Lab 11

Keypad interfacing (port multiplexing)

Spring 2025

Submitted by: Mohsin Sajjad

Registration No: 22pwsce2149

Class Section: A

“On my honor, as student of University of Engineering and Technology, I have neither given nor received unauthorized assistance on this academic work.”

Mohsin Sajjad

Student Signature: ______________

Submitted to:

Engr. Faheem Jan

Month Day, Year (25 05, 2025)

Department of Computer Systems Engineering

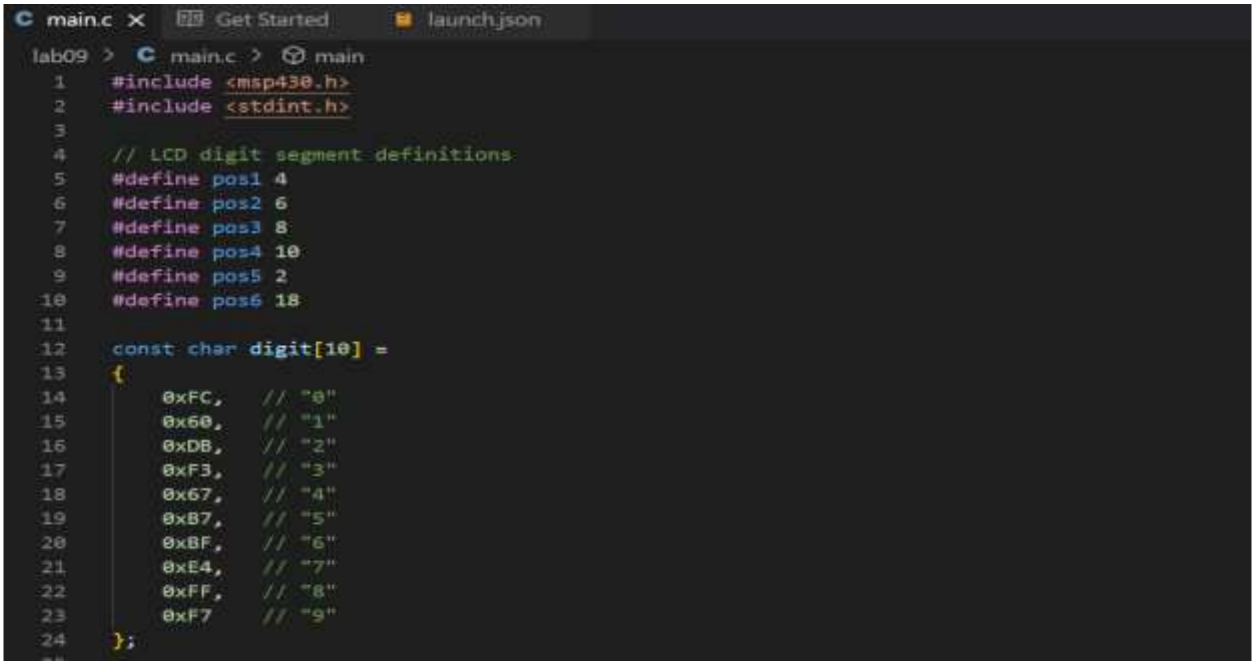
University of Engineering and Technology, Peshawar

Keypad interfacing (port multiplexing)

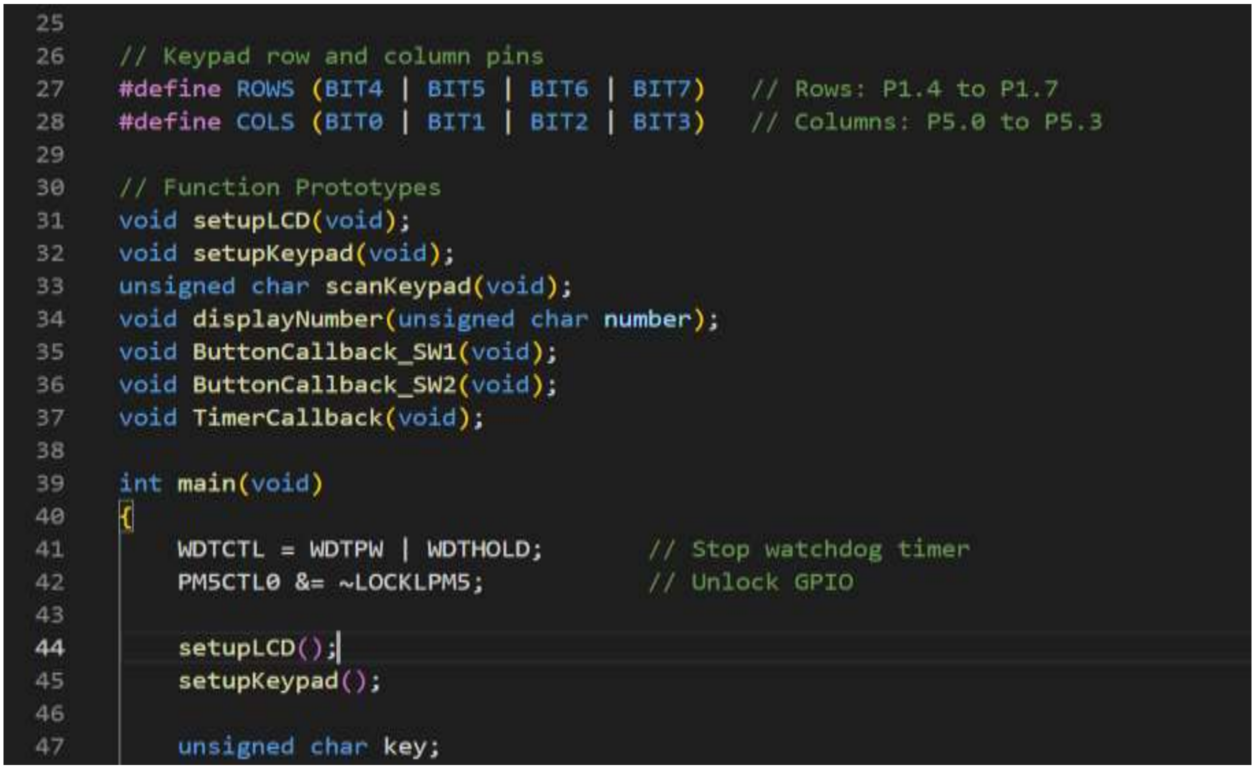
TASKS:

1) Run the program given in the lecture

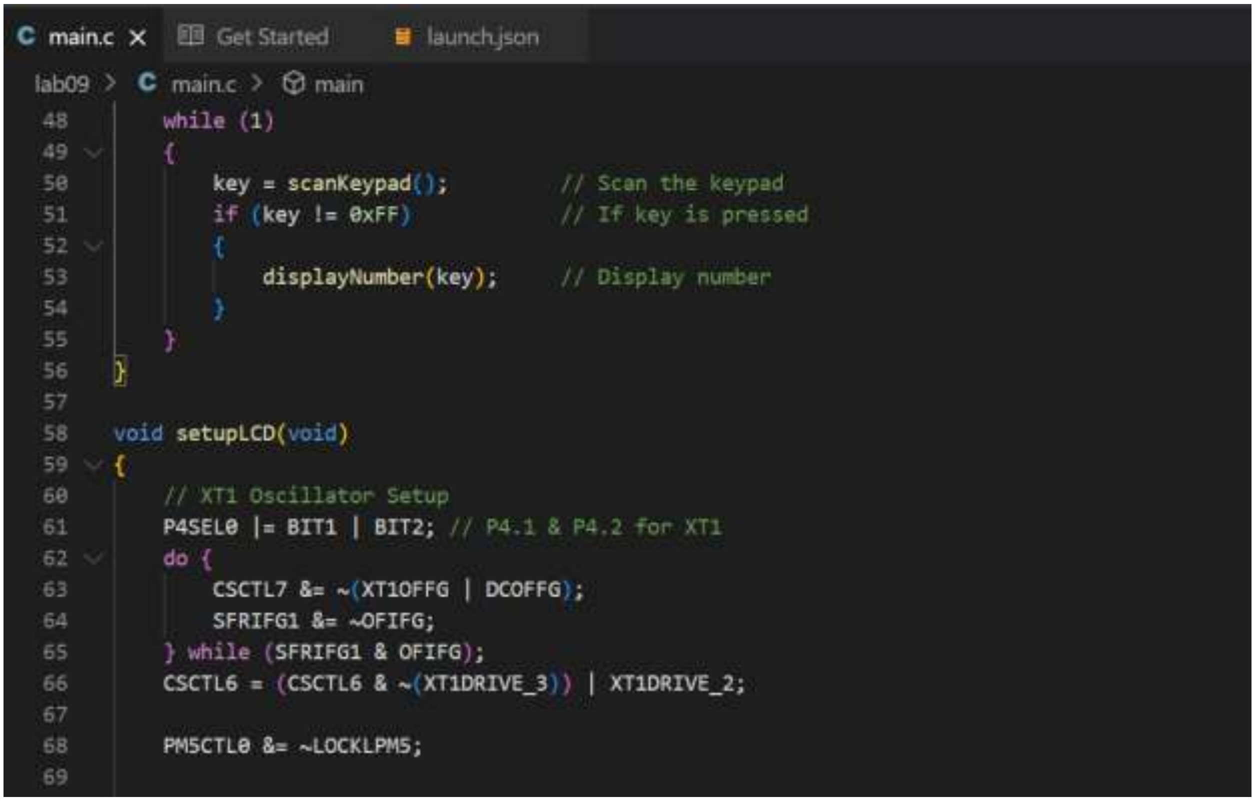
CODE:



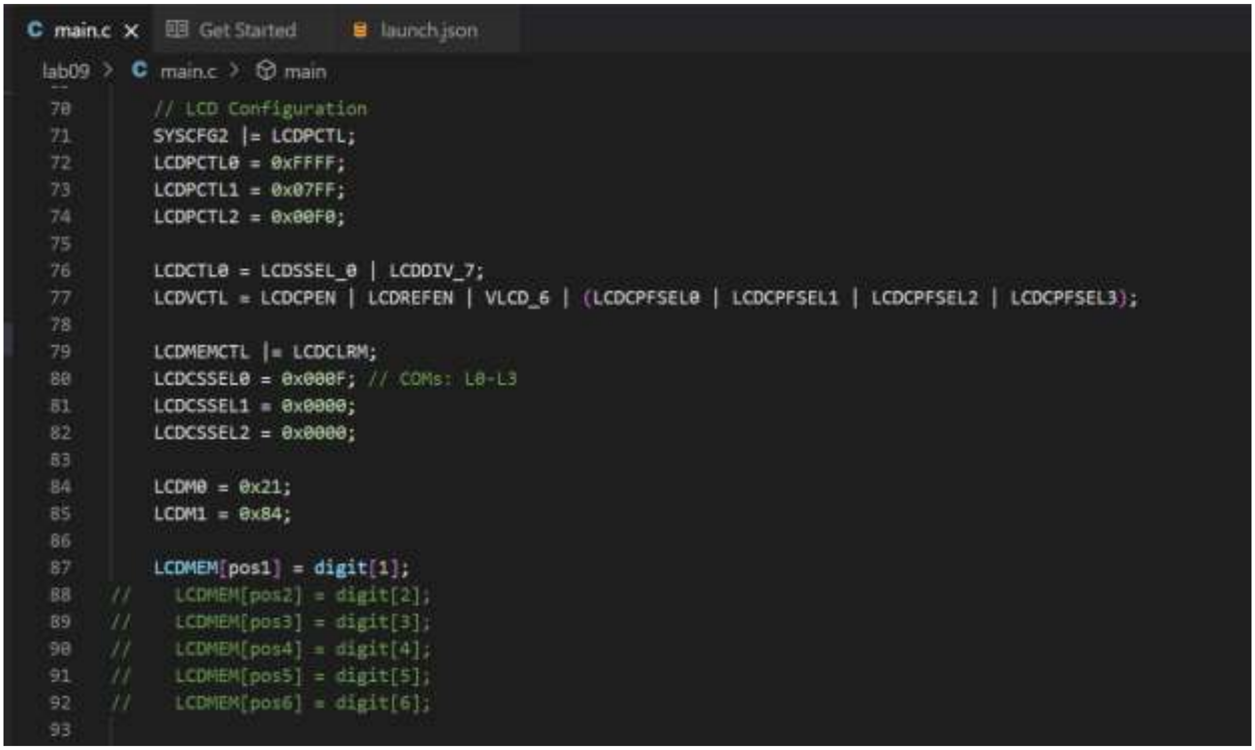
```c
#include <msp430.h>
#include <stdint.h>

// LCD digit segment definitions
#define pos1 4
#define pos2 6
#define pos3 8
#define pos4 10
#define pos5 2
#define pos6 18

const char digit[10] =
{
    0xFC,   // "0"
    0x60,   // "1"
    0xDB,   // "2"
    0xF3,   // "3"
    0x67,   // "4"
    0xB7,   // "5"
    0xBF,   // "6"
    0xE4,   // "7"
    0xFF,   // "8"
    0xF7    // "9"
};
```



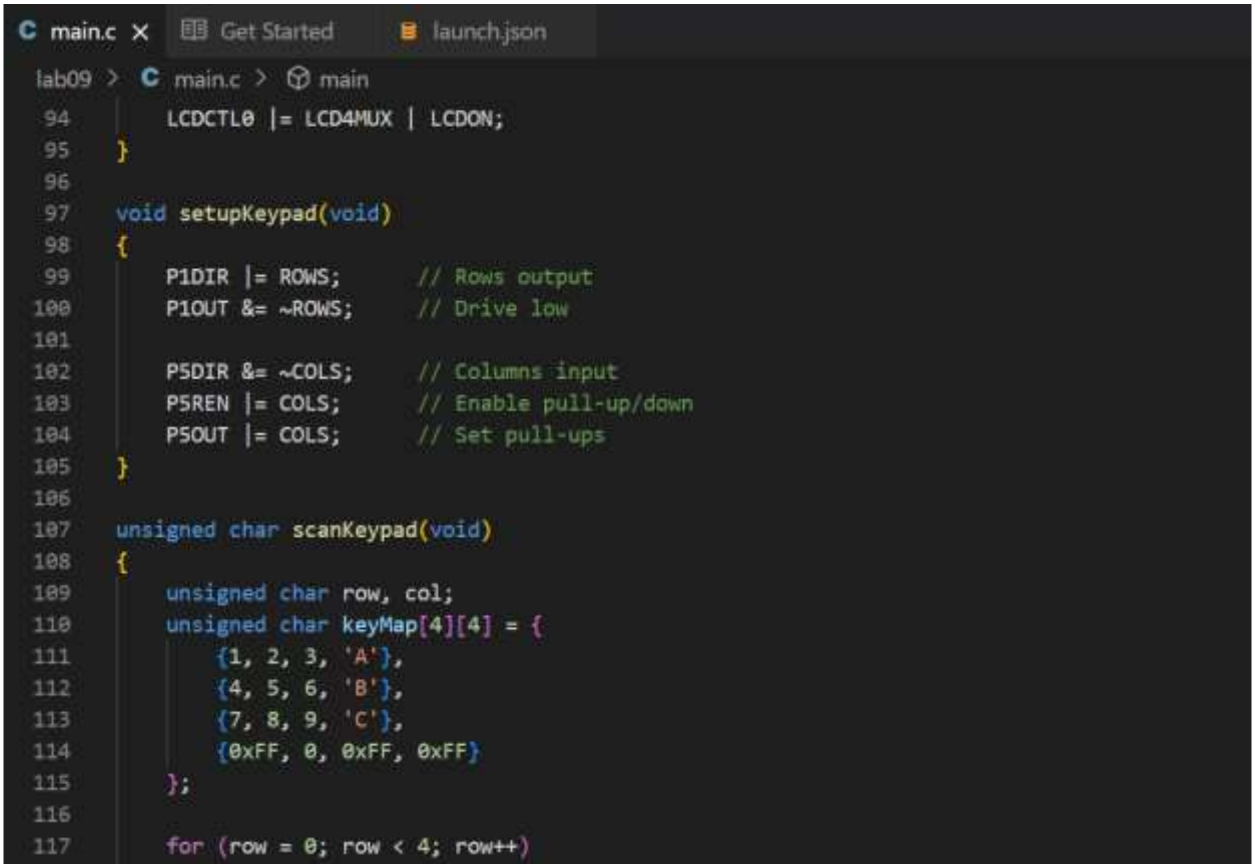
```c

// Keypad row and column pins
#define ROWS (BIT4 | BIT5 | BIT6 | BIT7)   // Rows: P1.4 to P1.7
#define COLS (BIT0 | BIT1 | BIT2 | BIT3)   // Columns: P5.0 to P5.3

// Function Prototypes
void setupLCD(void);
void setupKeypad(void);
unsigned char scanKeypad(void);
void displayNumber(unsigned char number);
void ButtonCallback_SW1(void);
void ButtonCallback_SW2(void);
void TimerCallback(void);

int main(void)
{
    WDTCTL = WDTPW | WDTHOLD;        // Stop watchdog timer
    PM5CTL0 &= ~LOCKLPM5;            // Unlock GPIO

    setupLCD();
    setupKeypad();

    unsigned char key;
```



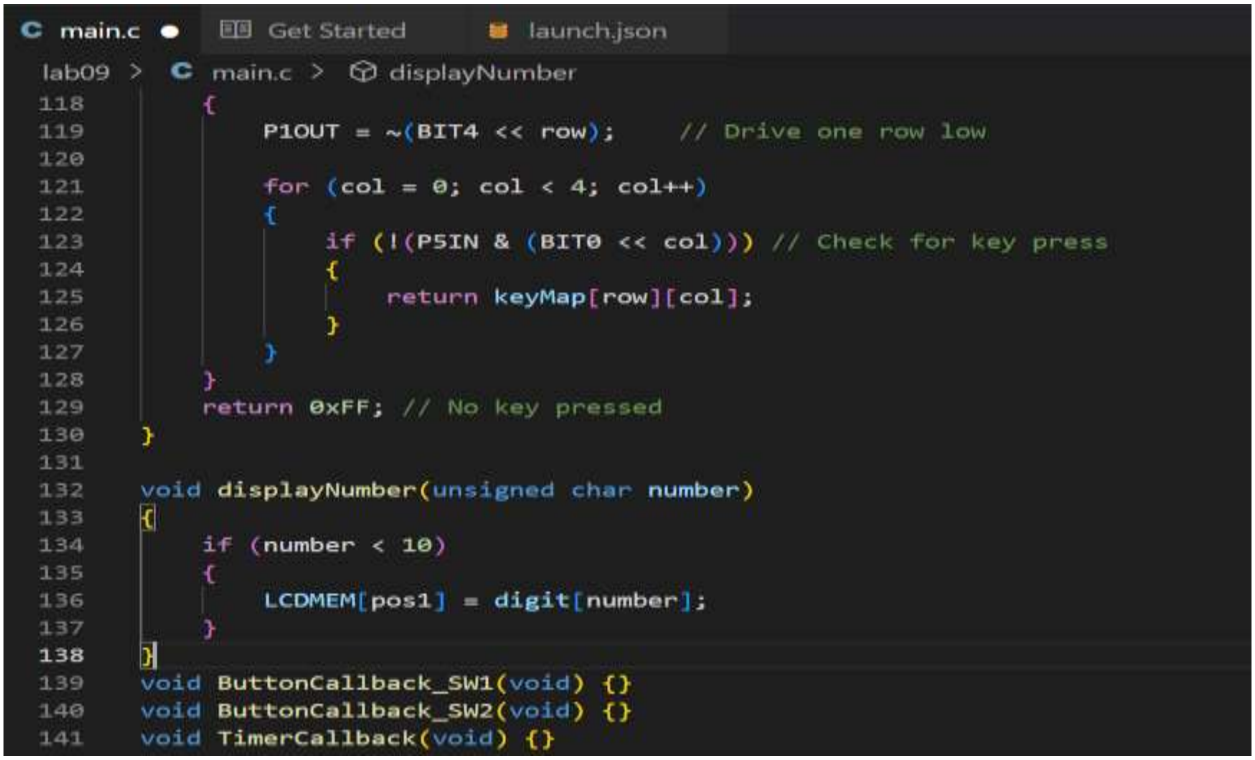
```c
    while (1)
    {
        key = scanKeypad();         // Scan the keypad
        if (key != 0xFF)            // If key is pressed
        {
            displayNumber(key);     // Display number
        }
    }
}

void setupLCD(void)
{
    // XT1 Oscillator Setup
    P4SEL0 |= BIT1 | BIT2; // P4.1 & P4.2 for XT1
    do {
        CSCTL7 &= ~(XT1OFFG | DCOFFG);
        SFRIFG1 &= ~OFIFG;
    } while (SFRIFG1 & OFIFG);
    CSCTL6 = (CSCTL6 & ~(XT1DRIVE_3)) | XT1DRIVE_2;

    PM5CTL0 &= ~LOCKLPM5;

```



```c
    // LCD Configuration
    SYSCFG2 |= LCDPCTL;
    LCDPCTL0 = 0xFFFF;
    LCDPCTL1 = 0x07FF;
    LCDPCTL2 = 0x00F0;

    LCDCTL0 = LCDSSEL_0 | LCDDIV_7;
    LCDVCTL = LCDCPEN | LCDREFEN | VLCD_6 | (LCDCPFSEL0 | LCDCPFSEL1 | LCDCPFSEL2 | LCDCPFSEL3);

    LCDMEMCTL |= LCDCLRM;
    LCDCSSEL0 = 0x000F; // COMs: L0-L3
    LCDCSSEL1 = 0x0000;
    LCDCSSEL2 = 0x0000;

    LCDM0 = 0x21;
    LCDM1 = 0x84;

    LCDMEM[pos1] = digit[1];
//    LCDMEM[pos2] = digit[2];
//    LCDMEM[pos3] = digit[3];
//    LCDMEM[pos4] = digit[4];
//    LCDMEM[pos5] = digit[5];
//    LCDMEM[pos6] = digit[6];

```

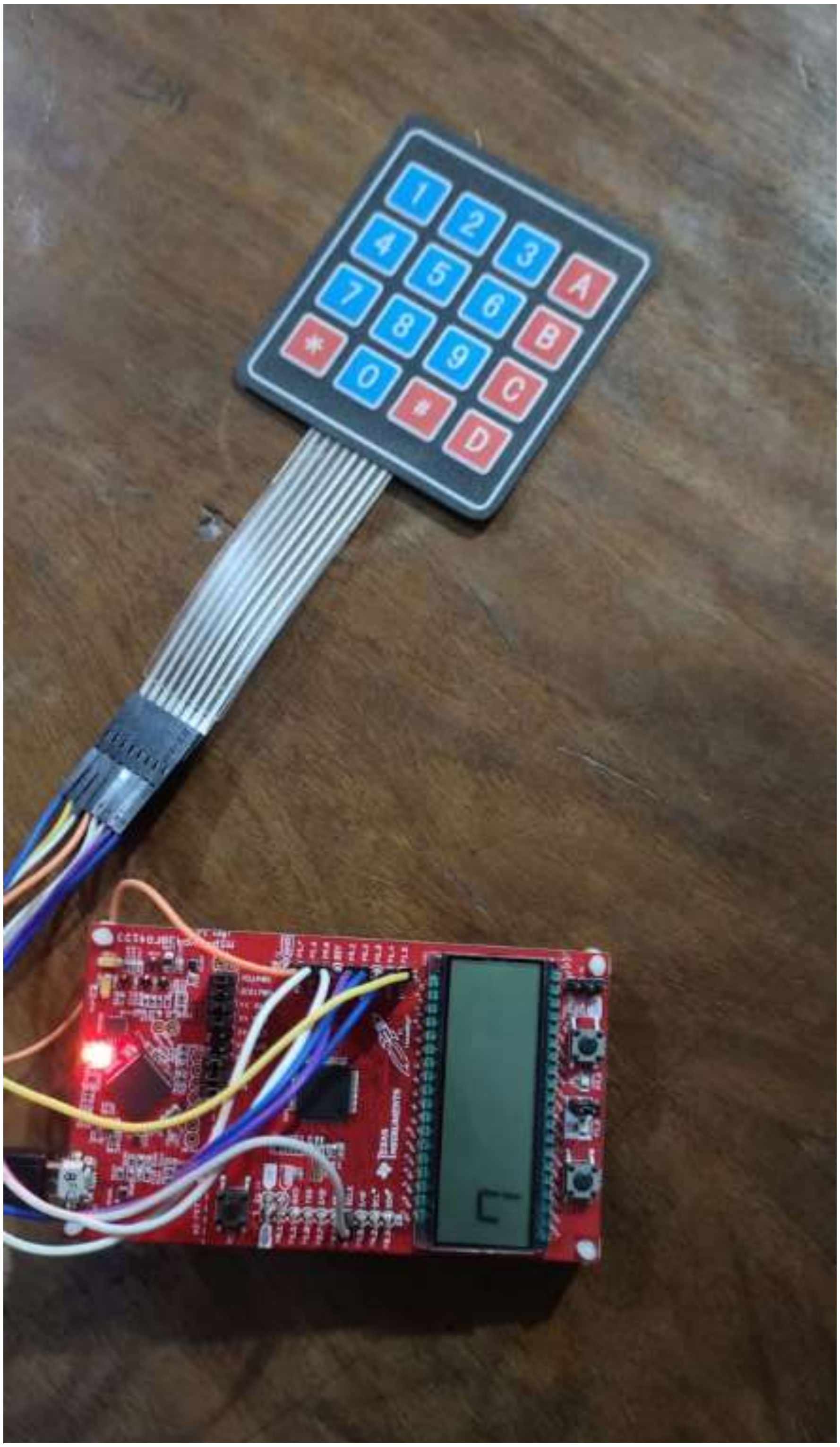


```c
    LCDCTL0 |= LCD4MUX | LCDON;
}

void setupKeypad(void)
{
    P1DIR |= ROWS;       // Rows output
    P1OUT &= ~ROWS;      // Drive low

    P5DIR &= ~COLS;      // Columns input
    P5REN |= COLS;       // Enable pull-up/down
    P5OUT |= COLS;       // Set pull-ups
}

unsigned char scanKeypad(void)
{
    unsigned char row, col;
    unsigned char keyMap[4][4] = {
        {1, 2, 3, 'A'},
        {4, 5, 6, 'B'},
        {7, 8, 9, 'C'},
        {0xFF, 0, 0xFF, 0xFF}
    };

    for (row = 0; row < 4; row++)
```



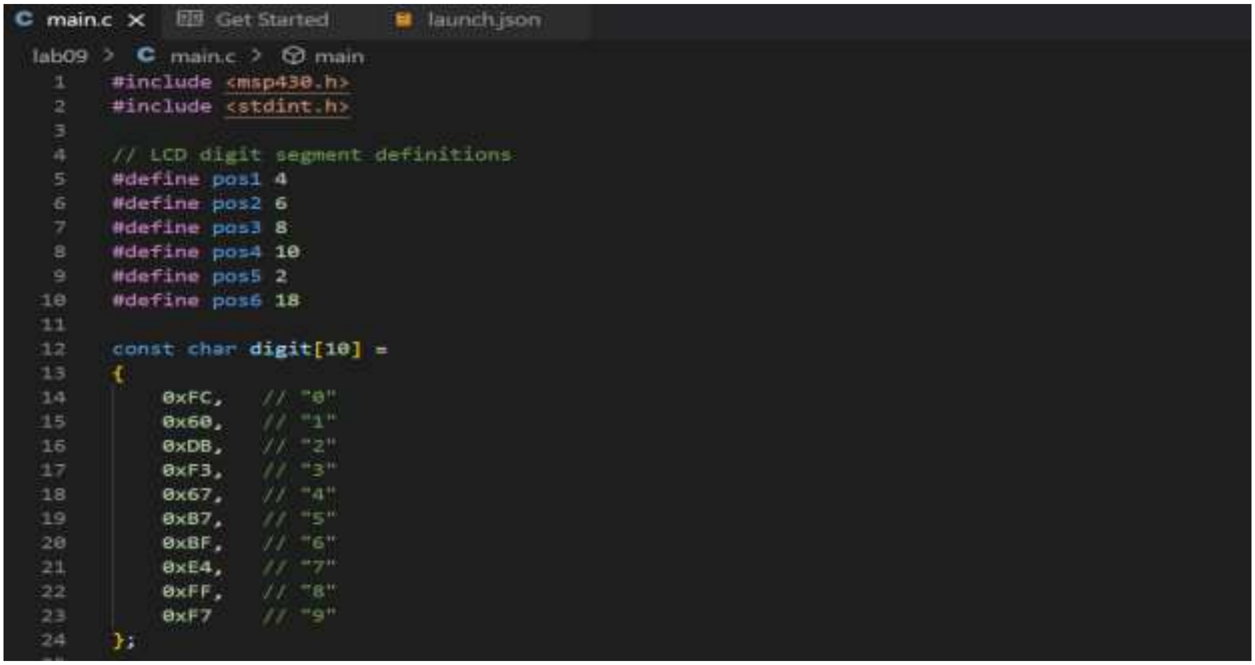
```c
    {
        P1OUT = ~(BIT4 << row);     // Drive one row low

        for (col = 0; col < 4; col++)
        {
            if (!(P5IN & (BIT0 << col))) // Check for key press
            {
                return keyMap[row][col];
            }
        }
    }
    return 0xFF; // No key pressed
}

void displayNumber(unsigned char number)
{
    if (number < 10)
    {
        LCDMEM[pos1] = digit[number];
    }
}
void ButtonCallback_SW1(void) {}
void ButtonCallback_SW2(void) {}
void TimerCallback(void) {}
```

OUTPUT:

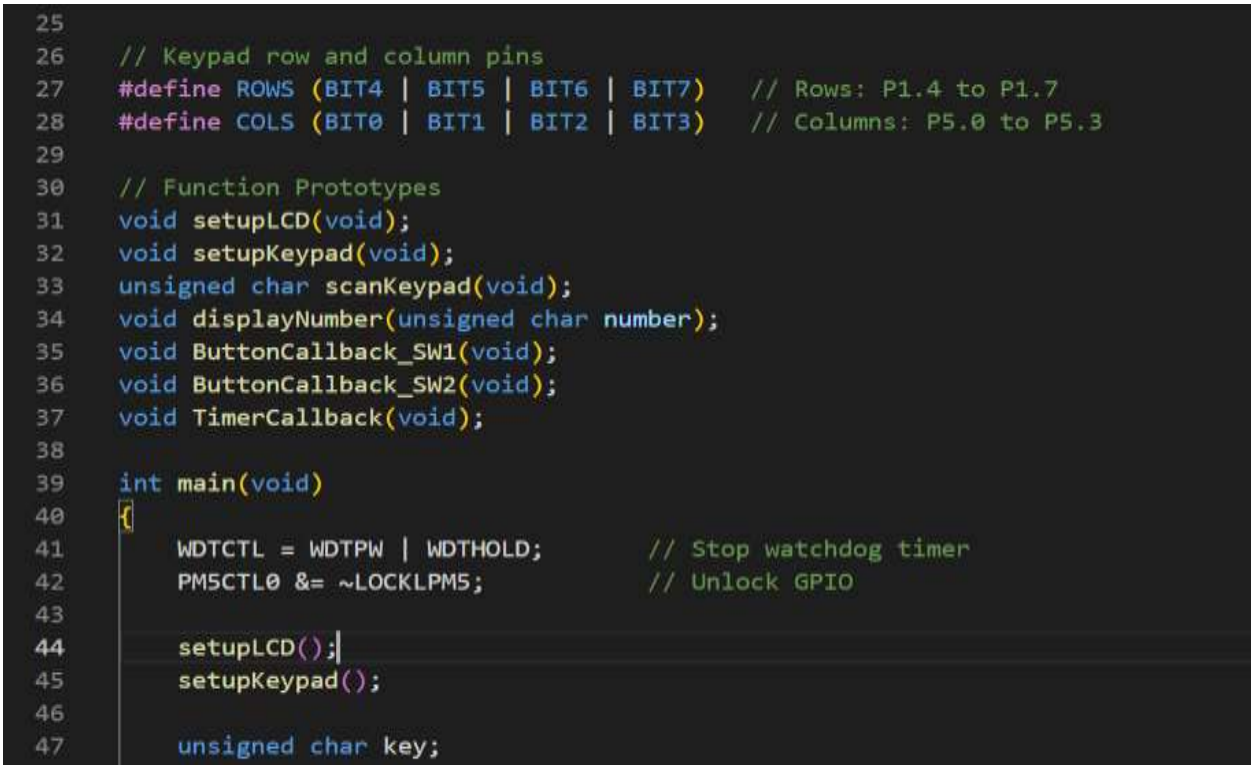


2) Display Number from to 1 to 9 on the onboard LCD (available on the Launchpad) of the MSP430Fr4133 MCU.

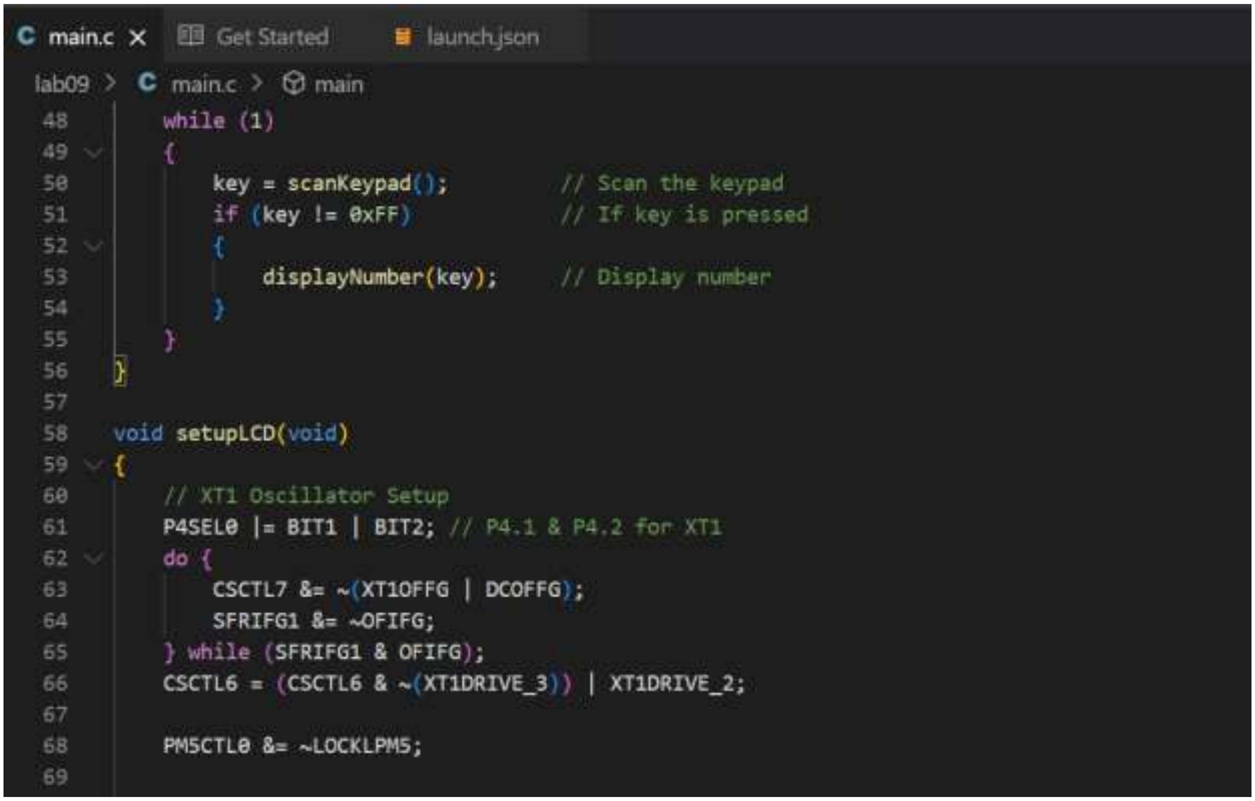
CODE:



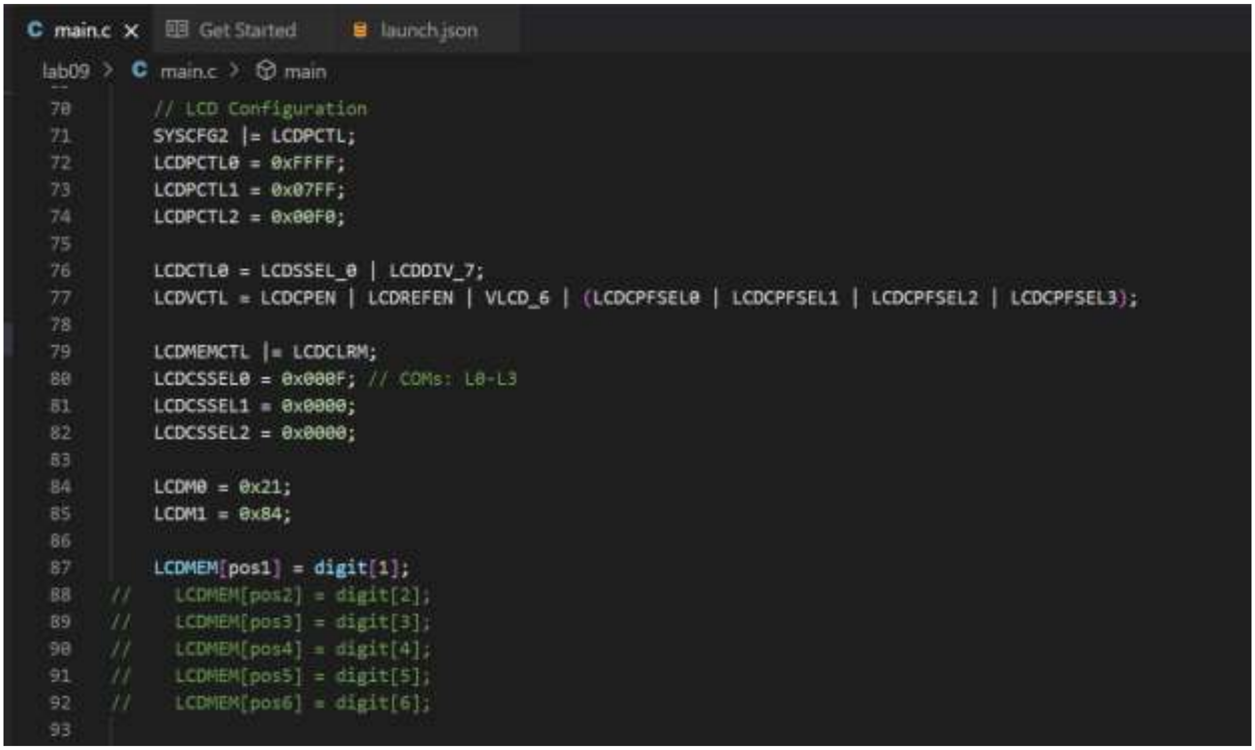
```c
#include <msp430.h>
#include <stdint.h>

// LCD digit segment definitions
#define pos1 4
#define pos2 6
#define pos3 8
#define pos4 10
#define pos5 2
#define pos6 18

const char digit[10] =
{
    0xFC,   // "0"
    0x60,   // "1"
    0xDB,   // "2"
    0xF3,   // "3"
    0x67,   // "4"
    0xB7,   // "5"
    0xBF,   // "6"
    0xE4,   // "7"
    0xFF,   // "8"
    0xF7    // "9"
};
```



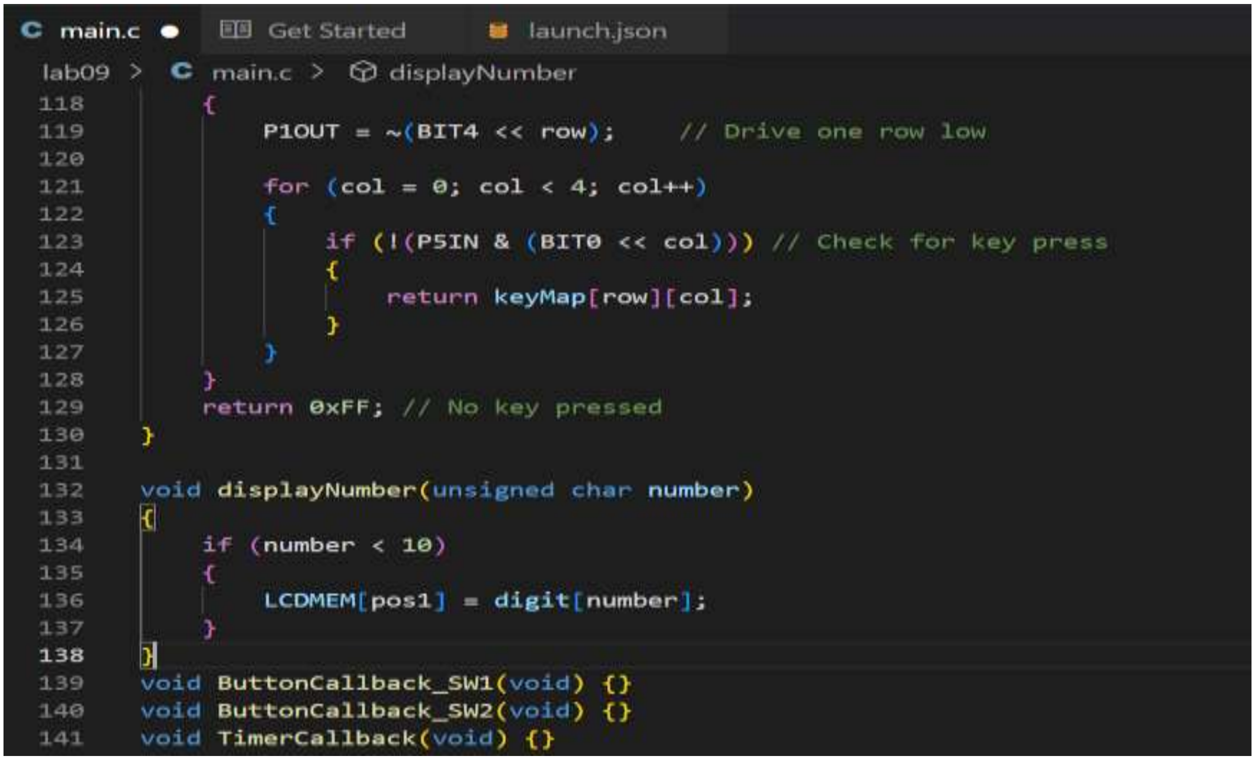
```c

// Keypad row and column pins
#define ROWS (BIT4 | BIT5 | BIT6 | BIT7)   // Rows: P1.4 to P1.7
#define COLS (BIT0 | BIT1 | BIT2 | BIT3)   // Columns: P5.0 to P5.3

// Function Prototypes
void setupLCD(void);
void setupKeypad(void);
unsigned char scanKeypad(void);
void displayNumber(unsigned char number);
void ButtonCallback_SW1(void);
void ButtonCallback_SW2(void);
void TimerCallback(void);

int main(void)
{
    WDTCTL = WDTPW | WDTHOLD;       // Stop watchdog timer
    PM5CTL0 &= ~LOCKLPM5;           // Unlock GPIO

    setupLCD();
    setupKeypad();

    unsigned char key;
```



```c
    while (1)
    {
        key = scanKeypad();         // Scan the keypad
        if (key != 0xFF)            // If key is pressed
        {
            displayNumber(key);     // Display number
        }
    }
}

void setupLCD(void)
{
    // XT1 Oscillator Setup
    P4SEL0 |= BIT1 | BIT2; // P4.1 & P4.2 for XT1
    do {
        CSCTL7 &= ~(XT1OFFG | DCOFFG);
        SFRIFG1 &= ~OFIFG;
    } while (SFRIFG1 & OFIFG);
    CSCTL6 = (CSCTL6 & ~(XT1DRIVE_3)) | XT1DRIVE_2;

    PM5CTL0 &= ~LOCKLPM5;

```

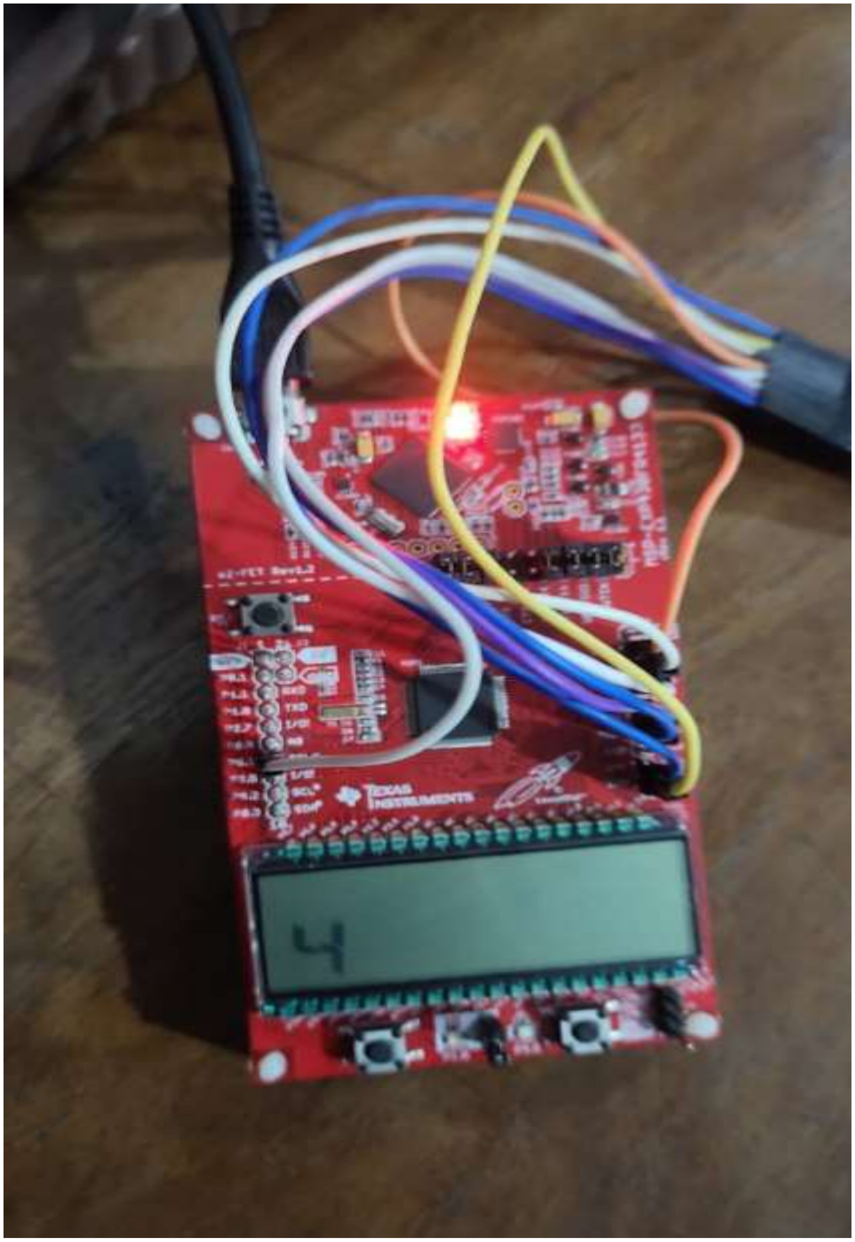


```c
    // LCD Configuration
    SYSCFG2 |= LCDPCTL;
    LCDPCTL0 = 0xFFFF;
    LCDPCTL1 = 0x07FF;
    LCDPCTL2 = 0x00F0;

    LCDCTL0 = LCDSSEL_0 | LCDDIV_7;
    LCDVCTL = LCDCPEN | LCDREFEN | VLCD_6 | (LCDCPFSEL0 | LCDCPFSEL1 | LCDCPFSEL2 | LCDCPFSEL3);

    LCDMEMCTL |= LCDCLRM;
    LCDCSSEL0 = 0x000F; // COMs: L0-L3
    LCDCSSEL1 = 0x0000;
    LCDCSSEL2 = 0x0000;

    LCDM0 = 0x21;
    LCDM1 = 0x84;

    LCDMEM[pos1] = digit[1];
//    LCDMEM[pos2] = digit[2];
//    LCDMEM[pos3] = digit[3];
//    LCDMEM[pos4] = digit[4];
//    LCDMEM[pos5] = digit[5];
//    LCDMEM[pos6] = digit[6];

```



```c
    {
        P1OUT = ~(BIT4 << row);     // Drive one row low

        for (col = 0; col < 4; col++)
        {
            if (!(P5IN & (BIT0 << col))) // Check for key press
            {
                return keyMap[row][col];
            }
        }
    }
    return 0xFF; // No key pressed
}

void displayNumber(unsigned char number)
{
    if (number < 10)
    {
        LCDMEM[pos1] = digit[number];
    }
}
void ButtonCallback_SW1(void) {}
void ButtonCallback_SW2(void) {}
void TimerCallback(void) {}
```

Output:

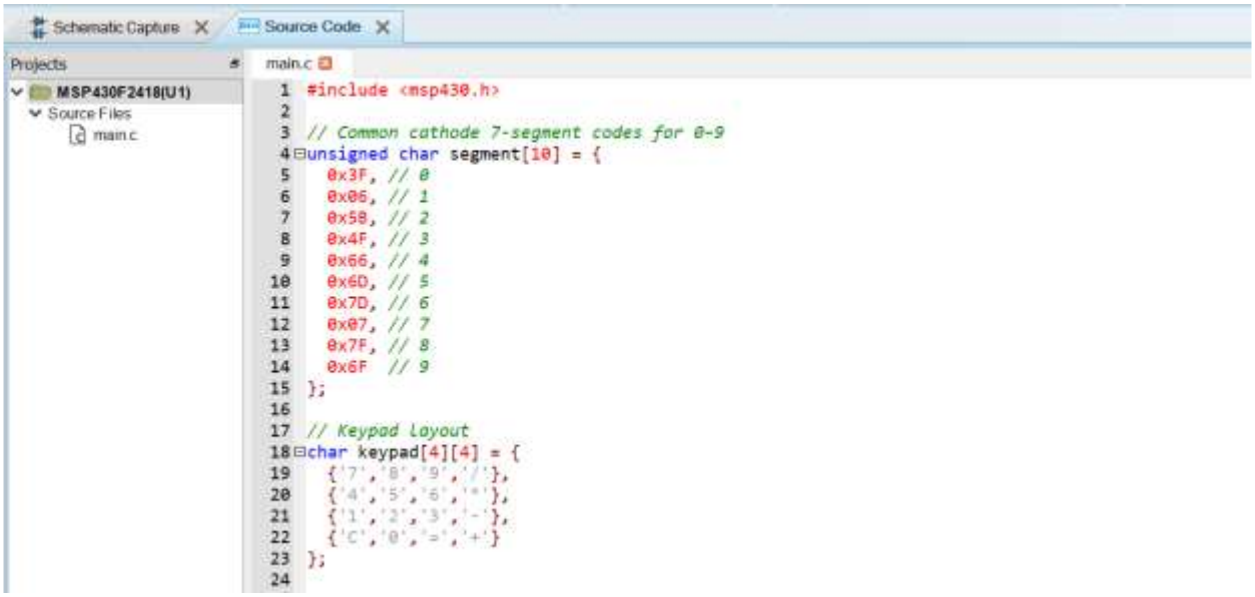


(Home Task)

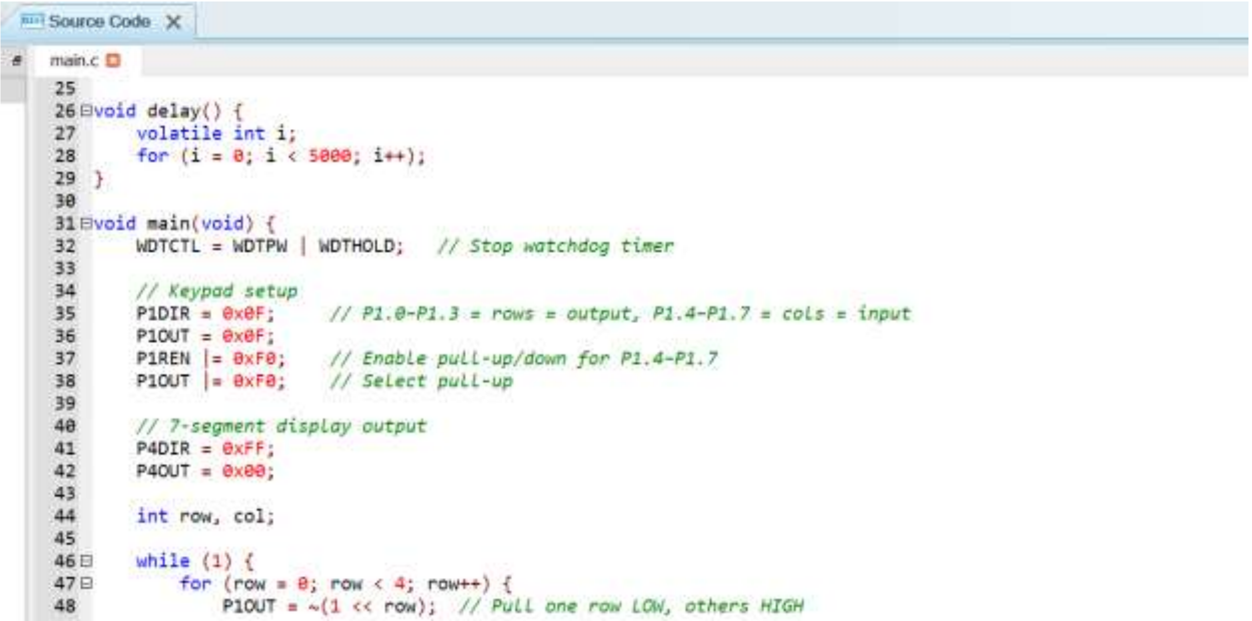
2) Display Numbers from 1 to 9 on the seven segment display and ON the corresponding LED’s attached with any PORT of the MCU

Note: Attach the seven segment with P3 or any other PORT and keypad with P1 or any other Port (you can use proteus).

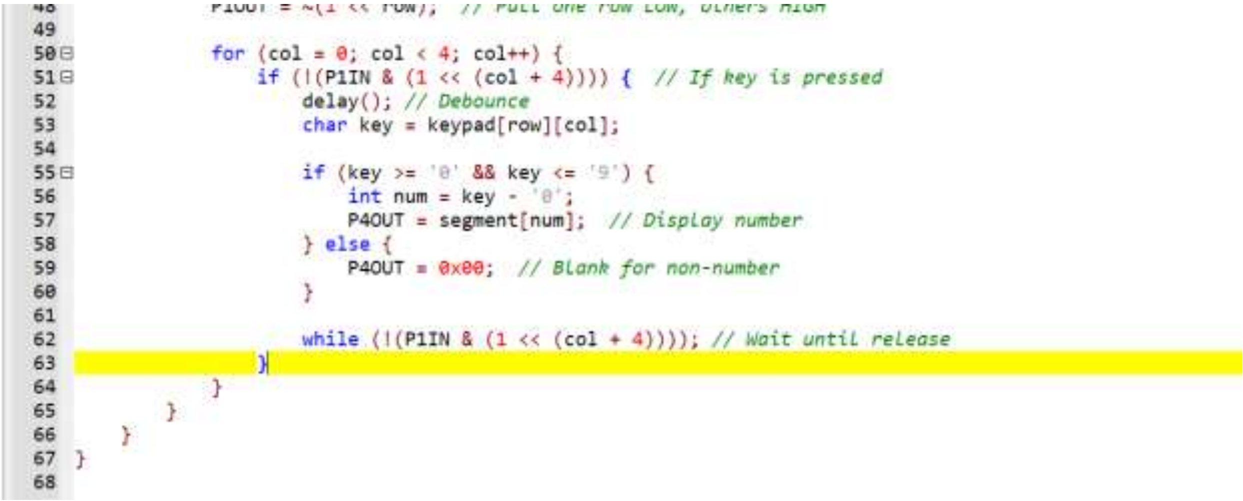
CODE:



```
#include <msp430.h>

// Common cathode 7-segment codes for 0-9
unsigned char segment[10] = {
  0x3F, // 0
  0x06, // 1
  0x5B, // 2
  0x4F, // 3
  0x66, // 4
  0x6D, // 5
  0x7D, // 6
  0x07, // 7
  0x7F, // 8
  0x6F  // 9
};

// Keypad Layout
char keypad[4][4] = {
  {'7','8','9','/'},
  {'4','5','6','*'},
  {'1','2','3','-'},
  {'C','0','=','+'}
};

```

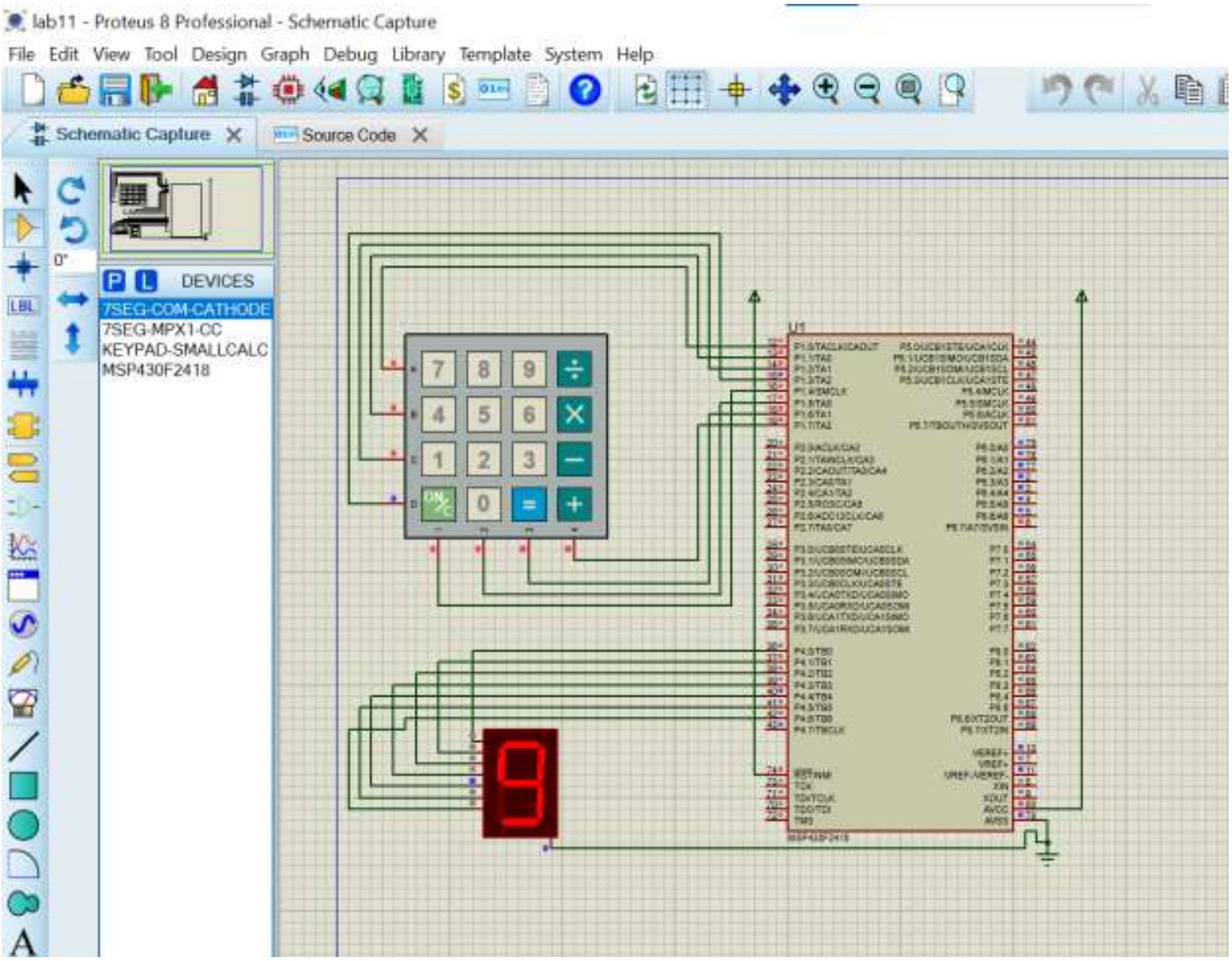


```

void delay() {
    volatile int i;
    for (i = 0; i < 5000; i++);
}

void main(void) {
    WDTCTL = WDTPW | WDTHOLD;   // Stop watchdog timer

    // Keypad setup
    P1DIR = 0x0F;     // P1.0-P1.3 = rows = output, P1.4-P1.7 = cols = input
    P1OUT = 0x0F;
    P1REN |= 0xF0;    // Enable pull-up/down for P1.4-P1.7
    P1OUT |= 0xF0;    // Select pull-up

    // 7-segment display output
    P4DIR = 0xFF;
    P4OUT = 0x00;

    int row, col;

    while (1) {
        for (row = 0; row < 4; row++) {
            P1OUT = ~(1 << row);  // Pull one row LOW, others HIGH
```



```
            P1OUT = ~(1 << row);  // Pull one row LOW, others HIGH

            for (col = 0; col < 4; col++) {
                if (!(P1IN & (1 << (col + 4)))) {  // If key is pressed
                    delay(); // Debounce
                    char key = keypad[row][col];

                    if (key >= '0' && key <= '9') {
                        int num = key - '0';
                        P4OUT = segment[num];  // Display number
                    } else {
                        P4OUT = 0x00;  // Blank for non-number
                    }

                    while (!(P1IN & (1 << (col + 4)))); // Wait until release
                }
            }
        }
    }
}
```

Output:



lab11 - Proteus 8 Professional - Schematic Capture
File Edit View Tool Design Graph Debug Library Template System Help
Schematic Capture
Source Code
0°
P L DEVICES
7SEG-COM-CATHODE
7SEG-MPX1-CC
KEYPAD-SMALLCALC
MSP430F2418
7 8 9 ÷
4 5 6 ×
1 2 3 −
ON/C 0 = +
U1
MSP430F2418
9