

Programar el EV3 con Swift Playgrounds

Lección 2 – Mover en línea recta

Por Ahmad Sahar

Hola, soy Shah de nuevo. Para aquellos que no lo sepan, soy formador profesional especializado en software y hardware MacOS e iOS y también doy clases de desarrollo de aplicaciones iOS y LEGO® Mindstorms.

En la anterior lección creamos un sencillo programa para mostrar texto en la pantalla. En esta lección construiremos un robot y lo haremos mover hacia delante y hacia atrás.

Lo que necesitas:

- Un iPad con Swift Playgrounds instalado
- LEGO® Mindstorms EV3 Education (set no. 45544) o Home (set no. 31313)

Antes de empezar

Para esta lección tienes que construir bien el robot educacional con el set 45544 o DrivingBas3 con el set 31313.



Las instrucciones para el robot educacional están aquí: http://robotsquare.com/wp-content/uploads/2013/10/45544_educator.pdf

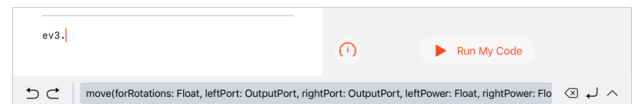
Las instrucciones para el DrivingBas3 están aquí: <http://ev3lessons.com/RobotDesigns/instructions/DrivingBas3.pdf>

Asegúrate de tener pilas nuevas en el robot y de que tu iPad esté bien cargado. Conecta tu iPad con el EV3 mediante Bluetooth.

Mover en línea recta

Puedes usar la plantilla que usamos en la anterior lección, o puedes descargar una copia nueva. Elimina todo el código de la plantilla y conecta con el EV3..

Pulsa ev3 en la lista de sugerencias. Aparece en la página. Pulsa el punto en la lista de sugerencias. Busca en la lista de sugerencias hasta encontrar esta que empieza por “move”:



Pulsalo para insertarlo en la página.

Esta es una función de Swift que es similar al bloque EV3-G de movimiento de tanque.

Tiene algunos parámetros:

forRotations: Número de rotaciones que se ejecutan

leftPort: El puerto al que se encuentra conectado el motor izquierdo. Puede ser .a, .b, .c o .d

rightPort: El puerto al que se encuentra conectado el motor derecho. Puede ser .a, .b, .c o .d

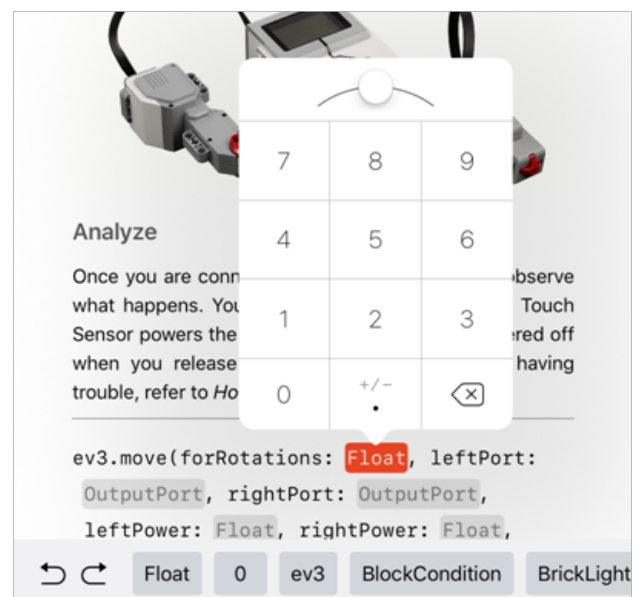
leftPower: El nivel de potencia especificado para leftPort.

rightPower: El nivel de potencia especificado para rightPort.

brakeAtEnd: Frenar o no frenar al final de “move”.

En general, para cambiar el valor de algún parámetro, pulsa el parámetro y tendrás bien un selector para elegir el valor o podrás escoger una opción de una lista.

Comenzaremos cambiando el parámetro de *forRotations*. Pulsa el parámetro representado por la palabra *Float*. Cuando aparezca el selector, elige 1. Esto fija el número de rotaciones a ejecutar en 1.



Pulsa el parámetro *OutputPort* de *leftPort*, y pulsa el punto en la lista de sugerencias.

Después aparecen los puertos de motor a, b, c y d en la lista de sugerencias. Elige b. Esto selecciona el puerto B del ladrillo programable como parámetro de *leftPort*.

Pulsa el parámetro *OutputPort* de *rightPort*, y pulsa en el punto de la lista de sugerencias. Elige c de la lista de sugerencias. Esto fija el puerto C del ladrillo para *rightPort*.

Pulsa el parámetro *Float* de *leftPower*. Aparece un seleccionador. Entra 50. Esto fija el nivel de potencia para el motor izquierdo en 50.

Pulsa el parámetro *Float* de *rightPower*. Aparece un seleccionador. Entra 50. Esto fija el nivel de potencia para el motor derecho en 50.

Finalmente, pulsa el parámetro *Bool* de *brakeAtEnd*. Pulsa true en la lista de sugerencias.

En este punto la función tendrá este aspecto:

```
ev3.move(forRotations: 1, leftPort: .b,
        rightPort: .c, leftPower: 50, rightPower:
        50, brakeAtEnd: true)
```

Ahora vamos a hacer que el robot espere un segundo con la función *wait*.

Pulsa en el espacio debajo del código que has creado y elige ev3 de la lista de sugerencias. Pulsa el punto y busca en la lista de sugerencias hasta encontrar *waitFor(seconds: Float)*. Seleccionalo.

```
ev3.move(forRotations: 1, leftPort: .b,
        rightPort: .c, leftPower: 50, rightPower:
        50, brakeAtEnd: true)
ev3.|
```

Esta función de Swift es similar al bloque esperar de EV3-G.

Tiene un solo parámetro, *seconds*, que representa el número de segundos a esperar.

Pulsa en el parámetro *Float* de *seconds*. Elige 1 en el seleccionador para que el programa espera 1 segundo.

La función debería tener este aspecto:

```
ev3.waitFor(seconds: 1)
```

Selecciona todo el código que has ingresado hasta el momento y copialo. Pulsa debajo del código existente y pegalo.

Para hacer que el robot se mueva hacia atrás, vamos a cambiar los valores de potencia de la segunda función *move* a -50 para ambos motores. Pulsa el parámetro *leftPower* de la segunda función y, cuando aparezca el seleccionador, busca la tecla que muestra +/- . Pulsalo y arrástralo hacia abajo. Verás que el valor se cambia a -50. haz lo mismo para el parámetro *rightPower* . El resultado debería ser el siguiente:

```
ev3.move(forRotations: 1, leftPort: .b,
        rightPort: .c, leftPower: 50, rightPower:
        50, brakeAtEnd: true)
ev3.waitFor(seconds: 1)
ev3.move(forRotations: 1, leftPort: .b,
        rightPort: .c, leftPower: -50,
        rightPower: -50, brakeAtEnd: true)
ev3.waitFor(seconds: 1)
```

Pulsa debajo del código que has escrito hasta ahora y vuelve a pegar.

Elimina la última línea de código, y obtendrás esto:

```
ev3.move(forRotations: 1, leftPort: .b,
        rightPort: .c, leftPower: 50, rightPower:
        50, brakeAtEnd: true)
ev3.waitFor(seconds: 1)
ev3.move(forRotations: 1, leftPort: .b,
        rightPort: .c, leftPower: -50,
        rightPower: -50, brakeAtEnd: true)
ev3.waitFor(seconds: 1)
ev3.move(forRotations: 1, leftPort: .b,
        rightPort: .c, leftPower: 50, rightPower:
        50, brakeAtEnd: true)
```

Ejecuta el programa. El robot debería avanzar una rotación, pausar un segundo, mover hacia atrás una rotación, pausar otro segundo y mover nuevamente una rotación hacia delante.

¡Buen trabajo! Hemos llegado al final de esta lección. En la siguiente lección haremos girar al robot..

Si quieres saber más acerca de mí y de mi trabajo, puedes visitar la página de mi empresa, <http://tomafuwi.tumblr.com>, mi página de Facebook <http://facebook.com/tomafuwi>, o seguirme en Twitter https://twitter.com/shah_apple

¡Hasta la próxima!
#