Final Report - CS-190

Kush Dave, SID# 61699159

⇒ Introduction

The final project was a sensor that counts the net total people in a room as well as the Air Quality inside the room. Since both the sensors needed to work all the time, I implemented the Air Quality Sensor on Sparkfun ESP8266 Thing Dev and people counter on Arduino Uno. By the means of Serial Connection, I used to communicate between Arduino and ESP.

⇒ Sensors

For the Air Quality Sensor, I used Adafruit CCS811 Air Quality Sensor. This sensor measures two things. It measures Carbon Dioxide levels and VOC (Volatile Organic Compound) levels. The Carbon Dioxide level is measured in ppm (Parts Per Million) and VOC level is measured in µg/m3 (Microgram per cubic metre). For this project, I determined Carbon Dioxide levels below 1000 ppm were safe and VOC levels below 500 µg/m3 were safe. If any of the two exceeded the predetermined limit, an alert would be sent to the Server notifying the user of the same. The Sensor was interfaced to ESP8266 using I2C interface where SDA on Sensor was connected to PIN-2 of ESP8266 and SCL/SCK on Sensor was connected to PIN-14. A power of 3.3 Volts was supplied to the sensor.

For the people counter on Arduino, I used two HC-SR04 Ultrasonic Sensors. The Algorithm was simple, I calibrated both the sensors (One for incoming people and one for outgoing people) to maintain a constant distance. As soon as the sensors detected a change in distance, the count was triggered and depending upon which Sensor was triggered, I was able to determine the total number of people in the room.

Additionally, I attached the Grove LCD to the Arduino using Grove Base Shield. This was used to monitor the data and during debugging phases,

check the sanity of the data being sent. As Arduino and ESP were connected using Serial, I was able to receive the Air Quality data on Arduino from ESP and then present it on the LCD. By using a simple digital switch, I was able to cycle between displaying People counter as well as Carbon Dioxide data on the LCD. I used the Grove Touch Sensor to turn the LCD on and off. I also used the Grove Rotary Switch to control the brightness on the LCD.

⇒ Sending Data

I sent the data to Azure IoT hub using MQTT protocol. I parsed my data into JSON text and using the onboard WiFi on ESP8266, I sent the data to the hub where I can receive and process it.

⇒ Processing the Data

By using a C application which polls the IoT hub for message in the message queue, I was able to receive data directly on my machine. I then piped the data from that C application to my C++ application where it parsed the JSON text and did the data analysis on it.

⇒ Data Analysis

For Data Analytics, I accumulated data into a list and then presented the monthly average Carbon Dioxide, monthly average VOC and monthly average number of people. I divided the monthly average into 4 weekly averages as well. In case when I received an alert from the device, I displayed that alert on my application as well, notifying the user.

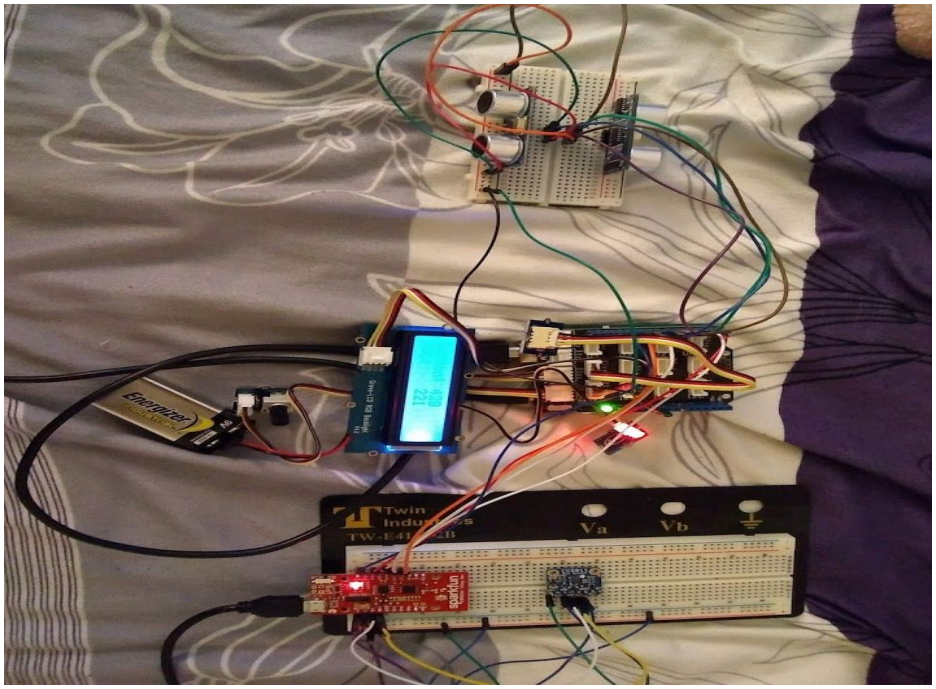
⇒ Code

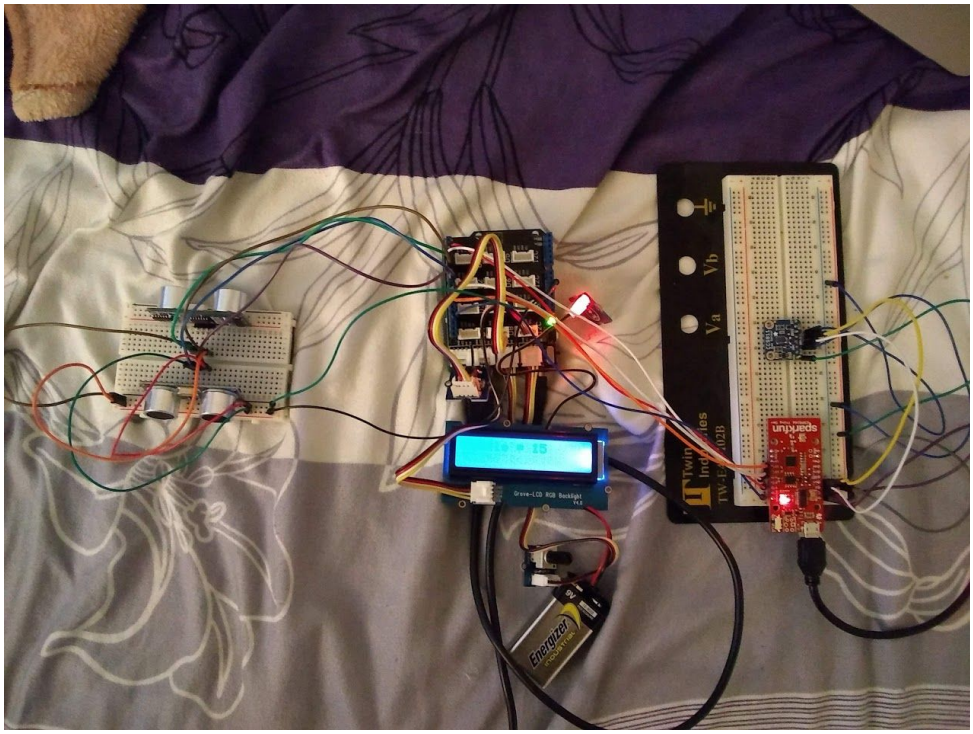
I have created three different zip files for three different categories of code I used, each of those can be found using the links posted below. These links can be followed in the PDF as well.

1. The code I used for ESP8266 can be found here: ESPApplication
2. The code I used for Arduino Uno can be found here: ArduinoApplication

3. The Application Code I used for Data Analytics can be found here: MainApplication

⇒ Pictures and Schematics





⇒ Connections

1. Arduino

   Note: Arduino Ground PIN is connected to ESP8266 Ground PIN
   Note: All the input devices, except the Ultrasonic Sensors are connected to Arduino using Grove Base Shield

Arduino PIN	**Mode**	**Connection**
Digital PIN-2	Software Serial	ESP8266 PIN-5
Digital PIN-3	Software Serial	ESP8266 PIN-4
Digital PIN-4	Digital Output	ECHO PIN HC-SR04 (1)
Digital PIN-5	Digital Input	TRIG PIN HC-SR04 (1)
Digital PIN-6	Digital Input	ECHO PIN HC-SR04 (2)
Digital PIN-7	Digital Output	TRIG PIN HC-SR04 (2)
Digital PIN-8	Digital Input	Push-Button
Digital PIN-9	Software Serial	ESP8266 PIN-13
Digital PIN-10	Software Serial	ESP8266 PIN-12
Analog PIN-A0	Analog Input	Rotary Switch
Analog PIN-A1	Analog Input	Touch Switch
I2C Grove Shield	Multiple (I2C)	Grove LCD

2. ESP8266

Note: Arduino Ground PIN is connected to ESP8266 Ground PIN

ESP8266 PIN	**Mode**	**Connection**
Digital PIN-2	SDA (Air Q Sensor)	SDA on Air Q Sensor
Digital PIN-14	SCL/SCK (Air Q Sensor)	SCL/SCK (Air Q Sensor)
Digital PIN-4	Software Serial	Arduino PIN-3
Digital PIN-5	Software Serial	Arduino PIN-2
Digital PIN-12	Software Serial	Arduino PIN-10
Digital PIN-13	Software Serial	Arduino PIN-9

3. Air Quality Sensor (Adafruit CCS811)

Note: Sensor connected to ESP8266 using I2C interface

PIN	**Connection**
VCC	3.3 Volts on ESP8266
GND	Ground on ESP8266
SCL	ESP8266 Digital PIN-14
SDA	ESP8266 Digital PIN-2
WAKE	Ground on ESP8266

4. People Counter (Dual HC-SR04 ultrasonic sensors)

Note: Sensors connected to Arduino UNO.

a. Incoming Sensor

PIN	**Connection**
TRIG	Arduino Digital PIN-5
ECHO	Arduino Digital PIN-4
VCC	5 Volts
GND	Ground

b. Outgoing Sensor

PIN	**Connection**
TRIG	Arduino Digital PIN-7
ECHO	Arduino Digital PIN-6
VCC	5 Volts
GND	Ground

⇒ Final Notes

This project has taught me what is like to work with devices that have real world applications. It has also taught me the importance of optimization when it comes to designing IoT devices that are critical to the betterment of society. Doing this project has made me understand the challenges and trade-offs that come with designing a system crucial to human life. Overcoming overwhelming challenges was a difficult task, but it has given me an in depth view of the enormous IoT world around me. What I take from this project, the course and the challenges I faced is that I am confident and I believe that I have the knowledge and capability to design

and create interesting and life saving devices like the ones we saw over the duration of the course.
Thank You.