

Sphero - Abril 2017

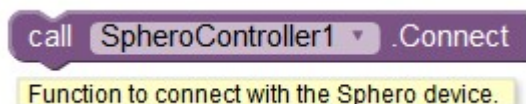


SPI&FM y Robótica Aplicada

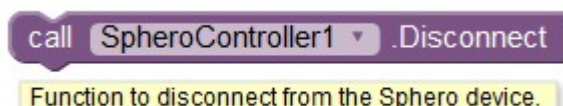
Edson Castro. José María Rodríguez.

Guía visual de la API *SpheroController*

1. Conexión y desconexión.

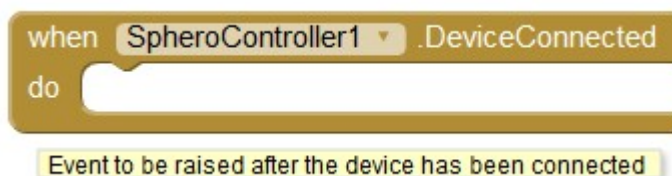


Bloque *Connect*: Permite a un dispositivo móvil establecer una conexión con un robot Sphero previamente vinculado al mismo.

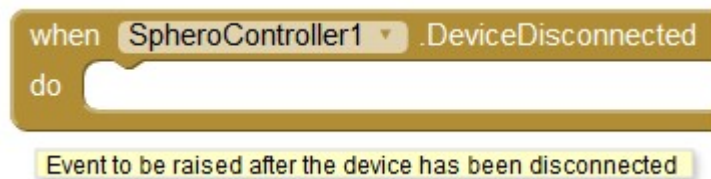


Bloque *Disconnect*: Cierra la conexión entre existente entre el robot Sphero y el dispositivo móvil.

2. Eventos de conexión y desconexión.

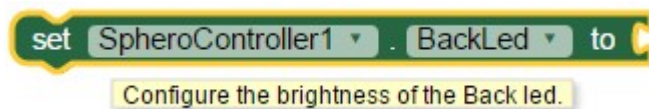


Bloque *DeviceConnected*: Permite la realización de un conjunto de acciones cuando se origine un evento de conexión de un robot *Sphero* a un dispositivo móvil.

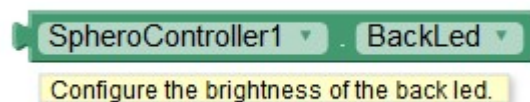


Bloque *DeviceDisconnected*: Permite la realización de un conjunto de acciones cuando se origine un evento de desconexión de un robot *Sphero* de un dispositivo móvil.

3. Luz de guía (trasera).



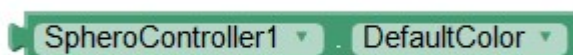
Bloque *BackLed*: Enciende la luz de guía de *Sphero*. El parámetro *intensity* es un entero entre 0 y 255 que indica la intensidad de encendido.



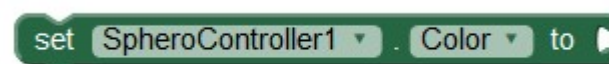
4. Color por omisión.



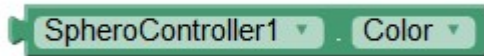
El bloque *DefaultColor* configura el color por omisión, que es el color que se verá cuando el robot esté emparejado pero no conectado. Cuando se utilice el bloque, se configura el color de emparejamiento. Aunque se desconecte el dispositivo de la aplicación, cuando se vuelva a conectar se verá el color en el emparejamiento, pero si se apaga el robot *Sphero* se pierde el color.



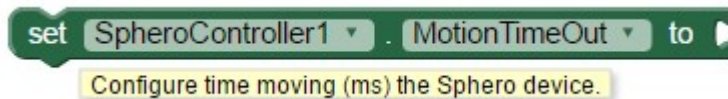
5. Color.



El bloque *Color* configura el color del LED RGB del robot cuando éste se encuentre en el estado conectado.



6. Límite de tiempo de movimiento (*Motion Timeout*).



Configure time moving (ms) the Sphero device.

Bloque *MotionTimeout*: Configura el tiempo que el robot *Sphero* estará en movimiento. El parámetro *time* indica el tiempo en milisegundos durante el que se moverá *Sphero*. Una vez que se usa el bloque *MotionTimeout*, *Sphero* memoriza el tiempo límite de movimiento con que se ha configurado. Para que el robot se mueva por tiempo indefinido se debe indicar el valor 0. El tiempo límite de movimiento para Sphero por omisión es de 2000 milisegundos.



Configure time moving (ms) the Sphero device.

7. Orientación.

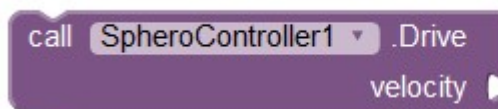


angle

Function to Rotate the Base Angle, value from 0 to 360.

Bloque *RotateBaseAngle*: Sitúa la luz de guía en el ángulo indicado (parámetro *angle*) para ajustar el sistema de referencias del robot. Cuando Sphero se mueve hacia delante (ángulo de cero grados), se desplaza en sentido opuesto al señalado por la luz de guía.

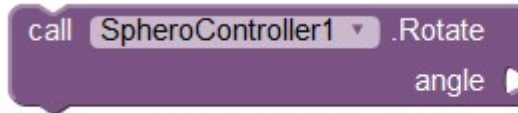
8. Movimiento.



velocity

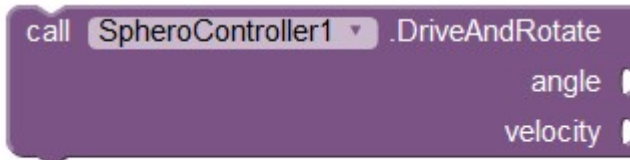
Function to drive the Sphero device, velocity value from 0.0 to 1.0.

Bloque *Drive*: Inicia el movimiento del robot Sphero hacia delante (ángulo de cero grados) con la velocidad indicada mediante el parámetro *velocity*.



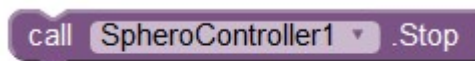
Function to rotate the Sphero device, angle value from 0.0 to 360.0

Bloque *Rotate*: Efectúa la rotación del robot *Sphero* en el ángulo indicado mediante el parámetro *angle*.



Function to move the Sphero device, angle value from 0.0 to 360.0, velocity value from 0.0 to 1.0.

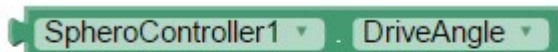
Bloque *DriveAndRotate*: Inicia el movimiento del robot *Sphero* en la dirección y la velocidad indicadas mediante los parámetro *angle* y *velocity* respectivamente.



Function to stop the Sphero device.

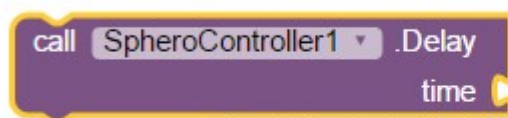
Bloque *Stop*: Detiene el movimiento del robot *Sphero*.

9. Accesores para obtener ángulo y velocidad.



Son dos bloques individuales que proporcionan los valores de ángulo y velocidad del último movimiento realizado. Ambos bloques están relacionados con el bloque *DriveAndRotate*.

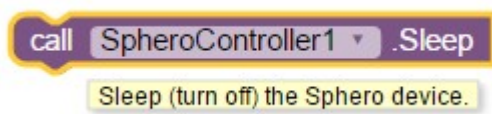
10. Pausa.



Dealy time (ms) at the Sphero device, max time 60000 ms.

Bloque *Delay*: Se indica el tiempo de pausa en milisegundos mediante el parámetro *time*. Se permite un tiempo máximo 60.000 milisegundos (un minuto) para evitar que la pausa sea demasiado grande.

11. Apagado.

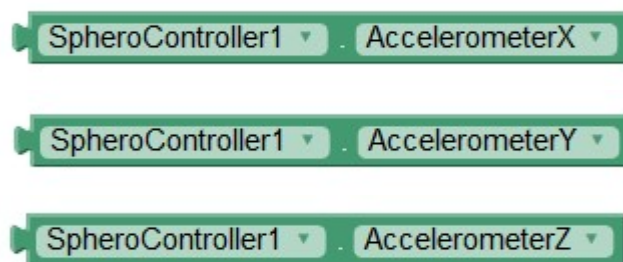


Bloque *Sleep*: Apaga el robot *Sphero*. Se recomienda al programador que cuando apague el robot también detenga la ejecución de la aplicación. El uso de este bloque se deja bajo la responsabilidad del programador.

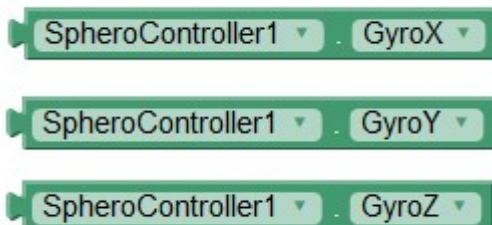


Bloque *DisconnectAndSleep*: Apaga el robot *Sphero* y además realiza la desconexión del vínculo *Bluetooth* con la aplicación. Ello permite activar nuevamente al robot para que se pueda volver a conectar. La ejecución de este bloque dispara el evento *DeviceDisconnected* asociado al componente no visible *SpheroController*.

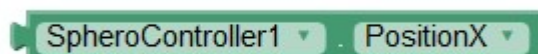
12. Sensores internos.



Bloques de información de acelerómetro: Proporcionan los valores medidos por el acelerómetro en cada uno de los tres ejes.



Bloques de información de giroscopio: Proporcionan los valores medidos por el giroscopio en cada uno de los tres ejes.



SpheroController1 . PositionY

Bloques de información de posición: Proporcionan los valores de posición para los ejes X e Y expresados en centímetros.

SpheroController1 . VelocityX

SpheroController1 . VelocityY

Bloques de información de velocidad: Proporcionan los valores de velocidad para los ejes X e Y expresados en milímetros por segundo.

13. Configuración de parámetros para detección de colisiones.

call SpheroController1 .ConfigureCollision
Turn on the collision configuration, with default parameters, can be modify with the properties

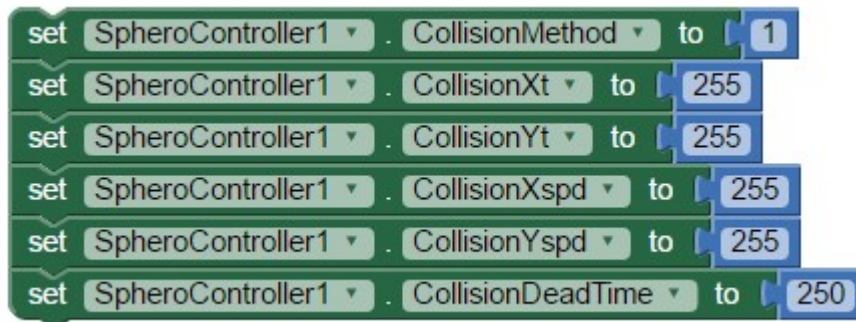
El bloque *ConfigureCollision* permite modificar los umbrales de detección de colisiones para obtener diferentes resultados en la práctica. Se han establecido valores por defecto en la aplicación para poder usar la función con parámetros preasignados. Sus valores son: *Meth*: 1. *Xt*, *Yt*: 100. *Xspd*, *Yspd*: 125. *Dead*: 100.

param	description
Meth	Detection method type to use. Supported methods are 01h, 02h, and 03h (see the collision detection document for details). Use 00h to completely disable this service.
Xt, Yt	An 8-bit settable threshold for the X (left/right) and Y (front/back) axes of Sphero. A value of 00h disables the contribution of that axis.
Xspd, Yspd	An 8-bit settable speed value for the X and Y axes. This setting is ranged by the speed, then added to Xt, Yt to generate the final threshold value.
Dead	An 8-bit post-collision dead time to prevent retriggering; specified in 10ms increments.

Estos valores pueden ser leídos mediante de las propiedades siguientes, tanto si se usan por defecto o si se llegan a modificar.

SpheroController1 . CollisionMethod
SpheroController1 . CollisionXt
SpheroController1 . CollisionYt
SpheroController1 . CollisionXspd
SpheroController1 . CollisionYspd
SpheroController1 . CollisionDeadTime

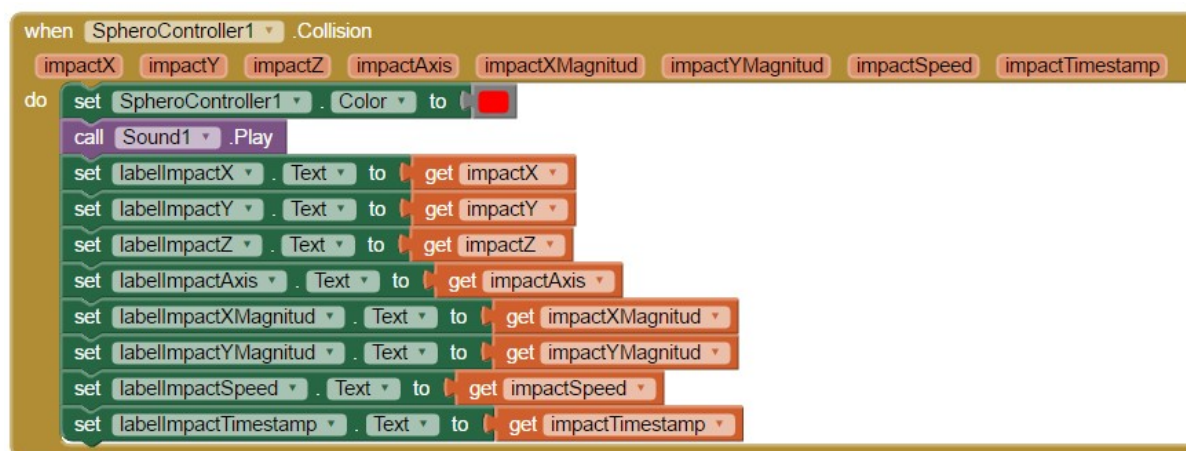
Para modificar los valores se usarán las propiedades descritas a continuación. Los valores máximos son 255 excepto para *DeadTime*, cuyo máximo es 250, y *Method*, que solo puede tomar valores entre comprendidos entre 0 y 3.



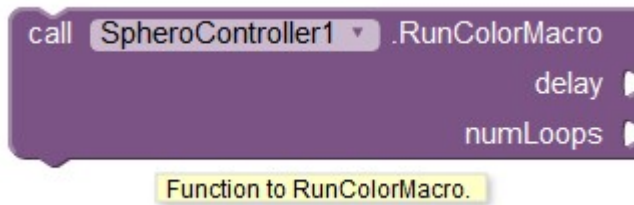
14. Evento de detección de colisión.

El bloque *collision* permite la realización de un conjunto de acciones asociado a la detección de una colisión y proporciona información sobre la intensidad y la zona de la última colisión detectada:

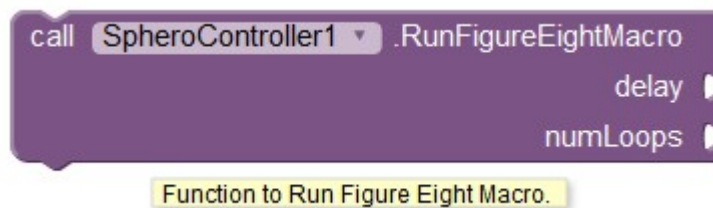
param	description
X, Y, Z	Impact components normalized as a signed 16-bit value. Use these to determine the direction of collision event. If you don't require this level of fidelity, the two Magnitude fields encapsulate the same data in pre-processed format.
Axis	This bitfield specifies which axes had their trigger thresholds exceeded to generate the event. Bit 0 (01h) signifies the X axis and bit 1 (02h) the Y axis.
xMagnitude	This is the power that crossed the programming threshold Xt + Xs.
yMagnitude	This is the power that crossed the programming threshold Yt + Ys.
Speed	The speed of Sphero when the impact was detected.
Timestamp	The millisecond timer value at the time of impact; refer to the documentation of CID 50h and 51h to make sense of this value.



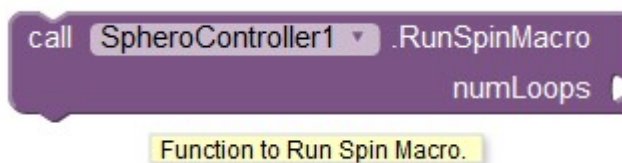
15. Ejecución de macros predefinidas.



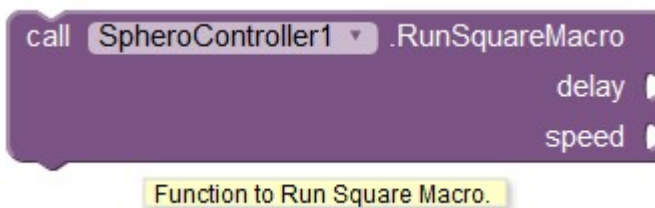
Bloque *RunColorMacro*: Controla al robot Sphero para que muestre toda la gama de colores disponibles. El cambio entre colores es gradual y muy vistoso.



Bloque *RunEightFigureMacro*: Controla al robot *Sphero* para que ejecute trayectorias con forma de la clásica figura de ocho.



Bloque *RunSpinMacro*: Activa la luz de guía de *Sphero* y controla al robot para que realice un movimiento de rotación sin desplazarse de su sitio original.



Bloque *RunSquareMacro*: Controla al robot *Sphero* para que ejecute trayectorias cuadradas desplazándose, parando y girando 90 grados cuatro veces por cada trayectoria.



Bloque *RunVibrateMacro*: Controla al robot *Sphero* para que mueva los motores hacia atrás y hacia delante muy rápidamente con el fin de simular un efecto de vibración.