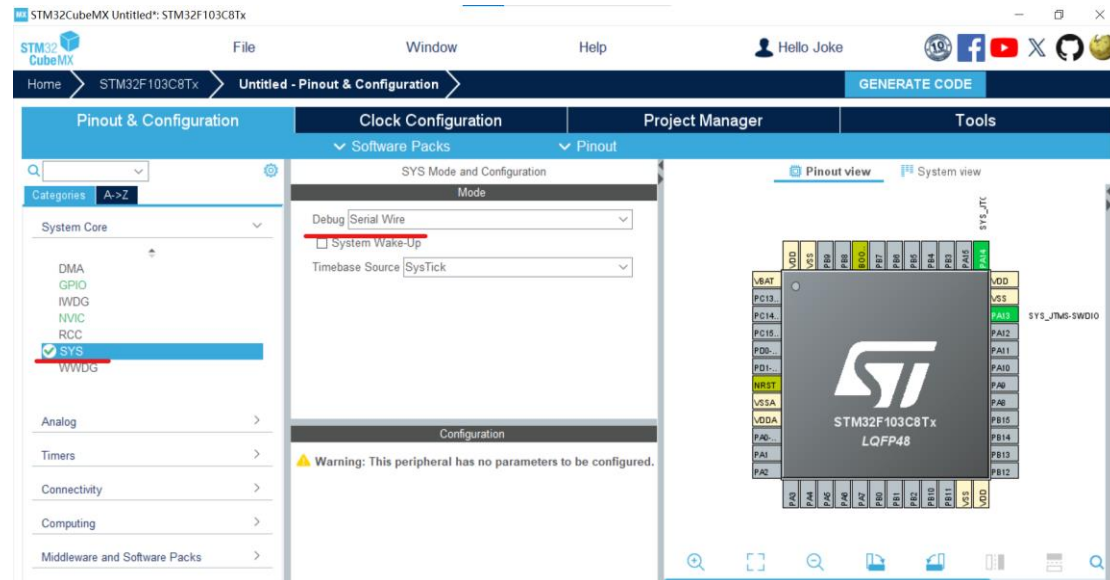


任务要求：使用串口空闲中断，输入字符串“123”时，PC13 亮，当输入字符串“456”时，PC13 灭，其他状态保持不变。

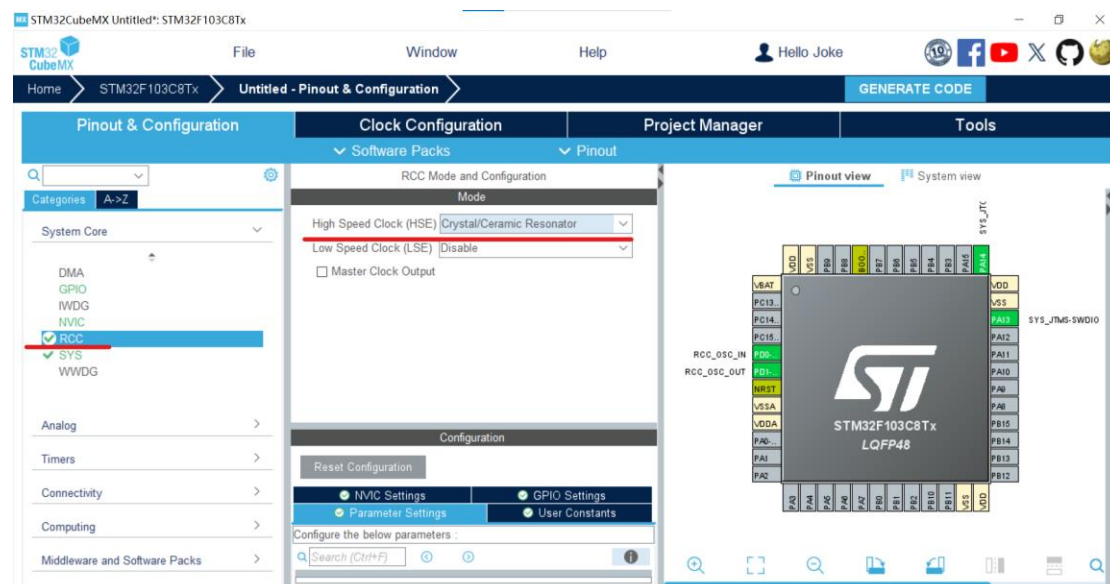
使用到的外设：GPIO、USART 串口

## 一、STM32CubeMX 创建工程步骤

### 1、开启 SYS 系统时钟

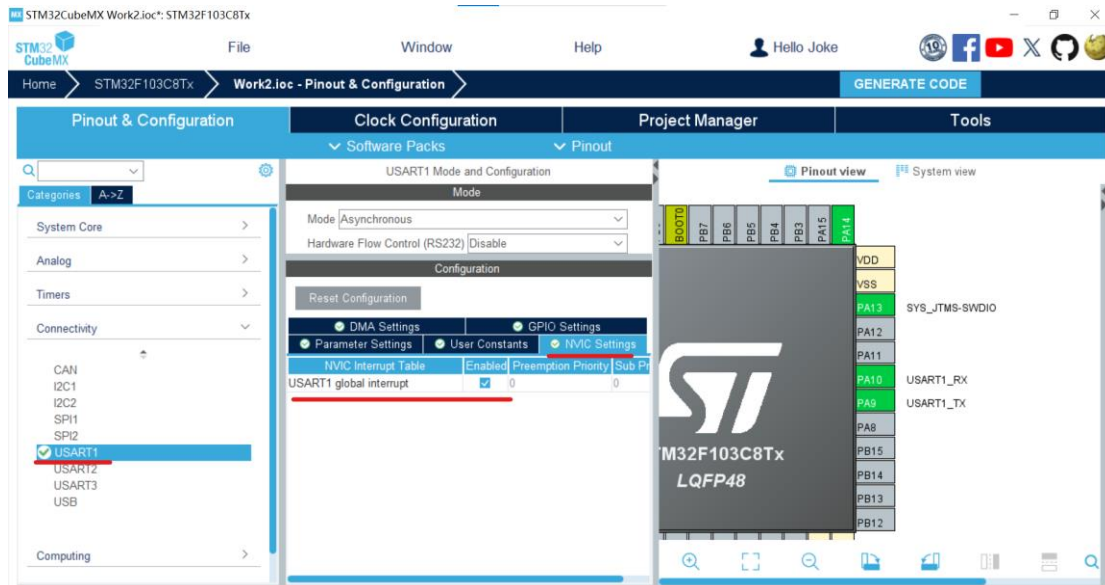


### 2、开启 RCC 晶振

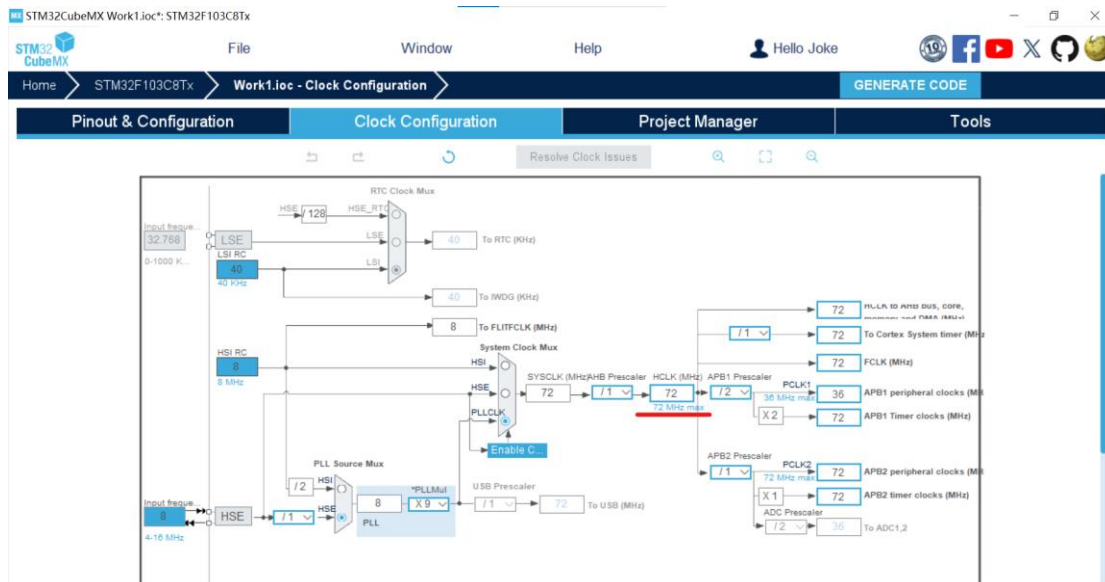


3、选择 PC13 端口设置为 GPIO\_Output 模式，设置为推挽输出，并且初始化电平为高电平

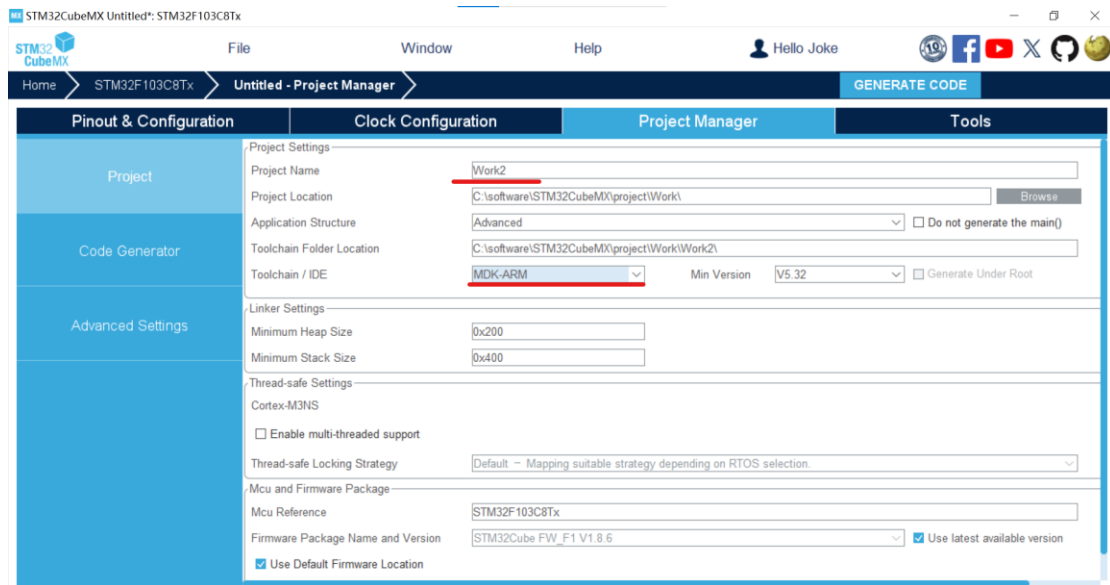




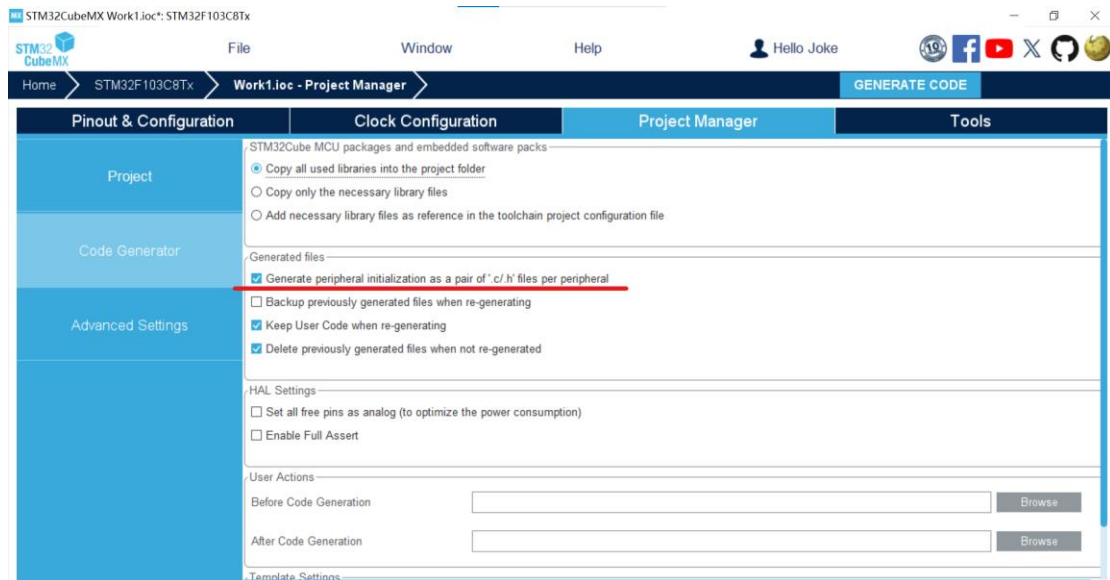
## 6、时钟树选择 72MHZ



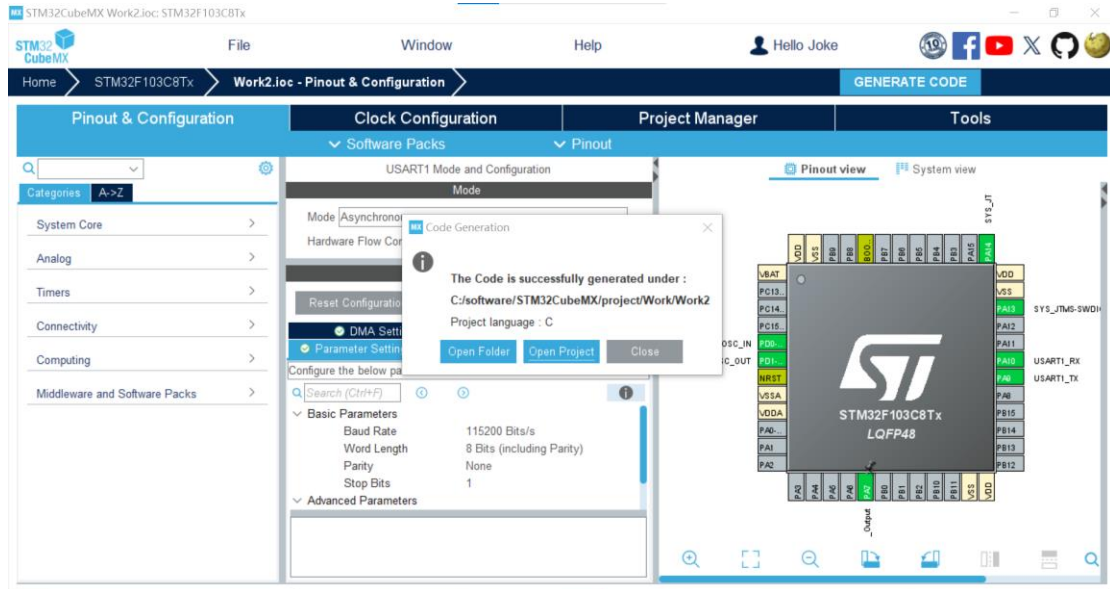
## 7、创建文件名，并存在对应的盘中，选择 MDK-ARM



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

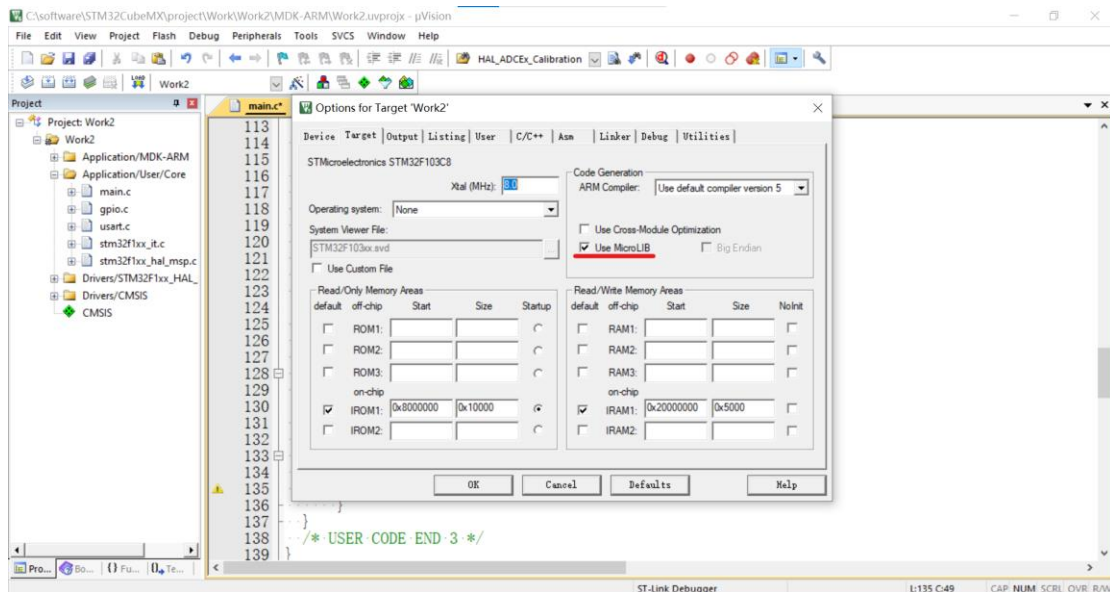


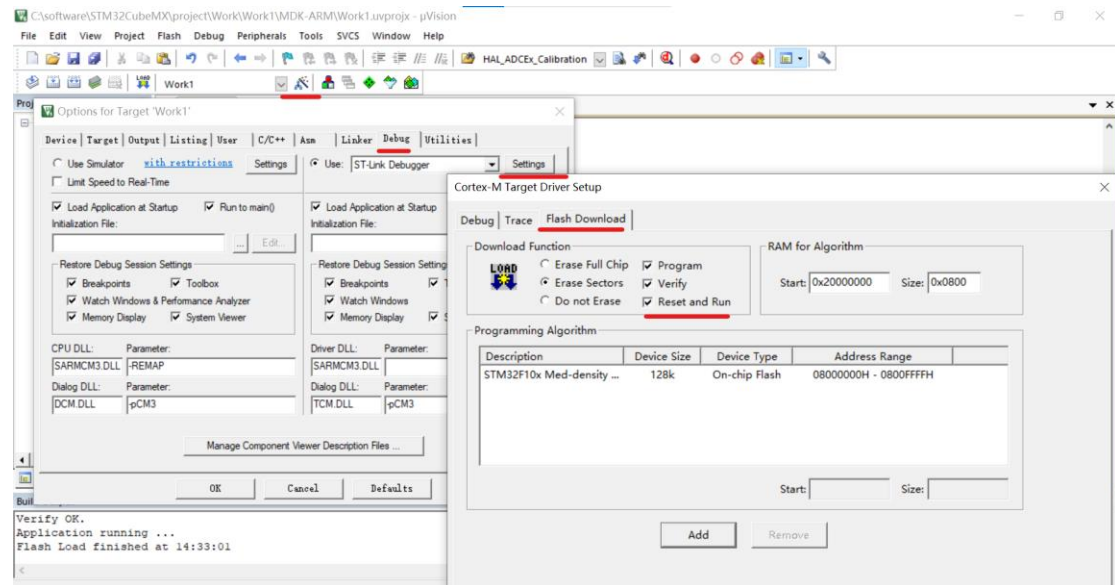
9、生成文件 GENERATE CODE，点击 CLOSE，并打开所生成的路径的 MDK 文件



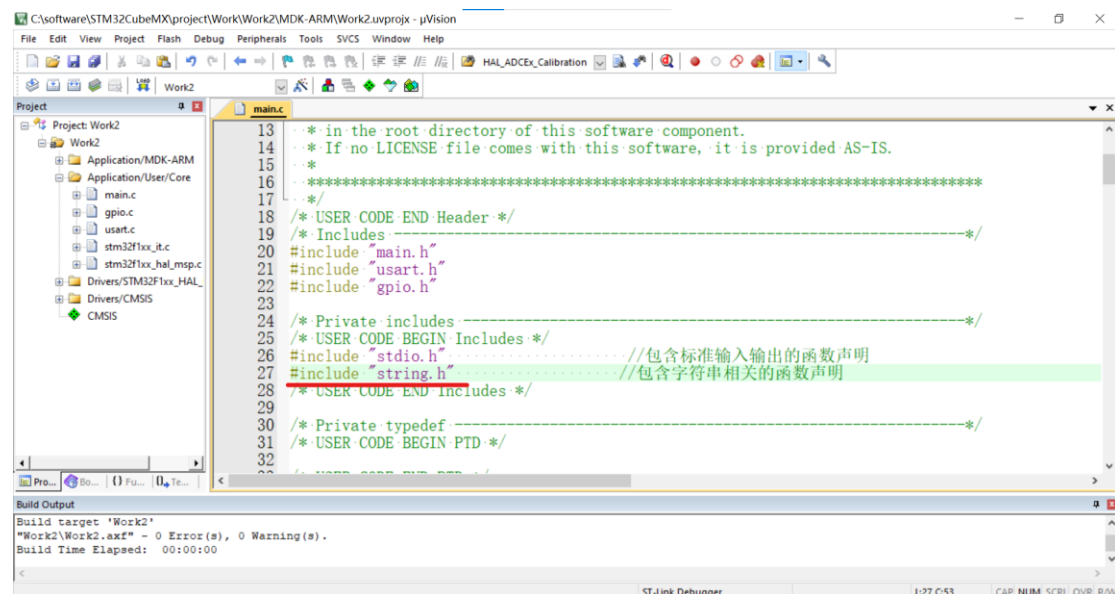
## 二、MDK 程序编写

### 1、设置复位下载选项和勾选 Use MicroLIB



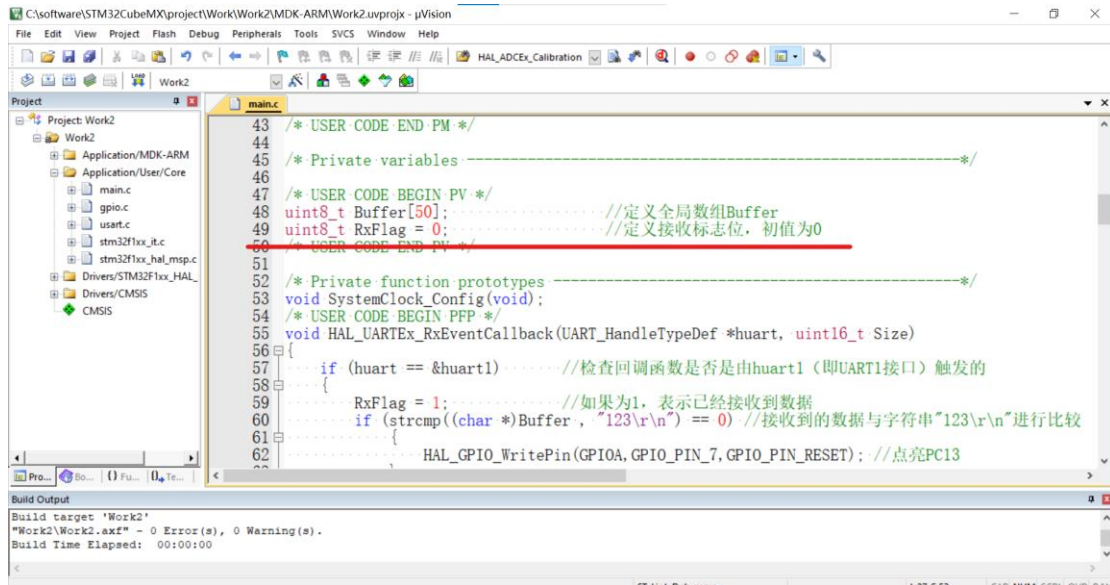


## 2、引用 C 库的头文件

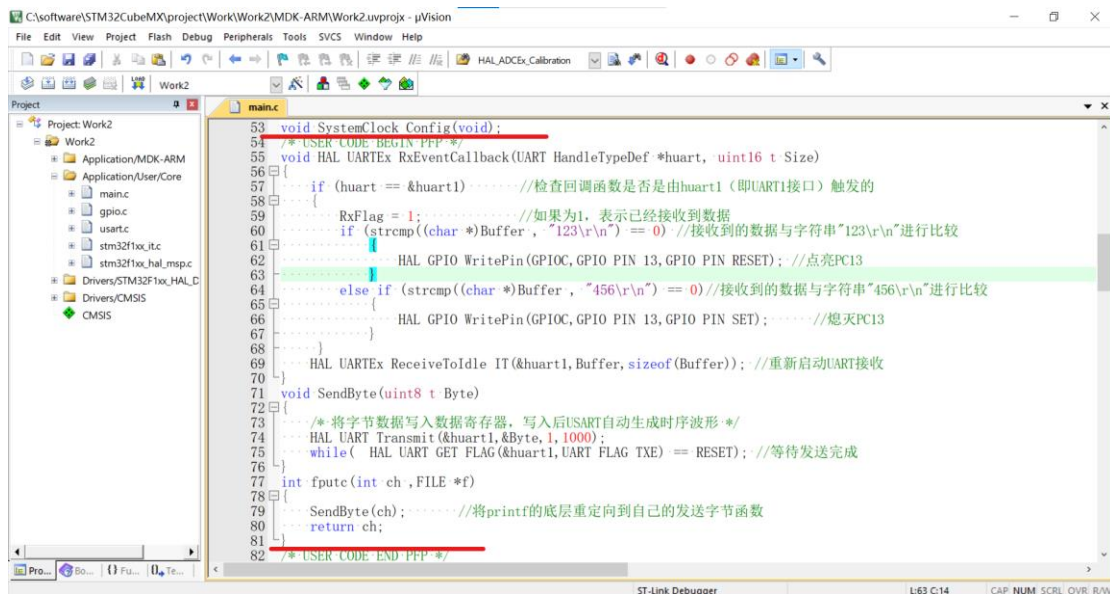


## 3、定义全局变量

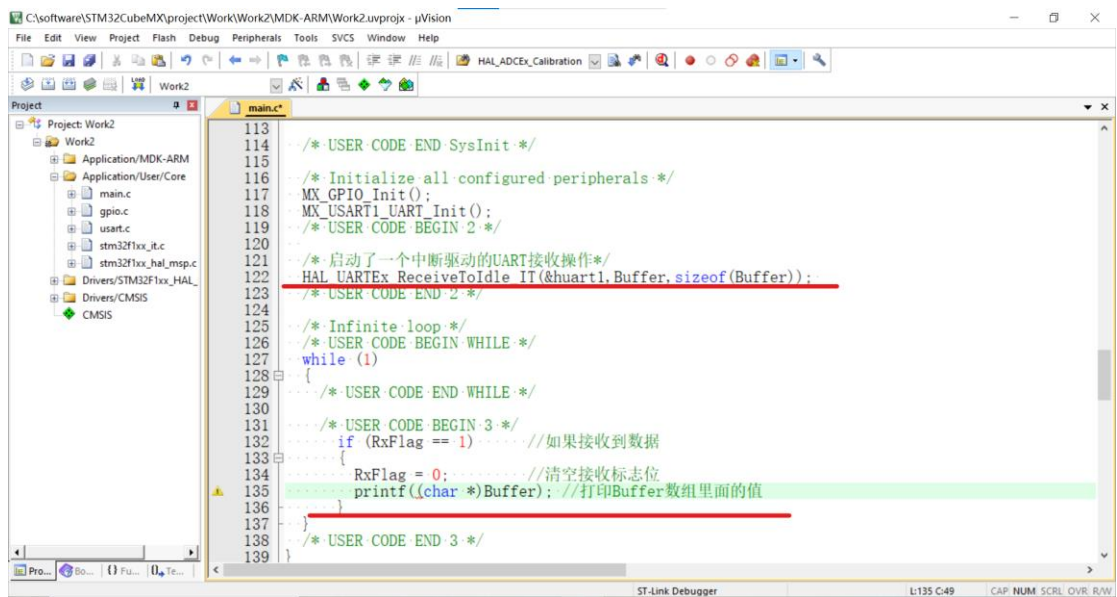




#### 4、重写中断回调函数，创建发送字节函数和 fputc 重定向



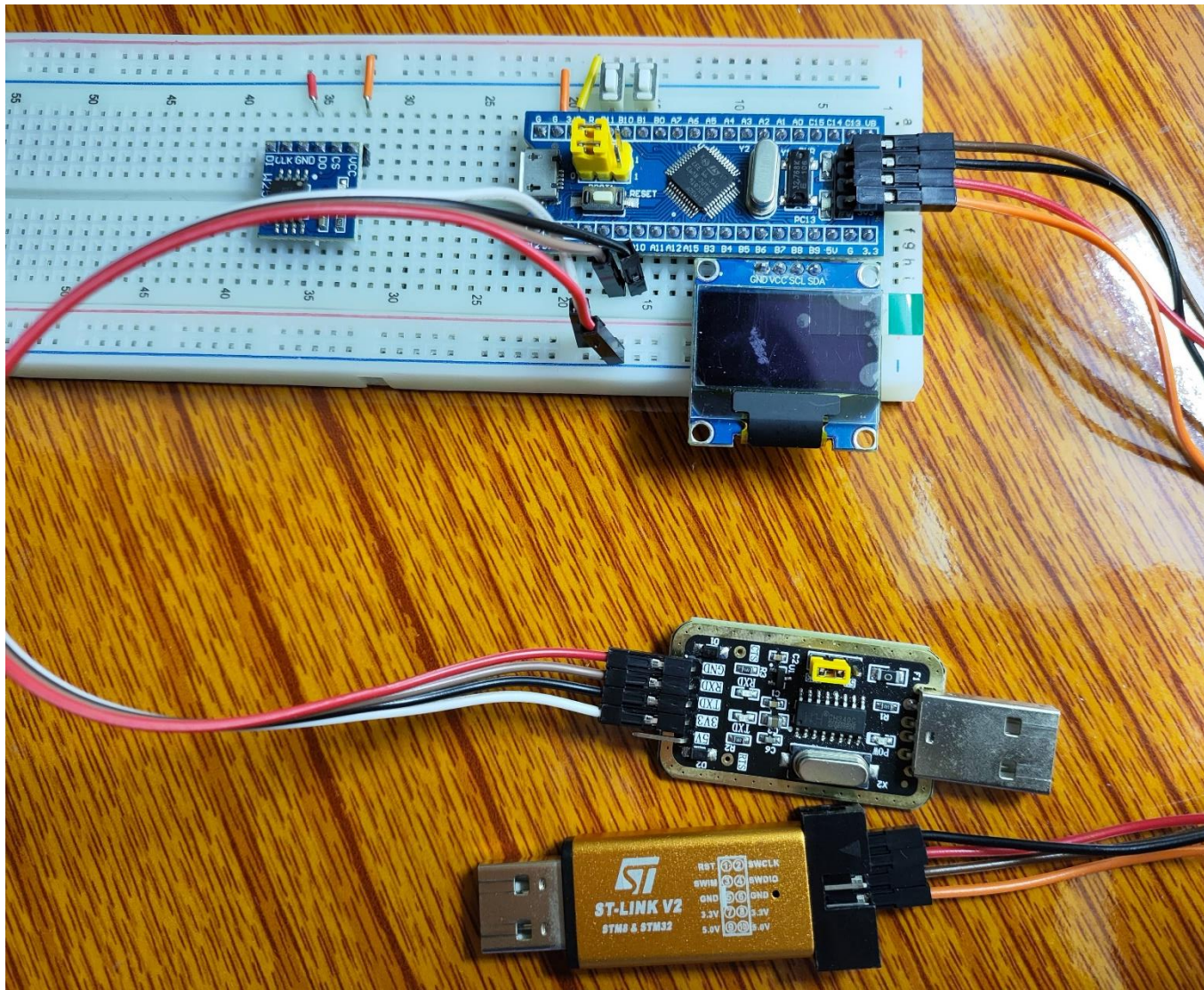
#### 5、开启接收中断，打印接收到的数据发回串口软件



### 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





## 四、代码部分

```

1.      /* Private includes -----
--*/

2.      /* USER CODE BEGIN Includes */

3.      #include "stdio.h"                //包含标准输入输出的函数声明
4.      #include "string.h"              //包含字符串相关的函数声明
5.      /* USER CODE END Includes */


1.      /* USER CODE BEGIN PV */
2.      uint8_t Buffer[50];                //定义全局数组 Buffer
3.      uint8_t RxFlag = 0;                //定义接收标志位，初值为0
4.      /* USER CODE END PV */


1.      /* Private function prototypes -----
--*/

```

```

2.     void SystemClock_Config(void);
3.     /* USER CODE BEGIN PFP */
4.     void HAL_UARTEx_RxEventCallback(UART_HandleTypeDef *huart, uint16_t Size)
5.     {
6.         if (huart == &huart1)           //检查回调函数是否是由 huart1（即 UART1 接口）触
发的
7.         {
8.             RxFlag = 1;                   //如果为 1，表示已经接收到数据
9.             if (strcmp((char *)Buffer , "123\r\n") == 0) //接收到的数据与字符串
"123\r\n"进行比较
10.            {
11.                HAL_GPIO_WritePin(GPIOC,GPIO_PIN_13,GPIO_PIN_RESET); //点亮
PC13
12.            }
13.            else if (strcmp((char *)Buffer , "456\r\n") == 0)//接收到的数据与字符
串"456\r\n"进行比较
14.            {
15.                HAL_GPIO_WritePin(GPIOC,GPIO_PIN_13,GPIO_PIN_SET);      //熄
灭 PC13
16.            }
17.        }
18.        HAL_UARTEx_ReceiveToIdle_IT(&huart1,Buffer,sizeof(Buffer)); //重新启动
UART 接收
19.    }
20.
21.    void SendByte(uint8_t Byte)
22.    {
23.        /* 将字节数据写入数据寄存器，写入后 USART 自动生成时序波形 */
24.        HAL_UART_Transmit(&huart1,&Byte,1,1000);
25.        while(__HAL_UART_GET_FLAG(&huart1,UART_FLAG_TXE) == RESET); //等待发送完
成
26.    }
27.
28.    int fputc(int ch ,FILE *f)
29.    {
30.        SendByte(ch);           //将 printf 的底层重定向到自己的发送字节函数
31.        return ch;
32.    }
33.    /* USER CODE END PFP */

1.    /* USER CODE BEGIN 2 */
2.
3.    /* 启动了一个中断驱动的 UART 接收操作*/
4.    HAL_UARTEx_ReceiveToIdle_IT(&huart1,Buffer,sizeof(Buffer));

```

```

5.      /* USER CODE END 2 */
6.
7.      /* Infinite loop */
8.      /* USER CODE BEGIN WHILE */
9.      while (1)
10.     {
11.         /* USER CODE END WHILE */
12.
13.         /* USER CODE BEGIN 3 */
14.         if (RxFlag == 1)      //如果接收到数据
15.         {
16.             RxFlag = 0;      //清空接收标志位
17.             printf((char *)Buffer); //打印 Buffer 数组里面的值
18.         }
19.     }
20.     /* USER CODE END 3 */

```

## 五、实现效果

