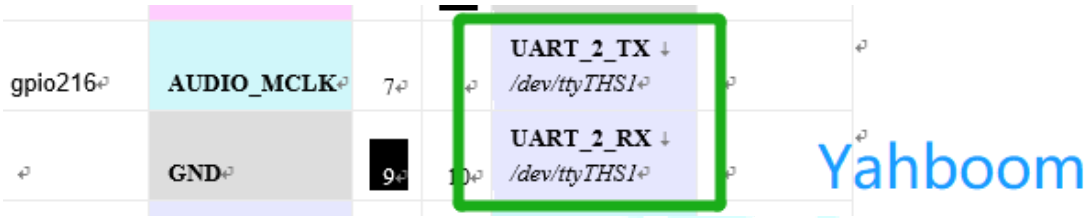
4.7 Use of PTZ servos

1. Introduction to PTZ servo driving

The drive of the PTZ servo is built on the serial communication. In the introduction of the 【5.1 Using of GPIO 】, we learned that serial servo uses UART2 serial 2 TX, because the servo only receives data, so we only use Go to the UART_2_TX pin of the serial port, which is mounted on the /dev/ttyTHS1 device of the Jetbot Ubuntu 18.04 system.

!!!Note: If you are not using the Yahboom_jetbot_car_image provided by Yahboom, please confirm whether you have enabled the usage rights in accordance with 【2.4 Install Jetbot】--【1. Enable peripheral permissions we will use】



About specific drive functions in the PTZ serial servo bus :

The PTZ serial bus servo drive has imported the following module package.

```
import RPi.GPIO as GPIO
import time
import string
import serial
```

Different from the previous one, the serial port module and the string processing module are introduced, and the string processing module is used to process the command stacking process for controlling the serial port bus servo.

When the module is called to create an instance object, the initialization function is automatically run to initialize the driver.

After the initialization is completed, the "Serial Open!" string prompt will be printed on the console. When the program exits, the __del__() function will be automatically run to close the serial port and the "serial Close!" prompt will be printed.

```
    def __init__(self):
        self.ser = serial.Serial("/dev/ttyTHS1", 115200, timeout = 0.001)
        print ("serial Open!")
    def __del__(self):
        self.ser.close()
        print ("serial Close!")
```

Control example method for a single servo:

```
Servo_serial_control(self, index, angle)
```

Two motor control example methods:

```
Servo_serial_double_control(self, index_1, angle_1, index_2, angle_2)
```

The code shown in the following figure exists in both methods:

```
    #Keep the steering pulse of the servo within a safe range
    if index == 1:
        if angle<600:
            angle=600
        elif angle>3600:
            angle = 3600
    elif index == 2:
        if angle<1300:
            angle=1300
        elif angle>4095:
            angle=4095
```

About Serial bus servo protocol:

Path of package : [Jetbot-AI Car] --> [Annex] --> [Hardware]---> [Serial bus servo protocol file]

2. Use the servo drive the asynchronous write command to control the servo

We import the package that needs to be used, and then create an instance of the PTZ bus servo. The PTZ bus servo is initialized by default in the background.

Import Yahboom official packaged serial bus servo drive library

Create and initialize a programmable servo control object

```
from servoserial import ServoSerial
import time
servo_device = ServoSerial()
```

Yahboom

Next, we can start using the example method to control the steering wheel rotation.

Make servo rotate to the center position by asynchronous commands

```
servo_device.Servo_serial_control(1, 2100)
time.sleep(0.1)
servo_device.Servo_serial_control(2, 2048)
```

Make servo rotate to 1300 by asynchronous commands

```
servo_device.Servo_serial_control(1, 1300)
time.sleep(0.1)
servo_device.Servo_serial_control(2, 1300)
```

Make servo rotate to 3600 by asynchronous commands

```
servo_device.Servo_serial_control(1, 3600)
time.sleep(0.1)
servo_device.Servo_serial_control(2, 3600)
```



```
servo_device.Servo_serial_control(1, 2100)
```

servo_serial_control(self, index, angle)

Indenx = 1, it means that the PTZ servo servo is controlled to move in the left and right direction (X-axis direction).

Indenx = 2, it means that the PTZ servo servo is controlled to move in the up and down direction (Y-axis direction).

As the pan/tilt rotation described in the previous section is limited to the safe range, the values we set are within this range for efficient operation:
The motor in the left and right direction, ie index = 1, moves in the range of 600 – 3600
The motor in the up and down direction, ie index = 2, moves in the range 1300 – 4095

The corresponding complete source code is located at:
/home/jetbot/Notebook/8.Using of PTZ servos bus/Asynchronous control servo.ipynb

3. Use the servo drive the synchronous write command to control the servo

Sometimes when we move the PTZ, both motors need real-time control.
In order to achieve higher real-time responsiveness of the servo and avoid data jam on the export bus, we added a synchronous write command to control the servo based on asynchronous communication.

One command can control two servos at the same time, so that when the PTZ servo is used for real-time motion, its sensitivity is greatly improved.

The code is shown below:

Import Yahboom official packaged serial bus servo drive library

Create and initialize a programmable servo control object

```
from servoserial import ServoSerial
import time
servo_device = ServoSerial()
```

Make servo rotate to the center position by synchronize commands

```
servo_device.Servo_serial_double_control(1, 2100, 2, 2048)
```

Make servo rotate to 1300 by synchronize commands

```
servo_device.Servo_serial_double_control(1, 1300, 2, 1300)
```


Make servo rotate to 3600 by synchronize commands

```
servo_device.Servo_serial_double_control(1, 3600, 2, 3600)
```

servo_serial_double_control (self, index_1, angle_1, index_2, angle_2)

Indenx = 1, it means that the PTZ servo servo is controlled to move in the left and right direction (X-axis direction).

Indenx = 2, it means that the PTZ servo servo is controlled to move in the up and down direction (Y-axis direction).

The other procedures used are consistent with the asynchronous write command control servos introduced above.

The corresponding complete source code is located at: /home/jetbot/Notebook/8.Using of PTZ servos bus/Synchronous control servo.ipynb