



ROBOBO SCRATCH EXTENSION

Manual de usuario

23 de diciembre de 2016

Table of Contents

1Carga de la extensión..... 3

2Bloques de la extensión..... 4

2.1Tipos de bloque..... 4

2.2Bloques de gestión de conexión..... 4

2.3Bloques de Movimiento..... 5

2.4Bloques de interacción..... 8

2.5Bloques de sensorización..... 9

1 CARGA DE LA EXTENSIÓN

La extensión del ROBOBO está diseñada para su uso con la plataforma ScratchX, que permite la integración de extensiones “experimentales” con el entorno Scratch habitual.

Para ello las extensiones deben estar alojadas en Github de manera pública y se cargarán en ScratchX mediante una URL.

En el caso de la extensión para ROBOBO, la url es la siguiente:

<https://mytechia.github.io/robobo-scratch-extension/robobo-scratch-extension.js>

Para cargar la extensión en ScratchX debemos navegar a la página <http://scratchx.org/#home> en la que nos aparecerán dos opciones “open extension file” y “open extensión url”. Haciendo clic en la segunda nos aparecerá un cuadro de texto donde introduciremos la url mencionada anteriormente. Esto nos creará un proyecto de Scratch vacío en el que tendremos a nuestra disposición bajo la categoría de “more blocks” los bloques propios de la extensión de ROBOBO, así como los bloques clásicos de Scratch.

2 BLOQUES DE LA EXTENSIÓN

La extensión está formada por bloques de diferente tipos de bloques con diversas funciones, estos bloques se pueden separar en varias categorías en función de su cometido (conexión, sensorización, movimiento...) o su forma de funcionamiento (command blocks, reporter blocks y hat blocks).

A continuación se elabora la explicación de las dos clasificaciones.

2.1 Tipos de bloque

Los bloques de la extensión se separan en tres categorías dependiendo del comportamiento de los mismos

2.1.1 Command Blocks (CB)

Los "Command blocks" son el tipo de bloque más simple, ya que unicamente ejecutan el comando indicado, sin devolver ninguna clase de dato.

2.1.2 Reporter Blocks (RB)

Los "Reporter Blocks" son bloques que devuelven un valor al ser ejecutados.

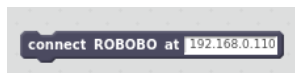
2.1.3 Hat Blocks (HB)

Los "Hat blocks" son bloques que comienzan la ejecución de una subrutina al darse cierto evento.

2.2 Bloques de gestión de conexión

Debido a que el programa de Scratch se ejecuta de manera remota en un ordenador, es necesario abrir una conexión entre el Robobo y el ordenador. Para este cometido existen los bloques de conexión, que permiten abrir el canal de forma simple.

2.2.1 Bloque de conexión (CB)



El bloque de conexión establece la conexión con el Robobo. Al iniciar la aplicación en el robot, se indicará al usuario la dirección IP del mismo, que debe ser introducida en este bloque.

Atención: Tanto el dispositivo donde se este ejecutando Scratch como el Robobo deben estar en la misma red.

Atención: La ejecución de este bloque no basta para comenzar a usar el Robobo, es necesario ejecutar posteriormente el bloque de autenticación.

2.2.2 Bloque de autenticación (CB)



El bloque de autenticación permite garantizar la seguridad de la conexión mediante el uso de una contraseña previamente definida en la aplicación del Robobo. Esta contraseña será introducida en este bloque, que debe ser ejecutado después del bloque de conexión.

Si la contraseña es la correcta, se comenzará a recibir datos de los sensores del Robobo y ejecutará los comandos que se le indiquen.

2.2.3 Bloque de desconexión (CB)



El bloque de desconexión cierra la conexión con el robot.

Si se desea volver a conectar es necesario repetir los pasos de conexión y autenticación. Bloque de parada

El bloque de parada permite cancelar todos los comandos de movimiento que se estén ejecutando en ese instante.

2.3 Bloques de Movimiento

Para controlar los movimientos del Rob, la extensión proporciona una serie de bloques centrados en el movimiento.

2.3.1 Bloque de movimiento de ruedas simultaneas (CB)

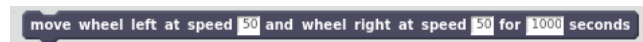


Este bloque permite mover las ruedas individualmente, o simultaneamente.

Recibe cuatro argumentos:

- la o las ruedas que se moverán
- la cantidad de movimiento
- el tipo de movimiento (segundos, grados o centímetros)
- la velocidad de movimiento (de -127 a 127, siendo las velocidades negativas marcha atrás)

2.3.2 Bloque de movimiento de ruedas independientes (CB)

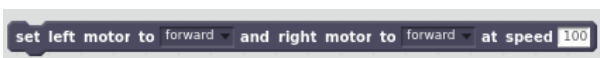


Este bloque permite mover las ruedas de manera independiente.

Recibe tres argumentos:

- velocidad de la rueda izquierda
- velocidad de la rueda derecha
- tiempo de ejecución

2.3.3 Bloque de encendido de motores (CB)



Este bloque enciende los motores por un tiempo indefinido, hasta que se ejecute el bloque de parada o un nuevo comando de movimiento.

Recibe tres argumentos:

- sentido de la rueda izquierda
- sentido de la rueda derecha
- velocidad

2.3.4 Bloque de movimiento de pan absoluto (CB)

Este bloque permite mover el “Pan” del Robobo a una posición absoluta.



Recibe dos argumentos:

- la posición a la que debe moverse el pan
- la velocidad de movimiento

2.3.5 Bloque de movimiento de pan relativo (CB)



Este bloque permite mover el “Pan” del Robobo a una posición relativa.

Recibe dos argumentos:

- la cantidad de grados que debe moverse el pan con respecto a la posición actual
- la velocidad de movimiento

2.3.6 Bloque de movimiento de tilt absoluto (CB)

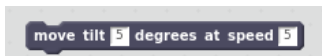


Este bloque permite mover el “Tilt” del Robobo a una posición absoluta.

Recibe dos argumentos:

- la posición a la que debe moverse el tilt
- la velocidad de movimiento

2.3.7 Bloque de movimiento de tilt relativo (CB)



Este bloque permite mover el “Tilt” del Robobo a una posición relativa.

Recibe dos argumentos:

- la cantidad de grados que debe moverse el tilt con respecto a la posición actual
- la velocidad de movimiento

2.4 Bloques de interacción

El objetivo de los bloques de interacción es proporcionar al usuario una manera de que el robot pueda interactuar de diferentes formas con humanos

2.4.1 Bloque de habla (CB)



El bloque de habla permite sintetizar voz al Robobo. El bloque recibe como argumento la frase que se desee pronunciar.

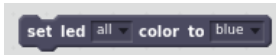
Atención: Actualmente solo el Español está soportado por la extensión.

2.4.2 Bloque de cambio de emociones (CB)

Este bloque permite cambiar la emoción mostrada en la pantalla del Robobo.

Recibe como argumento la emoción que se quiera mostrar.

2.4.3 Bloque de color de leds (CB)

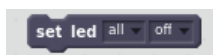


Este bloque permite cambiar de color los leds del Rob .

Recibe dos argumentos:

- el led a cambiar de color
- el nuevo color

2.4.4 Bloque de encendido de leds (CB)

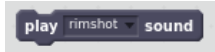


Este bloque permite encender o apagar los leds.

Recibe dos argumentos:

- el led a cambiar de estado
- el nuevo estado

2.4.5 Bloque de reproduccion de sonidos (CB)



Este bloque permite reproducir una serie de sonidos predefinidos.

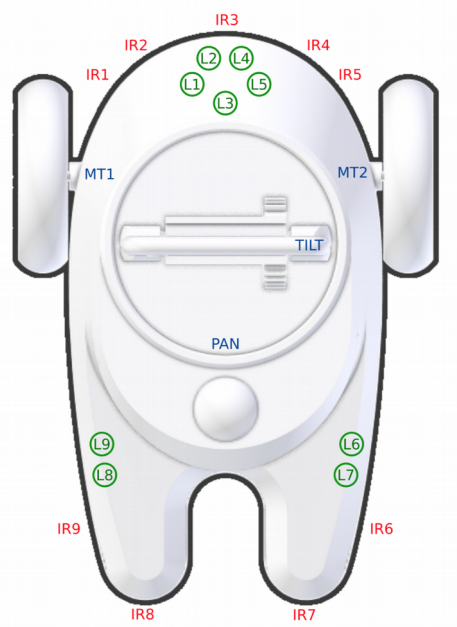
Como argumento recibe el sonido a reproducir.

2.5 Bloques de sensorización

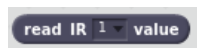
Los bloques de sensorización permiten al usuario acceder a los diferentes sensores con los que cuenta el Robobo. Pueden ser divididos en dos clases, los de sensorización del Rob (de la parte “física” del robot), y los del Obo (el smartphone).

2.5.1 Sensorización del ROB

El Rob cuenta con varios sensores a los que se puede acceder mediante los bloques de sensorización. A continuación se puede ver la disposición de los sensores en el chasis.



2.5.1.1 Bloque de lectura de Infrarrojos (RB)



Este bloque permite acceder a los valores de los sensores de infrarrojos del Rob.

Recibe como argumento el numero de sensor que se desea leer.

Devuelve el valor de la lectura del sensor (a mayor valor, menor distancia)

2.5.1.2 Bloque de lectura de batería de ROB (RB)

read ROB battery level

Este bloque devuelve el porcentaje de carga de la batería del Rob.

2.5.1.3 Bloque de lectura de obstaculos (RB)

read obstacle

Este bloque devuelve "true" si se ha detectado un obstaculo en cualquiera de los sensores del Rob.

2.5.1.4 Bloque de lectura de caidas. (RB)

read fall at

Este bloque permite anticipar las caidas cercanas en cada uno de los sensores de infrarojos del Rob.

2.5.1.5 Bloque de lectura de gaps. (RB)

read gap at

Este bloque permite anticipar las caidas lejanas en cada uno de los sensores de infrarojos del Rob.

2.5.1.6 Cambio de Infrarrojos (HB)

when ir changed

Este bloque se ejecuta cuando el sensor de infrarrojos indicado tiene un cambio sustancial con respecto a la última medida.

2.5.1.7 Batería del ROB baja (HB)

when ROB battery level is low

Este bloque se ejecuta siempre que la batería del Rob este cerca de agotarse.

2.5.1.8 Caída detectada (HB)



El bloque se ejecuta cuando se detecta una caída en el sensor de infrarojos indicado.

2.5.1.9 Gap detectado (HB)



El bloque se ejecuta cuando se detecta un gap en el sensor de infrarojos indicado.

2.5.2 Obstaculo detectado (HB)



El bloque se dispara cuando se detecta un obstaculo en cualquiera de los sensores del rob.

2.5.3 Sensorización del OBO

Además de los sensores propios del robot es posible acceder a los sensores propios del smartphone, a continuación se enumeran los bloques que lo permiten

2.5.3.1 Bloque de lectura de batería de OBO (RB)



Este bloque devuelve el porcentaje de carga de la batería del smartphone.

2.5.3.2 Bloque de lectura de color (RB)

Este bloque devuelve el ultimo color detectado por el detector de colores.

2.5.3.3 Bloque de lectura de distancia facial (RB)



Este bloque devuelve la distancia a la última cara detectada.

2.5.3.4 Bloque de lectura de angulo de flings (RB)

Este bloque devuelve el valor del ángulo del último fling detectado en la pantalla en grados.

read fling angle

Comienza a contar desde el centro del lado derecho de la pantalla en sentido antihorario.

2.5.3.5 Bloque de lectura de posición de caras (RB)

read face position at axis

Este bloque devuelve la posición de la última cara en el eje que se le indique.

La posición es medida en pixeles y el origen de coordenadas se situa en la esquina superior izquierda de la imagen capturada.

2.5.3.6 Bloque de lectura de posición de taps. (RB)

read tap position at axis

Este bloque devuelve la posición del último toque detectado en el eje que se le indique.

La posición es medida en pixeles y el origen de coordenadas se situa en la esquina superior izquierda de la pantalla.

2.5.3.7 Bloque de lectura de orientación. (RB)

read orientation at axis

Este bloque devuelve la última orientación medida por el giroscopio del smartphone en cualquiera de los tres ejes del espacio.

La orientación es medida en grados

2.5.3.8 Bloque de lectura de aceleración. (RB)

read acceleration at axis

Este bloque devuelve la última orientación medida por el acelerómetro del smartphone en cualquiera de los tres ejes del espacio.

La aceleración es medida en metros/segundo²

2.5.3.9 Bloque de lectura de brillo ambiente. (RB)

A Scratch 'read brightness' block, which is a light gray rectangle with a dark gray rounded rectangle in the center containing the text 'read brightness' in white.

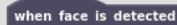
Este bloque permite leer el nivel de luminosidad ambiental.

2.5.3.10 Color detectado (HB)

A Scratch 'when color is detected' block, which is a light gray rectangle with a dark gray rounded rectangle in the center containing the text 'when color is detected' in white.


Este bloque se ejecuta cuando un nuevo color es detectado.

2.5.3.11 Cara detectada (HB)

A Scratch 'when face is detected' block, which is a light gray rectangle with a dark gray rounded rectangle in the center containing the text 'when face is detected' in white.

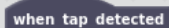
Este bloque se ejecuta cuando se detecta una cara.

2.5.3.12 Batería del OBO baja (HB)

A Scratch 'when ROB battery level is low' block, which is a light gray rectangle with a dark gray rounded rectangle in the center containing the text 'when ROB battery level is low' in white.

Este bloque se ejecuta siempre que la batería del smarphone este cerca de agotarse.

2.5.3.13 Tap detectado (HB)

A Scratch 'when tap detected' block, which is a light gray rectangle with a dark gray rounded rectangle in the center containing the text 'when tap detected' in white.

Este bloque se ejecuta cuando se detecta un tap en la pantalla del Robobo

2.5.3.14 Fling detectado (HB)

A Scratch 'when fling detected' block, which is a light gray rectangle with a dark gray rounded rectangle in the center containing the text 'when fling detected' in white.

Este bloque se ejecuta cuando se detecta un fling en la pantalla del Robobo

2.5.3.15 Aceleración detectada (HB)

A Scratch 'when acceleration detected' block, which is a light gray rectangle with a dark gray rounded rectangle in the center containing the text 'when acceleration detected' in white.

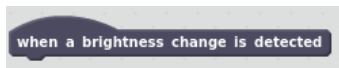
Este bloque es ejecutado cuando se detecta una aceleración brusca

2.5.3.16 Clap detectado (HB)



Este bloque se dispara al detectar una palmada.

2.5.3.17 Cambio de brillo detectado (HB)



El bloque se ejecuta cuando se detecta un cambio sustancial con respecto a la última medición de luminosidad.