

深圳大学

实 验 项 目 :入门单片机编写程序
主 讲 人 :-
日 期 :2025 年 2 月 24 日

目录

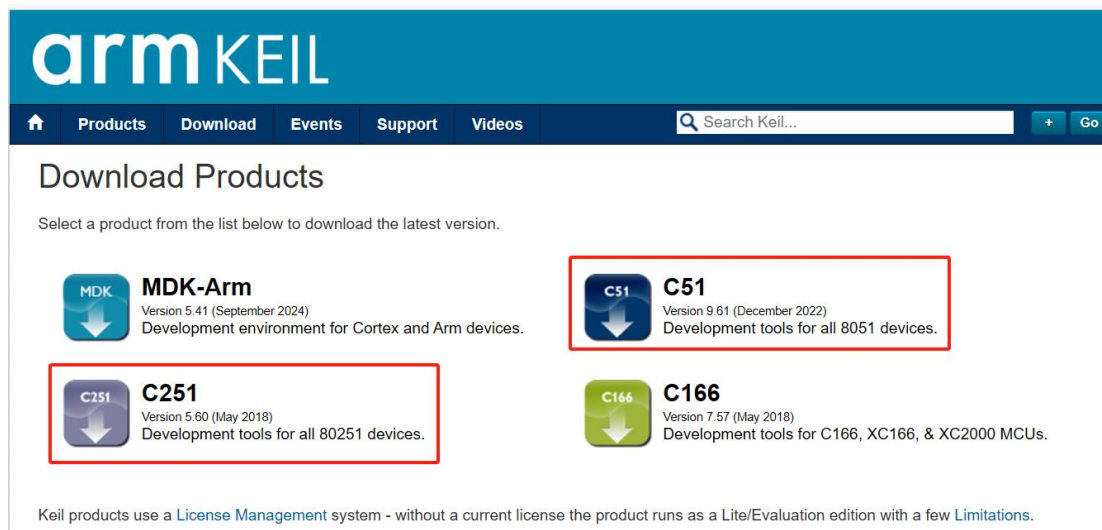
安装 Keil C51 和 C251	3
下载安装包	3
安装过程	5
下载 STC-ISP 软件	7
添加 STC 型号和头文件	8
和谐 Keil 部分	10
打开 Keil Keygen	10
管理员身份打开 Keil	10
打开许可证界面进行激活	10
建立一个新工程	12
新建一个项目文件夹	12
新建工程	12
选择单片机	12
设置 4Byte 字节对齐	13
设置 Xsmall 模式	14
使用 REMOVEUNUSED 功能	14
选择输出 HEX 文件功能	15
点亮 LED 灯	16
添加头文件	16
原理介绍	16
代码部分	17
下载用户程序到实验箱	19
连接到实验箱	19
硬件 USB 下载用户程序步骤	19
装载程序文件	20
下载代码	21
下载后的效果	22
让 LED 闪烁起来	23
通过 STC-ISP 的小工具辅助生成代码	23
添加头文件	23
代码部分	23
下载后的效果	24
通过 USB 控制 AI8051U 进行 LED 点灯	24
下载 USB 库文件	24
CDC 和 HID 的区别	25
使用 USB-CDC 部分	25
代码部分	26
下载后的操作	27
如何设置多字符串发送?	28
使用 HID 操作方式	29
实验箱原理图	31
下载链接	31

安装 Keil C51 和 C251

下载安装包

首先登录 Keil 官网，下载最新版的 C51 和 C251 安装包

手动在浏览器的地址栏输入以下网址：<https://www.keil.com/download/product/>



进入首页，我们可以看到 C51 和 C251 的下载链接，这里将这两个安装包都下载到桌面上

armKEIL

Home

Products

Download

Events

Support

Videos

Search Keil...

+

Go

Product Information

Software & Hardware Products

Arm Development Tools

C166 Development Tools

C51 Development Tools

C251 Development Tools

Debug Adapters

Evaluation Boards

Product Brochures

Newsletters

Device Database®

Device List

Compliance Testing

ISO/ANSI Compliance

Validation and Verification

Distributors

Overview

Home / Product Downloads

C51

Development tools for Classic and Extended 8051 Microcontrollers

Version 9.61

MD5: 8f0de26cd2613974defd3dd9aedc36ee

MDK-ARM v5.xx is part of Keil MDK v6

Complete the following form to download the Keil software development tools.

Enter Your Contact Information Below

First Name: x

Last Name: y

E-mail: x@qq.com

Company: w

Job Title: 1

Country/Region: China

State/Province: -- United States --

Phone: 1111111111

☐ Send me e-mail when there is a new update.

NOTICE:
If you select this check box, you **will** receive an e-mail message from Keil whenever a new update is available. If you don't wish to receive an e-mail notification, don't check this box.

Which device are you using?
(eg, STM32)

Arm will process your information in accordance with the Evaluation section of our [Privacy Policy](#).

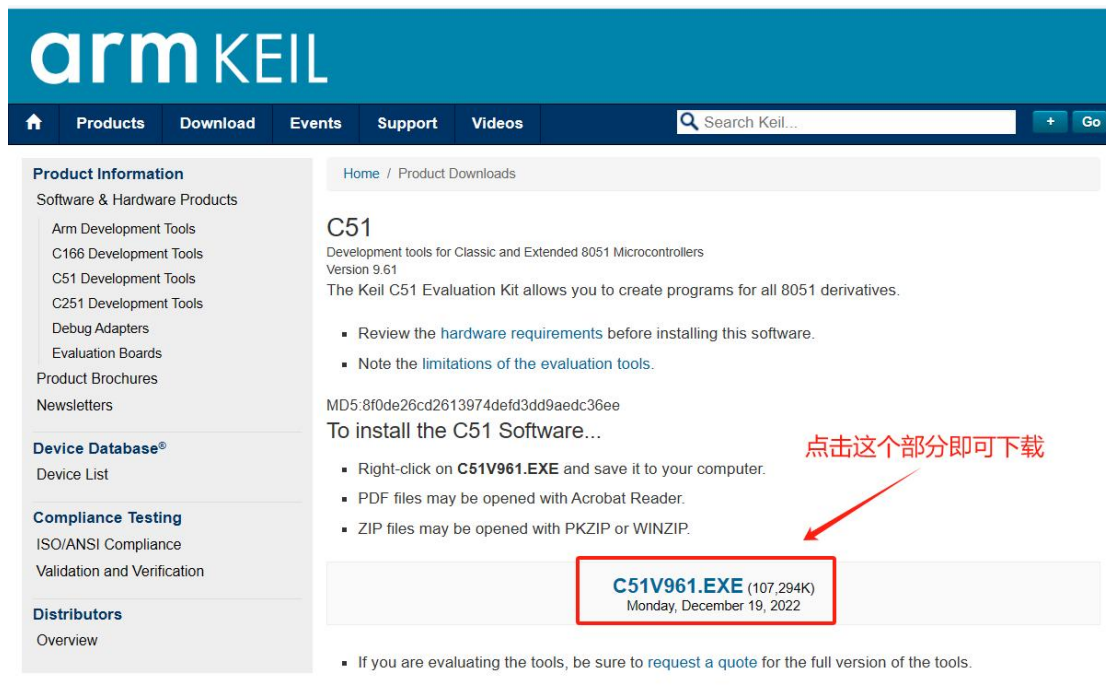
☒ Please keep me updated on products, services and other relevant offerings from Arm. You can change your mind and unsubscribe at any time.

Submit

Reset

这里以 C51 作为示例，点击进入后，会让填入信息，这里随便填一些就可以了，然后点击 Submit

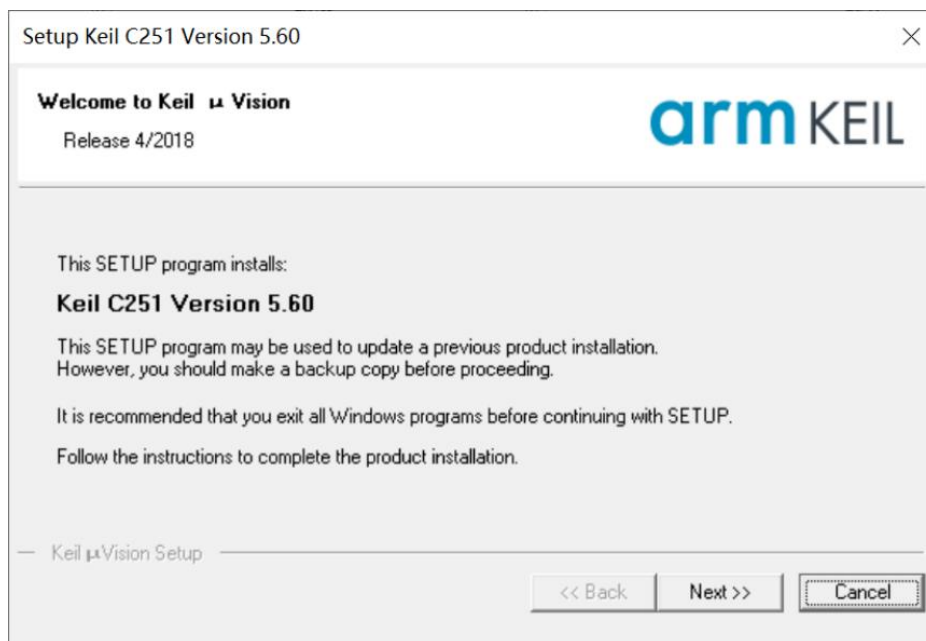
4



点击完后，就可以出现 C51V961.EXE，点击粗体蓝色字即可下载 C51 安装包。C251 部分操作同上
安装过程

双击下载的安装包开始安装，。

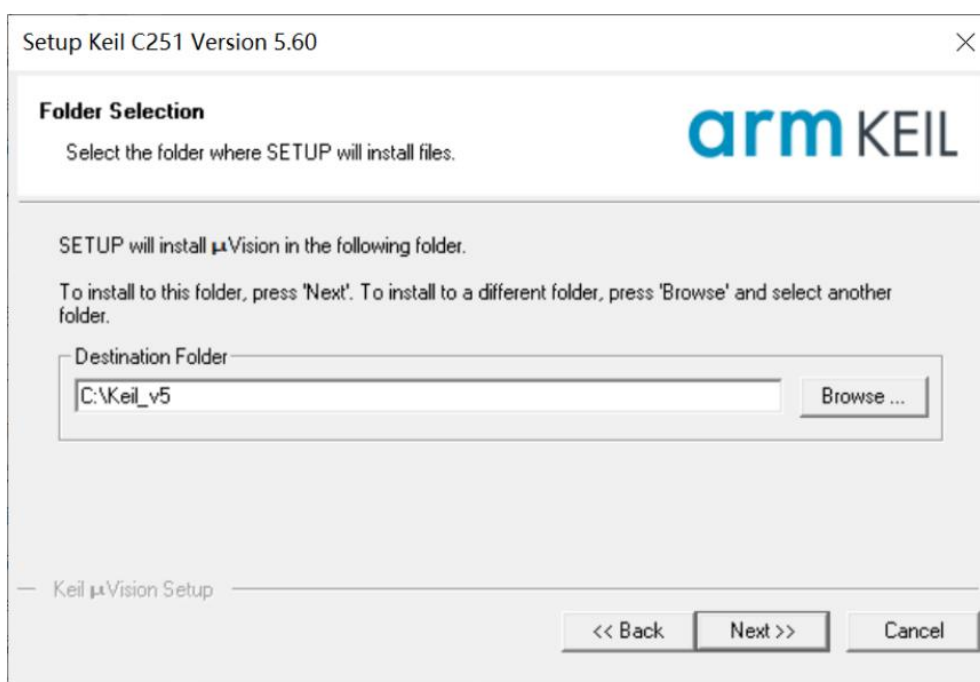
点击“Next”（这里以 C251 安装为例，安装完 C251 后，再继续安装 C51 即可）



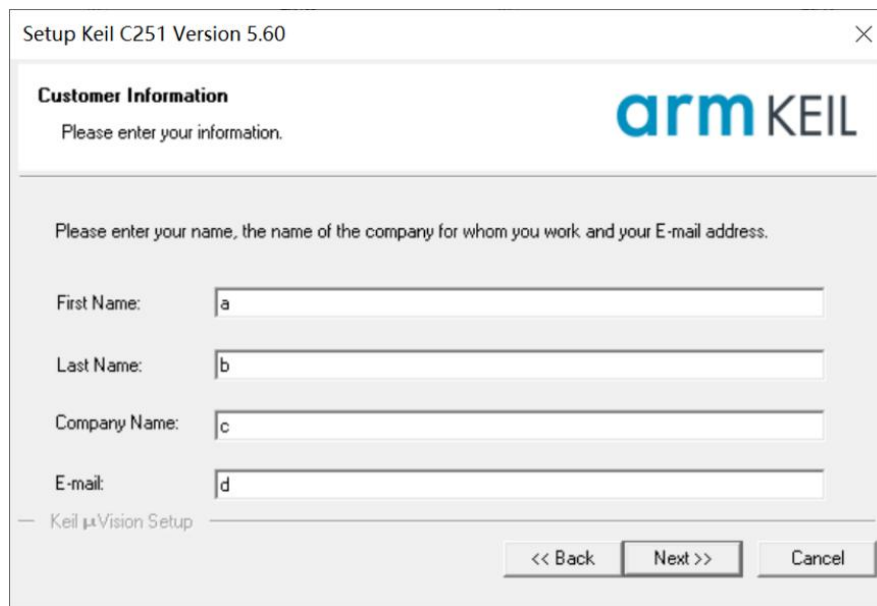
勾选“I agree to all the terms of the preceding License Agreement”，然后点击“Next”：



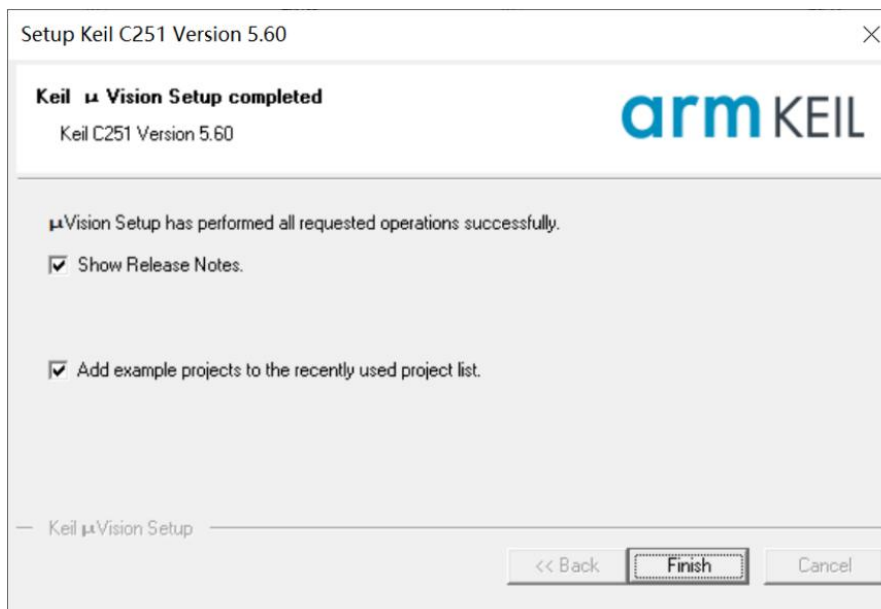
选择安装目录，然后点击“Next”：



填写个人信息，然后点击“Next”：



安装完成，点击“Finish”结束。



下载 STC-ISP 软件

STC-ISP 的下载链接是：<https://www.stcai.com/gjrrj>



点击软件下载即可下载最新版本的 STC-ISP

添加 STC 型号和头文件

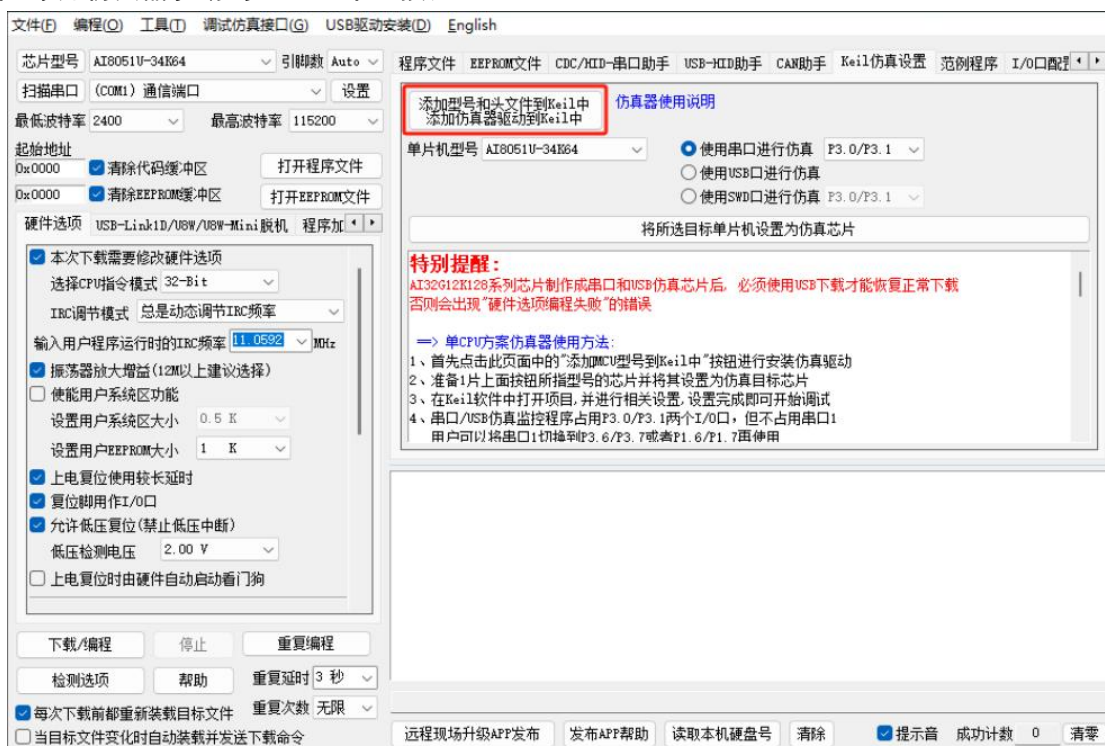
安装完成后，我们还需要添加 STC 型号和头文件

因为这时候，Keil 内还是没有 STC 单片机部分的型号及头文件。

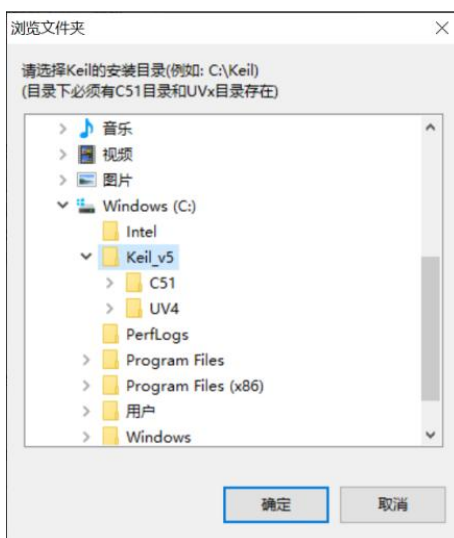
所以，我们通过 STC-ISP 进行添加

使用 Keil 之前需要先安装 仿真驱动。仿真驱动的安装步骤如下：

首先打开 ISP 下载软件，然后在软件右边功能区的“Keil 仿真设置”页面中点击“添加型号和头文件到 Keil 中 添加仿真器驱动到 Keil 中”按钮：



按下后会出现如下画面：



将目录定位到 Keil 软件的安装目录，然后确定。安装成功后会弹出如下的提示框：



即表示驱动正确安装了

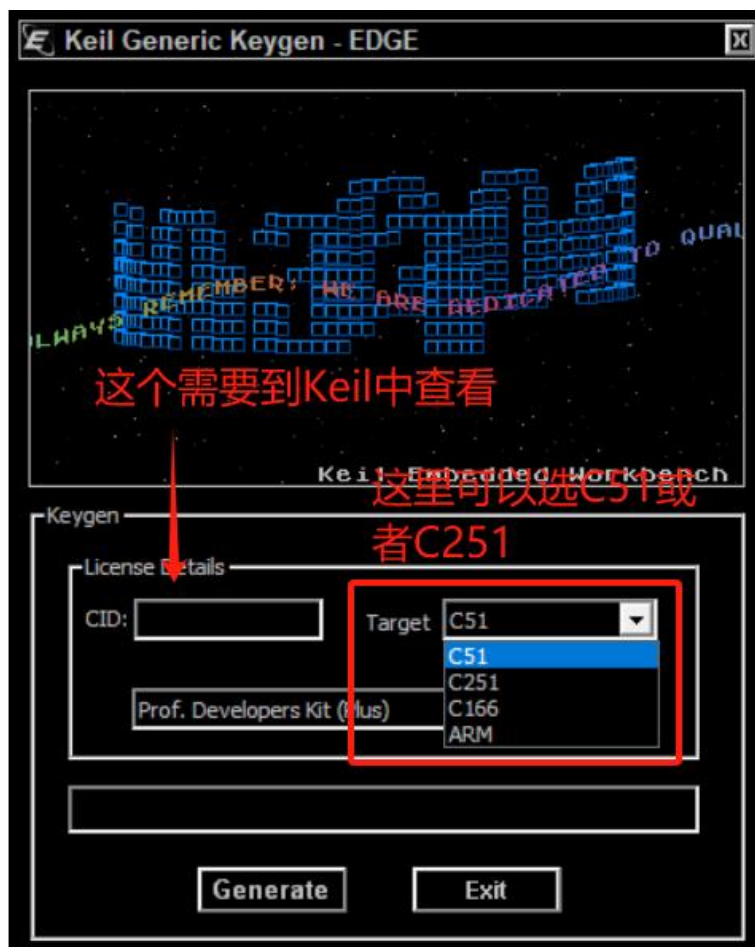
头文件默认复制到 Keil 安装目录下的“C251\INC\AI”目录中

在 C 代码中使用“`#include <Ai8051U.H>`”或者“`#include "Ai8051U.H"`”进行包含均可正确使用

和谐 Keil 部分

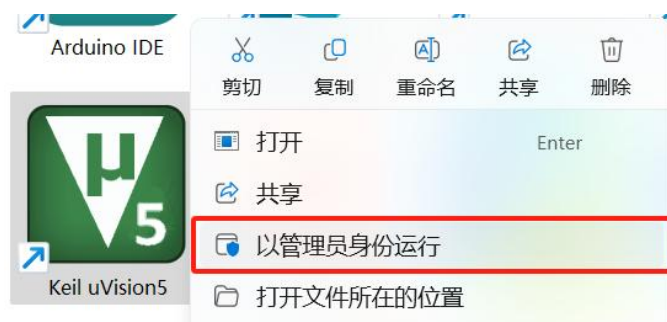
打开 Keil Keygen

这个软件可以从向老师或者学长要一下



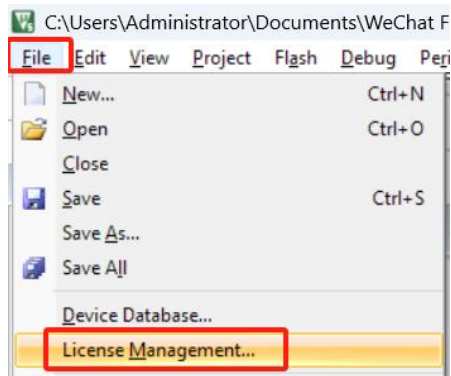
管理员身份打开 Keil

然后，右键使用管理员身份打开 Keil，否则后面会因为权限不够导致注册失败

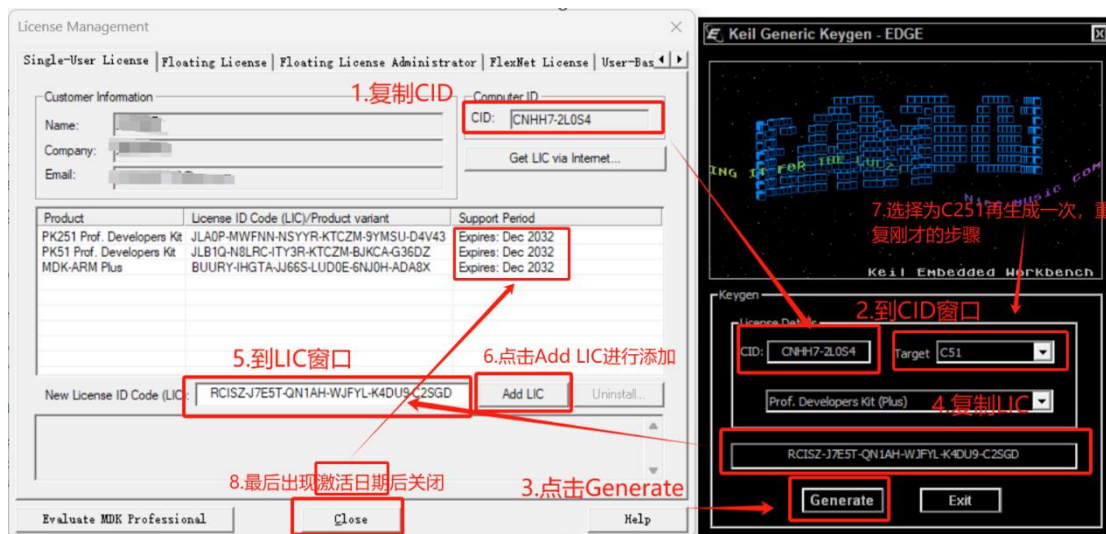


打开许可证界面进行激活

打开 Keil 后，点击 File 里面的 Lincese Management



然后按照如下图的序号顺序进行操作，复制 CID 并得到 LIC 进行激活。
 需要注意 C51 和 C251 都要激活。



C251 和 C51 和谐完成后，点击 close，关闭并且重新打开 Keil 就可以使用了

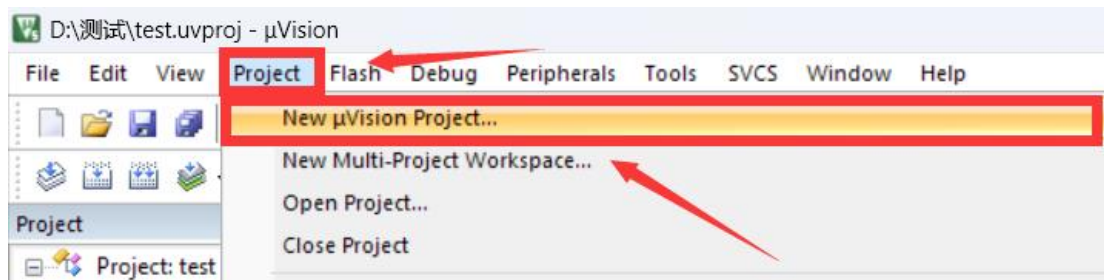
建立一个新工程

新建一个项目文件夹

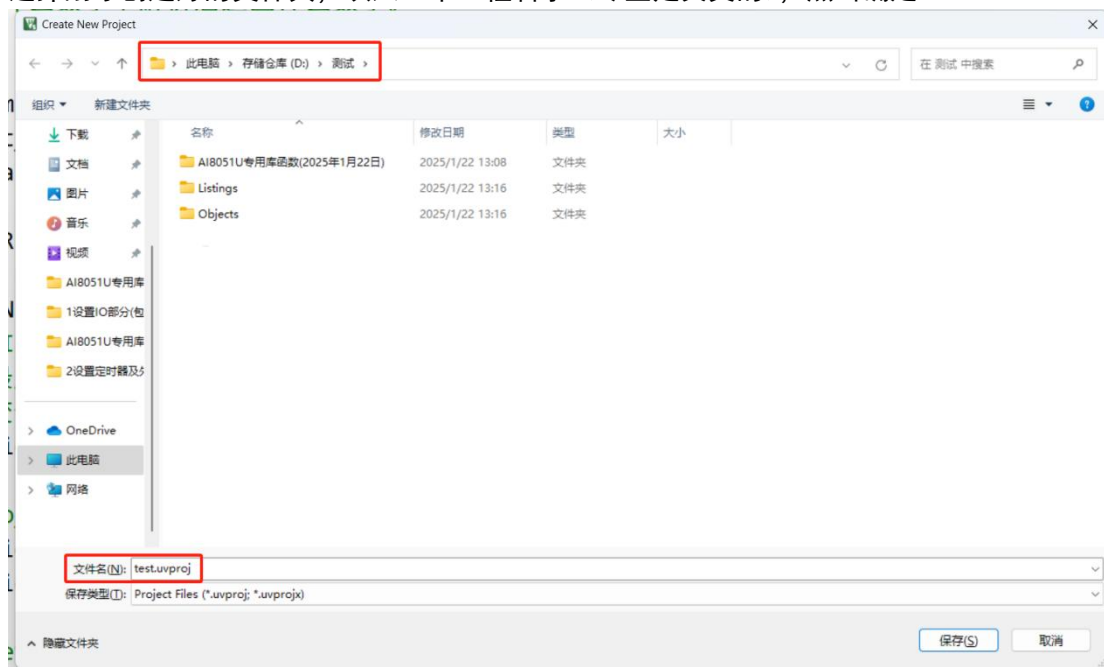
在一个纯英文目录下（例如 D 盘根目录）新建一个文件夹，用于存放项目的文件。不直接在桌面创建是因为路径有可能包含中文。

新建工程

打开 Keil 软件，在上方的‘Project’选项中，找到 New uVision Project 选项，进行新建工程。

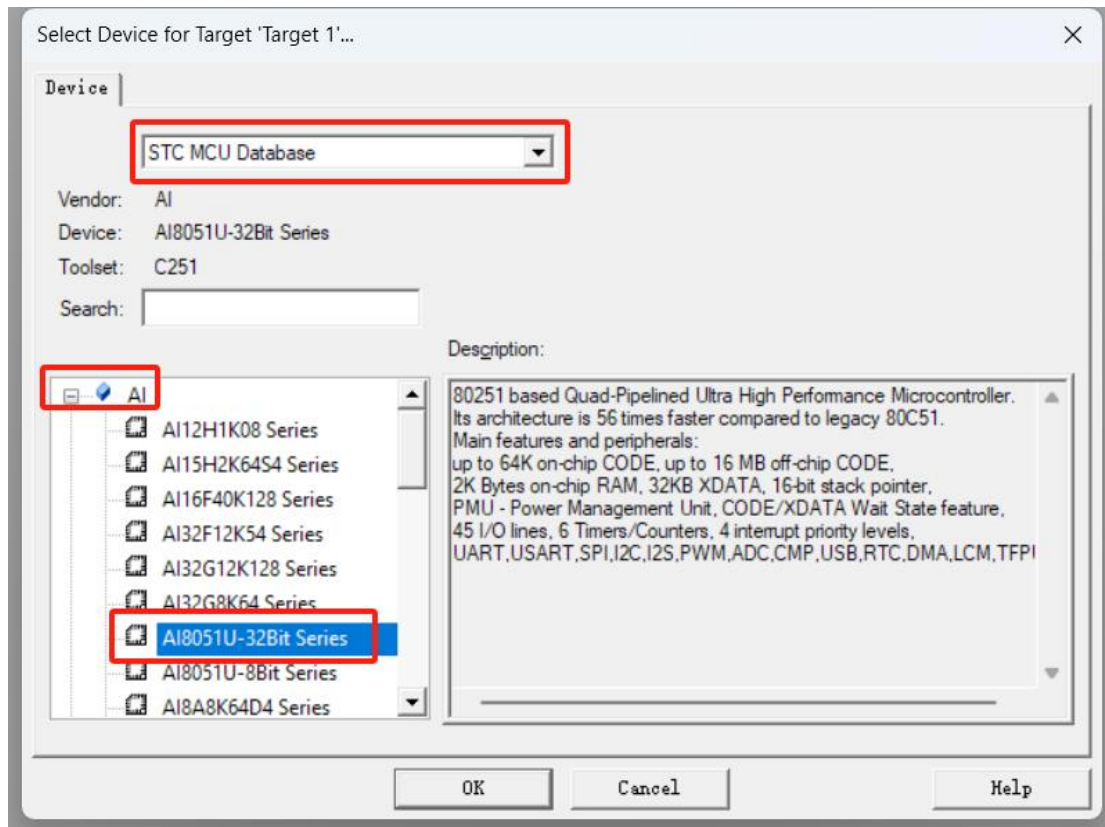


选择刚才创建好的文件夹，填入一个工程名字（尽量是英文的），点击确定。



选择单片机

在上方的 Database 选项中选择‘STC MCU Database’，然后展开 AI 选项，找到其中的‘AI8051U-32Bit’选项点击，最后点击‘OK’以完成单片机选择。

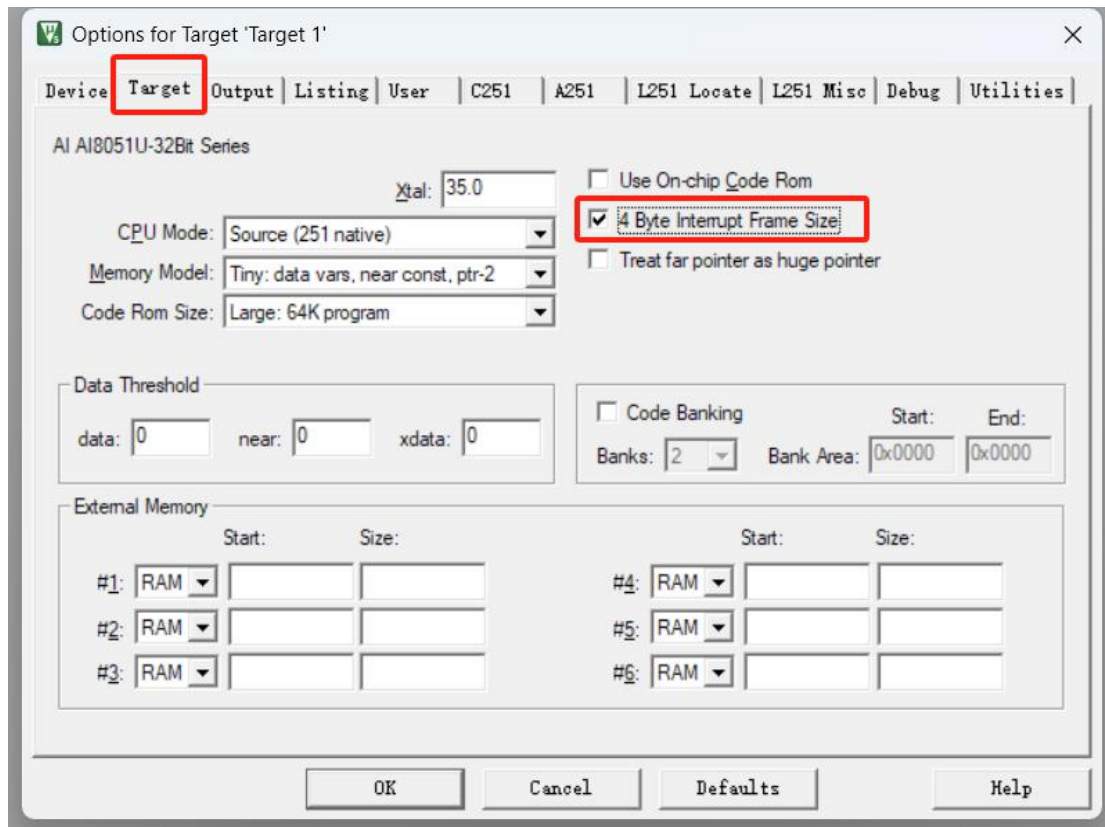


设置 4Byte 字节对齐

选择完单片机后，项目是空白的。此时需要设置一下工程的选项。

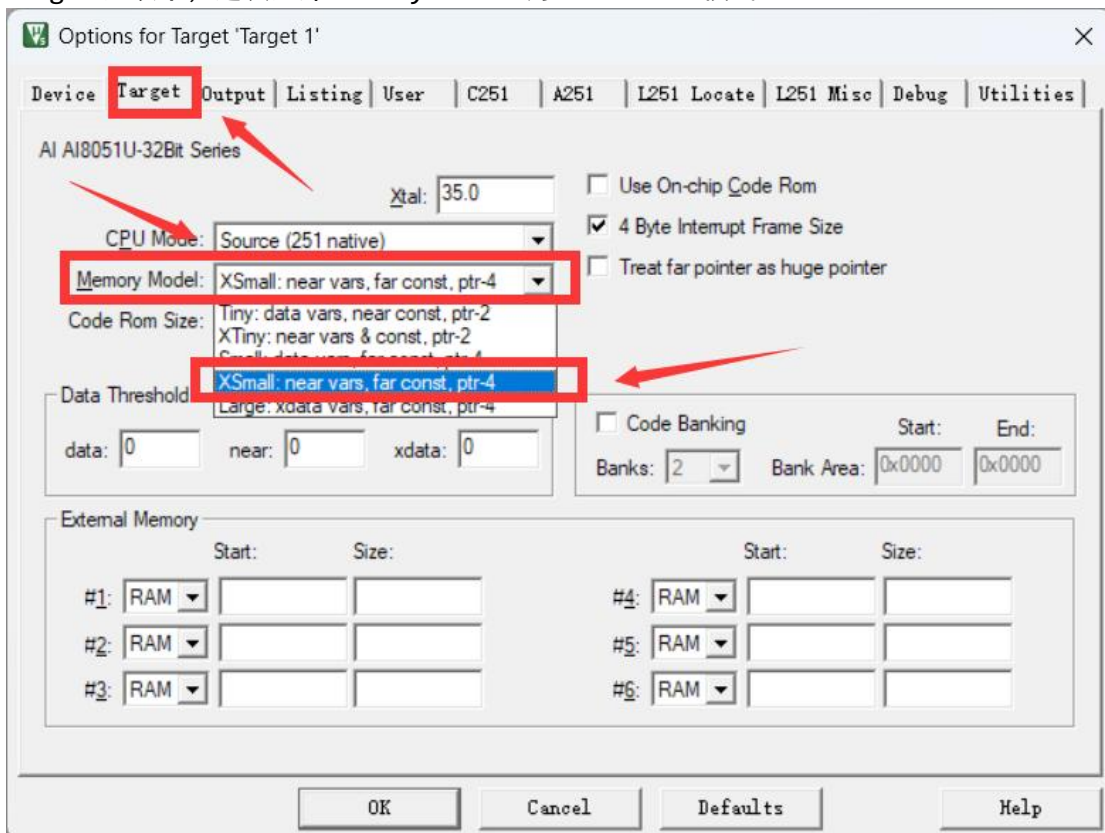
在左侧的 Project 选项卡中，右键 Target 1 文件夹图标，然后选择 'Options for Target 'Target 1'...' 选项，进入后默认就是 Target 选项卡，如图所示点击 '4Byte Interrupt Frame Size' 选项，以规范程序内部调用指针时候的对齐，防止出现奇怪的问题。





设置 Xsmall 模式

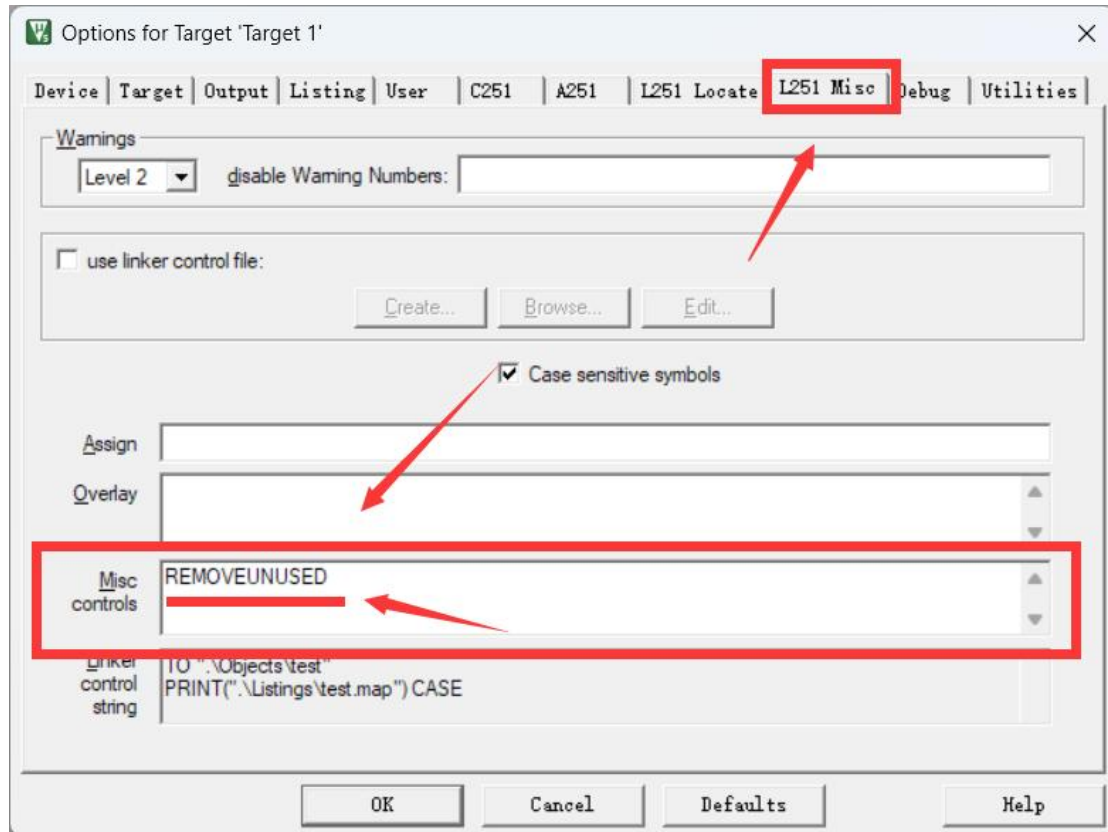
还是在 Target 选项卡，这次选择 Memory Model 为 'XSmall' 模式



使用 REMOVEUNUSED 功能

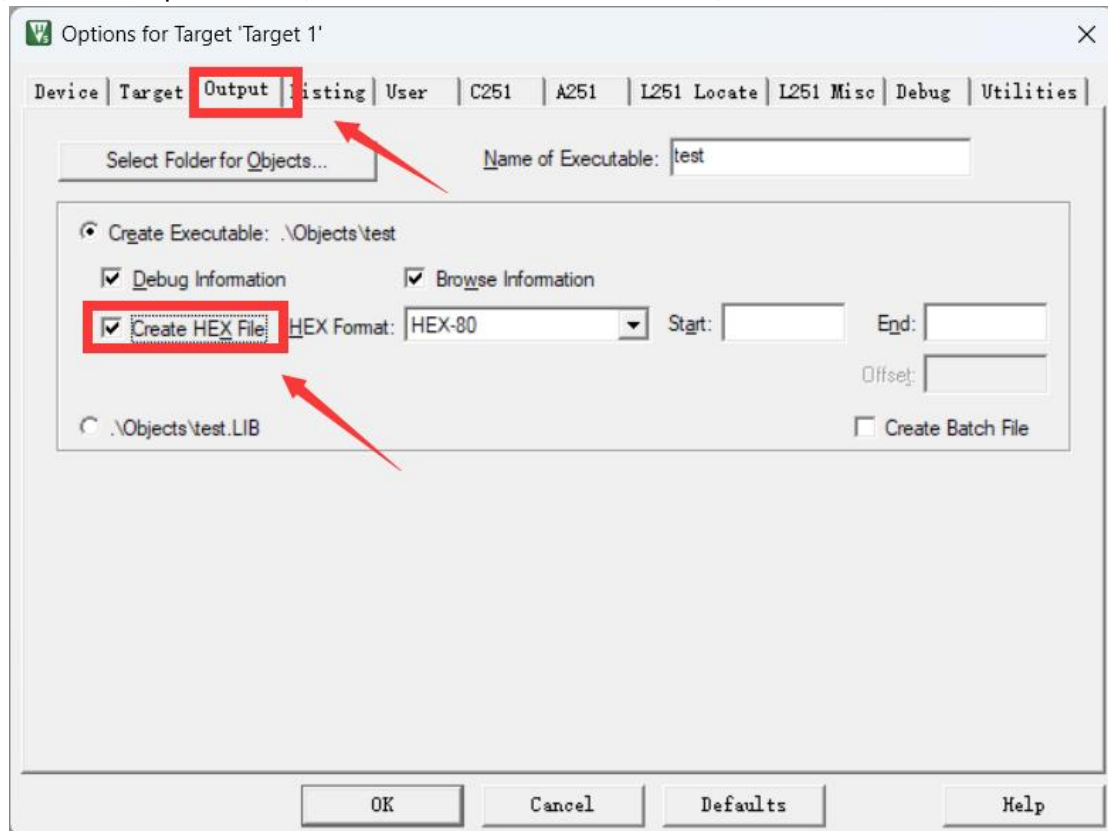
接下来切换到 L251 Misc 选项卡，在 Misc controls 输入框中，输入大写的

‘REMOVEUNUSED’，以实现不使用的函数不进行调用。



选择输出 HEX 文件功能

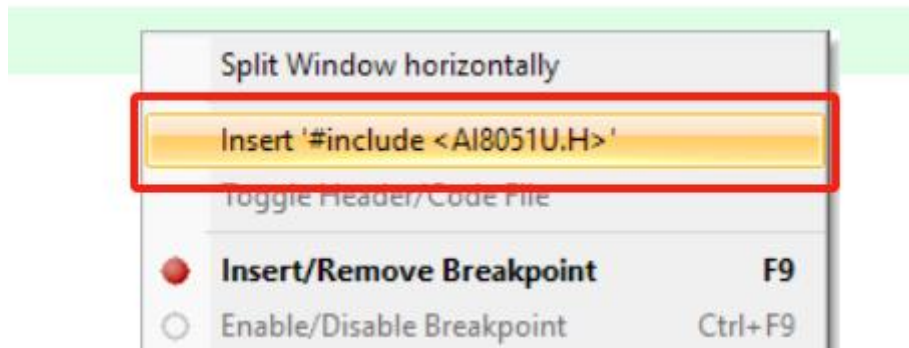
接下来，切换到 Output 选项卡，勾选 ‘Creat HEX File’ 选项，否则编译后是不会创建 HEX 文件的。



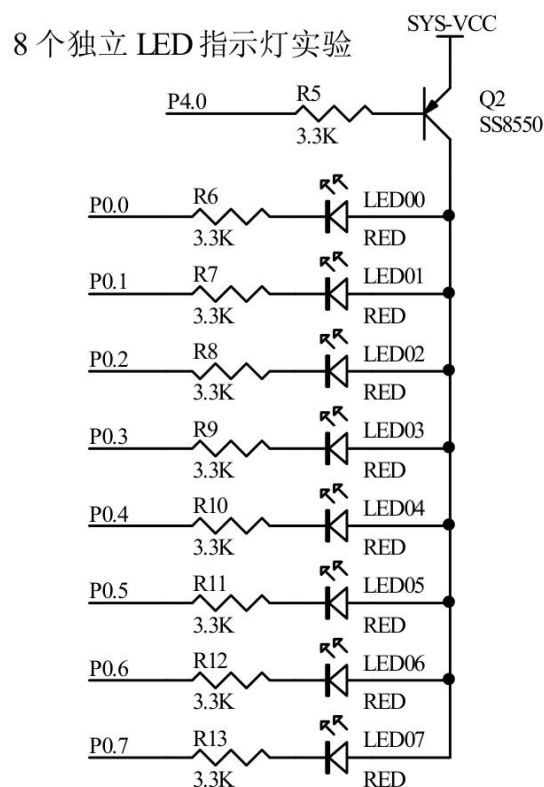
点亮 LED 灯

添加头文件

首先是添加头文件，通过右键添加即可



原理介绍



实验箱内，所有的 LED 通过一个 PNP 三极管控制总的通断，属于共阳极控制。如上图所示，给 P40 接口一个低电平后，给对应 P0 的端口低电平即可控制对应的 LED 亮起来

（注：P4.0 和 P40 是不同的叫法，但是在程序中只能使用 P40 这种方式，因为 C 语言中点有其他用处）。

[<实验箱的全部原理图可以通过点击这段文字进行查看>](#)

代码部分

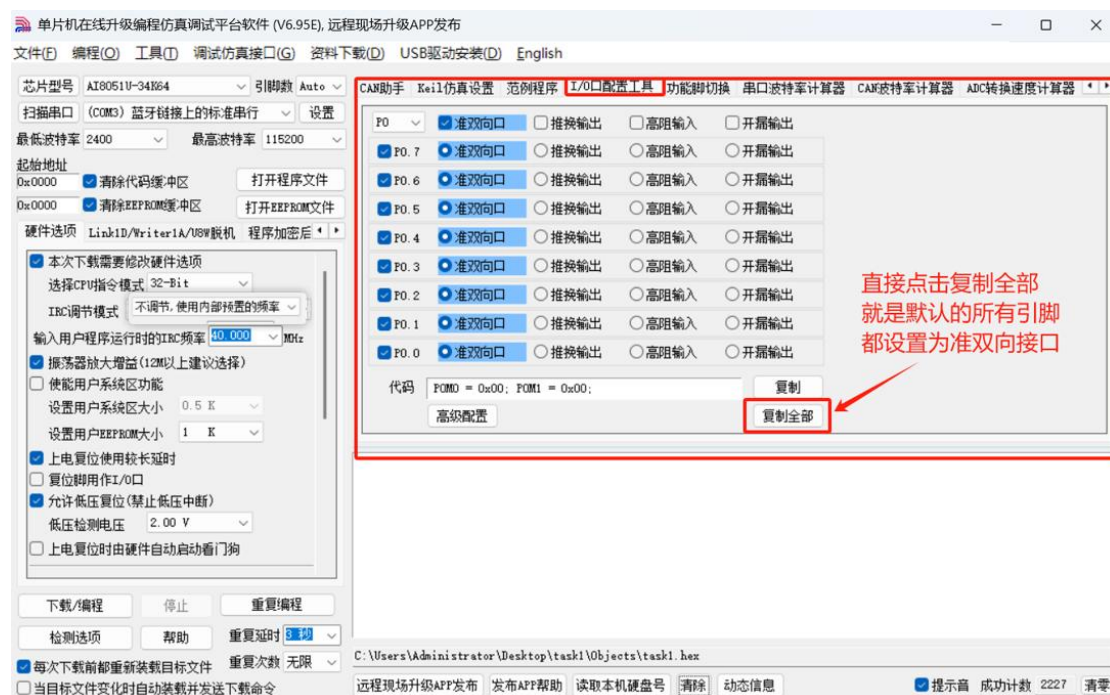
首先，我们先来写一个点亮 LED 的程序

```
#include <AI8051U.H> //包含 AI8051U 的头文件
```

```
void main(void)
{
    EAXFR = 1; //允许访问扩展的特殊寄存器, XFR
    WTST = 0;    //设置取程序代码不等待, 程序以最快速度运行
    CKCON = 0;    //设置访问片内扩展的 xdata 部分速度为最快, 不等待
    P0M0 = 0x00; P0M1 = 0x00;
    P1M0 = 0x00; P1M1 = 0x00;
    P2M0 = 0x00; P2M1 = 0x00;
    P3M0 = 0x00; P3M1 = 0x00;
    P4M0 = 0x00; P4M1 = 0x00;
    P5M0 = 0x00; P5M1 = 0x00;
    P6M0 = 0x00; P6M1 = 0x00;
    P7M0 = 0x00; P7M1 = 0x00;

    P40 = 0; //打开 LED 部分的供电
    while(1)
    {
        //LED 点亮部分
        P0 = 0; //对 P0 端口置 0, 点亮整组 LED
        //P00 = 0; //对单个端口置 0, 点亮单个 LED, 这个点亮的是 LED0
        //P01 = 0; //对单个端口置 0, 点亮单个 LED, 这个点亮的是 LED1
    }
}
```

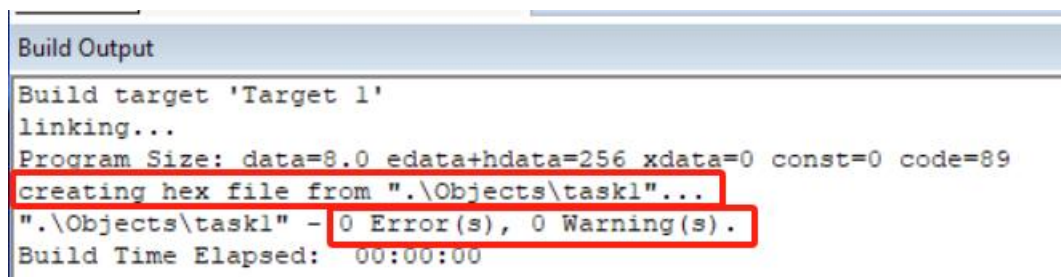
这里面, I/O 口的初始化部分使用 STC-ISP 的 I/O 口配置工具进行生成。



编写完成代码后，通过按键盘上的 F7 或者工具栏上的 Build 按钮进行编译



编译完成后，会在底部的 Build Output 生成编译信息



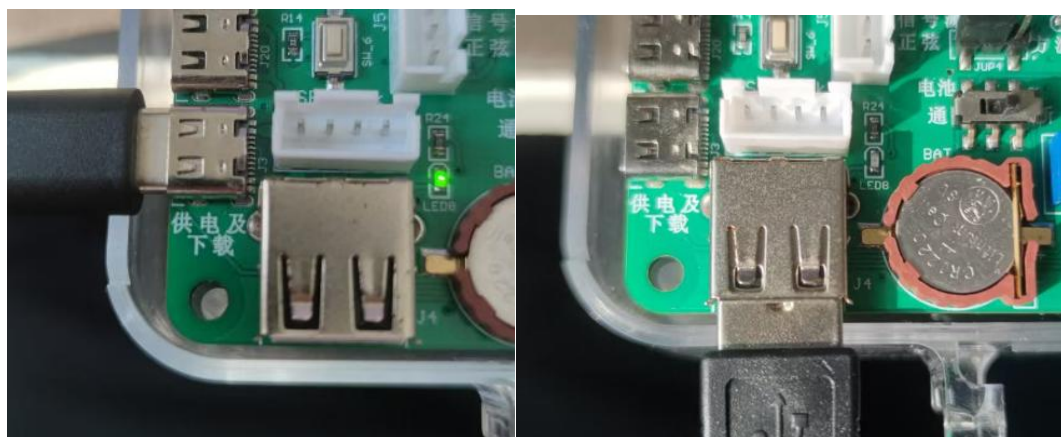
出现 creating hex file 和 0error0warning 即为编译成功，意为无报错且成功创建了 hex 文件

下载用户程序到实验箱

连接到实验箱

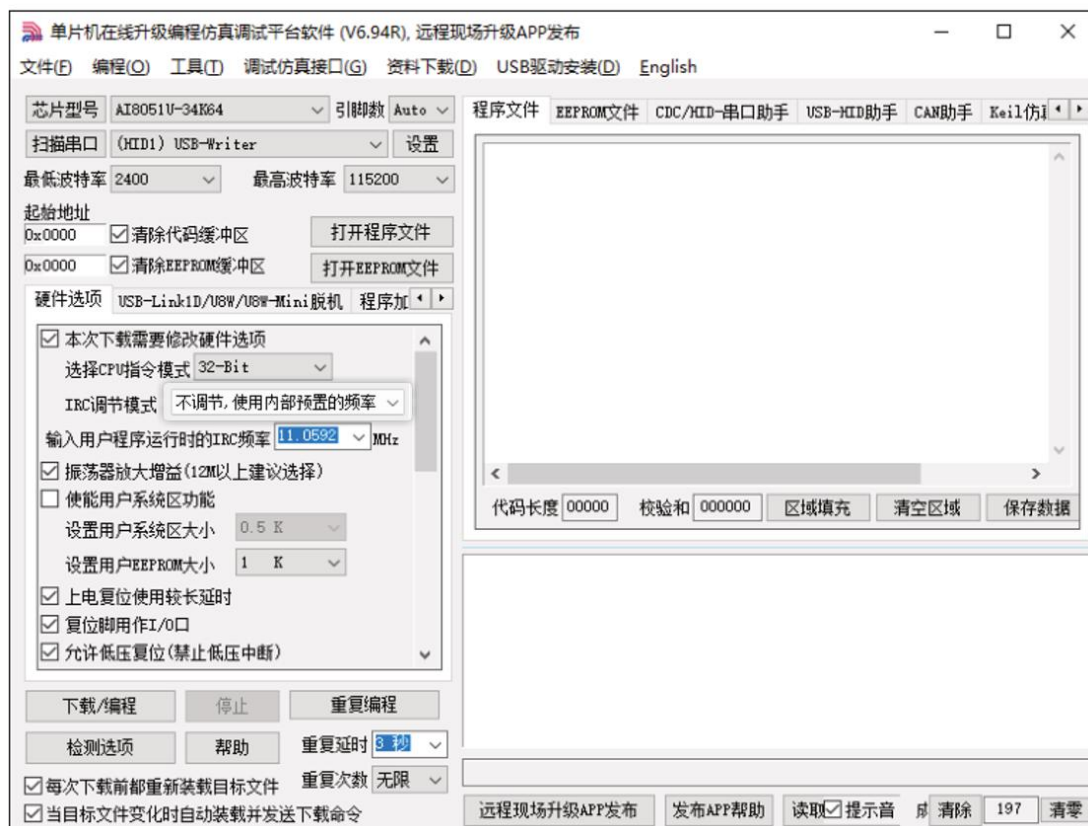
首先使用 USB 线将实验箱与电脑正确连接。

这里有两种连接方式，都是一样的：

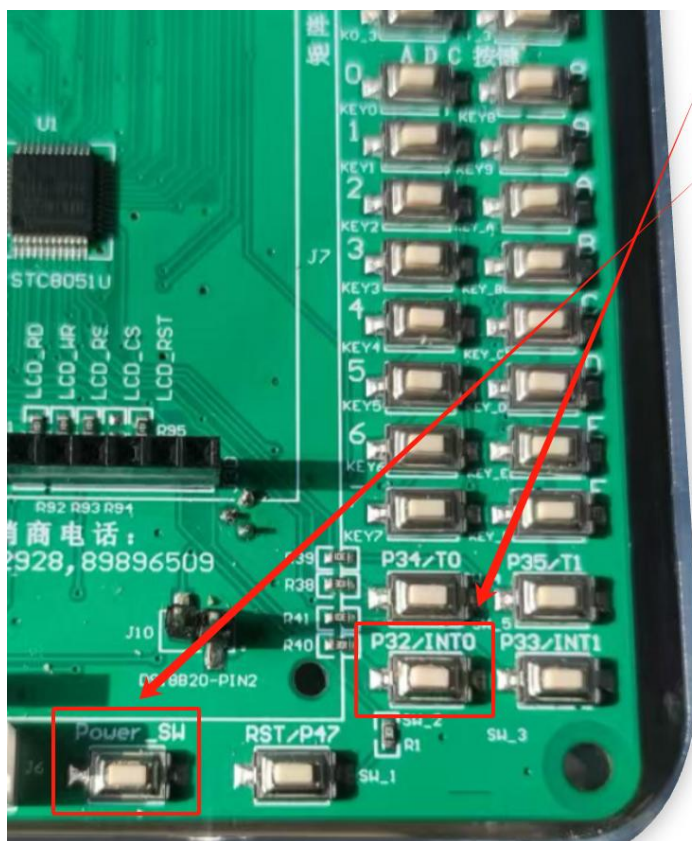


左边的图片是 TypeC 方式连接的，右边是 TypeA 方式连接的，这两个选择其中一种方式连接就可以了。

然后打开 AIapp-ISP-v6.94F 或以上版本。



硬件 USB 下载用户程序步骤



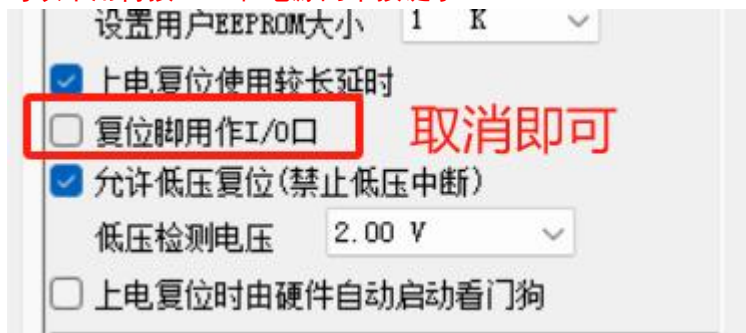
USB下载顺序:

- 1.按下P32按键，保持不动
- 2.按下Power_SW按键，此时会给单片机断电
- 3.抬起Power_SW按键
- 4.继续按住P32按键，STC-ISP端口中出现 (HID1)USB-Writer后再松开

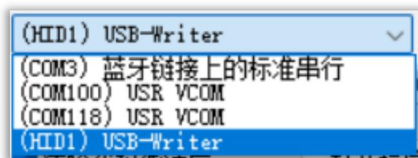


继续按住P32按键的时间不会超过2秒，如果发现按下好一会还没出现HID端口，可以检查一下连线是否出现问题，然后重新重复上述步骤

另外：如果 P4.7/nRST 已被设置为复位脚，通电状态下，外部手动复位也可进入 USB 下载
可以不用再按 P32 和电源两个按键了



出现 (HID1) USB-Writer 后，就代表可以下载了，如果没有出现，可以重复一下上述流程，然后拉开下拉框找一找看看有没有。

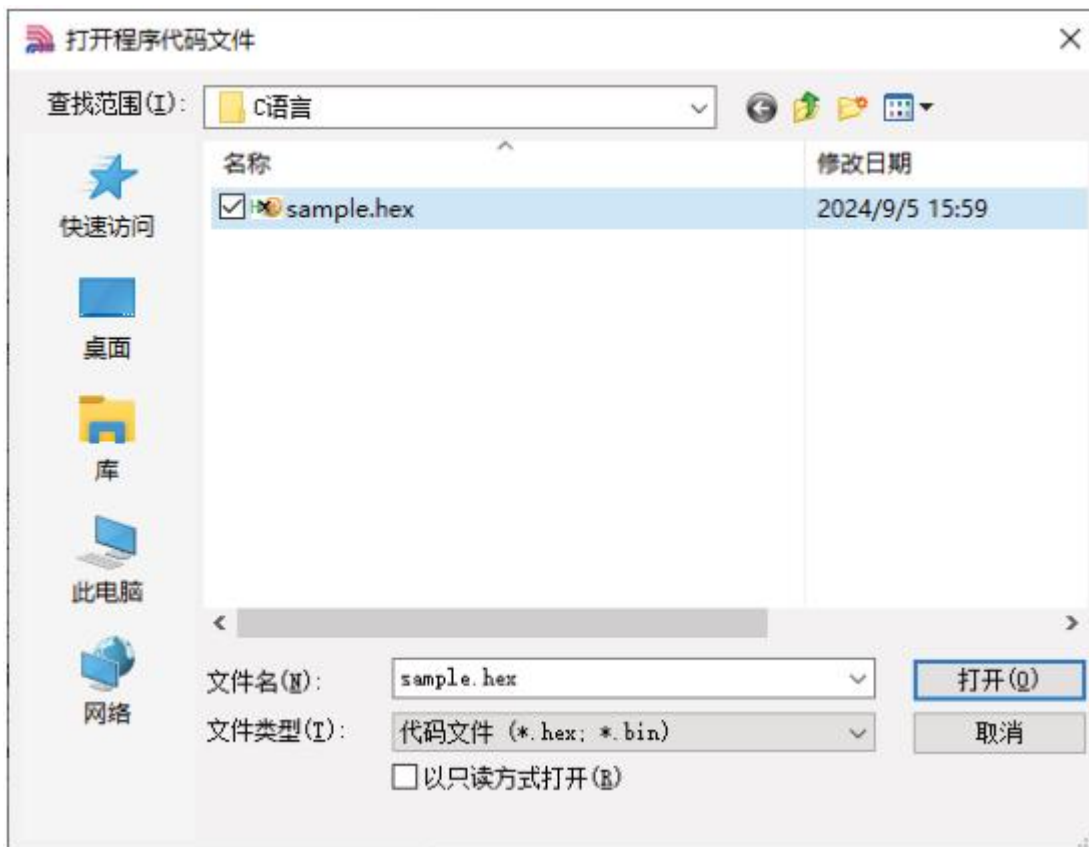
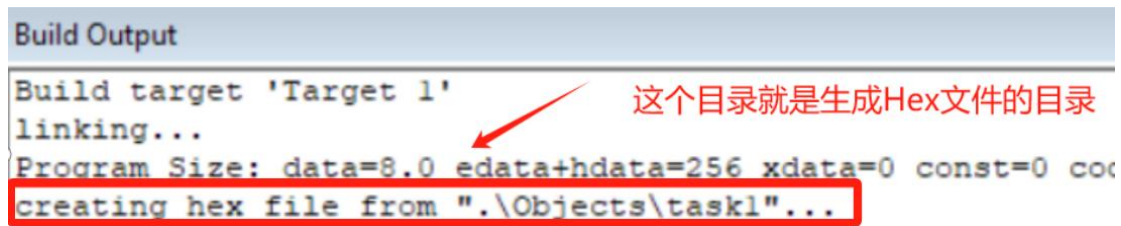


装载程序文件

然后是装载程序文件，通过点击界面上的“打开程序文件”

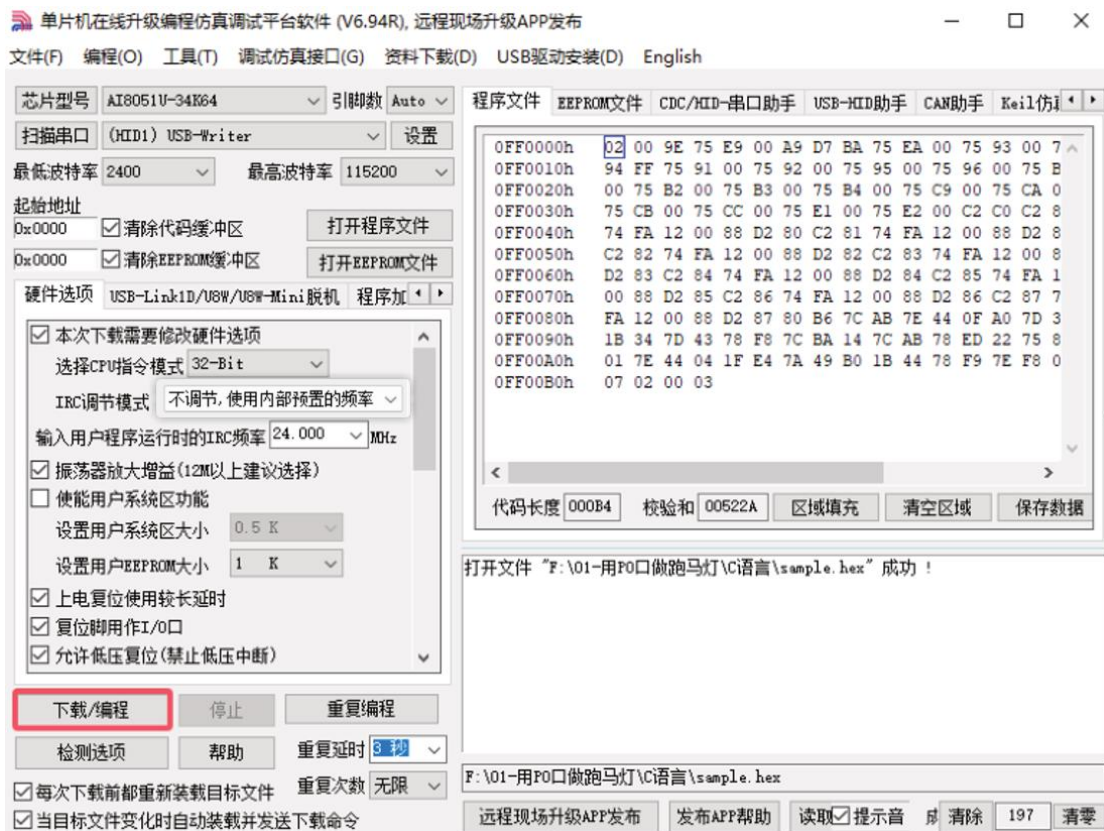


就可以选择对应的程序文件了，弹出以下对话框，选择文件编译生成的 Hex 文件，一般是在对应工程目录下的 Objects 文件夹，具体可以回顾编译时的 Output 输出信息

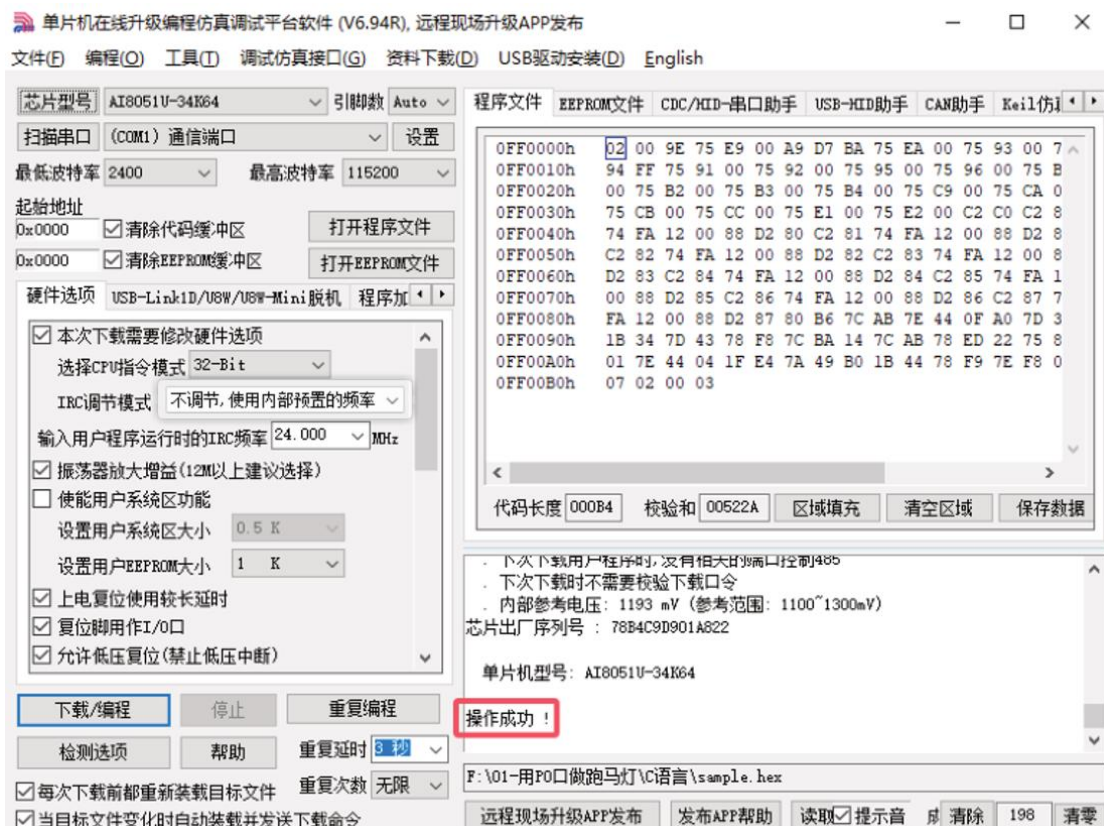


下载代码

文件正确打开后，点击界面中的“下载 / 编程”按钮开始下载代码。



下载成功，会出现如下画面：



下载后的效果

下载完成后，应该就可以看到 LED 灯被点亮了。

让 LED 闪烁起来

通过 STC-ISP 的小工具辅助生成代码

首先，延时函数通过 STC-ISP 软件进行生成：



添加头文件

需要注意的是，程序中，需要在头文件部分添加"intrins.h"

#include "intrins.h" //使用_nop_()函数所必须要包含的头文件

否则延时函数中调用的_nop_()函数没有被头文件引用过来，会导致编译器找不到这个而函数而报错。

这里的主频选择 40Mhz，因为单片机内部存在了一个 RC 时钟，所以可以在允许的范围内随意设定时钟。一般使用情况下，就不用去外挂一个晶振了。

代码部分

程序代码如下：

```
#include <AI8051U.H> //包含 AI8051U 的头文件
```

```
#include "intrins.h" //使用_nop_()函数所必须要包含的头文件
```

```
void Delay500ms(void) //@40MHz
```

```
{
    unsigned long edata i;

    _nop_();
    _nop_();
    i = 4999998UL;
    while (i) i--;
```



```

}

void main(void)
{
    EAXFR = 1; //允许访问扩展的特殊寄存器, XFR
    WTST = 0;      //设置取程序代码不等待, 程序以最快速度运行
    CKCON = 0;     //设置访问片内扩展的 xdata 部分速度为最快, 不等待
    P0M0 = 0x00; P0M1 = 0x00;
    P1M0 = 0x00; P1M1 = 0x00;
    P2M0 = 0x00; P2M1 = 0x00;
    P3M0 = 0x00; P3M1 = 0x00;
    P4M0 = 0x00; P4M1 = 0x00;
    P5M0 = 0x00; P5M1 = 0x00;
    P6M0 = 0x00; P6M1 = 0x00;
    P7M0 = 0x00; P7M1 = 0x00;

    P40 = 0; //打开 LED 部分的供电
    while(1)
    {
        //LED 循环自动闪烁部分
        P0 = ~P0; //对 P0 端口取反
        //P00 = ~P00; //对单个端口取反
        Delay500ms(); //延时 0.5s
    }
}

```

下载后的效果

重复上面的下载操作, 就可以看到 LED 灯已经闪烁起来了。

通过 USB 控制 AI8051U 进行 LED 点灯

下载 USB 库文件

下面是通过 CDC 串口方式来控制这个 LED 灯, 这里就需要添加常用头文件

```
#include "ai_usb.h" //使用 usb 所需要包含的头文件
```

然后是添加 usb_cdc 和 usb_hid 的库函数文件

这部分可以到 <https://www.stcai.com/khs>

的 USB 库文件中进行下载



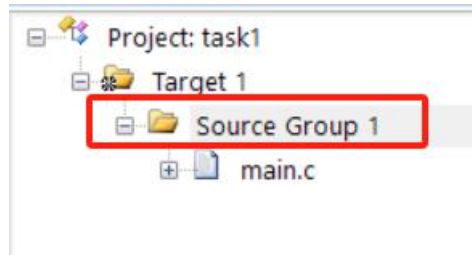
USB库文件

带硬件USB模块的芯片，USB-HID/CDC库文件以及配套的头文件，应用范例。
Ai8051U根据所选的8位/32位模式可使用相同位数的库文件，20241210版本

文件下载

下载完成后，在'库文件\查询模式 STC-HID 库文件'和'库文件\查询模式 STC-CDC 库文件'中，复制出来 stc_usb_cdc_32.LIB 和 stc_usb_hid_32.LIB 两个文件。

将这两个文件粘贴到创建工程所在的文件夹中，然后在 Keil 中，双击 Source Group



在弹出的 Add Files to Group 'Source Group 1' 界面中，选择两个 LIB 文件中的其中之一添加进入工程（不可以同时添加），添加 stc_usb_cdc_32.LIB 就是使用的 USB-CDC 功能，添加 stc_usb_hid_32.LIB 就是使用的 USB-HID 功能。

CDC 和 HID 的区别

这两个功能的主要区别就是 USB-HID 是普通 USB 键盘鼠标和 USB 鼠标的协议，兼容性更好，无需安装驱动。USB-CDC 串口则更类似传统串口的形式，使用 COM 口进行通讯，但是只有 Win10 及其以上才默认携带 USB-CDC 的驱动，较低的系统版本可能要手动安装驱动才能使用。

使用 USB-CDC 部分

这里我们先添加 USB-CDC 库文件，来进行 CDC 控制 LED 灯的实验



这样就是添加了 USB-CDC 库

代码部分

添加完成后，编写如下程序：

```
#include <AI8051U.H> //包含 AI8051U 的头文件
#include "ai_usb.h" //使用 usb 所需要包含的头文件

char *USER_DEVICEDESC = NULL;
char *USER_PRODUCTDESC = NULL;
char *USER_STCISPCMD = "@STCISP#";

void main(void)
{
    EAXFR = 1; //允许访问扩展的特殊寄存器，XFR
    WTST = 0; //设置取程序代码不等待，程序以最快速度运行
    CKCON = 0; //设置访问片内扩展的 xdata 部分速度为最快，不等待
    P0M0 = 0x00; P0M1 = 0x00;
    P1M0 = 0x00; P1M1 = 0x00;
    P2M0 = 0x00; P2M1 = 0x00;
    P3M0 = 0x00; P3M1 = 0x00;
    P4M0 = 0x00; P4M1 = 0x00;
    P5M0 = 0x00; P5M1 = 0x00;
    P6M0 = 0x00; P6M1 = 0x00;
    P7M0 = 0x00; P7M1 = 0x00;

    usb_init(); //USB 接口配置
    EA = 1; //打开总中断
    while (DeviceState != DEVSTATE_CONFIGURED); //等待 USB 完成配置
    P40 = 0; //打开 LED 部分的供电
    while(1)
    {
        //使用 USB 串口控制来手动控制 LED 部分
        if(bUsbOutReady)
        {
            P0 ^= UsbOutBuffer[0];
            LED40_SetPort(0,P0);
            usb_OUT_done();
        }
    }
}
```

```

}
}

```

这其中，

```

char *USER_DEVICEDESC = NULL;
char *USER_PRODUCTDESC = NULL;
char *USER_STCISPCMD = "@STCISP#";

```

是使用 USB 必须声明的部分

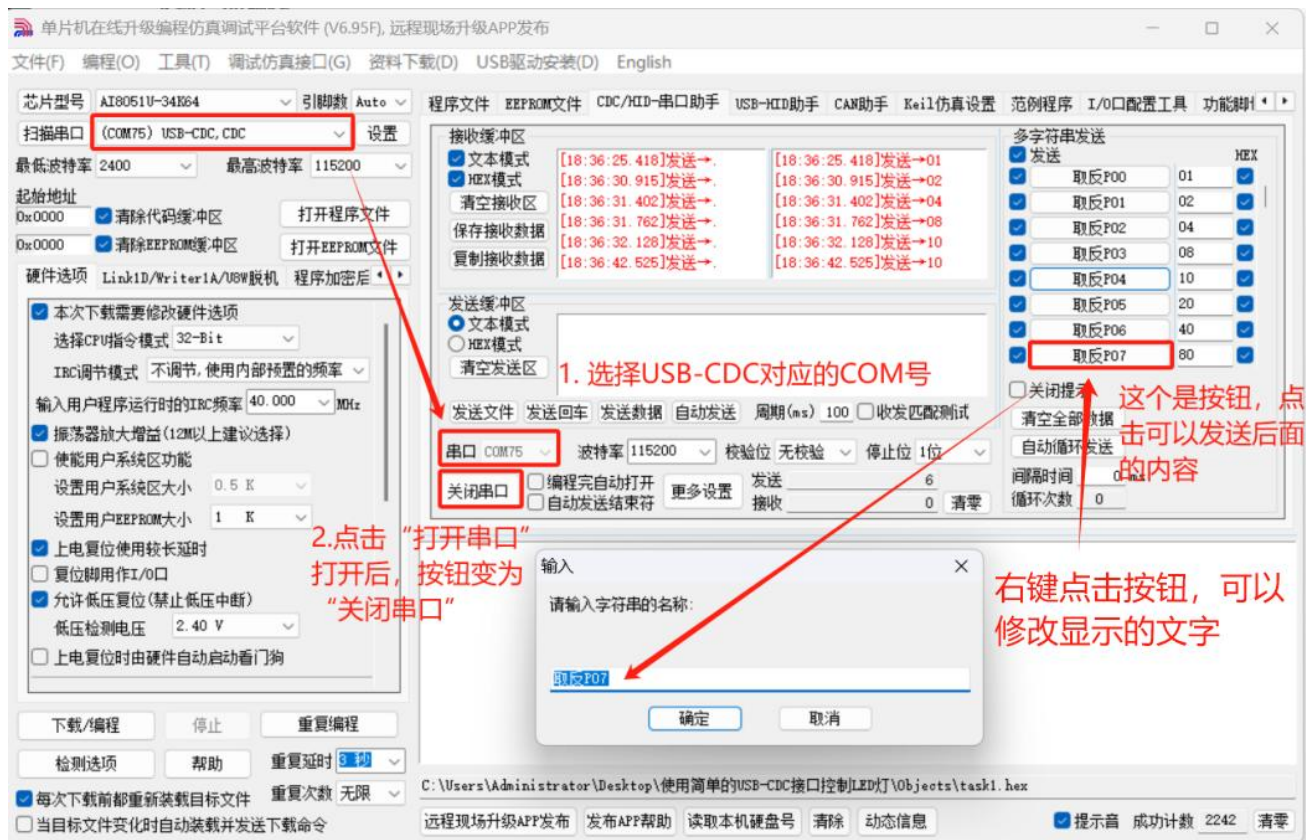
bUsbOutReady 标志位是用于标明 USB 主机发送完成。

$P0 \wedge = \text{UsbOutBuffer}[0]$; 这一句的意思是从发送的数据中，取出第一位来与 P0 口数据进行异或处理，即对应位会进行取反。例如发送 0xff，则所有的位都会进行取反，发送 0x01，则只有 P00（LED0）会被取反。

下载后的操作

下载步骤同上，下载后，打开 STC-ISP，应该可以看到串口部分已经出现了一个 USB-CDC 串口，如果电脑系统并没有产生 CDC 端口，可能是因为系统版本较低（低于 Windows10 1903 版本），可以跳转到后面的 [HID 部分实验控制](#) 实现进行

看到 USB-CDC 端口后，点击 CDC/HID 串口助手选项卡，然后按照下图中步骤操作。



此时就可以看到串口扫描部分出现了 USB-CDC 端口 COM75（因不同电脑，这个 COM 号有所不同）

此时通过串口助手打开对应的 CDC 串口，在右侧的多字符串发送部分对对应的指令进行命名，按照如图所示格式输入后，按下按键即可完成对应 LED 的亮/灭的取反，十分直观。

如何设置多字符串发送？

当然，第一次打开串口助手时，右侧的多字符串发送可能没有这么宽，并且也没有带有我的这些字，这些是需要自己手动输入一下的，具体步骤如下：



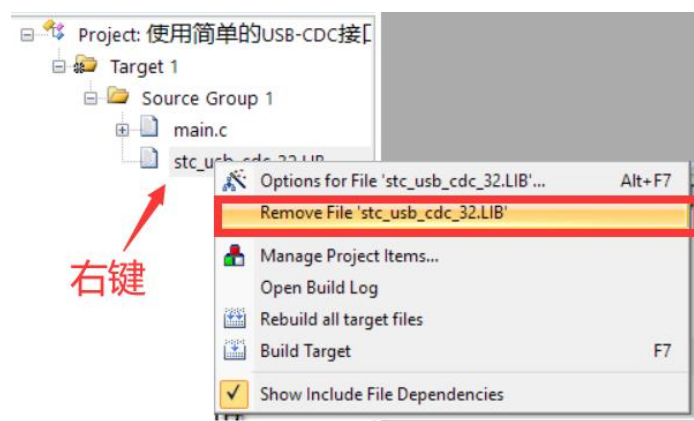
首先是调整边框，通过调整整个界面的大小和内部边框的大小，即可实现改变这部分控件的大小。这里面所有淡蓝色的条都是可以拖动调整的，可以根据自己的喜好来调整。



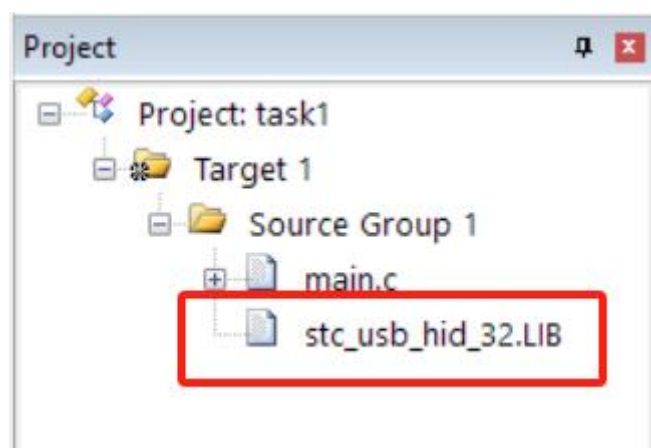
这个按钮是可以通过鼠标右键进行名称修改的，右侧的 01、02 这些则是按下按钮时对应要发送的值，对应按照截图内的内容填好，（记得右侧的 HEX 选项框要勾选）即可实现点按时改变实验箱上对应 LED 的亮/灭状态了。

使用 HID 操作方式

如果使用 USB-HID 方式，则需要将库文件切换到 stc_usb_hid_32.LIB



删除之前的 LIB 库文件，只需要在 LIB 文件上右键，然后选择 Remove File 选项，即可删除。
添加方式同上，代码部分并不需要更改



添加好后如图所示

在下载完成后，通过 STC-ISP 打开 USB-HID 助手的中断输入输出来进行数据交换。
设备名字叫 STC HID Demo，同时也可以使用右侧的多字符串发送进行控制 LED 灯。



这里同样按照图中步骤顺序依次点击后，修改多字符串发送的内容。
修改好后，点击对应的按钮，即可看到实验箱上对应 LED 的亮灭被切换了。

实验箱原理图

下载链接

可以通过这个链接进行下载：

<https://www.stcaimcu.com/data/download/Schematic/%E5%AE%9E%9AA%8C%E7%AE%B1AI8051U-SCH.pdf>

