Interfacing LCD with TIVA Launchpad

e-Yantra Team

Embedded Real Time Systems Lab

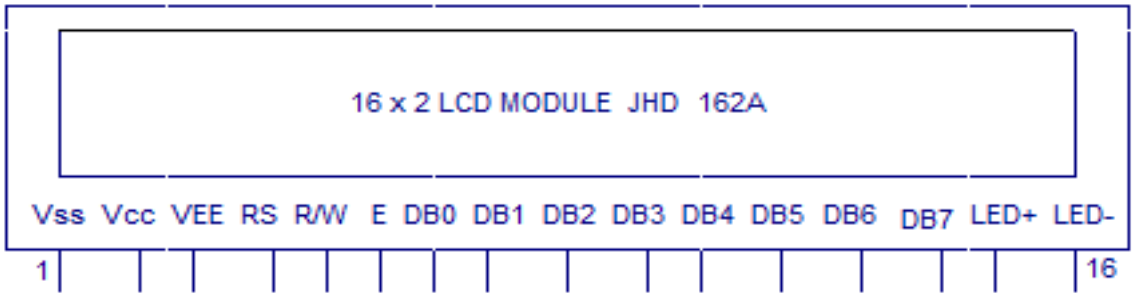
June 22, 2016

Contents

Introduction

- 16X2 is the most commonly used LCD Module
- It can display 32 ASCII characters in 2 lines
- We have used JHD162A module
- It can work in 4 bit and 8 bit mode

Pin Configuration



Pin	Description
Vss	Ground
Vcc	Supply Voltage
Vee	Contrast Voltage
RS	Register Select
RW	Read/Write
E	Enable
D0-D7	Data Lines
LED+,LED-	Backlight Supply

Control Pins

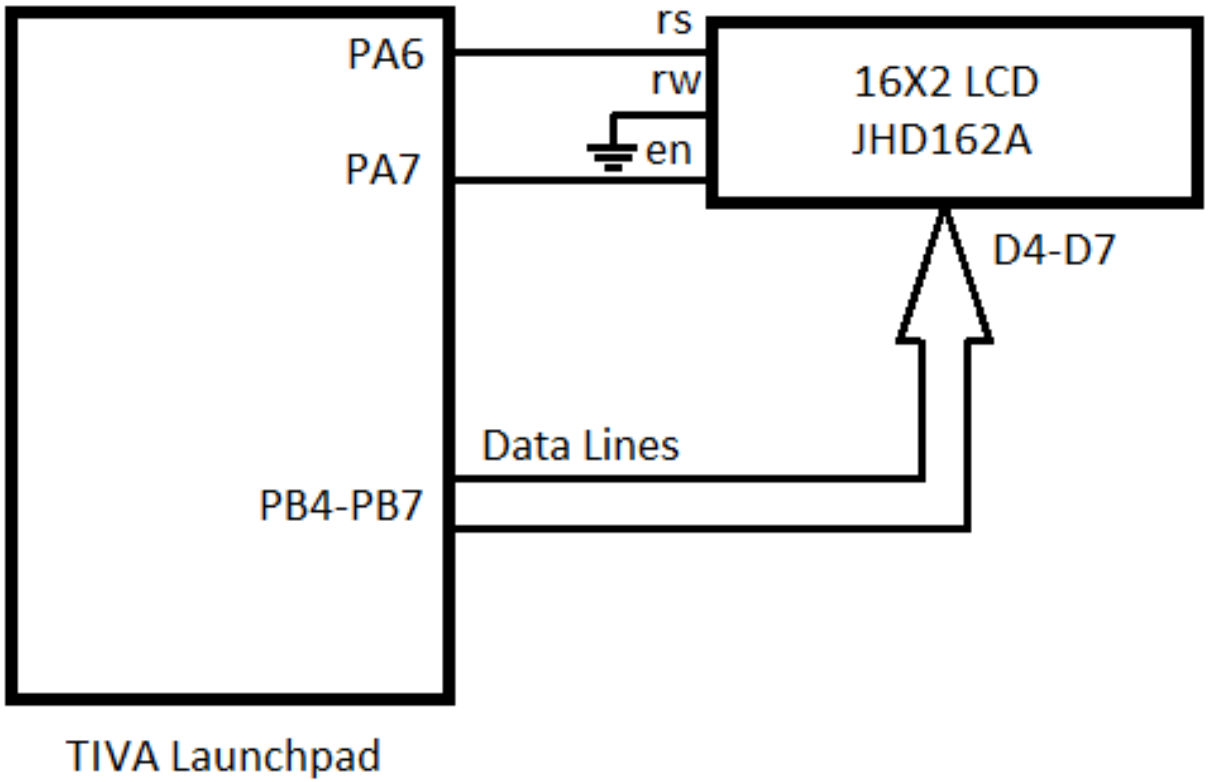
1. Register Select
   - If RS=0 : Command Register
   - If RS=1 : Data Register
2. Read/Write Select
   - If RW=0 : Write Mode
   - If RW=1 : Read Mode
3. Enable
   - Used to latch the data present on data pins.
   - A high to low edge is required to latch the data.

Data Pins

Data Lines:

- There are 8 data lines(D0-D7)
- In 8 bit mode all the data lines are used
- In 4 bit mode data lines(D4-D7) are used
- Characters are sent in ASCII format

LCD Interfacing



Important commands

Description	Hex
Function Set(8 bit interface,2 lines, 5*7 pixels)	38
Function Set(4 bit interface,2 lines, 5*7 pixels)	28
Clear Display Screen	01
Return Home (First line First block)	02
Display ON Cursor blinking	0F
Address for Line 1	80
Address for Line 2	C0
Display ON, Cursor OFF	0C
Increment Cursor	06

LCD Initialization - 4 bit

To send LCD Commands

1. Initialize PORT A and PORT B as output PORTs.
2. Set RS=0, RW=0.
3. Send LCD command.
4. Generate high to low pulse on enable pin.

To send LCD Data

1. Initialize PORT A and PORT B as output PORTs.
2. Set RS=1, RW=0.
3. Send LCD data.
4. Generate high to low pulse on enable pin.

LCD Initilization - 4 bit

For initializing LCD in 4 bit mode, following commands are sent to the LCD.

Function : lcd_set_4bit()

1. Make RS=0, RW=0.
   GPIOPinWrite(GPIO_PORTA_BASE, GPIO_PIN_6 |GPIO_PIN_7, 0x00);
2. Send 3 on Data lines(0x30)
   GPIOPinWrite(GPIO_PORTB_BASE,GPIO_PIN_4 |GPIO_PIN_5|GPIO_PIN_6 |GPIO_PIN_7,0x30);
3. Set the enable pin.
   GPIOPinWrite(GPIO_PORTA_BASE, GPIO_PIN_6 |GPIO_PIN_7, 0x80);
4. Call delay.
   SysCtlDelay(67000);
5. Clear enable pin.
   GPIOPinWrite(GPIO_PORTA_BASE, GPIO_PIN_6 |GPIO_PIN _7, 0x00);

Repeat the above process thrice to set LCD in 4 bit mode.

LCD Initialization - 4 bit

Function : lcdinit()

1. lcd_set_4bit();
2. lcdcmd(0x33);
3. lcdcmd(0x32);
4. lcdcmd(0x28); (To set the LCD in 4 bit mode)
5. lcdcmd(0x01); (To clear the display)
6. lcdcmd(0x06); (To increment the cursor)
7. lcdcmd(0x80); (To select the 1st row and 1st block)

LCD Initialization - 4 bit

Function : lcdcmd()
For sending 8 bit data over 4 data lines, send the data in nibbles instead of bytes.
The command is stored in variable cmd.

1. **Masking of higher nibble -**
   temp = cmd;
   temp = temp & 0xF0;
   lcd_port &= 0x0F;
   lcd_port |= temp;
2. **Masking of lower nibble -**
   cmd = cmd & 0x0F;
   cmd = cmd<<4;
   lcd_port &= 0x0F;
   lcd_port |= cmd;

LCD Initialization - 4 bit

```
void lcdcmd(unsigned char cmd)
{
    unsigned char temp;                          //variable used for masking and storing value
    unsigned char lcd_port;
        temp = cmd;
        temp = temp & 0xF0;
        lcd_port &= 0x0F;
        lcd_port |= temp;
        GPIOPinWrite(GPIO_PORTB_BASE,GPIO_PIN_4 | GPIO_PIN_5| GPIO_PIN_6 | GPIO_PIN_7,lcd_port);//sending higher 4 bits
        GPIOPinWrite(GPIO_PORTA_BASE, GPIO_PIN_6 | GPIO_PIN_7, 0x00);                //RS=0 --- Command Input
        GPIOPinWrite(GPIO_PORTA_BASE, GPIO_PIN_6 | GPIO_PIN_7, 0x80);                //enable high to low
        SysCtlDelay(67000);
        GPIOPinWrite(GPIO_PORTA_BASE, GPIO_PIN_6 | GPIO_PIN_7, 0x00);


        cmd = cmd & 0x0F;
        cmd = cmd<<4;
        lcd_port &= 0x0F;
        lcd_port |= cmd;
        GPIOPinWrite(GPIO_PORTB_BASE,GPIO_PIN_4 | GPIO_PIN_5| GPIO_PIN_6 | GPIO_PIN_7,lcd_port);//sending lower 4 bits
        GPIOPinWrite(GPIO_PORTA_BASE, GPIO_PIN_6 | GPIO_PIN_7, 0x00);                //RS=0 --- Command Input
        GPIOPinWrite(GPIO_PORTA_BASE, GPIO_PIN_6 | GPIO_PIN_7, 0x80);                //enable high to low
        SysCtlDelay(67000);
        GPIOPinWrite(GPIO_PORTA_BASE, GPIO_PIN_6 | GPIO_PIN_7, 0x00);
}
```

LCD Initialization - 4 bit

```
void lcddata(unsigned char cmd)
{
        unsigned char temp;      //variables used for sending higher and lower four bit
        unsigned char lcd_port;
            temp = cmd;
            temp = temp & 0xF0;
            lcd_port &= 0x0F;
            lcd_port |= temp;
            GPIOPinWrite(GPIO_PORTB_BASE,GPIO_PIN_4 | GPIO_PIN_5| GPIO_PIN_6 | GPIO_PIN_7,lcd_port); //sending higher 4 bits
            GPIOPinWrite(GPIO_PORTA_BASE, GPIO_PIN_6 | GPIO_PIN_7, 0xC0);              //RS=1, enable high to low.
            SysCtlDelay(2);
            GPIOPinWrite(GPIO_PORTA_BASE, GPIO_PIN_6 | GPIO_PIN_7, 0x40);

            cmd = cmd & 0x0F;
            cmd = cmd<<4;
            lcd_port &= 0x0F;
            lcd_port |= cmd;
            GPIOPinWrite(GPIO_PORTB_BASE,GPIO_PIN_4 | GPIO_PIN_5| GPIO_PIN_6 | GPIO_PIN_7,lcd_port);//sending lower 4 bits
            GPIOPinWrite(GPIO_PORTA_BASE, GPIO_PIN_6 | GPIO_PIN_7, 0xC0);              // RS=1 , enable high to low.
            SysCtlDelay(2);
            GPIOPinWrite(GPIO_PORTA_BASE, GPIO_PIN_6 | GPIO_PIN_7, 0x40);
            SysCtlDelay(67000);
}
```