Interfacing GLCD with TIVA Launchpad

e-Yantra Summer Internship Program

Embedded Real Time Systems Lab

July 8, 2016

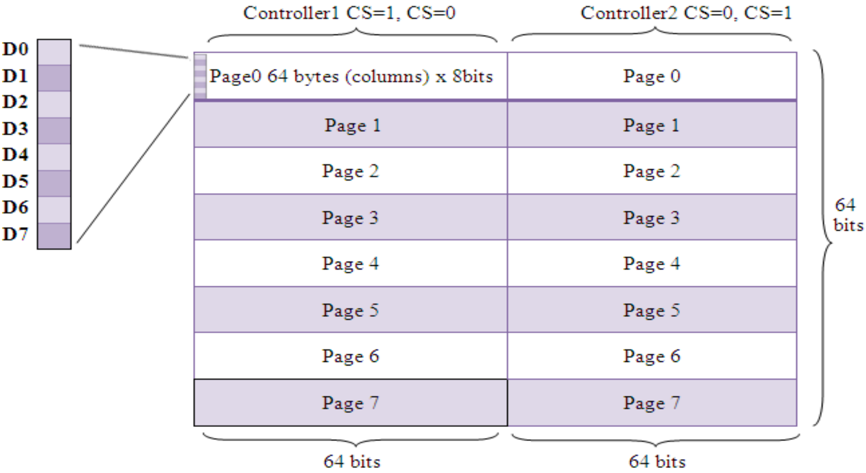
Contents

1. Introduction
   - Understanding GLCD
   - Pin Configuration
   - Data Pins
   - Control Pins
2. GLCD Programming
   - GLCD Interfacing
   - Important commands
   - Functions for using GLCD
3. Displaying image on GLCD

JHD12864E GLCD

- Based on KS107/KS108 LCD Controller
- 128 X 64 pixels
- 8 bit parallel interface
- Data Lines - 8
- Control Lines - 6
- Display is logically divided in half
- Each part is controlled using separate controller

Understanding GLCD



Controller1 CS=1, CS=0

Controller2 CS=0, CS=1

D0
D1
D2
D3
D4
D5
D6
D7

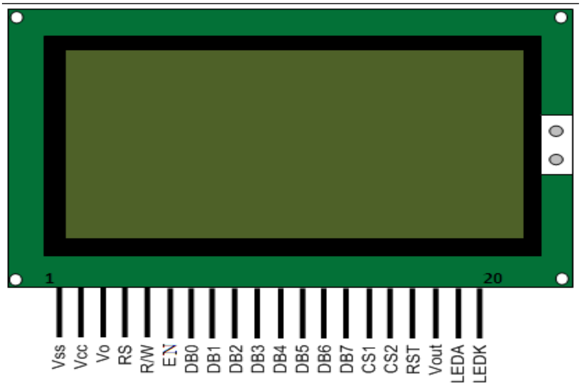
Page0 64 bytes (columns) x 8bits	Page 0
Page 1	Page 1
Page 2	Page 2
Page 3	Page 3
Page 4	Page 4
Page 5	Page 5
Page 6	Page 6
Page 7	Page 7

64 bits

64 bits

64 bits

Pin Configuration



Data Pins

Data Lines:

- There are 8 data lines(D0-D7)
- Used to send commands and data
- Data to be written on GLCD is sent in pixels(the hex values of character to be written)
  Eg. To write "A" on GLCD following data is to be sent :
  0x7E, 0x11, 0x11, 0x11, 0x7E, 0x00
- Before sending the data , page and column of GLCD is selected

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.

4. **Chip Select (CS1 and CS2)**

   CS1 = 1 : Selects the left side of GLCD

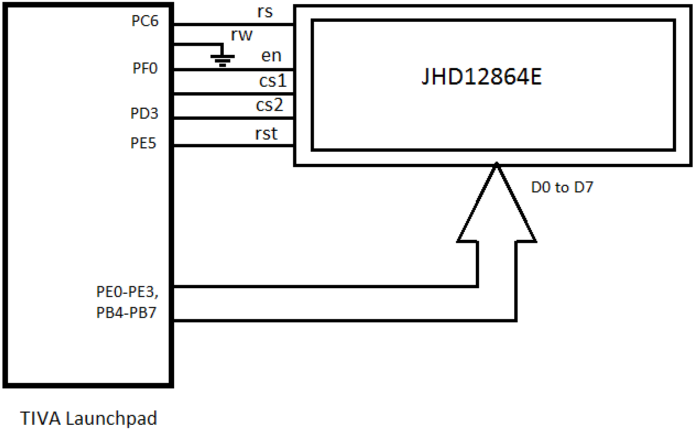
   CS2 = 1 : Selects the right side of GLCD

5. **RST**

   Resets the GLCD.

   (Active Low)

GLCD Interfacing



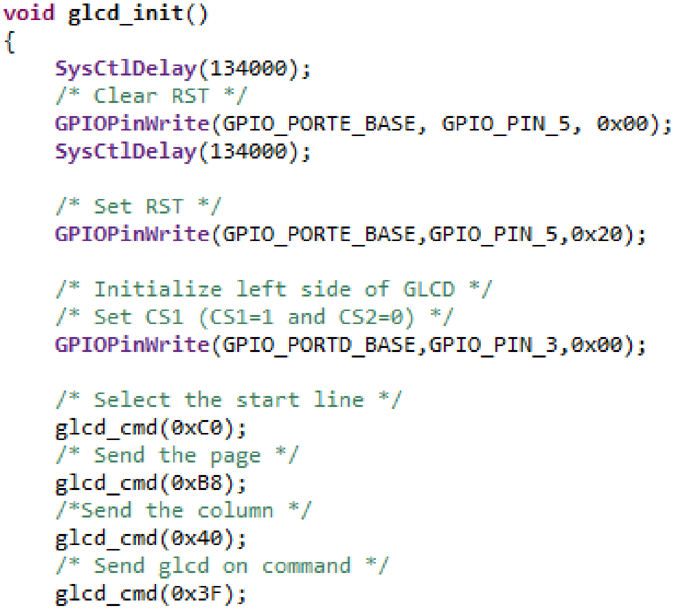
Important commands

1. To turn ON the GLCD display -
   Command = **0x3F**
2. To turn OFF GLCD display -
   Command = **0x3E**
3. To set X address (Page) -
   Command = **0xB8 — (Page number)**
4. To set Y address (Column) -
   If column number is less than 64 -
   CS1 = 1 and CS2 = 0
   Command = **0x40 — (column number)**
   If column number is greater than 64 -
   CS1 = 0 and CS2 = 1
   Command = **0x40 —(column number - 64)**

Functions for using GLCD

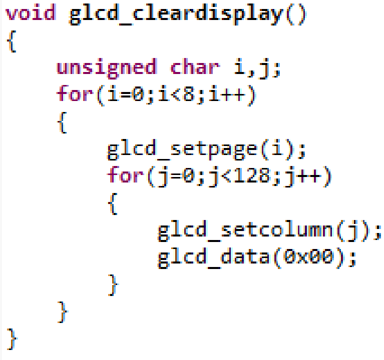
1	*glcd_init();*	(Initialize GLCD)
2	*glcd_cleardisplay();*	(Clear the GLCD display)
3	*glcd_cmd(cmd);*	(Sending commands to GLCD)
4	*glcd_data(data);*	(Sending data to GLCD)
5	*glcd_setpage(page);*	(Selecting the page or X address)
6	*glcd_setcolumn(column);*	(Selecting the column or Y address)

Function - glcd_init();



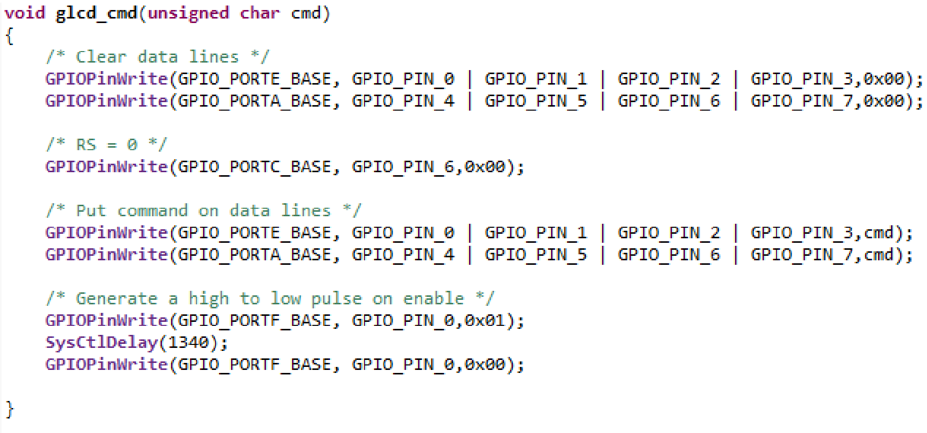
```
void glcd_init()
{
    SysCtlDelay(134000);
    /* Clear RST */
    GPIOPinWrite(GPIO_PORTE_BASE, GPIO_PIN_5, 0x00);
    SysCtlDelay(134000);

    /* Set RST */
    GPIOPinWrite(GPIO_PORTE_BASE,GPIO_PIN_5,0x20);

    /* Initialize left side of GLCD */
    /* Set CS1 (CS1=1 and CS2=0) */
    GPIOPinWrite(GPIO_PORTD_BASE,GPIO_PIN_3,0x00);

    /* Select the start line */
    glcd_cmd(0xC0);
    /* Send the page */
    glcd_cmd(0xB8);
    /*Send the column */
    glcd_cmd(0x40);
    /* Send glcd on command */
    glcd_cmd(0x3F);
```

Function - glcd_cleardisplay();



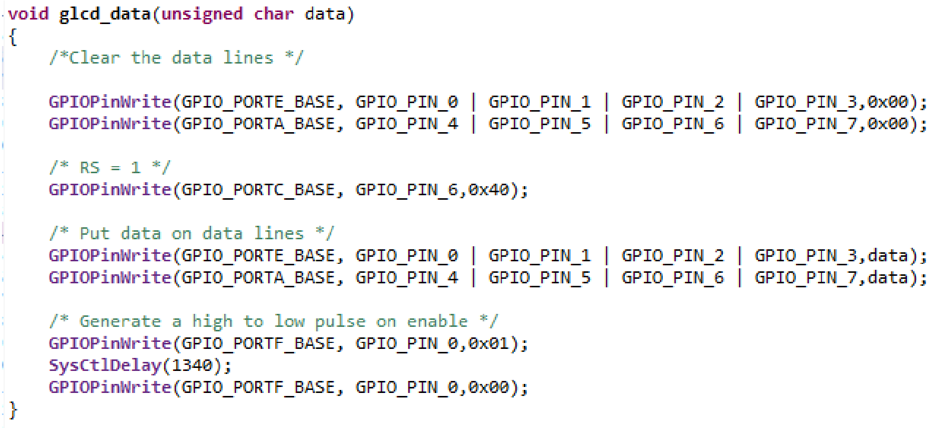
```
void glcd_cleardisplay()
{
    unsigned char i,j;
    for(i=0;i<8;i++)
    {
        glcd_setpage(i);
        for(j=0;j<128;j++)
        {
            glcd_setcolumn(j);
            glcd_data(0x00);
        }
    }
}
```

Function - glcd_cmd(cmd);



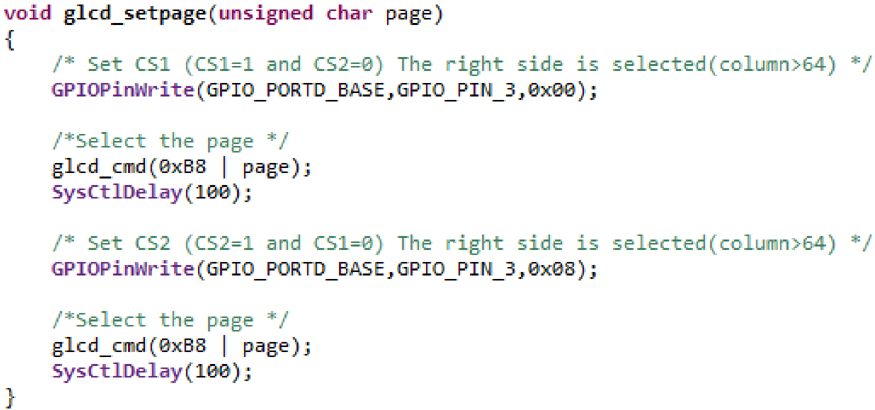
```
void glcd_cmd(unsigned char cmd)
{
    /* Clear data lines */
    GPIOPinWrite(GPIO_PORTE_BASE, GPIO_PIN_0 | GPIO_PIN_1 | GPIO_PIN_2 | GPIO_PIN_3,0x00);
    GPIOPinWrite(GPIO_PORTA_BASE, GPIO_PIN_4 | GPIO_PIN_5 | GPIO_PIN_6 | GPIO_PIN_7,0x00);

    /* RS = 0 */
    GPIOPinWrite(GPIO_PORTC_BASE, GPIO_PIN_6,0x00);

    /* Put command on data lines */
    GPIOPinWrite(GPIO_PORTE_BASE, GPIO_PIN_0 | GPIO_PIN_1 | GPIO_PIN_2 | GPIO_PIN_3,cmd);
    GPIOPinWrite(GPIO_PORTA_BASE, GPIO_PIN_4 | GPIO_PIN_5 | GPIO_PIN_6 | GPIO_PIN_7,cmd);

    /* Generate a high to low pulse on enable */
    GPIOPinWrite(GPIO_PORTF_BASE, GPIO_PIN_0,0x01);
    SysCtlDelay(1340);
    GPIOPinWrite(GPIO_PORTF_BASE, GPIO_PIN_0,0x00);

}
```

Function - glcd_data(data);



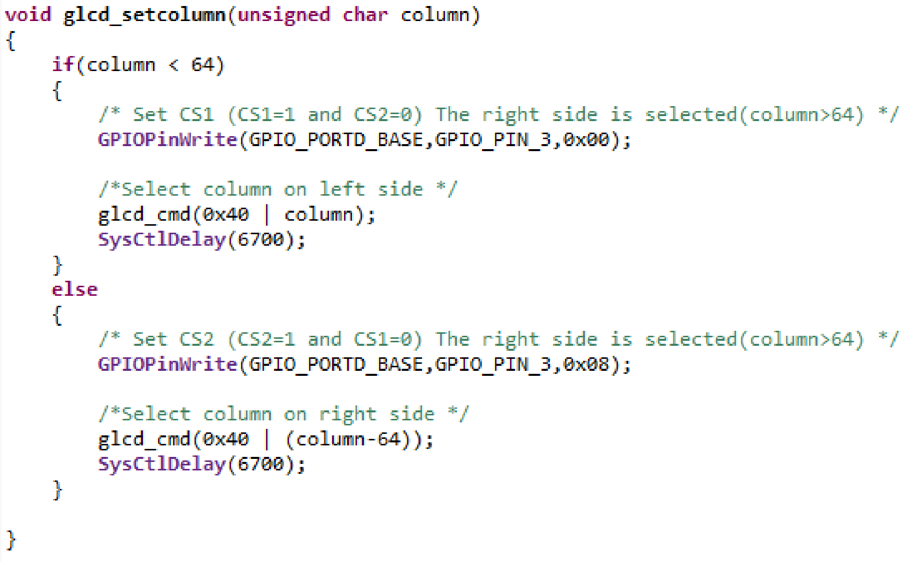
```
void glcd_data(unsigned char data)
{
    /*Clear the data lines */

    GPIOPinWrite(GPIO_PORTE_BASE, GPIO_PIN_0 | GPIO_PIN_1 | GPIO_PIN_2 | GPIO_PIN_3,0x00);
    GPIOPinWrite(GPIO_PORTA_BASE, GPIO_PIN_4 | GPIO_PIN_5 | GPIO_PIN_6 | GPIO_PIN_7,0x00);

    /* RS = 1 */
    GPIOPinWrite(GPIO_PORTC_BASE, GPIO_PIN_6,0x40);

    /* Put data on data lines */
    GPIOPinWrite(GPIO_PORTE_BASE, GPIO_PIN_0 | GPIO_PIN_1 | GPIO_PIN_2 | GPIO_PIN_3,data);
    GPIOPinWrite(GPIO_PORTA_BASE, GPIO_PIN_4 | GPIO_PIN_5 | GPIO_PIN_6 | GPIO_PIN_7,data);

    /* Generate a high to low pulse on enable */
    GPIOPinWrite(GPIO_PORTF_BASE, GPIO_PIN_0,0x01);
    SysCtlDelay(1340);
    GPIOPinWrite(GPIO_PORTF_BASE, GPIO_PIN_0,0x00);
}
```

Function - glcd_setpage(page);



```
void glcd_setpage(unsigned char page)
{
    /* Set CS1 (CS1=1 and CS2=0) The right side is selected(column>64) */
    GPIOPinWrite(GPIO_PORTD_BASE,GPIO_PIN_3,0x00);

    /*Select the page */
    glcd_cmd(0xB8 | page);
    SysCtlDelay(100);

    /* Set CS2 (CS2=1 and CS1=0) The right side is selected(column>64) */
    GPIOPinWrite(GPIO_PORTD_BASE,GPIO_PIN_3,0x08);

    /*Select the page */
    glcd_cmd(0xB8 | page);
    SysCtlDelay(100);
}
```

Function - glcd_setcolum(column);



```
void glcd_setcolumn(unsigned char column)
{
    if(column < 64)
    {
        /* Set CS1 (CS1=1 and CS2=0) The right side is selected(column>64) */
        GPIOPinWrite(GPIO_PORTD_BASE,GPIO_PIN_3,0x00);

        /*Select column on left side */
        glcd_cmd(0x40 | column);
        SysCtlDelay(6700);
    }
    else
    {
        /* Set CS2 (CS2=1 and CS1=0) The right side is selected(column>64) */
        GPIOPinWrite(GPIO_PORTD_BASE,GPIO_PIN_3,0x08);

        /*Select column on right side */
        glcd_cmd(0x40 | (column-64));
        SysCtlDelay(6700);
    }

}
```

Displaying Image on GLCD

1. The size of image to be should be 128X64 pixels
2. Convert the image in .bmp(bitmap) format
3. Generate hex values for this image
4. Create a ".h" file and store the hex values in a array
5. Include this as header file in the main program
6. Use the values from this array to display image

Note :

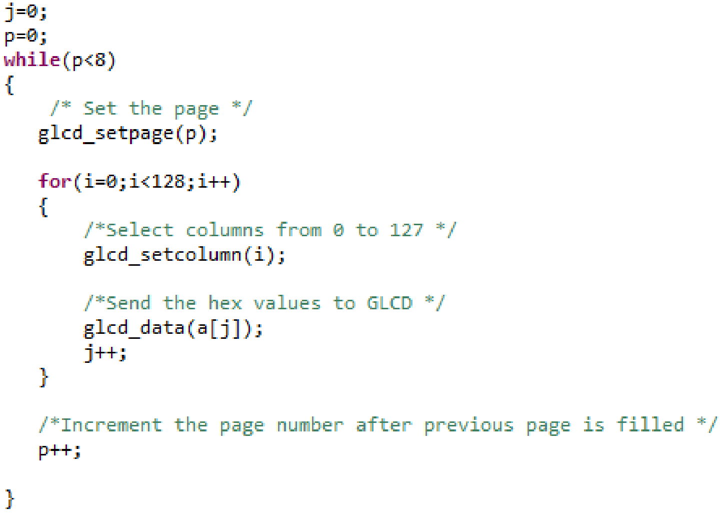
Use a black and white image(.png format is preferable)

Various softwares are available online which convert images into .bmp format

"BMP-LCD" software can be used to obtain hex values of image

Displaying Image on GLCD

Function to write values from ".h file" to GLCD



```
j=0;
p=0;
while(p<8)
{
     /* Set the page */
    glcd_setpage(p);

    for(i=0;i<128;i++)
    {
        /*Select columns from 0 to 127 */
        glcd_setcolumn(i);

        /*Send the hex values to GLCD */
        glcd_data(a[j]);
        j++;
    }

    /*Increment the page number after previous page is filled */
    p++;

}
```