STM32-Serial Port Method

Experimental preparation

1. STM32 motherboard
2. 8-channel patrol module
3. Several Dupont cables

STM32 needs to download the serial communication source code provided in the document

Experimental purpose

The content of this experiment is mainly to use the STM32 master to receive the data of the 8-channel patrol module through the serial port.

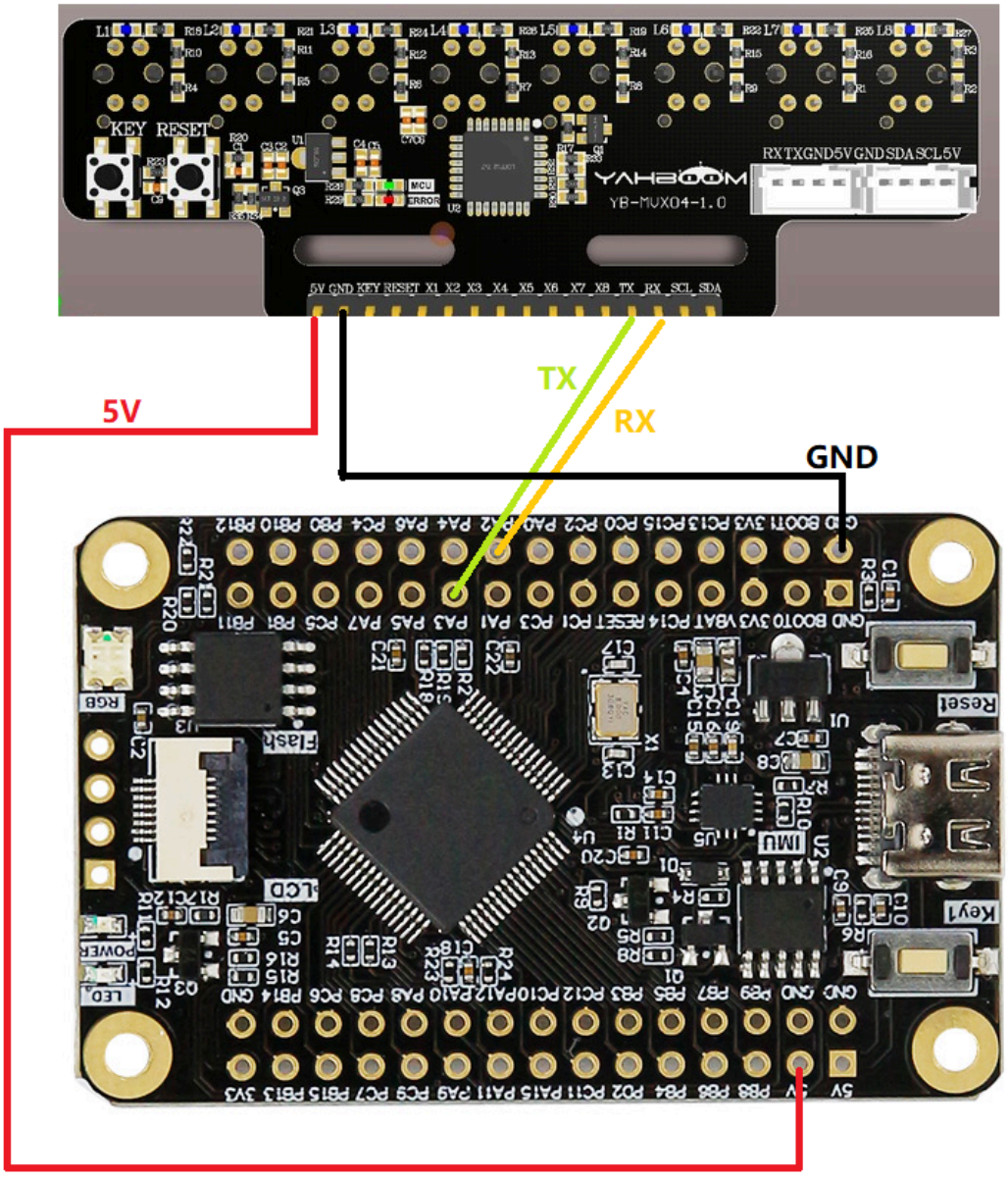
Experimental wiring

STM32 to serial port assistant
If you are not using the black stm32 of Yabo, you need to use a usb to ttl module to connect to the computer. The wiring is described in the following table

stm32	usb to ttl
PA10	TX
PA9	RX
vcc	vcc
GND	GND
If you are using the Yabo black stm32, you can directly use type-c to connect to the computer's serial port assistant	

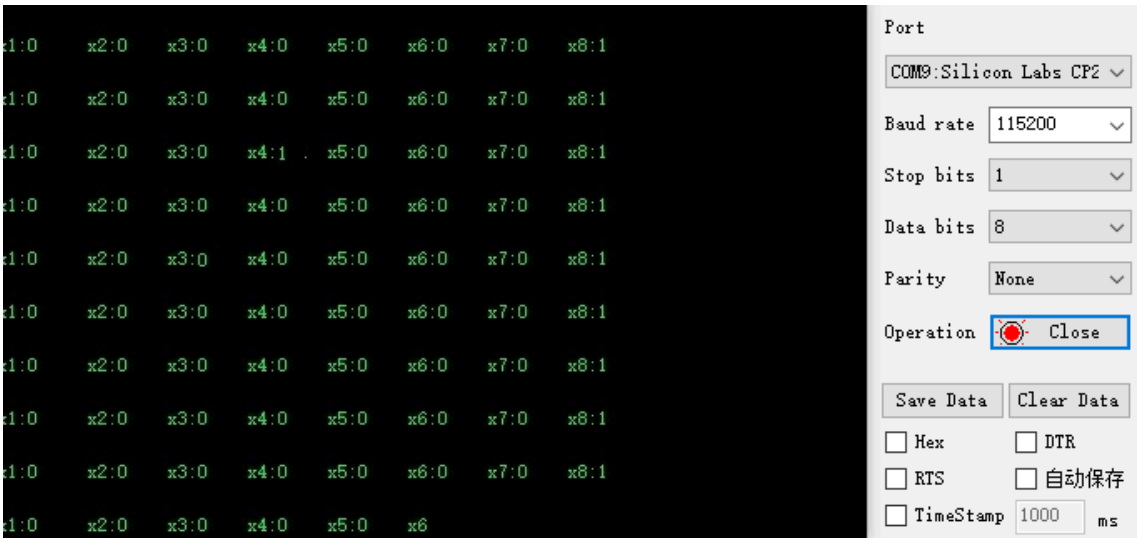
STM32	8-channel line patrol module
PA2	RX
PA3	TX
5V	5V
GND	GND

As shown in the figure:



Experimental steps and phenomena

1. After connecting the wires, open the serial port assistant and you can see the numerical data of the infrared module. Set the baud rate to 115200.
   As shown below



Experimental source code

```
int main()
{

    SystemInit();
    delay_init();

    uart_init(115200);
    USART2_init(115200);

    delay_ms(1000);//Wait for infrared to be normal
    send_control_data(0,0,1); //Just send the corresponding command according to
the protocol. This only receives numerical data.

    while(1)
    {
        if(g_new_package_flag == 1)
        {
            g_new_package_flag = 0;

            if(g_Dmode_Data == 1)
            {
                Deal_Usart_Data();//Numerical data processing

printf("x1:%d,x2:%d,x3:%d,x4:%d,x5:%d,x6:%d,x7:%d,x8:%d\r\n",IR_Data_number[0],IR
_Data_number[1],IR_Data_number[2],IR_Data_number[3],IR_Data_number[4],IR_Data_num
ber[5],IR_Data_number[6],IR_Data_number[7]);
            }
            if(g_Amode_Data == 1)
            {
                Deal_Usart_AData();//Analog data processing

printf("x1:%d,x2:%d,x3:%d,x4:%d,x5:%d,x6:%d,x7:%d,x8:%d\r\n",IR_Data_Anglo[0],IR_
Data_Anglo[1],IR_Data_Anglo[2],IR_Data_Anglo[3],IR_Data_Anglo[4],IR_Data_Anglo[5]
,IR_Data_Anglo[6],IR_Data_Anglo[7]);
```

```
            }

        }


    }
}
```

SET_Eight_Mode(0,0,1);： The first parameter of this function is the calibration mode (0: exit calibration mode 1: enter calibration mode) The second parameter is whether to receive analog data The third parameter is whether to receive numerical data.
This routine only provides parsing of numerical data. If you need to parse analog data, you can parse it yourself according to the protocol. The serial port parsing file of this project also has a function for parsing analog values. You can refer to it and call the **Deal_Usart_AData** function.