**Prompt:**

For System B, students must write a C/C++ program for the following functions.

For System B, you will write a program to implement the following functions.

Use the ADC module on the MSP430FR5994 MCU, and make sure to use it as a 12-bit resolution (unsigned) as we studied in class.

For the first line of the LCD, display your name(s).

For the second line of the LCD, display the converted voltage that can be read through the P4.1 pin.

The voltage range from 0 V to 3.3 V on the LCD display should be matched with the actual full range of the rotation of the potentiometer knob.

The decimal places should be two. For instance, 1.23

You should play a different note or stop playing a note as you tweak the potentiometer and as follows:

|  |  |
| --- | --- |
| Voltage < 1V | Stop playing a note |
| 1V <= Voltage < 2V | C note (CCR0 value of 956) |
| 2V <= Voltage < 3V | D note (CCR0 value of 851) |
| Voltage >= 3V | E note (CCR0 value of 758) |

**Example of a program using the LCD and similar logic (That might help):**

**Example of program that uses the buzzer that may help:**

#include <msp430.h>

/\*\*

\* Brink van Eeden

\*

\*/

int main(void)

{

WDTCTL = WDTPW | WDTHOLD; // stop watchdog timer

PM5CTL0 &= ~LOCKLPM5; // clear locklpm5 bit

// //Part A -----

//

// P6DIR |= BIT0; // set the output direction of P6.0 as an output

// TA1CCR0 = 667; //set the timer a ccr0 value (1 MHz / 2\*(750 Hz) = 667)

// TA1CTL = TASSEL\_2 | MC\_1 | TACLR; //TA1CTL Setup

//

// while(1){

// if((TA1CCTL0 & CCIFG) != 0){ //Checks whether the CCIFG flag is set to set the TAIFG

// P6OUT ^= BIT0; //toggle pin 6.00

// TA1CCTL0 &= ~CCIFG; //clear the CCIFG flag

// }

// }

//Part B -----

TA1CTL = TASSEL\_2 | MC\_1 | TACLR;

TA1CCTL0 = CCIE;

\_\_enable\_interrupt();

P6DIR |= BIT0; // output for P6.0 -- Buzzer

P8DIR |= 0xFF; // Output for 8.0, 8.1, 8.2, -- LEDS

P8OUT = 0XFF; // SET IT TO HIGH (OFF INITIALLY)

P7DIR |= 0X03; //OUTPUT 7.0, 7.1 -- CONTROLS THE ROWS

P7DIR = 0XFF;

P3DIR &= ~0x0F; // Input for 3.0-3.3 -- COLUMNS

P3REN |= 0x0F; // Pullup/Pulldown enabled for 3.0

P3OUT |= 0x0F; // Set 3.0 as pullup

while(1){

P7OUT = 0X02; //CHECK ROW 1

if((P3IN & 0X01) == 0){//BUTTON ONE

TA1CCR0 = 956;

\_\_delay\_cycles(250000);

TA1CCR0 = 0;

}

else if((P3IN & 0X02)==0){//BUTTON 2

TA1CCR0 = 851;

\_\_delay\_cycles(250000);

TA1CCR0 = 0;

}

else if((P3IN & 0X04)==0){

TA1CCR0 = 758;

\_\_delay\_cycles(250000);

TA1CCR0 = 0;

}

else if((P3IN & 0X08)==0){

TA1CCR0 = 716;

\_\_delay\_cycles(250000);

TA1CCR0 = 0;

}

P8OUT |= 0X0F; //TURN OFF ALL LEDS CONTROLLED BY P8

P7OUT = 0X01;//CHECK ROW 2

if((P3IN & 0X01) == 0){//BUTTON ONE

P8OUT &= ~0X03;//LED 1 ON

}

else if((P3IN & 0X02)==0){//BUTTON 2

P8OUT &= ~0X05;//LED 2 ON

}

else if((P3IN & 0X04)==0){

P8OUT &= ~0X06;

}

else if((P3IN & 0X08)==0){

TA1CCR0 = 851;

\_\_delay\_cycles(125000\*2);

TA1CCR0 = 676;

\_\_delay\_cycles(125000\*2);

TA1CCR0 = 451;

\_\_delay\_cycles(250000\*2);

TA1CCR0 = 506;

\_\_delay\_cycles(250000);

TA1CCR0 = 676;

\_\_delay\_cycles(250000\*2);

TA1CCR0 = 751;

\_\_delay\_cycles(250000\*2);

TA1CCR0 = 451;

\_\_delay\_cycles(250000\*3);

TA1CCR0 = 506;

\_\_delay\_cycles(250000);

TA1CCR0 = 676;

\_\_delay\_cycles(250000\*2);

TA1CCR0 = 851;

\_\_delay\_cycles(250000\*3);

TA1CCR0 = 568;

\_\_delay\_cycles(250000\*2);

TA1CCR0 = 638;

\_\_delay\_cycles(250000\*1);

TA1CCR0 = 851;

\_\_delay\_cycles(250000\*2);

TA1CCR0 = 1012;

\_\_delay\_cycles(250000\*2);

TA1CCR0 = 568;

\_\_delay\_cycles(250000\*3);

TA1CCR0 = 638;

\_\_delay\_cycles(250000);

TA1CCR0 = 851;

\_\_delay\_cycles(250000\*5);

TA1CCR0 = 638;

\_\_delay\_cycles(250000\*2);

TA1CCR0 = 676;

\_\_delay\_cycles(250000);

TA1CCR0 = 851;

\_\_delay\_cycles(250000\*2);

TA1CCR0 = 1012;

\_\_delay\_cycles(250000\*2);

TA1CCR0 = 1276;

\_\_delay\_cycles(250000\*8);

TA1CCR0 = 1136;

\_\_delay\_cycles(250000\*1);

TA1CCR0 = 1351;

\_\_delay\_cycles(250000\*2);

TA1CCR0 = 1276;

\_\_delay\_cycles(250000\*2);

TA1CCR0 = 1136;

\_\_delay\_cycles(250000\*2);

TA1CCR0 = 1012;

\_\_delay\_cycles(250000\*2);

TA1CCR0 = 902;

\_\_delay\_cycles(250000\*2);

TA1CCR0 = 851;

\_\_delay\_cycles(250000\*2);

TA1CCR0 = 758;

\_\_delay\_cycles(250000\*2);

TA1CCR0 = 676;

\_\_delay\_cycles(250000\*2);

TA1CCR0 = 0;

}

}

return 0;

}

#pragma vector = TIMER1\_A0\_VECTOR

\_\_interrupt void Timer1\_A0\_ISR(void){

P6OUT ^= BIT0;

}

**Example of program snippet that professor gave to help for this specific lab:**

Unsigned int var=123; //1.23

LCD\_write(‘0’+var/100); //1

LCD\_write('.’);

LCD\_write(‘0’+(var/10)%10); //2

LCD\_write(‘0’+var%10); //3

**Example of empty program:**

#include <msp430.h>

/\*\*

\* main.c

\*/

int main(void)

{

WDTCTL = WDTPW | WDTHOLD; // stop watchdog timer

return 0;

}

**Connections:**

MSP430FR5994 Launchpad

16x2 LCD screen

P8.3 connects to RS

P8.2 connects to R/W

P8.1 connects to E

P3.0 connects to DB0

P3.1 connects to DB1

P3.2 connects to DB2

P3.3 connects to DB3

P3.4 connects to DB4

P3.5 connects to DB5

P3.6 connects to DB6

P3.7 connects to DB7

P6.0 connects to the buzzer.

P4.1 connects to a potentiometer.