TEMPERATURE AND HUMIDITY MONITORING AND RAISING Email alert using esp32

A MINI PROJECT REPORT

Submitted in partial fulfilment of requirements for the Award of the degree

BACHELOR OF TECHNOLOGY
IN
ELECTRONICS AND COMMUNICATION ENGINEERING

By

Ms. V. PAVAN MANIKANTA KUMAR	**(20475A0407)**
Mr. N. JAGADEESH	**(20475A0408)**
Mr. P. VENKATA GANESH	**(19471A04G5)**
Mr. SK. DARIYAVALI	**(19471A04H2)**
Mr. S.GOPI	**(19471A04G8)**

Under the guidance of
Mr. N. SRINIVASA RAO
Professor

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
NARASARAOPETA ENGINEERING COLLEGE (Autonomous)
(Approved by AICTE, New Delhi, Accredited thrice by NBA, Accredited by NAAC with 'A' Grade, Permanently Affiliated to J.N.T.U.K, Kakinada
Kotappakonda Road, Yallamanda,
Narasaraopet - 522601, Palnadu(Dist.), Andhra Pradesh.

June, 2022

NARASARAOPETA ENGINEERING COLLEGE (Autonomous)

(Approved by AICTE, New Delhi, Accredited thrice by NBA, Accredited by NAAC with 'A' Grade, Permanently Affiliated to J.N.T.U.K, Kakinada

Kotappakonda Road, Yallamanda,

Narasaraopet - 522601, Palnadu (Dist.), Andhra Pradesh.

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

CERTIFICATE

This is to certify that the mini project entitled TEMPERATURE AND HUMIDITY MONITORING AND RAISING Email and sms alert using esp32 is the Bonafede work carried out by Ms. V .Pavan Manikanta Kumar (20475A0407), Mr. N. Jagadeesh(2047504078), Mr. P. Venkata Ganesh (19471A04G5), Mr. Sk. Dariya vali (19471A04H2) and Mr. S. Gopi (19471A04G8) in partial fulfilment of the requirements for the award of the degree Bachelor of Technology in Electronics and Communication Engineering from Jawaharlal Nehru Technological University Kakinada, Kakinada during the year 2020-21 under my supervision and guidance.

Guide
(Mr. N. SRINIVASARAO)

Head of the Department
(Dr. V. VENKATA RAO)

External Examiner

ACKNOWLEDGEMENTS

It is our pleasure to express our deep and sincere gratitude to our mini project guide **Dr .N.SRINIVASARAO**, Professor, for extending his sincere and heartfelt guidance throughout this project work.

It is our desire to express sincere thanks to our mini project coordinator

Dr. K. Sk. Bajid vali Professor, for extending his cooperation and guidance throughout this project work.

We are thankful to our principal **Dr. M. Sreenivasa Kumar** for providing us sparkling environment during our course of study.

We also express sincere thanks to **Sri M.V. Koteswara Rao**, Chairman of Narasaraopeta Engineering College, Narasaraopet for providing excellent infrastructural facilities to complete our course.

We extend our sincere thanks to all other **Teaching** and **Non-Teaching Faculty members** of the department for their cooperation and encouragement throughout our course of study.

We affectionately acknowledge our **friends** for their motivation and suggestions which helps us in successfully completing our course.

We have no words to acknowledge the warm affection, constant inspiration and encouragement that we received from our **Parents and Family Members.**

Mr. V. Pavan Manikanta Kumar (20475A0407)
Mr. N. Jagadeesh (20475A0408)
Mr. P. Venkata Ganesh (19471A04G5)
Mr. Sk. Dariya vali (19471A04H2)
Mr. S. Gopi (19471A04G8)

INDEX

ABSTRACT

- In this project We are going to build a small weather station by using DHT11 sensor in ESP32 kit. The DHT11 sensor is a basic, ultra-low-cost digital temperature and humidity sensor. It uses a capacitive humidity sensor and a thermistor to measure the surrounding air and spits out a digital signal on the data pin. In This project we are using PYTHON code and the components we are going to use in this project are ESP32 board, DHT11 sensor. The basic Idea of this project is not only to measure the current temperature and humidity but also show the data of maximum and minimum temperature value.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIO

Gnd : Ground

IDE : Integrated Development Environment

PDE : Processing Development Environment

CHAPTER 1

INTRODUCTION

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

- In this project We are going to build a small weather station by using DHT11 sensor in ESP32 kit. The DHT11 sensor is a basic, ultra-low-cost digital temperature and humidity sensor. It uses a capacitive humidity sensor and a thermistor to measure the surrounding air and spits out a digital signal on the data pin. In This project we are using PYTHON code and the components we are going to uses are breadboard,

- ESP32 (Node Mcu), DHT11 sensor. The basic Idea of this project is not only to measure the current temperature and humidity but also show the data of maximum and minimum temperature value.

1.2 OBJECTIVES OF THE PROJECT

- It is possible to setup ESP32's Wi-Fi to access point and station. ESP32 can connect to another hotspot and share the connection. An access point (AP) is a networking hardware device which allows other Wi-Fi devices to connect to a network. Some basic theory required to know for making basic code better.

- Attractive features and capabilities of ESP32 microcontroller, has made it a pretty good option for a variety of IoT –Internet of Things- projects and applications. In order to make the fullest use of the ESP32 abilities and features, a sufficient knowledge of its pins is so essential. The goal of this tutorial is to introduce all types of pins available in this microcontroller and the features associated with each of those pins. So, stay with us until the end of this tutorial!

.

1.3 LITERATURE SURVEY

- We are going to build a small weather station by using DHT11 sensor in ESP32 kit. The DHT11 sensor is a basic, ultra-low-cost digital temperature and humidity sensor. It uses a capacitive humidity sensor and a thermistor to measure the surrounding air and spits out a

digital signal on the data pin. In This project we are using PYTHON code and the components we are going to uses are breadboard.

1.4 ORGANIZATION OF THE REPORT

Chapter 1 describes Introduction about the project. In chapter 2, the Proposed methods of the project are presented. In chapter 3, the Components description and Simulation of the project are presented. In chapter 4, the Results of the project are presented. In chapter 5, Advantages, Limitations and Disadvantages are presented. In chapter 6 Conclusion and Future Scope of the project are presented.

CHAPTER 2

PROPOSED METHOD

CHAPTER 2

PROPOSED METHOD

2.1 Pin Diagram

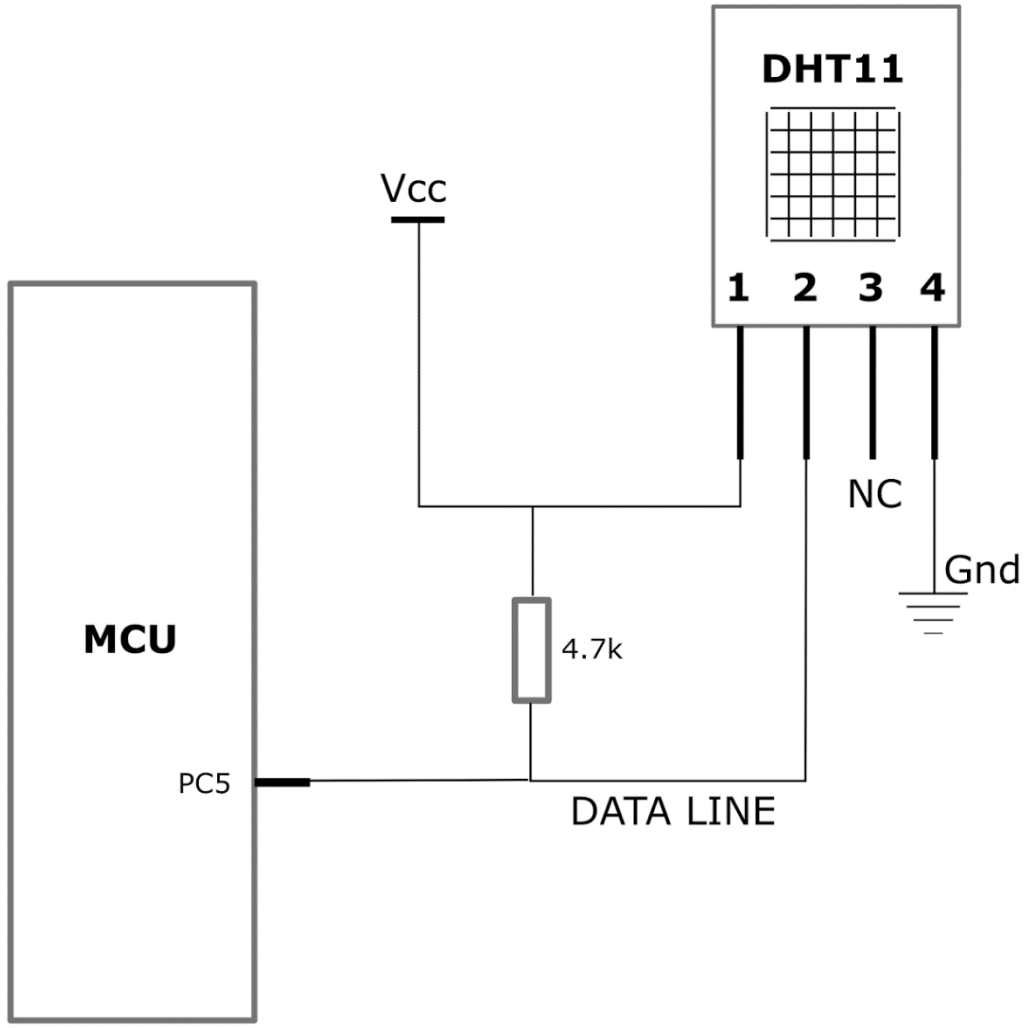


Fig. 2.1: Block diagram

WORKING

As per the above block diagram. DHT11 Sensor is connected to the Esp32 board that it consists of 3 pins as Vcc, Gnd, Data. Vcc will be connected to the Vcc pin of Esp32, Data pin is connected to the pc5 pin of esp32 . Gnd (Ground) pin is connected to the Gnd pin of the esp32.

The result of this project is observed in the system, so that connect your personal computer through USB plug.

Circuit

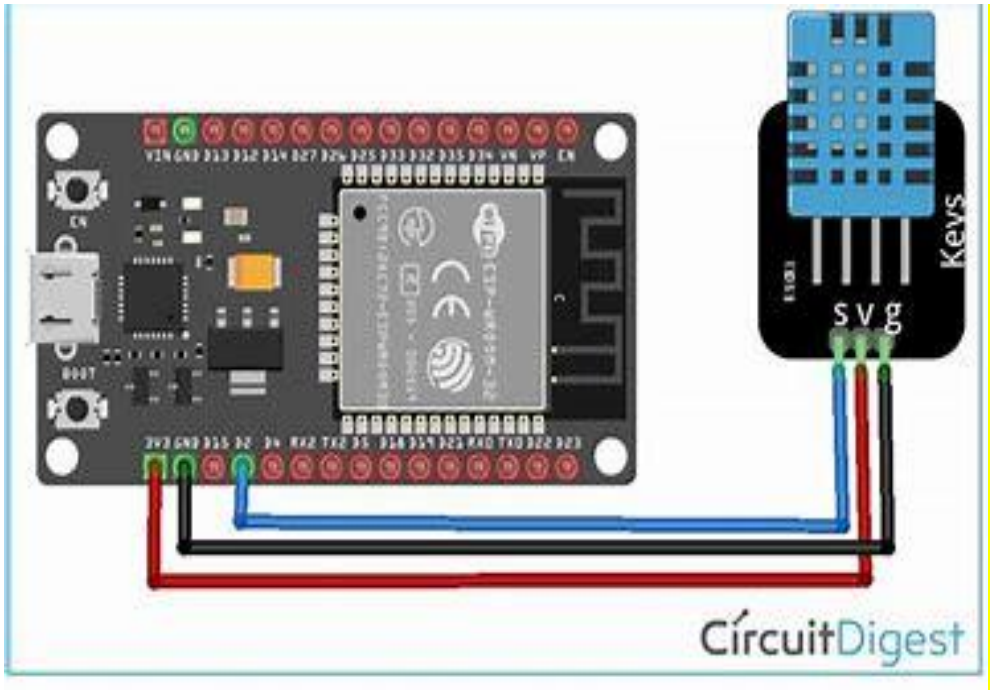


Fig. 2.2: Circuit diagram

2.2 Circuit Connections

servo motor:

black wire	esp32,Gnd
red wire	vcc,3v
blue wire	esp32 data pin D2

DHT11 Sensor:

vcc	Esp32 3v pin
Gnd	Esp32,Gnd
Data	Esp32 pin D2

Esp32:

Esp32

USB cable

Personal computer

2.3 Simulation

Initially upload the code to Arduino after making the connections as shown in the above circuit diagram by using Arduino IDE software. You can observe the Temperature from 0 degrees to 50 degrees.

Now, open the processing application and paste the sketch. In the processing sketch, make necessary changes in the COM port selection and replace it with the COM port number to which your Esp32 is connected to.

- Now, run the sketch in the processing and if everything goes well, a new processing window opens up like the one shown below in the results.
- A serial monitor is represented in a data type display, If the DHT11 sensor dectet Temperature and humidity. the same will be displayed Serial Monitor on the screen.

CHAPTER 3

COMPONENTS DESCRIPTION

CHAPTER 3

COMPONENTS DESCRIPTION

3.1 About Arduino

Arduino is an open-source microcontroller which can be easily programmed, erased and reprogrammed at any instant of time. Introduced in 2005 the Arduino platform was designed to provide an inexpensive and easy way for hobbyists, students and professionals to create devices that interact with their environment using sensors and actuators. Based on simple microcontroller boards, it is an open-source computing platform that is used for constructing and programming electronic devices. It is also capable of acting as a mini computer just like other microcontrollers by taking inputs and controlling the outputs for a variety of electronics devices. It is also capable of receiving and sending information over the internet with the help of various Arduino shields, which are discussed in this paper. Arduino uses a hardware known as the Arduino development board and software for developing the code known as the Arduino IDE (Integrated Development Environment). Built up with the 8-bit Atmel AVR microcontroller's that are manufactured by Atmel or a 32-bit Atmel ARM, these microcontrollers can be programmed easily using the C or C++ language in the Arduino IDE. Unlike the other microcontroller boards in India, the Arduino boards entered the electronic market only a couple of years ago, and were restricted to small scale projects only. People associated with electronics are now gradually coming up and accepting the role of Arduino for their own projects. This development board can also be used to burn (upload) a new code to the board by simply using a USB cable to upload. The Arduino IDE provides a simplified integrated platform which can run on regular personal computers and allows users to write programs for Arduino using C or C++.

3.1.1 ELEMENTS OF ESP32

Elements of an Arduino Board can be done into two categories:

- Hardware
- Software

- **Hardware**

The Arduino Development Board consists of many components that together makes it work. Here are some of those main component blocks that help in its functioning:

• **Microcontroller**: This is the heart of the development board, which works as a mini computer and can receive as well as send information or command to the peripheral devices connected to it. The microcontroller used differs from board to board; it also has its own various specifications.

• **External Power Supply**: This power supply is used to power the Arduino development board with a regulated voltage ranging from 9 – 12 volts.

• **USB plug**: This plug is a very important port in this board. It is used to upload (burn) a program to the microcontroller using a USB cable. It also has a regulated power of 5V which also powers the Arduino board in cases when the External Power Supply is absent.

• **Internal Programmer:** The developed software code can be uploaded to the microcontroller via USB port, without an external programmer.

• **Reset button**: This button is present on the board and can be used to resets the Arduino microcontroller.

• **Analog Pins**: There are some analog input pins ranging from A0 – A7 (typical). These pins are used for the analog input / output. The no. of analog pins also varies from board to board.

• **Digital I/O Pins**: There are some digital input pins also ranging from 2 to 16 (typical).These pins are used for the digital input / output. The no. of these digital pins also varies from board to board.

• **Power and GND Pins:** There are pins on the development board that provide 3.3, 5 volts and ground through them.

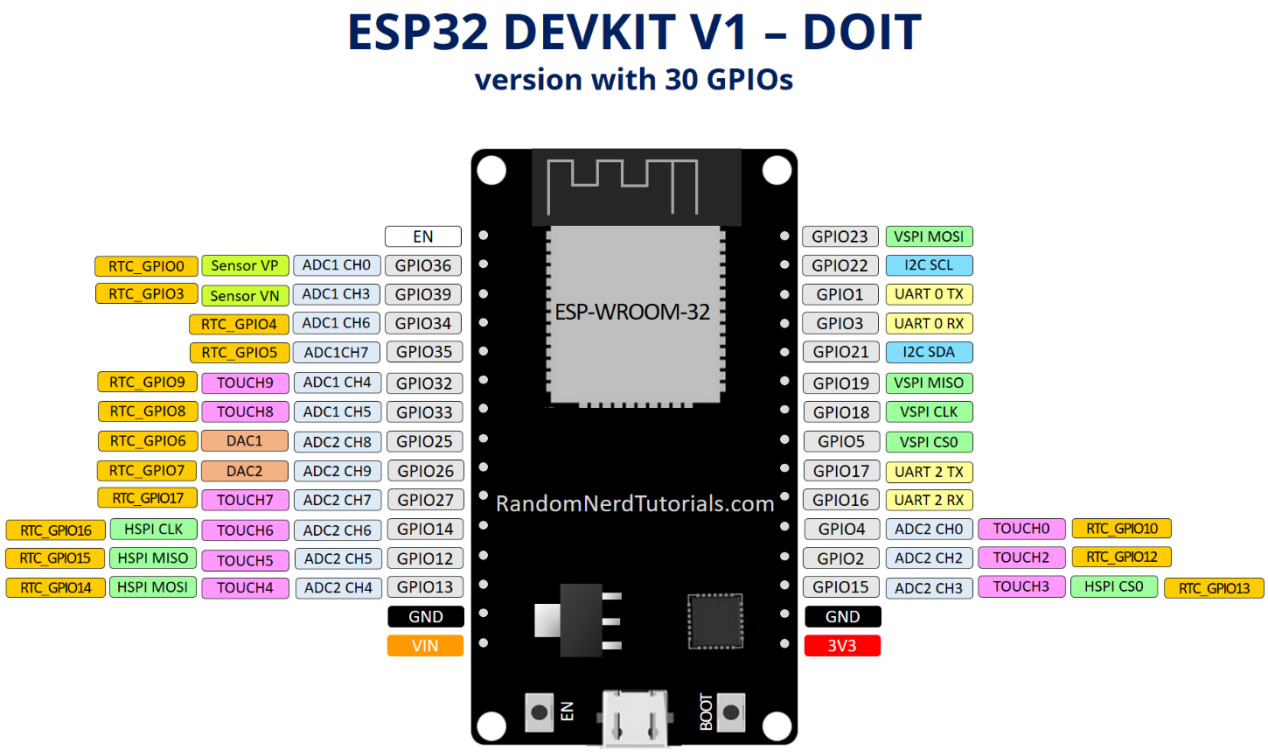
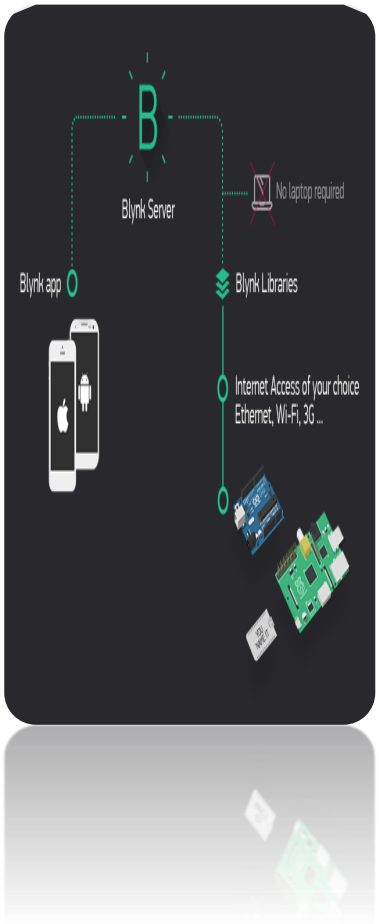


Fig. 3.1: A label diagram of Esp32 Board

- **Software**

- **Blynk**
- **Blynk** is a new platform that allows you to quickly build interfaces for controlling and monitoring your hardware projects from your iOS and Android device. After downloading the Blynk app, you can create a project dashboard and arrange buttons, sliders, graphs, and other widgets onto the screen.



Create a Blynk Account

After you download the Blynk App, you'll need to create a New Blynk account. This account is separate from the accounts used for the Blynk Forums, in case you already have one.

We recommend using a **real** email address because it will simplify things late.

Why do I need to create an account?

An account is needed to save your projects and have access to them from multiple devices from anywhere. It's also a security measure.

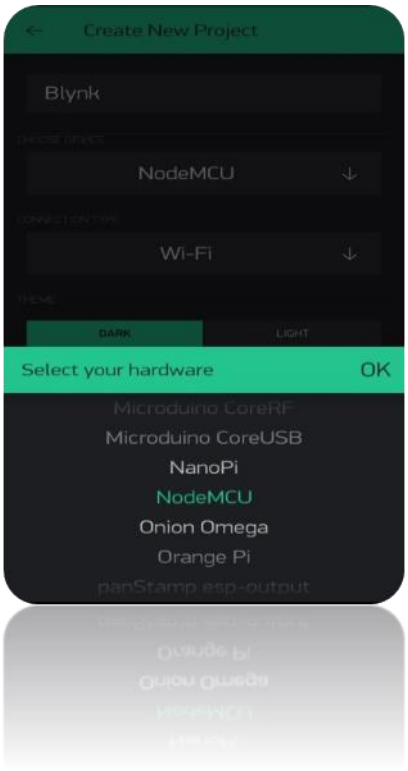
You can always set up your own Private Blynk Server and have full control.

Create a new project

After you've successfully logged into your account, start by creating a new project.

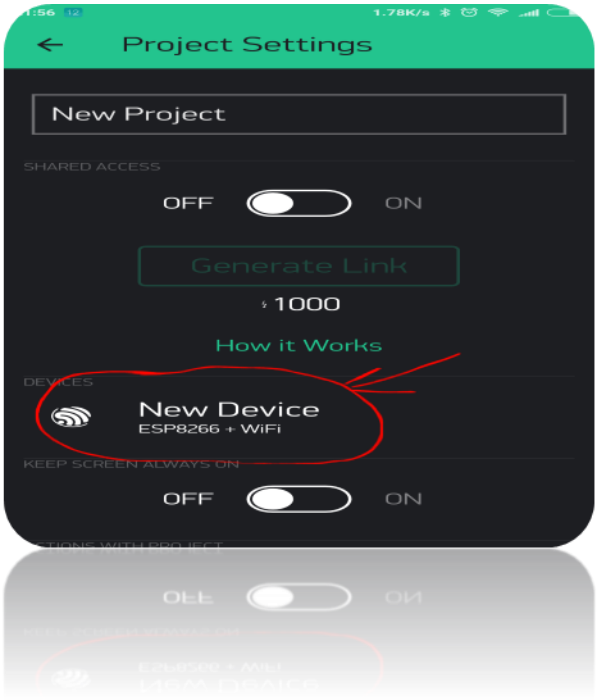
Choose your Hardware

Select the hardware model you will use. Check out the list of supported hardware e

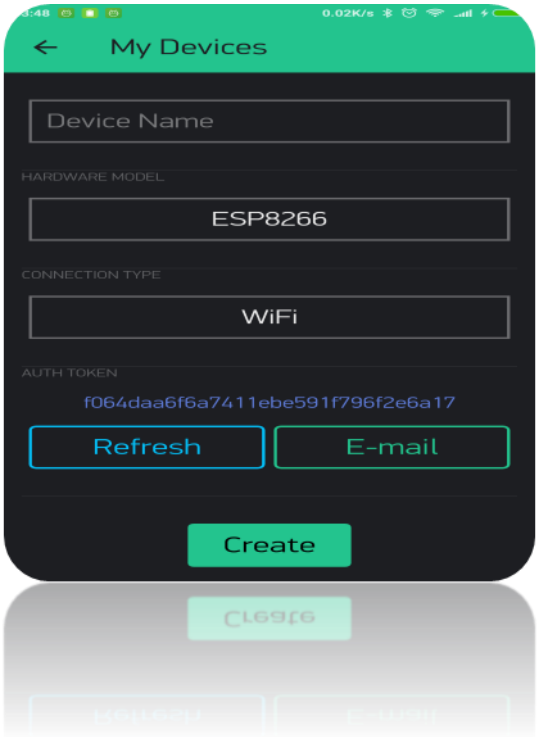


1. Auth Token

Auth Token is a unique identifier which is needed to connect your hardware to your smart phone. Every new project you create will have its own Auth Token. You'll get Auth Token automatically on your email after project creation. You can also copy it manually. Click on devices section and selected required device .



And you'll see token :



NOTE: Don't share your Auth Token with anyone, unless you want someone to have access to your hardware.

It's very convenient to send it over e-mail. Press the e-mail button and the token will be sent to the e-mail address you used for registration. You can also tap on the Token line and it will be copied to the clipboard.

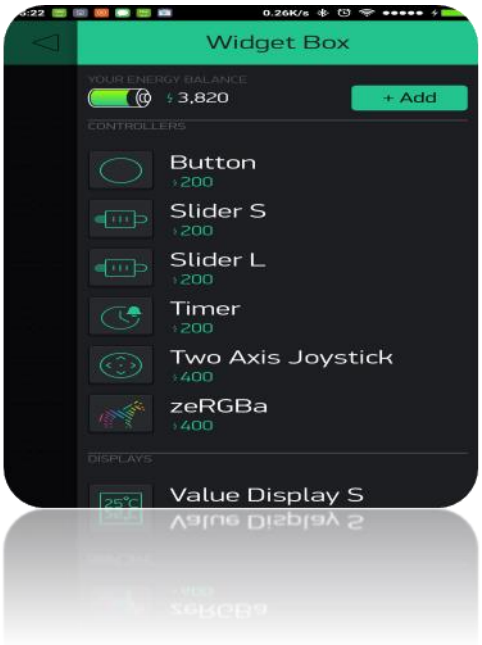
Now press the **"Create"** button.

5.Add a Widget

Your project canvas is empty, let's add a button to control our LED.

Tap anywhere on the canvas to open the widget box. All the available widgets are located here. Now pick a button.

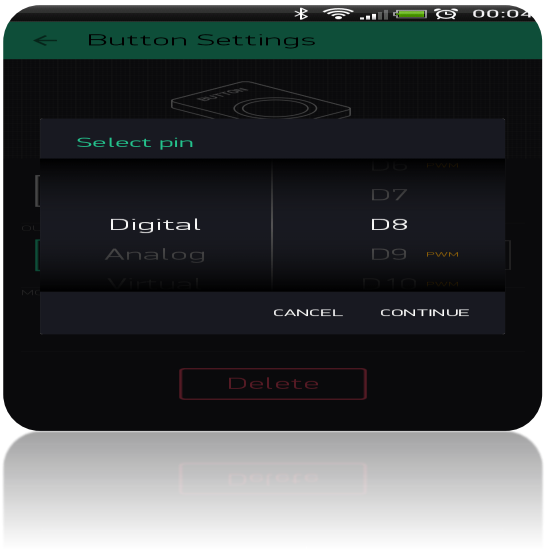
Widget Box



Drag-n-Drop - Tap and hold the Widget to drag it to the new position.

Widget Settings - Each Widget has it's own settings. Tap on the widget to get to them.

The most important parameter to set is **PIN** . The list of pins reflects physical pins defined by your hardware. If your LED is connected to Digital Pin 8 - then select **D8** (**D** - stands for **D**igital).



6. **Run The Project**

   When you are done with the Settings - press the **PLAY** button. This will switch you from EDIT mode to PLAY mode where you can interact with the hardware. While in PLAY mode, you won't be able to drag or set up new widgets, press **STOP** and get back to EDIT mode.

Fig. 3.2: Arduino software(version1.8.1)

The program code written for Arduino is known as a sketch. The software used for developing such sketches for an Arduino is commonly known as the Arduino IDE. This IDE contains the following parts in it:

• **Text editor**: This is where the simplified code can be written using a simplified version of C++ programming language.

• **Message area**: It displays error and also gives a feedback on saving and exporting the code.

• **Text:** The console displays text output by the Arduino environment including complete error messages and other information

• **Console Toolbar**: This toolbar contains various buttons like Verify, Upload, New, Open, Save and Serial Monitor. On the bottom right hand corner of the window there displays the Development Board and the Serial Port in use.

Fig. 3.3: Arduino Software Interface

3.1.2 Features of Arduino IDE

• The project file or the sketches for a project are saved with the file extension.

• Features such as cut / copy / paste are supported in this IDE.

• There also is a facility for finding a particular word and replacing it with another by pressing the Ctrl + F buttons on the keyboard

• The most basic part or the skeleton of all Arduino code will have two function.

3.2 Processing

Fig. 3.4: Processing (programming language)

The Processing Development Environment (PDE) makes it easy to write Processing programs. Programs are written in the Text Editor and started by pressing the Run button. In Processing, a computer program is called a *sketch*. Sketches are stored in the *Sketchbook*, which is a folder on your computer.

Sketches can draw two- and three-dimensional graphics. The default renderer is for drawing two-dimensional graphics. The P3D renderer makes it possible to draw three-dimensional graphics, which includes controlling the camera, lighting, and materials. The P2D renderer is a fast, but less accurate renderer for drawing two-dimensional graphics. Both the P2D and P3D renderers are accelerated if your computer has an OpenGL compatible graphics card.

The capabilities of Processing are extended with *Libraries* and *Tools*. Libraries make it possible for sketches to do things beyond the *core* Processing code. There are hundreds of libraries contributed by the Processing community that can be added to your sketches to enable new things like playing sounds, doing computer vision, and working with advanced 3D geometry. Tools extend the PDE to help make creating sketches easier by providing interfaces for tasks like selecting colours’0 .

Processing has different *programming modes* to make it possible to deploy sketches on different platforms and program in different ways. The Java mode is the default. Other programming modes may be downloaded by selecting "Add Mode..." from the menu in the upper-right corner of the PDE.

Processing Development Environment (PDE)

The Processing Development Environment (PDE) consists of a simple text editor for writing code, a message area, a text console, tabs for managing files, a toolbar with buttons for common actions, and a series of menus. The menus options change from mode to mode. The default Java mode is documented here.

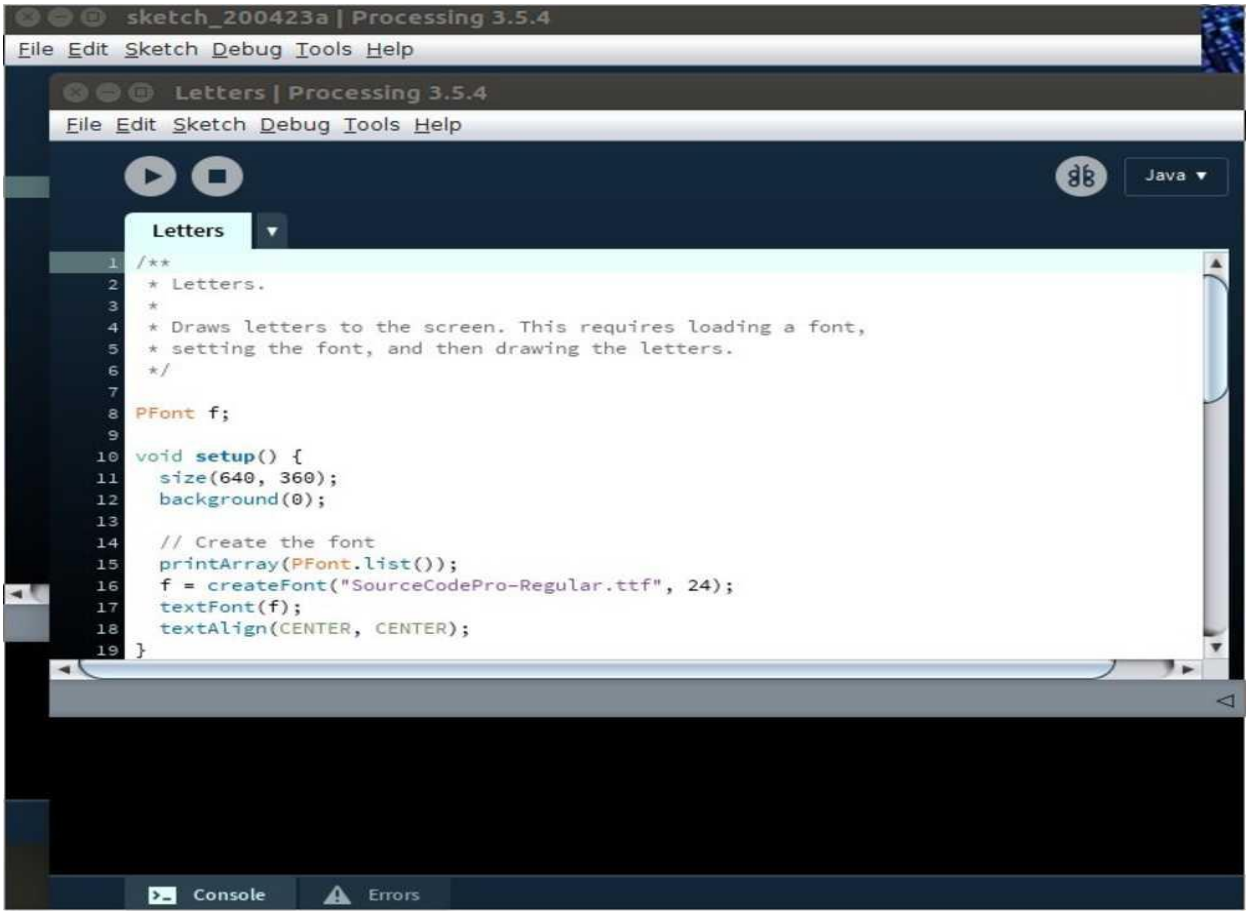


Fig. 3.5: Processing Interface

Programs written using Processing are called sketches. These sketches are written in the text editor. It has features for cutting/pasting and for searching/replacing text. The message area gives feedback while saving and exporting and also displays errors. The console displays text output by Processing sketches including complete error messages and text output from sketches with the print() and println() functions.

3.DHT 11 SENSOR.

The temperature range of DHT11 is from 0 to 50 degree Celsius with a 2-degree accuracy. Humidity range of this sensor is from 20 to 80% accuracy 5% . The sampling rate of this sensor is 1Hz .i.e. it gives one reading for every second. DHT11 is small in size with operating voltage from 3 to 5 volts. The maximum current used while measuring is 2.5mA.

DHT11 sensor has three pins- VCC, GND, Data pin. A pull-up resistor of 5k to 10k ohms is provided for communication between sensor and micro-controller.

Table 1: Main Features of DHT11 sensor

Parameter	Value
Cost	low
Sampling rate	1Hz
Operating Voltage	5 V
pins	3
Measuring Range	0 to 50 degree
Humidity Measuring	**20 to 80%**
size	small
Maximum current used while measuring	2.5mA
version	Dht11

Table 2: DHT11 sensor pinout & Description

No.	Pin Name	Pin Description
1	VCC	The power supply pin of the sensor that mainly operates at 5V DC.
2	Data pin	Input data pin in esp 32
4	Ground	This pin is connected to ground.

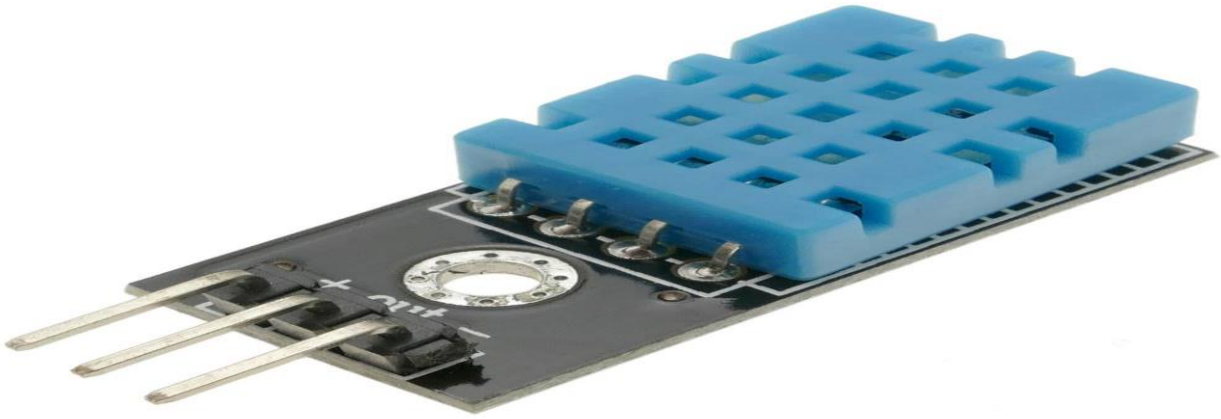


Fig. 3.6: DHT11 sensor

3.5 Jumper Wires

A jump wire (also known as jumper wire, or jumper) is an electrical wire, or group of them in a cable, with a connector or pin at each end (or sometimes without them – simply "tinned"), which is normally used to interconnect the components of a breadboard or other prototype or test circuit, internally or with other.

Fig. 3.9: Jumper Wires

CHAPTER 4

RESULTS AND DISCUSSIONS

CHAPTER 4

RESULTS AND DISCUSSIONS

4.1 RESULTS

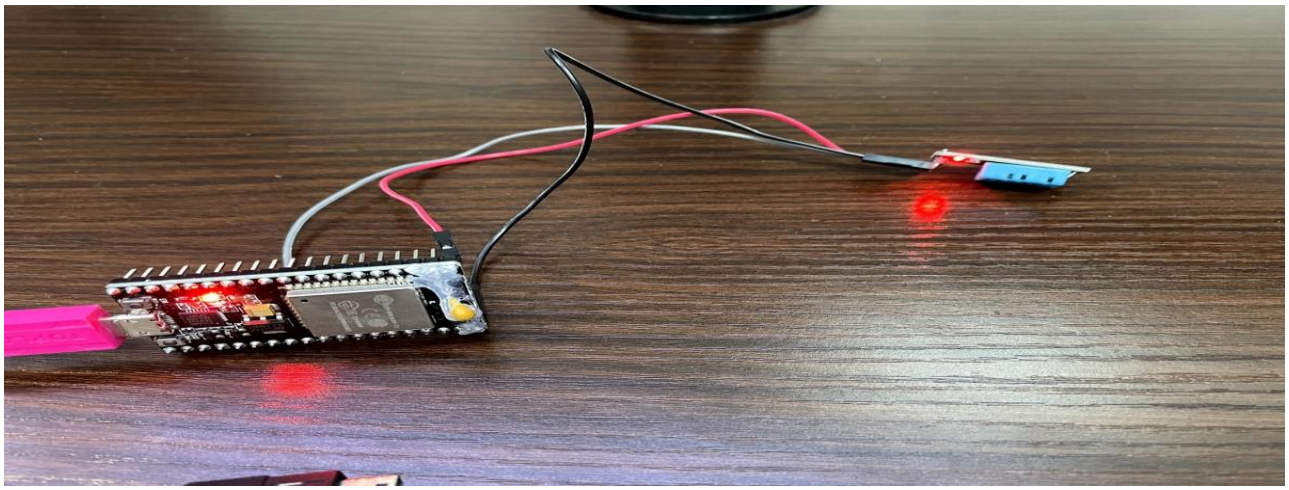


Fig. 4.1 Overall Circuit Operation

figure 4.1 shows the original circuit operation.

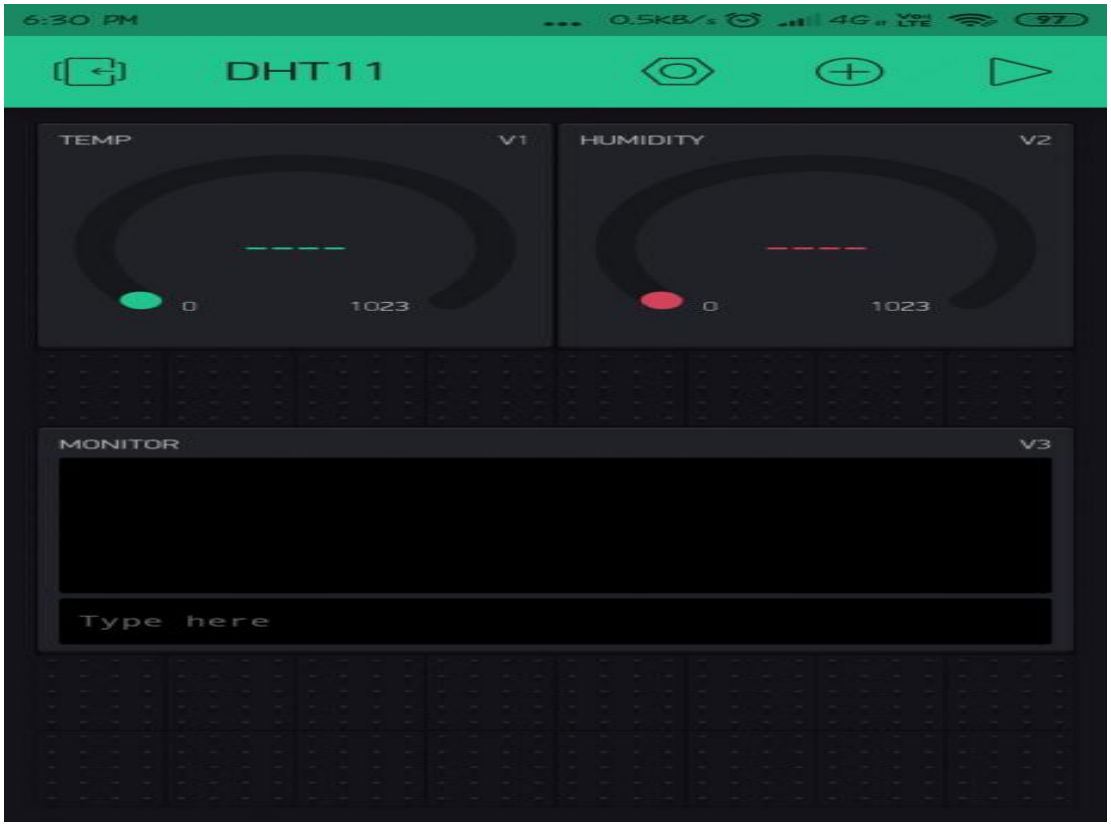


Fig. 4.2: Output when there is DHT not Detect temperature and humidity

As per the above figures, Fig 4.3

Fig. 4.3: Output temperature and humidity

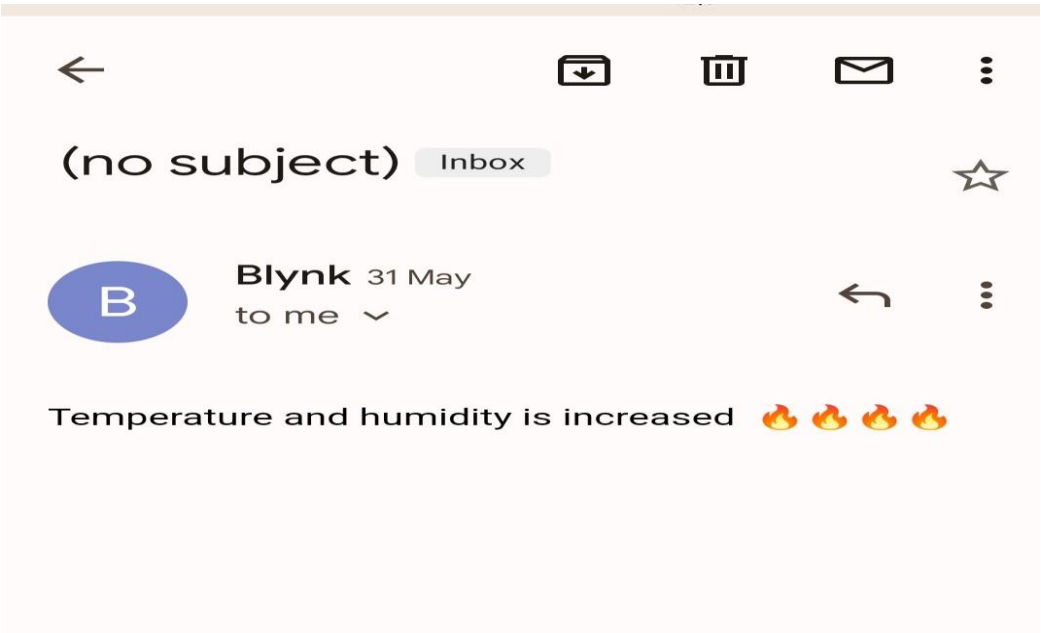


fig 4.4: when temperature greater than 35 degree raise email to user

CHAPTER 5

ADVANTAGES, LIMITATIONS AND APPLICATIONS

CHAPTER 5

ADVANTAGES, LIMITATIONS AND APPLICATIONS

5.1 Advantages

- It is very low cost.
- Fast response.
- Requrie very low power supply 3v or 5v.
- Esp32 has Wifi Module
- Internet access.

5.2 Limitations

. DHT 11 sensor is not a water proof sensor.

. It has limited measurement range.

. It is limited accuracy

5.3 Applications

- DHT11 sensors have been used throughout many applications and industries.
- Medical equipment for Measuring humidity.
- Home automation System.
- Air conditioner.
- Refrigerators.
- Air station to measure air quality.

CHAPTER 6

CONCLUSION AND FUTURE SCOPE

CHAPTER 6

CONCLUSION AND FUTURE SCOPE

6.1 CONCLUSION

The objective of this project is to design and implement DHT11 sensor measures temperature and humidity. If the temperature is greater than 35 degree raises alert to user.

1. This device has wifi module facility.
2. By using Dht11 sensor we created small weather station .
3. This offers a low cost and efficient solution for measuring temperature and humidity.

6.2 FUTURE SCOPE

This is a very economic technology and can be used in several other fields as well, few are listed as below:

1. Can be used as Industries.
2. Can be used for homes and offices.
3. Can be used air quality measure.
4. Can be used to medical.

REFERENCES

REFERENCES

1. Y. B. Gandole, "Simulation and data processing in Dht11 temperature ad humidity measurements", Anadolu University Journal of Science and Technology, Vol.:12, pp.119-127,2011
2. ESP32 - Temperature Humidity Sensor | ESP32 Tutorial (esp32io.com)
3. **https://www.bing.com/search?q=circuit+digest+temperature+and+humidity&qs=n&form=QBRE&msbsrank=1_1__0&sp=-1&pq=circuit+digest+temperature+and+humidity&sc=1-39&sk=&cvid=694727B3B191409DAFF231E30962FB48**
4. **https://www.researchgate.net/publication/336231656_IoT_based_Temperature_and_Humidity_Controlling_using esp32**
5. Humidity and Temperature Measurement using esp32 | Arduino | Maker Pro
6. IoT based Temperature and Humidity Monitoring using ThingSpeak and ESP8266 (iotdesignpro.com)
7. Publishing Temperature and Humidity Sensor Data to AWS - IoT using Raspberry Pi (circuitdigest.com)

APPENDIX

Source code of mini project

APPENDIX: Source code of the mini project

CODE FOR ARDUINO

```
#include <WiFi.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>
#include <DHT.h> //  DHT.h library

#define DHTPIN 2
#define DHTTYPE  DHT11
DHT dht(DHTPIN, DHTTYPE);
#include<Adafruit_Sensor.h>
char auth[]="Par3JQmL9WQdLxlC9STF1ke7LInD7atU";
char ssid[]="Mani";
char pass[]="12345678";
void setup() {
  // put your setup code here, to run once:

Serial.begin(115200);
dht.begin();
Blynk.begin(auth,ssid,pass);
}

void loop() {
  // put your main code here, to run repeatedly:

   float humidity = dht.readHumidity();
float temperature = dht.readTemperature();
if (isnan(humidity) || isnan(temperature)) {
return;
}
Blynk.virtualWrite(V1,temperature);
Serial.print("temperature=");
```

```
Serial.println(temperature);
Blynk.virtualWrite(V2,humidity);
Serial.print("humidity=");
Serial.println(humidity);
Blynk.run();
delay(500);

}
```