实验 1　　ADS 1.2 集成开发环境练习

一、实验目的

了解 ADS 1.2 集成开发环境的使用方法。

二、实验设备

硬件：PC 机一台

软件：WindowsXP 系统，ADS 1.2 集成开发环境

三、实验内容

1. 建立一个新的工程；
2. 建立一个 C 源文件，并添加到工程中；
3. 设置编译连接控制选项；
4. 编译连接工程。

四、实验步骤

1. 启动 ADS1.2 IDE 集成开发环境，选择【File】->【New…】，使用 ARM Executable Image 工程模板建立一个工程，工程名称为 ADS，见图 1.1。

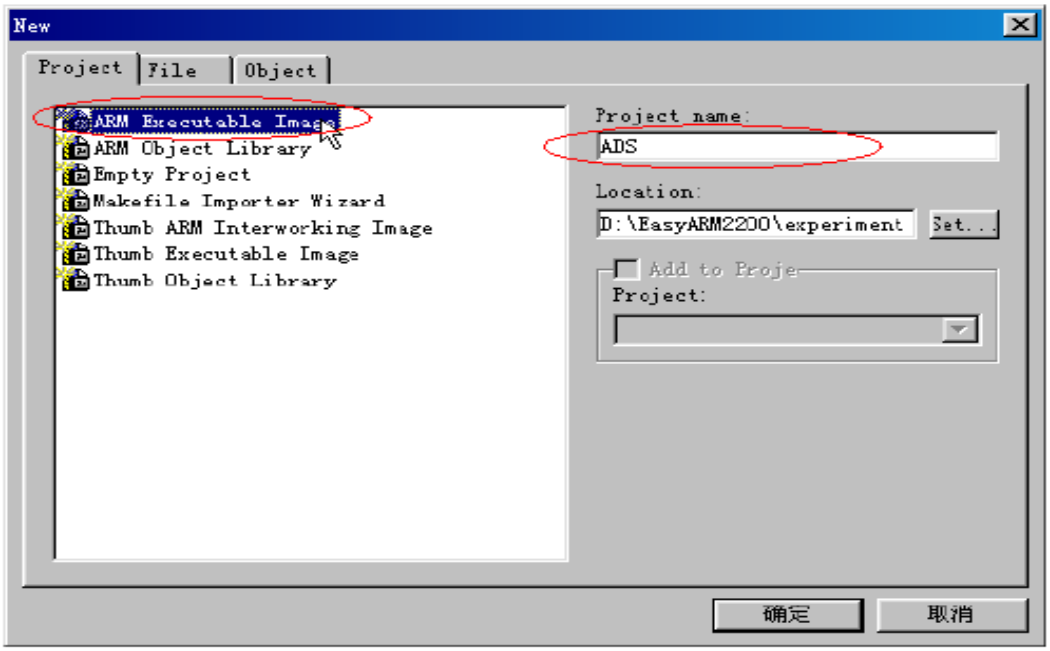

图 1.1　　建立 ARM 指令代码的工程

2. 选择【File】->【New…】建立一个新的文件 TEST1.S，设置直接添加到项目中，见图 1.2。输入如程序清单 1.1 所示的代码，并保存，见图 1.3。

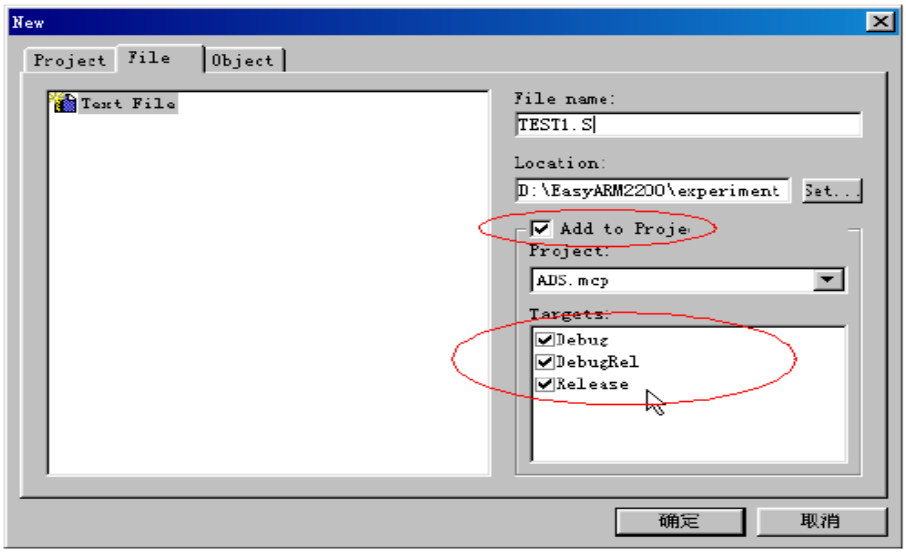

图 1.2 新建文件 TEST1.S

程序清单 1.1 TEST1.S 文件代码

```
        AREA    Example1,CODE,READONLY; 声明代码段 Example1
        ENTRY                           ; 标识程序入口
        CODE32                          ; 声明 32 位 ARM 指令
START   MOV     R0,#15                  ; 设置参数
        MOV     R1,#8
        ADDS    R0,R0,R1                ; R0 = R0 + R1
        B
        START
        END
```

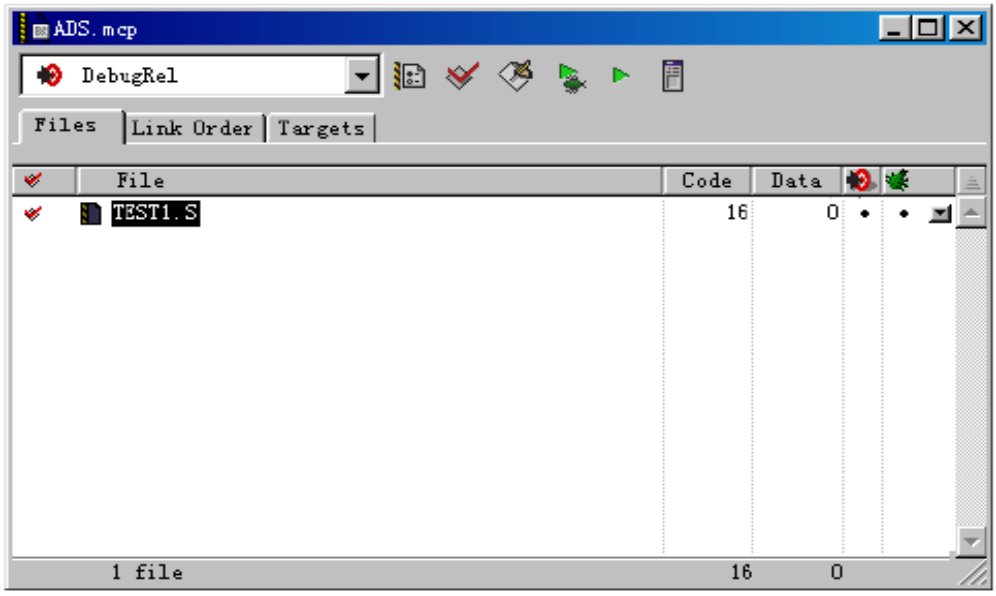

图 1.3 添加了 TEST1.S 的工程管理窗口

3. 选择【Edit】->【DebugRel Settings…】，在 DebugRel Settings 对话框的左边选择 ARMLinker 项，然后在 Output 页设置连接地址(见图 1.4)，在 Options 页设置调试入口地址(见图 1.5)。

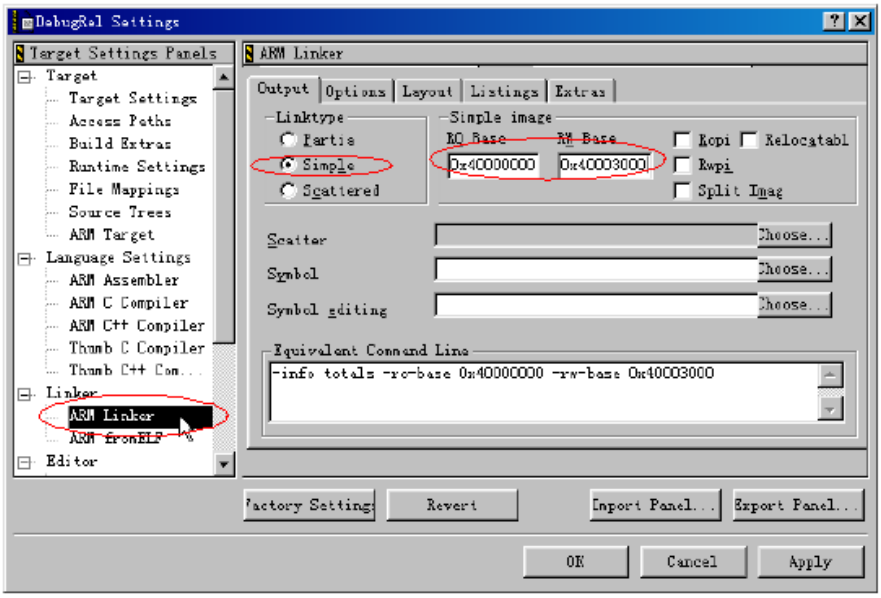

图 1.4 工程连接地址设置

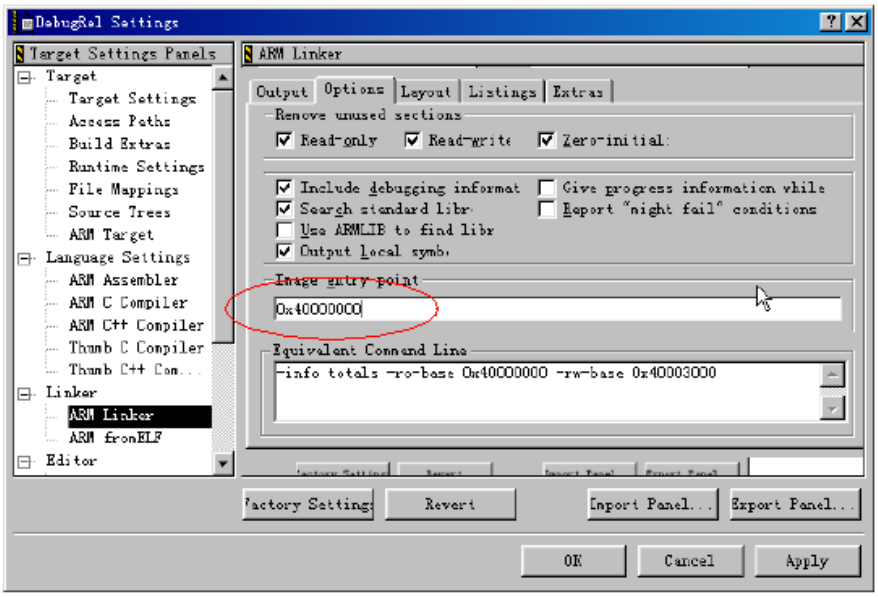

图 1.5 工程调试入口地址设置

4. 选择【Project】->【Make】，将编译连接整个工程。

五、思考

1. 工程模板有何作用？(提示：编译控制设置)

2. 如何强行重新编译工程的所有文件？(提示：选择【Project】->【Remove Object Code…】删除工程中的*.obj 文件)

实验一 汇编指令实验（一）

一、实验目的

1. 了解 ADS 1.2 集成开发环境及 ARMulator 软件仿真；

2．掌握 ARM7TDMI 汇编指令的用法，并能编写简单的汇编程序；

3. 掌握指令的条件执行和使用 LDR/STR 指令完成存储器的访问。

二、实验设备

硬件：PC 机一台

软件：WindowsXP 系统，ADS 1.2 集成开发环境

三、实验内容

使用 LDR 指令读取 0x40003100 上的数据，将数据加 1，若结果小于 10 则使用 STR 指令把结果写回原地址，若结果大于等于 10，则把 0 写回原地址。

使用 ADS 1.2 软件仿真，单步、全速运行程序，设置断点，打开寄存器窗口(ProcessorRegisters)监视 R0、R1 的值，打开存储器观察窗口(Memory) 监视 0x40003100 地址上的值。

四、实验步骤

1. 启动 ADS 1.2，使用 ARM Executable Image 工程模板建立一个工程 Instruction1。

2. 建立汇编源文件 TEST2.S，编写实验程序，然后添加到工程中。

3. 设置工程连接地址 RO Base 为 0x40000000，RW Base 为 0x40003000。设置调试入口地址 Image entry point 为 0x40000000。

4. 编译连接工程，选择【Project】->【Debug】，启动 AXD 进行软件仿真调试。

5. 打开寄存器窗口(Processor Registers)，选择 Current 项监视 R0、R1 的值。打开存储器观察窗口(Memory) 设置观察地址为 0x40003100，显示方式 Size 为 32Bit，监视 0x40003100 地址上的值。

说明：在 Memory 窗口中点击鼠标右键，Size 项中选择显示格式为 8Bit、16Bit、32Bit。如图 1.6 所示。

6. 可以单步运行程序，可以设置/取消断点，或者全速运行程序，停止程序运行，调试时观察寄存器和 0x40003100 地址上的值。运行结果见图 1.7。

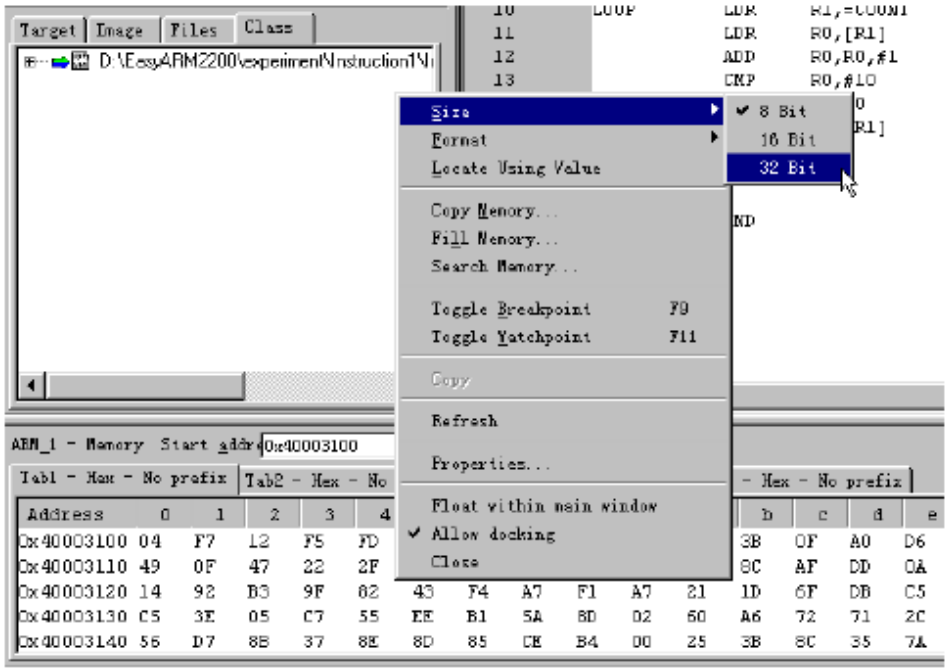

图 1.6　Memory 窗口显示格式设置

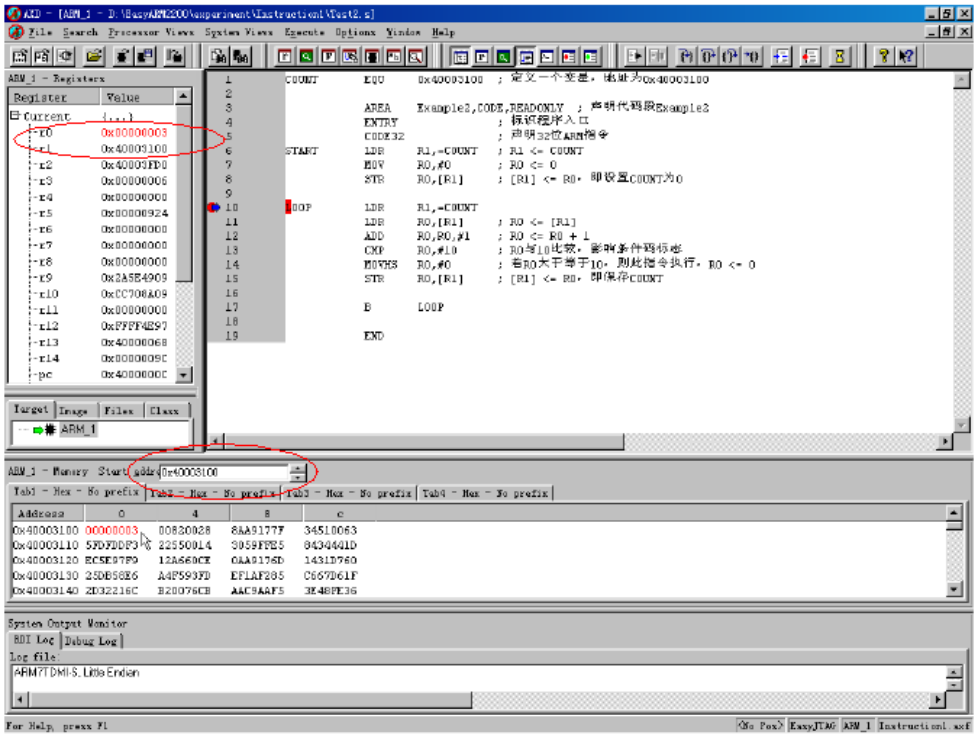

图 1.7　汇编实验程序运行结果

五、实验参考程序

汇编指令实验的参考程序见程序清单 1.2。

程序清单 1.2　汇编指令实验 1

```
COUNT   EQU   0x40003100                ; 定义一个变量，地址为 0x40003100
        AREA  Example2,CODE,READONLY ; 声明代码段 Example2
        ENTRY                          ; 标识程序入口
        CODE32                         ; 声明 32 位 ARM 指令
START   LDR     R1,=COUNT               ; R1 <= COUNT
        MOV     R0,#0                   ; R0 <= 0
        STR     R0,[R1]                ; [R1] <= R0，即设置 COUNT 为 0
LOOP    LDR     R1,=COUNT
        LDR     R0,[R1]                ; R0 <= [R1]
        ADD     R0,R0,#1                ; R0 <= R0 + 1
        CMP     R0,#10                  ; R0 与 10 比较，影响条件码标志
        MOVHS   R0,#0              ; 若 R0 大于等于 10，则此指令执行，R0 <= 0
        STR     R0,[R1]            ; [R1] <= R0，即保存 COUNT
        B       LOOP
END
```

六、　思考

（1）若使用 LDRB/STRB 代替程序清单 1 中的所有加载/存储指令(LDR/STR)，程序会得到正确的执行吗？

（2）LDR 伪指令与 LDR 加载指令的功能和应用有何区别，举例说明？(提示：LDR 伪指令的形式为“LDR Rn,=expr”)

（3）在 AXD 调试时如何复位程序？(提示：选择【File】->【Reload Current Image】重新加载映象文件)。