EECE 4038C Embedded System Design
Laboratory Report

Lab 8

LCD 16x2 and 4x4 Keypad

Presented
By

Group #19
Will Tekulve - M08786036
Steven Campbell - M10442056

Fall 2019

University of Cincinnati
Electrical Engineering and Computer Science

Objectives and Problem Description

The objective of this lab is to use the knowledge we have learned from previous labs to program and build a circuit to operate a LCD 16x2 with a 4x4 keypad function.

Procedure

A. Display names displayed on LCD display
    a. Connect 16x2 LCD to the pins of available port on Curiosity board. (either 8-bit or 4-bit mode).
    b. Write a program code in C language to display your full names (all team members) on the LCD. One name per line.

B. Keypad entry
    a. With the LCD kept connected, connect the 4x4 keypad to Curiosity board, use the available ports and pins.
    b. Write a program code in C language to display the depressed key numbers (0 ~ 9) and characters (A, B, C, *, and #) on the 16x2 LCD. The key "D" will be used to clear the LCD display.
    c. The numbers and characters entered need to be displayed in sequence, one after another (not to override).

You may use any of the available I/O ports and pins for these tasks.
Remember all the precautions discussed in the class and the current source/sink limitations of the microcontroller.
Include photographs of your circuit setups in your report. Submit the PBASE programs as separate files.

Expected Results

The expected results of this lab is for us to be able to successfully display both of our names at the same time on separate lines using the LCD display. We also are expecting to be able to use the 4x4 keypad to be able to clear the LCD display and be able to write the number 1-9 and the letters A-C. D should be used to clear the display.

Experiment and Design Revisions

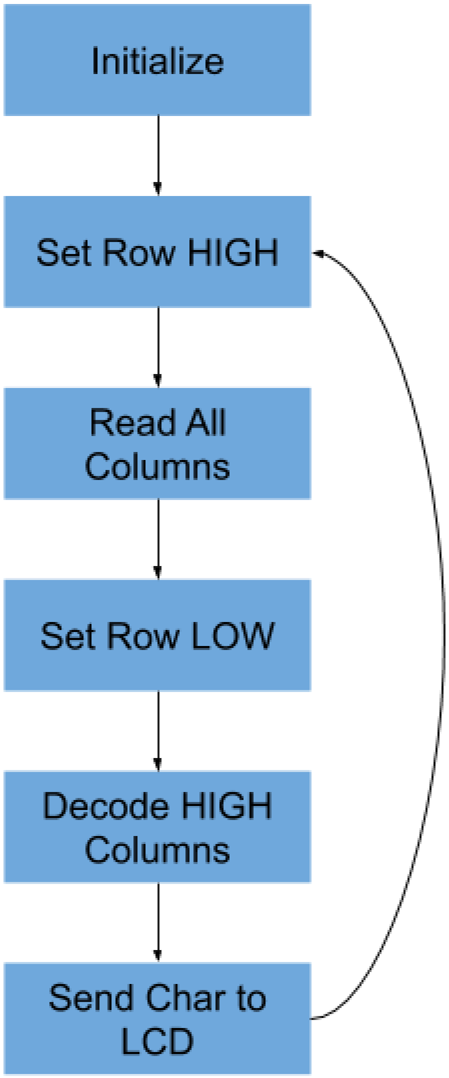

Figure 1: Lab 8 State Machine

We spent most of our time in lab attempting to program the LCD display using 4-bit mode, but due to the lack of information and resources available, we resorted to programing the LCD display in 8-bit mode because it is a simpler method of communication to the LCD and we were given more information on how to program the display using 8-bit mode. This allowed us to successfully complete the first part of the lab, but we ran into a capacity problem in adding the 4x4 keypad. Using 8-bit mode meant that we needed to use all available pins on the curiosity board, the only problem is that pin (add pin number here) cannot be overridden to be an input or output pin. It is permanently set to be a reset pin, meaning anything signal to it will reset the board. After talking to the TA, we decided to only use a 3x4 keypad instead. This will still allow the keypad to work, we just will not be able to use all of the available buttons instead.

Observations

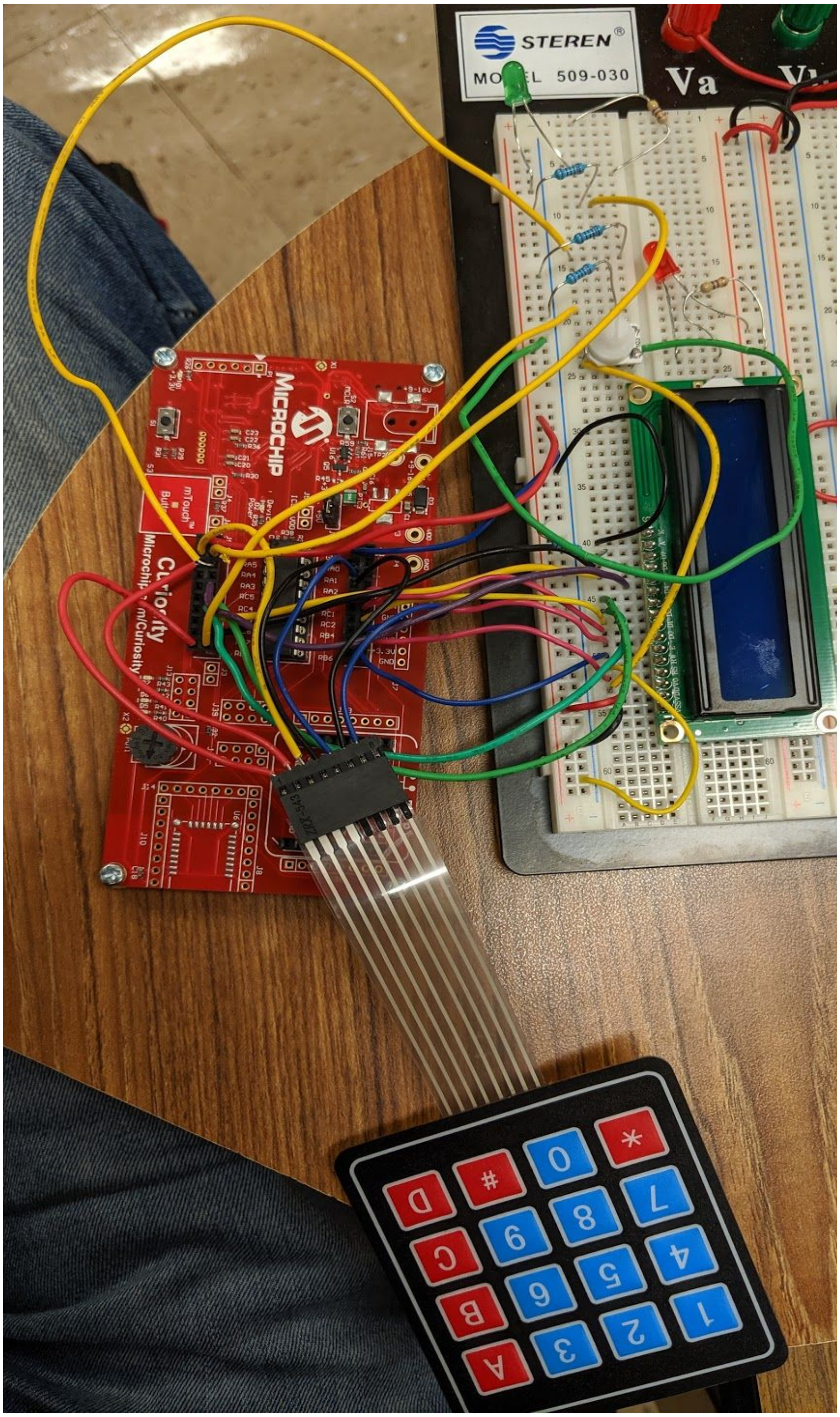

Figure 2: Completed Lab Circuit

The circuit we created during lab was capable of fully implementing the lab procedure but ended up not performing to expectations. We originally tried to implement LCD communications in 4 bit mode but failed to see any recognizable results on the screen. Switching to 8 bit mode allowed us to successfully display characters. Due to using 8 bit mode, we were required to use all 18 available GPIO pins on the PIC16F18346 microcontroller to control the 4x4 touchpad matrix. However, one of those pins is permanently locked to master clear and we saw our MCU reset every time the corresponding touchpad button was pressed. This forced us to reduce the 4x4 matrix to 3x3. We then discovered problems with floating voltage on the matrix column input pins and had to add 10k pull-down resistors on all column inputs to ensure we got accurate readings. After that problem was solved, we realized that our MCU had stopped outputting HIGH across the bottom row of the touchpad matrix, this is likely indicating a broken pin on the MCU and prevented us from being able to correctly demonstrate functionality.

Discussion

Steven Campbell

Overall we were able to overcome some physical problems we were having with the lab and were able to achieve the goal of the lab with some exceptions. The main thing to take away from the results were that we were able to complete the experiment as instructed but were limited to issues of the keypad not working and not having enough pins on the circuit board. In other words, if all the parts were working then we would have had no issue with the lab.

Will Tekulve

Overall, the lab was (mostly) successful. While we had issues with the components we received (shorts across the touchpad, non-working pins on the MCU), we were still able to demonstrate the core functionality expected during this lab. Getting the LCD screen to work in 4-bit mode was tricky and not documented well, 8 bit mode was much simpler to use even though it required more GPIO pins. The code to control the touchpad matrix is fairly straightforward, but we had to add additional circuitry components (pull-down resistors) to get accurate readings.

Programs

Main.c (all other files included in .zip)

```
/**
  Generated Main Source File

  Company:
    Microchip Technology Inc.

  File Name:
    main.c

  Summary:
    This is the main file generated using PIC10 / PIC12 / PIC16 / PIC18
MCUs

  Description:
    This header file provides implementations for driver APIs for all
modules selected in the GUI.
    Generation Information :
        Product Revision  :  PIC10 / PIC12 / PIC16 / PIC18 MCUs - 1.77
        Device            :  PIC16F18346
        Driver Version    :  2.00
*/

/*
    (c) 2018 Microchip Technology Inc. and its subsidiaries.

    Subject to your compliance with these terms, you may use Microchip
software and any
    derivatives exclusively with Microchip products. It is your
responsibility to comply with third party
    license terms applicable to your use of third party software (including
open source software) that
    may accompany Microchip software.

    THIS SOFTWARE IS SUPPLIED BY MICROCHIP "AS IS". NO WARRANTIES, WHETHER
    EXPRESS, IMPLIED OR STATUTORY, APPLY TO THIS SOFTWARE, INCLUDING ANY
    IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, AND FITNESS
    FOR A PARTICULAR PURPOSE.

    IN NO EVENT WILL MICROCHIP BE LIABLE FOR ANY INDIRECT, SPECIAL,
PUNITIVE,
    INCIDENTAL OR CONSEQUENTIAL LOSS, DAMAGE, COST OR EXPENSE OF ANY KIND
    WHATSOEVER RELATED TO THE SOFTWARE, HOWEVER CAUSED, EVEN IF MICROCHIP
    HAS BEEN ADVISED OF THE POSSIBILITY OR THE DAMAGES ARE FORESEEABLE. TO
    THE FULLEST EXTENT ALLOWED BY LAW, MICROCHIP'S TOTAL LIABILITY ON ALL
    CLAIMS IN ANY WAY RELATED TO THIS SOFTWARE WILL NOT EXCEED THE AMOUNT
    OF FEES, IF ANY, THAT YOU HAVE PAID DIRECTLY TO MICROCHIP FOR THIS
```

```c
    SOFTWARE.
*/

#include "mcc_generated_files/mcc.h"
#include "lcd.h"
#include <string.h>

const char* NAME_WT = "Will Tekulve";
const char* NAME_SC = "Steven Campbell";

/*
 * Touchpad PIN8 = NA  // ROW 0
 *          PIN7 = RB4 // ROW 1
 *          PIN6 = RA2 // ROW 2
 *          PIN5 = RA1 // ROW 3
 *          PIN4 = RA0 // COL 0
 *          PIN3 = RA5 // COL 1
 *          PIN2 = RA4 // COL 2
 *          PIN1 = RB7 // COL 3
 */

const char* TOUCHPAD[4][4] = {
    {"0", "0", "0", "0"},
    {"4", "5", "6", "B"},
    {"7", "8", "9", "C"},
    {"*", "0", "#", "D"}
};

/*
                         Main application
 */

void main(void)
{
    // initialize the device
    SYSTEM_Initialize();

    // When using interrupts, you need to set the Global and Peripheral Interrupt Enable bits
    // Use the following macros to:

    // Enable the Global Interrupts
    //INTERRUPT_GlobalInterruptEnable();

    // Enable the Peripheral Interrupts
    //INTERRUPT_PeripheralInterruptEnable();

    // Disable the Global Interrupts
    //INTERRUPT_GlobalInterruptDisable();

    // Disable the Peripheral Interrupts
    //INTERRUPT_PeripheralInterruptDisable();

    LCD_Initialize();

    LCDPutStr(NAME_WT);
    LCDLine2();
    LCDPutStr(NAME_SC);

    __delay_ms(2000);
    DisplayClr();

    char activeCols = 0;
    int currentRow = 0;

    while (1)
    {
        for (int i = 1; i < 4; i++) {
            switch (i) {
                case 1:
                    ROW1_SetHigh();
                case 2:
                    ROW2_SetHigh();
                case 3:
                    ROW3_SetHigh();
            }
            __delay_ms(1000);

            activeCols |= (1 & COL0_GetValue()) << 0;
            activeCols |= (1 & COL1_GetValue()) << 1;
            activeCols |= (1 & COL2_GetValue()) << 2;
            activeCols |= (1 & COL3_GetValue()) << 3;

            if (COL0_GetValue() == HIGH) {
                LCDPutStr("4");
            }

            switch (i) {
                case 1:
                    ROW1_SetLow();
                case 2:
                    ROW2_SetLow();
                case 3:
                    ROW3_SetLow();
            }
            for (int j = 0; j < 4; j++) {
                if (activeCols & (1 << j)) {
                    if (i == 3 && j == 3) {
                        DisplayClr();
                    } else {
                        char* val = TOUCHPAD[i][j];
                        LCDPutStr(val);
                    }
                }
            }
            activeCols = 0;
            __delay_ms(1000);
        }
    }
}

/**
 End of File
*/
```