

## ●主题论文

# C51 语言与图形点阵液晶显示模块 FM12864I 的接口控制技术

中国人民解放军第六九零三工厂技术开发处 高健

## Interface Control Technology of C51 and Graphics Matrix LCD FM12864I

Gao Jian

摘要：FM12864I 液晶显示模块内置有 HD61202 液晶显示控制驱动器，可直接与 8 位微处理器相连。文中给出了 MPU 与 FM12864I 的接口电路，同时根据内置 HD61202 液晶驱动器的汉字显示指令特性，给出了相应的控制软件程序。

关键词：液晶；图形显示；FM12864I；C51 语言

分类号：TN873 文献标识码：B 文章编号：1006-6977(2002)05-0059-05

### 1 模块简介

FM12864I 是一种图形点阵式液晶显示器，它主要由行驱动器/列驱动器及 128×64 全点阵液晶显示器 HD61202 组成。既可显示图形，也可显示 8×4 个 (16×16 点阵) 汉字。其主要技术参数和性能如下：

●电源为 +5V，模块内自带 -10V 负压，可作为 LCD 的驱动电压；

●显示 128(列)×64(行)点阵；

●可全屏幕点阵显示；

●具有七种操作指令；

●与 CPU 接口采用 8 位数据总线并行输入输出和 8 条控制线；

●占空比为 1/64；

●工作温度范围为 0℃ ~ +50℃，存储温度为 -20℃ ~ +70℃；

●模块体积：54×50×6.5mm。

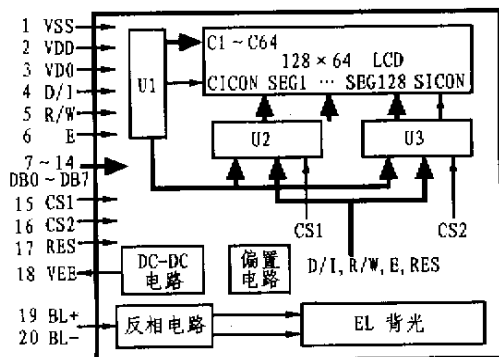


图 1 FM12864I 的内部结构

FM12864I 模块的主要硬件构成框图如图 1 所示。图 2 所示为其行/列驱动器显示 RAM 的地址结构。

### 2 引脚功能和操作时序

FM12864I 模块的外部引脚功能如表 1 所列。图 3 所示是 FM12864I 的读写操作时序图。

### 3 模块指令系统

FM12864I 模块共有 7 种指令，分别介绍如下：

●显示开/关指令格式：

R/W	D/I	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	1	1	1	1	1	1/0

当 DB0=1 时，LCD 显示 RAM 中的内容；DB0=0 时，关闭显示。

●显示起始行 (ROW) 设置指令格式：

R/W	D/I	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	1	1	显示起始行 (0~63)					

		CS1=1					CS2=1					行号
Y=		0	1	...	62	63	0	1	...	62	63	
X=0	DB0	DB0	DB0	DB0	DB0	DB0	DB0	DB0	DB0	DB0	DB0	0
	DB7	DB7	DB7	DB7	DB7	DB7	DB7	DB7	DB7	DB7	DB7	7
I	DB0	DB0	DB0	DB0	DB0	DB0	DB0	DB0	DB0	DB0	DB0	8
	DB7	DB7	DB7	DB7	DB7	DB7	DB7	DB7	DB7	DB7	DB7	55
X=7	DB0	DB0	DB0	DB0	DB0	DB0	DB0	DB0	DB0	DB0	DB0	56
	DB7	DB7	DB7	DB7	DB7	DB7	DB7	DB7	DB7	DB7	DB7	63

图 2 FM12864I 中行/列驱动器显示 RAM 的地址结构

表1 FM12864I 的引脚功能

管脚号	管脚名称	参数值	管脚功能描述
1	VSS	0	电源地
2	VDD	+5V	电源电压
3	V0	-	液晶显示器驱动电压
4	D/I	H/L	H 表示显示数据 L 表示显示指令数据
5	R/W	H/L	E = H, R/W = H 表示数据 被读到 DB7 ~ DB0 ; E = H → L R/W = L 表示数据被写 到指令和数据寄存器
6	E	H/L	R/W = L, E 信号下降沿 锁存 DB7 ~ DB0 ; R/W = H, E = H DDRAM 数据读 到 DB7 ~ DB0
7 ~ 14	DB0 ~ DB7	H/L	数据线
15	CS1	H/L	H 选择左半屏
16	CS2	H/L	H 选择右半屏
17	RET	H/L	复位信号, 接高电平
18	VOUT	-10V	LCD 驱动负电压
19	LED +	-	LED 背光板电源
20	LED -	-	LED 背光板电源

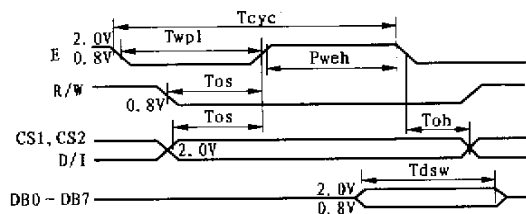


图3 FM12864I 的读写操作时序

该指令设置了对应液晶屏最上一行的显示 RAM 的行号, 有规律的改变显示起始行, 可以使 LCD 实现滚屏显示。

#### ●页(PAGE)设置指令格式:

R/W	D/I	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0

显示 RAM 共 64 行, 共分 8 页, 每页 8 行。

#### ●列地址(Y ADDRESS)设置指令格式:

R/W	D/I	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0

该指令可用于设置页地址和列地址, 以唯一确定显示 RAM 中的一个单元, 以便使单片机可以通过读、写指令读出该单元中的内容或向该单元写进一个字节数据。

#### ●读状态指令格式:

R/W	D/I	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
1	0	BUSY	0	ON/OFF	REST	0	0	0	0

该指令可用来查询液晶驱动器的状态, 各参量含义如下:

BUSY: 1—内部在工作 0—正常状态

ON/OFF: 1—显示关闭 0—显示打开

REST: 1—复位状态 0—正常状态

由于在 BUSY 和 REST 状态时, 除读状态指令外, 其它指令均不产生作用。因此, 在对液晶驱动器操作之前要查询 BUSY 状态, 以确定是否可以对液晶驱动器进行操作。

#### ●写数据指令格式:

R/W	D/I	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	1	写数据							

#### ●读数据指令格式:

R/W	D/I	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
1	1	读显示数据							

## 4 应用举例

FM12864I 与单片机 8031 的间接控制方式接口电路如图 4 所示。该间接控制方式的驱动程序如下:

```
#include "d:\c51\inc\stdlib.h"
#include "d:\c51\inc\reg51.h"
#define byte unsigned char
data byte column; //定义“列”变量
data byte read_dat, page;
char code hanzi[4][32] =
{ 0x82, 0x8a, 0x92, 0xb2, 0x02, 0xa7, 0x92, 0x5e,
  0x2a, 0xaf, 0x2a, 0x5a, 0x4a, 0x83, 0x82, 0x00,
  0x00, 0x80, 0x40, 0x3f, 0x41, 0x90, 0x95, 0x95,
  0x95, 0xff, 0x95, 0x95, 0x95, 0xd0, 0x40, 0x00, //蓬
```

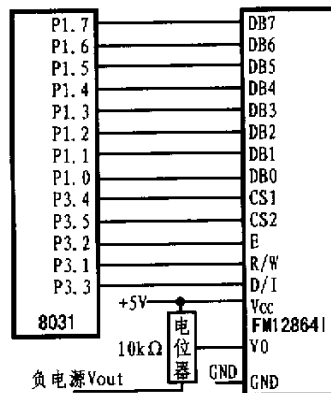


图4 FM12864I 与单片机 8031 的间接控制方式接口电路

```

0x80, 0x64, 0x24, 0x24, 0x3f, 0x24, 0xa6, 0x64,
0x20, 0x20, 0xff, 0x20, 0x20, 0xf0, 0x20, 0x00,
0x00, 0x08, 0x49, 0x89, 0x7d, 0x0b, 0x89, 0x40,
0x30, 0x0e, 0x41, 0x80, 0x40, 0x3f, 0x00, 0x00, //勃
0x40, 0x42, 0x44, 0xcc, 0x00, 0x20, 0x22, 0xe2,
0x22, 0x22, 0xe2, 0x23, 0x32, 0x20, 0x00, 0x00,
0x00, 0x40, 0x20, 0x1f, 0x20, 0x50, 0x4c, 0x43,
0x40, 0x40, 0x47, 0x48, 0x48, 0x6e, 0x20, 0x00, //远
0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0xa0, 0x7f,
0xa0, 0x20, 0x20, 0x20, 0x20, 0x30, 0x20, 0x00,
0x00, 0x40, 0x40, 0x20, 0x10, 0x0c, 0x03, 0x00,
0x01, 0x06, 0x08, 0x10, 0x20, 0x60, 0x20, 0x00 } //大

sbit d_i = 0xb3;
sbit r_w = 0xb1;
sbit e = 0xb2;
sbit cs1 = 0xb4;
sbit cs2 = 0xb5;
sbit busy = read_dat^7;
sbit page_4 = page^4;
void inquire_busy_left();
void send_dat_left(byte d);
void inquire_busy_right();
void send_dat_right(byte d);
void clear();
void ini();
void send_dat_disp_left(byte d);
void send_dat_disp_right(byte d);
void led_disp(char *p);
void hanzi_disp();

void inquire_busy_left() //左半屏查询是否“忙”
{
    r_w = 1;
    r_w = 1;
    d_i = 0;
    d_i = 0;
    cs1 = 1;
    cs1 = 1;
loop: P1 = 0xff;
    e = 1;
    e = 1;
    read_dat = P1;
    e = 0;
    e = 0;

    if(busy) { goto loop; }
    cs1 = 0;
    cs1 = 0;
    d_i = 1;
    d_i = 1;
    r_w = 1;
    r_w = 1;
}

void send_dat_left(byte d) //送左半屏指令
{
    r_w = 0;
    r_w = 0;
    d_i = 0;
    d_i = 0;
    cs1 = 1;
    cs1 = 1;
    e = 1;
    e = 1;
    P1 = d;
    P1 = d;
    e = 0;
    e = 0;
    cs1 = 0;
    cs1 = 0;
    d_i = 1;
    d_i = 1;
    r_w = 1;
    r_w = 1;
}

void inquire_busy_right() //右半屏查询是否“忙”
{
    r_w = 1;
    r_w = 1;
    d_i = 0;
    d_i = 0;
    cs2 = 1;
    cs2 = 1;
loop2: P1 = 0xff;
    e = 1;
    e = 1;
    read_dat = P1;
    e = 1;
    e = 1;

```

```

    e = 0;
    e = 0;
    if(busy) goto loop2;
    cs2 = 0;
    cs2 = 0;
    d_i = 1;
    d_i = 1;
    r_w = 1;
    r_w = 1;
}

```

```

    P1 = d;
    e = 0;
    e = 0;
    cs1 = 0;
    cs1 = 0;
    d_i = 1;
    d_i = 1;
    r_w = 1;
    r_w = 1;
}

```

void send\_dat\_right(byte d) //送右半屏指令

```

{ r_w = 0;
  r_w = 0;
  d_i = 0;
  d_i = 0;
  cs2 = 1;
  cs2 = 1;
  e = 1;
  e = 1;
  P1 = d;
  P1 = d;
  e = 0;
  e = 0;
  cs2 = 0;
  cs2 = 0;
  d_i = 1;
  d_i = 1;
  r_w = 1;
  r_w = 1;
}

```

void send\_dat\_disp\_right(byte d) //送右半屏显示

```

{ r_w = 0;
  r_w = 0;
  d_i = 1;
  d_i = 1;
  cs2 = 1;
  cs2 = 1;
  e = 1;
  e = 1;
  P1 = d;
  P1 = d;
  e = 0;
  e = 0;
  cs2 = 0;
  cs2 = 0;
  d_i = 1;
  d_i = 1;
  r_w = 1;
  r_w = 1;
}

```

void send\_dat\_disp\_left(byte d) //送左半屏显示字符

```

{ r_w = 0;
  r_w = 0;
  d_i = 1;
  d_i = 1;
  cs1 = 1;
  cs1 = 1;
  e = 1;
  e = 1;
  P1 = d;

```

void delay()

```

{ data byte i,j;
  for(i = 0; i < 255; i++)
  { for(j = 0; j < 255; j++) ; }
}

```

void ini() //初始化子程序

```

{ e = 0; r_w = 1; d_i = 1; cs1 = cs2 = 0; delay();
  inquire_busy_left(); //关显示指令
  send_dat_left(0x3e);
  inquire_busy_right();

```

```

send_dat_right(0x3e);
inquire_busy_left(); //开显示指令
send_dat_left(0x3f);
inquire_busy_right();
send_dat_right(0x3f);
inquire_busy_left(); //起始行设置
send_dat_left(0xc0);
inquire_busy_right();
send_dat_right(0xc0);
}

void clear() //清屏子程序
{ data byte i,j,disp_page;
  for(i=0;i<8;i++)
  { disp_page=i;
    disp_page=disp_page|0xb8; //页地址设置
    inquire_busy_left();
    send_dat_left(disp_page);
    inquire_busy_left();
    send_dat_left(0x40); //列地址设置
    for(j=0;j<64;j++) //清左半屏
    { send_dat_disp_left(0x00); }
  }
  for(i=0;i<8;i++) //清右半屏
  { disp_page=i;
    disp_page=disp_page|0xb8;
    inquire_busy_right();
    send_dat_right(disp_page);
    inquire_busy_right();
    send_dat_right(0x40);
    for(j=0;j<64;j++)
    { send_dat_disp_right(0x00); }
  }
}

void led_disp(char *p)
{ idata byte i;
  idata char ch;
  for(i=0;i<8;i++)
  { ch=*p;
    p++;
    send_dat_disp_left(ch);
  }
}

void hanzi_disp() //汉字显示子程序
{ data byte i;
  clear();
  page=0xbc; //页地址
  column=0x40; //列地址
  inquire_busy_left();
  send_dat_left(page);
  inquire_busy_right();
  send_dat_right(page);
  inquire_busy_left();
  send_dat_left(column);
  for(i=0;i<4;i++) //汉字上半部
  { led_disp(&hanzi[i][0]);
    led_disp(&hanzi[i][8]);
  }
  page=0xbd; //页地址加1
  column=0x40; //列地址
  inquire_busy_left();
  send_dat_left(page);
  inquire_busy_left();
  send_dat_left(column);
  for(i=0;i<4;i++) //汉字下半部
  { led_disp(&hanzi[i][16]);
    led_disp(&hanzi[i][24]);
  }
}

main()
{ data byte i;
  delay(); delay(); delay(); //延时
  IE=0x81; TCON=0x00;
  ini(); //初始化
  clear(); //清屏
  hanzi_disp();
}

```

## 参考文献

1. 李维提, 郭强. 液晶显示应用技术. 电子工业出版社  
 收稿日期: 2001-10-29  
 咨询编号: 020523