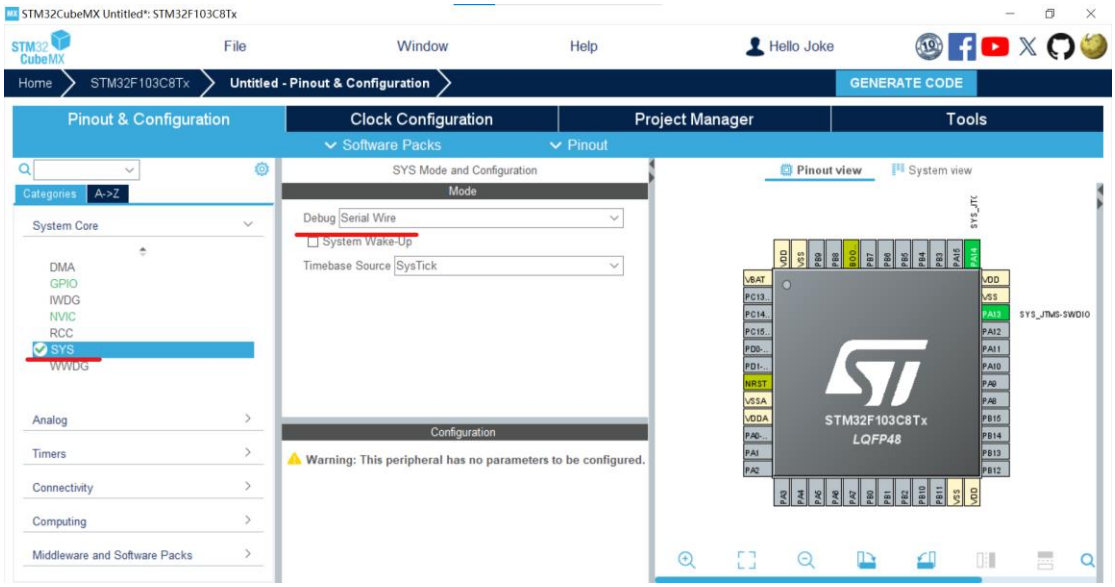
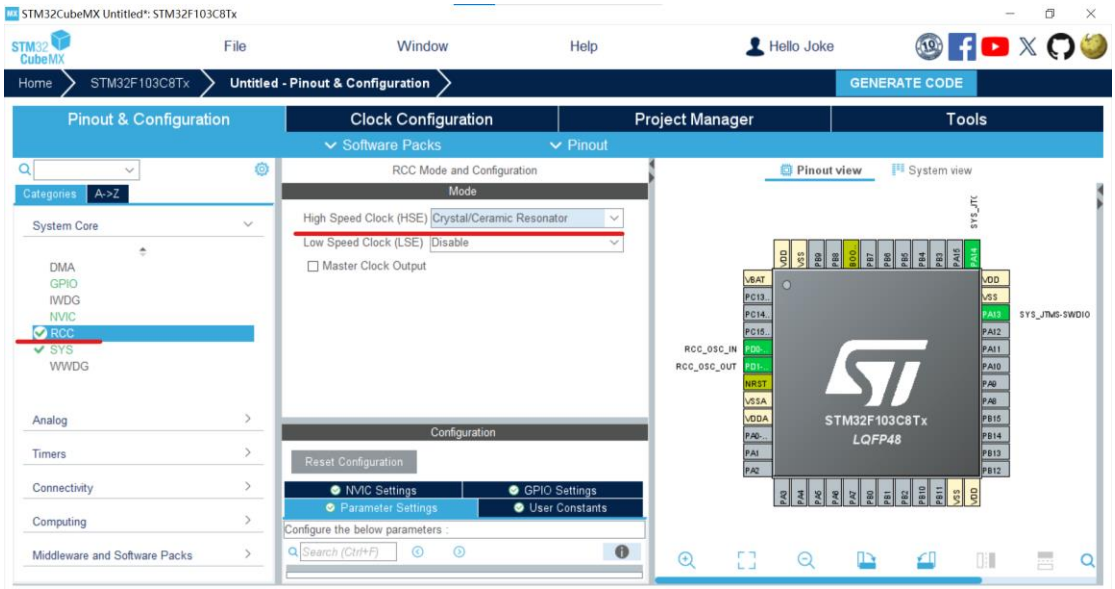
任务要求：使用**串口空闲中断**，使用串口 1，PA9,PA10，以及定时器 1，PA8,定时器 ARR 设置为 7199，Prescaler 设置为 0。
当串口发送 0~7199 时，对 PA8 端口的 LED 进行对应亮度的调节。（PWM 调节）
设置 0 为亮度最小，设置 7199 亮度最大;
发送数据超出范围会发送” report errors! Please enter a numerical value in the range of 0-7199”返回串口提示
使用到的外设：**GPIO、USART 串口、TIM**

一、STM32CubeMX 创建工程步骤

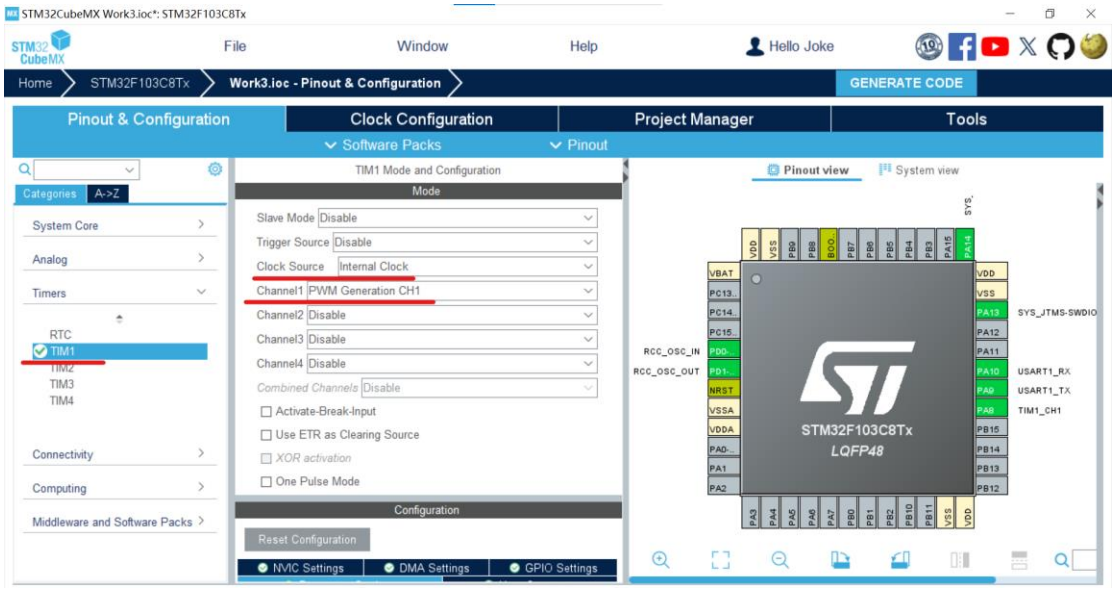
1、开启 SYS 系统时钟



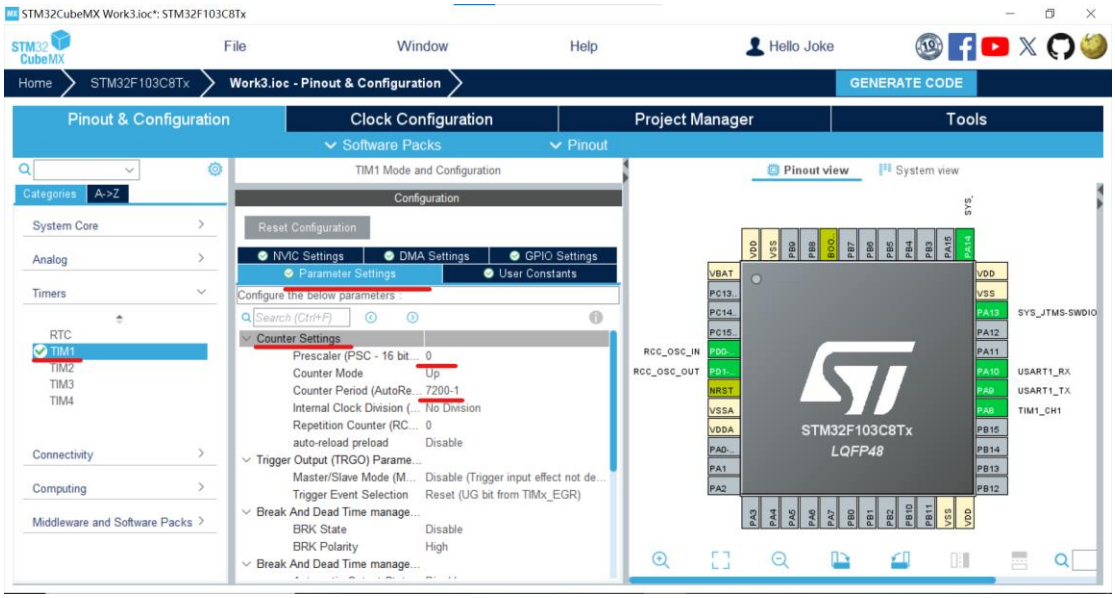
2、开启 RCC 晶振

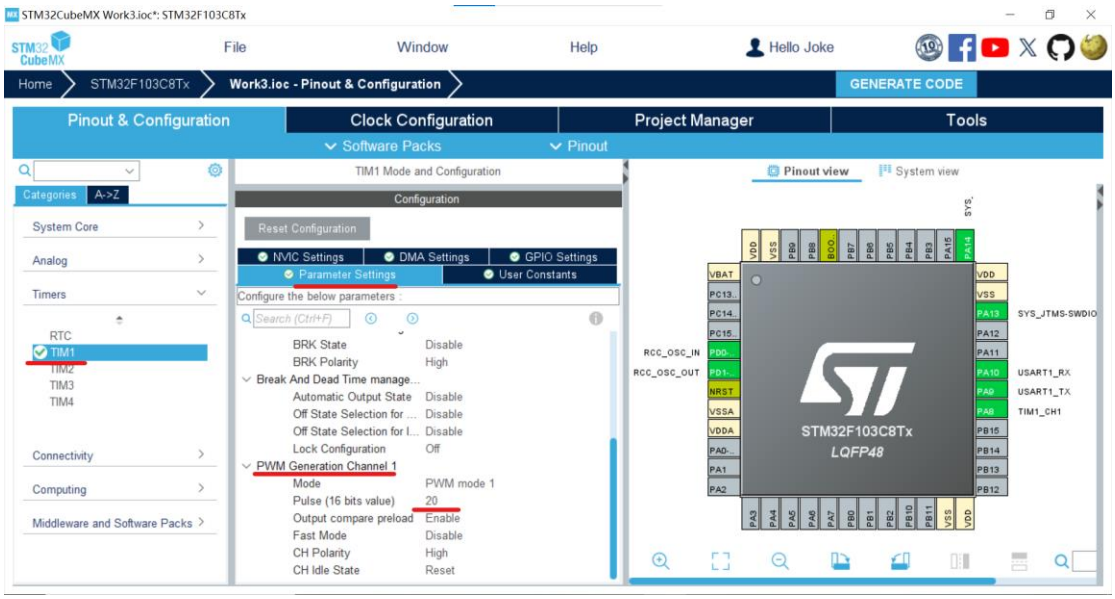


3、选择 TIM1，选择内部时钟，选择 PWM Generated CH1

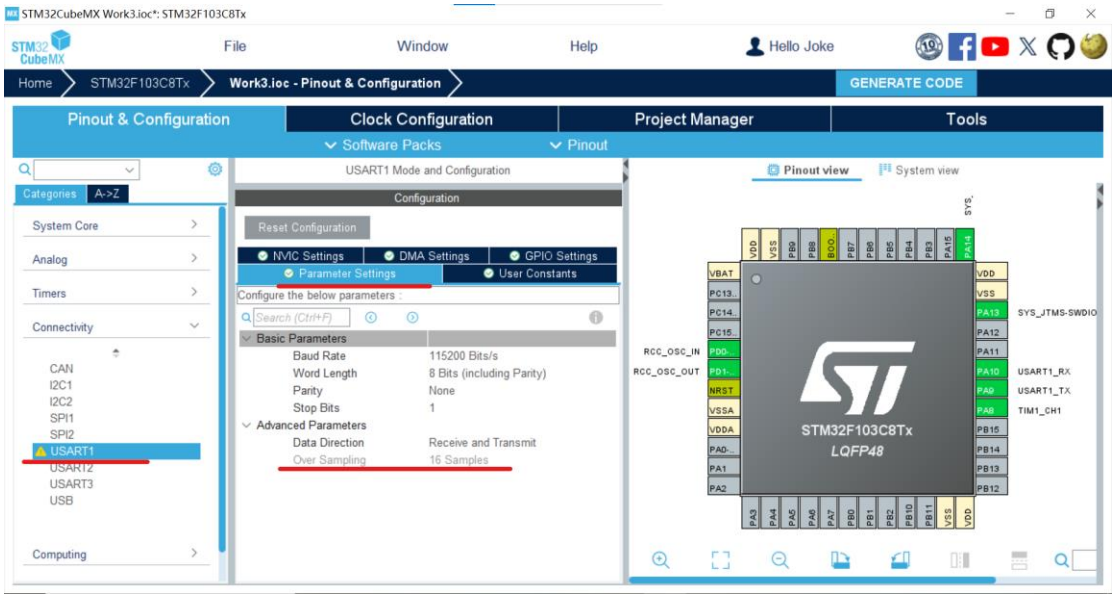


4、设置 PSC 预分频器为 0，自动重装寄存器 ARR 为 7199，再设置 Pulse 为 20，设置了 Pulse 才会有 PWM 波形

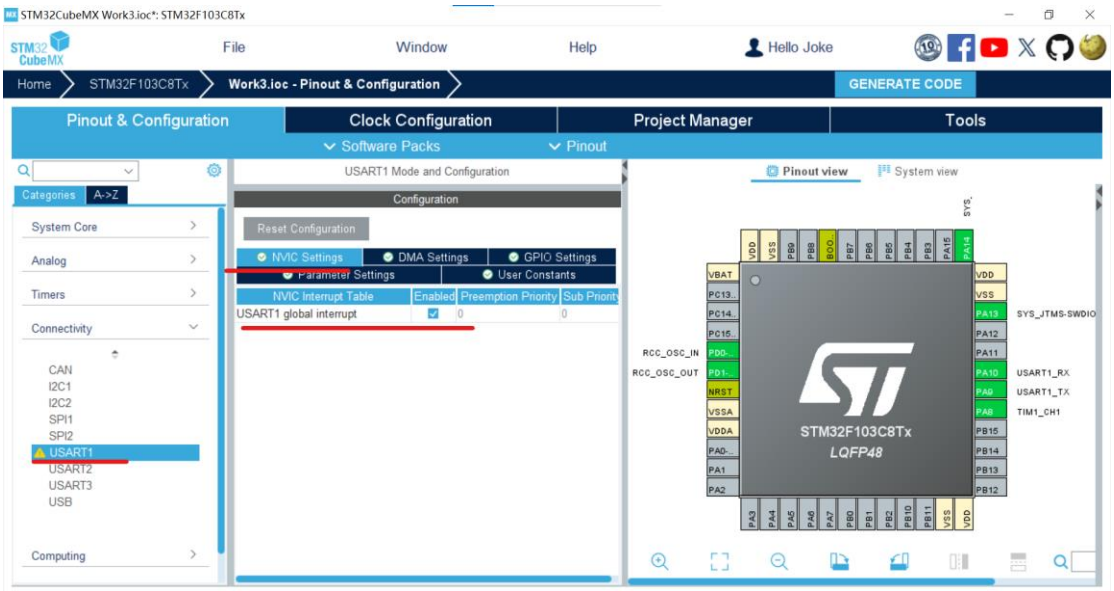




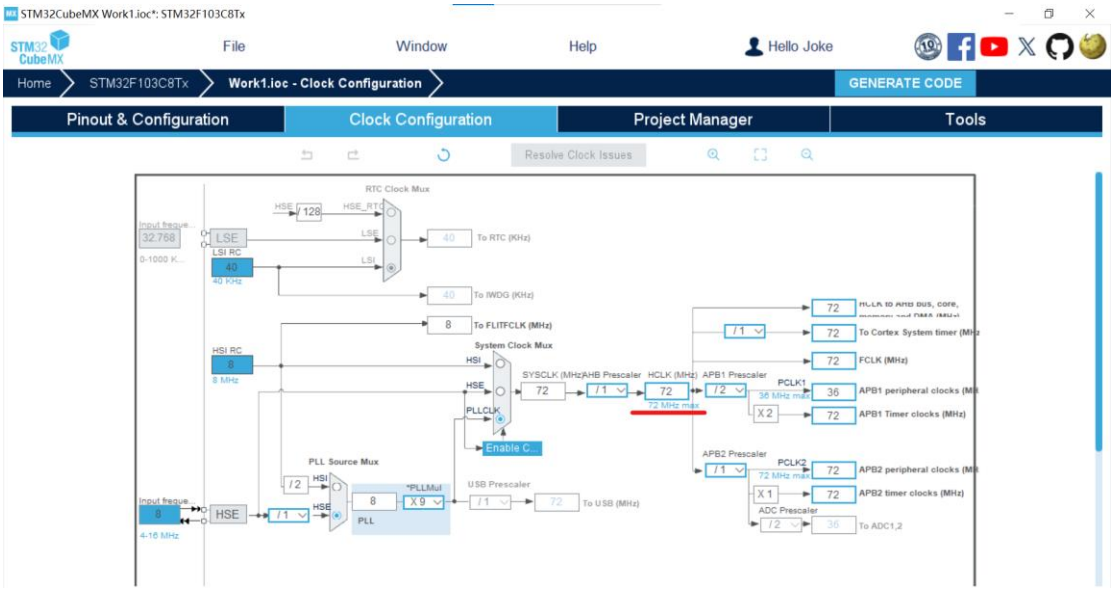
5、选择 USART1 作为串口通信，波特率设置为 115200， 8 位字节数据位，无校验位，1 位停止位，选择发送和接收模式



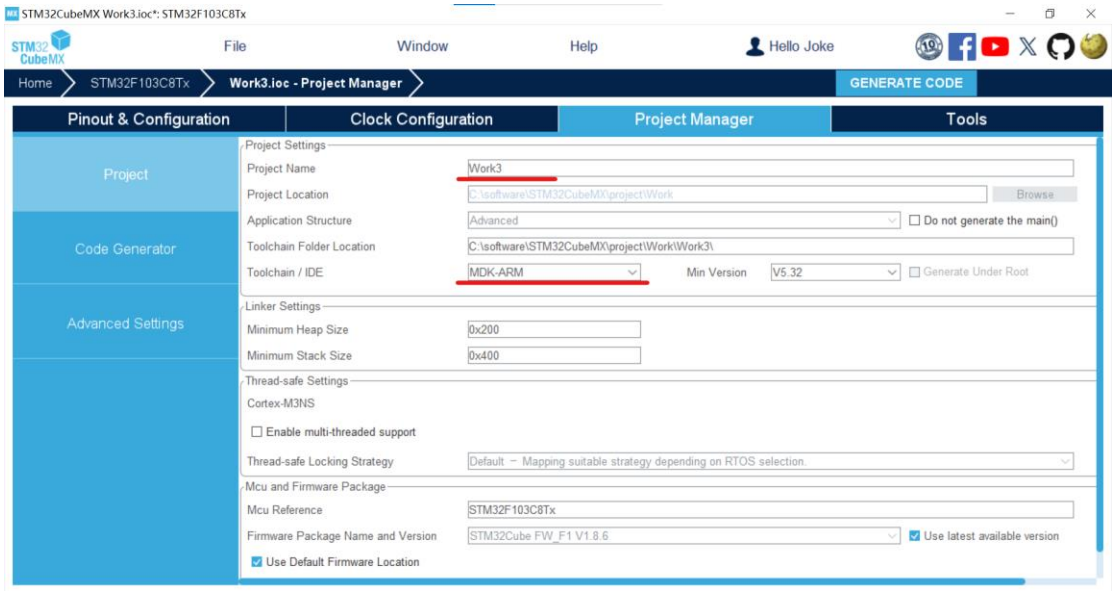
6、开启串口中断



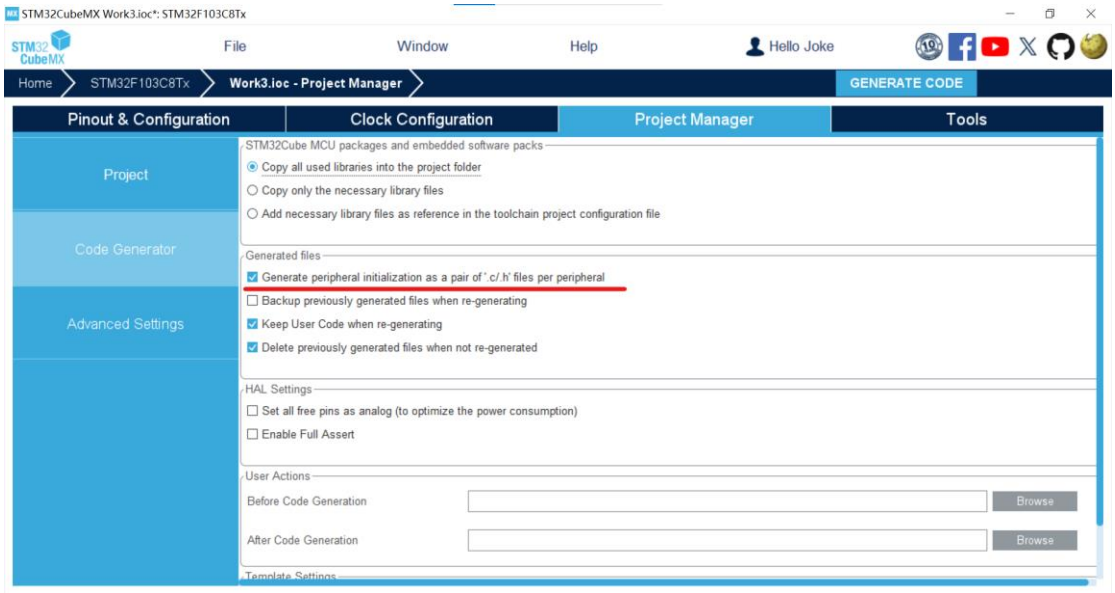
7、时钟树选择 72MHZ



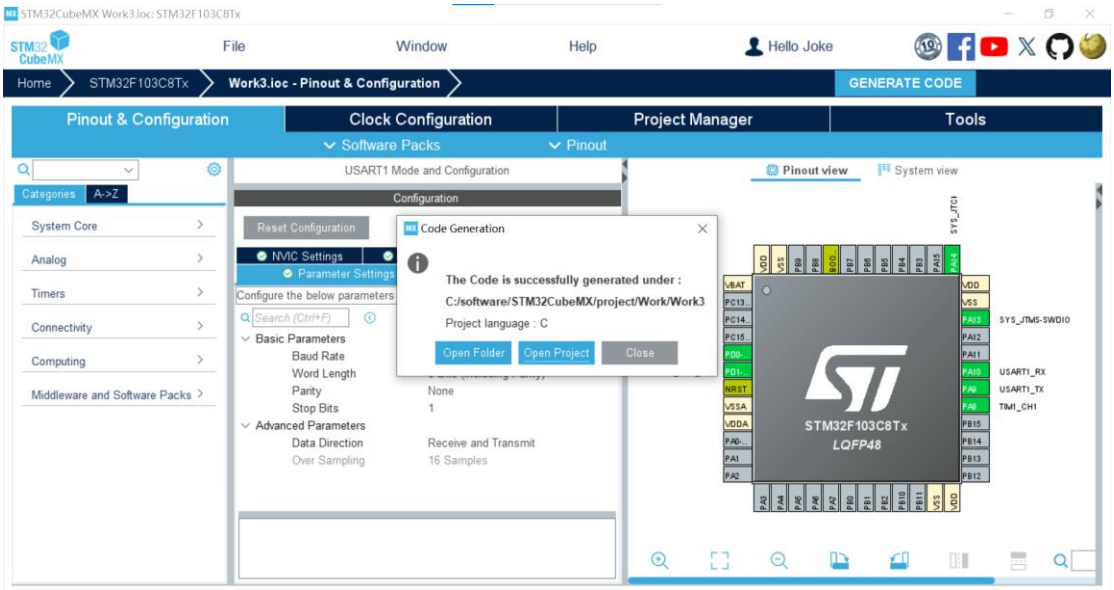
8、创建文件名，并存在对应的盘中，选择 MDK-ARM



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

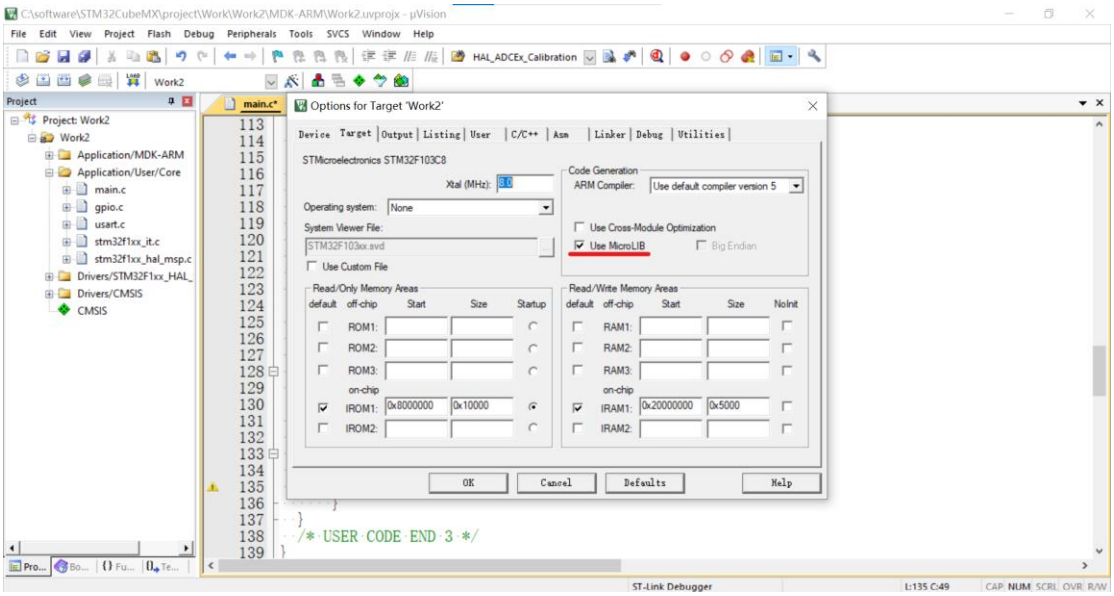


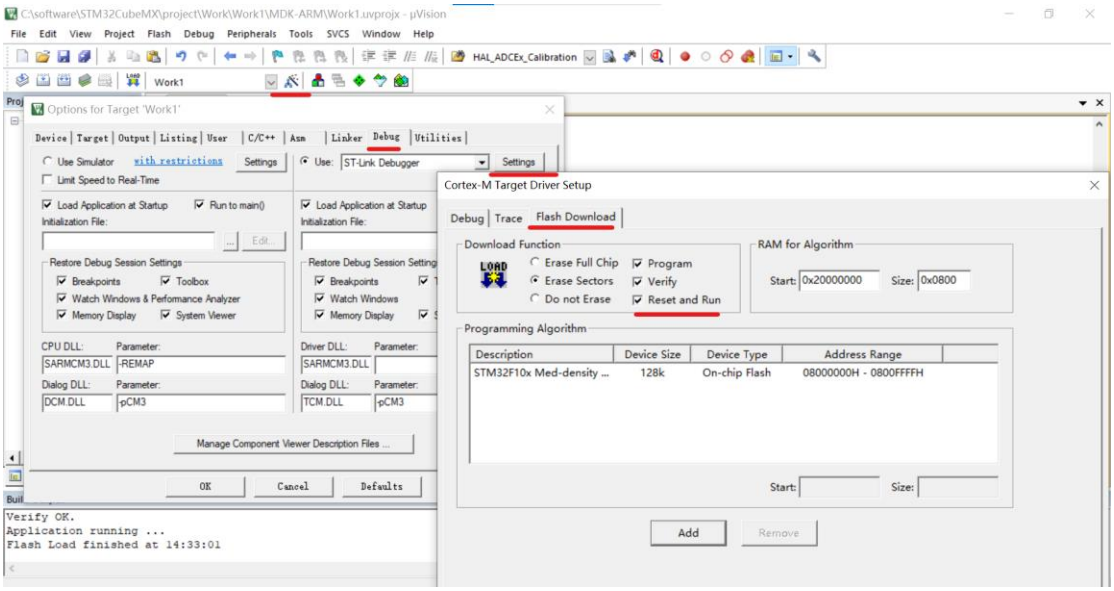
10、生成文件 GENERATE CODE，点击 CLOSE，并打开所生成的路径的 MDK 文件



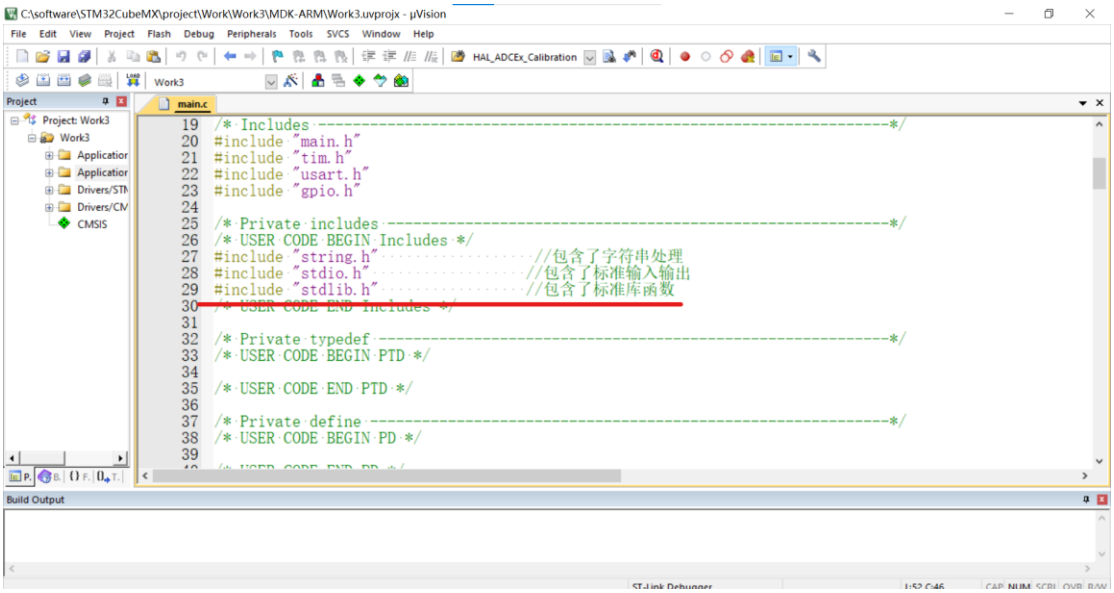
二、MDK 程序编写

1、设置复位下载选项和勾选 Use MicroLIB

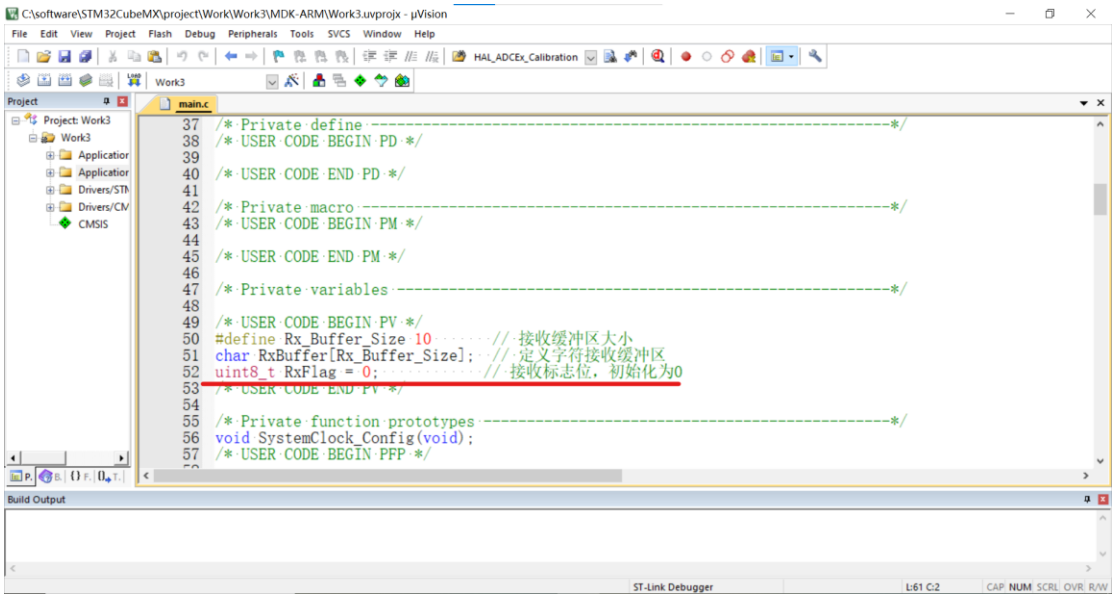




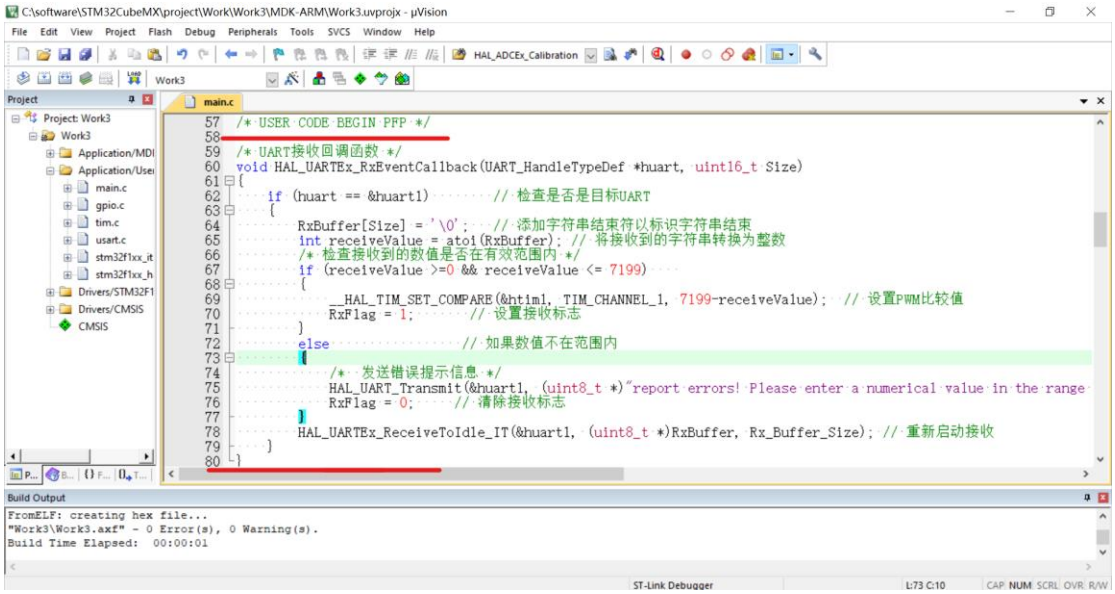
2、引用 C 库的头文件



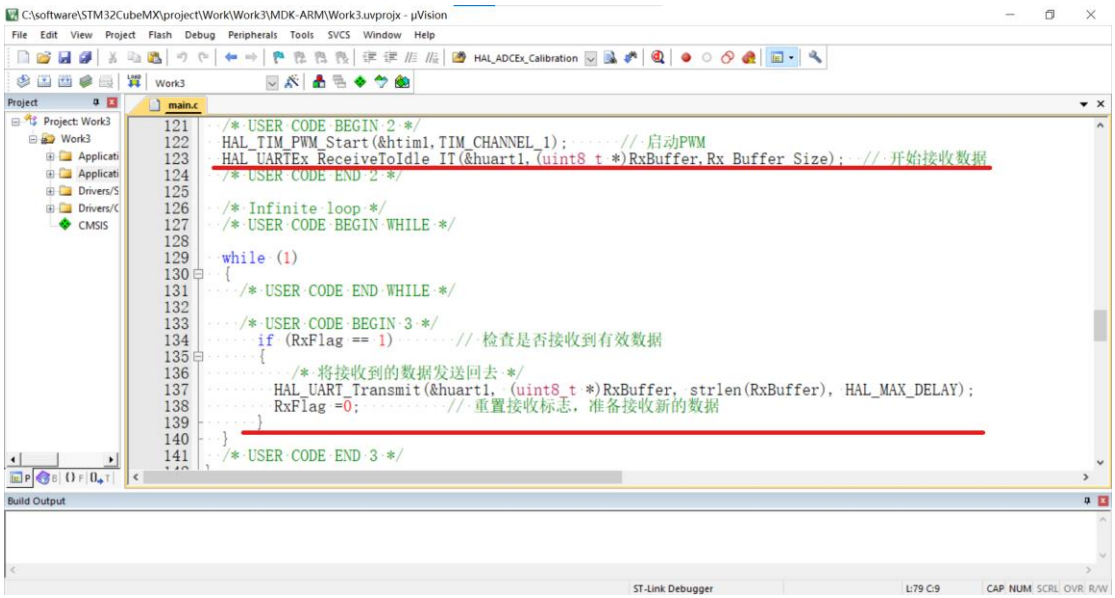
3、定义全局变量



4、重写中断回调函数



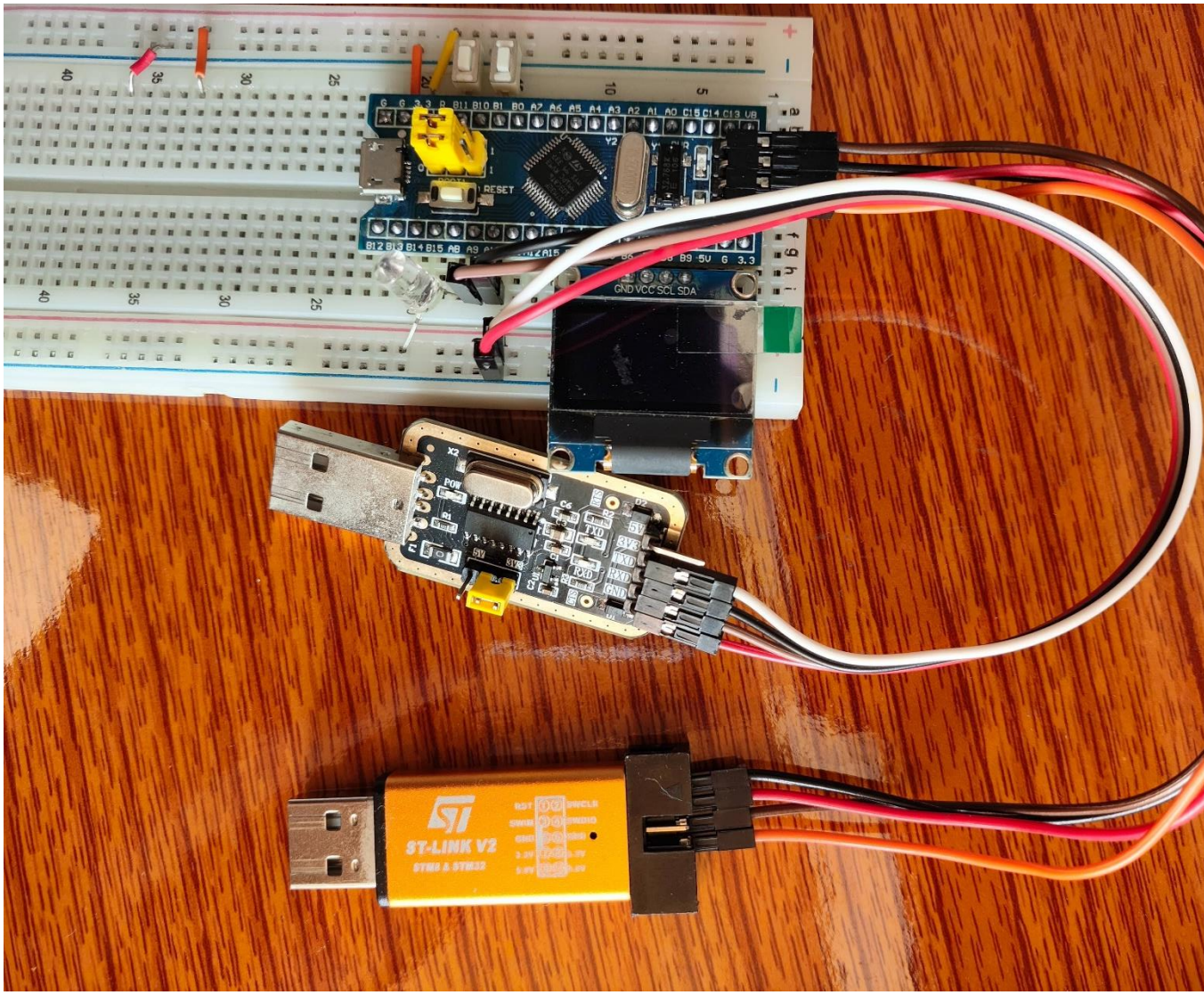
5、开启接收中断，打印接收到的数据发回串口软件



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

LED 灯长脚接 3.3V，短脚接 PA8



四、代码部分

```
/* Private includes ----------------------------------------------------------*/
/* USER CODE BEGIN Includes */
#include "string.h"                    //包含了字符串处理
#include "stdio.h"                     //包含了标准输入输出
#include "stdlib.h"                    //包含了标准库函数
/* USER CODE END Includes */
```

```
/* USER CODE BEGIN PV */
#define Rx_Buffer_Size 10        // 接收缓冲区大小
char RxBuffer[Rx_Buffer_Size];  // 定义字符接收缓冲区
uint8_t RxFlag = 0;             // 接收标志位，初始化为0
/* USER CODE END PV */
```

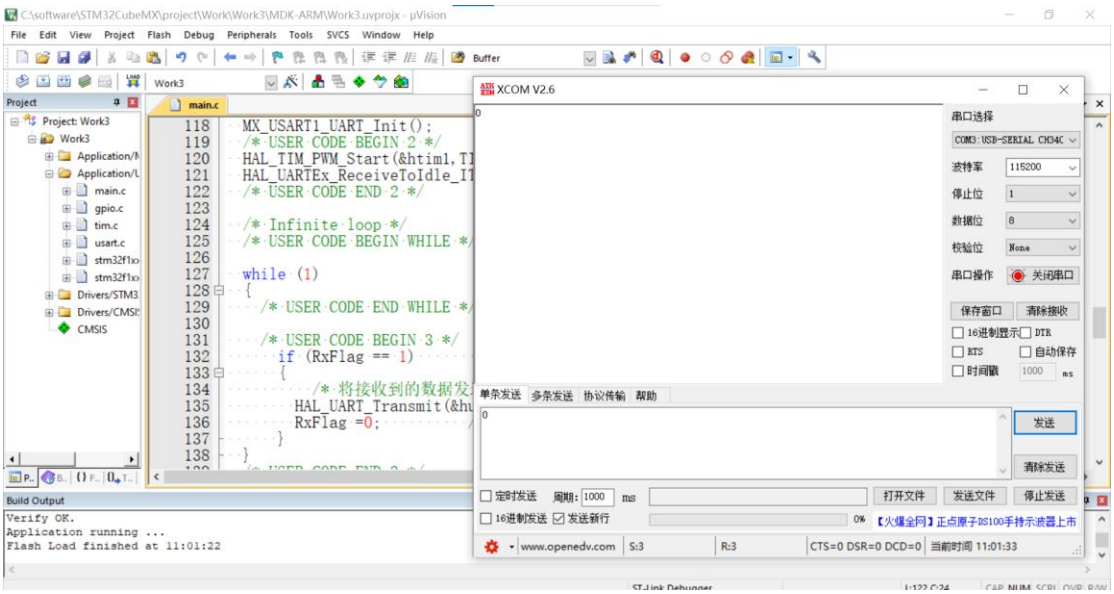
```
/* Private function prototypes -----------------------------------------------*/
void SystemClock_Config(void);
/* USER CODE BEGIN PFP */

/* UART 接收回调函数 */
void HAL_UARTEx_RxEventCallback(UART_HandleTypeDef *huart, uint16_t Size)
{
    if (huart == &huart1)        // 检查是否是目标 UART
    {
        RxBuffer[Size] = '\0';   // 添加字符串结束符以标识字符串结束
        int receiveValue = atoi(RxBuffer); // 将接收到的字符串转换为整数
        /* 检查接收到的数值是否在有效范围内 */
        if (receiveValue >=0 && receiveValue <= 7199)
        {
            __HAL_TIM_SET_COMPARE(&htim1, TIM_CHANNEL_1, 7199-receiveValue);  // 设置 PWM 比较值
            RxFlag = 1;       // 设置接收标志
        }
        else                  // 如果数值不在范围内
        {
            /*  发送错误提示信息 */
            HAL_UART_Transmit(&huart1, (uint8_t *)"report errors! Please enter a numerical value in the range of 0-7199\r\n",70,1000);
            RxFlag = 0;     // 清除接收标志
        }
        HAL_UARTEx_ReceiveToIdle_IT(&huart1, (uint8_t *)RxBuffer, Rx_Buffer_Size); // 重新启动接收
    }
}
/* USER CODE END PFP */
```

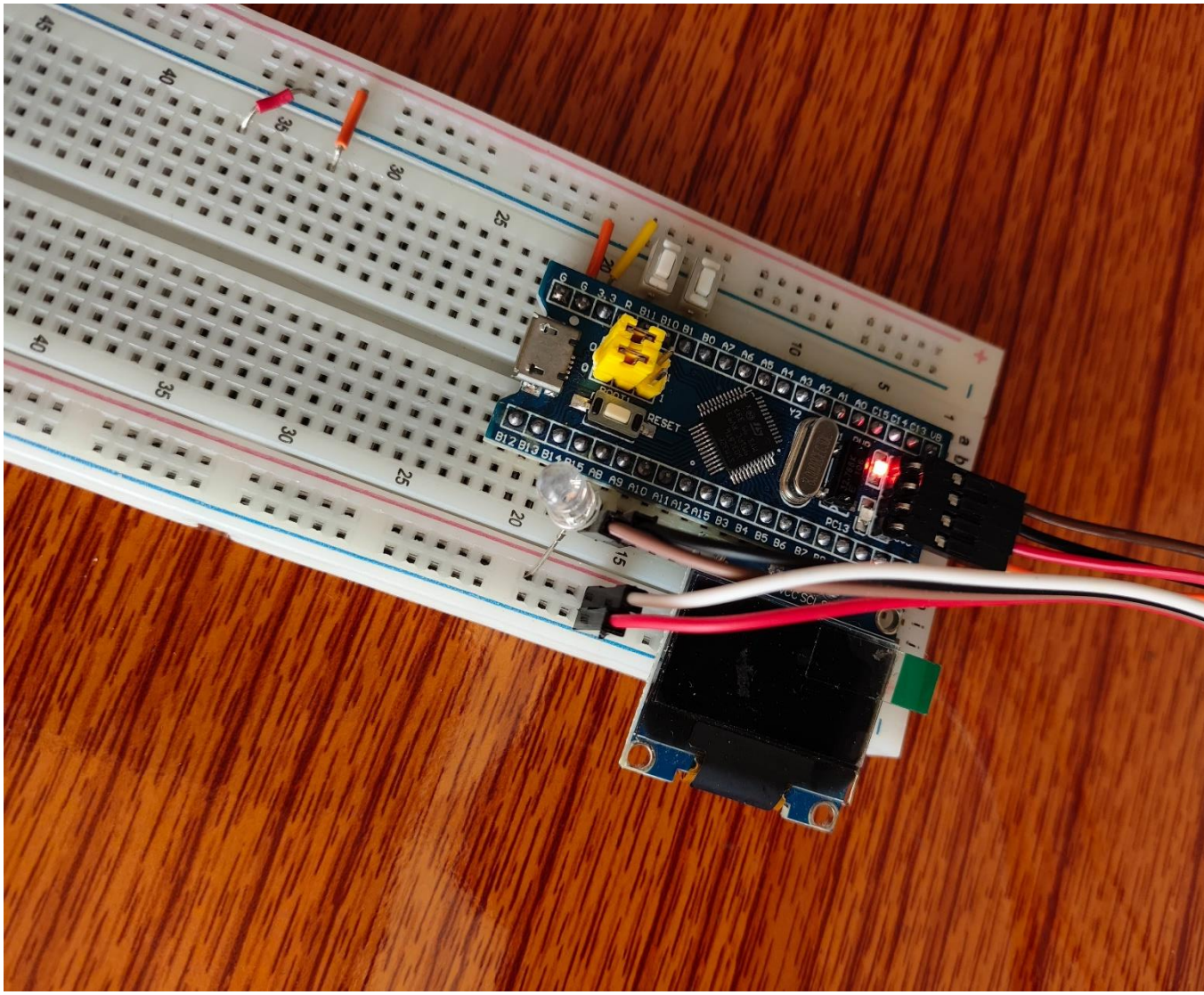
```
  /* USER CODE BEGIN 2 */
  HAL_TIM_PWM_Start(&htim1,TIM_CHANNEL_1);      // 启动 PWM
  HAL_UARTEx_ReceiveToIdle_IT(&huart1,(uint8_t *)RxBuffer,Rx_Buffer_Size); // 开始接收数据
  /* USER CODE END 2 */

  /* Infinite loop */
  /* USER CODE BEGIN WHILE */

  while (1)
  {
```

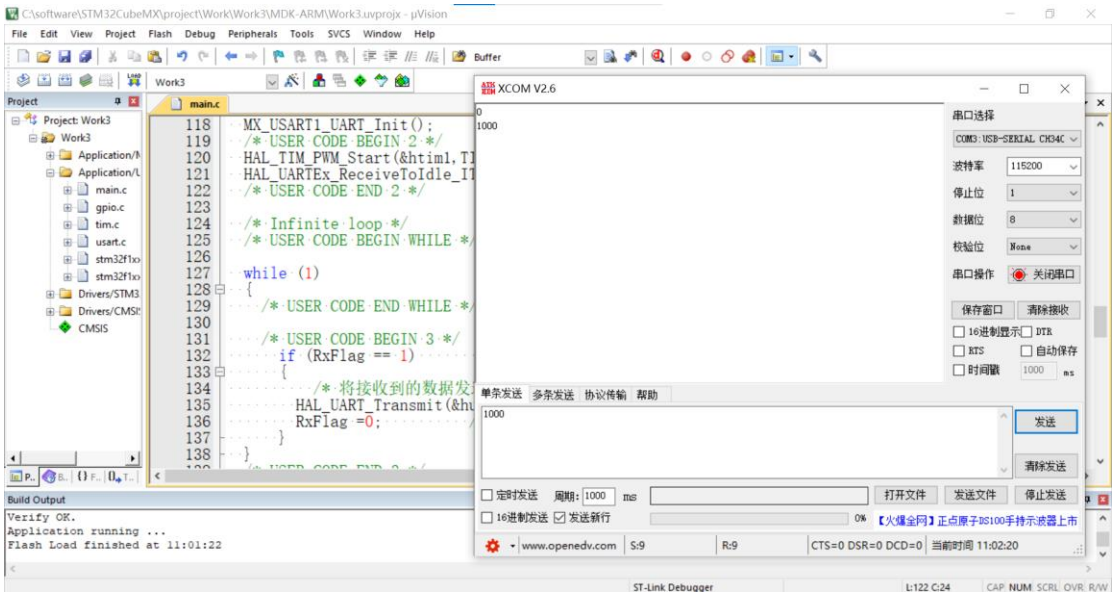
```
        /* USER CODE END WHILE */

        /* USER CODE BEGIN 3 */
          if (RxFlag == 1)       // 检查是否接收到有效数据
          {
              /* 将接收到的数据发送回去 */
            HAL_UART_Transmit(&huart1, (uint8_t *)RxBuffer, strlen(RxBuffer), HAL_MAX_DELAY);
            RxFlag =0;           // 重置接收标志，准备接收新的数据
          }
      }
      /* USER CODE END 3 */
```

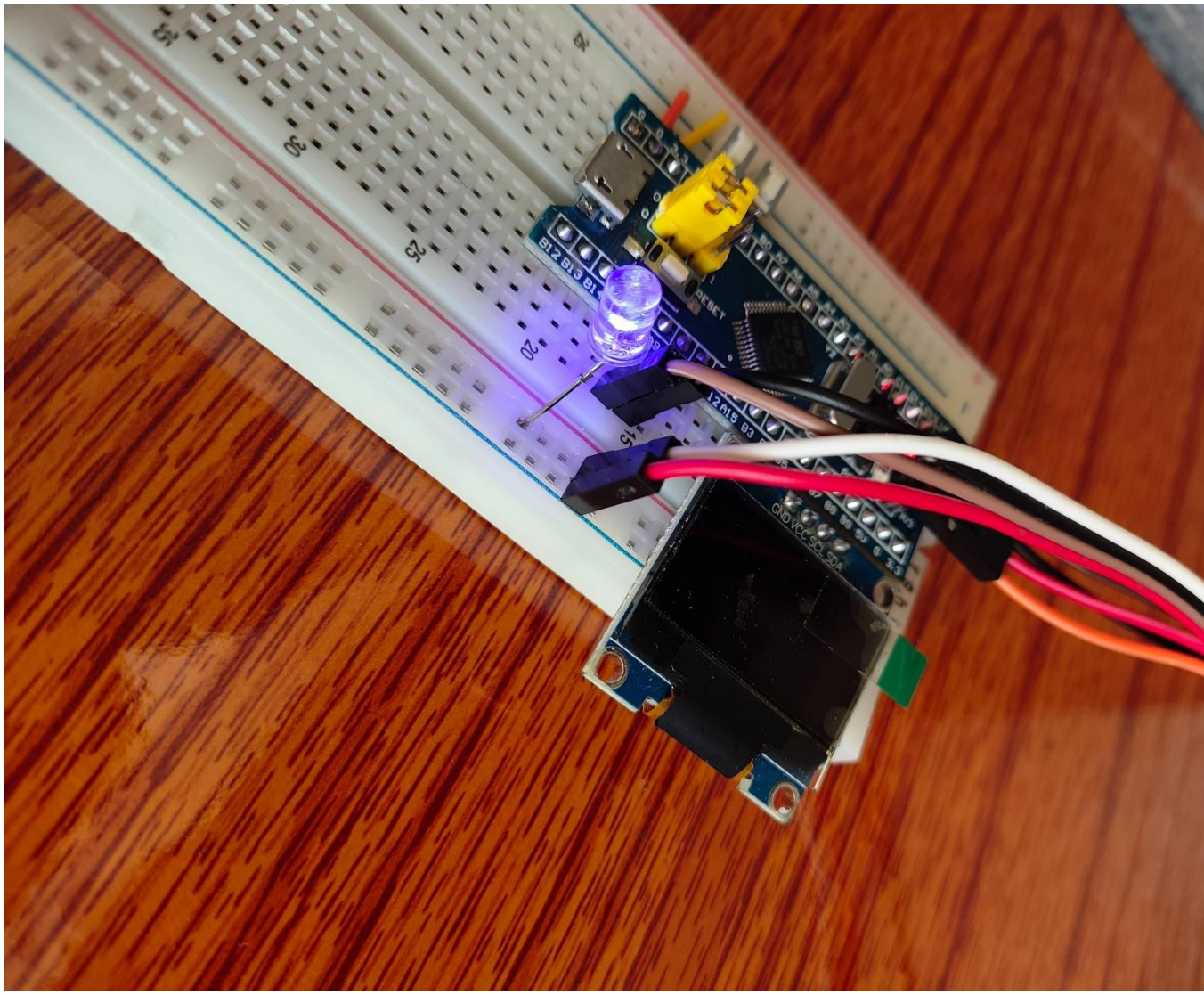
五、实现效果

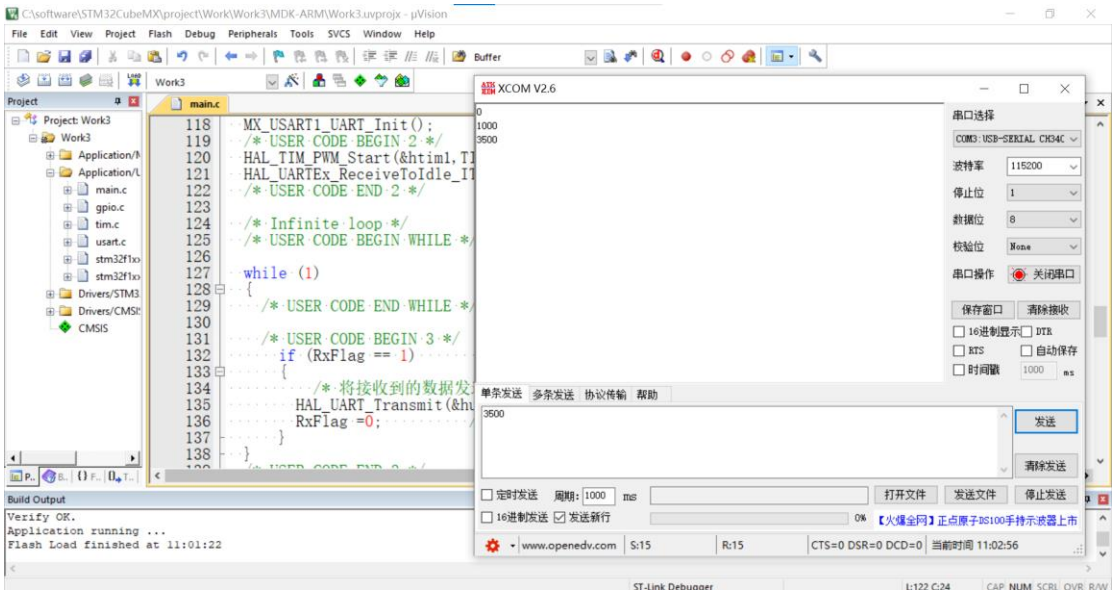
当输入 0 时，占空比为 0%
输入 1000，占空比为（1000/7199）* 100% = 13.9%
输入 3500，占空比为（3500/7199）* 100% = 48.6%
输入 7199，占空比为（7199/7199）* 100% = 100%
当超过 7199 数值，会发送报错信息回串口

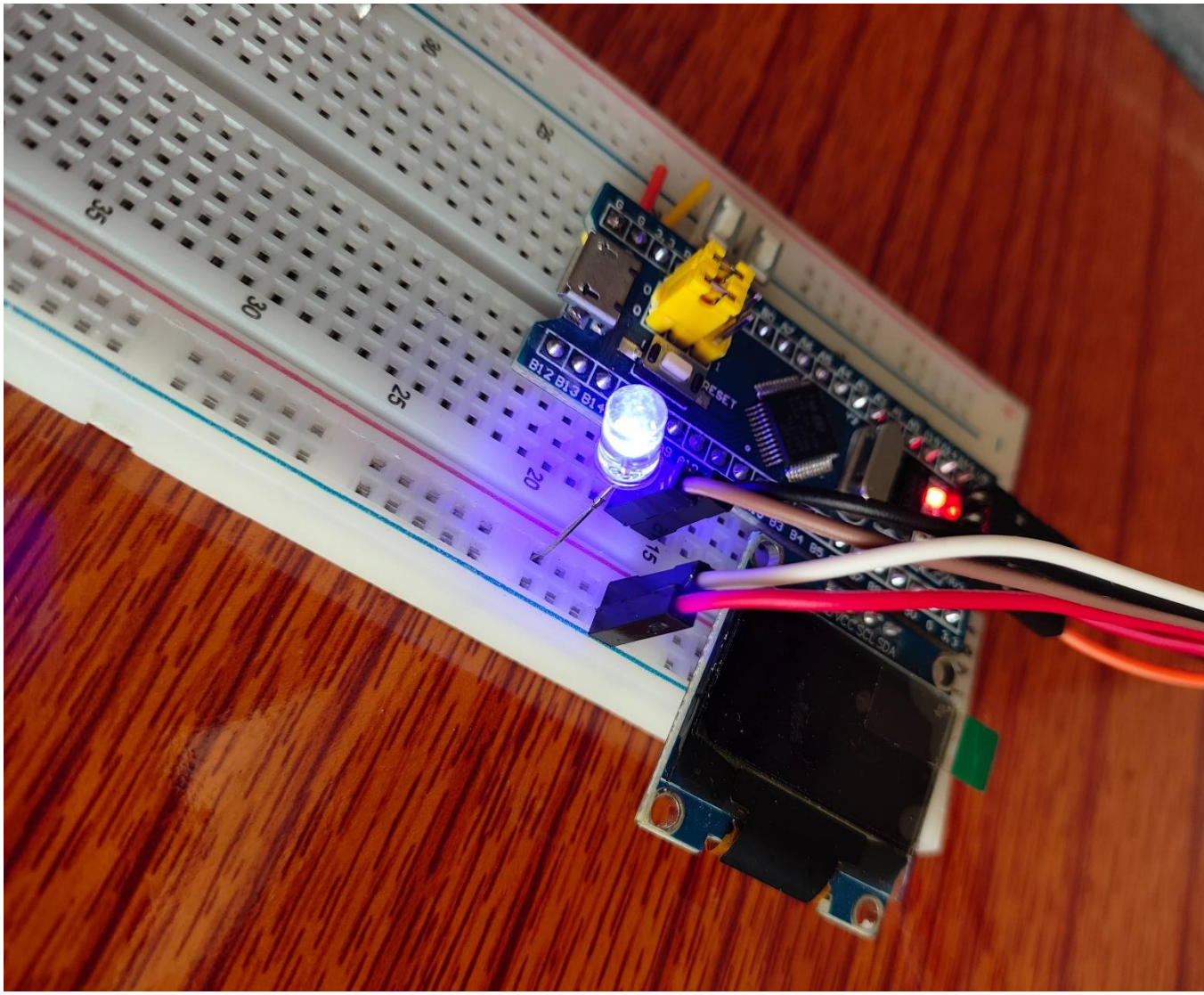


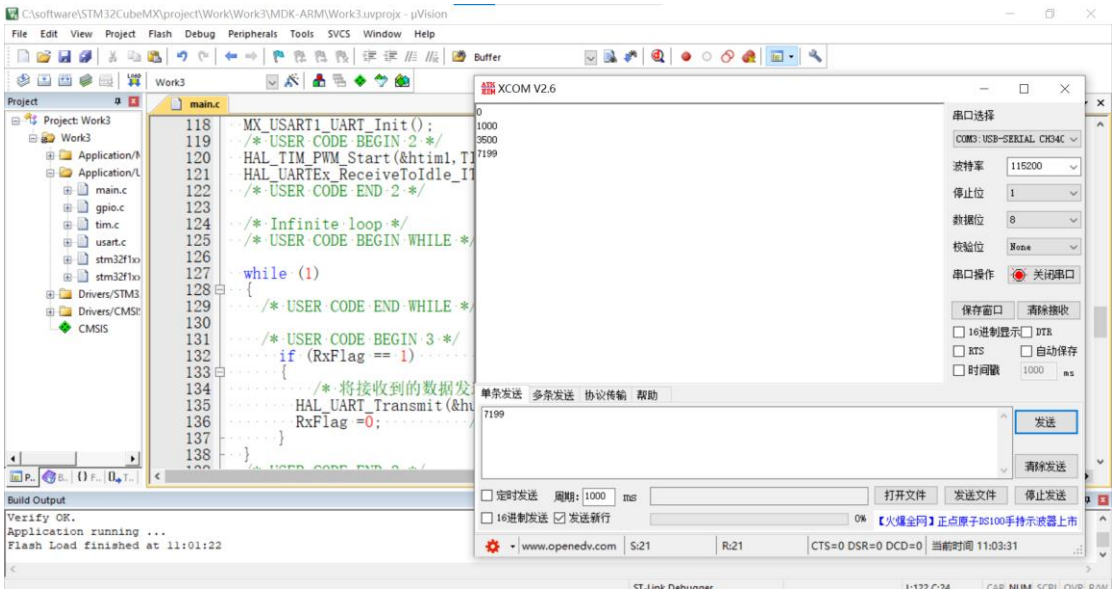


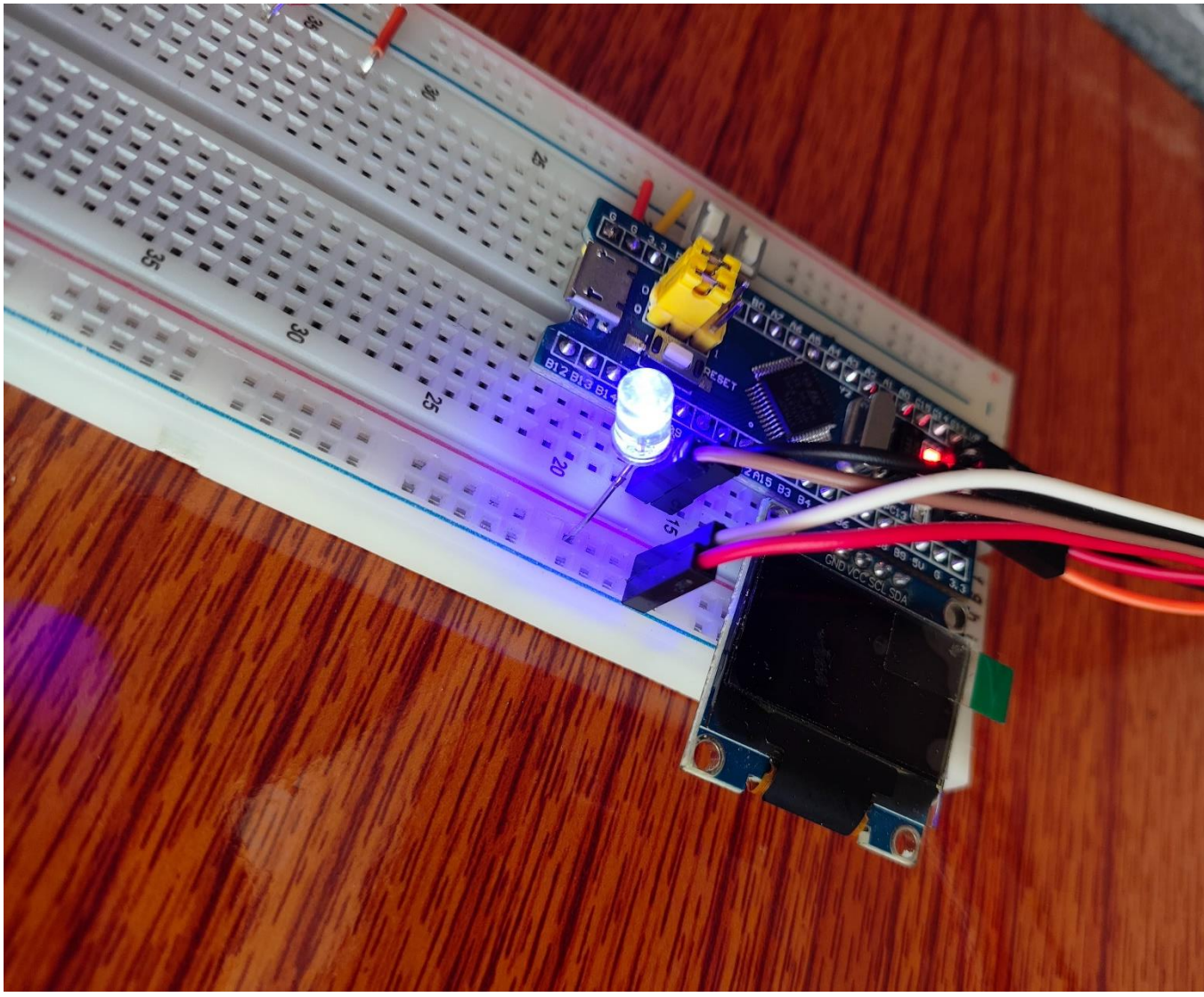

C:\software\STM32CubeMX\project\Work\Work3\MDK-ARM\Work3.uvprojx - µVision
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Project: Work3
Work3
main.c
gpio.c
tim.c
usart.c
CMSIS
118 MX_USART1_UART_Init();
119 /* USER CODE BEGIN 2 */
120 HAL_TIM_PWM_Start(&htim1,TI
121 HAL_UARTEx_ReceiveToIdle_IT
122 /* USER CODE END 2 */
123
124 /* Infinite loop */
125 /* USER CODE BEGIN WHILE */
126
127 while (1)
128 {
129 /* USER CODE END WHILE */
130
131 /* USER CODE BEGIN 3 */
132 if (RxFlag == 1)
133 {
134 /*将接收到的数据发
135 HAL_UART_Transmit(&hu
136 RxFlag =0;
137 }
138 }
Build Output
Verify OK.
Application running ...
Flash Load finished at 11:01:22
XCOM V2.6
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
【火爆全网】正点原子DS100手持示波器上市
www.openedv.com S:9 R:9 CTS=0 DSR=0 DCD=0 当前时间 11:02:20
ST-Link Debugger

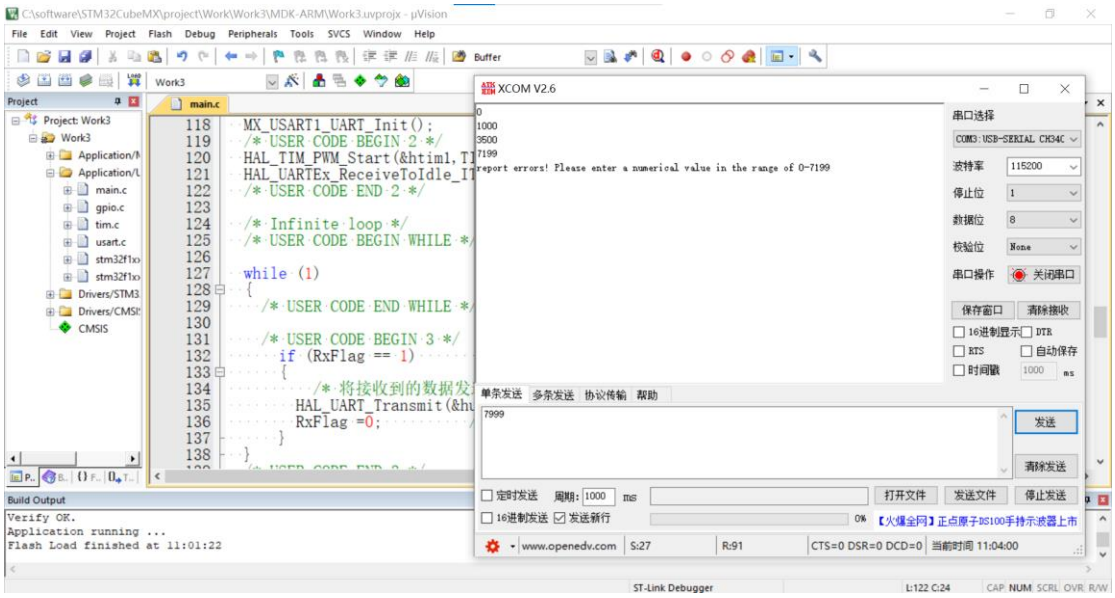



C:\software\STM32CubeMX\project\Work\Work3\MDK-ARM\Work3.uvprojx - µVision
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Project: Work3
Work3
main.c
MX_USART1_UART_Init();
/* USER CODE BEGIN 2 */
HAL_TIM_PWM_Start(&htim1,TI
HAL_UARTEx_ReceiveToIdle_IT
/* USER CODE END 2 */
/* Infinite loop */
/* USER CODE BEGIN WHILE */
while (1)
{
/* USER CODE END WHILE */
/* USER CODE BEGIN 3 */
if (RxFlag == 1)
{
/*将接收到的数据发
HAL_UART_Transmit(&hu
RxFlag =0;
}
}
Build Output
Verify OK.
Application running ...
Flash Load finished at 11:01:22
XCOM V2.6
0
1000
3500
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
3500
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
www.openedv.com S:15 R:15 CTS=0 DSR=0 DCD=0 当前时间 11:02:56
ST-Link Debugger




C:\software\STM32CubeMX\project\Work\Work3\MDK-ARM\Work3.uvprojx - µVision
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Project: Work3
Work3
main.c
gpio.c
tim.c
usart.c
CMSIS
main.c
118 MX_USART1_UART_Init();
119 /* USER CODE BEGIN 2 */
120 HAL_TIM_PWM_Start(&htim1,TI
121 HAL_UARTEx_ReceiveToIdle_IT
122 /* USER CODE END 2 */
123
124 /* Infinite loop */
125 /* USER CODE BEGIN WHILE */
126
127 while (1)
128 {
129 /* USER CODE END WHILE */
130
131 /* USER CODE BEGIN 3 */
132 if (RxFlag == 1)
133 {
134 /* 将接收到的数据发
135 HAL_UART_Transmit(&hu
136 RxFlag =0;
137 }
138 }
XCOM V2.6
0
1000
3500
7199
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
7199
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
www.openedv.com S:21 R:21 CTS=0 DSR=0 DCD=0 当前时间 11:03:31
Build Output
Verify OK.
Application running ...
Flash Load finished at 11:01:22


RESET
GND VCC SCL SDA


C:\software\STM32CubeMX\project\Work\Work3\MDK-ARM\Work3.uvprojx - µVision
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Project: Work3
Work3
main.c
gpio.c
tim.c
usart.c
CMSIS
118 MX_USART1_UART_Init();
119 /* USER CODE BEGIN 2 */
120 HAL_TIM_PWM_Start(&htim1,TI
121 HAL_UARTEx_ReceiveToIdle_IT
122 /* USER CODE END 2 */
123
124 /* Infinite loop */
125 /* USER CODE BEGIN WHILE */
126
127 while (1)
128 {
129 /* USER CODE END WHILE */
130
131 /* USER CODE BEGIN 3 */
132 if (RxFlag == 1)
133 {
134 /*将接收到的数据发
135 HAL_UART_Transmit(&hu
136 RxFlag =0;
137 }
138 }
Build Output
Verify OK.
Application running ...
Flash Load finished at 11:01:22
XCOM V2.6
0
1000
3500
7199
report errors! Please enter a numerical value in the range of 0-7199
串口选择
COM3:USB-SERIAL CH340
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
7999
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
【火爆全网】正点原子DS100手持示波器上市
www.openedv.com S:27 R:91 CTS=0 DSR=0 DCD=0 当前时间 11:04:00
ST-Link Debugger
L:122 C:24