现在各种广告牌不再是白底黑字了，也不再是单一的非电产品，而是用上了丰富多彩的 LED 电子产品，为城市的增添了一道靓丽的风景。而且它采用低电压扫描驱动，具有耗电少、使用寿命长、成本低、发光效率高、故障少、视角大、可视距离远、可靠耐用、组态灵活、安全、响应时间短、绿色环保、控制灵活、色彩丰富以及对室内外环境适应能力强等特点。近年来 LED 显示屏市场得到了迅猛的发展，已经广泛应用到银行、邮电、税务、机场、车站、证券市场及其它交易市场、医院、电力、海关、体育场等需要进行多种公告、宣传的场合。

因此，学习 LED 系统原理与工程技术很有必要。通过设计一个可显示文字与图形的 16*64 点阵控制器来学习和熟悉 LED 的使用。

## 1　系统的设计与分析

本文是通过设计一个可显示文字与图形的 16*64 点阵控制器电路来学习和熟悉 LED 的使用，LED 点阵控制器分为五个模块，系统框图(如图 1)，一是参数输入部分，例如温度、湿度、亮度等；二是字库部分，如 16 点阵、24 点阵或者 32 点；三是单片机控制部份；四是接口部分（如图 2），一般采用 08 接口，如果需要不同的接口类型，可以根据 08 接口为基础进行转接，这也是本系统采用 08 接口的原因之一；五是串口通讯部分（图 3），其中亮度、温度、时钟模块由于时间的原因没有完善，等待扩展。

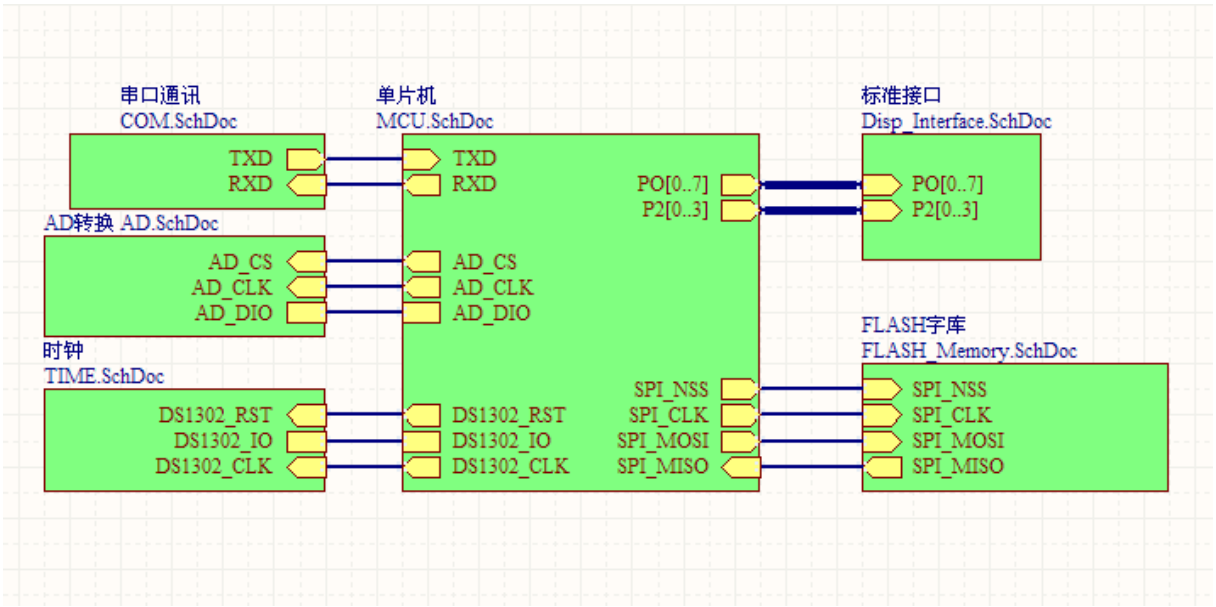



图 1　系统框图

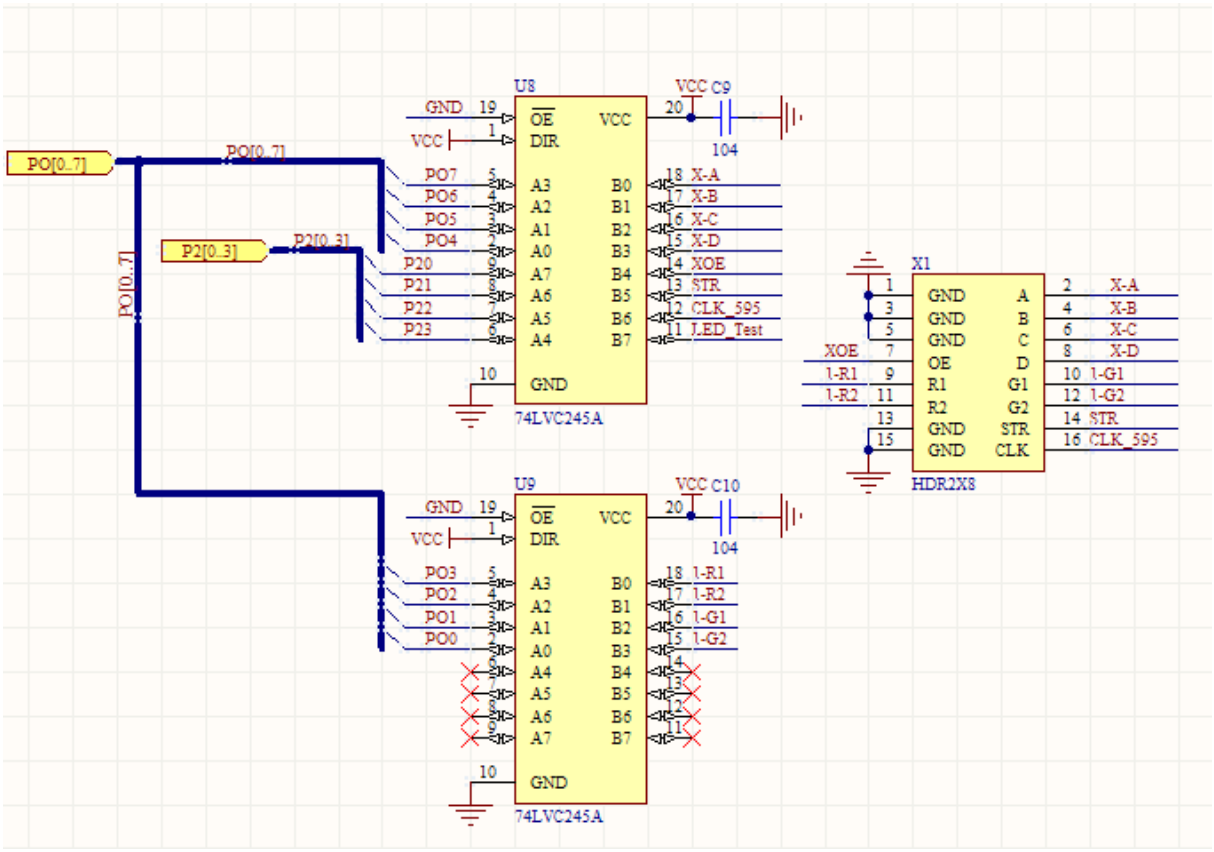



图 2 标准 08 接口部分

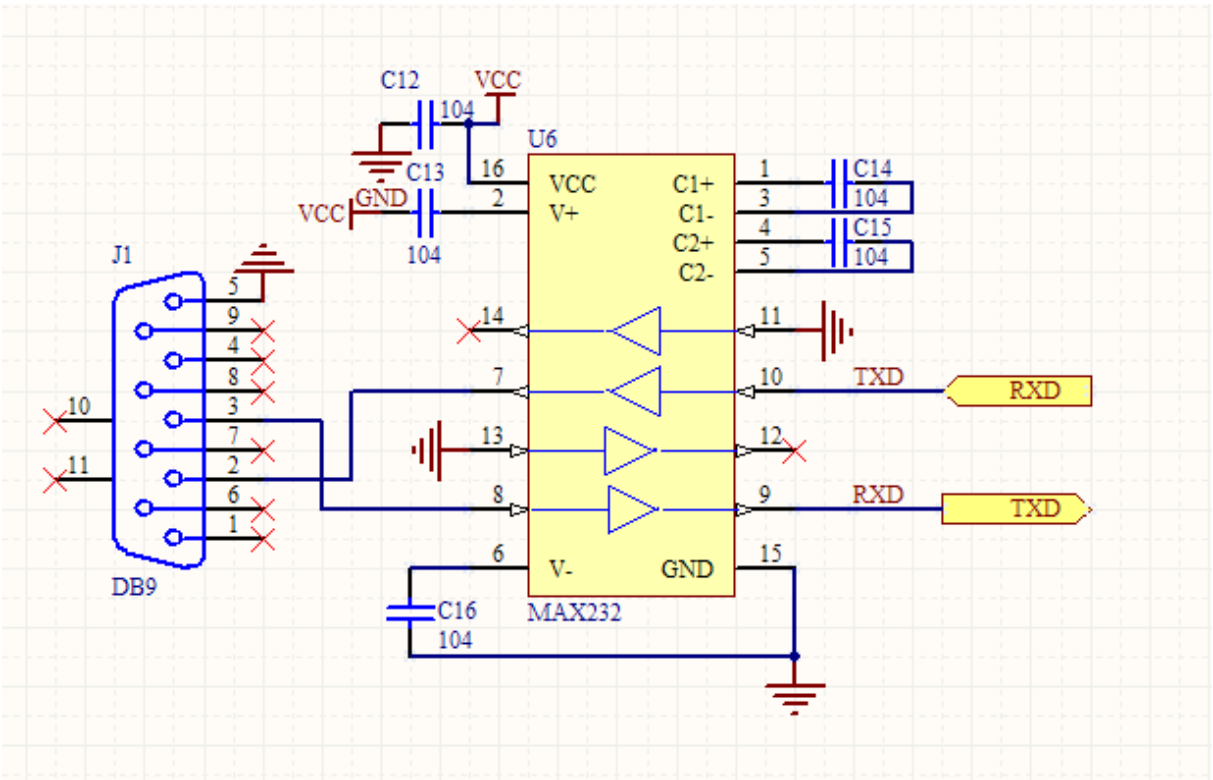



图 3 串口通讯部分

## 1.1 LED 点阵控制原理

显示屏是由发光二极管行列组成的 LED 点阵模块组成显示屏体。

### 1.1.1 LED 简介

LED 是发光二极管英文 Light Emitting Diode 的缩写格式，LED 器件种类繁多，早期的 LED 产品是单个发光管，随着数字化设备的出现，LED 数码管和字符管得到了广泛的应用，LED 点阵等显示器件的出现，适应了信息化社会发展的需要，成为了大众传媒的重要工具。

LED 发光灯按类型可以分为单色发光灯、双色发光灯、三色发光灯、面发光灯、闪烁发光灯、电压型发光灯等；按发光强度可分为普通亮度发光灯、高亮度发光灯、超高亮度发光灯等；

LED 发光灯结构如图 2 所示，它由芯片 3、阳极引脚 1、阴极引脚 2 和环氧树脂封装外壳四部分组成。它核心部分是具有复合发光功能的 PN 结，即芯片 3。环氧树脂封装外壳具有保护芯片的作用，还有透光聚光的能力，以增强显示效果。

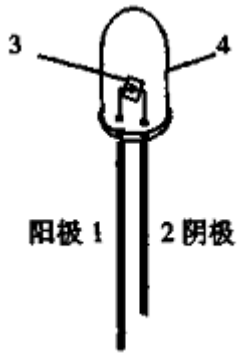



图 4

### 1.1.2 LED 点阵

随着 LED 应用领域的扩大，要求生产更为直接和方便的 LED 显示器件。因而出现了数码管、字符管、电平管、LED 点阵等多种 LED 显示器。不管显示器的结构怎么变，它的核心部件仍然是发光半导体芯片。

例如一个 8*8 的点阵是由 64 个发光二极管按一个规律组成的，如图 3。

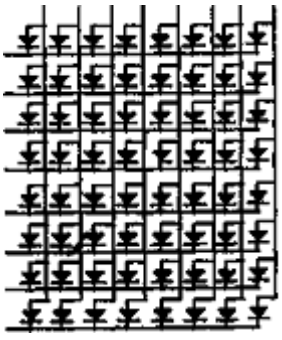


图 5

如图 3 所示的发光二极管，行接低电平，列接高电平，发光二极管导通发光。

### 1.1.3 显示原理

人眼的亮度感觉不会因光源的消失而立即消失，要有一个延迟时间，这就是视觉的惰性。视觉惰性可以理解为光线对人眼视觉的作用、传输、处理等过程都需要时间，因而使视觉具有一定的低通性。实验表明，当外界光源突然消失时，人眼的亮度感觉是按指数规律逐渐减小的。这样当一个光源反复通断，在通断频率较低时，人眼可以发现亮度的变化；而通断频率增高时，视觉就逐渐不能发现相应的亮度变化了。不致于引起闪烁感觉的最低反复通断频率称为临界闪烁频率。通过实验证明临界闪烁频率大约为24Hz。因此采用每秒 24 幅画面的电影，在人看起来就是连续活动的图象了。同样的原理，日光灯每秒通断 50 次，而人看起来却是一直亮的。由于视觉具有惰性，人们在观察高于临界闪烁频率的反复通断的光线时，所得到的主观亮度感受实际上是客观亮度的平均值。

视觉惰性可以说是 LED 显示屏得以广泛应用的生理基础。首先，在 LED 显示屏中可以利用视觉惰性，改善驱动电路的设计，形成了目前广为采用的扫描驱动方式。扫描驱动方式的优点在于 LED 显示屏不必对每个发光灯提供单独的驱动电路，而是若干个发光灯为一组共用一个驱动电路，通过扫描的方法，使各组发光灯依次点燃，只要扫描频率高于临界闪烁频率，人眼看起来各组灯都在发光。由于 LED 显示屏所使用的发光灯数量很大，一般在几千只到几十万只的范围，所以节约驱动电路的效益是十分可观的。

### 1.1.4 显示屏的原理图及结构

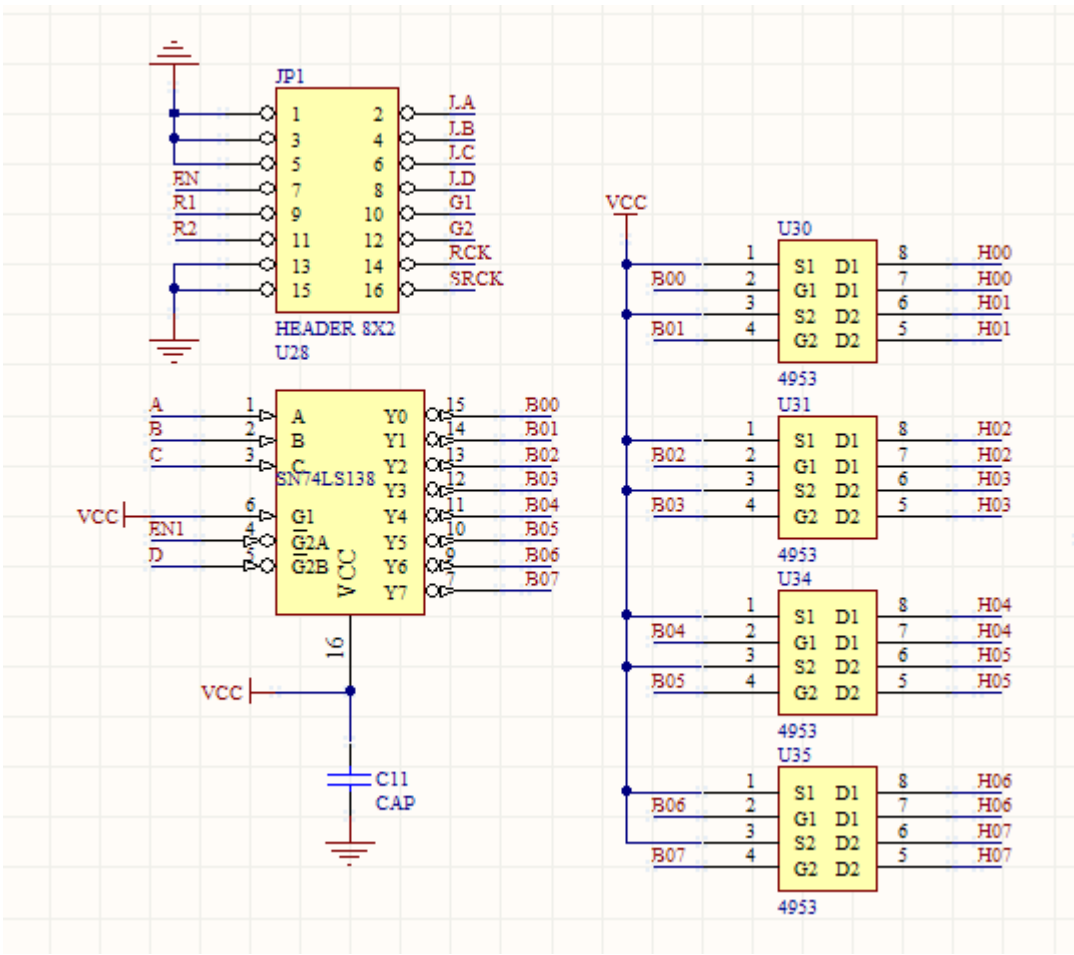



图 6 行扫描部分

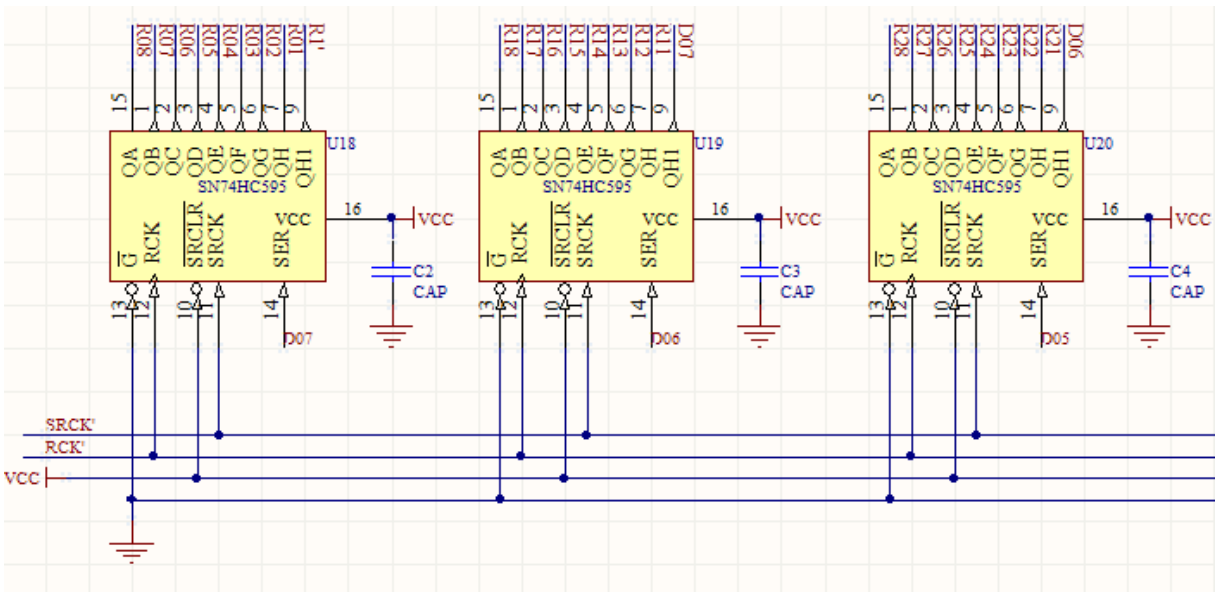



图 7 列扫描部分

根据显示屏的原理图结构，分析 LED 点阵控制器的控制原理：

如显示 10 个汉字，先将这 10 个汉字的点阵从字库中读出，放到显示缓存,如果要实现左移或者其它的显示效果则将显示缓存中的每个位进行移位或者其它处理,然后再调用扫描显示函数就可以实现所规定的效果。

扫描显示函数是显示缓存的内容，如果要实现不同的内容，比如说图片、汉字、英文等内容，只需要将这些内容按扫描显示函数对显示缓存的协议要求就可以显示出来。

由于显示屏中采用 74HC595 移位缓存器，所以需要做 74HC595 的驱动，这个相对简单，只要将数据按位传输，一位送一个时钟，送完一行所有的数据送一个锁存时钟，再通过 74HC138 选通该行，这样一直循环，人眼就会看到一幅完整的内容。

对于显示屏来说，显示使能端是比较重要的，主要是因为送完一行后需要一个消隐的动作，所谓的消隐就是让显示屏黑屏一段时间，如果不做该动作，则在显示的过程中会有拖影的现象。

## 1.3 单片机小系统

### 1.3.1 51 系列单片机的概述

单片机也被称作“单片机微型计算机”、“微控制器”、“嵌入式微控制器”，国际上采用“MCU”(Micro Controller Unit)称呼单片机。如果将 8 位单片机的推出作为起点（1976 年），那么单片机的发展的历史大致可以分为 4 个阶段。第一阶段是单片机探索阶段，主要探索如何把计算机的主要部件集成在单芯上；第二阶段是单片机完善阶段，完善了 8 位单片机的并行总线结构、外围功能单元由 CPU 集中管理模式、体现控制特性的位地址空间和位操作方式、指令系统趋于丰富和完善，并且增加了许多突出控制功能的指令；第三阶段是向微控制器发展的阶段，说的是在 51 系列的基本结构的基础上，加强了外围电路的功能，突出了单片机的控制功能，将一些用测控对象的模数转换器、数模转换器、程序运行监视器、脉宽调制器等纳入芯片中，体现单片机的微控制器特征；第四阶段是单片机的全面发展阶段，很多大半导体和电气厂商都开始加入单片机的研制和生产，单片机世界出现了百花齐放，欣欣向荣的景象。随着单片机在各个领域全面深入地发展和应用，出现了高速、大寻址范围、强运算能力的 8 位、16 位、32 位通用型单片机，以及小型谦价的专用型单片机。目前，单片机正朝着高性能和多品种方向发展，今后单片机的发展趋势将是进一步向着 CMOS 化、低功耗、小体积、大容量、高性能、低价格和外围电路内装等方面发展。

### 1.3.2 单片机的组成

图 7 是单片机典型组成框图，由图可见它通过内部总线把计算机的各主要部件连为一体，其内部总线包括地址总线、数据总线和控制总线。其中，地址总线的作用是为进行数据交换时提供地址，CPU 通过将地址输出到存储器或 I/O 接口；数据总线用于在 CPU 与存储器或 I/O 接口之间或存储器与外设之间交换数据；控制总路线包括 CPU 发出的控制信号线和外部送入 CPU 的应答线等。

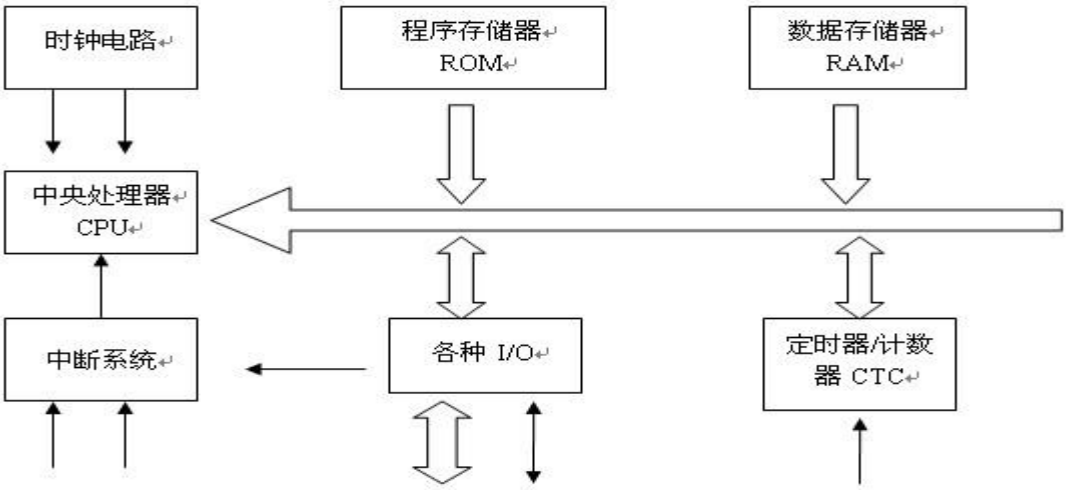



图 8 单片机结构框图

51 系列有 3 种封装形式，一种是 DIP（Dual Inline Package）封装形式，一种是 LCC（Quad Flat Package）封装形式。这种形式是具有 44 个“J”形脚的方型芯片。另一种是 QFP（Quad Flat Package）封装形式, 这种形式是具有 44 个“J”形脚的方型芯片,但它的体积更小、更薄，是一种表面贴焊的封装形式。下面介绍下 89S52 单片机的引脚的功能和其内部结构图。AT89S52 单片机实际有郊的引脚为 40 个，以下是 89S52 单片机的 DIP 封装形式的引脚的名称，如图 8。

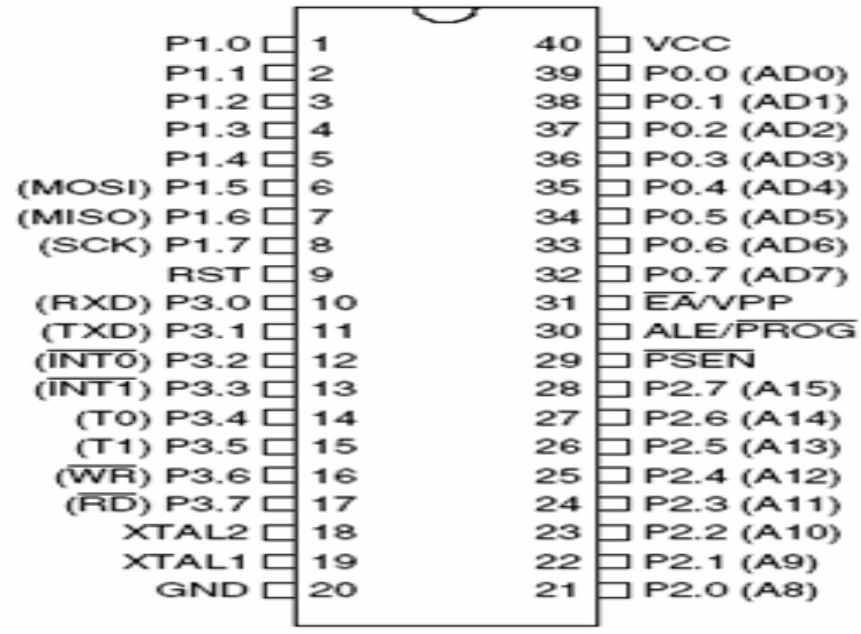



图 9 单片机的引脚说明

图 9 是 89S52 的内部结构图，由图可以看到在单片机内部除了有 CPU、RAM、ROM 和定时器、串行口等主要功能部件之外，还有驱动器、锁存器、指令寄存器、地址寄存器

等辅助电路部分，以及各功能模块在单片机中的位置和相互关系。

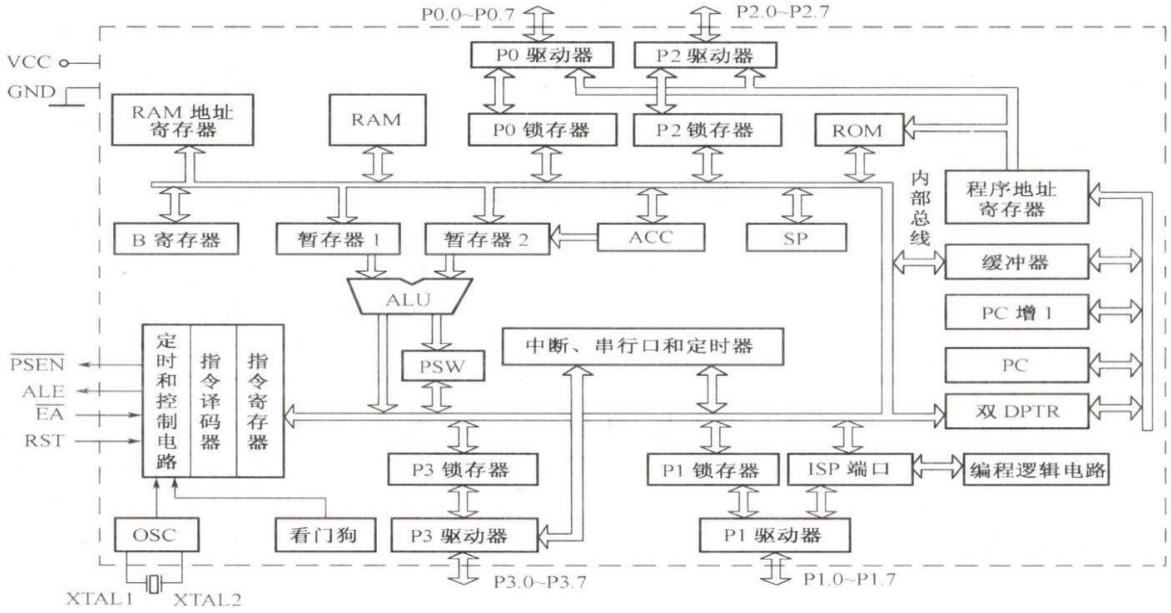



图 10　89S52 的内部结构

## 2　程序设计

### 2.1 程序的思路

根据点阵的显示原理就可以编写出显示函数，整个程序的设计分为串口通讯协议、扫描程序函数、效果处理函数。

#### 2.1.1 串口通讯协议

#p:为显示图片

#c:为显示汉字

#l:为显示英文

#n:为显示数字

所有的控制命令必须以回车结束，指令接收并解码正确则返回“Receive OK”，否则返回“error”

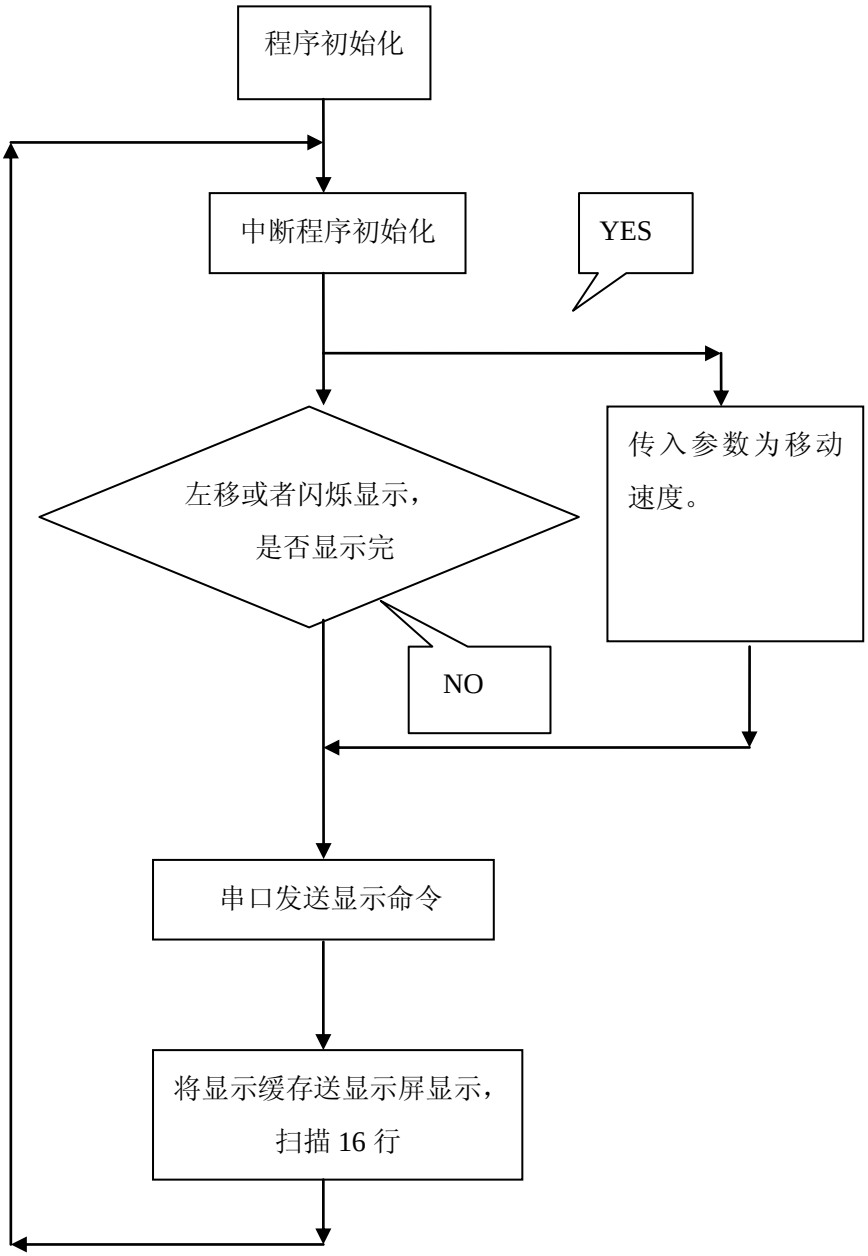
#### 2.1.2 显示处理效果函数

本系统只做了两个显示效果，一个是闪烁，一个是左移。闪烁相对简单，先显示延时，清屏延时，再显示。左移则相对复杂，需要将显示缓存的内容全部左移一点，左移的时候取前一个字节的低位等于后一个字节的低位，一屏移完一点后送显示，再移一点，这样循环就可以实现了左移的效果。

#### 2.1.3 显示函数

这个函数是将显示缓存的内容送到显示屏，做一个 74HC595 的驱动即可，送完一行数据后点亮，再送下一行，这样循环 16 次，就可以扫描完 16 行高的显示屏。

### 2.2 程序流程图


程序初始化
中断程序初始化
YES
传入参数为移动速度。
左移或者闪烁显示，是否显示完
NO
串口发送显示命令
将显示缓存送显示屏显示，扫描 16 行

主要程序见附录。

# 附　录

具体程序如下：

串口通讯部分：

```
/*************************************************
函数名称：Interrupt_serial()
传入参数：无
函数功能：串口通讯协议
*************************************************/
void Interrupt_serial() interrupt 4 using 1
{
    EA = 0;
    if(RI)
     {
        Serial_Int_temp[Receiv_Count]=SBUF;    //
        Receiv_Count++;
       if(Receiv_Count>=4)
        {
 if(Serial_Int_temp[Receiv_Count-2]==0x0d && Serial_Int_temp[Receiv_Count-1]==0x0a)//以回车结尾
              {
               Send_char(&Serial_Int_temp[0],Receiv_Count);
               if(Serial_Int_temp[0]=='#')    //指令格式
                {
                switch(Serial_Int_temp[1])
                   {
                    case 'p':
                             Send_char(&Serial_Recei_OK[0],10); //命令正确,反馈接收成功信息
                             break;
                    case 'n':
                             Send_char(&Serial_Recei_OK[0],10); //命令正确,反馈接收成功信息
                             break;
                    case 'l':
                             Send_char(&Serial_Recei_OK[0],10); //命令正确,反馈接收成功信息
                             break;
                    case 'c':
                             Send_char(&Serial_Recei_OK[0],10); //命令正确,反馈接收成功信息
```

```
                    break;
                default: Send_char(&Serial_Error[0],7); //指令错误
                    break;
                }
            scan_mod=Serial_Int_temp[1];     //显示内容
            cldispb(); //清缓存
            }
            else {Send_char(&Serial_Error[0],7);} //指令错误
          }
        else
          {
            Send_char(&Serial_Error[0],7);     //指令错误
          }
        Receiv_Count=0;//接收数据字节计数器
      }
      if(Receiv_Count>=39)Receiv_Count=0;
    }
  RI=0;     //清接收标志
  EA = 1; //开中断
}

/*************************************************
函数名称：display1p()
传入参数：无
函数功能：扫描显示屏
*************************************************/
void display1p()
{
  uchar temp,j,k,i;
  unsigned char *point,*point1;
  uchar hangxu=1;
  point1=&disp_buf[0];
  point = point1;

 for(j=0;j<16;j++)
```

```
    {
        for(k=0;k<plong;k++)
        {
            temp=*point;
            for(i=0;i<8;i++)
                {
                  CLK=0;
                  R1=1;
                 if((temp&0x80)==0x00)
                   {
                    R1=0;
                   }
                    CLK=1;
                  temp<<=1;
                }
            point++;
        }

        OE=0;
        STR=0;
        STR=1;
        STR = 0;
        Showline(j);
        OE=1;
        ddlay(9);
        OE=0;
      point=point1+plong*(j);
    }
}
/************************************************
函数名称：moveleft(unsigned char sspp)
传入参数: unsigned char sspp  移动速度
函数功能：左移显示效果
************************************************/
void moveleft(unsigned char sspp)
```

```
{
    uchar i,j,k;
    switch(scan_temp)
    {
      case 'p':
                photo_todisp();
                break;
      case 'l':
                letter_todisp();
                break;
      case 'n':
                num_todisp();
                break;
      case 'c':
                dztodisp();
                break;
      default: dztodisp();
                break;
     }
    for(i=0;i<plong;i++)            //屏的长度
    {
        for(j=0;j<8;j++)
        {
            leftoned();                 //左移 1 点
              for(k=0;k<sspp;k++)
              {
              display1p();
              if(scan_mod != scan_temp){scan_temp=scan_mod;goto exit;} //是否接收到命令
              }

        }
    }
exit:;          //退出标记
   cldispb();   //清显示缓存
}
```

# 完整源程序

**1，main.c　主函数**

```
/*********************************************************************
文件清单:
main.c          主函数
code_area.c  字库表
display.c      显示相关函数
com.c           串口通信相关函数

CPU 含 256B RAM

现分配如下：

00H -- 07FH    程序常规使用          128B
7FH -- FFH      显示缓存                128B
*********************************************************************/

#include <reg51.h>
#include <intrins.h>
#include <display.h>
#include <com.h>
#include <main.h>
#define uint unsigned int
#define uchar unsigned char
uchar Serial_Int_temp[5];
uchar Receiv_Count=0;    //串口接收字节计数
uchar scan_mod;

sbit R1= P0^0 ;
sbit R2= P0^1;
sbit G1= P0^2;
sbit G2= P0^3;
sbit LL1 = P0^4;
sbit LL2= P0^5;
sbit LL3= P0^6;
```

```
sbit LL4= P0^7;
sbit OE= P2^3 ;
sbit STR= P2^2;
sbit CLK= P2^1;

void main()//主函数
{
      //SetTime( 0x12 , 0x23 ,0x50 );
      SP = 0X30;
      Init_com();
      Send_char(&Serial_Strar_inf[0],27);
      //onedisp(20);
      while(1)
      {
          cldispb();
         moveleft(20);
         flicker(5,5);
      /*    GetTime();   //读 DS1302
          temp= ADC0832();
       */
      }
}

/************************************************
函数名称：Interrupt_serial()
传入参数: 无
函数功能：串口通讯协议
************************************************/
void Interrupt_serial() interrupt 4 using 1
{
    EA = 0;
    if(RI)
      {
         Serial_Int_temp[Receiv_Count]=SBUF;   //
            Receiv_Count++;
```

```
if(Receiv_Count>=4)
 {
  if(Serial_Int_temp[Receiv_Count-2]==0x0d && Serial_Int_temp[Receiv_Count-1]==0x0a)
      {
        Send_char(&Serial_Int_temp[0],Receiv_Count);
        if(Serial_Int_temp[0]=='#')   //指令格式
            {
              switch(Serial_Int_temp[1])
               {
                 case 'p':
                           Send_char(&Serial_Recei_OK[0],10); //命令正确,反馈接收成功信息
                           break;
                 case 'n':
                           Send_char(&Serial_Recei_OK[0],10); //命令正确,反馈接收成功信息
                           break;
                 case 'l':
                           Send_char(&Serial_Recei_OK[0],10); //命令正确,反馈接收成功信息
                           break;
                 case 'c':
                           Send_char(&Serial_Recei_OK[0],10); //命令正确,反馈接收成功信息
                           break;
                 default: Send_char(&Serial_Error[0],7); //指令错误
                           break;
               }
              scan_mod=Serial_Int_temp[1];     //显示内容
            cldispb();
            }
         else {Send_char(&Serial_Error[0],7);} //指令错误
      }
   else
    {
        Send_char(&Serial_Error[0],7);     //指令错误
     }
    Receiv_Count=0;
  }
```

```
            if(Receiv_Count>=39)Receiv_Count=0;
        }
    RI=0;
    EA = 1;
}
```

Main.c 所用到的头文件源代码:

**(1)，display.h**

```
#define __DISPLAY_H
#include <reg51.h>
#define uint unsigned int
#define uchar unsigned char
//extern void display1p();//显示函数
extern void process_disbuf(unsigned char char_num);
extern void onedisp(unsigned char ttime);
extern void display1p();
extern void moveleft(unsigned char sspp);
extern void cldispb(void);              //清显示缓存区
extern void flicker(unsigned char sspeed,unsigned char stop);//闪烁显示
//extern void display(uint disp_long,uchar disp_times,uchar effect);
//extern void display();
//extern unsigned char xdata disp_buf[128];
//extern unsigned char    disp_buf[32];
```

**(2)，com.h**

```
#define __COM_H
#include <reg51.h>
#define uint unsigned int
#define uchar unsigned char
extern void Send_char(uchar *Data_addr,uchar char_num);
extern unsigned char code Serial_data[42];
extern unsigned char code Serial_Recei_OK[10];
extern unsigned char code Serial_Error[7];
extern unsigned char code Serial_Strar_inf[27];
extern void Init_com();
```

**(3)，main.h**

```
#define __MAIN_H
#include <reg51.h>
#define uint unsigned int
#define uchar unsigned char
extern uchar scan_mod;
```

**2，code_area.c　　LED 点阵的字库**

```
//英文的 ASCII 为 97-122(十进制) 61-7A(ASCII 码)
#include <code_area.h>
unsigned char code num_dotmatrix[10][16]={
/*--  文字: 0 --*/
/*--  宋体 12;  此字体下对应的点阵为：宽 x 高=8x16 --*/
0xFF,0xFF,0xFF,0xE7,0xDB,0xBD,0xBD,0xBD,0xBD,0xBD,0xBD,0xDB,0xE7,0xFF,0xFF,/*"0",0
*/
0xFF,0xFF,0xFF,0xEF,0x8F,0xEF,0xEF,0xEF,0xEF,0xEF,0xEF,0xEF,0x83,0xFF,0xFF,/*"1",1*/
0xFF,0xFF,0xFF,0xC3,0xBD,0xBD,0xBD,0xFB,0xF7,0xEF,0xDF,0xBD,0x81,0xFF,0xFF,/*"2",2*/
0xFF,0xFF,0xFF,0xC3,0xBD,0xBD,0xFB,0xE7,0xFB,0xFD,0xBD,0xBB,0xC7,0xFF,0xFF,/*"3",3*/
0xFF,0xFF,0xFF,0xFB,0xF3,0xEB,0xDB,0xDB,0xBB,0xBB,0x81,0xFB,0xFB,0xE1,0xFF,0xFF,/*"4",4*/
0xFF,0xFF,0xFF,0x81,0xBF,0xBF,0xBF,0xA7,0x9B,0xFD,0xFD,0xBD,0xBB,0xC7,0xFF,0xFF,/*"5",5*/
0xFF,0xFF,0xFF,0xE3,0xDB,0xBF,0xBF,0xA7,0x9B,0xBD,0xBD,0xBD,0xDB,0xE7,0xFF,0xFF,/*"6",6*/
0xFF,0xFF,0xFF,0x81,0xBB,0xBB,0xF7,0xF7,0xEF,0xEF,0xEF,0xEF,0xEF,0xEF,0xFF,0xFF,/*"7",7*/
0xFF,0xFF,0xFF,0xC3,0xBD,0xBD,0xBD,0xDB,0xE7,0xDB,0xBD,0xBD,0xBD,0xC3,0xFF,0xFF,/*"8",8
*/
0xFF,0xFF,0xFF,0xE7,0xDB,0xBD,0xBD,0xBD,0xD9,0xE5,0xFD,0xFD,0xDB,0xC7,0xFF,0xFF/*"9",9*/
};
unsigned char code picture_code[128]={
/*--图片--*/
/*--宽度 16*64*/
0xFF,0xFF,0xFF,0xE0,0x7F,0xFE,0x03,0xFF,0xF8,0x01,0xFF,0xCF,0x3F,0xFC,0xF8,0x7F,
0xF3,0xFC,0xFF,0xBF,0x9F,0xF3,0xFF,0x3F,0xE6,0x66,0x3E,0x7F,0xCF,0xF5,0xAD,0xDF,
0xEC,0x63,0xBE,0xC4,0x27,0xEC,0x21,0xCF,0xE9,0x5B,0xBC,0xFF,0xF3,0xCE,0x73,0xEF,
0xEF,0xFF,0xBD,0xFF,0xFB,0xDF,0xFF,0xEF,0xEF,0xFF,0xBD,0xFF,0xF3,0xDF,0xFF,0xEF,
0xEE,0xF7,0xBC,0xE0,0x77,0xCC,0xFD,0x9F,0xEE,0x67,0xBE,0xFF,0xF7,0xEE,0xF9,0x9F,
0xEF,0x0F,0x3E,0x7F,0xF7,0xEE,0x03,0x3F,0xE7,0xFF,0x7F,0x3F,0xC7,0xF7,0xFF,0x7F,
```

```
0xF7,0xFE,0x7F,0xBF,0x8F,0xF1,0xFE,0x7F,0xF3,0xFC,0xFF,0x9F,0xBF,0xFC,0xFC,0xFF,
0xF8,0x01,0xFF,0xC0,0x7F,0xFF,0x01,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,/*"未命名文件
",0*/
};
unsigned char code dotmatrix_chinese[8][32]=
{
0xDF,0xBF,0xDF,0xBF,0xDF,0xBF,0xDC,0x03,0x03,0xBF,0xDF,0xBF,0x88,0x01,0x97,0xFF,
0x97,0xBF,0x5F,0xBF,0xDC,0x03,0xDF,0xBF,0xDF,0xBF,0xDF,0xBF,0xD0,0x01,0xDF,0xFF,/*"桂",0*/
0xEF,0xDF,0xEF,0xDF,0xEF,0xDF,0xEF,0xDF,0x02,0x01,0xEF,0xDF,0xEF,0x9F,0xC7,0x8F,
0xCB,0x4F,0xAB,0x57,0xAE,0xD7,0x6D,0xD9,0xEB,0xDB,0xEF,0xDF,0xEF,0xDF,0xEF,0xDF,/*"  林
",1*/
0xFE,0xFF,0xFE,0xFF,0xFE,0xFF,0xC0,0x07,0xDE,0xF7,0xDE,0xF7,0xC0,0x07,0xDE,0xF7,
0xDE,0xF7,0xDE,0xF7,0xC0,0x07,0xDE,0xF7,0xFE,0xFD,0xFE,0xFD,0xFF,0x01,0xFF,0xFF,/*"电",2*/
0xFF,0xFF,0xC0,0x0F,0xFF,0xDF,0xFF,0xBF,0xFF,0x7F,0xFE,0xFF,0xFE,0xFF,0xFE,0xFB,
0x00,0x01,0xFE,0xFF,0xFE,0xFF,0xFE,0xFF,0xFE,0xFF,0xFE,0xFF,0xFA,0xFF,0xFD,0xFF,/*"子",3*/
0xF9,0xF7,0x87,0x77,0xF7,0xB7,0xF7,0xB7,0x01,0xF7,0xE7,0x77,0xE3,0xB7,0xD5,0xB7,
0xD7,0xF1,0xB7,0x87,0x74,0x77,0xF7,0xF7,0xF7,0xF7,0xF7,0xF7,0xF7,0xF7,0xF7,0xF7,0xF7,/*"科",4*/
0xEF,0xDF,0xEF,0xDF,0xEF,0xDF,0x02,0x01,0xEF,0xDF,0xEB,0xDF,0xE6,0x03,0xCE,0xF7,
0x2F,0x77,0xEF,0x6F,0xEF,0x9F,0xEF,0x9F,0xEF,0x6F,0xEE,0xF1,0xA9,0xFB,0xDF,0xFF,/*"技",5*/
0xFE,0xFF,0xFE,0xFF,0xFE,0xFF,0xFE,0xFF,0xFE,0xFF,0x00,0x01,0xFE,0xFF,0xFD,0x7F,
0xFD,0x7F,0xFD,0xBF,0xFB,0xBF,0xFB,0xDF,0xF7,0xEF,0xEF,0xE7,0xDF,0xF1,0xBF,0xFB,/*"大",6*/
0xFE,0xF7,0xEF,0x73,0xF3,0x37,0xF7,0x6F,0x80,0x01,0xBF,0xFB,0x70,0x17,0xFF,0xBF,
0xFF,0x7F,0x80,0x01,0xFF,0x7F,0xFF,0x7F,0xFF,0x7F,0xFF,0x7F,0xFD,0x7F,0xFE,0xFF/*"学",7*/
} ;
unsigned char code letter_matrix[26][16]=
{
0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xC3,0xBD,0xE1,0xDD,0xBD,0xBD,0xC0,0xFF,0xFF,/*"a",0*/
0xFF,0xFF,0xFF,0x3F,0xBF,0xBF,0xBF,0xA7,0x9B,0xBD,0xBD,0xBD,0x9B,0xA7,0xFF,0xFF,/*"b",1*/
0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xE3,0xDD,0xBF,0xBF,0xBF,0xDD,0xE3,0xFF,0xFF,/*"c",2*/
0xFF,0xFF,0xFF,0xF9,0xFD,0xFD,0xFD,0xE1,0xDD,0xBD,0xBD,0xBD,0xD9,0xE4,0xFF,0xFF,/*"d",3*/
0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xC3,0xBD,0x81,0xBF,0xBF,0xBD,0xC3,0xFF,0xFF,/*"e",4*/
0xFF,0xFF,0xFF,0xF0,0xEE,0xEF,0xEF,0x81,0xEF,0xEF,0xEF,0xEF,0xEF,0x83,0xFF,0xFF,/*"f",5*/
0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xC1,0xBB,0xBB,0xC7,0xBF,0xC3,0xBD,0xBD,0xC3,/*"g",6*/
0xFF,0xFF,0xFF,0x3F,0xBF,0xBF,0xBF,0xA3,0x9D,0xBD,0xBD,0xBD,0xBD,0x18,0xFF,0xFF,/*"h",7*/
0xFF,0xFF,0xFF,0xCF,0xCF,0xFF,0xFF,0x8F,0xEF,0xEF,0xEF,0xEF,0xEF,0x83,0xFF,0xFF,/*"i",8*/
```

```
0xFF,0xFF,0xFF,0xF3,0xF3,0xFF,0xFF,0xE3,0xFB,0xFB,0xFB,0xFB,0xFB,0xBB,0x87,/*"j",9*/
0xFF,0xFF,0xFF,0x3F,0xBF,0xBF,0xBF,0xB1,0xB7,0xAF,0x97,0xB7,0xBB,0x11,0xFF,0xFF,/*"k",10*/
0xFF,0xFF,0xFF,0x8F,0xEF,0xEF,0xEF,0xEF,0xEF,0xEF,0xEF,0xEF,0xEF,0x83,0xFF,0xFF,/*"l",11*/
0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0x01,0xB6,0xB6,0xB6,0xB6,0xB6,0x12,0xFF,0xFF,/*"m",12*/
0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0x23,0x9D,0xBD,0xBD,0xBD,0xBD,0x18,0xFF,0xFF,/*"n",13*/
0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xC3,0xBD,0xBD,0xBD,0xBD,0xBD,0xC3,0xFF,0xFF,/*"o",14*/
0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0x27,0x9B,0xBD,0xBD,0xBD,0xBB,0x87,0xBF,0x1F,/*"p",15*/
0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xE1,0xDD,0xBD,0xBD,0xBD,0xDD,0xE1,0xFD,0xF8,/*"q",16*/
0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0x11,0xCD,0xDF,0xDF,0xDF,0xDF,0x07,0xFF,0xFF,/*"r",17*/
0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xC1,0xBD,0xBF,0xC3,0xFD,0xBD,0x83,0xFF,0xFF,/*"s",18*/
0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xEF,0xEF,0x83,0xEF,0xEF,0xEF,0xEF,0xEF,0xF3,0xFF,0xFF,/*"t",19*/
0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0x39,0xBD,0xBD,0xBD,0xBD,0xB9,0xC4,0xFF,0xFF,/*"u",20*/
0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0x18,0xBD,0xDB,0xDB,0xD7,0xEF,0xEF,0xFF,0xFF,/*"v",21*/
0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0x28,0x6D,0x6D,0x55,0x55,0xBB,0xBB,0xFF,0xFF,/*"w",22*/
0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0x91,0xDB,0xE7,0xE7,0xE7,0xDB,0x89,0xFF,0xFF,/*"x",23*/
0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0x18,0xBD,0xDB,0xDB,0xD7,0xE7,0xEF,0xEF,0x1F,/*"y",24*/
0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0x81,0xBB,0xF7,0xEF,0xEF,0xDD,0x81,0xFF,0xFF /*"z",25*/
};
```

**code_area.c 所用到的头文件源代码：**

**(1)，code_area.h**

```
#define __CODE_AREA_H
#include <reg51.h>
#define uint unsigned int
#define uchar unsigned char
extern unsigned char code num_dotmatrix[10][16];
extern unsigned char code dotmatrix_chinese[8][32];

extern unsigned char code picture_code[128];
extern unsigned char code letter_matrix[26][16];
//extern void display(unsigned char char_num);//显示函数
//extern unsigned char xdata disp_buf[128];
//extern unsigned char xdata    disp_buf[64];
```

## 关键的显示子程序源代码：

### 3，display.c

```
/*******************************************************************************
 *文件：display.c
 *功能：处理缓存及显示
*******************************************************************************/
#include <reg51.h>
#include <intrins.h>
#include <display.h>
#include <code_area.h>
#include <main.h>

#define dispbuf_mul 16       //显示屏行数
#define plong 8     //屏宽为 8*8=64
uchar scan_temp;//显示内容标志

sbit R1= P0^0 ;
sbit R2= P0^1;
sbit G1= P0^2;
sbit G2= P0^3;
sbit LL1 = P0^4;
sbit LL2= P0^5;
sbit LL3= P0^6;
sbit LL4= P0^7;
sbit OE= P2^3 ;
sbit STR= P2^2;
sbit CLK= P2^1;

unsigned char xdata disp_buf[plong*dispbuf_mul];       //显示屏宽度的单倍显示缓存
void dztodisp(void);                  //汉字数据到显示缓存区
void photo_todisp(void);              //图片到缓存
void letter_todisp(void);            //字母到缓存
void num_todisp(void);                //数据到显示缓存区
void Showline(uchar line_num);    //行扫
void ddlay(uchar ms);       //延时
```

```
void display1p();//显示函数
void moveleft(unsigned char sspp);    //左移显示效果
void leftoned();//左移一点
void flicker(unsigned char sspeed,unsigned char stop);//闪烁显示

/************************************************
函数名称：cldispb()
传入参数：无
函数功能：清显示缓存区
************************************************/
void cldispb(void)          //清显示缓存区
{
    uchar *disbuff;
    uint i;
    disbuff=&disp_buf[0];
    for(i=0;i<plong*dispbuf_mul;i++)
    {
        *disbuff=0xff;
        disbuff++;
    }
}
/************************************************
函数名称：onedisp()
传入参数：unsigned char ttime  显示时长
函数功能：立即显示
************************************************/
void onedisp(unsigned char ttime)
{
    uint i,stopt;
    dztodisp();                          //计算缓存直接送入显示缓存
    stopt=ttime*50;                 //显示时间
    for(i=0;i<stopt;i++)
    {
      display1p();                       //调用扫描子程序
    }
```

```
   cldispb();
// display1p();
}

/*****************************************************************
函数名称：flicker()
传入参数：unsigned char sspeed, ttime  闪烁速度及持续时间
函数功能：立即显示
*****************************************************************/
void flicker(unsigned char sspeed,unsigned char stop)//闪烁显示
{
      uint i,kk,stopt,sped;
      uchar j;
      sped=sspeed*10;
      stopt=stop*10;
      for(j=0;j<4;j++)
      {
            switch(scan_temp)
                  {
                     case 'p':
                                 photo_todisp();
                                 break;
                     case 'l':
                                 letter_todisp();
                                 break;
                     case 'n':
                                 num_todisp();
                                 break;
                     case 'c':
                                 dztodisp();
                                 break;
                     default: dztodisp();
                                 break;
                   }
            for(i=0;i<sped;i++)
```

```
        {
            display1p();
        }
        cldispb();
        for(i=0;i<sped;i++)
        {
            display1p();
        }

    }
        switch(scan_temp)
            {
              case 'p':
                        photo_todisp();
                        break;
              case 'l':
                        letter_todisp();
                        break;
              case 'n':
                        num_todisp();
                        break;
              case 'c':
                        dztodisp();
                        break;
              default: dztodisp();
                        break;
             }
    for(kk=0;kk<stopt;kk++)
    {
        display1p();
    }

}

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

```
函数名称：photo_todisp()
传入参数：
函数功能：图片数据到显示缓存区
************************************************/
void photo_todisp(void)              //数据到显示缓存区
{
   unsigned char i,k;
   unsigned char *point,*disp_temp;
   point=&picture_code[0];          //汉字库首地址
   disp_temp=&disp_buf[0];                  //取计算缓存首地址
 for(i=0;i<16;i++)                          //行 16
 {
          for(k=0;k<plong;k++)
        {
            *disp_temp=*point;
             disp_temp++;
             point++;
        }
   }
}

/************************************************
函数名称：dztodisp()
传入参数：
函数功能：汉字数据到显示缓存区
************************************************/
void dztodisp(void)              //数据到显示缓存区
{
   unsigned char i,k;
   unsigned char *point,*temp1,*disp_temp;
   point=&dotmatrix_chinese[0][0];          //汉字库首地址
   disp_temp=&disp_buf[0];                  //取计算缓存首地址
 for(k=0;k<plong/2;k++)                             //4 个字
 {
        point=&dotmatrix_chinese[k][0]; //从 ROM 中的下一个汉字的首地址
```

```
        for(i=0;i<16;i++)
        {
              temp1=disp_temp+(plong*i); //对应行地址=汉字数*2(字节数)*行号
              *temp1=*(point+i*2);              //汉字对应行的第二个字节
              temp1++;
              *temp1=*(point+i*2+1);                //汉字对应行的第一个字节
        }
        disp_temp=disp_temp+2;        //下一个汉字
   }
}


/************************************************
函数名称：num_todisp()
传入参数：
函数功能：数字数据到显示缓存区
************************************************/
void num_todisp(void)               //数据到显示缓存区
{
   unsigned char i,k;
   unsigned char *point,*temp1,*disp_temp;
   point=&num_dotmatrix[0][0];          //汉字库首地址
   disp_temp=&disp_buf[0];                 //取计算缓存首地址
 for(k=0;k<plong;k++)                              //4 个字
 {
        point=&num_dotmatrix[k][0]; //从 ROM 中的下一个汉字的首地址
        for(i=0;i<16;i++)
        {
              temp1=disp_temp+(plong*i); //对应行地址=汉字数*2(字节数)*行号
              *temp1=*(point+i+1+1);              //汉字对应行的第二个字节
        }
        disp_temp=disp_temp+1;        //下一个数字
   }
}
```

```
/************************************************
函数名称：letter_todisp()
传入参数：
函数功能：字母数据到显示缓存区
************************************************/
void letter_todisp(void)              //数据到显示缓存区
{
   unsigned char i,k;
   unsigned char *point,*temp1,*disp_temp;
   point=&letter_matrix[0][0];         //汉字库首地址
   disp_temp=&disp_buf[0];                 //取计算缓存首地址
 for(k=0;k<plong;k++)                              //4 个字
 {
        point=&letter_matrix[k][0]; //从 ROM 中的下一个汉字的首地址
        for(i=0;i<16;i++)
        {
              temp1=disp_temp+(plong*i); //对应行地址=汉字数*2(字节数)*行号
              *temp1=*(point+i+1+1);              //汉字对应行的第二个字节
        }
        disp_temp=disp_temp+1;        //下一个数字
   }
}
/*******************************************************************************
函数名称：display()
传入参数：uint disp_long,显示时长  uchar disp_times,显示次数  uchar effect  显示效果
函数功能：显示函数
*******************************************************************************/
//void display(uint disp_long,uchar disp_times,uchar effect)


/************************************************
函数名称：display1p()
传入参数：uchar char_num  显示屏宽度
```

```
函数功能：扫描显示屏
***********************************************/
void display1p()
{
   uchar temp,j,k,i;
   unsigned char *point,*point1;
   uchar hangxu=1;
   point1=&disp_buf[0];
   point = point1;

 for(j=0;j<16;j++)
     {
          for(k=0;k<plong;k++)
          {
              temp=*point;
              for(i=0;i<8;i++)
                   {
                     CLK=0;
                     R1=1;
                    if((temp&0x80)==0x00)
                      {
                       R1=0;
                      }
                       CLK=1;
                     temp<<=1;
                   }
              point++;
          }

          OE=0;
          STR=0;
          STR=1;
          STR = 0;
          Showline(j);
          OE=1;
```

```
            ddlay(9);
            OE=0;
          point=point1+plong*(j);
      }
}

/*void disp_test()     //测试显示屏
{
     uchar temp,j,k,i;
   unsigned char *point,*point1;
   unsigned char disp_test[16]={0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0};
   point1=&disp_test[0];
   point = point1;

  for(j=0;j<16;j++)
      {
            for(k=0;k<plong;k++)
            {
                temp=*point;
                for(i=0;i<8;i++)
                      {
                        CLK=0;
                        R1=1;
                       if((temp&0x01)==0x00)
                          {
                            R1=0;
                          }
                            CLK=1;
                        temp>>=1;
                      }
            }
            point++;
            OE=1;
            STR=0;
            STR=1;
```

```
            STR = 0;
            Showline(j);
            OE=0;
            ddlay(1);
          point=point+1;
       //     point=point+1;
      }


}*/

/************************************************
函数名称：ddlay()
传入参数：uchar ms  延时长度
函数功能：延时
************************************************/
void ddlay(uchar ms)
{
      uchar i,j;
      for(i=0;i<ms;i++)
      {
            for(j=0;j<20;j++);
      }
}

/************************************************
函数名称：Showline()
传入参数：uchar line_num  行号
函数功能：扫描传入参数值的行
************************************************/
void Showline(uchar line_num)
{
      switch(line_num){
      case 15:LL4=1;LL3=1;LL2=1;LL1=1;
                  break;
```

```
        case 14:LL4=1;LL3=1;LL2=1;LL1=0;
                break;
        case 13:LL4=1;LL3=1;LL2=0;LL1=1;
                break;
        case 12:LL4=1;LL3=1;LL2=0;LL1=0;
                break;
        case 11:LL4=1;LL3=0;LL2=1;LL1=1;
                break;
        case 10:LL4=1;LL3=0;LL2=1;LL1=0;
                break;
        case 9:LL4=1;LL3=0;LL2=0;LL1=1;
                break;
        case 8:LL4=1;LL3=0;LL2=0;LL1=0;
                break;
        case 7:LL4=0;LL3=1;LL2=1;LL1=1;
                break;
        case 6:LL4=0;LL3=1;LL2=1;LL1=0;
                break;
        case 5:LL4=0;LL3=1;LL2=0;LL1=1;
                break;
        case 4:LL4=0;LL3=1;LL2=0;LL1=0;
                break;
        case 3:LL4=0;LL3=0;LL2=1;LL1=1;
                break;
        case 2:LL4=0;LL3=0;LL2=1;LL1=0;
                break;
        case 1:LL4=0;LL3=0;LL2=0;LL1=1;
                break;
        case 0:LL4=0;LL3=0;LL2=0;LL1=0;
                break;
        default:break;
        }

}
```

```
/************************************************
函数名称：leftoned(unsigned char *bufdz)
传入参数：unsigned char *bufdz  保留使用用到
函数功能：左移一点
************************************************/
void leftoned()
{
    uchar qq,ww;
    uchar bb2,bb3,bb4;
//  uchar  *buff1;
    uchar  *buff2;
        for(qq=0;qq<16;qq++)
        {
        /*  buff1=&disp_buf[0]+(plong*qq);      //
            temdat1=*buff1;
            bb1=temdat1&0x01;
            if(bb1==1)
            {
                bb2=0x80;
            }
            else
            {
                bb2=0x00;
            }
            temdat1>>=1;
            *buff1=temdat1;*/
            buff2 = &disp_buf[0] + qq *plong;          //显示
//            buff2 = &disp_buf[0] + qq *plong;        //显示
            for(ww=0;ww<plong;ww++)
            {
                //if(ww<plong-1)bb3=*buff2&0x01;
                //else bb3=1;
                bb3=*buff2&0x01;
                bb4=*buff2;
                bb4>>=1;
```

```
                bb4=(bb4&0x7f)|bb2;     //超过 4 个字
                *buff2=bb4;
                buff2++;
                if(bb3==1)
                {
                    bb2=0x80;
                }
                else
                {
                    bb2=0x00;
                }
               if(ww==plong-1)bb2=0x80;
            }

        }
}

/************************************************
函数名称：moveleft(unsigned char sspp)
传入参数: unsigned char sspp  移动速度
函数功能：左移显示效果
************************************************/
void moveleft(unsigned char sspp)
{
    uchar i,j,k;
    switch(scan_temp)
    {
      case 'p':
                photo_todisp();
                break;
      case 'l':
                letter_todisp();
                break;
      case 'n':
                num_todisp();
```

```
                    break;
          case 'c':
                    dztodisp();
                    break;
          default: dztodisp();
                    break;
        }
      for(i=0;i<plong;i++)            //ilong
      {
           for(j=0;j<8;j++)
           {
                leftoned();                  //左移 1 点
                  for(k=0;k<sspp;k++)
                  {
                       display1p();
                       if(scan_mod != scan_temp){scan_temp=scan_mod;goto exit;}
                  }

           }
      }
exit:;
    cldispb();
}
```

## 4，com.c　串口通信初始化

```
#include<com.h>
#include <intrins.h>
unsigned char code Serial_data[42]={"Serial Port Test Source,Wellcom Use Me!"};
unsigned char code Serial_Strar_inf[27]={"Please Select Display Mode!"};
unsigned char code Serial_Error[7]={"error"};
unsigned char code Serial_Recei_OK[10]={"Receive OK"};
void Init_com()
{
    SCON=0x50;
```

```
    TMOD=0x20;
    TH1 =0xF3;
    TL1 =0xF3;
    PCON=0x00;
    TR1 = 1;
    IE=0x90;
}
void Serial_one_Byte(uchar Serial_temp)
{
   SBUF =    Serial_temp;
   while(!TI);
   TI = 0;
}
void Send_char(uchar *Data_addr,uchar char_num)
{
   uchar i;
   for(i=0;i<char_num;i++)
    {
       Serial_one_Byte(*Data_addr);
       Data_addr++;
      }

}
```