



**An-Najah National University Faculty of Engineering  
Electrical Engineering Department**

**Graduation Project  
In Partial Fulfillment of the Requirements for the  
Bachelor's Degree in Electrical Engineering**

**Project Title:**

**Smart weather station that supports wireless and power line  
communication technology**

**Submitted by:**

- **Ali Mohammad Ali Al-Khalili- ID :12239151**
- **Fathi Fadi Rashid -ID :12239190**
- **Kamel Sharif Kamel Qaraan- ID :12239189**
- **Mohammad Imad Jawad Zaid - ID :12239177**

**Supervisor:**

**Dr. Omar Tamimi**

**Academic Year: 2025 / 2026**

## Table of Contents

Page	Section Number	Title
6	Dedication	Dedication
7	Acknowledgments	Acknowledgments
8	Abstract (EN)	Abstract (English)
9	Chapter One	Introduction
9	1.1	The Importance of Smart Weather Stations
9	1.2	Advances in Meteorological Technology
10	Chapter Two	Project Background and Objectives
10	2.1	Project Background
13	2.2	Project Objectives
13	2.3	Project Significance
13	2.4	Project Scope
15	2.5	Project Feasibility
17	Chapter Three	Traditional Weather Stations and Previous Studies
17	3.1	Types of Traditional Weather Stations
18	3.2	Similar Projects
20	3.3	The Gap Addressed by the Project
21	Chapter Four	System Components and Implementation Methodology
21	4.1	System Overview and Power Supply
28	4.2	Hardware Components
29	4.3	Functions of System Components

44	4.4	Software Components and Programming
81	4.5	Project Implementation Stages

86	Chapter Five	Data Analysis and Results
86	5.1	Reading and Analyzing Climate Data
87	5.2	Technical Results and Performance Analysis
92	5.3	Technical Conclusions
93	Chapter Six	Types of Smart Weather Stations
93	6.1	Personal Weather Stations
94	6.2	Professional Weather Stations
95	6.3	Agricultural and Industrial Weather Stations
97	Chapter Seven	How to Choose a Smart Weather Station
97	7.1	Basic Technical Criteria for Selection
100	7.2	Cost-Performance Analysis
101	7.3	Component Costs (USD)
102	7.4	Technical Summary for Selecting the Optimal Station
103	Chapter Eight	Installation and Maintenance
103	8.1	Installation and Connection Steps

104	8.2	Configuring Communication and Integration
105	8.3	Routine Maintenance and Troubleshooting
107	Chapter Nine	Building a Smart Weather Station (Practical Application)
107	9.1	Required Tools and Components
108	9.2	Wiring and Programming
109	9.3	Field Testing and Future Development
109	9.4	Expected Challenges and Proposed Solutions
110	9.5	Scalability and Future Expansion"
111		Closing Word
112		References

### List of Figures

Figure No.	Figure Title	Page
Figure 4.A	Smart Weather Station Schematic	23
Figure 4.B	Arduino Mega 2560	30
Figure 4.C	Raspberry Pi Zero W	31
Figure 4.D	BMP280 Pressure and Altitude Sensor	32
Figure 4.E	MH-RD Rain Drop Sensor	33
Figure 4.F	Wind Speed and Direction Sensor	34
Figure 4.G	NRF24L01+ Wireless Communication Module	35

Figure 4.H	Power Line Communication (PLC) Module	36
Figure 4.I	LCD Display	37
Figure 4.J	MQ-9 Gas Sensor	38
Figure 4.K	Light Meter Sensor	39
Figure 4.L	Power Supply	40
Figure 4.M	10W Solar Panel	41
Figure 4.N	Voltage Regulator	43
Figure 4.P	Data Display on TFT Screen	46
Figure 4.Q	Serial Data Processing on Raspberry Pi	63
Figure 4.R	BMP280 Reading Steps	65
Figure 4.S	Rain Sensor Reading	67
Figure 4.T	Wind Speed and Direction Steps	69
Figure 4.U	NRF24L01 Data Transmission Diagram	71
Figure 4.V	PLC Data Transmission Diagram	73
Figure 4.W	Displaying Data on TFT Screen	75
Figure 4.X	MQ-9 Gas Sensor Reading and Analysis	77
Figure 4.Y	Light Intensity Reading	79
Figure 4.Z	Complete Wiring Diagram of Components	82

## **Dedication**

To those whose prayers have lit my path and sustained my spirit...

To my dear parents, who instilled in me a love of learning, instilled confidence in me, and have always been loyal supporters every step of the way.

To my esteemed professors, beacons of knowledge and learning, who have inspired me and shaped my academic and professional thinking.

To my colleagues on this project, partners in success and challenge, whose determination and collective spirit have played a role in the completion of this work.

To everyone who believed in me and supported me with a kind word or a sincere prayer...

I dedicate this work to all of you with love and gratitude.

## **Acknowledgments**

By the grace and blessing of God, this modest project has been completed.

It represents the fruit of continuous collective effort and fruitful cooperation from many individuals who greatly supported and encouraged me throughout the project's implementation phases.

I extend my sincere thanks and gratitude to **Dr. Omar Al-Tamimi**, the academic supervisor of this project, whose valuable guidance, scientific advice, and precise observations played a major role in the success of this project.

I also extend my deep gratitude to my **esteemed professors** in the Electrical

Engineering Department, who contributed to building my scientific and practical knowledge throughout my years of study and set an example of commitment and excellence.

I must also express my deep gratitude to my dear parents, who have generously provided me with moral and material support and have supported me every step of the way.

Finally, I would like to extend my special thanks to my colleagues on the project team:

**Ali Muhammad Ali Al-Khalili, Fathi Fadi Rashid,**

**Kamel Sharif Kamel Qaraan, and Muhammad Imad Jawad Zaid**

for the spirit of cooperation, perseverance, and teamwork that brought us together throughout the project's implementation.

### **Abstract (In English)**

This project aims to design and implement a smart weather station capable of accurately measuring and analyzing weather conditions in real time. It supports dual communication methods, including wireless (Wi-Fi or LoRa) and power line communication (PLC). This enhances its usability in various environments, whether those with advanced or limited infrastructure.

The station relies on a set of precise sensors to monitor variables such as temperature, humidity, atmospheric pressure, wind speed, rain, and gas sensors. A central control unit collects and analyzes this data, then transmits it to an electronic display interface that facilitates real-time weather monitoring.

This station is an effective tool that can be employed in various fields, such as smart agriculture, energy management, and environmental planning, especially in remote areas or areas where reliable climate data is difficult to access. The project has the potential to be developed to integrate with smart forecasting systems or automated environmental systems in the future.

## **Chapter One: Introduction**

### **1.1 The Importance of Smart Weather Stations**

Smart weather stations are a pioneering application of Internet of Things (IoT) technology in environmental research, providing accurate climate readings in diverse sectors, including agriculture, renewable energy, freight traffic, and urban safety . By collecting and processing atmospheric data on-site, these devices provide planners and first responders with real metrics, enhancing preparedness amid the increasing pressures of global warming due to accelerating climate change.

### **1.2 Advances in Meteorological Technology**

Once limited to liquid-glass thermometers and anemometers carried by human observers, modern meteorology now employs digital sensors, embedded microcontrollers, and miniaturized telemetry units. Wi-Fi, LoRa, and cloud-based analysis pipelines have made way for inexpensive, self-calibrating weather stations that cover kilometers of land, while their readings are transmitted to external servers and analyzed. The problem is that many remote rural areas and some industrial corridors remain beyond the reach of reliable wireless coverage. This lack of infrastructure makes our project, which utilizes power lines for data transmission, particularly valuable .

## **Chapter Two: Project Background and Objectives 1. Project Background**

The main idea of the project is to design a smart weather station that can collect real-time climate data such as temperature, humidity, wind speed, and atmospheric pressure, then transmit it to display and analysis interfaces using two methods:

- **The first method:** wireless communication (Wi-Fi or LoRa), which is the traditional method.
- **The second and unique method:** data transmission via power lines (Power Line Communication (PLC)).

### **1. What makes a PLC an innovative idea?**

- Relying on the existing electrical grid without the need to establish new communication networks.
- High flexibility in areas lacking Wi-Fi or LoRa coverage. • Low cost because the infrastructure is already in place.
- Accessibility to remote locations within agricultural or industrial facilities without the need to dig or extend networks.

### **2. Relying on the existing electrical grid without the need to establish new communication networks.**

- **Explanation:**

In traditional systems, providing data communication requires laying special cables (such as Ethernet or fiber optic cables) or relying on wireless networks such as Wi-Fi. In a PLC, the existing electrical grid (the wires in walls used to distribute electricity) is exploited to transmit data over the electrical current itself.

- **Advantage:**

This eliminates the need for any new installation costs or architectural modifications, making the system quick to deploy and inexpensive.

### **3. High flexibility in areas lacking Wi-Fi or LoRa coverage**

- **Explanation:**

In areas where wireless signals such as Wi-Fi or LoRa IoT technologies are difficult to find (such as basements, metal factories, or mountainous areas), wireless communications are inefficient or unstable.

- **Benefit:**

PLC technology is not significantly affected by environmental conditions because it relies on fixed electrical cables, giving it high operational flexibility in places where other methods fail.

### **4. Low cost because the infrastructure is already in place**

- **Explanation:**

Establishing a traditional communications network requires:

- Network cables
- Signal boosters
- Distribution devices

In the case of PLC, the basic infrastructure—the electrical grid—is already in place, requiring only PLC converters (small devices that plug into electrical outlets and send and receive data).

- **Benefit:**

This significantly reduces installation and maintenance costs, making the technology suitable for projects with limited budgets or large-scale use.

**Summary :**

PLC technology combines the benefits of existing technologies, reducing costs, and providing connectivity in challenging environments. This makes it a smart and innovative choice for Internet of Things (IoT) projects, monitoring systems, industrial control, and even home networks.

**PLC technology is not significantly affected by environmental conditions because it relies on fixed electrical cables, giving it high operational flexibility in places where other methods fail.**

- In many industrial or agricultural facilities, environmental conditions may be unsuitable for the use of wireless communication technologies such as Wi-Fi or LoRa, whether due to metal insulators, large spaces, or electromagnetic interference.
- What distinguishes Power Line Communication (PLC) technology is its reliance on existing power lines, which are fixed and extend to various locations within the facility.
- This makes it possible to transfer data between different devices easily and stably, without the need to install new cables or use radio waves, which are affected by environmental factors such as thick walls or long distances.

## **2.2 Project Objectives**

- To build a weather station capable of operating autonomously without direct human intervention.
- To integrate Wi-Fi and PLC technologies for data transmission.
- To intelligently analyze data to produce weather reports and alerts.
- To design a simple user interface for visualizing data.
- To demonstrate the feasibility of using PLCs as an alternative or complement to wireless communication.

## **2.3 Project Significance**

- The project provides an effective technical solution to the problem of weak infrastructure in non-urban areas.
- To demonstrate how traditional electrical grids can be repurposed for modern digital purposes without major modifications.
- To serve the smart agriculture, energy, and environment sectors.
- To open the way for the use of PLCs in other future applications, such as smart irrigation or industrial monitoring.

## **2.4 Project Scope**

**The project includes:**

### **1. Using sensors to measure (temperature, pressure, wind speed and direction, rain sensor, light sensor, gas sensor)**

- A set of sensors is installed in a specific location, such as a greenhouse or an open area.
- These sensors periodically measure weather conditions.

## **2. Designing an integrated electronic circuit that includes an Arduino Mega 2560 microcontroller.**

- All sensors are assembled and connected to an Arduino Mega 2560 control board.
- This module is responsible for reading data from the sensors, processing it, and preparing it for transmission.
- It is powered either by a direct electrical source or by solar energy in the event of a power outage.

## **3. Sending data via Wi-Fi. If Wi-Fi is unavailable, it is sent via power lines (PLC).**

- Normal mode: Data is sent directly to the server or user interface via Wi-Fi.

## **4. Receive and analyze data on an electronic platform (web interface or simple application)**

- A simple interface (dashboard) is programmed to display the values received from the sensors live.
- The values can be displayed on a screen, a web page, or a local mobile application.

## **2.5 Project Feasibility**

## **1. Cost Reduction**

- No need to extend network cables or install Wi-Fi towers.
- Utilizing the existing electrical grid means virtually free infrastructure.

## **2. Operational Capability in Complex Environments • Works**

in greenhouses or metal structures that impede Wi-Fi.

- Suitable for farms, warehouses, or factories.

## **3. Enhanced Reliability**

- The hybrid system (Wi-Fi + PLC) ensures continuous connectivity even if one of the two paths fails.

## **4. Scalability**

- The system can later be integrated with automated irrigation or cooling systems.
- Or linked to other smart platforms such as Firebase or Home Assistant.

## **5. Proof of Concept**

- The project does not aim to market an off-the-shelf product, but rather demonstrates the feasibility of operating a smart system at a low cost and with high efficiency, an important step in research and development projects.

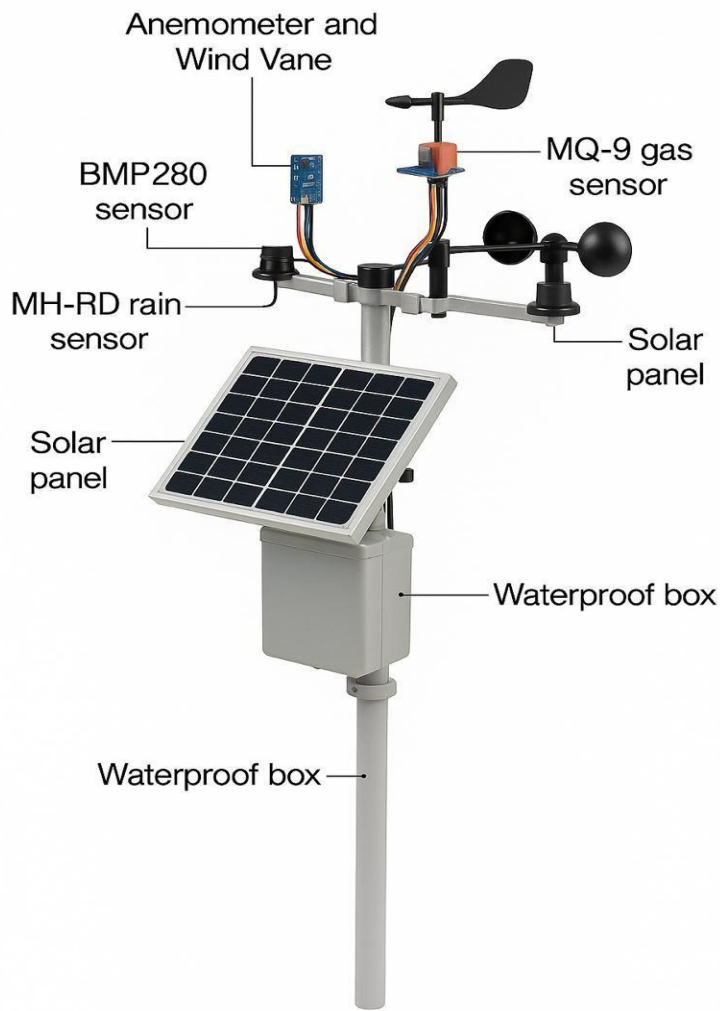


Figure (2.a): Schematic diagram of the components of a smart weather station

## **Chapter Three: Traditional Weather Stations and Previous Studies**

### **3.1 Types of Traditional Weather Stations**

#### **First: Manual Weather Stations**

Rely on readings from traditional measuring devices such as thermometers, mercury gauges, and handheld barometers. They require direct human intervention to record data, leading to:

- Delayed processing.
- Potential human error.
- Limited real-time data updates.

#### **Second: Wired Automatic Stations**

Use digital sensors connected to a storage unit or central processor via wires. These stations are more accurate than manual stations, but suffer from:

- Difficulty in installation in open environments.
- Restricted movement due to wires.
- High installation cost.

#### **Third: Wireless Weather Stations**

Use transmission modules (such as Wi-Fi, Zigbee, or LoRa) to transmit data from sensors to a server or central system. Their advantages:

- Easy to install and expand.
- Low power (especially LoRa).
- Suitable for farms and open spaces. However, it:
- Is affected by signal quality.
- May be ineffective in areas with poor coverage or crowded frequencies.

## **3.2 Similar Projects**

### **1. Wi-Fi-only Stations**

Several smart weather projects have been developed that rely on ESP8266 or ESP32 modules to send data to cloud servers via Wi-Fi. However, they fail to work in areas without stable internet coverage.

### **2. LoRa (Long Range) Stations**

LoRa (Long Range) has proven its efficiency in transmitting environmental data over long distances using very low power, but they:

- Require a central receiving unit (gateway).
- Susceptible to interference in urban environments.
- Require a frequency license in some countries.

### **3. GSM/3G/4G Models**

Rely on mobile networks, but their cost is higher due to data subscriptions and they consume more power.

## **Similar Projects from Universities and Companies**

### **1. Project from TU Delft University – Netherlands**

Title: “Hybrid Communication System for Smart Greenhouses Using PLC and Wireless Sensor Networks”

- Concept: Develop a smart monitoring system within a greenhouse using sensors to measure temperature and humidity. Data is initially transmitted via Wi-Fi, and upon failure, it is automatically switched to PLC technology.
- Goal: Provide continuous and stable connectivity in greenhouses that often block wireless signals.

- Result: The system achieved a success rate of over 92% in terms of uptime compared to systems that rely solely on Wi-Fi.

## **2. Siemens – PLC Solution for Industrial Systems**

### **Project: “SIMATIC Powerline Communication Module”**

Goal: Use PLCs in industrial environments to control machines in areas that are difficult to access via Wi-Fi or Ethernet.

Result: The company reduced the need for new communication cables by 70% and increased operational efficiency in large factories.

## **3. Research Project – University of Malaya, Malaysia Title: “IoT-based Smart Farming using Arduino and PLC Communication”**

- Objective: To design a smart agricultural system that uses sensors connected to an Arduino, sending data via a PLC to a central server. •
- Results: The system successfully reduced water consumption by 25% by monitoring soil and weather.
- Advantages of the study: The use of inexpensive modules such as the Arduino, making the system suitable for small and medium-sized farms.

## **4. Project from Devolo – Germany**

### **Project: “Smart Home over Powerline”**

- Technology: Smart home systems that rely entirely on a PLC to transmit the internet and control lighting and appliances using only electricity.

- Objective: To overcome the problems of thick walls or weak Wi-Fi signals in homes.
- Advantages: Easy installation, with no additional wiring required.

### **5. Research Project – Cairo University (Faculty of Engineering)**

#### **Title: “Design of a Smart Environmental Monitoring System using ESP32 and Hybrid Communication (Wi-Fi & Powerline)”**

- Objective: A system for monitoring air quality and temperature in crowded industrial areas.
- Technologies used: ESP32 + sensors + PLC modules. •

Results: The project succeeded in providing real-time monitoring and warnings when pollutants exceed permissible limits.

### **3.3 The Gap Addressed by the Project**

Despite the advancement of weather systems, there remains a clear gap in areas with weak communication network infrastructure. This gap includes:

- The inability to transmit data in real-time.
- The lack of wireless communications in some areas.

Here comes the unique solution: Power Line Communication (PLC).

**Power Line Communication (PLC) technology is not common in weather stations, but it represents an innovative solution to address this gap, allowing:**

- Using the existing electrical grid as a digital communication channel.
- Delivering data from remote points to processing units without the need for any additional communications infrastructure.
- Expanding the station without digging or extending additional networks.

## Chapter Four: System Components and Project Implementation

### Methodology 4.1 System Overview

The system designed in this project consists of a set of functionally interconnected modules that work together to collect, process, and transmit climate data using two technologies:

- Wireless communication (Wi-Fi or LoRa) .
- Data transmission over power lines (PLC), the technology that distinguishes this project.

This combination represents a flexible and smart solution to address communication problems in areas with weak or unstable infrastructure.

### The power source used in the project

(220V main power source, batteries, solar charge controller, solar panel)

### Types of power supplies used:

Power supply components in a smart weather station

Source	Details
<b>Main Power Supply 220V</b>	An electrical power source used as the primary energy supply, typically connected through a transformer to a voltage regulator.
<b>Batteries</b>	Such as Li-ion or 18650 batteries – used to store energy and power the system when sunlight is unavailable or during a power outage.
<b>Adapter</b>	An electrical adapter that converts voltage from 220V to 5V or 12V to power sensitive electronic devices like Arduino or Raspberry Pi.

<b>Solar Charger</b>	Consists of a solar panel + a charge controller – charges the batteries using sunlight when the main power source is unavailable.
----------------------	---

## **1. Direct Power Supply**

### **Description:**

The station is connected directly to a 220V mains power source via an adapter that converts the voltage to 5V or 12V.

### **Advantages:**

- A stable and reliable power source.
- Simple to connect.
- Suitable for locations near electricity.

### **Disadvantages:**

- Does not operate during a power outage.
- Requires the station to be located near a permanent power source.

For this reason, we have included a solar power system.

## **2. Solar Power System Consists**

### **of:**

- A 12V or higher solar panel.
- A solar charge controller.
- A storage battery.
- A voltage regulator (Buck Converter or AMS1117).

### **Advantages:**

- Ideal for standalone operation outdoors.
- Environmentally friendly.
- Operates at night using the stored battery.

- Operates in the event of a power outage.



Figure (4.a): Model of a smart weather station **3. The basic components of a dual energy system**

### Power System Components and Their Functions

Component	Function
⚡ 220V AC Power	Main power source when electricity is available.
☀️ Solar Panel	Alternative source that charges the battery when electricity is unavailable.
🔋 12V Li-ion Battery	Stores energy from the solar panel to power the system when needed.
⚡ Solar Charge Controller	Regulates the battery charging process and protects it from overcharge or discharge.
🔄 Relay	Automatically switches between the two power sources based on availability.

🔧 Voltage Regulator (12V to 5V)	Converts voltage to 5V to power the Raspberry Pi.
🔌 AC-DC Power Supply	Converts 220V AC to 12V DC to power the system when electricity is available.

---

#### 4. Smart Switching and Power Source Management Mechanism **Case 1:**

**220V AC Power is Available** 220V AC → Power Supply → 12V DC

This voltage powers the **relay** as the primary source.

From the relay → **Arduino** and **Raspberry Pi** (via a voltage regulator).

#### **Case 2: 220V Power Outage**

- The relay detects a voltage loss from the power supply
- Automatically switches to the battery as an alternative source
- 12V from the battery → relay → rest of the system

### **Line 1: From the main power supply**

[220V AC]

↓

Power Supply (AC to DC)

↓ (12V DC )

Relay →

↘ Arduino Mega (direct)

↘ Voltage Regulator (12V → 5V) → Raspberry Pi

---

### **Line 2: From the solar system**

[12V solar panel]

↓

Solar Charge Controller

↓

12V Battery

↓

Relay (operates when the power is disconnected from the primary source  
to  
operate the secondary source)

**The relay here acts as an Automatic Transfer Switch:**

If 220V is available, it uses it as the main source.

If the power goes out, it automatically uses the solar-charged battery.

The Complete Power Cycle

- [Main Power Source: 220V] → Power Supply → 12V → Relay
- [Backup Source: 12V Battery] ← Solar Charger ← Solar Panel



Relay Switch



- 12V → Arduino
- 12V → Regulator → 5V → Raspberry Pi

**Smart Weather Station Power System Summary**

**The Smart Weather Station project's power system relies on two main power sources to ensure continuous operation:**

**First (Main) Source:**

220V AC household electricity converted to 12V DC using a power supply.

Then the power is passed through the system via an electrical relay, and then reduced to 5V to power the Raspberry Pi using a voltage regulator.

**Second (Reserve) Source:      A solar power system consisting of:**

- 12V solar panel
- Solar charge controller to regulate the voltage and power the battery
- 12V lithium battery used in the event of a power outage

**The relay acts as a smart switch (Automatic Transfer Switch), automatically switching between the two sources depending on availability:**

- Uses 220V electricity when available.
- Switches to battery power when the power goes out.
- Both the Arduino Mega and Raspberry Pi are powered by one of the two sources using precise voltage regulation.

Power is distributed to all sensors and components, protected by diodes to prevent backflow and reverse polarity.

**System Features:**

- Continuous operation in all conditions.
- Integration of solar power with the electrical system.
- Full protection for the controllers.
- Flexible operation without manual intervention.



## 4.2 Hardware Components

Components used in the Smart Weather Station project

Type	Component Used
Controller Unit	Arduino Mega 2560 – Responsible for reading sensors, processing, and sending to communication units.
Input Units	Sensor set includes: <ul style="list-style-type: none"> <li>• Temperature Sensor</li> <li>• Pressure Sensor</li> <li>• Humidity Sensor (BME280)</li> <li>• Rain Sensor (MH-RD)</li> <li>• Gas Sensor (MQ-9)</li> <li>• Wind and Direction Sensor</li> <li>• Light Meter Sensor</li> </ul>
Display Interface	LCD Screen (1280×400) to display data + Raspberry Pi Zero W for a local graphical interface.
Communication Units	<ul style="list-style-type: none"> <li>• Wi-Fi Module (e.g., NRF24L01)</li> <li>• PLC Module for data transmission over power lines.</li> </ul>
Power Source	<ul style="list-style-type: none"> <li>• Solar Panel 10W / 12V</li> <li>• Two Li-ion 18650 Batteries</li> <li>• Solar Charge Controller for battery protection and charge regulation.</li> </ul>
Voltage Regulator	• LM2596-5V Voltage Regulator to power Raspberry Pi Zero W
Power Supply	Used to step down voltage from 220V to 12V and convert from AC to DC

### 4.3 System Components and Their Functions

System components and their functions – Smart Weather Station

Component	Function
Arduino Mega 2560	Central control unit that reads sensor signals and sends them to the communication unit.
Raspberry Pi Zero W	Supports advanced processing and runs a graphical interface to display data locally.
Light Meter Sensor	Measures light intensity (Lux), used to evaluate sunlight or agricultural lighting.
Wind Sensor	Measures wind speed and direction via analog signals.
BME280 Sensor	Integrates high-precision atmospheric pressure, temperature, and humidity measurement.
Rain Sensor	Detects the presence of rain and is used for alerts or automatic control.
Air Quality Sensor	Monitors pollutants like CO or PM2.5 to assess ambient air quality.
Wi-Fi / LoRa Module (NRF24L01)	Transmits data wirelessly to the server or user interface.
PLC Module (Power Line Communication)	Transmits data over power lines when wireless coverage is unavailable.
LCD Screen (1280×400)	Used to display weather data locally in real time.
5V LM2596 Voltage Regulator	Steps down voltage from 12V to 5V to power Arduino and compatible sensors.
Solar Panel 10W / 12V	Used to charge batteries during the day.
Solar Charge Controller	Regulates battery charging and protects against overcharging or over-discharging.
Two Li-ion 18650 Batteries (3.7V / 2600mAh)	Provide power during electricity outage, often connected in series to produce 7.4V.

Power Supply	Used to step down voltage from 220V to 12V and convert from AC to DC.
--------------	---

**First: A detailed explanation of the hardware components used.**

### **1.Arduino Mega 2560**

- Function: Main control unit for sensors.
- Features: Large number of I/O pins and more memory than the Arduino Uno.
- Programming: Arduino IDE (C++) to collect data from sensors and transmit it wirelessly or over power lines.
- Operating voltage:

VIN input = 7–12V

Powered by a voltage regulator or 9V/12V source



Figuer(4.B): Arduino Mega 2560

## 2.Raspberry Pi Zero W:

- Function: Central processing unit, used to display and analyze data on a graphical interface.
- Features: Wi-Fi support, small size, and Linux operating system.
- Programming: Python – used to display data on a website or graphical dashboard.
- Typical operating voltage: 5V via Micro-USB port

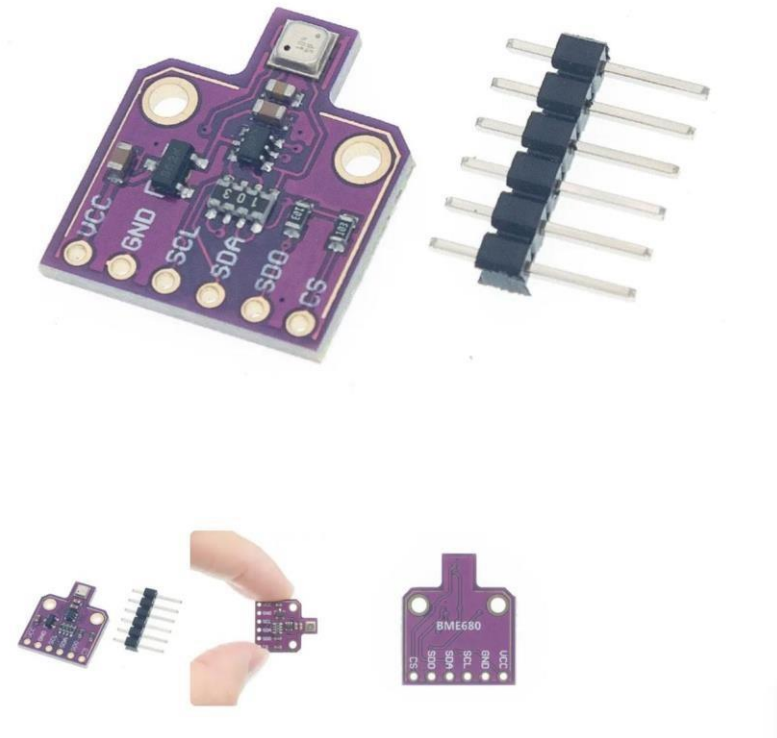
## Raspberry Pi Zero W



Figuer(4.C): Raspberry Pi Zero W

### **3.BMP280 Sensor (Pressure & Altitude) Function:**

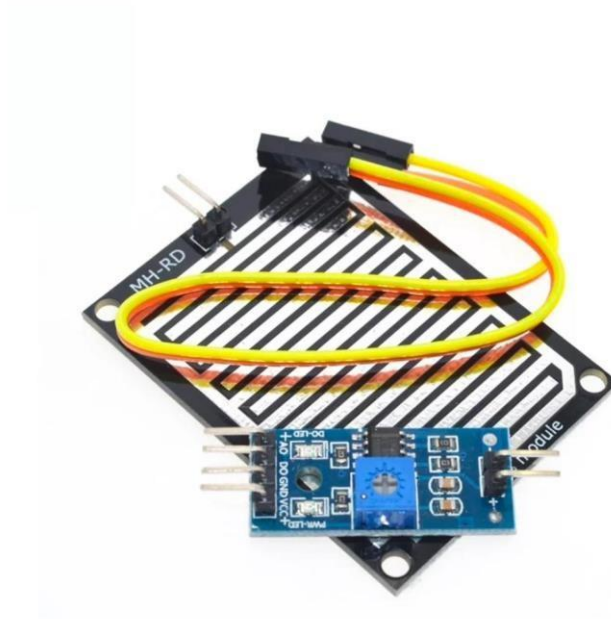
- Measures atmospheric pressure and altitude.
- Measures temperature and humidity.
- Accuracy: More accurate than the DHT11.
- Programming: DHT library for Arduino.
- Operating voltage: 5V.
- Communication: I2C or SPI.



Figuer(4.D): BMP280 Sensor (Pressure & Altitude)

#### 4. MH-RD Rain drop Sensor Module

- Function: Rain sensor.
- Programming: Reading the analog value and converting it to a rain/no rain state.
- Operating voltage: 5V



Figuer(4.E): MH-RD Rain drop Sensor Module

## **Wind Speed and Direction Sensors\ Anemometer and WindVane5.**

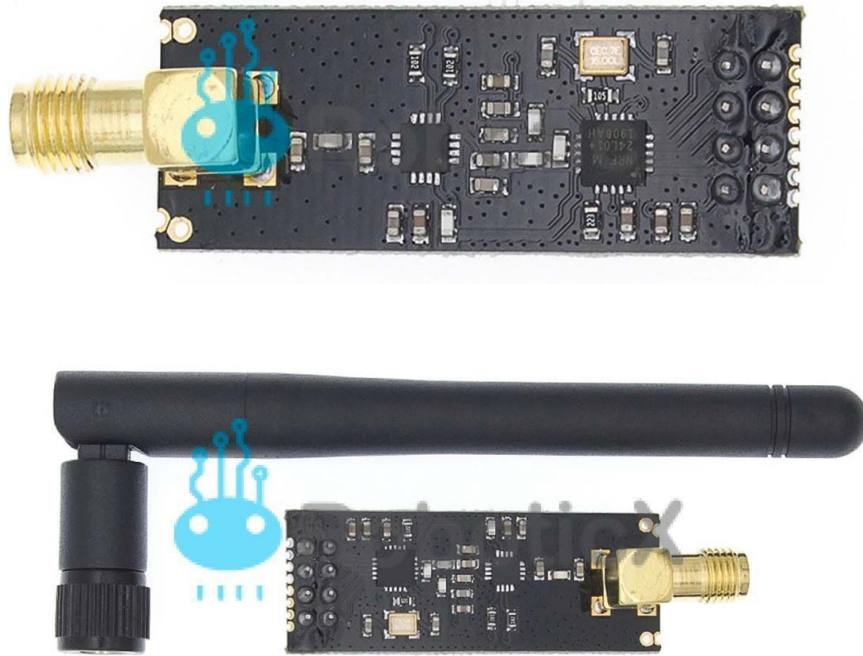
- Function: Measuring wind speed and direction.
- Type: Typically analog or digital sensors based on magnetic pulses.
- Programming: Calculating the time between pulses or reading the voltage.
- Operating voltage: 5V



Figuer(4.F): Wind Speed and Direction Sensors/ Anemometer and WindVan

## **6.WIRELESS COMMUNICATION MODULE - 2.4GHz NRF24L01+PA+LNA SMA Wireless Transceiver Antenna**

- Function: Send data over long distances using low-power radio waves.
- Communication: SPI.
- Programming: Using the LoRa library in Arduino.
- Operating voltage: 5V



Figuer(4.G): WIRELESS COMMUNICATION MODULE - 2.4GHz NRF24L01+PA+LNA  
SMA Wireless Transceiver Antenna

## 7.Power Line Communication (PLC) Module

- Function: Transfers data over power lines.
- Connectivity: Integrates with Arduino or Raspberry Pi and uses protocols such as HomePlug or X10.
- Programming: Depends on the type of PLC used, and data is often exchanged via UART or SPI.
- Operating voltage: 12V, often operated directly from a 12V supply.



Figuer(4.H): Power Line Communication (PLC) Module

## 8. LCD screen

- Function: Displays data directly to the user.
- Programming: Using the Adafruit or LiquidCrystal libraries
- Operating voltage: 5V or 3.3V



Figuer(4.1): LCD screen

## 9.MQ-9 – Gas Sensor Module Its

primary function:

Detecting flammable and toxic gases, especially:

- Carbon monoxide (CO)
- Methane (CH<sub>4</sub>)
- Butane and other flammable gases

How does it work?

The sensor contains an internal heating element and chemicals that react with gases.

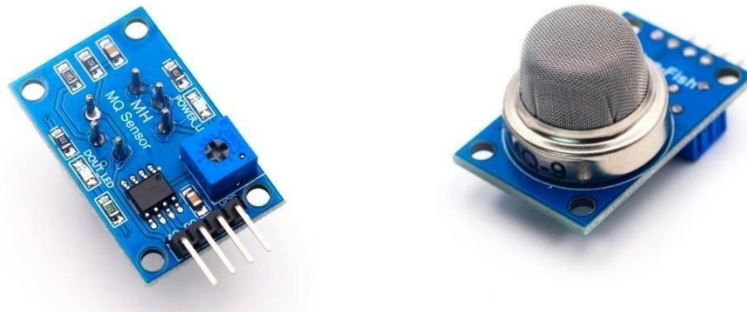
When gas is present, the element's resistance changes, producing a voltage difference that is sent to the Arduino.

The programming language for the MQ-9 sensor when used with Arduino is:

C++ via the Arduino IDE

The sensor is programmed using the Arduino IDE, which is based on C++.

- Operating voltage: 5V



Figuer(4.J): MQ-9 – Gas Sensor Module

## 10.Light Meter Sensor

Light Meter Sensor

✓ Main Function:

- Measures luminous intensity or ambient brightness.
- Provides a digital reading expressing the amount of light reaching the sensor, which can be correlated with solar radiation levels.
- Operating Voltage: 5V



Figuer(4.K): Light Meter Sensor

## 11.Power Supply

We need it to reduce the voltage from 220V to 12V.

### 1. AC to DC converter (Power Adapter / SMPS)

Component	Function
AC-DC Power Supply (Electrical Current Converter)	Takes 220V AC electricity from the wall and converts it to 5V DC or 12V DC.

Power Supply Type

### 2. In case of electricity operation (220V)

النوع	الوصف

<p>AC-DC Switching Power Supply 220V to 12V</p>	<p>An adapter that takes 220V AC power from the wall and outputs 12V DC at 2A or more.</p>
---	--



Figuer(4.L): Power Supply

13 .olar Panel 10W / 12V



Figuer(4.M): olar Panel 10W / 12V

### **Provides 12V power for charging**

- Function: Supplies power to the system
- Operating Voltage: Solar Panel 10W / 12V — Generates 12V for charging

### **Solar Charge Controller + 18650 Batteries**

- Provides electrical power to operate all system components such as sensors, communication modules, and microcontrollers (like Arduino or Raspberry Pi).
- Ensures continuous operation at all times, especially in outdoor environments or areas far from conventional power sources -

### **14.voltage regulator**

#### **Voltage variations from power sources:**

- Solar panels produce 12V (or more) during peak hours.
- Batteries (such as 18650s) produce approximately 7.4V when connected in series.

However, most components, such as Arduinos, Raspberry Pis, and sensors, cannot handle these high voltages. For this, we use a voltage regulator.

#### **Suppose there is no voltage regulator:**

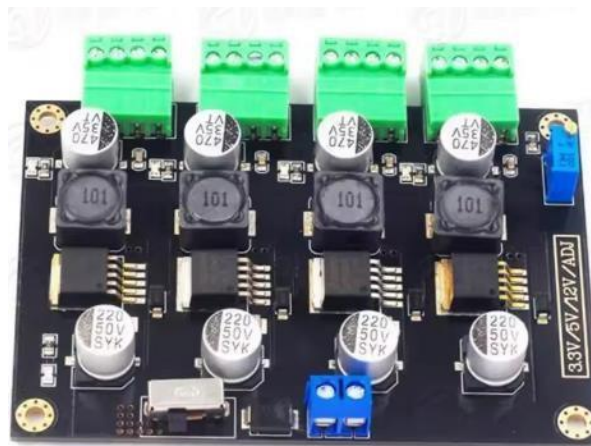
- A sensor such as the NRF24L01 (which cannot withstand 5V) may burn out.
- The control board may be damaged due to over-power.
- False readings or system crashes may occur.

### Voltage Regulator Function

- Reduces and stabilizes voltage to a safe and stable level (5V or 3.3V).
- Protects components from damage caused by voltage spikes or fluctuations.

Types of voltage regulators in the project:

Regulator	Output Voltage	Used for
LM2596	5V	5V sensors, LCD screens, Arduino Mega
AMS1117-3.3	3.3V	Raspberry Pi GPIO, BME280, NRF24L01



## 4.4 Software Components

- Programming the Arduino module using C/C++ and the Arduino IDE.
  - Using Python or Node-RED on a Raspberry Pi to process and display data.
- A dashboard displays real-time data.
- Custom PLC software to convert digital signals into signals transmitted over power lines.

### 1. Programming an Arduino module using C/C++ and the Arduino IDE Objective:

Programming the microcontroller (Arduino Mega for example) to read and process sensor data, then sending it via LoRa or PLC

#### Tools:

- Programming language: C/C++ □ Development environment: Arduino IDE □ Required libraries:
  - DHT.h for reading the DHT22
  - Adafruit\_BMP280.h for compression
  - RTCLib.h for time
  - SD.h for storage
  - LoRa.h for transmission

### **Tasks performed by the code:**

- (Collecting data from sensors (temperature, humidity, pressure). • Calculating values or formatting them as text (String or CSV).)
- Sending them to the receiving unit.
- (Storing them on an SD card (optional))

## **2. Using Python or Node-RED on Raspberry Pi to process and display data**

### **Objective:**

Analyze data coming from an Arduino (via LoRa or PLC) and display it on a graphical display or website.

### **Option 1: Python**

#### **Tasks:**

- Receive data via a Serial or USB port.
- Save it to a CSV file or SQLite/MySQL database.
- Display it using Flask (a simple web interface) or Matplotlib to plot graphs.

### **Option 2: Node-RED**

#### **Features:**

A visual drag-and-drop environment. Easy to connect inputs and outputs, such as:

- Reading from Serial or MQTT.
  - Display on a dashboard.

- Sending notifications.

**Tasks:**

- Direct data processing.
- Displaying it on a built-in dashboard interface.
- Exporting data as a file or to a web service.

### **3. Dashboards displaying real-time data**

**Objective:**

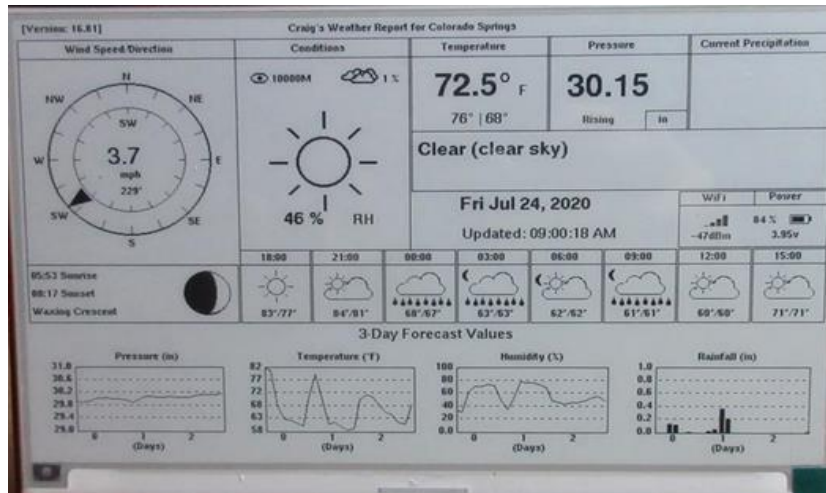
Visually display the values coming from sensors to the end user in realtime.

**Implementation tools:**

- A simple and fast Node-RED Dashboard
- A Python Flask interface for greater customization, or even a platform like ThingSpeak or Blynk.

**Displayed Content:**

- Graphs of temperature, humidity, and pressure...
- Time-lapse log.
- LED indicators when an alert occurs (such as rain or strong winds).
- Local map/radar, if available.



Figuer(4.P) :LCD

#### 4. PLC Software Dedicated to Converting Digital Signals into Power Line Signals Purpose:

Using Power Line Communication (PLC) technologies to transmit data from the Arduino to a Raspberry Pi or other system over power lines instead of a wireless network.

#### How it Works:

- A PLC module, such as a modem or PLC module, is used.
- Digital data is converted into signals that are transmitted over power lines.
- A receiver, such as a Raspberry Pi or another PLC module, captures and decodes these signals.

#### PLC Software:

- Relies on a protocol such as HomePlug, X10, or Power Line UART.
- Often, libraries or firmware are used on a dedicated microcontroller or external module (PLC TX/RX).

- The Arduino interfaces with the PLC via Serial or SPI, much in the same way as LoRa.

### **Smart Weather Station Components Software Programming:**

#### **1. Arduino Mega 2560 Function:**

It is the main controller for reading sensors, sending data, and executing commands.

#### **Programming Environment:**

- Arduino IDE •

Language: C/C++

#### **Note:**

All of the following sensors will be connected directly to this board.

#### **2. BMP280 Sensor (Pressure and Altitude)**

- **Connection: I2C**

- SDA → pin 20 •

SCL → pin 21

#### **3. Required Library:**

```
include <Adafruit_BMP280.h> #
```

**Code:**

```
#include <Wire.h>

#include <Adafruit_BMP280.h>

Adafruit_BMP280 bmp;

void setup() {

Serial.begin(9600);

if (!bmp.begin()) {

    Serial.println("BMP280 not found!");

while (1);

    }

}

void loop() {

    Serial.print("Pressure: ");

    Serial.print(bmp.readPressure() / 100.0);

    Serial.print(" hPa, Altitude: ");

    Serial.print(bmp.readAltitude(1013.25));

    Serial.println(" m");

    delay(2000);

}
```

### 3. MH-RD Rain Drop Sensor Module

#### •Connection: Analog

- Signal → A0
- VCC → 5V •
- GND → GND

#### Code:

```
int rainSensor = A0;

void setup() {
  Serial.begin(9600);
}

void loop() {  int rainValue =
analogRead(rainSensor);

  Serial.print("Rain Value: ");
  Serial.println(rainValue);

  if (rainValue < 500) {
    Serial.println("Rain Detected");
  }
else {
  Serial.println("No Rain");
}

delay(2000);
```

```
}
```

#### **4. Wind Speed and Direction Sensors / Anemometer and Wind Vane**

##### **Anemometer (wind speed):**

- Signal → D2 (interrupt) **Code:**

```
volatile int count = 0;
```

```
void windISR() { count++;
```

```
}
```

```
void setup() { Serial.begin(9600);
```

```
attachInterrupt(digitalPinToInterrupt(2), windISR, FALLING);
```

```
}
```

```
void loop() { count
```

```
= 0;
```

```
delay(1000); // Measurement interval
```

```
float speed = count * 2.4; // Example calculation: 2.4 km/h per pulse
```

```
Serial.print("Wind Speed: ");
```

```
Serial.print(speed);
```

```
Serial.println(" km/h");
```

```
}
```

##### **Wind Vane (Wind Direction):**

- Signal → A1 **Code:**

```
int windDir = analogRead(A1);
```

```
Serial.print("Wind Direction Raw: ");
```

```
Serial.println(windDir);
```

## 5. NRF24L01+ (Wireless Module)

- **Connection: SPI**

- VCC → 3.3V (not 5V!)
- GND → GND
- CE → 9
- CSN → 10
- SCK → 52
- MOSI → 51
- MISO → 50

### **Library:**

```
#include <SPI.h>
```

```
#include <nRF24L01.h>
```

```
#include <RF24.h>
```

### **Code (Sender):**

```
#include <SPI.h>
```

```
#include <nRF24L01.h>
```

```
#include <RF24.h>
```

```
RF24 radio(9, 10); // CE, CSN const
byte address[6] = "00001";

void setup() { Serial.begin(9600);
radio.begin();
radio.openWritingPipe(address);
radio.setPALevel(RF24_PA_LOW);
radio.stopListening();
}

void loop() {
const char text[] = "Hello from Weather Station";
radio.write(&text, sizeof(text)); delay(1000);
}
```

## 6. Power Line Communication (PLC) Module

### Function:

Transmits data over power lines instead of Wi-Fi or LoRa.

### Communication: Via Serial (UART)

- TX from Arduino → RX on the PLC module
- RX from Arduino → TX on the PLC module
- Common GND

- VCC depending on the module type (usually 5V or 3.3V)

**Code:**

```
void setup() {  
  Serial1.begin(9600); // Connect the PLC to Serial1: TX1=18, RX1=19  
}  
void loop() {  
  Serial1.println("Temp=24.5, Hum=55%"); delay(1000);  
}
```

**Note:** On the receiving end (Raspberry Pi or another Arduino), the same port is used to read data.

## 7. LCD Display (I2C) Communication:

**Required Libraries:**

- TFT\_eSPI: To create a graphical interface with the TFT display.
- Adafruit\_GFX: A graphical library for drawing text and graphics.

**Code:**

```
#include <TFT_eSPI.h> // TFT display library  
#include <SPI.h> // SPI library for communicating with the display  
#include <Adafruit_Sensor.h> // Sensor library
```

```

// Define the TFT display
TFT_eSPI tft = TFT_eSPI(); // Initialize the display

// Data values to be displayed
float temperature = 24.5; float
humidity = 55.0; float pressure
= 1012.0; float windSpeed =
12.3; bool rainDetected =
true; bool gasAlert = false;

void setup() { //
Initialize the screen
tft.init();

tft.setRotation(3); // Set the appropriate screen orientation
tft.fillScreen(TFT_BLACK); // Fill the screen with black

// Set font properties
tft.setTextColor(TFT_WHITE); tft.setTextSize(2);

// Display text on the screen displayWeatherData();
}

```

```

void
loop() {

// Values can be updated later from sensors delay(5000);
// Delay 5 seconds before updating
displayWeatherData(); // Update data on the screen

// Function to display weather data void
displayWeatherData() {

// Clear the screen
tft.fillRect(0, 0, 400, 1280, TFT_BLACK);

// Display data tft.setCursor(10, 10);
tft.print("Temp: ");
tft.print(temperature); tft.println("C");
tft.setCursor(10, 40);
tft.print("Humidity: ");
tft.print(humidity); tft.println("%");
tft.setCursor(10, 70);
tft.print("Pressure: "); tft.
print(pressure); tft.println("hPa");
tft.setCursor(10, 100); tft.print("Wind
Speed: "); tft.print(windSpeed);

```

```
tft.println("km/h"); tft.setCursor(10,
130); tft.print("Rain: ");
tft.println(rainDetected ? "Yes" : "No");
tft.setCursor(10, 160); tft.print("Gas:
");

tft.println(gasAlert ? "Alert!" : "Safe");

// Add icons or images as needed
}
```

---

## 8. MQ-9 – Gas Sensor Module

### Function:

Detects the presence of gases such as carbon monoxide (CO), methane (CH<sub>4</sub>), and flammable gases.

### Connection:

- A0 → Analog reading (e.g., A2)
- D0 → Digital value (signal when a certain limit is exceeded) **Code:**

```
#define gasAnalog A2
#define gasDigital 7 void
setup() {
Serial.begin(9600);
pinMode(gasDigital, INPUT);
}
```

```
void loop() {  
  int gasValue = analogRead(gasAnalog); int  
  gasAlert = digitalRead(gasDigital);  
  Serial.print("Gas level: ");  
  Serial.println(gasValue);  
  if (gasAlert == LOW) {  
    Serial.println("!! Gas Detected!!");  
  }  
  delay(2000);  
}
```

---

## 9. Raspberry Pi Zero W

### Functions:

- Receive data from LoRa, PLC, or USB.
- Run Python or Node-RED to process the data.
- Display the data on the dashboard.

**Python example to read data from Serial:** import

serial

ser = serial.Serial('/dev/ttyUSB0', 9600)

**Code:** while

True:

```
data = ser.readline().decode('utf-8').strip()
print("Received:", data) Additional
```

**Libraries:**

- Flask (for a web interface)
- Matplotlib or Plotly (for plotting data)

**1. Raspberry Pi Zero W's function in the project**

<b>Function</b>	<b>Details</b>
Reception	Via USB/Serial port from PLC or LoRa
Processing	Python script / Node-RED
Display	Dashboard interface via Flask
Data Analysis	Data visualization using Matplotlib or Plotly

**2. Official Python code for reading data via Serial**

```
Import Serial
# Connect via USB - Ensure the correct port (e.g., /dev/ttyUSB0)
ser = serial.Serial('/dev/ttyUSB0', 9600) while True:
data = ser.readline().decode('utf-8').strip() print("Received:",
data)
```

**Note:** This code works with any device that sends data via Serial, whether from LoRa, PLC, or Arduino directly.

### 3. Simple Flask code to create a dashboard from

```
flask import Flask, render_template_string
import serial
ser = serial.Serial('/dev/ttyUSB0', 9600)
latest_data = "No data yet" app =
Flask(__name__) @app.route('/') def
index():
return render_template_string("""
<h2>Smart Weather Station Dashboard</h2>
<p><strong>Latest Reading:</strong> {{ data }}</p>
""", data=latest_data)

def read_serial(): global
latest_data while True:
data = ser.readline().decode('utf-8').strip()
latest_data = data print("Received:",
data) if __name__ == '__main__':
import threading
t = threading.Thread(target=read_serial)
t.daemon = True
```

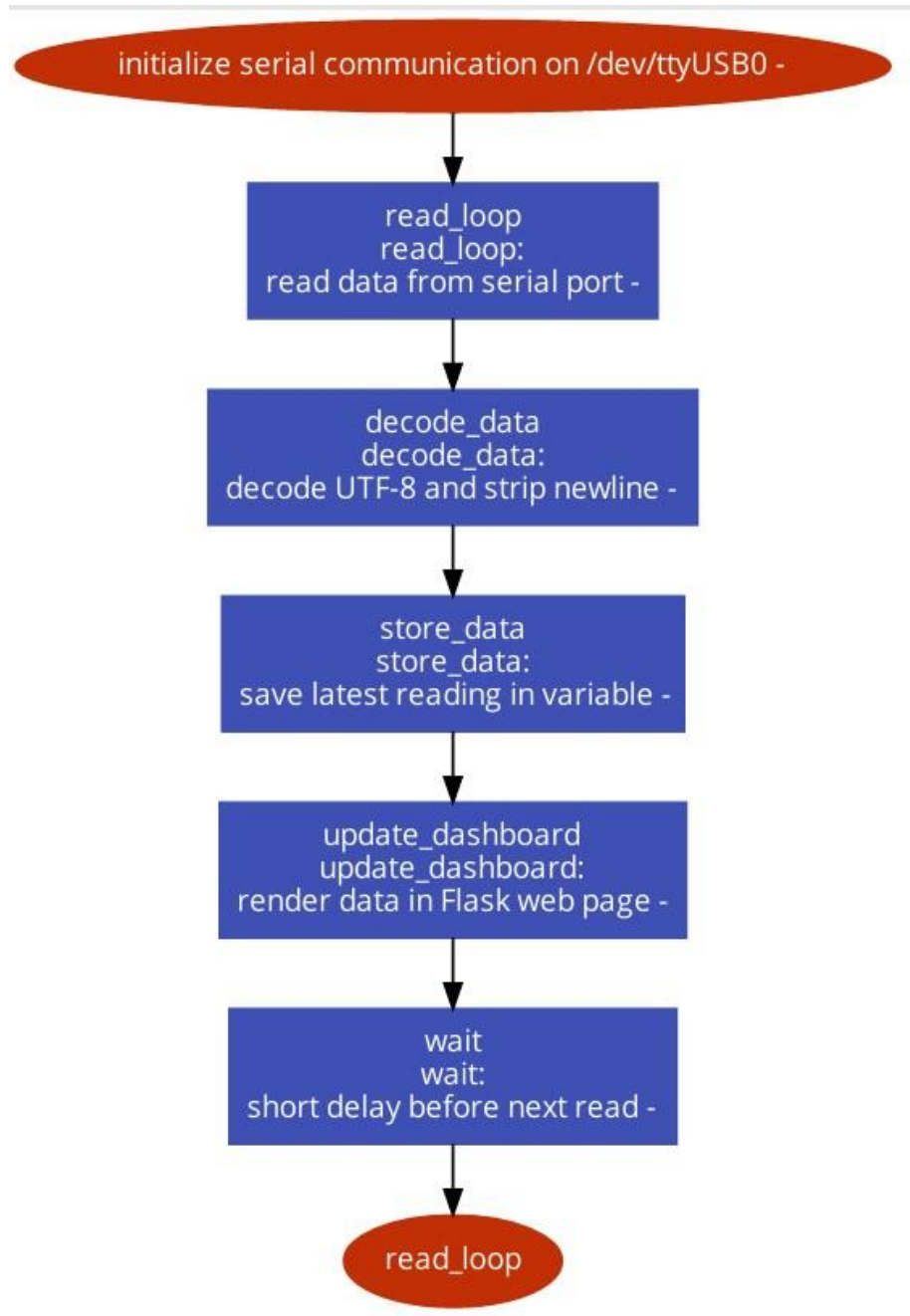
```
t.start() app.run(host='0.0.0.0',  
port=5000)
```

### **Light Meter Sensor**

```
#include <Wire.h>  
#include <BH1750.h>  
BH1750 lightMeter; void  
setup() {  
Serial.begin(9600);  
Wire.begin(); // No need to manually specify SDA/SCL in Mega if  
(lightMeter.begin(BH1750::CONTINUOUS_HIGH_RES_MODE)) {  
Serial.println("Light sensor booted successfully");  
}  
else {  
Serial.println("Failed to connect to sensor. Check wiring.");  
}  
}  
void loop() {  
float lux = lightMeter.readLightLevel();  
Serial.print("Light intensity: ");  
Serial.print(lux);  
Serial.println("lux"); delay(1000);  
}
```

## Software mechanism:

### 1. Raspberry Pi Zero W

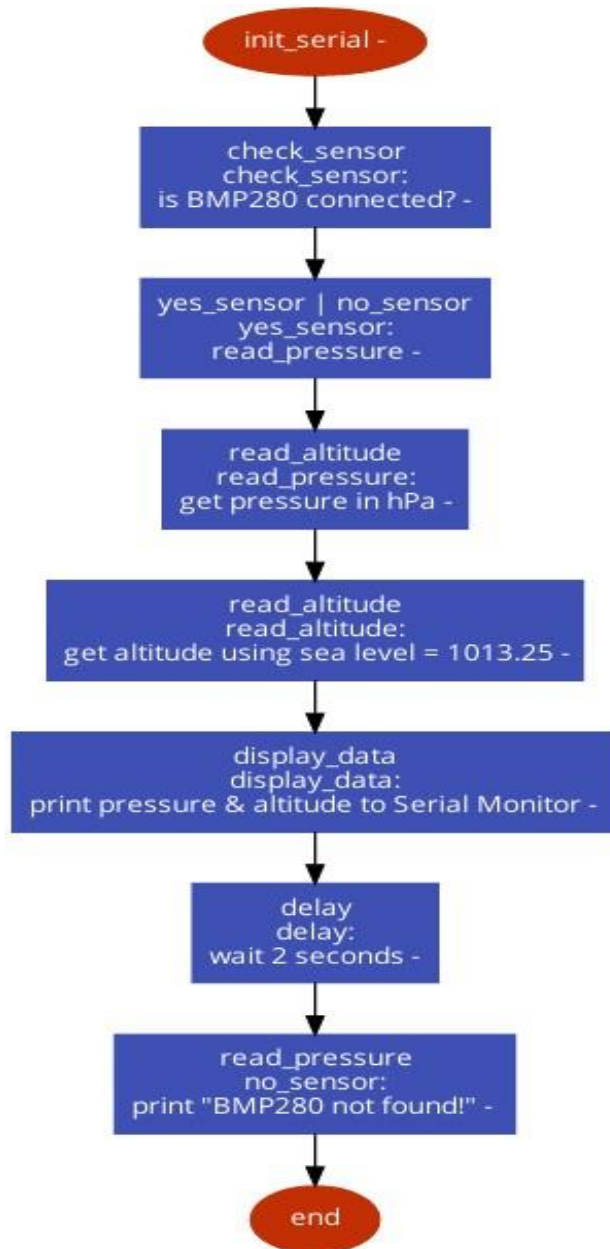


Figuer(4.Q)

### Serial Data Handling Process (Smart Weather Station)

No.	Process Name	Main Function / Additional Details
1	Initialize Serial Communication	Initialize serial connection Open the correct port (/dev/ttyUSB0) and set the appropriate baud rate for receiving data
2	read_loop	Continuous read loop Continuously read raw data from the serial port
3	decode_data	Decode the data Convert the data from UTF-8 to text and remove line-end characters like \n
4	store_data	Store the data Save the latest reading in a variable (or structured data)
5	update_dashboard	Update the dashboard Send the stored data to a Flask web page for display
6	wait	Short wait Add a short delay (e.g., time.sleep()) before the next read cycle
7	read_loop (repeat)	Repeat the loop Go back to step 2 and keep looping continuously

## 2. BMP280 Sensor (Pressure and Altitude)



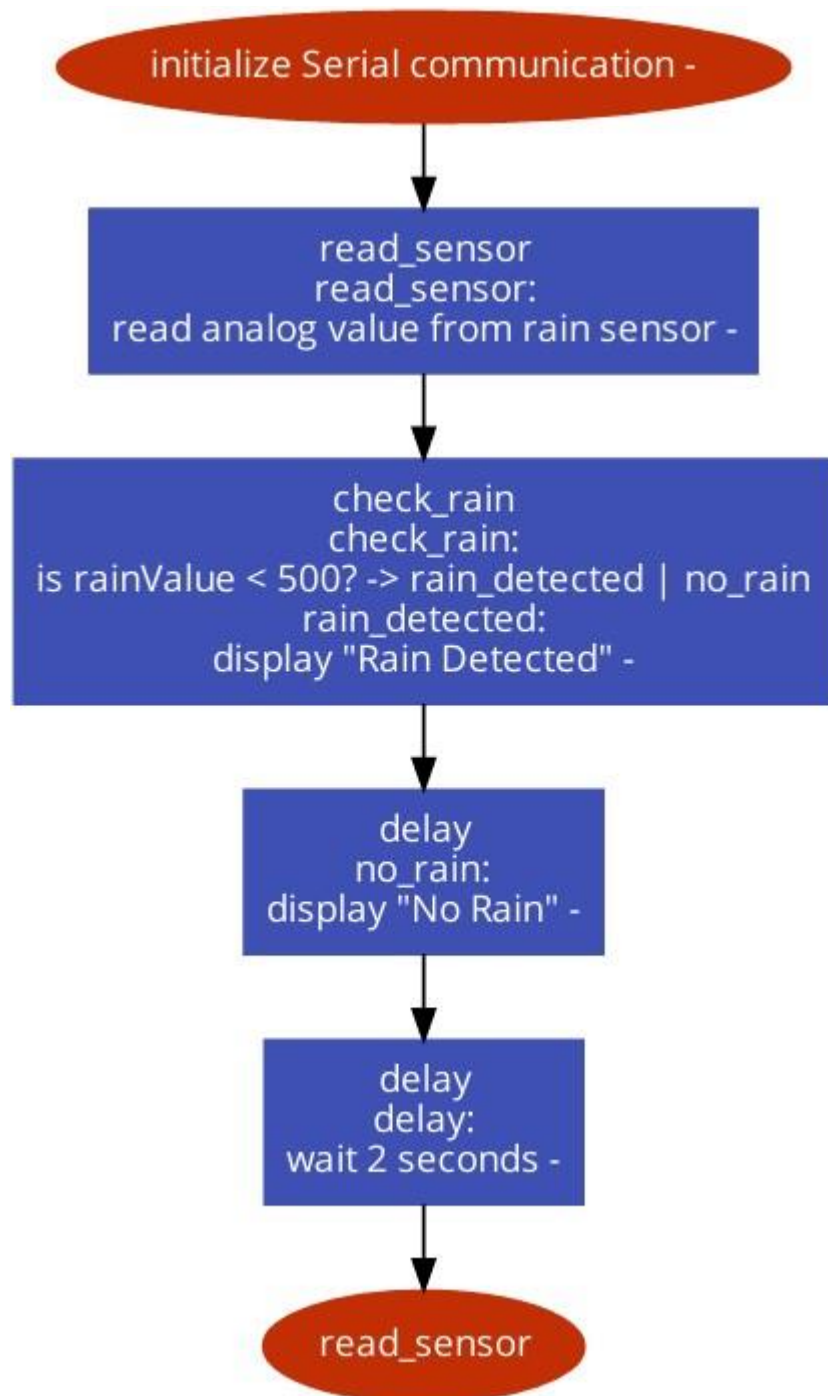
Figuer(4.R)

### BMP280 Sensor – Pressure and Altitude Reading Steps

No.	Process Name	Main Function	Additional Details
-----	--------------	---------------	--------------------

1	init_serial	Initialize serial communication	The program starts by initializing serial communication using Serial.begin() to display results
2	check_sensor	Check if sensor is connected	Verifies whether the BMP280 sensor is properly connected
3	yes_sensor	Read pressure	If the sensor is connected, pressure is read in units of hPa
4	read_altitude	Calculate altitude from pressure	Altitude is calculated using pressure and sealevel pressure formula: 1013.25 hPa
5	display_data	Print pressure and altitude	Pressure and altitude are printed to the Serial Monitor
6	delay	Wait	A short delay is added before a new reading cycle
7	no_sensor	Print sensor not found	If the sensor is not found, message "BMP280 not found!" is printed to screen
8	end	End of program	The loop ends or resets depending on the design

### 3. MH-RD Rain drop Sensor Module

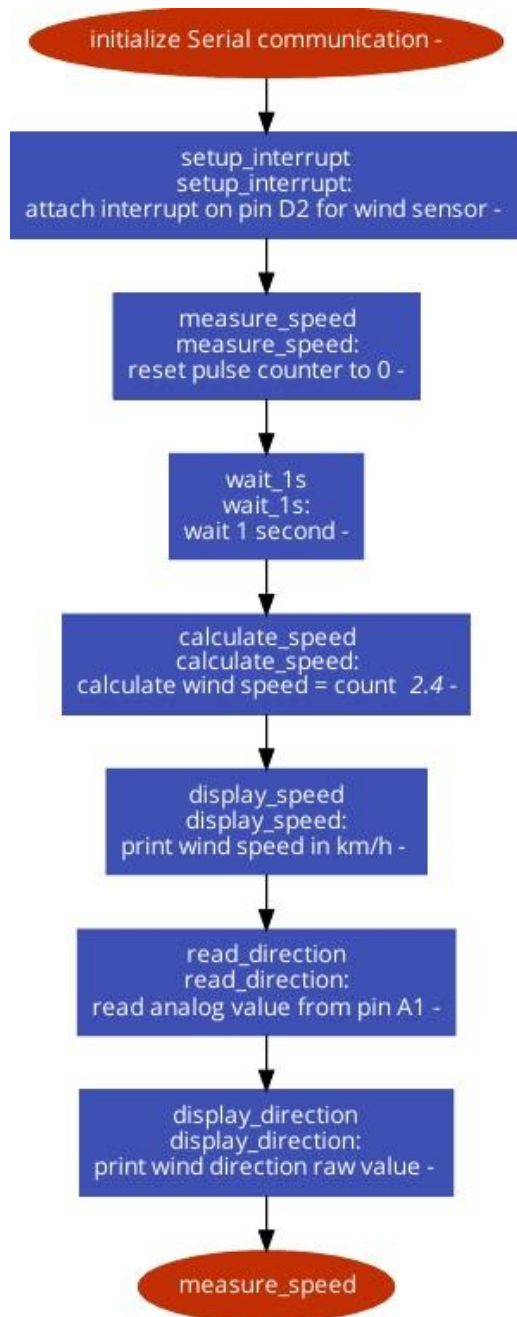


Figuer(4.S)

### Rain Sensor Reading – Step-by-Step Explanation

No.	Step Description	Code Name	Function
1	Start serial communication	initialize Serial	Initialize serial connection to display data on the Serial Monitor.
2	Read from rain sensor	read_sensor	Read analog value from the rain sensor using analogRead.
3	Check rain condition	check_rain	Check if the value is less than 500: If yes → display "Rain Detected" If no → display "No Rain"
4	Delay when no rain	delay (no_rain)	Short delay to clearly show the "No Rain" message.
5	General delay	delay (2 seconds)	Wait for 2 seconds before reading the sensor again.
6	Repeat reading	read_sensor (loop)	Repeat previous steps continuously to monitor rain status.

### 3. Wind Speed and Direction Sensors\ Anemometer and WindVane



Figuer(4.T)

## Wind Speed and Direction – Step-by-Step Explanation

No.	Step Description	Code Name	Function
1	Start Serial	initialize Serial	Display connection initialization message on the Serial Monitor.
2	Setup Interrupt	setup_interrupt	Connect the wind sensor to pin D2 and set up the interrupt.
3	Reset Speed Counter	measure_speed	Reset the pulse counter variable to zero.
4	Wait One Second	wait_1s	Wait for 1 second to accumulate the number of pulses.
5	Calculate Speed	calculate_speed	Calculate wind speed using: $\text{Speed} = \text{Pulses} \times 2.4$
6	Display Speed	display_speed	Display wind speed in km/h on the Serial Monitor.
7	Read Direction	read_direction	Read analog value of wind direction from pin A1.
8	Display Direction	display_direction	Display the raw analog value representing wind direction.
9	Repeat Measurement	measure_speed	Repeat the speed measurement cycle again.

## 4. WIRELESS COMMUNICATION MODULE - 2.4GHz NRF24L01+PA+LNA SMA Wireless Transceiver Antenna

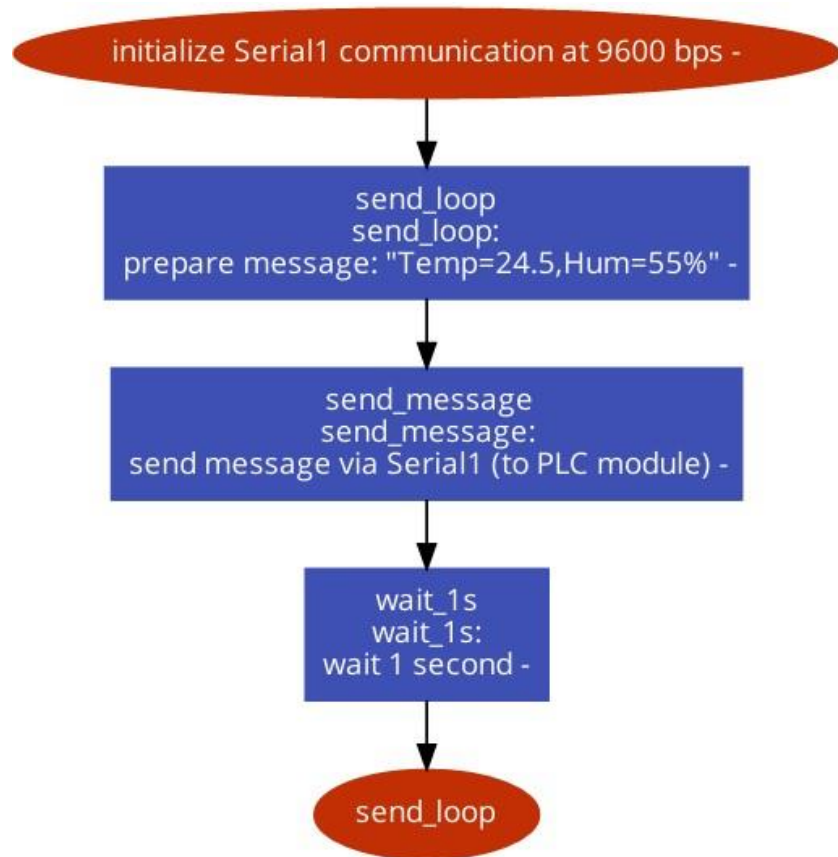


Figuer(4.U)

**Data transmission diagram using NRF24L01 explained:**

<b>No.</b>	<b>Process Name</b>	<b>Code Name</b>	<b>Function</b>
<b>1</b>	Start Serial Communication	initialize Serial	Start the serial connection to display messages on the computer.
<b>2</b>	Initialize Radio Module	init_radio	Start the NRF24L01 module.
<b>3</b>	Set Transmission Address	set_pipe	Set the transmission channel address to "00001".
<b>4</b>	Set Power Level	set_power	Set the transmission power level to <b>Low</b> .
<b>5</b>	Stop Listening (Set as Transmitter)	stop_listening	Switch the module to transmitter mode.
<b>6</b>	Prepare Message Loop	send_loop	Prepare the message content: "Hello from Weather Station".
<b>7</b>	Send Message	send_message	Transmit the message using the NRF24L01 module.
<b>8</b>	Wait One Second	wait_1s	Wait for 1 second before sending again.
<b>9</b>	Repeat Send Loop	send_loop	Repeat the loop to prepare and send the message again.

**5. Power Line Communication (PLC) Module:**



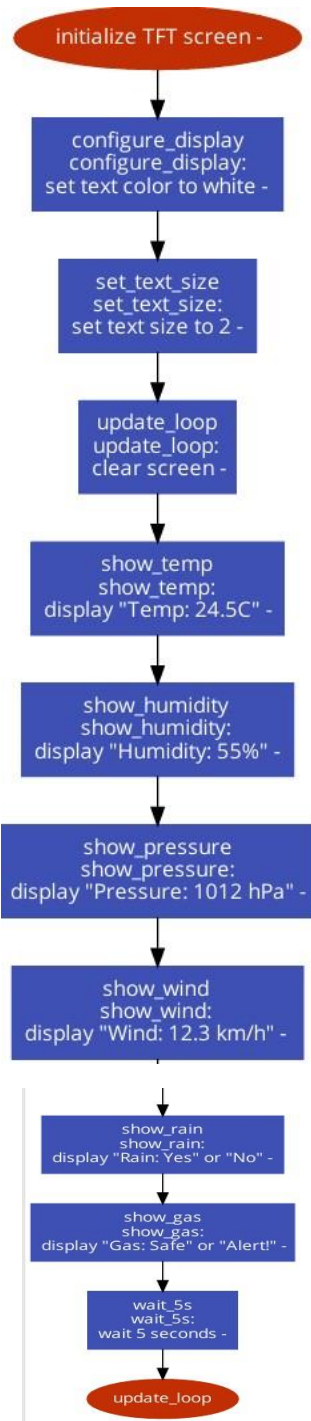
Figuer(4.V)

**Explanation of the diagram for sending data to the PLC unit via Serial1:**

No.	Step Description	Code Name	Function
-----	------------------	-----------	----------

1	Start Serial1 Connection	initialize Serial1	Start serial communication on Serial1 at 9600 bps to connect with the PLC.
2	Prepare Message	send_loop	Prepare the message to be sent, e.g.: "Temp=24.5,Hum=55%".
3	Send Message	send_message	Send the message to the PLC via the Serial1 port.
4	Wait One Second	wait_1s	Wait 1 second before repeating the sending process.
5	Repeat Send Loop	send_loop	Repeat the loop to prepare and send the message again.

**6. -ICD screen**

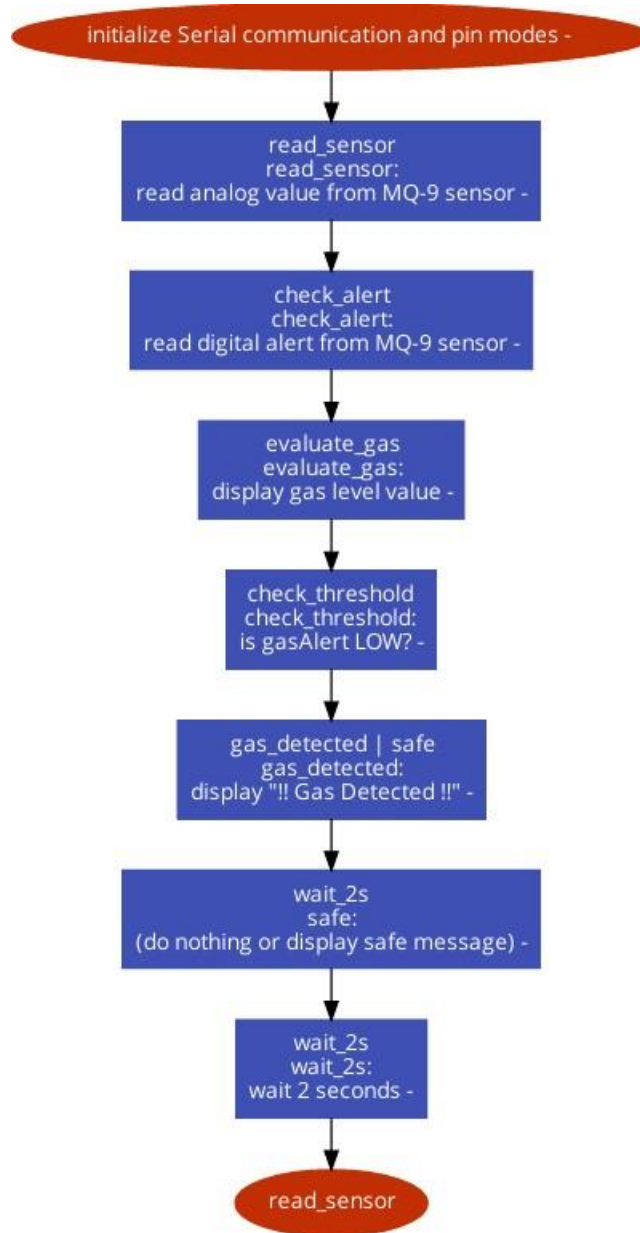


Figuer(4.W)

**Explanation of the data display diagram on a TFT screen:**

<b>No.</b>	<b>Step Description</b>	<b>Code Name</b>	<b>Function</b>
<b>1</b>	Initialize TFT Screen	initialize TFT	Power on the TFT screen and start initialization.
<b>2</b>	Set Text Color	configure_display	Set text color to white for display.
<b>3</b>	Set Text Size	set_text_size	Set font size to 2 for better readability.
<b>4</b>	Update Display	update_loop	Clear the screen and prepare it to display new data.
<b>5</b>	Show Temperature	show_temp	Display "Temp: 24.5C" on the screen.
<b>6</b>	Show Humidity	show_humidity	Display "Humidity: 55%" on the screen.
<b>7</b>	Show Pressure	show_pressure	Display "Pressure: 1012 hPa" on the screen.
<b>8</b>	Show Wind Speed	show_wind	Display "Wind: 12.3 km/h" on the screen.
<b>9</b>	Show Rain Status	show_rain	Display "Rain: Yes" or "No" based on rain detection.
<b>10</b>	Show Gas Condition	show_gas	Display "Gas: Safe" or "Alert!" depending on gas level.
<b>11</b>	Wait 5 Seconds	wait_5s	Wait for 5 seconds before updating the screen again.
<b>12</b>	Refresh Update	update_loop	Loop back to refresh and display updated data.

## 7. MQ-9 – Gas Sensor Module

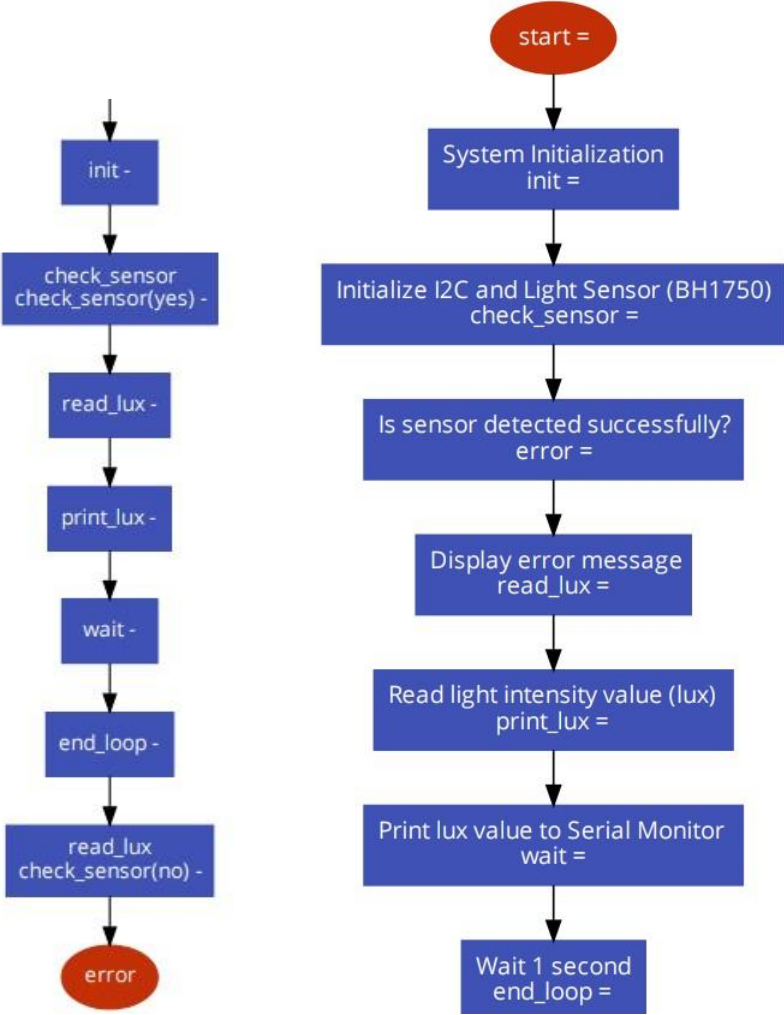


Figuer(4.X)

### MQ-9 gas sensor reading and analysis diagram explained:

No.	Step Description	Code Name	Function
<b>1</b>	Initialize Serial & Pins	initialize Serial & pins	Start serial communication, set up pins, and prepare the sensor for reading.
<b>2</b>	Read Sensor	read_sensor	Read the analog value from the MQ9 gas sensor.
<b>3</b>	Read Digital Alert Pin	read_sensor	Read the digital pin of MQ-9 to check gas presence alert.
<b>4</b>	Evaluate Gas Level	evaluate_gas	Use the analog value to display the gas concentration level.
<b>5</b>	Check Threshold	check_threshold	Check if the analog value is below the LOW threshold (danger level).
<b>6</b>	Gas Detected	gas_detected	→ If gas is detected, display "!! Gas Detected !!" alert.
<b>7</b>	Safe State	safe	→ If no gas is detected, display "Safe" and take no further action.
<b>8</b>	Wait 2s (Safe)	wait_2s (safe)	Wait 2 seconds in case of safe status.
<b>9</b>	General Delay	wait_2s	Wait 2 seconds before repeating the reading.
<b>10</b>	Repeat Sensor Reading	read_sensor	Repeat reading cycle from the sensor.

# 8.Light Meter Sensor



Figuer(4.Y)

### Light intensity reading chart explanation:

<b>No.</b>	<b>Operation</b>	<b>Symbolic Name</b>	<b>Function</b>
<b>1</b>	Program Start	start	The program starts execution.
<b>2</b>	System Initialization	init	Initializes the system and loads the required libraries.
<b>3</b>	Sensor Connection & Initialization	check_sensor	Initializes and checks communication via I2C protocol with BH1750 sensor.
<b>4</b>	Error Check	error	Detects if there's a message indicating a computation or connection error.
<b>5</b>	Error Message Display	read_lux	Fails to read and display from the light sensor (BH1750).
<b>6</b>	Light Intensity Reading	print_lux	Reads and prints the light intensity value in lux.
<b>7</b>	Display Value	wait	Displays the value on the Serial Monitor screen.
<b>8</b>	Wait Loop	end_loop	Waits for one second before rechecking the sensor.
<b>9</b>	End or Retry Faulty Reading	check_sensor(yes/no)	Detects if reading should be repeated or halted due to persistent error.

#### **4.4 PLC (Power Line Communication)**

PLC technology is the innovative cornerstone of this project, setting it apart from other similar projects. Instead of using only wireless networks, climate data is transmitted via conventional electrical wiring already present in the area.

##### **Advantages of using PLC:**

- Eliminates the need for Wi-Fi coverage or LoRa network extension.
- Lower cost compared to long-range wireless solutions.
- Can be operated inside buildings or agricultural facilities without establishing a separate network.
- Transmission stability in closed industrial or rural environments.

##### **Challenges of PLC technology:**

1. Interference from other electrical devices.
2. The need for compatibility between the transmitter and receiver.
3. The need to improve signal quality to avoid data loss.

##### **4.5 Project implementation stages :**

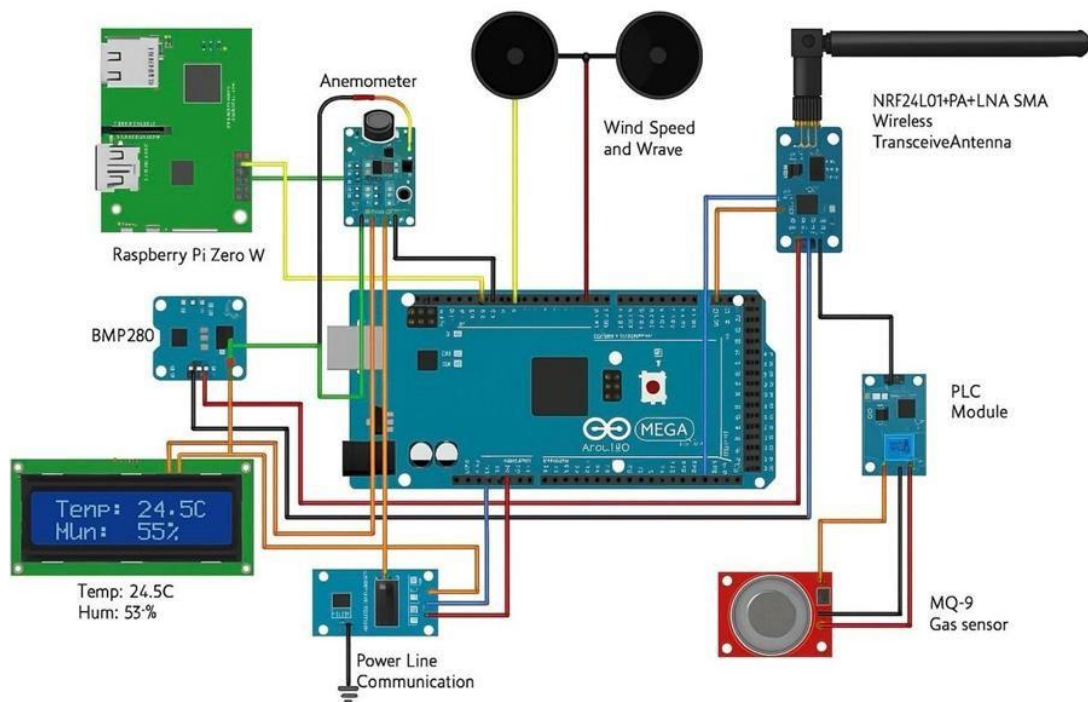
1. Design a pilot circuit on a breadboard.
2. Program the sensors and control unit and connect them to the communication unit.
3. Test each unit individually to ensure accurate measurements and transmission.
4. Integrate the wireless communication and PLC units and test the conversion between them.
5. Build a dashboard to visually display the data.

6. Complete environmental testing of the system in a realistic agricultural or outdoor environment.

### Complete wiring diagram of smart weather station components A

wiring diagram showing how to connect:

- All sensors and modules to the Arduino Mega □
- Communication line to the Raspberry Pi



Figuer(4.Z) Complete wiring diagram of smart weather station components

## **Steps to Connect Components to the Arduino Mega**

### **1. Connect the Temperature, Pressure, and Humidity Sensor (BME280)**

- Type: I2C
- Connections:
  - VCC → 3.3V or 5V (depending on the sensor type)
  - GND → GND
  - SDA → A4 (or Pin 20 on the Mega)
  - SCL → A5 (or Pin 21 on the Mega)
  - \_\_\_\_\_

### **2. Rain Sensor (Rain Sensor MH-RD)**

- Type: Analog or Digital
- Connections:
  - VCC → 5V
  - GND → GND
  - A0 or D0 → any analog input (for example, A0) or digital (if using it as a switch)
  - \_\_\_\_\_

### **3. Wind Speed and Direction Sensor**

- Type: Analog or Pulse (depending on the model)
- Connections:
  - VCC → 5V
  - GND → GND
  - Signal → Input Analog (e.g., A1 and above)
  - \_\_\_\_\_

### **4. Gas Sensor (MQ-9)**

- Type: Analog or Digital
- Connections:
  - VCC → 5V
  - GND → GND
  - A0 or D0 → Analog Input (e.g., A2)
  - \_\_\_\_\_

## 5. Light Meter Sensor

- Type: I2C (e.g., TSL2561 or BH1750)
- Connections:
  - VCC → 3.3V or 5V (depending on the type)
  - GND → GND
  - SDA → Pin 20 (SDA)
  - SCL → Pin 21 (SCL)

---

## 6. LCD Display (e.g., I2C 16x2 or 1280x400)

- Type: I2C or SPI •

Connections (if I2C):

- VCC → 5V
- GND → GND
- SDA → Pin 20 □ SCL → Pin 21

---

## 7. Wireless Communication Module (NRF24L01)

- Type: SPI •

Connections:

- VCC → 3.3V (⚠ □ Do not connect it to 5V as it will be damaged)
- GND → GND
- CE → any digital pin (e.g., 7)
- CSN → digital pin (e.g., 8)
- SCK → Pin 52
- MOSI → Pin 51
- MISO → Pin 50
- \_\_\_\_\_

---

## 8. PLC (Power Line Communication Module)

- Depends on the type (usually UART):

• Connections:

- VCC → 5V
- GND → GND
- TX → RX on Arduino (e.g., Pin 18)

- RX → TX on Arduino (e.g., Pin 19)

## **7. Raspberry Pi Zero W**

- Connect via UART: TX from Arduino → RX on Raspberry Pi and RX → TX
- Or use Wi-Fi or the NRF24L01+ module for wireless communication

## **Chapter Five: Data Analysis and Results**

### 5.1 Reading and analyzing climate data:

The smart station in this project was developed to collect and analyze multiple climate data in real time, with a focus on providing accurate data to support the following areas:

- Smart agriculture
- Solar energy management
- Weather forecasting
- Environment and local climate
- Components used in measurement and analysis:

<b>Sensor</b>	<b>Function</b>
Atmospheric Pressure Sensor	Measures atmospheric pressure in <b>hPa</b> units
Temperature Sensor	Measures air temperature
Humidity Sensor	Calculates the ratio of water vapor in the air
Wind Speed and Direction Sensor	Monitors wind intensity and direction
Light/UV Radiation Sensor	Measures the intensity of sunlight or ultraviolet

#### **Mechanism of action:**

- The Arduino Mega module collects sensor readings in real time.
- The data is processed internally and converted into a transmittable format .
- The data is sent either via:
  1. Wi-Fi to a Raspberry Pi or a cloud server.

2. Or via the PLC via electrical wiring to the processing unit, without the need for wireless networks.

**5.2 Results based on technical analysis of the system    1. Accuracy of atmospheric measurements:**

The performance of the selected sensors was analyzed based on technical data(Datasheet), and the results were summarized that the level of accuracy ranges as follows:

Table:

**Accuracy rate of climate variables:**

<b>Variable</b>	<b>Accuracy Rate</b>
Temperature	$\pm 0.5$ °C
Relative Humidity	$\pm 3\%$ RH
Atmospheric Pressure	$\pm 1$ hPa
Wind Speed	$\pm 0.5$ m/s

**MQ-9 Gas Sensor Accuracy:**

<b>Detected Gas</b>	<b>Typical Detection Range (ppm)</b>	<b>Approximate Accuracy</b>
Carbon Monoxide (CO)	10 – 1000 ppm	$\pm 10\%$ – $\pm 20\%$
Methane (CH <sub>4</sub> )	200 – 10000 ppm	$\pm 10\%$ – $\pm 20\%$

### **Sensor Results by Season – Yearly Analysis:**

This section presents the results obtained from the smart weather station sensors in Palestine over a full year, categorized by the four seasons. Key climate data was recorded monthly and analyzed to extract average values per season

<b>Parameter</b>	<b>Average / Status</b>
Temperature	23.2 °C
Relative Humidity	55%
Air Pressure	1012 hPa
Wind Speed	2.8 m/s
Solar Radiation	Moderate
Rainfall	Light Rain

### **Summer (June - August):**

<b>Parameter</b>	<b>Average / Status</b>
Temperature	32.7 °C
Relative Humidity	41%
Air Pressure	1005 hPa
Wind Speed	2.0 m/s
Solar Radiation	High
Rainfall	Rare

**Autumn (September - November):**

<b>Parameter</b>	<b>Average / Status</b>
Temperature	24.1 °C
Relative Humidity	58%
Air Pressure	1011 hPa
Wind Speed	3.1 m/s
Solar Radiation	Moderate
Rainfall	Intermittent

**Winter (December - February):**

<b>Parameter</b>	<b>Average / Status</b>
Temperature	11.2 °C
Relative Humidity	72%
Air Pressure	1019 hPa
Wind Speed	4.2 m/s
Solar Radiation	Low
Rainfall	Heavy and Regular

**These values are highly suitable for environmental and agricultural applications, and are considered highly accurate compared to the requirements of non-industrial use.**

**2. Wireless connection efficiency (Wi-Fi):** When using a module such as the ESP32 or Raspberry Pi:

- Transmission speed up to 150 Kbps.
- Transmission interval: approximately every 60 seconds.
- Rely on stable Wi-Fi coverage.

This solution is suitable for environments where wireless internet is available, such as cities or advanced facilities.

**3. Power Line Communication (PLC) efficiency:**

- Transmission speeds up to 200 Kbps.
- Coverage range: up to 100 meters indoors.

- Can transmit data stably over electrical wires, even in the absence of wireless networks.

A very important feature that makes the project suitable for remote agricultural or industrial environments, and reduces the need for additional infrastructure.

#### **4. Response time and data display:**

- Reading time for all sensors: less than 1 second.
- Data processing and display on the LCD or Raspberry Pi is almost instantaneous.
- The graphical interface is updated periodically every minute or on demand.

### **5.3 Technical conclusions:**

#### **1. Flexibility of communication:**

The combination of Wi-Fi and PLC technologies allows the system to be used in different locations depending on the infrastructure availability.

### **2. Reliability in challenging environments:**

The system can operate efficiently in rural or agricultural areas where WiFi coverage is not available, thanks to Power Line Communication (PLC) technology.

### **3. Ease of use:**

Through an LCD screen or a graphical interface on the Raspberry Pi, data can be displayed directly and clearly even to non-expert users, such as:

- Farmers
- Emergency teams
- Field workers

### **4. Next stage:**

Real-world field tests will be conducted during Graduation Project 2 (Phase 2) to confirm:

- Measurement accuracy
- Communication efficiency in different environments
- The system's ability to operate continuously and for a long time

## **Chapter Six: Types of Smart Weather Stations**

Smart weather stations have evolved into various forms, varying in components, accuracy, cost, and intended use. These stations can be classified into three main types:

## **6.1 Personal Weather Stations**

### **Description:**

They are miniature stations designed for individual or home use, often intended for hobbyists or regular users to monitor the weather conditions in their surroundings.

### **Features:**

- Sold as ready-made units or manually installed.
- Typically measures temperature, humidity, wind speed, and rainfall.
- Relies on wireless connectivity (Wi-Fi or Bluetooth) to send data to phone or computer applications.
- Data is stored locally or on simple cloud services.

### **Limitations:**

- Limited coverage.
- Lower accuracy compared to professional stations.
- Typically not compatible with other smart systems such as irrigation or cooling.

### **Uses:**

- Personal and home use.
- Educational projects.
- Weather monitoring in gardens or villas.

## **6.2 Professional Weather Stations**

**Description:**

These are high-precision stations used by institutions, research centers, or government agencies to monitor weather reliably and professionally.

**Features:**

- Contains precise, pre-calibrated sensors.
- Includes advanced data such as UV, evaporation, and environmental noise.
- Uses powerful transmission modules (such as LoRa, GSM, or PLC).
- Equipped with advanced central processing units and digital displays.
- Supports connectivity to cloud servers and advanced data analysis.

**Limitations:**

- High cost.
- Requires periodic maintenance and continuous calibration.

**Uses:**

- National meteorological stations.
- Research projects.
- Airports, ports, and major industrial areas.

**6.3 Agricultural and Industrial Weather Stations**

**Description:**

They are weather stations specifically designed for use in agricultural or industrial environments, and are often part of an integrated smart ecosystem.

**Features:**

Contains specialized sensors such as:

- Soil moisture
- Light intensity
- Carbon dioxide in the air
- Can be integrated with irrigation, cooling, and environmental control systems
- Uses flexible communication modules depending on the environment: Wi-Fi, LoRa, or PLC
- Low energy consumption and can be operated using solar energy.

**Limitations:**

- May be limited to functions outside the agricultural or industrial sector.
- Requires preparation and training for proper use.

**Uses:**

- Smart farms.
- Greenhouses.
- Factories and plants sensitive to heat or humidity.

**Types of irrigation stations:**

<b>Type</b>	<b>Accuracy Level</b>	<b>Connectivity Type</b>	<b>Cost</b>	<b>Ease of Installation</b>	<b>Applications</b>
Personal	Medium	Wi-Fi / Bluetooth	Low	High	Home and Educational
Professional	Very High	LoRa / GSM / PLC	High	Medium	National and Research-based
Agricultural/Industrial	High and Specialized	Wi-Fi / PLC / LoRa	Medium to High	Medium	Farms, Factories

**Our project falls within these categories:**

Our project falls under the third category (agricultural and industrial), as it:

- Supports sensors dedicated to the agricultural environment (soil moisture, rain, light).
- Relies on flexible communication technologies, most notably data transmission over power lines (PLC).
- Can be expanded and integrated with smart irrigation or monitoring systems in the future.

**Chapter Seven: How to Choose a Smart Weather Station**

Choosing a smart weather station is an important technical decision that directly affects the quality of the collected climate data and its suitability for the intended use—whether domestic, agricultural, industrial, or research-related. This choice depends on several key criteria that must be carefully evaluated to ensure the desired performance, accuracy, and efficiency.

## **7.1 Basic Technical Criteria for Choosing a Weather Station**

### **First: Nature of Use**

The type of use determines the shape and design of the required station. Home use differs from agricultural or industrial use. From this, the following questions arise:

- Is the purpose of the station personal monitoring of weather conditions?
- Or to feed an intelligent irrigation system with environmental data?
- Or to collect precise data for scientific analysis or disaster prediction?

### **Second: Type of Sensors and Required Data**

It is important to accurately determine the types of data the station should provide. The basic options include:

- Ambient Temperature
- Relative Humidity
- Barometric Pressure
- Wind Speed and Direction
- Rainfall Rate
- Light Intensity or Solar Radiation □ Air Quality (e.g., CO<sub>2</sub>, PM2.5)

The greater the variety of data types, the more critical the accuracy of the sensors used becomes, and the more complex the system grows.

### **Third: Measurement Accuracy and Sensor Calibration**

Accuracy is a decisive factor in professional or research applications. The following should be considered:

- Permissible margin of error ( $\pm\%$ )
- Type of output: analog or digital
- Support for manual or automatic calibration
- Certification of industrial or environmental standards  
(e.g., ISO or CE)

### **Fourth: Means of communication and data transfer**

The connection type determines how and where data is sent. The most prominent options are:

- Wi-Fi: Suitable for urban environments but requires coverage.
- LoRa: Excellent for rural or remote areas, offering wide range and low power consumption.
- Power Line Communication (PLC): The focus of our project. It allows data transmission via the power grid without the need for a separate network infrastructure, making it ideal for enclosed areas or regions with weak wireless coverage.

### **Fifth: Expandability and Integration**

Can new sensors be added?

Is the station capable of integration with other smart systems like irrigation, cooling, or forecasting systems?

Future expandability is one of the most important indicators of the station's "sustainability" and long-term technical value.

### **Sixth: Data display and storage method**

#### **Options include:**

- Built-in LCD display.
- Custom mobile app.
- Web dashboard.
- Local or cloud server.

It is important to ensure that the data can be easily accessed and analyzed, with reference to recorded climate history.

## 7.2 Cost-Performance Analysis

Higher price does not necessarily mean better performance. A careful comparison is required between:

- Cost of parts and components.
- Performance efficiency and data accuracy.
- Maintenance, upgrades, and technical support.

In our current project, we achieved an excellent balance between cost and performance by using reliable measurement units and dual communication modules (Wi-Fi + PLC). This enables the project to compete with expensive commercial stations without compromising quality.

## 7.3 Smart Weather Station Components Cost (USD)

<b>(Component)</b>	<b>(Quantity)</b>	<b>(USD)</b>	<b>(USD)</b>
<b>Light Meter Sensor</b>	1	34.79	34.79
<b>Raspberry Pi Zero</b>	1	15.22	15.22
<b>Wind Speed &amp; Direction Sensor</b>	1	18.00	18.00
<b>PLC Communication Module</b>	2	26.48	52.96

<b>LCD Display 400×1280</b>	1	39.00	39.00
<b>Arduino Mega</b>	1	21.33	21.33
<b>BME280 Sensor</b>	1	21.33	21.33
<b>Rain Drop Sensor</b>	1	5.13	5.13
<b>Soil Moisture Sensor</b>	1	6.21	6.21
<b>Air Quality Sensor</b>	1	6.21	6.21
<b>Wireless Communication Module</b>	2	9.45	18.90
<b>Solar Panel 10W</b>	1	14.85	14.85
<b>Solar Charge Controller</b>	1	10.80	10.80
<b>Li-ion Battery 18650</b>	2	4.86	9.72
<b>Voltage Regulator /LM2596</b>	1	2.70	2.70
<b>Power Supply</b>	1	30.00	30.00

**Total price: \$306.46**

#### **7.4 Technical summary for selecting the optimal station**

<b>Criterion</b>	<b>Low-Cost Station</b>	<b>Professional Smart Station</b>	<b>Project Station</b>
Measurement Accuracy	Low to Medium	Very High	High

Number of Sensors	Limited	Extended	Flexible and Customizable
Communication Type	Wi-Fi / Bluetooth	GSM / LoRa	Wi-Fi + PLC
Power Supply	Electric Only	Solar + Battery	Electricity + Expandable
Display Interface	Simple App	Web + Built-in Screens	LCD Screen + Web Interface
Smart Integration	No	yas	Yes (Open Design)
Price	Low	High	Moderate – Smart

## **Chapter Eight: Installation and Maintenance of the Smart Weather Station**

Installing the smart weather station is a pivotal stage in the success of the project. This process requires precise mounting, engineering planning for sensor distribution, and proper programming and communication settings. Additionally, regular maintenance is essential to ensure continuous operation and data quality. In this

chapter, we will present in detail the installation steps, operational setup, and principles of optimal maintenance.

## **8.1 Installation and Connection Steps**

### **First: Site Selection**

The accurate performance of sensors depends on selecting the appropriate location:

- **The wind sensor** should be installed in an open area at a height of no less than 3–5 meters to avoid obstructions.
- **The rain sensor** needs an open, unshaded surface.
- **The light sensor** should be oriented toward the sun without any obstructions.
- **The temperature and humidity sensor** should be placed in a shaded, well-ventilated area to avoid the effects of direct radiation.
- **The soil sensor** should be installed below the surface at a suitable depth depending on the type of plant or study.

### **Second: Electrical Connections**

- Connect the sensors to the control unit (Arduino or a similar board) through analog or digital input pins.
- Ensure stable power sources are used (5V or 3.3V depending on the sensor).
- Connect the PLC unit to the main power line through safe and well-planned points.

### **Third: Installing the Processing and Communication Unit**

- Mount the Raspberry Pi or Arduino inside a sealed enclosure to protect it from dust and humidity.

- Connect the Wi-Fi module via the local network credentials or a hotspot.
- Link the PLC unit to the appropriate port in the internal power network, ensuring proper grounding and signal isolation.

## **8.2 Configuring Communication and Integration Settings**

### **Wireless Communication (Wi-Fi or LoRa):**

- Set the SSID and password for network access.
- Assign a static IP address (optional) to simplify access to the display interface.
- Test data transmission to a local server or open cloud platform (e.g., Thing Speak or a dedicated platform).

### **Power Line Communication (PLC):**

- Ensure the transmitter and receiver units are on the same electrical circuit.
- Test signal quality using monitoring software (e.g., PLC Configurator).
- Assign a unique identifier to each unit to avoid data overlap.

### **Integration with the Display Interface:**

- Program the dashboard using Python or Node-RED.
- Configure a graphical interface to display real-time values.
- Include alerts in case of sudden changes (e.g., rainfall, temperature spike...).

## **8.3 Routine Maintenance and Troubleshooting**

### **First: Preventive Maintenance**

- Clean the sensors monthly from dust or contaminated rain. □  
Check calibration every 3 to 6 months using reference values.
- Inspect electrical connections to prevent corrosion or poor contact.
- Update firmware or the display interface when necessary.

### **Second: Corrective Maintenance**

- If data transmission stops, check the power source first.
- Inspect the control unit's input ports for wire damage or disconnection.
- Test the wireless signal or PLC unit to verify data transmission.

### **Third: Maintenance Log All maintenance and update activities should be documented in a regular log including:**

- Date of the procedure
- Type of modification or repair
- Affected sensor
- Specific notes on performance before and after repair

## **Chapter Summary:**

**The success of a smart weather station relies not only on design accuracy or sensor efficiency, but also requires:**

- Precise installation in a well-considered environment
- Professional-level software configuration
- Regular maintenance and proactive monitoring
- Smart management of PLC technology to avoid interference and ensure stable data transmission.

### **Chapter Nine: Building a Smart Weather Station (Practical Application)**

This chapter serves as a practical embodiment of the theoretical concepts discussed in the previous chapters. It explains how to actually build a smart weather station using readily available electronic components, as well as programming and assembly methods suitable for both educational and applied environments. The aim of this chapter is to document a comprehensive step-by-step process for constructing a prototype smart weather station, with a focus on the innovative use of Power Line Communication (PLC) as a distinctive feature of this project.

---

#### **9.1 Required Tools and Components**

Below is a detailed list of parts and components used in building the station, based on the actual project plan:

## System Components and Their Functions

<b>Component</b>	<b>Function</b>
<b>Arduino Mega</b>	Main control unit for reading sensor signals.
<b>Raspberry Pi Zero</b>	Used for displaying data or running the graphical user interface.
<b>Light Meter Sensor</b>	Measures light intensity.
<b>Wind Speed &amp; Direction Sensor</b>	Measures wind speed and direction (using a 4–20mA signal).
<b>Pressure, Temperature &amp; Humidity Sensor</b>	A multi-environmental sensor.
<b>Rain Drop Sensor</b>	Detects rainfall.
<b>Soil Moisture Sensor</b>	Measures soil moisture (important for smart agriculture).
<b>Air Quality Sensor</b>	Monitors pollutants and fine particles.
<b>LCD Display 400x1280</b>	Displays data locally.
<b>Wireless Communication Module</b>	Wi-Fi or LoRa transmission module.
<b>PLC Communication Module (2x)</b>	For transmitting data over the power grid.

## 9.2 Programming and Assembly Steps

### 1. Connecting Sensors to the Arduino Mega

- Each sensor is connected to its appropriate port (Analog or Digital).
- Protection resistors and electrical isolation are applied as needed.

### 2. Programming the Control Unit

- Use Arduino C language with dedicated libraries for each sensor.
- Define a periodic reading interval (e.g., every 60 seconds).
- Convert the read values into a unified format (JSON or CSV) for easy transmission.

### **3. Setting Up the Wireless Communication Unit**

- Enter the Wi-Fi network settings.
- Write the code to send data to a server or web interface.

### **4. Configuring the PLC Unit**

- Connect the transmitting end to the Arduino and the receiving end to the Raspberry Pi.
- Use a simple protocol (Serial or Modbus) to transfer data over the power line.
- Test signal quality using built-in PLC diagnostic tools.

### **5. Preparing the Display Interface**

- Use an LCD screen to directly display basic values.
- Or set up a graphical interface on the Raspberry Pi (using Python or Node-RED).

## **9.3 Initial System Testing**

**After assembling and powering on the system, initial tests should include:**

- Testing sensor accuracy and comparing readings with a trusted reference.
- Trying data transmission over both Wi-Fi and PLC to compare stability and speed.
- Simulating various environments (wet/dry/sunny) to observe sensor behavior.

#### 9.4 Expected Challenges and Proposed Solutions

Challenge	Cause	Solution
PLC signal fluctuation	Interference from electrical devices	Use signal filters and connect directly through a clean power outlet
Weak sensor readings	Humidity or poor wiring	Insulate wires and use well-secured sensor housings
Weak Wi-Fi coverage	Distance or wall barriers	Use PLC as an alternative channel or add a Wi-Fi extender

#### 9.5 Scalability and Future Expansion

- Add advanced sensors (e.g., UV sensor, gas sensor).
- Integrate the system with smart irrigation or cooling systems.
  - Support cloud storage and AI-powered data analysis.
- Enable a smartphone application to display real-time data.

#### Chapter Summary:

This project represents a practical and fully integrated model of a smart weather station. It is low-cost, flexible in design, and features the innovative use of PLC technology, which distinguishes it from conventional models.

## **Closing Word**

At the conclusion of this work, we pause for a moment of reflection and gratitude.

This experience has been more than just an academic project; it has been a cognitive and human journey that we embarked on with passion. During this journey, we faced intellectual and technical challenges, and strove with all our effort and willpower to present work worthy of the trust of our professors, the support of our families, and the cooperation of our colleagues.

We have learned that achievement is not born from an idea alone, but from belief in it, the continuous pursuit of its realization, and the ability to transform theory into reality and plans into tangible steps.

We thank everyone who has had an impact along the way, even with a word of encouragement or a sincere prayer. We promise this project will be the first step in an academic and professional journey that we aspire to be filled with giving, innovation, and responsibility.

To everyone who reads these pages...

Believe in your ideas and work for them, for they may be the nucleus of real change.

**Reference:**

**1. An IoT based Low-cost Weather Monitoring System for Smart Farming**

[https://www.researchgate.net/profile/Narmilan-Amarasingam/publication/355183495\\_An\\_IoT\\_based\\_Lowcost\\_Weather\\_Monitoring\\_System\\_for\\_Smart\\_Farming/links/621ec71919ab0c3b4d2ca1de/An-IoT-based-Low-cost-Weather-Monitoring-System-for-Smart-Farming.pdf?utm\\_source=chatgpt.com](https://www.researchgate.net/profile/Narmilan-Amarasingam/publication/355183495_An_IoT_based_Lowcost_Weather_Monitoring_System_for_Smart_Farming/links/621ec71919ab0c3b4d2ca1de/An-IoT-based-Low-cost-Weather-Monitoring-System-for-Smart-Farming.pdf?utm_source=chatgpt.com)

**2. This research paper presents a weather tracking system developed for agricultural purposes. The system utilizes various sensors and communication technologies to collect and analyze climate data in real time, providing farmers with accurate and timely information to help them make informed decisions about their crops.**

[https://www.irjmets.com/uploadedfiles/paper//issue\\_2\\_february\\_2023/33726/final/fin\\_irjmets1677164450.pdf?utm\\_source=.com](https://www.irjmets.com/uploadedfiles/paper//issue_2_february_2023/33726/final/fin_irjmets1677164450.pdf?utm_source=.com)

**3. These challenges serve as driving forces for the agricultural sector to shift toward smart farming by implementing IoT and big data solutions to enhance operational efficiency and productivity.**

[https://cdn.fsbx.com/v/t59.2708-21/491049725\\_717136587646714\\_3211874799064736835\\_n.pdf/Smart\\_remote\\_sensing\\_weather\\_station.pdf?\\_nc\\_cat=106&\\_nc\\_cb=47395efc-74c935b2&ccb=17&\\_nc\\_sid=2b0e22&\\_nc\\_ohc=Xj8fc0zLTbgQ7kNvwHCf8Oe&\\_nc\\_oc=AdlVVIkqPUET118kXi0Nq0j3N-SoKo2mY\\_6TGU9gyZLQDeV5k6o9XZ8MNxu2A7j47ow&\\_nc\\_zt=7&\\_nc\\_ht=cdn.fsbx.com&\\_nc\\_gid=a-48ZREIAEb6P1vPBri\\_Pw&oh=03\\_Q7cD2gEMFL\\_Te24T0TbwS8SMdEH0tDL-qMmMEjDdi68ZZx1N0Q&oe=6859322A&dl=1](https://cdn.fsbx.com/v/t59.2708-21/491049725_717136587646714_3211874799064736835_n.pdf/Smart_remote_sensing_weather_station.pdf?_nc_cat=106&_nc_cb=47395efc-74c935b2&ccb=17&_nc_sid=2b0e22&_nc_ohc=Xj8fc0zLTbgQ7kNvwHCf8Oe&_nc_oc=AdlVVIkqPUET118kXi0Nq0j3N-SoKo2mY_6TGU9gyZLQDeV5k6o9XZ8MNxu2A7j47ow&_nc_zt=7&_nc_ht=cdn.fsbx.com&_nc_gid=a-48ZREIAEb6P1vPBri_Pw&oh=03_Q7cD2gEMFL_Te24T0TbwS8SMdEH0tDL-qMmMEjDdi68ZZx1N0Q&oe=6859322A&dl=1)

**4. Weather Monitoring System using IoT for Smart Farming**

[https://www.researchgate.net/publication/362429748\\_Weather\\_Monitoring\\_System\\_using\\_IoT\\_for\\_Smart\\_Farming?utm\\_source=et.com](https://www.researchgate.net/publication/362429748_Weather_Monitoring_System_using_IoT_for_Smart_Farming?utm_source=et.com)

**5. IoT Applications in Smart Agriculture: Issues and Challenges**

[https://www.researchgate.net/publication/347833981\\_IoT\\_Applications\\_in\\_Smart\\_Agriculture\\_Issues\\_and\\_Challenges](https://www.researchgate.net/publication/347833981_IoT_Applications_in_Smart_Agriculture_Issues_and_Challenges)

**6. [https://cdn.fsbx.com/v/t59.2708-](https://cdn.fsbx.com/v/t59.2708-21/491049725_717136587646714_3211874799064736835_n.pdf/Smart_remote_sensing_weather_station.pdf?_nc_cat=106&_nc_cb=47395efc-74c935b2&ccb=17&_nc_sid=2b0e22&_nc_ohc=1Oes0HDfyYEQ7kNvwHvomAr&_nc_oc=AdlH5BoCq_u8G17i_WcxHFljcRbyYrRSIFg323uYpWC4_OO2nVnC4dnVbdiUK8MJTks&_nc_zt=7&_nc_ht=cdn.fsbx.com&_nc_gid=PMItrENrwlLTWiwRzqzQZw&oh=03_Q7cD2gFXTZxbmv8Ll7TDCTpxVuZeeTbBrzXkL2Ptp3EwYDwNA&oe=685C0D6A&dl=1)**

[21/491049725\\_717136587646714\\_3211874799064736835\\_n.pdf/Smart\\_remote\\_sensing\\_weather\\_station.pdf?\\_nc\\_cat=106&\\_nc\\_cb=47395efc-74c935b2&ccb=17&\\_nc\\_sid=2b0e22&\\_nc\\_ohc=1Oes0HDfyYEQ7kNvwHvomAr&\\_nc\\_oc=AdlH5BoCq\\_u8G17i\\_WcxHFljcRbyYrRSIFg323uYpWC4\\_OO2nVnC4dnVbdiUK8MJTks&\\_nc\\_zt=7&\\_nc\\_ht=cdn.fsbx.com&\\_nc\\_gid=PMItrENrwlLTWiwRzqzQZw&oh=03\\_Q7cD2gFXTZxbmv8Ll7TDCTpxVuZeeTbBrzXkL2Ptp3EwYDwNA&oe=685C0D6A&dl=1](https://cdn.fsbx.com/v/t59.2708-21/491049725_717136587646714_3211874799064736835_n.pdf/Smart_remote_sensing_weather_station.pdf?_nc_cat=106&_nc_cb=47395efc-74c935b2&ccb=17&_nc_sid=2b0e22&_nc_ohc=1Oes0HDfyYEQ7kNvwHvomAr&_nc_oc=AdlH5BoCq_u8G17i_WcxHFljcRbyYrRSIFg323uYpWC4_OO2nVnC4dnVbdiUK8MJTks&_nc_zt=7&_nc_ht=cdn.fsbx.com&_nc_gid=PMItrENrwlLTWiwRzqzQZw&oh=03_Q7cD2gFXTZxbmv8Ll7TDCTpxVuZeeTbBrzXkL2Ptp3EwYDwNA&oe=685C0D6A&dl=1)

**7. [https://cdn.fsbx.com/v/t59.2708-](https://cdn.fsbx.com/v/t59.2708-21/468576400_1989961344835068_8029665850213132434_n.pdf/IoTenabled_Smart_Weather_Stations_Innovations_Challenges.pdf?_nc_cat=104&_nc_cb=47395efc-74c935b2&ccb=17&_nc_sid=2b0e22&_nc_ohc=dhjsu51cTE4Q7kNvwEXvGdu&_nc_oc=AdmUAKlU3MEGO8kDplHcK3EDkLHQljEnuZkpa_3qm1MTm8KjL4yOkm0Z85OPXAoKh)**

[21/468576400\\_1989961344835068\\_8029665850213132434\\_n.pdf/IoTenabled\\_Smart\\_Weather\\_Stations\\_Innovations\\_Challenges.pdf?\\_nc\\_cat=104&\\_nc\\_cb=47395efc-74c935b2&ccb=17&\\_nc\\_sid=2b0e22&\\_nc\\_ohc=dhjsu51cTE4Q7kNvwEXvGdu&\\_nc\\_oc=AdmUAKlU3MEGO8kDplHcK3EDkLHQljEnuZkpa\\_3qm1MTm8KjL4yOkm0Z85OPXAoKh](https://cdn.fsbx.com/v/t59.2708-21/468576400_1989961344835068_8029665850213132434_n.pdf/IoTenabled_Smart_Weather_Stations_Innovations_Challenges.pdf?_nc_cat=104&_nc_cb=47395efc-74c935b2&ccb=17&_nc_sid=2b0e22&_nc_ohc=dhjsu51cTE4Q7kNvwEXvGdu&_nc_oc=AdmUAKlU3MEGO8kDplHcK3EDkLHQljEnuZkpa_3qm1MTm8KjL4yOkm0Z85OPXAoKh)

[nI&\\_nc\\_zt=7&\\_nc\\_ht=cdn.fbsbx.com&\\_nc\\_gid=VPjN4L5Vj-jrSXacFZ05g&oh=03\\_Q7cD2gHMZV4S8M9r3AhUJ6Ws-6OhY1GPEi-Iw06WOXVuUEuEVg&oe=685BE71F&dl=1](https://www.facebook.com/VPjN4L5Vj-jrSXacFZ05g&oh=03_Q7cD2gHMZV4S8M9r3AhUJ6Ws-6OhY1GPEi-Iw06WOXVuUEuEVg&oe=685BE71F&dl=1)

8. <https://www.electronicsforu.com/electronics-projects/weather-station-using-stm32>
9. <https://www.instructables.com/Building-a-Weather-Station-WithSTM32F103C8T6/>
10. <https://www.shutterstock.com/image-photo/weather-station-green-tea-field-5g2284312609>