

在 ME300 上演示 AVR 实验入门教程

作者：伟纳电子(www.willar.com)网站 Sauwa

原文发表在伟纳电子论坛：http://www.willar.com/forum/view.asp?forum_id=25&view_id=5359

ME300 系列单片机学习开发系统配合 ME300_V3.1 版软件可以支持 AVR 系列单片机的烧写和实验。本文以一个流水灯程序(C 代码，实现 8 只 LED 的跑马灯程序)为例子，介绍 AVR 单片机开发的基本流程及如何在 ME300 上烧写和实验。

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，该芯片的引脚与 89S51/52 系列单片机完全兼容，只是其复位电平不一样而已，ME300 系列开发板在监控 CPU 的控制下可正常复位 AVR 芯片，无需改动硬件

2. 软件准备：

AVR 的开发语言与工具很多，常用的是 C，有许多不同的开发工具(如 ICCAVR, CodeVersion, IAR, GCC...), 彼此不兼容。其中 WinAVR (即 GCC, 完全免费) 是全球开发 AVR 使用人数最多的软件，国外最大的 AVR 资源网站 [avrfreaks](http://www.avrfreaks.com) 也是推荐使用这个软件。我们使用它进行 C 代码的编写与编译。WinAVR 下载地址：<http://sourceforge.net/projects/winavr>。在 ME300 的配套光盘的工具软件中也收录有这个软件！

3. AVR C 语言学习环境的建立

3.1 安装 WinAVR，按默认安装即可

3.2 在硬盘上建立一个文件夹，专门用来放项目文件，例：在 D 盘上建立一个名为 AVR_LED 的文件夹

3.3 运行 WinAVR 的 Programmers Notepad [WinAVR] 开启程序编辑器，键入以下代码，保存在前面建立的 AVR_LED 文件夹中，并命名为 LED.C

```
#include "avr/io.h"
static void io_init(void)
{
    PORTA = 0xff ;
    DDRA = 0xff ;
}
void mDelay(uint16_t DelayTim)
{ uint16_t i ;
  for( ;DelayTim>0 ;DelayTim--)
  { for(i=0 ;i< ;40000 ;i++)
    { ;}
    for(i=0 ;i< ;39000 ;i++)
    { ;}
  }
}
```

```

int main(void)
{
uint8_t i=0 ;
io_init() ;
while(1)
{ if(++i>=8)
i=0 ;
PORTA=~(0x01< ;< ;i) ; //改为 0x80>>i 可改变流水灯方向
mDelay(500) ;
}
}

```

3.4 运行 WinAVR—>MFile[WinAVR]，编辑 Makefile，开启之后，即有一个 makefile 的模板，如图 1 所示。

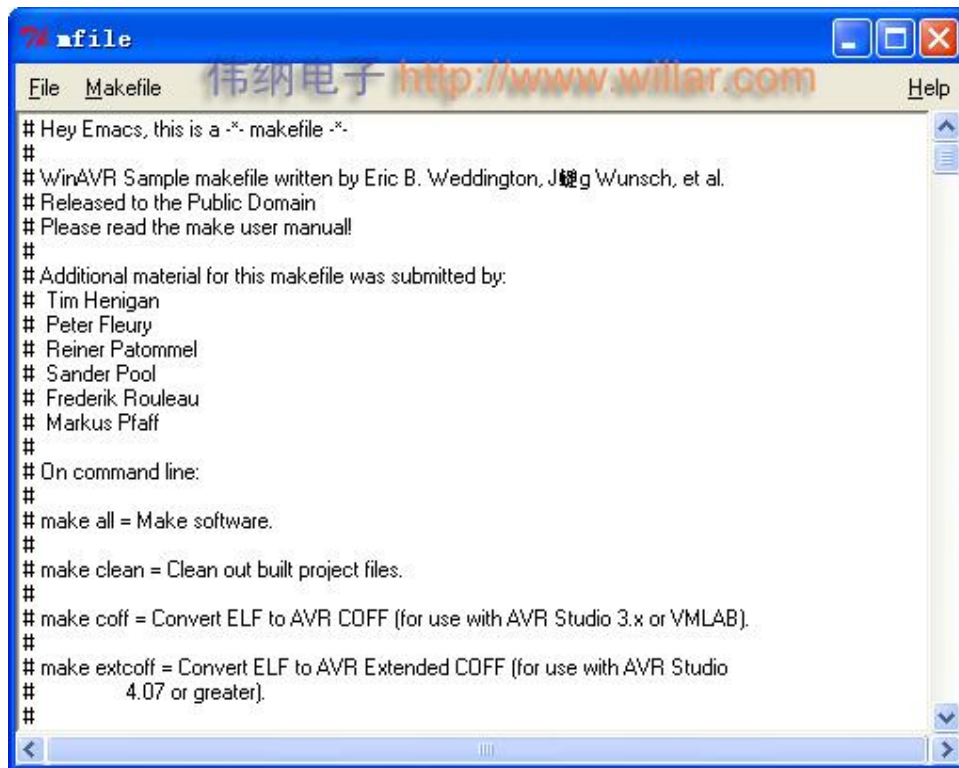


图 1 Makefile 的模板

点击菜单 Makefile->Main file name...即出现如图 2 所示的对话框，通常不需要更改此选项，最终产生的 hex 文件即 main.hex。



图 2 设置主文件名

接着点击 Makefile->Mcu type, 设定所用单片机的型号, 如图 3 所示。



图 3 设置单片机型号

我们选择 ATmega8515 为我们所用的型号。

3.5 点击: Make-->Debug Format: 可选择输出的调试文件格式, 应选择最后一项: ELF/DRARF-2 (AVR Studio ELF, GDB[experimental]), 这样可以由产生供 AVR studio 4 以上版本调试用的 ELF 格式的文件。

3.6 点击: Make-->C/C++ Source Files.....出现选择文件名的对话框, 如图 4 所示, 请用 “Choose file...” 钮选择你所要编译的文件, 注意, 一定要将 Include main.c 前面的勾去掉, 除非你的确有一个名为 main.c 的文件。

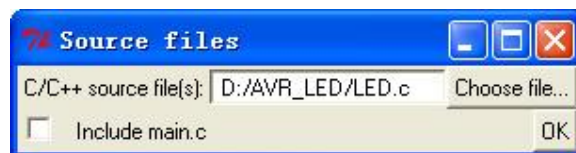


图 4 选择主文件

其他的项一般可以不必改动。

3.7 更改后用 File-->Save as...将该 makefile 文件保存至刚才所建文件夹中;

3.8 编译 (Make All):

回到 Programmer Notepads 中, 点击: Tools->Make All, 即可对该文件进行编译、链接等处理, 有关信息在 output 窗口中呈现。如有错误, 单击错误行, 即可定位, 以便对此进行修改, 直到编译、链接完全正确为止。此时, output 窗口显示: > Process Exit Code: 0 (图 5)

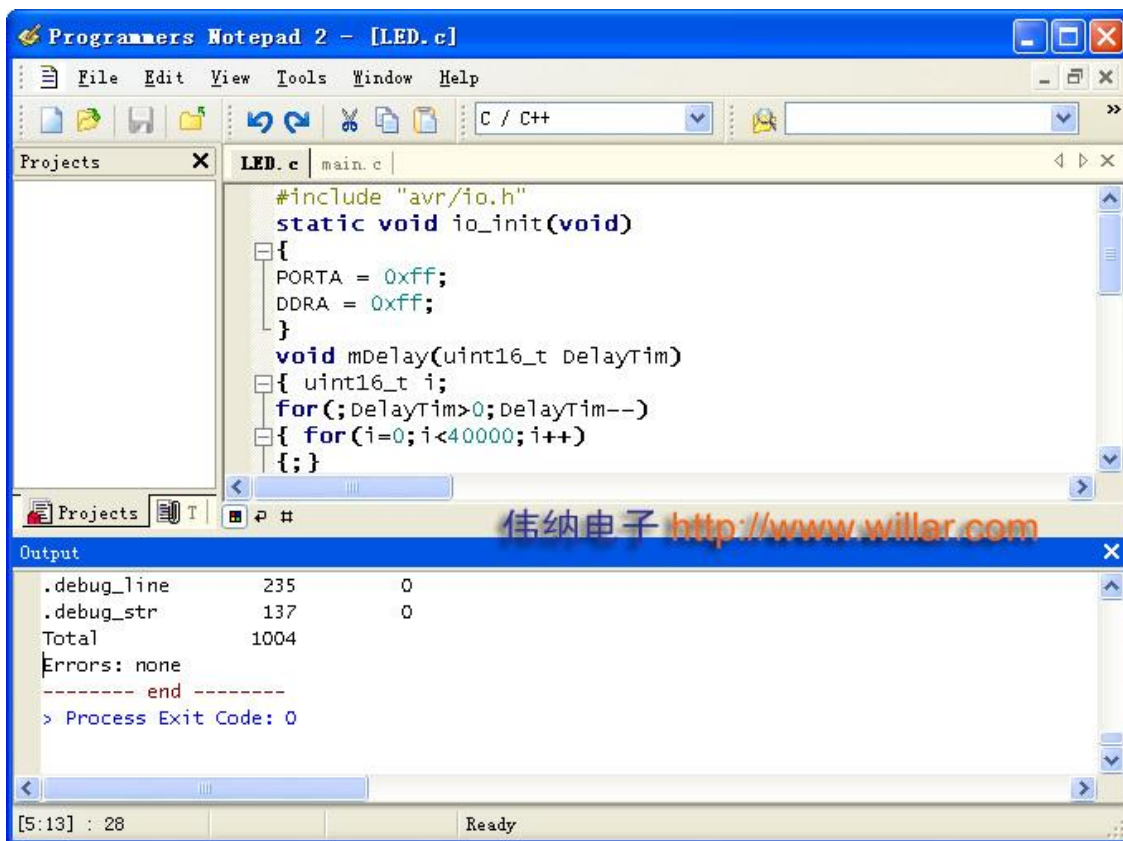


图 5

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 图 6 设置



图六 ME300 时钟源配置

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。时钟源的选择非常重要，请参考附表说明。如上图中设置的 CKSEL3..0 的值为：0100，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 8MHz

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 main.hex

4.5 烧写程序，正常即可观查到流水灯效果！

附表：Atmega8515/L 的熔丝配置说明（详细资料请参考官方 Datasheet）

表 1. 时钟源选择

芯片时钟选项	CKSEL3..0
外部晶体 / 陶瓷振荡器	1111 - 1010
外部低频晶体	1001
外部 RC 振荡器	1000 - 0101
标定的内部 RC 振荡器	0100 - 0001
外部时钟	0000

Note: 1. 对于所有的熔丝位, "1" 表示未编程, "0" 代表已编程。

默认时钟源:

芯片在出厂时 CKSEL = “0001”, SUT = “10”。默认时钟源为有最长启动时间的内部 RC 振荡器(频率 1M)。默认设置可以保证用户通过系统内或并行编程得到他们期望的时钟源。

标定的片内 RC 振荡器提供了固定的 1.0、2.0、4.0 或 8.0 MHz 的时钟。这些频率都是 5V、25° C 下的标称数值。这个时钟也可以作为系统时钟, 只要按照表 2 对熔丝位 CKSEL 进行编程即可。

表 2. 片内标定的 RC 振荡器工作模式

CKSEL3..0	标称频率 (MHz)
0001 ⁽¹⁾	1.0
0010	2.0
0011	4.0
0100	8.0

Note: 1. 出厂时的设置。