

Servo Driver HAT

From Waveshare Wiki

Jump to: navigation, search

Overview

Introduction

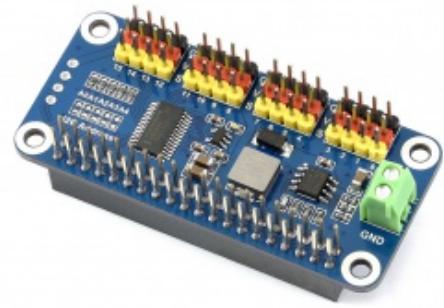
This product is a PWM/servo expansion board based on Raspberry Pi. It can expand 16-ch servo control or PWM output through the PCA9685 chip, and each channel has a 12-bit resolution. With the I2C interface for controlling, it does not require any other pins. The onboard 5V voltage regulator chip can be connected to the battery, and the maximum output current is 3A. It is suitable for controlling robotic arms and various servo robots.

Features

- Power supply: 6v-12v
- Servo voltage: 5V
- Logic voltage: 3.3V
- Driver: PCA9685
- Control interface: I2C
- Dimension: 65mm x 30mm
- Mounting hole size: 3.0mm

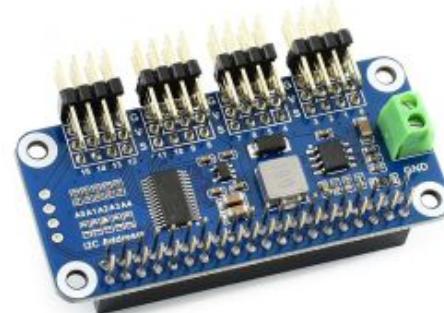
Hardware Description

Servo Driver HAT



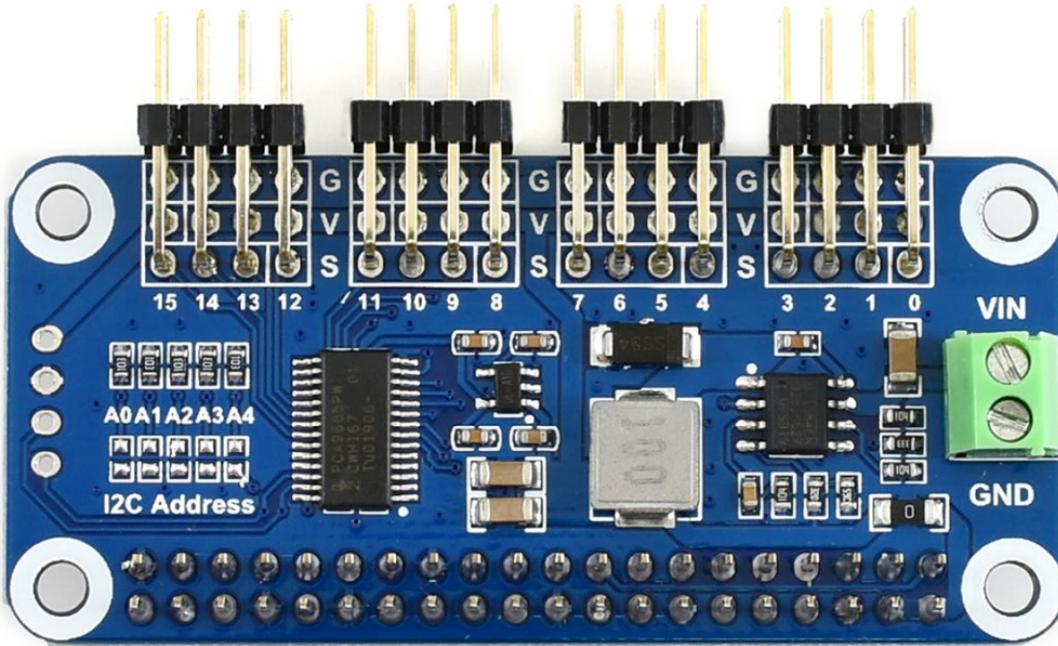
(<https://www.waveshare.com/servo-driver-hat.htm>)

Servo Driver HAT (B)



(<https://www.waveshare.com/servo-driver-hat-b.htm>)

RPI/I2C



(/wiki/File:Servo_Driver_HAT03.png)

It does not need additional power as its power supply is from Raspberry Pi.

Also, you can power it by the green terminal VIN on the right side with a range of 6V-12V. When the servo and Raspberry Pi are provided by 5V from the onboard constant voltage chip, the maximum current output is 3A.

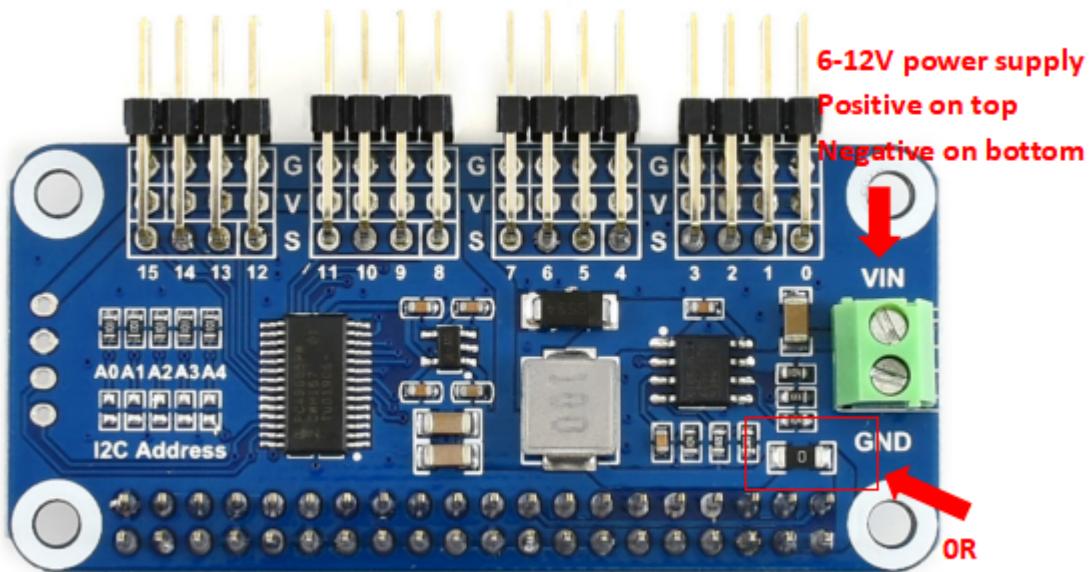
A0-A4 can be used to set the I2C device address of the PCA9685 chip and can be connected to multiple Servo Driver HAT at the same time.

The pin headers on the top are for connecting the servo. The black pin headers are for GND (mostly connected to the brown wire of the servo), the red ones are for 5V power supply and the yellow ones are for PWM signal cable. There are 0-15 channels and can connect to 16 servos at the same time. Please do not connect the servo cable reversely, or the servo will not rotate.

Note:

If a high-power servo is connected, the power supply may be insufficient as the whole board is powered by 5V. As both the RPI and the servo are powered by 5V, the RPI will reboot as a result of insufficient power when the power consumption is too high. In this

case, we should remove the onboard 0R resistor and connect the external power by VIN (6-12V).



(/wiki/File:Servo_Driver_HAT1.png)

Using with Raspberry Pi

To use this module, we provide python examples for test PCA9685, WiFi remote control and bluetooth remote control.

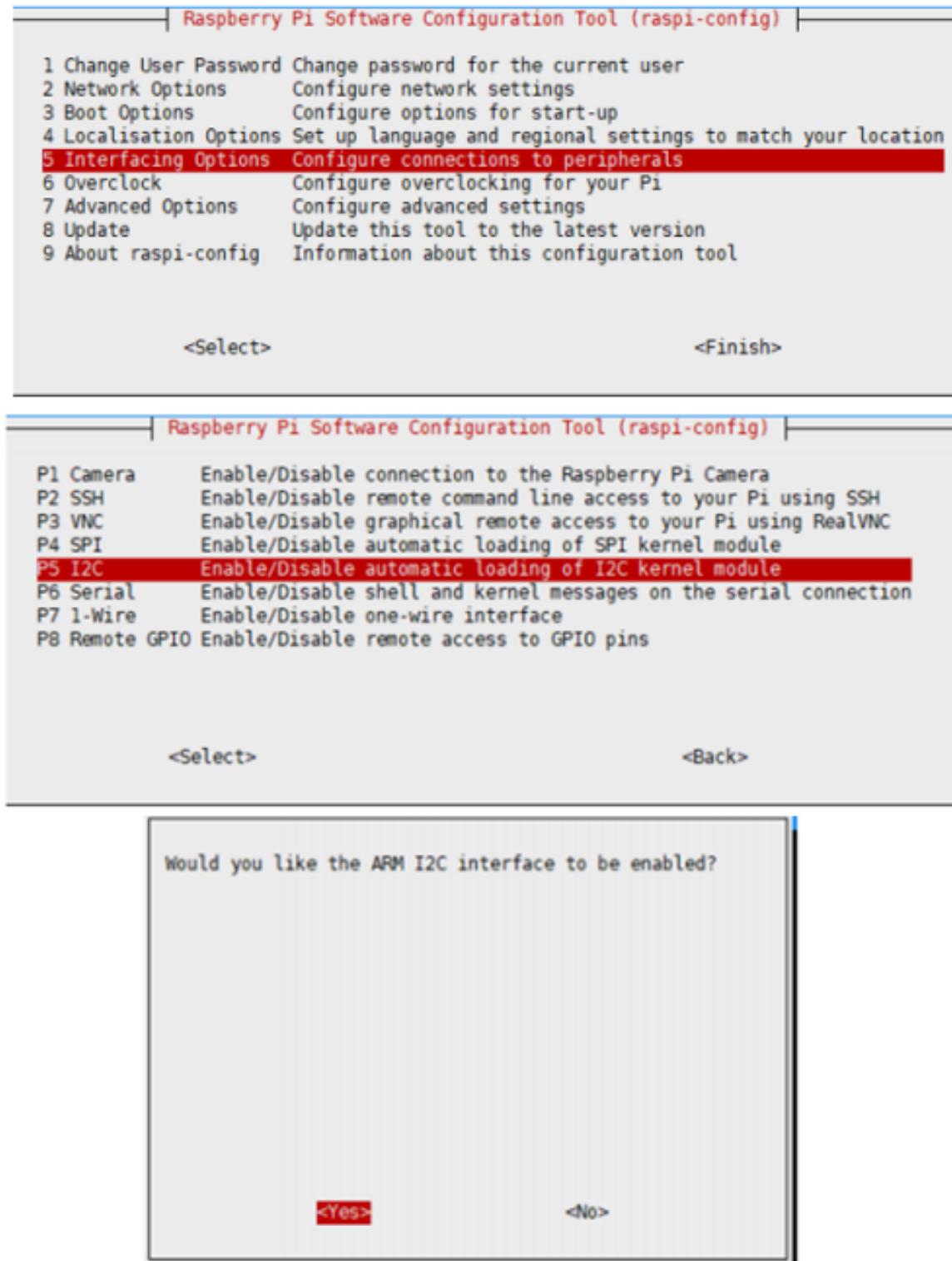
Enable I2C Interface

Open a terminal and run the following commands:

```
sudo raspi-config  
Choose Interfacing Options -> I2C ->yes
```

Reboot Raspberry Pi:

```
sudo reboot
```



(/wiki/File:RPI_open_i2c.png)

Install libraries

```
sudo apt-get update  
sudo apt-get install python-pip  
sudo pip install RPi.GPIO  
sudo apt-get install python-smbus
```

Downalod the demo codes and unzip

You should start the Raspberry Pi, open a terminal and run the following commands:

```
sudo apt-get install p7zip-full  
wget https://www.waveshare.com/wiki/File:Servo_Driver_HAT.7z  
7zr x Servo_Driver_HAT.7z -r -o./Servo_Driver_HAT  
sudo chmod 777 -R Servo_Driver_HAT  
cd Servo_Driver_HAT/Raspberry\ Pi/
```

Python Examples

Open a terminal and runt the following comamnds:

```
#For python2  
cd ~/Servo_Driver_HAT/Raspberry\ Pi/  
cd python/  
sudo python PCA9685.py  
#For python3  
cd ~/Servo_Driver_HAT/Raspberry\ Pi/  
cd python3/  
sudo python3 PCA9685.py
```

Expected result:Connect a servo to Channel 0, the servo will rotate..

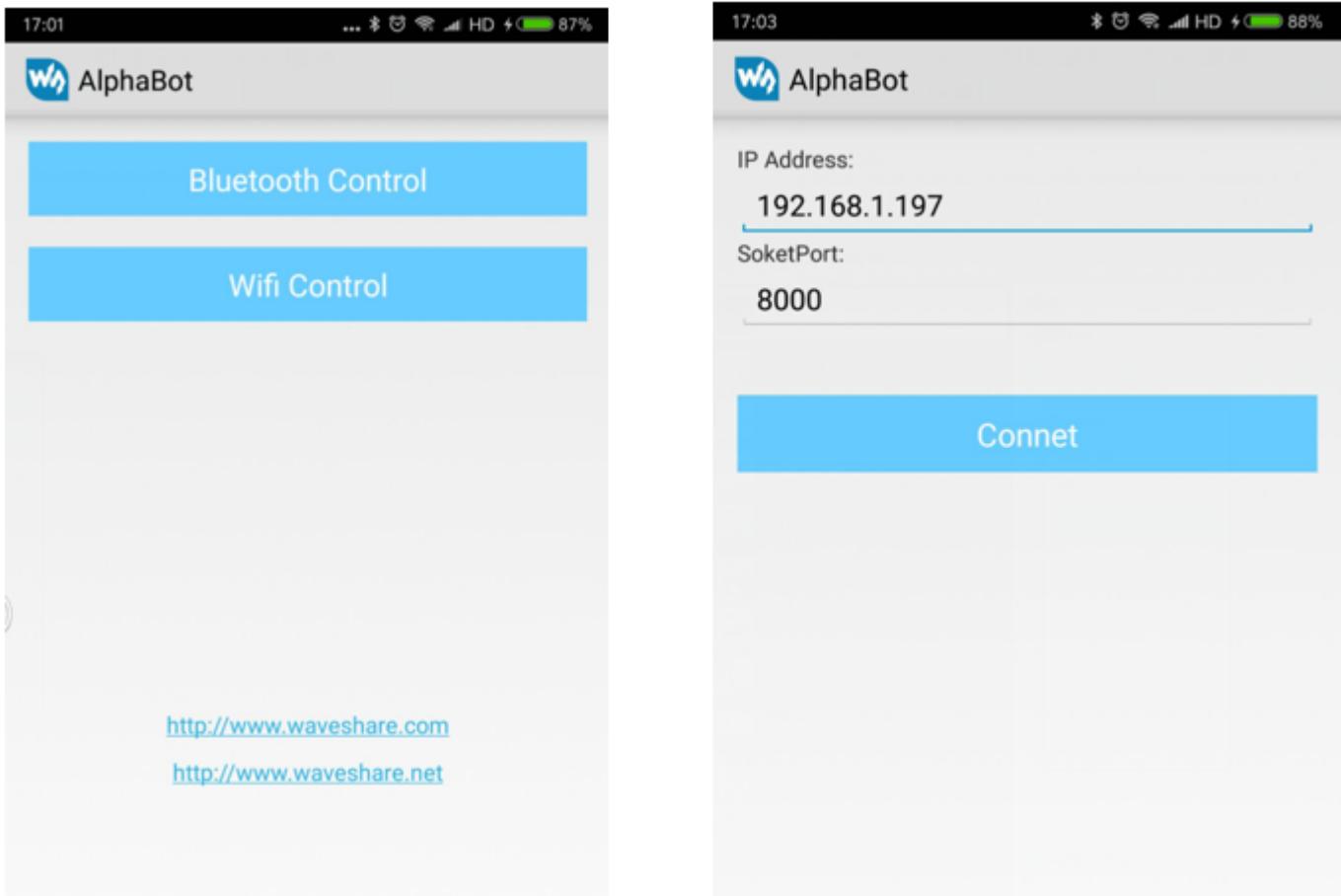
WIFI Remote Control

Open a terminal and run the following commands:

```
cd Wifi-Control/  
sudo python main.py
```

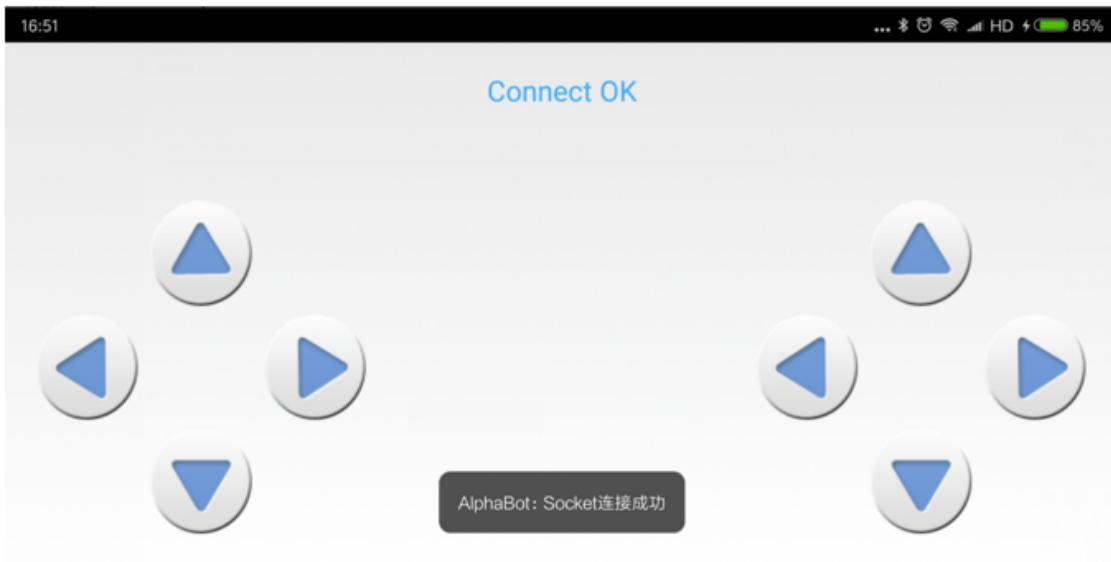
Please connect the Raspberry Pi and telephone to the same WLAN network. Data will be transmitted by TCP protocol. After running the program, the IP address will be displayed and the port number is 8000.

Open the APP and choose the WIFI control, input the IP address and the port then connect.



(/wiki/File:Servo_driver_hat_wifi.png)

It enters the control page if connected successfully. You can click the button to control the four servos (Channel 0 to Channel 4).

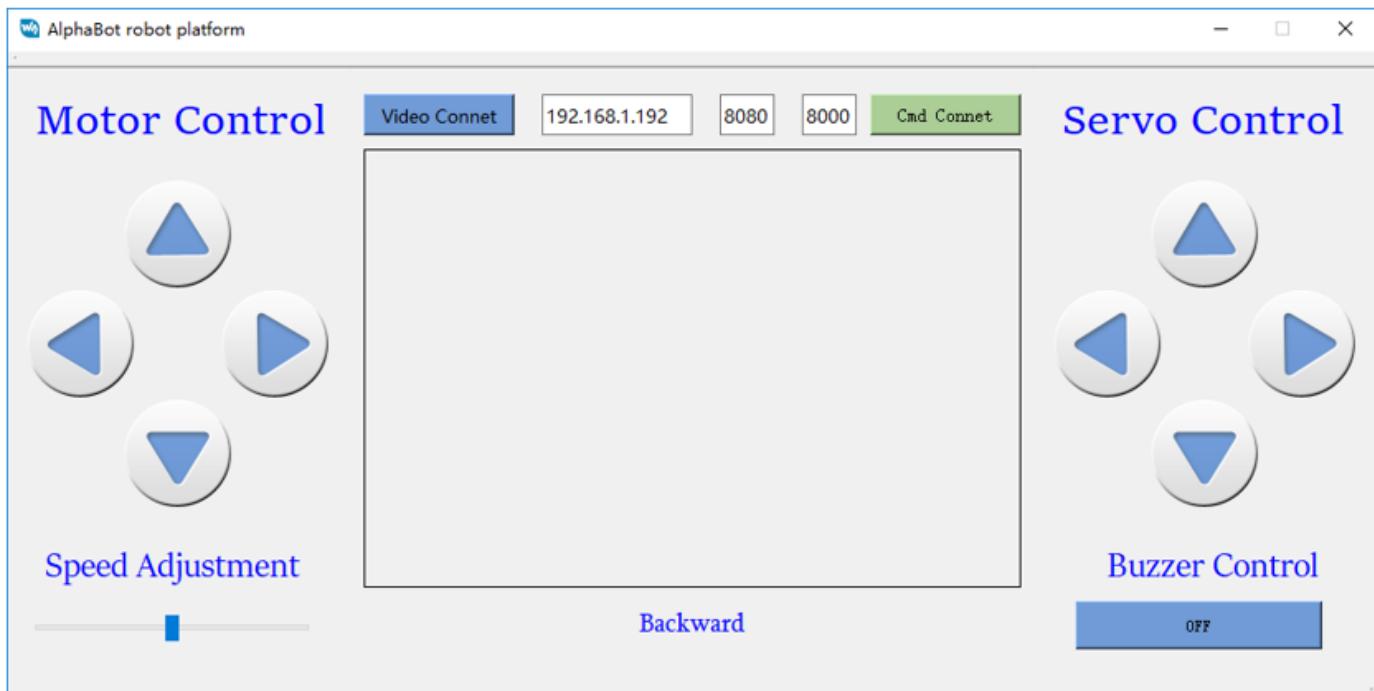


```
pi@raspberrypi:~/Servo_Driver_HAT/python/Wifi-Control $ sudo python main.py
start
192.168.1.192

server is running....
('got connection from ', ('192.168.1.184', 59089))
Forward
Stop
Backward
Stop
```

(/wiki/File:Servo_driver_hat_wifi1.png)

You can also connect it with QT software in windows PC.



(/wiki/File:Servo_driver_hat_wifi2.png)

Note: You can download the APP and the QT software from the Resources part.

Bluetooth Remote Program

Execute:

```
sudo apt-get update  
sudo apt-get upgrade -y  
sudo apt-get dist-upgrade -y  
sudo apt-get install pi-bluetooth bluez bluez-firmware blueman  
sudo usermod -G bluetooth -a pi  
sudo vi /etc/systemd/system/dbus-org.bluez.service  
sudo reboot
```

Start/add SPP, turn on the bluetooth device

```
sudo vi /etc/systemd/system/dbus-org.bluez.service
```

```
1 [Unit]  
2 Description=Bluetooth service  
3 Documentation=man:bluetoothd(8)  
4  
5 [Service]  
6 Type=dbus  
7 BusName=org.bluez  
8 ExecStart=/usr/lib/bluetooth/bluetoothd -C  
9 ExecStartPost=/usr/bin/sdptool add SP  
10 NotifyAccess=main  
11 #WatchdogSec=10  
12 #Restart=on-failure  
13 CapabilityBoundingSet=CAP_NET_ADMIN CAP_NET_BIND_SERVICE  
14 LimitNPROC=1  
15  
16 [Install]  
17 WantedBy=bluetooth.target  
18 Alias=dbus-org.bluez.service
```

(/wiki/File:Servo_driver_hat_bt.png)

After restarting the Raspberry Pi, enter the hciconfig command (similar to the ifconfig command) to check the Bluetooth service.

```
pi@raspberrypi:~ $ sudo hciconfig
hci0:  Type: BR/EDR  Bus: UART
      BD Address: B8:27:EB:2D:00:87  ACL MTU: 1021:8  SCO MTU: 64:1
      UP RUNNING
      RX bytes:717 acl:0 sco:0 events:42 errors:0
      TX bytes:1532 acl:0 sco:0 commands:42 errors:0
```

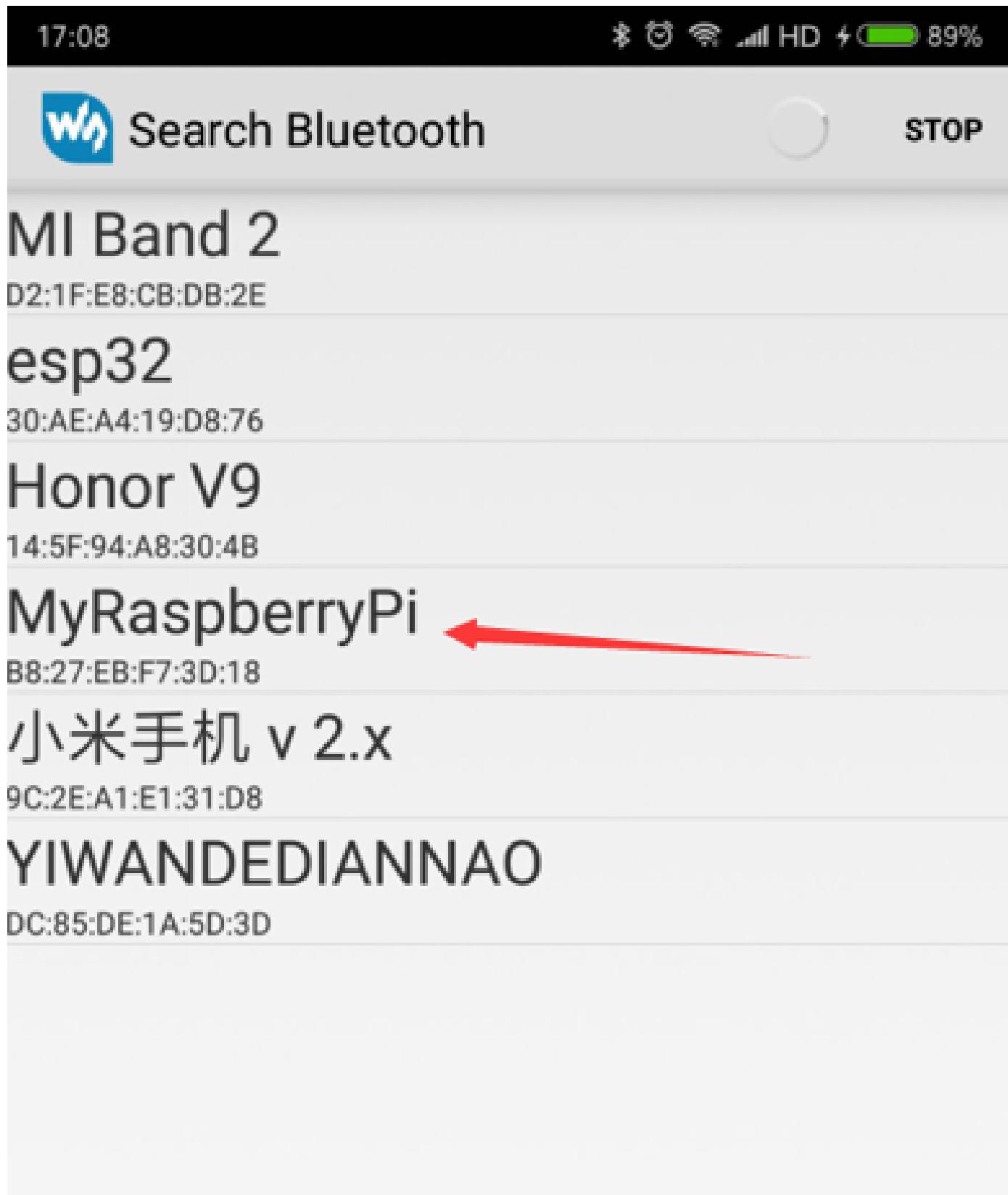
(/wiki/File:Servo_driver_hat_bt2.png)

If you see the hci0 device, the bluetooth has been turned on. If not then no bluetooth device is recognized.

Run the following command to start the program.

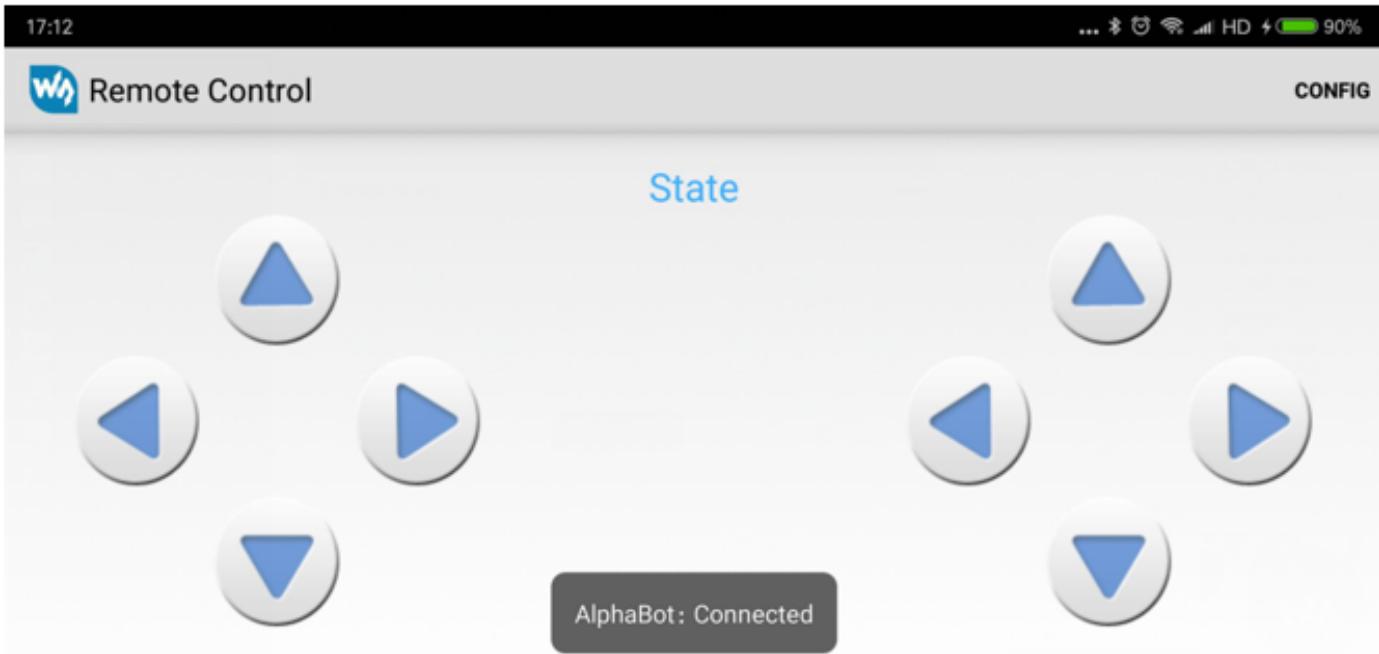
```
cd Servo_Driver_HAT/python/Bluetooth-Control
sudo ./Bluetooth.sh
```

After the program runs, you will be prompted to wait for the bluetooth connection, choose the bluetooth remote control on the mobile app, clicks scan, discovers the raspberrypi device and connects.



(/wiki/File:Servo_driver_hat_bt3.png)

After connecting the program, enter the control interface, and click the button to remotely control the forward and reverse rotation of the servos of channels 0~4.



(/wiki/File:Servo_driver_hat_bt4.png)

Note: If the received command is incorrect, you may need to configure the command sent when the corresponding button is pressed and released in the APP.

```
pi@raspberrypi:~/Servo_Driver_HAT/python/Bluetooth-Control $ sudo ./bluetooth.sh
[NEW] Controller B8:27:EB:F7:3D:18 raspberrypi [default]
[NEW] Device 94:87:E0:DA:12:1D 小米手机
[bluetooth]# discoverable on
[DEL] Controller B8:27:EB:F7:3D:18 raspberrypi [default]
[bluetooth]#
(bluetoothctl:32748): GLib-CRITICAL **: Source ID 23 was not found when attempting to remove it
Waiting for connection on channel 1
Connection from 94:87:E0:DA:12:1D to /dev/rfcomm0
Press CTRL-C for hangup
serial test start ...
Forward
Stop
Backward
Stop
```

(/wiki/File:Servo_driver_hat_bt5.png)

Note: The default discovery time of Raspberry Pi is only 180s. If you want to always discover and pair, you can configure it by modifying the following files.

```
sudo vi /etc/bluetooth/main.conf
```

Find the following two statements and uncomment them.

```
# Defaults to 'BlueZ X.YZ'
#Name = BlueZ

# Default device class. Only the major and minor device class bits are
# considered. Defaults to '0x000000'.
#Class = 0x000100

# How long to stay in discoverable mode before going back to non-discoverable
# The value is in seconds. Default is 180, i.e. 3 minutes.
# 0 = disable timer, i.e. stay discoverable forever
DiscoverableTimeout = 0 ←

# How long to stay in pairable mode before going back to non-discoverable
# The value is in seconds. Default is 0.
# 0 = disable timer, i.e. stay pairable forever
PairableTimeout = 0 ←

# Automatic connection for bonded devices driven by platform/user events.
# If a platform plugin uses this mechanism, automatic connections will be
# enabled during the interval defined below. Initially, this feature
# intends to be used to establish connections to ATT channels. Default is 60.
#AutoConnectTimeout = 60
```

(/wiki/File:Servo_driver_hat_bt6.png)

The program is compatible with python3, just replace python with python3 when running.
For more information on Raspberry Pi wifi and bluetooth remote control, please refer to AlphaBot2:

<http://www.waveshare.com/wiki/AlphaBot2> (<http://www.waveshare.com/wiki/AlphaBot2>)
<https://www.waveshare.com/wiki/AlphaBot2> (<https://www.waveshare.com/wiki/AlphaBot2>)

Using with Jetson Nano

Install Library

- Enable the terminal interface and enter the following command to install the library.

```
sudo apt-get update
sudo apt-get install python3-pip
sudo pip3 install Jetson.GPIO
sudo groupadd -f -r gpio
sudo usermod -a -G gpio your_user_name
sudo cp /opt/nvidia/jetson-gpio/etc/99-gpio.rules /etc/udev/rules.d/
sudo udevadm control --reload-rules && sudo udevadm trigger
```

【Note】your_user_name is the name you used such as waveshare.

- Install I2C

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

- Install image processing library:

```
sudo apt-get install python3-pil  
sudo apt-get install python3-numpy
```

Downlaod the example program & unzip to the specified directory

```
sudo apt-get install p7zip  
wget https://www.waveshare.com/wiki/File:Servo_Driver_HAT.7z  
7zr x Servo_Driver_HAT.7z -r -o./Servo_Driver_HAT  
sudo chmod 777 -R Servo_Driver_HAT  
cd Servo_Driver_HAT/Jetson\ Nano/
```

Run the test demo

- python 2

```
cd python2/  
sudo python test.py
```

- python 3

```
cd python3/  
sudo python3 test.py
```

Resources

Document

- [User Manual \(\[https://www.waveshare.com/w/upload/1/1b/Servo_Driver_HAT_User_Manual_EN.pdf\]\(https://www.waveshare.com/w/upload/1/1b/Servo_Driver_HAT_User_Manual_EN.pdf\)\)](https://www.waveshare.com/w/upload/1/1b/Servo_Driver_HAT_User_Manual_EN.pdf)
- [Schematic \(\[https://www.waveshare.com/w/upload/a/a2/Servo_Driver_HAT_Schematic_.pdf\]\(https://www.waveshare.com/w/upload/a/a2/Servo_Driver_HAT_Schematic_.pdf\)\)](https://www.waveshare.com/w/upload/a/a2/Servo_Driver_HAT_Schematic_.pdf)
- [Demo code \(\[https://www.waveshare.com/wiki/File:Servo_Driver_HAT.7z\]\(https://www.waveshare.com/wiki/File:Servo_Driver_HAT.7z\)\)](https://www.waveshare.com/wiki/File:Servo_Driver_HAT.7z)

Software

- QT software for PC (/wiki/File:AlphaBot_Qt.7z)
- App for Android (/wiki/File:AlphaBot_LITE.rar)
- App source code (/wiki/File:AlphaBot_LITE_Code.7z)

Datasheet

- PCA96 (/wiki/File:PCA96_datasheet.pdf)
- MP1584 (/wiki/File:MP1584.pdf)
- RT9193 (/wiki/File:RT9193.pdf)

Third-Parties Project

- Node-Red project shared by liot2K (<https://github.com/iiot2k/node-red-contrib-iiot-rpi-pca9685>)

FAQ

Question:Can the angle of rotation be precisely controlled?

Answer:

No, this is just entry-level.

Question:Why does it shake?

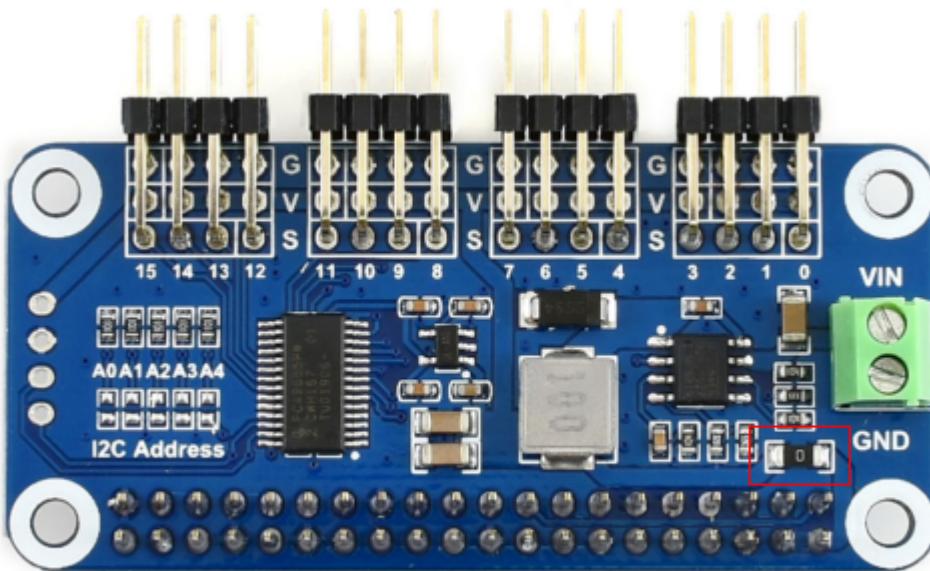
Answer:

It is normal for it to shake slightly. Since the actual angle of the servo is smaller than the minimum physical angle, there will be a current to maintain its angle, which will cause shaking.

Question:Why does the Raspberry Pi restart when controlling high-power servos?

Answer:

Since the Raspberry Pi provides 5V to the module by default, if the control servo power is too large (such as MG996R, DS3120MG), the 5V of the Raspberry Pi will be pulled down. You can remove the onboard 0R resistor and provide 6-12V for the VIN.



(/wiki/File:Servo_Driver_HAT06.png)

Question:Can it be used to control a stepper motor?

Answer:

No, this output is PWM and cannot control the servo.

Question:Servo squeaking and accompanying heat?

Answer:

The SG90 is a servo for beginners. As the potentiometer inside is inaccurate, it will loosen after use for a period, which causes the gear to be inconsistent with the theoretical rotation angle, resulting in noise and heat.

Question:Why do two I2C device addresses appear when scanning an I2C address?**Answer:**

The control chip used corresponds to PCA9685. When powered on, there are two I2C addresses, one is the address configured according to the onboard resistance, the default is 0X40, and the other is 0X70, which is configured by the ALLCALLADR register. You can run the demo again to check the value of the register.

Question:Can two identical Servo Driver HATs be stacked?**Answer:**

It can be stacked, but the resistance of the I2C address needs to be changed. In the I2C Address on the left, solder the default upper resistance to the bottom with a soldering iron. Different combinations generate different I2C address combinations, and there are a total of 32 combinations of 2 to the 5th power.

Support

If you require technical support, please go to the Support (<https://support.waveshare.com/hc/en-us/requests/new>) page and open a ticket.

Retrieved from "https://www.waveshare.com/w/index.php?title=Servo_Driver_HAT&oldid=44809" (https://www.waveshare.com/w/index.php?title=Servo_Driver_HAT&oldid=44809)"