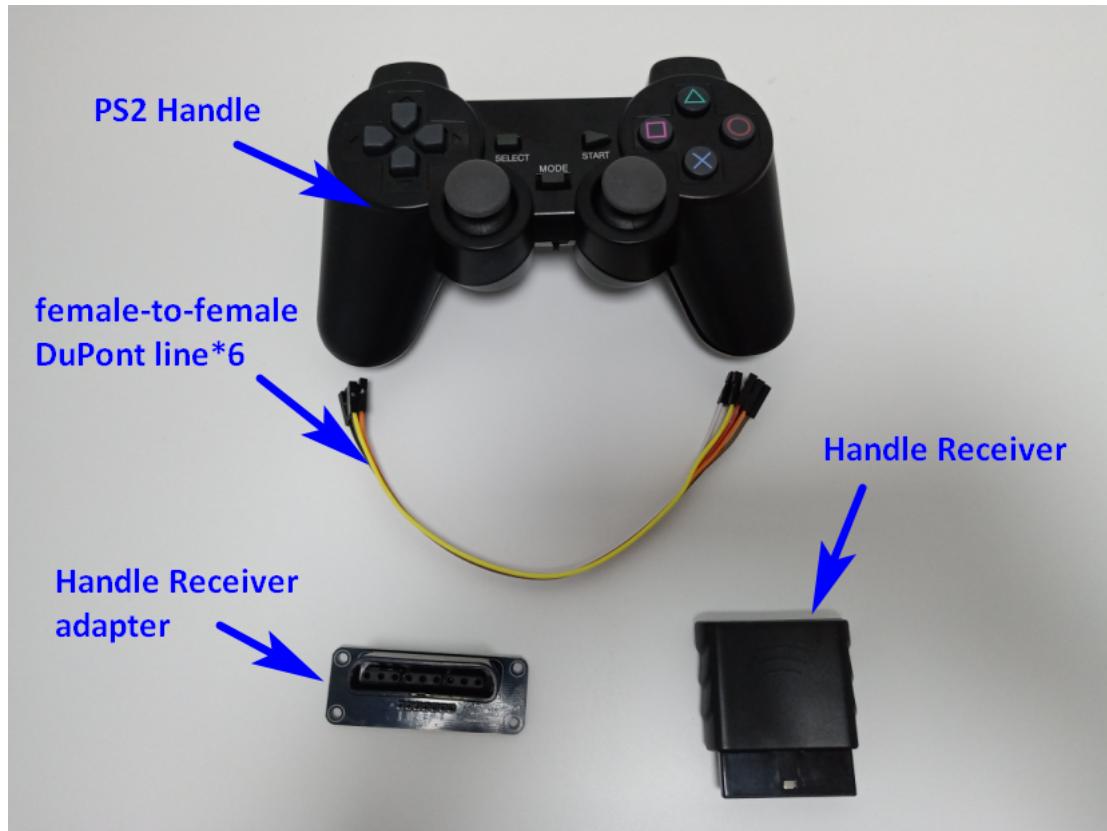


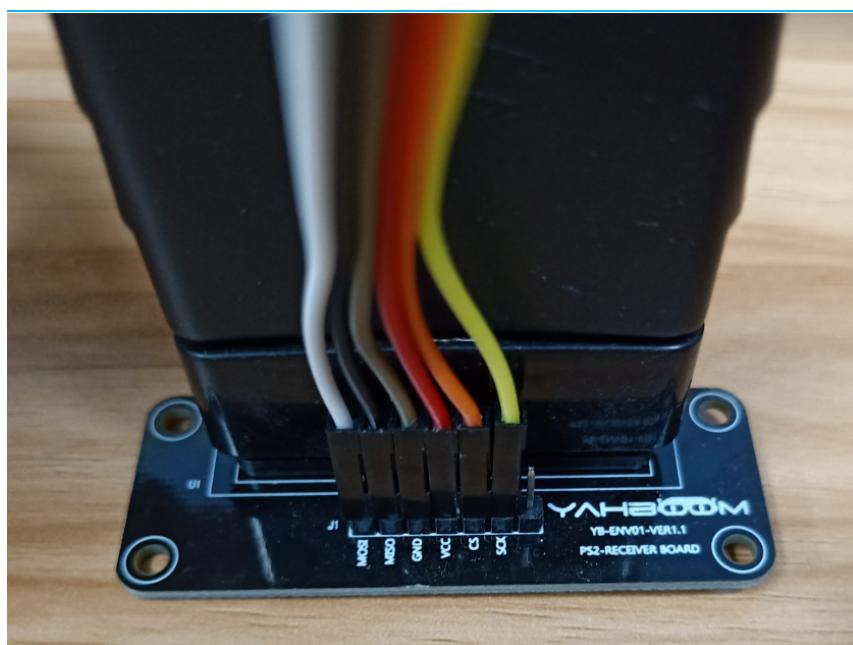
15. Raspberry Pi platform ----- PS2_control

Part1--- Wiring of HardWare

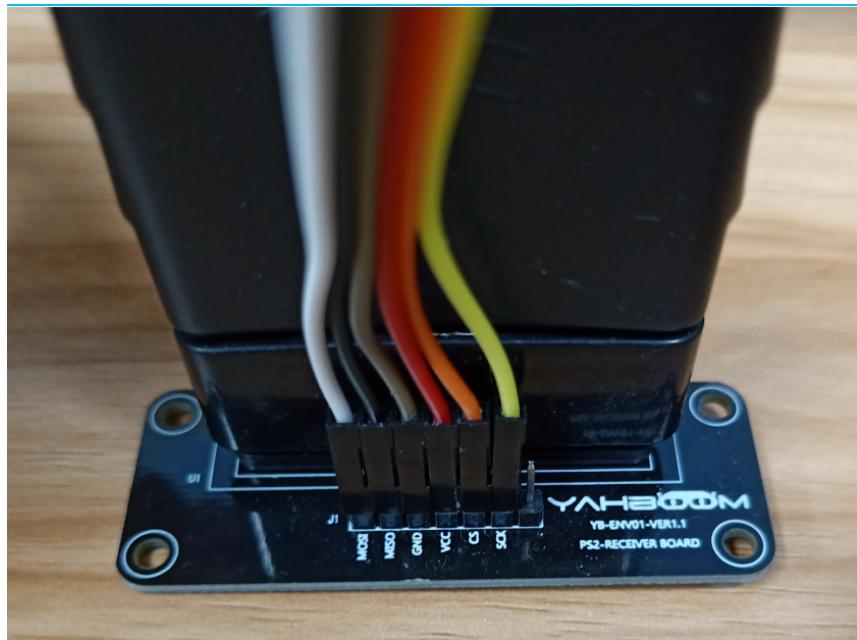
1.1) You need to prepare the Yahboom PS2 handle kit, 4WD car and the DuPont lines. As shown below.

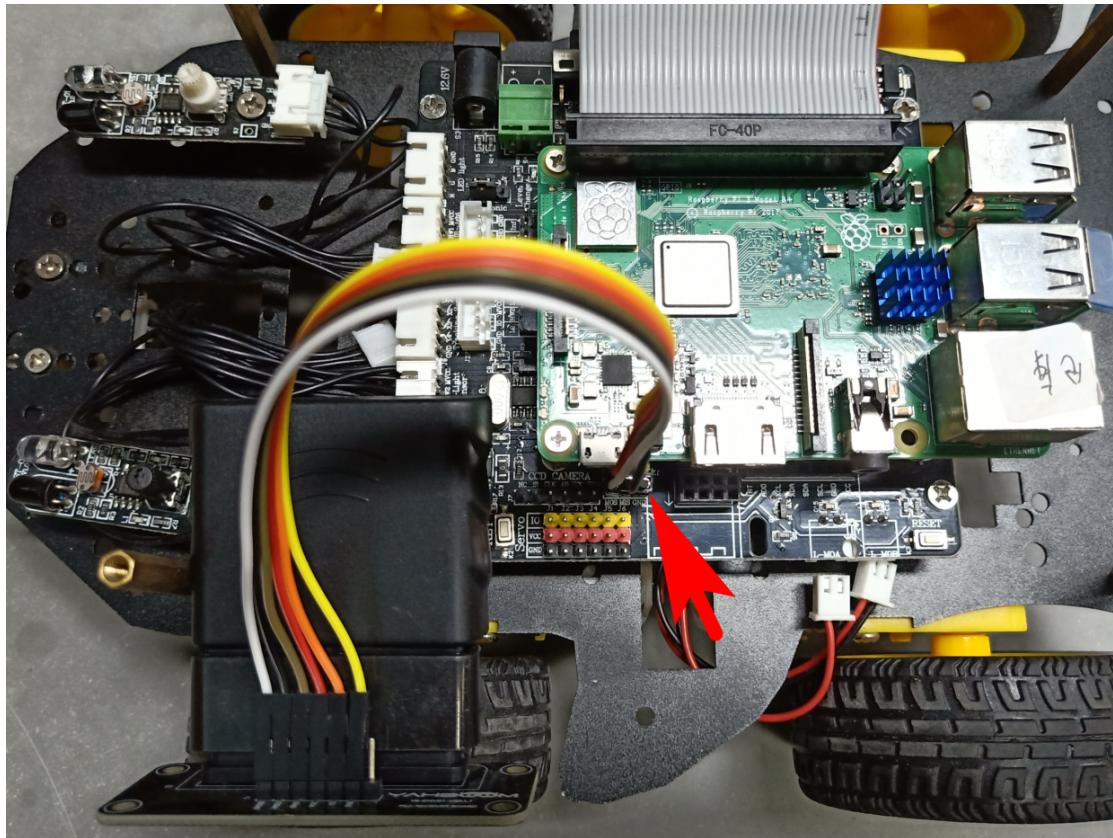


1.2) We need to connect Handle Receiver to expansion board.



YAHBOOM





1.3) Open the power switch of PS handle, you will see green light on PS handle is flashing.

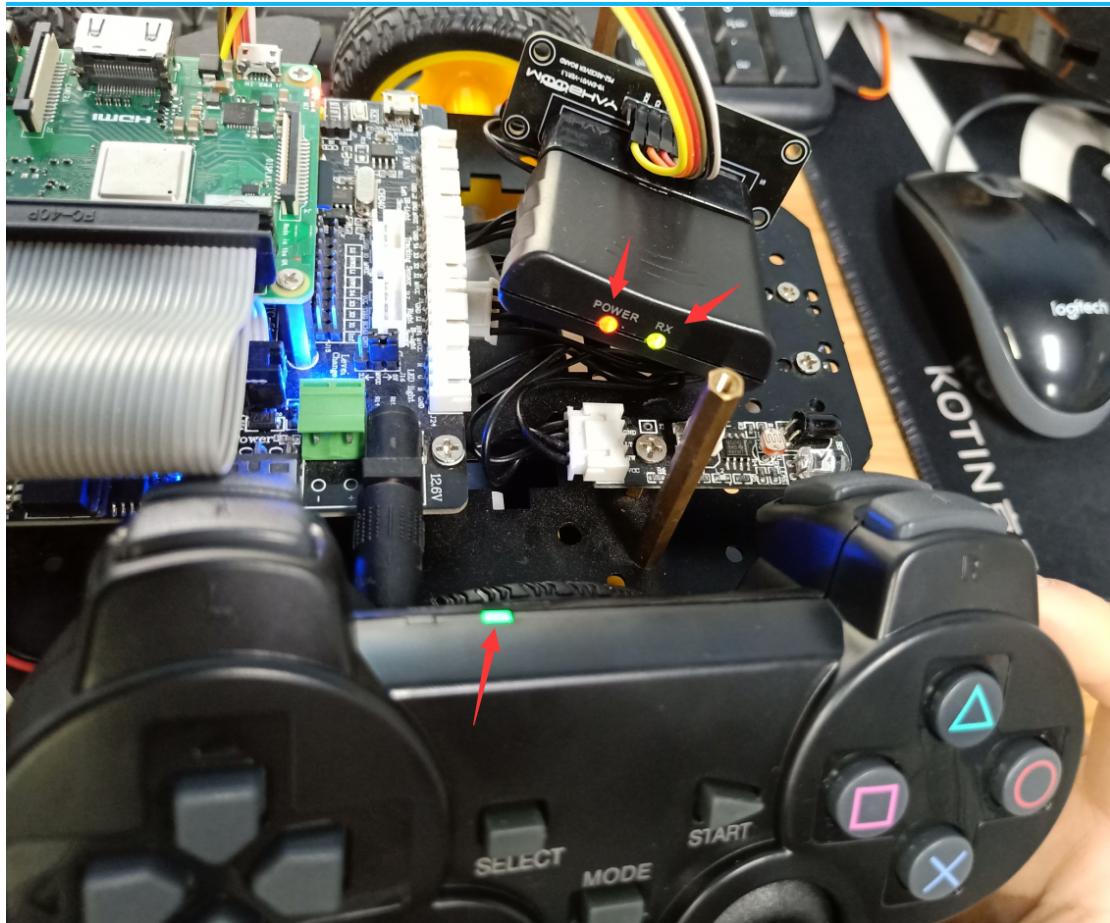
Open the power switch of robot car, you will see red light on Handle Receiver is keep on, green light on Handle Receiver is flashing.

Wait patiently for a while, they will automatically pair and connect.

1.4) After successfully connected, you can see green light on Handle Receiver stop flashing and keep on. Green light on PS handle stop flashing and keep on.

As shown below.





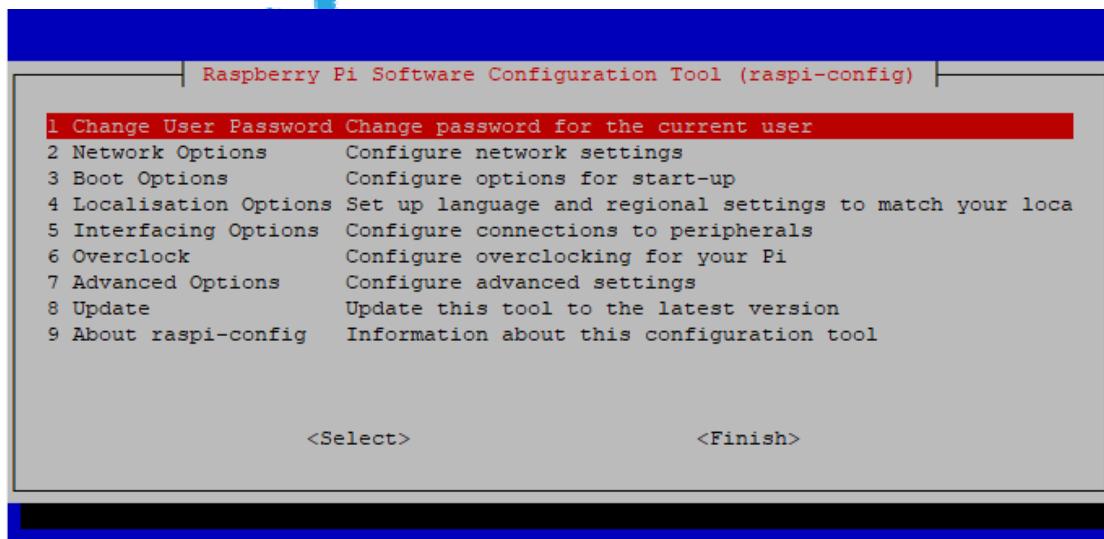
Part 2---Open SPI service

2.1) We need to open SPI service of Raspberry Pi system.

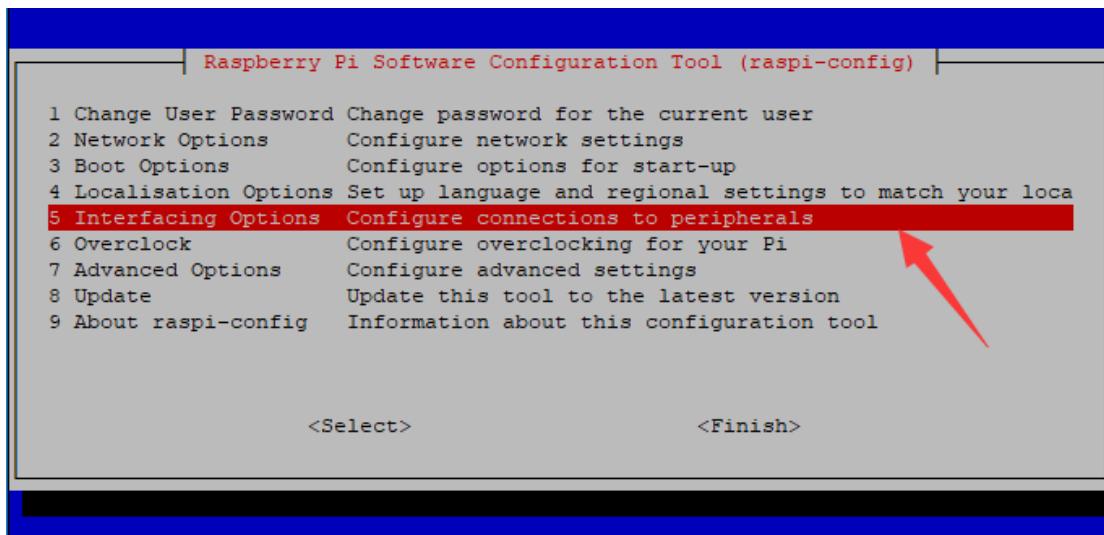
Input command: `sudo raspi-config`

```
pi@raspberrypi:~ $ sudo raspi-config
```

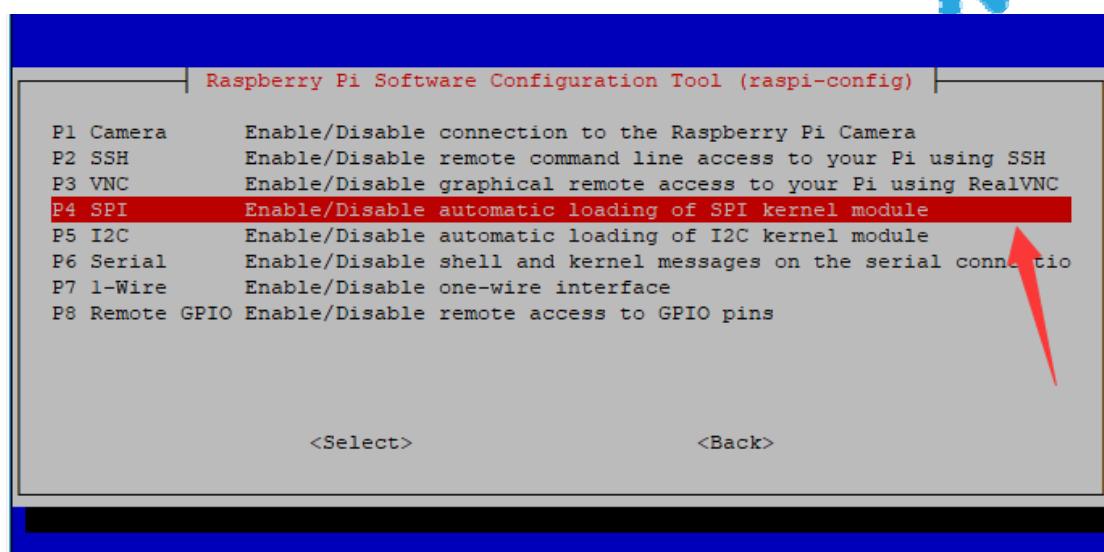
You will see interface as shown below:



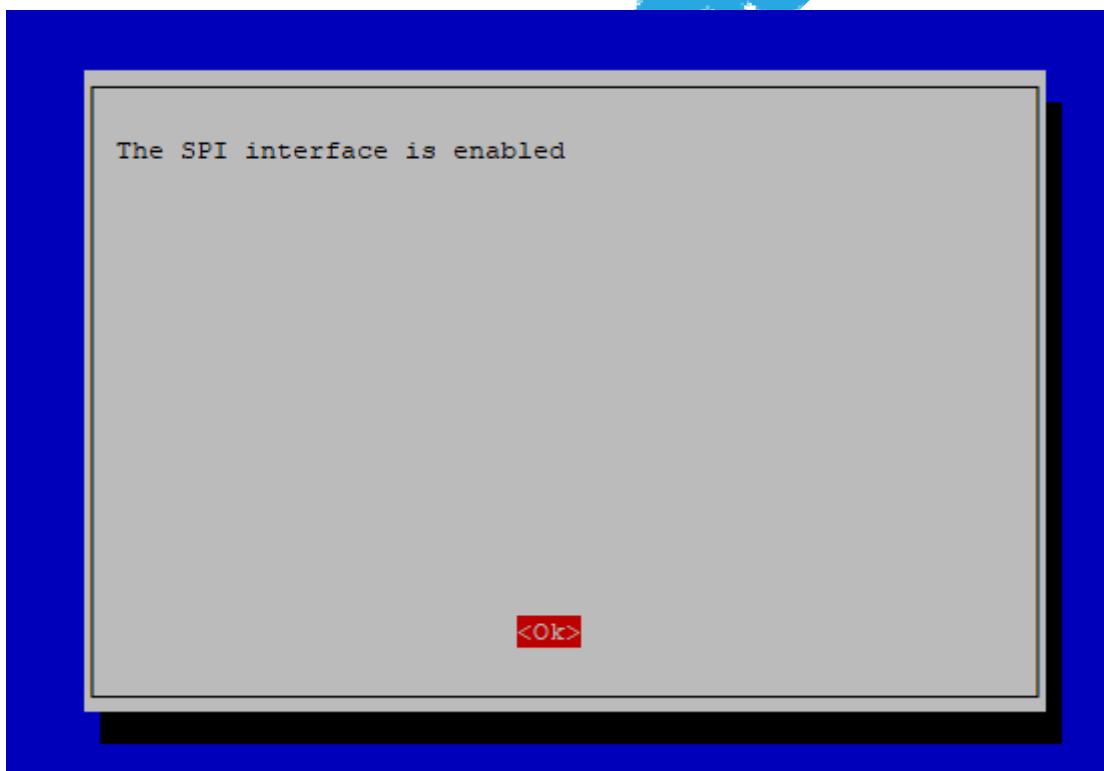
2.2) We need to choose 【Interfacing Options】.



2.3) We need to choose **【SPI】**.



2.4) We need to enable **【SPI】**.



2.5 Restart Raspberry Pi.

Part 3---Transfer Code

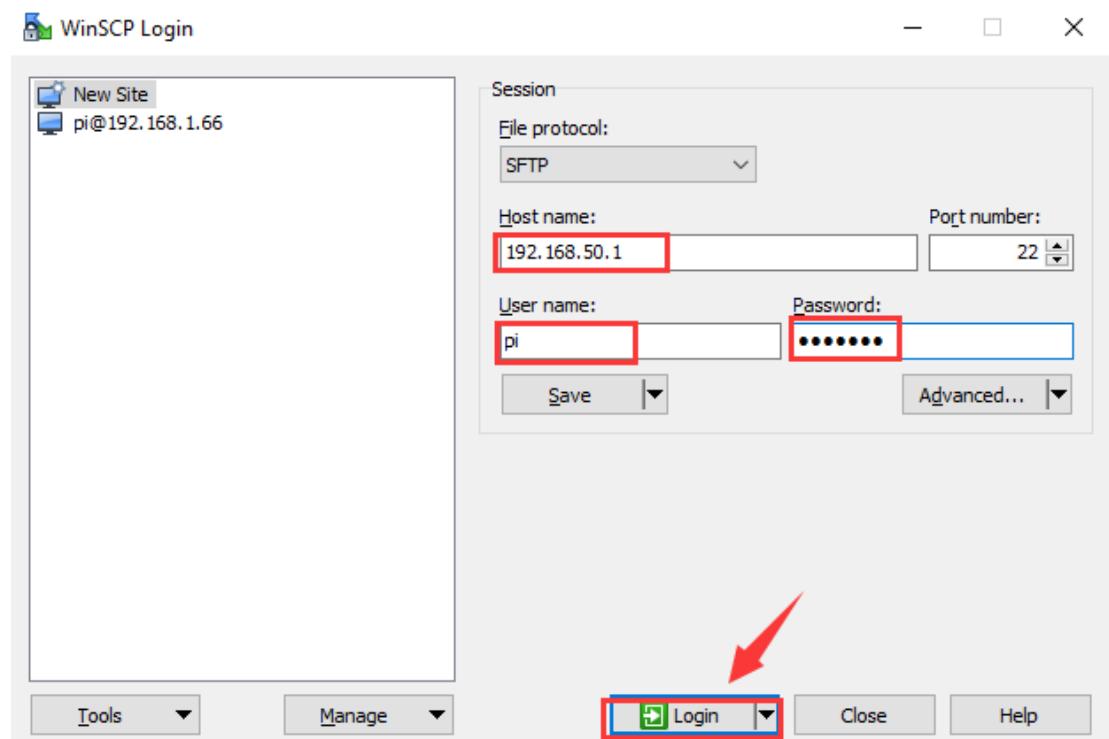
You need to remotely transfer the [PS2_Control.c](#) or [PS2_control.py](#) we provide to the Raspberry Pi image system via SSH.

!Note: After my test, I found that PS2_Control.c control is better. It is recommended to use PS2_Control.c instead of PS2_control.py.

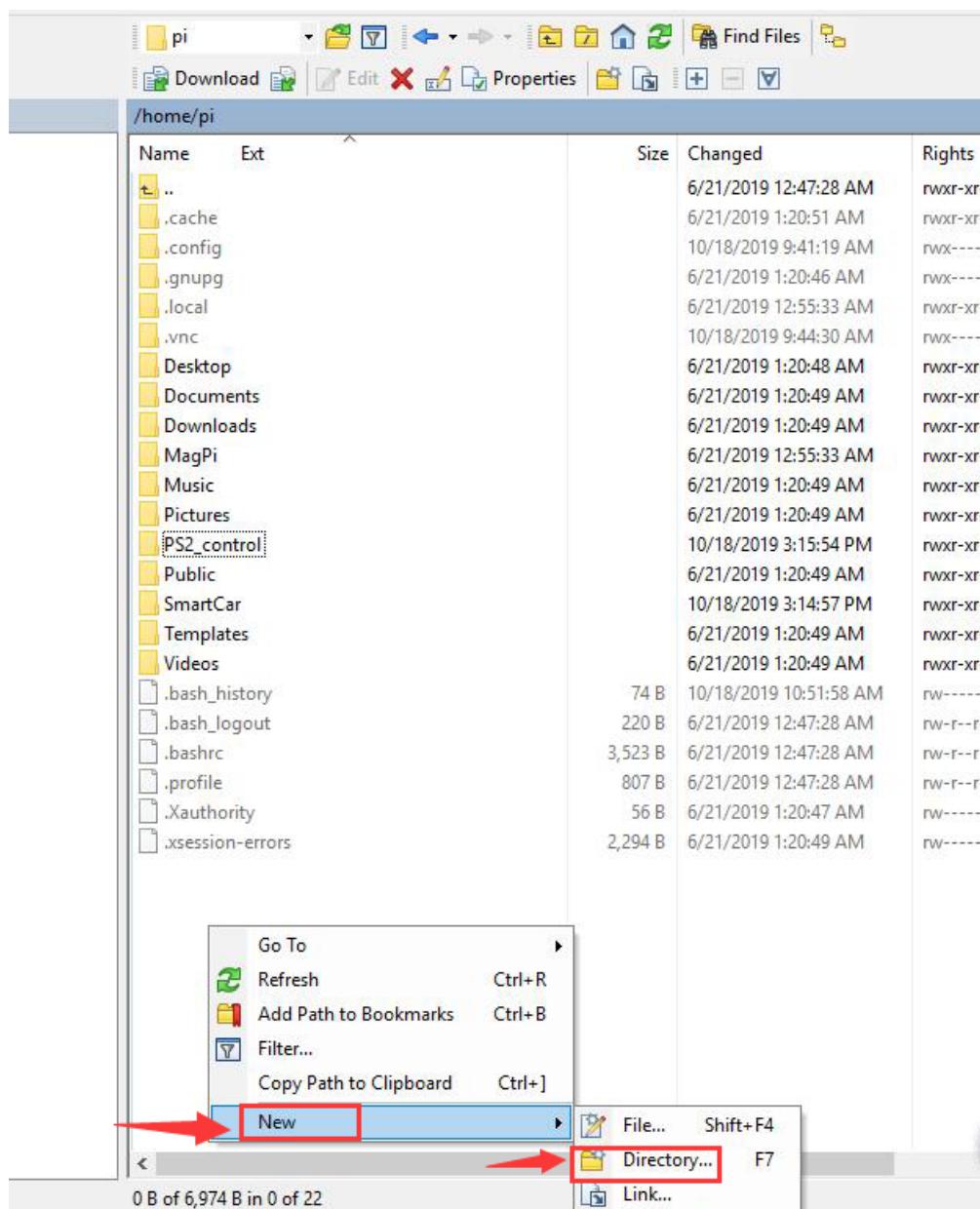
The transfer steps are as follows:

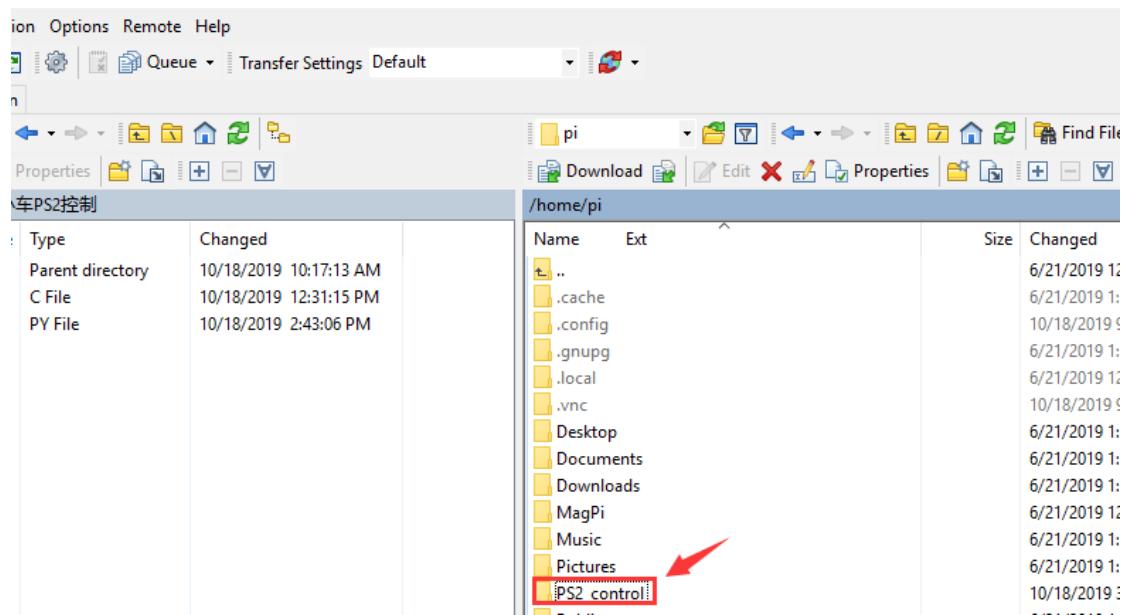
3.1)Get the WinSCP transfer tool and click on the location shown below to log in to the Raspberry Pi system.

(The IP address is 192.168.50.1 and the password is yahboom. Just for my image)

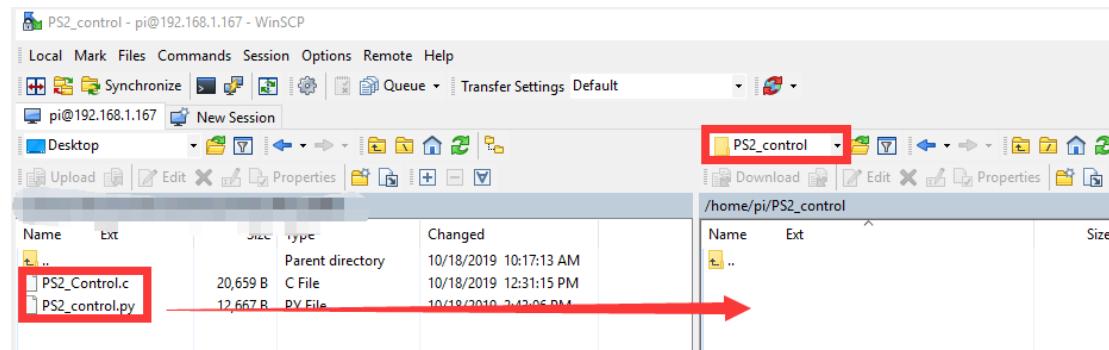


3.2)You can create a new folder to Storage PS2 control code. As shown below.

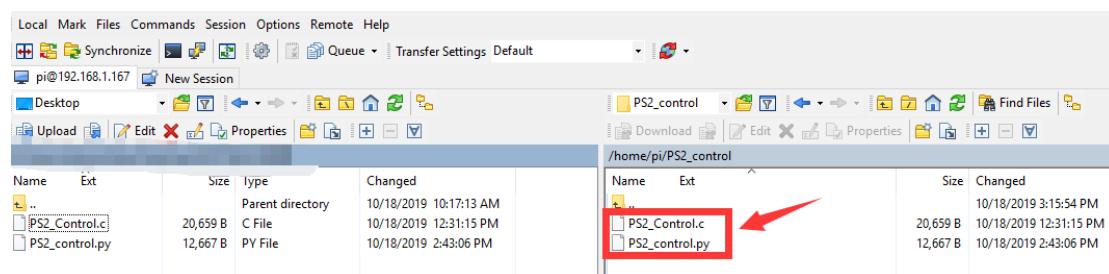




3.3) We need to drag the PS2_Control.c or PS2_control.py file to the right Raspberry Pi system.



3.4) After the drag and drop is completed, as shown in the figure below, we can see PS2_Control.c or PS2_control.py in the Raspberry Pi system on the right.



After the above steps, we have successfully transferred the PS2_Control.c or PS2_control.py file to the Raspberry Pi image.

Part 4---Running Code

4.1) You need to remotely log in to the Raspberry Pi system via putty, as shown below:

```
pi@raspberrypi: ~
login as: pi
pi@192.168.0.1's password:
Linux raspberrypi 4.9.80-v7+ #1098 SMP Fri Mar 9 19:11:42 GMT 2018 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Fri Apr 13 16:30:06 2018 from 192.168.0.11
```

4.2) Switch the root user.

You need to enter the command: **su**

Then enter the password **yahboom**

(the password here is hidden and can't be seen, please be careful not to enter the error!)

At this point we successfully entered the root privileges, as shown below:

```
pi@raspberrypi: ~
login as: pi
pi@192.168.0.1's password:
Linux raspberrypi 4.9.80-v7+ #1098 SMP Fri Mar 9 19:11:42 GMT 2018 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Fri Apr 13 16:30:06 2018 from 192.168.0.11
pi@raspberrypi:~ $ ls
Desktop           Music      RPi.GPIO-0.6.3  vision.py
Documents         Pictures   SmartCar       wiringPi-8d188fa
Downloads         Public     Templates    wiringPi-8d188fa.tar.gz
master.zip        Python    Tools
mjpg-streamer-master python_games Videos
pi@raspberrypi:~ $ su
Password:
root@raspberrypi:/home/pi#
```

4.3) You need to enter the command: **top**

This command is to view the process ID of all processes in the Raspberry Pi system.

As shown in the figure below, we can see that the **519** process number is a Bluetooth remote process, you need to remember this number.

(Note! Different Raspberry Pi process numbers are different. Please refer to the process shown in your own system)

```

pi@raspberrypi: ~
login as: pi
pi@192.168.0.1's password:
Linux raspberrypi 4.9.80-v7+ #1098 SMP Fri Mar 9 19:11:42 GMT 2018 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Fri Apr 13 16:30:06 2018 from 192.168.0.11
pi@raspberrypi:~ $ ls
Desktop           Music        RPi.GPIO-0.6.3  vision.py
Documents         Pictures     SmartCar       wiringPi-8dl88fa
Downloads         Public       Templates    wiringPi-8dl88fa.tar.gz
master.zip        Python      tools
mjpg-streamer-master python_games Videos
pi@raspberrypi:~ $ su
Password:
root@raspberrypi:/home/pi# top
top - 16:33:43 up 16 min, 4 users, load average: 0.05, 0.15, 0.11
Tasks: 158 total, 1 running, 157 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.4 us, 1.7 sy, 0.0 ni, 96.7 id, 0.0 wa, 0.0 hi, 1.1 si, 0.0 st
KiB Mem : 949580 total, 679424 free, 98728 used, 171428 buff/cache
KiB Swap: 102396 total, 102396 free, 0 used. 786496 avail Mem

PID USER PR NI VIRT   RSS   SHR S %CPU %MEM TIME+ COMMAND
519 root  20  0 43944  604   496 S 23.3  0.1  3:54.18 bluetooth_contr
1120 root  20  0 8264  3224  2736 R  1.3  0.3  0:01.06 top
288 systemd+ 20  0 17276 3632  3236 S  0.7  0.4  0:00.69 systemd-timesyn
  1 root  20  0  9652  6024  4892 S  0.3  0.6  0:02.66 systemd
243 root  20  0 0  0 0  0 S  0.3  0.0  0:00.16 brcmf_wdog/mmcl
262 root  20  0 0  0 0  0 S  0.3  0.0  0:00.50 kworker/u8:2
486 dnsmasq 20  0 9080  1688  1336 S  0.3  0.2  0:00.28 dnsmasq
817 pi  20  0 140008 23636 19940 S  0.3  2.5  0:04.79 lxpanel
1093 pi  20  0 11652  3452  2720 S  0.3  0.4  0:00.04 sshd
  2 root  20  0 0  0 0  0 S  0.0  0.0  0:00.00 kthreadd
  3 root  20  0 0  0 0  0 S  0.0  0.0  0:00.07 ksoftirqd/0
  5 root  0 -20 0  0 0  0 S  0.0  0.0  0:00.00 kworker/0:0H
  7 root  20  0 0  0 0  0 S  0.0  0.0  0:00.59 rcu_sched
  8 root  20  0 0  0 0  0 S  0.0  0.0  0:00.00 rcu_bh
  9 root  rt  0 0  0 0  0 S  0.0  0.0  0:00.01 migration/0
 10 root  0 -20 0  0 0  0 S  0.0  0.0  0:00.00 lru-add-drain
 11 root  20  0 0  0 0  0 S  0.0  0.0  0:00.00 cpuhp/0
 12 root  20  0 0  0 0  0 S  0.0  0.0  0:00.00 cpuhp/1
 13 root  rt  0 0  0 0  0 S  0.0  0.0  0:00.00 migration/1
 14 root  20  0 0  0 0  0 S  0.0  0.0  0:00.02 ksoftirqd/1
 16 root  0 -20 0  0 0  0 S  0.0  0.0  0:00.00 kworker/1:0H
 17 root  20  0 0  0 0  0 S  0.0  0.0  0:00.00 cpuhp/2
 18 root  rt  0 0  0 0  0 S  0.0  0.0  0:00.00 migration/2
 19 root  20  0 0  0 0  0 S  0.0  0.0  0:00.03 ksoftirqd/2

```

4.4) After obtaining the Bluetooth process number, press **ctrl+z** on the keyboard to exit this interface.

4.5) We must kill the Bluetooth process in order to avoid conflicts between the PS2_control process and the Bluetooth process.

Enter the command: **kill -9 519**

As shown below:

```

pi@raspberrypi:~ $ su
Password:
root@raspberrypi:/home/pi# top
top - 16:33:43 up 16 min, 4 users, load average: 0.05, 0.15, 0.11
Tasks: 158 total, 1 running, 157 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.4 us, 1.7 sy, 0.0 ni, 96.7 id, 0.0 wa, 0.0 hi, 1.1 si, 0.0 st
KiB Mem : 949580 total, 679424 free, 98728 used, 171428 buff/cache
KiB Swap: 102396 total, 102396 free, 0 used. 786496 avail Mem

      PID USER      PR  NI    VIRT    RES    SHR S %CPU %MEM     TIME+ COMMAND
  519 root      20   0  43944   604   496 S 23.3  0.1  3:54.18 bluetooth_contr
1120 root      20   0   8264  3224  2736 R  1.3  0.3  0:01.06 top
 288 systemd+  20   0  17276  3632  3236 S  0.7  0.4  0:00.69 systemd-timesyn
  1 root      20   0   9652  6024  4892 S  0.3  0.6  0:02.66 systemd
 243 root      20   0       0     0     0 S  0.3  0.0  0:00.16 brcmf_wdog/mmcl
 262 root      20   0       0     0     0 S  0.3  0.0  0:00.50 kworker/u8:2
 486 dnsmasq   20   0   9080  1688  1336 S  0.3  0.2  0:00.28 dnsmasq
  817 pi       20   0 140008 23636 19940 S  0.3  2.5  0:04.79 lxpanel
1093 pi       20   0  11652  3452  2720 S  0.3  0.4  0:00.04 sshd
  2 root      20   0       0     0     0 S  0.0  0.0  0:00.00 kthreadd
  3 root      20   0       0     0     0 S  0.0  0.0  0:00.07 ksoftirqd/0
  5 root      0 -20       0     0     0 S  0.0  0.0  0:00.00 kworker/0:0H
  7 root      20   0       0     0     0 S  0.0  0.0  0:00.59 rcu_sched
  8 root      20   0       0     0     0 S  0.0  0.0  0:00.00 rcu_bh
  9 root      rt  0       0     0     0 S  0.0  0.0  0:00.01 migration/0
 10 root      0 -20       0     0     0 S  0.0  0.0  0:00.00 lru-add-drain
 11 root      20   0       0     0     0 S  0.0  0.0  0:00.00 cpuhp/0
 12 root      20   0       0     0     0 S  0.0  0.0  0:00.00 cpuhp/1
 13 root      rt  0       0     0     0 S  0.0  0.0  0:00.00 migration/1
 14 root      20   0       0     0     0 S  0.0  0.0  0:00.02 ksoftirqd/1
 16 root      0 -20       0     0     0 S  0.0  0.0  0:00.00 kworker/1:0H
 17 root      20   0       0     0     0 S  0.0  0.0  0:00.00 cpuhp/2
 18 root      rt  0       0     0     0 S  0.0  0.0  0:00.00 migration/2
 19 root      20   0       0     0     0 S  0.0  0.0  0:00.03 ksoftirqd/2
[1]+  Stopped                  top
root@raspberrypi:/home/pi# kill -9 519
root@raspberrypi:/home/pi# █

```

4.6) We need to go to the PS2_control directory:

Enter command: `cd PS2_control`

`ls`

We can see the `PS2_Control.c` and `PS2_control.py` file inside.

As shown below:

```

login as: pi
pi@192.168.1.167's password:
Linux raspberrypi 4.19.50-v7+ #896 SMP Thu Jun 20 16:11:44 BST 2019 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Fri Oct 18 08:17:54 2019 from 192.168.1.16

SSH is enabled and the default password for the 'pi' user has not been changed.
This is a security risk - please login as the 'pi' user and type 'passwd' to set
a new password.

pi@raspberrypi:~ $ ls
Desktop  Downloads  Music      PS2_control  SmartCar  Videos
Documents MagPi      Pictures  Public      Templates
pi@raspberrypi:~ $ cd PS2_control/
pi@raspberrypi:~/PS2_control $ ls
PS2_Control.c  PS2_control.py
pi@raspberrypi:~/PS2_control $ █

```

For PS2_Control.c

Enter the command:

```
sudo gcc PS2_Control.c -o PS2_Control -lwiringPi -lpthread
```

This command is to compile PS2_Control.c generates the executable file PS2_Control., which is correct as long as no error is reported during compilation, as shown in the following figure.

```
pi@raspberrypi:~/PS2 control $ sudo gcc PS2 Control.c -o PS2 Control -lwiringPi -lpthread
```

Enter the command:

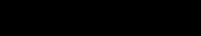
./PS2 Control

This command is to run the executable.

```
pi@raspberrypi:~/PS2 control $ ./PS2 Control
```

Then, you can control the robot car by PS2 handle. At the same time, we can see that some data will be printed.

```
pi@raspberrypi:~/PS2_control $ sudo gcc PS2_Control.c -o PS2_Control -lwiringPi -lpthread  
pi@raspberrypi:~/PS2_control $ ./PS2_Control  
PSB_TRIANGLE  
PSB_TRIANGLE  
PSB_TRIANGLE  
PSB_PAD_UP  
PSB_PAD_UP  
PSB_PAD_UP  
PSB_PAD_UP  
PSB_PAD_UP  
PSB_PAD_UP  
PSB_PAD_DOWN  
PSB_PAD_DOWN  
PSB_PAD_DOWN  
PSB_PAD_LEFT  
PSB_PAD_LEFT  
PSB_PAD_LEFT  
PSB_PAD_RIGHT  
PSB_PAD_RIGHT
```



For PS2 control.py

Enter the command:

sudo python PS2_control.py

```
pi@raspberrypi: ~/PS2_control
Linux raspberrypi 4.19.50-v7+ #896 SMP Thu Jun 20 16:11:44 BST 2019 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Fri Oct 18 09:24:04 2019

SSH is enabled and the default password for the 'pi' user has not been changed.
This is a security risk - please login as the 'pi' user and type 'passwd' to set
a new password.

pi@raspberrypi:~ $ cd PS2_control/
pi@raspberrypi:~/PS2_control $ ls
PS2_Control PS2_control_1.py PS2_control_2.py PS2_Control.c PS2_control.py
pi@raspberrypi:~/PS2_control $ sudo python PS2_control.py
PS2_control.py:311: SyntaxWarning: name 'CarSpeedControl' is assigned to before
global declaration
  global CarSpeedControl
PS2_control.py:312: SyntaxWarning: name 'g_ServoState' is assigned to before glo
bal declaration
  global g_ServoState
PSB_PAD_UP
PSB_PAD_UP
PSB_PAD_RIGHT
PSB_PAD_RIGHT
PSB_PAD_UP
PSB_PAD_DOWN

These warnings do not affect the program operation

```

PS2 Handle data

Then, you can control the robot car by PS2 handle. At the same time, we can see that some data will be printed.

Part 5---Handle button function definition

