

CSE411  
Real-Time Embedded Systems Design



Team 10

Mahmoud Abdalla Mohasseb	20P2787
Daniel Tarek Lewis	18P1185
Ahmed Amr Behairy	18P5837
Abdelrahman Shemis	18P9565
Abdelrahman Tarek Mahdy	18P7159
Tasneem Osama	18P3208

## Project Description

This project consists of a temperature sensing system that has a potentiometer that delivers its reading to the MCU, The MCU starts to convert the analog signal to a digital signal using the ADC peripheral.

The user enters the Set Temperature, if the temperature read is lower than the input the heater will start (Red LED) else if the temperature read is higher than the input then the heater (Red LED) will not work

The input is entered by the user using keypad

If the user needs to enter a new set point, he must press on the TIVA - C built in button (PF4) to fire a GPIO interrupt and wait for the new user input.

The temperature is displayed of the LCD (2x16 8 bits)

We are using the TIVA-c internal voltage 3,3v (for the ADC configurations and LCD Power ON)

## Free RTOS implemented Tasks

### 1- Task 1 (KEYPAD\_getkey)

- 1- It tries to take semaphore given by GPIOF Handler, if it can't it will be in blockage state
- 2- After taking semaphore it clears past value of set temperature and make total variable which stores values taken from keypad
- 3- It scans rows and columns until getting value of keypad
- 4- When user press \* in keypad that means user finished entering value of set temperature
- 5- It tries to take mutex of set temperature to edit global variable with new value

### 2- Task 2 (ADC\_readMeasuredTemp)

- 1- First it tries to take mutex of global variable set temperature to read it and store it in local variable
- 2- It reads ADC value and convert it to temperature by dividing it by 80
- 3- It clears flag of ADC then write measured temperature to lcd
- 4- It checks if measured temperature is less than set temperature and state is more than set value, if it is, it changes state to 0 and give semaphore to RED led task to toggle
- 5- It checks if measured temperature is more than set temperature and state is less than set value, if it is, it changes state to 1 and give semaphore to RED led task to toggle
- 6- It checks if measured temperature is more than alarm temperature. If it is, it gives semaphore to buzzer task to start blink

### 3- Task 3 (redTask)

- 1- It tries to take semaphore given by task 2
- 2- If got semaphore it will toggle red led

### 4- Task 4 (buzzerTask)

- 1- It tries to take semaphore given by task 2
- 2- If got semaphore it will turn on and green led

**Files of Project:**

Lcd.c, lcd.h: these files contain functions used to initialize LCD, write data, write string and write numbers on LCD

Delay.c, delay.h: these files contain function used for delay

Keypad.c, keypad.h: these files contain function to initialize keypad

Led.c, led.h: these files contain function to initialize red led and green led

Pot.c, pot.h: these files contain function to initialize ADC channel 2

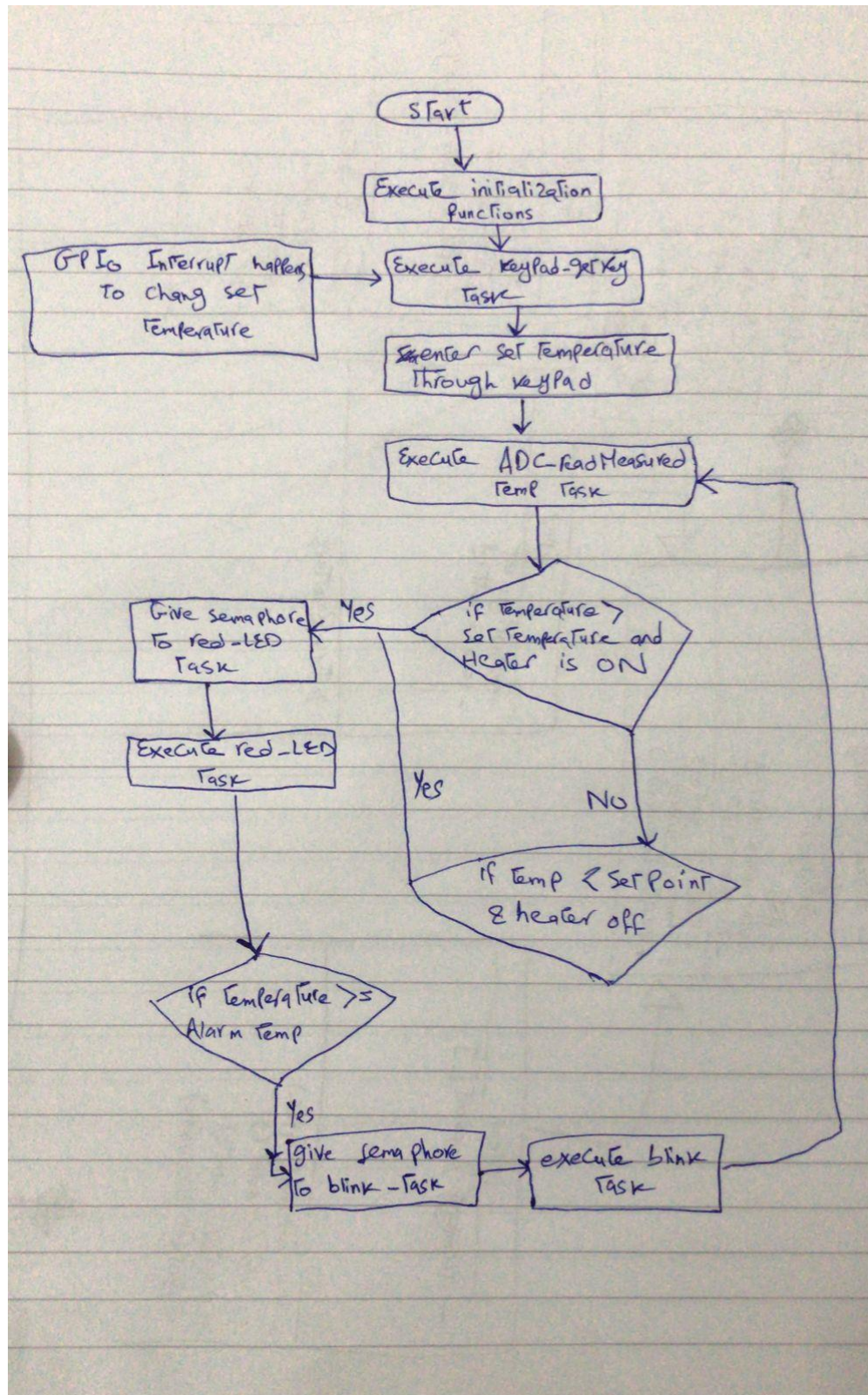
**Connections of Pins:**

ADC: PORTE PIN 1

LCD: PORTB PIN0 – PIN7 for Data, PA7: EN, PA6: R/W, PA5: RS

Keypad: PORTC PIN4 – PIN7 for columns, PORTE PIN2 – PIN5 for rows

## Flow Chart:



**Drive link:**

[Code + Video](#)