使用单片机的用户系统区
开发自己的 ISP 程序

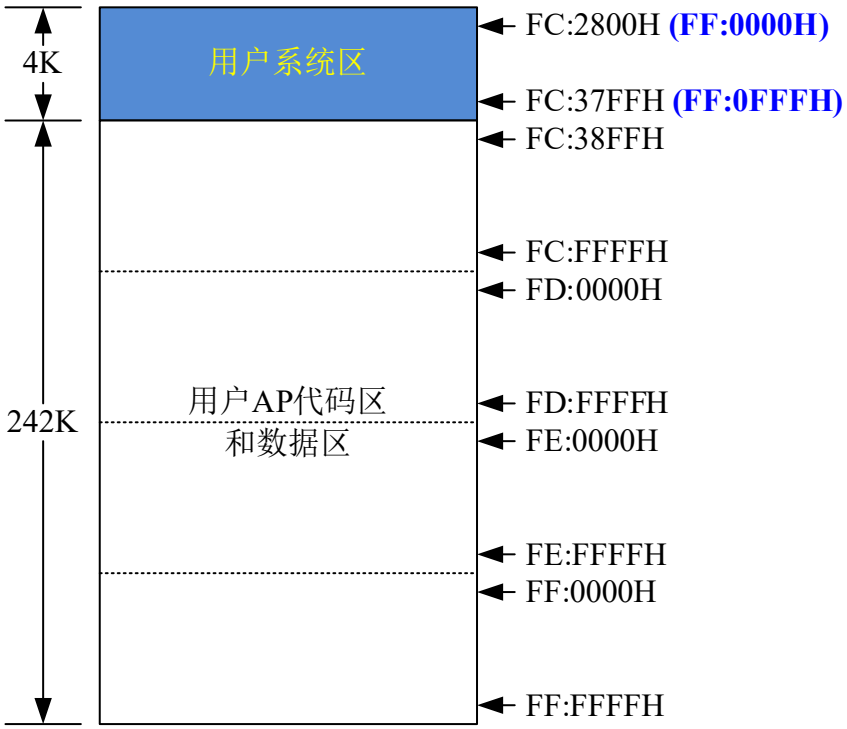
——基于 STC32G144K246

用户系统区是指在用户的 EEPROM 区分配部分区域当作用户系统程序区，使用用户系统区的用户 ISP 代码对用户代码区和用户 EEPROM 区进行编程，已达到使用用户 ISP 代码对用户区进行代码更新的目的。

本文以 STC32G144K246 单片机为例，详细说明使用单片机的用户系统区开发用户自己的 ISP 程序的方法，并给出了基于 Keil 环境的 C 源码。

一. 内部 FLASH 规划

示例单片机使用 STC32G144K246，用户可以使用的最大程序空间为 246K 字节，整个 Flash 空间划分如下：（注：示例代码为用户系统区分配 4K 字节大小，用户可根据实际需要在 ISP 软件中进行任意分划分）

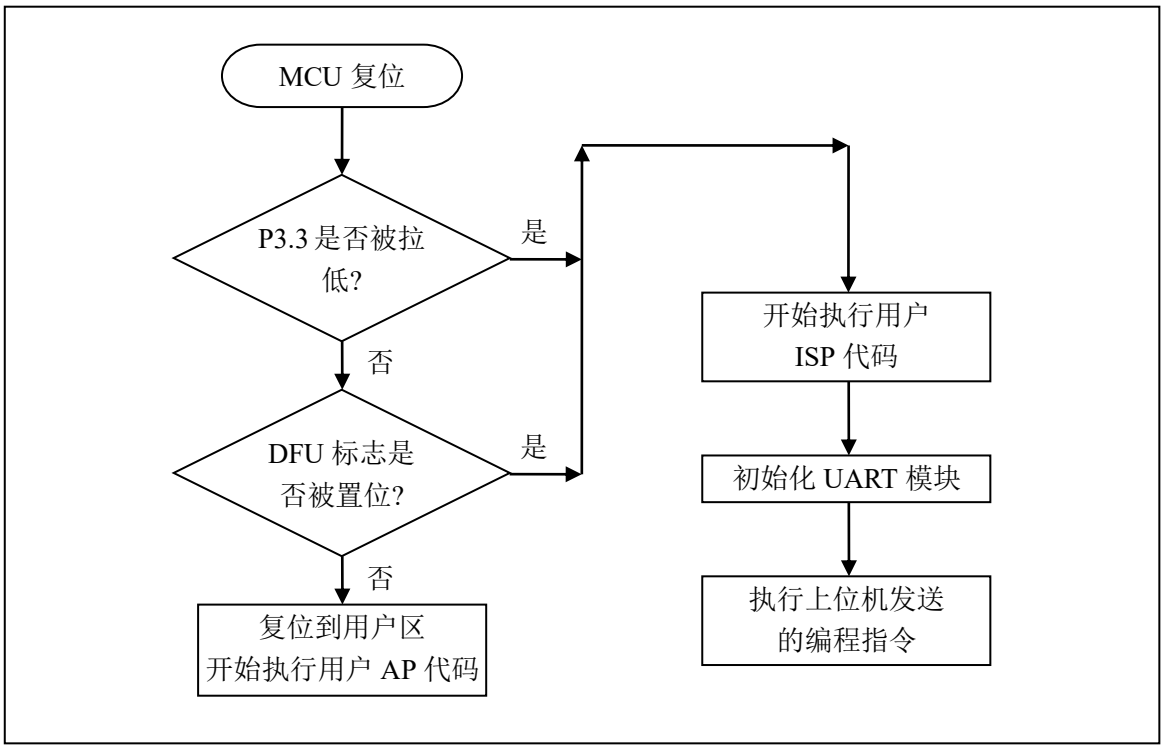


246K 的用户 FLASH，地址范围为 FC:2800H~ FF:FFFFH，用户 ISP 代码会占据其中 FC:2800H～FC:37FFH 的 4K 字节（整个 246K 字节 Flash 的最前 4K 字节），剩余的 242K 字节的空间为用户程序区。由于用户系统区是一块独立的程序区，地址有进行重映射，FC:2800H～FC:37FFH 这块地址空间，相对于用户系统区，逻辑地址范围为：FF:0000H～FF:0FFFH。当满足特定的下载条件时，用户 ISP 会根据上位机发送的指令对 242K 字节的用户 AP 代码区和数据区的 FLASH 进行擦除和编程，以达到更新用户程序的目的。

二． 代码权限

Flash 区 / 代码	用户系统区	用户代码区	用户 EEPROM 区
用户 ISP 代码	读取数据：允许 编程数据：禁止 运行代码：允许	读取数据：允许 编程数据：允许 运行代码：禁止	读取数据：允许 编程数据：允许 运行代码：禁止
用户代码	读取数据：禁止 编程数据：禁止 运行代码：禁止	读取数据：允许 编程数据：禁止 运行代码：允许	读取数据：允许 编程数据：允许 运行代码：禁止

三． 程序的基本框架



四. 下位机固件程序说明

下位机固件程序包括两部分：用户 ISP 代码和用户 AP 代码

- **用户 ISP 代码：**

用户 ISP 代码主要用户接收上位机的编程指令，对用户 AP 区进行代码更新。用户 ISP 代码的通讯接口为 UART 接口。

用户 ISP 代码执行的条件：

1、 断电并重新上电后，P3.3 口被拉到低电平。

当用户需要上电后直接执行用户 ISP，可将 P3.3 口经过 1K 电阻连接到 GND，再对 MCU 进行重新上电，用户 ISP 代码会立即执行。

2、 DFU 标志被置位。

用户 ISP 代码在扩展 RAM 的最后地址（XDATA：FFFCH～FFFFH）定义了一个 4 字节的 DFU 标志变量（DfuFlag）。当 DfuFlag 被赋值为 0x12abcd34 时，用户 ISP 代码也会开始执行。这个功能主要应用于：当 MCU 已经在执行用户 AP 代码后，若用户希望在 P3.3 没有接到 GND 时也能运行用户 ISP 代码更新程序，则可将 DfuFlag 标志变量赋值为 0x12abcd34，然后软复位（向 IAP_CONTR 寄存器中写 **28H**），用户 ISP 代码就会开始运行了。

- **用户 AP 代码**

用户 AP 代码是用户的正常功能代码。由于用户 ISP 代码是一块独立于用户区的区域，相对于用户 AP 区是透明的，所以用户 AP 代码项目无需做任何特殊处理。

若需要在用户 AP 代码中实现软复位到用户 ISP 升级程序的功能，则需要用户端 AP 代码进行如下所示的几点设置。

```
#include "stc32g144k246.h"
#include "intrins.h"

#define FOSC            48000000UL
#define T1MS            (65536 - FOSC/1000)

#define DFU_TAG         0x12abcd34      //DFU强制执行标志
long xdata DfuFlag _at_ 0xfffc;         //DFU标志，定义在xdata的最后4字节

void sys_init();

int cnt200;

        if (P32 == 0)
        {
            DfuFlag = DFU_TAG;          //当需要执行用户ISP代码时,将强制执行标志赋值到DFU标志变量中
            IAP_CONTR = 0x28;           //然后执行软复位指令触发芯片软复位到用户系统区
        }

void sys_init()
{
    WTST = 0x00;
    CKCON = 0x00;
    EAXFR = 1;

    P2M0 = 0xff;
    P2M1 = 0x00;
    P3M0 = 0x00;
    P3M1 = 0x00;

    P32 = 0;
    _nop_();
    _nop_();
    P32 = 1;
    P3PU |= 0x04;

    TMOD = 0x00;
    T0x12 = 1;
    TL0 = T1MS;
    TH0 = T1MS >> 8;
    TR0 = 1;
    ET0 = 1;

    DfuFlag = 0;                        //上电正常执行用户AP,时需要将DFU标志清零
    cnt200 = 0;

    EA = 1;
}
```

当检测到满足用户 ISP 下载条件后，只需要将 DFU_TAG 赋值到 DfuFlag 变量中，然后软复位即可。

五． 上位机应用程序说明

上位机演示程序是基于 MFC 的对话框项目，对于串口的访问是直接调用 Windows 的 API 函数。界面较简单，只是为这一功能的实现提供了一个框架，其他的功能及要求均还可以往上面添加。

上位机程序的核心模块是基于类 CStcIsp_UserDlg 的一个静态线程函数“static UINT Download(LPVOID pParam);”

```
////////////////////////////////////////////////////////////////////////////
// CStcIsp_UserDlg dialog

class CStcIsp_UserDlg : public CDialog
{
// Construction
public:
    CStcIsp_UserDlg(CWnd* pParent = NULL);  // standard constructor
    virtual ~CStcIsp_UserDlg();

    void ShowMessage(LPCTSTR lpMsg, ...);
    BOOL LoadCode(LPCTSTR lpszFile, BOOL bHex);

    static UINT Download(LPVOID pParam);

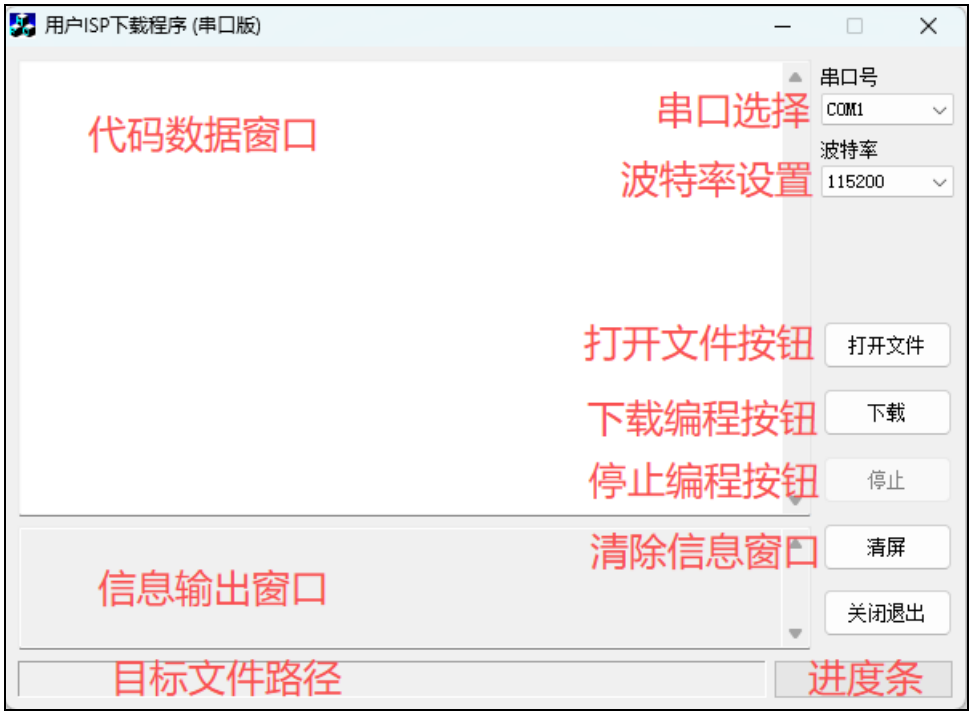
    BOOL HidOpen(WORD VID = 0x34bf, WORD PID = 0xff01);
    void HidClose();
    BOOL HidRead(BYTE *pData, DWORD dwInterval = 100);
    BOOL HidWrite(BYTE bCmd, DWORD dwAddress, BYTE bSize, BYTE *pData, DWORD dwInterval = 100);

// Dialog Data
    //{{AFX_DATA(CStcIsp_UserDlg)
    enum { IDD = IDD_STCISP_USER_DIALOG };
    CStatic m_ctlMessageStatic;
```

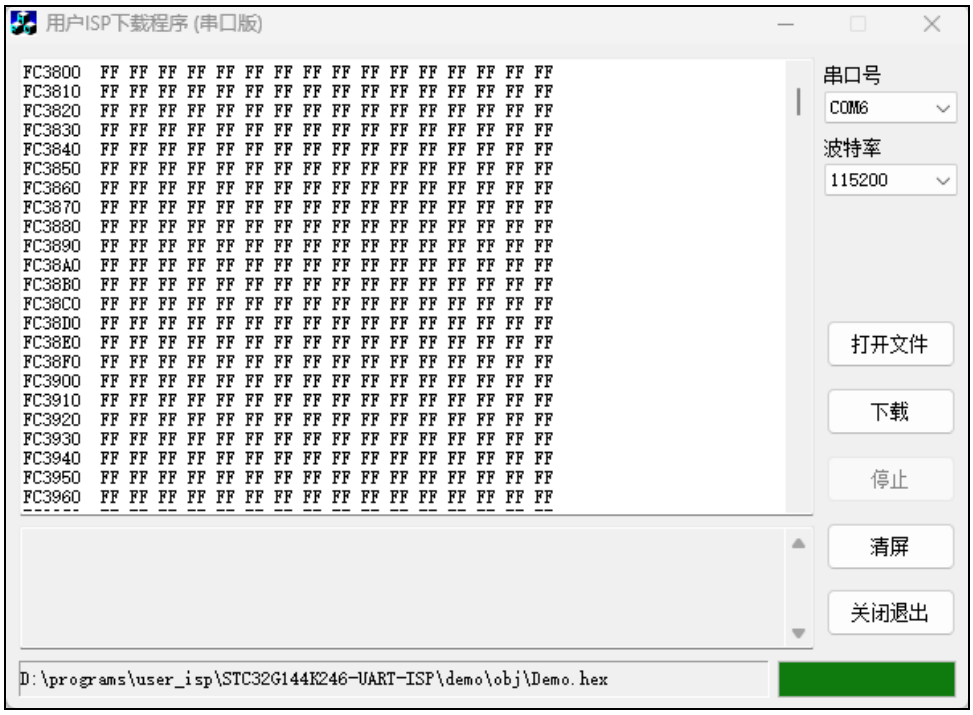
此函数负责与下位机通讯，发送各种通讯命令来完成对用户程序的更新。用户可以根据各自不同的需求增加命令。

六． 上位机应用程序的使用方法

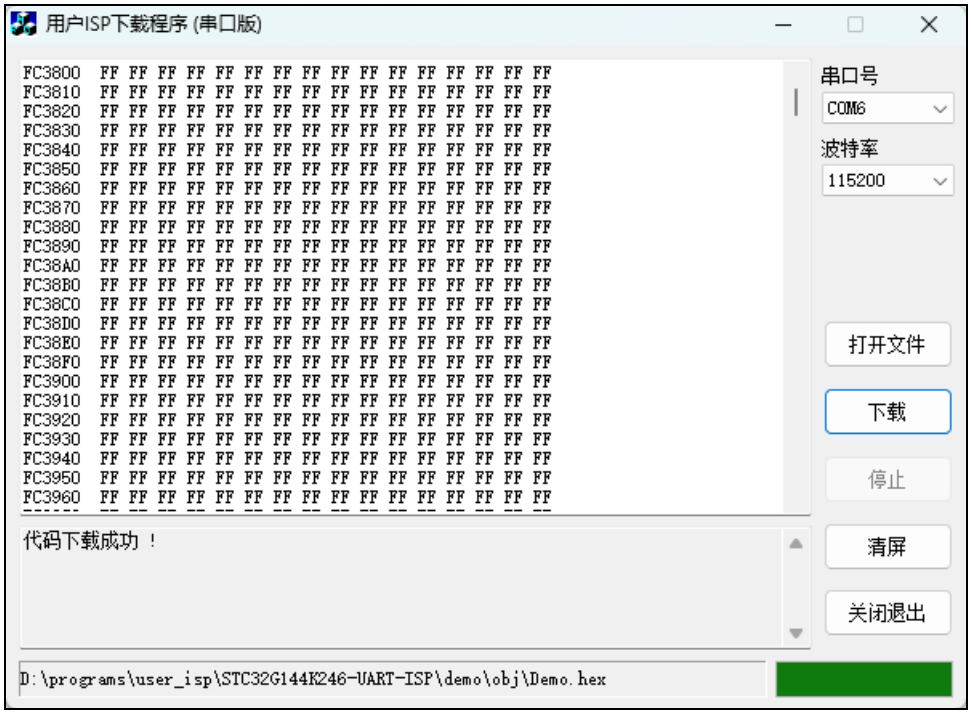
1、打开上位机界面，如下图



2、点击“打开文件”按钮来打开要下载的源数据文件，Bin 或者 Intel hex 格式均可以。打开代码文件，如下图所示：



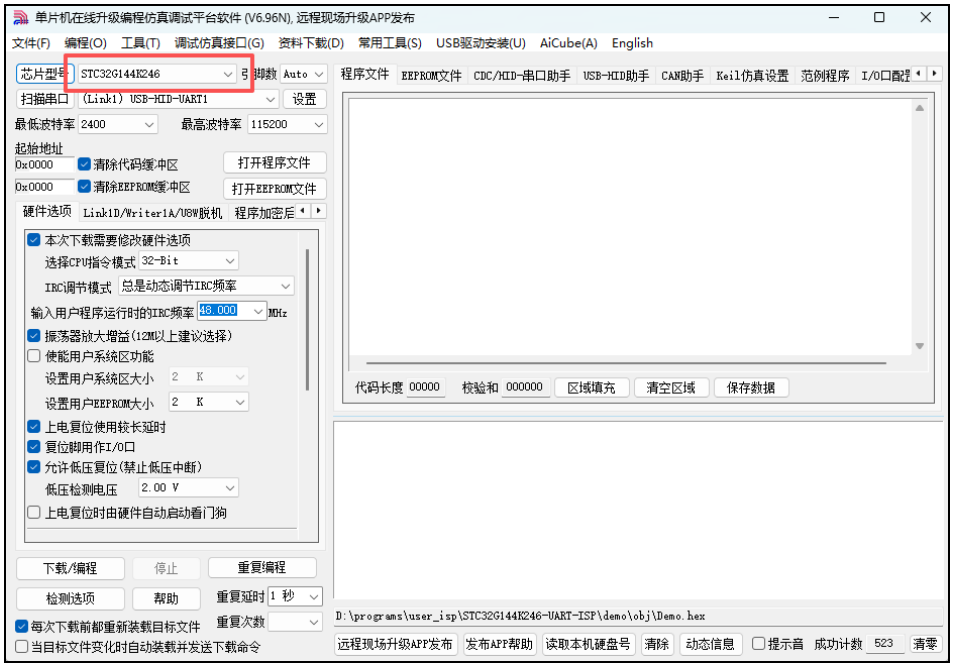
3、点击“下载”按钮即可开始下载数据



七. 整体操作流程

1、拿到一片全新的 STC32G144K246 的单片机时，首先需要参考数据手册中的串口 ISP 下载参考线路图，将单片机和电脑的串口连接好。

2、打开最新的 STC-ISP 下载软件，选择“STC32G144K246”目标单片机型号。

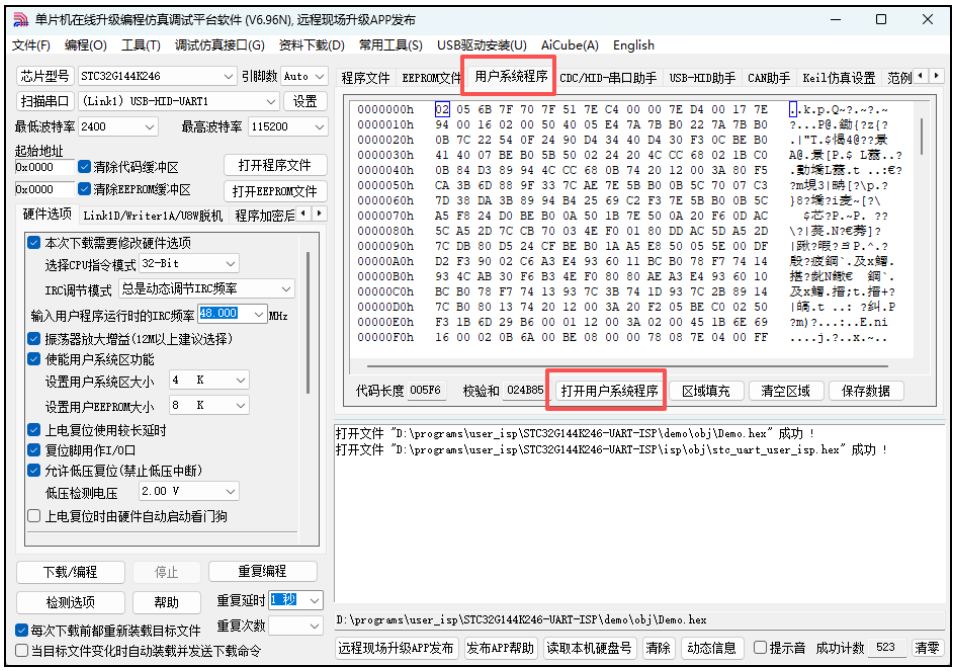


3、点击界面中的“打开程序文件”按钮，打开范例程序包中的用户 AP 示例代码文件“demo\obj\demo.hex”

4、按照如下图所示设置硬件选项，工作频率为 48MHz、EEPROM 大小为 8K、勾选“使能用户系统区功能”、设置用户系统区大小为 4K



5、点击“用户系统程序”界面中的“打开用户系统程序”按钮，打开范例程序包中的用户 ISP 代码文件“isp\obj\stc_uart_user_isp.hex”



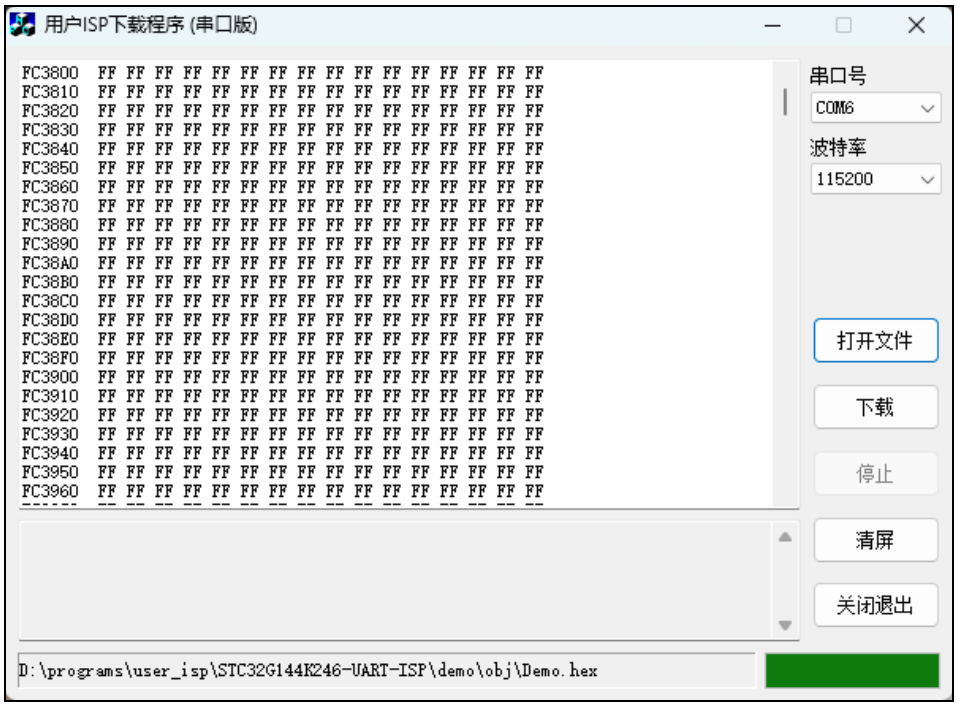
6、点击下载/编程按钮，将用户 ISP 代码和用户 AP 演示代码一起下载到

目标单片机中

7、给目标单片机重新断电并上电一次，设置的用户系统区功能、用户系统区大小以及 EEPROM 大小的硬件选项必须重新上电后才能生效（重要，容易被忽略）

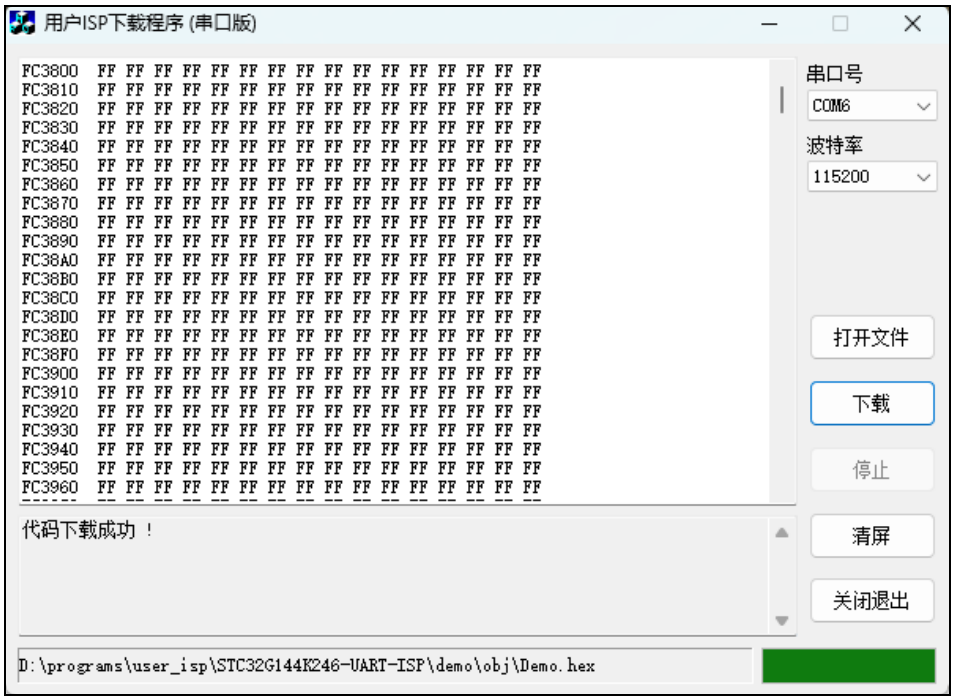
8、经过第 7 步的重新上电后，用户系统区功能已经被使能，用户 AP 代码也已经开始运行（Demo 程序是 P2 口闪烁）。

9、打开范例程序包中的“\app\Release\StcIsp_User.exe”，点击“打开文件”按钮，可再次打开范例程序包中的“demo\obj\demo.hex”，也可打开用户需要更新的 AP 代码



10、 将 P3.2 口通过 1K 电阻短接到 GND 然后松开，程序会复位到用户系统区（此功能在 AP 演示代码中已实现）。或者将 P3.3 口通过 1K 电阻连接到 GND，再给目标芯片重新上电，程序也会复位到用户系统区。点击“下载”按钮，即可完成用户 AP 代码的更新

11、 下载完成后，会显示如下图所示的“代码下载成功” 画面



12、 若在后续的开发中，在没有 P3.3 口被接 GND 或者 DFU 标志变量被置位时，每次重新上电都会跳过用户 ISP 代码而直接运行 AP 代码。若需要重新执行用户 ISP 代码来更新用户 AP 代码时，只需要回到第 9 步，按照指示操作，即可继续使用用户 ISP 更新用户 AP 代码了。