第三章　AVR 单片机开发工具

说明：为了使读者和用户迅速掌握AVR指令系统的功能，边学习，边实践，希望大家先学习<<第三章 AVR 开发工具>>。根据我们的实际教学经验，有的书籍是根据英文源文翻译，程序及说明可能不合中国人习惯，又由于印刷等多种原因，内容有出入，学起来较难。我们是参考有关资料，并在实际工作中验证，并编写有关测试程序(含中文注释)，在模拟调试软件窗口观察通过，或在实时仿真器或在SL-AVR开发下载实验器上验证通过，把测试实验程序刻在光盘上(也可上网下载 http://WWW.SL.COM.CN)，保证用户学习、实验时少走弯路。所以我们先学习系统软件的使用，然后学指令系统，用户一边学习 AVR 指令系统，一边学习系统软件编程调试，这样使指令功能流向看得见听得见，学习起来有声有色，达到事半功倍的效果。当学完所有指令，你也学会了用软件编程开发调试。我们的想法希望你能去边学边实践，并得到你的认可，我们就谢谢了。

3.1　AVR 单片机的编辑、编译

AVR 单片机实用程序源文件供用户学习参考，今后还将不断增加新内容，也欢迎用户来交流新程序。源程序在 :\AVR\AVR\asmpack\appnotes 目录下，源程序经编译(Assembler)后生成.OBJ 调试文件，HEX 下载文件，LIS 列表打印文件。

3.1.1 AVR Assembler 汇编文件的安装与打开：

打开光盘文件 *:\AVR\AVR\asmpack 文件夹，双击图标 

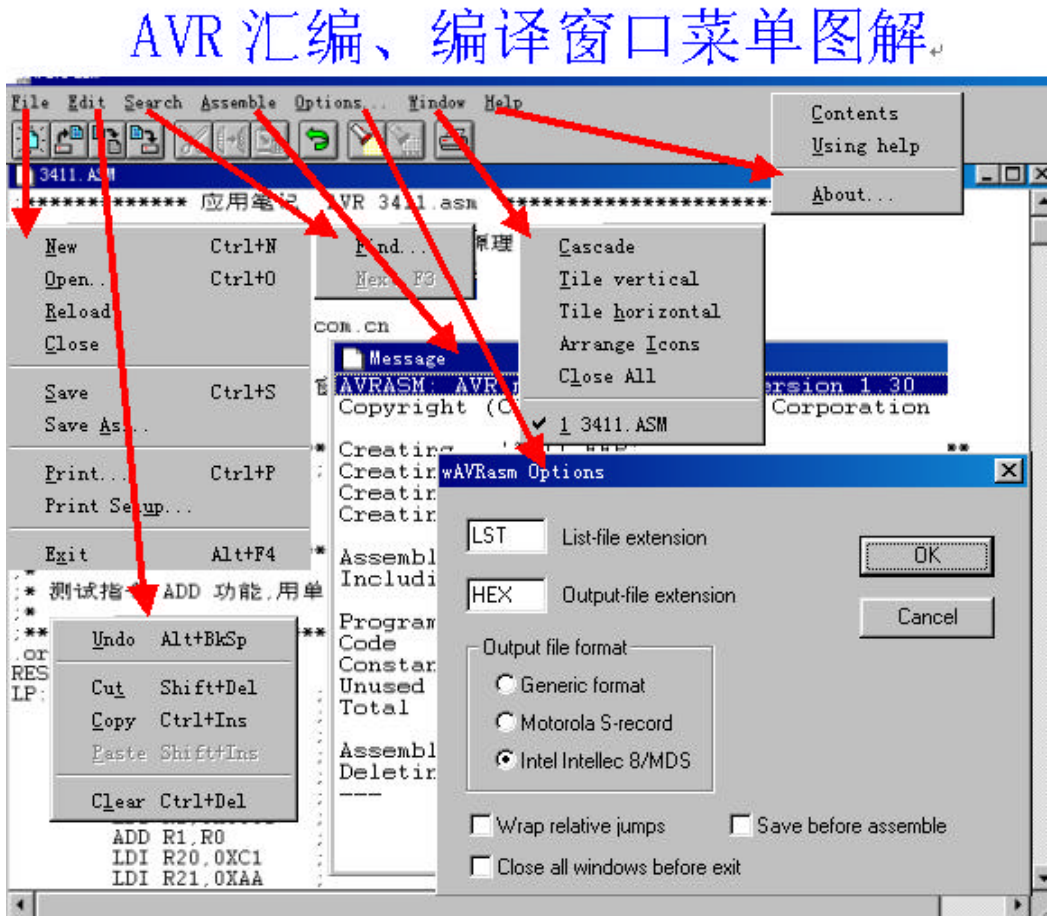
安装。

安装好后双击图标 

进入源文件编辑汇编窗口，

也可使该图标移到桌面成快捷菜单，

点击图标进入 AVR Assembler 源文件编辑汇编窗口。



3.1.2 AVR 单片机汇编语言源程序举例:

例一: 文件头不可少,以便了解该程序有关资料

```
;************* 应用举例  AVR 3411.asm  ************************
;*  <4411>表示对应第几章第几节第几段第几个实例
;* 标题:     测试指令功能原理
;* 版本:     1.0
;* 最后更新日期:2000.08.08
;*
;* 支援 E-mail: gzsl@sl.com.cn
;*
;* 描述
;*  用 AVR Studio 调试软件窗口观察指令执行变化情况
;* 作者: SL.
;* 程序适用于所有单片机
;********************************************************************************
.include "8515def.inc"   ;在编译调试中用到,绝不可少" *.inc "文件头
.org  $0000
    rjmpRESET        ;复位
```

```
;*********************************************************************
;*
;* 测试指令 ADD 功能,用单步或连续单步调试
;*
;*********************************************************************
.org $0010            ;跳过中断区
RESET:
LP: LDI R16,0X11      ;立即数送寄存器 LDI 指令中寄存器必须 R≥R16,才能汇编成功!
    LDI R17,$33       ;0X11,$33 均为十六进制表示法
    ADD R17,R16       ;R17=0X44 SREG=0X00,H=0,S=0,V=0,N=0,Z=0,C=0,高位低位均无进位
    STS 0X0060,R16    ;内部 SRAM 地址必须≥0X0060,0X0060=0X11
    STS 0X0061,R17    ;0X0011=0X44
    LDS R0,0X0060     ;R0=0X11
    LDS R1,0X0061     ;R1=0X44
    ADD R1,R0         ;R1=0X55  SREG=0X00 , 高位低位均无进位
    LDI R20,0XC1;
    LDI R21,0XAA;
    ADD R21,R20       ;R21=0X6B SREG=0X19,S=1,V=1,C=1 ,高位有进位,低位无进位
    LDI R22,0X46;
    LDI R23,0X6A;
    ADD R23,R22       ;R23=0XB0 SREG=0X2C,H=1,V=1,N=1 , 高位无进位,低位有进位
    LDI R24,0XFF;
    LDI R25,0XFF;
    ADD R25,R24       ;R25=0XFE SREG=0X35,H=1,S=1,N=1,C=1 , 高位有进位,低位有进位
    RJMPLP       ;可循环反复调试检查

                    ;调试时打开 Registers(寄存器窗口),
                    ;Processor(处理器窗口--程序,堆栈,状态寄存器,X/Y/Z 等项),
                ;New Memory View(存储器窗口--数据,I/O,E2PROM,程序存储器窗口,
                    ;该程序仅需打开片内 SRAM 数据窗口)
```

例二:利用另存文件名,把例一变为例二,仅略修改文件头

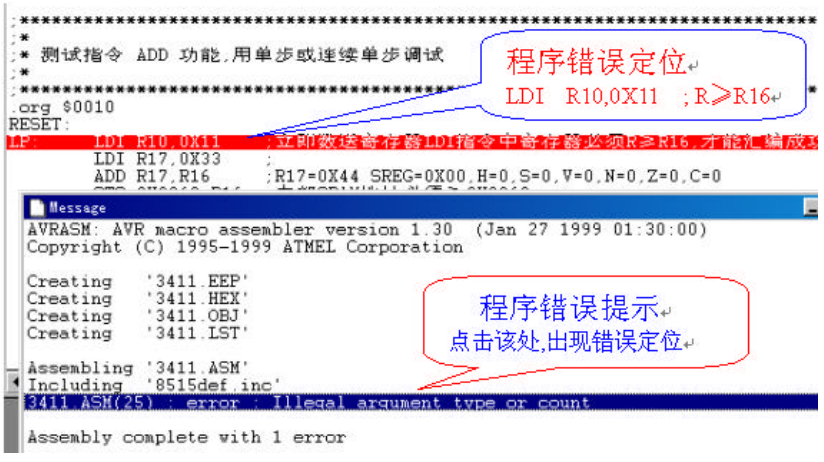
```
;************** 应用举例  AVR 4411B.asm  ************************
;* 标题:     测试指令功能原理
;* 版本:     1.0
;* 最后更新日期:2000.08.08
;* 支援 E-mail: gzsl@sl.com.cn
;* 描述
;*  在 AVR Studio 调试软件窗口中置数方法输入数据,用单步观察指令执行变化情况
;* 作者: SL.
;* 程序适用于所有单片机
;*********************************************************************
.include "8515def.inc" ;在编译调试中用到, 绝不可少" *.inc "文件头
.org  $0000
```

```
    rjmpRESET        ;复位

;*************************************************************************
;*
;* 测试指令 ADD 功能,用置数方法输入数据,用单步或连续单步调试
;*
;*************************************************************************

.org $0010           ;跳过中断处
RESET:
LP:
                ;在寄存器窗口中点击 R16,修改 R16=0X11
                ;在寄存器窗口中点击 R17,修改 R17=0X33
    ADD R17,R16 ;R17=0X44 SREG=0X00,H=0,S=0,V=0,N=0,Z=0,C=0  , 高位低位均无进位
                ;在寄存器窗口中点击 R0,修改 R0=0X11
                ;在寄存器窗口中点击 R1,修改 R1=0X44
    ADD R1,R0   ;R1=0X55  SREG=0X00 ,            高位低位均无进位
                ;在寄存器窗口中修改 R20=0XC1
                ;在寄存器窗口中修改 R21=0XAA
    ADD R21,R20 ;R21=0X6B SREG=0X19,S=1,V=1,C=1  ,高位有进位,低位无进位
                ;在寄存器窗口中修改 R22=0X46
                ;在寄存器窗口中修改 R23=0X6A
    ADD R23,R22 ;R23=0XB0 SREG=0X2C,H=1,V=1,N=1   ,       高位无进位,低位有进位
                ;在寄存器窗口中修改 R24=0XFF
                ;在寄存器窗口中修改 R25=0XFF
    ADD R25,R24 ;R25=0XFE SREG=0X35,H=1,S=1,N=1,C=1  ,     高位有进位,低位有进位
    RJMPLP   ;调试时打开 Registers(寄存器窗口),
                    ;Processor(处理器窗口--程序,堆栈,状态寄存器,X/Y/Z 等项),
                ;New Memory View(存储器窗口--数据,I/O,E2PROM,程序存储器窗口,
                    ;该程序仅需打开片内 SRAM 数据窗口)
```

编辑源程序文件注意事项:

程序编译出错,有程序错误定位提示,指示错误原因及错误行号,只需鼠标点击错误提示,光标自动转到源程序的错误行,该行变为红色,以示注意,请修改。

注意指令中的 Rd,Rr,K,k,P,SRAM 等的数据选择范围;



指令中小写 d 为 16≤d≤31 的指令有:SUBI/SBCI/CPI/ANDI/CBR/ORI/SBR/SER/LDI;
其余指令中 Rd,Rr 的 R 为 0≤d≤31 ， 0≤r≤31;

指令中大写 K 为 0≤K≤63 的指令有:ADIW/SBIW;
指令中小写 k 为-64≤k≤+63 的指令有:BRBS/BRBC/BREQ/BRNE/BRCS/BRCC/BRSHBRLO/BRMI
BRPL/BRGE/BRLT/BRHS/BRHC/BRTS/BRTCBRVS/BRVC/BRIE/BRID;
指令中小写 k 为 0≤k≤255 的指令有:SUBI/SBCI/CPI/ANDI/CBR/ORI/SBR/LDI;
指令中小写 k 为 -2K≤k≤2K 的指令有:RJMP/RCALL;
指令中小写 k 为 0≤k≤65535 的指令有:LDS/STS;
指令中小写 k 为 0≤k≤4M 的指令有:JMP/CALL;

指令中大写 P 为 0≤P≤31 的指令有:SBI/CBI;
指令中大写 P 为 0≤P≤63 的指令有:IN/OUT;
片内 SRAM 地址 ≥0X0060;

3.1.3 源文件说明:供用户学习 AVR 汇编语言编程时参考

1. avrled.asm 验证 SL-AVR 万用串行下载开发实验器及 AT90S1200 的 A 口、B 口 LED 灯亮灭程序,也同时验证实验器通讯接口连机正常否,avrled2.asm 和 avrled3.asm 仅延时常数不同,所以 LED 闪动快慢不同;
2. AVRSTEP.ASM 用模拟调试单步验证 AT90S1200 B 口、D 口的输出状态;
3. DIP40LED.ASM 验证 SL-AVR 万用串行下载开发实验器及具有 DIP40 封装的 AT90S4414/AT90S8515/AT90S8535 等器件的 A 口、B 口、C 口、D 口 LED 灯亮灭程序,可修改延时常数;
4. AVR100.ASM (访问 E2PROM);
5. AVR102.ASM (数据块传送);
6. AVR108.ASM 加载程序存储器;
7. AVR128.ASM 安装和应用相同比较仪;
8. AVR200.ASM (乘法和除法应用一);
9. AVR200B.ASM (乘法和除法应用二);
10. AVR202.ASM (16 位运算);
11. AVR204.ASM (BCD 运算);
12. AVR220.ASM (冒泡分类算法) ;
13. AVR222.ASM (8 点平均滤波);
14. AVR235.ASM (CRC 程序存储的检查);
15. AVR240.ASM (4X4 键区休眠触发方式);
16. AVR242.ASM (多工法驱动和键区扫描);
17. AVR300.ASM (I2C 总线);
18. AVR302.ASM (I2C 工作);
19. AVR304.ASM (半双工中断方式 UART 应用一);
20. AVR305.ASM (半双工中断方式 UART 应用二);
21. AVR320.ASM (SPI 软件);
22. AVR400.ASM (设置和使用模拟比较器);
23. AVR401.ASM (8 位精度 A/D 转换器);

3.1.4 AVR 汇编器

AVR 汇编器覆盖了 AT90S 微控制器家族的全部范围。汇编器用于把汇编代码编译成目标代码，生成的目标代码可以用于模拟仿真器的输入或 ATMEL AVR 在线仿真器的输入。汇编器还能产生能够直接写入程序存储器和 E2PROM 存储的 PROM(可编程只读存储器)代码和一个任意 E2PROM 文件。

汇编器产生固定的代码分配，因此没有链接的必要。

汇编器可在 Miccrosoft Windows 3.11、Microsoft Windows 95/98/2000 和 Microsoft Windows NT 下运行。另外，还有一个 MS—DOS 版本。Windows 版本包括一个在线帮助功能，覆盖了所有这些说明。

3.1.4.1 编译器快速启动

AVR 编译器和所有配的程序文件都正确地安装在你的计算机上，请参考安装指令。

一、开始

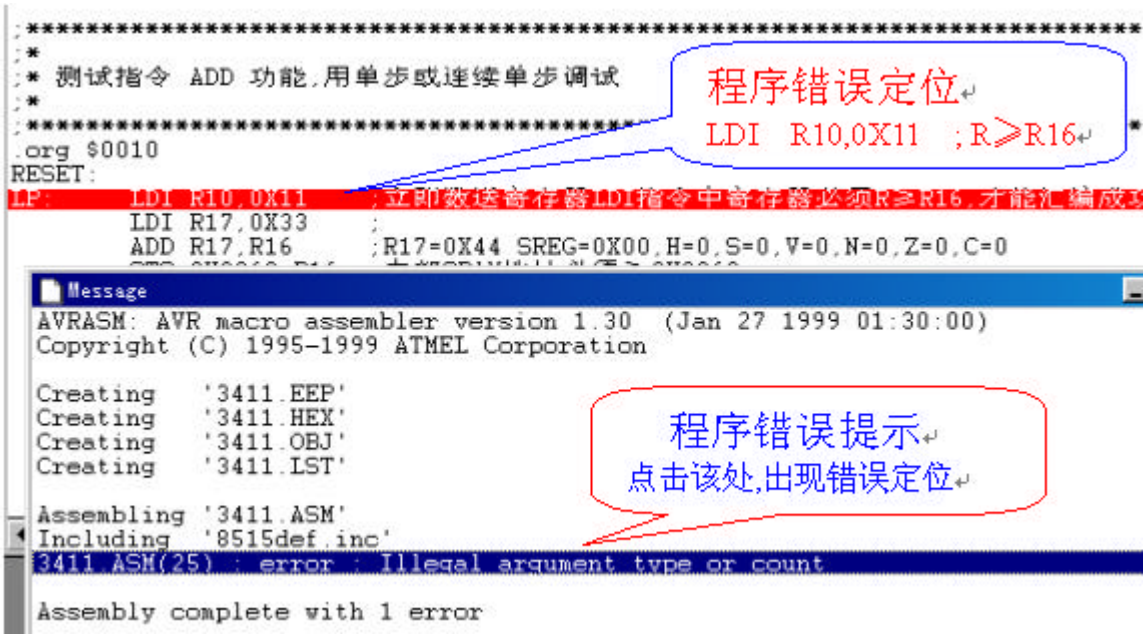
开始 AVR 汇编器。通过从某单中选择“File→Open”或按下工具条中的图标，打开文件“4411.asm”。这样就把汇编文件装入了编辑窗口，读读程序头，看看程序，但是不要改动它。

二、编译第一个文件

一旦看完了这个程序，从某单中选择编译，第二个窗口（信息窗口）就会出现，并包括一些错误信息。这个窗口将会覆盖编辑窗口，所以在屏幕上应先清除工作空间。选择包含程序代码的编辑窗口，并从菜单中选择“Window→Tile Horizontal”。让编辑窗口比信息窗口大一些是比较好的，所以让信息窗口向下移动一点，并让它挨着编辑窗口的底部，屏幕上将会出现如图 3.22 所示的内容。

三、寻找和纠正错误

从信息窗口来看，好像正尽力编译一个有很多缺陷的程序，为了进行下一步，错误必须被发现和纠正。指向信息窗口中的第一个错误，按下鼠标左键。注意，在编辑窗口，一个红色的横条覆盖在行上。错误信息指出 R 必须等于大于 R16。 修改后再编译就 OK。



四、 重新编译

为查明是否所有的错误都已改正，双击任意一个错误（为了激活编辑窗口）或在再次编译之前，单击编辑窗口。如果到现在你都这样做了，信息窗口会告诉你已经顺利完成了。

3.1.4.2 Microsoft 窗口特性

本节描述 WAVRASM 的特性，仅描述汇编器菜单项的特性。这是假设用户已经对 Ser一ach 和 Windows 菜单项比较熟悉。一个汇编器编辑的典型例子如图 3.24 所示。

一、打开一个汇编文件

可以在 WAVRASM 中打开一个新的或已经存在的文件。理论上来说一次打开多少个文件是没有限制的，但是 MS一Windows 有一个限制，就是每个文件的尺寸必须限制在 28K 以下。编辑比这大的汇编文件也是可能的，但是它们不能在一个完整的编辑器中编辑。每打开一个汇编文件，都将产生一个新的编辑窗口。

单击工具条中的按钮或从菜单中选择“File→New”（Ctrl + N）以创建一个新的汇编文件。单击工具条中的按钮或从菜单中选择“ File→Open”（Ctrl + O）以打开一个已经存在的汇编文件。

二、完整的编辑器

当 WAVRASM 装入了一个文件，文本编辑器将被激活。一旦一个文件被装入汇编器的编辑窗口，插入点就出现在窗口的左上角。▪

三、输入和格式文本

当输入时，插入点向右移动。如果文本输入超过右边边界，文本将自动向左滚动以使插入点可见。

四、移动插入点

只要把鼠标光标移动到想放插入点的地方并按下左键，插入点就可以移动到任何地方。用键或键的组合移动插入点，见表 3.2。

移动插入点	按键	移动插入点	按键
向文本的右边移动	右方向键	到一行文本的起点	Homn 键
向文本的左边移动	左方向键	到一行文本的结尾	End 键
向文本的上边移动	上方向键	到文本的起点	Ctrl+Home 键
向文本的下边移动	下方向键	到文本的结尾	Ctrl+End 键

图 3.2 键盘移动插入法

五、格式文本

表 3.3 中描述的键是为了得到一定文本形式的必要操作。

表 3.3 用键格式文本

操作	按键	操作	按键
插入一个空格	Spacebar	结束行	Enter
向右删除一个字符	Del	缩排一行	Tab
向左删除一个字符	Backspace	插入一个制表停止位	Tab

为了分开一行，可以把插入点移动到要断开的位置，然后按下 Enter 键。为了连接两行，

可以把插入点移动到要移动行的开始位置，然后按下 Backspace 键，编辑器就会把这一行连接到前一行上。

六、滚动

如果文本的一行要比上一次能够显示的长或者宽，这个文件可以通过滚动条来移动。

七、编辑文本

编辑菜单中包含一些功能，能够对编辑工作提供很多帮助。文本可以被删除、移动或复制到新的位置。Undo 命令可以用于取消上一次的编辑操作。文本与别的窗口或应用程序之间的文本传输可以通过剪贴板实现。当文本通 Cut 或 Copy 命令删除或复制，这些文本就放置在剪贴板中。粘贴命令把文本从剪贴板复制到编辑器中。

八、选择文本

从 Edit 菜单中选择一个命令去编辑文本之前，被操作的文本必须首先被选择。

用键盘来选择文本：

(1) 用方向键把插入点移动到要选择文本的起始部位。

(2) 按住 Shift 键，直到把插入点移动到要选择文本的结束部位，释放 Shift 键。取消这次选择，按任一个方向键。

用鼠标来选择文本：

(1) 把鼠标光标移动到要选择文本的起始部位；

(2) 按住鼠标左键，直到把插入点移动到要选择文本的结束部位，释放鼠标键。

（3）取消这次选择，按下鼠标左键或任一个方向键。

九、替换文本

当文本被选择时，可以通过输入新的文本立即替换被选择的文本。当第一个新的字符被输入时，所选择的文本就被删除了。

要替换文本：（1）选择要被替换的文本；（2）输入新的文本。

要删除文本：（1）选择要被删除的文本；（2）按下 Del 键。

要恢复被删除的文本。在删除文本后，应立即单击工具条中的　按钮或从菜单中选择“Edit→Undo”（Alt＋ Backspace）。

十、移动文本

要想在编辑器中移动文本，可以通过 Cut 命令把要移动的文本复制到剪贴板中，然后用 Paste 命令把它粘贴到新的位置。

要移动文本：

(1) 选择要移动的文本；

(2) 在工具条中按下　钮或从某单中选择“Edit→Cut”（Shift ＋ Del），文本就被放置在剪贴板中；

(3) 把插入点移动到新的位置；

（4）在工具条中按下　按钮或从菜单中选择“ Edit→Paste” （Shift ＋ Ins）复制文本。

如果一些文本要用到一次以上，不必每次都重新输入它。文本可以用 Copy 命令复制到剪贴板中，然后用 Paste 命令把它粘贴到其它地方。

要复制文本：

(1)选择要复制的文本；

（2）在工具条中按下　按钮或从菜单中选择“ Edit→Copy”（ Ctrl＋ Ins），文本就被放置在剪贴板中；

（3）把插入点移动到要放置文本的位置；

（4）在工具条中按下　按钮或从某单中选择“Edit→Paste”（ Shift＋　Ins）复制文本。

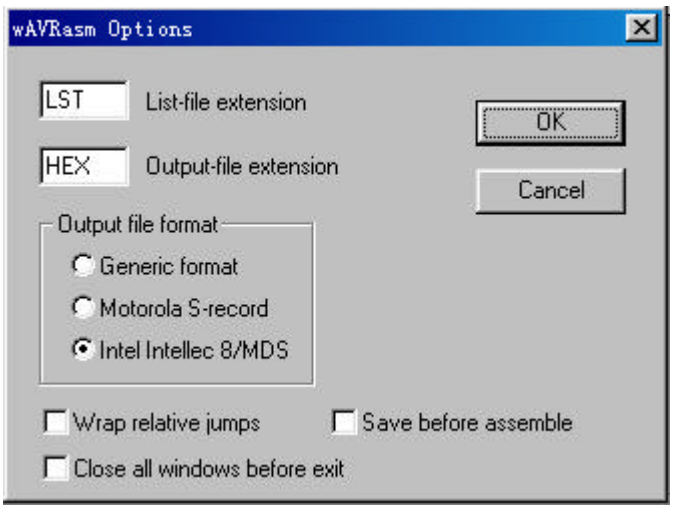
十一、取消一次编辑操作

Undo 命令可以用于取消上一次的编辑操作。例如，文本可能被意外地删除或复制到一个错误的位置。如果在错误发生后立即选择 Undo 命令，文本将被恢复到错误发生以前的状态。为了取消上一次的编辑操作，在工具条中按下　按钮或从某单中选择“Edit→Und。”（Alt＋Backspace）。

十二、单击错误信息

汇编器有一个单击错误信息的功能。当编译完一个程序时，一个信息窗口将出现在屏幕上。如果有错误出现，这些错误将排列在信息窗口中。如果信息窗口中的一个错误信息被单击，相应的源代码行就变成了红色。如果错误信息出现在包含文件中，什么也不会发生。

如果信息窗口行被双击,包含错误信息的窗口将变成活动窗口,光标将放置在包含错误信息行的起始部位。如果包含错误信息的文件未被打开，这个文件将被自动打开。注意这个功能仅对编译过的文件有效。这就是说，如果源代码文件的行被增加或重新移动过，这个文件必须被重新编译，以得到正确的行数。



十三、设置程序选项

WAVRASM 的一些缺省值可以在选项菜单中被修改。如果在某单中选择 Option，图所示的对话框将被弹出。在标有“ List — file extension”的框中，缺省的列表文件扩展名被填写。在标有“Output —fileextension”的框中，缺省的输出文件扩展名“HEX”被填写。在标有“Output file format”的框中，输出文件的格式可以被选择。如果单击了 OK 按钮，这些值将在以后汇编器运行时出现。注意目标文件（用于模拟仿真）将不被这些选项影响。目标文件的扩展名总是 OBJ，格式也是相同的。如果在源代码中定义了 E2PROM 段，汇编器将产生一个以 EEP 为扩展名的文件。这个文件用于初始化 E2PROM 存储器的值。E2PROM 初始化文件的格式与被选择的输出文件的格式相同。 “Wrap relative jumps”选项告诉汇编器使用地址约束方式。这个特性仅用于有 4K 程序存储器的器件的编译。在那样的器件上用此选项，相应的跳转指令和调用指令将能到达所有的程序空间。

“Save before assemble”选项使汇编器在编译之前自动地保存编辑器中的内容。

十四、命令行方案

在 MS—DOS 命令行方案中，汇编器可以通过命令调用。

AVRASM［—m ｜—I｜—g］［—w］ input.asm output.lst output. rom

AVRASM 将从 input.asm 中读入源代码，产生列表文件 output.lst、output.asm 和目标文件 input.obj。目标文件将被 MS—Windows 模拟仿真器调用。

用户可以通过选项 -m（Motolora S-record）、-I(Intel Hex)、-g（Generic）中的一个选择所产生的输出文件的格式。缺省值是 Generic 文件格式。

—w 选项告诉汇编器使用地址约束方式。这个特性仅用于有 4K 程序存储器的器件的编译。在那些器件上用此选项，相应的跳转指令和调用指令将能到达所有的程序空间。