

## Serial control servo

### 1. Learning goals

In this course, we mainly learn to use Raspberry Pi and 16-channels servo debugging board to control the servo through serial port.

### 2. Preparation

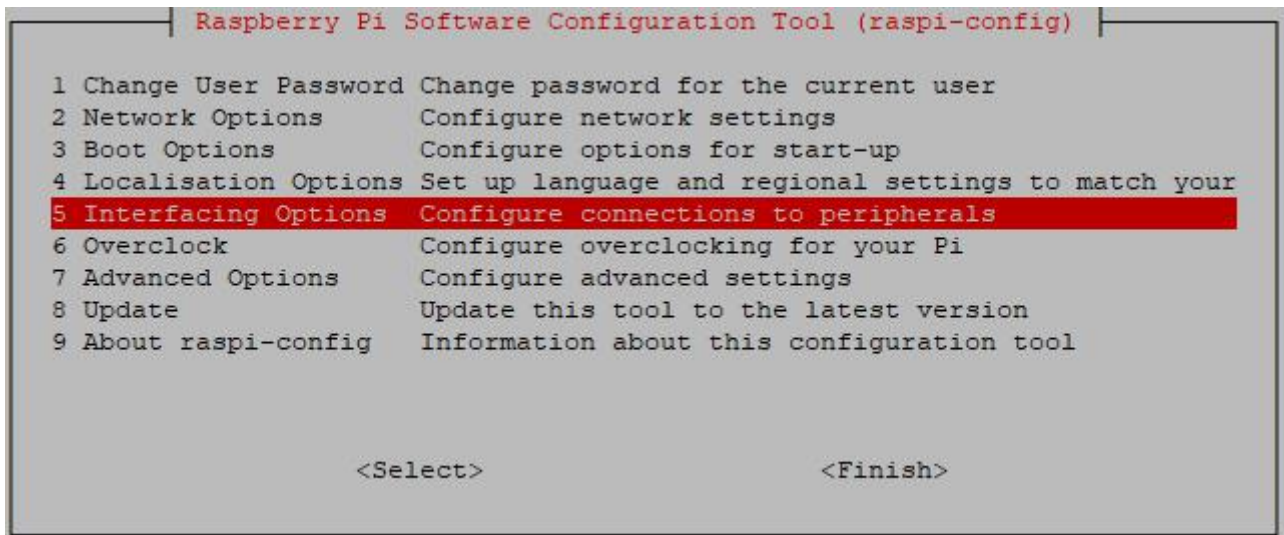
Connect the TXD and RXD of the module to the IO15 and IO14 pins of the Raspberry Pi board. VCC and GND are connected to 3.3V and GND of Raspberry Pi board respectively. As shown below.

Raspberry Pi GPIO Header + PoE Header				
Pin#	NAME		NAME	Pin#
01	3.3v DC Power		DC Power 5v	02
03	GPIO02 (SDA1 , I <sup>2</sup> C)		DC Power 5v	04
05	GPIO03 (SCL1 , I <sup>2</sup> 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 <sup>2</sup> C ID EEPROM)		(I <sup>2</sup> 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

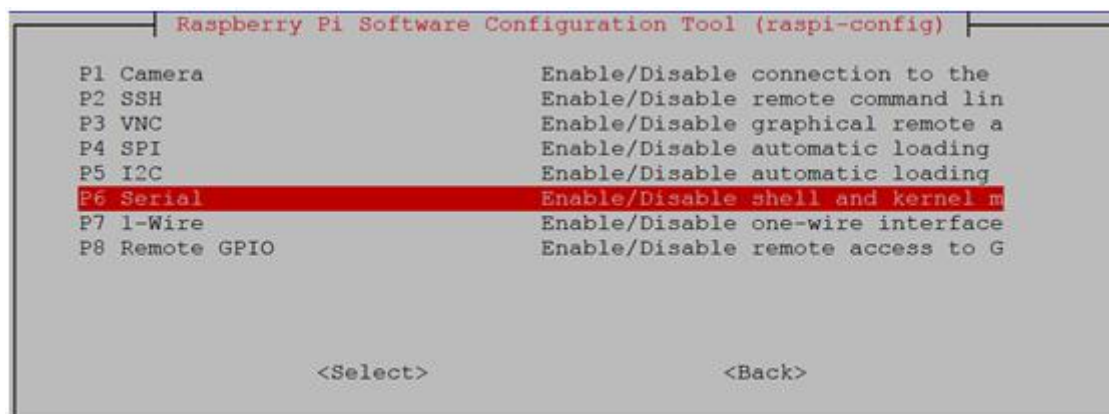
### 3. Assign the serial port of the Raspberry Pi

3.1 Raspberry Pi needs to assign ttyAMA0 port to GPIO serial port TXD0, RXD0, the specific method is as follows.

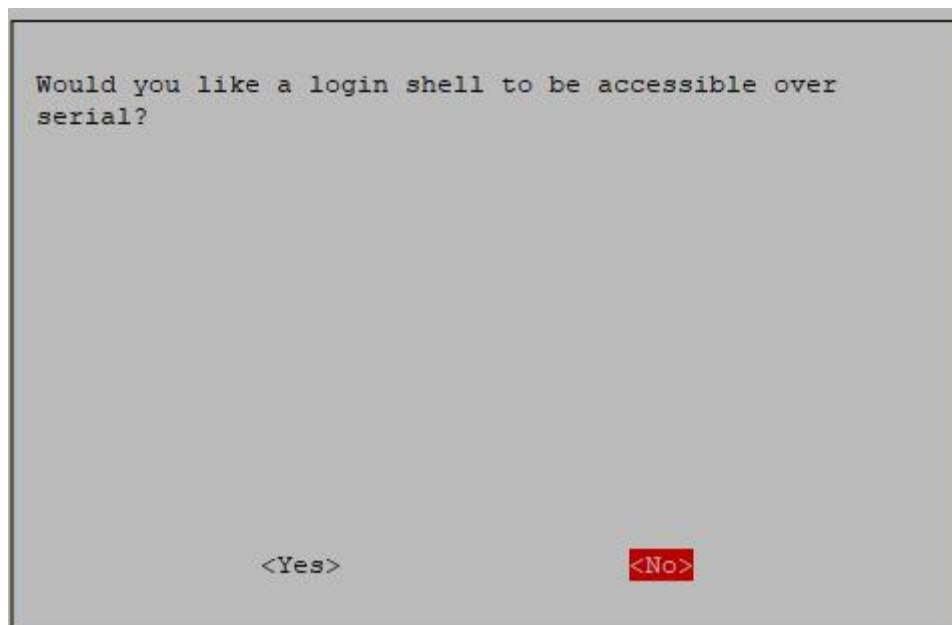
Enter `sudo raspi-config` to enter the Raspberry Pi system configuration interface, and select the fifth Interfacing Options:

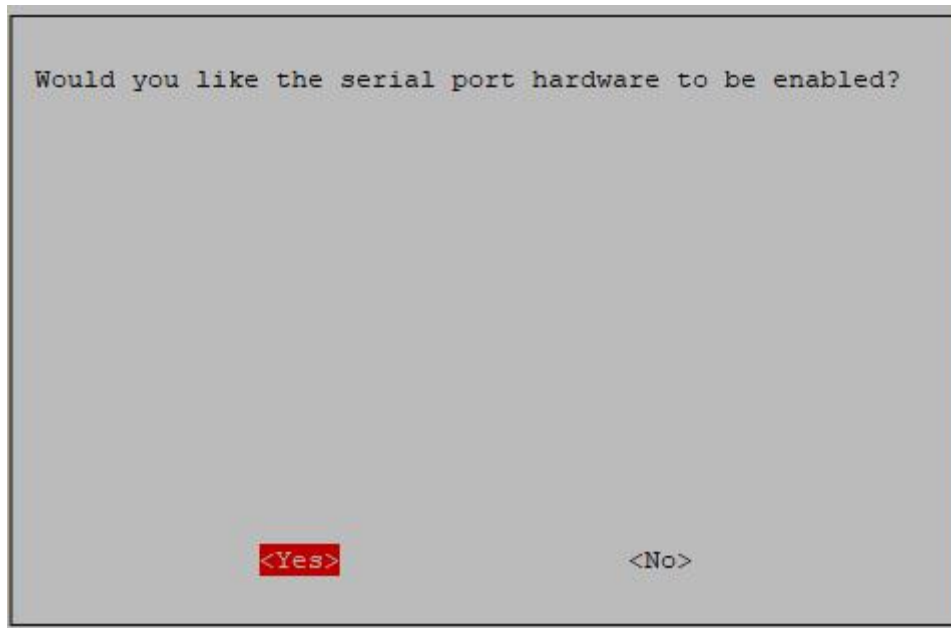


Enter "P6 Serial".



Choose to close the serial port log in function and open the hardware serial port debugging function.





Click "OK".



Exit raspi-config settings, and restart the Raspberry Pi according to the prompts.

3.2 Edit the **config.txt** file in the /boot directory: `sudo nano /boot/config.txt`

Add the following two lines to the end:

`dtoverlay=pi3-miniuart-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

```

3.3 Press "Ctrl+O" save file. Press "Ctrl+X" save file.

3.4 Input following command to restart Raspberry Pi.

**sudo reboot**

After restarting the Raspberry Pi, enter **ls /dev -al**, you can see that 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 shm
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

```

#### 4. Code

Please refer to **16CServo-uart.py**

Configure the serial port.

```
ser = serial.Serial("/dev/ttyAMA0", 9600)
```

Serial port control servo function

```

def UARTServo(servonum, angle):
    servonum = 64 + servonum
    date1 = int(angle/100 + 48)
    date2 = int((angle%100)/10 + 48)
    date3 = int(angle%10 + 48)
    cmd=bytearray([36,servonum,date1,date2,date3,35])
    ser.write(cmd)
    time.sleep(0.05)

```

Set the servo S1 angle to 0

```
UARTServo(1,0)
```

#### 4. Running code

Input following command in the terminal to run the program.

```
python 16CServo-uart.py
```

#### 5. Phenomenon

After the program is run successfully. The servo will rotate 0°, after 2s it will rotate 180°.