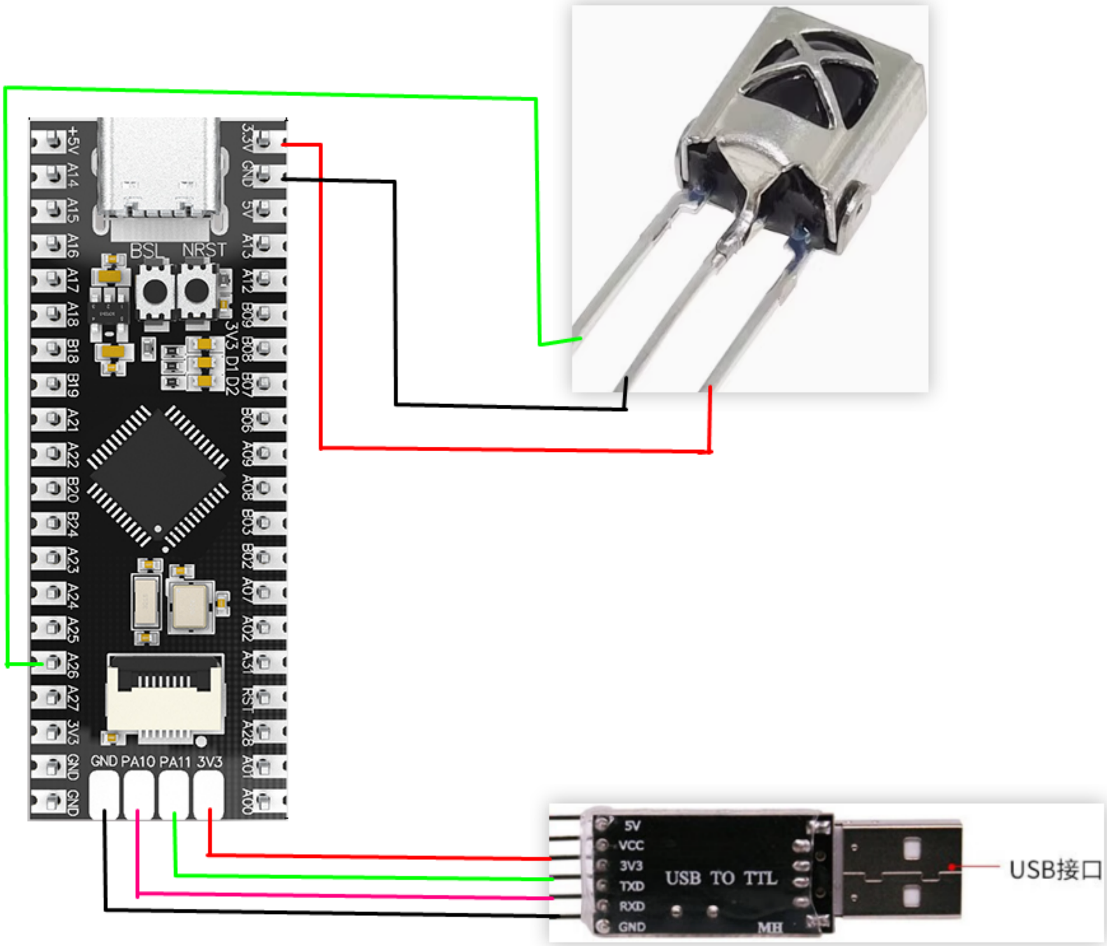
Serial port prints infrared remote control key values

1. Learning Objectives

Receive the infrared remote control key value through the 1838 infrared receiving module and print the key value through the serial port

2. Hardware Connection

Wiring between infrared receiving module and MSPM0G3507:



Note: The infrared receiving module needs to be connected to the mainboard through a breadboard; if there is no TTL module, you can also use the type-c serial port directly

3. Program Description

- usart.c

```
void USART_Init(void)
{
    // SYSCFG初始化
    // SYSCFG initialization
    SYSCFG_DL_init();
    //清除串口中断标志
    //Clear the serial port interrupt flag
    NVIC_ClearPendingIRQ(UART_0_INST_INT_IRQN);
    //使能串口中断
    //Enable serial port interrupt
    NVIC_EnableIRQ(UART_0_INST_INT_IRQN);
}

//串口的中断服务函数
//Serial port interrupt service function
void UART_0_INST_IRQHandler(void)
{
    uint8_t receivedData = 0;

    //如果产生了串口中断
    //If a serial port interrupt occurs
    switch( DL_UART_getPendingInterrupt(UART_0_INST) )
    {
        case DL_UART_IIDX_RX://如果是接收中断  If it is a receive interrupt

            // 接收发送过来的数据保存  Receive and save the data sent
            receivedData = DL_UART_Main_receiveData(UART_0_INST);

            // 检查缓冲区是否已满    Check if the buffer is full
            if (recv0_length < RE_0_BUFF_LEN_MAX - 1)
            {
                recv0_buff[recv0_length++] = receivedData;
            }
            else
            {
                recv0_length = 0;
            }

            // 标记接收标志   Mark receiving flag
            recv0_flag = 1;

            break;

        default://其他的串口中断   Other serial port interrupts
            break;
    }
}
```

- USART_Init: Initialization of the system and serial port.
- UART_0_INST_IRQHandler: Receive interrupt processing function of the serial port.

- irremote.c

```
//接收红外数据  Receive infrared data
/*
数据发送0码：0.56ms低电平+ 0.56ms的高电平   Data sent 0 code: 0.56ms low level +
0.56ms high level
```

```
数据发送1码：0.56ms低电平+ 1.68ms的高电平   Data sent 1 code: 0.56ms low level +
1.68ms high level
收到数据位0: 0.56ms低电平+ 0.56ms的高电平   Received data bit 0: 0.56ms low level +
0.56ms high level
收到数据位1: 0.56ms低电平+ 1.68ms的高电平   Received data bit 1: 0.56ms low level +
1.68ms high level
*/
void Receiving_Infrared_Data(void) {

    uint16_t Group_num = 0, Data_num = 0;                  // 定义变量，用于存储数据组
的索引和数据位的索引  Define variables to store the index of the data group and the
index of the data bit
    uint32_t time=0;                                       // 定义变量，用于存储时间测
量值                Define variables to store the time measurement value
    uint8_t Bit_data = 0;                                  // 定义变量，用于存储单个数
据位的值（0或1）    Define variables to store the value of a single data bit (0 or
1)
    uint8_t ir_value[5] = {0};                             // 定义数组，用于存储接收到
的红外数据（4个字节的数据 + 1个字节用于其他目的）   Define an array to store the
received infrared data (4 bytes of data + 1 byte for other purposes)
    uint8_t Guide_Repeat_Code = 0;                         // 定义变量，用于存储引导码
和重复码的判断结果  Define variables to store the judgment results of the boot code
and repeat code
    Guide_Repeat_Code = Guide_Repeat_Judgment();           // 调用函数判断接收到的是否
是引导码或重复码    Call the function to determine whether the received code is a
boot code or a repeat code

    if(Guide_Repeat_Code == 1) {                           // 如果判断结果不是引导码，
则打印错误信息并结束函数    If the judgment result is not a boot code, print an error
message and end the function
//          printf("err\r\n");
        return;
    }

    for(Group_num = 0; Group_num < 4; Group_num++) {       // 循环4次，每次处理一个字节
的数据      Loop 4 times, processing one byte of data each time
        for(Data_num = 0; Data_num < 8; Data_num++) {      // 循环8次，每次处理一个数据
位        Loop 8 times, processing one data bit each time
            Infrared_low(&time);                           // 调用函数获取红外低电平时
间        Call function to get infrared low level time

            if((time > 60) || (time < 20))
            {
                return;// 如果低电平时间不在0.4ms到1.2ms之间，说明数据错误，结束函数
If the low level time is not between 0.4ms and 1.2ms, it means the data is wrong
and the function ends.
            }
            time = 0;                                      // 重置time变量，为下一次测
量做准备    Reset the time variable to prepare for the next measurement
            Infrared_high(&time);                          // 调用函数获取红外高电平时
间  Call the function to obtain the infrared high level time

            if((time >= 60) && (time < 100)) {             // 如果高电平时间在1.2ms到
2ms之间，说明接收到的是数据位1   If the high level time is between 1.2ms and 2ms, it
means that the received data bit 1

                Bit_data = 1;
            }
            else if((time >= 10) && (time < 50)) {       // 如果高电平时间在0.2ms到1ms之间，说明接收到的是数据位0    If the high level time is between 0.2ms and 1ms, it means that the received data bit is 0
                Bit_data = 0;
            }

            ir_value[Group_num] <<= 1;                   // 将接收到的数据位左移1位，为下一个数据位腾出位置        Shift the received data bit left by 1 bit to make room for the next data bit
            ir_value[Group_num] |= Bit_data;             // 将接收到的数据位写入数组  Write the received data bit into the array
            time=0;                                      // 重置time变量，为下一次测量做准备  Reset the time variable to prepare for the next measurement
        }
    }
    Infrared_Data_True(ir_value);                        // 调用函数判断接收到的数据是否正确，并保存正确数据     Call the function to determine whether the received data is correct and save the correct data
}

// 外部中断3的中断服务函数  Interrupt service function of external interrupt 3
void GROUP1_IRQHandler(void)
{
        //读取Group1的中断寄存器并清除中断标志位     Read the interrupt register of Group 1 and clear the interrupt flag
    switch( DL_Interrupt_getPendingGroup(DL_INTERRUPT_GROUP_1) )
    {
        //检查是否是端口中断    Check whether the port is interrupted
        case Signal_INT_IIDX:

            if( GET_OUT == 0 )// 如果是低电平    If it is low level
                    Receiving_Infrared_Data(); //接收一次红外数据    Receive infrared data once
        break;
    }
}
```

- GROUP1_IRQHandler: External interrupt set for the output pin of the infrared receiving tube. When an infrared signal is received, the interrupt is triggered, and then the received level is judged.
- Receiving_Infrared_Data: Judge the received level to see if it complies with the NEC protocol. If it does, the data is received and verified.

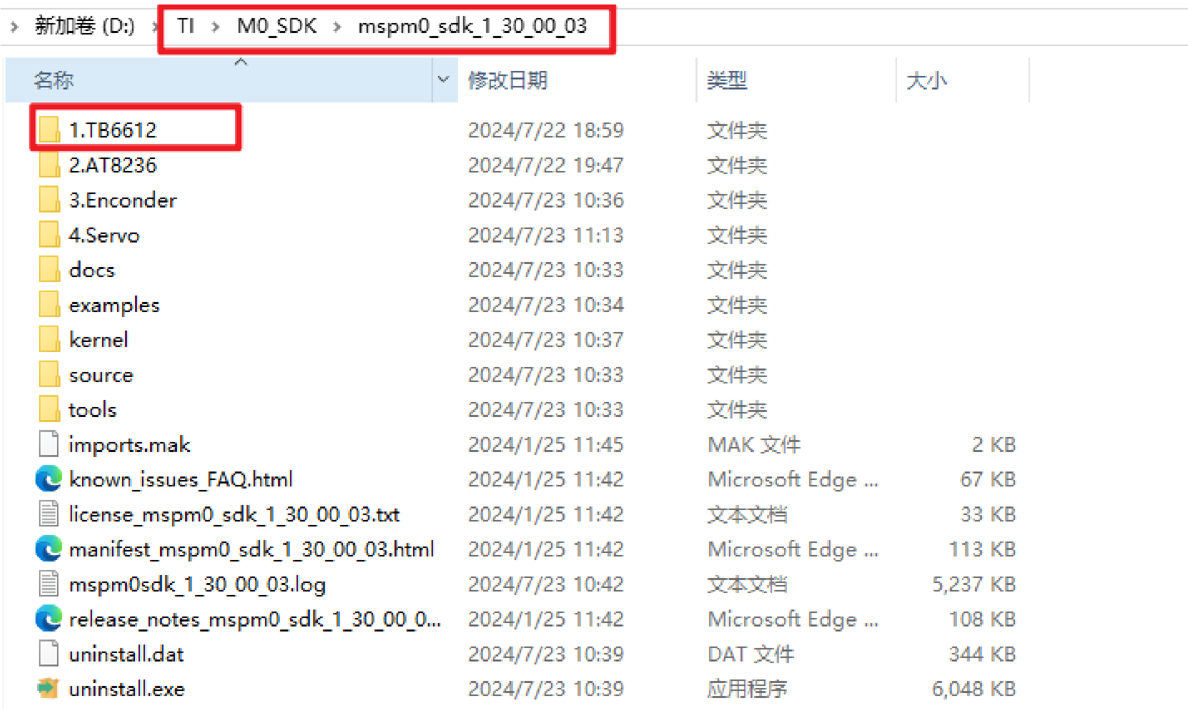
- empty.c

```
int main(void)
{
    int sign;

    USART_Init();
    NVIC_EnableIRQ(Signal_INT_IRQN);//使能外部中断  Enable external interrupt
    printf("Hello,World!\r\n");
    while (1)
    {

    }
}
```

Initialize the serial port and interrupt, and print Hello, World! on the serial port.

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



› 新加卷 (D:) › TI › M0_SDK › mspm0_sdk_1_30_00_03

名称	修改日期	类型	大小
1.TB6612	2024/7/22 18:59	文件夹	
2.AT8236	2024/7/22 19:47	文件夹	
3.Enconder	2024/7/23 10:36	文件夹	
4.Servo	2024/7/23 11:13	文件夹	
docs	2024/7/23 10:33	文件夹	
examples	2024/7/23 10:34	文件夹	
kernel	2024/7/23 10:37	文件夹	
source	2024/7/23 10:33	文件夹	
tools	2024/7/23 10:33	文件夹	
imports.mak	2024/1/25 11:45	MAK 文件	2 KB
known_issues_FAQ.html	2024/1/25 11:42	Microsoft Edge ...	67 KB
license_mspm0_sdk_1_30_00_03.txt	2024/1/25 11:42	文本文档	33 KB
manifest_mspm0_sdk_1_30_00_03.html	2024/1/25 11:42	Microsoft Edge ...	113 KB
mspm0sdk_1_30_00_03.log	2024/7/23 10:42	文本文档	5,237 KB
release_notes_mspm0_sdk_1_30_00_0...	2024/1/25 11:42	Microsoft Edge ...	108 KB
uninstall.dat	2024/7/23 10:39	DAT 文件	344 KB
uninstall.exe	2024/7/23 10:39	应用程序	6,048 KB

4. Experimental Phenomena

Burn the infrared remote control program to MSPM0G3507 and connect according to the wiring diagram. After the connection is completed, close other serial port programs on the computer, open the serial port assistant, select the serial port number, and set the baud rate to 9600. Press the reset button of the board, and the string "Hello, World!" will be printed. Then press the infrared remote control button, and the serial port will print the remote control key value.