

液晶显示器(LCD)具有功耗低、体积小、重量轻、超薄等许多其它显示器无法比拟的优点,近几年来被广泛用于单片机控制的智能仪器、仪表和低功耗电子产品中。LCD 可分为段位式LCD、字符式LCD 和点阵式LCD。其中,段位式LCD 和字符式LCD 只能用于字符和数字的简单显示,不能满足图形曲线和汉字显示的要求;而点阵式LCD 不仅可以显示字符、数字,还可以显示各种图形、曲线及汉字,并且可以实现屏幕上下左右滚动,动画功能,分区开窗口,反转,闪烁等功能,用途十分广泛。本文介绍了点阵式液晶显示器MGLS12864 与单片机的接口及编程的方法,同时介绍了创建 $8 \times 16$  字符和 $16 \times 16$  点阵汉字的方法,及常用的字符显示和汉字显示程序。

这里着重介绍液晶显示器与单片机的接口技术。有关

MGLS12864 图形液晶引脚功能及控制器HD61202的接口时序波形,可查看图形液晶显示器产品有关手册。

单片机可以通过数据总线与控制信号直接采用存储器访问形式、I/O 设备访问形式控制该液晶显示模块。本文以华邦公司的W78E58 为例,它是51 系列单片机兼容的微控制器,其内部有32KB 的FLASH EEPROM,用户编制的程序及需要显示的英文字母、数字、汉字、曲线和图形都可以存储在里面,免去了扩展外部存储器的麻烦,使得以W78E58 单片机为核心的控制系统电路更简单。因此十分适用于液晶显示。

MGLS12864 与W78E58 单片机接口电路如图1 所示。该图采用直接访问方式,单片机通过低位地址A2控制CSA;A3 控制CSB,以选通液晶显示屏上各区的控制器HD61202;同时W78E58 用地址A1 作为R/W 信号控制数据总线的数据流向;用地址A0 作为D/I 信号控制寄存器的选择;E 信号由W78E58 的读信号/RD 和写信号/WR 合成产生;另外单片机的复位脚(9 脚)经反相器后连接到液晶显示器复位脚(17 脚/RST),当单片机上电复位或手动复位时,液晶显示器同时也复位;从而实现了W78E58 对内置HD61202图形液晶显示器模块的电路连接。电路中LCD 电源控制端V0 是用来调节显示屏灰度的,调节该端的电压,可改变显示屏字符、图形的颜色深浅。

单片机对液晶显示模块的操作可分为两部分,即左半屏和右半屏操作。下面是根据图1 的连接确定对应左半屏(前 $64 \times 64$ )和右半屏(后 $64 \times 64$ )操作地址:

```
CWADR1 EQU 0004H 左半屏写指令代码地址
DWADR1 EQU 0005H 左半屏写显示数据地址
CRADR1 EQU 0006H 左半屏读状态字地址
DRADR1 EQU 0007H 左半屏读显示数据地址
CWADR2 EQU 0008H 右半屏写指令代码地址
DWADR2 EQU 0009H 右半屏写显示数据地址
CRADR2 EQU 000AH 右半屏读状态字地
DRADR2 EQU 000BH 右半屏读显示数据地址
```

图1 MGLS12864 与W78E58 接口图

液晶控制器HD61202 一共有七条指令,从作用上可分为两类,显示状态设置指令和数据读/写操作指令。详见指令系统可查看图形液晶显示器产品有关手册。显示起始行设置中L5~L0 为显示起始行的地址,取值在0~3FH(1~64 行)范围内。页面地址设置中P2~P0 为选择的页面地址,取值范围为0~7H,代表1~8 页。列地址设置中C5~C0 为Y 地址计数器的内容,取值在0~3FH(1~64 行)范围内。

显示器上128 点 $\times$ 64 点,每8 点为一字节数据,都对应着显示数据RAM(在HD61202 芯片内),一点



```
else { beginbyte="15"; shiftn="15";}
```

```
for(j=0;j<8;j++)
```

```
ascii8x16[m]=(ascii8x16[m]+ (buf[beginbyte-j]>>(shiftn-m))&0x01)<<1;
```

也可选用UCDOS 的ASC16 文件做字模库。ASC16 文件的字符为8×16 点阵。所有字符按照ASCII 值从小到大升序排列。计算字符首地址的公式：字符首地址=字符的ASCII 码值×16+字模库首地址。

(2) 建立所用到的16×16 点阵字模库。汉字字符可选用UCDOS 的HZK16 文件做字模库。HZK16 文件的字符为16×16 点阵。所有字符按照区位码从小到大升序排列。计算汉字字符首地址的公式如下：汉字首地址=((区码-1)×94+位码-1)×32。作者用C 语言编写的读取UCDOS 点阵字库字模程序，完成字模读取、

数据重新排列，并按MCS-51 汇编程序的要求写成相应格式的文本文件。

```
p=((quma-1)*94+weima-1)*32;
```

```
cclibfile="fopen"("HZK16
```

```
", "rb");
```

```
fseek(cclibfile, (long)p, SEEK_SET);
```

```
fread(buf, sizeof(unsigned char), 32, cclibfile); /*读32 字节点阵数据*/
```

```
for(m=0;m<32;m++) { /*点阵数据转换成LCD 格式数据*/
```

```
if(m<8) { beginbyte="14"; shiftn="7";}
```

```
else if( m>= 8 && m<16 ) { beginbyte="15"; shiftn="15";}
```

```
else if( m>=16 && m<24 ) { beginbyte="30"; shiftn="23";}
```

```
else { beginbyte="31"; shiftn="31";}
```

```
for(j=0;j<8;j++)
```

```
hzk16x16[m]=(hzk16x16[m]+ (buf[beginbyte-2*j] >>(shiftn-m))&0x01)<<1;
```

```
}
```

(3) 常用图形（如产品商标等）的点阵图形的建立。对已有的图形可采用扫描仪进行扫描，然后用图形处理软件进行处理，再将BMP 格式文件转换成MCS-51 的汇编文件的格式。

以上所有的字模数据都存放在单片机W78E58 的程序存储器中，如用到的汉字、图形较多，可选用较大容量的程序存储器。

通用子程序：通用子程序分左半屏、右半屏写指令代码子程序和写显示数据子程序。液晶显示驱动器HD16202 内部有个忙标志寄存器，当BF=1 时，表示内部操作正在运行，不能接受外部数据或指令。下面子程序中设指令代码寄存器为COMM, 数据寄存器为DATA。

```
(COMM EQU 20H /*指令寄存器*/ DATA EQU 21H /*数据寄存器*/)
```

(1) 左半屏写指令子程序

```
WR_CMD1: MOV DPTR, #CRADR1 ; /*读状态字口地址*/ WAIT1: MOVX A, @DPTR ; /* 读状态字 */
```

```
JB ACC. 7, WAIT1 ; /*判忙标志BF, 如BF=1 忙, 等待*/
```

```
MOV DPTR, #CWADR1 ; /*写指令字口地址*/
```

```
MOV A, COMM ; /*取指令代码*/
```

```
MOVX @DPTR, A ; /*写指令代码*/
```

```
RET
```

(2) 左半屏写数据子程序

```
WR_DATA1: MOV DPTR, #CRADR1 ; /*读状态字口地址*/
```

```
WAIT11: MOVX A, @DPTR ; /* 读状态字 */
```

```
JB ACC. 7, WAIT11 ; /*判忙标志BF, 如BF=1 忙, 等待*/
```

```

MOV DPTR, #DWADR1 ; /*写数据字口地址*/
MOV A, DATA ; /*取数据/
MOVX @DPTR, A ; /*写数据*/
RET

```

(3)右半屏写指令子程序WR\_CMD2 和右半屏写数据子程序WR\_DATA2 的编制同左半屏子程序相同，只是对应口地址不同。

8X16 字符显示子程序：MGLS12864 液晶显示屏由二片HD61202 控制，LCD 显示中应尽量避免一个字符一半在左半屏显示，另一半在右半屏显示的情况。设列地址寄存器为COLUMN, 页地址寄存器为PAGE, 要显示的字符代码寄存器为ASCII CODE, W78E58 内RAM28H-RAM37H 共16 个字节存放8×16 的点阵数据, 生成的8×16 点阵库文件存放在单片机W78E58 存储器中的首地址定义为ASCII\_DOT816。

```

DISP_ASCII816: MOV DPTR, #ASCII_DOT816 ; 8×16 点阵库首地址

```

```

MOV A, ASCII CODE ; 显示字符代码ASCII CODE

```

```

MOV B, #16 ; 每个字符点阵占16 个字节

```

```

MUL AB ; 计算显示字符在字库的首地址

```

```

ADD A, DPL

```

```

MOV DPL, A

```

```

MOV A, DPH

```

```

ADDC A, B

```

```

MOV DPH, A

```

```

; MOV R0, #28H ; 将点阵数据放到RAM28H-RAM37H

```

```

MOV R2, #00H

```

```

LP_MOVDOT16: MOV A, R2

```

```

MOVC A, @A+DPTR

```

```

MOV @R0, A ; 如要将字符反显（黑底白字），则读出点

```

```

INC R0 ; 阵数据后求反放入单片机的RAM 中

```

```

INC R2

```

```

CJNE R2, #16, LP_MOVDOT16

```

```

;

```

```

PUSH COLUMN

```

```

MOV A, COLUMN ; 显示列数COLUMN 是否在右半屏

```

```

CJNE A, #64, ASCII_IF64

```

```

ASCII_IF64: JNC ASCII_YGE64

```

```

MOV DPTR, #CWADR1 ; 在左半屏时，选择左半屏写指令代码地址

```

```

CLR FIRST0_SECOND1_BIT ; 左半屏列数标志BIT=0

```

```

SJMP ALL_COLUMN

```

```

ASCII_YGE64: CLR C

```

```

SUBB A, #64

```

```

MOV COLUMN, A

```

```

MOV DPTR, #CWADR2 ; 在右半屏时，选择右半屏写指令代码地址

```

```

SETB FIRST0_SECOND1_BIT ; 右半屏列数标志BIT=1

```

```

ALL_COLUMN: MOV A, PAGE

```

```

ADD A, #10111000B ; 设置页地址命令

```

```

MOVX @DPTR, A

```

```

MOV A, COLUMN ; 设置列地址命令

```

```

ADD A, #01000000B

```

