mbed C Programming Assignment
Wireless Sensor Networks

Akhil Kumar Donka

January 2022

1 Introduction

Aim is to understand C Programming on mbed simulator with help of sensor and miscellaneous components. This is followed by execution of below mentioned instructions on mbed microcontroller.

- To the given board, connect the following components:
  - an sht31 temperature and humidity sensor
  - a red LED.
  - a blue LED.
  - a white LED.
  - a button.
- The system should perform the following tasks continuously.
  - White LED should be blinking every 2 seconds. (0.5 seconds on time, 2 seconds off time)
  - Read temperature and humidity values every 10 seconds (not continuously). The intensity of red LED and blue LED should change depending on temperature and humidity values, respectively. (higher the value higher intensity the corresponding led should have)
  - On button press, take new readings immediately, and update both LED intensity.

2 Explanation of steps to carried out

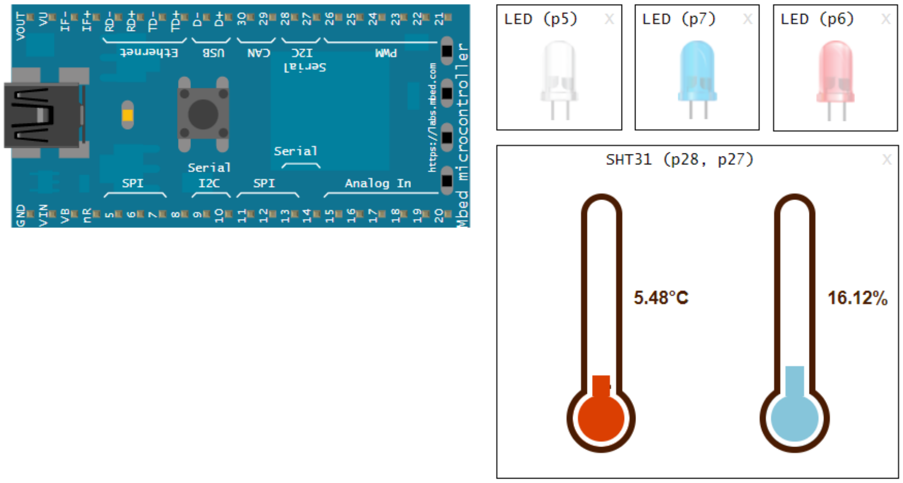

Figure 1: mbed component setup

All the necessary components are placed in mbed simulator to start with. 3 Different LEDs of type ”PwmOut” are initialized on pin p5, p6, p7. Also one input button is initialised to send interrupt command. To multithread our operations, we initialize two tickers t1 & t2.

Then we define two functions to carry tasks of blinking white LED and temperature & humidity monitoring. In case of white LED blinking, we code such that LED gets ON and stay as it is for 0.5 seconds and turns backs OFF. Second task is to vary intensities of LEDs (which take PWM inputs between range of 0 to 1) according to variations in temperature. Since temperature and humidity ranges in sensor 0-50 & 0-100 respectively, we accordingly normalize the temperature values to scale it down between 0-1. And then normalized values are given as PWM input to LEDs. We do updates in intensities as per the timing intervals given.

Ticker t1 handles the task of calling white LED blinking function i.e. blink_white() at every 2.5 seconds and Ticker t2 handles the task of sensor monitoring by calling temp_intensity() function at every 10 seconds. These both tickers run in parallel and whenever button is pushed, temp_intensity function gets again called and executed.

3 Coding

```
#include ”mbed.h”
#include ”Sht31.h”
```

```
Sht31 sht31(I2C_SDA, I2C_SCL);
PwmOut blueled(p7);
PwmOut redled(p6);
PwmOut whiteled(p5);

InterruptIn btn(BUTTON1);
Ticker t1;
Ticker t2;

void blink_white(){
    printf(”White LED Blinking\n”);
        whiteled = 1;
        wait_ms(500);
        whiteled = 0;
}

void temp_intensity(){
    printf(”Temperature & Light Intensity Control\n”);
    float temp = sht31.readTemperature();
    float humidity = sht31.readHumidity();
    redled = (temp/50);
    blueled = (humidity/100);
}

int main() {
        t1.attach(callback(&blink_white), 2.5f);
        t2.attach(callback(&temp_intensity), 10.0f);
    btn.fall(callback(&temp_intensity));
        wait_ms(osWaitForever);
}
```