

Raspberry Pi Serial port communicates with external devices

The Raspberry Pi possess two serial ports available, the hardware serial port (/ dev / ttyAMA0) and the mini serial port (/ dev / ttyS0). The hardware serial port has a separate baud rate clock source with good performance and better stability. The mini serial port with simple functions and poor stability, the baud rate is provided by the CPU core clock, which is affected by the core clock. The Raspberry Pi (3B/B+/4B) has an on-board Bluetooth. The default hardware serial port is assigned to the Bluetooth module, and the mini serial port is assigned to the GPIO serial ports(TXD0/RXD0).

Raspberry Pi GPIO Header + PoE Header

Pin#	NAME	NAME	Pin#
01	3.3v DC Power	DC Power 5v	02
03	GPIO02 (SDA1 , I ² C)	DC Power 5v	04
05	GPIO03 (SCL1 , I ² C)	Ground	06
07	GPIO04 (GPIO_GCLK)	(TXD0) GPIO14	08
09	Ground	(RXD0) GPIO15	10
11	GPIO17 (GPIO_GEN0)	(GPIO_GEN1) GPIO18	12
13	GPIO27 (GPIO_GEN2)	Ground	14
15	GPIO22 (GPIO_GEN3)	(GPIO_GEN4) GPIO23	16
17	3.3v DC Power	(GPIO_GEN5) GPIO24	18
19	GPIO10 (SPI_MOSI)	Ground	20
21	GPIO09 (SPI_MISO)	(GPIO_GEN6) GPIO25	22
23	GPIO11 (SPI_CLK)	(SPI_CE0_N) GPIO08	24
25	Ground	(SPI_CE1_N) GPIO07	26
27	ID_SD (I ² C ID EEPROM)	(I ² C ID EEPROM) ID_SC	28
29	GPIO05	Ground	30
31	GPIO06	GPIO12	32
33	GPIO13	Ground	34
35	GPIO19	GPIO16	36
37	GPIO26	GPIO20	38
39	Ground	GPIO21	40

01	TR01	TR00	02
03	TR03	TR02	04

Run the following command to view the default serial port allocation method:

`ls /dev -al`

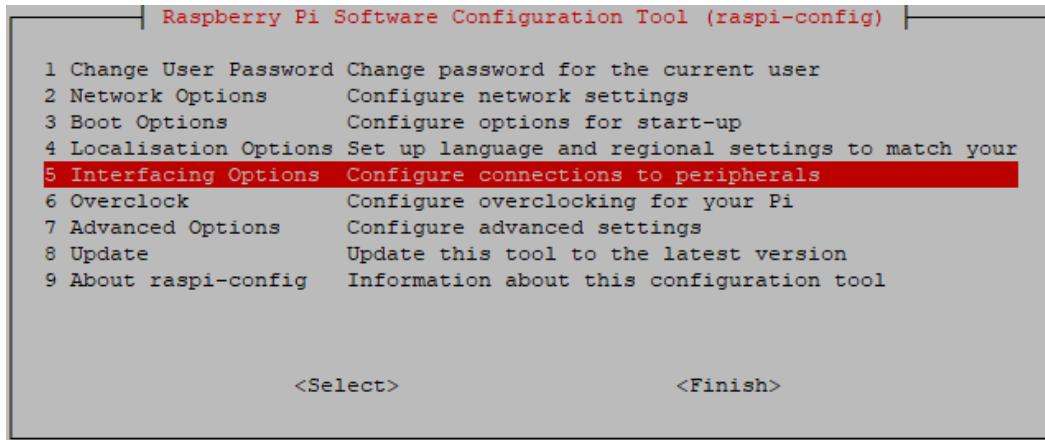
```
drwxr-xr-x  2 root root          60 Jan  1 1970 raw
crw-rw-r--  1 root netdev    10,  57 Aug 26 14:56 rfkill
lrwxrwxrwx  1 root root          5 Aug 26 14:56 serial0 -> ttyS0
lrwxrwxrwx  1 root root          7 Aug 26 14:56 serial1 -> ttyAMA0
drwxrwxrwt  2 root root         40 Feb 14 2019 Sdio
drwxr-xr-x  3 root root        160 Aug 26 14:56 snd
crw-rw----  1 root spi      153,   0 Aug 26 14:56 spidev0.0
crw-rw----  1 root spi      153,   1 Aug 26 14:56 spidev0.1
```

Since the hardware serial port is assigned to the on board Bluetooth by default, we will release it and assign the hardware serial port to the GPIO serial port.

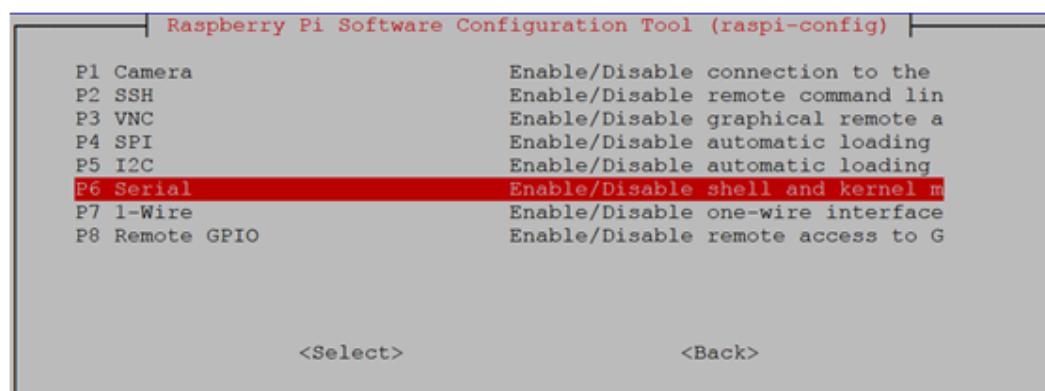
1. After we enter the Raspberry Pi system by SSH, we need to input following command.

sudo raspi-config

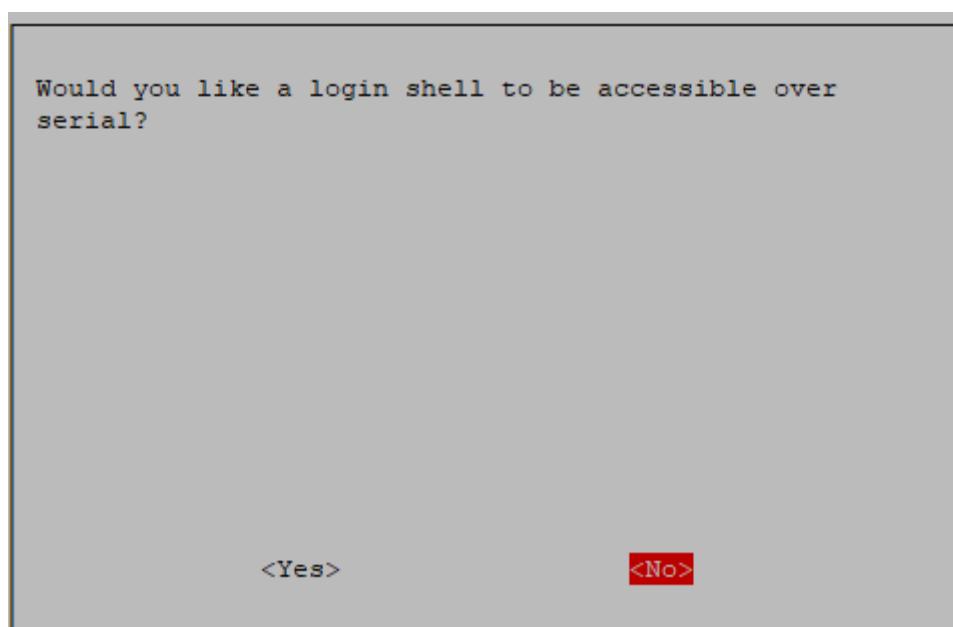
Enter the Raspberry Pi system configuration interface and select the fifth **【Interfacing Options】** as shown below.

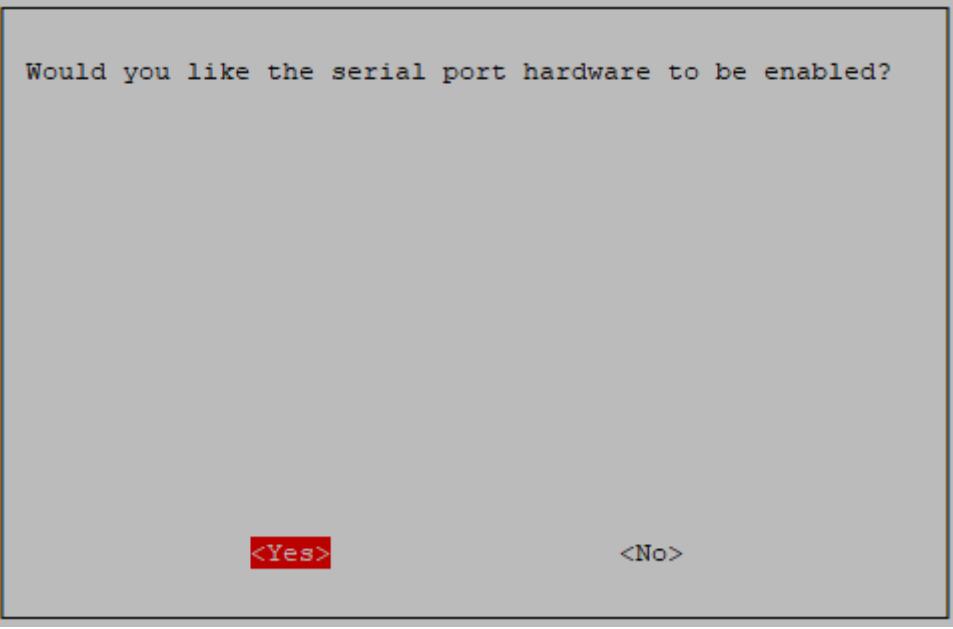


Select **【P6 Serial】**



Select to close the serial port login function and enable the hardware serial port debugging function.



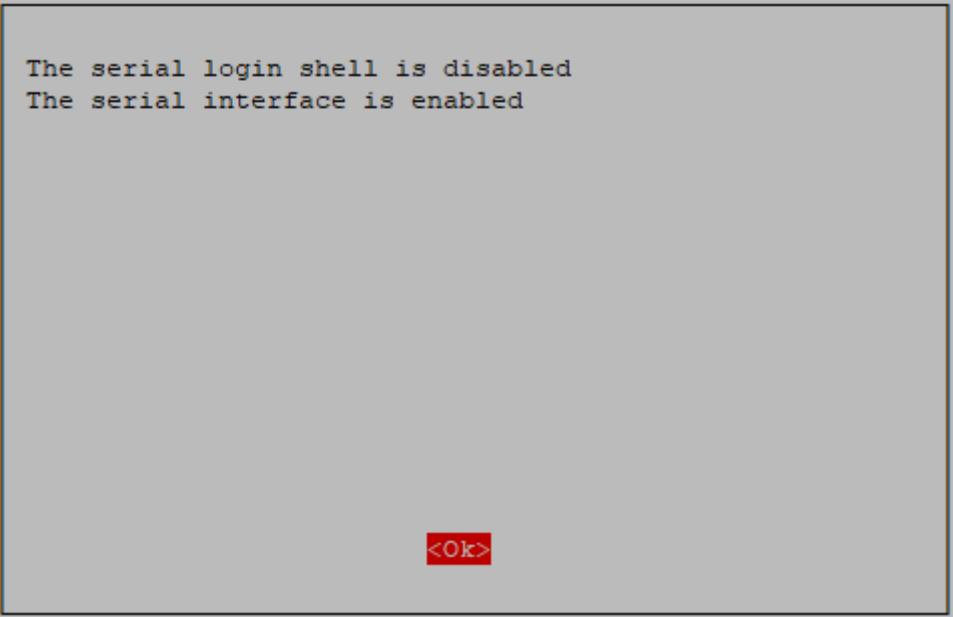


```
Would you like the serial port hardware to be enabled?
```

<Yes>

<No>

After the completion, Raspberry Pi interface will prompt following interface. Press "OK".



```
The serial login shell is disabled
The serial interface is enabled
```

<Ok>

Exit the raspi-config settings and restart the Raspberry Pi as prompted.

2. Set hardware serial port as GPIO serial port

Edit /boot/config.txt file the directory.

`sudo nano /boot/config.txt`

Add the following two lines command to the end of the file.

`dtoverlay=pi3-minuart-bt`

`force_turbo=1`

As shown below.

```
GNU nano 3.2                               /boot/config.txt

#dtoverlay=lirc-rpi

# Additional overlays and parameters are documented /boot/overlays/README

# Enable audio (loads snd_bcm2835)
dtparam=audio=on
start_x=1
gpu_mem=128

dtoverlay=pi3-miniuart-bt

force_turbo=1
```

Save: **Ctrl+O**

Exit: **Ctrl+X**

Input following command to restart Raspberry Pi

sudo reboot

After restart is complete, input **ls /dev -al**, we can see the two serial ports have changed positions. As shown below.

```
drwxr-xr-x  2 root root      60 Jan  1 1970 raw
crw-rw-r--  1 root netdev   10,  57 Aug 26 11:55 rfkill
lrwxrwxrwx  1 root root       7 Aug 26 11:55 serial0 -> ttyAMA0
lrwxrwxrwx  1 root root       5 Aug 26 11:55 serial1 -> ttyS0
drwxrwxrwt  2 root root      40 Feb 14 2019 bcm
drwxr-xr-x  3 root root     160 Aug 26 11:55 snd
crw-rw----  1 root spi     153,    0 Aug 26 11:55 spidev0.0
crw-rw----  1 root spi     153,    1 Aug 26 11:55 spidev0.1
```

3. mini com serial port test

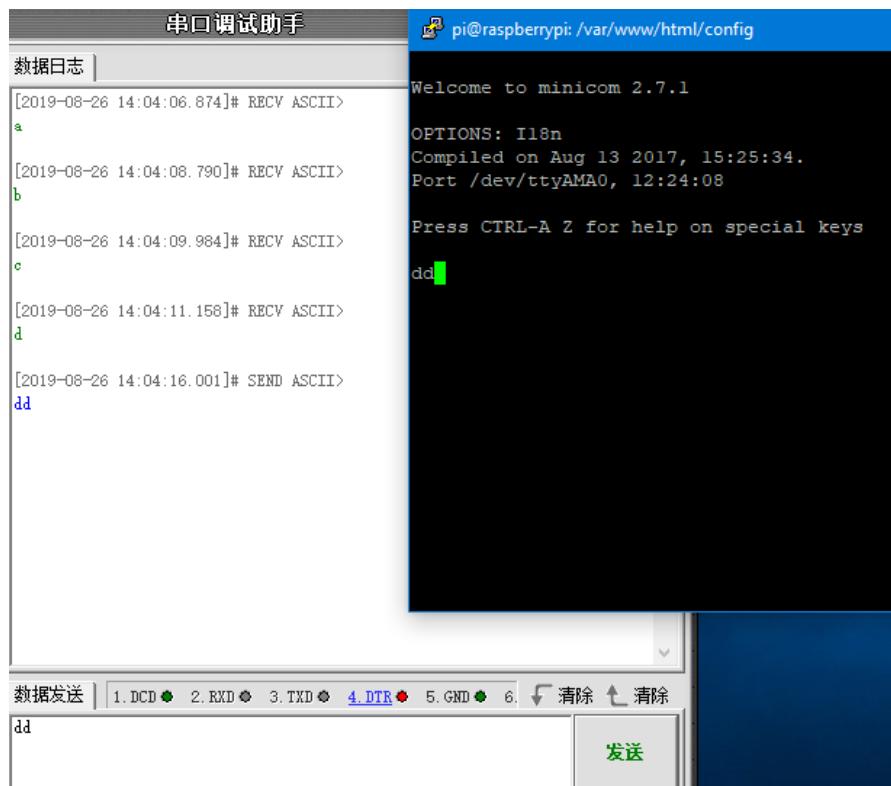
Input following command to install minicom

sudo apt-get install minicom

After installation is complete, we can input following command to start minicom

minicom -D /dev/ttyAMA0 -b 9600

-D indicates that the serial port /dev/ttyAMA0 is selected, and -b sets the baud rate to 9600. (This parameter didn't need to be set. The default is 115200.)



After serial port is opened, we can transfer data by USB to TTL module.

Exit minicom steps:

First, press “Ctrl+A”, then press “Z”, you can see following menu.

```
+-----+
| Minicom Command Summary
|
| Commands can be called by CTRL-A <key>
|
| Main Functions           Other Functions
|
| Dialing directory...D   run script (Go)....G | Clear Screen.....C
| Send files.....S        Receive files.....R | cOnfigure Minicom..O
| comm Parameters....P    Add linefeed.....A | Suspend minicom....J
| Capture on/off.....L   Hangup.....H | eXit and reset....X
| send break.....F        initialize Modem...M | Quit with no reset.Q
| Terminal settings...T   run Kermit.....K | Cursor key mode....I
| lineWrap on/off....W   local Echo on/off..E | Help screen.....Z
| Paste file.....Y        Timestamp toggle...N | scroll Back.....B
| Add Carriage Ret...U
|
| Select function or press Enter for none.
+-----+
```

Next, press “X”. Finally, select “YES” and press “Enter” key.

USB to TTL module connect to Raspberry Pi.

Raspberry Pi	USB to TTL module
5v	VCC
GND	GND

RXD0	RXD
TXD0	TXD

4.C language reference code, print hello world!

```
#include <stdio.h>
#include <wiringPi.h>
#include <wiringSerial.h>
int main()
{
    int fd;
    if(wiringPiSetup() < 0) return 1;

    if((fd = serialOpen("/dev/ttyAMA0",9600)) < 0) return 1;

    printf("serial test start ...n");

    serialPrintf(fd,"Hello World!!!n");
    while(1)
    {
        serialPutchar(fd,serialGetchar(fd));
    }
    serialClose(fd);
    return 0;
}
```

```
#include <stdio.h>
#include <wiringPi.h>
#include <wiringSerial.h>

int main()
{
    int fd;
    if(wiringPiSetup() < 0) return 1;

    if((fd = serialOpen("/dev/ttyAMA0",9600)) < 0) return 1;

    printf("serial test start ...\\n");

    serialPrintf(fd,"Hello World!!!\\n");

    while(1)
    {
        serialPutchar(fd,serialGetchar(fd));
    }
    serialClose(fd);

    return 0;
}
```

Input following command to new create [testCom.c](#) file

```
nano testCom.c
```

Copy above code into testCom.c file.

Save: **Ctrl+O**

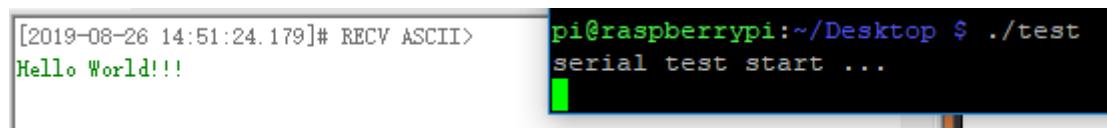
Exit: **Ctrl+X**

Input following command to compile this program.

```
gcc testCom.c -o test -lwiringPi
```

Input following command to run code

```
./test
```



A terminal window titled "pi@raspberrypi:~/Desktop \$./test" is shown. On the left side, there is a green text area containing the command "RECV ASCII>" and the output "Hello World!!!". On the right side, there is a black text area containing the command "serial test start ..." followed by three dots indicating continuation.

You can send data to the Raspberry Pi by serial port, the Raspberry Pi will directly return to the serial port for display.

Note: If garbled characters are displayed, please check the baud rate. It needs to be set to 9600(the same as program).