

# 单片机第五、六、八次实验报告

53160813

刘国晟

## 一.实验五(重力测量)

### 1.1.实验原理

- 1) 参考附录六，学习点阵式液晶显示屏的控制方法。
- 2) 在液晶显示中，自定义图形和文字的字模对应的字节表需要使用专门的字模软件来生成。可以使用 **PCtoLCD2002** 字模软件提取。
- 3) 字符点阵等数据，需要定义在 **code** 数据段中，具体原理参见示例程序设计部分。
- 4) 向 **LCM** 输出一个命令或数据时，应当在选通信号为高时准备好数据，然后延迟若干指令周期，再将选通信号置为低。
- 5) 重量传感器采用压敏电阻。利用压敏电阻采集应变,产生变化的阻值。利用放大电路将其转化为电压值，通过数模转换将电压值转化成 **CPU** 处理的数字信号。传感器根据编制的程序将数字信号转换为砝码重量显示输出。

### 1.2.实验内容

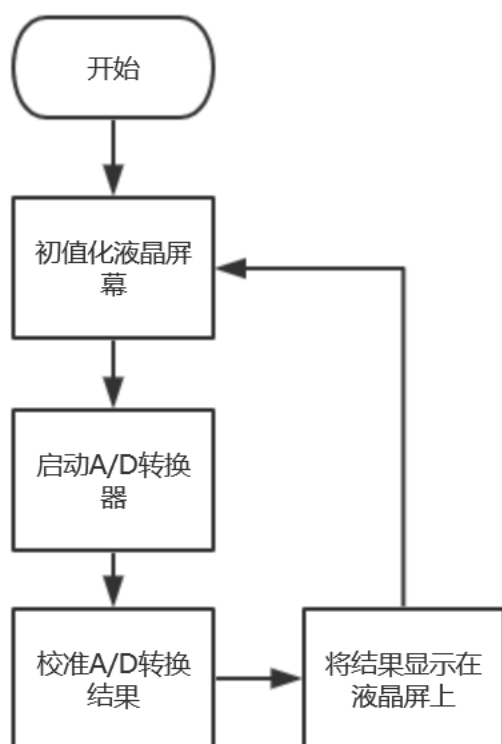
- 1) 参考辅助材料，学习 **C51** 语言使用
- 2) 编写 **C51** 程序，使用重量测量实验板测量标准砝码的重量，将结果（以克计）显示到液晶屏上。误差可允许的范围之间。

### 1.3.实验过程及结果

- 1) 阅读实验原理，掌握 **YM12864C** 的控制方式，编写出基本的输出命令和数据的子程序；
- 2) 掌握点阵字模的构成方式。使用字模软件 **PCtoLCD2002**，设定正确的输出模式，生成点阵数据；
- 3) 使用 **C51** 语言编写重量测量程序。

流程图如下：

图 1.实验五流程图



#### 1.4.思考题及解答

##### 1) 调零的原理，软件调零和调零调零的区别。

答：软件调零是采用软件进行补偿的方法，又称数字调零；调零调零是采用电路检测的方法对硬件进行机械调零。

##### 2) 模/数和数/模的信号转换原理。

答：数字电路处理的信号一般是多位二进制信息号是二进制数字量，输出模拟信号则是与输入数字量成正比的电压或电流数/模转换器的组成寄存器用来暂时存放数字量串行输入，但输出只能是并行输出。 $n$  位寄存器的输出分别控制  $N$  个模拟开关的接通或断开数模和模数转换器数模和模数转换器数模和模数转换器数模和模数转换器。

A/D 转换器是将模拟量转换成数字量的器件，模拟量可以是电压、电流等信号，也可以是声、光、压力、温度等随时间连续变化的非电的物理量。非电量的模拟量可以通过适当的传感器（如光电传感器、压力传感器、温度传感器）转换成电信号。

### 3) I2C 总线在信号通讯过程中的应用。

答：目前有很多半导体集成电路上都集成了 I2C 接口。带有 I2C 接口的单片机有：CYGNAL 的 C8051F0XX 系列，PHILIPSP87LPC7XX 系列，MICROCHIP 的 PIC16C6XX 系列等。很多外围器件如存储器、监控芯片等也提供 I2C 接口。

## 二.实验六

### 2.1.实验原理

使用计算机软件 Protel 99se 对实验一中设计的电路图进行制版操作并进行布线，并生成电路板图纸。

### 2.2.实验内容

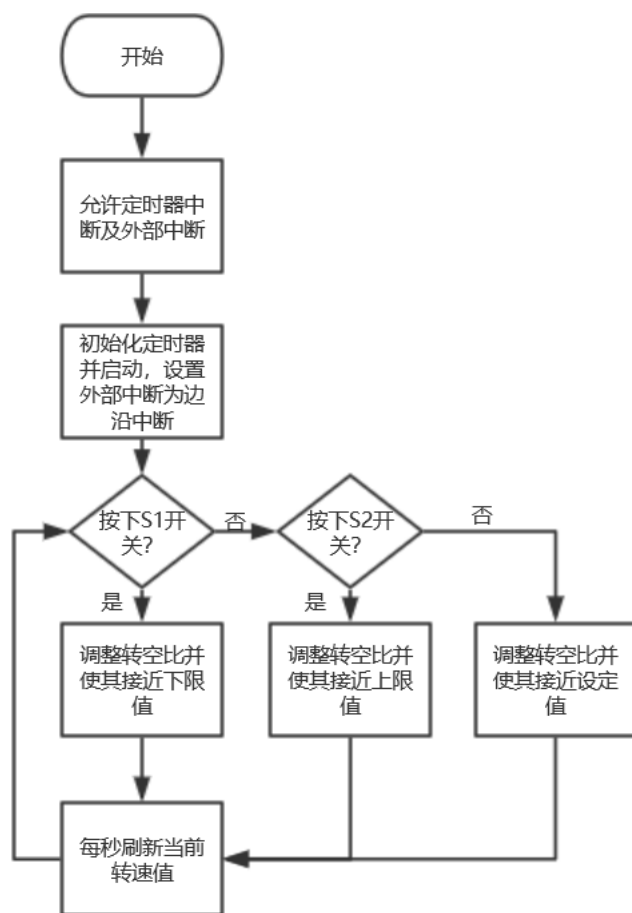
- 1) 在液晶显示屏上显示出直流电机的：当前转速、低目标转速、高目标转速。
- 2) 固定向 P1.1 输出 0，然后测量每秒钟电机转动的转数，将其显示在数码管，每秒刷新一次即可。
- 3) 使用脉宽调制的方法，动态调整向 P1.1 输出的内容，使得电机转速能够稳定在一个预定值附近，同时实时显示当前转速。
- 4) 根据输入修改电机得目标转速值，设置两个转速目标值：低转速和高转速。
- 5) 每隔一秒钟读取两个开关的状态，如果 S1 按下，动态调整输出，使得电机转速能够稳定到低转速目标值附近，如果 S2 按下，动态调整输出，使得电机转速能够稳定到高转速目标值附近。交替显示目标值和当前转速值。

### 2.3.实验过程及流程图

- 1) 建立工程，实现实验内容 1。
- 2) 编写中断程序，测量电机转速。
- 3) 完成控制转速程序。
- 4) 完成整体实验内容。

流程图如下图：

图 2.实验六流程图



## 2.4.思考题及解答

### 1) 讨论脉宽调速和电压调速的区别、优缺点和应用范围。

答: 调压是改变加大电枢上的电压大小, 一般是连续的供电, 电机低速连续转动; PWM 是改变加到电枢上电源时间长短, 也就是加一定时间的电, 然后断开, 过一定时间再加电, 断续加电, 相当于电机转一下, 减速 (停止), 然后继续这个循环, 微观看是非匀速的, 宏观看是匀速的。对于脉宽调速, 优点是效率高, 调速范围还行, 缺点是在最低转速时电机运行时脉动的, 噪音变大, 而且负载越大越严重; 对于电压调速, 优点是电机运行在整个调速范围内都平稳。调速范围也最大, 从电机的始动电压可以调到额定电压。

### 2) 说明程序原理中累加进位法的正确性。

答: 从概率学的角度, 在各个值设定合理的情况下, 累加进位法可以最大程度上实现每隔一定时间驱动电机转动一次, 对实验的顺利进行有至关重要的作用。

### 3) 计算转速测量的最大可能误差, 讨论减少误差的办法。

答: 缩短累加进位法中断的计时间隔, 以及在计秒中断中每个小循环都添加对 于当前速度的监测并随之更改  $N$  的值。

## 三.实验八

### 3.1.实验原理

1) 本实验使用的 DS18B20 是单总线数字温度计, 测量范围从 $-55^{\circ}\text{C}$ 到 $+125^{\circ}\text{C}$ , 增量值为  $0.5^{\circ}\text{C}$ 。

2) 用于贮存测得的温度值的两个 8 位存贮器 RAM 编号为 0 号和 1 号。

3) 1 号存贮器存放温度值的符号, 如果温度为负 ( $^{\circ}\text{C}$ ), 则 1 号存贮器 8 位全 为 1, 否则全为 0。

4) 0 号存贮器用于存放温度值的补码 LSB(最低位)的 1 表示  $0.5^{\circ}\text{C}$  。

5) 将存贮器中的二进制数求补再转换成十进制数并除以 2, 就得到被测温度值。

6) 温度检测与控制系统由加热灯泡, 温度二极管, 温度检测电路, 控制电路和 继电器组成。温度二极管和加热灯泡封闭在一个塑料保温盒内, 温度二极管监测 保温盒内的温度, 用温控实验板内部的 A/D 转换器 ADC7109 检测二极管两端的电 压, 通过电压和温度的关系, 计算出盒内空气的实际温度。

### 3.2.实验内容

1) 掌握使用传感器测量与控制温度的原理与方法, 使用 C51 语言编写实现温度 控制的功能, 使用超声波/温度实验板测量温度, 将温度测量的结果 (单位为摄 氏度) 显示到液晶屏上。

2) 编程实现测量当前教室的温度, 显示在 LCM 液晶显示屏上。

3) 通过 S1 设定一个高于当前室温的目标温度值。

4) 编程实现温度的控制, 将当前温度值控制到目标温度值并稳定的显示。

### 3.3.实验步骤及流程图

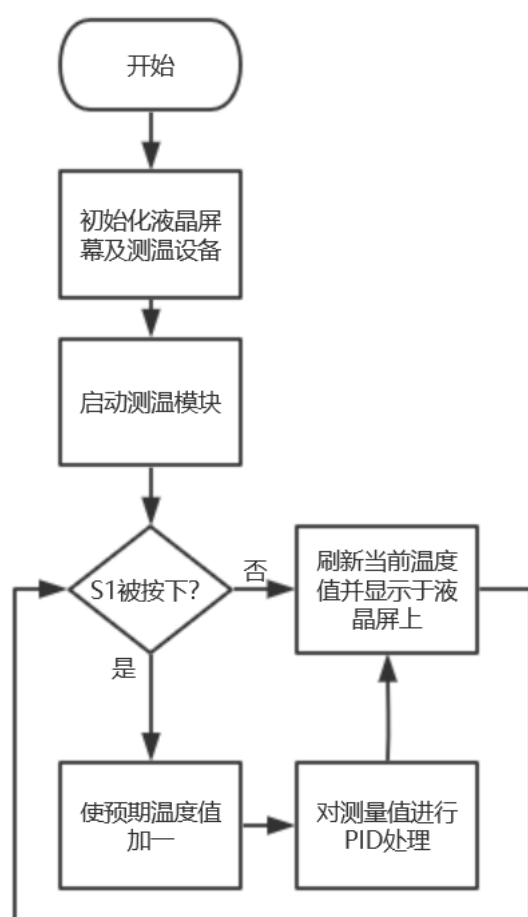
1) 预习, 参考附录三, 预习 DS18B20 的编程结构, 编程时注意 DS18B20 的时间 要求, 必须准确满足。根据实验原理附录中的流程图进行编程。

2) 将编译后的程序下载到 51 单片机，观察温度的测量结果。

3) 程序调试

流程图如下：

图 3.实验八流程图



### 3.4.思考题

1) 进行精确的延时的程序有几种方法？各有什么优缺点？

答：使用 `_nop_()` 语句：简洁方便，但精确度不够且占用 CPU 资源，适用于短延时；使用 `while` 循环或 `for` 循环：延时长短易于调节，但精确度不够；使用定时器中断延时，精确度高，应用性强，能适用于各种情况，但使用方法较为复杂。

2) 参考其他资料，了解 **DS18B20** 的其他命令用法。

指令类型	指令	功能	详细描述
ROM 指令	[F0H]	搜索 ROM 指令	当系统初始化时,总线控制器通过此指令多次循环搜索 ROM 编码,以确认所有从机器件
	[33H]	读取 ROM 指令	当总线上只有一只 DS18B20 时才会使用此指令,允许总线控制器直接读取从机的序列码
	[55H]	匹配 ROM 指令	匹配 ROM 指令,使总线控制器在多点总线上定位一只特定的 DS18B20
	[CCH]	忽略 ROM 指令	忽略 ROM 指令,此指令允许总线控制器不必提供 64 位 ROM 编码就使用功能指令
	[ECH]	报警搜索指令	当总线上存在满足报警条件的从机时,该从机将响应此指令
功能指令	[44H]	温度转换指令	此条指令用来控制 DS18B20 启动一次温度转换,生成的温度数据以 2 字节的形式存储在高速暂存器中
	[4EH]	写暂存器指令	此指令向 DS18B20 的暂存器写入数据,开始位置在暂存器第 2 字节 (TH 寄存器),以最低有效位开始传送
	[BEH]	读暂存器指令	此指令用来读取 DS18B20 暂存器数据,读取将从字节 0 开始,直到第 9 字节 (CRC 校验位) 读完
	[48H]	拷贝暂存器指令	此指令将 TH、TL 和配置寄存器的数据拷贝到 EEPROM 中得以保存
	[B8H]	召回 EEPROM 指令	将 TH、TL 以及配置寄存器中的数据从 EEPROM 拷贝到暂存器
	[B4H]	读电源模式指令	总线控制器在发出此指令后启动读时隙,若为寄生电源模式,DS18B20 将拉低总线,若为外部电源模式,则将总线拉高,用以判断 DS18B20 的电源模式 <a href="http://www.Ndiy.cn">www.Ndiy.cn</a>

## 四.实验问题与感想

本次实验中主要解决了以下几个问题:

- 1.对于液晶屏幕点阵显示模式与字模生成方式之间关系的理解(逆序列行式生成)
2. 在实验过程中,开始时对直流电机如何触发中断理解有误差,对电信号如何 控制电机转动认识不足,出现了编程错误,此后解决。(硬件遮光触发中断)
- 3.直流电机实验中,对于计数器计数值的设置不科学,导致中断时间产生误差,后在老师指导下发现错误并解决。

## 五.代码附录

### 5.1.实验五代码

```
//????:??,??,16*16
#include <reg52.h>
#include <intrins.h>
#define uchar unsigned char
#define uint unsigned int
```



```
#define LCD_databus P2 // LCD 8bit data bus
```

```
uchar num;
```

```
sbit RS=P3^5;
```

```
sbit RW=P3^4;
```

```
sbit EN=P3^3;
```

```
sbit CS1=P1^7;
```

```
sbit CS2=P1^6;
```

```
uint loop=0;
```

```
uint times=0;
```

```
uchar code fa[]={
```

```
{0x02,0x82,0xE2,0x5E,0x42,0xC2,0x02,0x10,0x10,0x10,0xFF,0x10,0x10,0x18,0x10,0x0  
0,
```

```
0x02,0x01,0x7F,0x10,0x10,0x3F,0x01,0x21,0x79,0x27,0x21,0x29,0x31,0x61,0x01,0x0  
0};/*"?",0*/
```

```
uchar code ma[]={
```

```
{0x02,0x82,0xE2,0x5E,0x42,0xC2,0x00,0x02,0xFA,0x82,0x82,0x82,0xFE,0x80,0x00,0x  
00,
```

```
0x01,0x00,0x7F,0x10,0x10,0x3F,0x00,0x04,0x04,0x04,0x44,0x84,0x40,0x3F,0x00,0x00  
};/*"?",1*/
```

```
uchar code zhong[]={
```

```
{0x08,0x08,0x0A,0xEA,0xAA,0xAA,0xAA,0xFF,0xA9,0xA9,0xA9,0xE9,0x08,0x08,0x08,0  
x00,
```

```
0x40,0x40,0x48,0x4B,0x4A,0x4A,0x4A,0x7F,0x4A,0x4A,0x4A,0x4B,0x48,0x40,0x40,0x  
00};/*"?",2*/
```

uchar code liang[]={

{0x40,0x40,0x40,0xDF,0x55,0x55,0x55,0xD5,0x55,0x55,0x55,0xDF,0x40,0x40,0x40,0x00,

0x40,0x40,0x40,0x57,0x55,0x55,0x55,0x7F,0x55,0x55,0x55,0x57,0x50,0x40,0x40,0x00};/\*"?",3\*/

uchar code wei[]={

{0x00,0x10,0x10,0x12,0x14,0x1C,0x10,0xF0,0x9F,0x10,0x10,0x10,0x10,0xF8,0x10,0x00,

0x00,0x00,0x40,0x20,0x10,0x08,0x06,0x01,0x00,0x11,0x26,0x40,0x20,0x1F,0x00,0x00};/\*"?",4\*/

uchar code mao[]={

{0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,

0x00,0x00,0x36,0x36,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00};/\*":",5\*/

uchar code ling[]={

{0x00,0x00,0x00,0xF8,0x04,0x02,0x02,0x02,0x02,0x02,0x04,0xF8,0x00,0x00,0x00,0x00,

0x00,0x00,0x00,0x1F,0x20,0x40,0x40,0x40,0x40,0x40,0x20,0x1F,0x00,0x00,0x00,0x00};/\*"0",0\*/

uchar code yi[]={

{0x00,0x00,0x00,0x00,0x00,0x08,0x04,0xFE,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,

0x00,0x00,0x00,0x00,0x00,0x40,0x40,0x7F,0x40,0x40,0x00,0x00,0x00,0x00,0x00,0x00

```
0};/*"1",1*/
```

```
uchar code er[]=
```

```
{0x00,0x00,0x00,0x18,0x04,0x02,0x02,0x02,0x82,0x82,0x84,0x78,0x00,0x00,0x00,0x00,
```

```
0x00,0x00,0x00,0x78,0x44,0x42,0x41,0x41,0x40,0x40,0x40,0x70,0x00,0x00,0x00,0x00};/*"2",2*/
```

```
uchar code san[]=
```

```
{0x00,0x00,0x00,0x0C,0x02,0x02,0x02,0x82,0x82,0x42,0x22,0x1C,0x00,0x00,0x00,0x00,
```

```
0x00,0x00,0x00,0x30,0x40,0x40,0x40,0x40,0x40,0x41,0x22,0x1C,0x00,0x00,0x00,0x00};/*"3",3*/
```

```
uchar code si[]=
```

```
{0x00,0x00,0x00,0x00,0x80,0x60,0x1C,0x02,0xFE,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
```

```
0x00,0x00,0x0C,0x0A,0x09,0x08,0x48,0x48,0x7F,0x48,0x48,0x08,0x00,0x00,0x00,0x00};/*"4",4*/
```

```
uchar code wu[]=
```

```
{0x00,0x00,0x00,0xFE,0x82,0x42,0x42,0x42,0x42,0x42,0x82,0x02,0x00,0x00,0x00,0x00,
```

```
0x00,0x00,0x00,0x31,0x40,0x40,0x40,0x40,0x40,0x40,0x20,0x1F,0x00,0x00,0x00,0x00};/*"5",5*/
```

```
uchar code liu[]=
```

```
{0x00,0x00,0x00,0xF8,0x04,0x82,0x82,0x82,0x82,0x82,0x04,0x18,0x00,0x00,0x00,0x00,  
00,
```

```
0x00,0x00,0x00,0x1F,0x21,0x40,0x40,0x40,0x40,0x40,0x21,0x1E,0x00,0x00,0x00,0x0  
0};/*"6",6*/
```

```
uchar code qi[]=
```

```
{0x00,0x00,0x00,0x0E,0x02,0x02,0x02,0x02,0x82,0x42,0x32,0x0E,0x00,0x00,0x00,0x  
00,
```

```
0x00,0x00,0x00,0x00,0x00,0x00,0x70,0x0E,0x01,0x00,0x00,0x00,0x00,0x00,0x00,0x0  
0};/*"7",7*/
```

```
uchar code ba[]=
```

```
{0x00,0x00,0x00,0x38,0x44,0x82,0x82,0x82,0x82,0x82,0x44,0x38,0x00,0x00,0x00,0x  
00,
```

```
0x00,0x00,0x00,0x1E,0x21,0x40,0x40,0x40,0x40,0x40,0x21,0x1E,0x00,0x00,0x00,0x0  
0};/*"8",8*/
```

```
uchar code jiu[]=
```

```
{0x00,0x00,0x00,0x78,0x84,0x02,0x02,0x02,0x02,0x02,0x84,0xF8,0x00,0x00,0x00,0x  
00,
```

```
0x00,0x00,0x00,0x18,0x20,0x41,0x41,0x41,0x41,0x41,0x20,0x1F,0x00,0x00,0x00,0x0  
0};/*"9",9*/
```

```
uchar code ke[]=
```

```
{0x00,0x04,0x04,0xE4,0x24,0x24,0x24,0x3F,0x24,0x24,0x24,0xE4,0x04,0x04,0x00,0x  
00,
```

```
0x00,0x00,0x80,0x43,0x31,0x0F,0x01,0x01,0x01,0x3F,0x41,0x43,0x40,0x40,0x70,0x0
0};/*"?",10*/
```

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

```
* ?????,i??256,??256????
```

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

```
void delay(uint i)
```

```
{
```

```
    while(--i);
```

```
}
```

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

```
* ??BUSY=0, busy p2^7
```

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

```
/*
```

状态字结构： BUSY 0 ON/OFF RESET 0 0 0 0

BUSY 表示当前 LCM 接口控制电路运行状态。BUSY=1 表示 LCM 正在处理 MPU 发过

来的指令或数据。此时接口电路被封锁，不能接受除读状态字以外的任何操作。

BUSY=0

表示 LCM 接口控制电路已外于“准备好”状态，等待 MPU 的访问。

ON/OFF 表示当前的显示状态。ON/OFF=1 表示关显示状态，ON/OFF=0 表示开

显

示状态。

RESET 表示当前 LCM 的工作状态，即反映/RES 端的电平状态。当/RES 为低电平

状态时，LCM 处于复位工作状态，标志位 RESET=1。当/RES 为高电平状态时，LCM 为

正常工作状态，标志位 RESET=0。

在指令设置和数据读写时要注意状态字中的 BUSY 标志。只有在 BUSY=0 时，MPU 对 LCM 的操作才能有效。因此 MPU 在每次对 LCM 操作之前，都要读出状态字判断 BUSY

是否为“0”。若不为“0”，则 MPU 需要等待，直至 BUSY=0 为止。

\*/

```
void Read_busy()
```

```
{
    P2 = 0x00;
    RS = 0;           // RS/RW=0/1,
    RW = 1;
    EN = 1;           // LCM
    while (P2 & 0x80); // P2.7=0.
    EN = 0;           // LCM
}
```

```
void SelectScreen(uchar screen)
```

```
{
    switch(screen)
    {
        case 0: CS1=1; CS2=1; break;    // 关闭两屏
        case 1: CS1=1; CS2=0; break;    // 开启左半屏
        case 2: CS1=0; CS2=1; break;    // 开启右半屏
        default: break;
    }
}
```

```
}
```

```
/*
```

LCM 有四种 8 位控制指令，分别为显示开关设置、显示起始行指设置、页面地址设置、列地址设置，其头两位分别为 00、11、10、01，写入指令时根据该两位判断指令种类

```
*/
```

```
void write_LCD_command(uchar value)
```

```
{
```

```
    LCD_databus=0xff;
```

```
    Read_busy();
```

```
    RS=0;
```

```
    RW=0;
```

```
    LCD_databus=value;
```

```
    EN=1;
```

```
    delay(100);
```

```
    EN=0;
```

```
}
```

```
void write_LCD_data(uchar value)
```

```
{
```

```
    LCD_databus=0xff;
```

```
    Read_busy();
```

```
    RS=1;
```

```
    RW=0;
```

```
    LCD_databus=value;
```

```
    EN=1;
```

```
    delay(100);
```

```
    EN=0;  
}
```

```
void Set_page(uchar page)  
{  
    page = 0xb8 | page;  
    write_LCD_command(page);  
}
```

```
void Set_line(uchar startline)  
{  
  
    startline = 0xC0 | startline;  
    write_LCD_command(startline);  
}
```

```
void Set_column(uchar column)  
{  
    column = column & 0x3f;  
    column = 0x40 | column;  
    write_LCD_command(column);  
}
```

```
void SetOnOff(uchar onoff)  
{
```



```
    onoff = 0x3e | onoff;//最后一位置 1，开显示状态
    write_LCD_command(onoff);
}
```

```
void ClearScreen(uchar screen)
{
    uchar i,j;
    SelectScreen(screen);
    for(i=0;i<8;i++)
    {
        Set_page(i);
        Set_column(0);
        for(j=0;j<64;j++)
        {
            write_LCD_data(0x00);
        }
    }
}
```

```
void InitLCD()
{
    Read_busy();
    SelectScreen(0);
    SetOnOff(0);
    SelectScreen(0);
    SetOnOff(1);
    SelectScreen(0);
    ClearScreen(0);
}
```

```

        Set_line(0);
    }

void Display(uchar ss,uchar page,uchar column,uchar *p)
{
    uchar i;
    SelectScreen(ss);
    Set_page(page);
    Set_column(column);
    for(i=0;i<16;i++)
    {
        write_LCD_data(p[i]);
    }
    Set_page(page+1);
    Set_column(column);
    for(i=0;i<16;i++)
    {
        write_LCD_data(p[i+16]);
    }
}

/*=====
                        AD
=====*/

#define FOSC      1843200L//晶振频率
#define BAUD      9600    //波特率

/*Declare SFR assoicated with the ADC*/
sfr    ADC_CONTR = 0xBC;

```

```
sfr    ADC_RES      = 0xBD;
```

```
sfr    ADC_LOW2     = 0xBE;
```

```
sfr P1ASF          = 0x9D;
```

```
sfr AURX1          = 0xA2;
```

```
/*Define ADC operation const for ADC_CONTR*/
```

```
#define ADC_POWER 0x80    //电源位，第一位，置 1 以启动 A/D 转换
```

```
#define ADC_FLAG 0x10     //转换结束位，第四位，预置 1 以等待清零。
```

```
#define ADC_START 0x08    //转换开始位，第五位，设定为 1 时开始转换。
```

```
#define ADC_SPEEDLL 0x00    //以下 4 行为时钟定义，依次分别  
540,396,180,90 个时钟周期转换一次，由于在八位中 SPEED1 和 SPEED0 处高 2、3  
位，单独定义转速时只定义高 2、3 位，其余置零
```

```
#define ADC_SPEEDL 0x20
```

```
#define ADC_SPEEDH 0x40
```

```
#define ADC_SPEEDHH 0x60
```

```
void InitUart();
```

```
void InitADC();
```

```
void InitADC_n(uchar n);
```

```
uint ADC_GET(uchar n);
```

```
void DelayMs(uint n);
```

```
/*Intital UART*/
```

```
void InitUart()
```

```
{
```

```
    SCON = 0x5a;采用方式 1 工作，并将 REN 置 1 以允许串行口接收，
```

```
    TMOD = 0x20;
```

```
    TH1 = TL1 = -(FOSC/12/32/BAUD);
```

```

        TR1 = 1;
    }

/*Intital ADC sfr*/
void InitADC()
{
    P1ASF = 0xff;    //I 令 P1 口中所有位均可被作为 A/D 使用
    ADC_RES = 0;    //
    ADC_CONTR = ADC_POWER | ADC_SPEEDLL;
    DelayMs(2);
}

```

```

/*?n????*/
void InitADC_n(uchar n)
{
    n &= 0x07;        // ???0~7??
    AURX1 |= 0x04;    // ADRI 位置 1
    P1ASF = 1<<n;
}

```

```

/**?n??ADC????*/
uint ADC_GET(uchar n)
{
    uint adc_data;

    n &= 0x07;        // 频道选择，即 1.0-1.7
    ADC_RES = 0;        // ???????
    ADC_LOW2 = 0;    // ???????
    ADC_CONTR = 0;    // 清零 8 位控制位
}

```

```

ADC_CONTR |= (ADC_POWER | ADC_SPEEDLL | n | ADC_START);

_nop();_nop();_nop();_nop();_nop();_nop();

while(!((ADC_CONTR & ADC_FLAG) == 0x10))

adc_data = (ADC_RES & 0x03) * 256 + ADC_LOW2;
ADC_CONTR &= 0xef;  // ??????0,ADC_FLAG??
return adc_data;    // adc_data??(0~1023)
}

/*??????*/
void DelayMs(uint n)
{
    uint x;
    while(n--)
    {
        x = 5000;
        while(x--);
    }
}

```

```

void display_LCD(uint n)
{

```

```
uchar a,b,c;
Display(1,loop,2*16,fa);
Display(1,loop,3*16,ma);
Display(2,loop,0*16,zhong);
Display(2,loop,1*16,liang);
Display(1,loop+4,2*16,wei);
a = n / 100;
b = n % 100 / 10;
c = n % 10;
switch(a)
{
case 0: Display(1,loop+4,3*16,ling);break;
case 1: Display(1,loop+4,3*16,yi); break;
case 2: Display(1,loop+4,3*16,er); break;
case 3: Display(1,loop+4,3*16,san); break;
case 4: Display(1,loop+4,3*16,si); break;
case 5: Display(1,loop+4,3*16,wu); break;
case 6: Display(1,loop+4,3*16,liu); break;
case 7: Display(1,loop+4,3*16,qi); break;
case 8: Display(1,loop+4,3*16,ba); break;
case 9: Display(1,loop+4,3*16,jiu); break;
}
switch(b)
{
case 0: Display(2,loop+4,0*16,ling);break;
case 1: Display(2,loop+4,0*16,yi); break;
case 2: Display(2,loop+4,0*16,er); break;
case 3: Display(2,loop+4,0*16,san); break;
case 4: Display(2,loop+4,0*16,si); break;
case 5: Display(2,loop+4,0*16,wu); break;
```

```

    case 6: Display(2,loop+4,0*16,liu); break;
    case 7: Display(2,loop+4,0*16,qi);  break;
    case 8: Display(2,loop+4,0*16,ba);  break;
    case 9: Display(2,loop+4,0*16,jiu); break;
}
switch(c)
{
    case 0: Display(2,loop+4,1*16,ling);break;
    case 1: Display(2,loop+4,1*16,yi);  break;
    case 2: Display(2,loop+4,1*16,er);  break;
    case 3: Display(2,loop+4,1*16,san); break;
    case 4: Display(2,loop+4,1*16,si);  break;
    case 5: Display(2,loop+4,1*16,wu);  break;
    case 6: Display(2,loop+4,1*16,liu); break;
    case 7: Display(2,loop+4,1*16,qi);  break;
    case 8: Display(2,loop+4,1*16,ba);  break;
    case 9: Display(2,loop+4,1*16,jiu); break;
}

    DelayMs(50);
}

/*???????*/

void main()
{
    InitLCD();
    InitADC();
    ClearScreen(0);
    Set_line(0);
    while(1){
        uint ad_0 = 0;
        InitADC_n(0);

```

```

        ad_0 = ADC_GET(0);
        if (ad_0 > 13 && ad_0 < 16)
        {
            ad_0 = 21;
        }
        else if (ad_0 > 90 && ad_0 < 110)
            ad_0 = 70;
        else if (ad_0 > 135 && ad_0 < 175)
        {
            ad_0 = 102;
        }
        else if (ad_0 > 225 && ad_0 < 265)
            ad_0 = 152;
        // ad_0 = (ad_0 - 70)/ 2;
        display_LCD(ad_0);

    }
}

```

## 5.2.实验六代码

```

#include <reg52.h>
#include <intrins.h>

#define uchar unsigned char
#define uint unsigned int

//p4 LED
sfr P4=0xC0;
sfr P4SW=0xBB;
sbit sclk=P4^4;

```



```
sbit sdata=P4^5;
```

```
//yejing
```

```
sbit CS1=P1^7;
```

```
sbit CS2=P1^6;
```

```
sbit E=P3^3;
```

```
sbit RW=P3^4;
```

```
sbit RS=P3^5;
```

```
sbit RES=P1^5;
```

```
sbit BUSY=P2^7;
```

```
sbit swh1=P3^6;//S1
```

```
sbit swh2=P3^7;//S2
```

```
sbit motor=P1^1;
```

```
uchar code zima[20][32]=
```

```
{  
0x00,0x00,0xC0,0xE0,0x30,0x10,0x08,0x08,0x08,0x08,0x08,0x18,0x30,0xE0,0xC0,0x0  
0,  
0x00,0x00,0x07,0x0F,0x18,0x10,0x20,0x20,0x20,0x20,0x20,0x10,0x18,0x0F,0x07,0x0  
0,///  
"0"*0/  

```

```
0x00,0x00,0x00,0x10,0x10,0x10,0x10,0xF0,0xF8,0x00,0x00,0x00,0x00,0x00,0x00,0x0  
0,  
0x00,0x00,0x00,0x20,0x20,0x20,0x20,0x3F,0x3F,0x20,0x20,0x20,0x20,0x00,0x00,0x0  
0,///  
"1"*1/  

```

```
0x00,0x00,0x60,0x50,0x10,0x08,0x08,0x08,0x08,0x08,0x08,0x98,0xF0,0x70,0x00,0x0  
0,  

```

0x00,0x00,0x20,0x30,0x28,0x28,0x24,0x24,0x22,0x22,0x21,0x20,0x30,0x18,0x00,0x0  
0,///<sup>2</sup>2/

0x00,0x00,0x30,0x30,0x08,0x08,0x88,0x88,0x88,0x88,0x58,0x70,0x30,0x00,0x00,0x0  
0,  
0x00,0x00,0x18,0x18,0x20,0x20,0x20,0x20,0x20,0x20,0x31,0x11,0x1F,0x0E,0x00,0x0  
0,///<sup>3</sup>3/

0x00,0x00,0x00,0x00,0x00,0x80,0x40,0x20,0x10,0xF0,0xF8,0xF8,0x00,0x00,0x00,0x0  
0,  
0x00,0x04,0x06,0x05,0x05,0x04,0x24,0x24,0x24,0x3F,0x3F,0x3F,0x24,0x24,0x24,0x00  
,///<sup>4</sup>4/

0x00,0x00,0x00,0xC0,0x38,0x88,0x88,0x88,0x88,0x88,0x88,0x88,0x08,0x08,0x00,0x0  
0,  
0x00,0x00,0x18,0x29,0x21,0x20,0x20,0x20,0x20,0x20,0x30,0x11,0x1F,0x0E,0x00,0x0  
0,///<sup>5</sup>5/

0x00,0x00,0x80,0xE0,0x30,0x10,0x98,0x88,0x88,0x88,0x88,0x88,0x98,0x10,0x00,0x0  
0,  
0x00,0x00,0x07,0x0F,0x19,0x31,0x20,0x20,0x20,0x20,0x20,0x20,0x11,0x1F,0x0E,0x0  
0,///<sup>6</sup>6/

0x00,0x00,0x30,0x18,0x08,0x08,0x08,0x08,0x88,0x48,0x28,0x18,0x08,0x00,0x0  
0,  
0x00,0x00,0x00,0x00,0x00,0x00,0x38,0x3E,0x01,0x00,0x00,0x00,0x00,0x00,0x00,0x0  
0,///<sup>7</sup>7/

0x00,0x00,0x70,0x70,0xD8,0x88,0x88,0x08,0x08,0x08,0x08,0x98,0x70,0x70,0x00,0x0  
0,

0x00,0x0C,0x1E,0x12,0x21,0x21,0x20,0x21,0x21,0x21,0x23,0x12,0x1E,0x0C,0x00,0x0  
0,///  
"8"\*8/

0x00,0xE0,0xF0,0x10,0x08,0x08,0x08,0x08,0x08,0x18,0x10,0xF0,0xC0,0x00,0x0  
0,  
0x00,0x00,0x11,0x33,0x22,0x22,0x22,0x22,0x32,0x11,0x1D,0x0F,0x03,0x00,0x0  
0,///  
"9"\*9/

0x08,0x08,0x0A,0xEA,0xAA,0xAA,0xAA,0xFF,0xA9,0xA9,0xA9,0xE9,0x08,0x08,0x08,0x  
00,  
0x40,0x40,0x48,0x4B,0x4A,0x4A,0x4A,0x7F,0x4A,0x4A,0x4A,0x4B,0x48,0x40,0x40,0x  
00,///  
"?"\*10/

0x40,0x40,0x40,0xDF,0x55,0x55,0x55,0xD5,0x55,0x55,0x55,0xDF,0x40,0x40,0x40,0x0  
0,  
0x40,0x40,0x40,0x57,0x55,0x55,0x55,0x7F,0x55,0x55,0x55,0x57,0x50,0x40,0x40,0x0  
0,///  
"?"\*11/

0x00,0x00,0x00,0x00,0x00,0x00,0xC0,0xC0,0xC0,0xC0,0x00,0x00,0x00,0x00,0x0  
0,  
0x00,0x00,0x00,0x00,0x00,0x00,0x30,0x30,0x30,0x30,0x00,0x00,0x00,0x00,0x0  
0,///  
":"\*12/

0x00,0x04,0x04,0xE4,0x24,0x24,0x24,0x3F,0x24,0x24,0x24,0xE4,0x04,0x04,0x00,0x0  
0,  
0x00,0x00,0x80,0x43,0x31,0x0F,0x01,0x01,0x01,0x3F,0x41,0x43,0x40,0x40,0x70,0x0  
0,///  
"?"\*13/

};

```
uchar tab[15]=  
    {0xC0,0xF9,0xA4,0xB0,0x99,  
      0x92,0x82,0x0F8,0x80,0x90}; //0-9
```

```
uchar tspeed=0;  
uchar cspeed=0;  
uchar xspeed=120;  
uchar speedUp = 140;  
uchar speedLow =100;
```

```
uchar t1_cnt=0; // 50ms*20=1s  
int N=50;  
int M=256;  
int X=0;
```

```
void send_byte(uchar dat ,uchar cs1,uchar cs2);  
void send_all(uint page,uint lie,uint offset);  
void init();  
void clearscreen();  
void init_yejing();  
void sendbyte(uchar ch);  
void display(uchar n);  
void delay1();  
void delay2();  
void delay(uint x)  
{  
    while(x--);  
}
```

```
void main()
```

```

{
    init();
    init_yejing();
    motor=0;
    while(1)
    {
        clearsreen();
        send_all(1,3,speedLow/100);
        send_all(1,4,(speedLow/10)%10);
        send_all(1,5,speedLow%10);
        //current
        send_all(3,3,cspeed/100);
        send_all(3,4,(cspeed/10)%10);
        send_all(3,5,cspeed%10);///?

        send_all(5,3,speedUp/100);
        send_all(5,4,(speedUp/10)%10);
        send_all(5,5,speedUp%10);

        delay1();
        display(cspeed);
        delay(50000);
    }
}

```

```

void init()
{
    P4SW=0x30;

```

```

    IT0=1;

    EA=1;

    ET1=1;

    ET0=1;

    EX0=1;

/*
50ms == 50000
    == 65536-50000 D= 15536 D= 0x3CB0

0.1ms == 100
    == 65536-100 D= 65436 D= 0xFF9C
*/

    TMOD=0x11; ///??1 16????

    TH1=0x3C;

    TL1=0xB0; ///50ms,中断 1 定时

    TH0=0xFF;

    TL0=0x9C; ///0.1ms,中断 0 定时


    TR0=1;

    TR1=1; ///允许定时器 0、1 计数
}

///???

void ex_int0() interrupt 0 ///INT0
{
    tspeed++;
}

void t1_int() interrupt 3 ///T1
{
    if(++t1_cnt<20)
    {
        TH1=0x3C;
    }
}

```

```

        TL1=0xB0;

    return;
}

if(swh1==0)
{
    xspeed = speedLow;
}

if(swh2==0){
    xspeed = speedUp;
}

if(swh1==1 && swh2==1 ){
    xspeed = (speedLow+speedUp)/2;
}

t1_cnt=0;
cspeed=tspeed;
tspeed=0;
if(cspeed>xspeed) N--;
if(cspeed<xspeed) N++;
}

```

```

void t0_int() interrupt 1  ///0.1ms???? T0
{
    TH0=0xFF;
    TL0=0x9C;
    X+=N;

    if(X>M)
    {
        motor=0;
        X-=M;
    }
}

```

```

        else
            motor=1;
    }
void init_yejing()
{
    send_byte(192,1,1);//起始行
    send_byte(63,1,1);///开关
}
void send_byte(uchar dat,uchar cs1,uchar cs2)
{
    P2=0xff;
    CS1=cs1; CS2=cs2;
    RS=0; RW=1; E=1;
    while(BUSY) ;
    E=0;
    RS=!(cs1&&cs2),RW=0;
    P2=dat;
    E=1; delay(3); E=0;
    CS1=CS2=0;
}

void send_all(uint page,uint lie,uint offset)
{
    uint i,j,k=0;
    for(i=0;i<2;++i)
    {
        send_byte(184+i+page,1,1);
        send_byte(64+lie*16-(lie>3)*64,1,1);
        for(j=0;j<16;++j)
            send_byte(zima[offset][k++],lie<4,lie>=4);
    }
}

```



```

    }
}

void clearsreen()
{
    int i,j;
    for(i=0;i<8;++i)
    {
        send_byte(184+i,1,1);
        send_byte(64,1,1);
        for(j=0;j<64;++j)
        {
            send_byte(0x00,0,1);
            send_byte(0x00,1,0);
        }
    }
}

```

```

void sendbyte(uchar ch)
{
    uchar shape,c;
    shape=tab[ch];
    for(c=0;c<8;c++)
    {
        sclk=0;
        sdata=shape & 0x80;
        sclk=1;
        shape <<= 1;
    }
}
} //sent LED

```

```
//LED??
void display(uchar n)
{
    sendbyte(n%10);
    sendbyte((n/10)%10);
    sendbyte(n/100);
}
```

```
void delay1()
{
    int i,j;
    for(i=0;i<1000;i++)
        for(j=0;j<500;j++);
}
```

```
void delay2()
{
    int i,j;
    for(i=0;i<1000;i++)
        for(j=0;j<1000;j++);
}
```

### 5.3.实验八代码

```
//????:??,16*16
#include<reg52.h>
#include<intrins.h>
#include<math.h>

typedef unsigned char uchar;
```

```
typedef unsigned int uint;
```

```
sbit s1 = P3^6;
```

```
sbit s2 = P3^7;
```

```
sbit RS=P3^5;
```

```
sbit RW=P3^4;
```

```
sbit EN=P3^3;
```

```
sbit CS1=P1^7;
```

```
sbit CS2=P1^6;
```

```
sbit DQ=P1^4;
```

```
sbit up=P1^1;
```

```
uchar Ek,Ek1,Ek2;
```

```
uchar Kp,Ki,Kd;
```

```
uint res,Pmax;
```

```
uint xx=0;
```

```
uint times=0;
```

```
void delay_us(uchar n)
```

```
{
```

```
    while (n--)
```

```
    {
```

```
        _nop_();
```

```
        _nop_();
```

```
    }
```

```
}
```

```
unsigned char code shu[10][32]={
```

```
0x00,0x00,0xC0,0xE0,0x30,0x10,0x08,0x08,0x08,0x08,0x08,0x18,0x30,0xE0,0xC0,0x0
```

```
0,
```

```
0x00,0x00,0x07,0x0F,0x18,0x10,0x20,0x20,0x20,0x20,0x20,0x10,0x18,0x0F,0x07,0x0
```

0,/\*"0",0\*/

```
0x00,0x00,0x00,0x10,0x10,0x10,0x10,0xF0,0xF8,0x00,0x00,0x00,0x00,0x00,0x00,0x0
```

0,

```
0x00,0x00,0x00,0x20,0x20,0x20,0x20,0x3F,0x3F,0x20,0x20,0x20,0x20,0x00,0x00,0x00
```

0,/\*"1",1\*/

```
0x00,0x00,0x60,0x50,0x10,0x08,0x08,0x08,0x08,0x08,0x08,0x98,0xF0,0x70,0x00,0x0
```

0,

0x00,0x00,0x20,0x30,0x28,0x28,0x24,0x24,0x22,0x22,0x21,0x20,0x30,0x18,0x00,0x0

$$0,/^* "2",2^*/$$

0x00,0x00,0x30,0x30,0x08,0x08,0x88,0x88,0x88,0x88,0x58,0x70,0x30,0x00,0x00,0x00

0,

```
0x00,0x00,0x18,0x18,0x20,0x20,0x20,0x20,0x20,0x20,0x31,0x11,0x1F,0x0E,0x00,0x0
```

0,/\*"3",3\*/

```
0x00,0x00,0x00,0x00,0x00,0x80,0x40,0x20,0x10,0xF0,0xF8,0xF8,0x00,0x00,0x00,0x0
```

0,

0x00,0x04,0x06,0x05,0x05,0x04,0x24,0x24,0x24,0x3F,0x3F,0x3F,0x24,0x24,0x24,0x00

,/\*"4",4\*/

```
0x00,0x00,0x00,0xC0,0x38,0x88,0x88,0x88,0x88,0x88,0x88,0x88,0x08,0x08,0x00,0x0
```

0,

0x00,0x00,0x18,0x29,0x21,0x20,0x20,0x20,0x20,0x20,0x30,0x11,0x1F,0x0E,0x00,0x0

0,/\*"5",5\*/

0x00,0x00,0x80,0xE0,0x30,0x10,0x98,0x88,0x88,0x88,0x88,0x88,0x98,0x10,0x00,0x0

0,

0x00,0x00,0x07,0x0F,0x19,0x31,0x20,0x20,0x20,0x20,0x20,0x20,0x11,0x1F,0x0E,0x0

0,/\*"6",6\*/

0x00,0x00,0x30,0x18,0x08,0x08,0x08,0x08,0x88,0x48,0x28,0x18,0x08,0x00,0x0

0,

0x00,0x00,0x00,0x00,0x00,0x00,0x38,0x3E,0x01,0x00,0x00,0x00,0x00,0x00,0x0

0,/\*"7",7\*/

0x00,0x00,0x70,0x70,0xD8,0x88,0x88,0x08,0x08,0x08,0x08,0x98,0x70,0x70,0x00,0x0

0,

0x00,0x0C,0x1E,0x12,0x21,0x21,0x20,0x21,0x21,0x21,0x23,0x12,0x1E,0x0C,0x00,0x0

0,/\*"8",8\*/

0x00,0xE0,0xF0,0x10,0x08,0x08,0x08,0x08,0x08,0x08,0x18,0x10,0xF0,0xC0,0x00,0x0

0,

0x00,0x00,0x11,0x33,0x22,0x22,0x22,0x22,0x22,0x32,0x11,0x1D,0x0F,0x03,0x00,0x0

0,/\*"9",9\*/

};

void delay(uint i)

{

while(--i);

}

void Read\_busy() //BUSY=0

{ //busy p2^7

P2=0xff;

RS=0;//RS/RW=0/1

RW=1;

EN=1;//

while(P2&0x80);//P2.7=0.

```
    EN=0;//  
}
```

```
void write_command(uchar value)  
{  
    P2=0xff;  
    Read_busy();  
    RS=0;//RS/RW=00,  
    RW=0;  
    P2=value;  
    EN=1;//  
    delay(100);  
    EN=0;  
}
```

```
void write_data(uchar value)  
{  
    P2=0xff;  
    Read_busy();  
    RS=1;// RS/RW=10  
    RW=0;  
    P2=value;  
    EN=1;  
    delay(100);  
    EN=0;  
}
```

```
void Set_column(uchar column)  
{  
    column=column&0x3f;
```

```
column=0x40|column;//01000000|column  
write_command(column);  
}
```

```
void Set_line(uchar startline)  
{  
    startline=0xC0|startline;  
    write_command(startline);  
}
```

```
void Set_page(uchar page)  
{  
    page=0xb8|page;  
    write_command(page);  
}
```

```
void display(uchar ss,uchar page,uchar column,uchar *p)  
{  
    uchar i;  
    switch(ss)  
    {  
        case 0: CS1=1;CS2=1;break;  
        case 1: CS1=1;CS2=0;break;  
        case 2: CS1=0;CS2=1;break;  
        default: break;  
    }  
    page=0xb8|page;  
    write_command(page);  
  
    column=column&0x3f;  
    column=0x40|column;
```

```

write_command(column);
for(i=0;i<16;i++)//+1
{
    write_data(p[i]);
}
page++;
write_command(page);
// column--;
write_command(column);
for(i=0;i<16;i++)
{
    write_data(p[i+16]);
}
}
void SetOnOff(uchar onoff)
{
    onoff=0x3e|onoff;//00111110|onoff,
    write_command(onoff);
}
void ClearScreen()
{
    uchar i,j;
    CS2=1;
    CS1=1;
    for(i=0;i<8;i++)
    {
        Set_page(i);
        Set_column(0);
        for(j=0;j<64;j++)
        {

```



```

        write_data(0x00);
    }
}
}

```

```

void InitLCD()//???

```

```

{
    Read_busy();
    CS1=1;CS2=1;
    SetOnOff(0);
    CS1=1;CS2=1;
    SetOnOff(1);
    CS1=1;CS2=1;
    ClearScreen();
    Set_line(0);
}

```

```

bit DS_init()

```

```

{
    bit flag;
    DQ = 0;
    delay_us(255);
    DQ = 1;
    delay_us(40);
    flag = DQ;
    delay_us(150);
    return flag;
}

```

```

uchar read()    //byte

```

```

{

```

```

    uchar i;

    uchar val = 0;
    for (i=0; i<8; i++)
    {
        val >>= 1;

        DQ = 0;

        delay_us(1);

        DQ = 1;

        delay_us(1);

        if (DQ) val |= 0x80;

        delay_us(15);
    }
    return val;
}

void write(char val)    //byte
{
    uchar i;

    for (i=0; i<8; i++)
    {
        DQ = 0;

        delay_us(8);

        val >>= 1;

        DQ = CY;

        delay_us(35);

        DQ = 1;

        delay_us(10);
    }
}

```

```

void PID()
{
    uchar Px,Pp,Pi,Pd,a,b,c;

    uint count;

    Pp = Kp*(Ek-Ek1);
    Pi = Ki*Ek;
    Pd = Kd*(Ek-2*Ek1+Ek2);
    Px = Pp+Pi+Pd;
    res = Px;

    a=res/100;
    b=res%100/10;
    c=res%10;

    display(1,4,2*16,shu[a]);delay(255);
    display(1,4,3*16,shu[b]);delay(255);
    display(2,4,0*16,shu[c]);delay(255);
    Ek2 = Ek1;
    Ek1 = Ek;
    count = 0;
    if(res>Pmax)
        res =Pmax ;
    while((count++)<=res)
    {
        up = 1;
        delay_us(250);
        delay_us(250);
    }
    while((count++)<=Pmax)
    {
        up = 0;

```

```
        delay_us(250);
        delay_us(250);
    }

}
```

```
void main()
{
    uchar aim,low,high,b,c;
    uint result;
    InitLCD();
    Set_line(0);
    aim = 40;
    Kp = 4;
    Ki = 5;
    Kd = 2;
    Pmax = 5;
    Ek1 = 0;
    Ek2 = 0;
    res = 0;

    while(1)
    {
        if(s1 == 0)
            aim++;
        if(s2 == 0)
            aim--;
        while(DS_init());
        write(0xcc);
        write(0x44);
    }
}
```

```

delay(600);
while(DS_init());
write(0xcc);
        write(0xBE);

low = read();
high = read();
delay(255);
result = high;
result <<= 8;
result |= low;
result >>= 4 ; //result /= 16;


Ek = aim - result;


b=result/10;
c=result%10;


        display(1,0,3*16,shu[b]);delay(255);
display(2,0,0*16,shu[c]);delay(255);
b=aim/10;
c=aim%10;
        display(1,2,3*16,shu[b]);delay(255);
display(2,2,0*16,shu[c]);delay(255);
if(aim>=result)
        PID();
else
        up = 0;

}

}

```