

Sigue líneas I

Mi primer siguelíneas

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

```
al pulsar el botón A
para siempre
  si ¿L1, R1 de línea del sensor cuádruple RGB 1 en estado (0) 00 ? entonces
    retrocede a 50 rpm
  si ¿L1, R1 de línea del sensor cuádruple RGB 1 en estado (3) 11 ? entonces
    avanza a 50 rpm
  si ¿L1, R1 de línea del sensor cuádruple RGB 1 en estado (1) 01 ? entonces
    girar a la derecha a 50 rpm
  si ¿L1, R1 de línea del sensor cuádruple RGB 1 en estado (2) 10 ? entonces
    girar a la izquierda a 50 rpm
```

- 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

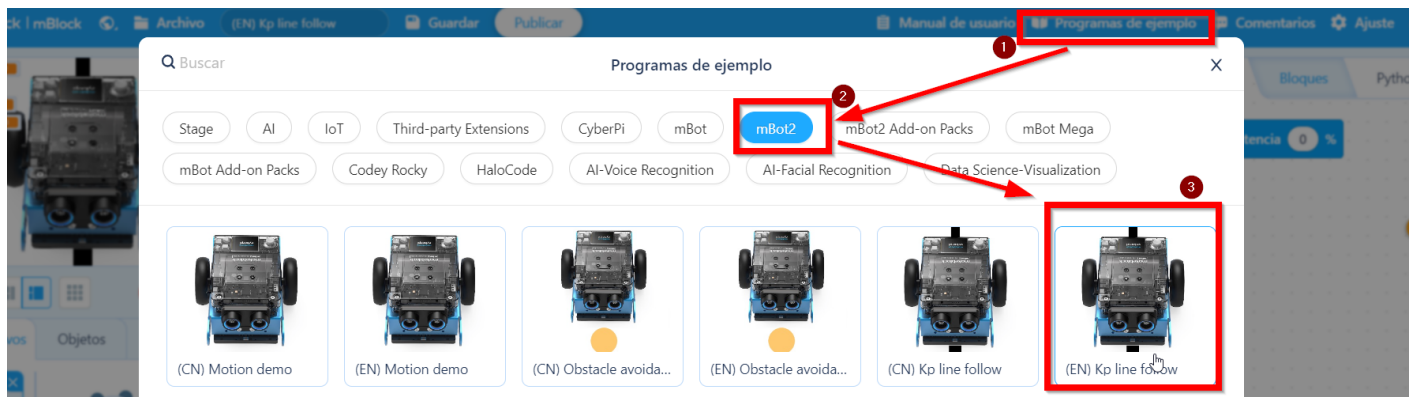
Como se puede ver, hace movimientos hacia atrás sobre todo en las curvas cerradas pero **¡¡no se escapa!!**

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

El sígue líneas de "Programas de ejemplo"

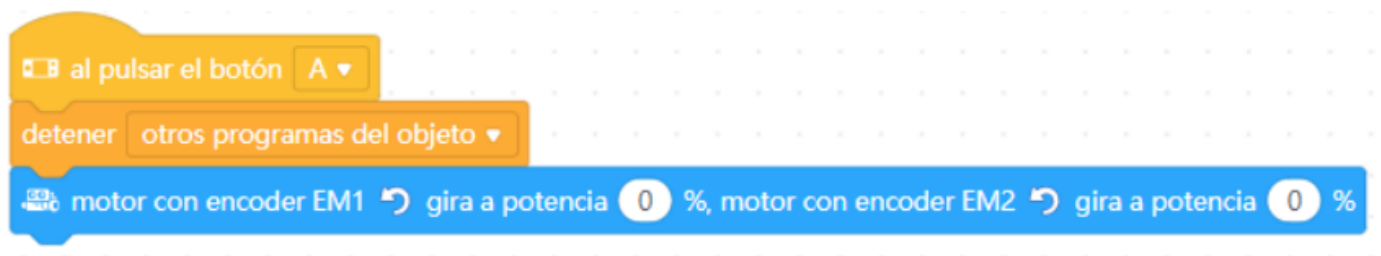
Acuérdate que para manejar el sensor de distancia, hay que instalar la extensión correspondiente. Ver <https://libros.catedu.es/books/cyberpi-y-mbot2/page/como-usar-mbot2-en-mblock>

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

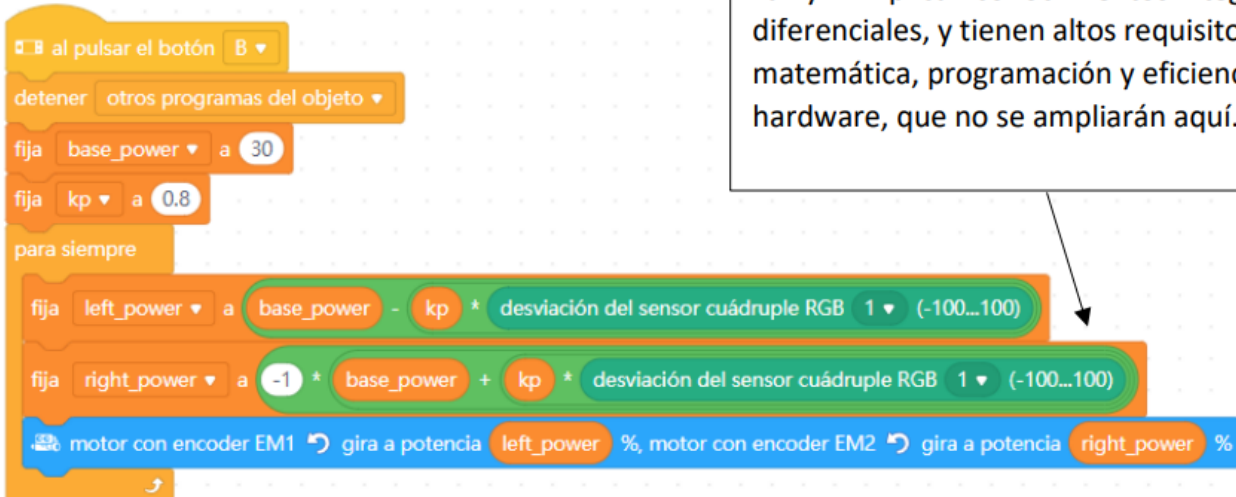
kp

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.

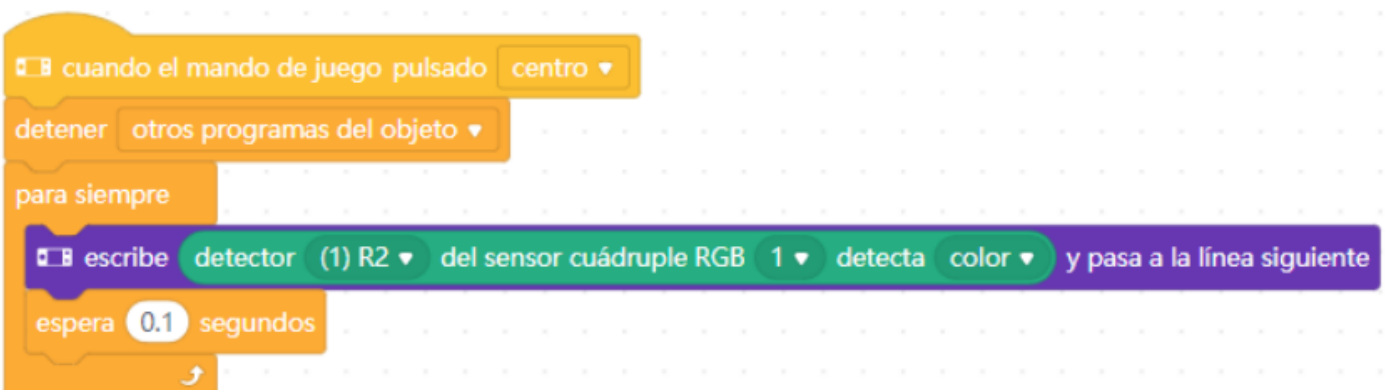
La I y D implican conocimientos integrales y diferenciales, y tienen altos requisitos de matemática, programación y eficiencia de hardware, que no se ampliarán aquí.



```

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>

Revision #5

Created 2024-07-17 10:29:41 CEST by Javier Quintana



Updated 2025-04-30 13:43:03 CEST by Javier Quintana