Lab 12: LCD and Keypad Interfacing

Name: Syed Muhammad Ghufran Alvi ID: sa08235 Section: L-1

Objective

To introduce 3x4 Keypad and 16x2 LCD interfacing using TM4C123 TivaC.

In-Lab

Task 1: Interface 3x4 Keypad and 16x2 LCD together.
Task 2: Design a Password (pin) based Safe Lock.
Task 2: Design a Calculator.

1 Introduction to LCD

1.1 16x2 Liquid Crystal Display (LCD)

16×2 LCD is named so because; it has 16 Columns and 2 Rows. There are a lot of combinations available like, 8×1, 8×2, 10×2, 16×1, etc. But the most used one is the 16*2 LCD. This type of LCD can display 32 ASCII characters. It consists of 2 rows and 16 column. Each row and column intersection can display one ASCII character. Hence, the position of each character is defined in terms of rows and columns order pairs (x,y). For example, (0,0) means the first row and first column, and (1,15) means the second row and 16th column and so on.

1.2 16×2 LCD Pin Configuration

It consists of 16 pins such as data, control, power supply, and backlight LED pins as shown in Fig. 1, description for each given below.

Data Pins : Pins D0 to D7 are data pins which are used to send data to be displayed or commands to LCD controller. Hence, these lines will be connected with TM4C123 GPIO pins to transfer data. But this LCD can be used either in 4-bit (only D0 to D3 pins are used) or 8-bit mode(all D0 to D7 pins are used).

Read/Write (R/W) : As its name suggests, R/W is used to select read or write mode of LCD. When this pin is set to active high, LCD will be in read mode which means we can read data from the LCD. Similarly, when this pin is set to active low, we will be able to write data

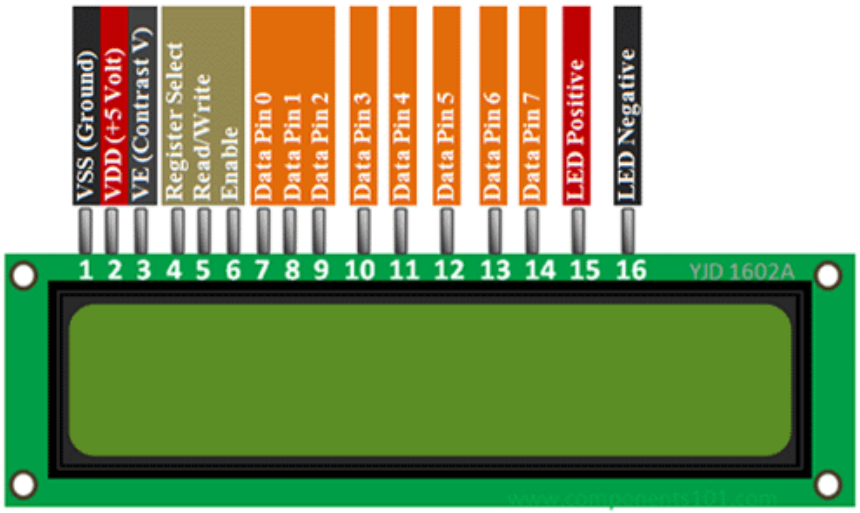


Figure 1: 16x2 LCD Pin-out

to the LCD. We will connect this pin to the ground. Because we are using a 16×2 LCD as an output device only.

Enable (E) : This pin is used to enable and disable LCD. When this pin is active low, LCD controller will be disabled. That means control pins and data pins will not have any effect on the display. On the other hand, when the enable pin is set to active high, the LCD will work normally and process all data and control instructions.

Backlight LED Pins : These pins are cathode and anode pins of back light LED. They are used to provide +5 volts and ground to anode and cathode pins.

Power Supply Pins VDD : +5 volt power supply pins.

Control Pins :

LCD Contrast (Vo) : It is used to adjust the contrast of LCD with respect to text display. This contrast can be set through by using a potential divider circuit with a variable resistor.

Register Select (RS) : With the help of this pin, TM4C123 microcontroller informs the LCD controller either we are sending commands or data to LCD. In other words, it helps to differentiate between data and commands. For example, when we want to send commands to LCD from TM4C123 microcontroller, we set this pin active low by sending an active signal from the GPIO pin of Tiva Launchpad. These commands are setting cursor position, cursor on or off, scrolling text left or right, clearing text from LCD, etc. On the contrary, when we want to send data to LCD, we provide active high signal to this pin from TM4C123 microcontroller

1.3 Modes of 16×2 LCD

16x2 LCDs can be interfaced with TM4C123 Tiva Launchpad either in **4-bit mode** or **8-bit mode**.
For **4-bit interface** we need to use 6 GPIO pins of Launchpad. In 4-bit mode, data transfers from microcontroller to the LCD in two consecutive half bytes.
For **8-bit interface** we need to use 10 GPIO pins of Launchpad. one byte data transfers to the LCD in one go.
To save GPIO pins, we will use 4-bit mode of 16×2 LCD. Because, displaying data on LCD is not a time-critical action. Hence. 4-bit use will save us 4 GPIO pins of the Tiva launchpad and reduce complexity of circuit. Table in Fig. 4 summarizes the commands to control operation of LCD, but since we won't be using Register level programming in this lab, library will take care of these commands.

Command	Function
0x01	Clear display screen
0x02	Return cursor to home i.e starting position
0x06	Shift the cursor one position right
0x0F	Turn on LCD display
0x80	Takes cursor to beginning of first row
0xC0	Takes cursor to beginning of Second row
0x28	Select 4-Bit Mode of LCD
0x38	Select 8-Bit Mode of LCD

Figure 2: 16×2 LCD Commands

2 Introduction to Keypad

2.1 3x4 Keypad

3x4 (12-button) keypad provides a useful human interface component for microcontroller projects. The keys are connected into a matrix, so you only need 7 microcontroller pins (3-columns and 4-rows) to scan through the pad. Beneath each key is a membrane switch. Each switch in a row is connected to the other switches in the row by a conductive trace underneath the pad. Each switch in a column is connected the same way – one side of the switch is connected to all of the other switches in that column by a conductive trace. Pressing a button closes the switch between a column and a row trace, allowing current to flow between a column pin and a row pin.

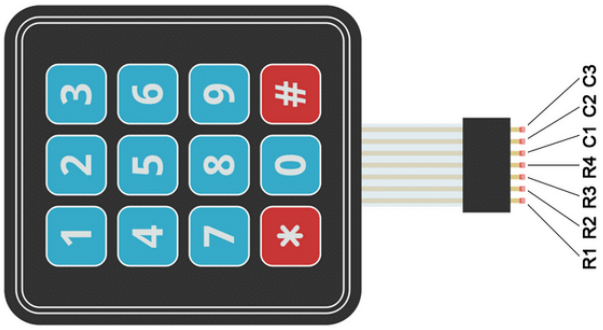


Figure 3: Pinout of 4×3 Membrane Keypad

2.2 Keypad Working and Scanning

The working principle is very simple. Pressing a button shorts one of the row lines to one of the column lines, allowing current to flow between them. For example, when key '1' is pressed, column 1 and row 1 are shorted as shown in Fig. 4.

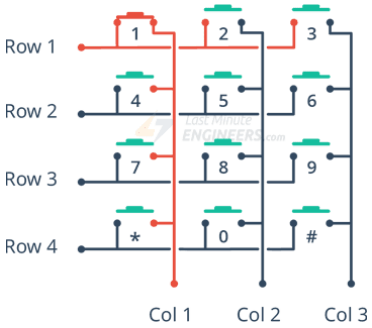


Figure 4: Working of 4×3 Membrane Keypad

A microcontroller can scan these lines for a button-pressed state. To do this, it follows below procedure.

1. Microcontroller sets all the column and row lines to input.
2. Then, it picks a row and sets it HIGH.
3. After that, it checks the column lines one at a time.
4. If the column connection stays LOW, the button on the row has not been pressed.
5. If it goes HIGH, the microcontroller knows which row was set HIGH, and which column was detected HIGH when checked.
6. Finally, it knows which button was pressed that corresponds to detected row and column.

3 TivaC LaunchPad with Energia

In Energia IDE, unlike Kiel uVision, we can use TivaC LaunchPad pins for various peripherals without the need to activate Ports using registers or specifying function of the pin. But it also comes with limitations of usability and programmable scope of the board. In Energia, we can refer to Pins of TivaC directly using numeric digit like 1,2,3.. and so on. Pin map for the EK-TM4C123GXL LaunchPad is given in Fig. 5 with Black Columns under J1, J2, J3 and J4.

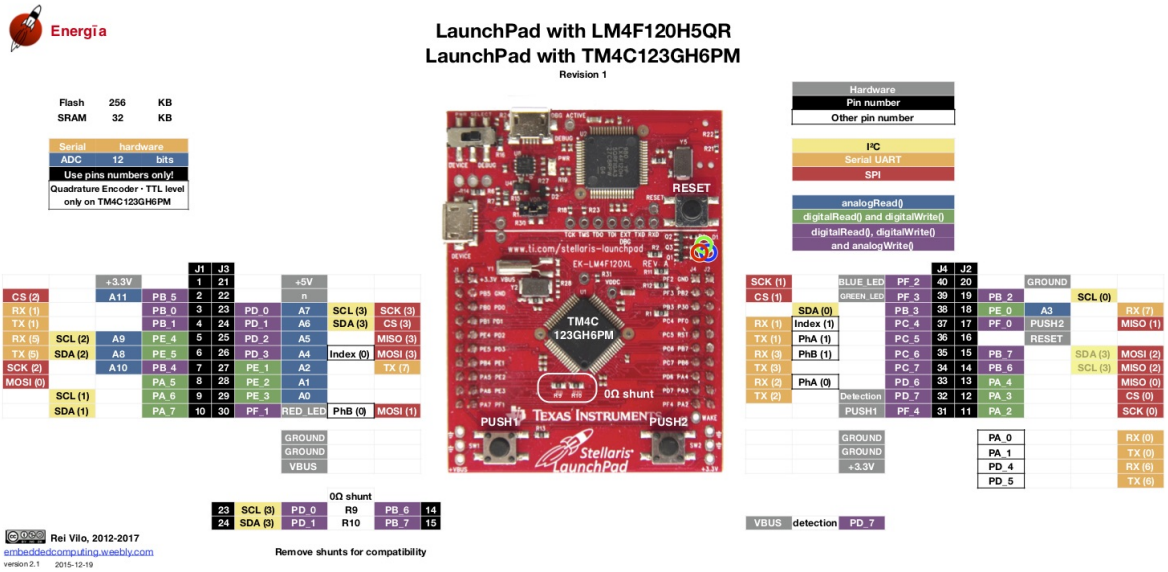


Figure 5: TivaC Pin Distribution with Energia

In-Lab Tasks

Task 1: Interface 3x4 Keypad and 16x2 LCD together

In this task, we will interface the LCD and Keypad with TivaC with connections as shown in Fig. 6. Make connections of circuit, download and upload the code ”lcd_ keypad.ino” from LMS on TivaC using Energia. If connections are correct and code is uploaded you should be able enter any number with Keypad and able to observe it on LCD. You are not required to modify the code in this task but only observe the Keypad Input and LCD Output. Get circuit interfaced connections checked with RA.

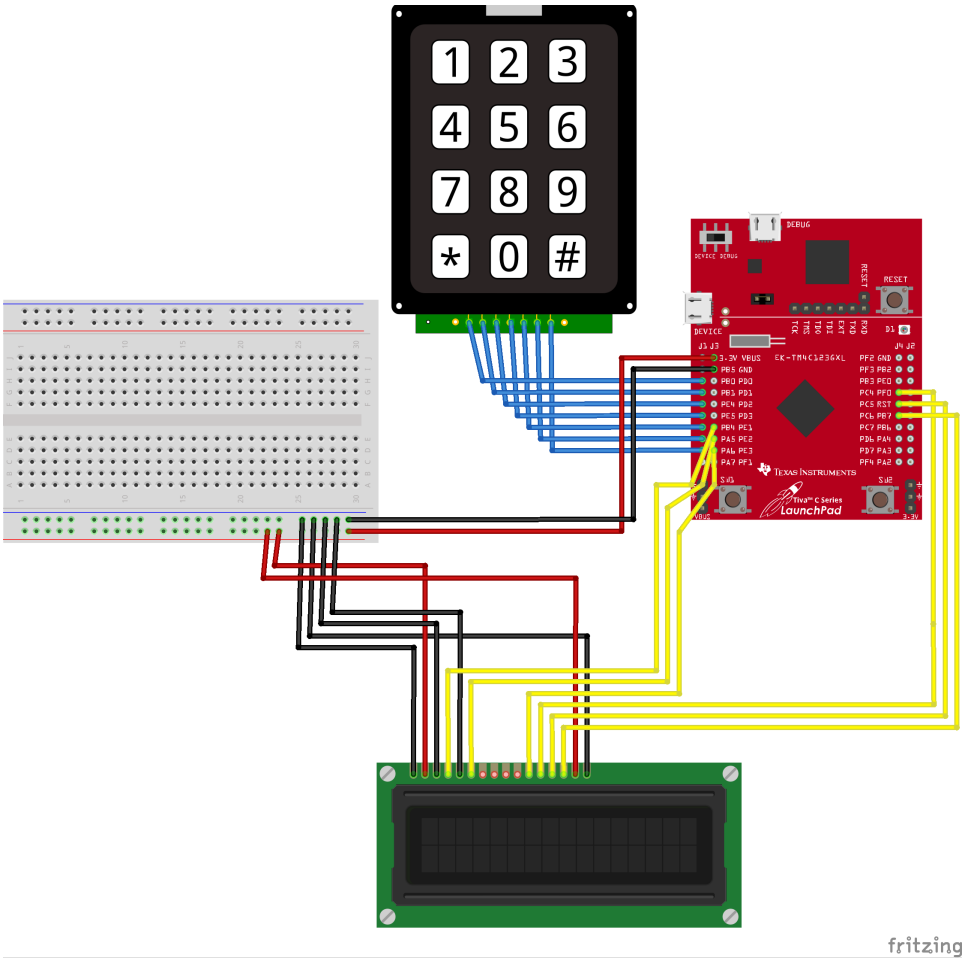
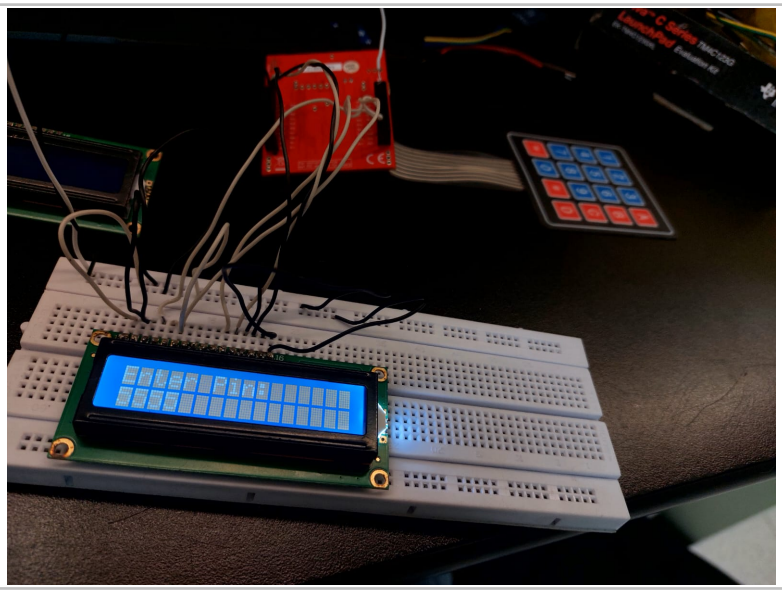


Figure 6: Circuit Connections for LCD and Keypad with TivaC

Provide your clear circuit image below.



Task 2: Design a Password (pin) based Safe Lock

In this task, we will modify the "lcd_ keypad.ino" code to design a Safe Lock with a password. The safe lock will have following functionality:

1 LCD will display "Safe Lock" every time board restarts.

2 Next, LCD will display "Enter Pin:" in the first row.

3 The input from Keypad will be displayed in the second row.

4 The Pass Code should be your 5 digit student ID i.e. 05201.

5 If the Pass Code entered is invalid, the built-in RED LED will blink once and LCD will display "Invalid" and you will return to step 2 above.

6 If the Pass Code entered is valid, then build-in BLUE LED will blink once and LCD will display "Valid" and you will return to step 2 above.

In order to achieve the above functionality of Safe Lock, we will use the libraries "LiquidCrystal.h" and "Keypad.h", whose built-in functions helps us select data and commands to be displayed easily on LCD and take inputs from Keypad easily. You can download "Keypad-master.zip" from LMS and add it to Energia by clicking "Sketch" and then "Include Library".

Go through each line of code and read comments to understand how the library functions are working, modify the code where you are required to obtain the functionality for Safe Lock above.

Provide your code here with appropriate comments below (Get the circuit demonstration checked with RA within Lab)

```
#include <LiquidCrystal.h>    // Library for LCD Display
#include <Keypad.h>           // Library for Keypad

// Initializing LCD Pins with object "lcd"
LiquidCrystal lcd(27, 28, 29, 37, 36, 35);  // RS,EN,D4,D5,D6,D7 at PE1,PE2,PE3,PC4,PC5,PC6

// Built-in LED Indicators for answers
const int RED_led = 30; // At PF1
const int BLUE_led = 40; // At PF2

////////////////////////////////////////////////////////////////////////
// Defining the Keymap layout
const byte ROWS = 4;      // Four rows
const byte COLS = 3;      // Three columns
char keys[ROWS][COLS] = {
  {'1','2','3'},
  {'4','5','6'},
  {'7','8','9'},
  {'*','0','#'}
};

// Connect keypad ROW0, ROW1, ROW2 and ROW3 to these TivaC pins PB0, PB1, PE4, PE5
byte rowPins[ROWS] = { 3, 4, 5, 6 };
// Connect keypad COL0, COL1 and COL2 to these TivaC pins PB4, PA5, PA6
byte colPins[COLS] = { 7, 8, 9 };

// Create the Keypad object
Keypad customKeypad = Keypad(makeKeymap(keys), rowPins, colPins, ROWS, COLS);
#define pin_length 5              // Passcode/pin length+1
char Data[pin_length];            // Creating array of size of the passcode
char Master[pin_length] = "5522";    // Enter your student ID here
byte data_count = 0, master_count = 0;
bool Pass_is_good;            // flag to detect if password is correct
char customKey;               // stores the character entered by user on keypad
////////////////////////////////////////////////////////////////////////

void setup() {
   pinMode(27, OUTPUT); // Pins 8-14 are enabled as output for LCD
   pinMode(28, OUTPUT);
   pinMode(29, OUTPUT);
   pinMode(37, OUTPUT);
   pinMode(36, OUTPUT);
   pinMode(35, OUTPUT);

   pinMode(RED_led, OUTPUT);
   pinMode(BLUE_led, OUTPUT);

   // Initialize data to be printed on LCD whenever program starts
   lcd.begin(16, 2);              // Initializing LCD
```

Task 3: Design a Calculator

In this task, you are required to create a simple calculator that should be able to perform the following tasks:

1 LCD will display "Calculator" every time board restarts.

2 You will take first input as "First Number: ".

3 You will take second input as "Operation: " which can be either "$*$" to perform multiplication or "# " to perform division.

4 You will take third input as "Second Number: "

5 To display output result, user should press "$*$" or "# " again.

6 If numbers or operations are invalid, then you should display "Invalid" and return to step 1 above.

Provide your code here with appropriate comments below (Get the circuit demonstration checked with RA within Lab)

```
#include <LiquidCrystal.h>    // Library for LCD Display
#include <Keypad.h>           // Library for Keypad

// Initializing LCD Pins with object "lcd"
LiquidCrystal lcd(27, 28, 29, 37, 36, 35);  // RS,EN,D4,D5,D6,D7 at PE1,PE2,PE3,PC4,PC5,PC6

// Built-in LED Indicators for answers
const int RED_led = 30; // At PF1
const int BLUE_led = 40; // At PF2

//////////////////////////////////////////////////////////////////////////
// Defining the Keymap layout
const byte ROWS = 4;      // Four rows
const byte COLS = 3;      // Three columns
char keys[ROWS][COLS] = {
  {'1','2','3'},
  {'4','5','6'},
  {'7','8','9'},
  {'*','0','#'}
};

// Connect keypad ROW0, ROW1, ROW2 and ROW3 to these TivaC pins PB0, PB1, PE4, PE5
byte rowPins[ROWS] = { 3, 4, 5, 6 };
// Connect keypad COL0, COL1 and COL2 to these TivaC pins PB4, PA5, PA6
byte colPins[COLS] = { 7, 8, 9 };

// Create the Keypad object
Keypad customKeypad = Keypad(makeKeymap(keys), rowPins, colPins, ROWS, COLS);

//////////////////////////////////////////////////////////////////////////

void setup() {
    pinMode(RED_led, OUTPUT);
    pinMode(BLUE_led, OUTPUT);

    // Initialize data to be printed on LCD whenever program starts
    lcd.begin(16, 2);              // Initializing LCD
    lcd.print("Calculator");       // Prints in the first row
    delay(3000);                   // Left the data written on LCD for 3 seconds
    lcd.clear();                   // Clears everything on the LCD
}

void loop() {
  calculator();  // Function to execute calculator logic
}

void calculator() {
  String input;
  char customKey;
```

4 Assessment Rubrics

Marks distribution

		LR1	LR2	LR9
In-lab	Task 1	20 points	-	20 points
	Task 2	-	30 points	
	Task 3	-	30 points	
Total Marks 100				

Marks obtained

		LR1	LR2	LR9
In-lab	Task 1		-	
	Task 2	-		
	Task 3	-		
Total Marks 100				