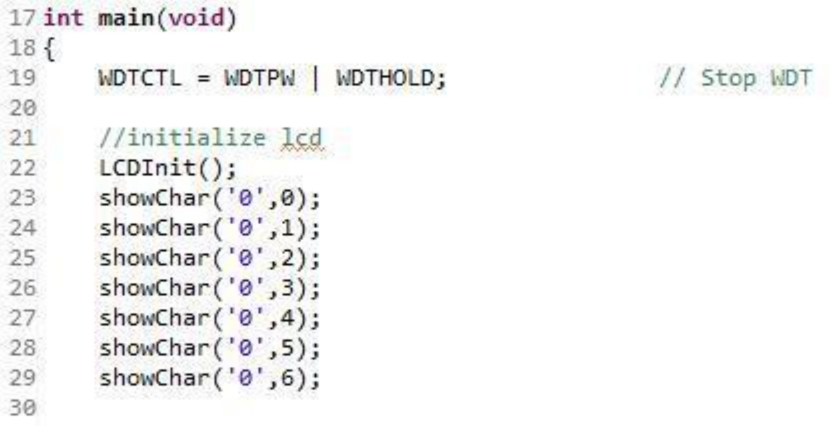
Lab 5 – LCD Display

Austin Huang and Matthew Rodriguez

Figures

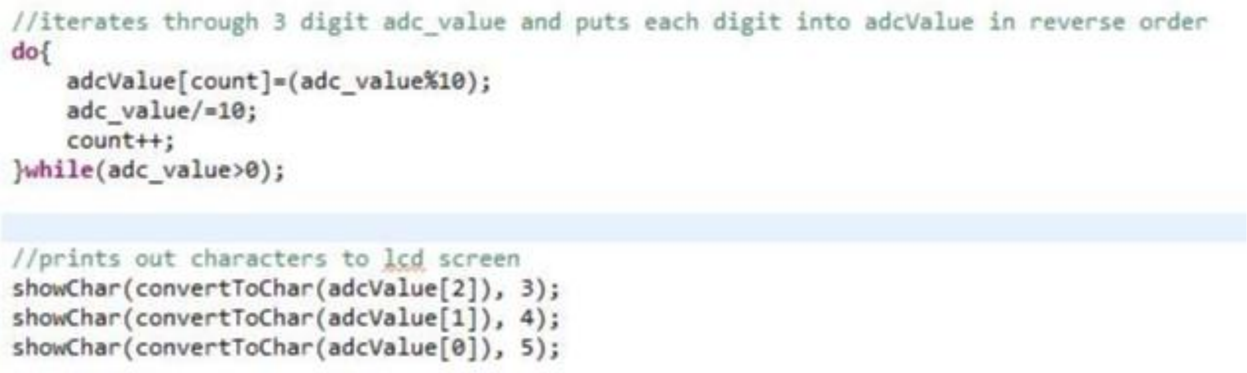


```
int main(void)
{
    WDTCTL = WDTPW | WDTHOLD;               // Stop WDT

    //initialize lcd
    LCDInit();
    showChar('0',0);
    showChar('0',1);
    showChar('0',2);
    showChar('0',3);
    showChar('0',4);
    showChar('0',5);
    showChar('0',6);

```

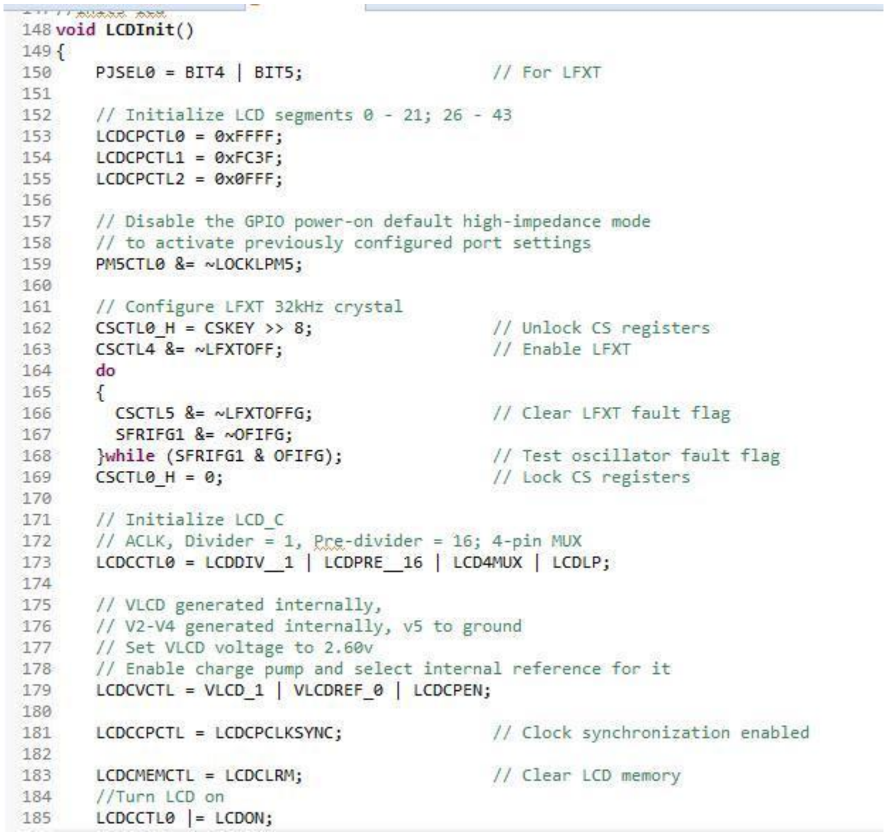
Figure 1: LCD code



```
//iterates through 3 digit adc_value and puts each digit into adcValue in reverse order
do{
    adcValue[count]=(adc_value%10);
    adc_value/=10;
    count++;
}while(adc_value>0);


//prints out characters to lcd screen
showChar(convertToChar(adcValue[2]), 3);
showChar(convertToChar(adcValue[1]), 4);
showChar(convertToChar(adcValue[0]), 5);
```

Figure 2: More LCD code



```
void LCDInit()
{
    PJSEL0 = BIT4 | BIT5;                   // For LFXT

    // Initialize LCD segments 0 - 21; 26 - 43
    LCDCPCTL0 = 0xFFFF;
    LCDCPCTL1 = 0xFC3F;
    LCDCPCTL2 = 0x0FFF;

    // Disable the GPIO power-on default high-impedance mode
    // to activate previously configured port settings
    PM5CTL0 &= ~LOCKLPM5;

    // Configure LFXT 32kHz crystal
    CSCTL0_H = CSKEY >> 8;                  // Unlock CS registers
    CSCTL4 &= ~LFXTOFF;                     // Enable LFXT
    do
    {
      CSCTL5 &= ~LFXTOFFG;                  // Clear LFXT fault flag
      SFRIFG1 &= ~OFIFG;
    }while (SFRIFG1 & OFIFG);               // Test oscillator fault flag
    CSCTL0_H = 0;                           // Lock CS registers

    // Initialize LCD_C
    // ACLK, Divider = 1, Pre-divider = 16; 4-pin MUX
    LCDCCTL0 = LCDDIV__1 | LCDPRE__16 | LCD4MUX | LCDLP;

    // VLCD generated internally,
    // V2-V4 generated internally, v5 to ground
    // Set VLCD voltage to 2.60v
    // Enable charge pump and select internal reference for it
    LCDCVCTL = VLCD_1 | VLCDREF_0 | LCDCPEN;

    LCDCCPCTL = LCDCPCLKSYNC;               // Clock synchronization enabled

    LCDCMEMCTL = LCDCLRM;                   // Clear LCD memory
    //Turn LCD on
    LCDCCTL0 |= LCDON;
```

Figure 3: Initialization code

LCD Discussion

The temperature was read from a voltage regulator and displayed on the LCD screen of the FR6989. The temperature reading's accuracy varies

depending on how far the temperature being read deviates from the 30 degrees celcius range. As the temperature leaves the range, it varies

by a few degrees depending on how far the temperature deviates from the 30's range, but the reading is still relatively accurate

regardless. The goal of this part of the lab however, is not to measure the accuracy of the temperature. The lab focuses on being able

to present data on the LCD display of the MSP430 board, which was successfully done. The result is seen in the attached video labeled "Lab5LCD"
in the repository. Additionally, the code used to implement this functionality is uploaded to this repository.

LCD Code

The code should be compiled in CSS and run in the MSP430. The MSP430 should then be placed in the appropriate part of the temperature sensor circuit.

The above code shows the usage of the showChar() method. This is the method used to output values to the LCD display.

The MSP430 takes the value from adc_value and converts it into decimal as seen in the Figure 2 code. It then stores each digit into a fixed-size array called adcValue[]. It uses the showChar() function to print the specified character or digit onto a specified section on the LCD display. The convertToChar() function takes in an integer input and converts it into a character that can be displayed on the LCD display. This allows for the board to be able to show a temperature reading of the system.

The above code was given by Russel and is used for initialization. It initializes all 43 LCD segments and sets up the board to be used for the labs.