**ITCS446 Embedded Systems and Applications**

**Project 2:**

RPI and ESP32 for Humidity and Temperature Monitoring Utilizing DHT10 to Control Temperature with DCMotor

**Group K:** Peerapat Potch-a-nant 6088126, Sunat Praphanwong 6088130

**Objective**

To measure humidity and temperature and monitor them overtime.

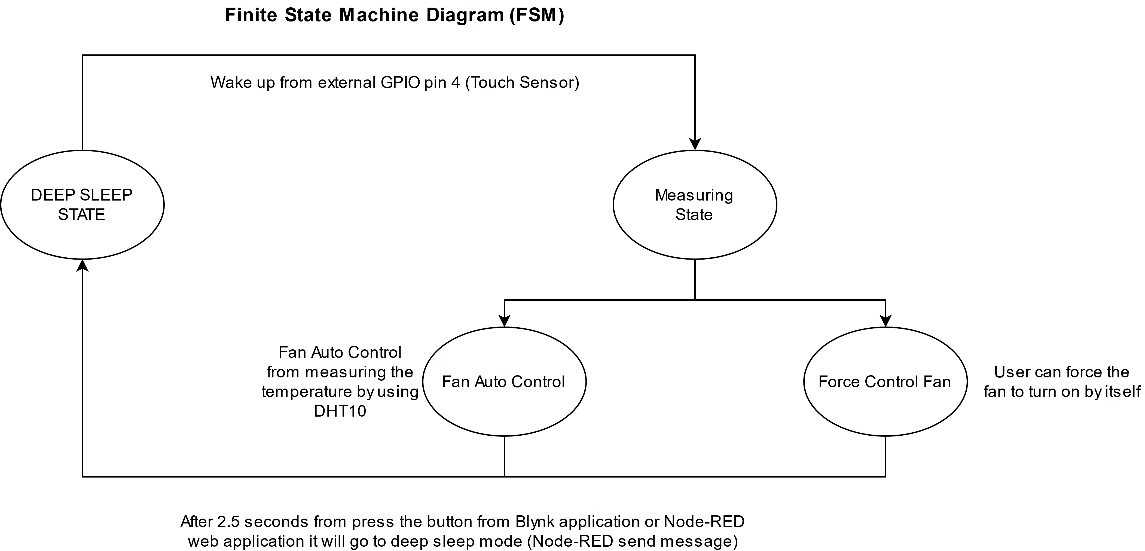
To control temperature in certain area using DCMotor to generate airflow

**Scenarios**

When there is in need for temperature and humidity monitoring over a period of time, and you are not available in the area for several hours. This creation allows you to measure the humidity and temperature on your phone using Blynk application or Node-Red web application. Moreover, you can control the DCMotor to on, off or auto state to control the temperature around the area.

For example, if you have owned a pet, espscially dogs that can’t live in hot climates, you can use this device to monitor the temperature of the room and control the fan from anywhere.

**Project Design**



The program has four state

1. **Deep Sleep State:** Arduino ESP32 is able to cut-off the power to reduce the power consumption from other sensors.
2. **Measuring State:** Read the Humidity and Temperature from the DHT10 sensor then display the value on Node-RED and Blynk application also display the light on RGB LEDs which Blynk application and Node-RED web app can send the command to get the ESP32 to go to deep sleep state.
3. **Fan Auto Control:** ESP32 will adjust the fan speed automatically by depends on the temperature surrounding.
4. **Force Control Fan:** The user is able to turn on the fan with maximum speed by using Blynk or Node-RED web application.

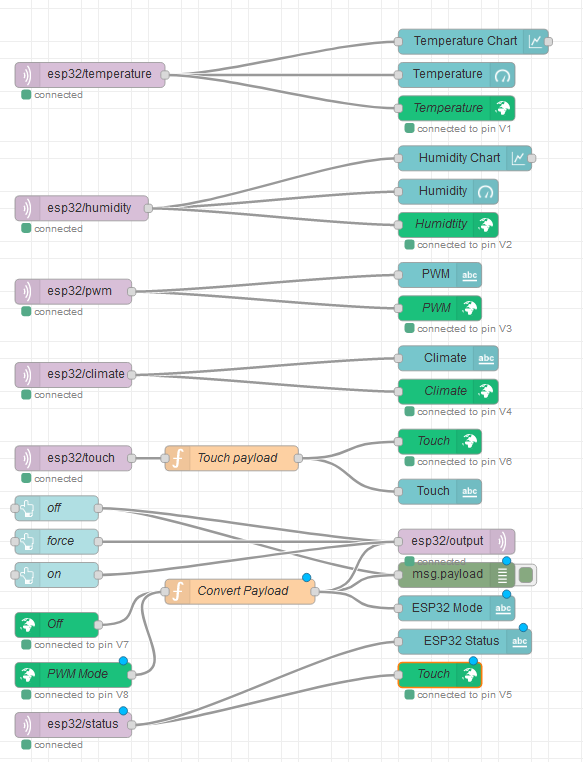
**State Transition**

From deep sleep state to Measuring State: Wake up by using the external wake up from GPIO pin 4 (Touch sensor)

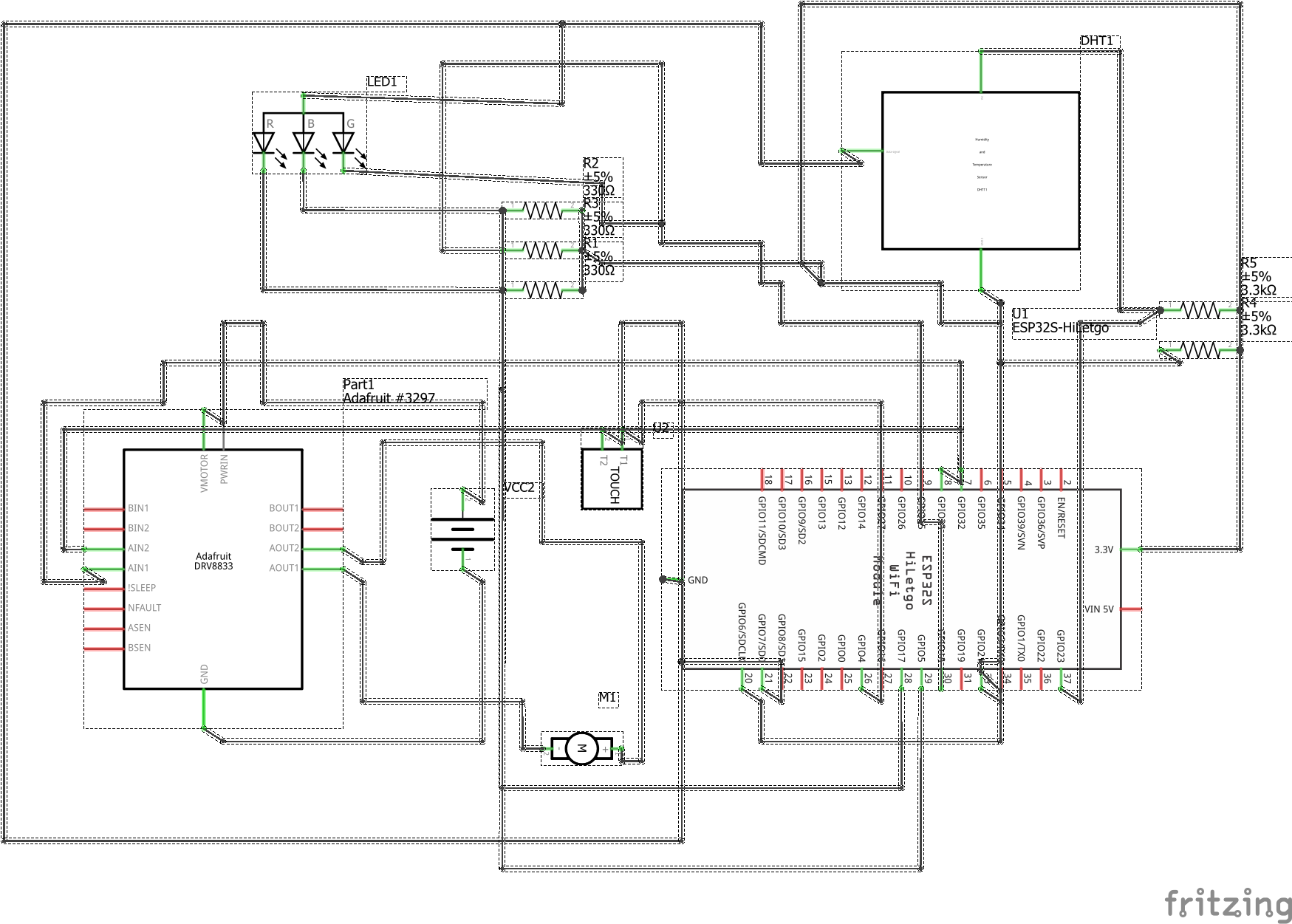
From Measuring State to deep sleep state: After 2.5 seconds pass from sending the message from Blynk or Node-RED web app will change the state

From Fan Control State which depends on the user to override by using “Force Control Fan” or “Fan Auto Control” if the user decided to use “Fan Auto Control” ESP32 will adjust the value by using the temperature surrounding the DHT10 sensor and ESP32 if the user decided to use “Force Control Fan”, the user can press the button to override the value to maximum speed.

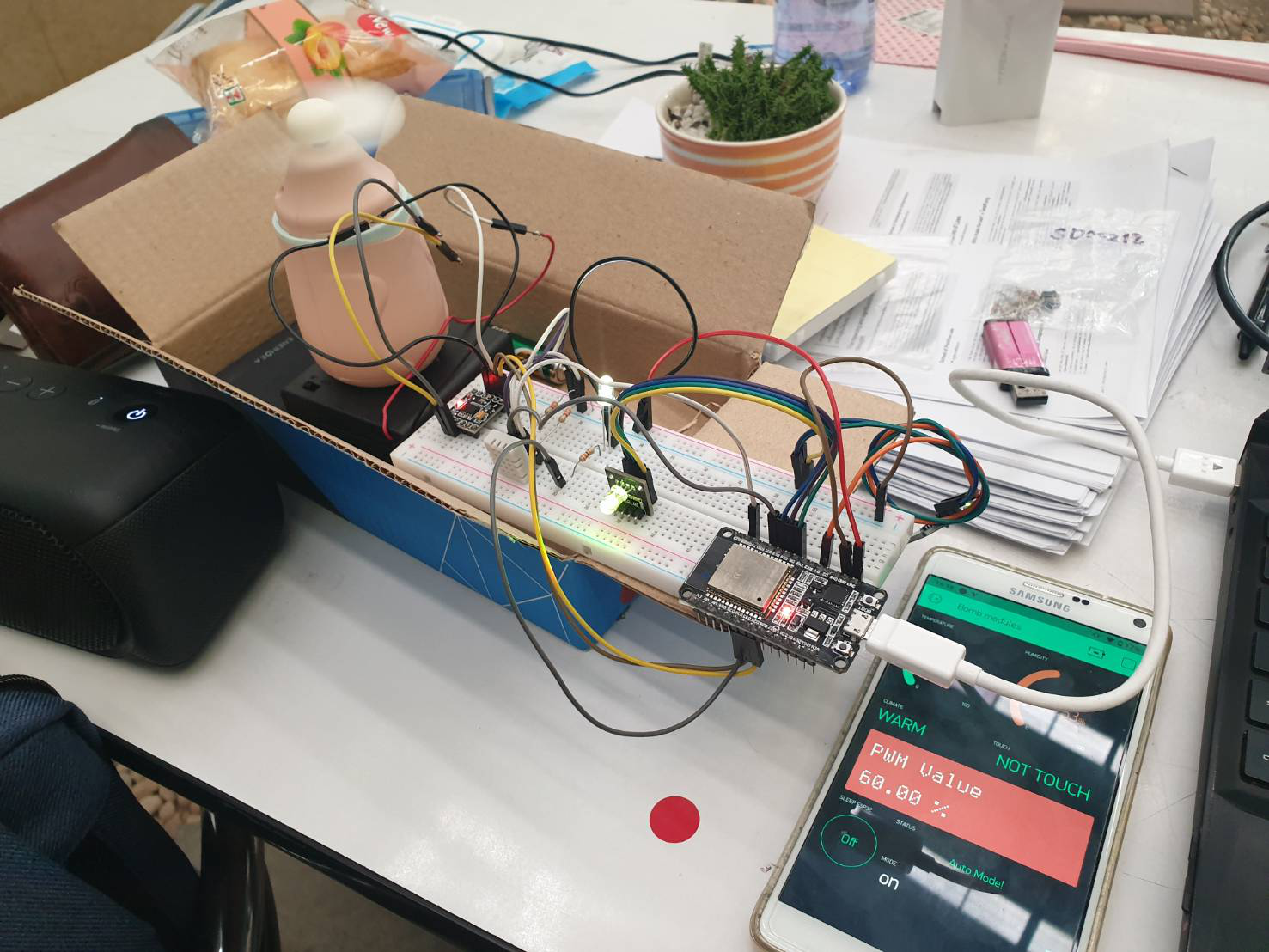
**Node-RED Schematic**

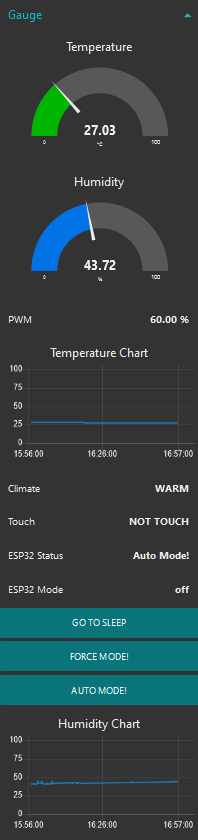
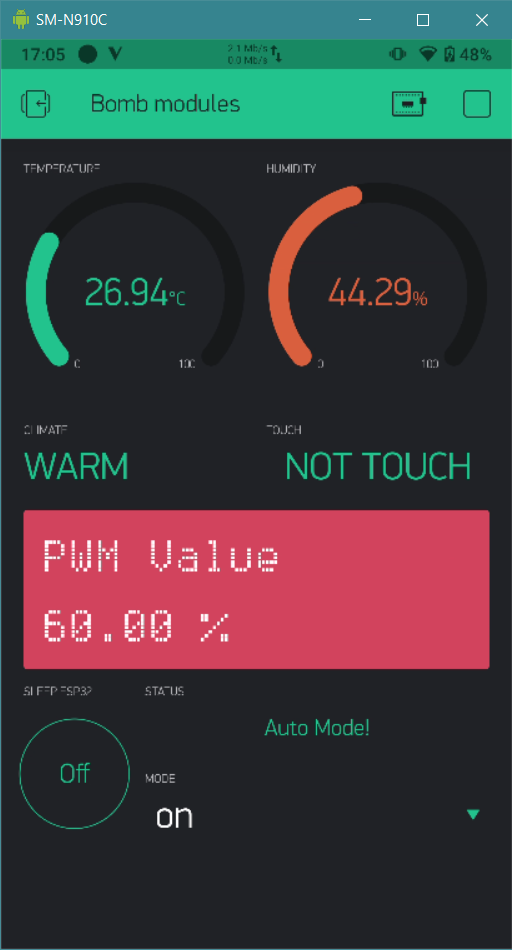


**Circuit Schematic**

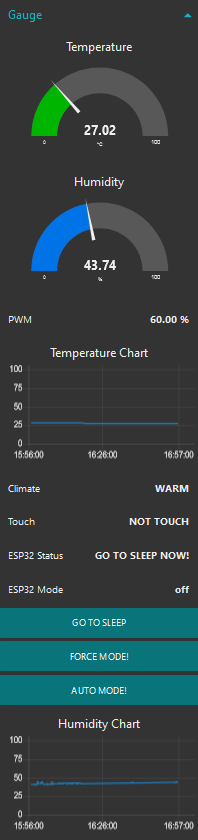
Components: Arduino ESP32, DRV8833, Touch Sensor, LEDs RGB, DHT10, White LED, 1 x Resistor 330 Ohm, 2 x Resistor 3.3 K Ohm, DC Motor, Wire Cable

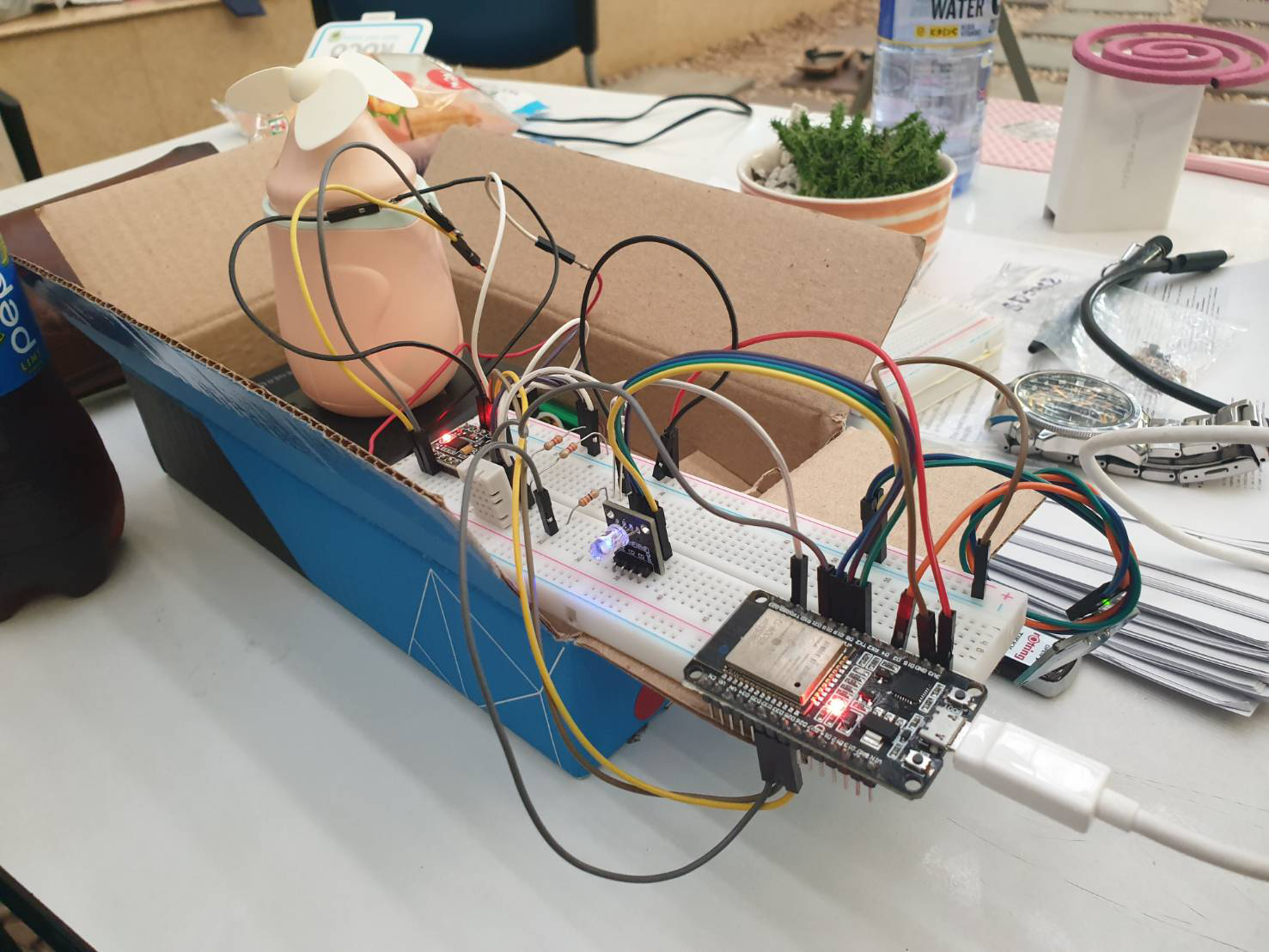
**Project Result**

1. After power on and start Blynk and Node-RED server on Raspberry Pi 4, ESP32 is ready to measure the temperature and the humidity and then send these values using MQTT protocol to Node-RED. Then these values will be displayed in the Node-RED web application and Blynk application

1. The user can decide to get the ESP32 change the state to deep sleep mode by Blynk application or Node-RED application



1. The user can force to turn the fan on at maximum speed by using Blynk application or Node-RED web application

