

EN3240: Embedded Systems Engineering

Assignment 2 — IoT Project

Name: John Doe
Index No: XXXXX

December 17, 2022

This is an individual assignment!
Due Date: 25 September 2022 by 11.59 PM

Instructions

Follow the steps given below. Submit the node-red flow as XXXXX.json and Arduino code as XXXXX.ino to Moodle, where XXXXX is your index number. This assignment accounts for 20% of your final grade.

Part I: Weather Application (12 marks)

Build a node-red application to get the weather data in an user-provided location from OpenWeather (A sample of the Dashboard is shown in Figure 2). It should have the following features:

- The dashboard should have three main tabs:
 1. Weather
 2. Charts
 3. Settings
- Weather Tab
 1. The user should be able to enter the location as i) coordinates or ii) city and country through a text field(s) and submit.
 2. The title of the description section should be the location entered by the user. If coordinates are used to specify the location, retrieve the city using OpenWeather.
 3. Show temperature, humidity, pressure, wind speed, wind direction, sunrise time, sun setting time of the current day. These data should be updated every 30s by default.
 4. Use suitable customized gauges for the above parameters (e.g., a compass for wind direction).
 5. There should be a speaker icon, which, when clicked, would enable the user to hear the description of the weather.
 6. There should be a warning limit for a selected parameter (e.g., temperature, pressure, ...). Use a dropdown list to choose the parameter and a slider to set the limit. (e.g., 25°C for temperature, 30 km for wind, etc.). When this limit is exceeded, a notification should appear as a warning displaying the current value of the parameter (e.g., Warning! Temperature is 30°C . Exceeds the limit by 5°C)
- Charts Tab
 1. There should be charts that show the temperature, humidity, wind speed of the past hour.

- **Settings Tab**

1. The user should be able to select whether the location is entered as i) coordinates or ii) city and country.
2. The user should be able to change the refresh time (use numeric palette).
3. Use a switch to turn the charts feature on and off.

Expected three tabs of the dashboard are illustrated in Figure 2. **Submit the node-red flow to Moodle.**

Part II: Weather Warning Device (8 marks)

Extend your Weather Application to include a Weather Warning Device implemented with the NodeMCU. Link the above application to your NodeMCU via MQTT. The complete system is illustrated in Figure 1.

- Blink the built-in LED of the node-MCU in a warning situation. The threshold/limit for the warning situation is set as described in Part I.
- Change the blinking interval from 5s to 500ms according to the exceeded amount. For an example, 5s interval if the value is just above the warning limit and decreasing up to 500ms interval if it reaches a maximum value. You may decide what the maximum value is.
- Use test.mosquitto.org/ as the MQTT broker.

Submit the resulting Arduino code to Moodle after verifying that it works properly with the hardware.

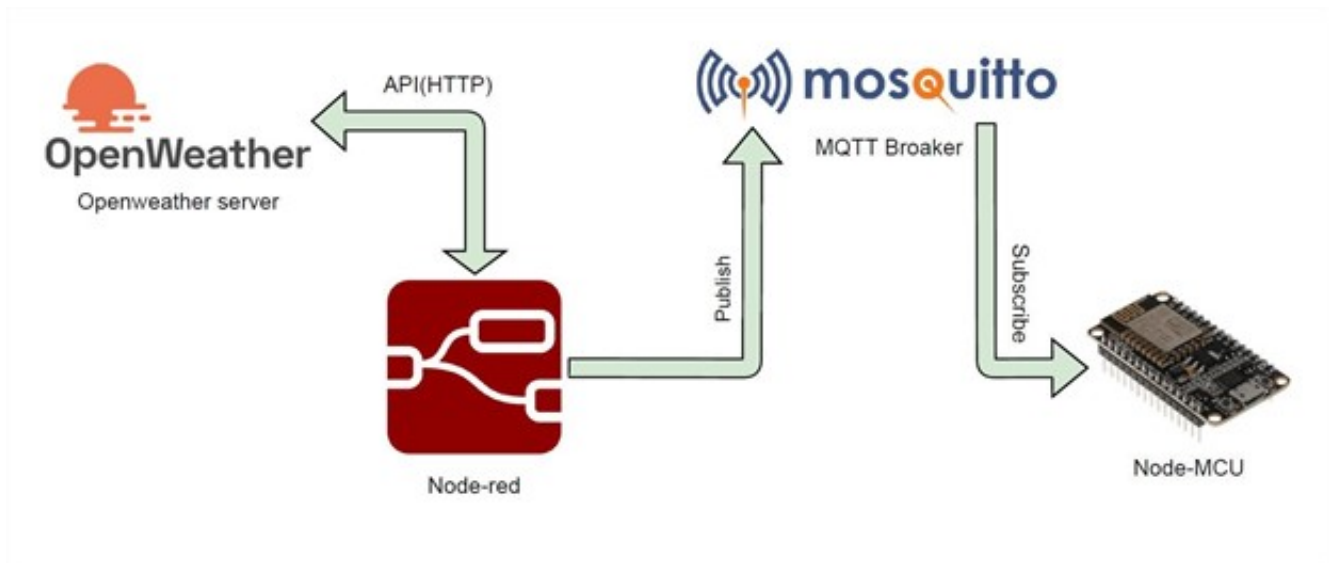


Figure 1: Overview of different components.

Useful Links

- A demo of the dashboard can be found [here](#).
- An overview of the weather API can be found [here](#).
- A NodeMCU Simulator can be found here [here](#). (If you are unable to find a NodeMCU)

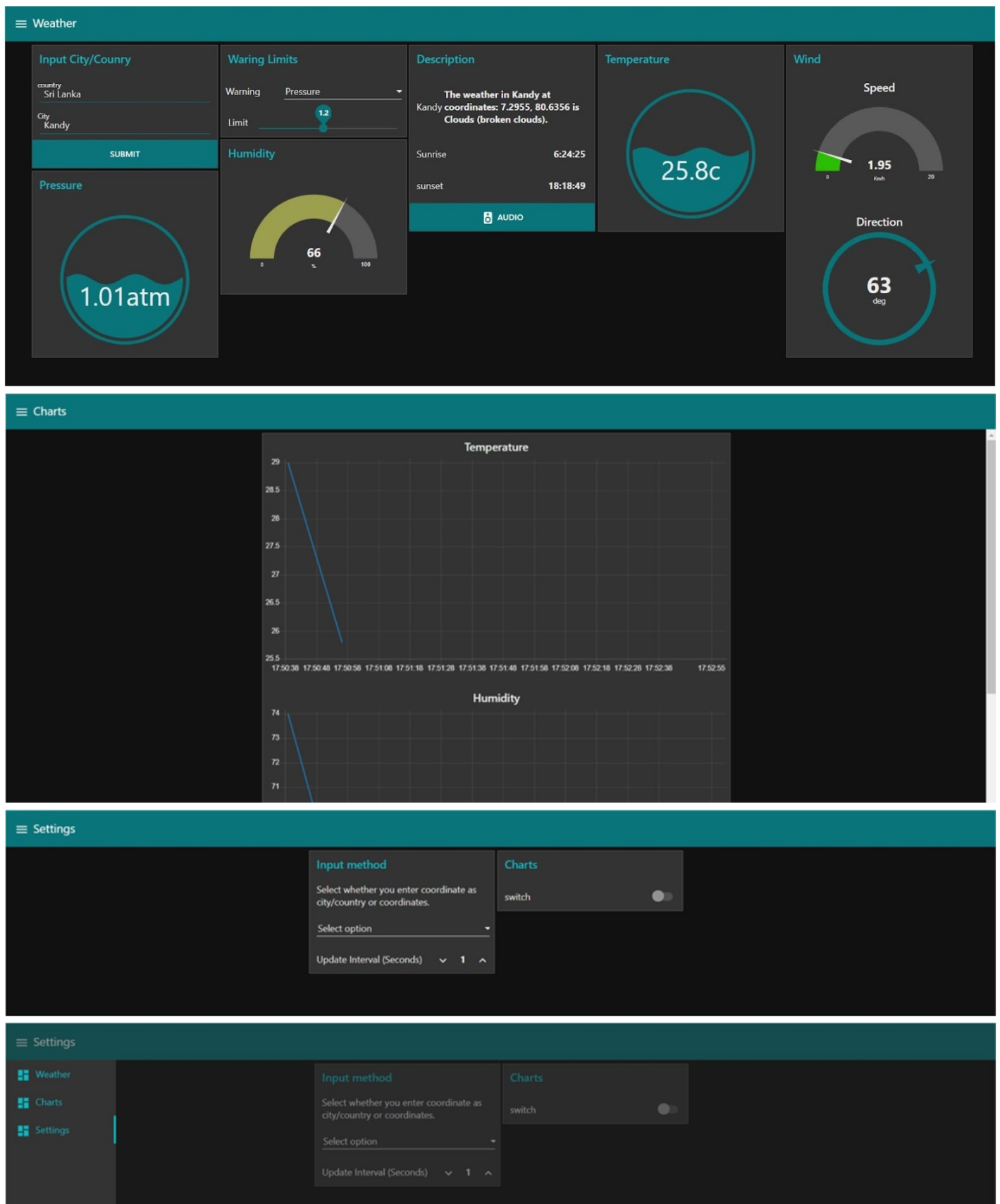


Figure 2: Expected three tabs of the dashboard.