



EXTENSION SCRATCH PARA ROBOBO

Manual de usuario

13 de enero de 2017

1	Carga de la extensión.....	3
2	Bloques de la extensión.....	3
2.1	Bloques para gestión de conexión	3
2.2	Bloques de Movimiento	4
2.3	Bloques de interacción	6
2.4	Bloques de sensorización.....	8

1 CARGA DE LA EXTENSIÓN

La extensión del ROBOBO está diseñada para su uso con la plataforma ScratchX (<http://scratchx.org>), que permite la integración de extensiones “experimentales” con el entorno Scratch estándar.

ScratchX requiere que las nuevas extensiones se carguen mediante una URL que enlace al fichero de extensión. La extensión para el robot ROBOBO se puede cargar utilizando la siguiente URL:

<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>. Allí aparecerán dos opciones: “**Open extension file**” y “**Open extensión URL**”. Haciendo clic en la segunda aparecerá un cuadro de texto donde introduciremos la URL mencionada anteriormente. Esto creará un nuevo 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, además de, como no, los bloques habituales del lenguaje Scratch.

2 BLOQUES DE LA EXTENSIÓN

La extensión añade un nuevo conjunto de bloques Scratch que permiten crear programas Scratchs que controlen el comportamiento del robot Robobo, moviéndolo físicamente y accediendo a numerosas de sus capacidades de sensorización, actuación, movimiento y procesamiento.

En esta sección se proporciona una breve explicación de cada uno de los bloques soportados. Para ello se han dividido en diferentes grupos de bloques según su función: conexión de la extensión, movimiento del robot, interacción y sensorización.

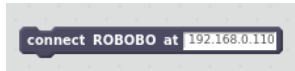
Adicionalmente, los bloques pueden ser diferentes tipos según su naturaleza de su comportamiento:

- **Command Blocks:** Son el tipo de bloque más simple. Únicamente ejecutan el comando indicado, sin devolver ningún resultado. Se indican en el documento con la abreviatura (CB).
- **Reporter Blocks:** Son bloques que devuelven un resultado después de su ejecución. Se indican en el documento con la abreviatura (RB).
- **Hat Blocks:** Son bloques que se ejecutan cuando se cumple un determinado evento. Permite la ejecución paralela y asíncrona de código. Se indican en el documento con la abreviatura (HB).

2.1 Bloques para gestión de conexión

La extensión de Scratch para Robobo necesita conectarse remotamente al robot, a través de la red WiFi, para poder ejecutar los diferentes comportamientos programados en un programa Scratch. Para este cometido existen los bloques de conexión, que permiten abrir el canal de comunicaciones fácilmente.

2.1.1 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 esté 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.

2.1.2 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. Es necesario llamar a este bloque después de ejecutar el bloque de conexión, y antes de ejecutar ningún otro bloque de funcionalidad.

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

2.1.3 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.

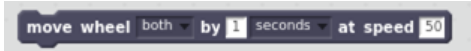
2.1.3.1 Bloque de parada

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

2.2 Bloques de Movimiento

La extensión proporciona una serie de bloques centrados en el movimiento para controlar los movimientos de la base robótica ROB.

2.2.1 Bloque de movimiento de ruedas simultaneas (CB)



Este bloque permite mover las ruedas individual o simultáneamente.

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)

2.2.2 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

2.2.3 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



2.2.4 Bloque de movimiento de pan absoluto (CB)

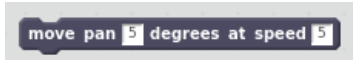
Este bloque permite mover el "Pan" del Robobo a una posición absoluta.

Recibe dos argumentos:

- la posición a la que debe moverse el pan

- la velocidad de movimiento

2.2.5 Bloque de movimiento de pan relativo (CB)



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

Recibe dos argumentos:

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

2.2.6 Bloque de movimiento de tilt absoluto (CB)

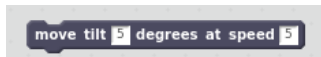


Este bloque permite mover el “Tilt” del Robobo a una posición absoluta.

Recibe dos argumentos:

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

2.2.7 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

2.3 Bloques de interacción

El objetivo de los bloques de interacción es proporcionar al usuario la posibilidad hacer programas Scratch que permite al robot interactuar con personas.

2.3.1 Bloque de habla (CB)



El bloque de habla permite hacer que el robot sintetice voz. Recibe como argumento la frase que se desee pronunciar.

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

2.3.2 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.

2.3.3 Bloque de color de leds (CB)

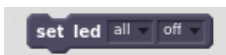


Este bloque permite cambiar de color los leds del ROB .

Recibe dos argumentos:

- el led a cambiar de color
- el nuevo color

2.3.4 Bloque de encendido de leds (CB)

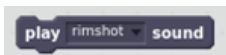


Este bloque permite encender o apagar los leds.

Recibe dos argumentos:

- el led a cambiar de estado
- el nuevo estado

2.3.5 Bloque de reproducción de sonidos (CB)



Este bloque permite reproducir una serie de sonidos predefinidos.

Como argumento recibe el sonido a reproducir.

2.4 Bloques de sensorización

Los bloques de sensorización permiten al usuario crear programas Scratch que utilicen los diferentes sensores con los que cuenta el Robobo. Estos sensores se dividen en dos clases, los de sensores del ROB (la base móvil robotizada), y los del OBO (el smartphone).

2.4.1 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.



2.4.1.1 Bloque de lectura de Infrarrojos (RB)

read IR 1 value

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.

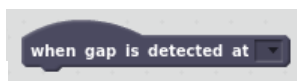
Devuelve el valor de la lectura del sensor (a mayor valor, menor distancia)

2.4.1.2 Bloque de lectura de batería de ROB (RB)

read ROB battery level

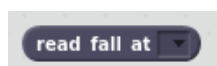
Este bloque devuelve el porcentaje de carga de la batería del Rob.

2.4.1.3 Bloque de lectura de obstáculos (RB)



Este bloque devuelve “true” si se ha detectado un obstáculo en cualquiera de los sensores del ROB.

2.4.1.4 Bloque de lectura de caídas. (RB)



Este bloque permite anticipar las caídas cercanas en cada uno de los sensores de infrarrojos del ROB.

2.4.1.5 Bloque de lectura de gaps. (RB)



Este bloque permite anticipar las caídas lejanas en cada uno de los sensores de infrarrojos del ROB.

2.4.1.6 Cambio de Infrarrojos (HB)



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

2.4.1.7 Batería del ROB baja (HB)



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

2.4.1.8 Caída detectada (HB)



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

2.4.1.9 Gap detectado (HB)

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

2.4.2 Obstáculo detectado (HB)



El bloque se dispara cuando se detecta un obstáculo en cualquiera de los sensores del ROB.

2.4.3 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

2.4.3.1 Bloque de lectura de batería de OBO (RB)



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

2.4.3.2 Bloque de lectura de color (RB)

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

2.4.3.3 Bloque de lectura de distancia facial (RB)



Este bloque devuelve la distancia a la última cara detectada.

2.4.3.4 Bloque de lectura de ángulo de flings (RB)



Este bloque devuelve el valor del ángulo del último fling detectado en la pantalla en grados. Comienza a contar desde el centro del lado derecho de la pantalla en sentido anti horario.

2.4.3.5 Bloque de lectura de posición de caras (RB)



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

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

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



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

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

2.4.3.7 Bloque de lectura de orientación. (RB)



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

2.4.3.8 Bloque de lectura de aceleración. (RB)



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

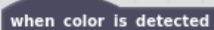
La aceleración es medida en metros/segundo²

2.4.3.9 Bloque de lectura de brillo ambiente. (RB)



Este bloque permite leer el nivel de luminosidad ambiental.

2.4.3.10 Color detectado (HB)



Este bloque se ejecuta cuando un nuevo color es detectado.

2.4.3.11 Cara detectada (HB)



Este bloque se ejecuta cuando se detecta una cara.

2.4.3.12 Batería del OBO baja (HB)



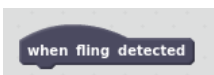
Este bloque se ejecuta siempre que la batería del smarphone este cerca de agotarse.

2.4.3.13 Tap detectado (HB)



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

2.4.3.14 Fling detectado (HB)



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

2.4.3.15 Aceleración detectada (HB)



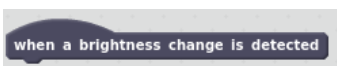
Este bloque es ejecutado cuando se detecta una aceleración brusca

2.4.3.16 Clap detectado (HB)



Este bloque se dispara al detectar una palmada.

2.4.3.17 Cambio de brillo detectado (HB)



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

