WEATHER MONITORING SYSTEM

Nithya Sharma 19MIA1028
Tharani Kumar 19MIA1033
Parvathy AJ 19MIA1048

1 Abstract

Weather is an undeniably important part of human life, and it can affect situations ranging from the day-to-day lives of people to the functioning of large-scale industries. A system to monitor the weather would prove to be greatly helpful in order to gain insight into current weather patterns and ensure preparedness for handling them. In this project, a weather monitoring system based on IoT components is built to collect weather data with the help of BMP180 (barometric power and altitude sensor), LM393 (raindrop sensor), DHT22 (humidity and temperature sensor) and HW103 (moisture sensor). The implementation of the system is done on Raspberry Pi with a program deployed in Python. The system outputs whether it is raining and the values of the current temperature, humidity, altitude and pressure.

2 Keywords

Raspberry Pi, Temperature, Weather, Sensors

3 Introduction

Weather monitoring plays a meaningful part in the functioning of our world on a daily basis, from helping a person choose their clothes for the day, to helping big companies plan out their activities. Protecting people's lives and property while enhancing their health, safety, and economic success are the ultimate goals of weather monitoring. It is the first step in weather forecasting, which can be used to alert and prepare individuals for upcoming weather events and climatic conditions.

An IoT based weather monitoring system would provide simple access from anywhere in the globe to real-time local weather monitoring. It would help in the short, as well as long-term archiving of meteorological and environmental data in order to research changes in weather patterns and comprehend how locally produced climate change has impacted weather. This also helps to provide an efficient setup for monitoring local atmospheric variables and microclimates to predict and forecast the weather.

4 Literature Survey

Andriy Holovatyy in their paper, "Development of IoT Weather Monitoring System Based on Arduino and ESP8266 Wi-Fi Module" developed an IoT system for observing and detecting changes in the weather with Arduino Mega2560 as the microcontroller device. Digital pressure, temperature and humidity sensors were used for the monitoring process and Proteus VSM was used for creating the electronic circuit and the model. Software modules were used for communication between the BME280 sensor and Wi-Fi module ESP-01 and with the IoT platform ThingsBoard and Mosquitto MQTT were created. A dashboard was built using Node-RED for visualization of the data and simulation of the decise was done in Proteus ISIS.

André M. de Oliveira, Marcelo A. Sudo, Sérgio R. Barros Dos Santos and Sidney N.Givigi, in their paper "Design and Analysis of a Low-Cost Weather Monitoring System based on Standard IoT Data Protocols" tested the performance of IoT protocols such as CoAP, HTTP REST and MQTT for a weather monitoring system built on Arduino UNO with Node MCU ESP8266. Several sensors were used for the monitoring of factors such as altitude, atmospheric pressure, humidity, luminosity, presence of flammable substance, smoke, temperature and toxic gasses. The measures from these sensors were input into a JAVA application which stresses them by varying the length of the message and time interval between requests and the resulting data was stored in a MySQL database. MQTT was found to transmit messages faster and completely, irrespective of size, making it more reliable and more appropriate.

Bidyadhar Subudhi and Debashish Mohapatra in their paper, "Development of a Cost-Effective IoT-Based Weather Monitoring System" used open-source technologies to develop and deploy an IoT system to for the monitoring and archival of weather data. The weather factors involved in this system were temperature, humidity, atmospheric pressure and dust particles and suitable sensors were used for their monitoring, in a residential area. The data was sent from the IoT devices to a remote VPS and collected and logged into a data base by a server application. MQTT was used for setting up and securing the VPS and IoT servers. NodeMCU ESP8266 and Raspberry Pi Zero W were used for the implementation of the system. The weather monitoring system was found to be scalable from scratch, time and cost efficient and secure from privacy and safety concerns.

5 Objective

The goal of the project is to develop a system to monitor the weather and collect data. A hardware implementation will be built using various sensors to measure temperature, moisture, humidity, pressure and rainfall. The system will use Raspberry Pi for data collection and processing. The ultimate aim of the project is to derive meaningful insights from the data to detect, record and display the various weather parameters.

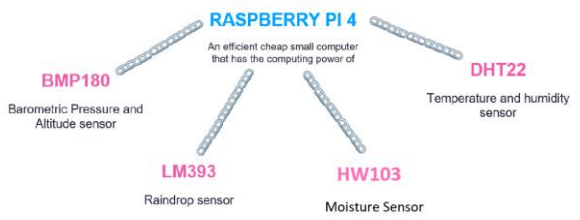
6 Motivation

Weather monitoring systems are essential in the field of agriculture, as crops are grown in large part due to temperature, humidity, and precipitation. Farmers used to make weather predictions based on observations of the sky, which was not always accurate. With the help of IoT based weather monitory, they can have access to precise weather forecasts due to the growth of meteorology, which uses supercomputers to gather data. Farmers may determine the ideal time for crops and agricultural operations by routinely reviewing weather forecasts. They can also better comprehend and monitor the growth status to make potentially expensive judgments.

Weather has a significant influence on traffic safety and functionality. Severe weather, such as snow, rain, or storms, might hinder driving ability. Additionally, there have apparently been several road accidents that were brought on by inclement weather. Weather monitoring can also aid in reducing energy consumption, flight durations, and delays for air travel, as well as ensuring passenger comfort and safety. The information supplied about hazardous weather conditions, such as high winds, thunderstorms, tornadoes, and ice, that might threaten an aircraft during take-off, landing, and during flight is of utmost significance to this business.

Apart from the above-mentioned applications, weather monitoring has a wide necessity in various areas, such as the planning of logistics, detection and control of forest fires, livestock protection initiatives and launch of clothing items, among others.

7 Design



- **Raspberry Pi 4:**

A Raspberry Pi can be used for a variety of purposes. A Raspberry Pi can be used as the brain of a robot, security system, Internet of Things device, or dedicated Android device. It can also be transformed into a retro arcade machine and utilised as a web server. The Raspberry Pi features a set of GPIO (general purpose input/output) pins that enable you to explore the Internet of Things and manage electronic components for physical computing in addition to being a reasonably priced Linux-running computer (IoT).

In our project, the Raspberry Pi has been utilised to run a system that consists of a raindrop sensor, temperature and humidity sensors, a barometric pressure and altitude sensor, and a long-range data transfer device.

- **BMP180 – Barometric Power and Altitude Sensor**

The barometer sensor shown above is a module that can measure atmospheric data. It can provide information on altitude, atmospheric pressure at sea level, and atmospheric pressure at ground level.

It has the following pins:

  - SDA – Serial data.
  - SCL – Serial clock.
  - Vcc – 3.3V.
  - GND – ground

- **LM393 – Raindrop Sensor**

Rain Sensor is made of two parts:

  - **The Sensor Pad:** It is made up of exposed copper particles on both sides. It performs as a variable resistor. Its resistance rating is influenced by the amount of water that falls on it. Water content is inversely correlated with that resistance value. The conductivity of the sensor surface increases as more water flows over it. After that, the resistance drops. The little water that does flow across this surface has a poor conductivity, which further increases resistance. This idea allows us to get an output voltage.

- **The Module:** This needs to be attached to the sensor pad that was previously shown. Therefore, using this module, we can obtain the sensor pad's output voltage as an analogue value. Additionally, the analogue value is converted into a digital value in this instance using the LM393 High Precision Comparator. With this potentiometer, you may also modify the sensitivity of the digital value.

  This module displays six PINs. The two pins mentioned above should be used to connect the sensor pads (+) and (-). The following describes the other four pins.

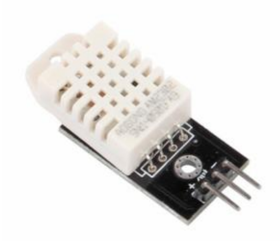
  VCC – This pin must have a potential between 3.3v – 5v.

  GND – This should attach to the cathode terminal.

  AO – This pin can be used to get the rainfall value as an analog value.

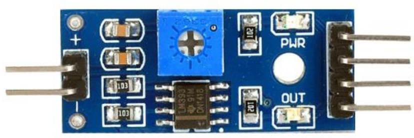
  DO – This pin can be used as a digital value for rainfall.

- **DHT22 – Temperature and humidity sensor**



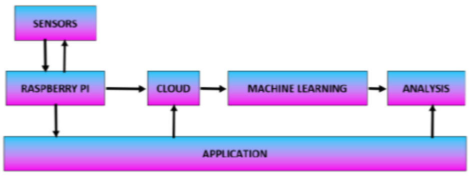
The dht22 is a simple, low-cost digital temperature and humidity sensor (rht03). It generates a digital signal on the data pin by employing a capacitive humidity sensor and a thermistor to measure the air's humidity (no analogue input pins needed). Although relatively simple to use, data collection calls for exact timing. Additionally, the DHT11 humidity range is from 20 to 80% with 5% accuracy, but the DHT22 sensor's humidity measuring range is from 0 to 100% with 2-5% accuracy.

- **HW103 – Moisture Sensor**



This sensor calculates the moisture level by calculating the volumetric content of water. The sensor can be utilised in analogue or digital mode because it has an analogue and a digital output. Two probes make up a moisture sensor, which measures the volumetric content of water. The current can flow via the two probes, and after that, the resistance value is obtained to calculate the moisture content.

8 Architecture



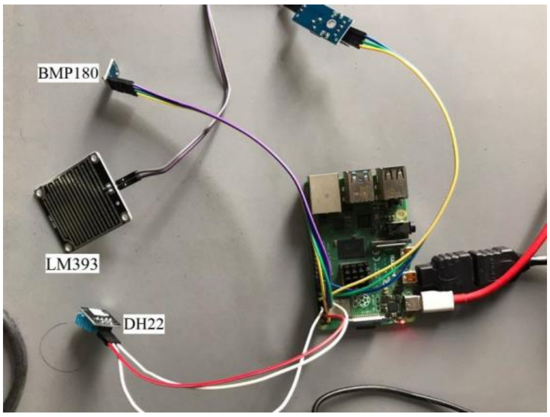
Weather conditions can be tracked and updated online using this technology. The system keeps track of three variables: temperature, humidity, and rainfall. The values are then shown on the PC's screen. The user can get a good understanding of the weather in a certain location by looking at measurements on the display. The localised area's weather can be determined with the use of this system. A raindrop sensor, temperature and humidity sensors, a barometric pressure and altitude sensor, and a Raspberry Pi is used to power the system. The system connects to the website through WiFi after being turned on. Five parameters, including temperature, humidity, pressure, altitude, and rainfall, are tracked by the system. When the area is dry, the system to monitor and update the status through the IOT shows zero value when the weather changes. The system displays the value of the increase in rainfall when a raindrop is detected. The value is updated as the temperature rises. The user can check the weather conditions in a certain area from any distance.

9 Implementation

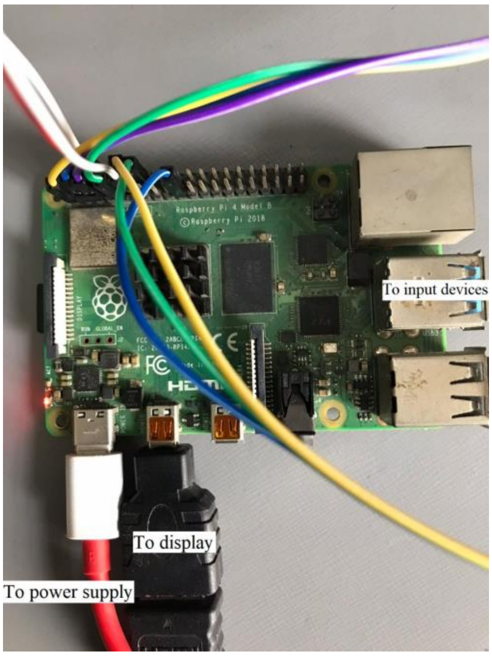
Raspberry Pi 4 was used for the implementation of the weather monitoring system. The sensors involved in the monitoring process are as follows:

BMP180	Barometric Power and Altitude Sensor	Measures atmospheric pressure
LM393	Raindrop Sensor	Detects precipitation
DHT22	Temperature and Humidity Sensor	Measures temperature and humidity levels
HW103	Moisture Sensor	Detects moisture content

The connections were made between the Raspberry Pi board and the sensors as depicted below:



The BMP180 and DH22 sensors are directly connected to the Raspberry Pi board. The LM393 sensor is connected to the board via the HW103 sensor.



The Raspberry Pi microcontroller was connected to a power supply to power the system by means of a USB-C cable. It was connected to a monitor to display the contents and the output using a HDMI cable. Input devices, keyboard and mouse were connected to the microcontroller via the USB ports as well to navigate the contects on the display screen.

The software implementation of the system was done on Python in the Raspberry Pi environment, using Wi-Fi to connect to the website.

The necessary libraries are imported for deploying the system. The 'time' module is used for various representations of the time in code, measuring time periods, waiting during execution of the code and efficiency measurement of the code. The 'gpiozero' module allows us to read input devices in a variety of ways.

```
from time import sleep
from gpiozero import Buzzer, InputDevice
```

The following code is used to generate a function which takes in the input from the LM393 sensor via the HW103 sensor. If a raindrop (moisture content) is detected by the sensor, a message will be displayed saying, "It's raining - get the washing in!"

```
no_rain = InputDevice(18)

def buzz_now(iterations):
    for x in range(iterations):
        buzz.on()
        sleep(0.1)
        buzz.off()
        sleep(0.1)

while True:
    if not no_rain.is_active:
        print("It's raining - get the washing in!")
        buzz_now(5)
    sleep(1)
```

The following code is used to take in the input from the BMP180 sensor. The atmospheric pressure, temperature and altitude measured by the sensor are printed.

```
bmp = BMP180(0x77,"/dev/i2c-1")

for x in range(0,5):
        print("temperature: %s" % bmp.temperature())
        print("pressure: %s" % bmp.pressure())
        print("altitude: %s" % bmp.altitude())
        sleep(2)
```

The The following code is used to take in the input from the DHT22 sensor. The temperature and humidity measurements detected by the sensor are printed and then the system waits for 5 seconds before reading the values again to detect if there are any changes in the values.

```
#Set DATA pin
DHT = 4
while True:
    #Read Temp and Hum from DHT22
    h,t = dht.read_retry(dht.DHT22, DHT)
    #Print Temperature and Humidity on Shell window
    print('Temp={0:0.1f}*C  Humidity={1:0.1f}%'.format(t,h))
    sleep(5) #Wait 5 seconds and read again
```

The code was executed in the Linux terminal and the results were obtained as follows.

```
rasp_pi4/admin> python weather.py
It's raining - get the washing in!
temperature: 28
pressure: 1012
altitude: 6.5
Temp=28*C  Humidity=72%
```

During the execution, water was sprinkled on the raindrop sensor (LM393) to test its functioning and thus the code returned the message announcing that it is raining. The temperature, pressure, altitude and humidity levels detected by the barometric power and altitude sensor (BMP180) and temperature and humidity and temperature sensor (DHT22) were also printed on the display screen.

10 Conclusion

The weather monitoring system was built with the help of Raspberry Pi, temperature and humidity sensor, raindrop sensor, moisture sensor and barometric pressure sensor. The code required to get the result/output was done with the help of Python. The weather monitoring system was able to sense 5 parameters, raindrop sensor, a temperature and humidity sensor, a Barometric Power and Altitude Sensor as well as print all the values on the screen.

11 Future Scope

As for the future scope of this project, the weather monitoring system can be deployed in the cloud for increased accessibility of the data. For instance, a google cloud SQL can be created to allocate and save all the weather variables collected by the sensors as it is easy to set up, easy to run and with secure built in features. A dashboard can be created from the data with the use of Web based APIs to provide a user interface for the system. The data collected from the sensors can be displayed using various methods such as table, graph and chart in Google Data Studio as it supports monitoring, logging, and diagnostics of weather conditions, and data analysis which can be ac-cessed by both administrator and users through web page. The functioning and complexities of the weather monitoring system can also be improved by including various sensors for the measurement of other weather factors such as wind speed, air quality index, etc.

12 References

- Holovatyy, A. (2021). Development of IOT weather monitoring system based on Arduino and ESP8266 Wi-Fi Module. IOP Conference Series: Materials Science and Engineering, 1016(1), 012014. https://doi.org/10.1088/1757-899x/1016/1/012014
- Sudo, M. A., Santos, S. R., Oliveira, A. M., & Givigi, S. N. (2022). Design and analysis of a low-cost weather monitoring system based on standard IOT data protocols. 2022 IEEE International Systems Conference (SysCon). https://doi.org/10.1109/syscon53536.2022.9773845
- Mohapatra, D., & Subudhi, B. (2022). Development of a cost-effective IOT-based weather monitoring system. IEEE Consumer Electronics Magazine, 11(5), 81–86. https://doi.org/10.1109/mce.2021.3136833
- Islam, M. M. (2019). Weather Monitoring System using internet of things. Trends in Technical & Scientific Research, 3(3). https://doi.org/10.19080/ttsr.2019.03.555615
- Sudo, M. A., Santos, S. R., Oliveira, A. M., & Givigi, S. N. (2022). Design and analysis of a low-cost weather monitoring system based on standard IOT data protocols. 2022 IEEE International Systems Conference (SysCon). https://doi.org/10.1109/syscon53536.2022.9773845
- Huddleston, C. (2006). Sensor application—Temperature Sensor. Intelligent Sensor Design Using the Microchip DsPIC, 161–207. https://doi.org/10.1016/b978-075067755-4/50009-2

- Yuwono, T., Nahar, A., Ruzardi, & Aminudin, A. (2018). Design of IOT for weather monitoring using raspberry. AIP Conference Proceedings. https://doi.org/10.1063/1.5066939
- Dht22 – temperature and humidity sensor. Components101. (n.d.). Retrieved November 18, 2022, from https://components101.com/sensors/dht22-pinout-specs-datasheet
- Sharma, P., & Prakash, S. (2021). Real time weather monitoring system using IOT. ITM Web of Conferences, 40, 01006. https://doi.org/10.1051/itmconf/20214001006
- BMP180 - atmospheric pressure sensor. Components101. (n.d.). Retrieved November 18, 2022, from https://components101.com/sensors/bmp180-atmospheric-pressure-sensor