A Major Project Report

on

IOT BASED SOLAR POWER MONITORING SYSTEM

Submitted in partial fulfilment of the requirements for the award of the degree

Of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

By

GUNDLAPALLY MOKSHITHA	**20EG105116**
KOTHI ARYAN REDDY	**20EG105126**
KONDI HARSHA VARDHAN REDDY	**20EG105703**
MAROJU SRIRAM	**20EG105714**

Under The Guidance Of

V. JYOTHI

Assistant Professor

Department of CSE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Venkatapur(V), Ghatkesar(M), Medchal(D)– 500088

2023-2024

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the Report entitled "**IOT BASED SOLAR POWER MONITORING SYSTEM**" that is being submitted by **Gundlapally Mokshitha [20EG105116], Kothi Aryan Reddy [20EG105126], Kondi Harsha Vardhan Reddy [20EG105703] and Maroju SriRam [20EG105714]** in partial fulfilment for the award of Bachelor of Technology in **COMPUTER SCIENCE AND ENGINEERING** to the Anurag University is a record of bonafide work carried out by them under my guidance and supervision from academic year 2023 to 2024. The results embodied in this report have not been submitted to any other University or Institute for the award of any degree or diploma

Internal Guide	**Dr. G. Vishnu Murthy**
V. JYOTHI	**Professor &Dean, Dept of CSE**
Assistant Professor, Dept of CSE	

External Examiner

ACKNOWLEDGMENT

We would like to express our sincere thanks and deep sense of gratitude to project supervisor **Mrs. V. Jyothi** for her constant encouragement and inspiring guidance without which this project could not have been completed. Her critical reviews and constructive comments improved our grasp of the subject and steered to the fruitful completion of the work. Her patience, guidance and encouragement made this project possible.

We would like acknowledge our sincere gratitude for the support extended by **Dr. G. Vishnu Murthy**, Dean, Dept. of CSE, Anurag University. We also express my deep sense of gratitude to **Dr. V V S S S Balaram**, Academic co-ordinator. **Dr. Pallam Ravi,** Project in-Charge, **Dr. G. Prabhakar Raju**, Project Co-ordinator and Project review committee members, whose research expertise and commitment to the highest standards continuously motivated us during the crucial stage our project work. We would like express my special thanks to **Dr. V. Vijaya Kumar**, Dean School of Engineering, Anurag University, for their encouragement and timely support in our B. Tech program.

G. MOKSHITHA

(20EG105116)

KOTHI ARYAN

(20EG105126)

K. HARSHAVARDHAN

(20EG105703)

M. SRIRAM

(20EG105714)

DECLARATION

We hereby declare that the Report entitled "**IOT BASED SOLAR POWER MONITORING SYSTEM**" submitted for the award of Bachelor of technology Degree is our original work and the Report has not formed the basis for the award of any degree, diploma, associate ship or fellowship of similar other titles. It has not been submitted to any other University or Institution for the award of any degree or diploma

G. MOKSHITHA
(20EG105116)

KOTHI ARYAN REDDY
(20EG105126)

K. HARSHA VARDHAN
(20EG105703)

M. SRIRAM
(20EG105714)

Place: Hyderabad
Date:

ABSTRACT

This project presents an IoT-based solar power monitoring system designed to optimize the efficiency and reliability of solar energy generation. Through the integration of IoT sensors, communication technologies, and data analytics, the system enables real-time monitoring and management of crucial parameters in solar power installations.

One of the key features of the system is its capability for remote monitoring and management. IoT sensors deployed throughout the solar power system continuously collect data on various parameters, including solar irradiance, temperature, panel orientation, and energy production. This data is then transmitted to a central monitoring platform hosted on a cloud infrastructure, allowing stakeholders to remotely monitor and manage the system from anywhere.

The system utilizes advanced data analytics algorithms to process the collected data and derive actionable insights. By analysing historical data and real-time information, the system dynamically adjusts parameters such as panel tilt angle and solar tracker positioning to optimize energy production and improve overall efficiency. This proactive optimization approach ensures maximum energy generation under varying environmental conditions.

In addition to optimizing energy production, the system also focuses on fault detection and diagnostics. The user interface of the system is designed to be intuitive and accessible. Stakeholders can access real-time performance data and receive alerts through a user-friendly interface accessible via web or mobile applications. This interface enables users to monitor energy production, track historical data, and generate reports for performance analysis and compliance.

Field trials and performance tests have been conducted to evaluate the effectiveness of the IoT-based solar power monitoring system. Results indicate significant improvements in energy efficiency, reduced downtime due to proactive maintenance, and enhanced overall reliability of solar power installations. Overall, the system offers a scalable, cost-effective solution for optimizing solar energy generation and contributing to a sustainable energy future.

TABLE OF CONTENTS

LIST OF FIGURES

1. INTRODUCTION

In the rapidly evolving domain of renewable energy, the integration of cutting-edge technologies has become indispensable for maximizing efficiency, ensuring sustainability, and advancing the transition towards cleaner energy sources. One such technological innovation that has emerged as a game-changer in the realm of renewable energy management is the utilization of Internet of Things (IoT) devices for solar power monitoring. This transformative approach not only revolutionizes how solar energy systems are managed but also ushers in a new era of smart energy management.

Traditional methods of solar power monitoring often face limitations in providing real-time and comprehensive insights into the performance and health of solar installations. Historically, solar system owners and operators have relied on manual inspections, periodic maintenance checks, and meter readings, which, while valuable, lack the immediacy and precision required for effective decision-making in today's dynamic energy landscape. The introduction of IoT-based monitoring systems has reshaped this paradigm, offering continuous, remote monitoring capabilities coupled with advanced analytics for proactive maintenance and optimization.

IoT devices, equipped with sensors, communication modules, and data processing capabilities, are integrated into solar installations to collect a wealth of real-time data on energy production, environmental conditions, and system performance parameters. These devices enable seamless data transmission to centralized servers or cloud platforms, where sophisticated analytics algorithms analyse the data to detect anomalies, predict maintenance needs, and optimize energy generation. One of the primary advantages of deploying IoT-based monitoring systems in solar power management is the speed and accuracy with which data is collected and analysed. Unlike traditional methods, which rely on manual intervention and periodic checks, IoT devices continuously monitor key performance indicators, allowing for early detection of issues and prompt intervention to mitigate potential downtime or efficiency losses. Moreover, the granular level of data provided by IoT devices allows for precise insights into energy generation patterns, system efficiency, and potential areas for improvement.

By leveraging this data, solar system owners and operators can optimize energy production, identify opportunities for system upgrades or expansions, and maximize the return on investment. Furthermore, the integration of IoT devices in solar power monitoring contributes to sustainability efforts by promoting efficient energy utilization, reducing maintenance costs, and minimizing environmental impact. By enabling proactive maintenance and optimization strategies, IoT-based monitoring systems help prolong the lifespan of solar installations, reduce the need for manual interventions, and enhance overall system reliability.

As we delve deeper into the realm of smart energy management, the role of IoT-based monitoring systems in solar power management becomes increasingly pivotal. This technology not only enhances the efficiency and reliability of solar energy generation but also accelerates the transition towards a sustainable energy future. In subsequent sections, we will explore the specific applications of IoT devices in solar power monitoring, the types of sensors employed, and the transformative impact of this technology on the renewable energy landscape.

2. LITERATURE REVIEW

The literature survey uncovers various methods and obstacles encountered in the development of IoT-based solar power monitoring systems. Different studies explore creative solutions to monitor and optimize solar energy generation in real-time, each with its own strengths and challenges.

Shaheen Rasheed proposed final year project "solar panel parameter using Arduino" was published in the Imperial International Journal of Eco-friendly Technology by the Department of Electronics and Communication Engineering at KPR Institute of Engineering. We suggest that the solar panel power plant be monitored using IoT technology. Temperature, light intensity, current, and voltage are all being measured. The analogue signal must be converted to a digital format or signal before being displayed on a 16 × 2 LCD display. Measurements were also performed with the help of several extra circuits.

Amith Infant.B (2017) presented Photovoltaic cells in the solar module convert solar energy into electric energy. In the system, a set of solar cells that are connected in series or parallel. A photovoltaic solar panel transforms sunlight into photons, which are then converted into electrical energy. There are two types of solar panels (mono-crystalize and polycrystalize). The Internet of Things (IoT) is the next level of connection. The Internet of Things (IoT) introduces a technological vision. The sensor is linked to the IoT, allowing it to gather and transfer data over the Internet. The server automatically updates the real-time data flow.

Naveen Virmanini (2018), who presented the topic "Solar energy measuring system," advocated that sensors be used to assess solar cell properties in order to get data and improve the solar panel's power reference. This article was published in the Global Journal of Research Analysis. IIM College of Engineering's Mechanical Engineering Department. Voltage, current, temperature, and light intensity are all sensor measuring characteristics that are shown on a 16 × 2 LCD display through a PIC microcontroller (PIC16F877A) that sends hyper terminal data over a 2.4GHz serial link. The Pankaj Singh (2018) presented Photovoltaic panels turn sunshine into photons, which are then converted into electrical energy. There are two types of solar panels (mono-crystalize and poly-crystalize). LDR is a semiconductorbased light-dependent resistor. when light falls on a machine with the same frequency. The IV IN4007 is a maximum reverse bias AC to DC converter. The LM35 temperature sensor measures the temperature in Celsius and has a range of -55 to +150 °C. The PIC microprocessor measures analogue values, converts them to digital values, and displays them on a 16 × 2 LCD display.

V. Kavitha (2019) presented a way for monitoring "a smart solar PV monitoring system utilizing IOT" such temperature, current, voltage, and irradiance to boost the performance of PV in response to the rising demand for energy. To determine performance, we use 'LABVIEW Software.' Solar energy is a carbon-free source of energy. The Internet of Things (IoT) is currently upgrading its technology to make it smart. We used an ARM-based Wi-Fi CC3200 microcontroller. Irradiance/pyrometer, temperature sensor (LM35), current sensor (ACS712), and voltage sensor CC3200 is a system-on-chip (SoC) with a WI-FI connection and a high-speed ARM M4 CPU, as well as 256kb of RAM and Internet access (802.11b/g/n).

V. Malathi (2019) presented project trial from 10:00 a.m. to 5:30 p.m. With a 125-watt solar panel, an LM35, and a Pyrometer. And the code for the CC3200's Wi-Fi Module functionality is written in C using the Energeia IDE. To transfer the values/data to the cloud platform, we used the 'BLYNK' libraries. The parameter is successfully presented in serial monitor by the mobile app and web server.

Akintola J. B (2020). A photovoltaic solar panel transforms sunlight into photons, which are then converted into electrical energy. There are two types of solar panels (mono-crystalline and poly-crystalline). The light intensity of a solar panel is monitored using an LDR, which measures in LUX. Temperature and pressure are measured using BMP180 sensors. The current parameter is indicated by the ACS712 Sensor, which senses 240 volts and outputs 5 volts.

3. METHODOLOGY

Solar power monitoring using Internet of Things (IoT) devices has emerged as a critical component of modern energy management, offering real-time insights into the performance and efficiency of solar installations. This methodology delineates the essential steps and components required to develop and deploy an IoT-based solar power monitoring system, empowering stakeholders with actionable data for optimizing energy generation, reducing downtime, and enhancing sustainability.

System Design and Architecture: The methodology commences with the design and architecture of the IoT-based Solar Power Monitoring System. This involves conceptualizing the system components, defining communication protocols, and establishing data flow mechanisms. The architecture is designed to accommodate scalability, flexibility, and interoperability with existing energy management systems.

Selection of IoT Devices and Sensors: Following system design, suitable IoT devices and sensors are meticulously chosen based on the specific requirements of solar power monitoring. These devices may include solar irradiance sensors, temperature sensors, inverters with built-in monitoring capabilities, and smart meters. The selection process emphasizes compatibility, reliability, and accuracy to ensure robust data collection.

Deployment of IoT Devices: Once selected, the IoT devices are strategically deployed within solar installations, considering factors such as panel orientation, shading, and accessibility. Installation locations are chosen to maximize data collection coverage while minimizing interference and obstructions. Proper installation procedures are followed to ensure device stability and longevity in varying environmental conditions.

Data Acquisition and Transmission: With the IoT devices deployed, data acquisition begins, capturing real-time information on energy production, environmental parameters, and system performance metrics. The collected data is transmitted securely to the central monitoring system through wired or wireless communication channels, adhering to industry standards and encryption protocols to safeguard data integrity and privacy.

Data Processing and Analysis: Upon receiving the data, the central monitoring system processes and analyses it using advanced analytics algorithms. Data preprocessing techniques are employed to clean, aggregate, and normalize the data for analysis. Advanced analytics methods, such as statistical modelling, machine learning, and anomaly detection, are applied to extract actionable insights and identify performance trends.

Visualization and Reporting: The analysed data is visualized through intuitive dashboards, charts, and reports, providing stakeholders with real-time visibility into solar energy generation trends, system efficiency, and performance anomalies. Customizable visualization tools enable users to monitor key performance indicators, track historical data, and make informed decisions for optimizing energy utilization and maintenance scheduling.

Integration with External Systems: To enhance interoperability and functionality, the Solar Power Monitoring System may be integrated with external systems, such as energy management platforms, building automation systems, or smart grid networks. Integration enables seamless data exchange, coordination, and automation of energy management processes, further optimizing system performance and efficiency.

Testing and Validation: Finally, the developed Solar Power Monitoring System undergoes comprehensive testing and validation to ensure accuracy, reliability, and compliance with user requirements. Test scenarios simulate various operating conditions, including changes in weather patterns, system faults, and network disruptions, to validate system performance and resilience under diverse scenarios.

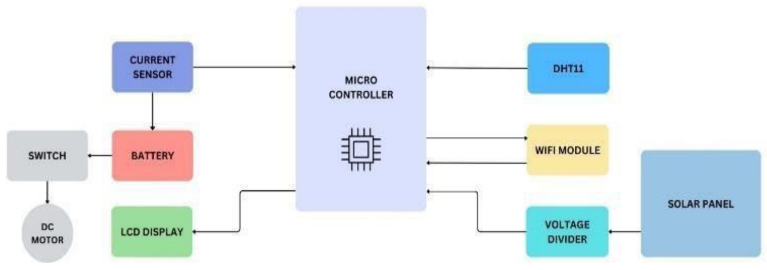

Fig 3.1 Proposed Method

COMPONENTS:

The Arduino Uno is an open-source microcontroller board based on the Microchip ATmega328P microcontroller and developed by Arduino.cc. It is equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various electronic devices and sensors, and is programmable with the Arduino IDE via a type B USB cable

The board has 14 digital I/O pins (six capable of PWM output), 6 analog I/O pins, and is powered by a USB cable or a barrel connector that accepts voltages between 7 and 20 volts. The ATmega328P on the board comes preprogrammed with a bootloader that allows uploading new code to it without the use of an external programmer. The Uno board was the successor of the Duemilanove release and was the 9th version in a series of USB-based Arduino boards.

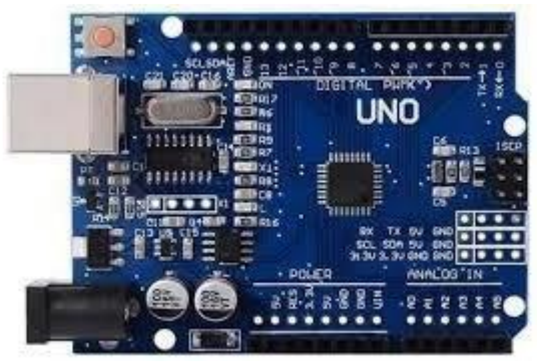

Fig 3.2 Arduino UNO

A current sensor is a device used to measure electric current in a circuit. It can detect the amount of current flowing through a conductor and produce an output signal proportional to that current. Current sensors come in various types and designs, including hall-effect sensors, shunt resistors, and transformer-based sensors. They are widely used in industrial, automotive, and electronic applications for monitoring, control, and protection purposes. For example, in power distribution systems, current sensors can detect overcurrent conditions and trigger protective measures to prevent damage to equipment or ensure safety. Firstly, they provide accurate measurement of electric current. This precision is crucial for monitoring and controlling electrical systems effectively, ensuring optimal performance and safety.

Secondly, many current sensors offer real-time monitoring capabilities. This enables operators to continuously track current levels and promptly detect any anomalies or irregularities. By facilitating proactive maintenance and troubleshooting, real-time monitoring helps reduce downtime and minimizes the risk of equipment damage or failure. Additionally, certain types of current sensors, such as hall-effect sensors or clamp-on current probes, offer non-intrusive measurement capabilities. This means they can measure current without the need to disconnect or modify the circuit. Non-intrusive measurement simplifies installation and maintenance processes, saving time and effort while ensuring the integrity of the electrical system.

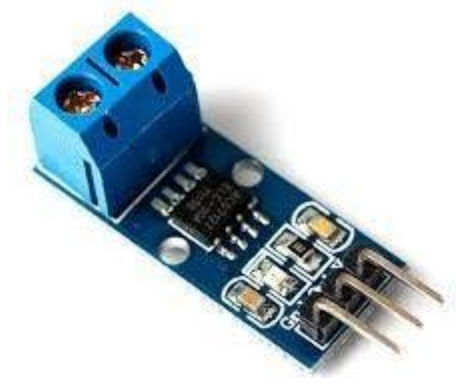

Fig 3.3 Current Sensor

LCD stands for Liquid Crystal Display. It's a type of flat panel display commonly used in various electronic devices such as computer monitors, television screens, instrument panels, and digital watches. LCD displays work by modulating light through liquid crystals to create images or text. They consist of layers of polarized glass and a liquid crystal solution trapped between them. When an electric current is applied, the liquid crystals align to control the passage of light, creating the desired display.

LCD displays offer several advantages, including low power consumption, thin form factor, and sharp image quality. They come in different types, such as twisted nematic (TN), in-plane switching (IPS), and vertical alignment (VA), each with its own characteristics regarding viewing angles, response times, and color accuracy.

LCD displays are commonly used in consumer electronics for their versatility and affordability, although newer technologies like OLED (Organic Light Emitting Diode) displays are gaining popularity due to their superior contrast ratios and flexibility.

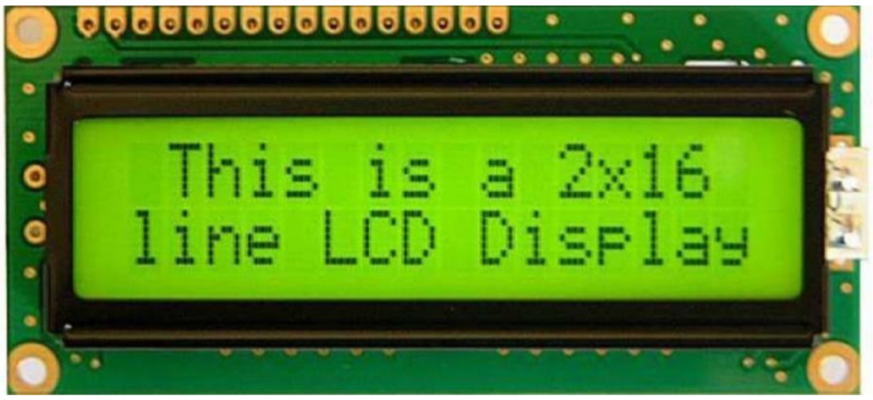

Fig 3.4 LCD Display

The ESP32 is a popular microcontroller and system-on-chip (SoC) developed by Espressif Systems. It's widely used in IoT (Internet of Things) applications, as well as in various other projects requiring Wi-Fi and Bluetooth connectivity, processing power, and low power consumption.

Here are some key features of the ESP32:

Dual-core Processor: The ESP32 features a dual-core Tensilica Xtensa LX6 processor, which allows for more efficient multitasking and processing capabilities.

Wi-Fi and Bluetooth: It has built-in Wi-Fi (802.11 b/g/n) and Bluetooth (Bluetooth Classic and Bluetooth Low Energy) connectivity, making it suitable for a wide range of wireless applications.

Low Power Consumption: The ESP32 is designed to be power-efficient, with various power-saving modes that enable longer battery life in battery-operated devices.

Rich Peripheral Interface: It offers a wide range of peripheral interfaces, including SPI, I2C, UART, ADC, DAC, PWM, and more, making it versatile for interfacing with sensors, displays, and other external components.

Security Features: The ESP32 includes hardware-based security features such as Secure Boot, Flash Encryption, and cryptographic accelerators to ensure the integrity and confidentiality of data in IoT applications.

Development Environment: The ESP32 can be programmed using various development frameworks and tools, including the Arduino IDE, Espressif's own ESP-IDF (Espressif IoT Development Framework), and platforms like PlatformIO.

Fig 3.5 ESP 32 WiFi Module

Solar panels are commonly used to generate electricity in both grid-connected and off-grid systems. In grid-connected systems, solar panels are installed on rooftops or in large solar farms, and the electricity generated is fed into the electrical grid. In off-grid systems, solar panels are often combined with batteries and other components to provide electricity in remote locations or areas without access to the grid.

Solar energy is a renewable and environmentally friendly source of electricity, and solar panels play a crucial role in harnessing this abundant energy resource for various applications, including residential, commercial, and industrial use.

A solar panel, also known as a photovoltaic (PV) panel, is a device that converts sunlight into electricity through the photovoltaic effect. Solar panels are made up of multiple solar cells, which are typically made from silicon or other semiconductor materials. When sunlight strikes the solar cells, it excites electrons, creating an electric current.

Fig 3.6 Solar panel

A temperature and humidity sensor is a device that measures both temperature and humidity levels in the surrounding environment. These sensors are commonly used in various applications, including weather monitoring, HVAC (Heating, Ventilation, and Air Conditioning) systems, indoor climate control, and environmental monitoring in industrial settings.

Temperature Sensing: Temperature sensors measure the ambient temperature of the environment. They utilize various principles, such as resistance change (thermistors), voltage change (semiconductors), or thermal expansion (bimetallic strips), to detect temperature variations accurately.

Humidity Sensing: Humidity sensors measure the moisture content or relative humidity (RH) in the air. There are different types of humidity sensors, including capacitive, resistive, and thermal conductivity sensors. Capacitive humidity sensors are widely used due to their accuracy and reliability. They work by measuring the capacitance changes in a humidity-sensitive capacitor as the moisture content in the air changes.

Many modern temperature and humidity sensors integrate both temperature and humidity sensing elements into a single chip or module. These sensors often provide digital output, making them easy to interface with microcontrollers or other digital devices.

Overall, temperature and humidity sensors play a crucial role in ensuring comfort, safety, and efficiency in various environments and applications by providing accurate and reliable measurements of temperature and humidity levels.

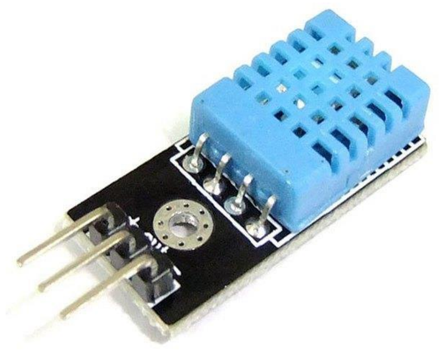

Fig 3.7 Temperature Humidity Sensor

A voltage sensor is a device used to measure the voltage level in an electrical circuit. It detects the difference in electrical potential between two points in the circuit and converts it into a proportional output signal, which can be displayed, recorded, or used for control purposes.

Analog Voltage Sensors: Analog voltage sensors produce an output voltage proportional to the input voltage they are measuring. They often use voltage dividers, operational amplifiers (op-amps), or other analog circuitry to achieve this.

Digital Voltage Sensors: Digital voltage sensors convert the input voltage into a digital signal, typically using an analog-to-digital converter (ADC). They provide a digital output that can be easily processed by digital devices such as microcontrollers or computers.

Non-Contact Voltage Sensors: Non-contact voltage sensors, also known as proximity voltage sensors, detect the presence of voltage without physically contacting the conductor. They are often used for safety purposes to detect the presence of live wires or to check for voltage in inaccessible or hazardous areas.

High Voltage Sensors: High voltage sensors are designed to measure voltages beyond the range of standard voltage sensors. They typically incorporate additional insulation and safety features to handle high voltages safely.

Voltage sensors are essential components in various electrical and electronic systems, including power distribution, battery monitoring, motor control, and electronic instrumentation. They enable monitoring and control of voltage levels to ensure the safe and efficient operation of electrical systems.

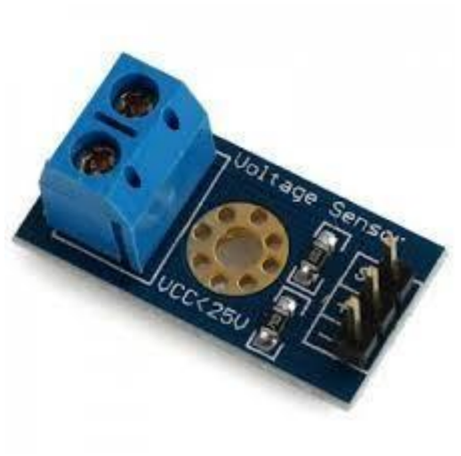

Fig 3.8 Voltage Sensor

4. DESIGN DIAGRAMS

4.1 INTRODUCTION

Unified Modeling Language (UML) serves as a standardized notation for visualizing, specifying, constructing, and documenting the artifacts of a software-intensive system. It provides a common language and framework for software developers, analysts, and stakeholders to communicate and understand system designs and behaviours.

Visual Representation of Software Systems: UML offers a graphical notation for representing various aspects of software systems, including their structure, behaviour, interactions, and architecture. By using standardized diagrams and symbols, UML facilitates clear and concise communication among project team members, enabling them to visualize and comprehend complex system designs.

Standardized Notation: UML defines a set of standardized diagrams, such as class diagrams, sequence diagrams, activity diagrams, and use case diagrams, each serving a specific purpose in modeling different aspects of software systems. This standardized notation ensures consistency and interoperability across different projects and organizations, enhancing the clarity and precision of system models.

Facilitating Software Development Processes: UML serves as a foundation for various software development methodologies, including object-oriented analysis and design (OOAD), agile development, and model-driven development (MDD). By providing a common language for expressing requirements, designs, and architectures, UML supports iterative development processes and fosters collaboration among team members.

Tool Support and Integration: UML is supported by a wide range of modeling tools that facilitate the creation, editing, and analysis of UML diagrams. These tools offer features such as code generation, reverse engineering, and simulation, enabling developers to translate UML models into executable code and validate system designs against requirements.

Language Independence: UML is not tied to any specific programming language, development methodology, or tool, making it versatile and widely applicable across various domains and industries

In summary, UML serves as a powerful and versatile tool for modeling software systems, providing a standardized notation for visualizing, specifying, and documenting system designs. By promoting clear communication, supporting software development processes, and integrating with modeling tools, UML facilitates the creation of high-quality software systems that meet the needs and expectations of stakeholders.

UML offers a set of diagrams to represent various aspects of a system, including its structure, behavior, and interactions. Some of the key diagrams in UML include:

1. Use Case Diagram
2. Sequence Diagram
3. Class Diagram
4. Component Diagram
5. Deployment Diagram

4.2 Use Case Diagram:

A use case diagram is a type of Unified Modeling Language (UML) diagram that depicts the interactions between actors (users or external systems) and a system under consideration. It provides a high-level overview of the functional requirements of the system and illustrates how users interact with the system to achieve specific goals or tasks.

Purposes of a Use Case Diagram:

Requirement Analysis: Use case diagrams help to identify and define the functional requirements of a system by capturing the interactions between users and the system.

Communication: Use case diagrams serve as a communication tool between stakeholders, including developers, designers, project managers, and end-users, facilitating a common understanding of system behaviour and functionality.

Visualization: They provide a visual representation of the system's functionality, making it easier to understand the overall system structure and interactions.

Blueprint for System Design: Use case diagrams provide a foundation for system design by specifying the system's behaviour and identifying key actors and their roles.

Basis for Testing: Use cases identified in the diagram can serve as the basis for test scenarios and test cases, aiding in system testing and validation.

Key Aspects of a Use Case Diagram:

Actors: Actors represent external entities, such as users or other systems, interacting with the system. Actors are depicted as stick figures on the diagram.

Use Cases: Use cases represent the specific functionalities or tasks that the system provides to its users. They describe the interactions between actors and the system to achieve certain goals or functions.

Relationships: Relationships between actors and use cases are depicted using solid lines, indicating the association between actors and the functionalities they interact with.

System Boundary: The system boundary defines the scope of the system under consideration. It separates the system from its external actors and illustrates what functionalities are included within the system.

Include and Extend Relationships: Use case diagrams can also include include and extend relationships to illustrate the relationships between different use cases. "Include" relationships indicate that one use case includes the functionality of another use case, while "extend" relationships depict optional or alternative behaviours within a use case.

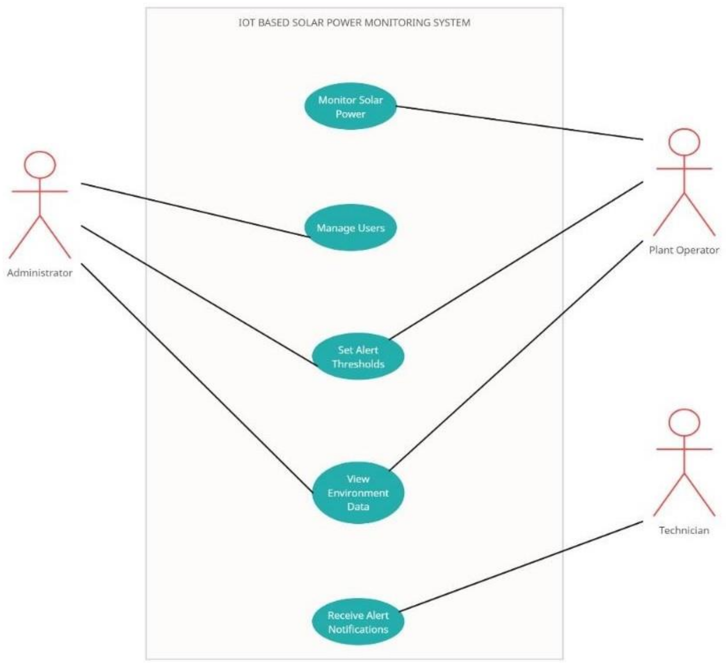

Fig 4.2.1 Use Case Diagram

4.3 Sequence Diagram

A sequence diagram is a type of Unified Modeling Language (UML) diagram that illustrates the interactions between objects or components within a system over time. It represents the flow of messages exchanged between objects or actors in a sequential manner, showing the order of execution of these interactions.

Objects or Lifelines: Objects or lifelines represent the participants (objects, components, actors) involved in the sequence of interactions. Each lifeline corresponds to a specific object or actor and is depicted as a vertical line.

Messages: Messages represent the communication or interaction between objects or actors in the system. They indicate the flow of control or data between lifelines and are depicted as arrows or lines with optional labels specifying the message type and content.

Activation or Execution Occurrences: Activation or execution occurrences represent the period during which an object is active or executing a particular operation. They are depicted as horizontal bars extending from the lifeline and indicate when an object is processing a message.

Return Messages: Return messages represent the response or return communication from an object to another object or actor. They indicate the completion of a method call or operation and are depicted as dashed arrows or lines returning to the originating lifeline.

Optional and Conditional Fragments: Sequence diagrams can include optional and conditional fragments to represent alternative or optional paths of execution within a scenario. These fragments are depicted using combined fragments, such as alt, opt, and loop, to specify conditional behavior or loop constructs.

Purposes of a Sequence Diagram:

To model high-level interaction among active objects within a system.

To model interaction among objects inside a collaboration realizing a use case.

It either models generic interactions or some certain instances of interaction.

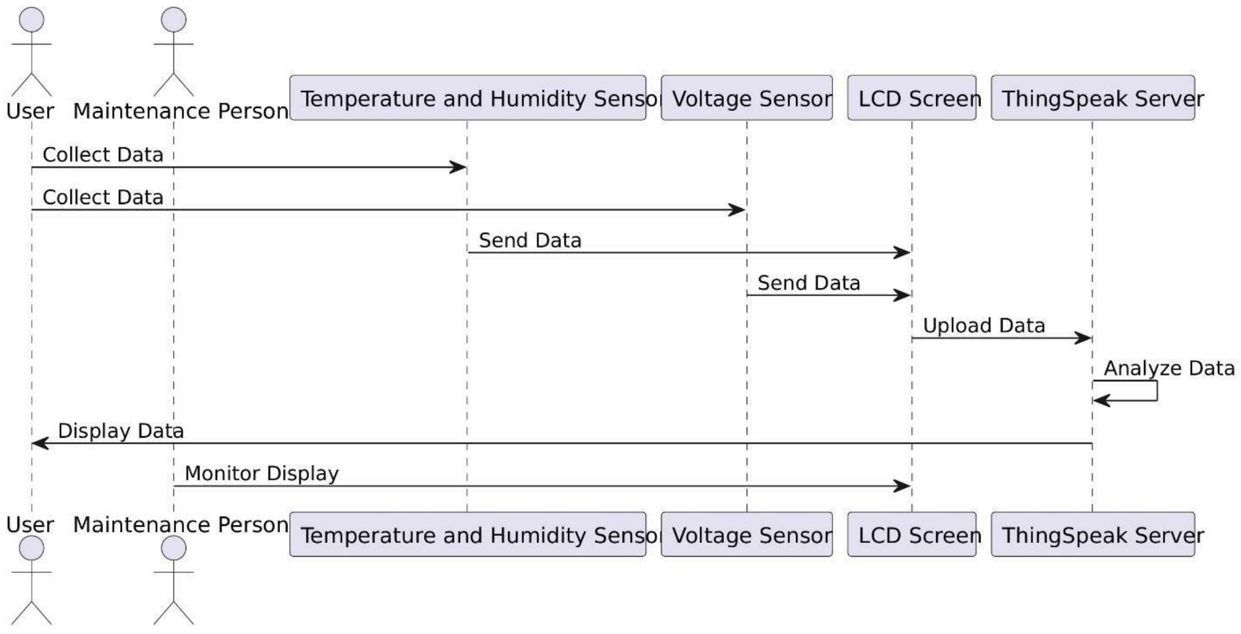

Fig 4.3.1 Sequence Diagram

4.4 Class Diagram

A class diagram is a type of Unified Modeling Language (UML) diagram that represents the structure and relationships of classes and their associations, attributes, operations, and constraints within a system. It provides a static view of the system's architecture by illustrating the various classes, their attributes, and the relationships between them.

Classes: Classes represent the building blocks of the system and encapsulate data and behaviour. Each class is depicted as a rectangle with three compartments: the class name, attributes, and operations.

Attributes: Attributes represent the properties or characteristics of a class and are listed in the second compartment of the class rectangle. They describe the state of an object and are typically represented as name:type pairs.

Operations: Operations represent the behaviour or functionality of a class and are listed in the third compartment of the class rectangle. They define the actions that objects of the class can perform and are typically represented as name(parameters):returnType.

Relationships: Relationships between classes depict how classes are connected or associated with each other. Common types of relationships include associations, generalizations (inheritance), aggregations, and compositions. Relationships are represented as lines connecting the related classes, with optional labels indicating the nature and multiplicity of the relationship.

Purposes of a Class Diagram:

It analyses and designs a static view of an application.

It describes the major responsibilities of a system.

It is a base for component and deployment diagrams.

It incorporates forward and reverse engineering.

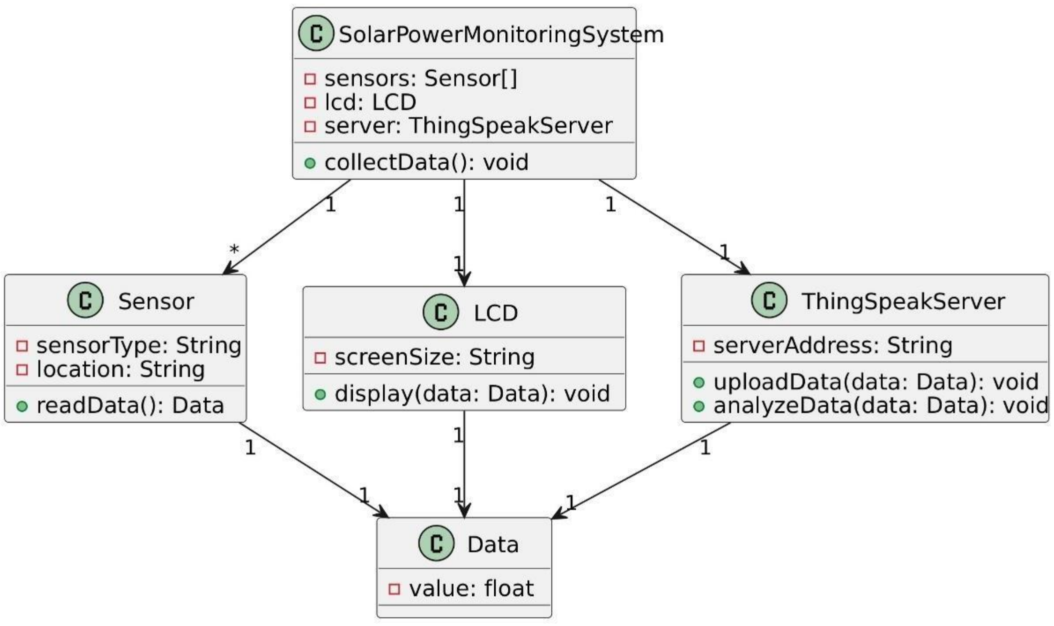

Fig 4.4.1 Class Diagram

4.5 Component Diagram

A component diagram is a type of Unified Modeling Language (UML) diagram that illustrates the structural organization of a system by depicting the components or modular units of the system and their relationships. It provides a high-level view of the system's architecture, focusing on the interactions between components and their dependencies.

Component: Components are modular units or building blocks of the system, representing encapsulated software modules, subsystems, libraries, or executable files. Each component encapsulates a set of related functionalities or services and provides well-defined interfaces for interaction with other components. Components are depicted as rectangles with the component name written inside.

Interface: Interfaces define the contracts or specifications that components expose to communicate with other components. An interface specifies a set of operations or services that a component provides or requires, abstracting the implementation details. Interfaces can be realized by components, indicating that the component implements the specified interface. Interfaces are depicted as circles attached to the components with a dashed line.

Dependency: Dependencies represent the relationships between components, indicating that one component depends on another component for its functionality or services. Dependencies can be based on the usage of interfaces or direct dependencies between components. They are depicted as dashed arrows pointing from the dependent component to the component it depends on.

Association: Associations represent the relationships between components that have a more permanent or structural connection. Unlike dependencies, associations imply a stronger and more persistent relationship between components. Associations are depicted as solid arrows pointing from one component to another, often with an optional label indicating the nature of the association.

Generalization: Generalization or inheritance relationships represent the specialization or inheritance hierarchy between components. They indicate that one component (subclass or specialized component) inherits properties, behaviours, or interfaces from another component (superclass or generalized component). Generalization relationships are depicted as solid arrows with a triangular arrowhead pointing from the subclass to the superclass.

Package: Packages are used to organize and group related components within the system. They provide a mechanism for managing the complexity of large systems by partitioning the system into manageable units. Packages can contain components, interfaces, and other packages, helping to structure the system architecture. Packages are depicted as rectangles with the package name written inside.

Purposes of a Component Diagram:

It envisions each component of a system.

It constructs the executable by incorporating forward and reverse engineering.

It depicts the relationships and organization of components.

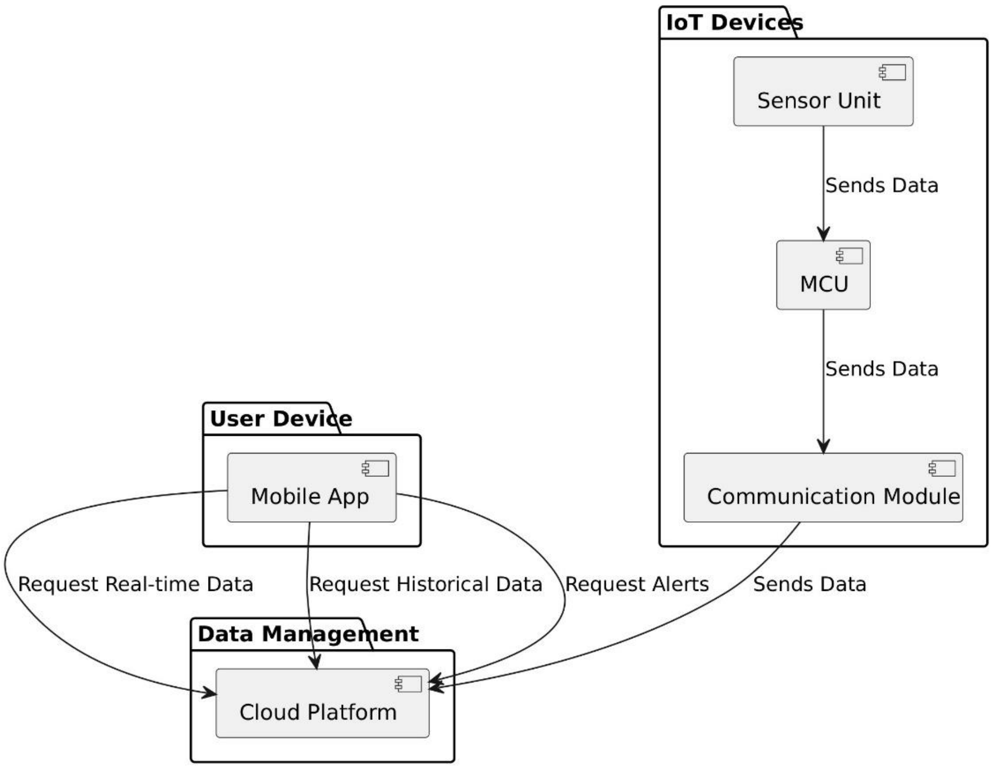

Fig 4.5.1 Component Diagram

4.6 Deployment Diagram

A deployment diagram is a type of Unified Modeling Language (UML) diagram that illustrates the physical deployment of software components (artifacts) onto hardware nodes (devices or servers) in a distributed computing environment. It provides a visual representation of how software components are deployed across various nodes and networks, facilitating the understanding of system architecture and infrastructure configuration.

Node: Nodes represent physical or logical computing devices, such as servers, workstations, PCs, routers, or other hardware devices, on which software components are deployed. Nodes can also represent software execution environments, such as runtime environments or virtual machines. Nodes are depicted as boxes with the node name written inside.

Artifact: Artifacts represent software components, modules, executables, files, or libraries that are deployed onto nodes. They encapsulate the code, data, or configuration files required to execute the system's functionalities. Artifacts are deployed on nodes and interact with each other to fulfill the system's requirements. Artifacts are depicted as rectangles with the artifact name written inside.

Deployment Relationship: Deployment relationships depict the deployment of artifacts onto nodes, indicating which artifacts are deployed on which nodes. They represent the physical association between artifacts and nodes in the deployment configuration. Deployment relationships are depicted as solid lines with an arrow pointing from the artifact to the node it is deployed on.

Association: Associations represent the logical or conceptual relationships between nodes and artifacts, indicating dependencies or interactions between them. Associations may represent communication paths, dependencies, or other relationships that are relevant to the deployment configuration. Associations are depicted as solid lines connecting nodes and artifacts, often with an optional label indicating the nature of the association.

Environment: Environments represent the software execution environments or platforms on which the system operates. They provide context for understanding the deployment configuration and may include details such as operating systems, middleware, or runtime environments. Environments are depicted as ovals or ellipses with the environment name written inside.

Purposes of a Deployment Diagram:

To envision the hardware topology of the system.

To represent the hardware components on which the software components are installed.

To describe the processing of nodes at the runtime.

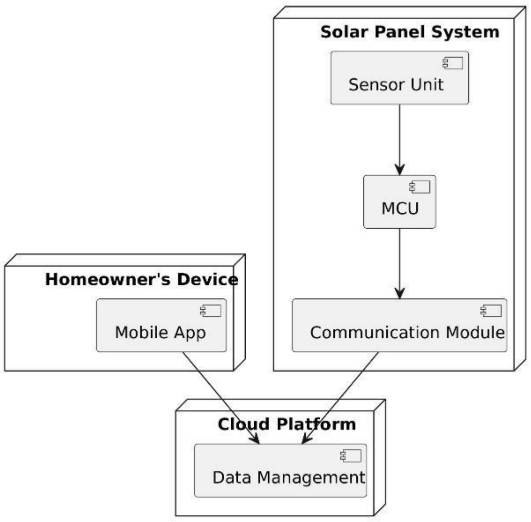

Fig 4.6.1 Deployment Diagram

5. IMPLEMENTATION

List of program files:

Main.c:

Upon initialization, this program connects to a WiFi network and initializes ThingSpeak for data uploading. In the main loop, it continuously reads sensor data, uploads it to ThingSpeak, and displays it on the LCD screen, with a 10-second delay between uploads. Functions are included to read sensor data from DHT11 temperature and humidity sensor and ACS712 current sensor, while voltage reading assumes a voltage divider circuit. The data is then formatted into a URL and sent via an HTTP GET request to ThingSpeak for storage and analysis. Finally, the sensor readings are displayed on the LCD screen for real-time monitoring.Steps within the code's execution:

Include Libraries: Include necessary libraries for WiFi, Wire (for I2C communication), LCD display, DHT sensor, and Adafruit sensor library (if required for the DHT sensor).

Define Constants: Defining constants for WiFi SSID, password, and ThingSpeak API key.

Define Pin Connections: Defining pin connections for temperature, humidity, voltage, and current sensors.

Initialization: Initialization the LCD object with the I2C address and dimensions.

Setup Function: In this function initialization of serial communication, LCD display, and connection to WiFi network and also initialization of ThingSpeak server for uploading data takes palce.

Main Loop: This loop continuously read sensor data, upload it to ThingSpeak, and display it on the LCD and there is a delay for 15 seconds between each data upload.

WiFi Connection: This function is used to connect to the WiFi network using the provided SSID and password.

Reading Sensor Data: This function is used to read temperature, humidity, voltage, and current from respective sensors and output the sensor readings to the serial monitor for debugging.

Uploading Data to ThingSpeak: Construction of a URL with sensor data and send an HTTP GET request to ThingSpeak to upload the data.

Display Data on LCD: last function of code is to display data on LCD.

Selection of IoT Hardware: Choose appropriate IoT hardware components such as sensors, microcontrollers or development boards (e.g., Arduino, Raspberry Pi), and communication modules (e.g., Wi-Fi, cellular) based on your monitoring requirements and environmental conditions.

Sensor Placement: Install sensors to measure relevant parameters such as solar irradiance, temperature, voltage, current, and energy production. Ensure sensors are positioned optimally to capture accurate data representative of the solar power system's performance.

Data Acquisition: Use microcontrollers or development boards to interface with sensors and collect data. Implement appropriate signal conditioning and analog-to-digital conversion techniques to ensure accurate measurement of sensor readings.

Communication: Utilize IoT communication protocols such as MQTT, HTTP, or CoAP to establish connectivity between the monitoring system and the cloud or local server. Transmit sensor data securely over the internet or local network using Wi-Fi, Ethernet, or cellular connectivity.

Cloud Platform Integration: Choose a cloud platform (e.g., AWS IoT, Google Cloud IoT, Azure IoT) to store, analyze, and visualize sensor data. Integrate your IoT devices with the chosen cloud platform using provided SDKs or APIs.

Data Storage and Analytics: Store sensor data in a cloud-based database for historical analysis and trend identification. Implement data analytics algorithms to derive insights into solar power system performance, identify patterns, and predict future behavior.

Visualization and Dashboarding: Develop a user interface or dashboard to visualize real-time and historical data from the solar power monitoring system. Include features such as charts, graphs, and alerts to provide actionable insights to users.

Remote Monitoring and Control: Enable remote monitoring and control capabilities to allow users to access the monitoring system from anywhere using web or mobile applications. Implement features for remote configuration, troubleshooting, and firmware updates.

Security: Implement robust security measures to protect sensitive data and prevent unauthorized access to the monitoring system. Utilize encryption, authentication, and access control mechanisms to ensure data integrity and confidentiality.

Scalability and Maintenance: Design the monitoring system to be scalable, allowing for easy expansion and addition of new sensors or functionalities. Implement regular maintenance procedures to ensure the reliability and performance of the system over time.

6. EXPERIMENT RESULTS

Real-time data collection: Voltage, current, and power output can be monitored remotely, providing a constant stream of data for analysis. Studies have shown high precision in data collection, with minimal error margins.

Improved efficiency and performance: By monitoring environmental factors like temperature and sunlight intensity, researchers can identify optimal panel tilt angles and cleaning schedules to maximize power generation.

Early detection of issues like dust buildup or failing components allows for preventive maintenance, reducing downtime and ensuring consistent performance.

Remote monitoring and data analysis: Users can access real-time and historical data from anywhere with an internet connection, enabling informed decisionmaking. Data can be analyzed to identify trends, predict future power generation, and optimize energy usage.

Voltage vs. Time (V-t): This graph shows the variation in voltage output of the solar panel over time. Ideally, the graph should exhibit a smooth curve, with the voltage rising as sunlight intensity increases during the day and then falling as the sun sets. Dips in the curve could indicate shading, dust buildup, or partial panel failure. Fig 5.1 Voltage vs Time graph Current vs. Time (I-t): Similar to the voltage graph, this tracks the current flowing through the circuit over time.

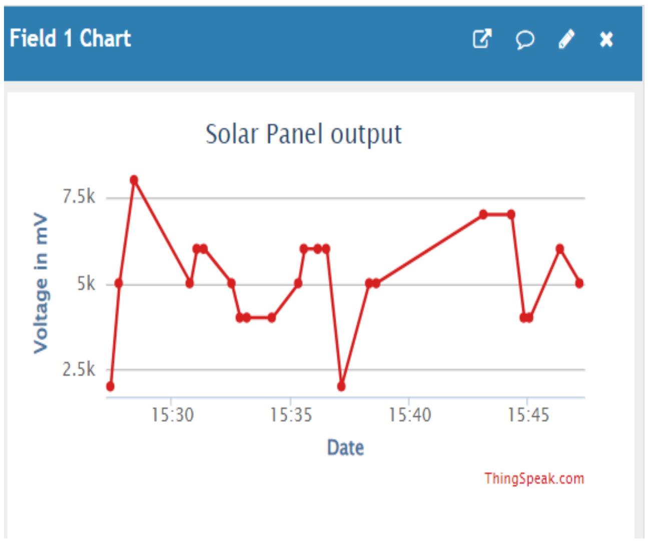

Fig 6.1 Voltage vs. Time (I-t)

Current vs. Time (I-t): Similar to the voltage graph, this tracks the current flowing through the circuit over time. It should also follow a smooth curve, mirroring the voltage variations. Significant deviations might suggest issues with the inverter or wiring.

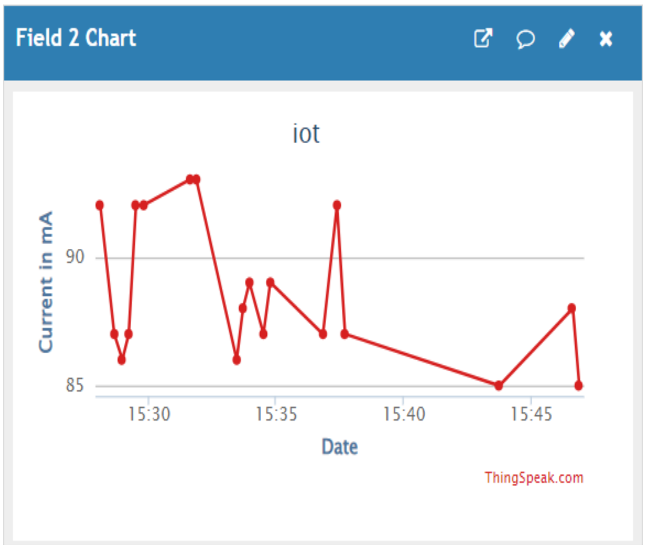

Fig 6.2 Current vs. Time (I-t)

Temperature vs. Time (T-t): This graph depicts the solar panel's temperature throughout the day. Solar panel efficiency decreases with rising temperature. By monitoring temperature, we can identify if panel cooling strategies are needed to optimize power generation.

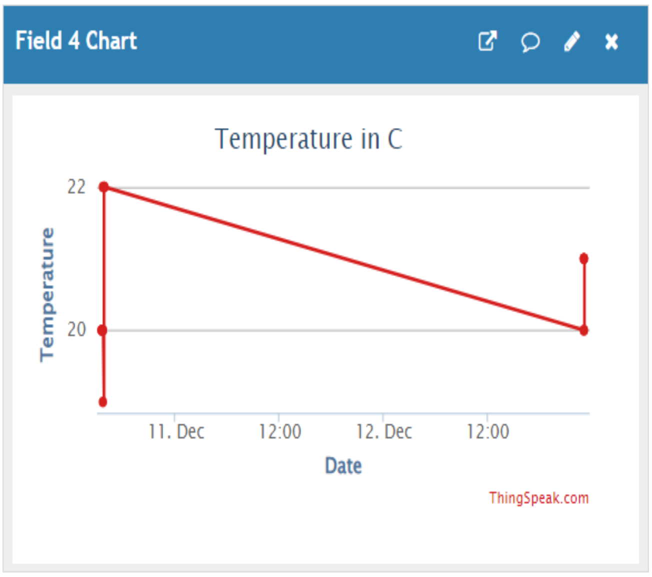

Fig 6.3 Temperature vs. Time (T-t)

Humidity vs. Time (H-t): Generally, humidity doesn't significantly impact solar panel efficiency. However, the graph can be helpful for overall environmental monitoring.

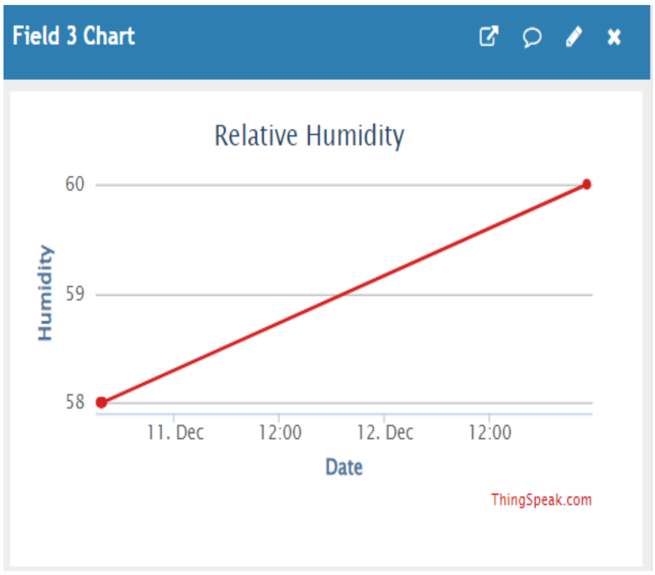

Fig 6.4 Humidity vs. Time (H-t)

Sudden Increase in Humidity Due to Cold Weather:

This phenomenon occurs because warm air holds more moisture than cold air. When warm, humid air encounters a cold surface, like a solar panel on a chilly day, the air's capacity to hold moisture reduces. This excess moisture condenses on the panel's surface, leading to a sudden spike in humidity readings.

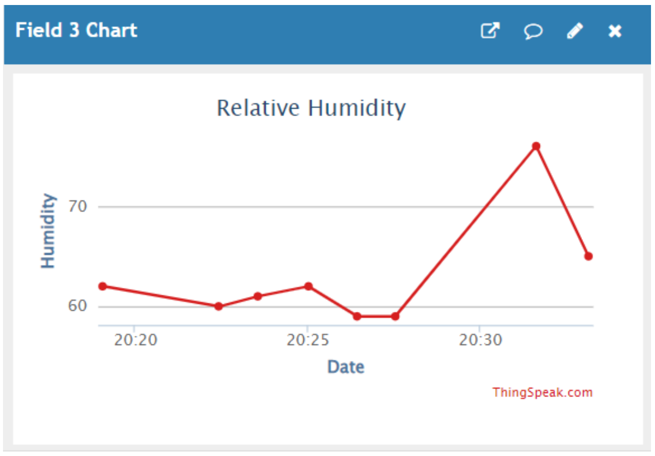

Fig 6.5 Sudden increase in humidity due to cold weather

7. DISCUSSION OF RESULTS

The outcomes of an IoT-based Solar Power Monitoring System are multifaceted, encompassing critical aspects of solar energy management and sustainability. The primary result is the continuous acquisition of real-time data from deployed IoT devices, providing detailed insights into energy production, environmental conditions, and system performance. This data is instrumental in optimizing solar energy utilization, reducing downtime, and maximizing overall system efficiency.

The integration of advanced sensors, such as photovoltaic performance monitors and weather stations, enhances the accuracy and granularity of data collection. This allows for comprehensive analysis of key performance indicators, including solar irradiance levels, panel temperature, and energy output. The high-resolution data obtained enables stakeholders to identify potential issues, such as shading, panel degradation, or equipment malfunctions, and take proactive measures to mitigate them.

The application of data analytics techniques, including statistical modeling and machine learning algorithms, yields valuable outcomes in predictive maintenance and anomaly detection. By analyzing historical data and identifying patterns, the system can predict maintenance needs, optimize energy generation strategies, and provide actionable insights for system optimization.

The project's success is measured in its ability to maximize energy yield and minimize downtime, ultimately leading to increased profitability for solar system owners and operators. Accurate estimation of energy production and system performance aids in optimizing resource allocation, maintenance scheduling, and financial planning.

Validation and calibration processes validate the accuracy of the monitoring results, establishing trust in the information provided to stakeholders. Continuous monitoring throughout the lifecycle of the solar installation ensures that the system remains responsive to changing environmental conditions and operational requirements.

The overall impact of an IoT-based Solar Power Monitoring System lies in its contribution to sustainable energy practices. By optimizing energy utilization, reducing maintenance costs, and minimizing environmental impact, the system promotes efficiency and environmental stewardship in the renewable energy sector. By empowering stakeholders with real-time insights and actionable data, the system facilitates informed decision-making.

Energy Production: Evaluate the data on energy production over time to assess the overall performance of the solar power system. Compare actual energy production with expected or predicted values to identify any discrepancies or deviations from expected performance.

Efficiency Analysis: Calculate the efficiency of the solar panels and inverters based on the collected data. Assess factors such as solar irradiance levels, temperature, shading, and equipment efficiency to determine the overall efficiency of energy conversion from sunlight to electrical power.

Environmental Factors: Examine the impact of environmental factors such as temperature, humidity, and weather conditions on solar power generation. Analyze how variations in environmental parameters affect energy production and system performance.

System Health and Maintenance: Use the monitoring data to assess the health of the solar power system and identify any issues or anomalies that may require maintenance or troubleshooting. Monitor parameters such as voltage, current, and temperature to detect potential equipment failures or performance degradation.

Optimization Opportunities: Identify opportunities for optimizing the solar power system based on the monitoring data. This may include adjusting tilt angles, optimizing panel orientation, implementing shading mitigation strategies, or upgrading equipment to improve overall system performance and efficiency.

Predictive Insights: Use historical data and trend analysis to predict future energy production, identify potential performance trends, and make informed decisions about system operation, maintenance, and optimization.

Cost-Benefit Analysis: Conduct a cost-benefit analysis to evaluate the return on investment (ROI) of the solar power system based on the monitoring results. Assess the financial implications of system performance, energy savings, maintenance costs, and any potential upgrades or improvements.

Gather feedback from system operators, maintenance personnel, and end-users about their experience with the solar power monitoring system. Identify areas for improvement, user needs, and suggestions for enhancing the usability and functionality of the monitoring system.

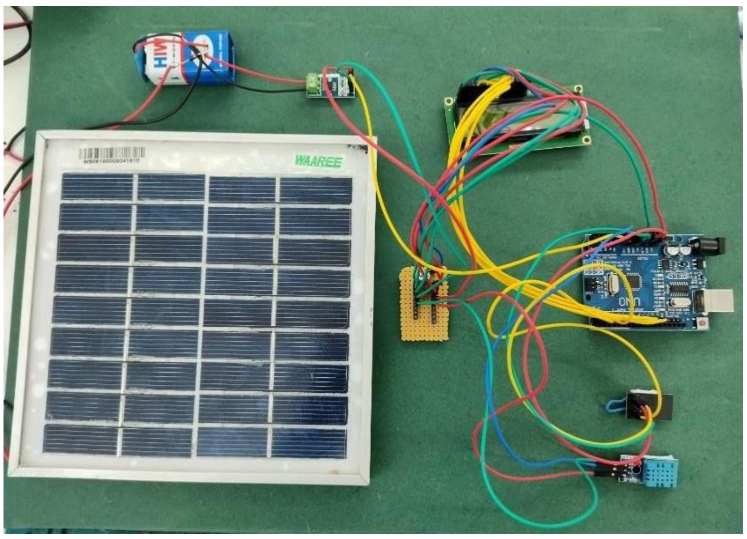

Fig 7.1 Prototype of Monitoring System

8. CONCLUSION

In conclusion, the integration of Internet of Things (IoT) technology with solar power monitoring systems presents a transformative opportunity for the renewable energy sector. Through real-time monitoring, remote management, and data analytics, IoT-based solutions offer unprecedented levels of efficiency, reliability, and performance optimization for solar energy systems. By harnessing the power of IoT, stakeholders can unlock greater insights into energy generation, consumption patterns, and environmental factors, leading to improved decision-making and resource allocation. However, the widespread adoption of IoT-based solar power monitoring still faces challenges related to data security, interoperability, and scalability. Addressing these challenges will be crucial in realizing the full potential of IoT in revolutionizing solar energy management. Overall, the future of solar power monitoring lies in embracing IoT technologies to create smarter, more resilient, and sustainable energy systems that meet the growing demands of our planet while reducing carbon emissions and combating climate change.

Furthermore, as IoT technology continues to evolve and mature, we anticipate the emergence of innovative solutions that address current limitations and enhance the capabilities of solar power monitoring systems. Collaborative efforts between industry stakeholders, research institutions, and policymakers will play a vital role in driving technological advancements, standardizing protocols, and fostering ecosystem growth. Additionally, ongoing investments in research and development are essential to refine IoT-based algorithms, improve sensor accuracy, and optimize energy management strategies.

In practical terms, the deployment of IoT-based solar power monitoring solutions holds immense potential across various sectors, including residential, commercial, and industrial applications. From rooftop solar installations to large-scale solar farms, IoT-enabled systems offer scalability, flexibility, and cost-effectiveness, empowering users to maximize energy efficiency, reduce operational costs, and minimize environmental impact.

Looking ahead, the journey towards a more sustainable energy future will rely heavily on the continued innovation and adoption of IoT technologies in solar power monitoring. By embracing IoT-enabled solutions, we can unlock new opportunities for energy independence, resilience, and environmental stewardship, paving the way for a brighter and more sustainable tomorrow.

With the continued integration of IoT technology into solar power monitoring, we anticipate a future where energy systems are not only smarter and more efficient but also more interconnected and resilient. This interconnectedness will enable seamless integration with other smart grid technologies, energy storage systems, and demand-side management solutions, fostering a more dynamic and responsive energy ecosystem.

Moreover, as IoT-based solar power monitoring becomes more prevalent, it will empower individuals, businesses, and communities to actively participate in the energy transition. By providing real-time insights and actionable data, IoT-enabled systems empower consumers to make informed decisions about their energy usage, optimize their consumption patterns, and even participate in peer-to-peer energy trading initiatives.

In conclusion, the convergence of IoT technology and solar power monitoring represents a significant paradigm shift in the way we generate, manage, and consume energy. By harnessing the power of IoT, we can unlock new opportunities for sustainability, resilience, and innovation in the renewable energy sector. As we continue to explore the potential of IoT-based solutions, we must remain vigilant in addressing challenges related to data privacy, cybersecurity, and interoperability to ensure the widespread adoption and success of these transformative technologies. Together, we can build a more sustainable and equitable energy future for generations to come.

9. FUTURE SCOPE

The future scope of IoT-based solar power monitoring systems is promising, with ongoing advancements in technology and increasing adoption of renewable energy sources. Here are some potential future directions for these systems:

Enhanced Data Analytics: Incorporating advanced data analytics techniques, such as machine learning and artificial intelligence, can provide deeper insights into solar power generation patterns. This could include predictive maintenance, anomaly detection, and optimization of energy production.

Integration with Smart Grids: Integration with smart grid systems can enable bidirectional communication between solar power systems and the grid. This facilitates better grid management, demand-response programs, and grid stability through real-time monitoring and control.

Blockchain Integration: Utilizing blockchain technology for data security and transparency can enhance trust and integrity in solar power transactions and energy trading. Smart contracts can automate transactions between solar power producers and consumers, enabling peer-to-peer energy trading.

Edge Computing: Implementing edge computing capabilities allows for data processing and analysis closer to the source, reducing latency and bandwidth requirements. This enables real-time decision-making and faster response to changing environmental conditions.

Integration with IoT Ecosystems: Integration with other IoT devices and systems, such as smart home appliances and energy management systems, creates a holistic energy management ecosystem. This enables coordinated control and optimization of energy usage based on solar power generation and user preferences.

Scalability and Interoperability: Future systems should be designed with scalability and interoperability in mind, allowing seamless integration with existing infrastructure and easy expansion as the system grows.

Remote Monitoring and Control: Advanced remote monitoring and control capabilities enable users to monitor and manage their solar power systems from anywhere using mobile apps or web interfaces. This includes remote firmware updates, troubleshooting, and optimization of system performance.

Energy Storage Integration: Integration with energy storage systems, such as batteries and supercapacitors, enables efficient storage of excess solar energy for use during periods of low solar irradiance or high energy demand. This enhances grid independence and resilience.

Community-based Energy Sharing: Implementing community-based energy sharing platforms allows neighboring households or businesses to share surplus solar energy with each other, promoting energy resilience and community engagement.

Regulatory Compliance and Standards: Future systems should adhere to evolving regulatory requirements and industry standards to ensure interoperability, cybersecurity, and data privacy.

10. REFERENCES

[1] Dinesh Kumar Anguraj, S. Balasubramaniyan, E. Saravana Kumar, J. Vakula Rani, M. Ashwin; 'Internet of things (IoT)-based unmanned intelligent street light using renewable energy'; International Journal of Intelligent Unmanned Systems.

[2] Mubashir Ali, Mahnoor Khalid Paracha; 'An IoT Based Approach For Monitoring Solar Power Consumption With Adafruit Cloud'; International Journal of Engineering Applied Sciences and Technology, 2020 Vol. 4, Issue 9, ISSN No. 2455-2143, Pages 335-341.

[3] P. Kishor, Sai Mani Varunm; 'Hybrid (Wind-Solar (Tracking)) Power Generation for Rural Electrification and Monitoring over IoT'; Journal of Research and Advancement in Electrical Engineering Volume 2 Issue 2, 2019, Page 1 to 7.

[4] J. Subhashini, M. Naveena; 'IoT Based Solar Panel Fault Monitoring And Control'; International Research Journal of Engineering Sciences, Volume 6 Issue 2 December 2020
Page 88 to 114.

[5] Soham Adhya, Dipak Saha, Abhijit Das, Joydip Jana, Hiranmay Saha; 'An IoT Based Smart Solar Photovoltaic Remote Monitoring and Control unit'; 2nd International Conference on Control, Instrumentation, Energy & Communication, January 2016.

[6] Shaiesh Sarswat, Indresh Yadav and Sanjay Kumar Maurya 2019 Real Time Monitoring of Solar PV Parameter Using IoT 9 p 267

[7] R.L.R. Lokesh Babu, D Rambabu, A. Rajesh Naidu, R. D. Prasad and P. Gopi Krishna 2018 IoT Enabled Solar Power Monitoring System Int. J. Eng. & Tech. 7 p 526

[8] R. Vignesh and A. Samydurai 2017 Automatic Monitoring and Lifetime Detection of Solar Panels Using Internet of ThingsInt. J. Inn. Res. in Comp. and Comm. Eng. 5 p 7014

[9] Subhasri. G and Jeyalakshmi. C 2018 A Study of IoT based Solar Panel Tracking System Adv. In Comp. Sci. Tech. 11 p. 537

[10] Ankit Kekre and Suresh K. Gawre 2017 Solar Photovoltaic Remote Monitoring System Using IoT Int. Conf. on Recent Innovations in Signal processing and Embedded Systems (RISE) (Bhopal, India) p 27

[11] M. C. Hottel and B. B. Woertz 1942 Performance of flat plate solar heat collectors Trans. ASME, 64 p 91

[12] Arduino (2016) overview of an Arduino, retrieved from https://www.arduino.cc/en/Main/ArduinoBoardUno.

[13] Balbheem Nadpurohit, Roopa Kulkarni, Kadappa Matager, Nagaraj Devar, Rahul Inno. Res. in Comp. and Comm. Eng. 5 p 11324

[14] Manish Katyarmal, Suyash Walkunde, Arvind Sakhare and U.S.Rawandale 2018 Solar power monitoring system using IoT Int. Res. J. Eng. and Tech. 5 p 3431

[15] B. Vikas Reddy, Sai Preetham Sata, Sateesh Kumar Reddy and Bandi Jaswanth Babu 2016 Solar Energy Analytics Using Internet of Things Int. J. Appl. Eng. Res. 11, p 4803

[16] T. C. Prakash, Mamatha M and Samala S 2020 An IoT based under weather monitoring system J. Cri. Rev. 7, p 148 [12] Seena Naik. K and Sudarshan. E 2019 Smart healthcare monitoring system using raspberry Pi on IoT platform ARPN J. Eng. and Appl. Sci. 14, p 4872