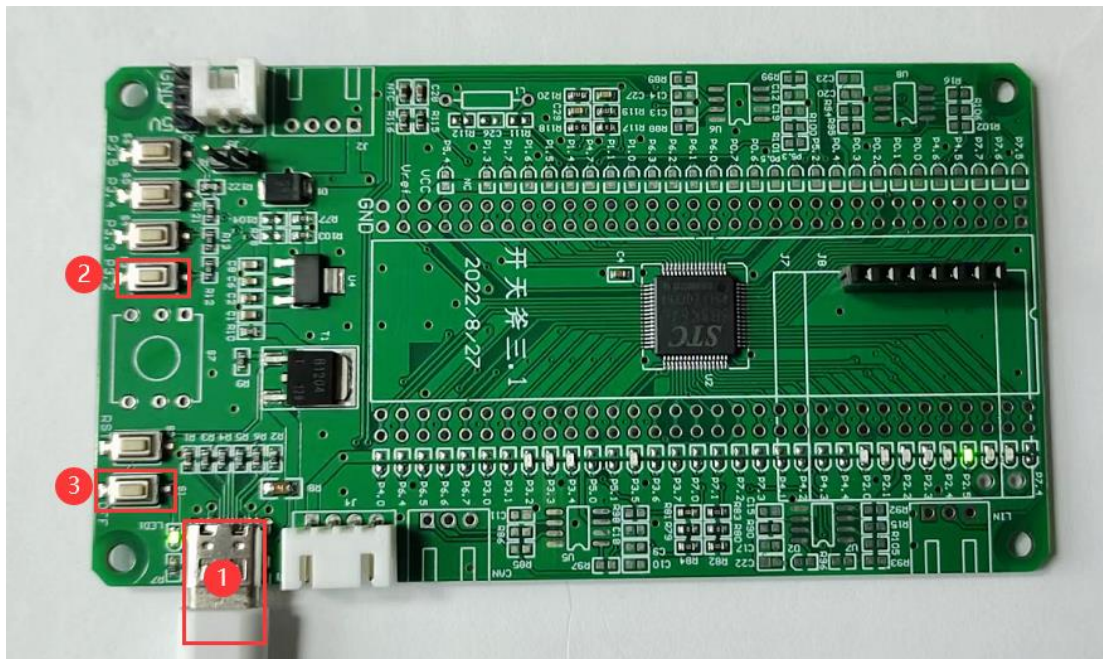


烧录说明

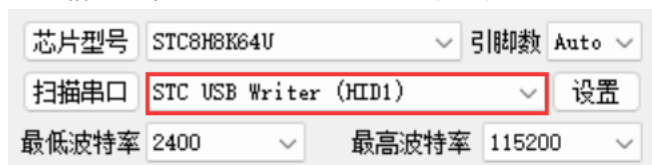
1 手动上电复位进行下载

1.1 硬件部分

如图所示：



- 1) 使用 TYPE-C 数据连接核心板到电脑；
 - 2) 按住 P3.2 口按键不放；
 - 3) 按一下电源 OFF 按键（按下-松开），然后可松开 P3.2 口按键；
- 正常情况下在 STC-ISP 软件上就可以识别出“STC USB Writer (HID1)”设备：

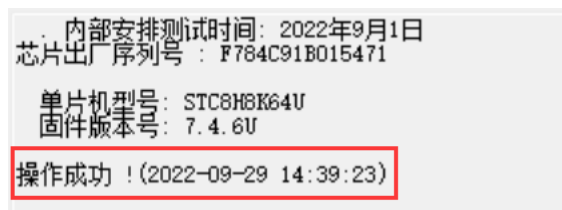


1.2 软件部分

如图所示：



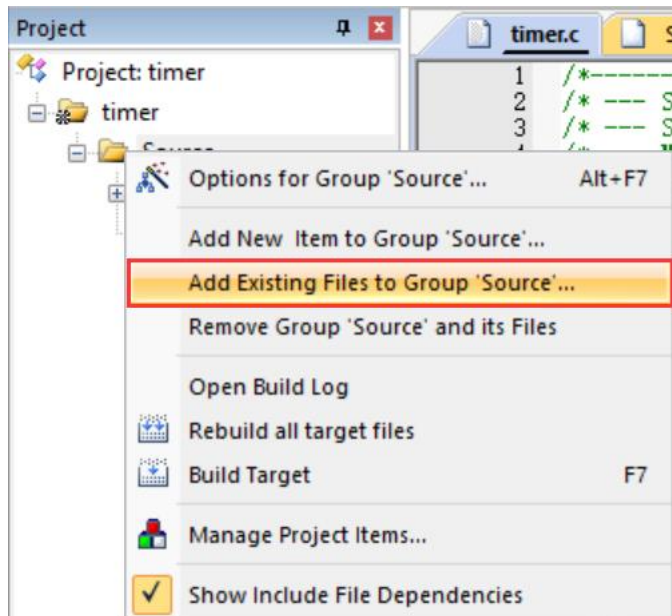
- 1) 选择芯片型号;
 - 2) 打开需要烧录的程序文件;
 - 3) 设置硬件选项;
 - 4) 然后就可以点击“下载/编程”按钮开始下载。
- 下载完成后右下角信息框会提示“操作成功!”。



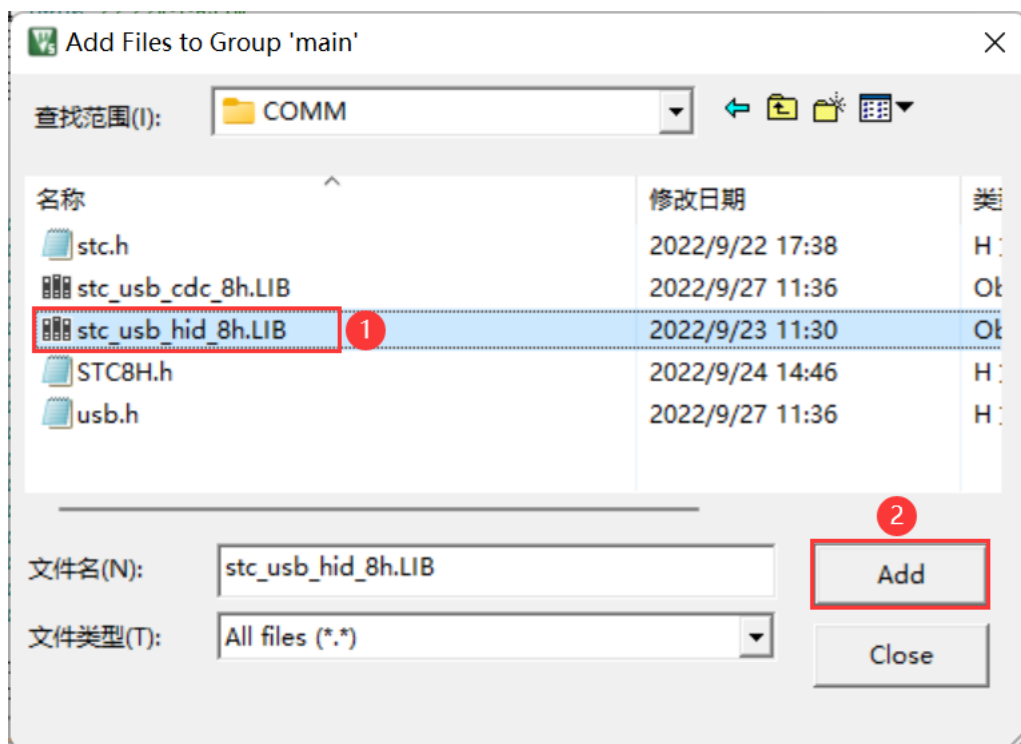
2 通过 USB HID 指令复位并下载

2.1 添加库文件

在项目栏程序组上点击鼠标右键，选择“Add Existing Files to Group ‘Source Group 1’...”：



在弹框里选择库文件，并添加到项目里：



2.2 添加软件配套代码

添加 USB 调试及复位所需定义：

```
//USB调试及复位所需定义
char *USER_DEVICEDESC = NULL;
char *USER_PRODUCTDESC = NULL;
char *USER_STCISPCMD = "@STCISP#"; //设置自动复位到ISP区的用户接口命令
```

在初始化程序里添加 USB 初始化及时钟源启动程序：

```
//USB调试及复位所需代码-----
P3M0 &= 0x03;
P3M1 |= 0x03;
IRC48MCR = 0x80;
while (!(IRC48MCR & 0x01));
usb_init();
//-----
EA = 1; //打开总中断
```

主循环或者中断等能够及时周期性处理程序里添加 USB 接收判断以及应答指令：

```
if (bUsbOutReady) //USB调试及复位所需代码
{
    usb_OUT_done();
}
```

注意：如果如果程序里有对 IE2 寄存器里 USB 以外的中断进行位操作使能的话，在 IE2 相关的中断位操作使能后，需要重新设置 EUSB：

IE2（中断使能寄存器 2）

| 符号 | 地址 | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
|-----|-----|------|-----|-----|-----|-----|-----|------|-----|
| IE2 | E7H | EUSB | ET4 | ET3 | ES4 | ES3 | ET2 | ESPI | ES2 |

IE2 |= 0x80; //IE2 相关的中断位操作使能后，需要重新设置 EUSB

至此，配套软件代码添加完毕。

2.3 发送指令触发 MCU 复位并自动下载

打开 STC-ISP 软件（V6.89E 以上版本），选择芯片型号、打开需要烧录的程序文件、设置好硬件选项后，切换到“收到用户命令后复位到 ISP 监控程序区”标签：

如下图所示，选择“USB(HID)模式”，设置默认 VID: 34BF，PID: FF01；

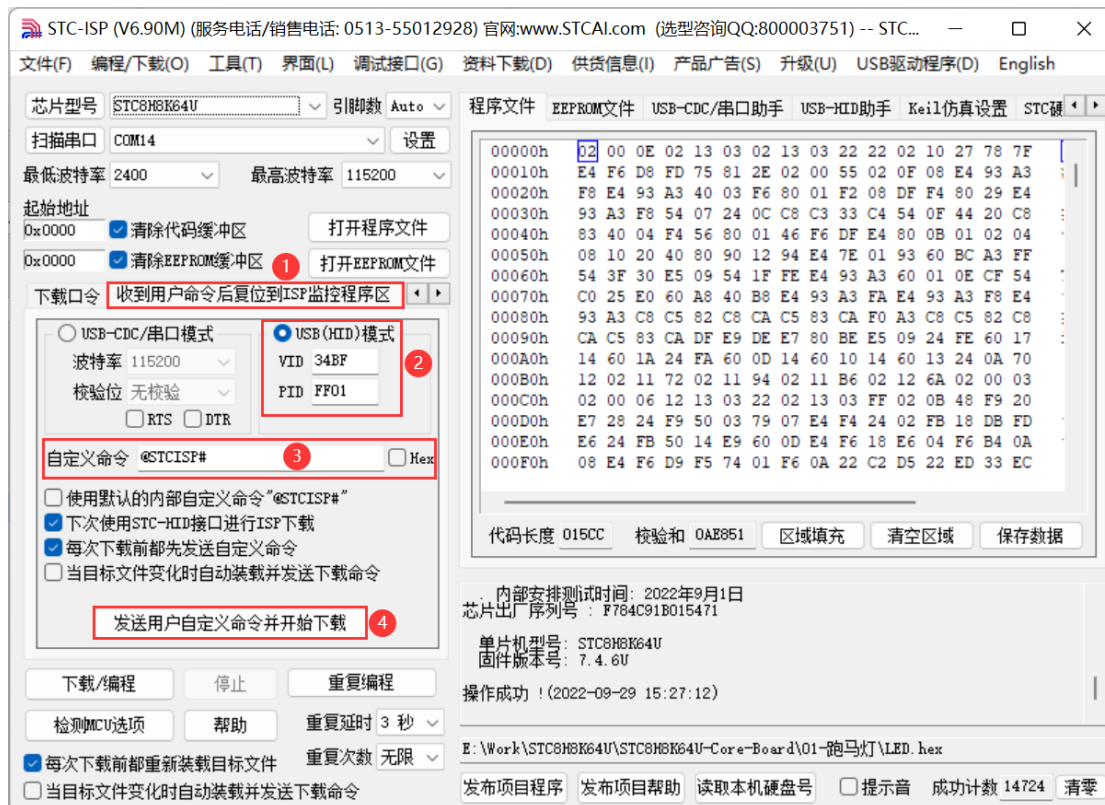
设置自定义命令：@STCISP#

取消“Hex”选项，命令应与程序里的定义相同：

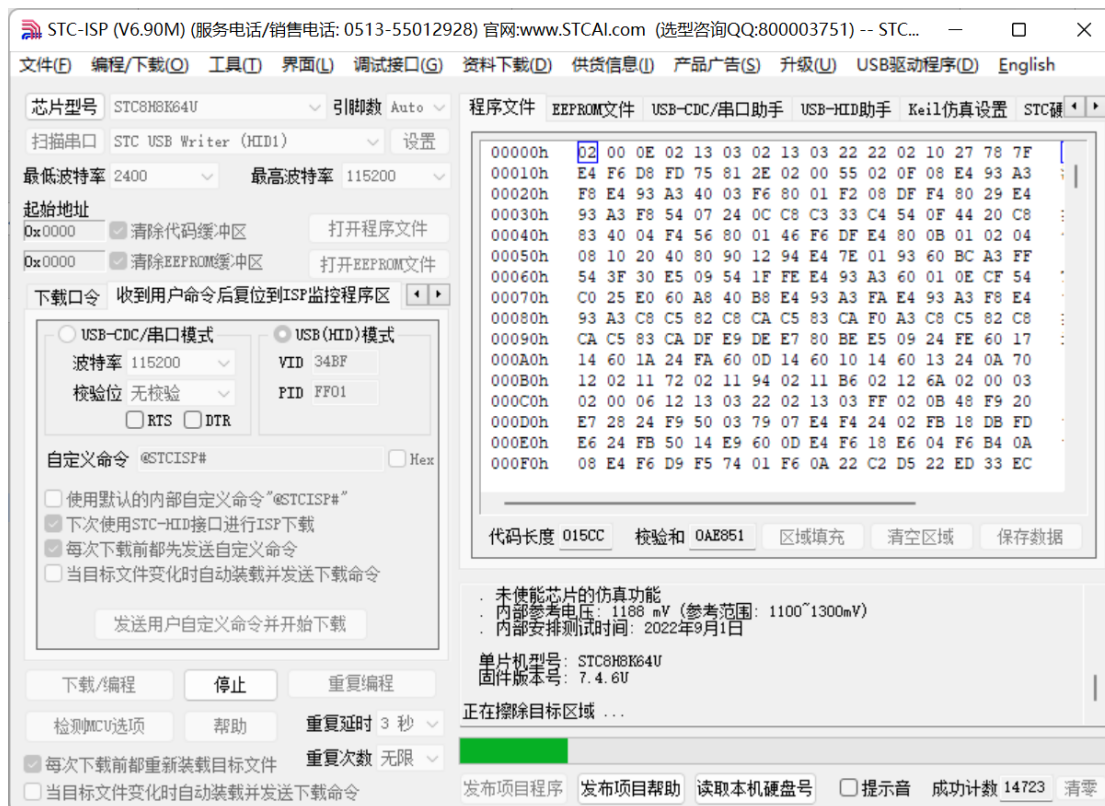
```
char *USER_STCISPCMD = "@STCISP#"; //设置自动复位到 ISP 区的用户接口命令
```

点击“发送用户自定义命令并开始下载”按钮，或者在勾选“每次下载前都先发送自定义命令”选项情况下也可点击“下载/编程”按钮开始下载。

（此步骤需要完成前两步，并将编译的程序通过手动上电复位的方式下载到芯片后才能使用）



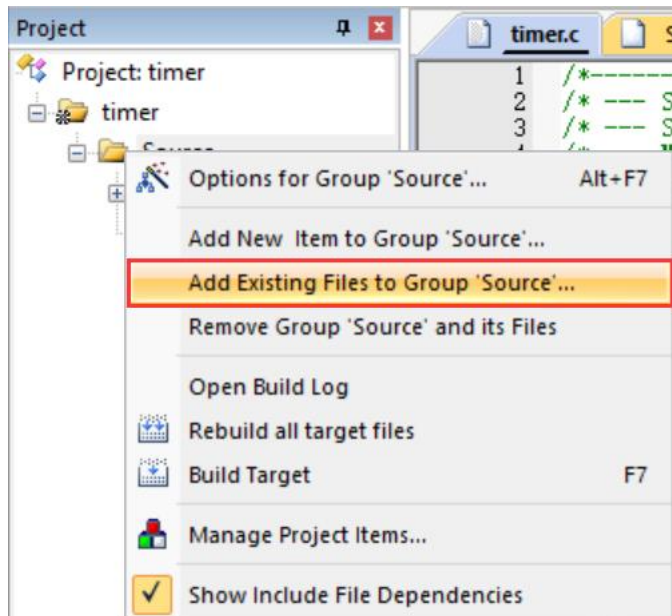
正常情况下芯片收到指令后会自动复位到“STC USB Writer (HID1)”模式并开始下载:



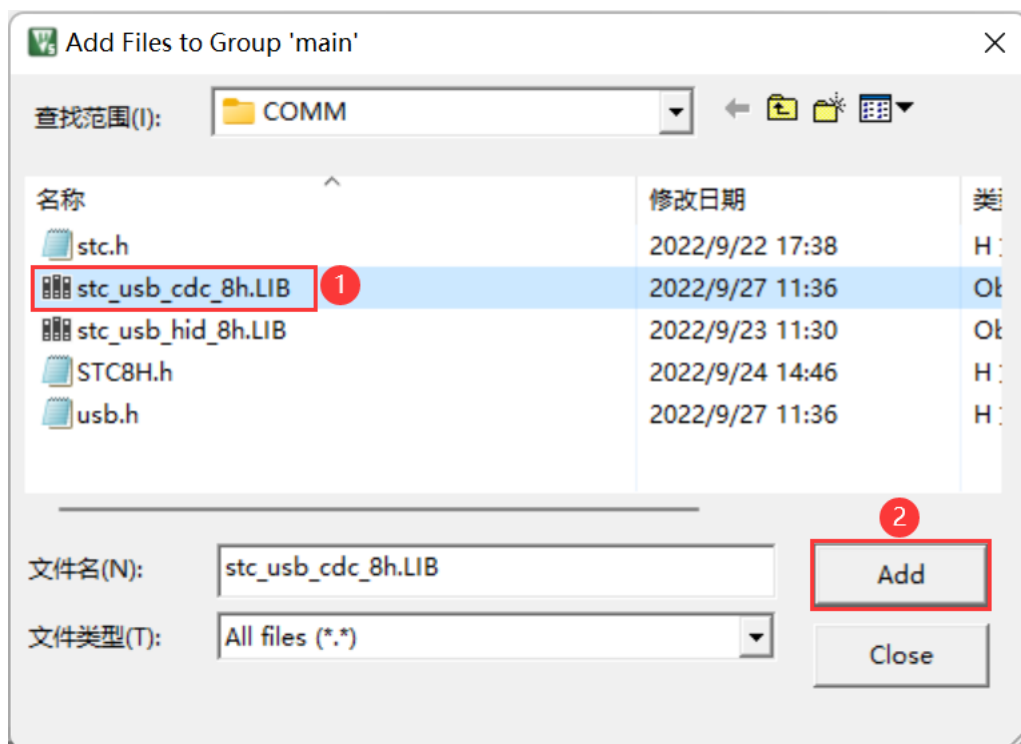
3 通过 USB CDC 指令复位并下载

3.1 添加库文件

在项目栏程序组上点击鼠标右键，选择“Add Existing Files to Group ‘Source Group 1’...”：



在弹框里选择库文件，并添加到项目里：



3.2 添加软件配套代码

添加 USB 调试及复位所需定义：

```
//USB调试及复位所需定义
char *USER_DEVICEDESC = NULL;
char *USER_PRODUCTDESC = NULL;
char *USER_STCISPCMD = "@STCISP#"; //设置自动复位到ISP区的用户接口命令
```

在初始化程序里添加 USB 时钟源启动，USB 模块进行初始化程序：

```
//USB调试及复位所需代码-----
P3M0 &= 0x03;
P3M1 |= 0x03;
IRC48MCR = 0x80;
while (!(IRC48MCR & 0x01));
usb_init();
//-----
EA = 1; //打开总中断
```

主循环里添加 USB 接收判断以及应答指令：

```
while (1)
{
    if (bUsbOutReady)
    {
        usb_OUT_done(); //接收应答（固定格式）
    }
}
```

注意：如果程序里有对 IE2 寄存器里 USB 以外的中断进行位操作使能的话，在 IE2 相关的中断位操作使能后，需要重新设置 EUSB：

IE2（中断使能寄存器 2）

| 符号 | 地址 | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
|-----|-----|------|-----|-----|-----|-----|-----|------|-----|
| IE2 | E7H | EUSB | ET4 | ET3 | ES4 | ES3 | ET2 | ESPI | ES2 |

IE2 |= 0x80; //IE2 相关的中断位操作使能后，需要重新设置 EUSB

至此，配套软件代码添加完毕。

3.3 发送指令触发 MCU 复位并自动下载

打开 STC-ISP 软件（V6.90I 以上版本），选择芯片型号、打开需要烧录的程序文件、设置好硬件选项后，切换到“收到用户命令后复位到 ISP 监控程序区”标签：

如下图所示，选择“串口模式”，设置默认波特率、校验位、停止位；

设置自定义命令：@STCISP#

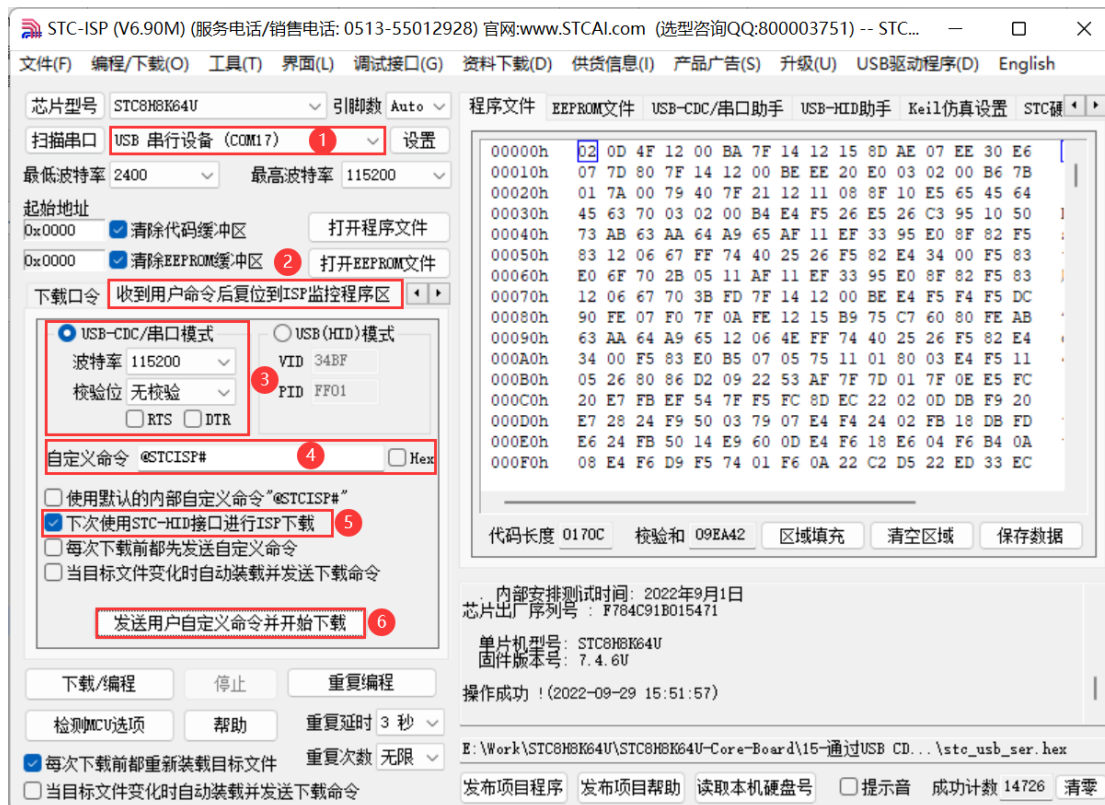
取消“Hex”选项，命令应与程序里的定义相同：

```
char *USER_STCISPCMD = "@STCISP#"; //设置自动复位到 ISP 区的用户接口命令
```

点击“发送用户自定义命令并开始下载”按钮，或者在勾选“每次下载前都先发送自定义命令”选项情况下也可点击“下载/编程”按钮开始下载。

（此步骤需要将支持自动下载功能的程序通过手动上电复位的方式下载到芯片后才能使用）

注意：使用 CDC 接口触发 MCU 复位并自动下载功能，需要勾选设置：“下次使用 STC-HID 接口进行 ISP 下载”：



正常情况下芯片收到指令后会自动复位到“STC USB Writer (HID1)”模式并开始下载。

4 通过 P3.2 口按键触发下载模式

4.1 添加软件配套代码

添加按键触发功能所需定义：

```
//P3.2口按键复位所需变量  
bit Key_Flag;  
ul6 Key_cnt;
```

添加 P3.2 口按键长按 1 秒触发软件复位，进入 USB 下载模式函数：

```
void KeyResetScan(void)  
{  
    if(!P32)  
    {  
        if(!Key_Flag)  
        {  
            Key_cnt++;  
            if(Key_cnt >= 1000)    //连续1000ms有效按键检测  
            {  
                Key_Flag = 1;    //设置按键状态，防止重复触发  
  
                USBCON = 0x00;    //清除USB设置  
                USBCLK = 0x00;  
                IRC48MCR = 0x00;  
  
                delay_ms(10);  
                IAP_CONTR = 0x60;    //触发软件复位，从ISP开始执行  
                while (1);  
            }  
        }  
    }  
    else  
    {  
        Key_cnt = 0;  
        Key_Flag = 0;  
    }  
}
```

在 1ms 周期性执行的中断或者主循环里调用按键扫描函数：

```
while (1)  
{  
    KeyResetScan();  
    delay_ms(1);  
}
```

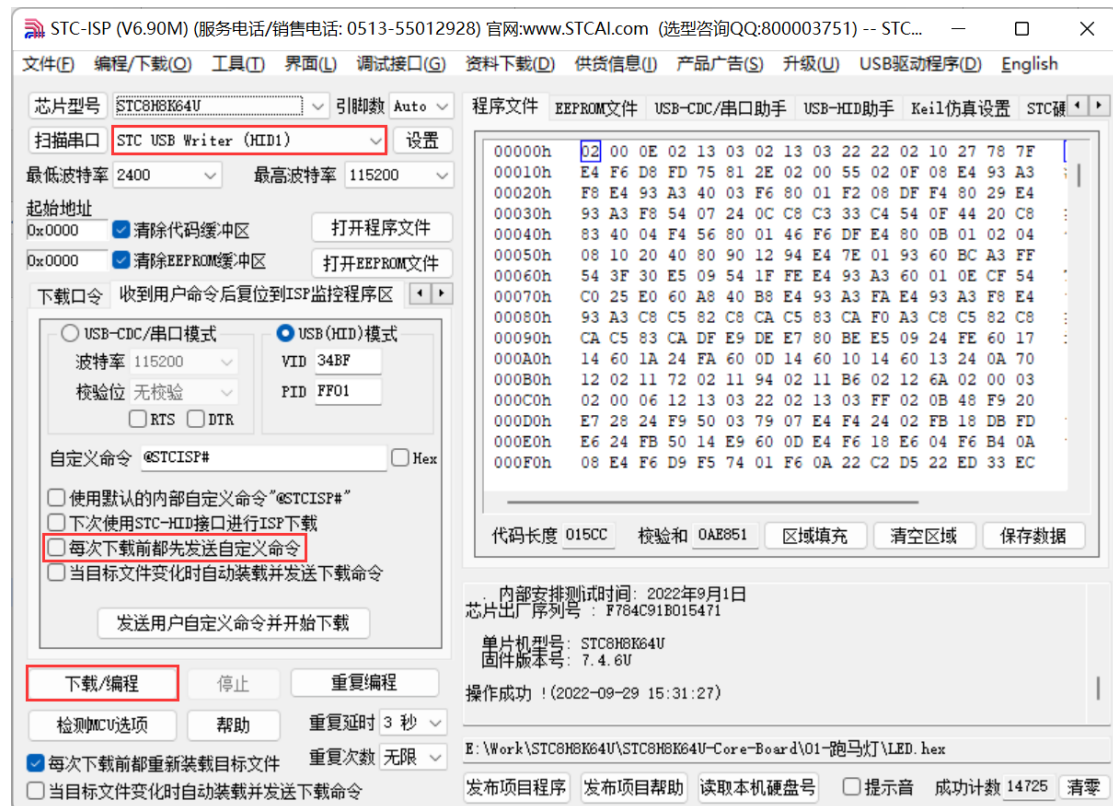
或者

```
void timer0_int(void) interrupt 1 //1ms 中断函数  
{  
    KeyResetScan();    //P3.2口按键触发软件复位，进入USB下载模式  
}
```

至此，配套软件代码添加完毕。

4.2 触发烧录模式并下载

完成以上步骤并将编译的程序通过手动上电复位的方式下载到芯片一次以后, 就可通过长按 P3.2 口按键方式自动触发进入 USB 下载模式:



注意, 需要将“收到用户命令后复位到 ISP 监控程序区”标签的设置, 取消“每次下载前都先发送自定义命令”选项, 然后再点击“下载/编程”按钮开始下载。

5 复位后执行区域设置

5.1 IAP 控制寄存器 (IAP_CONTR)

软件复位 MCU，可通过 IAP_CONTR 寄存器的 bit6 选择复位后执行区域。

SWBS 位置 0，软件复位后进入用户程序区域；

SWBS 位置 1，软件复位后进入 ISP 程序区域（USB 下载模式）：

| 符号 | 地址 | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
|-----------|-----|-------|------|-------|----------|----|----|----|----|
| IAP_CONTR | C7H | IAPEN | SWBS | SWRST | CMD_FAIL | - | - | - | - |

SWBS：软件复位启动选择

0：软件复位后从用户程序区开始执行代码。用户数据区的数据保持不变。

1：软件复位后从系统 ISP 区开始执行代码。用户数据区的数据会被初始化。

SWRST：软件复位触发位

0：对单片机无影响

1：触发软件复位

5.2 复位标志寄存器 (RSTFLAG)

看门狗复位、复位脚复位等硬件复位，默认会使芯片进入 USB 下载模式，可在初始化代码里将 RSTFLAG 的 bit2 写 1，清除 SWRSTF 标志，设置复位后需要检测 P3.2 的状态选择运行区域进行避免：

| 符号 | 地址 | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
|---------|-------|----|----|----|---------|---------|--------|--------|--------|
| RSTFLAG | FE99H | - | | | LVDRSTF | WDTRSTF | SWRSTF | ROMOVF | EXRSTF |

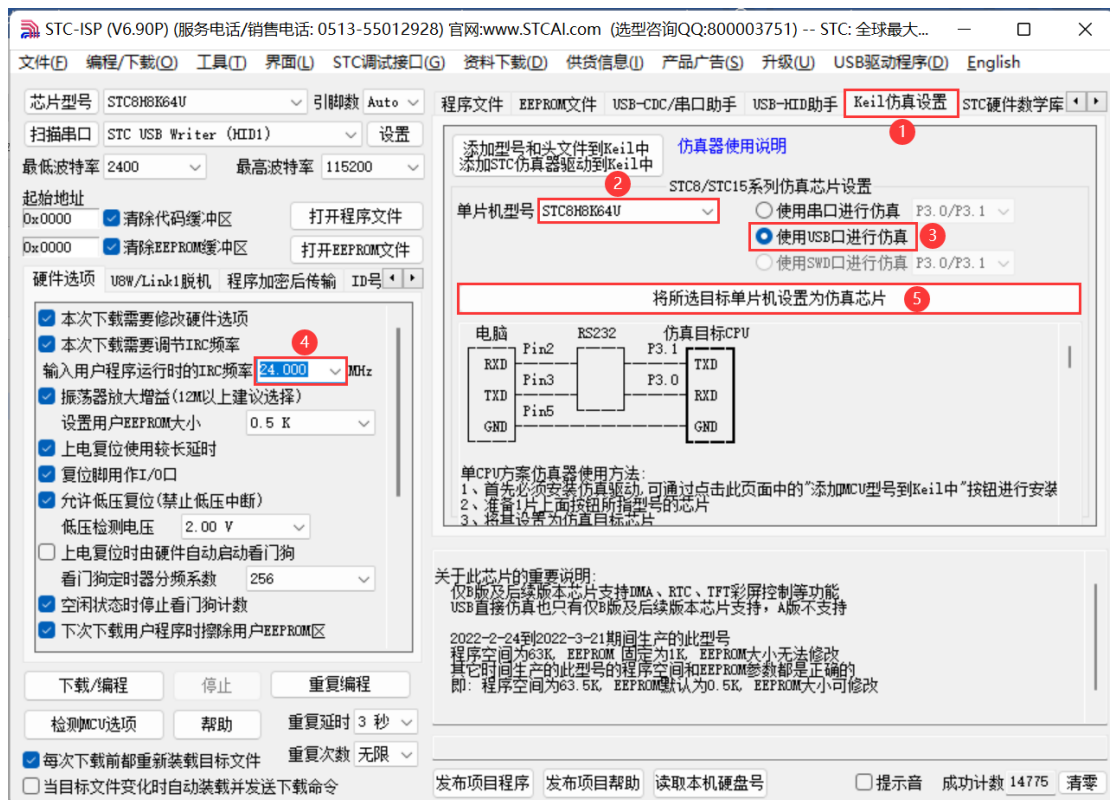
6 仿真说明

由于本例程包里面的大部分例子都添加了通过 USB 接口实现不停电下载功能，用到了 P3.0, P3.1 接口，而仿真也需要通过这两个接口进行通信，所以仿真前需要将这块功能先屏蔽掉：

```
//USB调试及复位所需代码-----  
// P3M0 &= 0x03;  
// P3M1 |= 0x03;  
// IRC48MCR = 0x80;  
// while (!(IRC48MCR & 0x01));  
// usb_init();  
//-----
```

6.1 制作仿真芯片

参考 1.1 章节方法进入 USB 下载模式，打开 STC-ISP 软件：



- 1) 选择“Keil 仿真设置”标签；
- 2) 选择单片机型号；
- 3) 使用 USB 口进行仿真；
- 4) 设置仿真频率（即单片机运行频率）；
- 5) 开始制作仿真芯片。

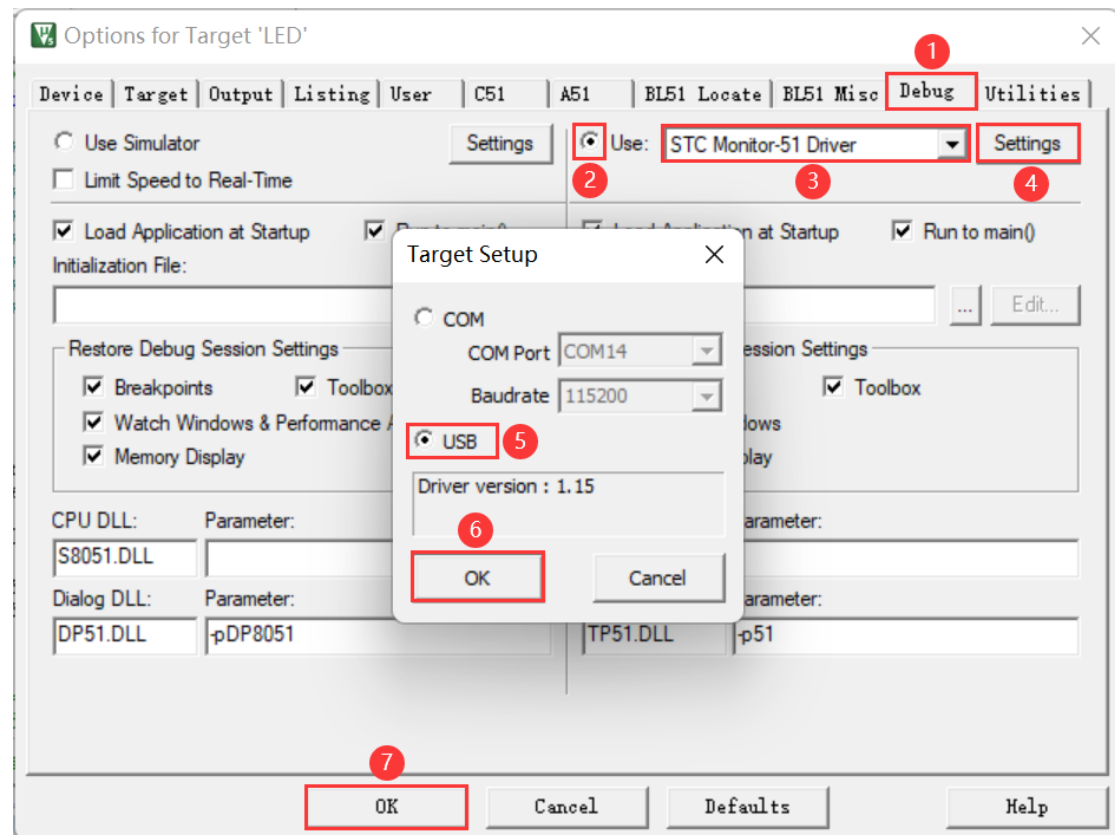
制作完仿真芯片后给 MCU 重新上电，然后再开始仿真。

6.2 Keil 仿真设置

Keil 打开需要仿真的项目，按下快捷键“Alt+F7”或者选择菜单“Project”中的“Option for Target ‘Target1’”或者通过“Option for target...”按钮打开设置界面：



按照如下步骤设置“Debug”标签：

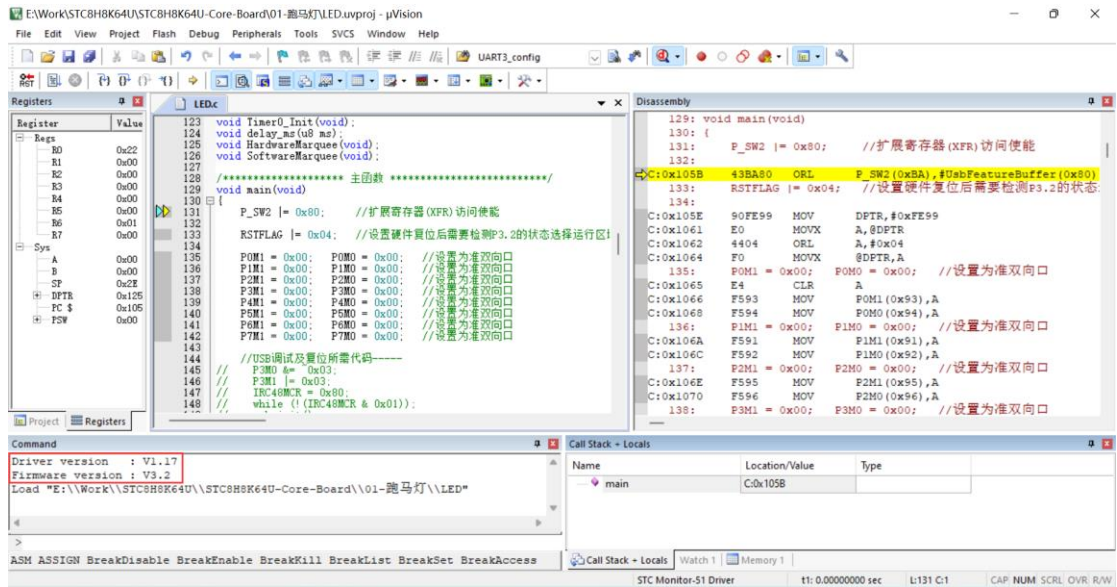


- 第 1 步、进入到项目的设置页面，选择“Debug”设置页；
 - 第 2 步、选择右侧的硬件仿真“Use …”；
 - 第 3 步、在仿真驱动下拉列表中选择“STC Monitor-51 Driver”项；
 - 第 4 步、点击“Settings”按钮，进入接口的设置画面；
 - 第 5 步、选择 USB 接口。
- 确定完成仿真设置。

注意：第 3 步在仿真驱动下拉列表中选择“STC Monitor-51 Driver”项，不是“~~Keil-Monitor-51-Driver~~”。

6.3 开始仿真

完成了上面所有的工作后，即可在 Keil 软件中按“Ctrl+F5”开始仿真调试。
若硬件连接无误的话，将会进入到类似于下面的调试界面，并在命令输出窗口显示当前的仿真驱动版本号和当前仿真监控代码固件的版本号，如下图所示：



仿真调试过程中，可执行复位、全速运行、单步运行、设置断点等多中操作：

