

# Interfacing Graphics LCD (GLCD)

---

E.R.T.S. Lab

## 1 Lab Objective

The objective of this lab is to introduce you to interfacing 128x64 Graphics LCD.

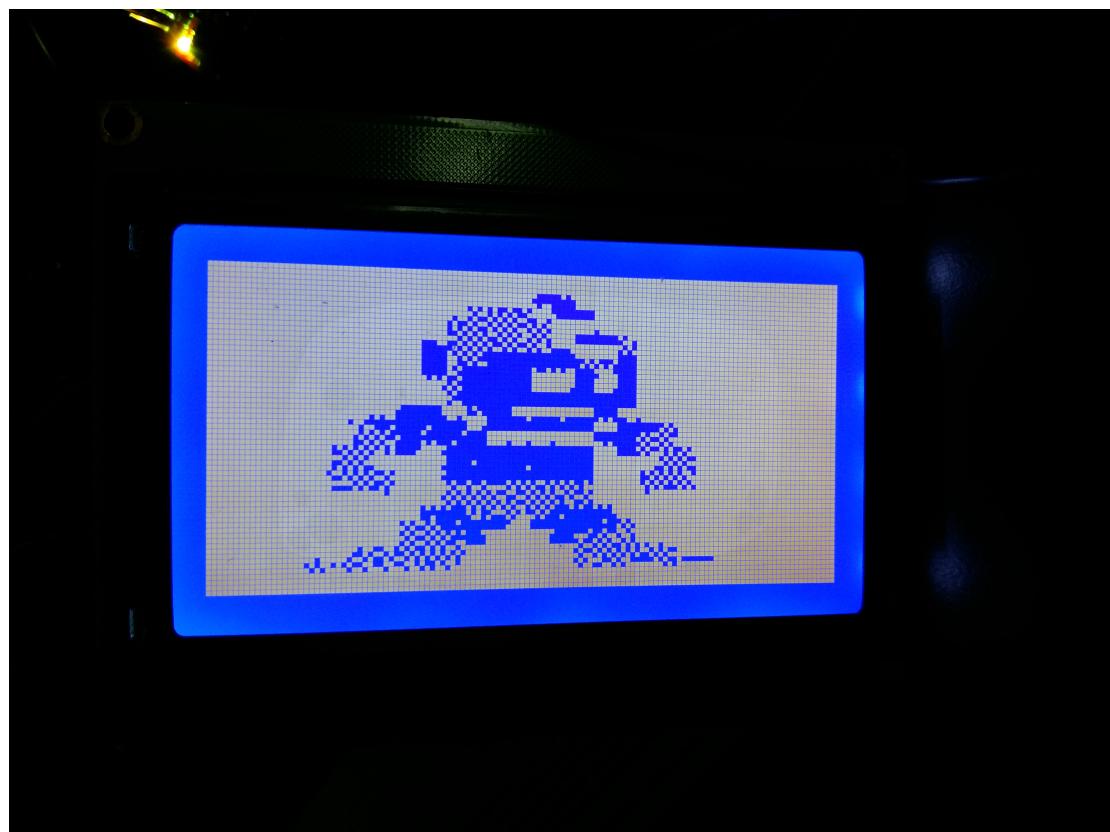
## 2 Pre-requisite

- Tiva C series based development board (Game Console) designed by E.R.T.S. Lab.
- Schematic of the development board. (Available on the course webpage in the resources section).
- Go through the video tutorial uploaded in the lab experiment video section.
- Lab 4: Interfacing joystick using ADC.

### 3 Problem Statement

#### 3.1 Displaying a static image on the GLCD

- You have been given a hexcode.h file containing an image in bitmap format with the experiment.
- The task is to display this image on the GLCD.
- Refer the procedure subsection 1 and video tutorial for completing this problem statement.
- Expected Output:



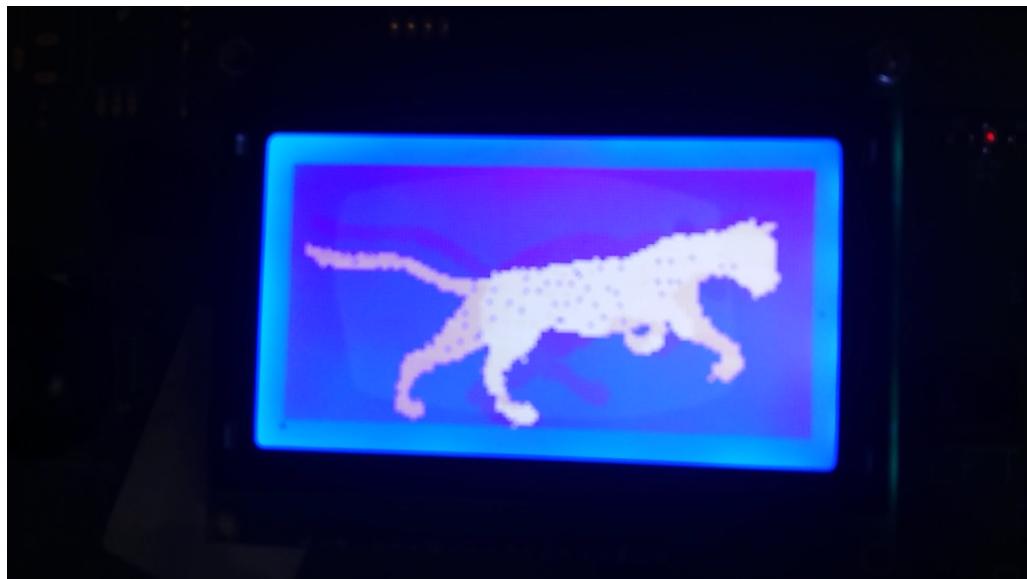
### 3.2 Using the Joystick to control an object on the GLCD

- You have interfaced the joystick in Lab 4 (Refer procedure subsection 5 for help and hints). We will use the joystick to move a square on GLCD.
- You have to move a square of 8x8 pixels on the GLCD according to the movement of the joystick (up-down, left-right).
- Refer the procedure subsection 2 for completing this problem statement.
- Expected Output:



### 3.3 Displaying a moving image on GLCD

- You have been given a set of 8 images converted to bitmap format (one.h, two.h .... eight.h) with this experiment.
- Your task is to display a moving animation using these 8 images. This can be achieved by displaying the images in sequential order from one to eight.
- Add a suitable delay between two images. Make sure that you clear the screen between two subsequent images displays.
- You can adjust the delay based on the value you get from the joystick, hence controlling the speed of the animation.
- Refer the procedure subsection 3 for completing this problem statement.
- Expected Output:



## 4 Pin Mapping

Use the following table for quick reference of pin connections.

Sr No.	Pin Number	Connected to
1.	PC7	SW1
2	PD6	SW2
3.	PD7	SW3
4.	PF4	SW4
5.	PF1	LED1
6.	PC5	LED2
7.	PB2	LED3
8.	PF3	LED3
9.	PD0	V of Joystick
10.	PD1	H of Joystick
11.	PE0 to PE3	D0 to D3 of LCD and GLCD
12.	PB4 to PB7	D4 to D7 of LCD and GLCD
13.	PC6	RS for LCD and GLCD
14.	PF0	Enable of LCD and GLCD
15.	PD3	CS2 of GLCD
16.	Not PD3	CS1 of GLCD
17.	PE5	RST of GLCD

Table 1: Pin mapping

## 5 Procedure

### 5.1 Programming 128X64 GLCD

1. To display image on GLCD the hex values are calculated. These values are stored in a array in a header file. This header file is included in main program. The array elements can be accessed from the program and written on GLCD data lines.
2. Include all the necessary header files.
3. Enable the System Clock PORT B, PORT D, PORT E, PORT F, PORT C.
4. Refer the section *Assembly of hardware* for information regarding connection of PORT pins with GLCD.
5. All the PORT pins are configured as output pins.
6. Initialize and clear the GLCD.
7. To set the GLCD page, page number is logically ORed with 0xB8 and the result is sent to GLCD as GLCD command.
8. To set the GLCD column, column number is logically ORed with 0x40 and the result is sent to GLCD as GLCD command.

9. To display a line on GLCD, select any page number from 0 to 7, select columns from 0 to 127 one after other and write data 0xFF to GLCD.

## 5.2 To display movement of Joystick on GLCD

1. Enable and configure System Clock.
2. Configure PORT B, PORT C, PORT D (PD3), PORT E and PORT F as output PORTs.
3. Configure PD0 and PD1 as analog input.(ADC inputs)
4. Configure the ADC.
5. Initialize and clear GLCD.
6. Select a square of 8X8 pixels which will show the position of Joystick on GLCD.
7. To obtain page number the digital value obtained is divided by 512 and for column number the digital value is divided by 32.
8. This value is then sent to GLCD and the movement of Joystick can be observed on GLCD.

## 5.3 To control the speed of image on GLCD

1. Enable and configure System Clock.
2. Include the header files which contain hex values of images to be displayed.
3. Configure PORT B, PORT C, PORT D (PD3), PORT E and PORT F as output PORTs.
4. Configure PD0 and PD1 as analog input.(ADC inputs)
5. Configure the ADC.
6. Initialize and clear GLCD.
7. Read the analog data from PD0 and PD1.
8. If the digital data obtained is less than 200 then the delay in between display of two consecutive images is increased.
9. If the digital data obtained is more than 3900 then the delay in between display of two consecutive images is decreased.

#### **5.4 (Optional) To program push buttons and LEDs**

1. Include all the required header files

```
#include <stdint.h>
#include <stdbool.h>
#include "inc/hw_types.h"
#include "inc/hw_memmap.h"
#include "driverlib/sysctl.h"
#include "driverlib/gpio.h"
#include "inc/hw_gpio.h"
#include "driverlib/rom.h"
```

2. Enable the System Clock, PORT B, PORT C, PORT D, PORT F. (*Pin mapping table* gives details about the PORT pins used for different peripherals)
3. Configure the port pins connected to LEDs as output and those connected to push buttons as input.
4. Unlock pin PF0.
5. Read the status of switch and turn on the corresponding LED if switch press is detected.
6. Switch debouncing can also be added if required.

#### **5.5 (Help and Hints to Lab 4) Programming Joystick using ADC**

1. Include all the required header files. Ensure that the following header file is included.

```
#include "driverlib/adc.h"
```

2. Enable the System Clock and PORT D.
3. PD0 and PD1 are ADC channels CH7 and CH6 respectively. Configure these pins as ADC input.
4. Configure ADC Sequencer as per requirement.
5. Enable the ADC.
6. Store the ADC converted data in a variable.
7. This data can be divided by a suitable value.
8. The final result can be displayed in Watch window, on Computer using UART. This movement of Joystick can also be displayed on GLCD.

### **6 Demo and Submissions**

Show the output of the problem statement 1,2 and 3 to your TA. Make sure that you have all the lab experiments on Github. You will be asked to the final submissions on Github. Also your project codes must be uploaded on Github, so we recommend you to use Github for working on your project.