Computer Science and

西南科技大学(SouthWest University of Science and Technology)

Experimental Instructor of Embedded System (Source Code)

嵌入式系统实验指导书（代码版）

信息教研室　编写

前言

FOREWORD

FOREWORD

FOREWORD

目 录

实验一　流水灯实验

说明：
主文件需要包括以下文件“config.h”,”lamp.h”
主函数：
KEYInit()，是用于需要用到的键盘的初始化
KEYSendWord(uint16)是用于向 595 送片选信号;
KEYSendBit();是让 595 中的数据依次移动一位，最后移位丢弃，并在第一位置 1

```
/************************************Copyright (c)*********************************/
/**
;**                                  西南科技大学计算机科学学院
;**                                    http://www.cs.suswt.edu.cn
;**
;**                日期：      2007.09.21
;**                描述：      西南科技大学计算机学院 CS-II 型实验板流水灯工程序头文件
;**                作者：
;**
;**-------------lamp.h 文件
;**--------------------------------------------------------------------------------------*/

/**--------------------------------------------------------------------------------
*****************************************************************************/
 #ifndef _LAMP_H_                          /* avoid being incude more than once */
 #define _LAMP_H_
/************************************************
 *                Header file                   *
 ************************************************/
 #include "config.h"

/************************************************
 *                Macros                        *
 ************************************************/

#define   KEYBOARD_SMAT   0xffff00ff
//虽然大部分的复位值为 00，即其 gpio 功能！但是部分引脚不是这样的所以这句很有必要
#define   KEYBOARD_SCK    0X00000010
#define   KEYBOARD_SI     0x00000040
#define   KEYBOARD_RCK    0x00000080
/************************************************
 *            Function declaration              *
 ************************************************/
```

```
extern void    KEYInit(void);
extern void    KEYSendWord(uint16 display_data);
extern void KEYSendBit(void);
extern void    LEDDelayNS(uint32    dly);
 #endif
 /*********************************************************************************
 *                              End of Entire File                              *
 *********************************************************************************/



/***********************************Copyright (c)*******************************/
/**
;**                              西南科技大学计算机科学学院
;**                              http://www.cs.suswt.edu.cn
;**
;**              日期：      2007.09.21
;**              描述：      西南科技大学计算机学院 CS-II 型实验板流水灯工程序
;**              作者：
;**
;**------------lamp.c 文件
;**---------------------------------------------------------------------------------------*/

/**---------------------------------------------------------------------------------------
****************************************************************************/
#define _LAMP_C_
#ifdef    _LAMP_C_
/**************************************************
 *            Header File                         *
 **************************************************/
#include "config.h"
#include "lamp.h"

/**************************************************
 *            Globale variable                  *
 **************************************************/

/*********************************************************************************
 * Function Name:     KEYInit
 * Description    :  键盘端口初始化
 *********************************************************************************/
void KEYInit(void)
{
     PINSEL0 &= KEYBOARD_SMAT;
```

```
    IO0DIR |= KEYBOARD_SCK;
    IO0DIR |= KEYBOARD_RCK;
    IO0DIR |= KEYBOARD_SI;

}

/*****************************************************************************
 * Function Name: KEYSendWord
 * Description   : 向 595 端口传一个字数据。用来选重键盘位的。其中比有且只有一位为 0
 *****************************************************************************/
void KEYSendWord(uint16 display_data)
{
    uint8 send_keycount;
    IO0CLR=KEYBOARD_RCK;
    for(send_keycount=0x00;send_keycount<0x10;send_keycount++)
    {
        IO0CLR=KEYBOARD_SCK;
        if((display_data&(0x01<<send_keycount))==0x00)//注意运算符号的优先级
        {
            IO0CLR = KEYBOARD_SI;
        }
        else
        {
            IO0SET = KEYBOARD_SI;
        }
        IO0SET=KEYBOARD_SCK;
    }
    IO0SET=KEYBOARD_RCK;
}

/*****************************************************************************
 * Function Name:    KEYSendBit
 * Description   : 向 595 端口传输一位为 1 的数据,可以通过它让 595 中的数据移位
 *****************************************************************************/
void KEYSendBit()
{
    IO0CLR=KEYBOARD_RCK;
    IO0CLR=KEYBOARD_SCK;
    IO0SET=KEYBOARD_SI;
    IO0SET=KEYBOARD_SCK;
    IO0SET=KEYBOARD_RCK;
}
/*****************************************************************************
 * Function Name:    LEDDelayNS
```

```
 * Description    :  延时函数
 *********************************************************************************/
void    LEDDelayNS(uint32    dly)
{
    for(; dly > 0; dly--)
    {
        uint32    i;
        for(i = 0; i < 5000; i++)
        {
          ;
        }
    }
}

#endif
/*********************************************************************************
 *                                  End of Entire File                           *
 *********************************************************************************/
```

实验二　数码管

说明：

在主文件中；需要包括以下两条语句：

#include “config.h”,#include “dataled.h”

在主函数中

（1）首先初始化数码管 LEDInit();

（2）调用 LEDPrint(uint32 value);将需要显示的 value 转换到 缓冲区形成数码管可认识的编码也即输入即所得，如输入 0X12345678,则数码管显示则是 12345678

（3）循环函数;调用 LEDScan()进行扫描，再调用 LEDDelayNS()进行短暂延时，再进行循环

```
/*****************************************Copyright*****************************/
/**
;**                                  西南科技大学计算机科学学院
;**                                   http://www.cs.suswt.edu.cn
;**
;**                日期：      2007.09.21
;**                描述：      DataLed.h  西南科技大学计算机学院 CS-II 型实验板数码管工
;**                            作程序头文件该文件给 DataLed.c 提供底层接口
;**                作者：
;**
;**-------------DataLed.h 文件
;**-------------------------------------------------------------------------------------*/
/**--------------------------------------------------------------------------------------
*******************************************************************************/
 #ifndef _DATALED_H_                        /* avoid being incude more than once */
 #define _DATALED_H_
/************************************************
 *                Header file                   *
 ************************************************/
 #include "config.h"

/************************************************
 *                Macros                        *
 ************************************************/
 #define   DATALED_SMAT    0xfffffc03
//虽然大部分的复位值为 00，（gpio 功能）但是部分引脚不是这样的！所以这句很有必要
 #define   DATALED_SCK     0X00020000                  //p0.17  控制单个移位，
 #define   DATALED_SI      0x00040000                  //p0.18 HC595 装载值引脚
 #define   DATALED_RCK     0x00100000                  //p0.20    控制八位发送
 #define   LEDCODENUMBER       0x10


 /************************************************
```

```
 *                Function declaration                    *
 ************************************************/
 extern void LEDInit(void);                                  //数码管初始化
 extern void LEDScan(void);                                  //数码管扫描函数
 extern void LEDPrint(uint32 display_msg);
 extern void LEDSendDisplay(uint8 display_data,uint8 display_number);
 extern void LEDDelayNS(uint32    dly);                      //延时函数

 #endif
 /*****************************************************************************
 *                                End of Entire File                              *
 *****************************************************************************/



/*************************************Copyright*********************************/
/**
;**                              西南科技大学计算机科学学院
;**                               http://www.cs.suswt.edu.cn
;**
;**              日期:     2007.09.21
;**              描述:     dataled.c  西南科技大学计算机学院 CS-II 型实验板数码管工作
;**                        程序，控制数码管底层显示功能，负责将数据传输到目标
;**              作者:
;**
;**-------------dataled.c 文件
;**-------------------------------------------------------------------------------------------*/
/**-------------------------------------------------------------------------------------------
*****************************************************************************/
#define _DATALED_C_
#ifdef    _DATALED_C_
/************************************************
 *                Header File                     *
 ************************************************/
#include "dataled.h"
#include "config.h"

/************************************************
 *                Globale variable                *
 ************************************************/
 const    uint8 gc_display_code[LEDCODENUMBER]=
                                                 {
                                                   0x3f,0x06,0x5b,0x4f,0x66,0x6d,0x7d,0x07,
                                                   0x7f,0x6f,0x77,0x7c,0x39,0x5e,0x79,0x71
```

```
                                                    };
uint8 gc_dataled_buffer[8];
uint8 gc_ptr_bit = 0x00;
uint8 gc_dis_bit = 0x7f;                                    //片选
/*****************************************************************************
 * Function Name:     LEDDelayNS
 * Description    :  延时函数
 *****************************************************************************/
void    LEDDelayNS(uint32    dly)
{
    for(; dly > 0; dly--)
    {
        uint32    i;
        for(i = 0; i < 5000; i++)
        {
          ;
        }
    }
}

/*****************************************************************************
 * Function Name:    LEDInit
 * Description    :  初始化数码管模
 *****************************************************************************/
void LEDInit(void)
{
     PINSEL1 &= DATALED_SMAT;                       //设置端口为 GPIO 功能
     IO0DIR   |= DATALED_SCK;                       //设置为输出方向
    IO0DIR   |= DATALED_RCK;
    IO0DIR   |= DATALED_SI;

}
/*****************************************************************************
 * Function Name: LEDPrint
 * Description    : 显示函数，将显示代码放入缓冲区。
 *****************************************************************************/
void LEDPrint(uint32 display_msg)
{
     uint8 temp = 0x00;
     while(temp <= (7 * 0x04))
     {
          gc_dataled_buffer[temp / 4] = gc_display_code[(display_msg >> temp) & 0x0f];
          //display_msg 从左向右移动，每次取四位，用于找出对应的编码放入缓冲区
          temp += 4;
```

```
    }
}
/*******************************************************************************
 * Function Name:   LEDSendDisplay
 * Description   :   通过 595 分别把片选码送给片选端，把缓冲区的显示码送给显示端
*******************************************************************************/
void LEDSendDisplay(uint8 display_data,uint8 display_number)
{
    uint8 send_count;
    IO0CLR = DATALED_RCK;                                   //用于并行发送时需要的下降沿
    for(send_count = 0x00;send_count < 0x08;send_count++)
    {
        IO0CLR = DATALED_SCK;                                //用于串行移位需要的下降沿

        if((display_number&(0x01<<send_count)) == 0x00) //注意运算符号的优先级
        {
            IO0CLR = DATALED_SI;
        }
        else
        {
            IO0SET = DATALED_SI;
        }


        IO0SET = DATALED_SCK;
    }
    for(send_count = 0x00;send_count < 0x08;send_count++)
    //与上同理，不过传送的是字符显示数据
    {
        IO0CLR = DATALED_SCK;

        if((display_data & (0x80 >> send_count)) == 0x00)
        {
            IO0CLR = DATALED_SI;
        }
        else
        {
            IO0SET = DATALED_SI;
        }

        IO0SET = DATALED_SCK;
    }

    IO0SET = DATALED_RCK;
```

```
}

/*******************************************************************************
 * Function Name:    LEDScan
 * Description    :   扫描数码管。以达到看上去 8 个数码管同时亮
*******************************************************************************/
 void LEDScan(void)
{
    LEDSendDisplay(gc_dataled_buffer[7-gc_ptr_bit],gc_dis_bit);
//传送字符显示数据和片选数据

    gc_dis_bit = ((gc_dis_bit >> 1) | 0x80);                    //片选移位用于选中下一数码管
    gc_ptr_bit++;                                               //用于记录当前显示的数码管
    if(gc_ptr_bit == 0x08)                                      //防止越界
    {
        gc_ptr_bit = 0x00;
        gc_dis_bit = 0x7f;
    }
}

#endif
/*******************************************************************************
 *                                End of Entire File                           *
 *******************************************************************************/
```

实验三　　键盘实验

说明：

主文件需要包括以下几句：

#include”config.h”, #include “dataled.h” #include “keyboard.h”

主函数中调用如下：

首先需要如下定义 uint8 ret_key;该定义变量用于判断是否有键按下

LEDInit();该函数主要用于数码管初始化

KEYInit();该函数主要用于键盘初始化

```
while(1)
  {
        LEDScan();
        ret_key = KEYScan(); //键盘扫描返回值赋给 ret_key 变量
        if(ret_key != 0xff)   //判断是否有键按下
        {
            LEDPrint(ret_key);
        }
    }
/*************************************Copyright (c)*****************************/
 ;**
;**                                西南科技大学计算机科学学院
;**                                 http://www.cs.suswt.edu.cn
;**
;**                日期：      2007.09.21
;**                描述：      keyboard.h  西南科技大学计算机学院 CS-II 型实验板键盘工作
;**                            程序头文件  该文件给 keyboard.c 做声明以及一些宏定义，同
;**                            时也给外部的提供接口函数
;**                作者：
;**
;**-------------keyboard.h 文件
;**-------------------------------------------------------------------------------------------*/
/**-------------------------------------------------------------------------------------------
*******************************************************************************/
 #ifndef _KEYBOARD_H_                        /* avoid being incude more than once */
 #define _KEYBOARD_H_
/************************************************
 *                Header file                   *
 ************************************************/
 #include "config.h"

/************************************************
 *                Macros                        *
 ************************************************/
```

```
#define    KEYBOARD_SMAT    0xffff00ff
//虽然大部分的复位值为 00，即其 gpio 功能！但是部分引脚不是这样的所以这句很有必要
#define    KEYBOARD_SCK     0X00000010
#define    KEYBOARD_SI      0x00000040
#define    KEYBOARD_RCK     0x00000080
#define    KEYBOARD_KEY     0x00000020
 /***********************************************
 *                Function declaration                    *
 ***********************************************/
extern void    KEYInit(void);
extern uint8 KEYScan(void);
extern void    KEYSendWord(uint16 display_data);

 #endif
 /*****************************************************************************
 *                                    End of Entire File                                    *
 *****************************************************************************/


/************************************Copyright(c)*********************************/
/**
;**                                    西南科技大学计算机科学学院
;**                                      http://www.cs.suswt.edu.cn
;**
;**                日期：     2007.09.21
;**                描述：     keyboard.c  西南科技大学计算机学院 CS-II 型实验板键盘工作程
;**                           序，控键盘扫描功能，负责将数据传输到目标，并控制 595 对
;**                           键盘进 行扫描完成键盘工作的功能
;**                作者：
;**
;**-------------keyboard.c 文件
;**-------------------------------------------------------------------------------------------*/
/**-------------------------------------------------------------------------------
*****************************************************************************/
#define _KEYBOARD_C_
#ifdef    _KEYBOARD_C_
/***********************************************
 *                Header File                             *
 ***********************************************/
#include "config.h"
#include "keyboard.h"
```

```
/*********************************************
 *                Globale variable                                *
 *********************************************/
uint8 gc_key_bit = 0x00;

/*****************************************************************************
 * Function Name:     KEYInit
 * Description    :  键盘端口初始化
 *****************************************************************************/
void KEYInit(void)
{
    PINSEL0 &= KEYBOARD_SMAT;
    IO0DIR |= KEYBOARD_SCK;
    IO0DIR |= KEYBOARD_RCK;
    IO0DIR |= KEYBOARD_SI;
    IO0DIR &= (KEYBOARD_KEY^0XFFFFFFFF); //设置端口为输入
}

/*****************************************************************************
 * Function Name: KEYSendWord
 * Description    : 向 595 端口传一个字数据。用来选重键盘位的。其中有且只有一位为 0
 *****************************************************************************/
void KEYSendWord(uint16 display_data)
{
    uint8 send_keycount;
    IO0CLR=KEYBOARD_RCK;
    for(send_keycount=0x00;send_keycount<0x10;send_keycount++)
    {
        IO0CLR=KEYBOARD_SCK;
        if((display_data&(0x01<<send_keycount))==0x00)          //注意运算符号的优先级
        {
            IO0CLR = KEYBOARD_SI;
        }
        else
        {
            IO0SET = KEYBOARD_SI;
        }
        IO0SET=KEYBOARD_SCK;
    }
    IO0SET=KEYBOARD_RCK;
}
```

```
/*****************************************************************************
 * Function Name:     KEYSendBit
 * Description   :  向 595 端口传输一位为 1 的数据,可以通过它让 595 中的数据移位
 *****************************************************************************/
void KEYSendBit()
{
    IO0CLR=KEYBOARD_RCK;
    IO0CLR=KEYBOARD_SCK;
    IO0SET=KEYBOARD_SI;                                  //置位，用于片选移动
    IO0SET=KEYBOARD_SCK;
    IO0SET=KEYBOARD_RCK;
}
/*****************************************************************************
 * Function Name:    KEYScan
 * Description   :  扫描键盘，并给予返回值，扫描的方式是通过移位的方式来进行的
 *****************************************************************************/
uint8 KEYScan(void)
{
    if(gc_key_bit==0x00)
    {
        KEYSendWord(0x7fff);
    }
    else
    {
        KEYSendBit();
    }
    if((IO0PIN&KEYBOARD_KEY)==0x00)              //如果返回值为 0，说明有键按下
    {
        return gc_key_bit;
    }
    gc_key_bit++;
    if(gc_key_bit==0x10)
    {
        gc_key_bit=0x00;
    }
    if((IO0PIN&KEYBOARD_KEY)!=0)                 //无键按下就返回 0xff  做标识
    {
        return 0xff;
    }
        return 0;
   #endif
/*****************************************************************************
 *                                End of Entire File                         *
 *****************************************************************************/
```

实验四　点阵屏汉字显示实验

说明：将 matrix.h 包含到头文件当中，初始化函数 MATRIXInit() 放到 main 函数的 void DeviceInit(void)函数中用以初始化；在 main 函数中调用 MATRIXPrintFont(uint16 FontNumber)函数实现将输出内容放入缓冲区，注意 FontNumber 为字库中要显示内容对应的字号；利用 MATRIXScan()函数实现对点阵屏的扫描，初学者可以将它直接放到 while 循环中，如果掌握了定时器的同学还可以利用定时器溢出中断等来实现点阵屏扫描。

```
/***********************************Copyright (c)*********************************/
/**
;**                                  西南科技大学计算机科学学院
;**                                    http://www.cs.suswt.edu.cn
;**
;**             日期：      2007/09/20
;**             描述：     西南科技大学计算机学院 CS-II 型实验板点阵屏驱动工作程序头文件
;**             作者：
;**
;**-------------matrix.h 文件
;**-------------------------------------------------------------------------------------------*/
/**-------------------------------------------------------------------------------------------
*********************************************************************************/
#ifndef _MATRIX_H_                  /* Performed as a on-off, to avoid the header */
#define _MATRIX_H_                  /* file being included more than once          */
/************************************************
 *                   Header file                *
 ************************************************/
 #include "config.h"
/************************************************
 *                   Macros                     *
 ************************************************/
 #define MATRIX_RCLK    0x00100000
 #define MATRIX_SRCLK 0x00020000
 #define MATRIX_SI_X    0x00040000
 #define MATRIX_SI_Y    0X00080000
 #define MATRIX_MAT     0xfffffc03
 #define MATRIX_NUMBER 0x10
 /************************************************
 *               Function declaration           *
 ************************************************/
 extern void MATRIXSendData(uint16 disdata,uint16 dispos);
 //串并转换发送 16 位片选位，16 位数据显示位
 extern void MATRIXInit(void);                                  //点阵初始化程序
 extern void MATRIXScan(void);                                  //点阵扫描程序
 extern void MATRIXPrintFont(uint16 FontNumber);
```

```
 //点阵显示程序，显示字库中内容
 #endif
/*******************************************************************************
 *                                End of Entire File                           *
 *******************************************************************************/




/*****************************************Copyright*******************************/
/**
;**                                西南科技大学计算机科学学院
;**                                http://www.cs.suswt.edu.cn
;**
;**        日期：    2007/09/20
;**        描述：    西南科技大学计算机学院 CS-II 型实验板点阵屏驱动工作程序,这里默认
;**                  字库为“我型我秀”四个字，根据实际可添加其他尽字库。
;**                  使用方法：首先初始化，然后将 MATRIXScan()函数放入定时器
;**                  或 while 中，再调用 MATRIXPrintFont()调用需要的字库数据。
;**        作者：
;**
;**-------------matrix.c 文件
;**-------------------------------------------------------------------------------------------------*/
/**-------------------------------------------------------------------------------
*********************************************************************************/
#define _MATRIX_C_
#ifdef   _MATRIX_C_
/*************************************************
 *              Header File                      *
 *************************************************/
 #include"matrix.h"
 #include"config.h"
/*************************************************
 *              Globale variable                 *
 *************************************************/
 uint8   matrix_dis_bit = 0x00;
 uint16 matrix_dis_pos = 0x0001;
 uint16 matrix_mem[MATRIX_NUMBER];
 const uint16 matrix_font_data[][16] = {
 /*--  文字:  我  --*/
 /*--  宋体 12;  此字体下对应的点阵为：宽 x 高=16x16    --*/

0xF9BF,0xC7AF,0xF7B7,0xF7B7,0xF7BF,0x0001,0xF7BF,0xF7B7,0xF1D7,0xC7CF,0x37DF,0
xF7AF,0xF66D,0xF7F5,0xD7F9,0xEFFD,
```

```
 /*--  文字:  型  --*/
 /*--  宋体 12;  此字体下对应的点阵为：宽 x 高=16x16    --*/

0xFFFB,0x807B,0xEDDB,0xEDDB,0x001B,0xEDDB,0xEDDB,0xDDDB,0xDDFB,0xBEEB,0xFEF7,0xC003,0xFEFF,0xFEFF,0x0001,0xFFFF,
 /*--  文字:  我  --*/
 /*--  宋体 12;  此字体下对应的点阵为：宽 x 高=16x16    --*/

0xF9BF,0xC7AF,0xF7B7,0xF7B7,0xF7BF,0x0001,0xF7BF,0xF7B7,0xF1D7,0xC7CF,0x37DF,0xF7AF,0xF66D,0xF7F5,0xD7F9,0xEFFD,
 /*--  文字:  秀  --*/
 /*--  宋体 12;  此字体下对应的点阵为：宽 x 高=16x16    --*/

0xFF07,0xC0FF,0xFEFF,0x8001,0xFC3F,0xF2CF,0xCEF1,0x3EFB,0xE00F,0xFDDF,0xFDBF,0xFB03,0xFBFB,0xF7FB,0xEFD7,0xDFEF
 };                                                    //利用字库软件生成字库
/*******************************
 *        Function  Implement    *
 *******************************/
/*****************************************************************************
 * Function Name:      void MATRIXSendData(uint16 disdata,uint16 dispos)      *
 * Description   :    串行发送 16 为数据位和 16 为片选位                      *
 *****************************************************************************/
void MATRIXSendData(uint16 disdata,uint16 dispos)
{
    uint8 tmp;
    IO0CLR = MATRIX_RCLK ;                          //同步发送脉冲 RCK
    for(tmp=0x00;tmp<0x10;tmp++)
    {
        IO0CLR = MATRIX_SRCLK;
        if((disdata&0x8000)==0x00)                  //      高---------------------->低  (0 亮)
        {                                           //      * * * * * * * * * * * * * * * *
            IO0CLR = MATRIX_SI_Y;                   //   高*
        }                                           //    | *
        else                                        //    | *
        {                                           //    | *
            IO0SET = MATRIX_SI_Y;                   //    | *
        }                                           //    | *
        if((dispos&0x8000)==0x00)                   //    | *
        {                                           //    | *
            IO0CLR = MATRIX_SI_X ;                  //    | *
        }                                           //    | *
        else                                        //    | *
        {                                           //    | *
```

```
            IO0SET = MATRIX_SI_X;              //     | *
        }                                      // 低  *
        disdata = disdata<<1;                  //      *
        dispos = dispos<<1;                    // （1 亮）
        IO0SET = MATRIX_SRCLK;
    }
    IO0SET = MATRIX_RCLK ;
}

/*****************************************************************************
 * Function Name:    void MATRIXInit(void)                                   *
 * Description   :   点阵初始化设置，包括引脚连接模块和输入输出设置          *
 *****************************************************************************/
void MATRIXInit(void)
{
    PINSEL1 &= MATRIX_MAT;                 //引脚连接模块设置，这里选择
    P0.17～P0.20 作 GPIO
    IO0DIR |= MATRIX_SRCLK;                //方向输出
    IO0DIR |= MATRIX_RCLK;
    IO0DIR |= MATRIX_SI_X;
    IO0DIR |= MATRIX_SI_Y;
    MATRIXSendData(0xffff,0x0000);         //初始化矩阵内数据，初始无显示
}

/*****************************************************************************
 * Function Name:     void MATRIXScan(void)                                  *
 * Description   :  点阵扫描函数，依次扫描显示一行数据                       *
 *****************************************************************************/

void MATRIXScan(void)
{
    MATRIXSendData(matrix_mem[matrix_dis_bit],matrix_dis_pos);
    matrix_dis_bit++;
    matrix_dis_pos = (matrix_dis_pos<<1)&0xfffe;
    if(matrix_dis_bit == 0x10)
    {
        matrix_dis_bit = 0x00;
        matrix_dis_pos = 0x0001;
    }
}

/*****************************************************************************
 * Function Name:     void MATRIXPrintFont(uint16 FontNumber)                *
 * Description   :  点阵显示函数，显示当前缓冲区的数据                       *
```

```
  ******************************************************************************/
void MATRIXPrintFont(uint16 FontNumber)
{
      uint8 tmp;
      for(tmp=0x00;tmp<0x10;tmp++)
      {
            matrix_mem[tmp] = matrix_font_data[FontNumber][tmp]; //调用字库
      }
}

#endif
/*******************************************************************************
  *                                End of Entire File                           *
  *******************************************************************************/
```

实验五　外部中断实验

说明：在 main 文件中要包含 dataled.h 和 sounder.h。dataled 模块可以用前面用到过的，其用法参考数码管的实验。sounder 模块实现发生器的发声、停止发声、初始化等功能。将 SOUNDERInit()放到 main 文件的设备初始化函数 void DeviceInit(void)中，SOUNDEROn 和 SOUNDEROff 分别是发生器响和发生器灭的函数，一次调用即可。这里将他们放到外部中断服务函数中，达到一次中断响应响一下再灭的效果。数码管的显示函数 LEDPrint()也放到外部中断服务函数中，每中断一次显示的数据加一，以用来计数中断次数。

```
/************************************Copyright (c)********************************/
;**                          西南科技大学计算机科学学院
;**                          http://www.cs.suswt.edu.cn
;**
;**        日期：    2007/09/20
;**        描述：    西南科技大学计算机学院 CS-II 型实验板外部中断工作程序头文件
;**        作者：
;**
;**------------exint.h 文件
;**--------------------------------------------------------------------------------------- */
/**---------------------------------------------------------------------------------------
******************************************************************************/
#ifndef _EXTINT_H_                    /* avoid being incude more than once */
#define _EXTINT_H_
/***********************************************
 *                Header file                  *
 ***********************************************/
 #include"config.h"
/***********************************************
 *                Macros                       *
 ***********************************************/

 /***********************************************
 *            Function declaration             *
 ***********************************************/
 extern void EINT0Init(void);                  //外部中断 0 初始化程序
 #endif

/******************************************************************************
 *                          End of Entire File                                *
 ******************************************************************************/
```

```
/***********************************Copyright (c)*********************************/
;**                                  西南科技大学计算机科学学院
;**                                    http://www.cs.suswt.edu.cn
;**
;**             日期：      2007/09/20
;**             描述：      西南科技大学计算机学院 CS-II 型实验板中断工作程序头文件
;**             作者：
;**
;**-------------exint.h 文件
;**-------------------------------------------------------------------------------------------*/
/**--------------------------------------------------------------------------------------------
*******************************************************************************************/
#define _EXINT_C_
#ifdef   _EXINT_C_
/***********************************************
 *              Header File                    *
 ***********************************************/
 #include"config.h"
 #include"sounder.h"
 #include"dataled.h"
/***********************************************
 *              Globale variable               *
 ***********************************************/



/**-------------------------------------------------------------**/
//                              外部中断 0
/*******************************************************************************
Function Name: void __irq IRQEint0(void)                                       *
Description   : 外部中断 0 的中断服务函数，需要利用外部中断 0 进行中断的程序都可以  *
                放在这个函数中                                                 *
                ads1.2 规定，在定义中断服务函数时，必须加入关键字__irq 保证函数返  *
                回时会切换处理器模式                                           *
                需要注意的是：注意在退出中断服务程序时，要清零相应外设的中断标志 *
                以及 VICVectAddr，为响应下一次中断做好准备                     *
*******************************************************************************/
uint32 tmp_cnt=100000;
uint32 dis_cnt = 0;
void __irq IRQEint0(void)
{
    SOUNDEROn();
    while(tmp_cnt!=0)
    {
```

```
            tmp_cnt--;
        }
        LEDPrint(dis_cnt++);
        SOUNDEROff();
        tmp_cnt=100000;
        /*  上面添加需要定时器中断的代码*/
        EXTINT = 0x01;                          //清除中断
        VICVectAddr = 0;                        //清除地址
}

/*****************************************************************************
Function Name:    void EINT0Init(void)                                       *
Description    :  外部中断 0 初始化设置,主要设置引脚连接模块,电平触发方式,中断    *
                  优先级,中断服务程序入口地址等                              *
*****************************************************************************/
void EINT0Init(void)
{
        PINSEL1 = (PINSEL1&0xfffffffc)|0x01;      //P0.16 EINT0
        EXTMODE |= 0x03; /* something wrong */    //EINT0 使用边沿触发
        EXTPOLAR &= 0X01;                         //EINT0 高电平或上升沿有效

        VICIntSelect &=0xffff7fff;                //设置 EINT0 为 IRQ 中断
        VICVectCntl1 = 0x20|14;
        //为中断源分配向量 IRQ 的优先级 ， VICVectCntl n，n 值越小优先级越高，0<=n<=15
                                                  //中断源序号为 14   代表外部中断 0 中断
        VICVectAddr1 = (uint32)IRQEint0;
        //向量地址寄存器   为该中断优先级设置服务程序入口地址
        VICIntEnable = 0x00004000;                //中断使能  允许相应中断源产生中断
}
#endif
/*****************************************************************************
 *                              End of Entire File                           *
 *****************************************************************************/
```

实验六　定时器实验

说明：将 timer.h 和 sounder.h 包含到 main 函数中；sounder 模块实现发生器的发声、停止发声、初始化等功能。将 SOUNDERInit()放到 main 文件的设备初始化函数 void DeviceInit(void)中，再将 SOUNDEROn()和 SOUNDEROff()函数放到定时器 c 文件中的中断处理函数中去，void __irq IRQTimer0(void)，用以实现相应控制。

Timer 模块主要实现定时器 0 和定时器 1 的相关初始化，并提供中断处理函数来实现相应功能。要使用哪一个定时器就将起初始化程序放入 main 文件的 void DeviceInit(void)中，TIMER0Init()或 TIMER1Init()，另外需要注意，如果要使用定时器，还要在 main 函数提前打开定时器，即调用函数 TIMER0Start()或 TIMER1Start()。在中断处理函数中实现相应功能。

```
/***********************************Copyright (c)*******************************/
/**
;**                                西南科技大学计算机科学学院
;**                                  http://www.cs.suswt.edu.cn
;**
;**            日期：     2007/09/20
;**            描述：     西南科技大学计算机学院 CS-II 型实验板定时器 0 和定时器 1 驱动工
;**                       作程序头文件
;**            作者：
;**
;**-------------timer.h 文件
;**-------------------------------------------------------------------------------------------*/
/**-------------------------------------------------------------------------------
*******************************************************************************/
#ifndef _TIMER_H_                       /* avoid being incude more than once */
#define _TIMER_H_
/*************************************************
 *                  Header file                  *
 ************************************************/
 #include"config.h"
/*************************************************
 *                  Macros                       *
 ************************************************/

 /*************************************************
 *              Function declaration             *
 ************************************************/
 extern void TIMER0Init(void);                    //定时器 0 初始化设置
 extern void TIMER0Start(void);                   //启动定时器 0
 #endif
/*******************************************************************************
 *                              End of Entire File                             *
 *******************************************************************************/
```

```
**********************************Copyright (c)*******************************/
/**
;**                                西南科技大学计算机科学学院
;**                                  http://www.cs.suswt.edu.cn
;**
;**            日期：     2007/09/20
;**            描述：     西南科技大学计算机学院 CS-II 型实验板定时器 0 和定时器 1 驱动
;**                       工作程序
;**            作者：
;**
;**------------timer.c 文件
;**-------------------------------------------------------------------------------------*/
/**--------------------------------------------------------------------------------------
*******************************************************************************/
#define _TIMER_C_
#ifdef   _TIMER_C_
/*********************************************
 *              Header File                  *
 *********************************************/
 #include"config.h"
 #include"sounder.h"
 #include"led.h"
/*********************************************
 *              Globale variable             *
 *********************************************/
 uint32 div0 = 0;
 uint32 div1 = 0;
 uint32 overflow = 0;

#define TIMER0_ENABLE
#ifdef   TIMER0_ENABLE
/*******************************************************************************
Function Name:void __irq IRQ_Timer0(void)                                      *
Description    : Timer0 的中断服务函数，需要利用 Timer0 进行中断的程序都可以放在这个*
                 函数中                                                        *
                 ads1.2 规定，在定义中断服务函数时，必须加入关键字__irq 保证函数返  *
                 回时会切换处理器模式                                          *
                 需要注意的是：注意在退出中断服务程序时，要清零相应外设的中断标志*
                 以及 VICVectAddr，为响应下一次中断做好准备                    *
*******************************************************************************/
uint8 sounder_flag = TRUE;
void __irq IRQTimer0(void)
 {
    if(sounder_flag)
```

```
    {
        SOUNDEROn();
        sounder_flag = FALSE;
    }
    else
    {
        SOUNDEROff();
        sounder_flag = TRUE;
    }

    /* 上面添加需要定时器中断的代码*/
    T0IR=0X01;                                          //清除中断
    VICSoftIntClear=0x00000010;
    VICVectAddr=0x00;                                   //清除地址
}

/*****************************************************************************
Function Name:      void Timer0_InterruptSet(void)                              *
Function Description :对中断向量的设置，包括中断类型，中断优先级，中断服务地址    *
                      中断使能。                                                *
*****************************************************************************/
void TIMER0InterruptSet(void)
{
    VICIntSelect &= 0xffffffef;
    //第 4、5 位分别代表定时计数器 0、1   设置中断类型
    VICVectCntl0 = 0x24;
    //为中断源分配向量 IRQ 的优先级 ， VICVectCntl n，n 值越小优先级越高，0<=n<=15
    //0010 0100 中断源序号为 4   代表计数器 0 中断
    VICVectAddr0 = (int)IRQTimer0;
    //向量地址寄存器   为该中断优先级设置服务程序入口地址
    VICIntEnable = 0x00000010;
    //中断使能 允许相应中断源产生中断
}
/*****************************************************************************
Function Name:   void Timer0_Init(void)                                         *
Description   :定时器计数器 0 初始化设置，包括预分频值，匹配模式，启动复位等。   *
*****************************************************************************/
void TIMER0Init(void)
//另外一种写法 void Timer1_Init(uint32 Div,uint32 Ovf)
{
    T0TC = 0;
    T0PR = div0;                //设置预分频值   定时器计数时钟频率 Fpclk/(PR+1)
    T0MCR = 0x03;               //设置匹配模式 匹配后复位 T0TC，并产生中断标志
    T0MR0 = Fpclk;              //设置匹配值   可以理解为计数器益出值 设置 1ms 匹配值
```

```
    T0TCR = 0x03;                  //启动并复位
    TIMER0InterruptSet();
  }

/*****************************************************************************
Function Name:     void Timer0_Start(void)                                    *
Function Description :   定时器计数器 0 启动                                  *
*****************************************************************************/
void TIMER0Start(void)
{
    T0TCR = 0x01;                              //定时器使能，相当与定时计数器的开关
}
#endif

//#define TIMER1_ENABLE
#ifdef TIMER1_ENABLE

/*****************************************************************************
Function Name:void __irq IRQ_Timer1(void)                                     *
Description    : Timer1 的中断服务函数，需要利用 Timer1 进行中断的程序都可以放在这个*
                 函数中                                                       *
                 ads1.2 规定，在定义中断服务函数时，必须加入关键字__irq 保证函数返  *
                 回时会切换处理器模式                                         *
                 需要注意的是：注意在退出中断服务程序时，要清零相应外设的中断标志 *
                 以及 VICVectAddr，为响应下一次中断做好准备                   *
*****************************************************************************/
void __irq IRQTimer1(void)
  {
    /* 上面添加需要定时器中断的代码*/
    T1IR=0X01;                                  //清除中断
    VICSoftIntClear=0x00000010;
    VICVectAddr=0x00;                           //清除地址
  }

/*****************************************************************************
Function Name:       void Timer1_InterruptSet(void)                           *
Function Description :对中断向量的设置，包括中断类型，中断优先级，中断服务地址      *
                      中断使能。                                              *
*****************************************************************************/
void TIMER1InterruptSet(void)
  {
    VICIntSelect &= 0xffffffdf;          //第 4、5 位分别代表定时计数器 0、1  设置中断类型
    VICVectCntl0 = 0x25;
    //为中断源分配向量 IRQ 的优先级 ， VICVectCntl n，n 值越小优先级越高，0<=n<=15
```

```
    //0010 0100  中断源序号为 4   代表计数器 0 中断
    VICVectAddr0 = (int)IRQTimer1;
    //向量地址寄存器   为该中断优先级设置服务程序入口地址
    VICIntEnable = 0x00000020;
    //中断使能  允许相应中断源产生中断
 }

/*****************************************************************************
Function Name:   void Timer1_Init(void)                                      *
Description   :定时器计数器 1 初始化设置，包括预分频值，匹配模式，启动复位等。      *
*****************************************************************************/
void TIMER1Init(void)
{
    T1TC = 0;
    T1PR = div1;                  //设置预分频值   定时器计数时钟频率  Fpclk/(PR+1)
    T1MCR = 0x03;                 //设置匹配模式  匹配后复位 T0TC，并产生中断标志
    T1MR0 = Fpclk/1000;           //设置匹配值   可以理解为计数器益出值  设置 1ms 匹配值
    T1TCR = 0x03;                 //启动并复位
    TIMER1InterruptSet();
 }

/*****************************************************************************
Function Name:    void Timer0_Start(void)                                    *
Description   :   定时器计数器 0 启动
*****************************************************************************/
void TIMER1Start(void)
{
    T1TCR = 0x01;                       //定时器使能，相当与定时计数器的开关
}
#endif


#endif
/*****************************************************************************
 *                              End of Entire File                           *
 *****************************************************************************/
```

```
/************************************Copyright (c)********************************/
/**
;**                              西南科技大学计算机科学学院
;**                                http://www.cs.suswt.edu.cn
;**
;**           日期：     2007/09/20
;**           描述：    西南科技大学计算机学院 CS-II 型实验板发声器驱动工作程序头文件
;**           作者：
;**
;**------------sounder.h 文件
;**------------------------------------------------------------------------------------*/
/**-------------------------------------------------------------------------------------
*********************************************************************************/
#ifndef _SOUNDER_                 /* Performed as a on-off, to avoid the header */
#define _SOUNDER_                /* file being included more than once              */
/*******************************
  *          Head File                     *
  *******************************/
  #include "config.h"
/*******************************
  *        Macros Segment              *
  *******************************/
  #define SOUNDER 0x01000000
  #define SOUNDER_PINSEL 0xfffcffff                          //P0.24
/*******************************
  *        Function Declaration      *
  *******************************/
  extern void SOUNDERInit(void);                              //发声器初始化
  extern void SOUNDEROn(void);                               //发声器发声
  extern void SOUNDEROff(void);                             //发声器停止

  #endif
/*********************************************************************************
*                                  The end of the entire file                                  *
*********************************************************************************/
```

```
/***********************************Copyright (c)*********************************/
/**
;**                                  西南科技大学计算机科学学院
;**                                    http://www.cs.suswt.edu.cn
;**
;**              日期：      2007/09/20
;**              描述：     西南科技大学计算机学院 CS-II 型实验板定时器驱动工作程序头文件
;**              作者：
;**
;**-------------matrix.h 文件
;**-------------------------------------------------------------------------------------------*/
/**-------------------------------------------------------------------------------------------
*********************************************************************************************/
#define SOUNDER_ONOFF
#ifdef    SOUNDER_ONOFF
/*******************************
 *          Head File                    *
 *******************************/
 #include "sounder.h"
 #include "config.h"
/*******************************
 *        Global Variable               *
 *******************************/

/*******************************
 *        Const Segment                 *
 *******************************/

/*******************************
 *          Function    Implement       *
 *******************************/
/*********************************************************************************************
Function Name: void sounder_init(void)                                                       *
Function Description :    相关初始化                                                         *
*********************************************************************************************/
void SOUNDERInit(void)
{
    PINSEL1 &= SOUNDER_PINSEL;
    //管脚功能选择寄存器 1 的 16、17 位置 0，功能选择为 GPIO P0.24
    IO0DIR |= SOUNDER;
}
```

```
/*****************************************************************************
Function Name:    void SOUNDEROn(void)                                     *
Description    :    发声器发声函数                                          *
*****************************************************************************/
void SOUNDEROn(void)
{
    IO0CLR = SOUNDER;
}

/*****************************************************************************
Function Name:    void SOUNDEROff(void)                                    *
Description    :    停止发声器发声函数                                      *
*****************************************************************************/
void SOUNDEROff(void)
{
    IO0SET = SOUNDER;
}


#endif
/*****************************************************************************
 *                          The end of the entire file                       *
*****************************************************************************/
```

实验七　UART 串口通信模块

说明：在UART0中我们用到了键盘，数码管还有定时器。我们在程序中一定要包含这些文件。主函数要包含这些模块的头文件，并对它们进行初使化。另外，由于我们发数码管要动态刷新，我们的键盘也要动态扫描。故在定时器里面也要包含相应的头文件。在定时器中断函数中要实现键盘和数码管的动态扫描。

```
/*********************************Copyright(c)** ******************************/
/**
;**                          西南科技大学计算机科学学院
;**                             http://www.cs.suswt.edu.cn
;**
;**              日期:     2007/09/21
;**              描述:     uart.h 西南科技大学计算机学院 CS-II 型实验箱串口 0 头件
;**              作者:
;**
;**-------------uart.h 文件
;**-------------------------------------------------------------------------------------------*/
/**-------------------------------------------------------------------------------------------
*****************************************************************************/
#ifndef _UART_H_                        /* avoid being incude more than once */
#define _UART_H_
/*************************************************
  *                 Header file                   *
  ************************************************/
#include "config.h"
/*************************************************
  *                 Macros                        *
  ************************************************/
#define UART0_ENABLE
#define UART0_BUFFERDATACOUNT   16    //缓冲区大小定义

/*************************************************
  *              Function declaration             *
  ************************************************/
extern void UART0Init(uint16 baud);          //串口 0 初使化
extern void UART0SendByte(uint8 data);       //串口 0 发送一个字节的数据
extern void UART0SendString(char data[]);    //串口 0 发送字符串,最多字符个数为 16
extern uint8 UART0RecevieByte(void);         //串口 0 接收一个字节的数据
 #endif                                        /*  end of switch _FILENAME_H_ */
/*****************************************************************************
  *                              End of Entire File                           *
*****************************************************************************/
```

```
/*******************************copyright(c)**********************************/
/**
;**                              西南科技大学计算机科学学院
;**                                http://www.cs.suswt.edu.cn
;**
;**                  日期：    2007/09/21
;**                  描述：     uart.c 是西南科技大学计算机学院 CS-II 的串口 0 的 C 文件，
;**                             主要实现串口的接收与发送功能。
;**                  作者：
;**
;**-------------uart.c 文件
;**-------------------------------------------------------------------------------------------*/
/**-------------------------------------------------------------------------------------------
*******************************************************************************/
#define _UART_C_
#ifdef   _UART_C_
/*************************************************
 *              Header File                      *
 *************************************************/
#include "uart.h"
#include "string.h"

/*************************************************
 *              Globale variable                 *
 *************************************************/
uint8 gc_uart0_receivedata = 0;             //保存串口 0 传过来的一个字节的数据
uint8 gc_uart0_receivebufpointer = 0;       //患口 0 接收字符指针
uint8 gc_uart0_sendbufpointer = 0;          //串口 0 发送字符指针
uint8 gc_uart0_receivebuf[UART0_BUFFERDATACOUNT];
/*串口 0 接收数组（接收缓冲区）*/

/*****************************************************************************
 * Function Name:   void UART0SendByte(uint8 data)                           *
 * Description   :  串口 0 发送函数，一次发送一个字节的数据（查询方式）      *
 *****************************************************************************/
void UART0SendByte(uint8 data)
{
    U0THR = data;
    while((U0LSR & 0x40) == 0);
}
```

```
/*****************************************************************
 * Function Name:    void UART0SendString(char data[])           *
 * Description   :   串口 0 发送字符串函数                        *
 *****************************************************************/
void UART0SendString(char data[])
{
    uint8 string_size;              //要发送的字符数数组的大小
    uint8 string_pointer = 0;       //字符数组中待发送的数据指针
    string_size = strlen(data);
    for(string_pointer = 0; string_pointer <= string_size; string_pointer++)
    {
        U0THR = data[string_pointer];
        while((U0LSR & 0x40) == 0);

            /*UART0SendByte(data[string_pointer])也可以用发送字符来实现，但效
              率相对低*/
    }
}

/*****************************************************************************
 * Function Name:    void __irq IRQ_UART0Receive(void)                       *
 * Description   :   串口 0 接收中断服务程序，通过判断中断标志寄存器中的标识  *
                     来识别是否有接收中断。实现把串口 0 传来的数据保存        *
*****************************************************************************/
void __irq IRQ_UART0Receive(void)
{
    if((U0IIR & 0x0e) == 0x04)          //清除中断
    {
        gc_uart0_receivedata = U0RBR;   //保存数据

        //实现对字符串的保存
        gc_uart0_receivebuf[gc_uart0_receivebufpointer] = gc_uart0_receivedata;
        gc_uart0_receivebufpointer++;
        if(gc_uart0_receivebufpointer >= UART0_BUFFERDATACOUNT)
        {
            gc_uart0_receivebufpointer = 0;
        }
    }
        VICVectAddr = 0x00;     //清除中断地址
}
```

```
/*****************************************************************************
 * Function Name:    uint8 UART0RecevieByte(void)                            *
 * Description   :   返回从串口接收到的数据                                  *
*****************************************************************************/
uint8 UART0RecevieByte(void)
{
    return gc_uart0_receivedata;
}

/*****************************************************************************
 * Function Name:    void UART0InterruptSet(void)                            *
 * Description   :   UAART0 中断设置，设置相应的中断类型和优先级，分配地址，  *
 *                   并使能                                                  *
 *****************************************************************************/
void UART0InterruptSet(void)
{
    VICIntSelect &= 0xffffffbf;     //中断类型设置，为 IRQ 类型
    VICVectCntl0 = 0x26;                //中断优先级设置
    VICVectAddr0 = (int)IRQ_UART0Receive;   //分配地址
    VICIntEnable = 0x00000040;      //使能该中断
}

/*****************************************************************************
 * Function Name:    void UART0Init(uint16 baud)                             *
 * Description   :   串口 0 初使化函数，设置波特率，帧格式，中断设置        *
*****************************************************************************/
void UART0Init(uint16 baud)
{
    uint32 temp;
    PINSEL0 = 0x05;
    temp = Fpclk / 16;
    temp = temp / baud;
    U0LCR = 0x80;
    U0DLL |= temp;
    U0DLM |= temp >> 8;
    U0LCR = 0x13;
    U0IER = 0x01;
    U0FCR = 0x01;
    UART0InterruptSet();
}
#endif
/*****************************************************************************
 *                              End of Entire File                           *
*****************************************************************************/
```

实验八　　I2C 存储程序

说明：
主文件需要包括以下文件：”config.h”,”Timer.h”,”dataled.h”,”I2C.h”,”string” ,”URAT.H”
主函数中：
I2CInit()I2C 初始化；URAT0Init(9600);串口波特率设置，一般为 9600 ，
Timer0_Init(0x00000010,0x000000ff);定时器设置 Timer0_Start();启动定时器
I2CWriteOneData(0xfe,0x08);　　向某单元写入一数据，后参数为地址，前为数据
I2CReadOneData(0x08);　　从某地址读出数据，I2CTest(void)用于一次向 256 个单元写数据，并从串口读出数据
如果需要在用数码管显示单一数据还需要在定时器中断中添加 LEDScan()

```
/*****************************************Copyright (c)*****************************/
/**
;**                                   西南科技大学计算机科学学院
;**                                     http://www.cs.suswt.edu.cn
;**
;**                日期：       2007.09.21
;**                描述：       西南科技大学计算机学院 CS-II 型实验板 I2C 工作程序头文件
;**                作者：
;**
;**-------------i2c.h 文件
;**-------------------------------------------------------------------------------------------*/
/**-------------------------------------------------------------------------------------------
*********************************************************************************/
 #ifndef _I2C_H_                          /* avoid being incude more than once */
 #define _I2C_H_
/*************************************************
 *                  Header file
 ************************************************/
 #include "config.h"

/*************************************************
 *                  Macros
 ************************************************/
 #define      DEVWRADD    0XA0                                    //设备写地址
 #define      DEVREADD    0XA1                                    //设备读地址

 #define      EEROM_WRITE   0                                     //写操作
 #define      EEROM_READ     1                                    //读操作

 #define     EEROM_OP_HUNG          0                            //操作未完成
 #define     EEROM_OP_FINISH       1                             //操作成功
```

```
#define     EEROM_OP_FAIL          2                              //操作出现错误
/************************************************
 *              Function declaration
 ************************************************/
extern void I2CInit(void);
extern void I2CTest(void);
extern void I2CWriteOneData(uint8 write_tmp,uint8 data_address);
extern void I2CReadOneData(uint8 data_address);

#endif
/********************************************************************************
 *                              End of Entire File                              *
 ********************************************************************************/
```

```
/*************************************Copyright (c)*****************************/
/**
;**                             西南科技大学计算机科学学院
;**                              http://www.cs.suswt.edu.cn
;**
;**              日期：    2007.09.21
;**              描述：    i2c.c  西南科技大学计算机学院 CS-II 型实验板工作程序，
;**              作者：
;**
;**-------------i2c.c 文件
;**------------------------------------------------------------------------------------*/
/**------------------------------------------------------------------------------------
*********************************************************************************
**********************/
#define _I2C_C_
#ifdef   _I2C_C_
/************************************************
 *              Header File
 ************************************************/
#include "config.h"
#include "i2c.h"
#include "string.h"                              //用于字符串写
#include "dataled.h"                             //用于数码管的显示
#include "URAT.h"                                //用于将 I2C 存储的数据用串口显示出来
/************************************************
 *              Globale variable
 ************************************************/
```

```
uint8 gc_eerom_writedata;                                   //I2C 写入存储的中间变量
uint8 gc_eerom_readdata;                                    //I2C 读出存储的中间变量

uint8 gc_eerom_byteaddress;                                 //读写地址的中间变量
uint8 gc_eerom_wrflag;                                      //读写标志
uint8 gc_byte_pionter = 0x00;
//字节指针，在本例中是单字节如果用到字符串写只需在中断中增大上限
volatile uint8 gc_eerom_resultflag;

/*****************************************************************************
 * Function Name: I2Cstart
 * Description   : 启动 I2C
 *****************************************************************************/
void I2CStart(void)
{
        //启动 I2C
       I2CONCLR = 0x2C;
       I2CONSET = 0X40;     //使能 I2C
       I2CONSET = 0X60;     //
}
/*****************************************************************************
 * Function Name:    __irq IRQ_I2C
 * Description   : I2C 中断函数
 *****************************************************************************/
void __irq IRQ_I2C(void)
{
//中断服务程序，由状态寄存器判断所处的中断并作相应的处理
uint8 i2cstatus;
i2cstatus = I2STAT;                                 //状态读取
if(gc_eerom_wrflag == EEROM_WRITE)                  //写数据
{

      switch(i2cstatus)
          {
            //Speak();
            case   0x08:

                I2DAT = 0XA0;                       //ATDevWRAdd;
                                                    //ATDevWRAdd=0xA0
                I2CONCLR = 0x28;                    // SI=0，清除中断标志//sla+w
                break;
            case   0x18:
                                                    //发送数据
                I2DAT = gc_eerom_byteaddress; //写字节地址
```

```
                I2CONCLR = 0X28;
                break;
            case    0x28:
                if(gc_byte_pionter < 0x01)
                {
                        I2DAT =gc_eerom_writedata;
                        gc_byte_pionter++;
                }
                else
                {
                    gc_byte_pionter=0x00;
                    I2CONSET = 0X10;
                    gc_eerom_resultflag = EEROM_OP_FINISH;
                }
                I2CONCLR = 0X28                 ;//完成，关闭 I2C
                break;
            default:                            //错误处理，返回错误标志，关闭 I2C
                 {

                    I2CONSET = 0X10;                        //停止标志
                    gc_eerom_resultflag = EEROM_OP_FAIL;    //操作失败
                    I2CONCLR = 0X68;//完成，关闭 I2C
                    break;
                 }
          }
  }

                                                            //读取数据
else if(gc_eerom_wrflag == EEROM_READ)
{
    switch(i2cstatus)
        {
            case   0x08:
                                                            // 已发送起始条件
                I2DAT = 0XA0;//ATDevREAdd;
                I2CONCLR = 0x28;                            // SI=0，清除中断标志
                break;
            case 0x10:
                                                            //重复开始条件
                I2DAT = DEVREADD;
                I2CONCLR = 0x28;                            // SI=0
                break;
            case   0x18:
                                                            //装数据字节
```

```
                I2DAT=gc_eerom_byteaddress;                    //写字节地址
                I2CONCLR=0X28;
                break;
            case    0x28:
                                                               //重新发送开始条件
                I2CONSET=0X20;                                 // 发送开始条件
                I2CONCLR=0X08;

                  break;
            case    0x40:

                  I2CONCLR = 0X3C;
                  break;
          case    0x58:

                  gc_eerom_readdata = I2DAT;          //已接受数据字节，返回 STOP
                  gc_eerom_resultflag = EEROM_OP_FINISH;

                  I2CONSET = 0X10;
                  I2CONCLR = 0x68;
                  break;
           default   :                                 //错误处理，返回错误标志，关闭 I2C
               {
                  I2CONSET = 0X10;                      /停止标志
                  gc_eerom_resultflag = EEROM_OP_FAIL;
                  I2CONCLR = 0X68;                    //完成，关闭 I2C
                  break;
               }
        }
  }
LEDDelayNS(10);                                          //读取需要一定的时间
VICVectAddr = 0x00;                                      //清除地址

}

/*****************************************************************************
 * Function Name:      I2CInterruptSet
 * Description   : 中断设置
 *****************************************************************************/
void I2CInterruptSet(void)
{

    VICIntSelect &= 0xffffdff;                           //中断类型分配
    VICVectCntl4 = 0x29;                                 // I2C 通道分配到 IRQ slot 3，
```

```
    VICVectAddr4 = (int)IRQ_I2C;                    // 设置 I2C 中断向量地址
    VICIntEnable = 0x0200;                          // 使能 I2C 中断

}
/*****************************************************************************
 * Function Name:       I2CInit
 * Description    :       I2C 始化
 *****************************************************************************/
void I2CInit(void)
{
    PINSEL0|=0X50;                                  //端口设置

    I2SCLH = 0X3A;
    I2SCLL = 0X3A;                                  //100K

    I2CInterruptSet();                              //中断设置

}

/*****************************************************************************
 * Function Name:    I2CWrite
 * Description    :  向 I2C  写数据
 *****************************************************************************/
void I2CWriteByty(uint8 write_tmp,uint8 data_address)
{
    gc_eerom_byteaddress = data_address;
    gc_eerom_writedata = write_tmp;                  //数据装入

                                                    //标志部分
    gc_eerom_wrflag=EEROM_WRITE;                    //写标志
    gc_eerom_resultflag = EEROM_OP_HUNG;

    I2CStart();

                                                    //等待操作完成
    while(gc_eerom_resultflag == EEROM_OP_HUNG)    //等待写成功
    {

//由 volatile 修饰全局变量 ， 这也必须加上一段延迟程序。
    }
}
```

```
/*****************************************************************************
 * Function Name:    I2CReadByty
 * Description   :   向 I2C 写一个字节
 *****************************************************************************/
void I2CReadByty(uint8 data_address)
{
                                                    //地址计算部分
    gc_eerom_byteaddress = data_address;
    gc_eerom_wrflag = EEROM_READ;                   //读标志
    gc_eerom_resultflag = EEROM_OP_HUNG;

    I2CStart();

                                                    //等待操作完成
  while(gc_eerom_resultflag == EEROM_OP_HUNG)       //等待写成功
     {
  //由 volatile 修饰全局变量 ， 这也必须加上一段延迟程序。
     }
}


/*****************************************************************************
 * Function Name:    I2CWriteOneData
 * Description   :   读取一个字节
 *****************************************************************************/
void I2CWriteOneData(uint8 write_tmp,uint8 data_address)
{
     I2CWriteByty(write_tmp,data_address);
}

/*****************************************************************************
 * Function Name:     I2CReadOneData
 * Description   :   从 I2C 读一个字节
 *****************************************************************************/
void I2CReadOneData(uint8 data_address)
{
       I2CReadByty(data_address);
       LEDPrint(gc_eerom_readdata);

}
```

```
/*******************************************************************************
 * Function Name:    I2CTest
 * Description    :  想 I2C 所有单元写数据，并读出，通过串口显示出来
*******************************************************************************/
 void I2CTest(void)
 {

     uint8 address;
   for(address=0;address<0xff;address++)
   {
         I2CWriteByty(0xff-address,address);
         LEDDelayNS(10);
   }
   for(address=0;address<0xff;address++)
   {
           I2CReadByty(address);
           URAT0SendByte(gc_eerom_readdata);
   }
 }

#endif
/*******************************************************************************
 *                                  End of Entire File
 *******************************************************************************/
```

实验九　电机实验

说明：在 main 文件中要包含 dataled.h, key.h, timer.h, motor.h 。dataled 模块使用前面实验中的用到的。Key 模块与前面 keyboard 模块稍有不同，但实现功能基本一样，可以作为另外一种实现方法进行参考；另外 KEYInit()加入到 main 文件初始化函数中，KEYScan()放到定时器中断函数中。Timer 模块也是使用的前面模块中的。可根据前面的讲解进行设置。

Motor 模块包含步进电机和直流电机的设置和调用。 MOTORInit()包含了直流和步进电机的初始化，要使用两种电机首先将将该函数包含到 main 文件设备初始化函数中。使用很简单，在 main 函数或者中断处理函数中调用 void MOTORDCRun(uint8 speed)就可以让直流电机运动。Speed 是直流电机的运动速率，0x0f 最快。该函数只需要执行一次即可保持直流电机运行。在 while 或中断处理函数中调用 void MOTORStepRun(uint8 speed,uint8 direction)函数可以运动步进电机。Speed 是速率，0x00 最快，direction 表示方向，有正向和逆向两种。该函数没运行一次前进一定的步进角，因此需要不停的调用它才能实现转动。这里放在中断处理函数中实现。操作方式见下描述。

```
/***********************************Copyright (c)*******************************/
;**                              西南科技大学计算机科学学院
;**                                http://www.cs.suswt.edu.cn
;**
;**            日期：     2007/09/20
;**            描述：     控制西南科技大学计算机学院 CS-II 型实验板电机运行的程序头文
;**                       件。
;**                       程序可分别控制直流和步进电机的转动。其中，直流电机转动可进
;**                       行 16 个速度等级的改变；步进电机可实现与键盘的互通，s0 是步进
;**                       电机启动键，s8 为停止键；s1 为速度调制键，按下后，s3-s7，s11-s15
;**                       代表速度 0-10（其中 0 最快），s9 停止速度调制，而其他键无效。
;**                       s2 让步进电机顺时针转动，s10 让电机逆时针转动。使用了定时器
;**                       1 和定时器 0 同时工作。
;**
;**             作者：
;**
;**------------motor.h 文件
;**-------------------------------------------------------------------------------------*/
/**-------------------------------------------------------------------------------------
*************************************************************************************/
#ifndef _MOTOR_H_                       /* avoid being include more than once */
#define _MOTOR_H_
/*************************************************
 *                  Header file                  *
 *************************************************/
 #include"config.h"
```

```
/*********************************************
 *                  Macros                   *
 *********************************************/
 #define DCMOTOR_SWITCH    0x00800000
 #define MOTOR_PINSEL      0xfffffc0                    //选择 P2.19-23
 #define DCMOTOR_SPEED_GRADE 0x10
 #define STEP_PHASE_A      0x00400000
 #define STEP_PHASE_B      0x00200000
 #define STEP_PHASE_C      0x00100000
 #define STEP_PHASE_D      0x00080000
 #define CLOCKWISE         1
 #define ANTICLOCKWISE     0
 #define MOTOR_DC_ENABLE
 #define MOTOR_STEP_ENABLE
 /*********************************************
 *              Function declaration          *
 *********************************************/
 extern void MOTORDCRun(uint8 speed);                  //直流电机驱动函数
 extern void MOTORInit(void);                          //电机初始化函数
 extern void MOTORStepRun(uint8 speed,uint8 direction);  //步进电机驱动函数

#endif
/*****************************************************************************
 *                          End of Entire File                                *
 *****************************************************************************/



/***********************************Copyright (c)*********************************/
;**                          西南科技大学计算机科学学院
;**                            http://www.cs.suswt.edu.cn
;**
;**          日期:    2007/09/20
;**          描述:    西南科技大学计算机学院 CS-II 型实验板直流电机和步进电机驱动
;**                   工作程序
;**          作者:
;**
;**------------motor.c 文件
;**-------------------------------------------------------------------------------------*/
/**-------------------------------------------------------------------------------------
*****************************************************************************/
#define _MOTOR_C_
#ifdef   _MOTOR_C_
```

```
/*********************************************
 *                Header File                *
 *********************************************/
 #include"motor.h"
 #include"config.h"

/*****************************************************************************
 * Function Name: void MOTORInit(void)                                       *
 * Description   : 步进电机、直流电机参数初始化程序                          *
 *****************************************************************************/
 void MOTORInit(void)
 {
    PINSEL2 &= MOTOR_PINSEL;
    //引脚设置，这里是选择 P2.19-23，包括步进电机和直流电机引脚，设置为 GPIO
    IO2DIR |= DCMOTOR_SWITCH;                    //直流电机初始化
    IO2SET = DCMOTOR_SWITCH;
    IO2DIR |= STEP_PHASE_A;                      //步进电机初始化
    IO2DIR |= STEP_PHASE_B;
    IO2DIR |= STEP_PHASE_C;
    IO2DIR |= STEP_PHASE_D;
    IO2SET = STEP_PHASE_A;
    IO2SET = STEP_PHASE_B;
    IO2SET = STEP_PHASE_C;
    IO2SET = STEP_PHASE_D;
 }
 /**-------------------------------------------------------**/
 //                    直流电机
 /**-------------------------------------------------------**/

#ifdef MOTOR_DC_ENABLE
uint8 uc_dcmotor_delay = 0;                      //让直流电机延时转动
/*****************************************************************************
 * Function Name: void DCMOTORRun(uint8 speed)                               *
 * Description   : 直流电机驱动函数，其中入口参数 speed 代表直流电机的转动速度，  *
 *                 这里将其速度等级设置 16 个，即 0x00～0x0f，0x0f 最快。      *
 *****************************************************************************/
 void MOTORDCRun(uint8 speed)
 {
    if(uc_dcmotor_delay < speed)
   //speed 是直流电机的转动速度，含义相当于在一定的时间周期内有多少时间让电机转动
    {
        IO2CLR = DCMOTOR_SWITCH;                 //驱动直流电机转动
    }
    else if(uc_dcmotor_delay > speed)
```

```
    {
        IO2SET = DCMOTOR_SWITCH;                    //停止直流电机转动
    }
    uc_dcmotor_delay++;
    if(uc_dcmotor_delay == DCMOTOR_SPEED_GRADE)
    {
        uc_dcmotor_delay = 0;
    }
}
#endif


/**-------------------------------------------------------------**/
//                      步进电机
/**-------------------------------------------------------------**/
#ifdef MOTOR_STEP_ENABLE

uint8 uc_stepmotor_delay = 0;                       //让步进延时
uint8 uc_phasenumber = 0;                           //控制相位

/*****************************************************************************
 * Function Name: void MOTORStepRun(uint8 speed)                             *
 * Description   : 步进电机驱动程序。入口参数 speed 为步进电机的速度控制参数，速    *
 *                 度级别 0x00-0x0f,其中 0x00 最快，当速度级别为 0x00 时转一周约需  *
 *                   8.19s（放在 1ms 中断一次的定时器 0 中）相应的如果速度级别增加 1*
 *                   转一圈的时间增加一倍                                        *
 *****************************************************************************/
void MOTORStepRun(uint8 speed,uint8 direction)
{
    if(uc_stepmotor_delay == speed)
    {
        if(direction == CLOCKWISE)                  //如果设置方向为顺时针方向
        {
            switch(uc_phasenumber++)                //相位转动顺序为 A->B->C->D
            {
                case 0:   IO2CLR = STEP_PHASE_A;
                          IO2SET = STEP_PHASE_B;
                          IO2SET = STEP_PHASE_C;
                          IO2SET = STEP_PHASE_D;
                          break;

                case 1:   IO2CLR = STEP_PHASE_B;
                          IO2SET = STEP_PHASE_A;
                          IO2SET = STEP_PHASE_C;
```

```
                IO2SET = STEP_PHASE_D;
                break;

        case 2:  IO2CLR = STEP_PHASE_C;
                IO2SET = STEP_PHASE_A;
                IO2SET = STEP_PHASE_B;
                IO2SET = STEP_PHASE_D;
                break;

        case 3:  IO2CLR = STEP_PHASE_D;
                IO2SET = STEP_PHASE_A;
                IO2SET = STEP_PHASE_C;
                IO2SET = STEP_PHASE_B;
                break;

        default: break;
    }
}
else                                        //如果设置方向为顺时针方向
{
    switch(uc_phasenumber++)                //相位转动顺序为 A->B->C->D
    {
        case 0:  IO2CLR = STEP_PHASE_A;
                IO2SET = STEP_PHASE_B;
                IO2SET = STEP_PHASE_C;
                IO2SET = STEP_PHASE_D;
                break;

        case 1:  IO2CLR = STEP_PHASE_D;
                IO2SET = STEP_PHASE_A;
                IO2SET = STEP_PHASE_C;
                IO2SET = STEP_PHASE_B;
                break;

        case 2:  IO2CLR = STEP_PHASE_C;
                IO2SET = STEP_PHASE_A;
                IO2SET = STEP_PHASE_B;
                IO2SET = STEP_PHASE_D;
                break;

        case 3:  IO2CLR = STEP_PHASE_B;
                IO2SET = STEP_PHASE_A;
                IO2SET = STEP_PHASE_C;
                IO2SET = STEP_PHASE_D;
```

```
                                break;

                        default: break;
                    }
                }
                if(uc_phasenumber == 4)                         //相位控制
                {
                    uc_phasenumber = 0;
                }
                uc_stepmotor_delay = 0;
            }
            else
            {
                uc_stepmotor_delay++;
            }

        }
        #endif

#endif
/*******************************************************************************
 *                                End of Entire File                           *
 *******************************************************************************/



/************************************Copyright (c)*********************************/
;**                                西南科技大学计算机科学学院
;**                                  http://www.cs.suswt.edu.cn
;**
;**            日期：      2007/09/20
;**            描述：      西南科技大学计算机学院 CS-II 型实验板键盘工作程序头文件
;**            作者：
;**
;**-------------key.h 文件
;**-------------------------------------------------------------------------------------*/
/**--------------------------------------------------------------------------------------
*******************************************************************************/
#ifndef _KEY_H_                         /* avoid being incude more than once */
#define _KEY_H_
/*************************************************
 *                  Header file                  *
 *************************************************/
 #include"config.h"
```

```
/*********************************************
 *                  Macros                   *
 *********************************************/
 #define   KEY_PINSEL     0xffff00ff
 #define   KEY_SRCLK     0X00000010
 #define   KEY_SI          0x00000040
 #define   KEY_RCLK      0x00000080
 #define   KEY_KEY       0x00000020
 /*********************************************
 *              Function declaration           *
 *********************************************/
 void KEYShift(uint8 mode);                    //数据移位函数
 uint8 KEYScan(void);                          //键盘轮转扫描函数
 void KEYInit(void);                           //键盘初始化设置
 #endif

/*****************************************************************************
 *                          End of Entire File                               *
 *****************************************************************************/




/***********************************Copyright (c)*********************************/
;**                          西南科技大学计算机科学学院
;**                            http://www.cs.suswt.edu.cn
;**
;**          日期:     2007/09/20
;**          描述:     西南科技大学计算机学院 CS-II 型实验板键盘工作程序
;**          作者:
;**
;**------------motor.c 文件
;**-------------------------------------------------------------------------------*/
/**--------------------------------------------------------------------------------
*******************************************************************************/
#define _KEY_C_
#ifdef   _KEY_C_
/*********************************************
 *              Header File                  *
 *********************************************/
 #include"config.h"
 #include"key.h"
```

```
/*********************************************
 *              Globale variable              *
 *********************************************/
 uint8 key_data=0x0f;//键盘初始状态
/*****************************************************************************
Function Name:     void KEYShift(uint8 mode)                                   *
Function Description :   进行轮转扫描时进行数据移位操作                        *
*****************************************************************************/
void KEYShift(uint8 mode)                          //如果 mode 是 0 就将 KEY_SI 置零  如果
mode 是 1 就将 KEY_SI 置 1
{

      IO0CLR=KEY_RCLK;                             //RCLK=0x00

      IO0CLR=KEY_SRCLK;                            //SRCLK=0
         if((mode&0x01)==0x00)                     //该位为 0
              {
                   IO0CLR = KEY_SI;
              }
         else
              {
                   IO0SET = KEY_SI;
              }
     IO0SET=KEY_SRCLK;                             //SLK=1

     IO0SET=KEY_RCLK;                               //RCK=0x00
}

/*****************************************************************************
Function Name:       uint8 KEYScan(void)                                       *
Function Description :   键盘扫描函数，进行轮转扫描                           *
*****************************************************************************/
uint8 tmp = 0x00;
uint8 KEYScan(void)
{
 //KeyData 变化范围为 0x00----0x0f
 //扫描过程
 //先进行键盘判断上次键盘的判断

  if((IO0PIN&KEY_KEY)==0)      //有键按下（IO0PIN&KEY_KEY）== 0，即 key 位为 0
    {
       tmp = key_data;
       return key_data;
    }                                 //如果有键盘按下，连续对该键进行扫描，返回该键键值。
```

```
   else
      {
          if(key_data==0x0f)
            {
              key_data=0x00;
              KEYShift(0);
            }
          else
            {
              key_data++;
              KEYShift(1);
            }
        return tmp;//0x10;
      }                                  //没有键盘按下，进行扫描轮转。
}

/*****************************************************************************
Function Name          : void KEYInit(void)                                    *
Function Description : 键盘的初始化程序，包括设置引脚链接模块，引脚输入输出设置    *
*****************************************************************************/
void KEYInit(void)
{
      PINSEL0&=KEY_PINSEL;        //端口选择，这里选择 P0.4~P0.7 实现 GPIO 功能
      IO0DIR|=KEY_SRCLK;          //方向输出
      IO0DIR|=KEY_RCLK;           //方向输出
      IO0DIR|=KEY_SI;                                  //方向输出
      IO0DIR&=(KEY_KEY^0XFFFFFFFF);                    //方向输入
}

#endif
/*****************************************************************************
 *                                End of Entire File                          *
 *****************************************************************************/
```

实验十 LCD 液晶显示模块

说明：在主函数(main)中，我们可以通过 LCDPrintString(uint8 linetemp,char displaystring[])和 LCDPrintImage(uint8 imagenumber)来实现字符串和图片的显示，当然在这之前应该初使化 LCD（调用 LCDInit（void）来实现）。

```
/********************************Copyright(c) *********************************/
/**
;**                              西南科技大学计算机科学学院
;**                               http://www.cs.suswt.edu.cn
;**
;**              日期：     2007.9.20
;**              描述：     LCDDriver.h  西南科技大学计算机学院 CS-II 型实验板 LCD
;**                          工作程序，实现对 LCD 控制的各种接口的定义。
;**              作者：
;**
;**------------LCDDriver.h 文件
;**-------------------------------------------------------------------------------------*/

/**----------------------------------------------------------------------------------------
*****************************************************************************/
#ifndef   _LCDDRIVER_H_
#define   _LCDDRIVER_H_
/***********************************************
  *             Header File                    *
  **********************************************/
#include "config.h"
/***********************************************
  *                Macros                        *
  **********************************************/
#define   LCD_ENABLE
/*address phase 82080000---820bffff*/
#define   LCD_CS1_BASE             0x82080000
#define   LCD_CS2_BASE             0x820c0000   //base address
#define   LCD_DISPLAY_ENABLE       0x00020000
#define   LCD_BACK_LED             0x00040000
#define   READ_DATA                0x00000003
#define   READ_INSTRUCT            0x00000002
#define   WRITE_DATA               0x00000001
#define   WRITE_INSTRUCT           0x00000000
#define   LCD_PAGE_LARGE           8
#define   LCD_LINE_LARGE           128
```

```
/*cs1 data register, read only*/
#define    LCD_CS1_READ_DATA
  (*((volatile uint8 *) (LCD_CS1_BASE+READ_DATA)))

/*cs1 instruct register, read only*/
#define    LCD_CS1_READ_INSTRUCT
(*((volatile uint8 *) (LCD_CS1_BASE+READ_INSTRUCT)))

/*cs1 data register, write only*/
  #define    LCD_CS1_WRITE_DATA
                (*((volatile uint8 *) (LCD_CS1_BASE+WRITE_DATA)))

/*cs1 instruct register, write only*/
    #define    LCD_CS1_WRITE_INSTRUCT
  (*((volatile uint8 *) (LCD_CS1_BASE+WRITE_INSTRUCT)))

/*cs2 data register, read only*/
#define    LCD_CS2_READ_DATA
(*((volatile uint8 *) (LCD_CS2_BASE+READ_DATA)))

/*cs2 instruct register, read only*/
  #define    LCD_CS2_READ_INSTRUCT
  (*((volatile uint8 *) (LCD_CS2_BASE+READ_INSTRUCT)))

/*cs2 data register, write only*/
#define    LCD_CS2_WRITE_DATA
  (*((volatile uint8 *) (LCD_CS2_BASE+WRITE_DATA)))

/*cs2 instruct register, write only*/
  #define    LCD_CS2_WRITE_INSTRUCT
  (*((volatile uint8 *) (LCD_CS2_BASE+WRITE_INSTRUCT)))

/************************************************
  *                  Function declaration                    *
  ************************************************/
extern void LCDDisplayLcdMemery(void);
extern void LCDInit(void);
extern void LCDPrintString(uint8 linetemp,char displaystring[]);
extern void LCDPrintImage(uint8 imagenumber);

#endif
/********************************************************************
  *                                    End of Entire File                                    *
  ********************************************************************/
```

```
/*********************************Copyright(c)*********************************/
/**
;**                              西南科技大学计算机科学学院
;**                                http://www.cs.suswt.edu.cn
;**
;**                日期：     2006.4.1
;**                描述：     LCDDriver.c 西南科技大学计算机学院 CS-II 型实验板 LCD
;**                           工作程序，实现 LCD 各种函数的定义，完成相应的 LCD
;**                           的功能
;**                作者：
;**
;**------------LCDDriver.c 文件
;**-------------------------------------------------------------------------------------------*/
/**-------------------------------------------------------------------------------
*******************************************************************************/
//LCDDriver.c
//注本程序中: 行为 Line,列为 Row
#define _LCDDriver_C_
#ifdef _LCDDriver_C_
/*************************************************
 *              Header File                      *
 *************************************************/
#include "config.h"
#include "LCDDriver.h"
#include "LCDFontData.h"
#include "string.h"


/*************************************************
 *              Globale variable                 *
 *************************************************/
uint8 gc_lcddisplaymemery[LCD_PAGE_LARGE][LCD_LINE_LARGE];

/*****************************************************************
函数名:void LCDInit()
功能:   LCD 初使化,设置外部数据线为 8 位,设置相应的控制线为输出
*****************************************************************/
void LCDInit()
{
    BCFG2 &= 0xcfffffff;              //定义数据线为 8 位
    IO2DIR |= LCD_DISPLAY_ENABLE;
    IO2DIR |= LCD_BACK_LED;
    IO2SET |= LCD_BACK_LED;
}
```

```
/*****************************************************************
函数名:void LCDEnable()
功能:   LCD 控制驱动器读写使能
        (注:LCDDisplayEnable (E)为 LCD 驱动器读写使能信号,当它的下降没时,数据被锁
存在驱动器中,当它在高电平时,信号可以被读出.)
*****************************************************************/
void LCDEnable()
{
    IO2SET = LCD_DISPLAY_ENABLE;
}

/*****************************************************************
函数名:void LCDDisable()
功能:   关闭 LCD 控制驱动器读写使能
        (注:LCDDisplayEnable (E)为 LCD 驱动器读写使能信号,当它的下降没时,数据被锁
存在驱动器中,当它在高电平时,信号可以被读出.)
*****************************************************************/
void LCDDisable()
{
    IO2CLR = LCD_DISPLAY_ENABLE;
}

/*****************************************************************
函数名:void LCDBusyWaite(uint8 chip)
功能:   查询 LCD 是否为空闲状态,如果不是一至等待,直到其为空闲状态.
*****************************************************************/
void LCDBusyWaite(uint8 chip)
{
    uint8 lcdstate = 0x80;
    if(chip == 0)
    {
        /*如果 LCD 不为空闲状态(内部在工作),通过查询方式一直查询*/
        while((lcdstate & 0x80) != 0)
        {
            /*LCD 控制驱动器读写使能为 1,即(E)处于高电平,此时可以读出信息*/
            LCDEnable();

            LCD_CS1_READ_INSTRUCT = 0xff;
            lcdstate = LCD_CS1_READ_INSTRUCT;          //读取信息
            LCDDisable();
        }
    }

    if(chip == 1)
```

```
    {
        while((lcdstate & 0x80) != 0)            //同上
        {
            LCDEnable();
            LCD_CS2_READ_INSTRUCT = 0xff;

            lcdstate = LCD_CS2_READ_INSTRUCT;
            LCDDisable();
        }
    }
}

/*****************************************************************
函数名:vois LCDWriteInstruction(uint8 instruction,uint8 chip)
功能:  将 LCD 指令写入到 LCD 控制驱动器中去.
*****************************************************************/
void LCDWriteInstruction(uint8 instruction, uint8 chip)
{
    LCDBusyWaite(chip);

    if(chip == 0)
    {
        LCD_CS1_WRITE_INSTRUCT = instruction;
    }
    if(chip == 1)
    {
        LCD_CS2_WRITE_INSTRUCT = instruction;
    }

    LCDEnable();
    LCDDisable();
}

/*****************************************************************
函数名:vois LCDWriteData(uint8 disdata,uint8 chip)
功能:  将相应的数据写入到 LCD 内部存储区(RAM)中去.LCD 内部存储区地址自动加 1
*****************************************************************/
void LCDWriteData(uint8 disdata, uint8 chip)
{
    LCDBusyWaite(chip);         //等待 LCD 空闲

    if(chip == 0)
    {
        LCD_CS1_WRITE_DATA = disdata;        //将数据送到相应的数据线
```

```
    }
    if(chip == 1)
    {
        LCD_CS2_WRITE_DATA = disdata;
    }


    LCDEnable();
    LCDDisable();                //(E)下降沿(写入时钟),将数据写入到 LCD 中的 RAM
}

/*************************************************************
函数名:vois LCDDisplaySwitchInstruction(uint8 lcdswitch,uint8 chip)
功能:   LCD 显示开关指令,当 lcdswitch 为 1 时,LCD 显示其 RAM 中的内容,为 0 时,关闭显示.
*************************************************************/
void LCDDisplaySwitchInstruction(uint8 lcdswitch, uint8 chip)
{
    uint8 temp = 0x3e;      //开关指令格式(具体参考 LCD 技术手册 P16)

    temp = temp | lcdswitch;
    LCDWriteInstruction(temp, chip);
}

/*************************************************************
函数名:void LCDSetFirstLineInstruction(uint8 lcdrow, uint8 chip)
功能:   LCD 设置显示起始行指令,设置其起始行为 lcdrow 行..
*************************************************************/
void LCDSetFirstLineInstruction(uint8 lcdrow, uint8 chip)   //行为 Line,列为 Row
{
    uint8 temp = 0xc0;      ////开关指令格式(具体参考 LCD 技术手册 P16)

    temp = temp | lcdrow;
    LCDWriteInstruction(temp,chip);
}


/*************************************************************
函数名:void LCDSetPageInstruction(uint8 lcdpage, uint8 chip)
功能:   LCD 设置页指令,此 LCD 共有 64 行,每 8 行分为一页,故共有 8 页,要定位 LCD 中的具
        体位置,需要定位到相应的页和相应的列
*************************************************************/
void LCDSetPageInstruction(uint8 lcdpage, uint8 chip)
{
    uint8 temp = 0xb8; //开关指令格式(具体参考 LCD 技术手册 P16)
```

```
    temp = temp | lcdpage;
    LCDWriteInstruction(temp, chip);
}

/*****************************************************************
函数名:void LCDSetRowAddressInstruction(uint8 lcdrow, uint8 chip)
功能:   LCD 设置列地址指令,些 LCD 共有 128 列
*****************************************************************/
void LCDSetRowAddressInstruction(uint8 lcdrow, uint8 chip)
{
    uint8 temp = 0x40; //开关指令格式(具体参考 LCD 技术手册 P16)

    temp = temp | lcdrow;
    LCDWriteInstruction(temp, chip);
}

/*****************************************************************
函数名:void LCDDisplayLcdMemery()
功能:   将缓冲区中的数据信息写入到 LCD 内部的 RAM 中去,以实现显示相应的信息的功能
          (注:此 LCD 中共有 2 * ( 64 * 64)个 bit,每一个 bit 对应着 LCD 中的相应点,当相应
          的 bit 为 1 时,LCD 中此点被点亮,为 0 时熄灭,故我们的实现思路为在 ARM 存储区
          中构建一个相同大小的存储区(128*64),然后把它一列列的
          写入到 LCD 内部 RAM 中去.)
*****************************************************************/
void LCDDisplayLcdMemery()
{
    uint8 rowtemp;     //向 LCD 内部的 RAM 写入数据一定要知道它的地址,即它是哪页哪列
    uint8 pagetemp;    //这样才能从 ARM 中将相应的数据写入到 LCD 内部 RAM 中

    for(pagetemp = 0; pagetemp < 8; pagetemp++)
    {
        for(rowtemp = 0; rowtemp < 64; rowtemp++)
        {
            LCDSetPageInstruction(7-pagetemp,1);       //定位页
            LCDSetRowAddressInstruction(63-rowtemp,1);  //定位列

            /*向此页此列写入相应的数据(片 1)*/
            LCDWriteData(gc_lcddisplaymemery[pagetemp][rowtemp],1);
            LCDSetPageInstruction(7-pagetemp,0);
            LCDSetRowAddressInstruction(63-rowtemp,0);

            /*向此页此列写入相应的数据(片 0)*/
```

```
                LCDWriteData(gc_lcddisplaymemery[pagetemp][rowtemp+64],0);
            }
        }


        LCDSetFirstLineInstruction(0,1); //设置片 1 的起始行为第 0 行
        LCDDisplaySwitchInstruction(1,1);    //显示第 1 片的数据

        LCDSetFirstLineInstruction(0,0); //设置片 0 的起始行为第 0 行
        LCDDisplaySwitchInstruction(1,0);    //显示第 0 片的数据
}

/*****************************************************************
函数名:void LCDPrintString(uint8 linetemp,char displaystring[])
功能:   在 linetemp 行中显示相应的字符串
*****************************************************************/
void LCDPrintString(uint8 linetemp,char displaystring[])
{
        uint8 linefontpointer,linefontbitpointer;
        uint8 templinesize;
        uint8 displayASCII,strlength;

        strlength = strlen(displaystring);

        /*如果欲显示的字符串大于 16(每行字符串的极限)则丢弃后面的字符串*/
        if(strlength > LCD_LINE_FONT_LARGE)
        {
            strlength = LCD_LINE_FONT_LARGE;
        }

        /*并要显示的字符串送到相应地址的缓冲区中去*/
        for(linefontpointer = 0; linefontpointer < strlength; linefontpointer++)
        {
            displayASCII = (uint8)displaystring[linefontpointer];

            /*字符串 ASCII 转换成相应的字库参数信息*/
            displayASCII = displayASCII - LCD_FONT_START_ASC;
            templinesize = linefontpointer * LCD_FONT_WIDTH;
            for(linefontbitpointer = 0; linefontbitpointer < LCD_FONT_WIDTH;
linefontbitpointer++)
            {
                gc_lcddisplaymemery[linetemp][templinesize+linefontbitpointer] =
gc_stringfontcode[displayASCII][linefontbitpointer];
            }
```

```
    }

    LCDDisplayLcdMemery();          //显示缓冲区中的数据.
}
/*************************************************************
函数名:void LCDPrintImage(uint8 imagenumber)
功能:   显示序号为 imagenumber 的图片
*************************************************************/
void LCDPrintImage(uint8 imagenumber)
{
    uint8 pagetemp, rowtemp;
    for(pagetemp = 0; pagetemp < LCD_PAGE_LARGE; pagetemp++)
    {
        for(rowtemp = 0; rowtemp < LCD_LINE_LARGE; rowtemp++)
        {
            gc_lcddisplaymemery[pagetemp][rowtemp] =
gc_lcdimagecode[imagenumber][pagetemp][rowtemp];
        }
    }

    LCDDisplayLcdMemery();
}

#endif
/*****************************************************************************
 *                                End of Entire File                             *
 *****************************************************************************/


/*********************************Copyright(c)*********************************/
/**
;**                          西南科技大学计算机科学学院
;**                            http://www.cs.suswt.edu.cn
;**
;**              日期:     2006.4.1
;**              描述:     LCDFontData.h  西南科技大学计算机学院 CS-II 型实验板 LCD
;**                        工作程序，控制字库与自定义图片的接口
;**              作者:
;**
;**-------------LCDFontData.h 文件
;**-------------------------------------------------------------------------------------*/
/**-------------------------------------------------------------------------------
*************************************************************************/
```

```
#ifndef _LCDFONTDATA_H_
#define _LCDFONTDATA_H_
/*********************************************
 *              Header File                  *
 *********************************************/
#include "config.h"
#include "LCDDriver.h"
/*********************************************
 *                Macros                     *
 *********************************************/
#define LCD_FONT_WIDTH        8      //8 个字节构成一个完整字体
#define LCD_FONT_LARGE        95     //字库个数
#define LCD_FONT_START_ASC    32     //其始字符 ASC 码
#define LCD_LINE_FONT_LARGE 16/    /一行显示的字符数
extern uint8 const gc_stringfontcode[LCD_FONT_LARGE][LCD_FONT_WIDTH];
extern uint8 const gc_lcdimagecode[][LCD_PAGE_LARGE][LCD_LINE_LARGE];

#endif
/*****************************************************************************
 *                              End of Entire File                           *
 *****************************************************************************/

/*****************************Copyright(c) ***********************************/
/**
;**                                  西南科技大学计算机科学学院
;**
;**                                 http://www.cs.suswt.edu.cn
;**
;**              日期:     2006.4.1
;**              描述:     LCDFontData.c 西南科技大学计算机学院 CS-II 型实验板 LCD
;**                        工作程序，实现字库和自定义图片的定义。
;**              作者:
;**
;**-------------LCDFontData.c 文件
;**-------------------------------------------------------------------------------------*/

/**-------------------------------------------------------------------------------------
*****************************************************************************
************************/
#define _LCDFONTDATA_C_
#ifdef   _LCDFONTDATA_C_
/*********************************************
 *              Header File                  *
 *********************************************/
```

```
#include "config.h"
#include "LCDFontData.h"
#include "LCDDriver.h"

/************************************************
 *              Globale variable                *
 ************************************************/
uint8 const  gc_stringfontcode[LCD_FONT_LARGE][LCD_FONT_WIDTH]={
/*--  文字:  空格  --*/
/*--  宋体 12;  此字体下对应的点阵为：宽 x 高=8x8   --*/
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
/*--  文字:  !  --*/
/*--  宋体 12;  此字体下对应的点阵为：宽 x 高=8x8   --*/
0x00,0x00,0x00,0xFD,0x00,0x00,0x00,0x00,
/*--  文字:  "  --*/
/*--  宋体 12;  此字体下对应的点阵为：宽 x 高=8x8   --*/
0x00,0x08,0x30,0x60,0x08,0x30,0x60,0x00,
/*--  文字:  #  --*/
/*--  宋体 12;  此字体下对应的点阵为：宽 x 高=8x8   --*/
0x24,0x3F,0xE4,0x24,0x3F,0xE4,0x24,0x00,
/*--  文字:  $  --*/
/*--  宋体 12;  此字体下对应的点阵为：宽 x 高=8x8   --*/
0x00,0x24,0x52,0xFF,0x52,0x4C,0x00,0x00,
/*--  文字:  %  --*/
/*--  宋体 12;  此字体下对应的点阵为：宽 x 高=8x8   --*/
0x01,0x02,0x64,0xA8,0xD3,0x25,0x46,0x80,
/*--  文字:  &  --*/
/*--  宋体 12;  此字体下对应的点阵为：宽 x 高=8x8   --*/
0x02,0x75,0x89,0x95,0x6A,0x0D,0x09,0x02,
/*--  文字:  '  --*/
/*--  宋体 12;  此字体下对应的点阵为：宽 x 高=8x8   --*/
0x08,0x68,0x70,0x00,0x00,0x00,0x00,0x00,
/*--  文字:  (  --*/
/*--  宋体 12;  此字体下对应的点阵为：宽 x 高=8x8   --*/
0x00,0x00,0x00,0x00,0x3C,0x42,0x81,0x00,
/*--  文字:  )  --*/
/*--  宋体 12;  此字体下对应的点阵为：宽 x 高=8x8   --*/
0x00,0x81,0x42,0x3C,0x00,0x00,0x00,0x00,
/*--  文字:  *  --*/
/*--  宋体 12;  此字体下对应的点阵为：宽 x 高=8x8   --*/
0x24,0x24,0x18,0xFF,0x18,0x24,0x24,0x00,
/*--  文字:  +  --*/
/*--  宋体 12;  此字体下对应的点阵为：宽 x 高=8x8   --*/
0x18,0x18,0x18,0xFF,0xFF,0x18,0x18,0x18,
```

```
/*--  文字:  '  --*/
/*--  宋体 12;  此字体下对应的点阵为：宽 x 高=8x8  --*/
0x08,0x68,0x70,0x00,0x00,0x00,0x00,0x00,
/*--  文字:  -  --*/
/*--  宋体 12;  此字体下对应的点阵为：宽 x 高=8x8  --*/
0x18,0x18,0x18,0x18,0x18,0x18,0x18,0x18,
/*--  文字:  .  --*/
/*--  宋体 12;  此字体下对应的点阵为：宽 x 高=8x8  --*/
0x00,0x00,0x00,0x06,0x06,0x00,0x00,0x00,
/*--  文字:  /  --*/
/*--  宋体 12;  此字体下对应的点阵为：宽 x 高=8x8  --*/
0x01,0x02,0x04,0x08,0x10,0x20,0x40,0x80,
/*--  文字:  0  --*/
/*--  宋体 12;  此字体下对应的点阵为：宽 x 高=8x8  --*/
0x00,0x3C,0x42,0x81,0x81,0x42,0x3C,0x00,
/*--  文字:  1  --*/
/*--  宋体 12;  此字体下对应的点阵为：宽 x 高=8x8  --*/
0x00,0x41,0x41,0xFF,0x01,0x01,0x00,0x00,
/*--  文字:  2  --*/
/*--  宋体 12;  此字体下对应的点阵为：宽 x 高=8x8  --*/
0x00,0x41,0x83,0x85,0x89,0x91,0x63,0x00,
/*--  文字:  3  --*/
/*--  宋体 12;  此字体下对应的点阵为：宽 x 高=8x8  --*/
0x00,0x00,0x81,0x89,0x99,0xAA,0x4C,0x00,
/*--  文字:  4  --*/
/*--  宋体 12;  此字体下对应的点阵为：宽 x 高=8x8  --*/
0x00,0x0C,0x1C,0x25,0x45,0xFF,0x05,0x00,
/*--  文字:  5  --*/
/*--  宋体 12;  此字体下对应的点阵为：宽 x 高=8x8  --*/
0x00,0xFA,0x89,0x91,0x91,0x91,0x8E,0x00,
/*--  文字:  6  --*/
/*--  宋体 12;  此字体下对应的点阵为：宽 x 高=8x8  --*/
0x00,0x3C,0x4A,0x91,0x91,0x8A,0x04,0x00,
/*--  文字:  7  --*/
/*--  宋体 12;  此字体下对应的点阵为：宽 x 高=8x8  --*/
0x00,0xC0,0x80,0x8F,0x90,0xA0,0xC0,0x00,
/*--  文字:  8  --*/
/*--  宋体 12;  此字体下对应的点阵为：宽 x 高=8x8  --*/
0x00,0x00,0x66,0x99,0x89,0x99,0x66,0x00,
/*--  文字:  9  --*/
/*--  宋体 12;  此字体下对应的点阵为：宽 x 高=8x8  --*/
0x00,0x72,0x89,0x89,0x89,0x52,0x3C,0x00,
/*--  文字:  :  --*/
/*--  宋体 12;  此字体下对应的点阵为：宽 x 高=8x8  --*/
```

```
0x00,0x00,0x00,0x66,0x66,0x00,0x00,0x00,
/*--  文字:  ;  --*/
/*--  宋体 12;  此字体下对应的点阵为：宽 x 高=8x8   --*/
0x00,0x00,0x01,0xCE,0xCC,0x00,0x00,0x00,
/*--  文字:  <  --*/
/*--  宋体 12;  此字体下对应的点阵为：宽 x 高=8x8   --*/
0x00,0x08,0x14,0x22,0x41,0x00,0x00,0x00,
/*--  文字:  =  --*/
/*--  宋体 12;  此字体下对应的点阵为：宽 x 高=8x8   --*/
0x00,0x14,0x14,0x14,0x14,0x14,0x14,0x00,
/*--  文字:  >  --*/
/*--  宋体 12;  此字体下对应的点阵为：宽 x 高=8x8   --*/
0x00,0x41,0x22,0x14,0x08,0x00,0x00,0x00,
/*--  文字:  ?  --*/
/*--  宋体 12;  此字体下对应的点阵为：宽 x 高=8x8   --*/
0x00,0x60,0x80,0x8D,0x90,0xA0,0x40,0x00,
/*--  文字:  @  --*/
/*--  宋体 12;  此字体下对应的点阵为：宽 x 高=8x8   --*/
0x3C,0x42,0x9D,0xA5,0x9D,0x46,0x3C,0x00,
/*--  文字:  A  --*/
/*--  宋体 12;  此字体下对应的点阵为：宽 x 高=8x8   --*/
0x01,0x07,0x39,0xC8,0x68,0x1C,0x03,0x01,
/*--  文字:  B  --*/
/*--  宋体 12;  此字体下对应的点阵为：宽 x 高=8x8   --*/
0x00,0x81,0xFF,0x89,0x89,0x89,0x7E,0x00,
/*--  文字:  C  --*/
/*--  宋体 12;  此字体下对应的点阵为：宽 x 高=8x8   --*/
0x00,0x18,0x66,0x81,0x81,0x81,0x42,0x00,
/*--  文字:  D  --*/
/*--  宋体 12;  此字体下对应的点阵为：宽 x 高=8x8   --*/
0x81,0xFF,0x81,0x81,0x81,0x42,0x3C,0x00,
/*--  文字:  E  --*/
/*--  宋体 12;  此字体下对应的点阵为：宽 x 高=8x8   --*/
0x00,0x81,0xFF,0x91,0x91,0xB9,0x83,0x00,
/*--  文字:  F  --*/
/*--  宋体 12;  此字体下对应的点阵为：宽 x 高=8x8   --*/
0x81,0xFF,0x91,0x90,0xB8,0x80,0x40,0x00,
/*--  文字:  G  --*/
/*--  宋体 12;  此字体下对应的点阵为：宽 x 高=8x8   --*/
0x18,0x66,0x81,0x81,0x89,0x4E,0x08,0x00,
/*--  文字:  H  --*/
/*--  宋体 12;  此字体下对应的点阵为：宽 x 高=8x8   --*/
0x81,0xFF,0x89,0x08,0x89,0xFF,0x81,0x00,
/*--  文字:  I  --*/
```

```
/*--  宋体 12;  此字体下对应的点阵为：宽 x 高=8x8   --*/
0x00,0x81,0x81,0xFF,0x81,0x81,0x00,0x00,
/*--  文字:  J  --*/
/*--  宋体 12;  此字体下对应的点阵为：宽 x 高=8x8   --*/
0x00,0x02,0x81,0x81,0xFE,0x80,0x80,0x00,
/*--  文字:  K  --*/
/*--  宋体 12;  此字体下对应的点阵为：宽 x 高=8x8   --*/
0x81,0xFF,0x89,0x18,0xA5,0xC3,0x81,0x00,
/*--  文字:  L  --*/
/*--  宋体 12;  此字体下对应的点阵为：宽 x 高=8x8   --*/
0x00,0x80,0xFF,0x81,0x01,0x01,0x03,0x00,
/*--  文字:  M  --*/
/*--  宋体 12;  此字体下对应的点阵为：宽 x 高=8x8   --*/
0x81,0xFF,0xF0,0x0F,0xF0,0xFF,0x81,0x00,
/*--  文字:  N  --*/
/*--  宋体 12;  此字体下对应的点阵为：宽 x 高=8x8   --*/
0x81,0xFF,0x41,0x30,0x8C,0xFF,0x81,0x00,
/*--  文字:  O  --*/
/*--  宋体 12;  此字体下对应的点阵为：宽 x 高=8x8   --*/
0x3C,0x42,0x81,0x81,0x81,0x42,0x3C,0x00,
/*--  文字:  P  --*/
/*--  宋体 12;  此字体下对应的点阵为：宽 x 高=8x8   --*/
0x81,0xFF,0x89,0x88,0x88,0x88,0x70,0x00,
/*--  文字:  Q  --*/
/*--  宋体 12;  此字体下对应的点阵为：宽 x 高=8x8   --*/
0x38,0x44,0x8A,0x8A,0x86,0x45,0x79,0x00,
/*--  文字:  R  --*/
/*--  宋体 12;  此字体下对应的点阵为：宽 x 高=8x8   --*/
0x81,0xFF,0x91,0x90,0x98,0x94,0x63,0x01,
/*--  文字:  S  --*/
/*--  宋体 12;  此字体下对应的点阵为：宽 x 高=8x8   --*/
0x00,0x62,0x91,0x89,0x89,0x85,0x42,0x00,
/*--  文字:  T  --*/
/*--  宋体 12;  此字体下对应的点阵为：宽 x 高=8x8   --*/
0xC0,0x80,0x81,0xFF,0x81,0x80,0xC0,0x00,
/*--  文字:  U  --*/
/*--  宋体 12;  此字体下对应的点阵为：宽 x 高=8x8   --*/
0x80,0xFE,0x81,0x01,0x01,0x81,0xFE,0x80,
/*--  文字:  V  --*/
/*--  宋体 12;  此字体下对应的点阵为：宽 x 高=8x8   --*/
0x80,0xF0,0x8C,0x03,0x06,0x98,0xE0,0x80,
/*--  文字:  W  --*/
/*--  宋体 12;  此字体下对应的点阵为：宽 x 高=8x8   --*/
0xF8,0x87,0x0C,0x70,0x0C,0x87,0xF8,0x00,
```

```
/*--  文字:  X  --*/
/*--  宋体 12;  此字体下对应的点阵为：宽 x 高=8x8  --*/
0x81,0xC3,0xA5,0x18,0x18,0xA5,0xC3,0x81,
/*--  文字:  Y  --*/
/*--  宋体 12;  此字体下对应的点阵为：宽 x 高=8x8  --*/
0x80,0xE0,0x99,0x0F,0x99,0xE0,0x80,0x00,
/*--  文字:  Z  --*/
/*--  宋体 12;  此字体下对应的点阵为：宽 x 高=8x8  --*/
0x43,0x87,0x8D,0x99,0xB1,0xE1,0xC3,0x00,
/*--  文字:  [  --*/
/*--  宋体 12;  此字体下对应的点阵为：宽 x 高=8x8  --*/
0x00,0x00,0x00,0xFF,0x81,0x81,0x00,0x00,
/*--  文字:  \  --*/
/*--  宋体 12;  此字体下对应的点阵为：宽 x 高=8x8  --*/
0x80,0x40,0x20,0x10,0x08,0x04,0x02,0x01,
/*--  文字:  ]  --*/
/*--  宋体 12;  此字体下对应的点阵为：宽 x 高=8x8  --*/
0x00,0x00,0x81,0x81,0xFF,0x00,0x00,0x00,
/*--  文字:  ^  --*/
/*--  宋体 12;  此字体下对应的点阵为：宽 x 高=8x8  --*/
0x00,0x00,0x20,0x40,0x80,0x40,0x20,0x00,
/*--  文字:  _  --*/
/*--  宋体 12;  此字体下对应的点阵为：宽 x 高=8x8  --*/
0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,
/*--  文字:  `  --*/
/*--  宋体 12;  此字体下对应的点阵为：宽 x 高=8x8  --*/
0x00,0x40,0x40,0x20,0x00,0x00,0x00,0x00,
/*--  文字:  a  --*/
/*--  宋体 12;  此字体下对应的点阵为：宽 x 高=8x8  --*/
0x00,0x26,0x49,0x51,0x51,0x51,0x3f,0x01,
/*--  文字:  b  --*/
/*--  宋体 12;  此字体下对应的点阵为：宽 x 高=8x8  --*/
0x00,0x80,0xFE,0x11,0x11,0x11,0x0E,0x00,
/*--  文字:  c  --*/
/*--  宋体 12;  此字体下对应的点阵为：宽 x 高=8x8  --*/
0x00,0x1C,0x22,0x41,0x41,0x41,0x22,0x00,
/*--  文字:  d  --*/
/*--  宋体 12;  此字体下对应的点阵为：宽 x 高=8x8  --*/
0x00,0x0E,0x11,0x11,0x11,0xFF,0x81,0x00,
/*--  文字:  e  --*/
/*--  宋体 12;  此字体下对应的点阵为：宽 x 高=8x8  --*/
0x00,0x3E,0x49,0x49,0x49,0x49,0x32,0x00,
/*--  文字:  f  --*/
/*--  宋体 12;  此字体下对应的点阵为：宽 x 高=8x8  --*/
```

```
0x00,0x10,0x11,0x7F,0x91,0x90,0x80,0x40,
/*--  文字:  g  --*/
/*--  宋体 12;  此字体下对应的点阵为：宽 x 高=8x8  --*/
0x00,0x6A,0x95,0x95,0x95,0xE2,0x80,0x00,
/*--  文字:  h  --*/
/*--  宋体 12;  此字体下对应的点阵为：宽 x 高=8x8  --*/
0x00,0x81,0xFF,0x09,0x10,0x11,0x0F,0x01,
/*--  文字:  i  --*/
/*--  宋体 12;  此字体下对应的点阵为：宽 x 高=8x8  --*/
0x00,0x21,0xA1,0xBF,0x01,0x01,0x00,0x00,
/*--  文字:  j  --*/
/*--  宋体 12;  此字体下对应的点阵为：宽 x 高=8x8  --*/
0x00,0x03,0x01,0x01,0xBE,0x00,0x00,0x00,
/*--  文字:  k  --*/
/*--  宋体 12;  此字体下对应的点阵为：宽 x 高=8x8  --*/
0x81,0xFF,0x05,0x0C,0x13,0x31,0x01,0x00,
/*--  文字:  l  --*/
/*--  宋体 12;  此字体下对应的点阵为：宽 x 高=8x8  --*/
0x00,0x81,0x81,0xFF,0x01,0x01,0x00,0x00,
/*--  文字:  m  --*/
/*--  宋体 12;  此字体下对应的点阵为：宽 x 高=8x8  --*/
0x41,0x7F,0x41,0x40,0x7F,0x41,0x40,0x3F,
/*--  文字:  n  --*/
/*--  宋体 12;  此字体下对应的点阵为：宽 x 高=8x8  --*/
0x41,0x7F,0x21,0x40,0x40,0x41,0x3F,0x01,
/*--  文字:  o  --*/
/*--  宋体 12;  此字体下对应的点阵为：宽 x 高=8x8  --*/
0x00,0x3E,0x41,0x41,0x41,0x41,0x3E,0x00,
/*--  文字:  p  --*/
/*--  宋体 12;  此字体下对应的点阵为：宽 x 高=8x8  --*/
0x00,0x80,0xFF,0x49,0x88,0x88,0x70,0x00,
/*--  文字:  q  --*/
/*--  宋体 12;  此字体下对应的点阵为：宽 x 高=8x8  --*/
0x00,0x60,0x90,0x08,0x09,0xFF,0x01,0x00,
/*--  文字:  r  --*/
/*--  宋体 12;  此字体下对应的点阵为：宽 x 高=8x8  --*/
0x00,0x81,0xFF,0x41,0x80,0x80,0x40,0x00,
/*--  文字:  s  --*/
/*--  宋体 12;  此字体下对应的点阵为：宽 x 高=8x8  --*/
0x00,0x33,0x49,0x49,0x49,0x49,0x66,0x00,
/*--  文字:  t  --*/
/*--  宋体 12;  此字体下对应的点阵为：宽 x 高=8x8  --*/
0x00,0x20,0x20,0xFE,0x21,0x21,0x00,0x00,
/*--  文字:  u  --*/
```

```
/*--  宋体 12;  此字体下对应的点阵为：宽 x 高=8x8    --*/
0x40,0x7E,0x01,0x01,0x01,0x42,0x7F,0x01,
/*--  文字:  v  --*/
/*--  宋体 12;  此字体下对应的点阵为：宽 x 高=8x8    --*/
0x80,0xE0,0x9C,0x03,0x04,0x98,0xE0,0x80,
/*--  文字:  w  --*/
/*--  宋体 12;  此字体下对应的点阵为：宽 x 高=8x8    --*/
0xFC,0x83,0x0C,0xF0,0x0C,0x83,0xFC,0x80,
/*--  文字:  x  --*/
/*--  宋体 12;  此字体下对应的点阵为：宽 x 高=8x8    --*/
0x00,0x41,0x63,0x1D,0x5C,0x63,0x41,0x00,
/*--  文字:  y  --*/
/*--  宋体 12;  此字体下对应的点阵为：宽 x 高=8x8    --*/
0x81,0xC1,0xB1,0x0E,0x18,0xA0,0xC0,0x80,
/*--  文字:  z  --*/
/*--  宋体 12;  此字体下对应的点阵为：宽 x 高=8x8    --*/
0x00,0x61,0x43,0x4D,0x51,0x61,0x43,0x00,
/*--  文字:  {  --*/
/*--  宋体 12;  此字体下对应的点阵为：宽 x 高=8x8    --*/
0x00,0x00,0x00,0x08,0x36,0x41,0x41,0x00,
/*--  文字:  |  --*/
/*--  宋体 12;  此字体下对应的点阵为：宽 x 高=8x8    --*/
0x00,0x00,0x00,0xFF,0xFF,0x00,0x00,0x00,
/*--  文字:  }  --*/
/*--  宋体 12;  此字体下对应的点阵为：宽 x 高=8x8    --*/
0x00,0x00,0x01,0x01,0xEE,0x10,0x00,0x00,
/*--  文字:  ~  --*/
/*--  宋体 12;  此字体下对应的点阵为：宽 x 高=8x8    --*/
0x20,0x40,0x40,0x20,0x10,0x10,0x20,0x40
};
uint8 const  gc_lcdimagecode[][LCD_PAGE_LARGE][LCD_LINE_LARGE]={
/*--  cs 图标，下方，西科大计算机学院 --*/
/*--  宽度 x 高度=128x64  --*/

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x
00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x
00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x03,0x07,0x1F,0x1F,0x3F,0x3F,0x1F,0x1F,0x07,0x01,0x0
0,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x
x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0
0,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0
x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x
00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
```

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x
00,0x00,0x01,0x07,0x0F,0x7F,0x7F,0xFE,0xFE,0xF0,0xE0,0x80,0x80,0xE0,0xF8,0xF8,0xFB,0x
7F,0x3F,0x0B,0x03,0x03,0x03,0x03,0x03,0x03,0x03,0x01,0x00,0x00,0x00,0x00,0x00,0x00,0x00
,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x
00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x
00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x01,0x03,0x0F,0x
1F,0x7F,0xFF,0xFE,0xF8,0xE0,0xC0,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x
C0,0xE0,0xFC,0xFE,0xFF,0xFF,0xDF,0xCF,0xC3,0xC1,0x80,0x00,0x00,0x00,0x00,0x00,0x00,0
x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0
0,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0
x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x
00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x70,0xF8,0xFC,0xFC,0
xFC,0xFC,0xFC,0x7C,0x7C,0x7C,0x7C,0x78,0x10,0x38,0x7C,0x7C,0x7C,0x7C,0x7C,0x7C,0x7
C,0x7C,0x7C,0x7C,0x7C,0xFC,0xFC,0xFC,0xFC,0xF8,0xF8,0x20,0x00,0x00,0x00,0x00,0x00,0
x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0
0,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0
x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x
00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x
00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x
00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x
00,0x00,0x00,0x00,0x00,0x00,
0x40,0x4F,0x48,0x48,0x48,0x7F,0x48,0x48,0x48,0x7F,0x48,0x48,0x48,0x4F,0x40,0x00,0x08,0x
48,0x49,0x4E,0x7F,0x8A,0x89,0x00,0x44,0x33,0x00,0x00,0xFF,0x00,0x00,0x00,0x04,0x04,0x0
4,0x04,0x04,0x04,0x05,0xFE,0x05,0x04,0x04,0x04,0x04,0x04,0x04,0x00,0x04,0x84,0x74,0x27,
0x00,0x00,0x04,0x04,0x04,0x04,0xFF,0x04,0x04,0x04,0x04,0x00,0x10,0xE0,0x5F,0x55,0x75,0x
55,0x55,0x15,0x35,0xD5,0x55,0x7F,0x50,0x40,0x40,0x00,0x10,0x10,0x13,0xFF,0x12,0x11,0x1
0,0x00,0x7F,0x40,0x40,0x40,0x7F,0x00,0x00,0x00,0x02,0x0C,0x08,0x48,0x3A,0x2A,0x0A,0x8
A,0x7A,0x2B,0x0A,0x18,0xEA,0x4C,0x08,0x00,0x7F,0x40,0x4C,0x52,0x61,0x30,0x24,0x24,0x
A4,0x64,0x24,0x24,0x24,0x30,0x20,0x00,
0x00,0xFE,0x14,0x24,0x44,0x84,0x04,0x04,0x04,0x84,0x44,0x44,0x44,0xFE,0x00,0x00,0x20,0
x40,0x80,0x00,0xFF,0x00,0x20,0x20,0x20,0x40,0x40,0x40,0xFF,0x80,0x80,0x00,0x00,0x01,0x0
2,0x04,0x08,0x30,0xC0,0x00,0x80,0x60,0x10,0x0C,0x06,0x03,0x02,0x00,0x00,0x00,0x00,0xFE,
0x04,0x08,0x10,0x00,0x00,0x00,0xFF,0x00,0x00,0x00,0x00,0x00,0x10,0x10,0xD1,0x56,0x78,0
x50,0x50,0x50,0x50,0x7F,0x50,0xD0,0x10,0x10,0x10,0x00,0x20,0xC0,0x00,0xFF,0x00,0x82,0x
0C,0x30,0xC0,0x00,0x00,0x00,0xFC,0x02,0x1E,0x00,0x00,0x40,0x40,0x40,0x40,0x40,0x42,0x4

1,0xFE,0x40,0x40,0x40,0x40,0x40,0x40,0x00,0xFF,0x00,0x40,0x20,0xC1,0x82,0x8C,0xF0,0x80,0x80,0xFE,0x81,0x81,0x81,0x8F,0x00,
0x00,

/*-- 文字: CS --*/
/*-- 宋体 48; 此字体下对应的点阵为：宽 x 高=128*64 --*/
0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,
0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0x00,0x00,0x00,0x00,0x00,0x00,0x01,0x03,0x07,0x0F,0x1F,0x1F,0x3F,0x3E,0x3C,0x38,0x38,0x38,0x30,0x38,0x38,0x38,0x3C,0x3E,0x1F,0x1F,0x3F,0x3F,0x3F,0x07,0x00,0x00,0x00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0x00,0x00,0x00,0x00,0x01,0x03,0x07,0x0F,0x1F,0x1F,0x3F,0x3C,0x38,0x38,0x38,0x30,0x30,0x38,0x38,0x38,0x3C,0x3E,0x1F,0x1F,0x1F,0x1F,0x3F,0x3F,0x3F,0x00,0x00,0x00,0x00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,
0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0x00,0x00,0x00,0x0F,0x3F,0x7F,0xFF,0xFF,0xFF,0xFE,0xF0,0x80,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0xC0,0xF0,0xFC,0xFC,0xFC,0x7C,0x00,0x00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0x00,0x00,0x00,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xE7,0x01,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0xC0,0xF0,0xFC,0xFC,0xFC,0xFC,0x00,0x00,0x00,0x00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,
0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0x00,0x00,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xF8,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0x00,0x00,0x00,0xC0,0xF0,0xF8,0xFC,0xFC,0xFE,0xFE,0xFF,0xFF,0x7F,0x3F,0x3F,0x1F,0x0F,0x0F,0x0

7,0x07,0x07,0x03,0x03,0x01,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0X00,0X00,0X00
,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,
0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00
,0x00,0x00,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0x1F,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0X00,0
X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,
0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x80,0x80,0xC0,0xC0,0xE0,0xE0,0xF0,0
xF8,0xF8,0xFC,0xFF,0xFF,0xFF,0xFF,0x7F,0x3F,0x1F,0x07,0x00,0x00,0x00,0X00,0X00,0X00,
0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,
0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00
,0x00,0x00,0xC0,0xF8,0xFE,0xFF,0xFF,0xFF,0xFF,0x3F,0x07,0x01,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x03,0x07,0x3F,0xFF,0xFF,0xFE,0xF8,0x00,0x00,0X00,0
X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,
0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0x00,0x00,
0x00,0xFF,0xFF,0xFF,0xFF,0x3F,0x07,0x01,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0
x00,0x00,0x00,0x00,0xCF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFE,0x00,0x00,0x00,0X00,0X00,0X00,
0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,
0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00
,0x00,0x00,0x00,0x00,0x00,0x80,0xC0,0xE0,0xF0,0xF0,0xF8,0xF8,0xFC,0x7C,0x3C,0x3C,0x3
C,0x1C,0x3C,0x3C,0x3C,0x7C,0x7C,0xF8,0xF8,0xF0,0xE0,0xC0,0x80,0x00,0x00,0x00,0x00,0
X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,
0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0x00,
0x00,0x00,0x80,0xFC,0xFC,0xFC,0xFC,0xF8,0xF8,0xF8,0x7C,0x3C,0x1C,0x1C,0x1C,0x0C,0x
1C,0x1C,0x1C,0x3C,0x7C,0xFC,0xF8,0xF8,0xF0,0xE0,0xC0,0x80,0x00,0x00,0x00,0x00,0X00,0
X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,
0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00
,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x
00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0X00,0X0
0,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X
00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0x00,0x00,0x0
0,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0
x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0X00,0X00,0X00,0X00,0
X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,

/*-- 文字: ARM --*/
/*-- 宋体 48; 此字体下对应的点阵为：宽 x 高=128*64 --*/
0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x
00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x
00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x

00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0
X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,
0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x01,0x1F,0x3F,0x3F,0x3F,0x3F,0x3F,0
x3F,0x1F,0x01,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x1
8,0x18,0x1C,0x1F,0x1F,0x1F,0x1F,0x1F,0x1F,0x1C,0x18,0x18,0x18,0x18,0x1C,0x1C,0x1C,0x1
E,0x1F,0x1F,0x1F,0x0F,0x0F,0x07,0x07,0x03,0x00,0x00,0x00,0x00,0x00,0x18,0x18,0x1C,0x1F
,0x1F,0x1F,0x1F,0x1F,0x1F,0x1F,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x01,0
x1F,0x1F,0x1F,0x1F,0x1F,0x1F,0x1F,0x1F,0x1C,0x18,0x18,0X00,0X00,0X00,0X00,0X00,0X00
,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,
0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x01,0x1F,0xFF,0xFF,0xFF,0xFF,0xFE,0xFF,0xFF,
0xFF,0xFF,0xFF,0x1F,0x01,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0
x00,0x00,0x00,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x
00,0x00,0x80,0xF9,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0xF
F,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0x0F,0x00,0x00,0x00,0x00,0x00,0x00,0x01,0x1F,0xF
F,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0x00,0x00,0x00,0X00,0X00,0X00,0X00,0X00,0
X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,
0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x01,0x1F,0xFF,0xFF,0xFF,0xFF,0xF0,0x00,0x00,0xE0,0xFE,
0xFF,0xFF,0xFF,0xFF,0xFF,0x1F,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0
x00,0x00,0x00,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0x01,0x01,0x01,0x01,0x01,0x03,0x03,0x03,0x
07,0x0F,0x1F,0xFF,0xFF,0xFF,0xFE,0xFC,0xF8,0xE0,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x
FF,0xFF,0xFF,0xFC,0xFF,0xFF,0xFF,0xFF,0xFF,0x7F,0x07,0x00,0x00,0x00,0x0F,0xFF,0xFF,0
xFF,0xFF,0xF0,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0x00,0x00,0x00,0X00,0X00,0X00,0X00,0X0
0,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,
0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0x00,0x00,
0x00,0x00,0x00,0x00,0x01,0x1F,0xFF,0xFF,0xFF,0xFF,0xFC,0x0C,0x0C,0x0C,0x0C,0x0C,0x0
C,0xEC,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0x0F,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0
0,0x00,0x00,0x00,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0x80,0x80,0x80,0xF0,0xFC,0xFF,0xFF,0xF
F,0xFF,0xFF,0xBF,0x87,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0xFF,0xFF,0xFF,0x00,0xC0,0xFE,0xFF,0xFF,0xFF,0xFF,0xFF,0x3F,0x0F,0xFF,0xFF,0xFF,0xF
F,0xF0,0x00,0x00,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0x00,0x00,0x00,0X00,0X00,0X00,0X00,0
X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,
0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0x00,0x00,
0x00,0x00,0x00,0x01,0x1F,0xFF,0xFF,0xFF,0xFF,0xF0,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0
x00,0x00,0xF8,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0x0F,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x
00,0x00,0x00,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0x00,0x00,0x00,0x00,0x00,0x80,0xE0,0xFC,0x
FF,0xFF,0xFF,0xFF,0xFF,0x3F,0x07,0x01,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0xF
F,0xFF,0xFF,0x00,0x00,0x00,0xE0,0xFE,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xF8,0x80,0x
00,0x00,0x00,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0x00,0x00,0x00,0X00,0X00,0X00,0X00,0X00,0
X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,
0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0x00,0x0C,
0x1C,0x1C,0xFC,0xFC,0xFC,0xFC,0xFC,0x1C,0x1C,0x0C,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0

0,0x00,0x0C,0x0C,0x9C,0xFC,0xFC,0xFC,0xFC,0xFC,0xFC,0xFC,0x1C,0x1C,0x0C,0x00,0x00,
0x00,0x0C,0x0C,0x1C,0xFC,0xFC,0xFC,0xFC,0xFC,0xFC,0x1C,0x0C,0x0C,0x00,0x00,0x00,0x
00,0x00,0x80,0xE0,0xFC,0xFC,0xFC,0xFC,0xFC,0xFC,0x3C,0x1C,0x0C,0x0C,0x00,0x00,0x0C,
0x0C,0x1C,0xFC,0xFC,0xFC,0x1C,0x0C,0x0C,0x00,0x00,0xF0,0xFC,0xFC,0xFC,0xFC,0xF8,0x
80,0x00,0x00,0x0C,0x0C,0x1C,0xFC,0xFC,0xFC,0xFC,0xFC,0xFC,0x1C,0x0C,0x0C,0X00,0X0
0,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,
0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x
00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0
0,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x
00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0
X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,

/*-- 文字: ARM 下方西南科技大学计算机 --*/
/*-- 宋体 48; 此字体下对应的点阵为：宽 x 高=128*64 --*/
0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x01,0x1F,0x3F,0x3F,0x3F,0x3F,0x3F,0
x3F,0x1F,0x01,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x1
8,0x18,0x1C,0x1F,0x1F,0x1F,0x1F,0x1F,0x1F,0x1C,0x18,0x18,0x18,0x18,0x1C,0x1C,0x1C,0x1
E,0x1F,0x1F,0x1F,0x0F,0x0F,0x07,0x07,0x03,0x00,0x00,0x00,0x00,0x00,0x18,0x18,0x1C,0x1F
,0x1F,0x1F,0x1F,0x1F,0x1F,0x1F,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x01,0
x1F,0x1F,0x1F,0x1F,0x1F,0x1F,0x1F,0x1F,0x1C,0x18,0x18,0X00,0X00,0X00,0X00,0X00,0X00
,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,
0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x01,0x1F,0xFF,0xFF,0xFF,0xFF,0xFE,0xFF,0xFF,
0xFF,0xFF,0xFF,0x1F,0x01,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0
x00,0x00,0x00,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x
00,0x00,0x80,0xF9,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0xF
F,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0x0F,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x01,0x1F,0xF
F,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0x00,0x00,0x00,0X00,0X00,0X00,0X00,0X00,0
X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,
0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x01,0x1F,0xFF,0xFF,0xFF,0xFF,0xF0,0x00,0x00,0xE0,0xFE,
0xFF,0xFF,0xFF,0xFF,0xFF,0x1F,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0
x00,0x00,0x00,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0x01,0x01,0x01,0x01,0x01,0x03,0x03,0x03,0x
07,0x0F,0x1F,0xFF,0xFF,0xFF,0xFE,0xFC,0xF8,0xE0,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x
FF,0xFF,0xFF,0xFC,0xFF,0xFF,0xFF,0xFF,0xFF,0x7F,0x07,0x00,0x00,0x00,0x0F,0xFF,0xFF,0
xFF,0xFF,0xF0,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0x00,0x00,0x00,0X00,0X00,0X00,0X00,0X0
0,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,
0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0x00,0x00,
0x00,0x00,0x00,0x00,0x01,0x1F,0xFF,0xFF,0xFF,0xFF,0xFC,0x0C,0x0C,0x0C,0x0C,0x0C,0x0

```
C,0xEC,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0x0F,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0
0,0x00,0x00,0x00,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0x80,0x80,0x80,0xF0,0xFC,0xFF,0xFF,0xF
F,0xFF,0xFF,0xBF,0x87,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0xFF,0xFF,0xFF,0x00,0xC0,0xFE,0xFF,0xFF,0xFF,0xFF,0xFF,0x3F,0x0F,0xFF,0xFF,0xFF,0xF
F,0xF0,0x00,0x00,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0x00,0x00,0x00,0X00,0X00,0X00,0X00,0
X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,
0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0x00,0x00,
0x00,0x00,0x01,0x1F,0xFF,0xFF,0xFF,0xFF,0xF0,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0
x00,0x00,0xF8,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0x0F,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x
00,0x00,0x00,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0x00,0x00,0x00,0x00,0x00,0x80,0xE0,0xFC,0x
FF,0xFF,0xFF,0xFF,0xFF,0x3F,0x07,0x01,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0xF
F,0xFF,0xFF,0x00,0x00,0x00,0xE0,0xFE,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xF8,0x80,0x
00,0x00,0x00,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0x00,0x00,0x00,0X00,0X00,0X00,0X00,0X00,0
X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,
0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0x00,0x0C,
0x1C,0x1C,0xFC,0xFC,0xFC,0xFC,0xFC,0x1C,0x1C,0x0C,0x00,0x00,0x00,0x00,0x00,0x00,0x0
0,0x00,0x0C,0x0C,0x9C,0xFC,0xFC,0xFC,0xFC,0xFC,0xFC,0xFC,0x1C,0x1C,0x0C,0x00,0x00,
0x00,0x0C,0x0C,0x1C,0xFC,0xFC,0xFC,0xFC,0xFC,0xFC,0x1C,0x0C,0x0C,0x00,0x00,0x00,0x
00,0x00,0x80,0xE0,0xFC,0xFC,0xFC,0xFC,0xFC,0xFC,0x3C,0x1C,0x0C,0x0C,0x00,0x00,0x0C,
0x0C,0x1C,0xFC,0xFC,0xFC,0x1C,0x0C,0x0C,0x00,0x00,0xF0,0xFC,0xFC,0xFC,0xFC,0xF8,0x
80,0x00,0x00,0x0C,0x0C,0x1C,0xFC,0xFC,0xFC,0xFC,0xFC,0xFC,0x1C,0x0C,0x0C,0X00,0X0
0,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,
0x40,0x4F,0x48,0x48,0x48,0x7F,0x48,0x48,0x48,0x7F,0x48,0x48,0x48,0x4F,0x40,0x00,0x08,0x
48,0x49,0x4E,0x7F,0x8A,0x89,0x00,0x44,0x33,0x00,0x00,0xFF,0x00,0x00,0x00,0x04,0x04,0x0
4,0x04,0x04,0x04,0x05,0xFE,0x05,0x04,0x04,0x04,0x04,0x04,0x04,0x00,0x04,0x84,0x74,0x27,
0x00,0x00,0x04,0x04,0x04,0x04,0xFF,0x04,0x04,0x04,0x04,0x00,0x10,0xE0,0x5F,0x55,0x75,0x
55,0x55,0x15,0x35,0xD5,0x55,0x7F,0x50,0x40,0x40,0x00,0x10,0x10,0x13,0xFF,0x12,0x11,0x1
0,0x00,0x7F,0x40,0x40,0x40,0x7F,0x00,0x00,0x00,0x02,0x0C,0x08,0x48,0x3A,0x2A,0x0A,0x8
A,0x7A,0x2B,0x0A,0x18,0xEA,0x4C,0x08,0x00,0x7F,0x40,0x4C,0x52,0x61,0x30,0x24,0x24,0x
A4,0x64,0x24,0x24,0x24,0x30,0x20,0x00,
0x00,0xFE,0x14,0x24,0x44,0x84,0x04,0x04,0x04,0x84,0x44,0x44,0x44,0xFE,0x00,0x00,0x20,0
x40,0x80,0x00,0xFF,0x00,0x20,0x20,0x20,0x40,0x40,0x40,0xFF,0x80,0x80,0x00,0x00,0x01,0x0
2,0x04,0x08,0x30,0xC0,0x00,0x80,0x60,0x10,0x0C,0x06,0x03,0x02,0x00,0x00,0x00,0x00,0xFE,
0x04,0x08,0x10,0x00,0x00,0x00,0xFF,0x00,0x00,0x00,0x00,0x00,0x10,0x10,0xD1,0x56,0x78,0
x50,0x50,0x50,0x50,0x7F,0x50,0xD0,0x10,0x10,0x10,0x00,0x20,0xC0,0x00,0xFF,0x00,0x82,0x
0C,0x30,0xC0,0x00,0x00,0x00,0xFC,0x02,0x1E,0x00,0x00,0x40,0x40,0x40,0x40,0x40,0x42,0x4
1,0xFE,0x40,0x40,0x40,0x40,0x40,0x40,0x00,0xFF,0x00,0x40,0x20,0xC1,0x82,0x8C,0xF0,0x80
,0x80,0xFE,0x81,0x81,0x81,0x8F,0x00,
};

  #endif
/***************************************************************************
  *                                   End of Entire File                                        *
  ***************************************************************************/
```

实验十一 AD 转换实验

说明：
主文件中需要包括以下头文件“config.h”,”dataled.h”, ”ad.h” ” timer.h”
主函数中：
ADInit()初始化 AD，其中主要是设置 ADCR，同时也初始化 LED，TIMER；
ADReadData()主要是读取数据；
在读取过程可以相隔一定时间读取，因为芯片很敏感！如果不这样会导致数码管显示变化速度较快，末位会因为不停的变化而模糊。

```
/************************************Copyright *********************************/
/**
;**                                  西南科技大学计算机科学学院
;**                                   http://www.cs.suswt.edu.cn
;**
;**                日期：  2007.09.21
;**                描述：西南科技大学计算机学院 CS-II 型实验板 AD 转换工作程序头文件
;**                作者：
;**
;**-------------ad.h 文件
;**-------------------------------------------------------------------------------------------------*/
/**--------------------------------------------------------------------------------
*******************************************************************************/
 #ifndef _AD_H_                          /* avoid being incude more than once */
 #define _AD_H_
/*************************************************
 *                 Header file                   *
 *************************************************/
 #include "config.h"
/*************************************************
 *                 Macros                        *
 *************************************************/
 #define      ADCLK       4500000
 #define      ADBIT       10
 #define      ADBIT2    (10-ADBIT)
  /*************************************************
 *              Function declaration             *
 *************************************************/
 extern void ADInit(void);
 extern uint16 ADReadData(void);
 #endif
 /*******************************************************************************
 *                                End of Entire File                            *
 *******************************************************************************/
```

```
***********************************Copyright (c)*******************************/
/**
;**                              西南科技大学计算机科学学院
;**                               http://www.cs.suswt.edu.cn
;**
;**              日期：     2007.09.21
;**              描述：     ad.c  西南科技大学计算机学院 CS-II 型实验板工作程序，控制
;**              作者：
;**
;**------------ad.c 文件
;**-------------------------------------------------------------------------------------*/
/**-------------------------------------------------------------------------------------
*************************************************************************/
#define _AD_C_
#ifdef _AD_C_
/***********************************************
 *            Header File
 ***********************************************/
#include "ad.h"
#include "config.h"
/***********************************************
 *            Globale variable
 ***********************************************/

/************************************************************************
 * Function Name:    void ADInit(void)
 * Description   :    AD 初始化程序
 ************************************************************************/
void ADInit(void)
{
     PINSEL1 = ( PINSEL1 &0XFC3FFFFF) | 0X01000000;
     ADCR = (0X01<<27)|(0X02<<24)|(0X00<<22)|(0X01<<21)
        |(ADBIT2<<17)|(0X01<<16)|((Fpclk/ADCLK+1)<<8)|(0X01<<0);

}
/************************************************************************
 * Function Name:        uint16 ADReadData(void)
 * Description   :        读取数据
 ************************************************************************/
uint16 ADReadData(void)
{
     uint16 ADdata;
     ADCR |=(1<<24);
     while((ADDR&0X80000000)!=0);
```

```
    ADdata = (ADDR >> 6)&0X3FF;
    return ADdata;
}

#endif
/*****************************************************************************
 *                              End of Entire File                           *
 *****************************************************************************/
```

实验十二　DA 转换

说明：因为 DA 是将数字信号转换成模拟信号。我们不能用数码管，LCD 来观察模拟信号。故我们应用示波器或是万用表来查看转换后的电压。当然我们必须在头文件中应该初使化 DA（DAInit（void））。当我们在主函数中调用 DASetVoltage(float voltage)函数时，我们用示波器测得的电压值应该在 voltage（伏）这个电压附近。

```
/*************************************copyright (c)*********************************/
/**
;**                                西南科技大学计算机科学学院
;**                                  http://www.cs.suswt.edu.cn
;**
;**                日期：     2007.9.20
;**                描述：     DA.h  西南科技大学计算机学院 CS-II 型实验板 DA 工作程
;**                           序，实现对 DA 控制的
;**                           各种接口的定义。
;**                作者：
;**
;**-------------DA.h 文件
;**-------------------------------------------------------------------------------------*/
/**-------------------------------------------------------------------------------------
*************************************************************************************/
#ifndef   _DA_H_
#define   _DA_H_
/************************************************
 *              Header File                     *
 ************************************************/
#include "config.h"

/************************************************
 *                  Macros                      *
 ************************************************/
#define DA_PORT      0xffff00ff          //DA 相应的端口
#define DA_SCK       0x00000010          //P0.4，SCK 时钟
#define DA_RCK       0x00000020          //P0.5，RCK 时钟
#define DA_SI        0x00000040          //P0.6，SI 串行数据
#define DA_LDAC      0x00000080          //P0.7，LDAC 信号

#define DA_REFV     3.85
/*参考电压为 3.85 伏，但理论上应该是 5V(用电压表测出参考电压为 5V),但
实际能输出的最大电压为 3.85V*/
```

```
/*********************************************
 *              Function declaration          *
 *********************************************/
void DAInit(void);                                  //DA 初使化
void DASet(uint8 output_select,uint8 rng);      //DA 设置，输出端口选择与精度选择
void DASetVoltage(float voltage);                   //DA 设置数据，设置要转换的电压大小

#endif
/*****************************************************************************
 *                              End of Entire File                           *

*****************************************************************************/


/*********************************Copyright(c)*********************************/
/**
;**                          西南科技大学计算机科学学院
;**                            http://www.cs.suswt.edu.cn
;**
;**              日期：      2006.4.1
;**              描述：      DA.c  西南科技大学计算机学院 CS-II 型实验板 DA 工作程
;**                          序，实现了 DA 模块中的函数的定义
;**              作者：
;**
;**-------------DA.c 文件
;**-------------------------------------------------------------------------------------------*/
/**-------------------------------------------------------------------------------------------
*****************************************************************************/
#define _DA_C_
#ifdef   _DA_C_
/*********************************************
 *              Header File                   *
 *********************************************/
#include "DA.h"

/*********************************************
 *              Globale variable              *
 *********************************************/
uint8 gc_da_control = 0;      //DA 控制变量，控制 DA 的输出端口与精度

/*********************************************
名称:void DAInit(void)
功能:DA 初使化
*********************************************/
```

```
void DAInit(void)
{
    PINSEL0 &= DA_PORT;        //选择端口功能为 GPIO
    IO0DIR  |= DA_SCK;         //分别设置 SCK,RCK,SI,LDAC 端口方向为输出
    IO0DIR  |= DA_RCK;
    IO0DIR  |= DA_SI;
    IO0DIR  |= DA_LDAC;

    IO0SET  |= DA_RCK;
/*设置 RCK 信号为高电平，当 RCK 产生一个下降沿时，开始执行 DA 转换*/
    IO0CLR  |= DA_LDAC;
/*DA 有几种模式，本实验采用的是 LDAC 一直是低电平（figure 3）*/
}


/*************************************************************
名称:void DASet(uint8 output_select,uint8 rng)
功能:DA 控制设置，可以设置 DA 输出端口,和 DA 精度,其中 output_select 为输出端口序号
(0~3),rng 为精度(0 或 1).
*************************************************************/
void DASet(uint8 output_select,uint8 rng)
{
    output_select = (output_select << 6);
/*把 DA 的输出端口序号与精度存放在 gc_da_control 的前三位*/.
    rng = 0x20 & (rng << 5);

    gc_da_control = output_select + rng;
}

/************************************************
名称:void DelayNs(uint16 time)
功能:延时（NS）
************************************************/
void DelayNs(uint16 time)          //延时函数（延时不精准，可以用定时器实现准确延时)
{
    uint16 i;
    i = time / 100;
    while(i--)
    {
        ;
    }
}
```

```
/************************************************
名称:void DATransfer(uint8 data)
功能:实现 DA 转换,把一个 DA 控制与数据信息写入到 DA 芯片中,并启动 DA 转换
************************************************/
void DATransfer(uint8 data)
{
    uint8 i;
    IO0SET = DA_RCK;                    //RCK 信号为高,在 RCK 下降沿实现 DA 转换

    for(i = 0; i < 3; i++)              //写入 3 位控制信息
    {
        IO0SET = DA_SCK;
/*SCK 信号，当 SCK 信号下降沿时，SI 位信息写入到芯片中*/

        /*把 DA 控制信息写入芯片中（共三位,前两位为端口选择,后一位为精度）*/
        if((gc_da_control & 0x80) == 0x00)          {
            IO0CLR = DA_SI;
        }
        else
        {
            IO0SET = DA_SI;
        }

        IO0CLR = DA_SCK;                //SCK 下降沿，SI 位信息写入到芯片中
        gc_da_control = gc_da_control << 1;
    }

    for(i = 0; i < 8; i++)          //写入 8 位数据
    {
        IO0SET = DA_SCK;

        if((data & 0x80) == 0x00)
        {
            IO0CLR = DA_SI;
        }
        else
        {
            IO0SET = DA_SI;
        }

        IO0CLR = DA_SCK;
        data = data << 1;
    }
```

```
    DelayNs(50);                    //数据传输完毕，要保持至少 50ns 的高电平
    IO0CLR = DA_RCK;                //RCK 下降沿，DA 信息写入到芯片中
    DelayNs(250);                   //保持 RCK 低电平至少 250ns
    IO0SET = DA_RCK;
/*RCK 设置为高电平。当 RCK 为高电平时，DA 转换结果不变*/
}

/*************************************************
名称:void DASetValue(uint8 voltage)
功能:DA 转换，使 DA 输出为设置的电压（voltage V），设置 DA 要转换的电压(如果 voltage
设置为 2.4 的话，那么 DA 转换结果为 2.4V)
*************************************************/
void DASetVoltage(float voltage)
 {
    uint8 value;
  /*公式为：Vo = DA_REFV * CODE * ( 1 + RNG);  RNG 为 gc_da_control 的左边第三位.*/
    if((gc_da_control & 0x20) != 0)
    {
        value = (uint8)((128 * voltage) / DA_REFV);
    }
    else
    {
        value = (uint8)((256 * voltage) / DA_REFV);
    }

    DATransfer(value);
  /*将 voltage 对应的电压转换成 8 位二进制数，并把它写入到 DA 芯片中执行 DA 转换*/
}

#endif
/*****************************************************************************
 *                                End of Entire File                         *
*****************************************************************************/
```

Experimental Instructor

计算机科学与技术学院

(School of Computer Science and Technology)

计算机科学与技术学院自编实验指导书

软件项目管理

计算机基本技能训练

计算机网络

计算机组成原理

C++程序语言设计

计算机图形学

编译原理

软件测试技术

汇编语言程序设计

网络程序设计

反病毒技术

Linux 系统及程序设计

Windows 程序设计

软件开发综合实验

Java 程序设计

网络攻防对抗

计算机制图

微机原理及应用

嵌入式系统实验

数据库应用设计

单片机与接口应用设计

信息电器分析与设计

数据结构应用设计

计算机操作系统综合实验

动态网页制作技术

数学实验

面向对象技术应用设计

信息安全综合实验