

Serial port control servo

1. Learning objectives

In this course, we mainly learn to use STM32F103RCT6 and 16-way servo drive modules to implement serial port control servos.

2. Prepare before class

In this example, the 16-way servo drive module adopts serial port communication, and the TXD and RXD of the module are connected to the PA3 and PA2 pins of the STM32F103RCT6 board, respectively. VCC and GND are connected to the 3.3V and GND of the STM32F103RCT6, respectively.

3. Programming

Initialize serial ports, interrupts, delays, etc.

```
//初始化函数
NVIC_PriorityGroupConfig(NVIC_PriorityGroup_2); //设置系统中断优先级分组2
delay_init(); //初始化延时函数
uart_init(115200); //初始化串口1波特率为115200, 用于支持USMART
usart2_init(usart2_baud); //初始化串口2, 用于与模块通讯
mem_init();
```

The servo control function, according to the protocol, 0x24 and 0x23 are the header and tail of the packet, respectively.

```
void UART_Servo(unsigned char servonum, unsigned char angle)
{
    servonum = 64 + servonum;
    date1 = angle/100 + 48;
    date2 = (angle%100)/10 + 48;
    date3 = angle%10 + 48;
    MYUSART_SendData(0x24); //发送包头
    MYUSART_SendData(servonum); //发送舵机编号
    MYUSART_SendData(date1); //发送角度
    MYUSART_SendData(date2); //发送角度
    MYUSART_SendData(date3); //发送角度
    MYUSART_SendData(0x23); //发送包尾
    delay_ms(100);
}
```

Control the servo S1 with a for cycle, select 5 degrees each time from 0 to 180, and finally return to 0 degrees.

```
for(i = 0; i < 180; i += 5) {
    UART_Servo(1, i);
}
UART_Servo(1, 0);
```

4. Experimental phenomenon

After the program is downloaded, it runs, and the servo goes from 0 degrees to 180 degrees, and then back to 0 degrees.