

在编写较大的程序时, 往往会遇到代码编译没问题, 但运行异常的情况, 或者要观察项目运行状况, 这时就可以借助仿真来查找问题, 进行程序调试, 或者对项目实时监控, 进行逻辑验证。

Ai8051U 可支持仿真功能。Ai8051U 芯片出厂时默认仿真功能是关闭的, 需要使用 AiCube-ISP 软件将目标单片机设置为仿真芯片, 才能开启芯片的仿真功能。

32 位 8051 芯片可支持 SWD 硬件仿真、USB 直接通信仿真、串口通信仿真。SWD 硬件仿真, 需要使用官方的 LinkID 工具作为仿真器才能进行仿真。**Ai8051U 的 A 版芯片的 SWD 有小瑕疵, 暂不支持 SWD 仿真。**

串口直接通信仿真, 不需要任何仿真器, 单片机本身就是仿真器, 直接与电脑的串口相连接就可进行仿真。**注意: 由于目前的电脑均无硬件串口, 所以需要使用 USB 转串口工具将目标芯片和电脑相连接。本文档使用的 STCAI 公司提供的“USB 转双串口工具@Ai8H2K12U”。**

下面我们就以“**深大--中断实验 1(外部中断 INT0/P3.2, INT1/P3.3 控制点灯并 USB 仿真)**”实验为例, 使用 **Ai8051U 实验箱仿真 -- 串口通信仿真**, 并观察实验现象。

1.1 安装 AiCube-ISP 软件, 制作仿真芯片

1.1.1 安装 AiCube-ISP 下载/编程/烧录 工具软件

安装 STCAI 公司研发的 AiCube-ISP 下载 / 编程 / 烧录 / 软件, 含强大的辅助开发工具包。登录 STCAI 官网, 下载最新版的 AiCube-ISP 安装包, 下载链接: <https://www.stcai.com/gjri>



下载下来的压缩包文件图标:

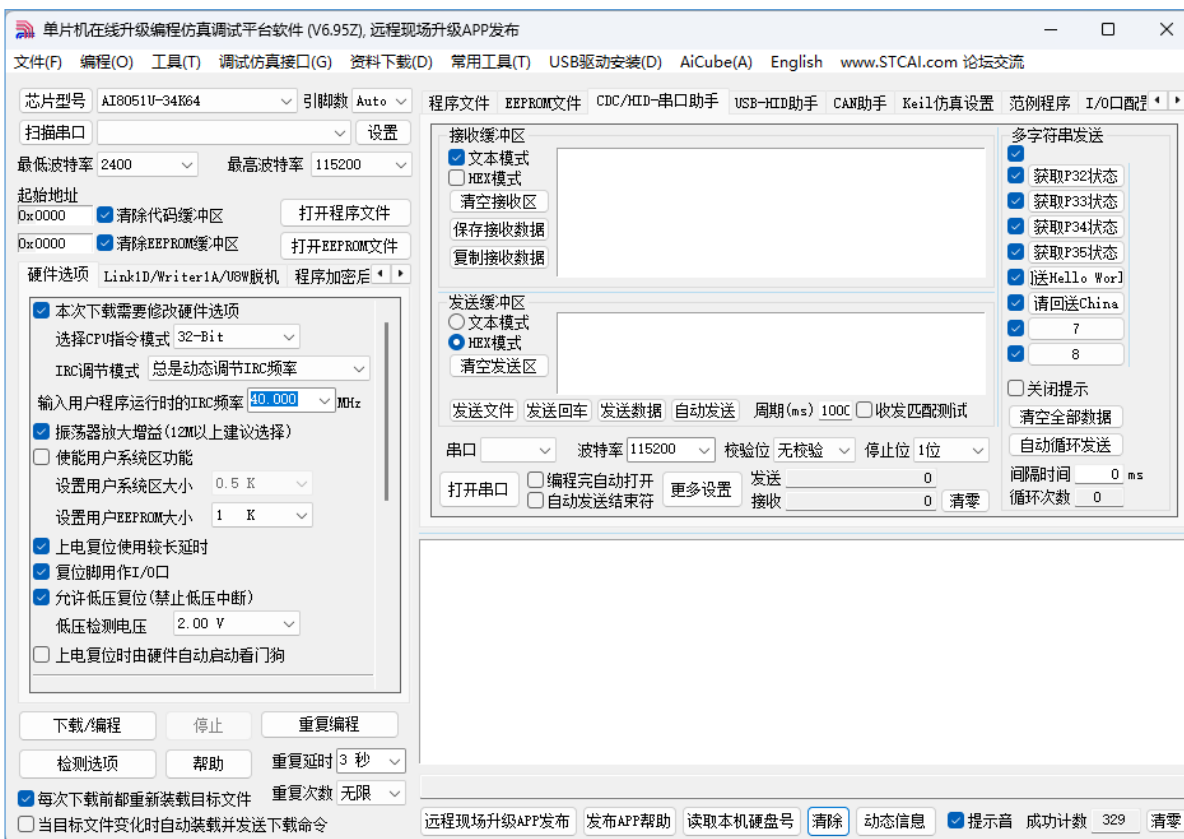


双击解压缩文件, AiCube-ISP 系统软件图标:



如果是解压缩到非桌面的其他目录, 如 D:\STCAI-ISP, 建议创建快捷方式到桌面方便打开。这实际是一个免安装的绿色软件。

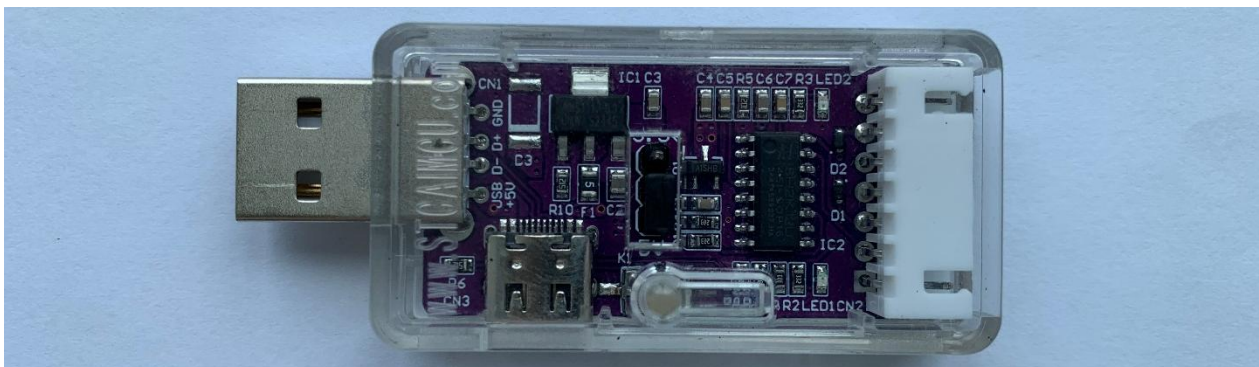
双击 AiCube-ISP 系统软件图标, 打开 AiCube-ISP 软件界面, 即可直接使用, 如下图:



1.1.2 Ai8051U 实验箱完整实物图（含外壳）：正面



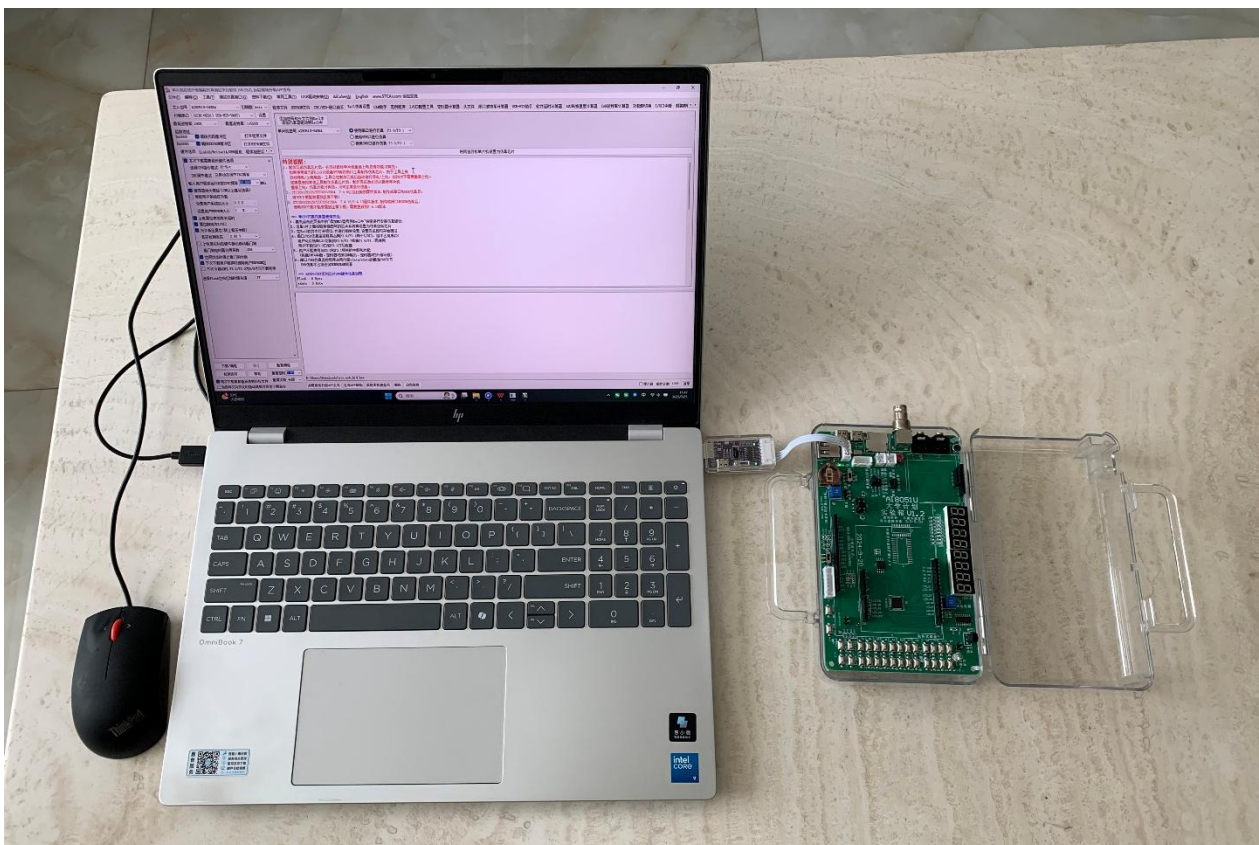
1.1.3 USB 转双串口工具@Ai8H2K12U: 正面



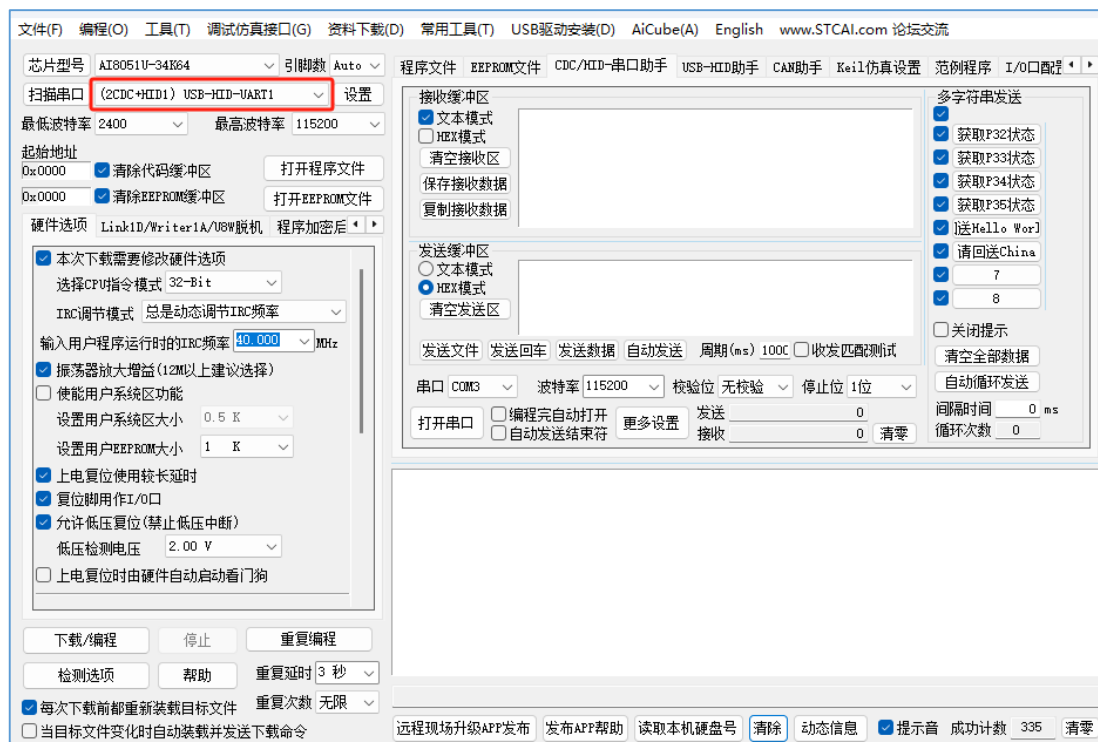
1.1.4 USB 转双串口/TTL 连接线: 正面



1.1.5 Ai8051U 实验箱通过 USB 转双串口工具与电脑连接实物图



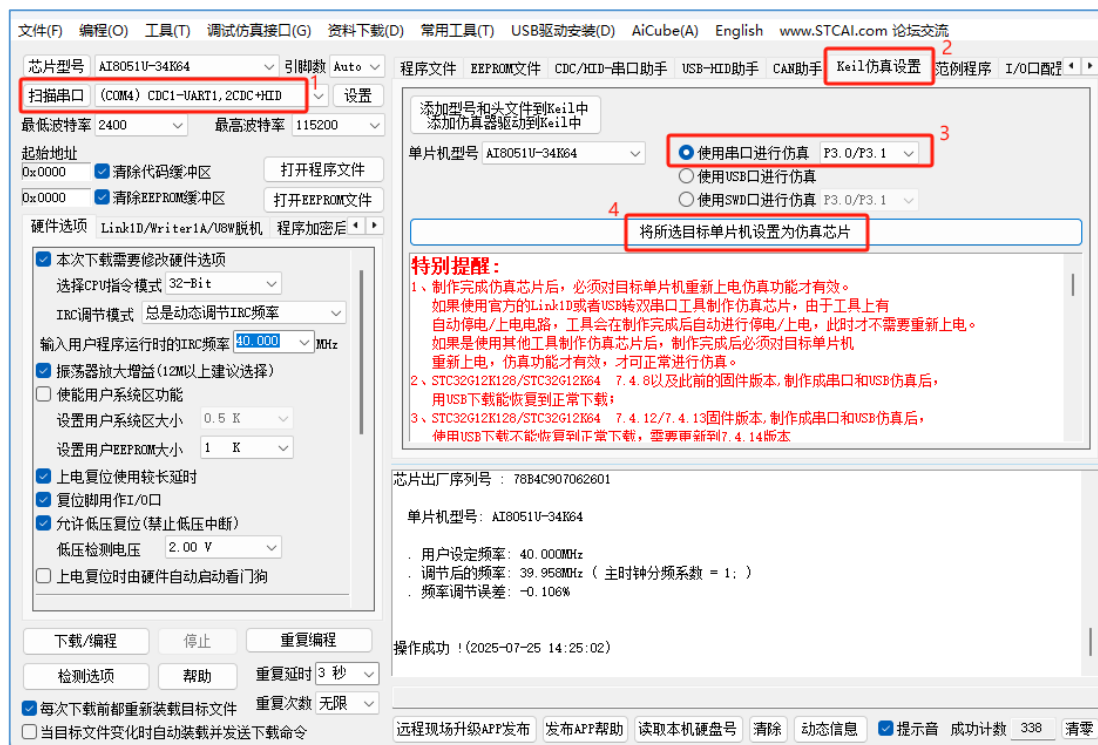
现在运行 AiCube-ISP-v6.95Z 或以上版本。选择单片机型号: Ai8051U-34K64, 如下图, 我们看到“扫描串口”后面的端口列表框显示:“(2CDC+HID1) USB-HID-UART1”, 如下图。



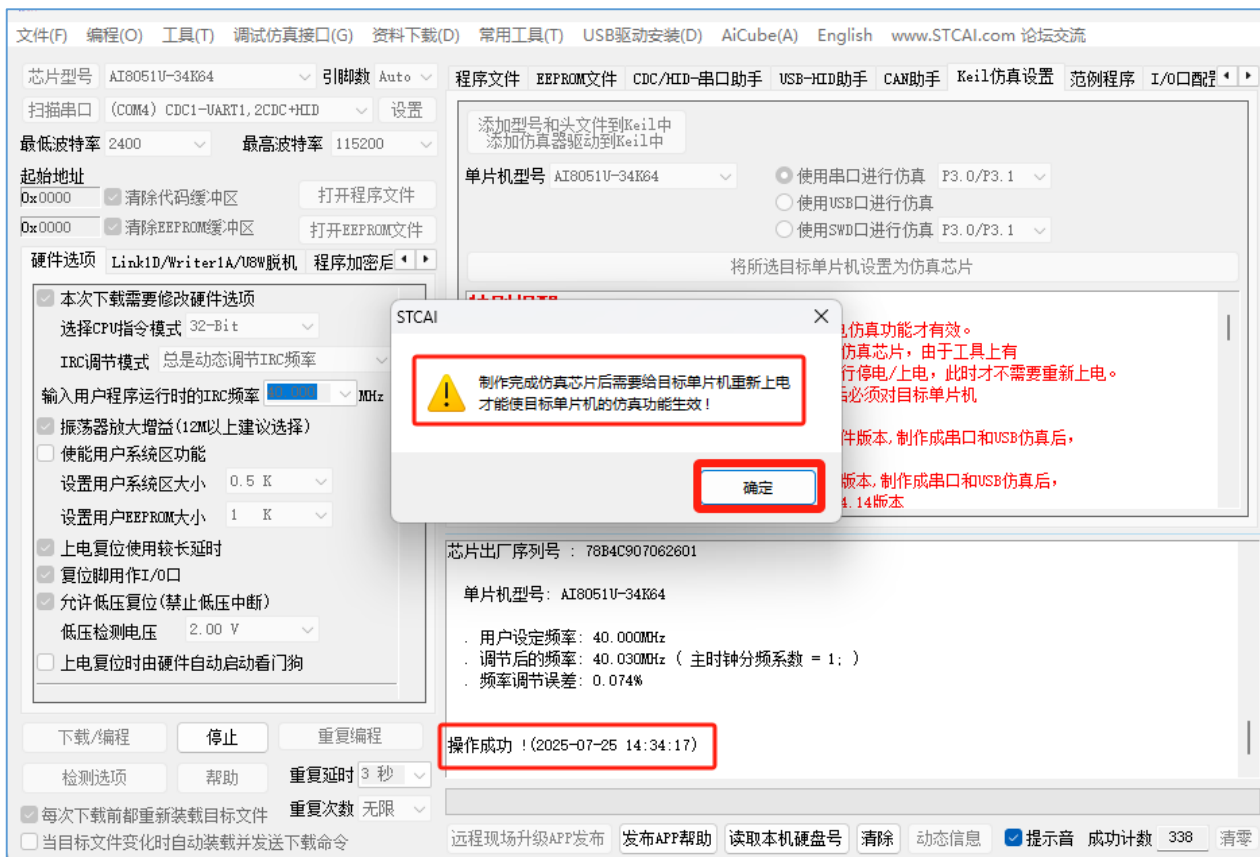
1.1.6 制作仿真芯片

在 AiCube-ISP 系统软件界面:

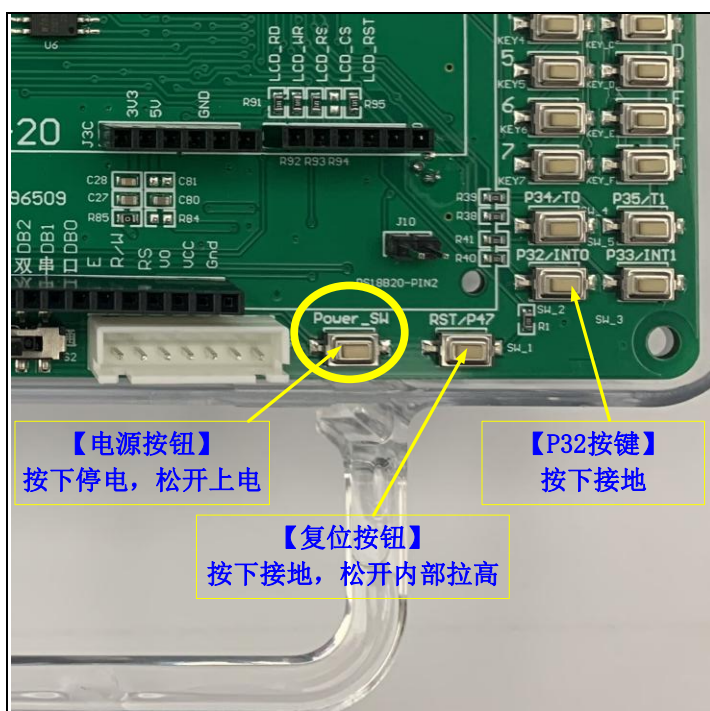
- 在“扫描串口”后面的端口列表框选择“(COM4) CDC1-UART1, 2CDC+HID”。COM 编号会因电脑不同而分配不同的串口号, 但必须选择 CDC1-UART1 对应的串口号。
- 点击“Keil 仿真设置”标签, 在“Keil 仿真设置”选项卡:
 - 选择“使用串口进行仿真”: 默认为“P3.0/P3.1”
 - 点击“将所选目标单片机设置为仿真芯片”



然后 AiCube-ISP 软件自动进行设置仿真芯片的操作，界面下面的进度条在推进，最后“操作成功!”界面出现特别提示：“制作完成仿真芯片后需要给目标单片机重新上电才能使目标单片机的仿真生效!”如下图：



所以，点击“确定”按钮后，按提示要求按一下 Ai8051U 实验箱上的“Power_SW”按钮，给目标单片机重新上电才能使目标单片机的仿真生效！下图为 Ai8051U 实验箱正面右下角放大图：

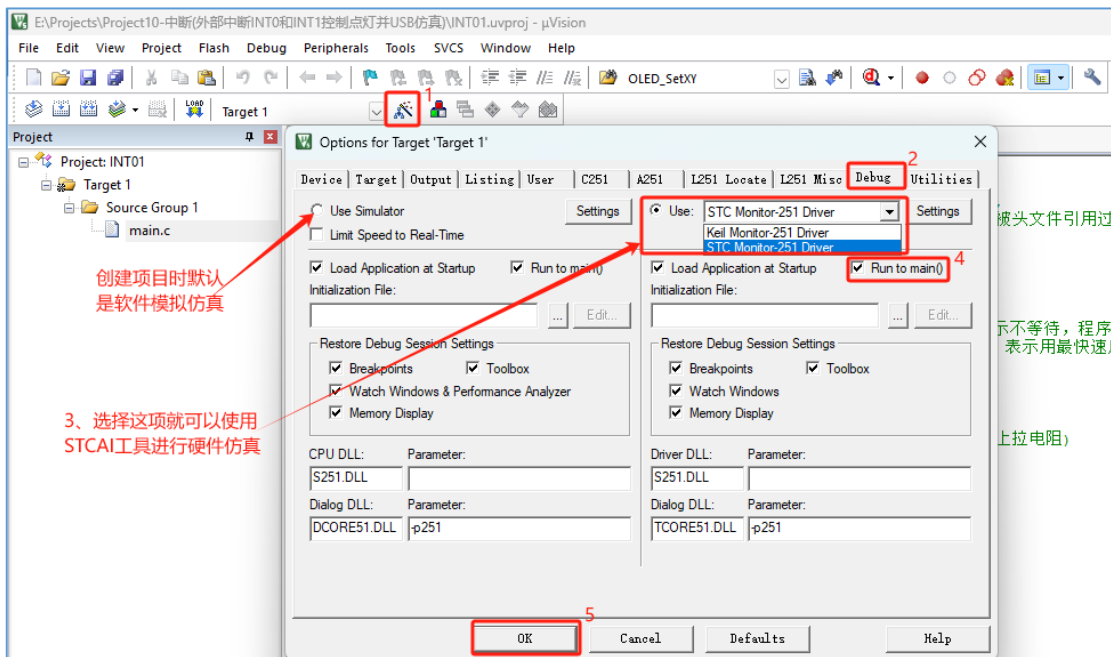


1.2 Debug 和 Utilities 选项设置--在 Keil 集成环境中

仿真芯片制作完成后, 打开要仿真的项目 Keil 文件, 在菜单栏下方找到一个类似魔术棒的工具, 点击打开项目选项设置界面, 在 Keil 集成环境中进行与 Debug 相关的必要设置。

1.2.1 Debug 选项卡--使用 STCAI 工具进行硬件仿真设置

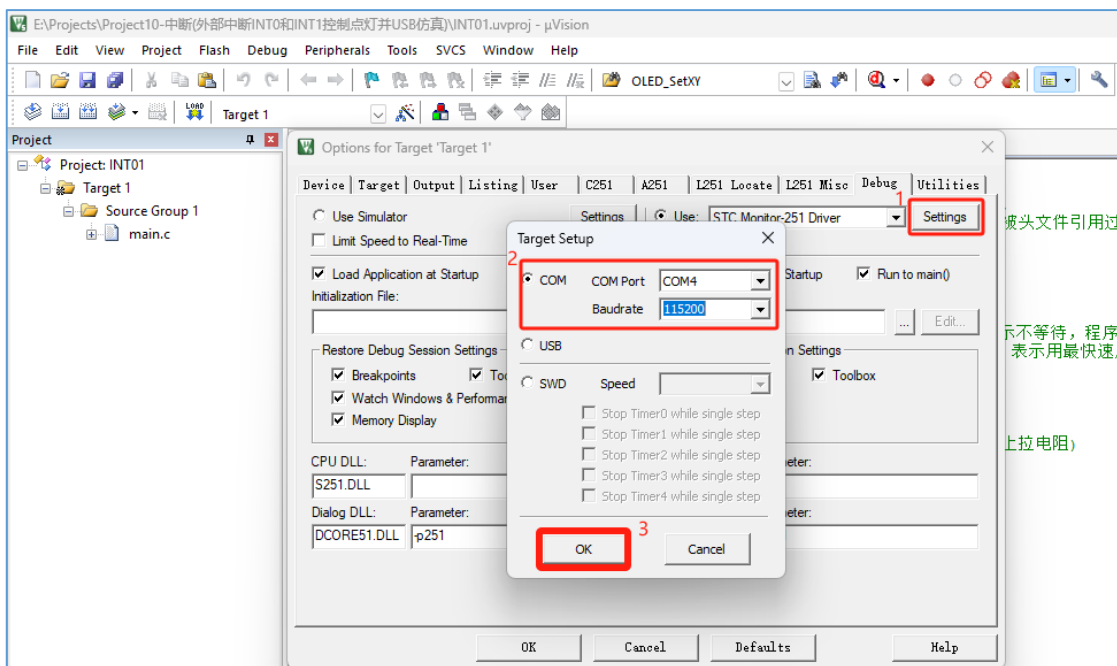
切换到 Debug 选项卡, 创建项目时, Keil 默认是软件模拟仿真。我们可以借助 STCAI 的工具进行硬件仿真, 如下图:



Ai8051U 如工作在 Ai8051U-32Bit 模式, 可以 USB 仿真和 COM 串口进行仿真, SWD 仿真暂不支持; Ai8051U 如工作在 Ai8051U-8Bit 模式, 暂不支持任何仿真。

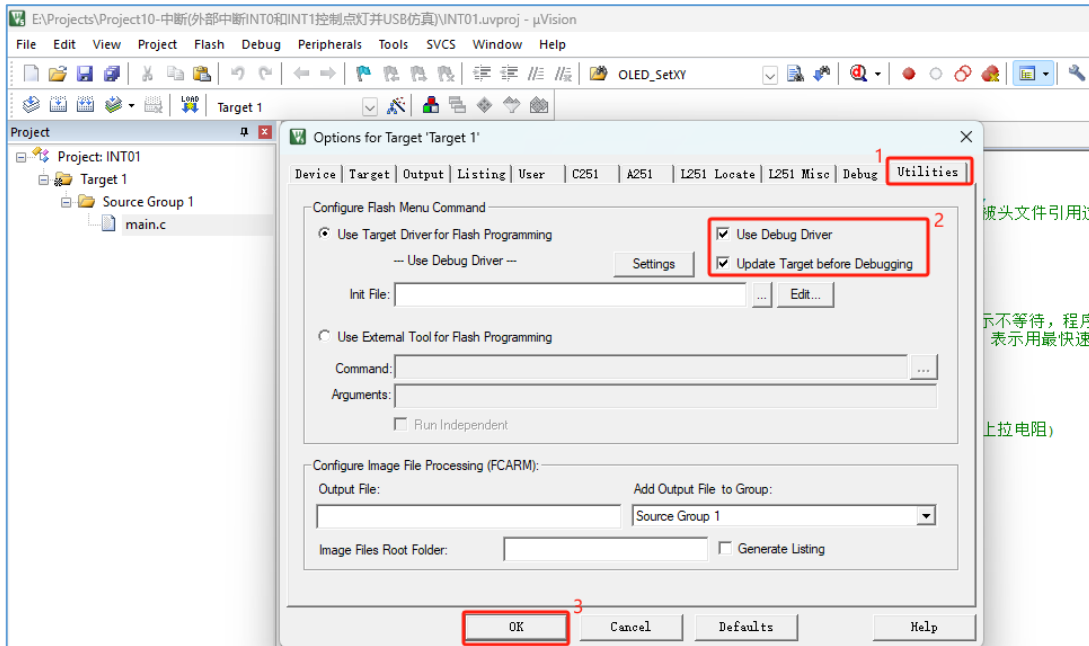
本文介绍 Ai8051U 实验箱通过 USB 转双串口工具进行 **串口仿真**, 如下图设置, 点选“COM”:

- ➡ COM Port: COM4 (与 AiCube-ISP 界面中“扫描串口”端口列表框中 CDC1 对应的 COM4 一致)
- ➡ Baudrate: 115200



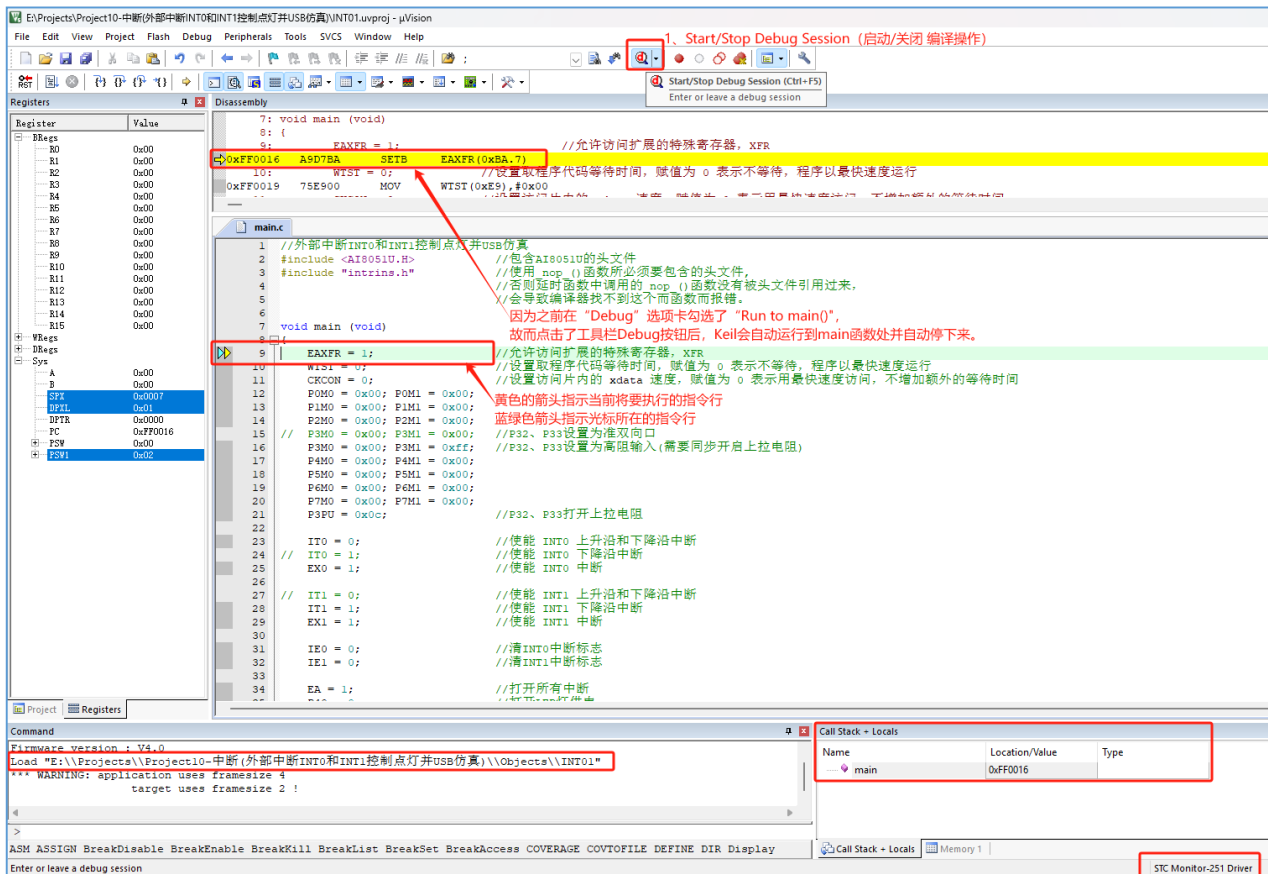
1.2.2 Utilities 选项卡--勾选“Use Debug Driver”

切换到“Utilities”选项卡，勾选“Use Debug Driver”，点击“OK”，完成设置。



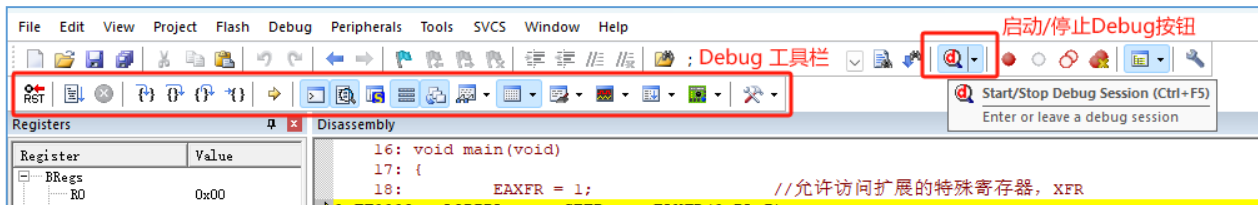
1.3 了解 Keil 中关于 Debug 的相关常用操作

设置完毕，返回 Keil 界面，代码程序编译无误后，点击工具栏的“Debug”（启动/关闭编译调试操作界面）按钮后，Keil 软件进入仿真界面，如下图：

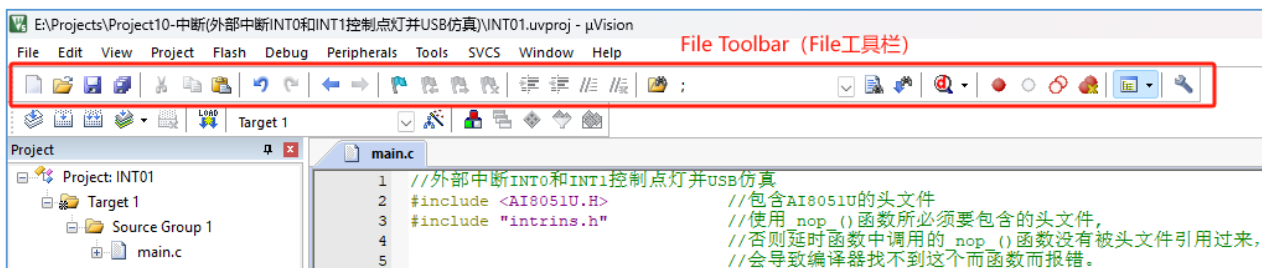


提示：退出 Debug 调试界面，同样点击工具栏的“Debug”按钮。

在 Debug 界面，我们要认识一下 **Debug Toolbar (Debug 工具栏)** 的常用调试/仿真按钮：

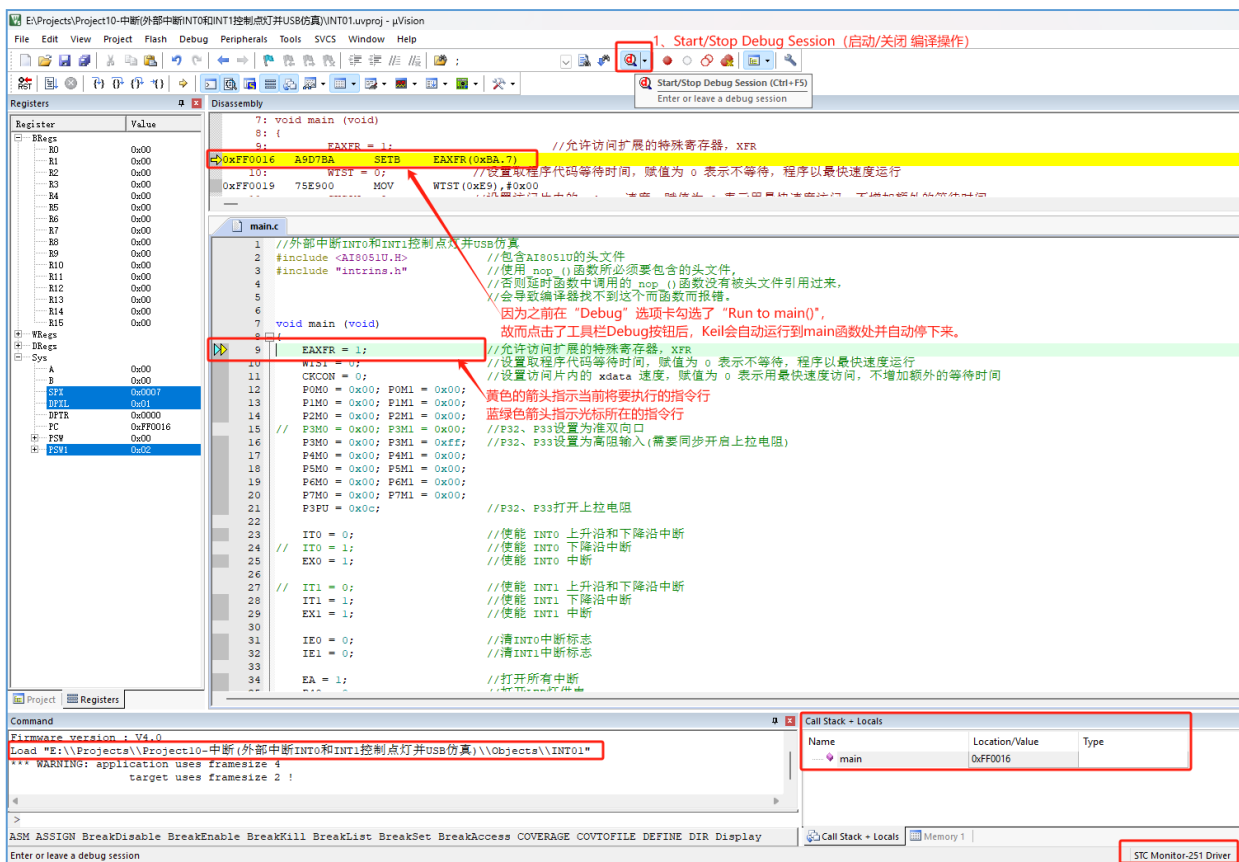


和 **File Toolbar (File 工具栏)** 中在调试/仿真时的几个常用按钮：



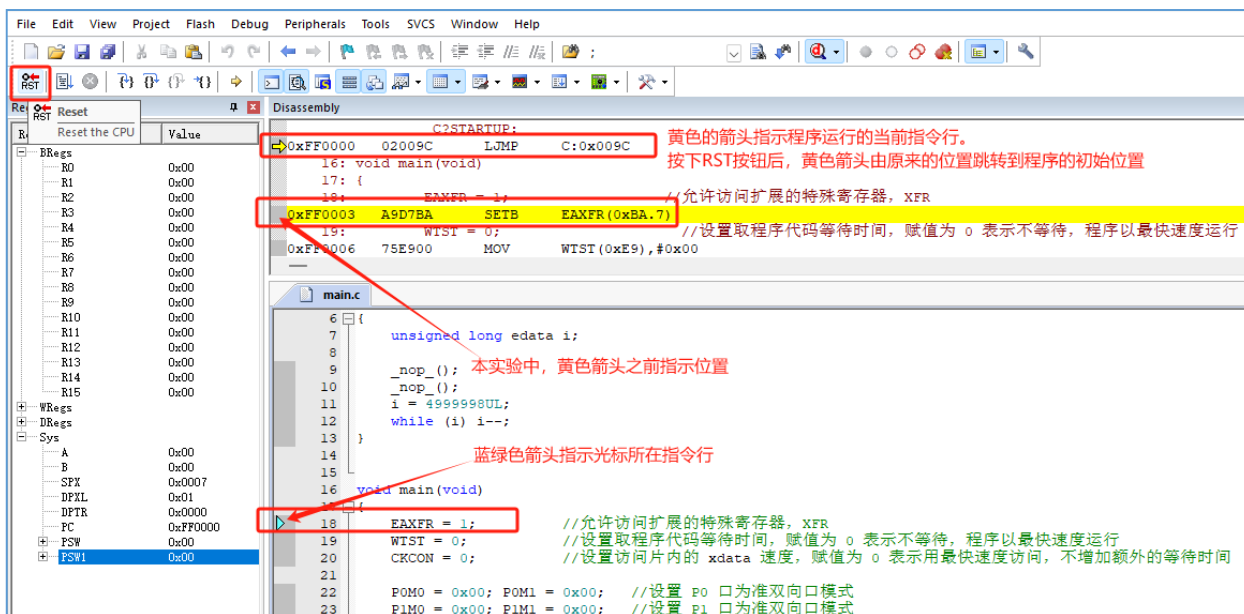
1.3.1 Start/Stop Debug Session, File Toolbar

Start/Stop Debug Session (Enter or leave a debug session) 按钮： 启动/停止 Debug 按钮。在软件开发过程中，调试是不可或缺的一环。Keil 提供了丰富的 Debug 调试工具，帮助我们快速定位和解决程序中的问题。点击此按钮，即刻进入 Debug 界面，如下图。再次点击此按钮，就会退出 Debug 界面。



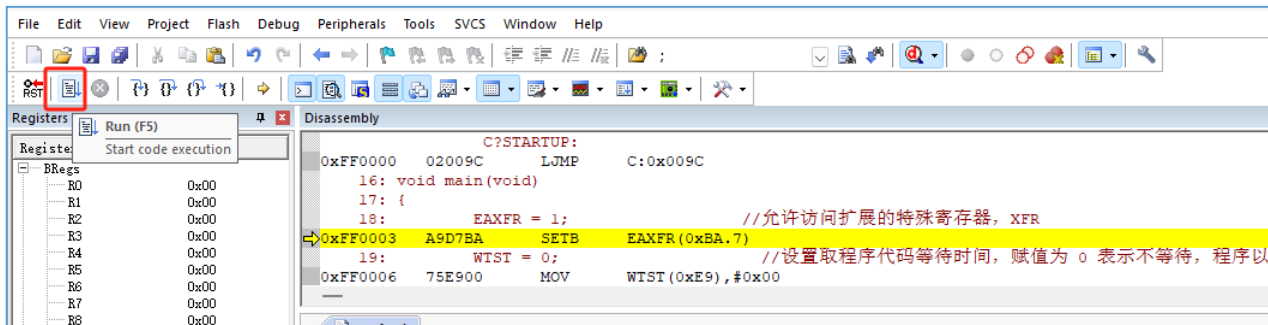
1.3.2 RST (Reset the CPU), Debug Toolbar

RST (Reset the CPU): 类似复位按钮, 相当于实现了一次硬复位。按下该按钮之后, 无论程序代码执行到哪个位置, 代码都会重新从头开始执行。



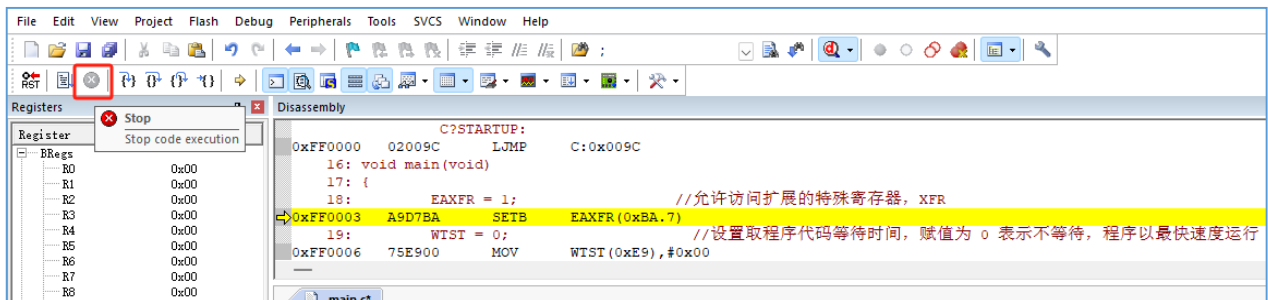
1.3.3 Run (Start code execution), Debug Toolbar

Run (Start code execution): 运行按钮。点击该按钮后, 程序将一直处于运行状态, 或者直接快速运行到断点处 (前提是你在查看的地方设置了断点)。有时候你并不需要观看程序的每步是怎么执行的, 而是想快速的执行到程序的某个地方看结果, 这个按钮就可以实现这样的功能。



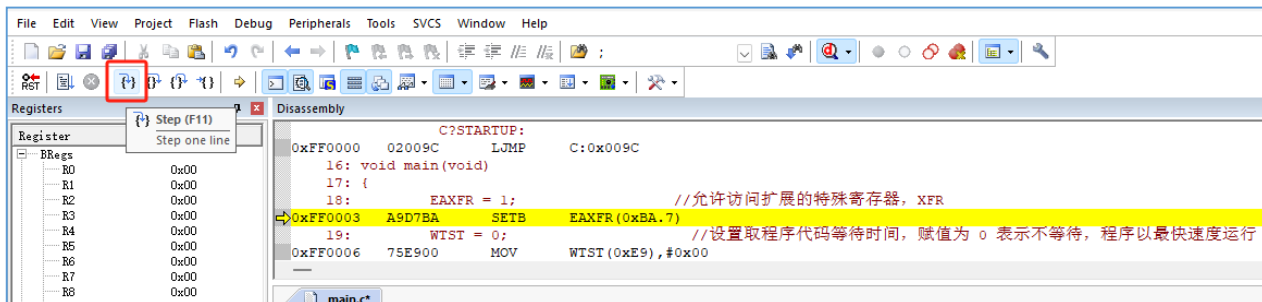
1.3.4 Stop (Stop code execution), Debug Toolbar

Stop (Stop code execution): 停止运行按钮。此按钮在程序一直执行的时候会变为有效, 通过按该按钮, 就可以使程序停止下来, 进入到单步调试状态。



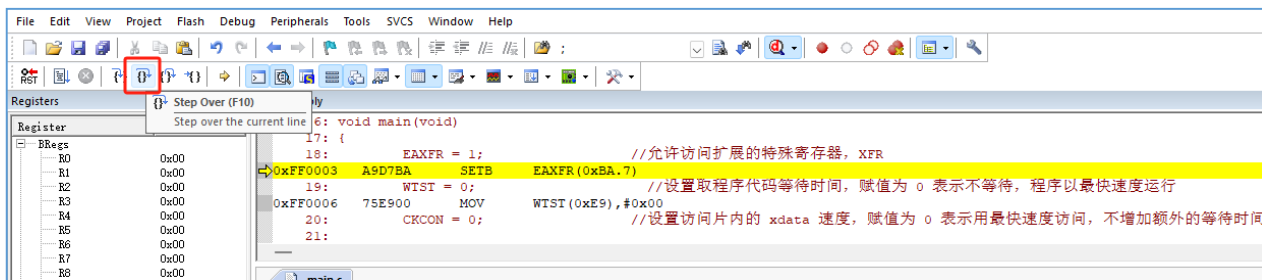
1.3.5 Step (Step one line) , Debug Toolbar

Step (Step one line): 单步运行按钮。每点击一次该按钮，程序将执行单条语句。如果遇到函数，则会进入函数内部执行。这是调试程序时最常用的功能之一。此按钮在程序停止运行时有效。



1.3.6 Step Over (Step over the current line) , Debug Toolbar

Step Over (Step over the current line): 单步跳过按钮，程序运行一行，按单条语句执行。与 Step 按钮不同的是，Step Over 按钮遇到函数不会进入函数内部，而是直接全速运行函数，并跳到下一条语句。在碰到有函数（特别是延时函数）的地方，通过该按钮就可以单步执行过这个函数，而不进入这个函数单步执行。此按钮在程序停止运行时有效。

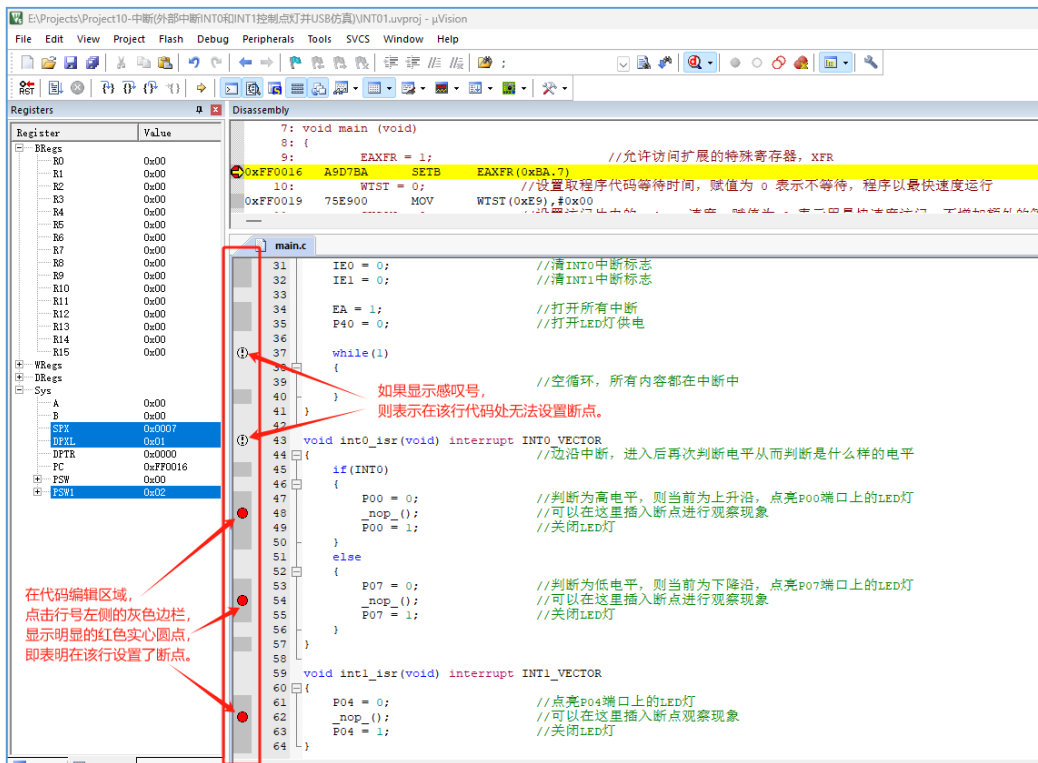


1.3.7 Breakpoint, File Toolbar

在调试/仿真程序时，会涉及一个很重要的工具：Breakpoint（断点）。

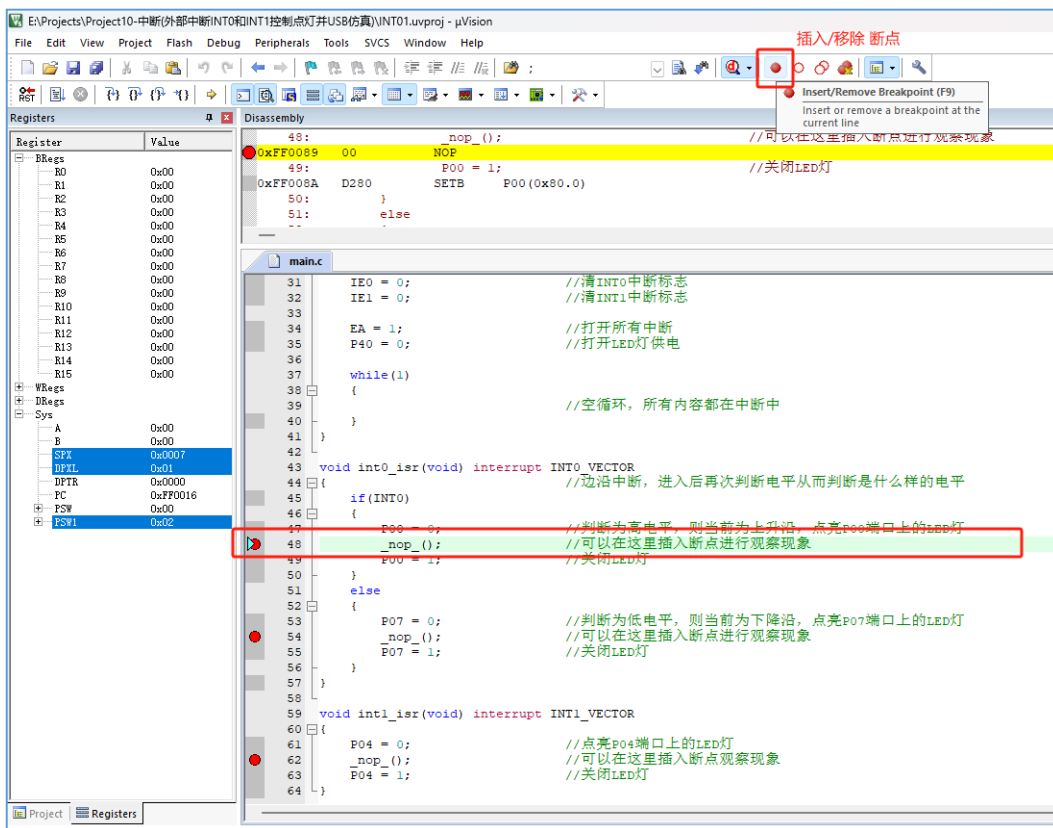
Breakpoint: 断点。在 Keil 中，断点是一种强大的调试工具。断点本身是程序中的一个位置标记，当程序执行到断点位置时，程序会暂停执行，从而方便我们观察实验现象，定位、分析和排查程序中的问题等。在 Keil 环境中，设置断点有多种方式，下面我们介绍最简单的 2 种断点设置方法。

1、在代码编辑区域，点击行号左侧的灰色边栏，显示明显的红色实心圆点，即表明在该行代码处插入了断点。如果显示感叹号，则表示在该行代码处无法设置断点。在断点处点击断点（红色实心圆点）即为取消断点。



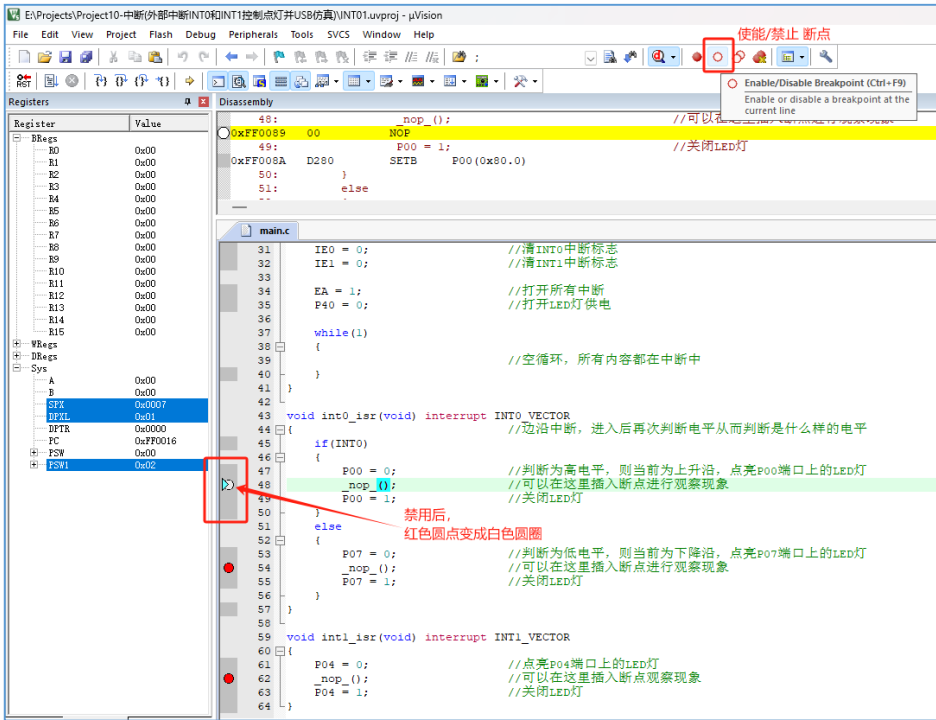
1.3.8 Insert/Remove Breakpoint, File Toolbar

2、Insert/Remove Breakpoint (Insert or remove a breakpoint at the current line) 按钮: 在当前光标所在行插入/移除断点。在当前光标所在行, 该按钮如果显示红色, 则表示可在该处插入断点。点击该按钮, 即在当前行插入断点, 当前行左侧就会显示一个红色圆点表示断点位置。该按钮如果显示灰色, 则无法在该处插入断点。定位光标在断点所在代码行, 点击该按钮, 即移除该处断点, 红色圆点消失。



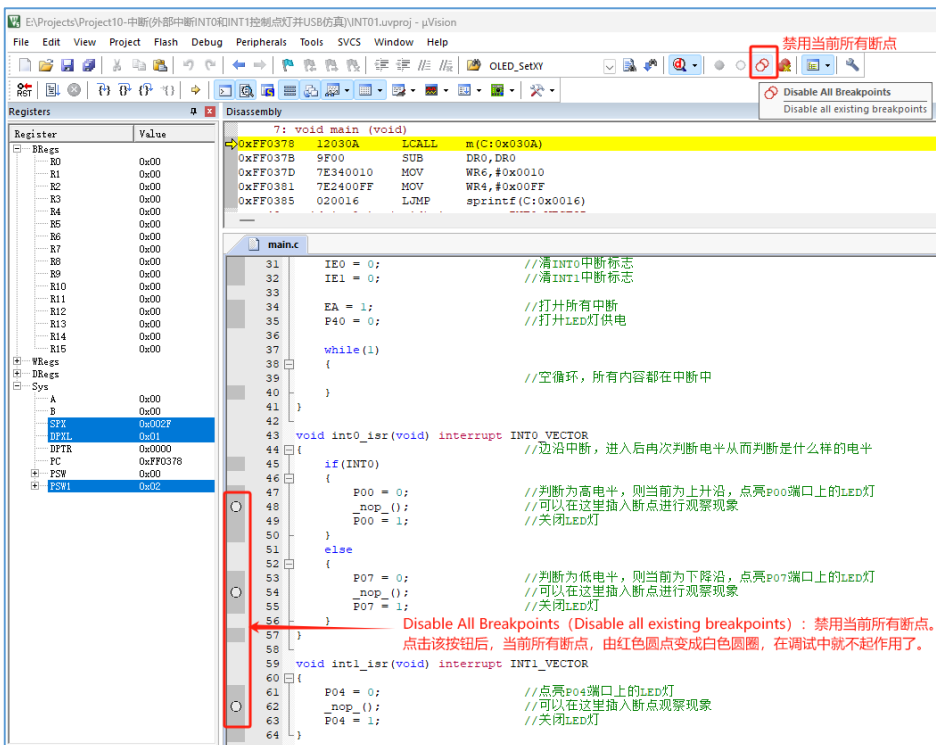
1.3.9 Enable/Disable Breakpoint, File Toolbar

Enable/Disable Breakpoint (Enable or disable a breakpoint at the current line): 使能/禁止断点。启用/禁用当前光标所在行的断点。断点在启用时, 红色圆点显示。禁用后, 红色圆点变成白色圆圈, 在调试中就不起作用了。断点使能/禁止按钮, 在设置断点后, 你可以使用此功能按钮来启用或禁用断点, 从而控制代码的暂停与继续。



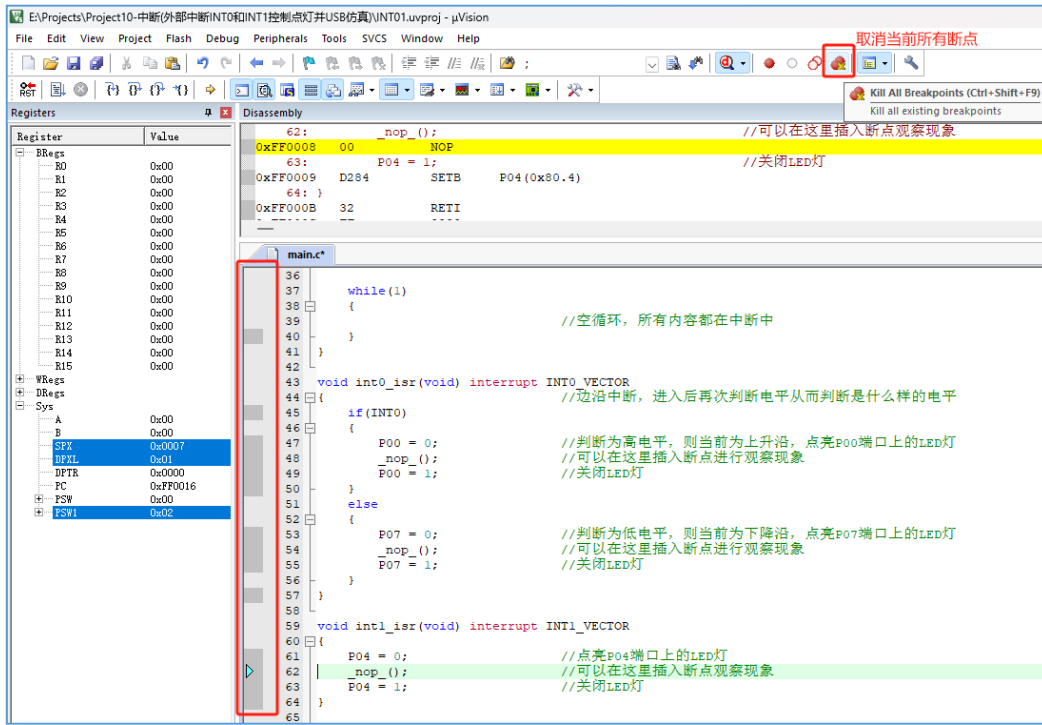
1.3.10 Disable All Breakpoints, File Toolbar

Disable All Breakpoints (Disable all existing breakpoints): 禁用/失能当前所有断点。点击该按钮后, 当前所有断点, 由红色圆点变成白色圆圈, 在调试中就不起作用了。但此按钮不能再次使能所有断点。

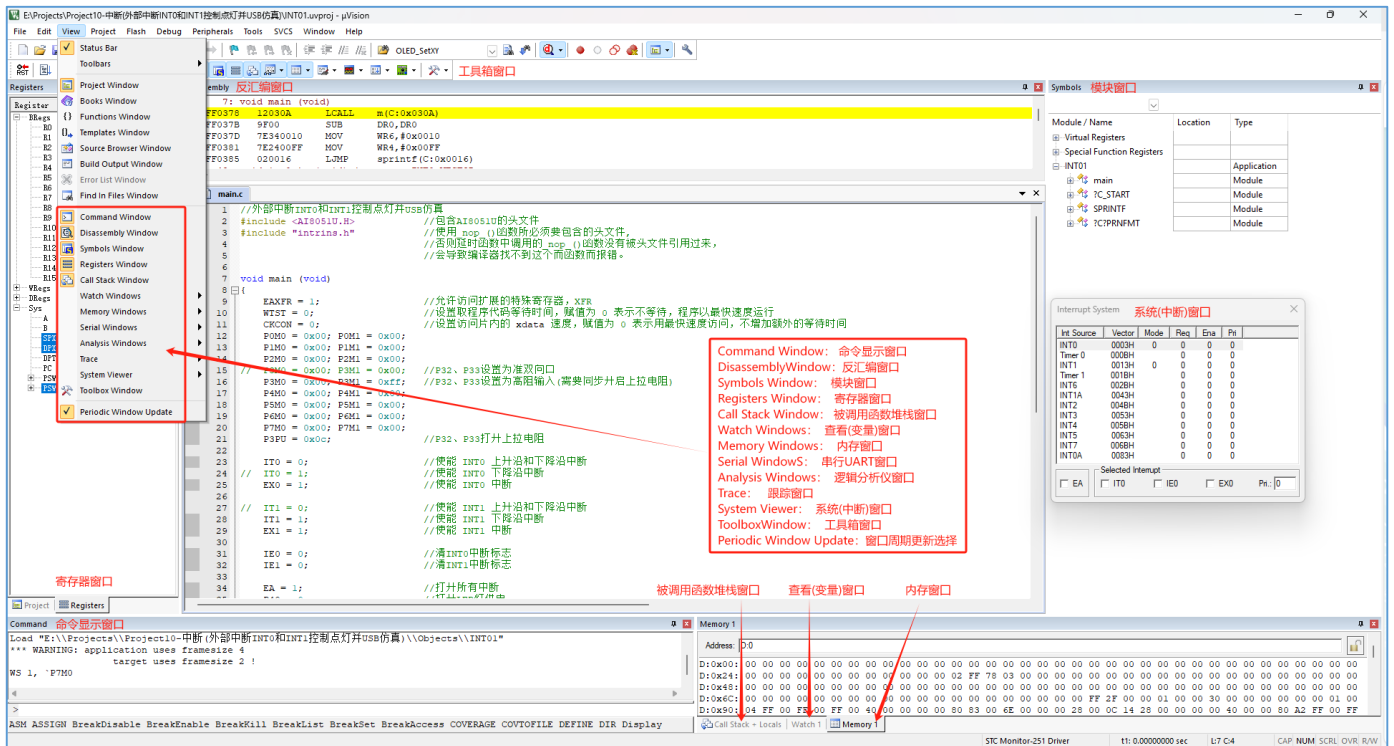


1.3.11 Kill All Breakpoints, File Toolbar

Kill All Breakpoints (Kill all existing breakpoints): 取消当前所有断点。和 **Disable All Breakpoints** 不同, 点击该按钮后, 所有断点标记消失。

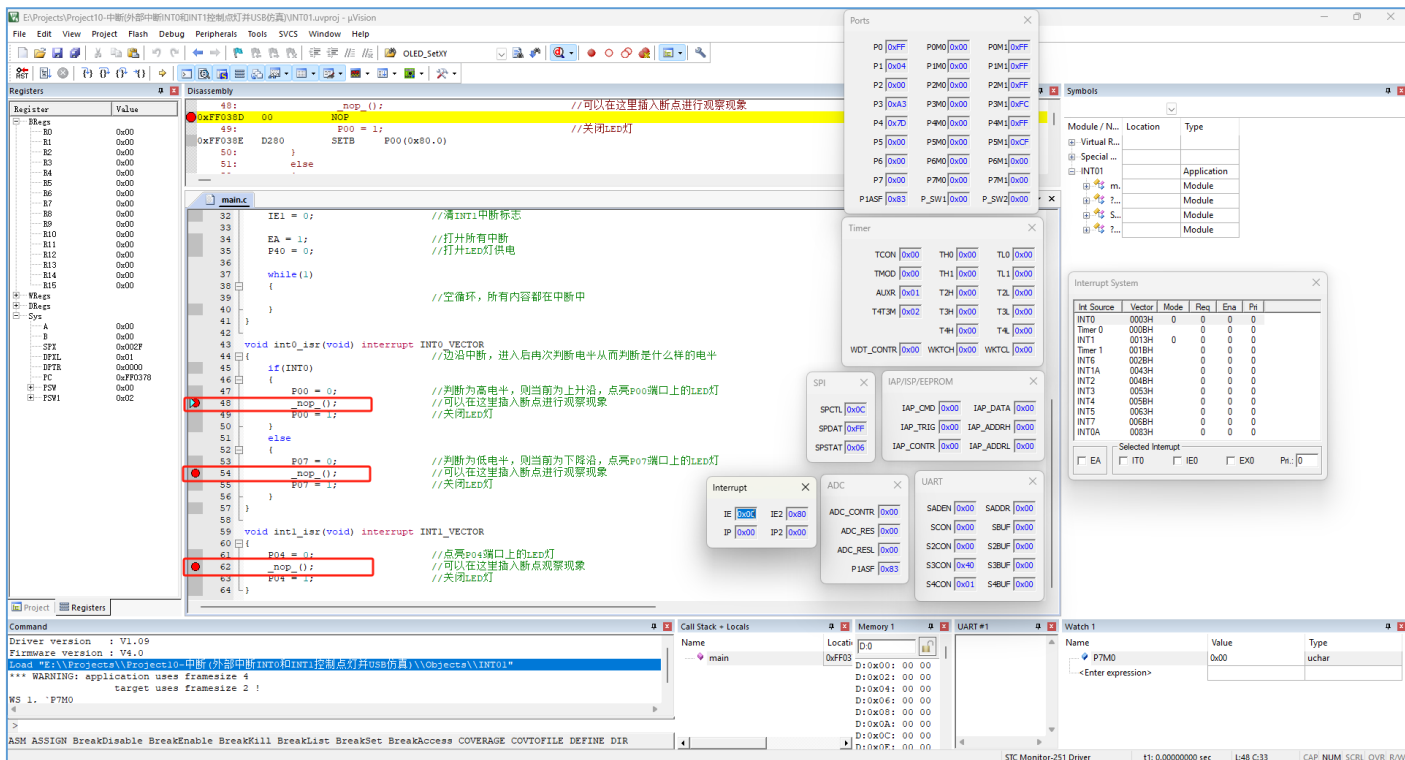


1.3.12 观察变量/寄存器/内存 (数据存储区、程序存储区) 窗口

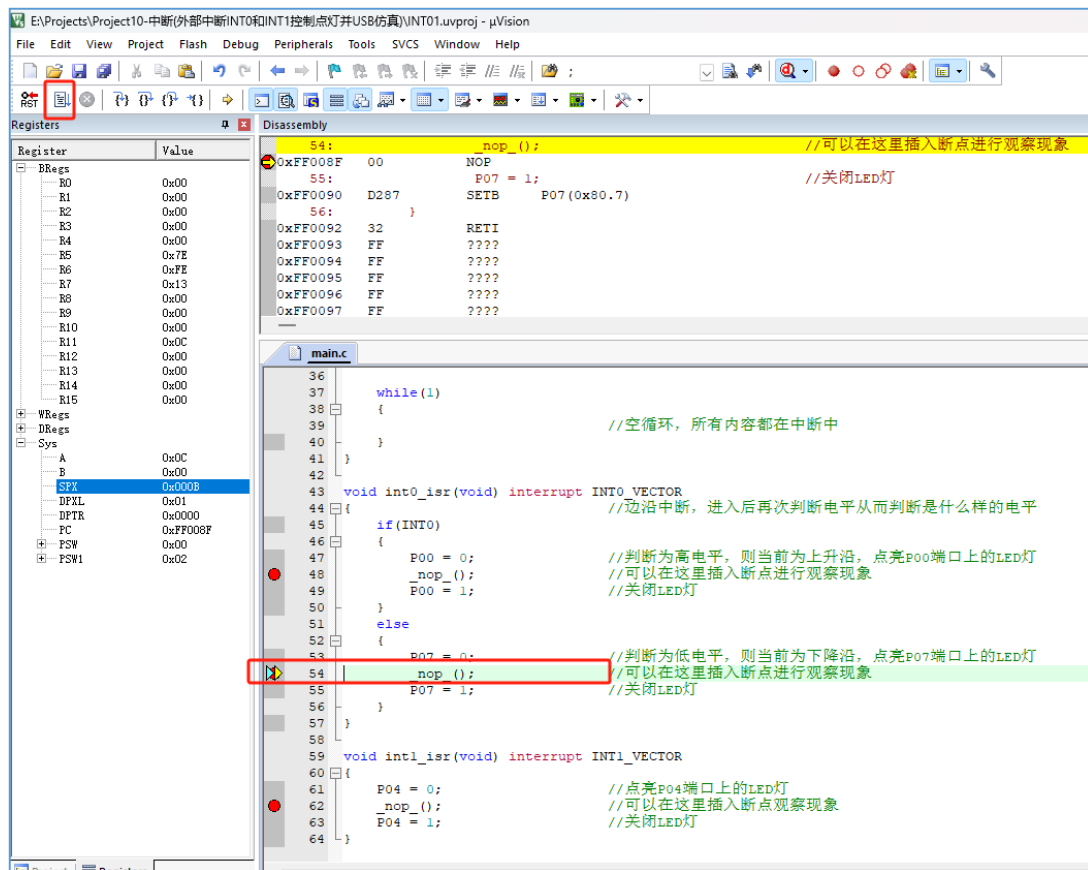


1.4 观察实验现象

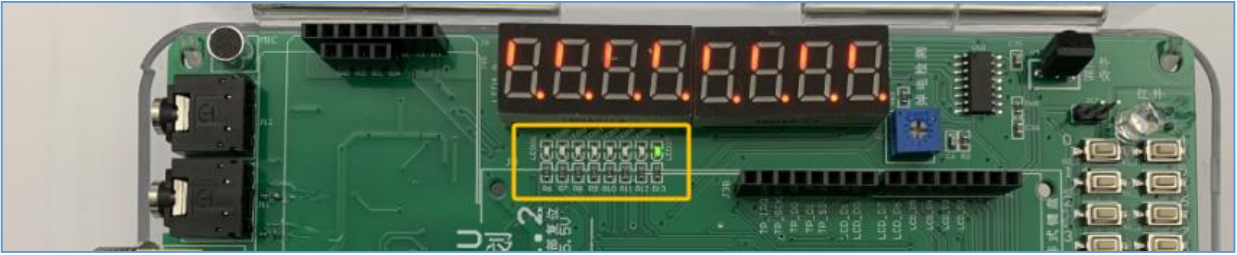
步骤一：利用仿真，可以设置运行到断点处，捕捉当前程序运行到哪，INT0/P3.2 是下降沿还是上升沿为方便观察实验现象，需在中断服务程序中设置如下 3 个断点：



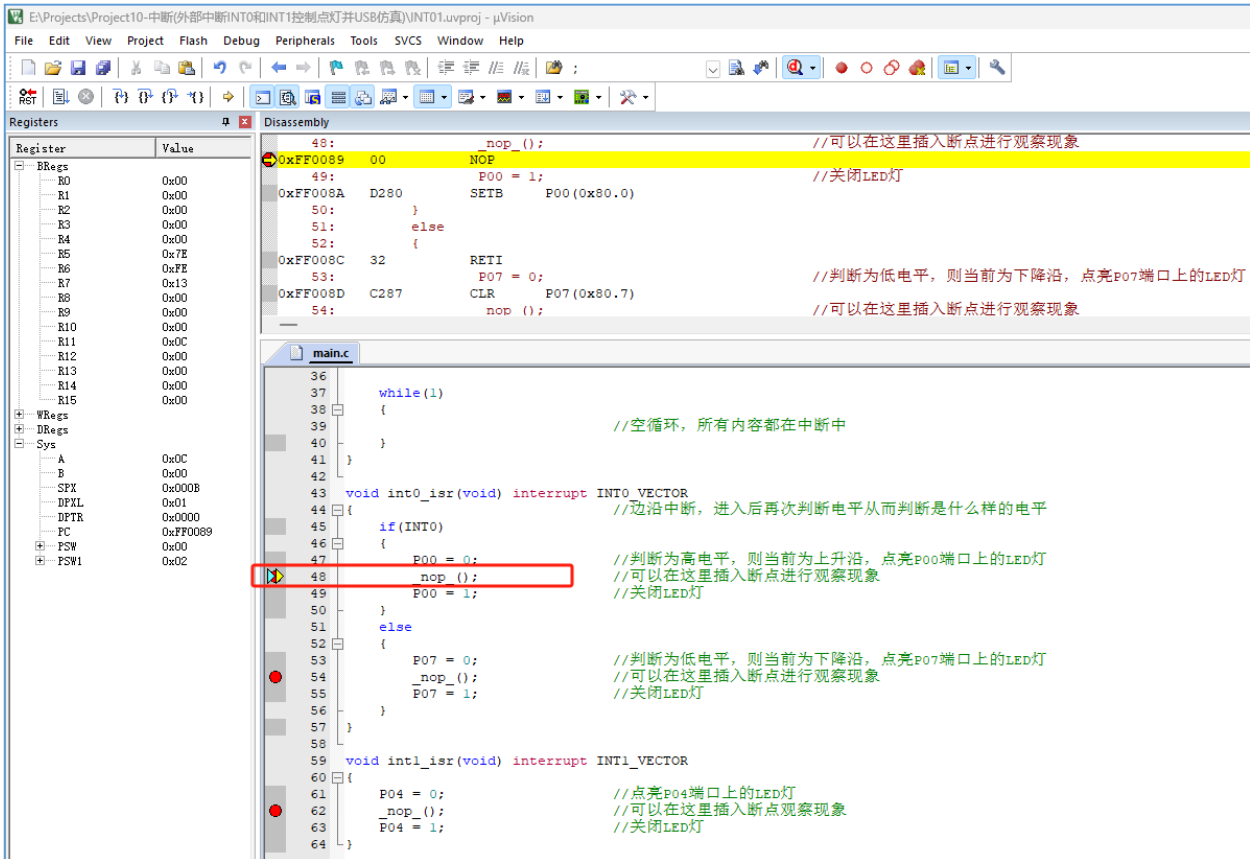
先点击 Run（运行）按钮，按下 P3.2 按键，保持不松开（为了验证下一步的实验），程序会停在如下图所示断点处：



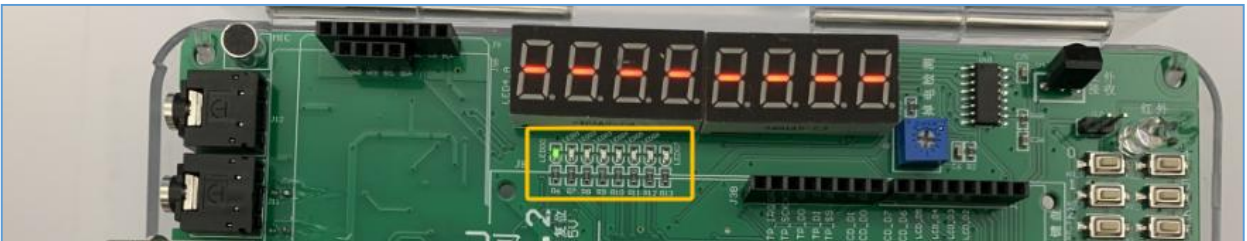
进 INTO 中断后, 读到 INTO/P3.2 口是 低, 下降沿, 点亮 P0.7, 实验箱如图:



继续点击 Run (运行) 按钮运行, 退出中断服务程序。再松开 P3.2 按键, 程序会停在如下图所示断点处:



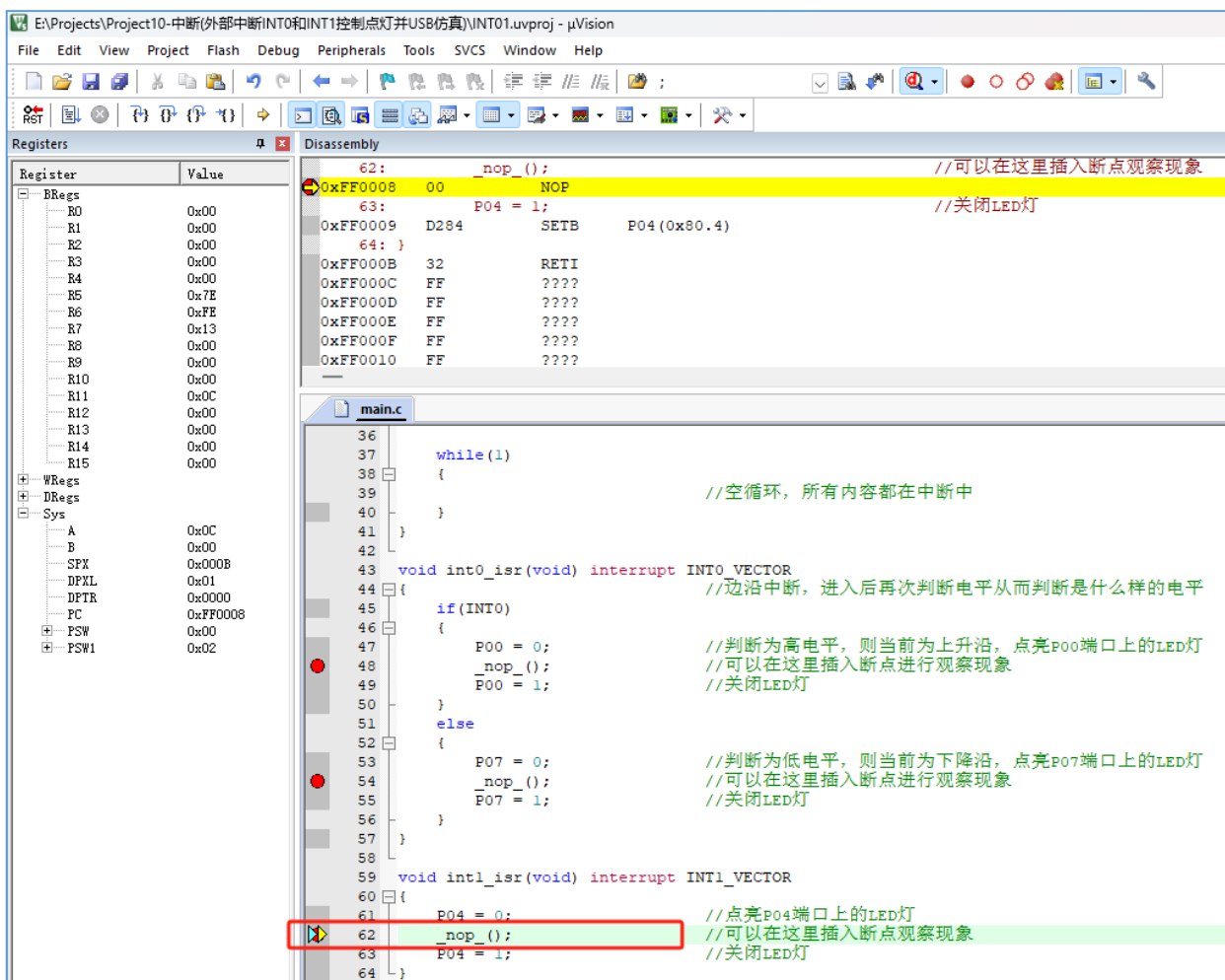
进 INTO 中断后, 读到 INTO/P3.2 口是 高, 上升沿, 点亮 P0.0, 实验箱如图:



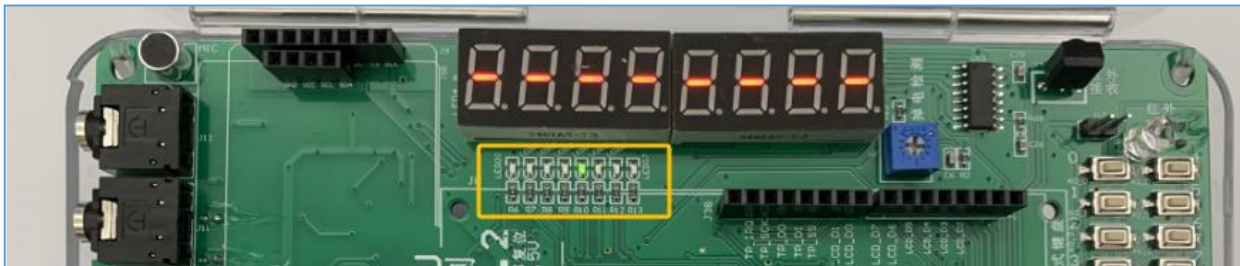
继续点击 Run (运行) 按钮运行, 退出中断服务程序。

步骤二：利用仿真，可以设置运行到 INT1/P3.3 外部中断的断点处

如果程序没有在运行，需先点击 Run（运行）按钮，按一下 P3.3 按键，程序会停在如下图所示断点处：



进 INT1 中断后，点亮 P0.4，实验箱如图：



继续点击 Run（运行）按钮运行，退出中断服务程序。

完成本实验。