```c
/* Copyright (C) 2022 MSOE
 *
 * All Rights Reserved
 * You may not use, distribute or modify this code without the
 * express written permission of MSOE
 *
 * Contact Info
 * jorgejuradogarcia2@gmail.com
 * 608-312-5950
 *
 */

#include <stdio.h>
#include "msp432.h"
#include "msoe_lib_all.h"
#include "defines.h"
#include <MSP432P401R_GPIO.h>

//Preprocessors
#define LAB1_ON FALSE
#define LAB2_ON FALSE
#define LAB3_ON FALSE
#define LAB4_ON FALSE
#define LAB5_ON TRUE
#define CLEAR 21


// Lab modules
#if LAB1_ON
#include "EE4930_LAB1.h"
#endif

#if LAB2_ON
#include "EE4930_LAB2.h"
//global variables for Lab 2
unsigned adc_val;
float ADC_Percentage;
float PWM_dutycycle;
int Calculated_dutycycle;
#endif

#if LAB3_ON
#include "EE4930_LAB3.h"
#endif


#if LAB4_ON
#include "EE4930_LAB4.h"
//global variables for Lab4
unsigned adc_val_A0;
unsigned adc_val_A2;
float ADC_A2_Percentage;
float ADC_A0_Percentage;
unsigned char ticks;
eSystemInputs Inputs;
#endif

```

```c


#if LAB5_ON
#include "EE4930_LAB5.h"
//global variables for Lab5
volatile float temp=0;
volatile int adc_value=0;
volatile char Ready = FALSE;
// Defines
#define MAX_12BIT_ADC_RANGE 4096
#endif

/*
 * main.c
 */



int main(void){

#if LAB1_ON
     Init_Lab1();

    while(1)
    {
        Lab1_Poll();
    }
#endif

#if LAB2_ON
    Clock_Init_48MHz();
    Init_Lab2();

    NVIC->ISER[0] |= ( 1<<24 ); //NVIC for ADC14 at ISER[24]
    NVIC->ISER[1] |= ( 1<<3 ); //NVIC for PORT 1 at ISER[35]
    NVIC->ISER[0] |= ( 1<<25 ); //NVIC for TIMER_32_1 at ISER[25]
    __enable_interrupt();

    while(1)
    {
        if(LCD_SET_FLAG == TRUE)
        {

            LCD_goto_xy(5, 3); // start at row 3, column 0
            LCD_print_udec5(adc_val);
            LCD_goto_xy(7, 4); // start at row 4, column 0
            LCD_print_udec5((int)PWM_dutycycle);
            LCD_SET_FLAG = FALSE;
        }
    }
#endif

#if LAB3_ON

    uint32_t counter = 0;
    setup();

    Small_Loops_rolling();

    Calculation_loop();

    div_multi_loop();

    Complex_if_else();

    function_prototype_only();

    if( counter > 0)
    {
        never_used_function_1();
        never_used_function_2();
        never_used_function_3();
        never_used_function_4();
    }

    //set P2.3 to low
    GPIO_setOutputLowOnPin( GPIO_PORT_P2, GPIO_PIN3 ); //Set P2.3 Low
#endif

#if LAB4_ON
    Clock_Init_48MHz();

    //Interrupt setup
    NVIC->ISER[0] |= ( 1<<24 ); //NVIC for ADC14 at ISER[24]
    NVIC->ISER[1] |= ( 1 << 3 ); //NVIC for PORT 1 at ISER[35]
    NVIC->ISER[0] |= ( 1<<25 ); //NVIC for TIMER_32_1 at ISER[25]

    __enable_interrupt();
    Dehumidifier_init();

    eSystemEvent eNewEvent = Last_event;
    eSystemState eNextState  = Off_state;

    while(1)
    {
        if( ticks == TICKS_MAX)
        {
            //read the inputs
            Dehumidifier_Read(Inputs);

            //Run the state machine
            eNewEvent = Dehumidifier_ReadEvent();


#if !EXAM1_SECTION1
            //poll of the state machine
            eNextState = Dehumidifier_Poll(eNewEvent, eNextState );
#else
            eNextState = Dehumidifier_IfElse(eNewEvent, eNextState );
#endif
        }
    }

#endif
```

```c

#if LAB5_ON

    WDT_A->CTL = WDT_A_CTL_PW | WDT_A_CTL_HOLD;

    // set unused pins to pullup/down enabled to avoid floating inputs
    P1->REN |= 0xFF;
    P2->REN |= 0xFF;
    P3->REN |= 0xFF;
    P4->REN |= 0xFF;
    P5->REN |= 0xFF;
    P6->REN |= 0xFF;
    P7->REN |= 0xFF;
    P8->REN |= 0xFF;
    P9->REN |= 0xFF;
    P10->REN |= 0xFF;

    Set_ports_to_out();

    Init_PCM(); //setup for power control module
    init_CS();  //setup for clock system
    //Init the watchdog timer
    Init_Watchdog();

    //Init the Temperature Sensor
    Init_Temp();
    //Interrupt setup
    NVIC->ISER[0] |= ( 1 << ADC14_IRQn ); //NVIC for ADC14 at ISER[24]
    NVIC->ISER[0] |= ( 1 << WDT_A_IRQn); //NCI for Watchdog atISER[3]
    __enable_interrupt();
    ADC14->CTL0 |= ADC14_CTL0_SC; // start a new ADC conversion
    //Processor Peripherals
    //System control register
    SCB->SCR = SCB_SCR_SLEEPDEEP_Msk;

    while(1)
    {
        __wfi();   // wait in LPM3 for watchdog

        while(!Ready);

        Ready = FALSE;
        P4->OUT &= ~BIT5;            // end indication
        printf("Temperature Value: %f F\n", temp);

    }

#endif

} // end main

#if LAB2_ON
// Interrupt Handler for ADC14
void ADC14_IRQHandler()
{
    int readIV = ADC14->IV; // Reading the IV register to clear
    // read A/D result
    adc_val = ADC14->MEM[0];
    ADC_Percentage = 100 * (float)adc_val/1024;
    PWM_dutycycle = 100 - ADC_Percentage;
    //printf("A/D reading = %d, ADC % = %f \n", adc_val, ADC_Percentage);
    //printf("Duty Cycle = %f\n", PWM_dutycycle);
    Calculated_dutycycle = Set_Duty( PWM_dutycycle );

}

// Interrupt Handler for SW1
void PORT1_IRQHandler()
{
    int readIV = P1->IV; // Reading IV register to clear
    // and set a LCD_flag to start to update with new values
    LCD_SET_FLAG = TRUE;

}

//Interrupt Handler for TIMRE32_Int1
void T32_INT1_IRQHandler()
{

    TIMER32_1->INTCLR = CLEAR;
    // start a new A/D conversion
    ADC14->CTL0 |= ADC14_CTL0_SC;
    //printf("starting ADC Timer\n");
}
#endif

#if LAB4_ON

// Interrupt Handler for ADC14
// Will enter into this interrupt source with the Interrupt of the ADC14 was set
void ADC14_IRQHandler()
{
    int readIV = ADC14->IV; // Reading the IV register to clear
    // read A/D result
    switch( readIV )
    {
        case 0x0C:
            adc_val_A0 = ADC14->MEM[0];
            ADC_A0_Percentage = 100 * (float)adc_val_A0/16384;
            //reading humidiity
            Inputs.Humidity_percentage = ADC_A0_Percentage;
            break;
        case 0x10:
            adc_val_A2 = ADC14->MEM[2];
            ADC_A2_Percentage =   (float)adc_val_A2/16384;
            //READING Room Temperature
            // 40-90 degrees so the values can range from 50 points
            Inputs.Room_temperature = (int)(ADC_A2_Percentage*50) + 40;
            break;

    }

}

```

```c
// Interrupt Handler for SW1
void PORT1_IRQHandler()
{
    int readIV = P1->IV; // Reading IV register to clear

    switch( readIV )
    {
    case 0x04: // P1.1
        Inputs.Humidity_setpoints += 5;
        break;
    case 0x0A: // P1.4
        Inputs.Humidity_setpoints -= 5;
        break;
    }

    if(Inputs.Humidity_setpoints > 100)
    {
        Inputs.Humidity_setpoints = 100;
    }

}

//Interrupt Handler for TIMRE32_Int1
void T32_INT1_IRQHandler()
{

    TIMER32_1->INTCLR = CLEAR;
    // Read the Ice sensor pin
    Inputs.Ice_sensor =   ( (P5->IN & GPIO_PIN4) >> 4 );
    // run adc converter
    ADC14->CTL0 |= ADC14_CTL0_SC;

    ticks++;
    if( ticks > TICKS_MAX)
    {
        ticks = 0;
    }

}


#endif


#if LAB5_ON
void ADC14_IRQHandler(void)
{
    int readIV = ADC14->IV; // Reading the IV register to clear
    ADC14->CLRIFGR0 = ADC14_CLRIFGR0_CLRIFG0;
    adc_value = ADC14->MEM[3];
    uint32_t adc_voltage = (( adc_value ) *
                            (3300.0f / MAX_12BIT_ADC_RANGE) ) + 75;
    float temp_c = ((adc_voltage) / 10.0f); // 10mV/C
    temp = ( temp_c * (9.0f / 5.0f) ) + 32;
    Ready = TRUE;

}
```

```c

// Interrupt Handler for the watchdog timer
void WDT_A_IRQHandler(void)
{
    // wakes up from LPM3 into LPM0
    P4->OUT |= BIT5;          // indicate temp reading is available

    //start a new conversion
    //sleep the board
    ADC14->CTL0 |= (ADC14_CTL0_ENC | ADC14_CTL0_SC); // start a new ADC conversion

}
#endif

```