CodeVisionAVR

VERSION 1.25.7

用户手册

翻译：谢剑波

2008 年 5 月 20 日

声明：

本手册是根据 CodeVisionAVR V1.25.3 User Manual 英文原文翻译而成，最初目的是用于本人和本人的学生学习之用，现将其整理成册，免费提供给各位初学者学习使用。

本手册可免费使用，但切务用于商业用途，如果因此而导致版权纠纷，责任自负。

由于本人英文水平有限，加之本从也正在学习 AVR，所以其中难免有错，望各位谅解并不吝赐教，本人将不胜感激。

本人联系方法：

e-mail：coolbor@163.com

QQ：947988474（很少登录）

Blog：http://blog.ednchina.com/coolbor/

谢剑波

2008 年 5 月 20 日

目 录

第 1 章 概述

CodeVisionAVR 是一款专为 Atmel AVR 系列微控制器而设计的交互式 C 编译器、集成开发环境（IDE）和自动程序生成器（APG）。

该软件可在 Windows98、Me、NT4、2000、XP 和 Vista 32 位操作系统下运行。

交互式 C 编译器几乎完全贯彻了 ANSI C 语言的标准，为了更好地支持 AVR 结构，添加了一些特性以充分发挥 AVR 结构和嵌入式系统需要的优势。

CVAVR 编译生成的“COFF”(一种通用的对象文件格式(Common Object File Format))目标文件支持 C 源代码级的调试，例如变量观察；同时“COFF”也能在 AVR 的官方调试仿真工具“Atmel AVR Studio debugger ”中进行仿真调试。

集成开发环境（IDE）内建了 AVR 芯片在系统编程器（Chip In-System Programmer）软件，这样在成功编译/汇编后能自动将程序传输到微控制芯片。

为了通过串行通信来调试嵌入式系统，IDE 提供一个内置终端（Terminal）。

除了标准的 C 语言函数库外，CVAVR 还提供了一些专用的库，例如：

- Alphanumeric LED modules
- Philips I2C bus
- National Semiconductor LM75 Temperature Sensor
- Philips PCF8563, PCF8583, Maxim/Dallas Semiconductor DS 1302 and DS1307 Real Time Clocks
- Maxim/Dallas Semiconductor 1 Wire protocol
- Maxim/Dallas Semiconductor DS1820, DS18S20 and DS18B20 Temperature Sensors
- Maxim/Dallas Semiconductor DS1621 Thermometer/Thermostat
- Maxim/Dallas Semiconductor DS2430 and DS2433 EEPROMs
- SPI
- Delays
- Gray code conversion

CodeVisionAVR 还包含了 CodeWizardAVR 自动程序生成器（APG），这样只需几分钟即可写出可执行以下功能的所有需要的代码：

- External memory access setup
- Chip reset source identification
- Input/Output Port initialization
- External Interrupts initialization
- Timers/Counters initialization
- Watchdog Timer initialization
- UART(USART) initialization and interrupt driven buffered serial communication
- Analog Comparator initialization
- ADC initialization
- SPI Interface initialization
- Two Wire Interface initialization
- CAN Interface initialization
- I^2C Bus, LM75 Temperature Sensor, DS1621 Thermometer/Thermostat and PCF8563, PCF8583,DS1302, DS1307 Real Time Clocks initialization
- 1 Wire Bus and DS1820/ Ds18S20 Temperature Sensors initialization
- LCD module initialization.

第 2 章 CodeVisionAVR 集成开发环境（IDE）

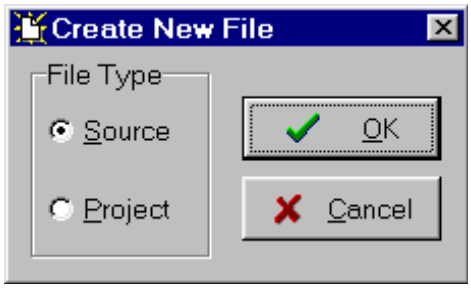
2.1 文件操作

使用 CodeVisionAVR IDE，可以查看和编辑任何已存在的或由 C 编译器或汇编器生成的文本文件。

2.1.1 创建一个新文件

要创建一个新的源文件，可使用 **File|New** 菜单命令或点击工具栏上的 **Create new file** 按键。

一个对话框出现，在其中必须选择 **File Type|Source**，然后点击 **Ok** 按钮。



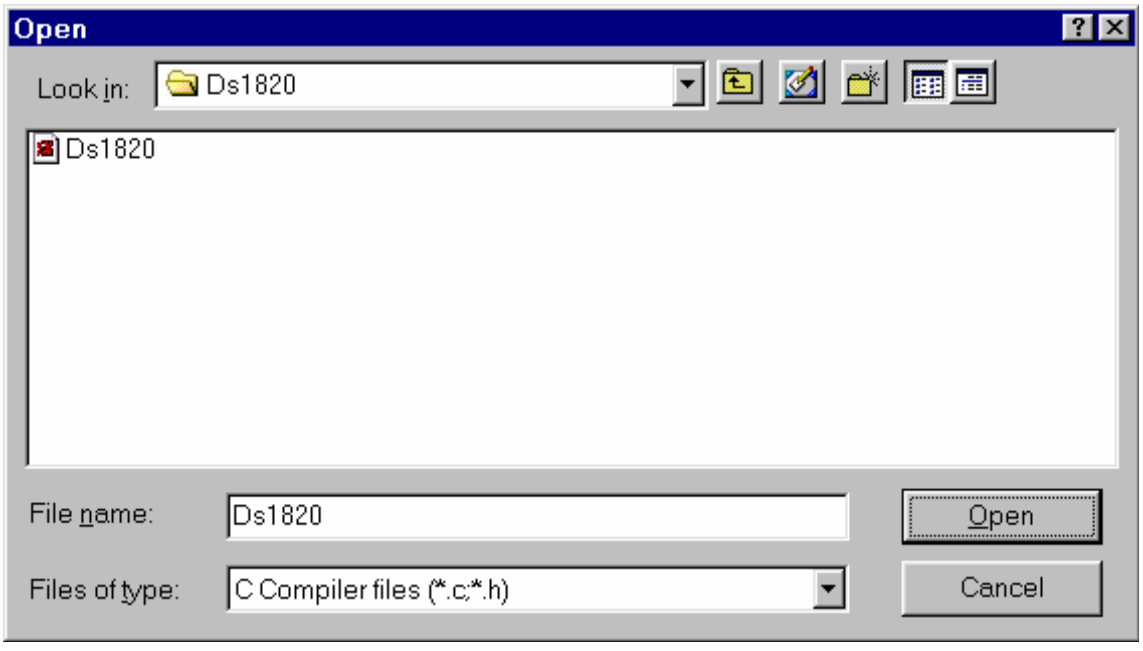
对新创建的文件，将出现一个新的编辑窗口。

新文件名为 **untitled.c**。可以使用 **File|Save** 菜单命令来将这个文件以新文件名保存。

2.1.2 操作一个已经存在的文件

使用 **File|Open** 菜单命令或点击工具栏上的 **Open file** 按钮新开一个已经存在的文件。

Open 对话框窗口出现。



必须选择要打开的文件名和文件类型。

点击 **Open** 按钮即可在新的编辑窗口打开文件。

2.1.3 文件历史

CodeVisionAVR IDE 保存着已经打开的文件的历史。

使用 **File|Reopen** 菜单命令可重新打开最近 8 个文件。

2.1.4 编辑文件

在编辑窗口中使用 **Tab**、**Arrows**、**Backspace** 和 **Delete** 键可编辑以前打开的或新创建的文件。

按 **Home** 键将光标移动到当前文本行的开头。

按 **End** 键将光标移动到当前文本行的结尾。

按 **Ctrl+Home** 键将光标移动到文件的开头。

按 **Ctrl+End** 键将光标移动到文件的结尾。

拉动鼠标可选择部分文本的。

使用 **Edit|Copy** 菜单命令、或按 **Ctrl+C** 键、或点击工具栏中的 **Copy** 按钮将所选文本复制到粘贴板。

使用 **Edit|Cut** 菜单命令、或按 **Ctrl+X** 键、或点击工具栏中的 **Cut** 按钮将所选文本复制到粘贴板并将其从文件删除。

使用 **Edit|Paste** 菜单命令、或按 **Ctrl+V** 键、或点击工具栏中的 **Paste** 按钮可将以前保存在粘贴板中的文本复制到当前光标处。

点击编辑窗口左边缘可选择整行文本。

使用 **Edit|Delete** 菜单命令、或按 **Ctrl+Delete** 键可删除所选文本。

Edit|Print Selection 菜单命令允许打印所选文本。

拖拉鼠标可移动部分文本。

按 **Ctrl+Y** 键删除在当前位置做了标记的文本行。

使用 **Edit|Find** 、分别使用 **Edit|Replace** 菜单命令或按工具栏中的 **Find** 或分别单击 **Replace** 按钮，可查找、分别替换编辑文件中的部分文本。

Edit|Find Next、分别使用 **Edit|Find Previous** 功能，可用于查找下一个、个别以前发生的搜索文本。同样的功能可使用 **F3**、**Ctr+F3** 键完成。

使用 **Edit|Undo**、**Edit|Redo** 菜单命令或按 **Ctrl+Z**、**Shift+Ctrl+Z** 键、或点击工具栏中的 **Undo**、**Redo** 按钮可以取消、重复在编辑的文本中的修改。

使用 **Edit|Goto Line** 菜单命令或按 **Alt+G** 键可到达编辑文本中指定的行号。

使用 **Edit|Toggle Bookmark** 菜单命令或按 **Shift+Ctrl+0...9** 键可在光标所在的行插入或移除标签。

Edit|Jump to Bookmark 菜单命令或 **Ctrl+0...9** 键可将光标定位于相应标签所在文本行的起始处。

单击鼠标右键可打开一个弹出菜单，该菜单也给出以上所述功能的相应命令。

2.1.5 保存文件

当前编辑的文件可使用 **File|Save** 菜单命令、按 **Ctrl+S** 键或点击工具栏中的 **Save** 按钮保存。

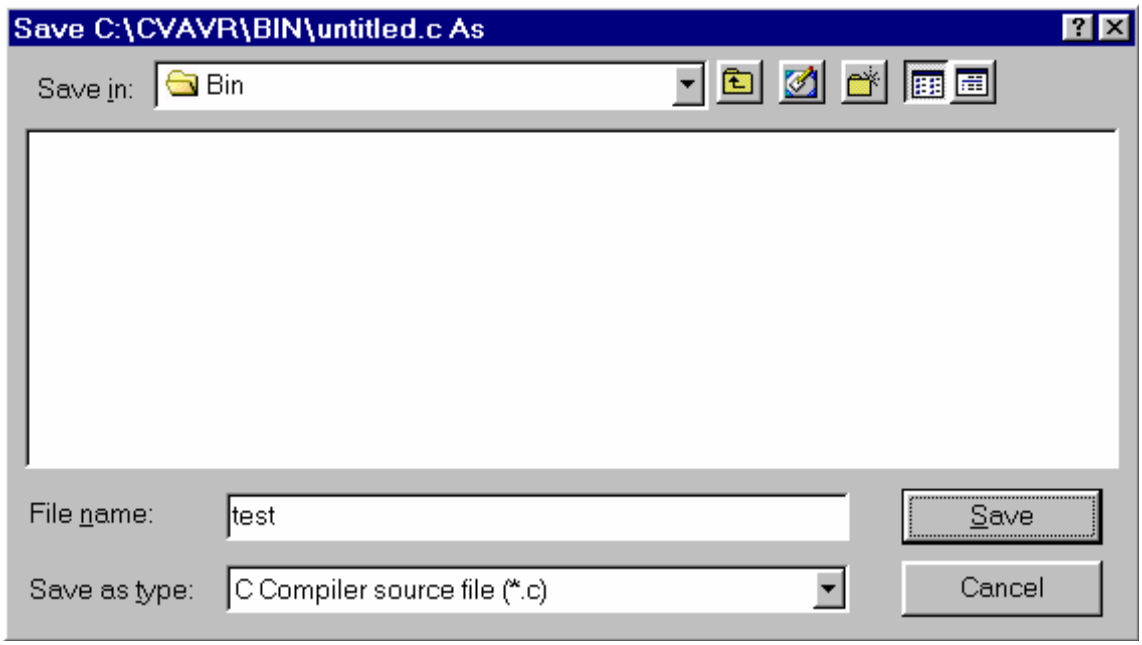
在保存时，编辑将创建一个备份文件，并将一个“~”符号附加在扩展名中。

使用 **File|Save All** 菜单命令可保存当前所有打开的文件。

2.1.6 重命名文件

使用 **File|Save As** 菜单命令可将当前编辑的文件以新文件名保存。

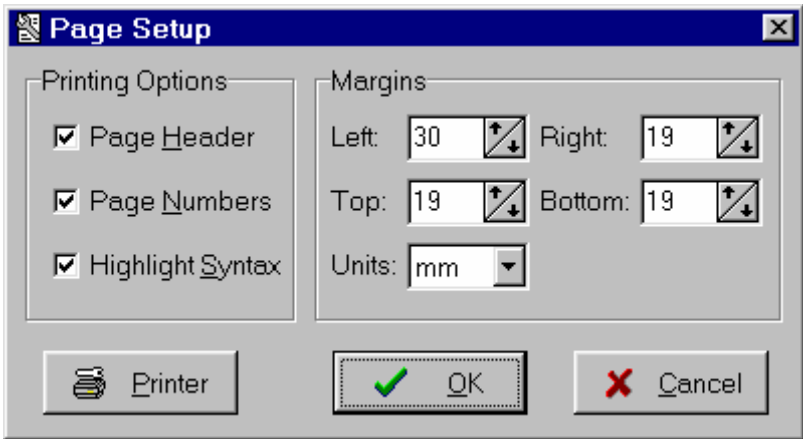
Save 对话框窗口打开。



指定新的文件名和文件类型，确定其位置。

2.1.7 打印文件

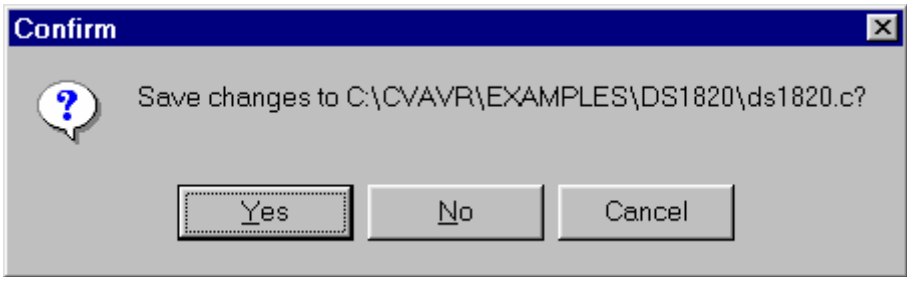
使用 **File|Print** 菜单命令或按工具栏中的 **Print** 按钮。
文件内容即可打印到 Windows 默认打印机。
打印时所用的页边距可使用 **File|Page Setup** 菜单命令来设置，该命令将打开 **Page Setup** 对话窗口。



设置页边距时在 **Units** 列表框中指定所使用的单位。
打印机的配置可点击该对话框窗口中的 **Printer** 按钮。
点击 **OK** 按钮可保存修改，点击 **Cancel** 按钮可取消。

2.1.8 关闭文件

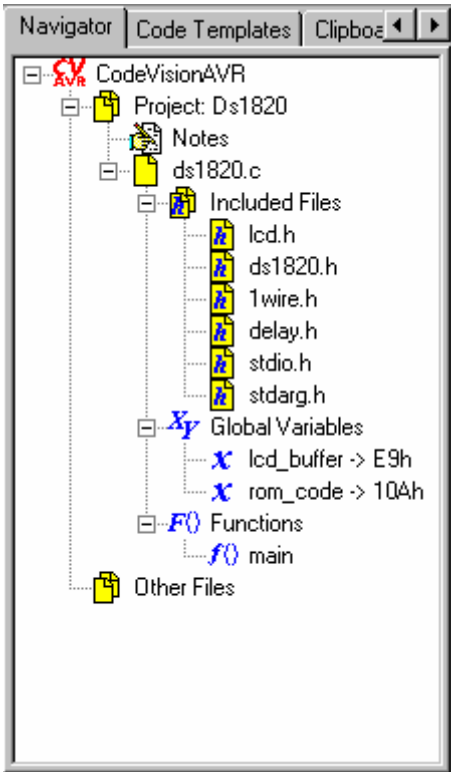
使用 **File|Close** 菜单命令可退出当前正在编辑的文件。
如果文件被修改，而还未保存，则有提示是否需要保存。



点击 **Yes** 保存修改并退出文件。
点击 **No** 不保存修改而退出文件。
点击 **Cancel** 禁止文件关闭进程。
所有当前打开的文件可使用 **File|Close All** 菜单命令来关闭。

2.1.9 使用导航器

导航器窗口使得源文件的显示或打开非常轻松。
点击文件名，相应的文件即可最大化或打开。



在执行了“**Compile**”或“**Make**”命令后，在导航窗口中会显示一个列表：（1）用#include 包含的文件（included files），（2）全局变量（global variables），（3）函数声明（functions），在每个对应的 C 源程序文件下拉显示。
点击变量、函数，则变量名、函数和声明就会在相应的 C 源文件中高亮显示。
如果在编译期间，出现错误或警告，也会在导航器窗口中显示。
点击错误或警告，相应的源代码行就会在相应的文件中高亮显示。
在导航器窗口中右击，可打开一个弹出菜单，并进行以下选择：

- **Open**——打开文件；
- **Save**——保存当前编辑的文件；
- **Save All**——保存所有打开的文件；
- **Close Current File**——关闭当前文件；
- **Close Project**——关闭项目；

- **Close All**——关闭所有打开的文件。

2.1.10 使用代码模板

代码模板（**Code Templates**）窗口使得添加常用代码序列到当前编辑的文件中非常容易。

在代码模板窗口中点击需要的代码序列，然后拖拉到编辑窗口中的适当位置。

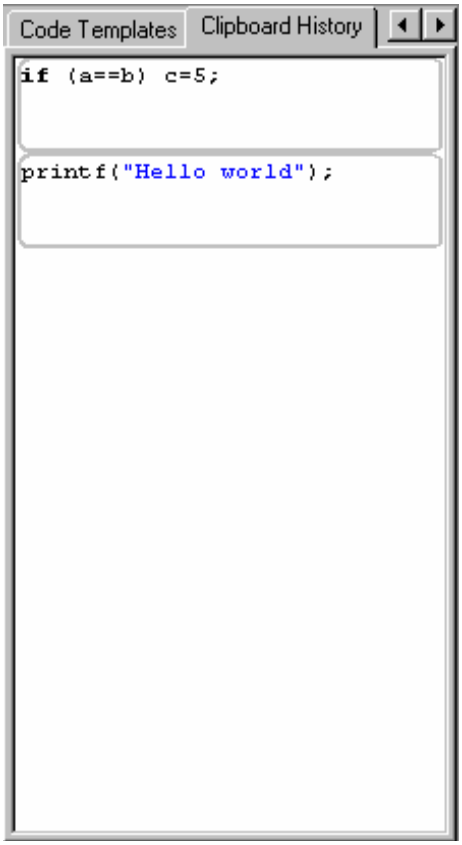
要添加新的代码模板到列表中，只要将一个文本选择从编辑窗口拖拉到代码模板窗口即可。

在代码模板窗口中右击可打开一个弹出菜单，可做以下选择：

- **Copy to the Edit Window**——将当前选择的代码模板复制到编辑窗口；
- **Paste**——将一个文本片段从粘贴板粘贴到代码模板窗口；
- **Move Up**——在列表中将当前所选的代码模板上移；
- **Move Down**——在列表中将当前所选的代码模板下移；
- **Delete**——从列表中将当前所选的代码模板删除。

2.1.11 使用粘贴板历史

粘贴板历史（**Clipboard History**）窗口允许查看或访问当前复制到粘贴板中的文本片段。



在粘贴板历史窗口中右击，可打开一个弹出菜单，并做以下选择：

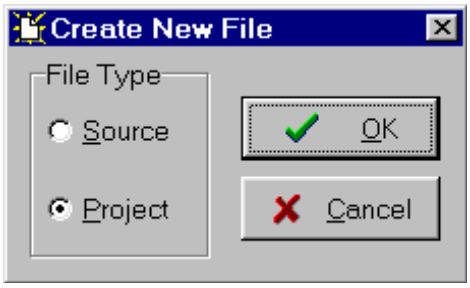
- **Copy to the Edit Window**——将当前所选文本片段从粘贴板历史窗口复制到编辑窗口；
- **Delete**——将当前所选文本片段从列表中删除；
- **Delete All**——从列表中删除所有文本片段。

2.2 项目操作

项目将源文件和编译器设置组合起来，以便创建特定的程序。

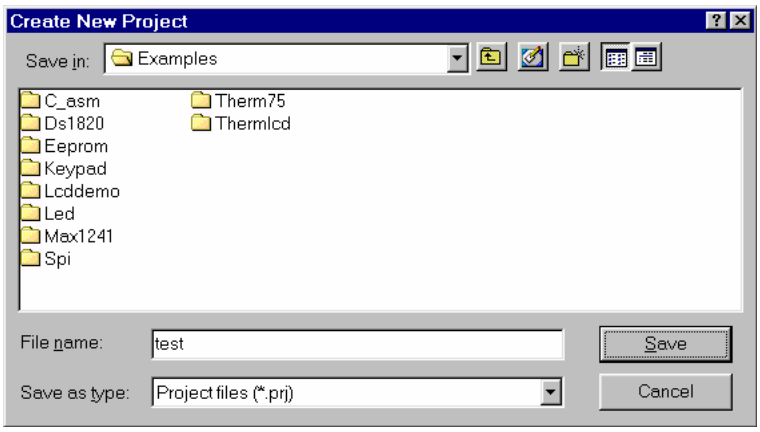
2.2.1 创建新项目

使用 **File|New** 菜单命令 或点击工具栏中的 **Create new file** 按钮可创建新项目。
一个对话框出现，在这个对话框中，必须选择 **File Type|Project** 并点击 **OK** 按钮。



一个对话框打开，提示确认是否使用 CodeWizardAVR 来创建新项目。

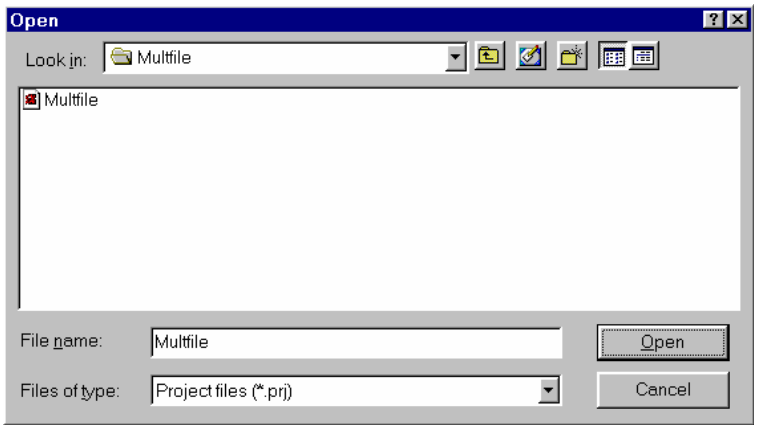
如果选择 **No** 则 **Create New Project** 将打开。
此时必须指定新项目文件名和位置。



项目名的扩展名为.prj。
此时必须使用 **Project|Configure** 菜单命令配置项目。

2.2.2 打开已经存在的项目

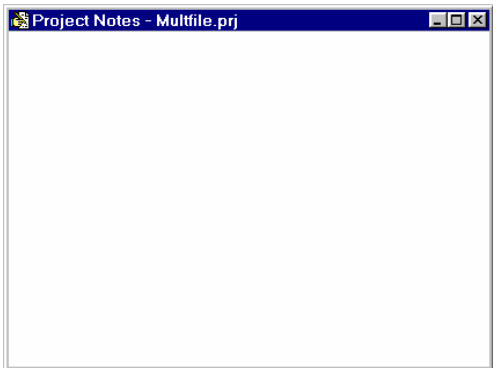
使用 **File|Open** 菜单命令或点击工具栏中的 **Open file** 按钮可打开一个已经存在的项目。
Open 对话框出现。



必须选择要打开的项目文件名。
点击 **Open** 按钮打开项目文件和相应的源文件。
使用 **Project|Configure** 菜单命令配置项目。

2.2.3 添加记录或命令到项目

在每个项目中，CodeVisionAVR IDE 都将创建一个放置记录和命令的文本文件。
使用 **Project|Notes** 或 **Windows** 菜单命令可访问该文件。



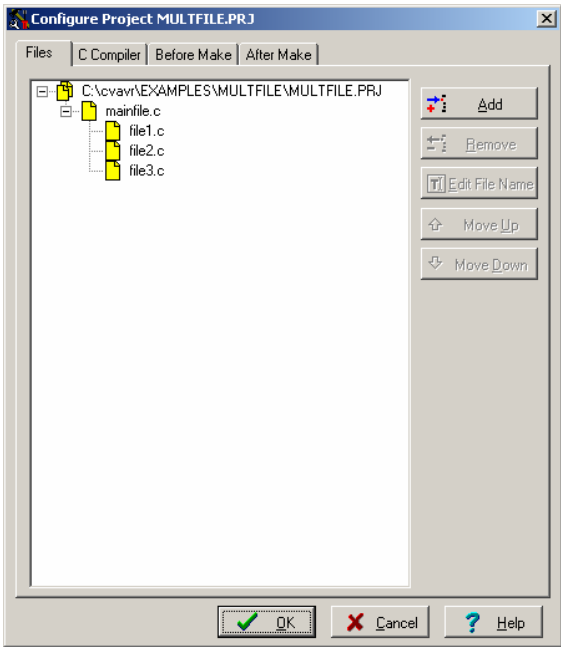
该文件可使用标准编辑器命令编辑。
在关闭项目或退出 CodeVisionAVR 程序时该文件将自动保存。

2.2.4 配置项目

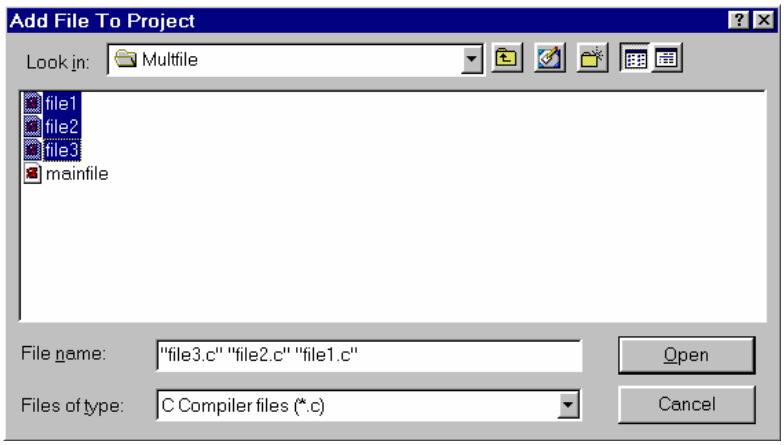
The Project can be configured using the **Project|Configure** menu command or the **Project Configure**
toolbar button.

2.2.4.1 从项目添加或移除文件

要从当前打开的项目中添加或移除文件，必须使用 **Project|Configure** 菜单命令。
Configure Project 对话框窗口打开，然后选择 **Files** 标签。



点击 **Add** 按钮添加源文件到项目中。
第一个添加到项目中的文件是主项目文件。
这个文件总是 **Make** -ed。
其余添加到项目中的文件将自动按下“**Make**”执行生成目标代码功时自动链接到主项目文件。
在 **Add File to Project** 对话框中选择时按住 Ctrl 键可添加多个文件。



在项目被打开后，所有项目文件都会在编辑器中打开。

单击文件，然后点击 **Remove** 按钮可从项目中删除该文件。

单击一个文件，并点击 **Move Up** 或 **Move Down** 按钮上移或下移，可改变项目的文件编译顺序。

点击 **OK** 或 **Cancel** 按钮可保存或不保存修改。

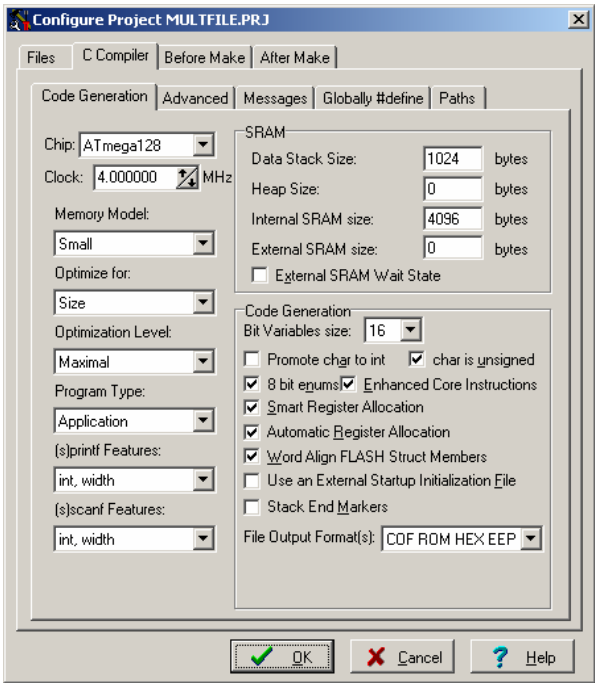
在创建有多个文件的项目时，必须遵守以下规则：

- 只有 C 文件必须添加到项目文件列表中；
- 不必从文件列表中包含 C 文件，因为这些会自动链接；
- 数据类型定义和函数声明必须放在.H 头文件中，这些头文件在 C 文件中是必要的；
- 全局变量声明必须放在必要的 C 文件中；
- 不必声明头.H 文件中的非静态全局变量，因为如果这些文件被包含超过一次，编译器就会给出有关变量声明的错误。

2.2.4.2 设置 C 编译器选项

要为当前打开的项目设置 C 编译器选项，必须使用 **Project|Configure** 菜单命令。

Configure Project 对话框窗口打开，此时必须选择 **C Compiler** 和 **Code Generation** 标签。



使用 **Chip** 下拉框可选择目标 AVR 微控制芯片型号。

另外还必须以 MHz 为单位指定 **CPU Clock Frequency**（CPU 时钟频率），这是延时函数、

1 线协议函数以及 Maxim/Dallas 半导体 DS1820/DS18S20 温度传感器函数所必须的。

使用 **Memory Model** 列表框可选择需要的存储模式。

使用 **Optimize for|Size** 和 **Optimize for|Speed**，编译程序可对最小容量和最快执行速度进行优化。

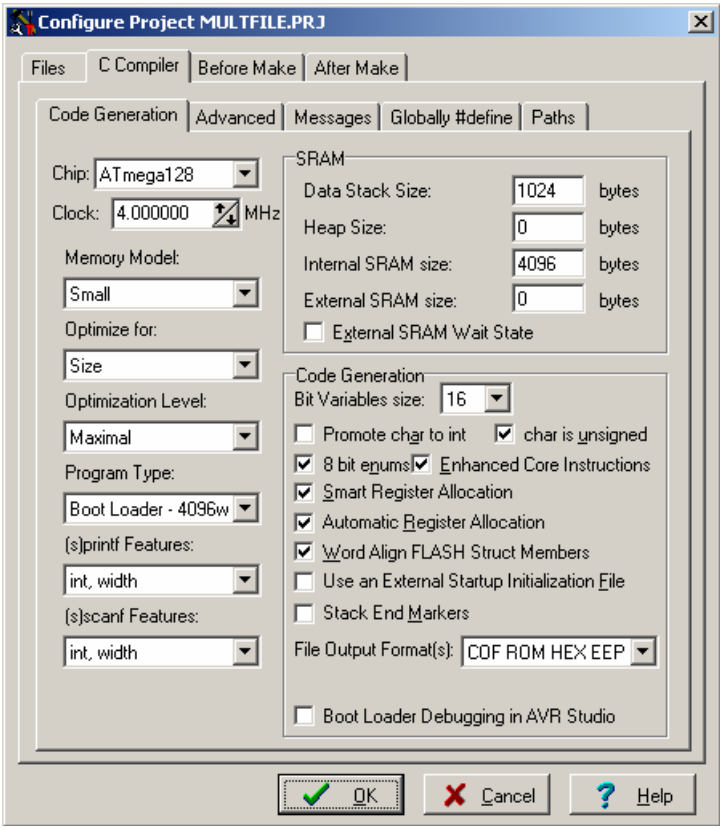
使用 **Optimization Level** 设置可指定代码优化的总量。

Maximal optimization level 可能会使得在 AVR Studio 中进行代码调试时有些困难。

对于具有自编程功能的器件，**Program Type**（程序类型）可选择以下类型：

- **Application**
- **Boot Loader**

如果选择 **Boot Loader** 程序类型，则附加的 **Boot Loader Debugging in AVR Studio** 选项也会生效。



如果选择这个选项，编译器就会生成附加代码，以支持 Boot Loader 作为源水平级在 AVR Studio simulator/emulator 中调试。

在使用最终的 Boot Loade 代码编程芯片时，Boot Loader Debugging 选项必须禁止。

(s)printf features 选项允许选择哪个版本的 printf and sprintf **Standard C Input/Oputput Functions** 链接到项目：

- **int**——支持以下类型字符的转换：'c'、's'、'p'、'i'、'd'、'u'、'x'、'X'、'%'，不支持指定宽度或精度，只支持'+'和' '标志，不支持输入大小的修改。
- **int, width**——支持以下类型字符的转换：'c'、's'、'p'、'i'、'd'、'u'、'x'、'X'、'%'，支持指定宽度，而不支持指定精度，只支持'+'和' '标志，不支持输入大小的修改。
- **long, width**——支持以下类型字符的转换：'c'、's'、'p'、'i'、'd'、'u'、'x'、'X'、'%'，支持指定宽度，而不支持指定精度，只支持'+'、'-'、'0'和' '标志和'l'输入大小的修改。
- **long, width, precision**——支持以下类型字符的转换：'c'、's'、'p'、'i'、'd'、'u'、'x'、'X'、'%'，支持指定宽度和精度，只支持'+'、'-'、'0'和' '标志和'l'输入大小的修改。
- **float, width, precision**——支持以下类型字符的转换：'c'、's'、'p'、'i'、'd'、'u'、'e'、'E'、

'f'、'x'、'X'、'%'，只支持'+'、'-'、'0'和' '标志和'l'输入大小的修改。
特性选择越多，为 printf and sprintf 函数生成的代码容量就越大。

(s)scanf features 选项允许选择哪个版本的 scanf and sscanf **Standard C Input/Oputput Functions** 链接到项目：

- **int, width**——支持以下类型字符的转换：'c'、's'、'i'、'd'、'u'、'x'、'%'，支持指定宽度，而不支持指定精度。
- **long, width**——支持以下类型字符的转换：'c'、's'、'i'、'd'、'u'、'x'、'%'，支持指定宽度，只支持'l'输入大小的修改。

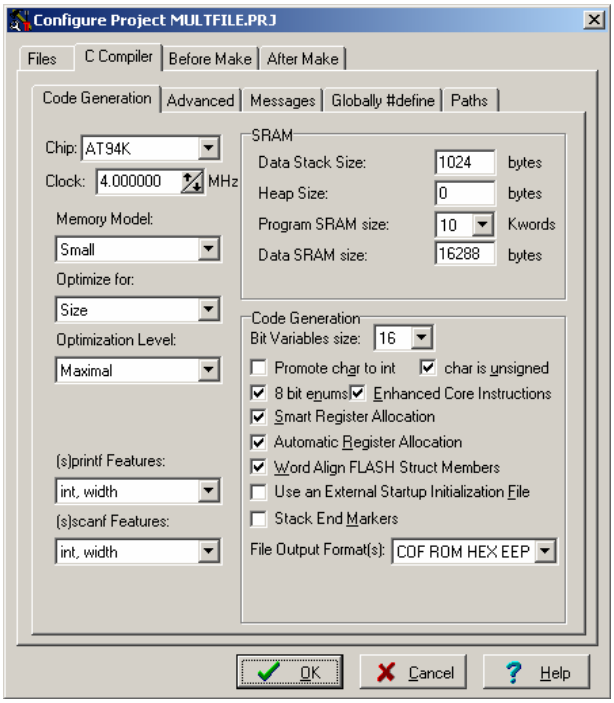
特性选择越多，为 scanf and sscanf 函数生成的代码容量就越大。

Data Stack Size 也必须指定。
如果使用了标准库中的动态存储分配函数，**Heap size** 也必须指定。
其大小可按下式计算：

$$heap\_size = (n+1)\cdot 4 + \sum_{i=1}^{n} block\_size_i$$

式中：
n —— 在堆中分配的存储块的数量
i block _ size —— 第 *i* 个存储块的大小
如果不使用存储分配函数，**Heap size** 必须为 0。
最后可能还要指定 **External SRAM Size**（如果微控制器连接有外部 SRAM 存储器的话。）
External SRAM Wait State 选项允许在访问外部 SRAM 期间插入等待状态。这在使用慢速度存储设备时是很有用的。

如果使用了 Atmel AT94K05, AT94K10, AT94K20 or AT94K40 FPSLIC 等器件，则有可能要以 K 字为单位指定 **Program SRAM size**。



放在 R2~R14 寄存器中的全局位变量的最大容量可使用 **Bit Variables size** 列表框指定。
选择 **Promote char to int** 复选框允许 **char** 操作数按 ANSI 标准升级为 **int**。
这个选项还可以使用**#pragma promotechar** 编译器指令来指定。

Char 到 **int** leads 的升级对于象 AVR 这样的 8 位单片微控制器会增加代码容量并降低运行速度。

如果选择 **char is unsigned** 复选框，编译器将把无符号 8 位数（0~255）当作默认的 **char** 数据类型。

如果没有选择该复选框，编译器将把有符号 8 位数（-128~+127）当作默认的 **char** 数据类型。

这个选项不冤枉路使用**#pragma uchar** 编译器指令来指定。

将 **char** 作为无符号数处理对代码容量和运行速度都是有好处的。

如果选择 **8 bit enums** 复选框，编译器将枚举（enumerations）作为 8 位 **char** 数据类型处理，这有助于优化代码容量和编程程序的执行速度。如果没有选择该复选框，枚举（enumerations）则根据需要被视作 ANSI 标准的 16 位 **int** 数据类型。

Enhanced Instructions 复选框允许或禁止新的 Atmega 和 AT94K FPSLIC 器件的增强核指令。

Smart Register Allocation 复选框允许 R2~R14（不用于位变量）和 R16~R21 的分配，按这种方法，16 位变量将更适合放在偶寄存器对中，这样增强核指令 MOVW 对它们的访问将是很有好的用法。这个选项只有在选择了 **Enhanced Instructions** 复选框时才有效。

如果 **Smart Register Allocation** 被禁止，寄存器将按变量声明的顺序分配。

如果程序是使用 CodeVisionAVR V1.25.3 版本开发的，**Smart Register Allocation** 选项应该被禁止，因为它包含了访问位于寄存器 R2~R14 和 R16~R21 中的汇编代码。

在选择了 **Automatic Register Allocation** 复选框后，寄存器 R2~R14（不用于位变量）可自动分配为 **char** 和 **int** 全局变量和全局指针变量。

选择 **Word Align FLASH Struct Members** 选项允许位于 FLASH 存储器中的指令数矫正为偶地址（字）。

该选项只对与 CodeVisionAVR V1.24.4a 版本中创建的项目的兼容时才出现。

对新项目，这个选项必须在不复选时才有利于减小代码容量。

选择 **Compilation|Use an External Startup File** 复选框可使用外部启动文件。

对于调试目的，还有 **Stack End Markers** 选项。如果选择，编译器将字符串 **DSTACKEND** 或 **HSTACKEND** 放到 **Data Stack** 或 **Hardware Stack** 区的最后。

在 AVR Studio 调试器中调试程序时，将可会看到这些字符串是否被覆盖和修改 **Data Stack Size**。

如果程序正确运行，可删除这些字符串，以减小代码容量。

使用 **File Output Format(s)**列表框，可为由编译器生成的文件选择以下格式：

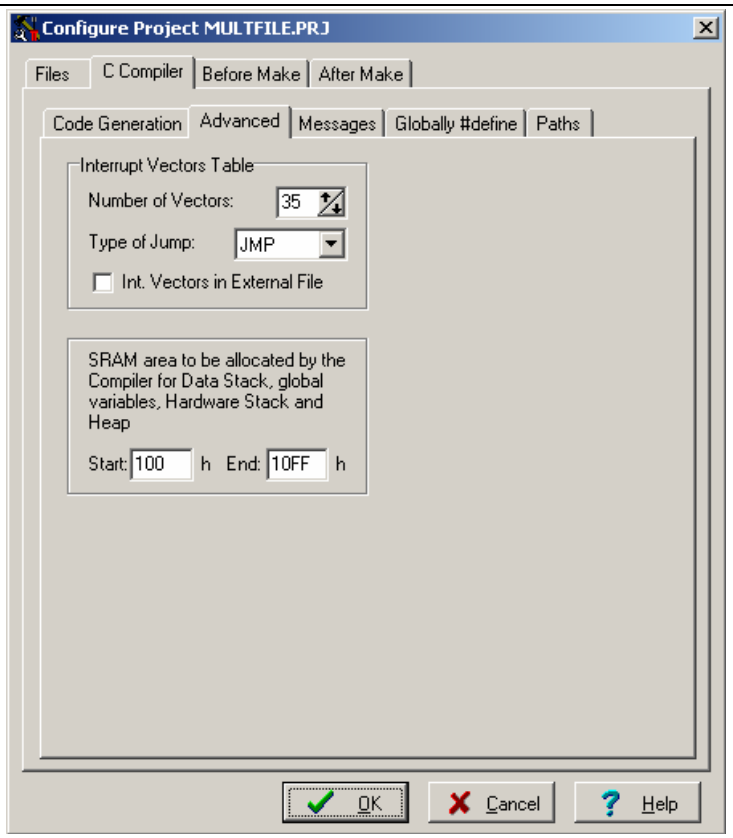
- COFF (Atmel AVR Studio 调试器的需要)、ROM、Intel HEX 和 EEP (In-System Programmer 的需要) ；
- Atmel generic OBJ、ROM、Intel HEX 和 EEP (In-System Programmer 的需要)。

如果选择 COFF 文件格式，并且选择 **Use the Terminal I/O in AVR Studio 3** 复选框，将生成特殊的调试信息以便对带有用于仿真 AVR 芯片串口的通讯使用 AVR Studio 3 Terminal I/O 窗口。

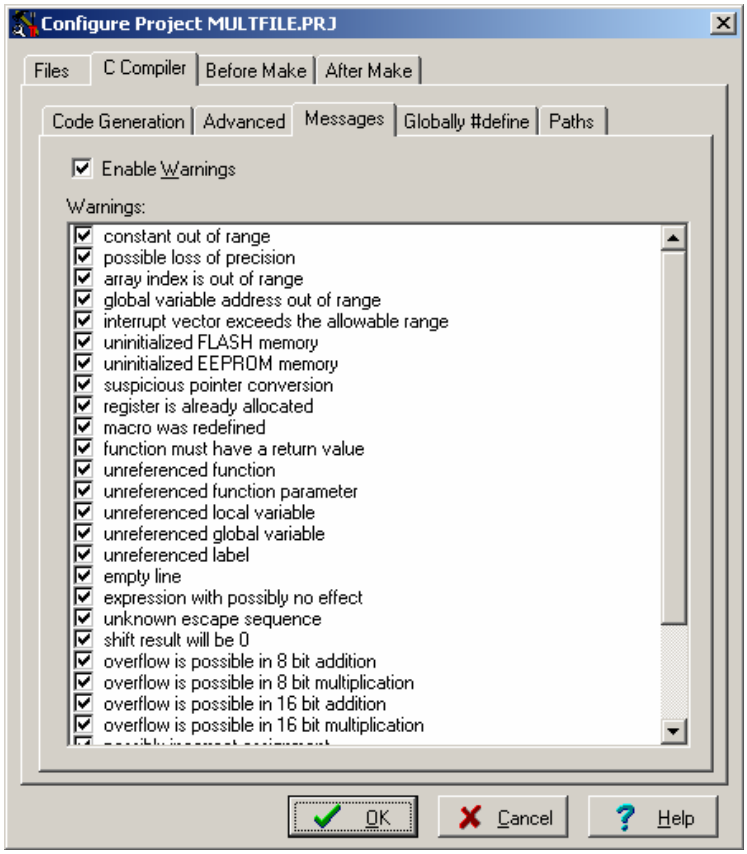
AVR Studio 4 不支持这个选项。

如果 **Use the Terminal I/O in AVR Studio 3** 选项被使能，UART 或 USART 代码在实际的 AVR 芯片中将不能正确运行。**这个选项仅用于调试目的。**

Advanced 标签只在 Professional 版的编译器中才有，该标签允许诸如中断微量和存储器使用的数量和跳转类型等更详细的自定义配置：



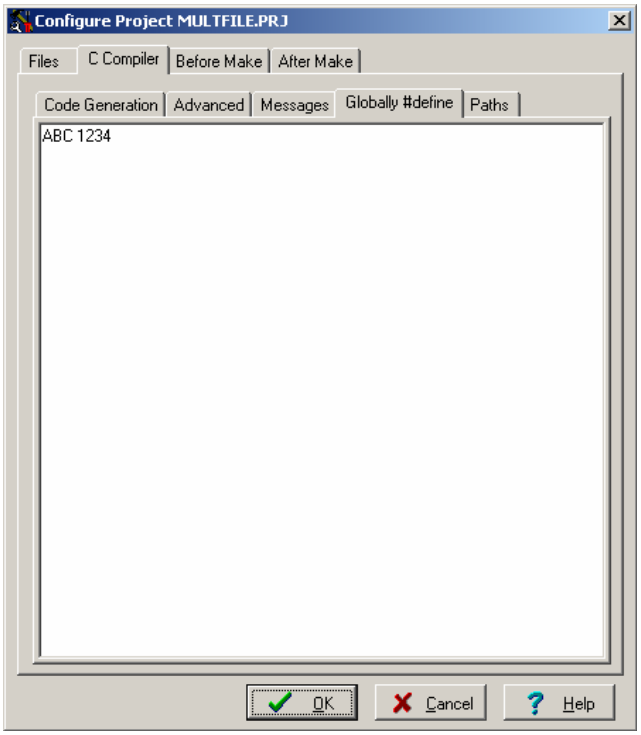
Messages 标签允许单独地允许或禁止各种编译器警告：



在编译时警告信息的生成可以使用 **Enable Warnings** 复选框全部允许或禁止。

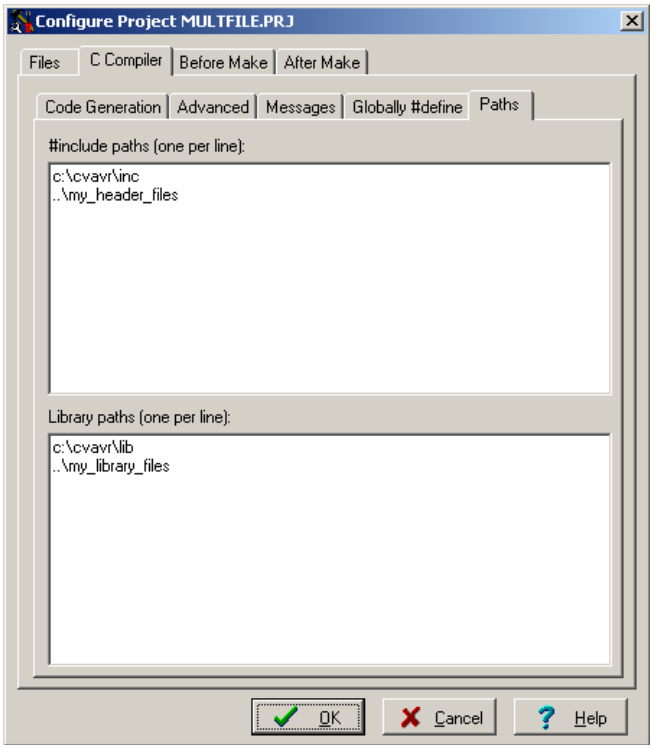
Globally #define 标签允许使用#define 的宏在所有的项目文件中都是可见的。

例如：



上图与在每个项目文件中做以下定义等价：
#define ABC 1234

Paths 标签允许指定#include 和库（library）文件的附加路径。
这些路径必须在相应的编辑控制区每行输入一个。

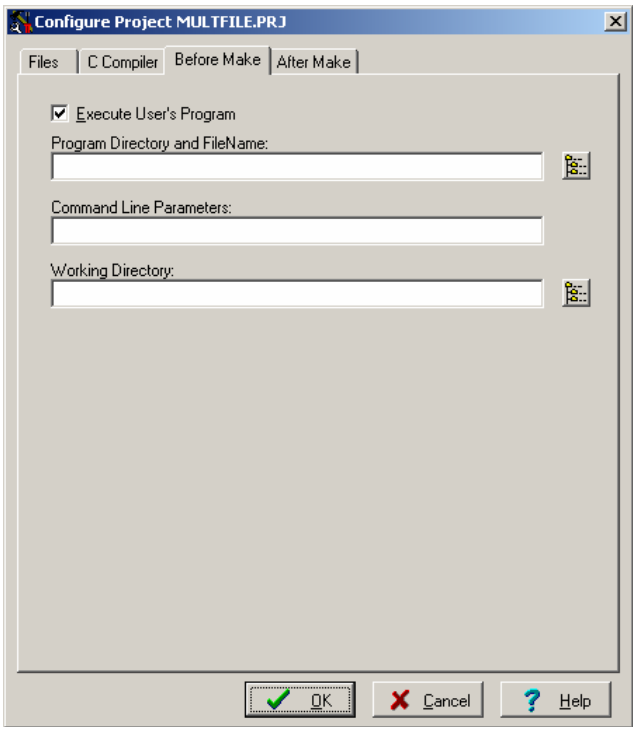


点击 **OK** 保存修改，或点击 **Cancel** 不保存修改。

2.2.4.3 在 Make 前执行用户指定的程序

如果选择项目配置窗口（Project Configure）中的 **Before Make** 标签，这个选项将可用。

如果选择 **Execute User's Program** 选项，那么先前指定的一个程序将在编译/汇编进程之前执行。

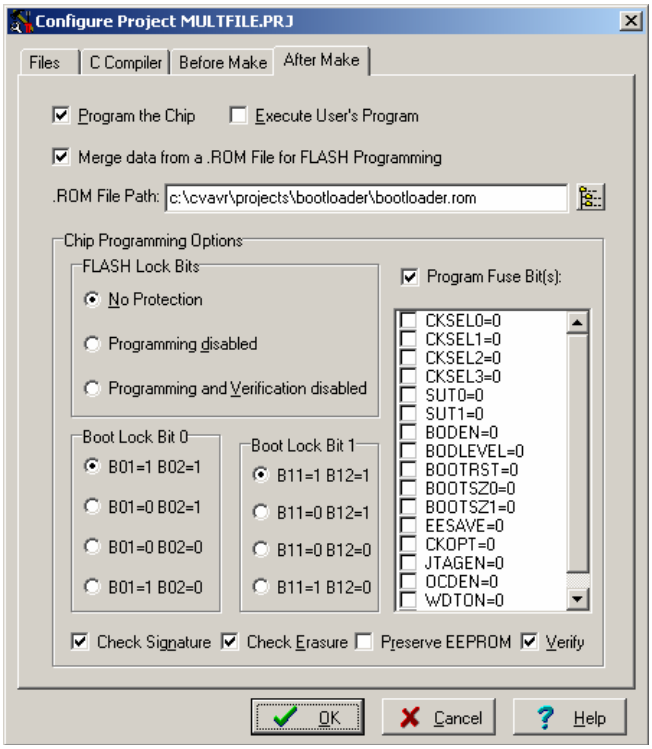


可为要执行的程序指定以下参数：

- Program Directory and File Name——程序目录和文件名；
- Program Command Line Parameters——程序命令行参数；
- Program Working Directory.——程序工程目录。

2.2.4.4 在 Make 后将已经编译的程序传送至 AVR 芯片

如果选择项目配置窗口（Project Configure）中的 **After Make** 标签，这个选项将可用。



如果复选 **Program the Chip** 选项，在成功编译/汇编后，程序将被内嵌编程软件自动传送

到 AVR 芯片。

以下步骤自动被执行：

- 芯片擦除；
- FLASH 和 EEPROM 空白检查；
- FLASH 编程和校验；
- EEPROM 编程和校验；
- 熔丝和锁定位编程。

如果复选 **Merge data from a .ROM File for FLASH Programming** 选项，将在 FLASH 编程缓冲区中合并.ROM 文件的内容，该.ROM 文件是在 Make 之后由编译器创建的，而其数据来自于在**.ROM File Path** 中指定的.ROM 文件。

这对被编程到 FLASH 存储器中的应用程序是很有用的，例如：在添加一个在其他项目中编译的可执行 boot loader 时。

使用 **Chip** 列表框可选择需要编程的芯片类型。

如果所选的芯片有可编程的熔丝位，则会出现一个附加的 **Program Fuse Bit(s)**复选框。

如果该复选框被复选，芯片的熔丝位将在 Make 后被编程。

熔丝位可设置在 Atmel data sheets 中所描述的各种芯片选项。

如果一个熔丝位复选框被复选，相应的熔丝位将被设置为 0，该熔丝位将被编程（根据每个 Atmel data sheets 中的给定）。

如果一个熔丝位复选框未被复选，相应的熔丝位将被设置为 1，该熔丝位将不编程。

如果要防止程序被复制，必须在 **FLASH Lock Bits** 区选择相应的选项。

如果要在编程前检查芯片的序列号，必须使用 **Check Signature** 选项。

如果要加速编程进程，可以取消复选 **Check Erasure** 复选框。

这样一来，FLASH 擦除的正确性将不被校验。

Preserve EEPROM 复选框允许在芯片擦除时保留 EEPROM 的内容。

要加速编程进程，可取消复选 **Verify** 复选框。

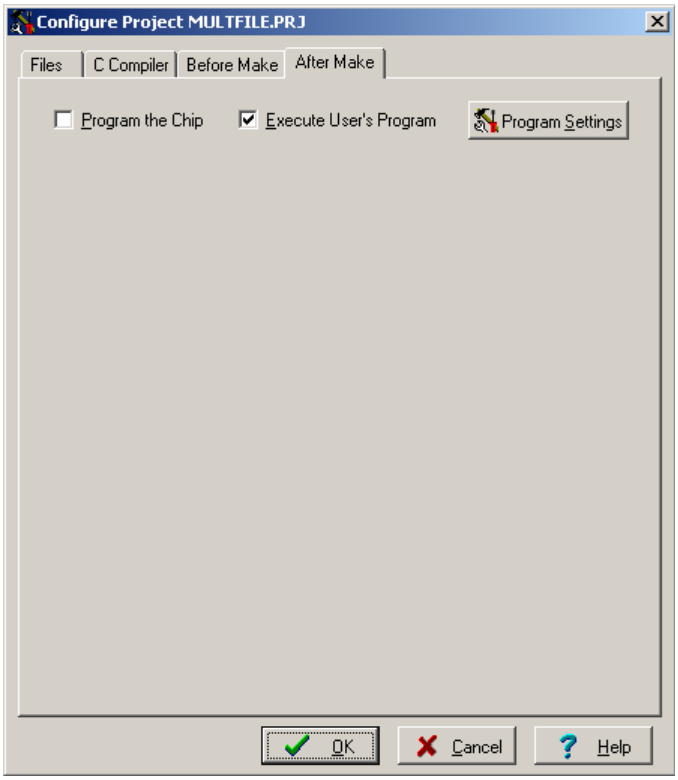
这样一来，FLASH 和 EEPROM 编程将不被校验。

点击 OK 保存修改，或点击 Cancel 不保存。

2.2.4.5 在 Make 之后执行用户指定的程序

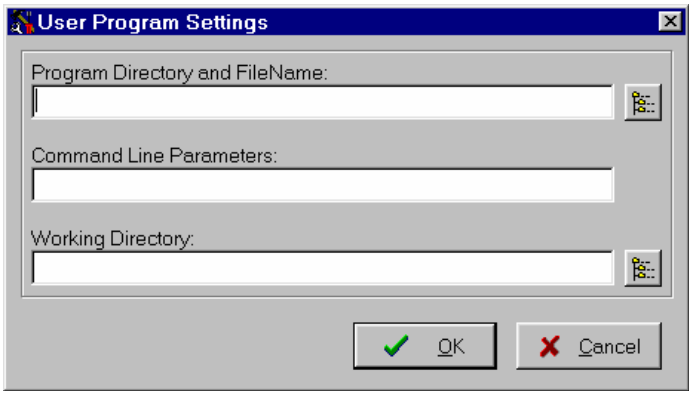
如果选择项目配置窗口（Project Configure）中的 **After Make** 标签，这个选项将可用。

如果复选 **Execute User's Program** 选项，一个在先前指定的程序将在编译/汇编进程之后被执行。



使用 **Program Settings** 可修改以下内容：

- 程序目录和文件名；
- 程序命令行参数；
- 程序工作目录。



点击 OK 保存修改，或点击 Cancel 不保存。

2.2.5 生成一个可执行的程序

生成一个可执行的程序需要按以下步骤进行：

1、使用 CodeVisionAVR C 编译器编译项目 C 源文件，并生成一个汇编源文件。

2、使用 Atmel AVR 汇编程序 AVRASM32 汇编汇编源文件

Compiling——执行步骤 1。

Making——执行步骤 1 和 2。

对于大项目，**Compiling** 将占用大量时间。

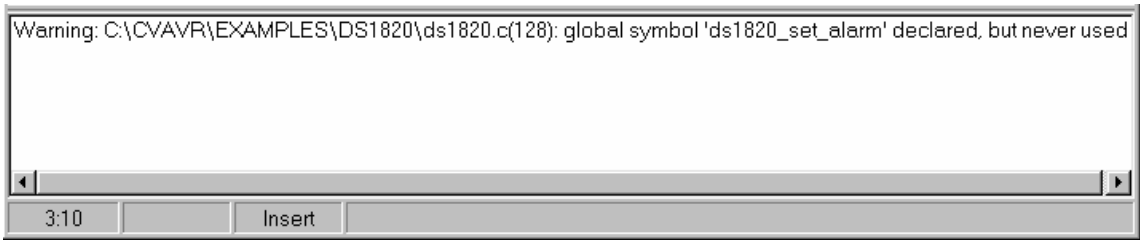
这样一来，建议先选择 **Check the Project for Syntax Errors**，这样由于在进程中不用创建文件可加快速度。

2.2.5.1 检查项目的语法错误

要检查所有项目文件的语法错误，必须使用 **Project|Check Syntax** 菜单命令或点击工具栏中的 **Check Syntax** 按钮。

停止检查进程可使用 **Project|Stop Compilation** 菜单命令或点击工具栏中的 **Stop Compilation** 按钮。

最终的编译错误和/或警告会在 **Editor** 窗口下方的 **Message** 窗口或在 **Navigator** 窗口中列出。



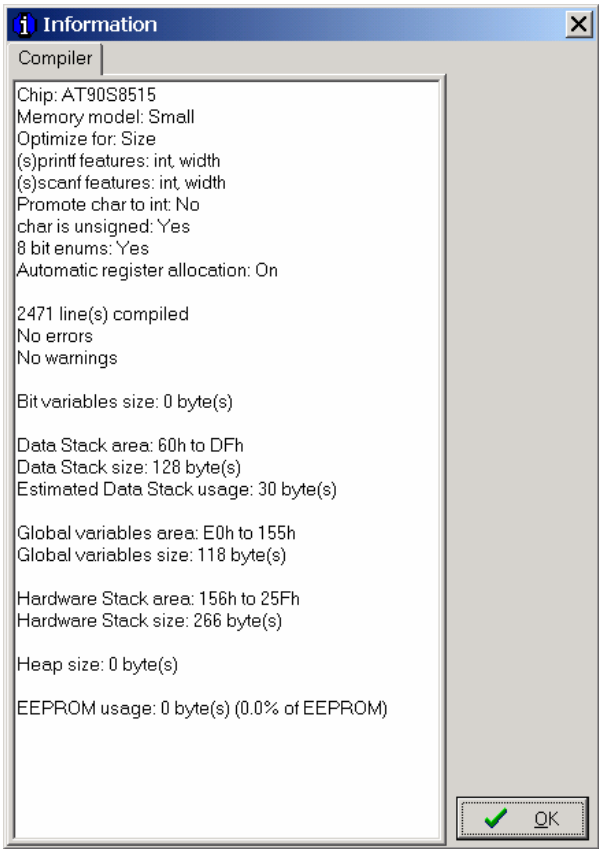
双击错误警告信息，有问题的行以高亮显示。

2.2.5.2 编译项目

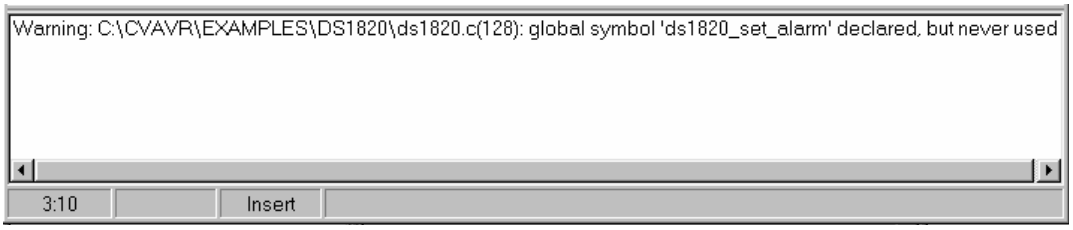
要编译项目，必须使用 **Project|Compile File** 菜单命令、按 F9 或点击工具栏中的 **Compile** 按钮。CodeVisionAVR C 编译器被执行，生成一个扩展名为.asm 的汇编源程序文件。这个文件可在编辑器中打开进行检查和修改。

停止编译进程可使用 **Project|Stop Compilation** 菜单命令或点击工具栏中的 **Stop Compilation** 按钮。

在编译完成后，一个 **Information** 将打开以显示编译结果。



最终的编译错误和/或警告会在 **Editor** 窗口下方的 **Message** 窗口或在 **Navigator** 窗口中列出。



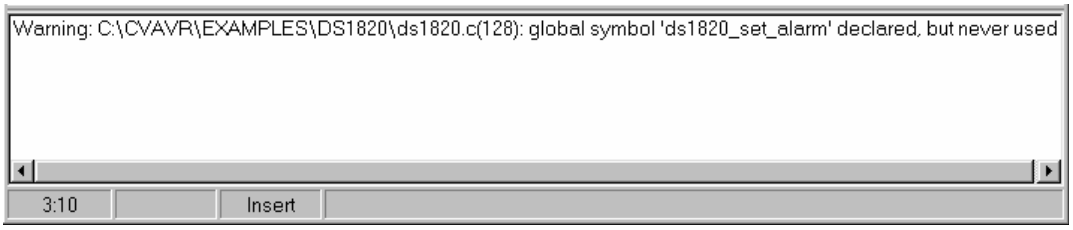
双击错误警告信息，有问题的行以高亮显示。

2.2.5.3 Making the Project

要 make 项目，必须使用 **Project|Make** 菜单命令、按 **Shift+F9** 键或点击工具栏中的 **Make** 按钮。CodeVisionAVR C 编译器被执行，生成一个扩展名为.asm 的汇编源程序文件。

停止编译进程可使用 **Project|Stop Compilation** 菜单命令或点击工具栏中的 **Stop Compilation** 按钮。

最终的编译错误和/或警告会在 **Editor** 窗口下方的 **Message** 窗口或在 **Navigator** 窗口中列出。

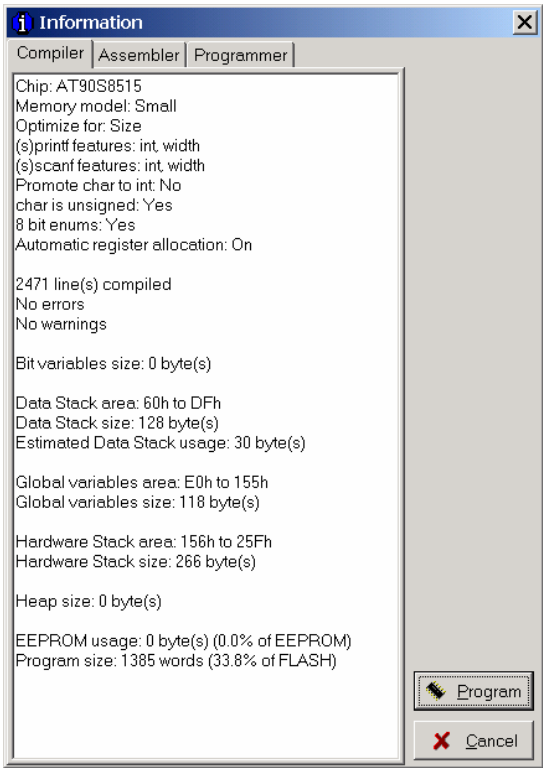


双击错误警告信息，有问题的行以高亮显示。

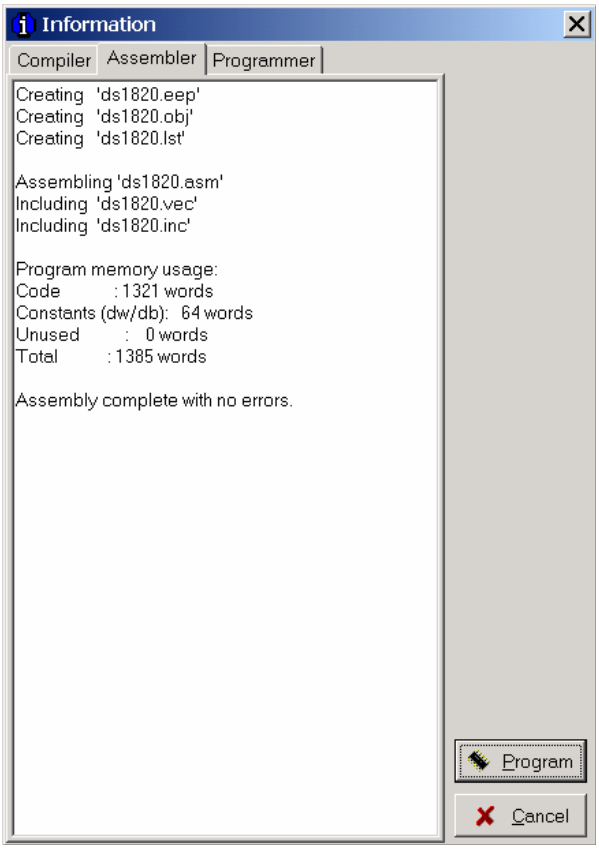
如果没有错误，Atmel AVR 汇编程序 AVRASM32 将被执行，并得到在 **Project|Configure|C Compiler|Code Generation** 中指定类型的输出文件。

在编译完成后，一个 **Information** 将打开以显示编译结果。

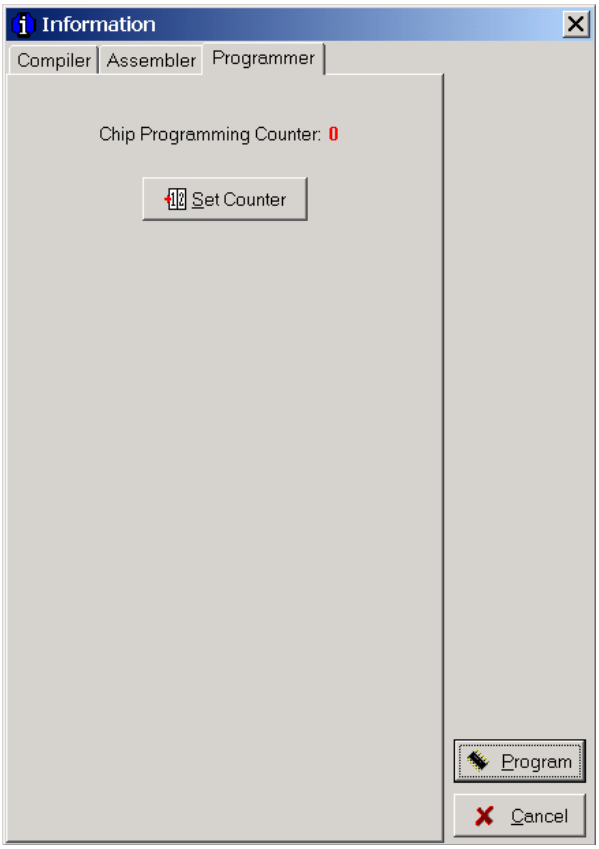
点击 **Compiler** 将显示编译结果。



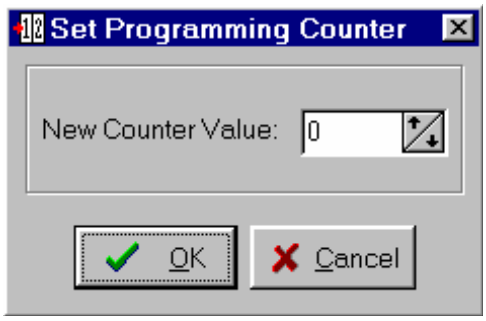
点击 **Assembler** 标签将显示汇编结果。



点击 **Programmer** 标签将显示 **Chip Programming Counter**，以显示 AVR 芯片编程多少次。



点击 **Set Counter** 按钮将打开 **Set Programming Counter** 窗口：



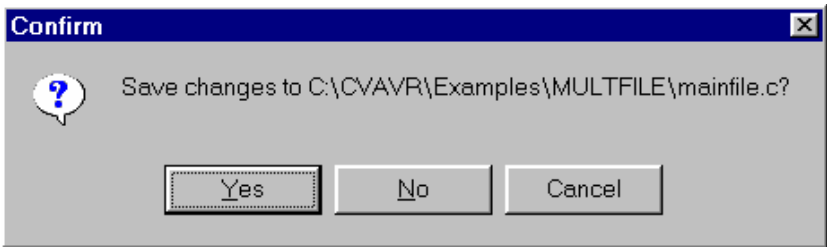
这个对话框允许设置新的 **Chip Programming Counter** 值。

点击 **Program** 按钮允许在成功编译后自动编程 AVR 芯片。点击 **Cancel** 按钮禁止自动编程。

2.2.6 关闭项目

使用 **File|Close Project** 菜单命令可以退出当前项目操作。

如果项目文件被修改并且还未被保存，则有提示是否要保存。



点击 **Yes** 保存修改并关闭项目。

点击 **No** 关闭项目而不保存修改。

点击 **Cancel** 禁止项目关闭进程。

在保存时，IDE 将创建一个扩展名为**.pr~**的备份文件。

2.3 工具（Tools）

使用 **Tools** 菜单可以执行其它程序而不用退出 CodeVisionAVR IDE。

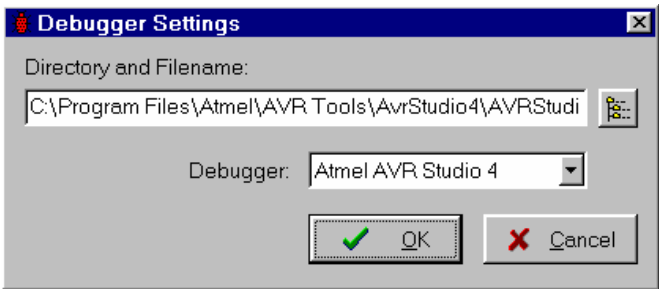
2.3.1 AVR Studio Debugger

CodeVisionAVR C 编译器可与 Atmel AVR Studio 调试器 v3 和 4.06 版（或更高版本）关联操作。

对于 v4.06 或更高版本的 AVR Studio 调试器，编译器会生成一个扩展的 COFF 对象文件，该文件允许观察结构和联合。因此强烈推荐将其用于 v4.06 或更高版本的 AVR Studio 调试器，而不要用于不支持这个特性的 v3 版。

早于 v4.06 版的 AVR Studio 4 不支持扩展的 COFF 对象文件格式，因此不能用于 CodeVisionAVR。

在调用调试器之前，必须先使用 **Settings|Debugger** 菜单命令指定其位置和文件名。



在 **Debugger** 列表框中必须指定 AVR Studio 版本。

点击 OK 保存修改，或点击 Cancel 不保存。

选择 **Tools|Debugger** 菜单命令或点击工具栏中的 **Debugger** 按钮即可运行调试器。

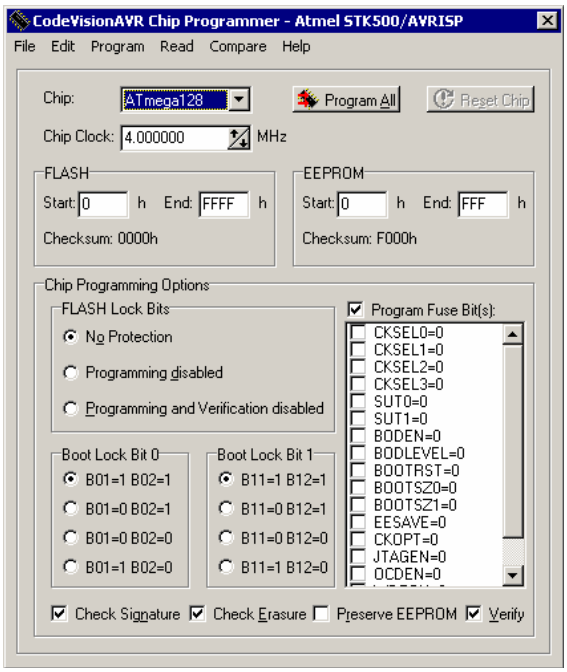
2.3.2 AVR Chip Programmer

CodeVisionAVR IDE 有一个内嵌的 **In-System AVR Chip Programmer**，使用该编程器可以轻松地将编译后的程序传送到微控制器进行测试。

该编程器支持 Atmel STK500、AVRISP、AVRISP MkII、AVRProg (AVR910 application note)、Kanda Systems STK200+、STK300、Dontronics DT006、Vogel Elektronik VTEC-ISP、Futurlec JRAVR 或 MicroTronics ATCPU、Mega2000 开发板。

可以使用 **Settings|Programmer** 菜单命令选择所用编程器和打印机接口类型。

选择 **Tools|Chip Programmer** 菜单命令或点击工具栏中的 **Chip Programmer** 按钮可运行该编程器。



在 **Chip** 列表框中可选择要编程的芯片类型。

Chip Clock 中允许指定对 AVR 芯片编程时的时钟频率。

Atmel STK500、AVRISP 和 AVRISP MkII 的 SCK 编程脉冲周期由此确定。

如果所选的芯片有可编程的熔丝位，附加的 **Program Fuse Bit(s)**复选框将出现。

如果复选该复选框，则在执行 **Program|All** 菜单命令或点击 **Program All** 按钮时将编程该熔丝位。

如果该复选框被复选，芯片的熔丝位将在 Make 后被编程。

熔丝位可设置在 Atmel data sheets 中所描述的各种芯片选项。

如果一个熔丝位复选框被复选，相应的熔丝位将被设置为 0，该熔丝位将被编程（根据每个 Atmel data sheets 中的给定）。

如果一个熔丝位复选框未被复选，相应的熔丝位将被设置为 1，该熔丝位将不编程。

如果要防止程序被复制，必须在 **FLASH Lock Bits** 区选择相应的选项。

编程器有两个存储缓冲区：

- FLASH 存储缓冲区；
- EEPROM 存储缓冲区。

使用 **File** 菜单可加载或保存这些缓冲区中的内容。

支持的文件格式有：

- Atmel .rom and .eep

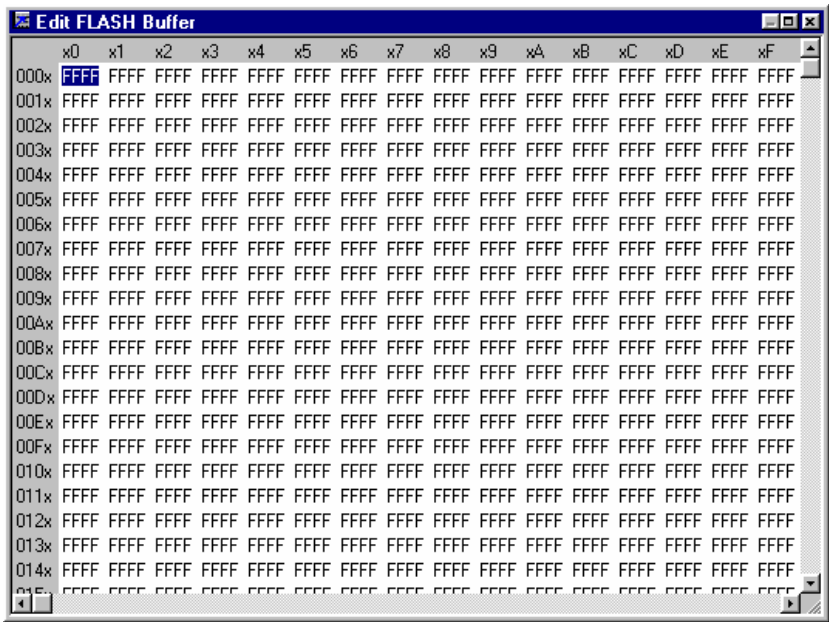
- Intel HEX
- Binary .bin

在相应的缓冲区中加载了一个文件后，**Start** 和 **End** 地址将相应更新。

也可以编辑这些地址。

FLASH 和 EEPROM 缓冲区的内容，可使用 **Edit|FLASH** 和 **Edit|EEPROM** 菜单命令显示和编辑。

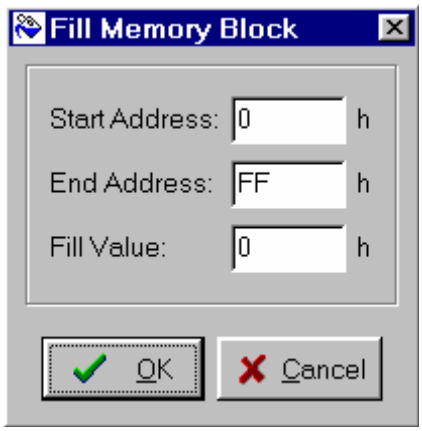
这些命令中有一个被调用时，一个编辑窗口将打开以显示相应缓冲区中的内容。



缓冲区中被高亮显示的地址中的内容，按 **F2** 键可进行编辑和输入新的值。按 **Tab** 或 **arrow** 键将保存编辑后的值。

高亮显示的地址可使用 **arrow**、**Tab**、**Shift+Tab**、**PageUp** 或 **PageDown** 键修改。

在编辑窗口中右击可打开 **Fill Memory Block** 对话框：



在这个窗口中，可指定被填充的存储区的 **Start Address**、**End Address** 和 **Fill Value**。

如果要在任何操作之前检查芯片序列号（signature），则必须使用 **Check Signature** 选项。

要加速编程进程，可以取消复选 **Check Erasure** 复选框。

这样就不会校验 FLASH 擦除的正确性。

Preserve EEPROM 复选框允许在芯片擦除时保留 EEPROM 的内容。

要加速编程进程，还可以取消 **Verify** 复选框。

这样就不会校验 FLASH 和 EEPROM 编程的正确性。

要擦除芯片的 FLASH 和 EEPROM，必须选择 **Program|Erase** 菜单命令。

在擦除后，芯片的 FLASH 和 EEPROM 自动进行空白检查。

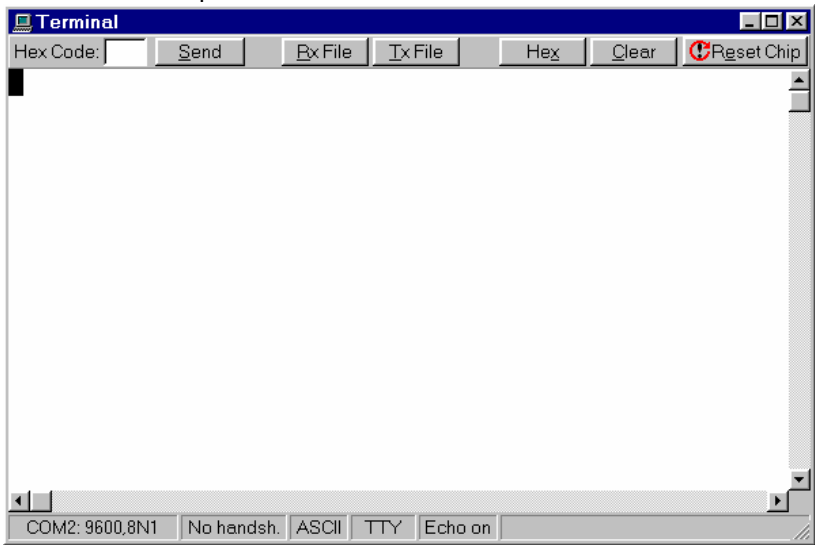
对于简单的空白检查，必须使用 **Program|Blank Check** 菜单命令。
如果要用 FLASH 缓冲区的内容编程 FLASH，必须使用 **Program|FLASH** 菜单命令。
对于编程 EEPROM，必须使用 **Program|EEPROM** 菜单命令。
在编程后，FLASH 和 EEPROM 自动进行校验。
要编程锁定或熔丝位，必须使用 **Program|Fuse Bit(s)**或 **Program|Lock Bits** 菜单命令。
Program|All 菜单命令允许自动进行以下操作：

- 擦除芯片；
- FLASH 和 EEPROM 空白检查；
- 编程和校验 FLASH；
- 编程和校验 EEPROM；
- 编程熔丝位和锁定位。

如果要读取芯片 FLASH 或 EEPROM 的内容，必须使用 **Read|FLASH** 或 **Read|EEPROM** 菜单命令。
要读取芯片的序列号，必须使用 **Read|Chip Signature** 菜单命令。
要读取锁定或熔丝位，必须使用 **Read|Lock Bits** 或 **Read|Fuse Bits** 菜单命令。
对某些器件，也有 **Read|Calibration Byte(s)**选项。
这允许读取芯片内部 RC 振荡器的标度字节值。
如果编程器是 Atmel STK500、AVRISP、AVRISP MkII 或 AVRProg (AVR910 application note)，将有对应的菜单命令：**Read|Programmer's Firmware Version**。这允许读取上述编程器的硬件主要的和次要的版本信息。
要芯片 FLASH 或 EEPROM 与相应存储缓冲区的内容进行比较，，必须使用 **Compare|FLASH** 或 **Compare|EEPROM** 菜单命令。
要退出编程器或返回 CodeVisionAVR IDE，必须使用 **File|Close** 菜单命令。

2.3.3 串行通信终端

终端（**Terminal**）是专为使用串行通信(RS232, RS422, RS485)调试嵌入式系统而设计的。
终端的调用可以使用 **Tools|Terminal** 菜单命令或工具栏中的 **Terminal** 按钮。



字符以 ASCII 或十六进制格式显示。显示模式可以使用 **Hex/ASCII** 按钮设置。
使用 **Rx File** 按钮可以将接收到的字符保存到文件中。
在终端窗口中输入的任何字符都将通过 PC 的串行口发送。
输入的字符可以使用 **Backspace** 键删除。
点击 **Send** 按钮，终端将发送一个字符，该字符的十六制代码在 Hex Code 编辑框中指定的。

点击 **Tx File** 按钮，一个文件的内容通过串行中发送。

点击 **Reset** 按钮，复位 STK200+/300、VTEC-ISP、DT006、ATCPU 或 Mega2000 开发板上的 AVR 芯片。

在终端窗口的底部，有一个状态栏，该栏显示以下内容：

- 计算机的通信口；
- 通信参数；
- 握手模式；
- 接收字符显示模式；；
- 仿真终端的类型；
- 发送字符的状态的禁止设置。

2.3.4 执行用户程序

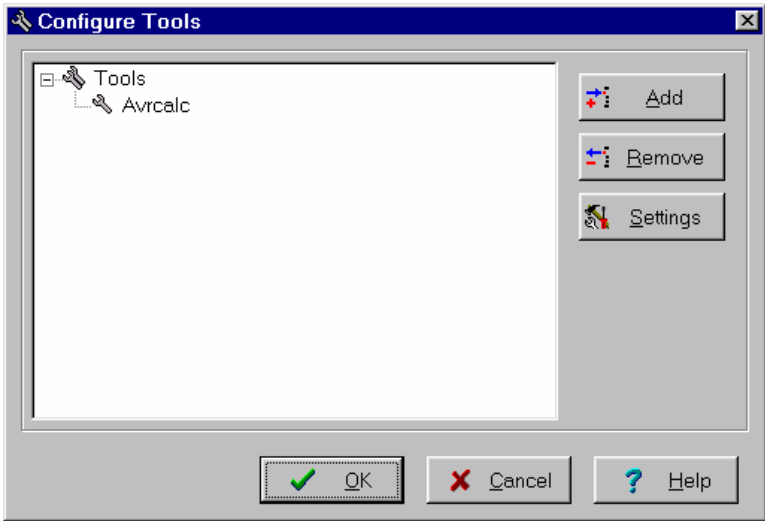
通过从 **Tools** 菜单选择相应的命令执行用户程序。

必须先添加程序名到菜单。

2.3.4.1 配置 Tools 菜单

使用 **Tools|Configure** 菜单命令可以从 **Tools** 菜单添加或移除用户程序。

带有使用程序列表的 **Configure Tools** 对话框窗口将打开。



点击 **Add** 按钮添加程序到 **Tools** 菜单。

点击 **Remove** 从 **Tools** 菜单移除程序。

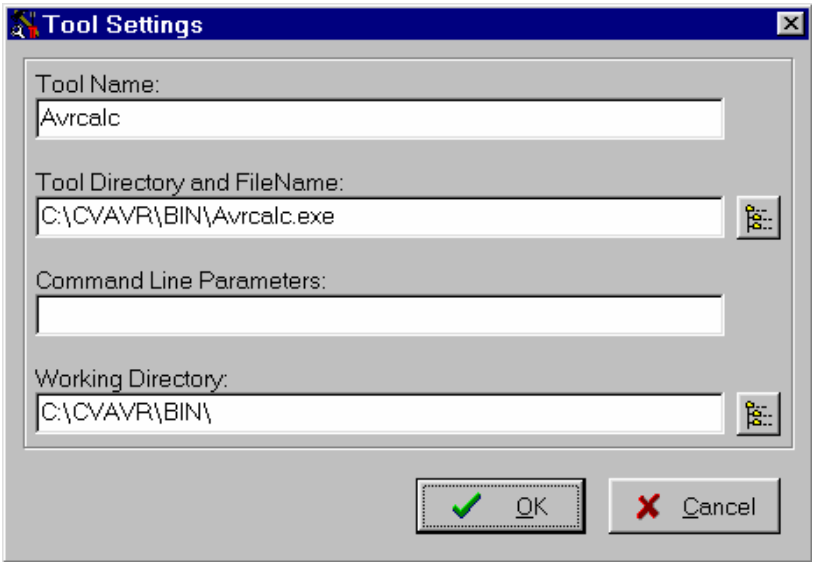
点击 **Settings** 按钮可修改以下内容：

Tool Menu Name——工具菜单名；

Tool Directory and File Name——工具目录和文件名；

Command Line Parameters——命令行参数；

Working Directory of a selected Program from the list.——从列表中所选程序的工作目录。



点击 OK 保存修改，或点击 Cancel 不保存。

2.4 IDE 设置

使用 **View** 和 **Settings** 菜单配置 CodeVisionAVR IDE。

2.4.1 View 菜单

以下设置通过 **View** 菜单命令配置：

如果复选 **View|Toolbar** 选项，将显示命令按钮工具栏；

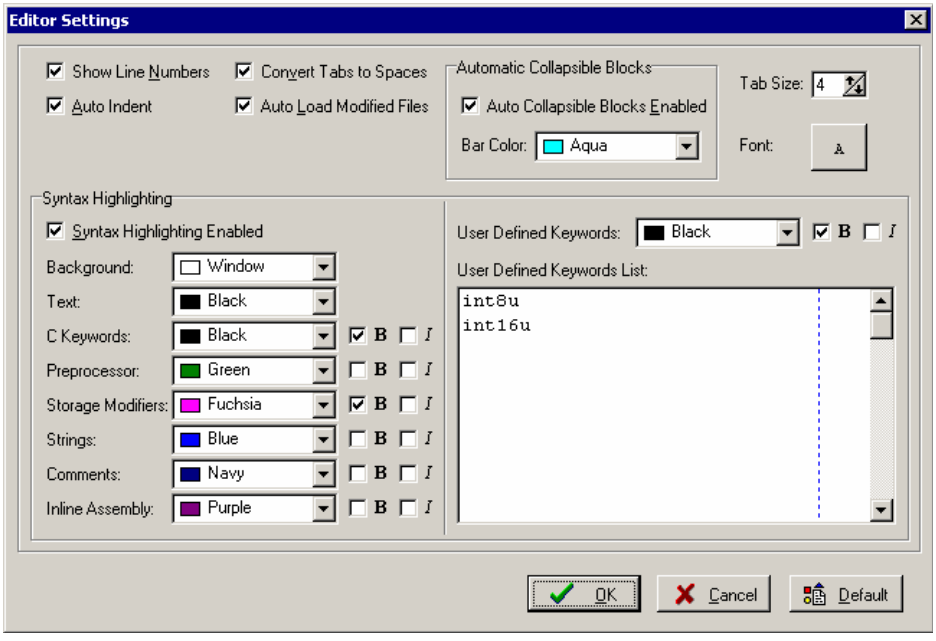
如果复选 **View|Navigator/Code Templates/Clipboard History** 选项，the **Navigator**、**Code Templates** 和 **Clipboard History** 窗口将显示在 **Editor** 窗口的左边；

如果复选 **View|Messages** 选项，**Message** 窗口将显示在 **Editor** 窗口下面；

如果复选 **View|Information Window after Compile/Make** 选项，在 **Compiling** 或 **Making** 之后会显示一个 **Information** 窗口。

2.4.2 配置 Editor

Editor 使用 **Settings|Editor** 菜单命令配置。



Show Line Numbers 复选框，允许或禁止显示 Editor 窗口中的行数。

Autoindent 复选框，允许或禁止在编辑时文本自动缩进。

Auto Load Modified Files 复选框，允许或禁止文件的自动加载，该文件是在 Editor 中打开的并由外部程序修改的。

Convert Tabs to Spaces 复选框，允许或禁止在输入时以适当数量的空格来代替制表符。在按 Tab 键时的空格数可以在 **Tab Size** 框中指定。

Auto Collapsible Blocks Enabled 复选框，允许或禁止在 Editor 窗口左边限定 collapsible 块的垂直栏的自动显示。

Automatic Collapsible Blocks|Bar Color 列表框指定 collapsible 块的垂直栏的颜色。

用于文件 **Editor** 和 **Terminal** 的字体可点击 **Font** 按钮指定。

复选或取消复选 **Syntax Highlighting Enabled** 复选框，可允许或禁止在 Editor 窗口中用颜色高亮显示 C 语法。

用于 Editor 窗口中的文本和 C 语法高亮显示的各种颜色、属性可以在 **Syntax Highlighting** 区中相应的列表框中指定。

Background 和 **Text** 颜色设置对 EditorEdit File 和 TerminalTerminal 都是一样的。

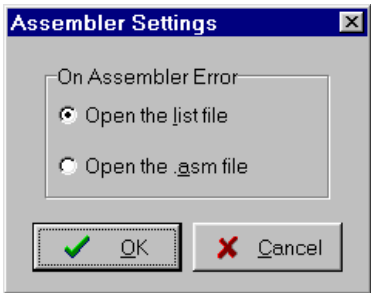
User Defined Keywords List 包含了需要高亮显示的语法的附加关键字。这些文本颜色和属性可以在 **User Defined Keywords** 列表框和 **B** 和 *I* 复选框中指定。

点击 OK 保存 **Editor** 配置的修改，或点击 Cancel 不保存。

点击 **Default** 按钮，恢复到默认的 **Editor** 设置。

2.4.3 配置 Assembler

Assembler 通过 **Settings|Assembler** 菜单命令配置。



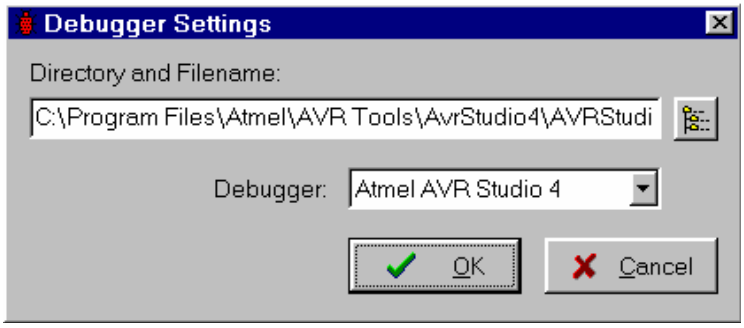
On Assembler Error 选项允许选择在发生汇编错误时由 **Editor** 自动打开哪个文件。

点击 OK 保存 **Assembler** 配置的修改，或点击 Cancel 不保存。

2.4.4 设置 Debugger 路径

CodeVisionAVR C 编译器可与 Atmel AVR Studio 调试器 v3 和 4.06 版（或更高版本）关联操作。

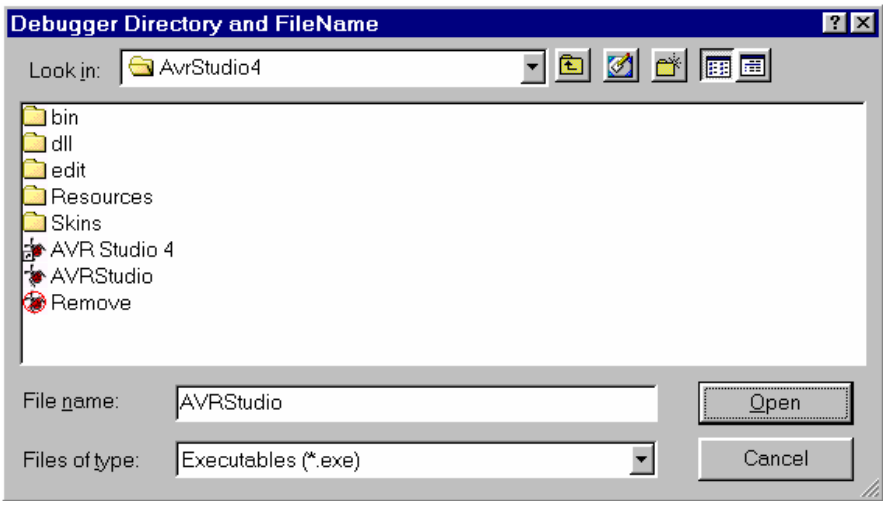
在调用调试器之前，必须通过 **Settings|Debugger** 菜单命令指定其位置和文件名。



在 **Debugger** 列表框中必须指定 AVR Studio 版本。

点击 OK 保存修改，或点击 Cancel 不保存。

点击 **Browse** 按钮打开一个对话框窗口选择调试器目录和文件名。



2.4.5 AVR 芯片 Programmer 设置

通过 **Settings|Programmer** 菜单命令，可以选择使用的在系统编程器（in-system programmer）的类型，以及编程器连接的计算机端口。

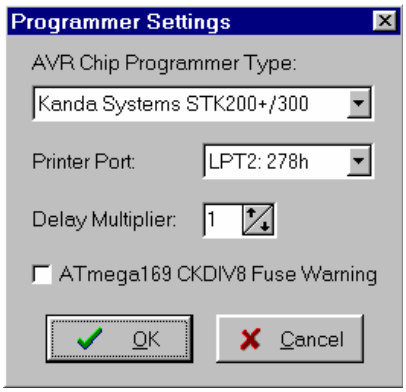
CodeVisionAVR 当前版本支持以下在系统编程器（in-system programmer）：

- Kanda Systems STK200+ and STK300
- Atmel STK500 and AVRISP (serial connection)
- Atmel AVRISP MkII (USB connection)
- Atmel AVRProg (AVR910 application note)
- Dontronics DT006
- Vogel Elektronik VTEC-ISP
- Futurlec JRAVR
- MicroTronics ATCPU and Mega2000

The STK200+、STK300、DT006、VTEC-ISP、JRAVR、ATCPU 和 Mega2000 在系统编程器（in-system programmer）使用并行打印机端口。

以下选项通过 **Printer Port** 列表框选择：

- LPT1——基地址为 378h;
- LPT2——基地址为 278h;
- LPT3——基地址为 3BCh.



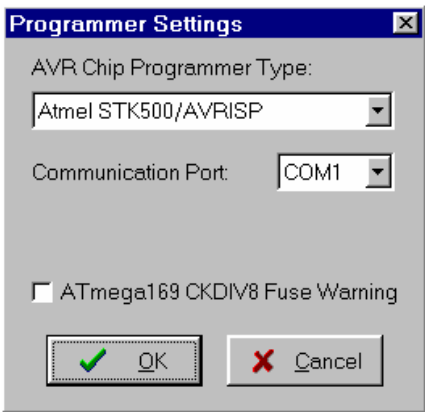
增大 **Delay Multiplier** 值可解决在快速机器上的编程问题。

当然这将增加全部编程时间。

如果复选 **Atmega169 CKDIV8 Fuse Warning** 复选框，在熔丝位被编程为 0 时，对 *Atmega169 Engineering Samples*，将允许在低电压串行编程时不产生警告。

对一般的 Atmega169 芯片，这个复选框必须保留为取消复选。

STK500、AVRISP 和 AVRProg 编程器使用 RS232C 串行通信端口，这可以在 **Communication Port** 列表框中指定。



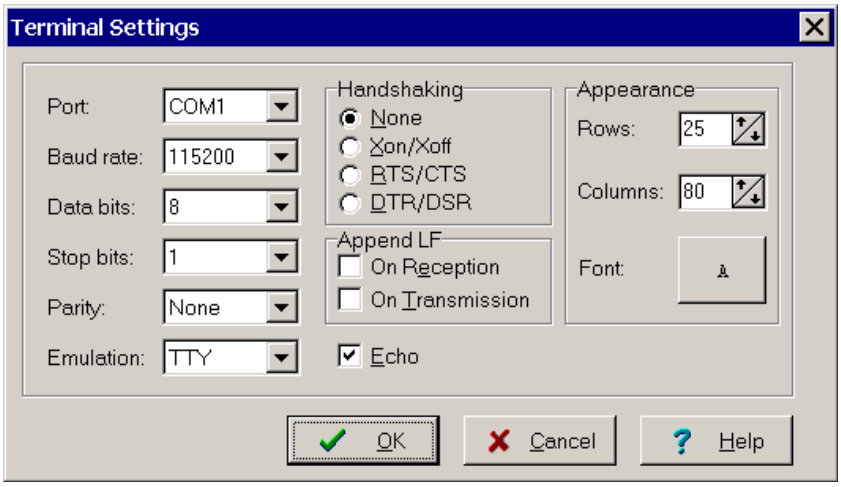
Atmel AVRISP MkII 使用 USB 连接与 PC 通信。

使用该编程器需要 Atmel's AVR Studio V4.12 或更高版本。

点击 OK 保存修改，或点击 Cancel 不保存。

2.4.6 串行通信 Terminal 设置

串行通信 **Terminal** 通过 **Settings|Terminal** 菜单命令设置。



在 **Terminal Setup** 窗口，可以选择：

- 由终端使用的计算机通信端口：COM1~COM6；
- 用于通信的波特率：110~115200；
- 接收和发送数据位数：5~8；
- 接收和发送的停止位数：1、1.5 或 2；
- 接收和发送的奇偶检验：None、Odd、Even、Mark 或 Space；
- 仿真终端的类型：TTY、VT52 或 VT100；
- 用于通信的握手类型：None、Hardware (CTS 或 DTR)或 Software (XON/XOFF)；

在接收和发送时附加在 CR 字符后的 LF 字符的可能性：

- 允许或禁止发送字符的禁止
- 在 Terminal 窗口中的字符 **Rows** 和 **Columns** 数；
- 用于在 Terminal 窗口中显示字符的 **Font**（字体）类型。

点击 OK 保存修改，或点击 Cancel 不保存。

2.5 访问 Help

CodeVisionAVR 帮助系统可通过调用 **Help|Help** 菜单命令或点击 **Help** 工具栏按钮进行访问。

2.6 传送 License 到另一台计算机

略

2.7 连接到 HP InfoTech 网站

Help|HP InfoTech on the Web 菜单命令打开默认的网页浏览器并连接到 HP InfoTech 网站 **http://www.hpinfotech.com**

Here you can check for the latest HP InfoTech's products and updates to CodeVisionAVR.

2.8 通过 E-Mail 联系 HP InfoTech

Help|E-Mail HP InfoTech 菜单命令打开默认的 e-mail 程序，并允许发送 e-mail 到：**office@hpinfotech.com**。

2.9 退出 CodeVisionAVR IDE

要退出 CodeVisionAVR IDE 必须选择 **File|Exit** 菜单命令。

如果某些源文件被修改而未保存，则有提示是否要保存。