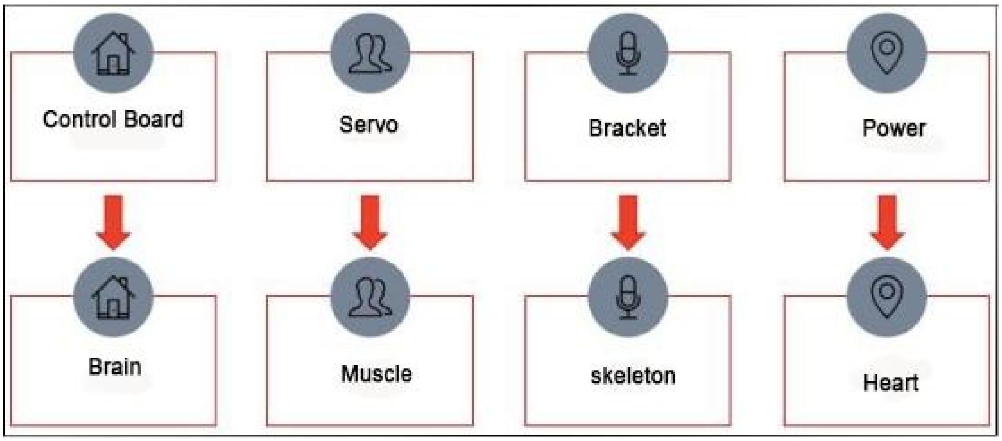
Lesson 3 Control Pan-tilt Rotation

Robot is the machine that performs work automatically. It can be commanded by human and runs pre-programmed programs or act according to principles developed with artificial intelligence technology.

Robot system mainly consists of control board, servo, brackets, power supply, sensors (camera and other electronic components) and software. Among them, the sensors of different robots are different. From the perspective of the four major structures on the hardware terminal of each robot, the relationship between them can be understood by referring to the following figure.

The servo of this Raspberry Pi humanoid robot is composed of sixteen LX-824HV serial bus servos and two LED-01M PWM digital servos. Let's start learning!



Robot structure

1. What is servo?

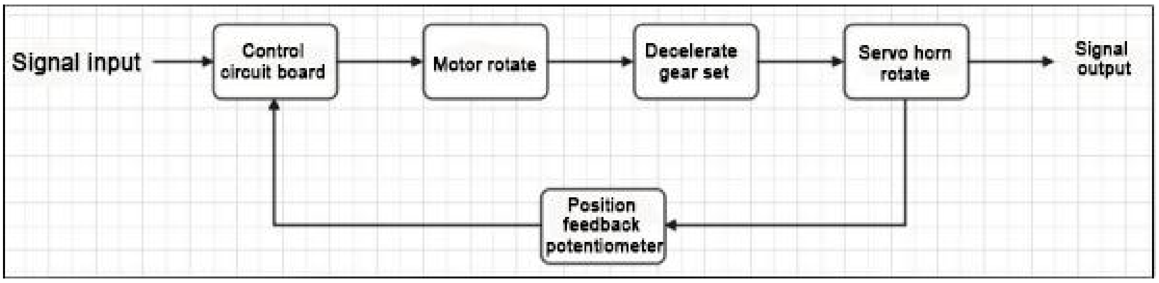
We learned to program and call action groups before and both are realized by adjusting servo.What is the servo?

Servo is an electronic device with an output shaft that can be positioned to a specific angular position by sending a coded signal. The servo will maintain the angular position of the shaft as long as there is an encoded signal in the input line. In real life, servos are usually applicable to robotics projects, remote control cars, etc.

2. The Composition and Working Principle of Servo

The servo is mainly composed of control circuit board, motor, gear set, servo horn, position feedback potentiometer. The control circuit board is the core of servo, which is responsible for receiving and sending signals to control the operation of the whole servo. The Motor can supply power. The servo horn is connected to other devices to drive the device to rotate, and then the potentiometer will transmits feedback signals back to the control board.

The basic working process of servo is as follows:



Servo Working Process

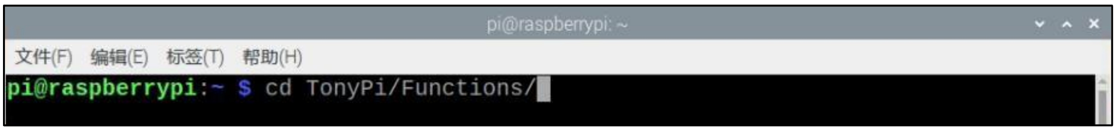
Controller receives signals from signal line to control the rotation of motor. The motor will drive the gear set. After the gear set slows down, it works to servo horn. When servo horn rotates, it will drive the feedback potentiometer connected to the output shaft, and then the potentiometer will transmit the voltage feedback signal to the control board. Then the control circuit board adjust the rotation direction and speed of motor so as to achieve the purpose of rotating to a certain angle.

3. Operation Steps

1) Turn on TonyPi Pro and connect to VNC.

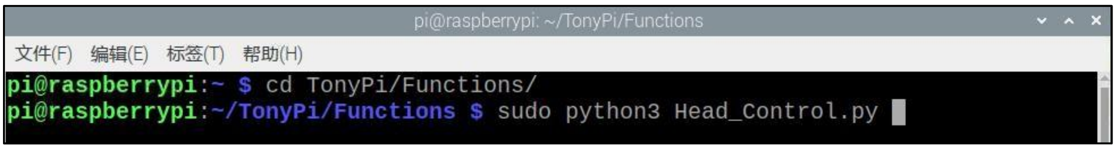
2) Press “Ctrl+Alt+T” or click >_ icon in the upper left corner to open LX terminal.

3) Enter “cd TonyPi/Functions/” command, and then press “Enter” to come to the category of games programmings.



```
pi@raspberrypi:~ $ cd TonyPi/Functions/
```

4) Enter “sudo python3 Head_Control.py” command, and then press “Enter” to start game.

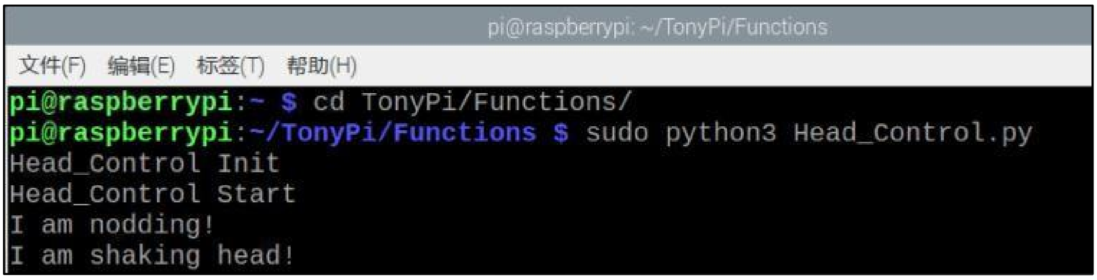

pi@raspberrypi: ~/TonyPi/Functions
文件(F) 编辑(E) 标签(T) 帮助(H)

```
pi@raspberrypi:~ $ cd TonyPi/Functions/
pi@raspberrypi:~/TonyPi/Functions $ sudo python3 Head_Control.py
```

5) If you want to exit the game programming, press “Ctrl+C” in the LX terminal interface. If the exit fails, please try it few more times.

4. Project Outcome

After starting the game, the robot will nod first and shake its head later, and then print the corresponding prompt in terminal.


pi@raspberrypi: ~/TonyPi/Functions
文件(F) 编辑(E) 标签(T) 帮助(H)

```
pi@raspberrypi:~ $ cd TonyPi/Functions/
pi@raspberrypi:~/TonyPi/Functions $ sudo python3 Head_Control.py
Head_Control Init
Head_Control Start
I am nodding!
I am shaking head!
```

5. Program Analysis

The program file “Board.py” for Raspberry Pi to control servo is saved in the “HiwonderSDK” folder. We just need to call “setPWMServoPulse” function to control servo to rotate. This function has three parameters in total, which are servo number, PWM signal value and rotation time.

The number of servo that controls robot’s head to move up and down is 1 and the number of servo that controls robot’s head to move right or left is 2. The PWM signal value determines the position that servo reaches. Since the rotation angle of pan-tilt ranges from 0°to 180°，corresponding to the pulse width is 500 to 2500, PWM value must be within the range of 500 to 2500, that is, the middle position is 1500. At this time, the robot will look straight ahead.

If you want the robot to nod, you only need to change the parameters of the No. 1 servo. For example, If want the robot to look up, you can set the PWM value to 1800.

```
Board.setPWMServoPulse(1, 1800, 200)
```

If the robot bows its head, PWM value is set to 1200.

```
Board.setPWMServoPulse(1, 1200, 200)
```

Perform nodding action by repeating these two actions few more times. The following is the complete program:

```
def nod_head():
    Board.setPWMServoPulse(1, 1800, 200) # look up
    time.sleep(0.2)
    Board.setPWMServoPulse(1, 1200, 200)#look down
    time.sleep(0.2)
    Board.setPWMServoPulse(1, 1800, 200)
    time.sleep(0.2) Board.setPWMServoPulse(1,
    1200, 200) time.sleep(0.2)
    Board.setPWMServoPulse(1, 1500, 100) #look straight ahead

    time.sleep(0.1)
```

In order to facilitate calling in the main program, we created a new "nod_head" function to represent this code. In addition, every time the servo rotates, a delay should be added, and the action transition will be smooth and stable. You can use the "sleep" function in the "time" module, and the parameter filled in is the delay time.

Similar to the method of controlling the robot to nod, we can create a new "shake_head" function to represent shaking. We Just need to copy the code above, start a new line, and change servo 1 to servo 2.

Finally, call these two functions in turn in the main program to realize the robot's nodding and shaking.

```
while True:

    print("I am nodding!")

    time.sleep(0.2)

    nod_head()

    time.sleep(0.5)


    print("I am shaking head!")

    time.sleep(0.2)

    shake_head()

    time.sleep(0.5)
```