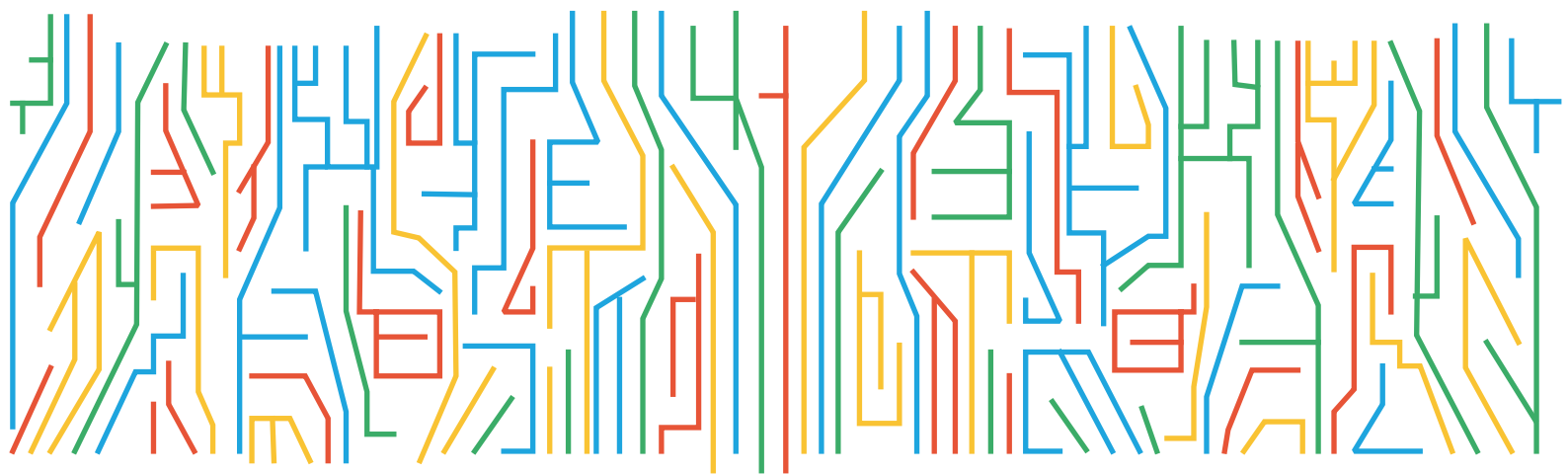


# ROBÓTICA

en primaria



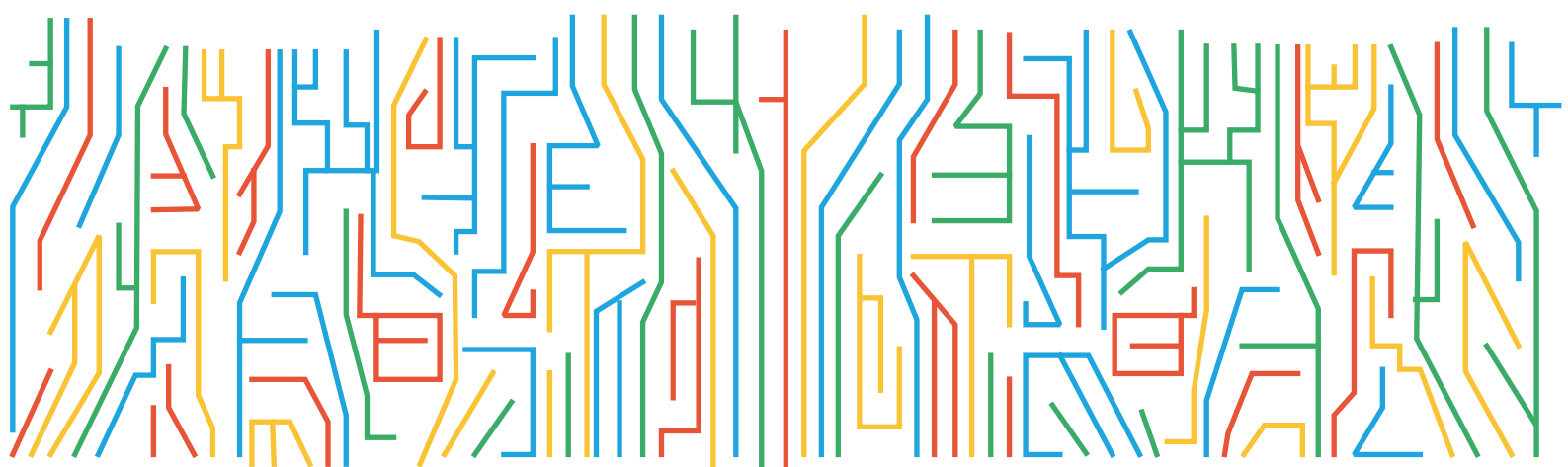


# ROBÓTICA

en primaria

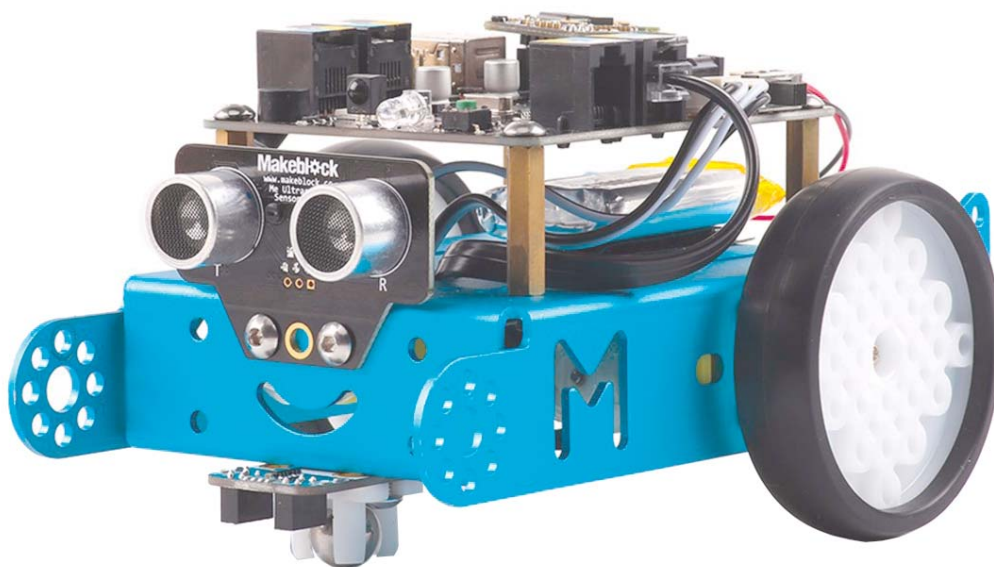
## ÍNDICE

GUÍA DIDÁCTICA DO PROFESORADO	1
ANEXO 1 DE PRÁCTICAS COMPLEMENTARIAS	64
ANEXO 2 DE PRÁCTICAS COMPLEMENTARIAS	80



# Guía didáctica del Profesorado

## INICIACIÓN A LA ROBÓTICA Y PROGRAMACIÓN



 **XUNTA  
DE GALICIA**

XUGA ANO: 2017  
CÓDIGO ARTÍCULO: 200206  
NOMBRE: EQUIPO ROBÓTICA EDUCATIVO PRIMARIA  
AÑO ADJUDICACIÓN: 2017  
Nº EXPEDIENTE: ED-05/17SU  
EMPRESA PROVEEDORA: MICROLOG TECNOLOGÍA Y SISTEMAS S.L.  
TELÉFONO DE ATENCIÓN AL USUARIO  
O MANTENIMIENTO: 91 759 59 10

 **MICRO-LOG<sup>®</sup>**

**MICRO-LOG TECNOLOGÍA Y SISTEMAS, S.L.**

C/ Andrés Obispo, 37 • 28043 Madrid  
Telf. 91 759 65 22 • Fax 91 759 54 80  
E-Mail: pedidos@microlog.es  
www.microlog.es

## **ÍNDICE**

### **0. IDENTIFICACIÓN DE LAS PIEZAS**

#### **1. SOFTWARE Y PROGRAMACIÓN**

- 1.1 Instalación en Linux
- 1.2 Instalación en Windows
- 1.3 Primeros pasos con mBlock
  - 1.3.1 Linux
  - 1.3.2 Windows
- 1.4 Instrucciones de mBlock
  - Control
  - Operadores
  - Robots
- 1.5 Como cargar una práctica del CD
  - 1.5.1 Entorno Linux Abalar
  - 1.5.2 Entorno Windows

#### **2. PRÁCTICAS DE INICIACIÓN CON LA PLACA**

- 2.1 Encendido y apagado de los LEDs RGB
- 2.2 Semáforo
- 2.3 Semáforo con aviso para invidentes
- 2.4 Semáforo con aviso para invidentes y botón para peatones
- 2.5 Arcoiris
- 2.6 Despertador solar

#### **3. PRÁCTICAS AÑADIENDO MÓDULOS A LA PLACA**

- 3.1 Cruce de semáforos
- 3.2 Activar un motor
- 3.3 Mover el servomotor con el Joystick
- 3.4 Control del ángulo del servomotor con el potenciómetro
- 3.5 Mostrar la distancia del ultrasonido en el Display
- 3.6 Cambio de color del LED RGB con el potenciómetro
- 3.7 Detector de presencia con el sensor PIR
- 3.8 Indicador con potenciómetro
- 3.9 Indicador de temperatura

#### **4. PROYECTOS ROBÓTICOS**

- 4.1 Robot infrarrojo
  - 4.1.1 Seguidor de línea negra
  - 4.1.2 Seguidor de línea blanca
- 4.2 Robot Ultrasonido
  - 4.2.1 Sensor de proximidad
  - 4.2.2 Explorador
  - 4.2.3 Sumo
- 4.3 Robot expresivo
  - 4.3.1 Matriz de LEDs
  - 4.3.2 Reloj Digital
  - 4.3.3 Medidor de distancias
- 4.4 Robot servomotor
  - 4.4.1 Control de un servomotor
  - 4.4.2 Portero
- 4.5 Robot control a distancia
  - 4.5.1 Control con mando a distancia
  - 4.5.2 Control por bluetooth

#### **5. APROXIMACIÓN A ARDUINO IDE**

- 5.1 Instalación de Arduino IDE
- 5.2 Cómo programar mi makeblock con Arduino IDE
- 5.3 Mostrar un mensaje en la matriz de LEDs
- 5.4 Mostrar los números del 1 al 100 en la matriz de LEDs

## 0. IDENTIFICACIÓN DE LAS PIEZAS

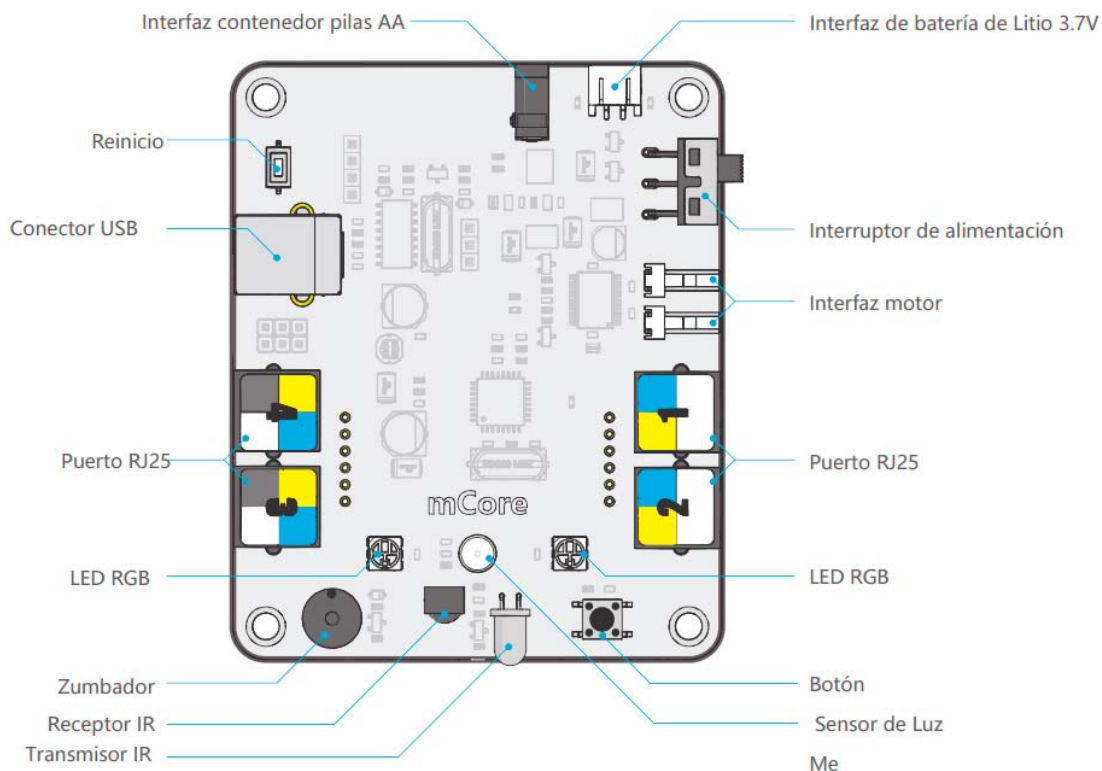
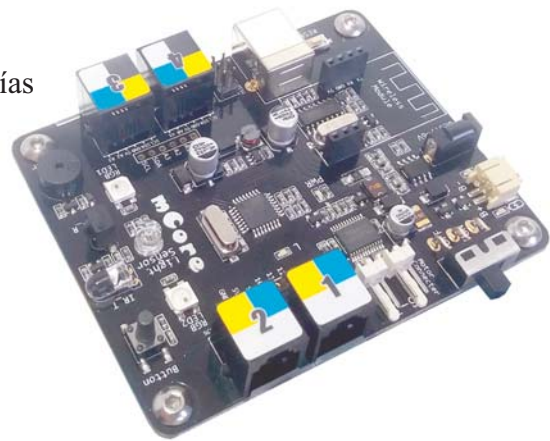
- **Placa controladora mCore:** La placa controladora es la encargada de dar vida a nuestro robot. A ella le conectaremos cada una de las partes que lo componen, y una vez que la programemos será la encargada de mandarle órdenes para que el robot se mueva o realice alguna acción concreta.

A esta placa le conectaremos diferentes componentes:

- Motores para que el robot pueda moverse
- Sensores para que el robot conozca el entorno por el que se mueve y pueda tomar datos que sean de utilidad para su cerebro electrónico (que es esta placa Mcore)

Para poder realizar esas conexiones dispone de las siguientes características:

- 2 Entradas para motores
- 4 Entradas numeradas para componentes electrónicos
- 1 Puerto USB
- 1 Interruptor On/Off
- Alimentación en portapilas de cuatro baterías recargables de 1.2V AA
- 2 LEDs RGB (Rojo-Verde-Azul)
- 1 Zumbador
- 1 Receptor IR
- 1 Transmisor IR
- 1 Pulsador
- 1 Sensor de luz



- **2 Motores con reductora:** Los motores son los encargados del desplazamiento de nuestros robots. Llevan asociado un eje al que aplica un movimiento giratorio continuo de 360°. En él podremos colocar una rueda, y cuando gire, moverá la rueda y desplazará nuestro robot



- **Servo motor:** Son motores cuyo eje sólo puede moverse entre 0° y 180°, es decir si colocásemos una rueda en ellos no daría una vuelta entera sino media vuelta. Una vez realizada esa media vuelta, tienen que volver a su posición inicial sobre su mismo eje. Estos motores no nos sirven para realizar el desplazamiento del robot, por que sólo realizarían un movimiento de vaivén hacia delante y hacia atrás con ligero desplazamiento.



Son utilizados para movimientos de más precisión como sería el movimiento de una articulación. Así por ejemplo si estuviésemos construyendo un brazo, el movimiento de un codo podría controlarse con un servomotor.

- **Matriz de LEDs:** Es una pantalla formada por 128 bombillas llamadas LEDs. Están colocadas formando una tabla o matriz de 8 filas y 16 columnas. Cada LED puede encenderse y apagarse por separado, pudiendo mostrar en la pantalla números, letras o dibujos, en función de las bombillas que encendamos y/o apaguemos.



- **Portapilas de 4xR6:** Sirve para alojar 4 pilas ó baterías tamaño AA. Necesitamos este portapilas para alimentar nuestra placa, ya que todo robot debe tener algún tipo de fuente de energía para poder funcionar.



- **Mando a distancia:** Utilizamos este mando para controlar los movimientos y acciones de nuestro robot sin necesidad de cables y desde una distancia prudencial.





- **Sensor ultrasonido:** Un sensor ultrasonido aporta a nuestro robot la capacidad de sentir si un objeto está cerca de él. Dispone de un emisor y un receptor de ondas, cuya misión es enviar esa onda y medir el tiempo que tarda en volver cuando ésta choca con el objeto más cercano. En función de ese tiempo, calcula la distancia que hay desde el robot hasta el obstáculo.



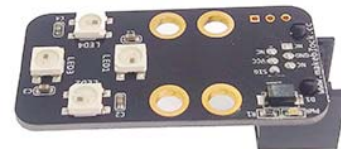
Nos sirve tanto para evitar obstáculos y que nuestro robot no se choque como para localizar objetos u obstáculos y dirigirse hacia ellos.

- **Sensor de línea:** Este sensor tiene la capacidad de detectar una línea negra sobre un fondo blanco o línea blanca sobre fondo negro. Para ello dispone de la electrónica necesaria para medir el reflejo de la luz sobre la superficie por la que se mueve el robot. En función del valor (de 0 a 1024) que envíe el sensor a la placa MCore del robot, ésta podrá decidir si el robot se encuentra sobre una superficie blanca o negra.

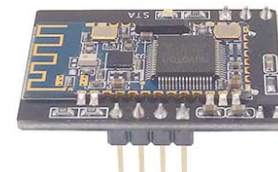


Podremos usar este sensor para crear robots que circulen sobre una línea negra/blanca siguiendo el camino marcado.

- **2 Módulos de 4 LEDs RGB:** Cada una de estas placas tiene 4 bombillas tipo LED con una característica especial: Pueden iluminarse variando su color. Las siglas RGB se traducen como Red-Green-Blue. Del mismo modo que cuando pintamos conseguimos un color mezclando dos pinturas, en el caso de estas bombillas LED disponen de 3 hilos de color (rojo, verde y azul) y en función de la cantidad de iluminación que indiquemos a cada uno de estos hilos conseguiremos que la bombilla nos muestre su luz de un color u otro cualquiera del arco iris.



- **Módulo bluetooth:** Con esta placa podemos controlar nuestro robot de manera inalámbrica desde una tablet, un móvil o un ordenador que tenga conexión bluetooth.



- **2 Módulos adaptador:** Cada una de las cajas cúbicas enumeradas de la placa MCore se llaman conexiones para cables de tipo RJ25. El servomotor y la sonda térmica no llevan conector RJ25 sino otro conector llamado JST.

Para poder conectarlos a la placa MCore, necesitamos este módulo adaptador. Por un lado conectamos los servomotores o la sonda térmica y por el otro el cable RJ25 que será el que conecte con la placa MCore.



- **Sensor de luz:** Con esta placa nuestro robot podrá saber si la luz del ambiente está encendida o apagada, y en función de la cantidad de luz que detecte, decidir cómo actuar (por ejemplo encender una luz propia de iluminación si detecta que está en un ambiente oscuro).



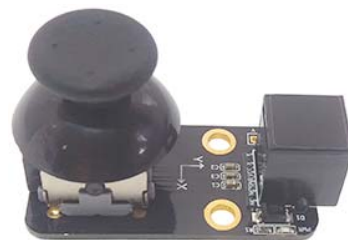
- **Sensor PIR:** Detectan movimientos en un entorno cercano. Se utilizan como sensores de presencia e interruptores. Si la placa detecta un cambio a través de su sensor infrarrojo, envía una señal a la placa MCore para que nuestro robot sepa que hay una presencia nueva cercana a él.



- **Módulo display:** Placa que contiene un visualizador de números de hasta 4 dígitos. Cada dígito está formado por una cuadrícula de 7 líneas que se llaman segmentos. Los números se visualizan iluminando los segmentos correspondientes.



- **Módulo Joystick:** Esta placa registra indicaciones de movimiento. Cuando movemos el mando hacia delante o hacia atrás, la placa MCore observará un cambio en la variable Y del bloque de programación asociado a esta placa. Si movemos el mando hacia la derecha e izquierda, observará un cambio en la variable X de su bloque de programación. Así, en función del contenido de esa variable decidiremos el movimiento de nuestro robot.



- **Módulo potenciómetro:** Contiene un mando que puede girarse hasta 270 grados. Esta placa nos devolverá un valor en función de la posición del mando que va desde 0 hasta 1024. En función de ese valor podemos programar la placa MCore para que cambien la intensidad de la luz de una bombilla LED, o mueva un servomotor.



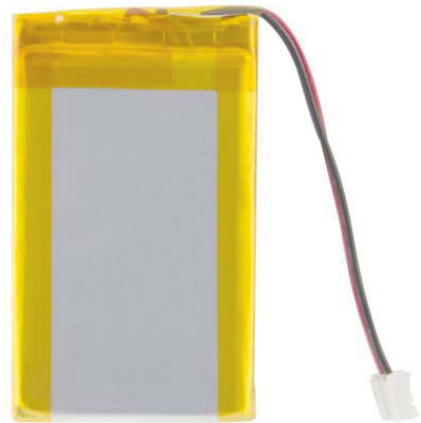
- **Sensor de temperatura:** Este sensor le indicará a la placa la temperatura del objeto en el que coloquemos la parte metálica de la sonda (en la punta del cable). Es útil para detectar temperaturas de objetos alejados de la placa.





- **Batería:** Proporciona 3,7V y 1450 mAh a la placa mCore cuando esté desconectada del PC y pueda funcionar de manera autónoma.

Se carga a través del puerto USB de la placa, para ello conectamos la batería a la placa mCore y con el botón en posición de apagado (OFF), conectamos la placa al PC mediante el cable USB. Se encenderá una luz roja indicando que se está cargando, cuando finalice la carga se apagará la luz roja y se encenderá otra de color verde.



- **Contenido del CD:**

- Software para Linux y Windows
- Prácticas resueltas
- Manuales y guía en PDF



- **Cable USB A-B**



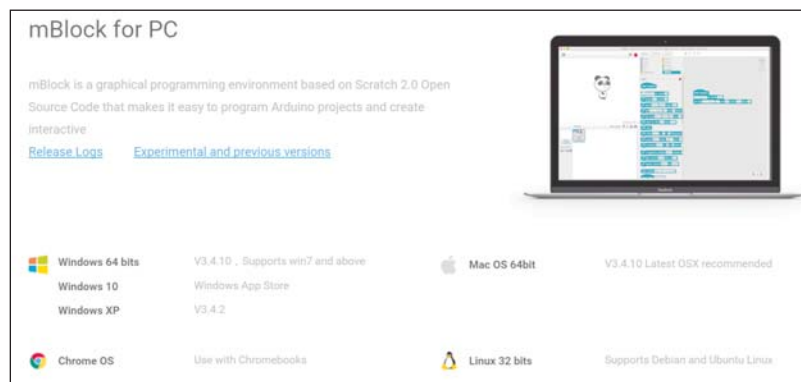
## 1. INSTALACIÓN DEL SOFTWARE

En el CD se incluyen los archivos de instalación del software tanto para Linux como para Windows.

No obstante recomendamos descargar la versión más actualizada del software en:

<http://www.mblock.cc/download>

Pulsando sobre el link del sistema operativo correspondiente descargarás la versión actualizada.



### 1.1 INSTALACIÓN EN LINUX (Versión ABALAR)

El proceso de instalación requiere de la clave de administrador de Linux (root). Si no dispone de esta clave, contacte con el responsable TIC del centro.

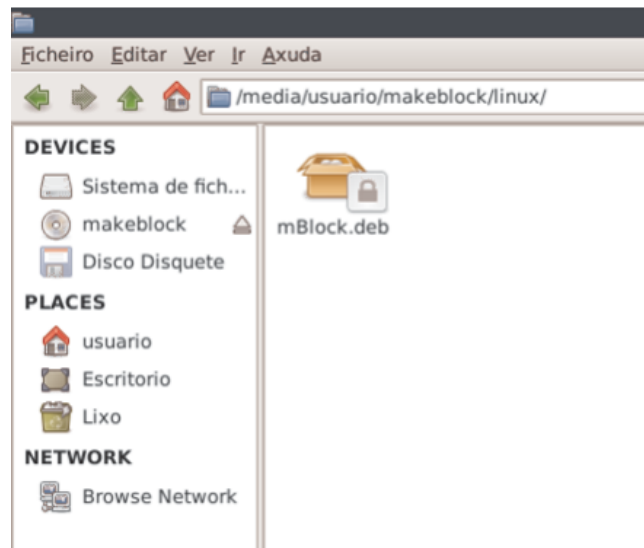
Introduce el CD en el lector de su PC y entre en el "**Xestor de ficheiros**".



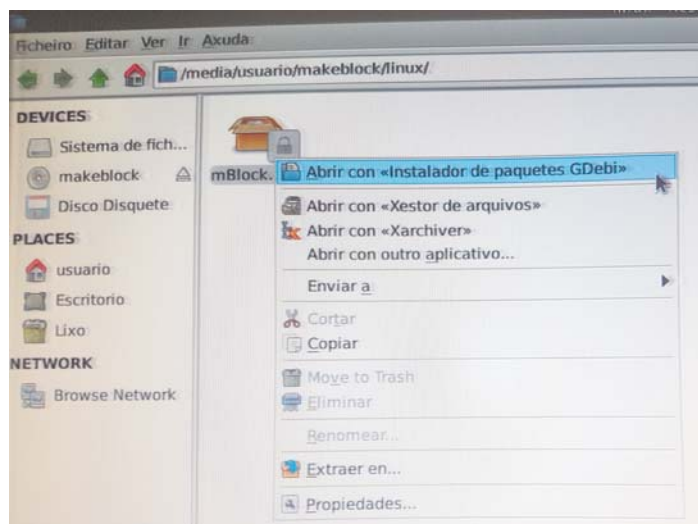
En DEVICES (dispositivos), seleccione el CD makeblock.



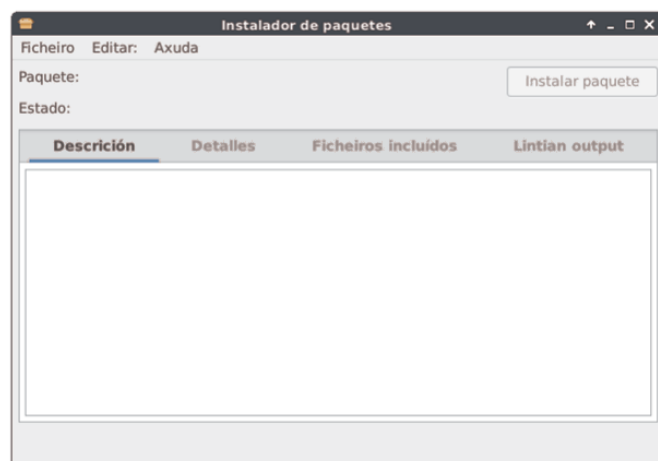
De las carpetas que aparecen, entra en linux haciendo doble click con el botón izquierdo del ratón sobre ella y aparecerá esta pantalla.



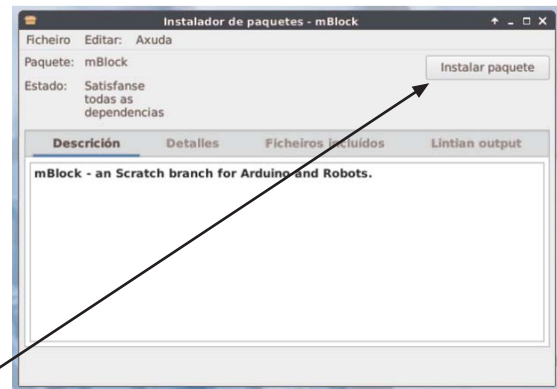
Con el botón derecho pulsa sobre mBlock.deb y selecciona "**Abrir con el instalador de paquetes Gdebi**", haciendo click, para proceder a la instalación del software mblock.



Se abrirá el instalador de paquetes:



- Esperamos unos segundos hasta que aparece mBlock dentro del instalador. Dependiendo de las características del PC esta espera podrá demorarse unos minutos.

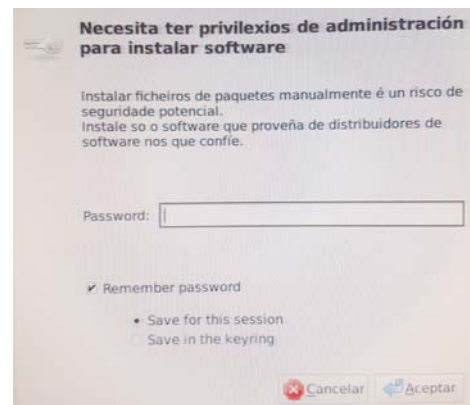


- Pulsamos en **"Instalar paquete"** y puede darse cualquiera de las siguientes situaciones:

- Opción 1: El sistema nos pide que introduzcamos la clave de administrador.

Si dispones de la clave, introdúcela. Si no, pide ayuda al responsable TIC de tu centro.

Y pulsamos en **"Aceptar"**.



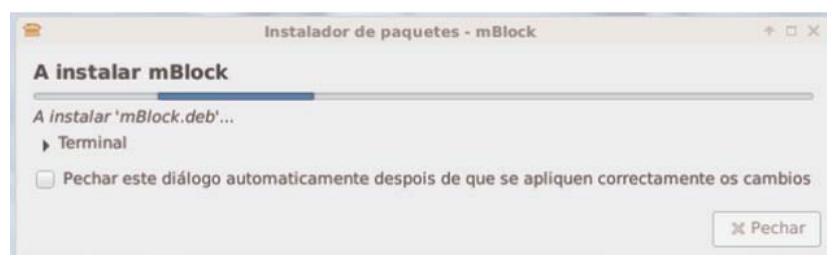
- Opción 2: El sistema tiene almacenada la clave de administrador y avisa con el siguiente mensaje.



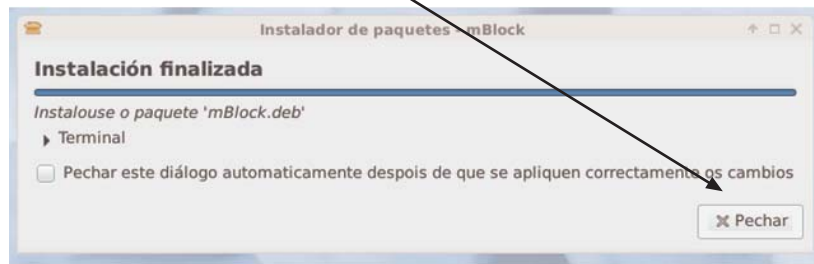
Pulsamos en **"Pechar"** y continuará el proceso de instalación.

- Opción 3: si ya estás trabajando en una sesión con permisos de administrador, es posible que no aparezca ningún mensaje y continúe la instalación automáticamente.

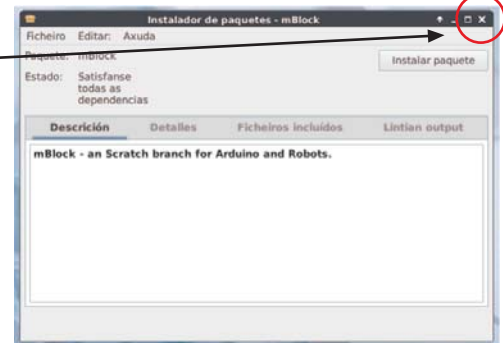
- A continuación comienza el proceso de instalación que no requiere de ninguna acción excepto esperar a que finalice. El proceso puede durar unos minutos.



- Una vez finalizada pulsamos en **"Pegar"**, puede demorarse unos minutos.



Si aparece esta pantalla, cerrarla pulsando en la X.

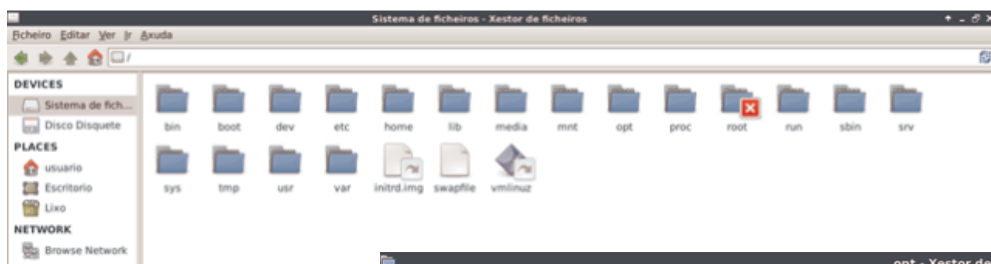


- Al finalizar la instalación tenemos que crear en el escritorio un acceso directo al programa mBlock para facilitar el acceso al alumnado.

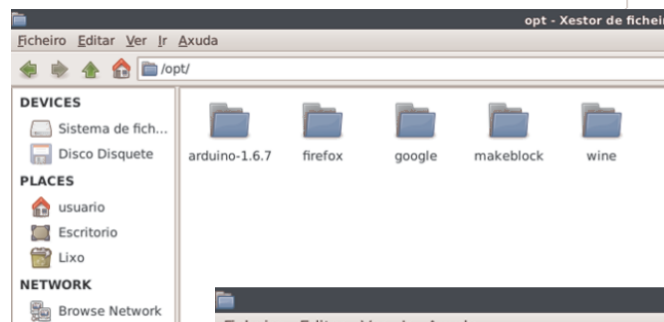
- Accedemos al **"Xestor de ficheiros"** y pulsamos en **"sistema de ficheiros"**, debajo de DEVICE.



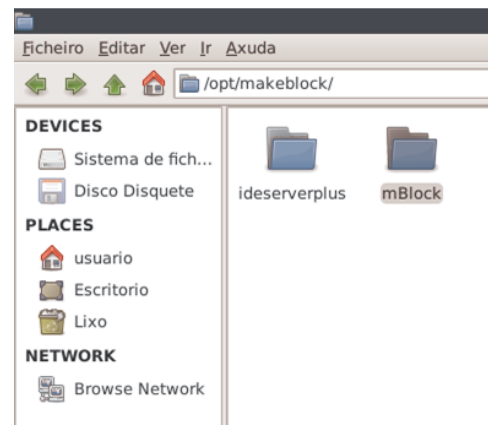
Aparecerá esta pantalla:



- Entramos en la carpeta opt.



- Entramos en la carpeta makeblock.

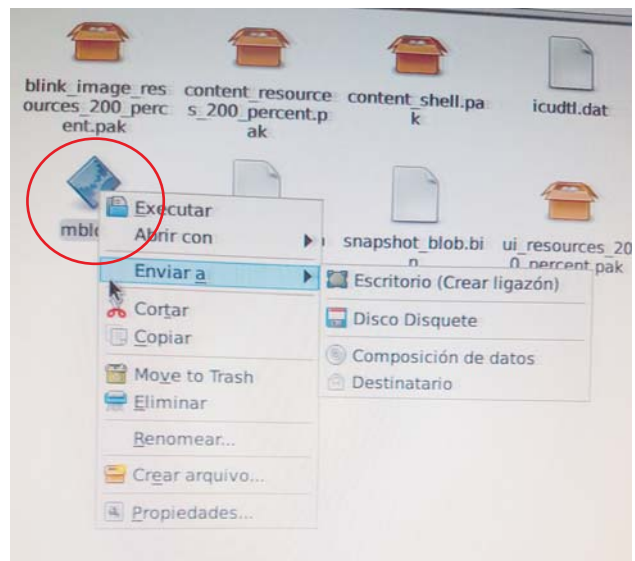




- Entramos en la carpeta mBlock.



Pulsamos con el botón derecho sobre el icono mblock y pulsamos en "enviar a" y después "escritorio (crear ligazón)".



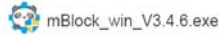
Y aparecerá el siguiente icono en el escritorio.



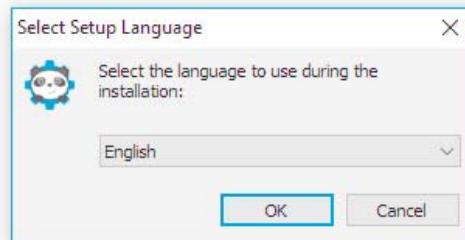
A partir de ahora para acceder al software basta con pulsar sobre el icono de mblock que hemos puesto en el escritorio.

## 1.2 INSTALACIÓN EN WINDOWS

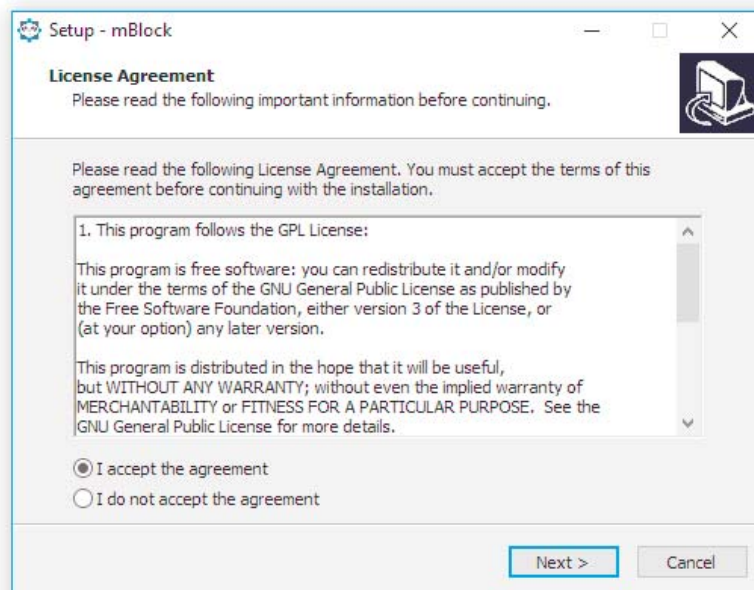
Localizamos en el CD el archivo mBlock\_win\_V3.4.6 y hacemos doble clic sobre él con el ratón para abrir la aplicación y comenzar la instalación.



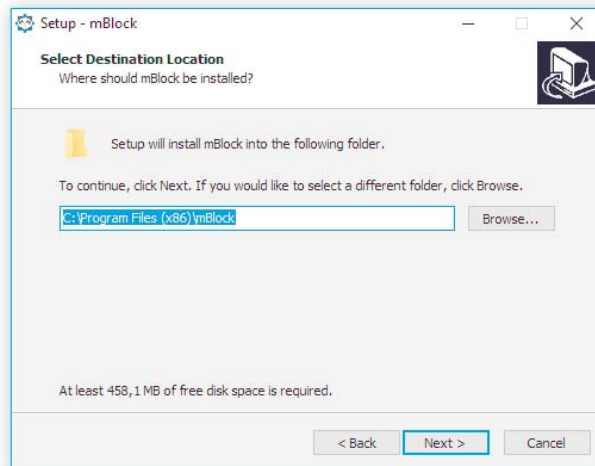
Seleccionar el idioma "English" y pulsar sobre el botón "OK".



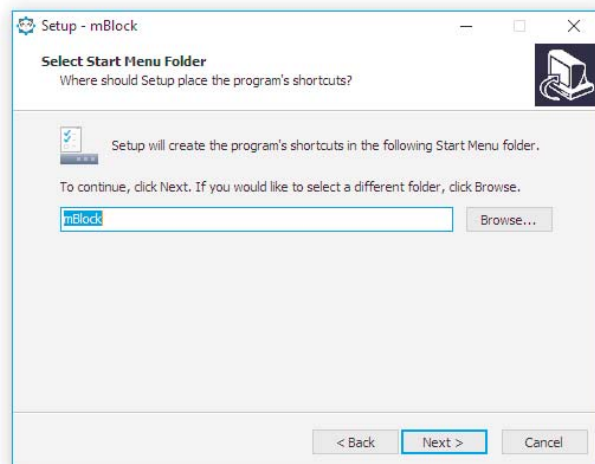
Aceptar el acuerdo de licencia pulsando sobre "I accept the agreement" y pulsar sobre el botón "Next".



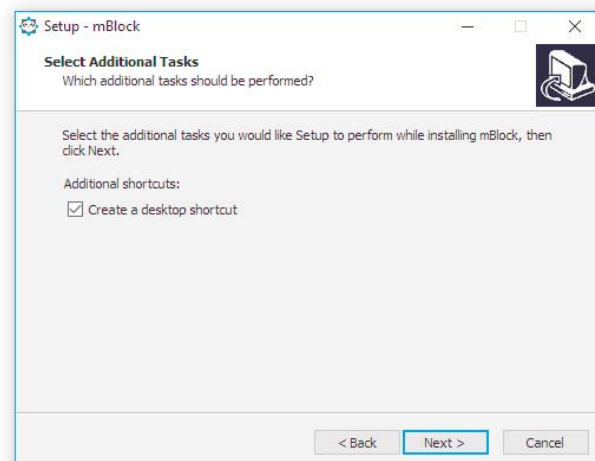
Seleccionar la carpeta donde queremos instalar el software o dejar la carpeta que viene por defecto. Pulsar sobre el botón "Next".



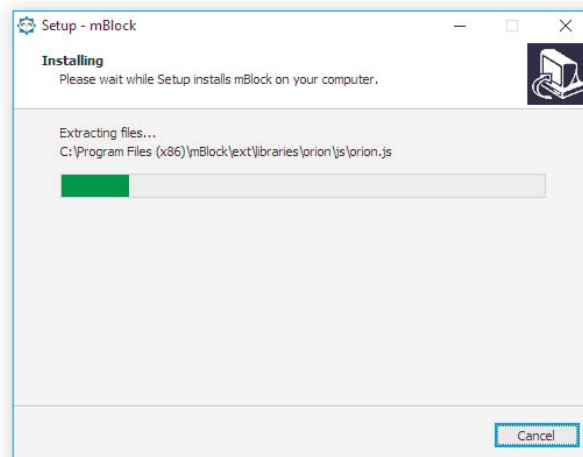
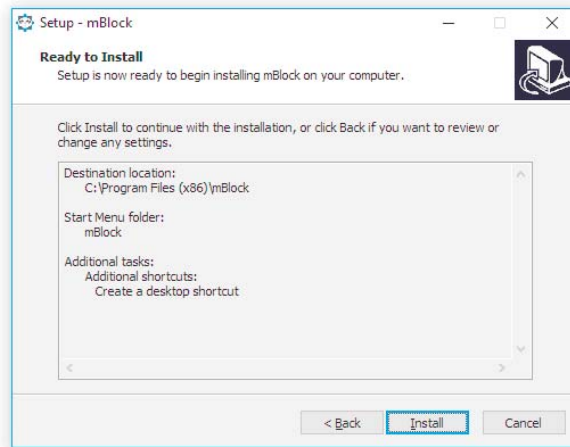
Seleccionar la carpeta donde queremos crear el acceso directo del menú de inicio de Windows o dejar la que viene por defecto. Pulsar sobre el botón "Next".



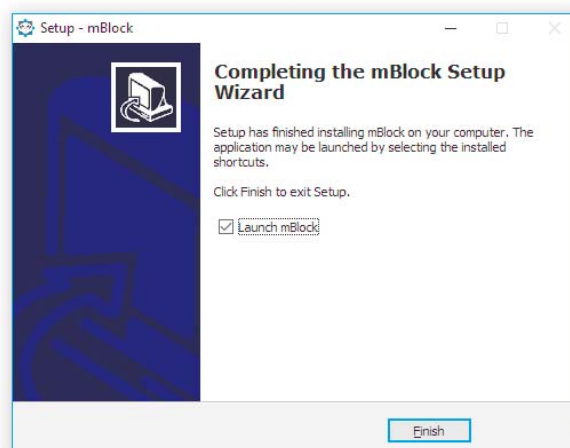
Hacer clic sobre "Create a desktop shortcut" si queremos crear un acceso directo al programa en el escritorio.



Revisar que todo está correcto y pulsar sobre el botón "Next".



Seleccionar si queremos ejecutar ahora el programa o no, y pulsar sobre el botón "Finish".

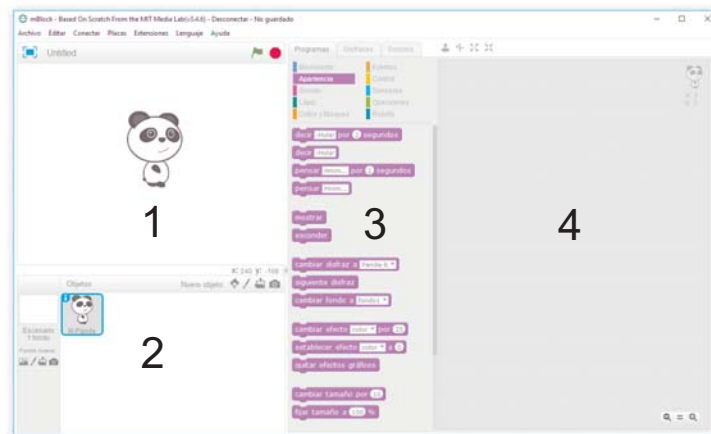


### 1.3 PRIMEROS PASOS CON MBLOCK

mBlock es un software libre parecido a Scratch que podemos utilizar para programar tarjetas Makeblock y Arduino.

Está basado en programación por bloques, y sus instrucciones han sido diseñadas con un lenguaje natural, eliminando términos técnicos y empleando una terminología más natural. Así se facilita el acceso a la programación en niveles educativos básicos.

Ejecutar mBlock.



Al igual que Scratch el entorno se divide en 5 zonas:

- Zona 1: Escenario, donde normalmente veríamos el resultado de nuestro programa cuando es virtual.
- Zona 2: Personajes, fondos... que van interviniendo en nuestro programa virtual.
- Zona 3: Conjunto de instrucciones, pulsando en cada uno de los botones de la zona superior se despliega todo el catálogo de instrucciones de la categoría seleccionada.
- Zona 4: Lugar donde estructuramos nuestro programa arrastrando las instrucciones de la zona 3.
- Zona 5: Panel desde el cual se carga el programa que vamos realizando en bloques en la tarjeta.

Para acceder a la zona 5, pinchar en el botón "Robots" de la zona 3 y pinchar encima de la instrucción "Programa de mBot".



Ahora se ha ocultado la zona 1 y 2, y nos aparece la zona 5.

Tenemos tres botones en la zona 5:

**Atrás:** volver a visualizar las zonas 1 y 2.

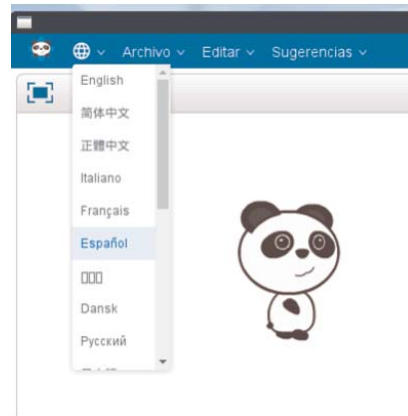
**Subir a Arduino:** guardar el programa en nuestra placa controladora.

**Editar con IDE de Arduino:** traduce nuestro programa al lenguaje de Arduino IDE.

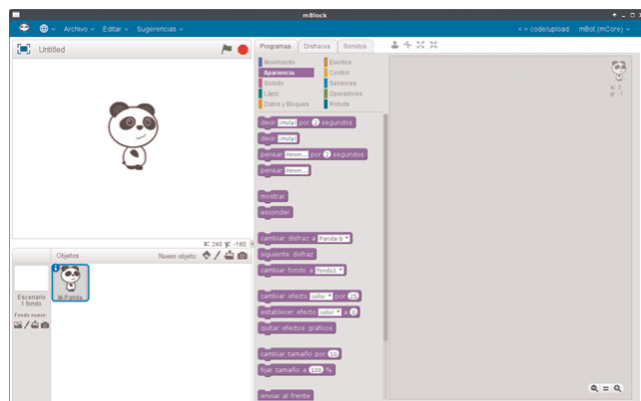


### 1.3.1 VERSIÓN DE LINUX

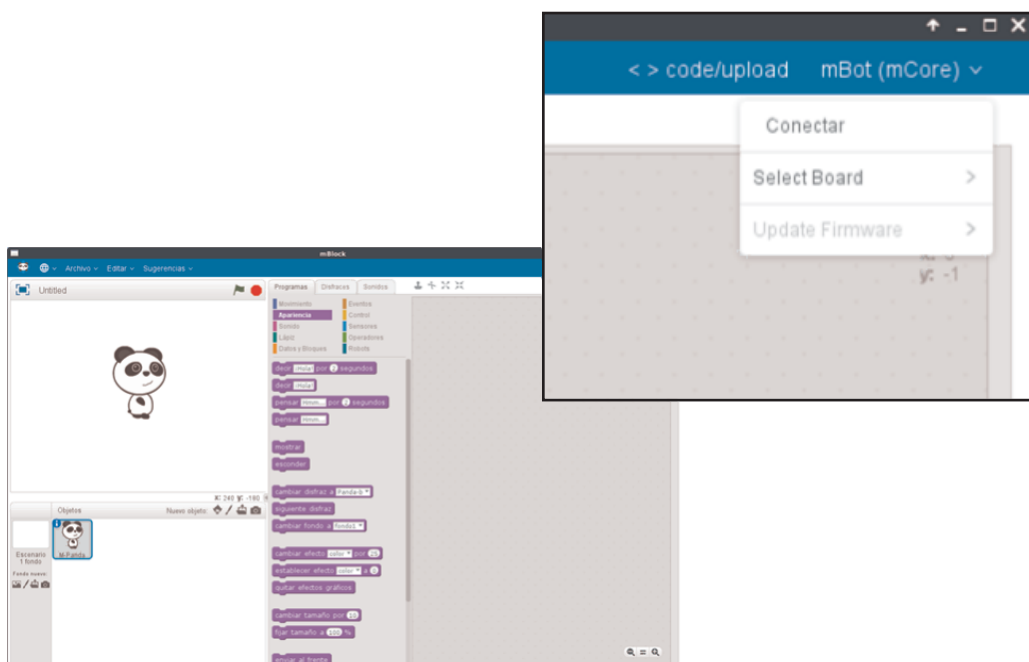
Podemos cambiar el idioma del software pulsando con el boton izquierdo en el icono de la esfera, y a continuación haciendo click sobre el idioma deseado.



El software para linux es muy parecido a la versión de windows. Las zonas son las mismas, lo único que cambia es el menú principal.

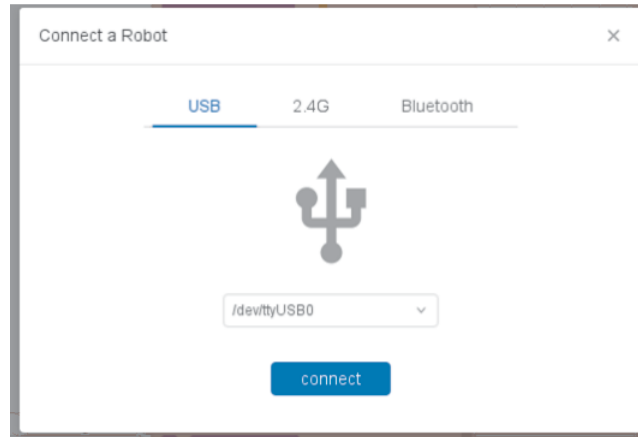


Para poder utilizar el robot con makeblock, lo primero que haremos es conectar el robot al puerto USB del ordenador. Pulsando sobre **"mBot (mCore)"** se despliega un menú.

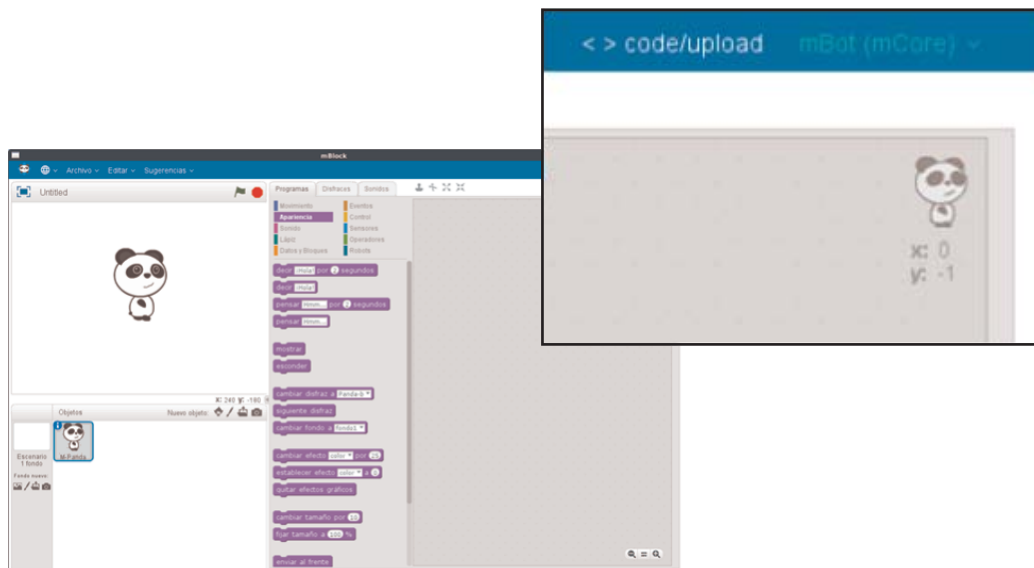


Encendemos colocando su interruptor en posición “on”.

Seleccionamos la placa y le damos a "**Conectar**" y podremos elegir entre USB, 2,4G o Bluetooth. Recomendamos el uso de la conexión USB.



En unos segundos nos aparecerá el puerto USB en el que tenemos conectado el robot. Pulsamos en conectar y una vez realizada la conexión, se cierra la ventana automáticamente y vemos que aparece mBot (mCore) en color verde.



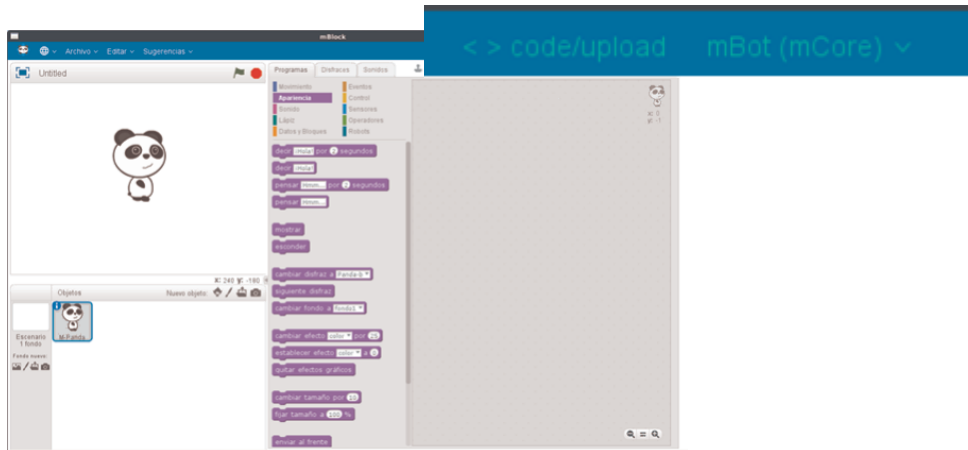
La placa de control que estamos programando se llama mCore y tenemos que asegurarnos de que es la que aparece en verde. En caso de aparecer otra placa, pulsando sobre mBot (mCore), aparece la opción "Select Board" en el menú. Pulsando esa opción veremos el conjunto de tarjetas que podemos utilizar con este software. Seleccionaremos mCore para poder trabajar correctamente.

A continuación actualizaremos el firmware para asegurarnos el correcto funcionamiento de la placa. Pulsando sobre mBot (mCore), desplegamos el menú cuya última opción es "Update firmware". Al pulsar sobre esta opción, seleccionamos "for mbot" y esperamos el proceso de carga.

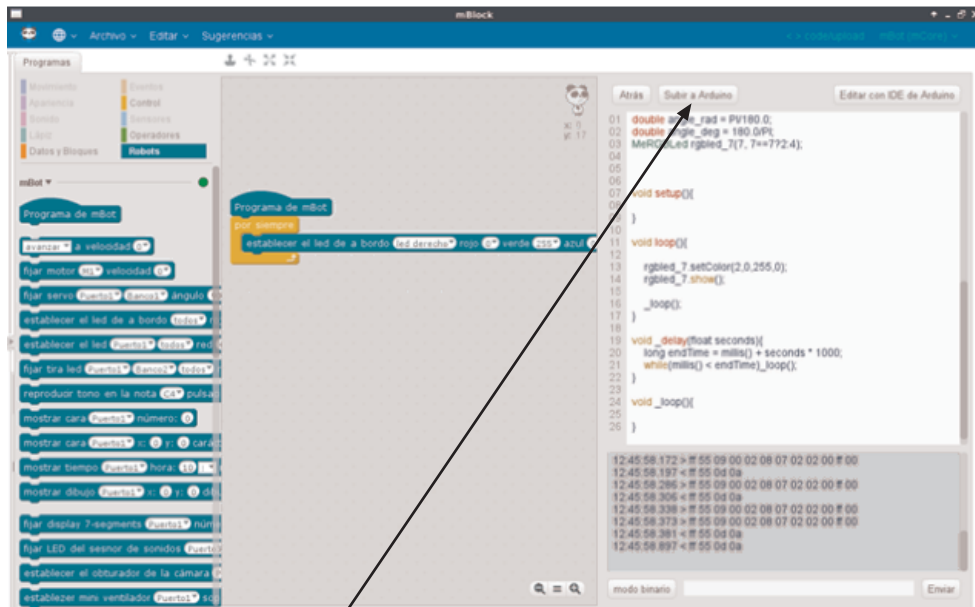
Una vez finalizado ya podremos trabajar con la tarjeta.

A la hora de programar podemos realizar dos tipos de programas:

- Programas no autónomos con scratch: Comienzan con la instrucción "al presionar bandera" y se ejecutan haciendo click sobre la bandera verde. No requieren de proceso de carga.
- Programas autónomos: Comienzan con la instrucción "programa de mbot" y requieren ser cargados en la placa para poder ejecutarlos. Para acceder al proceso de carga, pulsamos sobre code/upload.



Aparece la siguiente pantalla.



Si pulsamos sobre "**subir a arduino**" el programa se cargará en el robot y éste lo ejecutará de forma autónoma, pudiendo funcionar sin estar conectado al PC.

### 1.3.2 VERSIÓN DE WINDOWS

Accedemos a mBlock haciendo click sobre su icono en el menú de aplicaciones de windows.

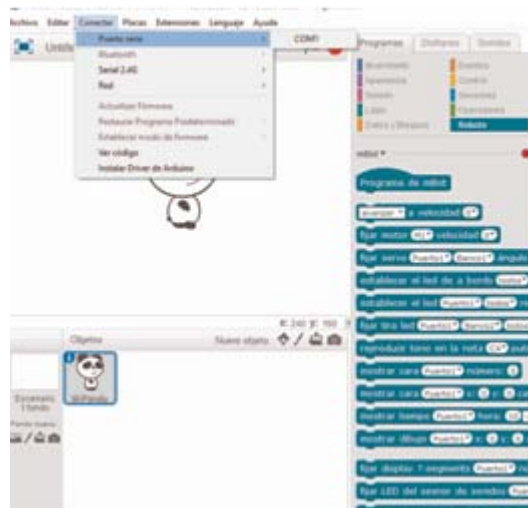


El programa comienza configurado en inglés. Si queremos cambiar el idioma, podemos seleccionar otro idioma desde el menú "Lenguaje".



Para empezar a trabajar tenemos que comprobar que el equipo detecta la tarjeta mCore que será reconocida como un puerto COM.

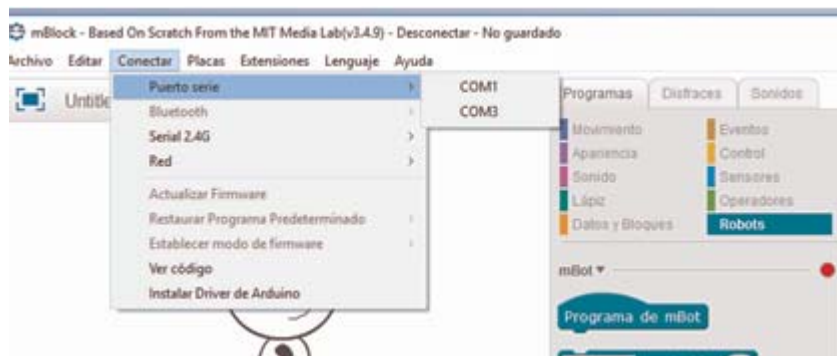
Para comprobarlo primero tenemos que ver qué puertos COM detecta mblock sin conectar la tarjeta, haciendo clic con el botón izquierdo del ratón en "Conectar" y después "Puerto serie".



En nuestro caso vemos que el equipo sólo tiene utilizado el puerto COM1.

A continuación conectamos la tarjeta mCore del robot al ordenador utilizando el cable USB y la encendemos colocando su interruptor en posición "on".

El sistema reconocerá la placa como un nuevo puerto com (normalmente COM3). Para comprobarlo hacemos click con el botón izquierdo sobre **"Conectar"** y después en **"Puerto serie"**.



Pulsamos sobre **"COM3"** con el botón izquierdo del ratón para seleccionar ese puerto como medio de comunicación con la placa.



Comprobamos que la aplicación está configurada para programar placas mCore pulsando en placas y seleccionando mCore haciendo clic con el botón izquierdo del ratón.



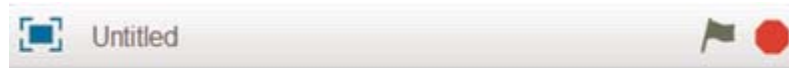
A la hora de programar el robot podemos hacer dos tipos de programas:

- Programas no autónomos: Son programas cuya primera instrucción es **"al presionar bandera"**.





La placa debe estar siempre conectada al PC con el cable USB y el robot cumplirá las órdenes de nuestro programa cuando hagamos clic con el botón izquierdo del ratón sobre la bandera de la primera instrucción de nuestro programa o sobre el símbolo de bandera del escenario.



Para poder trabajar con este modo, lo primero que tenemos que hacer es asegurarnos que la placa contiene el firmware cargado. Para ello pulsamos en **"Conectar"** y después en **"Actualizar firmware"**.



- Programas autónomos: Son programas cuya primera instrucción es **"Programa de mBot"**.

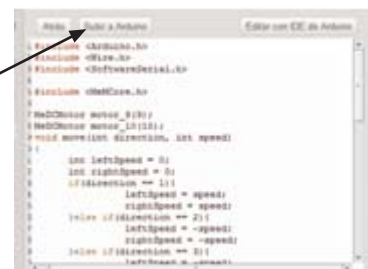


Se cargan dentro del robot, y una vez desconectado el cable USB del PC, el robot realiza las órdenes del programa.

Para cargar el programa en el robot, hacemos clic con el botón izquierdo del ratón sobre la primera instrucción de nuestro programa: "programa de mbot" y accedemos a la siguiente pantalla.

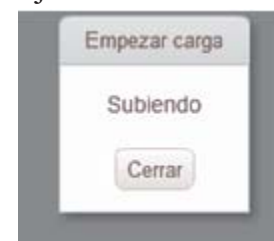


Para cargar el programa en la placa, hacemos click con el botón izquierdo del ratón sobre el botón **"Subir a Arduino"**.



Mientras se realiza el proceso de carga se visualizará el siguiente mensaje:

- Cuando el mensaje cambie por "subida finalizada" pulsaremos sobre el botón **"Cerrar"**.



## 1.4 INSTRUCCIONES MBLOCK

### CONTROL

#### Espera 1 segundo

Genera una pausa de un segundo durante la ejecución del programa.



#### Bucles

Las instrucciones de tipo bucle son aquellas que suponen la repetición de una secuencia de instrucciones. Pueden ser bucles de tipo infinito o bucles que se realizan un determinado número de instrucciones.

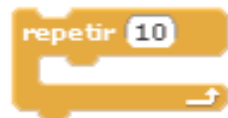
#### Por siempre

Genera un bucle infinito dentro del programa. La secuencia de instrucciones que incluyamos dentro de esta instrucción se realizará continuamente sin final.



#### Repetir

Realiza un determinado número de veces las instrucciones contenidas dentro del bucle.



#### Condicionales

Son instrucciones en las que se evalúa una condición, y en función de si ésta es cierta o falsa, se procede a realizar unas instrucciones u otras.

#### Si

Sólo se hacen las instrucciones contenidas dentro del “si” en caso de que sea cierta la condición que se incluye junto al “si”. Esta condición normalmente será una comparación de tipo numérico.



#### Si/Si no

En este caso, disponemos de dos juegos de instrucciones. Las instrucciones que introducimos bajo el “si” se hacen en caso de que la condición sea cierta. En caso de que sea falsa, se hacen las instrucciones bajo el “si no”.



## OPERADORES

En esta sección encontraremos instrucciones que utilizaremos para realizar operaciones matemáticas y comparaciones de tipo lógico para su uso en condiciones.

Las instrucciones redondeadas son operaciones matemáticas.



Las instrucciones con los bordes puntiagudos son comparaciones que utilizaremos para generar condiciones en bucles e instrucciones condicionales. Pueden enlazarse varias condiciones con operadores de tipo:

- **y**: deben cumplirse ambas condiciones.
- **o**: se cumple alguna de las condiciones.
- **no**: no se da la condición.



## ROBOTS

En esta sección se incluyen las instrucciones propias de Arduino o mBot.

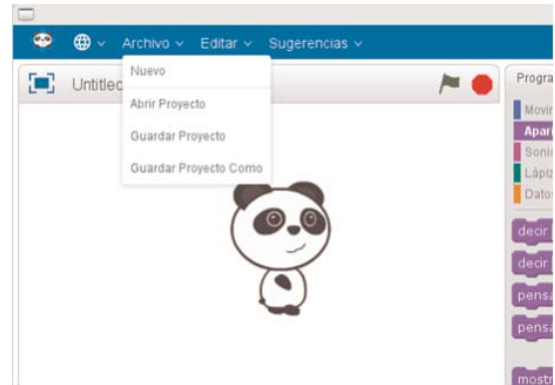
Se ha eliminado el concepto de pines y estados (high / low) y se utiliza un lenguaje más natural para acceder a cada una de las características de la tarjeta controladora.



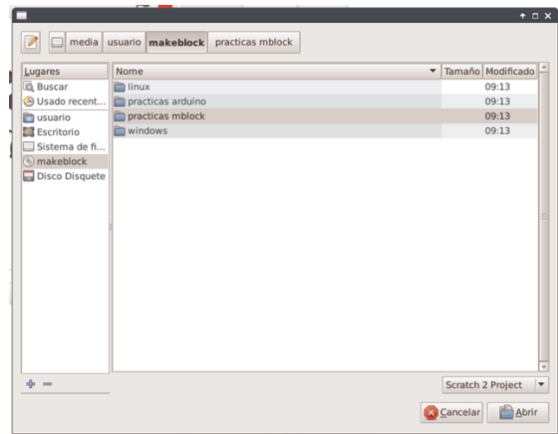
## 1.5 CÓMO CARGAR UNA PRÁCTICA DEL CD

### 1.5.1. ENTORNO LINUX ABALAR

Desde la aplicación mBot, en el menú superior, pulsamos con el botón izquierdo del ratón sobre **"Archivo"** y después en **"Abrir proyecto"**.

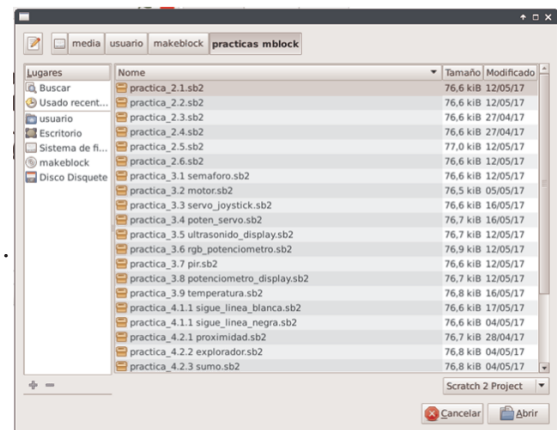


A continuación introducimos el CD en el lector de CD/DVD del ordenador y hacemos click con el botón izquierdo del ratón sobre la unidad de CD makeblock.

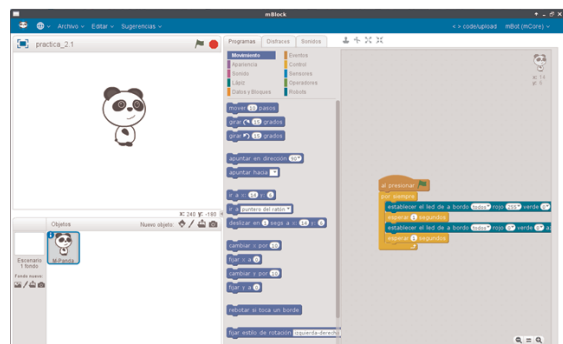


Hacemos doble click sobre la carpeta **"practicas mblock"**.

Y seleccionamos la primera práctica haciendo doble click sobre el archivo **"practica\_2.1.sb2"**.



A continuación podremos cargar la práctica en la placa mbot siguiendo los pasos indicados en el apartado 1.3.

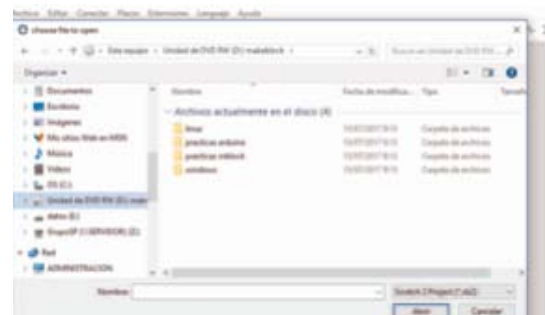


## 1.5.2. ENTORNO WINDOWS

En el menú de la aplicación mBot, hacemos clic con el botón izquierdo del ratón sobre la opción "**Archivo**" y después en "**Abrir Proyecto**".

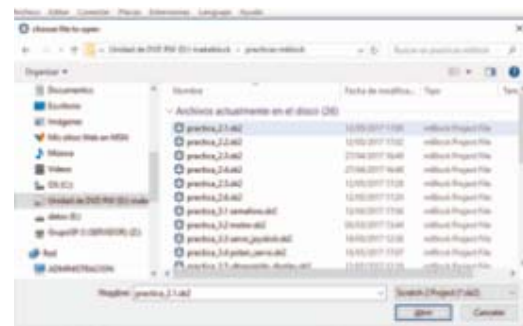


Introducimos el CD en el lector y seleccionamos la unidad de CD/DVD en la siguiente ventana.

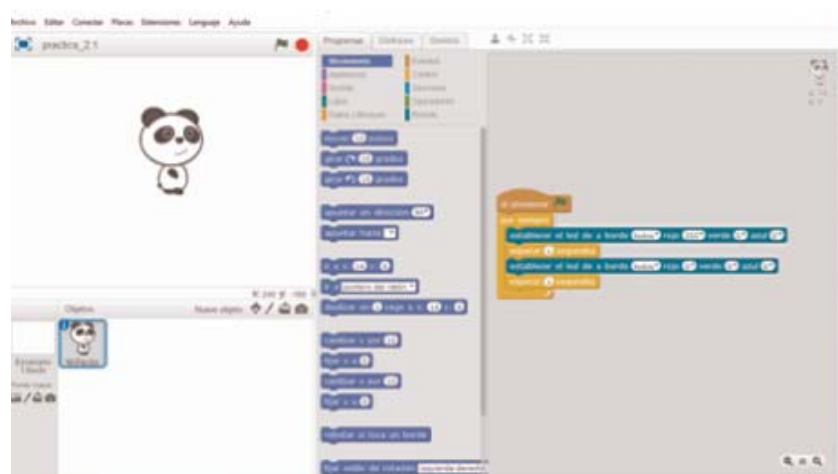


Hacemos doble clic con el botón izquierdo del ratón sobre la carpeta "**prácticas mblock**".

Hacemos doble clic sobre el nombre del archivo "**practica\_2.1.sb2**" y aparecerá el programa dentro de la aplicación mBot.



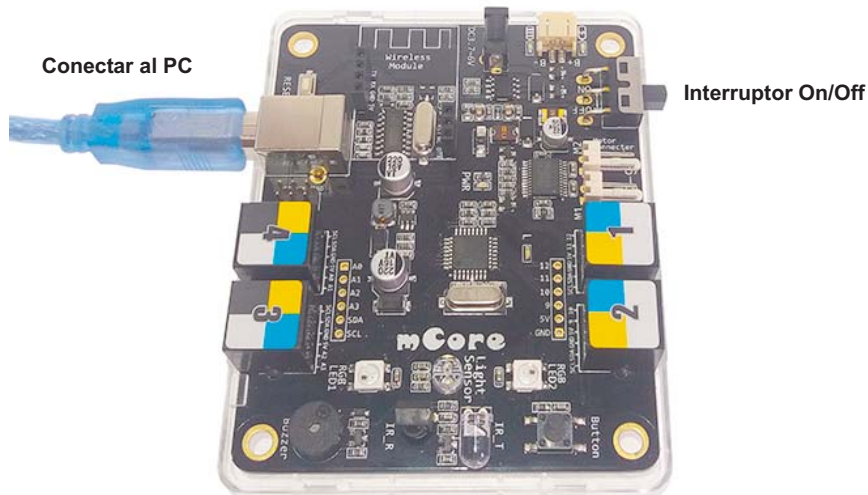
Para cargarlo en la placa, procedemos como se indicó en el apartado 1.3.



## 2. PRÁCTICAS DE INICIACIÓN CON LA PLACA

Para poder realizar todas las prácticas de este apartado hay que hacer estas conexiones sobre la placa (comunes a todas las prácticas):

- Conectar la placa al ordenador mediante el cable USB.
- Colocar el interruptor en la posición ON.



### 2.1 ENCENDIDO Y APAGADO DE LOS LEDs DE LA PLACA

#### Objetivo:

Encender y apagar las bombillas (diodos LED RGB) internos de la placa MCore, encendiéndolas todas en color rojo.

#### Indicaciones:

Los diodos LED pueden visualizar luces de distintos colores a través del bloque “establecer el LED de abordo”. En esta instrucción indicaremos qué LEDs queremos encender y el color de su luz. En función de la combinación de valores que indiquemos tendremos un color final u otro:

Rojo 0 verde 0 azul 255 nos iluminará una luz azul  
 Rojo 0 verde 255 azul 0 nos iluminará una luz verde  
 Rojo 255 verde 0 azul 0 nos iluminará una luz roja

La combinación de valores mayores que 0 de cada uno de estos colores nos dará colores alternativos





## 2.2 SEMÁFORO

### Objetivo:

Simular la iluminación de un semáforo utilizando un LED interno de la placa.

### Indicaciones:

Un semáforo realiza la secuencia de iluminación Verde - Amarillo - Rojo. Para conseguir que un LED RGB luzca de color amarillo tendremos que hacer una combinación de luces (colores) rojo y verde: Rojo 150 Verde 60.



```

al presionar
por siempre
  establecer el led de a bordo todos rojo 0 verde 255 azul 0
  esperar 1 segundos
  establecer el led de a bordo todos rojo 150 verde 60 azul 0
  esperar 1 segundos
  establecer el led de a bordo todos rojo 255 verde 0 azul 0

```

## 2.3 SEMÁFORO CON AVISO PARA INVIDENTES

### Objetivo:

Simular la iluminación de un semáforo incluyendo un aviso sonoro para que las personas invidentes sepan cuando cruzar.

### Indicaciones:

Para emitir un sonido utilizaremos la instrucción "reproducir tono en la nota". Cuando el semáforo esté en rojo, los coches habrán parado y será el momento de realizar un conjunto de pitidos intermitentes utilizando la instrucción "repetir".



```

al presionar
por siempre
  establecer el led de a bordo todos rojo 0 verde 255 azul 0
  esperar 1 segundos
  establecer el led de a bordo todos rojo 150 verde 60 azul 0
  esperar 1 segundos
  establecer el led de a bordo todos rojo 255 verde 0 azul 0
  repetir 6
    reproducir tono en la nota D5 pulsación Cuarto
    reproducir tono en la nota E5 pulsación Octavo
    esperar 0.05 segundos

```



## 2.4 SEMÁFORO CON AVISO PARA INVIDENTES Y BOTÓN PARA PEATONES

### Objetivo:

Simular un semáforo que pasa al estado “Rojo” cuando un peatón pulsa un pulsador

### Indicaciones:

Para saber si el botón de la placa se ha pulsado tenemos la instrucción "botón de la placa". Si el botón de la placa está "liberado" significa que aún no se ha pulsado.

Necesitamos utilizar una instrucción condicional de tipo "si / si no" que determine que cuando se haya pulsado el botón realice el ciclo "amarillo - rojo - sonido".

Si el botón de la placa está "liberado" el semáforo se mantendrá verde, y si no, realizará el ciclo del semáforo completo.



## 2.5 ARCOIRIS

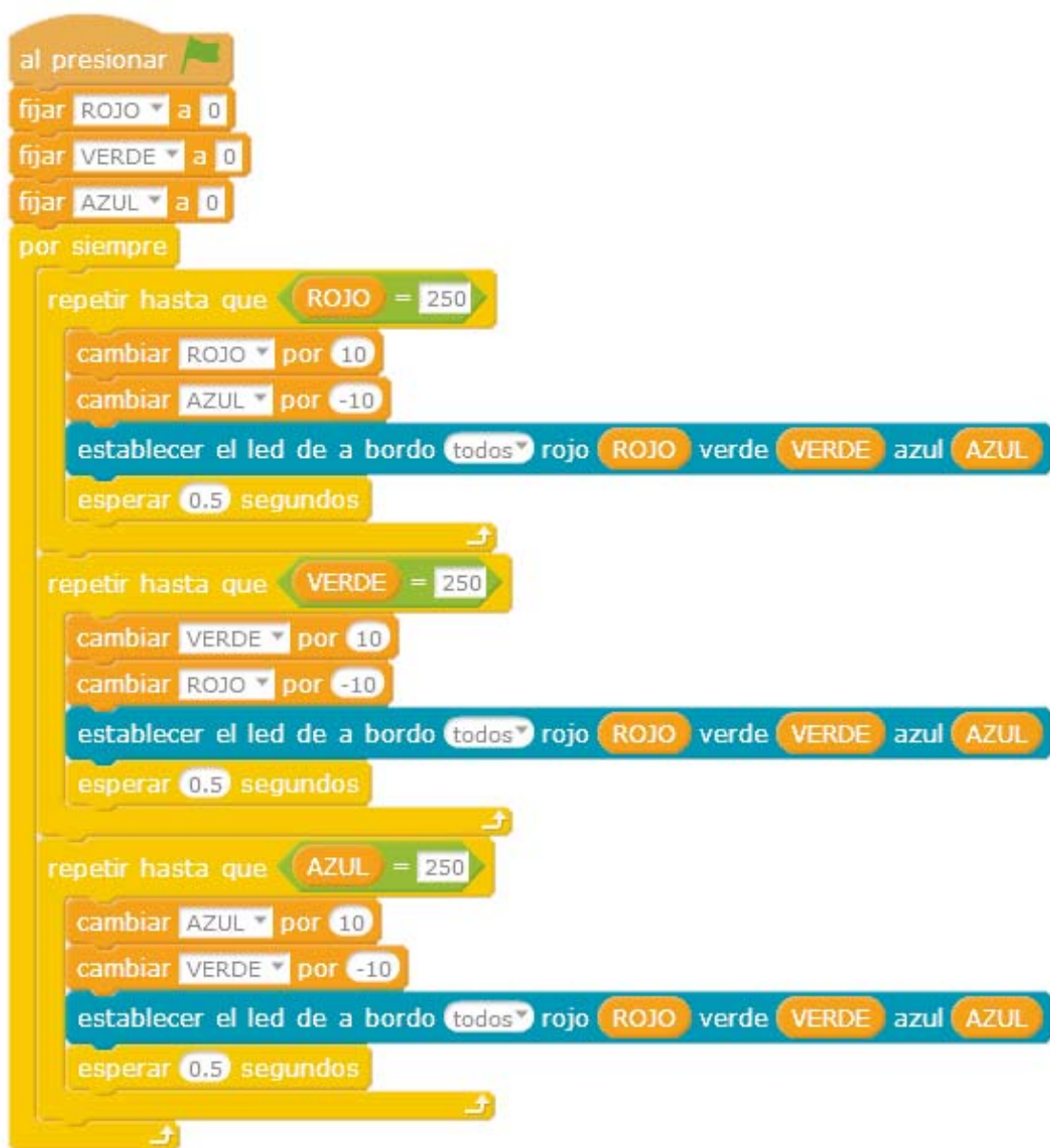
### Objetivo:

Iluminar las luces de la placa de forma que varíen su color como si fuese un arco iris.

### Indicaciones:

Utilizamos una variable para cada color que comenzará con valor 0.

Realizamos un bucle para cada color que va aumentando su luz en 10 unidades mientras que con -10 unidades disminuye el color anteriormente aumentado. Así vamos consiguiendo diferentes gamas de colores.



## 2.6 DESPERTADOR SOLAR

### Objetivo:

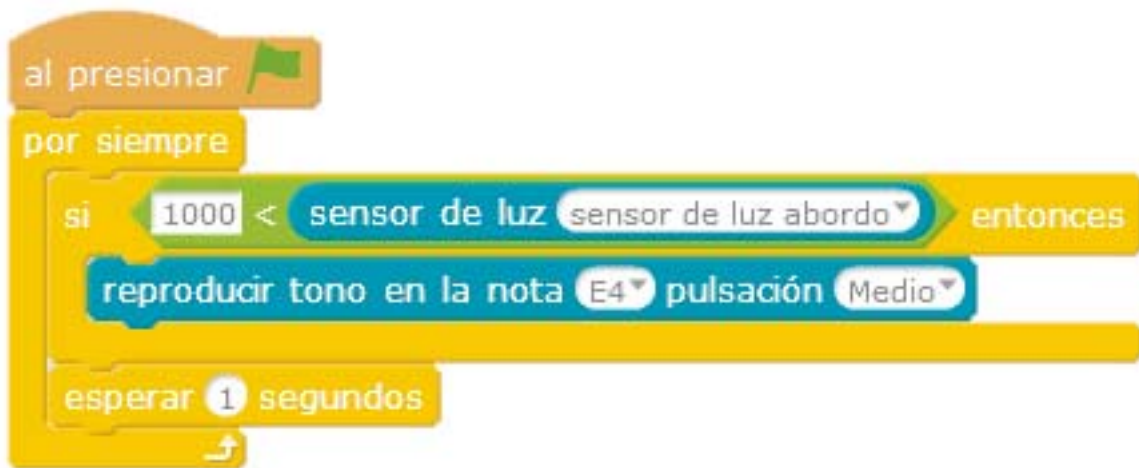
Hacer sonar una nota musical cuando detecte luz ambiente.

### Indicaciones:

El sensor de luz da a la placa valores por encima de 1000 cuando detecta luz ambiente.

Utilizaremos el bloque "sensor de luz" para leer la luz ambiente.

Con un bucle condicional, indicaremos que si "sensor de luz" es mayor que 1000, entonces la placa emitirá una nota musical.



### 3. PRÁCTICAS AÑADIENDO MÓDULOS A LA PLACA

#### 3.1 CRUCE DE SEMÁFOROS

##### Objetivo:

Simular un cruce de semáforos que regula el tráfico de dos calles.

##### Conexiones:

Conectar los módulos RGB a los puertos 1 y 4



##### Indicaciones:

Tenemos que representar dos semáforos que se coordinen:

- Si el primer semáforo está verde, el segundo tendrá que estar rojo.
- Si el primer semáforo está rojo, el segundo semáforo estará verde.
- Si el primer semáforo está amarillo, ¿cómo tendrá que estar el segundo semáforo?.

```

Programa de mBot
por siempre
  establecer el led Puerto1 todos red 0 green 255 blue 0
  establecer el led Puerto4 todos red 255 green 0 blue 0
  esperar 2 segundos
  establecer el led Puerto1 todos red 150 green 60 blue 0
  esperar 1 segundos
  establecer el led Puerto1 todos red 255 green 0 blue 0
  establecer el led Puerto4 todos red 0 green 255 blue 0
  esperar 2 segundos
  establecer el led Puerto4 todos red 150 green 60 blue 0
  esperar 1 segundos
  
```

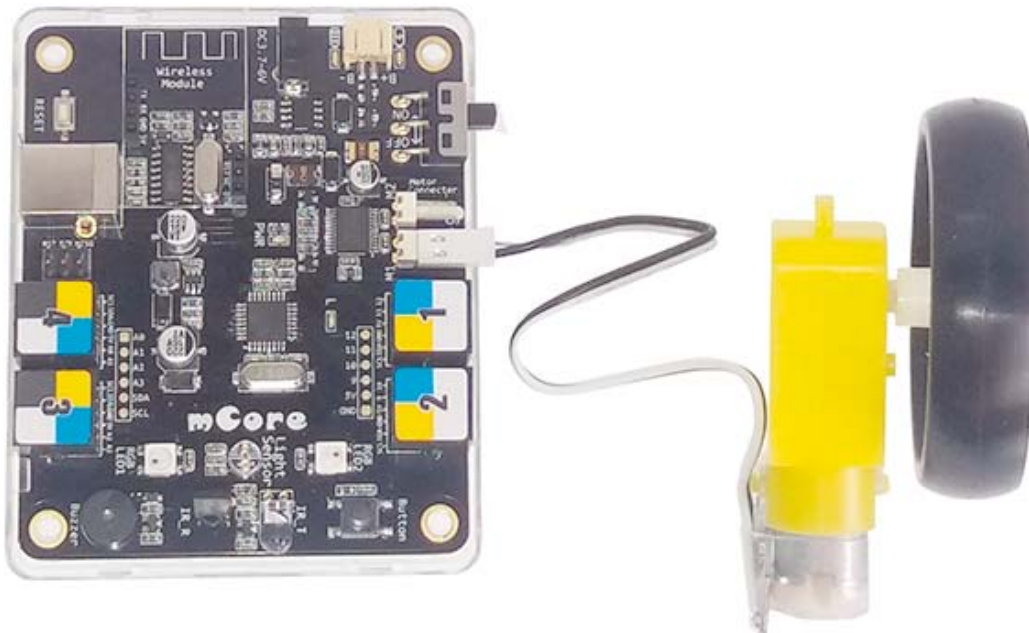
### 3.2 ACTIVAR UN MOTOR

#### Objetivo:

Mover un motor hacia delante y hacia atrás intermitentemente.

#### Conexiones:

Conectar el motor al puerto M1.



#### Indicaciones:

Para mover el motor utilizamos la instrucción “fijar motor” indicando a qué velocidad lo vamos a mover.

El sentido cambia en función de si la velocidad está por encima o por debajo de 0.





### 3.3 MOVER EL SERVOMOTOR CON EL JOYSTICK

#### Objetivo:

Poner el servomotor en posición 0° - 180° en función de la posición del joystick.

#### Conexiones:

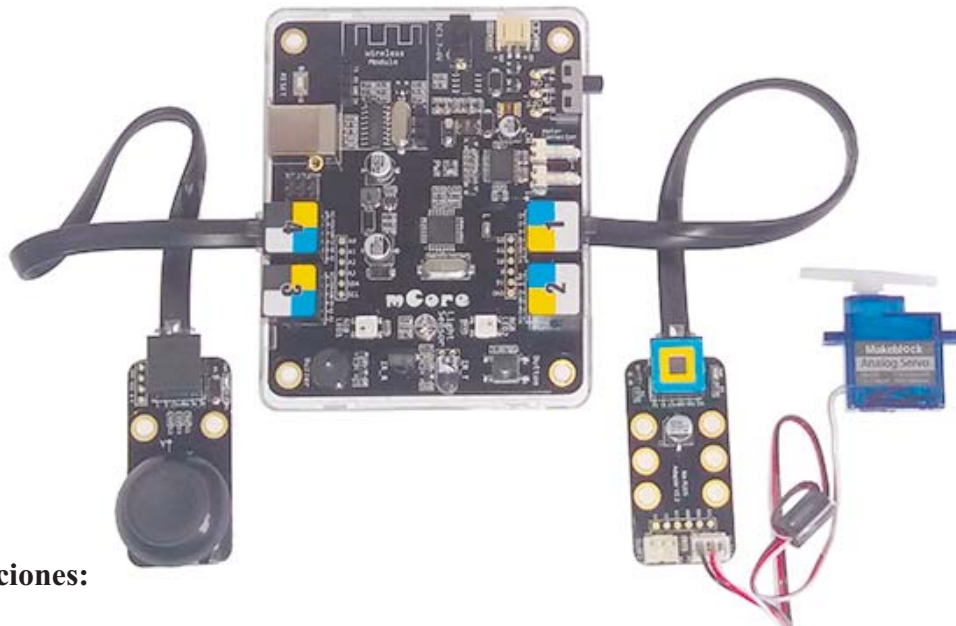
Conectar el Joystick al puerto 4 de la placa MCore.

Necesitamos utilizar el módulo adaptador RJ25.

Conectar el servomotor al conector 1 del adaptador.

Conectar un extremo de un cable RJ25 al módulo adaptador.

Conectar el otro extremo del cable RJ25 al puerto 1 de la placa MCore.



#### Indicaciones:

Utilizamos la instrucción Joystick para leer los movimientos que se hacen sobre el mismo.

Cuando el Eje Y del joystick alcance un valor inferior a -400, el motor se colocará en posición 0°.

Cuando el Eje Y del joystick alcance un valor por encima de 400, el motor se colocará en posición 180°.



### 3.4 CONTROL DEL ÁNGULO DEL SERVOMOTOR CON EL POTENCIÓMETRO

#### Objetivo:

Mover el servomotor utilizando el potenciómetro como mando.

#### Conexiones:

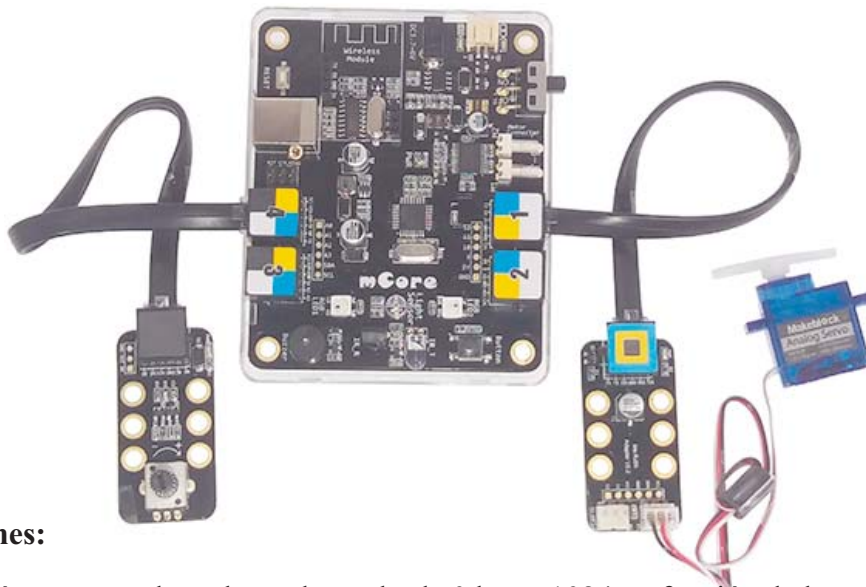
Conectamos el potenciómetro al puerto 4.

Necesitamos utilizar el módulo adaptador RJ25.

Conectar el servomotor al conector 1 del adaptador.

Conectar un extremo de un cable RJ25 al módulo adaptador.

Conectar el otro extremo del cable RJ25 al puerto 1 de la placa MCore.



#### Indicaciones:

El potenciómetro nos devuelve valores desde 0 hasta 1024 en función de la posición de su mando (aunque el rango puede variar ligeramente).

El servomotor puede posicionarse en ángulos desde 0 hasta 180.

Tenemos que hallar la relación entre el rango de ángulos y el rango del potenciómetro para conseguir un relación potenciómetro - ángulo.

La primera prueba que debemos realizar es una lectura del potenciómetro, y utilizando el bloque “decir” mostrar el valor del potenciómetro en el escenario para detectar cual es el valor más alto que proporciona.

Así, la relación entre el ángulo y el potenciómetro será  $180/\text{máx\_valor\_potenciómetro}$

La posición del motor será:  $\text{lectura\_potenciómetro} * (180/\text{máx\_valor\_potenciómetro})$





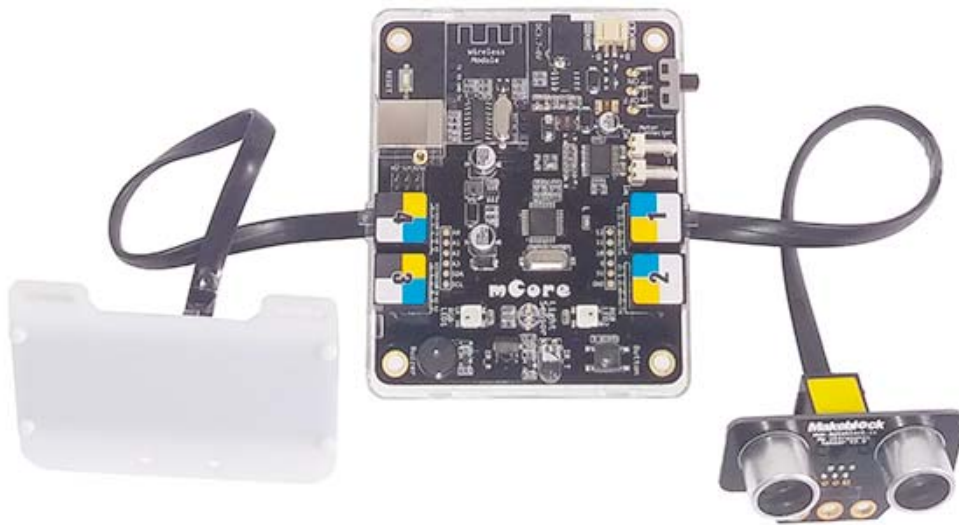
### 3.5 MOSTRAR LA DISTANCIA DEL ULTRASONIDO EN EL DISPLAY

#### Objetivo:

Visualizar la distancia a un obstáculo.

#### Conexiones:

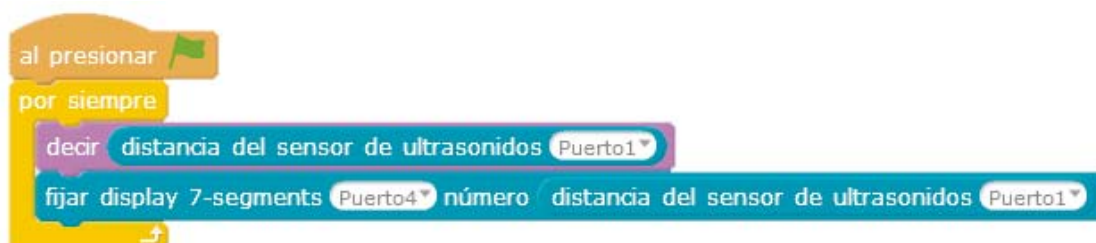
Conectamos la matriz de LEDs en el puerto 4.  
Conectamos el sensor ultrasonido en el puerto 1.



#### Indicaciones:

Para mostrar textos en el display utilizamos el bloque "fijar display".

Para obtener la distancia a un obstáculo utilizamos el bloque "distancia del sensor de ultrasonidos".



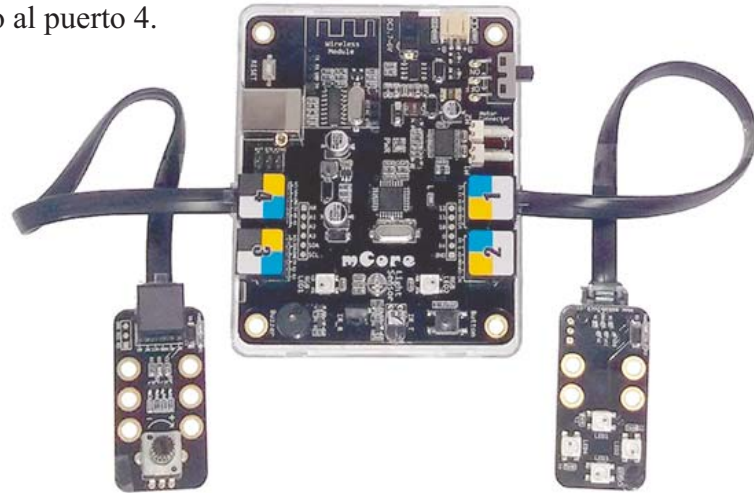
### 3.6 CAMBIO DE COLOR DEL LED RGB CON EL POTENCIÓMETRO

#### Objetivo:

Utilizar el potenciómetro para modificar el color del LED RGB.

#### Conexiones:

Conectamos el potenciómetro al puerto 4.



#### Indicaciones:

El potenciómetro nos devuelve valores desde 0 a 1024.

- Cuando el potenciómetro dé valores de 0 a 255, asignaremos estos valores al color azul del LED y los demás colores los mantenemos a 0.
- Cuando el potenciómetro dé valores de 255 a 510, asignaremos un valor de 0 a 255 al color rojo y los demás colores los mantenemos a 0.
- Cuando el potenciómetro dé valores por encima de 510, asignaremos un valor de 0 a 255 al color verde y los demás colores los mantenemos a 0.

```

al presionar bandera verde
  fijar azul a 0
  fijar verde a 0
  fijar rojo a 0
  por siempre
    si potenciómetro Puerto4 < 255 entonces
      fijar azul a potenciómetro Puerto4
      fijar verde a 0
      fijar rojo a 0
    si potenciómetro Puerto4 < 510 y potenciómetro Puerto4 > 255 entonces
      fijar azul a 0
      fijar verde a 0
      fijar rojo a potenciómetro Puerto4 - 255
    si potenciómetro Puerto4 > 510 entonces
      fijar azul a 0
      fijar verde a potenciómetro Puerto4 - 510
      fijar rojo a 0
  decir potenciómetro Puerto4
  establecer el led de a bordo todos rojo rojo verde verde azul azul
  
```

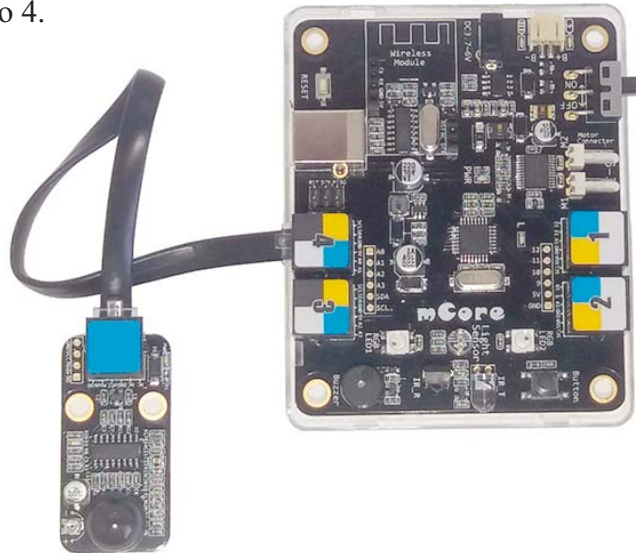
### 3.7 DETECTOR DE PRESENCIA CON EL SENSOR PIR

#### Objetivo:

Encender un LED rojo cuando se detecte una presencia en la habitación.

#### Conexiones:

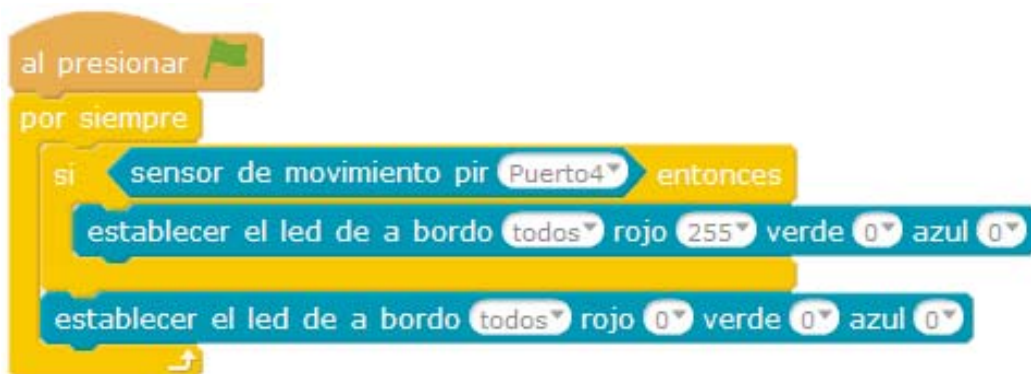
Conectamos el sensor PIR al puerto 4.



#### Indicaciones:

Para controlar la presencia de alguien en la habitación, disponemos del bloque "sensor de movimiento pir" que debemos utilizar dentro de una instrucción condicional "si... entonces...".

Así, si se detecta una presencia, utilizaremos "establecer LED" para encender o apagar el LED correspondiente.



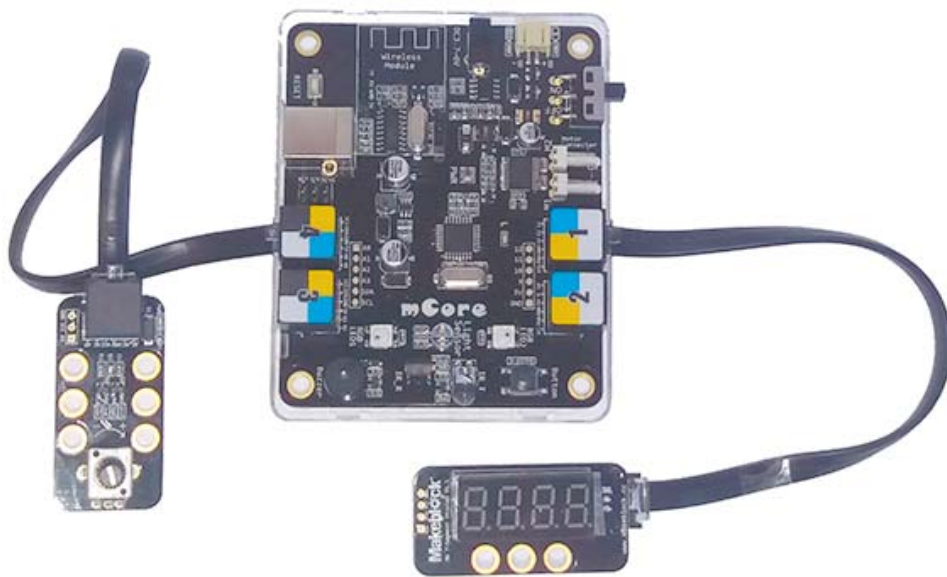
### 3.8 INDICADOR CON POTENCIÓMETRO

#### Objetivo:

Visualizar en el display de 4 dígitos el valor que aporta el potenciómetro.

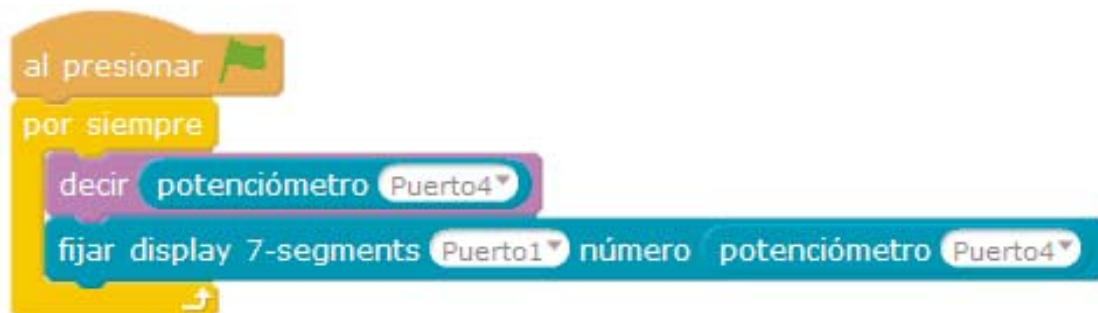
#### Conexiones:

Conectamos el potenciómetro al puerto 4.  
Conectamos el display al puerto 1.



#### Indicaciones:

Cuando movemos el mando del potenciómetro, éste genera valores de 0 a 1024 que vamos a mostrar en el display utilizando los bloques "fijar display 7 segmentos" y "potenciómetro".



### 3.9 INDICADOR DE TEMPERATURA

#### Objetivo:

Visualizar en el display de 4 dígitos la temperatura detectada por la sonda.

#### Conexiones:

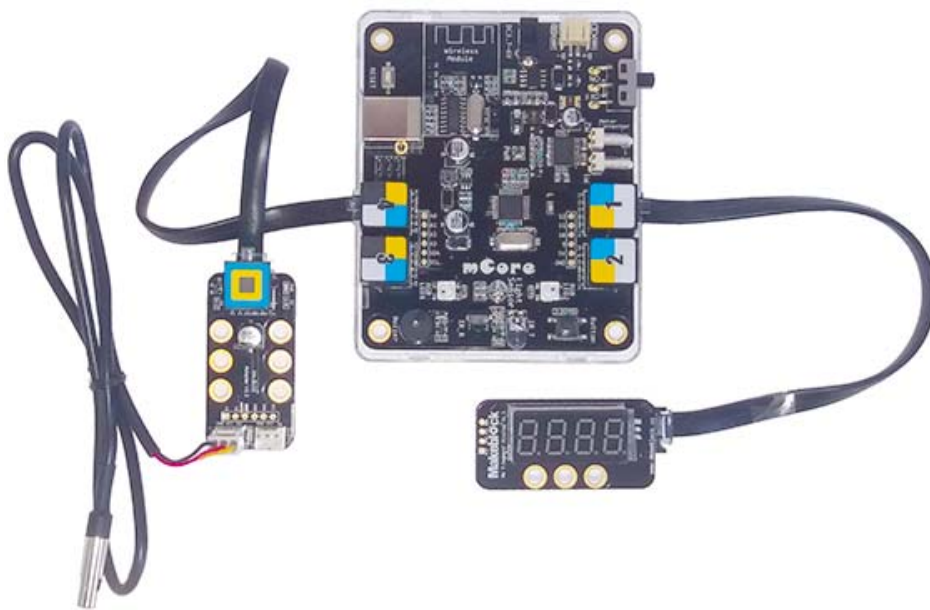
Conectamos el display al puerto 1.

Necesitamos utilizar el módulo adaptador RJ25.

Conectar la sonda al conector 2 del adaptador.

Conectar un extremo de un cable RJ25 al módulo adaptador.

Conectar el otro extremo del cable RJ25 al puerto 4 de la placa MCore.



#### Indicaciones:

Para leer la temperatura utilizamos el bloque “temperatura” que mostraremos en el display utilizando el bloque “fijar display 7 segmentos”.



## **4. PROYECTOS ROBOT**

### 4.1 Robot infrarrojo

- 4.1.1 Seguidor de línea negra
- 4.1.2 Seguidor de línea blanca

### 4.2 Robot Ultrasonido

- 4.2.1 Sensor de proximidad
- 4.2.2 Explorador
- 4.2.3 Sumo

### 4.3 Robot expresivo

- 4.3.1 Matriz de LEDs
- 4.3.2 Reloj Digital
- 4.3.3 Medidor de distancias

### 4.4 Robot servomotor

- 4.4.1 Control de un servomotor
- 4.4.2 Portero

### 4.5 Robot control a distancia

- 4.5.1 Control con mando a distancia
- 4.5.2 Control por bluetooth



## 4.1 ROBOT INFRARROJO

### 4.1.1 ROBOT SIGUE LÍNEA NEGRA

#### Objetivo:

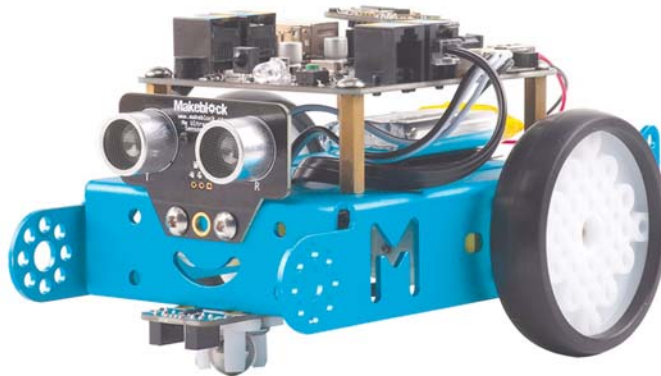
Construir y programar un robot capaz de circular de forma autónoma por encima de una línea negra.

#### Montaje:

Construir el robot siguiendo el manual de montaje.

#### Sensores y conexiones:

- El sensor de línea se conecta al puerto 2.
- Éste sensor de línea dispone de dos lectores de línea con los que controlaremos sus bordes.

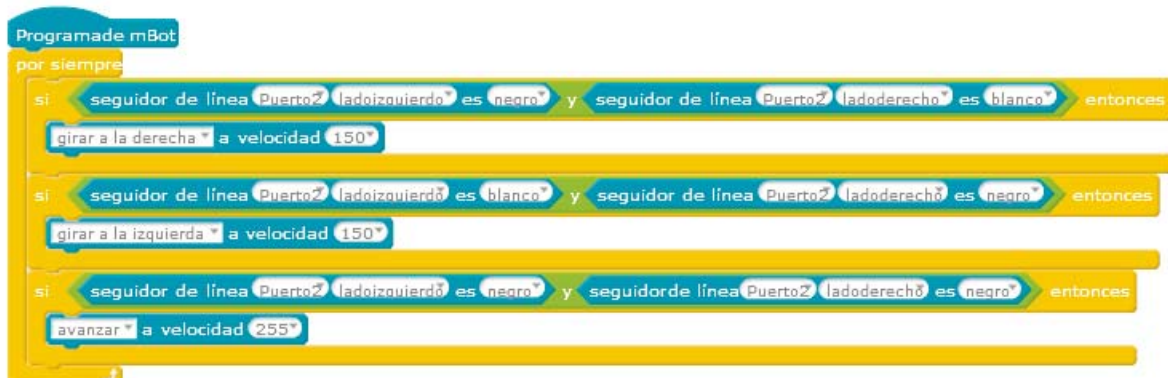


#### Programación:

Si el lector izquierdo del sensor de línea está encima de la línea negra y el sensor derecho está fuera de la línea, corrige la dirección para mantenerse en la línea, girando el vehículo hacia la derecha.

Si el lector derecho del sensor de línea está encima de la línea negra y el sensor izquierdo está fuera de la línea, corrige la dirección para mantenerse en la línea, girando el vehículo hacia la izquierda.

Si ambos sensores están sobre la línea negra, el vehículo avanza recto.





## 4.1.2 ROBOT SIGUE LÍNEA BLANCA

### Objetivo:

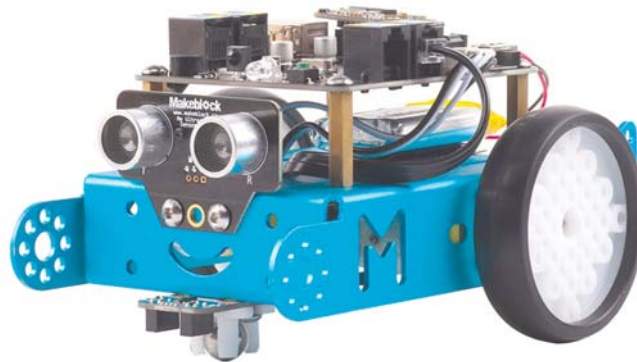
Construir y programar un robot capaz de circular de forma autónoma por encima de una línea blanca.

### Montaje:

Construir el robot siguiendo el manual de montaje.

### Sensores y conexiones:

- El sensor de línea se conecta al puerto 2.
- Éste sensor de línea dispone de dos lectores de línea con los que controlaremos sus bordes.



### Programación:

Si el lector izquierdo del sensor de línea está encima de la línea blanca y el sensor derecho está fuera de la línea, corrige la dirección para mantenerse en la línea, girando el vehículo hacia la derecha.

Si el lector derecho del sensor de línea está encima de la línea blanca y el sensor izquierdo está fuera de la línea, corrige la dirección para mantenerse en la línea, girando el vehículo hacia la izquierda.

Si ambos sensores están sobre la línea blanca, el vehículo avanza recto.

```

Programa de mBot
por siempre
  si seguidor de línea Puerto2 ladoizquierdo es blanco y seguidor de línea Puerto2 ladoderecho es negro entonces
    girar a la derecha a velocidad 150
  si seguidor de línea Puerto2 ladoizquierdo es negro y seguidor de línea Puerto2 ladoderecho es blanco entonces
    girar a la izquierda a velocidad 150
  si seguidor de línea Puerto2 ladoizquierdo es blanco y seguidor de línea Puerto2 ladoderecho es blanco entonces
    avanzar a velocidad 255
  
```

## 4.2 ROBOT ULTRASONIDO

### 4.2.1 SENSOR DE PROXIMIDAD

#### Objetivo:

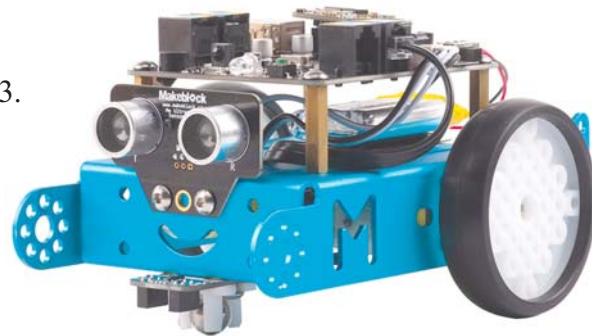
Construir y programar un robot que emita una alarma si se aproxima un obstáculo.

#### Montaje:

Construir el robot siguiendo el manual de montaje.

#### Sensores y conexiones:

- El sensor ultrasonido se conecta al puerto 3.



#### Programación:

El programa compara las lecturas:

- Si la lectura es menor que 20 y mayor de 15.1, reproduce la nota C4 con una pulsación de Octavo.
- Si la lectura es menor que 15 y mayor de 10.1, reproduce la nota C5 con una pulsación de Octavo.
- Si la lectura es menor que 10 y mayor de 3, reproduce la nota C6 con una pulsación de Octavo.
- Si la lectura es menor que 3, reproduce la nota C8 con una pulsación de doble.

```

Programade mBot
por siempre
si distancia del sensor de ultrasonidos Puerto3 < 20 y distancia del sensor de ultrasonidos Puerto3 > 15.1 entonces
  reproducir tono en la nota C4 pulsación Octavo
  esperar 0.1 segundos
si distancia del sensor de ultrasonidos Puerto3 < 15 y distancia del sensor de ultrasonidos Puerto3 > 10.1 entonces
  reproducir tono en la nota C5 pulsación Octavo
  esperar 0.05 segundos
si distancia del sensor de ultrasonidos Puerto3 < 10 y distancia del sensor de ultrasonidos Puerto3 > 3 entonces
  reproducir tono en la nota C6 pulsación Octavo
  esperar 0.01 segundos
si distancia del sensor de ultrasonidos Puerto3 < 3 entonces
  reproducir tono en la nota C8 pulsación Doble
  
```

## 4.2.2 EXPLORADOR

### Objetivo:

Construir y programar un robot capaz de circular por una sala sin chocarse con ningún obstáculo.

### Montaje:

Construir el robot siguiendo el manual de montaje.

### Sensores y conexiones:

- El sensor ultrasonido se conecta al puerto 3.

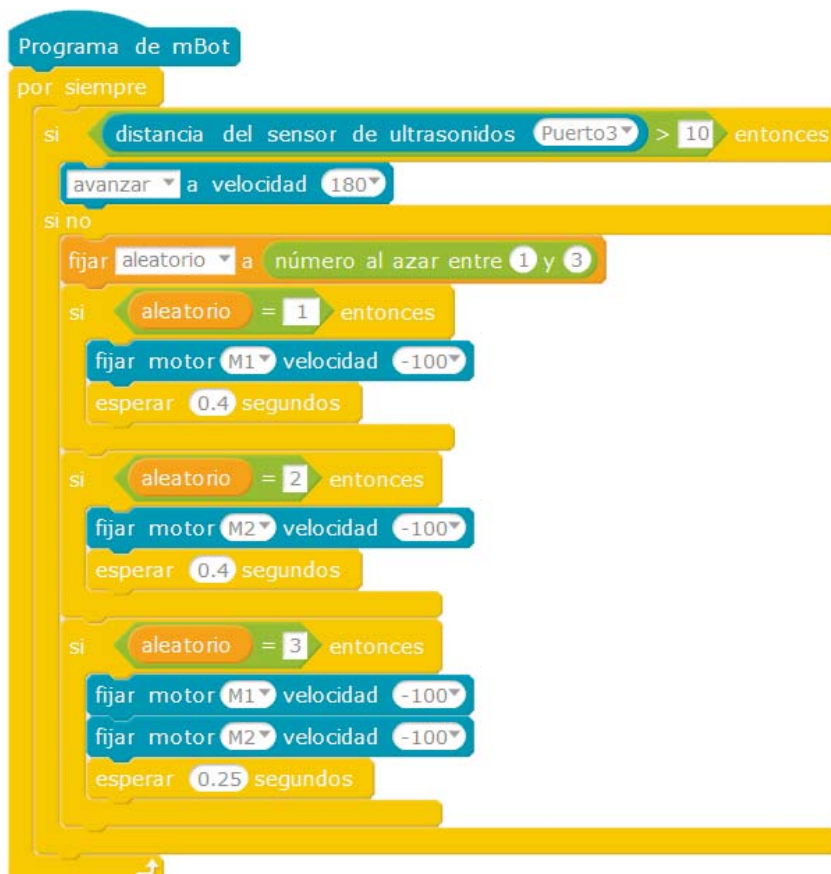
### Programación:

Si el robot no detecta ningún obstáculo a una distancia por debajo de 10, avanza en línea recta.

Si detecta un obstáculo, debe realizar un cambio de rumbo. Ese cambio de rumbo se realiza de manera aleatoria.

Rumbo 1: Gira a la derecha  
 Rumbo 2: Gira a la izquierda  
 Rumbo 3: Retrocede

Definimos una variable que adquiere un valor aleatorio de 1 a 3 y en función de su valor, se realizan las instrucciones asociadas a ese rumbo.



```

Programa de mBot
por siempre
  si distancia del sensor de ultrasonidos Puerto3 > 10 entonces
    avanzar a velocidad 180
  si no
    fijar aleatorio a número al azar entre 1 y 3
    si aleatorio = 1 entonces
      fijar motor M1 velocidad -100
      esperar 0.4 segundos
    si aleatorio = 2 entonces
      fijar motor M2 velocidad -100
      esperar 0.4 segundos
    si aleatorio = 3 entonces
      fijar motor M1 velocidad -100
      fijar motor M2 velocidad -100
      esperar 0.25 segundos
  
```

### 4.2.3 SUMO

#### Objetivo:

Construir y programar un robot capaz de localizar un contrincante y empujarlo fuera de un Ring.

#### Montaje:

Construir el robot siguiendo el manual de montaje.

#### Sensores y conexiones:

- El sensor ultrasonido se conecta al puerto 3.

#### Programación:

Si el sensor ultrasonido no detecta ningún contrincante en una distancia menor de 30, girará aleatoriamente hacia un lado en busca de su contrincante.

Si detecta un contrincante avanzará en línea recta hacia él para empujarlo fuera del ring.

```

Programa de mBot
por siempre
  si distancia del sensor de ultrasonidos Puerto3 > 30 entonces
    fijar aleatorio a número al azar entre 1 y 2
    si aleatorio = 1 entonces
      repetir hasta que distancia del sensor de ultrasonidos Puerto3 < 30
        fijar motor M1 velocidad -100
        fijar motor M2 velocidad 100
    si aleatorio = 2 entonces
      repetir hasta que distancia del sensor de ultrasonidos Puerto3 < 30
        fijar motor M1 velocidad 100
        fijar motor M2 velocidad -100
  si no
    avanzar a velocidad 255
  
```

### **4.3 ROBOT EXPRESIVO**

Montar el robot siguiendo la guía de montaje en pdf de la matriz de LEDs.



### 4.3.1 MATRIZ DE LEDS

Es una matriz de 16 x 8 LEDs.

X: 16 LEDs horizontales, que van de izquierda a derecha.

Y: 8 LEDs verticales, que van de arriba hacia abajo.

Para controlar la matriz contamos con 4 instrucciones "mostrar cara":

1) Numérica: podremos mostrar hasta 4 dígitos.



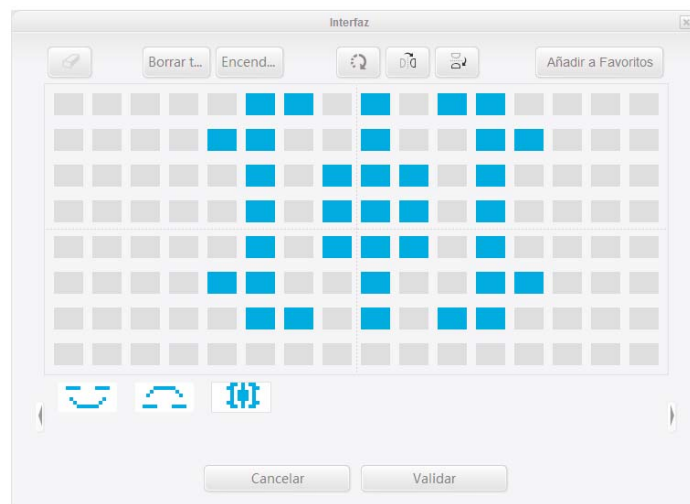
2) Carácter: podemos mostrar 2 caracteres y centrarlos con los vectores X e Y.



3) Tiempo: podemos mostrar un reloj digital.



4) Dibujo: nos muestra una plantilla en la cual podremos ir encendiendo o apagando LEDs hasta formar el dibujo deseado y desplazarlo con los vectores X e Y.





### 4.3.2 RELOJ DIGITAL

#### Objetivo:

Construir un reloj que nos muestre las horas de forma digital.

#### Montaje:

Construir el robot siguiendo el manual de montaje.

#### Sensores y conexiones:

- La matriz de LEDs se conecta al puerto 1.

#### Programación:

Creamos 3 variables para visualizar las horas, minutos y segundos.  
 Incrementamos los segundos en un unidad y cuando llegan a 60, vuelven a 0.  
 Cuando los segundos han llegado a 60, incrementamos en 1 los minutos.  
 Cuando los minutos han llegado a 60, incrementamos las horas y los minutos vuelven a 0.  
 Cuando las horas han llegado a 24 vuelven a 0.



```

Programa de mBot
fijar horas a 0
fijar minutos a 0
fijar segundos a 0
por siempre
  si segundos < 61 entonces
    cambiar segundos por 1
    esperar 1 segundos
  si segundos = 60 entonces
    cambiar minutos por 1
    fijar segundos a 0
  si minutos = 60 entonces
    cambiar horas por 1
    fijar minutos a 0
  si horas = 24 entonces
    fijar horas a 0
    fijar minutos a 0
    fijar segundos a 0
mostrar tiempo Puerto1 hora: horas min: minutos
  
```



### 4.3.3 MEDIDOR DE DISTANCIAS

#### Objetivo:

Construir y programar un robot que muestre la distancia a un obstáculo

#### Montaje:

Construir el robot siguiendo el manual de montaje.  
Añadir el sensor de ultrasonido al montaje.

#### Sensores y conexiones:

- La matriz de LEDs se conecta al puerto 1.
- El sensor ultrasonido se conecta al puerto 3



#### Programación:

Utilizando el bloque “mostrar cara”, visualizamos la distancia al próximo objeto.



## 4.4 ROBOT SERVOMOTOR

### 4.4.1 CONTROL DE UN SERVOMOTOR

#### Objetivo:

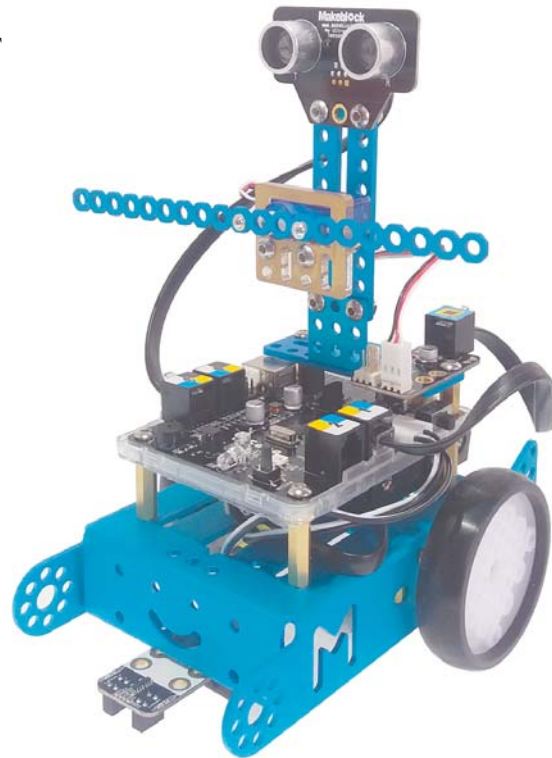
Construir y programar un robot capaz de mover intermitentemente su brazo.

#### Montaje:

Construir el robot siguiendo el manual PDF del CD "Gato bailarín".

#### Sensores y conexiones:

- El servomotor se conecta al puerto 1.



#### Programación:

Utilizando el bloque "fijar servo", movemos intermitentemente el motor desde la posición 0 a 180° y viceversa.



## 4.4.2 PORTERO

### Objetivo:

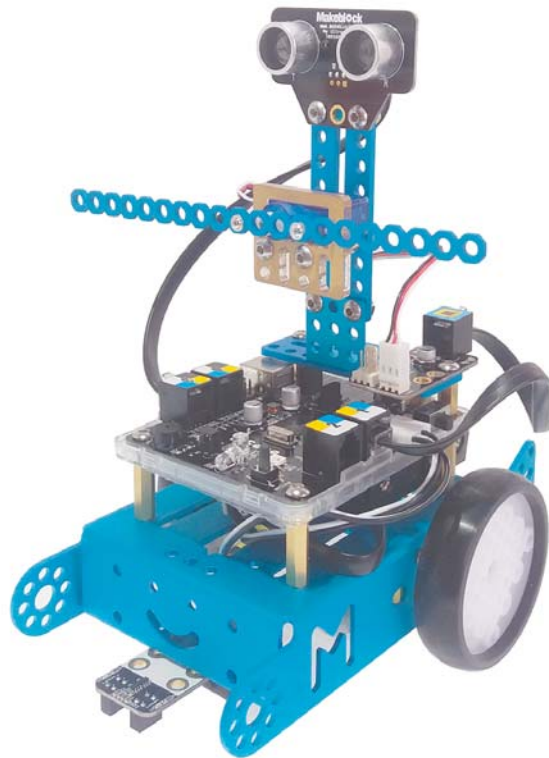
Construir y programar un robot capaz de parar obstáculos con su brazo.

### Montaje:

Construir el robot siguiendo el manual PDF del CD "Gato bailarín".

### Sensores y conexiones:

- El servomotor se conecta al puerto 1.
- El sensor ultrasonido se conecta al puerto 3.



### Programación:

Utilizando el bloque "fijar servo", movemos el motor.

Si el sensor ultrasonido detecta un objeto a una distancia menor de 7, mueve el brazo intentado esquivar el objeto.

```

Programa de mBot
por siempre
  si distancia del sensor de ultrasonidos Puerto3 < 7 entonces
    fijar servo Puerto1 Banco1 ángulo 0
    esperar 0.5 segundos
    fijar servo Puerto1 Banco1 ángulo 180
    esperar 0.5 segundos
  
```

## 4.5 ROBOT CONTROL A DISTANCIA

### 4.5.1 CONTROL CON MANDO A DISTANCIA

#### Objetivo:

Construir y programar un robot cuyos movimientos controlamos con el mando a distancia

#### Montaje:


Construir el robot siguiendo el manual de montaje.


#### Sensores y conexiones:

Precisa del uso del mando a distancia que se comunica con la placa a través del receptor incluido en la misma.


#### Programación:

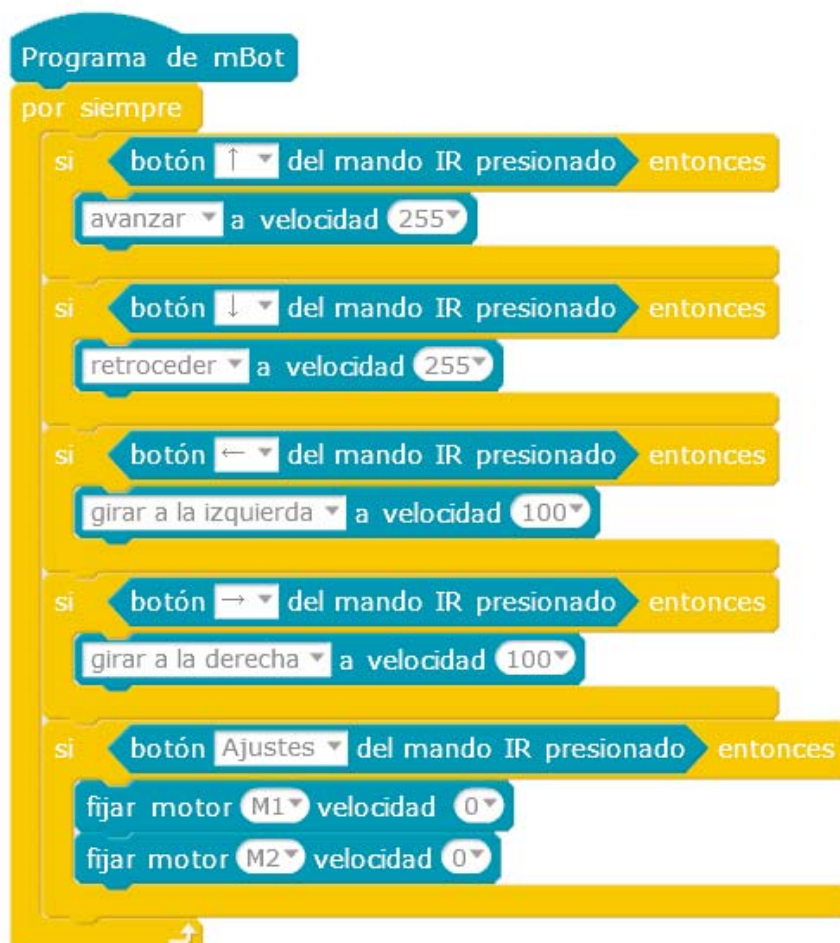
Utilizando el bloque “botón del mando IR presionado” detectamos qué botones hemos presionado en el mando a distancia.

Si se presiona el botón  el robot gira a la izquierda.

Si se presiona el botón  el robot circula hacia atrás.

Si se presiona el botón  el robot gira hacia la derecha.

Si se presiona el botón  el robot circula de frente.



## 4.5.2 CONTROL POR BLUETOOTH

### Objetivo:

Construir y programar un robot cuyos movimientos controlamos con un dispositivo móvil con conexión bluetooth

### Montaje:

Construir el robot siguiendo el manual de montaje.

Para controlar el robot por bluetooth debemos descargar la aplicación mbot de makeblock desde play store e instalarla en nuestro móvil o tablet.

Desde la aplicación de PC mBot, cargamos en nuestro robot el programa predeterminado:

- Pulsar menú
- Pulsar conectar
- Pulsar restaurar programa predeterminado
- Pulsar mbot

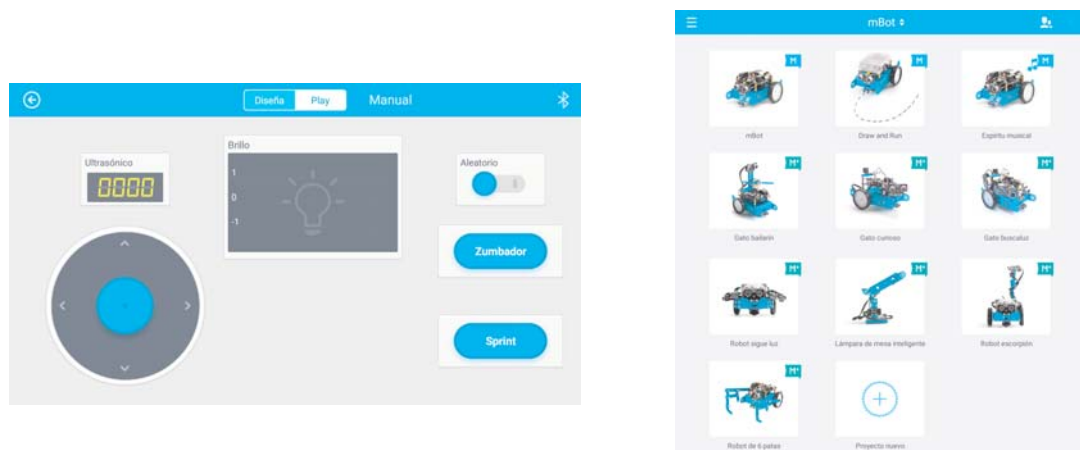
Con la aplicación de makeblock tenemos pre-cargados 10 robots distintos

Todos estos modos de la app, incluyen la posibilidad de controlar al robot como si de un coche teledirigido se tratase. Pero además cada una de ellas dispone de opciones extra.

Veamos algunos ejemplos:

- mbot: Además del modo teledirigido, podemos hacer sonar un zumbido y ordenar al robot que realice un sprint
- Draw and run: Incluye una zona de dibujo donde podemos dibujar con el dedo una ruta que posteriormente reproducirá el robot. Cuenta con la posibilidad de activar el modo radar para avisar si hay algún obstáculo dentro de esa ruta.
- Musical: Ordena al robot la reproducción de dos melodías preestablecidas además de cualquier otra que reproduzca el alumno a través del piano que incluye la app.
- Gato bailarín: podemos controlar el movimiento del brazo gracias a la opción servo.
- Gato curioso: explora el entorno y utiliza el sensor ultrasonido para detectar obstáculos.

Robot sigue luz: Activando la función sigue luz, se dirige hacia la luz de una linterna.



## 5. APROXIMACIÓN A ARDUINO IDE

### MAKEBLOCK Y ARDUINO:

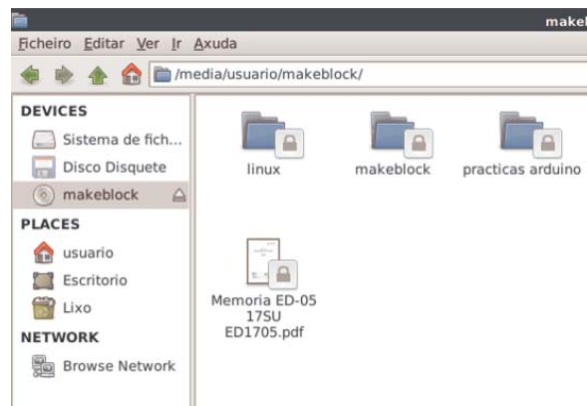
Hasta ahora la programación de los proyectos la hemos realizado bajo un lenguaje de programación gráfico. Makeblock es una placa muy versátil que nos permite avanzar dando un paso más en la programación y permitiéndonos realizar programas con un lenguaje de texto como es processing.

La placa de Makeblock está basada en las placas Arduino, y por lo tanto puedes programarse con el software Arduino IDE.

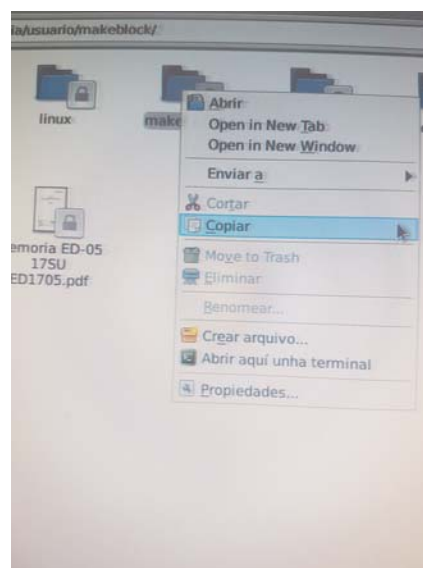
### 5.1 INSTALACIÓN DE ARDUINO IDE

El sistema operativo Linux Abalar incluye por defecto la aplicación Arduino IDE pero precisa que insertemos en el software las librerías propias de MakeBlock.

Para ello accedemos al "Xestor de ficheiros" y pulsamos sobre el CD makeblok, bajo DEVICES, pulsando sobre él con el botón izquierdo del ratón.

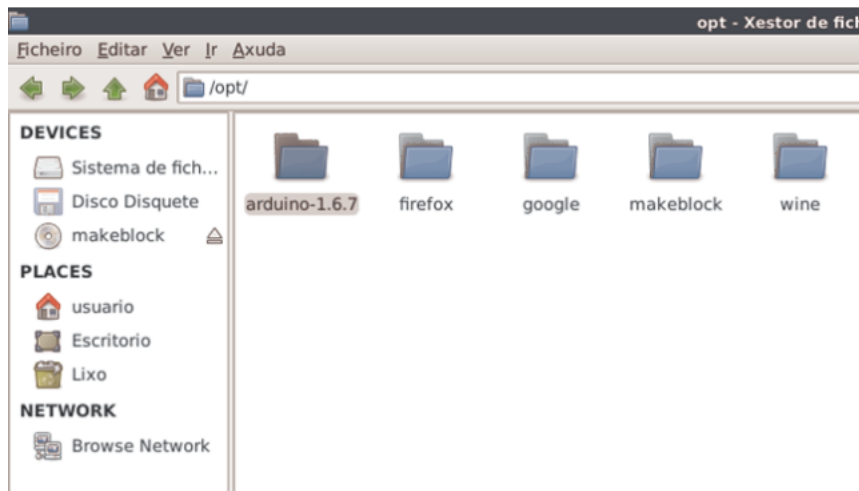


Comprobamos que disponemos de una carpeta llamada makeblock. Pulsamos sobre ella con el botón derecho del ratón y hacemos click sobre copiar con el botón izquierdo del ratón.





A continuación pulsamos sobre Sistema de ficheros, debajo de DEVICES, y localizamos la carpeta opt. Hacemos doble click con el ratón sobre la carpeta opt, y vemos que aparece la carpeta arduino.



Hacemos doble click sobre la carpeta arduino-1.6.7 y localizamos la carpeta libraries.

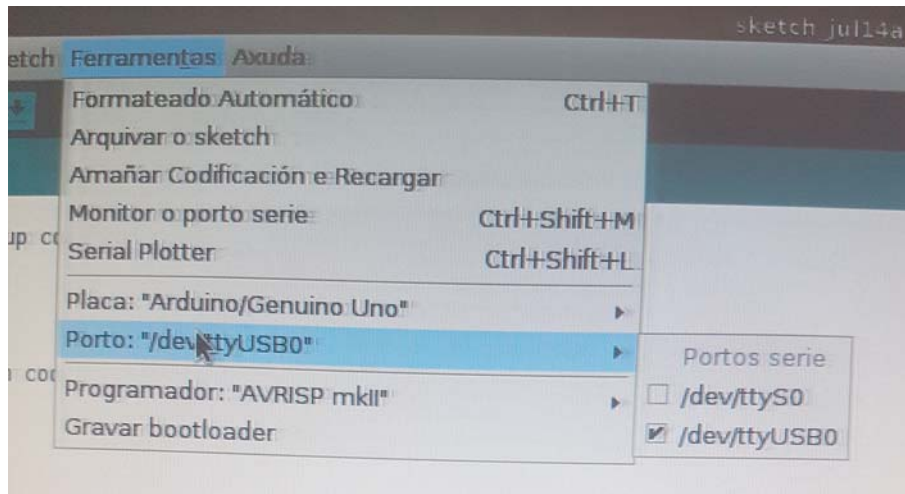


Hacemos doble click sobre la carpeta libraries. Una vez dentro pulsamos en el menú editar y luego pulsamos sobre pegar.

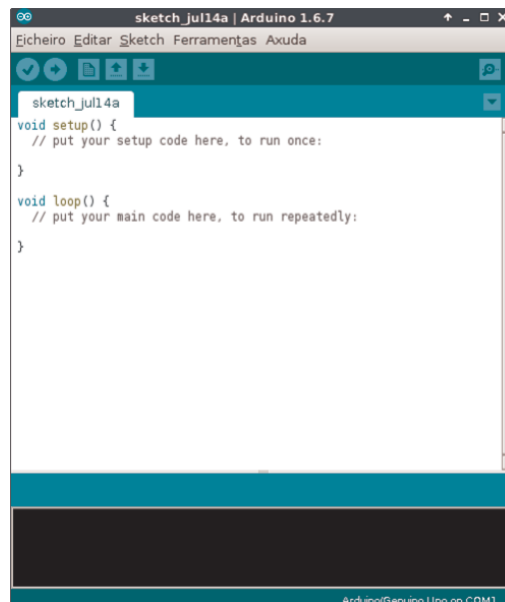


A continuación las librerías precisas para makeblock se quedan guardadas automáticamente en el software Arduino IDE.

Para acceder al software de Arduino IDE, pulsa con el botón derecho del ratón sobre "Menú de aplicativos", después pulsa sobre "programación" y por último sobre "Arduino IDE".



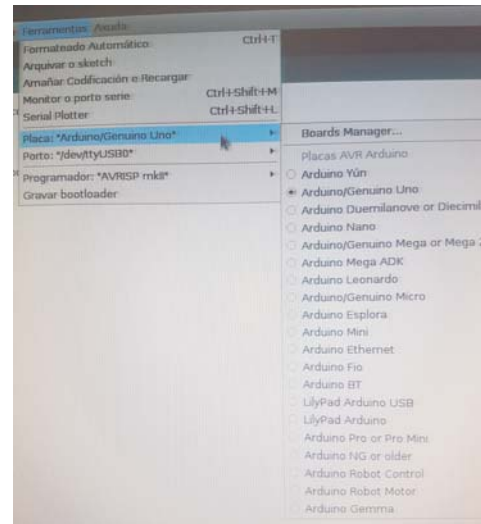
Se abrirá la siguiente ventana en la pantalla:



Para comprobar que la librería de makeblock se ha instalado correctamente, pulsa con el botón izquierdo del ratón sobre sketch y después sobre "incluir biblioteca". Veremos que al final del listado del menú que se ha abierto aparece "Make Block Drive". Son las librerías necesarias para trabajar con Makeblock y Arduino IDE.

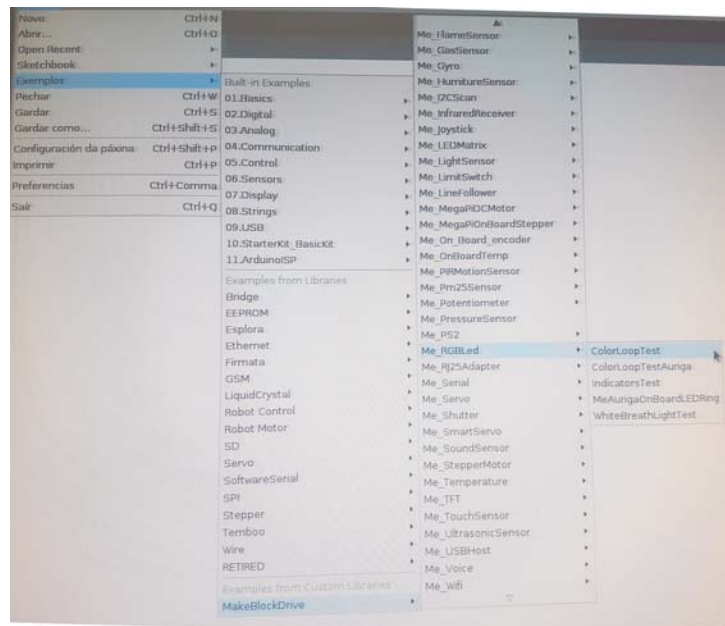
Si pulsas con el botón izquierdo del ratón sobre el menú Ficheiro, y después sobre exemplos, verás que al final del listado de ejemplo aparece la entrada "MakeBlockDrive". Si pulsamos sobre ella se desplegarán numerosos ejemplos para las tarjetas de MakeBlock. Puedes pulsar en ellos y se cargarán en el entorno Arduino IDE para que puedas analizar su código y probarlos.

Vamos a cargar un programa de ejemplo. Conectamos la placa MakeBlock al puerto USB del ordenador mediante el cable. A continuación pulsamos en "Ferramentas" y después en "Porto". Tenemos que seleccionar el puerto en el que hemos instalado la placa Arduino. Hacemos click con el botón izquierdo del ratón sobre /dev/ttyUSB0 (dependiendo del pc el número del puerto puede variar y no ser 0, elegimos el que nos aparezca en el listado).



A continuación comprobamos que la placa esta correctamente seleccionada en el software Arduino IDE. Pulsamos sobre el menú Ferramentas, y después en placa. Del listado de placas que aparece, Arduino/Genuino UNO debe tener un punto negro a su lado indicando que es la placa que vamos a utilizar. Si este punto negro no estuviera ahí, hacemos click con el botón izquierdo del ratón sobre Arduino / Genuino UNO.

A continuación vamos a elegir un programa de ejemplo para probar. Hacemos click sobre Ficheiros, y a continuación en Exemplos. Del listado que aparece, abajo del todo hacemos click en MakeBlockDrive. Nos aparecerá un largo listado de ejemplos. Hacemos click en Me-RGBled y después en ColorLoopTest.



Una vez abierto el archivo pulsamos en Sketch y luego en Subir

El software realiza el proceso de carga en la placa y en unos segundos veremos que el LED RGB realiza ciclos de cambios de color en respuesta al programa subido.

## 5.2 CÓMO PROGRAMAR MI MAKEBLOCK CON ARDUINO IDE

Para facilitar la tarea de programación, Makeblock ha diseñado una serie de instrucciones que nos permiten programar nuestra placa Makeblock con el software de Arduino de una manera más sencilla.

Lo primero que necesitamos es conocer el modelo de placa Makeblock que vamos a programar. En nuestro caso se trata de la placa MCore.

Los programas que vamos a realizar con Arduino IDE, tienen una serie de líneas que son comunes:

```
#include "MemCore.h"
```

```
void setup()
```

```
{  
  // Aquí introduciremos instrucciones para inicializar nuestro proyecto  
}
```

```
void loop()
```

```
{  
  // Aquí introduciremos las instrucciones de nuestro programa  
}
```

La primera línea nos sirve para indicar al sistema cuál es nuestra placa. A continuación los programas se forman en dos grandes bloques:

```
Void setup() { }
```

Es el bloque donde vamos a definir los datos iniciales de nuestro proyecto (ej, si un motor estará encendido o apagado al empezar el programa, si un LED estará encendido o apagado...).

```
Void Loop() { }
```

Es el bloque donde vamos a crear el programa cuyas instrucciones queremos que realice nuestro robot.

Una vez creado el programa, tenemos que guardarlo dentro de la placa Makeblock para que el robot pueda seguir sus órdenes.

Vamos a realizar dos ejemplos de programación.

### 5.3 MOSTRAR UN MENSAJE EN LA MATRIZ DE LEDs

Conectamos la matriz de LEDs a la placa MakeBlock, al conector número 4, y vamos a mostrar el texto HOLA.

```
#include "MemCore.h"
```

```
MeLEDMatrix ledMx(PORT_4); // Indicamos al programa que vamos a utilizar una  
// matriz de LEDs en el conector 4
```

```
void setup()
```

```
{  
  ledMx.setBrightness(6); // Indicamos el brillo de la luz de la matriz  
  ledMx.setColorIndex(1); // Indicamos el color de la luz  
}
```

```
void loop()
```

```
{  
  ledMx.drawStr(0,7,'HOLA'); // Indicamos que en la posición X=0 Y=7 escriba  
  // la palabra HOLA  
}
```

Dentro de la carpeta Makeblock del CD, localizarás varios archivos con extensión.h En estos archivos está definido el conjunto de instrucciones que podemos utilizar en Arduino IDE para trabajar con las tarjetas de Makeblock.

#### 5.4 MOSTRAR LOS NÚMEROS DEL 1 AL 100 EN LA MATRIZ DE LEDS

```
#include "MeMCore.h"           // Indicamos el tipo de tarjeta que vamos a
                                // programar

MeLEDMatrix ledMx(PORT_4);    // Indicamos que la tarjeta se conecta al conector 4

void setup()
{
  ledMx.setBrightness(6);      // Intensidad del brillo de la matriz de LEDs
  ledMx.setColorIndex(1);      // Color de la matriz de LEDs
}

int i=0;
void loop()
{
  for (int i=0; i<100; i++){   // desde el número 0 hasta el 100, incrementando
                                // de 1 en 1
    ledMx.showNum(i);          // Visualizamos el número en el display
    delay(100);                // esperamos 100 milisegundos
  }
}
```





**MICRO-LOG TECNOLOGÍA Y SISTEMAS, S.L.**

C/ Andrés Obispo, 37 • 28043 Madrid  
Telf. 91 759 65 22 • Fax 91 759 54 80  
E-Mail: [pedidos@microlog.es](mailto:pedidos@microlog.es)  
[www.microlog.es](http://www.microlog.es)



# ANEXO 1

## DE PRÁCTICAS COMPLEMENTARIAS

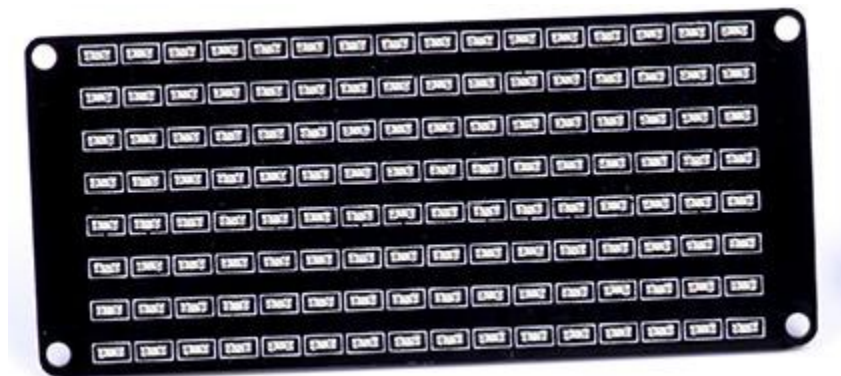
## Instrucciones para el módulo matriz Me LED 8×16

### 1. Introducción del módulo

**Voltaje de funcionamiento:** 5V DC

**Interfaz de comunicación:** interfaz digital doble (etiqueta azul). La matriz LED se puede conectar a la interfaz 1, 2, 3, y 4 de mCore y conectada a la interfaz 3, 4, 5, y 6 de Orion. La placa base se puede conectar con el interfaz de 3, 4, 5, 6, 7, y 8.

**Apariencia:**

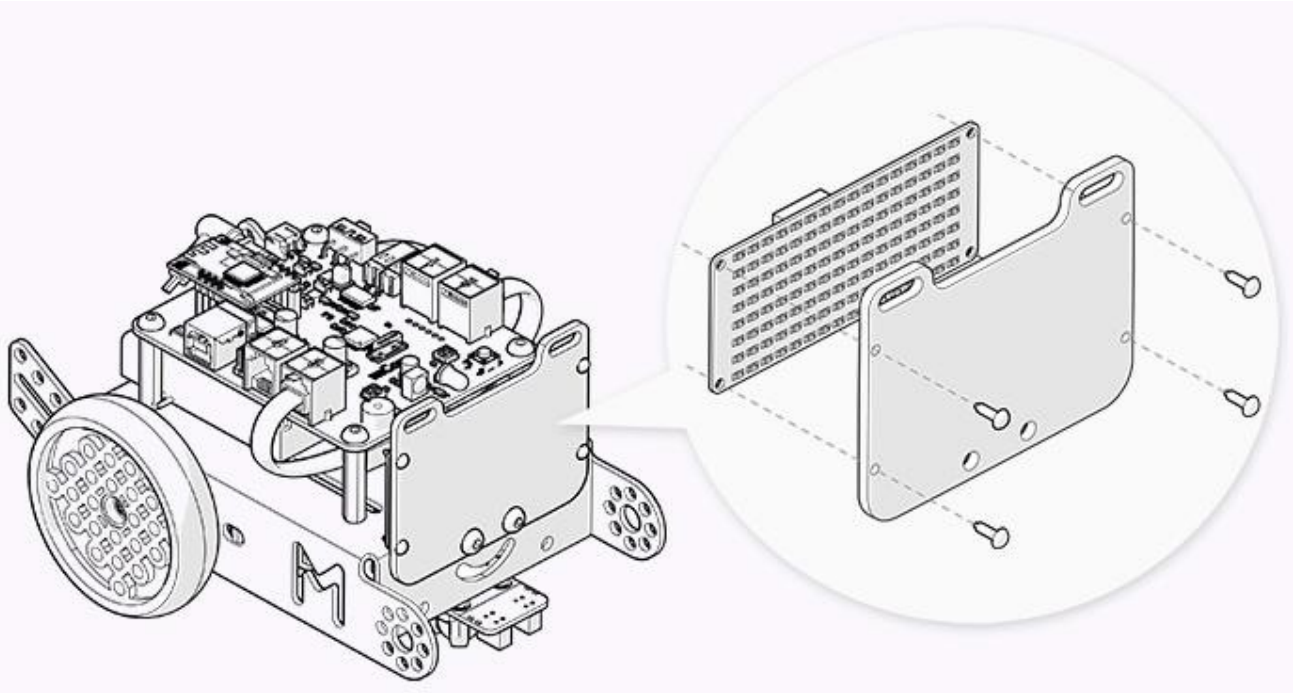


Frontal

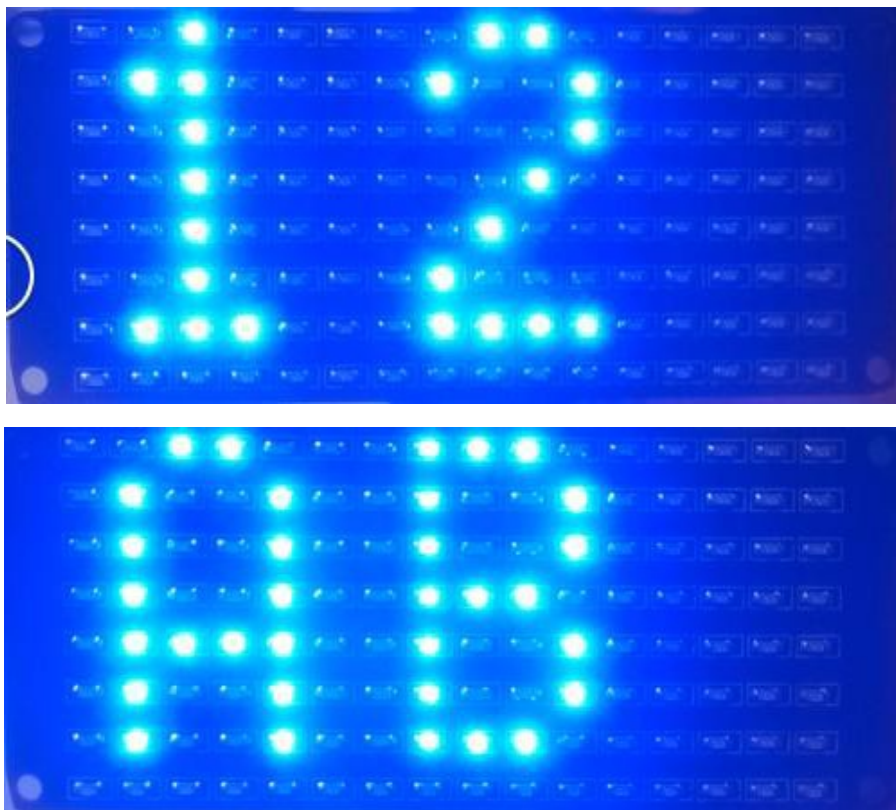


Posterior

Montaje:



**Aplicación:** El módulo es una pantalla de matriz de 8X16 LED compuesto por 8 lámparas LED azules verticales y 16 horizontales. Durante el encendido o apagado de las lámparas LED, podrán mostrarse ciertas figuras básicas, letras y patrones simples. Véase el siguiente diagrama:



## 2. Instrucciones

### 2.1 Programación de Arduino IDE

Consejos: Cómo ver el código fuente del módulo matriz de LED

Primero, abra el software mBlock, luego seleccione "Extensions->Manage ->mBot->view source code" (Extensiones, administrar, mbot, ver código Fuente)

#### Biblioteca de funciones

Nombre de la función	Descripción	Propiedad
<a href="#">MeLEDMatrix</a>	Constructor	Función pública
<a href="#">clearScreen</a>	Función de borrar pantalla	Función pública
<a href="#">setBrightness</a>	Función de ajuste de brillo	Función pública
<a href="#">setColorIndex</a>	Ajuste de índice de color (1: Iluminación normal, 0: sin color);	Función pública



<a href="#">drawBitmap</a>	Mostrar función de mapa de bits (diagram cuadrículado)	Función pública
<a href="#">drawStr</a>	Mostrar función de cadena de caracteres	Función Pública
<a href="#">showClock</a>	Mostrar función de reloj	Función pública
<a href="#">showStr</a>	Mostrar función de cadena de caracteres	Función privada

## Introducción de funciones de desarrollo

### 1. MeLEDMatrix Constructor 1

Nombre de función	MeLEDMatrix
Objeto de función	MeLEDMatrix ();
Descripción de desarrollo	constructor vacío
Parámetros de entrada	N/A
Valor devuelto	N/A
Prerequisito	N/A
Función de llamada	N/A

[Volver a la biblioteca de funciones→](#)

### 2. MeLEDMatrix Constructor 2

Nombre de función	MeLEDMatrix
Objeto de función	MeLEDMatrix (uint8_t port);
Descripción de desarrollo	Constructor. Inicializar el módulo de acuerdo con los parámetros del puerto;
Parámetros de entrada	Puerto: Módulo control de puerto. Rango de valores: PORT_1 ~ PORT_8

Valor devuelto	N/A
Prerequisito	N/A
Función de llamada	writeByte ( ); // Función de escritura de un byte; setBrightness ( ); // Función de ajuste de brillo; clearScreen ( ); // Función de borrado de pantalla;

[Volver a la biblioteca de funciones→](#)

### 3. MeLEDMatrix Constructor 3

Nombre de función	MeLEDMatrix
Objeto de función	MeLEDMatrix (uint8_t SCK_Pin,uint8_t DIN_Pin);
Descripción de desarrollo	Constructor. Inicializar módulo de acuerdo con los parámetros de pin;
Parámetros de entrada	SCK_Pin: Pin de módulo de control de línea de reloj DIN_Pin: Pin de módulo de control de línea de datos
Valor devuelto	Nil
Prerequisito	Nil
Función de llamada	writeByte ( ); // función de escritura de un byte; setBrightness ( ); // Función de ajuste de brillo; clearScreen ( ); // Función de borrado de pantalla;

[Volver a la biblioteca de funciones→](#)

### 4. clearScreen (limpiar pantalla)

Nombre de función	clearScreen
Objeto de función	void clearScreen ();
Descripción de desarrollo	Función de limpieza de pantalla
Parámetros de entrada	N/A

Valor devuelto	N/A
Prerequisito	N/A
Función de llamada	writeBytesToAddress ( ); // Escribe algunos bytes en la dirección especificada;

[Volver a la biblioteca de funciones→](#)

#### 5. setBrightness (ajustar brillo)

Nombre de función	setBrightness
Objeto de	void setBrightness (uint8_tBright);
Descripción de desarrollo	Función de ajuste de brillo
Parámetros de entrada	Bright: Parámetros de brillo. Rango de valores:  Brightness_0 // El más oscuro (apagar) Brightness_1 Brightness_2 Brightness_3 Brightness_4 Brightness_5 Brightness_6 Brightness_7 Brightness_8 // El más brillante
Valor devuelto	N/A
Prerequisito	N/A
Función de llamada	writeByte ( ); // función de escritura de un solo byte;

[Volver a la biblioteca de funciones→](#)

#### 6. setColorIndex (ajustar color)

Nombre de función	setColorIndex
Objeto de función	void setColorIndex (bool Color_Number);
Descripción de desarrollo	Color index show setting (1: Normally lighting, 0: Color negation);
Parámetros de entrada	Color_Number: Parámetros de color. 1: Mostrar diagrama con lámparas de iluminación; 0: Mostrar con las lámparas desactivadas;

	Observaciones: <1> El parámetro de color del Sistema de LED por defecto en pantalla de cuadrícula es 1 (mostrando el diagrama con las lámparas de iluminación) <2> clearScreen( ), después de llamar a la función de borrado de pantalla, el parámetro de color se reseteará al 1;
Valor devuelto	N/A
Prerequisito	N/A
Función de llamada	N/A

[Volver a la biblioteca de funciones](#)→

### 7. drawBitmap (dibujar mapa de bits)

Nombre de función	drawBitmap
Objeto de función	void drawBitmap (uint8_t x, uint8_t y, uint8_t Bitmap_Width, uint8_t *Bitmap);
Descripción de desarrollo	Mostrar la función de mapa de bits (diagrama cuadrícula. La altura del mapa de bits se fija en 8 (píxeles);
Parámetros de entrada	x: Rango de valores de la <b>esquina superior izquierda</b> de la coordenada x del mapa de bits (diagrama de red): 0 ~ 15; y: Rango de valores de la <b>esquina superior izquierda</b> de la coordenada y del mapa de bits(diagrama de red): 0 ~ 7; Bitmap_Width: Ancho (píxeles) del mapa de bits (diagrama de red) : Bitmap: nombre de la matriz del código de mapa de bits de red (buffer area);
Valor devuelto	N/A
Prerequisito	N/A
Función de llamada	writeBytesToAddress ( );      Escribe algunos bytes en la dirección especificada;

[Volver a la biblioteca de funciones](#)→

### 8. drawStr

Nombre de función	drawStr
Objeto de función	void drawStr (int16_t X_position, int8_t Y_position, const char *str);
Descripción de desarrollo	Función de muestra de las cadenas de caracteres. Provisionalmente, sólo es capaz de mostrar las letras mayúsculas y minúsculas, cifras y espacios en blanco, y el carácter desconocido se dejará en blanco;

	La altura de la cadena de caracteres se fijará en 8 (píxeles);
Parámetros de entrada	X_position: Esquina inferior izquierda x-coordina la primera cadena de caracteres. Rango de valores: El valor más pequeño se decide por la longitud de las cadenas de caracteres, Valor más grande: 16  Y_position: Bottom left corner Y-coordinate of first character of strings Value range: -1 ~ 15;  str: La dirección inicial de cadenas de caracteres en buffer área(matriz) que se muestra;
Valor devuelto	N/A
Prerequisito	N/A
Función de llamada	showStr (); // función de muestra de cadena de caracteres, función privada

[Volver a la biblioteca de funciones →](#)

#### 9. showClock (mostrar reloj)

Nombre de función	showClock
Objeto de función	void showClock (uint8_t hour, uint8_t minute, bool point_flag = PointOn);
Descripción de desarrollo	Función de muestra de reloj
Parámetros de entrada	hour: Hora;  minute: Minuto;  point_flag: Dos puntos muestran la etiqueta entre el reloj (colon) point_flag = PointOn (point_flag = 1) indica qué dos puntos son mostrados; point_flag = PointOff (point_flag = 0) indica qué dos puntos no son mostrados
Valor devuelto	N/A
Prerequisito	N/A
Función de llamada	writeBytesToAddress ( ); // Escribe algunos bytes a la dirección especificada;

[Volver a la biblioteca de funciones →](#)

Código de ejemplo:

```

#include<mBot.h>
#include <MeLEDMatrix.h>
#include <MePort.h>
//MeBoard myBoard(MakeblockOrion);
MeBoard myBoard(mBot);
MeLEDMatrix Matrix_1(PORT_4);
char string_data[]="MAKEBLOCK 012345678";
uint8_t Bitmap_Heart[16]=
{
    0x00,0x38,0x44,0x42,0x21,0x21,0x42,0x44,0x38,0x44,0x42,0x21,0x21,0x42,0x44,0x38,
};

int move_times =sizeof(string_data)*6;

void setup()
{
    Matrix_1.setBrightness(Brightness_8);//

}

void loop()
{
    /***** Ejemplo1 *****/
    Matrix_1.setColorIndex(1);//
    Matrix_1.drawBitmap(0, 0, sizeof(Bitmap_Heart), Bitmap_Heart);
    for(uint8_t k=0; k<3; k++)
    {
        for(uint8_t i=0;i<8;i++)
        {
            Matrix_1.setBrightness(i);
            delay(100);
        }

        for(uint8_t i=7;i>0;i--)
        {
            Matrix_1.setBrightness(i);
            delay(100);
        }
    }
    /***** Ejemplo 1 *****/

    /***** Ejemplo 2 *****/
    Matrix_1.setColorIndex(0);

```



```

Matrix_1.drawBitmap(0, 0, sizeof(Bitmap_Heart), Bitmap_Heart);
for(uint8_t k=0; k<3; k++)
{
    for(uint8_t i=0;i<8;i++)
    {
        Matrix_1.setBrightness(i);
        delay(100);
    }

    for(uint8_t i=7;i>0;i--)
    {
        Matrix_1.setBrightness(i);
        delay(100);
    }
}

```

\*\*\*\*\* Ejemplo 2\*\*\*\*\*/

/\*\*\*\*\* Ejemplo 3\*\*\*\*\*

```

for(int16_t i=0; i<move_times; i++)
{
    if(i >move_times)i=0;
    Matrix_1.drawStr(15-i,7,string_data);

    delay(100);
}

```

\*\*\*\*\* Ejemplo 3\*\*\*\*\*/

/\*\*\*\*\* Ejemplo 4 \*\*\*\*\*

```

for(uint8_t k=0; k<5; k++)
{
    Matrix_1.showClock(12, 34, PointOn);
    delay(500);
    Matrix_1.showClock(12, 34, PointOff);
    delay(500);
}

```

\*\*\*\*\* Ejemplo 3\*\*\*\*\*/

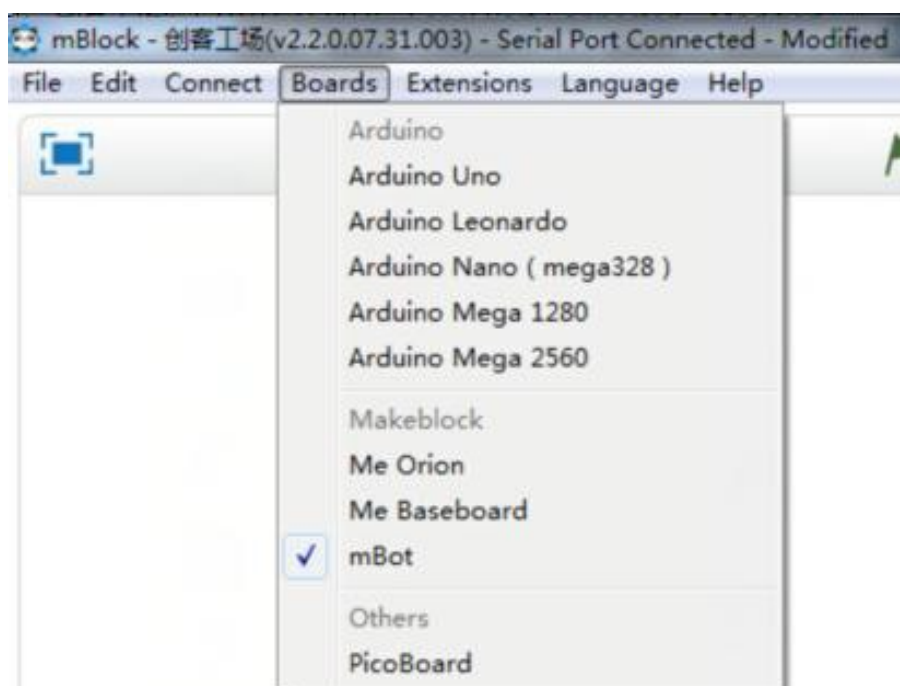
}

## 2.2 mBlock programación

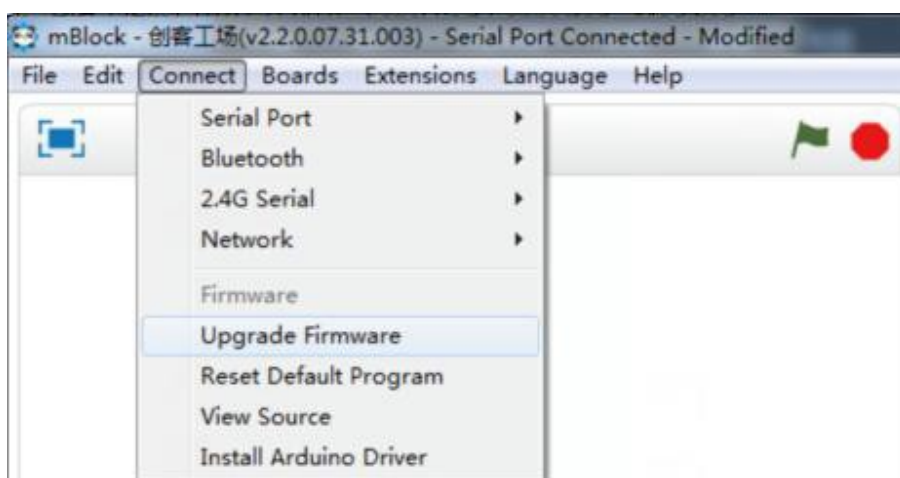
- **Paso 1** Después de conectar, mediante cable USB y encender el Mbot, tendrá que elegir el puerto de serie en primer lugar.



- **Paso 2**, elige el tipo de placa



- **Paso 3**, descargar el Firmware



- **Paso 4**, arrastre los bloques de script a la matriz LED



## Introducción bloques de script

### 1) Caracteres de entrada



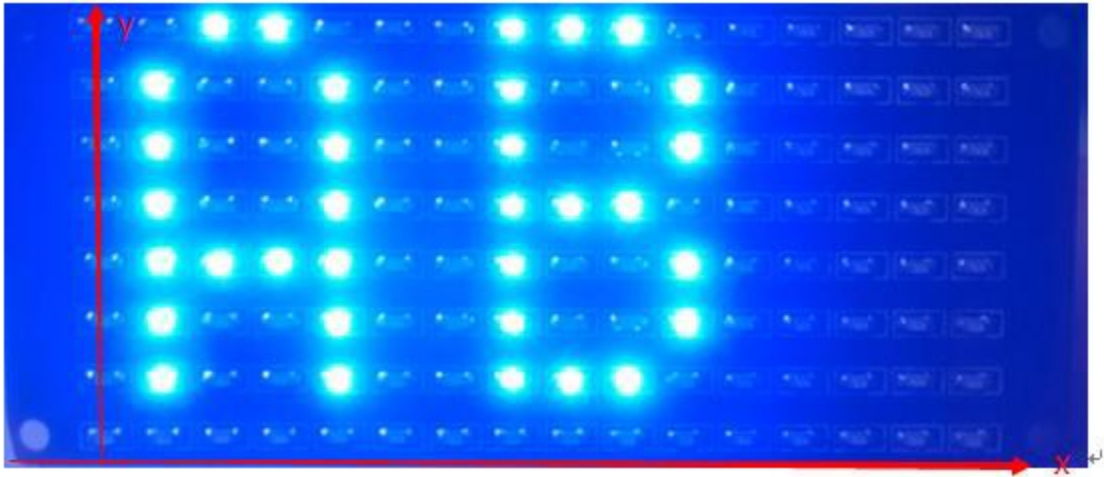
### Parámetro:

“Port1” – Elige el Puerto correspondiente al que ha conectado la matriz LED

“x:0” – Establece la coordenada x a 0.

“y:0” – Establece la coordenada y al 1.

“characters:Hello” – Establece los caracteres mostrados como Hello



## 2) Tiempo de entrada

show time **Port1** hour: **10** : min: **20**

### Parámetro:

“Port1” – Elige el Puerto correspondiente al que ha conectado la matriz LED.  
About hour&min – Los caracteres serán mostrados con la hora actual

## 3) Patrones y caracteres definidos por el usuario

show drawing **Port1** x: **0** y: **0** draw:

Haga clic en el área representada y el panel indicado se mostrará en la pantalla.



Seleccione las lámparas LED que desee activar y haga clic, se mostrará en -1 ~ 15; la pantalla

# ANEXO 2

## DE PRÁCTICAS COMPLEMENTARIAS





**mBot Servo Pack**  
**Gato Bailarín**

Apreciamos sus opiniones sobre nuestros productos. Por favor,  
póngase en contacto con nosotros con sus sugerencias en:

<http://www.spc-makeblock.es/soporte/>

[www.spc-makeblock.es](http://www.spc-makeblock.es)

**mBot Courses** Montaje, paso a paso, del gato bailarín

Tornillos (escala 1:1)



Tornillo M4\*8



Tornillo cabeza de cruz M2\*10



Tornillo autoperforante 2.2\*6.5



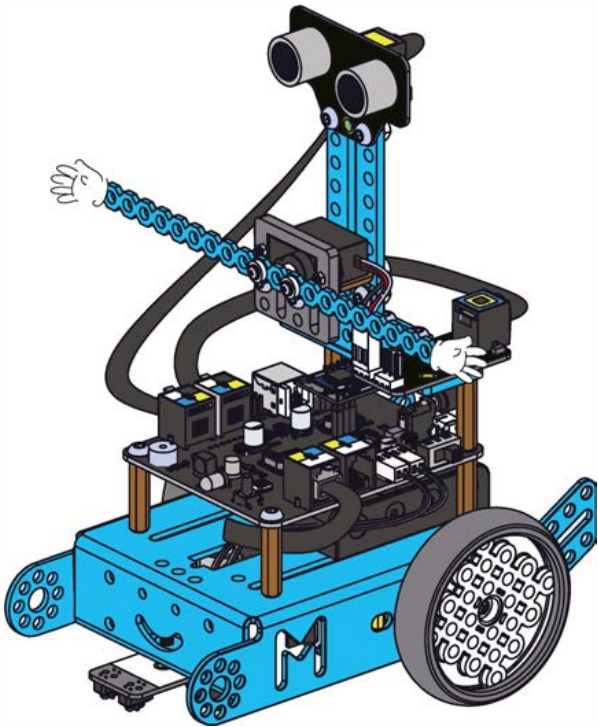
Tornillo autoperforante 2.2\*8



Tuerca: M2

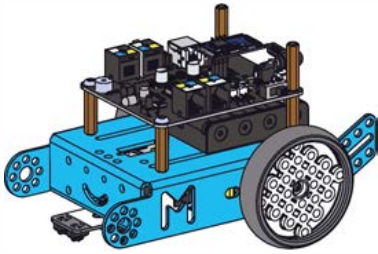


Tuerca M4



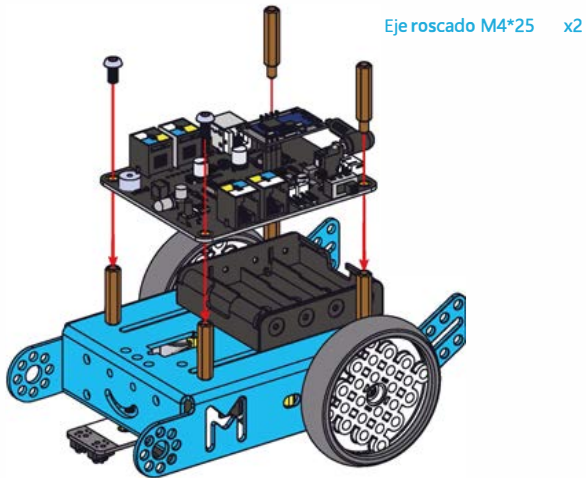
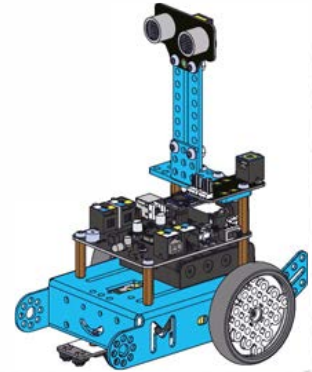
## 1 Reconocer el mBot

Después de reconponer



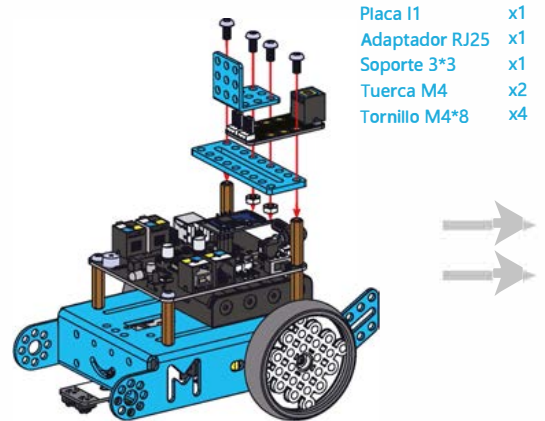
## 2 Instalar el soporte y el sensor de ultrasonidos

Después de la instalación



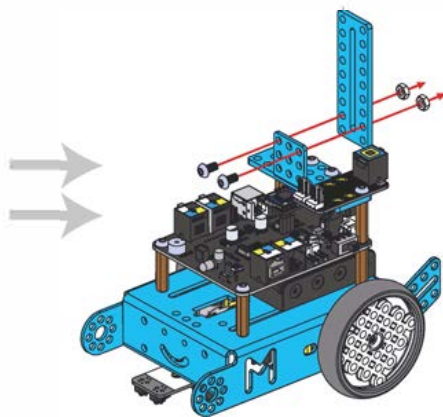
Eje roscado M4\*25 x2

3

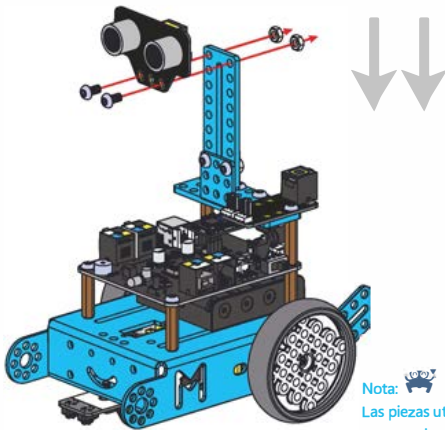



Placa I1 x1  
Adaptador RJ25 x1  
Soporte 3\*3 x1  
Tuerca M4 x2  
Tornillo M4\*8 x4

4



- Placa I1 x1
- Tornillo M4\*8 x2
- Tuerca M4 x2

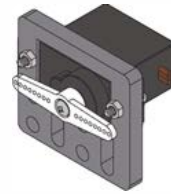


Nota:  Las piezas utilizadas en este paso son las piezas desmontadas del Mbot original

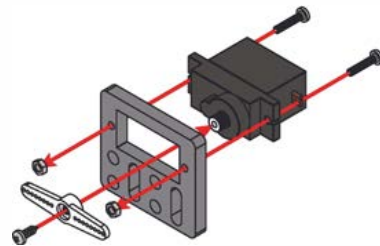


### 3 Instalar el servo Pack

Después de la instalación:

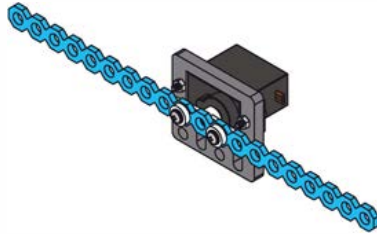


- 9g Micro Servo x1
- 9g Soporte Micro Servo x1
- Tornillo de cabeza decruz M2\*10 x2
- Tornillo autoperforante 2.2\*6.5 x1
- Tuerca M2 x2
- Servo brazo x1

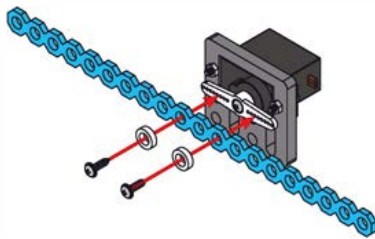


#### 4 Instalar la viga recortable

Después de la instalación:

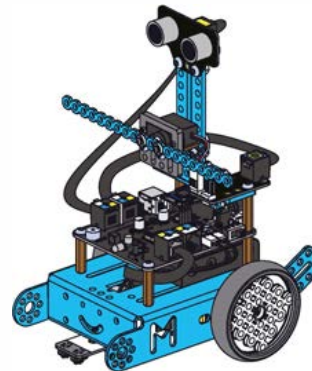


- Tornillo auto perforante 2.2\*8 x2
- Aro de plástico 4\*7\*2mm x2
- Viga recortable x1

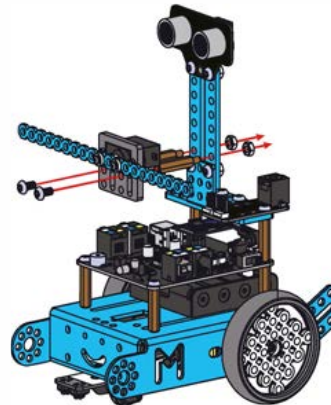



#### 5 Montaje completado

Montaje completado:



- Tuerca M4 x2
- Tornillo M4\*8 x2



Nota;  Puede añadir algún moldeado de papel en forma de mano o atar algunos globos en la viga recortable para realizar interacciones interesantes



