**14.Raspberry Pi platform -------** **Servo\_control**

**1.Preparation**



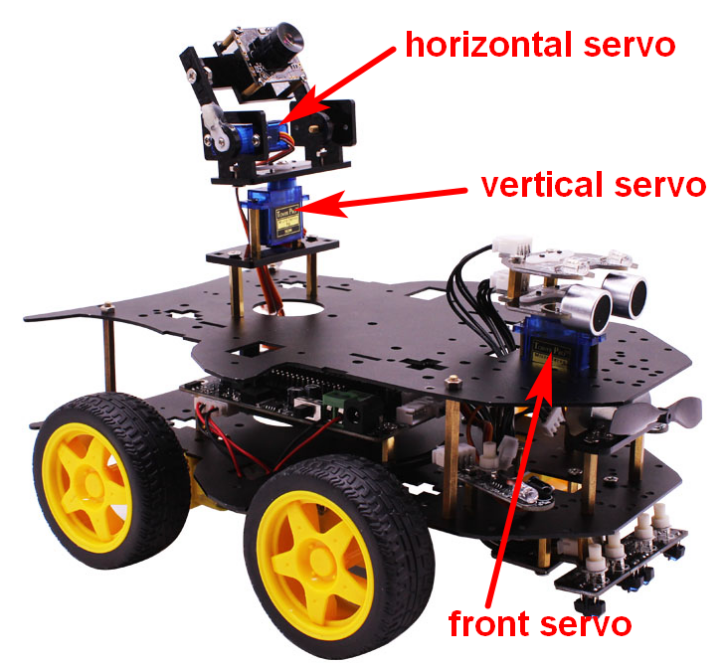
* 1. Raspberry Pi board



1-2 SG90 servo

**2.Purpose of Experimental**

After running the Servo\_control executable in the Raspberry Pi system. The servo will start to turn to different angles.



**3.Principle of experimental**

The working principle of the servo: the control signal enters the signal modulation chip from the channel of the receiver to obtain the bias voltage of the DC. It has a reference circuit inside, which generates a reference signal with a period of 20ms and a width of 1.5ms. It will compares the DC bias voltage with the voltage of the potentiometer to obtain a voltage difference and output. The positive and negative of the voltage difference is outputted to the motor drive chip to determine the forward and reverse of the motor.

Servo rotation angle is by adjusting the duty ratios of PWM (pulse width modulation) signal. The standard PWM (pulse width modulation) signal has a fixed period of 20ms (50Hz). Theoretically, pulse width distribution should be between 1 ms to 2 ms, but in fact between pulse width can be 0.5 ms and 2.5 ms. Pulse width and the servo rotation angle 0°～180° corresponds, as shown below.

0.5ms-----------------0°

1.0ms-----------------45°

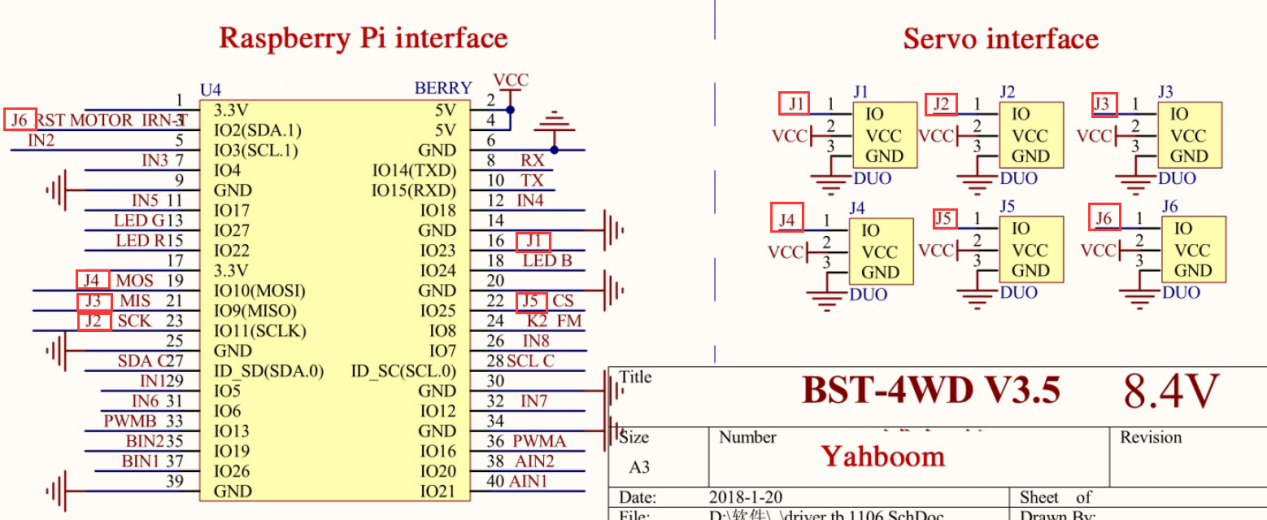
1.5ms-----------------90°

2.0ms-----------------135°

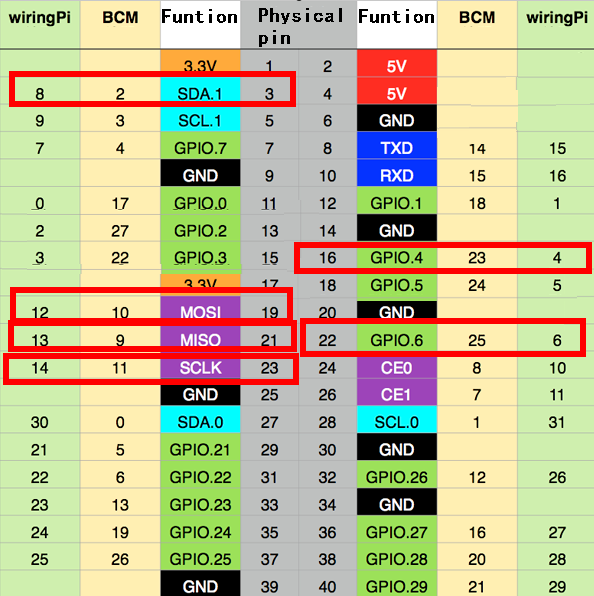
2.5ms-----------------180°

**4.Experimental principle**

4-1 About the schematic



4-1 Servo interface



4-2 Raspberry Pi 40 pins comparison table

The control interface of this 4WD car possess J1, J2, J3,J4,J5,J6 six interfaces (some of the six interfaces and other functions are multiplexed interfaces, which cannot be used at the same time).

**Note:The front servo is be connected to servo interface J1**

**The horizontal servo is be connected to servo interface J2**

**The vertical servo is be connected to servo interface J3**

4-2 According to the circuit schematic:

J1---16(Physical pin)-----4(wiringPi)

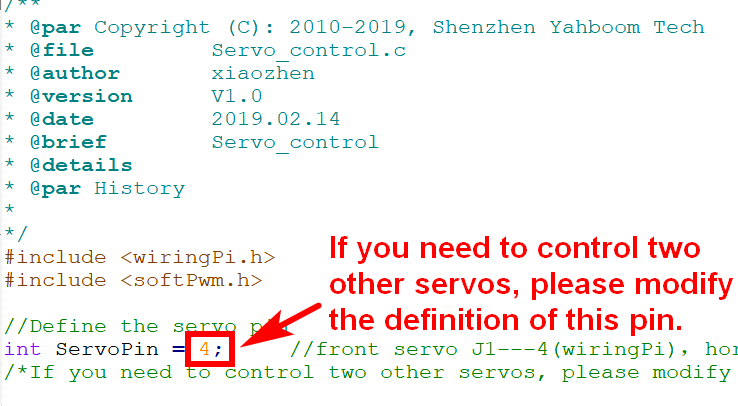
J2---23(Physical pin)-----13(wiringPi)

J3---21(Physical pin)-----14(wiringPi)

(Note: We use the wiringPi library to write code.)

4-3 About the code

See the Servo\_control.c file for the code.

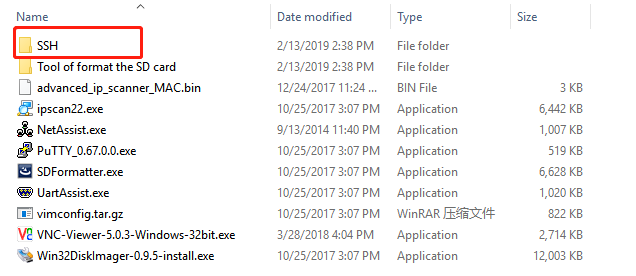


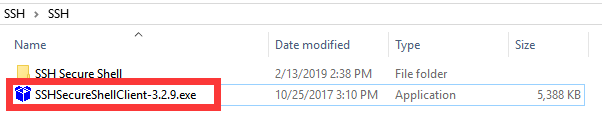
**5.Steps**

First, you need to remotely transfer the Servo\_control.c we provide to the Raspberry Pi image system via SSH.

The transfer steps are as follows:

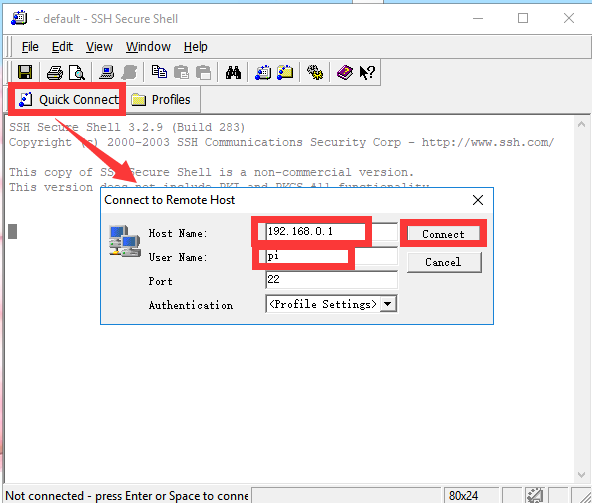
1).Click on the location shown below to get the SSH transfer tool.

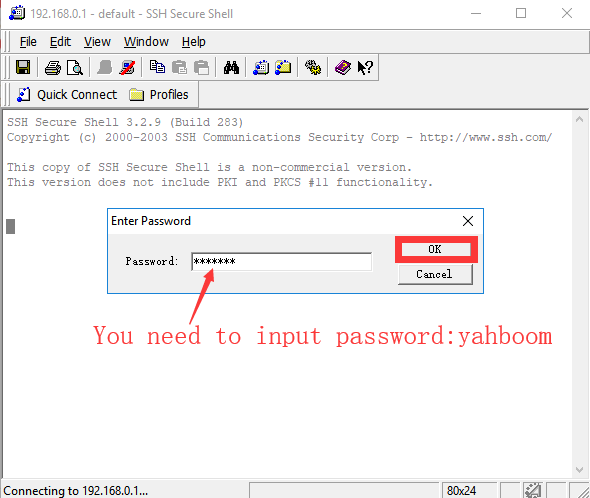


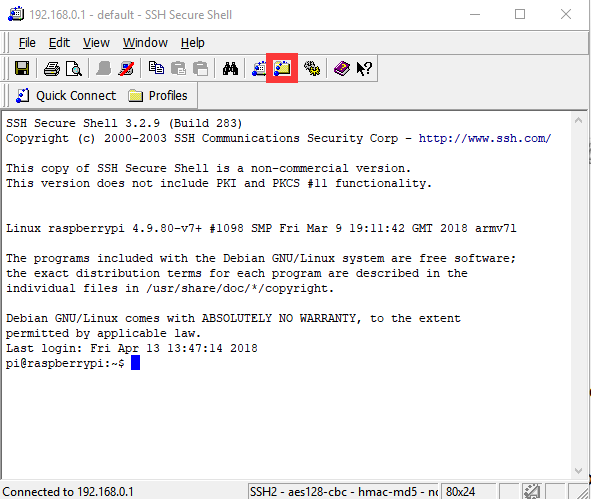


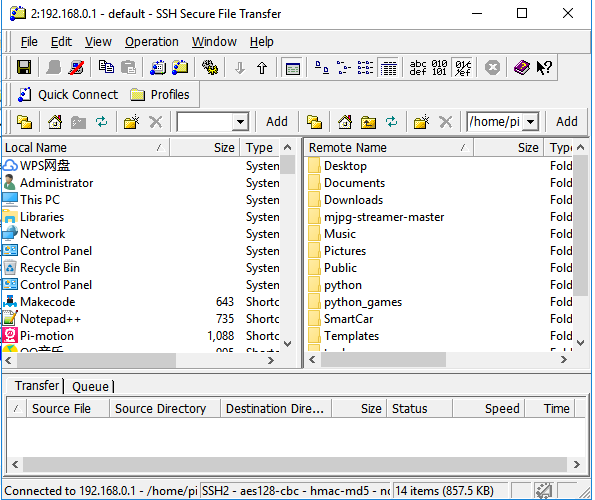
2).Click on the location shown below to log in to the Raspberry Pi system.

(The IP address is 192.168.0.1 and the password is yahboom.)

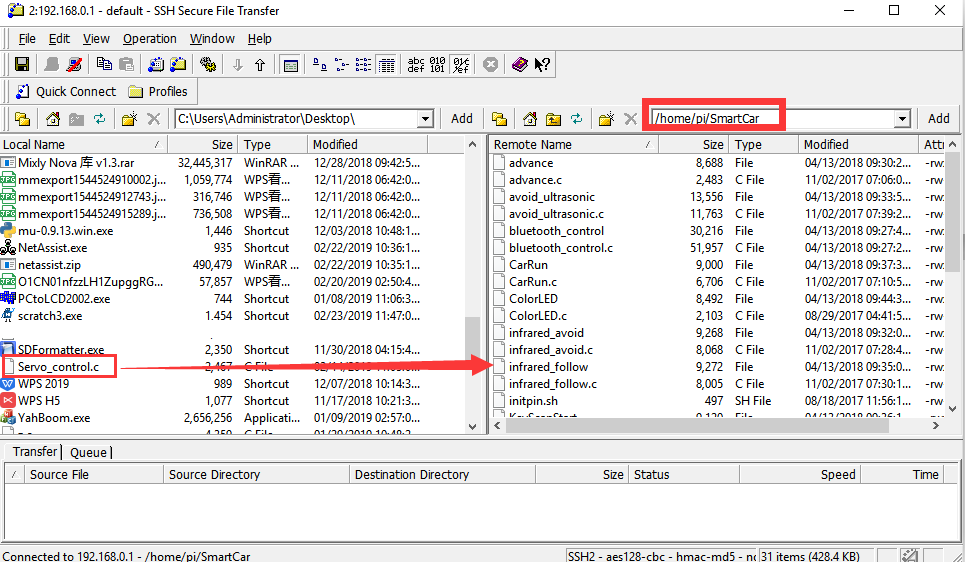




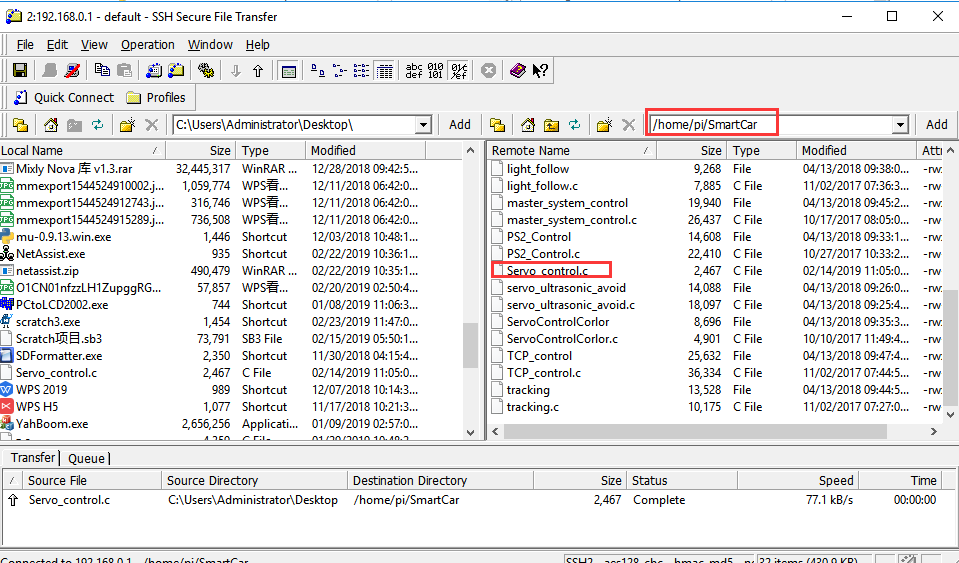
3）Click on the location shown below to transfer file. 



4). We need to drag the Servo\_control.c file to the right Raspberry Pi system.



5). After the drag and drop is completed, as shown in the figure below, we can see Servo\_control.c in the Raspberry Pi system on the right.



After the above steps, we have successfully transferred the Servo\_control.c file to the Raspberry Pi image.

Then, you need to remotely log in to the Raspberry Pi system by putty, as shown below:

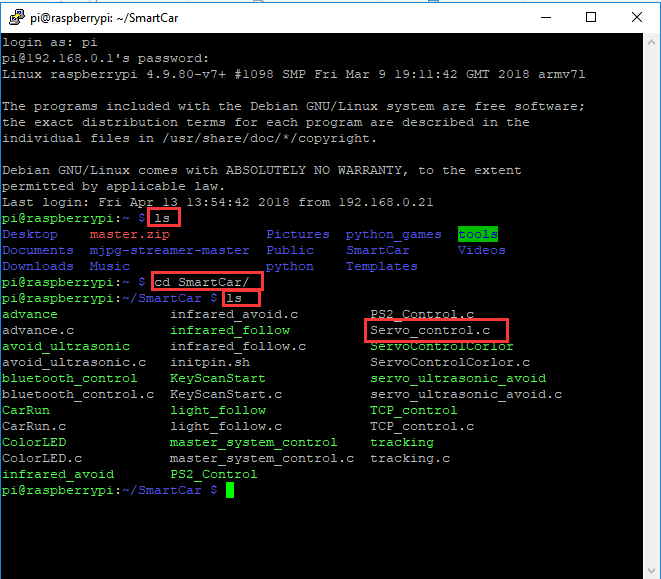
1).We need to go to the SmartCar directory:

Enter command: cd SmartCar

   ls

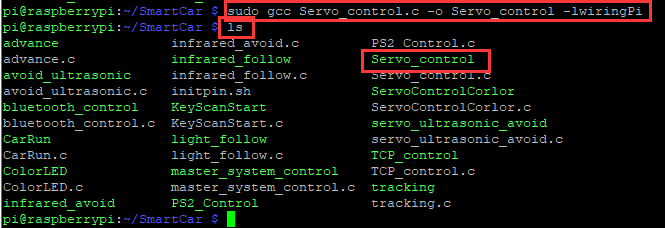
We can see the Servo\_control.c file inside.

As shown below:



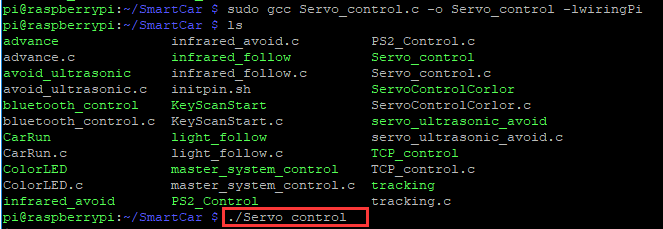
2).We need to compile this file in the Raspberry Pi system. (Note: we need to add -lwiringPi to the library file.)

We need to input: sudo gcc Servo\_control.c -o Servo\_control -lwiringPi



3).We need to run the compiled executable file in the Raspberry Pi system.We need to input: ./Servo\_control

As shown in the figure below.



4)We can input: ctrl+c to stop this process, which mean is send a signal to the linux kernel to terminate the current process, but the state of the relevant pin is uncertain at this time, we also need to run a script to initialize all pins.

(Note:The initpin.sh script file is included in the SmartCar directory.)

You need to input:  chmod 777 initpin.sh

./initpin.sh

After completing the above steps, the experiment is over.