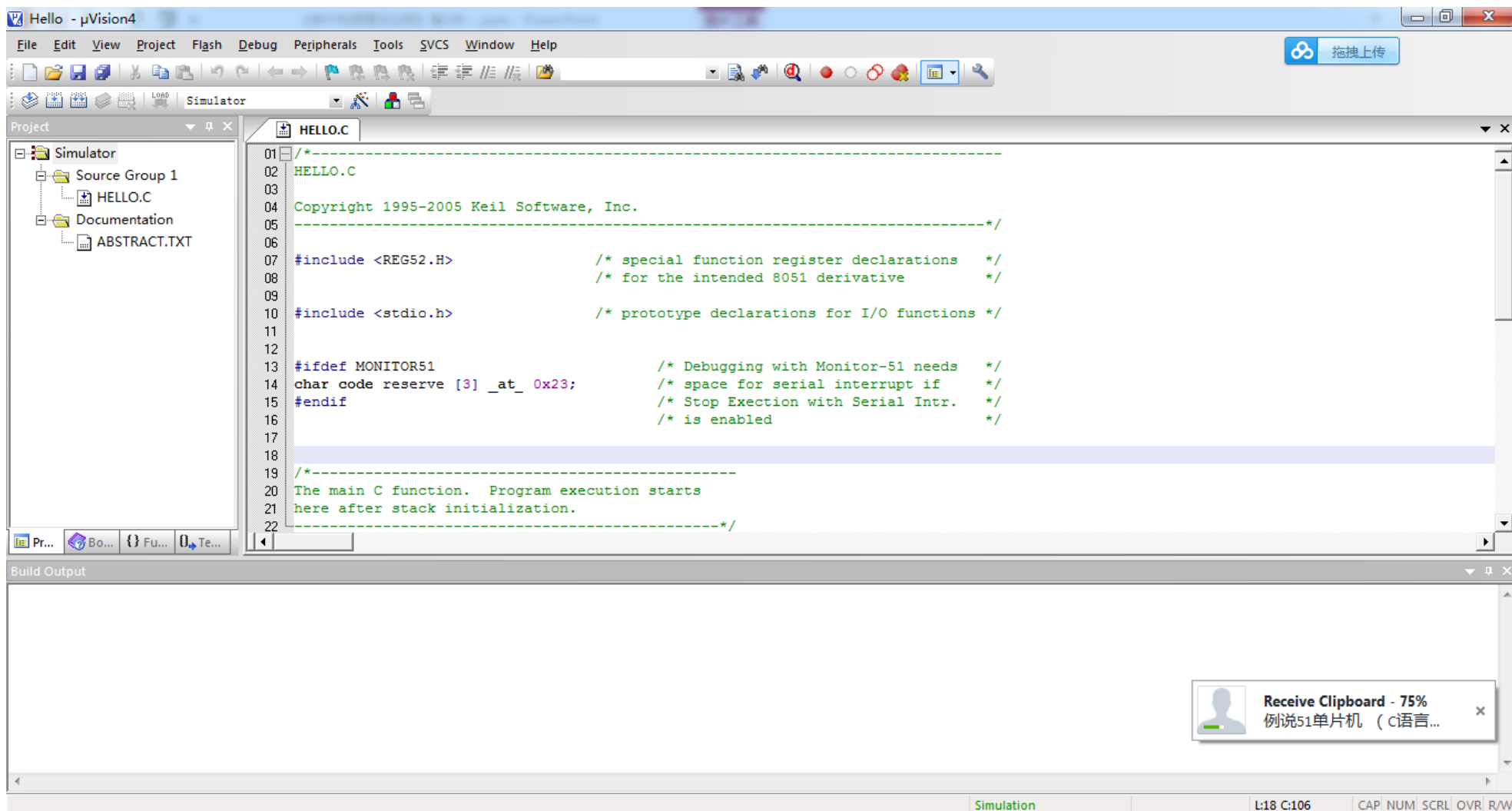


第二讲

Keil编程环境及软件编程



1. keil uVision 集成开发环境的使用



Keil Software 的8051开发工具可用于编译C源码、汇编源程序、连接和重定位标文件和库文件，创建HEX文件，调试目标程序。

- ▶ **uVision** 是一个集成开发环境它把项目管理源代码编辑程序调试等集成到一个功能强大的环境中。
- ▶ **C51**是标准优化C交叉编译器。从你的C源代码产生可重定位的目标文件
- ▶ **A51**宏汇编器。从你的8051汇编源代码产生可重定位的目标文件



- ▶ **BL51连接/重定位器**。组合你的由C51和A51产生的可重定位的目标文件生成绝对目标文件
- ▶ **LIB51库管理器**。组合你的目标文件生成可以被连接器使用的库文件。
- ▶ **OH51目标文件到HEX格式的转换器**。从绝对目标文件创建Intel HEX 格式的文件。
- ▶ **RTX-51实时操作系统**。简化了复杂和对时间要求敏感的软件项目。



测试版工具包和产品工具包

Keil Software把软件分成两种类型测试版和正式版

- ▶ **测试版**包括8051工具的测试版本和本用户手册你可以用它们产生目标代码小于2K字节的应用。
- ▶ **正式版**包括没有限制的8051工具和全套手册，正式版套件包含1年的免费技术支持和产品升级升级通过 www.keil.com提供。



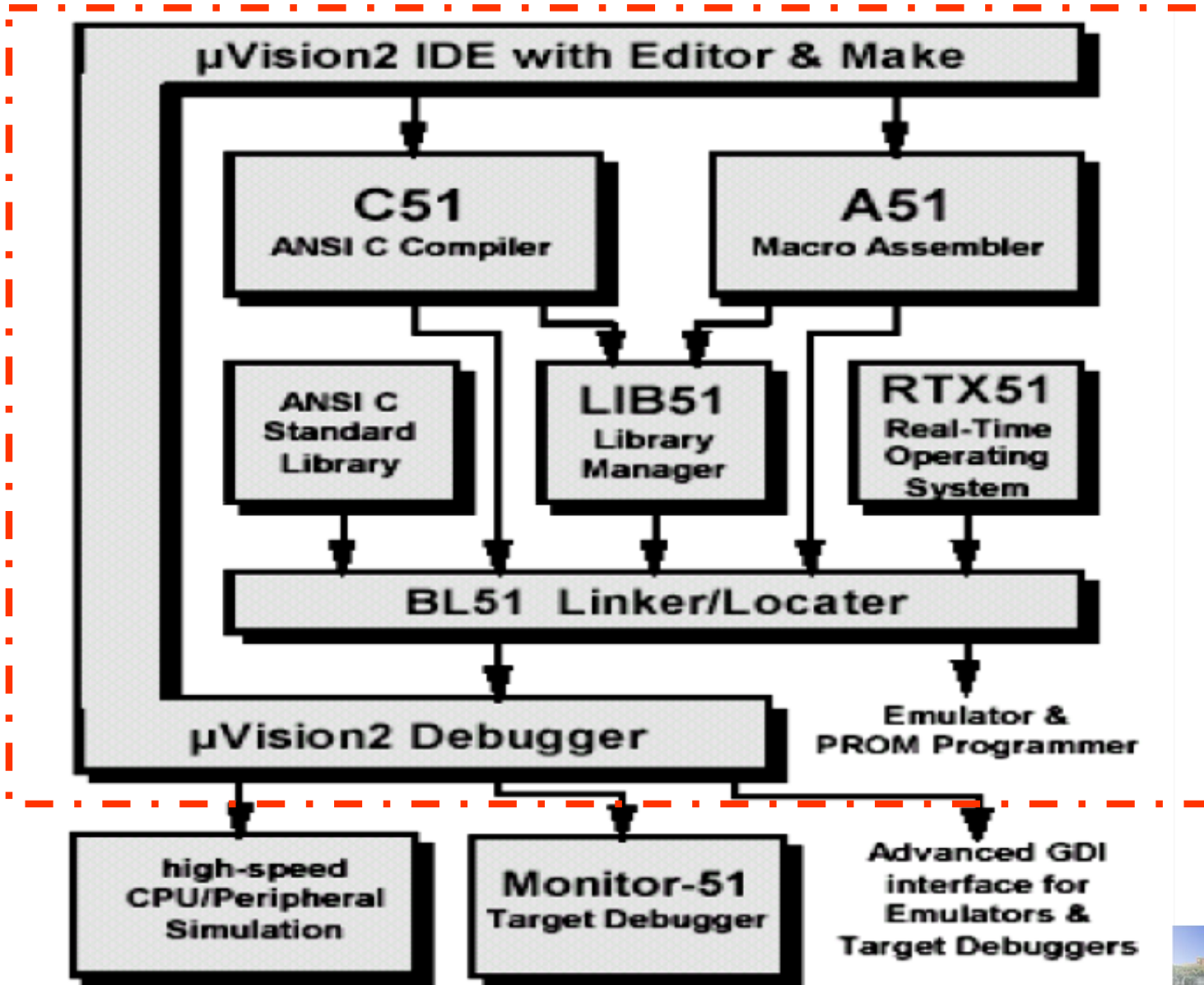
软件开发流程

使用Keil Software工具时你的项目开发流程和其它软件开发项目的流程极其相似。

- ▶ 创建一个项目，从器件库中选择目标器件，配置工具设置；
- ▶ 用C语言或汇编语言创建源程序；
- ▶ 用项目管理器构造应用；
- ▶ 修改源程序中的错误；
- ▶ 调试连接后的应用



一个完整的8051工具集的框图可以最好地表述此开发流程。



安 装

► 系统需求

❑ 为了取得比较好的运行效果最低的硬件和软件配置必须满足：

- ❑ 具有奔腾奔腾II或兼容的处理器个人计算机
- ❑ 操作系统为WIN95WIN98WINNT4.0或更高
- ❑ RAM大于16MB
- ❑ 20MB 的硬盘空余空间

► 安装详细说明

所有的Keil产品都带有一个安装程序。8051开发工具的安装步骤如下

- ❑ 插入Keil开发工具光盘
- ❑ 从CD浏览界面选择安装软件
- ❑ 跟随提示进行安装操作



► 文件夹组织结构

安装程序复制开发工具到基本目录的各个子目录中。
默认的基本目录是C:\KEIL。下表列出的文件夹结构是包括所有8051开发工具的全部安装信息。

C:\KEIL\C51\ASM	汇编SFR定义文件和模板源程序文件
C:\KEIL\C51\BIN	8051工具的执行文件
C:\KEIL\C51\EXAMPLES	示例应用
C:\KEIL\C51\RTX51	完全实时操作系统文件
C:\KEIL\C51\RTX_TINY	小型实时操作系统文件
C:\KEIL\C51\INC	C编译器包含文件
C:\KEIL\C51\LIB	C编译器库文件启动代码和常规I/O资源
C:\KEIL\C51\MONITOR	目标监控文件和用户硬件的监控配置
C:\KEIL\UV2	普通uVision2文件



uVision2 有两种工作模式：

- ▶ **Build模式(build mode)**：在这种工作模式中，建立工程、选择CPU、设置工程环境、创建源文件等、编译所有的应用文件，并生产可执行程序。
- ▶ **调试模式 (debug mode)**：使用调试器测试应用程序。



一、项目的建立、设置与编译Build模式(build mode):

基本步骤:

创建一个项目，从器件库中选择目标器件，配置工具设置；

用C语言或汇编语言创建源程序；

用项目管理器构造应用；

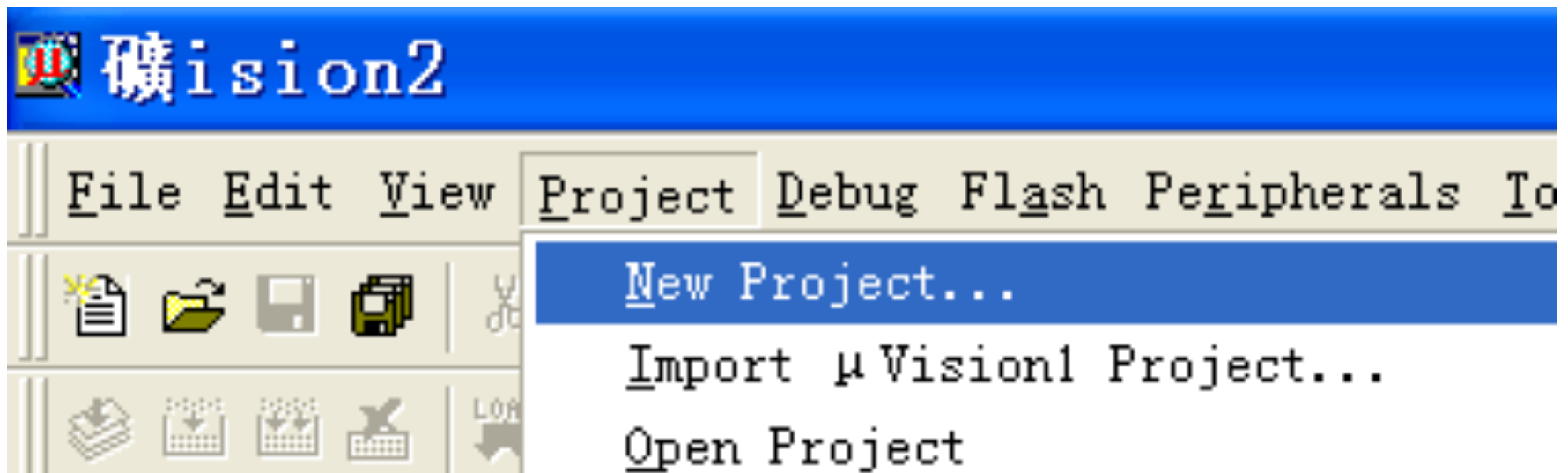
修改源程序中的错误；

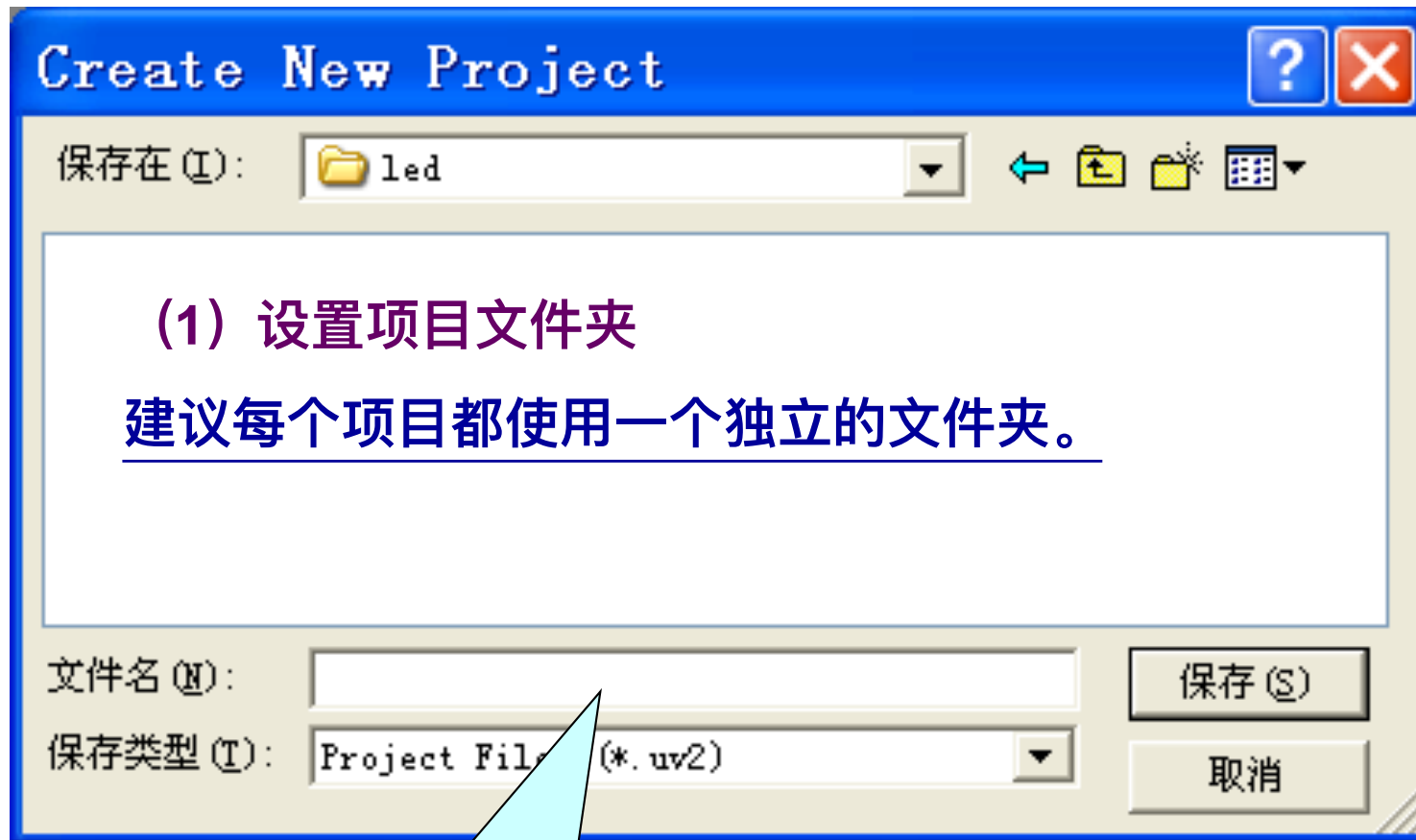
生成HEX文件。



(1) 创建一个项目

- 单击project/New Project, 出现创建新项目对话框





(2) 确定项目名称
使用英文名称

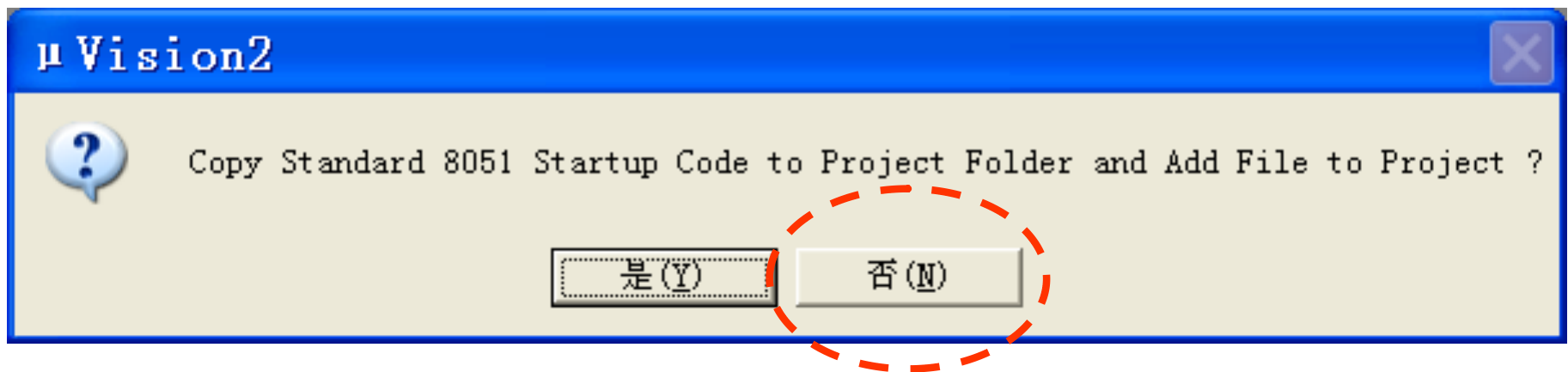


► 选择目标CPU

在输入项目名称后单击确定后，会弹出器件选择窗口；在此窗口中选择合适的CPU。窗口中的器件按公司名称排列。



器件选定后会出现如下的对话框，选择“否”。

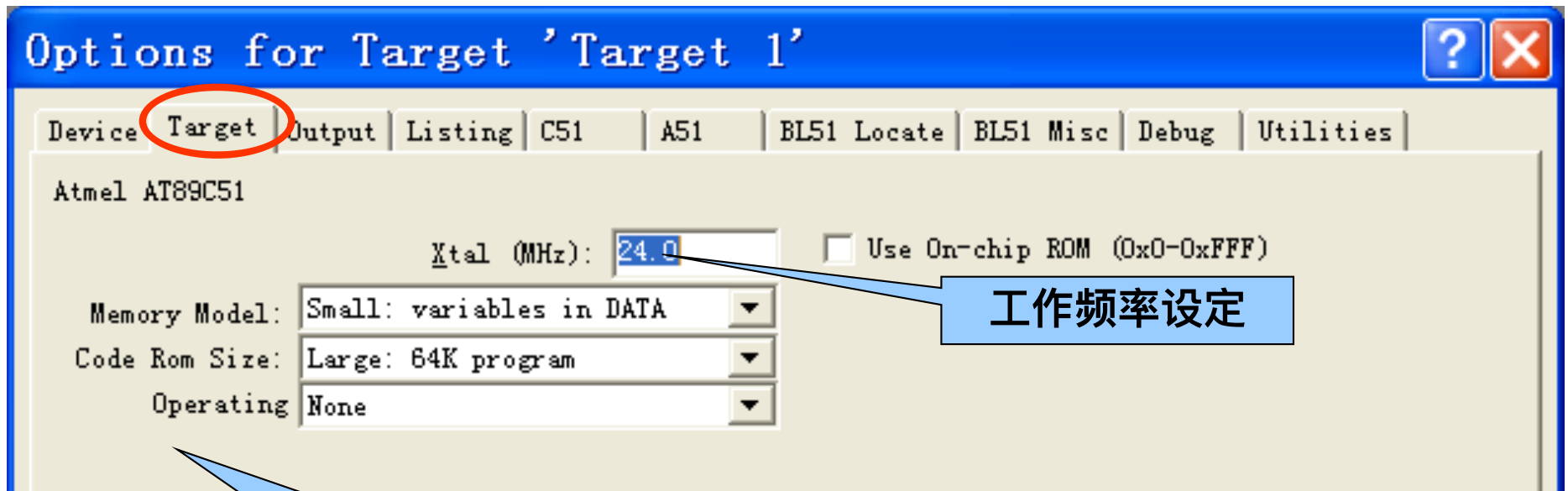


至此，一个新的工程已经建立。



(2) 配置项目

► 单击project/Option for Target 'Target1', 出现项目配置对话框



存储模式设定
程序存储器大小设定
操作系统选择



Options for Target 'Target 1'

Device | Target | **Output** | Listing | C51 | A51 | BL51 Locate | BL51 Misc | Debug | Utilities

Select Folder for Objects...

Name of Executable: led

输出文件目录设定
默认在当前项目工作目录

☒ Create Executable: .\led

☒ Debug Informatio

☒ Browse Informati

☐ Merge32K Hexfile

☒ Create HEX Fi

HEX HEX-80

输出文件选项

☐ Create Library: .\led.LIB

☐ Create Batch File

After Make

☒ Beep When Complete

☐ Start Debugging

☐ Run User Program #1

Browse...

☐ Run User Program #2

Browse...

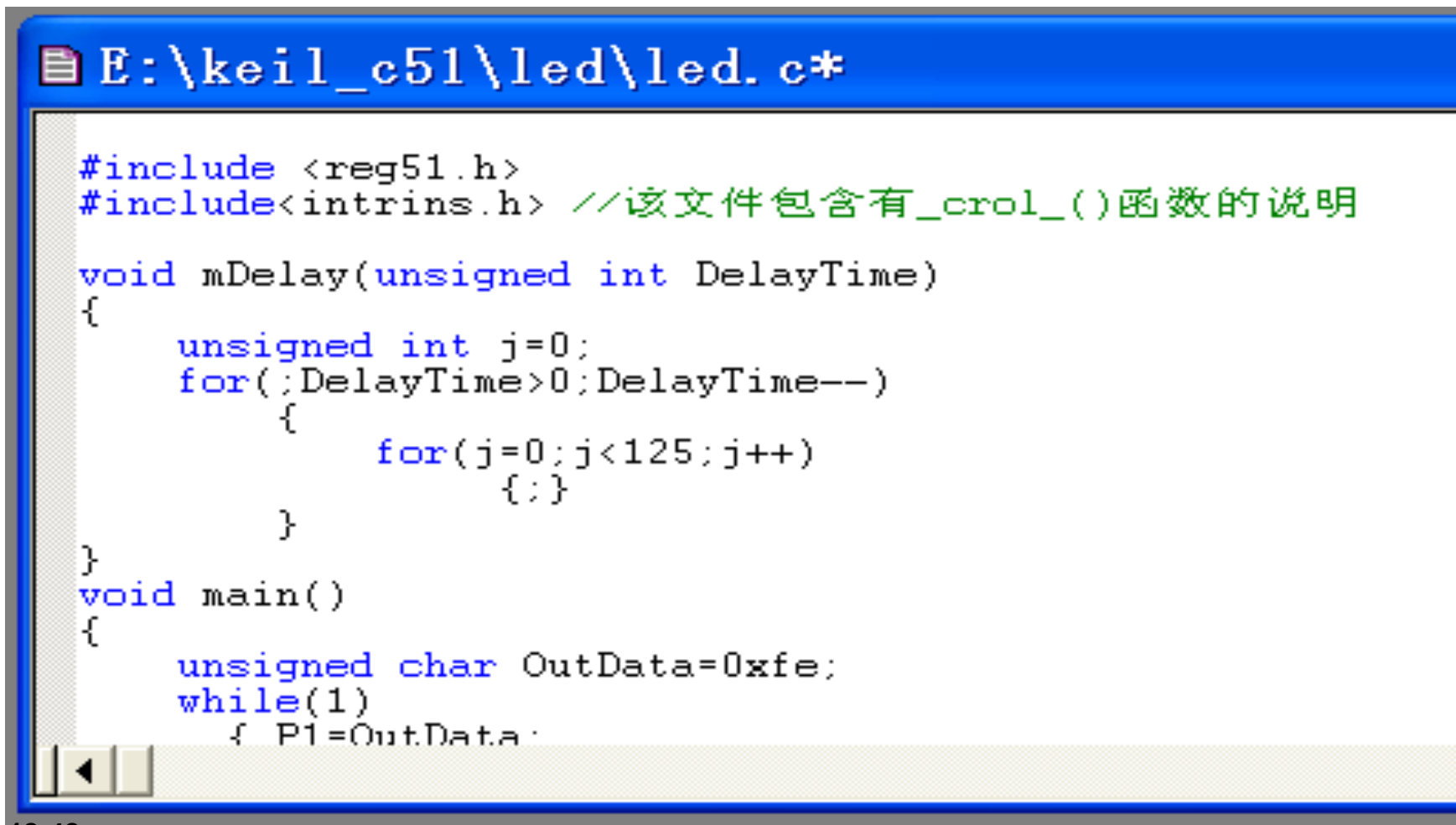
确定

取消

Defaults

(3) 创建源程序

► 单击Filet/New, 出现源代码编辑器。输入源代码。



```
E:\keil_c51\led\led.c*

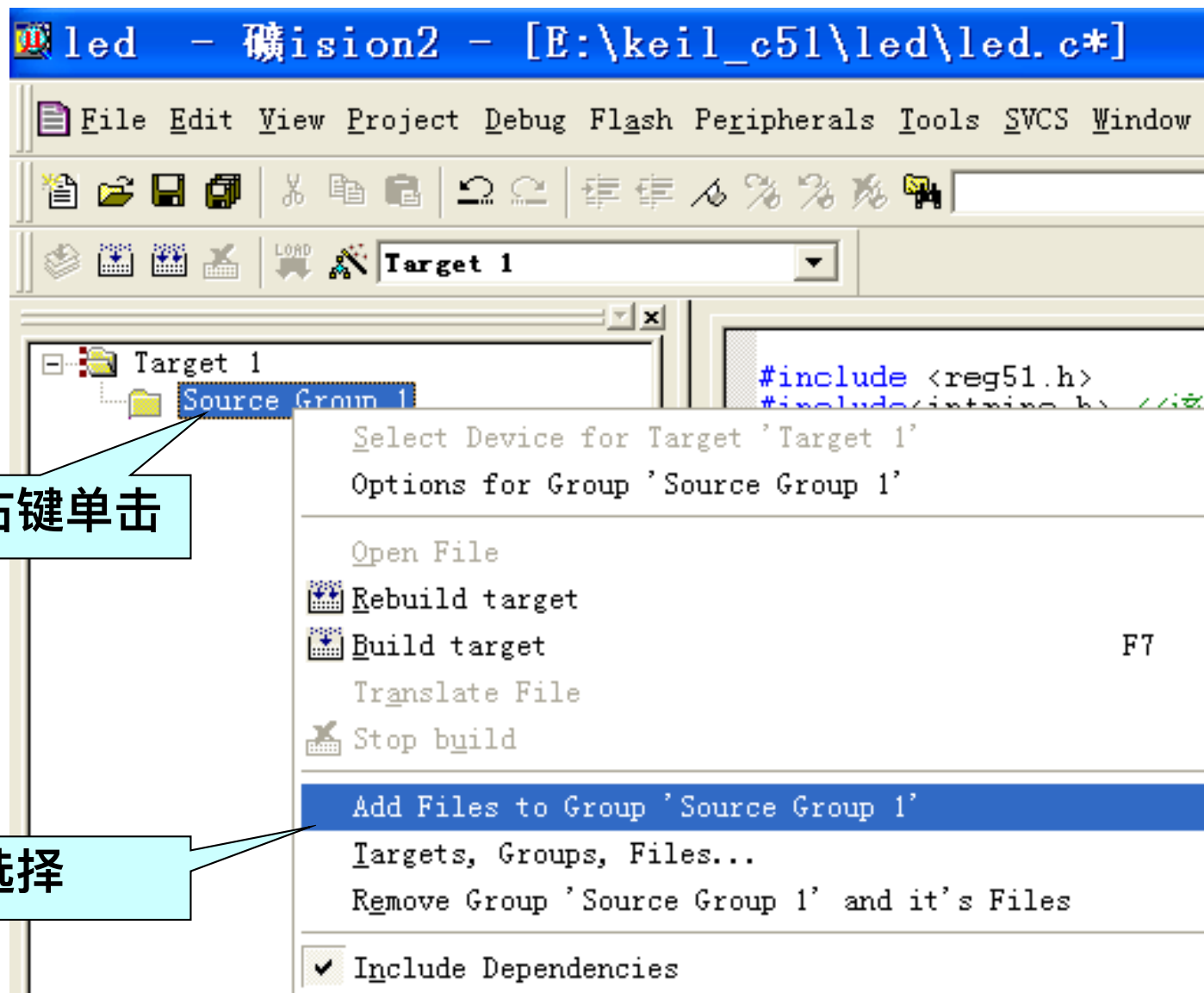
#include <reg51.h>
#include<intrins.h> //该文件包含有_crol_()函数的说明

void mDelay(unsigned int DelayTime)
{
    unsigned int j=0;
    for(;DelayTime>0;DelayTime--)
    {
        for(j=0;j<125;j++)
        {;}
    }
}

void main()
{
    unsigned char OutData=0xfe;
    while(1)
    { P1=OutData;
```



► 将文件添加到项目中。

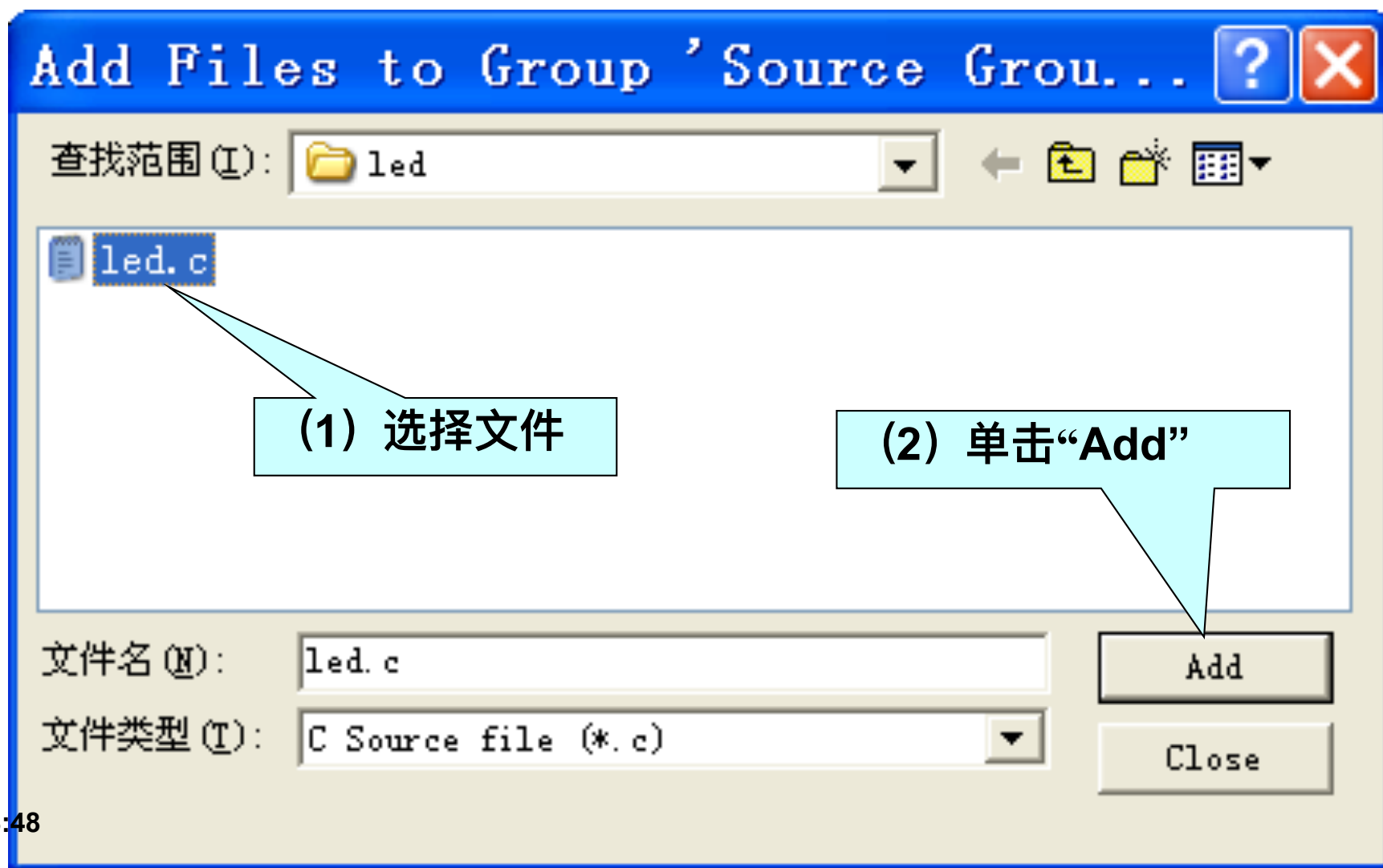


(1) 右键单击

(2) 选择



在弹出窗口选择需要添加到工程中的文件，然后单击“Add”。

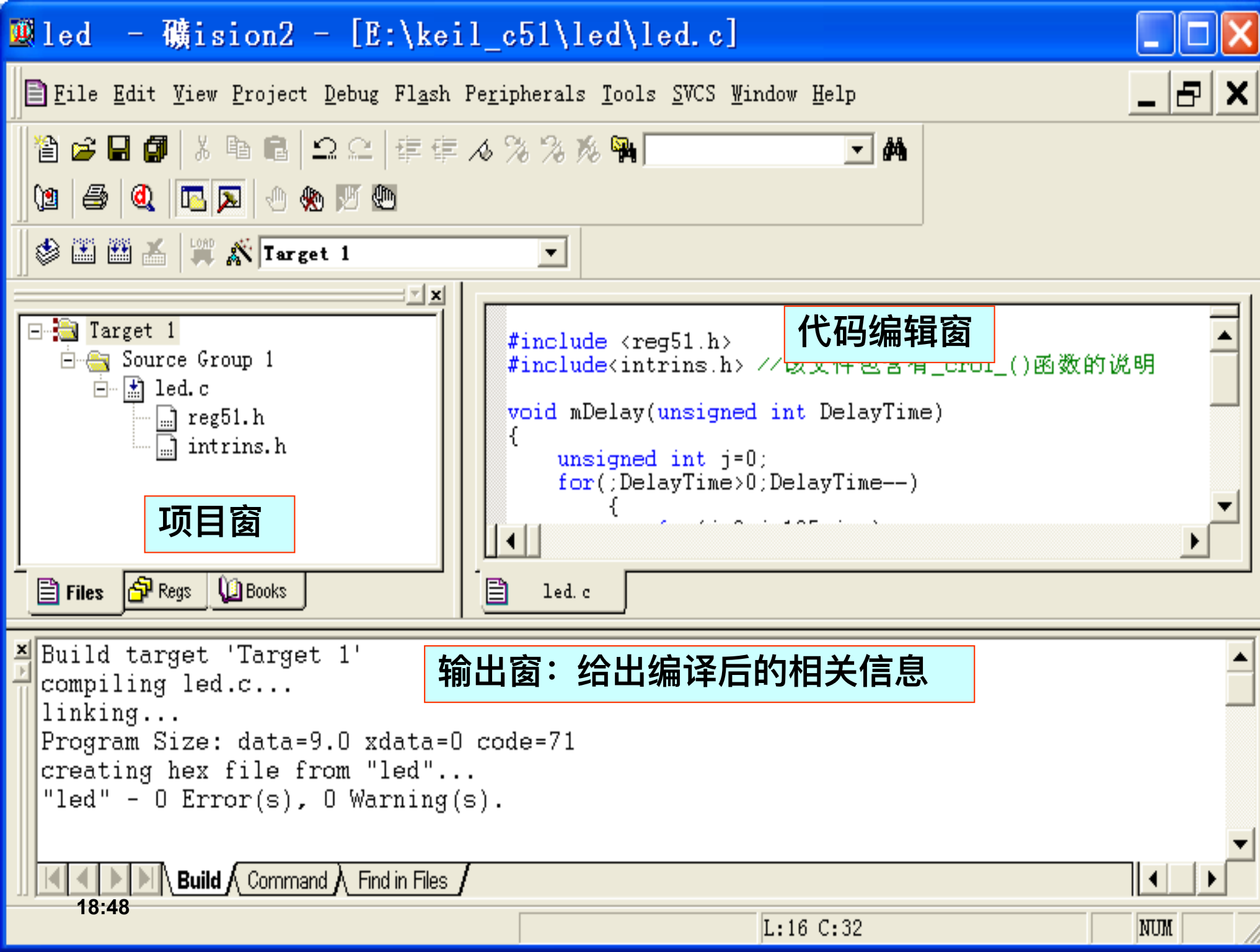


(4) 编译项目创建HEX文件



单击“Build Target”按钮，即可对所建项目进行编译，并生成HEX文件





二、项目的调试 — 调试模式 (debug mode)

(1) 调试器的两种工作模式

Use Simulator:

将uVision2调试器配置成纯软件产品。在此模式下，不需要实际的目标硬件就可以模拟51系列单片机的很多功能。

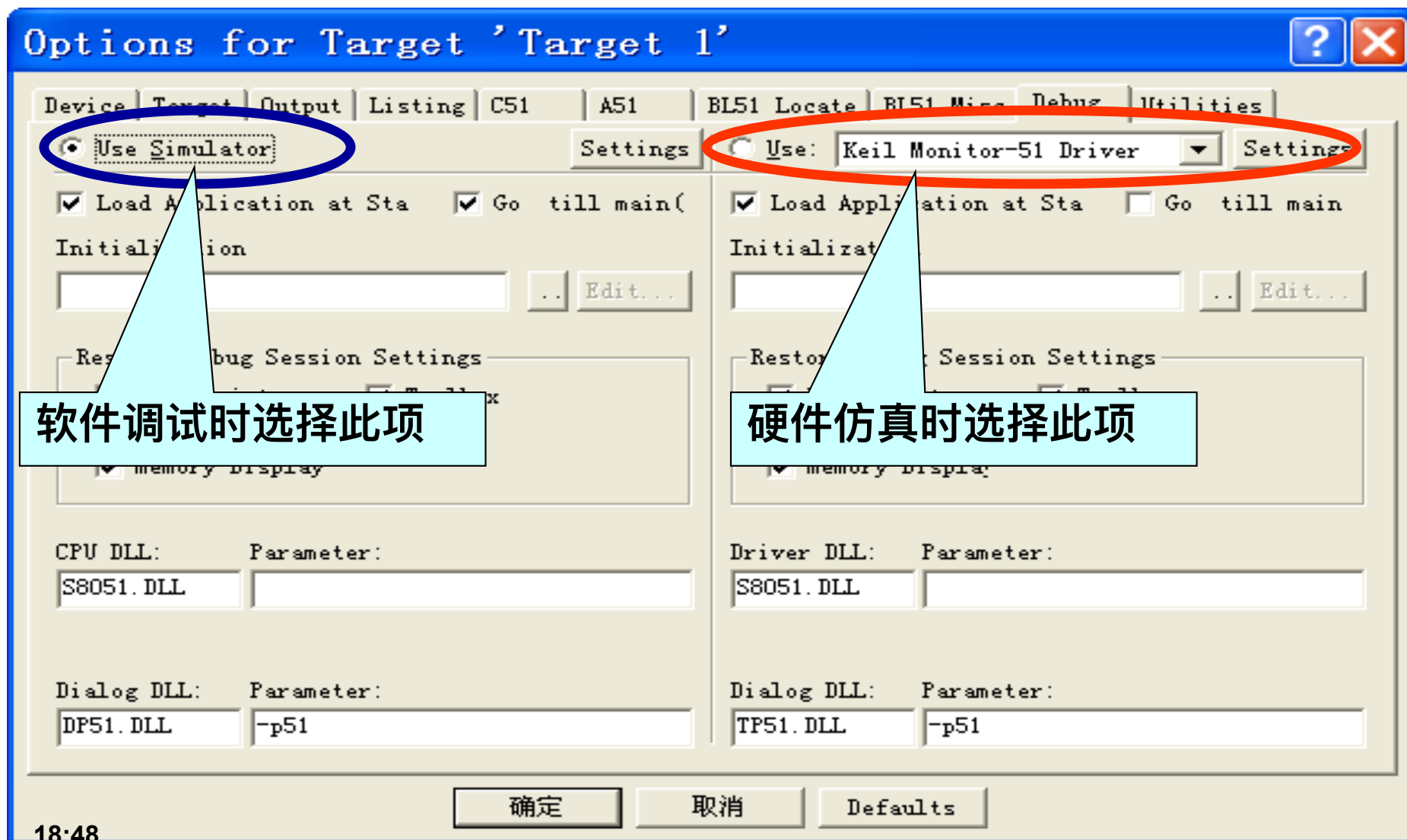
Use :

将uVision2调试器连接到仿真器。在此模式下，可以通过仿真器对目标硬件进行仿真测试。



工作模式的设定：

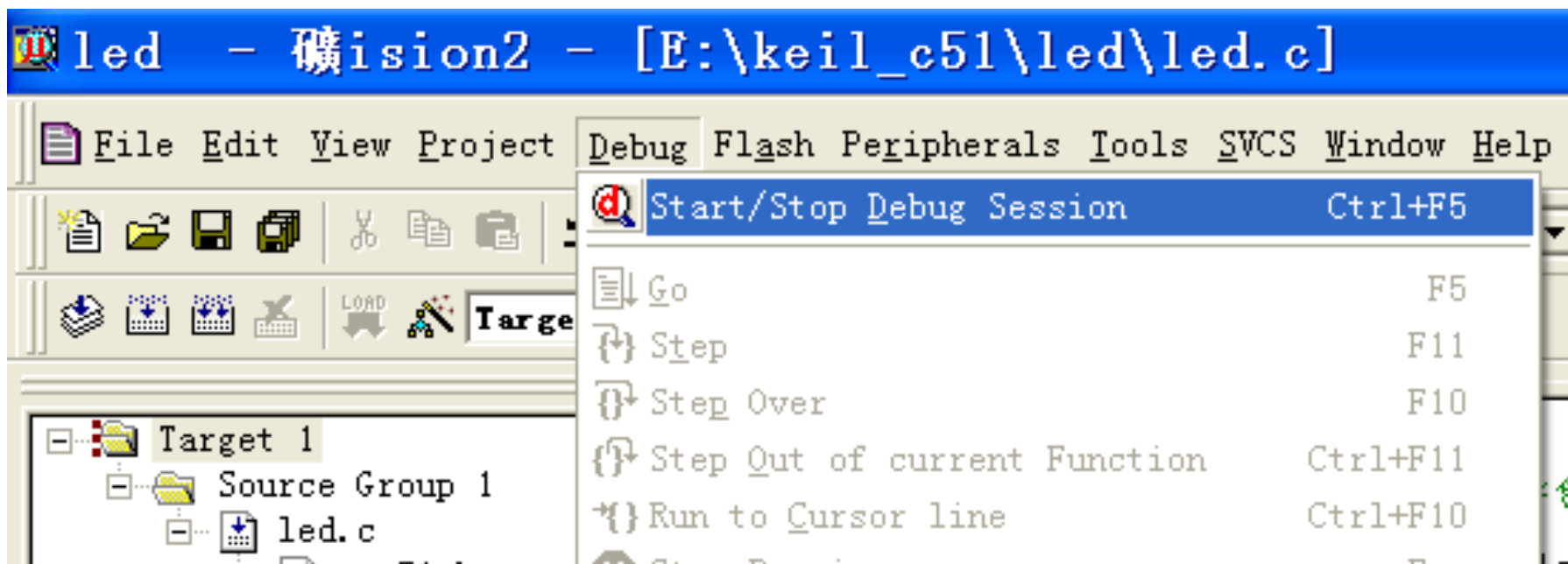
单击project/Option for Target 'Target1', 选择Debug页。

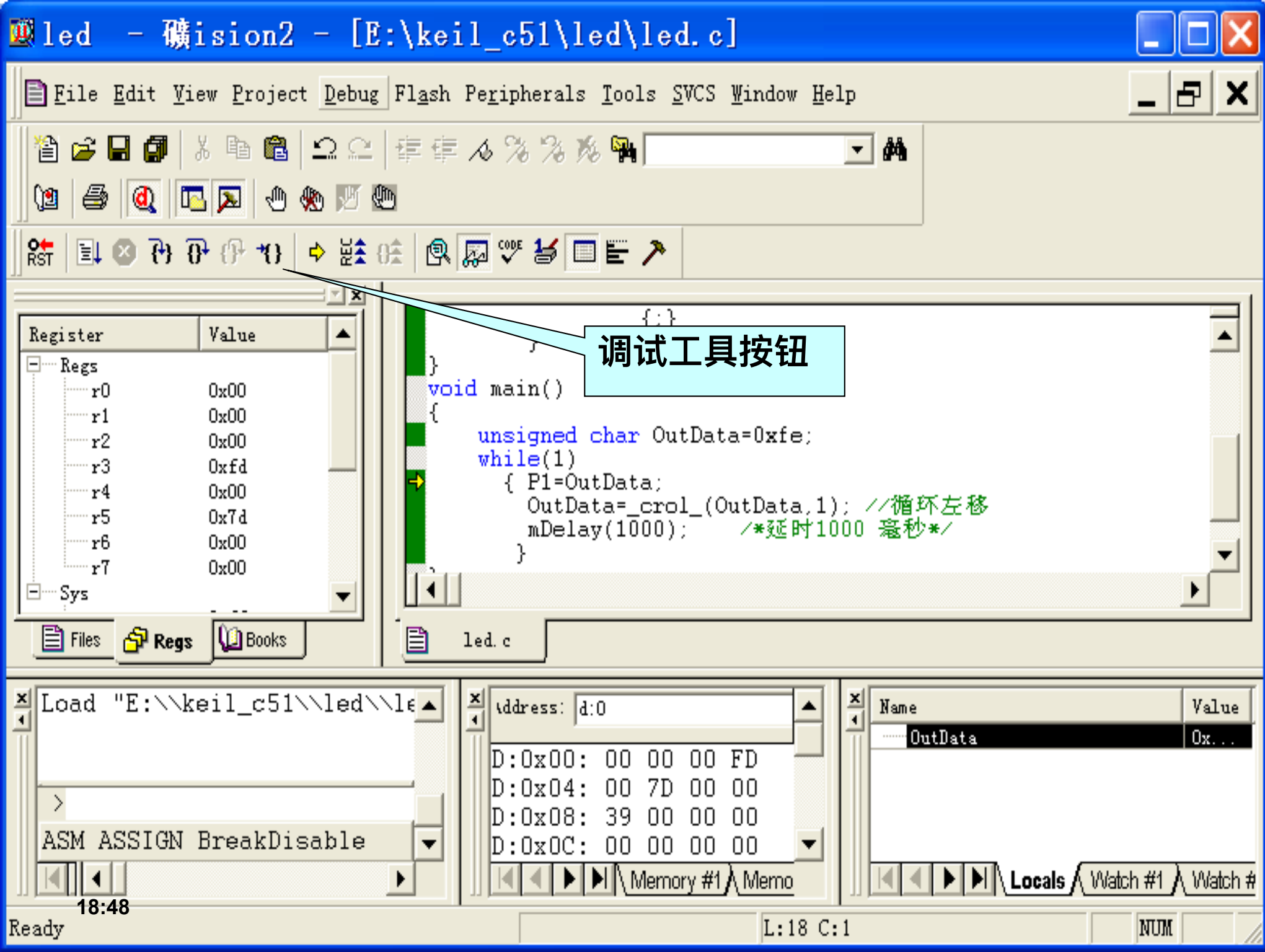


(2) 项目的调试方式和方法

► 启动调试。

选择Debug/Sart/Stop Debug Session,可以启动uVision2调试模式。
按照Option for Target – Debug的设定, uVision2会载入程序, 进入调试状态。





► 调试中目标程序执行方式。

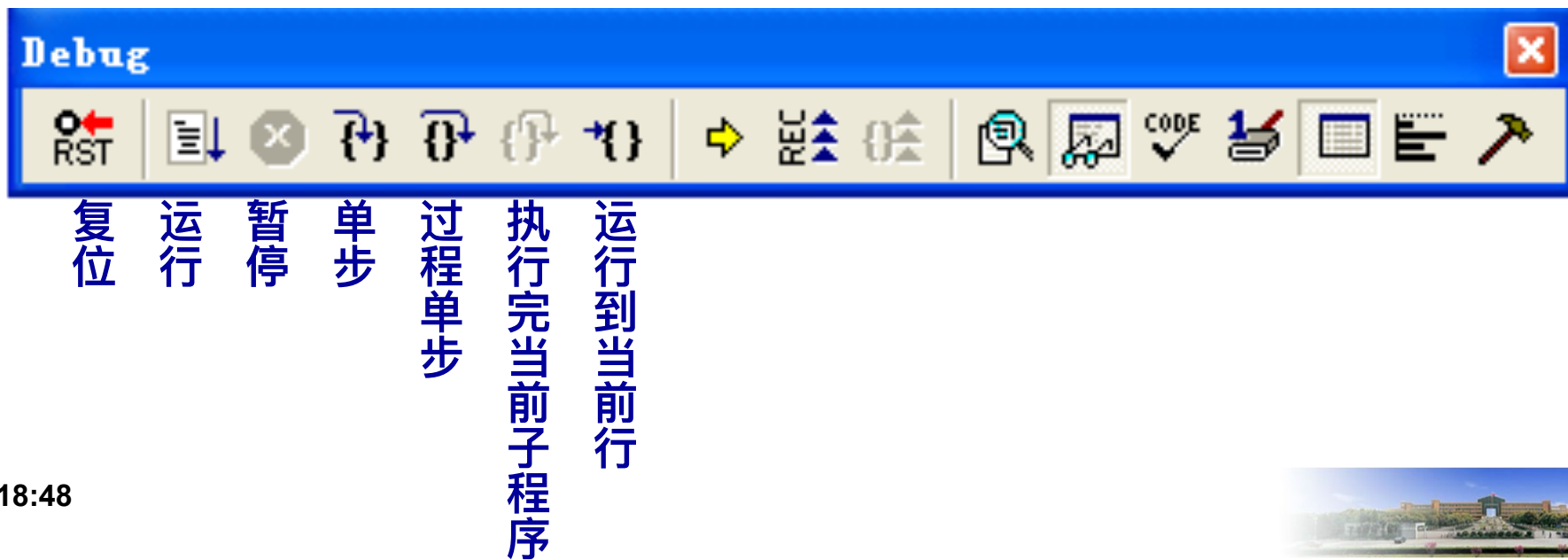
全速运行 (RUN)

单步(STEP INTO)

过程单步(STEP OVER)

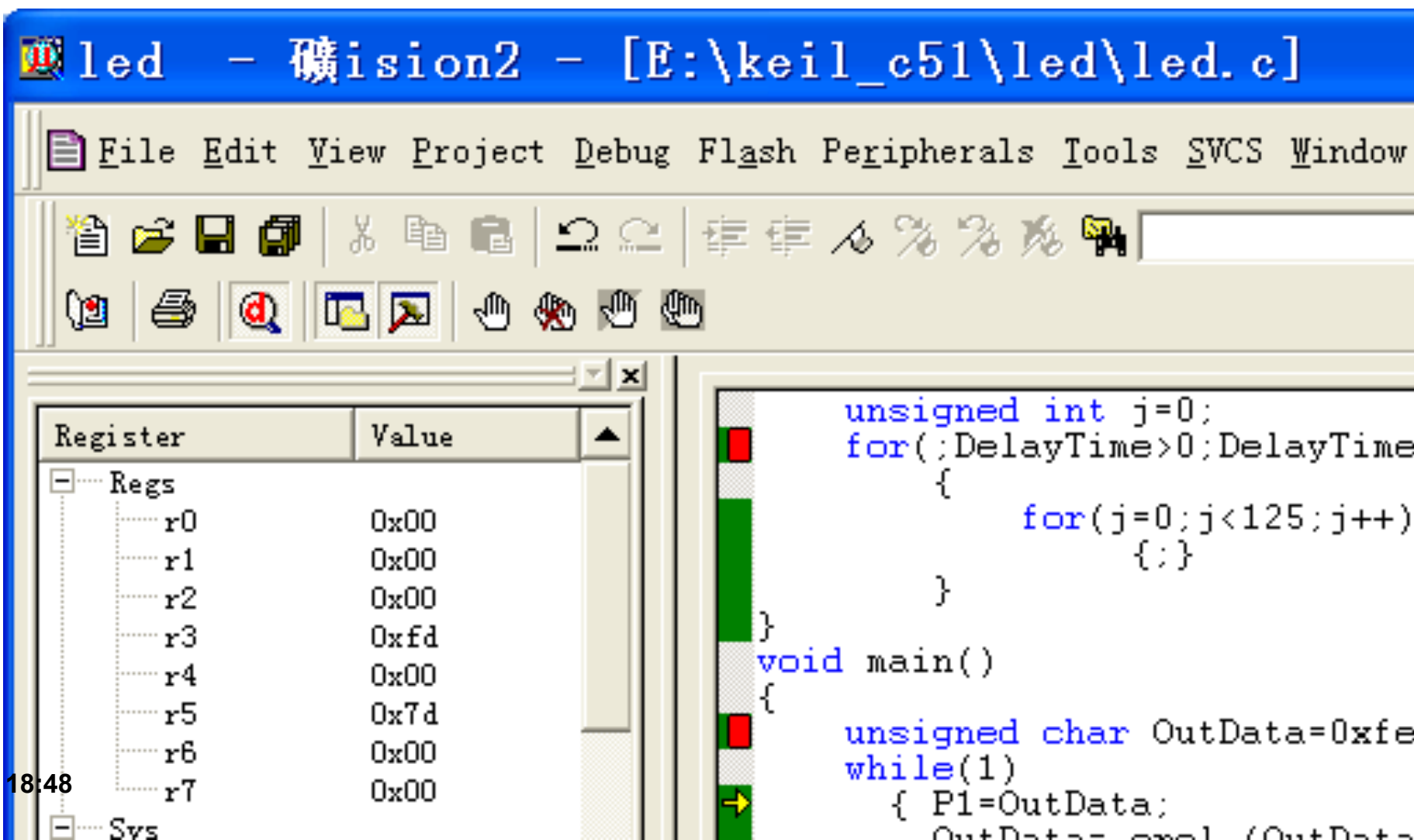
执行完当前子程序(STEP OUT)

运行到当前行(STEP TO CURSOR LINE)



► 调试中断点的设置。

断点：指在调试代码中设置的一个标志，当程序全速运行到该位置时，自动停止运行，此时调试者可以观察程序运行的结果。



断点控制按钮从左至右分别为：

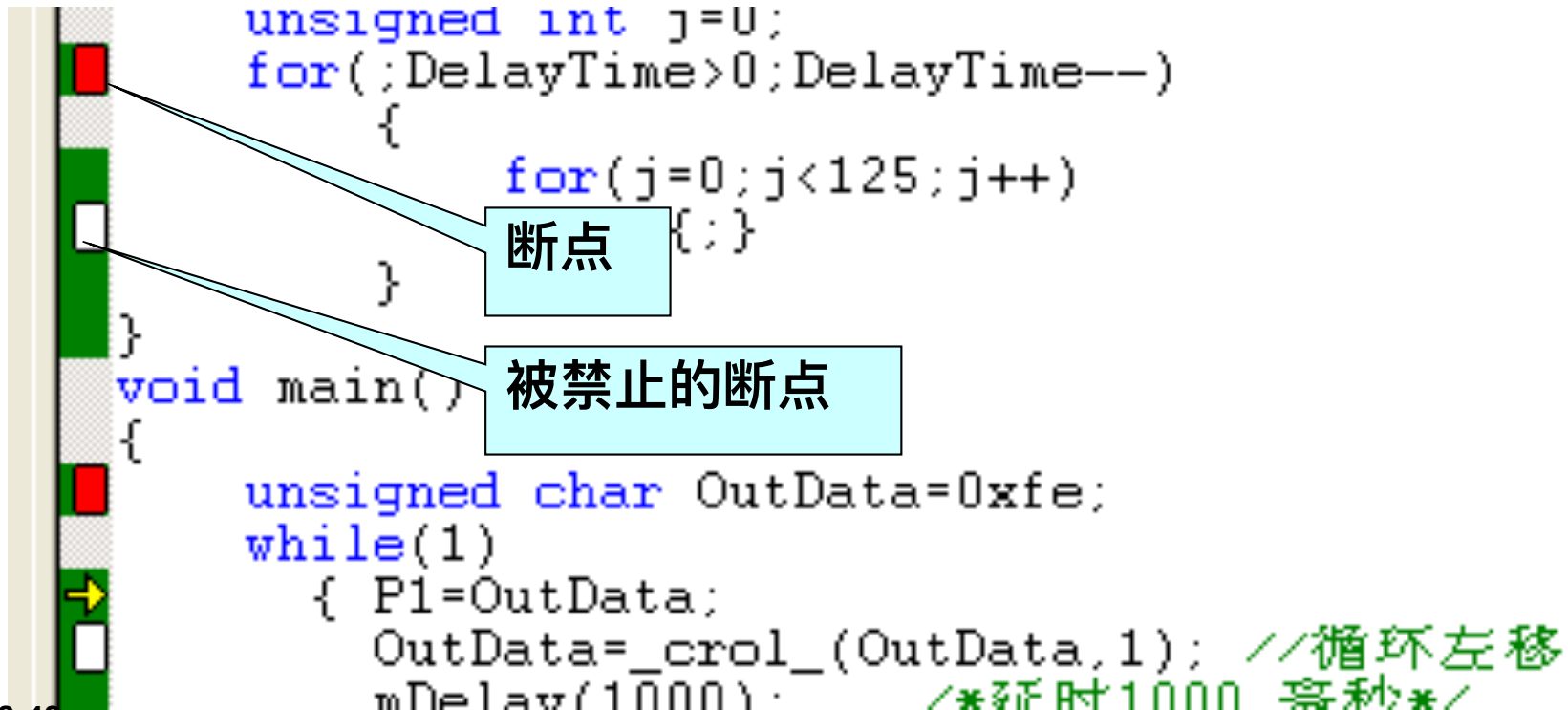


插入/除去断点 (Insert/Remove Breakpoint)

删除所有断点 (Kill all Breakpoint)

使能/禁止断点 (Enable/Disable Breakpoint)

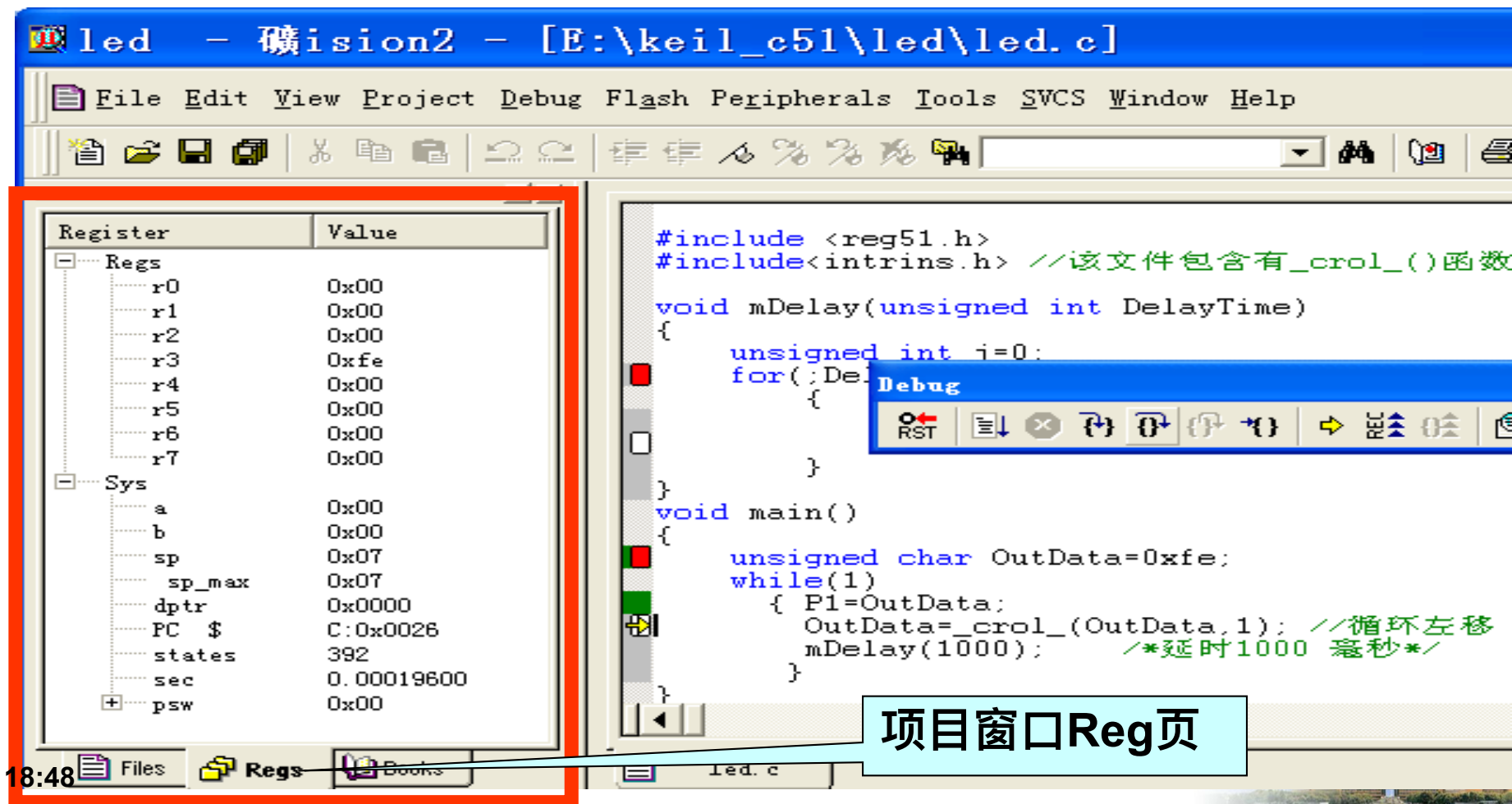
禁止所有断点 (Disable All Breakpoint)



► 调试中程序运行结果的观察。

存储器内容的观察：

(i) 工作寄存器及CPU寄存器内容的观察



The screenshot shows the Keil uVision2 IDE interface. The title bar indicates the project is 'led' and the file being edited is 'led.c' located at 'E:\keil_c51\led\led.c'. The menu bar includes File, Edit, View, Project, Debug, Flash, Peripherals, Tools, SVCS, Window, and Help. The toolbar contains various icons for file operations and debugging.

On the left, the 'Register' window is open, displaying a table of registers and their values. The 'Regs' section shows general-purpose registers r0 through r7, with r3 containing 0xfe. The 'Sys' section shows system registers like SP, SP_MAX, DPTR, PC, and SFRs.

On the right, the source code for 'led.c' is visible. It includes headers for 'reg51.h' and 'intrins.h'. A function 'mDelay' is defined, which uses a loop to delay for a specified time. The 'main' function initializes 'OutData' to 0xfe and enters a while loop that calls 'mDelay' and shifts 'OutData' left.

A 'Debug' toolbar is overlaid on the code, showing various debugging controls like 'RST', 'Step Into', 'Step Over', 'Breakpoint', etc.

At the bottom, a light blue box contains the text '项目窗口Reg页' (Project Window Register Page).

Register	Value
Regs	
r0	0x00
r1	0x00
r2	0x00
r3	0xfe
r4	0x00
r5	0x00
r6	0x00
r7	0x00
Sys	
a	0x00
b	0x00
sp	0x07
sp_max	0x07
dptr	0x0000
PC \$	C:0x0026
states	392
sec	0.00019600
psw	0x00

```

#include <reg51.h>
#include<intrins.h> //该文件包含有_crol_()函数

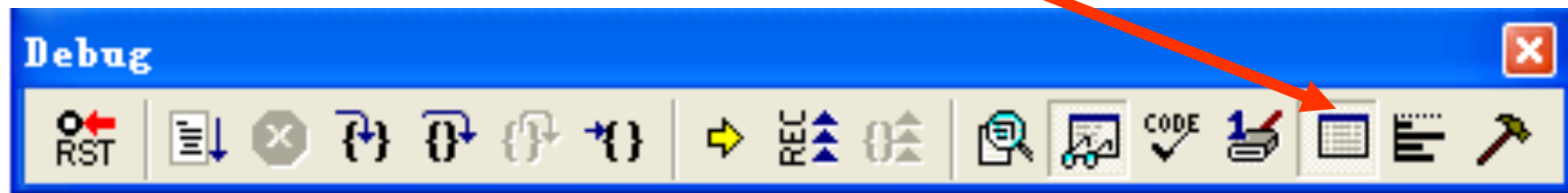
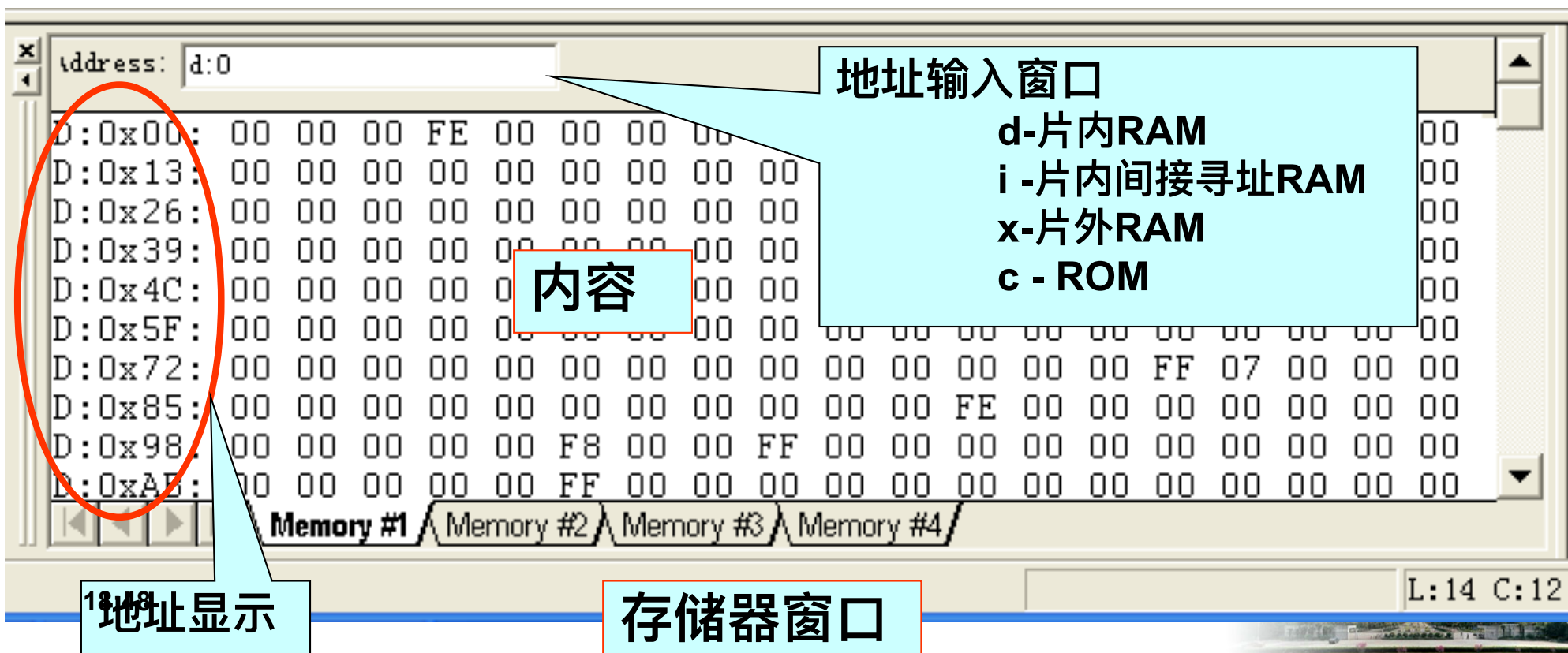
void mDelay(unsigned int DelayTime)
{
    unsigned int i=0;
    for(; i<DelayTime; i++)
    {
        ;
    }
}

void main()
{
    unsigned char OutData=0xfe;
    while(1)
    {
        P1=OutData;
        OutData=_crol_(OutData,1); //循环左移
        mDelay(1000); //延时1000 毫秒
    }
}
    
```

存储器内容的观察：

(ii) RAM及ROM内容的观察

选择**View/Memory Window**或单击**Memory Window**按钮会出现存储器窗口。

地址输入窗口
d-片内RAM
i-片内间接寻址RAM
x-片外RAM
c - ROM

内容

地址显示

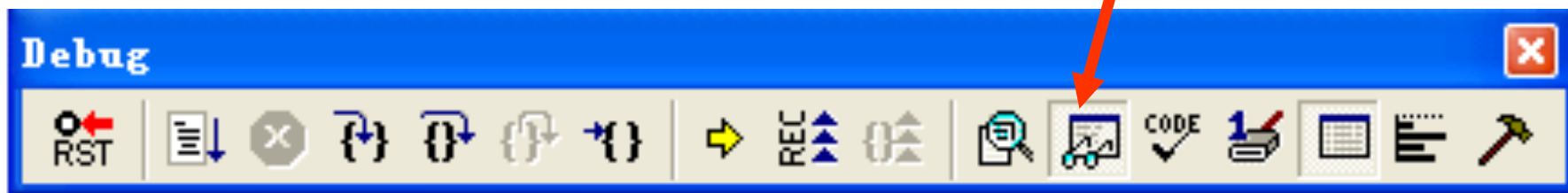
存储器窗口

Address	Memory #1	Memory #2	Memory #3	Memory #4
D:0x00:	00	00	00	FE
D:0x13:	00	00	00	00
D:0x26:	00	00	00	00
D:0x39:	00	00	00	00
D:0x4C:	00	00	00	00
D:0x5F:	00	00	00	00
D:0x72:	00	00	00	00
D:0x85:	00	00	00	00
D:0x98:	00	00	00	00
D:0xAB:	00	00	00	00

L:14 C:12

变量内容的观察：

选择 **View/Watch&Call Stack Window** 或单击 **Watch Window** 按钮会出现变量窗口。



观察窗口

Name	Value
DelayTime	0x03E8
j	0x0001

用于实时查看函数内部被定义的局部变量

用于实时查看全局变量

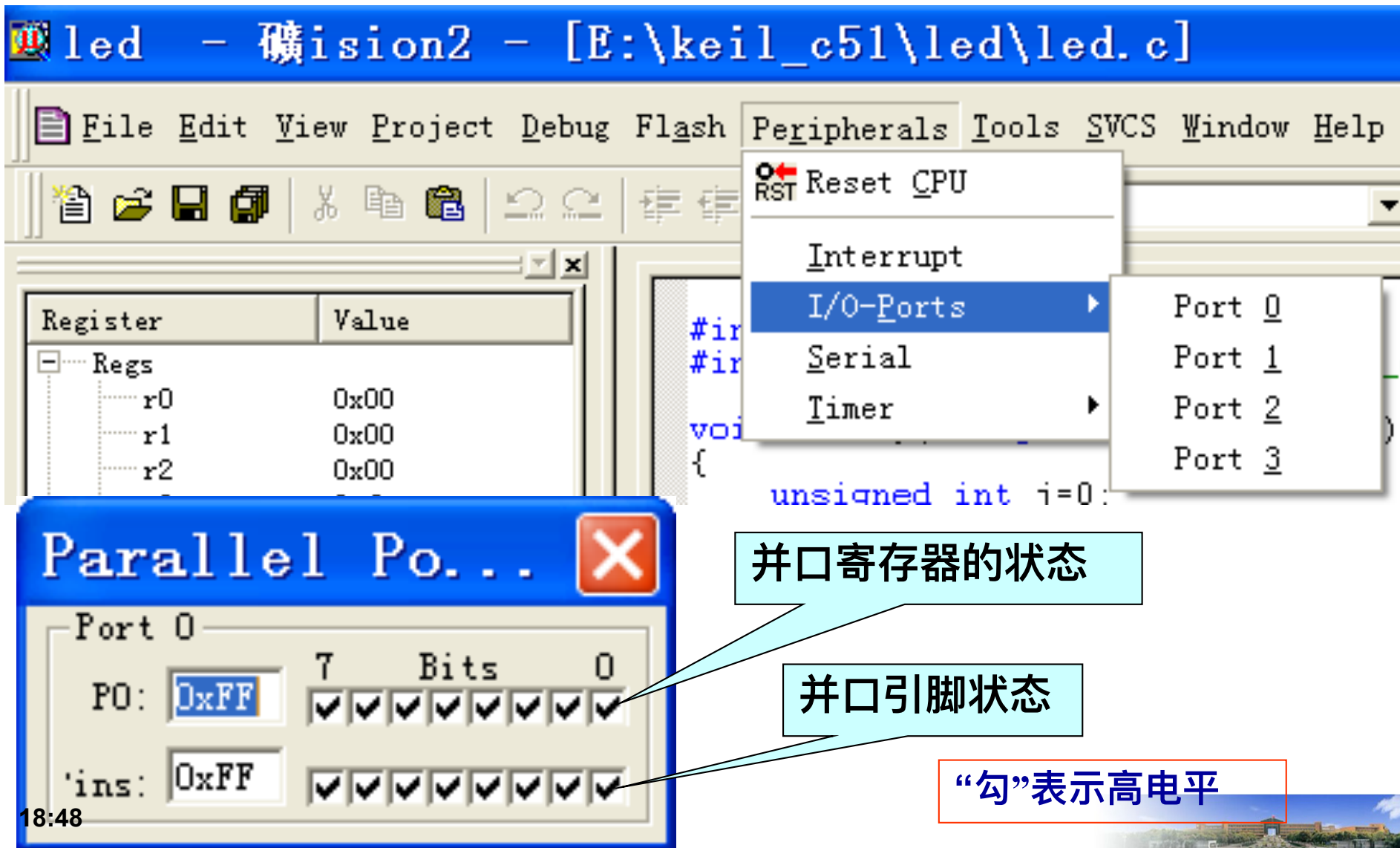
用于实时查看堆栈区情况

18:48

Locals / Watch #1 / Watch #2 / Call Stack

并口及片内资源的观察:

(i) 并口的观察



The screenshot shows the Keil IDE interface. The top menu bar includes File, Edit, View, Project, Debug, Flash, Peripherals, Tools, SVCS, Window, and Help. The Peripherals menu is open, showing options: RST, Reset CPU, Interrupt, I/O-Ports, Serial, and Timer. The I/O-Ports submenu is also open, showing Port 0, Port 1, Port 2, and Port 3. The Parallel Port window (Port 0) is open, showing the PO register value as 0xFF and the pins as 0xFF. The pins are represented by checkboxes, all of which are checked, indicating high level.

Register Value

Register	Value
Regs	
r0	0x00
r1	0x00
r2	0x00

File Edit View Project Debug Flash Peripherals Tools SVCS Window Help

RST Reset CPU

Interrupt

I/O-Ports

Serial

Timer

Port 0

Port 1

Port 2

Port 3

Parallel Po...

Port 0

PO: 0xFF

'ins: 0xFF

7 Bits 0

并口寄存器的状态

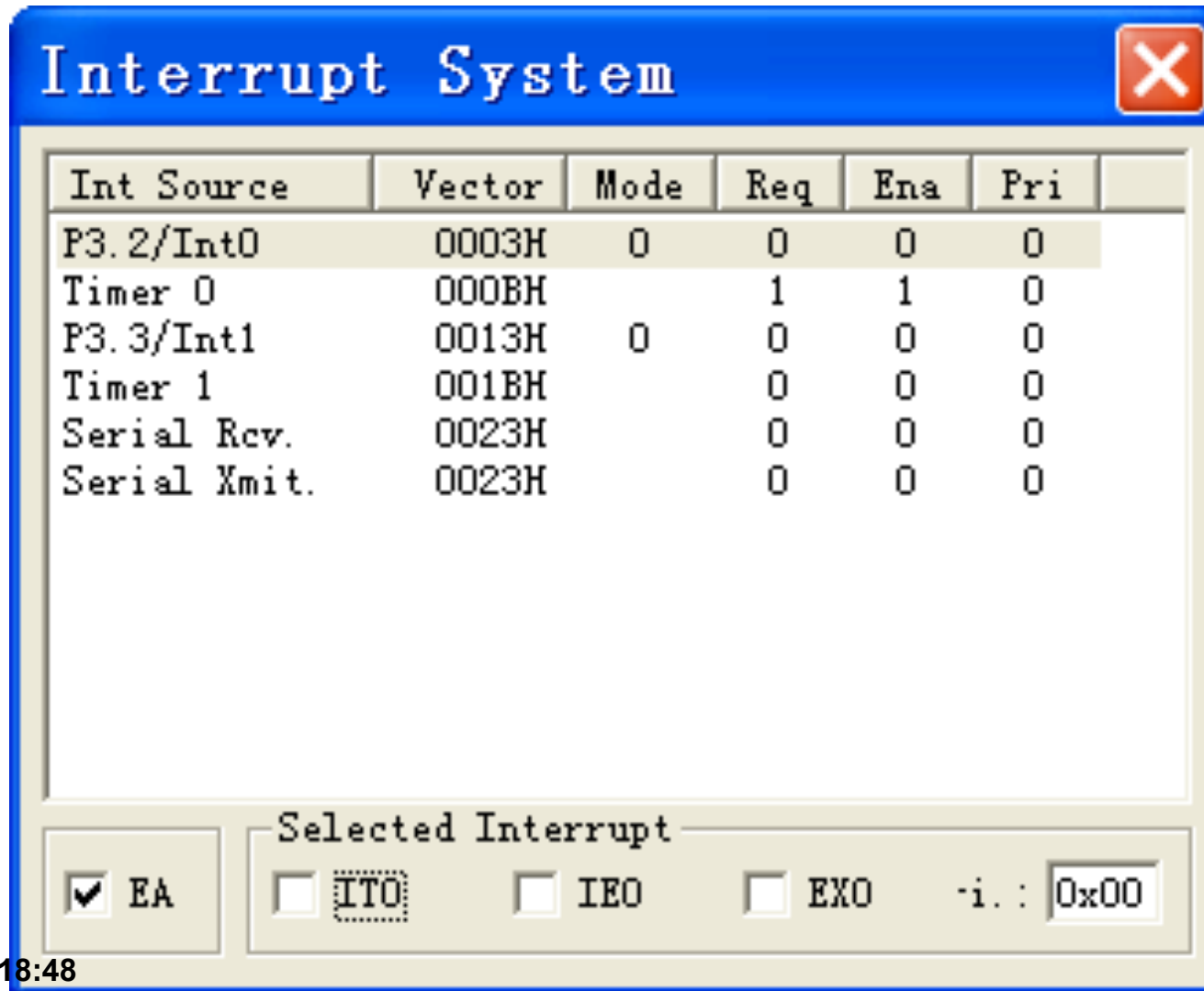
并口引脚状态

“勾”表示高电平

18:48

并口及片内资源的观察:

(ii) 中断的观察



The screenshot shows a software window titled "Interrupt System" with a table of interrupt sources and their status. The table has columns for Int Source, Vector, Mode, Req, Ena, and Pri. The rows are: P3.2/Int0, Timer 0, P3.3/Int1, Timer 1, Serial Rcv., and Serial Xmit. Below the table, there are checkboxes for EA, ITO, IEO, and EXO, and a field for -i. with the value 0x00.

Int Source	Vector	Mode	Req	Ena	Pri
P3.2/Int0	0003H	0	0	0	0
Timer 0	000BH		1	1	0
P3.3/Int1	0013H	0	0	0	0
Timer 1	001BH		0	0	0
Serial Rcv.	0023H		0	0	0
Serial Xmit.	0023H		0	0	0

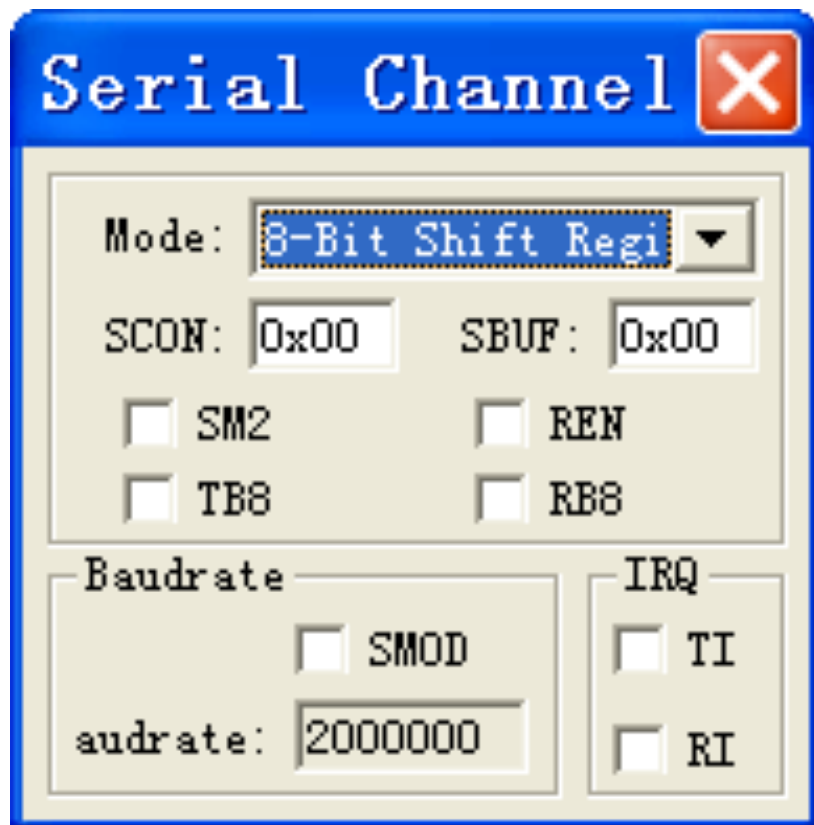
Selected Interrupt

☒ EA ☐ ITO ☐ IEO ☐ EXO -i.: 0x00



并口及片内资源的观察:

(iii) 中断及定时器的观察



Serial Channel

Mode: 8-Bit Shift Regi

SCON: 0x00 SBUF: 0x00

☐ SM2 ☐ REN

☐ TB8 ☐ RB8

Baudrate

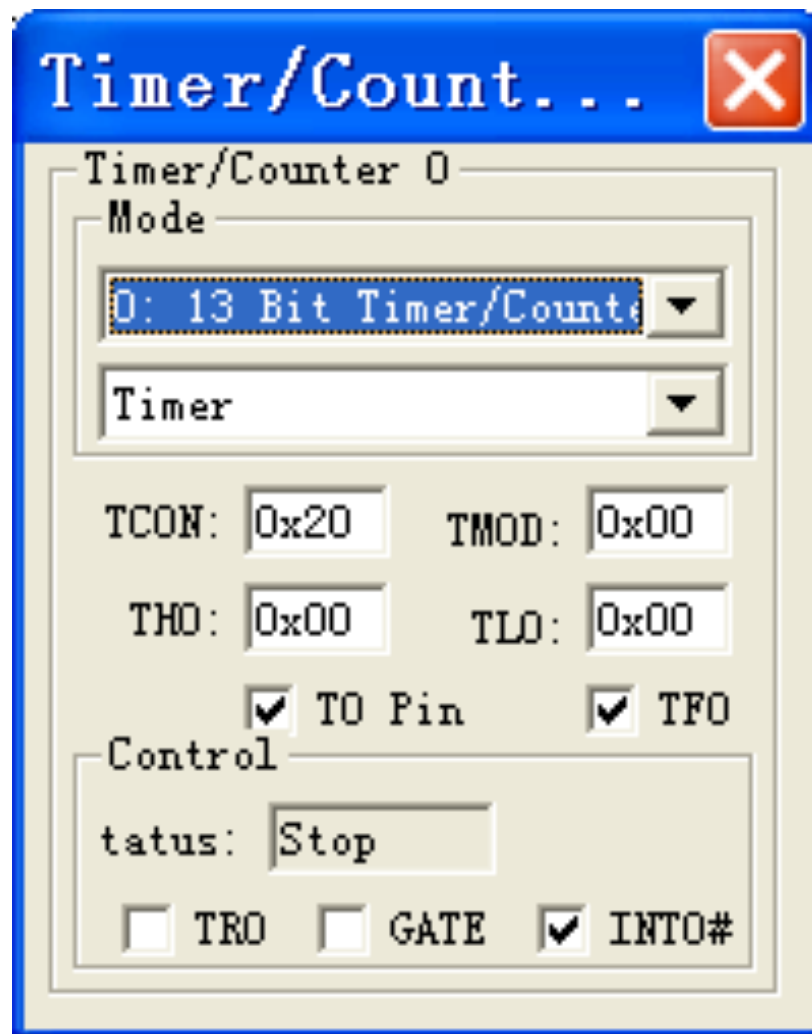
☐ SMOD

audrate: 2000000

IRQ

☐ TI

☐ RI



Timer/Counter...

Timer/Counter 0

Mode

0: 13 Bit Timer/Counter

Timer

TCON: 0x20 TMOD: 0x00

TH0: 0x00 TL0: 0x00

☒ TO Pin ☒ TFO

Control

tatus: Stop

☐ TRO ☐ GATE ☒ INTO#