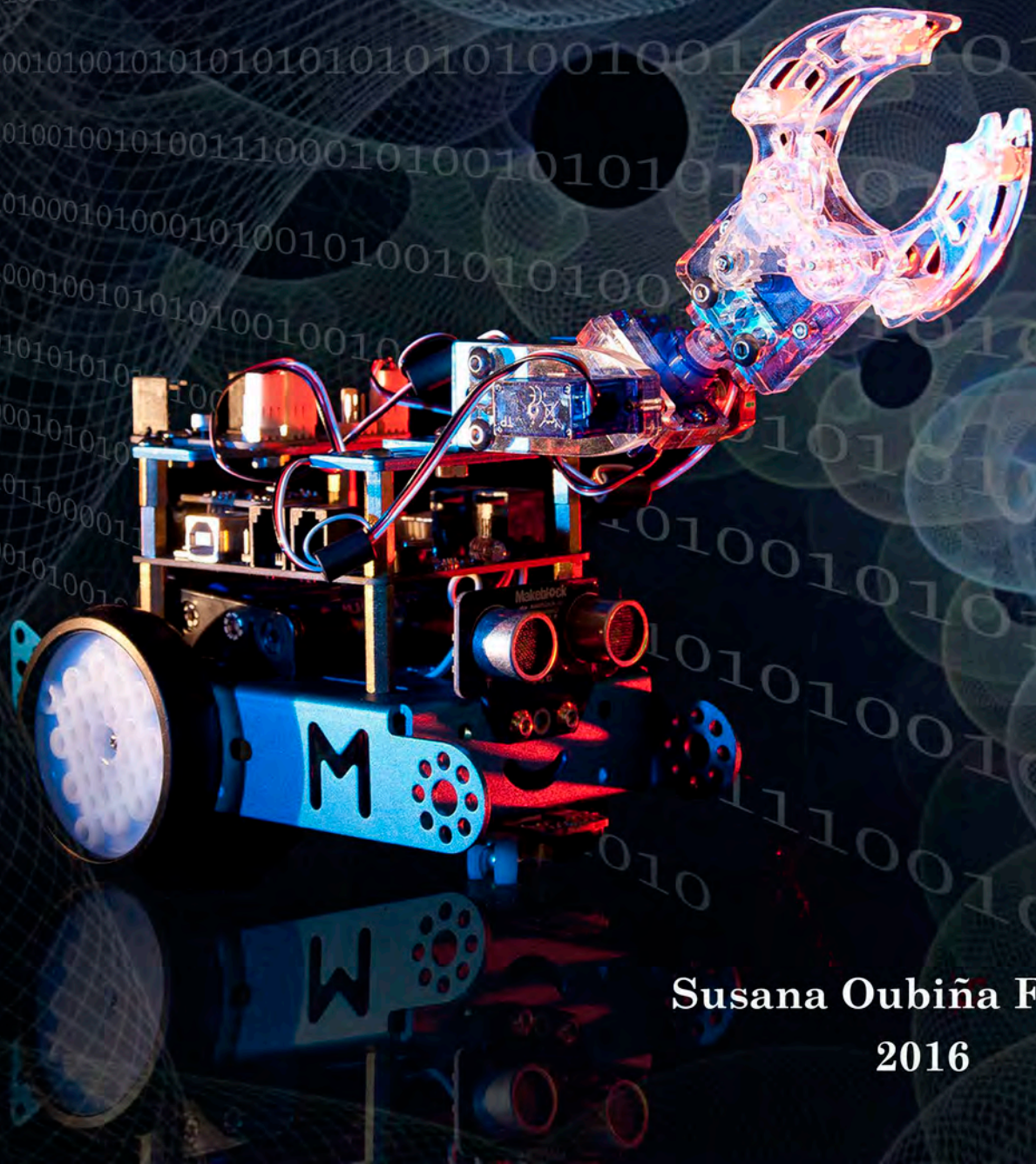


Divertíndome con mBot:

Guía de Manexo e Programación



Susana Oubiña Falcón

2016

Título orixinal: Divertíndome con mBot: Guía de manexo e programación.

Autor: Susana Oubiña Falcón

Deseño da portada: Sesé González García

Editor: Prodel S.A.

1ª Edición: xuño, 2016

 Susana Oubiña Falcón

ISBN-13: 978-84-608-9278-6



"Divertíndome con mBot: Guía de manexo e programación", por Susana Oubiña Falcón, é publicado baixo a licencia [Creative Commons Recoñecemento 4.0 Internacional License](https://creativecommons.org/licenses/by/4.0/).

Encántame a seguinte cita de Albert Einstein: "Se fas sempre o mesmo, non esperes resultados diferentes". Para mín, non esperes resultados diferentes". Para min, en educación, só hai un camiño: Investiga, lánzate e proba!

Índice

1.Introdución	6
1.1. Conectar mBot	6
1.2. Cargar o programa no meu mBot	9
2.Interface e Menú principal	14
2.1. Arquivo	15
2.2. Editar.....	15
2.3. Conectar.....	16
2.4. Placas	18
2.5. Extensións.....	19
2.6. Linguaxe	22
2.7. Axuda.....	23
3.Comandos dos diferentes bloques en mBlock.....	24
3.1. Movemento	24
3.2. Apariencia	25
3.3. Son	26
3.4. Lapis	27
3.5. Datos e Bloques	28
3.6. Eventos	29
3.7. Control	30
3.8. Sensores	31
3.9. Operadores	32
3.10. Robots.....	33
4.Exemplos de programación	37
4.1. Módulo de ultrasóns	37
4.2. Motores	38
4.3. Motor e detección ultrasónica.....	40
4.4. Diodos RGB da placa	41
4.5. Detector de liña	43
4.6. Matriz de LEDs 8×16.....	53
4.7. Robot sumo.....	60
4.8. Servomotor.....	69
4.9. Sensor de temperatura SD18B20.....	71

Divertíndome con mBot: Guía de manexo e programación

4.10. Sensor Me Temperatura e Me Humidade.....	77
4.11. Módulo PIR	79
4.12. Módulo Joystic.....	82
4.13. Potenciómetro	83
4.14. Módulo 4LEDs RGB	85
4.15. Módulo Display 7 segmentos.....	87
4.16. Sensor de Luz	92



Divirtíndome con mBot: Guía de manexo e programación

1. Introducción

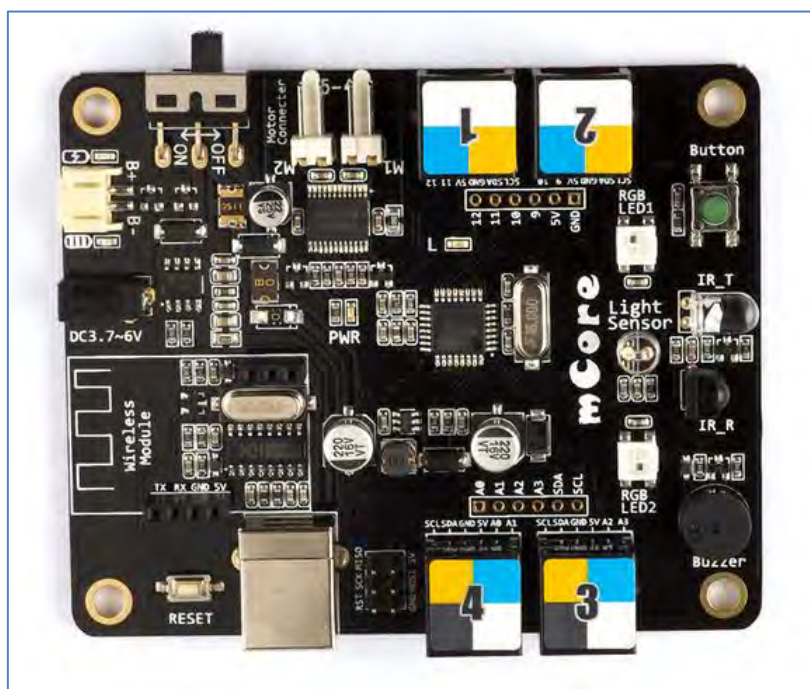
MBot é un robot de 400 gramos baseado en Arduino Un e creado pola casa Makeblock. Pódese alimentar por USB, por batería de litio ou a través do conector do portapilas con 4 baterías AA.

En canto á súa programación e control, hai 4 opcións: pódese programar desde *mBlock*, un software de programación gráfica moi similar a Scratch e tamén se pode controlar mBot, directamente, sen necesidade de programación, desde unha tablet ou o desde un Smartphone, grazas a su *app* (para Android ou iOS) e á posibilidade de acoplarlle o módulo bluetooth. Ademais, o robot é moi versátil xa que podemos usar Arduino con mBot, programando mBot co *IDE de Arduino* e non só con mBlock. Neste último caso, necesitamos instalar as librerías de Makeblock no IDE de Arduino. Librerías dispoñibles na casa comercial para libre descarga aos usuarios. Finalmente, a última opción é programalo co software *mBlockly*.

Este documento centrarase na programación do robot mBot desde o programa mBlock. Software que vexo moi apropiado para o alumnado xa que combina scratch con arduino en diferentes placas e robots (e non só con mBot). En moitas ocasións, entraremos en Arduino e veremos que as librerías que proporciona a casa Makeblock son moi fáciles de manipular, modificar e entender, abríndonos a porta a usar estes robots para aprender Arduino.

1.1. Conectar mBot

O robot mBot utiliza a placa mCore que pode verse na imaxe inferior. A placa, cun microcontrolador ATmega238, dispón de 4 portos con conexións RJ25 para conectar sensores e dous portos para conectar os motores. Ademais, mCore integra un interruptor de encendido, un botón, dous LEDs RGB, dous LEDs normais, un buzzer, un sensor de luminosidade e un sensor de infravermellos receptor-emisor.



Placa mCore

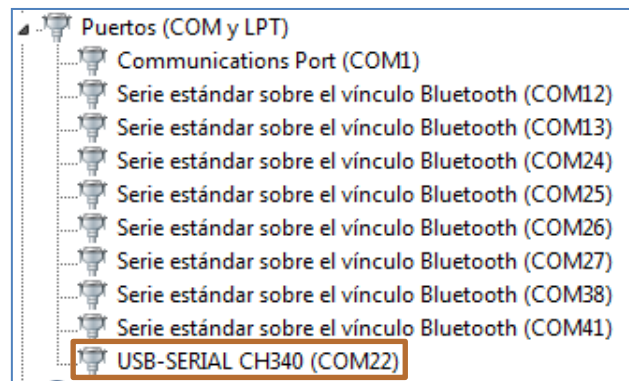
Divertíndome con mBot: Guía de manexo e programación

Na versión de mBot, saída ao mercado español en xuño de 2016, a placa vai protexida cunha carcasa plástica.

A descarga do software mBlock, tanto para Windows como para iOS, pode facerse desde o seguinte link: www.mblock.cc

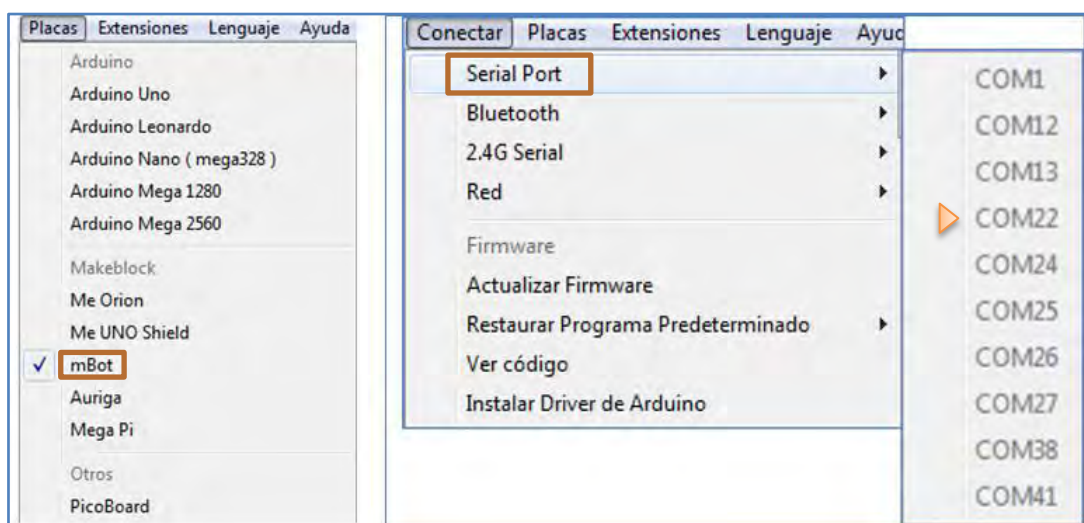
En canto a Linux, o programa funciona a través do Wine, software que nos permite instalar aplicaciónes de Windows en Linux, dando permisos ao usuario no arquivo ttyUSB0.

Tras abrir o programa mBlock, o primeiro que debemos facer é conectar o robot ao software que o controlará. Para iso, conectamos o cable USB desde o mBot ao ordenador. Nese momento, se os drivers de mBot están instalados, habilítasenos un porto COM de comunicación para realizar a conexión. No noso caso, o noso porto é o COM22, como pode verse na seguinte imaxe. Se queremos coñecer o porto que utiliza o noso PC nese momento, en Windows7 podemos acceder a *Equipo>Administrador de dispositivos*, no apartado de Portos (COM e LPT):



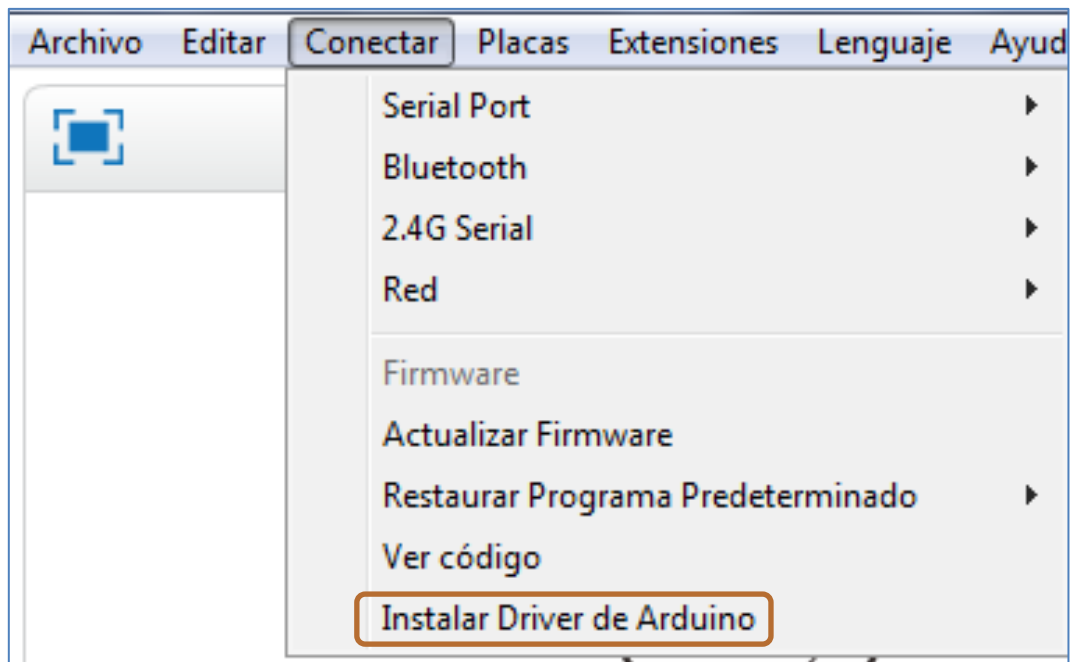
Porto de conexión USB-SERIAL CH340

Xa no programa *mBlock*, imos á pestana “*Placas*” e escogllemos *mBot* para finalmente, ir á pestana “*Conectar*” do menú principal e no seu apartado *Serial Port*, seleccionar o porto de conexión COM22:

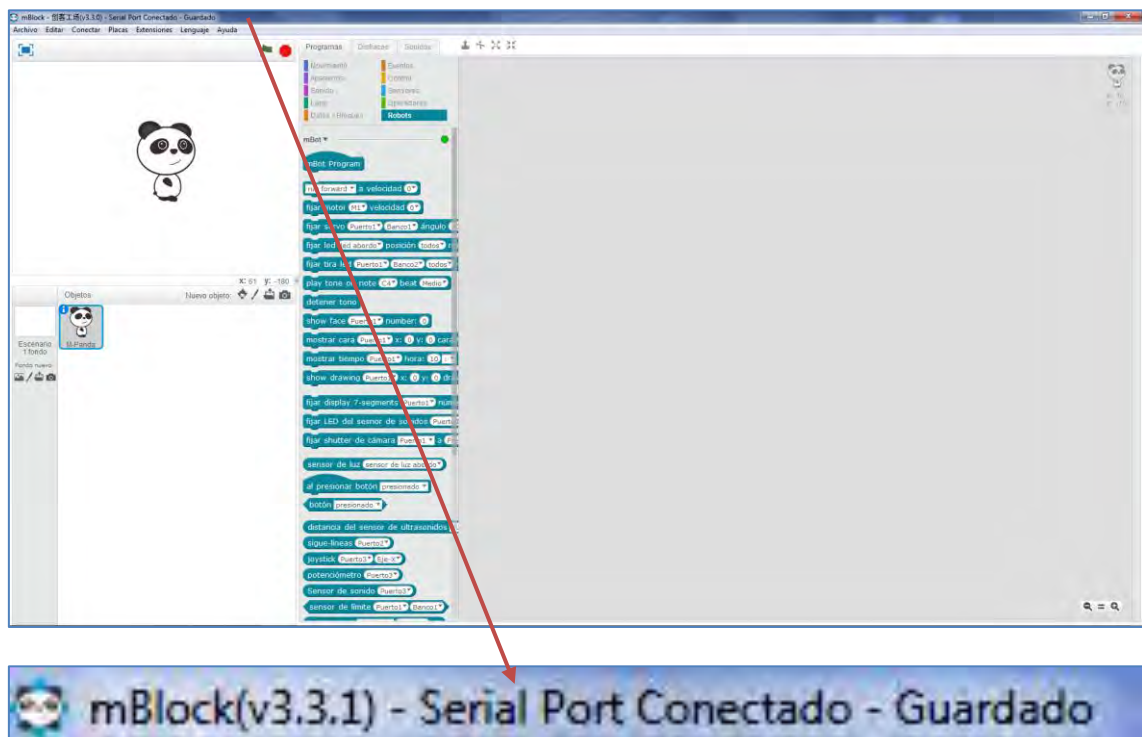


Divertíndome con mBot: Guía de manejo e programación

En caso de non conectar, só debemos revisar se instalamos o driver de arduino que nos proporciona o porto de conexión USB-SERIAL CH340. Este driver pode instalarse directamente desde o programa mBlock, a través da pestana *Conectar* usando a opción *Instalar Driver de Arduino*:



Tras estes pasos, o programa débemos amosar que mBot está conectado:



No bloque **Robots** observamos que mBot está conectado (círculo en verde):

Divertíndome con mBot: Guía de manexo e programación



O noso traballo agora, será simplemente programar o robot utilizando os diferentes bloques que nos proporciona mBlock. O software mBlock permítenos conectarnos co robot por porto USB (que é o que fixemos ata agora), por Bluetooth ou por 2.4G.

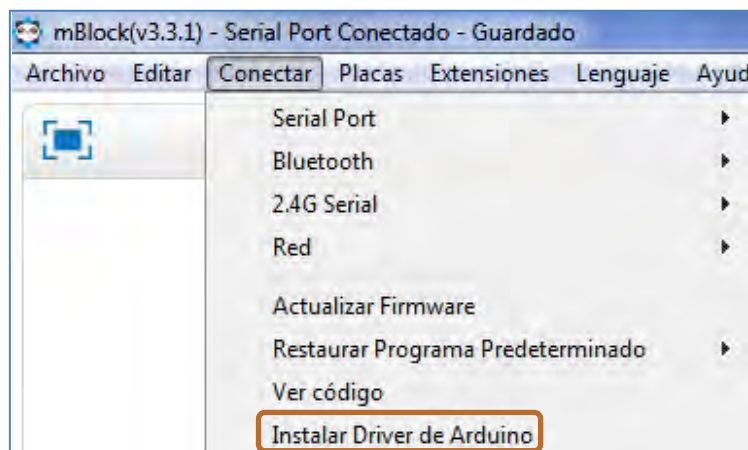
1.2. Cargar o programa no meu mBot

Podería interesarnos que o noso programa funcione completamente **independiente** do ordenador (sen cable USB, Bluetooth ou 2.4G). Para iso, unha opción é grabalo no Arduino do mBot:

IMPORTANTE: Con esta opción de grabado, no noso programa non podemos utilizar ningún comando propio de Scratch. Por exemplo, no bloque Apariencia, "Dicir..." correspondería á personaxe "Panda" do programa mBlock e chocaría coa opción de grabado.

Os pasos a seguir para grabar o programa creado no Arduino do mBot (mCore) son os seguintes:

- ❖ Paso 1: Debemos instalar (se procede) os drivers do mBot: *Conectar > Instalar Driver de Arduino*

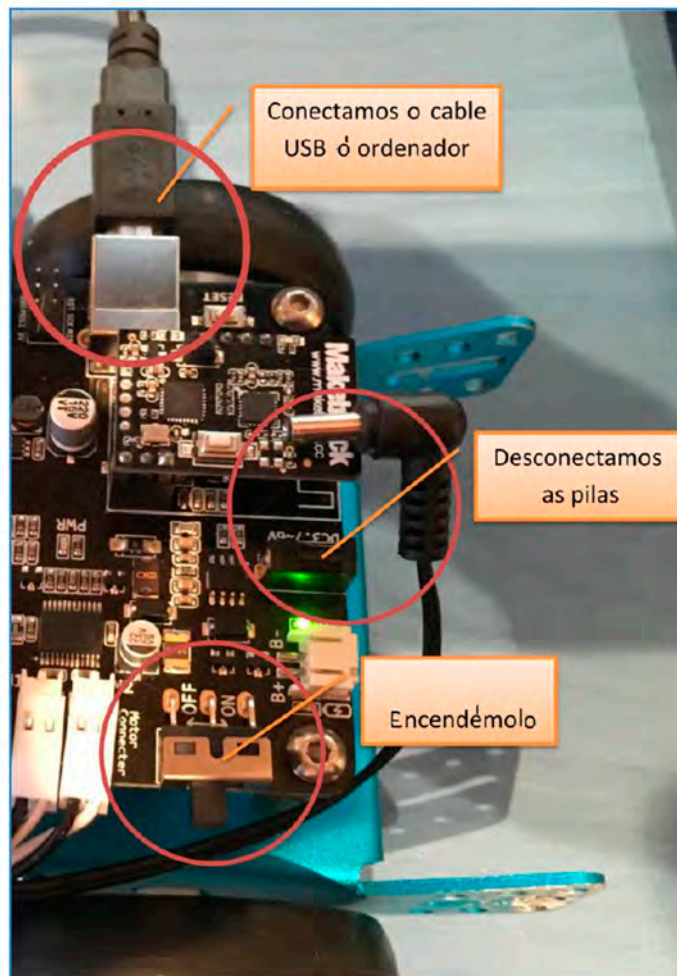


Divertíndome con mBot: Guía de manexo e programación

- ❖ Paso 2: Crear o programa que queremos grabar dentro da placa mCore.
- ❖ Paso 3: Desconectamos o noso mBot do ordenador xa que o rxe, vese que desconectamos (non activa) a conexión 2.4G:

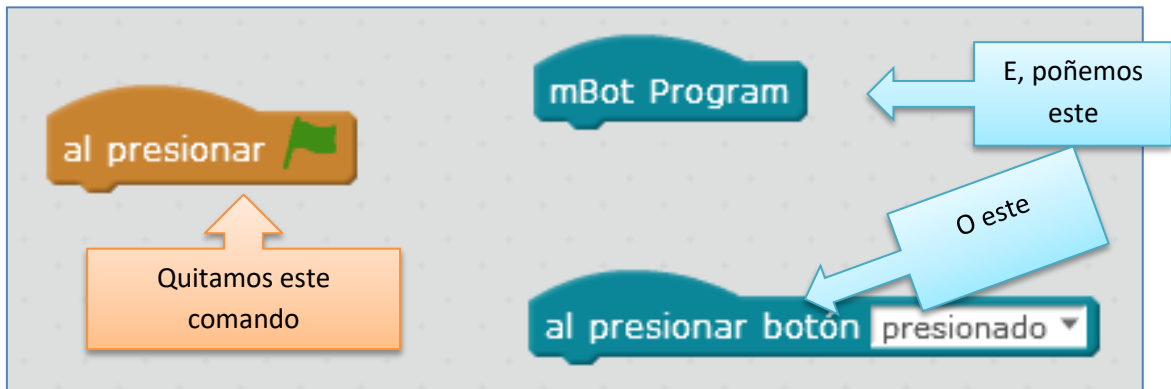


- ❖ Paso 4: Conectamos o mBot co cable USB, desconectándoo das baterías e encendémolo:



Divertíndome con mBot: Guía de manexo e programación

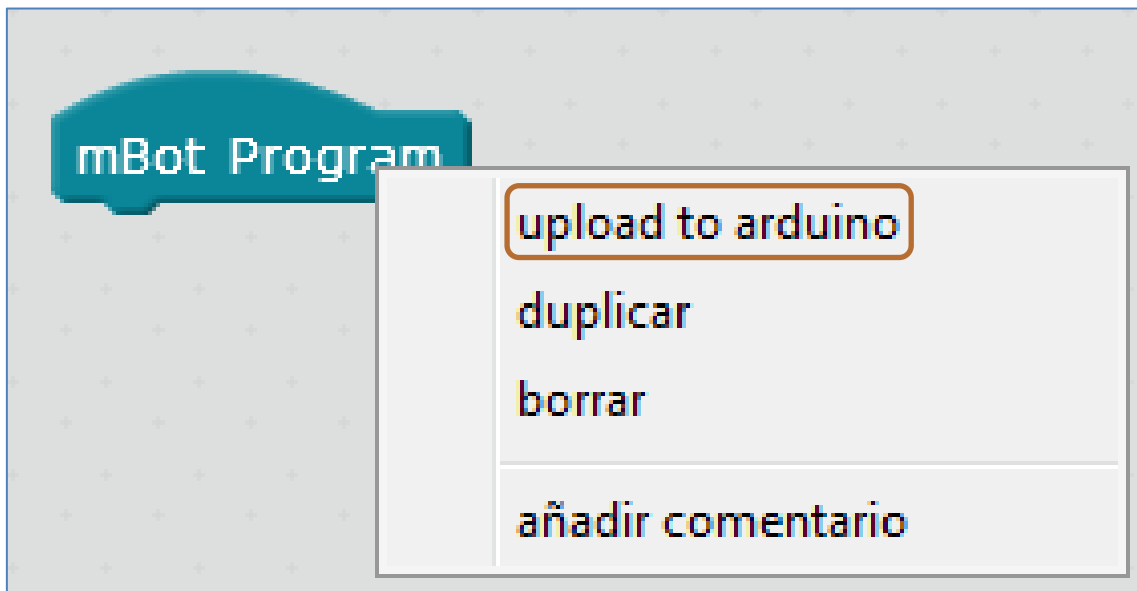
- ❖ Paso 5: Conectamos o mBlock con mBot polo porto serie (Serial Port) COM X correspondente:
- ❖ Paso 6: Cambiamos no noso programa o comezo do programa (o da bandeira) por un propio do robot “mBot Program”:



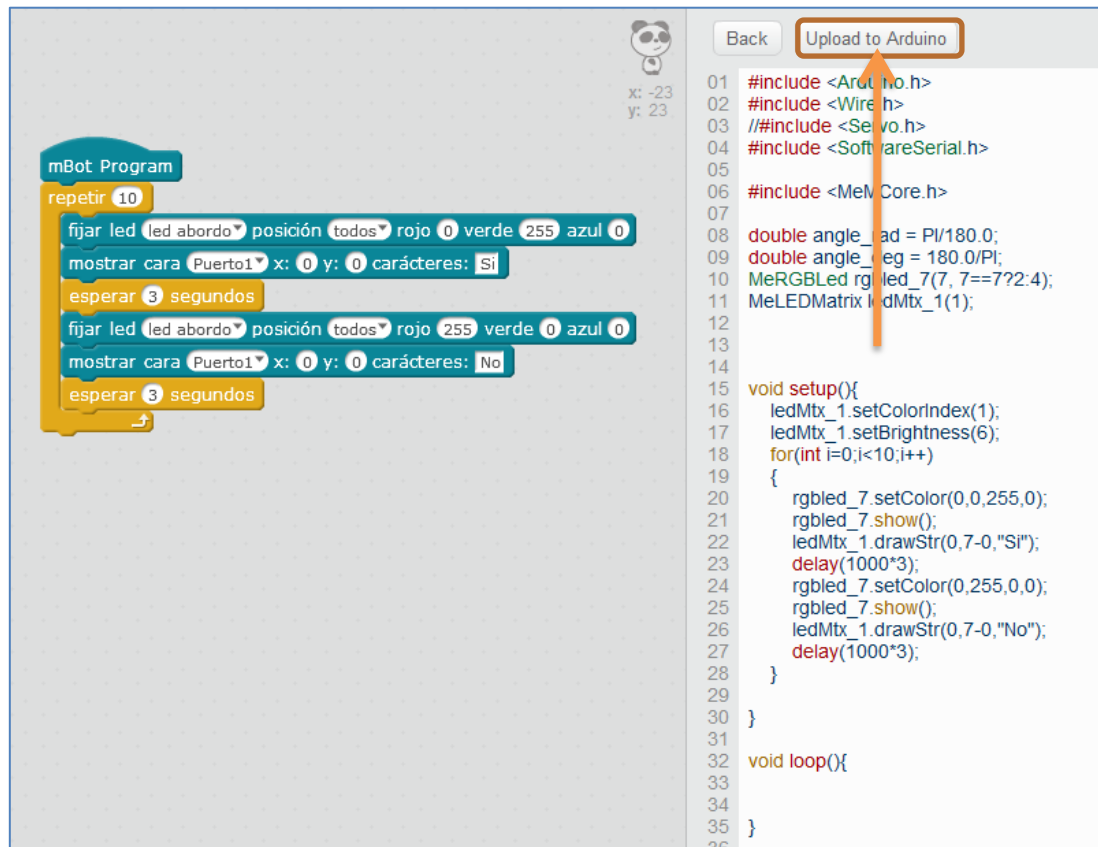
Usamos o comando “mBot Program”

IMPORTANTE: Só podemos cargar na placa un único comando “mBot Program”.

- ❖ Paso 7: Co botón dereito, pinchamos no comando **mBot Program** e eliximos *Upload to Arduino*. Outra opción é Editar > Modo Arduino > Upload to Arduino:



- ❖ Paso 8: Abrirásenos unha ventá co código para grabalo no Arduino do mBot. Incluso, podemos modificar o que queiramos. Finalmente, facemos clic en *Upload to Arduino* (ver seguinte imaxe):



3 Paso 9: Si no se han producido fallos, se debe recibir un mensaje que nos informa que el programa se ha ejecutado correctamente, pudiendo comenzar a disfrutar del programa introducido en el robot, sin necesidad de estar conectado al ordenador. Para ello, debemos desconectar el cable USB y conectar las pilas o la batería de litio del robot. Veremos que el mBot funciona de forma independiente.

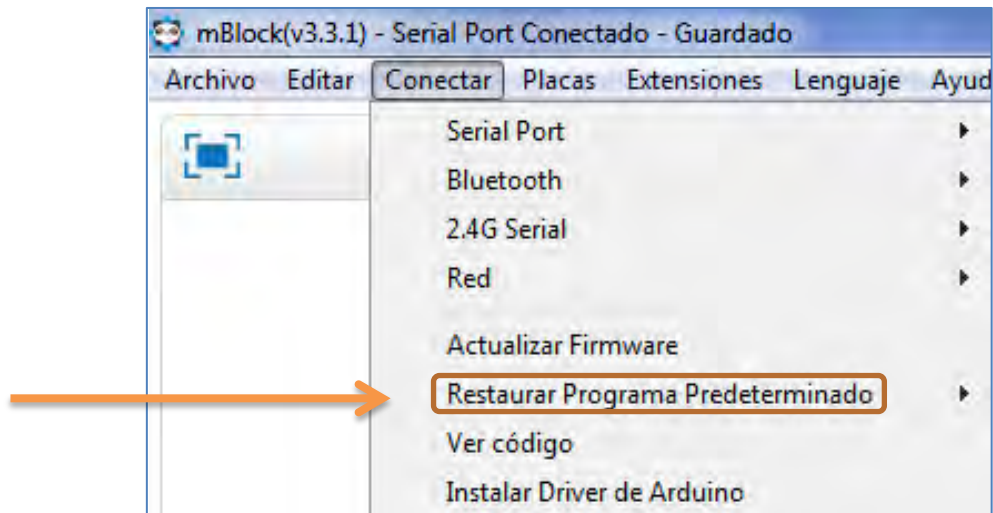
❖ Paso 10: Como se quita el programa que se ha ejecutado? Obviamente, en algún momento necesitaremos quitar el programa que se ha ejecutado para introducir otro nuevo o simplemente, seguir trabajando con mBlock (sin embargo, no podremos volver a conectarse con mBlock usando la bandeja verde, por ejemplo). Para ello, sólo debemos seguir los siguientes pasos:

1. Tenemos que volver a desconectar las pilas o la batería y conectar el cable USB del mBot al PC (ver paso 4).

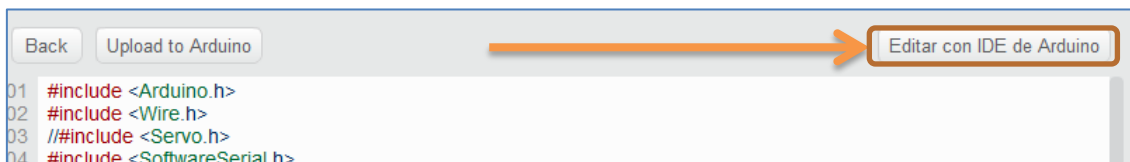
2. Conectar mBot por puerto serie correspondiente (ver paso 5). Si no aparece, debemos instalar otra vez los drivers.

3. Resetear el arduino del mBot. Este paso borrará el nuestro programa y dejará al robot como estaba antes, con el programa de fábrica: *Conectar > Restaurar Programa Predeterminado*

Divertíndome con mBot: Guía de manexo e programación



Se nos fixamos, no Paso 8 temos a opción de editar o noso arquivo co IDE de Arduino e, polo tanto, cargalo desde aí:



Resumindo, mBot ten dúas formas de funcionar atendendo á dependencia ou independencia dun PC:

- **Dependente do ordenador:** hai tres formas de conectalo. A saber, conexión inalámbrica 2.4G ou conexión Bluetooth e mediante unha conexión por cable USB.
- **Independente do ordenador:** Só por Conexión por cable USB cárgase o programa (Upload to Arduino). Logo, desconéctase e o robot xa pode ir só sen necesidade dun PC.

É interesante o seguinte tutorial para traballar con bluetooth:

<https://www.youtube.com/watch?v=FxAsufUNclI>

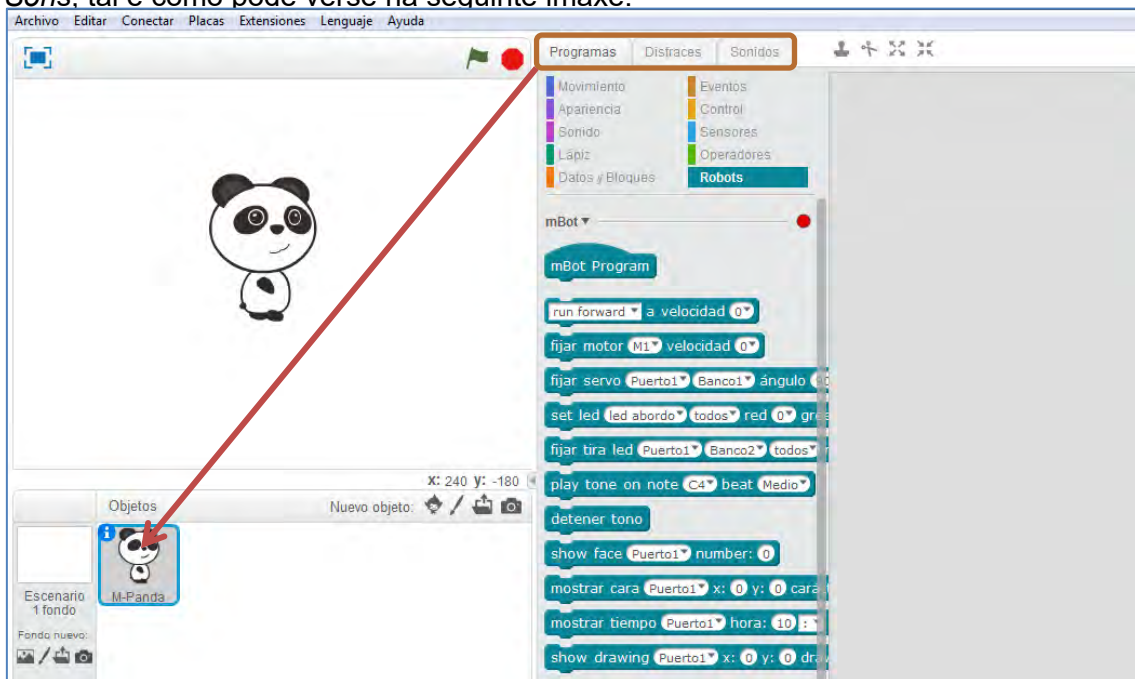
Divertíndome con mBot: Guía de manexo e programación

2. Interface e Menú principal

O software mBlock basease en scratch. De feito, ambas interfaces (Scratch2.0 y mBlock) son moi semellantes.

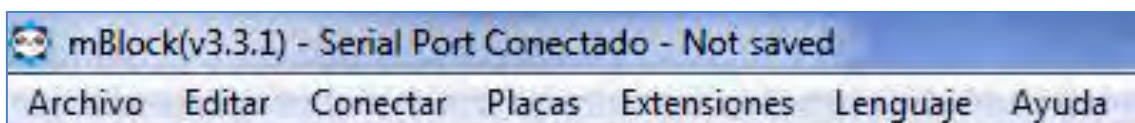
Scratch é unha linguaxe de programación desenvolto por un equipo dirixido por Mitchell Resnick no Media Lab del MIT. As súas orixes nacen na linguaxe LOGO, o cal foi pensado e creado para ser utilizado por nenos (estudantes), para que eles desenvolveran habilidades matemáticas. Programouse baixo a linguaxe chamada squeak e este, á súa vez, a partir da linguaxe smalltalk. Ambas, linguaxes orientadas a obxectos. En consecuencia, Scratch, que é unha evolución do LOGO, é *unha linguaxe de programación que se caracteriza por estar totalmente orientado a obxectos*. É dicir, os obxectos comunícanse entre eles dentro dun mundo virtual. Ademais, caracterízase pola súa simplicidade e por ser un linguaxe modular (utilizando bloques).

O software mBlock segue a súa mesma filosofía. Nel, cada obxecto e escenario presentan a súa propia pestana *Programas*, *Disfraces* (ou *Fondos* se é un escenario) e *Sons*, tal e como pode verse na seguinte imaxe:



Interface de mBlock

O menú principal componse de sete pestanas, comezando pola cela Archivo e finalizando na cela de selección Axuda. Na súa parte superior podemos ver a versión do programa que se está utilizando e se a placa que se pretende programar está ou non conectada co software por porto serie.

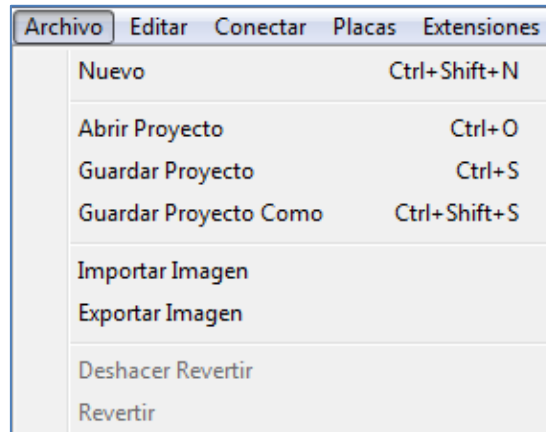


A continuación desenvolveranse as opcións de cada unha das pestanas do menú superior.

Divertíndome con mBot: Guía de manexo e programación

2.1. Arquivo

As opcións da pestana “Arquivo” son:

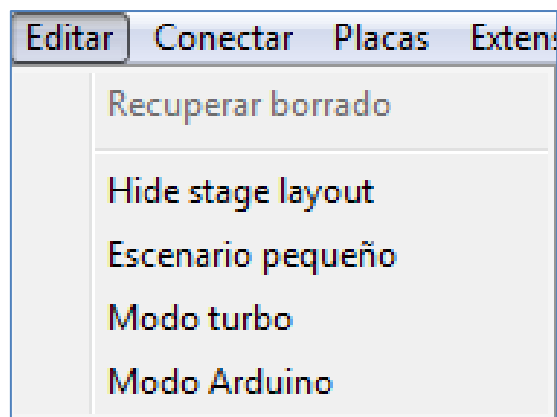


Opcións da pestana Arquivo

- *Novo*: crea un novo proxecto
- *Abrir Proxecto*: Abre un proxecto existente
- *Gardar Proxecto*: Garda o arquivo
- *Gardar Proxecto Como*: Garda o proxecto cun nome cuxa extensión é .sb2
- *Importar e exportar imaxe*: introduce ou exporta unha imaxe dun obxecto ou sprite.
- *Desfacer Revertir*: Desfai a acción revertir
- *Revertir*: Volve o programa ao estado que tiña anteriormente

2.2. Editar

As opcións da pestana “Editar” son:



Opcións da pestana Editar

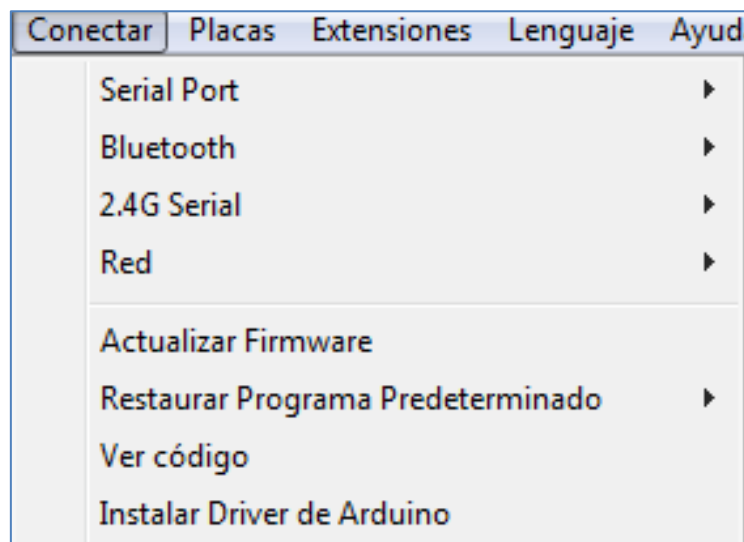
- *Recuperar borrado*: Restaura o borrado anteriormente
 - *Hide stage layout*: Oculta o escenario para facer o área de edición de scripts máis grande.
 - *Escenario pequeno*: Fai o escenario máis pequeno, aumentando o área de programación.
-

Divertíndome con mBot: Guía de manexo e programación

- *Modo turbo:* Acelera de velocidade de procesamento do programa na simulación scratch.
- *Modo Arduino:* Convertir o programa de mBlock a Arduino programa e súbeo á placa principal Arduino para realizar a operación fóra de liña.

2.3. Conectar

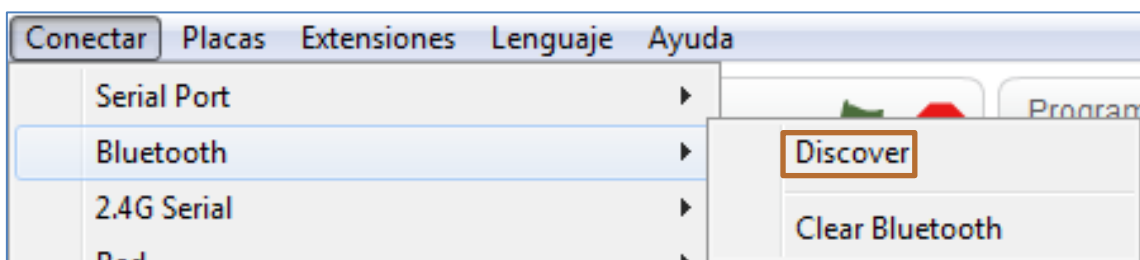
O robot mBot soporta a programación inarámica, e iso significa que non temos que ter o robot conectado continuamente ao PC para podelo programar en Scratch. Con mBot dispoñemos de varias opcións de comunicación inarámica: Bluetooth ou 2.4GHz e con calquera delas (non ambas á vez) podemos programar o mBot. Con 2.4GHz, o robot sincronízase automaticamente co ordenador sen necesidade de "buscar dispositivo..." evitando deste modo interferencias con outros mBots, cando se utilizan varios á vez nun aula. Tamén podemos descargar o programa ao controlador directamente, por comunicación inarámica.



Opcións da pestana Conectar

As opcións da pestana "Conectar" son:

- *Serial Port:* Conecta o mBot ao puerto COM onde está instalado o driver USB-SERIAL CH340
- *Bluetooth:* Descubre o Bluetooth para conectar e borra a conexión bluetooth. Débese desconectar o puerto anterior. Primeiro debemos descubrir o módulo bluetooth que se chamará Makeblock e despois conectarlo. Se o robot se conecta por bluetooth (dispón do módulo BT) podes acoplalo ao mBot cada vez que queiramos usalo desde o móvil ou tablet.



Divertíndome con mBot: Guía de manexo e programación

- **2.4G Serial:** Conéctase por 2.4G serial conectando primeiro o pendriver 2.4G no PC e sincronizándoo despois co módulo mBot. Tra-la vinculación con éxito, faise clic en conectar. O 2.4G permíteche usalo remotamente desde o PC dunha forma sinxela e en tempo real mentras vas facendo cambios en Scratch. Tamén poderás usar esta versión do robot (2.4G) co cable USB incorporado e co mando de control remoto (incluído en ambas versións) cuxas teclas se poden programar tamén desde Scratch. A versión mBot 2.4G non leva bluetooth e polo tanto, con ela non podemos manexar o robot desde o móvil.

A tecnoloxía 2.4G é a mesma que utilizan os ratos e teclados inarámicos para conectarse co PC. Dispoñemos de dous modos 2.4G e, para cambiar entre un modo ou outro, simplemente debemos premer o botón no módulo de 2,4G.

Modo lento intermitente: Cando o indicador 2.4G parpadea lentamente, significa que está en modo lento intermitente. Neste modo, o módulo 2.4G ten función de memoria e lembra o adaptador 2.4G ao que se conectou antes, de modo que, non se conectará a calquera novo adaptador automaticamente.

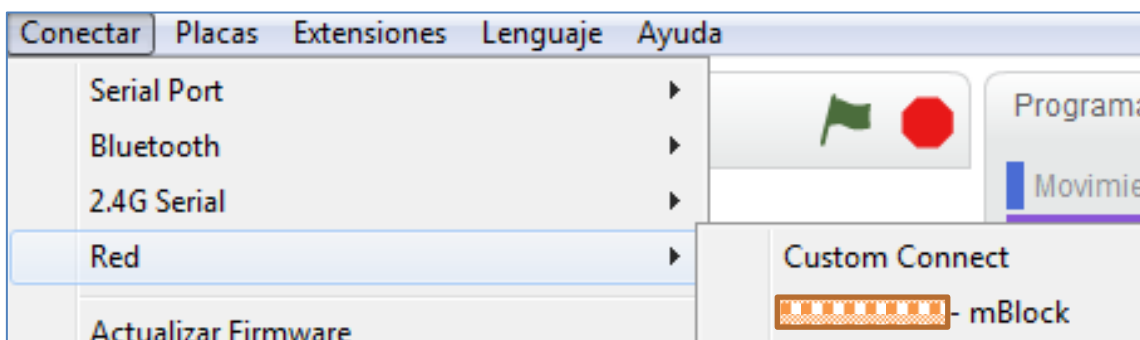
Modo rápido intermitente: Cando o indicador parpadea 2.4G con velocidade rápida, significa que está en modo rápido de parpadeo. Neste modo, o módulo 2.4G conectarase automaticamente ao adaptador 2.4G que está prendido. Sen función de memoria neste modo.

A conexión inarámica do módulo 2.4G trae un pincho USB (non necesita ningún driver xa que o ordenador interprétao como un rato inarámico). Cada pincho va asociado a un robot (ou mellor dito: A placa e o pincho 2.4G están emparellados e se o led da mochila parpadea é que perdeu conexión).

En caso de que o programa que realices con Scratch oqueiras deixar grabado permanentemente na placa do robot, deberás usar o cable USB. Isto é así para que o Scratch poida grabalo na placa que é un Arduino UN "tuneado"

Importante: *Decátate que o mBot só pode ser de 2 tipos. A saber: Bluetooth ou 2.4G, pero nunca á vez.*

- **Red:** Descarga o programa por conexión directa inarámica.

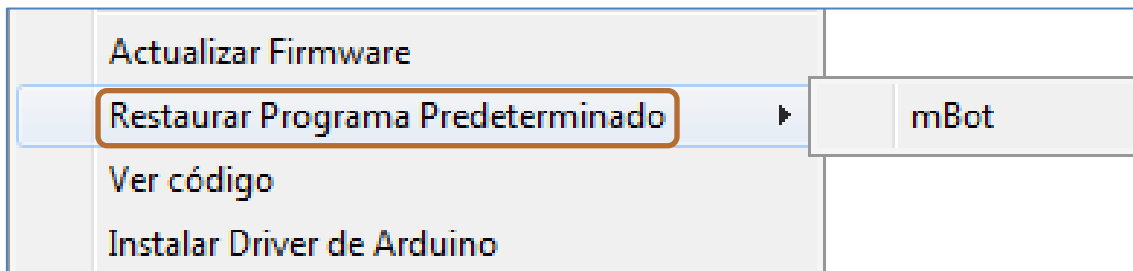


Un *firmware* é un programa informático (con lóxica de baixo nivel) que maneja o hardware do robot. Aquí, é un programa especial que se carga na placa para

Divertíndome con mBot: Guía de manexo e programación

poder utilizar a App para controlar o robot desde iPhone ou Android via bluetooth, ou para poder usar mBlock.

- *Actualizar Firmware:* Desta forma o noso robot entenderíase perfectamente con mBlock. Débese esperar a que a actualización finalice para non perxudicar a placa de mBot. Esta opción úsase coas placas ou shields de Makeblock.
- *Restaurar Programa Predeterminado:* Restaura a configuración de fábrica de forma manual. As opcións que se abren dependerán do robot conectado: mBot, Starter ou mBot Ranger. Debemos esperar a que a restauración se complete Como facelo?:



En primeiro lugar, asegúrese de que descargou a última versión do mBlock.

En segundo lugar, encenda o mBot coas baterías conectadas e conectar mBot ao ordenador co cable USB.

En terceiro lugar, abra o software mBlock. A continuación, seleccione o porto serie correcto no menú "Conectar" e elixa a placa mBot no menú "Placas".

En cuarto lugar, faga clic no "*Restaurar Programa Predeterminado*" no menú "Conectar" e espere ata que teña éxito 100%.

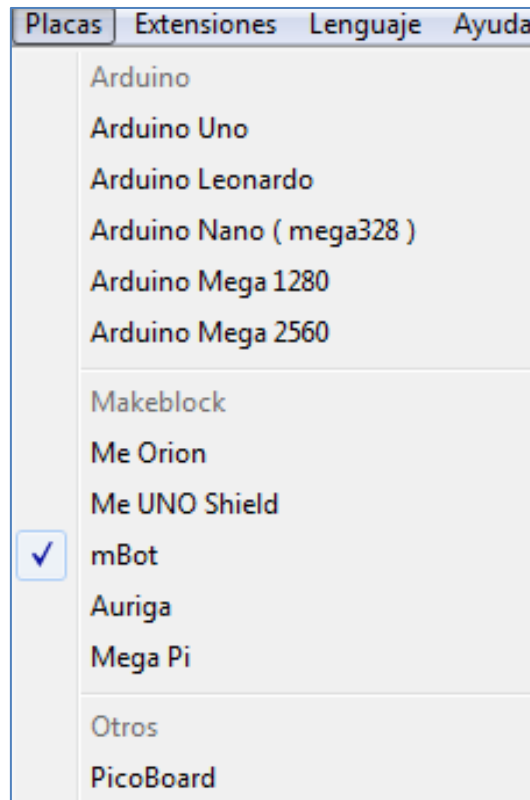
- *Ver código:* Abre unha pantalla lateral dereita co código do programa.
- *Instalar Driver de Arduino:* Instala os drivers da placa.

2.4. Placas

O software mBlock non só é aplicable ao robot mBot. Con el podemos programar outras placas electrónicas. Esta pestana clasifícaaas en tres bloques: Arduino, Makeblock e Outros.

Dentro do conxunto Arduino podemos conectar a placa Arduino Un, Leonardo e Arduino Mega 1280 e 2560. Na opción das placas da casa Mackeblock atopámonos coa placa mCore (mBot), Orion e Auriga, o shield para Arduino Un e a Mega Pi. Finalmente, no bloque "Outros" podemos traballar coa coñecida PicoBoard. Todas elas son opcións que nos ofrece a pestaa "Placas".

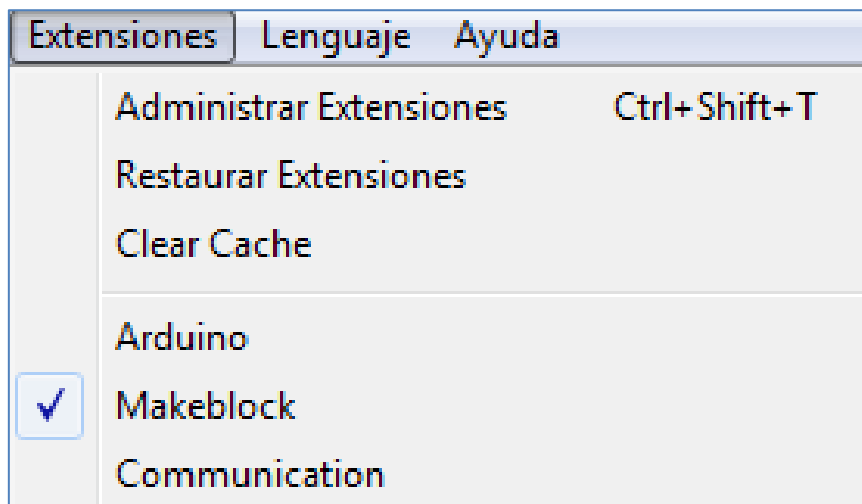
Obviamente, debemos elixir que opción das posibles programaremos. Isto faise desde a pestana Placas.



Pestana "Placas" na Versión 3.3.1 de mBlock

2.5. Extensiones

As opcións da pestana "Extensiones" amósanse na seguinte imaxe:

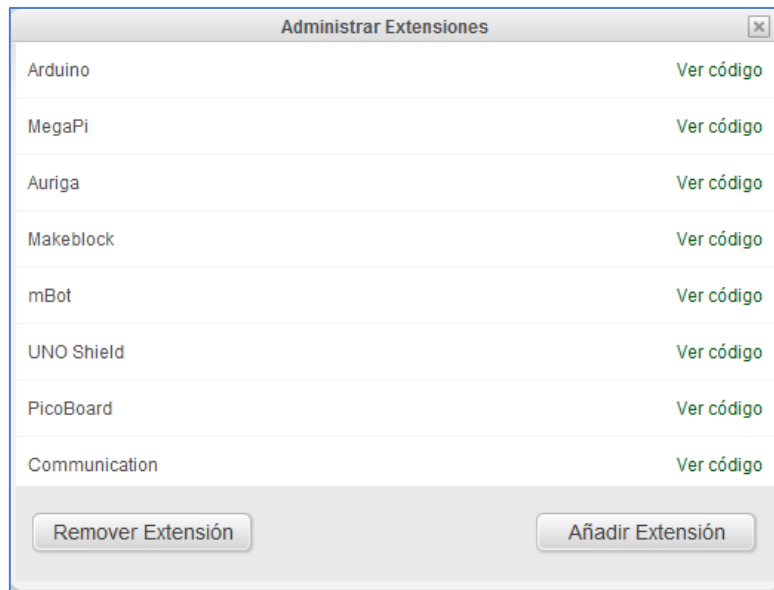


Opciones de la pestaña Extensiones

- *Administrar Extensiones*: Amósanos as extensións que temos instaladas. Podemos sumar máis extensións ou borrarlas. Para facelo, debemos seguir os pasos que se explican no seguinte link:

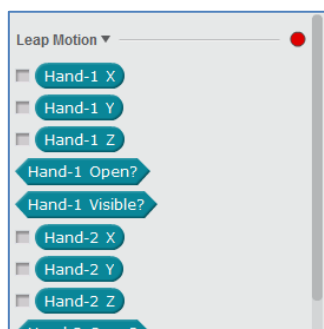
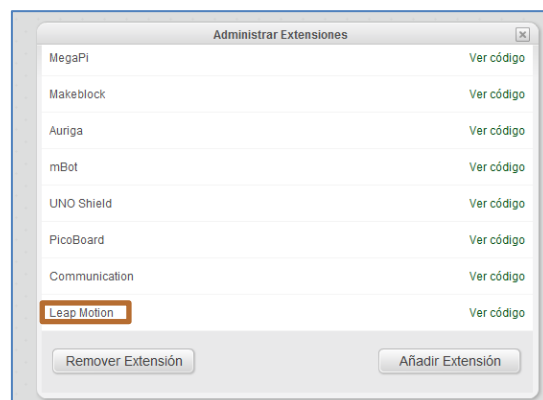
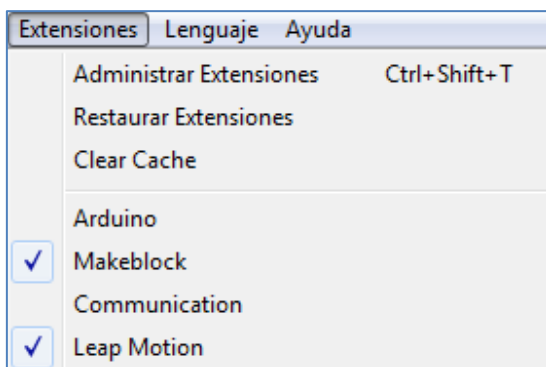
<http://forum.makeblock.cc/t/how-to-add-an-extension-for-mblock/2280>

Diviertíndome con mBot: Guía de manexo e programación

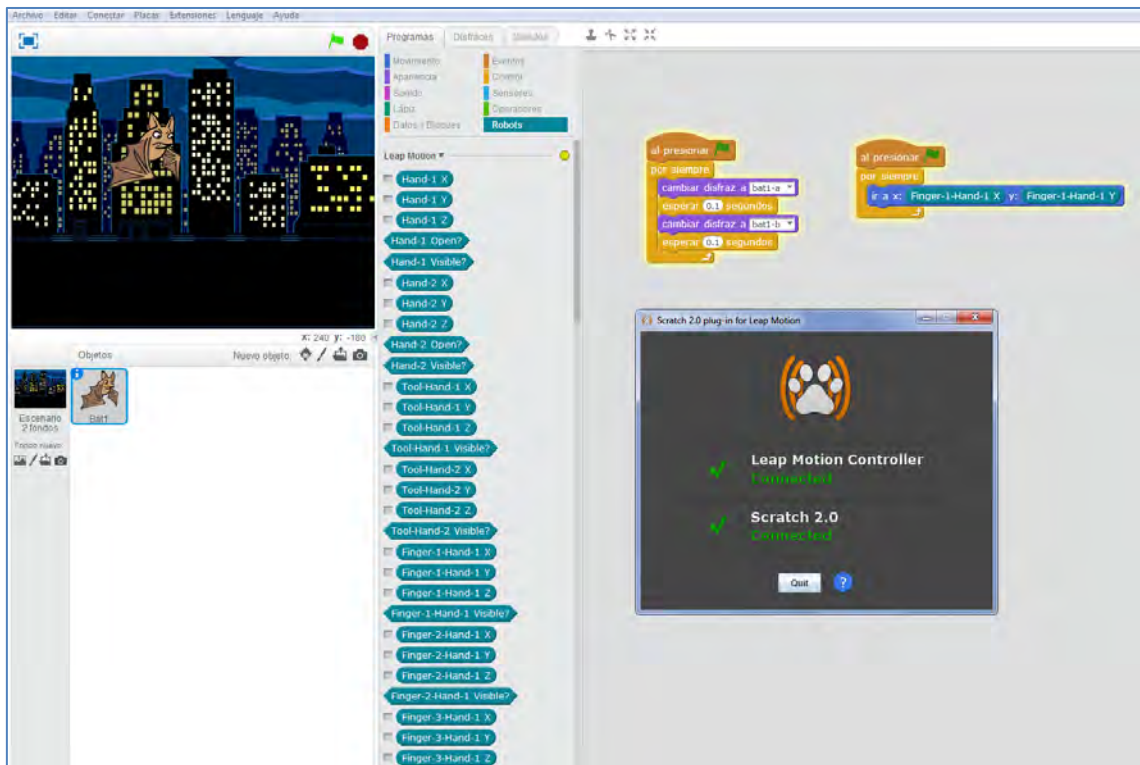


A opción que elixa na extensión afectará aos comandos amosados no bloque Robot do programa. Polo tanto, cada opción amosará os seus comandos específicos no bloque Robot do mBlock.

No mercado hai dispositivos que presentan unha extensión experimental co software Scratch2.0 e que pueden traballar con mBlock. No caso do dispositivo Leap Motion. Este dispositivo de tres pequenos infravermellos require dun plugin, dispoñible na súa casa App Home, que fai posible a detección do mesmo nun entorno Scratch. Tras lanzalo, só debemos engadir o arquivo *LeapMotion.json* ás extensións do software mBlock e conseguiremos que o programa recoñeza os comandos do Leap Motion e poidamos programalo. Nas seguintes imaxes podemos ver que se incluiu a extensión do Leap Motion:



Divertíndome con mBot: Guía de manejo e programación



Ejemplo con Leap Motion

- *Arduino*: Incluye bloques compatibles coa plataforma Arduino



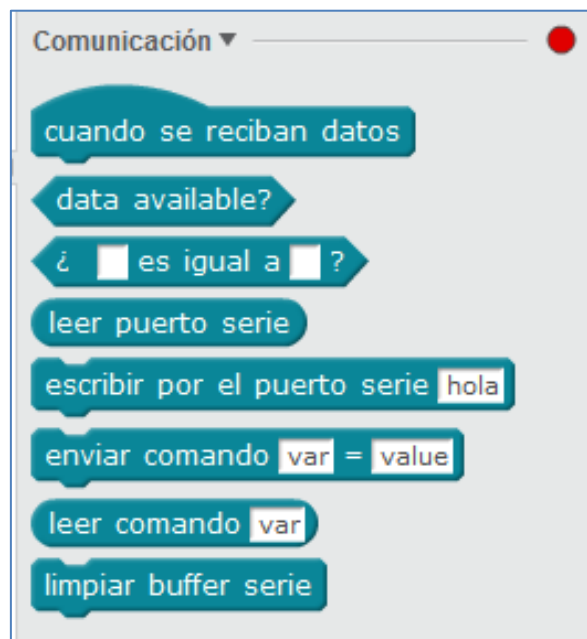
- *Makeblock*: Me Orion, Mega Pi, Auriga, Me Un Shield e mBot; placas base específicas de Makeblock

Divertíndome con mBot: Guía de manexo e programación

- *PicoBoard*: Control e testeio da placa PicoBoard desde scratch (deslizador, botón, sensor de luz, sensor de son, etc)

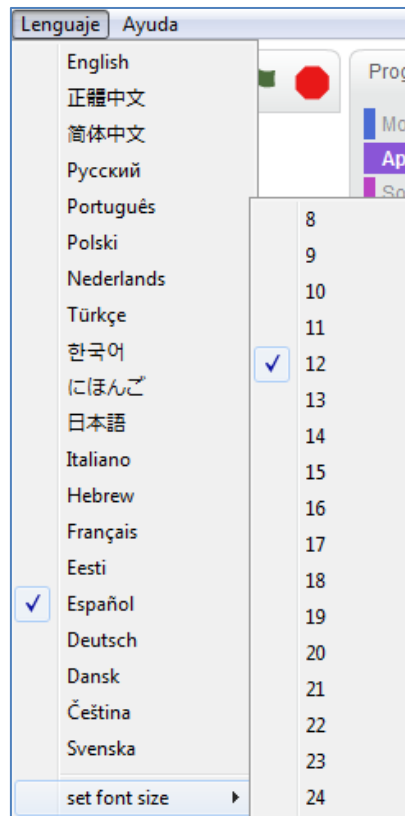


- *Comunicación*: Proporciona a función da comunicación LAN.



2.6. Linguaxe

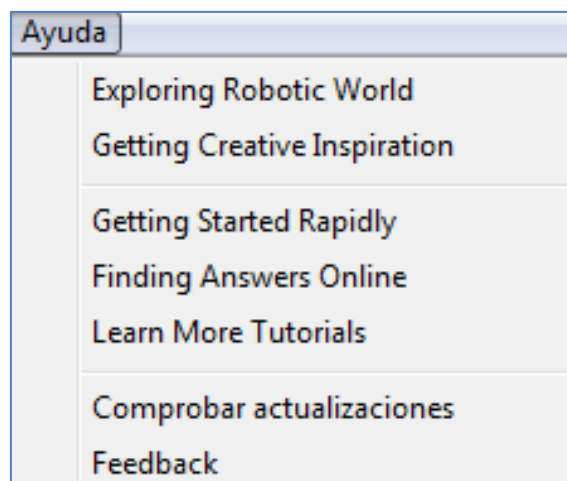
Nesta pestana podemos elixir o idioma e o tamaño da letra ou fonte do mBlock. Na seguinte imaxe seleccionouse o español e tamaño 12:



Opcións da pestana Linguaxe

2.7. Axuda

Con esta pestana podemos informar dalgún erro ou bug ao equipo que actualiza e perfeciona o software mBlock. Tamén podemos acceder ao foro e tutoriais, así como, comprobar se hai actualizacións do software.



Opcións da pestana Axuda

3. Comandos dos diferentes bloques en mBlock

3.1. Movimiento

The image shows the 'Movimiento' (Movement) category in the mBlock software. It features a sidebar with various block categories and a main area displaying several movement-related blocks. The sidebar categories are: Movimiento (blue), Apariencia (purple), Sonido (pink), Lápiz (green), Data&Blocks (orange), Eventos (brown), Control (yellow), Sensores (light blue), Operadores (light green), and Robots (teal). The main area contains the following blocks:

- mover 10 pasos**: A block to move the robot a specified number of steps.
- girar 15 grados** (left and right): Blocks to rotate the robot by a specified angle in either direction.
- apuntar en dirección 90**: A block to set the robot's heading to a specific direction.
- apuntar hacia**: A block to point the robot towards a specific target.
- ir a x: 13 y: -7**: A block to move the robot to specific x and y coordinates.
- ir a puntero del ratón**: A block to move the robot to the mouse cursor's position.
- deslizar en 1 segs a x: 13 y: -7**: A block to move the robot smoothly to specific coordinates over a set time.
- cambiar x por 10**: A block to change the x coordinate of the robot's position.
- fijar x a 0**: A block to set the x coordinate to a specific value.
- cambiar y por 10**: A block to change the y coordinate of the robot's position.
- fijar y a 0**: A block to set the y coordinate to a specific value.
- rebotar si toca un borde**: A block to make the robot bounce off a wall when it touches one.
- fijar estilo de rotación izquierda-derecha**: A block to set the rotation style to either 'left' or 'right'.

At the bottom, there are three checkboxes for the robot's position and direction:

- posición x
- posición y
- dirección

3.2. Apariencia

The screenshot displays the mBlock 2 software interface. On the left sidebar, the 'Apariencia' (Appearance) category is selected, highlighted in purple. Other categories visible include Movimiento, Sonido, Lápiz, Data&Blocks, Eventos, Control, Sensores, Operadores, and Robots. The main workspace contains the following blocks:

- decir Hello! por 2 segundos
- decir Hello!
- pensar Hmm... por 2 segundos
- pensar Hmm...
- mostrar
- esconder
- cambiar disfraz a Panda-b
- siguiente disfraz
- cambiar fondo a fondo1
- cambiar efecto color por 25
- establecer efecto color a 0
- quitar efectos gráficos
- cambiar tamaño por 10
- fijar tamaño a 100 %
- enviar al frente
- ir 1 capas hacia atrás
- # de disfraz
- nombre de fondo
- tamaño

3.3. Son

Movimiento

Apariencia

Sonido

Lápiz

Data&Blocks

Eventos

Control

Sensores

Operadores

Robots

tocar sonido **eat**

tocar sonido **eat** y esperar

detener todos los sonidos

tocar tambor **1** durante **0.25** pulsos

silencio por **0.25** pulsos

tocar nota **60** durante **0.5** pulsos

fijar instrumento a **1**

cambiar volumen por **-10**

fijar volumen a **100** %

volumen

cambiar tempo por **20**

fijar tempo a **60** ppm

tempo

3.4. Lápiz

The image shows a screenshot of the mBot programming environment. At the top, a palette of categories is visible, with 'Lápiz' (Pen) highlighted in green. Below the palette, a sequence of code blocks is shown, all connected to the 'Lápiz' category. The blocks are:

- borrar
- sellar
- bajar lápiz
- subir lápiz
- fijar color de lápiz a
- cambiar color del lápiz por 10
- fijar color de lápiz a 0
- cambiar intensidad de lápiz por 10
- fijar intensidad de lápiz a 50
- cambiar tamaño de lápiz por 1
- fijar tamaño de lápiz a 1

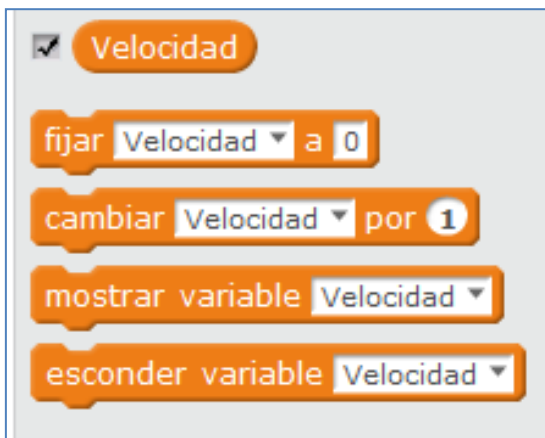
Divertíndome con mBot: Guía de manejo e programación

3.5. Datos e Bloques

Podemos crear variables, listas de datos e bloques dentro do programa



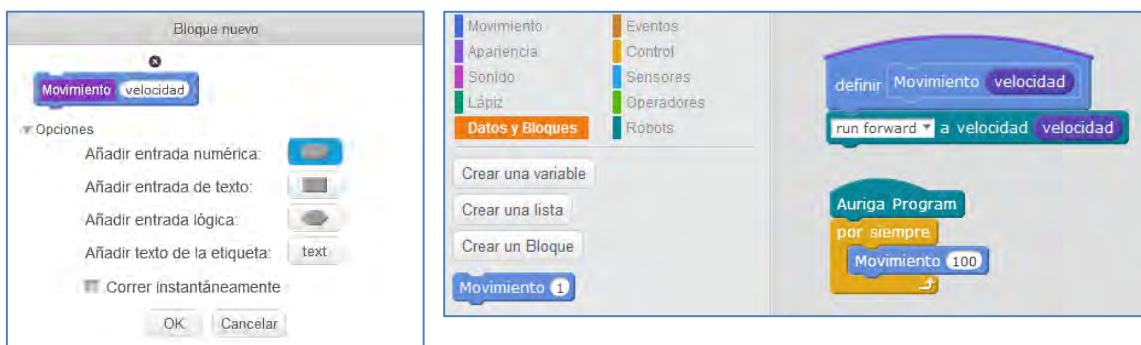
Variables:



Listas:



Bloques: primeiro creamos o bloque e despois xa podemos definir que realiza e lanzalo cando queiramos.



3.6. Eventos

The screenshot shows the 'Eventos' (Events) category selected in the top menu. Below the menu, several event blocks are displayed:

- al presionar
- al presionar tecla
- cuando tecla es liberada
- al clickear este objeto
- cuando el fondo cambie a
- cuando el sea >
- al recibir
- enviar
- enviar y esperar

3.7. Control

The screenshot displays the 'Control' category in the mBlock 2 software. At the top, a menu lists several categories: Movimiento, Apariencia, Sonido, Lápiz, Data&Blocks, Eventos, Control (highlighted in yellow), Sensores, Operadores, and Robots. Below the menu, the following blocks are visible:

- esperar 1 segundos
- repetir 10
- por siempre
- si entonces
- si entonces si no
- esperar hasta que
- repetir hasta que
- detener todos
- cuando comience como clon
- crear clon de este objeto
- borrar este clon

3.8. Sensores

The image shows the 'Sensores' (Sensors) category selected in the block palette. The main workspace contains the following blocks:

- ¿tocando ?
- ¿tocando el color ?
- ¿color tocando ?
- distancia a
- preguntar What's your name? y esperar
- respuesta
- ¿tecla espacio presionada?
- ¿ratón presionado?
- posición x del ratón
- posición y del ratón
- intensidad del sonido
- movimiento del video en este objeto
- video encendido
- fijar transparencia de video a 50 %
- cronómetro
- reiniciar cronómetro
- posición x de M-Panda
- minuto de ahora
- días desde el 2000

3.9. Operadores

Menú de Operadores:

- Movimiento
- Apariencia
- Sonido
- Lápiz
- Datos y Bloques
- Eventos
- Control
- Sensores
- Operadores**
- Robots

Operadores matemáticos:

- + (suma)
- (resta)
- * (multiplicación)
- / (división)

Operador de número al azar:

número al azar entre 1 y 10

Operadores de comparación:

- < (menor que)
- = (igual a)
- > (mayor que)

Operadores lógicos:

- y (AND)
- o (OR)
- no (NOT)

Operadores de texto y números:

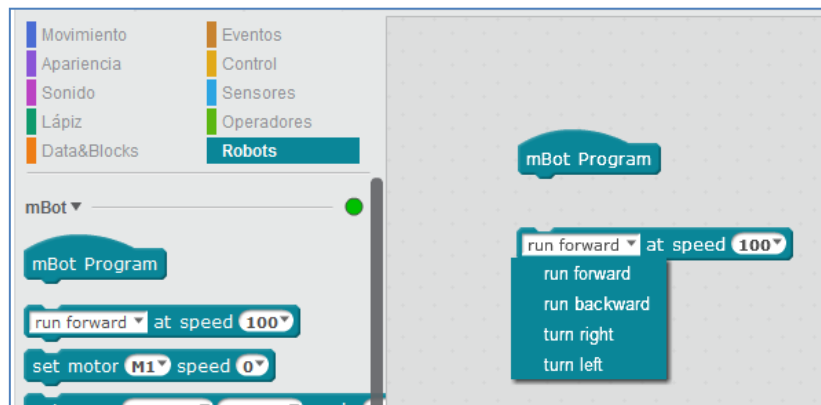
- unir hello world
- letra 1 de world
- longitud de world
- cast 1 to string
- mod
- redondear
- raíz cuadrada de 9

Divertíndome con mBot: Guía de manexo e programación

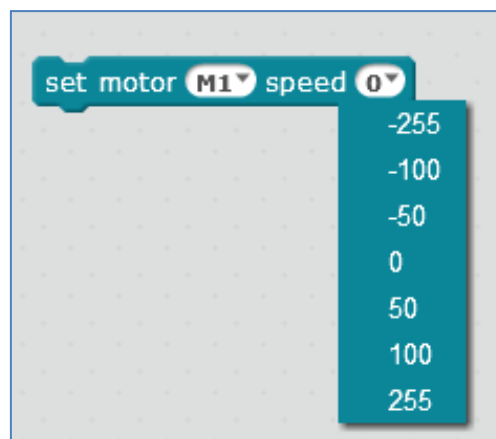
3.10. Robots

Bloque específico de comandos para as placas que se conectan co software mBlock. Falaremos dos comandos para o mBot, os cales a casa Makeblock amplía e actualiza con frecuencia de cara a novos sensores.

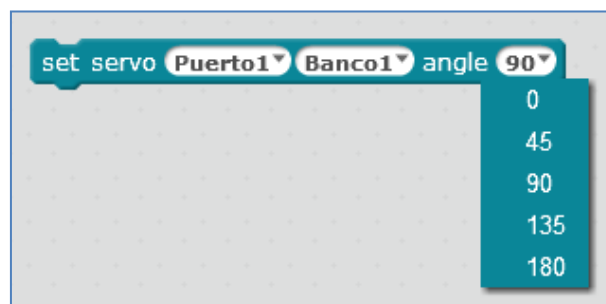
Neste bloque interézanos especialmente o comando "mBot Program". De feito, un preámbulo para poder grabar un programa na placa será cambiar o bloque da bandeiriña verde pola sentença "mBot Program" e, dese modo, todo o que colgue deste comando, executarase sempre que prendamos o mBot. Para grabalo na placa, debemos ter conectado o mBot mediante o cable USB, facer clic co botón dereito sobre este bloque (ou *Editar > Modo Arduino*) e seleccionar "Upload to Arduino".



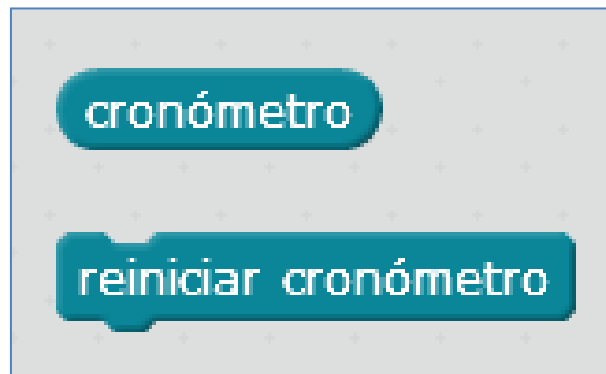
1. Dous motores M1 e M2 con diferentes velocidades:



2. Catro portos con 2 Bancos e diferentes ángulos



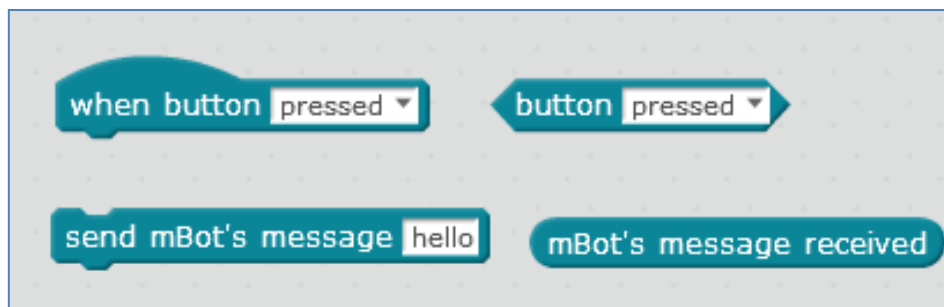
3. Cronómetro



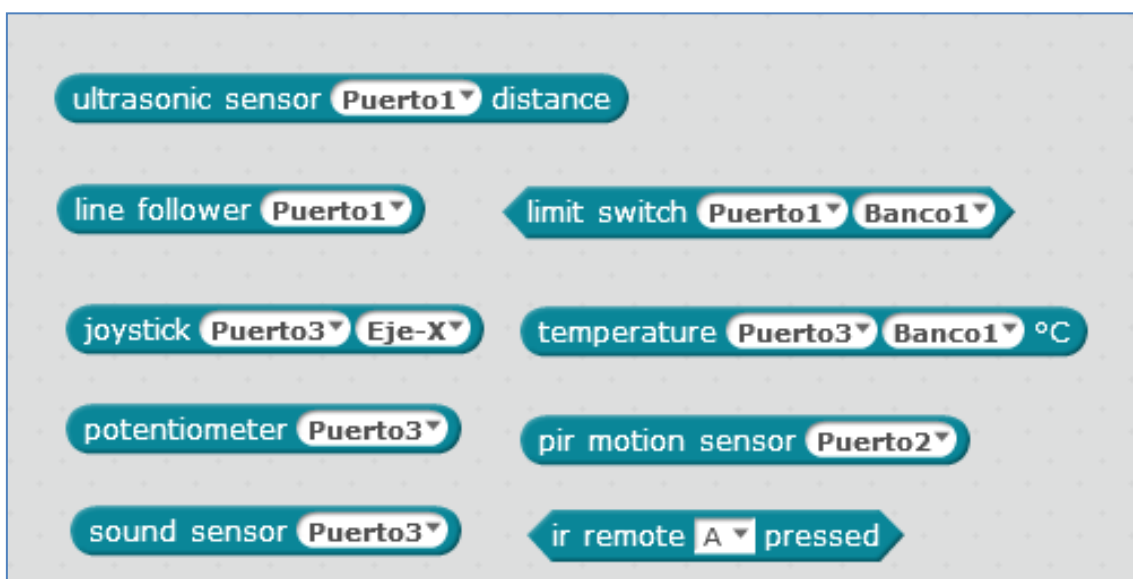
4. LED RGB



5. Botón e mensaxes



6. Sensores e Control Remoto



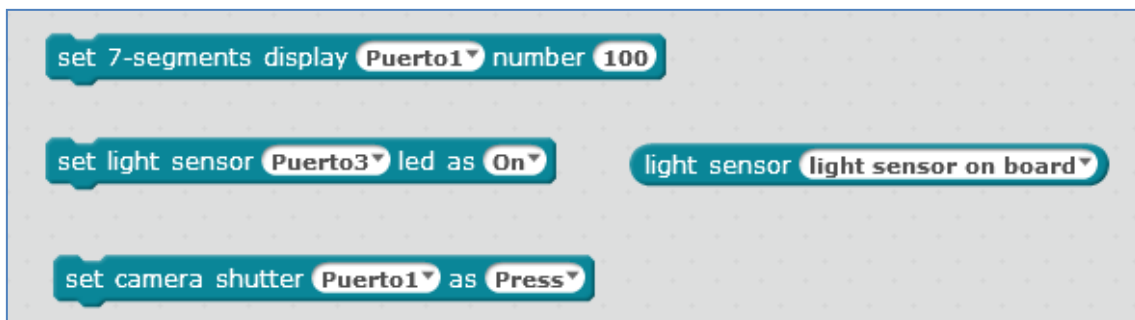
Na imaxe anterior vemos que dispoñemos de comandos para os seguintes sensores: Sensor de ultrasóns, seguidor de liña, sensor de temperatura, sensor de

Divertíndome con mBot: Guía de manexo e programación

son e sensor PIR. Tamén podemos programar o pulsador, un potenciómetro, un joystick e incluso, programar o robot de forma remota co seu mando especificando nós que debe hacer cada tecla do mando a distancia.

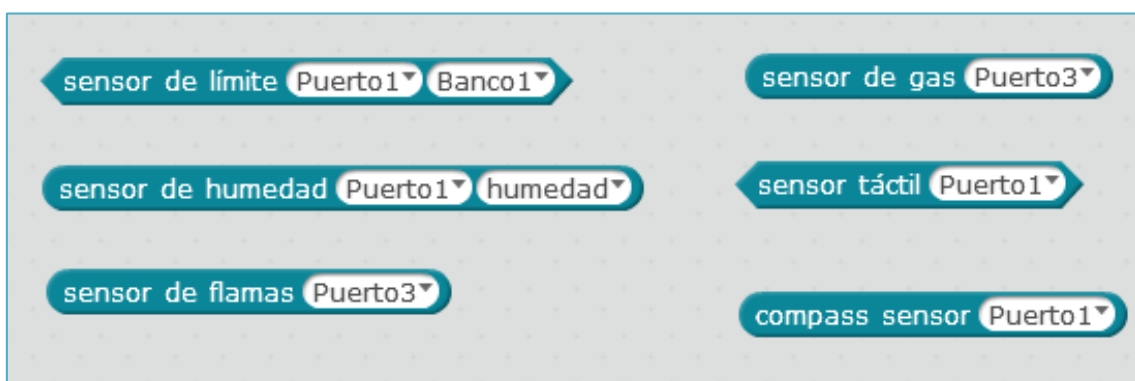
Os sensores PIR utilízanse para detectar o movemento dos animais / humanos a unha distancia aproximada de 6 m. Polo tanto, se alguén se move nese rango, as saídas dos sensores serán HIGH sobre o seu pin SIG. Cun potenciómetro soldado na súa placa de circuito, podemos axustar o seu rango de detección para as nosas necesidades. Algunhas ideas para a súa aplicación poden ser: detección de movemento, sistema ladrón-vixilancia, interruptor de automatización.

Nota: É posible que debamos esperar 10 segundos ata que o sensor comece a funcionar.

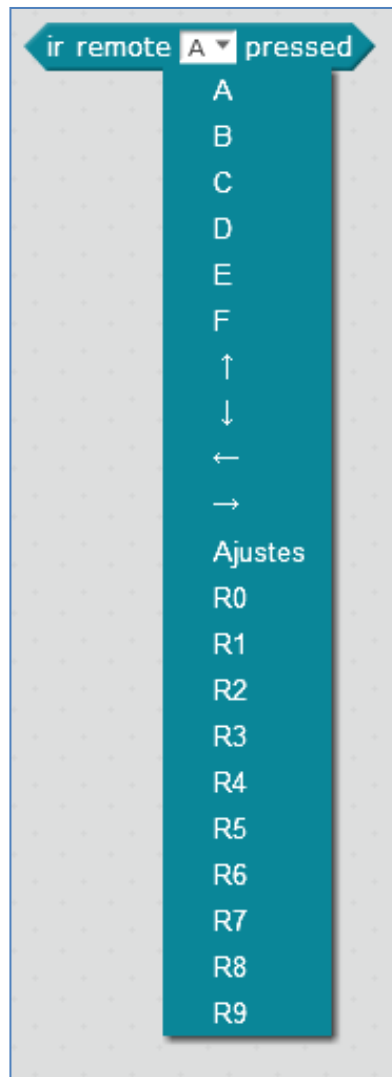


Tamén podemos programar o sensor de luz, o sensor de vídeo e o display 7 segmentos. Relativo ao sensor de luz, que está situado ao lado dos de IR, deteta a luz ambiente, pero, segundo a casa Makeblock, experimentaron que deteta a infravermella, polo que é sensible á calefacción falseando a súa medida.

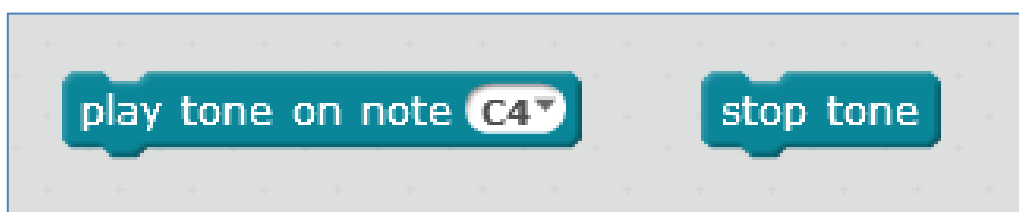
A casa Makeblock está incluíndo máis comandos para outros sensores que vai sacando ao mercado. A partir da versión 3.3.1 podemos utilizar os seguintes comandos para os sensores de gas, lapas, táctil, humidade, etc:



De forma remota, podemos controlar o noso robot, programando ao noso gusto cada unha das teclas do mando a distancia:



7. Música



8. Amosar

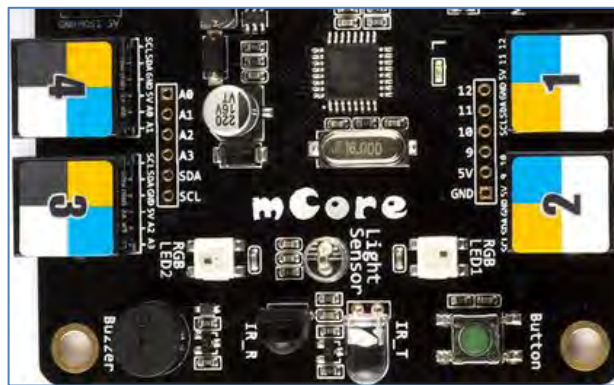


Divertíndome con mBot: Guía de manexo e programación

4. Exemplos de programación

A placa mCore del mBot pode programarse utilizando diferentes linguaxes de programación. De feito, non é máis que unha Arduino Uno, polo tanto, pode ser programada con Processing. Isto fai que poidamos sacarlle máis rendimento á placa xa que, en teoría, temos un dous en un. Aínda que neste apartado usaremos moitas veces scratch para programar diferentes módulos, eses mesmos compoñentes electrónicos poderían ser programados a través do IDE de Arduino e, ás veces, usaremos esta posibilidade.

Os módulos que pretendemos conectar á placa presentan e veñen clasificados pola súa cor ID. Esa cor debe corresponder coa cor do porto ao cal pretendemos conectalo. Por exemplo, na seguinte imaxe vemos que o porto 2 dispón de tres cores: amarelo, azul e branco. Pois ben, a el podreemos conectar calquera módulo cuxo RJ25 dispoña de, como mínimo, algún desas cores. Se o ID do módulo é negro, non poderíamos conectalo ao porto 2 (nin ao 1), pero si ao 3 ou 4.



4 portos de mCore

As cores ID que podemos atopar nos portos das diferentes placas de Makeblock son: Vermello (motores), Amarelo (interface dixital), Azul (interface dixital dual), Gris (Porto serie, bluetooth), Negro (interface analóxica e dual) e Branco (Porto I²C).



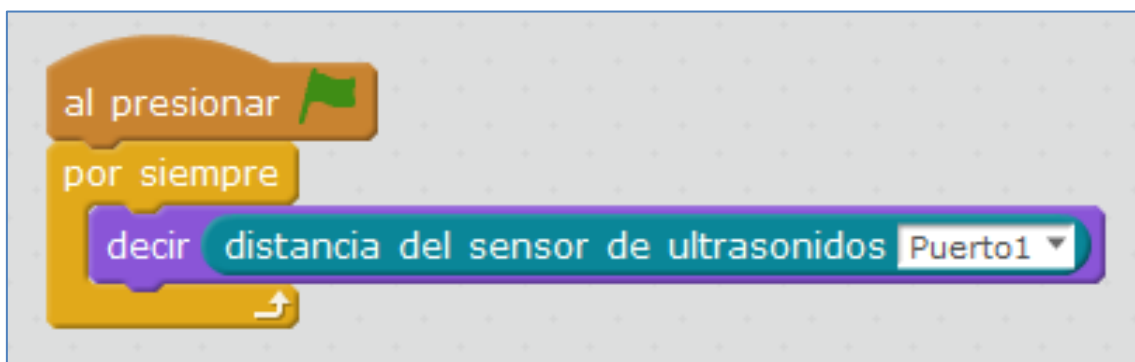
4.1. Módulo de ultrasóns

Un módulo de ultrasóns proporciónanos un dato numérico que se corresponde coa distancia entre o sensor e calquera obxecto que está en fronte del. Polo tanto, utilízase para medir distancias, logrando detetar obxectos que se atopan a 3 ou 4cm do sensor. A súa cor ID é amarelo e iso significa que pode conectalo a calquera dos catro portos dunha placa mCore do mBot.

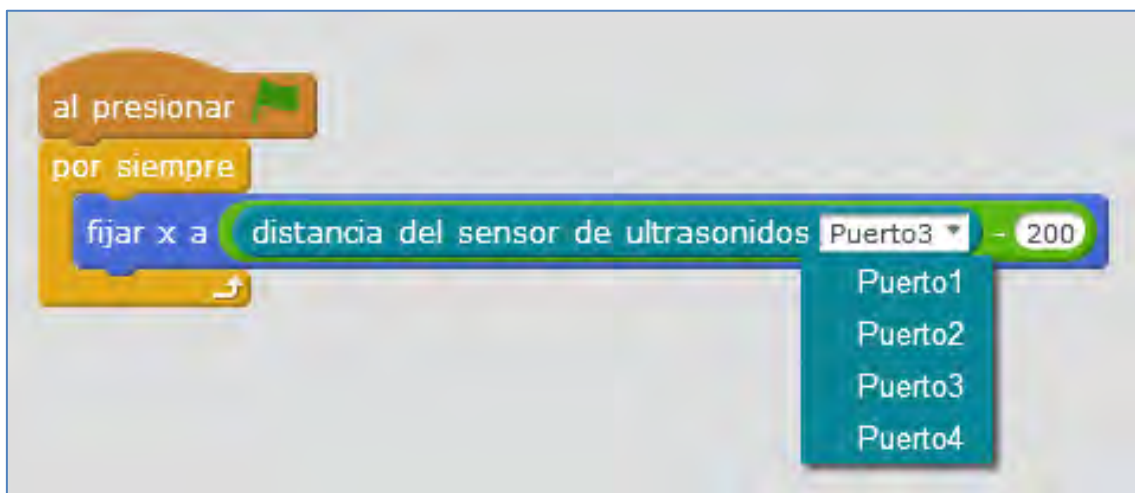


Módulo ultras5} s

Supofiamos que o conectamos a[p[rto 1 da placa mCore d[robot. U s^guinte script, æ [•ææ [• a distancia que mide } [escenario d[mBlock:



Tamén podemos controlar [movimiento d[rat[co sensor de ultras5} s. Un@ forma sería co s^guinte script:



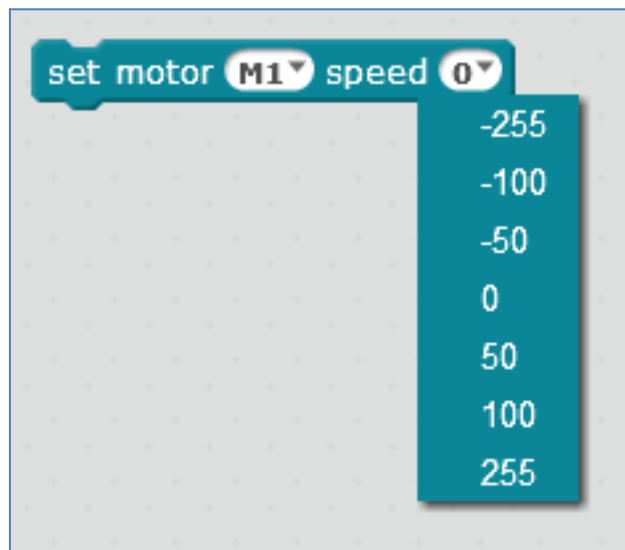
NOTA: O||o, haã que ver en qu p[rto dos 4 temos conectado [Ân[•o sensor de ultras5} s. U p[rto típico para [mBot s[e ser [p[rto 3.

4.2. Motores

U robot mBot & [] [] •^ de do~ s motores de continua de 6V a 200rpm. Pa placa, Á est^s motores å^} 5ca} •^ por M1 y M2, p[dando variar ææ} velocidad, á hora de programalos, desde 0 a 255. O valor mínimo, 0, é a velocidad máis lenta e 255 é a máis rápida. Os valores negativos farán que xire no outro sentido (rotación inversa).

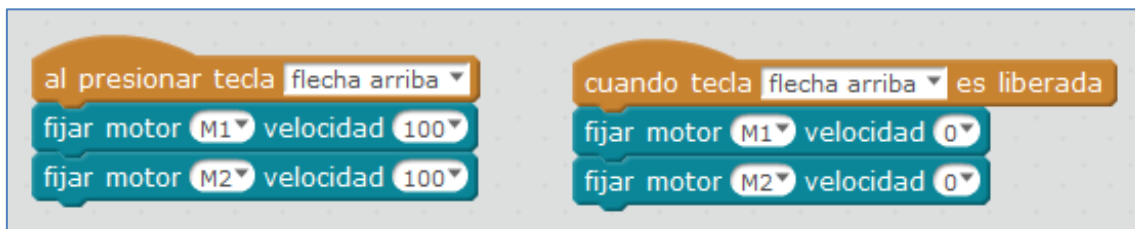


Motor DC 6V/200rpm



Podemos programar, por exemplo, o movementos dos motores a través do teclado do ordenador, ao premer ou liberar unha tecla. Debemos decidir qué ocorrerá cando premo a tecla “X” e que ocorrerá cando deixo de premer a tecla “X”.

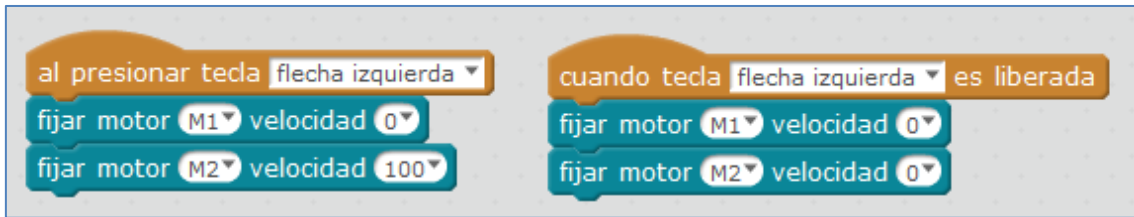
Por exemplo: Cando premo a tecla “flecha arriba” os dous motores comezan a funcionar cara adiante, cunha velocidade de 100, pero, cando deixo de premer esa tecla, pararán.



Para que xire cara a esquerda, debemos poñer el motor esquerdo a 0 (M1 ou M2, dependendo de como se conectaran²) e o dereito a unha velocidade (M1 ou M2) por exemplo de 100. Ao deixar de premer, debe parar. No exemplo que vemos na seguinte figura, M1 es él o motor esquerdo e M2 o dereito:

²En teoría, M1 é o esquerdo e M2 o dereito, pero podemos cambialos

Divertíndome con mBot: Guía de manexo e programación



Pódese mellorar o script anterior facendo que M1 xire no sentido contrario, con velocidade -100. O movemento de control completo nun coche sería:



Script para o movemento dos motores co teclado

Col tempo e os avatares que vaia pasando o robot, quizais debamos remprazar algunha peza dun motor do mBot. O seguinte tutorial amósanos cómo remprazar esas posibles pezas rotas ou defectuosas do motor:

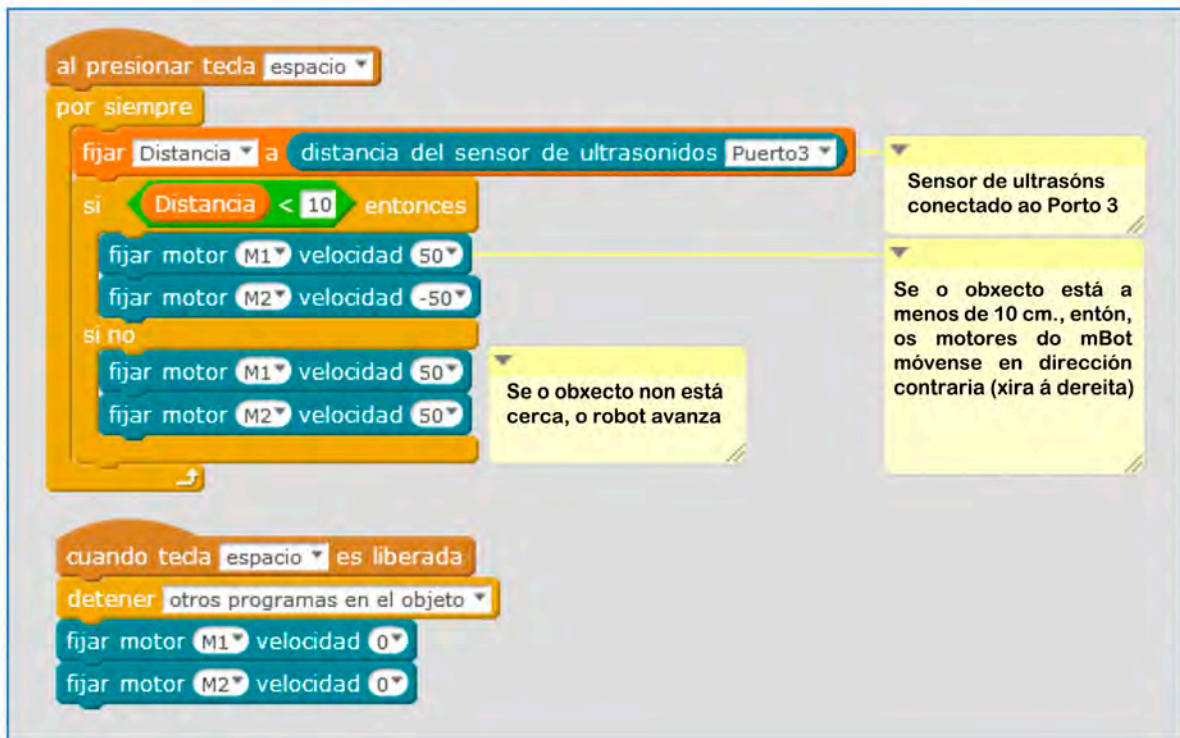
<http://learn.makeblock.cc/mbot-motor-shaft-broken/>

4.3. Motor e detección ultrasónica

Podemos unir os dous elementos que xa coñecemos, sensor de ultrasóns e motores, e intentar programar o seguinte reto:

Se o obxecto se atopa a menos de 10 cm do robot, debe xirar. En caso contrario, debe avanzar.

Divirtíndome con mBot: Guía de manexo e programación



Na imaxe anterior, traballamos sen cargar o programa no robot, é dicir, conectándoo por bluetooth ou a través do módulo 2.4G. Ás veces, co sensor de ultrasóns, pode ocorrer que ao cargar o programa, este non funcione correctamente. Nese caso, quizáis se deba engadir un tempo de espera en cada lectura para que logre captar a medida do sensor. No seguinte exemplo, decidín 200ms en cada lectura:



4.4. Diodos RGB da placa

A placa mCore dispón de dous RGB incrustados na mesma e que se denominan "led abordo". No seguinte exemplo, imos modificar o programa anterior

Divertíndome con mBot: Guía de manexo e programación

engadindo: o acendido dos LEDs en vermello (precaución), verde (adiante) e apagado dos LEDs.

al presionar tecla **espacio**

por siempre

fijar **Distancia** a **distancia del sensor de ultrasonidos Puerto3**

si **Distancia < 10** entonces

fijar led **led abordo** posición **todos** rojo **255** verde **0** azul **0**

fijar motor **M1** velocidad **50**

fijar motor **M2** velocidad **-50**

si no

fijar led **led abordo** posición **todos** rojo **0** verde **255** azul **0**

fijar motor **M1** velocidad **50**

fijar motor **M2** velocidad **50**

cuando tecla **espacio** es liberada

detener **otros programas en el objeto**

fijar led **led abordo** posición **todos** rojo **0** verde **0** azul **0**

fijar motor **M1** velocidad **0**

fijar motor **M2** velocidad **0**

Sensor de ultrasonidos conectado ao Porto 3

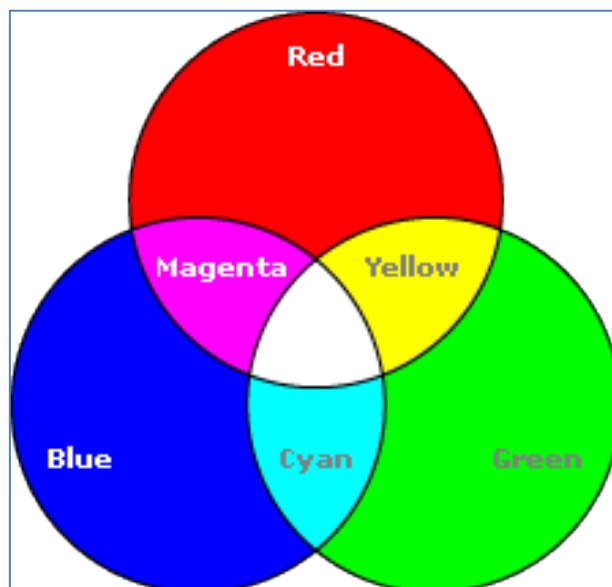
Se o obxecto está a menos de 10 cm., entón, os motores do mBot móvense en dirección contraria (xira á dereita). Os dous LEDs están en vermello.

Se o obxecto non está cerca, o robot avanza. Os dous LEDs están en verde

LEDs apagados

Os valores numéricos de cada cor do LED RGB están comprendidos entre 0 e 255. Combinando as tres cores podemos crear calquera cor do arcoí da vella. Para conseguir o branco, as tres cores (vermello, verde e azul), deben estar a 255.

Como se pode ver na seguinte imaxe, o amarelo conseguiríamolo combinando o vermello co verde (poñendo, por exemplo, o vermello a 125, o verde a 125 e o azul a 0):



Divertíndome con mBot: Guía de manexo e programación

4.5. Detetor de liña

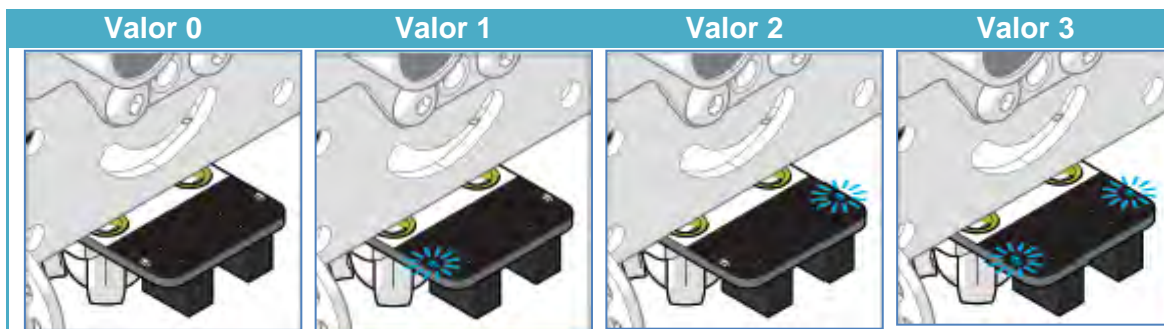
Este módulo sensor fai posible que o noso robot se mova ao longo dunha liña negra sobre o chan. No kit mBot sitúase na súa parte frontal funcionando *por reflexión*. No noso mBot, coma a cor do seu ID é azul, podémolo conectar aos portos 1, 2, 3 e 4. Nos exemplos que seguen, usaremos o Porto 2 do mBot.

O Módulo V2.2 está deseñado para detetar unha liña negra. Cada módulo ten dous partes: un *LED IR* que emite e un *fototransistor IR* que recibe o sinal. Grazas a eles poderás medir se estás sobre unha liña negra e seguira sobre un fondo branco ou viceversa. Os dous sensores dun módulo indican se están en "algo" negro ou en "algo" branco, e polo tanto, se poderían programar para ser usados como seguidores de liña branca. Ese "algo" branco ou "algo" negro non é, literalmente, a cor branca e negra, e explícome. O sensor, a toda cor brillante detéctao como branco, así como, unha cor mate escuro, detectarao como negro.

Á hora de programar o sensor, pódensenos dar 4 posibilidades. Eventualidades que o sensor da casa Makeblock, ten, implicitamente definidas cos seguintes valores numéricos: 0, 1, 2 e 3. Estes descríbense como segue:

- 0 é o valor numérico que representa que o sensor está totalmente enriba da liña negra.
- 3 é o valor numérico que representa que o sensor está totalmente enriba da liña branca.
- 1 e 2 son os valores numéricos que se asocian cando un lado do sensor está fóra da liña negra, mentras o outro lado está na cor negra.

As 4 posicións poden verse graficamente na seguinte imaxe:

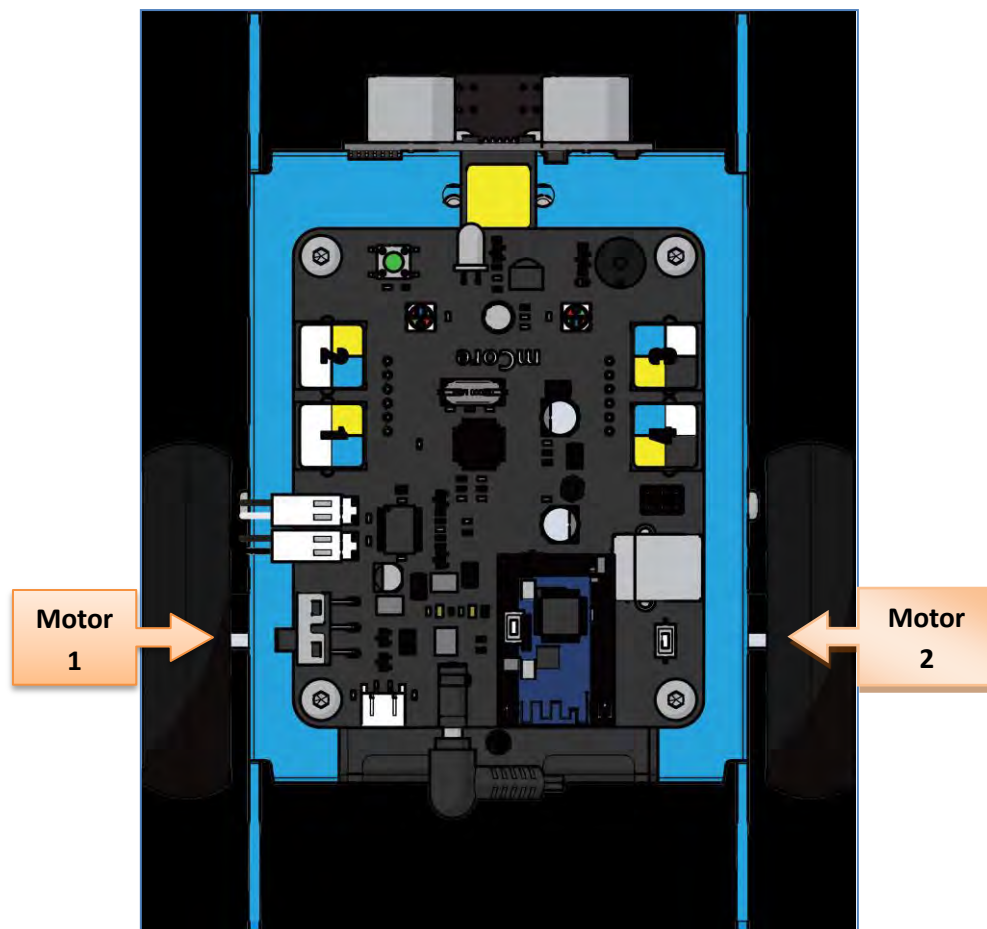


- A) Seguidor de liña negra: No seguinte exemplo, quixen que se accione ao premer a tecla espazo. Tras premer a tecla espazo, mBot executa o seguinte programa: testea se mBot ve completamente a liña escura e, se é así, móvese recto. Pola contra, testea se mBot está á esquerda (á dereita) e, se o está, xira á dereita (iesquerda) e fai que mBot regrese á liña de negra. Se mBot está lonxe da liña de negra, volve á liña de cor negra. Repite o ciclo ata que deixe de presionar a barra espaciadora.

Na súa programación, crearemos dous bloques (dereita e esquerda) con entradas numéricas que denoto pola variable "Velocidade":

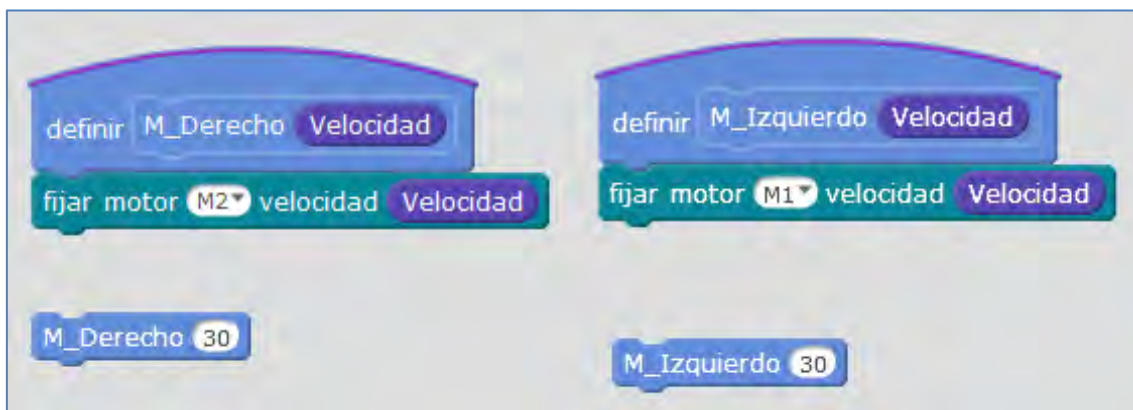
Divertíndome con mBot: Guía de manexo e programación

Véndoo frontalmente: **Supoñamos** (debemos comprobar as conexións dos motores do noso mBot) que a roda esquerda do mBot corresponde ao motor 1 e a roda dereita corresponde ao motor 2, tal e como se amosa na seguinte figura:



Robot mBot

Nestas condicións, cando executamos o Bloque "M_Esquerdo", execútanse os seus comandos. O valor 30 (-50, 0, 50, 100, 250³) no bloque "M_Esquerdo" e "M_Dereito", remprazará ao valor "Velocidade". Neste exemplo, as velocidades de rotación do motor 2 e do motor 1 serán de 30:



³En teoría, non debemos sobrepasar estas velocidades para non forzar os motores DC

Divertíndome con mBot: Guía de manexo e programación

Finalmente, o seguidor de liña amosaría o seguinte script: Ao premer a barra espaciadora (mantena pulsada), mBot comeza a xulgar o estado dos sensores de patrullaxe da liña (0, 1, 2, 3). A rotación do motor depende da desviación, que facilita o axuste da dirección de mBot. Podemos incluso incluír comandos de operación, como se ve no seguinte script:

The image shows a Scratch script for controlling an mBot. The script starts with a 'when green flag clicked' event, followed by a 'forever' loop. Inside the loop, the first block is 'set ValorSensorLinea to follow-lines Puerto2'. This is followed by four conditional blocks based on the value of 'ValorSensorLinea':

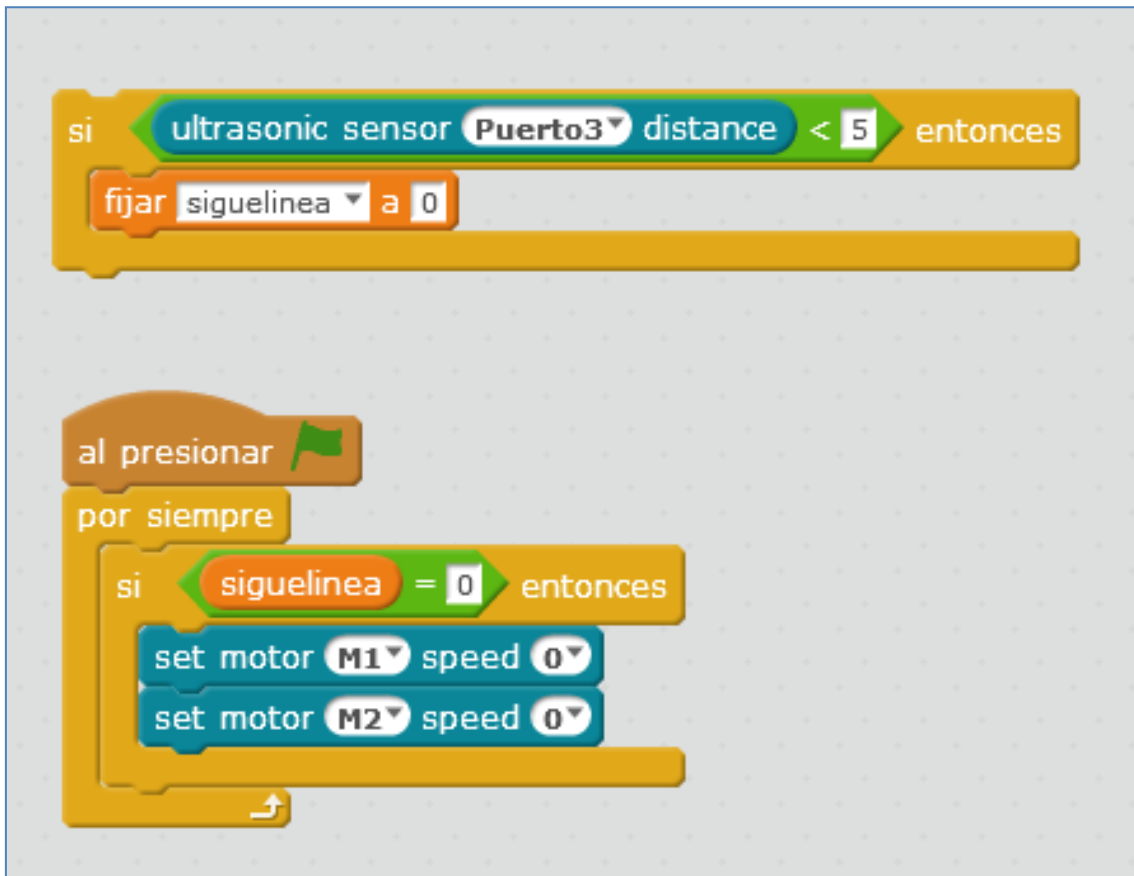
- if ValorSensorLinea = 0 then:** Set M_Izquierda to -100 and M_Derecha to -100. Callout: 'Está completamente fóra da liña branca. Debe volver'.
- if ValorSensorLinea = 1 then:** Set M_Izquierda to 225 and M_Derecha to 75. Callout: 'Debe xirar á dereita. O motor dereito desacelera e o esquerdo acelera'.
- if ValorSensorLinea = 2 then:** Set M_Izquierda to 75 and M_Derecha to 225. Callout: 'Debe xirar á esquerda. O motor esquerdo desacelera e o dereito acelera'.
- if ValorSensorLinea = 3 then:** Set M_Izquierda to 255 and M_Derecha to 255. Callout: 'Vai pola liña branca: os motores avanzaan.'

Ten en conta no segue-liñas que: se devolve un 0 é que vai por bo camiño, se devolve un 1 habería que xirar cara a esquerda, se devolve un 2 habería que xirar cara á dereita e se devolve un 3 é que se foi da liña negra, e ese caso, “o mellor” é que deas marcha atrás.

Poderíamos melloralo co sensor de ultrasóns, incluíndo outro condicional “Se... entón”, dentro do comando “Por sempre” anterior que pare o mBot antes de

Divertíndome con mBot: Guía de manexo e programación

que caia. Para iso podemos usar unha variable “segueliña” de modo que, se está a cero, pararase:



Poderíamos mellorar o programa anterior *engadindo pequenos tempos de espera* en cada movemento. É dicir, pequenos retardos de 0.1 segundos en cada movemento para que non se escape da liña se esta é fechada e lle dea tempo a xirar. Desesta forma, conseguiríamos que o robot se desviase menos veces da liña que queremos que siga.

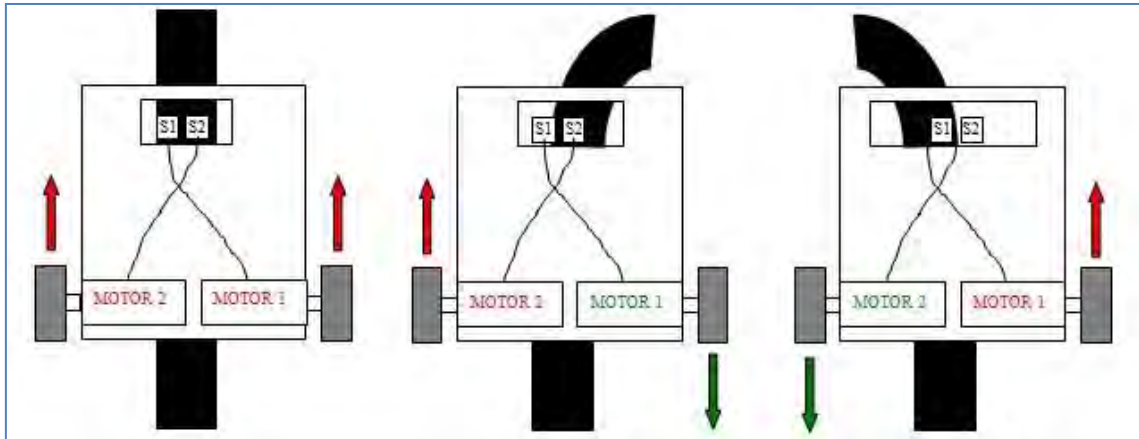
Nas imaxes seguintes se explica o funcionamento dun pequeno robot capaz de seguir unha liña negra sobre unha base branca. O noso robot consta de dous motores para mover, de maneira independente, as rodas traseiras e de dous sensores de liña que subministran unha voltaxe próxima a 0v (LOW) se a cor é negra e cercano a 5v (HIGH) se é branca.

Se os dous sensores están sobre a liña negra os dous motores xiran cara diante e o coche móvese en liña recta

Se o sensor 1 sáese da liña negra, o motor 1 móvese cara atrás e o vehículo xira cara á dereita.

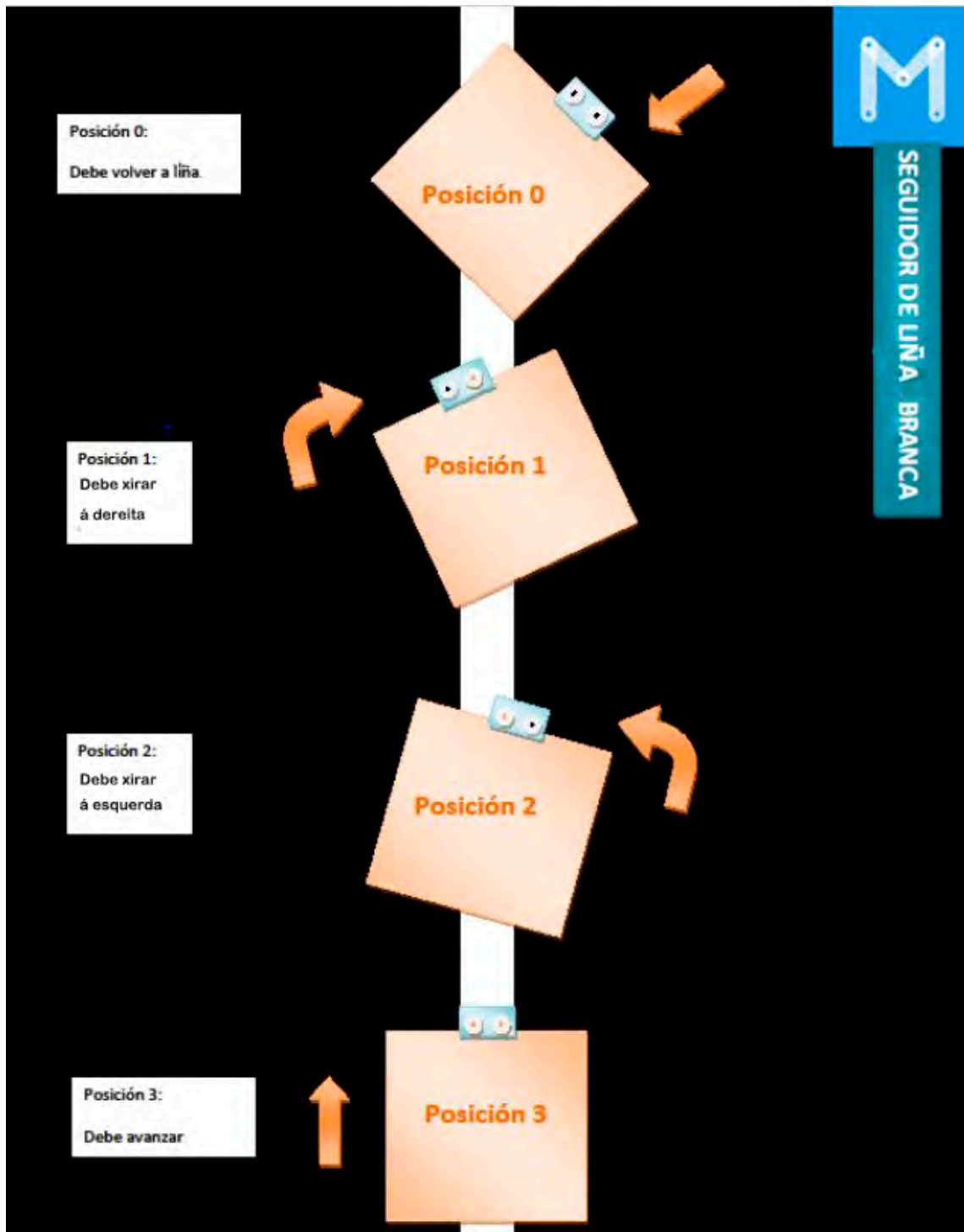
Se o sensor 2 se sae da liña negra, o motor 2 móvese cara atrás e o vehículo xira cara a esquerda.

Divertíndome con mBot: Guía de manexo e programación



Falta ter en conta unha última possibilidade, que o robot estea totalmente fóra da liña negra. A tarxeta mCore debe ler o estado dos sensores de liña e dar as ordes oportunas aos dous motores.

- B) Á hora de programar que mBot siga a liña branca, debemos tener en conta que posición nos devolve o sensor de línea cando detecta que está dentro ou fóra da liña branca. Estas posicións, poden verse representadas na seguinte imaxe:



Con pequenas modificacións no programa “seguidor de liña negra”, xa teríamos o seguidor de liña branca.

Divertíndome con mBot: Guía de manexo e programación

Supoñamos que as conexións dos motores M1 e M2 do noso mBot son as seguintes.

Visto de forma frontal: M1 é o Motor Esquerdo e M2 é o Motor Dereito.

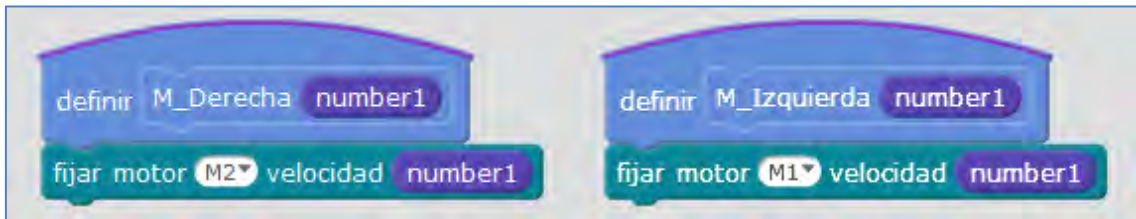


Decidín construír bloques para cada movemento de ambos motores: Motor dereito con M2 e motor esquerdo con M1. Cada bloque depende dunha variable numérica chamada “number1”:



Crear un bloque con entrada numérica (Datos e Bloques)

Divertíndome con mBot: Guía de manexo e programación



O programa final, que podemos utilizar para o noso mBot (neste caso, creouse sen engadir os tempos de espera ou retardos en cada movemento) é o seguinte:



Este script controla cada unha das posicións ou opcións que se poden dar. Graficamente, estas posicións simuláronse nas seguintes imaxes:



Posición 0 (fóra da liña branca)



Posición 1

Divertíndome con mBot: Guía de manejo e programación



Posición 2

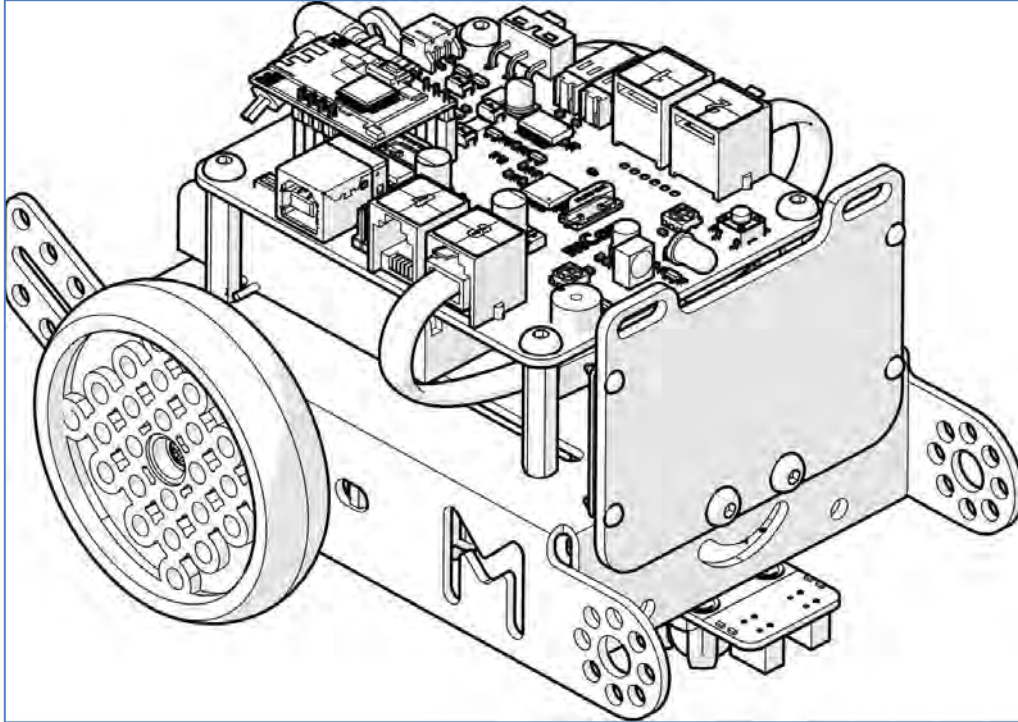


Posición 3 (Detecta blanco)

Divertíndome con mBot: Guía de manexo e programación

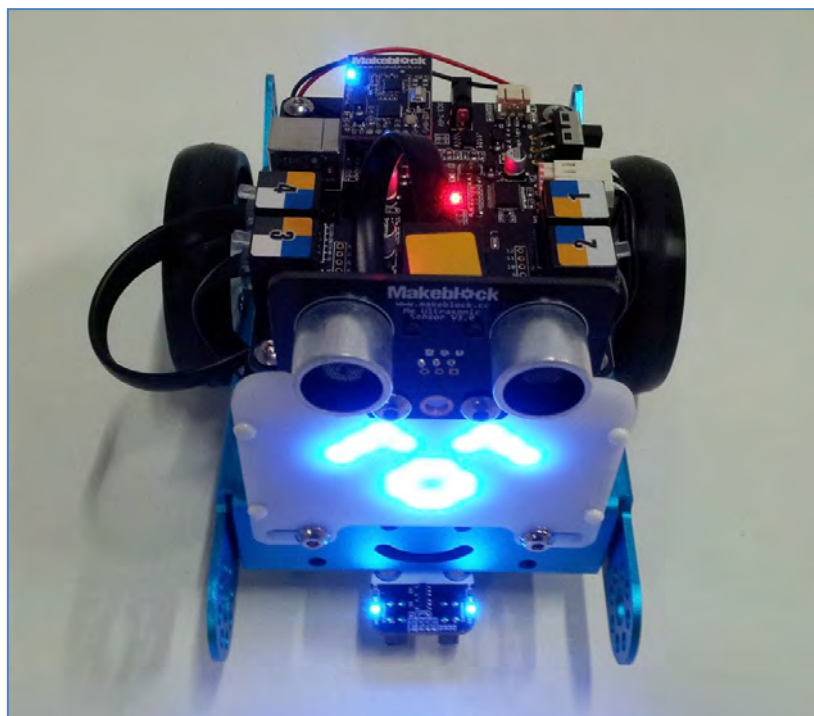
4.6. Matriz de LEDs 8x16

Antes de nada, debemos montar a matriz de LEDs no robot mBot. A súa posición correcta de montaxe é a que se amosa na seguinte figura, pero, se se desexa, podemos variar a montaxe e este cambio debe terse en conta á hora de programala:



Matriz de LED en posición correcta para a súa fácil programación

Outra forma de montaxe (invertida) que varía a súa programación é o seguinte:



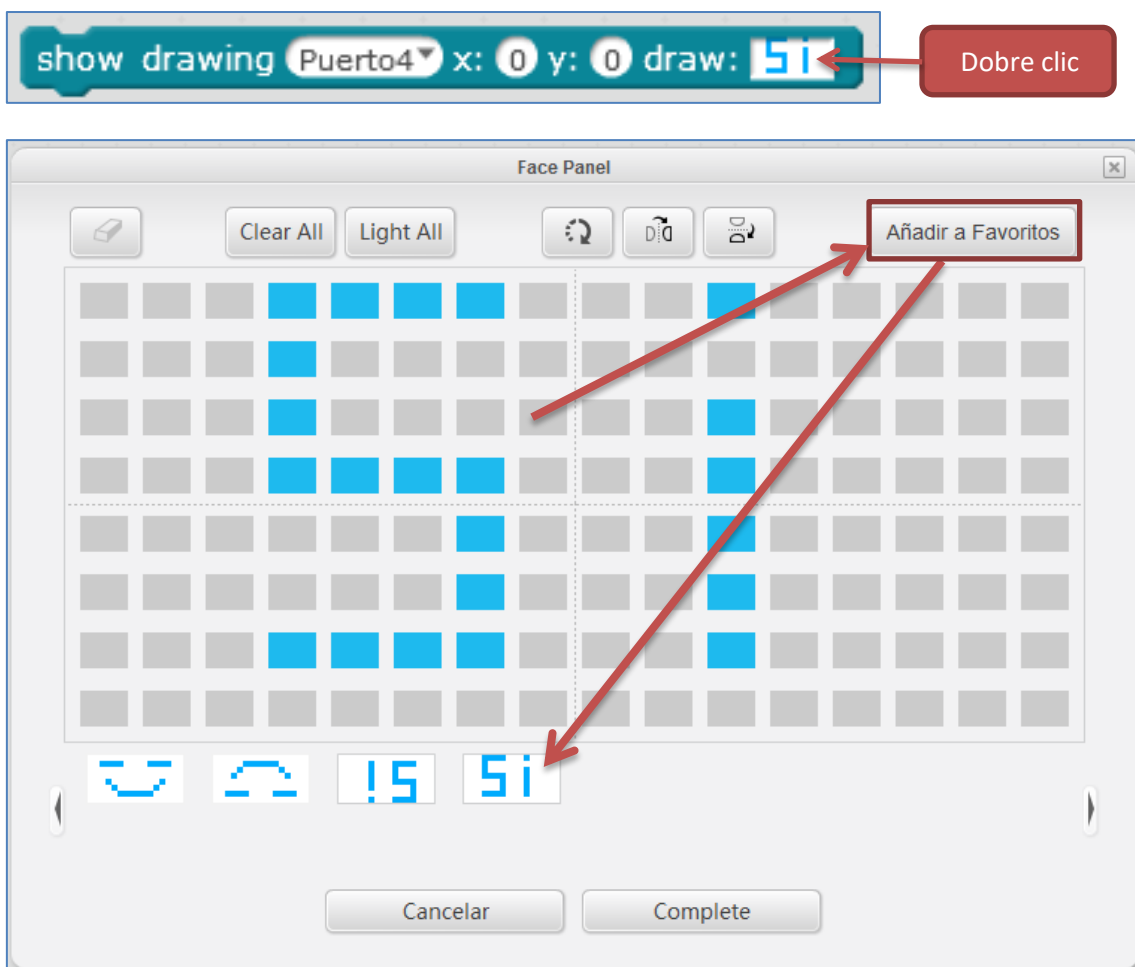
Divertíndome con mBot: Guía de manexo e programación

Despois de implementara debemos seguir os seguintes pasos:

1. Conectar o mBot ao ordenador mediante un cable USB e actualizar o firmware (mBot) en mBot. (*Conectar > Actualizar Firmware*)
2. Prender o mBot e seleccionar o puerto serie e placa correctos.
3. Escribir os programas e executalos con dobre clic y bandeira verde.

A instrución ou comando que máis se usa ao traballar coa matriz de LEDs é **show drawing**. Se facemos *dobre clic*, permítenos poñer o gráfico que queiramos, e incluso gardalo en favoritos!

(OLLO! O gráfico salería na montaxe correcta porque, se a montamos invertida, o seu gráfico tamén debemos crealo invertido. Isto explícase mellor nos seguintes exemplos).

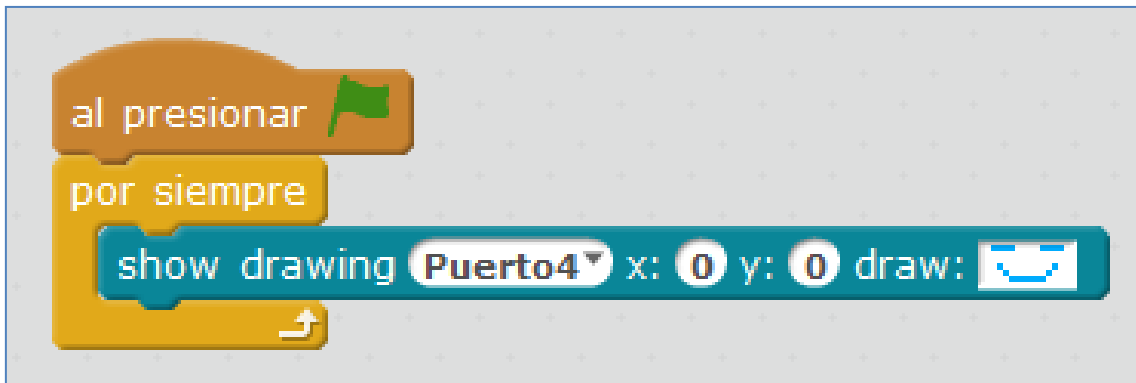


Vexamos varios exemplos de programación con mBlock:

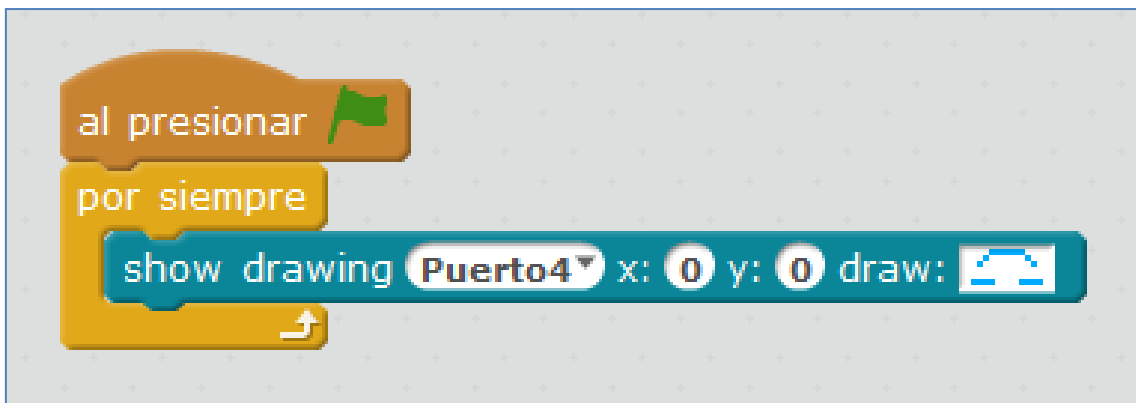
Divertíndome con mBot: Guía de manexo e programación

Exemplo 1: Queremos que na matriz se reflicta unha cara alegre:

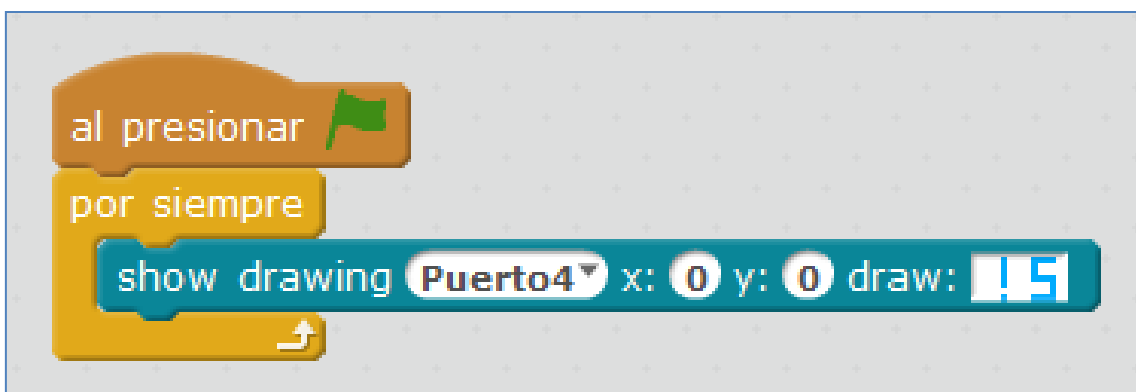
Coa matriz correctamente montada, crearíamos o seguinte programa:



Pero, coa matriz en montaxe invertida, deberíamos enviarlle ao robot o seguinte programa:

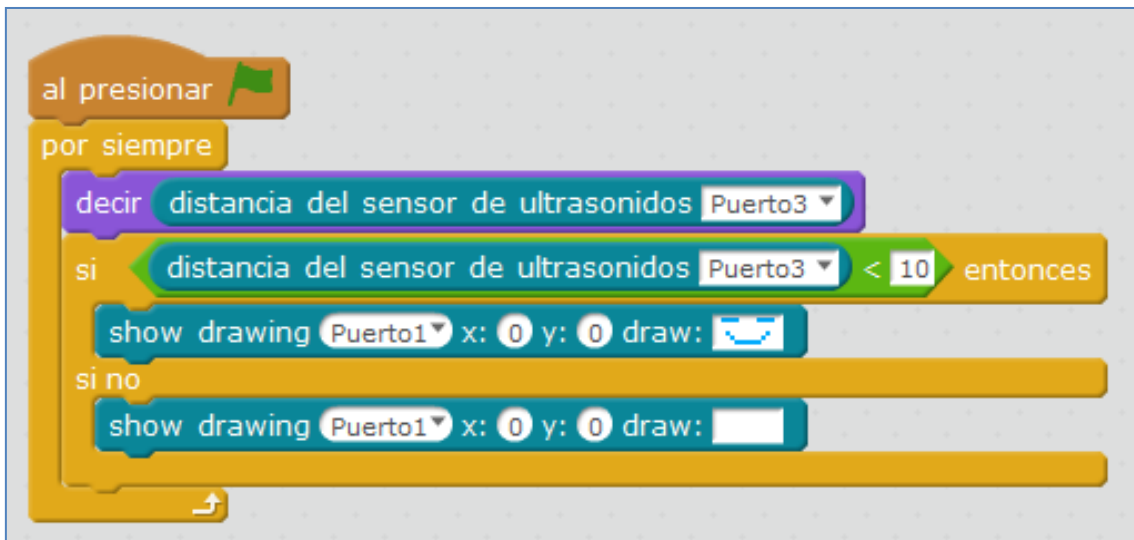


Exemplo 2: Queremos amosar un pequeno texto permanentemente. Por exemplo, a palabra “Si”, na configuración da matriz invertida. Programaríamolo da seguinte forma:



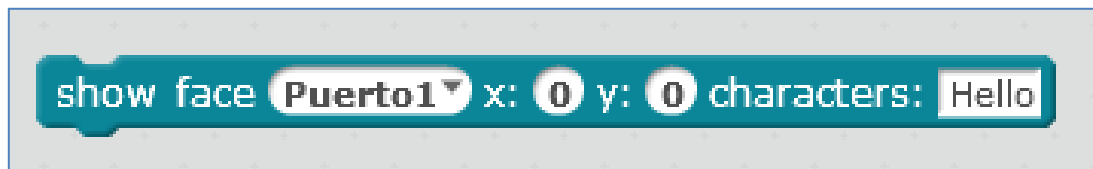
Exemplo 3: Combinando a matriz co sensor de ultrasóns:

Divertíndome con mBot: Guía de manejo e programación

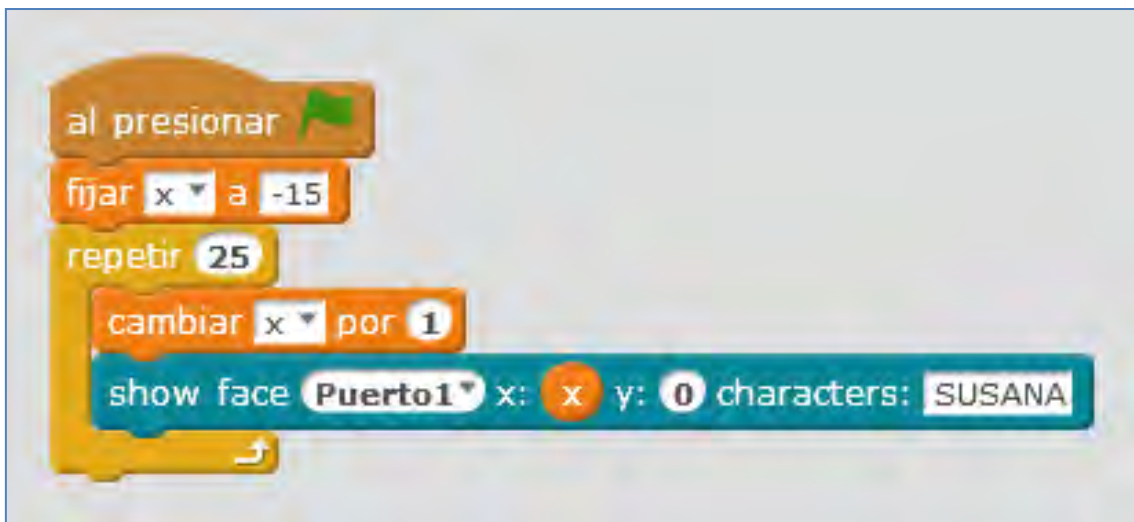


Exemplo 4: Podemos enviar un texto, coa instrución **show face**

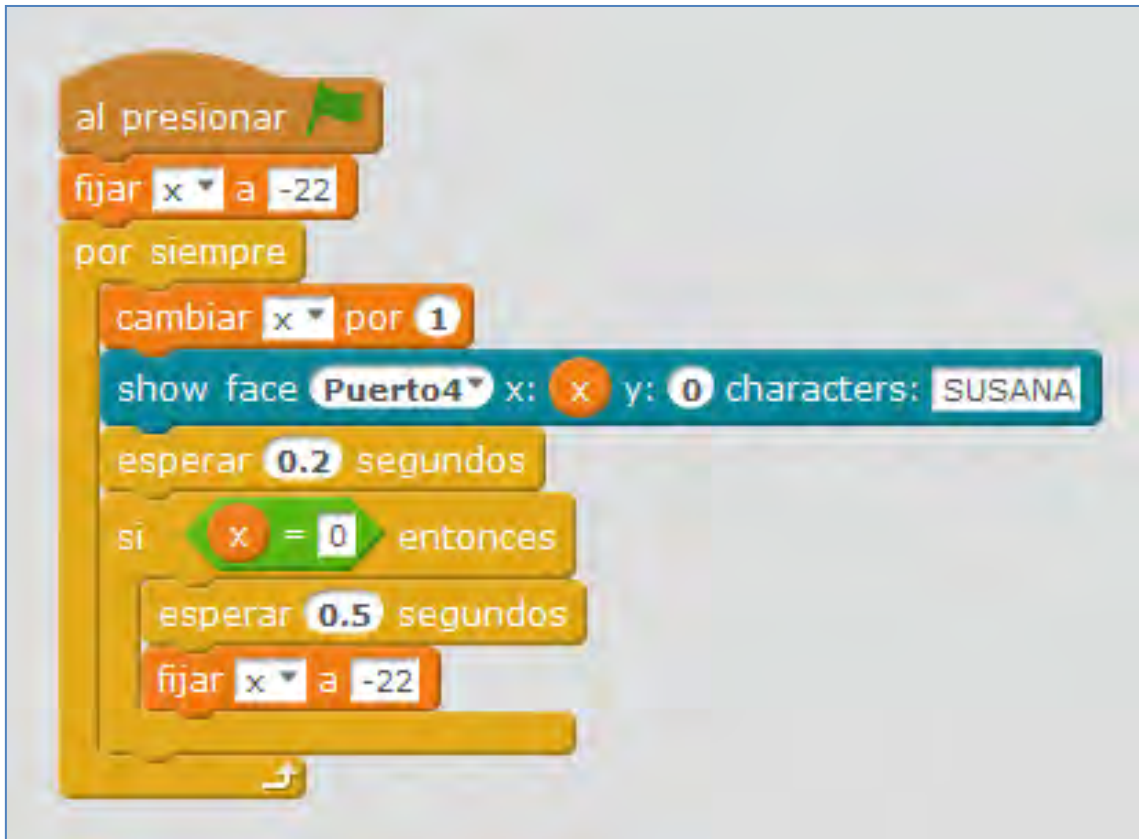
OLLO, se está invertida, o texto tamén ha que invertilo



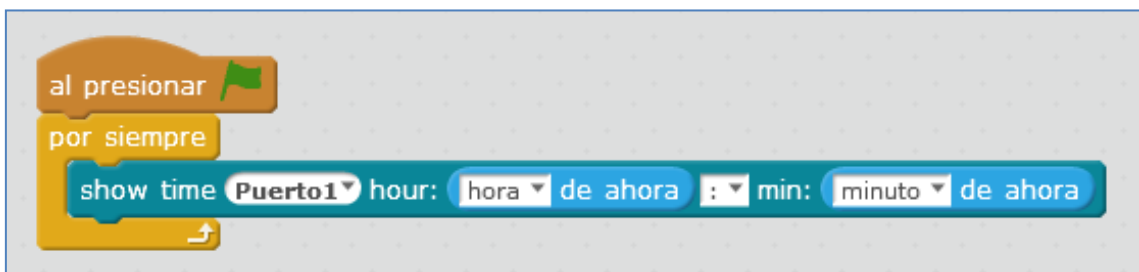
O seguinte programa envía ás coordenadas (0,0) a palabra SUSANA pero, para que colla, debemos programalo como segue:



Se queremos que se repita continuamente o texto:



Exemplo 5: Tamén podemos enviar a hora. Un script que nos serviría para ese fin é o seguinte:



Exemplo 6: No seguinte exemplo usamos o módulo de ultrasóns, e dous botóns do mando de IR do robot (as teclas R8 e R9).

O sensor de IR pode recibir do mando números, letras A-D, frechas e o botón configuración.

Ollo: O mando non é exclusivo dun robot, é dicir, os demais robots do aula reciben a mesma información polo que é importante apuntar o mando aos sensores do robot e non a outro robot do aula.

Realizar programas co sensor de IR ten unha dificultade: non se pode facer a través do ordenador. É dicir, non se pode realizar como o fixemos ata agora: Bandeira verde e comunicación entre PC e mBot

Divertíndome con mBot: Guía de manexo e programación

Por que? porque o mBot ten instalado nesta configuración o programa por defecto *Firmware* de ler o mando, e non podemos saltalo.

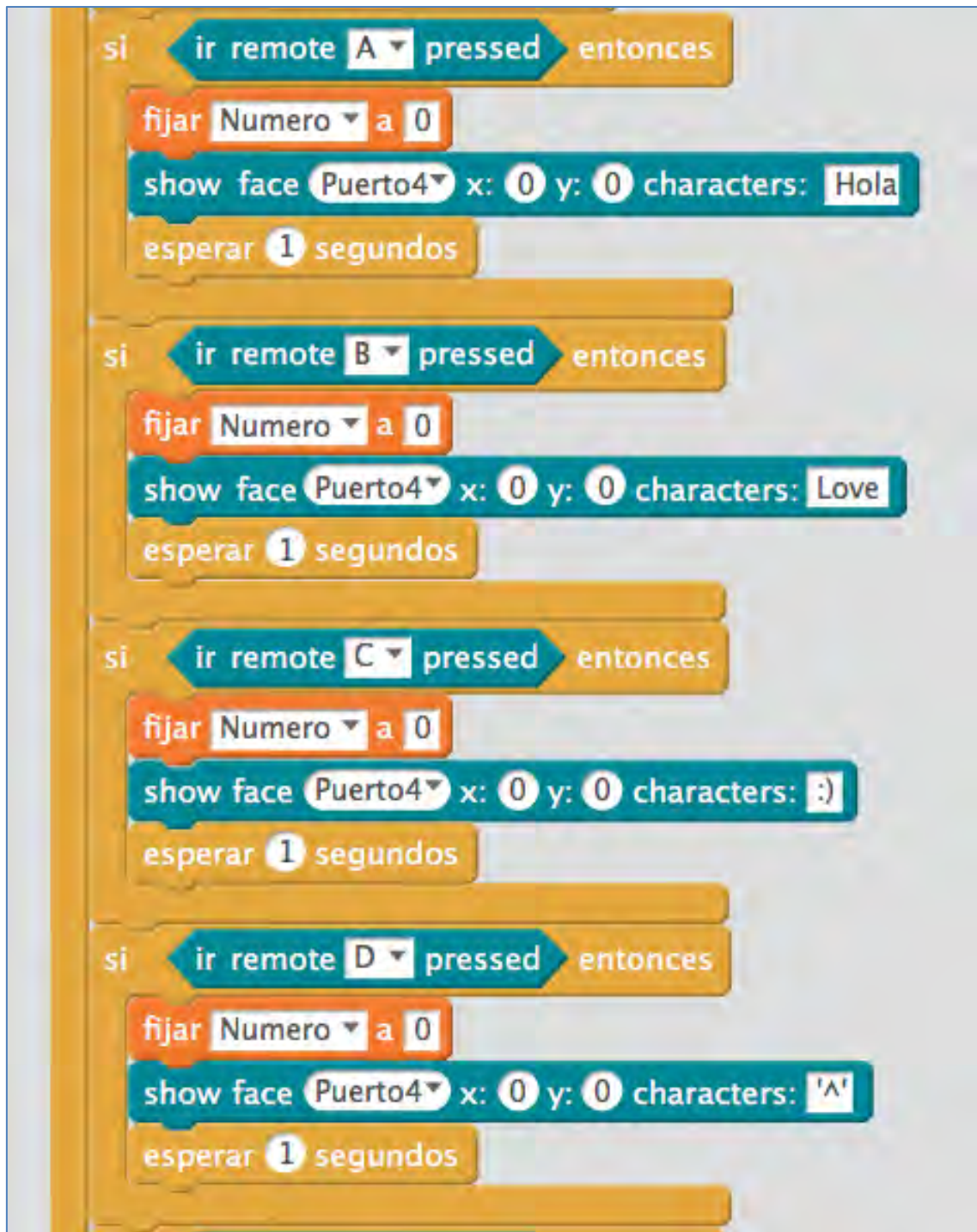
No seguinte exemplo, grazas ao módulo de ultrasóns, o robot testea a distancia á que está un obxecto (se é menor que 15 ou maior que 30) e, segundo as diferentes distancias, amosa unhas expresións na matriz de LED do robot mBot. O script é o seguinte:



```
al presionar
  fijar Numero a 2
  por siempre
    si ir remote R8 pressed entonces
      fijar Numero a 2
    si ir remote R9 pressed entonces
      fijar Numero a 1
    si Numero = 1 entonces
      show face Puerto4 x: 0 y: 0 characters: distancia del sensor de ultrasonidos Puerto3
    si Numero = 2 entonces
```



```
si Numero = 2 entonces
  si distancia del sensor de ultrasonidos Puerto3 < 15 entonces
    show face Puerto4 x: 0 y: 0 characters: ^_^
    esperar 0.2 segundos
  si no
    si distancia del sensor de ultrasonidos Puerto3 > 30 entonces
      show face Puerto4 x: 0 y: 0 characters: @@
      esperar 0.2 segundos
    si no
      show face Puerto4 x: 0 y: 0 characters: A^A
      esperar 0.2 segundos
```



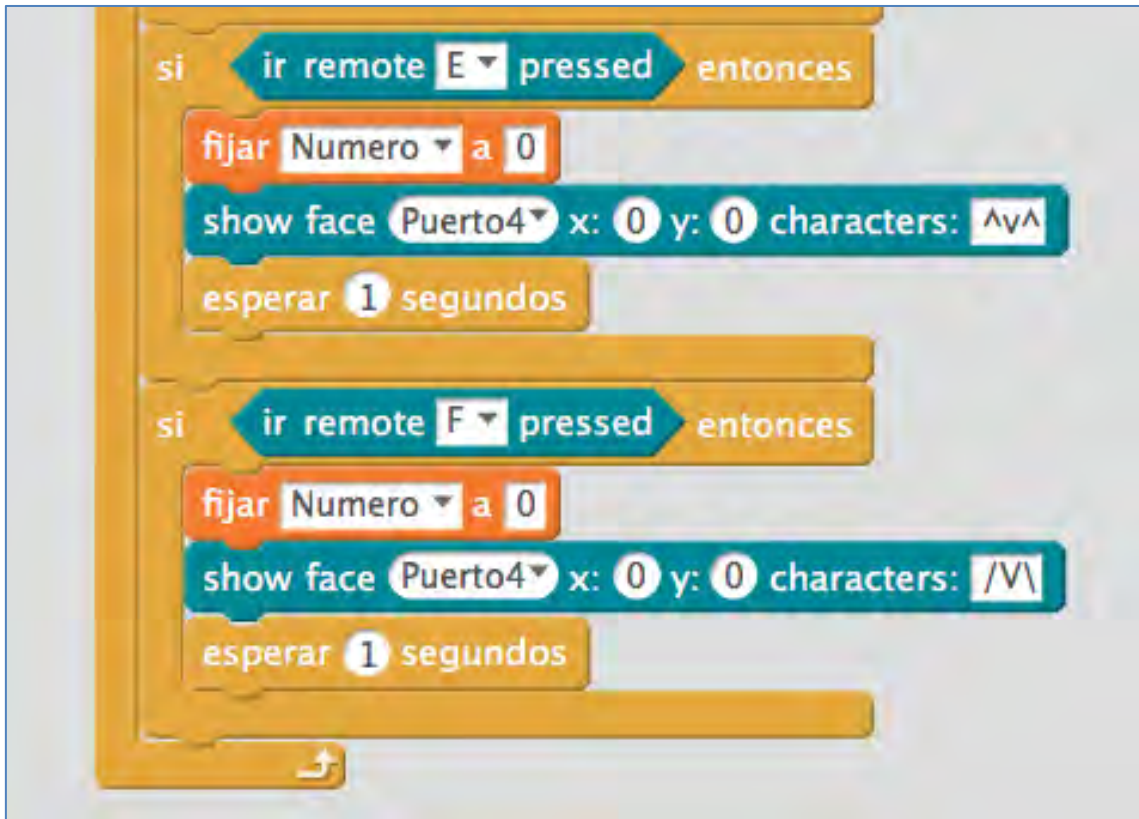
The image shows a Scratch script for controlling an mBot's face. It consists of four conditional blocks, each triggered by a remote button press (A, B, C, or D). Each block follows a similar sequence: set a variable 'Numero' to 0, display a specific message on the face, and wait for 1 second.

```
si ir remote A pressed entonces
  fijar Numero a 0
  show face Puerto4 x: 0 y: 0 characters: Hola
  esperar 1 segundos

si ir remote B pressed entonces
  fijar Numero a 0
  show face Puerto4 x: 0 y: 0 characters: Love
  esperar 1 segundos

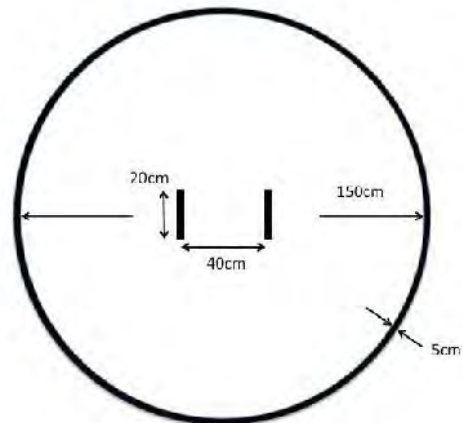
si ir remote C pressed entonces
  fijar Numero a 0
  show face Puerto4 x: 0 y: 0 characters: :)
  esperar 1 segundos

si ir remote D pressed entonces
  fijar Numero a 0
  show face Puerto4 x: 0 y: 0 characters: '^'
  esperar 1 segundos
```



4.7. Robot sumo

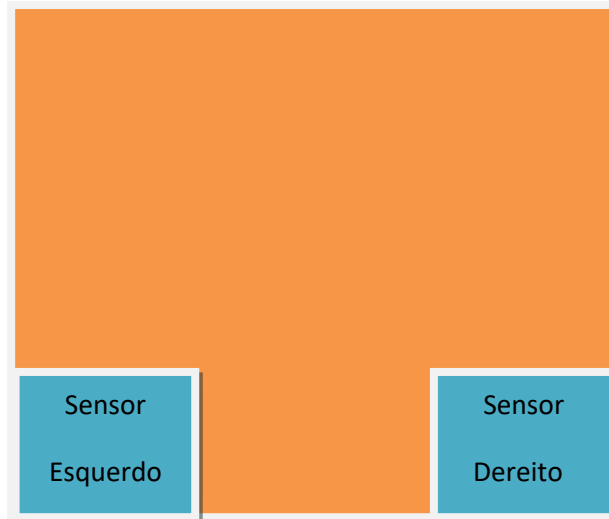
A programación para unha proba *sumo* varia drasticamente dependendo do número de sensores que lle poñas ao teu robot e de onde os acomodes. A proba *sumo* realizárase nunha pista circular. Dentro desa pista, o robot debe atacar e saber buscar aos contrincantes, pero, ao mesmo tempo, non debe saír dese recinto, é dicir, debe ser quen de detetar a cor do borde da pista e, nese momento, volver a dentro.



Un sensor que nos servirá para detetar ao inimigo é o sensor de ultrasóns. O noso robot pode dispoñer dun ou de dous, e no caso de que sexan dous, estes poden estar situados no lugar que nos pareza mellor (un diante e outro atrás, os dous diante a ambos lados do robot, etc).

Tra-la nosa elección, a programación é sinxela se temos claro como queremos que funcione o noso robot. Por exemplo, imaxinemos que nos decantamos por dous sensores de ultrasóns a ambos lados do robot na parte frontal (despois incluiremos os sensores de liña). Imos chamalos sensor dereito e sensor esquerdo. Na seguinte imaxe pode verse a súa situación esquemática nun bloque:

Divertíndome con mBot: Guía de manexo e programación



Parte frontal do robot

Podemos ter a estratexia de ir sempre cara adiante ou de xirar a la derecha o esquerda se ningún sensor detetou nada, pero, se algún sensor deteta ao oponente, entón debe dirixirse a el para atacar. Esta estratexia resúmese na seguinte taboa de verdade onde 0 significa que non detecta e 1 que detecta:

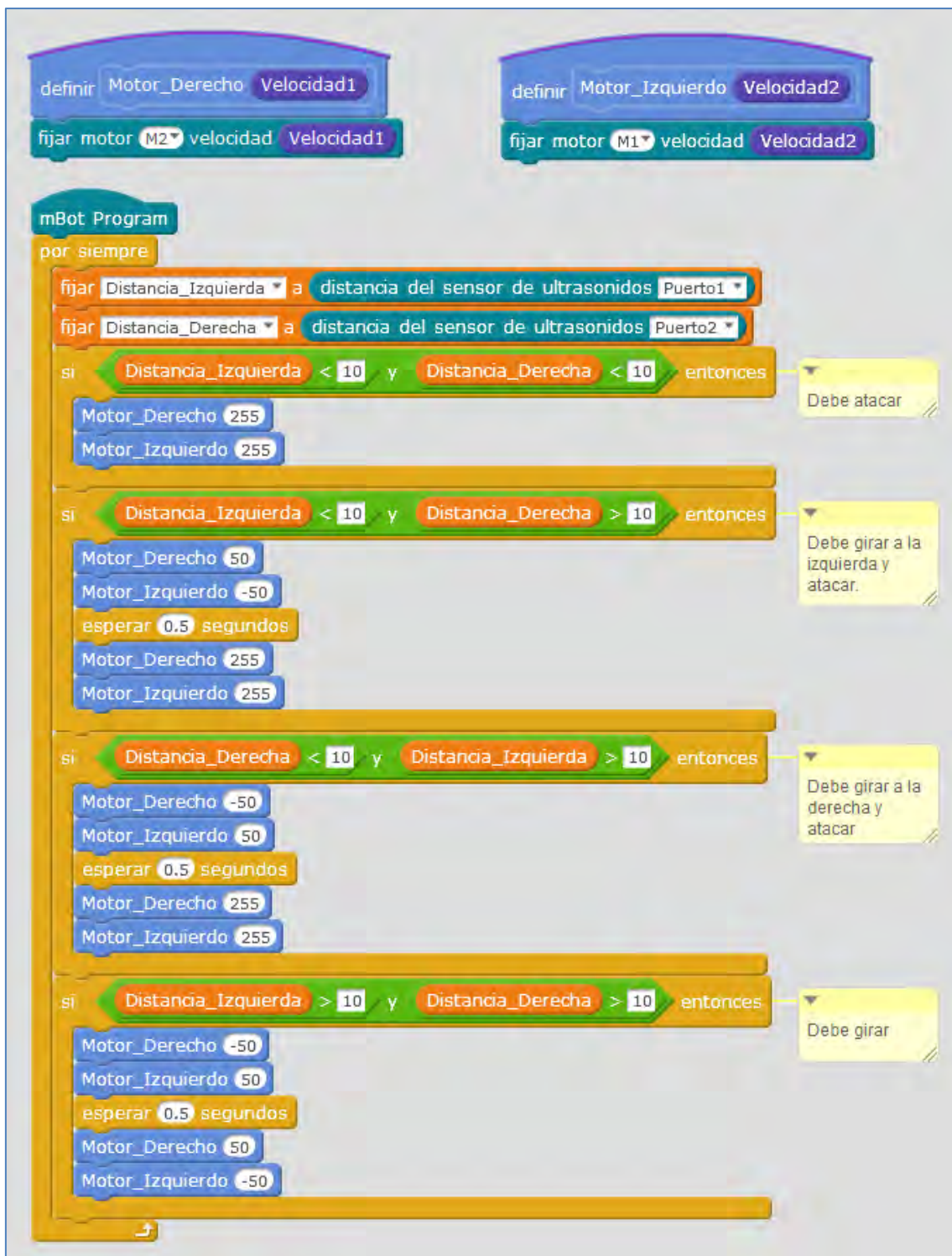
Sensor Esquerdo	Sensor Dereito	Inimigo	Acción
0	0	Non	Xirar ou seguir adiante
0	1	Si	Xirar á dereita
1	0	Si	Xirar á esquerda
1	1	Si	Ir de fronte



mBot con 2 sensores de ultrasóns

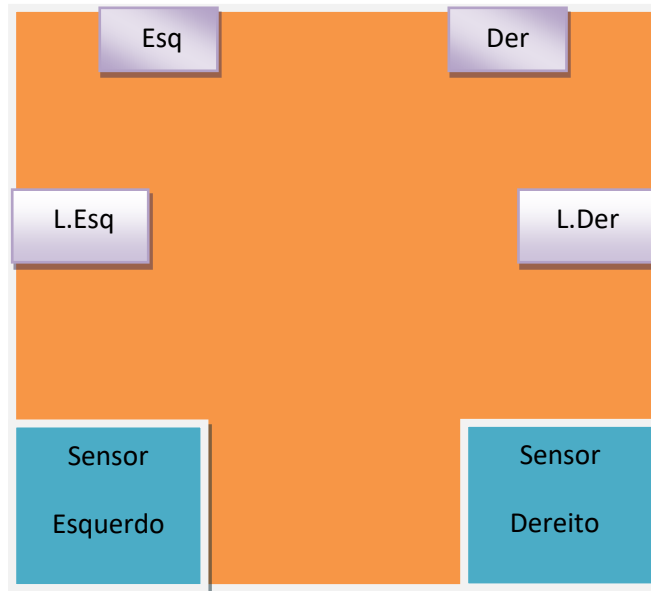
Divertíndome con mBot: Guía de manexo e programación

Relativo á distancia, no sentido de cal debemos usar como límite de detección, esta dependerá da situación exacta dos nosos sensores no robot e dos nosos desexos no programa. Neste exemplo, decíciuse que sexan 10cm:



Primeira parte do programa Sumo con dous ultrasóns

Persoalmente, o ideal é que o robot sempre ataque de fronte e ao mesmo tempo, que non saía da zona de loita, limitada por unha cor determinado de liña. Para conseguir isto último, podemos usar sensores detectores de liña e programalos para que, cando o sensor detecte a liña, o robot afástase dela. Supoñamos que usamos 4 e os situamos da seguinte forma:



Pensando na nosa estratexia, unha posible táboa de verdade, considerando 0 como non detección e 1 como detección, podería ser a seguinte:

L.9gq	9gq	Der	L.Der	Acción a programar
0	0	0	0	Buscar
0	0	0	1	Dereita Rápido: xira a roda esquerda e a roda dereita hacia atrás
0	0	1	0	Dereita Suave: só xira a roda esquerda e a roda dereita está quieta
0	0	1	1	Nada
0	1	0	0	Esquerda Suave
0	1	0	1	Nada
0	1	1	0	De fronte
0	1	1	1	Nada
1	0	0	0	Esquerda Rápido
1	0	0	1	Nada
1	0	1	0	
1	0	1	1	
1	1	0	0	
1	1	0	1	
1	1	1	0	
1	1	1	1	

Nela, basta con programar as filas resaltadas en lila para que o noso robot sumo ataque e evite saír do recinto de loita.

Divertíndome con mBot: Guía de manexo e programación

Á hora de programar ambos sensores á vez, ultrasóns e sensores de liña, “todo” é libre cara os intereses do programador. Explícome, podo considerar que o máis importante é que o meu robot priorice en escapar da liña banca tomando como secundario o seu ataque ou, calquera outra alternativa a esta estratexia. No meu caso, para mín, o máis importante é salvarme, polo que, nos seguintes exemplos particularizados ao mBot, verase esta defensa inicial.

Exemplo 1 de programa sumo

Neste caso, usaremos un sensor de ultrasóns para a parte dianteira do mBot e dous sensores de liña (un dianteiro e outro traseiro). O programa está pensado para loitar nun circuito limitado por unha liña branca.

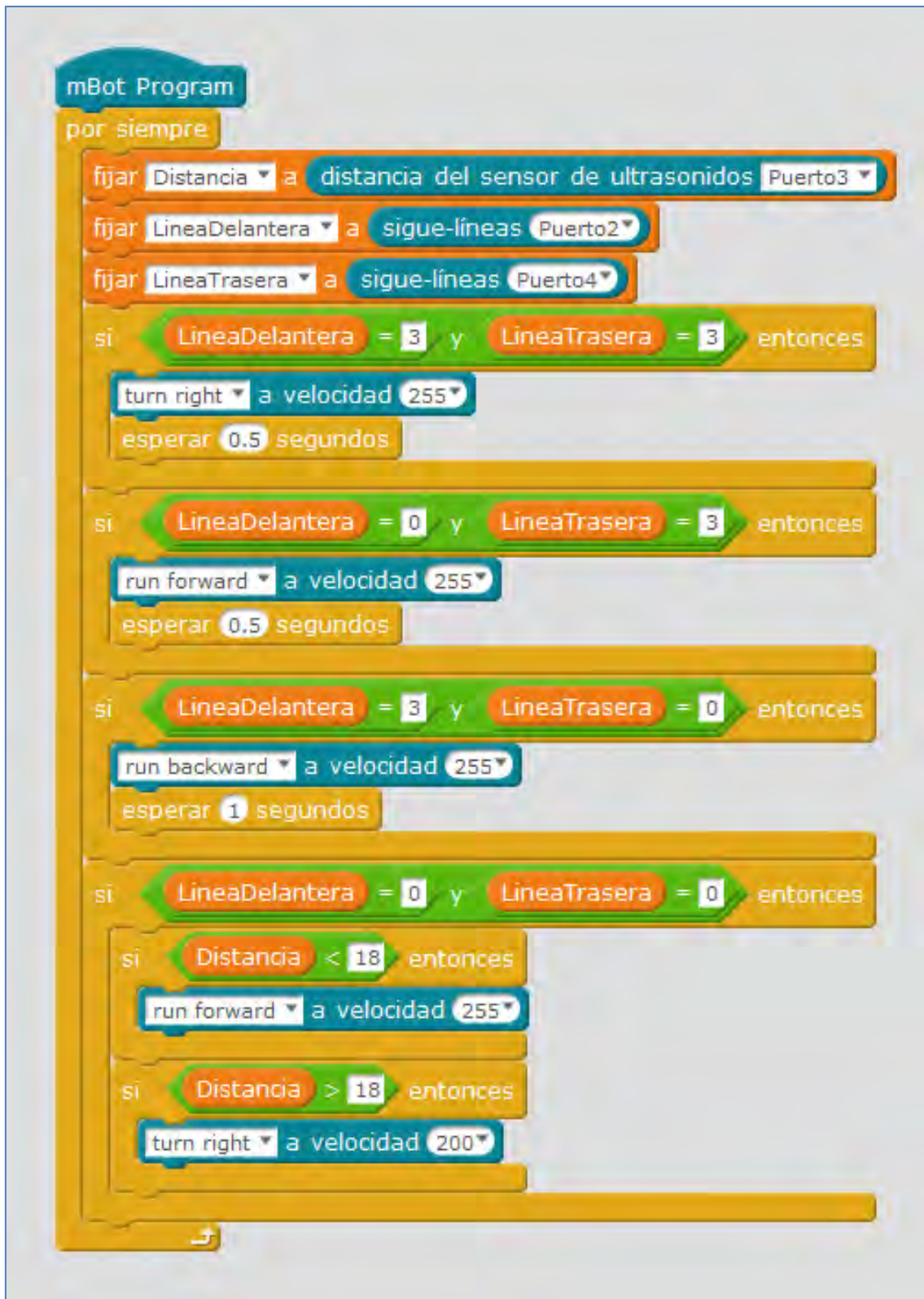
Dispoñemos de tres variables: *Distancia* (que é a que medirá o sensor de ultrasóns e que se estima en 18 cm para a detección), *LiñaDianteira* (medirá negro ou branco no sensor dianteiro) e *LiñaTraseira* (medirá negro ou branco no sensor traseiro):



Comezarei coa miña priridade: estar no recinto. Os sensores de liña, como xa vimos, presentan 4 posicións e verán branco na posición 3. Neste caso, debemos intentar que o noso robot non saia do circuito de loita. Polo tanto: Se ve liña branca o dianteiro, debemos ir cara atrás, e se ve branca o traseiro, debemos ir cara adiante. Xa conseguido isto, tamén debe atacar, e farao se está no recinto (ve negro) e detecta a un opoñente. Tamén debe buscar, e farao se está no recinto e non detecta a un opoñente.

Un posible programa sumo que afronta as consideracións anteriores é o seguinte:

Divertíndome con mBot: Guía de manejo e programación



The image shows a Scratch-style code editor for an mBot. The code is written in Spanish and is contained within a 'por siempre' (forever) loop. It initializes variables for distance and line sensors, then uses conditional logic to navigate based on sensor readings.

```
mBot Program
por siempre
  fijar Distancia a distancia del sensor de ultrasonidos Puerto3
  fijar LineaDelantera a sigue-líneas Puerto2
  fijar LineaTrasera a sigue-líneas Puerto4
  si LineaDelantera = 3 y LineaTrasera = 3 entonces
    turn right a velocidad 255
    esperar 0.5 segundos
  si LineaDelantera = 0 y LineaTrasera = 3 entonces
    run forward a velocidad 255
    esperar 0.5 segundos
  si LineaDelantera = 3 y LineaTrasera = 0 entonces
    run backward a velocidad 255
    esperar 1 segundos
  si LineaDelantera = 0 y LineaTrasera = 0 entonces
    si Distancia < 18 entonces
      run forward a velocidad 255
    si Distancia > 18 entonces
      turn right a velocidad 200
```

Divertíndome con mBot: Guía de manexo e programación

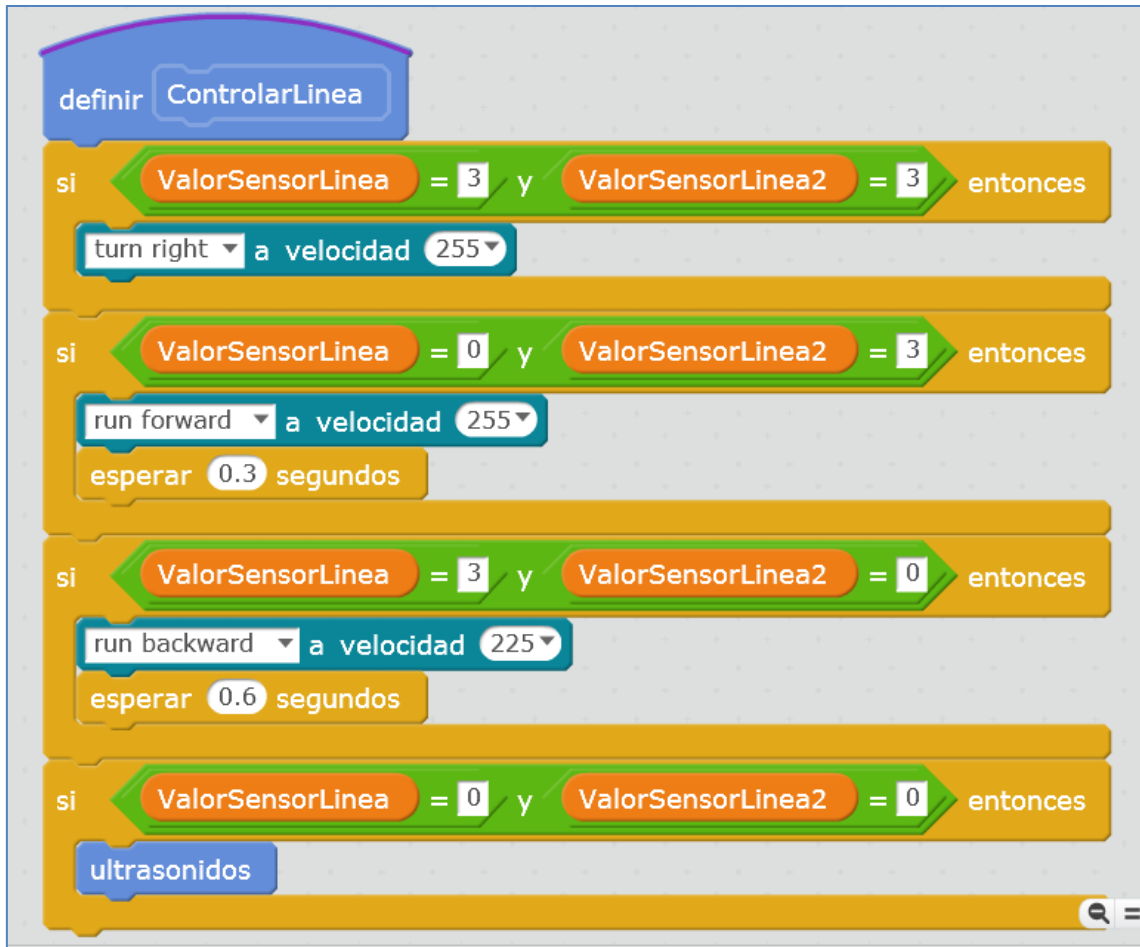
Exemplo 2 de programa sumo:

Supoñamos agora que queremos usar dous sensores de ultrasóns e que os situamos, un diante e outro detrás. Ademais, ao noso robot engadímoslle dous sensores de liña e da mesma forma que antes, un estará diante e outro detrás. Para o seu control, necesitamos definir 4 variables e estas poden ser as seguintes: *ValorSensorLiña* (para diante), *ValorSensorLiña2* (para atrás), *ultrasóns1* (para delante) e *ultrasóns2* (para atrás):



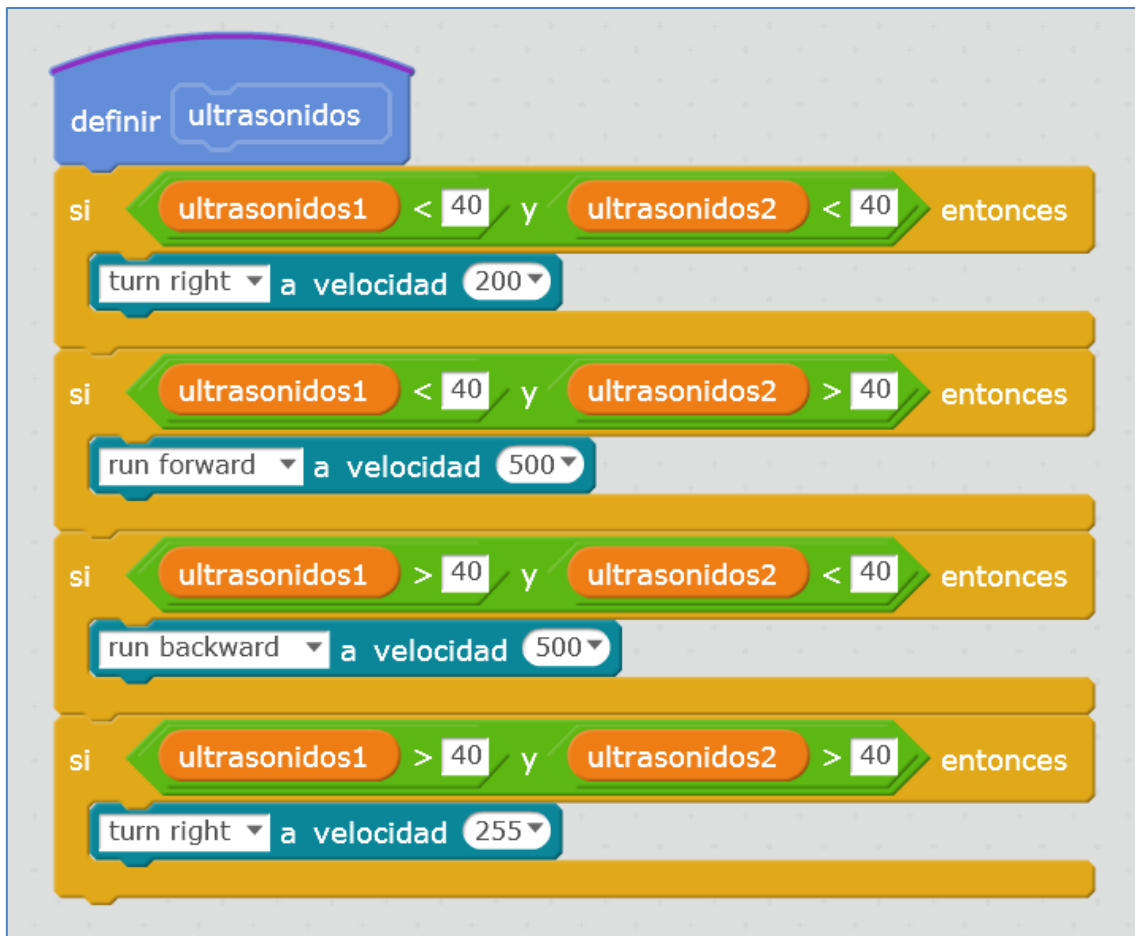
Se a miñ prioridade é a mesma; estar no recinto, comezarei con esa liña de programación. Os sensores de liña, como xa vimos, presentan 4 posicións e verán branco na posición 3. Neste caso, debemos intentar que o noso robot non saia do circuito de loita. Polo tanto: Se ve liña branca o dianteiro, debemos ir cara atrás, e se ve branca o traseiro, debemos ir cara adiante. Neste exemplo, programei estas posibilidades nun bloque que chamou *ControlarLiña* e que é o seguinte:

Divertíndome con mBot: Guía de manexo e programación



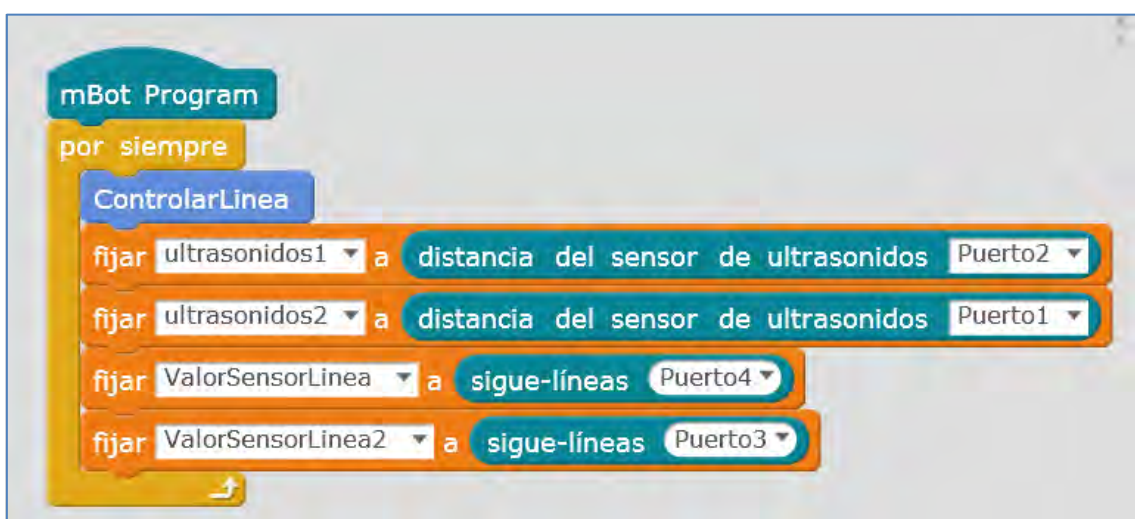
Como podedes ver, no caso de que o robot estea no recinto, vexa negro, debe comportarse como un bo sumo e atacar ou buscar. Isto consígoo usando os seus sensores de ultrasóns. Sensores que programei no bloque definido como “*ultrasóns*” e cuxa programación, limitada neste caso polo valor numérico de detección 40 cm, é a seguinte:

Divertíndome con mBot: Guía de manejo e programación



O bloque de ultrasóns só o executará cando o robot estea no recinto, atacando ou buscando na forma que desexe o programador.

Ata agora, definimos os bloques, pero non o programa de execución que os chama e que é o seguinte:



Divertíndome con mBot: Guía de manexo e programación

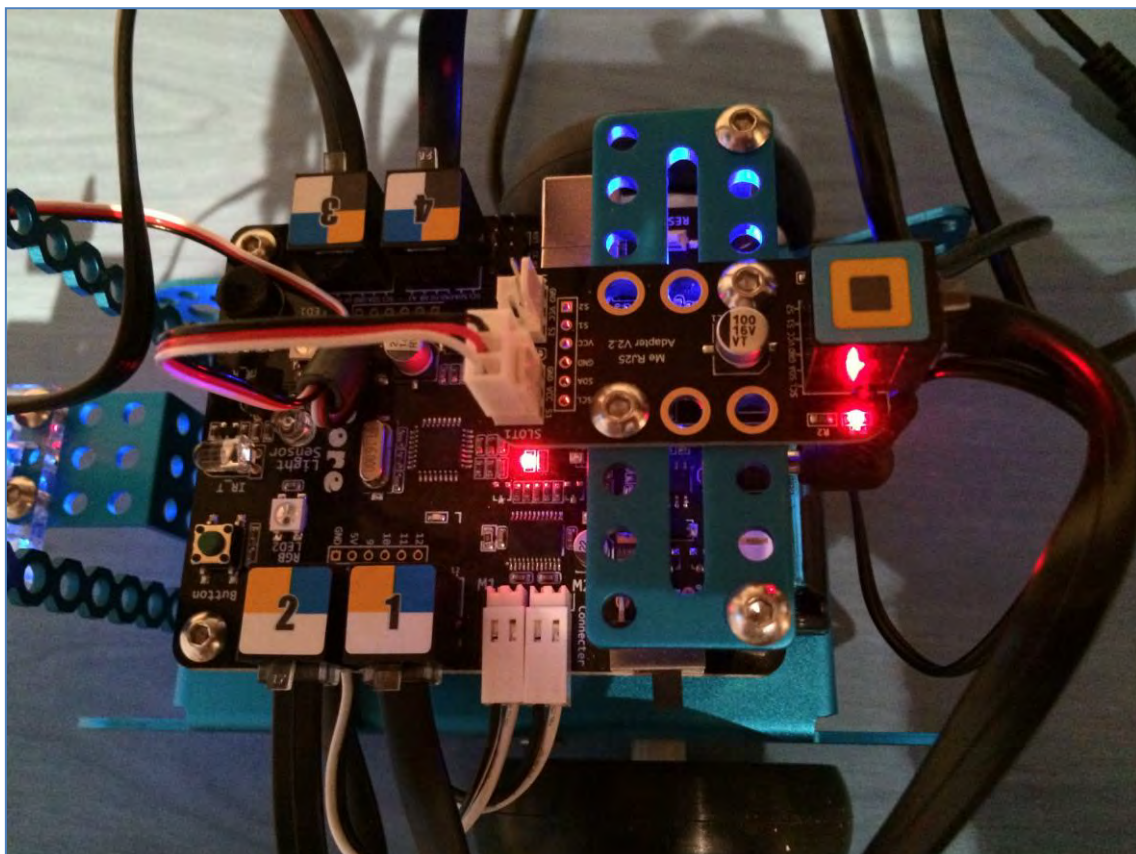
4.8. Servomotor

Un servomotor pódese considerar como un tipo especial de motor de continua que se caracteriza por ter a capacidade de xirar un ángulo determinado dentro dun intervalo de operación.



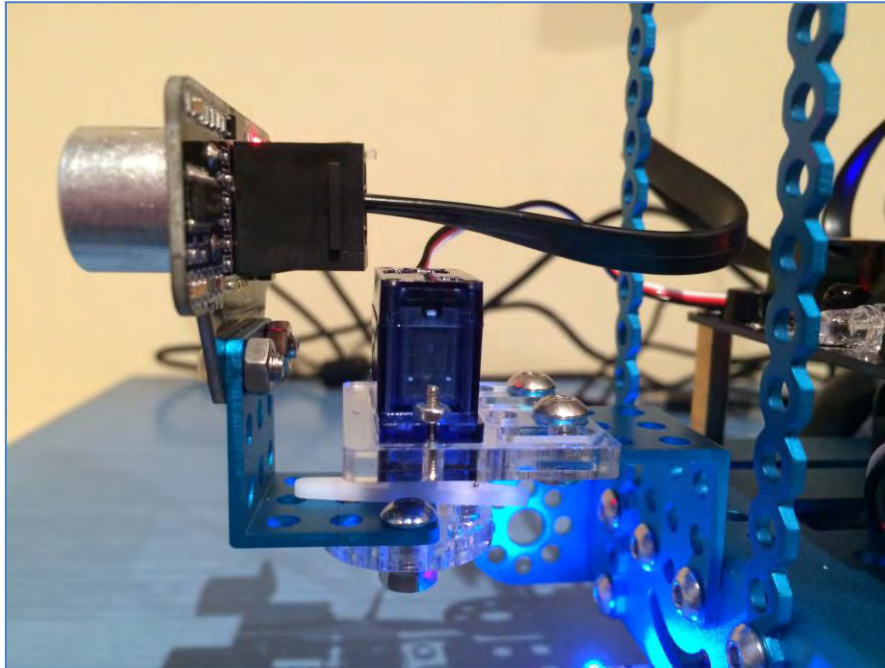
Micro Servo 9g (servomotor)

O servomotor da casa Makeblock necesita dun adaptador RJ25, como pode verse na montaxe da seguinte imaxe. Todo adaptador RJ25 dispón de 2 "Bancos" ou "Slots" ou conectores, para poder conectar no mesmo módulo un sensor de temperatura e un servo, por exemplo. O adaptador necesita dun porto e pode conectarse a calquera dos portos: 1, 2, 3 ou 4, xa que os seus cores ID son negro, amarelo e azul. No noso caso, conectámolo ao porto 1:



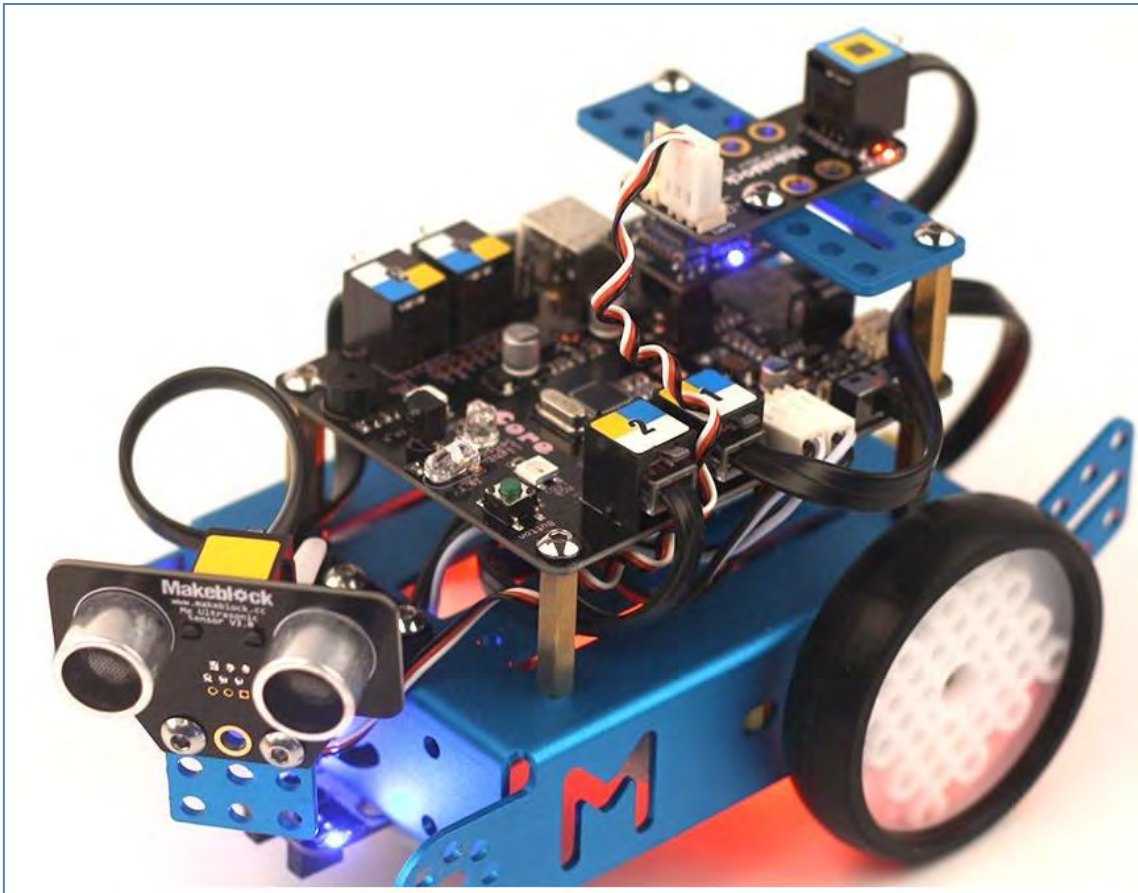
Divertíndome con mBot: Guía de manexo e programación

A maiores, no noso exemplo, o servo uniuse ao sensor de ultrasóns, de modo que, consiga movelo en diferentes posicións angulares do servo:



Un script para mover o servo en 5 diferentes posicións angulares (0, 45, 90, 135 e 180) sería o seguinte: Pódese elixir o “Banco1” (un banco é un slot ou un conector) porque non temos o adaptador conectado a máis dun compoñente.

```
al presionar
  fijar angulo a 0
  repetir 2
    set servo Puerto1 Banco1 angle angulo
    esperar 1 segundos
    cambiar angulo por 45
    set servo Puerto1 Banco1 angle angulo
    esperar 1 segundos
    cambiar angulo por 45
    set servo Puerto1 Banco1 angle angulo
    esperar 1 segundos
    cambiar angulo por 45
    set servo Puerto1 Banco1 angle angulo
    esperar 1 segundos
    cambiar angulo por 45
    set servo Puerto1 Banco1 angle angulo
    esperar 1 segundos
  fijar angulo a 0
```



4.9. Sensor de temperatura SD18B20

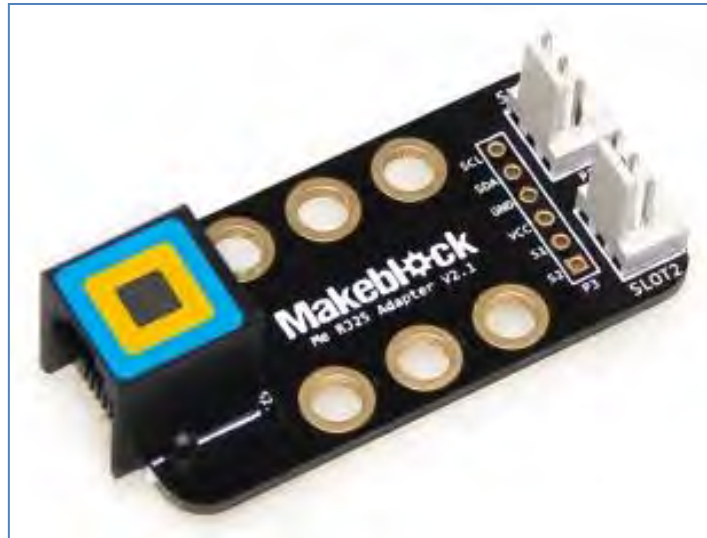
O sensor de temperatura sumerxible da casa Makeblock é o sensor dixital SD18B20. Presenta unha precisión de $\pm 0.5^{\circ}\text{C}$ (de -10°C a $+85^{\circ}\text{C}$), sendo, o seu tempo de captura da medida, inferior a 1 segundo. Grazas ao seu envoltorio estanco selado podemos mergullalo nun líquido e usalo para que nos proporcione a medida da temperatura dunha substancia líquida. Ao ser dixital, o valor de temperatura que nos proporcionará non se ve alterado pola distancia do cableado do sensor.

Este sensor pódese programar con arduino ou con mBlock, usando en scratch o correspondente comando:



Para conectalo á placa do mBot, necesita dun módulo adaptador RJ25 como o que se amosa na imaxe e que xa utilizamos co módulo servomotor:

Divertíndome con mBot: Guía de manexo e programación



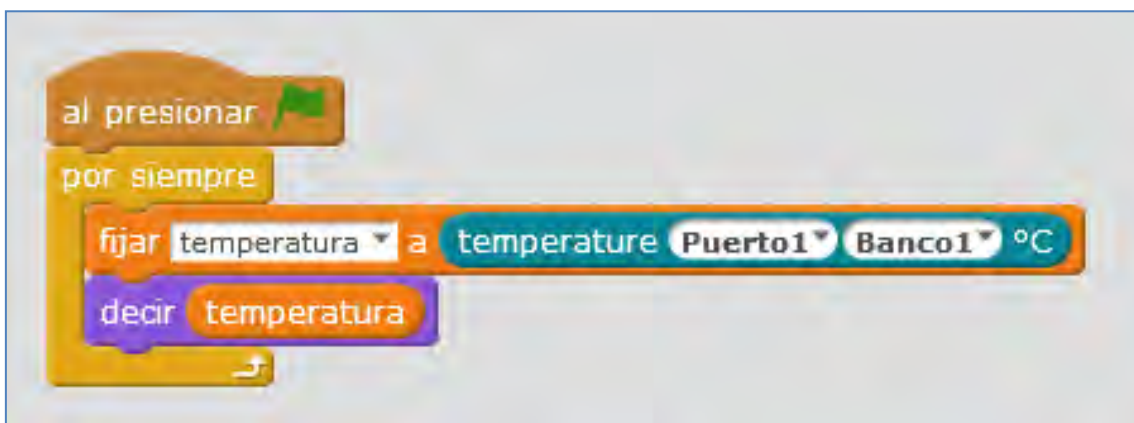
Módulo adaptador RJ25

O sensor de sonda mergullable, cuxo rango de medida é de -55°C a 125°C , pode verse na seguinte figura:



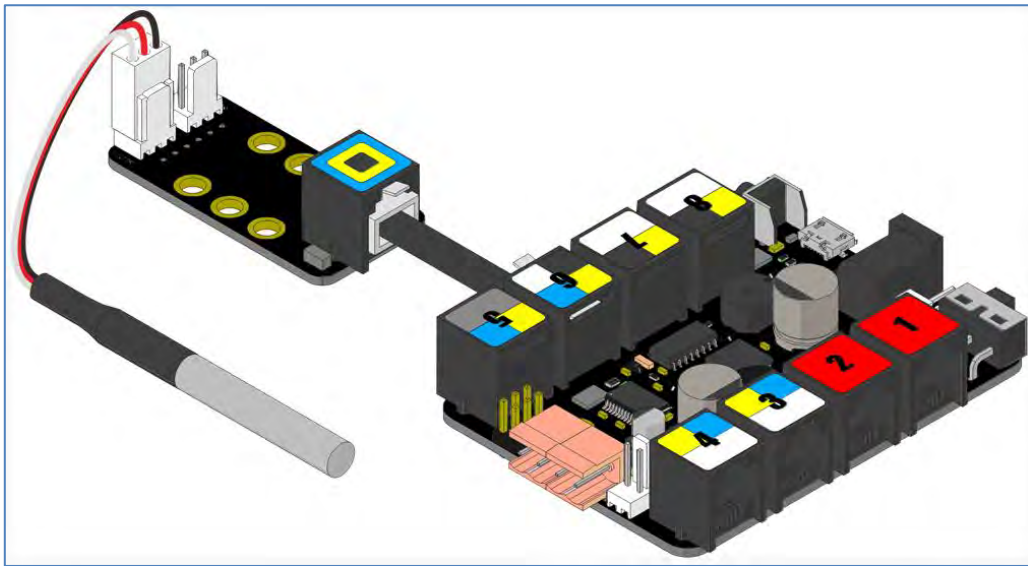
Módulo sensor de temperatura mergullable

Un programa sinxelo consistiría en medir a temperatura da sonda do sensor e amosala no escritorio. O script, estando o sensor conectado ao *Banco1* e o adaptador al *Porto 1*, sería o seguinte:



Divertíndome con mBot: Guía de manexo e programación

O conexionado, neste caso usando unha placa Orion conectando o sensor por medio do adaptador ao porto 6, sería o seguinte:



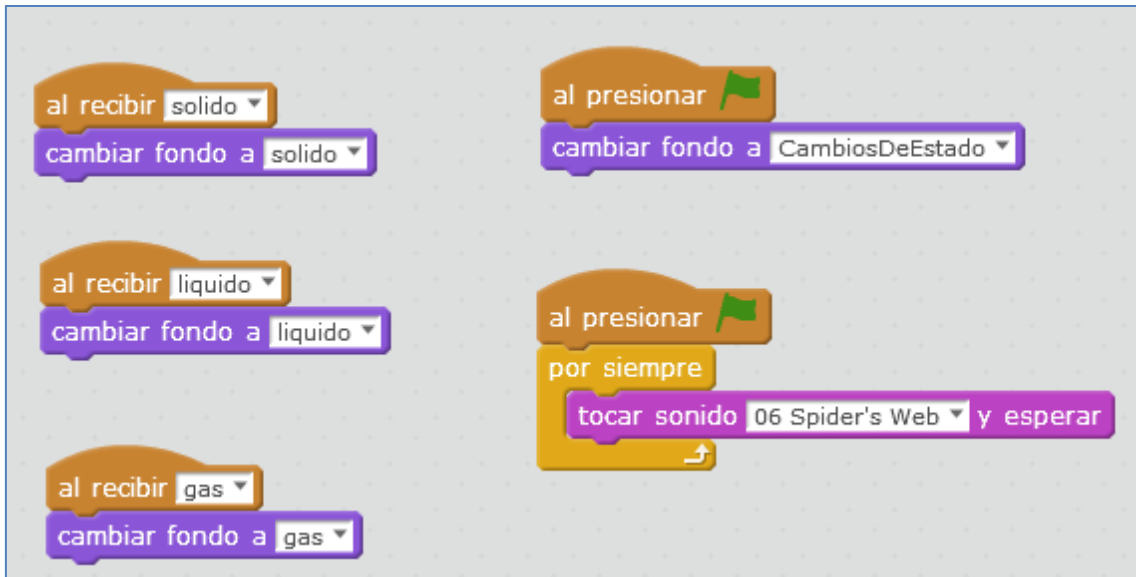
No seguinte exemplo conectouse á placa mCore no *banco 2* do *porto 4*. Creouse un programa para que o sensor mida a temperatura dunha substancia que teste e me amose o seu estado (sólido, líquido ou gas), simulando o movemento das súas moléculas. Para iso, utilizou un único obxecto, chamado “Molécula”, e 4 escenarios ou fondos. Nas seguintes imaxes obsérvase o obxecto molécula no escenario inicial e os 4 escenarios do programa:



Escenarios do programa

Cada escenario chámase mediante mensaxes. É dicir, ao recibir a mensaxe indicado (sólido, líquido ou gas), cambia o fondo do escenario ao que se programou.

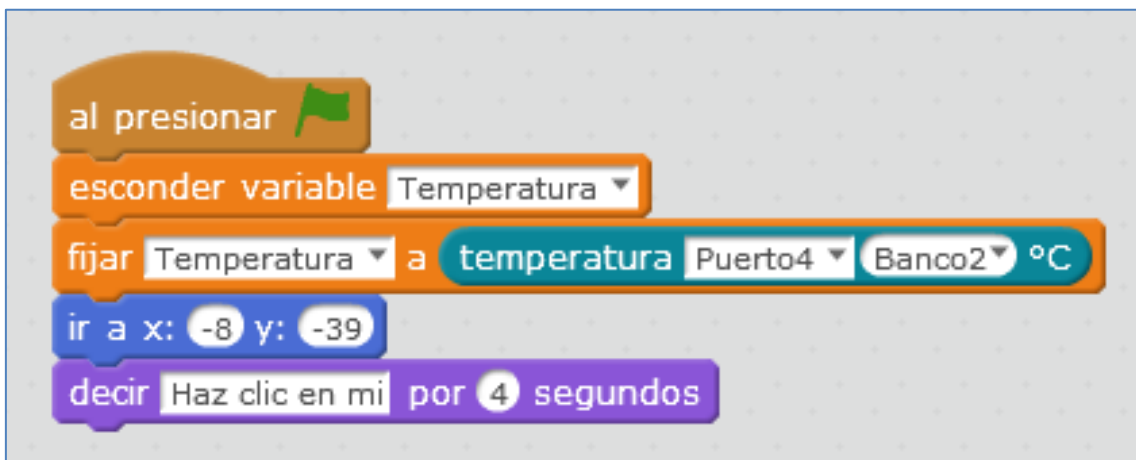
Divertíndome con mBot: Guía de manexo e programación



Script do escenario

O obxecto molécula é o que leva o peso do programa:

Inicialmente non busco que se amose a temperatura que está medindo o sensor de temperatura mergullable conectado ao *Banco2* dun adaptador RJ25 que se une ao *Porto 4* da placa do mBot, pero se quero que tome a primeira mostra para que as súas medidas se vaian regulando:



Sensor de temperatura conectado ao Porto 4

Tras facer clic no obxecto “Molécula” (xa pasaron 4 segundos), amósase o valor numérico da variable temperatura e o programa testea se se atopa en estado sólido, líquido ou gas, a través de diferentes intervalos de temperatura. Encontrado o seu rango de temperatura, envía a mensaxe correspondente (mensaxe que recolle o escenario). Se fosemos estritos e para o “agua”, as súas temperaturas serían 0°C e 100°C e non 10° e 100°, que é o que utilicei:

Divertíndome con mBot: Guía de manejo e programación



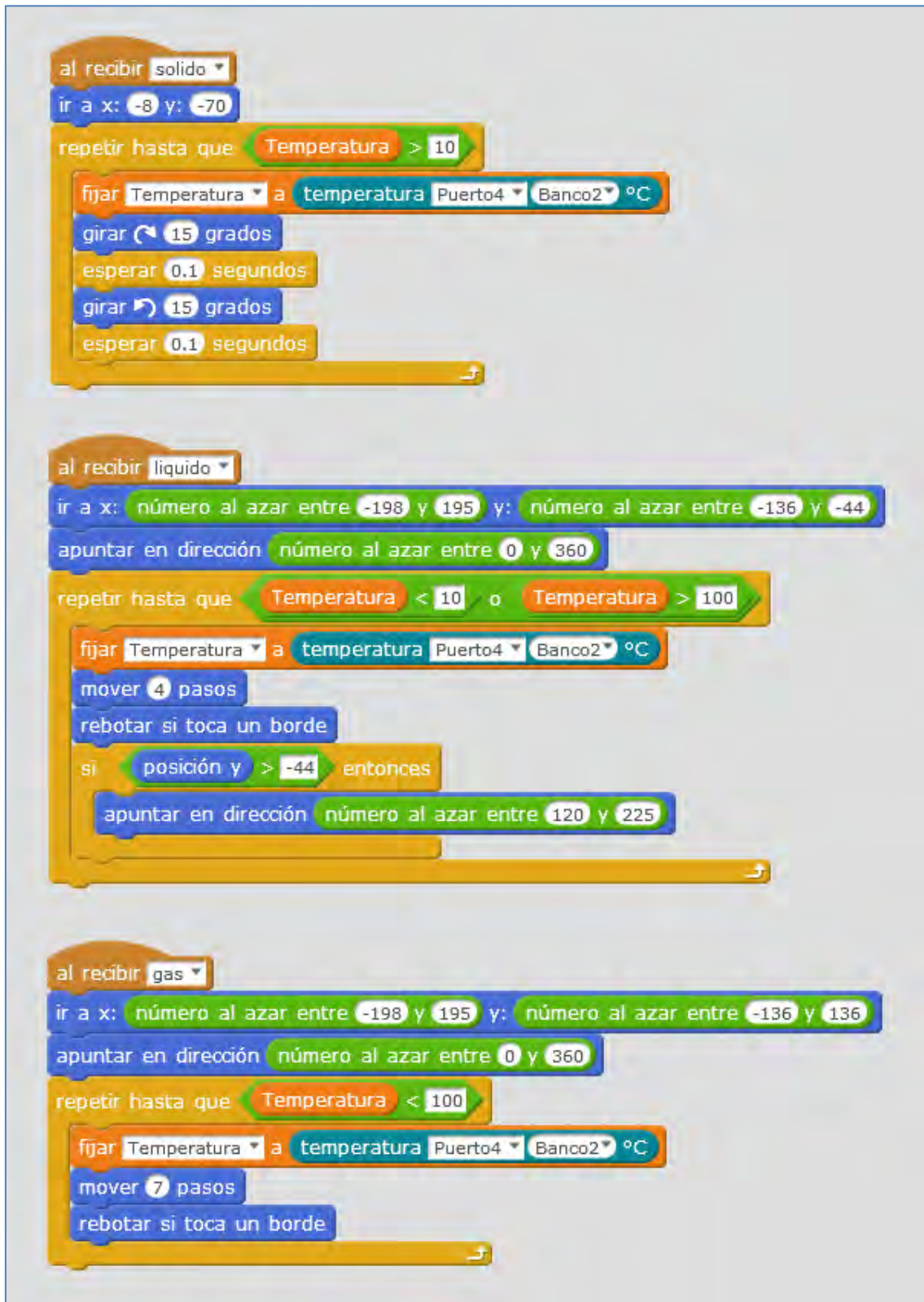
As mensaxes “sólido, líquido e gas” realizan a simulación no movemento da molécula.



Situación “Líquido”

Divertíndome con mBot: Guía de manexo e programación

En todo momento, testéanse cambios na variable temperatura, e deste modo asegurámonos que realiza a animación do seu “estado” correspondente:



A animación “sólido” simula que a molécula vibra, movéndose 15° á dereita e esquerda. A animación “líquido”, fai que a molécula se mova a unha determinada

Divertíndome con mBot: Guía de manexo e programación

velocidade (4 pasos) na zona do líquido (sen sobrepasar o límite do debuxo do auga) e rebotando se toca o bordo:

Finalmente, a animación “Gas”, fai que a molécula se mova máis rápido e por todo o escenario, rebotando se toca un bordo do mesmo.

4.10. Sensor Me Temperatura e Me Humidade

A casa Makeblock dispón dosensor DHT11, que é quen de medir a temperatura (NTC) e humidade (resistiva) ambiente. As súas especificacións son as seguintes:

- 5V DC
- Rango temperatura: 0°C a 50°C \pm 2°C (só T^a positiva)
- Rango Humidade: 20% a 90% \pm 5%
- Precisión: 1% humidade, 1° Temperatura
- Compatible con calquera placa Arduino

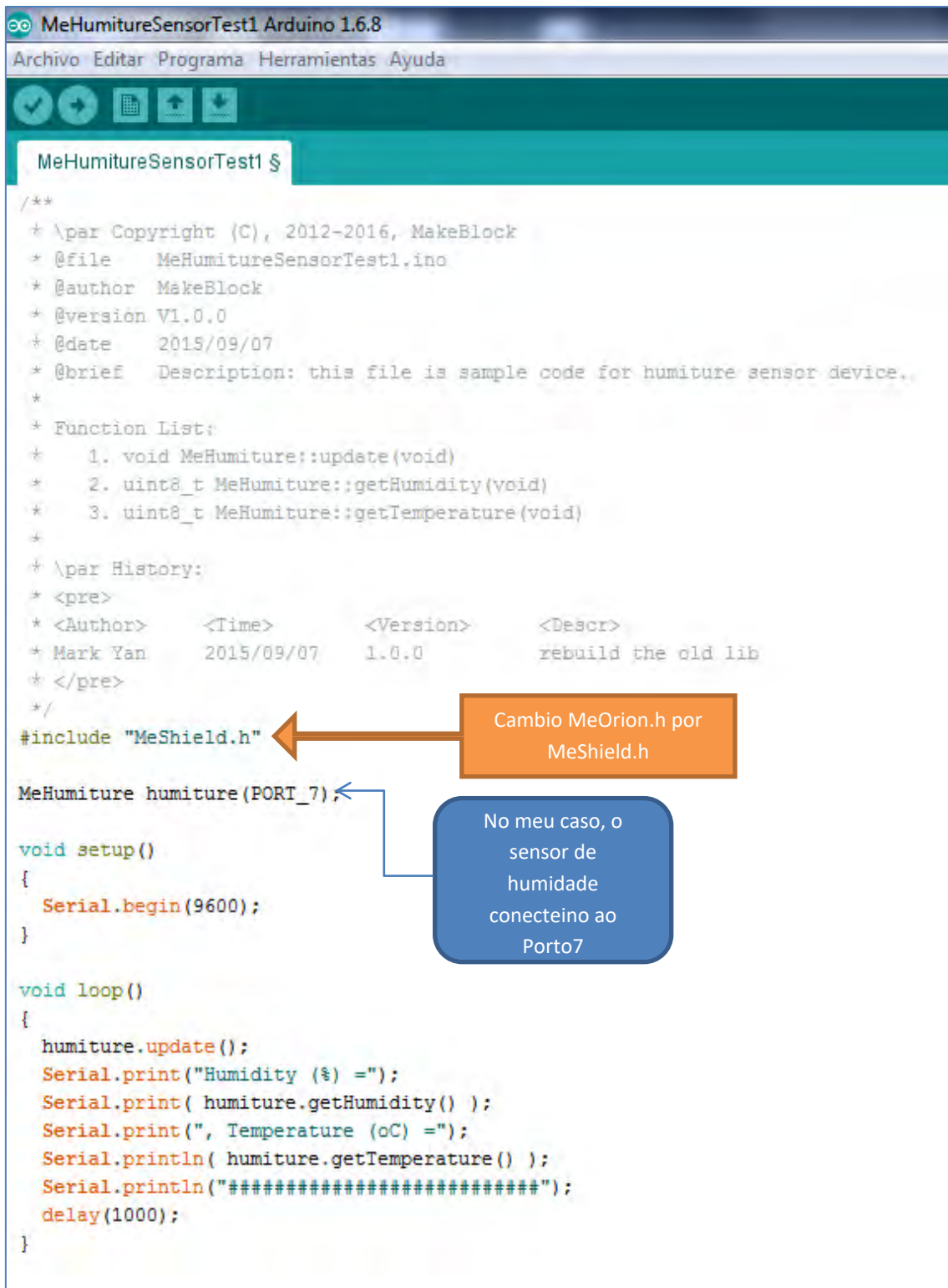


Sensor de temperatura e humidade

Este sensor proporciónanos unha saída dixital calibrada. A súa cor amarela no RJ25 significa que ten un porto simple dixital e necesita estar conectado ao porto con ID amarelo en Makeblock, aínda que, tamén pode conectarse a una placa Arduino Uno a través dos seus tres conexións GND, 5V e Data, sendo Data un pin dixital como por exemplo o pin3.

No seguinte exemplo vou usar o shield de Makeblock para unha Arduino Un, conectando o sensor de humidade e temperatura ao porto7. Nas librarías dispoñemos de 2 arquivos para o sensor de humidade. Usarei o *test 1*, que é o que nos proporciona a humidade e temperatura ambiente en °C (o *test2* apórtanolo en diferentes magnitudes para a temperatura). No *test 1* realizou unhas pequenas modificacións relacionadas coa placa e o porto:

Divertíndome con mBot: Guía de manexo e programación



```
MeHumitureSensorTest1 Arduino 1.6.8
Archivo Editar Programa Herramientas Ayuda

MeHumitureSensorTest1 $

/**
 * \par Copyright (C), 2012-2016, MakeBlock
 * @file MeHumitureSensorTest1.ino
 * @author MakeBlock
 * @version V1.0.0
 * @date 2015/09/07
 * @brief Description: this file is sample code for humiture sensor device.
 *
 * Function List:
 * 1. void MeHumiture::update(void)
 * 2. uint8_t MeHumiture::getHumidity(void)
 * 3. uint8_t MeHumiture::getTemperature(void)
 *
 * \par History:
 * <pre>
 * <Author> <Time> <Version> <Descr>
 * Mark Yan 2015/09/07 1.0.0 rebuild the old lib
 * </pre>
 */
#include "MeShield.h"

MeHumiture humiture(PORT_7);

void setup()
{
  Serial.begin(9600);
}

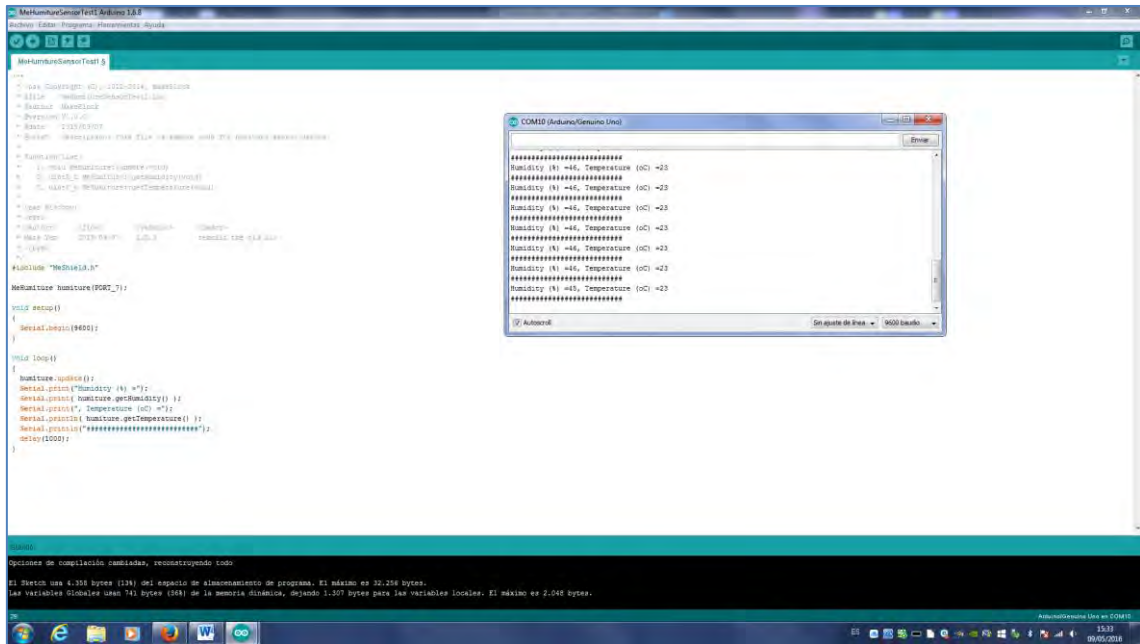
void loop()
{
  humiture.update();
  Serial.print("Humidity (%) =");
  Serial.print( humiture.getHumidity() );
  Serial.print(", Temperature (oC) =");
  Serial.println( humiture.getTemperature() );
  Serial.println("*****");
  delay(1000);
}
```

Cambio MeOrion.h por MeShield.h

No meu caso, o sensor de humidade conecteino ao Porto7

Na seguinte imaxe, referida ao Monitor Serie, podemos ver que nos aporta a temperatura e a humidade ambiente:

Divertíndome con mBot: Guía de manexo e programación



4.11. Módulo PIR

É un sensor pasivo que capta a luz infravermella do ambiente e reacciona aos seus cambios. Serviranos para detectar o movemento e basicamente o que fai é crear e transmitir un campo de luz infravermella que choca cos obxectos da habitación logrando detetar os cambios no eco que recibe. Ao alimentalo electricamente proporciona unha saída dixital (cero ou un), que pode ser regulada en sensibilidade mediante un potenciómetro. O seu ángulo de visión é de 90° e o seu radio de alcance sitúase entre 6 e 9 metros.

O sensor PIR de movemento da casa Makeblock úsase para detectar persoas ou animais nun rango de ata 6m. Se se move algo dentro dese rango de distancia, o sensor activa a saída dixital SIG a alto. Mediante un potenciómetro, soldado no módulo, poderemos axustar o rango de detección.

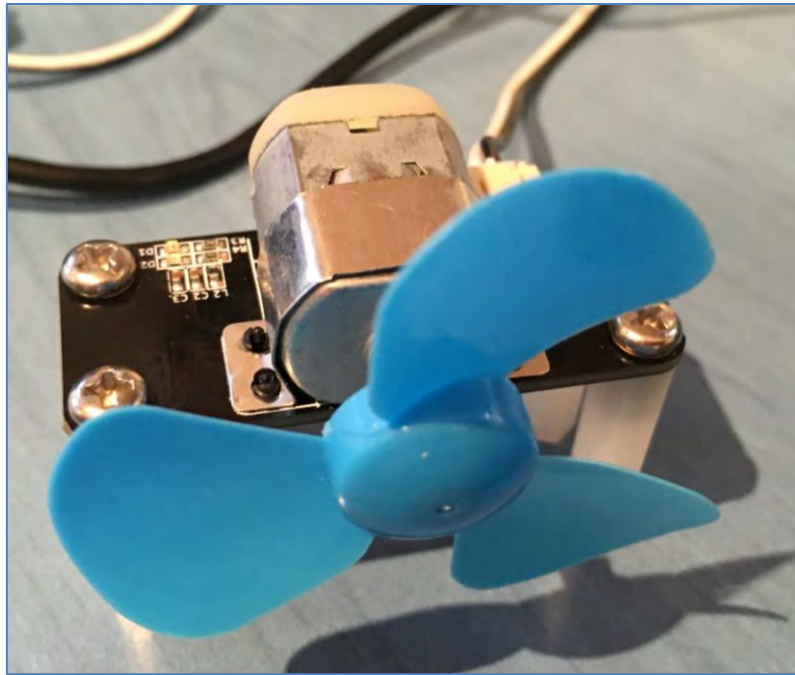
Importante: Despois de alimentalo, hai que esperar uns 10 segundos a que o sensor se inicialice.



Sensor Me PIR

O seguinte exemplo impleméntase sobre unha placa Makeblock Orion utilizando un módulo sensor Me PIR, un módulo servo motor co módulo adaptador RJ25 e un módulo 130 DC motor con hélice e que na seguinte programación conéctase a M1. O módulo PIR conéctase ao porto 4 e o servo, mediante o "Banco2" do adaptador, ao porto 1.

Divertíndome con mBot: Guía de manexo e programación



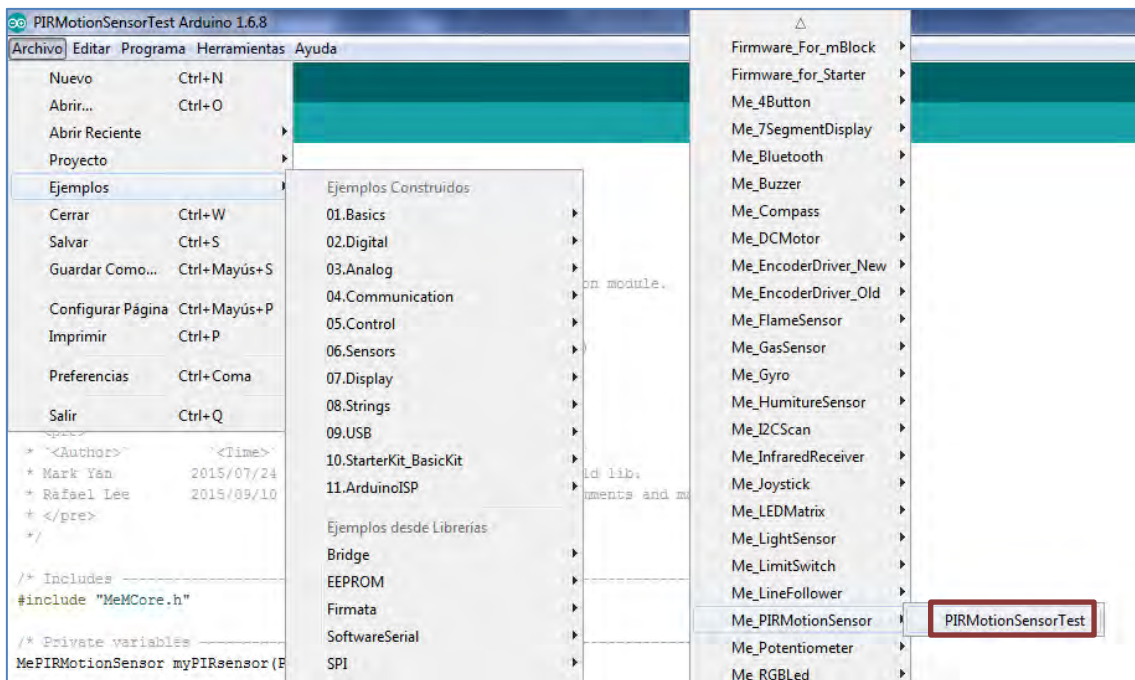
Módulo 130 DC con hélice

Cando o sensor PIR detecta o movement dun corpo, fai xirar o servo e o motor M1. En caso contrario, ambos páranse:

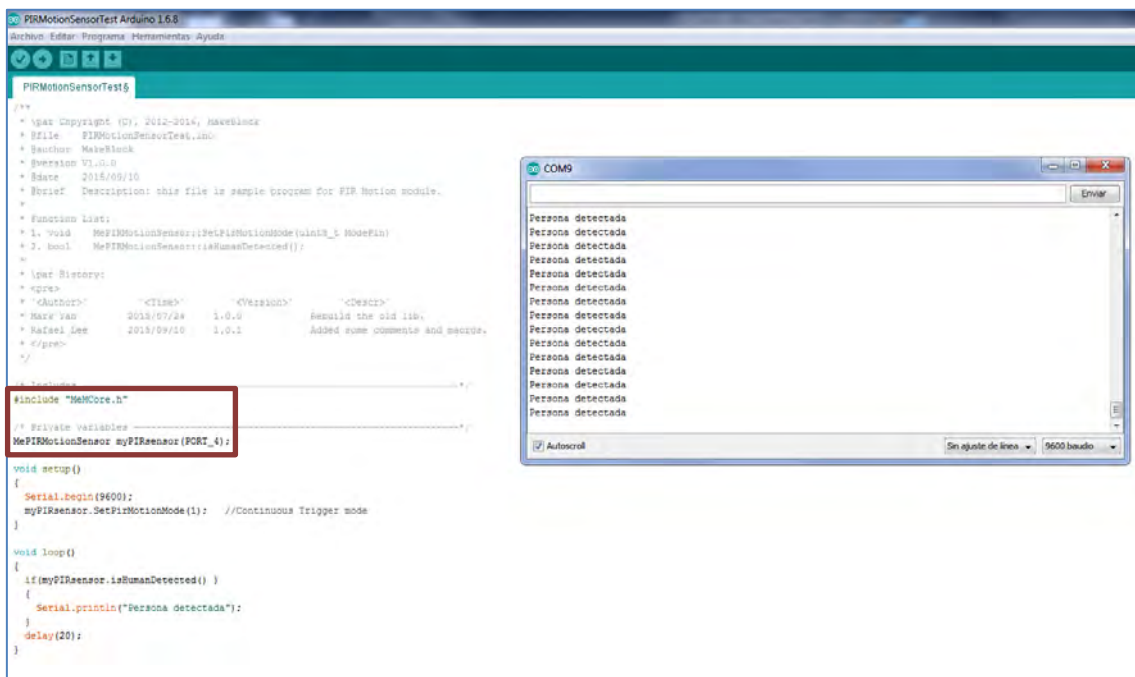
```
al presionar
por sempre
  si sensor de movemento pir Puerto4 > 0 entónces
    fijar servo Puerto1 Banco2 ángulo 100
    esperar 0.3 segundos
    fijar motor M1 velocidade 100
    esperar 1 segundos
  si sensor de movemento pir Puerto4 < 0 entónces
    fijar servo Puerto1 Banco2 ángulo 5
    esperar 0.3 segundos
    fijar motor M1 velocidade 0
    esperar 1 segundos
```


Divertíndome con mBot: Guía de manexo e programación

Dentro dos exemplos que nos proporciona a casa Makeblock nas súas librarías de Arduino atopámonos co arquivo *PIRMotionSensorTest*. Este arquivo creouse para que o sensor nos informe, mediante o porto Serial Begin, sobre a existencia dunha persoa na habitación:



Para comprobalo utilizarei a placa mCore conectando o sensor PIR ao Porto 4 (ID azul) da mesma. Despois cargarei o programa á placa desde o IDE de Arduino. O resultado é obvio, haberá detección porque eu estou presente, tal e como se amosa na seguinte imaxe:



Divertíndome con mBot: Guía de manexo e programación

4.12. Módulo Joystick

O módulo joystick pode usarse para, por exemplo, simular o movemento dun obxecto polo escenario. De feito, non é máis que unha panca de control de mando. No seguinte exemplo, creáronse 2 variables, referidas ás coordenadas X e Y que nos ofrece o módulo joystick.



Obsérvase que, inicialmente, o módulo ofrece as cantidades numéricas -2 e -18 para os eixes X e Y, respectivamente. Tamén sabemos que, no escenario scratch, nosa X varía entre -240 e 240 e a Y, entre -180 e 180. Así pois, tomando os seguintes puntos para ambos eixes, podemos calcular os parámetros m e n da función a programar:

X	Y=mx+n	X	Y=mx+n
-2	0	-18	0
240	490	490	180
Eixe X		Eixe Y	

Con estes puntos, os valores dos parámetros “m” e “n” no eixe X e no eixe Y son, respectivamente: 0,492 e 0,984 para o eixe X e, 0,354 e 6,378 para o eixe Y. Finalmente, o script para programar o joystick que está conectado ao puerto 7 que consiga que o sprite se mova polo escenario, é o seguinte:



4.13. Potenciómetro

O potenciómetro lineal d e Makeblock é, segundo nos informa a casa, de 50K Ω . É de 270 graos, sendo o mínimo valor cara a esquerda e o máximo valor alo tope da dereita. A súa cor de ID negro indícanos que só poderá conectarse aos portos 3 e 4 da placa mCore e aos portos 7 e 8 do shield de arduino.



Módulo Potenciómetro

Ao igual que noturos módulos, se non queremos usar o shield de arduino e pretendemos conectalo á placa Arduino Un, dispoñemos de 3 terminais: GND, VCC e AO; sendo esta última unha das saídas analóxicas da placa Arduino Un.

Un exemplo que nos axudará a programalo en arduino atoparémolo na súa librería, no arquivo *PotentiometerTest*. Simplemente, debemos modificar o nome da placa que usamos e o porto no que conectamos o potenciómetro.

Tamén podemos programalo con mBlock. No seguinte exemplo utilizo 0 valor numérico que me proporciona o potenciómetro para modificar a cor dos dous led RGB do mBot. A seguinte imaxe amosa o escenario do programa co valor máximo do potenciómetro (xirado 270 grados cara a derecha), que resulta de 975.



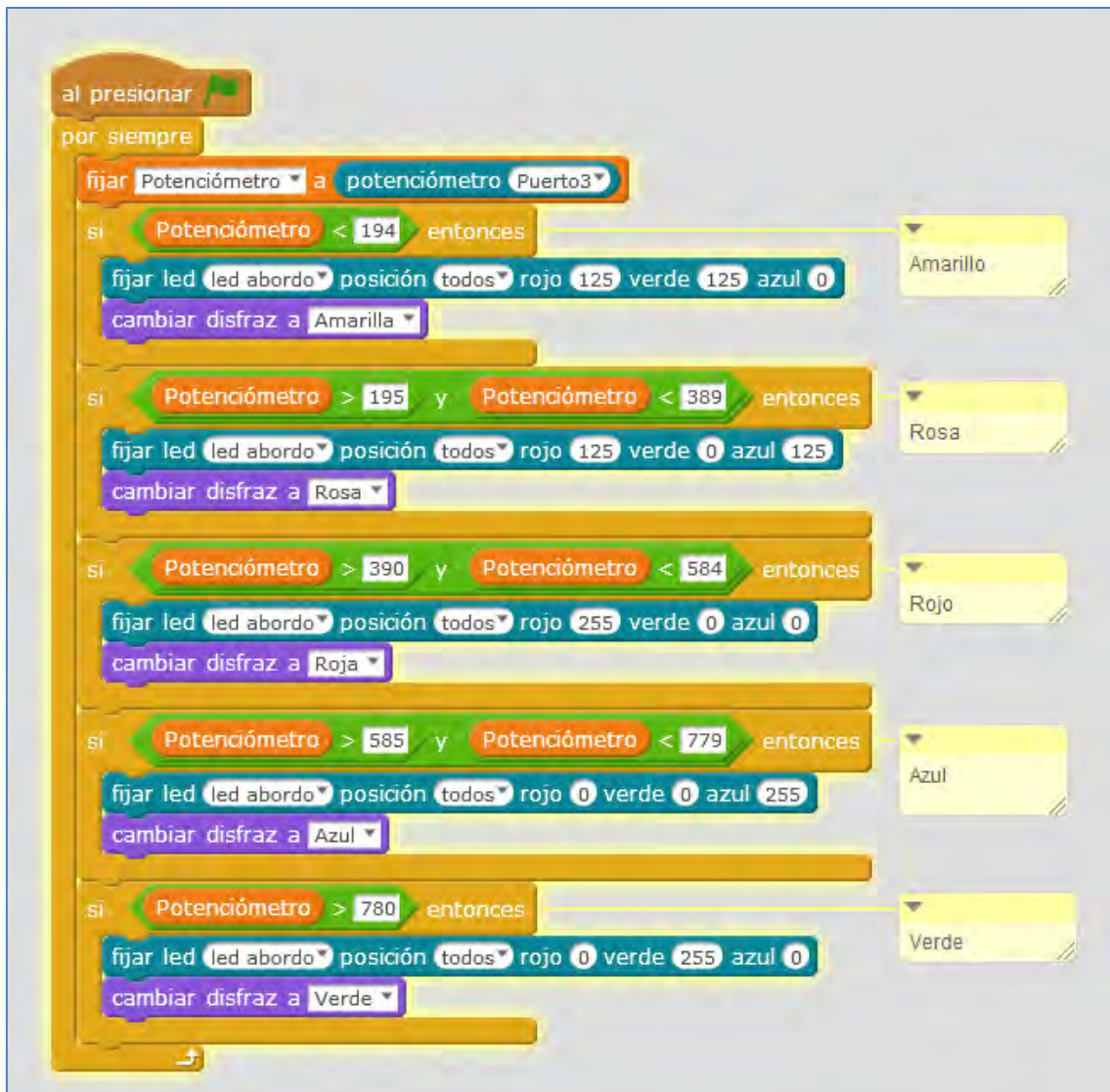
Divertíndome con mBot: Guía de manexo e programación

Como pode verse na seguinte imaxe, dispoño de 5 disfraces para percorrer os 975 puntos numéricos do potenciómetro. Decidín facer intervalos “simétricos” de módulo 195 ($975:5=195$) e, a cada intervalo asociarlle unha cor de LED RGB correspondente coa cor do disfraz:



Disfraces del objeto

O programa, para unha placa mCore del mBot, estando o potenciómetro conectado ao porto 3, é o seguinte:



4.14. Módulo 4LEDs RGB

Con este módulo que incluye 4 LEDs RGB, podremos controlar el brillo y el color de cada uno de sus LEDs de forma independiente.



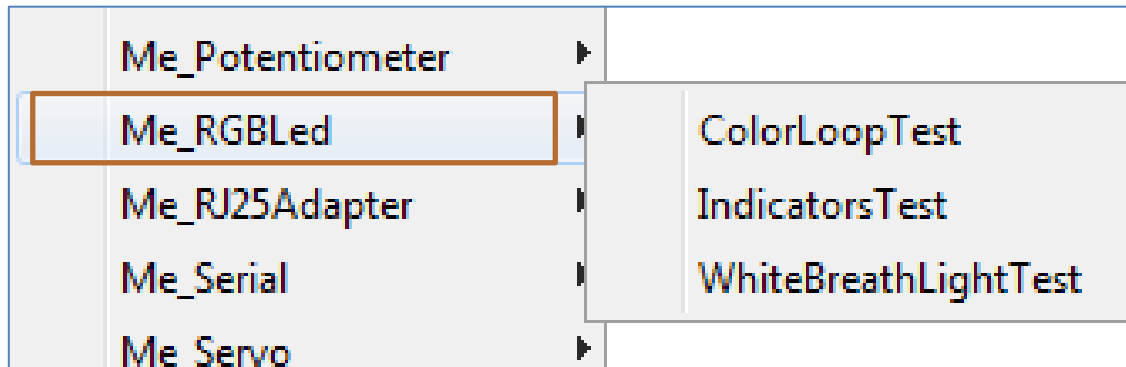
Módulo 4 LEDs RGB

Divertíndome con mBot: Guía de manexo e programación

O seu conector é de cor amarelo, así que, só o poderemos conectar aos portos que dispoñan desta color. Nunha placa mCore válenos calquera dos seus 4 portos.

Ao igual que noutros compoñentes, podemos usar este módulo directamente nunha Arduino Un, sen shield. Neste caso, as tres conexións amósansenos nun lateral: Vcc, GND e SIG, sendo o sinal un pin dixital da placa Arduino Un.

Nas librarías de Makeblok dispoñemos de tres programas de exemplo que nos axudarán a programalos desde o IDE de Arduino:



Desde mBlock podemos programar cada led de forma independente. Por exemplo, se o módulo está conectado ao porto 3 (cor amarelo), e queremos que o primeiro led teña a cor vermella, o segundo a cor verde, o terceiro a cor azul e o cuarto a cor branca, o programa que realizaría os nosos desexos sería o seguinte:



O programa anterior é moi simple e con mBlock podemos facer moito máis. Por exemplo, simular un arco da vella usando os leds de bordo e os do módulo 4 LEDs RGB. Este reto execútao o seguinte script:

Nel creanse 3 variables: LedAleatorio (que escolle aleatoriamente un LED dos 4 posibles do módulo 4LEDs), n (que por defecto toma o valor 1 dos leds de a bordo da placa e, grazas a n+1, selecciona o LED RGB 2 da placa mCore) e a variable cor (que tomará un valor ao chou entre 0 e 255).

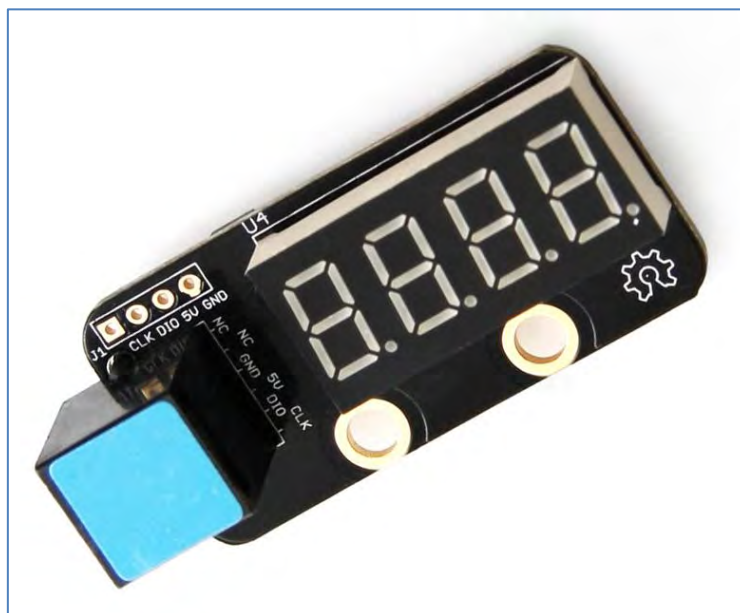
Divertíndome con mBot: Guía de manexo e programación

Todo este abano de cores permanecerá fixo na placa mCore (abordo) e no módulo 4 LEDs RGB (conectado ao porto 4) durante 1 segundo. Pasado ese tempo, prodúcese unha nova aleatoriedade de cor e elección de led principal:



4.15. Módulo Display 7 segmentos

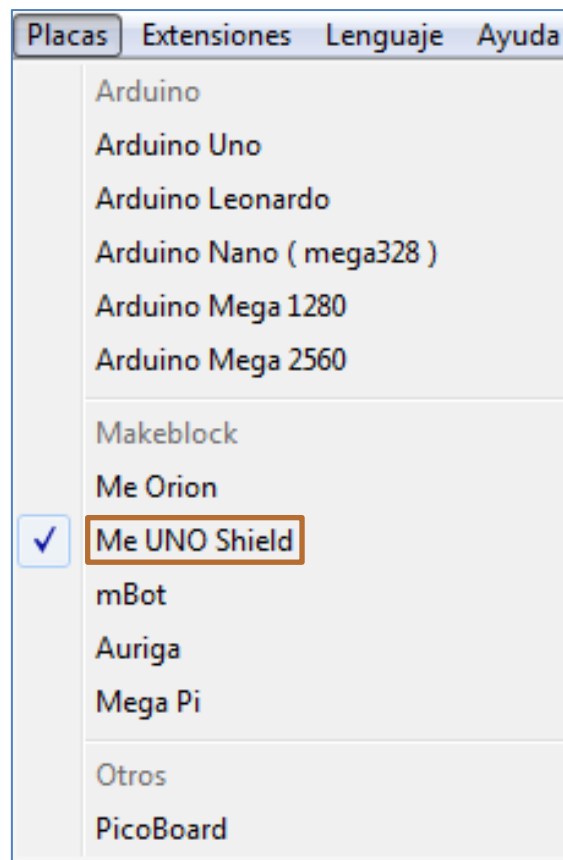
O módulo display 7 segmentos ten o seu ID azul (porto de sinal dixital dobre). Cor que nos indica que pode programarse coa placa mCore do robot mBot. Outras opcións son a placa Orion de Makeblock e a placa arduino un, por exemplo. A esta última, se se desexa, pódese acoplar o shield para arduino de Makeblock. Este display de 4 díxitos de ánodo común baséase no chip TM1637 e con él podemos controlar cada segmento do seu correspondente díxito, así como, o seu punto decimal.



Módulo Display 7 segmentos

Divertíndome con mBot: Guía de manexo e programación

Se usamos unha placa Arduino Un e queremos programala desde mBlock, debemos escoller a placa, neste caso *UN Shield*, e o porto de conexión correspondente para o noso Arduino Un, tal e como se amosa na seguinte imaxe:



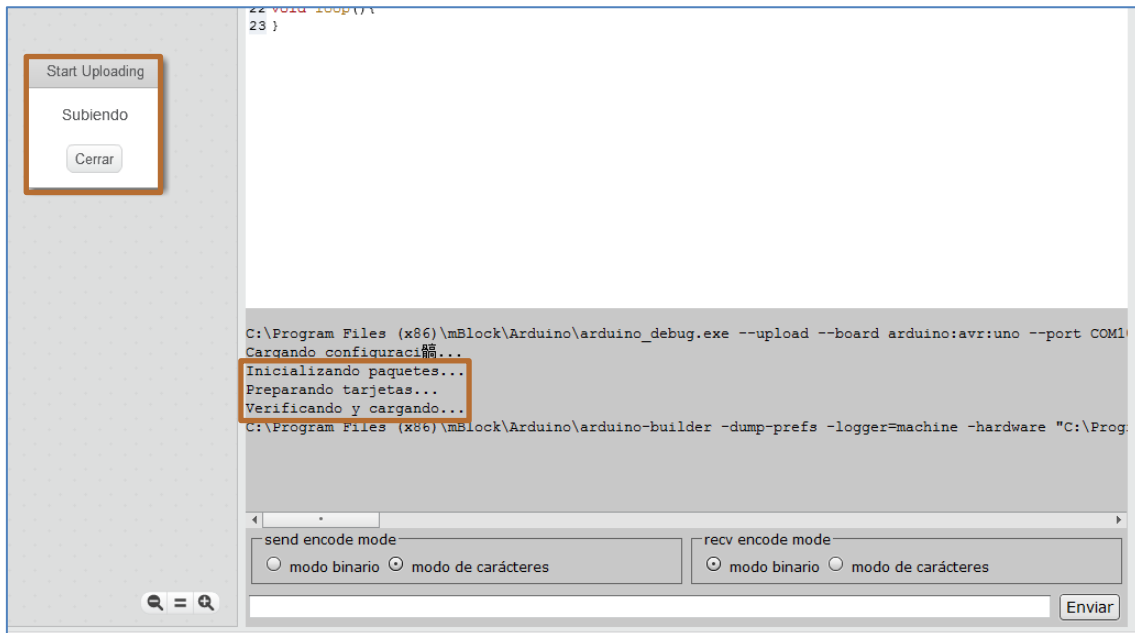
Supoñamos que queremos visualizar un contador ascendente de 100 a 110, usando un display 7 segmentos conectado ao porto 7 do shield de arduino. O programa en mBlock sería o seguinte:



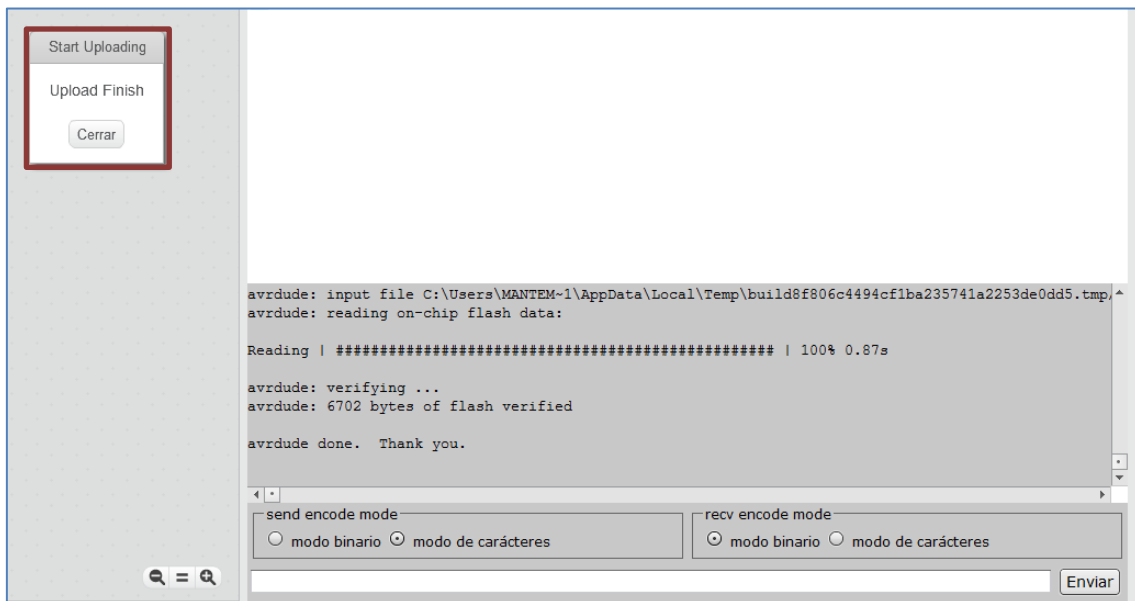
Para cargar o programa anterior na placa arduino, debemos facer clic no comando azul *UN Shield Program*. O programa infórmanos que está subindo o

Divertíndome con mBot: Guía de manexo e programación

programa: está cargando a nova configuración, inicializando os paquetes necesarios, preparando a tarxeta e verificando e cargando:



Cando finaliza a carga do programa á placa e, se non houbo fallos, a mensaxe que nos amosa é o de carga finalizada:



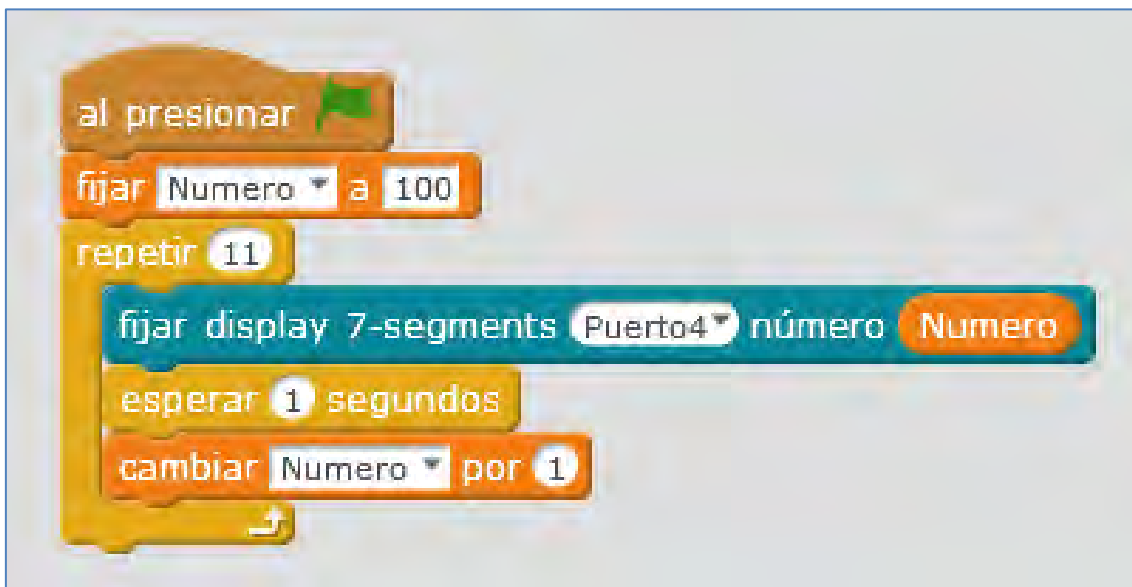
Se nos fixamos, o código arduino é o seguinte:

Divertíndome con mBot: Guía de manexo e programación

```
Back Upload to Arduino Editar con IDE de Arduino
01 #include <Arduino.h>
02 #include <Wire.h>
03 //include <Servo.h>
04 #include <SoftwareSerial.h>
05
06 #include <MeShield.h>
07
08 double angle_rad = PI/180.0;
09 double angle_deg = 180.0/PI;
10 double Numero;
11 Me7SegmentDisplay seg7_7(7);
12
13
14
15 void setup(){
16   Numero = 100;
17   for(int i=0;i<11;i++)
18   {
19     seg7_7.display(Numero);
20     delay(1000*1);
21     Numero += 1;
22   }
23 }
24
25
26 void loop(){
27
28
29 }
30
31
```

No noso display, veremos que van cambiando os números desde 100 a 110 nun intervalo de 1segundo cada un.

Se quixésemos usar a placa mCore, sen cargar o programa á mesma, o script que debemos executar coa bandeira verde é o seguinte: (notar que o display está conectado ao porto 4 da placa mCore)

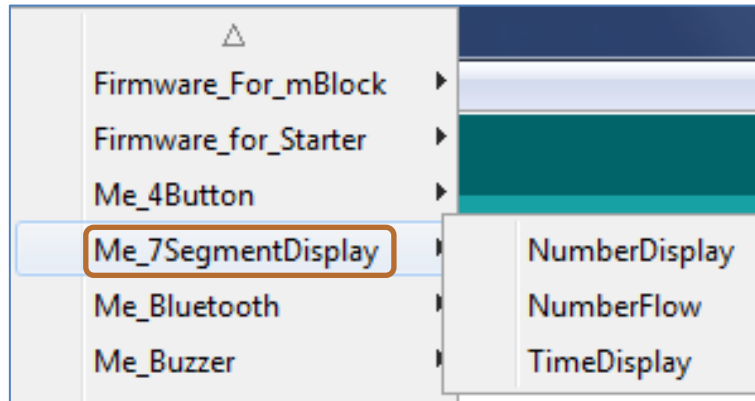


Como noutros módulos, tamén podemos conectar directamente o módulo display7 segmentos á placa arduino. Para iso, debemos conectar as 4 conexións do seu lateral á placa Arduino Un. Estas son: CLK (forma cadrada do sinal de reloxo),

Divertíndome con mBot: Guía de manexo e programación

DIO (liña de datos), 5V e GND; sendo os senais CLK e DIO pins dixitais da placa Arduino Un.

Se preferimos utilizar código arduino, podemos partir dos exemplos que se nos proporciona nas librarías da casa:



4.23. Sensor de Luz

Desenvolto sobre a base do principio de efecto fotoeléctrico en semicondutores, este sensor de luz pode utilizarse para detectar a intensidade da luz ambiente. É dicir, detecta facilmente se temos ou non a luz prendida ou apagada. O seu ID negro significa que ten un porto de sinal analóxica e necesita ser conectado ao porto con ID negro en Makeblock Orion, Auriga, mCore ou Shield de Arduino Un.



Módulo Sensor de Luz

No caso da mCore, podemos usar o puerto 3 ou 4. Se conectamos o sensor de luminosidade ao porto 3 e creamos un programa simple que nos amose a luminosidade numérica da habitación, con ou sen luz prendida, atoparémonos co seguinte resultado:

Divertíndome con mBot: Guía de manexo e programación



Luminosidade coa luz apagada

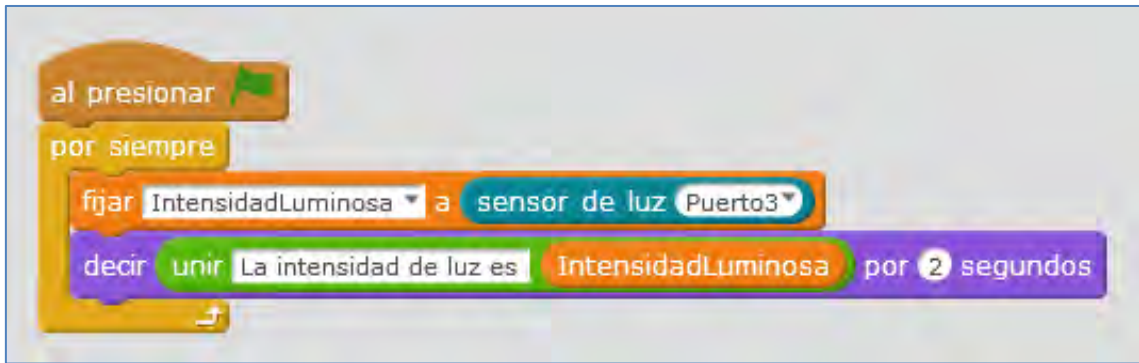


Luminosidade coa luz prendida

O rango de valores do sensor varía entre 0 e 980, de modo que, a maior valor, mayor luminosidade e a menor valor, maior escuridade.

O script que activa o sensor é o seguinte:

Divertíndome con mBot: Guía de manexo e programación



Como outros sensores, este compoñente tamén se podería conectar directamente á placa Arduino Un tendo en conta as súas 3 conexións do lateral: GND, Vcc e AO (soldadura cadrada). Sendo a sinal AO un pin analóxico de saída da placa Arduino Un. Finalmente, só dicir que na librería de Makeblock dispoñemos de tres programas que nos exemplifican a súa programación desde el IDE de Arduino:

Me_LightSensor	MeLightSensorTest
Me_LimitSwitch	MeLightSensorTestResetPort
Me_LineFollower	MeLightSensorTestWithLEDOn

