

### 1.2.8. Movimiento de servos de rotación continua (Ref libro 1 pg 105)

Esta actividad va a introducirnos en el uso de motores con nuestra placa. Los primeros motores que vamos a utilizar son servos de rotación continua. La actividad pretende poner en marcha dos servo motores al pulsar un botón.

Componentes necesarios para la actividad:

- Placa ZUM BT, Arduino o compatible.
- Cable USB para conectar la placa al PC
- 2 sercomotores del ZUM KIT o similares
- 1 botón del ZUM KIT o similar.

El conexionado que vamos a utilizar es el siguiente:

- Boton a PIN 13
- Servomotor 1 a PIN 11
- Servomotor 2 a PIN 1

Es importante reseñar que los motores y servomotores normalmente funcionan con señales PWM y deben conectarse a los pines destinados a este uso. En BitBloq no es necesario y podemos conectarlo a cualquier PIN digital.

La configuración hardware que vamos a definir en BitBloq es la que se muestra en la Figura 1.2.8-1. Los nombres dados a los componentes son “servo\_continuo\_0”, “servo\_continuo\_1” y “botón\_0”.

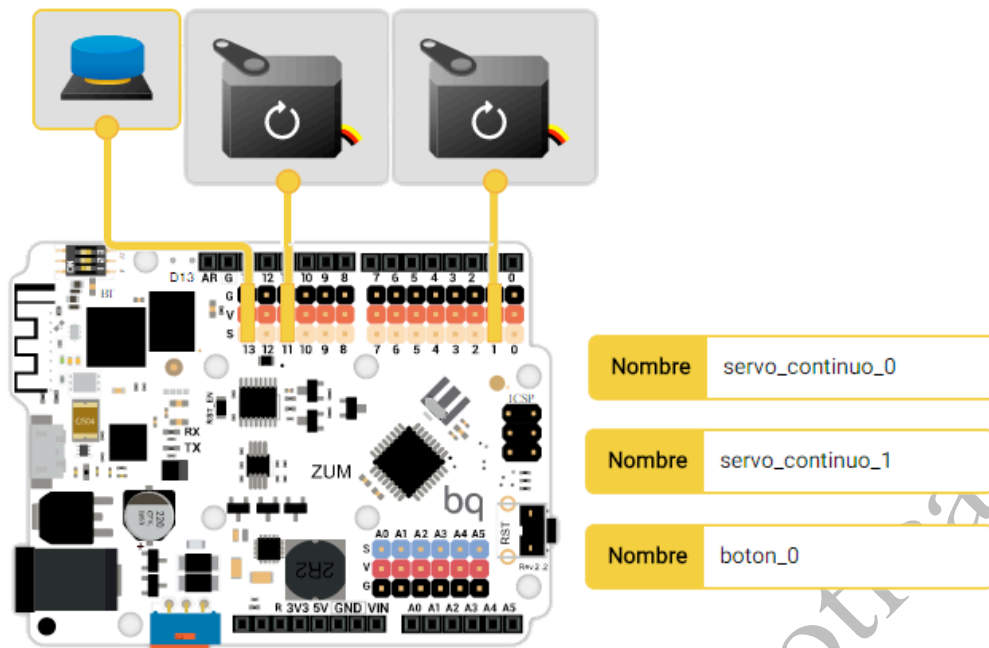


Figura 1.2.8-1. Conexión hardware BitBlok actividad de servomotores.

Para crear nuestro programa vamos a usar un nuevo tipo de estructura, la función. Las funciones ejecutan un fragmento de código que se usa muchas veces durante el programa. Necesitan ser declaradas y que se defina el código que van a ejecutar.

Un ejemplo es el que tenemos en esta actividad. Vamos a mover 2 motores que deben encenderse y apagarse, para mover cada motor necesitamos una instrucción. Para ahorrar tiempo y hacer más sencilla la programación usaremos una función que encienda los dos motores a la vez y otra que los pare.

La declaración de las funciones se realiza en la línea “Variables globales, funciones y clases”. Para crear la función vamos a la pestaña “Funciones” y arrastramos el bloque “Declarar función”. En este caso hemos llamado a una función Avanza y a otra Para. El código es el que se muestra en la Figura 1.2.8-2, como se puede ver tan solo contienen las instrucciones para encender o apagar los motores

## – Variables globales, funciones y clases

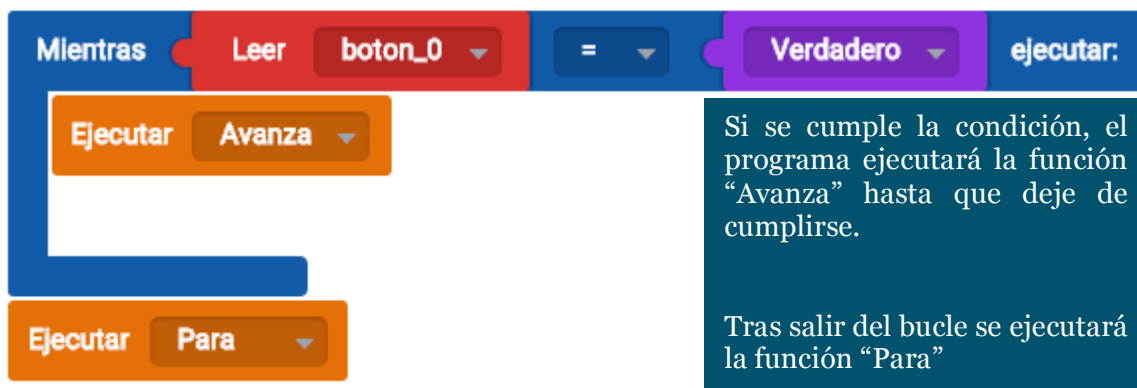


Figura 1.2.8-2. Declaración de funciones en BitBloq.

Vamos ahora al Bucle principal. Queremos que nuestros motores se muevan al mantener pulsado un botón y que se apaguen al soltarlo. Podríamos usar la estructura Si (If) como en actividades anteriores, pero vamos a introducir un nuevo tipo de bucle llamado Mientras (While en inglés). Este tipo de bucles ejecutan lo que hay en su interior hasta que deja de cumplirse la condición. A diferencia del If que comprueba la condición y ejecuta si se cumple siguiendo después con el resto de código, el bucle While no continua con el resto del código hasta que la condición no deja de cumplirse.

En la Figura 1.2.8-3, podemos ver en la parte superior como emplear las funciones y el bucle While para realizar nuestra actividad. En la figura vemos dos formas de resolver la actividad que son básicamente iguales, la segunda emplea una condición If y la primera un bucle While.

## – Bucle principal (Loop)



Si se cumple la condición, el programa ejecutará la función “Avanza” hasta que deje de cumplirse.

Tras salir del bucle se ejecutará la función “Para”

## – Bucle principal (Loop)



Figura 1.2.8-3. Código en BitBloq para mover dos servomotores con un botón.

El código en Arduino es:

```
/** Included libraries */
#include <Servo.h>

/** Global variables and function definition */
Servo servo_continuo_0;
Servo servo_continuo_1;
const int boton_0 = 13;
void Avanza() {
  servo_continuo_0.write(180);
  servo_continuo_1.write(0);
}
void Para() {
  servo_continuo_0.write(90);
```

## Libro de Actividades de Robótica Educativa

```
servo_continuo_1.write(90);  
}  
  
/**  Setup  **/  
void setup() {  
  servo_continuo_0.attach(11);  
  servo_continuo_1.attach(1);  
  pinMode(boton_0, INPUT);  
}  
  
/**  Loop  **/  
void loop() {  
  if (digitalRead(boton_0) == true) {  
    Avanza();  
  } else {  
    Para();  
  }  
}
```

[www.automaticayrobotic.com](http://www.automaticayrobotic.com)