

Proyecto: Implementación de un robot 5R, con motores de modelismo.

### Objetivo

Implementar un robot serial de cinco grados de libertad que conforme un banco de pruebas para prácticas de robótica, mediante el desarrollo de una interfase de control.

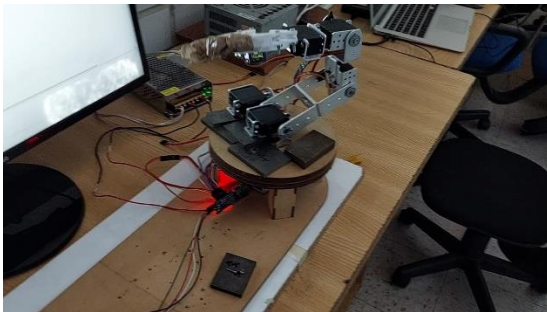


Figura 1: Robot 5R.

### Configuración e instalación

Se nos proporcionó el robot de la figura 1, una raspberry Pi 4 computer model B, una pantalla y un mouse con teclado para poder realizar las actividades previstas del proyecto y el módulo PCA9685.

El profesor nos hizo el favor de instalar la imagen que contenía el sistema operativo Raspbian.

Como se decidió trabajar en Python necesitamos agregar soporte SMBus que incluye I2C. Con ayuda del siguiente código ingresado directo a la terminal se instaló el SMBus:

```
sudo apt-get install python-smbus
```

```
sudo apt-get install i2c-tools
```

Con respecto a Python, utilizamos Thonny que ya venía instalado en el Raspbian.

Se conecto de primera instancia un Servomotor del robot modelo MG995 para empezar a hacer pruebas. Se conectó la Raspberry con el PCA9685 como se muestra en la siguiente figura:

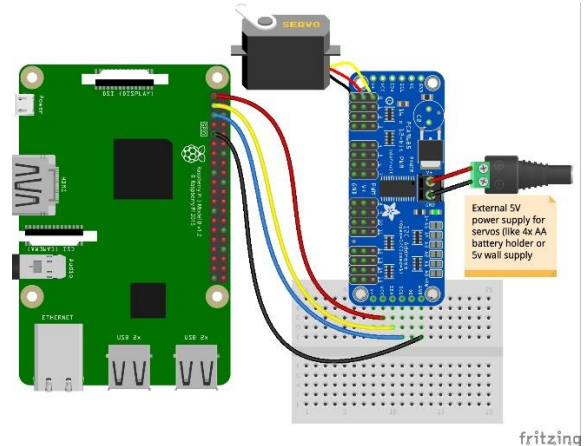


Figura 2: Conexión de un servomotor.

El PCA9685 funciona con el suministro de 3.3V por lo que debe de conectarse tal cual se ve en la imagen, esto para evitar estropear la RasPI.

Necesitamos instalar la librería Adafruit\_Blinka para poder controlar los servomotores con Python, para esto se necesita ejecutar desde la terminal el siguiente comando:

```
sudo pip3 install adafruit-circuitpython-servokit
```

En este punto empezamos a realizar la primera prueba que puede verse en el siguiente código:

```
#Prueba 1 del servomotor 0 ubicado en el canal 0 con  
continucion del servo 1 ubicado en el canal 1
```

```
import time
```

```
from adafruit_servokit import Servokit #en estas dos  
lineas importamos las librerias
```



*#establecer canales respecto a la cantidad de canales del servo*

*#8 para servo FeatherWing de 8x12 bit PWM, 16 para Sheld/Hat/Bonnet*

*kit = Servokit(channels=16) #nuestro caso fue de 16*

*kit.servo[0].angle=180 #mover el servo conectado al canal 0 a 180 grados*

*kit.continuous\_servo[1].throttle=1 #iniciar el servo de rotación continua conectado al canal 1 a toda velocidad hacia adelante*

*time.sleep(1) #tiempo de espera*

*kit.continuous\_servo[1].throttle=-1 #iniciar el servo de rotación continua conectado al canal 1 a toda velocidad hacia atras*

*time.sleep(1)*

*kit.servo[0].angle=0 #mover el servo conectado al canal 0 a 0 grados*

*kit.continuous\_servo[1].throttle=0 #detiene el servo de rotación continua conectado al canal 1*

Una vez verificamos el funcionamiento del código, proseguimos a conectar todos los servos como se puede ver en la figura 3

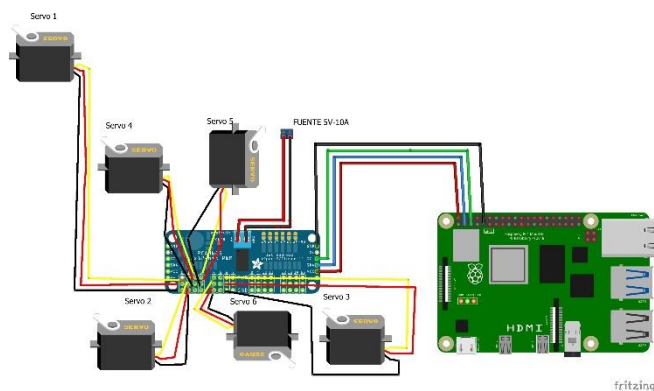


Figura 3: Conexión de los servomotores.

Continuamos con la prueba dos en la que movimos en conjunto todos los servos, también se hizo la prueba de moverlo uno a uno, pero finalmente se dejó como en el siguiente código:

*#Prueba 2 conexion de todos los servomotores*

*import time*

*from adafruit\_servokit import Servokit #en estas dos lineas importamos las librerias*

*#establecer canales respecto a la cantidad de canales del servo*

*#8 para servo FeatherWing de 8x12 bit PWM, 16 para Sheld/Hat/Bonnet*

*kit = Servokit(channels=16)*

*#servo1*

*kit.servo[0].angle=0 #mover el servo conectado al canal 0 a 0 grados*

*kit.continuous\_servo[1].throttle=1 #iniciar el servo de rotación continua conectado al canal 1 a toda velocidad hacia adelante*

*time.sleep(1) #tiempo de espera*

*kit.continuous\_servo[1].throttle=-1 #iniciar el servo de rotación continua conectado al canal 1 a toda velocidad hacia atras*

*time.sleep(1)*

*kit.servo[0].angle=0 #mover el servo conectado al canal 0 a 0 grados*

*kit.continuous\_servo[1].throttle=0 #detiene el servo de rotación continua conectado al canal 1*

*#servo2*

*kit.servo[2].angle=0 #mover el servo conectado al canal 2 a 0 grados*

*kit.continuous\_servo[1].throttle=1 #iniciar el servo de rotación continua conectado al canal 1 a toda velocidad hacia adelante*

*time.sleep(1) #tiempo de espera*

*kit.continuous\_servo[1].throttle=-1 #iniciar el servo de rotación continua conectado al canal 1 a toda velocidad hacia atras*



*time.sleep(1)*

*kit.servo[2].angle=0 #mover el servo conectado al canal 2 a 0 grados*

*kit.continuous\_servo[1].throttle=0 #detiene el servo de rotación continua conectado al canal 1*

*#servo4*

*kit.servo[3].angle=90 #mover el servo conectado al canal 3 a 90 grados*

*kit.continuous\_servo[2].throttle=1 #iniciar el servo de rotación continua conectado al canal 2 a toda velocidad hacia adelante*

*time.sleep(1) #tiempo de espera*

*kit.continuous\_servo[2].throttle=-1 #iniciar el servo de rotación continua conectado al canal 2 a toda velocidad hacia atras*

*time.sleep(1)*

*kit.servo[3].angle=0 #mover el servo conectado al canal 3 a 0 grados*

*kit.continuous\_servo[2].throttle=0 #detiene el servo de rotación continua conectado al canal 2*

*#servo5*

*kit.servo[4].angle=90 #mover el servo conectado al canal 4 a 90 grados*

*kit.continuous\_servo[3].throttle=1 #iniciar el servo de rotación continua conectado al canal 3 a toda velocidad hacia adelante*

*time.sleep(1) #tiempo de espera*

*kit.continuous\_servo[3].throttle=-1 #iniciar el servo de rotación continua conectado al canal 3 a toda velocidad hacia atras*

*time.sleep(1)*

*kit.servo[4].angle=0 #mover el servo conectado al canal 4 a 0 grados*

*kit.continuous\_servo[3].throttle=0 #detiene el servo de rotación continua conectado al canal 3*

*#servo6*

*kit.servo[5].angle=90 #mover el servo conectado al canal 5 a 90 grados*

*kit.continuous\_servo[4].throttle=1 #iniciar el servo de rotación continua conectado al canal 4 a toda velocidad hacia adelante*

*time.sleep(1) #tiempo de espera*

*kit.continuous\_servo[4].throttle=-1 #iniciar el servo de rotación continua conectado al canal 4 a toda velocidad hacia atras*

*time.sleep(1)*

*kit.servo[5].angle=0 #mover el servo conectado al canal 5 a 0 grados*

*kit.continuous\_servo[4].throttle=0 #detiene el servo de rotación continua conectado al canal 4*

Una vez verificamos el funcionamiento de los motores, nos percatamos que el robot estaba mal armado y la posición 0 de nuestros motores no correspondía con la posición 0 de las piezas, por lo que se prosiguió a corregir este error.

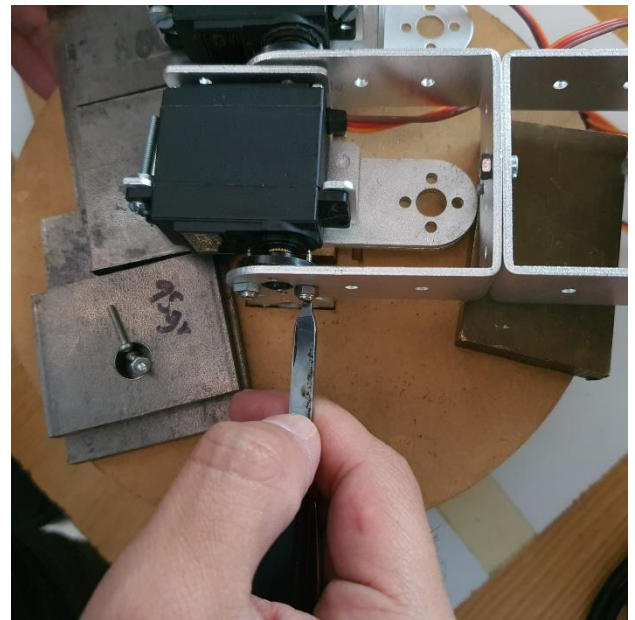


Figura 4: Corrección de las piezas del motor.

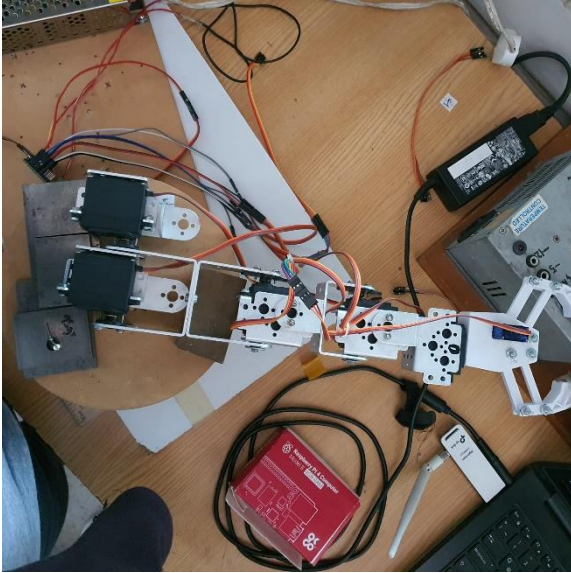


Figura 5: Robot en posición 0.

Continuamos con la prueba tres como en el siguiente código:

*#Prueba 3 conexion de todos los servomotores sin retrocesos*

```
from time import *
```

```
from adafruit_servokit import Servokit #en estas dos  
lineas importamos
```

```
#establecer canales respecto a la cantidad de canales del  
servo
```

```
#8 para servo FeatherWing de 8x12 bit PWM, 16 pata  
Sheld/Hat/Bonnet
```

```
kit = Servokit(channels=16)
```

```
#Se colocan los angulos iniciales de todos los servos
```

```
kit.servo[0].angle=0 #servo conectado al canal 0 a 0  
grados
```

```
kit.servo[1].angle=0 #servo conectado al canal 1 a 0  
grados
```

```
kit.servo[2].angle=0 #servo conectado al canal 2 a 0  
grados
```

```
kit.servo[3].angle=0 #servo conectado al canal 3 a 0  
grados
```

```
kit.servo[4].angle=0 #servo conectado al canal 4 a 0  
grados
```

```
def servo1():
```

```
    kit.servo[0].angle=0 #servo1 conectado en el  
    canal 0 se posiciona a 0 grados
```

```
def servo2():
```

```
    kit.servo[1].angle=90 #servo2 conectado en el  
    canal 1 se posiciona a 90 grados
```

```
def servo4():
```

```
    kit.servo[2].angle=0 #servo4 conectado en el  
    canal 2 se posiciona a 0 grados
```

```
def servo5():
```

```
    kit.servo[3].angle=0 #servo5 conectado en el  
    canal 3 se posiciona a 0 grados
```

```
def servo6():
```

```
    kit.servo[4].angle=0 #servo6 conectado en el  
    canal 4 se posiciona a 0 grados
```

```
#Se crea un bucle
```

```
while True:
```

```
    servo1()
```

```
    sleep(0.5)
```

```
    servo2()
```

```
    sleep(0.5)
```

```
    servo4()
```

```
    sleep(0.5)
```

```
    servo5()
```

```
    sleep(0.5)
```

```
    servo6()
```

```
    sleep(0.5)
```



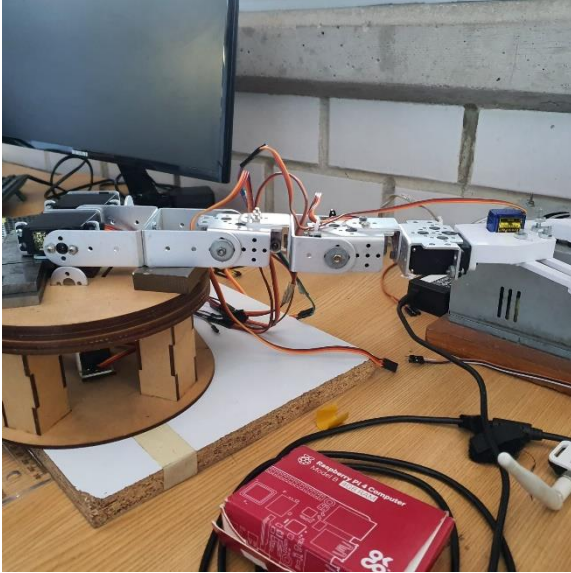


Figura 5: Robot en ejecución Prueba3.

Con la prueba 3 nos dimos cuenta de que la fuerza en el servomotor 2 no era suficiente para poner el robot en posición de 90°, algo que ya suponíamos porque las dimensiones y masa del robot suponen un par muy grande. Por este motivo y dando solución a este problema decidimos acoplar el motor 3 en paralelo al motor 2 y de esta forma poder soportar el par generado por el robot. Un problema que tuvimos con esto fue de nuevo ajustar a cero el servomotor 3 y, sobre todo, acoplarlo con el servomotor 2.

Después de volver a calibrar los motores se conectó el servomotor 3 como se puede apreciar en la figura 6 y se ejecutó el siguiente código:

*#Prueba Final robot levantado a 90 grados*

```
from time import *  
  
from adafruit_servokit import Servokit #en estas dos  
lineas importamos  
  
#establecer canales respecto a la cantidad de canales del  
servo  
  
#8 para servo FeatherWing de 8x12 bit PWM, 16 pata  
Shield/Hat/Bonnet  
  
kit = Servokit(channels=16)  
  
#Se colocan los angulos iniciales de todos los servos
```

```
kit.servo[0].angle=0 #servo conectado al canal 0 a 0  
grados
```

```
kit.servo[1].angle=0 #servo conectado al canal 1 a 0  
grados
```

```
kit.servo[2].angle=0 #servo conectado al canal 2 a 0  
grados
```

```
kit.servo[3].angle=0 #servo conectado al canal 3 a 0  
grados
```

```
kit.servo[4].angle=0 #servo conectado al canal 4 a 0  
grados
```

```
kit.servo[5].angle=0 #servo conectado al canal 5 a 0  
grados
```

```
def servo1():
```

```
    kit.servo[0].angle=0 #servo1 conectado en el  
    canal 0 se posiciona a 0 grados
```

```
def servo2():
```

```
    kit.servo[1].angle=90 #servo2 conectado en el  
    canal 1 se posiciona a 90 grados
```

```
def servo3():
```

```
    kit.servo[5].angle=90 #servo5 conectado en el  
    canal 1 se posiciona a 90 grados
```

```
def servo4():
```

```
    kit.servo[2].angle=0 #servo4 conectado en el  
    canal 2 se posiciona a 0 grados
```

```
def servo5():
```

```
    kit.servo[3].angle=0 #servo5 conectado en el  
    canal 3 se posiciona a 0 grados
```

```
def servo6():
```

```
    kit.servo[4].angle=0 #servo6 conectado en el  
    canal 4 se posiciona a 0 grados
```

```
#Se crea un bucle
```

```
while True:
```

```
    servo1()
```



`sleep(0.5)`

`servo2()`

`sleep(0.5)`

`servo3()`

`sleep(0.5)`

`servo4()`

`sleep(0.5)`

`servo5()`

`sleep(0.5)`

`servo6()`

`sleep(0.5)`

## Referencias

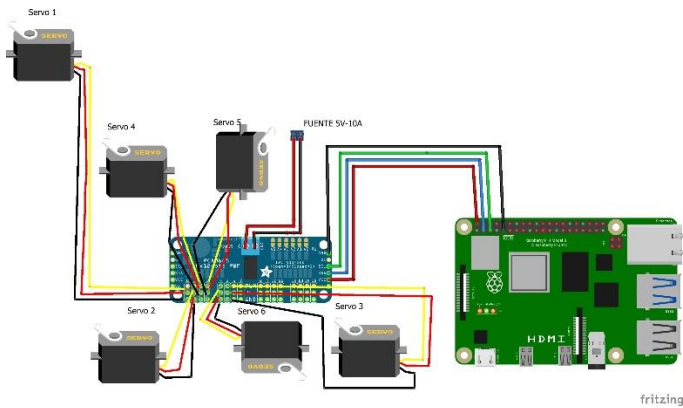


Figura 6: Conexión de los servomotores completa.



Figura 7: Robot levantado a 90°.