**东北大学计算机科学与工程学院**

**接口大作业报告**



**题目 ：GPS信息显示和基于MAX7219的8X8LED点阵**

**姓名 : 谢良彬**

**学号 : 20154518**

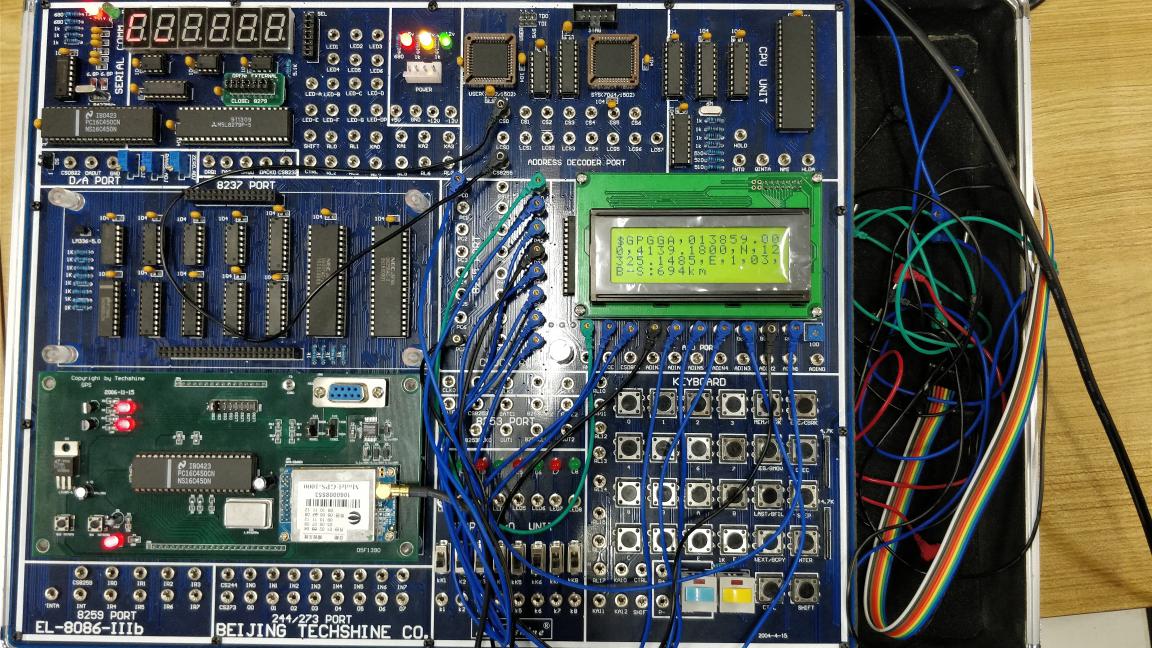
**日期：2018年1月11日**

**实验一（扩展实验）**

### 功能

GPS模块接收卫星的多颗卫星的定位信息，进行处理后按一定的格式输出串行数据，8050实现串并转换，通过总线与单片机通信。从GPGGA输出信息中提取实验中的数据，显示在液晶屏上。计算从北京到沈阳的距离并显示在屏幕上。同时可以判断GPS模块是否处于定位中状态。

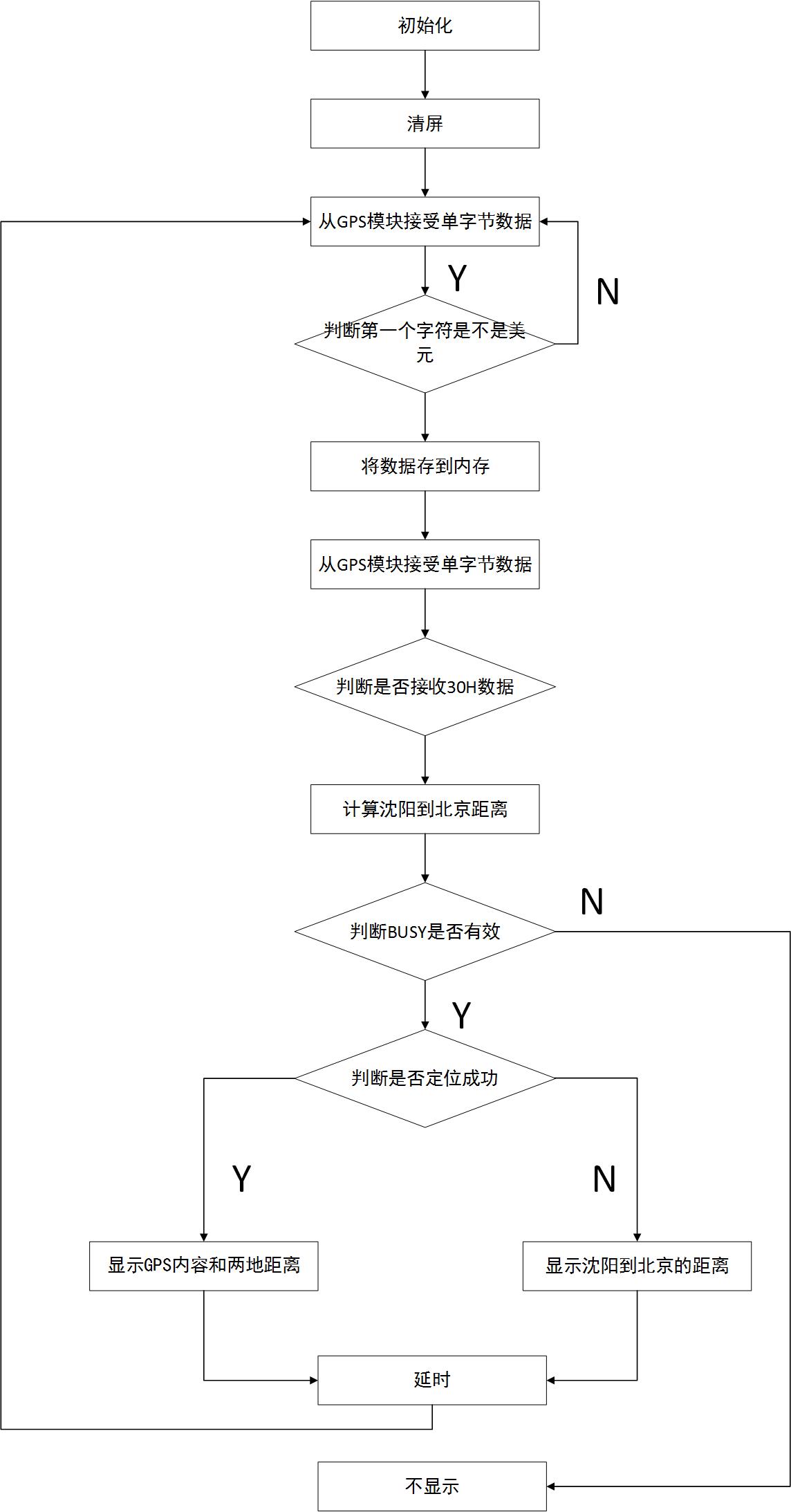
### 硬件电路设计



实验接线：CS0接CS8255,DB0-DB7接PA0-PA7,BUSY接PC7,REQ接PC0。模块上的J5(8250的片选)短接在CS1，TXSEL和RXSEL短接在TXDA侧。

编写程序，进行调试并全速运行。

### 程序流程图



### 程序代码

assume cs:code

code segment public

org 100h

start: call ini8250

call ini8255

mov al, 0f4h ;LCD清屏

call comd

mov ax,0100h

mov ds,ax ;数据段地址

mov es,ax

lp: mov si,1000h ;偏移地址

call recv

mov ah,al

mov cx,00h

xor al,'$' ;判断第一个是否为美元符号

jz lp1

jmp lp

lp1: mov al,ah

mov [si],al ;开始从内存单元的1000h处中取数

inc si

call recv

mov ah,al

cmp si,1030h ;和内存中的1030h作比较，即取48个字节

jnz lp1 ;如果没有达到48个字节，即再次进行循环操作

call dispb1

mov si,1000h

mov al, 0f4h ;LCD清屏

call comd

mov cx,1000h

call delay

mov si,102bh

mov al,[si]

cmp al, '1' ;判断是否定位

jz lp3

mov si, 1030h

;call dispb2

lp2: mov al,[si]

inc si

cmp si,1041h ;显示屏提供了64个字节

jz lp

call dispb

push cx

;mov cx,100h

mov cx,1000h ;改变刷新速度

call delay

pop cx ;保护现场

inc cl ;先对列进行输出

test cl,16

jz lp2

mov cl,0

add ch,8 ;先对行进行输出

test ch,32

jz lp2

mov ch,0

jmp lp2

lp3:

mov si, 1000h

ini8250: mov bx,04b0h ;8250的初始化

mov dx,bx

add dx,6

mov ax,80h

out dx,ax

mov dx,bx

mov ax,18h ;0018h---4800 ,clk=4.77MHZ/4

out dx,ax

add dx,2

mov ax,0h

out dx,ax

add dx,4

mov ax,07 ;no pe,8 bit, 1 stop

out dx,ax

mov dx,bx

add dx,2 ;no interupt

mov ax,0

out dx,ax

add dx,8h

in ax,dx

mov dx,bx

in ax,dx

ret

ini8255: mov dx, 04a6h ;8255的初始化

mov ax, 88h

out dx, ax

mov ax, 70h

out dx, ax

ret

recv: mov bx,04b0h ;接收GPS信号

mov dx,bx

add dx,0ah

in al,dx

test al,01h

jnz recv1

jmp recv

recv1: mov dx,bx

in al,dx

ret

dispb: mov ah,al ;输出一个字节

mov al,0f1h

call comd

mov al,cl

call comd

mov al,ch

call comd

mov al,ah

call comd

ret

dispb1: ;显示北京的经纬度

mov [si], '3'

inc si

mov [si], '9'

inc si

mov [si], 'N'

inc si

mov [si], ','

inc si

mov [si], '1'

inc si

mov [si], '1'

inc si

mov [si], '6'

inc si

mov [si], 'E'

inc si;

mov [si], ' '

inc si

mov [si], 'S'

inc si ;

mov [si], ':'

inc si

mov [si], '6'

inc si

mov [si], '9'

inc si

mov [si], '4'

inc si

mov [si], 'k'

inc si

mov [si], 'm'

inc si

ret

comd: mov dx, 04a0h ;往显示屏里传数据

out dx, al

mov dx, 04a6h

mov al, 71h

out dx, al

mon: mov dx, 04a4h ;监控BUSY，BUSY不停就不能往里传数据

in al, dx

and al, 80h

jz mon

mov dx, 04a6h

mov al, 70h

out dx ,al

ret

delay: nop ;空指令，什么都不做来表示延迟

loop delay

ret

code ends

end start

### 实现截图

显示北京到沈阳的距离：



显示GPS模块的经纬度并显示是否在定位中：



当BUSY是高电位的时候实现清屏：



### 六．心得体会

这个实验是和同学一起完成的，由于实验箱和之前所用的实验箱不一样，所以我们刚开始的时候用该实验箱重新做了接口的四个实验，熟悉了基本用法。之后，我们开始就GPS这个模块来完成相应的功能。GPS模块是一个已经封装好的模块，也就意味这其实我们只能对其的输出进行一定的控制，对其核心的内容我们是没有办法修改的。明确了这一点之后，我们开始对其输出的部分进行理解和修改。作为小组的负责人，如果对组内成员进行分工是一个比较重要的环节，其他两位同学我让他们对代码进行理解，我自己的话主要是就这个模块添加一些功能。

刚开始的时候，大家都看不懂这个最初的代码，总共4次的开放性实验课程，我们大约花了2节课才把这个代码搞得稍微明白了一些，知道其实这些数据就存在内存单元的1000h处，之后的30h就是模块的特定内容，显示屏可以给我们提供40h的内存长度，所以我们可以利用剩下的10h的长度进行我们的输出。SI是所谓的偏移地址，改变这个的值，我们就可以任意地在显示屏上输出我们想要的部分，即选择性输出。知道了工作方式之后，我们就将北京的经纬度和算出的北京到沈阳的距离写入内存单元之中，之后通过是否定位来控制我们的输出。

**实验二**

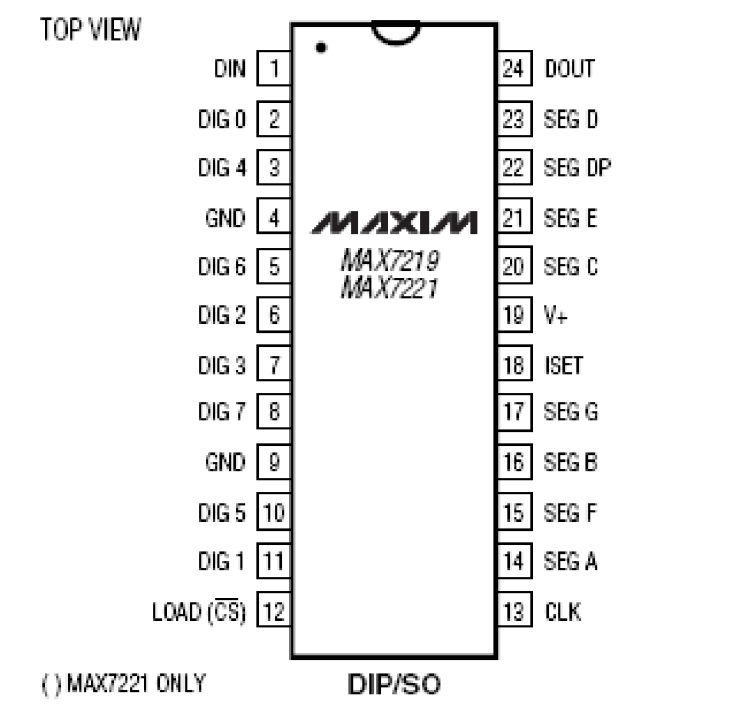
1. **功能：**

使用MAX7219实现点阵

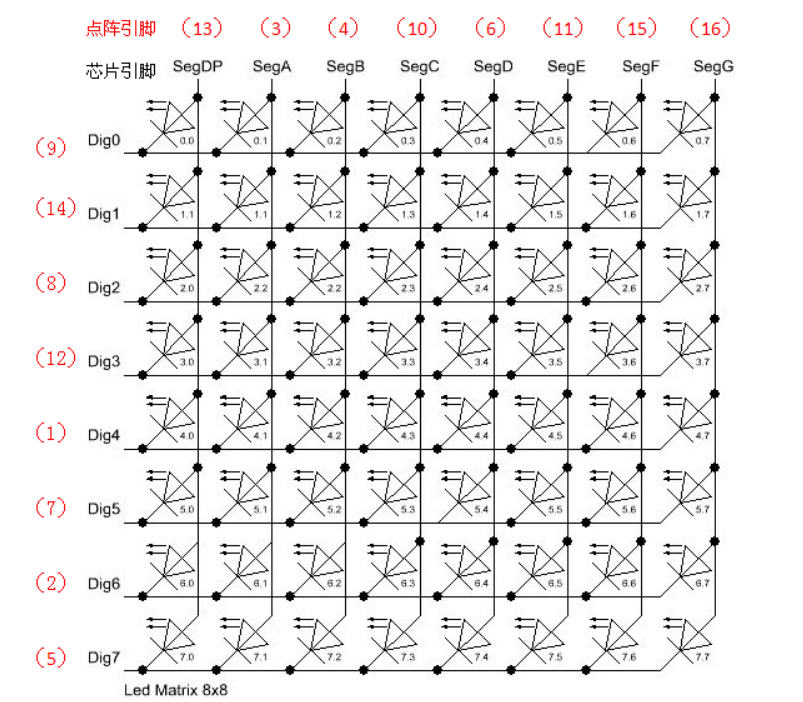
1. **原理：**

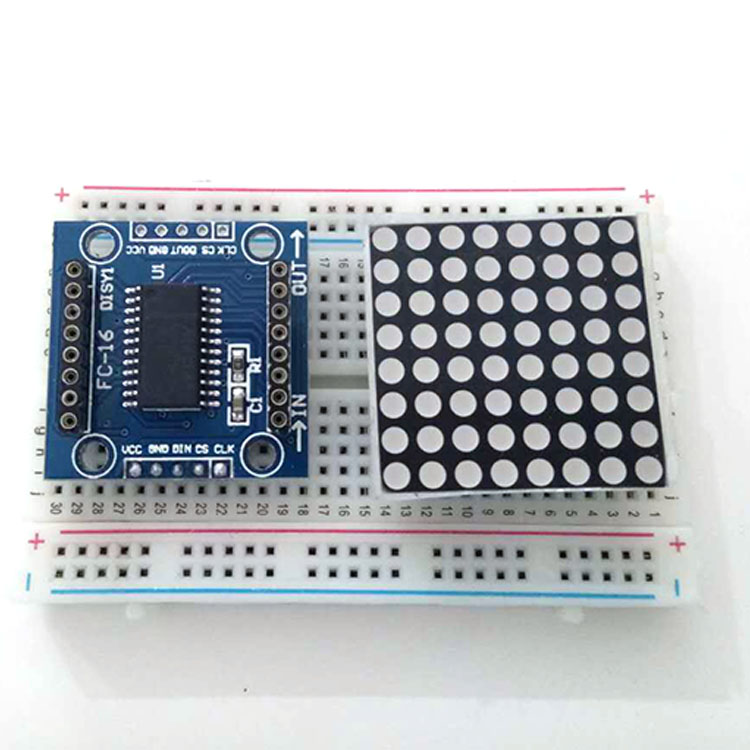
MAX7219/MAX7221是一种集成化的串行输入/输出共阴极显示驱动器,它连接微处理器与8位数字的7段数字LED显示，也可以连接条线图显示器或者64个独立的LED。其上包括一个片上的B型BCD编码器、多路扫描回路，段字驱动器，而且还有一个8\*8的静态RAM用来存储每一个数据。只有一个外部寄存器用来设置各个LED的段电流。MAX7221与SPI™、QSPI™以及MICROWIRE™相兼容，同时它有限制回转电流的段驱动来减少EMI（电磁干扰）。

一个方便的四线串行接口可以联接所有通用的微处理器。每个数据可以寻址在更新时不需要改写所有的显示。MAX7219/MAX7221同样允许用户对每一个数据选择编码或者不编码。整个设备包含一个150μA的低功耗关闭模式，模拟和数字亮度控制，一个扫描限制寄存器允许用户显示1-8位数据，还有一个让所有LED发光的检测模式。



1. **硬件电路设计：**





接线方式：

Arduino Uno --- MAX7219

5V <---> VCC

GND <---> GND

12 <---> DIN

11 <---> CS

10 <---> CLK

1. **程序：**
2. **Char\_lib\_for\_MAX7219.h**

**#ifndef CHAR\_LIB\_FOR\_MAX7219\_H\_**

**#define CHAR\_LIB\_FOR\_MAX7219\_H\_**

**#define uchar unsigned char**

**#include"Matrix\_8x8\_with\_MAX7219CNG.h"**

**extern void disch7219(char a);**

**#endif /\* CHAR\_LIB\_FOR\_MAX7219\_H\_ \*/**

1. **Char\_lib\_for\_MAX7219.c**

**#include"Char\_lib\_for\_MAX7219.h"**

**#include"Matrix\_8x8\_with\_MAX7219CNG.h"**

**uchar charlib[38][8]= //将特定的字符进行存储，可以输出0~9，A~Z,中，国**

**{**

**{0x3C,0x42,0x42,0x42,0x42,0x42,0x42,0x3C}, //0**

**{0x10,0x18,0x14,0x10,0x10,0x10,0x10,0x10}, //1**

**{0x7E,0x2,0x2,0x7E,0x40,0x40,0x40,0x7E}, //2**

**{0x3E,0x2,0x2,0x3E,0x2,0x2,0x3E,0x0}, //3**

**{0x8,0x18,0x28,0x48,0xFE,0x8,0x8,0x8}, //4**

**{0x3C,0x20,0x20,0x3C,0x4,0x4,0x3C,0x0}, //5**

**{0x3C,0x20,0x20,0x3C,0x24,0x24,0x3C,0x0}, //6**

**{0x3E,0x22,0x4,0x8,0x8,0x8,0x8,0x8}, //7**

**{0x0,0x3E,0x22,0x22,0x3E,0x22,0x22,0x3E}, //8**

**{0x3E,0x22,0x22,0x3E,0x2,0x2,0x2,0x3E}, //9**

**{0x8,0x14,0x22,0x3E,0x22,0x22,0x22,0x22}, //A**

**{0x3C,0x22,0x22,0x3E,0x22,0x22,0x3C,0x0}, //B**

**{0x3C,0x40,0x40,0x40,0x40,0x40,0x3C,0x0}, //C**

**{0x7C,0x42,0x42,0x42,0x42,0x42,0x7C,0x0}, //D**

**{0x7C,0x40,0x40,0x7C,0x40,0x40,0x40,0x7C}, //E**

**{0x7C,0x40,0x40,0x7C,0x40,0x40,0x40,0x40}, //F**

**{0x3C,0x40,0x40,0x40,0x40,0x44,0x44,0x3C}, //G**

**{0x44,0x44,0x44,0x7C,0x44,0x44,0x44,0x44}, //H**

**{0x7C,0x10,0x10,0x10,0x10,0x10,0x10,0x7C}, //I**

**{0x3C,0x8,0x8,0x8,0x8,0x8,0x48,0x30}, //J**

**{0x0,0x24,0x28,0x30,0x20,0x30,0x28,0x24}, //K**

**{0x40,0x40,0x40,0x40,0x40,0x40,0x40,0x7C}, //L**

**{0x81,0xC3,0xA5,0x99,0x81,0x81,0x81,0x81}, //M**

**{0x0,0x42,0x62,0x52,0x4A,0x46,0x42,0x0}, //N**

**{0x3C,0x42,0x42,0x42,0x42,0x42,0x42,0x3C}, //O**

**{0x3C,0x22,0x22,0x22,0x3C,0x20,0x20,0x20}, //P**

**{0x1C,0x22,0x22,0x22,0x22,0x26,0x22,0x1D}, //Q**

**{0x3C,0x22,0x22,0x22,0x3C,0x24,0x22,0x21}, //R**

**{0x0,0x1E,0x20,0x20,0x3E,0x2,0x2,0x3C}, //S**

**{0x0,0x3E,0x8,0x8,0x8,0x8,0x8,0x8}, //T**

**{0x42,0x42,0x42,0x42,0x42,0x42,0x22,0x1C}, //U**

**{0x42,0x42,0x42,0x42,0x42,0x42,0x24,0x18}, //V**

**{0x0,0x49,0x49,0x49,0x49,0x2A,0x1C,0x0}, //W**

**{0x0,0x41,0x22,0x14,0x8,0x14,0x22,0x41}, //X**

**{0x41,0x22,0x14,0x8,0x8,0x8,0x8,0x8}, //Y**

**{0x0,0x7F,0x2,0x4,0x8,0x10,0x20,0x7F}, //Z**

**{0x8,0x7F,0x49,0x49,0x7F,0x8,0x8,0x8}, //中**

**{0xFE,0xBA,0x92,0xBA,0x92,0x9A,0xBA,0xFE}, //国**

**};**

**void disch7219(char a) #显示ASCII字符**

**{**

**switch (a)**

**{**

**case '0': #显示0**

**display7219(charlib[0]);**

**break;**

**case '1': #显示1**

**display7219(charlib[1]);**

**break;**

**case '2': #显示2**

**display7219(charlib[2]);**

**break;**

**case '3': #显示3**

**display7219(charlib[3]);**

**break;**

**case '4': #显示4**

**display7219(charlib[4]);**

**break;**

**case '5': #显示5**

**display7219(charlib[5]);**

**break;**

**case '6': #显示6**

**display7219(charlib[6]);**

**break;**

**case '7': #显示7**

**display7219(charlib[7]);**

**break;**

**case '8': #显示8**

**display7219(charlib[8]);**

**break;**

**case '9': #显示9**

**display7219(charlib[9]);**

**break;**

**case 'A': #显示A**

**display7219(charlib[10]);**

**break;**

**case 'B': #显示B**

**display7219(charlib[11]);**

**break;**

**case 'C': #显示C**

**display7219(charlib[12]);**

**break;**

**case 'D': #显示D**

**display7219(charlib[13]);**

**break;**

**case 'E': #显示E**

**display7219(charlib[14]);**

**break;**

**case 'F': #显示F**

**display7219(charlib[15]);**

**break;**

**case 'G': #显示G**

**display7219(charlib[16]);**

**break;**

**case 'H': #显示H**

**display7219(charlib[17]);**

**break;**

**case 'I': #显示I**

**display7219(charlib[18]);**

**break;**

**case 'J': #显示J**

**display7219(charlib[19]);**

**break;**

**case 'K': #显示K**

**display7219(charlib[20]);**

**break;**

**case 'L': #显示L**

**display7219(charlib[21]);**

**break;**

**case 'M': #显示M**

**display7219(charlib[22]);**

**break;**

**case 'N': #显示N**

**display7219(charlib[23]);**

**break;**

**case 'O': #显示O**

**display7219(charlib[24]);**

**break;**

**case 'P': #显示P**

**display7219(charlib[25]);**

**break;**

**case 'Q': #显示Q**

**display7219(charlib[26]);**

**break;**

**case 'R': #显示R**

**display7219(charlib[27]);**

**break;**

**case 'S': #显示S**

**display7219(charlib[28]);**

**break;**

**case 'T': #显示T**

**display7219(charlib[29]);**

**break;**

**case 'U': #显示U**

**display7219(charlib[30]);**

**break;**

**case 'V': #显示V**

**display7219(charlib[31]);**

**break;**

**case 'W': #显示W**

**display7219(charlib[32]);**

**break;**

**case 'X': #显示X**

**display7219(charlib[33]);**

**break;**

**case 'Y': #显示Y**

**display7219(charlib[34]);**

**break;**

**case 'Z': #显示Z**

**display7219(charlib[35]);**

**break;**

**default:**

**display7219(charlib[36]);**

**\_delay\_cycles(1000000);**

**display7219(charlib[37]);**

**break;**

**}**

**}**

1. **Matrix\_8x8\_with\_MAX7219CNG.h**

**#ifndef MATRIX\_8X8\_WITH\_MAX7219CNG\_H\_**

**#define MATRIX\_8X8\_WITH\_MAX7219CNG\_H\_**

**//change the corresponding port and pin**

**//remember to change the port and pin in init7219()**

**#define pinCLK\_1 P2OUT |= BIT5 // CLKpin**

**#define pinCLK\_0 P2OUT &=~ BIT5**

**#define pinLOAD\_1 P2OUT |= BIT4 //LOADpin CSpin chip select**

**#define pinLOAD\_0 P2OUT &=~ BIT4**

**#define pinDIN\_1 P2OUT |= BIT3 //DINpin**

**#define pinDIN\_0 P2OUT &=~ BIT3**

**#define uchar unsigned char**

**extern void init7219();**

**extern void sendword7219(uchar addr,uchar dat);**

**extern void display7219(uchar \*a);**

**#endif /\* MATRIX\_8X8\_WITH\_MAX7219CNG\_H\_ \*/**

1. **Matrix\_8x8\_with\_MAX7219CNG.c**

**#include <msp430.h>**

**#include"Matrix\_8x8\_with\_MAX7219CNG.h"**

**void sendbyte (uchar dat)**

**{**

**uchar i,temp;**

**for (i=0;i<8;i++) //从低位递加**

**{**

**temp=dat&0x80; //取出dat里的最高位（实参数据位与0x80即10000000相与）**

**dat=dat<<1; 左移一位**

**if(temp)**

**pinDIN\_1; 时钟关**

**else**

**pinDIN\_0;**

**pinCLK\_0;**

**//\_delay\_cycles(50000);**

**pinCLK\_1;**

**}**

**}**

**void sendword7219 (uchar addr,uchar dat) #将字送入到7219中**

**{**

**pinLOAD\_0;**

**sendbyte (addr); //写入八位地址**

**sendbyte (dat); //写入八位数据**

**pinLOAD\_1;**

**}**

**void init7219() #7219的初始化**

**{**

**P2SEL&=~(BIT4+BIT5+BIT3);**

**P2DIR |=BIT4+BIT5+BIT3;**

**sendword7219 (0x0c,0x01); //0X0C为关断模式寄存器地址 //0x01为正常工作模式数据**

**sendword7219 (0x0a,0x09); //0X0a为亮度控制寄存器地址 //0x00为亮度控制数据0x00-0x0f 设置亮度**

**sendword7219 (0x0b,0x07); //0X0b为扫描界限寄存器地址 //0x07为八位模式数据 设置扫描界限**

**sendword7219 (0x09,0x00); //0X09为译码控制寄存器地址 //0x00为不译码模式数据//01为译码模式 设置译码模式**

**sendword7219 (0x0f,0x00); //0X0F为显示模式寄存器地址 //0x00为正常工作模式数据**

**}**

**void clear7219(void) #MAX7219清屏操作**

**{**

**uchar i;**

**for(i=8; i>0;i--)**

**sendword7219(i,0x00);**

**}**

**void display7219(uchar a[]) #显示图案**

**{**

**uchar i;**

**for(i=0;i<8;i++)**

**sendword7219(i+1,\*(a+i));**

**}**

1. **结果：**

文字说明：

我们首先对芯片进行初始化，设置MAX7219的一些基本参数，之后，通过写入数码管的编号与数码管显示数字来向MAX7219写入数据，之后调用disch7219函数来显示“A”。

图片：



1. **感想：**

这个项目主要是根据其他人的一些博客完成的，期间查了一些资料，主要是关于MAX7219这个模板的，网上大家的代码其实大同小异，原理都是一样的，明白原理之后，其实就是一个依葫芦画瓢的过程。在实现的过程中，由于我们的板子是已经集成好了的，所以只需要进行硬件连线即可，除去了很多麻烦。