



凌阳科技大学计划
Sunplus University Program

实验指导书

配合 SPCE061A 实验仪使用

凌阳科技大学计划编著

凌阳科技大学计划技术资料

<http://www.unsp.com.cn>

前言

为了使老师和同学更好地学习 SPCE061A 单片机，凌阳科技公司设计了 SPCE061A 实验仪，并设计了基于这个实验仪的 42 个实验，这些实验基本上涵盖了 SPCE061A 的全部硬件功能模块，每个实验都以《凌阳 16 位单片机应用基础》教材书为理论基础。

本书主要有四大部分：

第一章基础实验。本章针对 SPCE061A 的各硬件模块，设计了 21 个实验。设计这些实验的目的是使读者掌握 SPCE061A 的硬件结构、硬件模块、工作原理和简单的编程方法。

第二章语音实验。本章是凌阳音频应用实验，每一个实验针对凌阳音频函数库的一种使用方法，具有代表性。

第三章分立模组实验。这部分是基于 SPCE061A 实验仪的一些分立模组而设计的实验，从液晶显示到 UART/USB 通讯到外部存储器的扩展，选用这些实验是为了使读者了解在 SPCE061A 基础上扩展一些常用功能器件的方法。

第四章综合实验。本章共设计了 8 个实验，从几个模块的简单综合到实现一个完整功能，目的是使读者从整体上掌握利用 SPCE061A 开发设计的过程，实现比较复杂功能的编程方法。

全书的实验由浅入深，从部分到整体，从基本结构到模块再到应用的全部实验。读者按照实验顺序从基本的编程到功能实现的编程再到复杂的编程，一方面可以很好的学习和掌握单片机的基本知识，另一方面可以很容易的应用单片机进行设计，为以后的课程设计、毕业设计或者工作中的开发设计奠定了良好的基础。

本书涉及的全部范例代码在实验室赠送光盘的“实验仪实验指导书”文件夹都可以找到。读者在实验过程中可以参考。

在编写过程中难免会有错误纰漏，请广大读者朋友批评指正。

凌阳科技大学计划

2005.11.16

目录

前言	2
目录	3
第 3 章 分立模组实验	1
实验一 LCD 字符显示	1
实验二 LCD 汉字显示	7
实验三 LCD 动态图片显示	17
实验四 LCD 几何图形显示	25
实验五 UART/USB	30
实验六 SPR4096A FLASH 的擦除及其读写	42
第 4 章 综合实验	51
实验一 4 位 7 段 LED 数码管显示	51
实验二 1×8 键盘输入在 LED 数码管上的显示	58
实验三 1×8 键盘播放语音	63
实验四 语音识别&LCD 显示	67
实验五 带有背景音乐的动态图片	76
实验六 UART 控制液晶显示	80
实验七 0~3V 电压测量表	85
实验八 录音笔	89

第3章 分立模組实验

实验一 LCD 字符显示

【实验目的】

1. 了解 SPLC501 液晶显示器的工作原理。
2. 熟悉 SPCE061A 与 LCD 显示器的接口原理、控制方法。
3. 掌握利用液晶显示器显示字符的方法。

【实验设备】

1. 装有 Windows 系统和 $\mu'nSP^TM$ IDE 集成开发环境的 PC 机一台，SPCE061A 实验仪一套。
2. 本实验用到的实验仪硬件模块为：CPU 区电路模块，供电电路模块，下载模式选择电路模块，LCD 显示电路模块。

【实验要求】

1. 编程要求：主程序利用 C 语言编写，调用驱动程序（驱动程序已提供在实验仪指导书范例代码\example\model_Exa\driver\SPLC501driver）。
2. 实现功能：在 LCD 显示器上显示字符串“Sunplus”、“Technology for”和“Easy Living”，这三个字符串分三行显示。
3. 实验现象：清屏，延时，显示“Sunplus”，延时，显示“Technology for”和“Easy Living”，延时；再次清屏，延时，显示“Sunplus”，延时，显示“Technology for”和“Easy Living”，延时；如此循环。

【实验原理】

1. SPLC501 液晶显示模组工作原理

SPLC501 液晶显示模组采用的驱动控制芯片为凌阳科技的 SPLC501A 芯片，SPLC501A 为液晶显示控制驱动器，SPLC501A 单芯片液晶驱动，可以直接与其他微控制器接口总线相连。微控制器可以将显示数据通过 8 位数据总线或者串行接口写到 SPLC501A 的显存中。

SPLC501A 内置 8580 位显示 RAM。RAM 中的一位数据控制液晶屏上的一个象素点的亮、暗状态：“1”亮“0”暗。具有 65 行驱动输出和 132 列驱动输出（注：模组中的液晶显示面板仅为 64 行、128 列）。SPLC501 液晶显示模组的显示器上的显示点与驱动控制芯片中的显示缓存 RAM 是一一对应的；SPLC501A 芯片中共有 $65(8 \text{ Page} \times 8 \text{ bit} + 1) \times 132$ 位的显示 RAM 区。而显示器的显示点阵大小为 64×128 点，所以实际上在 SPLC501 液晶显示模组中有用的显示 RAM 区为 64×128 位；按 byte 为单位划分，共分为 8 个 Page，每个 Page 为 8 行，而每一行有 128 位（即 128 列）。

如要点亮 LCD 屏上的某一个点时，实际上就是对该点所对应的显示 RAM 区中的某一位进行置 1 操作；所以就要确定该点所处的行地址、列地址。SPLC501 液晶显示模组的行地址实际上就是 Page

的信息，每一个 Page 应有 8 行；而列地址则表示该点的横坐标，在屏上为从左到右排列，Page 中的一个 Byte 对应的是一列（8 行，即 8 个点），共 128 列。可以根据这样的关系在程序中控制 LCD 显示屏的显示。

2. SPLC501 液晶显示模组驱动程序及英文字符显示 API 函数介绍

凌阳大学计划为 SPLC501 液晶显示模组提供了显示驱动程序，由五个文件组成，底层驱动程序文件：SPLC501Driver_IO.inc、SPLC501Driver_IO.asm；用户 API 功能接口函数文件：SPLC501User.h、SPLC501User.c、DataOSforLCD.asm。

其中在 SPLC501User.c 定义了丰富易用的 API 函数，在 SPLC501User.h 文件里对这些函数进行了申明；SPLC501 液晶显示模组英文字符显示相关的 API 函数如下：

```
void LCD501_Init(unsigned int InitialData)           //初始化液晶显示
void LCD501_ClrScreen(unsigned int Mode)            //点亮屏幕或清屏
void LCD501_SetPaintMode(unsigned ModeCode)         //设置图形显示模式
unsigned int LCD501_GetPaintMode(void)               //获取图形显示模式
void LCD501_FontSet(unsigned int Font)              //选择显示字符的字体大小
unsigned int LCD501_FontGet(void)                   //获得显示字符的字体大小
void LCD501_Char(int x,int y,unsigned int a)        //显示字符
void LCD501_PutString(int x,int y,unsigned int a)   //显示字符串
```

这些函数的详细说明如下：

1) 【API 格式】C：void LCD501_Init(unsigned int InitialData)

【功能说明】液晶显示初始化

【参数】InitialData 初始化显示的字，低八位有效。

InitialData=0x00 表示全屏不显示

InitialData=0xFF 表示全屏显示

【返回值】无

【备注】在使用 LCD 前，首先应执行该函数，使 LCD 处于可以正常显示的状态。

2) 【API 格式】C：void LCD501_ClrScreen(unsigned int Mode)

【功能说明】液晶整屏清屏函数

【参数】清屏类型，Mode=1 表示全屏点亮，Mode=0 表示全屏清零。

【返回值】无

3) 【API 格式】C：void LCD501_SetPaintMode(unsigned int ModeCode)

【功能说明】设置图形显示模式

【参数】ModeCode 选择图形显示模式，可以是下列数值之一：

- 0 PAINT_COVER: 覆盖显示。图形所在位置的原有显示内容将被覆盖掉
- 1 PAINT_SET: 叠加显示。图形所在位置中原有的黑色点将被保留
- 2 PAINT_CLR: 擦除显示。图形反色显示，但图形所在位置中原有的白色点将被保留
- 3 PAINT_REV: 异或显示。图形中每一个点与原有的点做异或，即如果图形中某点的颜色与其所在位置原有的颜色相同，则该点显示白色，如果新颜色与原来颜色不同，则该点显示黑色
- 4 PAINT_RECO 取反并覆盖显示。原始图像数据取反，然后图形所在位置的原有

显示内容将被覆盖掉

【返回值】无

4) 【API 格式】C : unsigned int LCD501_GetPaintMode(void)

【功能说明】获得图形显示模式

【参数】: 无

【返回值】ModeCode - 图形显示模式，可以是下列数值之一：

- 0 PAINT_COVER: 覆盖显示。图形所在位置的原有显示内容将被覆盖掉
- 1 PAINT_SET: 叠加显示。图形所在位置中原有的黑色点将被保留
- 2 PAINT_CLR: 擦除显示。图形反色显示，但图形所在位置中原有的白色点将被保留
- 3 PAINT_REV: 异或显示。图形中每一个点与原有的点做异或，即如果图形中某点的颜色与其所在位置原有的颜色相同，则该点显示白色，如果新颜色与原来颜色不同，则该点显示黑色
- 4 PAINT_RECO 取反并覆盖显示。原始图像数据取反，然后图形所在位置的原有显示内容将被覆盖掉

5) 【API 格式】C : void LCD501_FontSet(unsigned int Font)

【功能说明】选择显示 ASCII 字符的大小类型

【参数】Font 选择 ASCII 字符的大小

Font=1，表示字体大小为 8×16

Font=0，表示字体大小为 6×8

【返回值】无

6) 【API 格式】C : unsigned int LCD501_FontGet(void)

【功能说明】获取当前显示 ASCII 字符的大小类型

【参数】无

【返回值】选择 ASCII 字符的大小

1，表示字体大小为 8×16

0，表示字体大小为 6×8

7) 【API 格式】C : void LCD501_Char(int x,int y,unsigned int a)

【功能说明】显示单个 ASCII 字符

【参数】x 表示显示字符的起始列，x 的范围为：0~127

y 表示显示字符的起始行，y 的范围为：0~63

a 表示字符的 ASCII 码

【返回值】无

8) 【API 格式】C : void LCD501_PutString(int x,int y,unsigned int a)

【功能说明】显示 ASCII 字符串

【参数】x 表示显示字符的起始列，x 的范围为：0~127

y 表示显示字符的起始行，y 的范围为：0~63

a 表示字符串的首地址送到指针变量 a。

【返回值】无

3. SPLC501 液晶显示模组显示英文字符

利用 SPLC501 液晶显示模组显示英文 (ASCII) 字符时, 需要对 LCD 进行初始化操作, 以初始化 LCD 内部的供电方式、驱动设置等; 在凌阳大学计划提供的 SPLC501 液晶显示模组的驱动程序中, 提供了对 SPLC501 液晶显示模组的初始化程序, 除了完成前面所述的操作外, 该函数还可以初始化液晶的显示。初始化 SPLC501 液晶显示模组后, 驱动程序默认设置图形显示模式为覆盖模式, ASCII 字符的字型默认为 8×16 的大小, 如果需要修改这些参数可以调用对应的函数进行设置。

SPLC501 液晶显示模组在完成初始化后, 就可以调用单个 ASCII 字符显示或者 ASCII 字符串显示函数进行字符显示了; 另外, 驱动程序中还提供了多种图形显示模式, 可以选择不同的图形显示模式来达到不同的显示效果。如图 3-1。

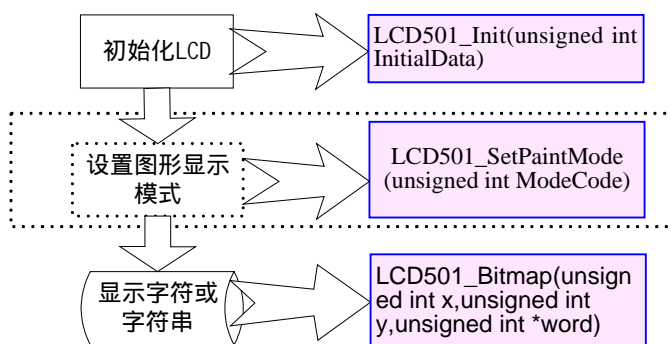


图 3-1 SPLC501 液晶显示模组显示英文字符的过程

注: 虚线框内表示这项可有可无, 如果不调用这个函数设置图形显示模式, 则默认设置图形显示模式为覆盖模式。

【硬件连接】

硬件连接图如图 3-2, 分别用跳线连接 IOB3 与 AO, IOB4 与 RWP, IOB5 与 EP, IOB2 与 /CS, IOA0~IOA7 与 DB0~DB7。即用跳线把实验仪 LCD (*) 的所有引针全部短接。

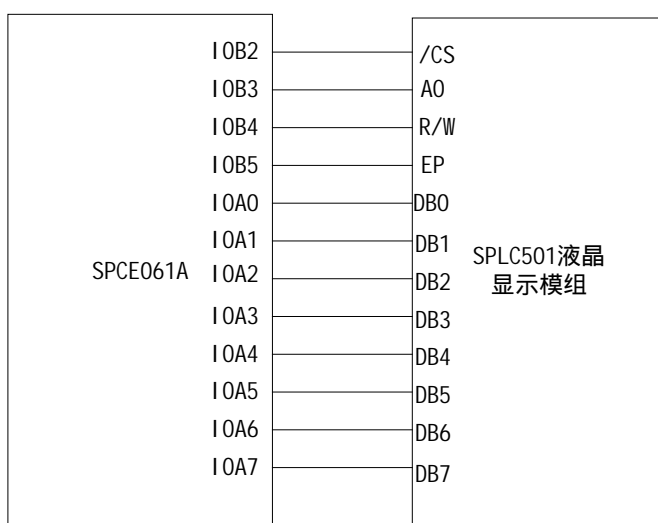


图 3-2 硬件连接框图

*注: 和上册相同, 本书中以 v2.0.2 版本实验仪的标号为准, 如果读者使用其他版本的实验仪, 以硬件连接图为准。

【程序流程图】

主程序流程图如图 3-3，按照原理中讲的利用 SPLC501 液晶显示模組显示英文字符过程：初始化 LCD，全屏点亮；进入显示字符循环，清屏后显示“Sunplus”，显示字号为 8×16 大小，显示在 (30, 3) 坐标开始的位置；延时；分别显示“Technology for”和“Easy Living”，显示字号为 6×8 大小，“Technology for”显示在 (0, 40) 坐标开始的位置，“Easy Living”显示在 (50, 50) 坐标开始的位置；延时。这里两次延时只是为了能看清实验现象，没有其他特别的作用。

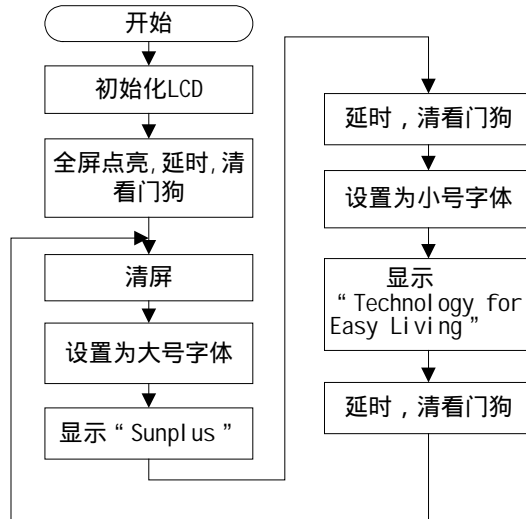


图 3-3 主程序流程图

【实验步骤】

1. 新建一个工程 ex1_LCD501_Character，在工程中新建一个 C 语言文件 main.c。
2. 按照实验原理，凌阳大学计划为 SPLC501 液晶显示模組提供了显示驱动程序，驱动程序里定义了常用的一些显示函数，所以这里先拷贝驱动程序文件到 ex1_LCD501_Character 工程：

第一步：实验仪指导书范例代码\example\model_Exa\Driver\SPLC501driver 下找到驱动文件 DataOSforLCD.asm、SPLC501Driver_IO.asm、SPLC501Driver_IO.inc、SPLC501User.c 和 SPLC501User.h。

第二步：拷贝这些文件到新建的工程文件夹 ex1_LCD501_Character 以方便新建工程加载使用。

3. 添加 DataOSforLCD.asm、SPLC501Driver_IO.asm 和 SPLC501User.c 文件到“Source Files”，添加 SPLC501Driver_IO.inc 和 SPLC501User.h 文件到“Head Files”。添加方法和语音实验 1 中添加“hardware.asm”文件方法一样，添加之后就编写程序时就可以直接使用其中的 API 函数了。

4. 按照实验原理，常用的 API 函数是在 SPLC501User.h 头文件中声明的，所以这里需要在 main.c 文件开始包含头文件 SPLC501User.h。

5. 按照程序流程图在 main.c 文件里编写 C 语言程序，程序里调用 SPLC501 驱动程序中的子函数，控制 LCD 显示。

6. Rebuild All。

7. 按照硬件连接图连接硬件。

8. 下载程序到实验仪，运行程序。



9. 观察实验仪 SPLC501 液晶显示模组的显示情况，分析是否和实验要求统一。

【范例路径】

在以下路径有可供大家参考的程序。

- ◆ 实验仪指导书范例代码*\example\model_Exa\ex1_LCD501_Character。
- ◆ 凌阳科技大学计划网站（www.unsp.com.cn）下载专区可以下载到相关资料。

*注：

和上册一样，路径中的“实验仪指导书范例代码”事实上指的是：光盘提供的实验仪指导书范例代码解压缩后的路径。光盘提供的实验仪指导书的范例代码在凌阳大学计划光盘\14. 书籍资料\实验仪指导书\example 路径可以找到。例如当在光盘上找到实验仪指导书的范例代码后，把 Example.rar 文件夹解压缩到 E 盘，则“实验仪指导书范例代码”指的就是“E:\Example”。

不要在光盘里直接打开范例程序下载、运行，因为当工程当中有某些文件的属性为只读时，工程是无法正常进行编译的。

【练习题】

编写程序在 SPLC501 液晶显示器上显示一段英文文字（50 字符左右），要求利用 2 个键盘控制显示。两个键盘的功能为：

K1：下移，即如果按 K1 键，所有字符下移

K2：上移，即如果按 K2 键，所有字符上移

实验二 LCD 汉字显示

【实验目的】

1. 掌握 DM Tool 字模提取工具的使用方法。
2. 掌握利用 SPLC501 液晶显示模组显示汉字的方法。

【实验设备】

1. 装有 Windows 系统和 $\mu'nSP^TM$ IDE 集成开发环境的 PC 机一台, SPCE061A 实验仪一套。
2. 本实验用到的实验仪硬件模块为: CPU 区电路模块, 供电电路模块, 下载模式选择电路模块, LCD 显示电路模块。

【实验要求】

1. 编程要求: 主程序利用 C 语言编写, 调用驱动程序(驱动程序已提供在实验仪指导书范例代码\example\model_Exa\ driver\ SPLC501driver)。
2. 实现功能: 在 LCD 显示器上显示 8 个汉字“凌”、“阳”、“科”、“技”、“落”、“实”、“生”、“活”, 这 8 个字分四行显示: “凌”、“阳”两个字在同一行显示, “科”、“技”两个字在同一行显示, “落”、“实”两个字在同一行显示, “生”、“活”两个字在同一行显示。
3. 实验现象: 清屏, 持续一段时间, 在第一行显示“凌”和“阳”, 持续一段时间, 在第二行显示“科”和“技”, 持续一段时间, 在第三行显示“落”和“实”, 持续一段时间, 在第四行显示“生”和“活”, 持续一段时间; 循环显示。

【实验原理】

1. DM Tool 字模提取工具

与显示英文字符(ASCII 码)不同, 汉字不能直接显示在点阵的 LCD 显示器上, 而是要借助一个工具, 先把汉字转换成 LCD 显示器能显示的形式。凌阳科技教育推广中心提供了针对 unSP 系列 16 位单片机的字模数据提取工具——DM Tool; 可以利用此工具提取汉字字模、BMP 位图字模, 还可以很方便地提取 ASCII 码字模。DM Tool 在大学计划网站 SPLC501 液晶模组资料里提供。

下面介绍利用 DM Tool 字模提取工具提取汉字字模的步骤。

第一步: 安装好 DM Tool 后, 在[开始]->[程序]->[sunplus]->[DM Tool]打开 DM Tool 工具, 界面如图 3-4。

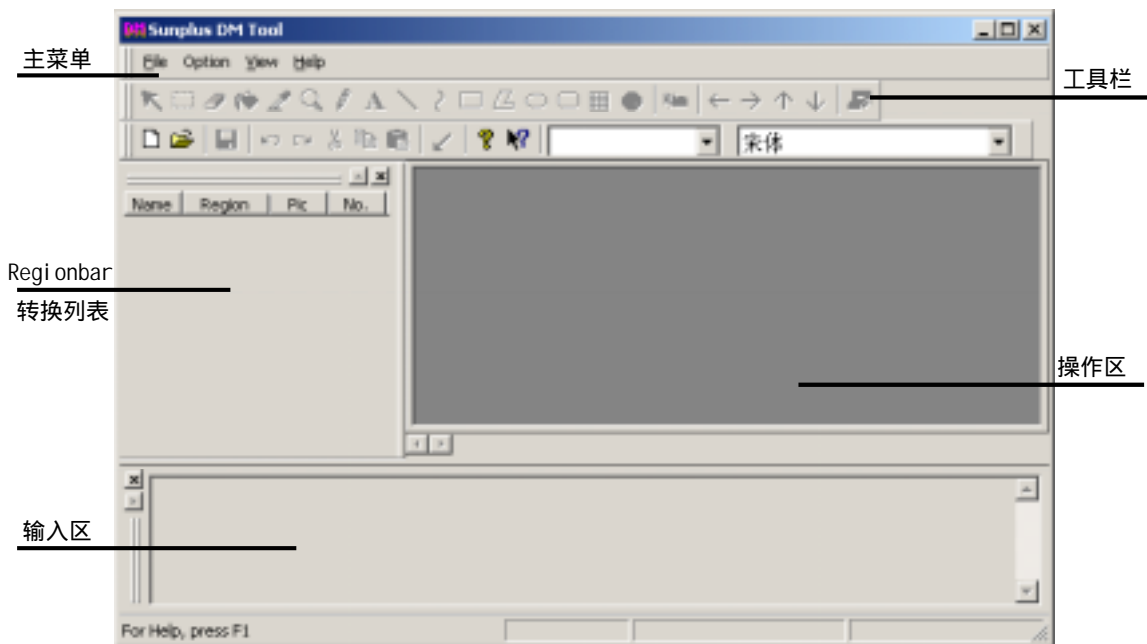


图 3-4 DM Tool 界面

第二步：选择[File]—[News]菜单项，新建工程，如图 3-5 所示：

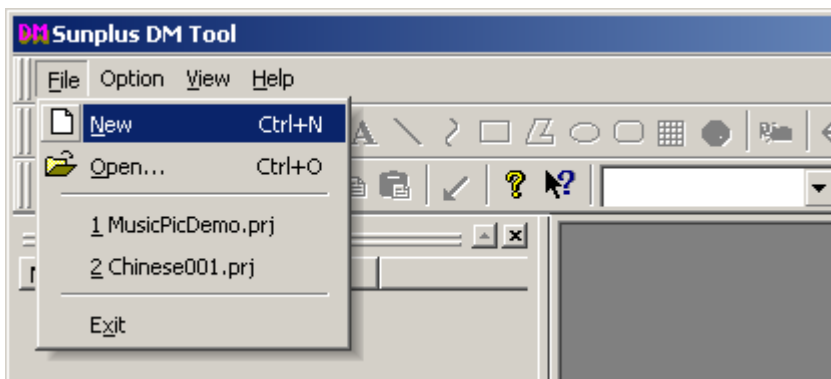


图 3-5 新建工程

选择后，即新建了一个工程；如图 3-6 所示。用户可以选择[File]菜单中的[Save]菜单项来保存工程，也可以直接对工程进行操作，进行字模数据提取的操作。

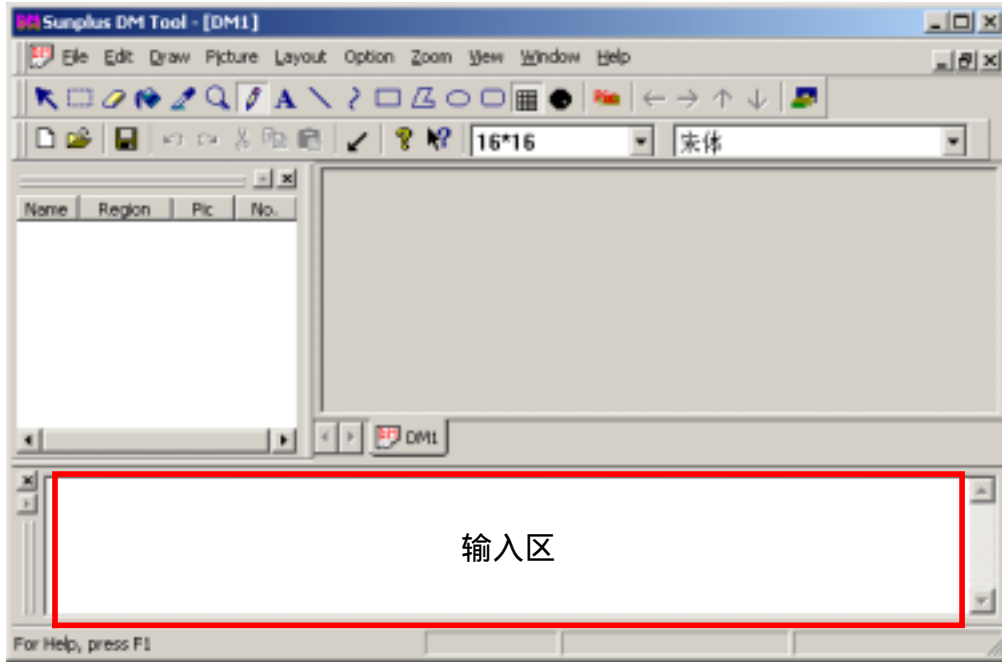


图 3-6 新建工程后在界面

第三步：在输入区中输入汉字：凌阳科技落实生活。如图 3-7。

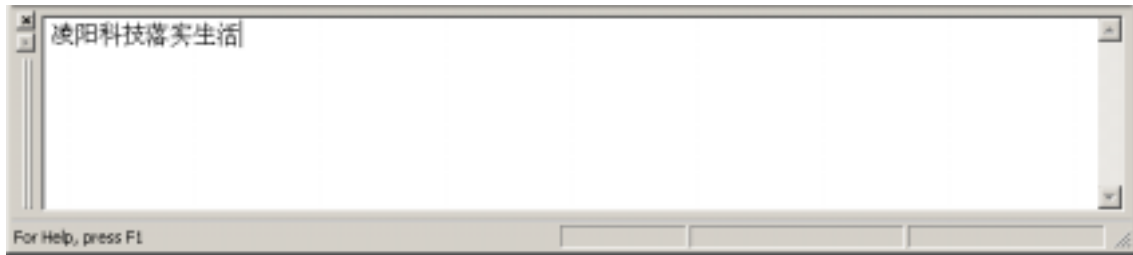


图 3-7 在输入区中输入“凌阳科技落实生活”

这时利用快捷键组合：“Ctrl+Enter”，完成汉字的输入操作，可以看到转换列表中出现“凌阳科技落实生活”四个汉字的列表，而在操作区中也会出现汉字的点阵图，如图 3-8：

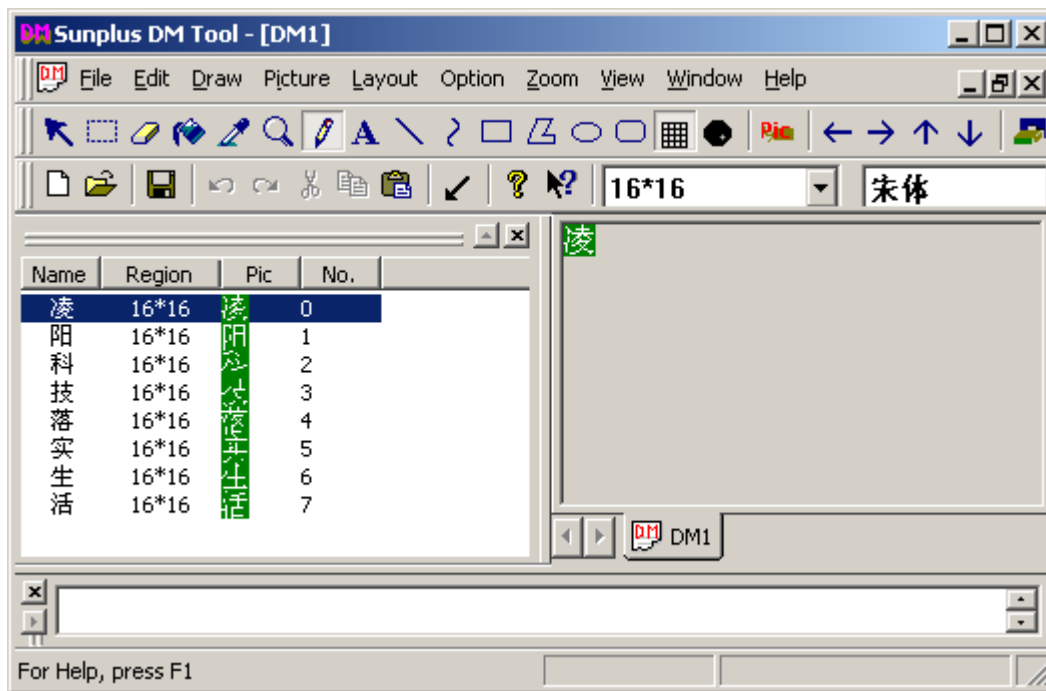


图 3-8 完成汉字输入操作

第四步：进行字模数据的提取。选择菜单[File]中的[Export CChar file...]菜单项，或者直接用工具栏最右边的“Export”工具，如图 3-9 所示；打开字模数据生成窗口，如图 3-10 所示：

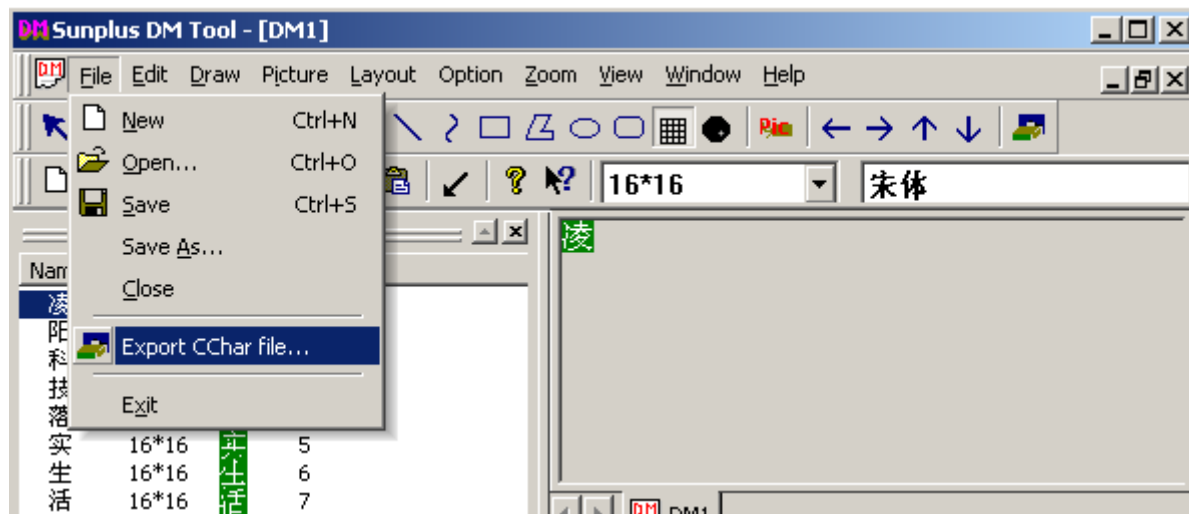


图 3-9 选择“Export...”选项切换窗口

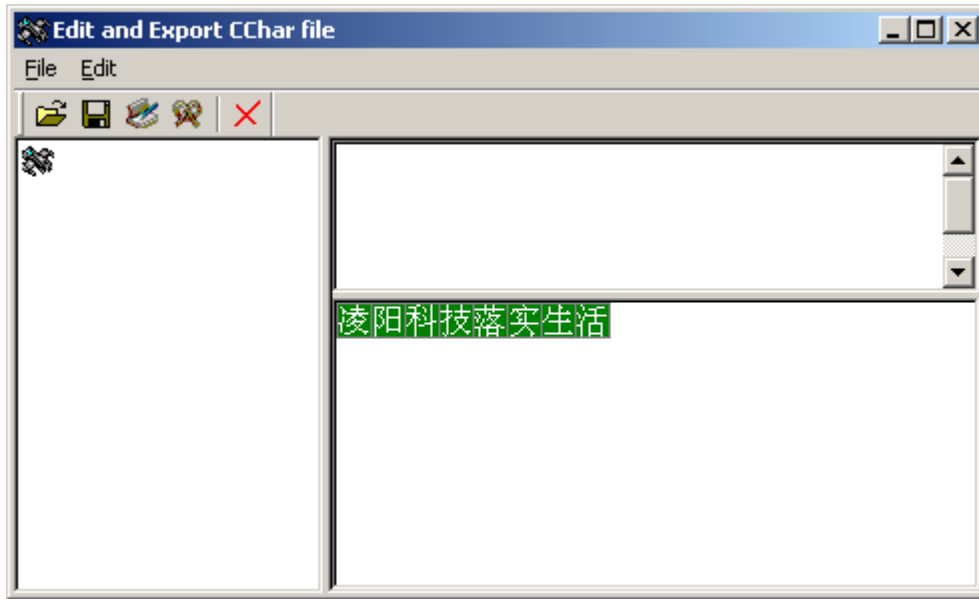


图 3-10 字模数据生成窗口 (Edit and Export CChar file)

第五步：在字模数据生成窗口中，选择[File]菜单中的[Export CChar file]菜单项，或工具栏中的“Export CChar file”工具，开始生成数据，即打开了“Export CChar file”对话框，如图 3-11 和图 3-12 所示：



图 3-11 选择“Export CChar file”选项打开对话框

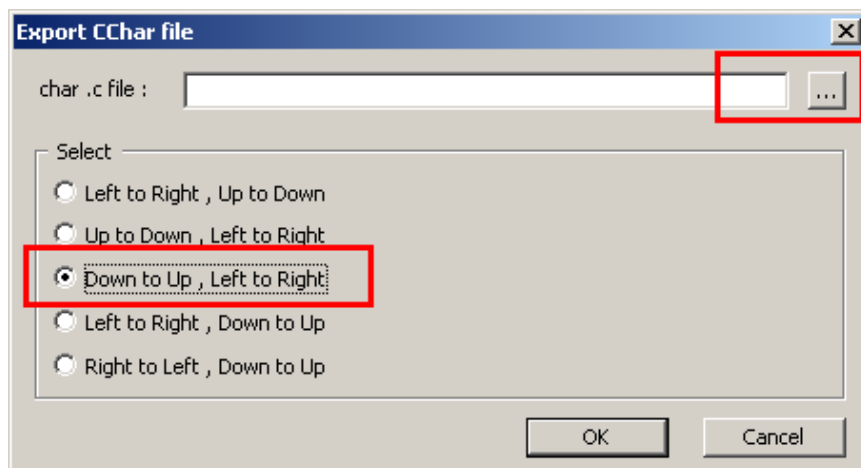


图 3-12 “Export CChar file”对话框

第六步：在对话框的“Select”选项中，选择“Down to Up,Left to Right”选项；在“char .c file:”输入框的右边的工具，可以选择导出文件的路径和文件名，这里保存在“我的文档”中，文件名为“Mychar”，如图 3-13 所示；然后点击“保存”返回“Export CChar file”对话框；再在“Export CChar file”中点击“OK”，即可生成保存有字模数据的两个文件：“Mychar.c”和“Mychar.h”；如图 3-14

所示：

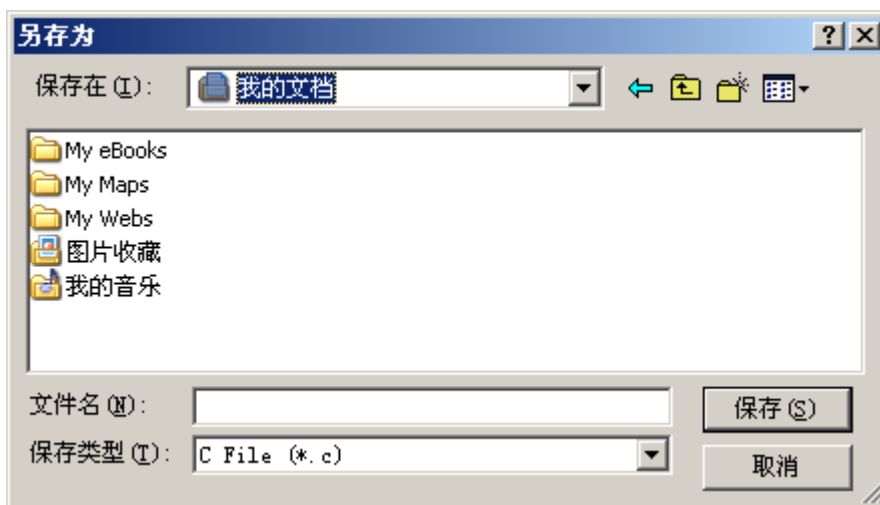


图 3-13 选择路径和保存文件名

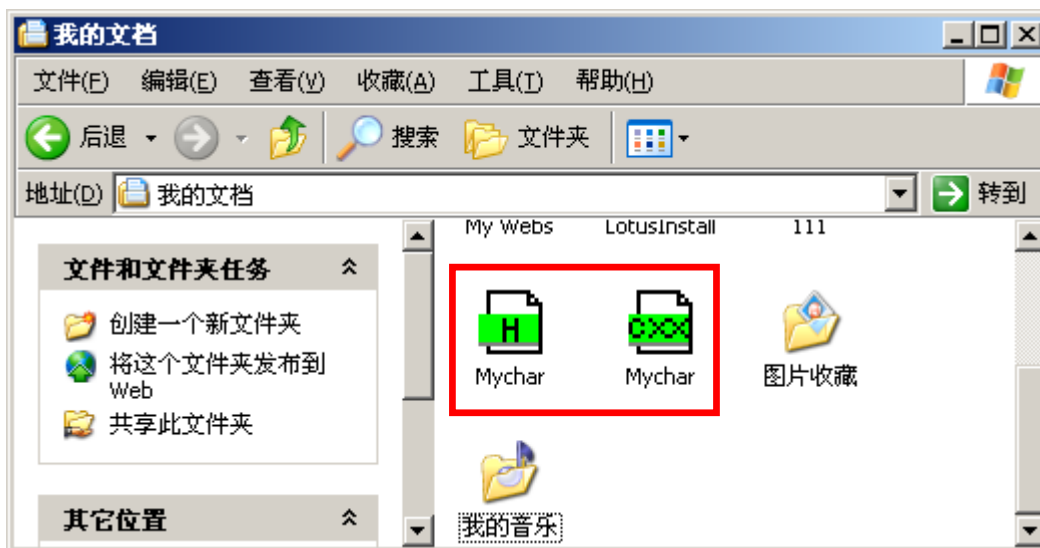


图 3-14 导出的字模数据文件

这时，汉字字模数据的生成操作已完成，已经生成了所需的字模数据文件：“Mychar.c”和“Mychar.h”。

导出文件后，用户可以关闭字模数据生成窗口，返回到 DM Tool 软件的主窗口，以便进行其它的操作，DM Tool 的使用方法可以参考 DM Tool 的用户手册。

生成的“Mychar.c”和“Mychar.h”两个文件用 unSPIDE 打开后如图 3-15 和图 3-16 所示：


```

Mychar.c
//Mychar.c file define all chars information.

typedef unsigned int WORD;

const WORD encoding_00[] = //凌
{
    0x1010,
    0x0200, 0x3e02, 0x01dc, 0x4008, 0x4510, 0x4490, 0x2254, 0x2374,
    0x15b4, 0x091f, 0x1534, 0x1334, 0x2154, 0x6090, 0x2010, 0x0000
};

const WORD encoding_01[] = //阳

```

图 3-15 生成的.C 文件

```

Mychar.h
//Mychar.h file define all chars index.

#ifndef _MYCHAR_H
#define _MYCHAR_H

typedef unsigned int WORD;

extern const WORD encoding_00[]; //凌
extern const WORD encoding_01[]; //阳
extern const WORD encoding_02[]; //科
extern const WORD encoding_03[]; //技

```

图 3-16 生成的.h 头文件

从图中可见，导出的字模数据以数组的形式保存，可以直接包含到工程中进行引用。

2. SPLC501 液晶显示模组汉字显示 API 函数介绍

驱动程序文件 SPLC501User.c 同样定义了显示汉字的 API 函数，在 SPLC501User.h 文件里也对这些函数进行了申明。SPLC501 液晶显示模组英文字符显示相关的 API 函数如下：

```

void LCD501_Init(unsigned int InitialData) //初始化液晶显示
void LCD501_ClrScreen(unsigned int Mode) //点亮屏幕或清屏
void LCD501_SetPaintMode(unsigned int ModeCode) //设置图形显示模式
unsigned int LCD501_GetPaintMode(void) //获取图形显示模式
void LCD501_Bitmap(unsigned int x,unsigned int y,unsigned int *word)//显示汉字或者位图的字模数据

```

前面四个函数在上一实验中已经有详细说明，这里只介绍最后一个函数：

1) 【API 格式】C：void LCD501_Bitmap(unsigned int x,unsigned int y,unsigned int *word)

【功能说明】显示汉字或者位图的字模数据

【参数】x 表示显示字符的起始列（0~128）

y 表示要显示的字模的起始行（0~64）

*word 表示要显示的汉字字模的数据首地址

【返回值】无

【备注】只显示用 DM Tool 取字模软件所取的汉字字模数据；该函数可以在指定位置显示位图。

参数 x 和 y 是位图的左上角坐标。word 是“位图数据”的起始地址。“位图数据”的第一个 Word 声明了位图的大小，低 8 位代表位图宽度，高 8 位代表位图高度。如 0x172B 表示位图的宽为 43 像素，高为 23 像素。从第二个 Word 开始描述位图每个点的颜色（1-黑色，0-白色）。点对应每个数据位的情况为：从下到上，自左到右，以 Word（16bit）为存储单位，不足 16 位的补 0。可以利用凌阳大学计划提供的 DM Tool 字模提取工具提取出汉字或 BMP 图像的字模数据，然后直接引用由其生成的字模数组。

如果在 SPLC501 液晶显示模块上显示前面利用 DM Tool 生成的“凌”的字模数据，可以调用 LCD501_Bitmap 函数显示；可以看到，生成的 Mychar.c 文件中，“凌”字的字模数组为：encoding_00；假设“凌”字显示在（0，0）位置，则显示“凌”字的代码如下：

```
LCD501_Bitmap(0,0,(unsigned int *)encoding_00);
```

3. SPLC501 液晶显示模组显示汉字

利用 SPLC501 液晶显示模组显示汉字比显示英文字符稍微复杂一点，因为英文字符在 ASCII 码中都有对应字符，而 SPLC501 液晶显示模组显示的都是 ASCII 码，汉字却不能与 ASCII 码对应，所以要先用字模提取工具把汉字提取成字模数据，才能显示。

利用 SPLC501 液晶显示模组显示汉字需要图 3-17 的操作过程。新建一个工程；利用字模提取工具提取字模数据；把字模数据文件保存在工程文件夹，以方便工程调用；添加字模数据文件到工程，程序可以直接使用这些字模数据；拷贝并添加驱动程序文件；编写程序；运行显示。

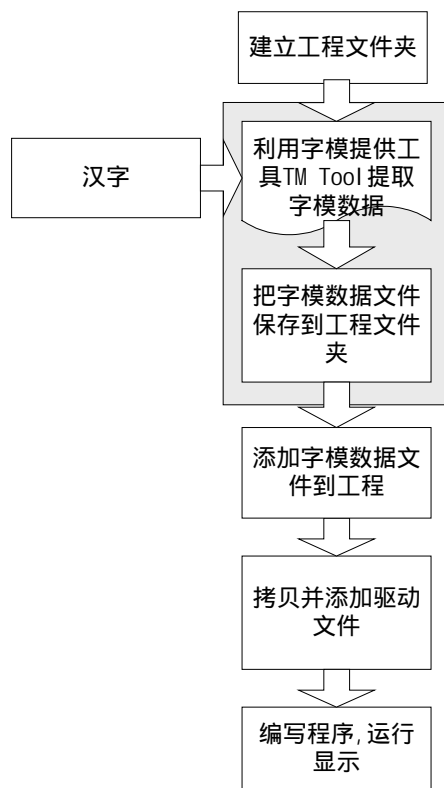


图 3-17 利用 SPLC501 液晶显示模组显示汉字需要的操作过程

利用 SPLC501 液晶显示模组显示汉字的过程见图 3-17，和显示英文字符的方法类似，所不同的是显示汉字的 API 函数为 LCD501_Bitmap(unsigned int x,unsigned int y,unsigned int *word)。

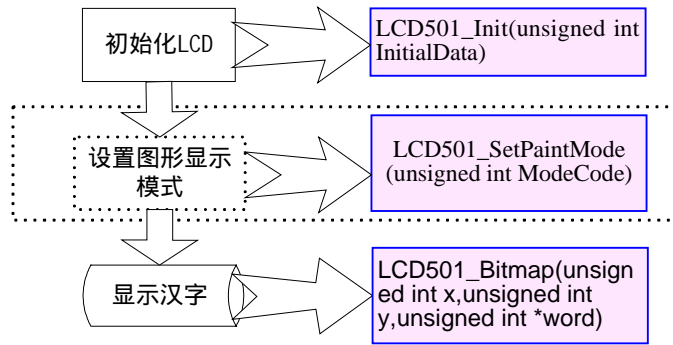


图 3-18 SPLC501 液晶显示模組显示汉字的过程

【硬件连接】

硬件连接图同上一实验图 3-2。分别用跳线连接 IOB3 与 AO ,IOB4 与 RWP ,IOB5 与 EP ,IOB2 与 /CS , IOA0~IOA7 与 DB0~DB7 ; 即用跳线把实验仪 LCD 的所有引针全部短接。

【程序流程图】

主程序流程图如图 3-19，和上面显示英文字符类似，初始化 LCD，进入显示汉字循环：清屏，延时；显示“凌”“阳”，“凌”显示在 (0,0) 位置，“阳”显示在 (16,0) 位置，延时；显示“科”“技”，“科”显示在 (32,16) 位置，“技”显示在 (48,16) 位置，延时；显示“落”“实”，“落”显示在 (64,32) 位置，“实”显示在 (80,32) 位置，延时；显示“生”“活”，“生”显示在 (96,48) 位置，“活”显示在 (112,48) 位置，延时。

实验中的几次延时同样是为了看清楚实验现象，同时符合实验要求，没有其他特别的功能。

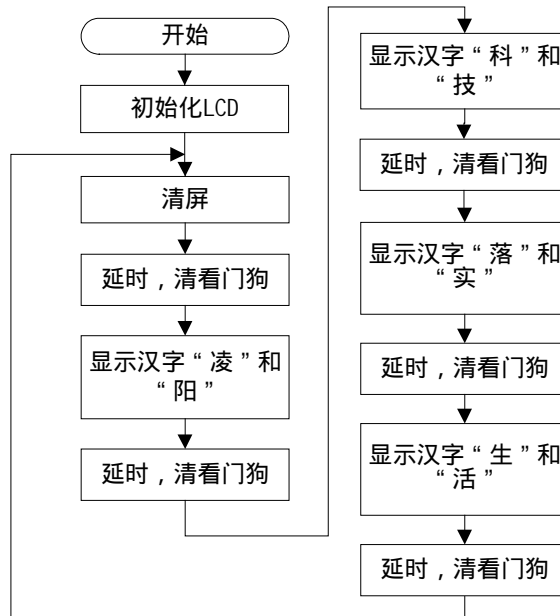


图 3-19 主程序流程图

【实验步骤】

1. 新建一个工程 ex2_LCD501_Chinese，在工程中新建一个 C 语言文件 main.c。
2. 利用字模提取工具提取“凌阳科技落实生活”8 个字的字模数据，把字模数据文件（命名为 Chinese_Table）保存在 ex2_LCD501_Chinese 工程文件夹。
3. 添加 Chinese_Table.c 文件到 ex2_LCD501_Chinese 工程的“Source Files”，添加 Chinese_Table.h 文件到工程的“Head Files”。添加方法和添加“hardware.asm”相同。
4. 拷贝 DataOSforLCD.asm、SPLC501Driver_IO.asm、SPLC501Driver_IO.inc、SPLC501User.c 和 SPLC501User.h 驱动程序文件到 ex1_LCD501_Character 工程，拷贝方法同上一实验。
5. 添加 DataOSforLCD.asm、SPLC501Driver_IO.asm 和 SPLC501User.c 文件到“Source Files”，添加 SPLC501Driver_IO.inc 和 SPLC501User.h 文件到“Head Files”。
6. 在 main.c 文件开始包含头文件 SPLC501User.h，以及字模数据文件的头文件 Chinese_Table.h。
7. 按照程序流程图在 main.c 文件里编写 C 语言程序。
8. Rebuild All。
9. 按照硬件连接图连接硬件。
10. 下载程序到实验仪，运行程序。
11. 观察实验仪 SPLC501 液晶显示模组的显示情况，分析是否和实验要求统一。

【范例路径】

在以下路径有可供大家参考的程序。

- ◆ 实验仪指导书范例代码\example\model_Exa\ex2_LCD501_Chinese。
- ◆ 凌阳科技大学计划网站（www.unsp.com.cn）下载专区可以下载到相关资料。

【练习题】

编写程序在 SPLC501 液晶显示器上显示一段汉字（20 个字左右，写什么字自行定义），要求利用 2 个键盘控制显示。两个键盘的功能为：

K1：下移，即如果按 K1 键，所有汉字下移

K2：上移，即如果按 K2 键，所有汉字上移

实验三 LCD 动态图片显示

【实验目的】

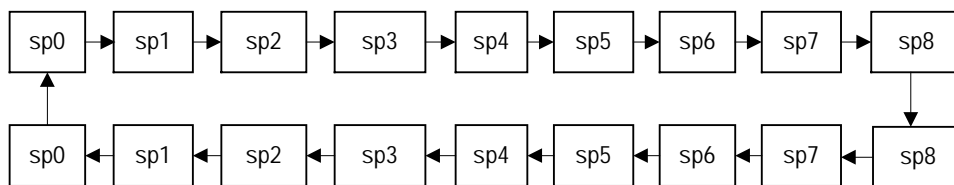
1. 熟悉利用 DM Tool 字模提取工具提取图片字模的方法。
2. 掌握利用液晶显示器显示动态图片的方法。

【实验设备】

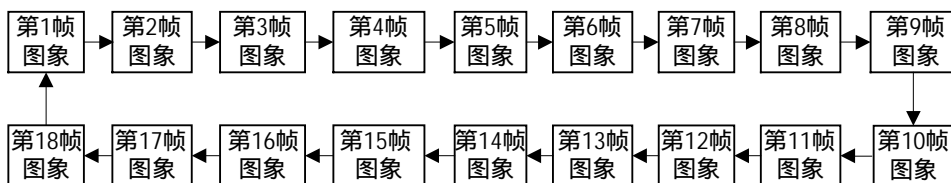
1. 装有 Windows 系统和 $\mu'nSP^TM$ IDE 集成开发环境的 PC 机一台，SPCE061A 实验仪一套。
2. 本实验用到的实验仪硬件模块为：CPU 区电路模块，供电电路模块，下载模式选择电路模块，LCD 显示电路模块。

【实验要求】

1. 编程要求：主程序和中断服务程序都利用 C 语言编写，调用驱动程序（驱动程序已提供在实验仪指导书范例代码\example\model_Exa\driver\SPLC501driver）。
2. 实现功能：在液晶显示器上动态显示已经提供的图片，这里显示凌阳科技的标识（sunplus 的所有 9 张图片）（提供路径为：实验仪指导书范例代码\example\model_Exa\ex3_LCD501_Graphic\picture\sunplus）。从第一张图象依次显示到第九张图象之后，再从第九张图象依次显示第一张图象，每张图象显示刷新时间为 0.2s，也就是说图象显示的顺序为：



0.2 的刷新时间要求利用 IRQ4_1KHz 中断实现。把每一个图象显示称为一帧图象显示，则根据实验要求，在显示一个完整的凌阳标识的旋转时需要显示 18 帧图象。则表示为图象帧显示的顺序为：



3. 实验现象：凌阳科技的标识在屏幕内旋转。

【实验原理】

1. DM Tool 字模提取工具

上一实验中介绍了利用 DM Tool 提取汉字的字模数据的方法，这里介绍利用此工具提取 BMP 位图字模数据的方法。和提取汉字字模数据类似，利用 DM Tool 字模提取工具提取 BMP 位图字模的

步骤如下：

第一步，导入图片：打开 DM Tool 工具，新建一个 DM Tool 的工程，在“Picture”的下拉菜单点击“Import Pictures...”，如图 3-20。

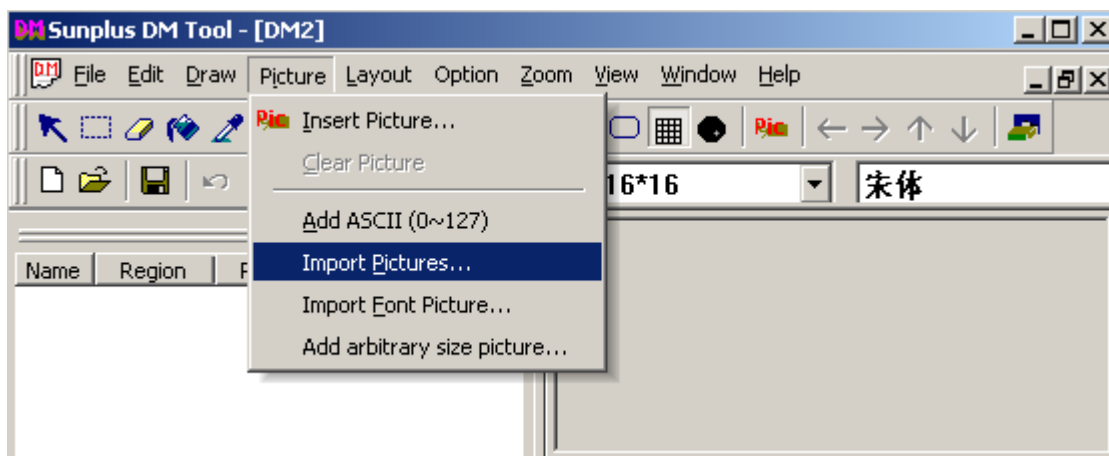


图 3-20 导入图片第一步

点击“Import Pictures...”后出现下图的对话框，在实验仪指导书范例代码\example\model_Example3_LCD501_Graphic\picture\sunplus 找到我们要提取的图片 sp1，如图 3-21。

注意：利用“Import Pictures...”菜单导入图象的选项只能导入 BMP 格式的位图；DM Tool 工具可以最终生成字模数据的尺寸为 240×240 点，所以用户在导图象时要注意图象的尺寸；但由于 SPLC501 液晶显示器的屏幕大小为 128×64 点，则在 SPLC501 液晶显示模块上显示的图片尺寸就不能超过 128×64 点，如超过则无法显示。用户可以利用图像处理软件对原始图像进行修改后再导入 DM Tool 中，提取合适的图像字模数据。

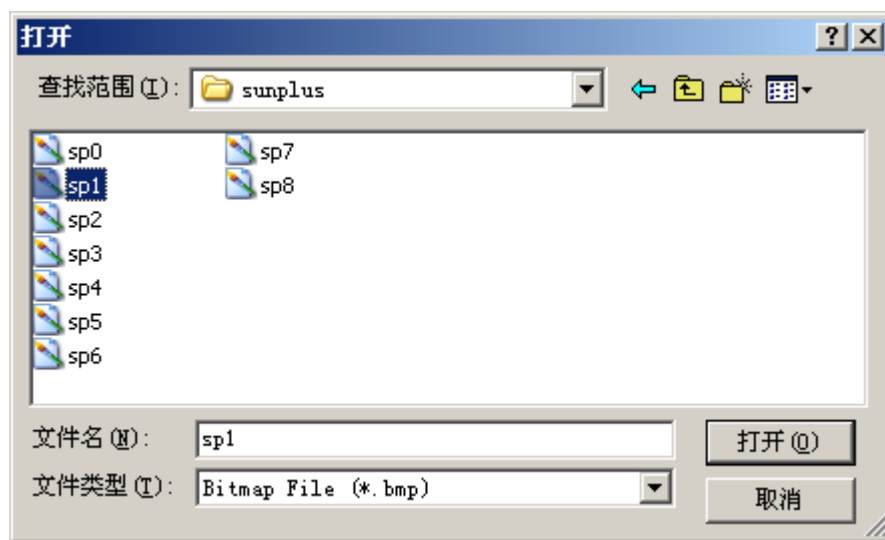


图 3-21 导入图片第二步

选择 sp1，点击“打开”，出现图 3-22 的对话框。按照图中选择“source image”和“Invariable”，点击“OK”就导入图片了。

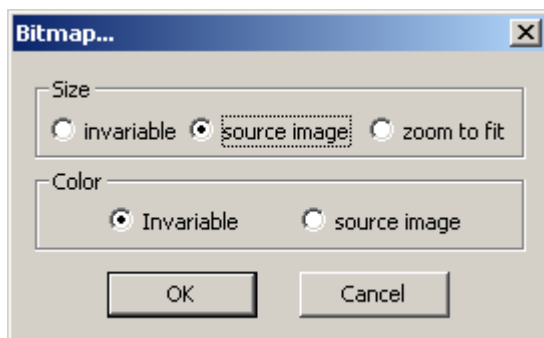


图 3-22 导入图片第三步

如图 3-23，导入的图片就出现在 Regionbar 转换列表和操作区中。

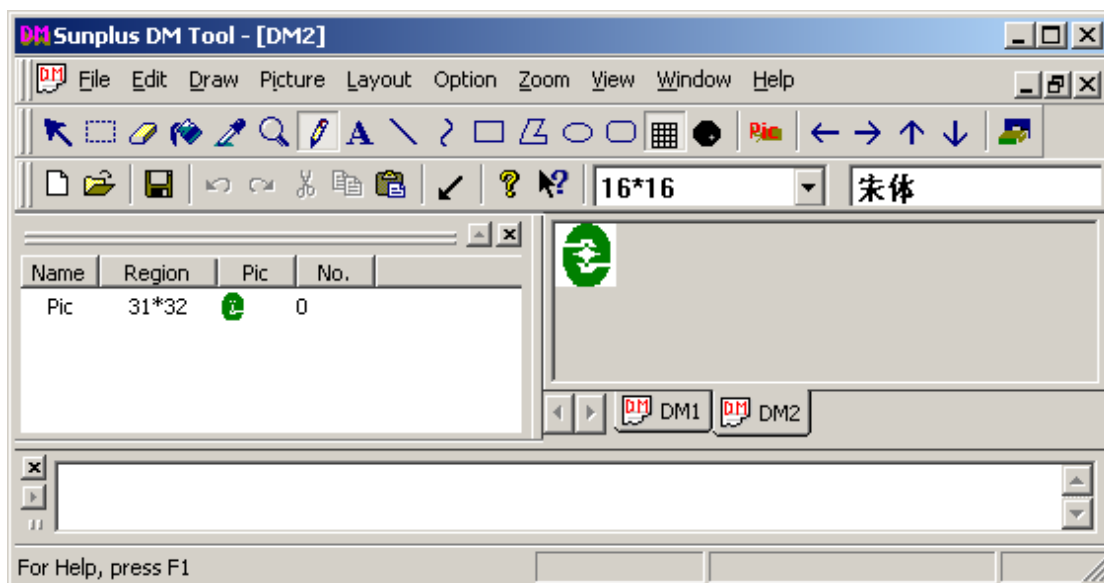


图 3-23 导入图片后 DM Tool 界面

第二步，提取字模并保存：点击  图标，打开图 3-24 的对话框。

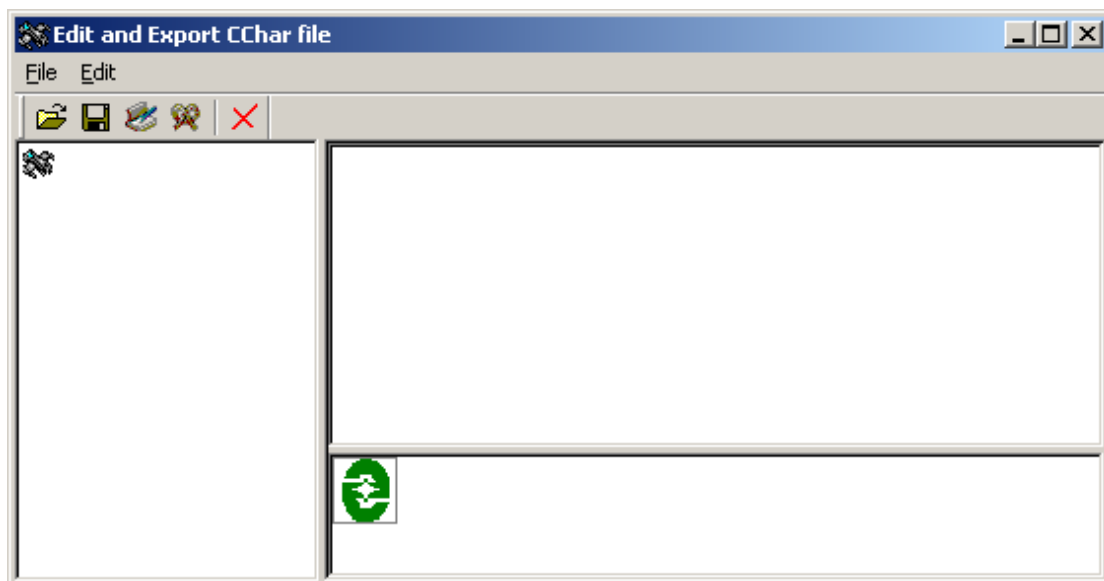



图 3-24 Edit and Export Cchar File 界面

点击  图标，打开图 3-25 的对话框。

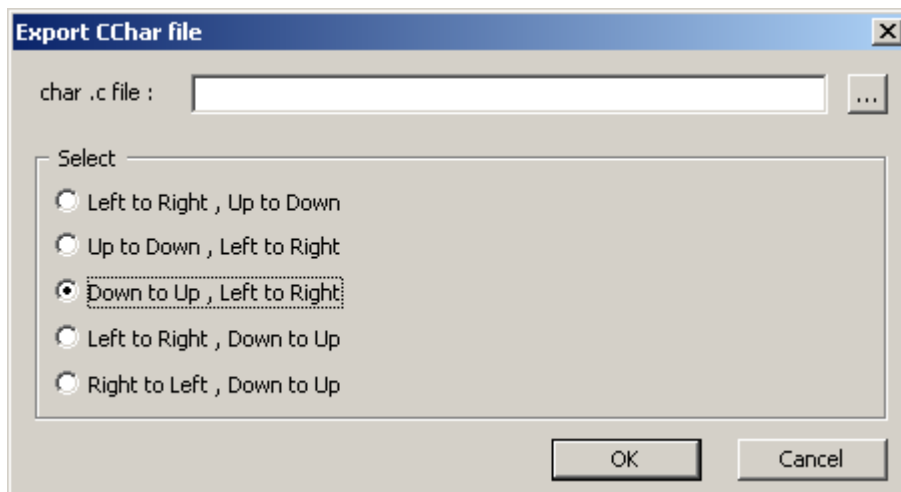


图 3-25 选择“Down to Up, Left to Right”

“Select”选择“Down to Up, Left to Right”，点击“char.c file:”右边的浏览图标，打开图 3-26 的对话框。

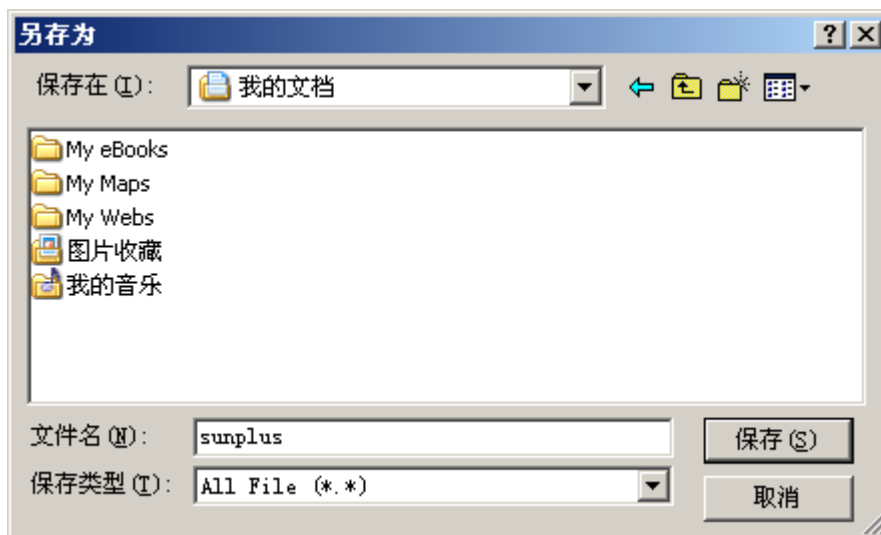


图 3-26 选择保存路径

这里选择保存到“我的文档”文件夹中，选择保存的文件名为 sunplus，点击保存，回到“Export Cchar File”窗口，点击“OK”。这时字模数据的生成操作已完成，生成了 sunplus.c 和 sunplus.h 两个字模数据文件，如图 3-27。

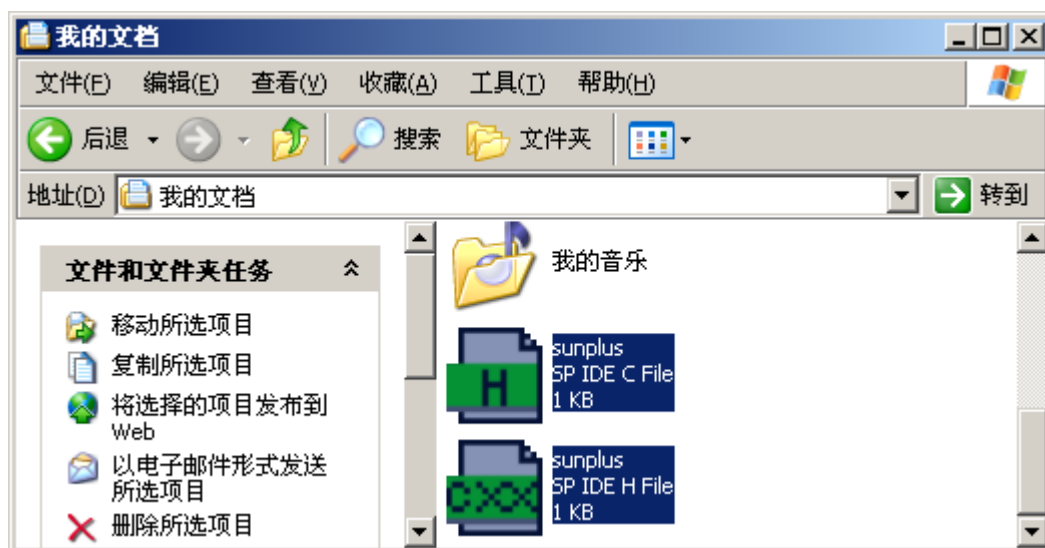


图 3-27 生成字模数据文件

导出文件后，用户可以关闭字模数据生成窗口，返回到 DM Tool 软件的主窗口，以便进行其它的操作，DM Tool 的使用方法可以参考 DM Tool 的用户手册。

和导出的汉字字模数据相同，BMP 位图字模数据也以数组的形式保存，可以直接包含到工程中进行引用。

2. SPLC501 液晶显示模组图片显示 API 函数介绍

SPLC501 液晶显示模组英文字符显示相关的 API 函数如下：

```
void LCD501_Init(unsigned int InitialData)           //初始化液晶显示
void LCD501_ClrScreen(unsigned int Mode)           //点亮屏幕或清屏
void LCD501_SetPaintMode(unsigned int ModeCode)    //设置图形显示模式
unsigned int LCD501_GetPaintMode(void)              //获取图形显示模式
void LCD501_Bitmap(unsigned int x,unsigned int y,unsigned int *word)//显示汉字或者位图的字模数据
```

这些函数在本章实验二里已经详细介绍，不再赘述！

3. SPLC501 液晶显示模组显示图片

和显示汉字类似，利用 SPLC501 液晶显示模组显示 BMP 位图的过程为：新建一个工程，利用字模提取工具提取字模，把字模数据文件保存在工程文件夹，添加字模数据文件到工程，拷贝并添加驱动程序文件，编写程序，运行显示。如图 3-28。

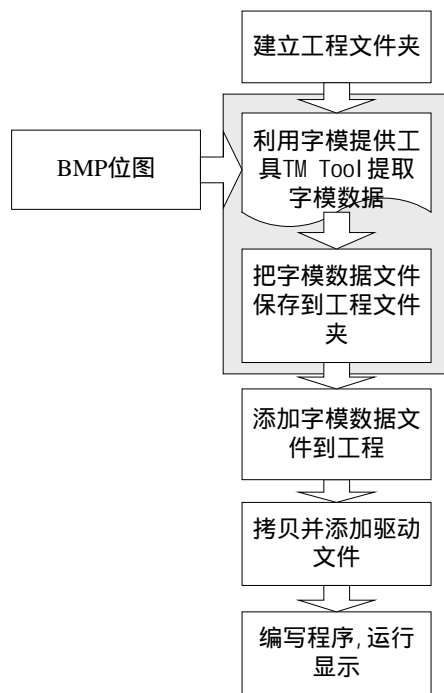


图 3-28 利用 SPLC501 液晶显示模組显示 BMP 位图需要的操作过程

利用 SPLC501 液晶显示模組显示 BMP 位图和显示汉字的过程基本相同，见图 3-29。

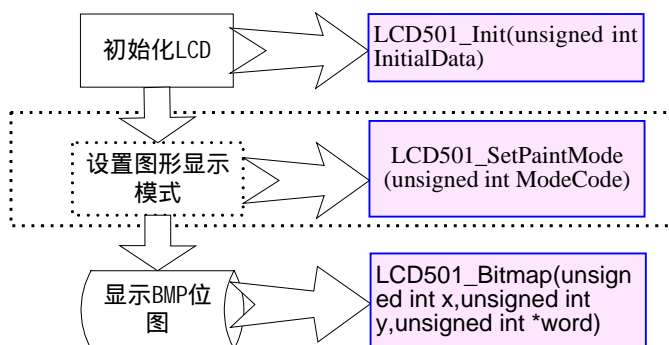


图 3-29 SPLC501 液晶显示模組显示 BMP 位图的过程

4. SPLC501 液晶显示模組显示位置移动图片

当想要让图片在屏幕上移动时，事实上就是让图片在屏幕上的位置改变，比如：图片在 (0, 30) 位置上显示，如果能让图片向右移动 50 个像素点，只要让图片在 (50, 30) 位置上显示即可。如图 3-30。



图 3-30 位置的改变

【硬件连接】

硬件连接图同前面的实验。如图 3-2，分别用跳线连接 IOB3 与 AO，IOB4 与 RWP，IOB5 与 EP，IOB2 与 CS，IOA0~IOA7 与 DB0~DB7。即用跳线把实验仪 LCD (*) 的所有引针全部短接。

【程序流程图】

主程序流程图如图 3-31，初始化 LCD，设置为取反并覆盖的显示模式，显示图片 sp1；进入显示循环，显示图片 sp1，延时 0.2s；显示 sp2，延时 0.2s，依次显示到 sp8，延时 0.2s；显示 sp8，延时 0.2s，显示 sp7，延时 0.2s，依次显示到 sp0，延时 0.2s。每帧图象都显示在 (48, 10) 位置上。

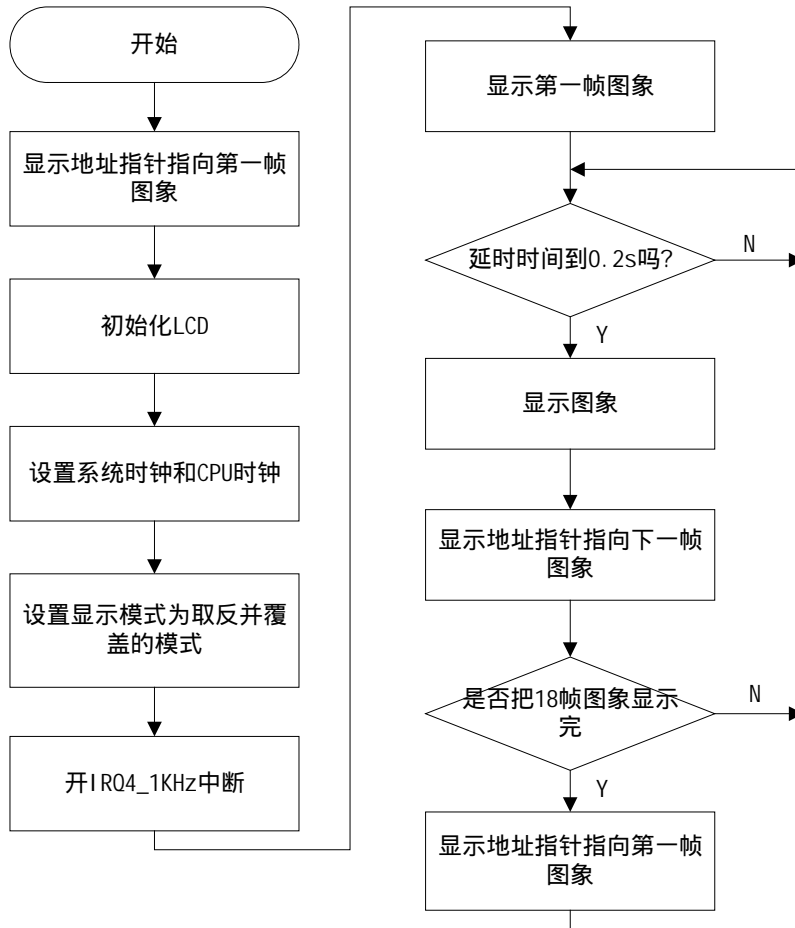


图 3-31 主程序流程图

中断服务流程图如图 3-32，在中断服务程序里，只需要让中断计数器加 1 即可，那么在主程序里，只要计数器等于 200，说明 0.2s 的定时已到，可以进行图像的刷新。

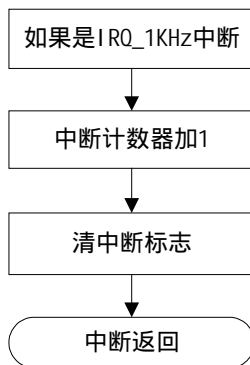


图 3-32 中断服务程序流程图

【实验步骤】

1. 新建一个工程 ex3_LCD501_DynamicGraphic，在工程中新建一个 C 语言文件 main.c。
2. 利用字模提取工具提取图片 sp0~sp9 的字模数据(可以一次性把多个图像导入 DM Tool 当中，转换后都生成在同一个字模数据文件中，方法和实验中转换一个图像相同)，把字模数据文件（字模文件命名为 Sunplus_Table）保存在 ex3_LCD501_DynamicGraphic 工程文件夹。
3. 添加 Sunplus_Table.h 到 ex3_LCD501_DynamicGraphic 工程的“Head Files”，添加 Sunplus_Table.c 到工程的“Source Files”。
4. 拷贝 DataOSforLCD.asm、SPLC501Driver_IO.asm、SPLC501Driver_IO.inc、SPLC501User.c 和 SPLC501User.h 驱动程序文件到 ex3_LCD501_Graphic 工程。
5. 添加 DataOSforLCD.asm、SPLC501Driver_IO.asm 和 SPLC501User.c 文件到“Source Files”，添加 SPLC501Driver_IO.inc 和 SPLC501User.h 文件到“Head Files”。
6. 在 main.c 文件开始包含头文件 SPLC501User.h 和字模数据头文件 Sunplus_Table.h。
7. 按照程序流程图在 main.c 文件里编写 C 语言程序。
8. Rebuild All。
9. 按照硬件连接图连接硬件。
10. 下载程序到实验仪，运行程序。
11. 观察实验仪 SPLC501 液晶显示模组的显示情况，分析是否和实验要求统一。

【范例路径】

在以下路径有可供大家参考的程序。

- ◆ 实验仪指导书范例代码\example\model_Exa\ex3_LCD501_DynamicGraphic。
- ◆ 凌阳科技大学计划网站（www.unsp.com.cn）下载专区可以下载到相关资料。

【练习题】

修改实验中程序，从最左边开始，要求图片转动同时向右移动：后一帧图象的位置是前一帧图象显示位置向右移动一个像素点，比如第一帧图象显示的位置是 (0, 15)，则第二帧图象显示的位置是 (1, 15)。

实验四 LCD 几何图形显示

【实验目的】

1. 熟悉利用 SPLC501 液晶显示模组显示几何图形的 API 函数。
2. 掌握利用液晶显示器显示几何图形的方法。

【实验设备】

1. 装有 Windows 系统和 $\mu'nSP^TM$ IDE 集成开发环境的 PC 机一台，SPCE061A 实验仪一套。
2. 本实验用到的实验仪硬件模块为：CPU 区电路模块，供电电路模块，下载模式选择电路模块，LCD 显示电路模块。

【实验要求】

1. 编程要求：主程序利用 C 语言编写，调用驱动程序（驱动程序已提供在实验仪指导书范例代码\example\model_Exa\driver\SPLC501driver）。
2. 实现功能：在 LCD 显示器实现实心圆和汉字的叠加显示。
3. 实验现象：LCD 显示器上显示一个实心圆，在实心圆的横向直径画一条横线，并在实心圆上叠加显示汉字：“凌阳科技”，最后向上滚屏。显示图形如图 3-33。

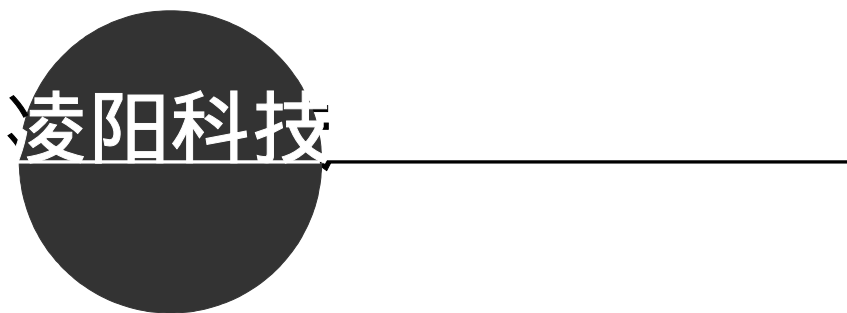


图 3-33 要求显示图形

【实验原理】

1. SPLC501 液晶显示模组几何图形显示 API 函数介绍

驱动程序文件 SPLC501User.c 同样定义了显示几何图形的 API 函数，在 SPLC501User.h 文件里也对这些函数进行了申明。SPLC501 液晶显示模组几何图形显示相关的 API 函数如下：

```
void LCD501_Init(unsigned int InitialData)           //初始化液晶显示
void LCD501_ClrScreen(unsigned int Mode)           //点亮屏幕或清屏
void LCD501_SetPaintMode(unsigned ModeCode)        //设置图形显示模式
unsigned int LCD501_GetPaintMode(void)              //获取图形显示模式
void LCD501_FontSet(unsigned int Font)             //选择显示字符的字体大小
```

```

unsigned int LCD501_FontGet(void) //获得显示字符的字体大小
void LCD501_Bitmap(unsigned int x,unsigned int y,unsigned int *word)//显示汉字或者位图的字模数据
void LCD501_PutPixel(unsigned int x,unsigned int y) //画点
void LCD501_Line(unsigned int x,unsigned int y,unsigned int e_x,unsigned int e_y)//画直线
void LCD501_Circle(unsigned int x,unsigned int y,unsigned int r,unsigned int Mode)//画圆
void LCD501_Rectangle(unsigned left, unsigned top,unsigned right, unsigned bottom, unsigned Mode)
//画矩形
void LCD501_ScrollUp(unsigned Rows) //屏幕向上滚动

```

前面七个函数在前面实验里都有详细说明，这里介绍后四个函数：

1) 【API 格式】C : void LCD501_PutPixel(unsigned int x,unsigned int y)

【功能说明】在液晶显示器上画一个点

【参数】x 表示要画点的 x 坐标；

y 表示要画点的 y 坐标；

【返回值】无

2) 【API 格式】C : void LCD501_Line(unsigned int x,unsigned int y,unsigned int e_x,unsigned int e_y)

【功能说明】在液晶显示器上画一条直线

【参数】x 表示要画的线的 x 起点坐标；

y 表示要画的线的 y 起点坐标；

e_x 表示要画的线的 x 终点坐标；

e_y 表示要画的线的 y 终点坐标。

【返回值】无

3) 【API 格式】C : void LCD501_Circle(unsigned int x,unsigned int y,unsigned int r,unsigned int Mode)

【功能说明】在液晶显示器上画一个圆

【参数】x 表示要画的圆心的 x 坐标；

y 表示要画的圆心的 y 坐标；

r 表示要画的圆的半径；

Mode 表示绘点模式，其中 Mode = M_DOT_CLEAR(0) 画圆框；Mode = M_DOT_SET(1) 画实心圆。

【返回值】无

4) 【API 格式】C : void LCD501_Rectangle(unsigned left, unsigned top,unsigned right, unsigned bottom, unsigned Mode)

【功能说明】在液晶显示器上画一个矩形

【参数】left - 矩形的左上角横坐标，范围 0 到 126

top - 矩形的左上角纵坐标，范围 0 到 62

right - 矩形的右下角横坐标，范围 1 到 127

bottom - 矩形的右下角纵坐标，范围 1 到 63

Mode - 绘制模式，可以是下列数值之一：

PAINT_HOLLOW：矩形框（空心矩形）

PAINT_SOLID：矩形面（实心矩形）

【返回值】无

5) 【API 格式】C : void LCD501_ScrollUp(unsigned Rows)

【功能说明】屏幕向上滚动

【参数】Rows 表示屏幕向上滚动的像素数，范围从 1 到 64

【返回值】无

2. SPLC501 液晶显示模组显示几何图形

利用 SPLC501 液晶显示模组显示几何图形和显示英文字符类似，只有 API 函数不同。如下图。

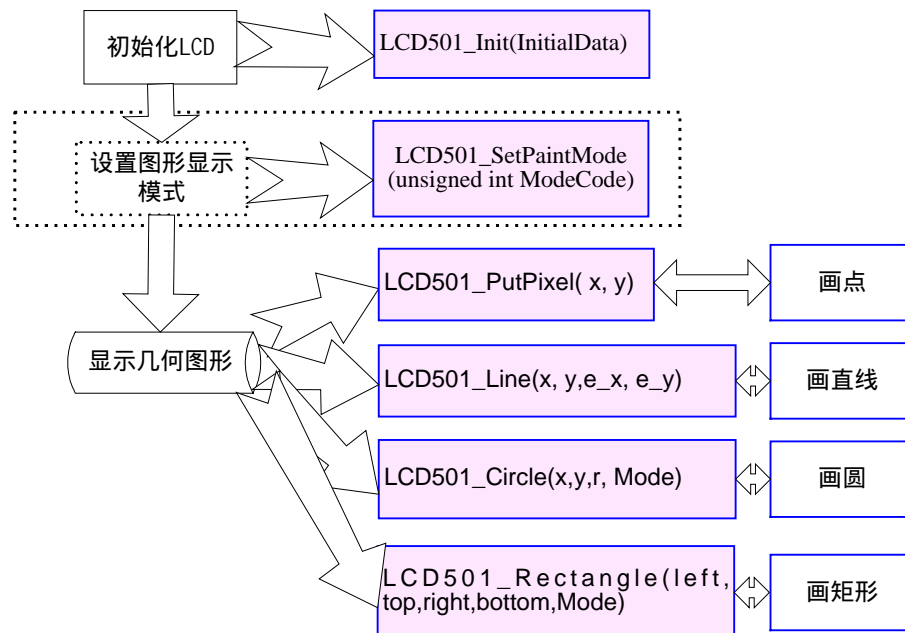


图 3-34 SPLC501 液晶显示模组显示几何图形的过程

【硬件连接】

硬件连接图同本章实验一。如图 3-2，分别用跳线连接 IOB3 与 AO，IOB4 与 RWP，IOB5 与 EP，IOB2 与 CS，IOA0~IOA7 与 DB0~DB7。即用跳线把实验仪 LCD 的所有引针全部短接。

【程序流程图】

主程序流程图如下图：初始化LCD；清屏，延时；在圆心为(30,30)处画一个半径为30的实心圆，延时；设置图象显示模式为异或显示模式；从(0,30)开始到(128,30)结束画一条直线，延时；显示汉字“凌”、“阳”、“科”、“技”，“凌”显示在(0,15)位置，“阳”显示在(16,15)位置，“科”显示在(32,15)位置，“技”显示在(48,15)位置，延时；进入图片滚动显示循环：屏幕向上滚动，每次滚动一个像素点，延时。程序中的延时同样是为了看清实验现象。

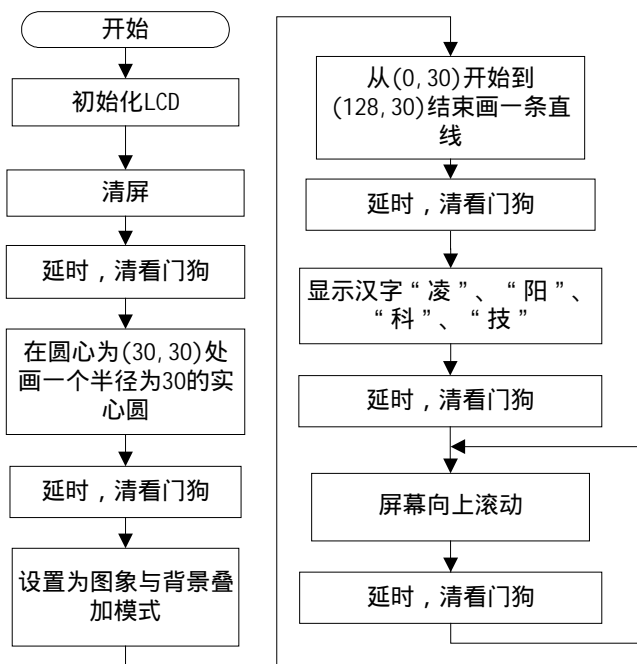


图 3-35 主程序流程图

【实验步骤】

1. 新建一个工程 ex4_LCD501_Ellipse，在工程中新建一个 C 语言文件 main.c 和 isr.asm。
2. 按照实验要求，实验中会显示汉字，利用字模生成工具（DM Tool）生成“凌阳科技”四个字的字模数据，生成数据以 Chinese_Table 命名，保存到 ex4_LCD501_Ellipse 工程。
3. 拷贝 DataOSforLCD.asm、SPLC501Driver_IO.asm、SPLC501Driver_IO.inc、SPLC501User.c 和 SPLC501User.h 五个驱动程序文件到 ex4_LCD501_Ellipse 工程。
4. 添加 Chinese_Table.c 文件到 ex4_LCD501_Ellipse 工程的“Source Files”添加 Chinese_Table.h 文件到“Head Files”。
5. 添加 DataOSforLCD.asm、SPLC501Driver_IO.asm 和 SPLC501User.c 文件到“Source Files”，添加 SPLC501Driver_IO.inc 和 SPLC501User.h 文件到“Head Files”。添加方法和语音实验一中添加“hardware.asm”文件方法一样。
6. 在 main.c 文件开始包含头文件 SPLC501User.h 和字模数据头文件 Chinese_Table.h。
7. 按照程序流程图在 main.c 文件里编写 C 语言程序。
8. Rebuild All。
9. 按照硬件连接图连接硬件。
10. 下载程序到实验仪，运行程序。
11. 观察实验仪 SPLC501 液晶显示模组的显示情况，分析是否和实验要求统一。

【范例路径】

在以下路径有可供大家参考的程序。

- ◆ 实验仪指导书范例代码\example\model_Exa\ex4_LCD501_Ellipse。
- ◆ 凌阳科技大学计划网站（www.unsp.com.cn）下载专区可以下载到相关资料。

【练习题】

编写程序在 SPLC501 液晶显示器上显示下图图片，屏幕向上滚动，每次滚动 1 个像素，并且能利用 2 个键盘控制是否需要滚动。两个键盘的功能为：

K1：屏幕滚动。

K2：屏幕停止滚动。

注：各个几何图形的大小和字体的大小由读者自己定义最适合的大小。



图 3-36 主程序流程图

实验五 UART/USB

【实验目的】

1. 了解 UART/USB 转换口的硬件原理。
2. 熟悉 UART 的各配置单元 P_UART_BaudScalarLow (7024H)、P_UART_BaudScalarHigh (7025H)、P_UART_Command1(7021H)和 P_UART_Command2 (7022H)的功能及控制方法。
3. 掌握简单的 UART/USB 通讯方法。

【实验设备】

1. 装有 Windows 系统和 $\mu'nSP^{TM}$ IDE 集成开发环境的 PC 机一台, SPCE061A 实验仪一套, USB 连接线, 串口调试工具。
2. 本实验用到的实验仪硬件模块为: CPU 区电路模块, 供电电路模块, 下载模式选择电路模块, UART/USB 通讯电路。

【实验要求】

1. 编程要求: 主程序用汇编语言编写。
2. 实现功能 PC 端利用串口调试工具通过 USB 接口把数据发送到实验仪, 实验仪通过 SPCP825 芯片转换, SPCE061A 利用 UART 接收数据; 当 SPCE061A 接收到数据后, 又通过 UART 发送数据, 发送数据经过 SPCP825 芯片转换后通过 USB 接口送给 PC 机, PC 机把接收到的数据显示在串口调试工具的接收区。UART 波特率设置为 9600 bps。
3. 实验现象: 利用 PC 端串口调试工具发送数据, 在接收区可以接收到和发送数据相同的数据。

【实验原理】

实验仪中 UART 通讯方式转换成 USB 通讯方式是通过 SPCP825 芯片实现的。SPCP825A 芯片是凌阳公司设计开发的一款集成 USB、SPI 等通讯模块的 8 位单片机, 可应用于数据的上传和下载, 或者只作为 UART 到 USB、UART 到 PS/2、SPI 到 USB、SPI 到 PS/2 的转换等场合。实验仪的 SPCP825 芯片仅有 USB/UART 转换功能; SPCP825 芯片与 PC 连接后将在 PC 端虚拟出一个新的串口, PC 通过这个新的虚拟串口发送和接收的数据都是通过 USB 接口传输的。

如图 3-37, SPCE061A 发送数据时, 把 UART 通讯模式的数据送到 SPCP825 芯片的 PA6, 转换成 USB 通讯模式后, 通过 USB 接口发送给 PC; 同样, 当 USB 接口接收到 PC 发来的数据时, SPCP825 芯片将数据转换成 UART 模式, 再通过 PA7 发送到 SPCE061A 的 UART 接收口。因此, 虽然实验仪和 PC 机用 USB 连接线连接, 但事实上单片机端与 PC 端使用的都是串口通信程序。

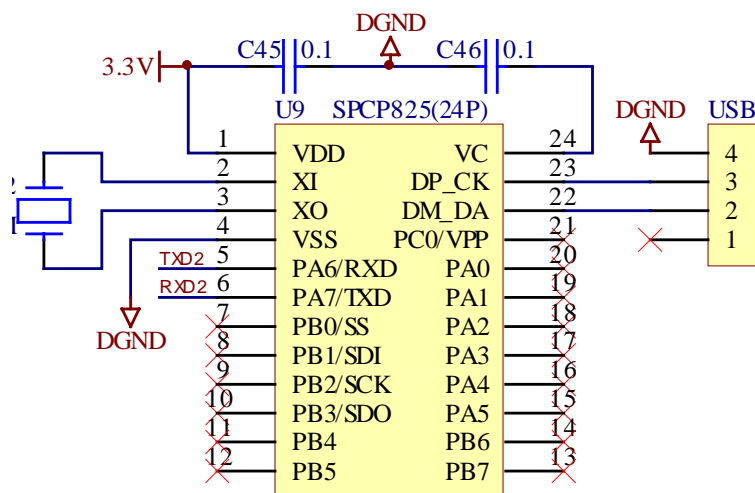


图 3-37 SPCP825 芯片

【硬件连接图】

如图 3-38，把 TXD2 (SPCP825 的 RXD) 和 SPCE061A 的 IOB10 连接，RXD2 (SPCP825 的 TXD) 和 SPCE061A 的 IOB7 连接，即用两个跳线把 COMM 接口靠右边 (标有 USB 字样) 的两个引针短接；用实验仪配套提供的 USB 连接线连接实验仪 USB 接口和 PC 机 USB 接口。

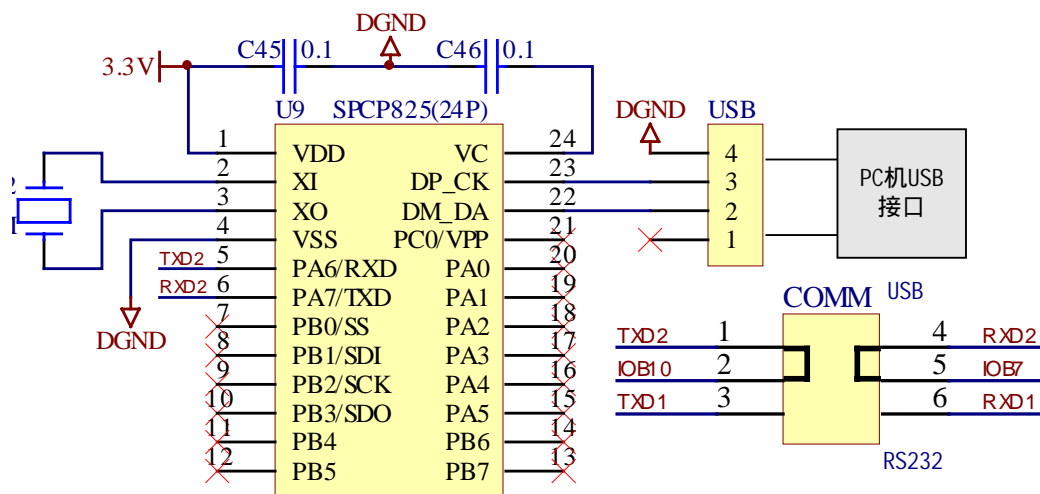


图 3-38 硬件连接图

【程序流程图】

主程序流程图如图 3-39，初始化 IOB7 为输入口，IOB10 为同相高电平输出口，波特率设置成 9600bps，判断是否接收到数据；如接收到了数据，判断发送数据是否准备好，如果准备好则把接收到的数据发送出去。

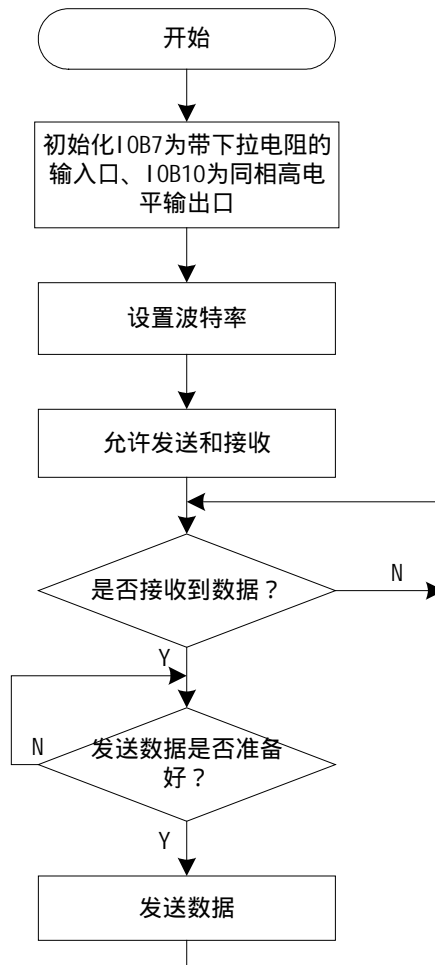


图 3-39 主程序流程图

【实验步骤】

1. 新建一个工程 ex5_UART_USB，在新建工程里新建汇编语言文件 main。
2. 按照程序流程图编写汇编语言程序。
3. 使用 USB 线将实验仪的 USB 接口与 PC 机的 USB 接口连接起来。
4. 然后给实验仪上电，如果在实验之前没有装过 SPCP825 的 PC 端驱动程序，则 Windows 系统会提示安装新硬件。
5. 根据提示安装驱动程序：
第一步，如图 3-40 所示。

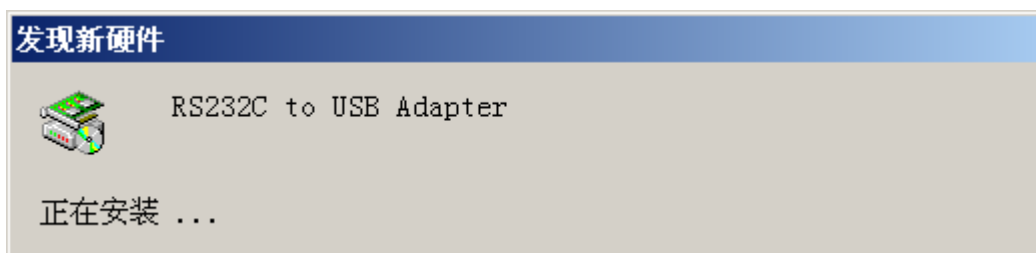


图 3-40 提示安装驱动程序第一步

第二步：如图 3-41，点击“下一步”。



图 3-41 安装驱动程序第二步

第三步：如图 3-42，选择“搜索适于我的设备的驱动程序”，点击“下一步”。

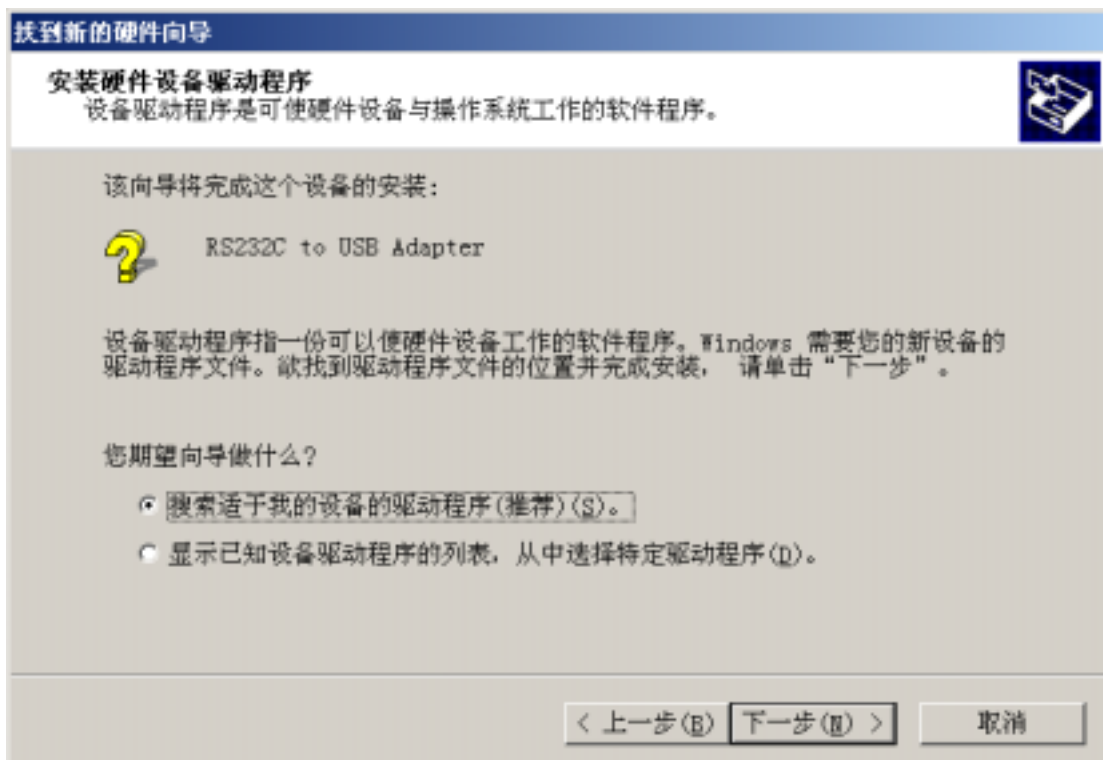


图 3-42 安装驱动程序第三步

第四步：如图 3-43，选择“指定一个位置”，点击“下一步”。

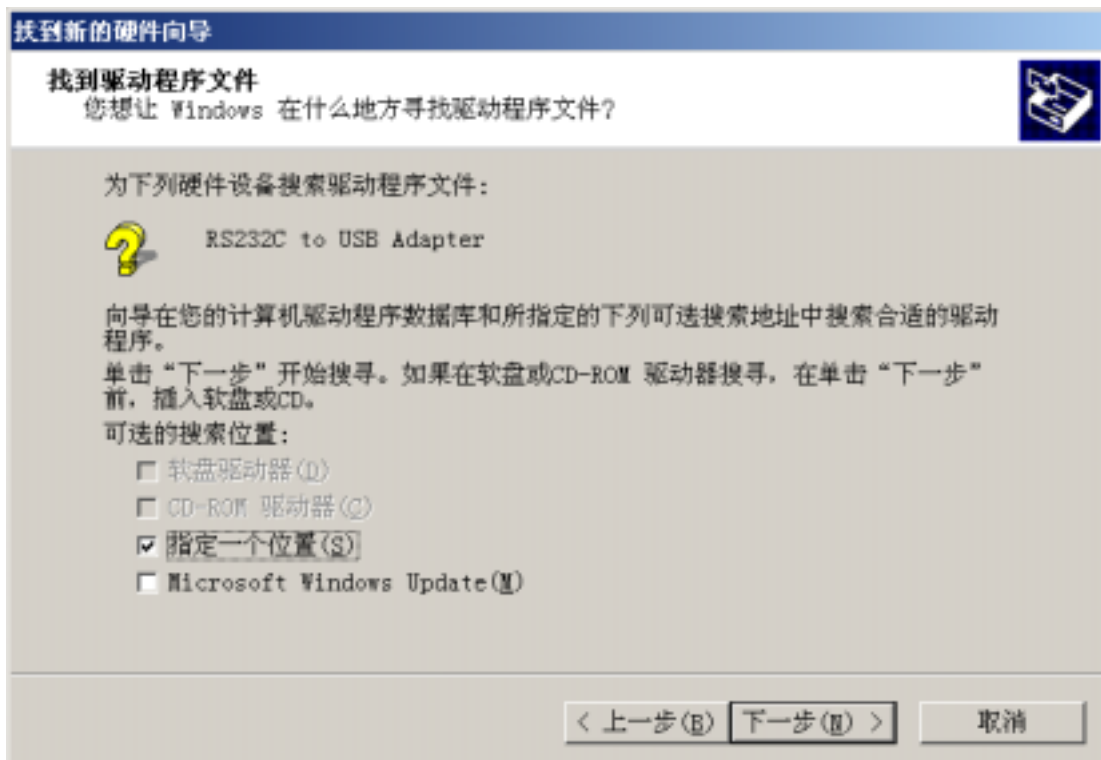


图 3-43 安装驱动程序第四步

第五步 :在实验仪指导书范例代码\example\model_Exa\ex5_UART_USB\pcdriver 找到相应的驱动程序。注意如果 PC 机的运行系统是 Windows2000 或者 WindowsXP，选择 Win2KXP 文件夹里驱动程序；如果 PC 机的运行系统是 Windows98 或者 WindowsME，选择 Win98ME 文件夹里驱动程序。如图 3-44 和图 3-45。

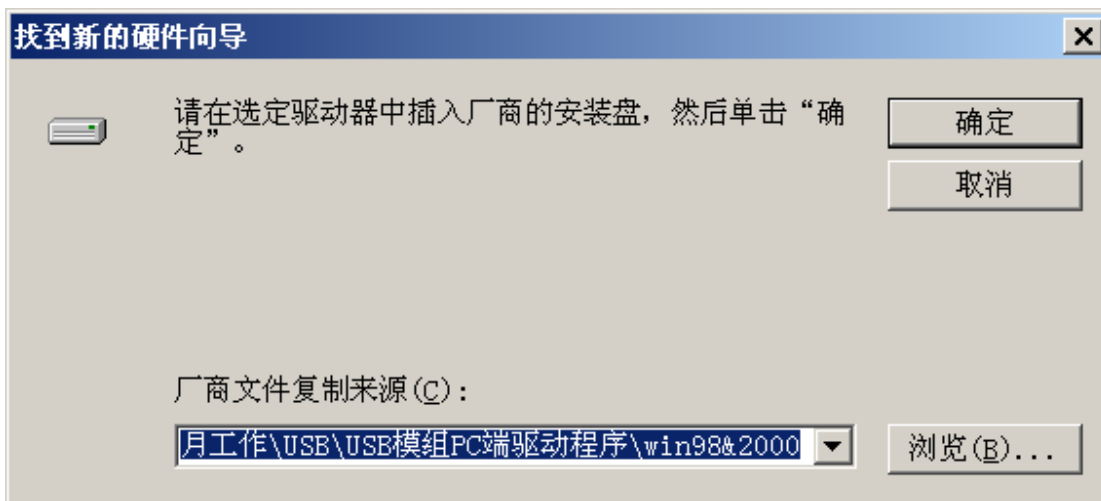


图 3-44 安装驱动程序第五步

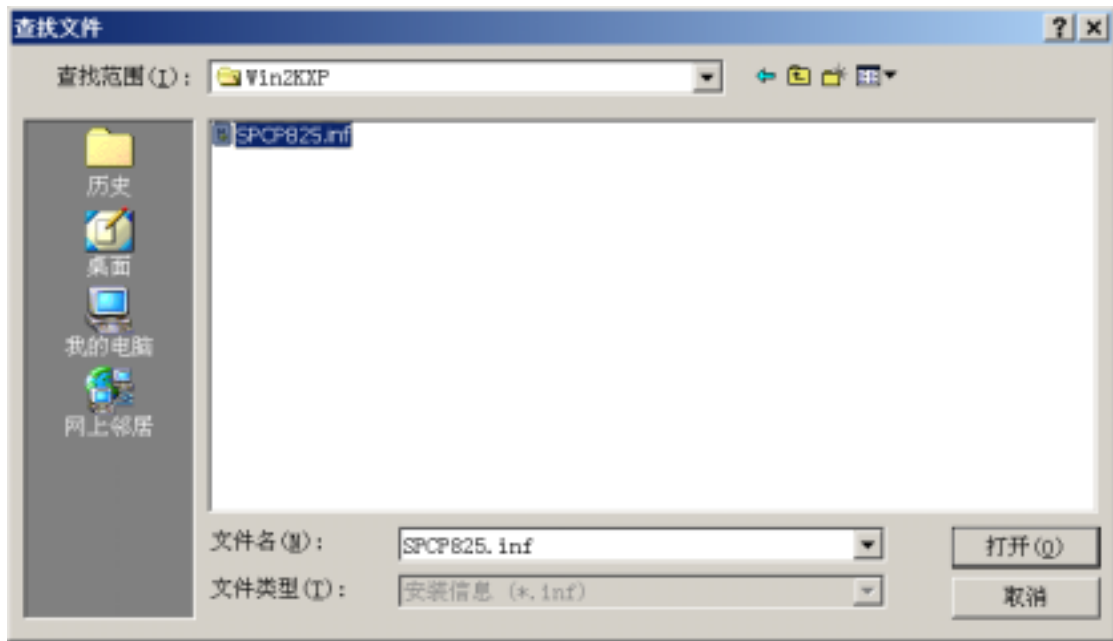


图 3-45 安装驱动程序第五步

第六步：如图 3-46，选择好驱动程序后，点击“下一步”。



图 3-46 安装驱动程序第六步

第七步：如图 3-47，点击“完成”，完成驱动程序安装。



图 3-47 安装驱动程序第七步

注意：也可以在连接 PC 机与 USB/UART 转换口模组之前，点击安装文件（安装文件在实验仪指导书范例代码\example\model_Exa\ex5_UART_USB\pcdriver 提供）进行安装。如图 3-48 和图 3-49 所示：

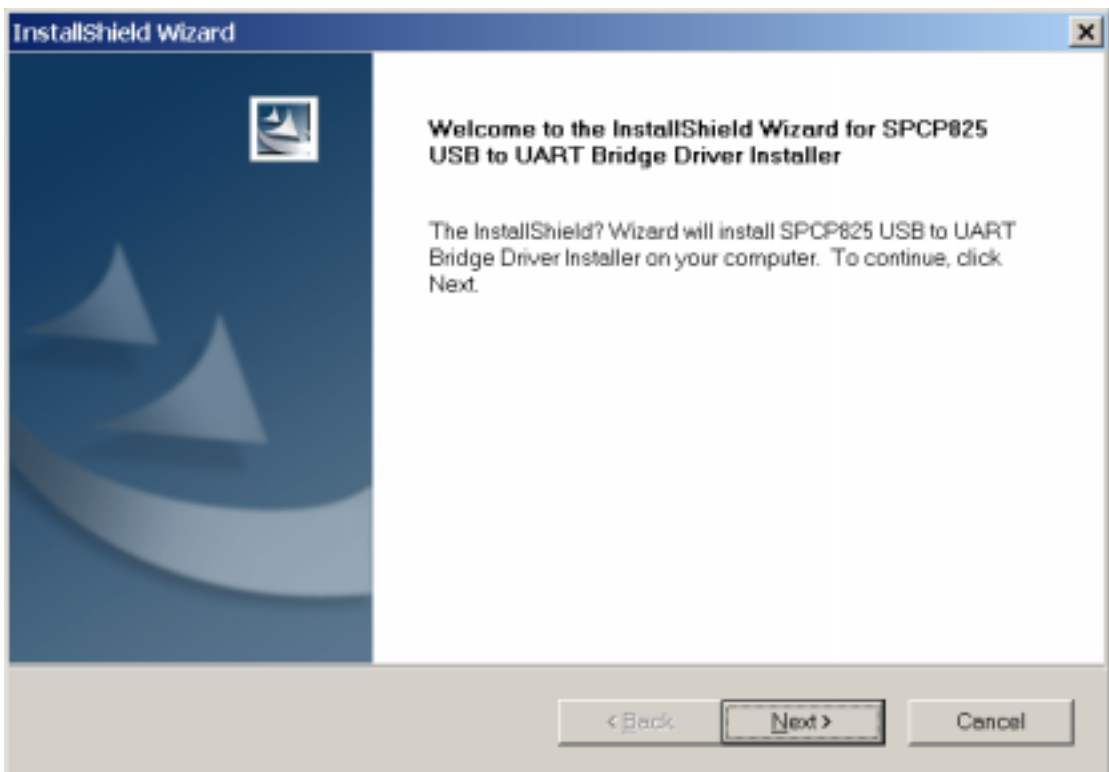


图 3-48 点击文件安装驱动第一步

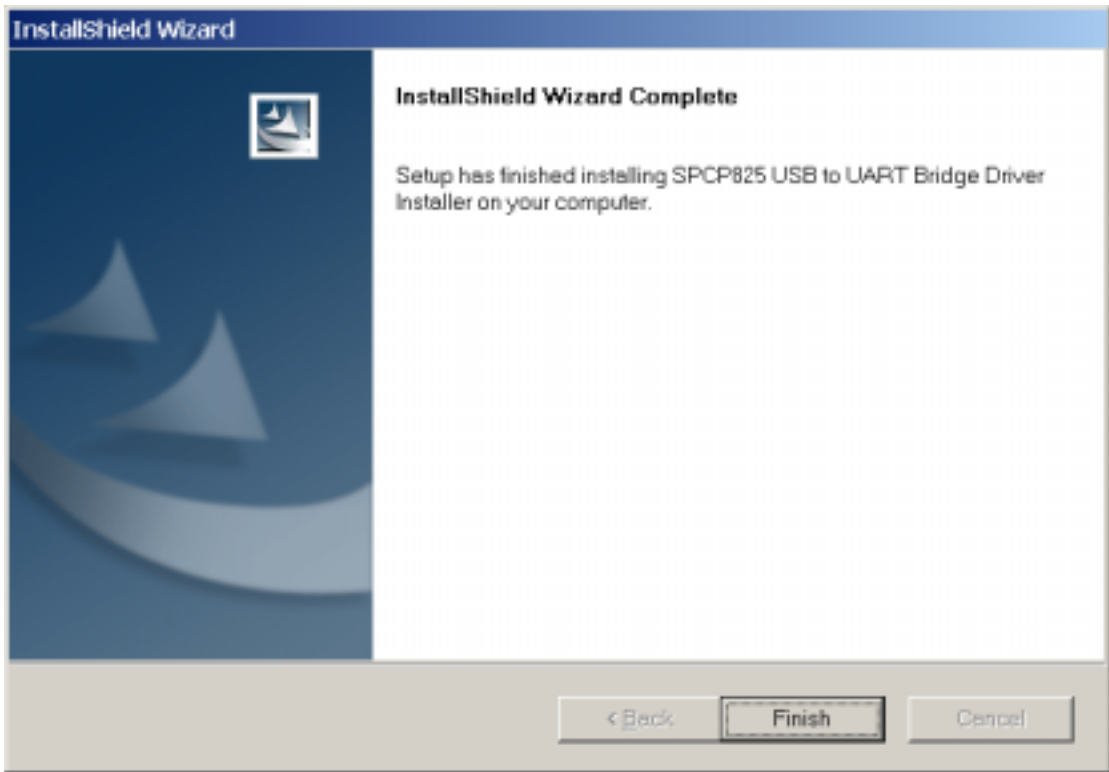


图 3-49 点击文件安装驱动第二步

第八步 :确定 USB/UART 已经与 PC 机正确连接。(在 PC 机资源管理器中查看)如图 3-50 所示 :

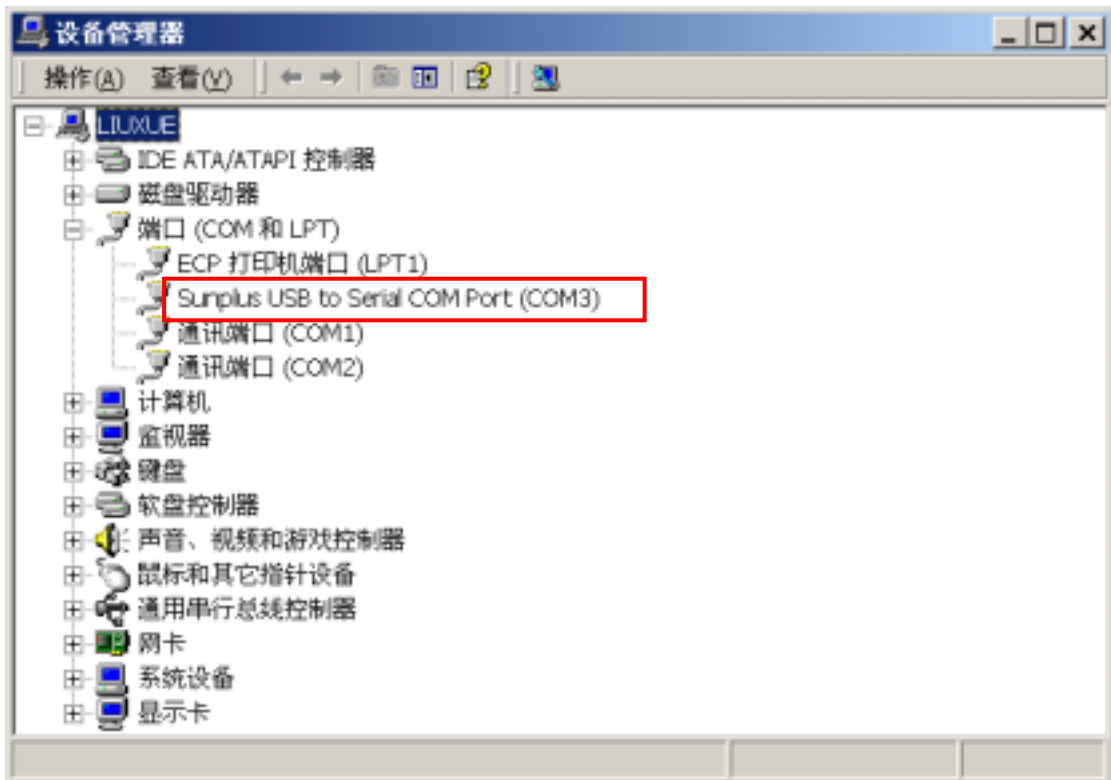


图 3-50 确认正确连接

6. 在 IDE 集成开发环境编译、链接确认没有错误。如图 3-51 所示 :

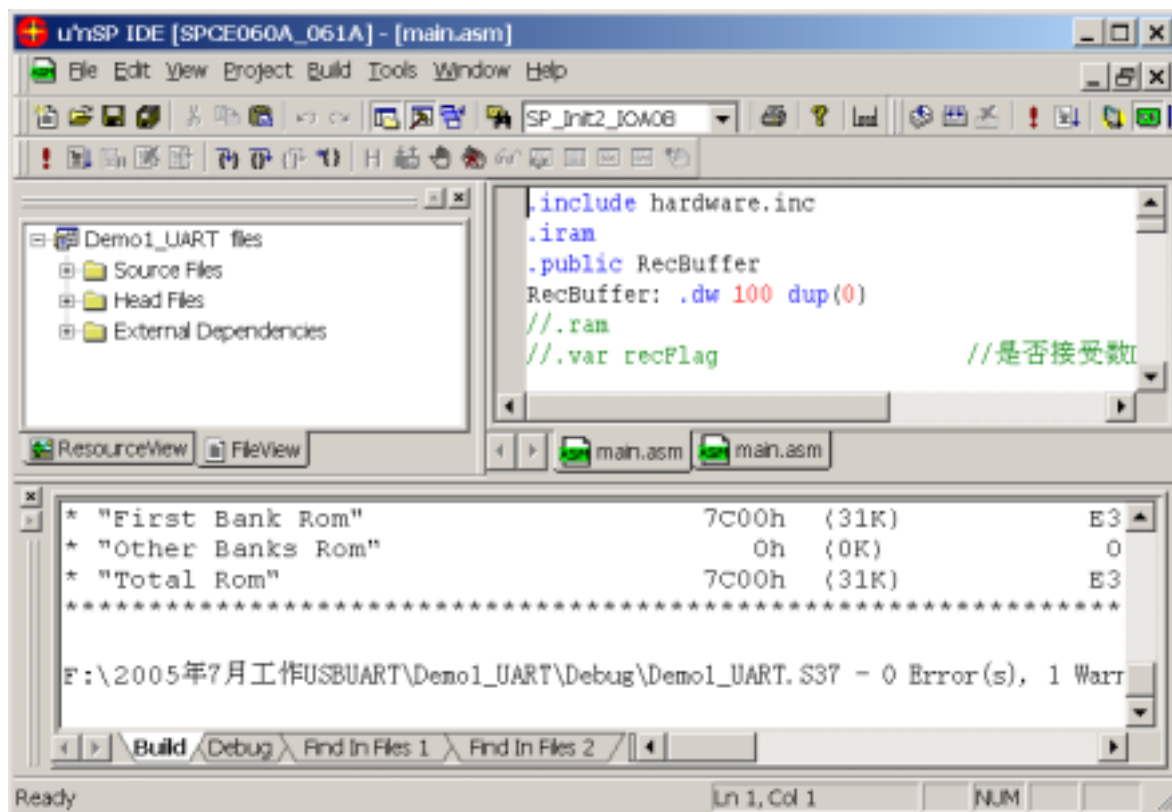


图 3-51 编译、连接图

7. 将正确无误的程序下载到实验仪上。
8. 按照如图 3-38 硬件连接图连接硬件。
9. 运行程序。
10. 打开串口调试工具，这里选择 PortExpert2.0，设置串口调试工具，如图 3-52 所示。

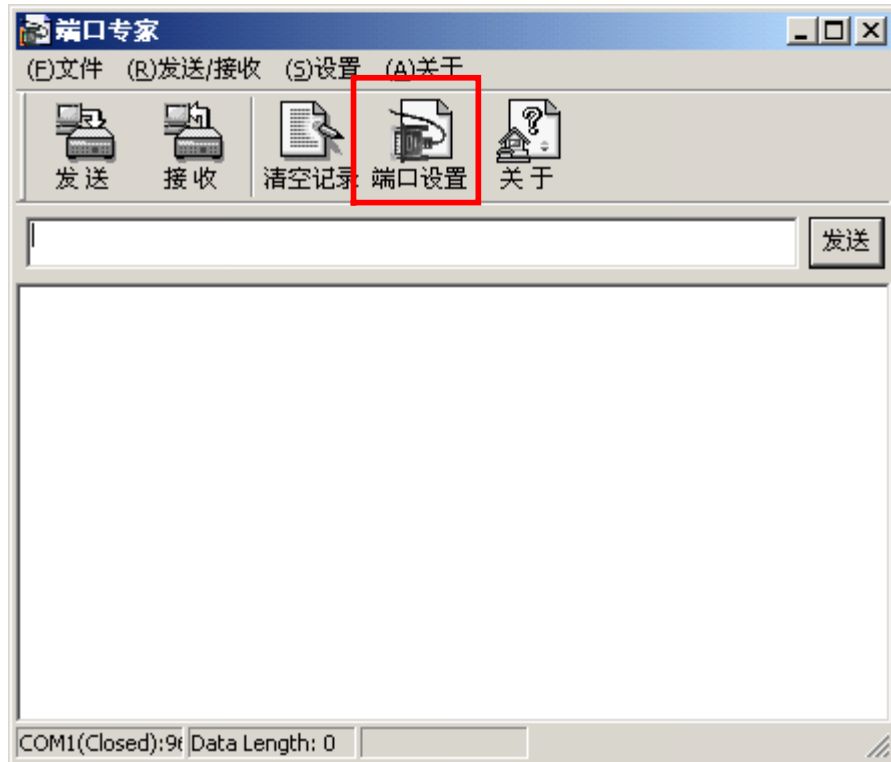


图 3-52 串口调试工具界面

点击“端口设置”后，打开如图 3-53 界面，端口设置为“COM3”，因为上面的安装驱动步骤中安装了“Sunplus USB to Serial COM Port (COM3)”，如图 3-50；设置波特率为“9600”；数据位设置为“8”。



图 3-53 端口设置界面

11. 发送数据：在串口调试工具的发送区输入要发送的字符，点击“发送”，如图 3-54。

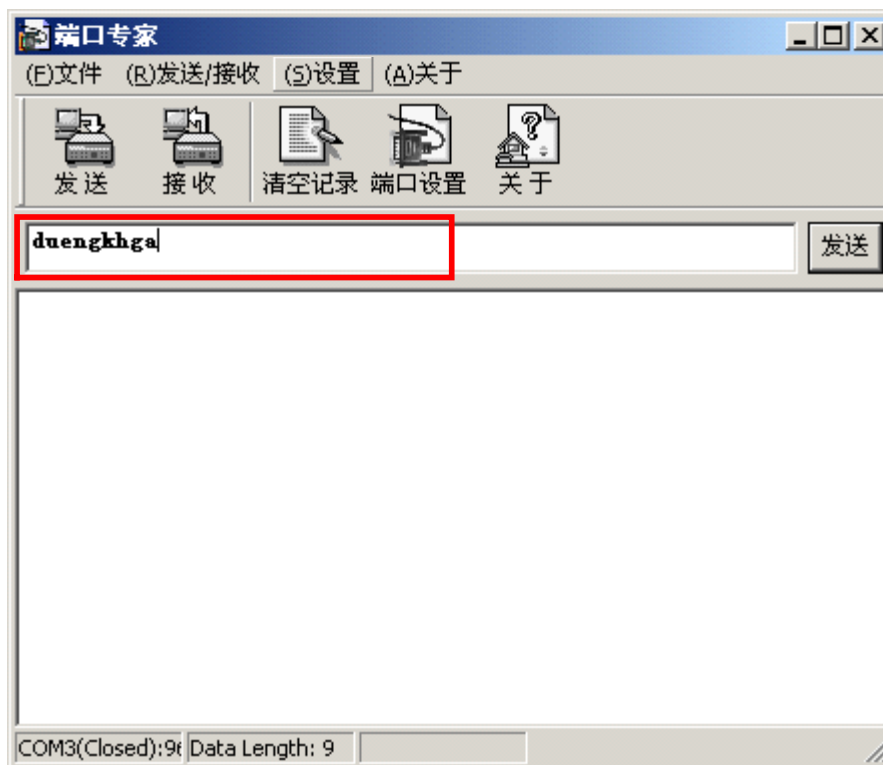


图 3-54 发送数据

12. 接收数据：如图 3-55，点击“接收”，在接收区就可以接收到和发送区相同的数据，观察串口调试工具接收区的数据，比较是否和发送的数据相同。

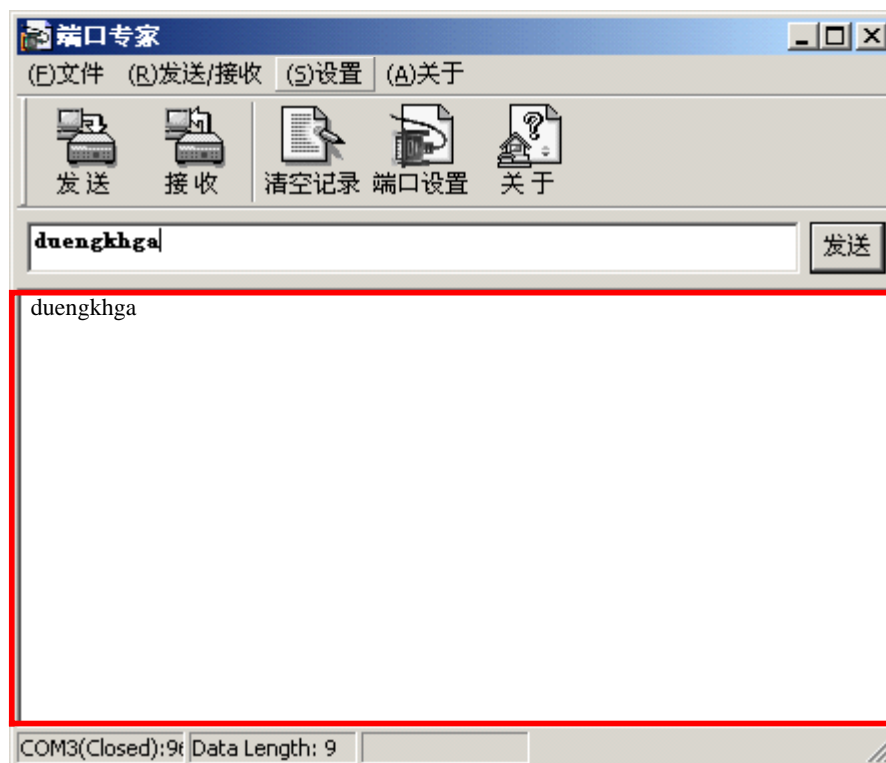


图 3-55 接受数据

【范例路径】

在以下路径有可供大家参考的程序。

- ◆ 实验仪指导书范例代码\example\model_Exa\ex5_UART_USB。
- ◆ 凌阳科技大学计划网站 (www.unsp.com.cn) 下载专区可以下载到相关资料。

实验六 SPR4096A FLASH 的擦除及其读写

【实验目的】

1. 了解 SIO 的基本使用方法。
2. 掌握通过 SIO 扩展 Flash 的方法。
3. 学会读写和擦除 SPR4096A Flash。

【实验设备】

1. 装有 Windows 系统和 $\mu'nSP^TM$ IDE 集成开发环境的 PC 机一台，SPCE061A 实验仪一套。
2. 本实验用到的实验仪硬件模块为：CPU 区电路模块，供电电路模块，下载模式选择电路模块，SPR4096A 存储电路模块。

【实验要求】

1. 编程要求：主程序利用 C 语言编写，调用驱动程序（驱动程序已提供在实验仪指导书范例代码\example\model_Exa\driver\SPR4096Adriver ），
2. 实现功能：检测 SPR4096A Flash 读写数据是否正确：读出 0x0000 地址单元数据，输出控制发光二极管显示，读出数据加 1，写入 0x0000 地址单元；复位后或者断电后再上电时读出 0x0000 地址单元数据，输出控制发光二极管显示，根据发光二极管显示状态判断读出数据是否正确。这样相当于在 SPR4096A 的 Flash 里面保存了一个记忆开机次数的数据。
3. 实验现象：运行，发光二极管全部点亮；按复位键，所有发光二极管被熄灭；按复位键，第一个发光二极管点亮；再按复位键，第二个发光二极管点亮；接着按复位键，第一个和第二个发光二极管点亮；依次类推，按复位键后发光二极管显示状态对应的数据是复位前显示状态对应的数据加 1。即二极管显示开机或者复位合起来的总次数。点亮与熄灭状态顺序如下表 3-1 所示。其中，“ ”表示二极管是点亮状态，“ ”表示二极管是熄灭状态。表中 LED1~LED8 为实验仪上发光二极管电路模块中从左到右的 8 个发光二极管。

表 3-1 8 个发光二极管状态

序号	LED8	LED7	LED6	LED5	LED4	LED3	LED2	LED1
0								
1								
2								
3								
4								
5								
6								

7								
8								
9								
10								
.....
254								
255								
256								
257								
.....

【实验原理】

1. SPR4096A 工作原理

SPR4096A 是一个高性能的 4M 位 (512K × 8 位)FLASH，分为 256 个扇区(Sector)，每个扇区为 2K 字节。SPR4096A 还内置了一个 4K × 8 位的 SRAM。

SPR4096A 串行接口的工作频率可达 5MHz。SPR4096A 有两个电源输入端 VDDI 和 VDDQ。VDDI 是给内部 FLASH 和控制逻辑电路供电的；VDDQ 是专门为 I/O 供电的。供电电压分别为 VDDI：2.25V ~ 2.75V，VDDQ：2.25V ~ 3.6V。

SPR4096A 以串行接口模式工作，CF2 ~ CF0 均接高电平。CF7 为低电平时选中 FLASH，高电平时选中 SRAM。本实验中 CF7 外接一个选择接口，当 CF7 和 DGND 连接时选择 SPR4096A 的 FLASH，CF7 和 3.3V 连接时选择 SPR4096A 的 SRAM。如图 3-56 SPR4096A 的电路原理图。

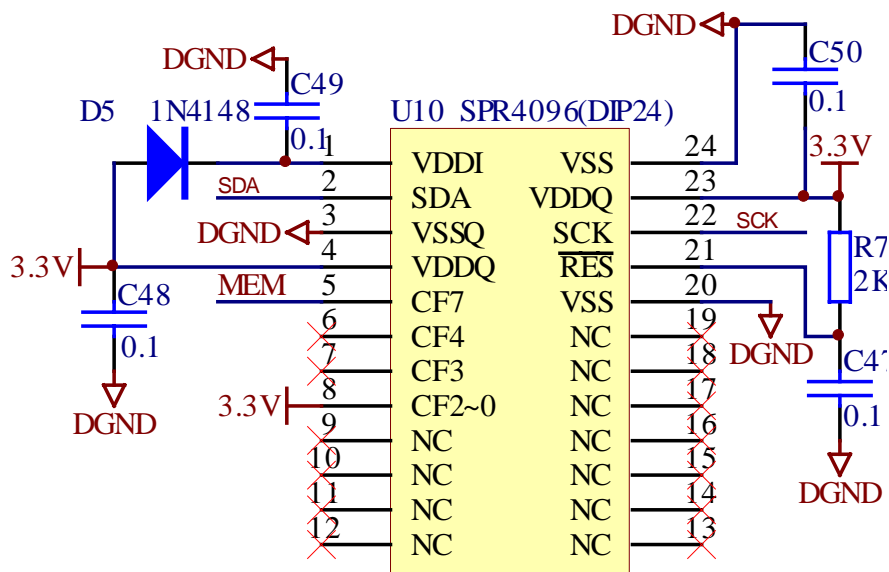


图 3-56 实验仪的 SPR4096A 模块原理图

2. SPR4096A 的 API 函数

凌阳科技大学计划提供了 SPR4096A 的读写驱动程序 4096.asm 文件（提供在实验仪指导书范例代码\example\model_Exa\Driver\SPR4096Adriver），文件里包含 SPR4096A 的 API 函数：

```
void SP_SIOInitial(void) //初始化 SIO
void SP_SIOMassErase(void) //擦除 SPR4096A 的所有扇区
void SP_SIOSectorErase(unsigned int uiSector) //擦除 SPR4096A 的一个扇区
void SP_SIOSendAByte (unsigned long int ulAddr,unsigned int uiData)//写一个字节
void SP_SIOSendAWord(unsigned long int ulAddr,unsigned int uiData)//写一个字
unsigned int SP_SIOReadAByte(unsigned long int ulAddr) //读一个字节
unsigned int SP_SIOReadAWord (unsigned long int ulAddr) //读一个字
```

这些函数的说明如下：

1) 【API 格式】C：void SP_SIOInitial(void)

【功能说明】初始化 SIO

【参数】无

【返回值】无

2) 【API 格式】C：void SP_SIOMassErase(void)

【功能说明】擦除 SPR4096A 的所有扇区。

【参数】无

【返回值】无

3) 【API 格式】C：void SP_SIOSectorErase(unsigned int uiSector)

【功能说明】擦除 SPR4096A 的一个扇区。

【参数】uiSector 为扇区的编号，0~255 可选。

【返回值】无。

4) 【API 格式】C：void SP_SIOSendAByte (unsigned long int ulAddr,unsigned int uiData)

【功能说明】往 SPR4096A 写入一个字节的的数据。

【参数】ulAddr 为写入地址，uiData 为写入数据。

【返回值】无。

5) 【API 格式】C：void SP_SIOSendAWord(unsigned long int ulAddr,unsigned int uiData)

【功能说明】往 SPR4096 写入一个字的数据。

【参数】ulAddr 为写入地址，uiData 为写入数据。

【返回值】无。

6) 【API 格式】C：unsigned int SP_SIOReadAByte(unsigned long int ulAddr)

【功能说明】从 SPR4096 读出一个字节的数据。

【参数】ulAddr 为读出地址。

【返回值】读出的数据。

7) 【API 格式】C：unsigned int SP_SIOReadAWord (unsigned long int ulAddr)

【功能说明】从 SPR4096 读出一个字的数据。

【参数】ulAddr 为读出地址。

【返回值】读出的数据。

3. 驱动程序流程图

上面介绍了驱动函数的 API 函数，下面分别是各个函数的程序流程图。这些数据详细的传输时序可参考 SPR4096AV12 DataSheet（可以在凌阳大学计划网站的下载专区下载）。这里详细讲解写数据的操作原理。

图 3-57 是 SPCE061A SIO 的读写操作时序图。

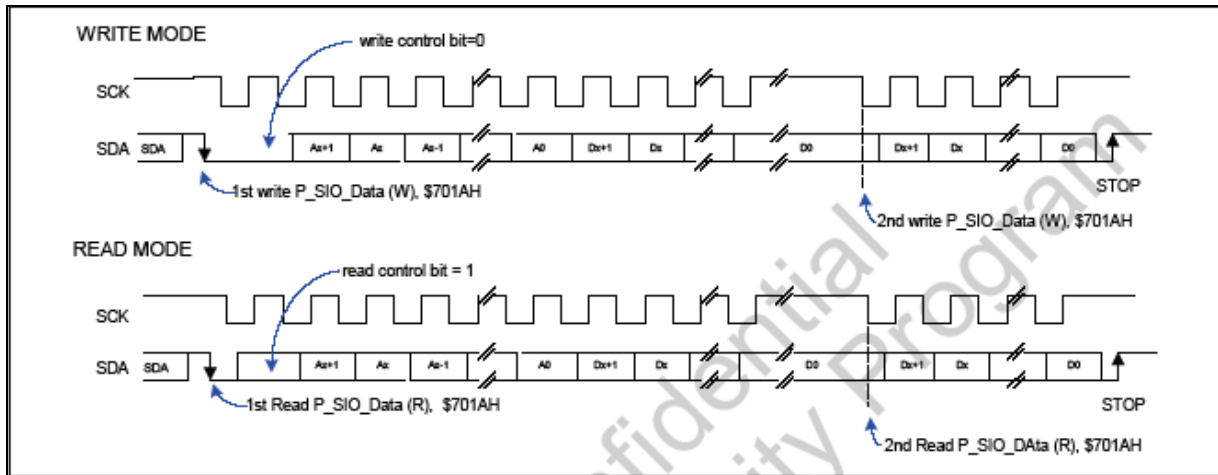


图 3-57 SPCE061A SIO 的时序图

仅看 SIO 的时序图，当需要写数据的时候，先送一个写控制位 bit=0，接着写入 FLASH 的地址，然后送数据。

图 3-58 图是 SPR4096A 写命令字时序图。

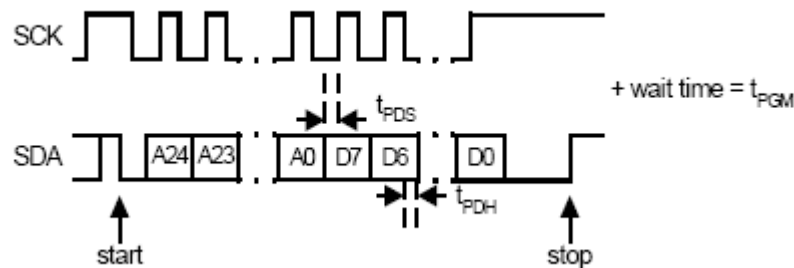
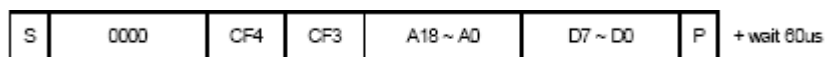


图 3-58 SPR4096A 写命令字时序

仅看 SPR4096A 写操作的时序图，它的命令字有一个 start 位，24 个地址位和 8 和数据位构成，即一次接收 33 位。所以在写数据时，它接收的第一位是 start 位，后面接着 24 位是地址数据，接着的 8 位才是写入的数据。下面是写控制字的各位功能。前面“0000”加上“CF4CF3”再加上 19 位地址信号和 8 位数据信号刚好是 33 位，也就是说事实上这时候只有低十六位地址有效。对应到 SPCE061A SIO 中 SDA 传输的数据：SPR4096A 写控制字的“0000”中后面三位、CF4、CF3、A18~A16 对应 SIO 的 A23~A16，A15~A0 对应 A15~A0，D7~D0 对应 D7~D0，然后发送一个结束标志位，延时等待 60us。



其他操作和写操作的原理类似，基于这些原理，下面介绍对 SPR4096A 进行各种操作的流程。

a) 初始化程序流程图

把与 SCK、SDA 和 CF7 连接的 IOB0、IOB1、初始化为同相高电平输出口，并把系统时钟初始化为 49MHz。如图 3-59。

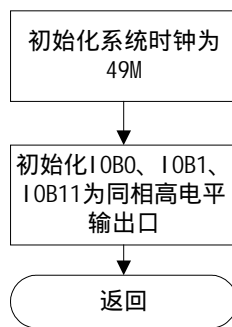


图 3-59 初始化程序流程图

b) 擦除整块 FLASH 程序流程图

其中，低字节地址、中字节地址和高字节地址合起来就形成了擦除整块 FLASH 的命令，由于串行设备地址为 16 位，所以事实上高字节地址并没有用到，而是在后面传输了一个字节的 0。如图 3-60。

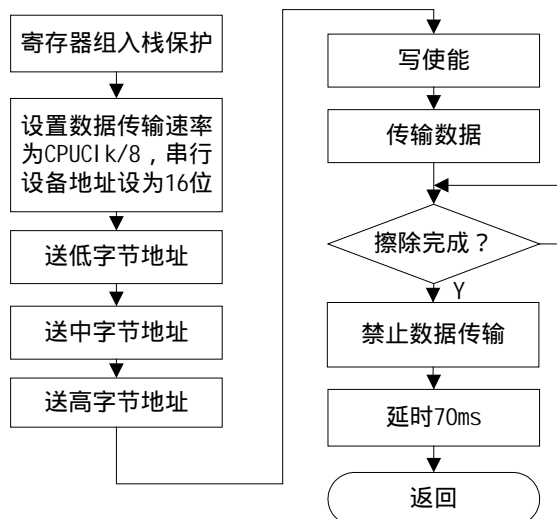


图 3-60 擦除整块 FLASH 程序流程图

c) 擦除一个扇区程序流程图

擦除一个扇区的程序流程和擦除整块 FLASH 类似，只不过由低字节地址、中字节地址和后面传输的一个数据组成的擦除命令不同而已。如图 3-61。

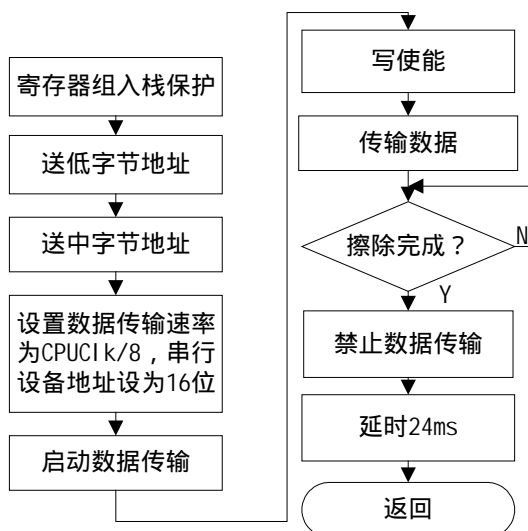


图 3-61 擦除一个扇区程序流程图

d) 向 FLASH 写入一个字节数据程序流程图

从上面的时序知道，在向 FLASH 写入一个字节的的数据时，先写入 FLASH 地址，再写入一个字节的的数据。如图 3-62。

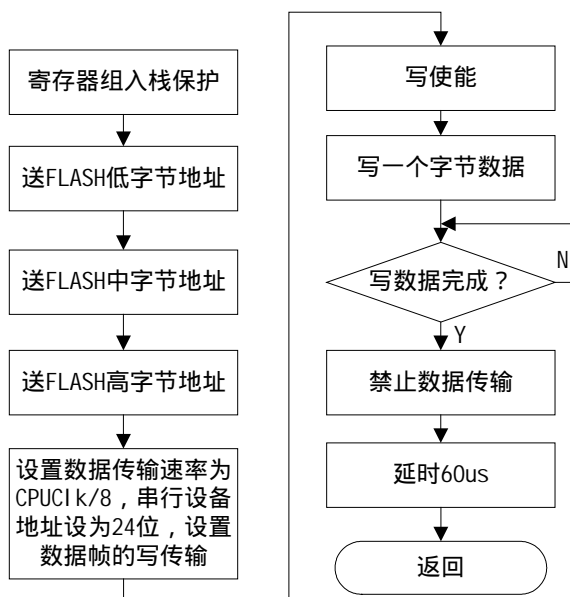


图 3-62 向 FLASH 写入一个字节数据程序流程图

e) 从 FLASH 读出一个字节数据程序流程图

读数据是写数据的反过程，先送由 FLASH 三个字节地址组成的命令字，再读出一个字节的数据。如图 3-63。

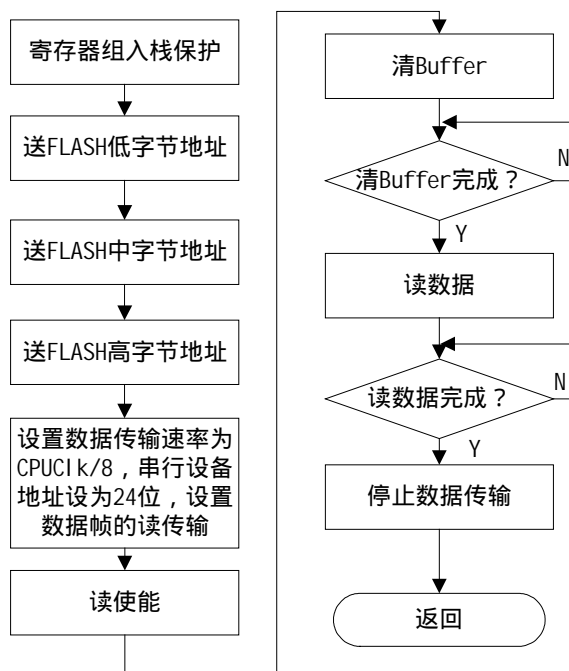


图 3-63 从 FLASH 读出一个字节数据程序流程图

f) 向 FLASH 写入一个字数据程序流程图

写一个字数据事实上相当于写两个字节数据即低字节数据和高字节数据的过程，只不过只需要送一次写字节数据的命令，在写高字节数据的时候不需要再送一次写字节数据的命令。如图 3-64。

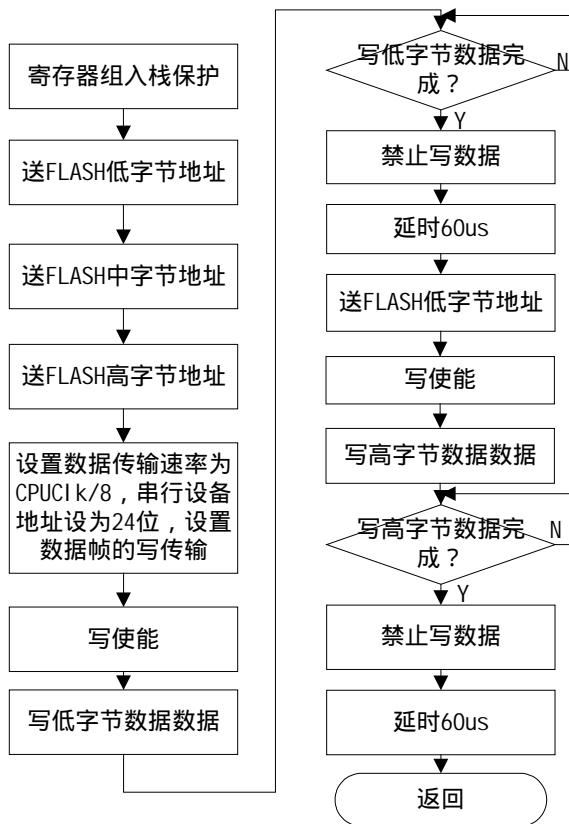


图 3-64 向 FLASH 写入一个字数据程序流程图

g) 从 FLASH 读出一个字数据程序流程图

从 FLASH 读出一个字数据同样只需要送一次读字节数据命令，只不过要读两次数据，即低字节数据和高字节数据。如图 3-65。

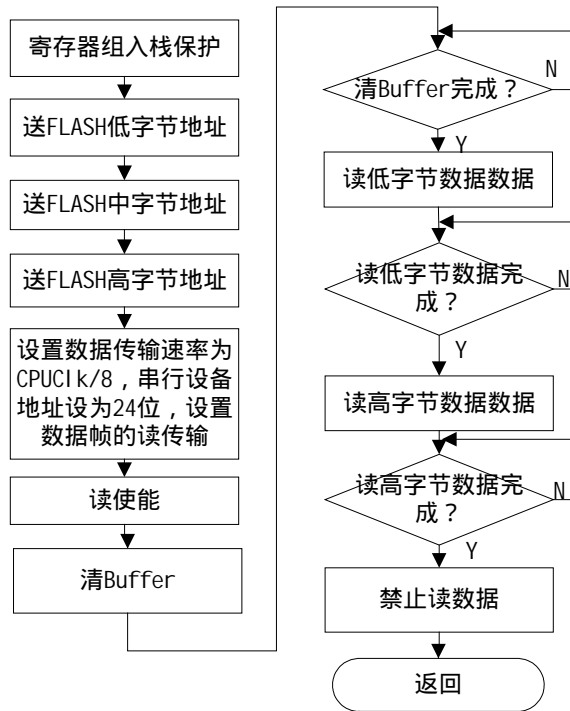


图 3-65 从 FLASH 读出一个字数据程序流程图

【硬件连接】

如图 3-66 硬件连接图 IOA 口低八位 IOA0~IOA7 连接 8 个 LED 的 a~h，即用跳线短接 LED_SEG 的左右两排引针；IOB6 连接 LED_DIG 的 DIG5，即把 LED_DIG 的最下面一排引针用跳线短接；IOB0 连接 SPR4096A 的 SCK，IOB1 连接 SPR4096A 的 SDA，即用跳线短接 4096PORT 的左右两排引针；SPR4096A 的 CF7 连接 DGND，即用跳线把 CF7 引针和 DGND 引针短接。

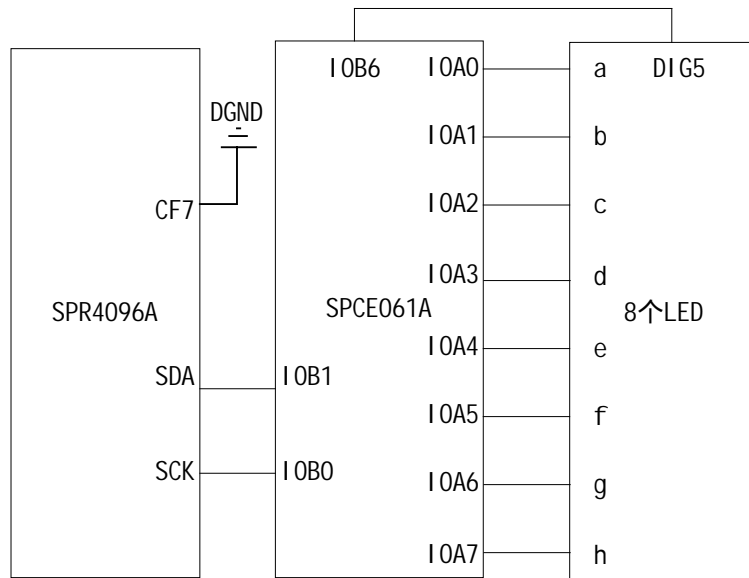


图 3-66 硬件连接图

【程序流程图】

主程序流程图如图 3-67。初始化 SIO，初始化 A 口低八位为同相低电平输出口，读出 0x0000 地址数据（读出数据为上次写入 0x0000 地址的数据），送读出数据到 IOA 口控制发光二极管显示；擦除 0x0000 地址的数据；把读出显示的数据加 1，写入 0x0000 地址单元；进入主程序循环，执行清看门狗操作。

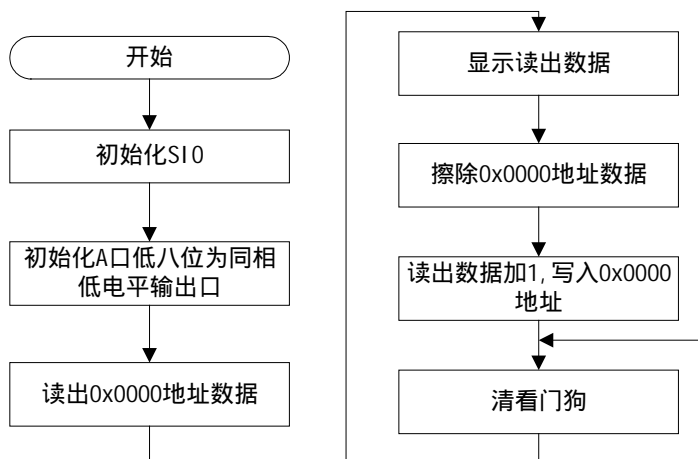


图 3-67 主程序流程图

【实验步骤】

1. 新建一个工程 ex6_SPR4096_FLASH，在新工程里新建一个 C 语言文件 main.c。
2. 拷贝驱动程序 4096.asm 到 ex6_SPR4096_FLASH 工程文件夹。（4096.asm 在实验仪指导书范例代码\example\model_Exa\Driver\SPR4096Adriver 路径可以找到）。
3. 添加 4096.asm 到“Source Files”，编写程序时就可以直接调用 4096.asm 里面的 API 函数。
4. 在 main.c 文件里编写 C 语言程序。
5. Rebuild All。
6. 按照硬件连接图连接电路，注意 CF7 选择 DGND。
7. 下载程序到实验仪，运行。
8. 按下实验仪的复位键，观察发光二极管的状态变化，分析是否实验要求统一。

【范例路径】

在以下路径有可供大家参考的程序。

- ◆ 实验仪指导书范例代码\example\model_Exa\ex6_SPR4096_FLASH。
- ◆ 凌阳科技大学计划网站（www.unsp.com.cn）下载专区可以下载到相关资料。

【练习】

修改实验程序，要求写入 SPR4096A 一段音乐，写完后读出音乐数据，利用手动播放的方式播放出来。

第4章 综合实验

实验一 4位7段LED数码管显示

【实验目的】

1. 了解4位7段LED数码管的工作原理。
2. 熟悉并进一步掌握定时器和时基信号的使用方法。
3. 掌握SPCE061A单片机控制LED数码管显示的方法。

【实验设备】

1. 装有Windows系统和 $\mu'nSP^TM$ IDE集成开发环境的PC机一台，SPCE061A实验仪一套。
2. 本实验用到的实验仪硬件模块为：CPU区电路模块，供电电路模块，下载模式选择电路模块，4位数码管电路模块。

【实验要求】

1. 编程要求：主程序利用C语言编写，中断服务程序利用汇编语言编写。
2. 实现功能：SPCE061A单片机控制4个LED数码管的显示。
3. 实验现象：运行开始点亮所有的数码管，4位LED数码管均显示0并持续1s。1s后，第一位数码管从0显示到9，刷新时间为0.5s，其他数码管全部显示0。当第一位数码管显示到9后，第一位数码管保持显示9，第二位数码管从0显示到9，刷新时间为0.5s，其他数码管显示0。依次直到第4位数码管显示9，即4位数码管全部显示9，4位数码管全部显示0，持续1s，如此循环。（1s的时间、0.5s的时间都使用2Hz的时基信号(IRQ5)提供）。4位数码管的显示状态见表4-1。

表 4-1 6 位数码管显示状态

序号	第 1 位数码管	第 2 位数码管	第 3 位数码管	第 4 位数码管
0	0	0	0	0
1~10	0~9	0	0	0
11~20	9	0~9	0	0
21~30	9	9	0~9	0
31~40	9	9	9	0~9
41	0	0	0	0
42~51	0~9	0	0	0
52~61	9	0~9	0	0
.....

注意：表中序号为 0 和 41 的状态持续 1s；其他状态刷新时间为 0.5s。

表中需要类似“1~10”表示有十个状态。

【实验原理】

1. 4 位 8 段数码管工作原理

实验仪上的数码管示意图如图 4-1，实验仪实际电路图如图 4-2。

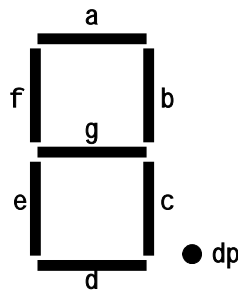


图 4-1 数码管示意图

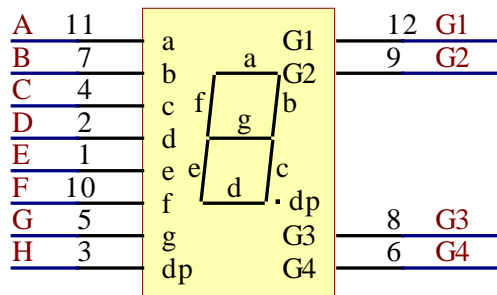


图 4-2 数码管电路图

首先介绍两个基本概念：段码和位码，段码即段选信号 SEG，它负责数码管显示的内容，图 4-1 中 a~g、dp 组成的数据（a 为最低位，dp 为最高位）就是段码。比如 1 的段码“0x06”（b=1，c=1，其他都为 0，即段码为 00000110b），8 的段码为“0x7f”；位码即位选信号 DIG，它决定哪个数码管工作，哪个数码管不工作，比如仅使能 DIG4，那么 4 个 LED 只有 LED4 工作，而其它的三个都不工作。

当需要某一位数码管显示数字时，只需要先选中这位数码管的位信号，再给显示数字的段码。比如当在第一个数码管上显示一个“6”时，如图 4-3，先选中第一位数码管的位信号（实验仪上标号是“DIG1”），即先给和“DIG1”相连接的 I/O 口送 1；再把段码设置为 0x007d，即在 a、c、d、e、f、g 各段引出的端口为高电平，就可以显示一个“6”出来。

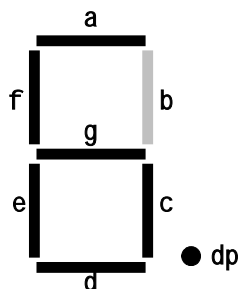


图 4-3 数码管显示“6”

2. 实验仪上 SPCE061A 控制 4 位 8 段数码管的显示

实验仪在 4 位 8 段数码管的左面有 LED_SEG 和 LED_DIG 两个排针接口，其中 LED_SEG 是控制此 4 位 8 段数码管的段码选择的，LED_DIG 是控制 4 位 8 段数码管位选择和发光二极管的公共端选择的，“h”控制小数点。把实验仪上 LED_SEG 所有两排引针和 LED_DIG 靠上面四排引针用跳线短接。框图如图 4-4。

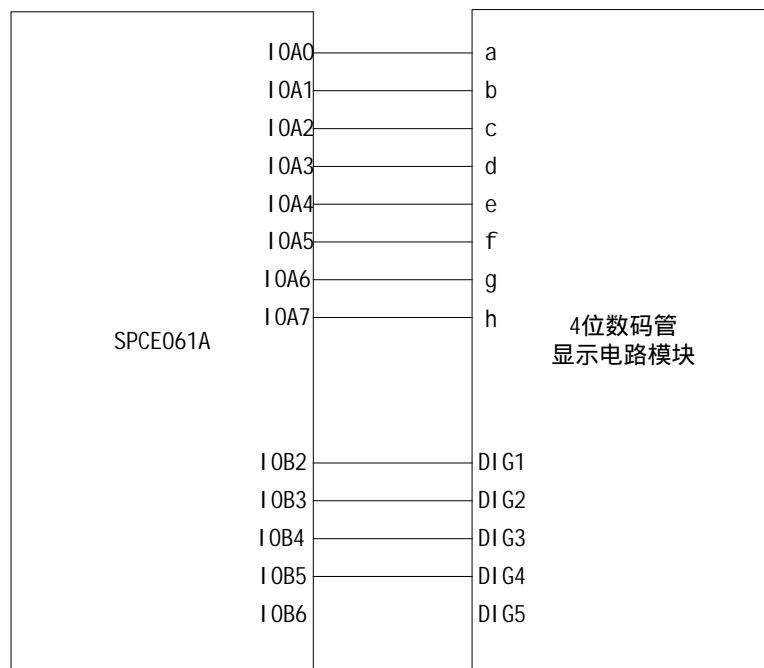


图 4-4 实验仪上 SPCE061A 和 4 位数码管显示电路模块的连接

按照上面数码管的显示原理，当在第四个数码管上显示“E”时，先通过 IOB5 端口给“DIG4”

端口送 1，选中第四个数码管；根据图 4-1，给 IOA0、IOA3、IOA4、IOA5、IOA6 端口各送一个 1，点亮 a、d、e、f、g 段就可以显示出“E”。

3. 动态显示原理

动态显示是数码管显示比较常用的方式，可以很好的解决端口资源紧张问题。如图 4-5 所示。

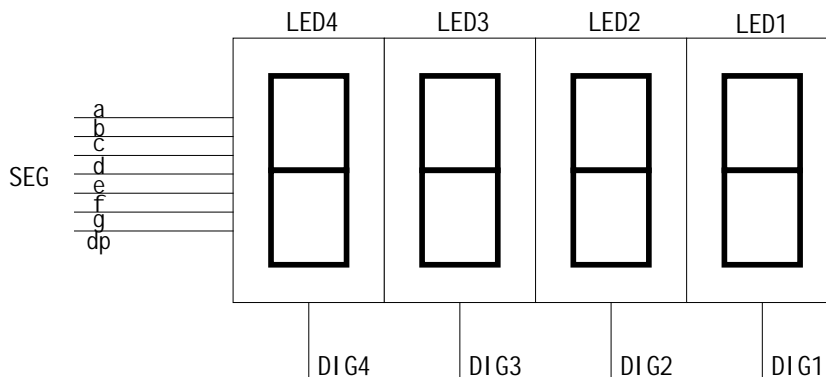


图 4-5 4 位*7 段数码管动态显示原理图

动态显示的过程：以显示“1234”为例说明，首先仅使能位信号 DIG4，然后发送“1”的段码“0x06”至数码管，LED4 显示“1”，其余的数码管都是不显示；延时一定时间之后仅使能位信号 DIG3，再发送“2”的段码“0x5b”至数码管，LED3 上显示“2”；延时之后使能位信号 DIG2，再发送“3”的段码，LED2 就会显示“3”；延时之后使能位信号 DIG1，再发送“4”的段码，LED1 就会显示“4”；如此重复循环点亮数码管。由于相邻两次（第一次点亮 LED4 和第二次点亮 LED4）的时间间隔很短（ $t < 10\text{ms}$ ），看起来仿佛 4 位数码管整体上一一直在显示“1234”。

动态显示的延时很重要，延时太短，数码管发光时间过短，数码管的亮度不够；延时太长，回扫间隔过大（超过 11ms），肉眼就会感觉到闪烁。该例程采用 4KHz 中断作为时间基准执行动态扫描，每响应一次中断，显示自动移位，回扫时间 $t = 0.25\text{ms} * 6 = 1.5\text{ms}$ 。

通过对 I/O 口的控制，定时 1s 的时间和 0.5s 的时间都使用 2Hz 的时基信号（IRQ5）。按照基础实验中 IRQ5 中断的工作原理，响应一次 IRQ5_2Hz 中断需要 0.5s 的时间，而响应两次中断的时间刚好是 1s。按照实验要求，当开 IRQ5_2Hz 中断时，6 位数码管全部显示“0”后，响应两次 IRQ5_2Hz 中断，再从第一个数码管开始显示；而刷新时间刚好是响应一次 IRQ5_2Hz 中断的时间。

【硬件连接】

硬件连接图如图 4-4，IOA0~IOA7 分别接数码管的 7 个段信号 a~g 和小数点信号 h，IOB2~IOB6 连接数码管的位信号 DIG1~DIG4 的公共端，即把实验仪上 LED_SEG 和 LED_DIG 的引针用跳线全部短接。

【程序流程图】

1. 主程序流程图

主程序流程图如图 4-6。先进行系统初始化；开 2Hz 中断；进入数码管循环显示程序：判断位信号变量是否为 0（位信号变量由读者自己定义，这个变量在中断里会改变），如果为 0，4 位数码管显示“0000”；如果不为零，按照位信号和段码显示数据进行显示。

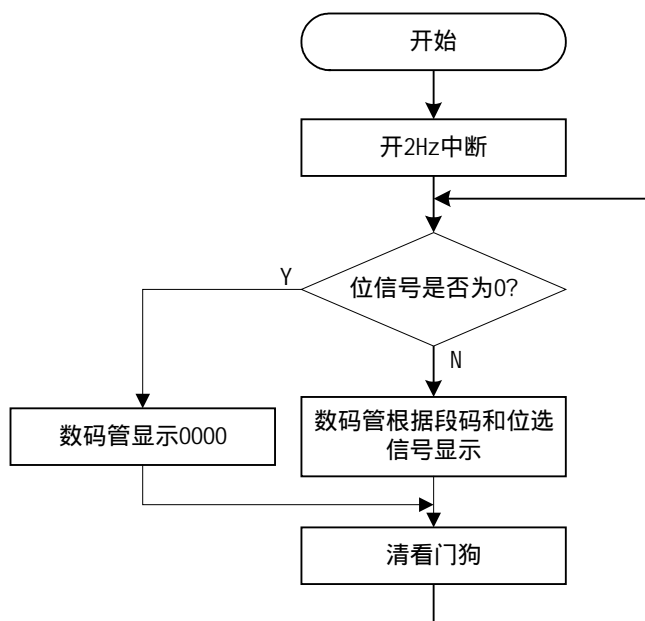


图 4-6 主程序流程图

2. 中断服务程序流程图

由于显示数据的刷新时间为 1s 和 0.5s，因此考虑采用 2Hz (0.5s) 时基中断。并且在中断内部完成段码数据更新。具体的流程如图 4-7 示：

在中断服务程序中会用到三个重要的变量：中断计数变量、显示数据变量、位信号变量。其中中断计数变量用于记录第几次进入中断（前 3 次有效），显示数据变量的内容为当前被刷新显示数据的数值（0~9），位信号变量用于标识被刷新数据所在位。结合主程序分析中断程序：首先显示“0000”时需要延时 1s，使用 2Hz 中断产生 1s 延时，需要连续响应两次中断，第二次响应中断延时满 1s，设置显示数据变量和位信号变量，让第一位数码管显示 1，其余各位仍为“0”，即显示“1000”；判断显示数据变量是否大于 9，如果小于，显示数据变量加 1；如果大于，判断位信号变量是否已经指向第四位数码管，否则位信号变量设置为指向下一位数码管，显示数据变量设置为 0；如果已经指向第四位数且显示数据变量为 9，变量赋零，清中断标志，中断返回。

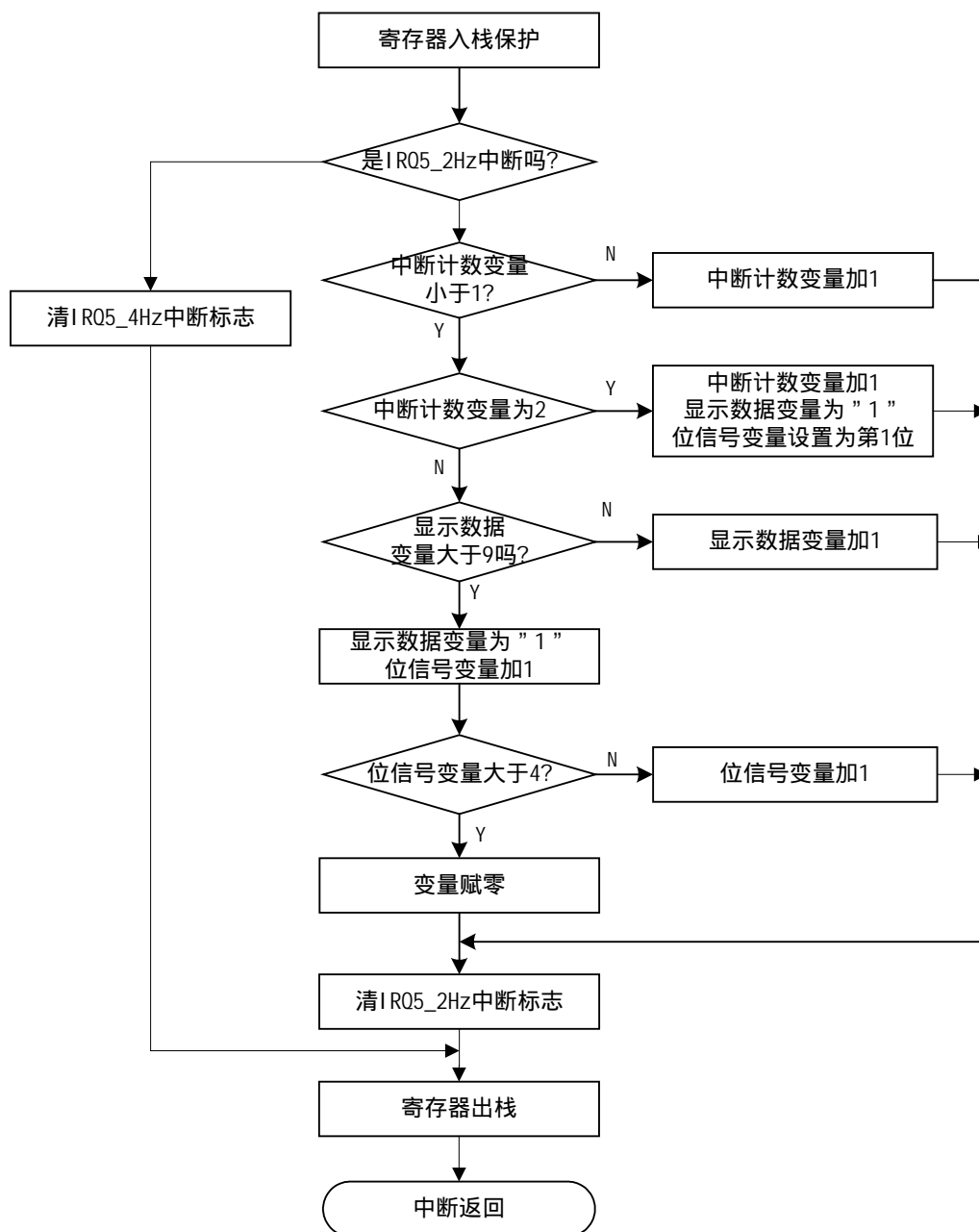


图 4-7 中断服务程序流程图

【实验步骤】

1. 新建一个工程 ex1_Led_Show ,新建一个 C 语言文件 main.c ,新建一个汇编语言文件 isr.asm。
(根据读者需要可以自己新建文件)
2. 拷贝头文件 SPCE061A.inc 和 SPCE061A.h 到新建工程 ex1_Led_Show ,这两个文件在实验仪
指导书范例代码\include 路径下可以找到。
3. 添加 SPCE061A.h、SPCE061A.inc 到工程的“Head Files”；添加后可以直接用这两个头文件
中声明的变量或者地址单元。
4. 按照程序流程图编写程序。
5. Rebuild All。
6. 按照硬件连接图连接电路 ,注意断开 LCD 中 IOB2 和 /CS 的连接 ,以免影响第一个数码管的

输出显示数据。

7. 下载程序到实验仪，根据数码管显示的现象，判断是否和实验要求相统一。

【范例路径】

在以下路径有可供大家参考的程序。

- ◆ 实验仪指导书范例代码\example\IntExa\ex1_Led_Show。
- ◆ 凌阳科技大学计划网站（www.unsp.com.cn）下载专区可以下载到相关资料。

【练习】

编写程序实现数码管显示时间，要求显示“时”和“分”，即按照 16.04 的格式显示。时间计数用中断去实现(选择什么中断读者可自行选择)，刚开机时显示时间是 00.00，接着计数显示。

实验二 1×8 键盘输入在 LED 数码管上的显示

【实验目的】

1. 熟悉 SPCE061A 单片机控制数码管显示的方法。
2. 熟悉 1×8 键盘的使用方法。
3. 掌握 1×8 键盘控制数码管显示的方法。

【实验设备】

1. 装有 Windows 系统和 $\mu'nSP^TM$ IDE 集成开发环境的 PC 机一台，SPCE061A 实验仪一套。
2. 本实验用到的实验仪硬件模块为：CPU 区电路模块，供电电路模块，下载模式选择电路模块，I/O 口引出接口模块，4 位数码管电路模块，1×8 键盘电路。

【实验要求】

1. 编程要求：主程序利用 C 语言编写。
2. 实现功能：给 1×8 键盘的每个键定义一个数字，当键按下时，将数码管上原有的显示内容左移一位，然后将按键代表的数字显示在最右边的数码管上。1×8 键盘各个键对应的数字见表 4-2。

表 4-2 键盘功能表

1	2	3	4
5	6	7	8

3. 实验现象：开机后（程序运行后），数码管全部显示 0，当键按下时，数码管上显示的数字左移一位，该键对应的数字显示到最右边的数码管上。

【实验原理】

1×8 键盘的工作原理在基础应用实验中进行了详细的说明，数码管的显示原理及 SPCE061A 控制数码管显示在上一实验中也已详细说明，不再赘述！

本实验中，键码与表 4-2 定义的功能键的对应关系如表 4-3。根据实验要求，当取得键码为下表中数字键对应的键码值时，就让数码管显示这些数字键所代表的数字。

表 4-3 实验要求定义的功能键与键码的对应关系

键码 (IOA0~IOA7)	功能键
0x80	1
0x40	2
0x20	3
0x10	4
0x08	5
0x04	6
0x02	7
0x01	8

【硬件连接图】

硬件连接图如图 4-8，IOA7~IOA0 连接 8 个按键的 COL1~COL8，IOA8 连接键盘的 ROW，即用跳线把 KEYPAD 的左右两排引针用跳线短接；IOB15~IOB8 连接 8 个 LED 的 a~h，即用排线分别连接 IOBHIG 与 LED_SEG，注意这里的连接顺序是 IOBHIG 的 IOB15 脚连接 LED_SEG 的 a，IOBHIG 的 IOB8 脚连接 LED_SEG 的 h；IOB2~IOB5 连接 LED_DIG 的 DIG1~DIG4，即用跳线把 LED_DIG 上面的四排引针用跳线短接。

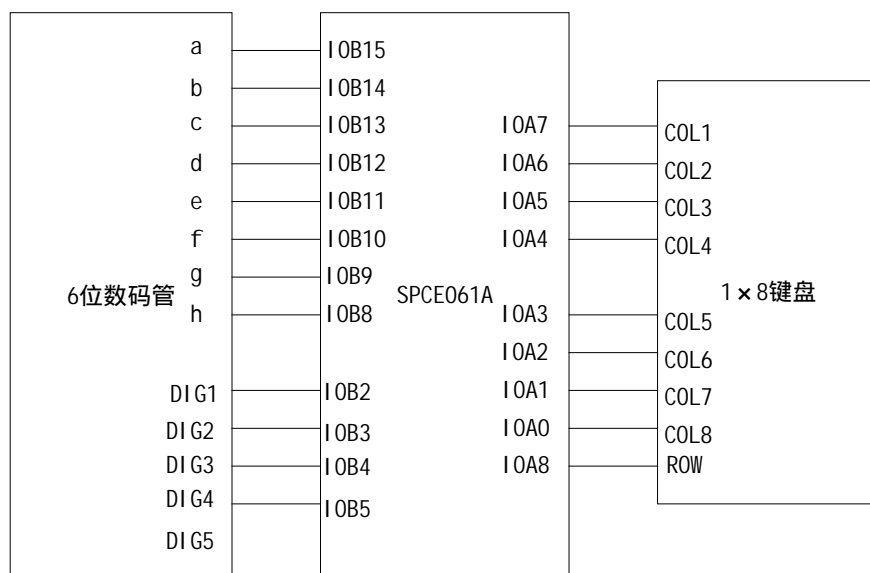


图 4-8 硬件连接图

【程序流程图】

1. 主程序流程图

主程序流程图如图 4-9，初始化系统，进入扫键循环，调用键盘程序取键值，根据键值进入相应的程序：原来显示的数字左移一位，再把数字键代表的数字显示在最右边的数码管上。

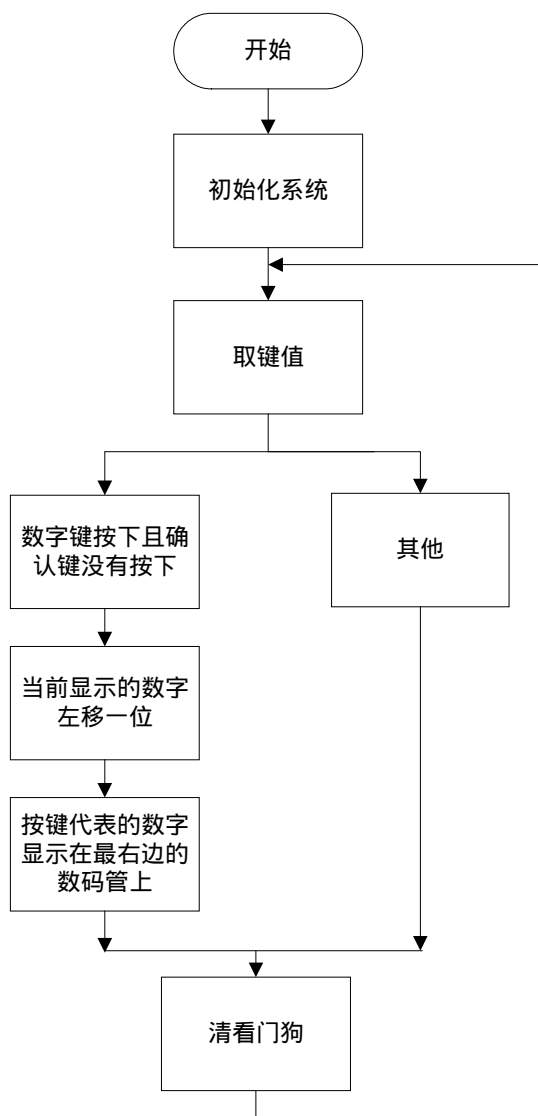


图 4-9 主程序流程图

2. 键值程序流程图

和以前实验使用 1×8 键盘程序一样，键盘程序是利用延时的方法进行消抖处理的：先读取 IOA 口键值保存在寄存器，判断读回值是否为零，为零则表示没有键按下，返回零；非零则延时大约 30ms 后再次读取 IOA 口键值，比较两次读取的键值是否相同，如果相同则返回键值，否则返回零。键盘程序流程图如图 4-10。

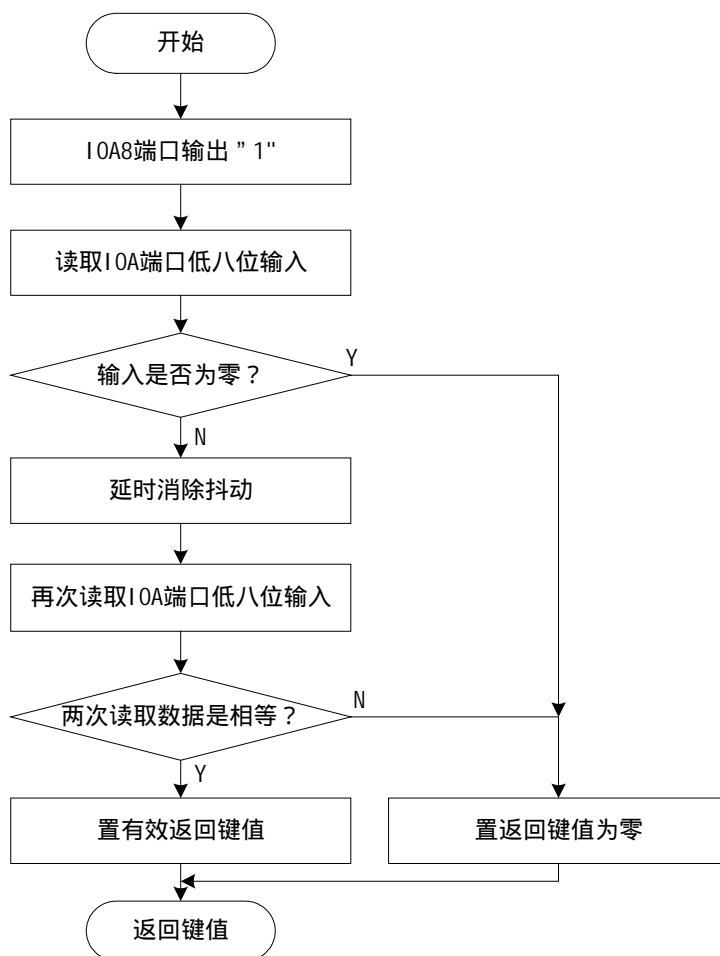


图 4-10 键操作程序流程

【实验步骤】

1. 新建工程 ex2_Led_Show_Key，在工程里 C 语言文件 main.c，新键文件 key.asm 或者 key.c（根据读者的编程习惯自己决定用 C 语言还是汇编语言编写。以后的实验中都只说明新建 main 文件，其他文件读者根据需要自己建立）。
2. 添加头文件 SPCE061A.h、SPCE061A.inc 到 ex2_Led_Show_Key 工程。
3. 根据程序流程图编写程序。
4. Rebuild All。
5. 按照硬件连接图连接硬件，注意把 LCD 接口中 IOB2 与 /CS 连接的跳线断开，以免影响数码管的显示内容。
6. 下载程序到实验仪，运行。
7. 按任意数字键“1~8”，观察数码管显示的内容，分析是否和实验要求的数字键功能相同，数码管的显示是否正确。

【范例路径】

在以下路径有可供大家参考的程序。

- ◆ 实验仪指导书范例代码\example\IntExa\ ex2_Led_Show_Key。
- ◆ 凌阳科技大学计划网站（www.unsp.com.cn）下载专区可以下载到相关资料。

【练习】

修改实验中程序，要求把数码管显示换成利用 SPLC501 液晶显示模块显示，要求在 SPLC501 液晶显示器上显示如

图 4-11 的图片，当有键按下时，这个键对应的功能字符及方格反色显示。比如，当按下 K2 时，显示的图片如图 4-12。

1	2	3	4
5	6	7	8

图 4-11 开始显示图片

1	2	3	4
5	6	7	8

图 4-12 当 S2 按下时显示图片

实验三 1×8 键盘播放语音

【实验目的】

1. 熟悉 1×8 键盘的使用方法。
2. 熟悉语音播放过程和方法。

【实验设备】

1. 装有 Windows 系统和 $\mu'nSP^TM$ IDE 集成开发环境的 PC 机一台，SPCE061A 实验仪一套。
2. 本实验用到的实验仪硬件模块为：CPU 区电路模块，供电电路模块，下载模式选择电路模块，I/O 口引出接口模块，4 位数码管电路模块，1×8 键盘电路，音频输出电路模块。

【实验要求】

1. 编程要求：主程序利用 C 语言编写；键盘程序可以利用前面几个实验编写的键盘程序。
2. 实现功能：按不同键播放不同的语音，并且在第一个数码管上显示键值，注意这里要求利用自动播放的方式播放各段语音。
3. 实验现象：按不同的键时会听到与实验仪连接的扬声器播放不同的语音，并能看到数码管显示键值。具体的键和语音的对应关系如表 4-4。

表 4-4 键和语音的对应关系

按键	语音资源名称	语音资源内容	数码管显示内容
K1 (数字键“1”)	S1.48k	凌阳专业于研发制造以及较高速与高速加价型集成电路产品	1
K2 (数字键“2”)	S2.48k	凌阳用全客户委托设计模式，提供客户满意的产品与服务	2
K3 (数字键“3”)	S3.48k	凌阳专向于多媒体语音信号处理核心技术，迎接多媒体时代的来临	3
K4 (数字键“4”)	S4.48k	凌阳以创新、专业保持产品的领导地位	4
K5 (数字键“5”)	S5.48k	凌阳以完整产品线满足客户需求	5
K6 (数字键“6”)	S6.48k	凌阳科技以人为本，诚信第一	6
K7 (数字键“7”)	S7.48k	凌阳科技走入您的生活	7
K8 (数字键“8”)	S8.48k	凌阳科技为专业集成电路设计公司	8

【实验原理】

以 SACM_S480 格式播放语音在第二章中已经有详细说明，1×8 键盘的工作原理基础应用实验中也已详细介绍，这里只简单介绍本实验的原理。

如图 4-13，把 IOA7~IOA0 连接键盘接口 (KEYPAD) 的 COL1~COL8。SPCE061A 单片机进行键盘扫描，确定哪个键按下；确定之后，计算键值，例如当 K4 按下时，使返回的键值为“4”。取得键值后，利用自动播放的方式播放相应索引号的语音资源。仍然以 K4 按下为例，当 SPCE061A 单片机判断取得的键值为“4”时，播放第四段语音。

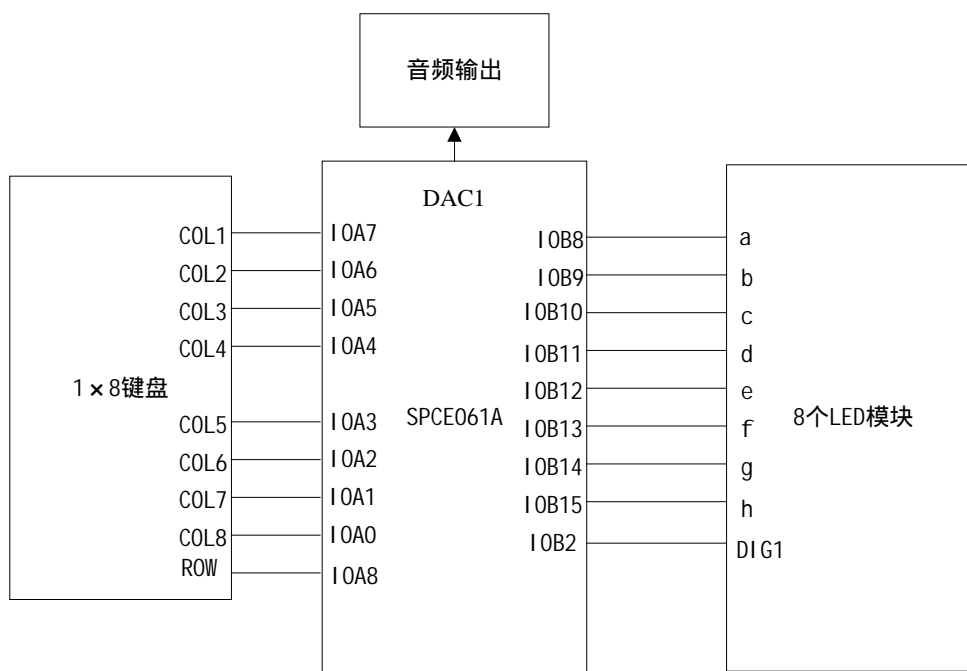


图 4-13 SPCE061A 与数码管、1×8 键盘及语音输出通道的连接框图

【硬件连接】

如图 4-13，IOA7~IOA0 连接键盘接口 (KEYPAD) 的 COL1~COL8，IOA8 连接键盘接口 (KEYPAD) 的 ROW，即用跳线直接把 KEYPAD 的左右两排引脚短接；IOB8~IOB15 连接数码管的 a~h，即用排线将 LED_SEG 和 IOBHIG 连接起来，注意 a 接 IOB8，h 连接 IOB15，IOB2 连接 LED_DIG 的 DIG1，即用跳线把 LED_DIG 的第一排引针用跳线短接。选择 DAC1 音频输出通道，即把扬声器接在上面一个扬声器接口，用跳线把“DAC1”和“音频”引针短接。

【程序流程图】

主程序流程图如图 4-14，在主程序中，先要初始化键盘并开中断，因为要播放语音，要对语音播放进行初始化，这里初始化为自动播放的方式；调用键盘程序取键值，这里可以用前几个实验的键盘程序；用第一个数码管显示取得的键值；根据键值播放实验要求相应的语音语句；执行清看门狗操作，返回继续扫描键盘。

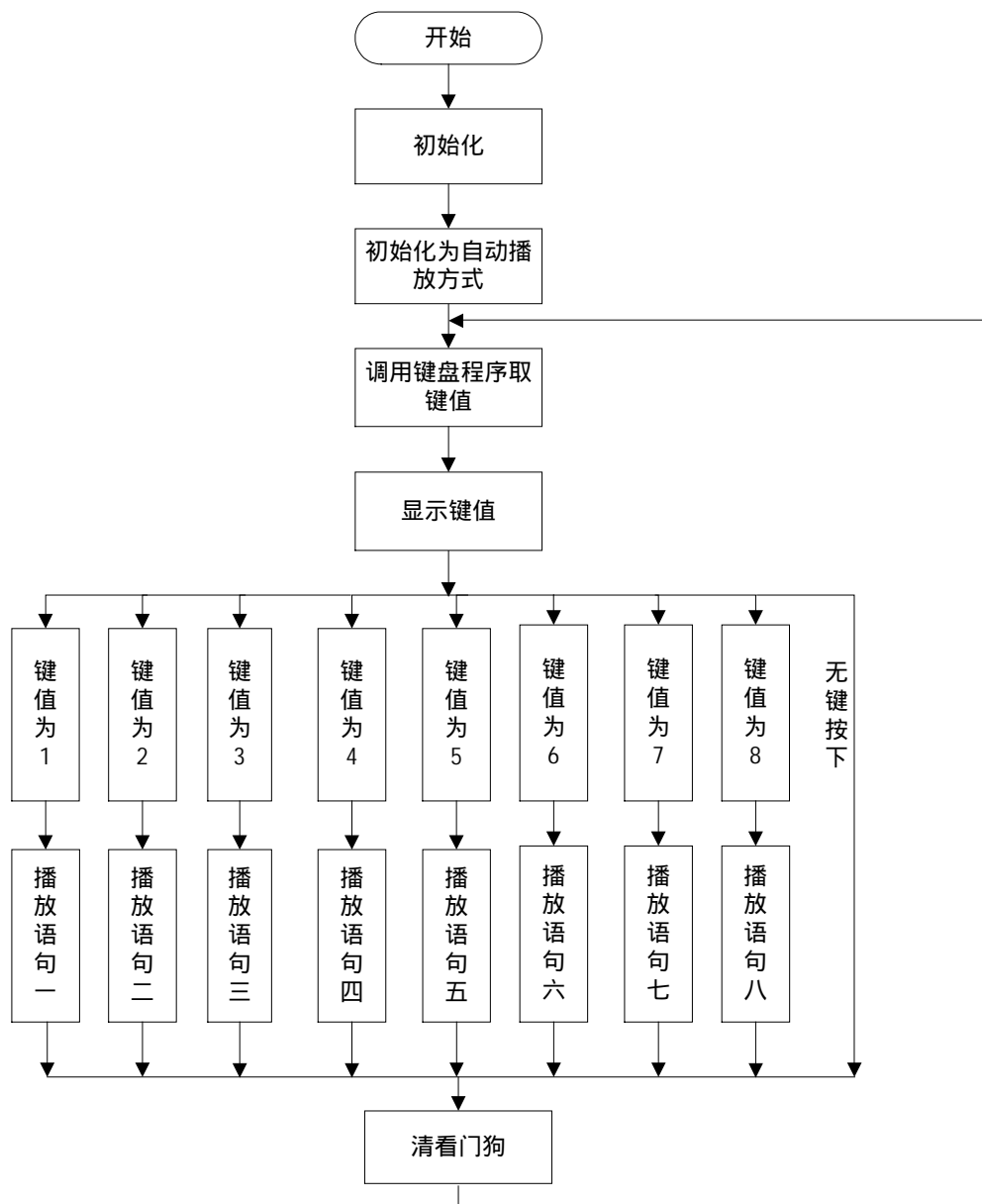


图 4-14 主程序流程图

【实验步骤】

1. 新建一个工程 `ex3_LED_Key_Music`，在工程里新建 C 语言文件 `main.c`。
2. 拷贝并添加语音库支持文件：`hardware.asm`、`hardware.h` 以及 `hardware.inc` 到 `ex3_LED_Key_Music` 工程。
3. 拷贝语音播放库 `sacmv26e.lib` 和头文件 `s480.h`、`s480.inc` 到新建的 `ex3_LED_Key_Music` 工程文件夹。（`sacmv26e.lib` 语音库在实验仪指导书范例代码\library 路径可以找到）
4. 如果在以前的实验中已经编写好键盘程序、数码管显示程序添加到 `ex3_LED_Key_Music` 工程，如果没有，读者可以自己定义文件，编写程序。
5. 在 IDE 集成开发环境的 `Project`→`Setting`→`link` 中链接 `sacmv26e.lib` 库到新建的工程中。链接库的方法见第二章实验一。
6. 添加 `hardware.asm` 到新建的工程的“Source Files”，添加 `s480.h`、`s480.inc`、`hardware.h`、



- hardware.inc 四个头文件到 “ Head Files ”。
7. 添加语音资源 S1.48k~S8.48k 到工程。S1.48k~S8.48k 语音资源在实验仪指导书范例代码\example\IntExa\ex3_LED_Key_Music\voice 路径下可以找到。
 8. 按照程序流程图在 main.c 文件中编写程序；如果没有编写键盘程序，自行按照 1×8 键盘的工作原理，编写键盘程序。
 9. Rebuild All。
 10. 按照硬件连接图连接电路，注意把 LCD 的 IOB2 和/CS 连接的跳线拔掉。
 11. 下载程序，运行。
 12. 按任意键，观察第一个数码管，根据与实验仪连接的扬声器播放的声音，分析是否和实验要求相符。

【范例路径】

在以下路径有可供大家参考的程序。

- ◆ 实验仪指导书范例代码\example\IntExa\ex3_LED_Key_Music。
- ◆ 凌阳科技大学计划网站（www.unsp.com.cn）下载专区可以下载到相关资料。

【练习题】

自己录制 8 段语音，这 8 段语音分别是 “ 1 ”、“ 2 ”、“ 3 ”、“ 4 ”、“ 5 ”、“ 6 ”、“ 7 ”、“ 8 ”；按照第二章中的实验步骤压缩成 SACM_S480 格式，第一次按键时数字键（同实验二的键盘定义）数字显示在第一个数码管上，同时语音播放对应的数字。

实验四 语音识别&LCD 显示

【实验目的】

1. 了解语音识别的原理。
2. 熟悉语音识别的 API 函数。
3. 熟悉 LCD 显示模块显示字符及图象的方法。
4. 掌握语音识别的使用方法。

【实验设备】

1. 装有 Windows 系统和 $\mu'nSP^TM$ IDE 集成开发环境的 PC 机一台, SPCE061A 实验仪一套。
2. 本实验用到的实验仪硬件模块为: CPU 区电路模块, 供电电路模块, 下载模式选择电路模块, LCD 显示电路模块, 音频 MIC 输入、音频输出电路。

【实验要求】

1. 编程要求: 主程序利用 C 语言编写。
2. 实现功能: 实现语音识别功能, 液晶显示器上显示不同图片, 以示响应不同的语音指令。
3. 实验现象: 开机后(运行后)LCD 显示器显示图片 2; 按照语音提示训练, 训练过程如下(详细的训练过程会在实验步骤中详细说明)。

- ◆ 提示音“1”, 训练名字, 读者可以自己取任意名字, 训练两遍。
- ◆ 提示音“2”, 训练语音“ How are you ”, 训练两遍;
- ◆ 提示音“3”, 训练语音“ happy new year ”, 训练两遍;
- ◆ 提示音“4”, 训练语音“ I love you ”, 训练两遍;
- ◆ 提示音“5”, 训练语音“ go to sleep ”, 训练两遍。

训练成功以后, 提示开始识别(提示音为 begin.24k), 进入识别过程(详细的识别过程在实验步骤中会详细说明):

- ◆ 识别名称, 显示图片 1(实验中的所有图在实验仪指导书范例代码\example\IntExa\ex4_Recognise_LCD 中都可以找到), 播放应答音“ lalalalala ”。
- ◆ 识别命令“ How are you ”, 显示图片 2, 播放应答音“ I'm fine ”。
- ◆ 识别命令“ happy new year ”, 显示图片 4, 播放应答音“ happy new year ”。
- ◆ 识别命令“ I love you ”, 显示图片 3, 播放应答音“ I love you too ”。
- ◆ 识别命令“ go to sleep ”, 显示图片 5 和图片 0 组成的动画图形播放应答音打呼噜声。

【实验原理】

1. 语音识别 API 函数

和语音播放类似，凌阳科技公司也提供了语音识别库 bsrV222SDL.lib，语音识别库支持语音的识别过程。在 bsrV222SDL.lib 库中提供了语音识别整个过程（包括训练和识别两个过程）的 API 函数：

```
int BSR_DeleteSDGroup(0)

int BSR_Train (int CommandID, int TraindMode)

void BSR_InitRecognizer(int AudioSource)

int BSR_GetResult(Void)

void BSR_StopRecognizer(void)

_BSR_FIQ_Routine
```

在上面的函数中，第一个函数为初始化函数，第二个函数为训练函数，第三到五个函数为识别函数，最后一个函数是中断调用的函数。下面分别详细说明这些函数。

1) 【API 格式】C：int BSR_DeleteSDGroup(0)

【功能说明】SRAM 初始化。

【参 数】该参数是辨识的一个标识符，0 代表选择 SRAM,并初始化。

【返 回 值】当 SRAM 擦除成功返回 0，否则返回 -1。

【备 注】语音命令的特征模型是通过 BSR_Train 函数(下面会介绍)保存在 RAM 空间中的。如果所需的 RAM 空间已被旧的特征模型数据占满，新特征模型则无法保存到 RAM 中。利用 BSR_DeleteSDGroup 函数可以把 RAM 空间中所有的特征模型删除，释放出所需空间。

2) 【API 格式】C：int BSR_Train (int CommandID, int TraindMode)

【功能说明】训练函数。

【参 数】CommandID：命令序号，范围从 0x100 到 0x105,并且对于每组训练语句都是唯一的。

TraindMode：训练次数，要求使用者在应用之前训练一或两遍：

BSR_TRAIN_ONCE 表示要求训练一次。

BSR_TRAIN_TWICE 表示要求训练两次。

【返 回 值】训练成功，返回 0；

没有声音，返回-1；

需要更多的语音数据来训练，返回-2；

当环境太吵时，返回-3；

当数据库满，返回-4；

当两次输入命令不同，返回-5；

当序号超出范围，返回-6。

【备 注】

在调用训练程序之前，确保识别器正确的初始化；

训练次数是 2 时，则两次一定会有差异，所以一定要保证两次训练结果接近；

为了增强可靠性，最好训练两次，否则辨识的命令就会倾向于噪音；

调用函数后，等待 1~2 秒开始训练，每条命令只有 1.3 秒，也就是说，当训练命令超出 1.3 秒时，只有前 1.3 秒命令有效。

3) 【API 格式】C : void BSR_InitRecognizer(int AudioSource)

【功能说明】辨识器初始化。

【参 数】该参数定义语音输入来源，通过 MIC 语音输入还是 LINE_IN 电压模拟量输入。

【返 回 值】无。

4) 【API 格式】C : int BSR_GetResult(Void)

【功能说明】辨识中获取数据。

【参 数】无。

【返 回 值】当无命令识别出来时，返回 0；

识别器停止、未初始化或识别未激活返回 - 1；

当识别不合格时返回 - 2；

当识别出来时返回命令的序号。

【备 注】该函数用于启动辨识。

5) 【API 格式】C : void BSR_StopRecognizer(void)

【功能说明】停止辨识。

【参 数】无。

【返 回 值】无。

【备 注】该函数是用于停止识别，当调用此函数时，FIQ_TMA 中断将关闭。

6) 【API 格式】ASM : _BSR_FIQ_Routine

【功能说明】在中断中调用。

【参 数】无。

【返 回 值】无。

【备 注】

该函数在中断 FIQ_TMA 中调用；

当主程序调用 BSR_InitRecognizer(int AudioSource)时，辨识器便打开 8K 采样率的 FIQ_TMA 中断并开始将采样的语音数据填入辨识器的数据队列中；

应用程序需要设置一下程序段在 FIQ_TMA 中。

2. 特定人语音识别 (SD)

特定人语音识别即语音样板由单个人训练，也只能识别训练人的语音命令，对他人的命令识别率较低或几乎不能识别。特定人语音识别由“训练”和“识别”两个步骤组成，只有进行语音训练后才会生成特征模型，根据这些特征模型才能够识别，如图 4-15。

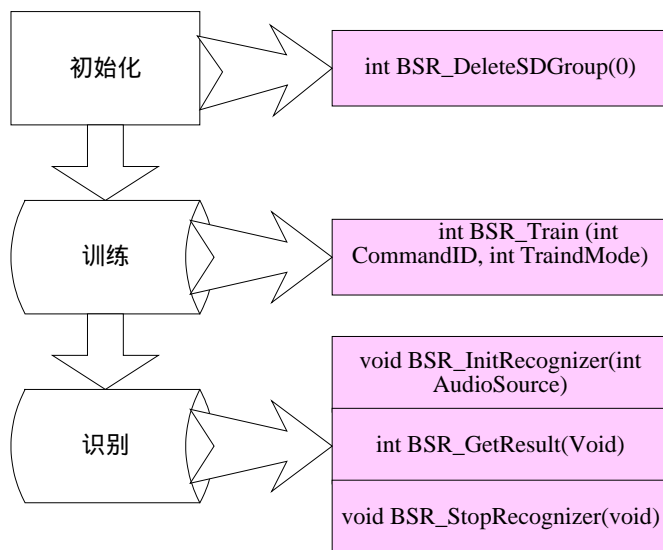


图 4-15 语音识别过程

需要说明的是，训练过程中，每条语音命令的长度不要超过 1.3 秒。训练后得到的语音特征模型保存在 RAM 中，每条命令占用 96 Word，由于 RAM 空间的限制，同时可识别的语音命令数量最大为 5 条。如果需要识别更多语音命令，可以采用命令分组的方法。

本实验中要求利用 SPLC501 液晶显示器显示整个识别过程，由 BSR_GetResult(Void)读回的值判断要进行哪些操作，从而控制液晶显示器显示哪一幅图片。由于和前面显示的图片留下的视觉效应，就可以看到是一个很听话的图片显示在屏幕上。

【硬件连接】

如图 4-16 硬件连接图，分别用跳线连接 IOB3 与 AO，IOB4 与 R/W，IOB5 与 EP，IOB2 与 /CS，IOA0~IOA7 与 DB0~DB7；即用跳线把实验仪 LCD 的所有引针全部短接。选择 DAC1 音频输出通道，即把扬声器接在上面一个扬声器接口，用跳线把“DAC1”和“音频”引针短接。

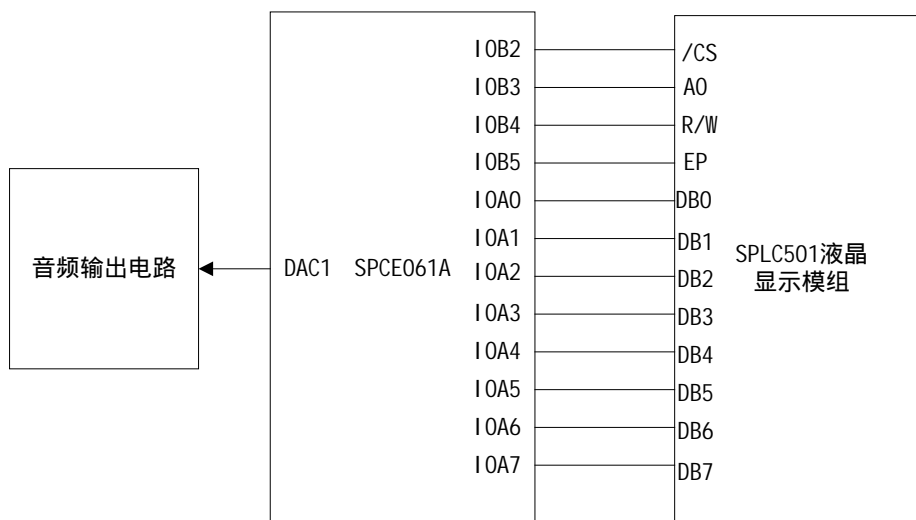


图 4-16 硬件连接图

【程序流程图】

主程序流程图如图 4-17，按照实验要求，初始化 LCD 后，设置 LCD 图象显示模式为取反叠加模式，显示图片 2；调用训练函数训练；训练成功后，播放识别提示音（语音资源为 begin.24k）；初始化识别器，显示图象 0；进入识别循环，取识别结果，如果识别出名字，显示图片 1，播放语音“lalalalala”（语音资源为 C7.24k）；如果识别出第一条命令，显示图片 2，播放语音“I'm fine”（语音资源为 C2.24k）；如果识别出第二条命令，显示图片 4，播放语音“Happy new year”（语音资源为 C9.24k）；如果识别出第三条命令，显示图片 3，播放语音“I love you too”（语音资源为 C3.24k）；如果识别出第四条命令，交替显示图片 5 和图片 0，显示 3 次，播放语音打呼噜声（语音资源为 ZZZ.24k）；判断是否连续有 600 次没有识别出任何命令，如果是，显示图片 0，执行清看门狗操作；如果不是，直接执行清看门狗操作，返回取识别结果。

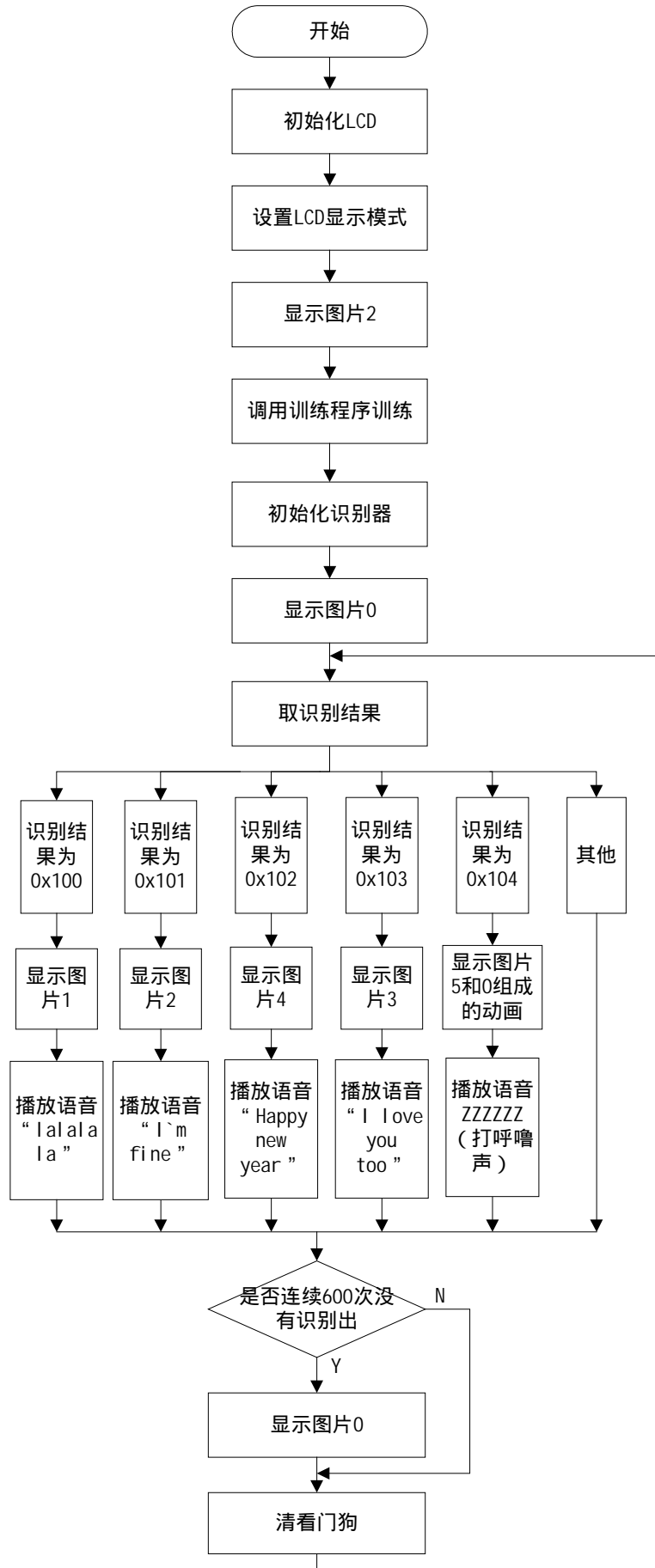


图 4-17 主程序流程图

训练程序流程图如图 4-18，训练命令时如 A) 图，播放训练提示音；取训练结果；当两次都训练成功时，返回训练结果(成功)；如果还需要训练一次时，播放提示音“HOO”(语音资源为 hoo.24k 或者 hoo.16k)，返回循环继续训练；如果没有训练成功：包括没有检测到任何声音、环境太吵、数据库满、两次检测出声音不同和序号错误的情况，播放语音“OHOH”(语音资源为 OHOH.24k 或者 OHOH.16k)，返回训练结果(没有训练成功)。注意这里返回到训练结果程序中，即右边的程序流程图中，如 B) 图。

B) 图为训练结果的程序，无论训练哪条语句，如果没有训练成功就一直循环训练，直到成功；如果训练成功，进入下一步。

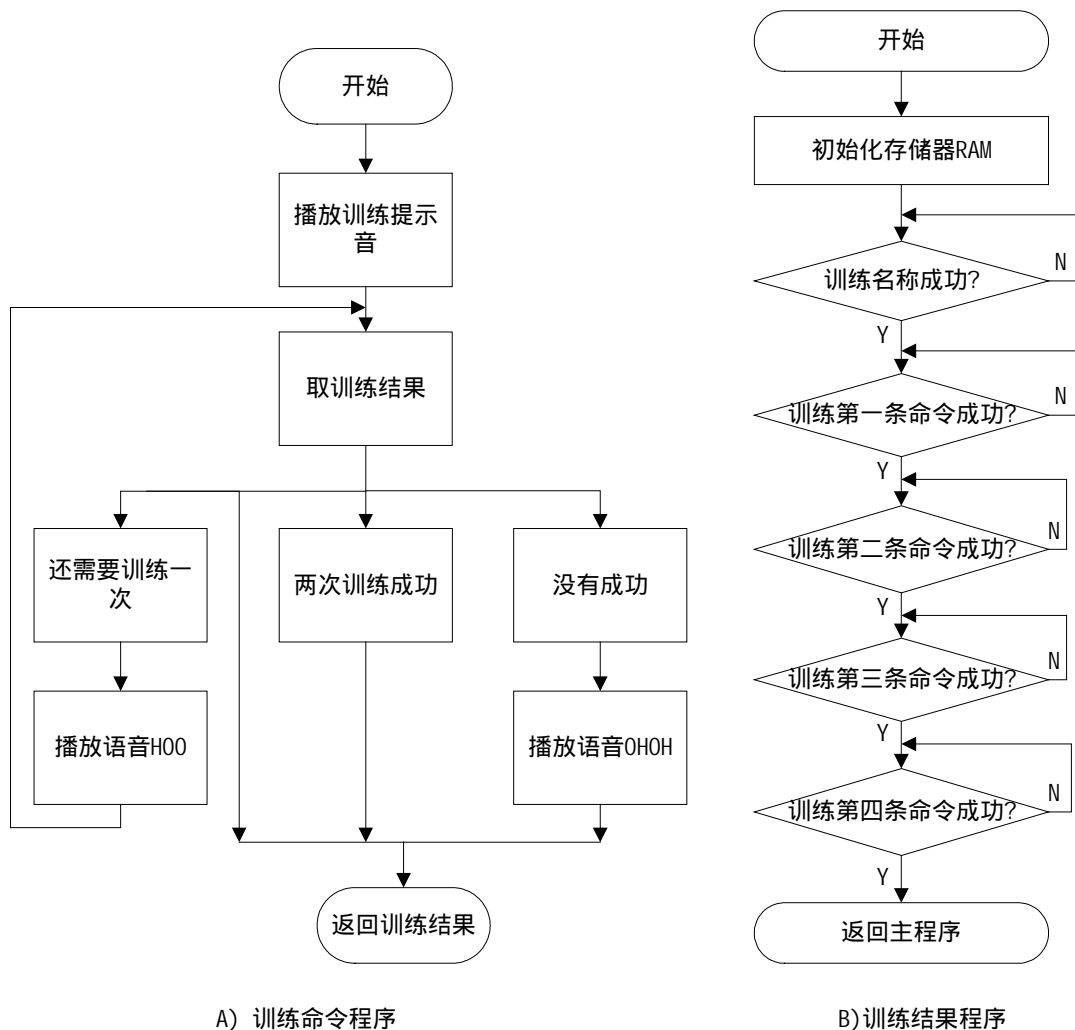


图 4-18 训练程序流程图

【实验步骤】

1. 新建一个工程 ex4_Recognise_LCD，在工程里建立一个新的 C 语言文件。
2. 程序中会用到语音识别的 API 函数，先拷贝语音识别库 bsrV222SDL.lib 到 ex4_Recognise_LCD 工程文件夹，bsrV222SDL.lib 在实验仪指导书范例代码 \example\IntExa\ex4_Recognise_LCD 路径可以找到。
3. 拷贝语音识别头文件 bsrSD.h、bsrSD.inc 到 ex4_Recognise_LCD 工程文件夹。
4. 拷贝语音播放库 sacmv26e.lib 和头文件 A2000.h、a2000.inc 到 ex4_Recognise_LCD 工程文件



夹。

5. 拷贝支持 sacmv26e.lib 的汇编语言文件 hardware.asm 和头文件 hardware.h、hardware.inc 到 ex4_Recognise_LCD 工程文件夹。
6. 拷贝 SPLC501 液晶显示 DataOSforLCD.asm、SPLC501Driver_IO.asm、SPLC501Driver_IO.inc、SPLC501User.c 和 SPLC501User.h 五个驱动程序（提供在实验仪指导书范例代码\example\model_Exa\Driver\SPLC501driver）到 ex4_Recognise_LCD 工程文件夹。
7. 在 IDE 集成开发环境的 Project→Setting→link 中链接 bsr222SDL.lib 和 sacmv26e.lib 库到工程中。链接库的方法见第二章实验一。
8. 添加 hardware.asm、DataOSforLCD.asm、SPLC501Driver_IO.asm 和 SPLC501User.c 到工程文件的“Source Files”，添加 bsrSD.h、bsrSD.inc、A2000.h、a2000.inc、hardware.h、hardware.inc、SPLC501Driver_IO.inc、SPLC501User.h 八个头文件到“Head Files”。
8. 拷贝语音资源，并加载到工程当中；语音资源可以从实验仪指导书范例代码\example\IntExa\ex4_Recognise_LCD\voice 路径下可以找到。
9. 提取 BMP 图片的字模数据，并将数据加载到工程中，提取方法在第三章当中已有介绍，这里不再赘述；图片的资源可以在实验仪指导书范例代码\example\IntExa\ex4_Recognise_LCD\Pic 路径下可以找到。
10. 按照程序流程图编写程序。
11. Rebuild All。
12. 按照硬件连接图连接硬件。
13. 下载程序，运行。
14. 训练：按照如下顺序训练：

提示音“1” 输入名称 提示音“HOO” 输入名称 提示音“2” 输入“ How are you ”
提示音“HOO” 输入“ How are you ” 提示音“3” 输入“ happy new year ” 提示音
“HOO” 输入“ happy new year ” 提示音“4” 输入“ I love you ” 提示音“HOO”
输入“ I love you ” 提示音“5” 输入“ go to sleep ” 提示音“HOO” 输入“ go to sleep ”。

在训练中要注意以下几点：

- a) 两次训练命令要尽量相同；
 - b) 在播放提示音后，等待 1~2 秒开始训练；这点在实验原理中介绍 BSR_Train 函数时有详细说明；
 - c) 当播放提示音“OHOH”时表示训练失败，继续训练正在训练的命令。
15. 识别：识别过程如图 4-19 流程图，训练成功后识别任意一条命令。

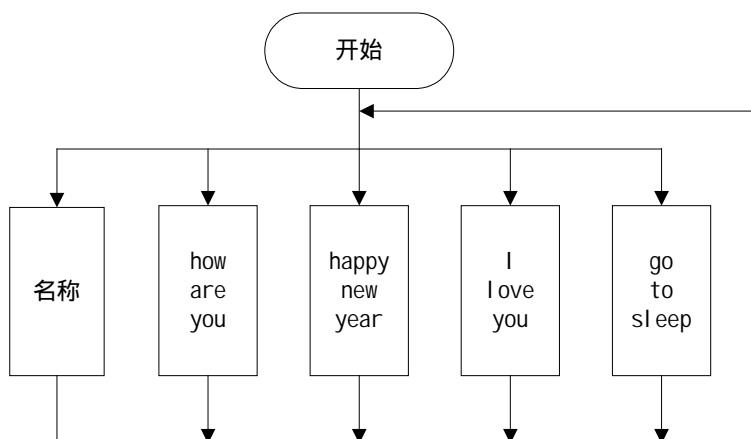


图 4-19 识别流程

识别过程中现象如下：

- a) 识别“名称时”听到与实验仪连接的扬声器播放语音“lalalalala”，LCD 显示模块上显示一个吐舌头的图片。
- b) 识别“ How are you ”时，听到与实验仪连接的扬声器播放语音“ I m fine ”，LCD 显示模块上显示一个笑脸的图片。
- c) 识别“ happy new year ”时，听到与实验仪连接的扬声器播放语音“ happy new year ”，LCD 显示模块上显示一个大笑的图片。
- d) 识别“ I love you ”时，听到与实验仪连接的扬声器播放语音“ I love you too ”，LCD 显示模块上显示一个眨一只眼睛的图片。
- e) 识别“ go to sleep ”时，听到与实验仪连接的扬声器播报打呼噜的语音，LCD 显示模块上显示一个大哭和睡觉交替显示的动画图片。

16. 根据实验现象，分析是不是和实验要求相统一。

【程序范例】

在以下路径有可供大家参考的程序。

- ◆ 实验仪指导书范例代码\example\IntExa\ex4_Recognise_LCD。
- ◆ 凌阳科技大学计划网站（www.unsp.com.cn）下载专区可以下载到相关资料。

实验五 带有背景音乐的动力图片

【实验目的】

1. 巩固 SPLC501 液晶显示模块的使用方法。
2. 掌握显示动态图片并伴有音乐背景的方法。

【实验设备】

1. 装有 Windows 系统和 $\mu'nSP^TM$ IDE 集成开发环境的 PC 机一台, SPCE061A 实验仪一套。
2. 本实验用到的实验仪硬件模块为: CPU 区电路模块, 供电电路模块, 下载模式选择电路模块, LCD 显示电路模块, 音频输出电路。

【实验要求】

1. 编程要求: 主程序利用 C 语言编写。
2. 实现功能: 图片显示配合背景音乐播放。由于实验中要播放音乐, 对音质的要求比较高, 所以本实验要求利用 SACM_A2000 格式播放音乐。
3. 实验现象: 开机后显示图象 ON_01 (实验中的图象都可以在实验仪指导书范例代码 \example\IntExa\ex5_LCD_DynamicGraphicMusic\pic 路径找到), 接着按 ON_01 ON_02 ON_03 ON_04 ON_05 ON_06 ON_07 ON_08 ON_010 的顺序循环显示动态图片, 同时播放背景音乐 (背景音乐在实验仪指导书范例代码 \example\IntExa\ex5_LCD_DynamicGraphicMusic\voice 路径可以找到)。

【实验原理】

SPLC501 液晶显示模组的工作原理在第三章的前五个实验中已经介绍, SACM_A2000 格式语音播放的原理在第二章实验一和实验二也已经介绍, 这里简单介绍一下本实验的原理。

在第二章实验一的自动播放中, 语音播放的主循环当中调用 SACM_A2000_ServiceLoop() 函数, 完成获取语音资源并填充语音队列、进行语音解码的操作; 但是在本实验中, 主循环中要先显示一帧或者几帧图象, 而显示程序相对来说耗时较长; 为了保证语音资源的及时获取, 同时得到及时的解码, 在设计当中利用 IRQ4 的 1KHz 中断调用 SACM_A2000_ServiceLoop() 函数, 主循环当中只进行刷新显示即可。

【硬件连接】

如本章实验四图 4-16 硬件连接图, 分别用跳线连接 IOB3 与 AO, IOB4 与 RWP, IOB5 与 EP, IOB2 与 CS, IOA0~IOA7 与 DB0~DB7; 即用跳线把实验仪 LCD 的所有引针全部短接。选择 DAC1 音频输出通道, 即把扬声器接在上面一个扬声器接口, 用跳线把 “DAC1” 和 “音频” 引针短接。

【程序流程图】

主程序流程图如图 4-20，初始化操作包括初始化 LCD 和初始化语音播放，语音播放初始化为自动播放方式；开 IRQ4_1KHz 中断，在 1KHz 中断当中完成语音的服务程序（SACM_A2000_ServiceLoop()）调用，以及进行计数累加（定义有一个全局的变量，用于中断计时），响应 300 次 IRQ4_1KHz 中断的时间刚好是 LCD 动态图片的显示刷新时间 0.3s；显示第一帧图象，也就是 ON_01 图象。调用 SACM_A2000_Play 函数播放语音；进入显示动态图片加背景音乐循环，按照实验要求顺序显示图片，刷新时间为 0.3s，判断是否播放结束，如果播放结束，则停止播放后，调用 SACM_A2000_Play 函数循环播放，如果没有播放结束，请看门狗。IRQ4_1KHz 中断服务程序流程图如图 4-21。

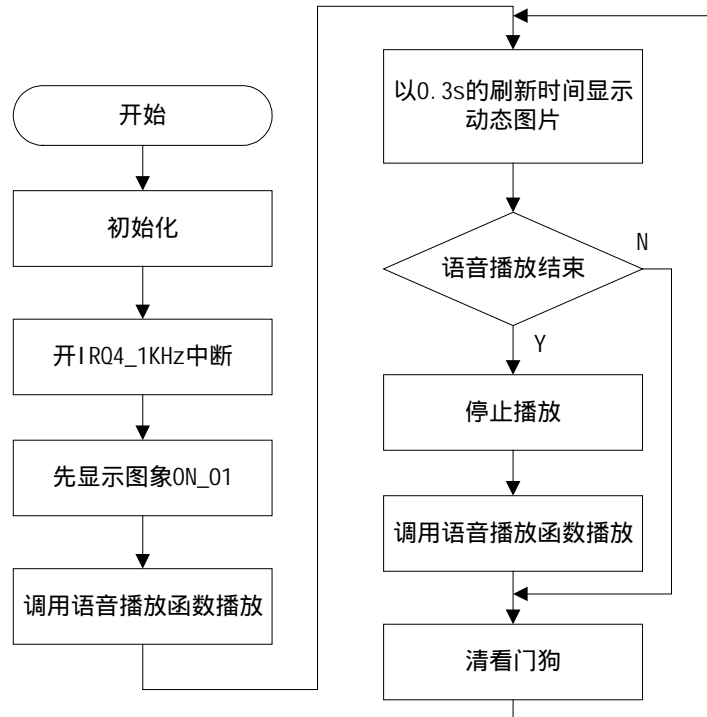


图 4-20 主程序流程图

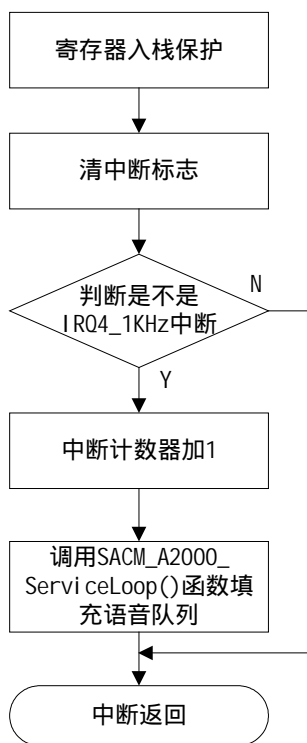


图 4-21 IRQ4_1KHz 中断

【实验步骤】

1. 新建一个工程 ex5_LCD_DynamicGraphicMusic ,在新工程里建立一个 C 语言文件 main.c。(按照读者自己的编程习惯或者编程要求,可以根据实际情况自行定义更多的文件。)
2. 在实验仪指导书范例代码\example\IntExa\ex5_LCD_DynamicGraphicMusic\pic 路径下找到实验要求的 9 幅图片。
3. 提取这 9 幅图片的字模数据,一次提取它们的全部图片,不需要一幅幅提取;字模文件保存在新建的 ex5_LCD_DynamicGraphicMusic 工程文件夹当中,命名为 PicData。
4. 拷贝语音播放库 sacmv26e.lib 和头文件 a2000.h、a2000.inc 到新建的 ex5_LCD_DynamicGraphicMusic 工程文件夹。
5. 拷贝支持 sacmv26e.lib 的汇编语言文件 hardware.asm 和头文件 hardware.h、hardware.inc 到新建的 ex5_LCD_DynamicGraphicMusic 工程文件夹。
6. 拷贝 SPLC501 液晶显示 DataOSforLCD.asm、SPLC501Driver_IO.asm、SPLC501Driver_IO.inc、SPLC501User.c 和 SPLC501User.h 五个驱动程序文件(提供在实验仪指导书范例代码\example\model_Exa\Driver\SPLC501driver)到新建的 ex5_LCD_DynamicGraphic Music 工程文件夹。
7. 在 IDE 集成开发环境的 Project→Setting→link 中链接 sacmv26e.lib 库到新建的工程中。链接库的方法见第二章实验一。
8. 添加 PicData.c、hardware.asm、DataOSforLCD.asm、SPLC501Driver_IO.asm 和 SPLC501User.c 到新建的工程的“Source Files”,添加 PicData.h、a2000.h、a2000.inc、hardware.h、hardware.inc、SPLC501Driver_IO.inc、SPLC501User.h 六个头文件到“Head Files”。
9. 按照程序流程图编写程序。

10. Rebuild All。
11. 按照硬件连接图连接硬件。注意把 KEYPAD 的所有跳线断开，以免对 I/O 口数据造成影响。
12. 下载程序，运行。
13. 根据实验现象，分析是不是和实验要求相统一。

【程序范例】

在以下路径有可供大家参考的程序。

- ◆ 实验仪指导书范例代码\example\IntExa\ex5_LCD_DynamicGraphicMusic。
- ◆ 凌阳科技大学计划网站（www.unsp.com.cn）下载专区可以下载到相关资料。

实验六 UART 控制液晶显示

【实验目的】

1. 巩固 SIO 的基本使用方法。
2. 巩固 UART 的基本使用方法。
3. 巩固 LCD 的基本使用方法。

【实验设备】

1. 装有 Windows 系统和 $\mu'nSP^TM$ IDE 集成开发环境的 PC 机一台, SPCE061A 实验仪一套, 9 针标准串口线一根; 串口调试工具, 如“串口调试助手 V2.1.exe”。
2. 本实验用到的实验仪硬件模块为: CPU 区电路模块, 供电电路模块, 下载模式选择电路模块, LCD 显示电路模块, UART/USB 通讯电路, SPR4096 存储电路模块。

【实验要求】

1. 编程要求: 主程序利用 C 语言编写。
2. 实现功能: SPLC501 液晶显示模组显示 UART 通讯的过程, 包括“准备”、“接收中”和“接收完成”; SPCE061A 接收到从 PC 机应用程序发送的数据后, SPR4096A FLASH 存储器存储通讯数据, SPLC501 液晶显示模组显示接收到的数据; UART 通讯速率设置为 9600bps。
3. 实验现象: 运行程序后, SPLC501 液晶显示模组显示“UART Ready”; 设置 PC 端串口调试工具后, 发送一串数据, SPLC501 液晶显示模组显示“Receiving...”; 当数据发送完成并已保存到 SPR4096A 存储器, SPLC501 液晶显示模组显示“Complete”; 延时后显示“The Data is:”, 并显示从 PC 机应用程序发送的数据。

【实验原理】

前面的实验已分别讲述过 SPR4096A FLASH 存储器、UART 和 LCD 单独使用的原理, 本实验是三者结合在一起工作时的情况, 下面简要介绍一下它们结合在一起的工作原理。

刚开始还没有发送数据, 所以只需要调用 SPLC501 液晶显示模组驱动程序显示英文字符串“UART Ready”; 当接收到从 PC 机应用软件发送的第一个数据时, 调用 SPLC501 液晶显示模组驱动程序显示英文字符串“Receiving...”; 延时等待接收完毕, 如果已经接收完成, 调用 SPLC501 液晶显示模组驱动程序显示英文字符串“Complete”; 之后把接收到的数据显示在 SPLC501 液晶显示模组上, 这样, 就完成一次 UART 通讯, 也可以说 SPLC501 液晶显示模组显示了通讯过程。

【硬件连接】

如硬件连接图, 分别用跳线连接 IOB3 与 AO, IOB4 与 RWP, IOB5 与 EP, IOB2 接 CS, IOA0~IOA7 与 DB0~DB7; 即用跳线把实验仪 LCD 的所有引针全部短接; IOB0 连接 SPR4096A 的 SCK, IOB1 连接 SPR4096A 的 SDA, 即用跳线短接 4096PORT 的左右两排引针; SPR4096A 的 CF7 连接 DGND, 即用跳线把 CF7 引针和 DGND 引针短接; 用 9 针标准串口线连接 PC 机和实验仪 UART 接口, 用两

个跳线把实验仪的 COMM 端口中靠左边的两排引针短接。

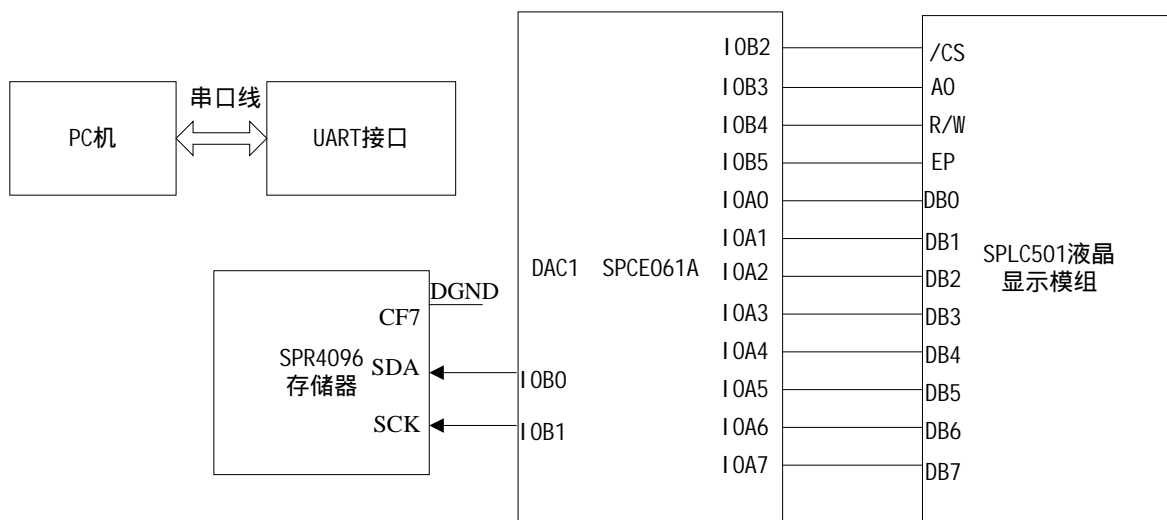


图 4-22 硬件连接图

【程序流程图】

1. 主程序流程图

主程序流程图如图 4-23，先进行初始化操作，包括初始化 SIO、初始化 UART 和初始化 LCD；开 IRQ 中断；擦除整个 FLASH；把写入地址指向 0x5000 地址单元（即从第 10 扇区开始存储）；显示“UART Ready”到（10，15）位置；等待接收直到接收到第一个数据；清屏显示“Receiving...”到（10，15）位置；延时等待接收完成，如果没有接收完数据，继续接收，如果已经接收完成，清屏显示“Complete”到（10，15）位置；延时的目的是能在 LCD 上看清楚数据发送已经完成；清屏显示“The Data is:”到（5，5）位置；从 SPR4096 FLASH 一个字节一个字节读出数据，清屏后从（5，20）开始显示读出的数据：当显示到最右边时，即 X 已经达到了 128（事实上不能达到 128），则换行显示，如果一页不能显示完全，则清屏换页显示；进入主程序循环，执行清看门狗操作。

在主程序的循环当中，有一个等待接收完毕的过程，可以利用接收超时的概念进行设计；在等待接收完成的循环等待中，利用一个全局变量（定义为超时计数器）进行不断的累加，并判断是累加值是否大于设置的超时值，当检测到大于时，则表示数据已接收完成，进入显示接收字符数据的处理。在 UART 的接收中断服务程序当中，还有一步对前面所介绍的超时计数器进行清零的操作，也就是说，只要有连续的串行数据接收到来，就会不断对该变量清零，则在主循环当中判断接收完毕的检测会认为数据正在接收当中。只要在一定时间内没有串行数据的接收发生，超时计数器就会在循环当中累加到超时的数值，从而表明接收完毕。

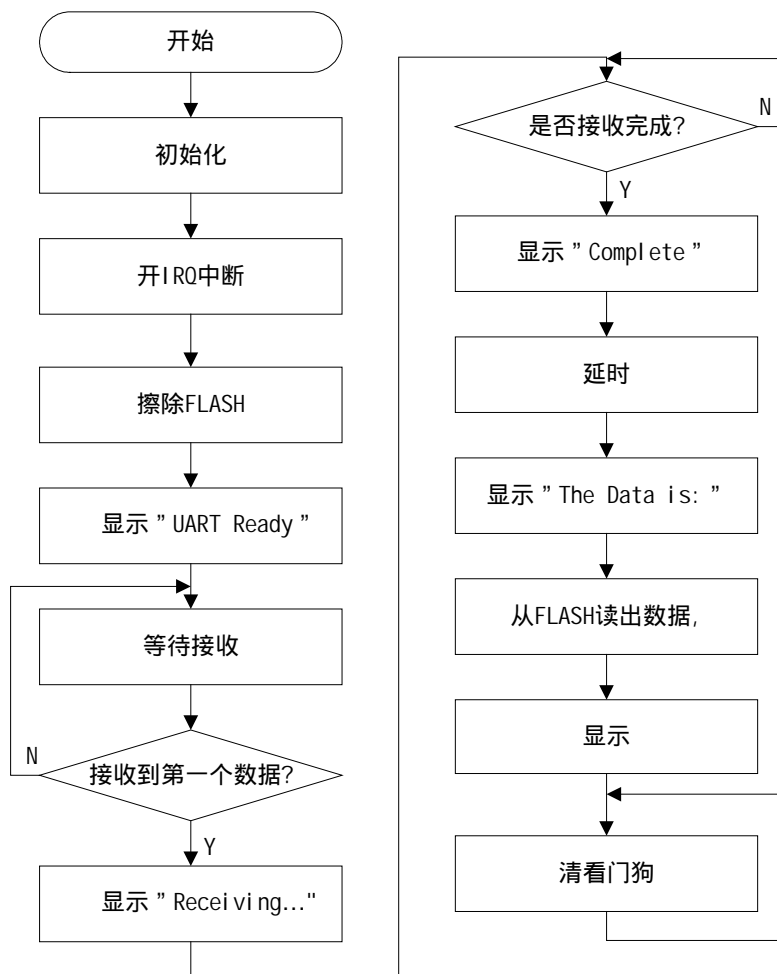


图 4-23 主程序流程图

2. UART 接收中断服务程序：

中断服务程序流程图如图 4-24，因为每响应一次中断，说明接收到一个字节数据，向 FLASH 写一个字节的的数据，改变 FLASH 的地址指针；当收到数据时，给收到数据标志置 1，返回主程序，这时候当接收到第一个数据时，在主程序就返回一个已经接收到第一个数据的标志。

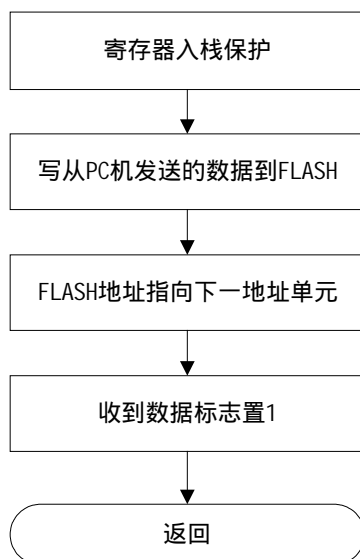



图 4-24 UART IRQ (IRQ7) 中断服务程序流程图

【实验步骤】

1. 新建一个工程 ex6_UART&LCD&SPR4096，在新工程里新建 C 语言文件 main.c。（根据需要建立其他的文件。）
2. 拷贝 SPCE061A 硬件寄存器定义头文件 SPCE061A.h、SPCE061A.inc 到 ex6_UART&LCD&SPR4096 工程文件夹。
3. 拷贝 SPLC501 液晶显示 DataOSforLCD.asm、SPLC501Driver_IO.asm、SPLC501Driver_IO.inc、SPLC501User.c 和 SPLC501User.h 五个驱动程序到 ex6_UART&LCD&SPR4096 工程文件夹。
4. 拷贝 SPR4096 的驱动程序 4096.asm（提供在实验仪指导书范例代码\example\model_Example\Driver\SPR4096Adriver）到 ex6_UART&LCD&SPR4096 工程文件夹。
5. 添加 DataOSforLCD.asm、SPLC501Driver_IO.asm 和 SPLC501User.c、4096.asm 到工程文件的“Source Files”，添加 SPCE061A.h、SPCE061A.inc、SPLC501Driver_IO.inc、SPLC501User.h 四个头文件到“Head Files”。
6. 按照程序流程图编写程序。
7. Rebuild All。
8. 按照硬件连接图连接硬件。注意把 KEYPAD 的所有引脚全部断开，以免对 I/O 口的数据造成影响。另外注意用跳线把 COMM 左边的两排引针短接。
9. 下载程序，全速运行。
 - A) 当 LCD 显示“UART Ready”时，把串口工具的波特率设置为 9600，无检验位，8 位数据位，1 位结束位。
 - B) 从 PC 端串口工具发送数据（可以是一个字节也可以是多个字节），观察 LCD 显示内容，当收到第一个数据时应该显示“Receiving...”。
 - C) LCD 显示“Complete”时，通过存储器观察窗口（Memory）观察[P_UART_Data]单元（7023H）是不是已经接收到最后一个数据。
 - D) LCD 显示“The Data is :”和从 PC 端应用程序发送的数据。
 - E) 观察 LCD 显示内容，分析接收到的数据是不是正确。
10. 如果读者感觉第 9 步骤没有看清实验要求的状态，比如没有看清楚显示“Receiving...”，只是一晃而过了，可以点击  图标重新运行，加断点运行观察完整的过程。
 - A) 在显示“UART Ready”语句的后一句加断点，运行。
 - B) 在显示“Receiving...”的语句后一句加断点，运行；从 PC 端串口工具发送一个字节数据，这时候程序运行会停在显示“Receiving...”语句的后一句。
 - C) 在显示“Complete”的语句后一句加断点，运行，这时候可以看到 LCD 显示“Complete”。
 - D) 运行，如果第 9 步骤显示的数据和发送数据相同，这时候 LCD 显示的数据也应该为发送的一个字节数据。



【程序范例】

在以下路径有可供大家参考的程序。

- ◆ 实验仪指导书范例代码\example\IntExa\ex6_UART&LCD&SPR4096。
- ◆ 凌阳科技大学计划网站 (www.unsp.com.cn) 下载专区可以下载到相关资料。

实验七 0~3V 电压测量表

【实验目的】

1. 熟悉凌阳音频算法和 SACM_S480 格式的语音播放方式。
2. 熟悉实验仪 LCD 显示模块的使用方法。
3. 进一步掌握 SPCE061A 内部 ADC 的使用方法。

【实验设备】

1. 装有 Windows 系统和 $\mu'nSP^TM$ IDE 集成开发环境的 PC 机一台，SPCE061A 实验仪一套。
2. 本实验用到的实验仪硬件模块为：CPU 区电路模块，供电电路模块，下载模式选择电路模块，LCD 显示电路模块，音频输出电路，模拟电压（0~3.3V）输入电路模块。

【实验要求】

1. 编程要求：主程序利用 C 语言编写，中断服务程序利用汇编语言编写。
2. 实现功能：实现一个模拟电压表的功能，要求电压表能够测量电压值，在液晶显示器上显示并播报测量电压值。
3. 实验现象：开机（运行）后，显示一个图片 MM.bmp（它的字模数据 MM.c、MM.h 在实验仪指导书范例代码\example\IntExa\ex7_VoltageMeasureFrom0To3 提供），显示并播放当前电位器所在位置的电压；变化 R73 电位器，显示并播报 R73 电位器变化后的电压；即只要 R73 电位器改变，系统就能自动测量出 R73 电位器的电压，并把测量得到的电压值播报并显示出来。（要求显示和播报电压值的精度为小数点后第 3 位）

【实验原理】

本实验中，通过 SPCE061A 内部 ADC 采集数据，如图 4-25。当滑动头变化时，IOA0 口输入的电压就随着变化；当已经通过[P_ADC_Ctrl]启动转换，通过[P_ADC_MUX_Ctrl]设置从 LINE_IN1 输入时，读取[P_ADC_MUX_Data]即可得到转换数据。把这些数据换算成电压值，然后把电压值按从个位到小数点后第 3 位的顺序播报和显示。

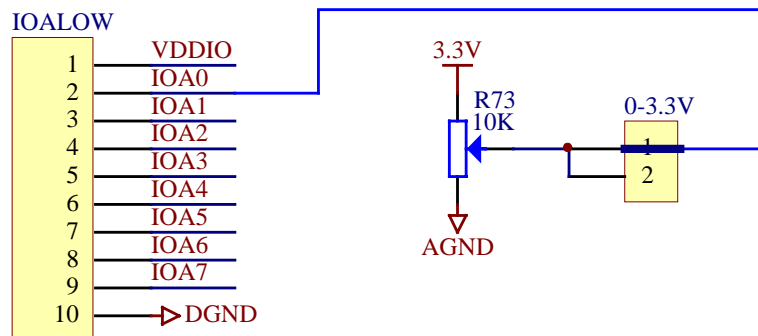


图 4-25 0~3.3V 直流电平输入电路

【硬件连接】

硬件连接图如图 4-26。分别用跳线连接 IOB3 与 AO ,IOB4 与 RWP ,IOB5 与 EP ,IOB2 与/CS , IOA8~IOA15 与 DB0~DB7。即用跳线把实验仪 LCD 的上面五排引针全部短接,用排线连接 IOAHIG 和 LCD 的数据口 DB0~DB7;选择 DAC1 音频输出通道,即把扬声器接在上面一个扬声器接口,用跳线把“DAC1”和“音频”引针短接;IOA0 连接 R73 电位器,即用排线把 IOALOW 的 IOA0 引针和 0-3.3V 接口(模拟电压输入电路模块)中任一引针连接起来。

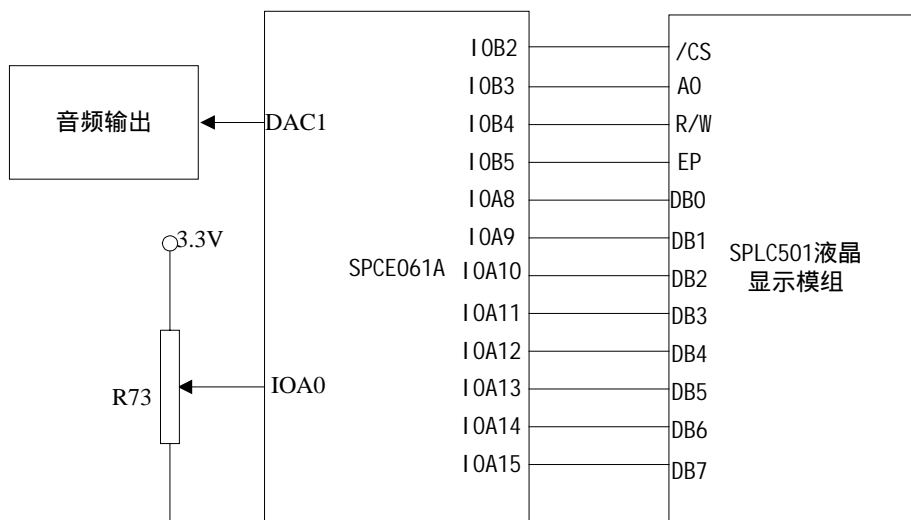


图 4-26 硬件连接图

【程序流程图】

主程序流程图如图 4-27,初始化 LCD 后,显示开机图片 MM;进入电压测量循环:启动转换,采集 20 个转换数据并取平均值,主要是为了减少误差;计算电压值;如果和上次电压值之差小于 $\pm 0.1V$,则电压没有变化,返回继续采集数据;如果和上次电压值之差大于 $\pm 0.1V$,表示有电压变化,计算电压各位(包括个位到小数点后 3 位)的数;清屏,显示图片 MM;播报并显示电压值。

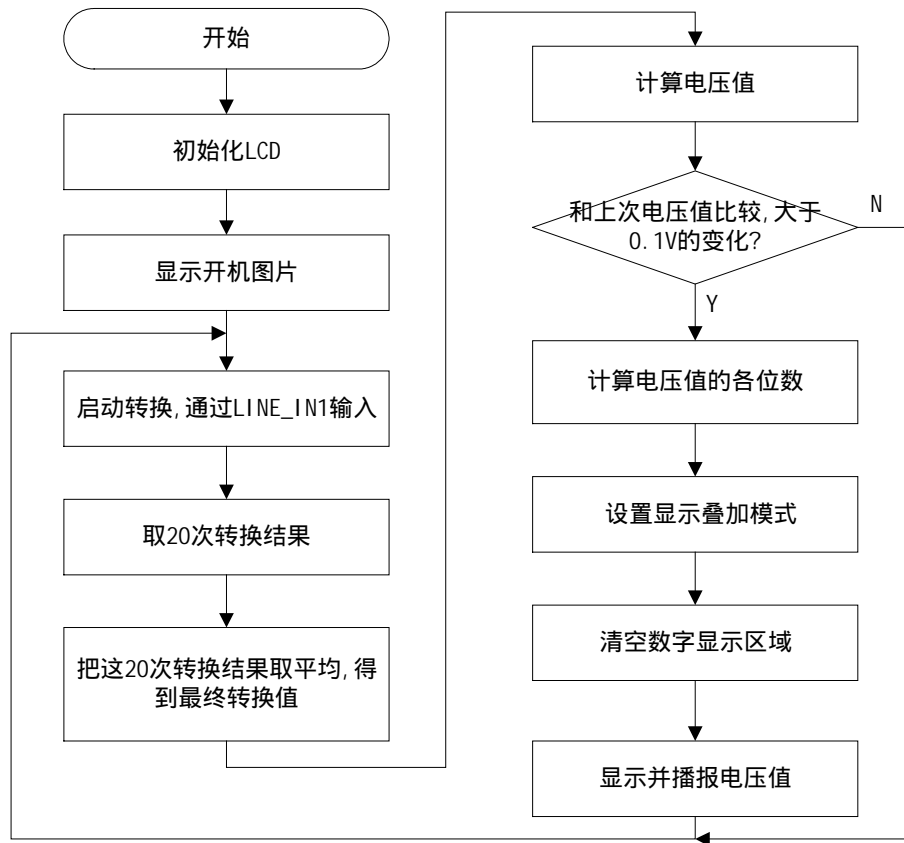


图 4-27 主程序流程图

【实验步骤】

1. 新建一个工程 ex7_VoltageMeasureFrom0To3，新建一个 C 语言程序 main.c。（根据编程需要，读者可自行新建其他文件）。
2. 拷贝支持 sacmv26e.lib 的汇编语言文件 hardware.asm 和头文件 hardware.h、hardware.inc 以及头文件 SPCE061A.h 和 SPCE061A.inc 到 ex7_VoltageMeasureFrom0To3 工程文件夹。
3. 拷贝语音播放库 sacmv26e.lib 和头文件 s480.h、s480.inc 到 ex7_VoltageMeasureFrom0To3 工程文件夹。
4. 拷贝语音资源到当前工程的文件夹中，并将语音资源加载到工程中；语音资源可在实验仪指导书范例代码\example\IntExa\ex7_VoltageMeasureFrom0To3\voice 文件夹中找到。
5. 拷贝 SPLC501 液晶显示 DataOSforLCD.asm、SPLC501Driver_IO.asm、SPLC501Driver_IO.inc、SPLC501User.c 和 SPLC501User.h 五个 LCD 驱动程序到 ex7_VoltageMeasureFrom0To3 工程文件夹。
6. 在 IDE 集成开发环境的 Project→Setting→link 中链接 sacmv26e.lib 库到工程中。
7. 添加 hardware.asm、DataOSforLCD.asm、SPLC501Driver_IO.asm 和 SPLC501User.c 到工程文件的“Source Files”，添加 s480.h、s480.inc、hardware.h、hardware.inc、SPLC501Driver_IO.inc、SPLC501User.h 六个头文件到“Head Files”。
8. 按照程序流程图编写程序。
9. Rebuild All。
10. 按照硬件连接图连接硬件。注意把 IOALOW 的 IOA0 引针和 0-3.3V 接口中任一引针连接，把



KEYPAD 的所有跳线全部拔掉，以免对 I/O 口的数据造成影响。

11. 下载程序，运行。

12. 调节 R73 电位器（模拟电压输入电路模块），观察 LCD 显示数据和与实验仪连接的扬声器播报的数据，分析是不是和实验要求相统一。

【范例路径】

在以下路径有可供大家参考的程序。

- ◆ 实验仪指导书范例代码\example\IntExa\ex7_VoltageMeasureFrom0To3。
- ◆ 凌阳科技大学计划网站（www.unsp.com.cn）下载专区可以下载到相关资料。

实验八 录音笔

【实验目的】

1. 巩固 SIO 的基本使用方法。
2. 巩固 LCD 的基本使用方法。
3. 巩固凌阳音频算法及语音播放方法。

【实验设备】

1. 装有 Windows 系统和 $\mu'nSP^TM$ IDE 集成开发环境的 PC 机一台，SPCE061A 实验仪一套”。
2. 本实验用到的实验仪硬件模块为：CPU 区电路模块，供电电路模块，下载模式选择电路模块，LCD 显示电路模块，SPR4096 存储电路模块，1×8 键盘电路，音频输出电路模块。

【实验要求】

1. 编程要求：主程序利用 C 语言编写。
2. 实现功能 实现一个录音笔功能 要求 1×8 键盘控制录放音过程 录入语音存储到 SPR4096A 存储器，LCD 显示操作过程，如图 4-28，图中“STOP”的后面显示一个空心的方框，表示录音或者放音结束。



图 4-28 液晶显示器显示现象

3. 实验现象：运行程序后 LCD 显示如图 4-28；按 K1 键录音，LCD 显示如图 4-29a) 所示；按 K2 键录音结束，LCD 显示如图 4-28；按 K3 键放音，听到第一段录的语音被播放出来，同时 LCD 显示如图 4-29b)；按 K2 键语音播放结束，LCD 显示如图 4-28；按 K4 键，听到下一段（第一次按这个键时，播放第二段录音，第二次按下这个键时，播放第三段录音）录音被播放出来，同时 LCD 显示图 4-29b)；按 K5 键删除录到的语音，在删除过程中，和上面几个图片类似，LCD 显示方框到“DELETE”字符串后面。



a)录音

b)放音

图 4-29 录音与放音过程中的 LCD 显示

【实验原理】

本实验中，利用显示字符串函数先把这些字符串全部显示在 LCD 上，利用画矩形函数画一个空心的矩形框，根据取到的键值判断在 LCD 的哪个位置显示空心的方框，在录放音的整个过程中只需要纵向移动这个空心方框就可以了。

录音时，通过 MIC 录入语音，利用 DVR 的 API 函数，把录入语音队列中的语音资源先用 SACM_DVR_FetchQueue()函数取出来，写到 SPR4096 FLASH 中去；下一次录音时，从前面一段语音资源存放结束地址的下一个地址开始存放；放音时，先找到语音资源存储的起始地址，从 SPR4096 FLASH 中读出语音数据，一个个字节的填充到语音播放队列中解码播放，一直到该段语音资源存储的结束地址。

【硬件连接】

硬件连接图如图 4-30，分别用跳线连接 IOB3 与 AO，IOB4 与 RWP，IOB5 与 EP，IOB2 接 CS，IOA0~IOA7 与 DB0~DB7，即用跳线把实验仪 LCD 的所有引针全部短接；IOB0 连接 SPR4096A 的 SCK，IOB1 连接 SPR4096A 的 SDA，即用跳线短接 4096PORT 的左右两排引针；SPR4096A 的 CF7 连接 DGND，即用跳线把 CF7 引针和 DGND 引针短接；IOB8~IOB15 连接 1×8 键盘接口 KEYPAD 的 COL1~COL8，KEYPAD 的 ROW 接 IOA8，即用排线连接 IOBHIG 和 KEYPAD，注意 IOB8 接 COL1，用跳线短接 KEYPAD 的最下面一排引针。选择 DAC1 音频输出通道，即把扬声器接在上面一个扬声器接口，用跳线把“DAC1”和“音频”引针短接。

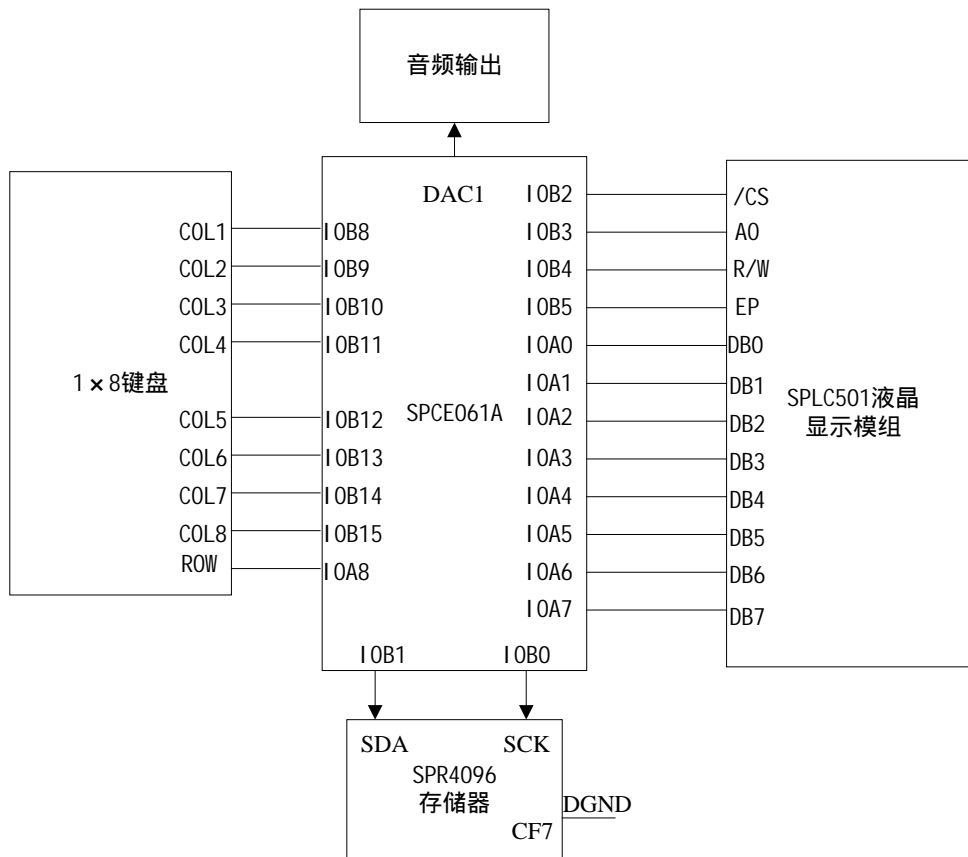


图 4-30 硬件连接图

【程序流程图】

主程序流程图如图 4-31，初始化包括 I/O 口初始化、LCD 的初始化和 SIO 初始化；擦除整个 FLASH 以存储录音数据；设置 LCD 显示模式为取反叠加模式，LCD 显示一个停止模式（分别显示“RECORD”、“PLAY”、“STOP”、“DELETE”到四行，在“STOP”后面显示一个空心的矩形方框），即显示如图 4-28 所示的图，在“STOP”后面显示方框；进入录放音循环：扫描键盘取键值，如果扫描到 K1 按下，进行录音处理；如果扫描到 K2 按下，进行停止录音或者停止放音处理；如果扫描到 K3 按下，播放第一段语音；如果扫描到 K4 按下，播放下一段录音；如果扫描到 K5 按下，擦除整个 FLASH，删除所有录音。

注：各个功能实现程序在前面的实验中都有练习，这里不再给出流程图。

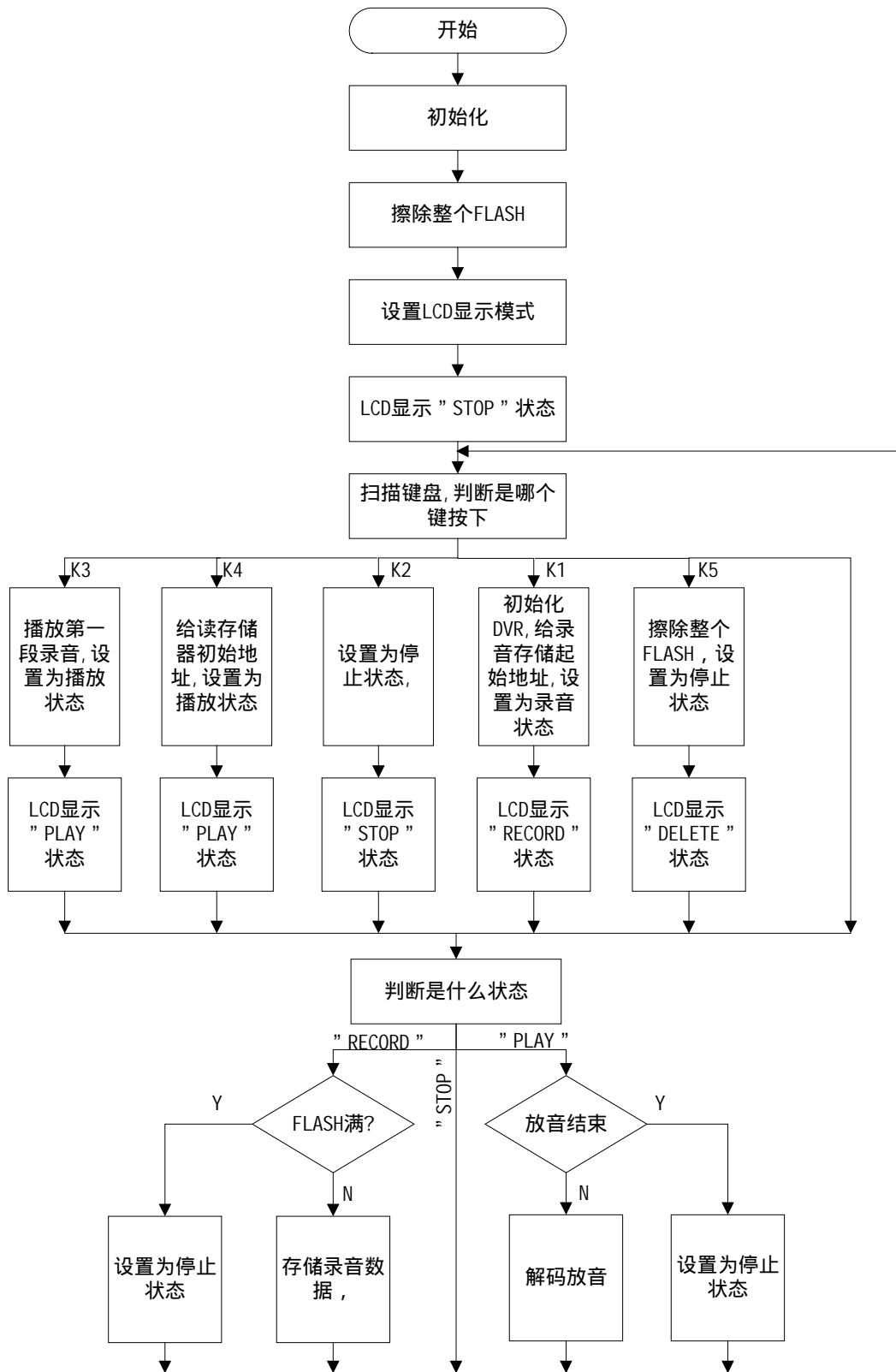


图 4-31 主程序流程图

【实验步骤】

1. 新建一个工程 ex8_Record , 在工程里新建一个 C 语言文件 main.c 和一个汇编语言文件 isr.asm。

2. 拷贝头文件 SPCE061A.h 和 SPCE061A.inc 到 ex8_Record 工程文件夹。
3. 拷贝语音播放库 sacmv26e.lib、支持文件 hardware.asm、hardware.inc、hardware.h、sram.asm 以及头文件 dvr.h、dvr.inc 到 ex8_Record 工程文件夹。
4. 拷贝 SPLC501 液晶显示 DataOSforLCD.asm、SPLC501Driver_IO.asm、SPLC501Driver_IO.inc、SPLC501User.c 和 SPLC501User.h 五个 LCD 驱动程序到 ex8_Record 工程文件夹。
5. 拷贝 SPR4096 的驱动程序 4096.asm 到 ex8_Record 工程文件夹。
6. 在 IDE 集成开发环境的 Project→Setting→link 中链接 sacmv26e.lib 库到工程中。链接库的方法见第二章实验一。
7. 添加 DataOSforLCD.asm、SPLC501Driver_IO.asm、SPLC501User.c、hardware.asm、sram.asm 和 4096.asm 到工程文件的“ Source Files ”,添加 dvr.h、dvr.inc、SPCE061A.h、SPLC501User.h、SPCE061A.inc、SPLC501Driver_IO.inc、SPLC501User.h 六个头文件到 “ Head Files ”。
8. 按照程序流程图编写程序。
9. Rebuild All。
10. 按照硬件连接图连接硬件。注意 “ CF7 ” 引针和 “ DGND ” 引针短接, “ DAC1 ” 和 “ 音频 ” 引针短接。
11. 下载程序,运行。观察 LCD 是不是显示 “ STOP ” 状态。
12. 按 “ 录音 ” (K1) 键,开始录音。LCD 显示 “ RECORD ” 状态。
13. 按 “ 停止 ” (K2) 键,停止录音。LCD 显示 “ STOP ” 状态。
14. 按 “ 放音 ” (K3) 键,开始放音。LCD 显示 “ PLAY ” 状态。
15. 按 “ 停止 ” (K2) 键,停止录音。LCD 显示 “ STOP ” 状态。
16. 按 “ 录音 ” (K1) 键,开始录音。LCD 显示 “ RECORD ” 状态。录入第二段语音。
17. 按 “ 停止 ” (K2) 键,停止录音。LCD 显示 “ STOP ” 状态。第二段语音录入结束。
18. 按 “ 放音 ” (K3) 键,开始放音。LCD 显示 “ PLAY ” 状态。这时候发现播放的仍然是第一段语音。
19. 按 “ 播放下一段 ” (K4) 键,播放第二段录入的语音。
20. 按 “ 删除 ” (K5) 键,LCD 显示 “ DELETE ” 状态。再按 “ 放音 ” (K3) 键,发现这时候的语音全部被删除。

【程序范例】

在以下路径有可供大家参考的程序。

- ◆ 实验仪指导书范例代码\example\IntExa\ex8_Record。
- ◆ 凌阳科技大学计划网站 (www.unsp.com.cn) 下载专区可以下载到相关资料。