LCD1602

LCD1602液晶显示器是广泛使用的一种字符型液晶显示模块。它是由字符型液晶显示屏（LCD）、控制驱动主电路HD44780及其扩展驱动电路HD44100，以及少量电阻、电容元件和结构件等装配在PCB板上而组成。不同厂家生产的LCD1602芯片可能有所不同，但使用方法都是一样的。为了降低成本，绝大多数制造商都直接将裸片做到板子上。

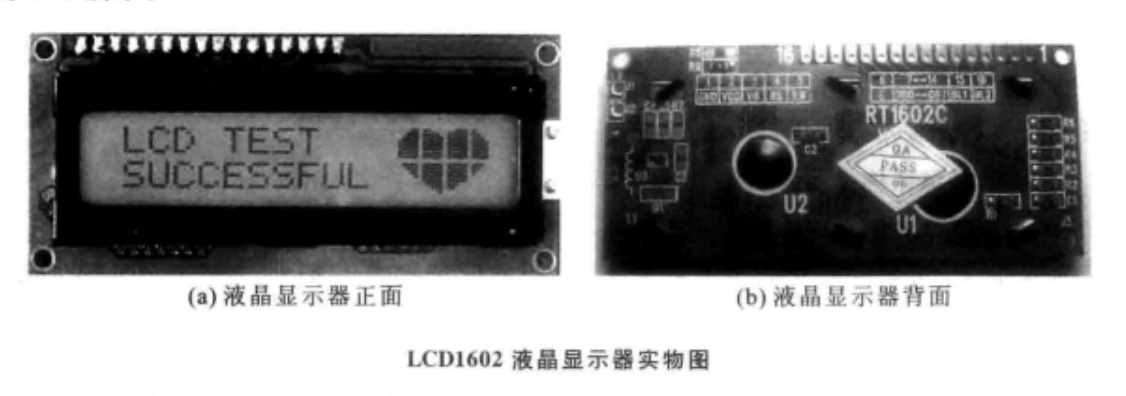
字符型液晶显示原理

[点阵图](https://baike.baidu.com/item/%E7%82%B9%E9%98%B5%E5%9B%BE/7542326)形式液晶由M×N个显示单元组成，假设LCD显示屏有64行，每行有128列，每8列对应1字节的8位，即每行由16字节，共16×8=128个点组成。显示屏上64×16个显示单元与显示RAM区的1024字节相对应，每一字节的内容与显示屏上相应位置的亮暗对应。例如显示屏第一行的亮暗由RAM区的000H~00FH的16字节的内容决定，当（000H）=FFH时，屏幕左上角显示一条短亮线，长度为8个点；当（3FFH）=FFH时，屏幕右下角显示一条短亮线；当（000H）=FFH，（001H）=00H，（002H）=FFH…，（00EH）=FFH，（00FH）=00H时，在屏幕的顶部显示一条由8条亮线和8条暗线组成的虚线。这就是LCD显示的基本原理。

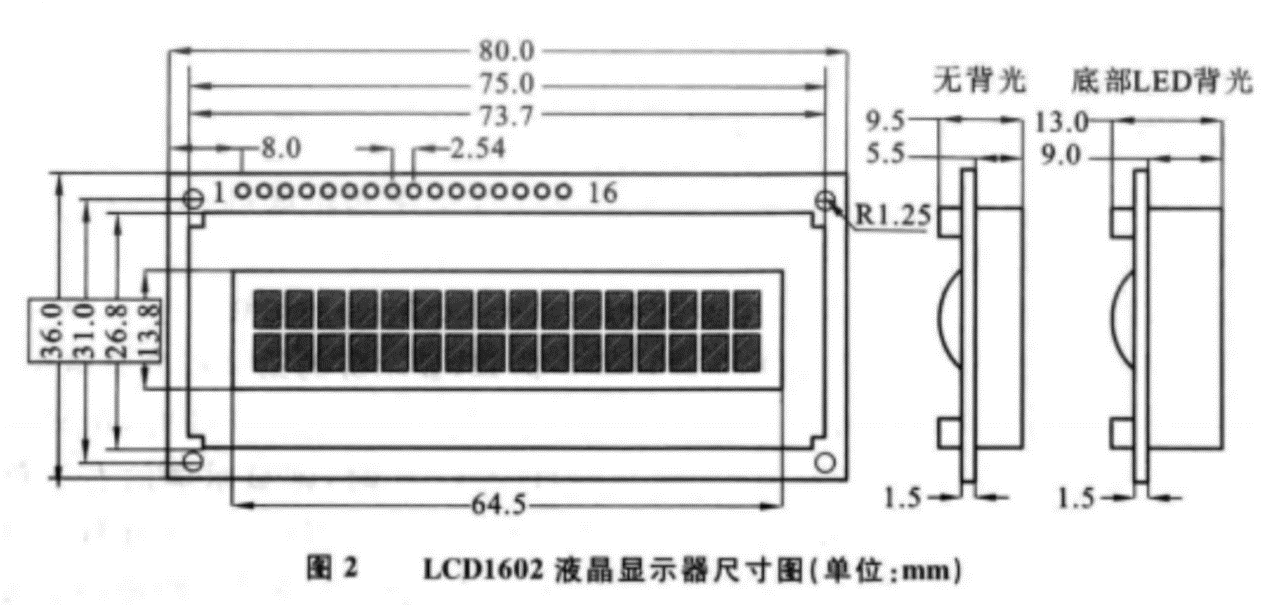
字符型液晶显示模块是一种专门用于显示字母、数字和符号等的点阵式LCD，常用16×1，16×2，20×2和40×2等的模块。一般的LCD1602字符型液晶显示器的内部控制器大部分为HD44780，能够显示英文字母、[阿拉伯数字](https://baike.baidu.com/item/%E9%98%BF%E6%8B%89%E4%BC%AF%E6%95%B0%E5%AD%97/426445" \t "_blank)、日文片假名和一般性符号。

外形尺寸

LCD1602字符型液晶显示器实物外形如图1所示。

[](https://baike.baidu.com/pic/LCD1602/6014393/0/b999a9014c086e062620df5f0d087bf40ad1cb1e?fr=lemma&ct=single)图1

LCD1602分为带背光和不带背光两种，其控制器大部分为HD44780。带背光的比不带背光的厚，是否带背光在实际应用中并无差别，具体的鉴别办法可参考图2所示的器件尺寸示意图。

[](https://baike.baidu.com/pic/LCD1602/6014393/0/d000baa1cd11728ba5ad6c36c7fcc3cec3fd2c26?fr=lemma&ct=single)

技术参数

（1）显示容量：16×2个字符。

（2）芯片工作电压：4.5~5.5V。

（3）工作电流：2.0mA（5.0V）。

（4）模块最佳的工作电压：5.0V。

（5）字符尺寸：2.95mm×4.35mm（宽×高）[3]  。

引脚功能

LCD1602采用标准的14脚（无背光）或16脚（带背光）接口，各引脚接口说明见表1。

**表1 LCD引脚功能表**

|  |  |  |
| --- | --- | --- |
|  | | |
|  | | |
| **编号** | **符号** | **引脚说明** | | **标号** | **符号** | **引脚说明** |
| 1 | VSS | 电源地 | | 9 | D2| | 数据 |
| 2 | VDD | 电源正极 | | 10 | D3 | 数据 |
| 3 | VL | 液晶显示偏压 | | 11 | D4 | 数据 |
| 4 | RS | 数据/命令选择 | | 12 | D5 | 数据 |
| 5 | R/W | 读/写选择 | | 13 | D6 | 数据 |
| 6 | E | 使能信号 | | 14 | D7 | 数据 |
| 7 | D0 | 数据 | | 15 | BLA | 背光源正极 |
| 8 | D1 | 数据 | | 16 | BLK | 背光源负极 |

各引脚的功能介绍如下。

**·**引脚1：VSS为地电源。

**·**引脚2：VDD接5V正电源。

**·**引脚3：VL为[液晶显示器对比度](https://baike.baidu.com/item/%E6%B6%B2%E6%99%B6%E6%98%BE%E7%A4%BA%E5%99%A8%E5%AF%B9%E6%AF%94%E5%BA%A6/1162217" \t "_blank)调整端，接正电源时对比度最弱，接地时对比度最高，对比度过高时会产生“鬼影”现象，使用时可以通过一个10kQ的电位器调整其对比度。

**·**引脚4：RS为寄存器选择脚，高电平时选择[数据寄存器](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E5%AF%84%E5%AD%98%E5%99%A8/1944231" \t "_blank)、低电平时选择[指令寄存器](https://baike.baidu.com/item/%E6%8C%87%E4%BB%A4%E5%AF%84%E5%AD%98%E5%99%A8/3219483)。

**·**引脚5：R/W为读/写信号线，高电平时进行读操作，低电平时进行写操作。当RS和R/W共同为低电平时可以写入指令或显示地址；当RS为低电平，R/W为高电平时，可以读忙信号；当RS为高电平，R/W为低电平时，可以写入数据。

**·**引脚6：E端为使能端，当E端由高电平跳变为低电平时，液晶模块执行命令。

**·**引脚7~14：D0~D7为8位双向数据线。

**·**引脚15：背光源正极。

**·**引脚16：背光源负极。

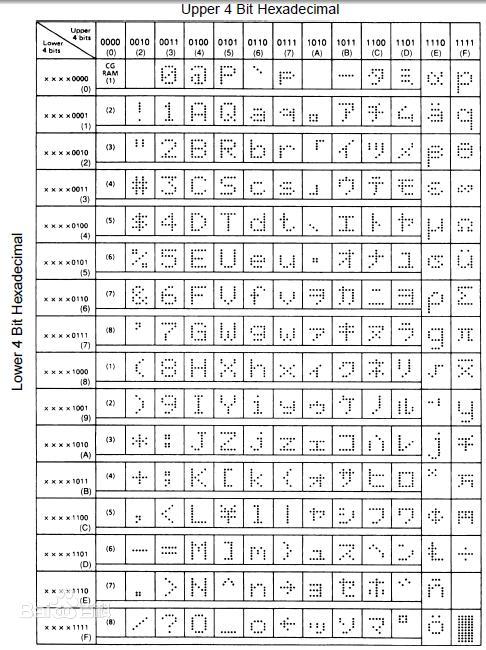
字符集

1602液晶模块内部的字符发生[存储器](https://baike.baidu.com/item/%E5%AD%98%E5%82%A8%E5%99%A8" \t "_blank)（[CGROM](https://baike.baidu.com/item/CGROM))已经存储了160个不同的[点阵](https://baike.baidu.com/item/%E7%82%B9%E9%98%B5" \t "_blank)字符图形，显示字符是从0x20开始的，这些字符有：[阿拉伯数字](https://baike.baidu.com/item/%E9%98%BF%E6%8B%89%E4%BC%AF%E6%95%B0%E5%AD%97/426445)、英文字母的大小写、常用的符号、和日文假名等，每一个字符都有一个固定的代码，比如大写的英文字母“A”的代码是01000001B（41H），显示时模块把地址41H中的点阵字符图形显示出来，我们就能看到字母“A”。

因为1602识别的是ASCII码，试验可以用ASCII码直接赋值，在[单片机](https://baike.baidu.com/item/%E5%8D%95%E7%89%87%E6%9C%BA)编程中还可以用字符型[常量](https://baike.baidu.com/item/%E5%B8%B8%E9%87%8F)或变量赋值，如'A’。

0x00-0x0f是留给自定义字符的（[CGRAM](https://baike.baidu.com/item/CGROM))，0x00-0x07和0x08-0x0f的内容是一样的。

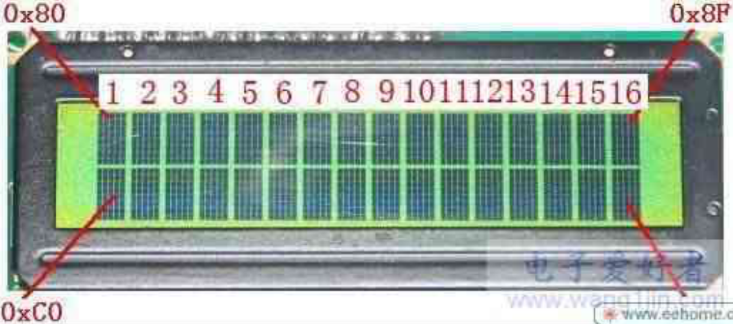
以下是1602的16进制ASCII码表：



显示地址码

1602字符液晶显示可分为上下两部分各16位进行显示，处于不同行时的字符显示地址如下

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 显示字符 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
| 第一行地址 | 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 0A | 0B | 0C | 0D | 0E | 0F |
| 第二行地址 | 40 | 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 | 49 | 4A | 4B | 4C | 4D | 4E | 4F |



80+地址即为上述显示地址

指令集

LCD1602液晶模块内部的控制器共有11条控制指令，见表2。

**表2 LCD1602控制指令**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **序号** | **指令** | **RS** | **R/W** | **D7** | **D6** | **D5** | **D4** | **D3** | **D2** | **D1** | **D0** |
| 1 | 清屏 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 2 | 光标复位 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | x |
| 3 | 输入方式设置 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | I/D | S |
| 4 | 显示开关控制 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | D | C | B |
| 5 | 光标或字符移位控制 | 0 | 0 | 0 | 0 | 0 | 1 | S/C | R/L | x | x |
| 6 | 功能设置 | 0 | 0 | 0 | 0 | 1 | DL | N | F | x | x |
| 7 | 字符发生存储器地址设置 | 0 | 0 | 0 | 1 | 字符发生存储器地址 | | | | | |
| 8 | 数据存储器地址设置 | 0 | 0 | 1 | 显示数据存储器地址 | | | | | | |
| 9 | 读忙标志或地址 | 0 | 1 | BF | 计数器地址 | | | | | | |
| 10 | 写入数据至CGRAM或DDRAM | 1 | 0 | 要写入的数据内容 | | | | | | | |
| 11 | 从CGRAM或DDRAM中读取数据 | 1 | 1 | 读取的数据内容 | | | | | | | |

LCD1602液晶模块的读/写操作、显示屏和光标的操作都是通过指令编程来实现的（其中，1为高电平，0为低电平），分别介绍如下。

（1）指令1：清屏。指令码01H，光标复位到地址00H。

（2）指令2：光标复位。光标复位到地址00H。

（3）指令3：输入方式设置。其中，I/D表示光标的移动方向，高电平右移，低电平左移；S表示显示屏上所有文字是否左移或右移，高电平表示有效，低电平表示无效。

（4）指令4：显示开关控制。其中，D用于控制整体显示的开与关，高电平表示开显示，低电平表示关显示；C用于控制光标的开与关，高电平表示有光标，低电平表示无光标；B用于控制光标是否闪烁，高电平闪烁，低电平不闪烁。

（5）指令5：光标或字符移位控制。其中，S/C表示在高电平时移动显示的文字，低电平时移动光标。

（6）指令6：功能设置命令。其中，DL表示在高电平时为8位总线，低电平时为4位总线；N表示在低电平时为单行显示，高电平时双行显示；F表示在低电平时显示5×7的点阵字符，高电平时显示5×10的点阵字符。

（7）指令7：字符发生器RAM地址设置。

（8）指令8：DDRAM地址设置。

（9）指令9：读忙信号和光标地址。其中，BF为忙标志位，高电平表示忙，此时模块不能接收命令或数据，如果为低电平则表示不忙。

（10）指令10：写数据。

（11）指令11：读数据。

连接方式

LCD1602与[单片机](https://baike.baidu.com/item/%E5%8D%95%E7%89%87%E6%9C%BA/102396" \t "_blank)的连接有两种方式，一种是直接控制方式，另一种是所谓的间接控制方式。它们的区别只是所用的数据线的数量不同，其他都一样。

**1.直接控制方式**

LCD1602的8根数据线和3根控制线E，RS和R/W与单片机相连后即可正常工作。一般应用中只须往LCD1602中写入命令和数据，因此，可将LCD1602的R/W读/写选择控制端直接接地，这样可节省1根数据线。VO引脚是液晶对比度调试端，通常连接一个10kΩ的[电位器](https://baike.baidu.com/item/%E7%94%B5%E4%BD%8D%E5%99%A8/6530251" \t "_blank)即可实现对比度的调整；也可采用将一个适当大小的电[阻](https://baike.baidu.com/item/%E9%98%BB/5874163)从该引脚接地的方法进行调整，不过电阻的大小应通过调试决定。

**2.间接控制方式**

间接控制方式也称为四线制工作方式，是利用HD44780所具有的4位数据总线的功能，将电路接口简化的一种方式。为了减少接线数量，只采用引脚DB4~DB7与单片机进行通信，先传数据或命令的高4位，再传低4位。采用四线并口通信，可以减少对[微控制器](https://baike.baidu.com/item/%E5%BE%AE%E6%8E%A7%E5%88%B6%E5%99%A8/6688343" \t "_blank)I/O的需求，当设计产品过程中单片机的I/O资源紧张时，可以考虑使用此方法。

如何看懂时序图？

操作时序永远使用是任何一片IC芯片的最主要的内容。一个芯片的所有使用细节都会在它的官方器件手册上包含。所以使用一个器件事情，要充分做好的第一件事就是要把它的器件手册上有用的内容提取，掌握。

我们首先来看1602的引脚定义,1602的引脚是很整齐的SIP单列直插封装，所以器件手册只给出了引脚的功能数据表：

我们只需要关注以下几个管脚：

3脚：VL，液晶显示偏压信号，用于调整LCD1602的显示对比度，一般会外接电位器用以调整偏压信号，注意此脚电压为0时可以得到最强的对比度。

4脚：RS，数据/命令选择端，当此脚为高电平时，可以对1602进行数据字节的传输操作，而此脚为低电平时，则是进行命令字节的传输操作。命令字节，即是用来对LCD1602的一些工作方式作设置的字节;数据字节，即使用以在1602上显示的字节。值得一提的是，LCD1602的数据是8位的。

5脚：R/W，读写选择端。当此脚为高电平可对LCD1602进行读数据操作，反之进行写数据操作。笔者认为，此脚其实用处不大，直接接地永久置为低电平也不会影响其正常工作。但是尚未经过复杂系统验证，保留此意见。

6脚：E，使能信号，其实是LCD1602的数据控制时钟信号，利用该信号的上升沿实现对LCD1602的数据传输。

7~14脚：8位并行数据口，使得对LCD1602的数据读写大为方便。

现在来看LCD1602的操作时序：

在此，我们可以先不读出它的数据的状态或者数据本身。所以只需要看两个写时序：

①当我们要写指令字，设置LCD1602的工作方式时：需要把RS置为低电平，RW置为低电平，然后将指令送到数据口D0~D7，最后E引脚一个高脉冲将数据写入。

②当我们要写入数据字，在1602上实现显示时：需要把RS置为高电平，RW置为低电平，然后将数据送到数据口D0~D7，最后E引脚一个高脉冲将数据写入。

发现了么，写指令和写数据，差别仅仅在于RS的电平不一样而已。以下是LCD1602的时序图：

大家要慢慢学会看时序图，要知道操作一个器件的精华便蕴藏在其中，看懂看准了时序，你操控这个芯片就是非常容易的事了。1602的时序是我见过的一个最简单的时序：

1、注意时间轴，如果没有标明(其实大部分也都是不标明的)，那么从左往右的方向为时间正向轴，即时间在增长。

2、上图框出并注明了看懂此图的一些常识：

(1).时序图最左边一般是某一根引脚的标识，表示此行图线体现该引脚的变化，上图分别标明了RS、R/W、E、DB0~DB7四类引脚的时序变化。

(2).有线交叉状的部分，表示电平在变化，如上所标注。

(3).应该比较容易理解，如上图右上角所示，两条平行线分别对应高低电平，也正好吻合(2)中电平变化的说法。

(4).上图下，密封的菱形部分，注意要密封，表示数据有效，Valid Data这个词也显示了这点。

3、需要十分严重注意的是，时序图里各个引脚的电平变化，基于的时间轴是一致的。一定要严格按照时间轴的增长方向来精确地观察时序图。要让器件严格的遵守时序图的变化。在类似于18B20这样的单总线器件对此要求尤为严格。

4、以上几点，并不是LCD1602的时序图所特有的，绝大部分的时序图都遵循着这样的一般规则，所以大家要慢慢的习惯于这样的规则。

也许你还注意到了上面有许多关于时间的标注，这也是个十分重要的信息，这些时间的标注表明了某些状态所要维持的最短或最长时间。因为器件的工作速度也是有限的，一般都跟不上主控芯片的速度，所以它们直接之间要有时序配合。话说现在各种处理器的主频也是疯狂增长，日后搞不好出现个双核单片机也不一定就是梦话。下面是时序参数表：

大家要懂得估计主控芯片的指令时间，可以在官方数据手册上查到MCU的一些级别参数。比如我们现在用AVR M16做为主控芯片，外部12MHz晶振，指令周期就是一个时钟周期为(2/12MHz)s，所以至少确定了它执行一条指令的时间是us级别的。我们看到，以上给的时间参数全部是ns级别的，所以即便我们在程序里不加延时程序，也应该可以很好的配合LCD1602的时序要求了。怎么看这个表呢?很简单，我们在时序图里可以找到TR1，对应时序参数表，可以查到这个是E上升沿/下降沿时间，最大值为25ns，表示E引脚上的电平变化，必须在最大为25ns之内的时间完成。大家看是不是这个意思?

现在我来解读我对这个时序图的理解：

当要写命令字节的时候，时间由左往右，RS变为低电平，R/W变为低电平，注意看是RS的状态先变化完成。然后这时，DB0~DB7上数据进入有效阶段，接着E引脚有一个整脉冲的跳变，接着要维持时间最小值为tpw=400ns的E脉冲宽度。然后E引脚负跳变，RS电平变化，R/W电平变化。这样便是一个完整的LCD1602写命令的时序。

看完了数据手册，有木有发现灵感？是不是好多都是可以看着自己就可以敲出来？

好了，我们开始写程序：

写液晶的时候分为以下步骤：

1、写指令（看看数据手册里有的）

void writecmd(uchar com)

{

E=0; //为什么要写这一条呢，看看时序图就清楚了

RS=0;

RW=0;

P0=com;

delay(2); //延时一下会稳定很多，后面一条也是一样的

E=1;

delay(2);

E=0;

}

//以上的P0就是液晶送数据的端口，当然你可以是P2口，看你的 电路是怎么样那就怎么改，至于为什么是com，这是个自定义形式参数来的，也可以改成别的，但是不能是关键字，这个参数传的就是你想要写的指令；下面写数据的P0是你想要写的数据。详情看看大程序便知

2、写数据（同样数据手册也有）

void writedata(uchar dat)//写数据，函数名随便你定义看得懂就行

{

E=0; //写数据和写命令的方法差不多，复制过来。照着数据手 //册的写数据要求去写！

RS=1;

RW=0;

P0=dat;

delay(2);

E=1;

delay(2);

E=0;

}

3、初始化（一样的，数据手册也有，照写）

void init(void)

｛ delay(15);

writecmd(0x38);

delay(5);

writecmd(0x38);

delay(5);

writecmd(0x38);

writecmd(0x38);

writecmd(0x08);

writecmd(0x01);

writecmd(0x06);

writecmd(0x0c);

}

好了一个液晶的基本操作就是这三部分组成的，不难吧？

接下来来看看我们如何显示如图所示的字符：

#include<reg51.h>

#define uchar unsigned char

#define uint unsigned int

//这三个引脚参考资料

sbit E=P2^7; //1602使能引脚

sbit RW=P2^6; //1602读写引脚

sbit RS=P2^5; //1602数据/命令选择引脚

uchar tab1[]=" www.51hei.com "; //要显示什么就在这儿改

uchar tab2[]="LCD1602 test ok!"; //[size=15.5555562973022px]要显示什么就在这儿改

void delay(uint del) //延时1ms

{

uint i,j;

for(i=0;i<del;i++)

for(j=0;j<=168;j++);

}

void writecmd(uchar com) //写指令

{

E=0;

RS=0;

RW=0;

P0=com;

delay(2);

E=1;

delay(2);

E=0;

}

void writedata(uchar dat)//写数据

{

E=0;

RS=1;

RW=0;

P0=dat;

delay(2);

E=1;

delay(2);

E=0;

}

void init(void) //初始化

{

delay(15);

writecmd(0x38);

delay(5);

writecmd(0x38);

delay(5);

writecmd(0x38);

writecmd(0x38);

writecmd(0x08);

writecmd(0x01);

writecmd(0x06);

writecmd(0x0c);

}

void main()

{ uchar j;

init();

writecmd(0x80); //告诉液晶在哪个地方显示 （设置地址指针）//80H+00

for(j=0;j<16;j++) //有16个字符，要循环16次

{

writedata(tab1[j]);

delay(2);

}

writecmd(0x80+0x40); //再设置第二行的起始位置80H+40H

for(j=0;j<16;j++) //有16个字符，要循环16次

{ writedata(tab2[j]);

delay(2);

}

while(1); //让液晶显示就一直停在这里。

}

好了，你们可以照着我这个程序去修改得到你们想要的显示结果，同样也可以发挥一下你们的想法，如何让液晶进行移屏显示呢？怎么样做出移动的效果呢？有待大家的努力噢！小编最主要的目的是要告诉大家，任何模块拿起来都不要惧怕，把数据手册看一遍，看不懂再看一遍，然后看看别人是怎么写的，自己动手改一遍，再写一遍，转化为自己的东西，这样就可以学到经验了。