

MANUALES

Armado e iniciación



Alumnos:

Arien Michelle López Jáuregui 20300712

Carlos Isaac Torres Méndez 20300715

Esteban Barboza Ochoa 20300719

Carrera:

Control automático e Instrumentación

Asesor:

Ph.D Alfonso Pérez Gonzales

Asesor externo:

Ing. Carlos Barboza Ochoa

Ph.D. César Méndez García



ÍNDICE

MANUAL ARMADO E INICIACIÓN.....	3
INTRODUCCIÓN.....	3
OBJETIVO.....	3
LISTA DE PARTES:.....	3
CALIBRACIÓN DE SERVOS:.....	6
ARMADO:.....	8
BLUETOOTH:.....	12

MANUAL ARMADO E INICIACIÓN

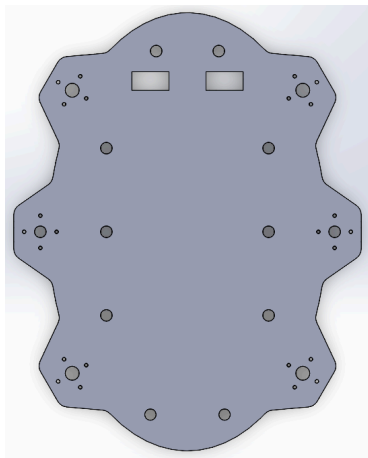
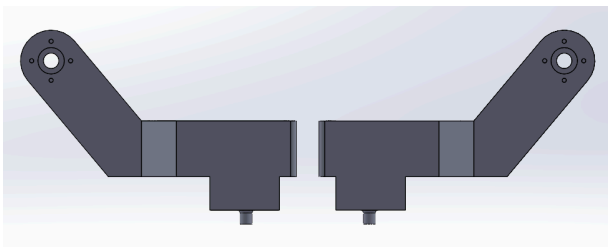
INTRODUCCIÓN

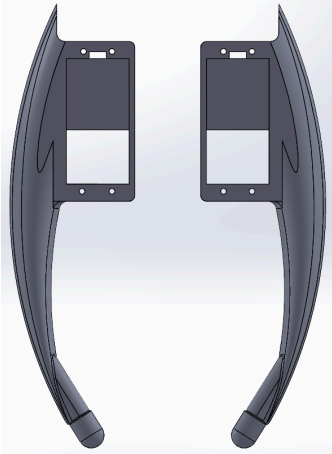
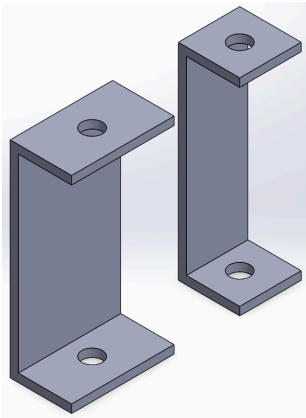

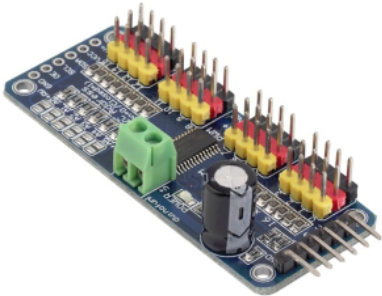
En este documento se mostrará y detalla todo lo necesario para el funcionamiento de Spike.

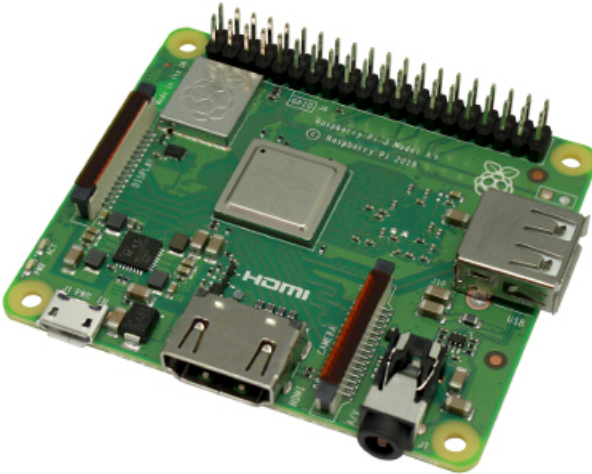




OBJETIVO

Explicar cómo hacer el armado de Spike, cómo hacer la iniciación del robot y las calibraciones que lleguen a ser necesarias para su buen funcionamiento.

LISTA DE PARTES:

Pieza	Cantidad	Imagen
Base (Imprimir)	2	
Fémur (Imprimir)	3 derecho 3 izquierdo	

Tarso (Imprimir)	3 derecho 3 izquierdo	
Soportes	4 largo 4 corto	
Servomotor MG995 (compra externa)	12	
Módulo Controlador de servos PCA9685	1	

Raspberry: Recomendamos modelos con bluetooth integrado (como la PI 3 modelo A+ o la Zero o Zero 2W) si se quiere utilizar la app, en caso de no querer usar esta función, se pueden usar otros modelos.	1	
Batería LiPo	1	
Conector circular (Incluida en servos)	12	
Tornillo A (Incluido en servos)	12	
Tornillo B (Incluido en servos)	24	

Tornillo C #2 x 3/8" (compra externa)	12	
Tornillo D M1.6 x 10 mm + tuercas (compra externa)	12	
Tornillo E 1/8" x 1/2 + tuercas (compra externa)	14	

CALIBRACIÓN DE SERVOS:

IMPORTANTE: solo realizar este apartado en caso de usar un modelo de servomotor diferente al MG995.

Para el lograr una calibración de servos seguir los siguientes pasos:

- Investigar la datasheet del servomotor a utilizar y buscar los siguientes datos:
 - Frecuencia de funcionamiento.
 - pulse width para 0 grados
 - pulse width para 180 grados
- Poner el conector circular al servo para que sea más sencillo visualizar los ángulos de los giros.
- Conectar un servo al módulo PCA9685, el cual fue conectado previamente a la raspberry como se muestra en el diagrama de conexión (manual técnico).
- Abrir el código “Motores.py” y realizar lo siguiente:
 - En caso de encontrar los valores en la datasheet:
 - Dentro de la variable `puls_0` guardar el valor del pulse width para 0 grados.
 - Dentro de la variable `puls_180` guardar el valor del pulse width para 180 grados.
 - Dentro de la función `pwm.set_pwm_freq()` ingresar el valor de la frecuencia de funcionamiento.
 - Iniciar el código y verificar que en efecto el servo gire de 0 a 180 grados, en caso contrario, realizar ajustes a prueba y error de los valores de `puls_0` y `puls_180`.



- En caso de no encontrar los valores en la datasheet, utilizar los puestos en el código como referencia y realizar los ajustes necesarios.

```
11 #Configurar los pulsos
12 puls_0 = 100
13 puls_180 = 500
14
15 #Setear la frecuencia de los servos
16 pwm.set_pwm_freq(50)
17
```

- El código realizará la siguiente secuencia (se puede modificar la variable de tiempo entre movimientos t si el usuario lo ve necesario):
 - Girar 0 grados y esperar
 - Girar 90 grados y esperar
 - Girar 180 grados y esperar

```
17
18 #tiempo entre movimientos (segundos):
19 t = 1
20
```

```
25
26 try:
27     while True:
28         print("0")
29         pwm.set_pwm(0, 0, girar(0))
30         sleep(t)
31         print("90")
32         pwm.set_pwm(0, 0, girar(90))
33         sleep(t)
34         print("180")
35         pwm.set_pwm(0, 0, girar(180))
36         sleep(t)
37
```

- La importancia de probar con estos 3 ángulos radica en que podría pasar que los servos giren 0 y 180 grados, pero no tengan el pulse width correcto (sucede cuando se eligen valores menores al correcto de 0 y mayores al correcto de 180), entonces el resto de ángulos no serán correctos. Para ello agregamos el ángulo de 90°.



- Si los tres ángulos son correctos, el servo estará calibrado, en caso contrario realizar las modificaciones necesarias.
- Verificar que funcione correctamente para todos los servomotores que se quieran utilizar.

ARMADO:

Para el apartado de armado del robot seguir los siguientes pasos:

- Conectar los servos al módulo de relevadores previamente conectado a la raspberry. La conexión de servos a módulo puede ser arbitraria.

IMPORTANTE: no tener armada ninguna parte del robot, los servos deben estar sueltos.

- Utilizando el código “Principal.py”, descomentar el bloque “while True”. Dentro de él, descomentar tarsos90() y femurs90(). Deja comentado el resto y corre el código; esto hará que todos los servos tengan un ángulo de 90 grados para poder ser unidos al resto de partes.

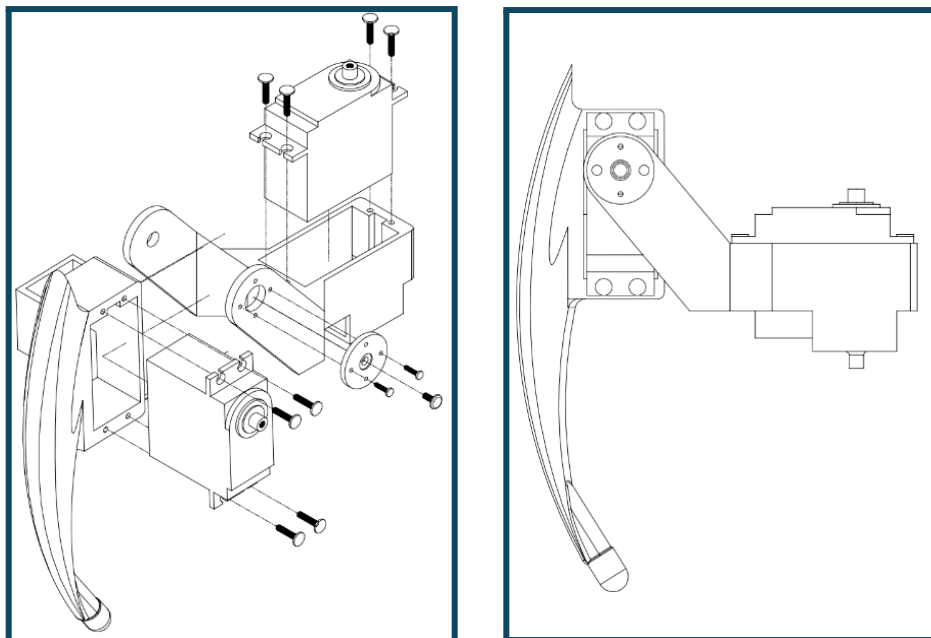
```

517 #-----
518 #Descomentar todas estas líneas en caso de querer hacer un bucle
519 while True:
520     #Read Gyroscope raw value
521     #x = leer_datos(x_dir)
522     #x = (-0.9956157522543754)*x + (0.7281501247343715)
523     #print(f"X: {x}g")
524     #print("90")
525     #pwm.set_pwm(0, 0, girar(90))
526     #sleep(1)
527
528     #iniciales()
529     #a_mimir()
530     #izquierda()
531     #derecha()
532     tarsos90()
533     femurs90()
534
535     #adelante()
536     #atras()
537 #-----
538
517 #-----
518 #Descomentar todas estas líneas en caso de querer hacer un
519 while True:
520     #Read Gyroscope raw value
521     #x = leer_datos(x_dir)
522     #x = (-0.9956157522543754)*x + (0.7281501247343715)
523     #print(f"X: {x}g")
524     #print("90")
525     #pwm.set_pwm(0, 0, girar(90))
526     #sleep(1)
527
528     #iniciales()
529     #a_mimir()
530     #izquierda()
531     #derecha()
532     tarsos90()
533     femurs90()
534
535     #adelante()
536     #atras()
537 #-----
538

```

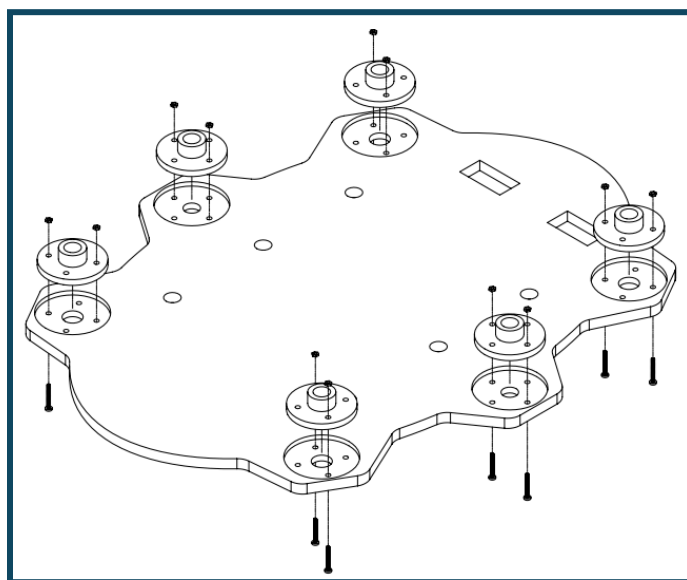
- Desconectar los servos del módulo y colocarlos en tarso y fémur, asegurándolos con los tornillos B. Unir el conector circular al fémur usando dos tornillos C y agregar la pieza del tarso con su motor usando la pieza circular como conexión, y un tornillo A para asegurar que quede sujeto.

Asegurarse que el motor tenga un ángulo de 90° en relación al fémur, como se muestra en las imágenes de referencia.



Repetir este paso con las 6 patas.

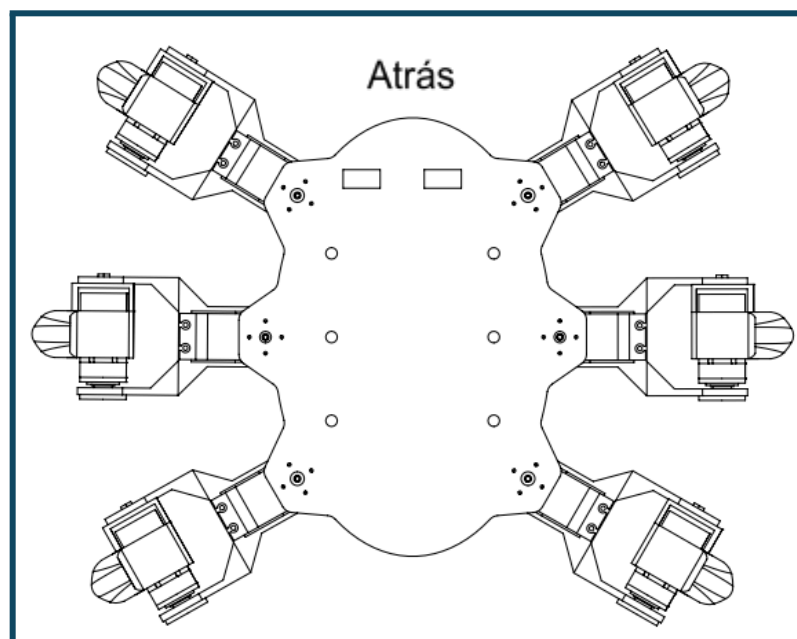
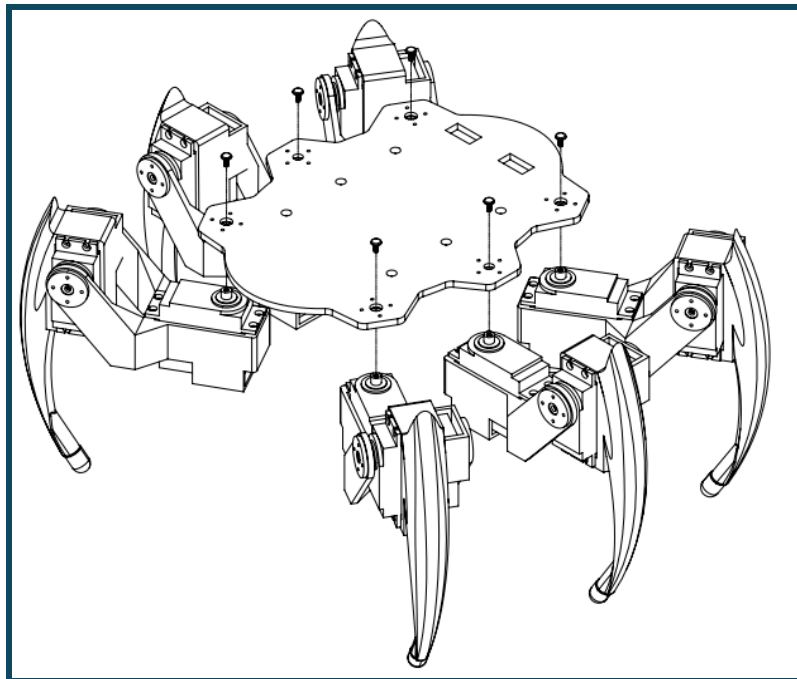
- Tomar una base, poner los conectores circulares donde se encuentran las hendiduras y asegurarlos bien utilizando los Tornillos D con sus respectivas tuercas.



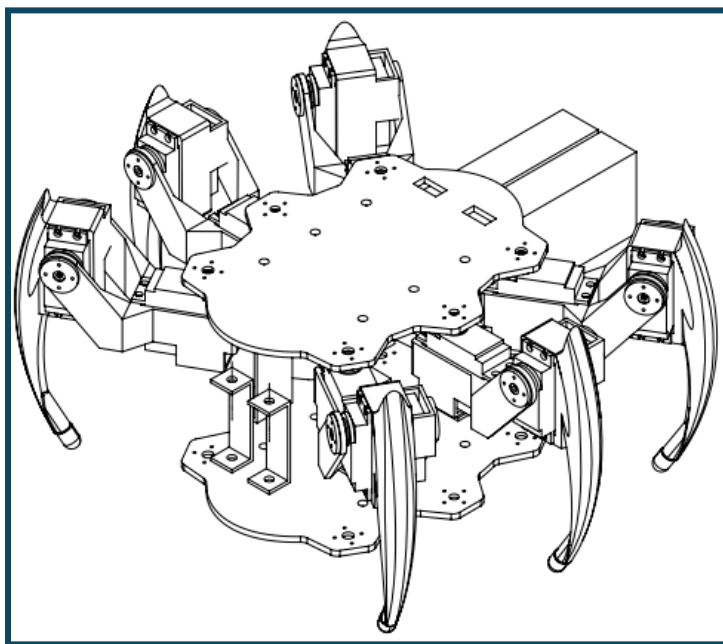
- Lo siguiente es agregar las piezas de las patas con ayuda de los tornillos A y los conectores circulares.

IMPORTANTE: las patas deben tener un acomodo como se muestra en la imagen.

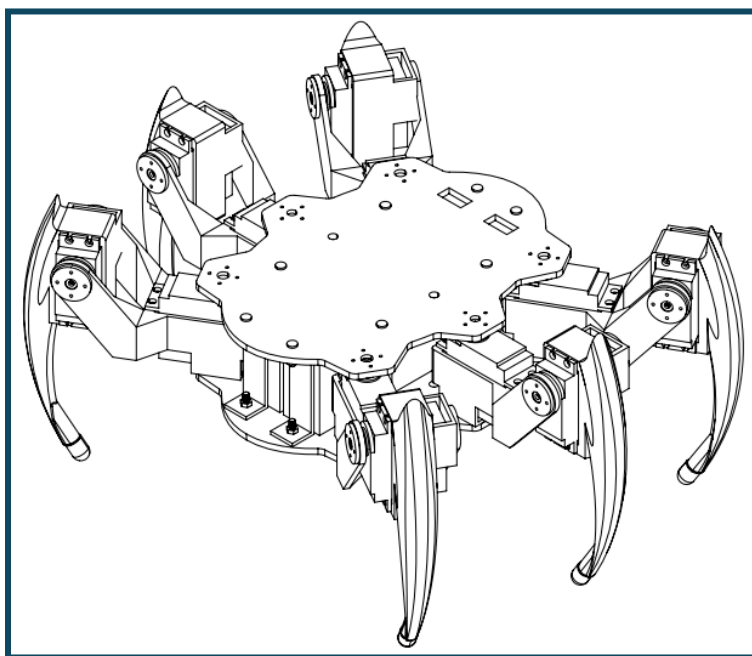
TIP: conectar los servos de atrás hacia adelante (atrás es donde la base tiene los rectángulos cortados) y cada que se conecte un servo pasar los cables a la parte superior a través de los rectángulos.



- Asegurar los soportes a la base con los tornillos E y sus tuercas, deslizar las pilas y colocar los otros dos soportes de la parte de atrás. Después, colocar la otra base y unirla con los tornillos E y sus tuercas a la parte de abajo de los soportes como se muestra en la imagen.



- Terminando el armado se vería algo así

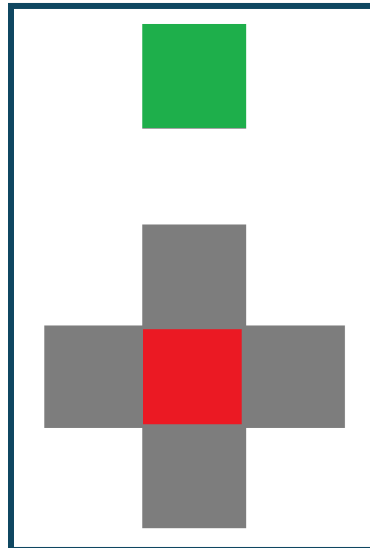


**BLUETOOTH:**

Para verificar el bluetooth es necesario que primero descarguen la app BlueDot (solo [android](#) o [python](#)).

Los pasos para verificar son los siguientes:

- Enlazar el bluetooth del dispositivo en el que tengamos descargada la app con nuestra raspberry. <https://descubrearduino.com/wifi-bluetooth-raspberry-pi/>
- Abrir el código “BT.py”.
- Iniciar el código en la raspberry, abrir la app BlueDot y verificar que se muestre la siguiente distribución de botones:



- El botón verde es el botón de inicio, el botón rojo será nuestro botón de descanso. Usando el rojo como referencia, el botón arriba de él será adelante, a su izquierda botón izquierda, a su derecha botón derecha y hacia abajo botón atrás.
- En la terminal de la raspberry deberá mostrarse la acción presionada en texto, en caso de que se muestran direcciones que no son las presionadas, verificar el código y que no se haya modificado algún punto.
- Si no hay problemas, el bluetooth funciona correctamente.