Embedded Networks Super Sensor

Omar Altawil 1047982

Maane Aldhanhani 1060207

Muhammad Zada 1031985

Supervised By: Eng. Yasmina Al Khalil

Submitted: February 21, 2018

Contents

List of Figures

List of Tables

Abstract

Technology is developing very fast, and a big part of technology that is on the rising is the Internet of Things(IoT). The IoT is where devices are connected together using the internet and speak with each other, they also can give each other commands. In this project we are going to exploit the usage of IoT and implement it so that sensors will record the data and then act accordingly depending on the analysis of data.

1 Introduction

This project is driven from the idea of having a smart home.A smart home is a house which is equipped with lighting, heating, and electronic devices that can be controlled remotely either by a smart phone or a computer.In order to turn the idea of smart home to reality, we need to have a number of sensors in the house. Here comes our super sensor, our idea is to make a chip that combines a various number of sensors in a small simple chip. A super sensor can be used to gather all sort of information within its perimeter. So basically, we have to place one in each room and cover the whole house, and then we can easily monitor the status of the house like the temperature, number of people in the house or even control the functions of some electronics like turn a device on or off.

2 Setup and Connection

2.0.1 BME280

The BME280 digital multifunction sensor developed specifically for where size and low power consumption are important,and this is the case in our super sensor. It can for Measuring TEMPERATURE,PRESSURE,HUMIDITY.

You can use either I2C or SPI wiring, although it's recommended to use SPI for speed. Heres we did wiring a feather MO to the sensor with I2c for simplicity. as shown in fig.

1. Step one, connect board 3V to sensor Vin.

2. Step two, connect board GND to sensor GND.

3. Step three, connect SCL to sensor SCK.

4. Step four, connect SDA to sensor SDI.

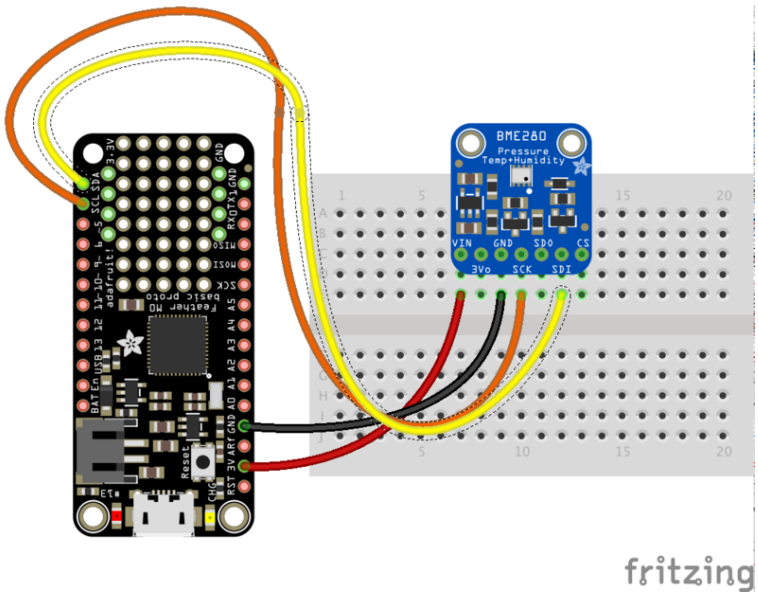

Figure 1:

2.1 Flame Sensor:

A flame sensor is a sensor that designed to respond and sense whenever there is a flame next to it.flame sensor dose not need any protocol to communicate you just can read the value directly. fig 2 show the connection to an arduino.

1. Step one, connect board 3V to sensor long leg.

2. Step two, connect board GND to sensor small leg throw a 40k resistor to eliminate noise.

3. Step three, connect A3 pin to small leg of the sensor.

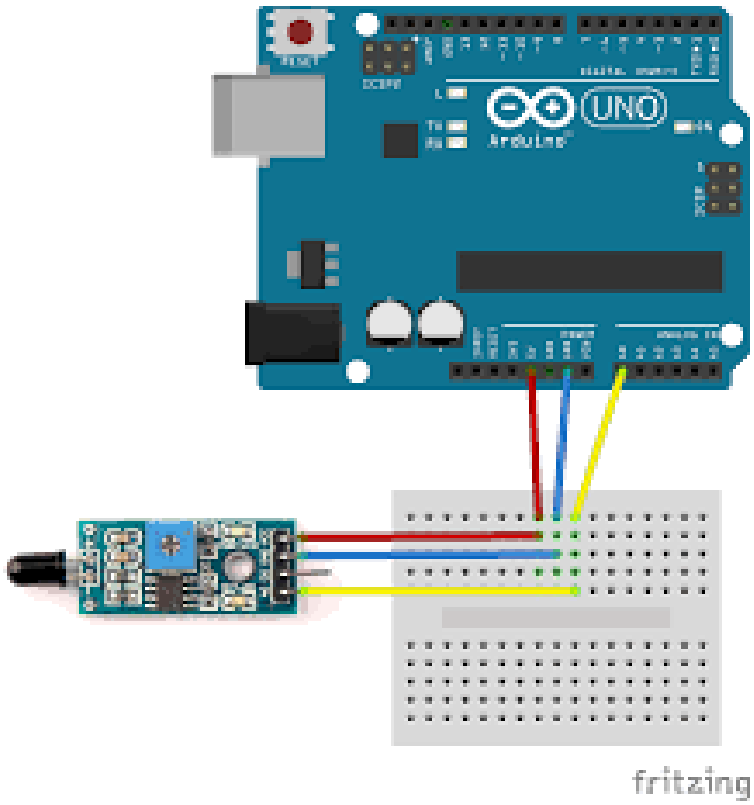

Figure 2:

2.2 LSM9DS1

The LSM9DS1 is a very power full motion sensor.It allow you to get a 3-axis accelerometer, 3-axis gyroscope, and 3-axis magnetometer,on top on top of all that it is very accurate.

Also it offer I2C and SPI connaction,but in this project we will stick to I2C.

1. Step one, connect board 3V to sensor Vin.
2. Step two, connect board GND to sensor GND.
3. Step three, connect SCL pin to sensor SCL.
4. Step four, connect SDA pin to sensor SDA.

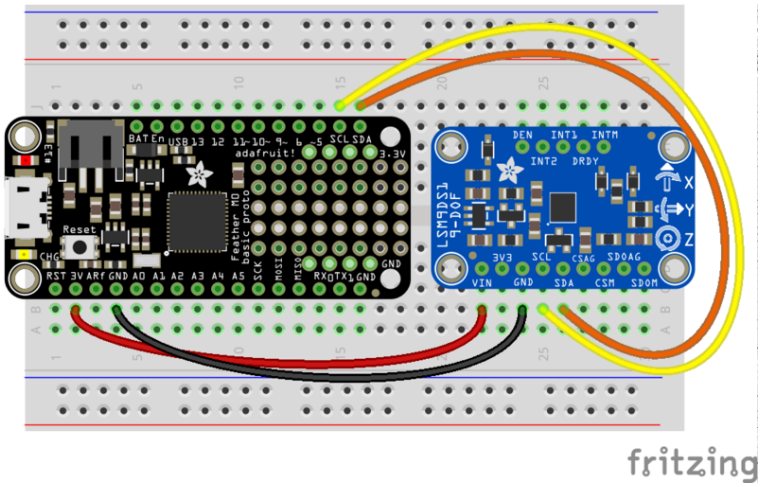

Figure 3:

2.3 SPH0645

SPH0645 is high accuracy microphone breakout the use I2S stander(design for digital audi devices).

1. Step one, connect board 3V to sensor 3V.
2. Step two, connect board GND to sensor GND.
3. Step three, connect TX pin to sensor BCLK.
4. Step four, connect pin 9 to sensor DOUT.
5. Step fifth, connect RX pin to sensor LRCL.

2.4 TSL2561

The TSL2561 luminosity sensor is light sensor,it is able to detect a wide rang of light reading.it use I2C protocol to communicate.

1. Step one, connect board 3V to sensor Vin.
2. Step two, connect board GND to sensor GND.
3. Step three, connect SCL pin to sensor SCL.
4. Step four, connect SDA pin to sensor SDA.

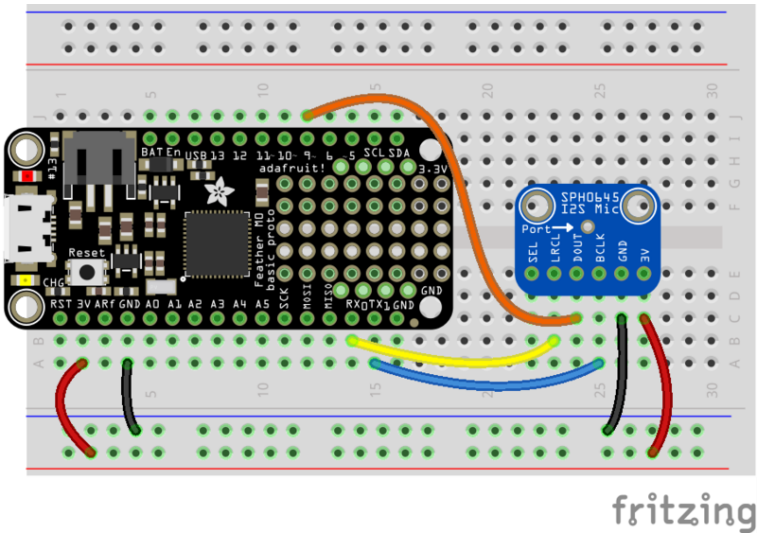

Figure 4:

3 Procedure

4 Machine Learning

One of the main goals of this project is to be able to sense what's going on in a specific room exactly with just a little chip in hand, so to achieve this we must use machine learning technique to train our sensor on the different event that may occur in different situation.so for this project we decided to start train our sensor using the mic to be able to sense the water coming out of the faucet and maybe we can train it to calculate how much water has been drooped,and all of that just depending on the sound.

So to achieve that we need to do some simple steps.

1. Step one, collect the data.
2. Step two, Store the data.
3. Step three, Train the collected data.

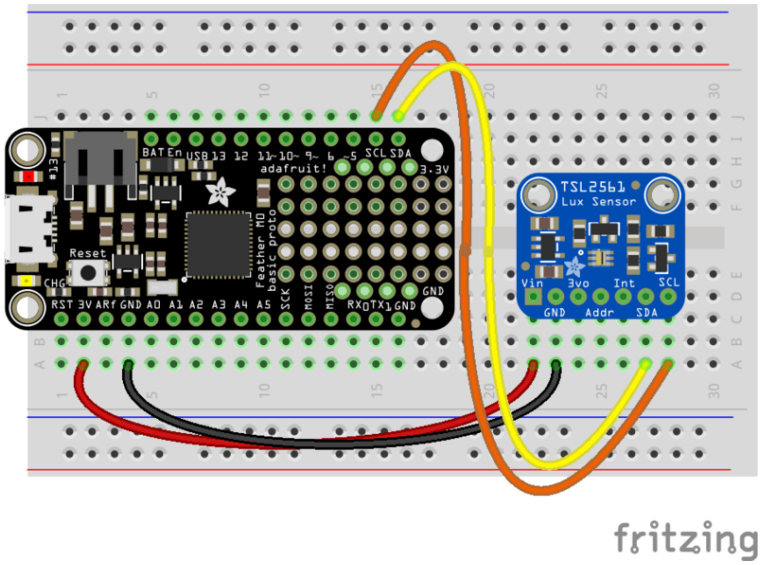

Figure 5:

4.1 collecting the data

You might ask why do we need to create our own dataset,there is a lot of website that offer all kind of datasets like FiveThirtyEight,Socrata OpenData and others.Also it will be way lager than yours?.will the answer is simple we can,but unfortunately i still don't know how to deal with sound datasets specially ,and also i am not sure if these datasets will work for our specific mic,so to avoid unnecessary wast of time i decided to create my own dataset using a mic and buttons.The code explained below..

```
1
2  //we include I2S(standard used for connecting digital audio
       devices together)
3  #include <I2S.h>
4  void setup() {
5    //we set the Serial to 115200 baud
6    Serial.begin(115200);
7    //we enable communication and pass (mode, sampleRate,
         bitsPerSample)
8    I2S.begin(I2S_PHILIPS_MODE, 17000, 32);
9    //we initialize pin A0(button) as Input
```

```
  pinMode(A0,INPUT);

}
 int i;
void loop() {
    //check if the button pressed
    if(digitalRead(A0)){
       //we start reading the value from the sensors
       int sample = I2S.read();
       //we filter 0 and -1 from the collected samples
       if ((sample == 0) || (sample == -1) ) {return;}
       // convert to 18 bit signed
       sample >>= 14;
       //we print the sample so the pySerial can read it
       Serial.println(sample);
       //increment i(i is number of sample capture)
        i++;
    //check if number of sample capture is 255
       if(i%255==254){
        //if the first swith is ON
         if(!digitalRead(A1)){
            //send label as 0 (water OFF)
            Serial.println(0);
            //wait 2S befor capture other data
            delay(2000);
         }
         //if the first swith is OFF
         else{
         //if the second switch is ON
         if(! digitalRead(A2)){
            //send label as 1 (water half ON)
            Serial.println(1);
            //wait 2S befor capture other data
            delay(2000);
         }
         //if the second switch is OFF
         else{
            //send label as 2 (water full ON)
            Serial.println(2);
            //wait 2S befor capture other data
```

```
            delay(2000);
          }
          }
        }

    }
}
```

4.2 Storing the data

After we collected the data and send it we need to receive it and store it in a table ,so we can do the training on it later .what i did was is i used the serial library to allow the python to get all the data from the serial monitor and i store it inside an Excel sheet using xlwt librar.At the end i collected 100 samples 40 water on high,30 water on low and 30 water off .The code explained below(you find the final dataset in the appendix below).

```
xlwt(a library for developers to use to generate spreadsheet files
    compatible with Microsoft Excel)
from xlwt import Workbook
Serial(it allow as to collact data from the serial)
import serial
#initialize a variable arduino that will be listing on COM8
arduino = serial.Serial("COM8", timeout=1)
#initialize Workbook object
wb=Workbook()
#we add a new sheet called "wataer sound" to the exal sheet
data_set_sheet=wb.add_sheet("wataer sound",cell_overwrite_ok=True)
#will run throw the 256 column that we have in the exal sheet
for i in range(255):
 #will name each feature by a number
 data_set_sheet.write(0,i,i+1)
#last column will be for the label
data_set_sheet.write(0,256,"label")

#c is number of row we will have on each run
c=0
#count is number of column
count = 0
```

```
#run for ever
while True:
 #reset the count to 0
 count = 0
 #if count less than 256
 while count < 256:
    #read from the serial and stor it in value
    value=str(arduino.readline())
    # we check if value is not empty
    if value != "":
     #we print value(not important)
     print(value)
     # we write value to the spicific loaction in the exal sheet
     data_set_sheet.write(c, count, int(value))
     #we move to the next column
     count += 1
 #we move to the next row
 c +=1
 #check if row ==10
 if c==10:
  #we save the file
  wb. save("AI wataer")
  #we exit the program
  break
```

4.3 Training the data set

Now we have the data set ready we need to do the important part which is the Training and testing.this part hard and easy at the same time easy because you don't actually write code you just copy past,but hard because you need to know which technique to use.what i did her was i used KNeighborsClassifier technique because it is the only technique i know,but obviously this is wrong because there is no relationship between the features because the sound waves can be captured any time and it can have any value.The code explained below.

```
#numpy(NumPy is the fundamental package for scientific computing
    with Python)
import numpy as np
```

```
#sklearn(scikit-learn is a Python module for machine learning
    built on top of SciPy)
from sklearn import preprocessing, cross_validation, neighbors
#panda(easy-to-use data structures and data analysis tools for the
    Python)
import pandas as pd
#store the data set in df
df = pd.read_csv('data.csv')
#x will contain the features(1-255 column)
X = np.array(df.drop(['label'],1))
#y will contain the label(256 column)
y = np.array(df['label'])

#split the data to training data(80%) and testing data(20%)
X_train, X_test, y_train, y_test =
    cross_validation.train_test_split(X, y, test_size=0.2)
#we use KNeighborsClassifier to train the data(not a good idea)
clf = neighbors.KNeighborsClassifier()
#the result of the training
clf.fit(X_train, y_train)
#find the accuracy based on the tested data
accuracy= clf.score(X_test, y_test)
#print out the accuracy
print(accuracy)
```

5 Monitoring

This part should be connected with machine learning part ,but because we are not done with training we did not add it her but it can be add easily.what we need to achieve is to be able to Monitor all the events that are happening in your house throw your mobile phone.

1. Step one, collect the data.
2. Step two, Store the data in firebase.
3. Step three,display the data in a mobile app.

5.1 collecting the data

In here we simply just getting the reading of each sensor and display it on the serial monitor.but we do not want to keep sending data all the time because first we are wasting space and second this will make the M0 busy all the time like for example we don't want the microcontroller to be sending temperature information while there is a fire because the temperature is not going to change dramatically just in few second,so we need to do it in a smart way.The code explained below.

```
//#include <SparkFunLSM9DS1.h>
//#include <I2S.h>
//include libraries for BME280
#include <stdint.h>
#include "SparkFunBME280.h"
//include wire librarie(I2C protocal)
#include <Wire.h>
//include libraries for TSL2561
#include <Adafruit_Sensor.h>
#include <Adafruit_TSL2561_U.h>
//LSM9DS1 imu;

//setup TSL2561
Adafruit_TSL2561_Unified tsl =
    Adafruit_TSL2561_Unified(TSL2561_ADDR_FLOAT, 12345);
//setup BME280
BME280 mySensor;
void setup() {
  //we set the Serial to 115200 baud
  Serial.begin(115200);
//  I2S.begin(I2S_PHILIPS_MODE, 17000, 32);
//  imu.settings.device.commInterface = IMU_MODE_I2C;
//  imu.settings.device.mAddress = LSM9DS1_M;
//  imu.settings.device.agAddress = LSM9DS1_AG;
//  imu.begin();
  //we do the settings for BME280(protocol is I2C)
  mySensor.settings.commInterface = I2C_MODE;
  mySensor.settings.I2CAddress = 0x77;
  mySensor.settings.runMode = 3; //Normal mode
  mySensor.settings.tStandby = 0;
```

```
  mySensor.settings.filter = 0;
  mySensor.settings.tempOverSample = 1;
  mySensor.settings.pressOverSample = 1;
  mySensor.settings.humidOverSample = 1;
 //we enable communication for the BME280
  mySensor.begin();}
// int i;
 //variable to store the old light sensor value
 int oldlight=0;
 //variable to store the old flame sensor value
 int oldflame=0;
 //variable to store the last time we got a reading from BME sensor
 int bmeTime=0;
 //variable to store the last time we got a reading from light
    sensor
 int lightTime=0;
 //variable to store the last time we got a reading from flame
    sensor
 int flameTime=0;
void loop() {
//***************we did not use the mic and the Lms9ds in the final
   code becouse the AI part is not done so the mic data right now
   is useless for us.***************
//int sample = I2S.read();
//imu.readGyro();
//imu.readAccel();
//imu.readMag();
//if ((sample == 0) || (sample == -1) ) {return;}
//sample >>= 14;
//Serial.println("1");
//Serial.println(sample);
//Serial.println("2");
// Serial.print(imu.calcGyro(imu.gx), 2);
// Serial.print(imu.calcGyro(imu.gy), 2);
// Serial.println(imu.calcGyro(imu.gz), 2);
//Serial.println("3");
// Serial.print(imu.calcAccel(imu.ax), 2);
// Serial.print(imu.calcAccel(imu.ay), 2);
// Serial.println(imu.calcAccel(imu.az), 2);
//Serial.println("4");
```

```
// Serial.print(imu.calcMag(imu.mx), 2);
// Serial.print(imu.calcMag(imu.my), 2);
// Serial.println(imu.calcMag(imu.mz), 2);
//i++;
//if(i%255==254){
    //create a variable event of type sensors_event_t
    sensors_event_t event;
    tsl.getEvent(&event);
    //get the intensity of the light
    int light=event.light;
    //get the flame sensor value
    int flame=analogRead(A3);
    //check if the value of the light increased or decreased dramatically and 5S pass since the last reading
    if((light> oldlight*2 && lightTime==50 ) || (oldlight>light*2 && lightTime==50)){
      //send 1 so the python know that the next value are for the light
      Serial.println("1");
      //send the light value
      Serial.println(light);
      //update the old value
      oldlight=light;
      //reset the time for light
      lightTime=0;
    }
    //check if the value of the flame increased or decreased dramatically and 5S pass since the last reading
    if((flame>oldflame*2 && flameTime==50) || (oldflame>flame*2 && flameTime==50)){
      //send 2 so the python know that the next value are for the flame
      Serial.println("2");
      //send the flame value
      Serial.println(flame);
      //update the flame value
      oldflame=flame;
      //rest the time for flame
      flameTime=0;
    }
```

```
100     //if 20S pass(every 20S the update these value)
101     if(bmeTime==200){
102     //send 3 so the python know that the next value are for the
            TempC,pressure,humidity and altitude
103     Serial.println("3");
104     //send the the temp
105     Serial.println(mySensor.readTempC());
106     //send the Pressure
107     Serial.println(mySensor.readFloatPressure());
108     //send the humidity
109     Serial.println(mySensor.readFloatHumidity());
110     //send the altitude
111     Serial.println(mySensor.readFloatAltitudeMeters());
112     //rest the time
113     bmeTime=0;
114     }
115     //increment the time
116     bmeTime++;
117     lightTime++;
118     flameTime++;
119     //delay 0.1S(so bmeTime=200 = 20S ,lightTime and flameTime=50
            =5S)
120     delay(100);
121 //      }
122   }
```

5.2 Firebase

Now after we did receive the data from the microcontroller we need we send to the firebase so we can access it from the app,for that we use firebase library in python,but what if the user have more than one sensor in his home and also what if we have more than one user?.for that we ask the user to enter his gmail username and the room which the sensor in.The code explained below.

```
1 #Serial(it allow as to collact data from the serial)
2 import serial
3 #firebas(it allow us read and write to firebase)
4 from firebase.firebase import FirebaseApplication,
```

```
    FirebaseAuthentication
#firebase setting
SECRET = 'pyDYJf7bXrujgAysluibQyuZnRm4qRvV504HbJq6'
DSN = 'https://supers-87d35.firebaseio.com'
EMAIL = 'omar.y.altawil@gmail.com'
authentication = FirebaseAuthentication(SECRET,EMAIL, True, True)
firebase = FirebaseApplication(DSN, authentication)
#initialize a variable arduino that will be listing on COM5
arduino = serial.Serial("COM5", 9600)
#initialize youer gmail that you will acsses from
email="omar7altawil"
#initialize the room that you will setup the super sensor in
location ="livingroom"
# location ="bedroom"
# location ="bathroom"
#location ="kitchen"
"""Receiving data and storing it in a list"""
#run for ever
while True:
    #***************we are supposed to receive my information here
        and do analize for the data and send to the data base the
        label*************************
                                    #***************but AI part is
                                        not complete yet
                                        *************************


 #read the value coming from the arduino
 value=arduino.readline().strip()
 print (value)
 #if we recive 1
 if value=="1":
   #we will store the next value in light variable
   light=arduino.readline().strip()
   #we will post light to firebase to the specific user and location
   snapshot = firebase.patch('/users/'+email+'/'+location,
       {'light':light})
 #if we recive 2
 elif value=="2":
     # we will store the next value in flame variable
```

```
39      flame =arduino.readline().strip()
40      # we will post flame to firebase to the specific user and
            location
41      snapshot = firebase.patch('/users/' + email + '/' + location,
            {'flame': flame})
42  # if we recive 1
43   elif value=="3":
44      # we will store the next value in TempC variable
45      TempC=arduino.readline().strip()
46      # we will store the next value in Pressure variable
47
48      Pressure=arduino.readline().strip()
49      # we will store the next value in Humidity variable
50      Humidity=arduino.readline().strip()
51      # we will store the next value in Altitude variable
52      Altitude=arduino.readline().strip()
53      # we will post all the variable to firebase to the specific
            user and location
54      snapshot = firebase.patch('/users/' + email + '/' + location,
            {'TempC': TempC})
55      snapshot = firebase.patch('/users/' + email + '/' + location,
            {'Pressure': Pressure})
56      snapshot = firebase.patch('/users/' + email + '/' + location,
            {'Humidity': Humidity})
57      snapshot = firebase.patch('/users/' + email + '/' + location,
            {'Altitude': Altitude})
```

5.3 Mobile App

For this final part we used android studio to build our app. we need an app that allow as to monitor what is happening in a specific room in real time,and also we need it to allow different users to use it an monitor there houses,so for that we created a Google login page so we can authenticate the users,also we created a sensor class to store the value of each sensor in it.(you can see the full code in the appendix).

6 Results and Demo

this how the final circuit look like fig6.

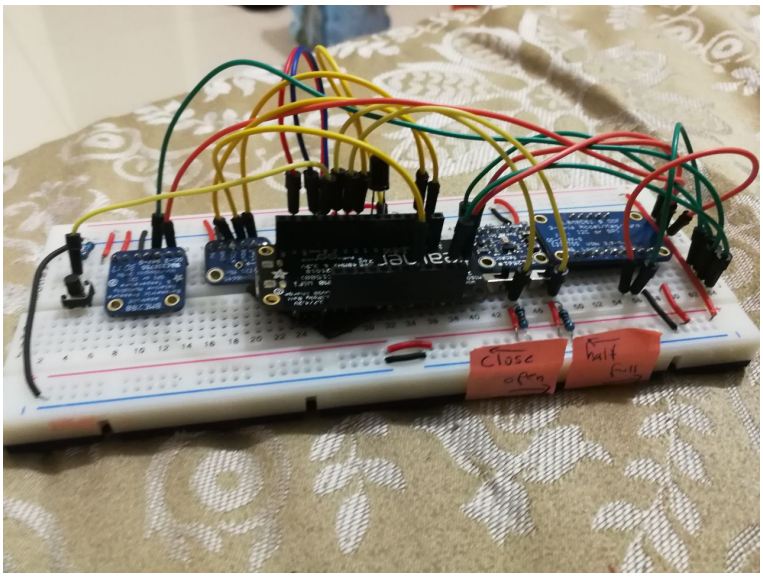

Figure 6:

This video will show the full functionality of the app:
video URL: https://www.youtube.com/watch?v=hbzRLlvM0kc

7 Conclusion

In this project we learned how to create a Super sensor. It was a very fun project where we learned may skills like, how to connect micro-controllers together using different protocols and how to send and receive data from the server and how to modify it on the server, also how to create an app and connect it to the server itself to show and control the data, we also learned a bit of machine learning techniques since we used python to get the data and analyze it to make it smarter and for it to understand different changes to the environment. In the end it was fun to do and we learned a lot doing this project.

8 Appendix

1. data.csv(dataset)
2. data.py.(python)
3. exal.py.(python)
4. Ai.py.(python)
5. final.ino(arduino)
6. collactingdata.ino(arduino)
7. superS.zip(android studio)