IIC control servo

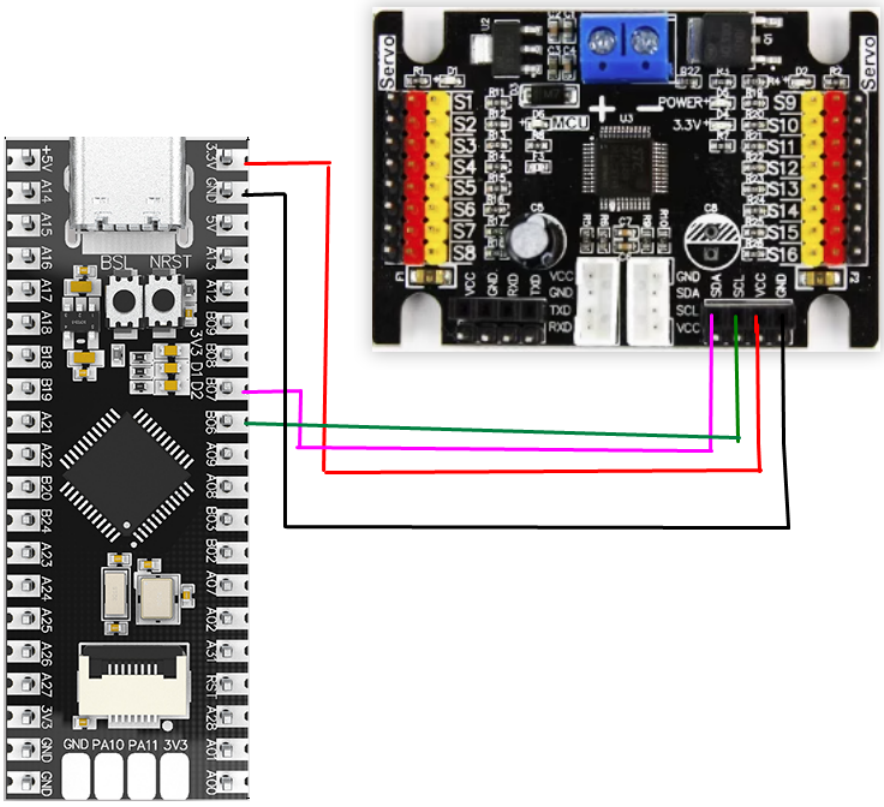
1. Learning objectives

Control the servo through the IIC of the 16-channel servo driver board

2. Hardware connection

The servo used in this case is a 180-degree servo, which is connected to the S1 pin of the 16-channel servo driver board

Connecting MSPM0G3507 with the 16-channel servo driver board



All modules should be connected together with a common ground line

3. Main program description

- bsp_i2c_gpio.h

```
#define I2C_ADDR                    0x5A

#define macI2C_WR   0       /* 写控制bit   Write control bit */
#define macI2C_RD   1       /* 读控制bit   Read control bit */

/* 定义I2C总线连接的GPIO端口，用户只需要修改下面4行代码即可任意改变SCL和SDA的引脚 */
/* Define the GPIO port connected to the I2C bus. Users only need to modify the
following 4 lines of code to change the SCL and SDA pins at will */
#define macGPIO_PORT_I2C    GPIOB           /* GPIO端口 GPIO Ports*/
```

```
#define macRCC_I2C_PORT     RCC_APB2Periph_GPIOB            /* GPIO端口时钟 GPIO
port clock*/
#define macI2C_SCL_PIN      SCL_PIN_6_PIN           /* 连接到SCL时钟线的GPIO GPIO
connected to SCL clock line*/
#define macI2C_SDA_PIN      SDA_PIN_7_PIN           /* 连接到SDA数据线的GPIO GPIO
connected to SDA data line*/


#define macI2C_SCL_1()  DL_GPIO_setPins(macGPIO_PORT_I2C, macI2C_SCL_PIN)
/* SCL = 1 */
#define macI2C_SCL_0()  DL_GPIO_clearPins(macGPIO_PORT_I2C, macI2C_SCL_PIN)
/* SCL = 0 */

#define macI2C_SDA_1()  DL_GPIO_setPins(macGPIO_PORT_I2C, macI2C_SDA_PIN)
/* SDA = 1 */
#define macI2C_SDA_0()  DL_GPIO_clearPins(macGPIO_PORT_I2C, macI2C_SDA_PIN)
/* SDA = 0 */

#define macI2C_SDA_READ()  DL_GPIO_readPins(macGPIO_PORT_I2C, macI2C_SDA_PIN)
/* 读SDA口线状态 Read SDA line status*/
#define macI2C_SCL_READ()  DL_GPIO_readPins(macGPIO_PORT_I2C, macI2C_SCL_PIN)
/* 读SCL口线状态 Read SCL line status*/


void Start_I2c(void);
void Stop_I2c(void);
void  I2C_SendByte(unsigned char  c);
unsigned char I2C_RcvByte(void);
void Ack_I2c(uint8_t a);

uint8_t IIC_Servo(unsigned char servonum,unsigned char angle);

void delay(int z);

#endif
```

Defines the I2C port and its operation function, and declares the basic I2C operation function and servo control function.

- bsp_i2c_gpio.c

```
uint8_t IIC_Servo(unsigned char servonum,unsigned char angle)
{
    Start_I2c();                //启动总线  Start bus
    I2C_SendByte(I2C_ADDR);     //发送器件地址  Send device address
    if(ack==0)return(0);
    I2C_SendByte(servonum);    //发送数据 Send data
    if(ack==0)return(0);
    I2C_SendByte(angle);    //发送数据  Send data
    if(ack==0)return(0);

    Stop_I2c();               //结束总线  End the bus
    delay_ms(100);
    return(1);
```

```
}
```

Define i2c control servo function. Servo range (1-16), angle (0-180).

- empty.c

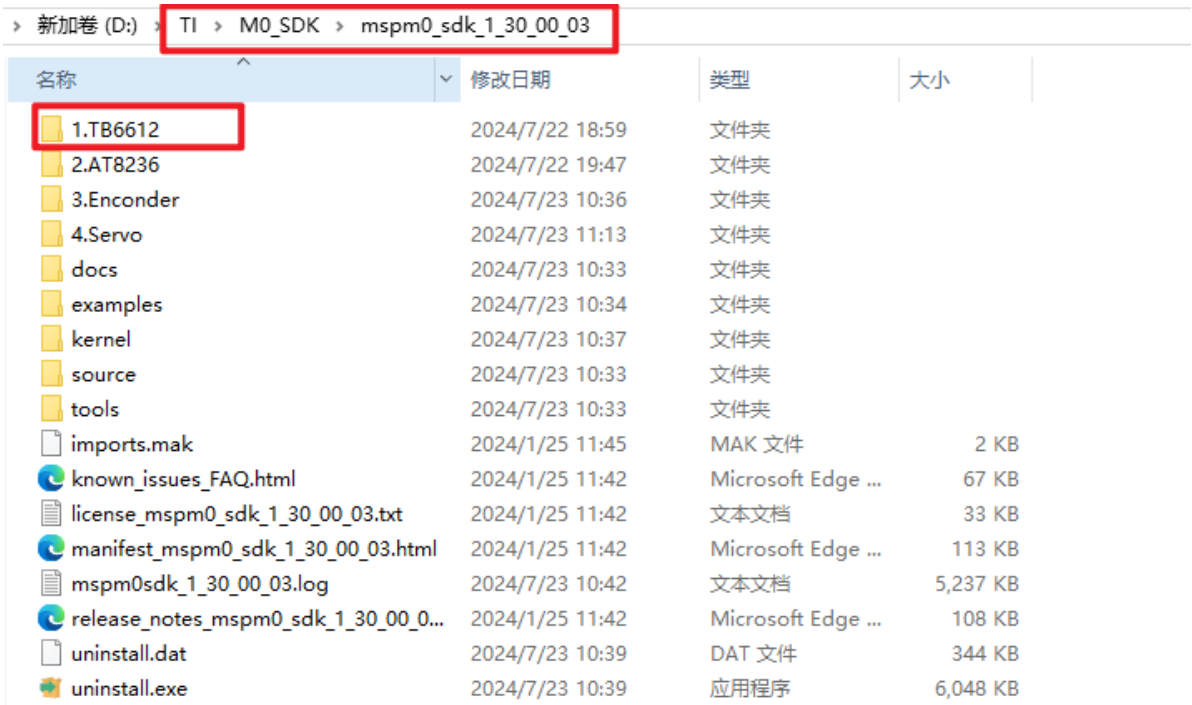
```
int main(void)
{
    int  i=0;
    SYSCFG_DL_init();

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

Control the No. 1 servo to rotate from 0 to 180 degrees, and then return to 0 degrees after reaching 180 degrees, and repeat this process.

Note: The project source code must be placed in the SDK path for compilation,

For example, the path: D:\TI\M0_SDK\mspm0_sdk_1_30_00_03\1.TB6612



4. Experimental phenomenon

Connect the servo to the S1 interface of the 16-channel servo board, burn the iic control program to MSPM0G3507, and connect the 16-channel servo driver board and MSPM0G3507 according to the wiring diagram. After power on, you can see that the No. 1 servo rotates from 0 degrees to 180 degrees, and then returns to 0 degrees, and repeats this process.