



ROBOBO SCRATCH EXTENSION

Manual de usuario

23 de diciembre de 2016

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.

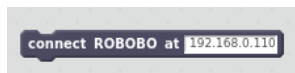
2 BLOQUES DE LA EXTENSIÓN

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

2.1 Command Blocks

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

2.1.1 Bloque de conexión

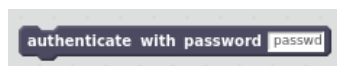


El bloque de conexión establece la conexión con el Robobo. Al iniciar la aplicación en el robot, se indicará al usuario la dirección IP del mismo, que debe ser introducida en este bloque.

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.

2.1.2 Bloque de autenticación



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.

2.1.3 Bloque de desconexión



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.

2.1.4 Bloque de habla



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.

2.1.5 Bloque de movimiento de ruedas simultaneas



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)

2.1.6 Bloque de movimiento de ruedas independientes



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

2.1.7 Bloque de encendido de motores



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

2.1.8 Bloque de movimiento de pan absoluto



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

2.1.9 Bloque de movimiento de pan relativo



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

2.1.10 Bloque de movimiento de tilt absoluto

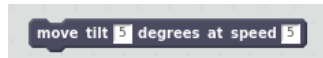


Este bloque permite mover el “Tilt” del Robobo a una posición absoluta.

Recibe dos argumentos:

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

2.1.11 Bloque de movimiento de tilt relativo



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

2.1.12 Bloque de cambio de emociones

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

Recibe como argumento la emoción que se quiera mostrar.

2.1.13 Bloque de color de leds

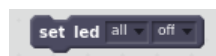


Este bloque permite cambiar de color los leds del Rob .

Recibe dos argumentos:

- el led a cambiar de color
- el nuevo color

2.1.14 Bloque de encendido de leds

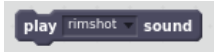


Este bloque permite encender o apagar los leds.

Recibe dos argumentos:

- el led a cambiar de estado
- el nuevo estado

2.1.15 Bloque de reproduccion de sonidos



Este bloque permite reproducir una serie de sonidos predefinidos.
Como argumento recibe el sonido a reproducir.

2.2 Reporter Blocks

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

2.2.1 Bloque de lectura de Infrarrojos



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)

2.2.2 Bloque de lectura de batería de ROB



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

2.2.3 Bloque de lectura de batería de OBO



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

2.2.4 Bloque de lectura de color

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

2.2.5 Bloque de lectura de distancia facial



Este bloque devuelve la distancia a la última cara detectada.

2.2.6 Bloque de lectura de obstaculos

read obstacle

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

2.2.7 Bloque de lectura de angulo de flings

read fling angle

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 antihorario.

2.2.8 Bloque de lectura de posición de caras

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.

2.2.9 Bloque de lectura de posición de taps.

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.

2.2.10 Bloque de lectura de orientación.

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

2.2.11 Bloque de lectura de aceleración.

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

2.2.12 Bloque de lectura de caídas.

read fall at 

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

2.2.13 Bloque de lectura de gaps.

read gap at 

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

2.2.14 Bloque de lectura de brillo ambiente.

read brightness

Este bloque permite leer el nivel de luminosidad ambiental.

2.3 Hat Blocks

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

2.3.1 Color detectado

when color is detected

Este bloque se ejecuta cuando un nuevo color es detectado.

2.3.2 Cara detectada



Este bloque se ejecuta cuando se detecta una cara.

2.3.3 Cambio de Infrarrojos



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

2.3.4 Batería del OBO baja



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

2.3.5 Batería del ROB baja



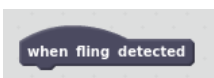
Este bloque se ejecuta siempre que la batería del Rob este cerca de agotarse.

2.3.6 Tap detectado



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

2.3.7 Fling detectado



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

2.3.8 Aceleración detectada



Este bloque es ejecutado cuando se detecta una aceleración brusca

2.3.9 Clap detectado



Este bloque se dispara al detectar una palmada.

2.3.10 Caída detectada



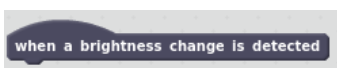
El bloque se ejecuta cuando se detecta una caída en el sensor de infrarojos indicado.

2.3.11 Gap detectado



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

2.3.12 Cambio de brillo detectado



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

2.3.13 Obstáculo detectado



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