International Journal of Trend in Scientific Research and Development (IJTSRD)
Volume 8 Issue 6, Nov-Dec 2024 Available Online: www.ijtsrd.com e-ISSN: 2456 – 6470

IoT-Based Weather Tracker with Data Logging Systems

Samrat Dutta

Student, M.Sc. Computer Science, Department of Computer Science and Engineering,
University of Calcutta, Kolkata, West Bengal, India

ABSTRACT

Weather data is crucial for analysing and predicting natural calamities. Accurate weather data collected over large areas with precision can help not only in predicting rainfall, storms, etc. but also aid farmers in planning the watering of their crops. This data is vital for weather stations, meteorological departments, aviation, marine industries, and the agricultural industry. The proposed project consists of two modules created with sensors and microcontrollers: a Weather Tracker Transmitter (WTT) module and a Weather Tracker Receiver (WTR) module. The WTT collects various real-time weather data, such as temperature, humidity, air pressure, precipitation, altitude, GPS data, latitude, and longitude, and transmits this data wirelessly to the WTR module using 2.4GHz radio waves. The project aims to collect weather data over large areas with very high precision using the WTT module, transmit it wirelessly to the WTR module, and provide this data in a CSV file stored on an SD card for future analysis and prediction. The project also uploads the collected data to a server for visualisation on mobile and web applications. Compared to other devices, this system is capable of collecting real-time weather data over very large distances, with up to 3,124 WTT modules. These modules could be spread over a 7 km radius from the WTR module to collect weather and other environmental data.

KEYWORDS: *IoT, Data Logging, Weather, Microcontrollers, BME280 Sensor, NEO-6M GPS*

How to cite this paper: Samrat Dutta "IoT-Based Weather Tracker with Data Logging Systems" Published in International Journal of Trend in Scientific Research and Development (ijtsrd), ISSN: 2456-6470, Volume-8 | Issue-6, December 2024, pp.871-885, URL: www.ijtsrd.com/papers/ijtsrd72710.pdf

IJTSRD72710

1. INTRODUCTION

1.1. Domain Description

The weather has always had a significant influence on people's lives, shaping their cultures, habits, attitudes, behaviours, and environments in general. Weather monitoring traditionally relies on ground-based weather stations equipped with both mechanical and digital sensors. These stations use thermometers, barometers, hygrometers, anemometers, wind vanes, and rain gauges. Weather balloons, filled with helium, are launched into the atmosphere to collect weather data and carry similar instruments. Additional methods for collecting traditional weather data include satellites, weather radars, and more. While these methods are effective and accurate, they come with high installation and maintenance costs. Among the various weather parameters, the most important ones affecting our daily lives are temperature, humidity, air pressure, precipitation, and altitude. Collecting these data with precision leads to more accurate predictions of future weather events. Manual observations are also taken into consideration [3, 4].

All these collected data are then sent to meteorological centres, where various computer models are used to analyse the data and predict weather conditions. These predictions are based on an understanding of historical data and are often very accurate, although they sometimes fail. Once the predictions are made, they are disseminated to the public through various channels, including television, radio, internet websites, and more [1, 5]. The entire process of determining whether it will be a cloudy or sunny day, or whether there will be any chances of precipitation, involves a complex journey. It starts with collecting data using the aforementioned methods, organising it, and transmitting it to the meteorological centres. There, the data is analysed, and predictions are made and then communicated to the public.

1.2. Motivation and Scope of Work

In earlier times, people used manual methods to understand the weather and collect data. Farming practices, in particular, were based on traditional knowledge and repeated at regular intervals.

However, weather patterns change annually, necessitating updated practices. Accurate real-time data collection is essential for making informed decisions, which is challenging with both expensive methods and cheap manual methods [1]. Achieving the right balance is the goal of the Weather Tracker with Data Logging project. Today, many digital sensors can collect the aforementioned weather data with high accuracy and at a low cost. By combining these sensors with microcontrollers—integrated circuits designed to govern specific operations in embedded systems—we can collect weather data, transmit it to a base station, and process it. The collected data is often used for prediction through machine learning, requiring an organised format. This can be achieved by saving the data into a CSV file. Microcontrollers enable the collection, transmission, and storage of weather data in CSV files on SD cards, which can be very useful to those needing weather and environmental data. The Internet of Things (IoT) is also beneficial when integrated with the system. A microcontroller like the ESP32, which has a built-in WiFi module, leverages IoT capabilities. IoT makes virtually everything "smart" by enhancing aspects of our lives through the power of data collection, AI algorithms, and networks. The IoT process begins with devices themselves, such as smartphones, smartwatches, and electronic appliances like TVs and washing machines, which communicate with the IoT platform (Smith, 2021) [9]. IoT allows the collected data to be uploaded to a web server and visualised through mobile or web applications from anywhere in the world. This enables the automation of many tasks based on real-time data collection.

These collected data can help farmers understand when to water their crops more or less, predict precipitation levels, and much more. Fields such as weather forecasting, disaster management, energy management, military operations, transportation, construction, research, and education, among others, can benefit from the collected data. This is a cheaper and quicker solution compared to traditional methods for providing useful and accurate weather data.

2. Background of Existing Literature

The authors in [1], proposed a system that addresses the challenge of understanding live environmental conditions using IoT-based real-time weather monitoring in the Gorakhpur Region. It employs a client-server architecture with two-tiered sensor monitoring (temperature, humidity, rainfall, atmospheric pressure) via NodeMCU and Arduino IDE. Data is transmitted to the cloud via serial monitor and IP address, visualised on a web server using HTTP protocol. This setup offers universal access to weather data, aiding informed decision-making with publicly available, accurate information. The system aims to enhance weather monitoring in Gorakhpur, surpassing traditional methods in performance and accessibility.

The authors in [2], proposed a smart weather reporting system that leverages web-based climate parameter monitoring, allowing users to directly access real-time weather details without relying on a weather forecasting agency. It integrates temperature, humidity, and rainfall sensors to monitor and report weather statistics continuously. The system utilises sensors to gather data, which is processed by a microcontroller and transmitted to an online web server via Wi-Fi for live updates. Users can set alerts for specific conditions, receiving notifications if weather parameters exceed predefined values. Additionally, the system includes a soil moisture monitoring feature that updates moisture levels and charts on platforms like ThingSpeak, offering an efficient IoT-based solution for weather reporting and soil monitoring.

The authors in [3], proposed a system that is an advanced IoT-based solution for weather monitoring, offering real-time data accessibility over a wide range. It monitors various weather and climate parameters such as temperature, humidity, wind speed, moisture, light intensity, UV radiation, and carbon monoxide levels using multiple sensors. Data from these sensors is transmitted to a web page where it is displayed as graphical statistics, accessible globally and useful for future reference. An accompanying mobile app provides alerts for sudden weather changes. For complex weather forecasts, beyond sensor capabilities, an API analyses sensor data to predict outcomes accurately and can be accessed anytime and anywhere. The system's compact design with minimal moving parts ensures low maintenance and its energy-efficient components can be solar-powered, making it cost-effective compared to existing solutions. This system promises significant benefits for meteorological departments, weather stations, aviation, marine industries, and agriculture, enhancing monitoring capabilities across various sectors.

The authors in [4], proposed a project that leverages IoT technology to develop a weather monitoring and reporting system aimed at enhancing the accuracy and reliability of weather data. Using sensors and IoT devices, the system collects real-time data on weather conditions like temperature and humidity. This data is then analysed to generate precise weather reports, crucial for sectors such as agriculture, aviation, and disaster management. The goal of this real-time weather monitoring system is to provide timely and accurate information via web and mobile dashboards, empowering users to make informed decisions based on current weather conditions.

The authors in [5], developed a cost-effective weather monitoring device using an Arduino Mega 2560 microcontroller and various sensors, equipped with a real-time data logger and an LCD display. The system offers an on-device display, compact size, affordability, flexibility, portability, and ease of operation. It aims to provide meteorological measurements effectively and is suitable for small-scale users such as farmers, travellers, aircraft personnel, researchers, and institutions. The data collected by the device showed good agreement with established sources, ensuring reliable weather information for monitoring, analysis, and forecasting. Ultimately, the device aims to mitigate climate-induced disasters by providing accessible and accurate climate data to its users.

The author in [6], proposed a weather monitoring system to track dynamic climatic conditions, collecting data essential for weather forecasting and understanding local environmental changes. This data is valuable for studying global climate trends and can be applied across diverse fields such as mining, geology, agriculture, and meteorology. The project focuses on developing a simple IoT-based weather monitoring system using Particle Photon, an Arduino-compatible IoT board. Developers register the Photon board on the Particle website, write and deploy code using Particle's Web IDE, and wirelessly upload it to the board via the Particle cloud service. This system monitors temperature and humidity, providing real-time data and operational insights based on programmed instructions.

The author in [7], proposed a system that is an advanced IoT-based solution for monitoring and displaying weather conditions globally. Using Internet of Things technology, it connects various sensors and devices to the Internet, enabling real-time monitoring and control of environmental factors such as temperature, humidity, light intensity, and CO levels. Sensor data is transmitted to a web page where it is graphically represented, providing accessible updates from anywhere in the world.

The authors in [8], proposed a project that offers real-time reporting of weather conditions, including temperature, humidity, moisture, and rainfall levels. It addresses the limitations of SMS-based systems, which are restricted in the number of recipients and increase in time with more recipients. This system allows scientists and analysts worldwide to monitor specific environments like volcanoes or rainforests remotely and access weather data instantly from any location via web platforms.

The authors in [9], proposed a system that utilises IoT technology to collect, analyse, and distribute real-time weather data efficiently. It deploys interconnected sensors across diverse locations to monitor parameters like temperature, humidity, air pressure, and rainfall. Data from these sensors is transmitted to a central hub for processing using microcontrollers and wireless communication modules. Advanced algorithms ensure data accuracy and quality, which are accessible via a user-friendly web application and mobile interface. The system supports scalability and remote management through cloud-based architecture, providing reliable weather information and alerts to users globally.

The authors in [10], proposed a system that revolutionises weather data collection and utilisation across industries like aviation, agriculture, and disaster management offering real-time data that improves decision-making with accuracy and cost-effectiveness compared to manual methods. Farmers can optimise crop management, while industries plan safer routes and disaster agencies prepare for extreme events. Despite benefits, challenges include data security risks due to wireless communication and the maintenance needs of sensors exposed to harsh conditions, which require regular upkeep and calibration. Addressing these ensures reliable data for informed decision-making and scientific research on weather patterns and climate change.

The descriptions above effectively illustrate the integration of sensors, microcontrollers, and IoT to create a cost-effective, reliable, and accurate weather monitoring system. Project Weather Tracker introduces an additional feature: the capability to deploy multiple Weather Tracker Transmitter (WTT) modules. This enhancement sets it apart by enabling data collection across a wide 6km radius, significantly increasing its utility. Furthermore, the integration of GPS allows precise mapping of latitude and longitude values for each data collector module, facilitating detailed geographical analysis. Collecting weather data wirelessly over a span of 6km with GPS coordinates will enable us to analyse the data with much greater accuracy, leading to more precise predictions.

3. Technical Details

3.1. Software Components Required

Name	Description
Arduino IDE Courtesy: https://www.arduino.cc/en/software	To program the microcontrollers Atmega328p and ESP32
EasyEDA Courtesy: https://easyeda.com/	To create the circuit schematics for the transmitter and receiver
draw.io	To create flowcharts and block diagrams
Fusion 360 (Student Version)	It is used for CAD. It generates 'stl' files of the 3D models. The enclosures of the modules are made using this software.
PrushaSlicer Courtesy: https://www.prusa3d.com/	It is used to generate the g-code from the 'stl' files for the 3D printer

3.2. Libraries and Packages Required

Name	Description
Communication a. Wire.h b. SPI.h c. RF24.h d. SD.h e. SoftwareSerial.h	These libraries are responsible for the communication between the sensor modules and wireless transceiver modules using SPI, Serial and I2C protocols. The Software Serial library is used to convert normal GPIO pins into Serial Tx and Rx pins for Serial communication.
Sensor a. Adafruit_Sensor.h b. Adafruit_BME280.h c. TinyGPSPlus.h	These libraries are responsible for the data interpretation and conversion of raw data into necessary units from the BME280 sensor and GPS Module.
WiFi a. WiFi.h b. WiFiClient.h c. WifiManager.h	These libraries are responsible for establishing a connection between the ESP32 microcontroller and the Internet using WiFi in order to upload data to the Blynk Server. They also host an access point for choosing the WiFi router to connect.
Blynk Server a. BlynkSimpleEsp32.h	This library is responsible for uploading and downloading data from the Blynk Server.
Display a. LiquidCrystal_I2C.h	This library is responsible for connecting to the LCD modules with their I2C bus addresses and helping to print the text and custom characters in them.
Memory a. EEPROM.h	This library is used to manage the EEPROM of the microcontroller to store SSID, Passwords, and other crucial data

3.3. Hardware Needed

Name	Description	Specification
Arduino Nano *Courtesy:* https://www.alldatasheet.com/datasheet-pdf/pdf/1425641/ATMEL/ATMEGA328P.html	Arduino is a development board that uses the ATmegs328 microcontroller unit. It can be programmed using the inbuilt CH340 chip via USB protocol. It is a low-cost MCU that has ADC and plenty of GPIO pins. It also offers inbuilt protocols like SPI, UART and I2C. It operates on 5v and has an internal oscillator of 8Mhz and an external crystal oscillator is used of 16MHz.	Operating Voltage (logic level): 5V 8 analog input ports: A0 ~ A7 14 Digital input/output ports: Tx, Rx, D2 ~ D13 1 pair of TTL level serial transceiver ports Rx/Tx Using Atmega328P MCU
ESP32 *Courtesy:* https://www.alldatasheet.com/datasheet-pdf/pdf/1243003/ESPRESSIF/E	The ESP32 is an MCU that comes with built-in WiFi and Bluetooth. It has a total of 3 cores, out of which one is an efficiency core. The WiFi tasks are handled by Core 0 and the programs written are executed in Core 1. It has an adjustable clock frequency of 80 to 240 MHz. ESP32S has a set of peripherals, ranging from capacitive touch sensors, hall sensors, low-noise sense amplifiers, and communication protocols	Operating voltage: 2.3v ~ 3.6v Operating current: 80mA Clock Frequency: 80 ~ 240 MHz Flash memory: 4MB Data Rate: 54 Mbps

SP32.html	like SPI, UART, and I²C. In addition, this module can also support data rates up to 150Mbps.	SRAM Memory: 512KB Antenna Type: PCB antenna
BME 280 *Courtesy:* https://www.alldatasheet.com/datasheet-pdf/pdf/1132060/BOSCH/BME280.html	The BME280 sensor is a versatile and precise chip initially developed for advanced smartphone applications. It offers comprehensive weather data capabilities through an I2C interface, allowing for accurate local weather predictions. Alongside measuring pressure and temperature with high accuracy, the BME280 also includes humidity-sensing capabilities. It supports both I2C and SPI interfaces, accommodating various connectivity needs with options for 3- or 4-wire SPI configurations. The sensor's integration on PCB ensures reliability and ease of use in diverse environmental monitoring and forecasting applications.	Operating Voltage: 1.8v ~ 5v Interface: I2C (3.4MHz) Operating Range Temperature: -40 ~ 85 °C Humidity: 0-100% Pressure: 300-1100 hPa Resolution: Temperature: +-1°C Humidity: +-3% Pressure: +-1Pa I2C Address: SDO LOW => 0x76 SDO HIGH => 0x77
NEO-6M GPS *Courtesy:* https://content.ublox.com/sites/default/files/products/documents/NEO-6_DataSheet_%28GPS.G6-HW-09005%29.pdf	The NEO-6M GPS module is a high-performance GPS receiver featuring a built-in 25 x 25 x 4mm ceramic antenna, ensuring robust satellite signal acquisition capabilities. It includes power and signal indicators for monitoring module status, and a data backup battery preserves information during accidental power loss. With 3mm mounting holes, it facilitates easy assembly on aircraft, enabling stable flight operations such as fixed-position hovering, automatic return-to-home, and waypoint navigation. Alternatively, it can be used in smart robot cars for automated return or navigation to specific destinations, enhancing its utility in autonomous applications. Key specifications include a default baud rate of 9600, module dimensions of 23mm x 30mm, antenna size of 12 x 12mm, and a 20mm cable.	Channels: 50 GPS L1 frequency, C/A Code SBAS: WAAS, EGNOS, MSAS Main Chip: NEO-6M Sensitivity: -160 ~ 156 dBm Navigation Update Rate: 5Hz Avg Cold Start Time: 27 s Warm Start Time: 27 s Maximum Speed: 500 m/s
DS3231 RTC *Courtesy:* https://www.alldatasheet.com/datasheet-pdf/pdf/254832/MAXIM/DS3231.html]	The DS3231 RTC module is a precise and cost-effective real-time clock (RTC) featuring a built-in 10-bit temperature sensor with a resolution of 0.25°C. It utilises an integrated temperature-compensated crystal oscillator (TCXO) and crystal, ensuring high accuracy over time. The module includes a battery input to maintain accurate timekeeping even during power interruptions. By integrating the crystal resonator, the device achieves enhanced long-term accuracy while simplifying manufacturing processes due to reduced component count. The DS3231 RTC module is available in commercial and industrial temperature ranges and is housed in a 16-pin, 300-mil SO package.	Operating Voltage: 2.7 ~ 5.5v Battery: 2032 Coin Battery. I2C interface Fast (400kHz) I2C Interface. EEPROM: AT24C32 32Kbit Serial I2C. 10 bit, ±3°C Accuracy and 0.25C resolution.
NRF24L01 *Courtesy:* http://api.supply	This transceiver module operates on 2.4GHz radio waves, enabling half-duplex communication. It operates at 3.3 volts and features both low and high power modes. The module offers a range of	Operating speeds: 2Mbps, GFSK modulation efficiency, Anti-interference ability,

frame.com/v1/t?d=50d38f4ap&p=NRF24L01PT&s=NRF24L01&h=cdt7BwbtjCVJ5X9DbLSWg&currency=&h_crc=6617936e48d6bff5655932f71d018833	approximately 900 to 1000 meters in open spaces and can transmit data packets of up to 32 bytes at once. This module is also capable of actively listening to 6 other modules at once.	125 Channels, Multi-point communication Frequency hopping to meet the communication needs Built-in hardware CRC error detection Low-power 1.9 ~ 3.6V, only 1uA on Power down mode
Micro SD Card Reader Module	The Micro SD Card Reader Module, or Micro SD Adapter, supports dual I/O voltages and facilitates seamless data transfer with standard SD cards. It is compatible with Arduino and other microcontrollers through a pinout that supports the SPI interface, enabling connectivity with any SD card. The module operates with either 5V or 3.3V, making it suitable for Arduino UNO/Mega. Common applications include data logging, audio, video, and graphics. Its six pins include GND, VCC, MISO, MOSI, SCK for SPI bus, and CS for chip select, along with a built-in 3.3V regulator for stable Micro SD card operation.	Supports: Micro SD Card (<=2G), Micro SDHC card (<=32G) (high-speed card) Operating voltage: 3.3 or 5.5v Communication: Standard SPI interface.
20x4 Character Y-B LCD Display Module	The 20×4 character Y-B LCD display module is a versatile device known for its compact size and clear contrast, featuring a grid of 20×4 characters. Using Y-B technology ensures enhanced visibility from multiple viewing angles. With 80 characters per screen, it provides ample space to display data effectively. This module is widely used in electronic projects, instrumentation, and industrial applications, offering reliable and informative visual output for diverse purposes.	Model: GM2004A Resolution: LCD1: 20×4 LCD2: 16x2 Operating Temperature: -20 to 70 °C No. of Pins: 16 Controller: SPLC780D
9v or 12v Power Adapter	Both the transmitter and receiver modules are made such that they can be powered with a 9-volt to 12-volt power supply	Output Voltage: 9v or 12 v Regulated DC Current: 1A
Connecting Wires and Soldering Station	All the connections are made with 22 to 28 AWG multithreaded wires	Wire AWG: 22 to 28
PLA Filament	PLA is polymerised lactic acid. It is very common and easy to print 3D printing filament.	Diameter: 1.75 Printing Temperature: 220 °C
3D Printer [Creality Ender 3 S1 Pro]	Fused Deposition Modeling (FDM) is a popular type of 3D printing technology known for its accessibility and versatility. It works by melting thermoplastic filaments, such as PLA or ABS, and extruding them layer by layer through a heated nozzle onto a build platform. This process allows for the creation of complex geometries and functional prototypes with ease. FDM 3D printers are widely used in industries ranging from engineering and product development to education and hobbyist applications due to their affordability, relatively simple operation, and capability to produce robust parts.	Build Volume: 220 x 220 x 250 mm Print Accuracy: ±0.1 mm Layer Thickness: 0.1 - 0.4 mm Filament Diameter: 1.75 mm Print Speed: Up to 180 mm/s

4. Methodology

4.1. Working of the Project Weather Tracker with Data Logging:

The project consists of two main components: a series of transmitter devices called *WTT* and one receiver device called *WTR*. The WTT device collects various weather data such as temperature, air pressure, precipitation, and humidity using the BME280 sensor via the I2C protocol. The altitude is calculated using the barometric formula and the air pressure along with pressure at sea level. Additionally, GPS data like latitude and longitude is captured using the NEO-6M module, and a data packet is created after validating its correctness. This packet with the transmitter's unique ID is then sent to the NRF24L01 transceiver module using the SPI protocol. The transmitter has the NRF24L01 configured to act solely as a transmitter with a unique address for identification within the network. This data is then transmitted to the allocated receiver and eventually reaches the receiver at the root of the network tree.

Upon receiving the packet, the WTR device validates its correctness and processes the data. The distance between the transmitter and the receiver is calculated using the Haversine formula. This data is displayed on two LCD screens and also sent to the Blynk server for visualisation through mobile and web applications. The data is also logged into an SD card into a CSV file for future analysis. Additionally, the receiver module includes an RTC module that keeps track of time even when the power is off.

4.2. NRF24L01 Network

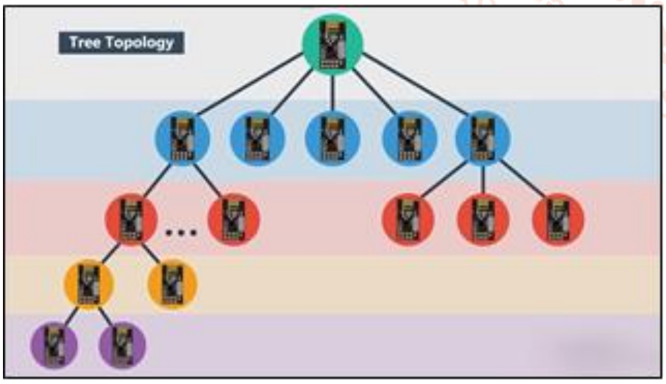

Figure 1: NRF24L01 Network

The NRF24L01 can actively listen to up to six other NRF24L01 modules in either transmitter or receiver mode using its six pipes. In our project, there is only one receiver and multiple transmitters, with all nodes except the root configured as transmitters. Therefore, the root node can actively listen to up to six transmitters, and each of these transmitters can listen to up to five other transmitters further down the tree. The level is limited to 5 due to the 15-bit addressing. Consequently, there can be a total of $5^5 = 3{,}125$ modules, with 1 being the receiver and the remaining 3,124 being transmitters. An individual NRF24L01 has a range of 1000 meters in open space therefore with a level of 5, we can have these 3,125 modules spread over a radius of 8 kilometers.

4.3. Weather Tracker Transmitter (WTT)

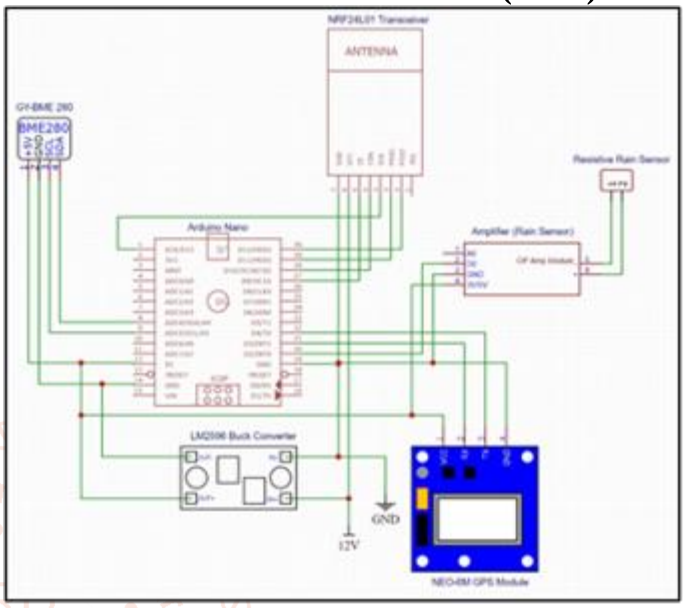

Figure 2.1: Weather Tracker Transmitter (WTT) module circuit schematic (Courtesy: EasyEDA)

Algorithm (WTT Module):

Step 1: Initialise the BME280 sensor and establish the I2C connection at address 0x76.

Step 2: Initialise the NEO-6M GPS modules and wait till Satellites are visible [if we do not find any satellites, we proceed to the next step by trying to establish a Satellite connections.]

Step 3: Read relative humidity, temperature and air pressure from the BME280 sensor using the I2C protocol. Read the precipitation status using the rain sensor. Read latitude, longitude and satellite count from the NEO-6M module using Serial protocol (baud rate 9600)

Step 4: Calculate altitude using the current air pressure and pressure at sea level.

Step 5: Create a data packet and store all the above data.

Step 6: Transmit the data packet to the NRF24L01 module using the SPI protocol.

Step 7: After one successful transmission of data, repeat from step 3.

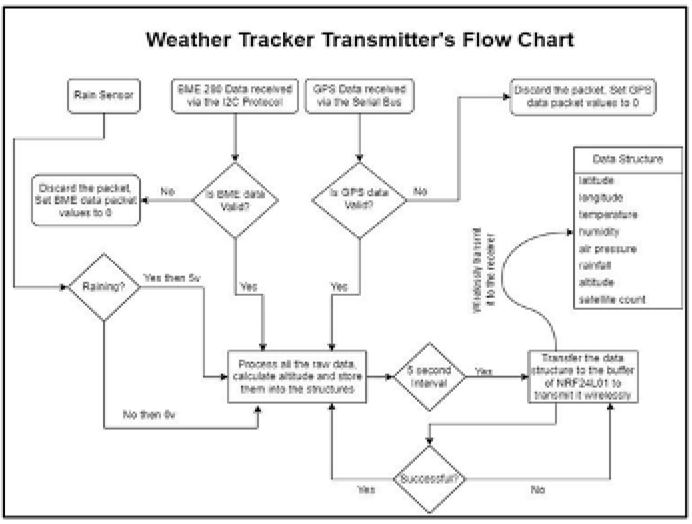

Figure 2.2: Weather Tracker Transmitter (WTT) flow chart (Courtesy: Draw.io)

4.4. Altitude Calculation using the Barometric Formula

We can use the barometric formula to calculate altitude using the current air pressure. The formula relates the pressure at a given altitude to the pressure at sea level, the temperature, and other constants. A simplified version of the barometric formula for calculating altitude in meters is

$h = L/T0 * (1 - (P0/P)^{R*L/g*M})$

h is the altitude in meters

T0 is the standard temperature at sea level in Kelvin, typically 288.15 K

L is the temperature lapse rate in the atmosphere, typically 0.0065 K/m

P is current air pressure, in Pascal

P0 is standard sea level pressure, 101325 Pascals

R is universal gas constant, 8.3144598 J/(mol·K)

g is the acceleration due to gravity, 9.80665 m/s²

M is molar mass of Earth's air, 0.0289644 kg/mol

4.5. Weather Tracker Receiver (WTR)

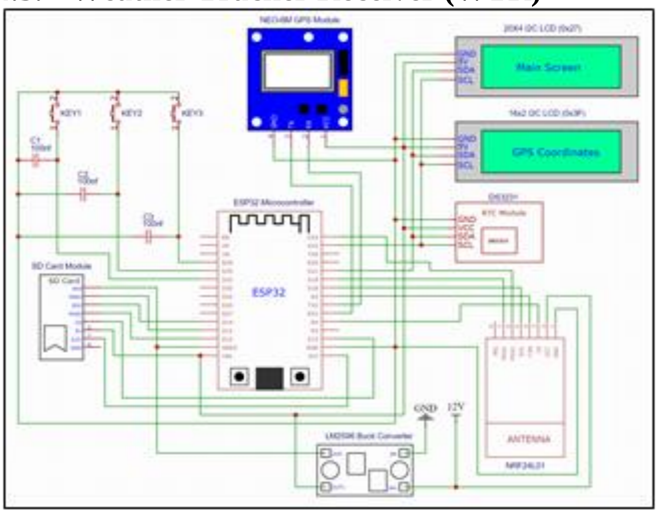

Figure 3.1: Weather Tracker Receiver circuit schematic (Courtesy: EasyEDA)

Algorithm (WTR Module):

Step 1: Initialise the NEO-6M GPS modules and wait till Satellites are visible [if we do not find any satellites, we proceed to the next step by trying to establish a Satellite connections.]

Step 2: Initialise the LCD modules

Step 3: Initialise the SD card module and if there is no 'data.csv' file present, create one.

Step 4: Establish a connection to the WiFi router using the last saved SSID and Password from the EEPROM

Step 5: If the connection to the WiFi is not successful, then host an access point for the user to connect to a new WiFi router or available networks for the Internet.

Step 6: Establish a connection to the Blynk Server [if we are unable to do so after a few tries, we proceed to the next step but we keep trying to establish the connection afterwards too]

Step 7: Listen to the radio for incoming data packets.

Step 8: When data packets are received, process them and check if they are valid or not.

Step 9: If data packets are corrupted, go to step 4.

Step 10: Calculate the distance between the latitude and longitudes of the transmitter and the receiver's latitude and longitude using the Haversine formula if GPS data is valid.

Step 11: Display all the received data on the LCD screens.

Step 12: If it is time to log data into the SD card, then send this data packet to the SD card module using SPI protocol and append the data to the 'data.csv' file

Step 13: Send all these data to the Blynk Server if the connection is established.

Step 14: Look for any button presses. If found then display the settings menu to allow the user to configure the device.

Step 15: Routine check to ensure all modules are still connected to the Receiver and are working. Go to step 7.

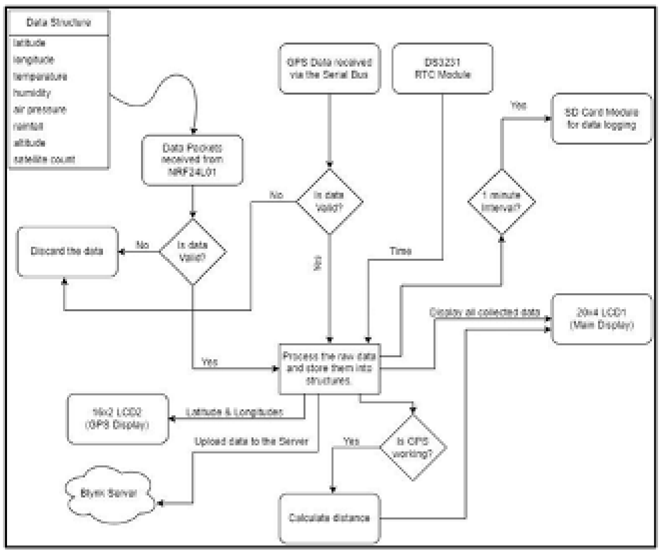

Figure 3.2: Weather Tracker Receiver flow chart (Courtesy: Draw.io)

4.6. Haversine Formula for Distance Calculation
The Haversine formula is used to calculate the great-circle distance between two points on the surface of a sphere given their latitudes and longitudes. This formula is beneficial for calculating distances on the Earth. The formula is:

$$a = \sin^2(\Delta\phi/2) + \cos(\phi1) * \cos(\phi2) * \sin^2(\Delta\lambda/2)$$

$$c = 2 * \text{atan2}\ (\sqrt{a}, \sqrt{(1-a)})$$

$$d = R * c$$

$\Phi1$ and $\Phi2$ are the latitudes of the two points in radians.

$\lambda1$ and $\lambda2$ are the longitudes of the two points in radians.

$\Delta\phi = \phi2 - \phi1$ is the difference in latitude.

$\Delta\lambda = \lambda2 - \lambda1$ is the difference in longitude.

R is the Earth's radius (mean radius = 6,371 km).

d is the distance.

4.7. Transmitter's enclosure
Dimension: 104mm x 104mm x 33mm

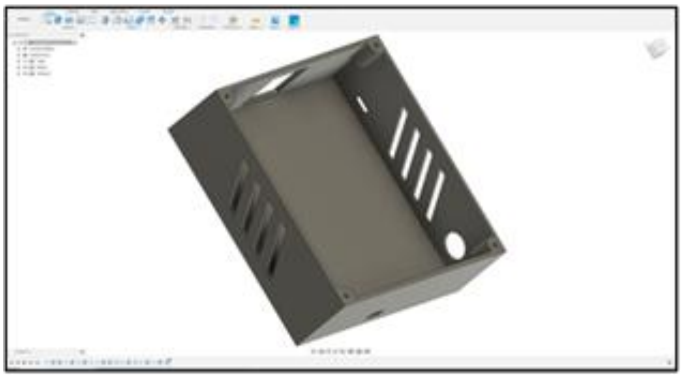

Figure 4.1: Transmitter Enclosure (Body) [CAD software: Fusion 360 (Student Version)]

Figure 4.2: Transmitter Enclosure (Body) after 3D Printing [3D Printer used: Ender 3 S1 Pro]

Dimension: 104mm x 104mm x 2mm

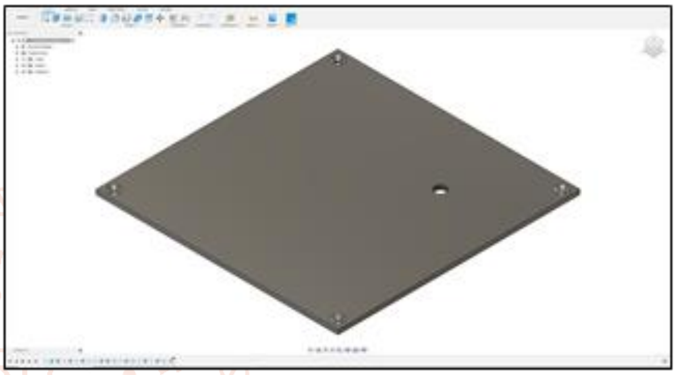

Figure 5.1: Transmitter Enclosure (Body) [CAD software: Fusion 360 (Student Version)]

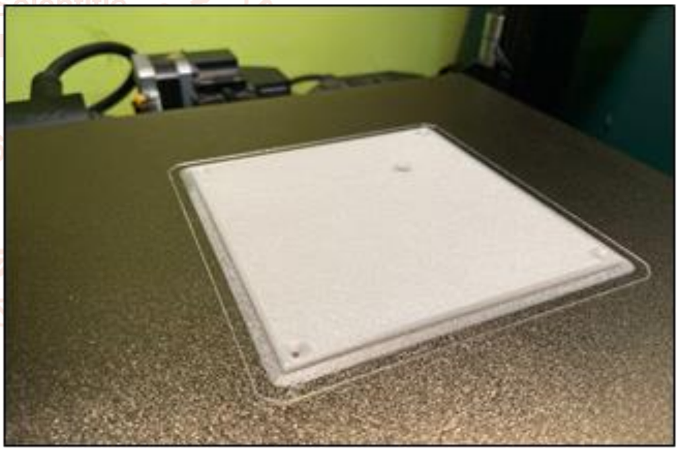

Figure 5.2: Transmitter Enclosure (Lid) after 3D Printing [3D Printer used: Ender 3 S1 Pro]

4.8. Receiver's enclosure
Dimension: 150mm x 110mm x 42.2mm

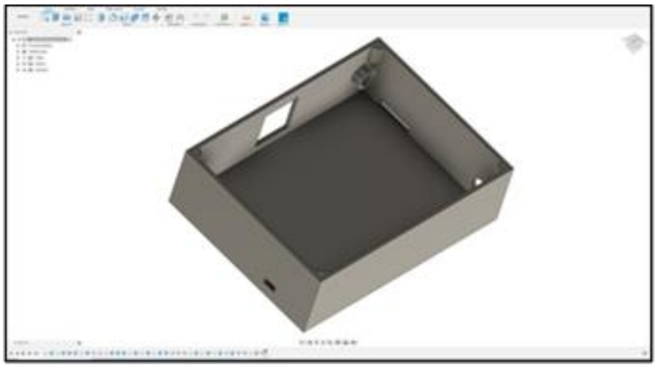

Figure 6.1: Receiver Enclosure (Body) [CAD software: Fusion 360 (Student Version)]

Figure 6.2: Receiver Enclosure (Body) after 3D Printing [3D printer used: Ender 3 S1 Pro]

Dimension: 150mm x 110mm x 2mm

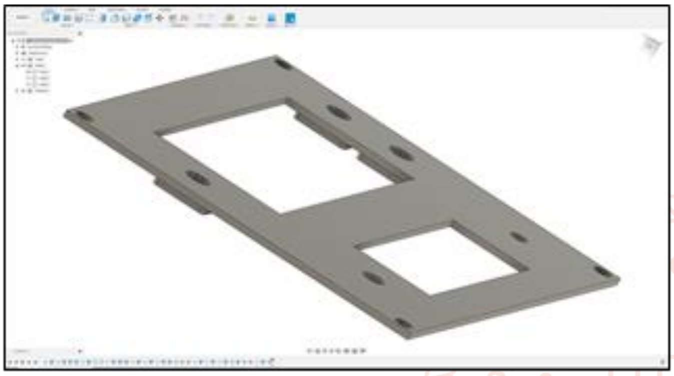

Figure 7.1: Receiver Enclosure (Lid) [CAD software: Fusion 360 (Student Version)]

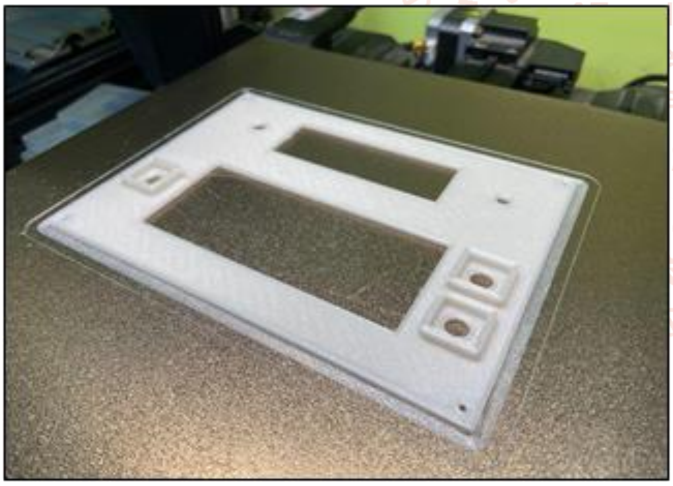

Figure 7.2: Receiver Enclosure (Lid) after 3D Printing [3D Printer used: Ender 3 S1 Pro]

5. Applications

5.1. Weather Prediction

The collected data can be used for predicting weather conditions, such as precipitation, by utilising historical data. Numerous machine learning algorithms can be employed to achieve this. These predictions can be utilised for weather forecasting or as part of a home automation system to activate an alarm when it is about to rain. Installing additional sensors like a UV index sensor and instruments such as an anemometer can aid in gathering more weather data for prediction purposes. Moreover, the data can be advantageous in fields like agriculture, automating crop irrigation and providing farmers with user-friendly weather updates. The transmitter device can even be installed into hot-air balloons to collect weather data, with the logged CSV file subsequently used for weather analysis.

5.2. Data Logger for Remote Exploration Robots

The transmitter module is designed to support multiple sensor modules, which can be connected using a simple one-wire protocol, I2C, SPI, or a serial bus. This makes adding more sensors to the device straightforward. Thus, the transmitter can be used inside a robot that explores environments inaccessible to humans, collecting data and logging it onto an SD card for future analysis.

5.3. Telemetry for drones and rockets

The transmitter can also be configured as a telemetry device, transmitting critical data from a drone or rocket to a base station. This includes parameters such as temperature, altitude, and more. By integrating this device into the vehicle, it can transmit data while in motion. The GPS functionality helps track the vehicle's location and velocity, ensuring comprehensive monitoring.

5.4. Factories for monitoring different stages of manufacturing

The transmitter can accurately measure temperature, humidity, and air pressure—three critical parameters for maintaining optimal conditions in factories to ensure product quality. This device can be installed in multiple locations both inside and outside the factory. The collected data is sent to a receiver, logged onto an SD card, and uploaded to a server. Automated tasks, such as controlling the air conditioner, humidifier, or heater, can be performed based on this data to maintain a consistent environment, regardless of external conditions.

6. Result and Discussions

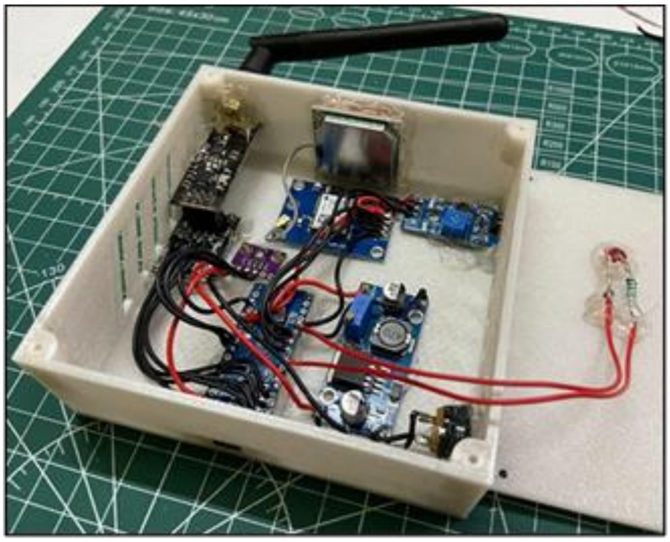

Figure 8.1: The Transmitter Circuit built into the 3D Printed enclosure

Figure 8.2: The Transmitter (WTT) after assembly

Figure 9.1: The Receiver Circuit built into the 3D Printed enclosure

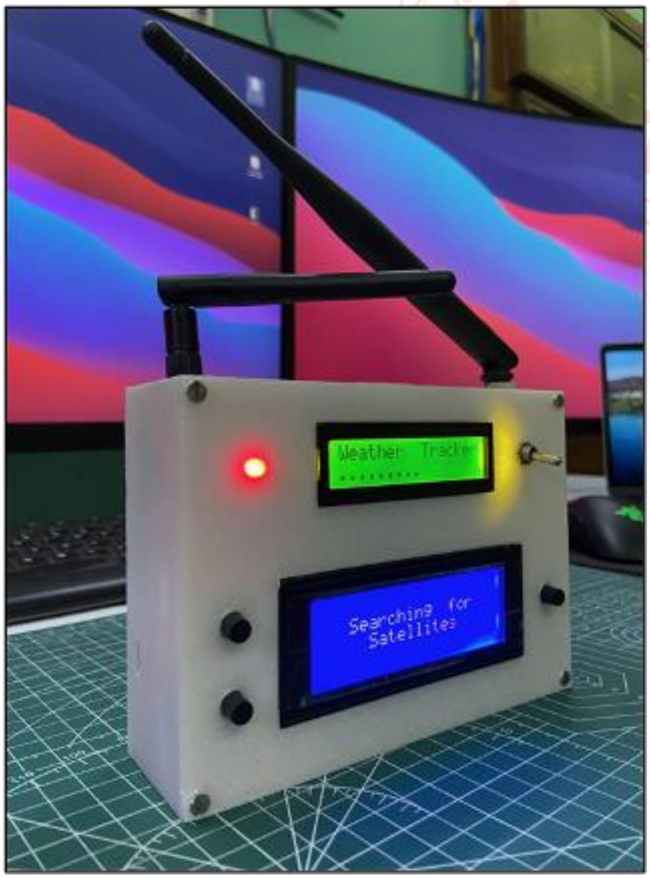

Figure 9.2: The Receiver Circuit (WTR) after assembly

Figure 10.1: The Transmitter Device (WTT)

In Figure 10.1, the transmitter device is turned on, and the red light indicates that it is searching for satellites. When enough satellites are found to provide valid GPS data, the red light will turn off. The temperature, humidity, air pressure, altitude, and precipitation data are not transmitted via radio for the first 27 seconds, as the device waits for the GPS connection to be established. Once the connection is established, or after 27 seconds have passed, the radio module is activated, and the data are sent wirelessly to the receiver module. The PCB connected to the transmitter device is the precipitation sensor. When it rains, water droplets on the PCB short the copper traces, causing a voltage change. This change is amplified by the operational amplifier module and detected as rainfall.

Figure 10.2: The Transmitter Device after establishing a successful GPS connection

Figure 11.1: The Receiver Device (WTR)

In Figure 10, the receiver device is switched on and begins searching for available satellites. It will wait for 27 seconds, as the cold start time for the NE0-6M module is 27 seconds. If sufficient satellites are found within this period to provide valid GPS data, the device processes the data and waits for radio signals from the transmitter modules (WTT). The red light on the device indicates valid GPS data. Once the device acquires enough GPS data, the red light turns off.

Figure 11.2: The Receiver Device after establishing a successful connection with the Transmitter and GPS

The models were placed on the roof of a building to collect all the data. The transmitter device (WTT) and the receiver device (WTR) were positioned approximately 2 metres apart. In Figure 11.2

‘Lat’ means Latitude of the Transmitter Device

‘Lng’ means Longitude of the Transmitter Device

‘T’ means the temperature in Celcius

‘H’ means Relative Humidity in %

‘A’ means altitude above sea level

‘P’ means air pressure in hectopascals

‘D’ means the distance between the Transmitter and Receiver devices.

Figure 12: The Receiver device when there is rainfall.

In Figure 12, when it rains, the receiver device replaces the distance information with the text "Raining." Measuring rainfall in millimetres is not possible with basic sensors, so in the CSV file, 0.0 is recorded when there is no rainfall, and 1.1 is recorded during rainfall in the precipitation column.

When the receiver module detects that the transmitter device is sending data packets via radio and verifies their validity, only then is the data recorded onto the SD card in a CSV file. If the transmitter module is unable to establish a GPS connection even after the 27-second delay for a cold start, the latitude, longitude, and distance are recorded as 0.0, 0.0, and -1, respectively. If only the receiver device is unable to establish a GPS connection, the distance is recorded as -1, indicating it is invalid.

Figure 13.1: The Transmitter (WTT) and Receiver (WTR) module set at a distance of approximately 2 meters

Figure 13.2: The Transmitter (WTT) and Receiver (WTR) module set at a distance of approximately 2 meters

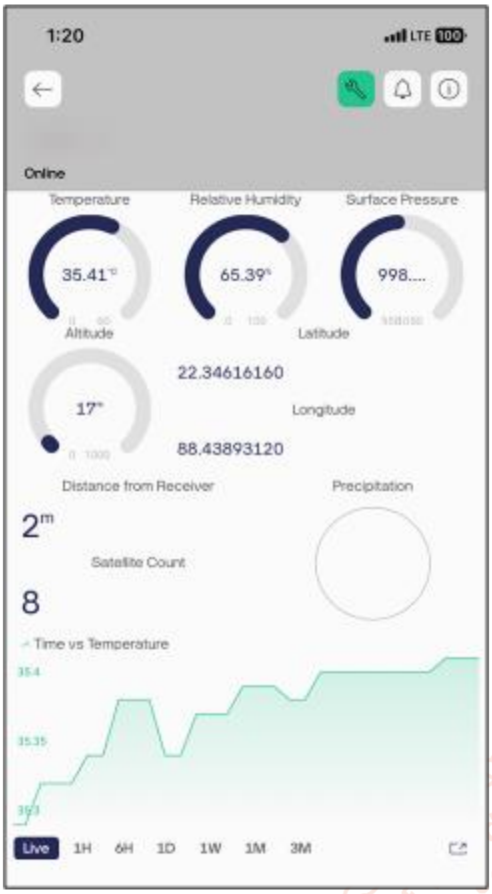

Figure 14.1: The Mobile Application Dashboard when it's not raining

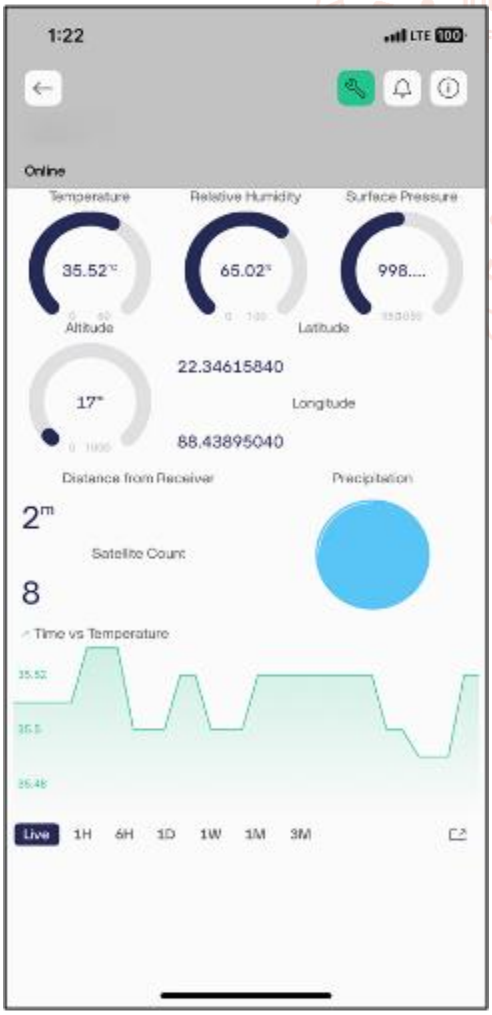

Figure 14.2: The Mobile Application Dashboard when it's raining

Figure 14.3: The Web Dashboard

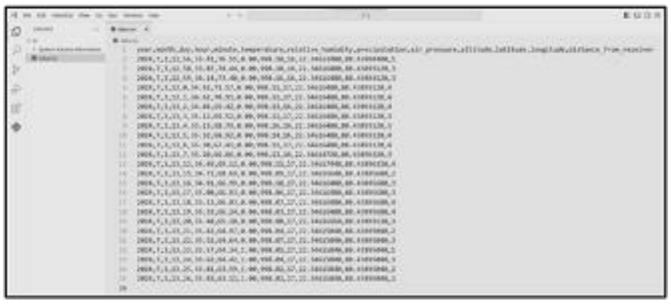

Figure 15: The 'data.csv' file recorded in the SD card during the measurements taken above

Discussions:

The Weather Tracker project can collect data from multiple geographic locations and report all the collected data to a single receiver device for logging into a CSV file. This approach provides the advantage of more accurate weather predictions, as it considers data from a larger area rather than a single point. In other references [1, 3, 5, 7, 9, 12], the proposed systems are capable of collecting data from only one device in one location. Having only one data-collecting device, with the display and SD card module integrated into the same device, requires the operator to physically access the device, making the installation inefficient. In this project, multiple tracker devices can be installed on towers and rooftops, eliminating the need for someone to be physically present near the transmitter devices. Additionally, implementing GPS allows the user to understand the collected data better, as the latitude and longitude can later be used to pinpoint the exact locations from which the data were collected. The LCD offers a 'Menu' option that allows the user to change various settings and configure the device. The receiver device can also connect to any available WiFi signals with known passwords. This project proposes the cheapest and most efficient way of collecting weather and other environmental data.

The transmitter device uses the ATmega328p microcontroller, which represents data in 8-bit. Therefore, the heavy calculations that the MCU needs to perform could yield better results if floating point numbers were represented with more bits.

Additionally, the antenna used is omnidirectional, which is suitable for sending radio waves in a straight linear path from the transmitter to the receiver.

However, if the transmitter devices are installed in regions where the transmitter antenna does not directly point to the receiver, connection issues can occur. This can be resolved by using a skew planar or a cloverleaf antenna. The transmitter device is powered by a 12V adapter, which is not ideal and can be replaced with a solar rechargeable battery system.

Other models that could be used to achieve the same results include satellite models, weather stations, radiosondes, Doppler radar, buoys and ocean-based sensors, automated surface observing systems, and mobile weather stations. These models are used in the modern world and yield accurate results, but they come with a very high cost of installation and maintenance.

7. Conclusion and Future Scope

The project aimed to develop a low-cost, high-efficiency system combining multiple transmitters with a single receiver. Up to 3,124 transmitters can send weather and environmental data to the receiver. The receiver logs this data onto an SD card and uploads it to the Blynk server for visualisation in web and mobile applications. Additionally, the receiver displays the collected data on LCD screens in a user-friendly manner. The goal was to create a cost-effective, accurate data-logging system with smart and innovative features. All data has been collected and processed in the Baruipur region. This device not only collects data but also ensures its validity, discarding any erroneous readings. The data is valuable to meteorologists, farmers (for predictive analysis), factories, and others interested in automation. The system's accuracy closely matches real-world data.

Future upgrades for the system include several enhancements to improve performance and reliability. Firstly, designing a weatherproof enclosure for both transmitter and receiver devices will prevent inaccurate readings from the BME280 sensor caused by direct sunlight. Upgrading the display to OLED screens can provide a more visually appealing user experience and reduce power consumption by using a single, properly sized OLED display. Replacing the current GPS modules with more reliable models like the NEO-8M will improve accuracy. For precise altitude calculations, consistent monitoring of sea level pressure is essential, and using a microcontroller with higher bit resolution can enhance calculation accuracy by representing more decimal points. Extending the range and increasing the number of transmitters can be achieved with transceivers like the LoRa module. Upgrading the microcontroller to more powerful models such as the STM32 or Raspberry Pi will enhance speed and performance. Additionally, integrating machine learning capabilities by modifying the receiver with a Raspberry Pi or similar device will enable autonomous data processing and weather prediction. Lastly, developing PCBs for both transmitter and receiver circuits will ensure secure soldering of components, reducing the risk of loose connections, short circuits, or data loss, thus enhancing overall system reliability and longevity.

References

[1] Sharma, P., & Prakash, S. (2021). Real-time weather monitoring system using IoT. In ITM web of conferences (Vol. 40). EDP Sciences.

[2] Subhadra, A., Ganesh, R., Maheshbabu, K., Sandeep, K. S., & Mahesh Kumar, J. (2020). IoT Based Real-Time Weather Monitoring System. Int J Eng Appl Sci Technol., 4, 384-92.

[3] Sivakumar, B., & Nanjundaswamy, C. (2021). Weather monitoring and forecasting system using IoT. Global Journal of Engineering and Technology Advances, 8(2), 008-016.

[4] Murthy, M. S., Kumar, R. R., Saikiran, B., Nagaraj, I., & Annavarapu, T. (2023). Real-Time Weather Monitoring System using IoT. In E3S Web of Conferences (Vol. 391, p. 01142). EDP Sciences.

[5] Ukhurebor, K., Abiodun, I., Azi, S., Otete, I., & Obogai, L. (2017). A cost-effective weather monitoring device. Archives of Current Research International, 7(4), 1-9.

[6] Mudgil, M. S. (2023). IoT-Based Smart Weather Monitoring System.

[7] Rao, B. S., Rao, K. S., & Ome, N. (2016). Internet of Things (IoT) based weather monitoring system. International Journal of advanced research in Computer and communication engineering, 5(9), 312-319.

[8] Bhagat, A. M., Thakare, A. G., Molke, K. A., Muneshwar, N. S., & Choudhary, P. V. (2019). IoT-based weather monitoring and reporting system project. International Journal of Trend in Scientific Research and Development (IJTSRD), 3(3), 1-3.

[9] OGUNBUNMI, S., TAIWO, A. A., OLADOSU, J. B., SANUSI, H., INAOLAJI, F. A., OLASUNKANMI, U. G., ... & ENABULELE, E. C. (2024). Internet of Things weather monitoring system. World Journal of Advanced Research and Reviews, 22(2), 2099-2110.

[10] Liyakat, K. S. S., & Liyakat, K. K. S. (2023). IoT Based Arduino-Powered Weather

Monitoring System. Journal of Telecommunication Study, 8(3), 25-31.

[11] Kodali, R. K., & Sahu, A. (2016, December). An IoT-based weather information prototype using WeMos. In 2016, the 2nd International Conference on Contemporary Computing and Informatics (IC3I) (pp. 612-616). IEEE.

[12] Tsao, Y. C., Te Tsai, Y., Kuo, Y. W., & Hwang, C. (2019, October). An Implementation of IoT-Based Weather Monitoring System. In the 2019 IEEE International Conferences on Ubiquitous Computing & Communications (IUCC) and Data Science and Computational Intelligence (DSCI) and Smart Computing, Networking and Services (SmartCNS) (pp. 648-652). IEEE.

[13] Prasanna, M., Iyapparaja, M., Vinothkumar, M., Ramamurthy, B., & Manivannan, S. S. (2019). An Intelligent Weather Monitoring System using the Internet of Things. International Journal of Recent Technology and Engineering (IJRTE) ISSN, 2277-3878.

[14] Thombare, V. K., Khan, V. I., Kapale, G. R., Patil, S. D., Koli, P. S., & Jadhav, S. R. (2021). Solar Operated Weather Forecasting Station. Asian Journal For Convergence In Technology (AJCT) ISSN-2350-1146, 7(1), 197-201.

[15] Olanrele, O., Adeaga, O. A., Adeyemi, O. A., Ajayi, O. K., & Mowemi, A. O. (2022). An IoT Based Weather Station using an Embedded System. Quantum Journal of Engineering, Science and Technology (QJOEST), 3(3), 31-40.

[16] Rajesh, M. (2021). IoT-Based Solar Panel Tracking System with Weather Monitoring System