A PROJECT REPORT

ON

AIR MONITORING SYSTEM USING IOT

Submitted to

KIIT Deemed to be University

In Partial Fulfillment of the Requirement for the Award

BACHELOR'S DEGREE IN COMPUTER SCIENCE

& ENGINEERING

By

Konderu Hrishikesh-1705041

Shubham Soni-1705073

Sachin Danuka-1705065

Harsh Bajpai-1705038

UNDER THE GUIDANCE OF

PROF SUJATA SWAIN

SCHOOL OF COMPUTER ENGINEERING

KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY

BHUBANESWAR,ODISHA-751024

KIIT Deemed to be University

School of computer Engineering

Bhubaneswar,Odisha-751024

CERTIFICATE

This is to certify that the project entitled

"AIR MONITORING SYSTEM USING IOT"

Submitted by:

Konderu Hrishikesh
Shubham Soni
Sachin Dhanuka
Harsh Bajpai

is a record of Bonafide work carried out by them, in the partial fulfillment of the requirement for the award of Degree of Bachelor of Engineering (Computer Science & Engineering) at KIIT Deemed to be university, Bhubaneswar. This work is done during year 2020-2021, under our guidance.

Date: 20/ 04 / 2020

Prof.SUJATA SWAIN
Project Guide

ACKNOWLEDGEMENT

We are profoundly grateful to Prof.Sujata Swain mam for her expert guidance and encouragement throughout to see that this project rights its target since its commencement to its completion. The work is a team effort minus which the completion of this project was not possible.

Submitted by:

ABSTRACT

Air pollution is a growing issue these days. It is necessary to monitor air quality and keep it under control for a better future and healthy living for all. Here we propose an air quality monitoring system that allows us to monitor and check live air quality on our smart phones as well as the cloud platforms such as Things speak . System uses air sensors to sense presence of harmful gases/compounds in the air and constantly transmit this data to Cloud using tcp connection. Also system keeps measuring temperature and humidity and reports it to the online server over IOT. The sensors interact with micro controller which processes this data and transmits it over internet. This allows authorities to monitor air pollution in different areas and take action against it. Also authorities can keep a watch on the real time data on the cloud platforms and if system detects bad air quality and high temperature it alerts authorities so they can take measures to control the issue and to take preventive measures to protect the environment.

CONTENTS

Topic

1.INTRODUCTION

1.1 Motivation

Our main motivation is to keep track of the air quality ,temperature and humidity of a particular place using Iot sensors and having total data of the air quality of that region in the cloud and also displaying the time to time air quality on the smartphones as well as storing the data on the cloud and displaying the data on the cloud platform.

1.2 Problem Definition

It is very important to maintain efficient equipment to handle information of any project. This project Provides a way to record the air quality information temperature and the humidity and displays them on the smart phone and the things speak cloud.

2.OBJECTIVE

Our main objective of this project is to retrieve the effective values of the air quality, temperature and humidity from the sensors and sending those values to the Things Speak cloud using tcp connection and also to the Blynk Application to View the data values on the mobile phones

3.ANALYSIS

3.1 Introduction

In this phase the requirements are gathered and analyzed. Users requirements are gathered in this phase. This phase is the mainly deals with How to use the system? What data should be input into the system? What data should be output by the system? These are general questions that get answered during a requirements gathering phase. After requirement gathering these requirements are analyzed for their validity and the possibility of incorporating the requirements in the system to be development is also studied.

Finally, a Requirement Specification document is created which serves the purpose of guideline for the next phase of the model.

3.2 Software Requirement Specification

The software requirements specification specifies the functional requirements and nonfunctional requirements. Functional requirements refers to how the system is going to react according to the input provided and how it is going to behave in particular situations and nonfunctional requirements refers to Usability, Reliability, Availability, Performance, Supportability , Interface.

3.3 User Interfaces

We have used the Blynk Application and the Things Speak platform for the user interface .

3.4 EXTERNAL INTERFACE REQUIREMENTS

3.4.1 Software Requirements

- **Arduino IDE**
- **Things Speak**
- **Blynk Application**

3.4.2 Hardware Requirements

- **Arduino Uno**
- **PM 2.5 sensor(SDS011).**
- **MQ135 sensor(Gas sensor)**
- **DHT 11 sensor**
- **Bread Board**
- **Jumper Wires**
- **ESP 8266 Wifi module**
- **Bluetooth Sensor.**
- **12 volts power supply.**

About Sensors used

PM 2.5 sensor(SDS011)

Particle pollution, also called particulate matter (or PM), is a mixture of solid particles and liquid droplets floating in the air. Some particles are released directly from a specific source, while others are formed in complicated chemical reactions in the atmosphere. It uses the principle of laser scattering in the air, and can detect suspended particulate matter concentration ranging from 0.3 to 10 microns. Data

collected by the sensor is stable and reliable. SDS011 sensor is connected to UART port (TX and RX) of Arduino Uno board.

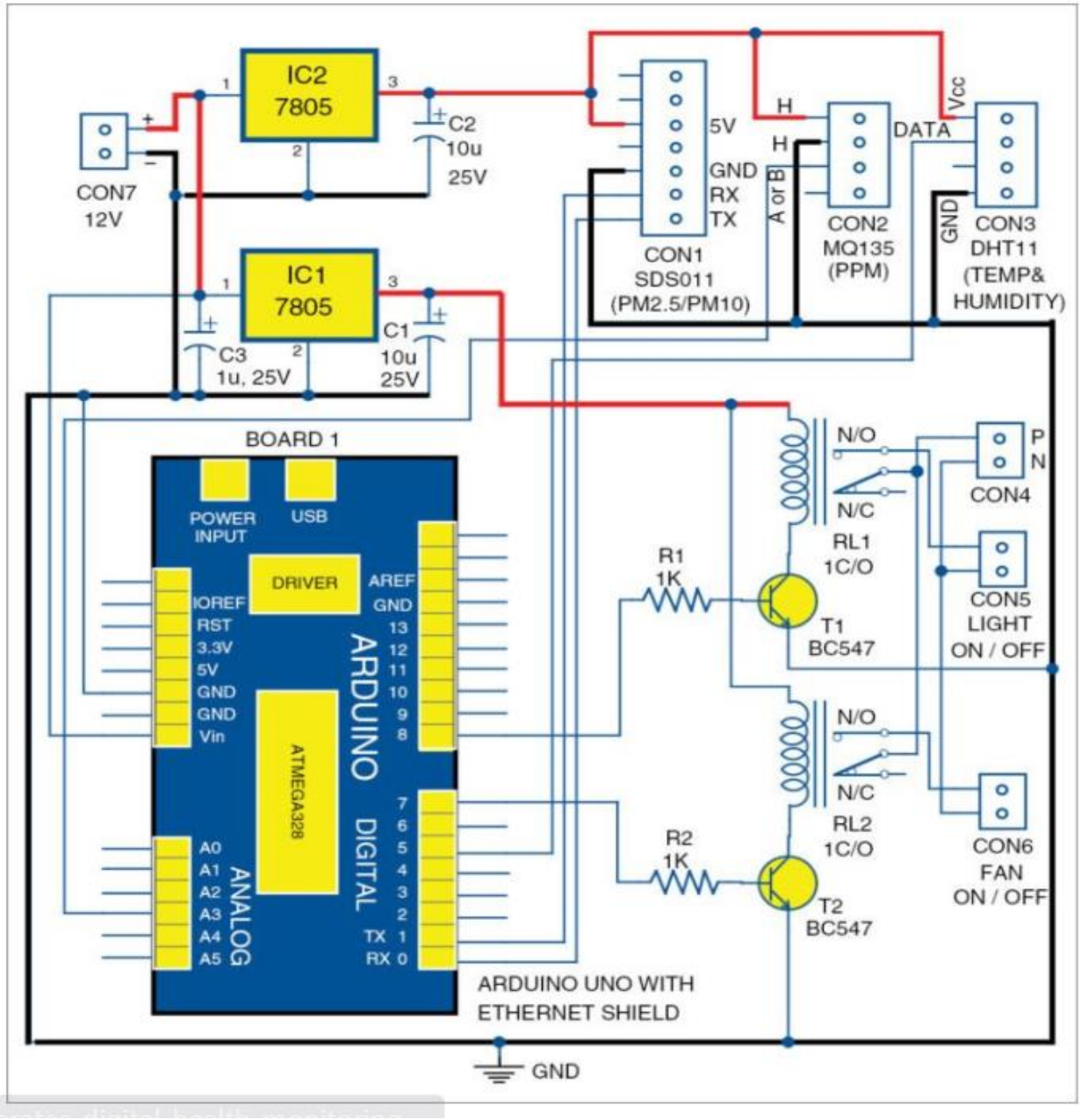
MQ135 sensor

Sensitive material of the sensor is tin-dioxide, whose conductivity increases with the concentration of gas. Change in conductivity is converted into output voltage signal, which varies corresponding to the concentration of combustible gas. MQ135 is highly sensitive to ammonia, sulphide and benzene steams, smoke and other harmful gases. It is a low-cost sensor, suitable for different applications. Output of the gas sensor is connected to analogue input pin A3 of Arduino Uno board through connector CON2.

Temperature and Humidity Sensor(DHT11)

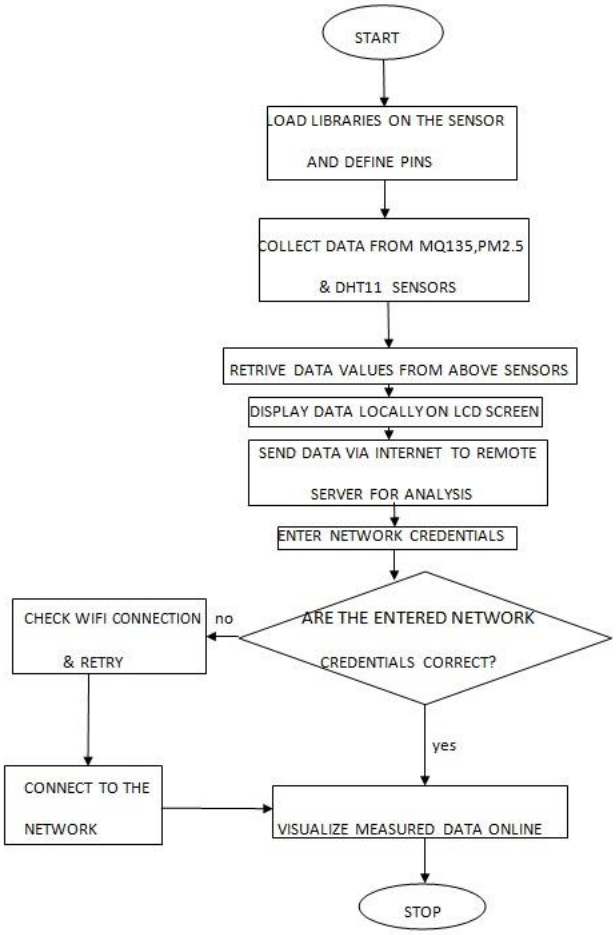
This composite sensor contains calibrated digital signal outputs of temperature and humidity. Connected to connector CON3, it includes a resistive-type humidity measurement component and an NTC temperature-measurement device. Its output pin is connected to digital pin 5 of Arduino Uno board. It is a relatively inexpensive sensor for its performance.

ESP8266 WIFI Module

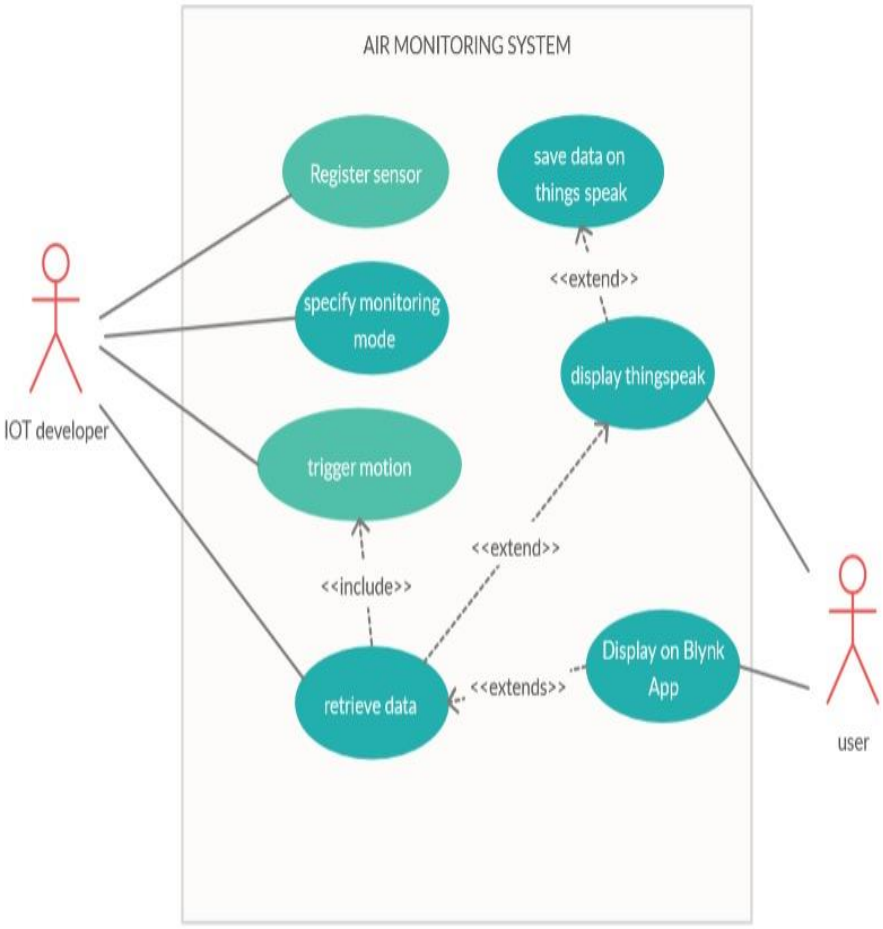


ImageReference:https://www.electronicsforu.com/electronics-projects/iot-enabled-air-pollution-meter

DATA FLOW DIAGRAM.



USE CASE DIAGRAM



USE CASE DIAGRAM

5.IMPLEMENTATION

5.1 Introduction

Implementation is the most crucial stage in achieving a successful system and giving the user's confidence that the new system is workable and effective. Implementation of the modified application to replace an existing one. This type of conversation is relatively easy to handle, provide there are no major changes in the system.

5.2 Code

Use the Arduino IDE to write the code for the given project.


```
WidgetLED led2(V2);
WidgetLED led4(V4);
SimpleTimer timer;
WidgetRTC rtc;
BLYNK_ATTACH_WIDGET(rtc, V5);
void setup()
{
  Serial.begin(9600);
//  pinMode(LEDPIN, OUTPUT);
    SEC1_flag = 0;
    led_cnt = 0;
    // initialize Timer1
    cli();          // disable global interrupts
    TCCR1A = 0;     // set entire TCCR1A register to 0
    TCCR1B = 0;     // same for TCCR1B

    // set compare match register to desired timer count:
    OCR1A = 15624;
    // turn on CTC mode:
    TCCR1B |= (1 << WGM12);
    // Set CS10 and CS12 bits for 1024 prescaler:
    TCCR1B |= (1 << CS10);
    TCCR1B |= (1 << CS12);
    // enable timer compare interrupt:
    TIMSK1 |= (1 << OCIE1A);
    // enable global interrupts:
    sei();
```

pollution.ino | Arduino 1.8.11

File Edit Sketch Tools Help

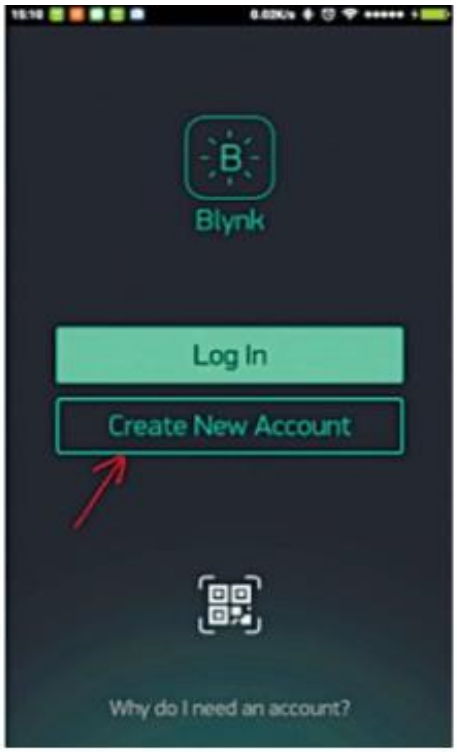
pollution.ino §

```
void loop()
{
  Blynk.run();
  timer.run();
  ProcessSerialData();

  int chk = DHT.read11(DHT11_PIN);

  //Serial.print(DHT.humidity, 1);
  //Serial.print(",\t");
  //Serial.println(DHT.temperature, 1);

  //ppm calibration
  count=analogRead(GasSensorPin);
  Vout=(count*4.88)/1000.0;
  Rs=((5.0*Rl)-(Rl*Vout))/Vout;
  ratio=Rs/approx;
  GasConc=146.15*(2.868-ratio)+10;
  ppm=GasConc;


  if(SEC1_flag == 1)
  {
  SEC1_flag = 0;
  led_cnt++;
  switch(led_cnt)
```

5.3 Blynk App Implementation.

Step 1: Connect your mobile with Wi-Fi. Download and install Blynk app from Google Play store. Then, create a new Blynk account . This account is separate from the accounts used for Blynk Forums, in case you already have one.

An account is needed to save your projects and have access to these from multiple devices anywhere in the world. It is also a security measure. We recommend using a real email address because it will simplify things later.



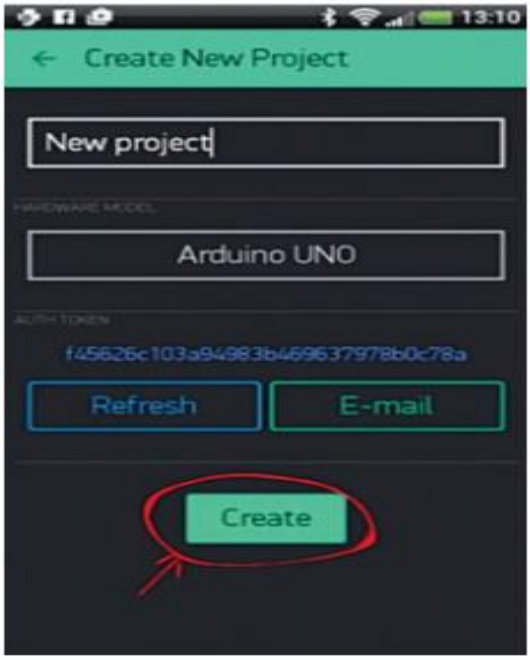
Step 2: After you have successfully logged into your account, start by creating a new project , and give it a name

Step 3 : Select the hardware model you intend to use. In this case it is Arduino Uno

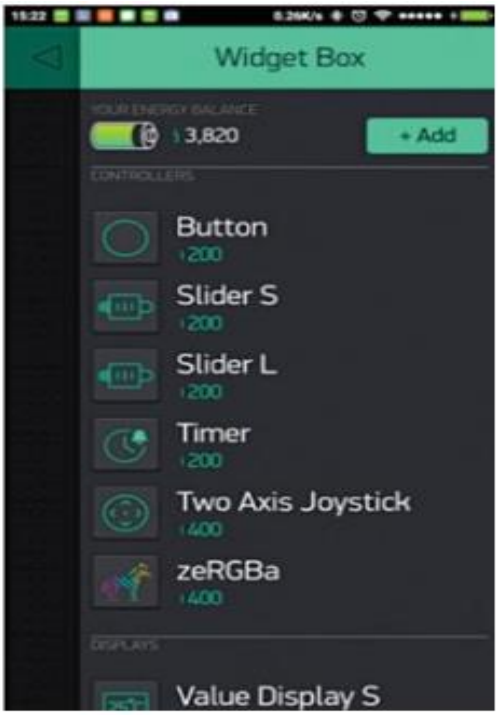
Step 4: Authorisation (or Auth) token is a unique identifier that is needed to connect the hardware to your smartphone. Every new project you create will have its own Auth token . Click email button and the token will be sent to the email address you used for registration. Use this token in auth[] = "your token" in pollution.ino file.

.

Step 5: Create the new project and



Step 6. Tap anywhere on the canvas to open the widget box; all available widgets are located here. Add widgets to display the data.

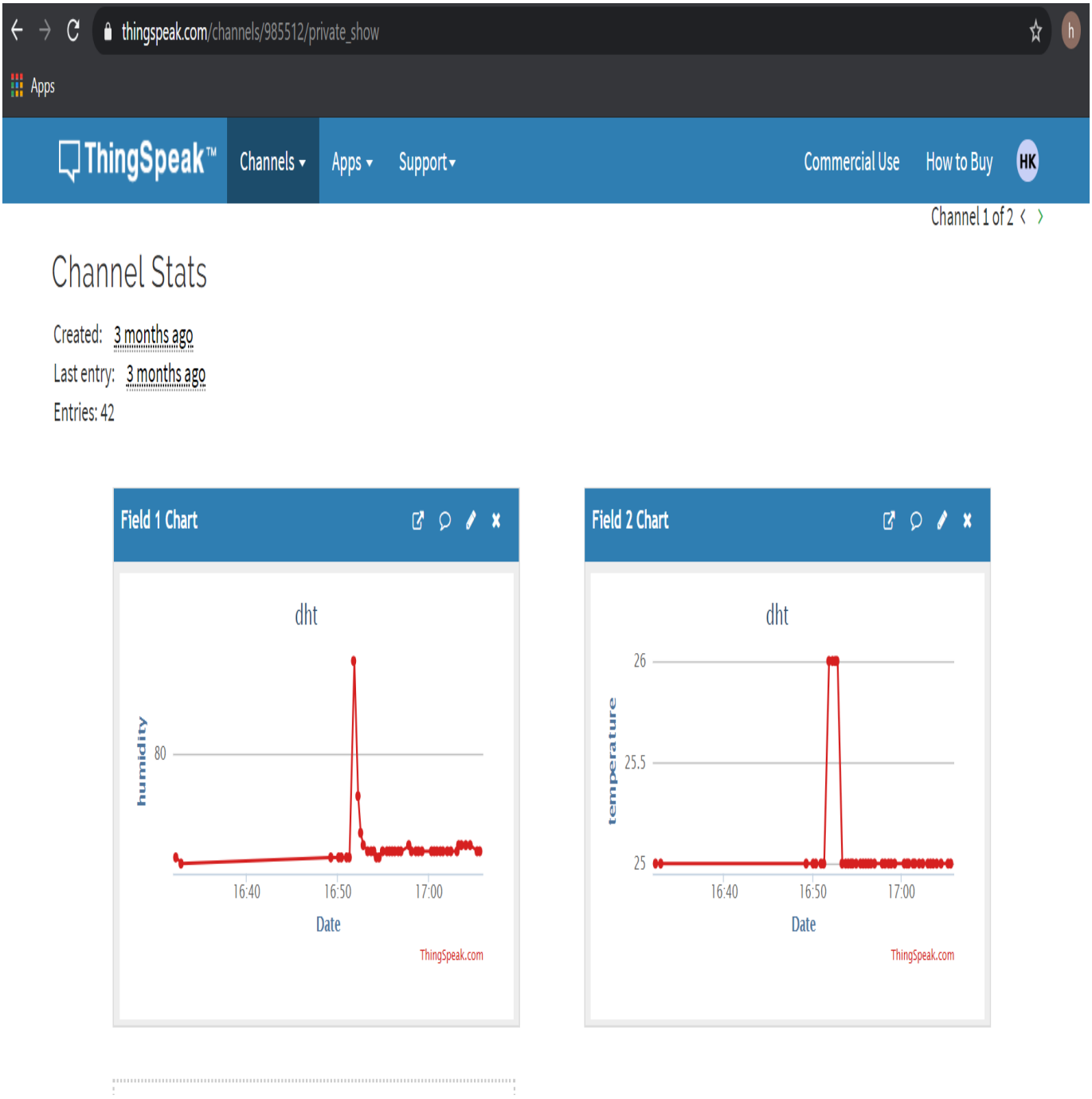


6.RESULTS

Data values will be displayed on the blynk app and also on the things speak.


Indoor Air Quality
HUMIDITY
17.383
100
75
50
25
0
4:03:51 pm
TEMP
30.319
40
30
20
10
0
4:03:51 pm
CO2
460
0
2000
TVOC
128
0
600

THINGS SPEAK DISPLAY



Advantages of The Project.

1.The data collected from air quality monitoring helps us assess impacts caused by poor air quality on public health.

2.Air quality data helps us determine if an area is meeting the air quality standards devised by CPCB, WHO or OSHA.

3.The data collected from air quality monitoring would primarily help us identify polluted areas, the level of pollution and air quality level.

4.Air quality monitoring would assist in determining if air pollution control programmes devised in a locality are working efficiently or not.

5.Air quality data helps us understand the mortality rate of any location due to air pollution. We can also assess and compare the short term and long term diseases/disorders which are a result of air pollution.

6.Based upon the data collected control measures can be devised for protection of environment and health of all living organisms.

Disadvantages

1.Outdated hardware and software.

2.Use of weak and default credentials.

3.Malware and ransomware.

4.Predicting and preventing attacks.

5.Difficult to find if a device is affected.

6.Data protection and security challenges.

7.Cost inefficiency on carriers .

7.Conclusion

so we can estimate the purity of the air using Internet Of Things efficiently and store and display that data.

8. References :: reference1

Submitted by:

Konderu Hrishikesh
Shubham Soni
Sachin Dhanuka
Harsh Bajpai