

Steps of TCP_Control:

Note :Please use a laptop to carry out the experiment.

1.First, you need to remotely transfer the [TCP_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 transfer tool -- Winscp.

Raspberry pi 4WD

- 1.Remote control operation
- 2.Development environment
- 3.Experimental tutorial
- 4.Battery and charging
- 5. Commonly used Video
- 6.Contact us

Download

- APP
- Code
- PC
- SCH
- Download image
- Tools

WinSCP file transfer between Raspberry Pi and windows

We need to transfer files between Windows and Raspberry Pi two different systems. Since this is two different file systems, we need to use the SSH service to transfer files across systems.

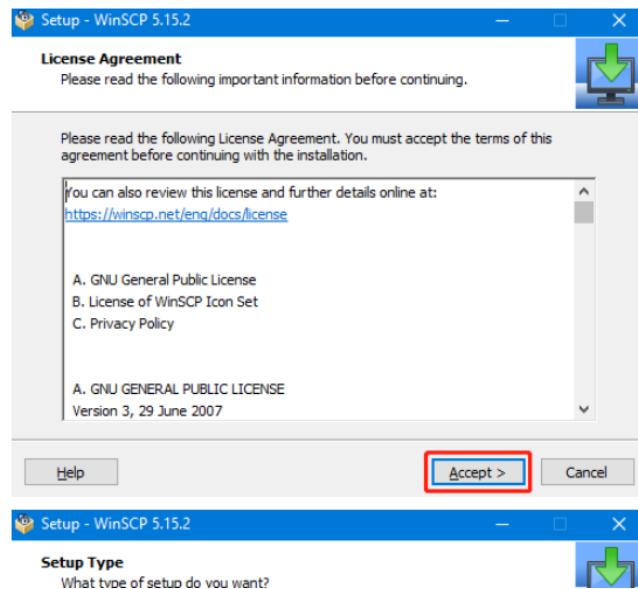
1.Install WinSCP

WinSCP official website: <https://winscp.net/eng/download.php>

(This software in the Tool folder.)

WinSCP-5.15.2-Setup.exe 7/3/2019 2:41 PM Application 9,609 KB

Double-click to open the program and start the installation, click "Accept" to accept the agreement, and then follow the prompts to install.



- 2.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**.)

3. We need to drag the [TCP_Control.c](#) file to the right Raspberry Pi system.

4. After the drag and drop is completed, as shown in the figure below, we can see [TCP_Control.c](#) in the Raspberry Pi system on the right.

File browser window showing the directory `/home/pi/SmartCar`. The `TCP_Control.c` file is highlighted with a red box.

Remote Name	/	Size	Type	Modified	Attributes
advance		8,688	文件	2018/04/13 21:4...	-rwxr-xr-x
advance.c		2,492	C 文件	2018/07/16 18:1...	-rw-r--r--
avoid_ultrasonic		13,556	文件	2018/04/13 21:3...	-rwxr-xr-x
avoid_ultrasonic.c		10,355	C 文件	2018/04/13 21:3...	-rw-r--r--
bluetooth_control		30,216	文件	2018/04/13 23:0...	-rwxr-xr-x
bluetooth_control.c		46,448	C 文件	2018/04/13 23:0...	-rw-r--r--
CarRun		9,000	文件	2018/04/13 22:4...	-rwxr-xr-x
CarRun.c		5,797	C 文件	2018/07/16 18:1...	-rw-r--r--
ColorLED		8,492	文件	2018/04/13 21:4...	-rwxr-xr-x
ColorLED.c		2,061	C 文件	2018/07/16 18:2...	-rw-r--r--
infrared_avoid		9,268	文件	2018/04/13 21:3...	-rwxr-xr-x
infrared_avoid.c		7,596	C 文件	2018/07/17 11:5...	-rw-r--r--
infrared_follow		9,272	文件	2018/04/13 22:1...	-rwxr-xr-x
infrared_follow.c		7,352	C 文件	2018/07/18 18:1...	-rw-r--r--
initpin.sh		497	SH 文件	2017/08/18 11:5...	-rwxrwx...
KeyScanStart		9,120	文件	2018/04/13 21:4...	-rwxr-xr-x
KeyScanStart.c		6,737	C 文件	2018/07/17 17:3...	-rw-r--r--
light_follow		9,268	文件	2018/04/13 22:1...	-rwxr-xr-x
light_follow.c		7,131	C 文件	2018/04/13 22:1...	-rw-r--r--
servo_ultrasonic_avoid		14,088	文件	2018/04/13 22:3...	-rwxr-xr-x
servo_ultrasonic_avoid.c		16,213	C 文件	2018/04/13 22:3...	-rw-r--r--
ServoControlColor		8,696	文件	2018/04/13 22:3...	-rwxr-xr-x
ServoControlColor.c		4,179	C 文件	2018/07/17 10:4...	-rw-r--r--
TCP_Control.c		27,847	C 文件	2019/01/09 18:0...	-rw-r--r--
tracking		13,528	文件	2018/04/13 21:4...	-rwxr-xr-x
tracking.c		9,874	C 文件	2018/07/17 11:5...	-rw-r--r--

After the above steps, we have successfully transferred the `TCP_Control.c` file to the Raspberry Pi image.

5. 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
```

6. 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

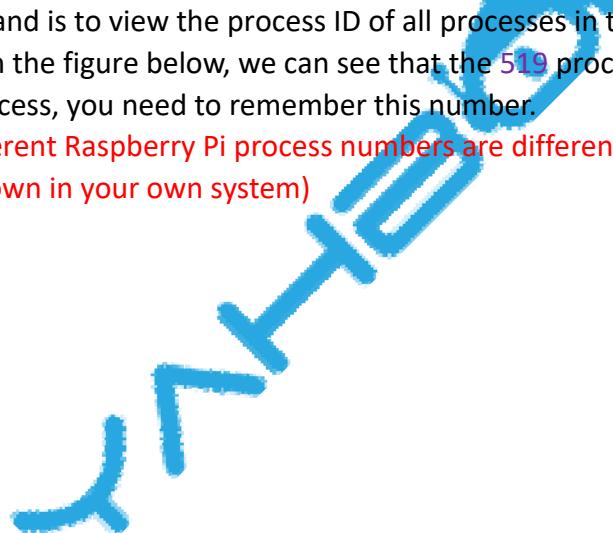
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

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-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#
```

7. 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 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 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 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 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 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

```

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

9.We must kill the Bluetooth process in order to avoid conflicts between the TCP 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# 

```

10. We need to go to the SmartCar directory:

Enter command: `cd SmartCar`

`ls`

We can see the `TCP_Control.c` file inside.

As shown below:

```

pi@raspberrypi:~ $ cd SmartCar/
pi@raspberrypi:~/SmartCar $ ls
advance          ColorLED.c      light_follow.c
advance.c        infrared_avoid ServoControlColor
avoid_ultrasonic infrared_avoid.c ServoControlColor.c
avoid_ultrasonic.c infrared_follow servo_ultrasonic_avoid
bluetooth_control infrared_follow.c servo_ultrasonic_avoid.c
bluetooth_control.c initpin.sh   TCP_Control.c
CarRun           KeyScanStart   tracking
CarRun.c         KeyScanStart.c tracking.c
ColorLED         light_follow

```

11. Enter the command:

`gcc TCP_Control.c -o TCP_Control -lwiringPi -lpthread`

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

```

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 16:35:42 2018
pi@raspberrypi:~ $ sudo restart
sudo: restart: command not found
pi@raspberrypi:~ $ cd SmartCar/
pi@raspberrypi:~/SmartCar $ ls
advance          ColorLED.c      light_follow.c
advance.c        infrared_void   ServoControlColor
avoid_ultrasonic infrared_void.c ServoControlColor.c
avoid_ultrasonic.c infrared_follow servo_ultrasonic_avoid
bluetooth_control infrared_follow.c servo_ultrasonic_avoid.c
bluetooth_control.c initpin.sh   TCP_Control.c
CarRun           KeyScanStart    tracking
CarRun.c         KeyScanStart.c  tracking.c
ColorLED         light_follow
pi@raspberrypi:~/SmartCar $ gcc TCP_Control.c -o TCP_Control -lwiringPi -lpthread
TCP_Control.c: In function `do_client recv':
TCP_Control.c:1060:2: warning: implicit declaration of function 'close' [-Wimplicit-function-declaration]
  close(sockfd);
^~~~~~
TCP_Control.c: In function `servo_control':
TCP_Control.c:1099:7: warning: implicit declaration of function 'sleep' [-Wimplicit-function-declaration]
  sleep(1);
^~~~~~

pi@raspberrypi:~/SmartCar $ ./TCP_Control

```

12.Enter the command:

./TCP_Control

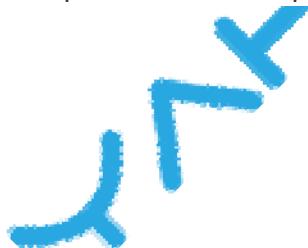
This command is to run the executable.

The prompt Listen.... will appear as shown below.

```

pi@raspberrypi:~/SmartCar $ ./TCP_Control
Listen....
```

13.We need to open the host computer software to choose BST_4WD_Raspberry Pi.



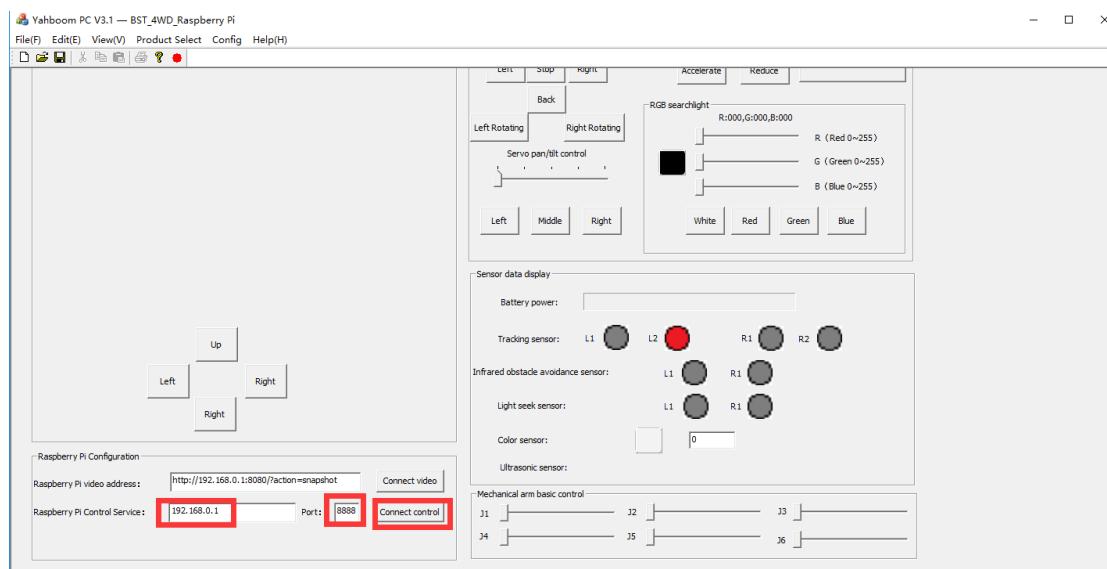
Please select the product type first:

X

Development Board
<input checked="" type="radio"/> V51 Development Board <input type="radio"/> Mini51 Development Board
Development Kit
<input type="radio"/> 51 Development Kit (Smart metal) <input type="radio"/> Arduino Basic Kit <input type="radio"/> Arduino Sensor Kit
Smart Robot Car
<input type="radio"/> Arduino Smart Robot Car <input type="radio"/> Arduino Bat Robot Car <input type="radio"/> 51 Smart Robot Car
Balance Car
<input type="radio"/> Arduino Balance Car <input type="radio"/> STM32 Balance Car
4WD Smart Robot Car
<input type="radio"/> BST_4WD_Arduino <input type="radio"/> BST_4WD_51 <input type="radio"/> BST_4WD_STM32 <input checked="" type="radio"/> BST_4WD_Raspberry Pi
TrikeBot Smart Robot Car
<input type="radio"/> TrikeBot Smart Robot Car-STM32 <input type="radio"/> TrikeBot Smart Robot Car-Raspberry Pi

OK **CANCEL**

14. Click **[Connect Video]** to see the video picture captured by the camera.

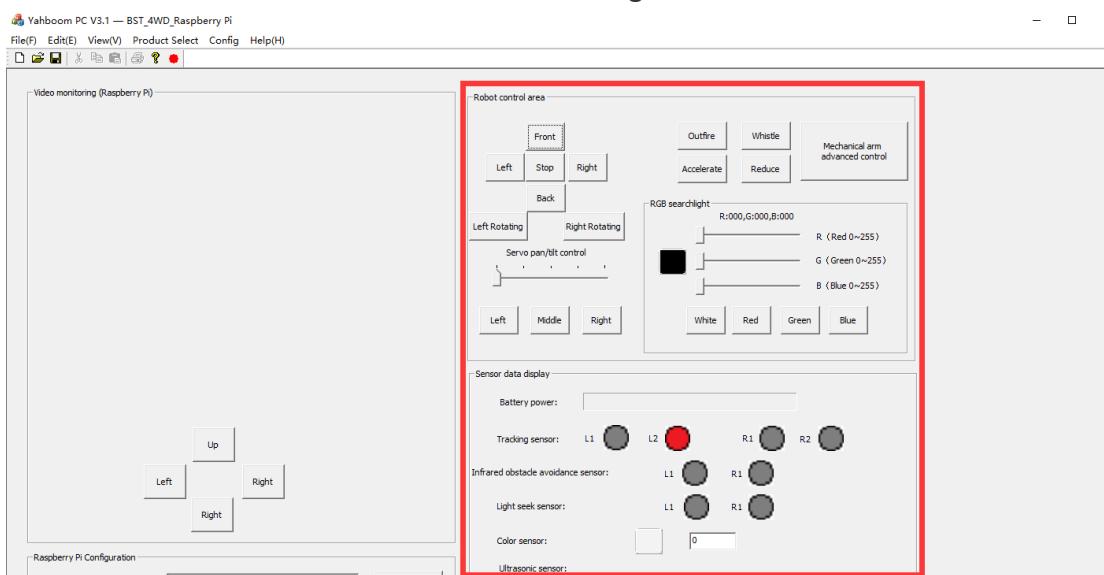


15. Click **[Connect Video]** again to close the video picture captured by the camera.

16. Click **[Connect control]** to enter remote control car mode.



17. You can control the car in the area on the right.



18. Click [Connect control] again to launch remote control car mode.

At the same time, the Raspberry Pi system will have the following tips:

```

pi@raspberrypi:~/SmartCar $ ./TCP_Control
Listen.....
=====
connect_fd : 5
client IP : 192.168.0.11
client port : 57819
$4WD,CSB0,PV8.4,GS0,LF1101,HW10,GM00#
send 37 bytes : $4WD,CSB0,PV8.4,GS0,LF1101,HW10,GM00#
Recv 19 bytes : $0,0,0,0,0,0,2,0,0#
Recv 19 bytes : $0,0,0,0,0,0,1,0,0#
[REPEATEDLY]

```