**Temperature Detection System with web integration**

Salman Albarri, Mahmood Alhawaj, Ali Aladraj

202007984,202010520,202002083

Department of Computer Engineering, University of Bahrain

ITCE331: Embedded Systems

Dr. Ali Hassan

Table of Contents

[Objectives 1](#_Toc135914592)

[Introduction 1](#_Toc135914593)

[Design and implementation 1](#_Toc135914594)

[Setup function 7](#_Toc135914595)

[1. Wi-Fi setup 8](#_Toc135914596)

[2. Server setup 9](#_Toc135914597)

[3. Server handle 11](#_Toc135914598)

[4. Display print 11](#_Toc135914599)

[5. Sensor read 12](#_Toc135914600)

[6. Output trig 13](#_Toc135914601)

[Simulation 13](#_Toc135914602)

[Conclusion 14](#_Toc135914603)

# Objectives

* To measure the temperature using a temperature sensor.
* To print the temperature value on an OLED screen.
* To trigger an output based on trigger value.
* To be able to view the temperature and modify the trigger value from a website.
* To achieve the previous tasks using FreeRTOS.

# Introduction

The System will take the reading of a temperature sensor as an input, and it should trigger an output if the temperature exceeded a certain degree. This degree can be modified by accessing a local website that will also show the current temperature and the current trigger value. Also, the system should print the current temperature, the trigger value, and the IP address of the local server on an OLED screen.

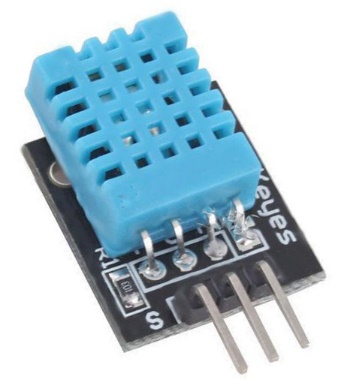
# Design and implementation

Components needed:

* ESP32



* Temperature sensor



* 0.96” OLED screen

A picture containing electronics, electric blue, electronic device, circuit component

Description automatically generated

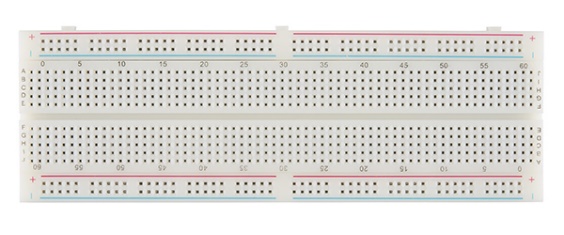
* LED light instead of a fan



* Wires



* breadboard





HTML code



Web page view A screenshot of a computer

Description automatically generated with medium confidence

## Setup function



The setup function will set the output pin mode, Create the trigger value queue and the temperature value queue then it will set initial values to them, and will define the 6 tasks then it will send a notification for the Wi-Fi setup task, sensor reading task, and the output trigger task.

We divided the project into several tasks:

## Wi-Fi setup

A picture containing text, screenshot, software, multimedia software

Description automatically generated

Task one will set up the Wi-Fi connection. At first it will wait for a notification to begin connecting to the Wi-Fi with the help of Wi-Fi library and the SECRET\_SSID and SECRET\_PASS which is defined in a file called head.h then after connecting to the Wi-Fi successfully it will print its local IP address and will give the server configuration task and the Printing to display task a notification. Finally, it will suspend itself.

## Server setup

A picture containing text, screenshot, software, multimedia software

Description automatically generated

The second task will be for setting up the web server. It will begin after it receives a notification from the Wi-Fi setup task. It will define the available web pages in the server and will route them to specific functions.

A picture containing text, screenshot, font

Description automatically generated

If the user selected the root directory from the browser, it will display the html page, else it will send a Not Found text. The update is the page that the user will be directed to when he changes the trigger value. It will read the user input and print it in the serial connection, and it will overwrite it to the trigger value queue and then it will send the main html page to the client.

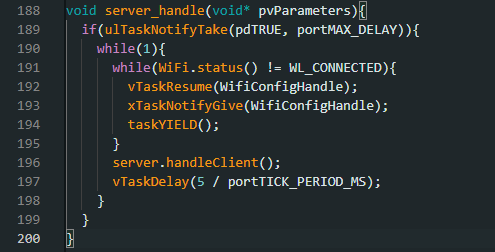
A screen shot of a computer program

Description automatically generated with medium confidence

my\_html() is the function that has been used previously to return a String value of the html page to the server functions. It reads the values stored in both trigger value queue and the temperature value queue, then it reads the html file that is stored in the flash by using ESP32 sketch data upload Arduino IDE tool with the help of the SPIFFS library then it will replace the placeholders with the data that have been read from the both of the queues and send it to the server handling task.

After finishing the setup of the server, it will print “server Online” and will send a notification to the server handling task then it will suspend itself.

## Server handle

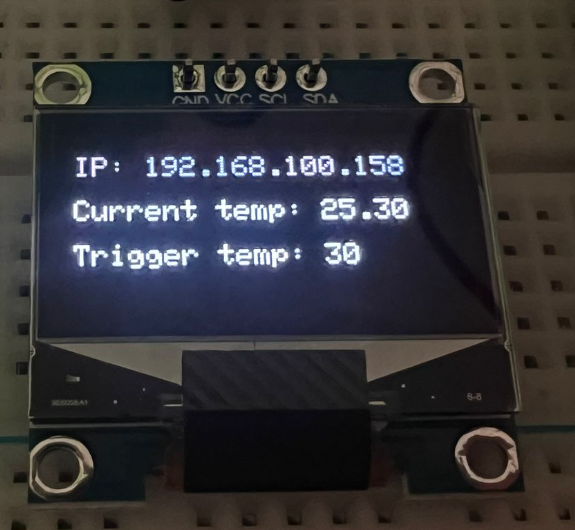


The server handling task is a simple task that will wait for the server configuration task to finish and send a notification then it will start in a loop every 5ms it will check if the Wi-Fi connection is connected then it will handle the client if there is any request. If the Wi-Fi connection is lost it will resume the Wi-Fi setup task and give it a notification, then it will request a context switch to switch to the Wi-Fi setup task.

## Display print



This task is associated with the OLED screen, as the one shown above. So, if the task notification is 1(the one in the Wi-Fi setup) meaning that the Wi-Fi has been set properly, the task will start. The task will fetch the current temperature as well as the trigger temperature, each from its corresponding queue. The OLED screen will clear the current display and print the new values, as well as printing the current IP for the esp. after printing, it will delay for 500ms and redo the printing process again.

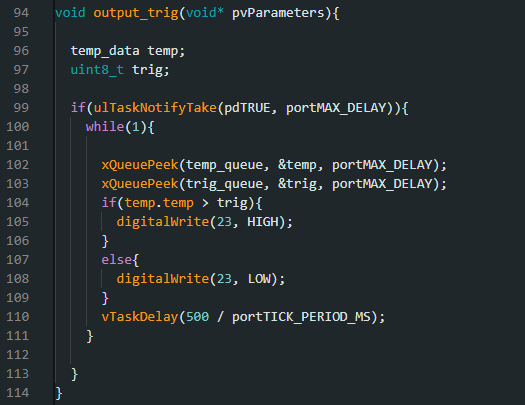


## Sensor read



In this task, the microcontroller will wait for the Wi-Fi setup notification (just like the previous tasks), after that it will read periodically(250ms) the current temp using the DHT sensor (that was initialized in the setup phase), and it will print it to the serial terminal. most importantly, the current temp will be sent to the queue.

## Output trigger

****

The main purpose of this task is to turn on the LED (on port 23) if the temp is higher than the trigger temp, and turn off else, after that delay for 500ms. Where both the trigger temp and the current temp will be fetched from the queue after the Wi-Fi has been set properly.

# Simulation

A picture containing electronics, electronic engineering, electrical wiring, circuit component

Description automatically generated

Since the trigger temp is lower than the current temp, the led light won’t turn on.

A picture containing electronics, electrical wiring, electronic engineering, cable

Description automatically generated

However, in this example the trigger temp is lower than the current temp, so the led was turned on.

A screen shot of a black screen

Description automatically generated with low confidence

The web page showing the current temp as well as the trigger temp, and offering the option to change trigger temp.

# Conclusion

In conclusion, the system described is designed to monitor temperature using a sensor as input and trigger an output if the temperature exceeds a certain degree. The trigger value can be modified by accessing a local website that also displays the current temperature and trigger value. Additionally, the system displays the current temperature, trigger value, and IP address of the local server on an OLED screen. This system provides a convenient and efficient way to monitor temperature and adjust trigger values in real-time, making it a valuable tool for various applications that require temperature monitoring. These functionalities have been implemented successfully using FreeRTOS APIs.