

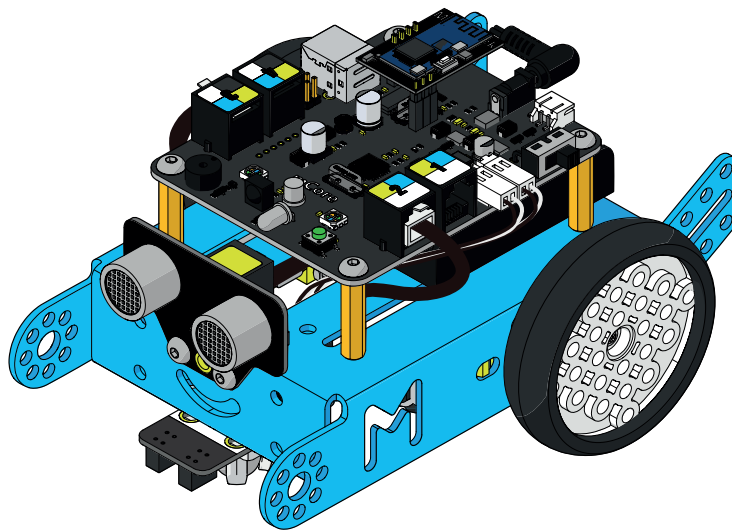


PC Factory
Área de profesionales y especialistas
Equipo de computación y electrónica

Manual de usuario Rev. 1

Makeblock mBot

Kit robótico educacional



Índice

| | | |
|-----------|--|-----------|
| 1 | Introducción..... | 2 |
| 2 | Especificaciones técnicas | 2 |
| 3 | Piezas del kit robótico | 3 |
| 4 | Microcontrolador | 3 |
| 5 | Ensamblaje y diseño..... | 4 |
| 6 | Sensores | 7 |
| 7 | Actuadores | 11 |
| 8 | Programa Base..... | 14 |
| | 8.1 IR Control remoto..... | 14 |
| | 8.2 Makeblock App..... | 14 |
| 9 | Consideraciones..... | 16 |
| 10 | Material Complementario | 17 |

1. Introducción

mBot es un robot diseñado por Makeblock con fines educativos. Este corresponde al modelo básico de la compañía con el cual se ingresa a todo lo que es la programación interactiva, el uso de sensores, actuadores y robótica en general. Para lograr este cometido cuenta con una gama de aplicaciones, vídeos y manuales para su uso. El kit educacional en el cual el mBot viene incluido cuenta con todo lo necesario para ser autosuficiente, es decir, incluye todo lo necesario para ensamblar y utilizar el robot a excepción de las baterías.

En las próximas secciones se utilizarán nombres en inglés de algunas componentes para evitar ambigüedades

2. Especificaciones técnicas

| Categoría | Especificación |
|--|---|
| Software para programación (Mac/Windows/Linux) | mBlock, Arduino IDE |
| App (iPad/Android) | Makeblock, mBot |
| Sensores | 1x Sensor de luz 1x Botón 1x Receptor IR (infrarrojo) 1x Sensor ultrasónico 1x Sensor seguidor de línea |
| Actuadores | 1x Buzzer (Pasivo) 2x LED RGB 1x Transmisor IR 2x Motor DC |
| Microcontrolador | Arduino Uno |
| Alimentación | 3.7V Batería de Litio (cargador integrado) o 4x Baterías AA (no incluidas) |
| Comunicación inalámbrica | Bluetooth o 2.4G serial |
| Estructura (Chasis) | 2mm Aluminio |
| Dimensiones (ensamblado) | 17 x 13 x 9 cm |
| Peso (ensamblado) | 1034 g |
| Capacidad de carga (ensamblado) | Sobre 1600g |
| Rapidez máxima | 0.35 m/s |

Tabla 1: Especificaciones técnicas mBot

3. Piezas del kit robótico



Figura 1: Componentes en el kit

Además de todo esto, el kit incluye piezas de repuesto para reemplazar las reducciones de uno de los motores DC en caso de que se gasten o rompan.

4. Microcontrolador

El cerebro del mBot corresponde a la tarjeta mCore, esta es una tarjeta de desarrollo diseñada por Makeblock la cual está construida entorno a un microcontrolador ATmega328p, mejor conocido como Arduino Uno. La tarjeta en cuestión se puede ver en la Figura 2, esta cuenta con una conexión USB tipo B para mejor durabilidad y permitir la programación con PC. Con respecto a la alimentación, esta puede trabajar con voltajes desde 3.7V a 6V los cuales pueden ser suministrados a través de conectar un pack de 4 baterías AA o una batería de litio de 3.7V, para esta última cuenta con un cargador USB integrado. El resumen de estos datos se puede ver en la Tabla 2. Además, cuenta con protección ante altas de corriente y de voltaje en la forma de Diodos Schottky en conjunto a diodos TVS (*Transient-voltage-suppression diode*) y fusibles reiniciables (*resettable fuses*). Todo esto último con fin de mantener los niveles de voltaje del Arduino estables en 5V y limitar la corriente máxima a 1.1A, para evitar problemas ante cortocircuitos o malas conexiones.

Para el manejo de sensores y actuadores, cuenta con conexiones RJ25 las cuales están mapeadas a diferentes pines del microcontrolador (ver Figura 3). Del *pin mapping* se desprende que los PORT3-PORT4 sirven para el manejo de dispositivos analógicos, como un sensor de temperatura, mientras que el PORT1-PORT2 sirven para procesar señales digitales. Dentro de esto cabe destacar la posibilidad de ocupar cualquiera de estos puertos para realizar comunicación serial I²C, con especial consideración en el PORT1 que además es capaz de realizar comunicación SPI (internamente el puerto de comunicación SPI está conectado a los LED RGB). Además a través de unos pines en la placa se puede monitorear la comunicación serial UART (esta es la misma que se utiliza para la comunicación por USB y para los módulos inalámbricos).

| Categoría | Especificación |
|----------------------|---|
| Voltaje de operación | 3.7-6 V_{DC} |
| Fuente de potencia | 4 x Baterías AA ó 1 x Batería Litio 3.7V |
| Microcontrolador | ATmega328 (Arduino UNO) |
| Conexión USB | Tipo B |
| GPIO | 2 x Entrada analógica (PORT3 y PORT4) 4 x Entrada digital (Todos los PORTn) |
| Comunicación | 1 x UART (USB y <i>Wireless</i>) 4 x I ² C (Todos los PORTn) 1 x SPI (PORT1, 3x2 Pin Header, LED RGB) |
| Dimensiones | 90 x 79 x 18mm (3.54 x 3.11 x 0.71inch) |
| Peso Neto | 43g (1.52oz) |

Tabla 2: Tabla de especificaciones

5. Ensamblaje y diseño

El empaquetado del kit está altamente organizado, en este las piezas tienen un espacio a su medida en el cual se guardan o vienen dentro de bolsas herméticas como se muestra en la Figura 4. Para el ensamblaje, el kit cuenta con un manual el cual muestra paso a paso lo que es necesario realizar para armar el robot. En la Figura 5 se ve un ejemplo de las instrucciones de ensamblado, el diseño de estas es similar al utilizado por la empresa LEGO y tiene como foco principal su fácil entendimiento y replicación. Es a través de estas instrucciones que el robot puede ser armado con facilidad por niños y jóvenes en menos de 30 minutos sin experiencia previa.

La estructura del robot tiene múltiples consideraciones para mejorar la experiencia del usuario, varias de estas están pensadas para aumentar su durabilidad o facilidad de uso. Es gracias a estas consideraciones que mBot puede ejercer su propósito de ser un robot para

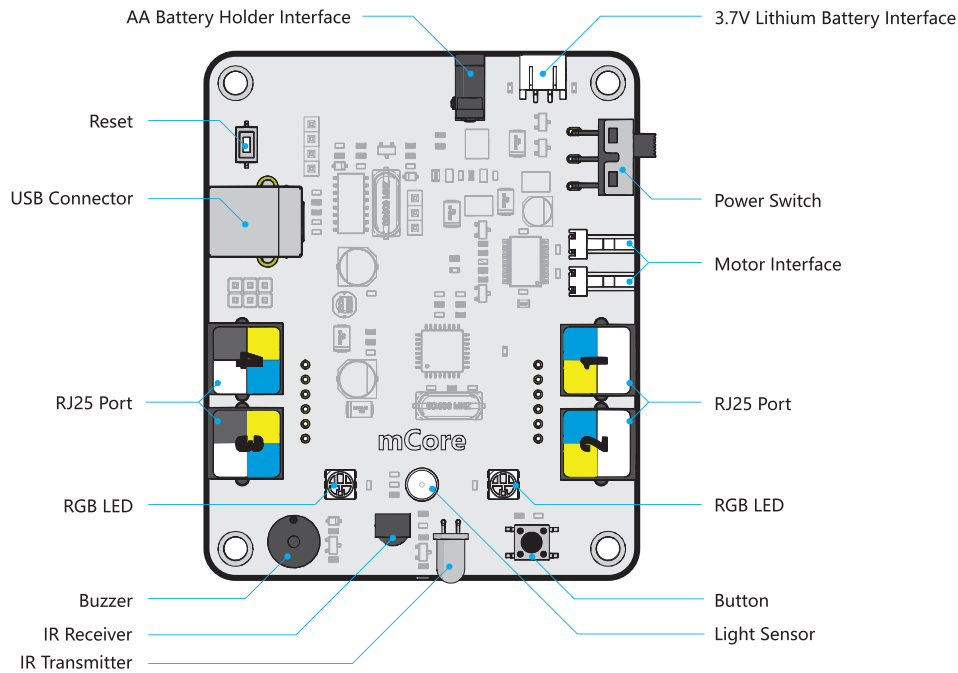


Figura 2: Tarjeta mCore

niños y no fallar en el proceso. En particular, las consideraciones más importantes son:

- El chasis cuenta con una ranura superior para organizar por donde pasarán los cables de conexión. Además, cuenta con brazos delanteros que evitan impactos de choque sobre los sensores y cuentan con ranuras para la extensión de piezas. Estas se presentan en la Figura 4 (a).
- La conexión de cables y posición de los elementos del robot es bastante intuitiva, para las conexiones además se tiene un código de colores que clasifica a los sensores con sus puertos compatibles en la tarjeta. Además se utilizan cables recubiertos con goma o similares, para aumentar su resistencia.
- Detalles visuales para conseguir apariencia adorable, entre ellos, la simulación de un rostro y manos. Dentro de estos se encuentra: la posición del ultrasónico, corte en forma de sonrisa en el chasis, posición de los brazos delanteros.

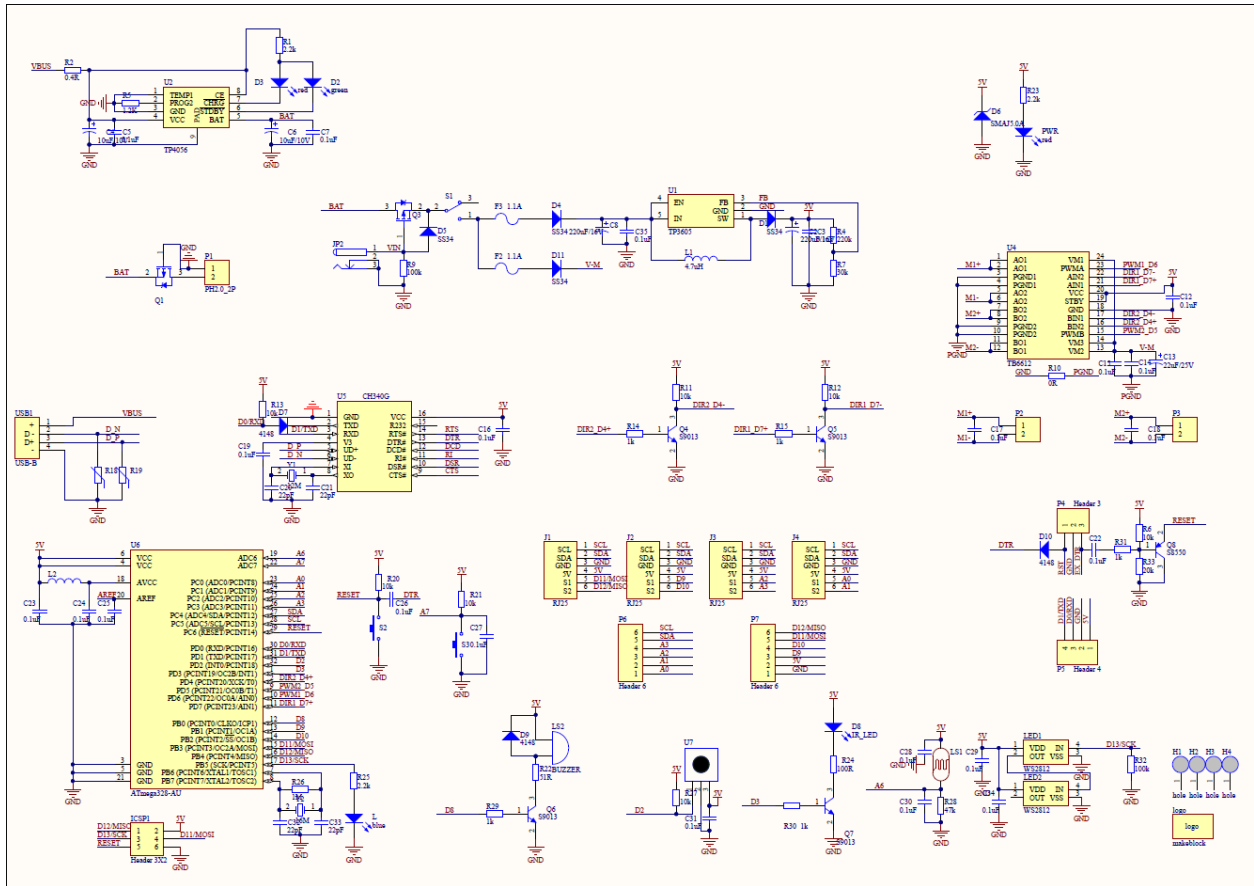


Figura 3: Esquemático mCore



(a) Presentación inicial del empaque



(b) Organización de las piezas

Figura 4: Empacado del robot

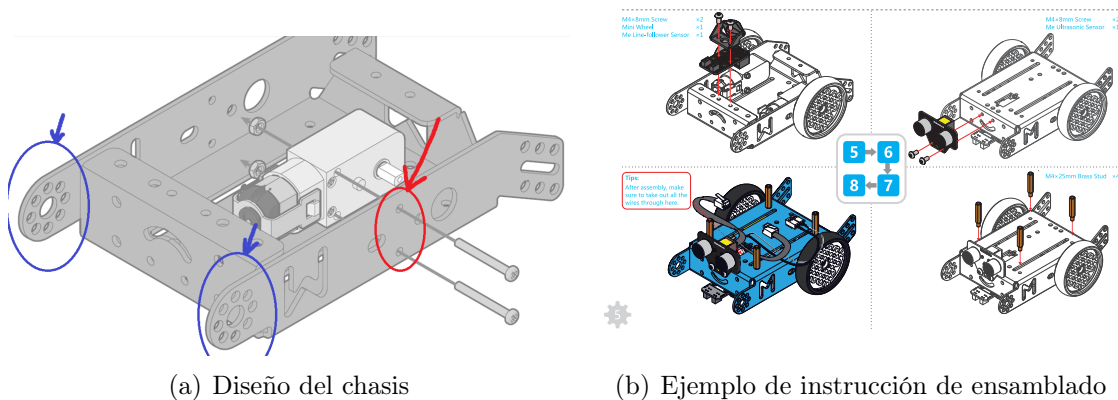


Figura 5: Ejemplo de instrucción de ensamblado

6. Sensores

El robot cuenta con una serie de sensores que pueden ser usados a través de la tarjeta mCore para crear diferentes algoritmos y proyectos. Estos se separan en dos bloques principales, aquellos que vienen integrados en la tarjeta y los que se conectan externamente en forma de módulos electrónicos.

- *Light Sensor*/Sensor de luminosidad: Foto-transistor cuya señal corresponde a una medición del nivel de luminosidad en el entorno, está conectado a la entrada analógica **A6** en el Arduino. Sus mediciones son directamente proporcionales al nivel de luz, donde una alta iluminación corresponde a un alto valor de voltaje. En condiciones normales, con la iluminación de una oficina, se pueden obtener valores cercanos a 990. Los datos de este se encuentran en la Tabla 3 y un código de ejemplo para usarlo es el siguiente:

```
MeLightSensor lightsensor_6(6); //Sensor de luz en pin 6
int lux = lightsensor_6.read(); //Valor luminico
```




Figura 6: Light Sensor

| Características primarias | | |
|----------------------------|----------------------|------|
| Categoría | Min | Max |
| Voltajes de operación (V) | 0.3 | 5 |
| Lectura MCU ⁽¹⁾ | 54 | 1023 |
| Librería asociada | MeLightSensor | |
| mBlock | light sensor (portN) | |
| Ubicación | Integrado | |

Tabla 3: Especificaciones sensor de luminosidad

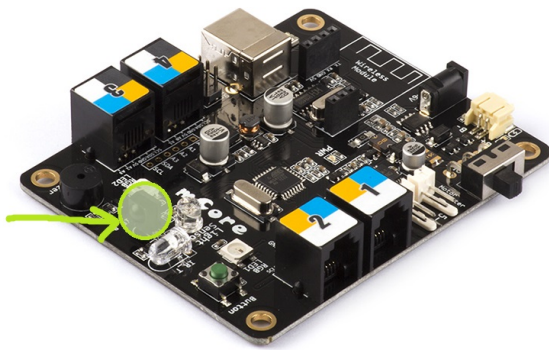
⁽¹⁾Valores medidos con la configuración predeterminada del ADC (Analog-to-Digital Converter)

- *IR Receptor*/Receptor de infrarrojo: Receptor de ondas infrarrojas, programado para decodificar instrucciones transmitidas en protocolo NEC IR [1] y se encuentra conectado a la entrada digital D2 del Arduino. Un código de prueba para este es el siguiente:

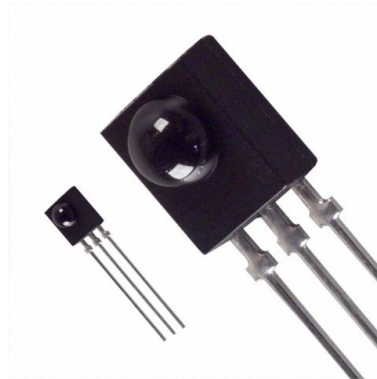
```

MeIR ir;
if(ir.keyPressed(70)){           // verifica si la tecla <B> (equivalente
                                // al numero 70) fue presionada
    //... mas codigo...
}

```



(a) Receptor IR en placa



(b) IR Receptor

Figura 7: Empacado del robot

- *Ultrasonic Sensor*/Sensor Ultrasónico: Dispositivo capaz de enviar y recibir ondas ultrasónicas, la información proveniente de este es de naturaleza digital y se obtiene del módulo electrónico externo Me Ultrasonic Sensor. En el programa por defecto del

mBot, este requiere ser conectado al PORT3 a través de un cable RJ25. Su método de funcionamiento corresponde al enviar una onda ultrasónica y luego recibir el eco de esta, con el tiempo transcurrido entre enviar el sonido y recibirlo se calcula la distancia hacia el objeto más cercano (se asumen 340m/s como velocidad del sonido para cálculos internos). Las especificaciones de este se encuentran en la Tabla 4. Un ejemplo de código corresponde al siguiente, donde se guarda la medición de distancia en la variable "dist":

```
MeUltrasonicSensor ultrasonic_3(3); //Ultra sonico en PORT3
int dist = ultrasonic_3.distanceCm(); //distancia medida en cm
```

| Características primarias | | |
|--------------------------------------|------------------------------------|------|
| Categoría | Min | Max |
| Voltajes digitales (V) | 0 | 5 |
| Rango distancias ⁽²⁾ (cm) | 3 | 4000 |
| Frecuencia portadora | 40 kHz | |
| Ángulo de medición ⁽³⁾ | 30° | |
| Librería asociada | MeUltrasonicSensor | |
| mBlock | ultrasonic sensor (PORTn) distance | |
| Ubicación | Externo | |

Tabla 4: Especificaciones módulo Me Ultrasonic Sensor

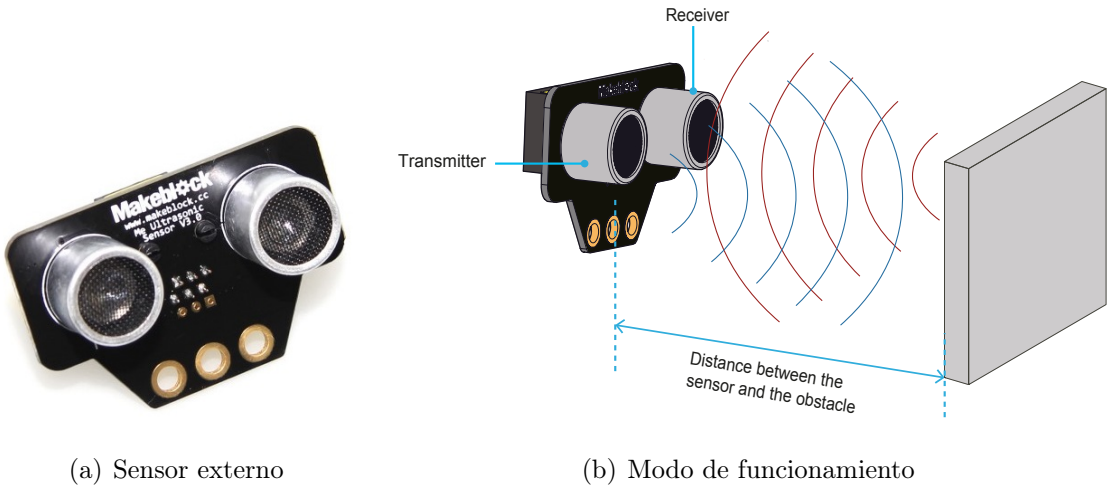


Figura 8: Módulo Me Ultrasonic Sensor

(2) Los valores cercanos y superiores a los 100cm son poco confiables debido a la alta probabilidad de ecos adicionales causados por el entorno.

(3) El ángulo de incidencia máximo se ve afectado por la distancia hacia el objeto.

- Line-follower sensor/Seguidor de línea:** Corresponde a un arreglo emisor-receptor de señales infrarojas, la información de este son señales de naturaleza digital y viene incluido en el módulo externo **Me Line Follower**. Este funciona a través de medir la diferencia de luz reflejada por superficies de diferentes colores, debido a la configuración digital del módulo de Makeblock, este solo puede diferenciar entre blanco y negro. El umbral utilizado en el módulo es tal que la tolerancia con respecto al negro es bastante baja, es decir, solo superficies bien opacas serán interpretadas como negro. La medición del sensor es entregada a través de una variable `uint_8` cuyos 2 bits menos significativos representan el estado actual de los receptores, por ejemplo, el estado IZQ = Blanco y DER = Negro se representa como $state = S_1S_2 = (10)_2 = 4$. Un código de prueba corresponde al siguiente:

```
MeLineFollower linefollower_2(2);           //sensor en PORT2
uint_8 state = linefollower_2.readSensors(); //estados: (S1S2)
```

| Características primarias | | |
|--------------------------------|---|--------|
| Categoría | Min | Max |
| Voltajes digitales (V) | 0 | 5 |
| Color detectado ⁽⁴⁾ | Negro | Blanco |
| Librería asociada | MeLineFollower | |
| mBlock | line follower (PORTn) (Side) is (color) | |
| Ubicación | Externo | |

Tabla 5: Especificaciones módulo Me Line Follower

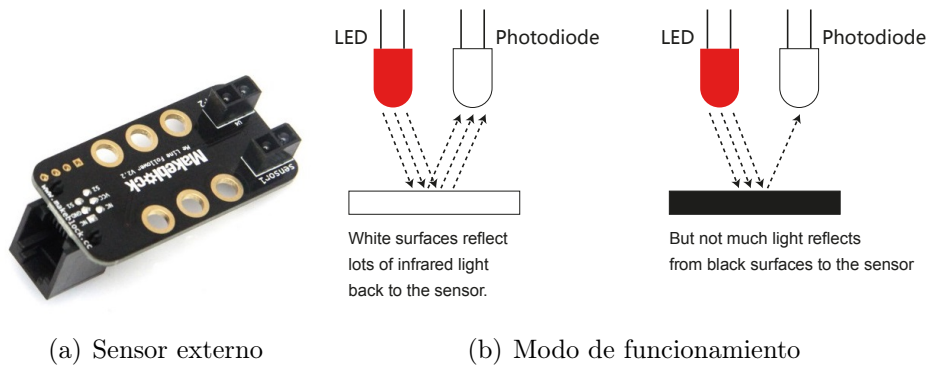


Figura 9: Módulo Me Line Follower

7. Actuadores

- *Buzzer*/Zumbador: Buzzer pasivo, elemento piezo eléctrico que se encarga de producir sonido de acuerdo a la señal que se le envíe, está conectado a la salida digital D5 del Arduino. Debido a ser del tipo pasivo, para su funcionamiento, se necesita enviar una señal PWM por el pin digital con la frecuencia deseada a sonar en el buzzer. En la Tabla 6 se presentan los datos del Buzzer. Un ejemplo de código es el siguiente, donde se toca una nota <La> durante 250 milisegundos:

```
MeBuzzer buzzer;  
int freq = 440;           //en Hz  
int duration = 250;      //en ms  
buzzer.tone(freq,duration); //toca la nota <La> durante 250ms
```

| Características primarias | | |
|-------------------------------|--|------|
| Categoría | Min | Max |
| Voltajes digitales (V) | 0 | 5 |
| Frecuencias de operación (Hz) | 50 ⁽⁵⁾ | 2500 |
| Librería asociada | MeBuzzer | |
| mBlock | play tone on note (Name) beat (Fraction) | |
| Ubicación | Integrado | |

Tabla 6: Especificaciones Buzzer

⁽⁵⁾Las notas con frecuencias inferiores a 200Hz presentan un alto nivel de frecuencias armónicas que distorsionan la señal.

- LED RGB: Diodo emisor de luz, corresponden a un arreglo de 3 diodos de diferente color (R, G y B) cuya suma de colores produce el color final resultante en el LED. En particular, se utiliza un arreglo de LEDs RGB provenientes del componente WS2812 para controlar varios LEDs con un solo pin de información y este se encuentra conectado a la salida D13/SCK del Arduino. Debido a esto último se requiere enviar la información a los LEDs de forma serial con el protocolo NZR, con este se deben enviar bloques de 24-bit con la configuración de cada componente de color (en el orden GRB). Las especificaciones se encuentran en la Tabla 7 y se puede encontrar mayor detalle sobre la comunicación con estos componentes en <https://cdn-shop.adafruit.com/datasheets/WS2812.pdf>. Por último un código de ejemplo de su uso:

```
MeRGBLed rgbled_7(7, 7==7?2:4); //LEDs RGB por defecto  
rgbled_7.setColor(0,60,15,0); //R=60, G=15, B=0 (placa)
```

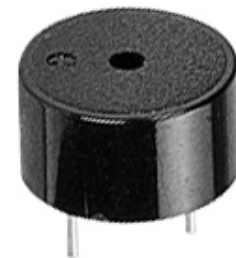
```
rgbled_7.show();
```

| Características primarias | Especificación | | |
|--------------------------------------|--|------|---------|
| Categoría | Min | Typ | Max |
| Voltajes alimentación (V) | 0 | 5 | 7 |
| Corriente directa (Vdd) | 0.1uA | - | 18.5 mA |
| Corriente comunicación (D_{OUT}) | - | 10mA | - |
| Cantidad de colores | 16.777.216 | | |
| Frecuencia comunicación | 800 kHz | | |
| Dimensiones | 50x50mm | | |
| Librería asociada | MeRGBLed | | |
| mBlock | set led on board (Side) red (value) green (value) blue (value) | | |
| Ubicación | Integrado | | |

Tabla 7: Especificaciones LED RGB



(a) RGB LED



(b) Buzzer

Figura 10: Actuadores integrados

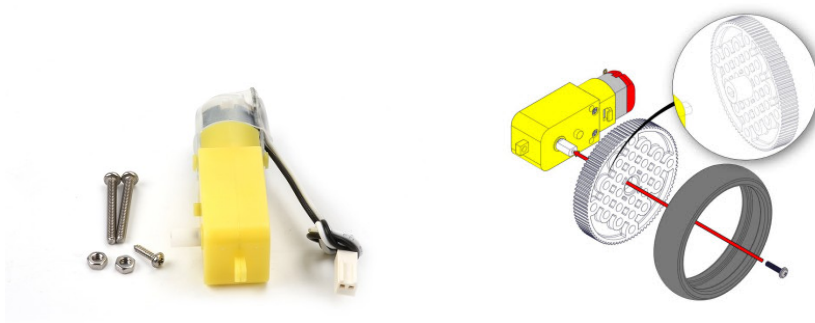
- Motor DC: Motores de corriente continua, funcionan a través de ser alimentados con un determinado valor de voltaje continuo entre sus terminales y producir un torque proporcional a dicha alimentación. Debido a trabajar con corrientes mayores a las que maneja el microcontrolador requiere de drivers para su uso (integrados), de esta manera su conexión corresponde a salidas PWM del Arduino. La señal PWM (*Pulse Width Modulation*) utilizada debe tener como ciclo de trabajo el porcentaje del voltaje total que se desea utilizar para los motores. Los motores incluyen estructuras de silicona para su ensamblaje y protección, además estas son compatibles con las ruedas de plástico incluidas en el kit (ver Figura 11). Los datos técnicos de estos motores se encuentran en la Tabla 8 y un ejemplo de código para utilizar estos motores es el siguiente:

```
MeDCMotor motor_9(9); //Conexion por defecto de M1
motor_9.run((9)==M1?-(-145):(-145)); //retroceso a rapidez 145
```

| Características primarias | Especificación | |
|--|--------------------------------|-----|
| Categoría | Min | Max |
| Voltaje nominal (V) | 3.7V | 6V |
| Duty Cycle ⁽⁶⁾ | 50 | 255 |
| Corriente SIN carga (mA) | <220mA | |
| Corriente CON carga (mA) | <1.1A | |
| Corriente máxima | <2.7A | |
| Rapidez máxima sin carga | 200 ± 10 % RPM | |
| Razón de engranajes | 1:48 | |
| Dimensiones (tubo protector de silicona) | D(1.5-4)mm*0.5mm | |
| Librería asociada | MeDCMotor | |
| mBlock | set motor (Name) speed (Value) | |
| Ubicación | Externo | |

Tabla 8: Especificaciones motores DC

⁽⁶⁾El porcentaje de duty cycle se calcula como $\%D = \frac{value}{255}$, donde value es un número entre 0 y 255 seteado por el usuario. En la práctica, valores menores a 45 no son capaces de producir un torque suficiente en el robot para que se mueva.



(a) Motor y piezas de unión

(b) Unión Motor DC a ruedas

Figura 11: Motor DC 200RPM

8. Programa Base

El programa base del mBot se puede separar en 2 bloques principales y estas dependen del dispositivo usado para enviarle instrucciones:

8.1. IR Control remoto

Esta corresponde al mando principal del mBot, ya que posee la prioridad más alta entre las otras señales de mando y entra en funcionamiento cuando se utiliza el control infrarrojo de regalo para hacerlo funcionar. Tiene 3 modalidades:

- Manual: Esta modalidad es elegida al presionar el botón <A> del control. En esta se encienden de color blanco los LEDs del robot y se procede controlar su movimiento con las flechas del control. Con los números del remoto se puede controlar la velocidad de movimiento, donde 0 corresponde a la más lenta y 9 a la más rápida. Además, cada vez que se presionan los números suena una nota musical por defecto.
- Esquiva-obstáculos: Se accede a esta modalidad al presionar la tecla en el remoto. En esta el robot comienza a moverse de forma automática y esquiva obstáculos en frente suyo haciendo uso del sensor ultrasónico. El algoritmo de evasión corresponde a dar un giro de 180° cuando se detecta un objeto a menos de 10cm de distancia.
- Seguidor de línea: Este modo se consigue al presionar la tecla <C> del control. En este, mBot empieza a avanzar en línea recta hasta que encuentra una línea negra (o similar) en el suelo y la comienza a seguir. La naturaleza de este modo requiere que haya un buen nivel de contraste entre el camino (línea negra) y el fondo (preferentemente blanco). Con este modo el robot puede seguir caminos arbitrarios dibujados por el usuario, por ejemplo, con plumón negro sobre una cartulina.

8.2. Makeblock App

El mBot viene programado para ser utilizado a través de la Makeblock App. Para el uso demostrativo de sus capacidades, mBot tiene acceso a varias secciones dentro de la categoría "Juego". En estas se presentan ejemplos de lo que puede hacer el mBot a través de interfaces didácticas e intuitivas, ejemplos de estas se pueden ver en la Figura 12. En (c) se tiene control del movimiento de manera gradual a través de lo que simula ser un mando analógico, además de tener botones con secuencias de movimientos como lo son "correr", "dar vueltas" y "terremoto". En (d) se muestra un piano musical con el cual se pueden tocar diferentes notas en el mBot, además de tener canciones de muestra que están pre-programadas en el robot (entre estas están Estrellita, Navidad, Feliz Cumpleaños y Fray Jacobo).

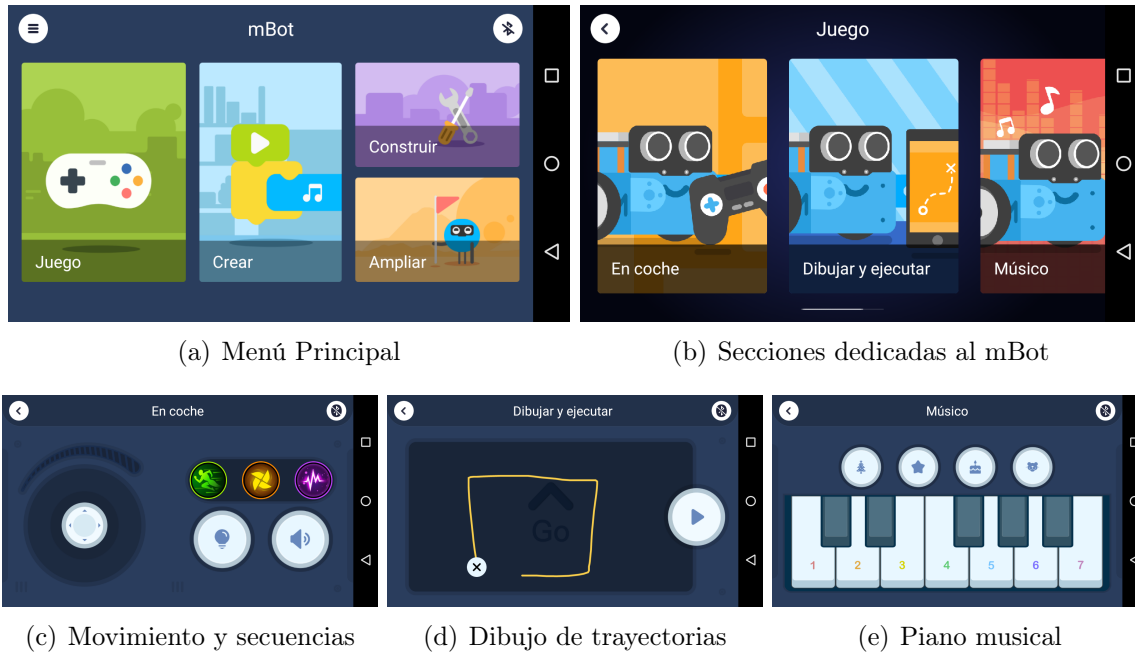


Figura 12: Zona de juegos

En la sección "Ampliar" se encuentran programas de prueba para los modelos armables con los packs de complementos (*Add-on*), como se muestra en Figura 13 (d). Los diferentes modos de operación disponibles a través de estas secciones, son los siguientes:

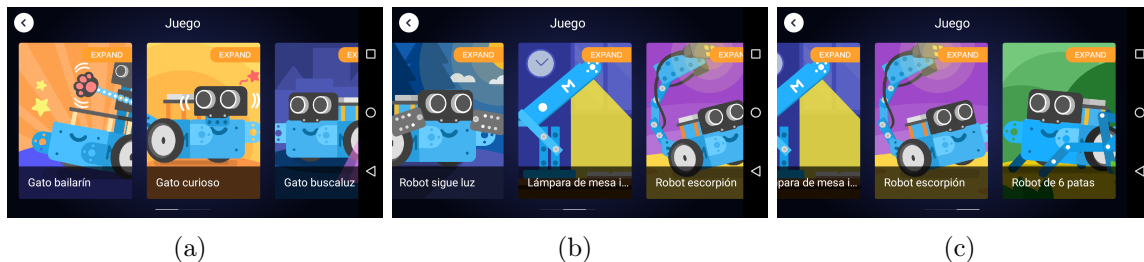
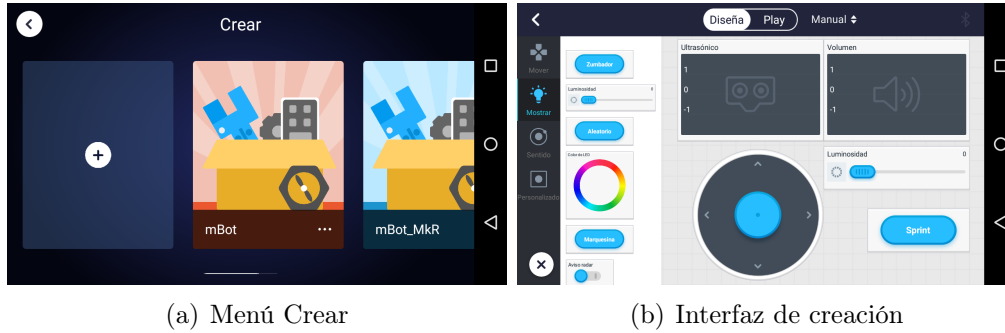


Figura 13: Secciones dedicadas a las extensiones del mBot

Además, está el acceso al menú "Crear" en el cual se puede entrar a lo que es la programación por bloques, junto al probar los sensores y actuadores a mayor cabalidad. Esto se puede ver en la Figura 14.

Finalmente, existe una aplicación "mBot" dedicada específicamente a enseñar a niños el *cómo* programar este robot. En esta se introducen conceptos base de lo que es la programación en bloques y algoritmos. En la Figura 15 se puede apreciar la interfaz de esta aplicación.



(a) Menú Crear (b) Interfaz de creación

Figura 14: Secciones dedicadas a la programación



(a) Interfaz de conexión (b) Menú de la App (c) Programación en modo historia

Figura 15: mBot App

9. Consideraciones

A continuación se presentan algunas consideraciones a tener en mente al momento de utilizar el robot mBot:

- **Control IR:** Requiere baterías CR2032 o CR2025, estas no son tan fáciles de encontrar y no vienen incluidas.
- **Modo manual:** control discreto de velocidades, no se pueden elegir valores intermedios. Las instrucciones de (\rightarrow) y (\leftarrow) a veces son confundidas.
- **Modo esquivo-obstáculos:** Mientras el mBot esta girando en este modo no es posible enviarle otras instrucciones, esto puede provocar que quede atascado entre un conjunto de obstáculos y sea necesario extraerlo de manera manual del sitio. Requiere de que el obstáculo tenga una altura y ancho mínimos para poder ser detectados, debido a la forma y posición del sensor ultrasónico (alrededor de unos 6~7cm sobre el suelo).
- **Seguidor de línea:** Requiere de un grosor mínimo para el camino utilizado (~3cm), que corresponde al ancho del módulo **Me Line Follower** utilizado. Además requiere

de que el camino tenga una buena curvatura para ser capaz de seguirla correctamente, el dibujo de un ∞ que viene incluido esta cerca del borde sus limitaciones. Como caso adicional a esto último, ante una curva cerrada (90°) es capaz de detectar el necesitar doblar inicialmente pero no es capaz de seguir el camino.

10. Material Complementario

Referencias

- [1] NEC, protocolo de comunicación con señales infrarojas. Explicación disponible en:
<http://www.sbprojects.com/knowledge/ir/nec.php>
- [2] mBot, kit robótico educacional de Makeblock. Disponible en:
<http://store.makeblock.com/mbot-v1-1-stem-educational-robot-kit>