# 第二章 Keil 应用

作者: 彭保基 西安交通大学 电信学院 电子 2002 级

版本: V1.1 写作时间: 2004年12月---2005年3月

Keil 是目前世界上最好的 51 单片机的汇编和 C 语言的开发工具。支持汇编,C 语言以及混合编程。Keil 提供包括 C 编译器、宏汇编、连接器、库管理和一个功能强大的仿真调试器在内的完整开发方案,通过一个集成开发环境( μ Vision)将这些部分组合在一起。掌握这一软件的使用对于 51 系列的单片机的用户来说是十分必要的,如果你使用 C 语言编程,那么 Keil 几乎就是你的不二之选,即使你不使用 C 语言而仅使用汇编语言,其方便易用的集成环境、强大的软件仿真调试工具也会令你事半功倍。

## 第一节 一般开发流程

1) 打开 Keil 后, 出现图 2-1 画面; 点击菜单 project,选择 new project:

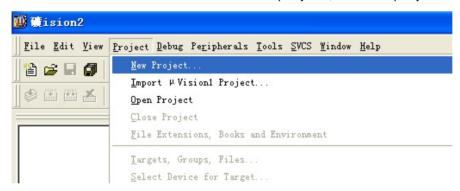


图 2-1

2) 然后选择你要保存的路径,输入工程文件的名字;比如:如图 2-2 所示,保存到 Keil 目录 里,工程文件的名字为 test.uv2,然后点击保存。

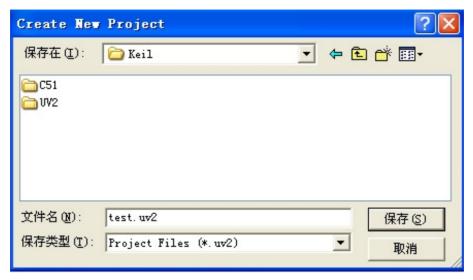


图 2-2

3) 这时会弹出一个对话框,要求你选择单片机的型号,Keil c51 几乎支持所有的 51 核的单片机;你可以根据你使用的单片机来选择,一般可选用 Atmel 的 AT89C51,写好的程序也可以在其它的 51 系列的单片机上用。如图 2-3 所示,选择 AT89C51 之后,然后点击确定。右边一栏(Description)是对这个单片机的基本的说明。包括 I/O 口、计数器/定时器、中断源、ROM、RAM等。

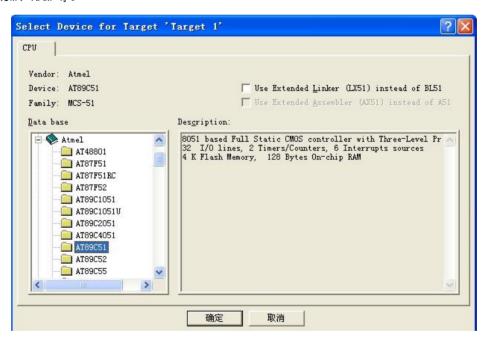


图 2-3

4) 这时要新建一个源程序文件,建立一个汇编或 C 文件,点击菜单 File->New,如图 2-4。如果你已经有源程序文件,可以点击菜单 File->Open,打开已经有的源程序文件。

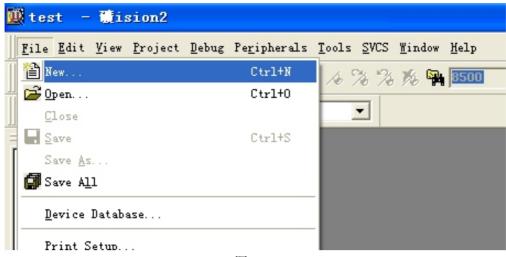


图 2-4

5) 如图 2-5, 选择菜单 File->Save。也可以先不保存, 但是先保存是一个好的编程习惯。

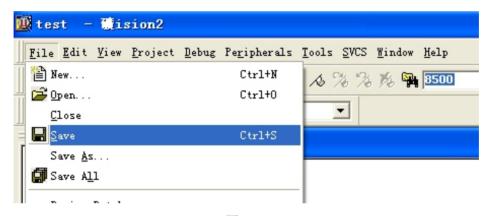


图 2-5

6) 选择你要保存的路径,在文件名里输入文件名,注意一定要输入扩展名,如果是C程序文件, 扩展名为.c,如果是汇编文件,扩展名为.asm。如图 2-6: 这里是要存储一个C源程序文件, 所以输入.c扩展名,保存为 test.c 的名字,(当然也可以保存为其他名字),点击保存。

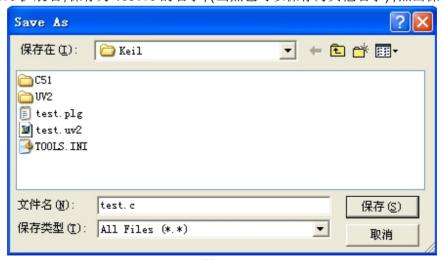


图 2-6

7) 输入一段简单的程序: #include <AT89X51.H> #define uint unsigned int

```
delay(uint t)
                              //延时函数
       {
        uint i:
        for(i=0;i<t;i++);
void main(void)
                                //主函数
      while(1)
           {
           P1 6=0;
           delay(50000); //延时 50ms 左右
           P1 6=1;
           delay(50000);
     }
             ∰test - ∰ision2 - [D:\Keil\test.c]
              File Edit View Project Debug Peripherals Tools SVCS Window Help
               🖺 🚅 🖫 🗿 🐰 🗎 📵 🗅 🗅 🗜 車 🛵 % % % 🥦 🕦
               Iarget 1
                                    #include <AT89X51.H>
                                    #define uint unsigned int
              + Target 1
                                    delay(uint t) //延时函数
                                          uint i;
                                          for(i=0;i<t;i++);
                                                 //主函数
                                    void main(void)
                                         while(1)
                                            Pl_6=0;
                                            delay(50000); //延时50ms左右
                                            Pl 6=1:
                                            delay(50000);
```

这时你就可以深刻体会到先保存的好处了;如图 2-7,先保存的话,C 或汇编语言的关键字就会变成蓝色;这样编程的时候就可以更少的犯错误了。

图 2-7

8) 点击 Target 1 前面的+号,展开里面的内容 Source Group 1,如图 2-8:

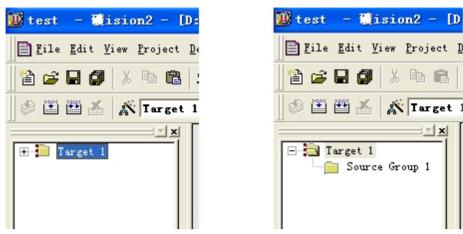


图 2-8

9) 如图 2-9: 用右键点击 Source Group 1(注意用鼠标的右键,而不是左键),将弹出一个菜单,选择 Add Files to Group 'Source Group 1'。

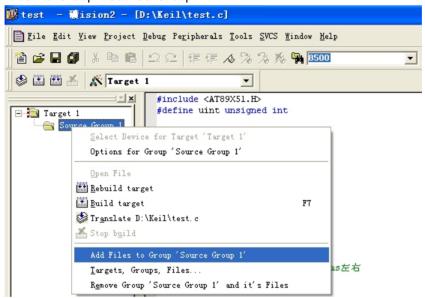


图 2-9

10) 选择刚才的文件 test.c,文件类型选择 C Source file(\*.c)。因为我们的是 C 程序文件,所以选择该类型;如果是汇编文件,就选择 asm source file;如果是目标文件,选择 Object file;如果是库文件,选择 Library file。最后点击 Add;点击 add 之后,窗口不会消失,(如果要添加多个文件,可以不断添加),添加完毕此时再点击 Close 关闭该窗口。

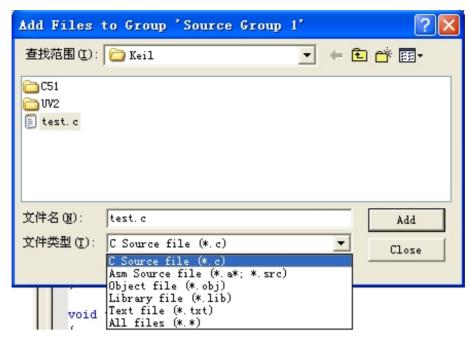


图 2-10

11) 这时在 Source group 1 前就多了一个+号,点击+号,就可以看见里就有 test.c 文件,如图 2-11:

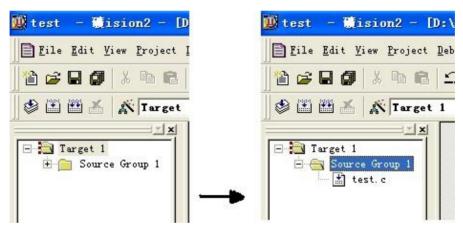


图 2-11

12) C程序文件已被我们加到了项目中了,下面就可以进行编译运行了。我们先来看图 2-12 吧,图中 1、2、3 都是编译按钮;不同是 1 是用于编译单个文件,2 是编译当前项目,3 是重新编译;如果先前编译过一次之后文件没有做动编辑改动,这时再点击 1 或 2 是不会再次重新编译的,但是每点击 3 一次均会再次编译链接一次,不管程序是否有改动。4 是停止编译按钮,只有点击了前三个中的任一个,停止按钮才会生效。6 窗口显示的是当前项目所包含的文件;7 是程序写入窗口;在 8 中可以看到编译的错误信息和使用的系统资源情况等,以后我们要查错就靠它了。5 左边有一个小放大镜的按钮,这就是开启\关闭调试模式的按钮。所有的快捷图标都存在于菜单中,不过笔者建议大家使用快捷图标,所以就不一一介绍菜单中的选项了。

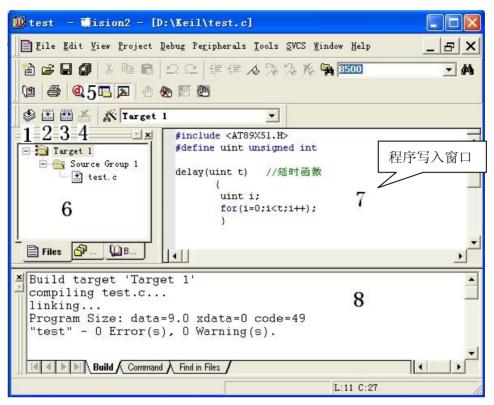


图 2-12

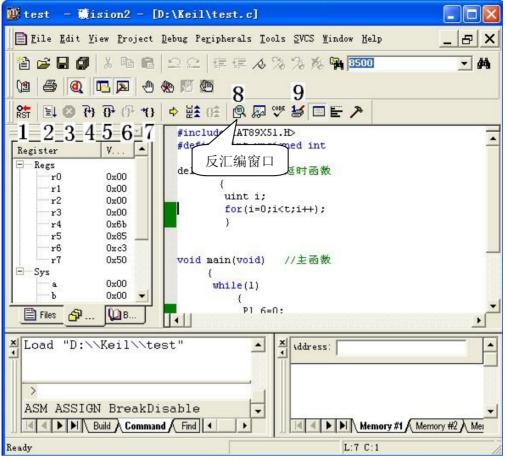


图 2-13

14) 我们再回过头看一下第7步写入的程序,这段程序的功能是让P1.6 口间隔 50ms 电平变化一次,也就是输出频率为10Hz 的方波;驱动接在P1.6 口的蜂鸣器发声。下面我们要从软件仿真看能否实现此功能,从中学习软件仿真的一般流程。首先,如图 2-14:点击Peripherals->I/O-Ports->Port 1,选择P1口。

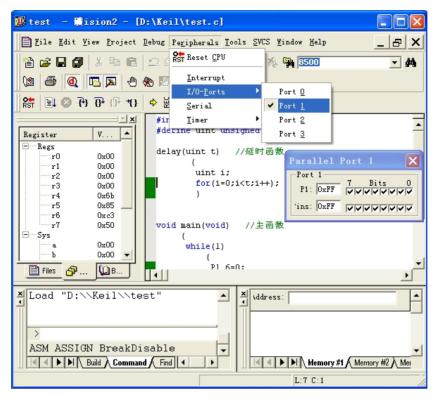


图 2-14

15) 如图 2-15, 点击 View->Periodic Window Update (前面加个勾), 定时刷新窗口,这样 P1 口的变化才可以随时看出来。

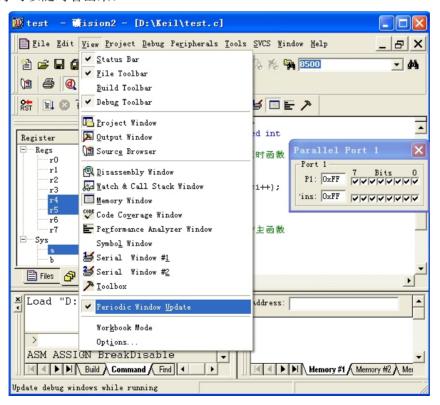


图 2-15

16)接下来就可以直接运行程序了,点击图 2-13 中的 2运行程序;如图 2-16 所示,这是我们可以看到 P1.6 的电平在来回变化(有勾表示是高电平,无勾表示为低电平)。但并不是 50ms

变化一次,变化的快慢取决于你的电脑的配置,这就显现了软件仿真的局限性。这也是要强调并重视硬件仿真的原因。

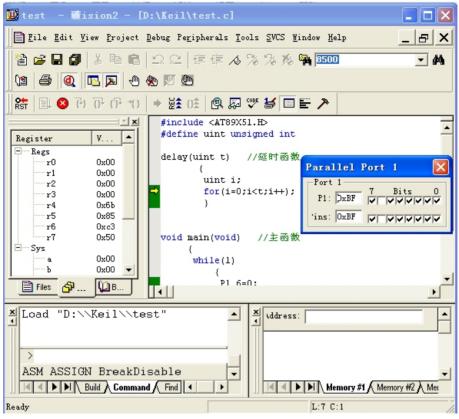


图 2-16

- 17) 若发现程序有误,则可点击图 2-13 中的 5,退出调试模式窗口,返回程序输入窗口;把程序修改好、保存、编译、再重新仿真;切记修改后一定要**保存、编译**之后才能进行**仿真**,因为仿真程序执行的是编译之后的文件。
- 18) \*\*\*\*\*\*\*\*\*\*\*\*\*
- 10) \*\*\*\*\*\*\*\*\*\*\*\*

# 第二节 硬件仿真应用

1) 用鼠标右键(注意用右键)点击左边的 Target 1,会出现一个菜单,选择 Options for Target 'Target 1',如图 2-17。

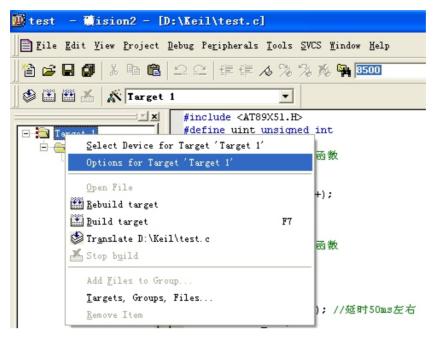


图 2-17

2) 在项目设置 Target 中,根据硬件,设置晶振频率为 11.0592M;如图



图 2-18

3) 在项目设置 Debug 页中选择对应本仿真器的 KeilMon51 驱动,如下图 2-19 所示。选择默认的 Use: Keil Monitor-51 Driver; Restore Debug Session Settings 下面的选项是选择在仿真时能使用的工具窗口,如内存显示,断点等等。



图 2-19

4) 单击图 2-19 中的 Setting 进行图 2-20 中的仿真器设置。设置好串口号,波特率;晶振为 11.0592M 时选 38400。Cache Options 为仿真缓存选项,选取后会加快仿真的运行的速度。

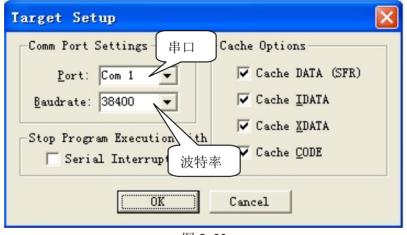


图 2-20

设置好后编译运行程序就可以连接仿真器了,切记先进行硬件**复位**之后再连接仿真。连接成功会出现如图 2-21 的画面。图 2-21 中指示仿真器的监控程序版本为 F-MON51V3.4 版。图中的小红点为断点,双击可以设置和取消断点,点击'运行'按钮可以运行到下一个断点。硬件仿真的使用方法和软件仿真相差不多。



图 2-21

5) 如连接不成功就出现图 2-22 的画面,这时可以先按硬件**复位键**后再按"Try Again",如果还不成功连接的话则应检查软件设置和硬件电路。排除故障之后可以重新连接。



图 2-22

6) 程序调试时常用窗口如图 2-23 所示;有关它们的详细说明请参考第四节。

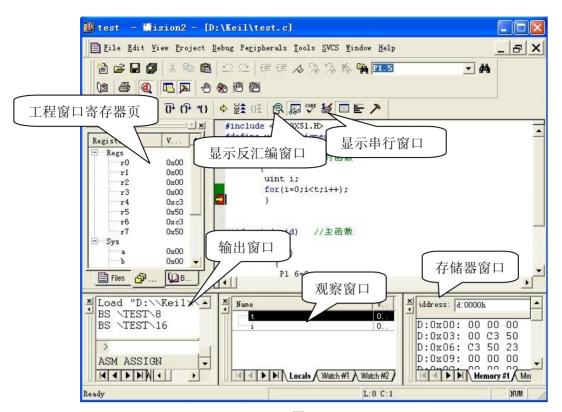


图 2-23

- 7) \*\*\*\*\*\*\*\*\*
- 8) \*\*\*\*\*\*\*\*\*
- 9) \*\*\*\*\*\*\*\*\*

# 第三节 Keil的常用快捷按钮

| 快捷图标           | 快捷键      | 说明               |
|----------------|----------|------------------|
| RST            |          | 复位CPU            |
|                | F5       | 全速运行,直到遇到一个中断    |
| 8              | ESC      | 停止程序运行           |
| <del>(4)</del> | F11      | 单步执行程序遇到子程序则进入   |
| <b>O</b> +     | F10      | 单步执行程序跳过子程序      |
| <b>(3</b> -    | Ctrl+F11 | 执行到当前函数的结束(跳出函数) |
| *{}            |          | 运行到光标位置处         |
| 4>             |          | 显示下一条指令          |
| ₩.             |          | 使能/禁止程序运行轨迹的标识   |
| 02             |          | 显示程序运行过的指令       |
| <b>_</b>       |          | 显示汇编窗口           |
| Fair I         |          | 显示变量观察窗口         |
| CODE           |          | 代码覆盖窗口           |
| 15             |          | 显示串口通信数据窗口       |
|                |          | 打开存储器窗口          |
| E              |          | 打开设置性能分析的窗口      |
| 7              |          | 工具栏              |
| *              |          | 查找               |
| <b>6</b>       |          | 资源浏览窗口           |
|                |          | 打印               |
|                | Ctrl+F5  | 开始/停止调试模式        |
|                |          | 工具箱窗口            |
| M              |          | 输出窗口             |
| 6              |          | 设置/取消当前行的断点      |
| <b>€</b>       |          | 取消所有的断点          |
| 選              |          | 使能/禁止当前行的断点      |
| ***            |          | 禁止所有的断点          |
| <b>♦</b>       | Ctrl+F7  | 编译当前文件           |
| 200220         | F7       | 编译修改过的文件并生成应用    |
|                |          | 重新编译所有的文件并生成应用   |
|                |          | 停止生成应用的过程        |
|                |          |                  |

## 第四节 Keil 程序调试窗口简介

Keil 软件在调试程序时提供了多个窗口,主要包括输出窗口(Output Windows)、观察窗口(Watch&Call Statck Windows)、存储器窗口(Memory Window)、工程窗口(Project Window)、反汇编窗口(Dissambly Window)串行窗口(Serial Window)等。进入调试模式后,可以通过菜单 View 下的相应命令或快捷图标打开或关闭这些窗口。下面只简单的介绍一下这几种窗口。

#### 一、 输出窗口

进入调试程序后,输出窗口自动切换到 Command 页。如图 2-24。该页用于输入调试命令和输出调试信息。对于初学者,可以暂不学习调试命令的使用方法。



图 2-24

#### 二、 观察窗口

观察窗口是很重要的一个窗口,工程窗口中仅可以观察到工作寄存器和有限的寄存器如A、B、DPTR等,如果需要观察其它的寄存器的值或者在高级语言编程时需要直接观察变量,就要借助于观察窗口了。如图 2-25。一般情况下,我们仅在单步执行时才对变量的值的变化感兴趣,全速运行时,变量的值是不变的,只有在程序停下来之后,才会将这些值最新的变化反映出来,但是,在一些特殊场合下我们也可能需要在全速运行时观察变量的变化,此时可以点击 View->Periodic Window Updata(周期更新窗口),确认该项处于被选中状态,即可在全速运行时动态地观察有关值的变化。但是,选中该项,将会使程序模拟执行的速度变慢。

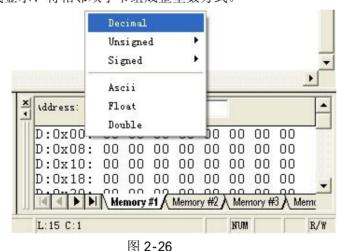


图 2-25

#### 三、 存储器窗口

如图 2-26;存储器窗口中可以显示系统中各种内存中的值,通过在 Address 后的编辑框内输入"字母:数字"即可显示相应内存值,其中字母可以是 C、D、I、X,分别代表代码存储空间、直接寻址的片内存储空间、间接寻址的片内存储空间、扩展的外部 RAM 空间,数字代表想要查看的地址。例如输入 D: 0 即可观察到地址 0 开始的片内 RAM 单元值、键入 C: 0 即可显示从 0 开始的 ROM 单元中的值,即查看程序的二进制代码。该窗口的显示值可以以各种形式显示,如十进制、十六进制、字符型等,改变显示方式的方法是点鼠标右键,在弹出的快捷菜单中选择,该菜单用分隔条分成三部份,其中第一部份与第二部份的三个选项为同一级别,选中第一部份的任一选项,内容将以整数形式显示,而选中第二部份的 ASCII 项则将以字符型式显示,选中 Float 项将相邻四字节组成的浮点数形式显示、选中 Double 项则将相邻 8 字节组成双精度形式显示。第一部份又有多个选择项,其中 Decimal 项是一个开关,如果选中该项,则窗口中的值将以十进

制的形式显示,否则按默认的十六进制方式显示。Unsigned 和 Signed 后分别有三个选项: Char、Int、Long,分别代表以单字节方式显示、将相邻双字节组成整型数方式。



St (1) (1) (1) (2) (2) × Register Value Regs rO 0x00 0x00 r1 0x00 0x00 0.15 r5 Oxeb 0xc3 r6 r7 0x50 Sys 0x520x000x09 SD 0x09 Sp max 0x0000 PC \$ C:0x.. states 3908 42.4. sec 0x81 psw

图 2-27

#### 四、 工程窗口寄存器页

如图 2-27; 工程窗口寄存器页包括了当前的工作寄存器组和系统寄存器,系统寄存器组有一些是实际存在的寄存器如 A、B、DPTR、SP、PSW等,有一些是实际中并不存在或虽然存在却不能对其操作的如 PC、Status 等。每当程序中执行到对某寄存器的操作时,该寄存器会以反色(蓝底白字)显示,用鼠标单击然后按下 F2 键,即可修改该值。

#### 五、 反汇编窗口

点击 View->Dissambly Window 或快捷图标可以打开反汇编窗口,该窗口可以显示反汇编后的代码、源程序和相应反汇编代码的混合代码,可以在该窗口进行在线汇编、利用该窗口跟踪已找行的代码、在该窗口按汇编代码的方式单步执行,这也是一个重要的窗口。打开反汇编窗口,点击鼠标右键,出现快捷菜单,其中 MixedMode 是以混合方式显示,Assembly Mode 是以反汇编码方式显示。如图 2-28

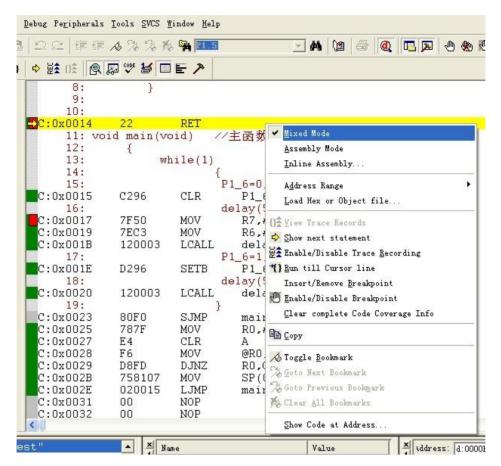


图 2-28

### 六、 串行窗口

Keil 提供了串行窗口,我们可以直接在串行窗口中键入字符,该字符虽不会被显示出来,但却能传递到仿真 CPU 中,如果仿真 CPU 通过串行口发送字符,那么这些字符会在串行窗口显示出来,用该窗口可以在没有硬件的情况下用键盘模拟串口通讯。由于部份 CPU 具有双串口,故 Keil 提供了两个串行窗口,我们选用的 SST89E564 芯片只有一个串行口,所以 Serial 2 串行窗口不起作用。

**小技巧:** 凡是鼠标单击然后按 F2 的地方都可以用鼠标连续单击两次(注意: 不是双击)来替代。

七、\*\*\*\*\*\*\*

第五节 \*\*\*\*\*\*\*\*