



第一章 51mini 仿真器产品简介



技术背景:

2003 年 SST 公司推出了 SST89C54/58 芯片, 并且在官方网站公布了单片机仿真程序, 配合 KEIL 可以实现标准 51 内核芯片的单步调试等等, 从而实现了一个简单的 51 单片机仿真方案, 将仿真器直接拉低到一颗芯片的价格。

但是, 1 分钱 1 分货, 这个仿真方案由于先天的缺陷存在若干重大问题:

占用 p30, p31 端口

占用定时器 2

占用 8 个 sp 空间

运行速度慢 最高通信速度只有 38400, 无法运行 c 语言程序。(由于 c 语言程序会调用库文件, 每单步一次的时间足够你吃个早饭)

所以, 网上大量销售的这种这种仿真器最多只能仿真跑马灯等简单程序, 并没有实际使用价值。51mini 是深圳市学林电子有限公司开发生产的具有自主知识产权的新一代专业仿真器, 采用双 CPU 方案, 一颗负责和 KEIL 解释, 另外一颗负责运行用户程序, 同时巧妙利用 CPU 的 P4 口通信, 释放 51 的 P30, P31, 完美解决了上述问题, 体积更小, 是目前价格最低的专业级别 51 单片机仿真器, 足以胜任大型项目开发。

USB采用标准尺寸接口和定做电缆
轻松通过500MA电流, 保障供电充足



创新设计:

- 1 三明治夹心双面贴片, 体积缩小到只有芯片大小, 真正的“嵌入式”结构。
- 2 大量采用最新工艺和器件, 全贴片安装, 进口钽电容, 贴片电解。
- 3 采用快恢复保险, 即便短路也可有效保护。
- 4 单 USB 接口, 无需外接电源和串口, 台式电脑、无串口的笔记本均适用。三 CPU 设计, 采用仿真芯片+监控芯片+USB 芯片结构, 是一款真正独立的仿真器, 不需要依赖开发板运行。
- 5 下载仿真通讯急速 115200bps, 较以前版本提高一个数量级(10 倍以上), 单步运行如飞。
- 6 不占资源, 无限制真实仿真(32 个 I/O、串口、T2 可完全单步仿真), 真实仿真 32 条 I/O 脚, 包括任意使用 P30 和 P31 口。
- 7 兼容 keil C51 UV2 调试环境支持单步、断点、随时可查看寄存器、变量、I/O、内存内容。可仿真各种 51 指令兼容单片机, ATME1、Winbond、INTEL、SST、ST 等等。可仿真 ALE 禁止, 可仿真 PCA, 可仿真双 DPTR, 可仿真硬件 SPI。媲美 2000 元级别专业仿真器!



8 独创多声响和 led 指示实时系统状态和自检。

9 独创长按复位键自动进入脱机运行模式，这时仿真机就相当于目标板上烧好的一个芯片，可以更加真实的运行。这种情况下实际上就变了一个下载器，而且下次上电时仍然可以运行上次下载的程序。

硬件说明：

USB 接口：连接 USB 线到电脑，同时起到通信和供电的功能。

晶体选择：默认是 22.1184 晶体，如果您的用户板上有晶体，那么可以拨向 user 一端，这时系统将按照你自己目标板晶体的频率运行。但是请注意，如果拨向 user，而目标板晶体不起振，系统将无法正常运行



状态指示灯：此灯为红色 LED，闪一下表示系统复位，连续闪烁表示系统运行。

复位键：短按一下，绿灯闪一下，喇叭响一下，则系统复位。（重新进入仿真之前请按一下复位键）

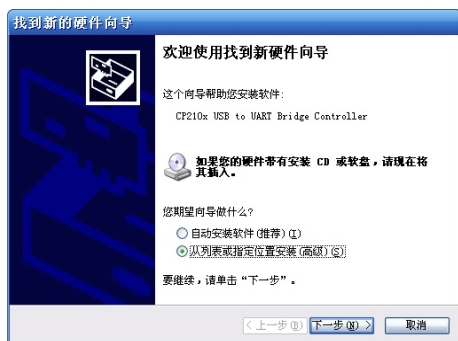
长按 3 秒，绿灯不停闪烁，喇叭响两下，则系统脱机运行。

喇叭：响三下“嘀嘀嘀”，则表示被仿真程序的 00H-0040H 被用户占用，请你参考后面说明更改程序。

第二章 usb驱动程序安装指南

51mini 采用 USB 通信方式请首先遵循如下步骤安装 USB 驱动。

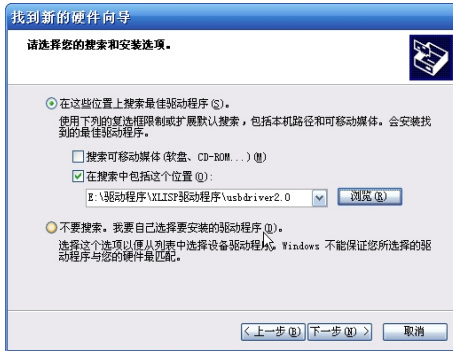
第一步：用随机 USB 电缆连接仿真器 USB 插座和计算机 USB 口；显示找到新硬件向导，选择“从列表或指定位置安装（高级）”选项，进入下一步；



第二步：选择“在搜索中包括这个位置”，点击“浏览”，定位到驱动光盘的文件夹，如 E:\驱动程序\XLIISP 驱

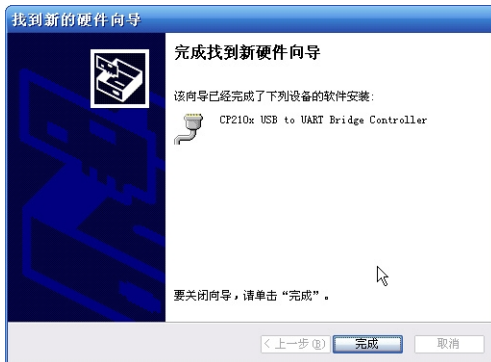


动程序\USBDRIVER2.0\, 点下一步;

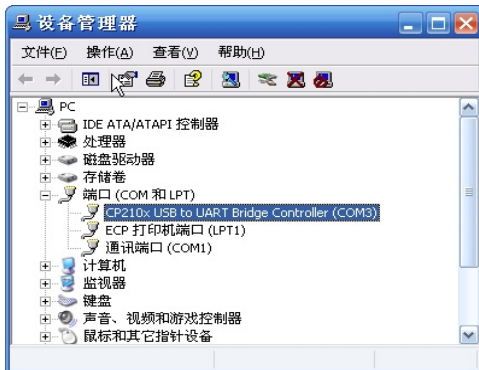


第三步: 弹出“硬件安装”对话框, 如果系统提示“没有通过 Windows 徽标测试...”, 不用理会, 点击“仍然继续”, 向导即开始安装软件;

然后弹出“完成找到新硬件向导”对话框, 点击完成。



第四步: 重复以上几个步骤; 右下角弹出对话框“新硬件已安装并可以使用了”, 你可以进入系统的: 控制面板\系统\硬件\设备管理器中看到以下端口信息, 表示系统已经正确的安装了 USB 驱动。



安装失败的解决方法:

如果在安装中途退出或掉电等, 可能造成安装失败。这时, 可以在设备管理器中删除带有黄色感叹号的 USB 设备, 然后在重复以上的安装步骤即可。

特别情况下的 usb 安装:

一般情况下的 usb 安装如上所述, 特别情况下, 您的电脑可能已经安装了其他的 usb 打印机, usb 编程器等。在



这种情况下，分配给 usb 虚拟串口可能是串口 5，甚至是串口 6! 而 KEIL 仿真的 COM 选择范围只限于 com1, com2, com3, com4 四个串口，如何解决呢？比较笨的办法是干脆重新安装系统，在新的系统中首先安装 XLISP 的 usb 驱动，这样系统分配的串口自然靠前的了。

图 1 当 usb 设备较多的时候，系统分配的虚拟口号为 com5，导致无法使用 KEIL 等软件。



图 2：指向 usb 虚拟出来的 com5，点右键，察看属性



图 3：进入 usb serial port(com5)属性后，点上面的“端口设置”然后“高级属性”在端口号中选择一个不用的 COM 端口号比如 COM2

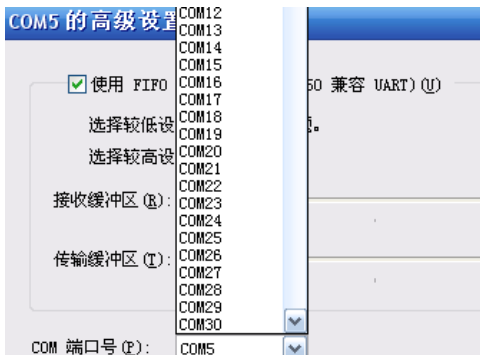


图 4：系统已经把 com2 分配给 usb 虚拟串口，以后您就可以使用此 com 口啦！



(仿真器连接示意图,仅仿真器连结 USB 线,同时兼顾通信和供电。开发板不需要连接任何供电线,同时仿真器特别有设计保护电路,有效防止短路)



第 3 章 51mini 仿真操作指南

51mini仿真器采用功能强大的keil 集成开发环境, 本光盘提供该软件的评估版, 如果您觉得这个软件不错可以联系代理商购买正式版本。

(由于KEIL 操作较复杂, 如果您是新手, 请严格参考以下说明书, 一步步的操作, 相信您一定能够成功!)

硬件安装: 首先连接好usb电缆, 此时仿真器上的喇叭响一下, 同时指示灯闪烁一下。表示系统自检正常。插入仿真头到目标板, 请注意仿真头插入方向!



软件安装: 直接拷贝光盘\开发工具\keil\到c盘的根目录, 去掉所有文件的只读属性。 (注意必须是c盘根目录, 不要安装在其他目录)



执行keil\uv2\uv2.exe即可(可将文件发送到桌面快捷方式)

KEIL软件是他的驱动软件。用户也可自行搜索下载更新版本的KEIL软件。



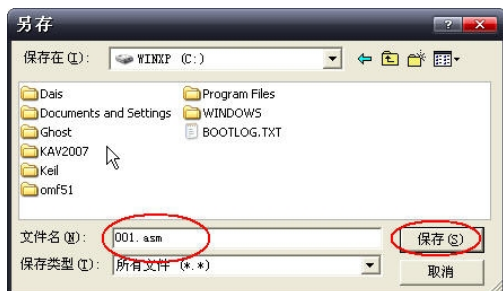
建立您的第一个项目(软件操作指南):

1 点击 Project(工程)菜单, 选择 New Project (新工程), 在文件名中输入您的第一个程序项目名称, 假定我们用"test"。"保存"后的文件扩展名为 uv2 这是 KEIL uVision2 项目文件扩展名, 以后我们可以直接点击此文件以打开先前做的项目。注意输入的时候不要输入文件的后缀名, 默认即可。为了查找方便, 假定我们保存在 c 盘的根目录。

这时会弹出让你选择单片机型号的对话框, 我们选择 ATMEL---AT89C52。注意我们实际使用的是 89s52, 他们的资源基本是一致的, 不必理会。



2 汇编源文件的建立: 点击 FILE (文件) 菜单中的 NEW... (新文件) 命令新建一个文本编辑窗口。在里面输入一个程序范例如流水灯程序, 然后点击 FILE (文件) 菜单中的 SAVE (保存) 命令保存文件, 注意必须输入文件的后缀名 .asm, 例如保存为 C: /prog/001.asm (注意不要有中文目录, 文件名不超过 8 个字符, 否则编译不通过! 如果您是第一次使用, 那么我建议您直接保存在 c 盘的根目录, 文件名同样取 001.ASM, 不要试图把他保存在桌面/我的文档等等!)



请注意:

C51 用户请在您的代码的 main() 函数前面, 加上一句:

```
char code dx516[3] _at_ 0x003b;
```

ASM 汇编用户请将你的主程序跳过中断区直到 0050H 以后, 如:

```
org 0000H
```



```
ljmp main
```

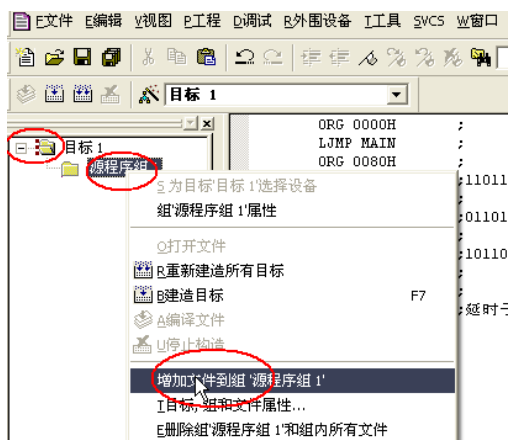
```
org 0050H
```

```
main: . . . ;程序开始
```

如果以上设置你没有做，在装载过程中，仿真器会发出“滴滴滴”的三声短声报警，这时的仿真结果将可能不正确。

现在随着单片机技术的发展，CPU的00-40H之间已经越来越多的被新的寄存器占用，而不是以前的00-30H，养成良好的程序书写习惯很重要。如果您的程序是30H。请你改为40H或者以后如50h，80h。如果您有所疑问请您仔细参考本光盘的测试程序举例中中的例子，都是修改过的。

3 点击 Target 1 (目标 1) 前面的“+”，出现 Source Group 1, (源程序组 1) 选中右键点选“Add files Group Source Group 1” (增加文件到源程序组 1) 这时选择文件类型为 Asm, 再选中 001.asm 文件, 再按 Add 添加, 在随后的提示框中按“确定”。



4 仿真部份采用Mon51协议，在使用之前应必须对软件项目进行如下设置：

单击“Project” (工程) 菜单, 再在下拉菜单中单击“options for target target 1” (目标 1 属性) 在下图中选择 output (输出) “Create HEX file” (产生 hex 文件) 的选项, 以便汇编后生成 HEX 代码, 供编程器使用。

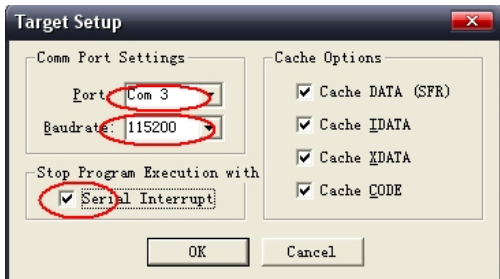


5 在“Debug” (调试) 中点选“Keil Monitor-51 Driver”, 同时选中加载代码到仿真器, 运行到main0, 在“恢复调试设置”中“断点”“工具栏”

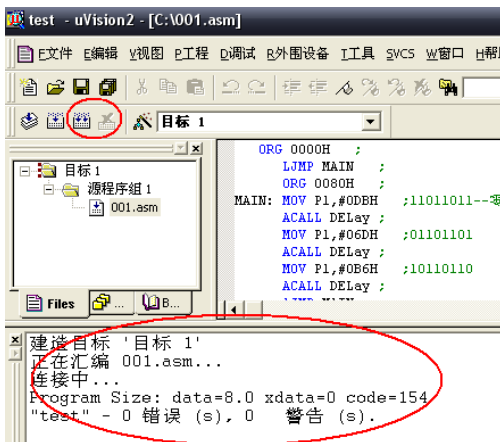
“浏览点” “储存器显示”也全部勾选上, 然后点击“设置”。



6 击"Settings"（设置）选择要使用串口必须和实际相符合，你的计算机可能是 COM2 或者其他，对此不能确定可以通过察看 控制面板\硬件\端口\通讯端口来解决，同时注意电脑上不能够同时运行其他可能占用串口资源的软件例如 XLISP 下载软件，串口调试软件等等也必须退出，否则将引起冲突，（如果您使用的是 USB 通信方式，那么选择系统分配的虚拟串口号）

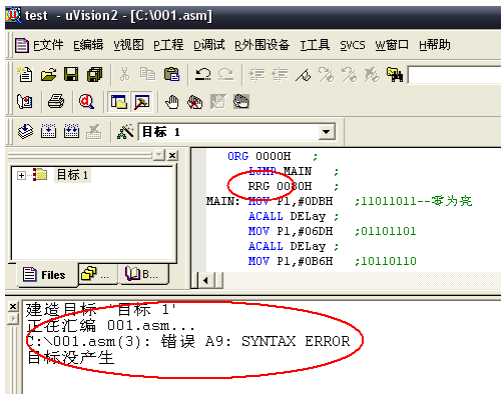


7 按以下两个向下小箭头的图标进行编译，（或者按 F7 快捷）编译成功后如会出现下图红箭头所指的文字正在汇编...0（错误），0（警告）这里的意思是没有错误，没有警告，表示编译成功。



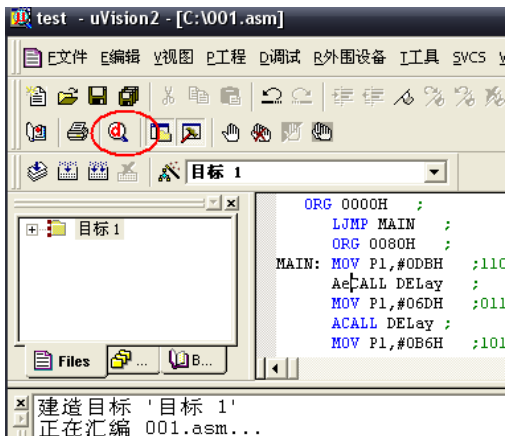
但是用户在开始编写程序时容易出现错误，现在我们故意将第 3 行的 ORG 0080H 改为 RRG 0080H，我们再次进行编译和链接编译/链接，因源程序中有语法错误，在信息输出窗口中有错误报告出现：“001.ASM(3): 错误 A9: SYNTAX ERROR”（第 3 行语法错误）

双击该行，定位到出错的位置，将错误改正后，再进行编译，程序编译和链接能顺利完成。

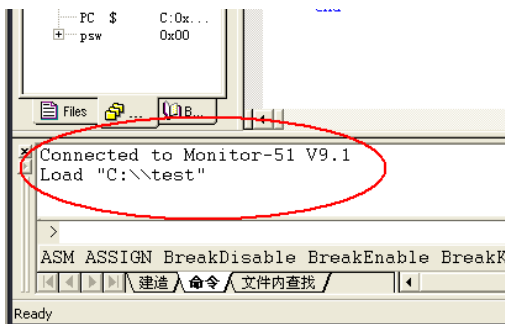


此时在 C 盘的根目录就生成了 test.hex 文件，有了这个 hex 文件，我们也可用 ISP 方式烧写到单片机实验）然后可以进行硬件仿真了，将仿真头插入目标板的 40pin 卡座，开始仿真，退出仿真时最好按一下 51mini 仿真器上的复位按键，小喇叭发出“滴”一声响。（一般不需要这样做，除非系统没有复位）

8 在按图第三个红色的 (debug) 或按 Ctrl + F5 快捷键可以进入仿真



此时界面将出现连接成功的提示! Connected to monitor-51 v9.1
其中 9.1 是仿真器的版本号（否则设置有误请仔细检查）



（仿真连接成功的提示）



第四章 51mini ni 仿真调试高级技巧

进入调试状态后，Debug 菜单项中的命令可以使用了，有关编译的工具栏按钮消失了，出现了一个用于运行和调试的工具栏，Debug 菜单上的大部份命令都有相应的快捷按钮。



从左到右依次是复位、运行、暂停、单步跟踪、单步、执行完当前子程序、运行到当前行、下一状态、打开跟踪、观察跟踪、反汇编窗口、观察窗口、代码作用范围分析、1# 串行窗口、内存窗口、性能分析、工具按钮命令；然后按一下图示第二个“运行”按钮。连接上相关的实验资源，本实验用一条 8PIN 的数据排线把实验仪的 CPU 部份的 P1 口（JP44）连接到八路指示灯部份的 JP32。这时你会看到实验仪的八个红色 LED，轮流点亮，表示运行成功，也可以查看相关的变量和参数，非常方便。

1.单步跟踪运行

使用菜单 Debug->Step 或上图第四个单步运行按钮或使用快捷键 F11 可以单步跟踪执行程序，在这里我们按下 F11 键，即可执行该箭头所指程序行，每按一次 F11，可以看到源程序窗口的左边黄色调试箭头指向下一行，如果程序中有 Delay 延时子程序，则会进入延时程序中运行。

2.单步运行

如果 Delay 程序有错误，可以通过单步跟踪执行来查找错误，但是如果 Delay 程序已正确，每次进行程序调试都要反复执行这些程序行，会使得调试效率很低，为此，可以在调试时使用 F10 来替代 F11（也可使用菜单 Step Over 或相应的命令按钮），在 main 函数中执行到 Delay 时将该行作为一条语句快速执行完毕。为了更好的进行对比，我们重新进入仿真环境，将反汇编窗口关闭，不断按 F10 键，可以看到在源程序窗口中的左边黄色调试箭头不会进入到延时子程序。

3.全速运行

点击工具栏上的“运行”按钮或按 F5 键启动全速运行，全速执行程序，此时用户板上的 P1 口所接 LED 以流水灯状态显示。

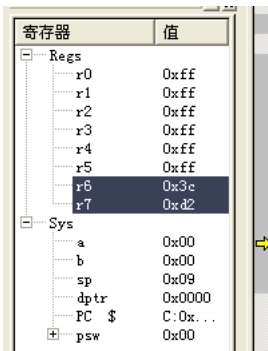
4.暂停

点击工具栏上的按钮，此时用户板上的 P1 口所接 LED 停止以流水灯状态显示，只有一个 LED 灯点亮（取决于暂停前的 P1 的值）。

5.观察 / 修改寄存器的值



Project 窗口在进入调试状态后显示 Regs 页的内容，包括工作寄存器 R0~R7 的内容和累加器 A、寄存器 B、堆栈指针 SP 的内容。



用户除了可以观察以外还可自行修改，例如将寄存器 a 的值 0x62 改为 0x85。

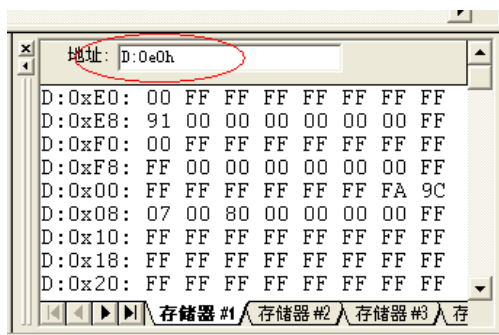
方法一：用鼠标点击选中单元 a，然后再单击其数值位置，出现文字框后输入 0x85 按回车键即可；

方法二：在命令行窗口，输入 A=0x85，按回车键将把 A 的数值设置为 0x85。

6. 观察 / 修改存储器的数据

点击菜单“视图 View->存储器窗口 Memory Windows”，便会打开 存储器 Memory 窗口(如窗口已打开，则会关闭)，Memory 窗口可以同时显示 4 个不同的存储器区域，点击窗口下部分的编号可以相互切换显示。

在 存储器 1 (Memory#1) 的地址输入栏内输入“D:0e0h”，按回车键后，可以从内部可直接寻址 RAM 的 e0H 地址处开始显示，e0H 地址的值就是寄存器 a 的值，应与主寄存器窗口下的值相同。



点击窗口下部分的存储器 2 (Memory#2)，在 Address 输入栏内输入“D:0e0h”，按回车键后，可以从内部可间接寻址 RAM 的 e0H 地址处开始显示。

点击窗口下部分的 Memory#3，在 Address 输入栏内输入“C:0x0021”，按回车键后，可以从代码区域 0000H 地址处开始显示，这时各地址值应与在反汇编窗口中的值相同。

点击窗口下部分的 Memory#4，在 Address 输入栏内输入“X:00h”，按回车键后，可以从 xdata RAM 区域 0000H 地址处开始显示。

通过 Memory 窗口修改数据

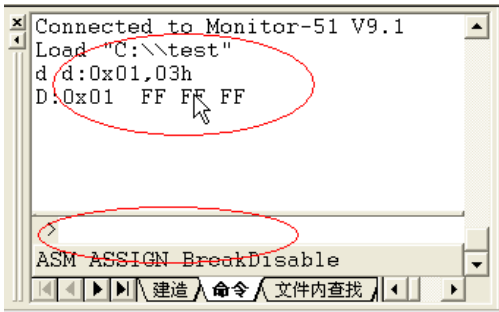
在 Memory 窗口中显示的数据可以修改。例如，要改动 data 区域 0XE0 地址的数据内容：

把鼠标移动到该数据的显示位置，按动鼠标右键在弹出的菜单中选中：更新存储器“Modify Memory at D:0xE0”，在弹出对话框的文本输入栏内输入相应数值，按回车键或点击 OK，修改完成。



通过命令行查看数据

例如，我们想查看 data 空间从 0x01 到 0x03 地址的内容，可在命令输入窗口输入“d d:0x01,03H”，回车即可完成，d 表示 data 空间，0x01 表示起始地址，03H 表示结束地址（注意两种 16 进制的表示方法在这里都可以接受），输出结果在信息输出窗口中显示出来。

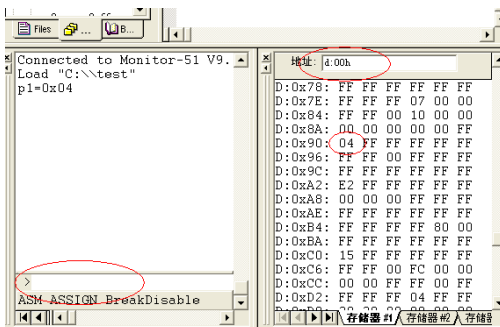


通过命令行修改数据

例如，我们想把 p1口的值从0x02数据修改成 0x05，可在命令输入窗口输入“p1=0x04”，回车即可完成，此时如果p1口接有led,将立即看到led状态改变。

通过 Memory 窗口可以看到修改后的数据：

在地址栏输入 d:00h然后回车。存储器1 的d:0x90后的第一个数据就变成了04，这正是我们刚才修改的结果。



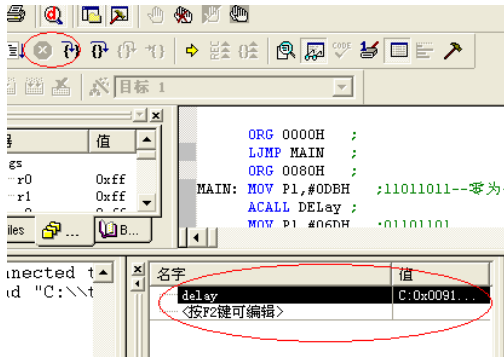
7. 观察 / 修改变量的值

在暂停程序运行时，可以观察到有关的变量值。

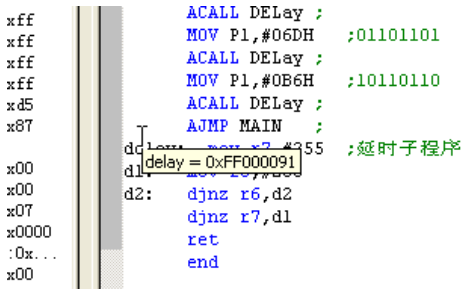
在 监视/调用堆栈 (Watch) 窗口“局部”页自动显示当前正在使用的局部变量，不需要用户自己添加。监视 (Watch)



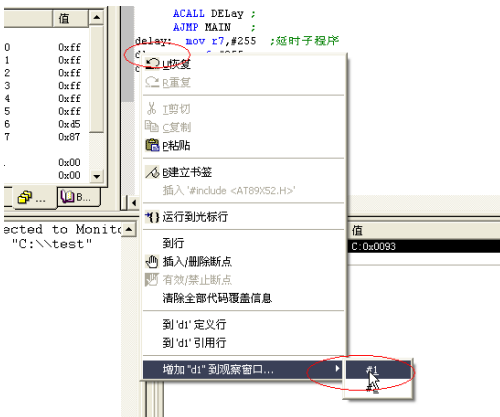
页显示用户指定的程序变量。（先按F2键,然后输入变量的名称例如“delay”然后回车）



移动鼠标光标到要观察的变量“delay”上停留大约一秒钟，就弹出一个“变量提示”块出来。



将鼠标移动到一个变量名“d1”的上面，点击鼠标右键，出现快捷菜单，选中：“增加d1到观察窗口”（Add “d1” to Watch Windows...）选项，子菜单中会出现#1 和#2 的选项，点击后该变量就会加入对应的监视/调用堆栈(Watch) 窗口。



增加d1到观察窗口”（Add “d1” to Watch Windows...）选项后 对应的监视窗口显示了d1 的值。



名字	值
d1	C:0x0093
<按F2键可编辑>	

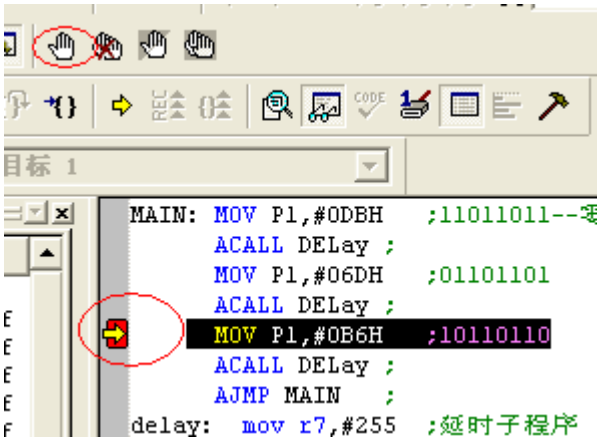
修改变量的数值方法：用鼠标左键点击该行的变量数据栏，然后按 F2 键出现文本输入栏后，输入修改的数据，确认正确后按回车键。

8. 复位

如果用户想重新开始运行用户程序，可以点击工具栏上的复位按钮，对仿真器的用户程序进行复位。仿真器复位后，程序计数器 PC 指针将复位成 0000H，另外，一些内部特殊功能寄存器在复位期间也将重新赋值，例如 A 将变为 00H，DPTR 变为 0000H，SP 变为 07H，I/O 口变为 0FFH，此时用户板上的 P1 口所接 LED 指示灯会全部熄灭。

9. 设置断点

将光标移至待设置断点的源程序行，如“MOV P1,#0B6H”行。点击工具栏上的“断点”图标，可以看到源程序窗口中该行的左边出现了一个红色的断点标记。（如果再点一下这个图标则清除这个断点）同样的方法，您可以设置多个断点。



10. 带断点的全速运行

按动 F5 启动全速运行，全速执行程序，当程序执行到第一个断点时，会暂停下来，这时你可以观察程序中各变量的值及各端口的状态，如下图：第一个断点在 mov p1,#06dh 之后，此时在存储器窗口 Memory Windows”，寄存器 1 (Memory#1) 的地址输入栏内输入“D:000h”，按回车键后，可以从内部可直接寻址地址数据为 6D 和程序设计的相同！



```

ORG 0000H ;
MAIN: MOV P1,#0DBH ;11011011--零为亮
      ACALL DELAY ;
      MOV P1,#06DH ;01101101
      ACALL DELAY ;
      MOV P1,#0B6H ;10110110
      ACALL DELAY ;
      AJMP MAIN ;
delay: mov r7,#255 ;延时子程序
      mov r6,#255
      djnz r6,d2

```

地址	内容
D:0x00	FF FF FF FF FF FF 00 00 8B 00 91 00 80 00 00 FF FF FF
D:0x30	FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
D:0x60	FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
D:0x90	6D FF FF FF FF FF FF FF FF 00 FF FF FF FF FF FF FF
D:0xC0	15 FF FF FF FF FF FF FF FF 00 FC 00 00 00 00 FF FF
D:0xF0	00 FF FF FF FF FF FF FF FF 00 00 00 00 00 00 FF FF
D:0x20	FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
D:0x50	FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
D:0x80	FF 07 00 00 FF FF 00 10 00 00 00 00 00 00 00 FC FF
D:0xB0	FF FF FF FF FF FF FF FF 80 00 FF FF FF FF FF 15 FF
D:0xE0	00 FF FF FF FF FF FF FF 91 00 00 00 00 00 00 FF FF
D:0x10	FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF

此时用户目标板上会显示当前断点的状态，继续按动 F5 启动全速运行，程序执行到第二个断点时，会暂停下来，在变量观察窗口中，RAM的值为B6；继续按动 F5 启动全速运行，程序又会执行到第一个断点处暂停，此时用户目标板上的灯又显示相应的状态。断点是仿真器调试的重要手段，请您仔细反复的练习直到熟练。


11.清除程序中所有断点

如果想取消全部的断点全速运行时，要是逐个取消将是很烦琐的事，我们只要点击工具栏相应的图标，就可清除程序中所有断点。

12.执行到光标处

在体验“执行到光标处”之前，我们先点击工具栏上复位的图标，对仿真器的用户程序进行复位，把鼠标放在想要停止的行点一下，再按“执行到光标处”程序全速执行到光标所在行，这与我们在前面看到的带断点的全速运行相类似。

13.退出仿真

先按暂停按钮，再按复位，再按  开启/关闭调试模式按钮，则退出仿真状态又重新回到编辑模式。（如果不能正确退出，请按一下仿真器上的复位按钮），

此时可以对程序修改，然后重新编译，再按开启/关闭调试模式按钮，就又进入仿真模式了。

请注意！由于KEIL软件在仿真状态时和仿真器始终保持通信联系，为此退出仿真的时候最好按照以上顺序退出，不要强行退出，否则容易造成电脑死机。

到这里也就完成了 keil 软件的标准操作过程，由于 KEIL 操作过程较复杂，请您自行练习直到熟练！

第五章：常见问题 FAQ



为何仿真状态不能正确的联机？

1 您可能没有正确的安装好 USB 驱动，如果安装正常，系统的：控制面板\系统\硬件\设备管理器中看到以下端口信息，表示系统已经正确的安装了 USB 驱动。如果没有显示下面的信息，请你重新安装 USB 驱动或者换一台电脑安装 USB 驱动：有的时候如果您更换了 USB 口也可能需要重新安装 USB 驱动。



2 仿真需要 keil uv2 7.0 以及以上的版本软件，（有些早期版本的 KEIL 不能兼容请谨慎）并且注意这个软件必须安装在 c 盘的根目录，不能安装在其他的目录！建立文件的时候，也不要保存在其他的目录，更不要使用中文的文件名和文件夹。

3 串口的选择是否正确（如果是新系统没有安装其他 usb 设备一般是 com3）？由于仿真/烧写都需要用到串口，为此：仿真操作时，必须关闭掉 XLI SP 烧写软件, 串口调试等可能用串口的软件。

4 仿真器是否已经复位？请按一下仿真器的复位键，注意短按一下，喇叭“滴”的响一下表示复位 ok. 如果是长按 3 秒以上喇叭“滴滴”响两下，是脱机运行状态并不是复位，需要重新按复位。

5 检查一下晶体频率选择，如果你选择的是 USER 那么注意检查用户板的晶体是否起振？（如果不能确定请选择板上的 22.1184MHZ）

6 检查您的仿真器的方向插入是否正确？特别注意，仿真器的第一脚写有 1PIN，对应 cpu 的第一脚。可以用万用表检查一下。如果插入目标板反了，则系统可能会发热，请立即调换方向。

7 您的目标板可能本身不正常，没有正确的时钟和复位（虽然这些不是仿真器所必需的）您可以用直接烧片的方式运行，如果运行正常，则表示目标板正常。

8 被仿真的目标板需要另外供电吗？

仿真头本身可以提供 5v500MA 的电源供给外部目标板使用，一般情况下已经足够 51 系统的使用，如果您感觉不够也可以在外部目标板另外供给 5V 电源并联运行，系统设计了完善的保护，不会导致损坏。

9 为何有些端口没有动作？

有可能是仿真针脚接触不良，最好是打开仿真器的外壳用万用量表一下并自行修复，相信对您不是难事。

10 被仿真的目标板的时钟和复位还有效吗？

在使用 51mini 仿真器的时候，目标板的时钟和复位都被仿真器接管，无论目标板是否有时钟和复位。

11 为何程序运行不正常？

C51 用户请在您的代码的 main() 函数前面，加上一句：

```
char code dx516[3] _at_ 0x003b;
```





ASM 汇编用户请将你的主程序跳过中断区直到 0050H 以后，如：


```
org 0000H
ljmp main
org 0050H
main: ...;程序开始
```


如果以上设置你没有做，在装载过程中，仿真器会发出“嘀嘀嘀”的三声短声报警，这时的仿真结果将可能不正确。

现在随着技术的发展，CPU 的 00-40H 之间已经越来越多的被新的寄存器占用，而不是以前的 00-30H，养成良好的程序书写习惯很重要。如果您的程序是 30H，请你改为 40H 或者以后如 50h，80h，如果您有所疑问请您仔细参考本光盘的测试程序举例中的例子，都是修改过的。

 12 如果无法确认异常，请在程序下载进去后再长按复位键 3 秒，程序直接运行，如果脱机运行结果也不正确，请检查您的程序编写的问题。

 13 仿真的概念就是模拟，51mini 不能保证完全真实的仿真所有 51 资源，不能访问外部存储器比如 8155,62256 等。如果您对仿真的状态有疑问，请您用直接烧片的方式运行。

 如果要仿真 2051，用户可参考 51 和 2051 对应的管脚资料自制一个转接头。一般我们建议您直接用 51 仿真，编程时避开 2051 没有的资源。

 (仿真器连接示意图连接 USB 到电脑，连接仿真头到目标用户板，注意晶体选择到您想要的频率)

(附录) 51 单片机的特殊功能寄存器详细列表

Register	(MSB)→(LSB) Byte								Address
Symbol	b7	b6	b5	b4	b3	b2	b1	b0	Address
P0	P0.7	P0.6	P0.5	P0.4	P0.3	P0.2	P0.1	P0.0	80H(128)
SP									81H(129)
DPL									82H(130)
DPH									83H(131)
PCON	SMOD	-	-	-	GF1	GF0	PD	IDL	87H(135)
*PCON	SMOD	-	-	WLE	GF1	GF0	PD	IDL	87H(135)
TCON	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	88H(136)
TMOD	GATE	C/T	M1	M0	GATE	C/T	M1	M0	89H(137)
TL0									8AH(138)
TL1									8BH(139)
TH0									8CH(140)
TH1									8DH(141)
P1	P1.7	P1.6	P1.5	P1.4	P1.3	P1.2	T2EX	T2	90H(144)
*P1	SDA	SCL	RT2	T2	CT3I	CT2I	CT1I	CT0I	90H(144)



SCON	SM0	SM1	SM2	REN	TB8	RB8	TI	RI	98H(152)
SBUF									99H(153)
P2	P2.7	P2.6	P2.5	P2.4	P2.3	P2.2	P2.1	P2.0	0A0H(208)
IE	EA	-	ET2	ES	ET1	EX1	ET0	EX0	0A8H(168)
*IEN0	EA	EAD	ES1	ES0	ET1	EX1	ET0	EX0	0A8H(168)
+CML0									0A9H(169)
+CML1									0AAH(170)
+CML2									0ABH(171)
+CTL0									0ACH(172)
+CTL1									0ADH(173)
+CTL2									0AEH(174)
+CTL3									0AFH(175)
P3	RD	WR	T1	T0	INT1	INT0	TXD	RXD	0B0H(176)
IP	-	-	PT2	PS	PT1	PX1	PT0	PX0	0B8H(184)
*IP0	-	PAD	PS1	PS0	PT1	PX1	PT0	PX0	0B8H(184)
+P4	CMT1	CMT0	CMSR5	CMSR4	CMSR3	CMSR2	CMSR1	CMSR0	0C0H(192)
+P5	ADC7	ADC6	ADC5	ADC4	ADC3	ADC2	ADC1	ADC0	0C4H(196)
+ADCON	ADC.1	ADC.0	ADEX	ADCI	ADCS	AADR2	AADR1	AADR0	0C5H(197)
+ADCH									0C6H(198)
T2CON	TF2	EXF2	RCLK	TCLK	*EXEN2	TR2	C/T2	*CP/RL2	0C8H(200)
*TM2IR	T2OV	CMI2	CMI1	CMI0	CTI3	CTI2	CTI1	CTI0	0C8H(200)
+CMH0									0C9H(201)
RCAP2L									0CAH(202)
*CMH1									0CAH(202)
RCAP2H									0CBH(203)
*CMH2									0CBH(203)
TL2									0CCH(204)
*CTH0									0CCH(204)
TH2									0CDH(205)
*CTH1									0CDH(205)
+CTH2									0CEH(206)
+CTH3									0CFH(207)
PSW	CY	AC	F0	RS1	RS0	OV	F1	P	0D0H(208)
+S1CON	ENS1	STA	STO	SI	AA	CR1	CR0		0D8H(216)
+S1STA	SC4	SC3	SC2	SC1	SC0	0	0	0	0D9H(217)
+S1DAT									0DAH(218)
+S1ADR									0DBH(219)
									SLAVE ADDRESSGC
ACC	ACC.7	ACC.6	ACC.5	ACC.4	ACC.3	ACC.2	ACC.1	ACC.0	0E0H(224)



+IEN1	ET2	ECM2	ECM1	ECM0	ECT3	ECT2	ECT1	ECT0	0E8H(232)
+TM2CON	T2IS1	T2IS0	T2ER	T2B0	T2P1	T2P0	T2MS1	T2MS0	0EAH(234)
+CTCON	CTN3	CTP3	CTN2	CTP2	CTN1	CTP1	CTN0	CTP0	0EBH(235)
+TML2									0ECH(236)
+TMH2									0EDH(237)
+STE	TG47	TG46	SP45	SP44	SP43	SP42	SP41	SP40	0EEH(238)
+RTE	TP47	TP46	RP45	RP44	RP43	RP42	RP41	RP40	0EFH(239)
B	B.7	B.6	B.5	B.4	B.3	B.2	B.1	B.0	0F0H(240)
+IP1	PT2	PCM2	PCM1	PCM0	PCT3	PCT2	PCT1	PCT0	0F8H(248)
+PWM0									0FCH(252)
+PWM1									0FDH(253)
+PWMP									0FEH(254)
+T3									0FFH(255)

Notes:

1. * denotes the difference between 80C552 and 8051
2. + denotes the addition of 80C552

2009-12-18 第一次印刷 印数: 2000 深圳市学林电子有限公司 版权所有 翻版必究 如果您对操作有所疑问请登陆 WWW.8951.COM 获得帮助