Ambient Condition Monitoring Information System

System Test Plan

By Christopher Sia

System Block Diagram

The System Block Diagram for the Ambient Condition Monitoring Information System (submitted in earlier assignment) is shown in the following. The System Test Plan covers the following test:

Hardware Components

a. DS18B20 Temperature Sensor

b. DHT11 Humidity Sensor

c. Raspberry Pi

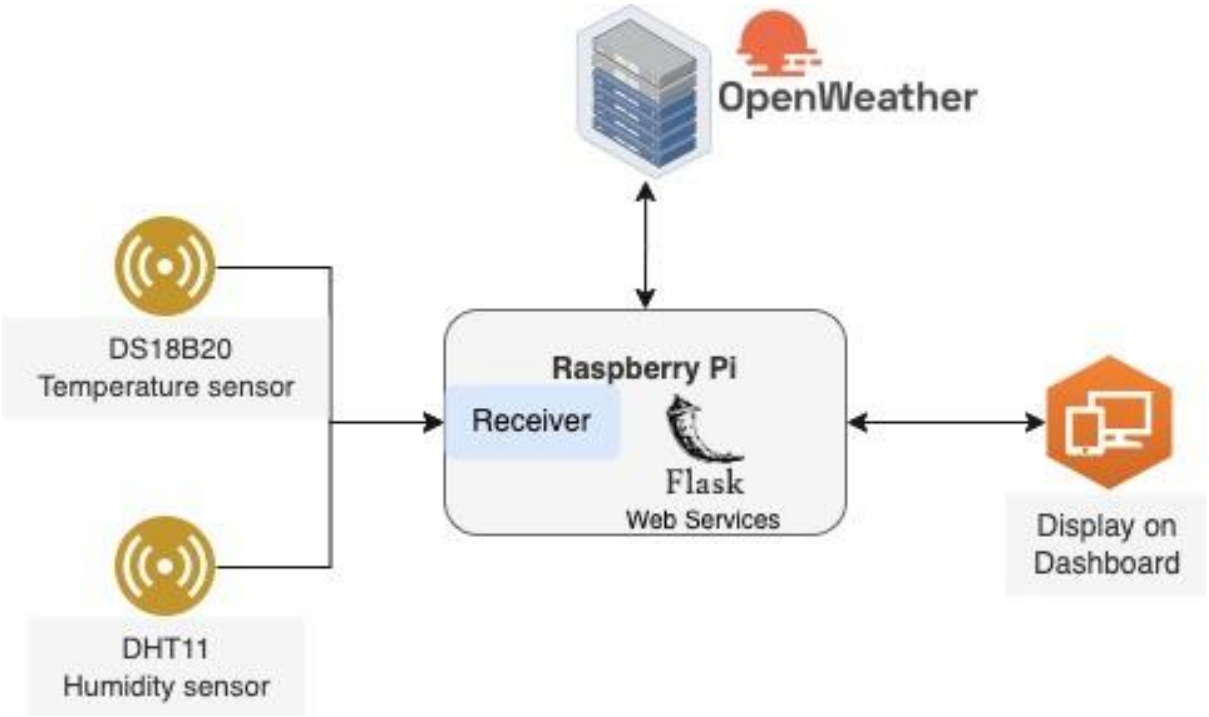
Software Components

a. Receiver Module

b. Flash Web Services

System Level testing

a. End-to-end workflow testing



Hardware Component Test Plan

a. Power up Raspberry Pi and ascertain it can be accessed via SSH connection or VNC connection. Ensure that the Raspberry Pi is able to access the Internet by going to Google or access the URL https://openweathermap.org/

b. Connect DS18B20 to GPIO4 on a breadboard and with 3.3V power supply. Write a Python script to enable GPIO4, read the data from DS18B20 sensor via GPIO4 and print the results on the system console. Compare the printed value against the reading of another known temperature sensor for validation (and calibration).

c. Connect DHT11 to GPIO17 on a breadboard and with 3.3V power supply. Write a Python script to enable GPIO17, read the data from DHT11 sensor via GPIO4 and print the results on the system console. Compare the printed value against the reading of another known humidity sensor for validation (and calibration).

Software Component Test Plan

a. Write a Python script to test the Receiver module to retrieve the data from DS18B20 and DHT11. Validate that the data returned by the Receiver module by checking against readings reported from other validated sensors.

b. Test the Web Services running on Raspberry Pi if it connects to the Receiver module and returns the data received by the module.

c. Test the Web Services running on Raspberry Pi to see if it can call the *openweathermap* API and it is able to retrieve the expected API response (refer to this link).

d. Test the alert threshold on the Web Service for the temperature and humidity level. Test that the alerts are sent when a higher than threshold values for temperature and humidity are received.

e. Test the display on the dashboard that shows the correct readings returned by the Web Services.

f. Test the display on the dashboard is able to show historical records on the temperature and humidity readings in a form of time series graph.

Integration Test Plan

The integration test plan is to test the end-to-end flow from the sensors to the dashboard and from the dashboard to the sensors.

a. Power up the entire system and go to the system dashboard on a local PC. The dashboard must show without any API response errors. This ensures the connection between the dashboard and the web service on the Raspberry Pi is working.

b. Observe the dashboard for the temperature reading and humidity reading. Try to artificially change the temperature and humidity level around the sensors to observe if the readings are updated on the dashboard. This test ensures the connection from the sensors to the web service and to the dashboard is working.

c. Observe the dashboard for the readings from the *openweathermap* API. The expected readings to display on the dashboard are:

- MIN and MAX atmospheric temperature

- Weather and cloud conditions
- Wind speed and direction
- Atmospheric pressure
- Sunset and sunrise timing

d. Enter a new alert threshold on the web service from the dashboard for the temperature and humidity. Artificially change the temperature and humidity conditions around the sensors to cause the sensor readings to go above the threshold. Check if the alerts are displayed on the dashboard. Adjust the temperature and humidity conditions back to lower than threshold settings and check if the alerts are no longer there.

e. Disable the Internet access. Verify that the temperature and humidity sensors are still working and sending updates to the dashboard. However, there is NO readings from the *openweathermap* API.