

# 用汇编语言编写 AVR 单片机程序入门教程

作者：伟纳电子 Gguoqing

原文发表在伟纳电子论坛：[http://www.willar.com/forum\\_view.asp?forum\\_id=25&view\\_id=5438](http://www.willar.com/forum_view.asp?forum_id=25&view_id=5438)

## 1. 硬件准备

实验系统：ME300 全系列单片机开发板。

实验芯片：ATmega8515L 或 Atmega8515。

## 2. 软件准备

AVRStudio V4.11 <http://www.atmel.com>

AVRStudio 是一个完整的开发工具，包括编辑、仿真功能，利用这个工具我们可以编辑源程序代码，并在 AVR 器件上运行。

ME300\_V3.1 版控制软件 [http://www.willar.com/down\\_view.asp?id=315](http://www.willar.com/down_view.asp?id=315)

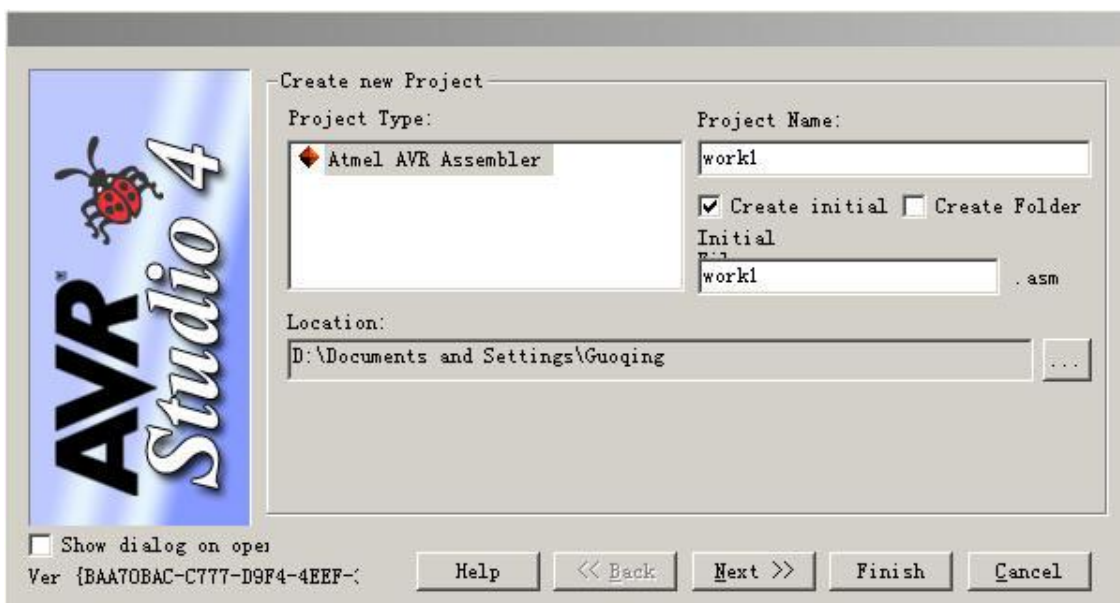
上述软件有收录在 ME300 随机光盘中

## 3. AVR 汇编语言学习环境的建立

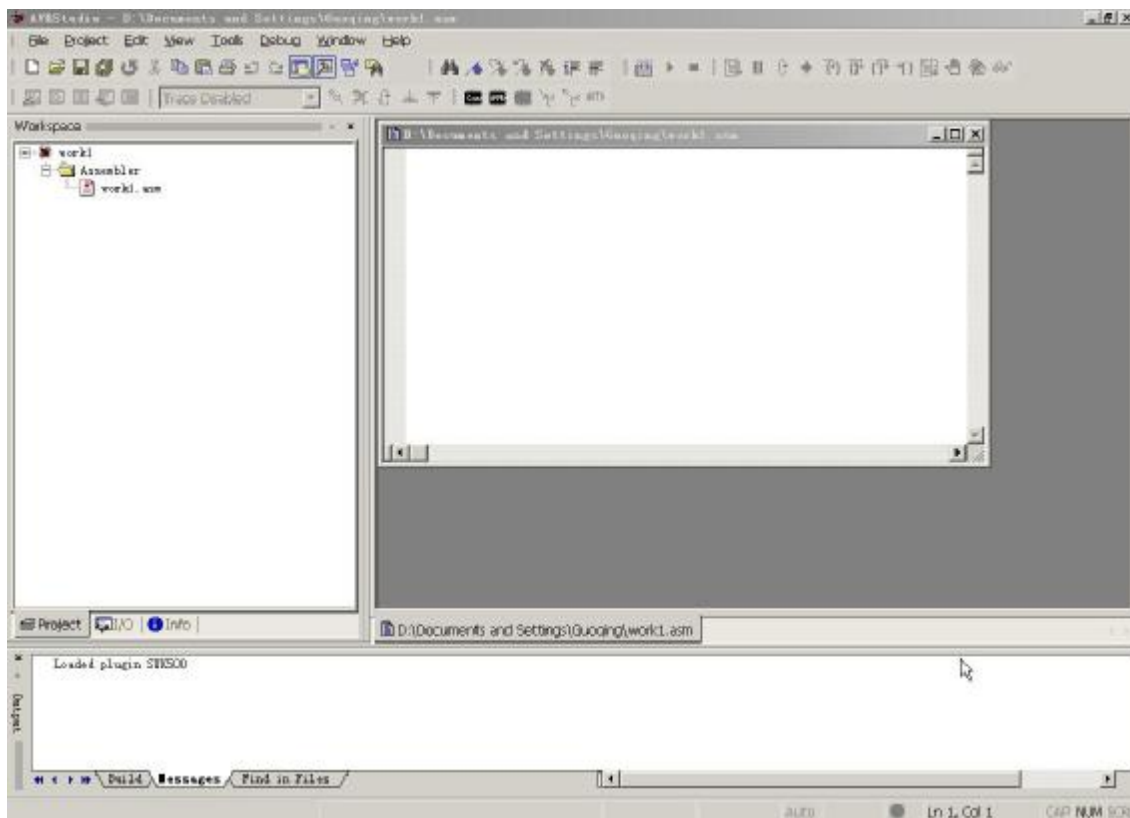
安装 AVRStudio 和 ME300\_V3.1 版控制软件。

### 1) 创建一个新项目：

打开 AVR Studio 软件，选择“Project”菜单中的“New project”命令，弹出下面窗口。这里新建一个名为 work1 的项目。并在“Location”栏目中确定存放文件的路径与相应的文件夹。



点击“Finish”存盘后出现下面界面。



## 2) 编辑汇编源程序:

用汇编语言编写的源程序如下:

;8 只 LED 的跑马灯演示程序

```
.INCLUDE "8515DEF.INC" ;CPU 配置文件
.ORG $0000
RJMP RESET           ;上电复位跳转到主程序
.ORG $0013           ;代码段定位, 跳过中断区
```

RESET:

```
LDI R16,LOW(RAMEND) ;设置堆栈指针
OUT SPL,R16
LDI R16,HIGH(RAMEND)
OUT SPH,R16
LDI R16,0B11111111 ;设置 PA 口为输出端口
OUT DDRA,R16
```

START:

```
LDI R17,0B11111110 ;PA0 输出低电平,LED0 亮。
OUT PORTA,R17
LDI R16,50 ;延时 1 秒
RCALL DELAY
```

```
LDI R17,0B11111101 ;PA1 输出低电平,LED1 亮。
OUT PORTA,R17
LDI R16,50 ;延时 1 秒
RCALL DELAY
```

```
LDI R17,0B11111011 ;PA2 输出低电平,LED2 亮。
OUT PORTA,R17
LDI R16,50 ;延时 1 秒
RCALL DELAY
```

```
LDI R17,0B11110111 ;PA3 输出低电平,LED3 亮。
OUT PORTA,R17
LDI R16,50 ;延时 1 秒
RCALL DELAY
```

```
LDI R17,0B11101111 ;PA4 输出低电平,LED4 亮。
OUT PORTA,R17
LDI R16,50 ;延时 1 秒
RCALL DELAY
```

```
LDI R17,0B11011111 ;PA5 输出低电平,LED5 亮。
OUT PORTA,R17
LDI R16,50 ;延时 1 秒
RCALL DELAY
```

```
LDI R17,0B10111111 ;PA6 输出低电平,LED6 亮。
OUT PORTA,R17
LDI R16,50 ;延时 1 秒
RCALL DELAY
```

```
LDI R17,0B01111111 ;PA7 输出低电平,LED7 亮。
OUT PORTA,R17
LDI R16,50 ;延时 1 秒
RCALL DELAY
```

```
RJMP START ;循环执行
```

```
DELAY: ;通用延时子程序
PUSH R16 ;延时时间由 R16 调整
```


```

DELAY0:
PUSH R16
DELAY1:
PUSH R16
DELAY2:
PUSH R16
DELAY3:
DEC R16
BRNE DELAY3
POP R16
DEC R16
BRNE DELAY2
POP R16
DEC R16
BRNE DELAY1
POP R16
DEC R16
BRNE DELAY0
POP R16
RET

```

可以运用复制 / 粘贴方式将程序代码复制到 AVRStudio 的编辑窗口。

演示源程序:  [ 点击下载 ] ;(文件大小: 530 Byte)

通用延时子程序资料:  [ 点击下载 ] ;(文件大小: 13.833 KB)

### 3) 汇编源程序编译

3.1: 点击“Project”菜单中的“AVR Assemble setup”命令，弹出编译设置窗口，然后按下图设置: Hex Output Format 选择 {Intel intellec 8/MDS (Intel hex)}, 点击OK。

程序调试方法:

用 AVR Studio 软件模拟单片机的运行，观察单片机执行情况是否符合我们的要求。

★ 启动 AVR Studio 软件，选择“File”菜单中的“Open File”命令，汇编语言编写的程序则选择打开 Object Files (\*.hex ; \*.d90 ; \*.a90 ; \*.Obj)。见图 1

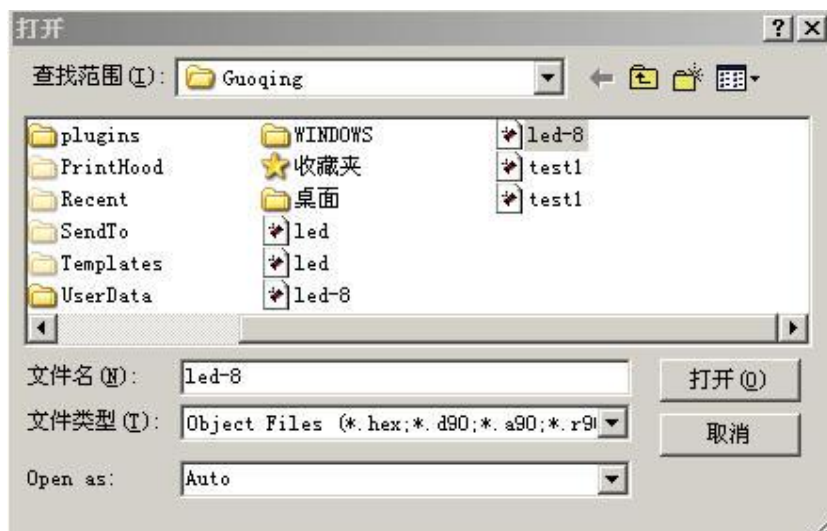


图 1 调入调试文件

(如果是调试 C 语言编写的程序则打开\*.cof 文件)。这时弹出仿真选项窗口，见图 2：这里选择软件仿真“AVR Simulator”，目标 CPU 选择“Atmega8515”。

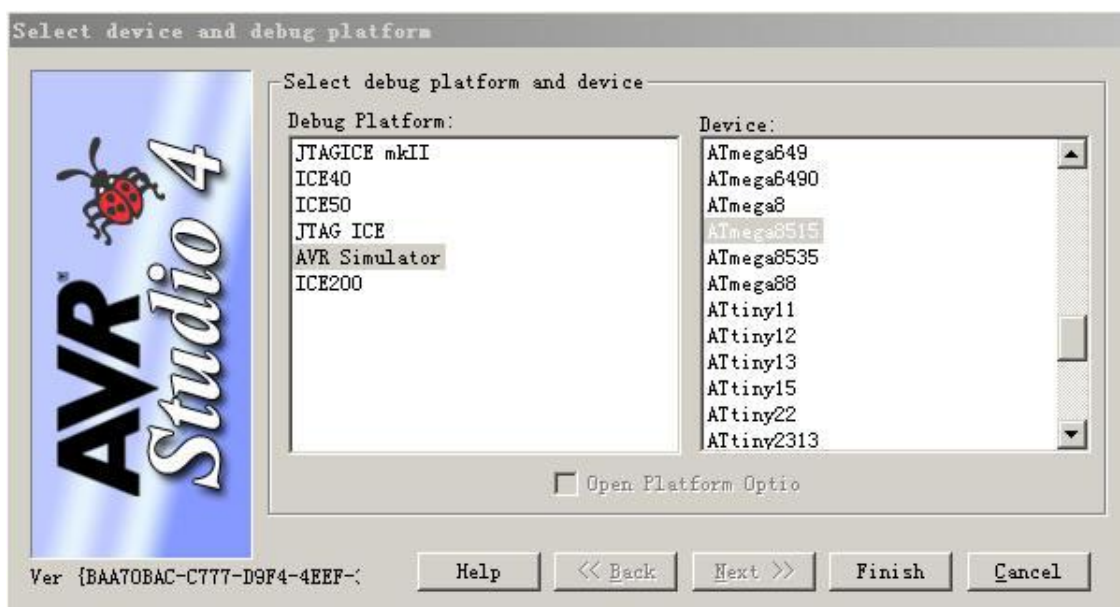


图 2 选择调试工具和目标 CPU

★ 点击上面窗口的“Finish”按钮后，出现下面界面见图 3：左边的 Workspace 的 I/O 子窗口显示的是 CPU 的状态，右边窗口显示的是你的源程序。

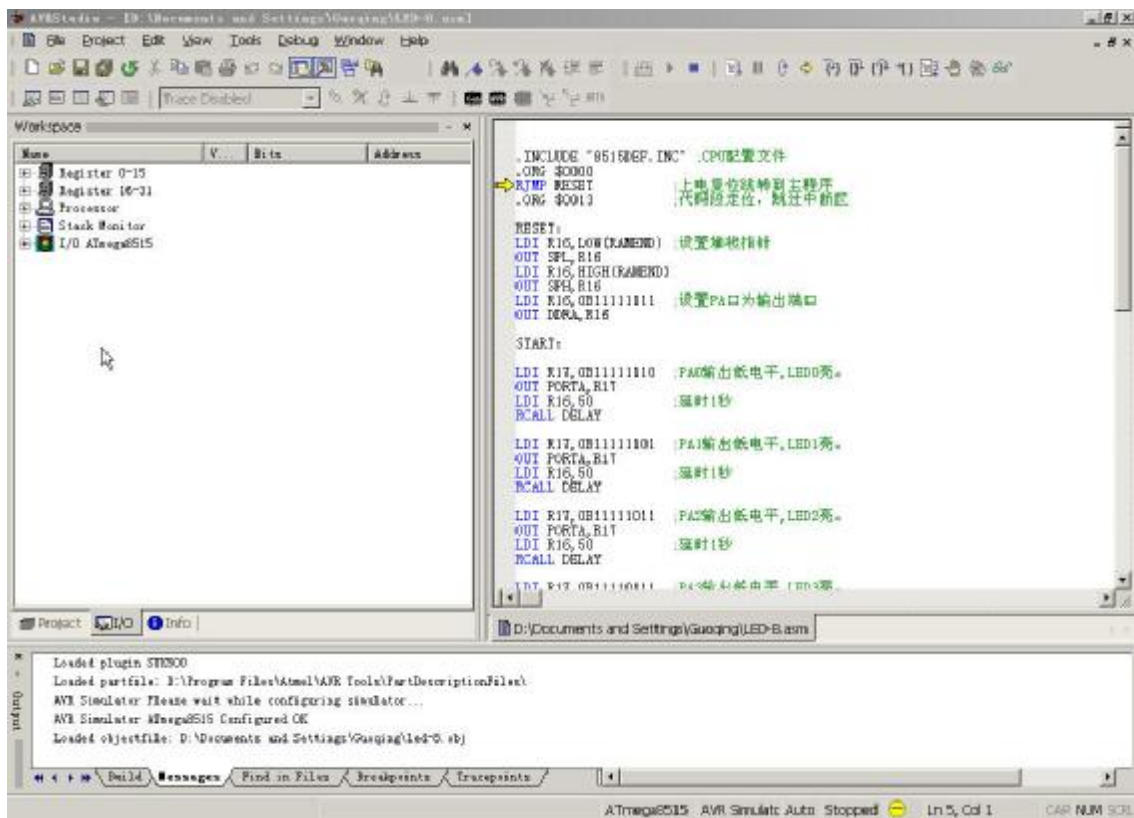


图 3 调试文件窗口

★ 将 Processor 和 I/O ATmega8515 状态展开，见图 4：在这里我们可以详细的观察控制本实验中 PA 口各脚的电平变化，以及延时情况。

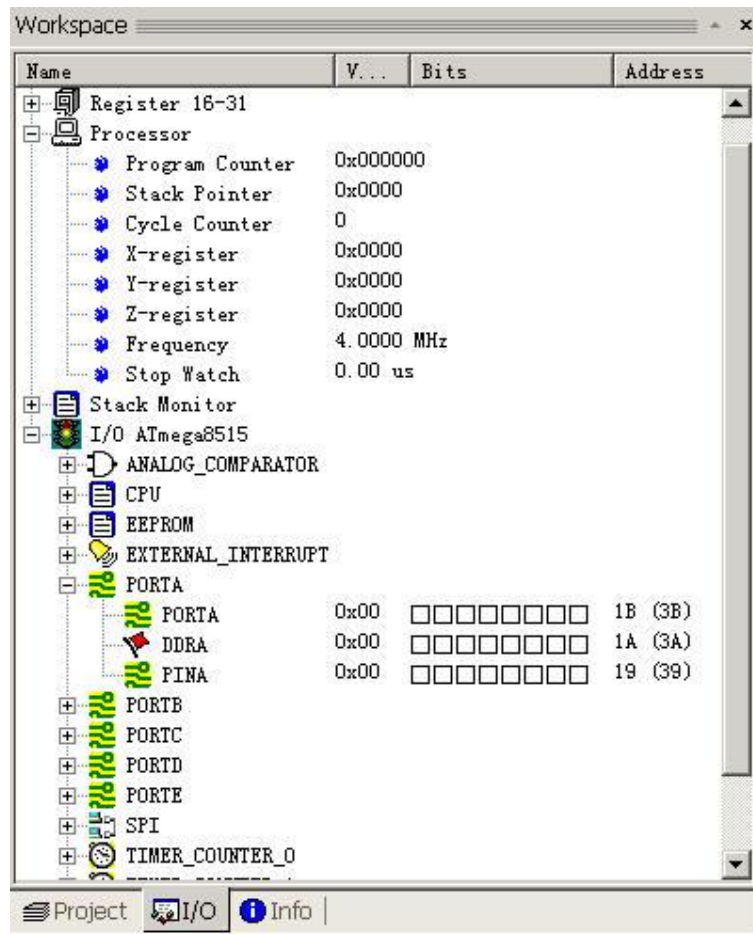
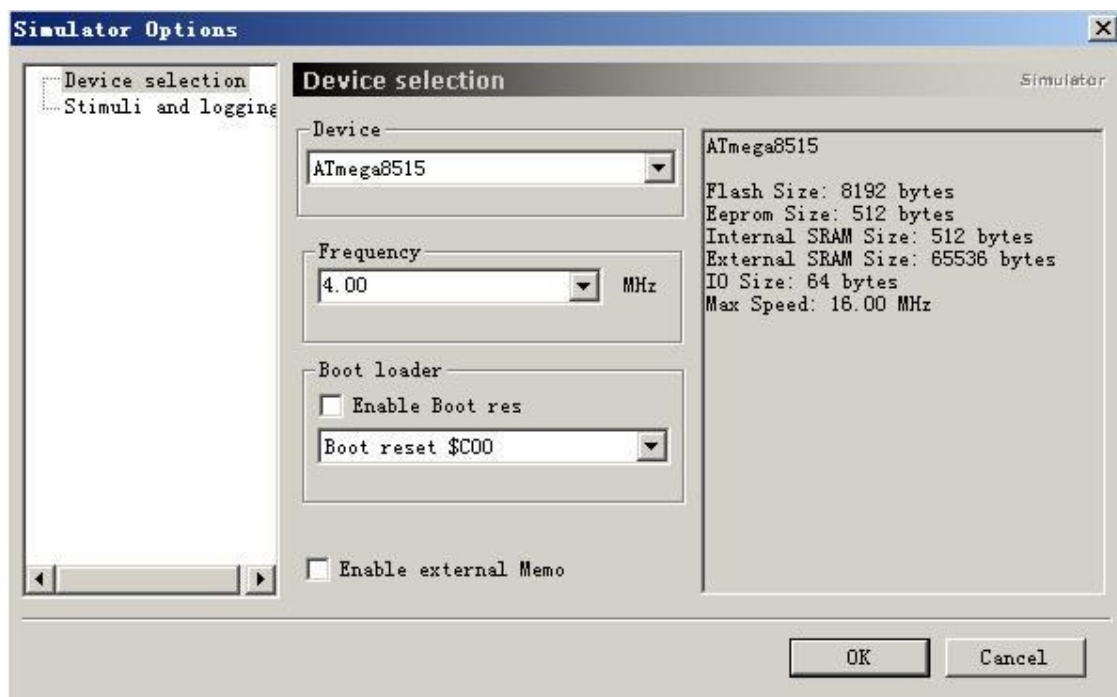
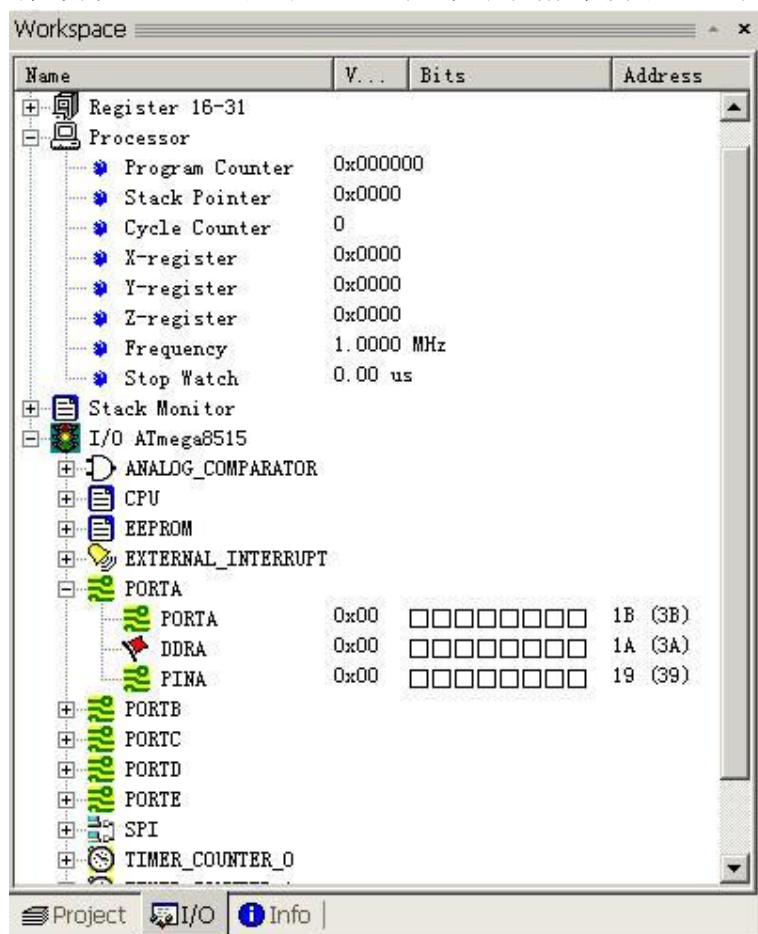


图 4

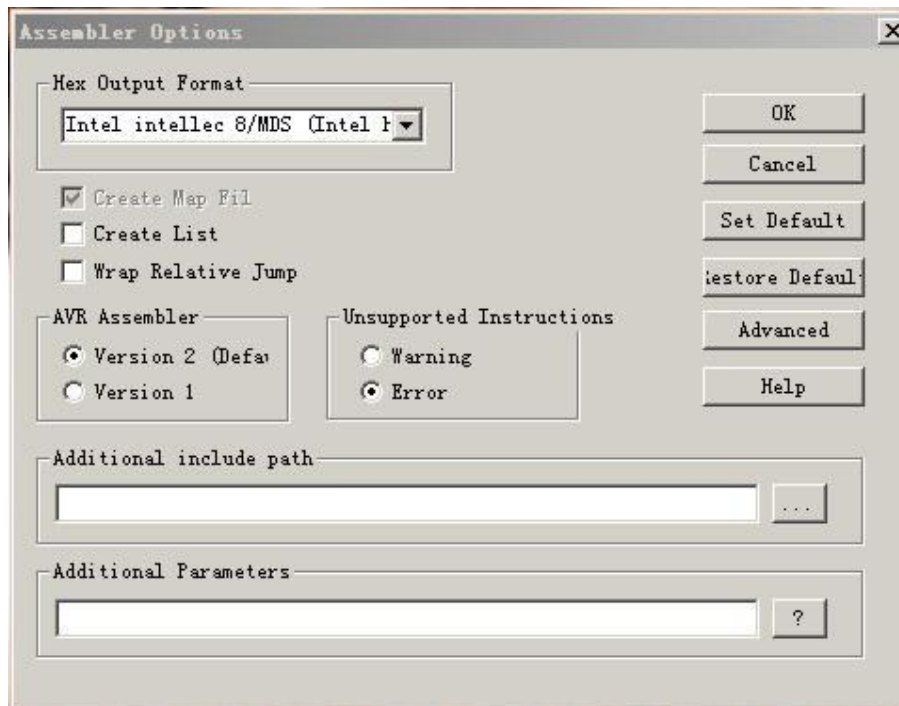
★ 我们注意到在上面窗口中，显示的时钟频率是 4MHz，但本实验采用的时钟频率是 1MHz，因此选择“Debug”菜单中的“AVR Simulator Options”命令，弹出下面窗口：



将 Frequency 框内的值改为 1.00 MHz，点击 OK。这时显示的时钟频率就是 1 MHz 了。见图 6







3.2: 点击“Project”菜单中的“Build”命令开始进行编译。源程序编译完成以后将生成 work1.hex 目标文件。

在 Output 窗口内显示:

Building project...

AVRASM: AVR macro assembler 2.0.31 (build 58 Apr 21 2005 14:17:40)

Copyright (C) 1995-2005 ATMEL Corporation

D:\documents and Settings\Guoqing\work1.asm(3): Including file 'D:\Program Files\Atmel\AVR Tools\AvrAssembler2\Appnotes\8515DEF.INC'

AT90S8515 memory use summary [bytes]:

Segment	Begin	End	Code	Data	Used	Size	Use%
-----							
[.cseg]	0x000000	0x000096	114	0	114	8192	1.4%
[.dseg]	0x000060	0x000060	0	0	0	512	0.0%
[.eseg]	0x000000	0x000000	0	0	0	512	0.0%

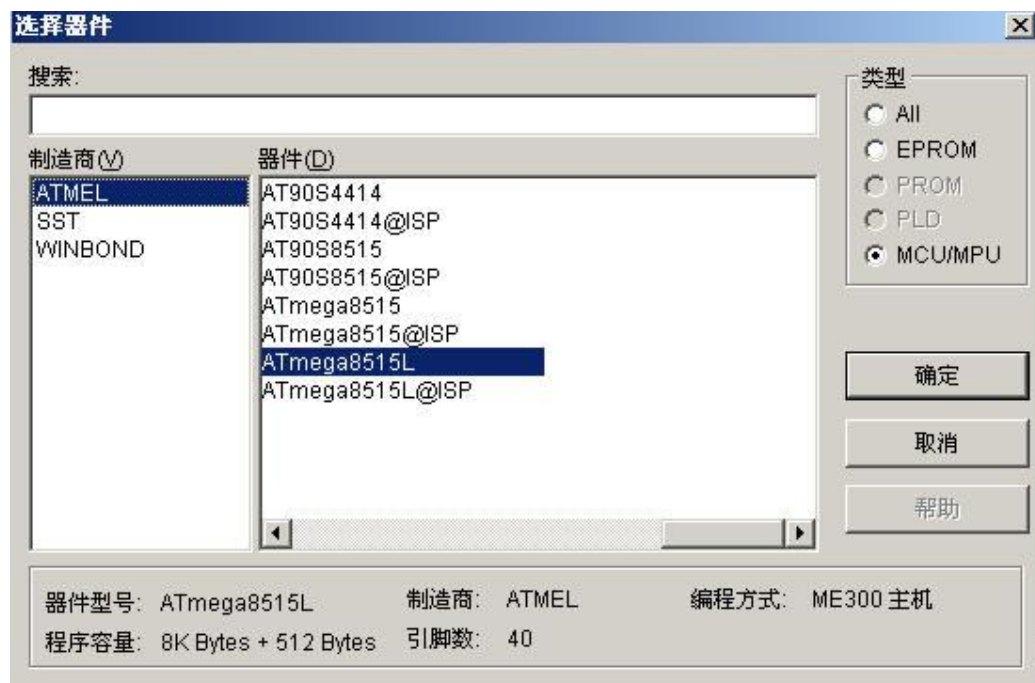
Assembly complete, 0 errors. 0 warnings

## 4. 烧写程序/演示

4.1 连接好 ME300，设置 ME300 相关跳线到 LED 位置，锁紧座上放入一片 ATmega8515L 或 Atmega8515，开启电源。

4.2 运行 ME300 软件（V3.1 或以上版本）。

#### 4.3 ME300 软件中选择相应的芯片型号。



器件配置选项中按下图设置:



这里只设置第 0 页即可，其中的“时钟源选择”（熔丝位）选项中，选中表示 0，不选中表示 1。如上图设置的 CKSEL3..0 的值为：0001，按上图设置并编程后，单片机即使用内部 RC 振荡器，工作频率 1MHz，

**注意：**熔丝位配置错误将会锁死芯片导致再编程失败。

#### 4.4 加载编译的文件 work1.hex

#### 4.5 烧写程序，正常即可观查到流水灯效果！