Chapter6: Raspberry Pi uses PCA9685 to drive the servo

1. Introduction of PCA9685, how to use I2C

wiringPi	BCM	Funtion	Physical pin		Funtion	BCM	wiringPi
		3.3V	1	2	5V		
8	2	SDA.1	3	4	5V		
9	3	SCL.1	5	6	GND		
7	4	GPIO.7	7	8	TXD	14	15
		GND	9	10	RXD	15	16
0	17	GPIO.0	11	12	GPIO.1	18	1
2	27	GPIO.2	13	14	GND		
3	22	GPIO.3	15	16	GPIO.4	23	4
		3.3V	17	18	GPIO.5	24	5
12	10	MOSI	19	20	GND		
13	9	MISO	21	22	GPIO.6	25	6
14	11	SCLK	23	24	CE0	8	10
		GND	25	26	CE1	7	11
30	0	SDA.0	27	28	SCL.0	1	31
21	5	GPIO.21	29	30	GND		
22	6	GPIO.22	31	32	GPIO.26	12	26
23	13	GPIO.23	33	34	GND		
24	19	GPIO.24	35	36	GPIO.27	16	27
25	26	GPIO.25	37	38	GPIO.28	20	28
		GND	39	40	GPIO.29	21	29

Figure 1-1 Raspberry Pi Pin table

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.

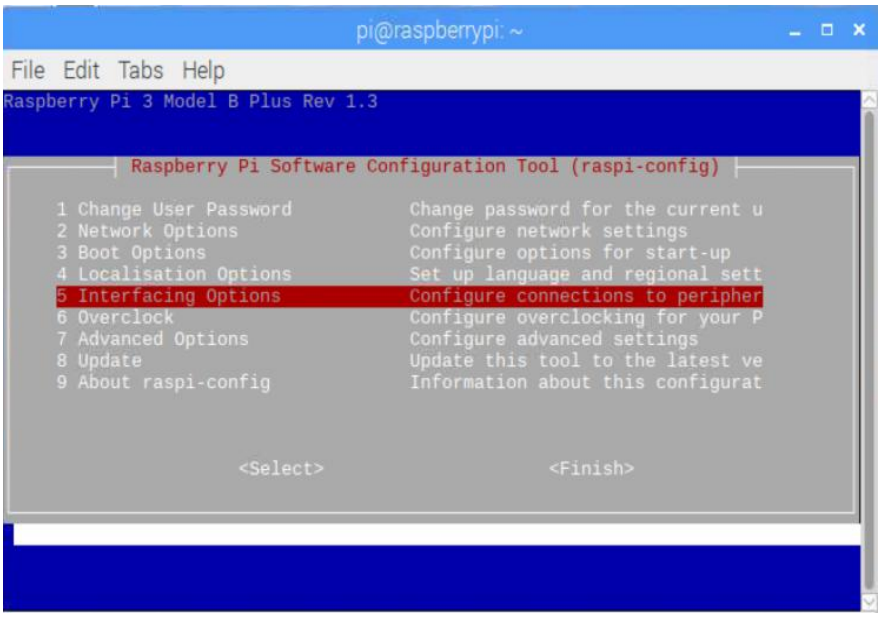


Figure 1-2 sudo raspi-config -> Interfacing Options

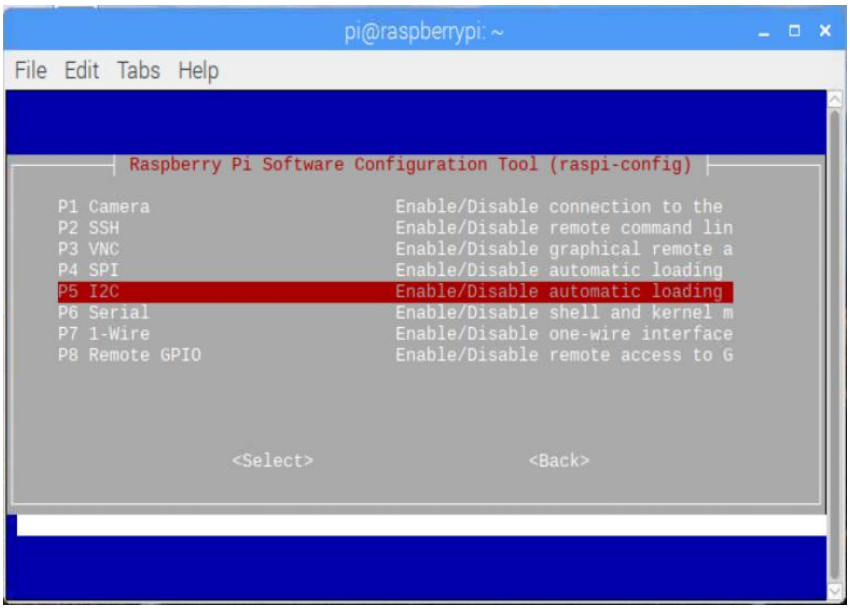


Figure 1-3

Figure 1-4

After restarting the Raspberry Pi, we need to input **lsmod** to see if i2c starts successfully. As shown in the figure 1-5 below.

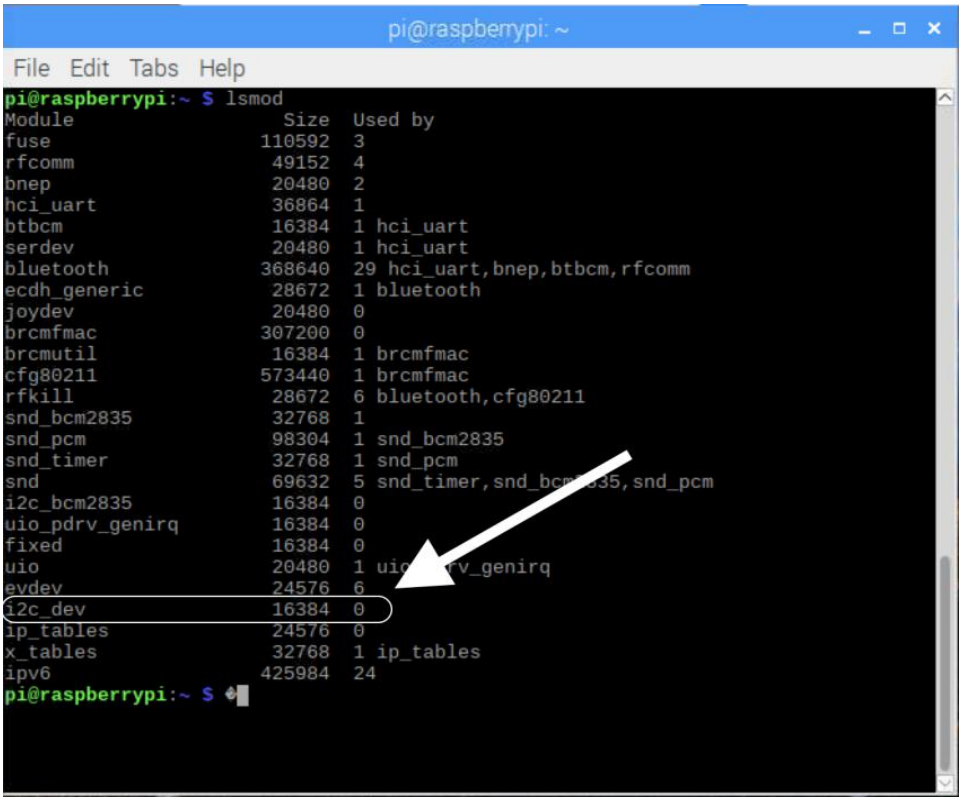


Figure 1-5

We can download i2c-tools, which monitors hardware usage and failures

Terminal input:

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

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 1-6 below.

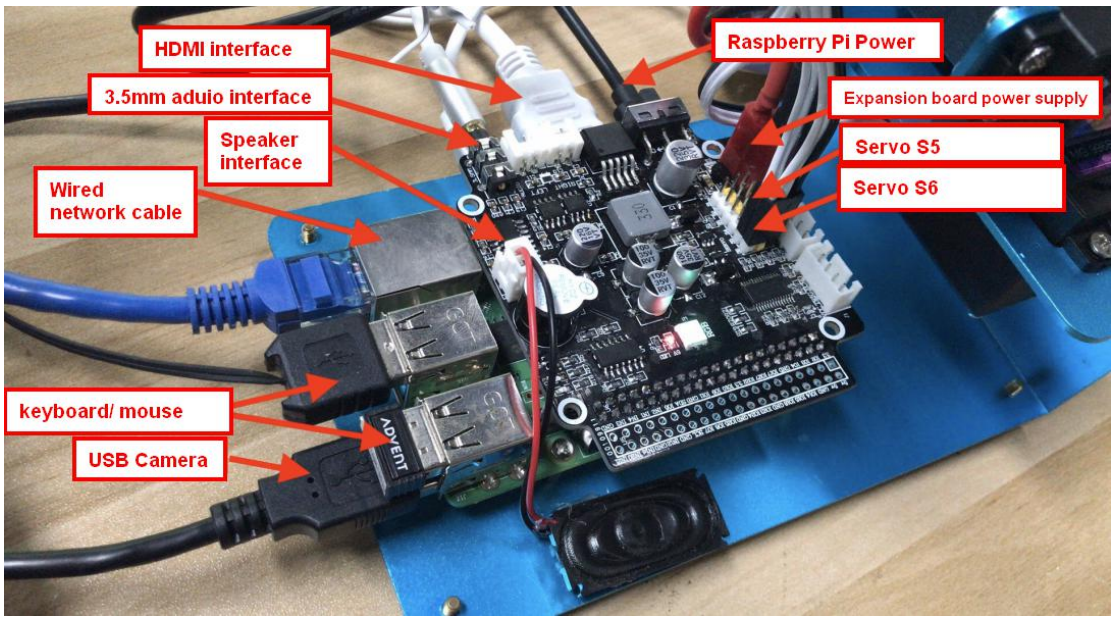


Figure 1-6

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 1-7 below, we can know that i2c address of this BST-AI board is 0x41.

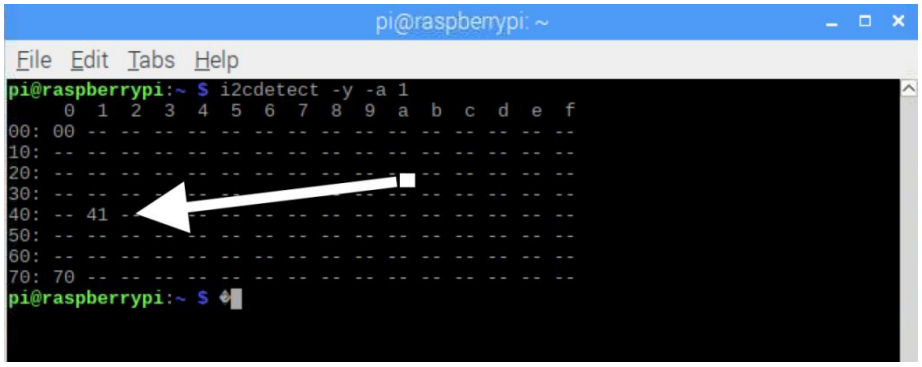
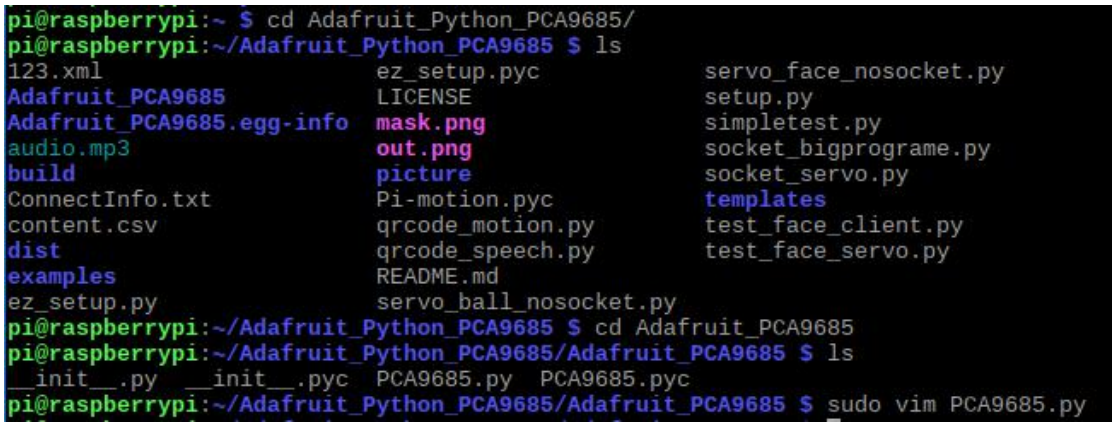


Figure 1-7

We need to enter **pi/Adafruit_Python_PCA9685/Adafruit_PCA9685/PCA9685.py** Change the value of parameter PCA9685_ADDRESS to 0x41. As shown in the figure 1-8 below.



File Edit Tabs Help

```
# Copyright (c) 2016 Adafruit Industries
# Author: Tony DiCola
#
# Permission is hereby granted, free of charge, to any person obtaining a copy
# of this software and associated documentation files (the "Software"), to deal
# in the Software without restriction, including without limitation the rights
# to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
# copies of the Software, and to permit persons to whom the Software is
# furnished to do so, subject to the following conditions:
#
# The above copyright notice and this permission notice shall be included in
# all copies or substantial portions of the Software.
#
# THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
# IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
# FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
# AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
# LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
# OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
# THE SOFTWARE.
from __future__ import division
import logging
import time
import math


# Registers/etc:
PCA9685_ADDRESS    = 0x41
MODE1              = 0x00
MODE2              = 0x01
SUBADR1            = 0x02
SUBADR2            = 0x03
SUBADR3            = 0x04
PRESCALE           = 0xFE
LED0_ON_L          = 0x06
LED0_ON_H          = 0x07
LED0_OFF_L         = 0x08
LED0_OFF_H         = 0x09
ALL_LED_ON_L       = 0xFA
ALL_LED_ON_H       = 0xFB
ALL_LED_OFF_L      = 0xFC
ALL_LED_OFF_H      = 0xFD

# Bits:
RESTART            = 0x80
SLEEP              = 0x10
```

Figure 1-8