调试器/下载器：dwdebug

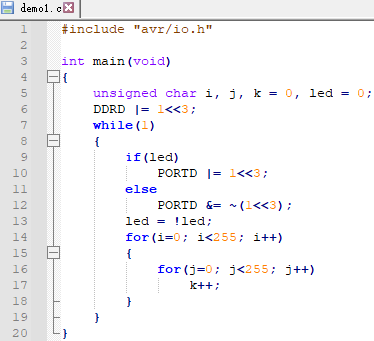
MCU：atmega328p

熔丝位：

|  |  |
| --- | --- |
| low | 0xe2 |
| high | 0x99 |
| extend | 0xff |

晶振：内部8MHz

第3页：



编译出目标文件：

avr-gcc -mmcu=atmega328p -c demo1.c

此时，生成demo1.o文件，即为目标文件。

链接目标文件至.elf文件：

avr-gcc -mmcu=atmega328p demo1.o -o demo1.elf

此时，生成demo1.elf文件。

将demo1.elf文件下载至单片机：

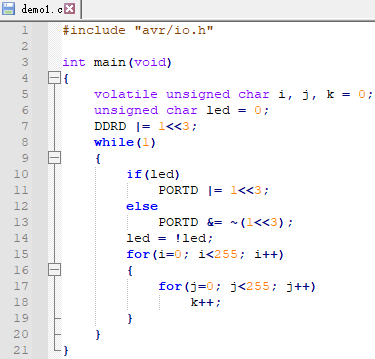
dwdebug device com6,l demo1.elf,qr

现象：PD3上的led灯快速闪烁。

注意1：avr-gcc默认是不开优化的，要想开优化，比如-Os优化，可以这样：

avr-gcc -mmcu=atmega328p -c -Os demo1.c

但是这样编译出的.elf文件下载到单片机后，led灯不再闪烁，原因是-Os优化将代码中的变量k给优化掉了，并且对i、j变量的执行也有所优化。为了避免这种情况，可将i、j、k变量定义时加上volatile关键词，即：



这样，用-Os优化时，程序正常执行。

注意2：

avr-gcc -mmcu=atmega328p -c demo1.c中，参数“-c”的含义：



avr-gcc -mmcu=atmega328p demo1.o -o demo1.elf中，参数“-o”含义：



从上面的解释可见，-o后面紧跟要生成的文件名。

若想生成.hex文件，可以这样：

avr-objcopy -j .text -j .data -O ihex demo1.elf demo1.hex

其中，-j的含义是：



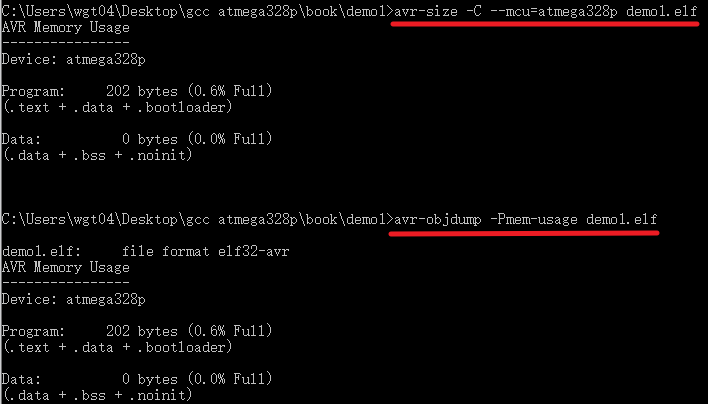
将-j后面指定的段复制到生成的文件中。avr-gcc编译出.elf文件时，将不同类型的数据以段的形式保存在.elf文件中，而与.hex文件相关的段为.text段和.data段，故要生成.hex文件时，只需从.elf文件中提取.text段和.data段，并将这两个段中的内容保存至.hex文件中。

-O的含义是：



-O后面紧跟输出格式，即ihex（intel HEX file）。

若想查看编译后程序的大小，有两种方法，即：

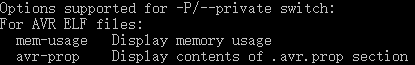


其中，avr-size的参数-C表示以AVR格式输出数据：



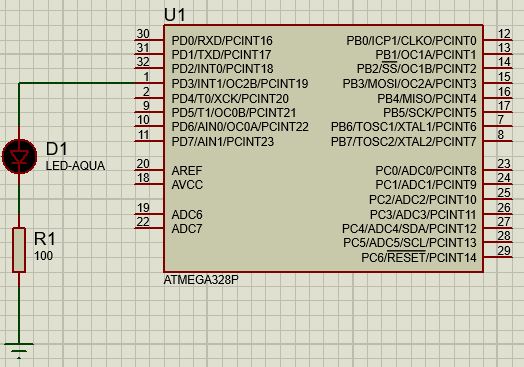
对于avr-objdump的参数-Pmem-usage表示：

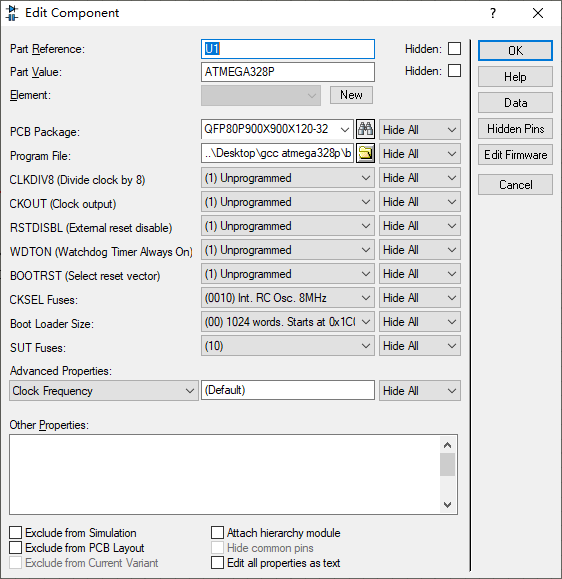




用avr-size.exe和avr-objdump.exe均能得到程序大小，只是avr-size.exe对新出的片子支持不太好，例如atmega328pb，就无法在输出的Device栏中显示atmega328pb，而是显示成unknown，而avr-objdump.exe就能正确显示成atmega328pb，故若想以AVR格式显示程序大小，建议使用avr-objdump.exe。

仿真：





注意熔丝位的设置，跟https://www.engbedded.com/fusecalc/网站的设置保持一致，不勾选表示1，即unprogrammed，而勾选表示0，即programmed。

选定program file为demo1.elf，运行仿真，可见跟实物现象一致。