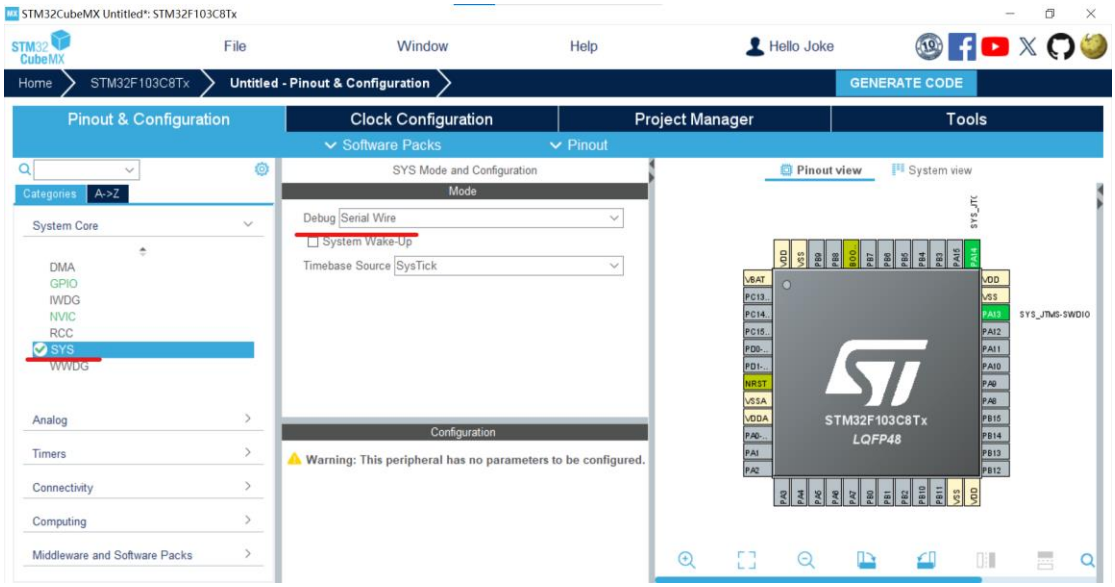
任务要求：
使用 ADC 及定时器 TIM1,定时器 TIM1 使用 PA8.TIM 的 Prescalar 设置为 0，ARR 设置为 7199.使用 HAL_TIM_SETCompare()
需要的器件有电位器，杜邦线
当 ADC 读取到电压值时，将电压值转换为 pwm 的占空比，例如当读取的数据为 3.3v（4095）时，对应的是占空比 7199
当读取的数值为 1.65v（2047）时，对应的占空比为 3599
当读取的数据为 0v 时，对应的占空比为 0

使用到的外设：**GPIO、USART1 串口、TIM1、ADC1、EXTI 中断**

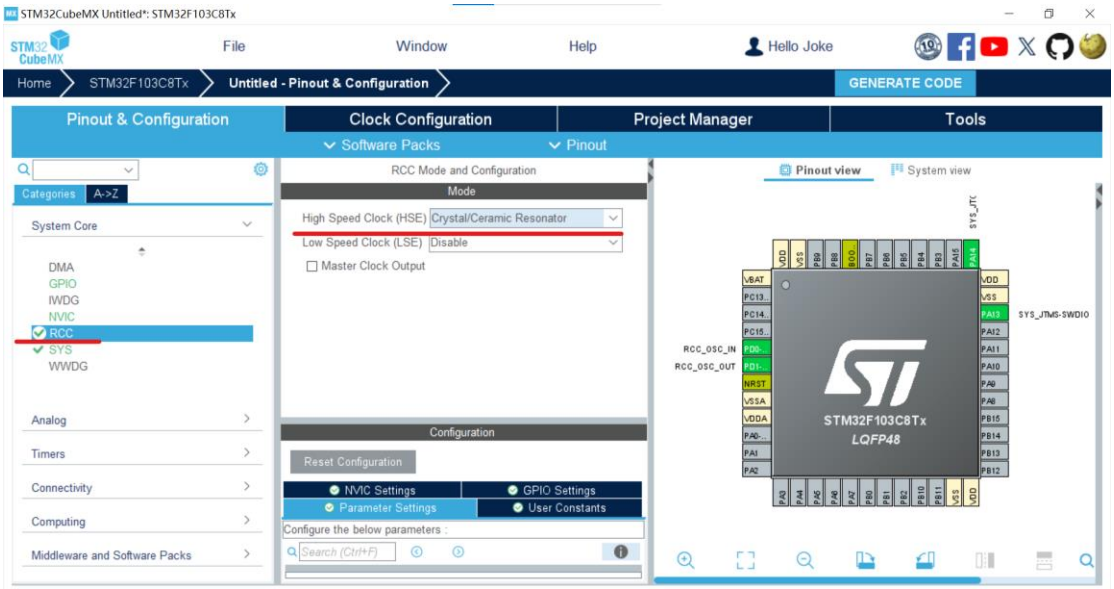
具体说明：电位器旋转，输出对应的模拟量，转换为对应电压 0-3.3V，(转换后的电压值 /3.3)*7199 可以设置为占空比

一、STM32CubeMX 创建工程步骤

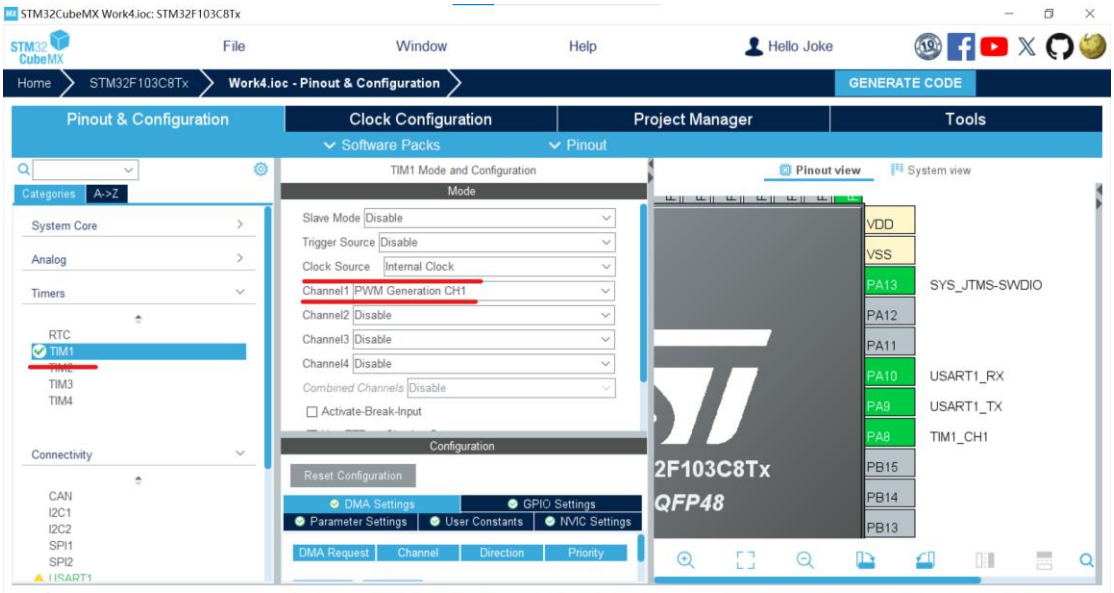
1、开启 SYS 系统时钟



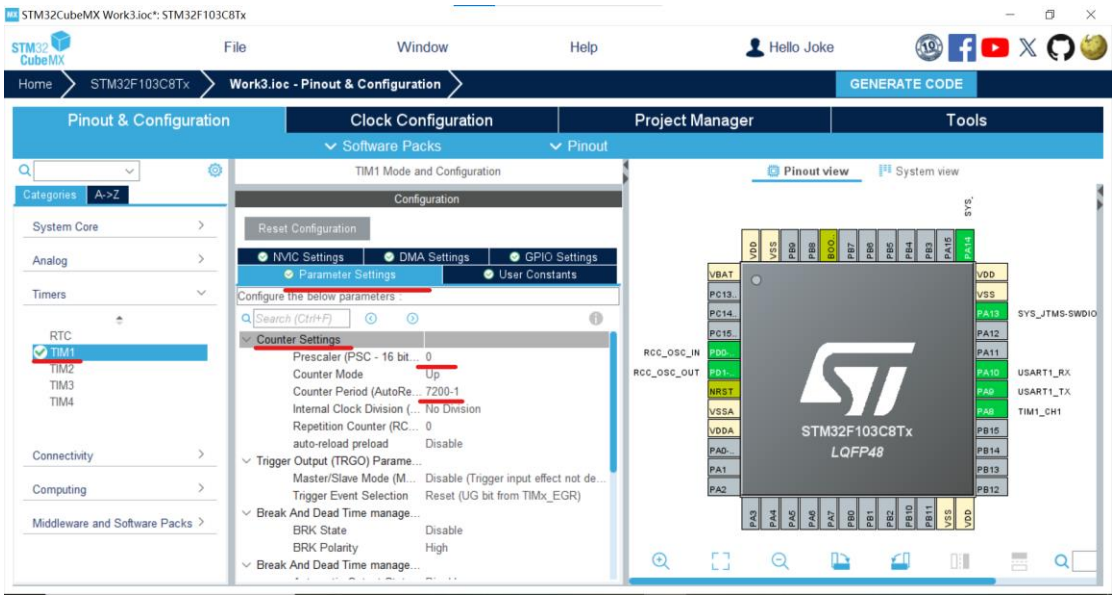
2、开启 RCC 晶振

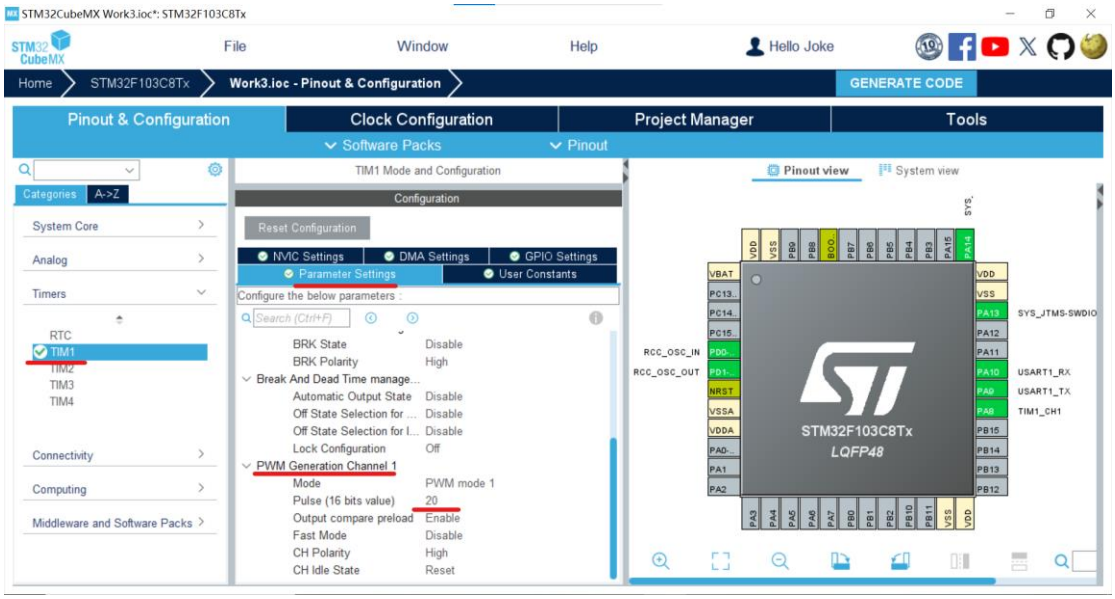


3、选择 TIM1，选择内部时钟，选择 PWM Generation CH1

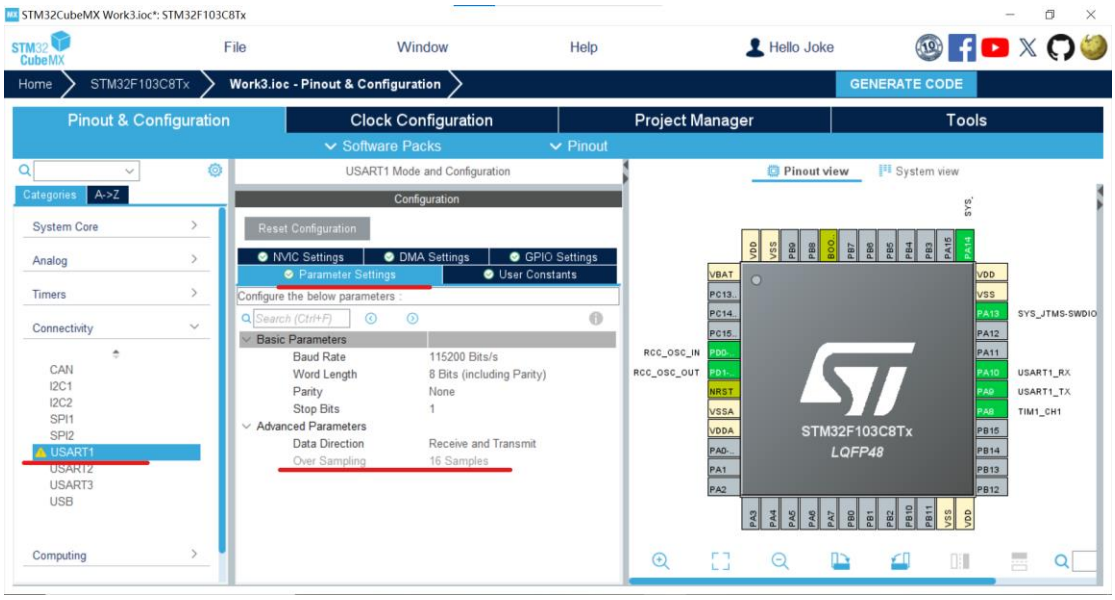


4、设置 PSC 预分频器为 0，自动重装寄存器 ARR 为 7199，再设置 Pulse 为 20，设置了 Pulse 才会有 PWM 波形

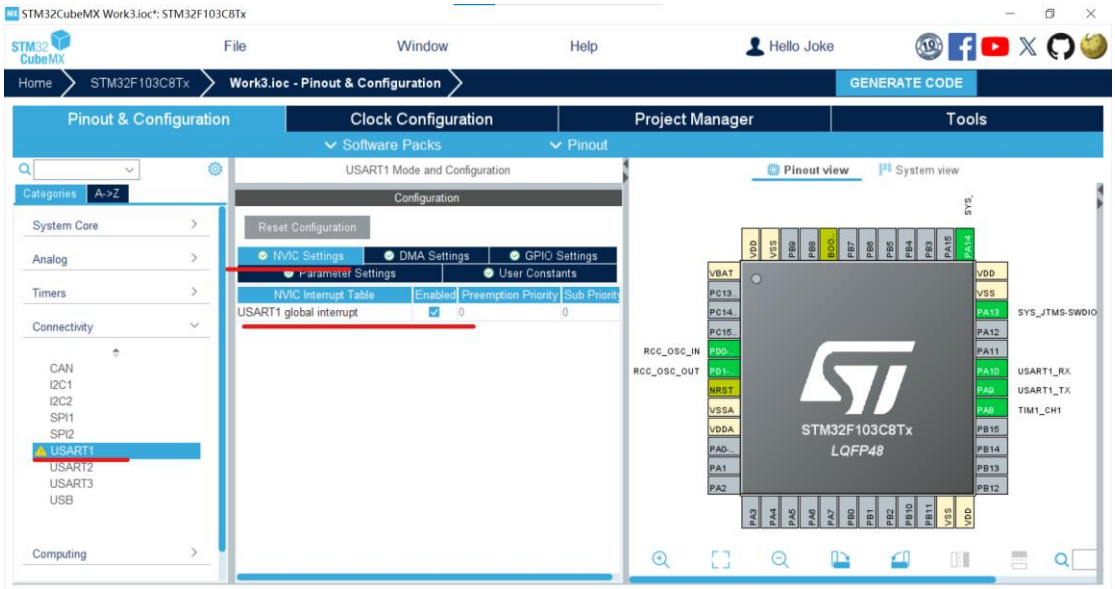




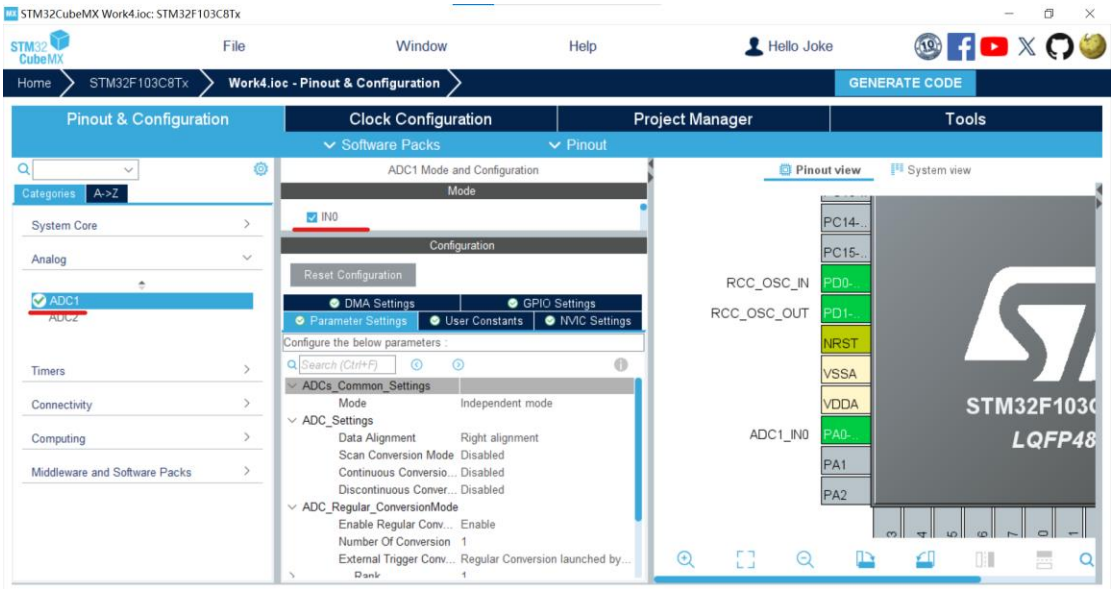
5、选择 USART1 作为串口通信，波特率设置为 115200， 8 位字节数据位，无校验位，1 位停止位，选择发送和接收模式



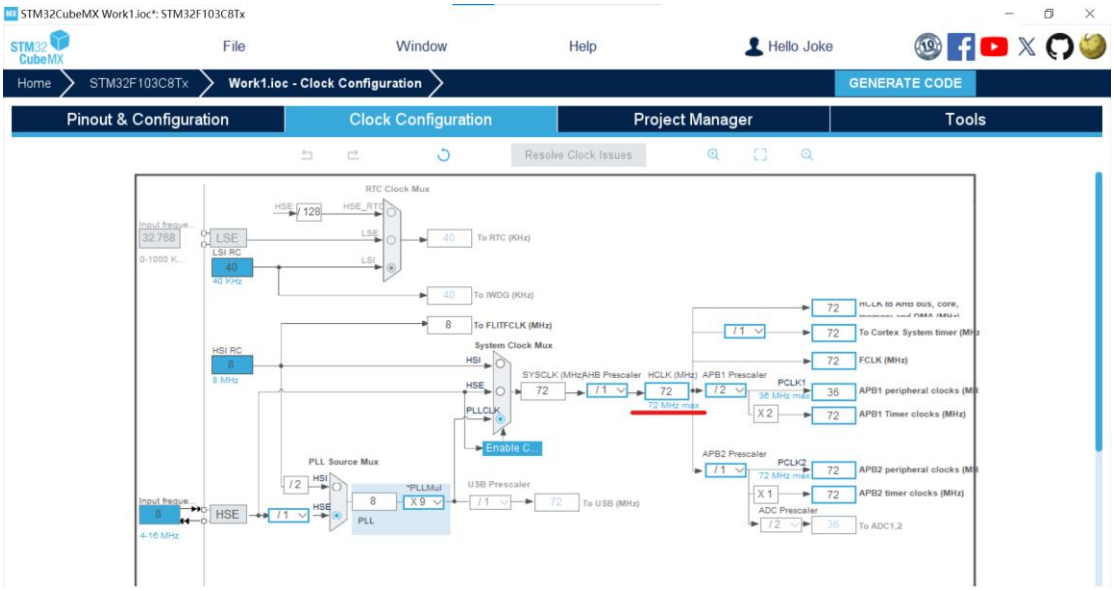
6、开启串口中断



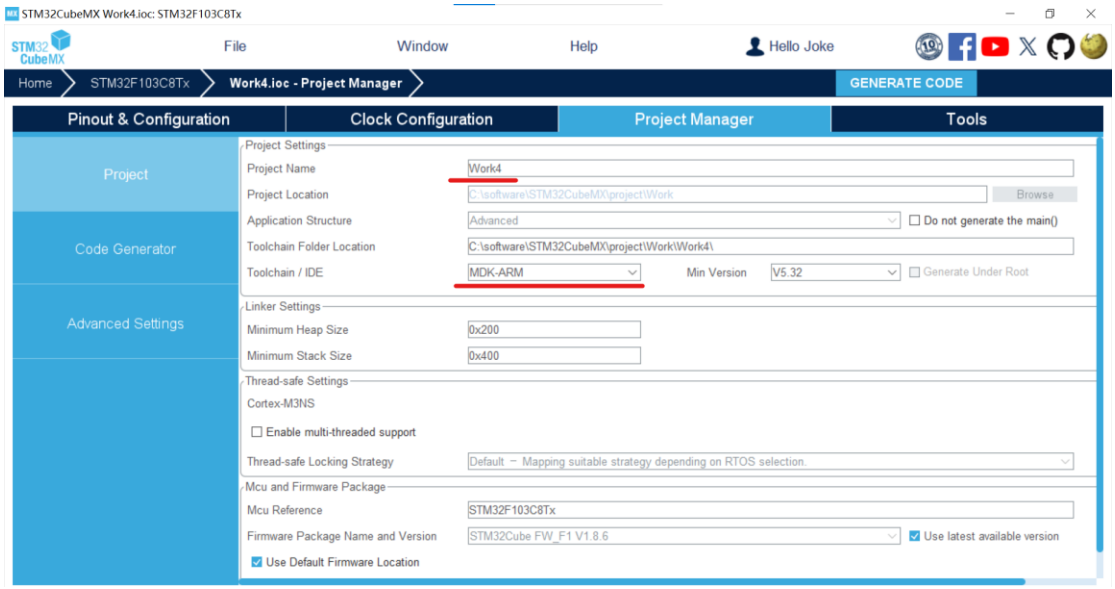
7、开启 ADC1 通道 1，选择默认的单次非扫描模式



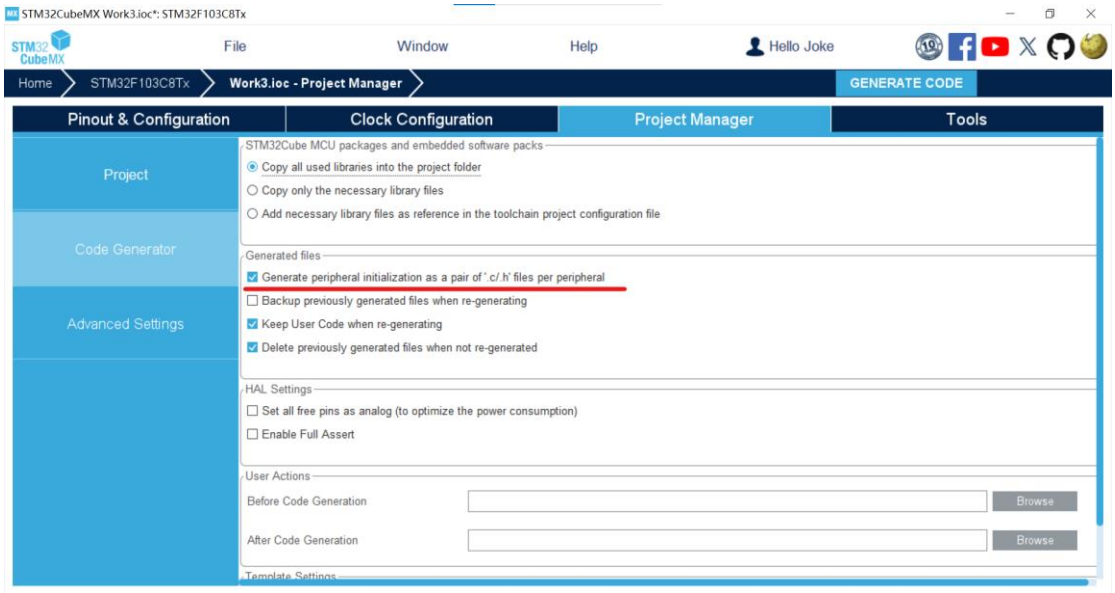
8、时钟树选择 72MHZ



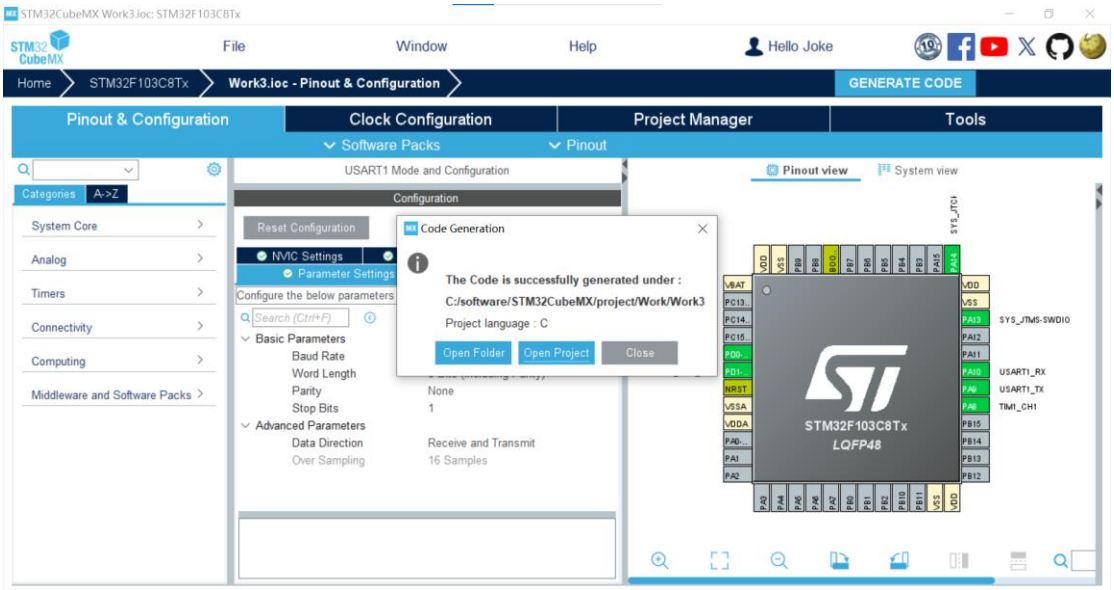
9、创建文件名，并存在对应的盘中，选择 MDK-ARM



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

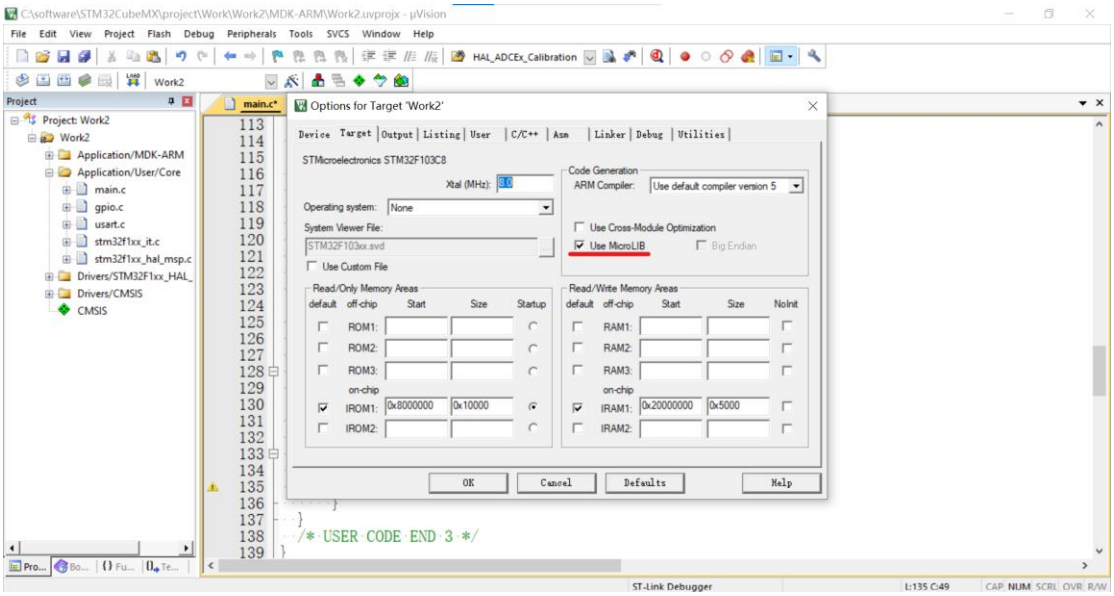


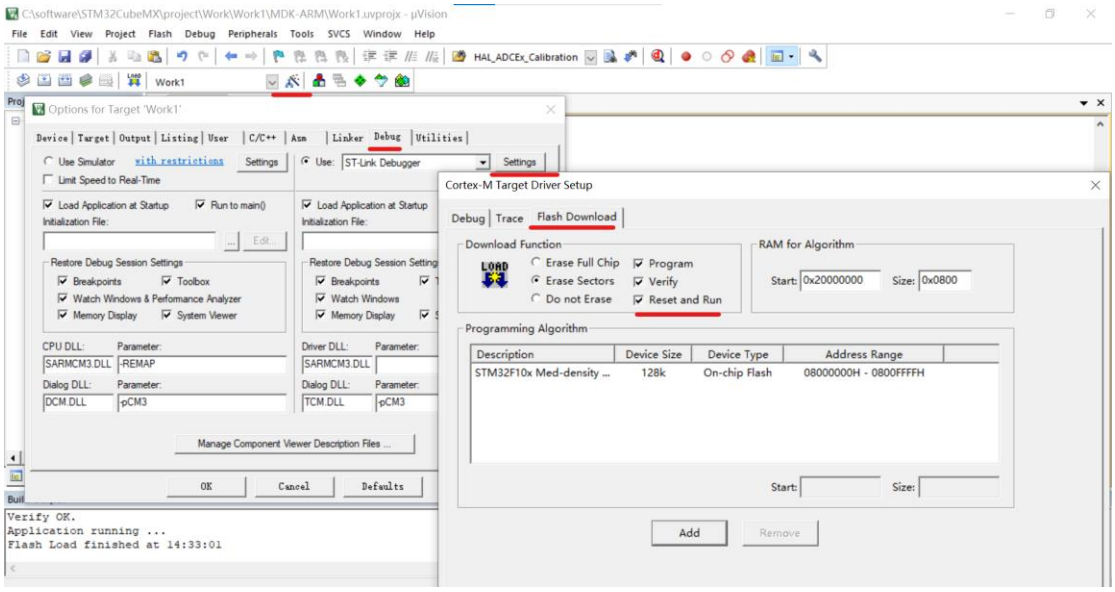
11、生成文件 GENERATE CODE，点击 CLOSE，并打开所生成的路径的 MDK 文件



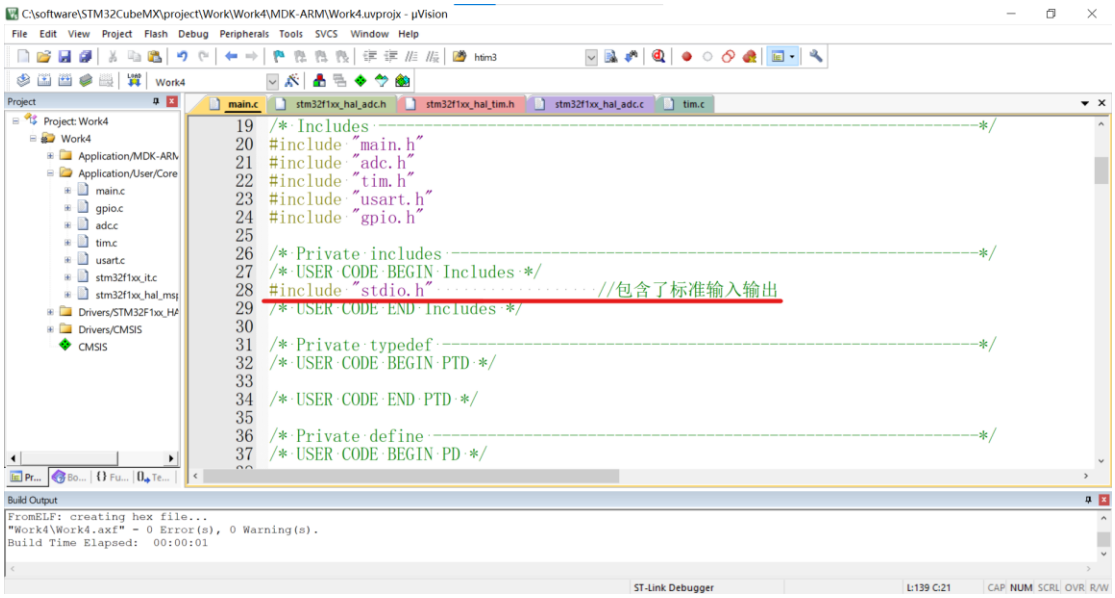
二、MDK 程序编写

1、设置复位下载选项和勾选 Use MicroLIB

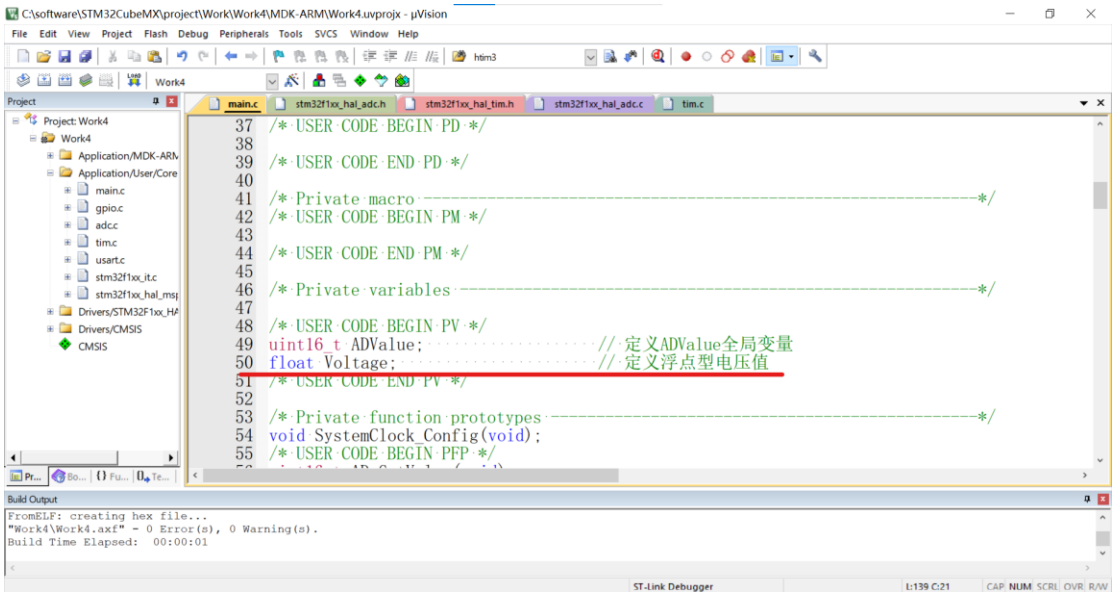




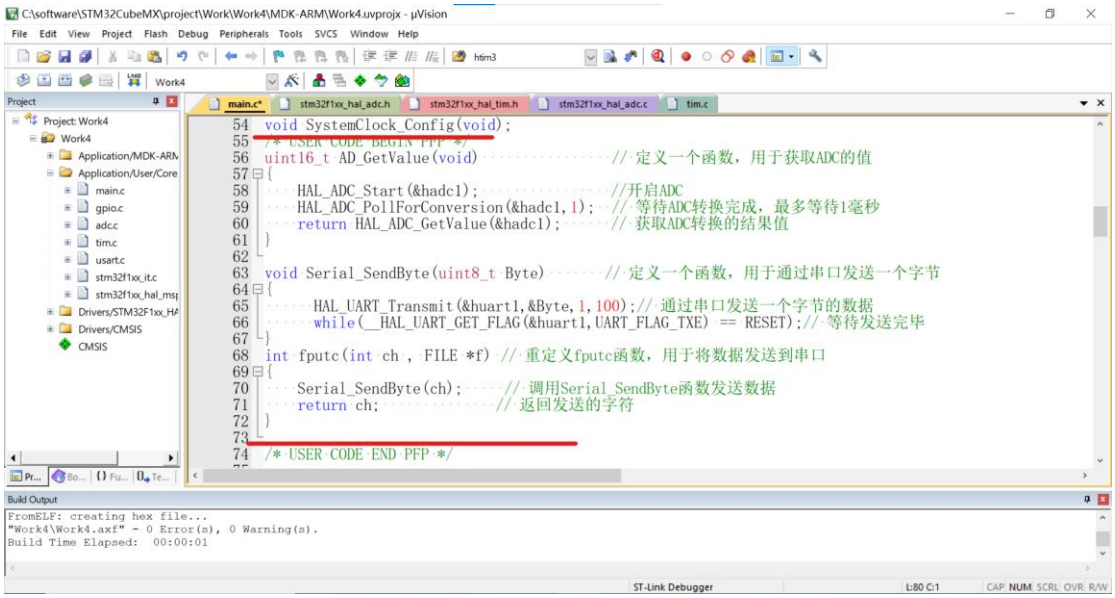
2、引用 C 库的头文件



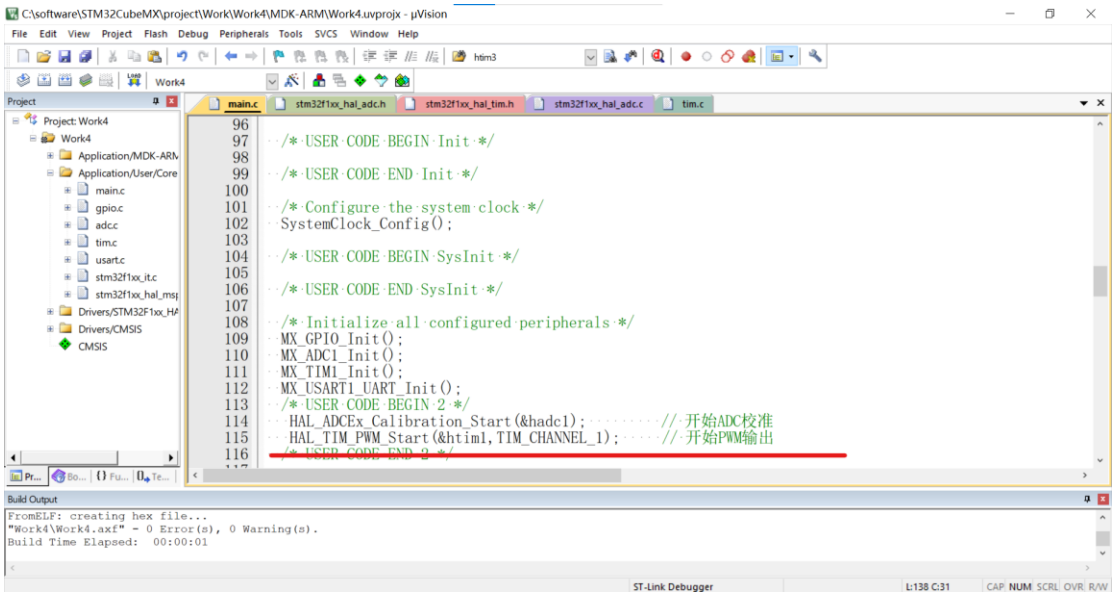
3、定义全局变量



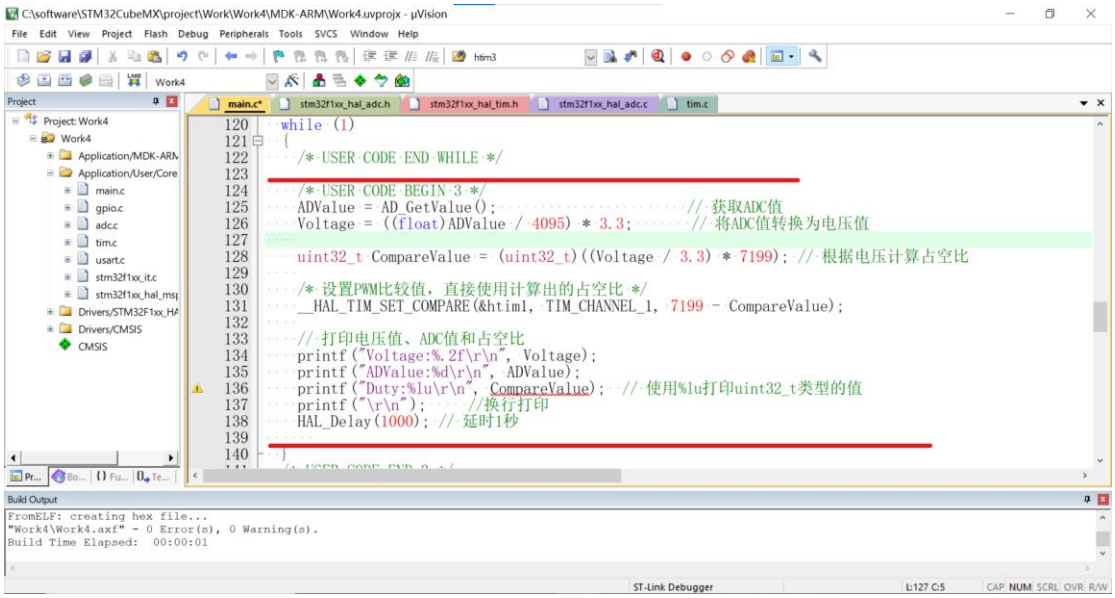
4、获取 ADC 的值并返回，fputc 重定向



5、初始化之后，ADC 校准和开启 PWM 输出



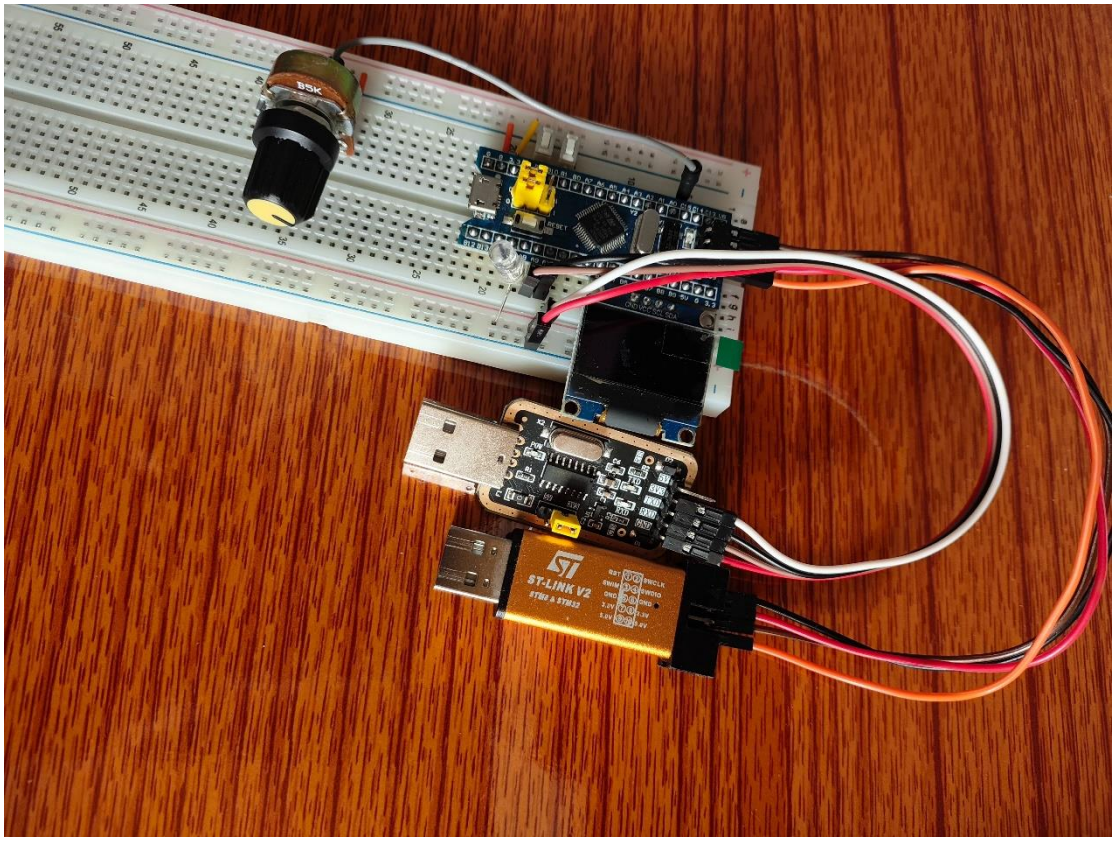
6、设置占空比，并将电压值、ADC 模拟值、占空比值在串口里打印出来



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

电位器中间线连接 PA0，左侧连接 GND，右侧连接 3.3V



四、代码部分

```
1.      /* Private includes ----------------------------------------------------------
--*/
2.      /* USER CODE BEGIN Includes */
3.      #include "stdio.h"                    //包含了标准输入输出
4.      /* USER CODE END Includes */
```

```
1.      /* USER CODE BEGIN PV */
2.      uint16_t ADValue;                     // 定义 ADValue 全局变量
3.      float Voltage;                        // 定义浮点型电压值
4.      /* USER CODE END PV */
```

```
1.      /* Private function prototypes -----------------------------------------------
--*/
2.      void SystemClock_Config(void);
3.      /* USER CODE BEGIN PFP */
4.      uint16_t AD_GetValue(void)                  // 定义一个函数，用于获取 ADC 的值
5.      {
6.          HAL_ADC_Start(&hadc1);                  //开启 ADC
7.          HAL_ADC_PollForConversion(&hadc1,1);    // 等待 ADC 转换完成，最多等待 1 毫秒
8.          return HAL_ADC_GetValue(&hadc1);        // 获取 ADC 转换的结果值
```

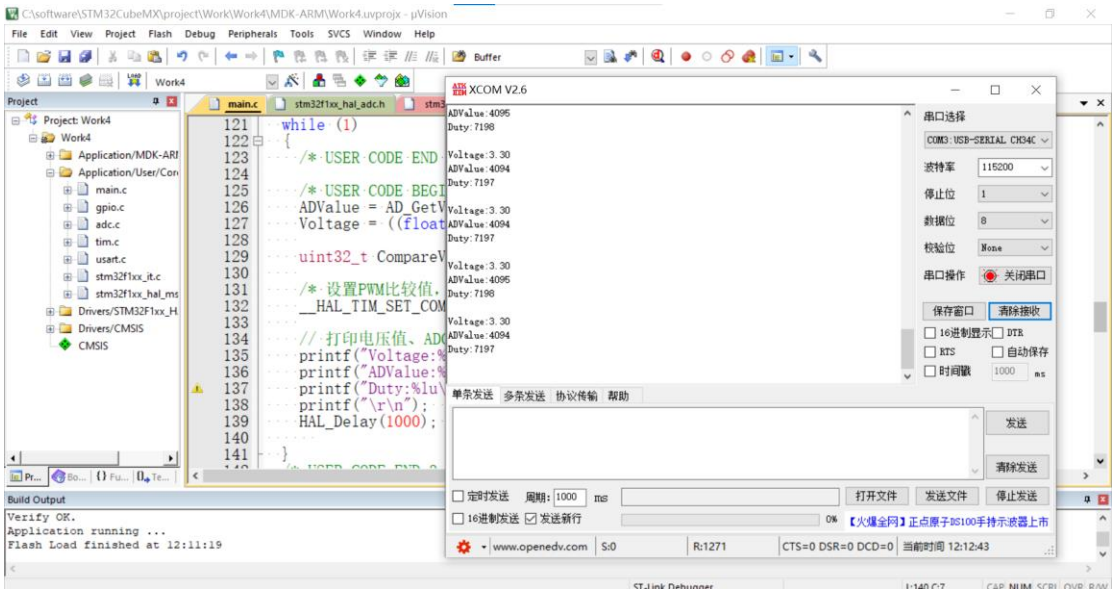
```
}

void Serial_SendByte(uint8_t Byte)       // 定义一个函数，用于通过串口发送一个字节
{
      HAL_UART_Transmit(&huart1,&Byte,1,100);// 通过串口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) // 重定义 fputc 函数，用于将数据发送到串口
{
    Serial_SendByte(ch);     // 调用 Serial_SendByte 函数发送数据
    return ch;              // 返回发送的字符
}

/* USER CODE END PFP */
```

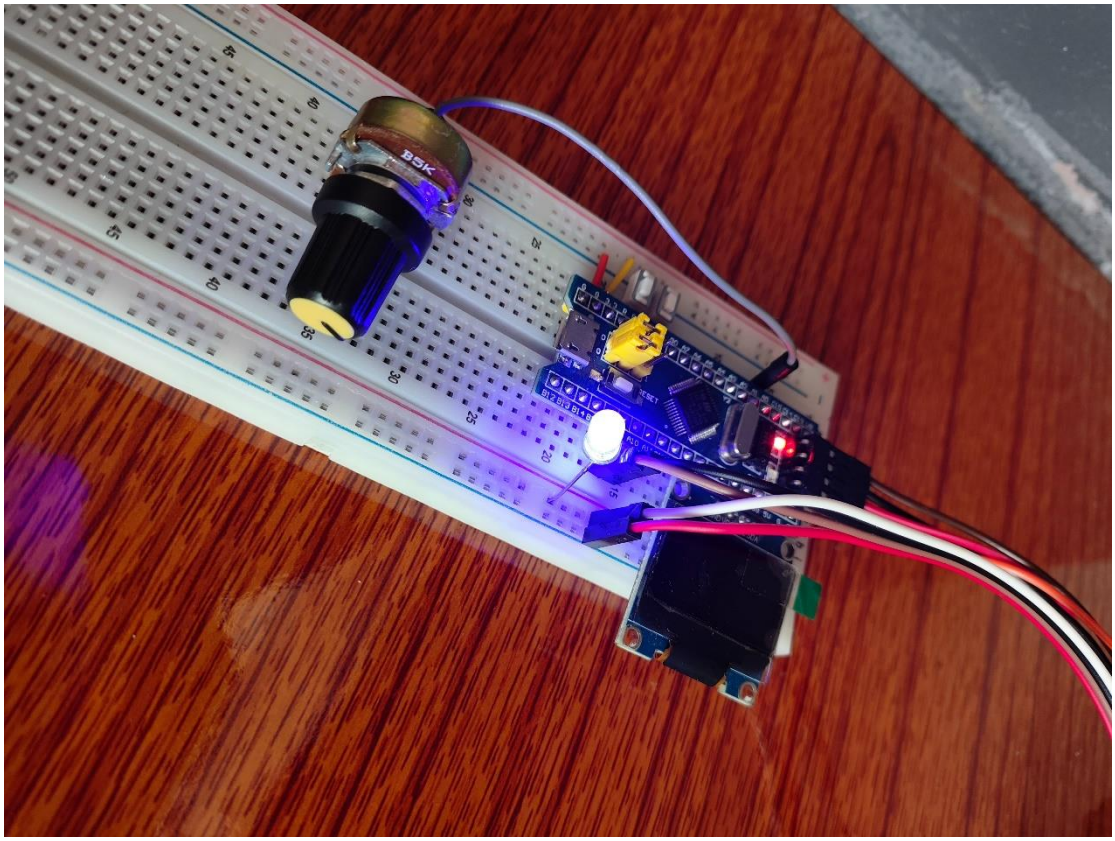
```
  /* USER CODE BEGIN 2 */
   HAL_ADCEx_Calibration_Start(&hadc1);          // 开始 ADC 校准
   HAL_TIM_PWM_Start(&htim1,TIM_CHANNEL_1);      // 开始 PWM 输出
  /* USER CODE END 2 */

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

    /* USER CODE BEGIN 3 */
    ADValue = AD_GetValue();                        // 获取 ADC 值
    Voltage = ((float)ADValue / 4095) * 3.3;        // 将 ADC 值转换为电压值

    uint32_t CompareValue = (uint32_t)((Voltage / 3.3) * 7199); // 根据电压计算占空比

    /* 设置 PWM 比较值，直接使用计算出的占空比 */
    __HAL_TIM_SET_COMPARE(&htim1, TIM_CHANNEL_1, 7199 - CompareValue);

    // 打印电压值、ADC 值和占空比
    printf("Voltage:%.2f\r\n", Voltage);
    printf("ADValue:%d\r\n", ADValue);
    printf("Duty:%lu\r\n", CompareValue);  // 使用%lu 打印 uint32_t 类型的值
    printf("\r\n");      //换行打印
```

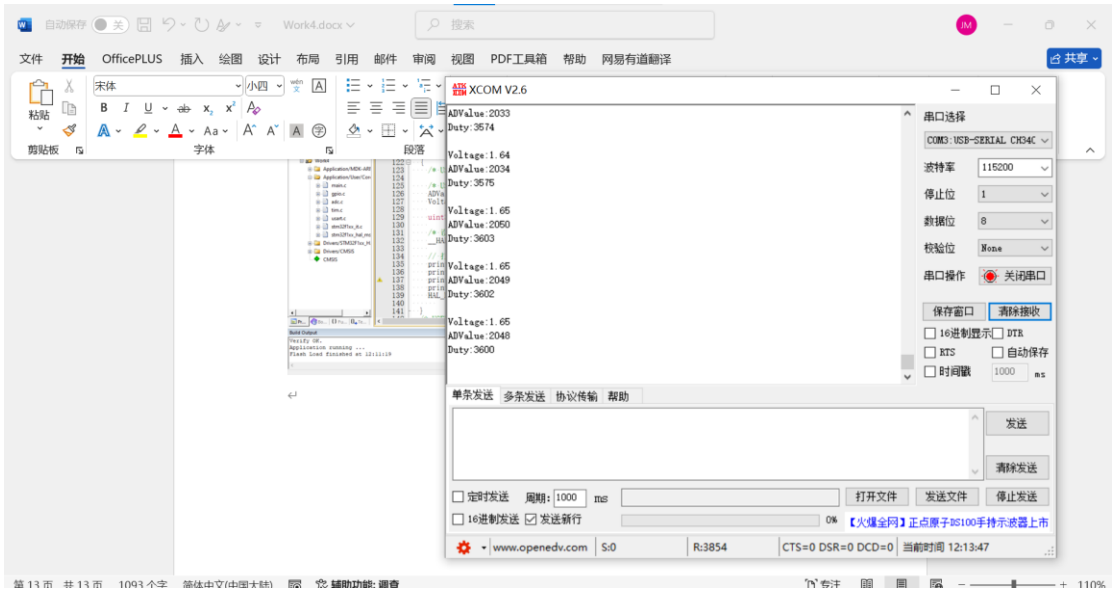
```
    //    HAL_Delay(1000); // 延时 1 秒，这里的延时 1 秒作用，不用串口打印速度太快

      }
      /* USER CODE END 3 */
```

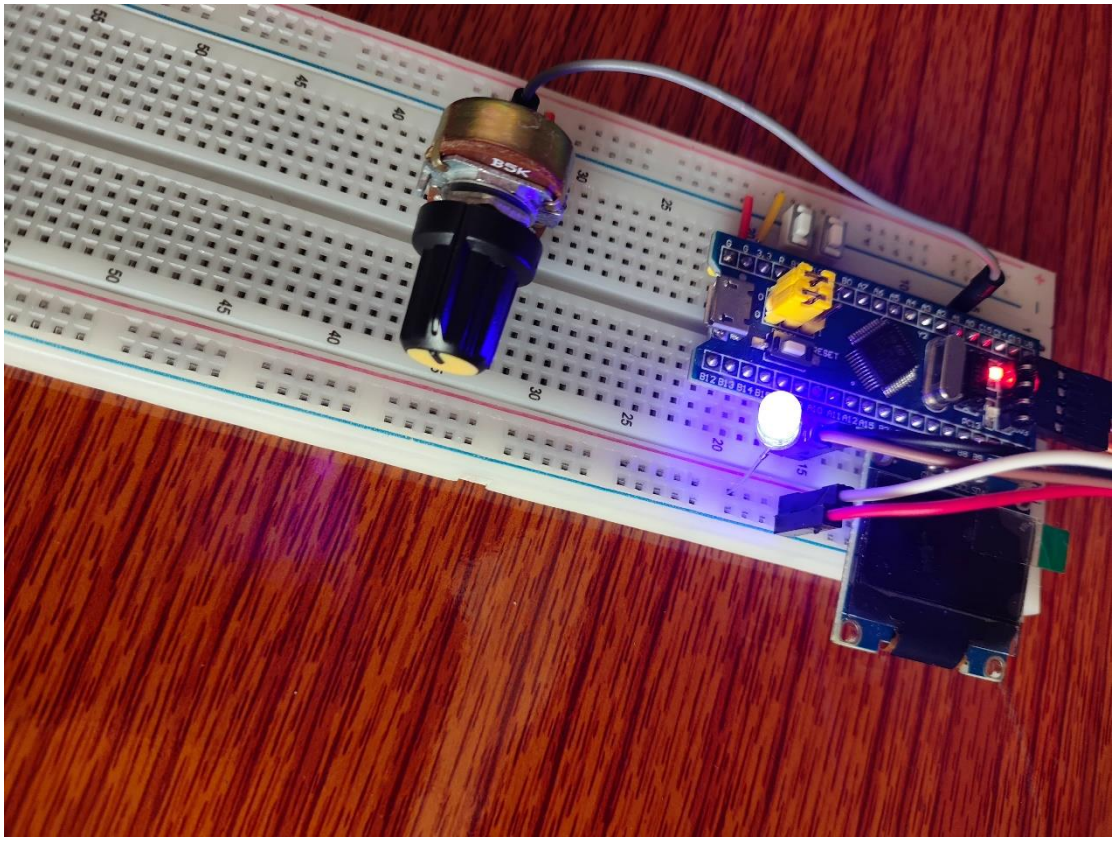
五、实现效果

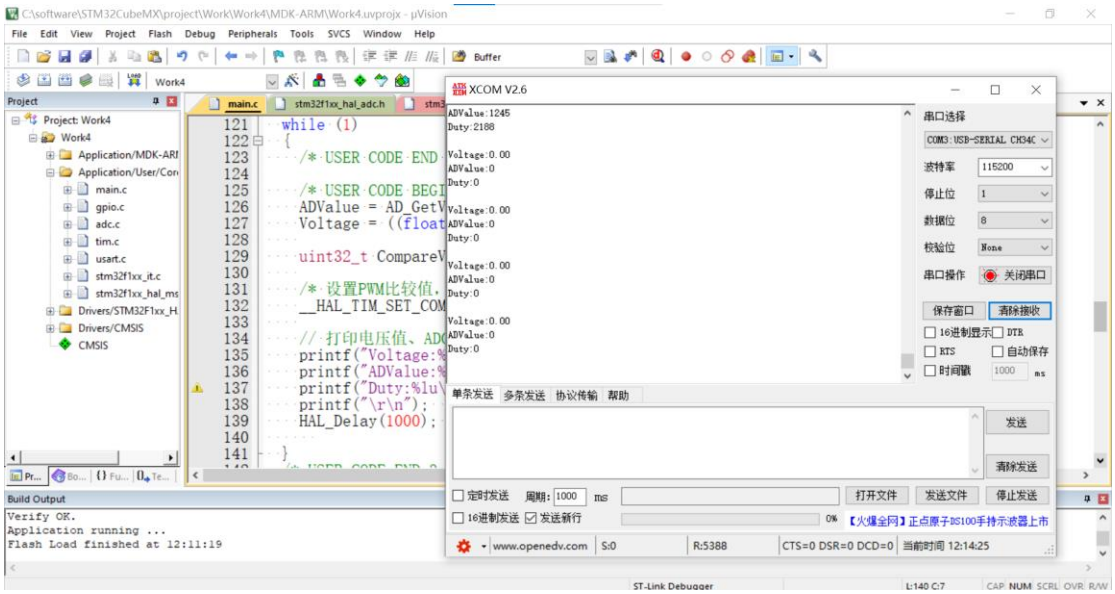
旋转电位器最大值 7199，设置的占空比 100%，并通过串口打印值
旋转电位器中间值 3599，设置的占空比 50%，并通过串口打印值
旋转电位器最小值 0，设置的占空比 0%，并通过串口打印值

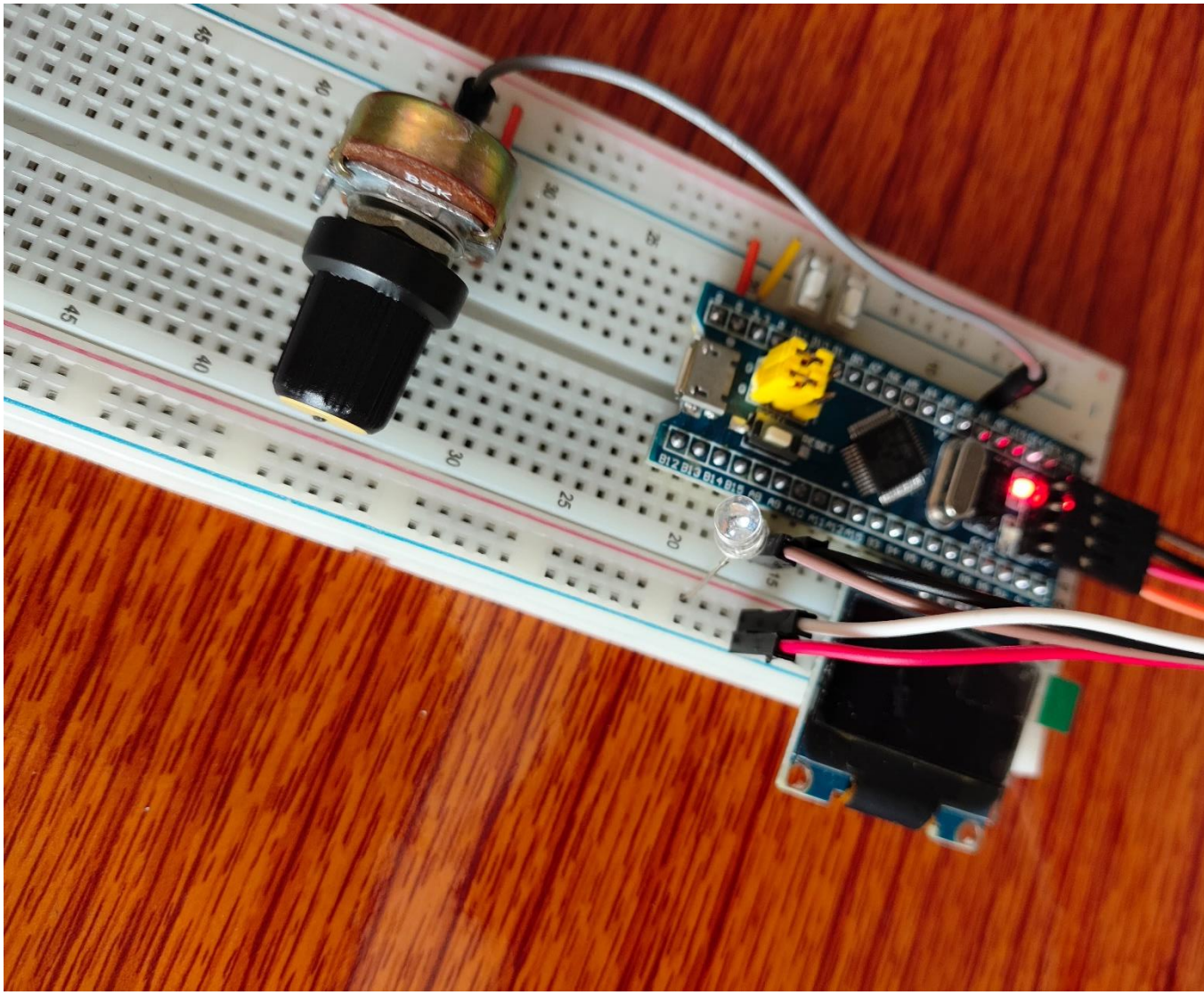





XCOM V2.6
ADValue:2033
Duty:3574
Voltage:1.64
ADValue:2034
Duty:3575
Voltage:1.65
ADValue:2050
Duty:3603
Voltage:1.65
ADValue:2049
Duty:3602
Voltage:1.65
ADValue:2048
Duty:3600
串口选择
COM3:USB-SERIAL CH340
波特率
115200
停止位
1
数据位
8
校验位
None
串口操作
关闭串口
保存窗口
清除接收
16进制显示
DTR
RTS
自动保存
时间戳
1000
ms
单条发送
多条发送
协议传输
帮助
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RS100手持示波器上市
www.openedv.com
S:0
R:3854
CTS=0 DSR=0 DCD=0
当前时间 12:13:47




C:\software\STM32CubeMX\project\Work\Work4\MDK-ARM\Work4.uvprojx - µVision
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Project: Work4
Work4
main.c
gpio.c
adc.c
tim.c
usart.c
stm32f1xx_it.c
stm32f1xx_hal_ms
Drivers/STM32F1xx_H
Drivers/CMSIS
CMSIS
main.c
stm32f1xx_hal_adc.h
while (1)
{
/* USER CODE END
/* USER CODE BEGI
ADValue = AD_GetV
Voltage = ((float
uint32_t CompareV
/* 设置PWM比较值,
__HAL_TIM_SET_COM
// 打印电压值、AD
printf("Voltage:%
printf("ADValue:%
printf("Duty:%lu\
printf("\r\n");
HAL_Delay(1000);
}
Build Output
Verify OK.
Application running ...
Flash Load finished at 12:11:19
XCOM V2.6
ADValue:1245
Duty:2188
Voltage:0.00
ADValue:0
Duty:0
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
www.openedv.com S:0 R:5388 CTS=0 DSR=0 DCD=0 当前时间 12:14:25
ST-Link Debugger
L:140 C:7


B5K