

# mBot2

- [¿Qué es mBot2?](#)
- [Cómo usar mBot2 en mBlock](#)
- [Mis primeros pasos](#)
- [Jugando con la luz](#)
- [Evitar obstáculos I](#)
- [Evitar obstáculos II](#)
- [Sigue líneas I](#)
- [Sigue líneas II](#)
- [Control remoto](#)
- [Más actividades](#)

# ¿Qué es mBot2?

Hay que entender mBot2 **como un chasis a CyberPi** es decir, el rotot es CyberPi y le añadimos sensores y actuadores que es mBot2

## Placa del mBot2

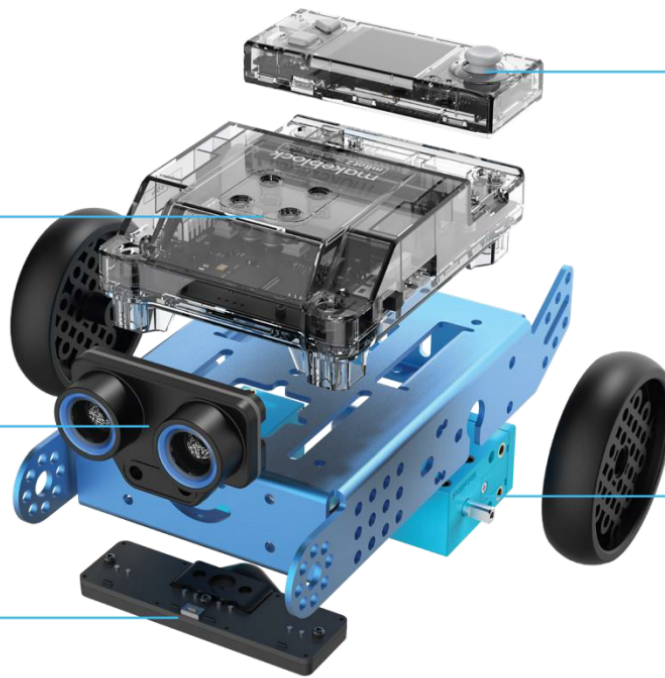
Compatible con una amplia variedad de componentes externos. Incluye una batería recargable incorporada.

## Sensor ultrasónico 2

La detección de objetos incluye 8 LED programables para una interacción mejorada.

## Sensor Quad RGB

Las 4 sondas del sensor admiten el reconocimiento de colores, así como programas de detección de líneas básicos y avanzados.



## CyberPi

Microprocesador ESP-32 para comunicación inalámbrica. Compatibilidad con codificación basada en bloques y codificación Python.

## Motores codificadores

Miden la rotación con una precisión de 1 grado, la distancia recorrida y controlan con precisión la velocidad hasta 200 RPM.

**Manual mBot2** <https://www.robotix.es/documentos/mbot2-actividades.pdf> contenidos e imágenes publicados con permiso de la empresa Robotix.

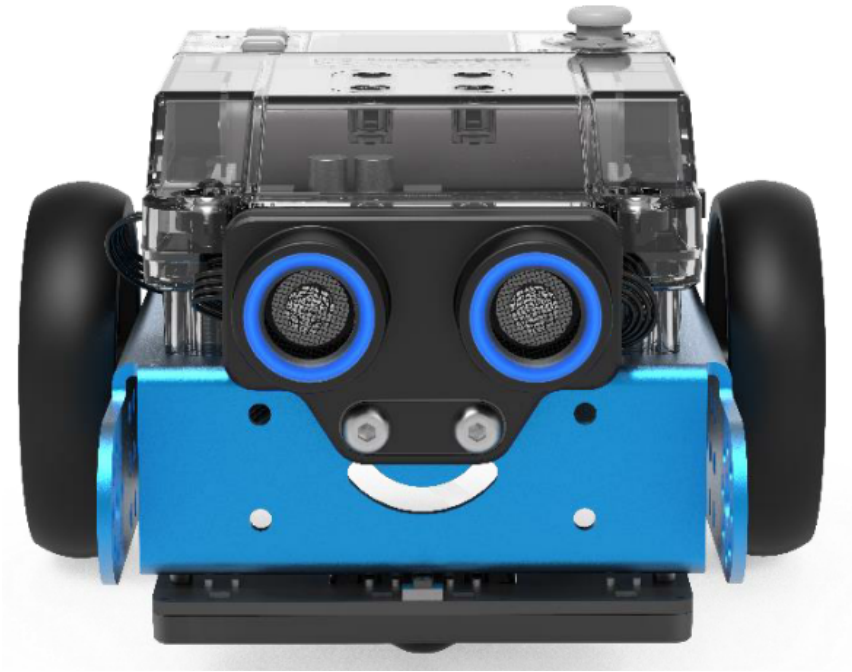
## MOTORES CODIFICADORES

Permite el control de la velocidad y de la posición pues están codificados (motores paso a paso o motores con un detector de posición) por lo tanto permite movimientos precisos y colocar mBot2 en lugares exactos.

Comparandolo con mBot1 su control se realizaba por más o menos potencia a los motores, lo que provocaba desviaciones indeseadas.

## SENSOR ULTRASONICO

Más preciso que su anterior. y además tiene unos leds que permite más interacción

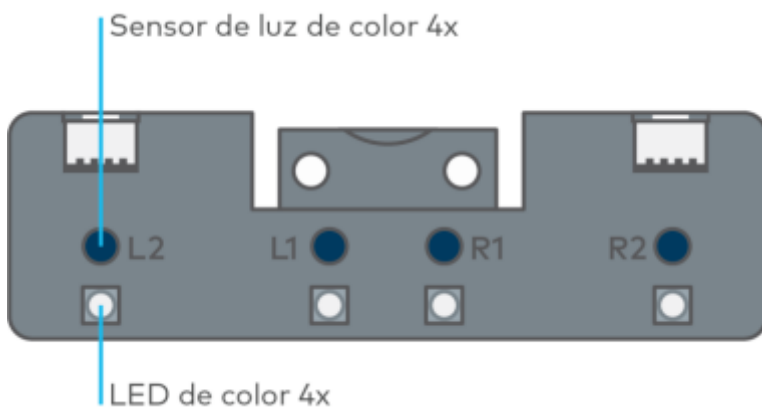


**Manual mBot2** <https://www.robotix.es/documentos/mbot2-actividades.pdf> contenidos e imágenes publicados con permiso de la empresa Robotix.

## SENSOR QUAD RGB

Es un sigue líneas más avanzado pues permite detectar cruces de 90º pues tiene 4 sensores frente a los 2 que tenía mBot1

Además tiene un sensor de color lo que permite conocer en qué posición de la línea estamos.



**Manual mBot2** <https://www.robotix.es/documentos/mbot2-actividades.pdf> contenidos e imágenes publicados con permiso de la empresa Robotix.

Para que el sensor interprete los colores que pertenecen a la línea mirar el vídeo :

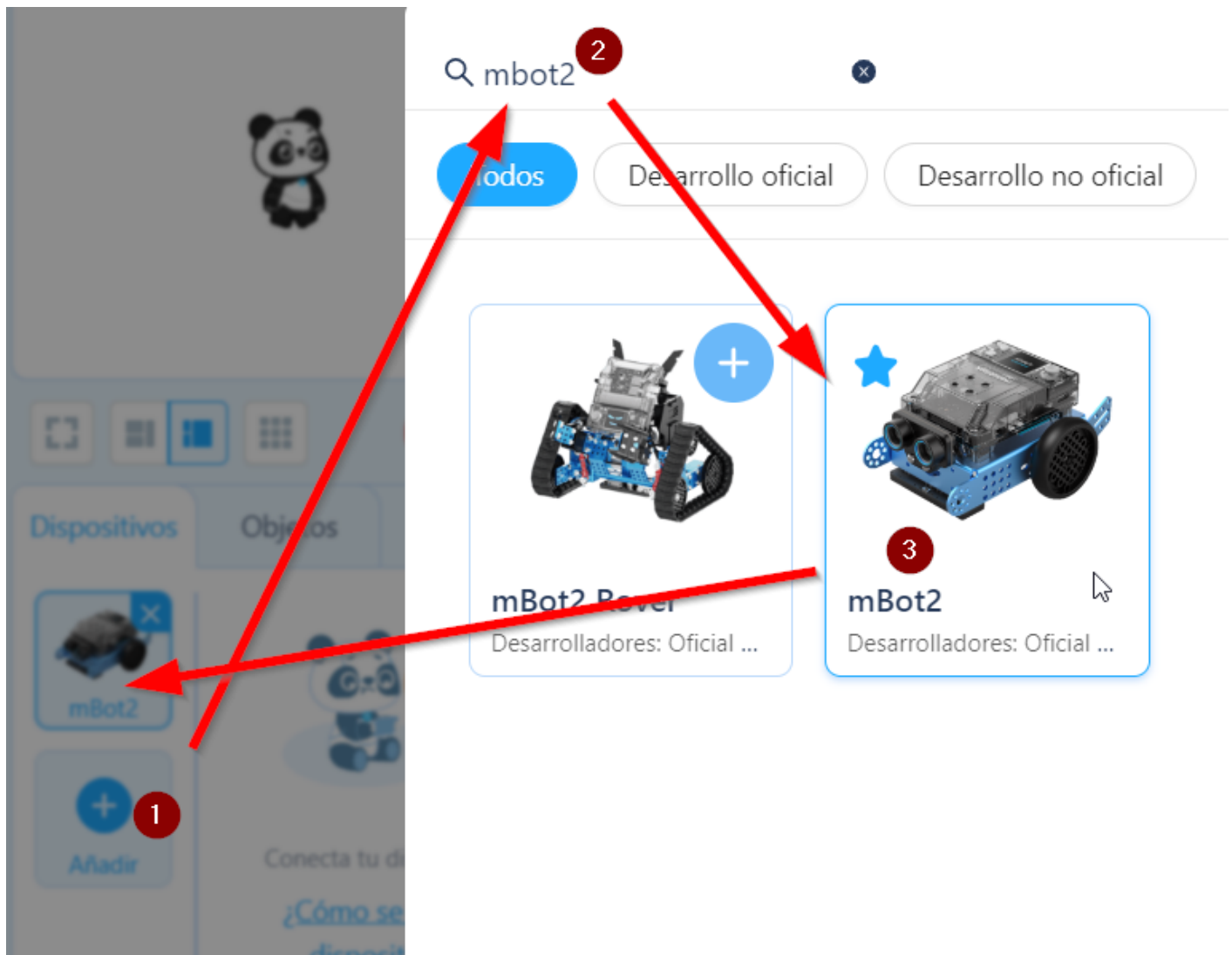


<https://www.youtube.com/embed/R7WgkpD2tvY>

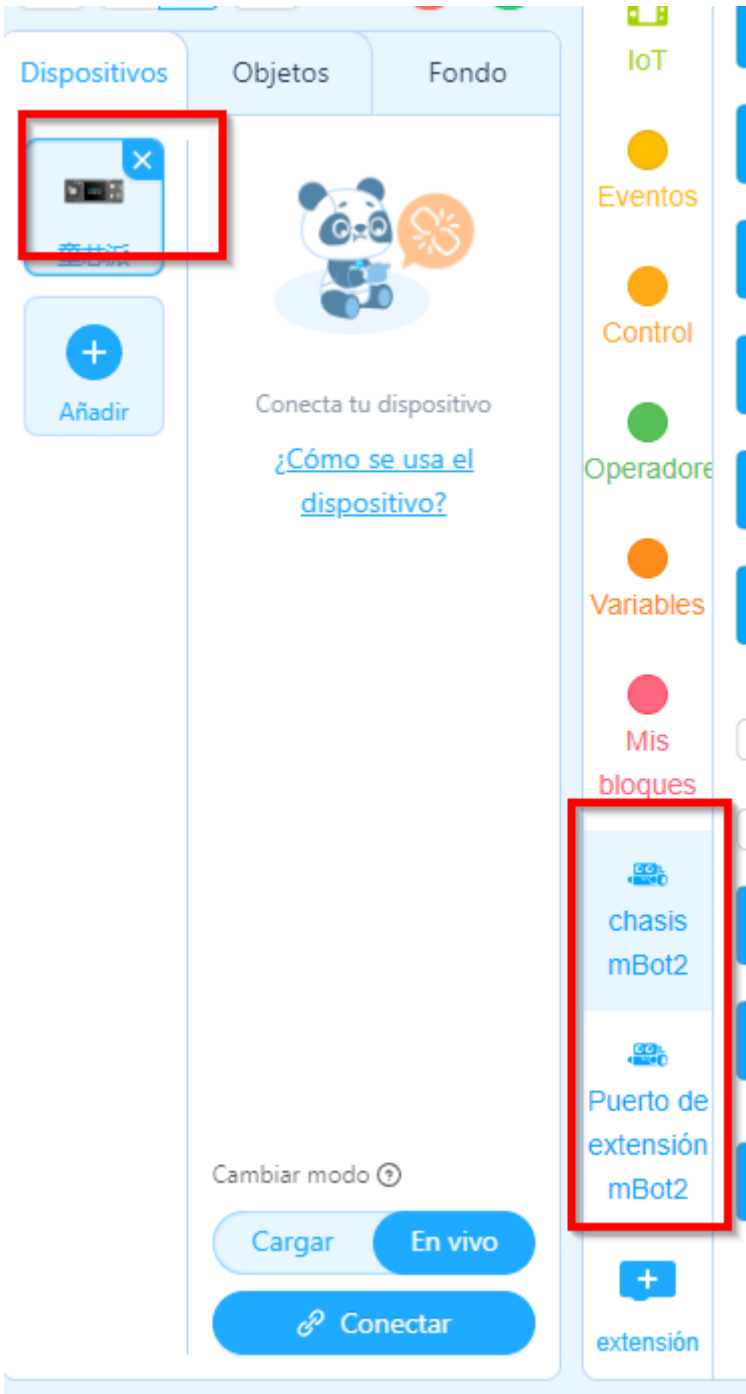
# Cómo usar mBot2 en mBlock

Podemos instalar directamente mBot2, lo interpreta como un chasis de Cyberpi

1. Vamos a extensión luego vamos a buscarlo en **extensión**
2. Buscamos mBot2 descargar
3. Añadir



Se instarán estos conjuntos de instrucciones :



The screenshot displays the 'Dispositivos' (Devices) tab of the CATEDU interface. At the top, there are three tabs: 'Dispositivos', 'Objetos', and 'Fondo'. Below the 'Dispositivos' tab, there is a red-bordered box containing a small window with a close button. To the left of the main content area is a blue button with a plus sign and the text 'Añadir'. The main content area features a panda illustration and the text 'Conecta tu dispositivo' followed by a link: '¿Cómo se usa el dispositivo?'. At the bottom of this area are buttons for 'Cargar', 'En vivo', and 'Conectar'. On the right side, a vertical menu lists various categories: IoT, Eventos, Control, Operadores, Variables, and Mis bloques. Under 'Mis bloques', there are two items highlighted with a red border: 'chasis mBot2' and 'Puerto de extensión mBot2'. Below these items is a button with a plus sign and the text 'extensión'.

# Mis primeros pasos

## Control remoto con el teclado

Vamos a realizar **EN VIVO** este programa

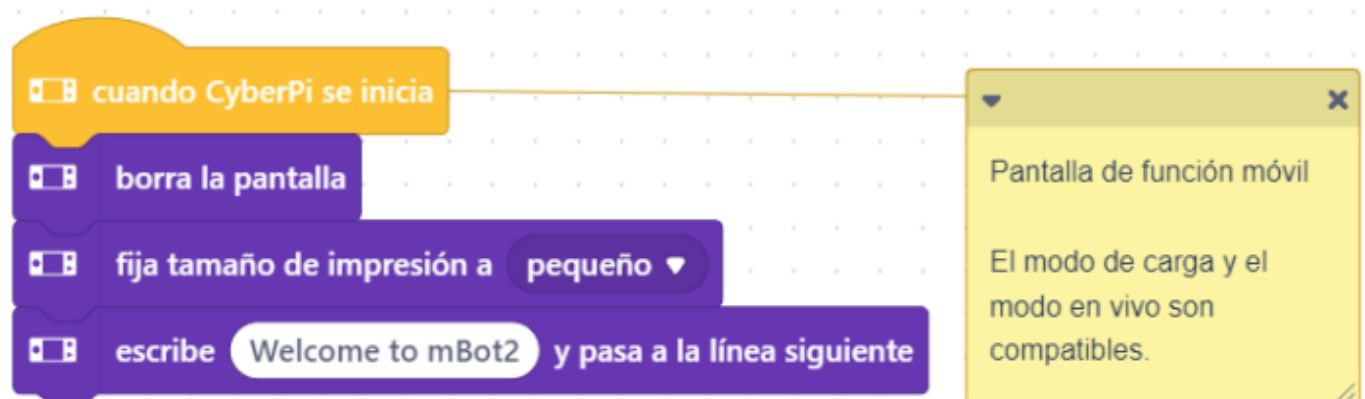


Como puedes ver es puedes controlar el robot pero esta "atado" al ordenador, esa es la desventaja de programar EN VIVO

## Programa de bienvenida

El contenido a partir de aquí es de esta página es de

Vamos a realizar este programa, primero una pantalla de bienvenida:



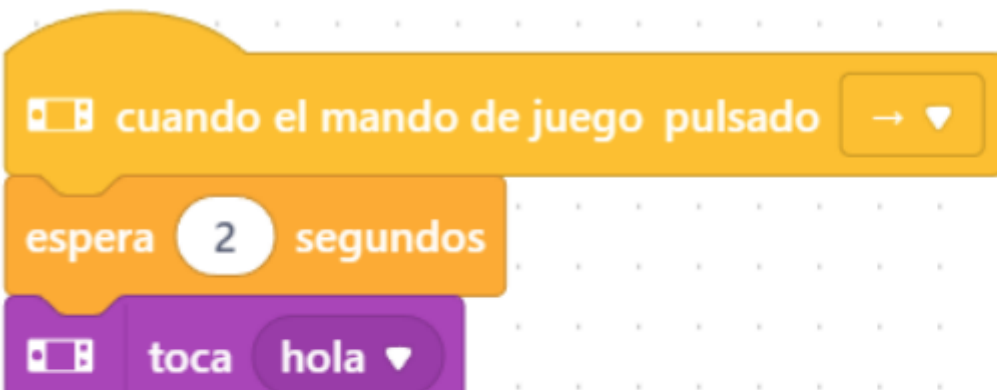
Luego interactuamos con el joystick y el movimiento:



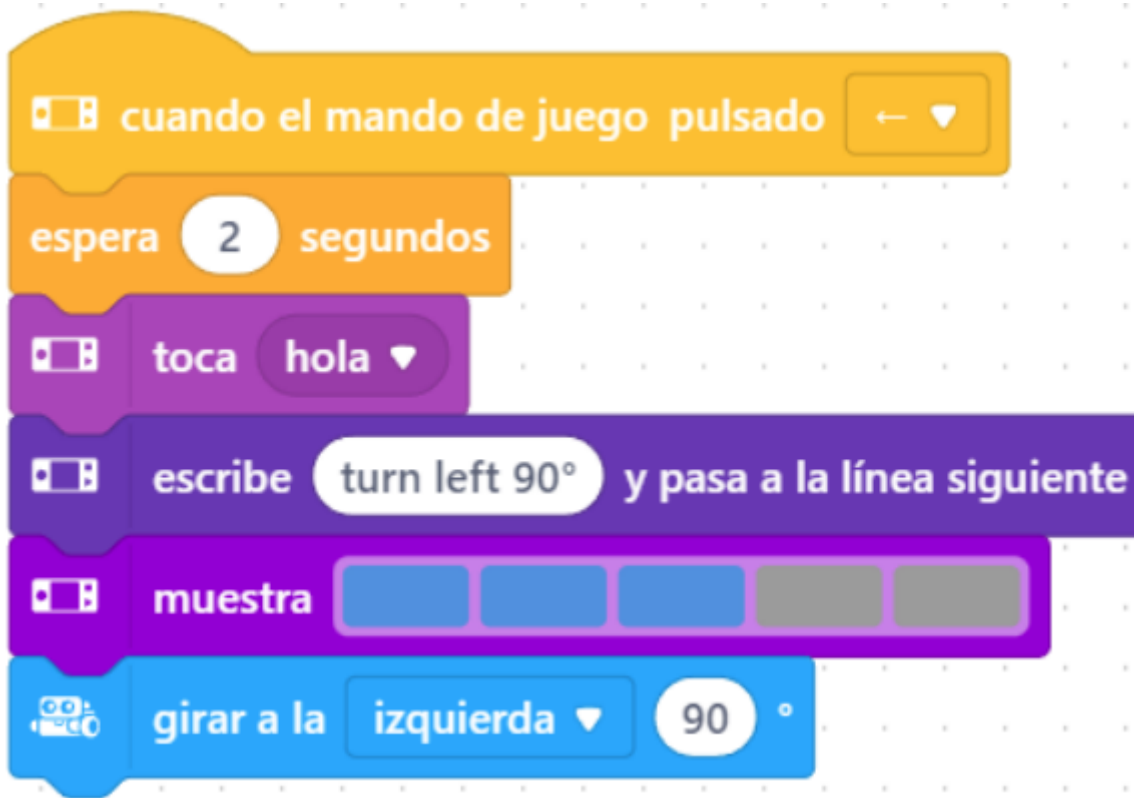
```
when gamepad button pressed (up)
  wait 2 seconds
  play sound hola
  write "move forward 10cm" and go to next line
  LED all on green
  cm forward 10
```



```
when gamepad button pressed (down)
  wait 2 seconds
  play sound hola
  write "move backward 10cm" and go to next line
  LED all on red
  cm backward 10
```



```
when gamepad button pressed (right)
  wait 2 seconds
  play sound hola
```



Cargamos estos programas **EN MODO CARGA**

## RESULTADO

<https://www.youtube.com/embed/LwjqWchUI0>

El contenido de esta página es de

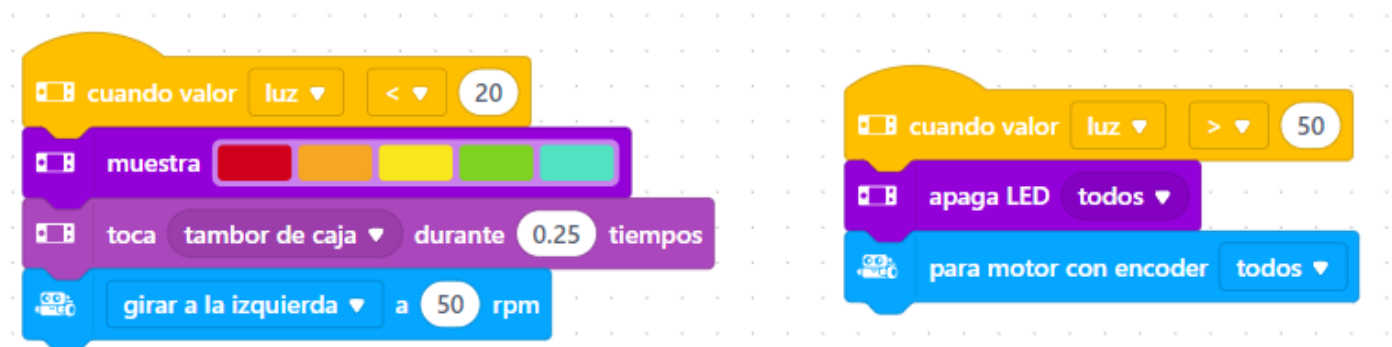
# Jugando con la luz

## mBot2 se quiere ir de marcha nocturna

Podemos hacer que el mBot2 se vaya de marcha nocturna

<https://www.youtube.com/embed/p9543Fjx4sM>

Un programa muy muy sencillo es este **hay que jugar con los valores de 20 y 50 según las condiciones de tu aula**



<https://www.youtube.com/embed/M3gOvOOQ-NA>

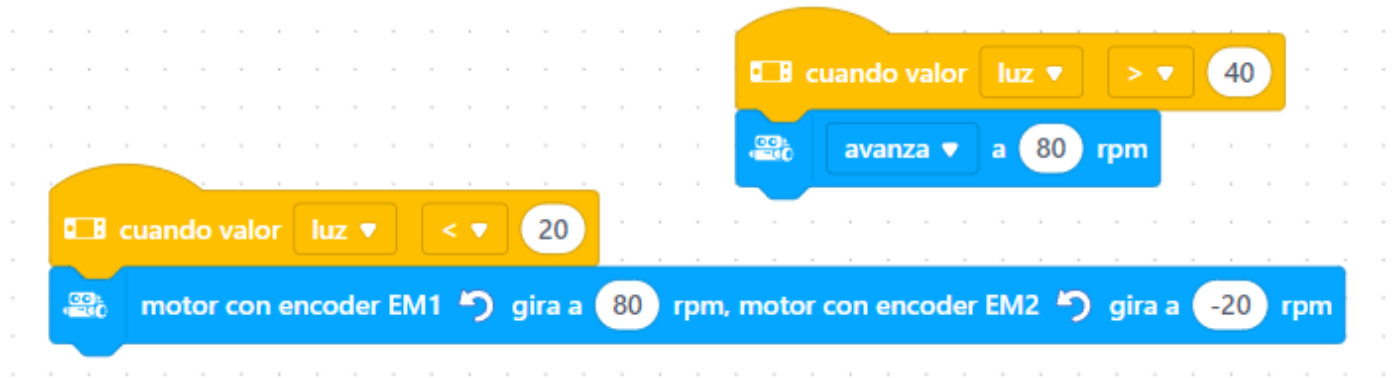
Es muy espectacular hacerlo con los alumnos, se deja el aula a oscuras, se encienden todas las



No te quedes con sólo girar, hazle piruetas, juegos de luces etc...

## Robot polilla

Podemos hacer este sencillo programa que avanza si hay luz, o va dando vueltas (lo busca) si no hay



<https://www.youtube.com/embed/8rucCGgt1gs>

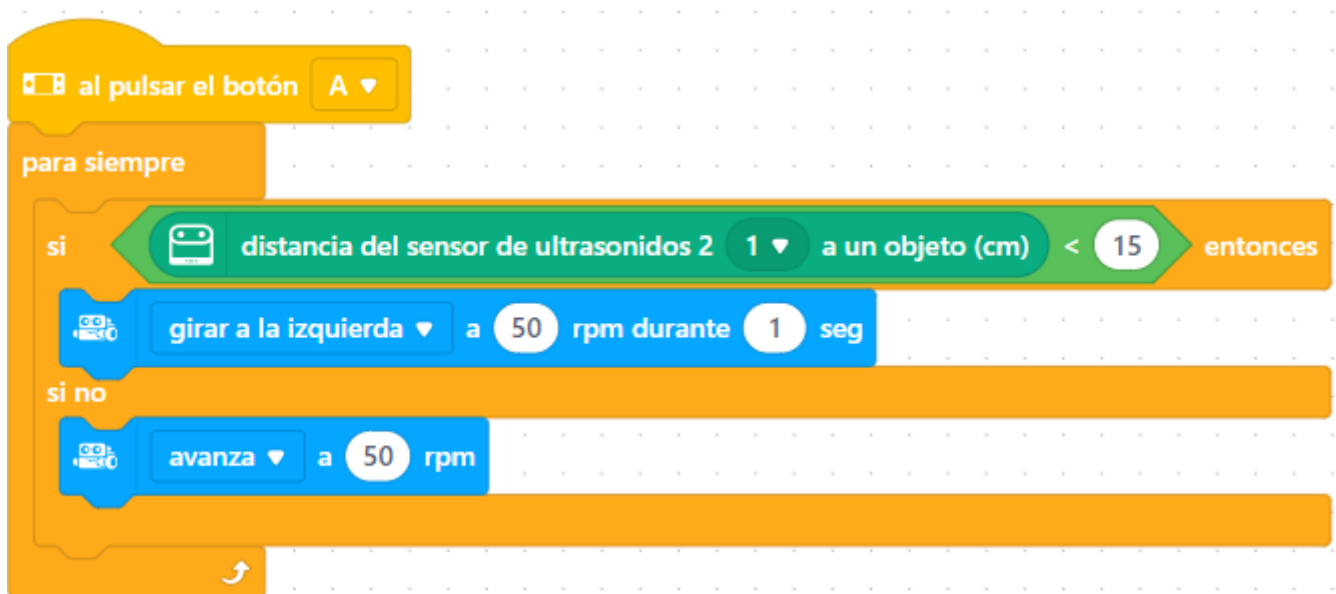
Puedes alterar el programa de tal manera que si no hay luz, que haga giros más largos, retroceda...

# Evitar obstáculos I

## Mi primer "Evita obstáculos"

Sugerimos un programa sencillo

<https://planet.mblock.cc/project/7548708>



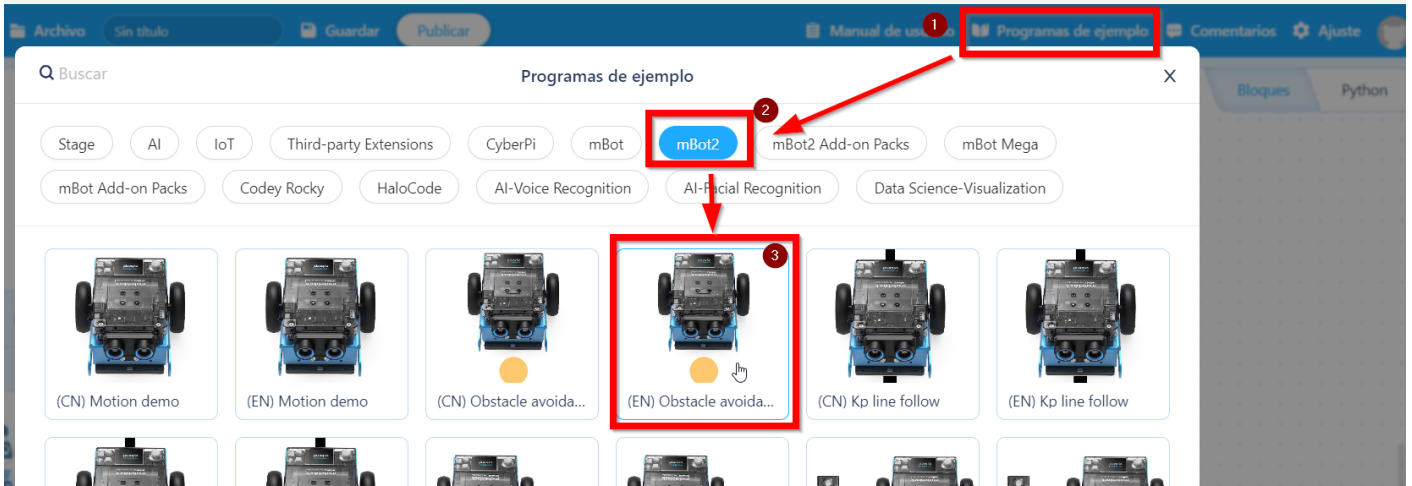
Como puedes ver, cuando hay un obstáculo a menos de 15cm (puedes modificar este número) retrocede un poco, gira y así evita el obstáculo. Si no hay objeto, pues avanza

<https://www.youtube.com/embed/r7fsP2kA9xY>

## El Evita obstáculos de programas de ejemplo

Acuérdate que para manejar el sensor de distancia, hay que instalar la extensión correspondiente. Ver <https://libros.catedu.es/books/cyberpi-y-mbot2/page/como-usar-mbot2-en-mblock>

El script que proponemos es el que se encuentra en "Programas de ejemplo ":



El contenido a partir de aquí es de



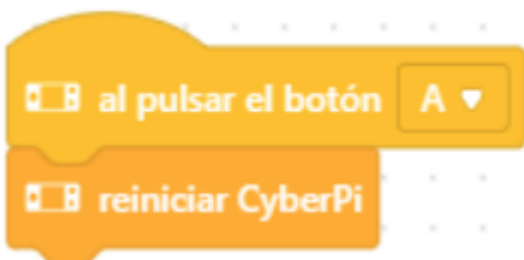
cuando CyberPi se inicia  
 fija tamaño de impresión a pequeño  
 escribe Press "B" to start obstacle avoidance y pasa a la línea siguiente  
 escribe Press "A" to restart y pasa a la línea siguiente  
 espera hasta que ¿pulsado botón B ?  
 para siempre  
 si distancia del sensor de ultrasonidos 2 1 a un objeto (cm) > 8 entonces  
 avanza a 100 rpm  
 si no  
 si número al azar entre 1 y 2 = 1 entonces  
 girar a la izquierda a 50 rpm  
 espera 1 segundos  
 si no  
 girar a la derecha a 50 rpm  
 espera 1 segundos

La aleatoriedad está diseñada para reducir la posibilidad de que el automóvil se atasque en un cierto ángulo.

No hacerlo al azar también puede lograr un buen efecto de evitación de obstáculos.

La distancia de 8cm es arbitraria y también la elección de hacerlo al azar o no.

Podemos añadir



al pulsar el botón A  
 reiniciar CyberPi



## RESULTADO

<https://www.youtube.com/embed/4bVuxd22klc>

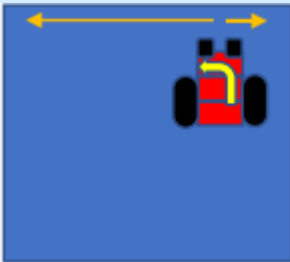
El contenido de esta página es de

# Evitar obstáculos II






El contenido de esta página es de

Acuérdate que para manejar el sensor de distancia, hay que instalar la extensión correspondiente. Ver <https://libros.catedu.es/books/cyberpi-y-mbot2/page/como-usar-mbot2-en-mblock>

Ahora, en vez de elegir al azar como Evitar obstáculos I qué dirección a girar, en este caso vamos a realizar un proyecto en el cual nuestro mBot2 tiene que detectar un objeto cuando esté a menos de 10 cm. En este momento se tiene que para y comprobar la distancia que hay a su derecha y la distancia que hay a su izquierda, siempre tendrá que ir por la dirección donde el obstáculo está más lejos.





```
cuando clic en 
para siempre
  avanza a 50 rpm
  espera hasta que  distancia del sensor de ultrasonidos 2 1 a un objeto (cm) < 10
  para motor con encoder todos
  girar a la izquierda 90 °
  fija izquierda a  distancia del sensor de ultrasonidos 2 1 a un objeto (cm)
  girar a la derecha 180 °
  fija Derecha a  distancia del sensor de ultrasonidos 2 1 a un objeto (cm)
  si  Derecha > izquierda entonces
    girar a la izquierda 0 °
  si no
    girar a la derecha 180 °
```

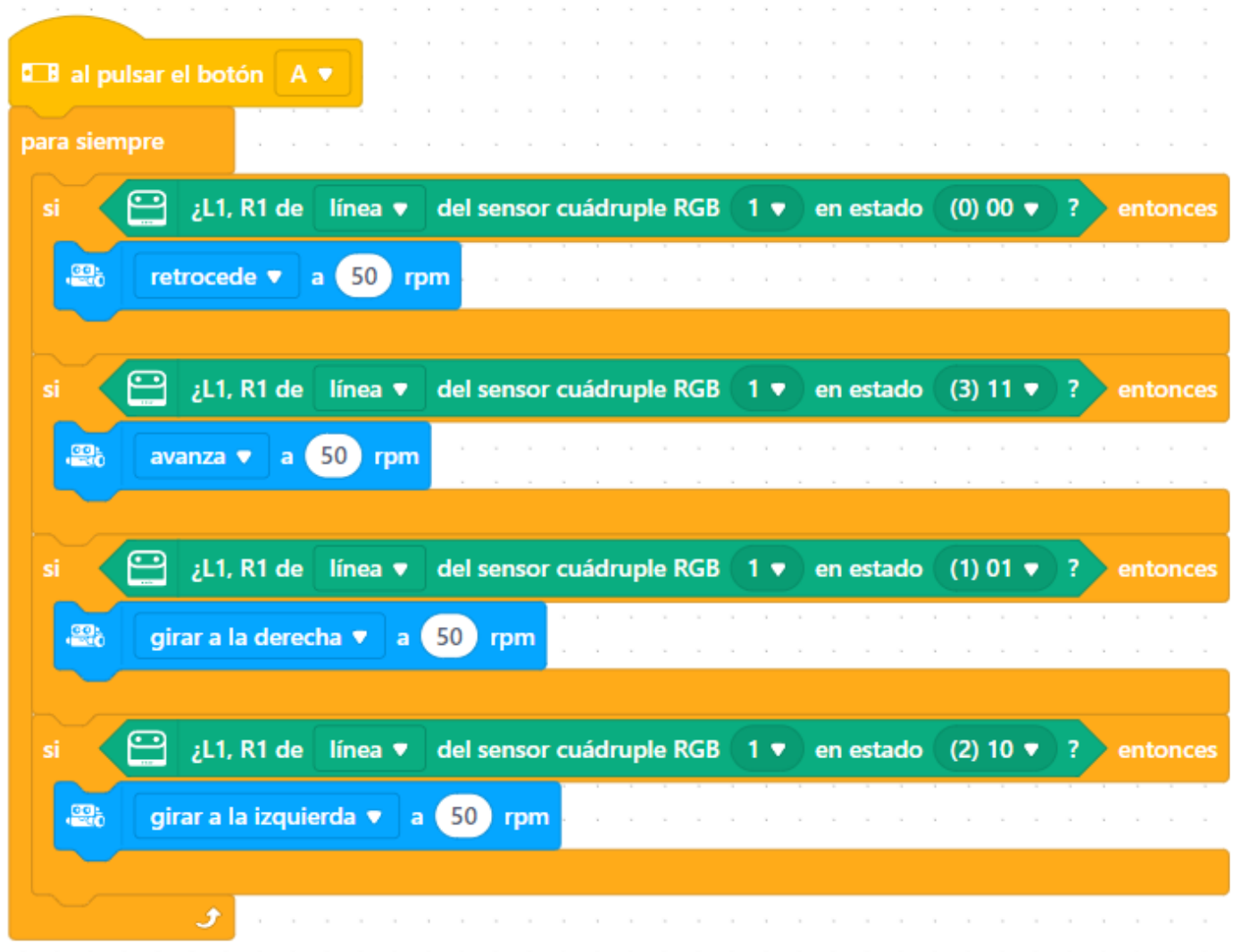
## RESULTADO

[https://www.youtube.com/embed/m4GC\\_5WbdXM](https://www.youtube.com/embed/m4GC_5WbdXM)

# Sigue líneas I

## Mi primer siguelíneas

El sigue líneas más sencillo es este, recomendamos empezar por aquí



- Si detecto 00 es decir "he perdido la línea" entonces ves hacia atrás para recuperarla
- Si detecto 11 es que estoy "en la línea" pues "pa'lante"
- Si detecto 01 es que la línea la tengo a mi derecha (0 en la izquierda 1 en la derecha) por lo tanto gira a la derecha para volver al redil
- lo mismo pero a la izquierda

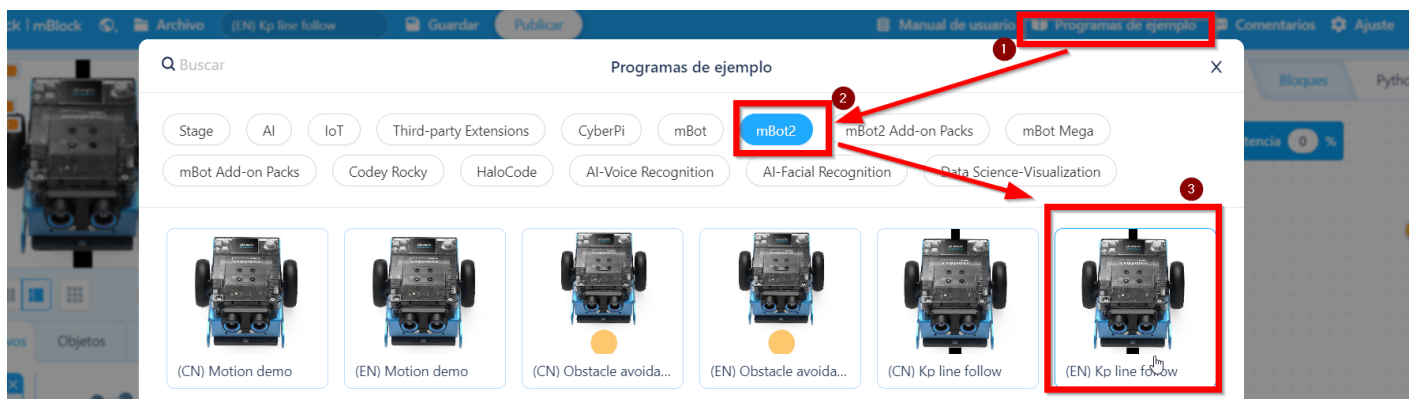
Como se puede ver, hace movimientos hacia atrás sobre todo en las curvas cerradas pero **¡¡no se escapa!!**

<https://www.youtube.com/embed/6itHWcvZnUs>

## El sígue líneas de "Programas de ejemplo"

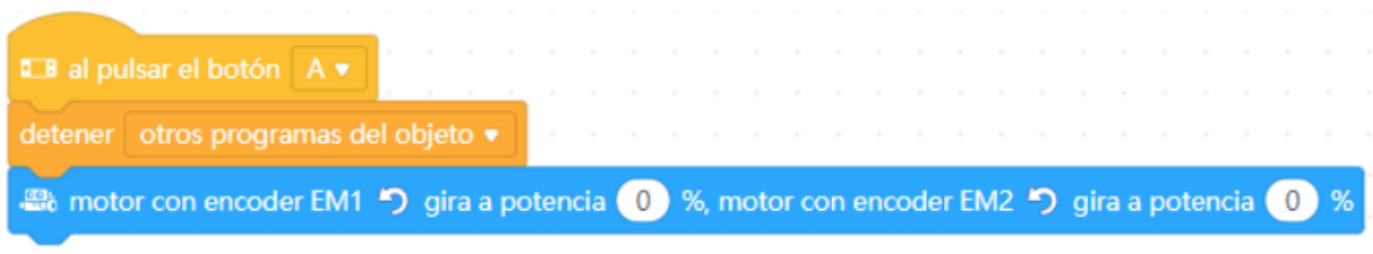
Acuérdate que para manejar el sensor de distancia, hay que instalar la extensión correspondiente. Ver <https://libros.catedu.es/books/cyberpi-y-mbot2/page/como-usar-mbot2-en-mblock>

El programa lo puedes encontrar en los "Programas de ejemplo":



El contenido a partir de aquí es esta página es de

El primer script es para parar el robot con el botón A



Vamos a crear las siguientes variables:

base\_power

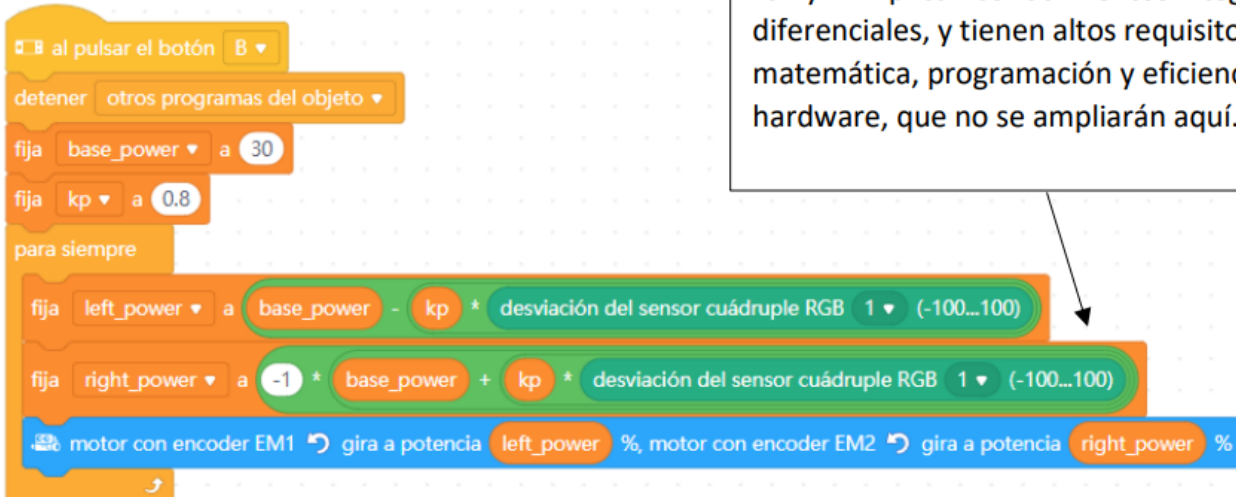
kp

left\_power

right\_power

Como "cantidad de desviación", la posición de desplazamiento de línea se puede utilizar para corregir la desviación de las tareas automáticas de seguimiento de línea. Es la "P" en lo que llamamos control PID.

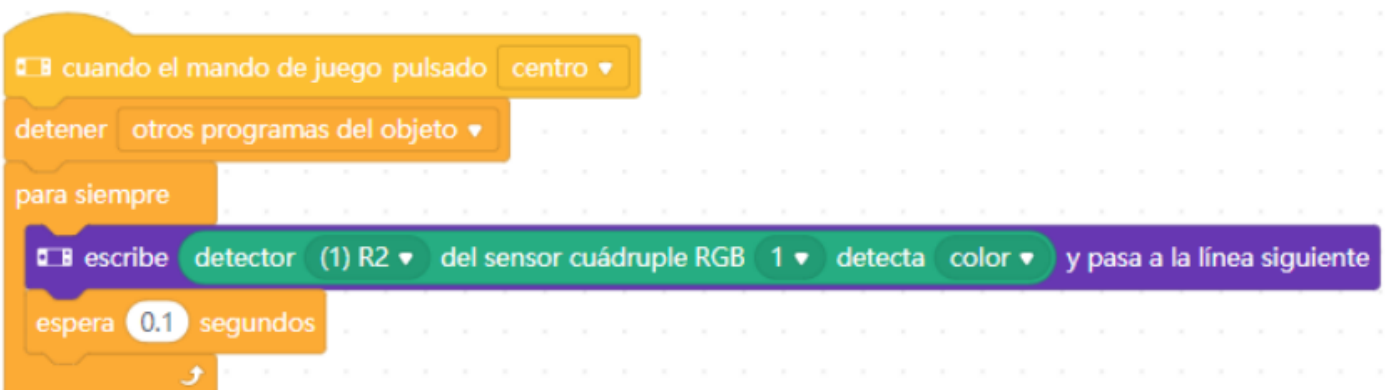
La I y D implican conocimientos integrales y diferenciales, y tienen altos requisitos de matemática, programación y eficiencia de hardware, que no se ampliarán aquí.



```

al pulsar el botón B
  detener otros programas del objeto
  fija base_power a 30
  fija kp a 0.8
  para siempre
    fija left_power a base_power - kp * desviación del sensor cuádruple RGB 1 (-100...100)
    fija right_power a -1 * base_power + kp * desviación del sensor cuádruple RGB 1 (-100...100)
    motor con encoder EM1 gira a potencia left_power %, motor con encoder EM2 gira a potencia right_power %
  
```

(optativo) Esta función para que nos muestre el color de la línea



```

cuando el mando de juego pulsado centro
  detener otros programas del objeto
  para siempre
    escribe detector (1) R2 del sensor cuádruple RGB 1 detecta color y pasa a la línea siguiente
    espera 0.1 segundos
  
```

## RESULTADO

<https://www.youtube.com/embed/hhTppcf9iPA>

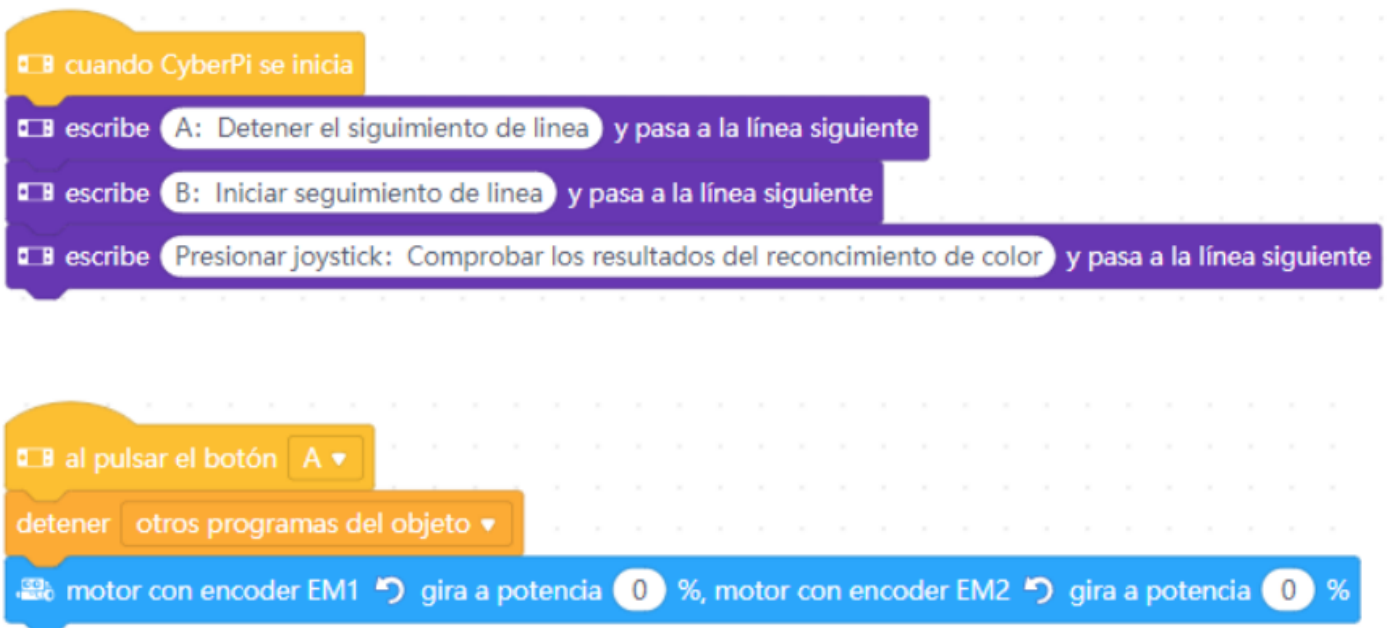
# Sigue líneas II

El contenido de esta página es de

Acuérdate que para manejar el sensor de distancia, hay que instalar la extensión correspondiente. Ver <https://libros.catedu.es/books/cyberpi-y-mbot2/page/como-usar-mbot2-en-mblock>

En este programa sigue líneas avanzado vamos a incorporar que nos vaya diciendo los colores que va detectando

El script que proponemos es el siguiente :



The image shows two Scratch scripts on a grid background. The first script starts with a yellow 'when CyberPi starts' block, followed by three purple 'write' blocks: 'A: Detener el seguimiento de línea y pasa a la línea siguiente', 'B: Iniciar seguimiento de línea y pasa a la línea siguiente', and 'Presionar joystick: Comprobar los resultados del reconocimiento de color y pasa a la línea siguiente'. The second script starts with a yellow 'when button A is pressed' block, followed by an orange 'stop other programs of the object' block, and a blue 'motor with encoder EM1' block that rotates at 0% power.

Y el algoritmo principal

- Cuando se presiona el botón B
  - Detener otros programas del objeto.
  - A la variable base\_power le asignamos el valor 55

- A la variable `kp` le asignamos el valor 0.8
- Para siempre (Bucle infinito)
  - A la variable `left_power` le asignamos el valor  $-1 * (base\_power + (kp * desviación\ del\ sensor\ cuádruple\ RGB))$
  - A la variable `right_power` le asignamos el valor  $(base\_power - (kp * desviación\ del\ sensor\ cuádruple\ RGB))$
  - moto EM1 gira a potencia `right_power %`, motor EM2 gira potencia `left_power %`
  - sí detector R1 del sensor cuádruple RGB detecta blanco entonces
    - muestra 5 leds color blanco
  - sí detector R1 del sensor cuádruple RGB detecta rojo entonces
    - muestra 5 leds color rojo
  - sí detector R1 del sensor cuádruple RGB detecta amarillo entonces
    - muestra 5 leds color amarillo
  - sí detector R1 del sensor cuádruple RGB detecta verde entonces
    - muestra 5 leds color verde
  - sí detector R1 del sensor cuádruple RGB detecta cian entonces
    - muestra 5 leds color cian
  - sí detector R1 del sensor cuádruple RGB detecta azul entonces
    - muestra 5 leds color azul
  - sí detector R1 del sensor cuádruple RGB detecta morado entonces
    - muestra 5 leds color morado
  - sí detector R1 del sensor cuádruple RGB detecta negro entonces
    - muestra 5 leds color negro

```

al pulsar el botón B
  detener otros programas del objeto
  fija base_power a 55
  fija kp a 0.8
  para siempre
    fija left_power a (-1 * base_power + kp * desviación del sensor cuádruple RGB 1 (-100...100))
    fija right_power a base_power - kp * desviación del sensor cuádruple RGB 1 (-100...100)
    motor con encoder EM1 gira a potencia right_power %, motor con encoder EM2 gira a potencia left_power %
    si ¿detector (2) R1 del sensor cuádruple RGB 1 detecta blanco ? entonces
      muestra [blanco]
    si ¿detector (2) R1 del sensor cuádruple RGB 1 detecta rojo ? entonces
      muestra [rojo]
    si ¿detector (2) R1 del sensor cuádruple RGB 1 detecta amarillo ? entonces
      muestra [amarillo]
    si ¿detector (2) R1 del sensor cuádruple RGB 1 detecta verde ? entonces
      muestra [verde]
    si ¿detector (2) R1 del sensor cuádruple RGB 1 detecta cian ? entonces
      muestra [cian]
    si ¿detector (2) R1 del sensor cuádruple RGB 1 detecta azul ? entonces
      muestra [azul]
    si ¿detector (2) R1 del sensor cuádruple RGB 1 detecta morado ? entonces
      muestra [morado]
    si ¿detector (2) R1 del sensor cuádruple RGB 1 detecta negro ? entonces
      muestra [negro]
  
```



Y además (optativo) igual que el sigue líneas I :

El color de reconocimiento está configurado para aparecer después de la clave porque se necesita una cierta cantidad de tiempo para imprimir el texto, lo que bloqueará el hilo actual y hará que el efecto de patrullaje disminuya.

Puede haber algún error en el reconocimiento del color, porque el material de la tarjeta de color basado en la calibración de fábrica no cubre todos los materiales.

Si necesita una determinación de color personalizada, puede considerar la implementación del parámetro "valor del objeto R / G / B" devuelto.

## RESULTADO

<https://www.youtube.com/embed/Nx1mIV6vddk>

# Control remoto

El contenido de esta página es de **Ejemplos mBot2** Pere Manel Verdugo Zamora <http://peremanelv.com/mblok.html> contenidos e imágenes publicados con permiso del autor

PARA REALIZAR ESTE PROGRAMA SE NECESITAN DOS CYBERPIS QUE VAN A ESTAR CONECTADOS EN LA MISMA LAN

Ahora un Cyberpi con su joystick controlará los movimientos del otro Cyberpi montado en su mBot2



**Ejemplos mBot2** Pere Manel Verdugo Zamora <http://peremanelv.com/mblok.html> contenidos e imágenes publicados con permiso del autor

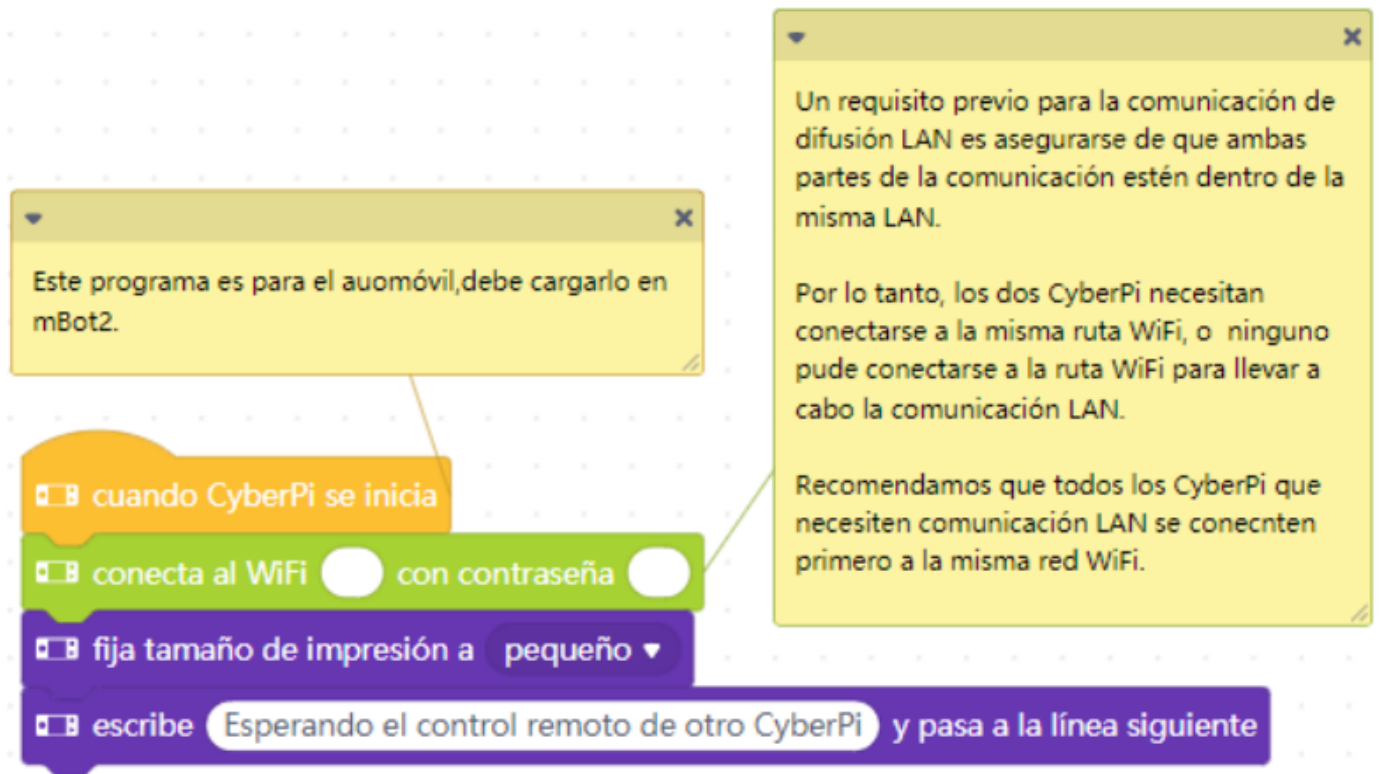
Para este proyecto hemos de cargar dos CyberPy.



## CODIGO EN EL CYBERPI RECEPTOR MONTADO CON MBOT2

<https://planet.mblock.cc/project/projectshare/7552114>

Hay que poner los dos Cyberpis en la misma Wifi (SSID y contraseña)



Este programa es para el auomóvil, debe cargarlo en mBot2.

cuando CyberPi se inicia

conecta al WiFi con contraseña

fija tamaño de impresión a pequeño

escribe Esperando el control remoto de otro CyberPi y pasa a la línea siguiente

Un requisito previo para la comunicación de difusión LAN es asegurarse de que ambas partes de la comunicación estén dentro de la misma LAN.

Por lo tanto, los dos CyberPi necesitan conectarse a la misma ruta WiFi, o ninguno puede conectarse a la ruta WiFi para llevar a cabo la comunicación LAN.

Recomendamos que todos los CyberPi que necesiten comunicación LAN se conecten primero a la misma red WiFi.

**Ejemplos mBot2** Pere Manel Verdugo Zamora <http://peremanelv.com/mblok.html> contenidos e imágenes publicados con permiso del autor



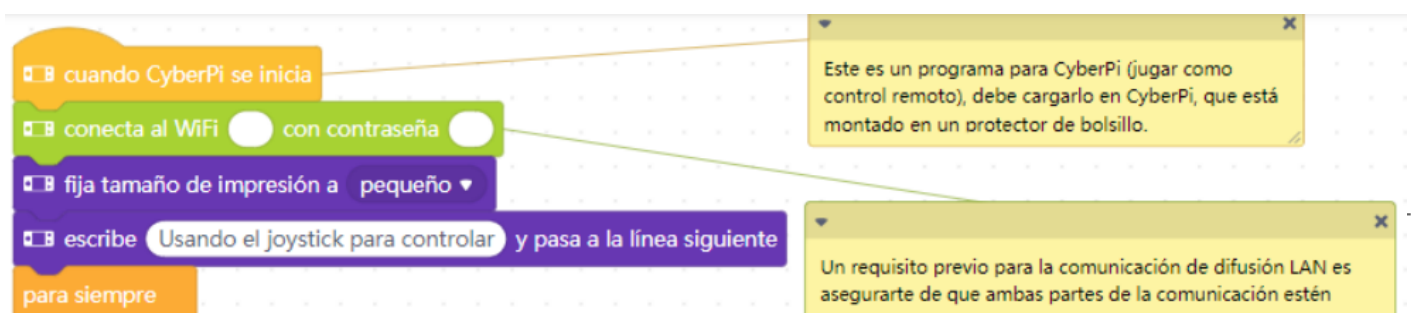
```

al recibir message en LAN
si valor recibido del mensaje de red message = 1 entonces
  avanza a 50 rpm
si no
  si valor recibido del mensaje de red message = 2 entonces
    retrocede a 50 rpm
  si no
    si valor recibido del mensaje de red message = 3 entonces
      girar a la izquierda a 50 rpm
    si no
      si valor recibido del mensaje de red message = 4 entonces
        girar a la derecha a 50 rpm
      si no
        para motor con encoder todos
  
```

**Ejemplos mBot2** Pere Manel Verdugo Zamora <http://peremanelv.com/mblok.html> contenidos e imágenes publicados con permiso del autor

## CODIGO EN EL CYBERPI QUE HARÁ DE CONTROL REMOTO

<https://planet.mblock.cc/project/7552213>



```

cuando CyberPi se inicia
  conecta al WiFi con contraseña
  fija tamaño de impresión a pequeño
  escribe Usando el joystick para controlar y pasa a la línea siguiente
  para siempre
  
```

Este es un programa para CyberPi (jugar como control remoto), debe cargarlo en CyberPi, que está montado en un protector de bolsillo.

Un requisito previo para la comunicación de difusión LAN es asegurarte de que ambas partes de la comunicación estén



*imágenes publicados con permiso del autor*

## RESULTADO

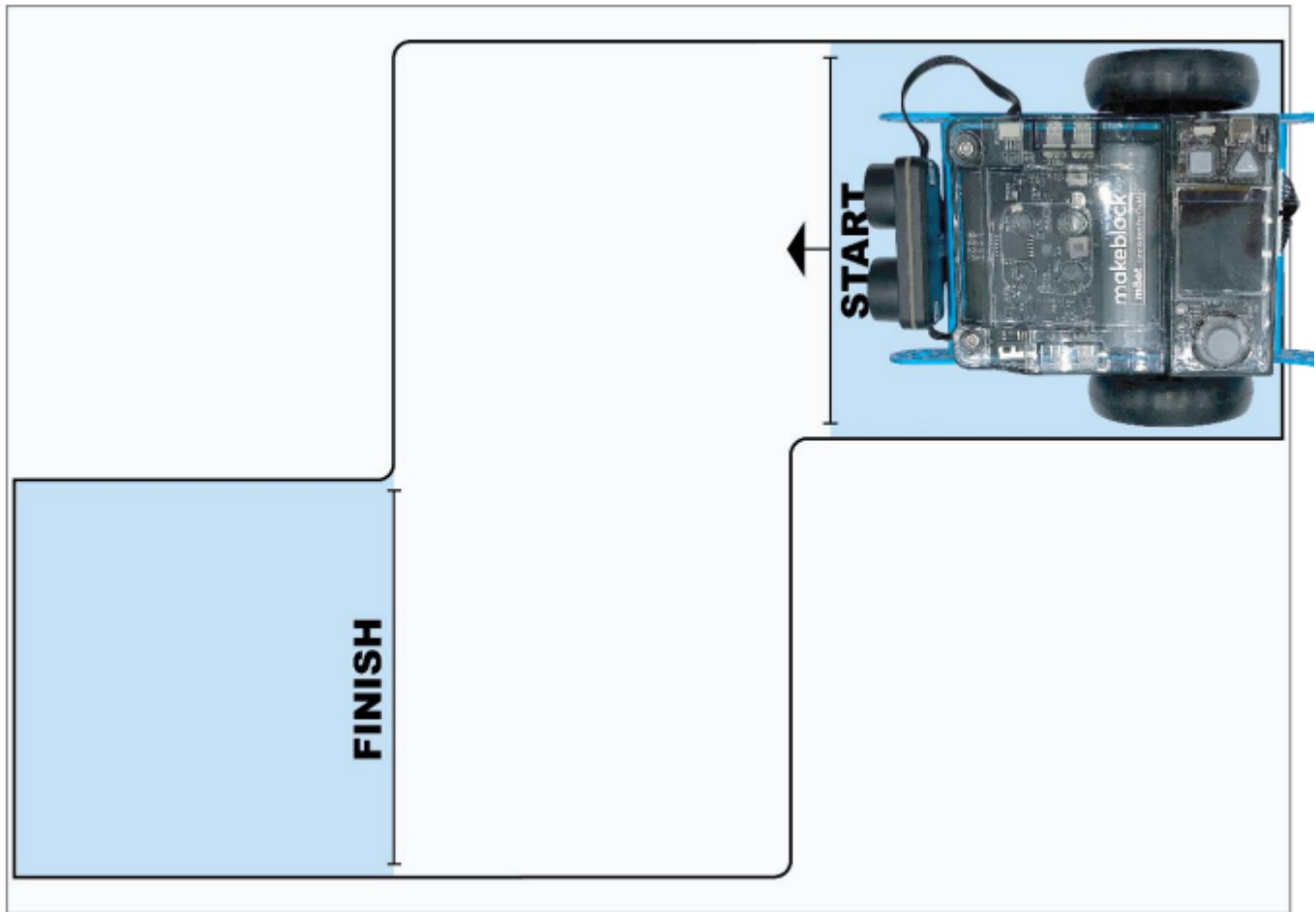
<https://www.youtube.com/embed/khrZ9OvGnY4>

El contenido de esta página es de

# Más actividades

## RETOS : HACER UN PROGRAMA QUE REALICE....

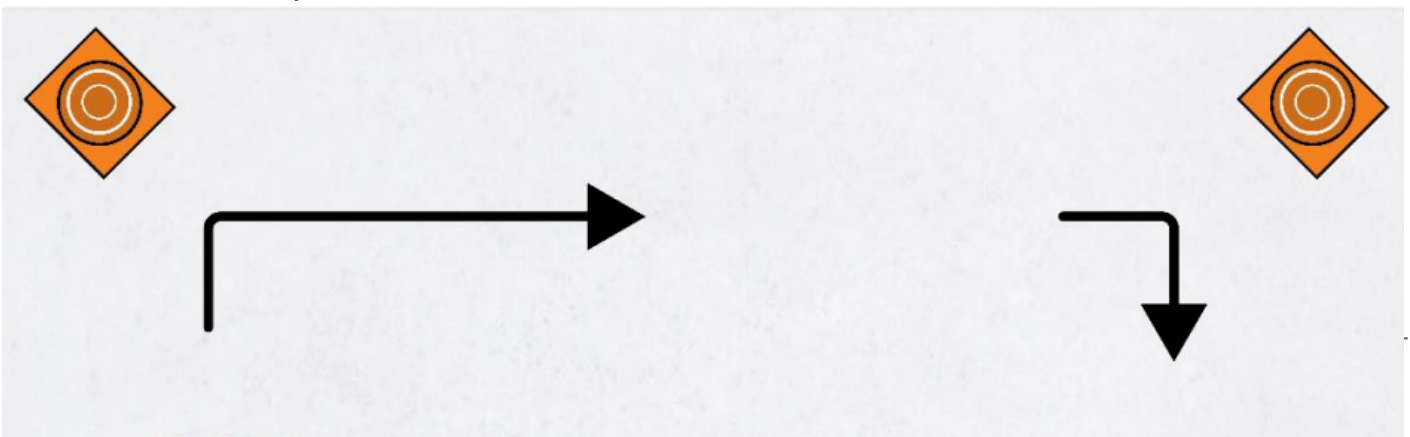
### RETO1 APARCAR



SOLUCIÓN pag 18 de **Manual mBot2** <https://www.robotix.es/documentos/mbot2-actividades.pdf>  
contenidos e imágenes publicados con permiso de la empresa Robotix.

### RETO 2 HACER UN CUADRADO

Los cuadrados naranjas son obstáculos.





Solución en pag 53 de **Manual mBot2** <https://www.robotix.es/documentos/mbot2-actividades.pdf> contenidos e imágenes publicados con permiso de la empresa Robotix.

### RETO 3 BUSCANDO EL COLOR AZUL ...

Un grupo de mBot2 buscan el color azul pintado en el suelo. Si uno lo encuentra, informará a los demás y ganará el juego.

Solución en pag 90 de **Manual mBot2** <https://www.robotix.es/documentos/mbot2-actividades.pdf> contenidos e imágenes publicados con permiso de la empresa Robotix.

### RETO 4 BUSCANDO EL COLOR AZUL... PERO EN SOLITARIO

Hacer el juego anterior pero **sólo teniendo un mbot2**

Solución en pag 9 de **Manual mBot2** <https://www.robotix.es/documentos/mbot2-actividades.pdf> contenidos e imágenes publicados con permiso de la empresa Robotix.

Es parecido a la solución con Broadcast que vimos en [LAN con cyberpi](#)

### RETO 5 EL NO VA MÁS ... UN CAMARERO

**El robot camarero que vas a fabricar debe ser capaz de hacer una serie de cosas:**

- Desplazarse en línea recta
- Detenerse en una zona roja (tu mesa), anotar un pedido y enviarlo a la cocina
- Opción adicional: si se dispone de un segundo mBot2 o CyberPi, entonces mostrar los pedidos a la cocina (para que puedan empezar a preparar la comida)
- Continuar desplazándose hasta que el mBot2 encuentre una superficie verde (la cocina) y se detenga para recoger la comida de la cocina
- Opcional: si se dispone de un segundo mBot2 o CyberPi, entonces notificar se está entregando la comida

```

when CyberPi starts up
  connect to Wi-Fi ssid password password
  display [red, orange, yellow, green, blue]
  wait until [network connected?]
  display [green, green, green, green, green]
  show [Waiterbot] at center of screen by middle pixel
  wait 1 seconds
  clear screen

when receiving food broadcast on LAN
  show [LAN broadcast food value received] at center of screen by middle pixel
  wait 1 seconds
  clear screen

when receiving food in delivery broadcast on LAN
  play LED animation rainbow until done
  play yummy until done
  turn off LED all

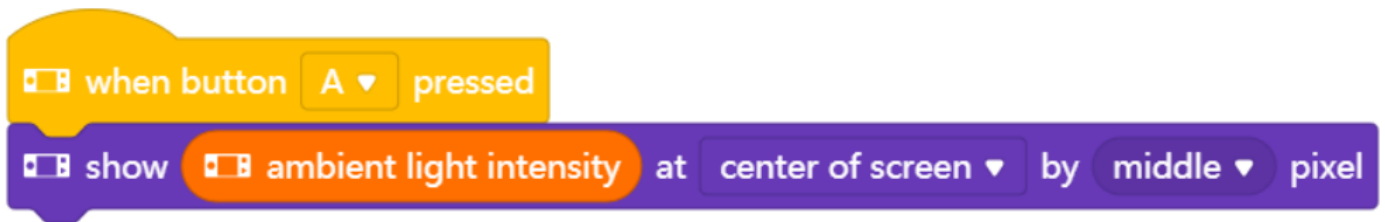
when joystick middle pressed
  clear screen
  moves forward at 50 RPM
  wait until [quad rgb sensor 1 probe (3) L1 detected red ? or quad rgb sensor 1 probe (2) R1 detected red ?]
  stop encoder motor all
  get_order
  moves forward at 50 RPM
  
```

```

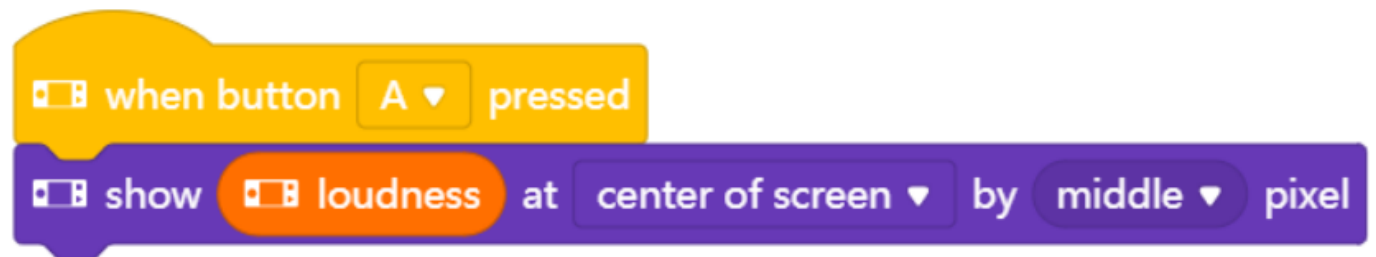
define get_order
  set order_status to 0
  LED all displays green
  speak auto Dear Guest! Your orders please?
  repeat until order_status > 0
    LED all displays red
    recognize (4) English 2 secs
    turn off LED all
    set order to speech recognition result
    show order at center of screen by middle pixel
    LED all displays green
    speak auto join Did you say order
    LED all displays red
    recognize (4) English 1 secs
    turn off LED all
    show speech recognition result at center of screen by middle pixel
    if speech recognition result contains yes ? then
      set order_status to 1
      speak auto Your meal will be prepared immediately.
      broadcast message food with value order on LAN
    else
      speak auto Sorry. Please repeat your order
  
```

Pag 107 **Manual mBot2** <https://www.robotix.es/documentos/mbot2-actividades.pdf> contenidos e imágenes publicados con permiso de la empresa Robotix.

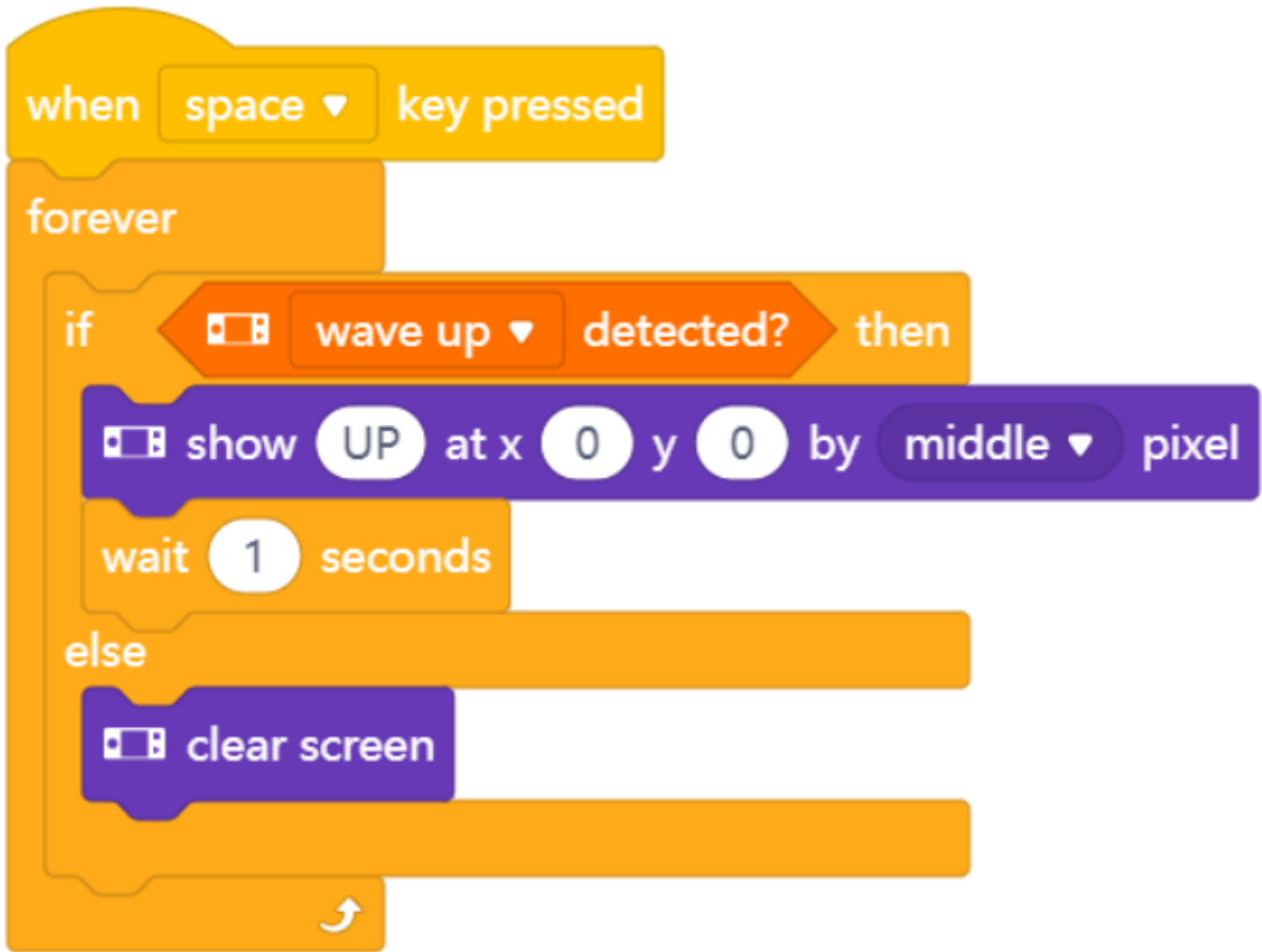
## ¿QUE HACEN ESTOS PROGRAMAS? ....



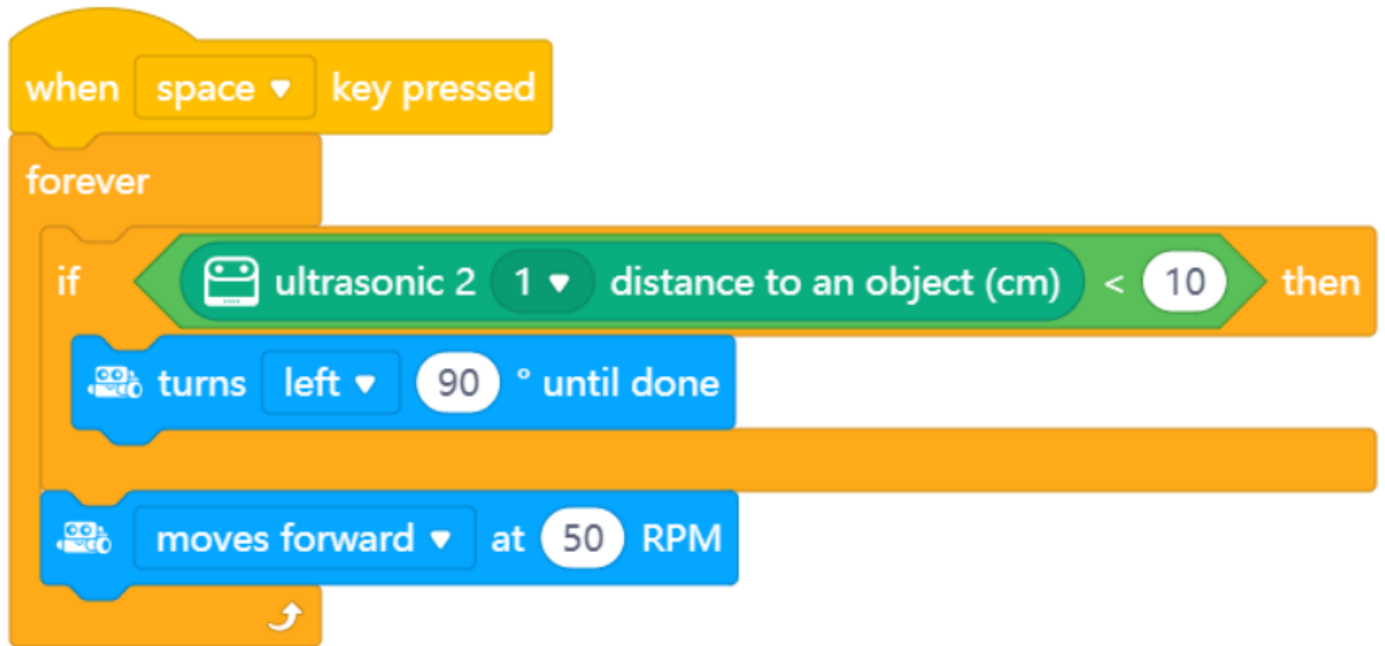
Pag 25 **Manual mBot2** <https://www.robotix.es/documentos/mbot2-actividades.pdf> contenidos e imágenes publicados con permiso de la empresa Robotix.



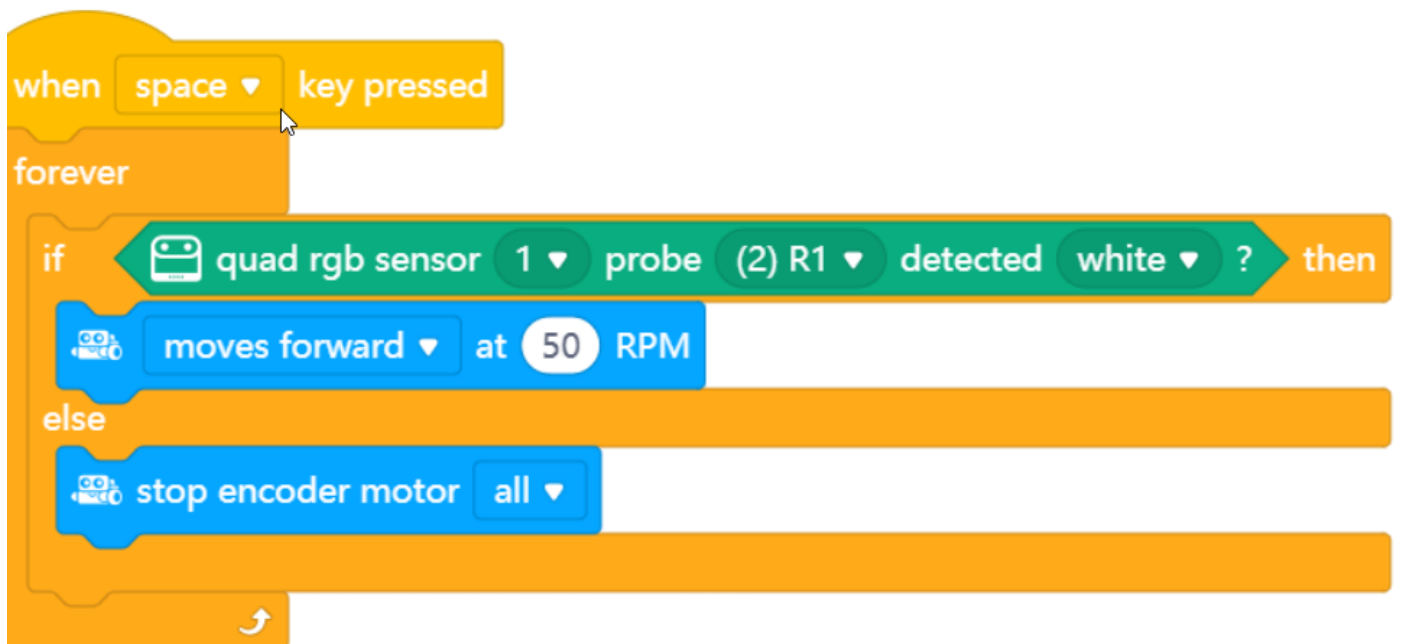
Pag 26 **Manual mBot2** <https://www.robotix.es/documentos/mbot2-actividades.pdf> contenidos e imágenes publicados con permiso de la empresa Robotix.



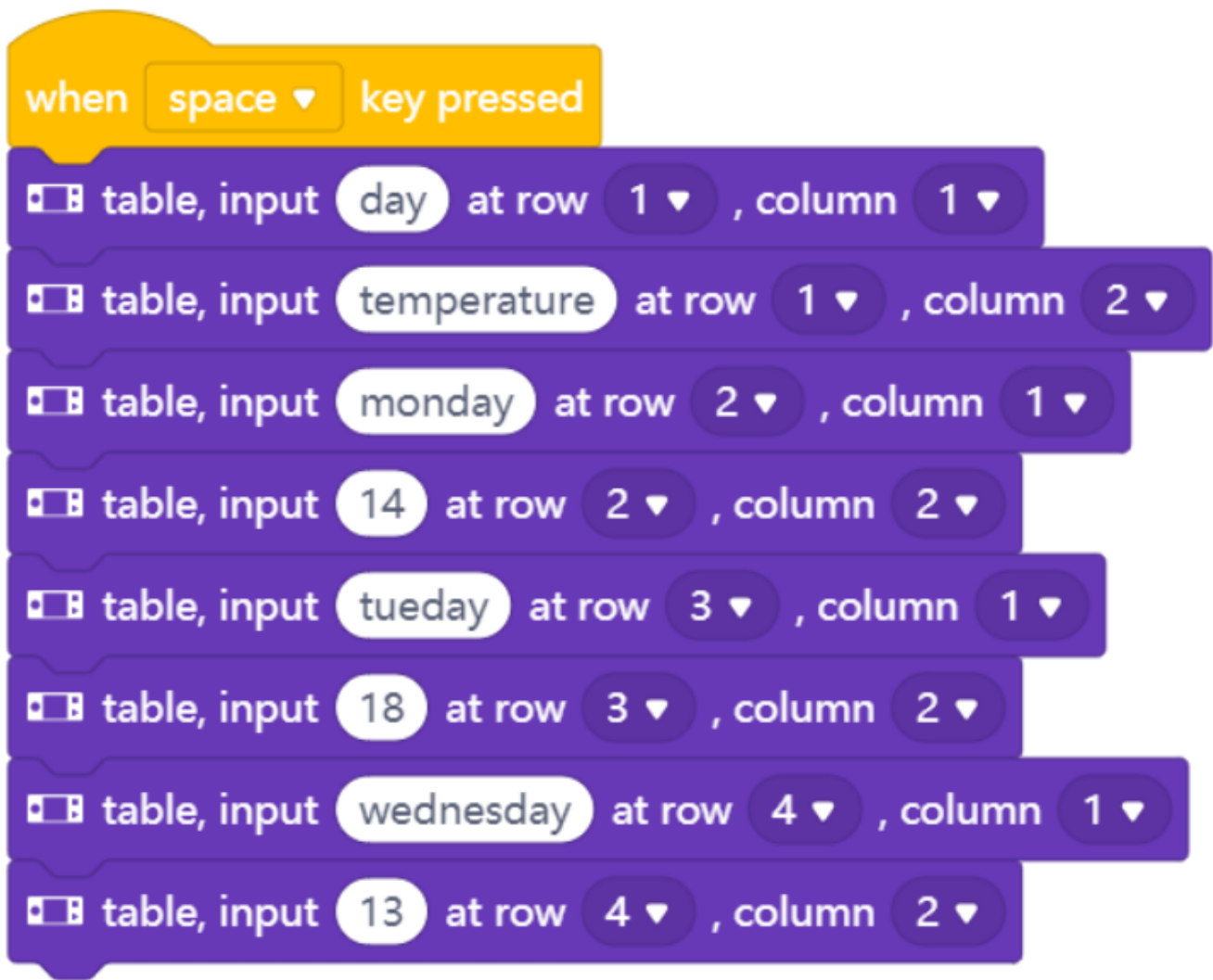
Pag 27 **Manual mBot2** <https://www.robotix.es/documentos/mbot2-actividades.pdf> contenidos e imágenes publicados con permiso de la empresa Robotix.



Pag 28 **Manual mBot2** <https://www.robotix.es/documentos/mbot2-actividades.pdf> contenidos e imágenes publicados con permiso de la empresa Robotix.



Pag 29 de **Manual mBot2** <https://www.robotix.es/documentos/mbot2-actividades.pdf> contenidos e imágenes publicados con permiso de la empresa Robotix.



Pag 33 de **Manual mBot2** <https://www.robotix.es/documentos/mbot2-actividades.pdf> contenidos e imágenes publicados con permiso de la empresa Robotix.

```

when button A pressed
  start recording
  wait 3 seconds
  stop recording
    
```

```

when button B pressed
  play recording
    
```

Pag 41 de **Manual mBot2** <https://www.robotix.es/documentos/mbot2-actividades.pdf> contenidos e imágenes publicados con permiso de la empresa Robotix.

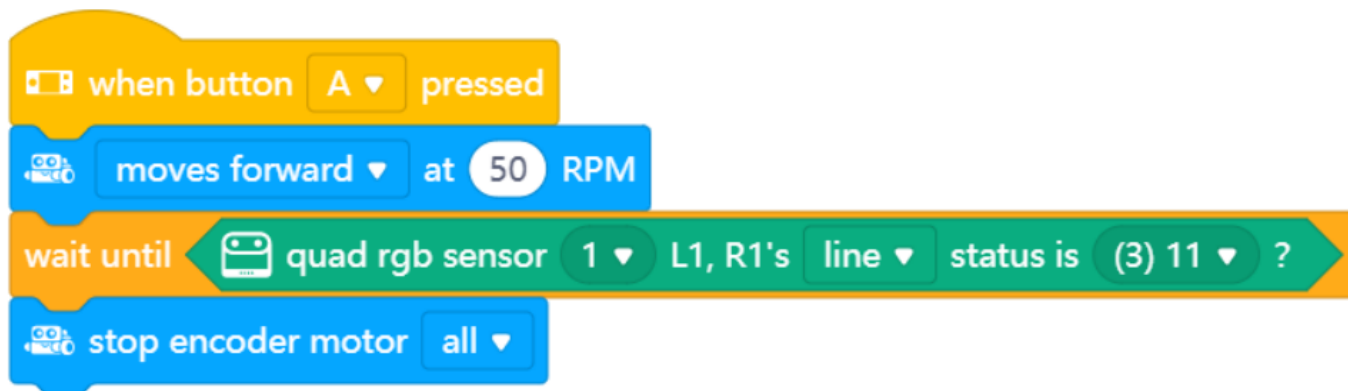
```

when button A pressed
  start recording
  wait 3 seconds
  stop recording
    
```

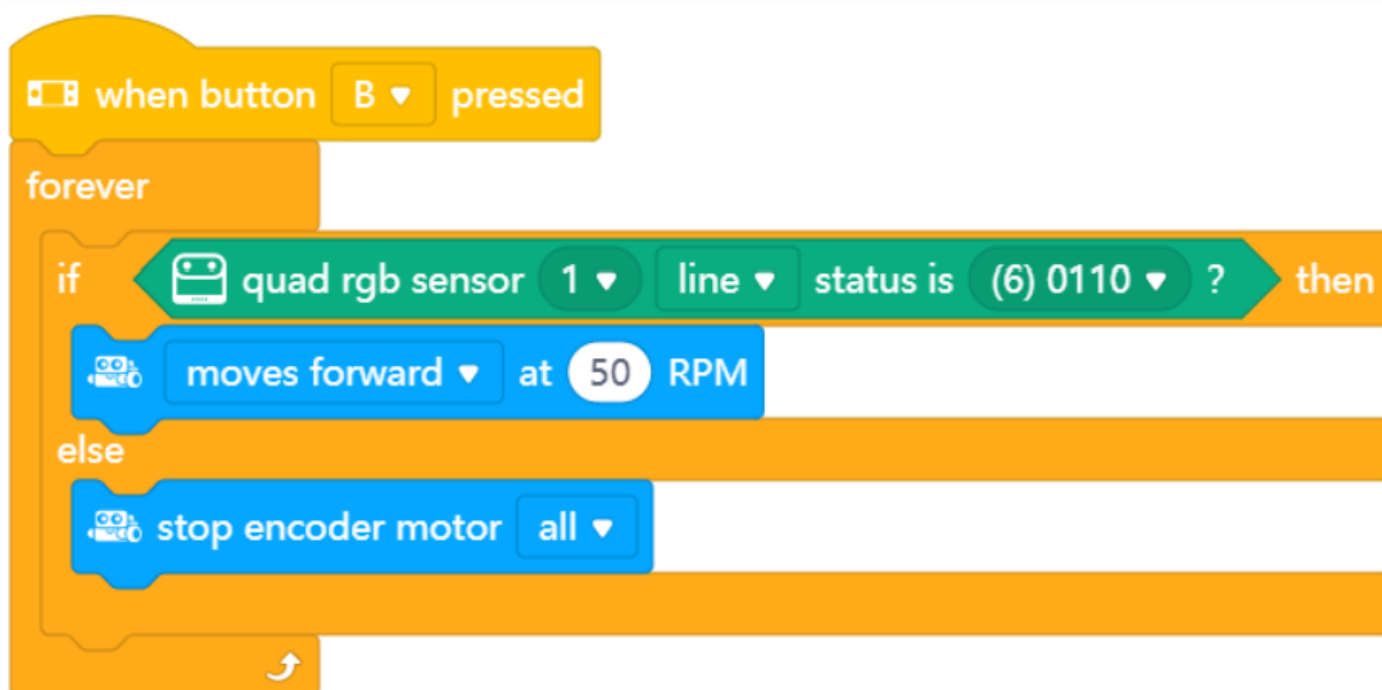
```

when button B pressed
  forever
    if ultrasonic 2 1 distance to an object (cm) < 50 then
      play recording until done
      encoder motor all rotates by 90
    else
      moves forward at 50 RPM
    
```

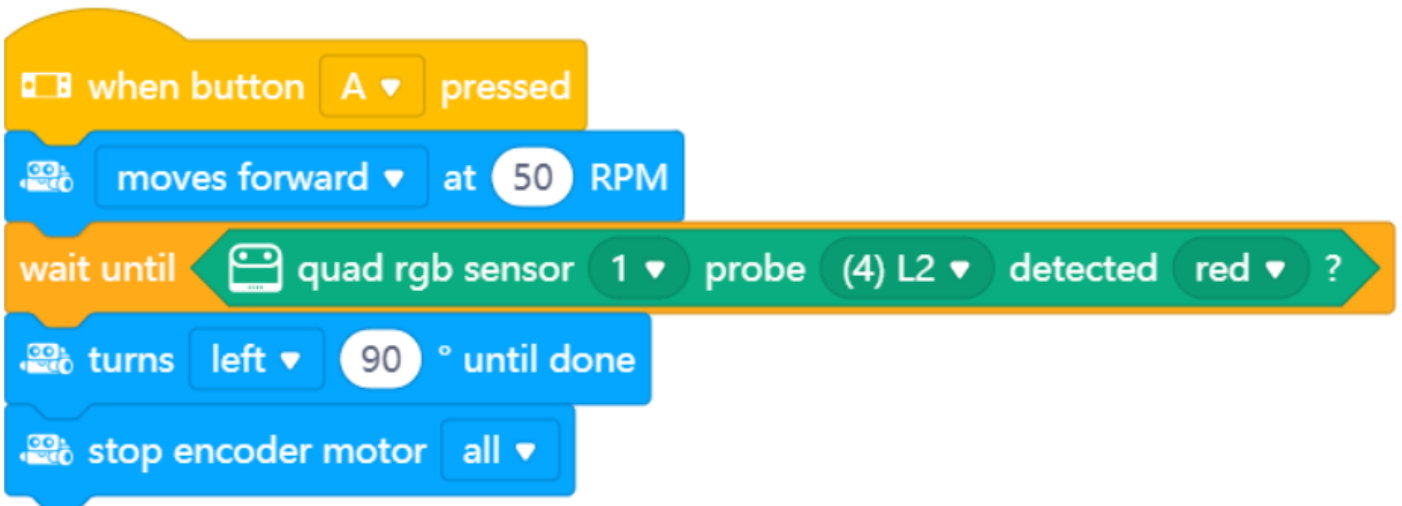
Pag 43 de **Manual mBot2** <https://www.robotix.es/documentos/mbot2-actividades.pdf> contenidos e imágenes publicados con permiso de la empresa Robotix.



Pag 60 de **Manual mBot2** <https://www.robotix.es/documentos/mbot2-actividades.pdf> contenidos e imágenes publicados con permiso de la empresa Robotix.



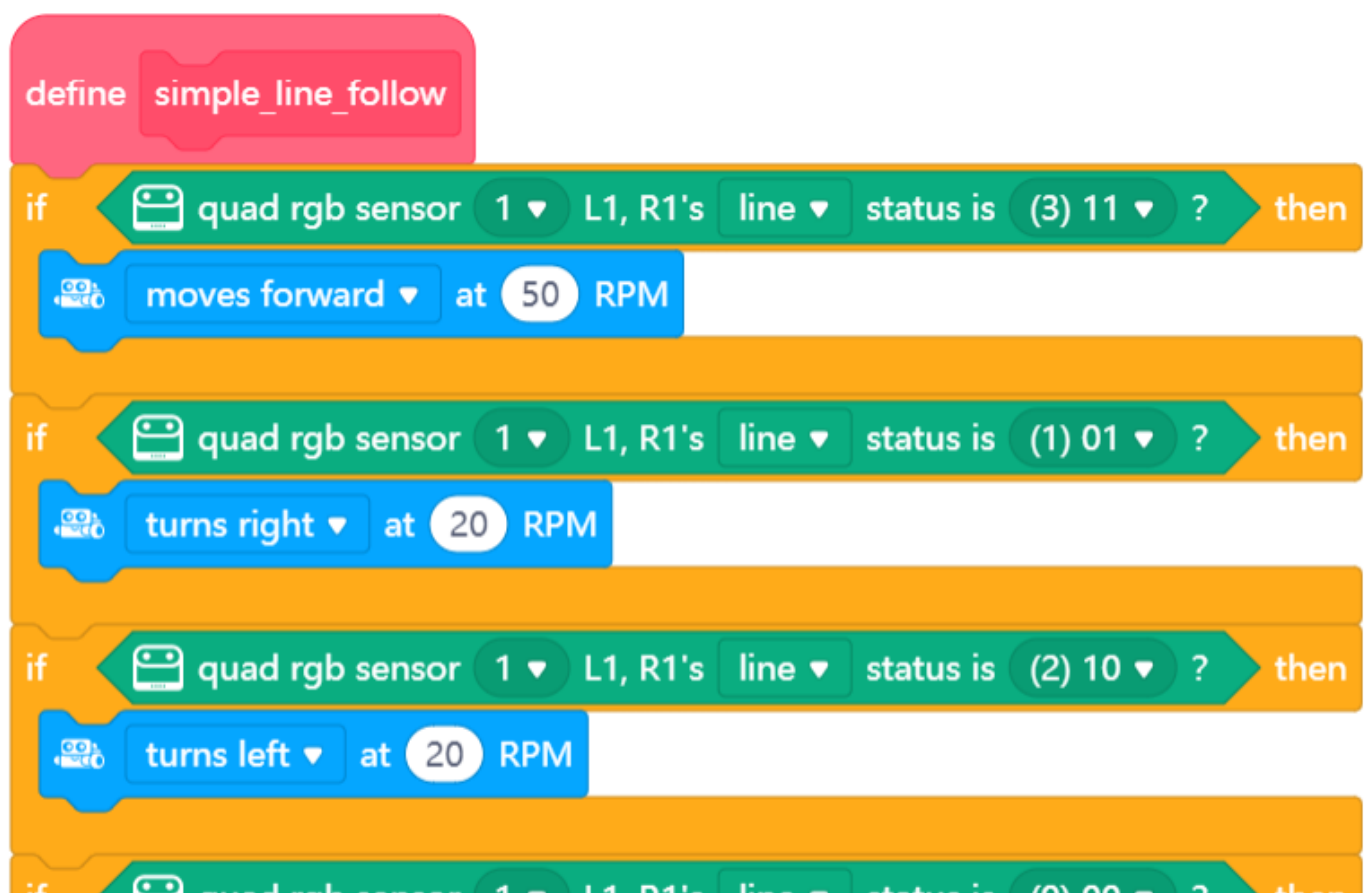
Pag 60 de **Manual mBot2** <https://www.robotix.es/documentos/mbot2-actividades.pdf> contenidos e imágenes publicados con permiso de la empresa Robotix.



Pag 62 de **Manual mBot2** <https://www.robotix.es/documentos/mbot2-actividades.pdf> contenidos e imágenes publicados con permiso de la empresa Robotix.



Pag 63 de **Manual mBot2** <https://www.robotix.es/documentos/mbot2-actividades.pdf> contenidos e imágenes publicados con permiso de la empresa Robotix.



```

when joystick middle pressed
  repeat until button B pressed?
    simple_line_follow
    if quad rgb sensor 1 probe (2) R1 detected blue ? then
      moves forward at 10 RPM for 1 secs
    if quad rgb sensor 1 probe (2) R1 detected red ? then
      turns left 180 ° until done
      moves forward 4 cm until done
    if quad rgb sensor 1 probe (2) R1 detected green ? then
      turns left 90 ° until done
      wait 1 seconds
      turns right 180 ° until done
      wait 1 seconds
      turns left 90 ° until done
      moves forward 4 cm until done
  stop encoder motor all
  
```

Pag 67-68 de **Manual mBot2** <https://www.robotix.es/documentos/mbot2-actividades.pdf>  
 contenidos e imágenes publicados con permiso de la empresa Robotix.



```

define line_follow speed steering sensor
set left to (-1 * speed + steering * sensor)
set left to (1 * speed - steering * sensor)
encoder motor EM1 rotates at left RPM, encoder motor EM2 rotates at right RPM

```

Y utilizar esta instrucción en la función principal

```

line_follow 50 0.6 quad rgb sensor 1 deviation (-100~100)

```

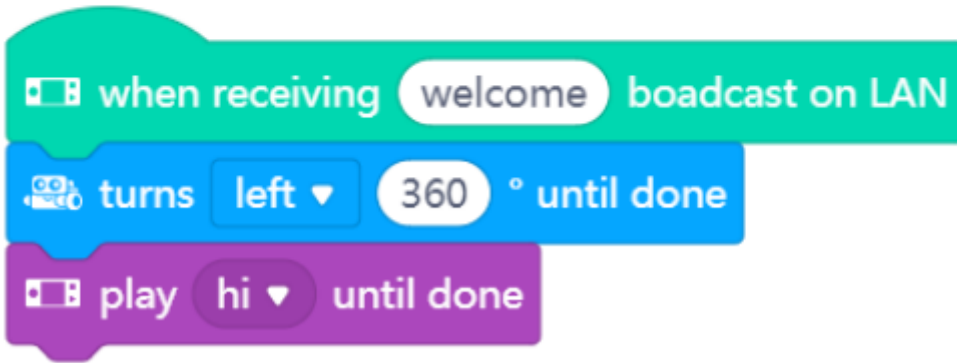
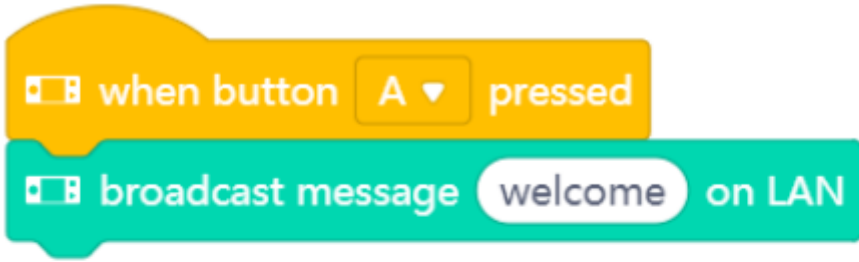
Pag 69 de **Manual mBot2** <https://www.robotix.es/documentos/mbot2-actividades.pdf> contenidos e imágenes publicados con permiso de la empresa Robotix.

```

when space key pressed
  reset rotated angle around all axis
  repeat until button B pressed?
    if tilted right(roll) angle (°) > 10 or tilted right(roll) angle (°) < -10 then
      moves forward at 20 RPM
    else
      moves forward at 40 RPM
  stop encoder motor all

```

Pag 82 de **Manual mBot2** <https://www.robotix.es/documentos/mbot2-actividades.pdf> contenidos e imágenes publicados con permiso de la empresa Robotix.



Pag 89 de **Manual mBot2** <https://www.robotix.es/documentos/mbot2-actividades.pdf> contenidos e imágenes publicados con permiso de la empresa Robotix.

```

when CyberPi starts up
  connect to Wi-Fi ssid password password
  display [red][orange][yellow][green][cyan]
  wait until [network connected?]
  display [green][green][green][green][green]
  show label 1 Shall the robot drive? at center of screen by middle pixel
  show label 2 press B to answer at middle at bottom by middle pixel
  
```

```

when button B pressed
  recognize (4) English 3 secs
  show label 1 [speech recognition result] at center of screen by middle pixel
  if [speech recognition result] contains yes ? then
    broadcast message yes on LAN
  
```

```

when receiving yes broadcast on LAN
  moves forward 10 cm until done
  
```

Pag 106 de **Manual mBot2** <https://www.robotix.es/documentos/mbot2-actividades.pdf> contenidos e imágenes publicados con permiso de la empresa Robotix.

En el siguiente programa se usa la extensión TEACHABLE MACHINE, en ella se enseña al objeto que reconozca (INTELIGENCIA ARTIFICIAL) un mBot2, un block de notas, mira

<https://libros.catedu.es/books/cyberpi-y-mbot2/page/software-educable-maquina-educable->

[teachable-machine](#) para saber cómo se utiliza esta extensión.

Una vez "aprendido" se pueden usar las instrucciones de "recognition result is... "

Tiene la pega que tienen que estar conectado por cable el mBot2 o usar un adaptador Bluetooth

Programa objeto

Programa mBot2

```
when this sprite clicked
  forever loop
    if recognition result is mBot2 ? then
      send upload mode message mBot2
    if recognition result is notebook ? then
      send upload mode message notebook
    if recognition result is nothing ? then
      send upload mode message nothing
```

```
when receiving upload mode message mBot2
  play yeah until done
  moves forward at 50 RPM for 2 secs

when receiving upload mode message notebook
  play surprised until done
  moves backward at 50 RPM for 2 secs
```

Pag 124 de **Manual mBot2** <https://www.robotix.es/documentos/mbot2-actividades.pdf> contenidos e imágenes publicados con permiso de la empresa Robotix.