ECE Laboratory

**DREXEL UNIVERSITY**

**To: Dr. Peters**

**From: Ehi Simon**

**Re: ECE 304 Lab 4 - JSON, APIs, and Thunder Client**

**PURPOSE:**

The purpose of this week’s lab is to understand how JSON, APIs, and Thunder Client work. The lab is supposed to help build these skills especially how they interact with the ESP32 to build a webpage and to control a circuit.

**Discussion:**

**A picture containing electronics, electronic engineering, computer component, computer hardware

Description automatically generated**

*Fig. 1. Circuit Connection for Project 2*

The circuit for the lab was built like the one above. It consists of 2 330 resistors, a red LED, a blue LED, a BME280 Environmental Sensor, an Adafruit BNO085 IMU, and an ESP32S microcontroller.

**Main.cpp**

In my main.cpp file, I initialized the multiple libraries that were needed for the sensors to work and provide readings. I also included libraries to get the ESP32 to connect to STA, and libraries needed to interact with a server in JSON. I had to install the Adafruit JSON library prior. The LEDs are defined, the sea level reference pressure is defined and the BNO08X chip is reset. The BME object is created and the network SSID and password. The web server is then opened on port 80. This can all be found in the figure below:



*Fig. 2. Figure Showing Initialization of Libraries and Variable Definitions*

A picture containing text, screenshot, software

Description automatically generated

*Fig. 3. Figure Showing Setup for Yaw, Pitch, and Roll Values*

The figure above shows the setup to obtain the yaw, pitch, and roll. The BNO08X chip is reset, and all the necessary initializations and calculations are performed to obtain the values.

In the figure below, I modified the handle\_OnConnect function to include the yaw, pitch, and roll values in the doc under the JSONVar class. This class also contains the temperature, humidity, altitude, and pressure taken from the BME 280 sensor.

A screen shot of a computer program

Description automatically generated with low confidence

*Fig. 4. Figure Showing Handle\_OnConnect Function*

A picture containing text, screenshot, software

Description automatically generated

*Fig. 5. Figure Showing Handle\_led\_change Function*

The figure above shows the Handle\_led\_change function. This function does almost the same as the previous one, but this one has the ability to toggle the blue LED off. The doc class is also updated and has the states of the LED shown on the webpage. Red is set to on because it is not controlled by the user, and blue is set to LED\_status, which changes every time the user loads the webpage.

The figure below shows the LED value post and get functions. This function allows us to change the LED value on the webpage and the circuit by using post and get commands. They also return values obtained from the sensors on the circuit.

A screen shot of a computer program

Description automatically generated with low confidence

*Fig. 6. Figure Showing POST and GET Functions*

A screen shot of a computer program

Description automatically generated with low confidence

*Fig. 7. Figure Showing Setup and Loop Functions*

The figure above shows the setup function of the file. This setup function does more than those in previous projects. It sets the LED pin modes. It also initializes the BME280 sensor and the BNO08X chip. It then connects to the local Wi-Fi network and turns on the server. The setup function makes it possible to open the webpage and GET or POST from the page. The loop function simply keeps handling the client.

The final result consists of multiple webpages and changes from GET/POST requests. They can be found in the figures below.

A screenshot of a computer

Description automatically generated

*Fig. 7. Figure Showing GET request made to webpage*

A screenshot of a computer

Description automatically generated

*Fig. 8. Figure Showing POST request made to webpage*

A screenshot of a computer

Description automatically generated

*Fig. 9. Figure Showing POST request made to webpage led\_set\_post*

A screenshot of a computer program

Description automatically generated with medium confidence

*Fig. 10. Figure Showing Python Output After Changing Intensity and LED Status*

A screenshot of a computer

Description automatically generated with medium confidence

*Fig. 11. Figure Showing Output on Webpage on Chrome*

A screenshot of a computer

Description automatically generated with medium confidence

*Fig. 12. Figure Showing Output on Webpage ledchange on Chrome*

**Conclusion**

In this experiment, I learned about GET and POST requests. I also understand JSON and APIs more than before.