

# LCD1602 使用说明书

## 1. 概述

LCD1602 (Liquid Crystal Display) 是一种工业字符型液晶，能够同时显示 16×02 即 32 字符(16 列两行)



图 1 LCD1602 液晶模块实物图

## 2. 模块尺寸

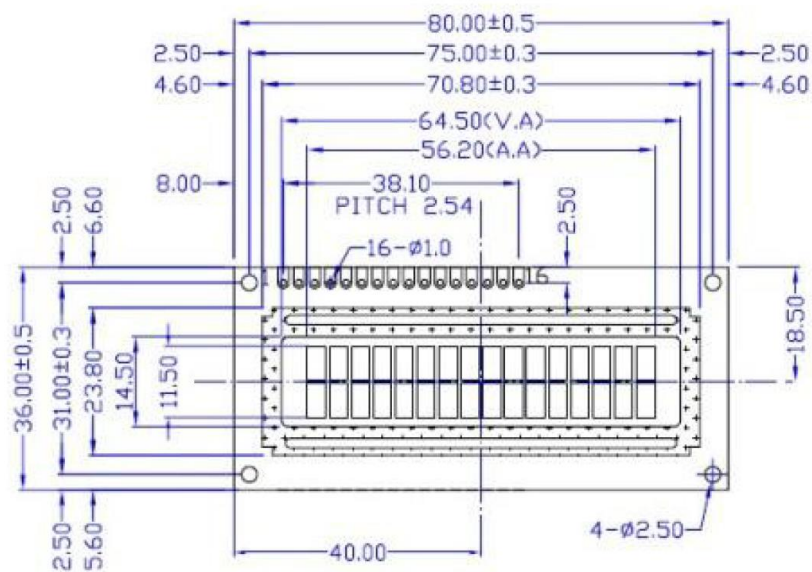


图 2 LCD1602 模块尺寸

3. 引脚接口说明表

表 1 LCD1602 引脚接口说明

编号	符号	引脚说明	编号	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

第 1 脚: VSS 为电源地

第 2 脚: VDD 接 5V 正电源

第 3 脚: VL 为液晶显示器对比度调整端,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”,使用时可以通过一个 10K 的电位器调整对比度。

第 4 脚: RS 为寄存器选择, 高电平时选择数据寄存器、低电平时选择指令寄存器。

第 5 脚: 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者显示地址, 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。

第 6 脚: E 端为使能端, 当 E 端由高电平跳变成低电平时, 液晶模块执行命令。

第 7~14 脚: D0~D7 为 8 位双向数据线。

第 15 脚: 背光源正极。

第 16 脚: 背光源负极。

4. LCD1602 的指令说明

LCD1602 液晶模块内部的控制器共有 11 条控制指令, 如表 2 所示:

**表 2** LCD1602 模块控制指令

序号	指令	RS	R/W	D7	D6	D5	D4	D3	D2	D1	D0
1	清显示	0	0	0	0	0	0	0	0	0	1
2	光标返回	0	0	0	0	0	0	0	0	1	*
3	置输入模式	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	0	0	0	0	0	1	S/C	R/L	*	*
6	置功能	0	0	0	0	1	DL	N	F	*	*
7	置字符发生存贮器地址	0	0	0	1	字符发生存贮器地址(自定义字符)					
8	置数据存贮器地址	0	0	1	显示数据存贮器地址(在哪里显示)						
9	读忙标志或地址	0	1	BF	计算器地址						
10	写数到 CGRAM 或 DDRAM	1	0	要写的 数据内容(显示什么)							
11	从 CGRAM 或 DDRAM 读数	1	1	读出的 数据内容							

1602 液晶模块的读写操作，屏幕和光标的操作都是通过指令编程来实现的。（说明：1 为高电平、0 为低电平）

**指令 1：**清显示，指令码 01H，光标复位到地址 00H 位置。

**指令 2：**光标复位，光标返回到地址 00H。

**指令 3：**光标和显示模式设置。I/D：光标移动方向，高电平右移，低电平左移。实际上就是控制从左到右写入还是从右至左的写入顺序。S：屏幕上所有文字是否左移或者右移。高电平表示有效，低电平则无效。S=1 当写一个字符，整屏显示左移(ID=1)或者右移(I/D=0),以得到光标不移动而屏幕移动的效果。S=0 当写一个字符，整屏显示不移动。

**指令 4：**显示开关控制。**D**：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**：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**：控制光标是否闪烁，高电平闪烁，低电平不闪烁。

**指令 5：**光标或显示移位 S/C：高电平时移动显示的文字，低电平时移动光标。R/L：文字或者光标移动方向，R 表示右移，L 表示左移。

**指令 6：**功能设置命令 DL：高电平时为 8 位总线，低电平时为 4 位总线。N：低电平时为单

行显示，高电平时双行显示。**F**：低电平时显示 5×8 的点阵字符，高电平时显示 5×10 的点阵字符。

**指令 7：** 字符发生器 RAM 地址设置。

**指令 8：** DDRAM 地址设置。

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

**指令 10：** 写数据。

**指令 11：** 读数据。

5. 基本操作时序表

读操作时序

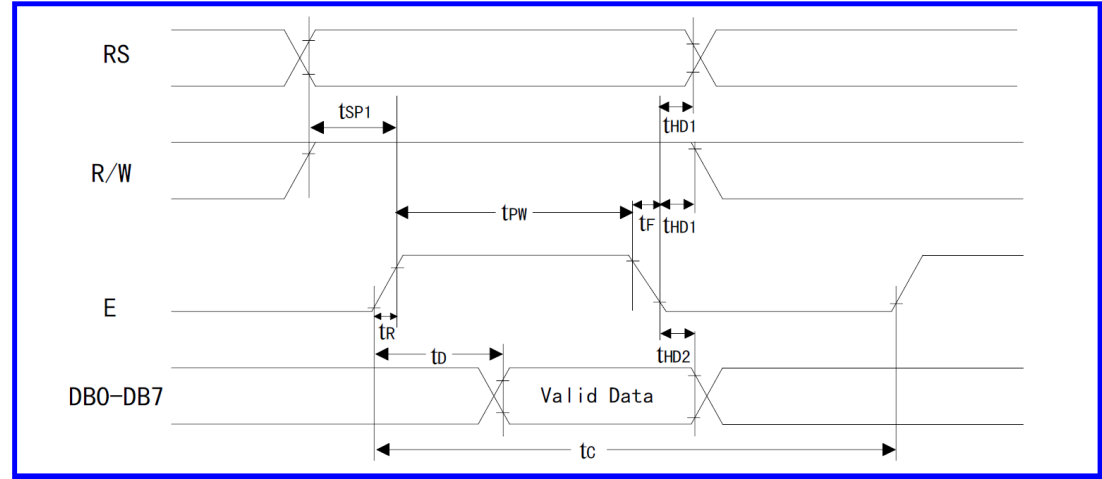


图 3 LCD1602 读操作时序图

写操作时序

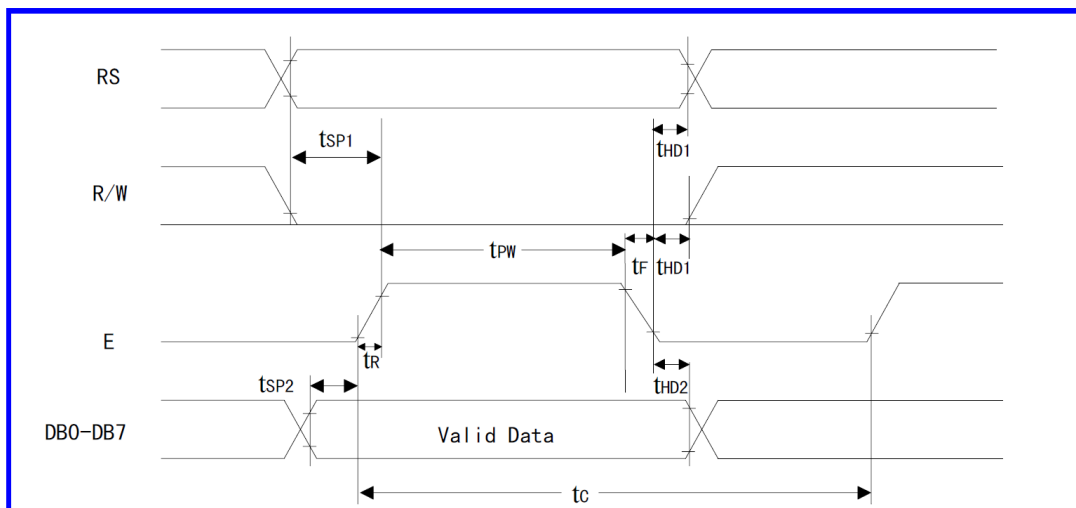


图 4 LCD1602 写操作时序图

## 时序参数

时序参数	符号	极限值			单位	测试条件
		最小值	典型值	最大值		
E 信号周期	$t_c$	400	—	—	ns	引脚 E
E 脉冲宽度	$t_{PW}$	150	—	—	ns	
E 上升沿/下降沿时间	$t_R, t_F$	—	—	25	ns	
地址建立时间	$t_{SP1}$	30	—	—	ns	引脚 E、RS、R/W
地址保持时间	$t_{HD1}$	10	—	—	ns	
数据建立时间(读操作)	$t_D$	—	—	100	ns	引脚 DB0~DB7
数据保持时间(读操作)	$t_{HD2}$	20	—	—	ns	
数据建立时间(写操作)	$t_{SP2}$	40	—	—	ns	
数据保持时间(写操作)	$t_{HD2}$	10	—	—	ns	

液晶显示模块是一个慢显示器件，所以在执行每条指令之前一定要确认模块的忙标志为低电平，表示不忙，否则此指令失效。要显示字符时要先输入显示字符地址，也就是告诉模块在哪里显示字符。

6. LCD1602 内部显示地址

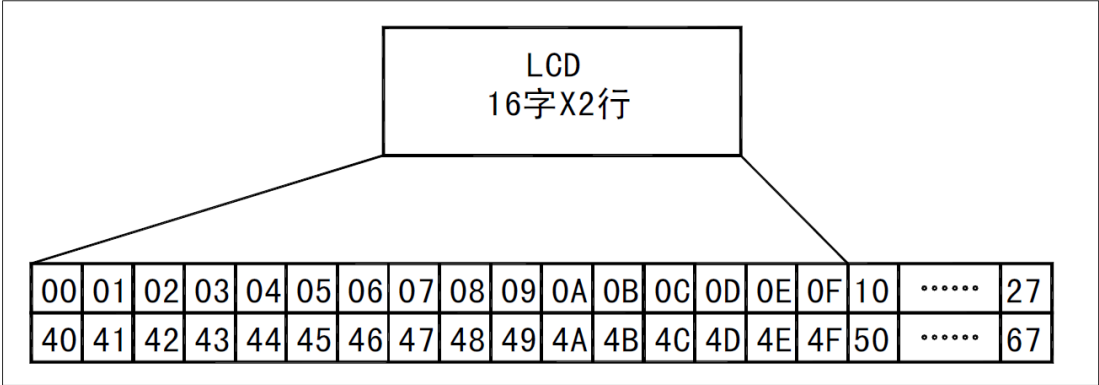


图 5 LCD1602 内部显示地址

例如第二行第一个字符的地址是 40H,那么是否直接写入 40H 就可以将光标定位在第二行第一个字符的位置呢？这样不行，因为写入显示地址时要求最高位 D7 恒定为高电平 1 所以实际写入的数据应该是 01000000B(40H) +10000000B(80H)=11000000B(C0H)。

7. 字符发生存储器

1602 液晶模块内部的字符发生存储器(CGROM)已经存储了 160 个不同的点阵字符图形，如表 3 所示，这些字符有：阿拉伯数字，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，比如大写的英文字符“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”。

表格 3 LCD1602 模块字库表

b7- b3 -b0	b4	0000	0010	0011	0100	0101	0110	0111	1010	1011	1100	1101	1110	1111
0000	CG RAM (1)		0	a	P	`	F		一	夕	ε		α	p
	(2)	!	1	A	Q	a	q	■	ア	チ	△		ä	q
0010	(3)	"	2	B	R	b	r	「	イ	ウ	×		ß	θ
	(4)	#	3	C	S	c	s	」	ウ	テ	ε		ε	∞
0100	(5)	\$	4	D	T	d	t	、	エ	ト	ト		μ	ω
	(6)	%	5	E	U	e	u	=	オ	ナ	ユ		σ	Ω
0110	(7)	&	6	F	V	f	v	ヲ	カ	ニ	ヨ		ρ	Σ
	CG RAM (8)	'	7	G	W	g	w	フ	キ	ヌ	ウ		g	π
1000	CG RAM (1)	(	8	H	X	h	x	ィ	ク	ホ	リ		フ	α
	(2)	)	9	I	Y	i	y	ウ	ケ	ル	ル		フ	γ
1010	(3)	*	:	J	Z	j	z	エ	コ	ハ	レ		j	〒
	(4)	+	:	K	[	k	[	オ	サ	ヒ	ロ		×	π
1100	(5)	,	<	L	¥	l	l	ト	シ	フ	ワ		φ	π
	(6)	-	=	M	]	m	]	ユ	ズ	ハ	ン		ト	÷
1110	(7)	.	>	N	^	n	→	ヨ	セ	ホ	ッ		ñ	
	CG RAM (8)	/	?	O	_	o	+	ッ	ソ	マ	°		ö	■

8. 注意事项

- (1) 在对液晶模块的初始化中要先设置其显示模式
- (2) 在液晶模块显示字符时光标是自动右移的，无需人工干预
- (3) 每次输入指令前都要判断液晶模块是否处于忙的状态

## 9. LCD1602 的一般初始化(复位)过程

### 9.1 LCD1602 初始化过程(8bit)

- (1) 延时 15ms
- (2) 写指令 38H(不检测忙信号)
- (3) 延时 5ms
- (4) 以后每次写指令，读/写数据操作均需要检测忙信号
- (5) 写指令 38H：显示模式设置
- (6) 写指令 08H：显示关闭
- (7) 写指令 01H：显示清屏
- (8) 写指令 06H：显示光标移动设置
- (9) 写指令 0CH：显示开及光标设置

### 9.2 LCD1602 初始化过程(4bit)

- (01) 延时 50ms
- (02) 发送 0x03(4bit)(rs=0,rw=0)
- (03) 延时 4.5ms
- (04) 发送 0x03(4bit)(rs=0,rw=0)
- (05) 延时 4.5ms
- (06) 发送 0x03(4bit)(rs=0,rw=0)
- (07) 延时 150 $\mu$ s
- (08) 发送 0x02(4bit)(rs=0,rw=0)
- (09) 写指令 28H(8bit)
- (10) 写指令 0CH(8bit)
- (11) 写指令 01H(8bit)
- (12) 延时 2ms(8bit)
- (13) 写指令 06H(8bit)



