

通过 USB-CDC 方式控制 Ai8051U 进行 LED 点灯

通过 USB-CDC 串口方式来控制 Ai8051U 对 8 个 LED 点灯

准备工作：

1、添加常用头文件 ai_usb.h

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

这个头文件在使用 AIapp-ISP 软件添加型号和头文件到 Keil 中时就已经一并被添加到 Keil 中了。

2、添加 usb_cdc 库函数文件

这部分可以到 STCAI 的官网：<https://www.stcai.com/khs> 的 USB 库文件中进行下载

软件工具

工具软件

库函数

实验箱

核心功能实验板

嵌入式系统软件

做自己的升级软件

其他

库函数

Ai8051U创新风格库函数
Ai8051U创新风格的库函数例程包，持续更新，20250219版本
[代码下载] [使用说明]

Ai8051U传统风格库函数
Ai8051U传统风格的库函数例程包，持续更新，20250214版本
[代码下载]

Ai8051U系列TFPU库文件
Ai8051U系列硬件三角函数与浮点运算单元 (TFPU) 的lib库文件下载，可以使用PLL高速时钟作为时钟源，20250115版本，更新atan函数
[8位模式] [32位模式]

32G库函数
32G12K128为主控芯片的库函数例程包，20250205版本
[代码下载] [使用说明]

8G-8H库函数
8G系列、8H系列单片机通用库函数例程包，20250205版本
[代码下载] [使用说明]

Ai8051U系列MDU库文件
Ai8051U系列硬件32位乘除单元 (MDU32) 的lib库文件下载
[8位模式] [32位模式]

32系列TFPU库文件
32系列硬件三角函数与浮点运算单元 (TFPU) 的Large模式与Huge模式lib库文件下载，20241223版本
[Large模式] [Huge模式]

32系列MDU库文件
32系列硬件32位乘除单元 (MDU32) 的Large模式与Huge模式lib库文件下载
[Large模式] [Huge模式]

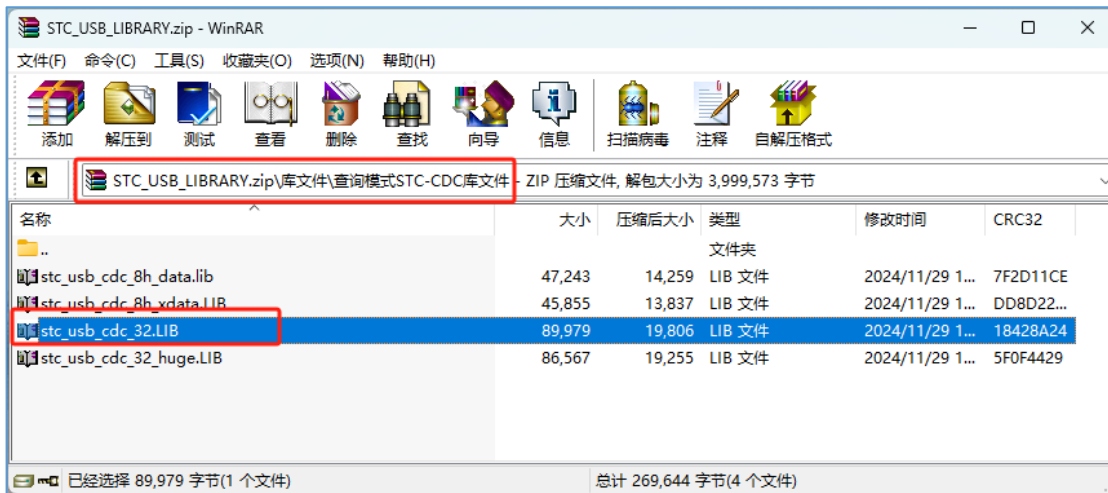
USB库文件
带硬件USB模块的芯片，USB-HID/CDC库文件以及配套的头文件，应用范例。Ai8051U根据所选的8位/32位模式可使用相同位数的库文件。20241210版本
[文件下载]

8A8K64D4库函数
8A8K64D4系列芯片通用库函数例程包
[代码下载]

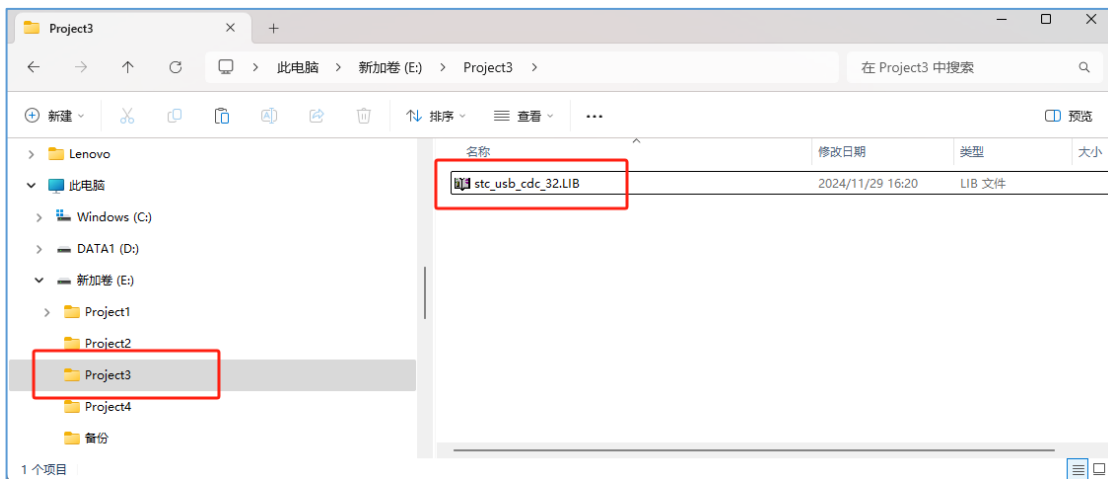
8系列MDU库文件
8系列硬件16位乘除单元 (MDU16) 和浮点运算单元 (FPU) 的lib库文件下载
[代码下载]

下载完成后，打开压缩包“STC_USB_LIBRARY.zip”，

在‘库文件\查询模式 STC-CDC 库文件’中，复制出来 stc_usb_cdc_32.LIB

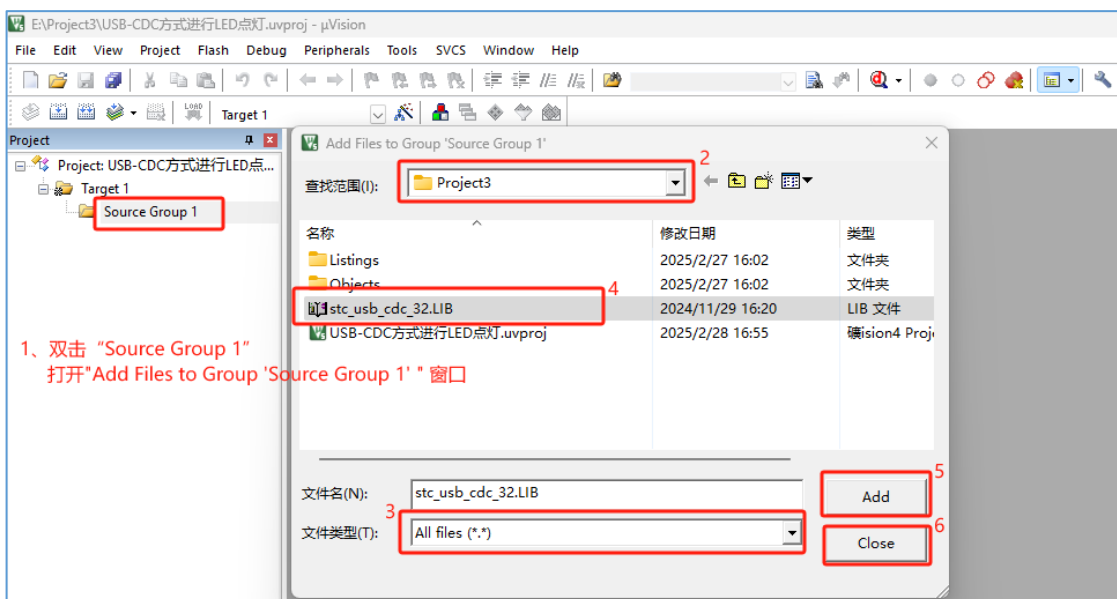


3、将库函数“stc_usb_cdc_32.LIB”复制到创建工程所在的文件夹 project3 中

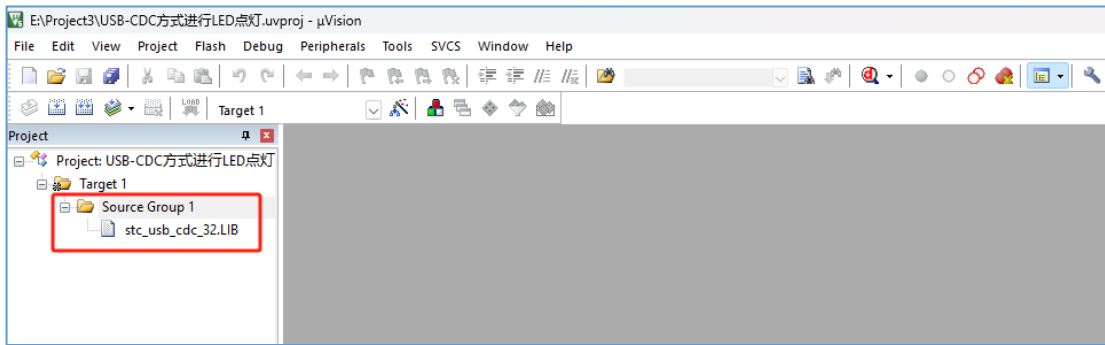


4、将库函数 stc_usb_cdc_32.LIB 添加进入此工程项目中

在 Keil 中，双击 Source Group，在弹出的 Add Files to Group 'Source Group 1' 窗口中，将库函数 stc_usb_cdc_32.LIB 添加进入此工程，就可以使用 USB-CDC 功能点亮 LED 灯。



这样，Keil 界面变化如下图：



注意：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。

代码部分

添加完成后，编写如下程序：

//通过 USB-CDC 方式控制 AI8051U 进行 LED 点灯

#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，这个头文件在使用 AIapp-ISP 软件
//添加型号和头文件到 Keil 中时就已经一并被添加到 Keil 中了。

char *USER_DEVICEDESC = NULL; //USB 库里预留的设备描述符，必须定义，但不要使用

char *USER_PRODUCTDESC = NULL; //USB 库里预留的产品描述符，必须定义，但不要使用

char *USER_STCISPCMD = "@STCISP#"; //这里使用"@STCISP#"这个字符串当作不停电自动

//下载命令，用户只需要在这里定义好这个字符串，

//后面添加的库文件的 USB 中断程序会自动判断从电脑端

//下传的数据是否和这个字符串一样，如果一样，则会自动

//软复位到系统区等待 USB 下载，

//用户不需要进行额外的处理。

void main(void)

{

BYTE dat; //定义变量保存接收的数据

EAXFR = 1; //允许访问扩展的特殊寄存器，XFR

//(32 位模式请使用这句，注释下一句)

// P_SW2 |= 0x80; //允许访问扩展的特殊寄存器，XFR

//(8 位模式请使用这句，注释上一句)。

WTST = 0; //设置取程序代码等待时间，

//赋值为 0 表示不等待，程序以最快速度运行

CKCON = 0; //设置访问片内的 xdata 速度，

//赋值为 0 表示用最快速度访问，不增加额外的等待时间

P0M0 = 0x00; P0M1 = 0x00; //设置 P0 口为准双向口模式

P1M0 = 0x00; P1M1 = 0x00; //设置 P1 口为准双向口模式

P2M0 = 0x00; P2M1 = 0x00; //设置 P2 口为准双向口模式

```

P3M0 = 0x00; P3M1 = 0x00;          //设置 P3 口为准双向口模式
P4M0 = 0x00; P4M1 = 0x00;          //设置 P4 口为准双向口模式
P5M0 = 0x00; P5M1 = 0x00;          //设置 P5 口为准双向口模式
P6M0 = 0x00; P6M1 = 0x00;          //设置 P6 口为准双向口模式
P7M0 = 0x00; P7M1 = 0x00;          //设置 P7 口为准双向口模式

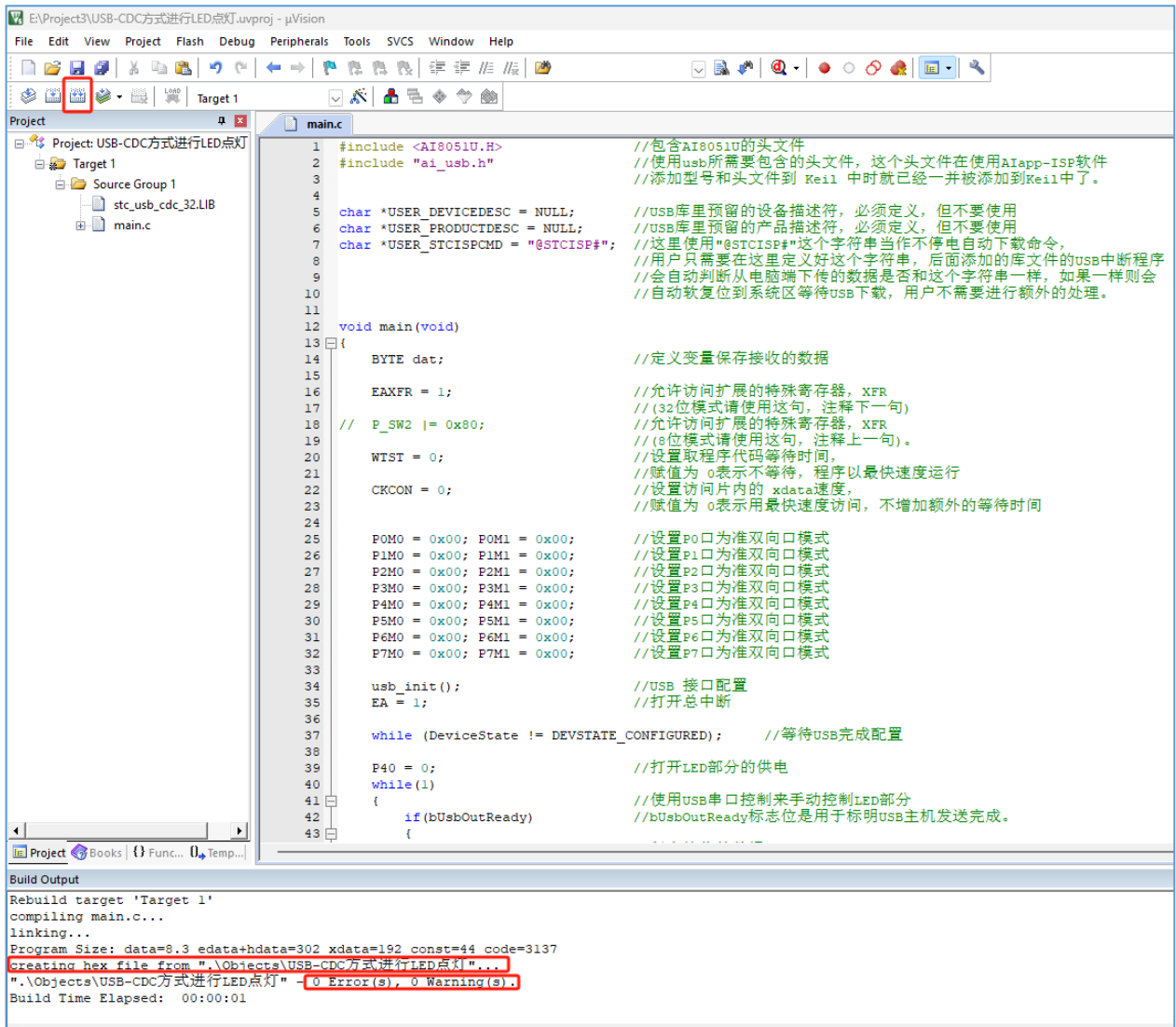
usb_init();                          //USB 接口配置
EA = 1;                             //打开总中断

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)                 //bUsbOutReady 标志位是用于标明 USB 主机发送完成。
    {
        dat = UsbOutBuffer[0];       //保存接收的数据
        P0 = dat;                    //将接收的数据送到 P0 口显示
        LED40_SetPort(0,dat);         //使用下载软件中的“调试仿真接口”中
        //“擎天柱-LED-DIP40”进行模拟显示 P0 口
        usb_OUT_done();               //本包接收的数据处理完毕，准备接收下一包数据
    }
}
}

```

代码编写完后，点击工具栏上的 Rebuild 按钮进行编译

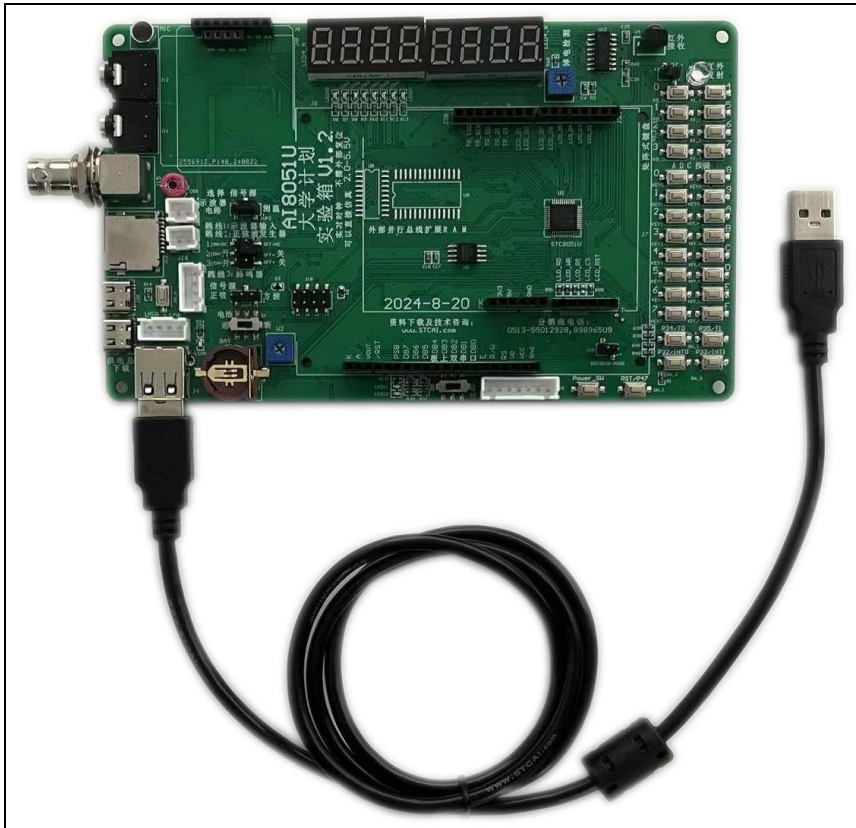


出现 creating hex file 和 0 Error(s) 0 Warning (s) 即为编译成功，成功创建了 hex 文件且无报错。此 hex 文件，一般是在对应工程目录下的 Objects 文件夹里。

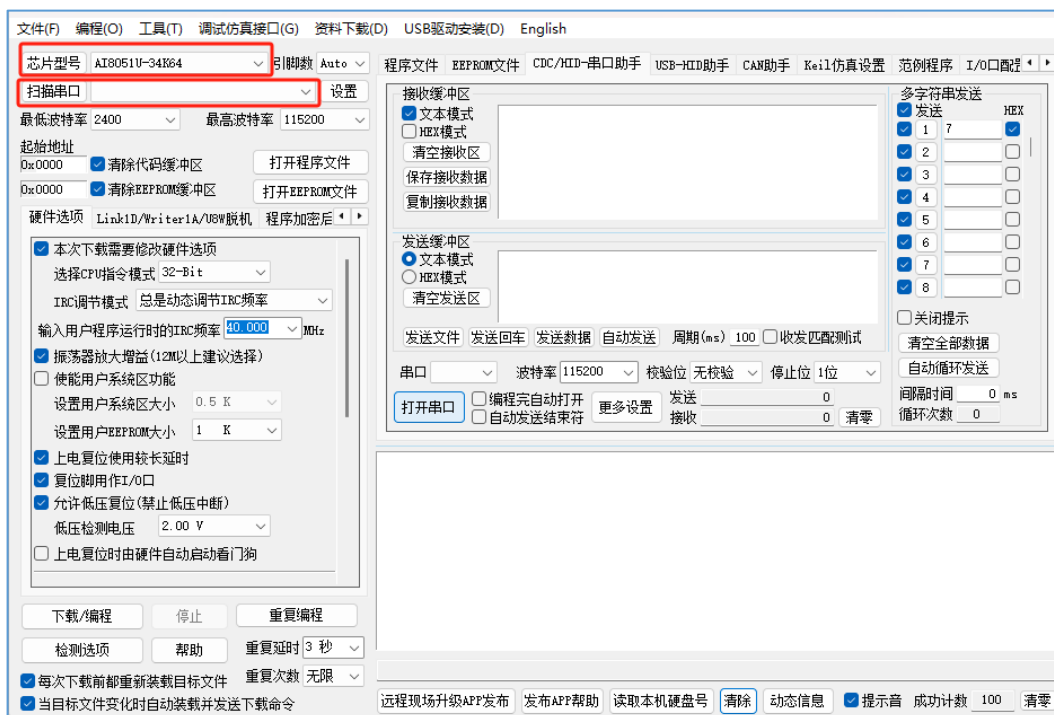
下载用户程序到实验箱

连接到实验箱

首先使用实验箱中配置的 USB-TypeA 到 USB-TypeA 线将实验箱与电脑正确连接。

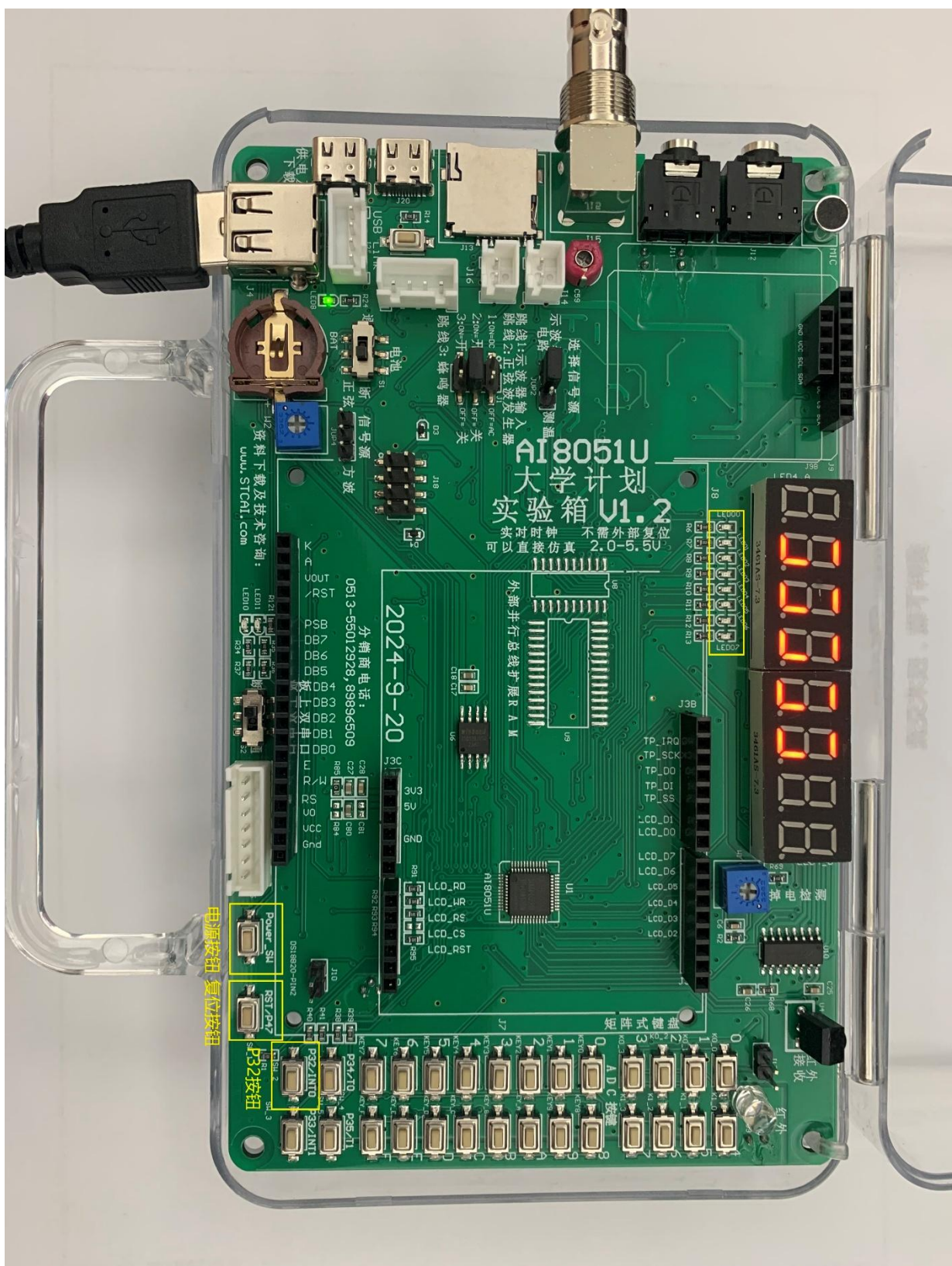


然后打开 AIapp-ISP-v6.95F 或以上版本。选择单片机型号：Ai8051U-34K64，如下图，我们看到“扫描串口”后面的端口列表框不是“（HID1）USB-Writer”，说明还不能下载程序到实验箱。

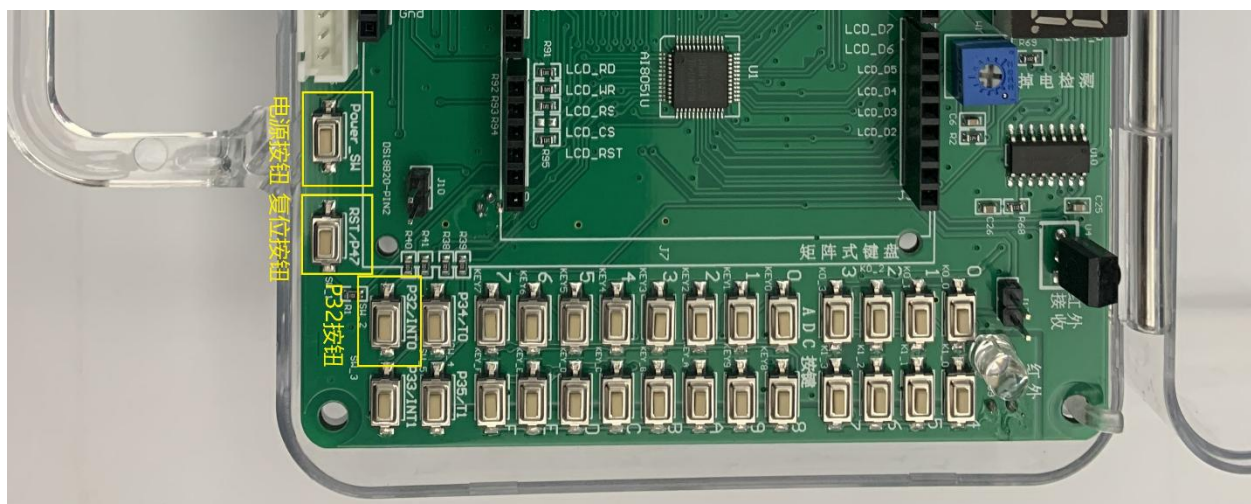


硬件 USB 下载用户程序步骤

下图是实验箱已经和电脑相连的正面图片，数码管是随机显示的，数码管下面的 8 个 LED 灯都是暗淡无光的，此实验的目的是通过 USB-CDC 方式控制 AI8051U 进行 LED 点灯。



实验箱正下方有两排按键，找到 P3.2 按键和 Power_SW 按键



【方法一】按下 P3.2 按键保持不动，再按一下 Power_SW 按键，你会发现 Alapp-ISP 软件界面的“扫描串口”后面的端口列表框有了内容：(HID1) USB-Writer，说明这时可以下载程序到实验箱。如下图所示。

【原理】：按下 P3.2 按键，就是将 P3.2 接地，再按下 Power_SW 按键停电/再松开上电，当 Alapp-ISP 软件界面的“扫描串口”后面的端口列表框内显示：(HID1)USB-Writer，说明 USB 连接上了，AI8051U 单片机在等待电脑送目标文件下载到单片机内部，这时可以松开 P3.2 按键了。

【方法二：复位管脚低电平复位下载】

USB 连接好并已上电的情况下，外部按键复位也可进入 USB 下载模式，注意：

P4.7-nRST 出厂时默认是 P4.7-I/O 功能，要改为复位功能，需 ISP 烧录时取消设置复位脚用作 I/O 口，停电一次再上电才生效，程序区中用户程序也可改为复位脚或 I/O，这个立即生效。

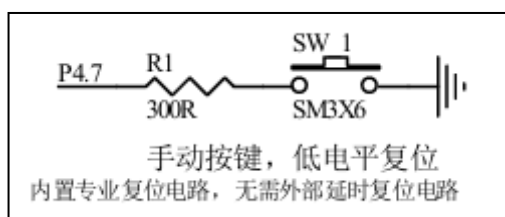
1、按下 P4.7-nRST 外接的低电平复位按键复位 MCU，

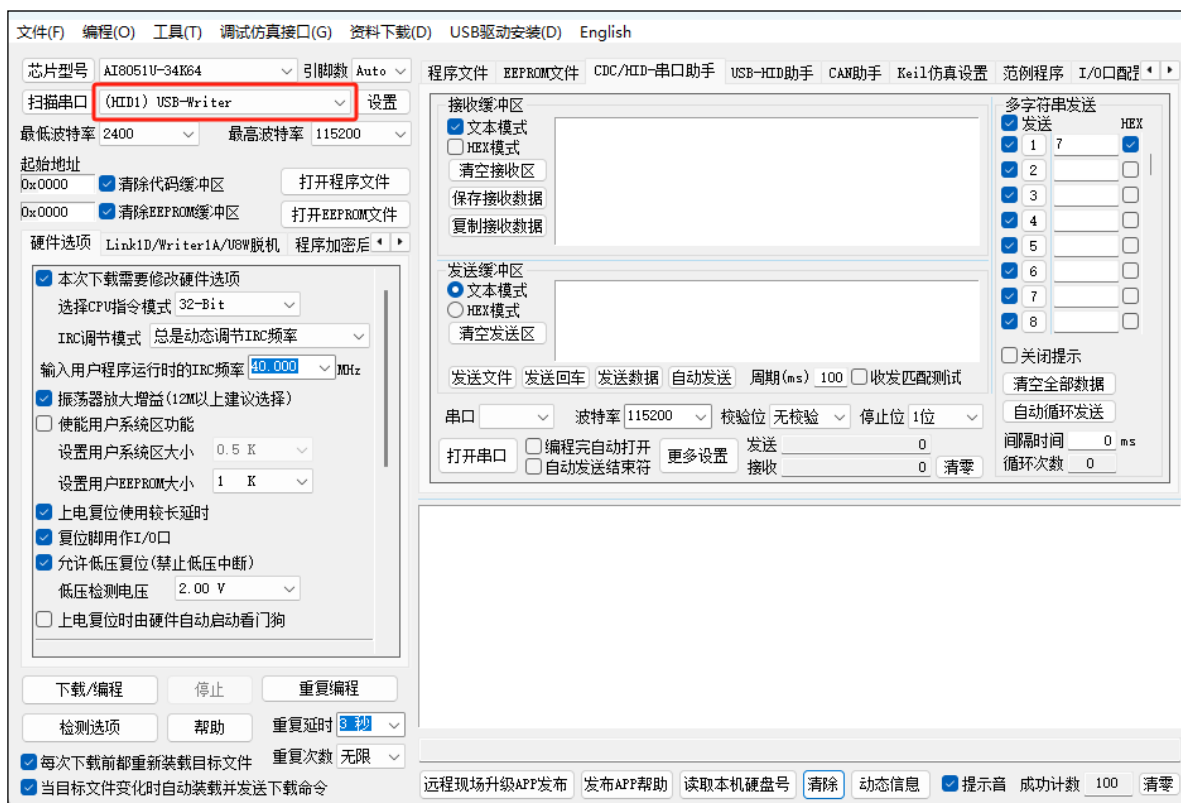
松开复位键，MCU 从系统程序区启动，判断是否要下载用户程序，

等待电脑端 ISP 下载软件中自动识别出“(HID1) USB Writer”后

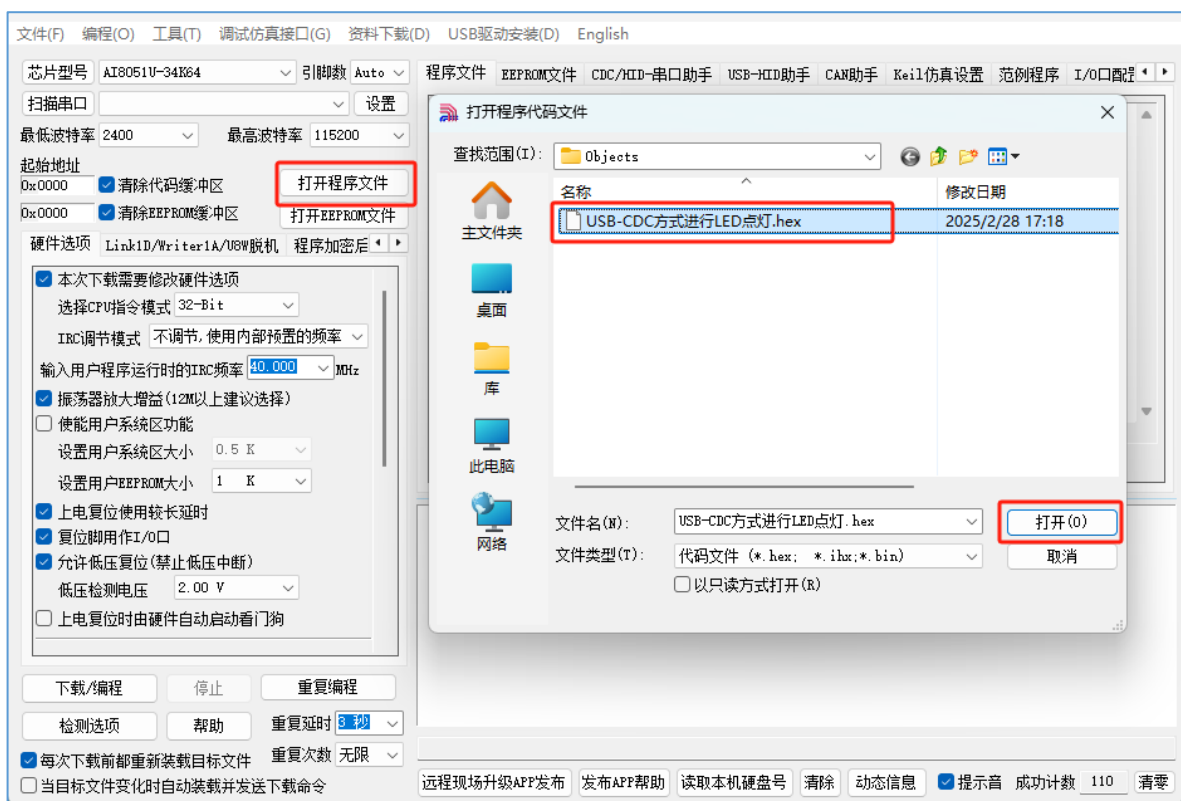
2、点击电脑端下载软件中的【下载/编程】按钮

下载进行中，几秒钟后，提示下载成功！

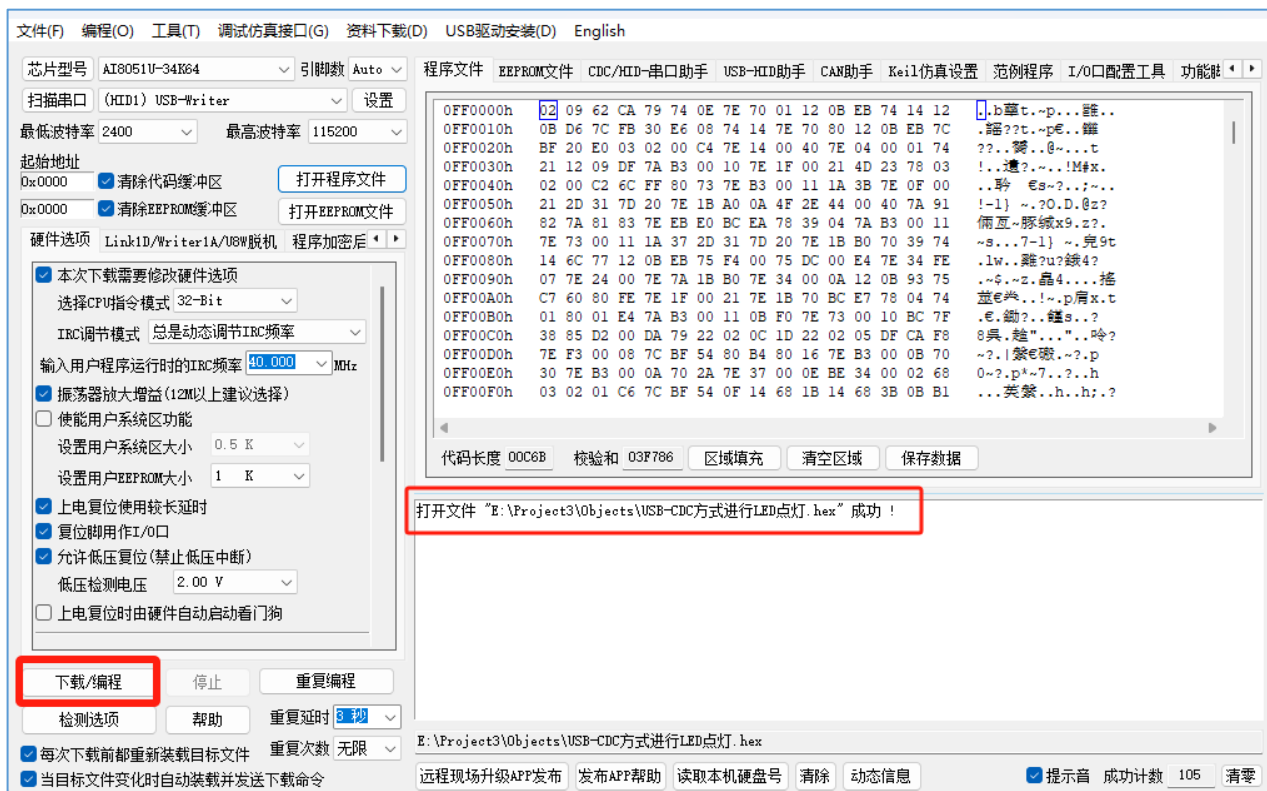




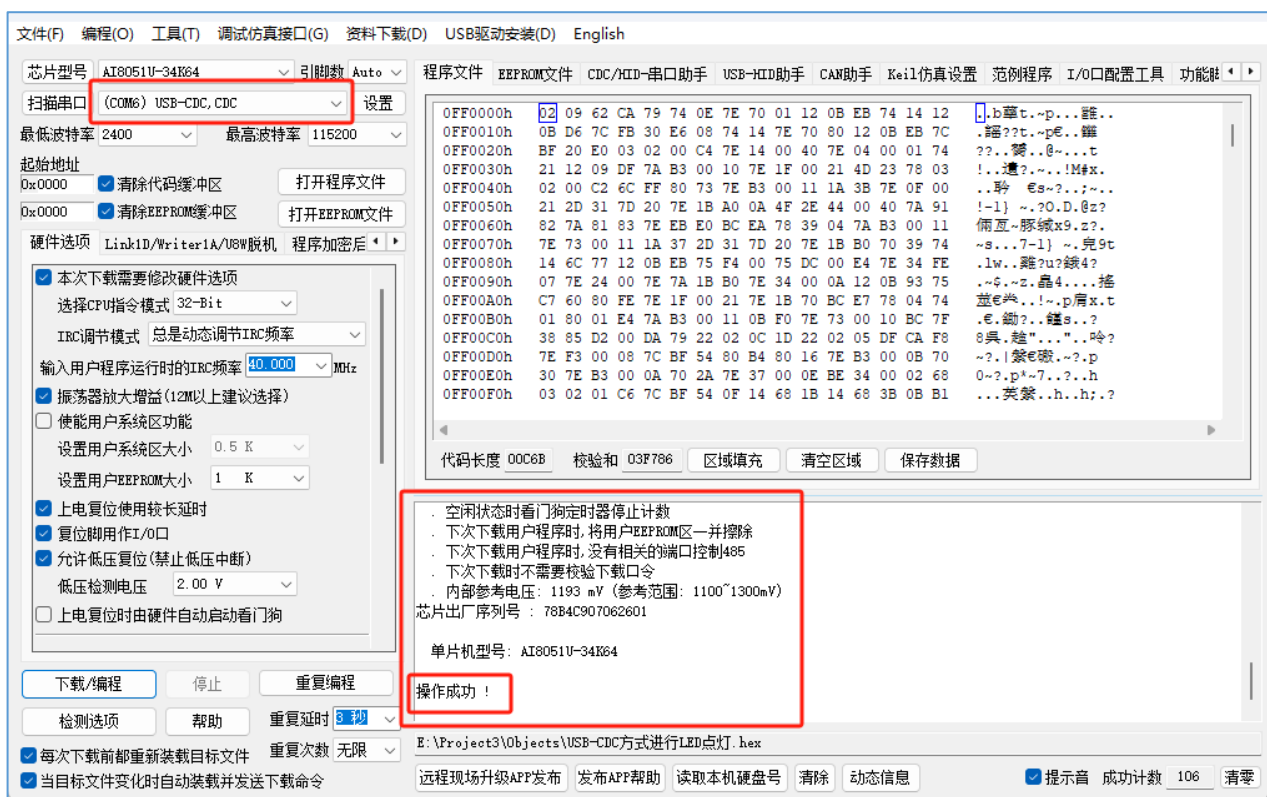
现在装载程序文件，通过点击界面中的“打开程序文件”，加载之前用 Keil 中编译好的 HEX 文件：E:\Project3\Objects\USB-CDC 方式进行 LED 点灯.hex



然后，Alapp-ISP 软件界面如下图所示，出现“‘打开文件***.hex’成功！”



点击“下载/编程”，把 task1.hex 烧录到实验箱，AIapp-ISP 软件界面出现：操作成功！
恭喜你！



这时在 AIapp-ISP 软件界面，我们看到串口选项框中已经出现了一个(COM6)USB-CDC,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 版本）。

在 AIapp-ISP 软件界面，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

The screenshot shows the Keil IDE interface with the following components and annotations:

- Top Menu Bar:** 文件(F), 编辑(O), 工具(T), 调试仿真接口(G), 资料下载(D), USB驱动安装(D), English.
- Project Explorer (Left):**
 - 芯片型号: AI8051U-34K64
 - 引脚数: Auto
 - 扫描串口: (COM6) USB-CDC, CDC
 - 最低波特率: 2400, 最高波特率: 115200
 - 起始地址: 0x0000, 清除代码缓冲区, 打开程序文件
 - 0x0000, 清除EEPROM缓冲区, 打开EEPROM文件
 - 硬件选项: LinkID/Writer1A/USW脱机, 程序加密后
 - 本次下载需要修改硬件选项: 选择CPU指令模式: 32-Bit, IRC调节模式: 总是动态调节IRC频率, 输入用户程序运行时的IRC频率: 40.000 MHz, 振荡器放大增益: 12M (以上建议选择), 使能用户系统区功能, 设置用户系统区大小: 0.5 K, 设置用户EEPROM大小: 1 K, 上电复位使用较长延时, 复位脚用作I/O口, 允许低压复位 (禁止低压中断), 低压检测电压: 2.00 V, 上电复位时由硬件自动启动看门狗
 - 按钮: 下载/编程, 停止, 重复编程, 检测选项, 帮助, 重复延时: 5秒, 每次下载前都重新装载目标文件, 重复次数: 无限, 当目标文件变化时自动装载并发送下载命令
- Source Explorer (Left):**
 - 程序文件: CDC/HID-串口助手
 - EEPROM文件: CDC/HID-串口助手
 - USB-HID助手, CAN助手, Keil仿真设置
 - 范例程序, I/O口配置工具, 功能键切换
- Receive Buffer (Center):**
 - 接收缓冲区: 文本模式, HEX模式 (checked), 清空接收区, 保存接收数据, 复制接收数据
 - 发送缓冲区: 文本模式 (selected), HEX模式, 清空发送区
 - 输入: 请输入字符串的名称: 点亮P00-P03灯
 - 发送文件: 发送, 串口: COM6, 关闭串口
- Multi-String Send (Right):**
 - 多字符串发送: 发送, 点灭P00-P03灯 (f0), 点亮P04-P07灯 (0f), 点亮P00-P07灯 (00), 熄灭P00-P07灯 (ff), 5, 6, 7, 8, 关闭提示, 清空全部数据, 自动循环发送, 间隔时间: 0 ms, 循环次数: 0
- Console (Bottom):**
 - 空闲状态时看门狗定时器停止计数
 - 下次下载用户程序时, 将用户EEPROM区一并擦除
 - 下次下载用户程序时, 没有相关的端口控制485
 - 下次下载时不需要校验下载命令
 - 内部参考电压: 1193 mV (参考范围: 1100~1300mV)
 - 芯片出厂序列号: 78B4C907062601
 - 单片机型号: AI8051U-34K64
 - 操作成功!
 - E:\Project3\Objects\USB-CDC方式进行LED点灯.hex
 - 远程现场升级APP发布, 发布APP帮助, 读取本机硬盘号, 清除, 动态信息
 - 提示音, 成功计数: 110, 清屏

Annotations:

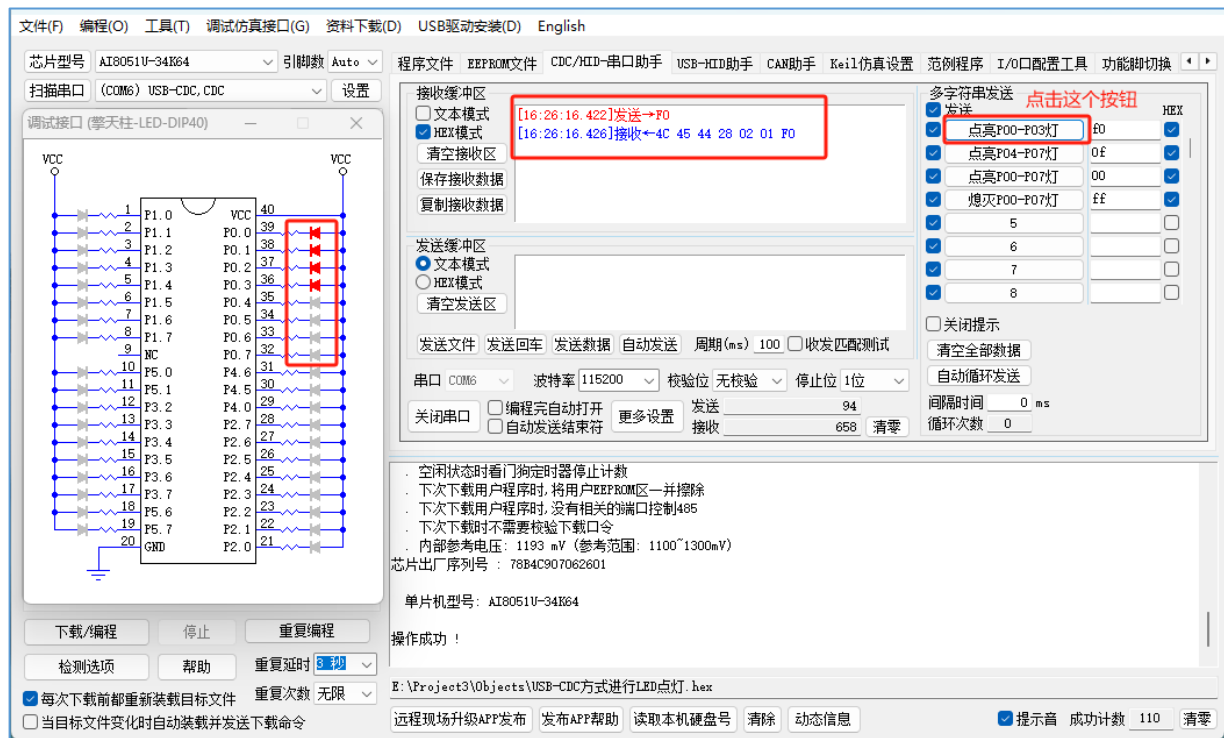
1. 指向 "CDC/HID-串口助手" 在 Source Explorer 中的位置。
2. 指向 "HEX模式" 在 Receive Buffer 中的复选框。
3. 指向 "点亮P00-P03灯" 在 Input field 中的文本。
4. 指向 "输入16进制数, 控制LED灯的亮 (0) 和灭 (1)" 在 Multi-String Send 中的 "f0" 值。
5. 指向 "勾选HEX模式" 在 Multi-String Send 中的 "HEX" 复选框。

The screenshot displays the Keil IDE's 'Options for Target' dialog box, specifically the 'Debug' tab. The 'Serial Port' is set to 'COM6'. The 'Hardware' section shows 'Link1D/Writer1' selected. The 'Simulation' section shows 'Keil' selected. The 'Debug' section shows 'Keil' selected. The 'Options for Target' dialog box is open, and the 'Debug' tab is selected. The 'Serial Port' is configured to 'COM6'. The 'Hardware' section shows 'Link1D/Writer1' selected. The 'Simulation' section shows 'Keil' selected. The 'Debug' section shows 'Keil' selected.

点击 CDC/HID-串口助手选项卡中的“点亮 P00-P03 灯”按钮：

1、观察 AIapp-ISP 软件的 CDC/HID-串口助手选项卡界面，如下图

- 在接收区会出现发送、接收数据；
- 擎天柱-LED-DIP40 的 P0.0-P0.3 这 4 个灯被点亮



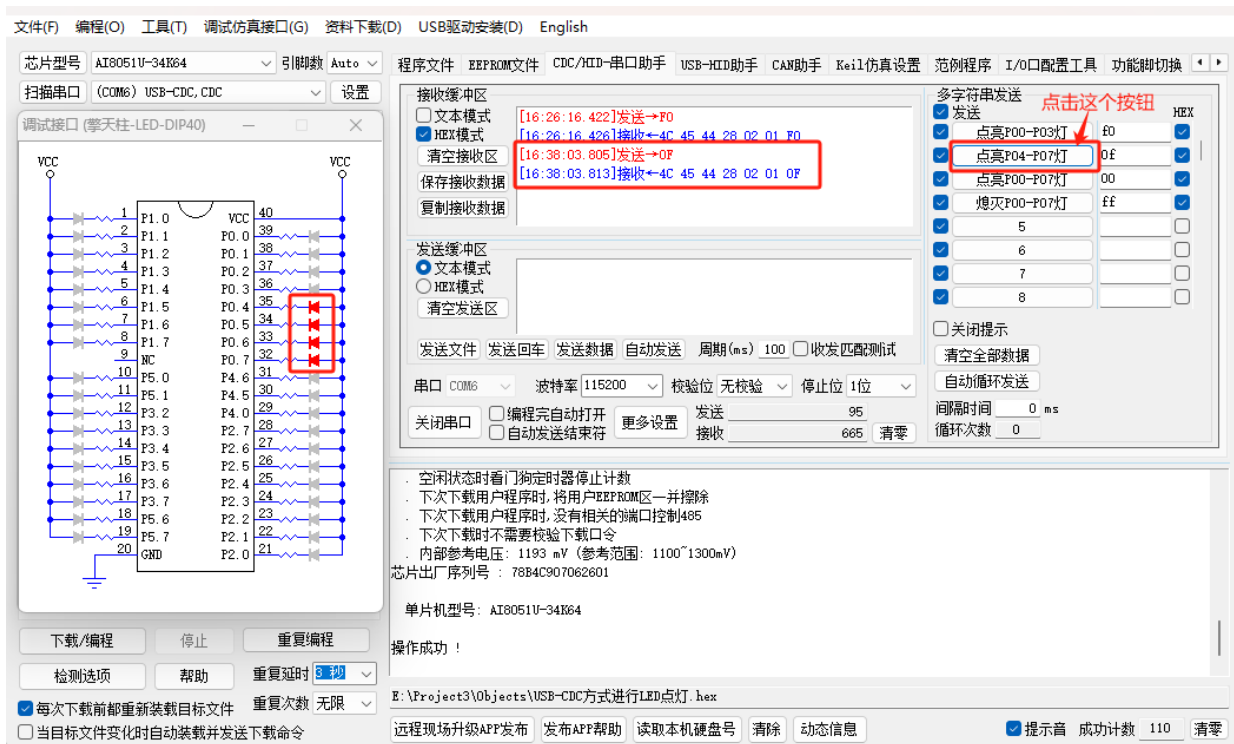
2、观察实验箱的 8 个 LED 灯，P0.0-P0.3 这 4 个 LED 灯被点亮，如下图



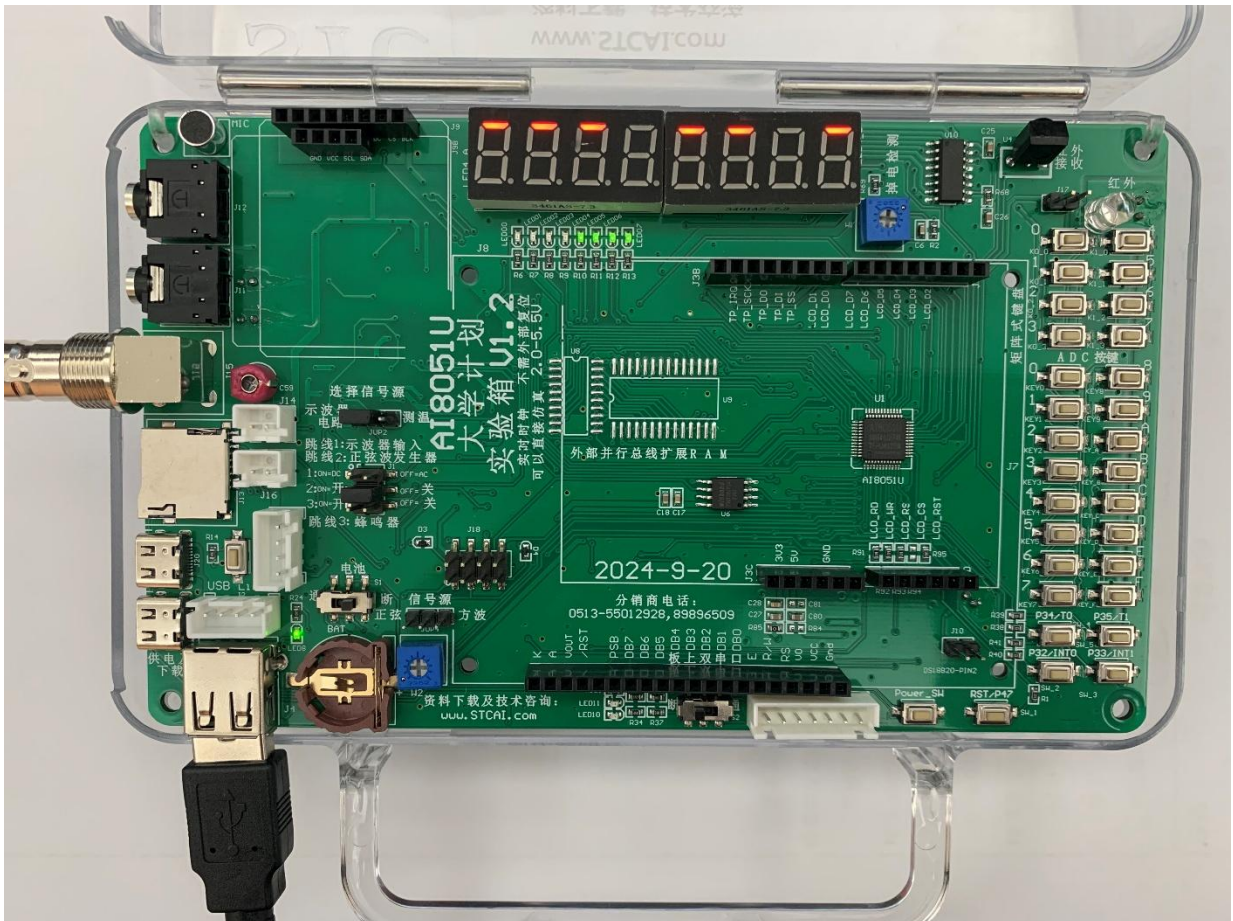
点击 CDC/HID-串口助手选项卡中的“点亮 P04-P07 灯”按钮：

1、观察 AIapp-ISP 软件的 CDC/HID-串口助手选项卡界面，如下图

- 在接收区会出现发送、接收数据；
- 擎天柱-LED-DIP40 的 P0.4-P0.7 这 4 个灯被点亮



2、观察实验箱的 8 个 LED 灯，P0.4-P0.7 这 4 个 LED 灯被点亮，如下图



至此，实验成功！