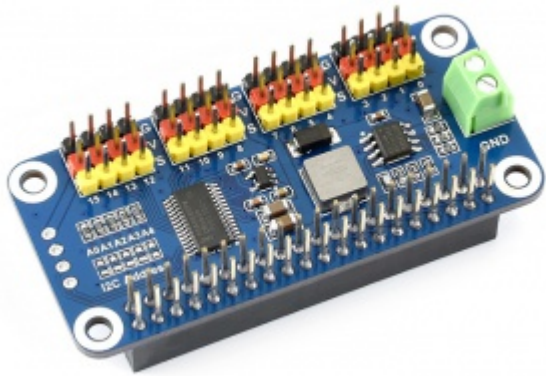


Servo Driver HAT

来自Waveshare Wiki

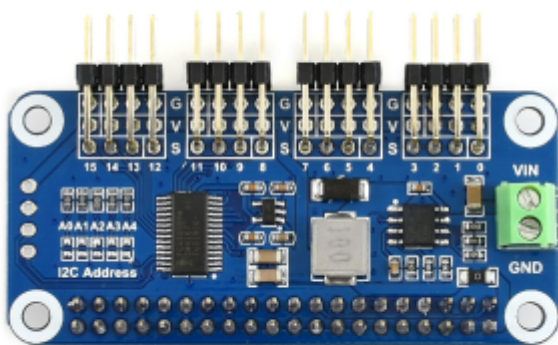
跳转至: [导航](#)、[搜索](#)

Servo Driver HAT



(<https://www.waveshare.net/shop/Servo-Driver->

[HAT.htm](#))



(<https://www.waveshare.net/shop/Servo-Driver-HAT->

[B.htm](#))

板载接口

RPi

I2C

(<https://www.waveshare.net/wiki/%E5%88%86%E7%B1%BB:RPi%E6%8E%A5%E5%8F%A3>) (<https://www.waveshare.net/wiki/%E5%88%86%E7%B1%BB:I2C%E6%8E%A5%E5%8F%A3>)

说明

前言

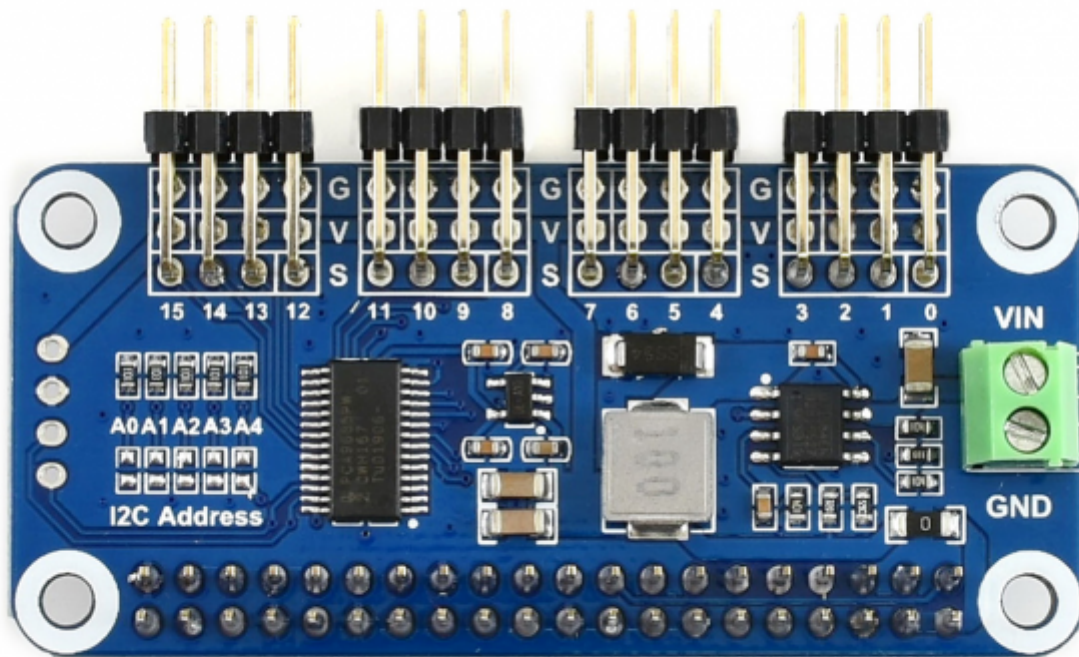
本产品是基于树莓派而设计的PWM/舵机扩展板，通过PCA9685芯片扩展16路舵机控制或者PWM输出，每个通道12位分辨率。通过I2C接口控制，无需占用额外的引脚。板载5V稳压芯片，可接电池供电，最大

输出3A电流。适用于控制机械手臂，以及各种舵机机器人。

产品特性

- 输入电压VIN: 6V~12V
- 舵机电压: 5V
- 逻辑电压: 3.3V
- 驱动芯片: PCA9685
- 控制接口: I2C
- 产品尺寸: 65mm x 30mm
- 固定孔通径: 3.0mm

硬件说明



(/wiki/%E6%96%87%E4%BB%B6:Servo_driver_hat.png)

板子可以从树莓派上取电不需要额外供电。

也可以通过右边绿色端子VIN接电池供电，输入电压范围6V~12V。经过板载的5V稳压芯片输出5V电源给舵机和树莓派供电，最大输出电流3A。

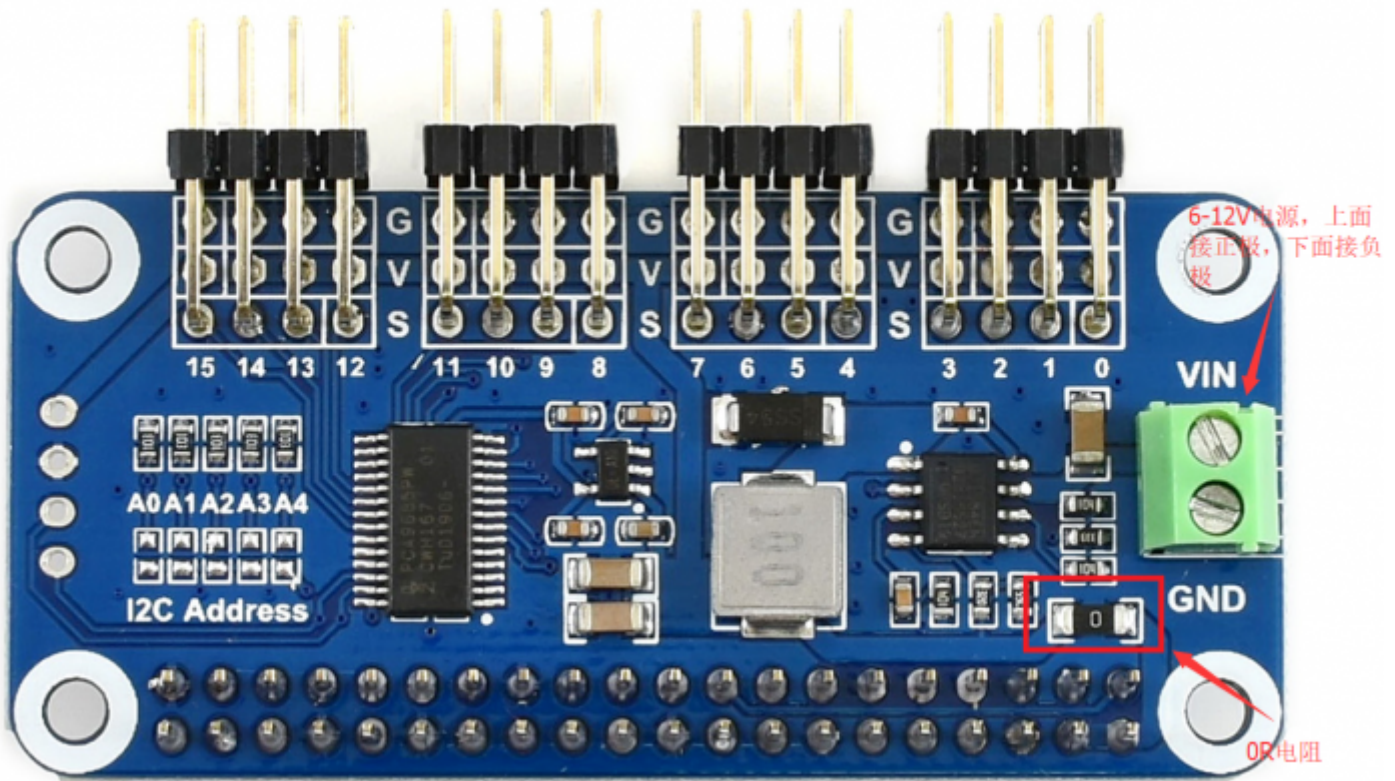
A0~A4可以设置PCA9685芯片的I2C设备地址，可以同时接多个Servo Driver HAT

最上面的排针是舵机接口，黑色排针是GND(大部分舵机对应的是褐色线)，红色排针是5V电源，黄色排针是PWM信号线，有0~15个通道，可以同时接16个舵机。注意舵机线不要接反，否则舵机不会转动。

注意:

如果接大功率的舵机可能会出现供电不足，因为整个板子是5V的供电，这个5V连接着树莓派和舵机供电，功率太大会拉低树莓派5V电源，导致欠压树莓派重启，把板载上的0R电阻去除，右侧绿色VIN端子接外部电源(6-12V)

树莓派请使用独立供电，不要从模块上反向给树莓派供电，这样的供电是不安全的。



(/wiki/%E6%96%87%E4%BB%B6:Servo_driver_hat_0R.png)

树莓派使用

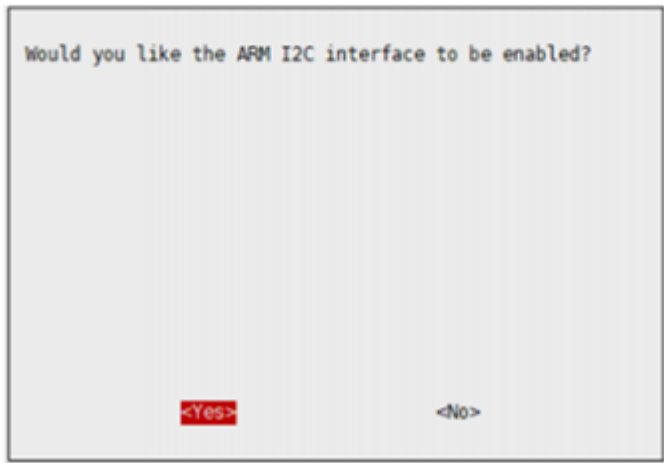
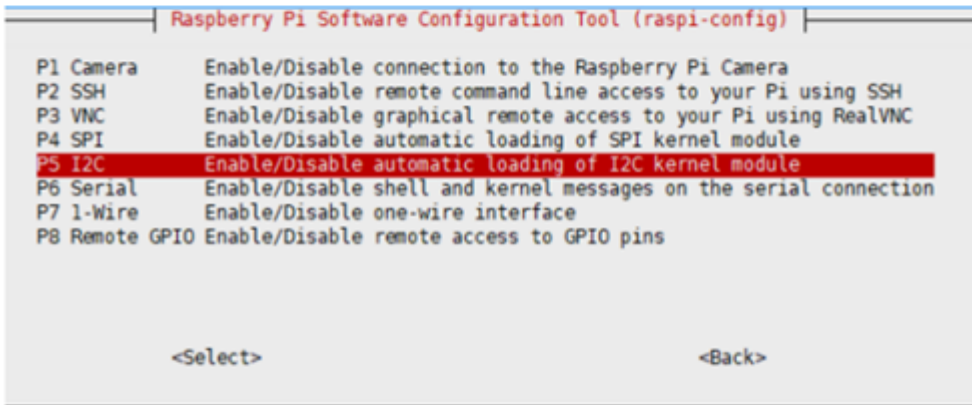
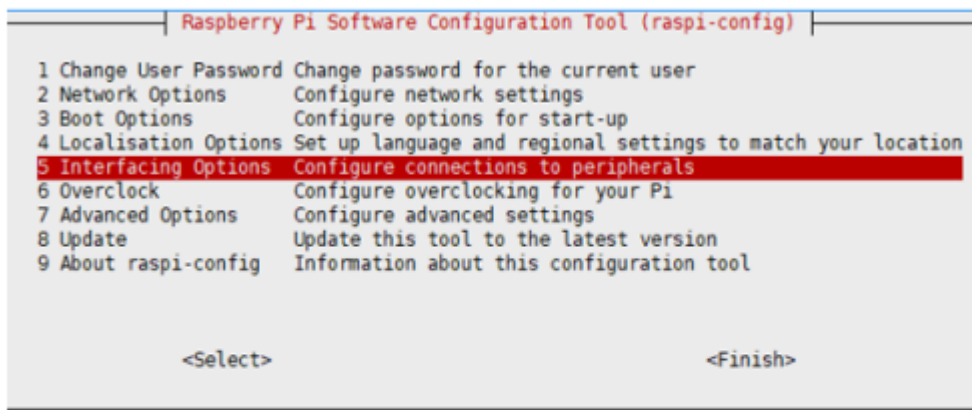
提供python控制，功能包括简单的PCA9685库测试程序，wifi遥控程序以及蓝牙遥控程序。

开启I2C接口

- 打开树莓派终端，输入以下指令进入配置界面

```
sudo raspi-config
```

选择 Interfacing Options -> I2C ->yes 启动 i2C 内核驱动



(/wiki/%E6%96%87%E4%BB%B6:RPI_open_i2c.png)

然后重启树莓派:

```
sudo reboot
```

安装库

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

下载示例程序,并解压到指定目录

```
sudo apt-get install p7zip-full
wget http://www.waveshare.net/w/upload/6/6c/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/
```

运行测试例程

python例程

```
#如果你是python2, 树莓派默认是python2.7
cd python/
sudo python PCA9685.py
#如果你是python3
cd python3/
sudo python3 PCA9685.py
```

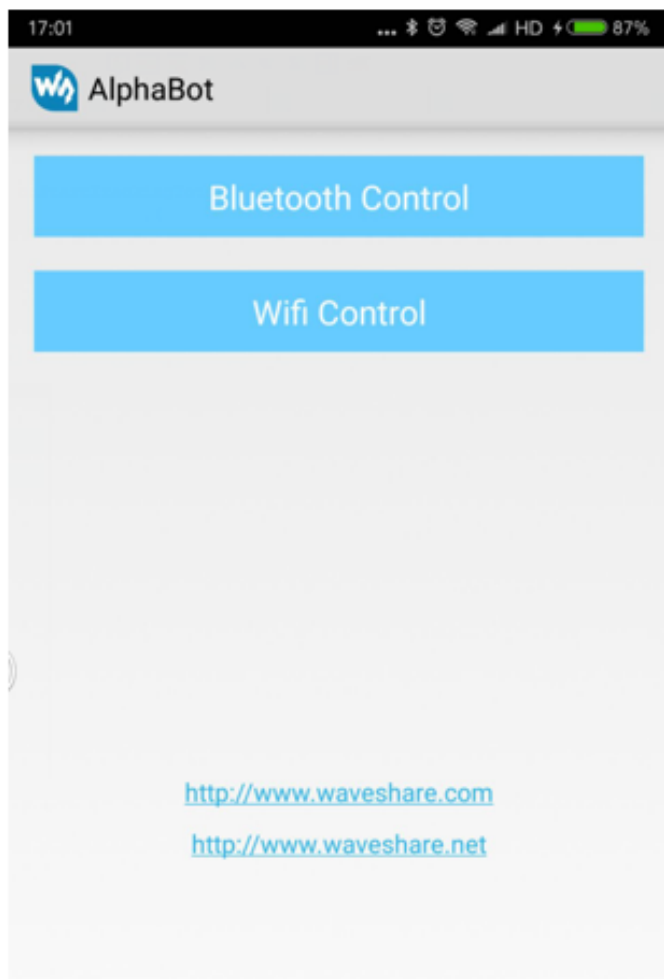
实验现象：将舵机接到0号通道，0号通道的舵机从0度转到180度，然后又从180度转动到0度，不断重复循环。

wifi遥控程序

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

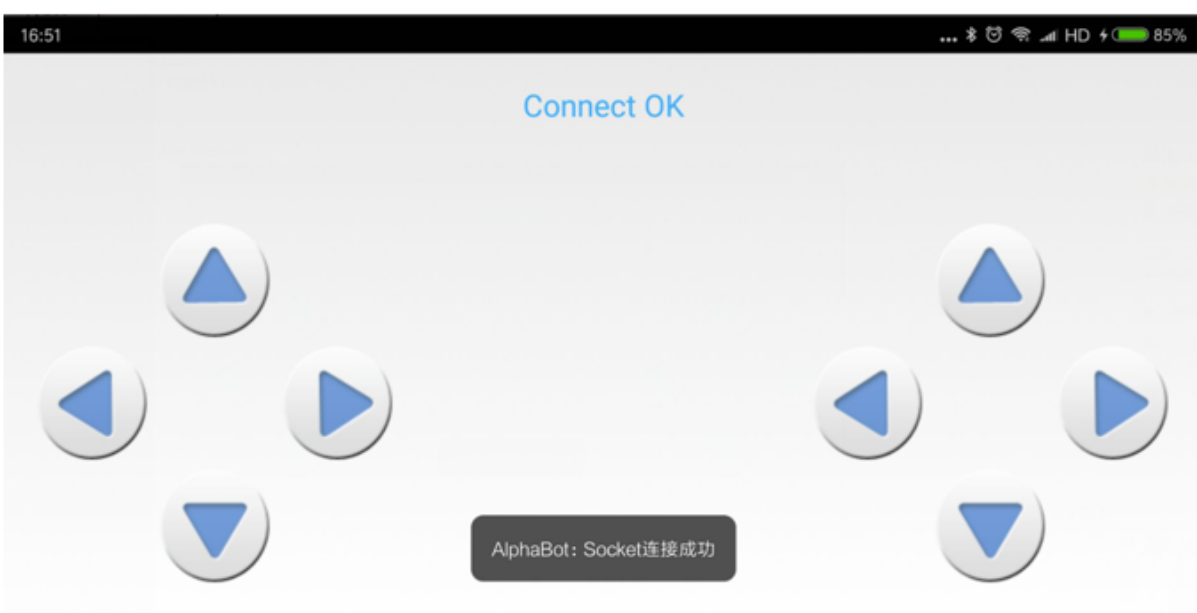
操作与现象:发送端(手机或者电脑)和接收端(树莓派)，需要连接到同一个局域网。程序是通过TCP协议传输数据，程序运行后会显示树莓派IP,服务器端口号是8000.

手机打开APP,选择Wifi遥控，输入对应的IP地址和端口号，点击连接.



(/wiki/%E6%96%87%E4%BB%B6:Servo_driver_hat_wifi.png)

连接成功进入控制界面，点击按键可以控制0~4号通道的舵机正反转。

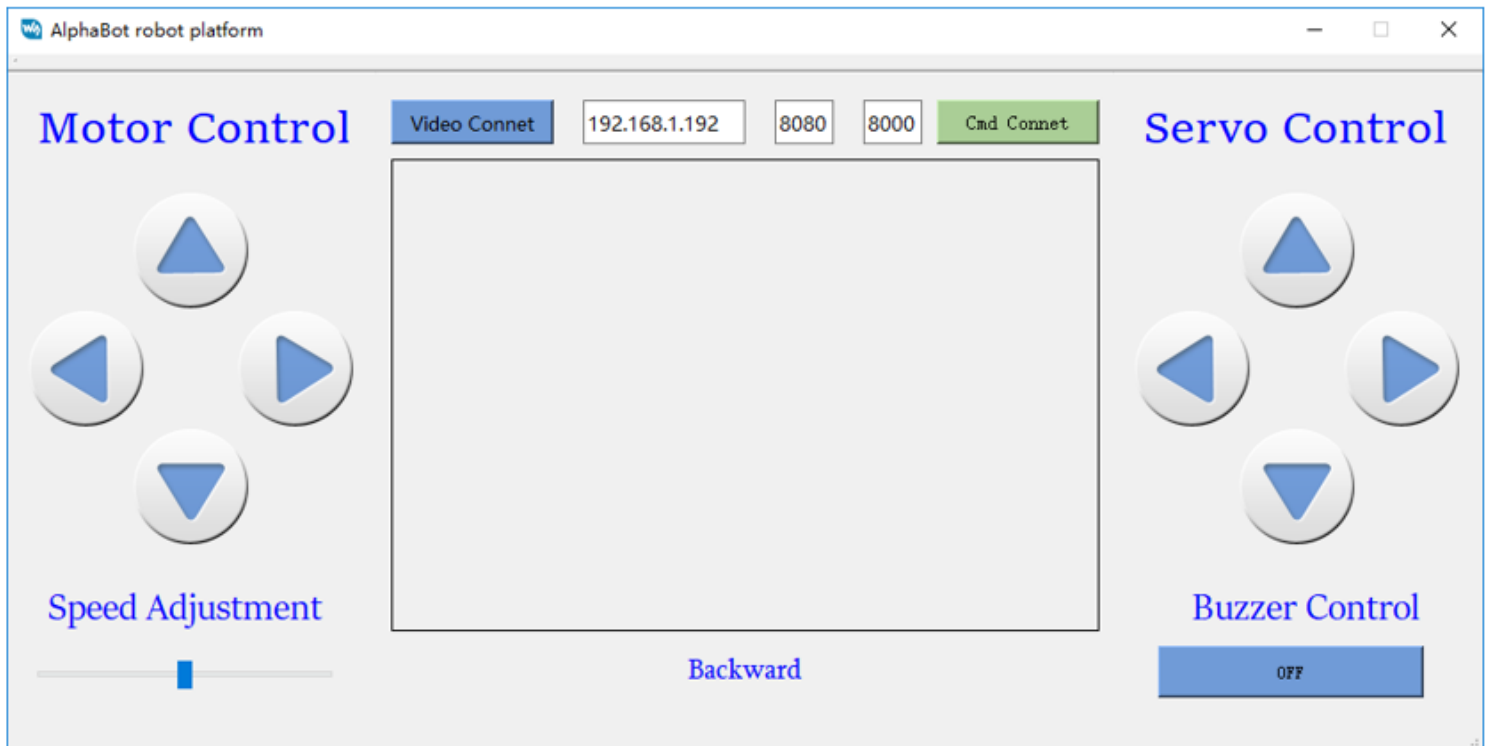


```
pi@raspberrypi:~/Servo_Driver_HAT/python/Wifi-Control $ sudo python main.py
start
192.168.1.192

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

(/wiki/%E6%96%87%E4%BB%B6:Servo_driver_hat_wifi1.png)

也可在电脑上运行Qt软件控制，如下图



(/wiki/%E6%96%87%E4%BB%B6:Servo_driver_hat_wifi2.png)

蓝牙遥控程序

执行:

```
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
```

启动/增加SPP, 开启蓝牙设备

```
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/%E6%96%87%E4%BB%B6:Servo_driver_hat_bt.png)

重启树莓派后, 输入hciconfig命令(类似ifconfig命令)查看蓝牙服务

```
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
```

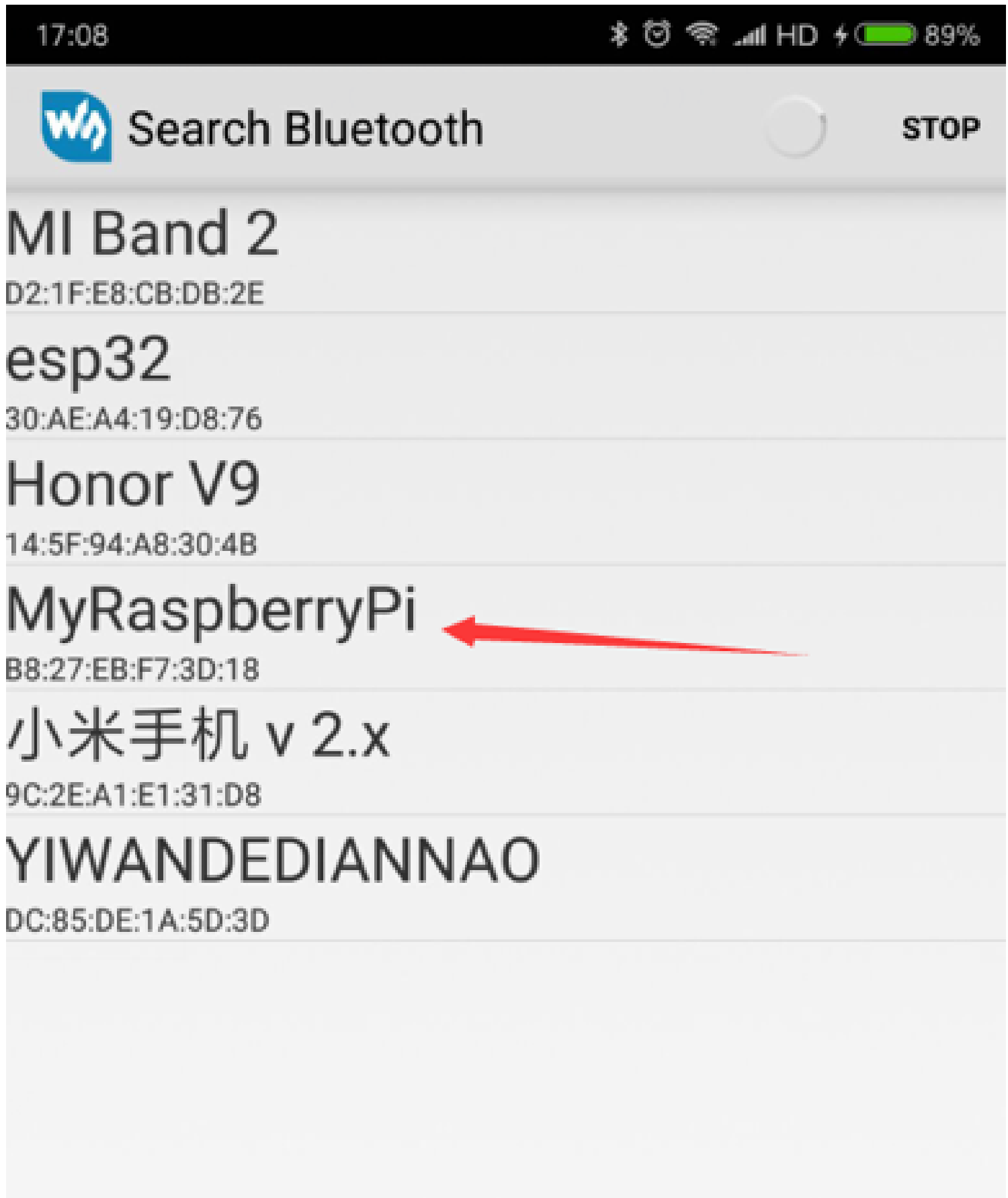
(/wiki/%E6%96%87%E4%BB%B6:Servo_driver_hat_bt2.png)

如果有看到hci0设备则蓝牙已经开启工作。如果没有则没有识别到蓝牙设备

运行如下命令启动程序

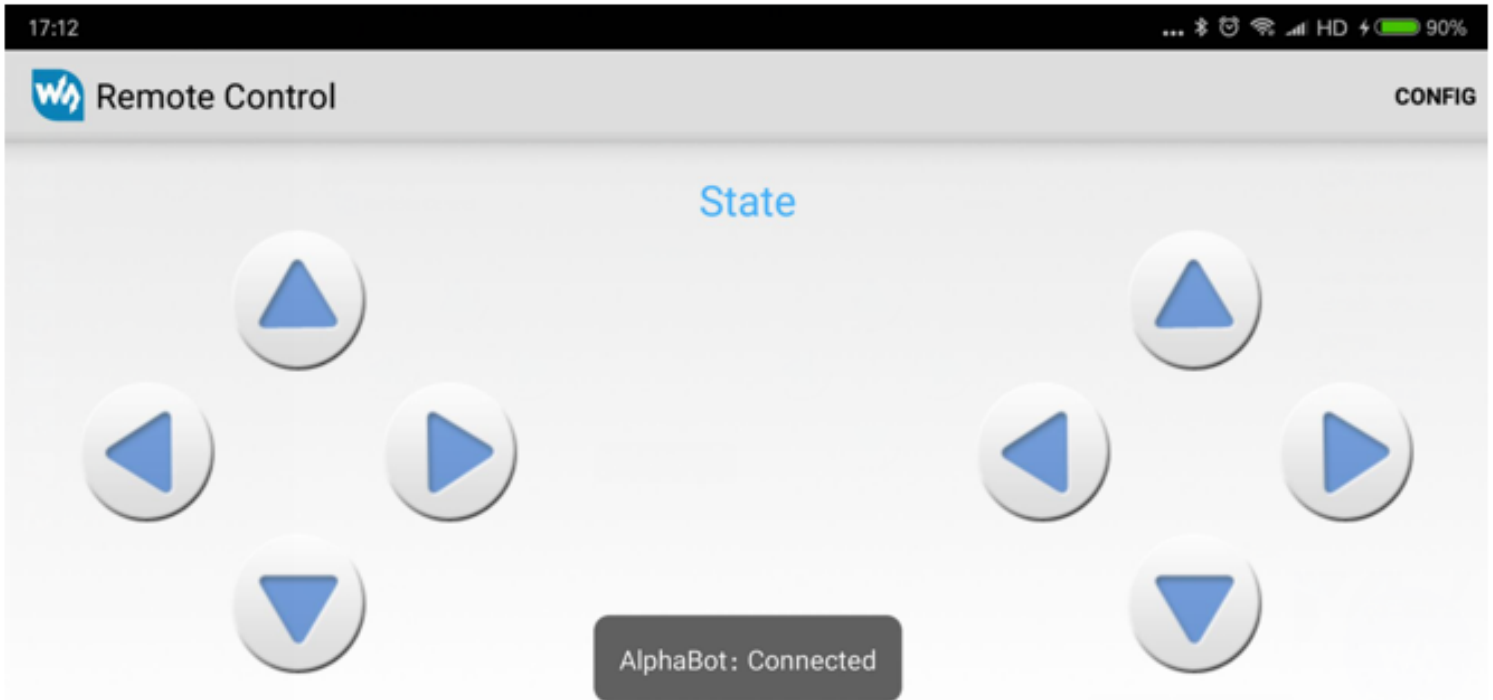
```
cd Servo_Driver_HAT/python/Bluetooth-Control
sudo ./Bluetooth.sh
```


操作与现象:程序运行后会提示等待蓝牙连接,手机App选择蓝牙遥控,点击扫描,发现raspberrypi设备并连接。



(/wiki/%E6%96%87%E4%BB%B6:Servo_driver_hat_bt3.png)

连接程序后进入控制界面,点击按键可以遥控0~4通道的舵机正反转了。



(/wiki/%E6%96%87%E4%BB%B6:Servo_driver_hat_bt4.png)

注意:如果接收到的命令不对,可能需要在APP中config一下对应按键按下和松开时发送的命令。

```
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/%E6%96%87%E4%BB%B6:Servo_driver_hat_bt5.png)

注意:树莓派默认是只有180s的发现时间,如果想一直可以发现和配对,如果通过修改下面的文件配置。

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

找到下面两个语句并去掉注释。

```
# 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/%E6%96%87%E4%BB%B6:Servo_driver_hat_bt6.png)

程序兼容python3 ,运行的时候将python替换成python3即可。

更多关于树莓派wifi和蓝牙遥控可以参考AlphaBot2:

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

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

Jetson nano

提供python控制

安装库

安装函数库

- 打开终端界面，输入以下指令安装相应的函数库

```
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
```

【注意】 your_user_name 是你使用的用户名，比如说 waveshare

- 安装I2C

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

- 安装图像处理库:

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

下载示例程序,并解压到指定目录

```
sudo apt-get install p7zip  
wget http://www.waveshare.net/w/upload/6/6c/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/
```

运行测试例程

- python2

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

- python3

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

资料

文档

- 原理图 (https://www.waveshare.net/w/upload/a/a2/Servo_Driver_HAT_Schematic_.pdf)

程序

- 示例程序 (https://www.waveshare.net/w/upload/6/6c/Servo_Driver_HAT.7z)

软件

- Windows Qt 客户端 (https://www.waveshare.net/w/upload/a/a9/AlphaBot_Qt.7z)
- 安卓App控制客户端 (https://www.waveshare.net/w/upload/8/8f/AlphaBot_LITE.apk)

- 安卓App控制客户端源码 (https://www.waveshare.net/w/upload/3/39/AlphaBot_LITE_Code.7z)

数据手册

- PCA9685 (https://www.waveshare.net/w/upload/6/68/PCA96_datasheet.pdf)
- MP1584 (<https://www.waveshare.net/w/upload/d/d3/MP1584.pdf>)
- RT9193 (<https://www.waveshare.net/w/upload/f/f6/RT9193.pdf>)

FAQ

问题： 能否精密控制转动角度？

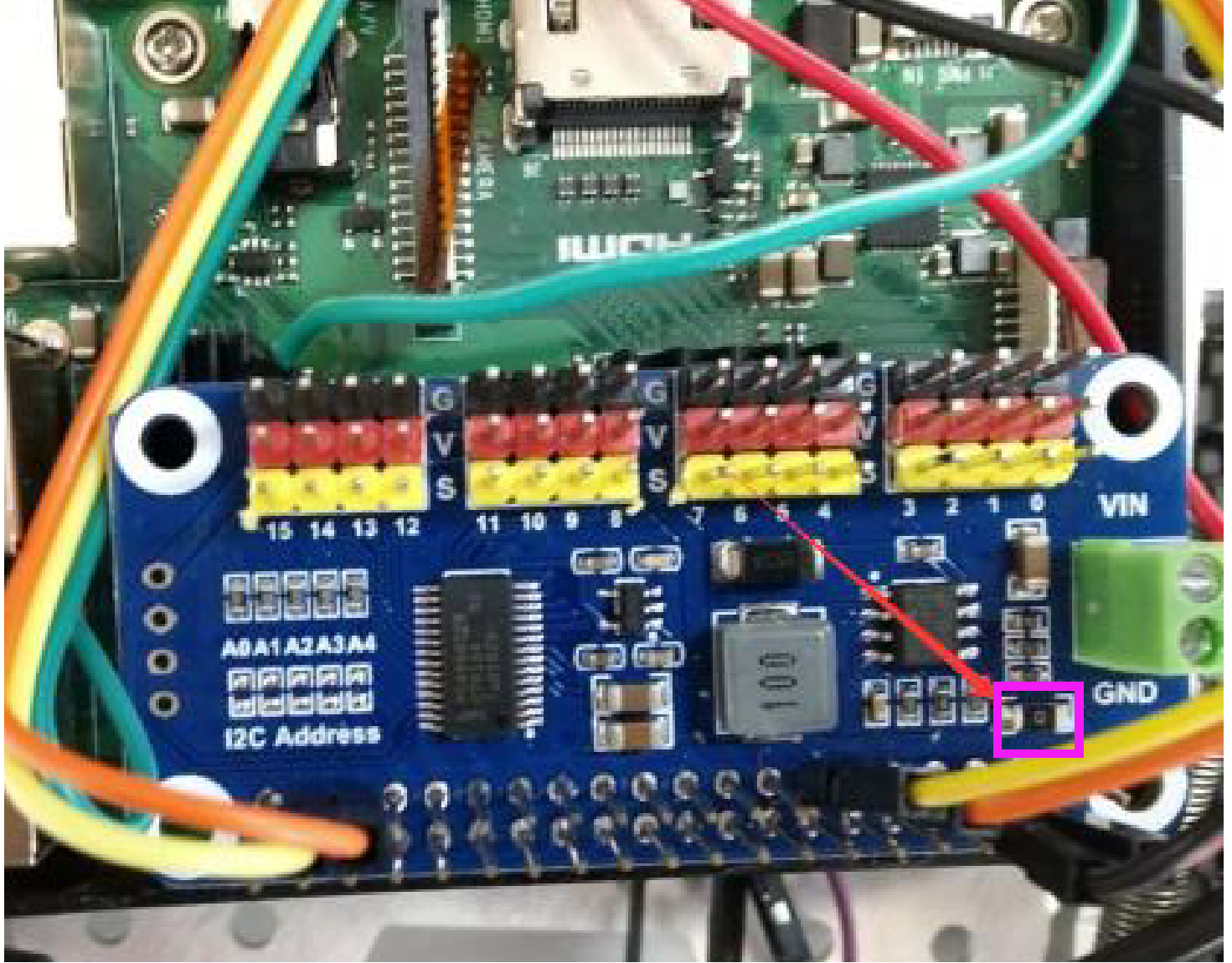
- 无法做到精密控制，这个只是入门级的。

问题： 为什么抖动？

- 轻微的抖动是正常的，由于舵机转动的实际角度小于舵机转动的最小物理角度，此时会有电流维持他的角度，就会引起抖动。

问题： 控制大功率舵机为什么树莓派重启？

- 由于默认是使用树莓派的5V给模块供电，如果控制的舵机功率太大（栗子：MG996R、DS3120MG），会拉低树莓派的5V，可以将板上的这个0R电阻移除。然后VIN端子供6-12v的电压



(/wiki/%E6%96%87%E4%BB%B6:Servo_driver_hat_faq1.png)

问题： 能否用来控制步进电机？

- 不可以，这个输出的是PWM，无法控制步进电机。

问题： 舵机吱吱作响，并伴随发热？

- SG90是入门级别舵机，主要是里面的电位器不准的原因，使用一段时间就发生了松动，于是就导致了齿轮与理论上转的角度不一致，导致响声并发热。

问题： 为什么扫描I2C地址会出现两个I2C设备地址？

- 使用的控制芯片对应的是PCA9685,在上电的时候,是有两个I2C地址的,一个是根据板载的电阻配置的地址,默认的是0X40,还有一个是0X70这个是ALLCALLADR寄存器配置出来的I2C地址,运行一遍例程即可清楚寄存器的值。

问题：两个相同的Servo Driver HAT能否叠加使用？

- 可以叠加,但是需要改变I2C地址的电阻,在左边的I2C Address出通过烙铁把默认在上方的电阻焊接到下面,不同的组合有不同的I2C地址组合,总共有2的5次方32个组合。