

EL LABERINTO



STEMJAM Teaching Guide

Developing make spaces to promote creativity
around STEM in schools

Acronym: STEMJAM

Project no. 2016-1-ES01-KA201-025470

www.stemjam.eu



Co-funded by the
Erasmus+ Programme
of the European Union

EL LABERINTO

RESUMEN

La actividad consiste en resolver un laberinto.

En la primera versión que proponemos los laberintos están hechos con línea negra. El mBot utiliza dos sensores Sigue Líneas.

En la segunda versión, los laberintos los construimos con muros. Para su resolución, el robot utiliza un sensor de Ultrasonidos para seguir la pared lateral y un sensor Sigue Líneas para detectar las paredes frontales.

OBJETIVOS DIDÁCTICOS

Esta actividad tiene como objetivos:

- ❖ Aprender a desarrollar un algoritmo para resolver el laberinto.
- ❖ Aprender sobre el funcionamiento de diferentes sensores y componentes.
- ❖ Desarrollar el pensamiento computacional.

Materia STEM: Ciencia Tecnología Ingeniería Matemáticas

Nivel educativo: 12-14 años 14-16 años

PLANTEAMIENTO DEL PROBLEMA

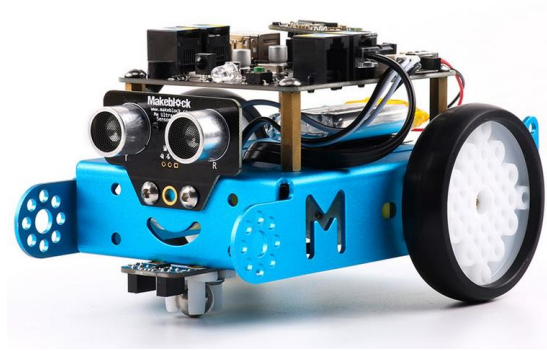
El mBot tiene que encontrar la salida de un laberinto:

1. Usando dos sensores Sigue Líneas, en el laberinto hecho con la línea negra
2. Usando tanto el sensor de Ultrasonidos como el sensor Sigue Líneas.

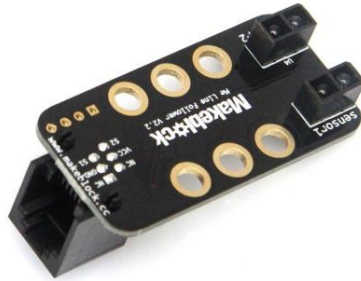


LISTA DE MATERIALES

❖ Robot mBot



❖ Sensor Sigue Líneas (x2):



❖ Sensor de Ultrasonidos:



➤ Diversas estructuras y vigas:



- ❖ Un laberinto, hecho con líneas negras.
- ❖ Un laberinto hecho con muros.

Versión 1:

ELEMENT	ID	CABLE	AMOUNT	PORT 1			PORT 2			PORT 3			PORT 4			P.MOT 1	P.MOT 2
				Y	B	W	Y	B	W	Y	B	W	B	I	Y		
Mbot Robot 2'4G			1														
Motor 1	W*		1													W*	
Motor 2	W*		1														W*
Me Line Follower	B		2					B			B						
RJ25 cables			2														
Structures and beams			1														
Laptops			1														
Attrezzatura (not essential)																	

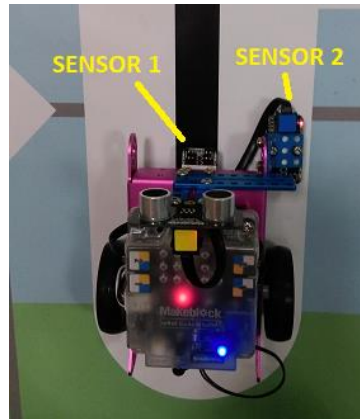
Versión 2:

ELEMENT	ID	CABLE	AMOUNT	PORT 1			PORT 2			PORT 3				PORT 4				P.MOT1	P.MOT2
				Am	Az	Bl	Am	Az	Bl	Am	Az	Bl	Ng	Am	Az	Bl	Ng		
Mbot Robot 2'4G			1																
Motor 1	Bl*																	Bl*	
Motor 2	Bl*																		Bl*
Ultrasonic sensor	Az	1	1							Az									
Me Line Follower	Az	1	1					Az											
RJ25 cables			2																
Structures																			
Brass Stud M4x20			2																
Cut-out beam			1																
Nuts and bolts (Pairs)			6																
Attrezzatura - Maze (with white walls)			1																

DESCRIPCIÓN DE LA ACTIVIDAD

Versión 1:

El robot está equipado con dos sensores Sigue Líneas. El sensor nº1 debe seguir la línea negra y darse cuenta del algoritmo clásico "ir a lo largo de la línea". El sensor nº2 comprueba qué color está ubicado a la derecha de los robots.



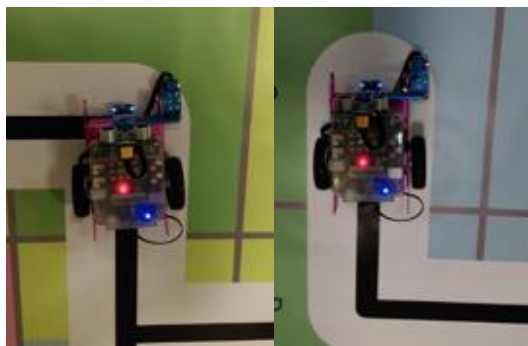
El algoritmo básico de moverse por el laberinto es un algoritmo de "mantenerse a la derecha". Se puede definir de la siguiente manera:



El mBot avanza recto cuando el sensor nº1 detecta negro y el sensor nº 2 detecta blanco.



Cuando el sensor nº2 detecta negro, el mBot debe girar hacia la derecha.



Cuando ambos sensores detectan blanco, el robot gira hacia la izquierda hasta que el sensor nº1 detecte negro.


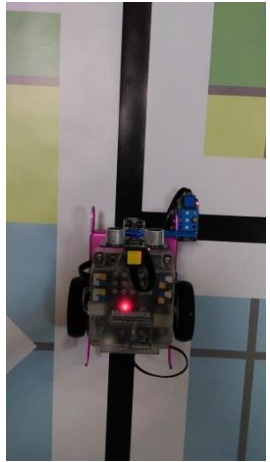
En el caso en el que la línea se haya terminado el robot girará.

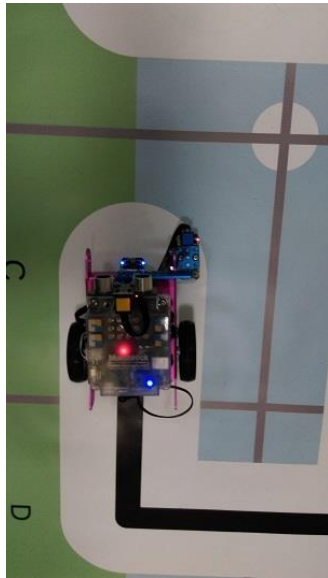
En caso de que haya un giro a la izquierda, el robot girará a la izquierda.

Las lecturas del sensor Sigue Líneas son:

Ambos sensores detectan blanco	3
Ambos sensores detectan negro	0
Sensor izquierda negro – Sensor derecha blanco	1
Sensor izquierda blanco – Sensor derecha negro	2

Ahora podemos crear las condiciones para controlar el movimiento del robot. Crearemos dos variables: *“line_detect”* y *“right_side”*.

	<p>El robot va recto cuando el sensor nº1 detecta negro y el sensor nº2 detecta blanco.</p> <p>Pero el sensor nº1 puede detectar que el robot pierde la línea. Tenemos que corregir la ruta como en el programa básico seguidor de línea.</p>	<ul style="list-style-type: none"> • Si <i>“line_detect”</i>=0 and <i>“right_side”</i>=3 SIGUE RECTO • Si <i>“line_detect”</i>=1 and <i>“right_side”</i>=3 GIRA A LA IZQUIERDA • Si <i>“line_detect”</i>=2 and <i>“right_side”</i>=3 GIRA A LA DERECHA
	<p>Cuando el sensor nº2 entra en la línea negra, el robot debe girar a la derecha.</p>	<ul style="list-style-type: none"> • Si <i>“line_detect”</i>=0 and <i>“right_side”</i>=0 GIRA A LA DERECHA <p>Esta situación es ideal, pero cuando intente ejecutar este programa, el robot no girará a la derecha con fluidez. Durante el giro, el sensor derecho cambia las lecturas a 1 ó 2. Así que cambiamos:</p> <ul style="list-style-type: none"> • Si <i>“line_detect”</i>=0 and <i>“right_side”</i> ≠3 GIRA A LA DERECHA



Cuando ambos sensores detectan blanco, significa girar a la izquierda hasta que el sensor nº1 encuentre negro.

En caso de que la línea termine, el robot girará, y cuando haya un giro a la izquierda, el robot girará a la izquierda.

- Si “line_detect”=3 and “right_side”=3 GIRA A LA IZQUIERDA

Todos estos casos los ponemos en bucle “para siempre”:

```
mBot Program
wait until on board button pressed
wait until on board button released
forever
  set line_detect to line follower Port2
  set right_side to line follower Port3
  if line_detect = 0 and right_side = 3 then
    run forward at speed 100
  else
    if line_detect = 2 and right_side = 3 then
      turn right at speed 100
    else
      if line_detect = 1 and right_side = 3 then
        turn left at speed 100
      else
        if line_detect = 3 and right_side = 3 then
          turn left at speed 100
        else
          if line_detect = 0 and not right_side = 3 then
            turn right at speed 100
          else
            turn right at speed 100
            wait 0.5 secs
```

En esta fotografía puedes ver un ejemplo de laberinto hecho con líneas negras:



Versión 2:

El robot permanecerá inmóvil hasta que se presione y suelte el botón de la placa. A partir de este momento, el programa principal comenzará a funcionar, combinando el sensor Sigue Líneas y el sensor de Ultrasonidos, para detectar las paredes del laberinto.

La primera condición se establece para con el sensor Sigue Líneas, (que se coloca en la parte frontal del mBot):

- ❖ Si detecta que el blanco es porque está caminando hacia la pared, entonces girará 90 grados a la derecha.
- ❖ Si no detecta blanco, se inicia la segunda condición establecida para con el sensor de Ultrasonidos. Este sensor se coloca en el lado derecho del mBot y realizará cuatro rutinas diferentes, dependiendo de la distancia a la pared:
 - Si la distancia es inferior a 5 cm: el mBot está demasiado cerca de la pared del laberinto, por lo que girará a la izquierda lentamente.
 - Si la distancia está entre 5 y 6 cm: el mBot gira a la derecha lentamente, corrigiendo la leve desviación del camino.
 - Si la distancia está entre 6 y 8 cm: el mBot gira a la derecha un poco más rápido, debido a que el desvío es más notable.
 - Si la distancia es mayor que 8 cm: significa que el mBot ha llegado a la esquina de una pared, por lo que girará muy rápido para encontrar nuevamente la pared a seguir.

El mBot funcionará de forma autónoma, siendo el código principal registrado en su placa Arduino.



Código principal:

```
mBot Program
wait until on board button pressed
wait until on board button released
forever
  set motor M1 speed 90
  set motor M2 speed 90
  set right to 90
  set left to 90
  repeat until 0 < line follower Port2
    set distance to ultrasonic sensor Port3 distance
    if distance < 5 then
      set right to 80
      set left to 150
    if 5 < distance and distance < 6 then
      set right to 150
      set left to 80
    if 6 < distance and distance < 8 then
      set right to 175
      set left to 80
    if 8 < distance then
      set right to 200
      set left to 50
      set led on board led left red 20 green 0 blue 0
      set led on board led right red 0 green 255 blue 0
    set motor M1 speed right
    set motor M2 speed left
  run backward at speed 100
  wait 0.1 secs
  set motor M1 speed -110
  set motor M2 speed 110
  set led on board led left red 0 green 0 blue 255
  set led on board led right red 20 green 0 blue 0
  wait 0.8 secs
```

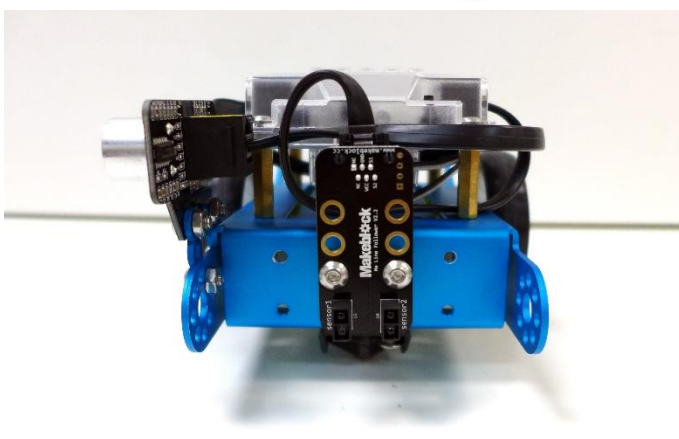
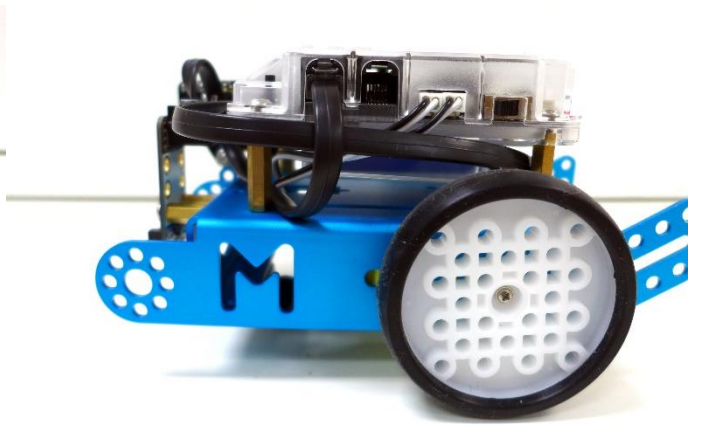
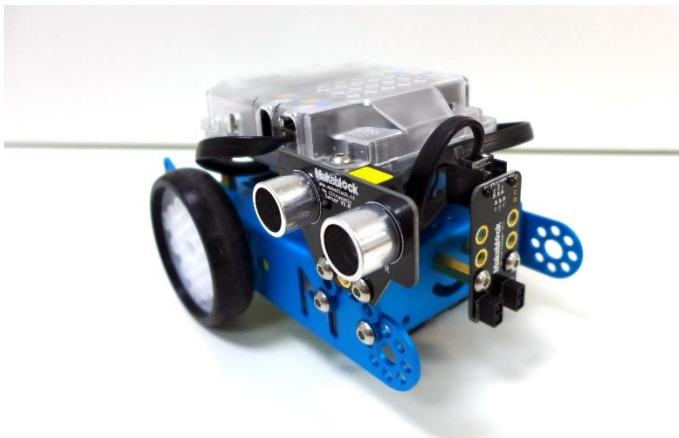
Grabaremos este código en la placa del mBot. De esta manera funcionará independientemente del ordenador y no se necesitará un ordenador personal para poder llevar a cabo la actividad.

Cómo cargar un programa en la placa mBot de Arduino usando mBlock:

Para cargar un programa en la placa usando mBlock:

1. Elija mBot en la pestaña "Placas" del menú mBlock.
2. Conecte el USB y elija "Puerto serie" en la pestaña "Conectar".
3. En la pestaña "Editar", elija "Modo Arduino" (en el programa que vamos a cargar, en lugar de la bandera verde, pondremos el comando azul "programa mBot")
4. Se abrirá una ventana con el código para grabarlo en la placa Arduino de mBot. Usted puede, si lo desea, modificar su programa. Finalmente, haga clic en Cargar en Arduino.
5. Si no ha habido errores, se enviará un mensaje informando que el programa se ha grabado correctamente. En este momento, podrá comenzar a disfrutar el programa introducido en el robot, sin tener que estar el ordenador encendido. Para hacer esto, debe desconectar el cable USB y conectar las pilas (o batería de litio) del robot. Verá que su mBot funciona de forma independiente.

Composición estructural: una vez que ya tenemos el código hecho, pasamos a montar la estructura del robot, con todos los componentes necesarios para esta actividad, tanto mecánicos como electrónicos.



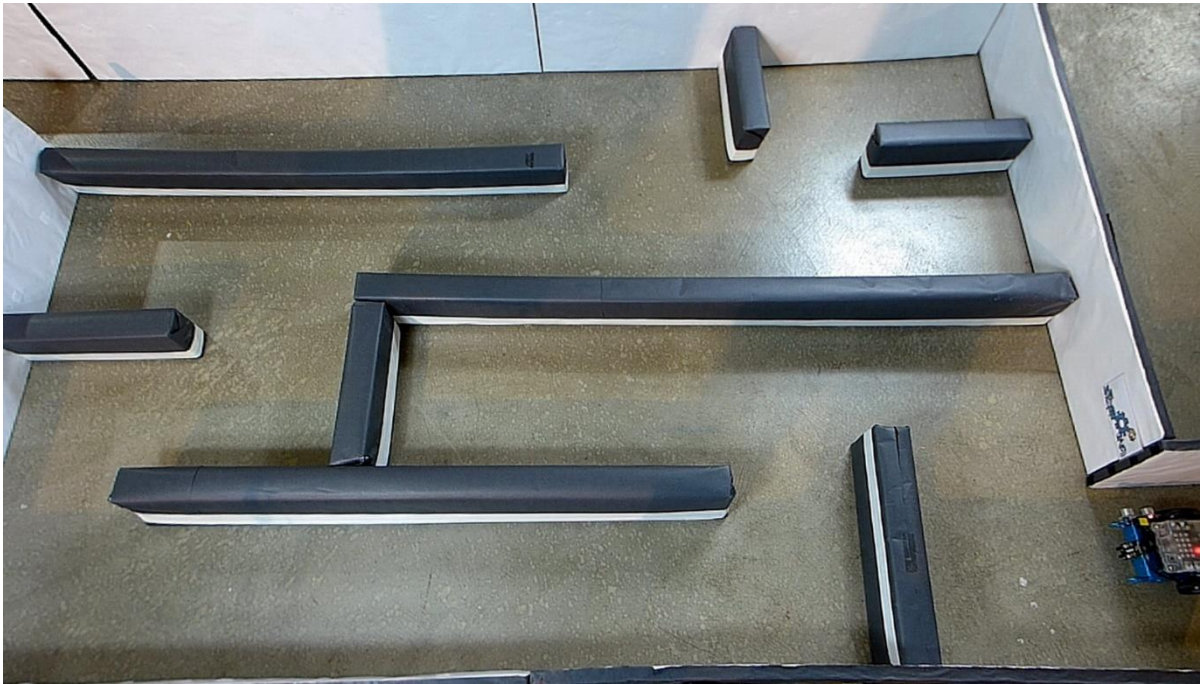
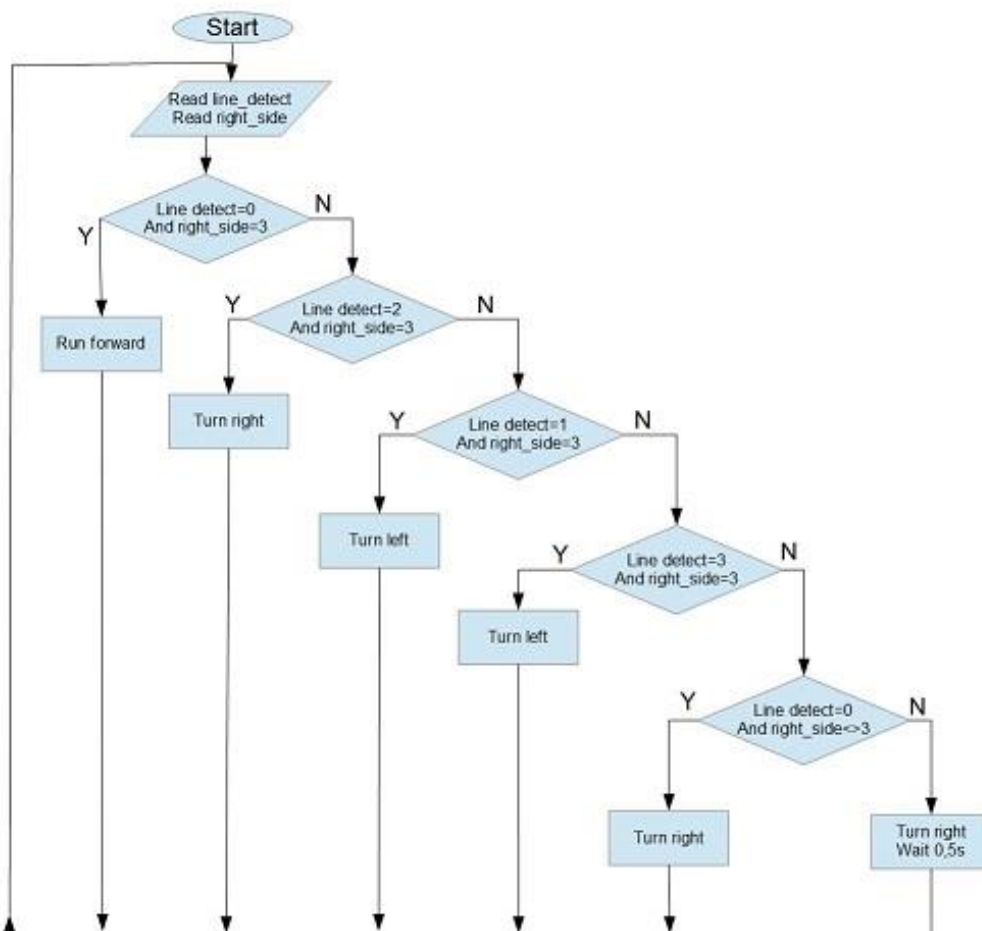
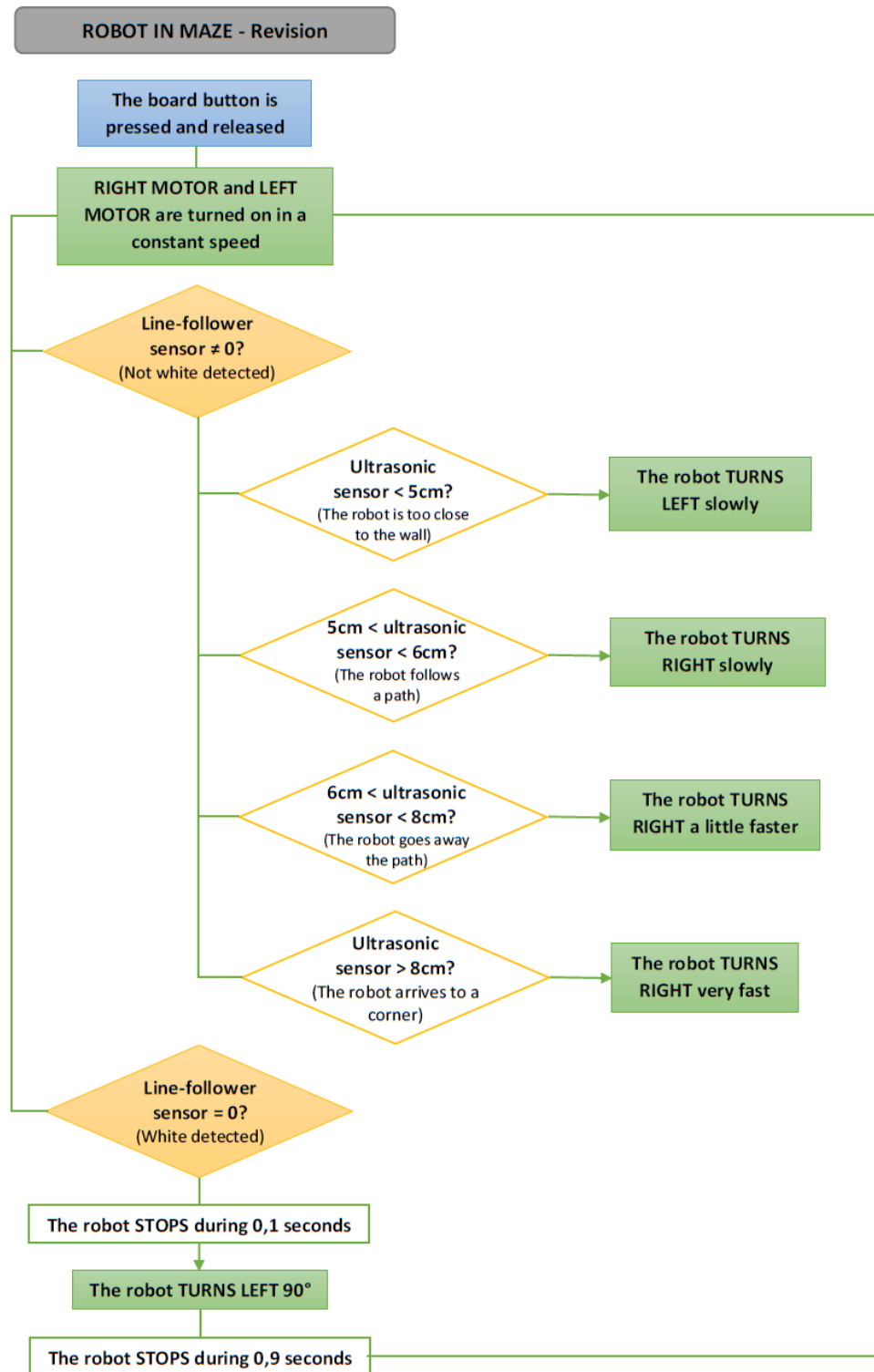


DIAGRAMA DE FLUJO

Versión 1:



Versión 2:



EVALUACIÓN DE LOS ESTUDIANTES

This exercise teaches algorithmic thinking. The right-hand algorithm can also be used for a robot with two distance sensors - this case is difficult and is a good exercise for students to overcome obstacles. The algorithm is clear, but it is technically necessary to choose the right time and distance.

Esta actividad enseña lo que llamamos “pensamiento computacional”. El algoritmo básico de moverse "manteniéndose a la derecha", también se puede usar para un mBot con dos sensores de Ultrasonidos; este caso es difícil pero es un buen ejercicio para que los estudiantes superen obstáculos, superen retos.

BIBLIOGRAFÍA

<http://www.makeblock.com/>

<https://makeblock.es/>

ESCALABILIDAD

Se podría hacer: un primer paseo para conocer el laberinto (su construcción). A continuación, el robot calcula la ruta más corta al destino y el siguiente pasaje se realiza sin callejones sin salida.

Este programa requiere un programa escrito en arduino.