JXD161A **SERIES**

CHARACTERISTICS

DISPLAY CONTENT   16CHAR x 1ROW

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± 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}=02mA	2.4	-	-	V
Output low voltage	V_{OL}	I_{OL}=1.2mA	-	-	0.4	V
Operating voltage	I_{DD}	V_{DD}=5.0V	-	1.3	3.0	mA

APPLICATION CIRCUIT
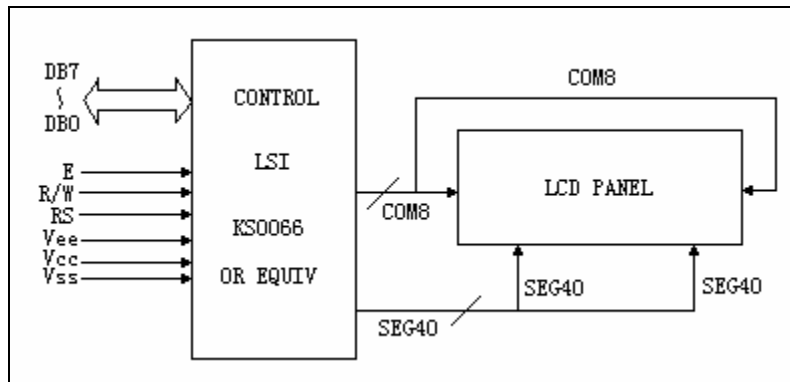
DIMENSIONS/DISPLAY CONTENT

TYPE	*1	*2
R&EL	4.5	11.0
LED	7.6	14.0
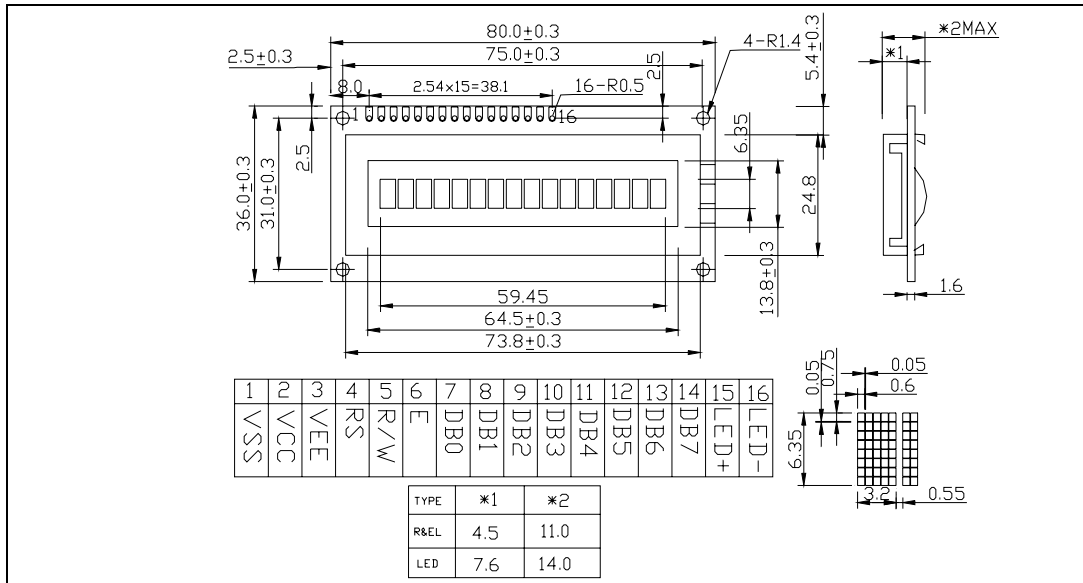
PIN CONFIGURATION

| 1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
VSS	VCC	VEE	RS	R/W	E	DB0	DB1	DB2	DB3	DB4	DB5	DB6	DB7	LED+	LED-

■ **AC Characteristics Read Mode Timing Diagram**

Table 12. AC Characteristics (V_{DD} = 4.5V ~ 5.5V, Ta = -30 ~ +85°C)

Mode	Characteristic	Symbol	Min.	Typ.	Max.	Unit
Write Mode (Refer to Fig-6)	E Cycle Time	tc	500	-	-	ns
	E Rise / Fall Time	t_R, t_F	-	-	20	
	E Pulse Width (High, Low)	tw	230	-	-	
	R/W and RS Setup Time	tsu1	40	-	-	
	R/W and RS Hold Time	t_{H1}	10	-	-	
	Data Setup Time	tsu2	80	-	-	
	Data Hold Time	t_{H2}	10	-	-	
Read Mode (Refer to Fig-7)	E Cycle Time	tc	500	-	-	ns
	E Rise / Fall Time	t_R, t_F	-	-	20	
	E Pulse Width (High, Low)	tw	230	-	-	
	R/W and RS Setup Time	tsu	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 ~ 4.5V, Ta = -30 ~ +85°C)

| Mode	Characteristic	Symbol	Min.	Typ.	Max.	Unit
Write Mode (Refer to Fig-6)	E Cycle Time	tc	1000	-	-	ns
	E Rise / Fall Time	$t_R t_F$	-	-	25	
	E Pulse Width (High, Low)	tw	450	-	-	
	R/W and RS Setup Time	tsu1	60	-	-	
	R/W and RS Hold Time	t_{H1}	20	-	-	
	Data Setup Time	tsu2	195	-	-	
	Data Hold Time	t_{H2}	10	-	-	
Read Mode (Refer to Fig-7)	E Cycle Time	tc	1000	-	-	ns
	E Rise / Fall Time	t_R, t_F	-	-	25	
	E Pulse Width (High, Low)	tw	450	-	-	
	R/W and RS Setup Time	tsu	60	-	-	
	R/W and RS Hold Time	t_H	20	-	-	
	Data Output Delay Time	t_D	-	-	360	
	Data Hold Time	t_{DH}	5	-	-	
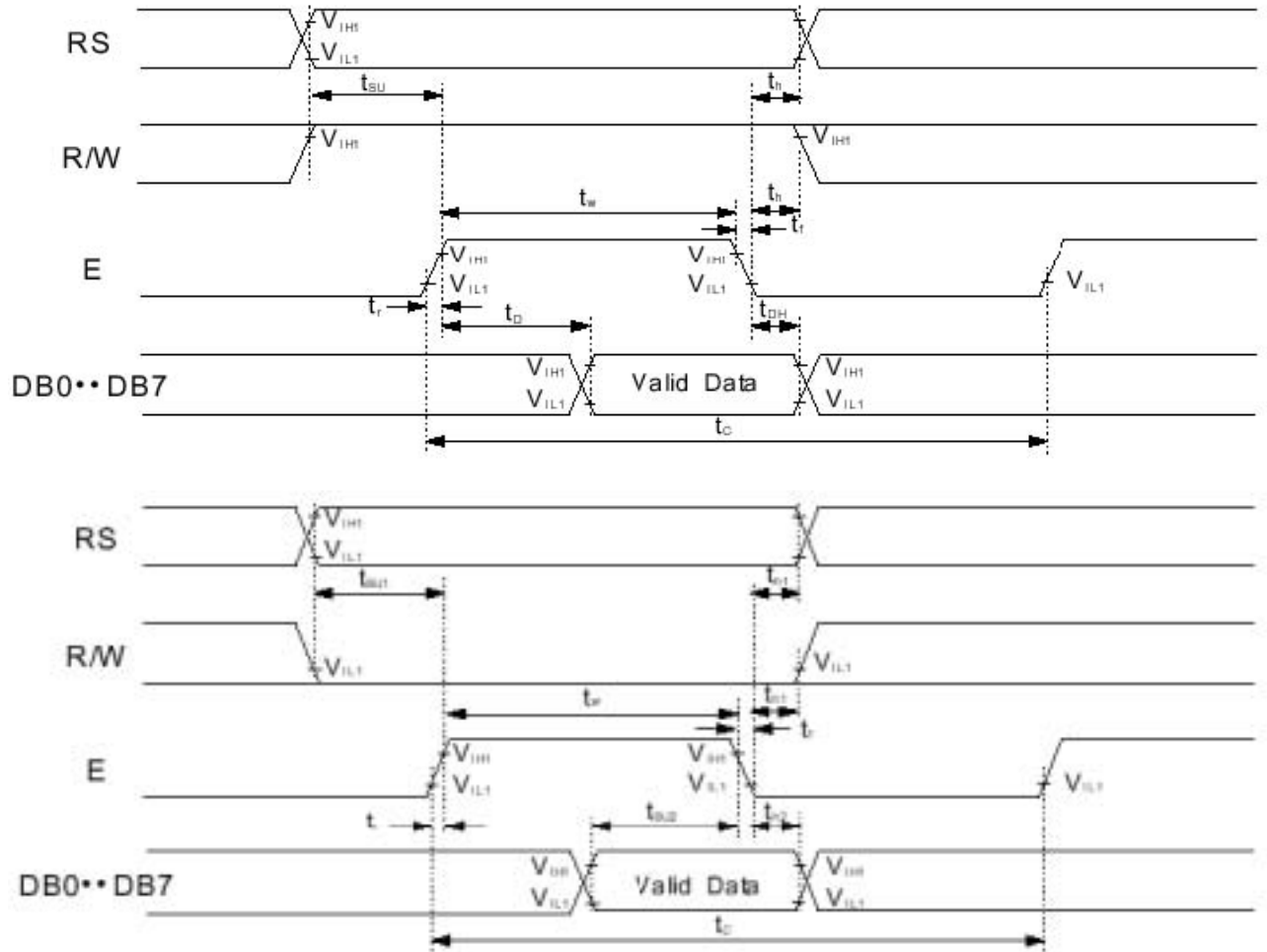
Write Mode Timing Diagram

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.
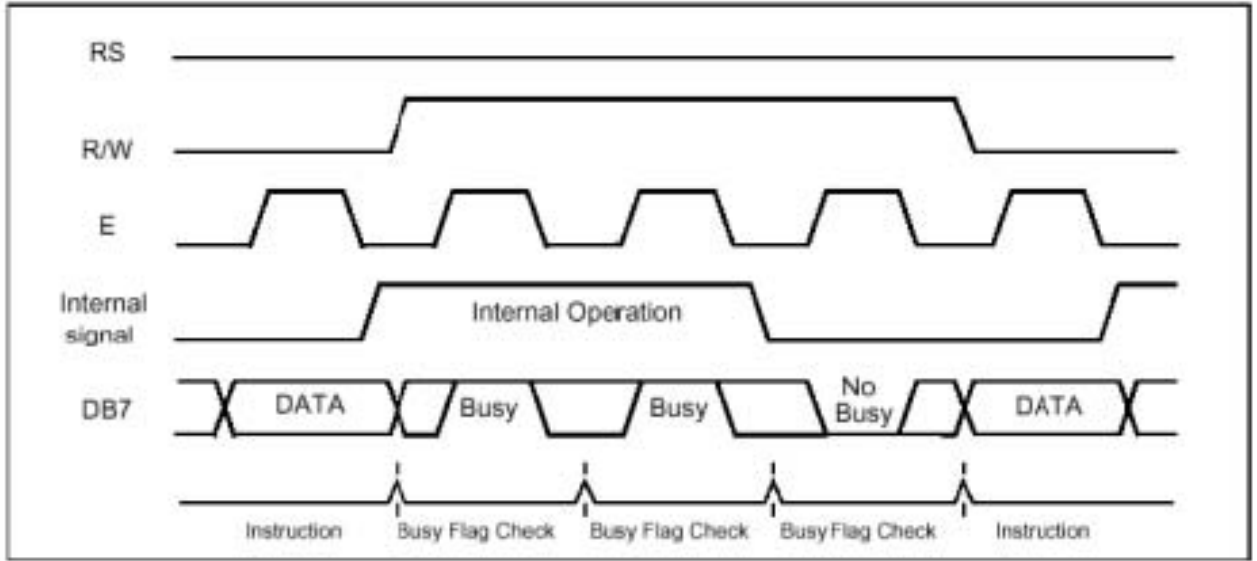
Connection
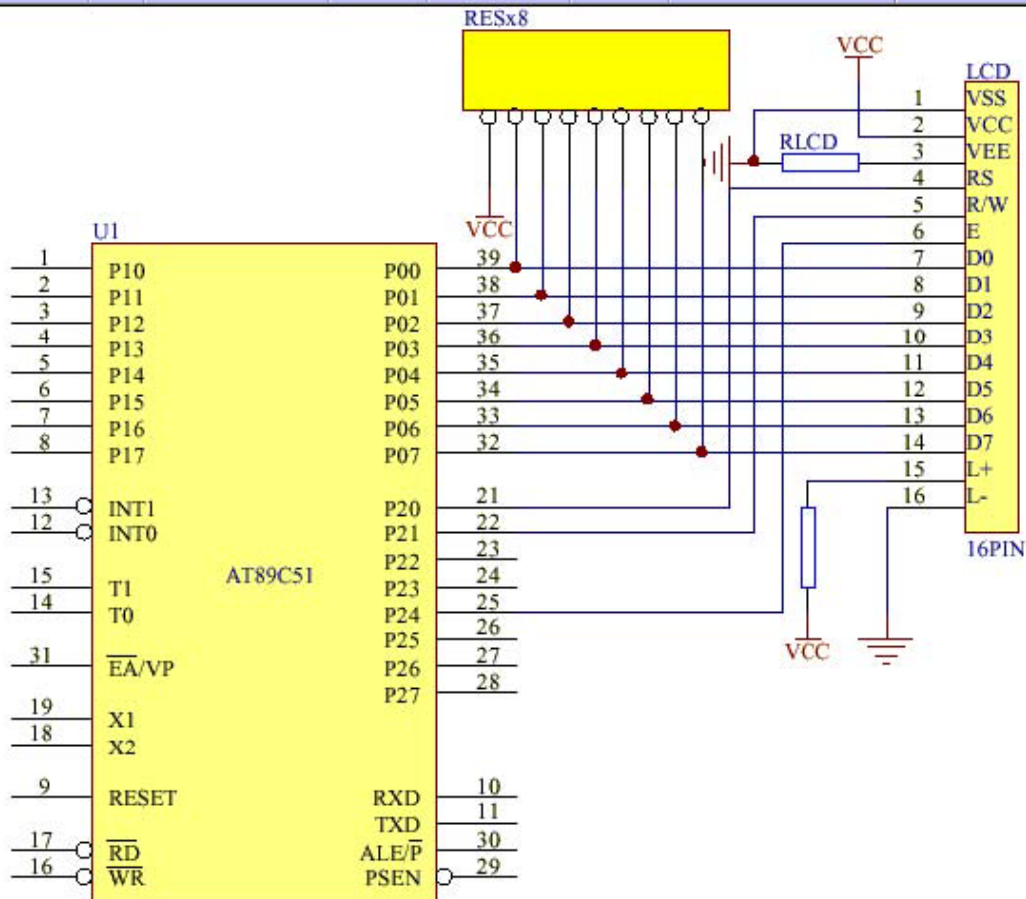
■ **CGROM**

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	×	0	0	0	0	0	0	0	0	0	×	×	×	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	×	1	1	1	0	0	0	0	0	0	×	×	×	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	

Operation Example

```c
#include <reg51.h>
#include <intrins.h>

sbit dc=0xa0;        /*P2.0  LCD's RS    21*/
sbit rw=0xa1;        /*P2.1  LCD's R/W  22*/
sbit cs=0xa4;        /*P2.4  LCD's E     25*/
sfr  lcdbus=0x80;    /*p0 LCD data D0=P0.0*/
unsigned int sys10mscounter;
unsigned char syslimitcounter;
char path1[8]={0x00,0x1f,0x00,0x1f,0x00,0x1f,0x00,0x1f};/*          1*/
char path2[8]={0x1f,0x00,0x1f,0x00,0x1f,0x00,0x1f,0x00};/*          2*/
char pats1[8]={0x15,0x15,0x15,0x15,0x15,0x15,0x15,0x15};/*          1*/
char pats2[8]={0x0a,0x0a,0x0a,0x0a,0x0a,0x0a,0x0a,0x0a};/*          2*/
void soft_nop(){}
void soft_10ms()/***********12MHZ   10MS         ***********/
{    register int i;
for(i=0;i<711;i++);
```

```c
}
void soft_20ms()/**********12MHZ        20MS              **********/
{    soft_10ms();
soft_10ms();
}
void hard_10ms(unsigned int delaytime) /*      10MS                 */
{    sys10mscounter=delaytime;
while(sys10mscounter);
}
unsigned char data lcdcounter;
bit lcdusing1,lcdusing2;
bit lcd_checkbusy()/*      L CD     */
{    register lcdstate;
    dc=0;                   /*dc=1        ,=0         .*/
    rw=1;                   /*rw=1     ,=0        .*/
    cs=1;                   /*cs=1      .*/
soft_nop();
lcdstate=lcdbus;
cs=0;
return((bit)(lcdstate&0x80));
}
void lcd_wrcmd(unsigned char lcdcmd) /*   L CD        */
{    lcdusing1=1;
while(lcd_checkbusy());
lcdbus=lcdcmd;
        dc=0;          /*dc=1        ,=0           .*/
        rw=0;          /*rw=1     ,=0        .*/
        cs=1;          /*cs=1      .*/
soft_nop();

    cs=0;
    lcdbus=0xff;
    lcdusing1=0;

}
            void lcd_moveto(char position) /*                        .0-79*/
            {register cmd=0x80;

        lcdcounter=position;
            if (position > 59)
            position += 0x18;
            else
```

```c
        {   if (position >  39)position - = 0x14;
              else
              {                  if (position >  19)position += 0x2c;
              }
        }


    cmd=cmd|position;
    lcd_wrcmd(cmd); }void lcd_wrdata(char lcddata) /*                              */ {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;
        }/*                 */

    lcdcounter++;
    lcdbus=lcddata;
    dc=1; /*dc=1         ,=0          .*/
    rw=0; /*rw=1      ,=0        .*/
    cs=1; /*cs=1      .*/
    soft_nop();
    cs=0;

    lcdbus=0xff;
    lcdusing2=0; }void lcd_string(char *strpoint) /*                            LCD           */
{register i=0;
    while(strpoint[i]!=0){
```

```c
                lcd_wrdata(strpoint[i]);
                i++;
                }

}void lcd_init()/*          */

{   lcd_wrcmd(0x38);        /*      8         ,2    ,5*7*/
    lcd_wrcmd(0x0c);        /*             ,        ,         */
    lcd_wrcmd(0x06);        /*                ,              */
    lcd_wrcmd(0x01);        /*          */
    lcdcounter=0;
}

void lcd_cls()/*          */ {lcd_wrcmd(0x01);
    lcdcounter=0; }void timer0(void) interrupt 1 /*T 0      */ {TH0=0xd8; /*12M,10ms*/
    TL0=0xf6;
    TR0=1;
    if(sys10mscounter!=0)sys10mscounter--; /*          10ms*/
    if(syslimitcounter!=0)syslimitcounter--; /*          10ms*/

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

    while(1)
    {
    /*                         U Q ABCD... */
    lcd_init(); /*      */
    for(j=0;j<80;j++){lcd_wrdata(0xff);}
    hard_10ms(50);
    lcd_init(); /*                         */
    lcd_wrcmd(0x40);
    for(j=0;j<8;j++)lcd_wrdata(path1[j]);

    for(j=0;j<100;j++)lcd_wrdata(0);
    hard_10ms(50);
    lcd_init(); /*      */
```

```
lcd_wrcmd(0x40);
for(j=0;j<8;j++)lcd_wrdata(path2[j]);
for(j=0;j<100;j++)lcd_wrdata(0);
hard_10ms(50);
lcd_init(); /*     */
lcd_wrcmd(0x40);
for(j=0;j<8;j++)lcd_wrdata(pats1[j]);
for(j=0;j<100;j++)lcd_wrdata(0);
hard_10ms(50);
lcd_init(); /*     */
lcd_wrcmd(0x40);
for(j=0;j<8;j++)lcd_wrdata(pats2[j]);
for(j=0;j<100;j++)lcd_wrdata(0);
hard_10ms(50);
lcd_init();
lcd_string("UUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUU
UUUUUUUUUUUUUUUUUUUUUUUU

UUUUU"); hard_10ms(50); lcd_init();
lcd_string("QQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQ
QQQQQQQQQQQQQQQQQQQQQQQQ
QQQQQ"); hard_10ms(50); lcd_init();
lcd_string("ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789abcdefghijklmnopqrstuvwx
yz0123456789+-!
#$%&?"); hard_10ms(50); }
}
```