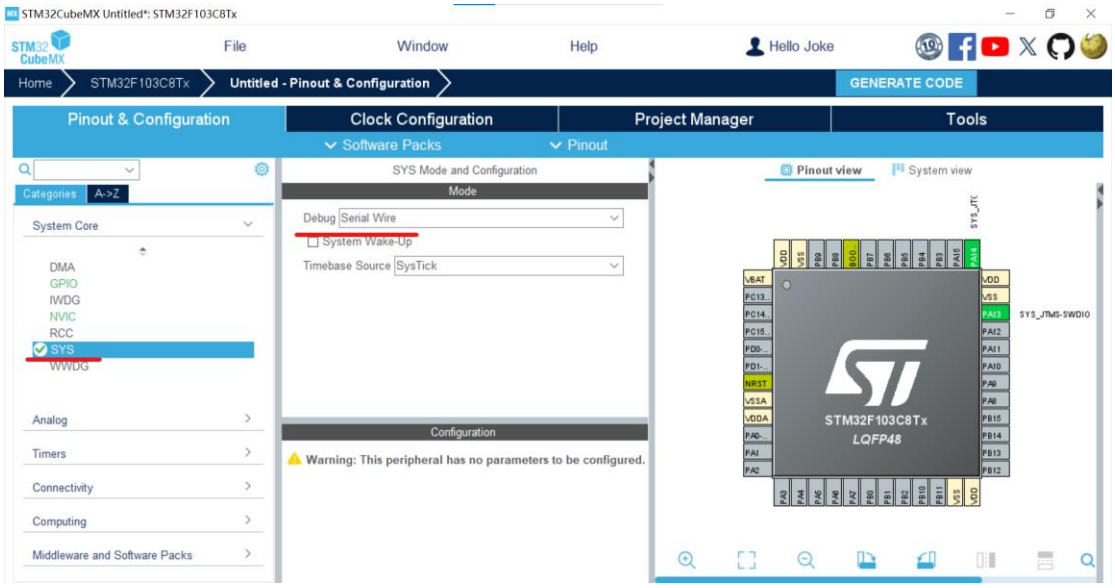
任务要求：使用 EXTI 外部中断口，实现打印。
需要实现：
1 使用 Fputc 重定向
2 使用 PA5 外部中断
3 使用串口进行中断后的打印。当有中断产生时则发送消息“clicked\r\n

使用到的外设：GPIO、USART 串口、EXTI 中断

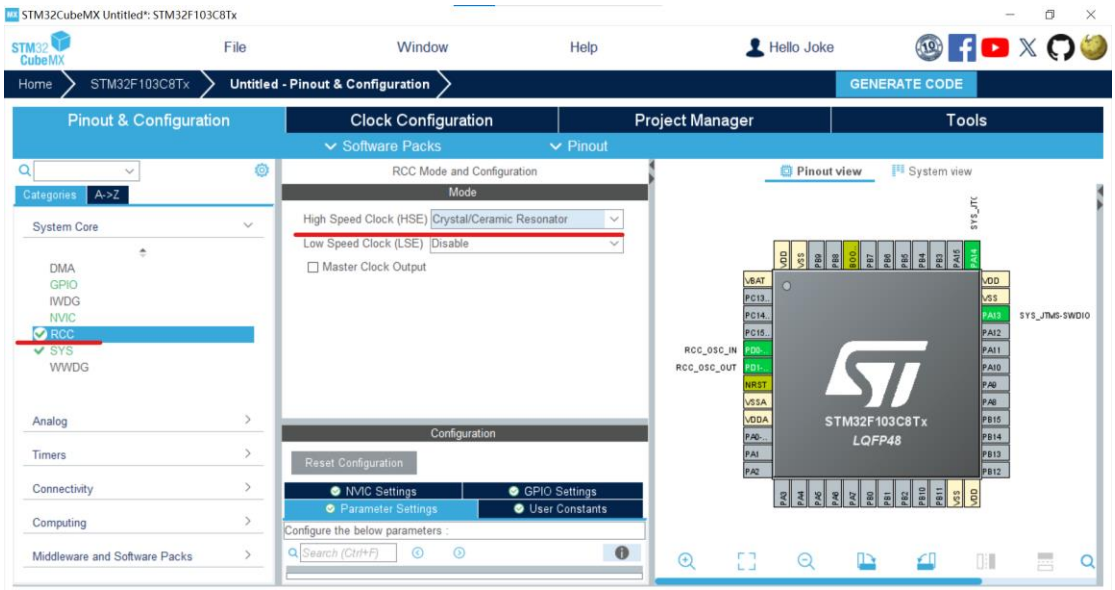
使用上升沿触发，内部下拉电阻，给于高电平就触发中断，发送字符串回串口助手软件

一、STM32CubeMX 创建工程步骤

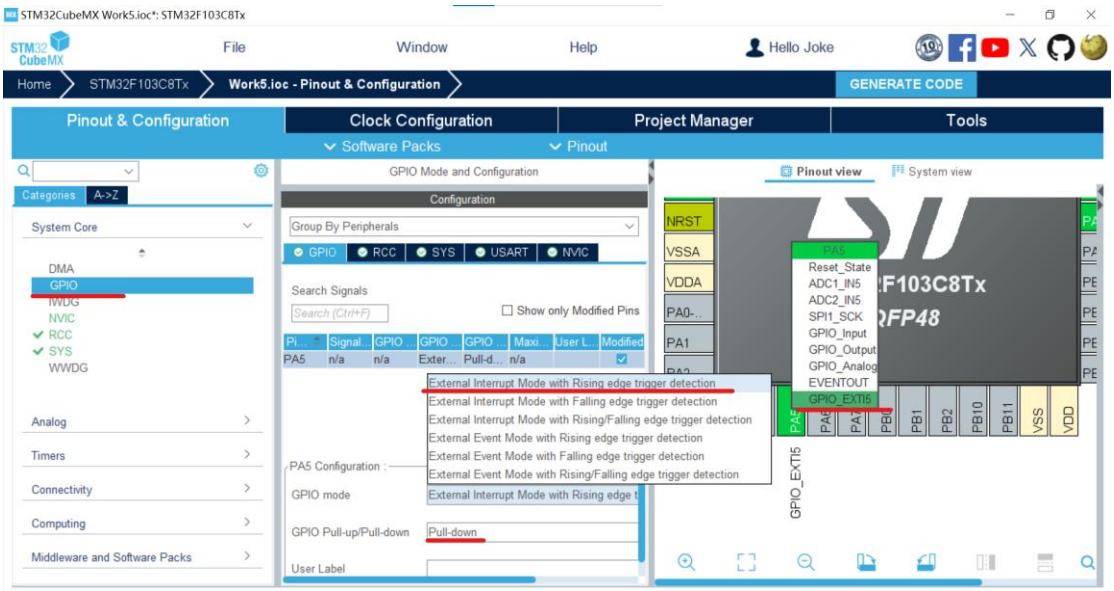
1、开启 SYS 系统时钟

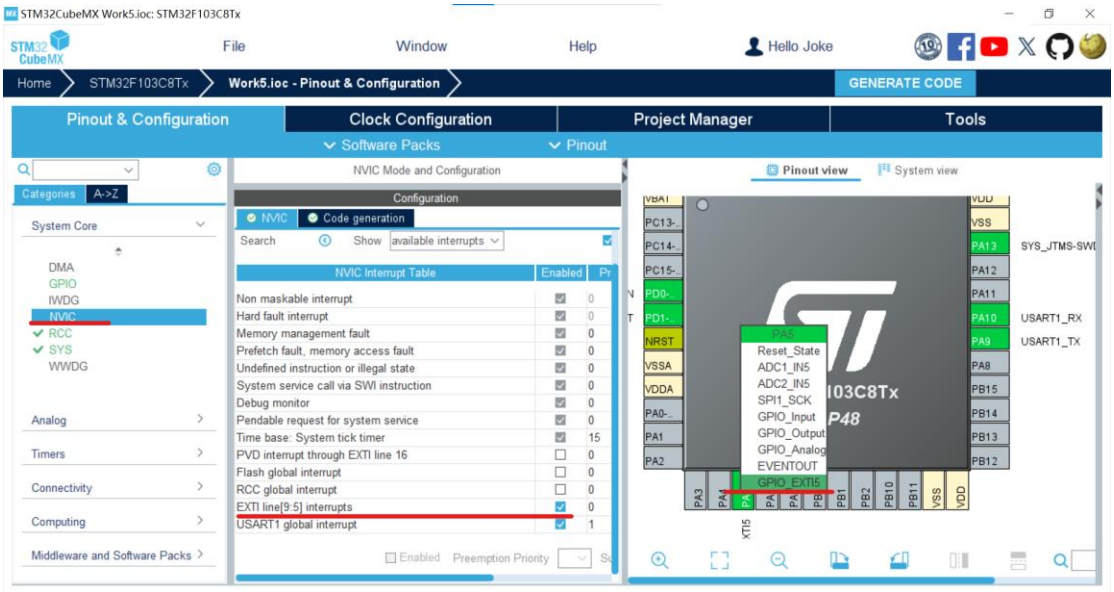


2、开启 RCC 晶振

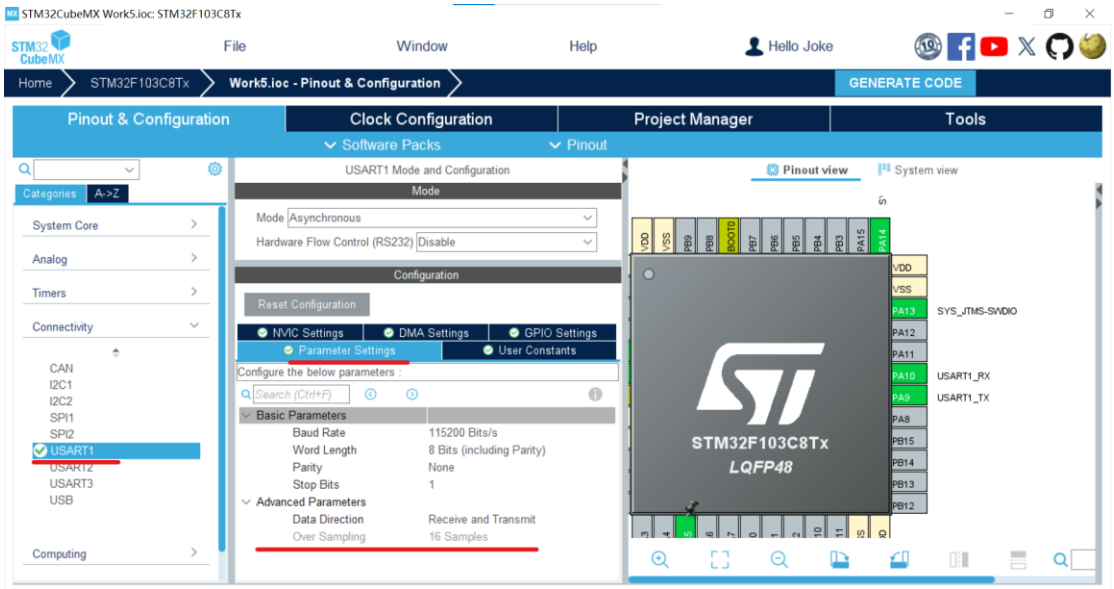


3、开启 PA5 中断，配置为上升沿触发，内部下拉电平，勾选中断使能线

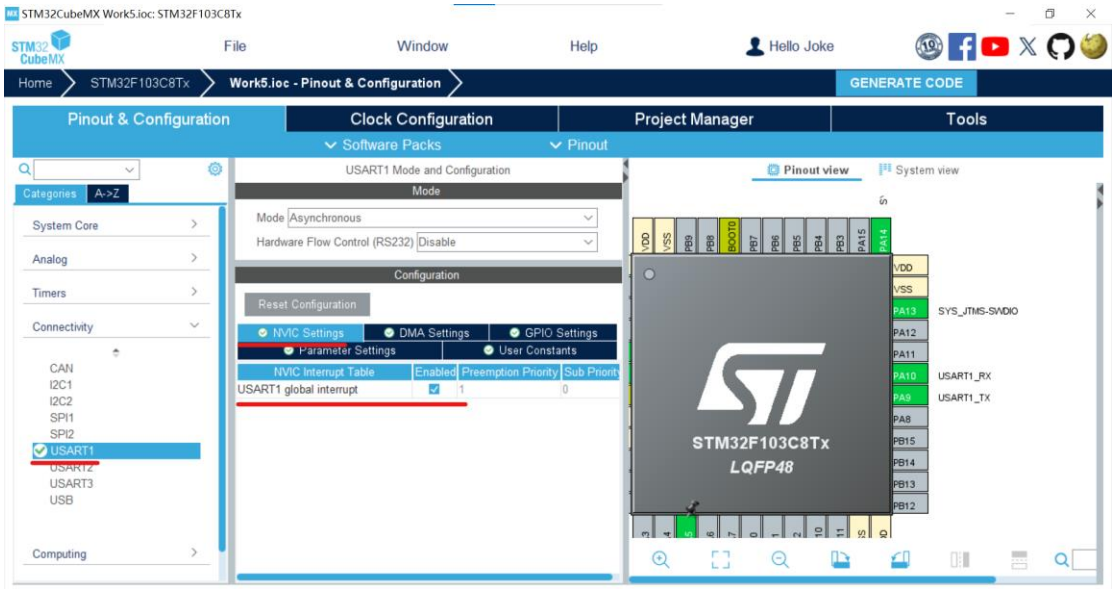




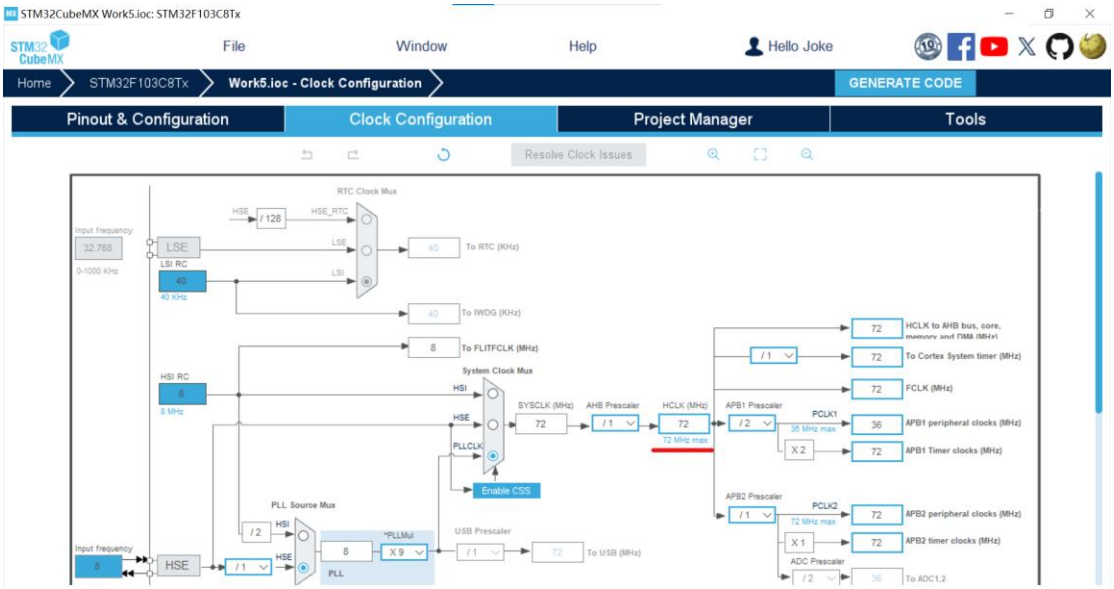
4、选择 USART1 作为串口通信，波特率设置为 115200， 8 位字节数据位，无校验位，1 位停止位，选择发送和接收模式



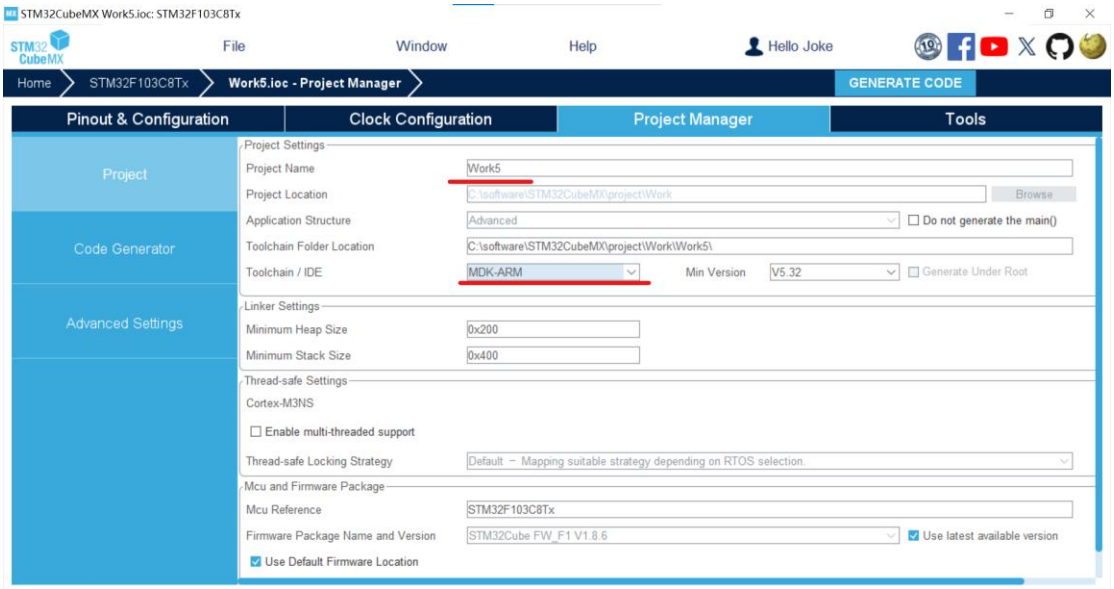
5、开启串口中断



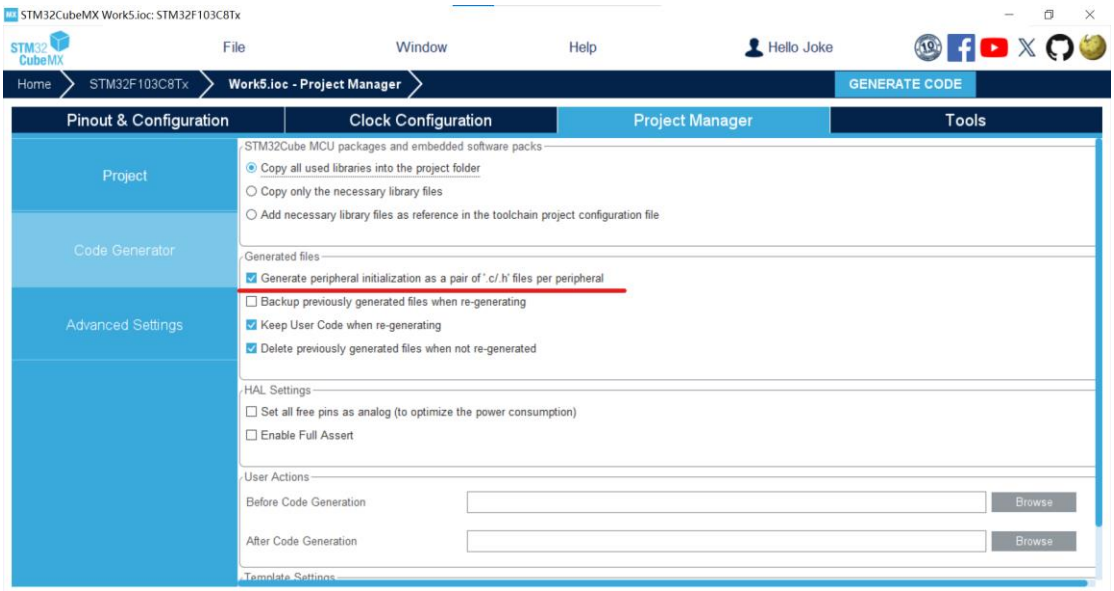
6、时钟树选择 72MHZ



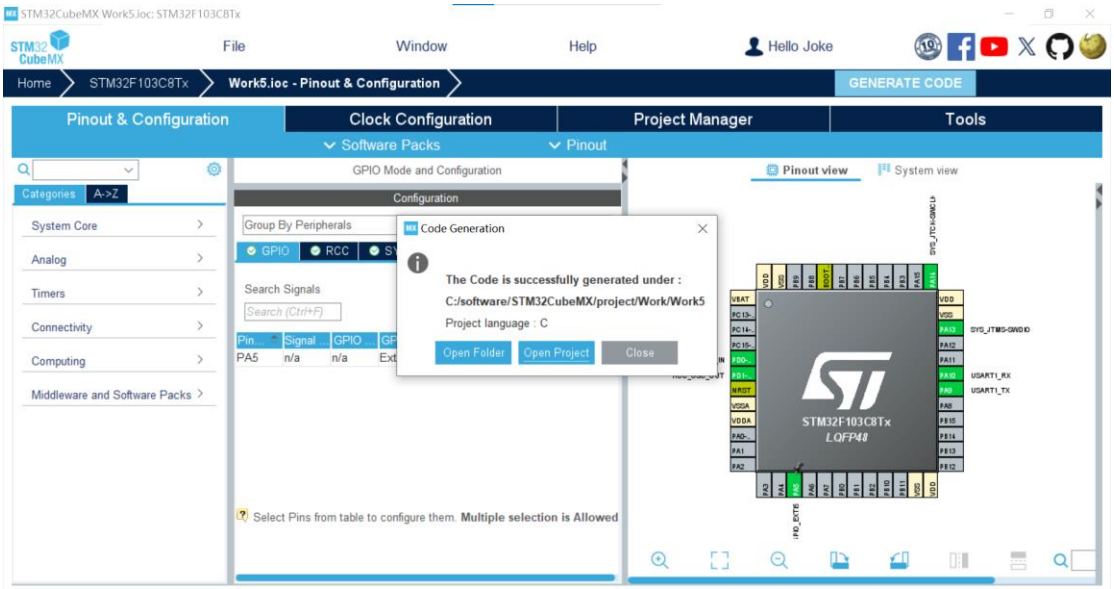
7、创建文件名，并存在对应的盘中，选择 MDK-ARM



8、勾选 Generated files 的第一个勾，将.c 和.h 文件分开存放

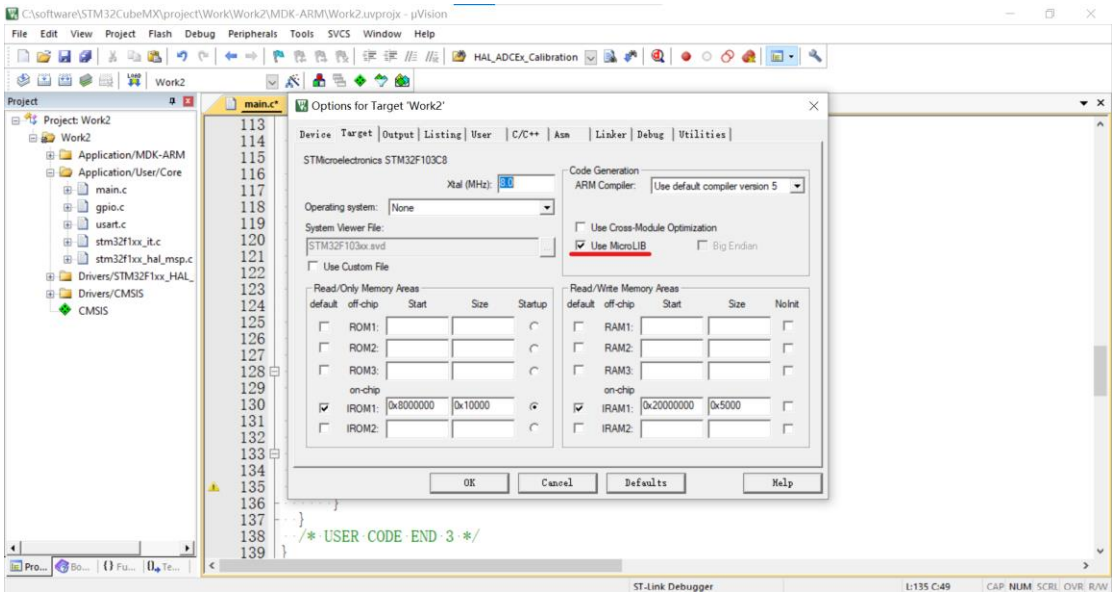


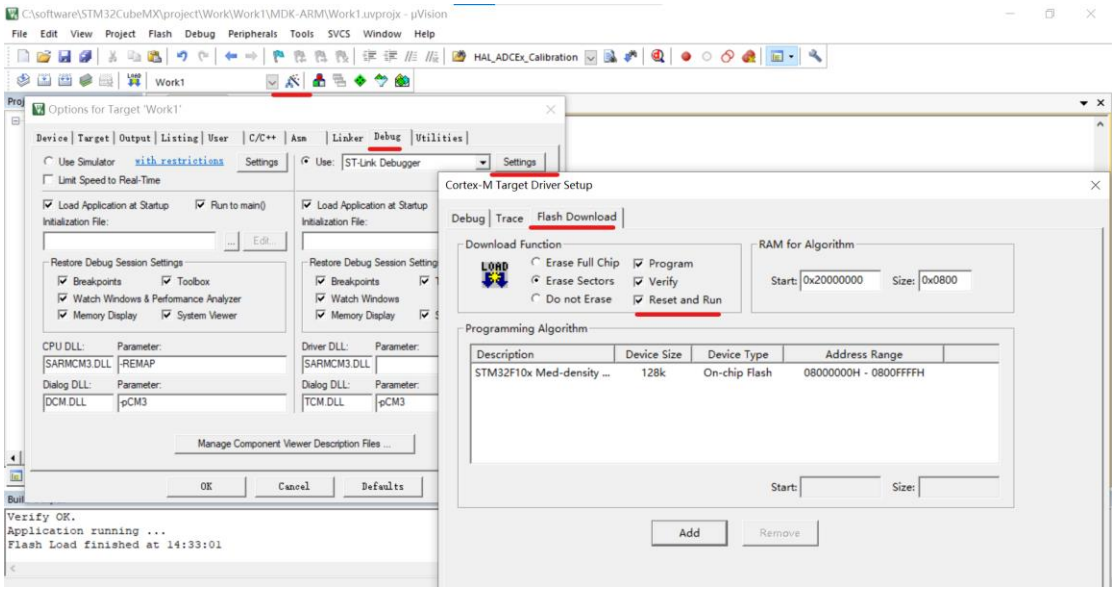
9、生成文件 GENERATE CODE，点击 CLOSE，并打开所生成的路径的 MDK 文件



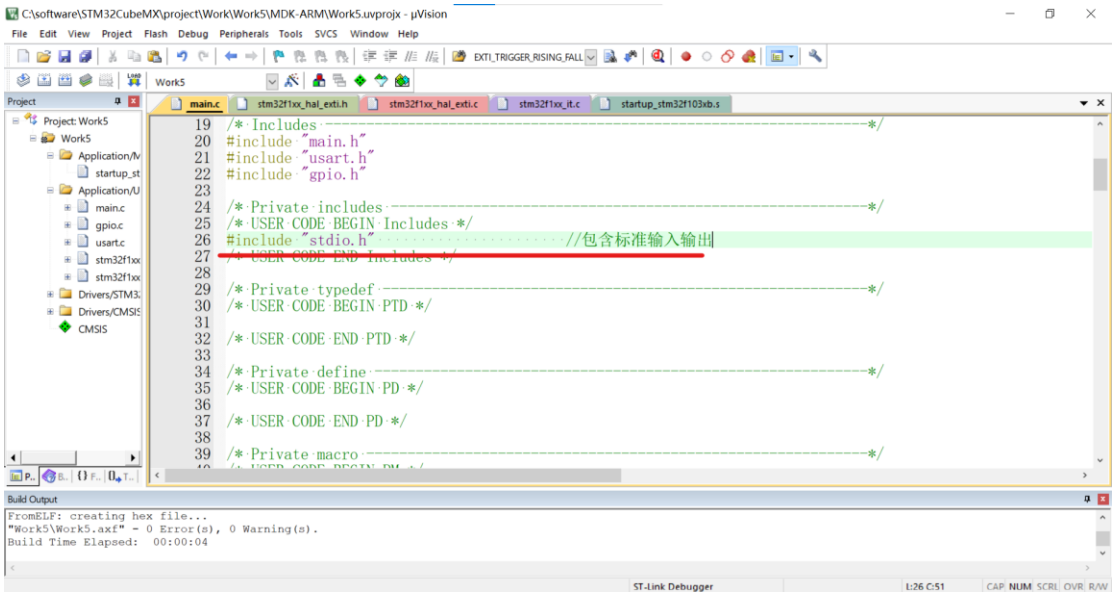
二、MDK 程序编写

1、设置复位下载选项和勾选 Use MicroLIB

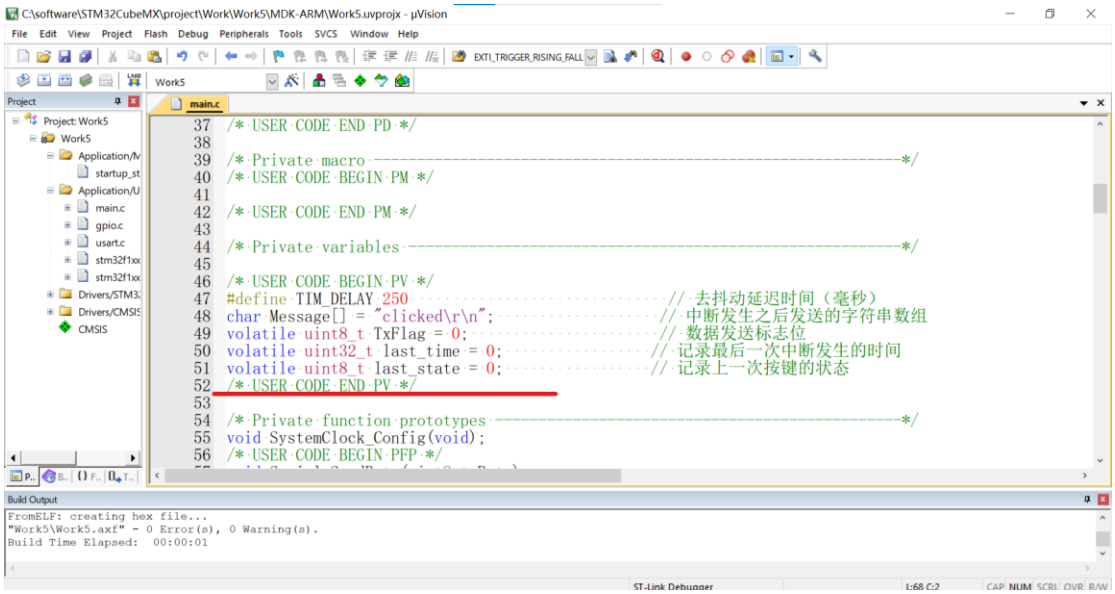

Options for Target 'Work2'
Use MicroLIB
Use Default compiler version 5
OK
Cancel
Defaults
Help


Options for Target 'Work1'
Use: ST-Link Debugger
Settings
Cortex-M Target Driver Setup
Debug
Trace
Flash Download
Erase Full Chip
Erase Sectors
Do not Erase
Program
Verify
Reset and Run
Start: 0x20000000
Size: 0x0800
STM32F10x Med-density ...
128k
On-chip Flash
08000000H - 0800FFFFH
Add
Remove
Verify OK.
Application running ...
Flash Load finished at 14:33:01

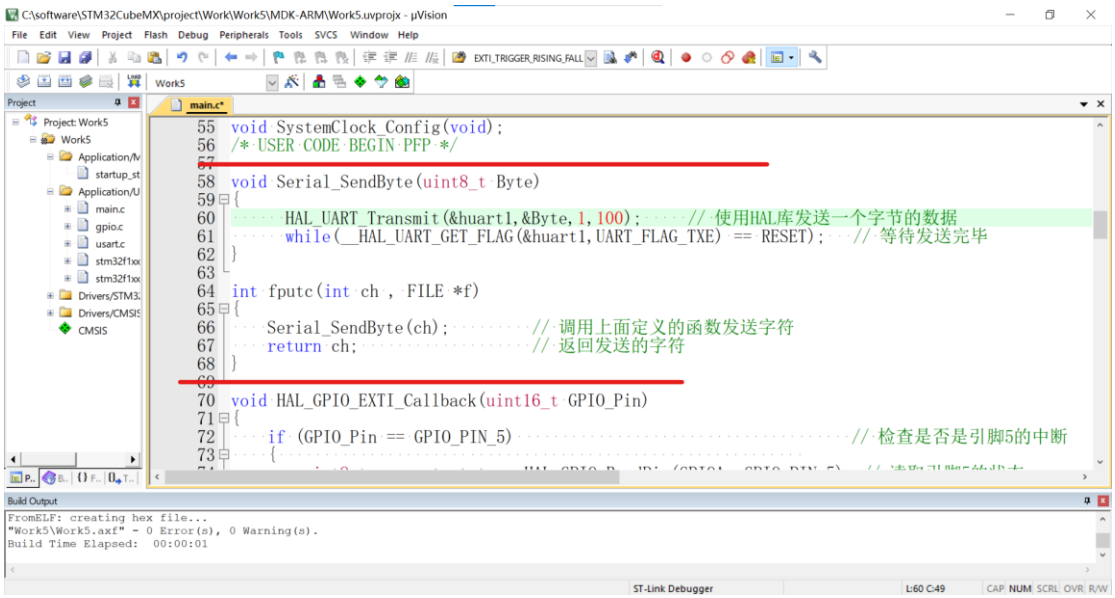
2、引用 C 库的头文件

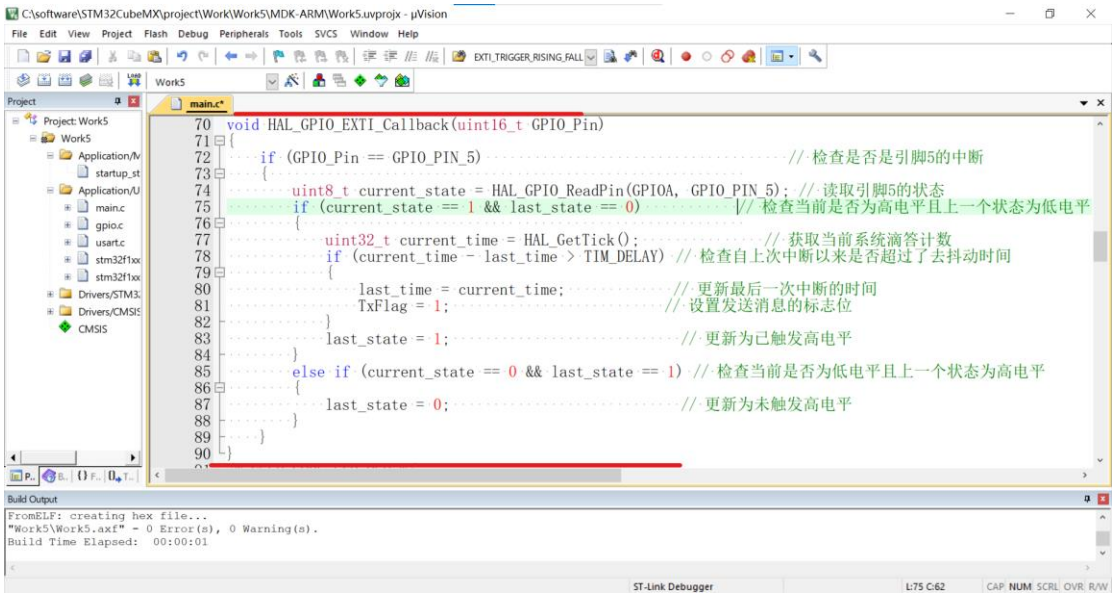


3、定义全局变量

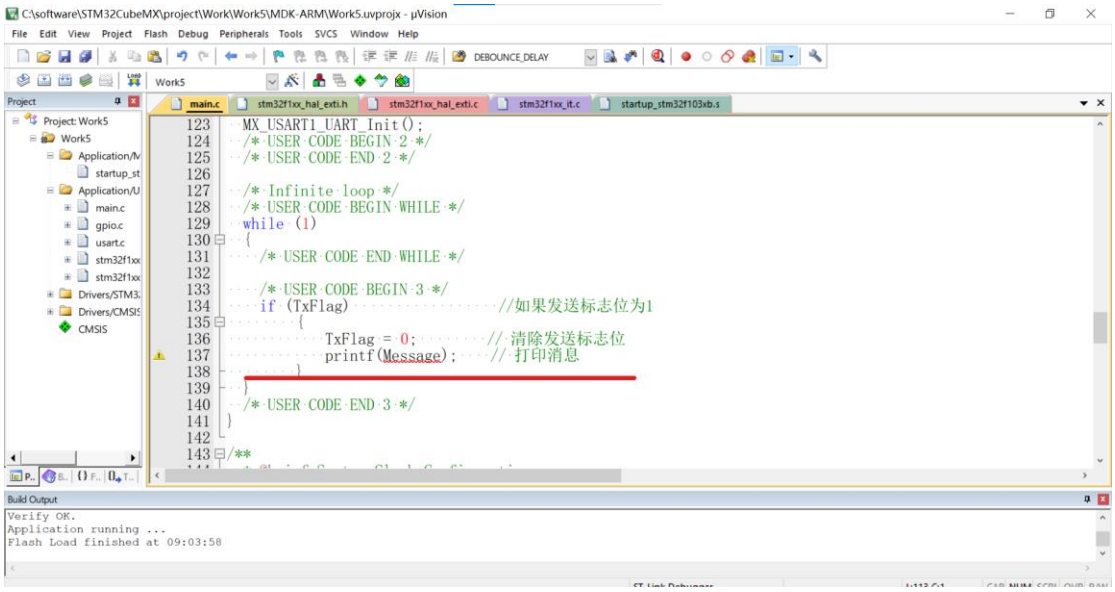


4、fputc 重定向、中断回调函数的重写





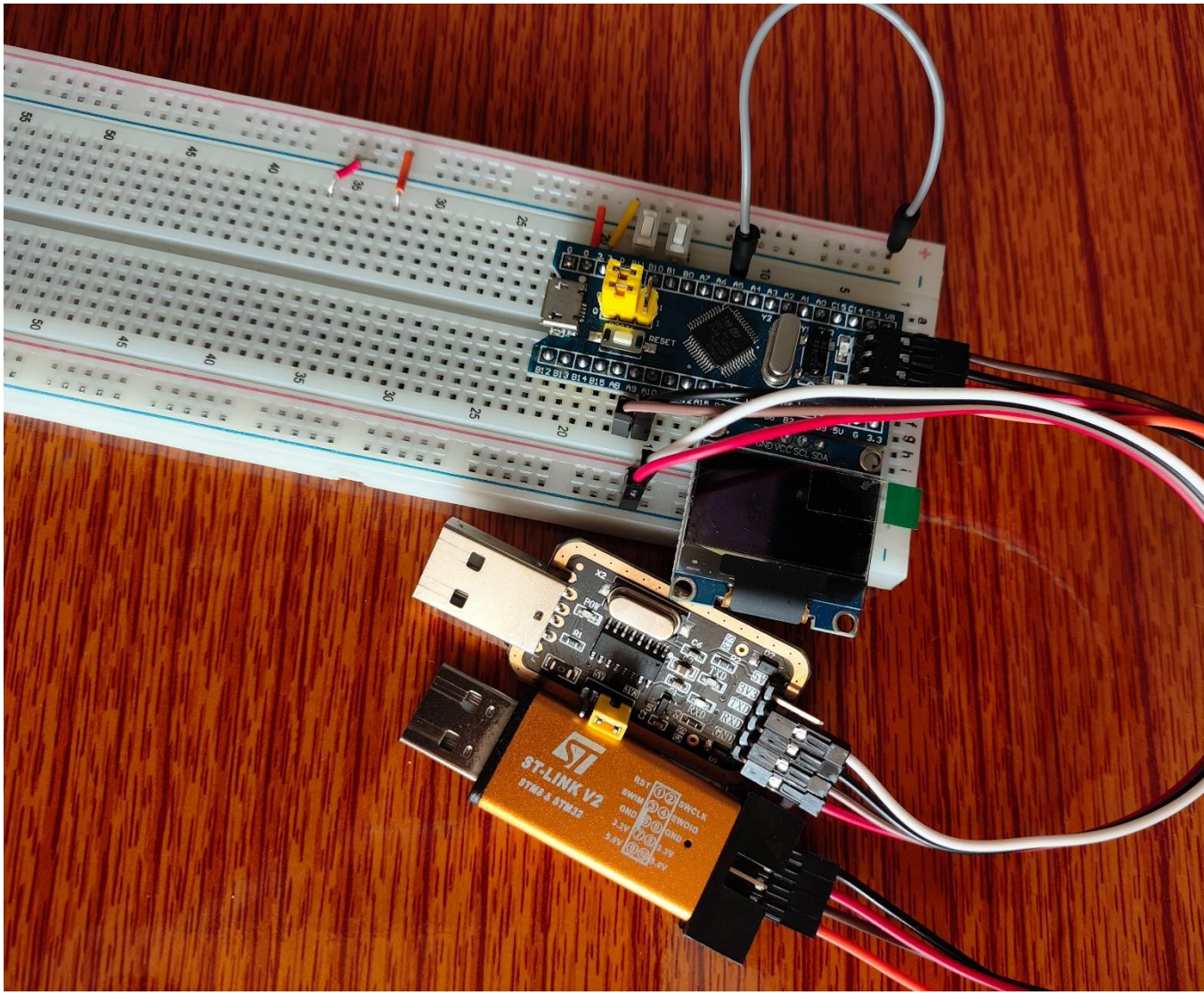
5、打印字符串数组消息，发送到串口软件



三、硬件连接

STlink	→	核心板	CH340	→	核心板
3.3V	→	3.3V	3.3V	→	3.3V
GND	→	GND	GND	→	GND
SWDIO	→	SWIO	TXD	→	PA10
SWCLK	→	SWCLK	RXD	→	PA9

增加一根飞线连接 3.3V 和 PA5 端口



四、代码部分

```
1.  /* Private includes ----------------------------------------------------------
--*/
2.  /* USER CODE BEGIN Includes */
3.  #include "stdio.h"
4.  /* USER CODE END Includes */
```

```
1.  /* USER CODE BEGIN PV */
2.  #define TIM_DELAY 250                          // 去抖动延迟时间（毫秒）
3.  char Message[] = "clicked\r\n";                // 中断发生之后发送的字符串数
组
4.  volatile uint8_t TxFlag = 0;                   // 数据发送标志位
5.  volatile uint32_t last_time = 0;               // 记录最后一次中断发生的时
间
6.  volatile uint8_t last_state = 0;               // 记录上一次按键的状态
```

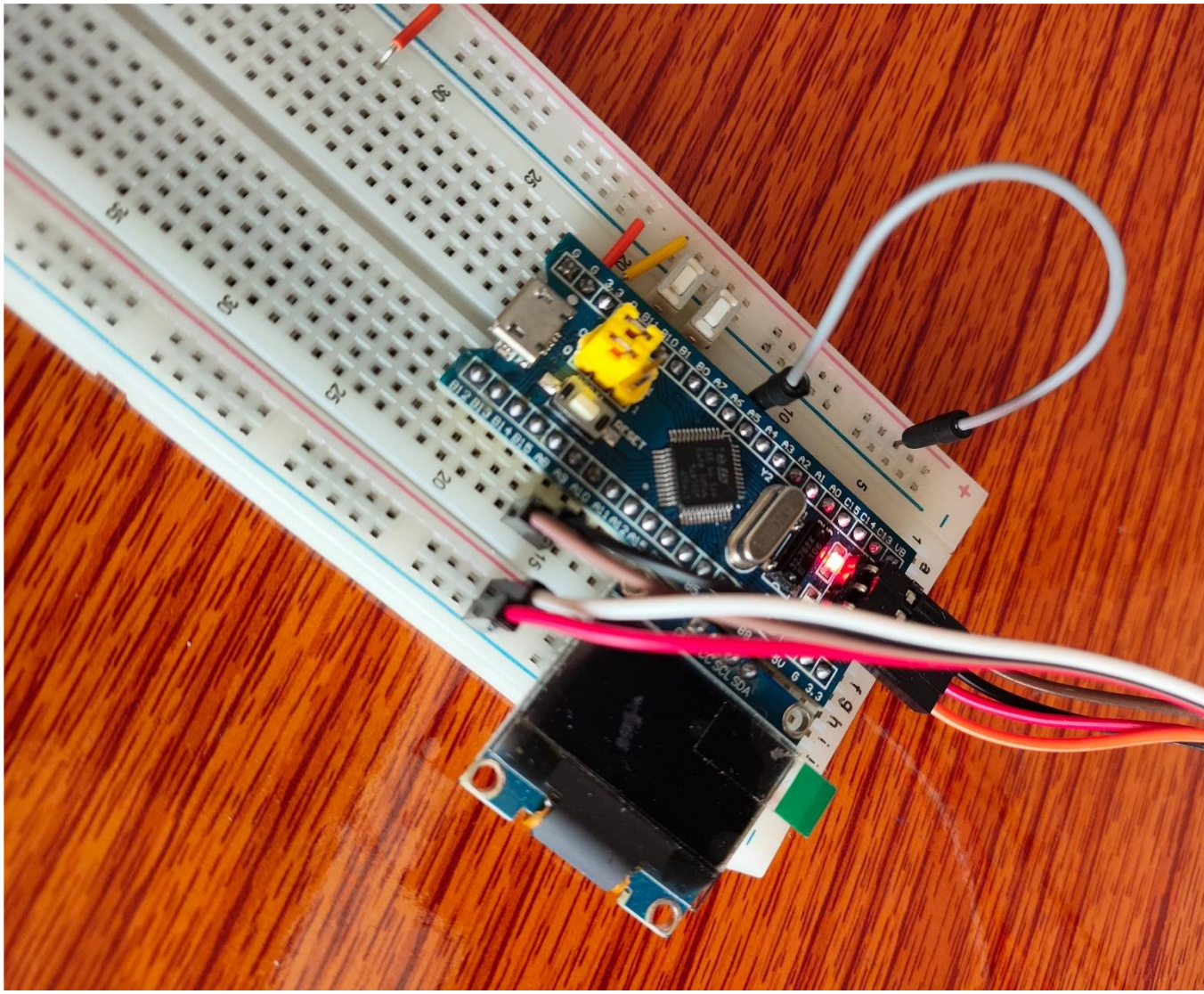
```
/* USER CODE END PV */
```

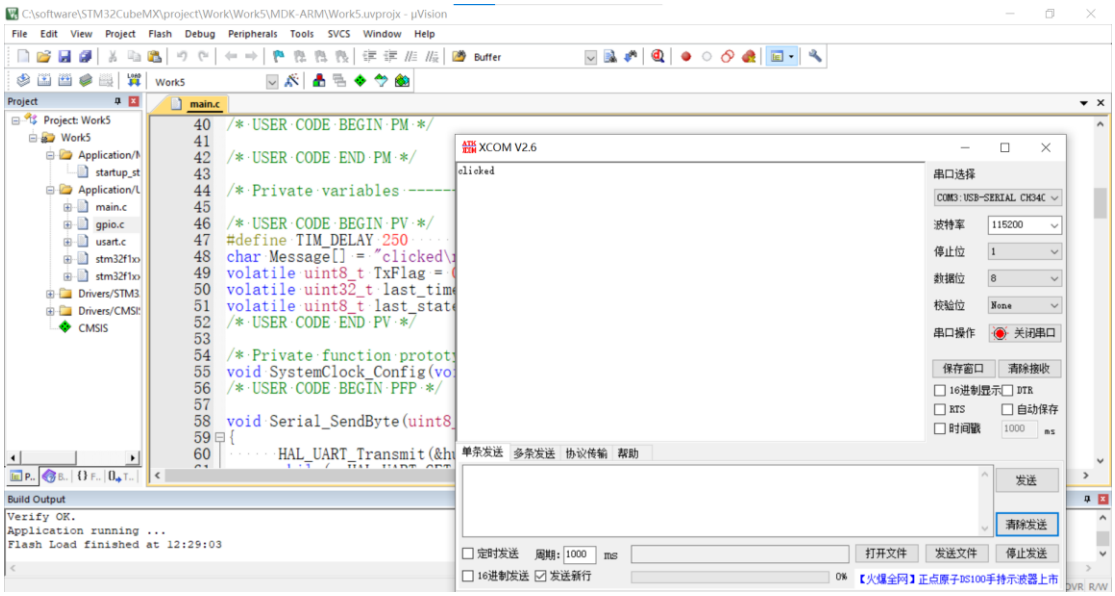
```
/* Private function prototypes -----------------------------------------------*/
void SystemClock_Config(void);
/* USER CODE BEGIN PFP */

void Serial_SendByte(uint8_t Byte)
{
      HAL_UART_Transmit(&huart1,&Byte,1,100);     // 使用 HAL 库发送一个字节的数据
      while(__HAL_UART_GET_FLAG(&huart1,UART_FLAG_TXE) == RESET);   // 等待发送完毕
}

int fputc(int ch , FILE *f)
{
    Serial_SendByte(ch);          // 调用上面定义的函数发送字符
    return ch;                    // 返回发送的字符
}

void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin)
{
    if (GPIO_Pin == GPIO_PIN_5)                                    // 检查是否是引脚 5 的中断
    {
        uint8_t current_state = HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_5); // 读取引脚 5 的状态
        if (current_state == 1 && last_state == 0)           // 检查当前是否为高电平且上一个状态为低电平
        {
            uint32_t current_time = HAL_GetTick();               // 获取当前系统滴答计数
            if (current_time - last_time > TIM_DELAY) // 检查自上次中断以来是否超过了去抖动时间
            {
                last_time = current_time;             // 更新最后一次中断的时间
                TxFlag = 1;                           // 设置发送消息的标志位
            }
            last_state = 1;                             // 更新为已触发高电平
        }
        else if (current_state == 0 && last_state == 1) // 检查当前是否为低电平且上一个状态为高电平
```

```
            {
                last_state = 0;                                    // 更新为未触发高电平
            }
        }
    }
    /* USER CODE END PFP */
```

```
    /* Infinite loop */
    /* USER CODE BEGIN WHILE */
    while (1)
    {
      /* USER CODE END WHILE */

      /* USER CODE BEGIN 3 */
      if (TxFlag)                  //如果发送标志位为 1
      {
          TxFlag = 0;             // 清除发送标志位
          printf(Message);        // 打印消息
      }
    }
    /* USER CODE END 3 */
```

五、实现效果




C:\software\STM32CubeMX\project\Work\Work5\MDK-ARM\Work5.uvprojx - µVision
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Project
Project: Work5
Work5
main.c
gpio.c
usart.c
CMSIS
main.c
/* USER CODE BEGIN PM */
/* USER CODE END PM */
/* Private variables
/* USER CODE BEGIN PV */
#define TIM_DELAY 250
char Message[] = "clicked\
volatile uint8_t TxFlag =
volatile uint32_t last_tim
volatile uint8_t last_stat
/* USER CODE END PV */
/* Private function prototy
void SystemClock_Config(vo
/* USER CODE BEGIN PFP */
void Serial_SendByte(uint8_
{
HAL_UART_Transmit(&hu
Build Output
Verify OK.
Application running ...
Flash Load finished at 12:29:03
XCOM V2.6
clicked
串口选择
COM3:USB-SERIAL CH34C
波特率 115200
停止位 1
数据位 8
校验位 None
串口操作 关闭串口
保存窗口 清除接收
16进制显示 DTR
RTS 自动保存
时间戳 1000 ms
单条发送 多条发送 协议传输 帮助
发送
清除发送
定时发送 周期: 1000 ms
打开文件 发送文件 停止发送
16进制发送 发送新行
0%
【火爆全网】正点原子DS100手持示波器上市