

# ***EDM16032-01***

## 液晶显示器模块 原理与应用手册

大连东福彩色液晶显示器有限公司

## 一、概述

160(W) X 32(H)点阵图形汉字液晶显示器，控制器 HD61830 (详见 HD61830 说明)。

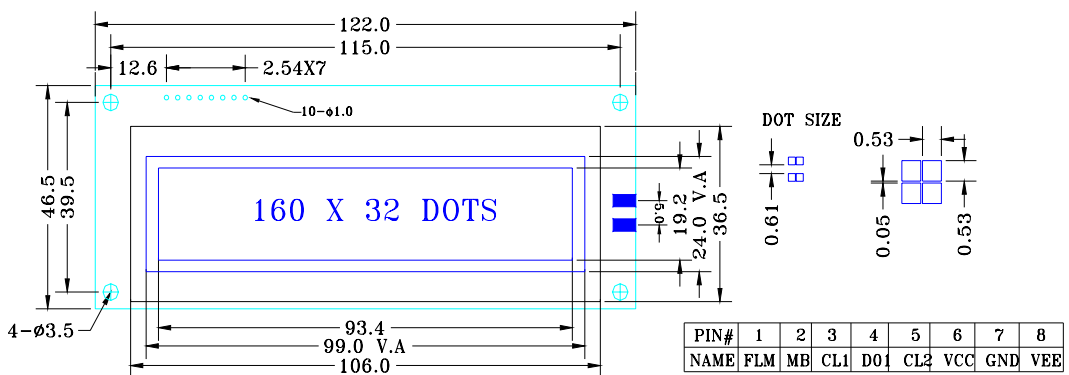
主要技术参数：

模块尺寸：	122W X 46.5H X 10T(最大) MM
有效视区：	99.0W X 24.0H MM
点数：	160W X32H 点
点的尺寸：	0.53W X 0.53H MM
点距：	0.58W X 0.58H MM
逻辑电路电源电压( $V_{DD} - V_{SS}$ ):	0 6.0V
LCD电源电压 ( $V_{DD} - V_{EE}$ ):	0 6.0V
输入电压 ( $V_I$ ):	$V_{SS} \quad V_{DD}$
工作温度 ( $T_A$ ):	0 50°C
存储温度 ( $T_{STG}$ ):	-20 60°C
输入高电平 ( $V_{IH}$ ):	最小 $0.8 \times V_{DD} V$
输入低电平 ( $V_{IL}$ ):	最大 $0.2 \times V_{DD} V$
最大时钟频率 ( $F_{CL2}$ ):	最小 3.3MHz(占空比=1/2)
电源电流:	( $I_{DD}$ )=5 MA ( $I_{EE}$ )=3 MA
显示占空比:	1/32

## 二、组件的外部接口

管脚号	管脚名称	功能描述	管脚号	管脚名称	功能描述
1	FLM	每个行显示周期的起始信号	5	CL2	LCD 驱动器显示数据的移位时钟
2	MB	将液晶驱动信号转移为交流的信号.B 型	6	VCC	电源电压+5V
3	CL1	LCD 驱动器显示数据的锁存信号.	7	GND	电源地 0V
4	DO1	串行显示数据	8	VEE	LCD 驱动电压 0V--+5V

EDM-16032 160×32 点阵汉字液晶显示器组件尺寸图：



为了显示内容必须外加控制器,现在以 HD61830 为例进行说明. 为说明的方便,现将 HD61830 的指令及控制时序加以说明.

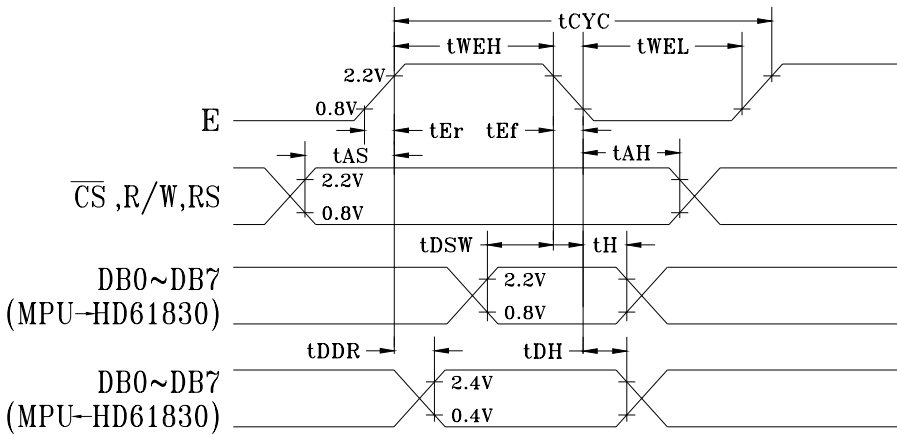


HD61830 有五种类型寄存器:指令寄存器 IR,数据输入寄存器 DIR,数据输出寄存器 DOR,点阵寄存器 DR,方式控制器 MCR,忙标志 BF。

HD61830 与 CPU 接口信号

管脚名称	功能描述
CS	片选:CS=0 有效
R/W	读/写 R/W=1:CPU←HD61830R/W=0:CPU→HD61830
RS	寄存器选择:RS=1 指令寄存器 RS=0 数据寄存器
RES	复位:RES=0,导致关显示;从机方式和 HP=6
E	使能信号:数据在 E 的下降沿写入,在 E=1 时数据可以读出
DB0-DB7	数据总线: 三态 I/O 口线
VCC	电源电压+5V
GND	电源地 0V

HD61830 与 CPU 接口时序



参数说明

参数	符号	最小值	典型值	最大值	单位
使能信号周期	T <sub>CYC</sub>	1.0	----	----	μS
使能信号脉冲宽度	高电平	T <sub>WEH</sub>	0.45	----	μS
	低电平	T <sub>WEL</sub>	0.45	----	μS
使能信号上升时间	T <sub>ER</sub>	----	----	25	nS
使能信号下降时间	T <sub>EF</sub>	----	----	25	nS
调整时间	T <sub>AS</sub>	140	----	----	nS
数据调整时间	T <sub>DSW</sub>	225	----	----	nS
数据延迟时间	T <sub>DDR</sub>	----	----	225	nS
数据保持时间	T <sub>H</sub>	10	----	----	nS
地址保持时间	T <sub>AH</sub>	10	----	----	nS
数据保持时间	T <sub>DH</sub>	20	----	----	nS

显示控制指令

显示是由写数据到指令寄存器来实现的,由 RS 信号来区分指令寄存器。

1.方式控制

寄存器	R/W	RS	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
指令寄存器	0	1	0	0	0	0	0	0	0	0
方式控制寄存器	0	0	0	0	方式数据					

DB5	DB4	DB3	DB2	DB1	DB0	光标/闪烁	字符发生器	图形/字符显示
1/0	1/0	0	0	0	0	光标关显示	内部字符 发生器	字符方式
		0	1			光标开显示		
		1	0			光标关显示字符闪烁		
		1	1			光标闪烁		
		0	0		1	光标关显示	外部字符 发生器	
		0	1			光标开显示		
		1	0			光标关显示字符闪烁		
		1	1			光标闪烁		
					0	0	1	
显示 开/关 *1	主/从 *2	闪烁	光标	图形/ 字符 方式	外部/ 内部字符 发生器			

\*1 1 开显示; 0 关显示

\*2 1 主机方式; 0 从机方式

## 2 设置字符点阵

寄存器	R/W	RS	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
指令寄存器	0	1	0	0	0	0	0	0	0	1
字符点阵	0	0	(VP-1)				0	(HP-1)		

VP 仅在字符方式下有效,表示每个字符的垂直点阵数.在字符方式下,HP 表示每个字符的水平点阵数.在图形方式下,HP 表示每个字符的显示的位数.

HP	DB2	DB1	DB0	水平字符点阵
6	1	0	1	6
7	1	1	0	7
8	1	1	1	8

## 3 设置字符个数

寄存器	R/W	RS	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
指令寄存器	0	1	0	0	0	0	0	0	1	0
字符个数寄存器	0	0	0	(HN-1)						

在字符方式下,HN 表示水平字符个数.在图形方式下,HN 表示水平字节的个数.HN 可设置为从 0 到 128 的偶数.

## 4 设置分时值(显示占空比的倒数)

寄存器	R/W	RS	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
指令寄存器	0	1	0	0	0	0	0	0	1	1
分时值寄存器	0	0	0	(NX-1)						

在多路显示中,NX 表示分时值,1/NX 表示显示占空比. NX 可设置为 1 到 128

## 5 设置光标位置

寄存器	R/W	RS	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
指令寄存器	0	1	0	0	0	0	0	1	0	0
光标位置寄存器	0	0	0	0	0	0	(CP-1)			

在字符方式下,CP表示光标在一个字符下的显示位置,例如对一个5X7点阵指定,CP=8则光标刚好显示在字符下,其长度为字符水平点阵HP。HP可取1到16。

#### 6 设置低位显示起始地址

寄存器	R/W	RS	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
指令寄存器	0	1	0	0	0	0	1	0	0	0
显示起始	0	0	(起始低位地址)							

显示起始地址表示一个RAM地址,在此地址处的数据将显示在屏幕的左上角.在图形方式下,起始地址包括高/低16位.在字符方式下起始地址包括高位地址的低4位(DB3-DB0),和低位地址的8位,高位地址高4位(DB7-DB4)忽略。

#### 设置高位显示起始地址

寄存器	R/W	RS	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
指令寄存器	0	1	0	0	0	0	1	0	0	1
显示起始	0	0	(起始高位地址)							

#### 7 设置光标低位地址(RAM写入的低位地址)

寄存器	R/W	RS	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
指令寄存器	0	1	0	0	0	0	1	0	1	0
光标起始	0	0	(光标低位地址)							

#### 设置光标高位地址(RAM写入的高位地址)

寄存器	R/W	RS	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
指令寄存器	0	1	0	0	0	0	1	0	1	1
显示起始地址寄存器	0	0	(光标高位地址)							

光标地址表示一个RAM地址,显示数据和字符码可以传送到该地址.也可以从该地址取走.即由光标地址指定的该地址可以读/写.在字符方式下,光标出现在光标地址指定的字符下,光标地址计数器为一个16位加1计数器,具有设置和复位功能。

#### 光标地址设置条件

- 1 重新设定低位和高位地址时,先设定低位地址在设定高位地址
- 2 当仅重新设定低位地址时,不要忘记重新设定一次高位地址
- 3 当仅重新设定高位地址时,不必再重新设定一次低位地址

#### 8 写显示数据

寄存器	R/W	RS	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
指令寄存器	0	1	0	0	0	0	1	1	0	0
RAM	0	0	图形数据/字符码							

RS=1 将码"0C"写入指令寄存器,RS=0 8位要显示的数据或字符码应写入数据寄存器.此数据被传送到由光标地址指定的RAM,此操作完成后,光标地址自动加1。

#### 9 读显示数据

寄存器	R/W	RS	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
指令寄存器	0	1	0	0	0	0	1	1	0	1
RAM	1	0	图形数据/字符码							

此指令输出数据寄存器的内容到数据总线上,然后将光标地址指定的RAM地址处的数

据传送到数据输出寄存器.此操作完成后.光标地址自动加 1.因此在光标地址设置完成后,正确的数据不能在第一次读后输出,第二次读后才能输出.所以光标地址设置完成后,必须做一次空读,再读一次才能读到正确的数据.

### 10 清位

寄存器	R/W	RS	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
指令寄存器	0	1	0	0	0	0	1	1	1	0
清位寄存器	0	0	0	0	0	0	0	(NB-1)		

### 11 置位

寄存器	R/W	RS	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
指令寄存器	0	1	0	0	0	0	1	1	1	1
置位寄存器	0	0	0	0	0	0	0	(NB-1)		

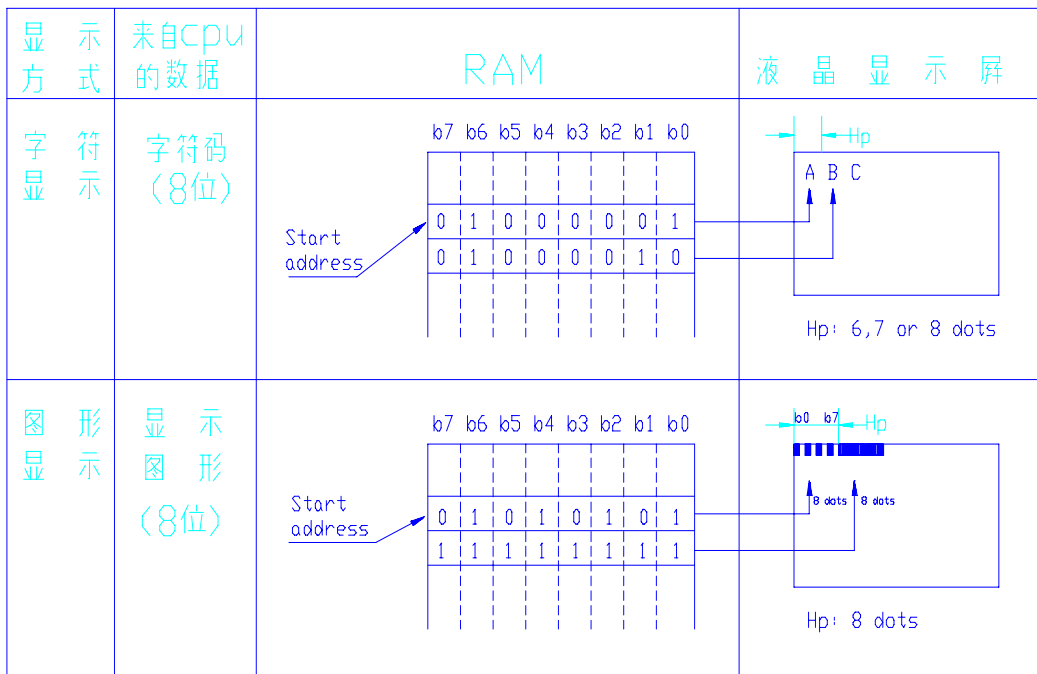
清/置位指令将 RAM 中的显示数据的一个字节的某位清零或置位.NB 指定了在一个字节中该位的位置.NB=1 为最低位,NB=8 为最高位.RAM 地址由光标地址指定此操作完成后,光标地址自动加 1.

### 12 读忙标志

寄存器	R/W	RS	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
忙标志	1	1	1/0							

当 RS=1 R/W=1 则忙标志输出到数据线 DB7 上,在其余任何指令执行期间,忙标志被设置成 1,执行完成后被设置成 0.下一条指令才可以被接受.忙标志为 1,任何指令不被接受.因此在执行任何一条指令或写数据时,必须进行忙标志检查.确保忙标志为 0,才能进行指令操作或写数据.

显示方式:

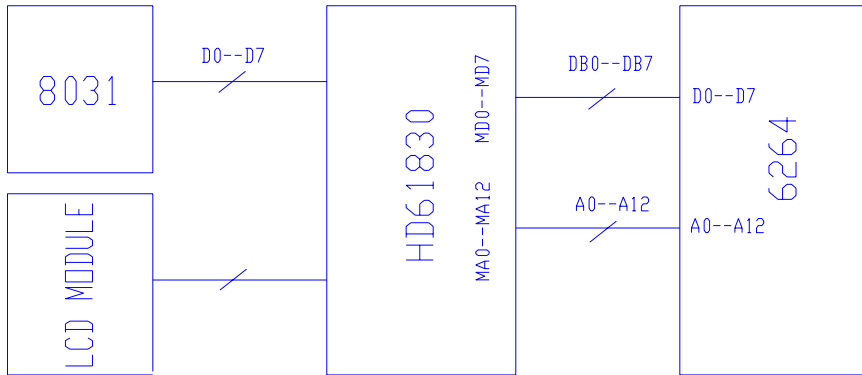




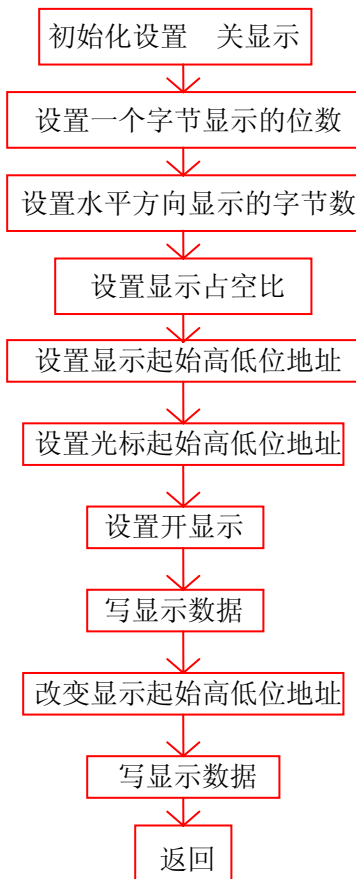


与 8031 单片机接口举例

与 8031 单片机接口硬件电路简略框图如下所示。为方便用户,现将程序清单附上。



程序框图



注: 每次写指令或写数据前必须检查忙标志,忙标志为零才能进行其它操作。

附录:程序清单

```
OEG 0000H
START:MOV A,#00H
      MOV P1,A
      MOV P1,#08H
      LCALL BF
      CLR P1.1
      SETB P1.2
      MOV A,#00H      ;SET DISPLAY OFF
      MOVX @RO,A
      LCALL BF
      CLR P1.1
      CLR P1.2
      MOV A,#012H
      MOVX @RO,A
      LCALL BF      ;BUSY CHECK
      CLR P1.1
      SETB P1.2
      MOV A,01H      ;NUMBER OF BITS OF 1-BYTE DISPLAY DATA
      MOVX @RO,A
      LCALL BF
      CLR P1.1
      CLR P1.2
      MOV A,#07H
      MOVX @RO,A
      LCALL BF
      CLR P1.1
      SETB P1.2
      MOV A,#02H      ;NUMBER OF HORIZONTAL BYTES
      MOVX @RO,A
      LCALL BF
      CLR P1.1
      CLR P1.2
      MOV A,#0FH
      MOVX @RO,A
      LCALL BF
      CLR P1.1
      SETB P1.2
      MOV A,#03H      ;SET MUNBER OF TIME DIVISION
      MOVX @RO,A
      LCALL BF
      CLR P1.1
      CLR P1.2
      MOV A,#01FH
      MOVX @RO,A
```

```

LCALL BF
CLR P1.1
SETB P1.2
MOV A,#08H      ;SET DISPLAY START LOW ORDER ADDRESS
MOVX @RO,A
LCALL BF
CLR P1.1
CLR P1.2
MOV A,#00H
MOVX @RO,A
LCALL BF
CLR P1.1
SETB P1.2
MOV A,#09H      ;SET DISPLAY START HIGH ORDER ADDRESS
MOVX @RO,A
LCALL BF
CLR P1.1
CLR P1.2
MOV A,#00H
MOVX @RO,A
LCALL BF
CLR P1.1
SETB P1.2
MOV A,#0AH      ;SET CURSOR LOW ORDER ADDRESS
MOVX @FO,A
LCALL BF
CLR P1.1
CLR P1.2
MOV A,#00H
MOVX @RO,A
LCALL BF
CLR P1.1      ;SET DISPALY ON
SETB P1.2
MOV A,#00H
MOVX @RO,A
LCALL BF
CLR P1.1
CLR P1.2
MOV A,#32H
MOVX @RO,A
MOV R1,#55H
MOV R3,#20H
L5: MOV R2,#10H
L6: LCALL BF
CLR P1.1

```

```

SETB P1.2
MOV A,#0CH
MOVX @RO,A
LCALL BF
CLR P1.1
CLR P1.2
MOV A,R1
MOVX @RO,A
DJNZ R2,L6
LCALL DELAY1
MOV A,R1
CPL A
MOV R1,A
DJNZ R3,L5
LCALL DELAY3
LCALL BF
CLR P1.1
SETB P1.2
MOV A,#08H      ;SET DISPLAY START LOW ORDER  ADDRESS
MOVX @RO,A
LCALL BF
CLR P1.1
CLR P1.2
MOV A,#00H
MOVX @RO,A
LCALL BF
CLR P1.1
SETB P1.2
MOV A,#09H      ;SET DISPLAY START HIGH ORDER ADDRESS
MOVX @RO,A
LCALL BF
CLR P1.1
CLR P1.2
MOV A,#02H
MOVX @RO,A
MOV R1,#0AAH
MOV R3,#20H
L2: MOV R2,#10H
L1: LCALL BF
CLR P1.1
SETB P1.2
MOV A,#0CH
MOVX @RO,A
LCALL BF
CLR P1.1

```

```
CLR P1.2
MOV A,R1
MOVX @R0,A
DJNZ R2,L1
LCALL DELAY1
MOV A,R1
CPL A
MOV R1,A
CPL A
MOV R1,A
DJNZ R3,L2
LCALL DELAY3
LJMP START
BF: SETB P1.2
    SETB P1.1
    MOVX A,@RO
    JB ACC 7,BF
    RET
DELAY1: MOV R7,#0FFH
DELAY2: DJNZ R7,DELAY2
        RET
DELAY3: MOV R6,#0AH
        MOV R5,#0FFH
        MOV R4,#0FFH
LOOP1: DJNZ R4,LOOP1
        DJNZ R5,LOOP1
        DJNZ R6,LOOP1
        RET
END
```