SPECIFICATION SHEET FOR:

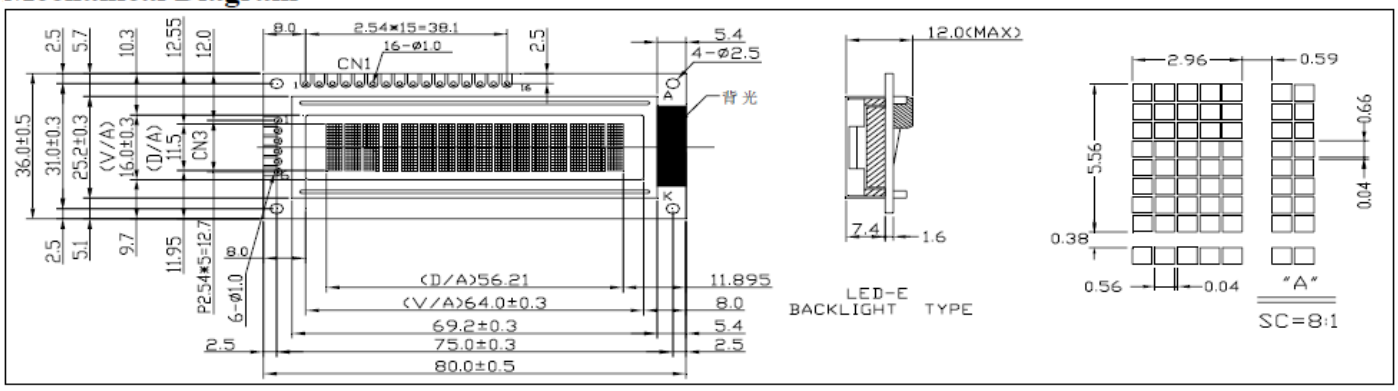
FDS16x2(75x31)SDC

Mechanical Specification

ITEM	STANDARD VALUE	UNIT
NUMBER OF CHARACTERS	16 CHARACTERS X 2 LINES	--
CHARACTER FORMAT	5 X 8 DOTS	--
MODULE DIMENSION EDGE LED BACKLIGHT	80.0 (W) X 36.0 (H) X 12.0 (T)	mm
VIEWING DISPLAY AREA	64.0 (W) X 16.0 (H)	mm
ACTIVE DISPLAY AREA	56.21 (W) X 11.50 (H)	mm
CHARACTER SIZE	2.96 (W) X 5.56 (H)	mm
CHARACTER PITCH	3.55 (W) X 5.94 (H)	mm
DOT SIZE	0.56 (W) X 0.60 (H)	mm
DOT PITCH	0.60 (W) X 0.70 (H)	mm

● EDEC LED BACKLIGHT COLOR	WHITE			
BACKLIGHT INPUT	DC +4.0V	V	20(TYPE)	mA
BACKLIGHT LIFT TIME	20,000 (AVOID LIGHTING CONTINUOUSLY，Ta=25℃)			HR.

Mechanical Diagram



Absolute Maximum Ratings

ITEM		SYMBOL	MIN.	TYPE	MAX.	UNIT
INPUT VOLAGE		VI	VSS	—	VDD	V
SUPPLY VOLTAGE FOR LOGIC		VDD-VSS	—	5.0	5.5	V
SUPPLY VOLTAGE FOR LCD		VLCD	—	—	5.5	V
STN	NORMAL TEMPERATURE RANGE	OPTERATING	0～+50	STORAGE	-10～+60	℃
STATIC ELECTRICITY		Be sure that you are grounded when handing LCM.				

Electrical Characteristics

ITEM	SYN	CONDITION	MIN.	TYPE	MAX.	UNIT
SUPPLY VOLTAGE FOR LOGIC	VDD－VSS	——	4.5	5.0	5.5	V
SUPPLY VOLTAGE FOR LCD	VLCD	Ta＝0℃ Ta＝25℃ Ta＝+50℃	—— ——	4.0 4.3 4.6	—— 4.5 ——	V V V
INPUT HIGH VOLTAGE	VIH	——	0.8VDD	——	VDD	V
INPUT LOW VOLTAGE	VIL	——	0	——	0.1VDD	V
OUTPUT HIGH VOLTAGE	VOH	——	0.8VDD	——	——	V
OUTPUT LOW VOLTAGE	VOL	——	——	——	0.1VDD	V
SUPPLY CURRENT	IDD	VDD=+5V	——	3.0	4.5	mA

Optical Characteristics

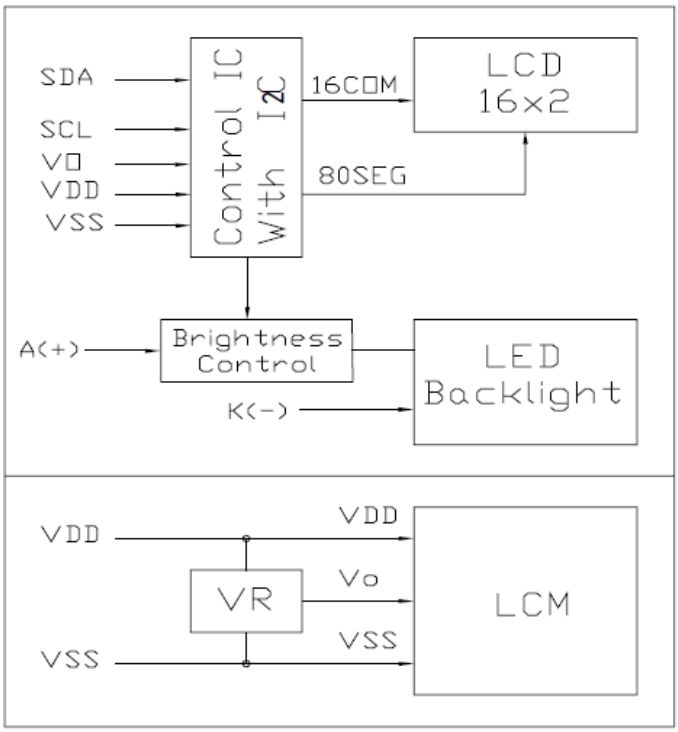
Ta at 25℃

ITEM	SYM	CONDITION	MIN.	TYPE	MAX.	UNIT
VIEW ANGLE (BOTTOM／TOP)	θ1～θ2	CR≧2	——	45° / 35°	——	deg.
VIEW ANGLE (LEFT／RIGHT)	φ1・φ2	CR≧2	——	35° / 35°	——	deg.
CONTRAST RATIO	CR	——	——	8	——	——
RESPONSE TIME (RISE)	TON／Tr	——	——	170	——	mS
RESPONSE TIME (DECAY)	TOFF／Tf	——	——	220	——	mS

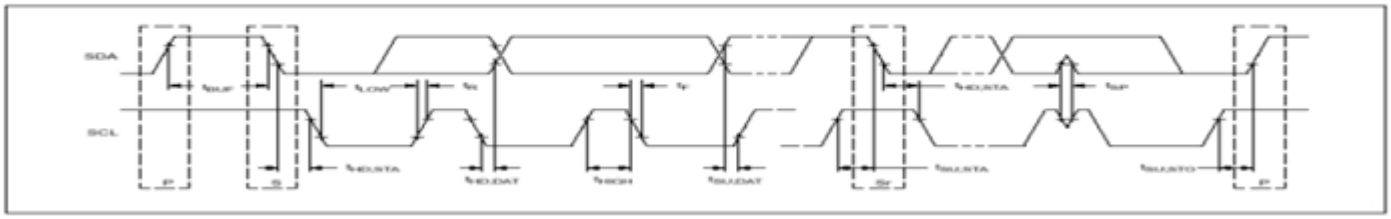
Interface Pin Connections(CN3)

NO	SYMBOL	LEVEL	FUNCTION
1	VSS	--	GND (0V)
2	VDD	--	DC +5V
3	VO	H/L	Contrast Adjust
4	SDA	H/L	Serial Data Line
5	SCL	H/L	Serial Clock Line
6	A(+)	--	LED Backlight

Block Diagram



Timing Control I2C Bus Timing



I2C Bus Timing Specification

PARAMETER	SYMBOL	STANDARD-MODE		UNIT
		MIN.	MAX.	
SCL clock frequency	f_{scl}	-	100	kHz
Hold tie (repeated) START condition.	$t_{HD;STA}$	4.0	-	us
Low period of the SCL clock	t_{LOW}	4.7	-	us
HIGH period of the SCL clock	t_{HIGH}	4.0	-	us
Set-up time for a repeated START condition	$t_{SU;STA}$	4.7	-	us
Data hold time. For I2C-bus device	$t_{HD;DAT}$	0	3.45	us
Data set-up time	$t_{SU;DAT}$	250	-	ns
Rise time of both SDA and SCL signals	t_r	30	1000	ns
Fall time of both SDA and SCL signals	t_f	30	300	ns
Set-up time for STOP condition	$t_{SU;STO}$	4.0	-	us
Bus free time between a STOP and START	t_{BUF}	4.7	-	us
Capacitive load for each bus line	C_b	-	400	pF
Noise margin at the LOW level for each connected device (including hysteresis)	V_{nL}	0.1 V_{DD}	-	V
Noise margin at the HIGH level for each connected device (including hysteresis)	V_{nH}	0.2 V_{DD}	-	V

Note 1: It depends on the "high" period time of SCL.

Standard I²C-bus devices

Symbol	Parameter	Conditions	Standard-mode		Unit
			Min	Max	
f_{SCL}	SCL clock frequency		10	100	kHz
$t_{HD;STA}$	hold time (repeated) START condition	After this period, the first clock pulse is generated.	4.0	-	µs
t_{LOW}	LOW period of the SCL clock		4.7	-	µs
t_{HIGH}	HIGH period of the SCL clock		4.0	-	µs
$t_{SU;STA}$	set-up time for a repeated START condition		4.7	-	µs
$t_{HD;DAT}$		CBUS compatible masters	5.0	-	µs
		I²C-bus devices	0	-	µs
$t_{SU;DAT}$	data set-up time		250	-	ns
t_r	rise time of both SDA and SCL signals		-	1000	ns
t_f	fall time of both SDA and SCL signals		-	300	ns
$t_{SU;STO}$	set-up time for STOP condition		4.0	-	µs
t_{BUF}	bus free time between a STOP and START condition		4.7	-	µs
C_b	capacitive load for each bus line		-	400	pF
$t_{VD;DAT}$	data valid time		-	3.45	µs
$t_{VD;ACK}$	data valid acknowledge time		-	3.45	µs
V_{nL}	noise margin at the LOW level		$0.1V_{DD}$	-	V
V_{nH}	noise margin at the HIGH level	for each connected device (including hysteresis)	$0.2V_{DD}$	-	V

[1] $t_{HD;DAT}$ is the data hold time that is measured from the falling edge of SCL, applies to data in transmission and the acknowledge.

[2] A device must internally provide a hold time of at least 300 ns for the SDA signal to bridge the undefined region of the falling edge of SCL.

[3] The maximum $t_{HD;DAT}$ could be 3.45 µs and 0.9 µs for Standard-mode and Fast-mode, but must be less than the maximum of $t_{VD;DAT}$ or $t_{VD;ACK}$ by a transition time. This maximum must only be met if the device does not stretch the LOW period (t_{LOW}) of the SCL signal. If the clock stretches the SCL, the data must be valid by the set-up time before it releases the clock.

[4] A Fast-mode I^2C-bus device can be used in a Standard-mode I^2C-bus system, but the requirement $t_{SU;DAT}$ 250 ns must then be met. This will automatically be the case if the device does not stretch the LOW period of the SCL signal. If such a device does stretch the LOW period of the SCL signal, it must output the next data bit to the SDA line $t_{r(max)}$ + $t_{SU;DAT}$ = 1000 + 250 = 1250 ns (according to the Standard-mode I^2C-bus specification) before the SCL line is released. Also the acknowledge timing must meet this set-up time.

[5] C_b = total capacitance of one bus line in pF.

[6] The maximum t_f for the SDA and SCL bus lines is specified at 300 ns. The maximum fall tim for the SDA output stage t_f is specified at 250 ns. This allows series protection resistors to b connected in between the SDA and the SCL pins and the SDA/SCL bus lines without exceeding the maximum specified t_f.

[7] In Fast-mode Plus, fall time is specified the same for both output stage and bus timing. If series resistors are used, designers should allow for this when considering bus timing.

[8] The maximum bus capacitance allowable may vary from this value depending on the actual operating voltage and frequency of the application.

[9] $t_{VD;DAT}$ = time for data signal from SCL LOW to SDA output (HIGH or LOW,depending on which one is worse).

[10] $t_{VD;ACK}$ = time for Acknowledgement signal from SCL LOW to SDA output (HIGH or LOW depending on which one is worse).

Note：After Power On，initialize LCD to receive command needs 50ms.

Instruction Set for I²C

function	Command Code																Data														DESCRIPTION
	1st byte								2nd byte								1st byte								2nd byte								
	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	
LCM Address	0	1	0	0	0	0	0	0	-								-																LCM I²C Address Send this value Before command
Set cursor To Address	0	0	0	1	1	1	1	1	0	0	1	0	0	1	0	0	X								Y								Set cursor position 01h≦X≦10h 01h≦Y≦02h
Clear Display	0	0	0	0	1	1	0	0	-								-																Clear Display
Initial Display	0	0	0	1	1	0	1	1	0	1	0	0	0	0	0	0	-																initial LCM
Set cursor off/on	0	0	0	1	1	0	1	1	0	1	0	1	1	1	1	1	N								-								Cursor off/on N = 00h / 01h
Backlight Brightness Minus	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	-								-								Brightness Minus
Backlight Brightness Plus	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	1	-								-								Brightness Plus
Write ASCII Code to LCD	Write Data																-																Write ASCII Code to LCD

Set cursor To Address

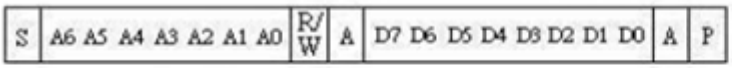
S	A6 A5 A4 A3 A2 A1 A0	R/W	A	D7 D6 D5 D4 D3 D2 D1 D0	A	D7 D6 D5 D4 D3 D2 D1 D0	A	D7 D6 D5 D4 D3 D2 D1 D0	A	D7 D6 D5 D4 D3 D2 D1 D0	A	P
	ID (7BITS)			1F		24		X		Y		

EX:

```
start_bit();
write_byte(0x40);//LCM   ADDRESS
write_byte(0x1F);//SET CURSOR to Address
write_byte(0x24);
write_byte(0x0A);
write_byte(0x02);
stop_bit();
wait_key();
```

==========

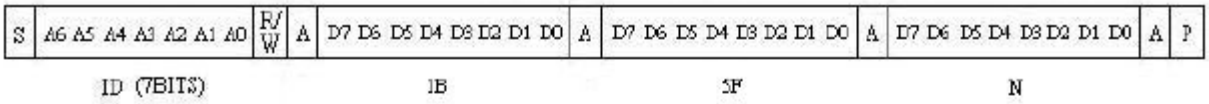
Clear Display




```
EX:     start_bit();
        write_byte(0x40); //LCM   ADDRESS
        write_byte(0x0C);//CLEAR DISPLAY
        stop_bit();
        wait_key();
```

==========

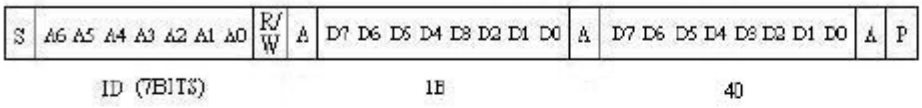
Set cursor On/Off




```
EX1:    start_bit();
        write_byte(0x40); //LCM   ADDRESS
        write_byte(0x1B);//SET CURSOR ON
        write_byte(0x5F);
        write_byte(0x01);
        stop_bit();
        wait_key();
//---------------------------------------
EX2:    start_bit();
        write_byte(0x40); //LCM   ADDRESS
        write_byte(0x1B);//SET CURSOR OFF
        write_byte(0x5F);
        write_byte(0x00);
        stop_bit();
        wait_key();
```

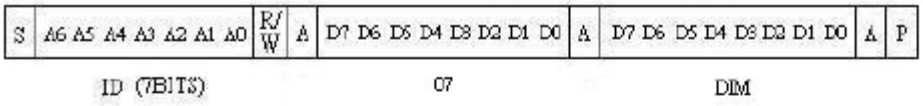
==========

Initial LCD




```
EX:     start_bit();
        write_byte(0x40); //LCM   ADDRESS
        write_byte(0x1B);//INITIAL LCD
        write_byte(0x40);
        stop_bit();
        wait_key();
```

Backlight Brightness Control

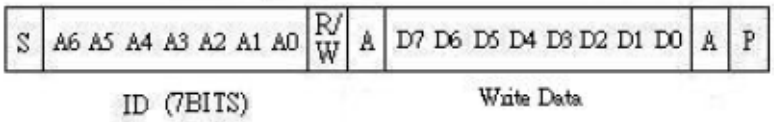



```
EX1:    start_bit();
        write_byte(0x40); //LCM    ADDRESS
        write_byte(0x07);// BACKLIGHT Brightness ( - )
        write_byte(0x00);
        write_byte(0x07);
        write_byte(0x00);
        write_byte(0x07);
        write_byte(0x00);
        write_byte(0x07);
        write_byte(0x00);
        stop_bit();
        wait_key();
```

==========

```
EX2:    start_bit();
        write_byte(0x40); //LCM    ADDRESS
        write_byte(0x07);// BACKLIGHT    Brightness ( + )
        write_byte(0x01);
        write_byte(0x07);
        write_byte(0x01);
        write_byte(0x07);
        write_byte(0x01);
        write_byte(0x07);
        write_byte(0x01);
        stop_bit();
        wait_key();
```

Write ASCII Code to LCM




```
EX:     start_bit();
        write_byte(0x40); //LCM ADDRESS
        for(i = 0x30;i < 0x43;i++)//WRITE ACSII CODE TO LCM
        write_byte(i);
        stop_bit();
        wait_key();
```

==========

write to I²C device external

```
void start_bit()
{
      SDA = 0;
      sdelay();
      SCL = 0;
      sdelay();
}
void stop_bit()
{
      SDA = 0;
      sdelay();
      SCL = 1;
      sdelay();
      SDA = 1;
      sdelay();
}
void wait_ack()
{
      SDA = 1;
      SCL = 1;
      while(SDA == 1);
      sdelay();
      SCL = 0;
      sdelay();
}
void write_byte(unsigned char i2c_data)
{
      char i;
      temp = i2c_data;
      for(i = 0;i < 8;i ++ )
      {
            if(i > 0)
            temp <<= 1;
            SDA = b7;
            SCL = 1;
            sdelay();
            SCL = 0;
            sdelay();
      }
      wait_ack();
}
void delay(unsigned int dly)
{
      while(dly--);
}
void sdelay()//DELAY 5u
{
}
```

Character Generator ROM Map