

# 使用 STC 的 IAP 系列单片机 开发自己的 ISP 程序

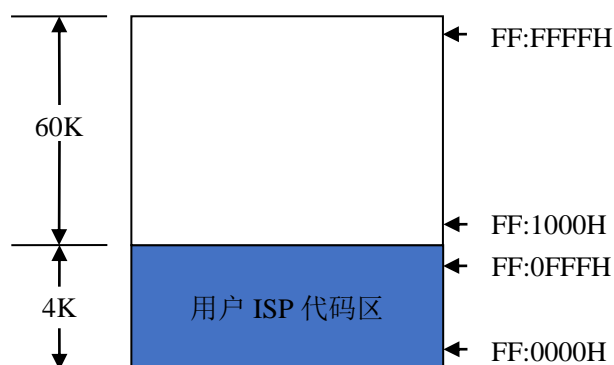
——基于 STC32G8K64

随着 IAP (In-Application-Programming) 技术在单片机领域的不断发展，给应用系统程序代码升级带来了极大的方便。STC 的串口 ISP (In-System-Programming) 程序就是使用 IAP 功能来对用户的程序进行在线升级的，但是出于对用户代码的安全着想，底层代码和上层应用程序都没有开源，为此 STC 推出了 IAP 系列单片机，即整颗 MCU 的 Flash 空间，用户均可在自己的程序中进行改写，从而使得有用户需要开发自己的 ISP 程序的想法得以实现。

本文以 STC32G8K64 为例，详细说明使用 STC 的 IAP 单片机开发用户自己的 ISP 程序的方法，并给出了基于 Keil 环境的 C 源码。

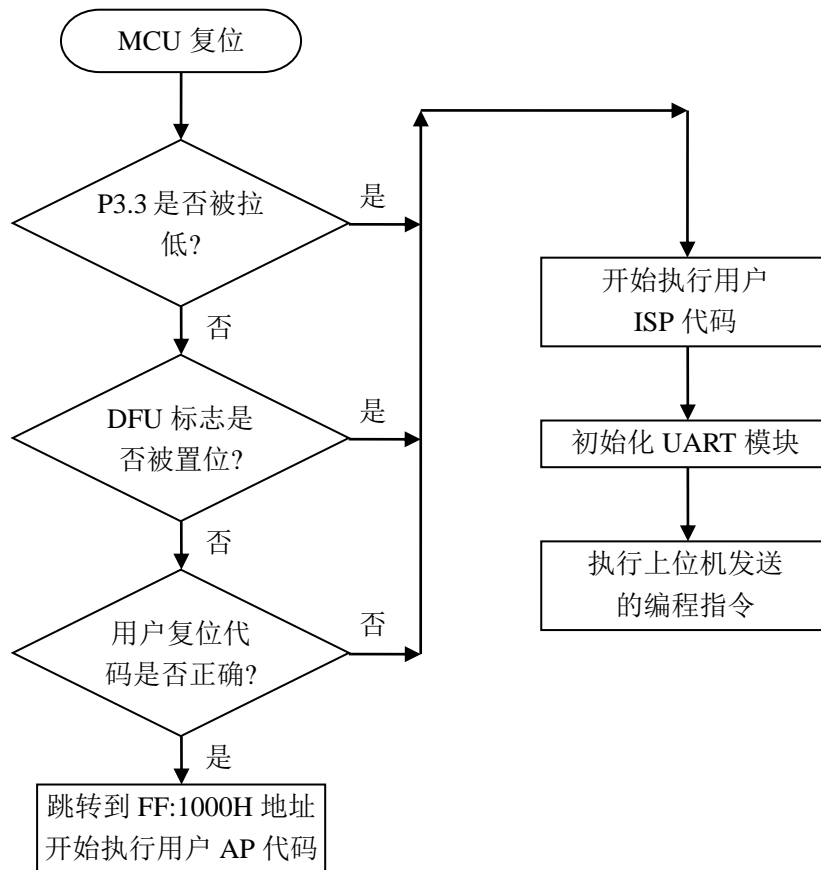
## 一. 内部 FLASH 规划

示例单片机使用 STC32G8K64, 用户可以使用的最大程序空间为 64K 字节, 整个 Flash 空间划分如下:



64K 的用户 FLASH, 地址范围为 FF:0000H~ FF:FFFFH, 用户 ISP 代码会占据其中 FF:0000H~FF:0FFFH 的 4K 字节, 剩余的 60K 字节的空间为用户程序区。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除 4K 用户 ISP 区以外的 60K 字节的 FLASH 进行擦除和编程, 以达到更新用户程序的目的。

## 二. 程序的基本框架



### 三. 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: 17FCH~17FFH)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 20H），用户 ISP 代码就会开始运行了。

- 3、 用户 AP 代码的复位指令不规范。

用户 AP 代码的复位代码必须是一条长跳转指令（第一个字节的指令码必须为 02H），且长跳转的地址必须跨过用户 ISP 代码的 FF:0000H~FF:0FFFH 的 4K 空间。否则用户 AP 的复位代码不规

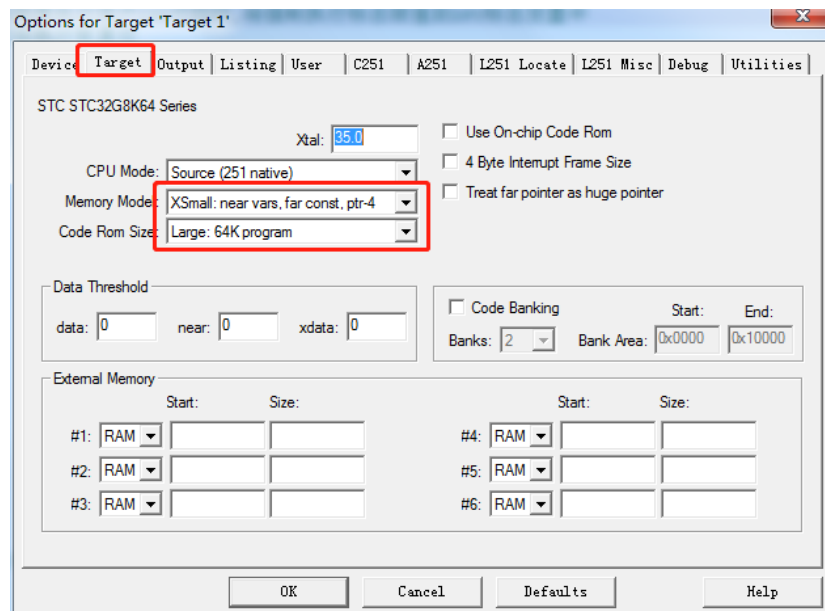
范，无法正常运行用户 AP 代码，此时会强制执行用户 ISP 代码。

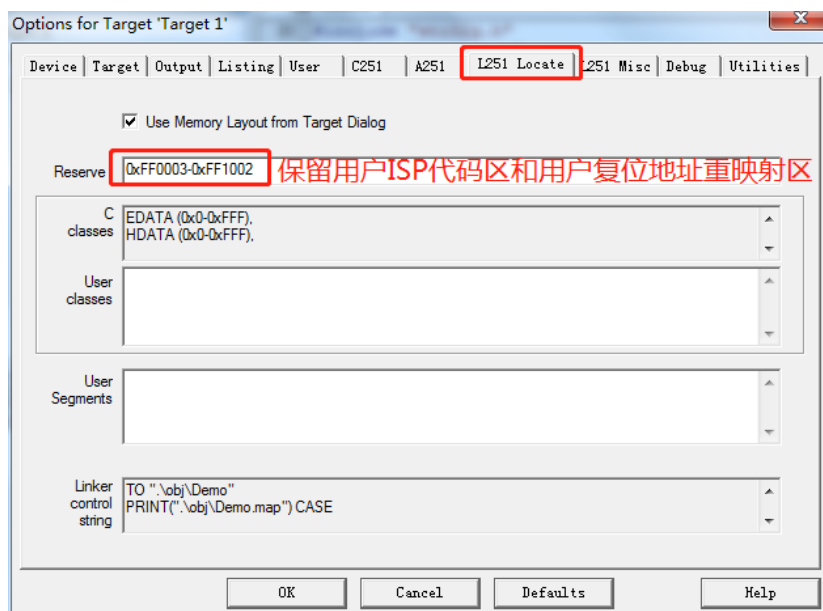
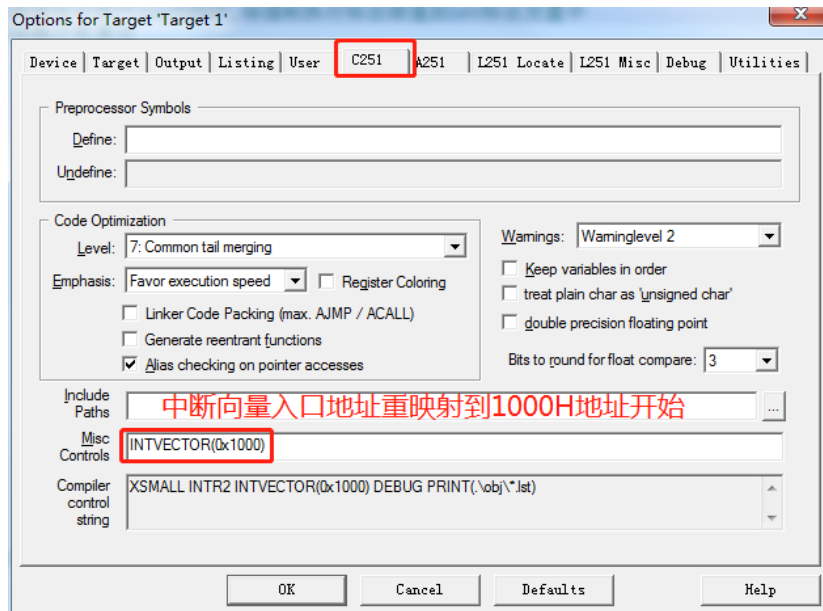
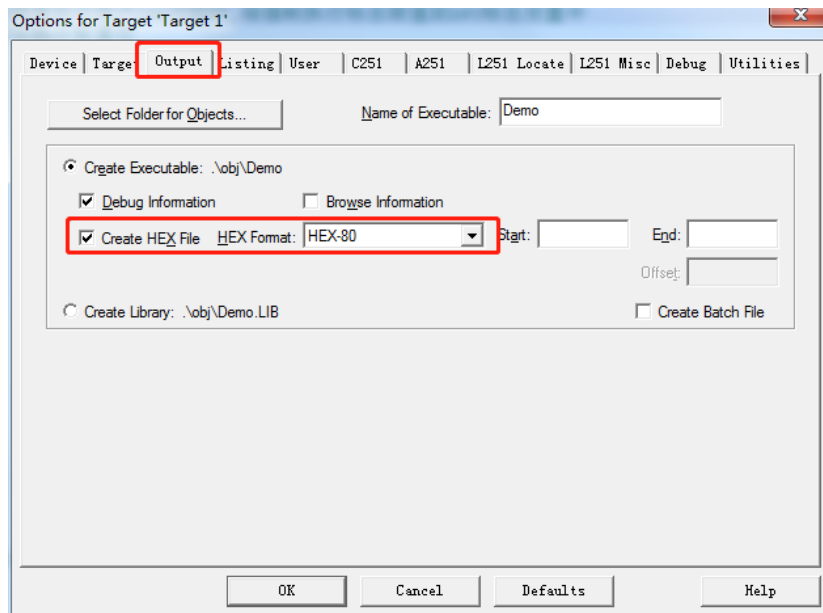
这种情况一般只会出现在第一次只单独下载了用户 ISP 代码，而没有用户 AP 代码的时候。

## ● 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 代码使用了 FF:0000H~FF:0FFFH 的 4K 空间，用户的 AP 代码必须从 FF:1000H 开始执行，用户 AP 代码原本位于 FF:0000H~FF:0002H 的复位跳转指令被重映射到 FF:1000H~FF:1002H 的地址（重映射的工作上位机应用程序会自动处理，用户在编写 AP 代码时无需关心）。另外单片机的中断入口地址也在用户 ISP 代码的 4K 空间以内，也需要重映射到 FF:1000H 开始的地方，这个重映射需要在 Keil 软件中对项目进行一些简单设置即可。

### 用户 AP 代码项目设置





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。

```
01 #include "atc32g.h"
02 #include "intrins.h"
03
04 #define FOSC           24000000UL
05 #define T1MS          (65536 - FOSC/1000)
06
07 #define DFU_TAG        0x12abcd34 //DFU强制执行标志
08 long xdata DfuFlag_at_ 0x17fc;    //DFU标志。定义在xdata的最后4字节
09
10 void sys_init();
11
12 int cnt200;
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32 {
33     if (P32 == 0)
34     {
35         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
36         IAP_CONTR = 0x20;  //然后执行软复位
37     }
38 }
39
40
41 void sys_init()
42 {
43     WST = 0x00;
44     CRCON = 0x00;
45     EAXFR = 1;
46
47     P2M0 = 0x00;
48     P2M1 = 0x00;
49     P3M0 = 0x00;
50     P3M1 = 0x00;
51
52     P3FU |= 0x04;
53
54     TMOD = 0x00;
55     T0x12 = 1;
56     T10 = T1MS;
57     TRQ = T1MS >> 8;
58     TRQ = 1;
59     ETO = 1;
60
61     DfuFlag = 0; //上电正常执行用户AP,时需要将dfu标志清零
62     cnt200 = 0;
63
64     EA = 1;
65 }
66
67
```

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。

## 四. 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);”

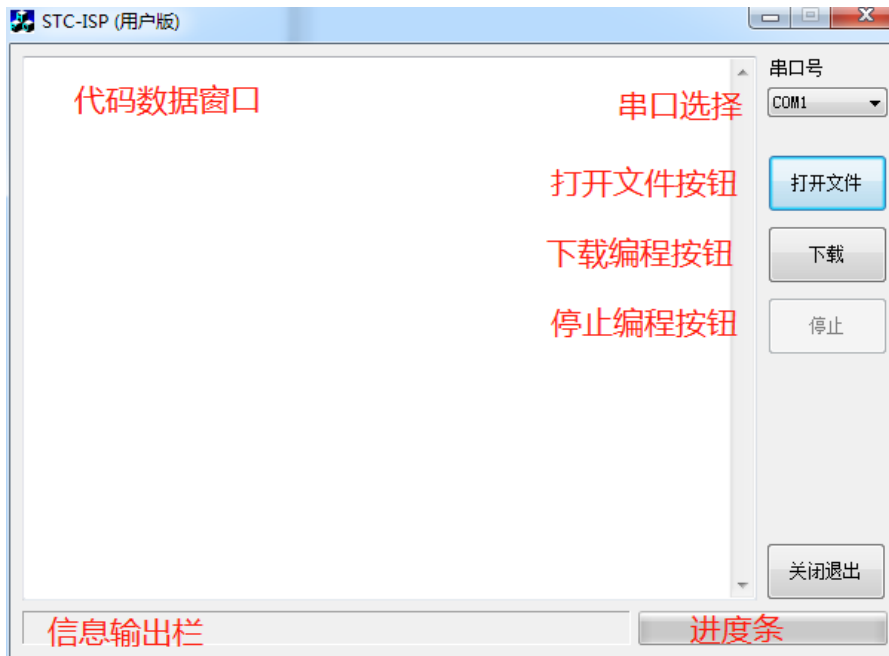
```
////////////////////////////////////  
// CStcIsp_UserDlg dialog  
  
class CStcIsp_UserDlg : public CDialog  
{  
// Construction  
public:  
    CStcIsp_UserDlg(CWnd* pParent = NULL); // standard constructor  
    virtual ~CStcIsp_UserDlg();  
  
    void ShowMessage(LPCTSTR lpMsg, ...);  
    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。



## 五. 上位机应用程序的使用方法

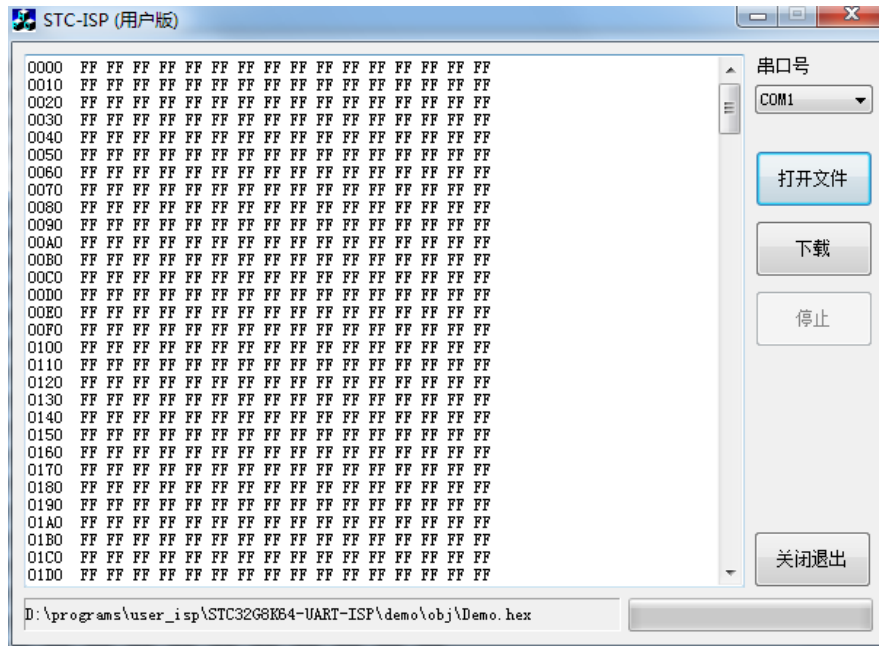
- 1、打开上位机界面，如下图



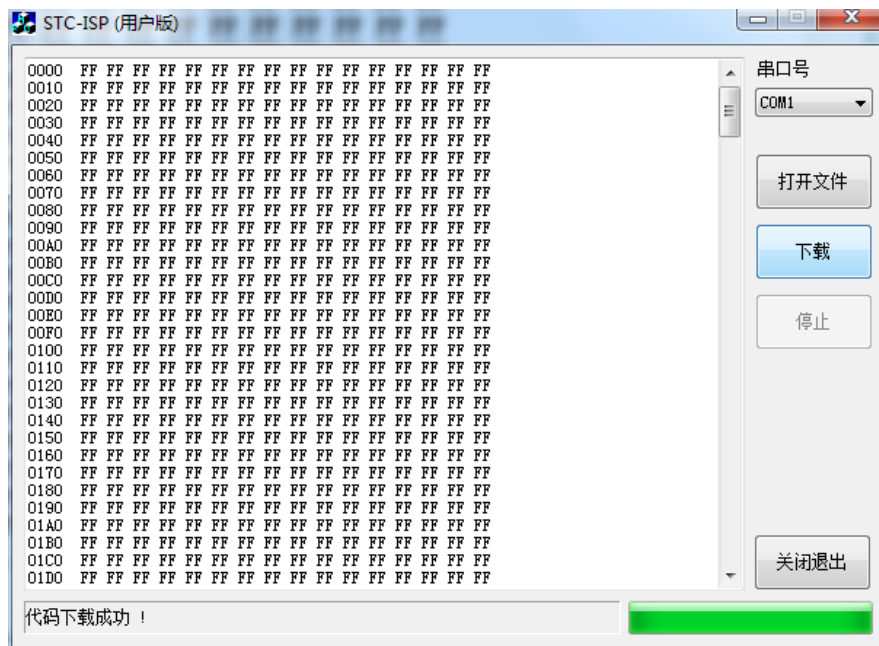
- 2、点击“打开文件”按钮来打开要下载的源数据文件，Bin 或者 Intel hex 格式均可以。注意用户 AP 代码项目一定要按照前面所介绍的方式进行建立和设置，否则格式不符合规范就会弹出如下弹窗。



打开格式规范的代码文件，如下图所示：

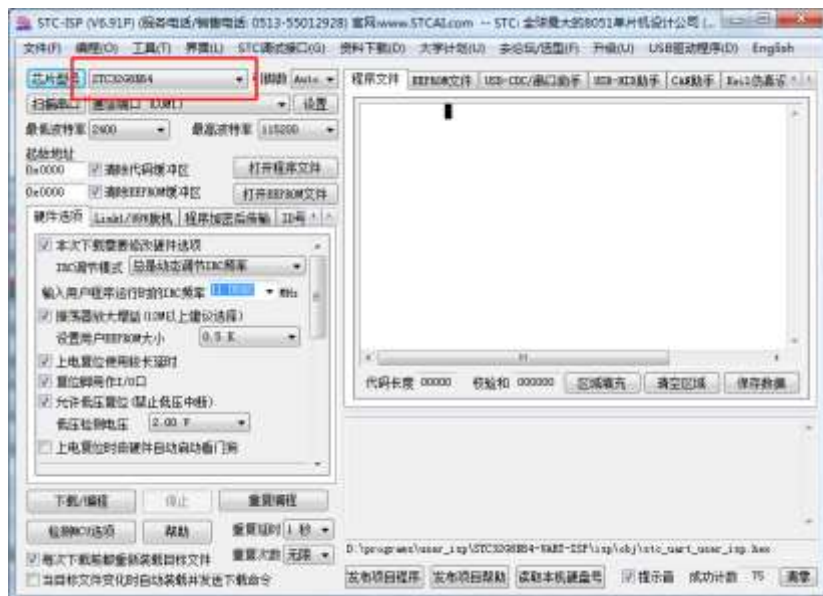


3、点击“下载”按钮即可开始下载数据

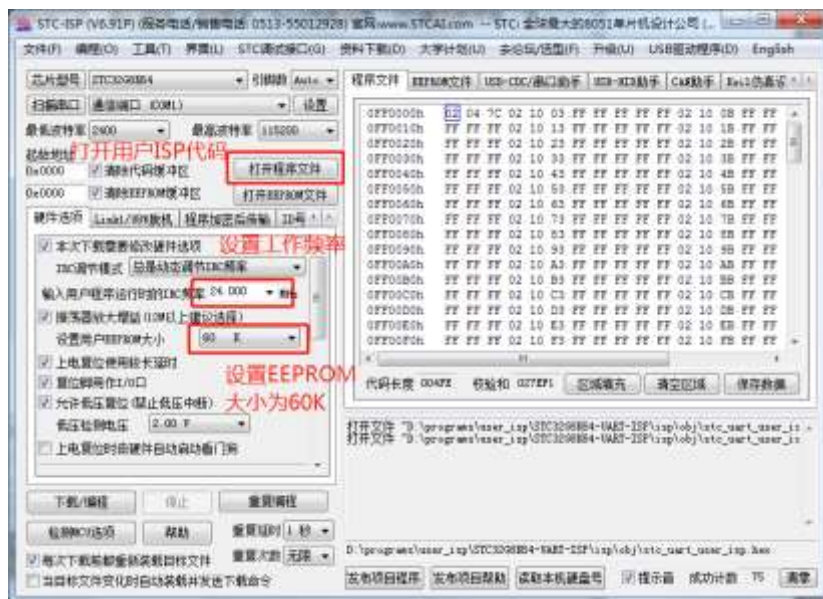


## 六. 整体操作流程

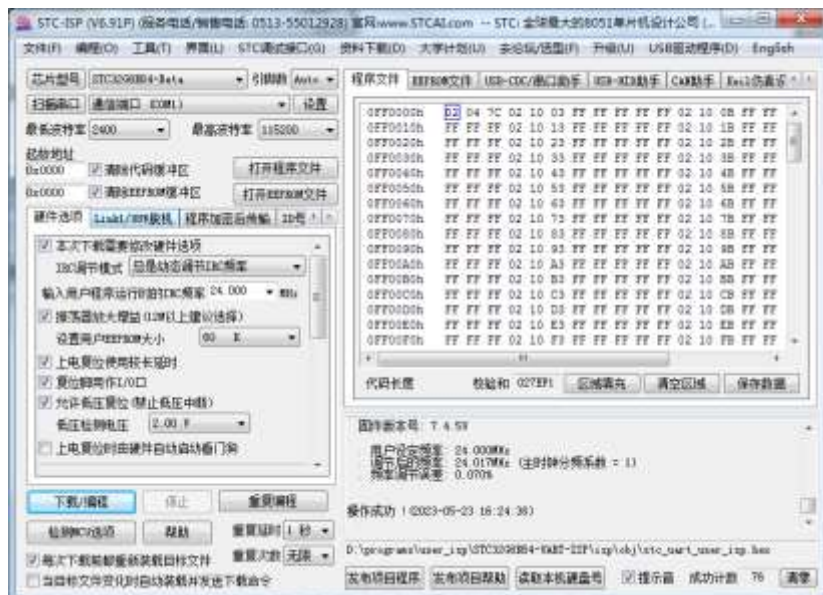
- 1、拿到一片全新的 STC32G8K64 的单片机时，首先需要参考 STC32G 的数据手册中的串口 ISP 下载参考线路图，将单片机和电脑的串口口连接好。
- 2、打开最新的 STC-ISP 下载软件，选择“STC32G8K64”目标单片机型号。



- 3、打开范例程序包中的 “\isp\obj\stc\_uart\_user\_isp.hex” 用户 ISP 代码 hex 文件，并按照如下图所示设置硬件选项，工作频率为 24MHz、EEPROM 大小为 60K（此项很重要，用户 ISP 的 4K 需要保护起来）



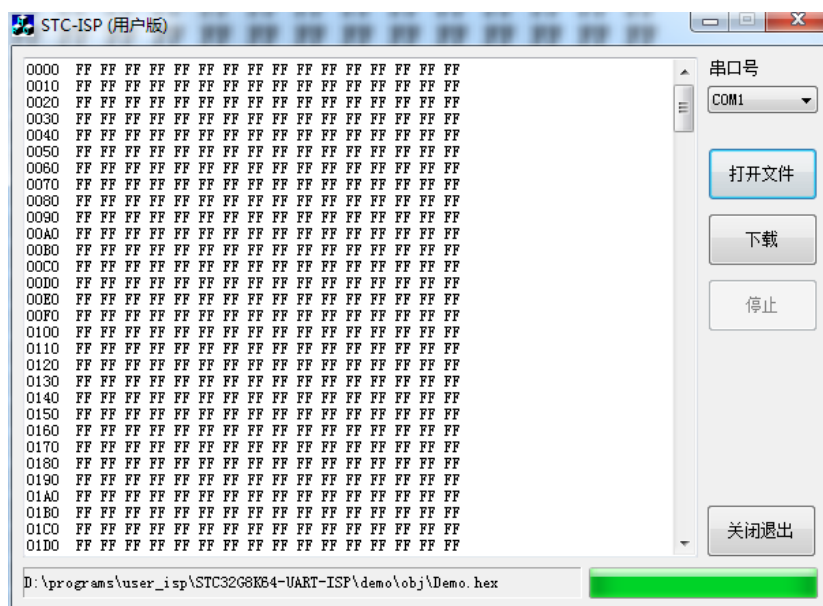
4、 点击下载/编程按钮，将用户 ISP 代码下载到目标单片机中



5、 给目标单片机重新断电并上电一次，设置的 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）

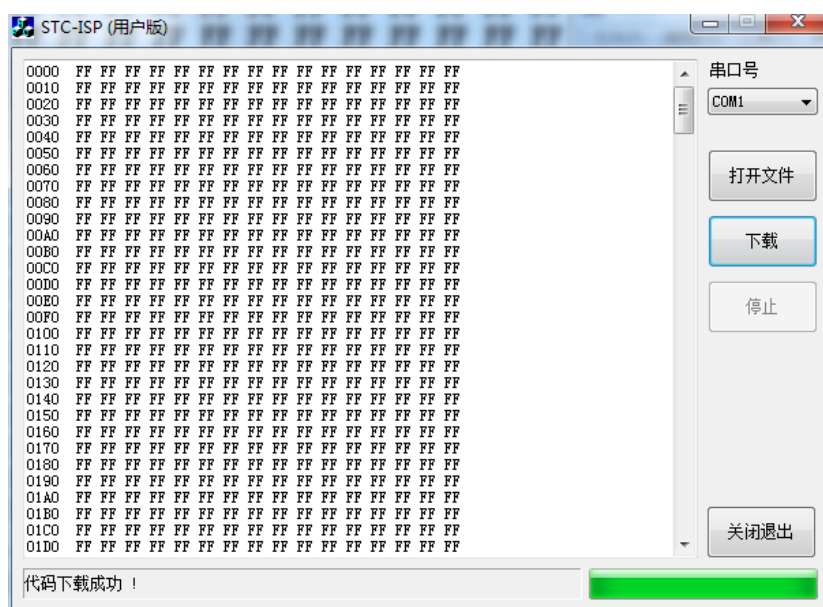
6、 经过第 5 步的重新上电后，由于此时还没有用户 AP 代码，用户 ISP 代码被强制运行。

7、 打开范例程序包中的“\app\Release\StcIsp\_User.exe”，点击“打开文件”按钮，打开范例程序包中的“\demo\obj\Demo.hex”



8、 点击“下载”按钮，即可完成用户 AP 代码的更新

下载完成后，会显示如下图所示的“代码下载成功”画面



9、 若在后续的开发中，由于已经存在用户 AP 代码，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，可以将 P3.3 口通过 1K 电阻连接到 GND，再给目标芯片重新上电，然后回到第 8 步，即可继续使用用户 ISP 更新用户 AP 代码了。