: “si no” si la distancia es menor o igual de 20 cm (resto de casos diferentes a mayor de 20)

Entonces: el robot para, retrocede y gira hacia un lado (en este caso se ha elegido derecha, pero se puede programar cualquier giro)

```

si no
  Parar motor All
  pausa (ms) 100
  Motor All dirección CCW velocidad 50
  pausa (ms) 200
  Motor M1 dirección CW velocidad 35
  Motor M2 dirección CCW velocidad 35
  pausa (ms) 200

```

Nota: el tener un solo ultrasonidos, hace que el robot tenga puntos muertos, ya que el ángulo de visión del ultrasonidos es de 15° solo, por lo que a veces se choca con obstáculos que se encuentran fuera de su campo de visión superior a los 15°. Para resolverlo o hacer que nuestro robot tenga más precisión, será necesario poner más ultrasonidos, siendo la mejor opción 3 para evitar tener ángulos muertos.

PROGRAMACIÓN FINAL

```

para siempre
  si Obtener distancia del sensor ultrasónico del pin P0 en (cm). > 20 entonces
    Motor All dirección CW velocidad 70
  si no
    Parar motor All
    pausa (ms) 100
    Motor All dirección CCW velocidad 50
    pausa (ms) 200
    Motor M1 dirección CW velocidad 35
    Motor M2 dirección CCW velocidad 35
    pausa (ms) 200

```