

## Chapter4: Raspberry Pi uses PCA9685 to drive the motor

### 1. Introduction of PCA9685, how to use I2C

The servo is controlled by three pins, VCC, GND and IO port(singal). The Raspberry Pi has only 29 gpio pins, and each servo requires a singal pin, which is a waste of resources. The PCA9685 is a drive board for multi-channel pwm control. It uses i2c communication, and only needs a few i2c lines to control 16 channels of pwm. Both the pulse period and the duty cycle are controllable.

First, we need to input:

```
sudo raspi-config
```

Then, we need to select “Interfacing Options”-“P5 I2C”—“yes”—“ok” and restart Raspberry Pi.

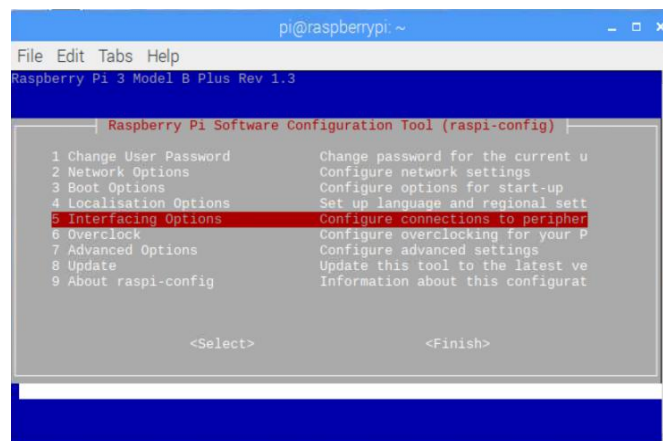


Figure 1-1 sudo raspi-config -> Interfacing Options

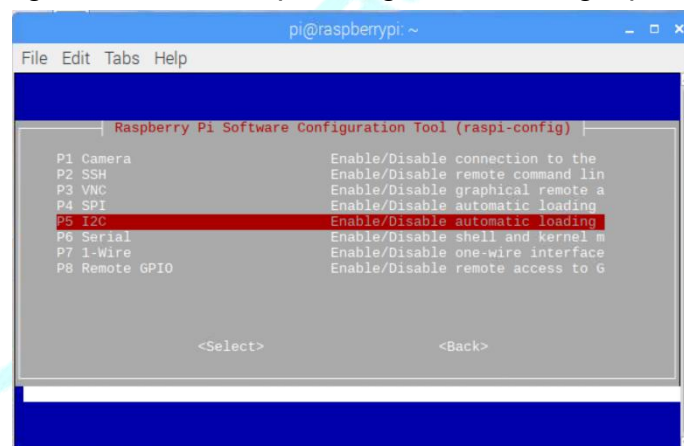


Figure 1-2

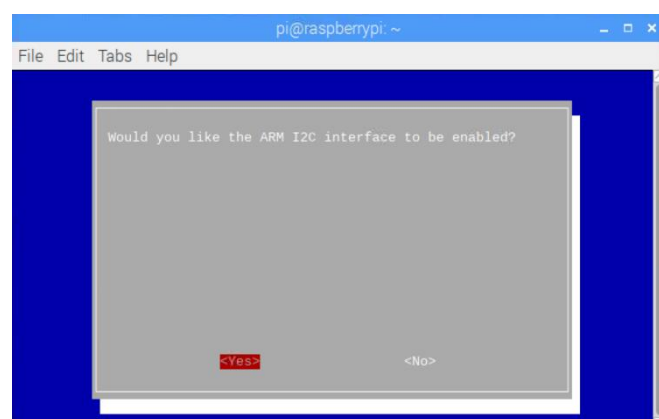
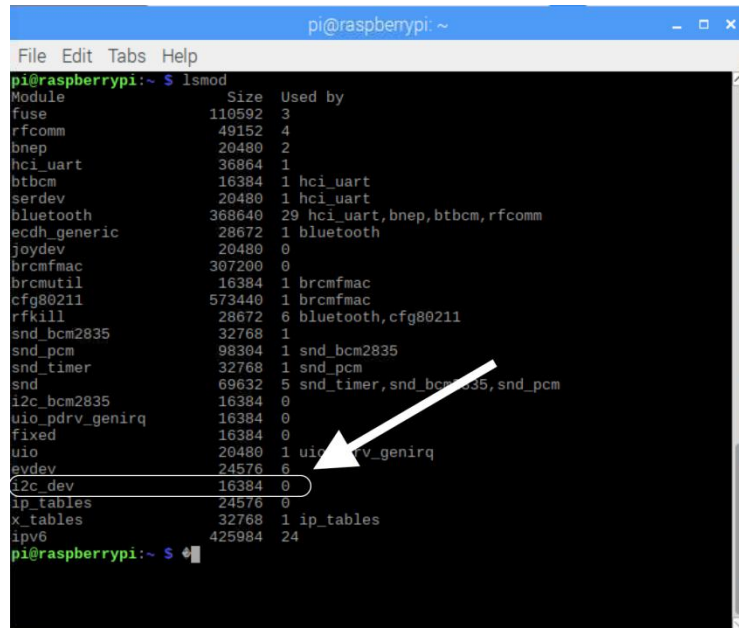


Figure 1-3

After restarting the Raspberry Pi, we need to input **lsmod** to see if i2c starts successfully. As shown in the figure 1-4 below.



```

pi@raspberrypi: ~
File Edit Tabs Help
pi@raspberrypi:~$ lsmod
Module                  Size      Used by
fuse                    110592    3
rfcomm                  49152     4
bnep                    20480     2
hci_uart                36864     1
btbcm                   16384     1 hci_uart
serdev                  20480     1 hci_uart
bluetooth               368640    29 hci_uart, bnep, btbcm, rfcomm
ecdh_generic            28672     1 bluetooth
joydev                  20480     0
brcmfmac                307200    0
brcmutil                16384     1 brcmfmac
cfg80211                573440    1 brcmfmac
rfkill                  28672     6 bluetooth, cfg80211
snd_bcm2835              32768     1
snd_pcm                 98304     1 snd_bcm2835
snd_timer               32768     1 snd_pcm
snd                      69632     5 snd_timer, snd_bcm2835, snd_pcm
i2c_bcm2835              16384     0
uio_pdrv_genirq          16384     0
fixed                   16384     0
uio                      20480     1 uio_pdrv_genirq
evdev                   24576     6
i2c_dev                  16384     0
ip_tables               24576     0
x_tables                32768     1 ip_tables
ip6v6                   425984    24
pi@raspberrypi:~$

```

Figure 1-4

## 2. Download the Adafruit-PCA9685 driver and use the BST-AI expansion board

Terminal input:

```

sudo apt-get update
sudo apt-get install build-essential python-pip python-dev python-smbus git
git clone https://github.com/adafruit/Adafruit_Python_PCA9685.git

```

After the download is complete, enter the generated boot driver folder

The terminal inputs in turn:

```

cd Adafruit_Python_PCA9685
sudo python setup.py install

```

Detail:

<https://cdn-learn.adafruit.com/downloads/pdf/adafruit-16-channel-servo-driver-with-raspberry-pi.pdf>

Next, we need to connect the BST-AI expansion board insert the 40pin GPIO pin of the Raspberry Pi, connect the audio interface of the Raspberry Pi and the expansion board by the dual 3.5mm audio cable, and insert the speaker into the speaker interface. The servo that controls the up and down rotation is inserted in S5, and the servo that controls the left and right rotation is inserted in S6. Finally installed the battery. As shown in the figure 1-5 below.

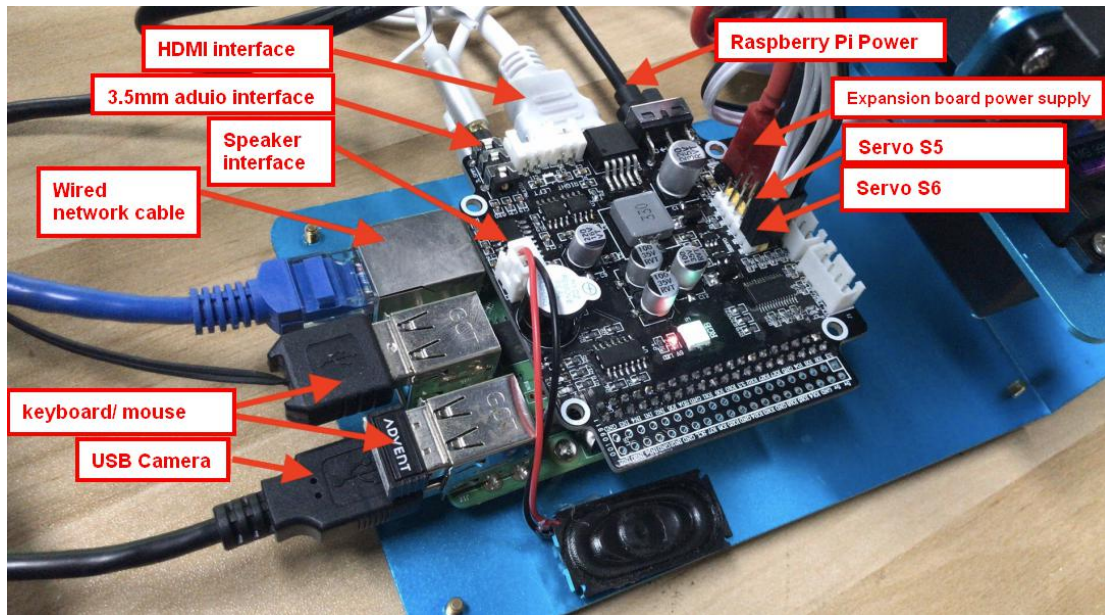


Figure 1-5

Note:

1. After the expansion version is installed, turn on the power switch, you can see the on-board RGB small lights will light up. We must use the battery we provide for the following experiments.
2. Whether it is the Raspberry Pi, the camera or the BST-AI board, they are belongs to the integrated circuit board. We should always pay attention to the protection of the components in the learning project. Do not touch the board and components with wet hands.

Next, we need to check the IIC address occupied by the BST-AI.

Terminal input:

```
i2cdetect -y -a 1
```

As shown in the figure 1-6 below, we can know that i2c address of this BST-AI board is 0x41.

```

pi@raspberrypi: ~
File Edit Tabs Help
pi@raspberrypi:~$ i2cdetect -y -a 1
 0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00: 00  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
10:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
20:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
30:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
40:  --  41  --  --  --  --  --  --  --  --  --  --  --  --  --  --
50:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
60:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
70: 70  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
pi@raspberrypi:~$

```

Figure 1-7

### 3.About code

The following program uses the PCA9685 to control two DC motors by PWM. We can control the rotation speed of the motor by changing the PWM pulse.

```

1  #!/usr/bin/env python2
2  # -*- coding: utf-8 -*-
3  """
4  Created on Thu Jan 10 07:51:54
5  Shenzhen Yahboom Tech
6
7  @author: LONGFU SUN
8  """
9
10 from __future__ import division
11 import time
12 import Adafruit_PCA9685
13
14 pwm = Adafruit_PCA9685.PCA9685()
15
16 mA1=8
17 mA2=9
18 mB1=10
19 mB2=11
20
21 pwm.set_pwm_freq(50)
22
23
24 #Maximum frequency is 4096, 4096, the highest speed, 1024, 2048, 3072
25 #forward, two servos rotate in the same direction
26 def forward():
27     pwm.set_pwm(mA1,0,1024)
28     pwm.set_pwm(mA2,0,0)
29     pwm.set_pwm(mB1,0,1024)
30     pwm.set_pwm(mB2,0,0)
31     time.sleep(1)
32     #back, two servo rotate in the opposite direction of forward()
33 def back():
34     pwm.set_pwm(mA2,0,1024)
35     pwm.set_pwm(mA1,0,0)
36     pwm.set_pwm(mB2,0,1024)
37     pwm.set_pwm(mB1,0,0)
38     time.sleep(1)
39     #Spin left, give the right servo forward speed, and give the left servo backward speed.
40 def spin_left():
41     pwm.set_pwm(mA1,0,0)
42     pwm.set_pwm(mA2,0,1024)
43     pwm.set_pwm(mB1,0,1024)
44     pwm.set_pwm(mB2,0,0)
45     time.sleep(1)
46     #Spin right, give the left servo forward speed, and give the right servo backward speed.
47 def spin_right():
48     pwm.set_pwm(mA1,0,1024)
49     pwm.set_pwm(mA2,0,0)
50     pwm.set_pwm(mB1,0,0)
51     pwm.set_pwm(mB2,0,1024)
52     time.sleep(1)
53     #Turn left, change the servo angle by the frequency difference
54 def left():
55     pwm.set_pwm(mA1,0,512)
56     pwm.set_pwm(mA2,0,0)
57     pwm.set_pwm(mB1,0,1024)
58     pwm.set_pwm(mB2,0,0)
59     time.sleep(1)
60     #Turn right, change the servo angle by the frequency difference
61 def right():
62     pwm.set_pwm(mA1,0,1024)
63     pwm.set_pwm(mA2,0,0)
64     pwm.set_pwm(mB1,0,512)
65     pwm.set_pwm(mB2,0,0)
66     time.sleep(1)
67 def stop():
68     pwm.set_pwm(mA1,0,0)
69     pwm.set_pwm(mA2,0,0)
70     pwm.set_pwm(mB1,0,0)
71     pwm.set_pwm(mB2,0,0)
72     time.sleep(1)
73 #Delay
74 for i in range(0,10):
75     forward()
76     time.sleep(1)
77     back()
78     time.sleep(1)
79     spin_left()
80     time.sleep(1)
81     spin_right()
82     time.sleep(1)
83     stop()
84
85
86 print('Moving servo on channel 0, press Ctrl-C to quit...')

```