

WaveShare LCD2004 系列

售前咨询：wvashare_1@163.com

售后服务：wvashare_2@163.com

公司地址：深圳市福田区华强北振华路鼎诚国际大厦621

研发分部：深圳市南山区深职院电子应用中心

■ 特性：

显示内容：20 字符 x4 行

字符点阵：5 x 8 点

驱动方式：1/16D

可供型号：TN，STN(黄绿模、灰模、黑白模)
反射型

■ 电参数 ($V_{DD}=5.0V \pm 10\%$, $V_{SS}=0V$, $T_a=25^\circ C$)

名称	符号	测试条件	标准值			单位
			最小	典型	最大	
电源电压	$V_{DD}-V_{SS}$	-	4.5	5.0	5.5	V
输入高电平	V_{IH}	-	2.2	-	V_{DD}	V
输入低电平	V_{IL}	-	-0.3	-	0.6	V
输入高电压	V_{OH}	$-I_{OH}=0.2mA$	2.4	-	-	V
输入低电压	V_{OL}	$I_{OL}=1.2mA$	-	-	0.4	V
工作电流	I_{DD}	$V_{DD}=5.0V$	-	2.0	5.0	mA

■ 电路图

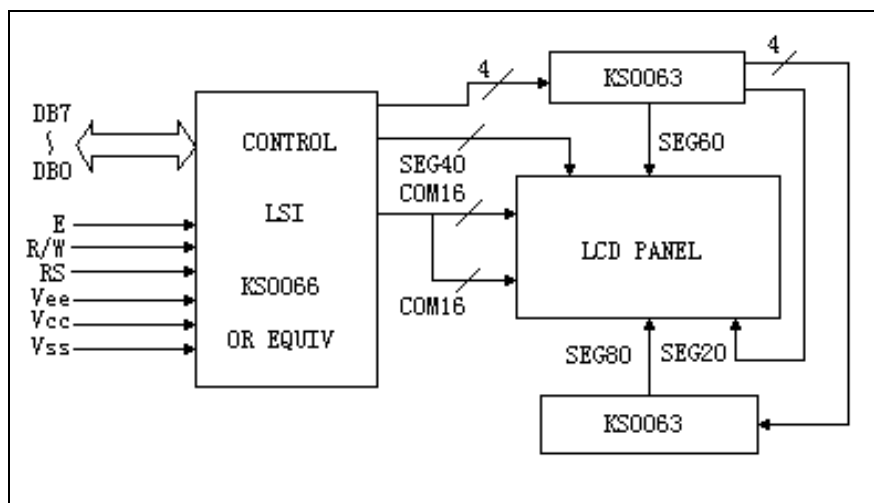
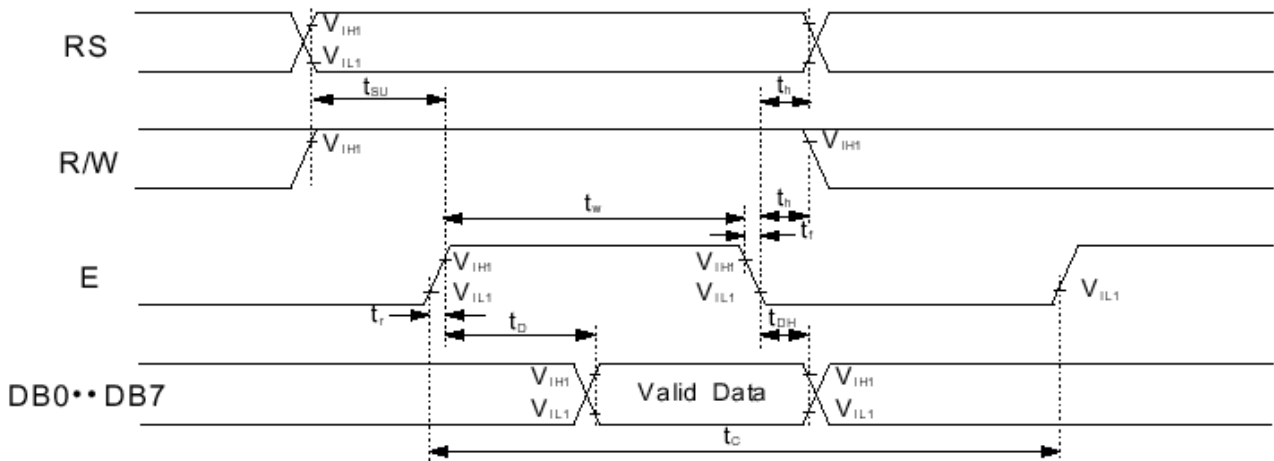
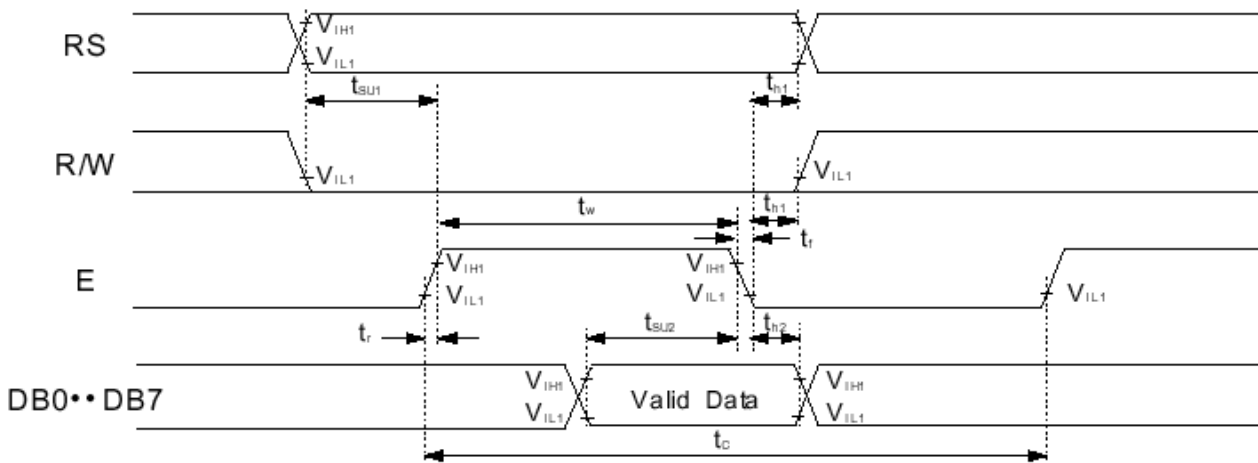


Table 13. AC Characteristics ($V_{DD} = 2.7V \sim 4.5V$, $T_a = -30 \sim +85^{\circ}C$)

Mode	Characteristic	Symbol	Min.	Typ.	Max.	Unit
Write Mode (Refer to Fig-6)	E Cycle Time	t_c	1000	-	-	ns
	E Rise / Fall Time	t_{RtF}	-	-	25	
	E Pulse Width (High, Low)	t_w	450	-	-	
	R/W and RS Setup Time	t_{su1}	60	-	-	
	R/W and RS Hold Time	t_{H1}	20	-	-	
	Data Setup Time	t_{su2}	195	-	-	
	Data Hold Time	t_{H2}	10	-	-	
Read Mode (Refer to Fig-7)	E Cycle Time	t_c	1000	-	-	ns
	E Rise / Fall Time	$t_{R,tF}$	-	-	25	
	E Pulse Width (High, Low)	t_w	450	-	-	
	R/W and RS Setup Time	t_{su}	60	-	-	
	R/W and RS Hold Time	t_H	20	-	-	
	Data Output Delay Time	t_D	-	-	360	
	Data Hold Time	t_{DH}	5	-	-	



Read Mode Timing Diagram

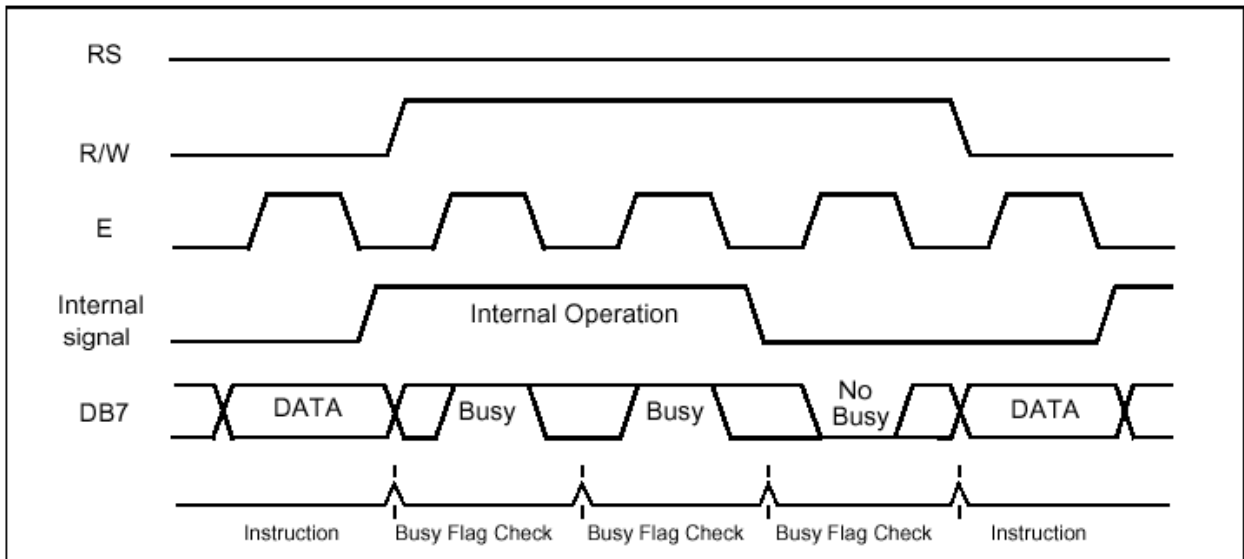


Write Mode Timing Diagram

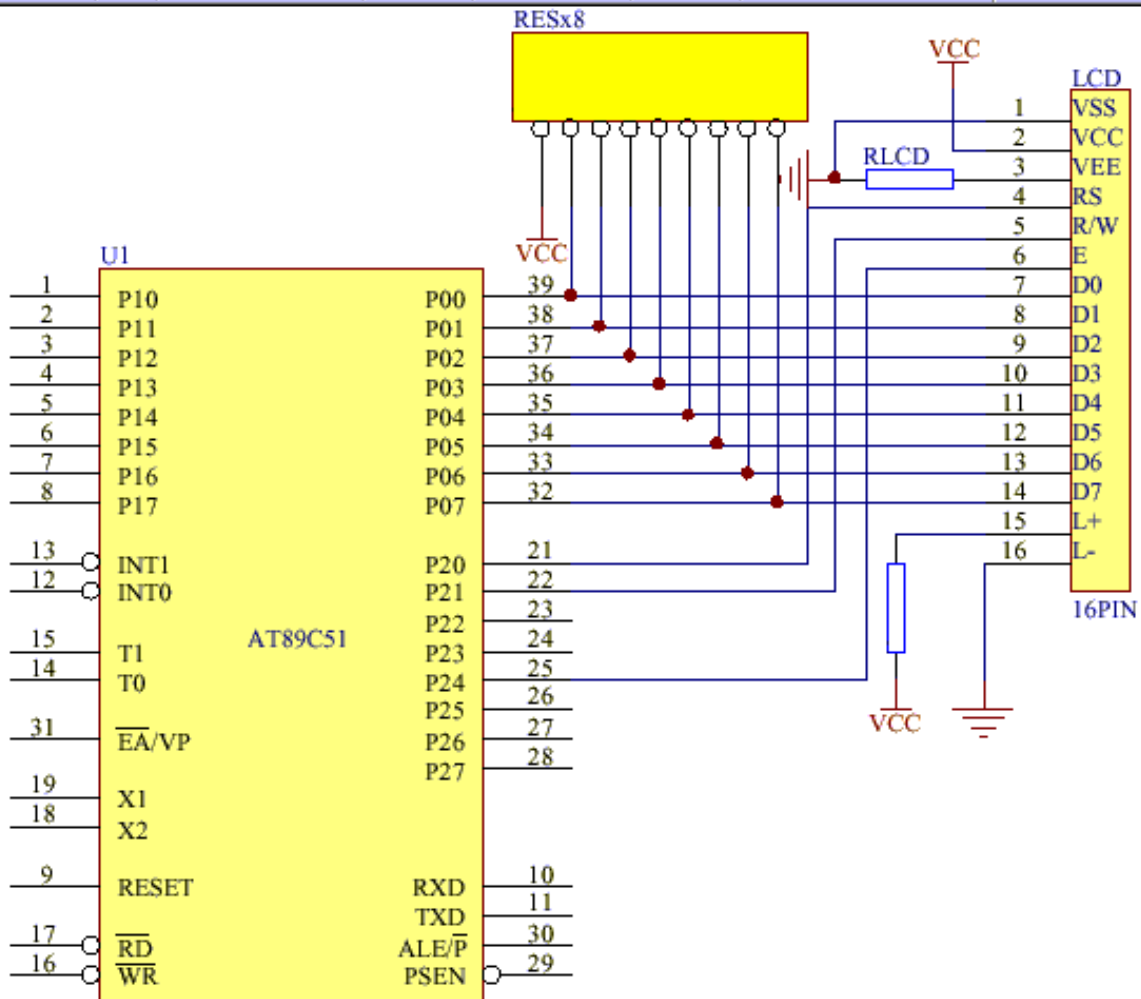
■使用时序

1) Interface with 8-bit MPU

When interfacing data length are 8-bit, transfer is performed at a time through 8 ports, from DB0 to DB7. Example of timing sequence is shown below.



■使用连接



CGROM
Table 5. Relationship between Character Code (DDRAM) and Character Pattern (CGRAM)

Character Code (DDRAM data)								CGRAM Address						CGRAM Data								Pattern number
D7	D6	D5	D4	D3	D2	D1	D0	A5	A4	A3	A2	A1	A0	P7	P6	P5	P4	P3	P2	P1	P0	
0	0	0	0	x	0	0	0	0	0	0	0	0	0	x	x	x	0	1	1	1	0	pattern 1
											0	0	1				1	0	0	0	1	
											0	1	0				1	0	0	0	1	
											0	1	1				1	1	1	1	1	
											1	0	0				1	0	0	0	1	
											1	0	1				1	0	0	0	1	
											1	1	0				1	0	0	0	1	
											1	1	1				0	0	0	0	0	
0	0	0	0	x	1	1	1	0	0	0	0	0	0	x	x	x	1	0	0	0	1	pattern 8
											0	0	1				1	0	0	0	1	
											0	1	0				1	0	0	0	1	
											0	1	1				1	1	1	1	1	
											1	0	0				1	0	0	0	1	
											1	0	1				1	0	0	0	1	
											1	1	0				1	0	0	0	1	
											1	1	1				0	0	0	0	0	

使用举例 (仅供参考)

```

#include <reg51.h>
#include <intrins.h>
sbit dc=0xa0;      /*P2.0 LCD 的 RS  21*/
sbit rw=0xa1;      /*P2.1 LCD 的 R/W 22*/
sbit cs=0xa4;      /*P2.4 LCD 的 E   25*/
sfr  lcdbus=0x80;  /*p0LCD 数据 D0=P0.0*/
unsigned int sys10mscounter;
unsigned char syslimitcounter;
char path1[8]={0x00,0x1f,0x00,0x1f,0x00,0x1f,0x00,0x1f};/*自定义符号: 横 1*/
char path2[8]={0x1f,0x00,0x1f,0x00,0x1f,0x00,0x1f,0x00};/*自定义符号: 横 2*/
char pats1[8]={0x15,0x15,0x15,0x15,0x15,0x15,0x15,0x15};/*自定义符号: 竖 1*/
char pats2[8]={0x0a,0x0a,0x0a,0x0a,0x0a,0x0a,0x0a,0x0a};/*自定义符号: 竖 2*/
void soft_nop() {}
void soft_10ms()/******12MHZ 提供 10MS 软件延时*****/
{ register int i;
  for(i=0;i<711;i++);
}
void soft_20ms()/******12MHZ 提供 20MS 软件延时*****/
{ soft_10ms();
  soft_10ms();
}
    
```

```

void hard_10ms(unsigned int delaytime) /*基于 10MS 的硬件延时*/
{
    sys10mscounter=delaytime;
    while(sys10mscounter);
}

unsigned char data lcdcounter;
bit lcdusing1, lcdusing2;
bit lcd_checkbusy() /*检查 LCD 忙*/
{
    register lcdstate;
    dc=0;          /*dc=1 为数据, =0 为命令.*/
    rw=1;          /*rw=1 为读, =0 为写.*/
    cs=1;          /*cs=1 选通.*/
    soft_nop();
    lcdstate=lcdbus;
    cs=0;
    return((bit)(lcdstate&0x80));
}

void lcd_wrcmd(unsigned char lcdcmd) /*写 LCD 命令*/
{
    lcdusing1=1;
    while(lcd_checkbusy());
    lcdbus=lcdcmd;
    dc=0;          /*dc=1 为数据, =0 为命令.*/
    rw=0;          /*rw=1 为读, =0 为写.*/
    cs=1;          /*cs=1 选通.*/
    soft_nop();
    cs=0;
    lcdbus=0xff;
    lcdusing1=0;
}

void lcd_moveto(char position) /*移动光标到指定位. 0-79*/
{
    register cmd=0x80;
    lcdcounter=position;
    if (position > 59)
        position += 0x18;
    else
    {
        if (position > 39) position -= 0x14;
        else
        {
            if (position > 19) position += 0x2c;
        }
    }
    cmd=cmd|position;
    lcd_wrcmd(cmd);
}

void lcd_wrdata(char lcddata) /*在当前显示位置显示数据*/
{
    char i;
    lcdusing2=1;
}

```

```

while(lcd_checkbusy());
if(lcdcounter==20){
    lcd_moveto(20);
    while(lcd_checkbusy());
}
if(lcdcounter==40){
    lcd_moveto(40);
    while(lcd_checkbusy());
}
if(lcdcounter==60){
    lcd_moveto(60);
    while(lcd_checkbusy());
}
if(lcdcounter==80){
    lcd_moveto(0);
    while(lcd_checkbusy());
    lcdcounter=0;
}          /*为通用而如此*/
lcdcounter++;
lcdbus=lcddata;
dc=1;          /*dc=1 为数据,=0 为命令.*/
rw=0;          /*rw=1 为读,=0 为写.*/
cs=1;          /*cs=1 选通.*/
soft_nop();
cs=0;
lcdbus=0xff;
lcdusing2=0;
}
void lcd_string(char *strpoint) /*在当前显示位置显示 LCD 字符串*/
{
    register i=0;
    while(strpoint[i]!=0){
        lcd_wrdta(strpoint[i]);
        i++;
    }
}
void lcd_init()/*初始化*/
{
    lcd_wrcmd(0x38);    /*设置 8 位格式, 2 行, 5*7*/
    lcd_wrcmd(0x0c);    /*整体显示, 关光标, 不闪烁*/
    lcd_wrcmd(0x06);    /*设定输入方式, 增量不移位*/
    lcd_wrcmd(0x01);    /*清除显示*/
    lcdcounter=0;
}
void lcd_cls()/*清除显示*/
{
    lcd_wrcmd(0x01);
    lcdcounter=0;
}

```


High 4BIT	MSB	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
LOW 4BIT	0000		0	@	P	`	p				一	夕	ミ	α	p
LSB xxxx0000	RAM (1)		0	@	P	`	p				一	夕	ミ	α	p
xxxx0001	(2)	!	1	A	Q	a	q			。	ア	チ	ム	ä	q
xxxx0010	(3)	“	2	B	R	b	r			┌	イ	ツ	メ	β	θ
xxxx0011	(4)	#	3	C	S	c	s			└	ウ	テ	モ	ε	∞
xxxx0100	(5)	\$	4	D	T	d	t			,	エ	ト	ヤ	μ	Ω
xxxx0101	(6)	%	5	E	U	e	u			•	オ	ナ	ユ	σ	ü
xxxx0110	(7)	&	6	F	V	f	v			、	カ	ニ	ヨ	ρ	Σ
xxxx0111	(8)	'	7	G	W	g	w			ア	キ	ヌ	ラ	g	π
xxxx1000	(1)	(8	H	X	h	x			イ	ク	ネ	リ	√	$\frac{_}{x}$
xxxx1001	(2))	9	I	Y	i	y			ウ	ケ	ノ	ル	∴	y
xxxx1010	(3)	*	:	J	Z	j	z			エ	コ	ハ	レ	j	千
xxxx1011	(4)	+	;	K	[k	{			オ	サ	ヒ	ロ	`	万
xxxx1100	(5)	,	<	L	¥	l				ヤ	シ	フ	ワ	φ	円
xxxx1101	(6)	-	=	M]	m	}			ユ	ス	ヘ	ン	キ	÷
xxxx1110	(7)	.	>	N	^	n	→			ヨ	セ	ホ	、	$\frac{_}{n}$	
xxxx1111	(8)	/	?	O	_	o	←			ツ	ソ	マ	°	Ö	■