

Obligatorio 2 - Servomotores y una pata.

IIE - Facultad de Ingeniería - Universidad de la República

Tallerine Biónico 2024

Este obligatorio busca implementar un código sencillo para el manejo de servomotores mediante Arduino IDE utilizando una placa Micro:bit. Además se dan las instrucciones para armar una pata del “bicho”, con el objetivo de ir enfrentándose a los problemas mecánicos y de código.

Para la comunicación entre la Micro:bit y los servomotores se hará uso de la placa **Kitronik 16 Servo Driver Board for the BBC Micro:bit** para la cual deberán descargar la librería **udriver_pca9685_fing** de la página del curso.

1. Servomotores

Un servomotor (también llamado servo) es un dispositivo basado en un motor de corriente continua que tiene la capacidad de ubicarse en cualquier posición dentro de su rango de operación, y mantenerse estable en dicha posición.

El servo que utilizaremos en el curso es el SG90 (figura 1). En el siguiente enlace se puede obtener datos técnicos del servo: <http://akizukidenshi.com/download/ds/towerpro/SG90.pdf>.



Figura 1: Servomotor SG90

La posición angular del piñón del motor se controla a través de una señal de modulación por ancho de pulso (PWM). No es necesario tener conocimientos sobre cómo generar la señal PWM ya que disponen en EVA junto con la letra del obligatorio una librería para Arduino IDE llamada **udriver_pca9685_fing**, que ya resuelve de forma práctica la comunicación entre la placa y el servo, de forma que se pueda fijar la posición angular del piñón, con solo proporcionar el ángulo deseado.

En la figura 2, se describe el “pinout” del servo SG90.

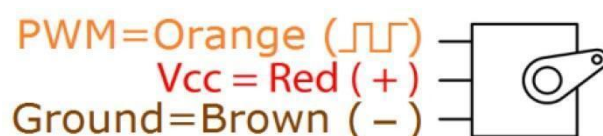


Figura 2: Pinout del servomotor SG90

El rango de operación de la posición angular del piñón es entre 0° y 180° (figura 3). Se recomienda utilizar ángulos entre 10° y 170° pues próximo a 0° y 180° el servo puede funcionar incorrectamente.

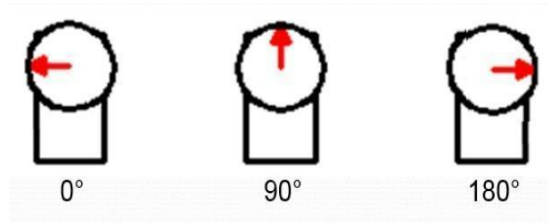


Figura 3: Posición del piñón en 0°, 90° y 180°

IMPORTANTE - No utilizar ángulos fuera del rango 0-180 ya que pueden forzar innecesariamente al servo y quemarlo. ¡¡¡Ser cuidadosos al realizar las conexiones ya que un error puede quemar la placa!!!!

2. Conexionado

La mayoría de los servos, así como los utilizados en el curso requieren una alimentación de 5V, por lo tanto no será posible conectarlos directamente a la Micro:bit. Recordar que solamente puede suministrar 3.3V con una corriente máxima de hasta 90 mA lo cual no es lo suficiente para mover un servo. Intentar consumir más energía de la posible a entregar, podría dañar el dispositivo.

Para la micro:bit, la forma más confiable de usar este tipo de servo es alimentar el servo a través de baterías, como se muestra en el esquema de la figura 4.

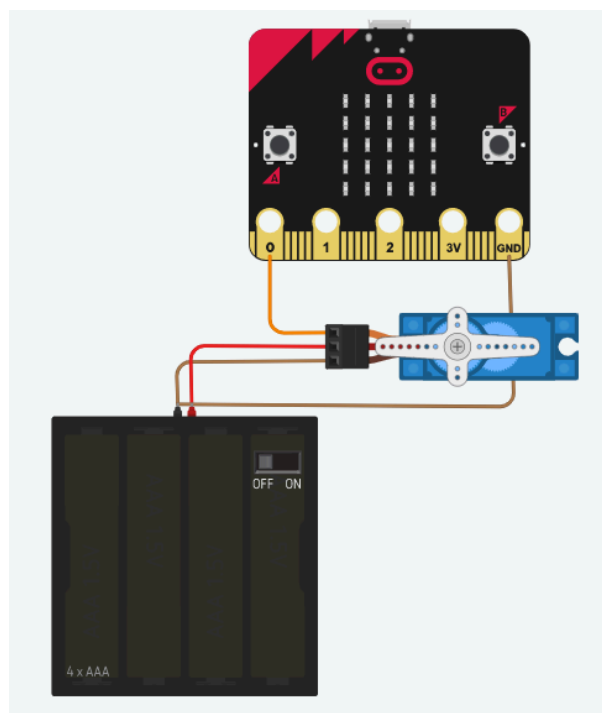


Figura 4: Conexionado entre el servo sg90 y la placa Microbit. Extraído de:

<https://support.microbit.org/support/solutions/articles/19000101864-using-a-servo-with-the-micro-bit>

En este curso para poder alimentar y controlar los servos se utilizará la placa Kitronik 16 Servo Driver Board, figura 5.



Figura 5: Conexión Micro Bit-Servo Driver

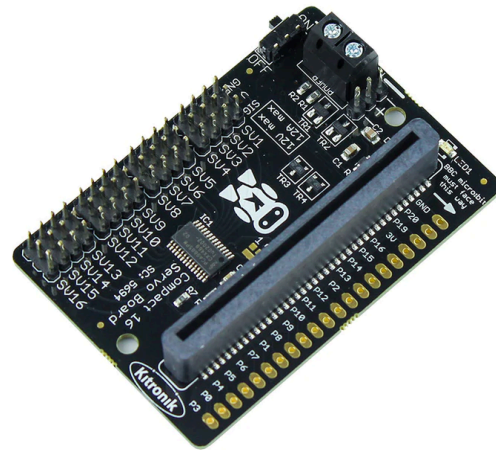


Figura 6: Kitronik 16 Servo Driver Board

Esta placa permite controlar y suministrar la corriente necesaria para poder mover hasta 16 servomotores en simultáneo sin tener que utilizar pines de la Micro:bit para generar las señales PWM. Esta placa posee un conector de borde "Edge Connector" que permite integrar la Micro:bit a la placa como se observa en la figura 5. La Micro:bit y la placa Kitronik se comunican a través del protocolo I2C (un protocolo de comunicación serial síncrono), por este medio la Micro:bit envía qué servomotor desea mover y a qué ángulo, encargándose la propia placa Kitronik de generar la señal PWM para moverlo.

Para conectar un servomotor a la placa Kitronik se deberá conectarlo en algunas de las 16 conexiones posibles (SV1, ..., SV16) conectando el cable de señal (naranja) del servomotor al pin SIG, el cable rojo V+ del servomotor al pin V y por último cable marrón GND al pin GND.

Para saber más detalles de la placa se puede acceder al datasheet en el siguiente link: <https://resources.kitronik.co.uk/pdf/5612-kitronik-i2c-servo-driver-board-microbit-datasheet.pdf>

Aclaración: Los pines 19 y 20 de la placa **Kitronik** son para la comunicación I2C. Tener en cuenta que estas salidas no se pueden usar con dispositivos que no utilicen esta comunicación como por ejemplo, para el manejo de LEDs externos. Para que los pines de la placa Kitronik y la microbit coincidan se debe insertar la microbit con la orientación indicada en la figura 5.

3. Librerías y funciones

Como se indicó anteriormente la posición angular del piñón se controla a través de una señal de pulso de ancho modulado (PWM). Junto con la letra del obligatorio se dispone de una librería predefinida llamada **udriver_pca9685_fing**, la cual provee de una manera sencilla manejar la posición angular mediante el uso de la placa para el manejo de servos.

Para utilizar una librería, debe estar agregada a Arduino IDE. Para ello, seleccionar *Programa > Incluir Librería > Añadir biblioteca.ZIP..* Una vez realizado esto, reiniciar Arduino IDE y verificar si se encuentra en el listado de librerías, *Programa > Incluir Librería.*

Para incluir la librería `udriver_pca9685_fing` en nuestro sketch y utilizar sus funciones se deben agregar las siguientes líneas en la “zona de declaración de variables”:

```
#include <udriver_pca9685_fing.h> // Para poder usar la librería.
using namespace UDriver_PCA9685;

const I2CAddress ADDR = 0x6A
PCA9685ServoController servo_ctrl(ADDR); // Crea una variable asociada al
// controlador llamada “servo_ctrl”
```

Una vez incluida creada la variable `servo_ctrl`, se pueden utilizar sus funciones, a continuación se detallarán las usadas en el obligatorio:

- `begin()` // Inicializa el módulo. Importante! Se debe inicializar en el setup.
- `move_servo(pin Pin, double angle_deg)` // Mueve el servo conectado en el pin Pin
// al ángulo `angle_deg`

Aclaración: los pines para mover los servos son SV1, SV2,... SV16.

El siguiente ejemplo muestra el sketch `servo_ini_microbit.ino` (disponible en EVA).

```
#include <udriver_pca9685_fing.h> // Para poder usar la librería.
using namespace UDriver_PCA9685;

///Servos
const I2CAddress ADDR = 0x6A;
PCA9685ServoController servo_ctrl(ADDR); // Crea una variable asociada al
// controlador llamada “servo_ctrl”

const pin_t PIN_SERVO = SV1; // Pin del servomotor (cable naranja)
int angulo = 90; // Ángulo inicial del piñón del servo

void setup()
{
  // Inicializo servos
  servo_ctrl.begin(); // Inicializa controlador de servos
  servo_ctrl.move_servo(PIN_SERVO, angulo); // Inicializa la posición del piñón
// del servo
}

void loop()
{
  // NO HACE NADA...
}
```

Nota: El texto a continuación de “///” son comentarios para aclarar lo que realiza el código. Este texto no es considerado como parte del programa que luego va a ser ejecutado en la Arduino. Es una muy buena práctica (y en el curso es obligación) utilizar estos comentarios para ir aclarando lo que se va haciendo.

4. Armado de una pata

En esta sección se describe el procedimiento para el armado de una pata, pero **antes de comenzar vea el ejercicio 5**, donde se dan los últimos detalles de cómo realizar el ensamble correctamente.

A los efectos de normalizar la nomenclatura a utilizar se detalla en la figura 7 cómo referirse a los servos de una pata completa.

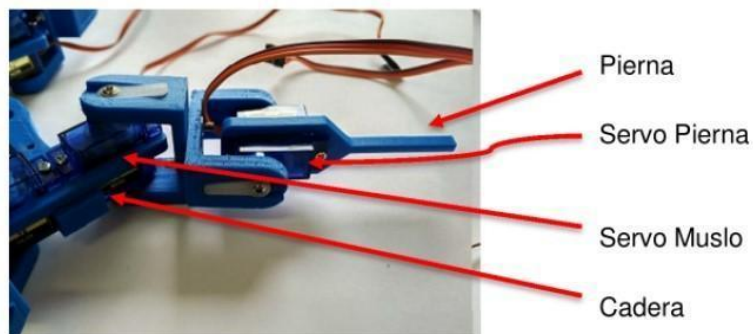


Figura 7: Nombres de los servos según la posición que ocupan.

A continuación se describen los pasos que se **deben** seguir para armar una pata:

A. Unión de articulaciones.

Para unir las articulaciones utilizar 2 tornillos con tuerca en sentidos opuestos (uno para cada lado) y en posición diagonal (figura 8). La cabeza del tornillo debe ir donde hay una forma de tuerca, del lado donde calza el piñón del servo (agujero grande). De igual forma, la tuerca debe colocarse en donde hay un redondel, opuesto al piñón del servo (agujero pequeñito) (figura 8). **Si no se respeta esto, el servo se tranca y se rompe.**



Figura 8: Unión de articulaciones.

B. Montaje del servo de la cadera.

Se monta el servo con el piñón hacia el lado externo de la cadera y se colocan los 2 tornillos como se muestra en la figura 9. Los tornillos a utilizar son los de tamaño medio.

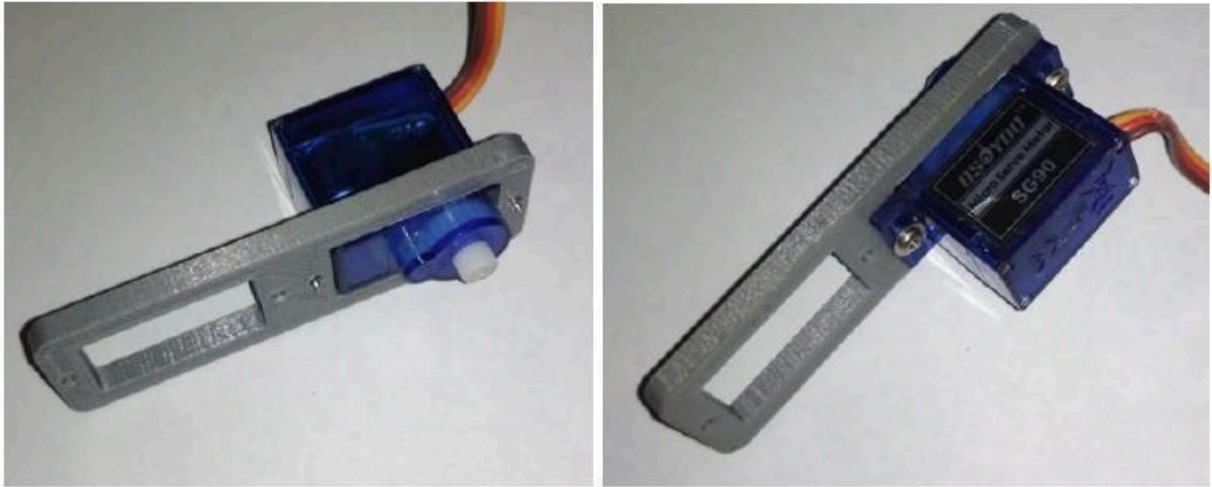


Figura 9: Montaje del Servo de la cadera.

C. Montaje del servo de la pierna.

El procedimiento es igual al caso de la cadera; se muestra en la (figura 10).

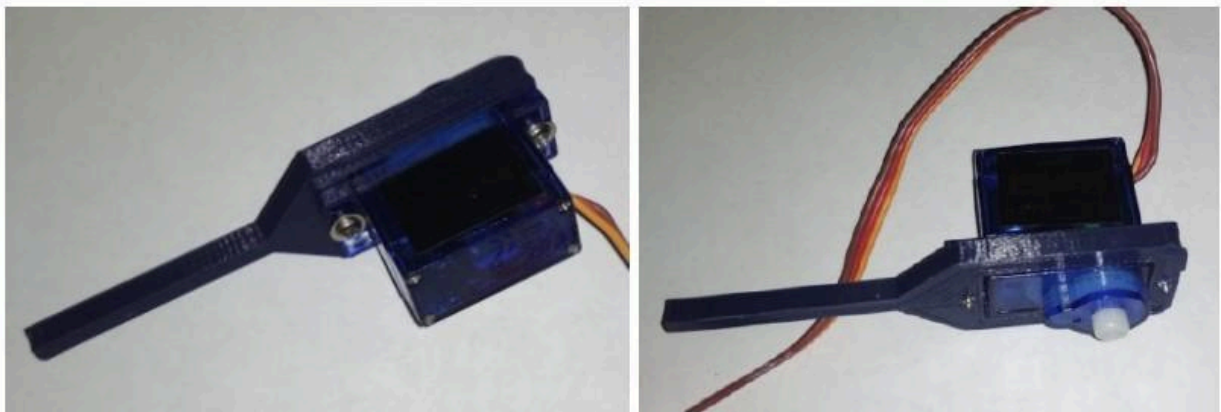


Figura 10: Montaje del Servo de la pierna.

D. Unir las partes.

- Montar el aspa blanca en la articulación (figura 11, izquierda).
- Abrir levemente la articulación con mucho cuidado de no partirla para colocar el servo en su posición, calzando el piñón con el aspa.
- Colocar el **tornillo pequeño** que une el aspa con el piñón del servo.
- En el lado opuesto al piñón colocar un **tornillo mediano** el cual va apenas apoyado sobre la carcasa del servo. Su función es minimizar el movimiento libre del servo respecto a la articulación.

- Una vez armada la pata, utilizando el cable suministrado, se debe hacer un lazo en cada servo para sostener sus cables (figura 6, derecha). Esto evita que el cable se quiebre justo en la carcasa del servo, es decir, **se rompa el servo**.

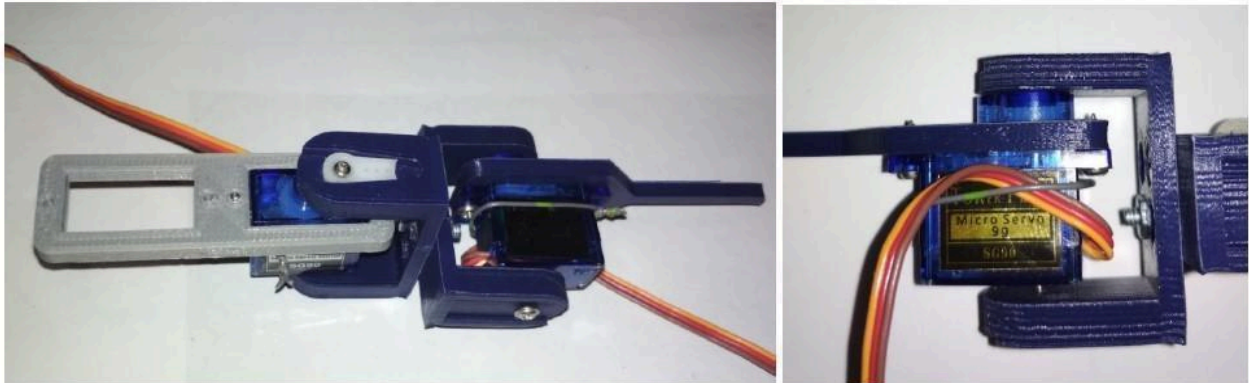


Figura 11: Pata completa y lazo para sostener los cables del servo.

5. Ejercicios

1. Probar, con un docente, el correcto funcionamiento de todos los servos entregados. No se debe comenzar con el siguiente ejercicio hasta no haber finalizado esta prueba.

Además de la prueba en sí, se explicará el programa utilizado con el objetivo de ir comprendiendo la mecánica de funcionamiento.

2.
 - a. Estudiar el comportamiento del Sketch `servo_ini_microbit.ino` (disponible en EVA).
 - b. Conectar la placa Microbit a la placa para el manejo de servos y a la PC. La placa para el control de servos también debe estar alimentada, puede ser mediante usb a la PC o a un cargador de celular. Conectar un servomotor al pin indicado por PIN SERVO. Cargar en la placa el código `servo_ini_microbit.ino`.
 - c. Modificar el código de forma que el eje del servo se inicialice en distintos ángulos (de 10° a 170°). ¿Qué observa?
 - d. Modificar el circuito de forma que tenga 2 servos conectados en los pines 7 y 8 respectivamente.
 - e. A partir del sketch `servo_ini_microbit.ino`, modificarlo para contemplar la parte d, de forma que el servo en el pin 7 se inicialice a 60° y el del pin 8 lo haga a 90° .
3.
 - a. Estudiar el Sketch `servo_if_microbit.ino` (disponible en EVA).
 - b. Cargar el código `servo_if_microbit.ino` y verificar el comportamiento.
 - c. Modificar el código de forma que se comporte igual, pero haciendo que el servo se mueva en el sentido inverso.
 - d. Utilizando los botones de la microbit (ver obligatorio 1), modificar el código de forma que:
 - Botón sin apretar (pin HIGH) → el motor deberá moverse como el caso original.
 - Botón apretado (pin LOW) → el motor deberá moverse como en la parte c.
4. Se va a utilizar un LED externo como señal luminosa y un servomotor para simular el movimiento de una barrera en el cruce de una carretera con una vía de tren. Considerando el

ejercicio 3 del obligatorio 1 y el ejercicio anterior, utilizar los botones de la Microbit como entradas de forma que:

- Si el botón B es apretado, indica que se aproxima un tren por lo que se deberá parpadear el LED y empezar a bajar la barrera (el servo deberá moverse de manera incremental como en el ejercicio anterior). Una vez que el servo llegue a su posición final deberá permanecer allí y el LED continuará parpadeando.
- Una vez que baja la barrera, al apretar el botón A se indica que el tren ya pasó. En este caso se deberá apagar el LED y comenzar a subir la barrera. Una vez que la barrera llegue a su posición más alta deberá permanecer allí.

Los pines donde se conectan el led y el servo, así como los ángulos a utilizar en el servo son de libre elección.

5. Procedimiento para armar una pata correctamente.

- a) Conectar 2 servos a la placa de control de servos (Kitronik 16) con la Microbit e inicializarlos en 90°. Puede ser útil referirse al código `servo_ini_microbit.ino`.
- b) Desconectar los servos.
- c) Armar una pata en posición completamente estirada. No modificar la posición del piñón al ensamblar, de lo contrario deberá volver al punto a)
- d) Con la placa desconectada, mover los servos con la mano a cualquier posición. Luego conectar la placa y ejecutar nuevamente la parte a). Seguramente la pata no volvió exactamente a la posición completamente estirada, pero está próxima.
- e) Modificar el código de forma de asegurar que al iniciar los servos la pata quede completamente estirada. A este procedimiento le llamaremos "calibrar".