External device serial port communication

1. Preparation

This section tests the serial port's self-transmitting and receiving functions of Jetson Nano. The serial port device number of Jetson Orin NX and Jetson Orin Nano motherboards needs to be modified to /dev/ttyTHS0.

From the figure below, you can see that the TXD and RXD pins of the serial port of JETSON NANO correspond to physical pins 8 and 10 respectively.

BCM编码	功能名	物理引脚		功能名	BCM编码
	3V3	1	2	5V	
2	SDA	3	4	5V	
3	SCL	5	6	GND	
4	D4	7	8	D14(TXD)	14
	GND	9	10	D15(RXD)	15
17	D17	11	12	D18	18
27	D27	13	14	GND	
22	D22	15	16	D23	23
	3V3	17	18	D24	24
10	D10	19	20	GND	
9	D9	21	22	D25	25
11	D11	23	24	D8	8
	GND	25	26	D7	7
0	DO(ID_SD)	27	28	D1(ID_SC)	1
5	D5	29	30	GND	
6	D6	31	32	D12	12
13	D13	33	34	GND	
19	D19	35	36	D16	16
26	D26	37	38	D20	20
	GND	39	40	D21	21

Wiring:
JETSON NANO pin 8 (TXD) → JETSON NANO pin 10 (RXD)

2. Use the routine to test the serial port function

Download a serial port communication case from github. The source code and explanation can be viewed at this github link

```
git clone https://github.com/JetsonHacksNano/UARTDemo
```

Enter the folder

```
cd UARTDemo
```

If you do not plan to use the serial console on the UART, you should disable the serial console (it depends on your choice):

```
systemctl stop nvgetty
systemctl disable nvgetty
udevadm trigger
```
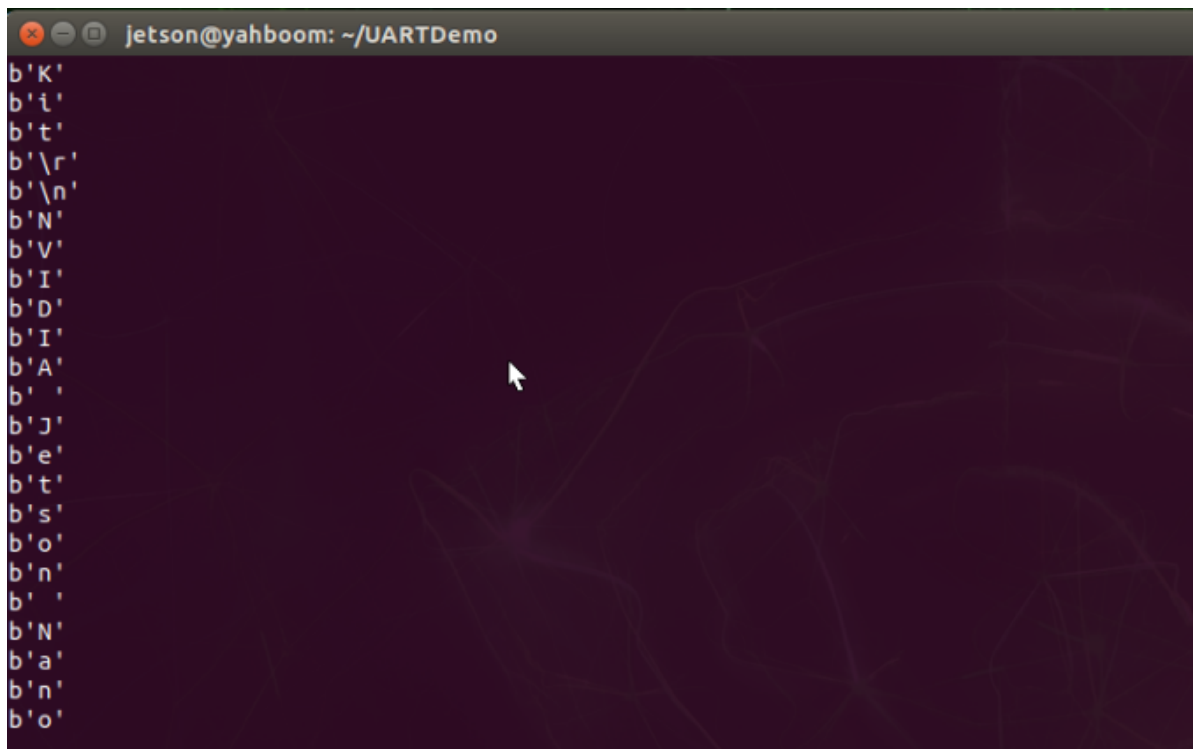
Install serial module

```
sudo apt-get install python3-serial
```

Enter the terminal to run the program:

```
sudo python3 uart_example.py
```


After running, you can see that Jetson Nano sends the characters in "NVIDIA Jetson Nano Developer Kit\r\n" in a loop and displays them on the terminal.

If you are using Jetson Orin NX and Jetson Orin Nano motherboards, you need to change the serial port device number to /dev/ttyTHS0, and you can communicate normally after running the program.

```
print("UART Demonstration Program")
print("NVIDIA Jetson Nano Developer Kit")


serial_port = serial.Serial(
    port="/dev/ttyTHS0",
    baudrate=115200,
    bytesize=serial.EIGHTBITS,
    parity=serial.PARITY_NONE,
    stopbits=serial.STOPBITS_ONE,
)
# Wait a second to let the port initialize
time.sleep(1)
```

2. Test using Linux serial port assistant
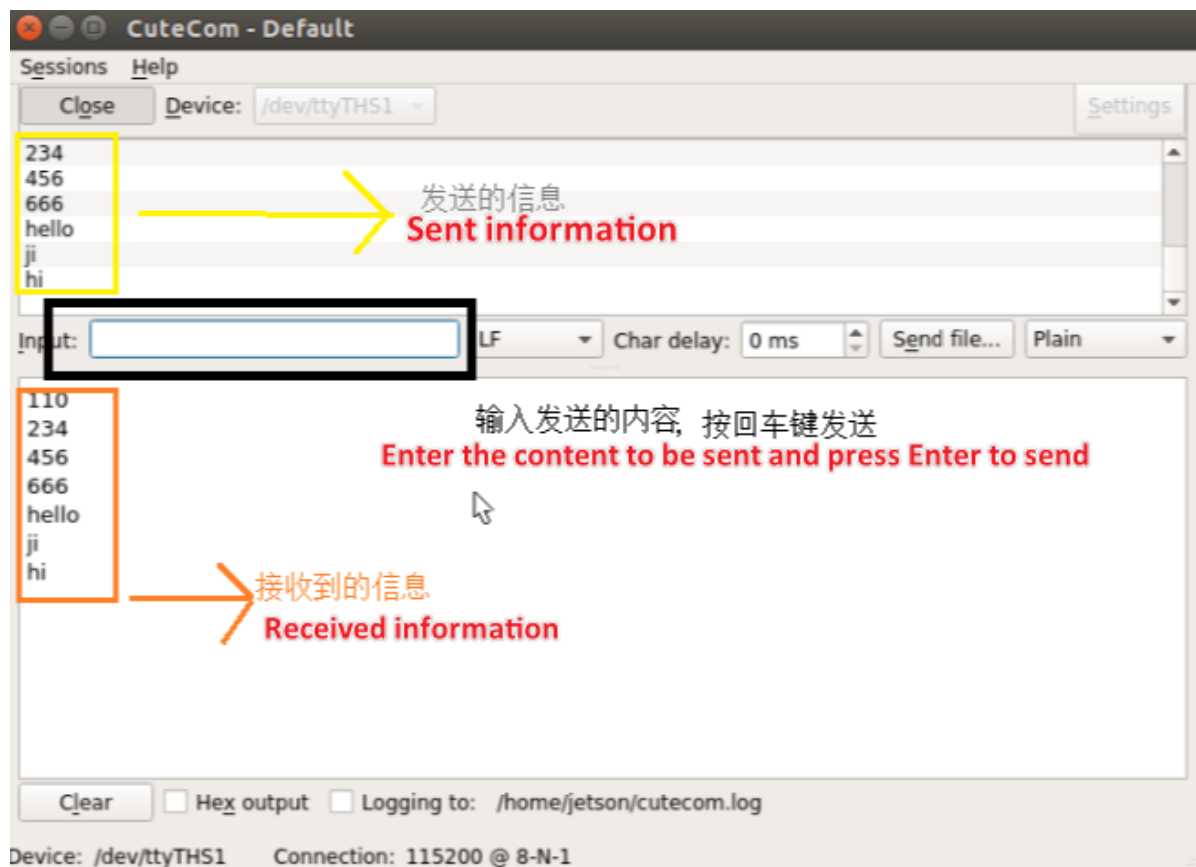
1. Run the following command

```
sudo apt install cutecom
sudo cutecom
```

You can see cutecom is open
Generally, no settings are required, just click **open** to use it, then enter text through InPUT, and press Enter to send the content.
The effect is as shown in the figure:

3. Notes

If you use the USB to TTL module to communicate between the computer and nano, pay attention to the following points

1. The Dupont line cannot be too long, otherwise it will cause garbled characters

2. The situation where you can only receive but not send is caused by insufficient voltage. Connect the 5V port of the USB to TTL module to the 5V of the nano
3. If the line is reasonable, but garbled characters appear, check whether the baud rate, parity check, and stop bit are consistent
4. Other situations: https://blog.csdn.net/lxj362343/article/details/89646731