VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

SWE4005 - INTERNET OF THINGS

PROJECT TITLE: WEATHER MONITORING SYSTEM USING ARDUINO

J-COMPONENT FINAL REPORT

TEAM MEMBERS:

NAME	**REG NO.**
SAI NIKHIL	16MIS0257
K. JEEVAN KUMAR	16MIS0339
K. SAI KIRAN KUMAR	16MIS0391
K. ADITYA	16MIS0480

SUBMITTED TO :

POUNAMBAL. M

ABSTRACT:

Weather is the state of the atmosphere, to the degree that it is hot or cold, wet or dry, calm or stormy, clear or cloudy. Most weather phenomena occur in the troposphere, just below the stratosphere. Weather generally refers to day-to-day temperature and precipitation activity, whereas climate is the term for the average atmospheric conditions over longer periods of time. When used without qualification, "weather", is understood to mean the weather of earth. Monitoring the weather conditions manually is difficult. The present work is to develop an automated system which monitors the weather condition. The weather condition is driven by air pressure (temperature and moisture) differences between one place and another. These pressure and temperature differences can occur due to the sun angle at any particular spot. Through this system we can automatically collect the information about humidity and temperature. The details are stored in a database and according to current and previous data we can produce the results in graphical manner in the system. The objective of this project is to formulate the weather and be able to forecast the weather without human error and applying the machine learning algorithm for the collected data using backpropagation neural network algorithm which gives highest accuracy rate in which all household and farmers will be benefited.

INTRODUCTION :

Weather forecasting is the application of science and technology to predict the state of the atmosphere for a given location. Human beings have attempted to predict the weather informally for millennium and formally since the nineteenth century. Weather forecasts are made by collecting quantitative data about the current state of the atmosphere on a given place and using scientific understanding of atmospheric processes to project how the atmosphere will evolve on that place.

Weather is driven by air pressure (temperature and moisture) differences between one place and another. These pressure and temperature differences can occur due to the sun angle at any particular spot, which varies by latitude from the tropics. The atmosphere is a chaotic system, so small changes to one part of the system can grow to have large effects on the system as a whole. This makes it difficult to accurately predict weather more than a few days in advance, though weather forecasters are continually working to extend this limit through the scientific study of weather, meteorology. It is theoretically

impossible to make useful day-today predictions more than about two weeks ahead, imposing an upper limit to potential for improved prediction skill.

Once an all-human endeavor based mainly upon changes in barometric pressure, current weather conditions, and sky condition, weather forecasting now relies on computer-based models that take many atmospheric factors into account. Human input is still required to pick the best possible forecast model to base the forecast upon, which involves pattern recognition skills, tele-connections, knowledge of model performance, and knowledge of model biases.

DESIGN METHODOLOGY:

1) Purpose and Requirement Specification

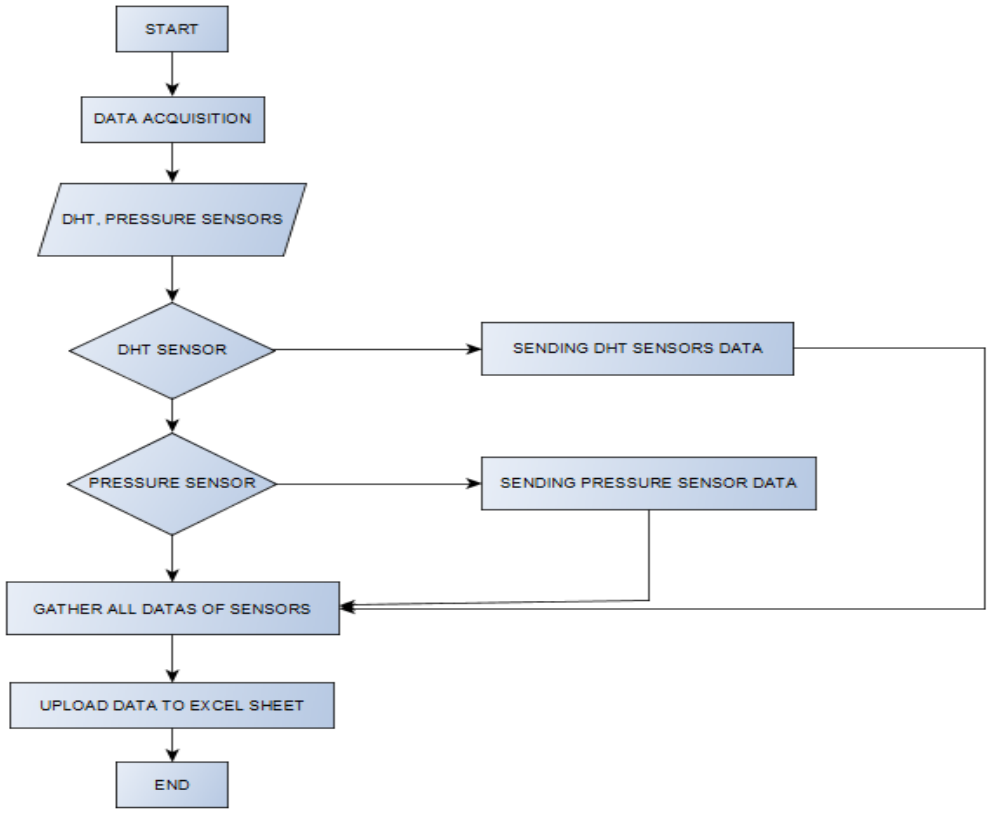
The main attributes have been chosen based on the sensors used to build the system in order to create an effective weather monitoring system. The proposed sensors are used to measure and store Temperature, Humidity and pressure data. The acquired data can be displayed in two ways identified as direct and indirect due to periodic data read and storing the data as real database system respectively. Real database creation technology is considered the main challenge of this work, which gives an opportunity to mine the data, recorded in the past. Furthermore, the entire system supervises and governs data locally based on the periodic change that occurs in the climate conditions.

Requirements:

1. Pressure Sensor

2. Temperature and humidity sensor (DHT Sensor)

2) Process specification (flow diagram)



3) Domain Model Specification

Physical entity

Weather data (calculated using temperature pressure and humidity data)

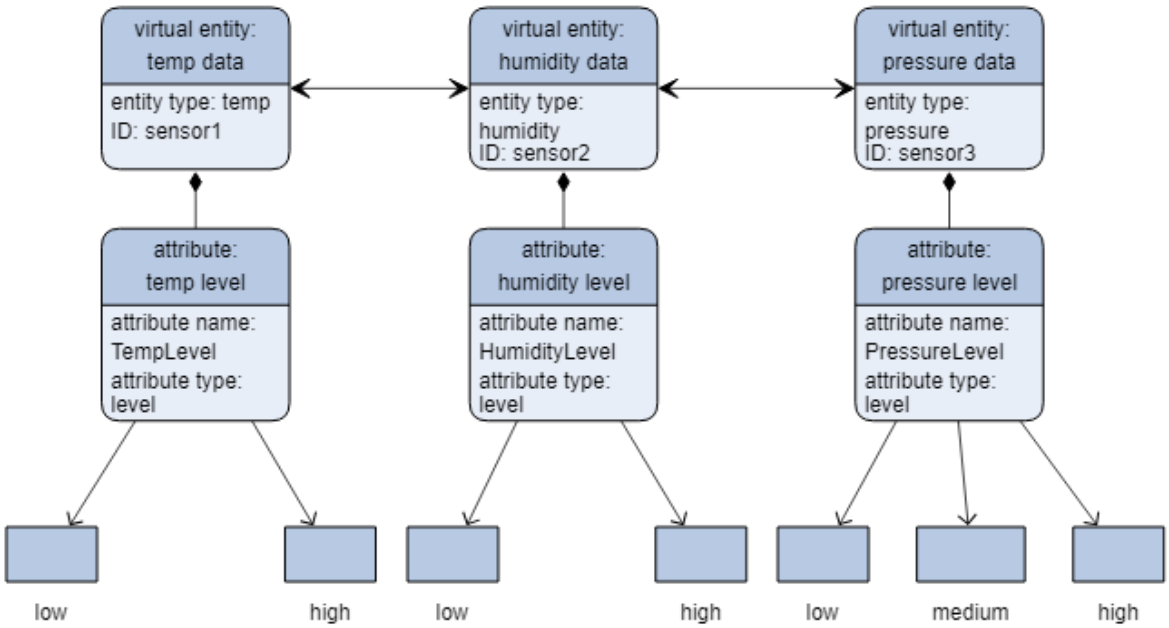
Virtual entity

Representation of physical entity in digital world using Thingspeak

Device

- Medium for interactions between Physical and Virtual Entities
- Sensors are used to gather information
- Arduino works as the CPU
- Wifi module is used to store, analyze and retrieve result from cloud

4) Information Model Specification



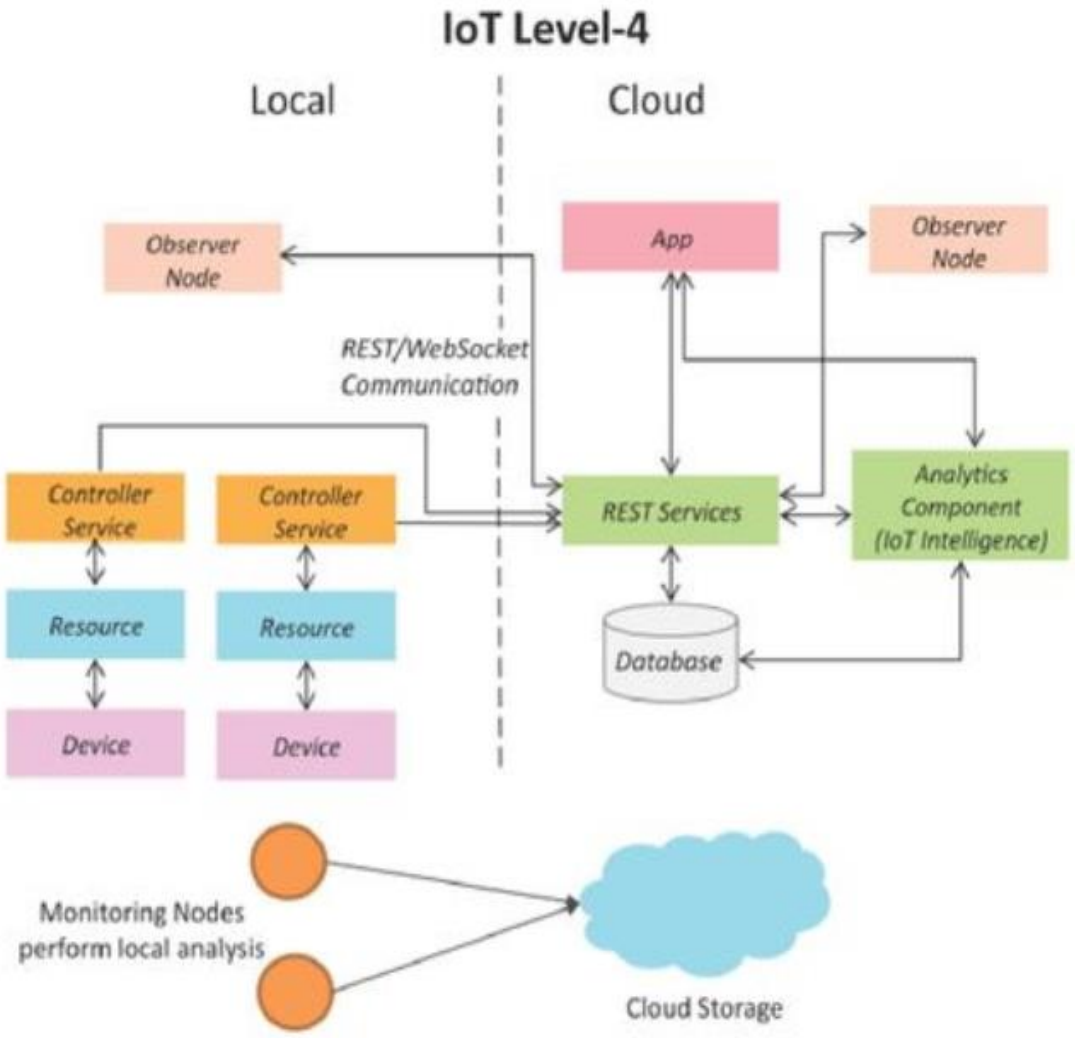
5) Service Specification

The services provided by the application are:

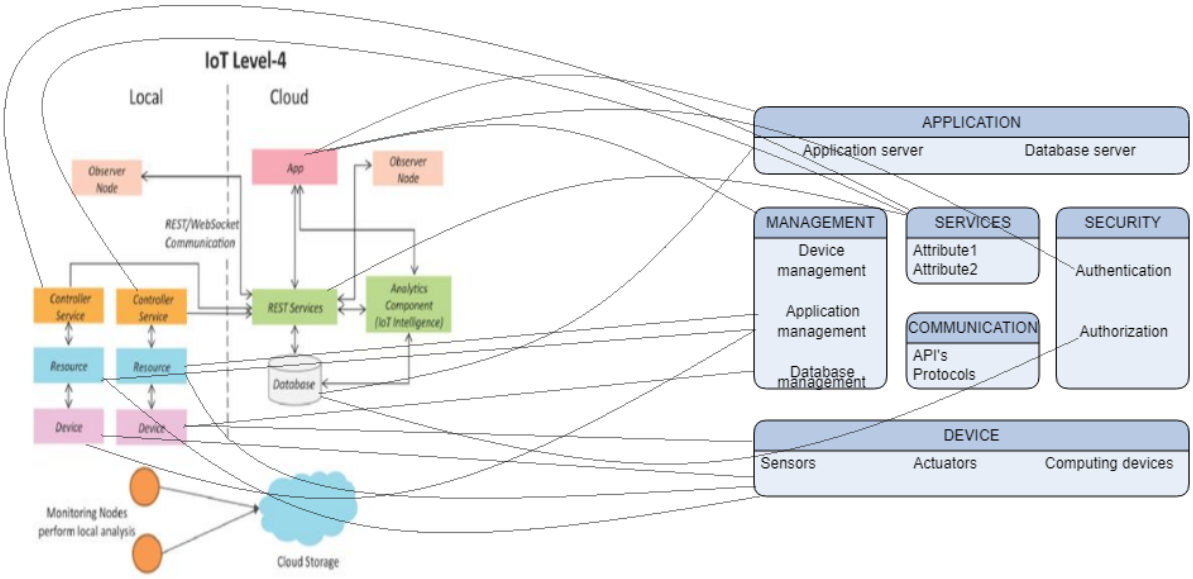
- The temperature, humidity and pressure data is read from the atmosphere by the help of sensors (DHT, PRESSURE).
- Data is read to Arduino device to store it on a Cloud Database.
- Data is monitored to keep the temperature in check. Data accessibility is a plus, as it is on the cloud.
- Temperature for maintaining the history and to take precautions in future.

6) IoT Level Specification

In the proposed solution we have used 2 sensors which would sense temperature, humidity and pressure. Hence as the no. of sensors increase size of data will increase that needs to be stored in the cloud. Hence we use IOT level-4 for this weather monitoring system.



7)Functional view specification



8) Operational View Specification

The operational view is the implementation of the content mentioned in the functional View specification.

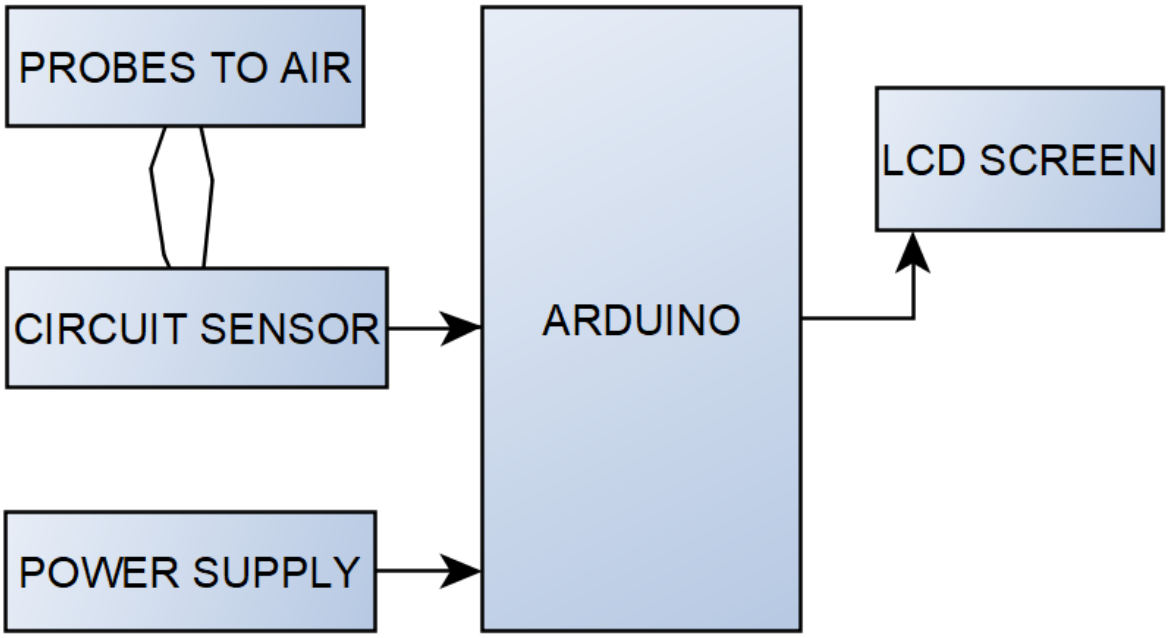
The prerequisites to implement the functionality should be clearly specified.

The data is used to check if the required temperature is being maintained or not.

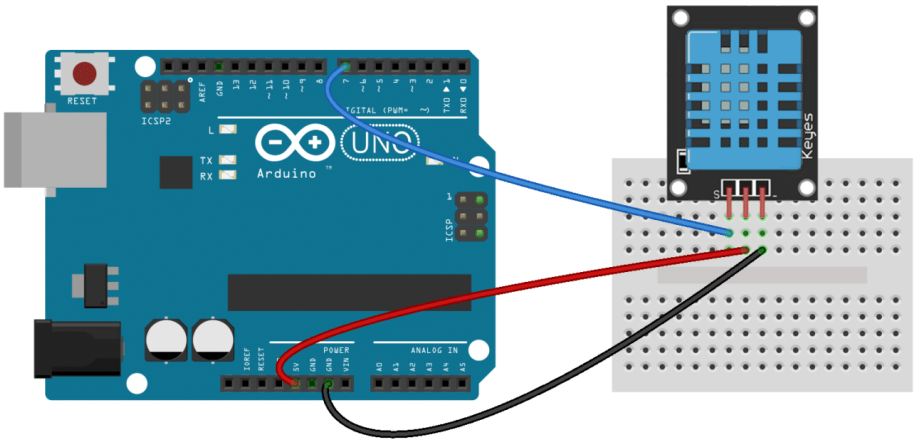
The temperature Sensors (DHT11) and pressure sensors are used to collect the data. Then send the data collected, to the database. Analyze the data based on few predefined temperature conditions (Predefined conditions include the temperature range specification). Based on the analysis of temperature data collected, we monitor the thermostat to check the current temperature adjust the air conditioner to maintain the required temperature at particular locations. We save the data for further analysis.

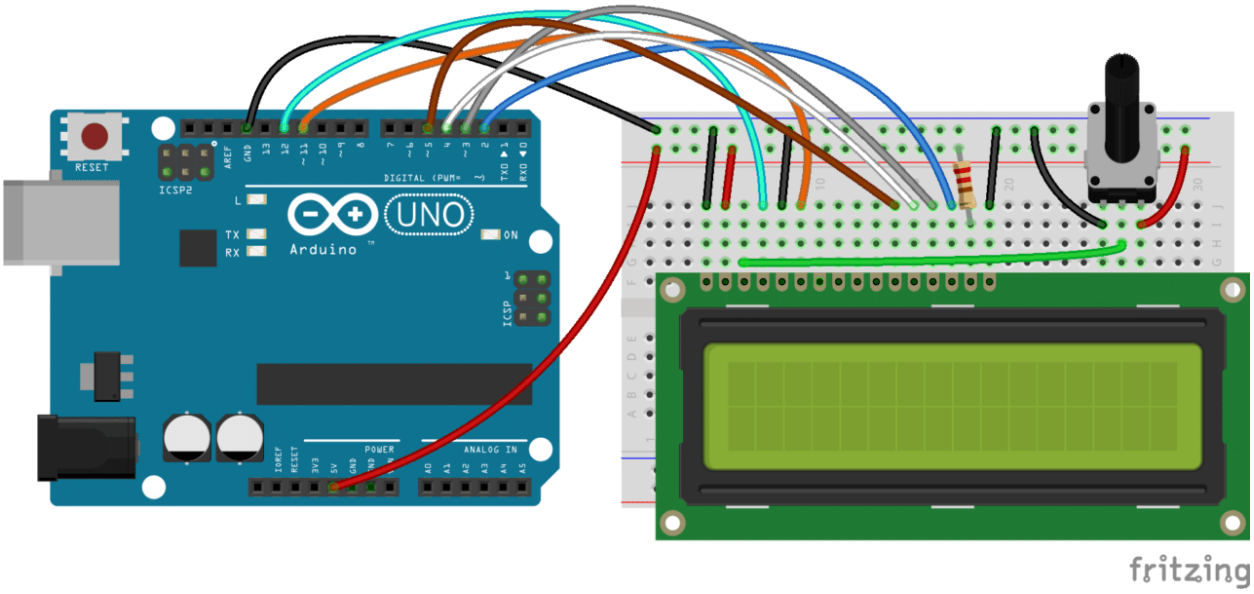
Finally, all the operations which are independent without the ability to communicate with each other can be made into a complete set of operational technology with the help of communication protocol called the **MQTT** protocol.

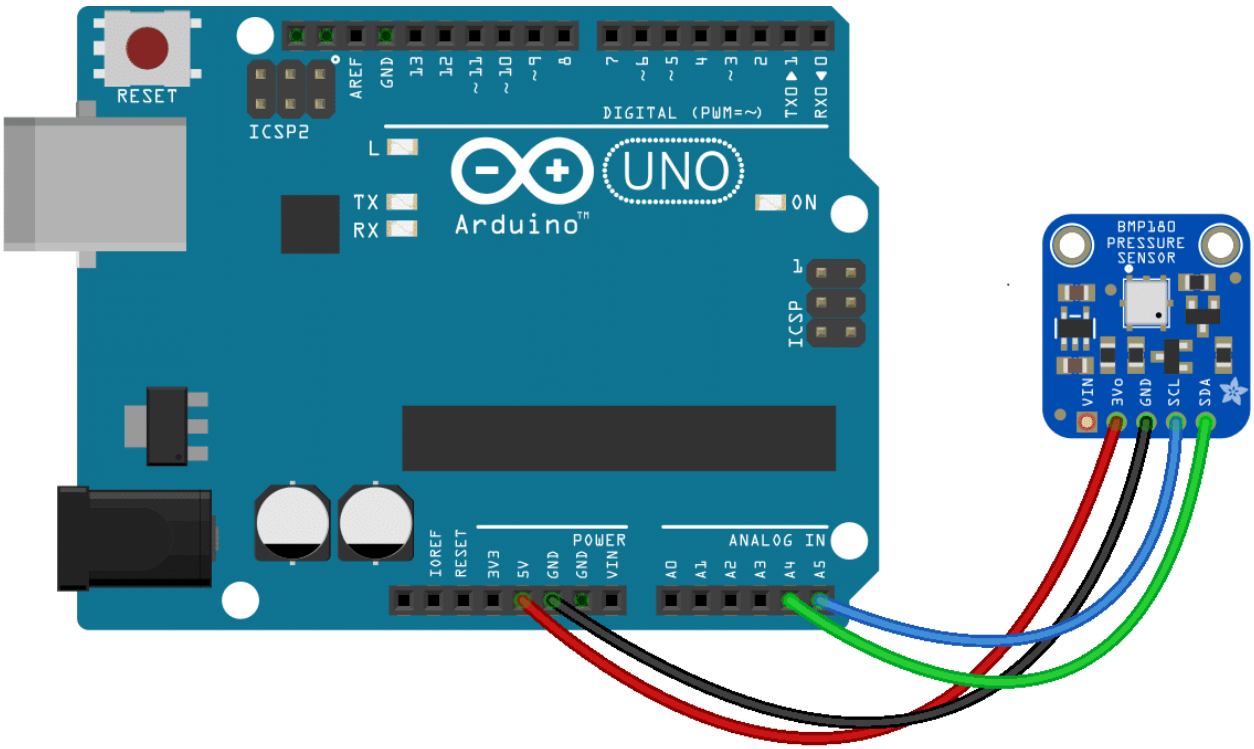
9)Device and Component Integration : Alternate Diagram



10) Application development




RESET
ICSP2
AREF
GND
DIGITAL (PWM=~)
TXD
RXD
L
TX
RX
Arduino
UNO
ON
ICSP
POWER
ANALOG IN
fritzing


RESET
ICSP2
AREF
GND
DIGITAL (PWM=~)
TXD
RXD
L
TX
RX
Arduino
UNO
ON
ICSP
POWER
ANALOG IN
IOREF
RESET
3V3
5V
GND
VIN
A0
A1
A2
A3
A4
A5
BMP180
PRESSURE
SENSOR
VIN
3VO
GND
SCL
SDA

Arduino code:

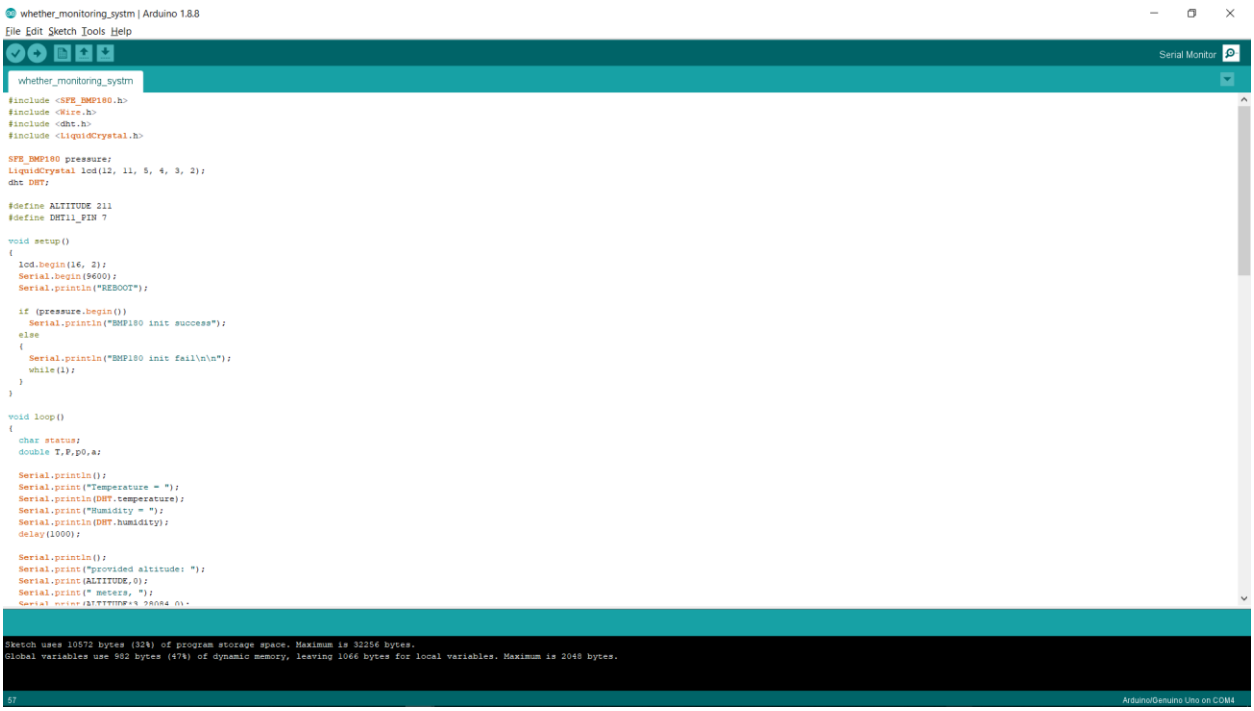
```
#include <SFE_BMP180.h>
#include <Wire.h>
#include <dht.h>
#include <LiquidCrystal.h>

SFE_BMP180 pressure;
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
dht DHT;

#define ALTITUDE 211
#define DHT11_PIN 7

void setup()
{
 lcd.begin(16, 2);
 Serial.begin(9600);
 Serial.println("REBOOT");

 if (pressure.begin())
  Serial.println("BMP180 init success");
 else
 {
  Serial.println("BMP180 init fail\n\n");
  while(1);
 }

 Serial.println();
 Serial.print("Temperature = ");
```

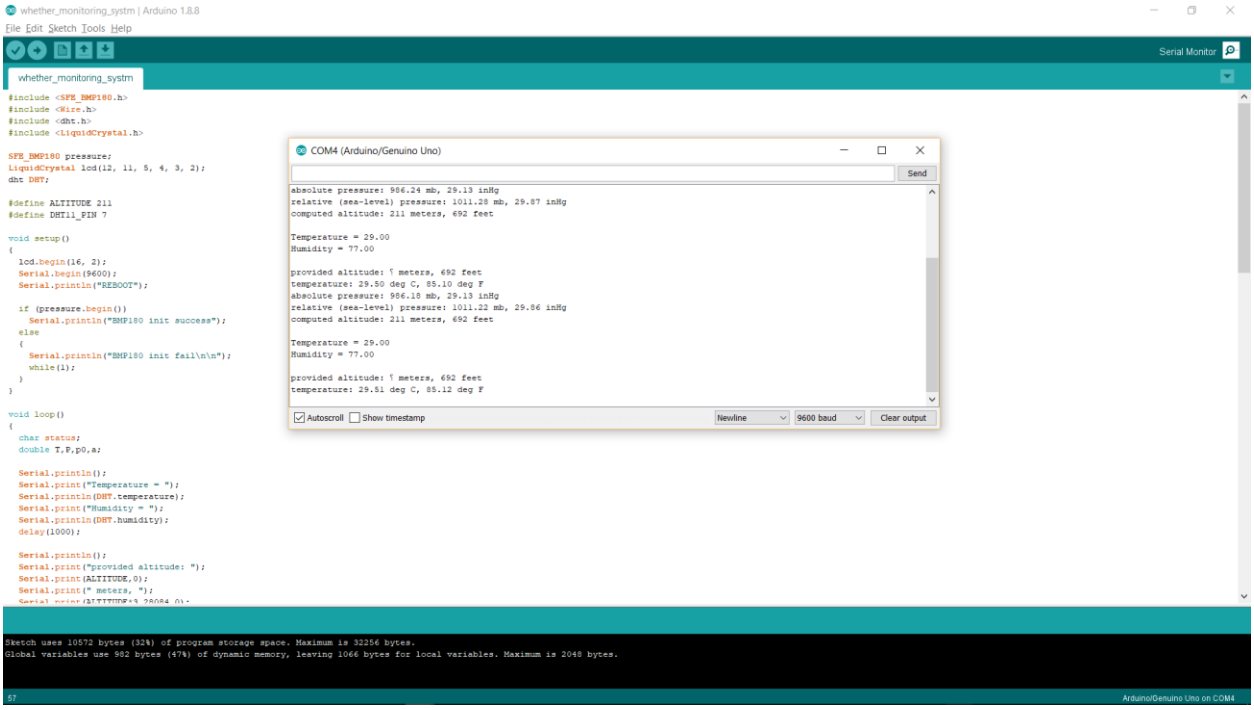
```
  Serial.println(DHT.temperature);
  Serial.print("Humidity = ");
  Serial.println(DHT.humidity);
  delay(1000);

  Serial.println();
  Serial.print("provided altitude: ");
  Serial.print(ALTITUDE,0);
}

void loop()
{
  char status;
  double T,P,p0,a;
  Serial.print(" meters, ");
  Serial.print(ALTITUDE*3.28084,0);
  Serial.println(" feet");

  status = pressure.startTemperature();
  if (status != 0)
  {
    delay(status);
    status = pressure.getTemperature(T);
    if (status != 0)
    {
      Serial.print("temperature: ");
      Serial.print(T,2);
      Serial.print(" deg C, ");
      Serial.print((9.0/5.0)*T+32.0,2);
```

```
Serial.println(" deg F");
lcd.setCursor(0,0);
lcd.print("Tempe: ");
lcd.print(T,2);
lcd.print("deg C");
delay(2000);

int chk = DHT.read11(DHT11_PIN);
lcd.setCursor(0,0);
lcd.print("Humidity: ");
lcd.print(DHT.humidity);
lcd.print("%");
delay(1000);


status = pressure.startPressure(3);
if (status != 0)
{
  delay(status);
  status = pressure.getPressure(P,T);
  if (status != 0)
  {
    Serial.print("absolute pressure: ");
    Serial.print(P,2);
    Serial.print(" mb, ");
    Serial.print(P*0.0295333727,2);
    Serial.println(" inHg");
    lcd.setCursor(0,0);
    lcd.print("Abs Pres: ");
```

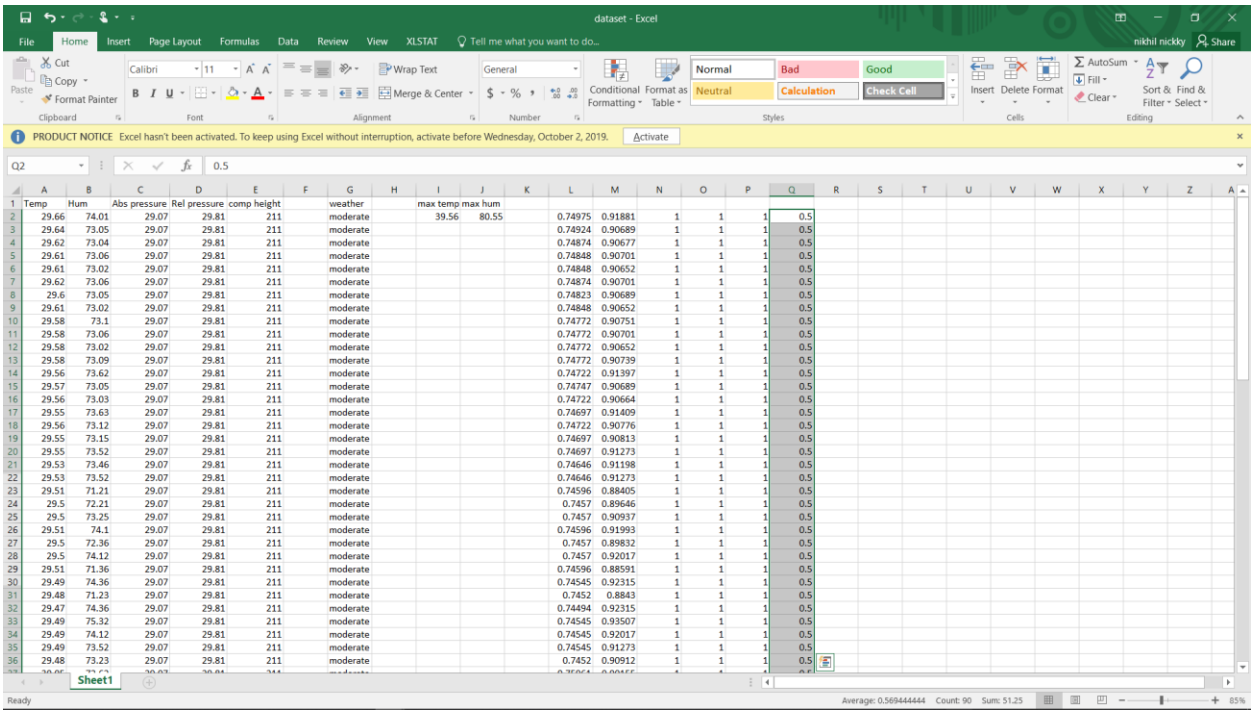
```
        lcd.print(P,2);
        lcd.print("mb");
        delay(1000);

        p0 = pressure.sealevel(P,ALTITUDE);
        Serial.print("relative (sea-level) pressure: ");
        Serial.print(p0,2);
        Serial.print(" mb, ");
        Serial.print(p0*0.0295333727,2);
        Serial.println(" inHg");
        lcd.setCursor(0,0);
        lcd.print("Com Pres: ");
        lcd.print(a,2);
        lcd.print("mts");
        delay(1000);

      }
      else Serial.println("error retrieving pressure measurement\n");
    }
    else Serial.println("error starting pressure measurement\n");
  }
  else Serial.println("error retrieving temperature measurement\n");
}
else Serial.println("error starting temperature measurement\n");
        setCursor(0,0);
        lcd.print("Rel Pres: ");
        lcd.print(p0,2);
        lcd.print("mb");
        delay(1000);
```

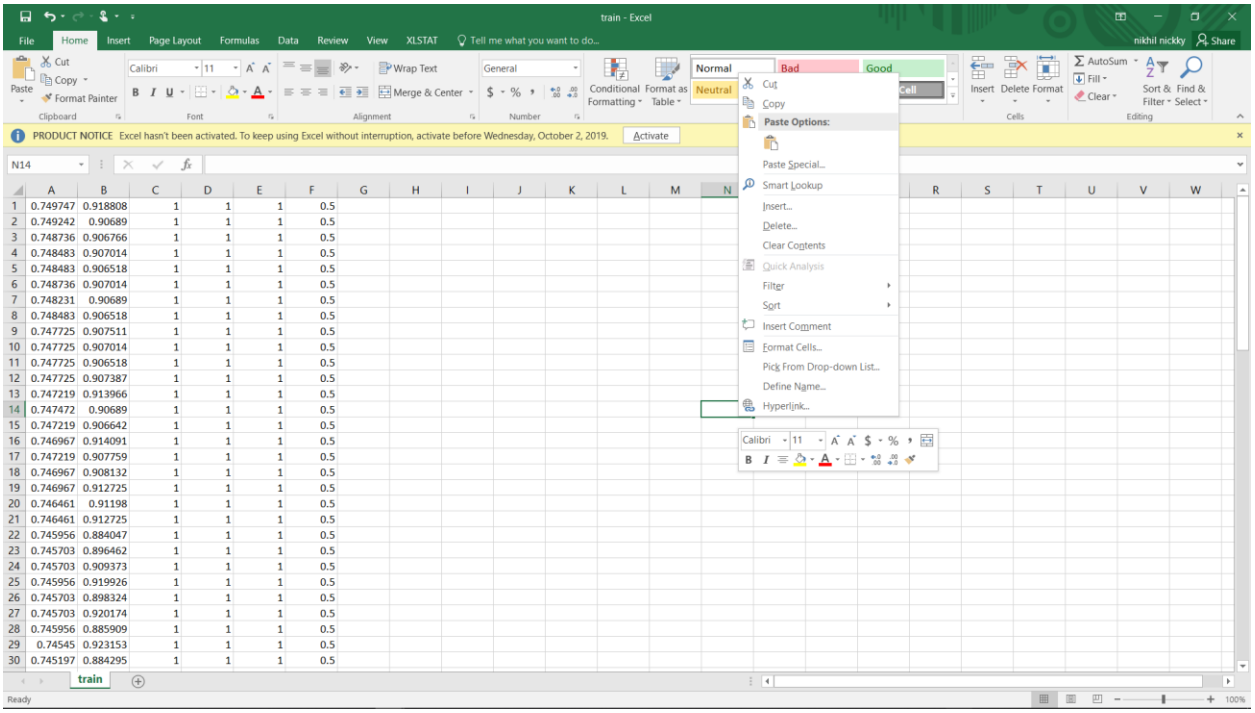
```
        a = pressure.altitude(P,p0);
        Serial.print("computed altitude: ");
        Serial.print(a,0);
        Serial.print(" meters, ");
        Serial.print(a*3.28084,0);
        Serial.println(" feet");
        lcd
  delay(1000);
}
```

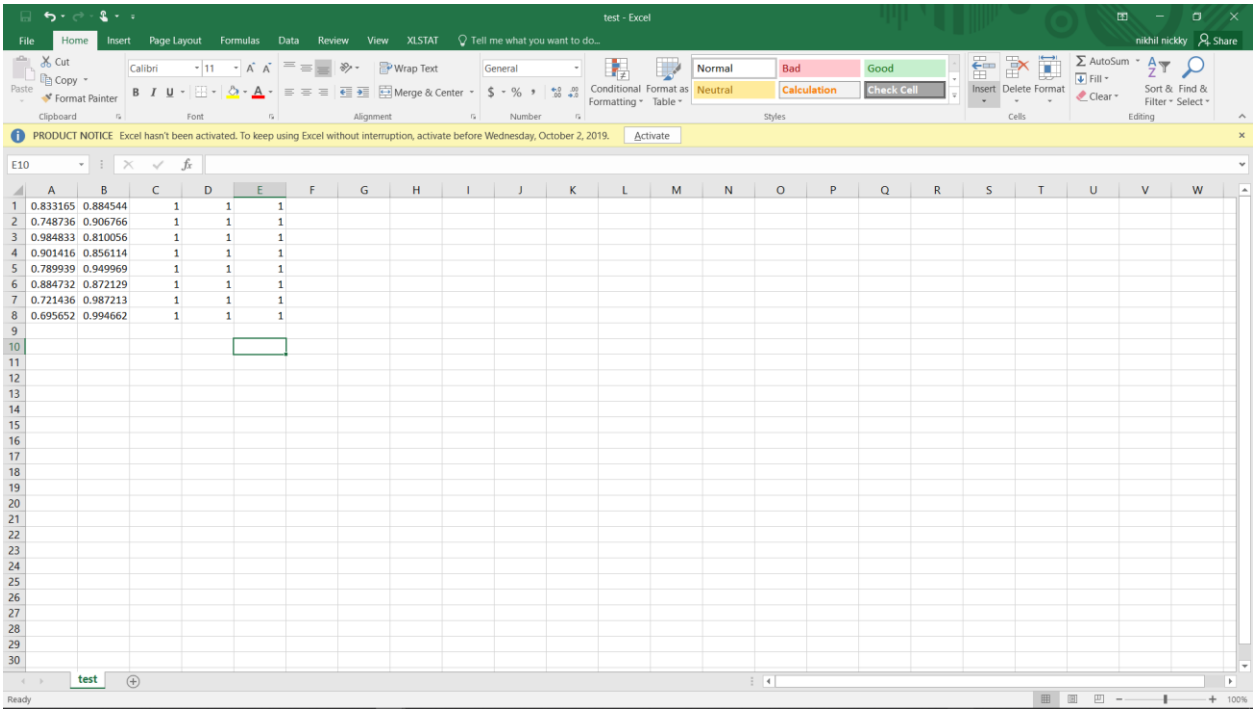



whether_monitoring_systm | Arduino 1.8.8
File Edit Sketch Tools Help
Serial Monitor
whether_monitoring_systm
#include <SFE_BMP180.h>
#include <Wire.h>
#include <dht.h>
#include <LiquidCrystal.h>
SFE_BMP180 pressure;
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
dht DHT;
#define ALTITUDE 211
#define DHT11_PIN 7
void setup()
{
lcd.begin(16, 2);
Serial.begin(9600);
Serial.println("REBOOT");
if (pressure.begin())
Serial.println("BMP180 init success");
else
{
Serial.println("BMP180 init fail\n\n");
while(1);
}
}
void loop()
{
char status;
double T,P,p0,a;
Serial.println();
Serial.print("Temperature = ");
Serial.println(DHT.temperature);
Serial.print("Humidity = ");
Serial.println(DHT.humidity);
delay(1000);
Serial.println();
Serial.print("provided altitude: ");
Serial.print(ALTITUDE,0);
Serial.print(" meters, ");
COM4 (Arduino/Genuino Uno)
Send
absolute pressure: 986.24 mb, 29.13 inHg
relative (sea-level) pressure: 1011.28 mb, 29.87 inHg
computed altitude: 211 meters, 692 feet
Temperature = 29.00
Humidity = 77.00
provided altitude: 211 meters, 692 feet
temperature: 29.50 deg C, 85.10 deg F
absolute pressure: 986.18 mb, 29.13 inHg
relative (sea-level) pressure: 1011.22 mb, 29.86 inHg
computed altitude: 211 meters, 692 feet
Temperature = 29.00
Humidity = 77.00
provided altitude: 211 meters, 692 feet
temperature: 29.51 deg C, 85.12 deg F
Autoscroll
Show timestamp
Newline
9600 baud
Clear output
Sketch uses 10572 bytes (32%) of program storage space. Maximum is 32256 bytes.
Global variables use 982 bytes (47%) of dynamic memory, leaving 1066 bytes for local variables. Maximum is 2048 bytes.
57
Arduino/Genuino Uno on COM4

Datasets :


dataset - Excel
File Home Insert Page Layout Formulas Data Review View XLSTAT Tell me what you want to do...
nikhil nickky Share
Cut Copy Format Painter
Calibri 11
Wrap Text
General
Merge & Center
Conditional Formatting
Format as Table
Normal Bad Good
Neutral Calculation Check Cell
Insert Delete Format
AutoSum Fill Clear
Sort & Filter Find & Select
Clipboard Font Alignment Number Styles Cells Editing
PRODUCT NOTICE Excel hasn't been activated. To keep using Excel without interruption, activate before Wednesday, October 2, 2019.
Activate
Q2
0.5
Temp Hum Abs pressure Rel pressure comp height weather max temp max hum
29.66 74.01 29.07 29.81 211 moderate 39.56 80.55 0.74975 0.91881 1 1 1 0.5
29.64 73.05 29.07 29.81 211 moderate 0.74924 0.90689 1 1 1 0.5
29.62 73.04 29.07 29.81 211 moderate 0.74874 0.90677 1 1 1 0.5
29.61 73.06 29.07 29.81 211 moderate 0.74848 0.90701 1 1 1 0.5
29.61 73.02 29.07 29.81 211 moderate 0.74848 0.90652 1 1 1 0.5
29.62 73.06 29.07 29.81 211 moderate 0.74874 0.90701 1 1 1 0.5
29.6 73.05 29.07 29.81 211 moderate 0.74823 0.90689 1 1 1 0.5
29.61 73.02 29.07 29.81 211 moderate 0.74848 0.90652 1 1 1 0.5
29.58 73.1 29.07 29.81 211 moderate 0.74772 0.90751 1 1 1 0.5
29.58 73.06 29.07 29.81 211 moderate 0.74772 0.90701 1 1 1 0.5
29.58 73.02 29.07 29.81 211 moderate 0.74772 0.90652 1 1 1 0.5
29.58 73.09 29.07 29.81 211 moderate 0.74772 0.90739 1 1 1 0.5
29.56 73.62 29.07 29.81 211 moderate 0.74722 0.91397 1 1 1 0.5
29.57 73.05 29.07 29.81 211 moderate 0.74747 0.90689 1 1 1 0.5
29.56 73.03 29.07 29.81 211 moderate 0.74722 0.90664 1 1 1 0.5
29.55 73.63 29.07 29.81 211 moderate 0.74697 0.91409 1 1 1 0.5
29.56 73.12 29.07 29.81 211 moderate 0.74722 0.90776 1 1 1 0.5
29.55 73.15 29.07 29.81 211 moderate 0.74697 0.90813 1 1 1 0.5
29.55 73.52 29.07 29.81 211 moderate 0.74697 0.91273 1 1 1 0.5
29.53 73.46 29.07 29.81 211 moderate 0.74646 0.91198 1 1 1 0.5
29.53 73.52 29.07 29.81 211 moderate 0.74646 0.91273 1 1 1 0.5
29.51 71.21 29.07 29.81 211 moderate 0.74596 0.88405 1 1 1 0.5
29.5 72.21 29.07 29.81 211 moderate 0.7457 0.89646 1 1 1 0.5
29.5 73.25 29.07 29.81 211 moderate 0.7457 0.90937 1 1 1 0.5
29.51 74.1 29.07 29.81 211 moderate 0.74596 0.91993 1 1 1 0.5
29.5 72.36 29.07 29.81 211 moderate 0.7457 0.89832 1 1 1 0.5
29.5 74.12 29.07 29.81 211 moderate 0.7457 0.92017 1 1 1 0.5
29.51 71.36 29.07 29.81 211 moderate 0.74596 0.88591 1 1 1 0.5
29.49 74.36 29.07 29.81 211 moderate 0.74545 0.92315 1 1 1 0.5
29.48 71.23 29.07 29.81 211 moderate 0.7452 0.8843 1 1 1 0.5
29.47 74.36 29.07 29.81 211 moderate 0.74494 0.92315 1 1 1 0.5
29.49 75.32 29.07 29.81 211 moderate 0.74545 0.93507 1 1 1 0.5
29.49 74.12 29.07 29.81 211 moderate 0.74545 0.92017 1 1 1 0.5
29.49 73.52 29.07 29.81 211 moderate 0.74545 0.91273 1 1 1 0.5
29.48 73.23 29.07 29.81 211 moderate 0.7452 0.90912 1 1 1 0.5
Sheet1
Ready
Average: 0.569444444 Count: 90 Sum: 51.25
85%


train - Excel
File Home Insert Page Layout Formulas Data Review View XLSTAT Tell me what you want to do...
nikhil nickky Share
PRODUCT NOTICE Excel hasn't been activated. To keep using Excel without interruption, activate before Wednesday, October 2, 2019. Activate
N14
Normal Bad Good
Neutral Calculation Check Cell
Cut
Copy
Paste Options:
Paste Special...
Smart Lookup
Insert...
Delete...
Clear Contents
Quick Analysis
Filter
Sort
Insert Comment
Format Cells...
Pick From Drop-down List...
Define Name...
Hyperlink...
train
Ready

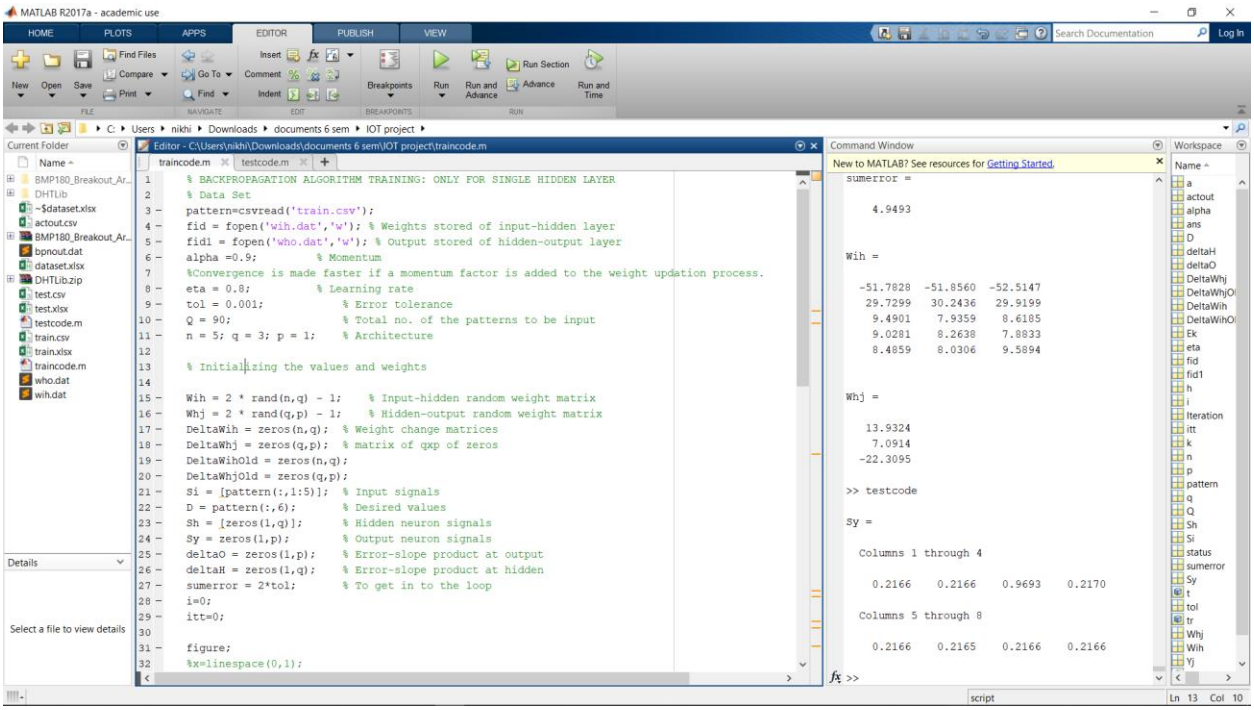

test - Excel
File Home Insert Page Layout Formulas Data Review View XLSTAT Tell me what you want to do...
nikhil nickky Share
PRODUCT NOTICE Excel hasn't been activated. To keep using Excel without interruption, activate before Wednesday, October 2, 2019. Activate
E10
Normal Bad Good
Neutral Calculation Check Cell
test
Ready

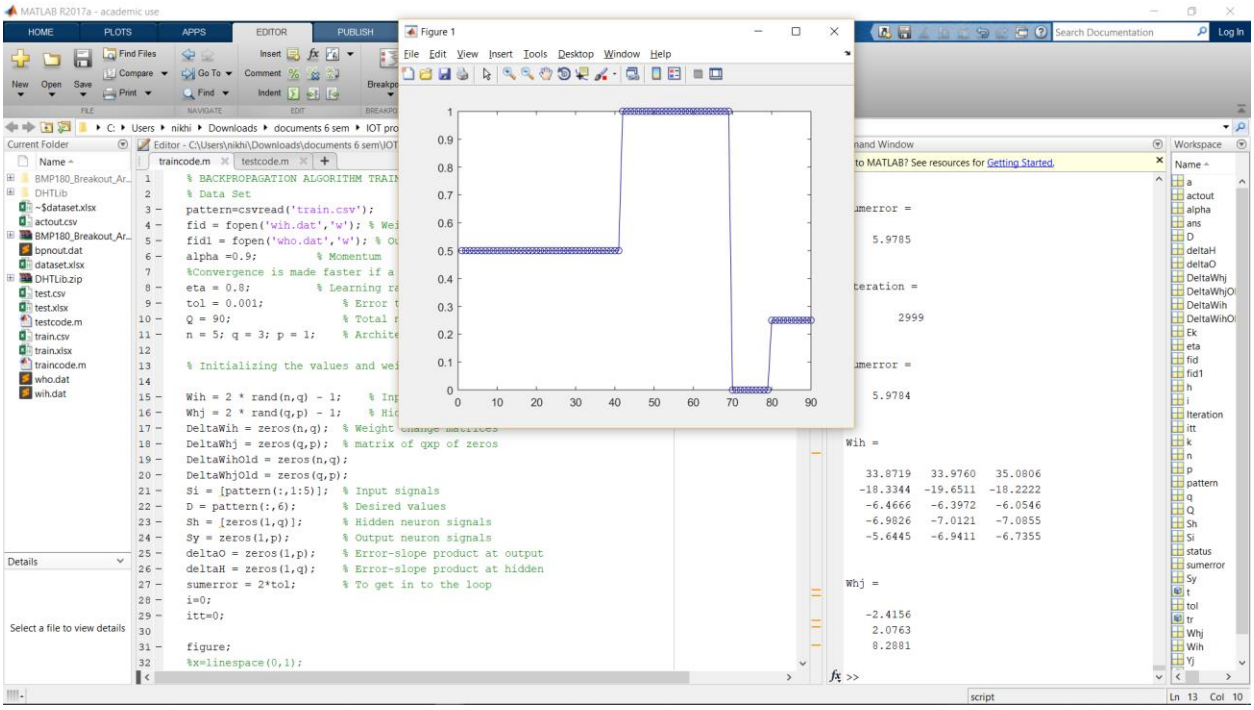
Train Code:

```
% BACKPROPAGATION ALGORITHM TRAINING: ONLY FOR SINGLE
HIDDEN LAYER
% Data Set
pattern=csvread('train.csv');
fid = fopen('wih.dat','w'); % Weights stored of input-
hidden layer
fid1 = fopen('who.dat','w'); % Output stored of hidden-
output layer
alpha =0.9;          % Momentum
%Convergence is made faster if a momentum factor is added
to the weight updation process.
eta = 0.8;           % Learning rate
tol = 0.001;             % Error tolerance
Q = 90;                  % Total no. of the patterns to be
input
n = 5; q = 3; p = 1;     % Architecture

% Initializing the values and weights

Wih = 2 * rand(n,q) - 1;     % Input-hidden random weight
matrix
Whj = 2 * rand(q,p) - 1;     % Hidden-output random weight
matrix
DeltaWih = zeros(n,q);  % Weight change matrices
DeltaWhj = zeros(q,p);  % matrix of qxp of zeros
DeltaWihOld = zeros(n,q);
DeltaWhjOld = zeros(q,p);
Si = [pattern(:,1:5)];  % Input signals
D = pattern(:,6);       % Desired values
Sh = [zeros(1,q)];      % Hidden neuron signals
Sy = zeros(1,p);        % Output neuron signals
deltaO = zeros(1,p);    % Error-slope product at output
deltaH = zeros(1,q);    % Error-slope product at hidden
sumerror = 2*tol;       % To get in to the loop
i=0;
itt=0;

figure;
%x=linespace(0,1);
tr=plot(D);
tr.Color=[0 0 0.5];
tr.Marker='o';
```

```
%hold on;

% Training BPA network

while sumerror>tol && itt<3000  % Iterate(Stops when error
tolerance = 0.001 or when iteration reaches 20,000
   sumerror = 0;
   for k = 1:Q % for loop of input data (Q=99 times)
      Zh = Si(k,:) * Wih;       % Hidden activations
      Sh = [1./(1 + exp(-Zh))]; % Binary sigmoid function
Hidden signals
      Yj = Sh * Whj;            % Output activations
      Sy = 1./(1 + exp(-Yj));   % Binary sigmoid function
Output signals
      Ek = D(k) - Sy;           % Error vector
      deltaO = Ek .* Sy .* (1 - Sy);% Delta output
      for h = 1:q                              % Delta W:
hidden-output
         DeltaWhj(h,:) = deltaO * Sh(h);
      end
      for h = 2:q                              % Delta hidden
         deltaH(h) = (deltaO * Whj(h,:)') * Sh(h) * (1 -
Sh(h));
      end
      for i = 1:n                         % Delta W: input-
hidden
         DeltaWih(i,:) = deltaH(q:q) * Si(k,i);
      end
      Wih = Wih + eta * DeltaWih + alpha * DeltaWihOld;
      Whj = Whj + eta * DeltaWhj + alpha * DeltaWhjOld;
      DeltaWihOld = DeltaWih;                % Update
weights(or)Store changes
      DeltaWhjOld = DeltaWhj;
      sumerror = sumerror + sum(Ek.^2); % Compute error

   end

Iteration = itt
   sumerror % Print epoch error
   itt=itt+1;
end

Wih
Whj
```

```
fprintf(fid,'%12.8f\n',Wih);
fprintf(fid1,'%12.8f\n',Whj);
status = fclose(fid);
status = fclose(fid1);
```

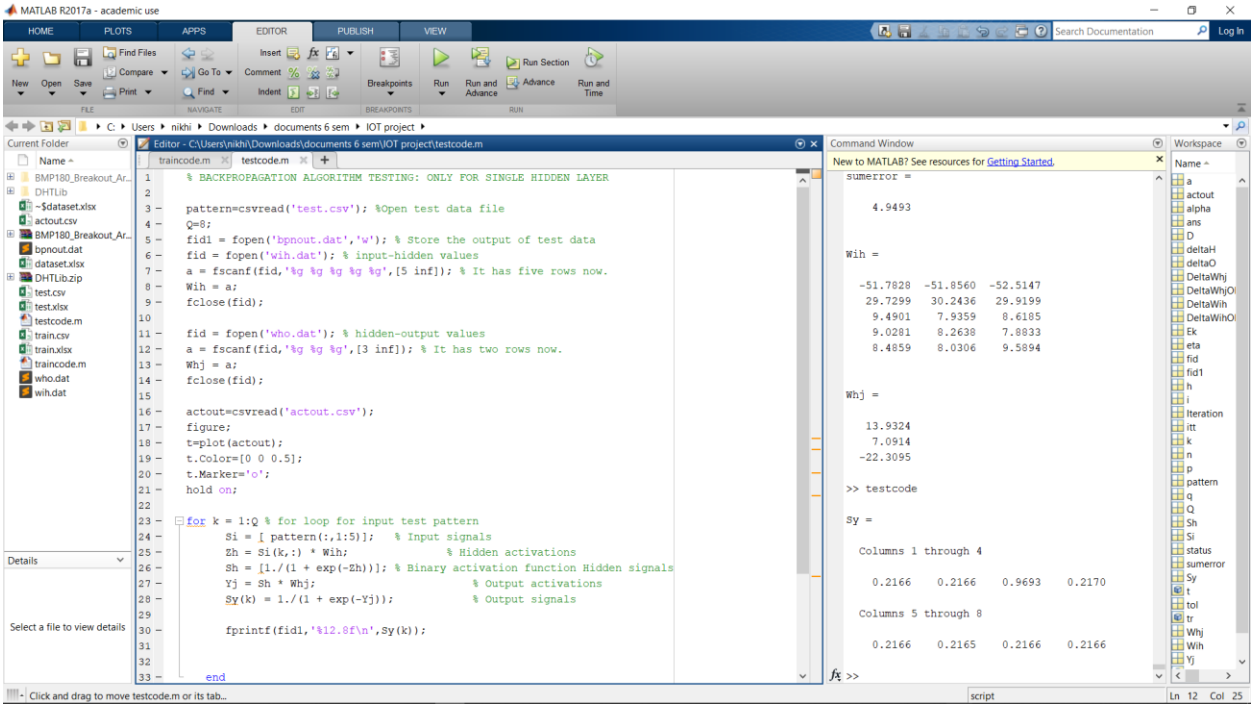


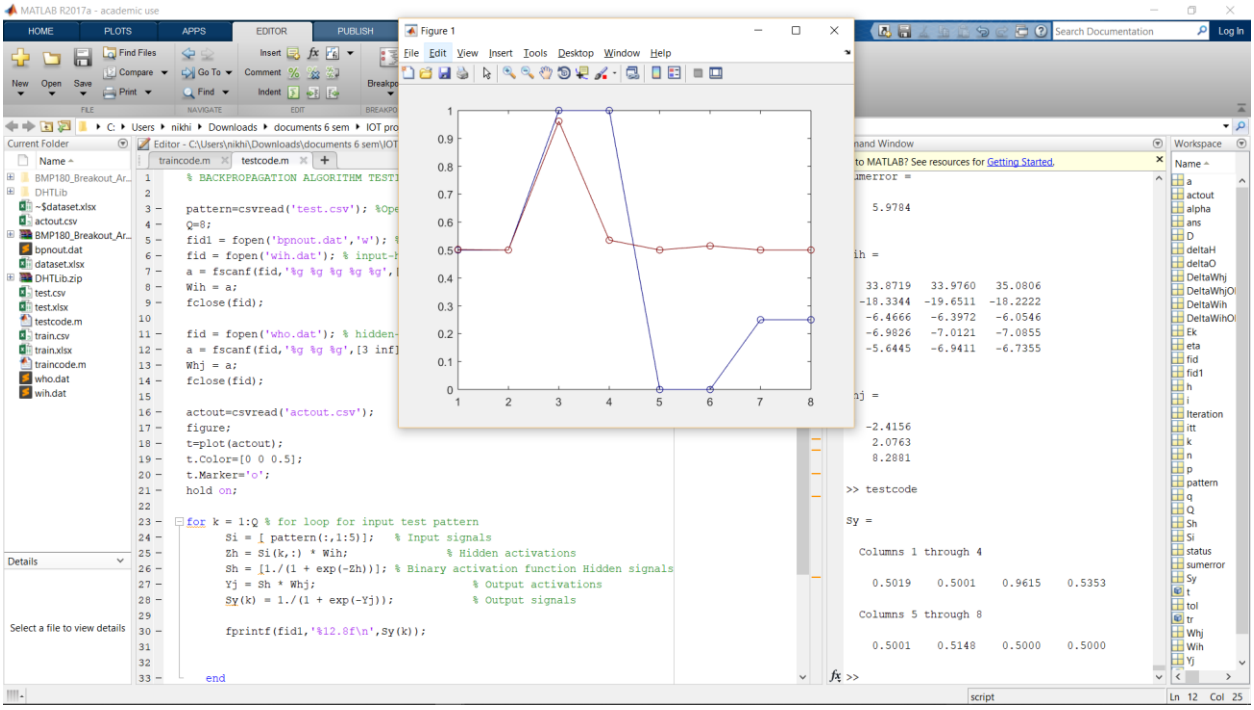


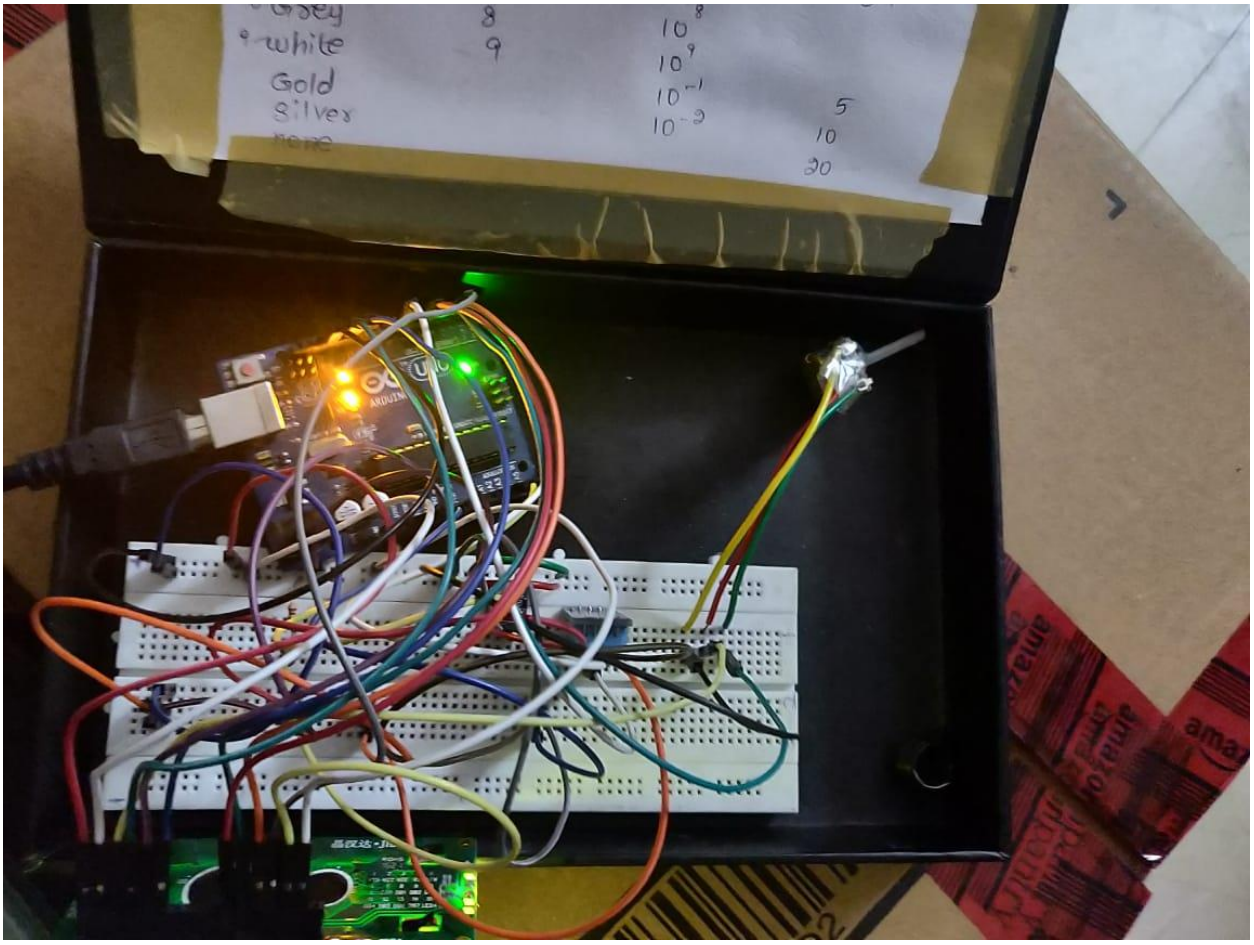
Test Code:

```
% BACKPROPAGATION ALGORITHM TESTING: ONLY FOR SINGLE HIDDEN
LAYER

pattern=csvread('test.csv'); %Open test data file
Q=8;
fid1 = fopen('bpnout.dat','w'); % Store the output of test
data
fid = fopen('wih.dat'); % input-hidden values
a = fscanf(fid,'%g %g %g %g %g',[5 inf]); % It has five
rows now.
Wih = a;
fclose(fid);

fid = fopen('who.dat'); % hidden-output values
a = fscanf(fid,'%g %g %g',[3 inf]); % It has two rows now.
Whj = a;
fclose(fid);

actout=csvread('actout.csv');
figure;
t=plot(actout);
t.Color=[0 0 0.5];
t.Marker='o';
hold on;

for k = 1:Q % for loop for input test pattern
      Si = [ pattern(:,1:5)];   % Input signals
      Zh = Si(k,:) * Wih;                % Hidden
activations
      Sh = [1./(1 + exp(-Zh))]; % Binary activation
function Hidden signals
      Yj = Sh * Whj;                        % Output
activations
      Sy(k) = 1./(1 + exp(-Yj));            % Output
signals

      fprintf(fid1,'%12.8f\n',Sy(k));


   end
  % status = fclose(fid);
```

```
Sy
z=plot(Sy);
z.Color=[0.5 0 0];
z.Marker='o';
hold off;
fclose(fid1);
```




MATLAB R2017a - academic use
Figure 1
File Edit View Insert Tools Desktop Window Help
% BACKPROPAGATION ALGORITHM TEST
pattern=csvread('test.csv'); %Ope
Q=8;
fid1 = fopen('bpnout.dat','w'); %
fid = fopen('wih.dat'); % input-h
a = fscanf(fid,'%g %g %g %g %g',[
Wih = a;
fclose(fid);
fid = fopen('who.dat'); % hidden-
a = fscanf(fid,'%g %g %g',[3 inf]
Whj = a;
fclose(fid);
actout=csvread('actout.csv');
figure;
t=plot(actout);
t.Color=[0 0 0.5];
t.Marker='o';
hold on;
for k = 1:Q % for loop for input test pattern
Si = [pattern(:,1:5)]; % Input signals
Zh = Si(k,:) * Wih; % Hidden activations
Sh = [1./(1 + exp(-Zh))]; % Binary activation function Hidden signals
Yj = Sh * Whj; % Output activations
Sy(k) = 1./(1 + exp(-Yj)); % Output signals
fprintf(fid1,'%12.8f\n',Sy(k));
end
5.9784
33.8719 33.9760 35.0806
-18.3344 -19.6511 -18.2222
-6.4666 -6.3972 -6.0546
-6.9826 -7.0121 -7.0855
-5.6445 -6.9411 -6.7355
-2.4156
2.0763
8.2881
>> testcode
Sy =
Columns 1 through 4
0.5019 0.5001 0.9615 0.5353
Columns 5 through 8
0.5001 0.5148 0.5000 0.5000
>>


White
Gold
Silver
None


Abs Pre: 986.13