

# JHD162A SERIES

## CHARACTERISTICS :

DISPLAY CONTENT : 16 CHAR x 2ROW

CHAR. DOTS : 5 x 8

DRIVING MODE : 1/16D

AVAILABLE TYPES :

TN , STN(YELLOW GREEN, GREY, B/W)

REFLECTIVE, WITH EL OR LED BACKLIGHT

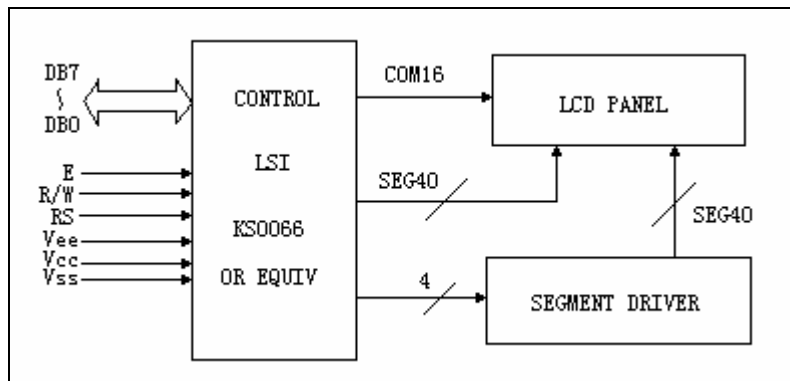
EL/100VAC , 400HZ

LED/4.2VDC

PARAMETER ( $V_{DD}=5.0V \pm 10\%$ ,  $V_{SS}=0V$ ,  $T_a=25$  )

| Parameter           | Symbol          | Testing Criteria | Standard Values |      |          | Unit |
|---------------------|-----------------|------------------|-----------------|------|----------|------|
|                     |                 |                  | Min.            | Typ. | Max      |      |
| Supply voltage      | $V_{DD}-V_{SS}$ | -                | 4.5             | 5.0  | 5.5      | V    |
| Input high voltage  | $V_{IH}$        | -                | 2.2             | -    | $V_{DD}$ | V    |
| Input low voltage   | $V_{IL}$        | -                | -0.3            | -    | 0.6      | V    |
| Output high voltage | $V_{OH}$        | $-I_{OH}=0.2mA$  | 2.4             | -    | -        | V    |
| Output low voltage  | $V_{OL}$        | $I_{OL}=1.2mA$   | -               | -    | 0.4      | V    |
| Operating voltage   | $I_{DD}$        | $V_{DD}=5.0V$    | -               | 1.5  | 3.0      | mA   |

## APPLICATION CIRCUIT



## DIMENSIONS/DISPLAY CONTENT

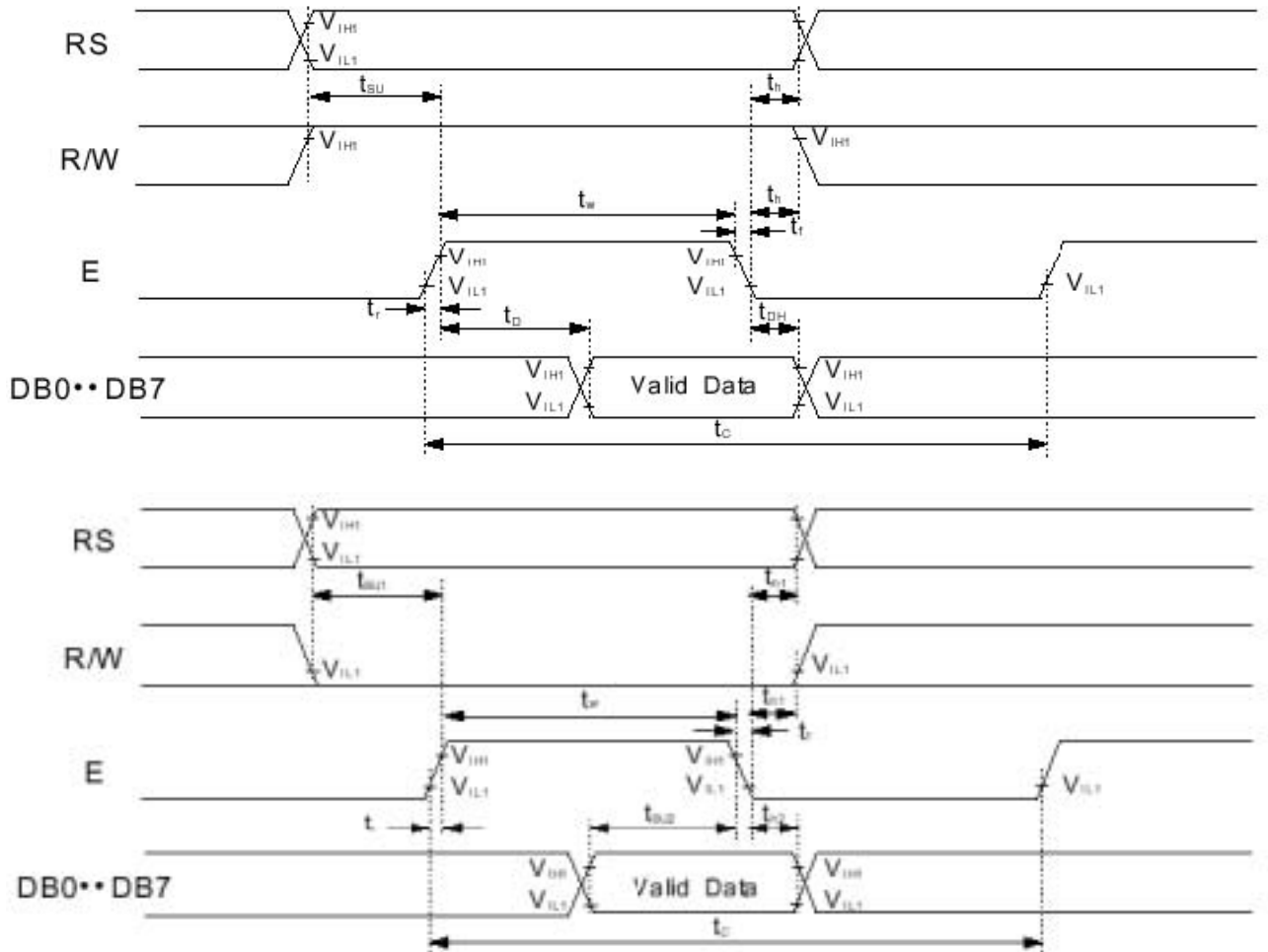


**Table 12. AC Characteristics ( $V_{DD} = 4.5V \sim 5.5V$ ,  $T_a = -30 \sim +85^{\circ}C$ )**

| Mode                           | Characteristic            | Symbol     | Min. | Typ. | Max. | Unit |
|--------------------------------|---------------------------|------------|------|------|------|------|
| Write Mode<br>(Refer to Fig-6) | E Cycle Time              | $t_c$      | 500  | -    | -    | ns   |
|                                | E Rise / Fall Time        | $t_{R,tF}$ | -    | -    | 20   |      |
|                                | E Pulse Width (High, Low) | $t_w$      | 230  | -    | -    |      |
|                                | R/W and RS Setup Time     | $t_{su1}$  | 40   | -    | -    |      |
|                                | R/W and RS Hold Time      | $t_{H1}$   | 10   | -    | -    |      |
|                                | Data Setup Time           | $t_{su2}$  | 80   | -    | -    |      |
|                                | Data Hold Time            | $t_{H2}$   | 10   | -    | -    |      |
| Read Mode<br>(Refer to Fig-7)  | E Cycle Time              | $t_c$      | 500  | -    | -    | ns   |
|                                | E Rise / Fall Time        | $t_{R,tF}$ | -    | -    | 20   |      |
|                                | E Pulse Width (High, Low) | $t_w$      | 230  | -    | -    |      |
|                                | R/W and RS Setup Time     | $t_{su}$   | 40   | -    | -    |      |
|                                | R/W and RS Hold Time      | $t_H$      | 10   | -    | -    |      |
|                                | Data Output Delay Time    | $t_D$      | -    | -    | 120  |      |
|                                | Data Hold Time            | $t_{DH}$   | 5    | -    | -    |      |

**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<br>(Refer to Fig-6) | 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     | $t_{su1}$  | 60   | -    | -    |      |
|                                | R/W and RS Hold Time      | $t_{H1}$   | 20   | -    | -    |      |
|                                | Data Setup Time           | $t_{su2}$  | 195  | -    | -    |      |
|                                | Data Hold Time            | $t_{H2}$   | 10   | -    | -    |      |
| Read Mode<br>(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    | -    | -    |      |

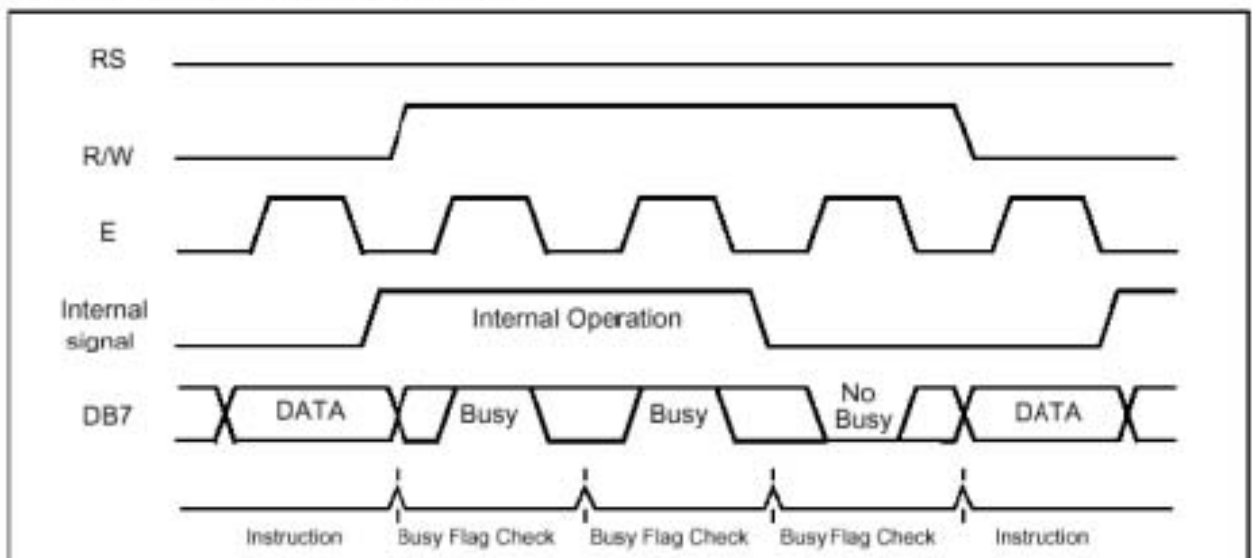


**Write Mode Timing Diagram**

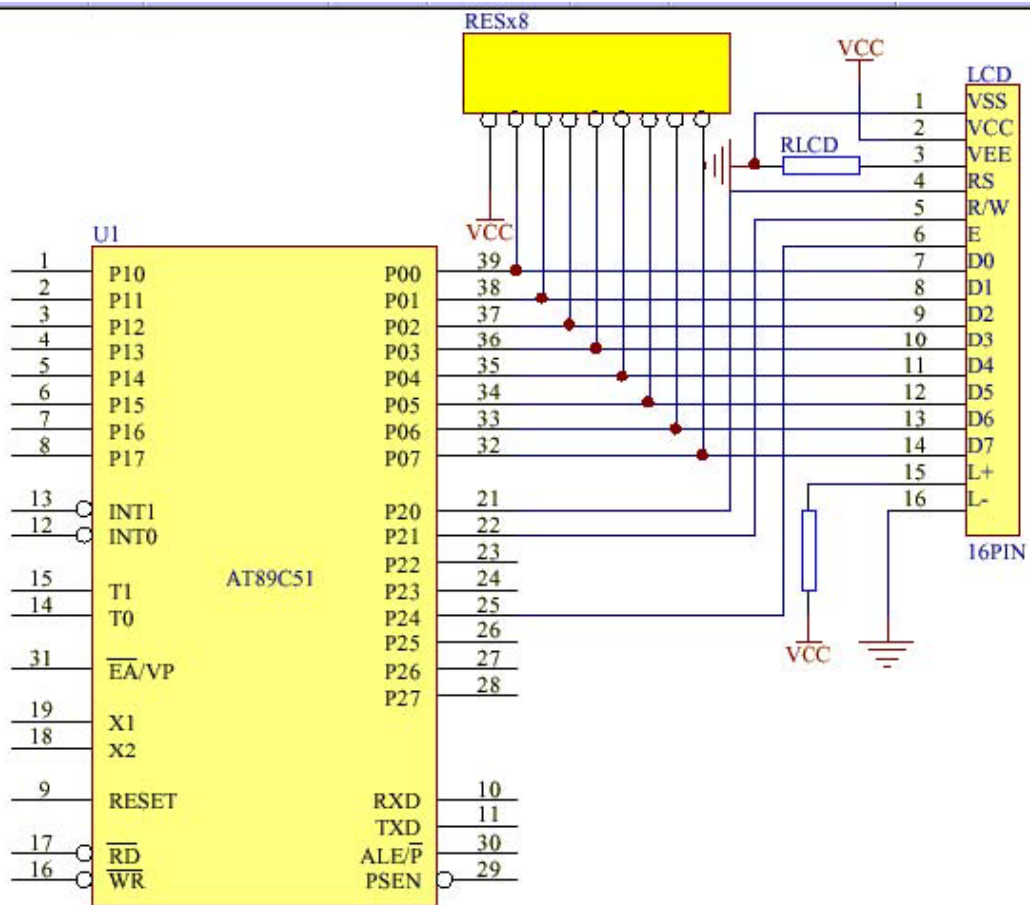
**Timing**

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.



## Connection



■ 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  |                |

Example

```

#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_wldata(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_wldata(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; } void timer0(void) interrupt 1 /*T0 中断*/ { TH0=0xd8; /*12M,10ms*/
    TL0=0xf6;
    TR0=1;
    if(sys10mscounter!=0)sys10mscounter - - ; /*定时器10ms*/
    if(syslimitcounter!=0)syslimitcounter - - ; /*定时器10ms*/

}

    main()
    {
        unsigned char j;
        IE=0;P0=0xff;P1=0xff;P2=0xff;P3=0xff; /*初始化T*/
        lcd_init();soft_20ms();
        TMOD=0x51;
        TH0=0xd8; /*12M,10ms*/
        TL0=0xf6;
        TR0=1;ET0=1;EA=1;

        while(1)
        {
            /*全黑横一横二竖一竖二U Q ABCD... */
            lcd_init(); /*全黑*/
            for(j=0;j<80;j++){lcd_wldata(0xff);}
            hard_10ms(50);
            lcd_init(); /*横一可参考自行设计符号*/
            lcd_wrcmd(0x40);
            for(j=0;j<8;j++){lcd_wldata(path1[j]);

            for(j=0;j<100;j++){lcd_wldata(0);
            hard_10ms(50);
            lcd_init(); /*横二*/

```

