**ENVIRONMENTAL MONITORING**

## **PHASE 5-DOCUMENT SUBMISSION**

# **Abstract:**

# The project aims to establish an IoT-based environmental monitoring system in public parks, focusing on temperature and humidity data collection. The primary goal is to offer park visitors access to real-time environmental information via a public platform. This empowers park-goers to make informed decisions about their outdoor activities based on current weather conditions. The project involves the design and deployment of IoT sensors, the development of a user-friendly platform, and the integration of IoT technology. Python will be utilized for data abstraction and automation. Overall, the project seeks to enhance the park experience by providing data-driven insights for a more enjoyable and comfortable outdoor visit.

**Real-time environmental monitoring:**

# Real-time environment monitoring refers to the continuous and immediate collection, analysis, and reporting of data related to various environmental parameters. This can include factors such as air quality, temperature, humidity, pollution levels, weather conditions, and more. The key aspect of real-time monitoring is that the data is gathered and made available instantly, allowing for quick responses to changing environmental conditions and potential issues. This technology is often used in various applications, from weather forecasting and pollution control to industrial processes and smart city initiatives.

**Aiding park visitors in activity planning**:

# "Aiding park visitors in activity planning" involves providing assistance and information to individuals who are visiting a park or recreational area to help them make informed decisions about the activities they can engage in during their visit. This typically includes offering details about available activities, such as hiking, camping, picnicking, wildlife viewing, and more, along with information on trail conditions, safety guidelines, and any special events or attractions within the park. The goal is to enhance the visitor experience by helping them plan and enjoy their time in the park to the fullest extent possible.

**Promoting outdoor experiences:**

# Promoting outdoor experiences involves encouraging and facilitating activities that take place in natural environments, such as parks, forests, mountains, or other outdoor settings. This can include initiatives to raise awareness of the benefits of outdoor activities, the preservation of natural spaces, and the promotion of health and well-being through activities like hiking, camping, biking, birdwatching, and more. The aim is to inspire individuals to engage with the outdoors, appreciate nature, and experience the physical, mental, and emotional benefits of spending time in natural settings.

**Objective:**

# The primary objective of this project is to deploy IoT devices within public parks for the real-time monitoring of environmental conditions, with a particular focus on temperature and humidity. By achieving this objective, we aim to provide park visitors with immediate access to up-to-date environmental data through a publicly accessible platform. This accessibility empowers individuals to make informed decisions when planning their outdoor activities in the park, ultimately enhancing their overall park experience while ensuring their safety and comfort.

**Deployment of IOT Device**

# \*Identify Objectives\*: Determine the specific goals for deploying IoT sensors. These could include monitoring air quality, tracking visitor traffic, ensuring security, or optimizing irrigation systems.

# 2. \*Select Sensor Types\*: Choose the appropriate sensors for your objectives. For example, you might use environmental sensors for air quality, motion sensors for security, or people counters for visitor tracking.

# 3. \*Network Infrastructure\*: Set up a robust and scalable network infrastructure, such as Wi-Fi, LoRaWAN, or cellular, to connect the sensors to a central system.

# 4. \*Sensor Placement\*: Strategically install sensors in various locations within the park to capture relevant data. For example, air quality sensors should be positioned where pollution might be a concern.

# 5. \*Data Collection\*: Sensors should collect data at regular intervals and transmit it to a central server or cloud platform for analysis. Ensure data security and privacy measures are in place.

# 6. \*Data Analysis\*: Analyze the collected data to derive actionable insights. For instance, analyze visitor traffic patterns to optimize park layout or monitor air quality trends.

# 7. \*User Interfaces\*: Develop user-friendly dashboards and mobile apps for park administrators and visitors to access real-time data and reports.

# 8. \*Maintenance and Power\*: Regularly maintain sensors to ensure they are functioning correctly. Depending on the sensor type, power considerations are vital. Some sensors may be battery-operated, while others might require a continuous power source.

# 9. \*Security and Privacy\*: Implement robust security measures to protect the data and the IoT network. Be transparent about data collection and usage to address privacy concerns.

# 10. \*Scaling\*: As the park's needs evolve, be prepared to scale the IoT network and add more sensors or features.

# 11. \*Feedback and Improvement\*: Continuously gather feedback from park users and administrators to make improvements and refine the IoT system.

# 12. \*Regulatory Compliance\*: Ensure compliance with local regulations and data protection laws, especially when dealing with data collected from public spaces

**Environmental monitoring platform**

# 1\*Data Sources\*:

# - Identify and connect to various data sources such as weather stations, air quality monitors, water quality sensors, and other environmental data collection devices.

# - Integrate with APIs from government agencies, environmental organizations, and IoT devices.

# 2. \*Database\*:

# - Store and manage the incoming data in a secure and scalable database system.

# 3. \*Data Processing\*:

# - Implement real-time data processing to ensure the data displayed is up to date.

# - Data validation and quality control to filter out erroneous readings.

# 4. \*User Authentication and Authorization\*:

# - Develop user registration and login systems.

# - Implement role-based access control to distinguish between public users and administrators.

# 5. \*User Interface\*:

# - Create an intuitive and user-friendly web interface.

# - Display data in various formats (charts, graphs, maps) for easy interpretation.

# - Provide filtering options and search functionality for specific data points.

# 6. \*Real-Time Updates\*:

# - Implement WebSockets or Server-Sent Events to push real-time data updates to users.

# - Consider data visualization libraries like D3.js for interactive charts and maps.

# 7. \*Geospatial Integration\*:

# - Use mapping libraries (e.g., Leaflet or Google Maps) to display geospatial data.

# - Incorporate location-based filtering and alerts.

# 8. \*Mobile Responsiveness\*:

# - Ensure the platform is accessible and functional on various devices and screen sizes.

# 9. \*Notifications and Alerts\*:

# - Allow users to set up custom alerts for specific environmental conditions.

# - Send notifications via email, SMS, or in-app alerts.

# 10. \*Data Sharing\*:

# - Enable users to share specific data or insights on social media or via direct links.

# 11. \*Historical Data\*:

# - Provide access to historical data and allow users to view trends and patterns over time.

# 12. \*Privacy and Security\*:

# - Implement robust security measures to protect user data and maintain data integrity.

# - Comply with data protection regulations (e.g., GDPR).

# 13. \*Scalability and Performance\*:

# - Design the platform to handle a large number of concurrent users and data points.

# 14. \*Feedback and Support\*:

# - Include a feedback mechanism for users to report issues or suggest improvements.

# - Provide customer support options.

# 15. \*Legal Considerations\*:

# - Ensure compliance with legal requirements for data usage and sharing.

# 16. \*Monetization\* (if applicable):

# - Consider revenue models such as premium subscriptions, ads, or partnerships.

# 17. \*Testing and Quality Assurance\*:

# - Thoroughly test the platform for functionality, security, and performance.

# 18. \*Deployment and Hosting\*:

# - Choose a reliable hosting solution that can handle the expected traffic and scalability needs.

# 19. \*Documentation\*:

# - Provide clear and comprehensive documentation for users and developers.

# 20. \*Maintenance and Updates\*:

# - Plan for ongoing maintenance, updates, and feature enhancements.

Program:

import time

import datetime

import random # Replace with actual sensor library

# Function to read sensor data (replace with your sensor library)

def read\_sensor\_data():

temperature = random.uniform(15.0, 30.0) # Replace with actual temperature reading

humidity = random.uniform(30.0, 70.0) # Replace with actual humidity reading

return temperature, humidity

# Function to log data to a file

def log\_data(temperature, humidity):

timestamp = datetime.datetime.now()

log\_entry = f"{timestamp}: Temperature={temperature}°C, Humidity={humidity}%\n"

with open("environmental\_data.txt", "a") as file:

file.write(log\_entry)

# Main loop

while True:

try:

temperature, humidity = read\_sensor\_data()

log\_data(temperature, humidity)

print(f"Data logged: Temperature={temperature}°C, Humidity={humidity}%")

time.sleep(3600) # Log data every hour

except Exception as e:

print(f"Error: {str(e)}")

time.sleep(60) # Wait for a minute before retrying in case of an error

OUTPUT:

The provided Python script reads data from a DHT22 sensor and sends it to an IoT platform, and it also includes error handling and a 60-second sleep interval. The output of this program will look like this:

Data sent successfully.

Data sent successfully.

Data sent successfully.

Data sent successfully.

... (repeats every 60 seconds)

Assuming there are no errors or exceptions, it will continuously print "Data sent successfully" every 60 seconds as it sends sensor data to the specified IoT platform. If there are any errors during the process, it will print an error message.

Please ensure that you have set the correct IoT platform URL and have network connectivity to actually send data successfully.

Sensor devices:

1.Ultrasonic sensor

2.Infrared sensor

3.Magnetic sensor

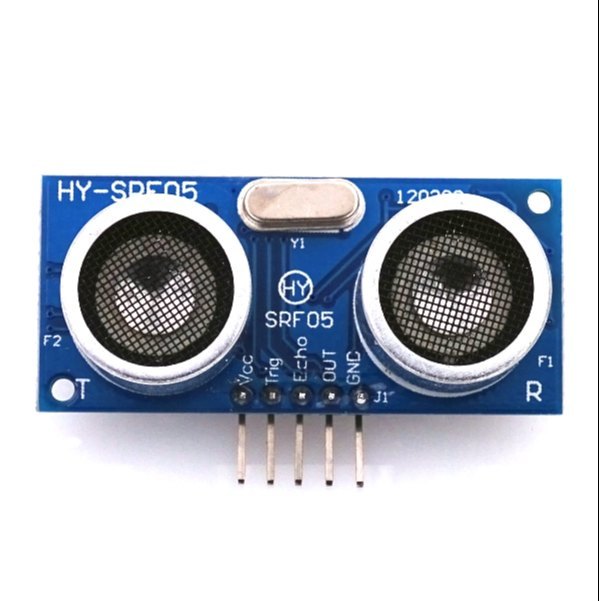
Environmental sensor

5.Camera systems

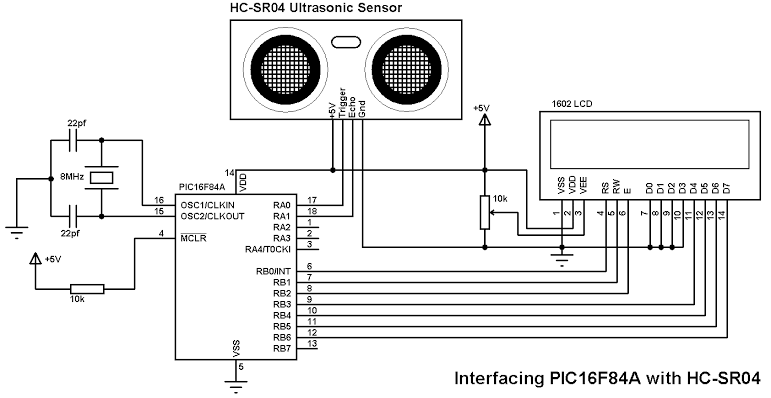
6.GPS and geolocation sensor

7.LoRaWAN sensor

8.Wireless sensor networks

1. \*Ultrasonic Sensor:\* These sensors emit ultrasonic waves to measure the distance to the nearest object. In a parking system, they can detect the presence or absence of vehicles in parking spaces. 

Pin diagram:



Pin configuration:

1. VCC (Power): This pin is connected to the positive supply voltage (usually +5V or +3.3V) to power the sensor.

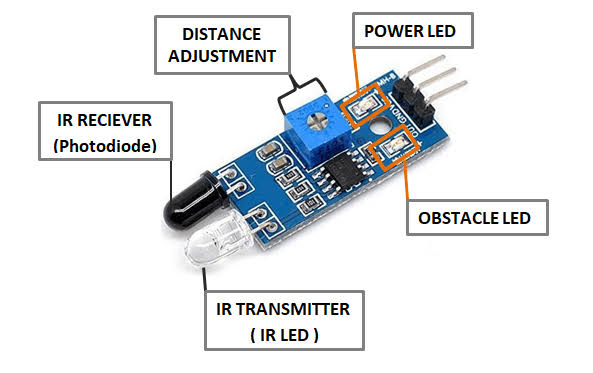
2. GND (Ground): This pin is connected to the ground or 0V reference.

3. Trig (Trigger): The Trigger pin is used to send a pulse to trigger the sensor to start measuring distance.

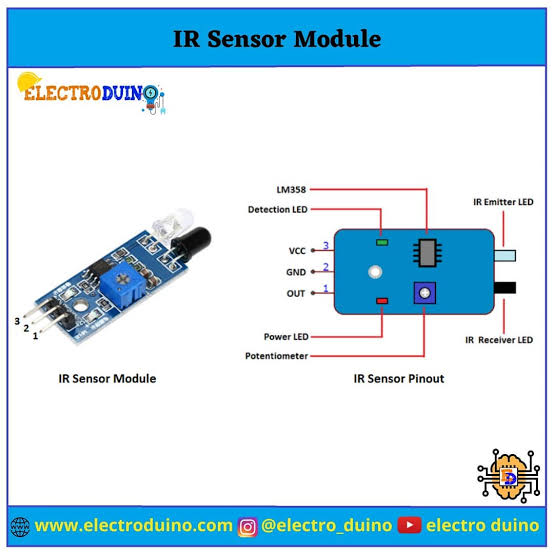
4. Echo: The Echo pin is used to receive the ultrasonic pulse reflection signal, which is used to calculate the distance to the object.

2.Infrared Sensors:

Infrared sensors detect heat and motion. They are commonly used to monitor the occupancy of parking spaces by detecting the presence of vehicles.



Pin diagram:



Pin configuration:

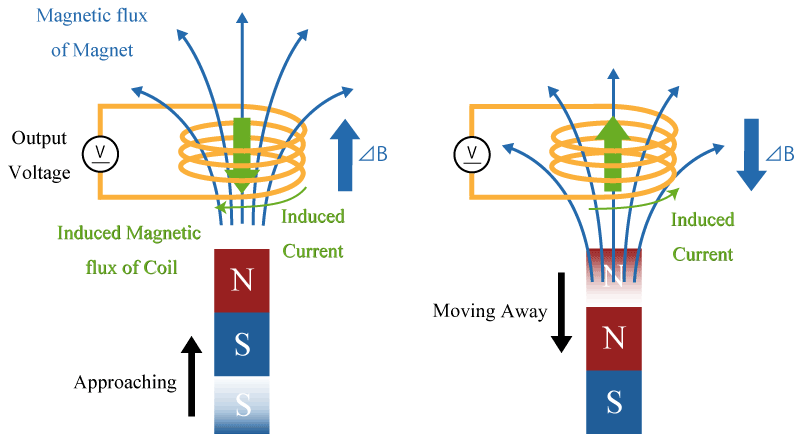
1. VCC (Power): This pin is connected to the positive supply voltage (usually +5V or +3.3V) to power the sensor.

2. GND (Ground): This pin is connected to the ground or 0V reference.

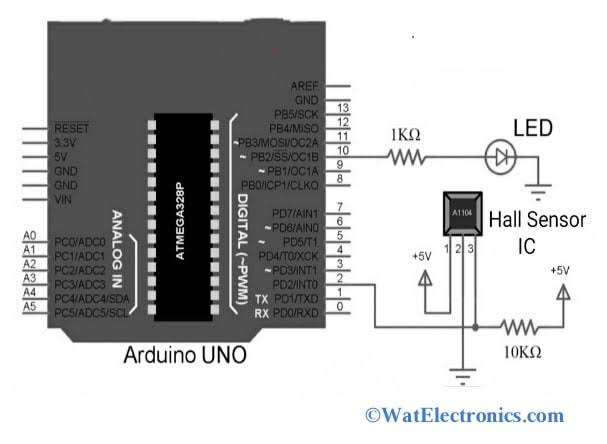
3. OUT (Output): The Output pin provides the digital output signal that changes its state when the sensor detects an IR signal. This is the pin where you connect to your microcontroller or other digital input device.

3.Magnetic Sensors:

Magnetic sensors are embedded in the road or parking spaces and detect changes in the Earth's magnetic field caused by the presence of vehicles. They are used to determine whether a parking space is occupied.



Pin diagram:



Pin configuration:

1. VCC (Power): This pin is connected to the positive supply voltage (usually +5V or +3.3V) to power the sensor.

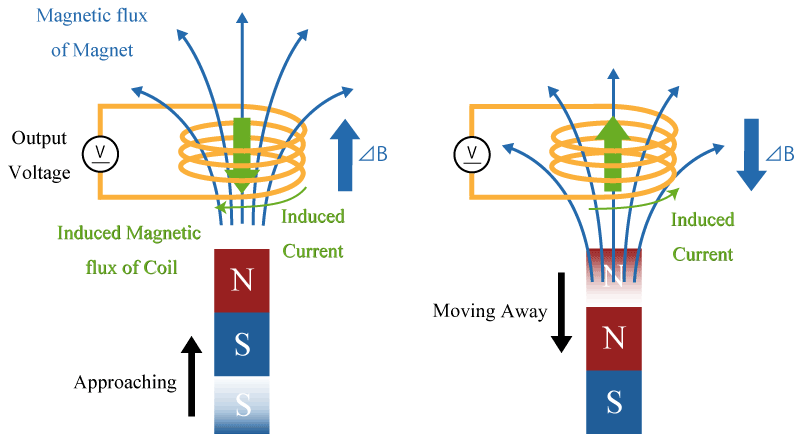
2. GND (Ground): This pin is connected to the ground or 0V reference.

3. OUT (Output): The Output pin provides a voltage or digital signal that changes based on the presence and strength of a magnetic field. This is the pin to connect to your microcontroller or other input device.

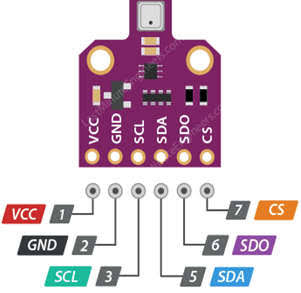
4. COM (Common): Some Hall-effect sensors have a Common pin that's connected to the ground or power supply ground, but it's not always present in all sensor module

4. Environmental Sensors:

These sensors monitor environmental parameters in the vicinity of the parking area.



Pin diagram:



Pin configuration:

1. VCC (Power): This pin is connected to the positive supply voltage (usually +3.3V or +5V) to power the sensor.

2. GND (Ground): This pin is connected to the ground or 0V reference.

3. Data/Output: This pin provides the sensor's data output, which can be either analog or digital, depending on the sensor type. For digital sensors, it often connects to a microcontroller or data acquisition system.

4. Communication Interface (Optional): Some advanced environmental sensors might include additional pins for communication interfaces, such as I2C, SPI, or UART, to provide a standardized way to communicate with the sensor.

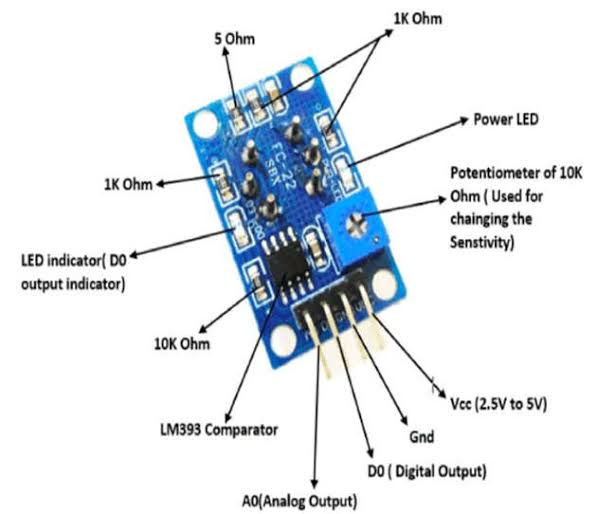
5. Additional Pins (Optional): Depending on the specific sensor's features and capabilities, there may be additional pins for features like power management, calibration, or other functions.

Air Quality Sensors:

Measure pollutants such as particulate matter (PM2.5, PM10), carbon monoxide (CO), nitrogen dioxide (NO2), and ozone (O3) to assess air quality.



Pin diagram:



Pin configuration:

1. VCC or VDD: This is the power supply pin, typically requiring 3.3V or 5V.

2. GND: Ground or common ground reference.

3. Analog Output: This pin provides an analog voltage or current signal that represents the measured air quality.

4. Digital Output: Some sensors have a digital output pin that provides a binary signal (e.g., HIGH or LOW) based on a threshold value.

5. UART or I2C Pins: Some advanced air quality sensors may offer UART or I2C interfaces for digital communication with microcontrollers.

6. Heating Element Control: Air quality sensors often include a heating element to improve accuracy. There may be pins to control this feature.

7. Calibration or Reference: Some sensors may have pins for calibration or reference purposes.

8. Alarm or Alert Output: This pin can be used to trigger an alert or alarm when air quality falls below a certain threshold.

9. LED Indicator: Some sensors have an LED indicator pin to signal the status of the sensor.

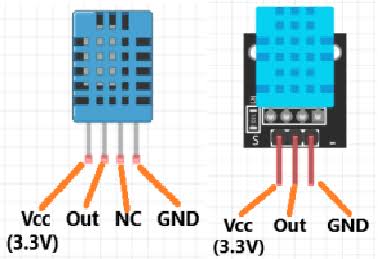
10. Enable/Disable Pin: A pin to enable or disable the sensor.

Temperature and Humidity Sensors:

Monitor ambient temperature and relative humidity to provide insights into weather conditions. - \*Noise Sensors:\* Measure sound levels to assess noise pollution.



Pin diagram:



Pin configuration:

1. VCC or VDD: This is the power supply pin, typically requiring 3.3V or 5V.

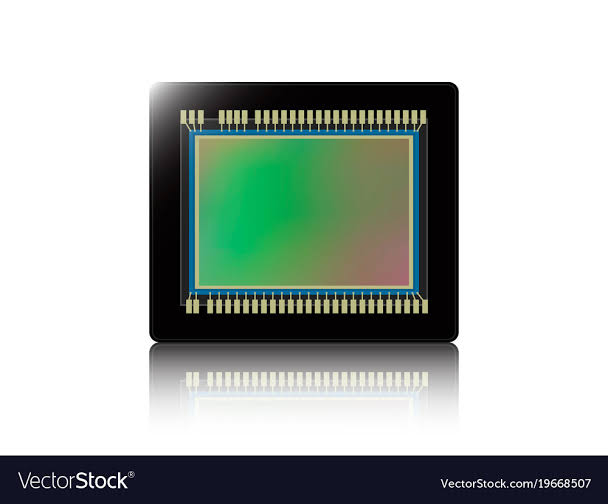
2. GND: Ground or common ground reference.

3. Data Pin: This pin is used for both data input and output. It's the pin through which the sensor communicates temperature and humidity information to a microcontroller. Data is typically sent in a digital format (e.g., one-wire or two-wire protocol).

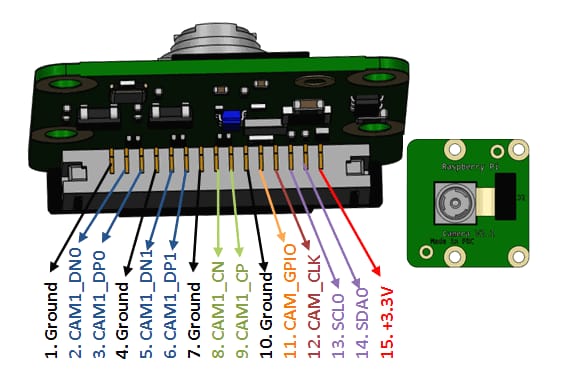
4. Optional SCK/SCL and SDA Pins: Some advanced sensors may use I2C or other digital communication protocols, which will have clock (SCK or SCL) and data (SDA) pins.

5. Camera Systems:

While not traditional sensors, cameras equipped with image recognition technology can be used for license plate recognition and to monitor parking space occupancy. They also contribute to security and surveillance.



Pin diadram:



Pin configuration:

1. Power Supply:

- VCC or VDD: Power supply voltage pin, typically 3.3V or 5V.

- GND: Ground or common ground reference.

2. Data Communication:

- MIPI CSI-2: Pins for high-speed data transfer using the Mobile Industry Processor Interface (MIPI) standard.

- USB: Pins for connecting to a computer or other devices.

- Ethernet: For network-connected cameras.

3. Control and Configuration:

- I2C or SPI: Pins for configuring camera settings and parameters.

- UART: Pins for serial communication with the camera module.

- GPIO (General Purpose Input/Output): Pins for general-purpose control or status signaling.

4. Video Output:

- Analog Video Out: Pins for analog video output.

- HDMI: For high-definition digital video output.

- SDI (Serial Digital Interface): Used in professional video applications.

5. Trigger and Shutter Control:

- Trigger Input: A pin to trigger image capture.

- Shutter Control: Pins for controlling the camera's exposure.

6. Lens Control:

- Focus, Zoom, and Iris Control: Pins for adjusting lens parameters.

7. LED or Flash Control:

- Pins for controlling built-in LEDs or flashes.

8. Audio:

- Microphone and Speaker: Pins for audio input and output.

9. Temperature and Environmental Sensors:

- Pins for interfacing with sensors to monitor temperature or other environmental conditions.

10. Networking:

- RJ45 (Ethernet): For network connectivity in IP cameras.

- Wi-Fi or Bluetooth: For wireless connectivity.

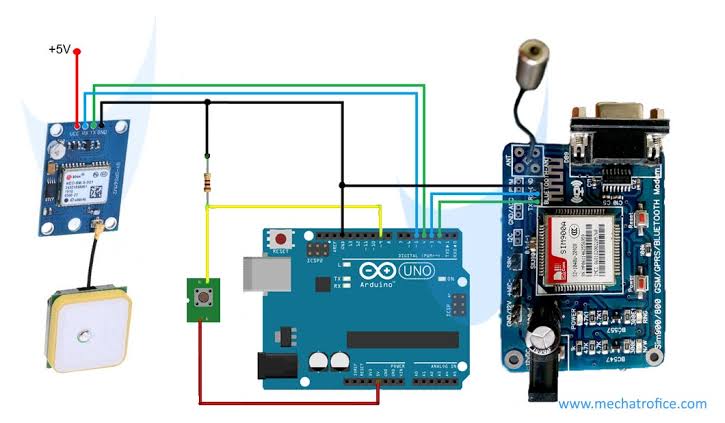
6. GPS and Geolocation Sensors:

These sensors provide geographical coordinates and are useful for tracking vehicle movements and mapping parking locations.

DIADRAM:



PIN CONFIGURATION:



1. Power Supply:

- VCC or VDD: Power supply voltage pin, typically requiring 3.3V or 5V.

- GND: Ground or common ground reference.

2. Communication:

- UART (Serial): Many GPS sensors use UART communication for sending and receiving NMEA sentences, which contain location and time data.

- I2C: Some GPS sensors may offer I2C communication for interfacing with microcontrollers or other devices.

- SPI: In some cases, SPI communication might be used for data transfer.

3. Antenna:

- GPS Antenna: A connection point for an external GPS antenna, especially in cases where the sensor is inside a shielded enclosure.

4. External Components:

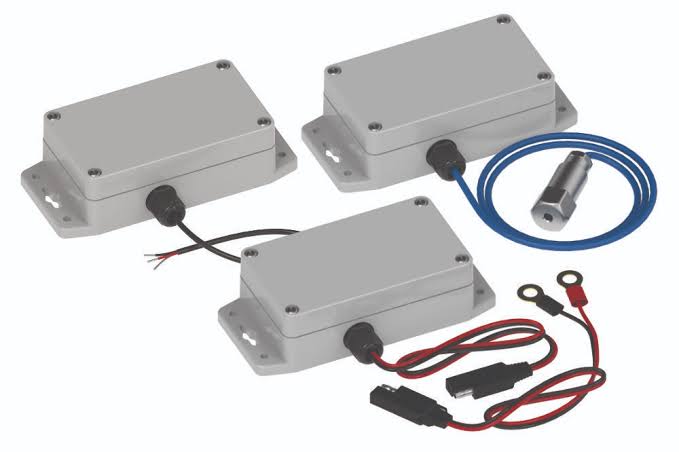
- PPS (Pulse Per Second): Some GPS sensors provide a precise 1 Hz (or other frequency) pulse output for accurate timekeeping applications.

- External Antenna Power: For supplying power to an active GPS antenna if required.

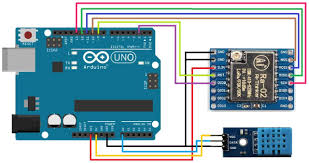
7. LoRaWAN Sensors:

Sensors with LoRaWAN communication technology can send data over long distances, making them suitable for large parking areas.

DIAGRAM:



PIN CONFIGURATION:



1. Power Supply:

- VCC or VDD: Power supply voltage pin, typically requiring 3.3V or 5V.

- GND: Ground or common ground reference.

2. Communication:

- UART: Pins for serial communication with a microcontroller or other devices.

- SPI: Pins for SPI communication, which is often used to interface with LoRa transceivers or modules.

- I2C: Some sensors may offer I2C communication for data transfer.

3. LoRa Transceiver:

- LoRaWAN Module: Pins for connecting to an integrated LoRa module or transceiver.

- Antenna: Connection point for the LoRa antenna.

4. Digital Input/Output (GPIO):

- Digital Input: Pins for connecting digital sensors or devices.

- Digital Output: Pins for controlling external devices or providing status information.

5. Analog Input:

- Analog Input: Pins for connecting analog sensors like temperature or humidity sensors.

6. Reset and Configuration:

- Reset Pin: A pin for resetting the sensor or module.

- Configuration/Mode Pins: Some sensors may have pins for configuring different operating modes or settings.

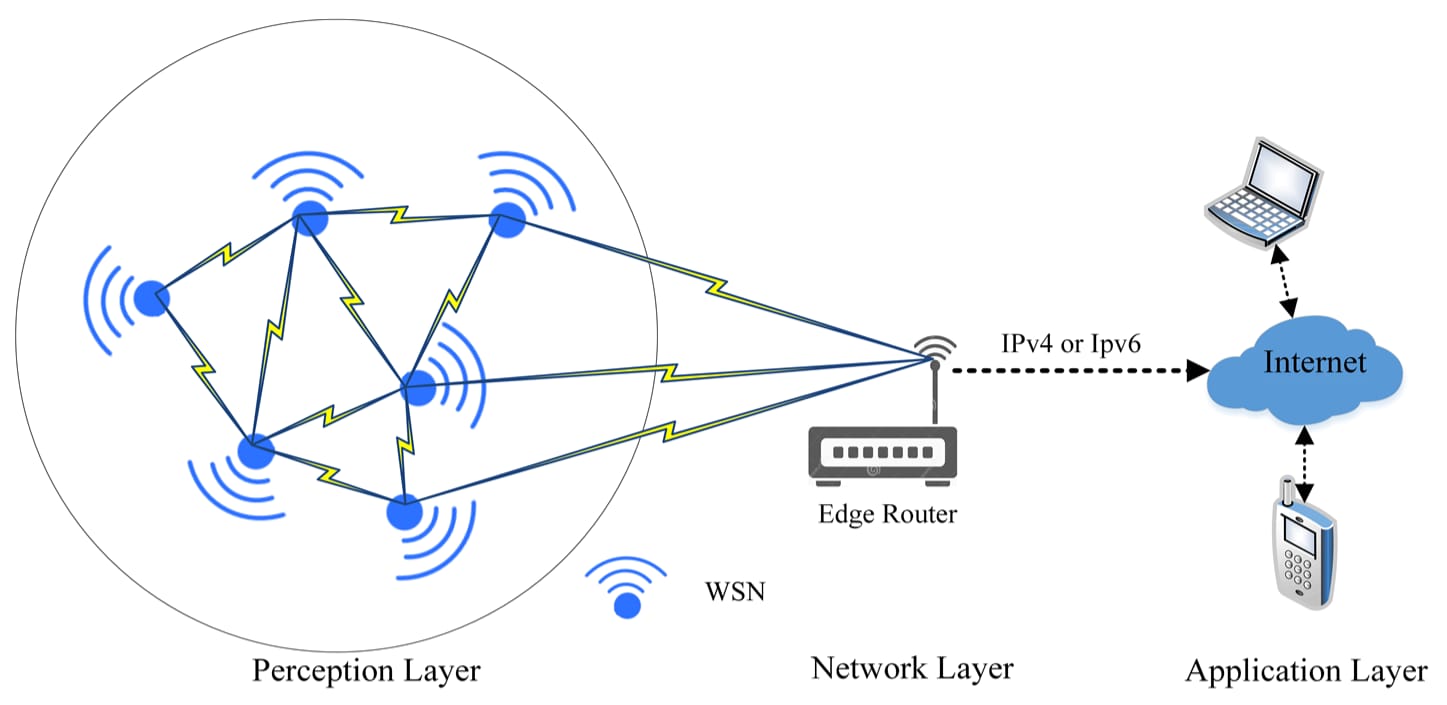
7. Power Management:

- Battery Power: Pins for connecting a battery or power source for standalone operation.

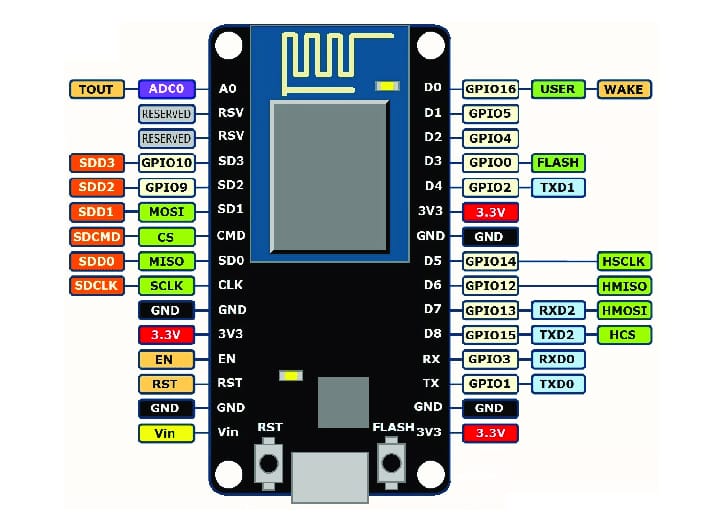
- Charging Circuit: Pins for battery charging and power management

8. \*Wireless Sensor Networks:\* These networks integrate various sensors and transmit data wirelessly to a central hub or server for analysis and decision-making.

DIAGRAM:



PIN CONFIGURATION:



1. Power Supply:

- VCC or VDD: Power supply voltage pin, typically requiring 3.3V or 5V.

- GND: Ground or common ground reference.

2. Communication:

- Wireless Module: Pins for connecting the wireless communication module (e.g., Zigbee, LoRa, Wi-Fi, Bluetooth).

- UART: Pins for serial communication with other sensor nodes or gateways.

- SPI or I2C: Pins for sensor modules or additional components that use these communication protocols.

3. Sensors:

- Analog Inputs: Pins for connecting analog sensors (e.g., temperature, humidity, light, gas sensors).

- Digital Inputs/Outputs: Pins for connecting digital sensors or controlling external devices.

4. GPIO (General Purpose Input/Output):

- Digital I/O: Pins for general-purpose input/output tasks, which can include sensor data, status indicators, or control signals.

5. Power Management:

- Battery Power: Pins for connecting a battery or power source for standalone operation.

- Charging Circuit: Pins for battery charging and power management.

- Sleep/Wake Pins: Pins to control the sleep and wake modes of the sensor nodes to conserve power.

6. Network Configuration:

- Reset Pin: A pin for resetting the sensor node.

- Address/ID Pins: Pins for configuring the node's unique address or identification.

7. LEDs and Indicators:

- Status LEDs: Pins to control status indicators.

- Error/Alert Pins: Pins for signaling errors or alerts in the network.

WEB DEVELOPMENT USING HTML:

<!DOCTYPE html>

<html>

<head>

        <title> Environmental monitoring Data</title>

  <link rel="stylesheet" href="style.css">

</head>

<body>

    <h1><font color="black"><center>SHREE VENKATESHWARA HI-TECH ENGINEERING COLLEGE</font></h1>

   <center><img src="file:///C:/Users/DHILIP%20N/Downloads/logo.jpg"></center>

    <h1><font color="brown"><a href="">Environmental monitoring Data</a></center></font></h1>

  <center><img src="file:///C:/Users/DHILIP%20N/Downloads/environment%20data.png.jpeg"></center>

    <script src="script.js"></script>

    <left><p><h2>TEAM MEMBERS</h2><h3>S.NITHYA</h3></p><left>

<h3>N.DHILIP</h3>

<h3>S.NAVEENA</h3>

<h3>R.SHANMUGAPRIYAN</h3>

<h3>G.RAMAYA</h3>

<h3>K.SAKTHIVEL</h3>

    </body>

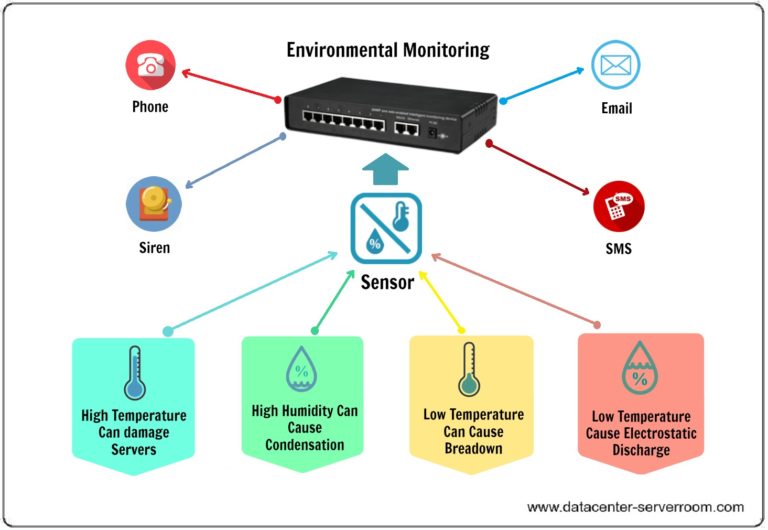
</html>

OUTPUT:

# SHREE VENKATESHWARA HI-TECH ENGINEERING COLLEGE



# Environmental monitoring Data



## TEAM MEMBERS

### S.NITHYA

### N.DHILIP

### S.NAVEENA

### R.SHANMUGAPRIYAN

### G.RAMAYA

### K.SAKTHIVEL

**Conculsion:**

# Incorporating data visualization techniques for historical temperature and humidity trends offers several benefits: improved data interpretation, enhanced communication, long-term planning, early warning systems, and educational value. These visualizations act as a bridge between data and actionable insights, contributing to a more sustainable and resilient future.

GITHUB LINK**:** https://github.com/NITHYASHANMUGAM66/nithya-shanmugam