Memo to: Mr. Cory Mettler
From: Julian Rechsteiner, Bennett Christenson
Date: March 11, 2021
Regarding: EELE 465, Lab 4 – LED and LCD communication via I2C

Summary:

The purpose of this lab was to use an MSP430-FR2355 microprocessor to interface with two slave microprocessors (MSP430-FR2310) via I2C. The master device received input from a keypad and each slave device interfaced with another device, a Light Emitting Diode (LED) bar and a Liquid Crystal Display (LCD). Depending on the keypad button pressed, the slaves blinked different patterns on the LED bar and displayed different characters on the LCD display.

Setup:

The keypad was connected to the master device using ports 1.0-1.7, while the I2C clock and data lines utilized ports 4.7 and 4.6 respectively, Figure 1. Two pull-up resistors were added to these lines to achieve a consistent logic high. These lines were then connected to ports 1.3 and 1.2 respectively to each slave device. Slave device 1 interfaced with the LED bar through ports 1.0, 1.1, and 1.4-1.7, along with ports 2.7 and 2.6. A resistor of 101-ohm sat in the middle to reduce the current flow. Slave device 2 interfaced with the LCD display through ports 1.7-1.4 as data ports, 2.6 as the RS port, and 2.7 as the enable port. The LCD was powered by a 3.3-volt supply through a potentiometer output to the display power and the logic was powered using just the 3.3-volt supply.

Step 1 – Interface with an LED bar using I2C:

The requirement for the first portion of this experiment was to communicate to a slave device from the master using I2C. Before any data transmission, the master was locked with a numeric passcode and can only be unlocked with the passcode 437. When those numbers were pressed sequentially, the master was able to begin transmission.

The master was programmed to send the ASCII code of the button pressed on the keypad, Figure 2. Upon a button press, the I2C protocol initialized and began data transmission. It first communicated to the LED bar via slave address 0x0012 and the LED bar displayed a pattern depending on the ASCII code received. All dynamic LED patterns were programmed to hold state, so moving from pattern to pattern via various ASCII reception continued the LED pattern where it left off. These patterns were defined in project 3.

Step 2 – Interface with an LCD display using I2C:

The objective of the second portion of this experiment was to communicate to a second slave device from the master using I2C. After the master initiated an I2C transmission with the LED bar, it generated another start condition to slave address 0x0013. In much the same manner to the LED process, the master sent the ASCII value of the button pressed to the slave. On reset, the slave device sent the LCD the proper commands to begin proper operation. The slave then waited for data to be received from the master device. Once received, the slave would send the data to the LCD to display the character received, Figure 3. Once the slave had sent 32 characters to the LCD, it would send the command to clear the display and start displaying at the first position once again.

Conclusion:

The focus of this experiment was to transmit data in real time from a master device to 2 different slave devices using I2C. After the master device was unlocked via a numeric passcode, the data was transmitted as an ASCII code of the character obtained via a keypad button press. Both slave devices received this data, and each performed different tasks based on the received input. Slave device 1 produced respective patterns that held state upon a pattern change, and slave device 2 displayed the actual character pressed on the keypad on an LCD which was able to retain the first 32 characters received before being commanded to clear and start from the beginning.

Lessons Learned:

- Clear flags upon entering interrupt
- Add pull-up resistors on the SDA and SCL for consistent results

- Develop first on the dev-board, then move to soldered board.
- Look at all possible datasheets before moving on to working with a part

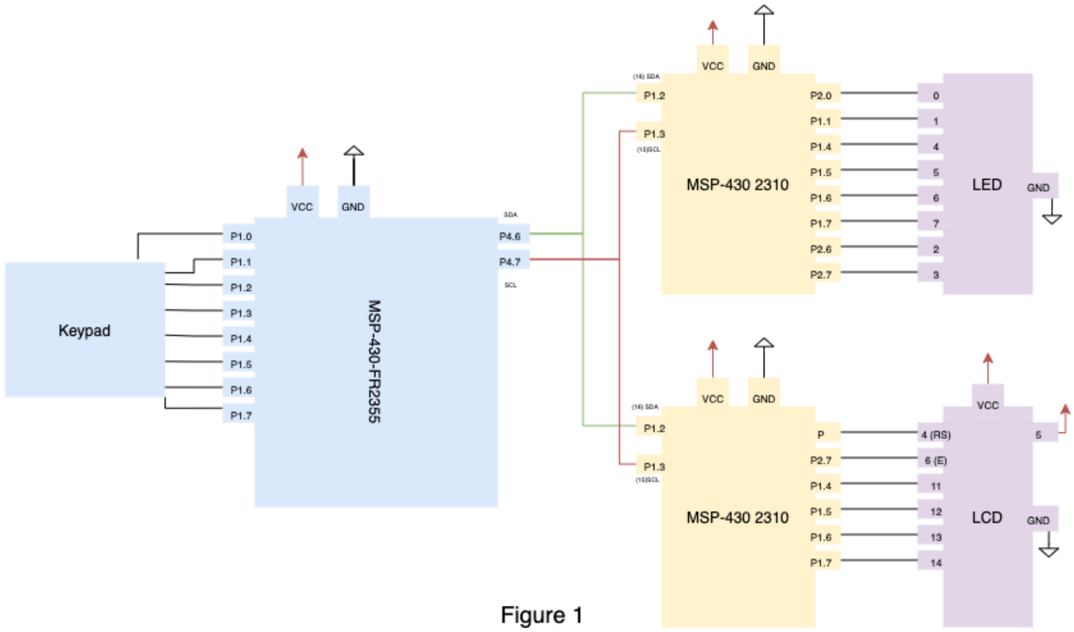

Figure 1

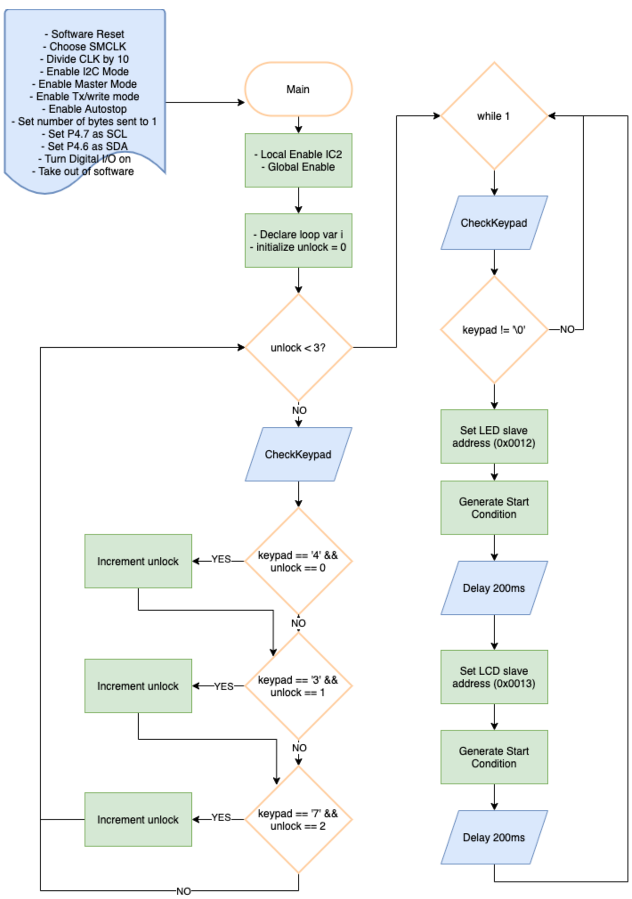

Figure 2

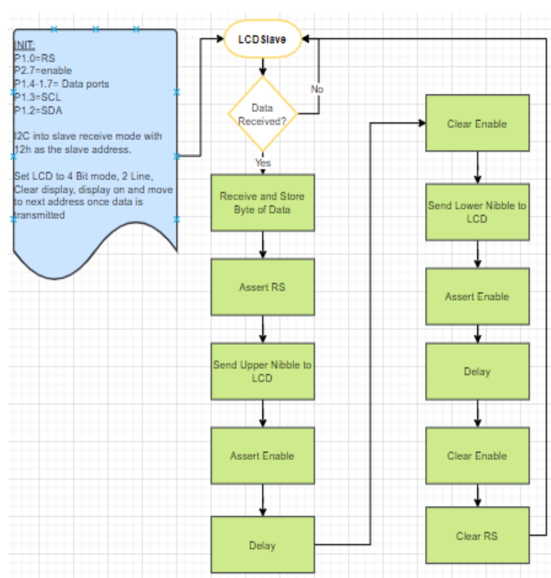

Figure 3: LCD Flowchart