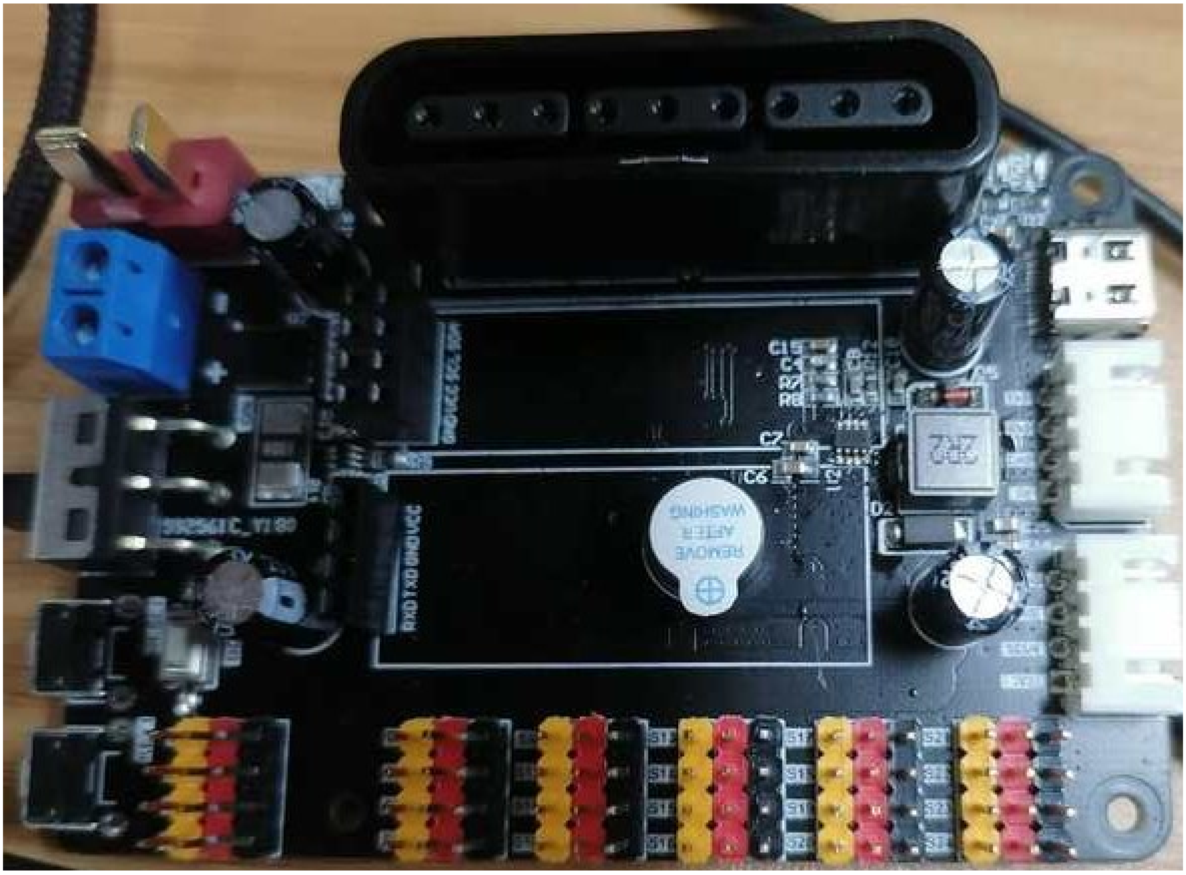
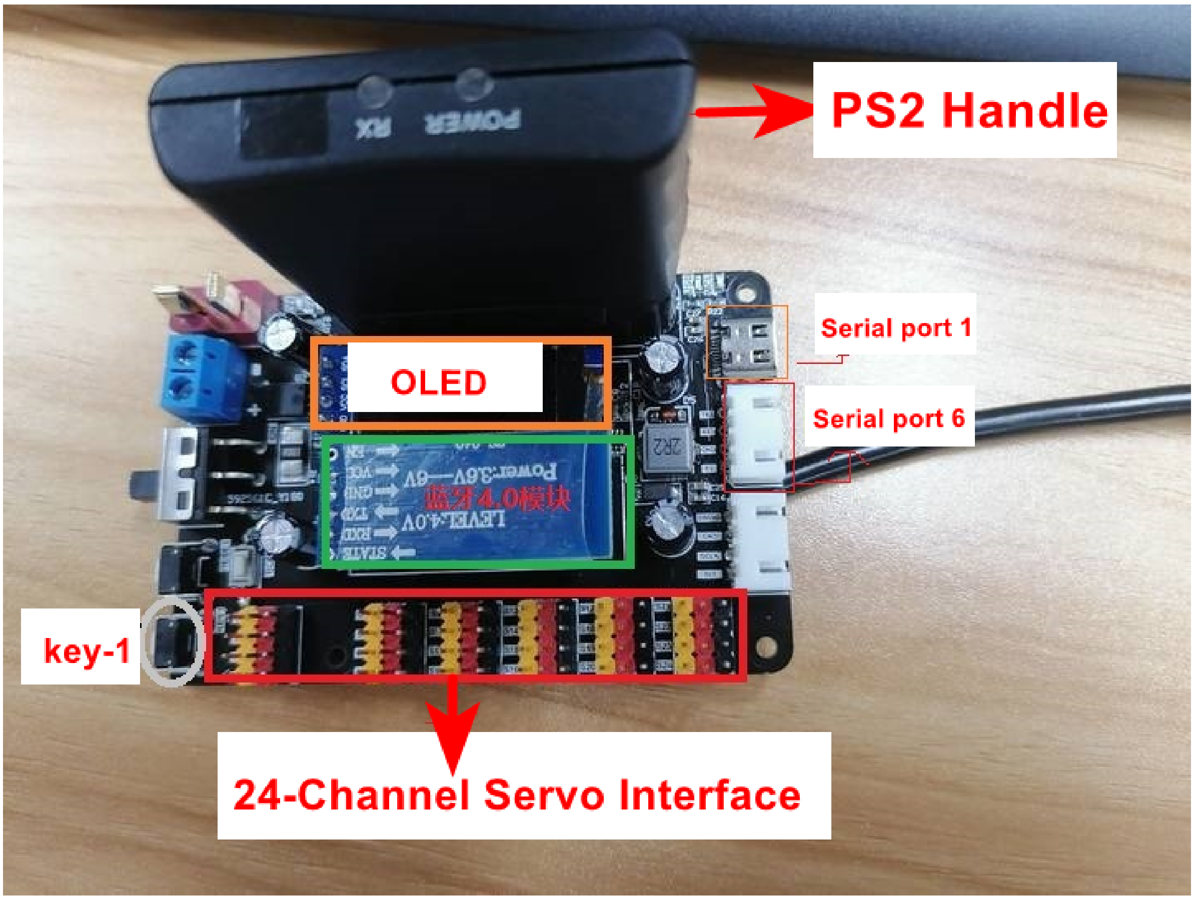
24-Channel Servo Drive Board

1.Hardware Preparation

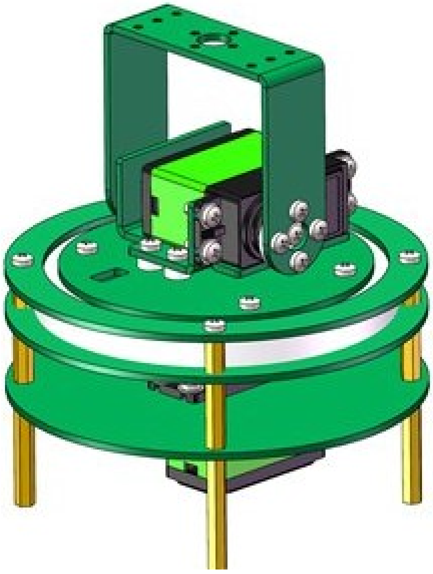
*** 24-channel steering gear control board ***



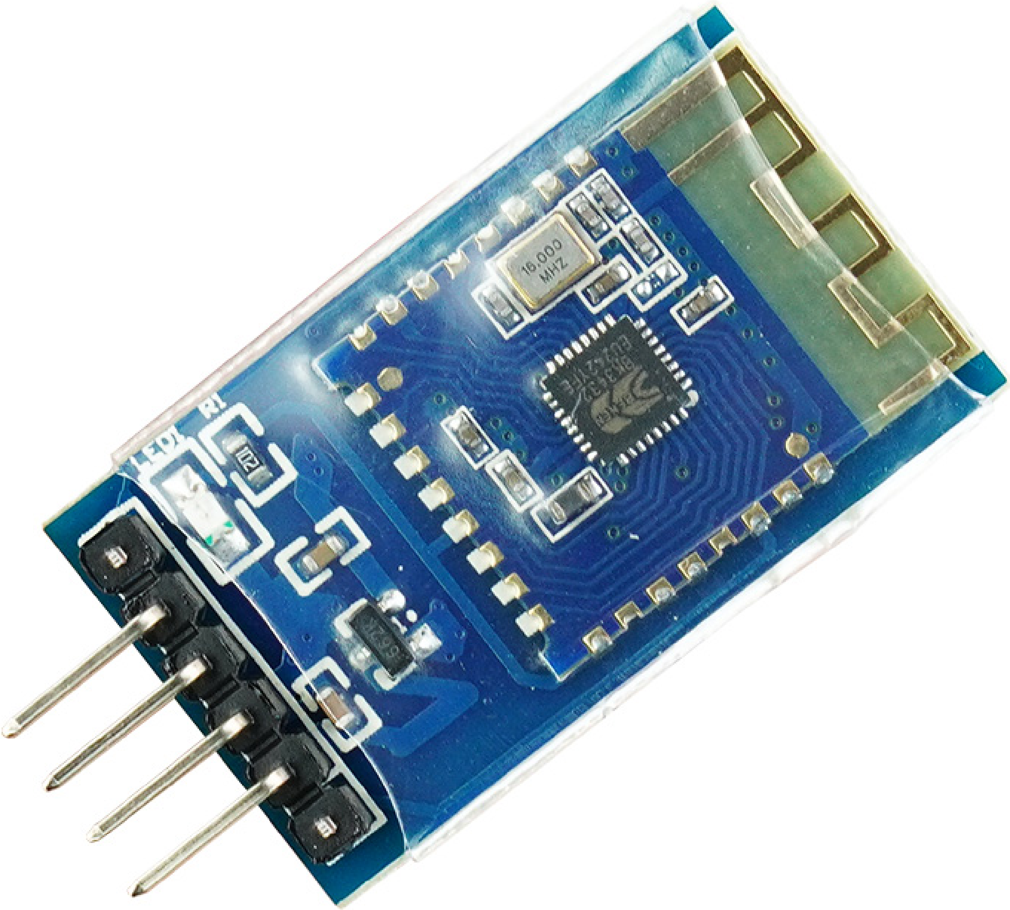
As shown below.



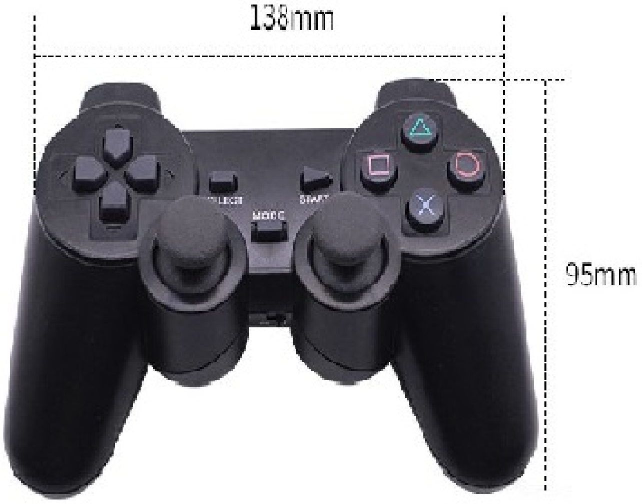
*** 2DOF steering gear PTZ *1 ***



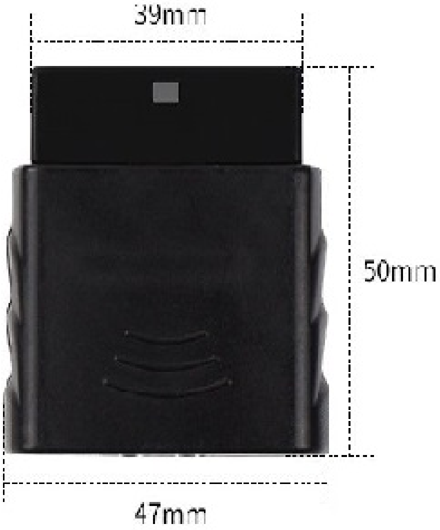
*** Bluetooth module *1 ***



*** ps2 handle ***



Controller



Receiver

*** 7.4V battery *1 ***

2.Hardware Introduction

2.1 24-Channel Servo Drive Board

The control chip used in this control board is STM32RCT6, with 32 * 128 OLED interface, Bluetooth 4.0 interface, PS2 handle interface and 24-channel PWM steering gear interface.

2.2 2DOF PTZ

PTZ is controlled up and down by a 180° servo, and left and right by a 270° servo. The voltage at which the two servo work normally is 6-7.4V.

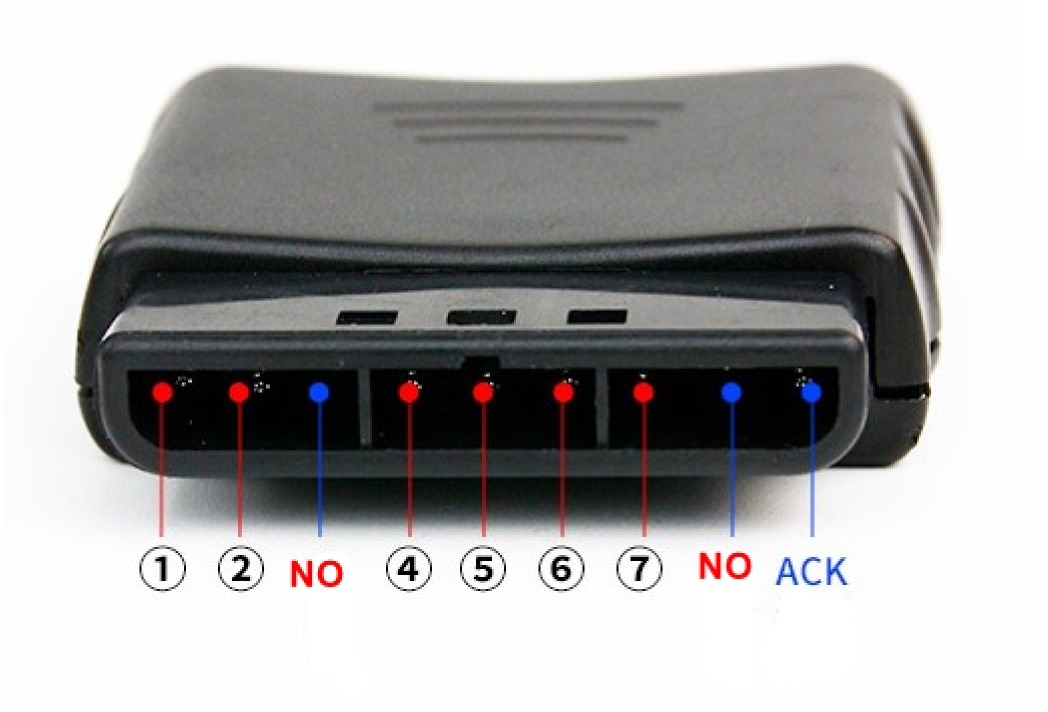
2.3 Introduction to Bluetooth module

It has the following advantages

- Support standard BLE protocol;
- Support UART and I2C interfaces;
- Support low power consumption mode;
- Support Bluetooth Class1 and Class2 modes;
- Industrial grade design;
- Data encryption;
- Built in PCB antenna;
- Input voltage: 3V;

2.4 Introduction to PS2 handle

DI/DAT: Signal flow direction, from the handle to the host, this signal is an 8-bit serial data that is transmitted synchronously in the falling edge of clock. The reading of the signal is done during the course of the clock from high to low.
DO/CMD: Signal flow direction, from the host to the handle. This signal is opposite to DI. The signal is an 8-bit serial data that is synchronously transmitted on the falling edge of the clock.
NC: empty port.
GND: power ground.
VDD: receiver working power supply, power supply range 3~5V;
CS/SEL: Used to provide the handle trigger signal. During communication, at a low level;
CLK: clock signal, sent by the host to keep data synchronized;
NC: empty port;
ACK: The response signal from the handle to the host.

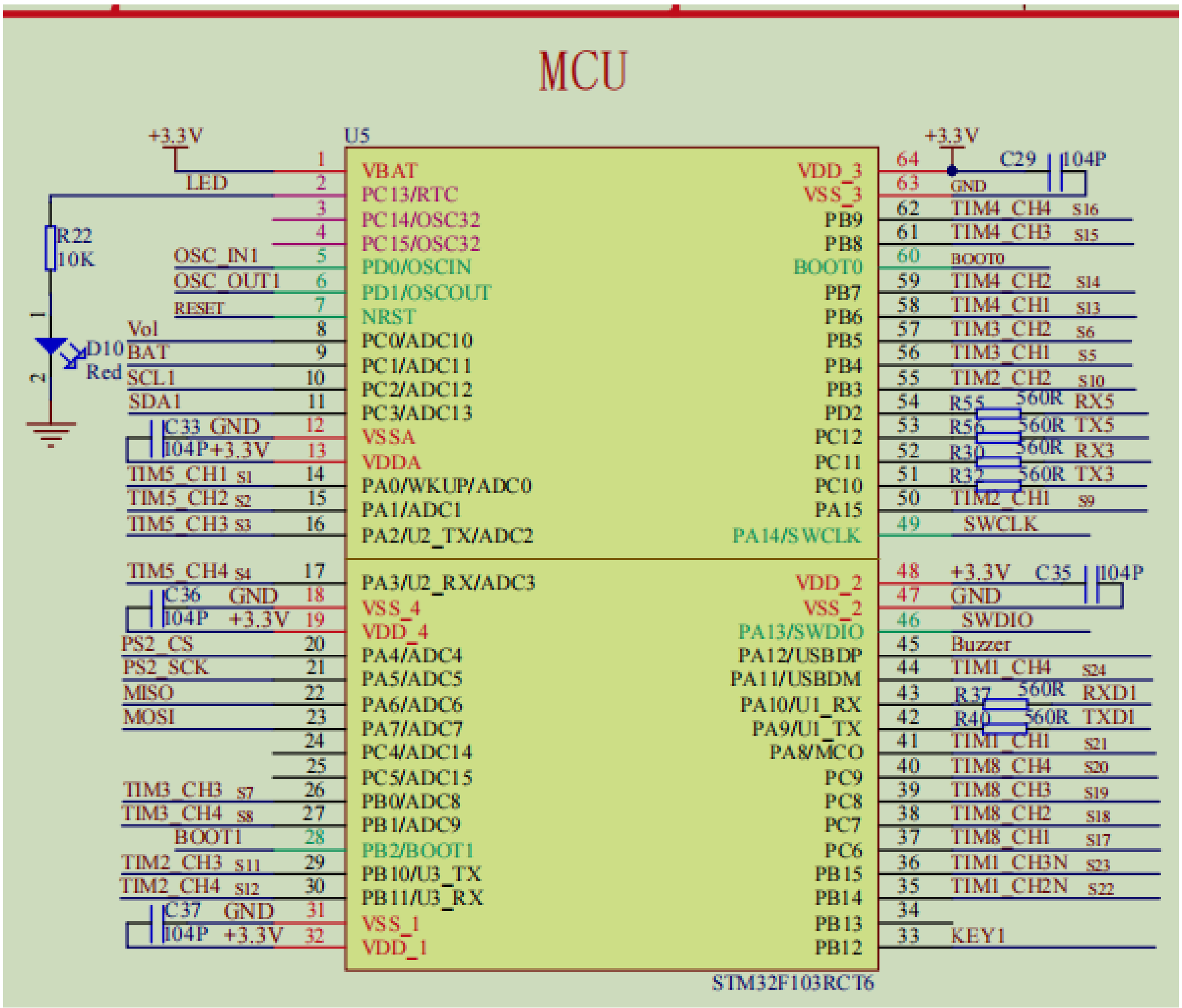


3.Experiment Purpose

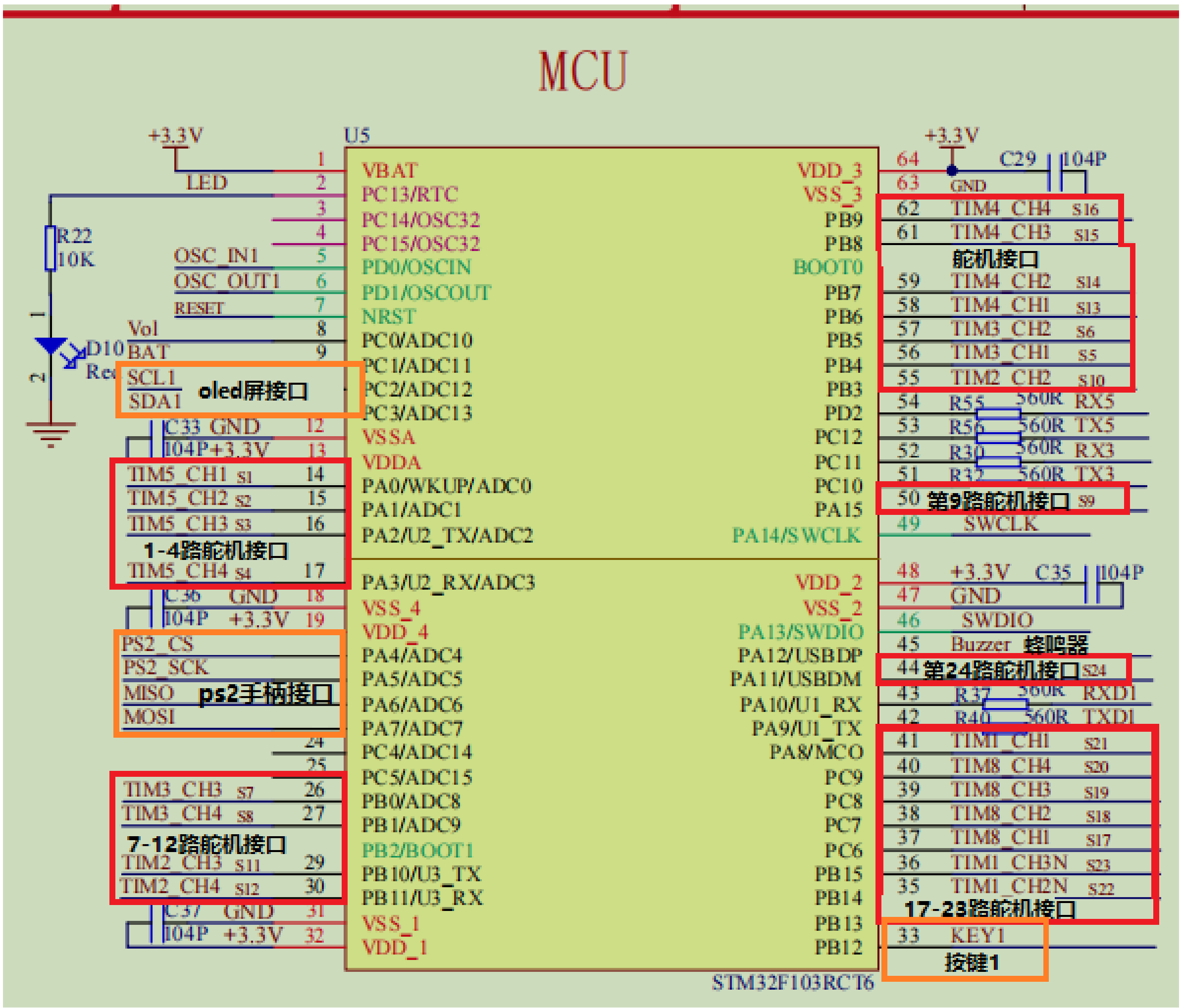
The 24 channel steering gear control board controls the movement of the 2DOF servo PTZ, realizing the process of mobile phone APP Bluetooth control and PS2 handle control.

4.Experimental Steps

4.1Schematic diagram



From the schematic diagram, we can clearly understand how each pin of stm32 controls the 24 channel steering gear control board.



- Serial port 5 communicates with Bluetooth
- Serial port 3 is reserved for communication with raspberry pie and Jetson series.

4.2 Description of pPS2 handle control

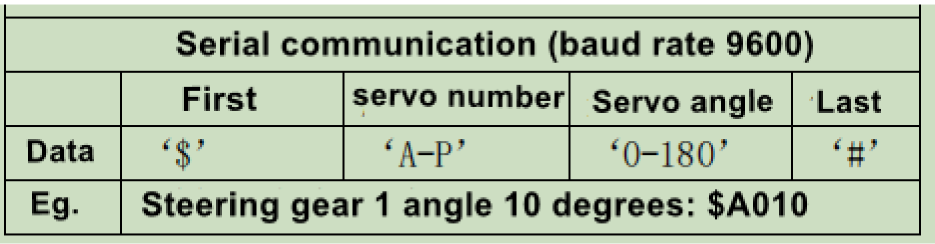
Initialize the STM32 pin according to the schematic diagram
Handle interface initialization:
Input DI ->PA6
Output DO ->PA7 CS ->PA4 CLK ->PA5
Then according to the working principle of PS2 handle, write the relevant driver.

Handle control principle
The handle control can only control 8-channel steering gear at one time, so this experiment divides the 24 channel steering gear into three groups, each group has 8-channel steering gear, and uses the key-1 on the board to switch the number of steering gear groups (value of key 1: 0 ->1 ->2 ->0), realizing the function of the handle control of 24-channel steering gear.
Handle control phenomenon
Each two buttons on the handle (one for increasing angle and one for decreasing angle) correspond to one way steering gear control.
The keys correspond to the servo as follows:

Increase angle	Decrease angle	Key value	Servo
SELECT	START	0	S1 servo
L3	R3		S2 servo
up	DOWN		S3 servo
RIGHT	LEFT		S4 servo
L2	R2		S5 servo
L1	R1		S6 servo
GREEN	BLUE		S7 servo
RED	PINK		S8 servo
SELECT	START	1	S9 servo
L3	R3		S10 servo
up	DOWN		S11 servo
RIGHT	LEFT		S12 servo
L2	R2		S13 servo
L1	R1		S14 servo
GREEN	BLUE		S15 servo
RED	PINK		S16 servo
SELECT	START	2	S17 servo
L3	R3		S18 servo
up	DOWN		S19 servo
RIGHT	LEFT		S20 servo
L2	R2		S21 servo
L1	R1		S22 servo
GREEN	BLUE		S23 servo
RED	PINK		S24 servo

4.3 Description of mobile APP control

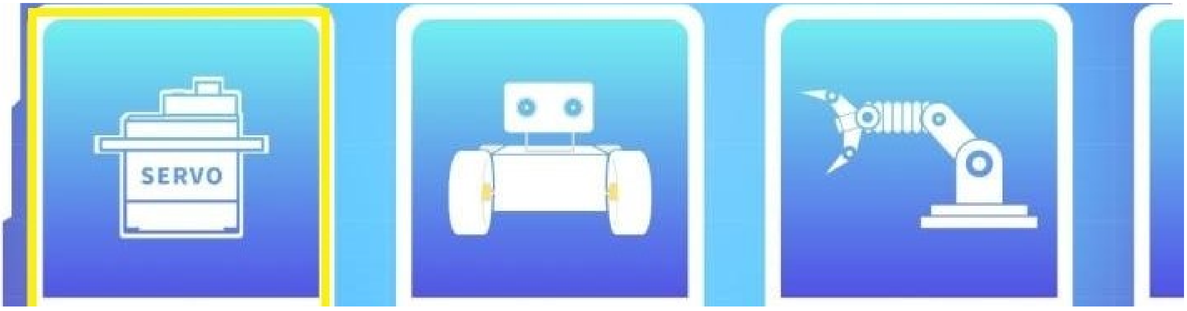
The mobile app communicates with the Bluetooth module on the 24 channel servo control board through Bluetooth, thus indirectly controlling the movement of the 2DOF steering gear PTZ.
The communication protocol is as follows:



Serial communication (baud rate 9600)				
	First	servo number	Servo angle	Last
Data	‘$’	‘A-P’	‘0-180’	‘#’
Eg.	Steering gear 1 angle 10 degrees: $A010			

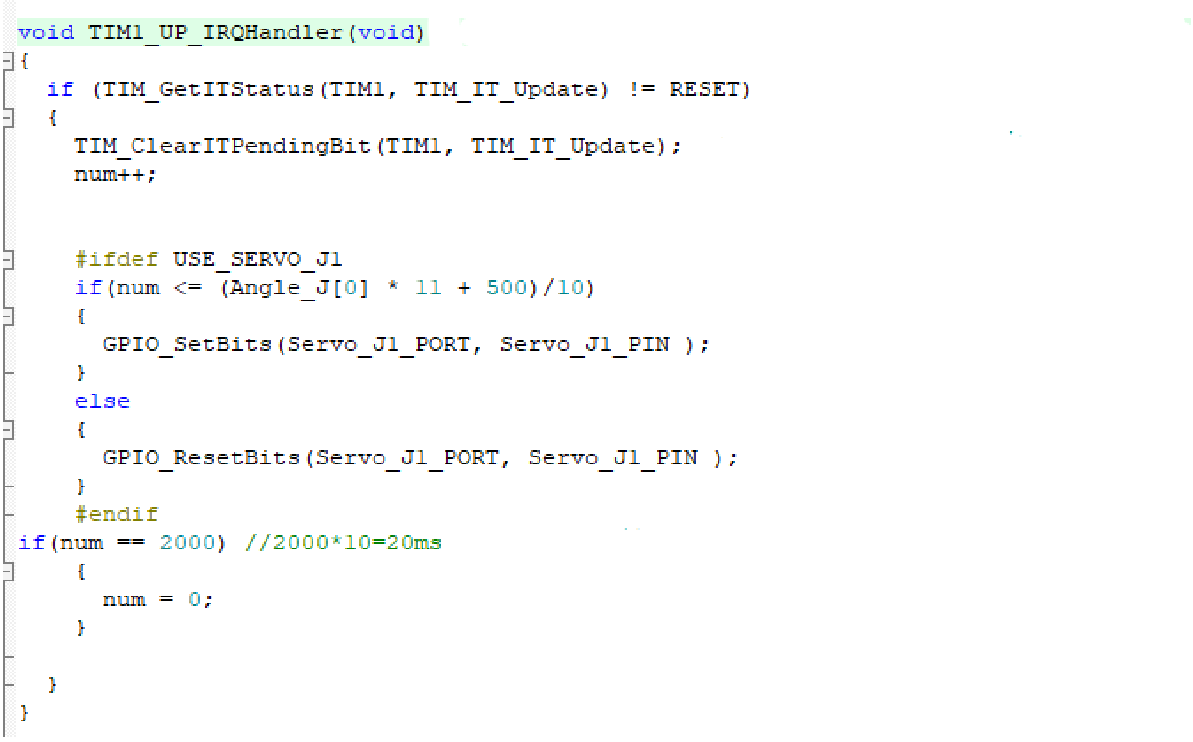
Install the APP provided by us and connect Bluetooth on the APP interface to achieve the goal of mobile app controlling the 2DOF PTZ.
The QR code for downloading the app, as shown below.

After connecting Bluetooth on the APP interface, enter the servo control page



4.4 Servo motion

This tutorial uses a timer to simulate PWM to realize the angle control of the steering gear. The main code is as follows: (Take a PWM servo as an example)



```
void TIM1_UP_IRQHandler(void)
{
  if (TIM_GetITStatus(TIM1, TIM_IT_Update) != RESET)
  {
    TIM_ClearITPendingBit(TIM1, TIM_IT_Update);
    num++;


    #ifdef USE_SERVO_J1
    if(num <= (Angle_J[0] * 11 + 500)/10)
    {
      GPIO_SetBits(Servo_J1_PORT, Servo_J1_PIN );
    }
    else
    {
      GPIO_ResetBits(Servo_J1_PORT, Servo_J1_PIN );
    }
    #endif
if(num == 2000) //2000*10=20ms
    {
      num = 0;
    }

  }
}
```

5.Source code analysis

```
int main(void)
{
    u8 Dnum;
    u8 Doup;
    //Steering gear angle initialization
    for(Doup = 0;Doup <GROUP_NUM;Doup++)
    {
        for(Dnum = 0;Dnum <DUOJI_NUM;Dnum++)
        {
            Angle_J[Doup][Dnum] = 90;
        }

    }

    bsp_init();//Perform hardware initialization again
    while (1)
    {
        dectect_beep();//Detection voltage
        void_jutce(); //Detect the total current of servo
        user_douji();   //Handle control
        if(Key1_State(1)==1)//Each click will only take effect once
        {
            grop++;
            if(grop>2)
            {
                grop = 0;
            }

        }
    }
}
```

- Angle_ J: It is a two-dimensional array that controls the angle of the steering gear
- GROUP_ NUM: it defines how to divide the 24 way steering gear into several groups
- DUOJI_ NUM: is the defined number of each group of steering gear
  -The timer controls the angle of the steering gear according to Angle_ J [0] [0] is a numerical value that controls the angle of the first steering gear, and the same is true for other steering gears.
- bsp_ Init: This function is used to initialize the IO port according to the schematic diagram. See the source code for details.
- dectect_ Beep: This function is used to detect whether the battery has a power supply of more than 6.5V. If the power supply is lower than 6.5V, an alarm will be given automatically (buzzer will sound). The alarm can only be removed by replacing a power supply of more than 6.5V.
- void_ Jutce: This function detects whether the total current of the steering gear is greater than 16A. If it is, the steering gear is overloaded, and the buzzer will ring immediately.
- user_ Douji: This function is a control to implement the handle. See the source code for details.
- The mobile APP is controlled by serial port interrupt. The source code of interrupt processing is in BSP_ usart.

  C Content of serial port 5In All Header The h file also defines some peripheral switches. If you do not connect some peripheral devices, you can turn off these peripheral functions, reduce power consumption and memory, and improve speed. Especially if you do not connect the PS2 handle, you must turn off the functions of the handle, otherwise it will affect the judgment.

6.Experiment summary

Through experiments and source code analysis, it can be clearly understood that the working principle of the steering gear is as follows:
Working principle of steering gear: the control signal enters the signal modulation chip from the channel of the receiver to obtain the DC bias voltage. There is a reference circuit inside it, which generates a reference signal with a period of 20ms and a width of 1.5ms. The DC bias voltage obtained is compared with the voltage of the potentiometer to obtain a voltage difference output. Finally, the positive and negative output of the voltage difference to the motor driver chip determines the positive and negative rotation of the motor. When the motor speed is fixed, this wheel drives the potentiometer to rotate through cascade deceleration, so that the voltage difference is 0 and the motor stops rotating.
Steering gear control: generally, a time base pulse of about 20ms is required, and the high-level part of the pulse is generally the angle control pulse within the range of 0.5ms-2.5ms. The steering gear used in this experiment is a 180 degree servo, and the control relationship is as follows:
0.5ms ---------------- 0°
1.0ms ----------------- 45°
1.5ms ----------------- 90°

2.0ms ----------------- 135°
2.5ms ----------------- 180°

7.Expand the characteristic functions of parts and boards

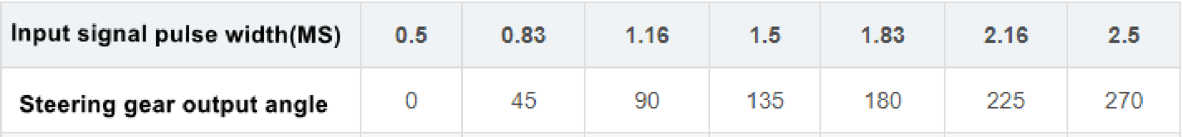
7.1 Expansion part - control 270° servo

- If you need to control 270° servo, BSP of source code_ Overwrite the interrupt function in timer. c file

  For example, if the first servo control is a 27° steering gear_ UP_ Rewrite the control formula of the first servo in IRQHandler() function.

```
//Turn this formula (180°)
num <= (Angle_J[0][0] * 11 + 500)/times
//Change to 270°
num <= (Angle_J[0][0] * 8 + 500)/times
```

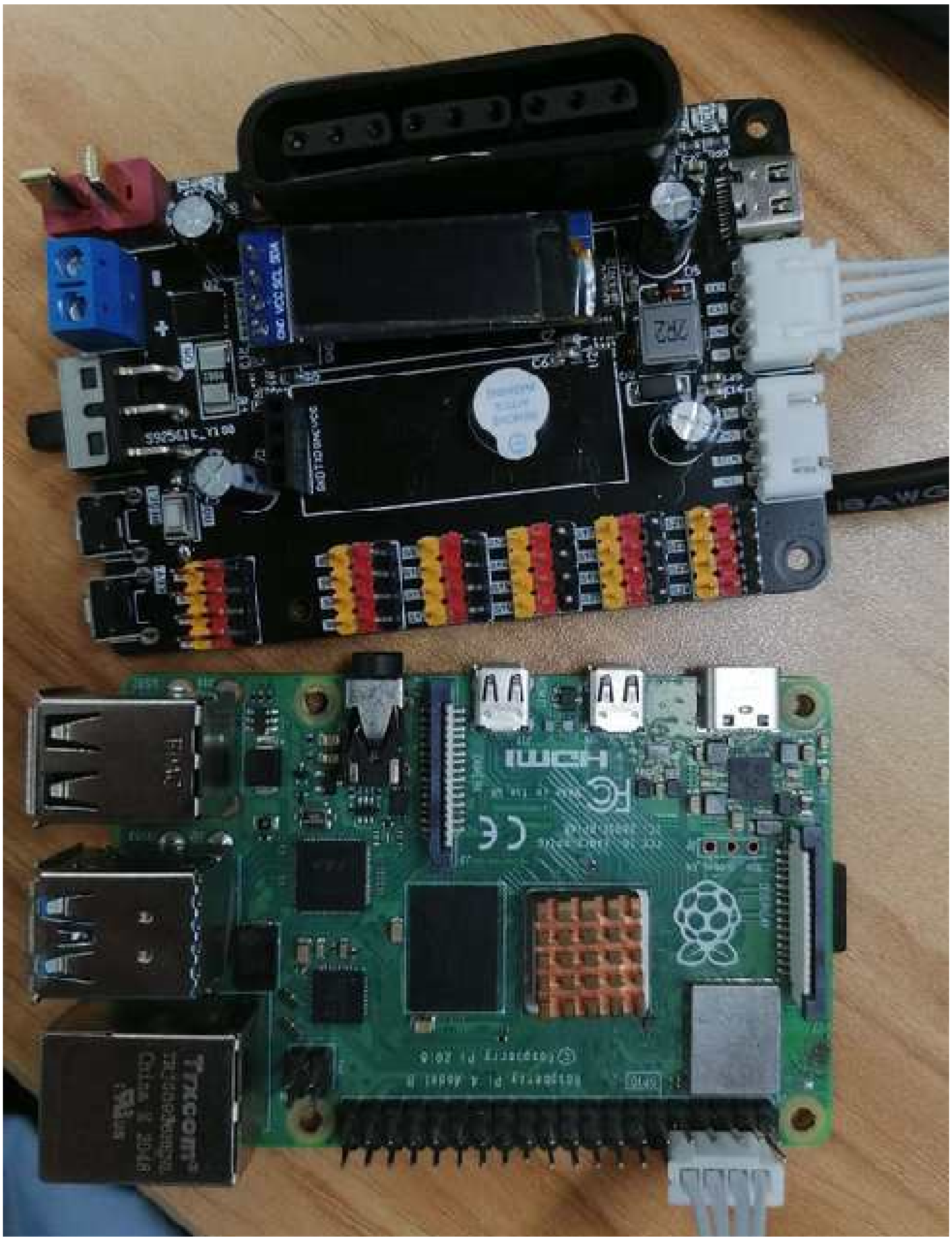
The range of control angle will change from 0-180° to 0-270° .
The corresponding relationship, as shown below.



Input signal pulse width(MS)	0.5	0.83	1.16	1.5	1.83	2.16	2.5
Steering gear output angle	0	45	90	135	180	225	270

2.Features

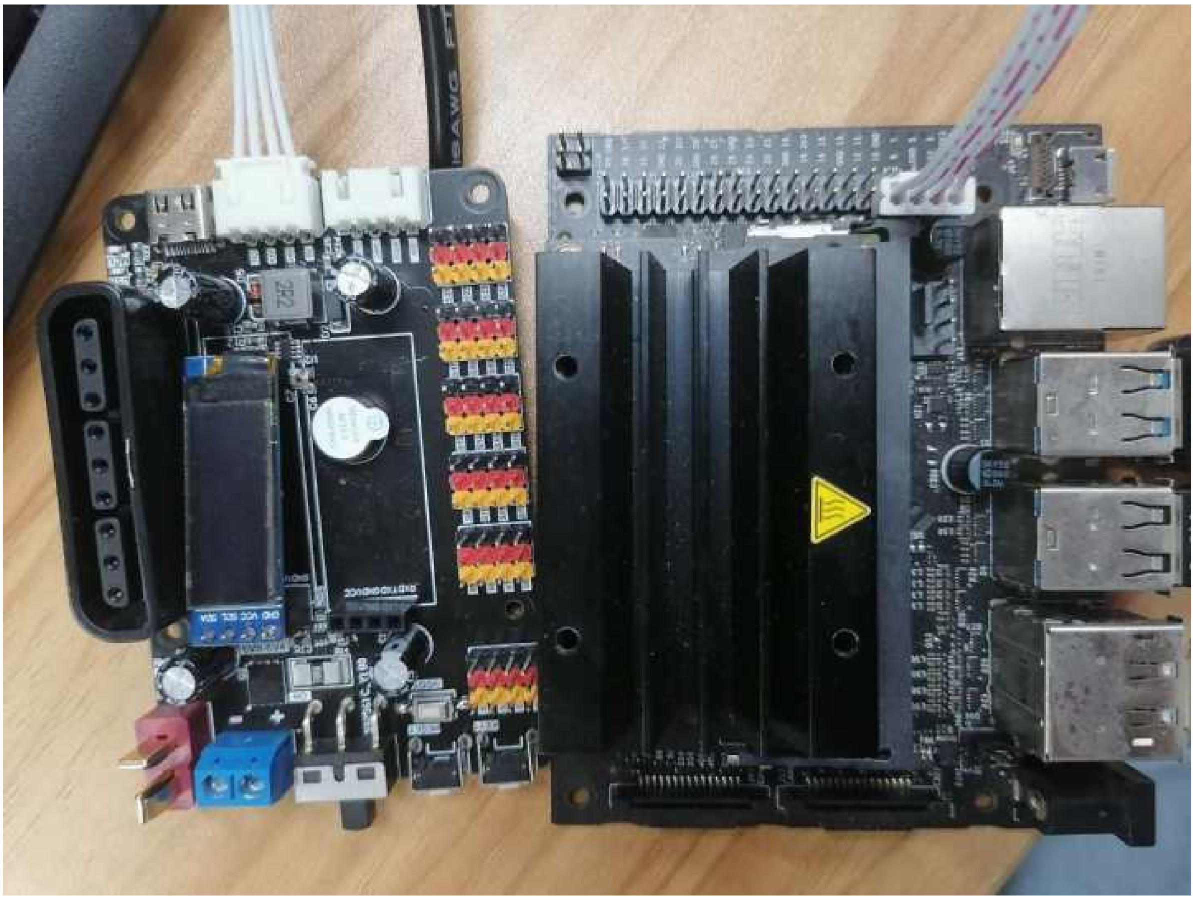
- The board has the function of detecting the current when the steering gear rotates, which has the function of protecting the board and alarming the current overload of the steering gear.
- The serial port of the board can be directly connected with the serial port on the raspberry pie and the Jetson nano pin, which can largely avoid the cumbersome wiring and greatly improve the wiring efficiency.
  As shown below.



Wiring diagram between steering gear control panel and Raspbeery Pi

If you encounter a program error, you can refer to the next tutorial document.

```
#If the module cannot be found, you can use the following command to install it
sudo pip install serial
sudo pip install pyserial
```



Wiring diagram of servo control board and Jetson NANO.

- In the folder of program source code, there are serial communication programs of Raspberry Pi, jetson nano and servo control board.

Procedural phenomenon

Raspberry Pi/Jetson nano sends the letter 'H' continuously through the serial port, and the servo control board will immediately return the received information, that is, what is sent back to what is sent back.