IV SEMISTER B.TECH (E & C ENIGINEERING)

EC-281 MINI PROJECT

ENDSEM EVALUATION

REPORT

PRESENTED BY:

M.PRANITHA 221EC132

K VICTOR VISHAL 221EC223

GUIDE:

Mr NIKHIL K S

National Institute of Technology

Karnataka

CERTIFICATE

This is to certify that the thesis entitled, ‘’WEATHER STATION USING ARDUINO” submitted by M.Pranitha and K.Victor Vishal in partial fulfillments for the requirements for the award of Bachelor of Technology Degree in Electronics & Communication Engineering at National Institute of Technology, Karnataka ,is an authentic work carried out by him under my supervision and guidance.

To the best of my knowledge, the matter embodied in the thesis has not been submitted to any other University / Institute for the award of any Degree or Diploma.

Prof. Nikhil K S

Dept. Electronics & Communication Engineering ,

National Institute of Technology

Karnataka, Surathkal - 575025

ACKNOWLEDGEMENT

We would like to articulate our deep gratitude to our project guide Prof. Nikhil K S who has always been our motivation for carrying out the project. It is our pleasure to refer Microsoft Word exclusive of which the compilation of this report would have been impossible. An assemblage of this nature could never have been attempted with out reference to and inspiration from the works of others whose details are mentioned in reference section. We acknowledge out indebtedness to all of them. Last but not the least, our sincere thanks to all of our friends who have patiently extended all sorts of help for accomplishing this undertaking.

M.Pranitha

K.Victor Vishal

EC 281 MINI PROJECT

PROJECT ON

WEATHER STATION USING ARDUINO

ABSTRACT :

This mini project introduces a weather station designed around Arduino microcontroller technology. The primary objective is to create a cost-effective and accessible solution for real-time weather monitoring. The weather station integrates various sensors including temperature, humidity, and barometric pressure sensors, which are essential for collecting accurate weather data. Leveraging the capabilities of Arduino, these sensors are seamlessly integrated into a compact and efficient system.

In the implementation, the Arduino microcontroller serves as the central processing unit for collecting, processing, and displaying weather data. Sensors such as DHT11 or DHT22 for temperature and humidity, and a barometric pressure sensor, are connected to the Arduino board. The microcontroller then processes the data obtained from these sensors and displays it on a user-friendly liquid crystal display (LCD). This interface provides users with instant access to vital weather information in an easily understandable format.

Furthermore, the weather station's versatility extends to its capability for wireless data transmission. By incorporating Bluetooth or Wi-Fi modules, the system can transmit weather data to a computer or smartphone for remote monitoring and data logging. This feature enhances the practicality and usefulness of the weather station, allowing users to monitor weather conditions from anywhere. Overall, this mini project demonstrates the feasibility and effectiveness of utilizing Arduino technology for creating a functional weather station suitable for educational purposes, personal projects, and small-scale weather monitoring applications

TABLE OF CONTENTS :

INTRODUCTION :

Weather monitoring plays a crucial role in various sectors ranging from agriculture and transportation to urban planning and disaster management. Real-time access to accurate weather data is essential for making informed decisions and mitigating potential risks associated with adverse weather conditions. However, traditional weather monitoring systems can be expensive, complex to deploy, and often inaccessible to individuals or small organizations. In response to these challenges, this mini project introduces a cost-effective and user-friendly weather station utilizing Arduino microcontroller technology.

Arduino microcontrollers have gained widespread popularity due to their versatility, affordability, and ease of use in electronics prototyping and DIY projects. Leveraging these advantages, we propose the development of a weather station that integrates various sensors to measure key meteorological parameters such as temperature, humidity, and atmospheric pressure.

The weather station design prioritizes simplicity, accessibility, and functionality, integrating sensors like DHT11 or DHT22 and a barometric pressure sensor for real-time weather data. Arduino serves as the central controller, enabling efficient data collection, processing, and display, while wireless communication modules enhance remote access, showcasing Arduino's potential in democratizing weather monitoring.

OBJECTIVES:

1.Design a compact and cost-effective weather monitoring system utilizing Arduino microcontroller technology.

2. Integrate sensors for measuring key meteorological parameters such as temperature, humidity, and atmospheric pressure.

3. Develop a user-friendly interface for displaying real-time weather data collected by the sensors.

4. Implement data processing algorithms to ensure accuracy and reliability in weather measurements.

5. Incorporate wireless communication modules to enable remote access and data transmission for monitoring and analysis.

6. Explore power-efficient designs to prolong the operational lifespan of the weather station, especially in remote or off-grid locations.

7. Enhance the versatility of the weather station by allowing compatibility with various sensor types and expansion options.

8. Provide customization options for users to configure alert thresholds and data logging intervals based on their specific needs.

9. Facilitate educational initiatives by documenting the project's development process and providing resources for learning about weather monitoring and Arduino programming.

LITERATURE REVIEW: (Sabharwal et.al)

Various smart sensing procedures have been proposed for environmental monitoring. One utilized Zigbee technology and SiLab51F020 microcontrollers for temperature, humidity, pressure, and sunlight measurements, wirelessly transmitting data to a central station. Another approach integrated temperature, humidity, pressure, and luminosity sensors with a GSM module for remote control via SMS, storing 24-hour data. Additionally, a LabVIEW-based weather station mounted sensors on a mast, retrieving and logging data into a database for real-time and historical analysis. Further research explored techniques like Zigbee wireless sensor networks, mesh technology, and advanced systems for broader applications, including irrigation and forest climate monitoring. Sabharwal et al. proposed a system using low-cost sensors for temperature, pressure, humidity, and wind rose calculations, albeit with a bulkier design and a power source requirement.

Arduino-Based Wireless Weather Station Authors: M. Shrestha, S. Shrestha, R. Shrestha.This paper presents the development of a wireless weather station using Arduino and XBee modules. The system measures temperature, humidity, and rainfall. The data is transmitted wirelessly to a base station for display and analysis.

Arduino-Based Weather Station for Environmental Monitoring Authors: A. Rashmi, A. Jyothi, S. Arathi.This paper presents the design and implementation of a weather monitoring system using Arduino. The system measures temperature, humidity, and pressure, and displays the data on an LCD screen. It also provides real-time data logging capabilities. These studies collectively demonstrate the versatility and feasibility of using Arduino for developing weather stations, offering various configurations and functionalities suited for different applications and budget constraints..

METHODOLOGY:

1.Requirement Analysis : Define the objectives and scope of the weather station project . Identify the necessary sensors and components required for measuring temperature, humidity, and atmospheric pressure.

2. Hardware Selection : Choose an appropriate Arduino board based on project requirements and budget constraints.
Select compatible sensors such as DHT11 or DHT22 for temperature and humidity sensing, and a barometric pressure sensor . Determine additional components such as an LCD display for user interface and wireless communication modules for data transmission.

3. Circuit Design and Assembly : Design the circuit layout, considering sensor connections, power requirements, and signal routing . Assemble the hardware components on a breadboard or custom PCB, ensuring proper wiring and connections . Test the hardware setup to verify functionality and identify any potential issues.

4. Arduino Programming : Develop Arduino code to initialize sensors, read sensor data, and process measurements.
Implement algorithms for data calibration, error handling, and calculation of weather parameters . Write code for displaying weather data on the LCD screen and configuring wireless communication modules for data transmission.

5. Integration and Testing : Integrate the Arduino code with the hardware setup, ensuring compatibility and proper communication between components . Conduct rigorous testing to validate the accuracy and reliability of weather measurements**.**

Verify the functionality of the user interface and wireless data transmission features.

6.Optimization and Calibration : Fine-tune sensor calibration settings to improve accuracy and consistency in weather data.
Optimize code efficiency to minimize power consumption and maximize performance . Conduct field tests to assess the weather station's performance under real-world conditions and make necessary adjustments.

7.Document the project's methodology, including circuit diagrams, code snippets, and testing procedures . Create user manuals or guides for assembling, configuring, and operating the weather station . Deploy the weather station in the intended location, ensuring proper installation and calibration for reliable operation.

8.Maintenance and Support : Establish procedures for periodic maintenance, including sensor calibration, battery replacement, and software updates . Provide ongoing support to users, addressing any issues or questions related to the weather station's operation . Collect feedback from users to identify potential areas for improvement and future enhancements**.**

HARDWARE COMPONENTS REQUIRED:

S.no	components
1.	Breadboard
2.	Aurdino Nano
3.	LCD display
4.	BMP-280 Pressure sensor
5.	DHT-11 Temperature and humidity sensor
6.	5v battery
7.	Jumper wires

SOFTWARE REQUIRED: (Arduino IDE)

DHT11 Humidity Temperature Sensor :

Varying temperature and humidity information of the environment are captured by the DHT11 component. It is a Temperature and Humidity Sensor which has a calibrated digital signal output. The DHT11 ensures a high reliability and long-term stability by using the exclusive digital-signal-acquisition technique and temperature & humidity sensing technology. With a resistive-type humidity measurement component and a temperature measurement component, the DHT11 provides a reliable data.

At 0% Relative humidity – expressed as a percentage – the air is considered totally dry, but condenses at 100%.The value of relative humidity can be calculated as:

$$\text{Relative Humidity (RH)} = \frac{Density of water vapour}{Density of water vapor at saturation} \, x \; 100\%$$

The ranges and accuracy of the DHT11 is as follows (D-Robotics, 2010):

- Humidity Range: 20-90% RH
- Humidity Accuracy: ±5% RH
- Temperature Range: 0-50 °C
- Temperature Accuracy: ±2% °C
- Operating Voltage: 3V to 5.5V

DHT11 Temperature and Humidity Sensor

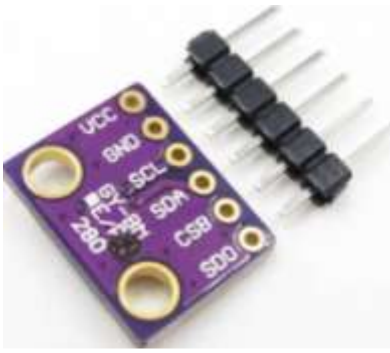
BMP 280 Pressure Sensor :

The BMP280 is an absolute barometric pressure sensor, which is especially feasible for mobile applications. Its small dimensions and its low power consumption allow for the implementation in battery-powered devices such as mobile phones, GPS modules or watches.

The BMP280 is based on Bosch's proven piezo-resistive pressure sensor technology featuring high accuracy and linearity as well as long-term stability and high EMC robustness. Numerous device operation options guarantee for highest flexibility. The device is optimized in terms of power consumption, resolution and filter performance.

the parameters of BMP 280 Pressure sensor:

Operation range	Pressure: 300...1100 hPa Temperature: -40...85°C
Absolute accuracy (950...1050 hPa, 0...+40°C)	~ ±1 hPa
Relative accuracy p = 700...900hPa (Temp. @ 25°C)	± 0.12 hPa (typical) equivalent to ±1 m
Average typical current consumption (1 Hz data rate)	3.4 µA @ 1 Hz
Average current consumption (1 Hz data refresh rate)	2.74 µA, typical (ultra-low power mode)
Average current consumption in sleep mode	0.1 µA
Average measurement time	5.5 msec (ultra-low power preset)
Supply voltage VDDIO	1.2 ... 3.6 V
Supply voltage VDD	1.71 ... 3.6 V
Resolution of data	Pressure: 0.01 hPa (< 10 cm) Temperature: 0.01° C
Temperature coefficient offset (+25°...+40°C @900hPa)	1.5 Pa/K, equiv. to 12.6 cm/K



BMP 280 Pressure Sensor

Liquid Crystal Display (LCD) :

A Liquid Crystal Display (LCD) was used as a monitor showing massages on the screen. A 16 by 2 LCD, as shown in Fig 2, was used for this study which was suitable
for the task. This means that the LCD has two(2) display lines with each line displaying 16
characters.
Although this class of LCD requires a 16 pin connection, a lesser number can be used if
only four(4) data lines are used instead of the default eight (8) data line connection.

Liquid Crystal Display

Jumper:

Jumpers are like on/off switches they may be removed or added to alternate component performance options. A jumper is made of materials that conduct electricity and is sheathed in a nonconductive plastic covering to prevent accidental short circuit. The jumper’s main advantage is its one-time configuration, which make it less vulnerable to corruption or power failure than firmware.

Jumpers

Arduino UNO :

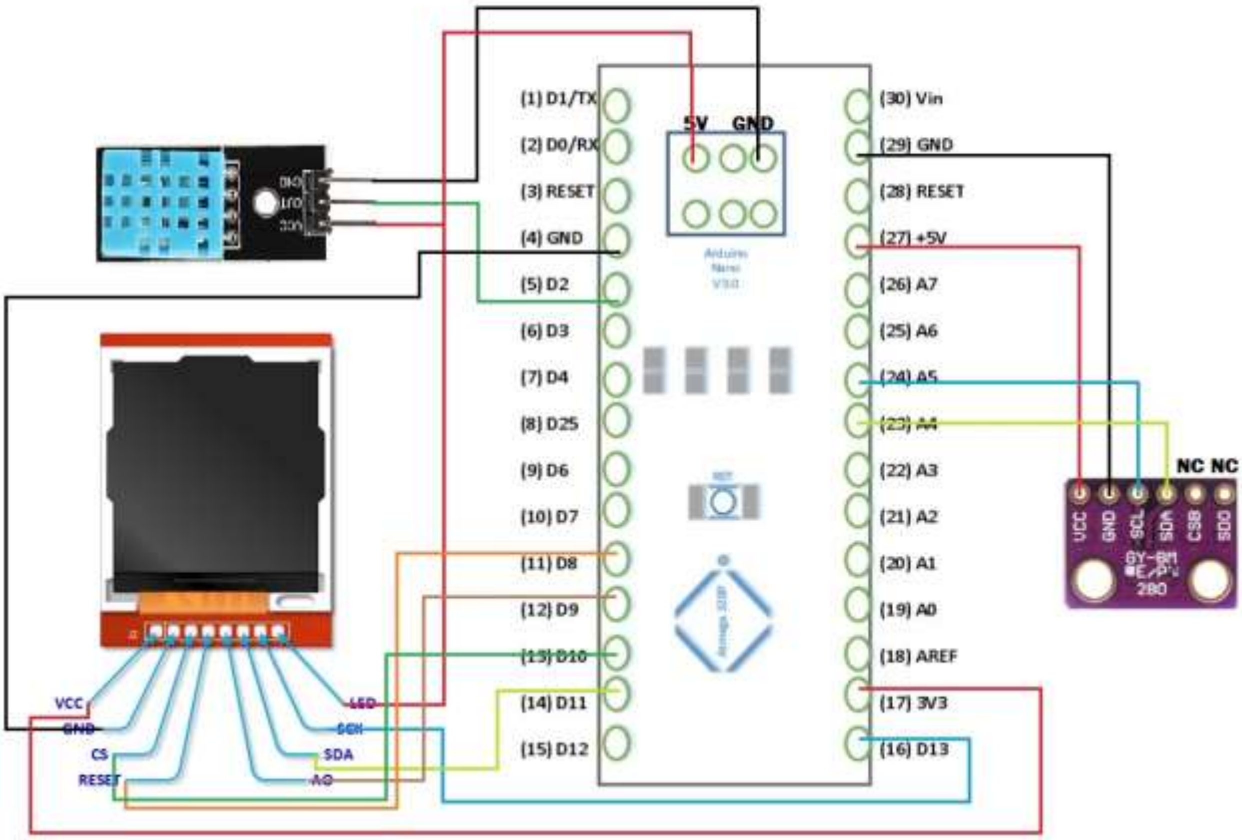
Arduino is an open-source platform comprising of both a physical Programmable Circuit
Board (often referred to as a microcontroller) and a piece of software that can be installed
on the computer, used to write and upload computer code to the physical board. The
Arduino software works on all known operating systems. It is an Integrated Development
Environment (IDE) that provides programmers with tools such as a source code editor,
automation tools, and a debugger (Arduino, 2018). There are several variants of the
Arduino hardware including the Arduino Uno which is used for this study. The Arduino
board is a vital component in this design. It has an inbuilt Atmel ATmega328P microcontroller which reads and reports signals from the DHT11 sensor.

The Arduino Uno has fourteen (14) digital input/output pins, six (6) analog inputs, a Universal Serial Bus (USB) connection, a power jack, a reset button and much more.
It contains everything needed to support the in-built microcontroller. It can be powered via
an AC-to-DC adapter or battery.
Arduino hardware to the computer and the instruction codes are uploaded from the Arduino software.

Arduino Nano

Overall, building a weather station with Arduino is a rewarding project that teaches electronics, programming, and weather monitoring fundamentals.

CIRCUIT DIAGRAM:



The weather station circuit diagram typically consists of various sensors interfaced with an Arduino microcontroller. For instance, a temperature and humidity sensor as DHT11 measures ambient conditions, while a barometric pressure sensor BMP280 detects atmospheric pressure. These sensors are connected to specific digital or analog pins on the Arduino board. Additionally, a rain gauge sensor can be integrated to measure precipitation levels, and a wind speed sensor (anemometer) can be added to gauge wind velocity. These sensors may require pull-up or pull-down resistors depending on their specifications.

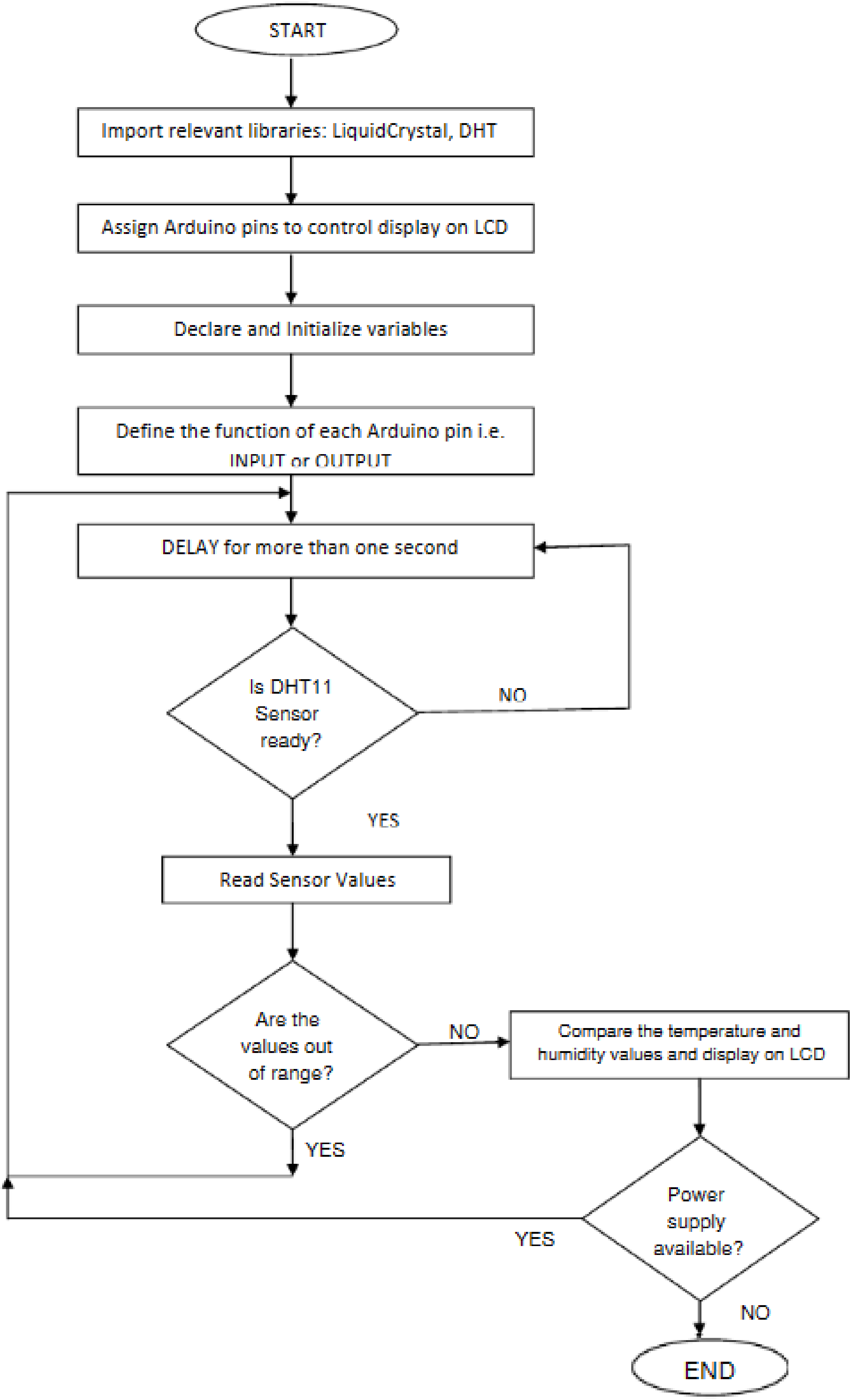
The pin locations are:

Pins on DH11	Pins on Arduino Nano
Signal Pin D0	D2
VCC	5V
GND	GND

Pins in BMP280	Pins on Arduino Nano
VCC	5V
GND	GND
SCL	A5
SDA	A4

Pins on TFT display	Pins on Arduino Nano
VCC	5V
GND	GND
CS	D10
RESET	D8
A0	D9
SDA	D11
SCL	D13
LED	5V

The Flow chart for program design :

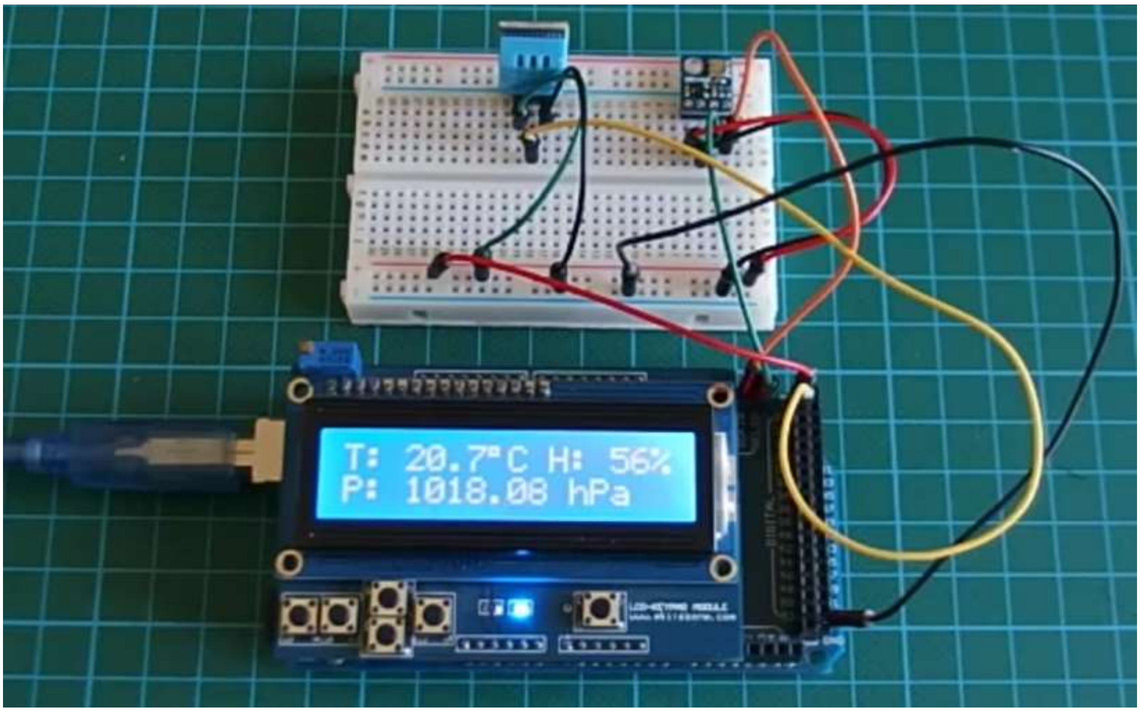


Arduino Sketch :

In this sketch, the Liquid Crystal library is used, which is included with the Arduino IDE, and the DHT library must be downloaded and installed separately. The LCD library displays the readings on the screen, while the DHT library controls the sensor's function.

```
/* Example testing sketch for various DHT humidity/temperature
sensors. Written by ladyada, public domain. */
#include <LiquidCrystal.h>
#include "DHT.h" // Call the DHT library
#define DHTPIN 8 // Pin connected to DHT
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
#define DHTTYPE DHT11 // Define the type of DHT module
DHT dht(DHTPIN, DHTTYPE); // Command to the DHT.h library
void setup() {
dht.begin(); // Start the sensor
lcd.begin(16, 2); // LCD screen is 16 characters by 2 lines
}
void loop() {
float h = dht.readHumidity(); // Value for humidity
float t = dht.readTemperature(); // Value for temperature
t = t * 9 / 5 + 32; // Change reading from Celsius to Fahrenheit
if (isnan(t) || isnan(h)) { // Check that DHT sensor is working
lcd.setCursor(0, 0);
lcd.print("Failed to read from DHT"); // If DHT is not working,
// display this
} else { // Otherwise show the readings on the screen
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Humidity: ");
lcd.print(h);
lcd.print("%");
lcd.setCursor(0, 1);
lcd.print("Temp: ");
lcd.print(t);
lcd.print("f");
}
}
```

WORKING IN HARDWARE:



- The weather station comprises of three parts : collecting the real-time weather data , processing the data in the Arduino board, showing the data to the user on the LCD.

- The sensors DHT11 and BMP-280 collect the data in real-time . That data is then converted into electrical signals . These signals are then sent to the Arduino board.

- Arduino board compiles these signals. Then sends the appropriate data in the form of electrical signals to the TFT LCD.

- The LCD then shows the final output to the user.

- All these processes happen in real-time. So the data also gets updated in real time.

WORKING SHOWING IN IOT CLOUD:

- **Hardware Setup**: Connect an ESP8266 microcontroller to a DHT11 or DHT22 humidity and temperature sensor using jumper wires on a breadboard.

- **Arduino IoT Cloud Setup:** Sign up for Arduino IoT Cloud, create a new "Thing," define properties for humidity and temperature, and generate an API key for authentication.

- **Programming ESP8266**: Write code to read humidity and temperature data from the DHT sensor and send it to Arduino IoT Cloud using Wi-Fi. Include Wi-Fi credentials and use the ArduinoIoTCloud library for easy integration.

- **Upload the Code:** Connect the ESP8266 to your computer via USB, select the correct board and port in Arduino IDE, and upload the code to the ESP8266.

- **Monitor the Data**: Once uploaded, the ESP8266 will start sending humidity and temperature data to Arduino IoT Cloud. Monitor this data in real-time through the Arduino IoT Cloud dashboard.

- **Optional Enhancements:** Add features like data logging, notifications, or control mechanisms based on humidity thresholds. Ensure robustness by handling network issues and sensor failures gracefully in your code.

RESULTS & DISCUSSIONS : (HARDWARE)

The Arduino IDE was used in developing the sketches that were uploaded as firmware into
the microcontroller. Thereafter, the system could work without the user's intervention.
Libraries are required for a robust firmware development using Arduino. In this case, we
used the 'Liquid Crystal' and 'DHL' libraries. Next we set the Arduino pins and attached
them to the LCD for display. Arduino pins 9, 10, 4, 5, 6, 7 were attached to the RS, E, D4,
D5, D6, D7 pins respectively on the LCD. The 'pin Mode' of Arduino pin 12 was set as
INPUT. This is the pin that reads the numeric values from the signal pin of the DHT11
sensor. At least a second delay is required to get reliable readings from the DHT11 sensor.
However, we used three(3) seconds delay to ensure that the previous values have been
displayed. It is also important to confirm that the temperate and humidity readings are
within the acceptable range for the sensor. In this work the humidity range was between 20
- 90 relative humidity, while the temperature ranged between 0 - 500c. Once the read values
are within range, it is displayed on the LCD screen

Weather forecasting system

Internal circuit Connections

RESULTS & DISCUSSIONS : (IOT CLOUD)

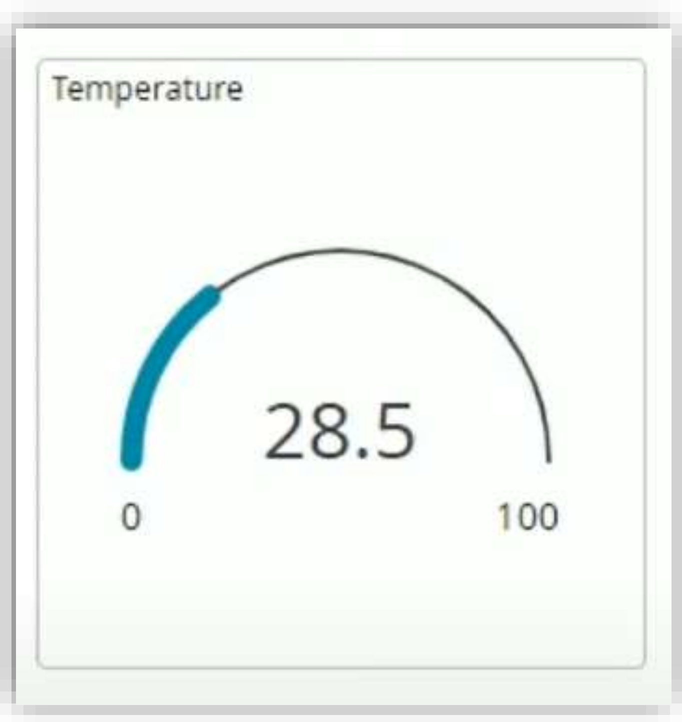
RESULTS WERE SHOWING IN THE IOT CLOUD MONITOR:

Connecting ESP8266 to the Cloud and defining the Wi-Fi credentials in with our code:-

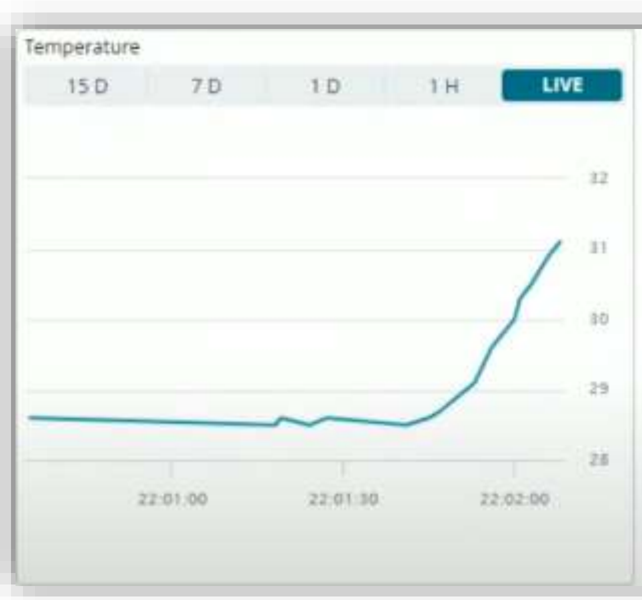
- Setting up a Thing in Arduino IoT Cloud with properties for humidity and temperature.
- Connect your ESP8266 to the Cloud and define Wi-Fi credentials in your code.
- Program the ESP8266 to read humidity and temperature from the sensor.
- Use the ArduinoIoTCloud library to send this data to the Cloud.
- Upload the code to the ESP8266 and ensure it's connected to Wi-Fi.
- Monitor the real-time values of humidity and temperature in the Cloud dashboard.
- Visualize the data using widgets like line charts or gauges.
- Customize the dashboard appearance and behavior as needed.
- Implement alerts or triggers for specific conditions, like humidity thresholds.
- Enjoy real-time monitoring and control of humidity and temperature from anywhere via the IoT Cloud dashboard.

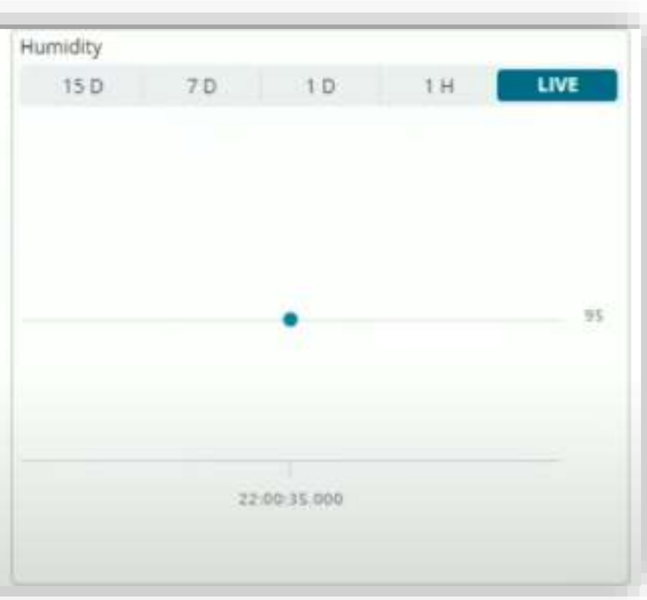
SCENERIO OF OUTPUT:

Temperature and Humidity showing widgets:



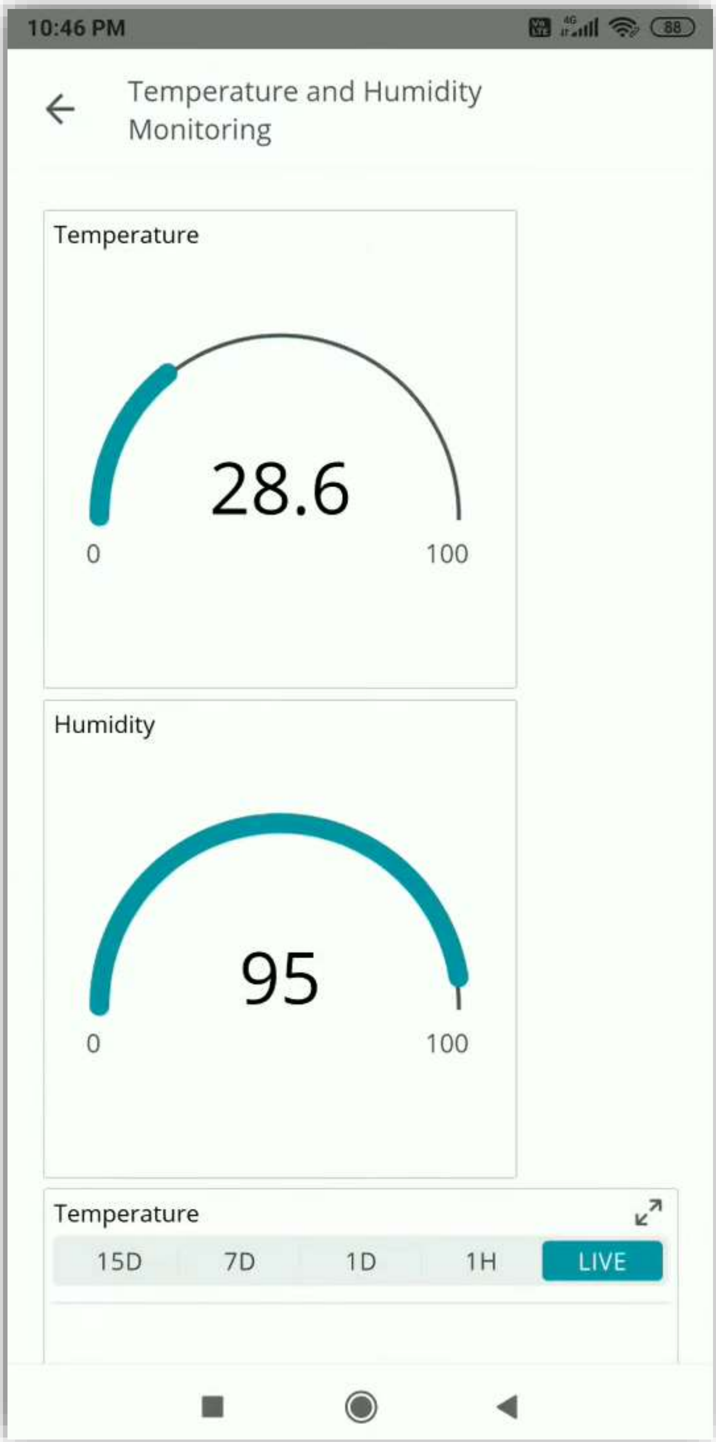
Temperature and Humidity charts:





SCENERIO OF OUTPUT: (MOBILE PHONE)

Temperature and Humidity showing widgets:



Temperature and Humidity charts:

FUTURE SCOPE:

- Advancements in Sensor Technology: Ongoing advancements in sensor tech promise more accurate and versatile data collection, covering parameters like air quality and soil moisture alongside traditional weather metrics.

- Integration with IoT and Cloud Services: Future weather stations will seamlessly integrate IoT capabilities, enabling real-time data transmission and remote monitoring via cloud services for enhanced accessibility and analysis.

- Enhanced Data Visualization and Analysis Tools: Expect improved user interfaces and mobile apps for intuitive access to weather data. Advanced analytics algorithms will enable predictive modeling and anomaly detection, enhancing the station's utility for various applications.

- Customization and DIY Community: Arduino's open-source ecosystem will continue to foster a vibrant DIY community, leading to the development of tailored sensor modules, enclosure designs, and software libraries for personalized weather monitoring solutions.

CONCLUSION :

- The implementation of a weather station using Arduino offers a cost-effective and versatile solution for environmental monitoring.

- Through the literature review, it is evident that Arduino-based weather stations can be tailored to meet specific requirements while providing reliable data collection and analysis capabilities.

- In conclusion, Arduino-based weather stations offer a practical solution for monitoring environmental conditions with the potential for further advancements and applications in fields such as agriculture, meteorology, and environmental research.

- As technology continues to evolve, Arduino-based weather stations are likely to play a significant role in enhancing our understanding of the environment and mitigating weather-related risks.

- We can determine if the weather is HOT, NORMAL, or COLD based on the air temperature and humidity read by a DHT11 sensor and pressure read by BMP 280. All the components used in this project were cased with plastic foam – which would have otherwise being discarded as waste. The circuit diagram and the component connections used for the design are presented.

- This weather monitoring system will provide farmers, pharmacists, event planners and others with accurate information to guide them to take appropriate action.

REFERENCES:

1. Danladi A., Stephen M., Aliyu B. M., Gaya G. K., Silikwa N. W., Machael Y. (2017). Assessing the influence of weather parameters on rainfall to forecast river discharge based on short-term, Alexandria Eng. J. (2017).

2..Doiron T. (2007). 20°C—A Short History of the Standard Reference Temperature
for Industrial Dimensional Measurements. Journal of Research of the National Institute of Standards and Technology, Volume 112, Number 1, January-February
2007.

3.Shelton, D. P. (2008). Air Properties: Temperature and Relative Humidity. OneGuide, University of Nebraska-Lincoln Extension, Institute of Agriculture and Natural Resources.

4.Fisher T. (2017). What is Firmware?. Retrieved online on 10 February 2018.

5. G. J. W. Visscher, Meas. Sci. Technol. 6, 1451 (1995).2. J. R. Simões-Moreira, Meas. Sci. Technol. 10, 302 (1999)