**Lab\_1**

preranasam edited this page 1 minute ago · [9 revisions](https://github.com/preranasam/IoT/wiki/Lab_1/_history)

**Introduction**

This lab assignment was the integration of the first 4 ICPs. We had to implement a weather station and pulse monitoring system using Arduino, MIT app inventor-2 and Adafruit dashboard. Each sensor must have a normal operating range and alerts when the values go out of range.

**Objective**

The objective of the lab was to interface the following sensors with Arduino. Read data from them and display it on LCD. Push the sensor data on the Adafruit IO dashboard using the ESP8266 WiFi module. Use alert LEDs to indicate when the values go out of range and also send alert text on the LCD.

1. Temperature and Humidity sensor
2. Barometer
3. UV Sensor
4. Dust Sensor
5. Light Sensor
6. Pulse Sensor

Design an Android app using MIT app inventor 2. Read all the sensor data in it and send alert notifications when value is out of the normal operating range.

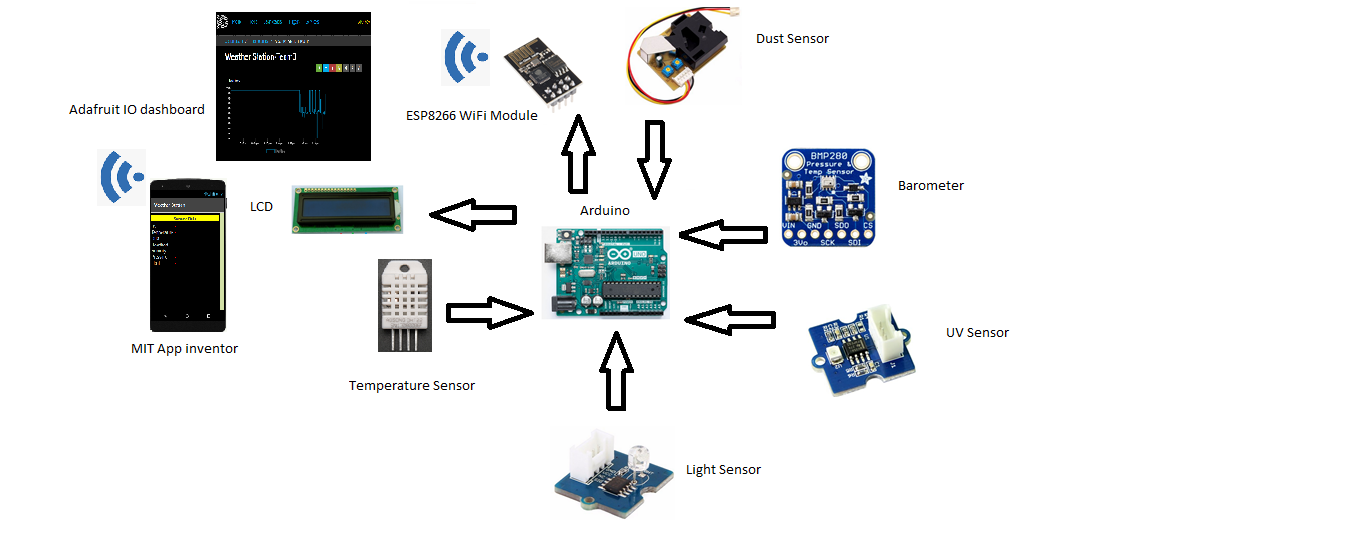
**Approaches/Methods**

Our approach was to modify the ICP3 integrated code to add a pulse sensor in it. Add the conditions for normal operating range and display alert messages on LCD. We initially thought of using the Nodemcu with Adafruit IO but it had a lot of connection challenges and we implemented the lab with Arduino Uno.

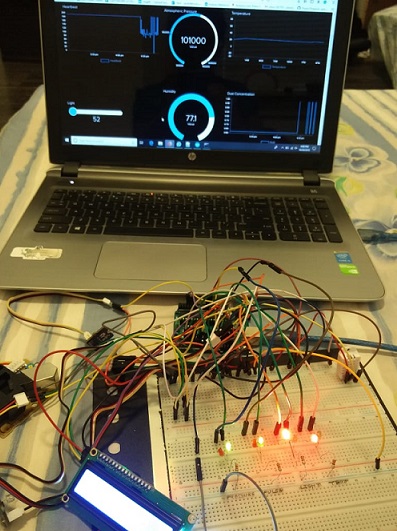
**Work Flow**

1. We first assembled all the sensors on the breadboard and connected it with Arduino.
2. Then we connected the WiFi module and checked if it is working as expected.
3. We then modified the ICP3 integrated code, to add pulse sensor code in it and limits to detect the alert conditions. we added LEDs only to 4 sensors due to limitation of IOs on the Arduino Board.
4. We ensured everything is working with the ThingSpeak cloud.
5. Next task was to design the dashboard. We searched for different alternatives like freeboard.io, Adafruit IO, Bluemix, etc. But Adafruit IO had better visualization. So we selected the Adafruit IO.
6. But, the Adfaruit IO is not compatible with Arduino Uno and we were planning to use the NodeMCU. Somehow, the NodeMCU had multiple library mismatch errors and has only one analog input. Hence, we finalized to use Arduino Uno + ESP8266 + ThingSpeak + Adafruit IO + Python Script.
7. We wrote a Python script to read URL from ThingSpeak. This data was obtained from the URL provided in the ThingSpeak profile in the JSON format. This JSON format was converted into python data using the JSON library and the Python data for specific field was uploaded to the respective Adafruit key.

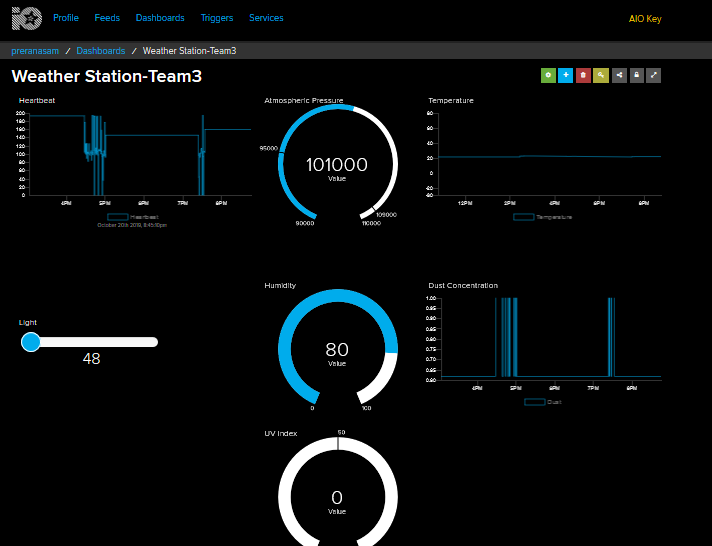
**Block Diagram**



**Circuit Diagram**



**Dashboard**



**Dashboard Link**

[AdafruitIO](https://io.adafruit.com/preranasam/dashboards/weather-station-team3)

**Android Application**



**Video**

[Weather\_Station\_and\_Pulse\_Monitoring\_System](https://youtu.be/Waa-o0QrFEQ)

**Parameters**

1. Following Pins were used for the inputs/outputs:

* Temperature Sensor --> D7
* Light Sensor --> A1
* UV Sensor--> A0
* Dust Sensor --> D8
* Barometer --> SDA & SCL
* Temperature sensor Green LED--> D2
* Temperature sensor Red LED--> D3
* Pressure sensor Green LED--> D6
* Pressure sensor Red LED--> D11
* Light sensor Green LED--> D4
* Light sensor Green LED--> D5
* Pulse sensor Green LED--> D13
* Pulse sensor Red LED--> D12

1. Limits Normal range. These ranges were selected as per the sensor calibrations and to intentionally generate alert condition for temperature sensor.

* 25<= Temp < 30
* 40<= Light < 200
* 90000 < Pressure <= 109000
* 200 < Pulse < 60

**Evaluation and Discussion**

1. We faced a lot of issues to upload data on the dashboard. Since, Adafruit IO is not compatible with Arduino Uno, we explored the Freeboard IO dashboard. But, the professor suggested using Adafruit IO since it is also used in the industries.
2. With the help of the professor's hint, we decided to write a script to upload ThingSpeak data to Adafruit IO. We had to start from scratch by installing Python, the PIP library and Adafruit-IO library for Python. Initially, we were struggling to convert the JSON data from ThingSpeak to Python data in order to push to the Adafruit IO. Finally, we could upload the data on Adafruit IO.

**Conclusion**

This was a challenging and a great learning experience to write a script in Python. We learned to read data from ThingSpeak and convert the JSON format provided by reading the URL into Python format and uploading the appropriate field data to the Adafruit feed keys. Exploring the Adafruit IO dashboard was interesting.

**Github Links**

1. [Prerana](https://github.com/preranasam/IoT/wiki/Lab_1)
2. [Zeal](https://github.com/zealpatel1990/iot_fall_2019/wiki)
3. [Ashish](https://github.com/akkipant/IoT-Fall-2019/wiki)