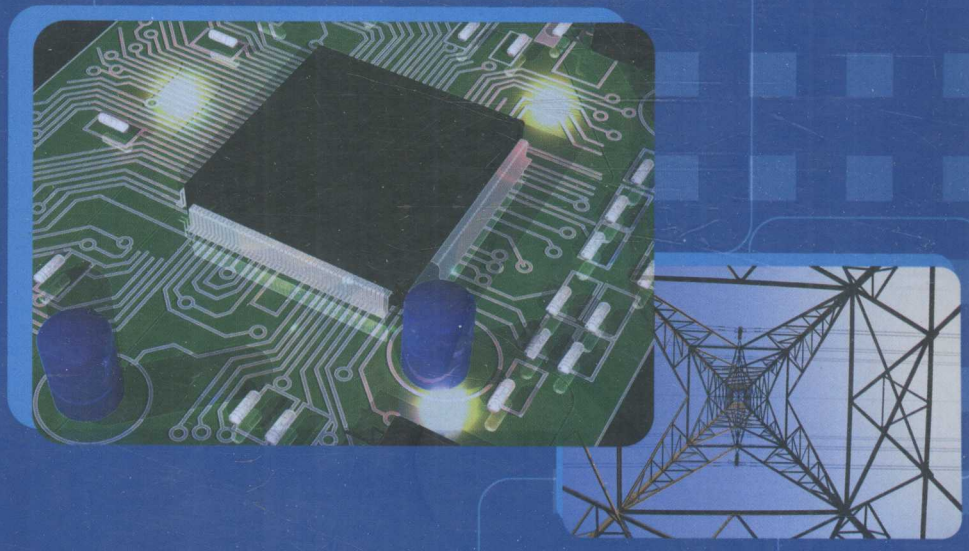
ARM

系统实践教程

主编 赵宏伟 刘萍萍 周国梁



吉林大学出版社

ARM 系统实践教程

主编　赵宏伟　刘萍萍　周国梁

吉林大学出版社

图书在版编目（CIP）数据

ARM 系统实践教程/赵宏伟，刘萍萍，周国梁主编.
长春：吉林大学出版社，2008.3
ISBN 978-7-5601-3811-4

Ⅰ.A… Ⅱ.①赵… ②刘… ③周… Ⅲ.微处理器，ARM-系统设计-教材 Ⅳ.TP332

中国版本图书馆 CIP 数据核字（2008）第 026207 号

书　名：ARM 系统实践教程
作　者：赵宏伟　刘萍萍　周国梁　主编

责任编辑、责任校对：邵宇彤
吉林大学出版社出版、发行
开本：880×1230 毫米　1/16
印张：8.75　字数：264 千字
ISBN 978-7-5601-3811-4

封面设计：创意广告
长春市东文印刷厂印刷
2008 年 3 月 第 1 版
2008 年 3 月 第 1 次印刷
定价：20.00 元

社址：长春市明德路 421 号　邮编：130021
发行部电话：0431-88499826
网址：http://www.jlup.com.cn
E-mail:jlup@mail.jlu.edu.cn

前　言

本书是针对MagicARM2200教学实验平台的实验教材。MagicARM2200教学实验平台配套了周立功等编写的《ARM嵌入式系统实验教程(三)》和《ARM嵌入式系统实验教程(三)附加实验》,但每个实验平台一套,这使得众多学生预习和操作实验变得有些困难。考虑到方便学生实验和日新月异的嵌入式系统创新设计,我们编写了这本教材。在教材编写中,我们选取了一些实验平台配套教材中比较典型的实例作为基础性实验,另外设计了一些提高性实验,尤其是以吉林大学本科生研究机会计划项目为基础,设计了综合性实验和开发性实验,为学生的嵌入式系统创新设计提供了思路和实例。

本书包括十三个实验项目。实验一涉及ADS 1.2集成开发环境和LPC2200专用工程模板的使用。实验二涉及ADS 1.2集成开发环境下ARM的汇编语言程序设计和C语言程序设计。实验三涉及GPIO控制蜂鸣器、键盘及LED。实验四涉及微控制器LPC2290的外部中断和定时器。实验五涉及微控制器LPC2290的UART串行通信。实验六涉及A/D和数码管的设计和使用。实验七涉及液晶屏和触摸屏的接口和使用。实验八涉及μCLinux操作系统环境搭建与应用。实验一至实验八包括基础性实验和提高性实验两个部分。实验九是综合性实验,使用A/D、LED、蜂鸣器等设备。实验十是综合性实验,使用键盘、蜂鸣器、直流电机、步进电机、LED灯等设备。实验十一是开发性实验,涉及SPI接口和NRF905无线通讯模块。实验十二是开发性实验,涉及键盘与LED驱动芯片ZLG7290和数码管、键盘等设备。实验十三是开发性实验,涉及μC/OS－II操作系统搭建,GPRS PACK开发板,I^2C总线,蜂鸣器、LED灯、LCD液晶显示器、步进电机、数码管、键盘等终端。

本书编写过程中,嵌入式系统课程教学梯队的教师们做了很多工作,尤其是研究生都虹云、房柯池、王鹏、徐方艾、赵鹏、宋波涛等同学做了大量的实验验证工作,为本书的顺利出版提供了强有力的支持和帮助,在此,向这些同学表示真心感谢。同时,还要感谢嵌入式系统实验室张仲明等老师们提供的良好的实验环境和资源,感谢吉林大学计算机学院和软件学院教务办公室的老师和领导为本教材顺利出版提出的建议和作出的努力。

本书编写过程中,参考了有关的优秀教材、专著、应用成果,以及优秀的网络站点,恕不一一列举。能够领略众多新颖的观点和技术,是原创者的无私贡献,是读者的集粹之想。本书编者在此向提供各种观点和技术的各位编著者表示最真诚的谢意。

我们对在本书编写、出版过程中给予支持和帮助的所有领导及朋友们,一并表示衷心的感谢。

由于编著者水平和经验所限,书中不足之处在所难免,恳请指正。

编　者

2008年1月于长春

目　录

实验一　嵌入式系统软硬件开发环境

预习要求：

(1)阅读《ADS 集成开发环境及 EasyJTAG 仿真器应用》，了解 ADS 工程编辑的内容。

(2)了解 MagicARM2200 教学实验开发平台的硬件结构。

(3)了解 LPC2200 专用工程模板、EasyJTAG 仿真器的应用。

一、基础性实验

1. 实验目的

熟悉 ADS 1.2 集成开发环境的使用方法。

2. 实验设备

(1)硬件：PC 机一台。

(2)软件：Windows98/XP/2000 系统，ADS 1.2 集成开发环境。

3. 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)设置文本编辑器支持中文。

(4)设置编译连接控制选项。

(5)编译连接工程。

(6)调试工程。

4. 实验步骤

(1)启动 ADS 1.2 集成开发环境，点击 WINDOWS 操作系统的[开始]->[程序]->[ARM Developer Suite v1.2]->[CodeWarrior for ARM Developer Suite]启动 Metrowerks CodeWarrior 或双击“CodeWarrior for ARM Developer Suite”快捷方式启动。

(2)选择[File]->[New...]，在 Project 标签中，使用 ARM Executable Image 工程模板建立一个工程。然后在[Location]项选择工程存放路径，并在[Project name]项输入工程名称，点击[确定]按钮即可建立相应工程，工程文件名默认后缀为 mcp。

(3)在主框架下，点击[File]菜单，选择[New...]，选择 File 标签，输入文件全名(如汇编源文件为 test.s，C 语言源文件为 test.c)建立一个新文件。注意将文件保存到相应工程的目录下。

(4)在工程窗口中[Files]页空白处点击鼠标右键，选择“Add Files...”，在弹出的对话框中，选择相应的源文件，点击[打开]按钮后，在弹出的对话框中单击[OK]，如需要对文件进行编辑时，在工程对话框中双击要编辑的文件即可。

(5)选择[Edit]->[DebugRel Settings...]，在 DebugRel Settings 对话框的左边单击 ARM Linker 项，然后在 Output 页，“RO Base”和“RW Base”处修改连接地址，见图 1.1，在 Options 页，“Image entry point”处设置调试入口地址，见图 1.2。

(6)选择[Project]->[Make]，将编译连接整个工程，如果编译成功，Errors&Warnings 对话框会报告编译错误为 0，那么就可以对工程进行仿真。

(7)选择[Project]->[Debug]，IDE 环境就会自动启动 AXD 调试软件。

注意：

(1)由于 ADS 安装以后默认字体是 Courier New，对于中文支持不完善，因此建议修改字体。

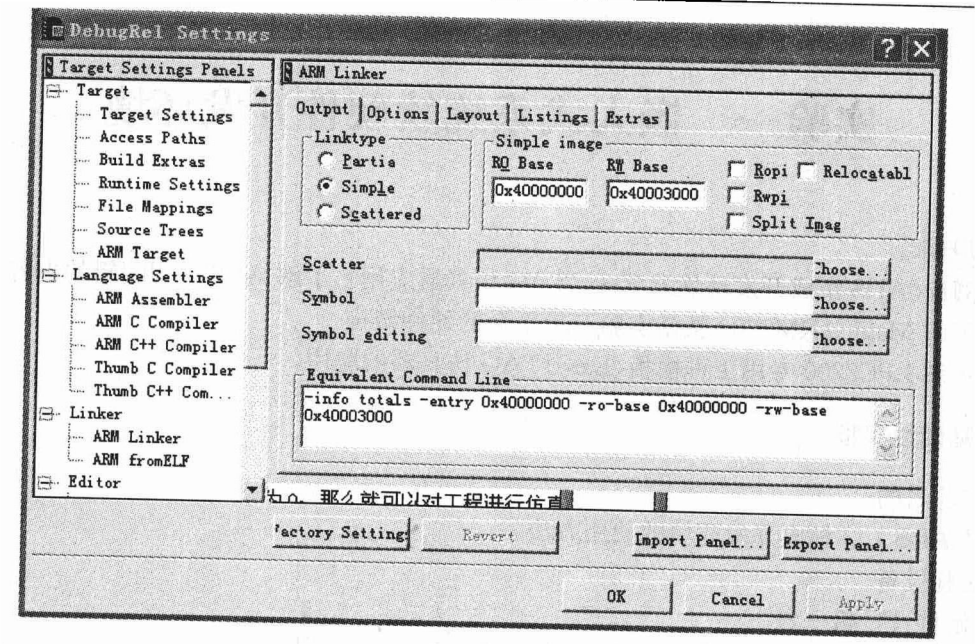


图 1.1　工程连接地址设置

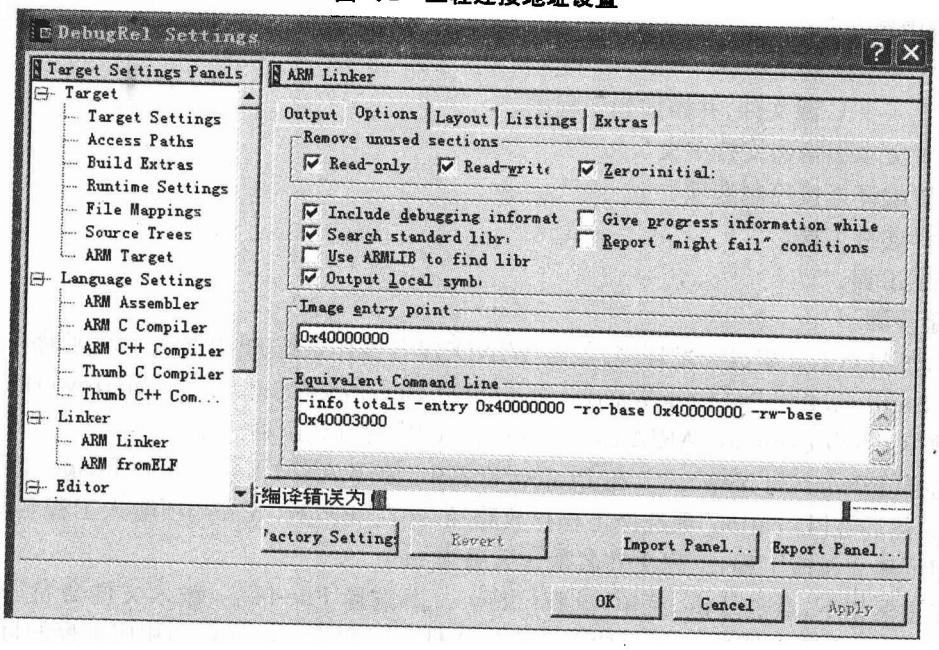


图 1.2　工程调试入口地址设置

选择[Edit]－>[Perferences...]，在 Font 选项设置字体是 Fixedsys，Script 是 CHINESE_GB2312。由于 Tab 在不同文本编辑器解释不同，建议在 Tab Inserts Spaces 前打勾，使 Tab 键插入的是多个空格。

(2)在 AXD 调试软件中断点调试方法：在要设置断点的行，双击鼠标即可，如果出现红色实心圆点，那么表示断点设置成功，然后选择[Execute]－>[Go]全速运行，可以发现程序会在断点处停止(蓝色箭头为程序当前执行到的位置)。通过断点调试可以观察 ARM 寄存器数值变化。双击断点则可以取消断点设置。

(3)在 AXD 中观察各寄存器值的调试方法：选择[Processor views]－>[Registers]，打开寄存器窗口，监视 R0～R14、PC、CPSR、SPSR 的值。

(4)在 AXD 中观察存储器值的方法：[Processor views]－>[Memory]，打开存储器观察窗口，

设置观察地址即可监视地址上的值。显示方式可以改变，在 Memory 窗口中点击鼠标右键，选择显示格式为 8Bit、16Bit 或 32Bit。地址上的值也可以改变，双击要改变的内容即可输入新值。

5. 实验参考程序

```
        AREA   Example,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
```

6. 实验思考题

(1)工程模板有什么作用?

(2)如何强行重新编译工程的所有文件?

二、提高性实验

1. 实验目的

(1)掌握 LPC2200 专用工程模板的使用。

(2)掌握 EasyJTAG 仿真器的安装和使用。

(3)能够在 MagicARM2200 教学实验开发平台上运行第一个程序(无操作系统)。

(4)通过实验了解使用 ADS1.2 编写 C 语言程序，并进行调试。

2. 实验设备

(1)硬件：PC 机一台，MagicARM2200 教学实验开发平台一套。

(2)软件：Windows98/XP/2000 系统，ADS 1.2 集成开发环境。

3. 实验内容

利用 ARM Executable Image for MagicARM2200 工程模板，编写一个 C 语言程序文件。C 程序使用加法来计算 1+2+3+…+(N−1)+N 的值(N>0)。

4. 实验步骤

(1)使用 LPC2200 专用工程模板建立工程

启动 ADS1.2 IDE，点击[File]菜单，选择[New...]即弹出 New 对话框。使用 ARM Executable Image for MagicARM2200 工程模板建立一个工程。

说明：由于事先增加了 LPC2200(for MagicARM2200)专用工程模板，所以在工程模板中多了几项工程模板选项。如果需要，自己重新添加模板，可以按照如下步骤：将“MagicARM2200 Project module”目录下(假设模板文件存放在这个目录)所有文件和目录拷贝到“<ADS1.2 安装目录>\Stationery\”即可。这个步骤只需 1 次，以后就可以直接使用工程模板了。

LPC2200 专用工程模板一般包含的设置信息有 FLASH 起始地址为 0x00000000、片内 RAM 起始地址 0x40000040、片外 RAM 起始地址为 0x81000000、编译连接选项及编译优化级别等等；模板中包含了 LPC2200 系列 ARM7 微控制器的启动文件，包括 STARTUP.S、TARGET.C；模板还包含了 LPC2200 系列 ARM7 微控制器的头文件、分散加载描述文件。

(2)在 user 组中的 main.c 中编写主程序代码。(提示：变量类型 uint32)。

(3)选用 DebugInExram 生成目标，然后编译连接工程。

说明：LPC2200 专用工程模板各生成目标的配置。

生成目标	分散加载描述文件	调试入口点地址	应用说明
DebugInExram	Mem_b. scf	0x81000000	片外 RAM 调试模式，程序在片外 RAM
DebugInChipFlash	Mem_c. scf	0x00000000	片内 FlASH 调试模式，程序在片内 FLASH 中
RelInChip	Mem_c. scf	0x00000000	片内 FLASH 工作模式，程序在片内 FLASH 中。程序写入芯片后芯片即被加密
RelOutChip	Mem_a. scf	0x80000000	片外 FLASH 工作模式，程序在片外 FLASH 中

(4)使用 EasyJTAG 仿真器

在 AXD 调试环境，打开【Options】->【Configure Target...】，弹出 Choose Target 窗口，在“Target Environments”框中选择“EasyJTAG...”项。点击“Configure”按钮，进入“EasyJTAG Setup”设置窗口。在“ARMcore”项中选取 CPU 类型 LPC2290，在“Halt Mode”项中选择 Halt program。然后点击“OK”，再点击“OK”，此时 EasyJTAG 将会进行连接(目标板)的操作。若连接成功，则目标板上的 LPC2000 系列芯片由 EasyJTAG 控制，原先运行的程序被停止。

说明：由于事先已经装好 EasyJTAG 仿真器，不需要自己操作。如果需要重新安装 EasyJTAG 仿真器，方法如下：

首先，将 EasyJTAG 仿真器的驱动程序复制到 ADS 的 BIN 目录，如 C:\Program Files\ARM\ADSv1_2\BIN(假设 ADS 的安装路径)。

接着，将 EasyJTAG 仿真器的 25 针接口通过并口延长线与 PC 机的并口连接，先给 MagicARM2200 实验箱供电，再将 EasyJTAG 仿真器的 20 针接口通过 20 PIN 连接电缆接到 MagicARM2200 实验箱主板的 J3 上。

然后，点击 Windows 系统的【开始】->【程序】->【ARM Developer Suite v1. 2】->【AXD Debugger】进入 AXD 调试环境，打开【Options】->【Configure Target...】，弹出 Choose Target 窗口。点击“ADD”添加仿真器的驱动程序，在添加文件窗口选择如 C:\ProgramFiles\ARM\ADSv1_2\BIN 目录下的 EasyJTAG. dll，点击“打开”即可。

注意：有时，AXD 会弹出错误对话框，如图 1. 3 所示，此时可以点击“Connect mode...”，然后选择“ATTACH ...”项确定，再点击“Restart”。若 EasyJTAG 正确连接目标板，AXD 代码窗口将显示空白，接下来就可以使用【File】->【Load Image...】加载调试文件，进行 JTAG 调试。

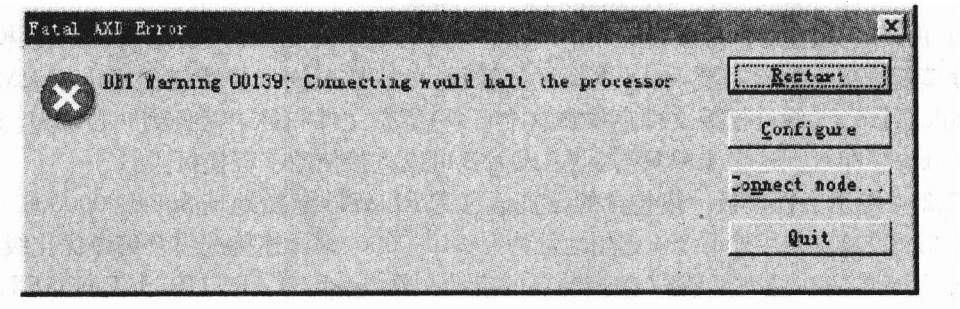


图 1.3 Fatal AXD 错误提示

(5)全速运行程序。

5. 实验参考程序

#define N 100

```
uint32 sum;
void main(void)
{
uint32 i;
sum = 0;
for( i = 0; i <= N; i++ )
{
sum += i;
}
while(1);
}
```

6. 实验思考题

(1)为什么这个实验的工程不需要设置连接地址?

(2)在 AXD 调试时如何复位程序?

实验二　汇编语言程序设计

预习要求：

(1)仔细阅读 ARM 指令系统内容。

(2)了解 ADS 工程编辑和 AXD 调试的内容(本次实验使用软件仿真)。

一、基础性实验

1. 实验目的

(1)熟悉 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。

(4)掌握断点调试，观察 ARM 寄存器数值变化。

2. 实验设备

(1)硬件：PC 机一台。

(2)软件：Windows98/XP/2000 系统，ADS 1.2 集成开发环境。

3. 实验内容

(1)使用 LDR 指令读取 0x40003100 上的数据，将数据加 1，若结果小于 8 则使用 STR 指令把结果写回原地址，若结果大于等于 8，则把 0 写回原地址。然后再次读取 0x40003100 上的数据，将数据加 1，判断结果是否小于 8……一直循环。

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

4. 实验步骤

(1)启动 ADS1.2，使用 ARM Executable Image 工程模板建立一个工程。

(2)建立汇编源文件 TEST.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。

注意：在 AXD 调试环境，打开【Options】->【Configure Target...】，弹出 Choose Target 窗口，在“Target Environments”框中选择“ARMulator...”项，进行 ARMulator 软件仿真。

(5)打开存储器观察窗口(Memory)设置观察地址为 0x40003000，显示方式为 32Bit，监视 0x40003000 地址上的值。

(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 0x40003000 地址上的值。

5. 实验参考程序

```
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
```

6. 实验思考题

(1)LDR 伪指令与 LDR 加载指令的功能和应用有何区别?

(2)若使用 LDRB/STRB 代替程序中的所有加载/存储指令(LDR/STR),程序会得到正确执行么?

(3)LDR/STR 指令的前索引偏移指令如何编写?指令是怎样操作的?

二、设计性实验

1. 实验目的

了解子程序编写及调用,实现结构化程序编程。

2. 实验设备

(1)硬件:PC 机一台。

(2)软件:Windows98/XP/2000 系统,ADS 1.2 集成开发环境。

3. 实验内容

编写汇编语言程序,实现以下 C 程序段的功能:

```
uint8 function(uint8 cc)
{
return (cc * 2);
}
void main(void)
{
  uint8 x = 1;
  uint8 y = 128;
while(1)
{
  x = 1;
while(x<y)
  {
    x = function(x);
}
}
}
```

4. 实验步骤

(1)启动 ADS1.2,使用 ARM Executable Image 工程模板建立一个工程。

(2)建立汇编源文件 TEST.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)全速运行。

5. 实验思考题

(1)指令 MOV R0,#0x12345678 是否正确,为什么?

(2)使用 ARM 汇编指令结构化程序编程,如何在 for、while 结构中实现 break、continue?

6. 实验参考程序

```
X       EQU     1
Y       EQU     128
        AREA    Lg,CODE,READONLY
        ENTRY
        CODE32
START   LDR     SP,=0x40003F00;设置堆栈
        LDR     R0,=X
        LDR     R1,=Y
LOOP    CMP     R0,R1
        LDRHI   R0,=X
        BL      SUB             ;调用子程序,返回值为 R0
        B       LOOP
SUB
        STMFD   SP!,{R0-R7,LR}
        MOV     R0,R0,LSL #1
        LDMFD   SP!,{R0-R7,PC}
        END
```

实验三　GPIO 特性实验

预习要求：

(1)仔细阅读参考文献中 LPC2200 的引脚连接模块、GPIO。

(2)了解 ADS1.2 集成开发环境、LPC2200 专用工程模板、EasyJTAG 仿真器的应用。

(3)了解蜂鸣器、键盘及 LED 工作原理。

一、基础性实验

1. 实验目的

(1)掌握 LPC2200 专用工程模板的使用。

(2)掌握 EasyJTAG 仿真器的安装和使用。

(3)熟练掌握 LPC2200 系列 ARM7 控制器的 GPIO 控制。

2. 实验设备

(1)硬件：PC 机一台。MagicARM2200 教学实验开发平台一套。

(2)软件：windows98/XP/2000 操作系统，ADS1.2 集成开发环境。

3. 实验内容

利用 GPIO 的输出特性控制蜂鸣器报警。使用片外 RAM(MT45W4MW16 芯片)进行调试。

每一个 I/O 口线可单独设置为输入/输出模式；有单独控制 I/O 输出的置位或清零寄存器；所有 I/O 口在复位后默认为输入。LPC2200 有 4 个 32 位的通用 I/O 口。

GPIO 寄存器意义如下：

(1)IOxPIN：GPIO 引脚值寄存器，不管方向如何，引脚的当前状态都可以从该寄存器中读出，只读。

IOxPIN(31：0)：GPIO 引脚值。IOxPIN[0]对应于 Px.0…IOxPIN[31]对应于 Px.31 引脚。复位值未定义。

(2)IOxDIR：GPIO 方向控制寄存器。该寄存器单独控制每个 IO 口的方向，可读/写。

IOxDIR(31：0)：IOxDIR[0]对应于 Px.0…IOxDIR[31]对应于 Px.31 引脚。复位值未定义。

(3)IOxSET：GPIO 输出置位寄存器。该寄存器控制引脚输出高电平，读/置位。

IOxSET(31：0)：IOxSETv[0]对应于 Px.0…IOxSET[31]对应于 Px.31 引脚。复位值未定义。

(4)IOxCLR：GPIO 输出清零寄存器。该寄存器控制引脚输出低电平，只清零。

IOxCLR(31：0)：IOxCLR[0]对应于 Px.0…IOxCLR[31]对应于 Px.31 引脚。复位值未定义。

蜂鸣器电路如图 3.1 所示。JP22 跳线器用于设置是否连接蜂鸣器电路，当 JP22 短接时，当 P0.7 输出低电平时，蜂鸣器蜂鸣；当控制 P0.7 输出高电平时，蜂鸣器停止蜂鸣。跳线器断开即可断开与 P0.7 的连接。

LPC2200 内部的引脚连接模块可以使同一个引脚具有多种功能，即引脚复用，通过配置相关寄存器控制多路开关来连接引脚与片内外设。

外设在激活和任何相关中断使能之前必须连接到适当的引脚。任何使能的外设功能如果没有映射到相关的引脚，则被认为是无效的。

引脚连接模块包含 3 个寄存器：PINSEL0、PINSEL1、PINSEL2。引脚功能选择寄存器 0(PINSEL0－0xE002C000)功能见表 3.1，引脚功能选择寄存器 1(PINSEL1－0xE002C004)功能见表 3.2，引脚功能选择寄存器 2(PINSEL2－0xE002C014)功能见表 3.3。

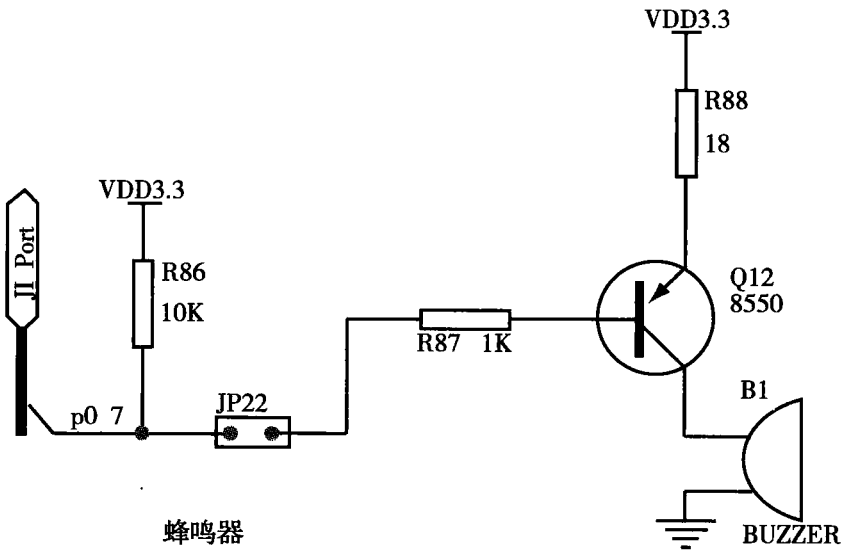


图 3.1 蜂鸣器电路

表 3.1 引脚功能选择寄存器 0(PINSEL0)

PINSEL0	引脚名	00	01	10	11	复位值
1：0	P0.0	GPIO P0.0	TxD(UART0)	PWM1	保留	00
3：2	P0.1	GPIO P0.1	RxD(UART0)	PWM3	EINT0	00
5：4	P0.2	GPIO P0.2	SCL(I2C)	捕获 0.0(TIMER0)	保留	00
7：6	P0.3	GPIO P0.3	SDA(I2C)	匹配 0.0(TIMER0)	EINT1	00
9：8	P0.4	GPIO P0.4	SCK(SPI0)	捕获 0.1(TIMER0)	保留	00
11：10	P0.5	GPIO P0.5	MISO(SPI0)	匹配 0.1(TIMER0)	保留	00
13：12	P0.6	GPIO P0.6	MISI(SPI0)	捕获 0.2(TIMER0)	保留	00
15：14	P0.7	GPIO P0.7	SSEL(SPI0)	PWM2	EINT2	00
17：16	P0.8	GPIO P0.8	TxD(UART1)	PWM4	保留	00
19：18	P0.9	GPIO P0.9	RxD(UART1)	PWM6	EINT3	00
21：20	P0.10	GPIO P0.10	RTS(UART1)	捕获 1.0(TIMER1)	保留	00
23：22	P0.11	GPIO P0.11	CTS(UART1)	捕获 1.1(TIMER1)	保留	00
25：24	P0.12	GPIO P0.12	DSR(UART1)	匹配 1.0(TIMER1)	保留	00
27：26	P0.13	GPIO P0.13	DTR(UART1)	匹配 1.1(TIMER1)	保留	00
29：28	P0.14	GPIO P0.14	CD(UART1)	EINT1	保留	00
31：30	P0.15	GPIO P0.15	RI(UART1)	EINT2	保留	00

表 3.2 引脚功能选择寄存器 1(PINSEL1)

PINSEL0	引脚名	00	01	10	11	复位值
1：0	P0.16	GPIO P0.16	EINT0	匹配 0.2 (TIMER0)	保留	00
3：2	P0.17	GPIO P0.17	捕获 1.2 (TIMER1)	SCK(SPI1)	匹配 1.2 (TIMER1)	00

5：4	P0.18	GPIO P0.18	捕获 1.3 (TIMER1)	MISO(SPI1)	匹配 1.3 (TIMER1)	00
7：6	P0.19	GPIO P0.19	匹配 1.2 (TIMER1)	MISI(SPI1)	匹配 1.3 (TIMER1)	00
9：8	P0.20	GPIO P0.20	匹配 1.3 (TIMER1)	SSEL(SPI1)	EINT3	00
11：10	P0.21	GPIO P0.21	PWM5	保留	捕获 1.3 (TIMER1)	00
13：12	P0.22	GPIO P0.22	保留	捕获 0.0 (TIMER0)	匹配 0.0 (TIMER0)	00
15：14	P0.23	GPIO P0.23	保留	保留	保留	00
17：16	P0.24	GPIO P0.24	保留	保留	保留	00
19：18	P0.25	GPIO P0.25	保留	保留	保留	00
21：20	P0.26	GPIO P0.26	保留			00
23：22	P0.27	GPIO P0.27	AIN0 (A/D 转换器)	捕获 0.1(TIMER0)	匹配 0.1 (TIMER0)	01
25：24	P0.28	GPIO P0.28	AIN1 (A/D 转换器)	捕获 0.2(TIMER0)	匹配 0.2 (TIMER0)	01
27：26	P0.29	GPIO P0.29	AIN2 (A/D 转换器)	捕获 0.3(TIMER0)	匹配 0.3 (TIMER0)	01
29：28	P0.30	GPIO P0.30	AIN3 (A/D 转换器)	EINT3	捕获 0.0 (TIMER1)	01
31：30	P0.31	GPIO P0.31	保留			00

表 3.3　引脚功能选择寄存器 2(PINSEL2)

PINSEL2	描述	复位值
1：0	保留	00
2	该位为 0 时，P1.31：26 用作 GPIO 该位为 1 时，P1.31～P1.26 用作一个调试端口	$\overline{\text{P1.26}}$/RTCK
3	该位为 0 时，P1.25：16 用作 GPIO 该位为 1 时，P1.25～P1.16 用作一个跟踪端口	$\overline{\text{P1.20}}$/ $\overline{\text{TRACESYNC}}$
5：4	控制数据总线和选通引脚的使用： 引脚 P2.7～P2.0　11＝ P2.7～P2.0　0x 或 10＝D[7：0] 引脚 P1.0　11＝P1.0　0x 或 10＝CS0 引脚 P1.1　11＝P1.1　0x 或 10＝OE 引脚 P3.31　11＝P3.31　0x 或 10＝BLS0 引脚 P2.15～P2.8　00 或 11＝ P2.15～P2.8　01 或 10＝D[15：8] 引脚 P3.30　00 或 11＝ P3.30　01 或 10＝BLS1 引脚 P2.27～P2.16　0x 或 11＝ P2.27～P2.16　10＝D[27：16] 引脚 P2.29～P2.18　0x 或 11＝ P2.29～P2.18　10＝D[29：18] 引脚 P2.31～P2.30　0x 或 11＝ P2.31～P2.30 或 AIN[5：4]　10＝D[31：30] 引脚 P3.29～P3.28　0x 或 11＝ P3.29～P3.28 或 AIN[6：7]　10＝BLS[2：3]	BOOT[1：0] (如 BOOT[1：0]＝01，该域的复位值就为 01)

6	如果位[5：4]不为10，由该位控制P3.29脚的使用：为0时，使能P3.29；为1时，使能AIN6	1
7	如果位[5：4]不为10，由该位控制P3.28脚的使用：为0时，使能P3.28；为1时，使能AIN7	1
8	该位控制P3.27脚的使用：为0时，使能P3.27；为1时，使能WE	0
10：9	保留	—
11	该位控制P3.26脚的使用：为0时，使能P3.26；为1时，使能CS1	0
12	保留	—
13	如果[27：25]不为111，有该位控制P3.23/A23/XCLK脚的使用：为0时，使能P3.23；为1时，使能XCLK	0
15：14	控制P3.25脚的使用：00使能P3.25，01使能CS2，10和11保留	00
17：16	控制P3.24脚的使用：00使能P3.24，01使能CS3，10和11保留	00
19：18	保留	—
20	如果位[5：4]不为10，由该位控制P2.29～P2.28的使用：为0时，使能P2.29～P2.28，为1时，保留	0
21	如果位[5：4]不为10，由该位控制P2.30的使用：为0时，使能P2.30，为1时使能AIN4	1
22	如果位[5：4]不为10，由该位控制P2.31的使用：为0时，使能P2.31，为1时使能AIN5	1
23	控制P3.0/A0用作端口引脚(0)或地址线(1)	如果$\overline{\text{RESET}}$＝0时BOOT[1：0]＝00，该位的复位值为1，反之为0
24	控制P3.1/A1用作端口引脚(0)或地址线(1)	如果复位时BOOT1＝0，该位复位值为1，反之为0
27：25	控制P3.23/A23/XCLK和P3.22～P3.2/A2.2～A2.2中地址线的数目： 000＝无地址线　100＝A[11：2]为地址线 001＝A[3：2]为地址线　101＝A[15：2]为地址线 010＝A[5：2]为地址线　110＝A[19：2]为地址线 011＝A[7：2]为地址线　111＝A[23：2]为地址线	如果复位时BOOT[1：0]＝11，该域的复位值为000，反之为111
31：28	保留	—

4. 实验步骤

(1)启动ADS1.2，使用ARMExecutable Image for MagicARM2200工程模板建立一个工程。

(2)在user组中的main.c中编写主程序代码。

(3)选用DebugInExram生成目标，然后编译连接工程。

(4)将MagicARM2200教学实验开发平台上跳线器JP22短接，JP20断开。

(5)选择Projet－>Debug，启动AXD进行JTAG仿真器调试。

(6)全速运行程序，检查蜂鸣器的效果。

(7)单击Context Variable图标打开变量观察窗口，通过此窗口可以观察局部变量和全局变量。

(8)选择System Views－>Debugger Internals即可打开LPC2000系列ARM7微控制器的片