PROBLEM 6

April 19, 2020

Aditya C
Ayush S
Atreya M
Chandran N
Harsh Jain
R Dhakad
Udbhav B

Contents

0.1 Objectives

The aim of the problem was to design a system to measure Air quality and control an alarm on drastic change of values and measure Temperature and Humidity and correspondingly control AC devices. A Mobile and Web application were created to display the information collected.

Objectives over the weeks	
Date	Objectives
7/4/2020	T1 - Found sensor and will try to send data to cloud using MQ-2 sensor T2 - we have to combine 4 codes. Reading temperature, PID control, IR control and Sending data to cloud. All acquired, need to integrate them. T3 - The front end is almost done. The back-end is still to be done for the web app. Learning Node and Express js. The database we will use is SQL database.
9/4/2020	T1- Working on IOT hub - 25% T2- Compiling the codes found, working on integrating them all. T3- Linked front end with back end - 25%
10/4/2020	T1 - Working on sending data to IOT hub T2 - Need to run and implement the IR and PID codes we found T3 - Learning PostgreSQL for database part -35%
11/4/2020	T1 - Learning from the examples how to send data to cloud. T2 - Combined the codes for Temperature, IR and PID. The code complies properly. Need final code to send the same data to IoT Hub - 65% T3 - Almost done completed the web app and learning PostgreSQL - 70%
14/4/2020	T1 - Sent data to cloud with R - Pi and NodeMCU. T2 - Need to test out code with hardware - 100% T3 - Web App complete just need integration with Azure database. For the mobile app, found a video which can help connect Thunkable to Firebase - Web - 100% and Mobile - 70%

0.2 Sensors and Circuit Diagram

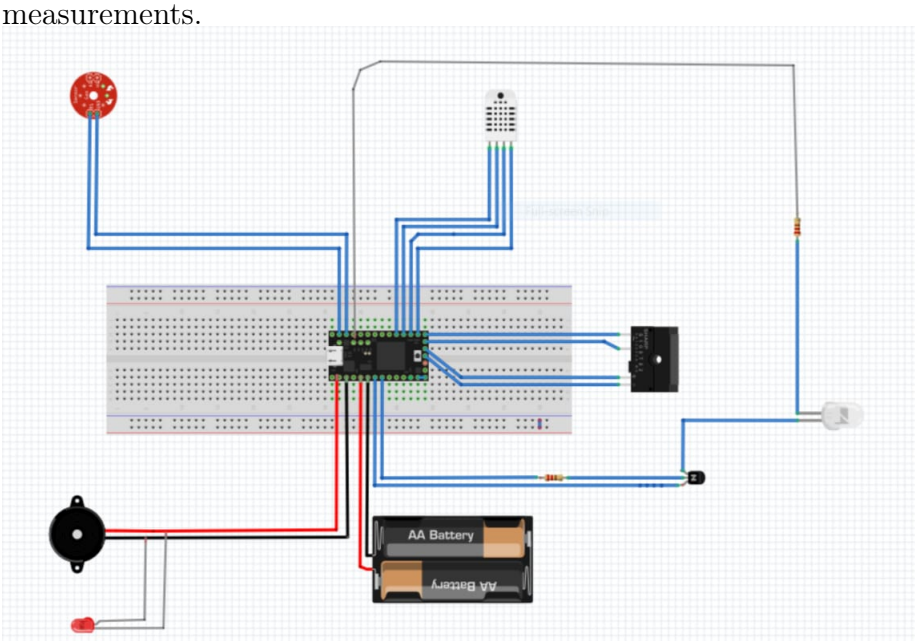
List of Sensors

1. DHT Sensor or BME280 for Temperature
2. Plantower PMS1003 and HPMA115S0-XXX for Air Quality

Factors considered for deciding sensors

1. Cost
2. Accuracy
3. Concentration Range
4. Response Time
5. Change in Performance vs Humidity

NOTE : HPMA115S0-XXX does much better in high humidity conditions and hence we decided to go with that as that's a major factor. The DHT sensor is preferable for internal measurements.



0.3 Working

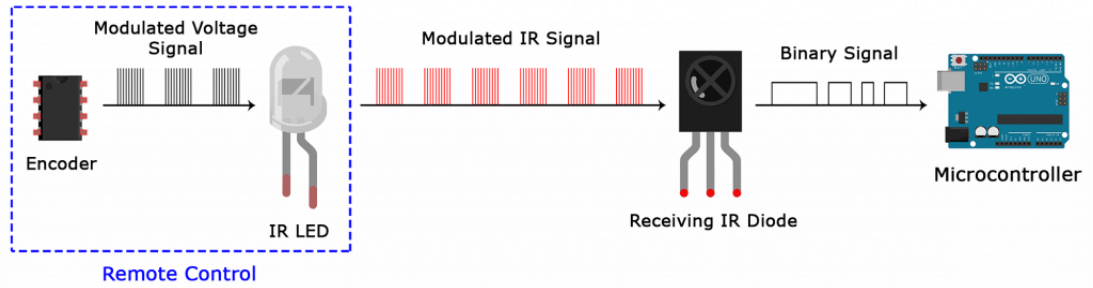
We were first under the impression that this would be a smoke detector alarm circuit. Upon further clarification and looking into Air Veda product was decided that we would be building a device to measure PM 2.5 PM 10 and CO2 level monitoring device and an Alarm would be triggered when air quality reaches Unhealthy.

Index Values	Category	Cautionary Statements	$PM_{2.5}$ ($\mu g\ m^{-3}$)	PM_{10} ($\mu g\ m^{-3}$)
0–50	Good	None	0–15.4	0–54
51–100	Moderate	Unusually sensitive people should consider reducing prolonged or heavy exertion	15.5–40.4	55–154
101–150	Unhealthy for sensitive groups	Sensitive groups should reduce prolonged or heavy exertion	40.5–65.4	155–254
151–200	Unhealthy	Sensitive groups should avoid prolonged or heavy exertion; everyone else should reduce prolonged or heavy exertion	65.5–150.4	255–354
201–300	Very unhealthy	Sensitive groups should avoid all physical activity outdoors; everyone else should avoid prolonged or heavy exertion	150.5–250.4	355–424

Source: US EPA, 1997

Searched for algorithms to factor in effect of humidity but were unable to find anything suitable. Looked into the possibility of heating air before it's tested to remove humidity.

For the AC devices, the idea was to Turn ON/OFF and Set Temperature and Fan Speed. This uses basic IR protocols and control using NodeMCU and an IR LED.
A typical IR system requires an IR transmitter and an IR receiver. We will make our signal a PWM signal at a certain frequency (modulation). Frequencies used in almost all IR remotes are 30KHz, 33KHz, 36KHz, 38KHz, 40KHz and 56KHz. Most common ones though, are **38KHz** and **40KHz**.

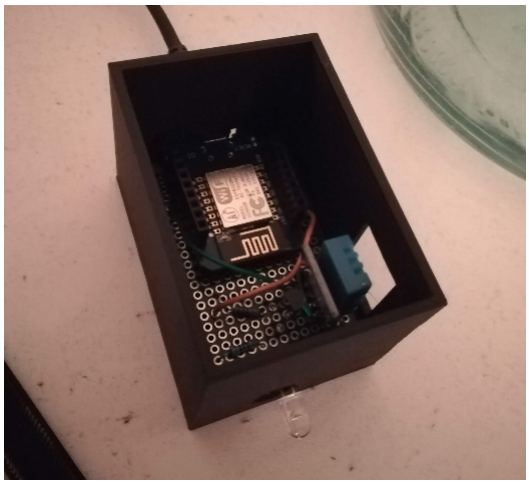


A receiver module(TSOP module) demodulates the carrier signal(Ex : 38KHz) to a more suitable TTL logic of GND and VCC. The duration of HIGH of LOW logic denotes bit '1' or '0' . The duration varies by every remote protocol.

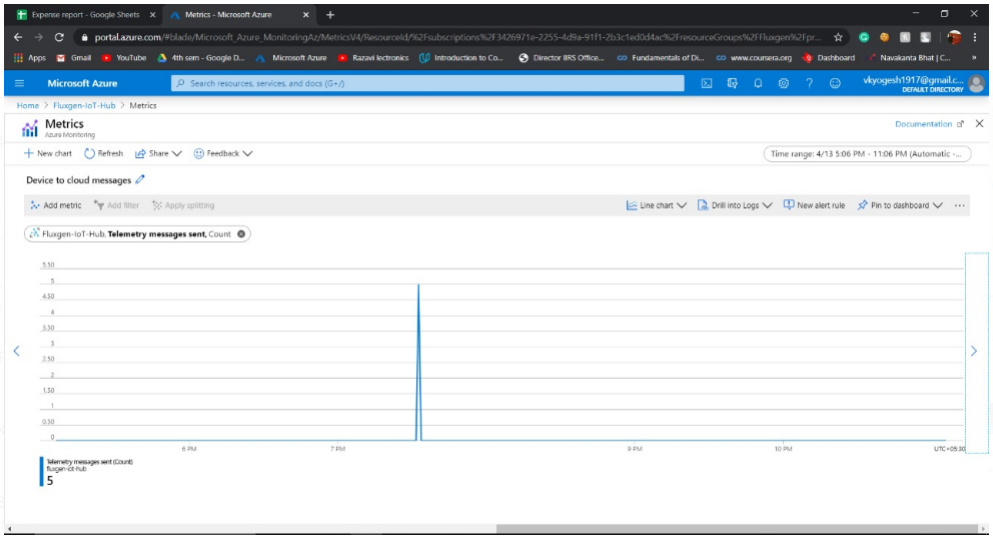
The web application collects information from the Azure database and displays it in three webpages. The main task dashboard gives an overview of all the data collected. The temperature and air quality pages give a more detailed interpretation of the data collected. The website is built using react.js and the backend is implemented using node.js. The database used is PostgreSQL. The Backend page includes a way to make a login page easily if necessary.

0.4 Final Model

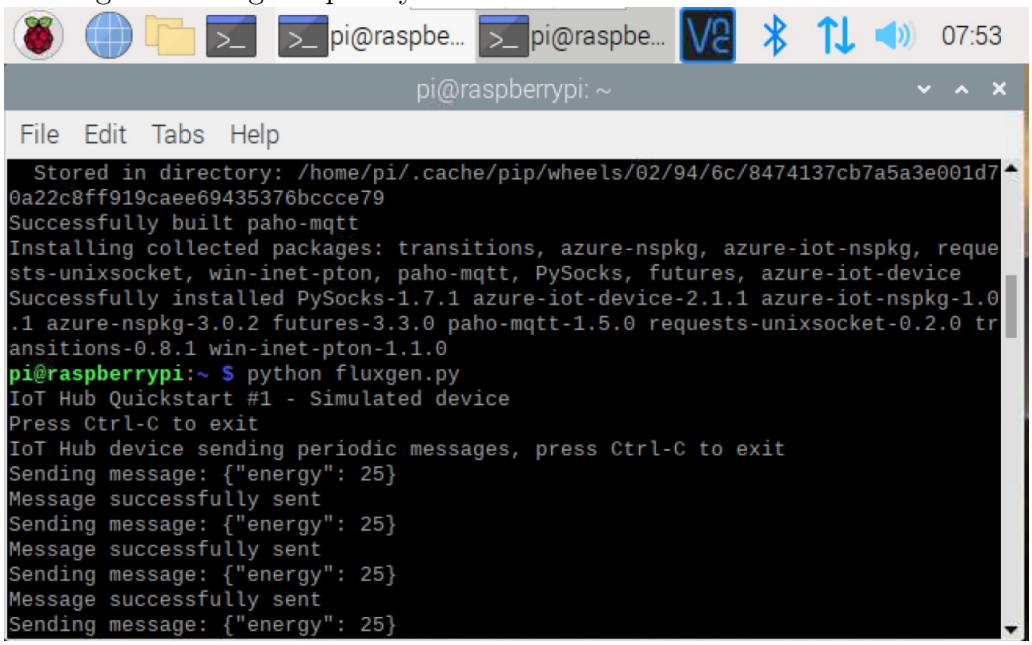
To have a display of the Temperature and Air Quality, a simple OLED Screen can be connected and used. An SD Card module can be attached to store data on the device, apart from sending it to the cloud and being monitored on the app. (If required for safety purposes)



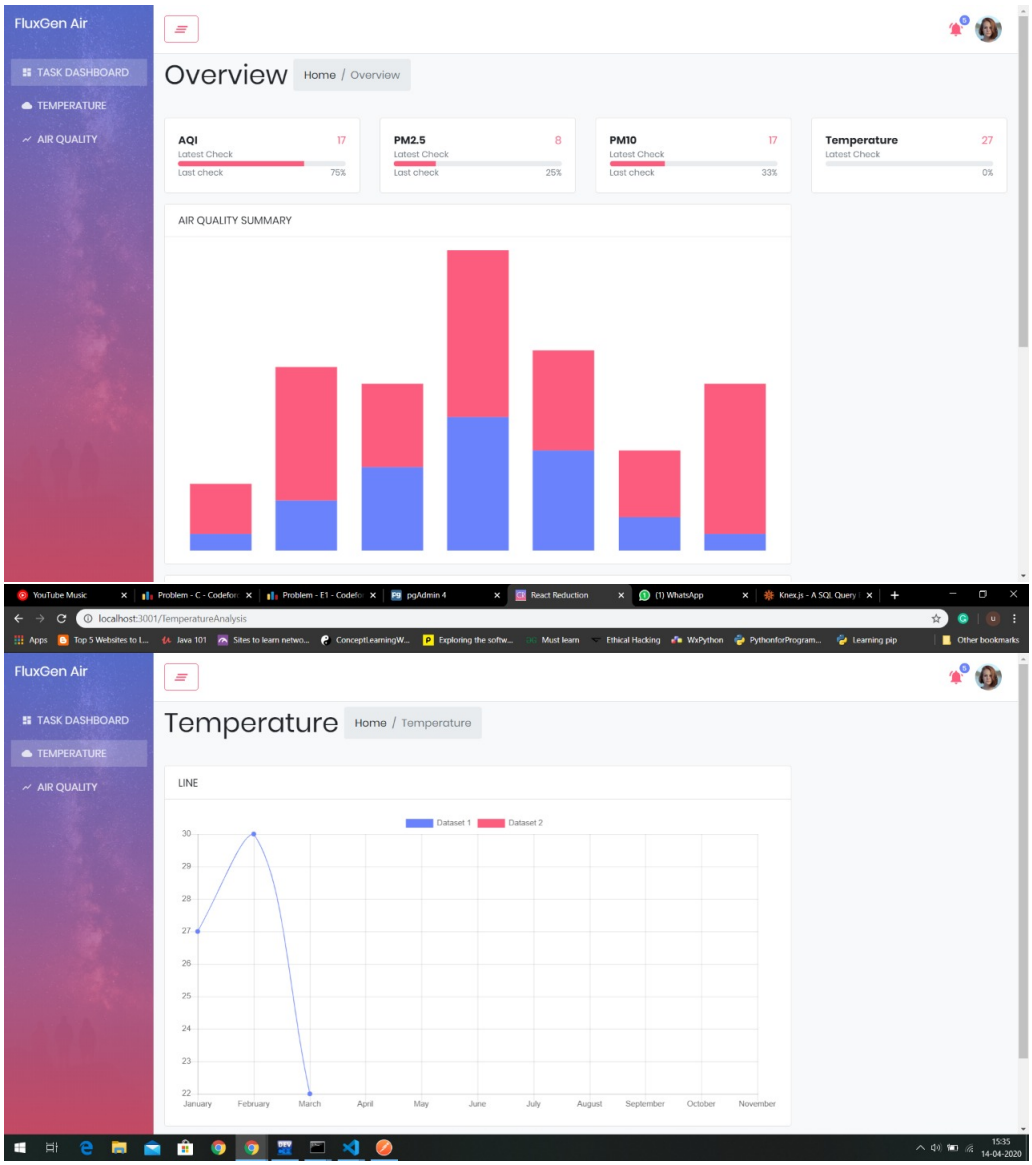
Sending data to Azure cloud

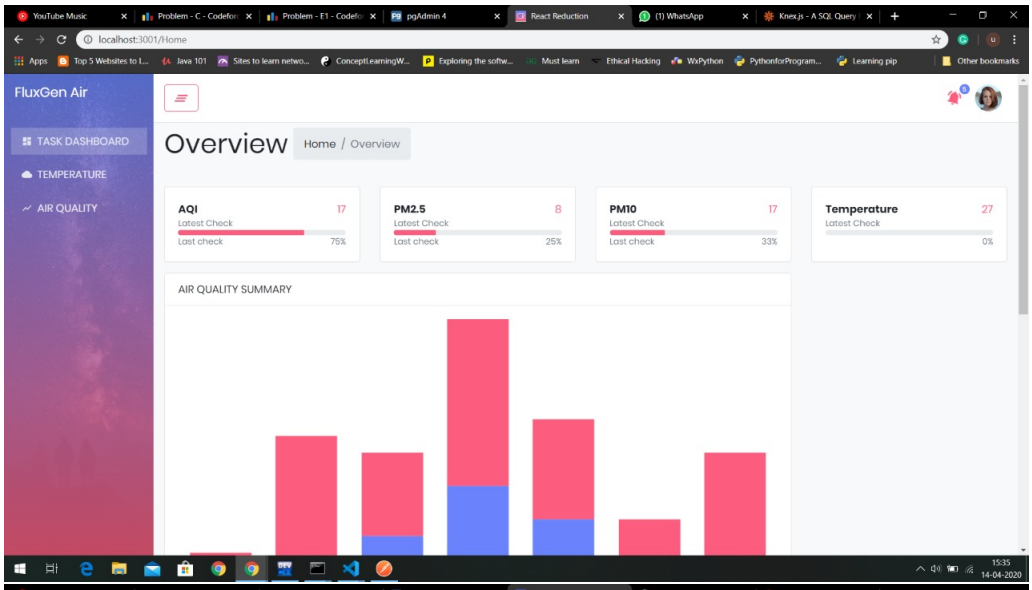


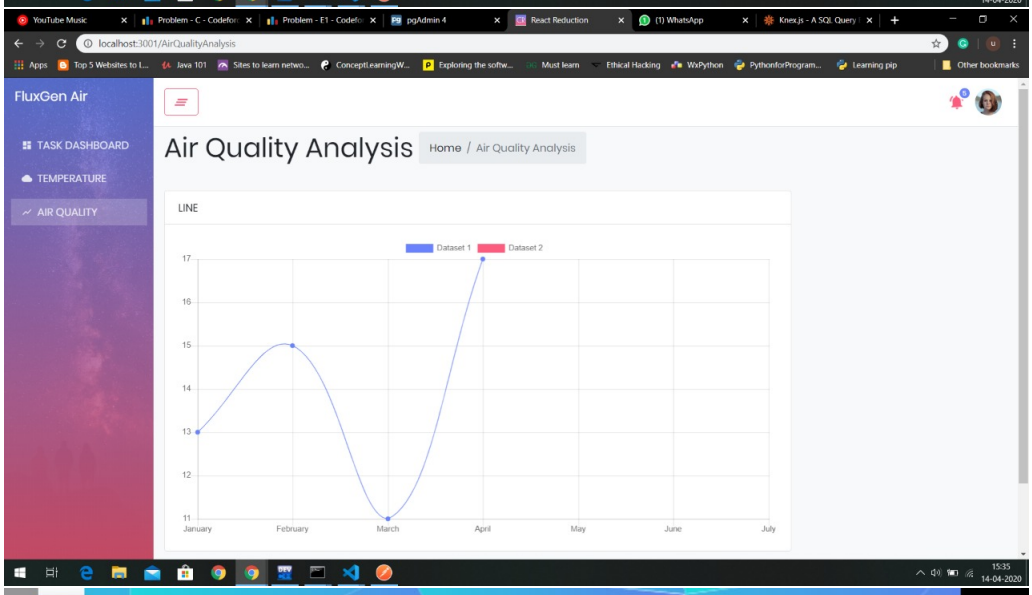
Sending data using Raspberry Pi

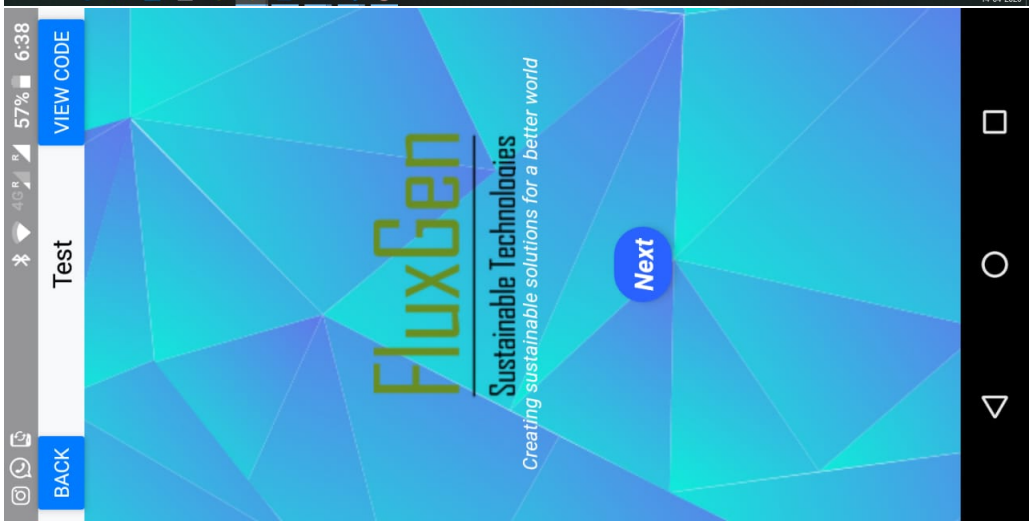


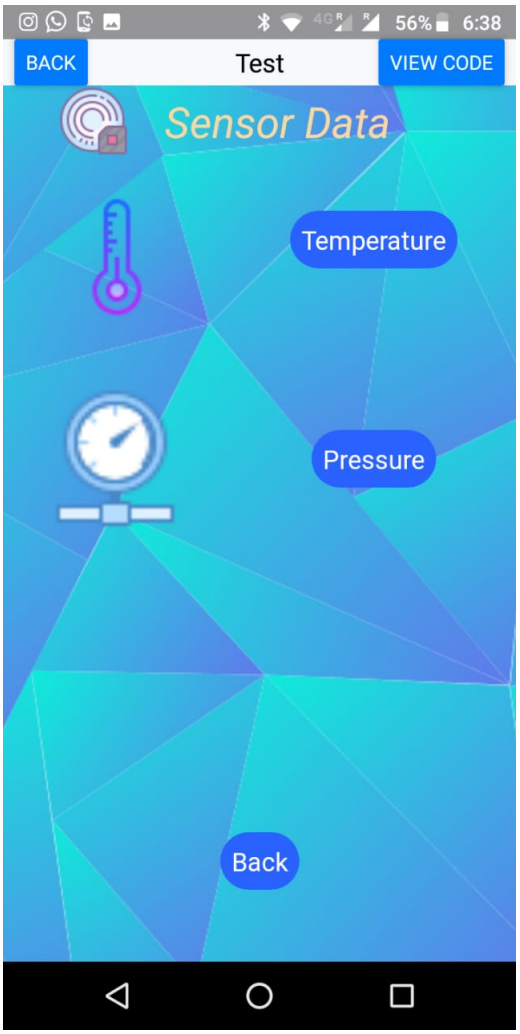
Here are a few snapshots of the website


FluxGen Air
TASK DASHBOARD
TEMPERATURE
AIR QUALITY
Overview
Home / Overview
AQI 17
Latest Check
Last check 75%
PM2.5 8
Latest Check
Last check 25%
PM10 17
Latest Check
Last check 33%
Temperature 27
Latest Check
0%
AIR QUALITY SUMMARY
localhost:3001/TemperatureAnalysis
FluxGen Air
TASK DASHBOARD
TEMPERATURE
AIR QUALITY
Temperature
Home / Temperature
LINE
Dataset 1
Dataset 2
30
29
28
27
26
25
24
23
22
January
February
March
April
May
June
July
August
September
October
November
15:35
14-04-2020


FluxGen Air
TASK DASHBOARD
TEMPERATURE
AIR QUALITY
Overview
Home / Overview
AQI
Latest Check
Last check
75%
PM2.5
Latest Check
Last check
25%
PM10
Latest Check
Last check
33%
Temperature
Latest Check
0%
AIR QUALITY SUMMARY


FluxGen Air
TASK DASHBOARD
TEMPERATURE
AIR QUALITY
Air Quality Analysis
Home / Air Quality Analysis
LINE
Dataset 1
Dataset 2
January
February
March
April
May
June
July


VIEW CODE
Test
BACK
FluxGen
Sustainable Technologies
Creating sustainable solutions for a better world
Next



References

GitHub Repository for the Web App
Google Drive Link of the Project
Sending Data to Azure from NodeMCU
Details for the Mobile Application