

## MSP430程序库<三>12864液晶程序库

液晶是单片机系统最常用的显示设备之一，这个程序库是在 MSP430F169、MSP1430F149单片机上测试通过的，可以放心使用；液晶选用的是金鹏的：OCMJ4X8C型号的12864液晶；控制液晶用的是并行方式，三个控制口是 P3.0、P3.1、P3.2三个 IO 口，数据用的是 P5数据 IO 口。

1.硬件介绍：

### **430的数字 IO 口：**

MSP430F149、MSP430F169均有 P1-P6 每个8位 共48个 IO 口；有大量的 IO 口可供使用，所以对液晶控制可以选用8位平行数据方式；430的每个 IO 口都是双向 IO 口，通过寄存器控制其数据传输方向，很方便实用；有关 msp430单片机的 IO 口介绍可以参考[德州仪器](#)提供的[用户指南](#)和[数据手册](#)等资料。

### **液晶 OCMJ4X8C：**

此模块可以显示字母、数字符号、中文字型及图形，具有绘图及文字画面混合显示功能。提供三种控制接口，分别是8位微处理器接口，4位微处理器接口及串行接口（OCMJ4X16A/B无串行接口）。所有的功能，包含显示 RAM，字型产生器，都包含在一个芯片里面，只要一个最小的微处理系统，就可以方便操作模块。

这款液晶内置2M-位中文字型 ROM (CGROM) 总共提供8192个中文字型(16x16点阵)，16K 位半宽字型 ROM(HCGROM) 总共提供126 个符号字型(16x8点阵)，64x16位字型产生 RAM(CGAM)，另外绘图显示画面提供一个64x256点的绘图区域（GDRAM），可以和文字画面混和显示。

OCMJ4X8C 的引脚说明：

引脚	名称	方向	说明
1	VSS	-	GND(0V)
2	VDD	-	Supply Voltage For Logic(+5V)
3	NC	-	Supply Voltage For LCD(悬空)
4	RS(CS I )		H:Data L:Instruction Code
5	R/W(S I TD)		H:Read L:Write
6	E(SCL I K)		Enable Signal,高电平有效
7	DB0	I/O	数据0
8	DB1	I/O	数据1

9	DB2	I/O	数据2
10	DB3	I/O	数据3
11	DB4	I/O	数据4
12	DB5	I/O	数据5
13	DB6	I/O	数据6
14	DB7	I/O	数据7
15	PSB	I	H:Parallel Mode L:Serial Mode
16	NC	-	空脚
17	/RST	I	Reset Signal,低电平有效
18	NC	-	空脚
19	LEDA	-	背光源正极 (+5V)
20	LEDK	-	背光源负极 (0V)

引脚连接方式：PSB、RST 接高电平（3.3v）；RS 接 P3.0；R/W 接 P3.1；E 接 P3.2；DB0-DB7接 P5口，电源接3.3v（包括背光） GND 接地（包括背光）。

#### 汉字图形显示步骤：

##### 1、显示资料 RAM（DDRAM）

显示数据 RAM 提供64x2 个字节的存储空间，最多可以控制4 行16 字（64 个字）的中文字型显示，当输入显示资料 RAM 时，可以分别显示 CGROM，HCGROM 与 CGRAM 的字型；本系列模块可以显示三种字型，分别是半宽的 HCGROM 字型、CGRAM 字型及中文 CGROM 字型，三种字型的选择，由在 DDRAM 中写入的编码选择，在0000H~0006H 的定字型，02H~7FH 的编码中将选择半编码中将选择 CGRAM 的自宽英数字的字型，至于 A1 以上的编码将自动的结合下一个字节，组成两个字节的编码达成中文字型的编码。

BIG5（A140~D75F） GB(A1A0~F7FF)，详细各种字型编码如下：

- 1) . 显示半宽字型：将8 位资料写入 DDRAM 中，范围为02H~7FH 的编码。
- 2) . 显示 CGRAM 字型：将16 位资料写入 DDRAM 中，总共有0000H, 0002H, 0004H, 0006H 四种编码。
- 3) . 显示中文字形：将16 位资料写入 DDRAM 中，范围为 A140H~D75FH 的编码 (BIG5) ， A1A0H~F7FFH 的编码(GB)。将16 位资料写入 DDRAM 方式为透过连写入两个字节的资料来完成，先写入高字节（D15~D8）再写入低字节（D7~D0）。

##### 2、绘图 RAM（GDRAM）

绘图显示 RAM 提供64x32 个字节的记忆空间(由扩充指令设定绘图 RAM 地址)，最多可以控制256x64点的二维绘图缓冲空间，在更改绘图 RAM 时，由扩充指令设定 GDRAM 地址先设垂直地址再设水平地址(连续写入两个字节的资料来完成垂直与水平的坐标地址)，再

写入两个8位的资料到绘图RAM，而地址计数器(AC)会自动加一，整个写入绘图RAM的步骤如下：

- 1) . 先将垂直的字节坐标(Y)写入绘图RAM地址。
- 2) . 再将水平的字节坐标(X)写入绘图RAM地址。
- 3) . 将D15~D8写入到RAM中(写入第一个Bytes)。
- 4) . 将D7~D0写入到RAM中(写入第二个Bytes)。

液晶屏显示地址：

	X 坐标							
Line1	80H	81H	82H	83H	84H	85H	86H	87H
Line2	90H	91H	92H	93H	94H	95H	96H	97H
Line3	88H	89H	8AH	8BH	8CH	8DH	8EH	8FH
Line4	98H	99H	9AH	9BH	9CH	9DH	9EH	9FH

有关液晶其他的或详细的介绍，请参考12864液晶的资料。

## 2.程序实现：

- 判忙：等待液晶模块空闲。

液晶模块要求：当模块在接受指令前，微处理顺必须先确认模块内部处于非忙碌状态，即读取BF标志时BF需为0，方可接受新的指令；如果在送出一个令前并不检查BF标志，那么在前一个指令和这个指令中间必须延迟一段较长的时间，即是等待前一个指令确实执行完成；在这里，我选用等待忙标志结束。程序如下：

```
void WaitForEnable()
{
    char busy;

    CLR_RS;
    SET_RW;

    DATA_DIR_IN;

do //判忙
{
    SET_EN;
    _NOP();
    busy = DATA_IN;
    CLR_EN;
}
while(busy & 0x80);

    DATA_DIR_OUT;
}
```

这样，每次向液晶写命令或数据时，只需先调用此函数即可，该函数将会阻塞，直到忙标志

变回0（内部空闲，可以接受命令）。

- 写入数据：向模块内部 RAM 写入数据。

写入数据到 DDRAM 即可显示到液晶，写入函数：

```
void LcdWriteData(char data)
{
    WaitForEnable();

    SET_RS;
    CLR_RW;

    DATA_OUT = data;    //写数据

    SET_EN;
    _NOP();
    CLR_EN;
}
```

同样，调用这个函数也可以向其他 RAM 写入数据，完成相应操作。

- 写入命令：向模块写入命令。

写入命令可以通过液晶的指令集，控制液晶完成相应的功能。程序如下：

```
void LcdWriteComm(char cmd)
{
    WaitForEnable();    //检测忙信号?

    CLR_RS;
    CLR_RW;

    DATA_OUT = cmd;    //写命令

    SET_EN;
    _NOP();
    CLR_EN;
}
```

如果 cmd 是 0x80-0x9F，则是向液晶写入地址；地址表参见前面硬件介绍部分。

- 写入字符串：写入字符串，以显示。

写入字符串即是多次调用写入数据，把字符串写入液晶以供显示。程序如下：

```
void LcdWriteString(char addr, char* str)
{
    LcdWriteComm(addr);
    while(*str != '\0')
    {
```

```

        LcdWriteData(*str);
        str++;
    }
}

```

这是向某个地址写入字符串，液晶显示到相应位置。这个函数有个要求，就是字符串是中文字符串；如果不是，每一处的英文必须两个相连，否则将显示乱码，如果只有一个英文字符，可以加入空格；如：**LcdWriteString(0x90,"1 abcd 你好啊");**1+空格+abcd+汉字中 1 只有一个字符，加空格，ab，cd 两个，直接显示到一个汉字的位置。

- 液晶初始化：液晶必须初始化之后才能正常使用。

初始化就是一系列命令，完成液晶状态的初始工作，以使液晶可供正常使用。程序如下：

```

void LcdInit()
{
    CTRL_DIR_OUT;

    DelayNms(500);
    LcdWriteComm(0x30);    //基本指令集
    LcdWriteComm(0x01);    //清屏，地址00H
    LcdWriteComm(0x06);    //光标的移动方向
    LcdWriteComm(0x0c);    //开显示，关光标
}

```

在运行过这个函数之后，液晶方能正常的显示；在调用液晶显示函数前，必须先调用这个函数。

程序实现就先到这儿，还可以加入显示图片等功能；要收拾东西回去了，其他功能暂不实现了，以后需要的时候再加入。

### 3.使用示例：

这个程序的使用方式和串口程序库的使用方式一样，把 C 文件加入工程；H 文件包含进要调用的程序源文件中即可。

```

void main( void)
{
    // Stop watchdog timer to prevent time out reset
    WDTCTL = WDTPW + WDTHOLD;
    ClkInit();
    LcdInit();
    LcdWriteString(0x90,"1 abcd 你好啊");
}

```

这个函数运行后，将在第二行显示 1 abcd 你好啊 字符串，如果把1后面的空格去掉，中文部分将是乱码。ClkInit(); 这个函数和前面一个里面调用的一样，把主系统时钟设为 8MHz，SMCLK 设为 1MHz。 有关详细内容参见程序库，mian.c。

到此，液晶的驱动基本完成，其他功能之后再添加了。

相关文章及附件下载：[http://www.ideyi.org/bbs/article\\_1077\\_369540.html](http://www.ideyi.org/bbs/article_1077_369540.html)

