CSEE5590-0005

IoT/ Robot Programming

(2018 Fall)

*Lab Assignment 4*

**IoT Smart Home Application with Fire Fighting Robot and NAO Robot**

Submitted On:

04 December, 2018

Submitted By:

Name: Farid Uddin Ahmed

Class ID: 01

Name: Mihir Manoj Pitale

Class ID: 09

Name: Zarin Tasnim Sandhie

Class ID: 10

Name: Kenton William Hanifl

Class ID: 12



LIST OF THE DOCUMENTATIONS

1. AUTHORS
2. OBJECTIVE
3. GOALS
4. PROJECT FLOW
5. MATERIALS REQUIRED
6. PLATFORM USED
7. CIRCUIT CONNECTION
8. NODE RED FLOW
9. MIT APP INVENTOR FLOW
10. CHOREGRAPHE FLOW
11. METHODOLOGY
12. APP VISUALIZATION
13. NOTIFICATION & WARNING
14. LIMITATIONS
15. CONCLUSION
16. LINKS

AUTHORS

This is the report containing the documentations of the Assignment #4. The contributors of the assinments are: Farid Uddin Ahmed (Class ID: 01), Mihir Manoj Pitale (Class ID: 09), Zarin Tasnim Sandhie (Class ID: 10) and Kenton William Hanifl (Class ID: 12)

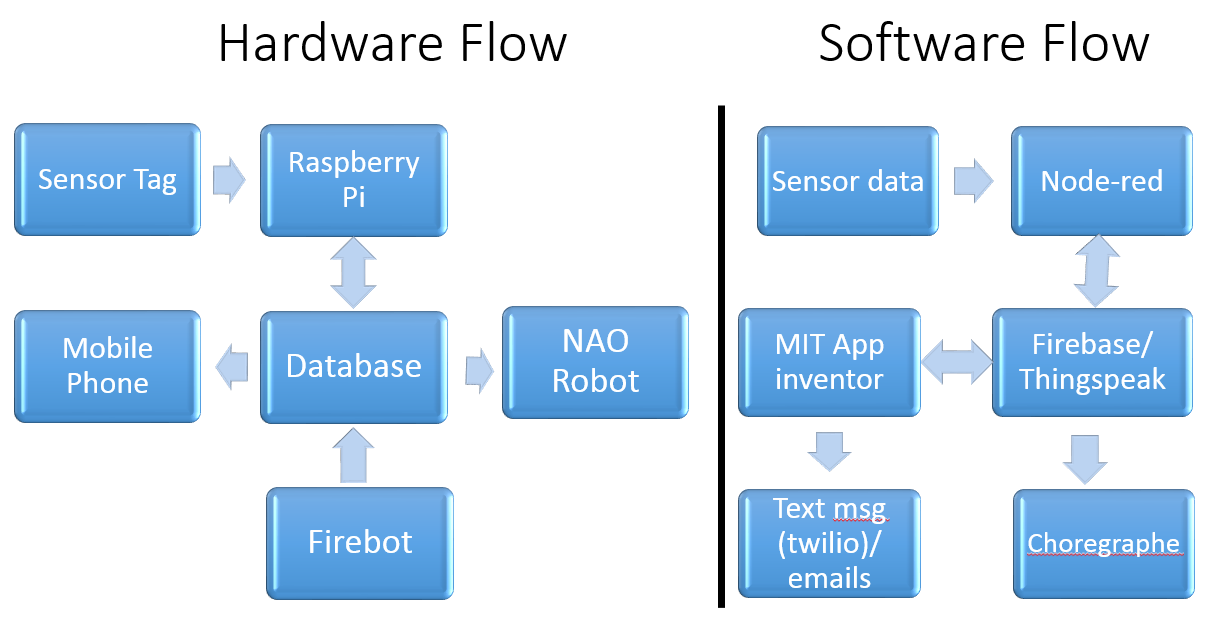
OBJECTIVE

To create a smart home application which would use sensors such as temperature, air, pressure, fire and motion sensor and use all of them together to make a firefighting robot. The system will able to generate notification via Text message, email and Android application. Also, the NAO robot is integrated with the whole system which will be able to say the weather condition as well as the fire situation at home.

GOALS

1. Use the Sensor Tag (Humidity, Temperature etc) on Arduino or Raspberry platform.
2. Visualize all the sensor data via Node-red dashboard.
3. Create an event based on the sensor data.
4. Send notification via text message, email, Android or Web-application.
5. Integrate the above mentioned sensor to make an emergency firefighting robot.
6. Integrate NAO robot with the system which will be able to mention the weather condition of the home along with the fire situation.

PROJECT FLOW



MATERIALS REQUIRED

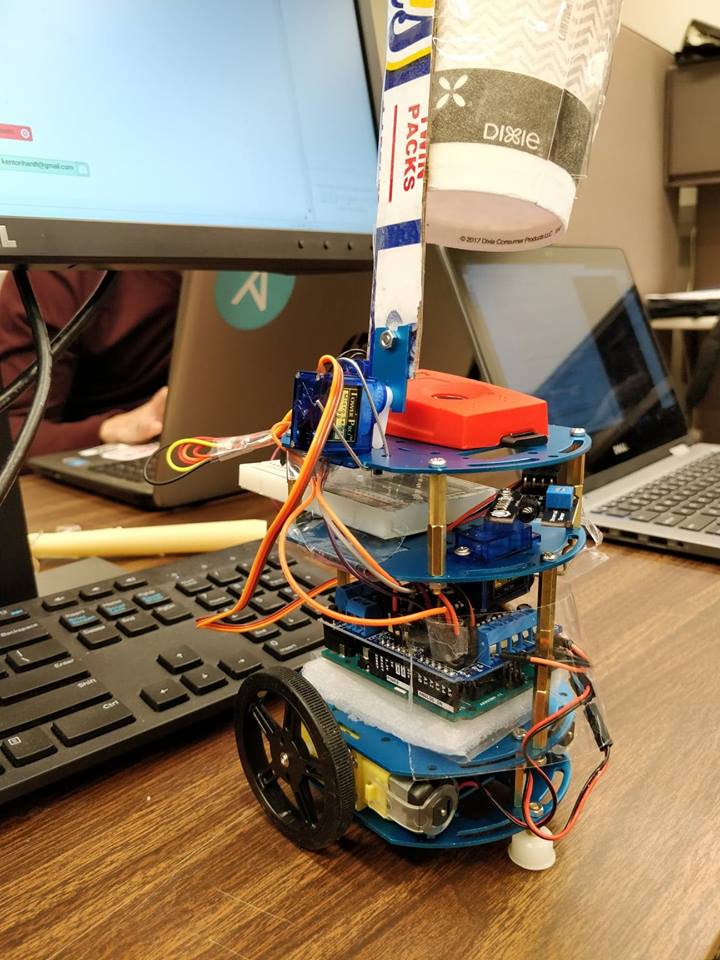
* Robot Chasis
* DC motor (2)
* Servo Motot (2)
* Flame Sensor
* Arduino Motor Shield
* ESP 8266 WIFI module
* Wheels
* Sensor Tag
* Water container
* LEDs
* Breadboard
* Connectors
* Resistors
* Arduino Board
* Raspberry Pi
* NAO robot

PLATFORM USED

* Raspberry PI (for Node-red and App inventor)
* Arduino Uno (Fire Fighting Robot)
* MIT app inventor (for controlling home appliances)
* Choregraphe (for NAO robot)

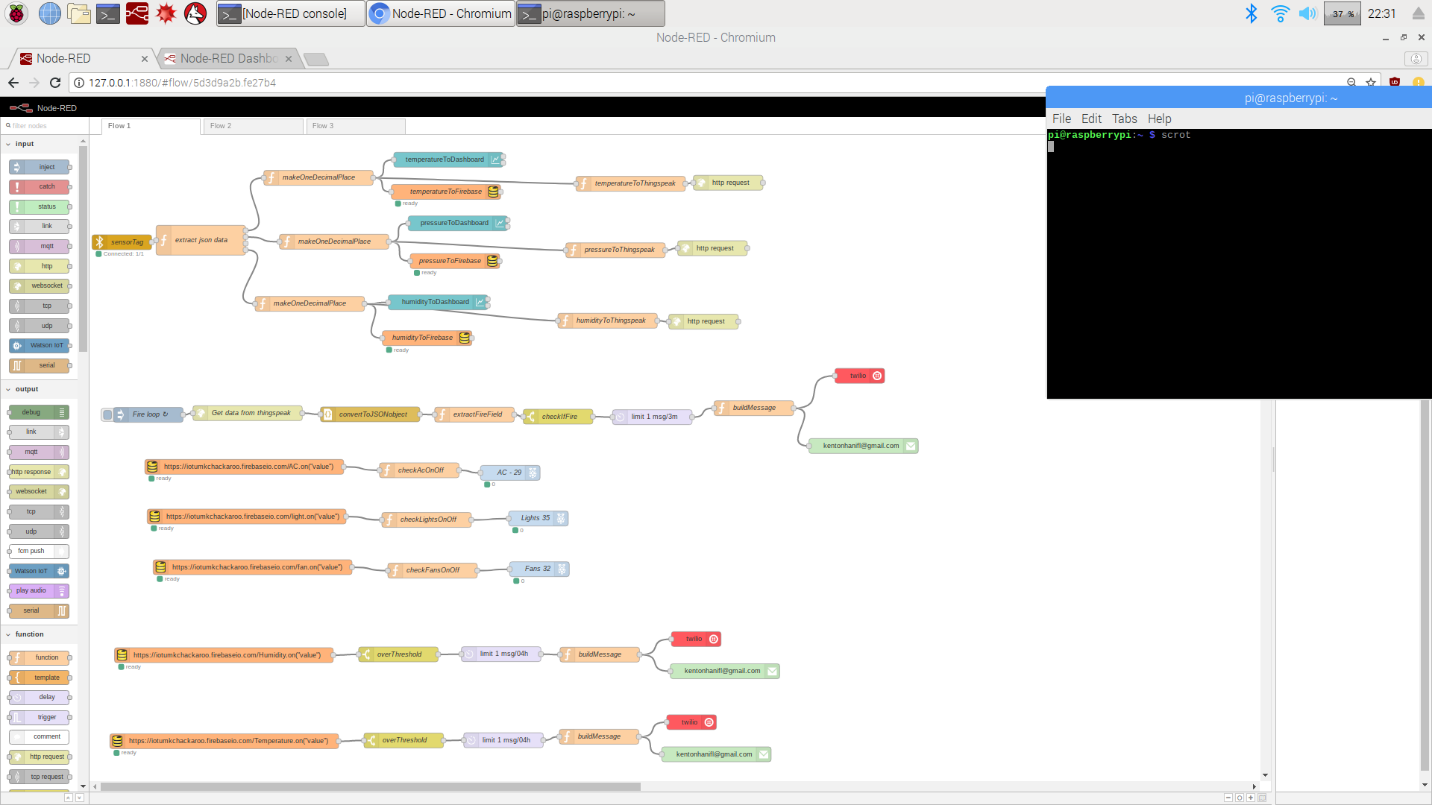
CIRCUIT CONNECTON

The circuit is connected as per figure:

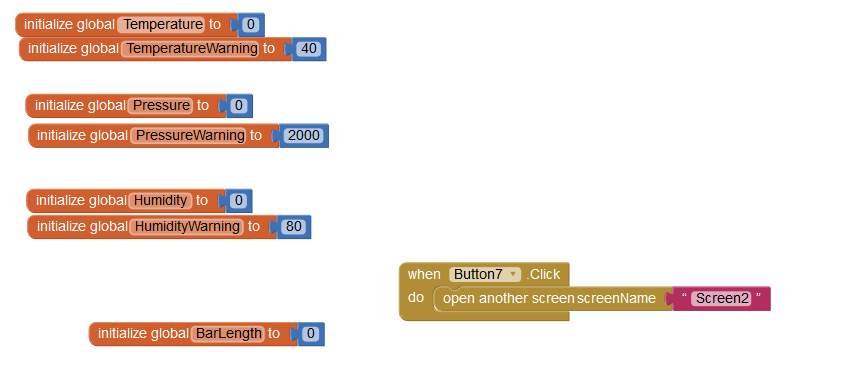
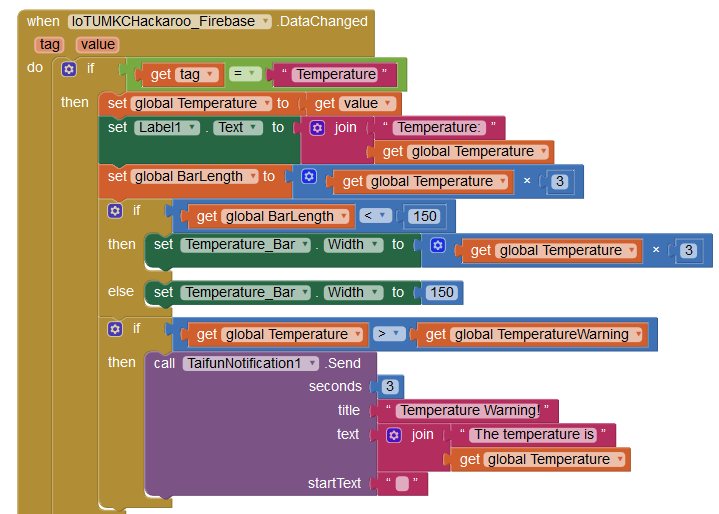


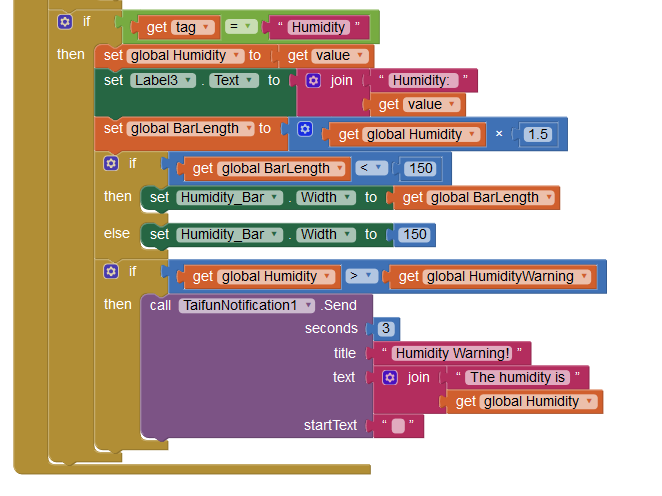
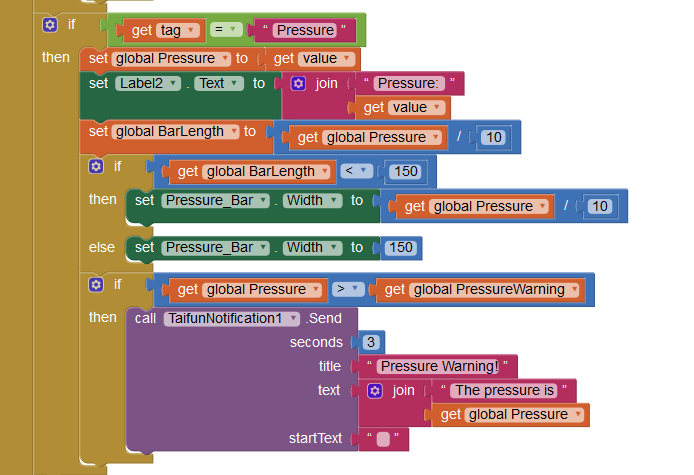
NODE-RED FLOW

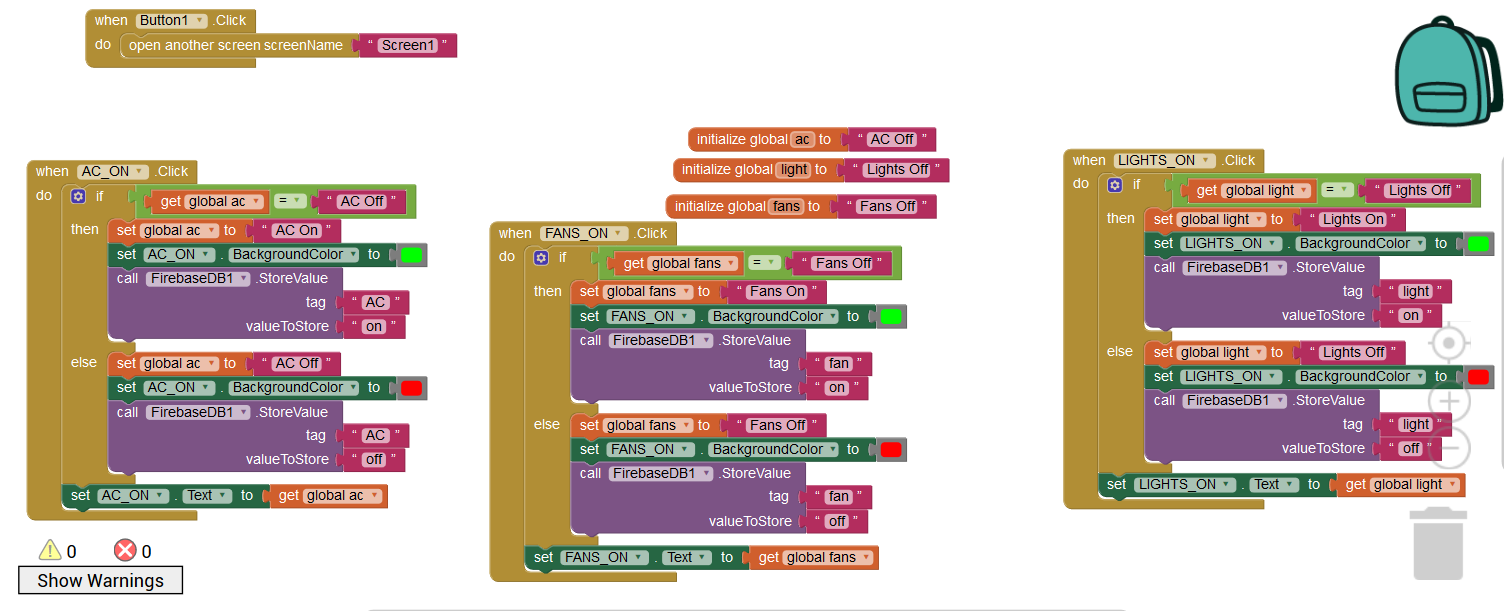
The flow made in node-red is given below:

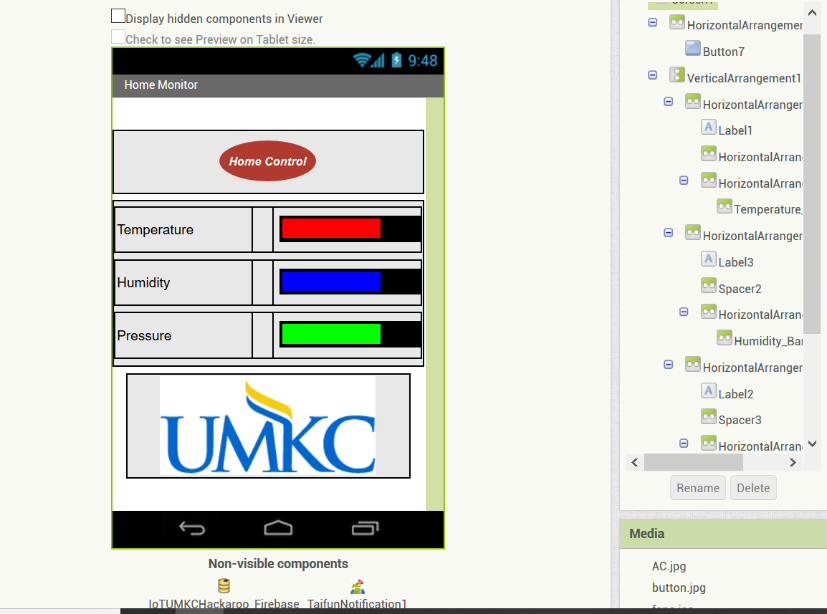


MIT APP INVENTOR FLOW



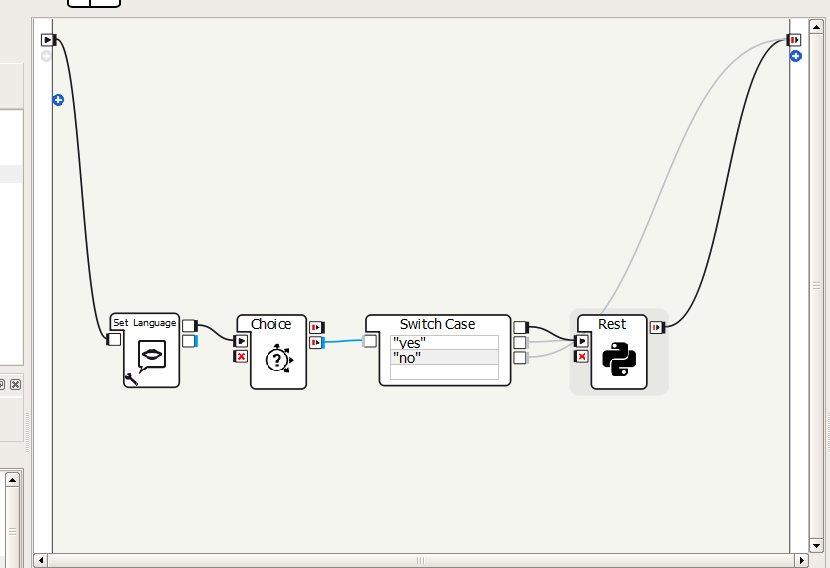






CHOREGRAPHE FLOW

The flow made in choregraphe is given below:



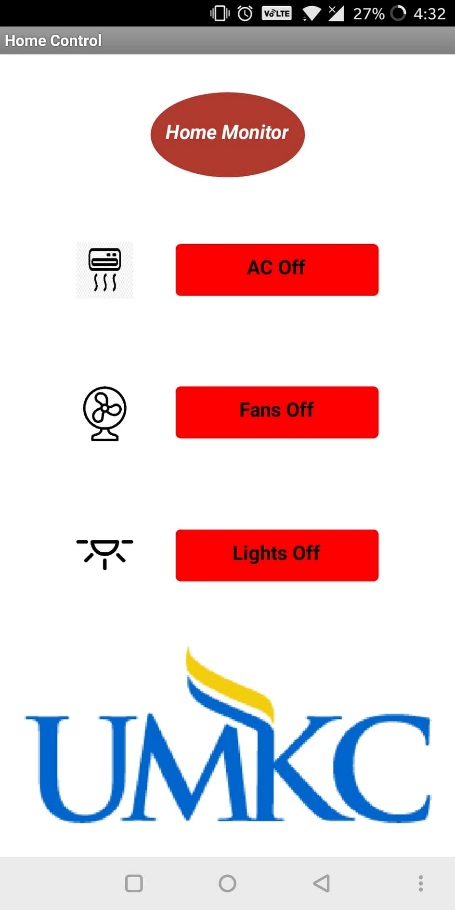
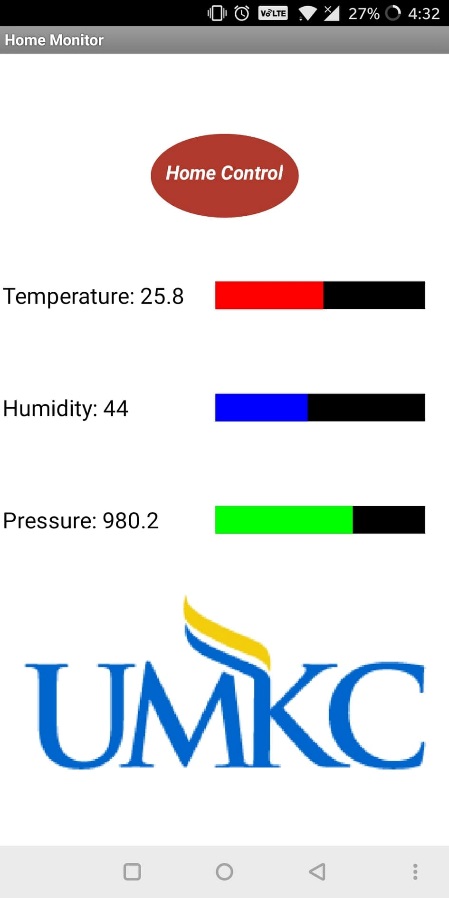
METHODOLOGY

* To extract the sensor data (Humadity, Pressure and Temperature) from the sensor.
* Use node red to upload the data to firebase.
* View those data remotely via a mobile app.
* Control the AC, Fan and Light through the app remotely via node-red.
* If any of the weather condition goes above safety limit, send a mobile notification and also send an email.
* If there is a fire at home, the fire robot will sense the fire and put-out the fire.
* A buzzer will make sound if there is any fire.
* And a text message and an email will be sent about the fire.
* The data from theingspeak will also be fetched by “Choregraphe” software through appropriate flow which in turn will be said by NAO robot.
* Again, in case of a fire, the fire information will be send to thingspeak via Arduino Uno board.
* That data also will be fetched by “Choregraphe” software through appropriate flow which in turn will be said by NAO robot.

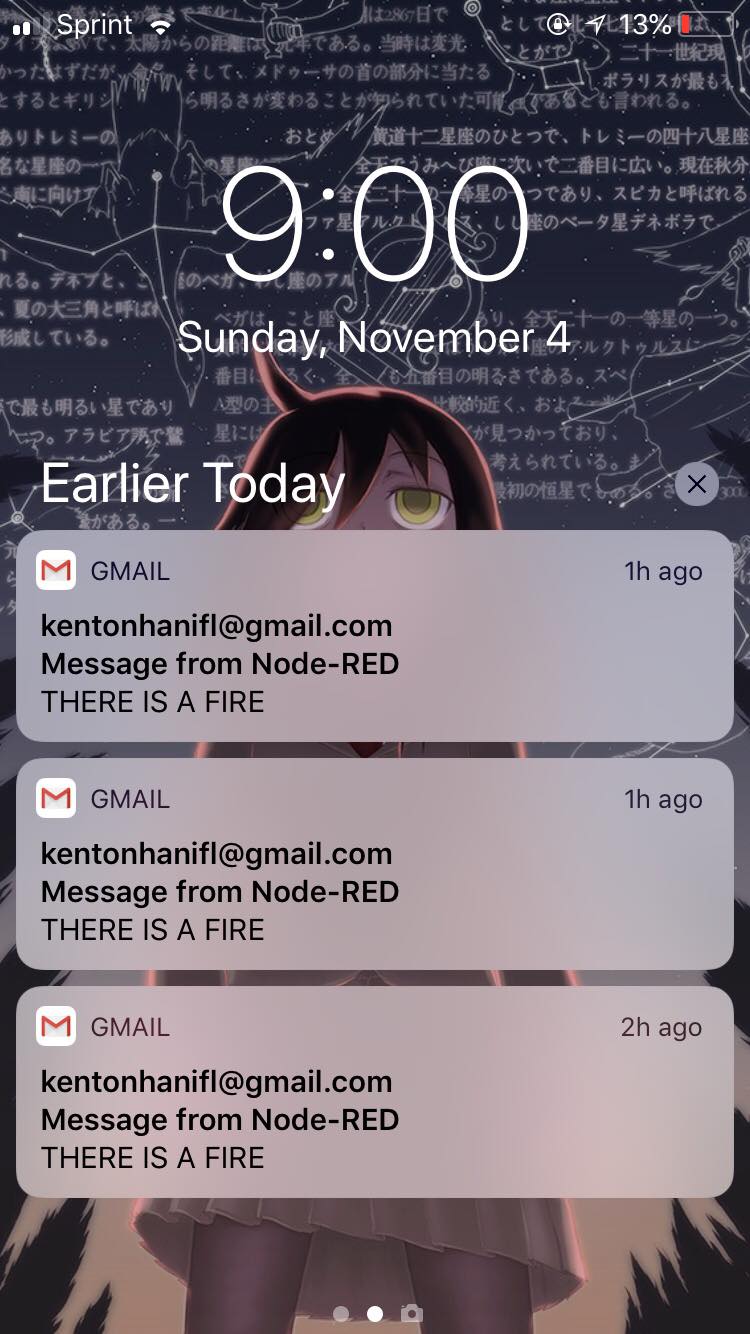
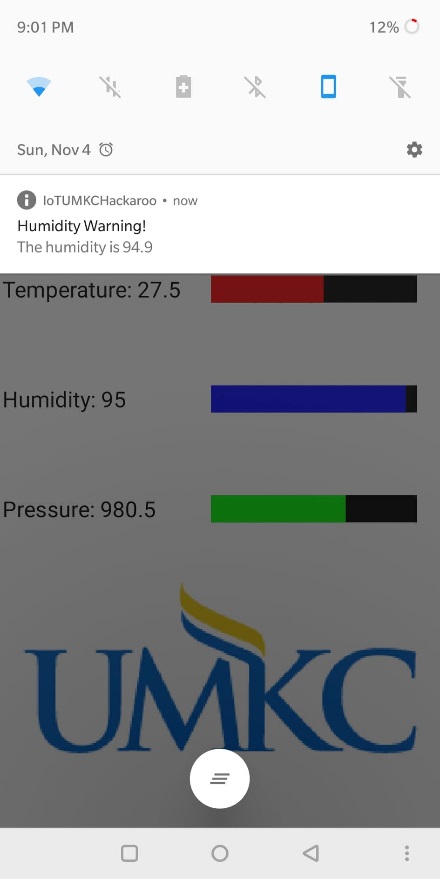
FEATURE

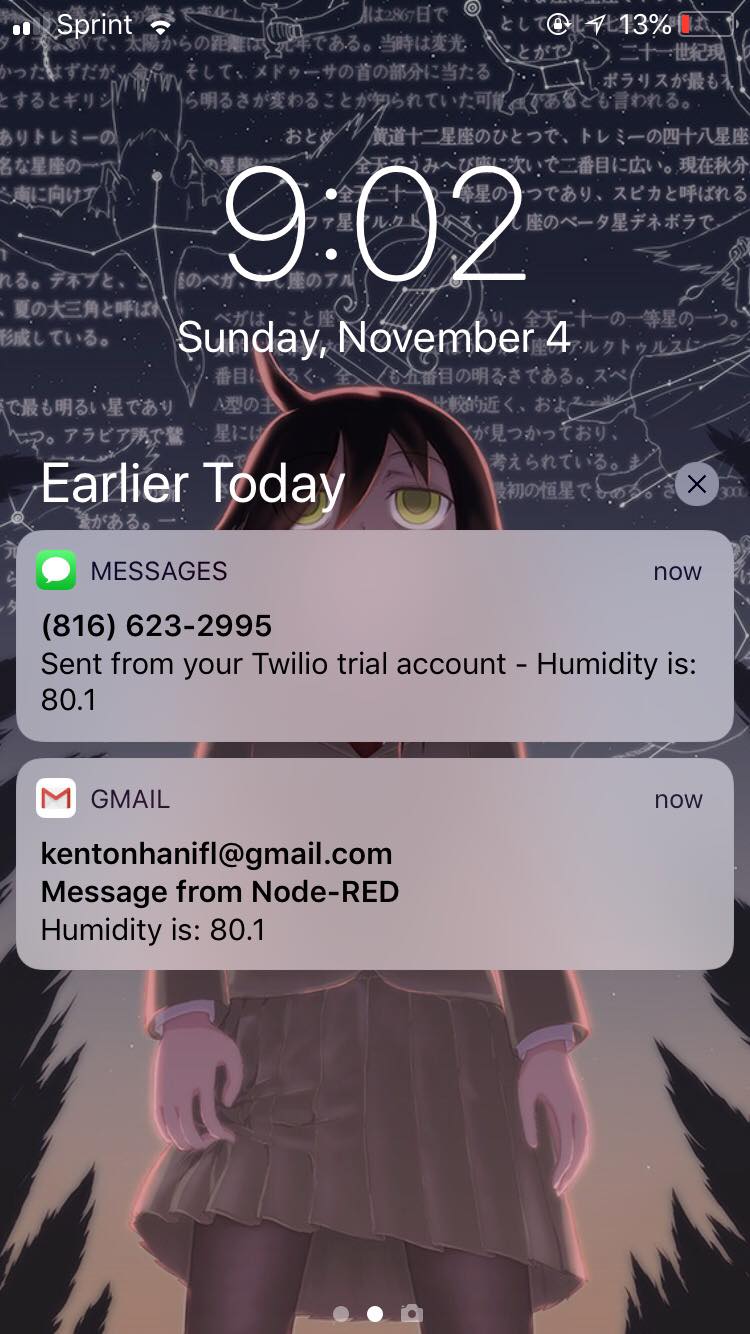
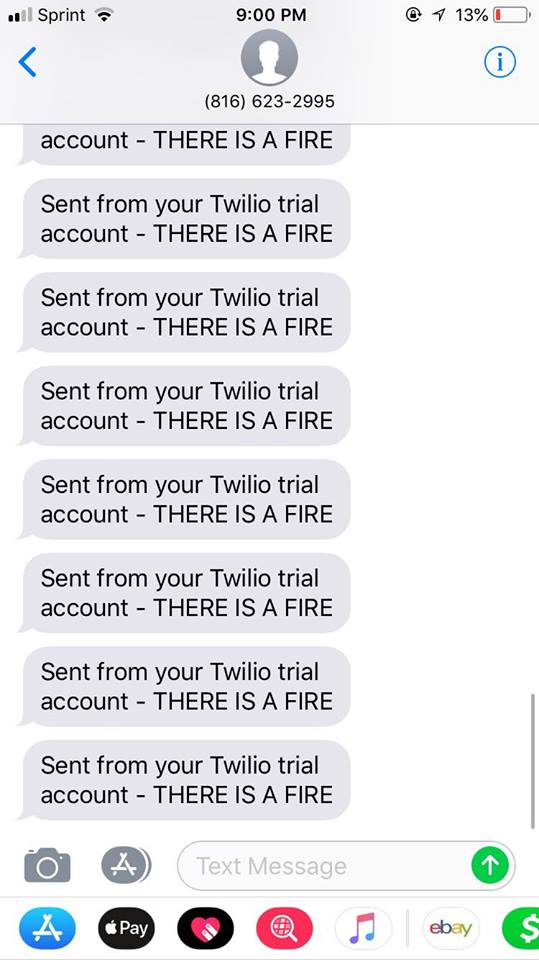
* Visualization and control of the temperature, pressure and humidity of the home from a remote place.
* Sending of an email if the temperature, pressure or humidity goes above a certain set value.
* Sensing a fire if there is one and sending a text message via twilio to the home owner.
* Putting out the fire by splashing water to the fire affected areas in the given proximity.
* Making a high pitched sound in case of fire which can warn the nearby people.
* Controlling the AC, fans, lights via one button build in MIT app inventor application.
* Using a NAO robot which can say the weather condition at home and also can describe the fire situation if there is any.
* Economic and cost effective design for a safe environment at home.

APP VISALIZATIONS

NOTIFICATION AND WARNING

LIMITATIONS

Fire Fighting Robot:

* The sensing of the fire is a bit tedious due to the limited number of Infra-Red (IR) flame sensors. As we used only one sensor to detect the fire, we had to use a servo motor which can rotate the sensor 180 degree and detect the fire.
* The range of the flame sensor is about 3 feet, so it cannot detect the fire which is far away.
* As we weren't given any water pump, we had to use servo motor to pour water over fire.
* Due to the lightness of the whole structure, water weight disrupts the motion of the robot.

Solution:

Using a good number of flame sensors and using more powerful motors and wheels can minimize the above mentioned limitations.

Mobile Application:

* We were able to turn on and off few LEDs which represents the home appliances like Air Conditioner, Light and Fan. Testing was not done with original devices.
* Also we only turned on and off the devices, but we did not control the levels on those devices, e.g. controlling the temperature on the AC.

Solution:

With proper controlling circuitry, it will be possible to control real devices. And also the parameters on those device can also be changed.

CONCLUSION

This assignment integrates of all the ICPs done during the previous three weeks along with the project. During this assignment, we were able to create an IoT smart home application with the NAO robot. Also the fire detection was also integrated with NAO robot.

LINKS

GitHub Link:

<https://github.com/farid7666/CS5690-IoT-Robot/tree/master/Project>

GitHub Wiki Link:

<https://github.com/farid7666/CS5690-IoT-Robot/wiki/Project>

Video link:

<https://www.youtube.com/watch?v=0Ki5mee7J8k&feature=youtu.be>

Source Code Link:

https://github.com/farid7666/CS5690-IoT-Robot/tree/master/Project/Source%20Code

Link for Node-red flow snippet:

<https://github.com/farid7666/CS5690-IoT-Robot/tree/master/Project/Documentation/Flow_snippets_node_red>

Link for app-inventor flow snippet:

<https://github.com/farid7666/CS5690-IoT-Robot/tree/master/Project/Documentation/Flow_snippets_app_inventor>

Link for Choregraphe flow snippet:

<https://github.com/farid7666/CS5690-IoT-Robot/blob/master/Project/Documentation/Flow_snippets_choregraphe/choregraphe_flow.png>

Link for Mobile/email/text snippet:

<https://github.com/farid7666/CS5690-IoT-Robot/tree/master/Project/Visualization/Mobile_email_text_notification_snippets>