Project 5 Temperature Sensor
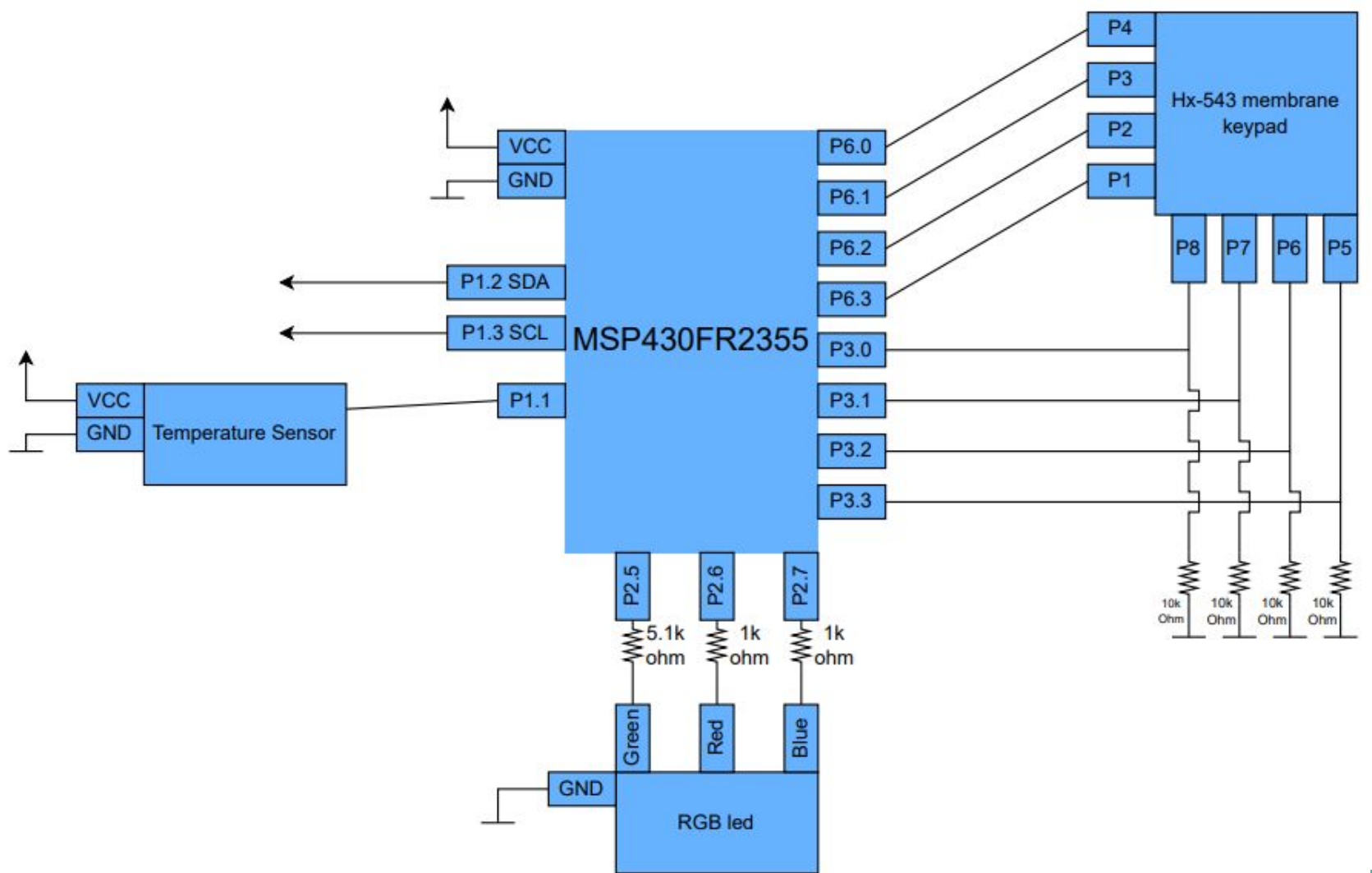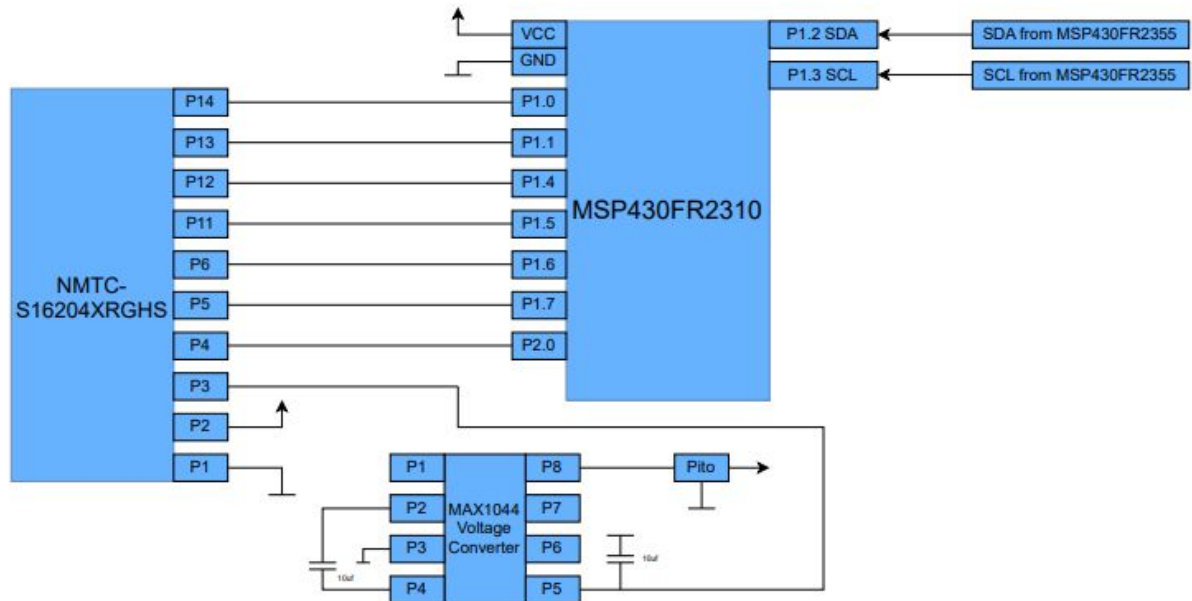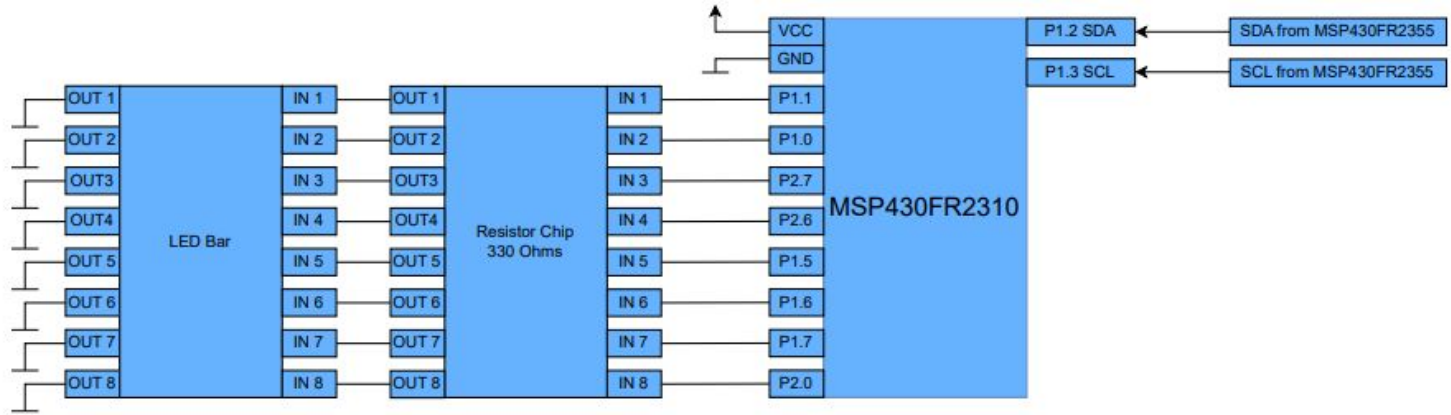
Noah Tegeder Luke Connolly

Goals

The goals of this project are similar to the last one.

Before selecting the pattern, a pattern selection mode must be entered, accessed by pressing A

A temperature sensor is added, and displays current temperature every half second on LCD.

The window for rolling average can be changed with another mode, from pressing B.

Diagram 1 (top):

LED Bar
- OUT 1, OUT 2, OUT3, OUT4, OUT 5, OUT 6, OUT 7, OUT 8
- IN 1, IN 2, IN 3, IN 4, IN 5, IN 6, IN 7, IN 8

Resistor Chip 330 Ohms
- OUT 1, OUT 2, OUT3, OUT4, OUT 5, OUT 6, OUT 7, OUT 8
- IN 1, IN 2, IN 3, IN 4, IN 5, IN 6, IN 7, IN 8

MSP430FR2310
- VCC
- GND
- P1.1, P1.0, P2.7, P2.6, P1.5, P1.6, P1.7, P2.0
- P1.2 SDA → SDA from MSP430FR2355
- P1.3 SCL → SCL from MSP430FR2355

Diagram 2 (bottom):

NMTC-S16204XRGHS
- P14, P13, P12, P11, P6, P5, P4, P3, P2, P1

MSP430FR2310
- VCC
- GND
- P1.0, P1.1, P1.4, P1.5, P1.6, P1.7, P2.0
- P1.2 SDA → SDA from MSP430FR2355
- P1.3 SCL → SCL from MSP430FR2355

MAX1044 Voltage Converter
- P1, P2, P3, P4
- P8, P7, P6, P5
- 10uf, 10uf
- Pito

Main.c for MSP430FR2355

Include:
keypad.h
msp430fr2355.h

Main
- Setup I2C Master Timer for sensor
- KeypadInit()
- enable interrupt
- While(1)
- scanPad()
- Switch (key pressed)

- If (A), write "Set Pattern" Let user choose pattern
- If (B), write "Set Window" Let user choose window

ISR Timer0 B0
- I2C transmit new temperature
- Clear interrupt flag

write_set_pattern
- Set I2C slave address
- Transmit letters

write_pattern_
- Set I2C slave address
- Transmit letters

transmit_pattern_led
- Set I2C slave address
- Transmit pattern code

ISR EUSCI_B1
- Send key pressed
- Clear interrupt flag

write_set_window
- Set I2C slave address
- Transmit letters

transmit_window_size
- Set I2C slave address
- Transmit new window

Keypad.h

Include
LEDbar.h
Define
characters = 4x4 char matrix of keys
rowPins & colPins = 1x4 matrix of BIT values

keypadInit()
- Setup 8 pins from Hx-543 keypad
- Setup Interrupts for row pins
- Setup red LED for heartbeat and green for keypress feedback
- Setup timer B0
- Setup and enable timer interrupts

lockKeypad()
- Turn off LED patterns, Disable interrupts
- Turn off LED patterns
- Wait for code
- Enable interrupts

scanKeypad()
- Make loop variables, clear cols
- For(cols)
- Set current col pin high
- For(rows)
- Check if row high
- Return key pressed (row, col)
- Return 0
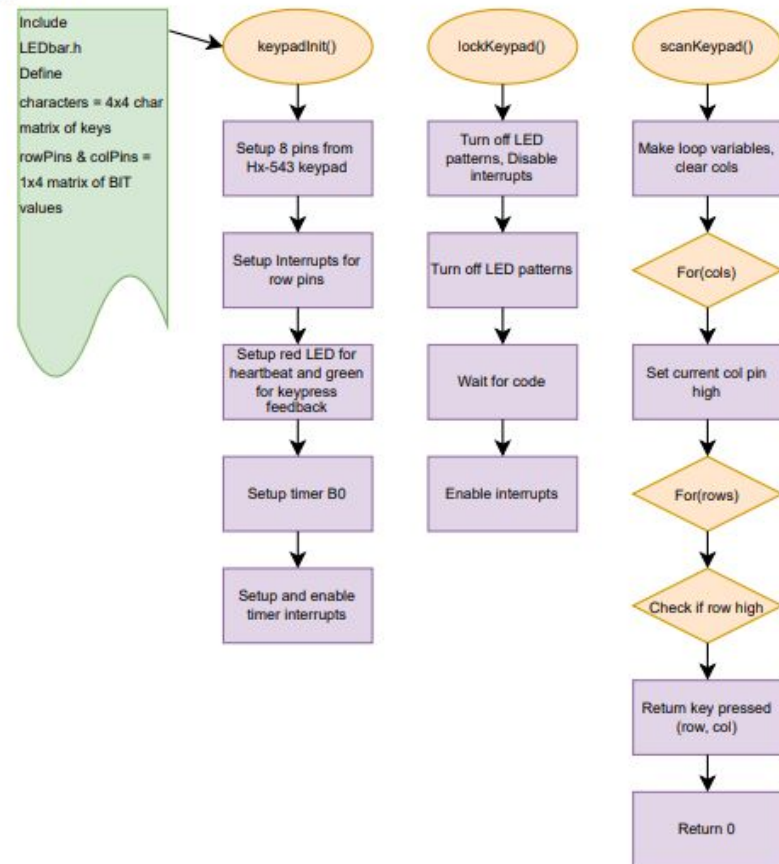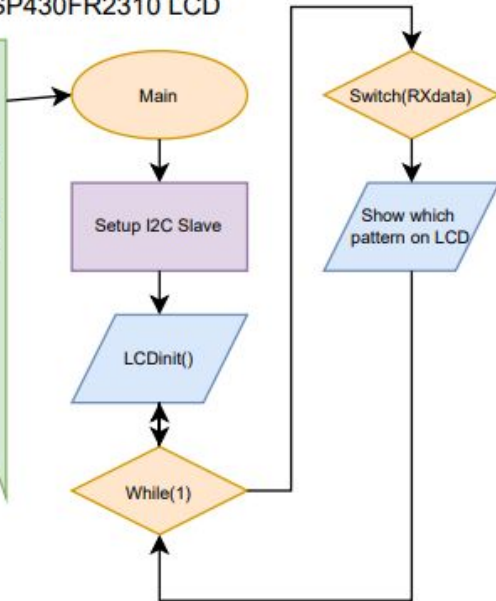
Main.c for MSP430FR2310 LCD

Include:
keypad.h
Define:
Slave Address = 0x48
Unsigned Char:
Recieved Data = 0x00
Transmit Data = 0x00
Char:
Pattern
Int:
Start1,2,3,5,6,7 = 0
Start4 = 255

Main → **Setup I2C Slave** → **LCDinit()** → **While(1)**

Switch(RXdata) → **Show which pattern on LCD**

LCD.h

LCDinit() → Initalize all pins for LCD

LCDsetup() → Clear LCD, 4 bit mode, 2 line mode, cursor

DB4(int status) → toggle bit4

DB5(int status) → toggle bit5

DB6(int status) → toggle bit6

DB7(int status) → toggle bit7

cursor_right() → Move the cursor one right

clear_cgram() → Clear the display

return_home() → Cursor back to top left

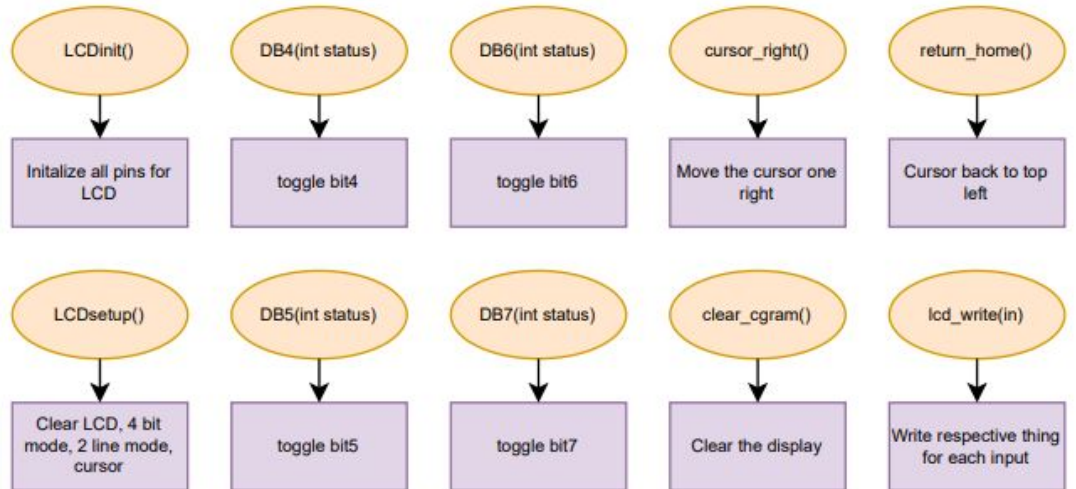**lcd_write(in)** → Write respective thing for each input

Main.c for MSP430FR2310 LEDbar

Include:

keypad.h

Define:

Slave Address = 0x48

Unsigned Char:

Recieved Data = 0x00

Transmit Data = 0x00

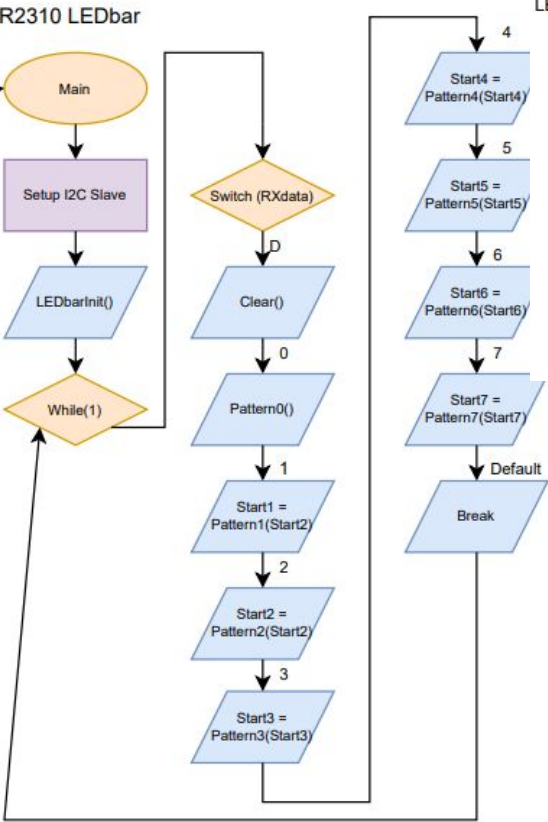Char:

Pattern

Int:

Start1,2,3,5,6,7 = 0

Start4 = 255

Main

↓

Setup I2C Slave

↓

LEDbarInit()

↓

While(1)

→ Switch (RXdata)

D: Clear()

0: Pattern0()

1: Start1 = Pattern1(Start2)

2: Start2 = Pattern2(Start2)

3: Start3 = Pattern3(Start3)

4: Start4 = Pattern4(Start4)

5: Start5 = Pattern5(Start5)

6: Start6 = Pattern6(Start6)

7: Start7 = Pattern7(Start7)

Default: Break

LEDbar.h

LEDbarInit() → Set all pins for LEDbar as outputs

On(int LED) → Switch Statement for which LED to turn on

Off(int LED) → Switch Statement for which LED to turn off

Pattern0_alt() → Holding position for LEDs, opposite of Pattern0

Pattern1(int start) → Swap between Pattern0 and Pattern0_alt

Pattern2(int start) → Up counter

Pattern7(int start) → Does the pattern

AllOn() → Turn on all LEDs

Clear() → Turn off all LEDs

Pattern0() → Holding position for LEDs

Pattern3(int start) → Does the pattern

Pattern4(int start) → Down counter

Pattern5(int start) → Does the pattern

Pattern6(int start) → Does the pattern