REAL-TIME DATA-DRIVEN INTELLIGENT IOT-BASED WIND ENERGY MANAGEMENT SYSTEM

A PROJECT REPORT

Submitted by

JANARTHANAN S (922521104063)

JAYARAJ V (922521104064)

PRANAV G (922521104116)

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

V.S.B ENGINEERING COLLEGE, KARUR - 639 111

(An Autonomous Institution, Affiliated to Anna University)

MAY-2025

BONAFIDE CERTIFICATE

Certified that this project report titled **"REAL-TIME DATA-DRIVEN INTELLIGENT IOT-BASED WIND ENERGY MANAGEMENT SYSTEM"** is the bonafide work of **JANARTHANAN S (922521104063), JAYARAJ V (922521104064) and PRANAV G (922521104116)** who carried out the project under my supervision.

SIGNATURE	**SIGNATURE**
Dr. P. ANBUMANI M.E., Ph.D.,	**Ms.G.YASIKA M.E.,**
HEAD OF THE DEPARTMENT	**SUPERVISOR**
Assistant Professor	Assistant Professor
Computer Science and Engineering	Computer Science and Engineering
V.S.B. Engineering College	V.S.B. Engineering College
Karur-639111.	Karur-639111.

Submitted for Project Viva-Voce held on ____________________________

INTERNAL EXAMINER **EXTERNAL EXAMINER**

DECLARATION

We jointly declare that the project report on **"REAL-TIME DATA-DRIVEN INTELLIGENT IOT-BASED WIND ENERGY MANAGEMENT SYSTEM"** is the result of original work done by us and best of our knowledge, similar work has not been submitted to **"ANNA UNIVERSITY CHENNAI"** for the requirement of Degree, Bachelor of engineering. This project report is submitted on the partial fulfilment of the requirement of the award of Degree of Bachelor of engineering.

Signature

JANARTHANAN S

JAYARAJ V

PRANAV G

Place : Karur

Date :

ACKNOWLEDGEMENT

First and foremost, we express our thanks to our parents for providing us with a very nice environment for doing this project. We wish to express our sincere thanks to our founder and Chairman **Shri. V. S. BALSAMY B.Sc., L.L.B.,** for his endeavor in educating us in this premier institution.

We extend our gratitude to **Dr. C. VENNILA M.E., Ph.D.,** Principal and **Dr. T. S. KIRUBHASHANKAR M.E., Ph.D.,** Vice Principal, **V.S.B ENGINEERING COLLEGE, KARUR** for their high degree of encouragement and moral support during this project.

We are grateful to Head of the Department, **Dr. P. ANBUMANI M.E., Ph.D.,** Assistant Professor, Department of Computer Science and Engineering and our project coordinator **Dr.S.KARTHI M.E., Ph.D.,** Associate Professor, Department of Computer Science and Engineering for their guidance throughout the course of our project.

We are grateful to our Project Supervisor **Ms. G. YASIKA M.E.,** Assistant Professor, Department of Computer Science and Engineering for her valuable support.

Our sincere thanks to all the faculty members of V.S.B. Engineering College and our friends for their help in the successful completion of this project work.

Finally, we bow before God, almighty who always had a better plan for us. We give our praise and glory to almighty god for successful completion of our project

V.S.B. Engineering College, Karur

(An Autonomous Institution)

Department of Computer Science and Engineering

VISION, MISSION, POs, PEOs and PSOs

Vision of the Institution:

We end eavour to impart futuristic technical education of the highest quality to the student community and to inculcate discipline in them to face the world with self-confidence and thus we prepare them for life as responsible citizens to uphold human values and to be of service at large. We strive to bring of the Institution as an Institution of academic excellence of International standard.

Mission of the Institution:

We transform persons into personalities by the state-of the art infrastructure, time consciousness, quick response and the best academic practices through assessment and advice.

Vision of the Department:

To produce competent Computer Science Engineers through Quality Education and Research.

Mission of the Department:

The Department strives to contribute to the expansion of knowledge in the discipline of Computer Science and Engineering by,

1. Adopting an efficient teaching learning process in correlation with industrial needs.
2. Ensuring technical proficiency, facilitating to pursue higher studies and carry out R&D activities.

3. Developing problem solving and analytical skills with sound knowledge in basic sciences and Computer Science Engineering.

4. Inculcating managerial skills to become socially responsible, ethical and competitive professionals.

Program Educational Objectives (PEOs)

The Graduates of Computer Science and Engineering are able to

PEO 1: Work in Multinational companies and become successful IT professionals.

PEO 2: Pursue higher studies and have their career in educational institutions research organizations, or be entrepreneurs.

PEO 3: Possess social responsibility, team work skills, leadership capabilities and urge for learning in their professional fields.

Program Specific Outcome (PSO)

PSO 1: Addressing societal problems through design and development of software and firmware solutions using latest Computer Science tools and technologies.

PSO 2: Involving enthusiastically in software development, software testing, storage, computing and business intelligence sectors.

PSO 3: Use their technical expertise in latest technologies and update their knowledge continuously in Computer Science and Engineering to excel in their career.

Program Outcomes (PO)

PO 1: Engineering knowledge: Apply the Mathematical knowledge and the basics of Science and Engineering to solve the problems pertaining to Computer Science and Engineering.

PO2: Problem analysis: Identify and formulate Computer Science and Engineering problems from research literature and be able to analyze the problem using first principles of Mathematics and Engineering Sciences

PO3: Design/development of solutions: Come out with solutions for the complex problems and to design system components or process that fulfil the particular needs taking into account public health and safety and the social, cultural and environmental issues.

PO4: Conduct investigations of complex problems: Draw well-founded conclusions applying the knowledge acquired from research and research methods including design of experiments, analysis and interpretation of data and synthesis of information and to arrive at significant conclusion.

PO5: Modern tool usage: Form, select and apply relevant techniques, resources and Engineering and IT tools for Engineering activities like electronic prototyping, modelling and control of systems and also being conscious of the limitations.

PO6: The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional Computer Science and Engineering practice.

PO7: Environment and sustainability: Be aware of the impact of professional Engineering solutions in societal and environmental contexts and exhibit the knowledge and the need for sustainable Development.

PO8: Ethics: Apply the principles of Professional Ethics to adhere to the norms of the engineering practice and to discharge ethical responsibilities.

PO9: Individual and team work: Function actively and efficiently as an individual or a member/leader of different teams and multidisciplinary projects.

PO10: Communication: Communicate efficiently the engineering facts with a wide range of engineering community and others, to understand and prepare reports and design documents; to make effective presentations and to frame and follow instructions

PO11: Project management and finance: Demonstrate the acquisition of the body of engineering knowledge and insight and Management Principles and to apply them as member / leader in teams and multidisciplinary environments.

PO 12: Life-long learning: Recognize the need for self and life-long learning, keeping pace with technological challenges in the broadest sense.

ABSTRACT

This project proposes the design and implementation of an intelligent, IoT-enabled wind energy system optimized for renewable power generation and efficient energy management. The system harnesses wind energy through a turbine and stores the generated power in a battery, with an emphasis on optimizing performance under varying environmental conditions. Voltage sensors monitor the output of both the wind generator and the battery, preventing overcharging or undercharging and enhancing battery lifespan. A temperature sensor further assists in determining optimal wind turbine operation, adapting dynamically to changes in environmental conditions for consistent energy production. An inverter is used to convert the generated DC power into AC, making the system suitable for domestic and industrial applications. An Arduino Uno microcontroller serves as the core of the system, processing data from all sensors and managing system responses. IoT functionality is integrated using a Wi-Fi module (ESP8266), allowing real-time data transmission to cloud platforms such as Blynk or Thing Speak. This enables users to monitor battery levels, wind turbine performance, and temperature remotely via a smartphone interface. The IoT dashboard provides live system status, alerts, and historical data for trend analysis and predictive maintenance. With smart automation, remote monitoring, and efficient energy management, this project presents a sustainable, user-friendly solution ideally suited for remote, off-grid, or smart grid applications.

Keywords : IoT-enabled, Renewable power generation, Energy management, Wind turbine, Arduino Uno, ESP8266, Real-time monitoring, Cloud platforms.

Note : Through this project work, it is able to map PO1 to PO12 and also to map PSO1,PSO2 and PSO3.

OBJECTIVE

The primary objective of this project is to design and implement an intelligent renewable energy system that efficiently harnesses wind energy and utilizes a battery for optimal energy storage and distribution. This system aims to promote sustainable power generation by continuously monitoring and controlling key environmental and electrical parameters such as wind speed, battery voltage, and temperature. By incorporating the Arduino Uno microcontroller along with a network of sensors, the system ensures precise data acquisition and real-time processing. One of the major goals is to enhance the reliability and efficiency of the energy system through automation and smart decision-making, using relays to regulate the operation of the wind turbine and connected loads based on sensor feedback. The integration of Internet of Things (IoT) technology enables remote monitoring and control, allowing users to access real-time data and system status through a cloud-based mobile application. This ensures that the wind turbine operates under optimal conditions while preventing overcharging or deep discharging of the battery. Another key objective is to provide a user-friendly interface that offers live updates on power generation, battery health, and energy consumption, thereby empowering users to manage their energy usage more efficiently. The project also focuses on making the system cost-effective and scalable for use in remote or off-grid areas, where access to conventional power sources is limited. By combining renewable energy generation, real-time monitoring, and smart automation, the objective is to develop a reliable, eco-friendly, and intelligent energy solution that contributes to environmental sustainability while providing practical energy management benefits.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATION

DC	**D**irect **C**urrent
AC	**A**lternating **C**urrent
IOT	**I**nternet **O**f **T**hings
BMS	**B**attery **M**anagement **S**ystem
LED	**L**ight **E**mitting **D**iode
LCD	**L**iquid **C**rystal **D**isplay
HV	**H**igh **V**oltage
LV	**L**ow **V**oltage
DC-DC	**D**irect **C**urrent **T**o **D**irect **C**urrent
ESP	**E**xternal **S**erial **P**eripheral
WI-FI	**W**ireless **F**idelity
GPS	**G**lobal **P**ositioning **S**ystem
FV	**F**orward **V**oltage
VRLA	**V**alve **R**egulated **L**ead **A**cid
PWM	**P**ulse **W**idth **M**odulation
PID	**P**roportional **I**ntegral **D**erivative
I/O	**I**nput/**O**utput

CHAPTER 1

INTRODUCTION

Renewable energy has emerged as a vital solution to the increasing global demand for electricity and the urgent need to reduce dependence on fossil fuels. Among various renewable sources, wind energy stands out due to its abundance, sustainability, and minimal environmental impact. The rapid depletion of non-renewable energy resources and growing environmental concerns have driven researchers and engineers to develop efficient systems that harness natural energy sources. Wind, as a clean and inexhaustible source, offers immense potential for both small-scale and large-scale energy applications. However, traditional wind energy systems often lack intelligent control and real-time monitoring capabilities, which limits their performance and adaptability in variable environmental conditions. Hence, there is a pressing need for integrating smart technologies into renewable energy systems to optimize their efficiency and usability.

This project introduces an innovative approach by combining wind energy generation with intelligent monitoring and control using IoT and microcontroller-based technology. The core of the system is built around an Arduino Uno microcontroller, which collects data from various sensors such as voltage, temperature, and wind condition sensors. The integration of IoT technology plays a crucial role in enhancing the functionality of the system by enabling remote access, real-time updates, and centralized control through a cloud-connected application. This allows users to monitor key parameters such as battery voltage, wind energy production, and system performance through a user-friendly mobile or web interface. The real-time data acquisition ensures that the wind turbine operates efficiently, and the energy storage system functions within safe and optimal thresholds. The ability to remotely control the load and turbine operations further ensures that the system remains reliable and reduces energy wastage, especially in off-grid and remote areas.

One of the most significant advantages of the proposed system is its ability to ensure energy management through intelligent decision-making and automation. For instance, the system uses relays to turn the wind turbine on or off based on weather conditions, or to disconnect the load when battery levels are too low, thus protecting the system and conserving energy. Additionally, the inverter included in the system converts the generated DC power to AC, making it suitable for regular household or industrial use. The cloud-based IoT platform not only displays real-time values but also stores historical data that can be used for trend analysis and future improvements. This makes the project highly scalable and adaptable for different energy needs and geographical conditions. The system's modular design, cost-effectiveness, and use of open-source technologies make it a promising solution for clean energy generation and intelligent energy management, especially in developing regions where access to power is inconsistent.

As the global push toward sustainable living intensifies, the adoption of smart renewable energy systems has become more than just an option—it's a necessity. Traditional energy systems are limited by manual operation, poor adaptability to changing environmental conditions, and an inability to offer real-time insights into performance metrics. The integration of Internet of Things (IoT) technology with renewable sources like wind energy marks a transformative step in the energy sector. IoT enables the automation of energy systems, providing constant data feedback that helps optimize operation, maintenance, and energy output. This combination empowers even remote users to manage their energy production and consumption with high precision, thereby contributing to smarter energy use and a lower carbon footprint.

Wind energy, while promising, is inherently variable. Wind speed and direction can change unexpectedly, which may lead to underperformance or even equipment damage if not handled appropriately. To mitigate such risks, this project employs a system where wind speed and environmental conditions are continuously monitored using sensors. These sensors relay data to the Arduino Uno microcontroller, which acts as the brain of the system. Based on this data, the system decides whether to store the energy, supply it to the load, or halt operations for safety. Moreover, temperature variations that could affect turbine performance are also tracked to enhance efficiency and reliability. The real-time feedback loop created by these components ensures that the wind turbine only operates when beneficial, saving power and reducing wear and tear.

Fig 1.1 wind mill

In modern energy systems, user accessibility and data transparency are key. This project incorporates a user-friendly dashboard through Arduino IoT Cloud, allowing users to access live data such as battery charge level, wind turbine output, and inverter status. This not only helps with monitoring but also offers control options—users can remotely start or stop the wind turbine and load system using their mobile devices or computers. Such features are crucial for real-world applications where the physical presence of a user at the site is impractical. Additionally, the cloud platform maintains historical records of energy generation and consumption, which can be used for performance analysis, predictive maintenance, and further system upgrades. Overall, this project aims to provide a reliable, intelligent, and self-sustaining renewable energy solution that is especially beneficial for rural, off-grid, or disaster-prone areas where consistent power supply is a challenge.

1.1 INTERNET OF THINGS

The Internet of Things (IoT) is the network of devices such as vehicles, and home appliances that contain electronics, software, actuators, and connectivity which allows these things to connect, interact and exchange data. The IoT involves extending Internet connectivity beyond standard devices, such as desktops, laptops, smartphones and tablets, to any range of traditionally dumb or non-internet-enabled physical devices and everyday objects. Embedded with technology, these devices can communicate and interact over the Internet, and they can be remotely monitored and controlled.

Fig:1.2 Representation of IoT

The definition of the Internet of things has evolved due to convergence of multiple technologies, real-time analytics, machine learning, commodity sensors, and embedded systems. Traditional fields of embedded systems, wireless sensor networks, control systems, automation (including home and building automation), and others all contribute to enabling the Internet of things.

1.2 HOW IOT WORKS

An IoT ecosystem consists of web-enabled smart devices that use embedded processors, sensors and communication hardware to collect, send and act on data they acquire from their environments. IoT devices share the sensor data they collect by connecting to an IoT gateway or other edge device where data is either sent to the cloud to be analyzed or analyzed locally. Sometimes, these devices communicate with other related devices and act on the information they get from one another. The devices do most of the work without human intervention, although people can interact with the devices -- for instance, to set them up, give them instructions or access the data. The connectivity, networking and communication protocols used with these web-enabled devices largely depend on the specific IoT applications deployed.

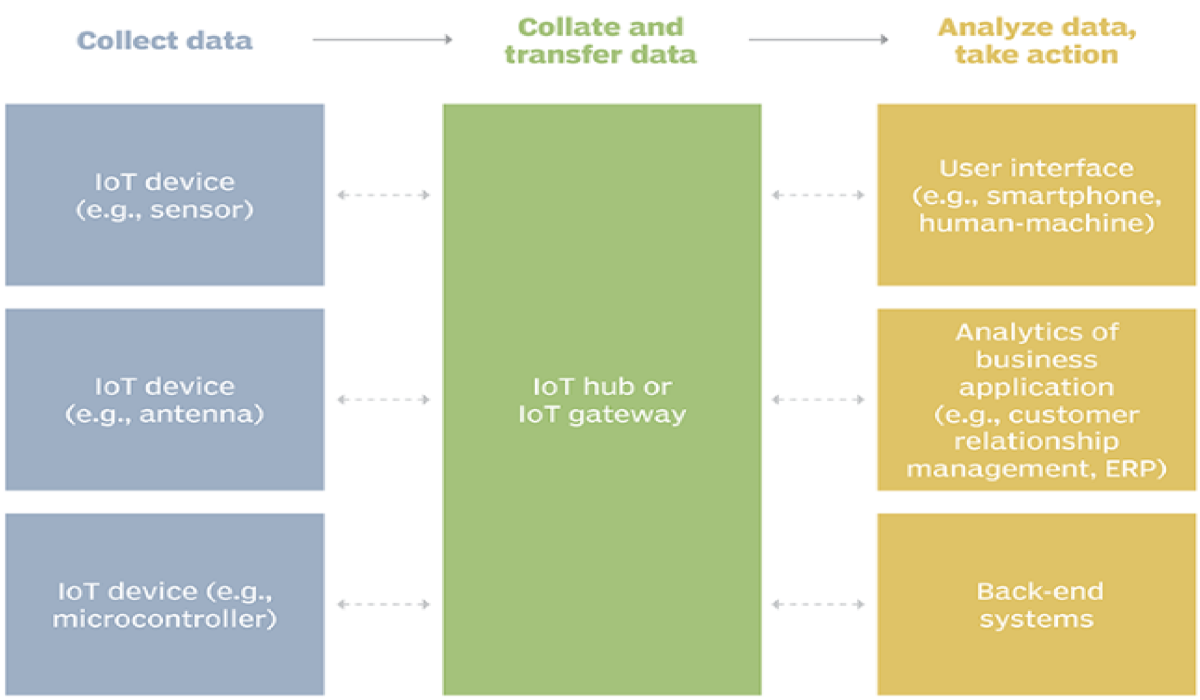

Fig1.3 Example of an IOT system

1.3 BENEFITS OF IOT

The internet of things offers a number of benefits to organizations, enabling them to:

- Monitor their overall business processes;
- Improve the customer experience;
- Save time and money;
- Enhance employee productivity;
- Integrate and adapt business models;
- Make better business decisions; and
- Generate more revenue.

IoT encourages companies to rethink the ways they approach their businesses, industries and markets and gives them the tools to improve their business strategies.

1.4 IOT SECURITY AND PRIVACY ISSUES

The internet of things connects billions of devices to the internet and involves the use of billions of data points, all of which need to be secured. Due to its expanded attack surface, IoT security and IoT privacy are cited as major concerns.

One of the most notorious recent IoT attacks was Mirai, a botnet that infiltrated domain name server provider Dyn and took down many websites for an extended period of time in one of the biggest distributed denial-of-service

(DDoS) attacks ever seen. Attackers gained access to the network by exploiting poorly secured IoT devices.

Because IoT devices are closely connected, all a hacker has to do is exploit one vulnerability to manipulate all the data, rendering it unusable. And manufacturers that don't update their devices regularly -- or at all -- leave them vulnerable to cybercriminals.

Additionally, connected devices often ask users to input their personal information, including names, ages, addresses, phone numbers and even social media accounts -- information that's invaluable to hackers.

However, hackers aren't the only threat to the internet of things; privacy is another major concern for IoT users. For instance, companies that make and distribute consumer IoT devices could use those devices to obtain and sell users' personal data.

Beyond leaking personal data, IoT poses a risk to critical infrastructure, including electricity, transportation and financial services.

1.5 IOT APPLICATION AREAS

Near Field Communication (NFC), Radio frequency Identification (RFID), Machine-to-Machine Communication (M2M) & Vehicle-to-Vehicle Communication (V2V) are the technologies by which IoT is being implemented exponentially. It is assumed that more than 50 billion IoT devices will be connected through internet. It is going to change human life, working style, entertaining ways and many more. IoT have many Applications Areas and domain of these application are increasing day by day.

There are ample of applications of IoT as follow:

- Smart Cities

- Building & Home automation
- Environmental Monitoring
- Automotive Industry
- Smart Retail
- Smart Agriculture
- Smart Industry
- Energy Management
- Healthcare Monitoring

1.6 EMBEDDED SYSTEM

An embedded system is one kind of a computer system mainly designed to perform several tasks like to access, process, store and also control the data in various electronics-based systems. Embedded systems are a combination of hardware and software where software is usually known as firmware that is embedded into the hardware. One of its most important characteristics of these systems is, it gives the o/p within the time limits. Embedded systems support to make the work more perfect and convenient. So, we frequently use embedded systems in simple and complex devices too. The applications of embedded systems mainly involve in our real life for several devices like microwave, calculators, TV remote control, home security and neighborhood traffic control systems, etc.

An embedded system is integration of hardware and software, the software used in the embedded system is set of instructions which are termed as a program. The microprocessors or microcontrollers used in the hardware circuits of embedded systems are programmed to perform specific tasks by following the set of instructions. These programs are primarily written using any programming software like Proteus or Lab-view using any programming languages such as C or C++ or embedded C. Then, the program is dumped into the microprocessors or microcontrollers that are used in the embedded system circuits.

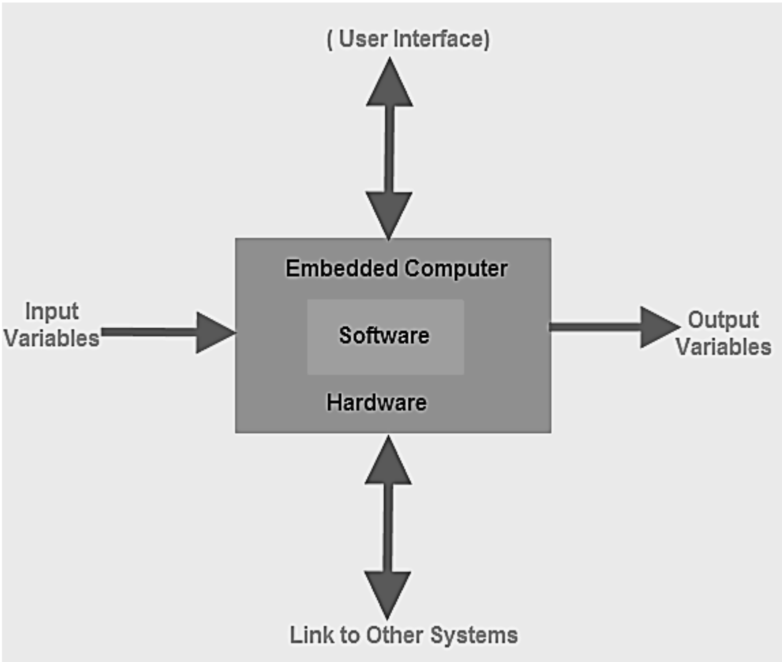

Fig 1.4 Embedded Syestem Diagram

EMBEDDED SYSTEM CLASSIFICATION

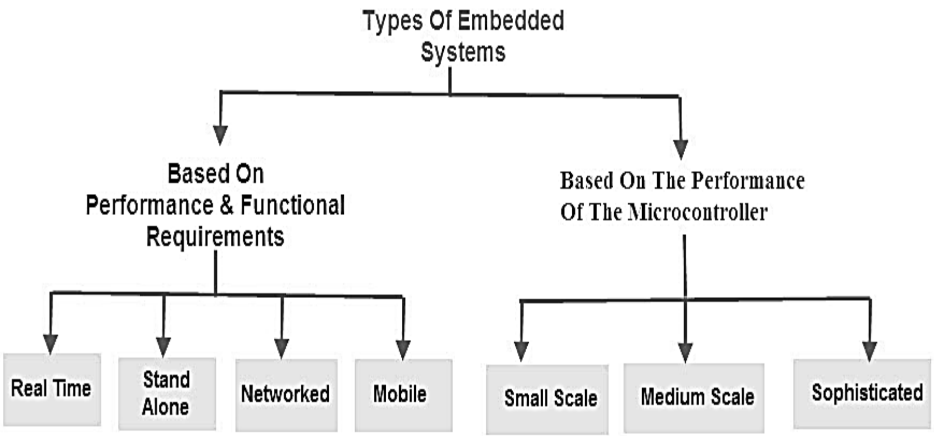

Fig 1.5 Embedded Syestem Classification

Embedded systems are primarily classified into different types based on complexity of hardware & software and microcontroller (8 or 16 or 32-bit). Thus, based on the performance of the microcontroller, embedded systems are classified into three types such as:

- Small scale embedded systems
- Medium scale embedded systems
- Sophisticated embedded systems

Further, based on performance and functional requirements of the system embedded system classified into four types such as Real time embedded systems

- Standalone embedded systems
- Networked embedded systems
- Mobile embedded systems

EMBEDDED SYSTEM HARDWARE

An embedded system uses a hardware platform to perform the operation. Hardware of the embedded system is assembled with a microprocessor/microcontroller. It has the elements such as input/output interfaces, memory, user interface and the display unit. Generally, an embedded system comprises of the following

- Power Supply
- Memory
- Processor
- Timers
- Output/Output circuits
- Serial communication ports
- SASC (System application specific circuits)

EMBEDDED SYSTEM SOFTWARE

The software of an embedded system is written to execute a particular function. It is normally written in a high-level setup and then compiled down to offer code that can be stuck within a non-volatile memory in the hardware. Embedded system software is intended to keep in view of the following three limits

- Convenience of system memory
- Convenience of processor's speed
- When the embedded system runs constantly, there is a necessity to limit power dissipation for actions like run, stop and wake up.

RTOS (Real Time Operating System)

A system which is essential to finish its task and send its service on time, then only it said to be a real time operating system. RTOS controls the application software and affords a device to allow the processor run. It is responsible for managing the different hardware resources of a personal computer and also host applications which run on the PC.

This operating system is specially designed to run various applications with an exact timing and a huge amount of consistency. Particularly, this can be significant in measurement & industrial automation systems where a delay of a program could cause a safety hazard.

MEMORY AND PROCESSORS

The different kinds of processors used in an embedded system include Digital Signal Processor (DSP), microprocessor, RISC processor, microcontroller, ASSP processor, ASIP processor, and ARM processor. The different types of memories of an embedded system are given in the below chart.

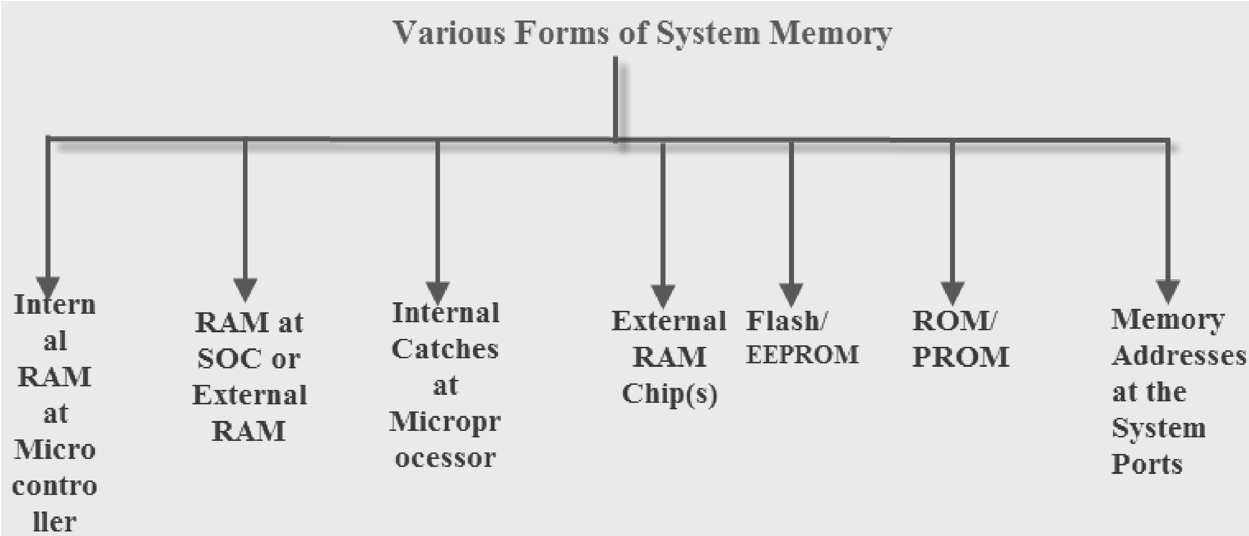

Fig 1.6 Types Of Memories Of System

EMBEDDED SYSTEM CHARACTERISTICS

- Generally, an embedded system executes a particular operation and does the similar continually. For instance: A pager is constantly functioning as a pager.
- All the computing systems have limitations on design metrics, but those can be especially tight. Design metric is a measure of an execution features like size, power, cost and also performance.
- It must perform fast enough and consume less power to increase battery life.
- Several embedded systems should constantly react to changes in the system and also calculate particular results in real time without any delay. For instance, a car cruise controller; it continuously displays and responds to speed & brake sensors. It must calculate acceleration/de-accelerations frequently in a limited time; a delayed computation can consequence in letdown to control the car.
- It must be based on a microcontroller or microprocessor based.
- It must require a memory, as its software generally inserts in ROM. It does not require any secondary memories in the PC.

- It must need connected peripherals to attach input & output devices.
- An Embedded system is inbuilt with hardware and software where the hardware is used for security and performance and Software is used for more flexibility and features.

EMBEDDED SYSTEM APPLICATIONS

The applications of an embedded system basics include smart cards, computer networking, satellites, telecommunications, digital consumer electronics, missiles, etc.

Fig 1.7 Applications Of Embedded System

- Embedded systems in automobiles include motor control, cruise control, body safety, engine safety, robotics in an assembly line, car multimedia, car entertainment, E-com access, mobiles etc.

CHAPTER 2

LITERATURE SURVEY

2.1 TITLE: Efficient Monitoring and Control of Wind Energy Conversion Systems Using Internet of Things (IoT): A Comprehensive Review

AUTHOR: S. Karad and R. Thakur

YEAR: 2022

DESCRIPTION: The review by S. Karad and R. Thakur provides an extensive overview of utilizing Internet of Things (IoT) technologies for efficient monitoring and control of Wind Energy Conversion Systems (WECS). As global energy demands shift towards renewable sources, ensuring the optimal performance and reliability of wind power systems has become a critical concern. This study explores how IoT-based frameworks can enhance real-time data acquisition, condition monitoring, and predictive maintenance, ultimately improving the efficiency and lifespan of WECS. The paper highlights that conventional monitoring approaches often rely on manual inspections or periodic assessments, which can lead to delayed fault detection, increased maintenance costs, and potential power losses. By integrating IoT-enabled smart sensors and data communication technologies, real-time operational data such as wind speed, turbine vibration, rotor temperature, and power output can be continuously collected and transmitted to cloud platforms for analysis. The authors emphasize that IoT platforms, coupled with machine learning algorithms, can facilitate predictive maintenance by detecting anomalies and predicting component failures before they occur. A key aspect discussed in the review is the use of wireless sensor networks (WSNs) for remote monitoring of wind turbines. These WSNs ensure continuous tracking of critical parameters and generate alerts when abnormal conditions are detected. Additionally, IoT-based Supervisory Control and Data Acquisition (SCADA) systems enable centralized control and automation of multiple wind turbines, allowing real-time adjustments to optimize energy output. Through intelligent analytics, these systems can dynamically modify the turbine's blade angles, rotational speeds, and yaw positions to enhance power generation efficiency. The authors also explore the role of big data analytics in wind energy management. IoT devices generate vast amounts of operational data, which, when processed using cloud-based platforms, provide actionable insights to operators.

2.2 TITLE: Model Based Predictive Control Strategy for Grid-Connected Wind Energy System

AUTHOR: M. Paul and D.V. Kumar

YEAR: 2022

DESCRIPTION: The study by M. Paul and D.V. Kumar introduces a Model-Based Predictive Control (MPC) strategy designed to optimize the performance of grid-connected wind energy systems. With the increasing integration of renewable energy sources into the power grid, ensuring system stability, reliability, and power quality has become a major challenge. The authors propose an MPC framework that dynamically adjusts wind turbine operations and grid-connected inverters to mitigate fluctuations in power output and enhance energy efficiency. MPC is a control strategy that predicts future system behavior based on a mathematical model and optimizes control actions over a defined prediction horizon. In the proposed system, real-time data from wind turbines, including wind speed, rotor speed, generator voltage, and grid conditions, are processed to predict power output and grid interactions. The predictive model estimates possible disturbances caused by variations in wind speed and grid load changes. Using this prediction, the controller optimizes turbine and inverter settings to maintain stability and maximize power transfer to the grid. A key strength of the proposed system is its ability to address grid voltage fluctuations and ensure smooth grid synchronization. The authors highlight that traditional Proportional-Integral-Derivative (PID) control methods, while effective in steady-state conditions, struggle to adapt to rapid variations in wind speed and load demand. In contrast, MPC continuously monitors dynamic changes and adjusts system parameters proactively, ensuring seamless energy delivery while maintaining grid frequency and voltage within permissible limits. The research emphasizes that the predictive model incorporates constraints such as generator power limits, inverter capacity, and grid voltage stability, ensuring that the control actions remain within operational boundaries. The system minimizes error margins by dynamically adjusting the turbine’s blade pitch angle and controlling reactive power injection to maintain optimal grid synchronization. Furthermore, the MPC approach reduces the likelihood of system faults by preventing overloading and ensuring smooth transitions between operating states. The authors conducted extensive simulations and performance analyses under varying wind and grid conditions to validate the effectiveness of the proposed MPC strategy. Results demonstrated a significant improvement in power quality, grid stability, and overall system efficiency compared to traditional control methods.

2.3 TITLE: A Review of Conventional and Advanced MPPT Algorithms for Wind Energy Systems

AUTHOR: D. Kumar and K. Chatterjee

YEAR: 2021

DESCRIPTION: D. Kumar and K. Chatterjee provide a comprehensive review of conventional and advanced Maximum Power Point Tracking (MPPT) algorithms for wind energy systems, highlighting their importance in maximizing energy extraction from wind turbines. As wind energy systems operate under variable wind conditions, the efficiency of power conversion largely depends on the system's ability to track the optimal operating point. This review critically analyzes various MPPT algorithms that enable wind turbines to operate at their maximum power point, ensuring optimal utilization of available wind energy. The authors categorize MPPT techniques into conventional methods and advanced approaches. Conventional methods, such as Tip Speed Ratio (TSR), Power Signal Feedback (PSF), and Hill Climbing Search (HCS), have been widely used in commercial wind energy systems. These algorithms adjust the rotor speed or generator load to ensure that the wind turbine operates at the optimal tip speed ratio, thereby maximizing power output. However, these methods often exhibit limitations such as slow convergence, sensitivity to parameter variations, and susceptibility to wind turbulence, which can degrade overall system performance. To address these limitations, the study explores advanced MPPT algorithms that incorporate artificial intelligence (AI), adaptive control, and optimization techniques. These include Fuzzy Logic Control (FLC), Neural Networks (NN), Genetic Algorithms (GA), and Particle Swarm Optimization (PSO), which offer improved accuracy and adaptability under dynamically changing wind conditions. FLC-based MPPT systems dynamically adjust control parameters by mimicking human decision-making processes, enabling faster convergence and better robustness. Similarly, NN-based algorithms utilize historical wind data and learning models to predict optimal operating points, while GA and PSO optimize control actions by iteratively refining solutions through evolutionary and swarm-based approaches.

2.4 TITLE: Modelling and Design of PID Controller for Voltage Control of AC Hybrid Micro-Grid

AUTHOR: M. Alhamrouni, M. A. Hairullah, N. S. Omar, M. Salem, A. Jusoh, and T. Sutikno

YEAR: 2022

DESCRIPTION: M. Alhamrouni, M. A. Hairullah, N. S. Omar, M. Salem, A. Jusoh, and T. Sutikno present an in-depth study on the modeling and design of a Proportional-Integral-Derivative (PID) controller to regulate voltage levels in AC hybrid micro-grid systems. As hybrid micro-grids (HMGs) integrate multiple renewable energy sources such as solar, wind, and battery storage, maintaining voltage stability becomes a critical challenge due to the dynamic nature of power generation and consumption. The authors propose a robust PID control strategy to ensure that the voltage in the micro-grid remains within acceptable limits, thereby enhancing system stability and performance. The study begins by analyzing the voltage fluctuation problems inherent in AC hybrid micro-grid environments. These fluctuations result from variations in power generation, sudden load changes, and the intermittent nature of renewable sources. Conventional control methods often fail to maintain consistent voltage regulation in such dynamic environments. To address these challenges, the authors design and optimize a PID controller capable of dynamically adjusting system parameters to minimize voltage deviations and ensure smooth power delivery. The proposed PID controller uses real-time voltage feedback from the micro-grid to adjust the duty cycle of the voltage source inverter (VSI), thereby regulating the voltage levels effectively. The mathematical model of the AC hybrid micro-grid is developed using system identification techniques, which define the system dynamics and enable accurate controller design. The PID controller is then fine-tuned using optimization techniques such as Ziegler–Nichols and Cohen–Coon methods to achieve optimal proportional, integral, and derivative gains. Simulation results demonstrate that the proposed PID controller significantly enhances voltage stability, even under fluctuating load and generation conditions. The system effectively mitigates voltage deviations and reduces transient response time, ensuring that the micro-grid operates within safe voltage margins. Additionally, the authors compare the PID controller’s performance with other control methods, such as fuzzy logic controllers (FLC) and model predictive controllers (MPC), highlighting the superior response time and stability provided by the PID approach.

2.5 TITLE: A Comparative Study of High-Performance Robust PID Controller for Grid Voltage Control of Islanded Microgrid

AUTHOR: S. K. Sarkar, F. R. Badal, and S. K. Das

YEAR: 2021

DESCRIPTION: S. K. Sarkar, F. R. Badal, and S. K. Das present a comprehensive comparative analysis of various high-performance robust Proportional-Integral-Derivative (PID) controllers designed for grid voltage regulation in islanded microgrids. Islanded microgrids operate independently from the main grid, relying on distributed energy resources (DERs) such as solar photovoltaic (PV) systems, wind turbines, and battery storage. Due to the unpredictable and intermittent nature of renewable energy sources, maintaining voltage stability in islanded microgrids presents significant challenges. This study aims to evaluate and compare the performance of different PID control strategies to identify the most effective approach for ensuring consistent voltage regulation. The authors discuss the limitations of conventional PID controllers in handling voltage fluctuations in islanded microgrids, particularly under dynamic load variations and changing generation patterns. To overcome these limitations, they introduce robust PID controllers that incorporate advanced tuning techniques, such as Ziegler–Nichols, Cohen–Coon, and Particle Swarm Optimization (PSO), to enhance system stability and minimize voltage deviations. Each controller's robustness is tested under various operating conditions, including sudden load changes, renewable energy intermittency, and fault scenarios. The study models the islanded microgrid environment, incorporating distributed generation units and voltage source inverters (VSIs) to regulate the grid voltage. The PID controllers are integrated into the system to dynamically adjust the inverter’s duty cycle, ensuring optimal voltage regulation. A detailed comparative analysis is conducted to evaluate the performance of each control method based on metrics such as settling time, overshoot, steady-state error, and total harmonic distortion (THD). Simulation results reveal that robust PID controllers, particularly those tuned using PSO, outperform conventional PID controllers by providing faster response times, reduced overshoot, and improved voltage stability. The authors highlight that PSO-tuned PID controllers dynamically optimize control parameters to minimize voltage deviations, ensuring superior performance across diverse operating conditions.

2.6 TITLE: A Review on Applications of Model Predictive Control to Wind Turbines

AUTHOR: W. H. Lio, J. A. Rossiter, and B. L. Jones

YEAR: 2020

DESCRIPTION: W. H. Lio, J. A. Rossiter, and B. L. Jones provide a detailed review of the applications of Model Predictive Control (MPC) in wind turbine operations, focusing on its potential to enhance efficiency, safety, and grid stability. Wind turbines operate in highly dynamic and unpredictable environments where wind speed, turbulence, and grid conditions continuously fluctuate. Traditional control approaches often struggle to maintain optimal performance under such varying conditions. This review highlights the advantages of employing MPC as a more adaptive and predictive control strategy to address these challenges effectively. MPC is an advanced control technique that uses a mathematical model of the system to predict future outputs and optimize control actions over a defined prediction horizon. By solving a constrained optimization problem at each control step, MPC dynamically adjusts system parameters to minimize deviations from desired performance targets. The authors explore various applications of MPC in wind turbine systems, including rotor speed control, pitch angle adjustment, and generator torque regulation, all of which play a critical role in maximizing energy capture and minimizing structural stress. One of the key benefits of MPC highlighted in this review is its ability to handle multiple system constraints simultaneously, ensuring that turbine operations remain within safe limits while maximizing power generation. For instance, MPC dynamically adjusts the blade pitch angle in response to sudden wind gusts, preventing excessive loads and extending the lifespan of turbine components. Similarly, MPC can regulate generator torque to maintain an optimal tip-speed ratio (TSR), ensuring maximum power extraction from available wind resources. The review also discusses the application of multi-objective MPC frameworks that balance conflicting objectives, such as maximizing power output while minimizing mechanical stress on turbine blades and towers. Simulation results from various case studies demonstrate that MPC-based approaches outperform traditional PID controllers by reducing oscillations, improving tracking accuracy, and enhancing overall system robustness.

2.7 TITLE: Robust Model Predictive Control-Based Voltage Regulation Method for a Distribution System with Renewable Energy Sources and Energy Storage Systems

AUTHOR: Y. Xie, L. Liu, Q. Wu, and Q. Zhou

YEAR: 2022

DESCRIPTION: Y. Xie, L. Liu, Q. Wu, and Q. Zhou introduce a Robust Model Predictive Control (RMPC) framework to regulate voltage in power distribution systems that incorporate renewable energy sources (RES) and energy storage systems (ESS). As the integration of renewable energy into the grid increases, maintaining voltage stability becomes more challenging due to the unpredictable and fluctuating nature of solar and wind power. This study proposes an advanced RMPC strategy to ensure voltage stability by dynamically managing power injection from distributed energy resources and optimizing the operation of energy storage systems. The proposed RMPC approach models the distribution system dynamics while considering system uncertainties arising from variations in renewable energy generation and load demand. Unlike conventional PID controllers, which react to disturbances after they occur, RMPC proactively predicts system behavior over a finite time horizon and computes optimal control actions that minimize deviations from the desired voltage levels. The control strategy ensures that voltage limits are maintained under all operating conditions while respecting operational constraints of power inverters and energy storage devices. A distinguishing feature of the proposed RMPC framework is its ability to handle system uncertainties and constraints effectively. By incorporating robust optimization techniques, the controller mitigates the impact of prediction errors, ensuring that voltage regulation remains within safe margins even under severe fluctuations in renewable power generation. The system dynamically adjusts the reactive power compensation from distributed generators and battery inverters to stabilize voltage and balance active and reactive power flows in the distribution network. Simulation results presented in the study demonstrate that the RMPC approach outperforms traditional voltage regulation methods, such as droop control and decentralized PID control, by providing faster response times and improved voltage stability. The RMPC framework minimizes voltage deviations and reduces transient oscillations, enhancing overall system resilience. Furthermore, the authors compare the performance of the RMPC model under various scenarios, including sudden load changes, extreme weather conditions, and fluctuating renewable energy inputs, highlighting the controller’s robustness and adaptability.

2.8 TITLE: Model Predictive Control for Microgrid Functionalities: Review and Future Challenges

AUTHOR: F. Garcia-Torres, A. Zafra-Cabeza, C. Silva, S. Grieu, T. Darure, and A. Estanqueiro

YEAR: 2022

DESCRIPTION: F. Garcia-Torres, A. Zafra-Cabeza, C. Silva, S. Grieu, T. Darure, and A. Estanqueiro provide a comprehensive review of the application of Model Predictive Control (MPC) in microgrid functionalities, emphasizing its potential to enhance system stability, reliability, and energy efficiency. As microgrids integrate a mix of distributed energy resources (DERs), including solar, wind, and battery storage, ensuring optimal management and seamless operation under dynamic and unpredictable conditions becomes essential. The authors evaluate various MPC strategies implemented in microgrid environments and highlight emerging challenges that must be addressed for widespread adoption. The paper categorizes microgrid functionalities where MPC plays a pivotal role, including power flow management, voltage and frequency regulation, demand response optimization, and fault detection. MPC, by leveraging a predictive model, forecasts system behavior over a defined time horizon and computes optimal control actions to maintain system performance within operational limits. The study emphasizes that MPC can dynamically balance active and reactive power flows, regulate voltage, and manage energy storage systems (ESS) to optimize microgrid operation. One of the key advantages of MPC highlighted in the review is its ability to handle multiple objectives and constraints simultaneously. For instance, MPC can optimize the charging and discharging cycles of ESS to minimize power losses while maintaining voltage stability. Additionally, the authors demonstrate that MPC can effectively manage power exchange between the microgrid and the main grid, ensuring seamless islanding and grid synchronization during transition states. Simulation studies show that MPC-based approaches outperform traditional PID and droop control methods by reducing transient oscillations and improving response times. The review also explores the application of Robust and Stochastic MPC to account for uncertainties in renewable energy generation and load demand. Robust MPC ensures system stability by considering worst-case scenarios, while Stochastic MPC optimizes system performance by incorporating probabilistic forecasts of renewable energy availability. These approaches significantly enhance microgrid resilience and adaptability to fluctuating conditions.

2.9 TITLE: Multi-Objective Model Predictive Control of Doubly-Fed Induction Generators for Wind Energy Conversion

AUTHOR: J. Hu, Y. Li, and J. Zhu

YEAR: 2022

DESCRIPTION: J. Hu, Y. Li, and J. Zhu propose a Multi-Objective Model Predictive Control (MPC) framework for optimizing the performance of Doubly-Fed Induction Generators (DFIGs) in wind energy conversion systems. As wind energy becomes a prominent source of renewable energy, ensuring the stability and efficiency of wind turbines with DFIGs is critical due to their inherent sensitivity to wind speed variations and grid disturbances. The authors develop a multi-objective MPC strategy that dynamically balances conflicting objectives, such as maximizing power capture, minimizing mechanical stress, and ensuring grid compliance. The study begins by analyzing the challenges associated with controlling DFIGs in wind energy systems, highlighting issues such as fluctuating wind speeds, transient disturbances, and grid voltage variations. Conventional control approaches, including Proportional-Integral-Derivative (PID) and vector control techniques, often struggle to maintain optimal performance under such dynamic conditions. To overcome these limitations, the authors introduce an MPC framework that predicts future system behavior over a defined prediction horizon and computes optimal control actions to meet multiple performance objectives.

The proposed MPC strategy simultaneously optimizes three key objectives:

Maximizing Active Power Extraction: The controller ensures that the turbine operates at the optimal tip-speed ratio (TSR) to maximize power output under varying wind conditions.

Minimizing Mechanical Stress: The MPC reduces mechanical stress on the turbine shaft and gearbox by minimizing torque oscillations and load variations, thereby extending the lifespan of wind turbine components.

Maintaining Grid Stability: The controller dynamically regulates reactive power to maintain grid voltage stability and ensure compliance with grid codes during fault conditions.

A distinguishing feature of the proposed MPC framework is its ability to handle multiple system constraints effectively, including rotor speed limits, inverter capacity, and grid voltage limits.

2.10 TITLE: Optimisation of Grid-Connected Hybrid Photovoltaic–Wind–Battery System Using Model Predictive Control Design

AUTHOR: N. T. Mbungu, R. Naidoo, R. C. Bansal, and M. Bipath

YEAR: 2021

DESCRIPTION: N. T. Mbungu, R. Naidoo, R. C. Bansal, and M. Bipath present a comprehensive study on the optimization of grid-connected hybrid photovoltaic (PV)–wind–battery systems using a Model Predictive Control (MPC) design. As hybrid renewable energy systems (HRES) integrate multiple energy sources, maintaining optimal power flow and ensuring grid stability becomes increasingly complex due to the intermittent nature of solar and wind energy. This study proposes an advanced MPC-based approach to dynamically manage power generation, load demand, and battery storage, ensuring optimal system performance and minimizing operational costs. The authors highlight the challenges associated with hybrid renewable systems, including power fluctuations, battery degradation, and grid instability during varying environmental conditions. Conventional energy management strategies often fail to effectively balance power generation and load demand, leading to inefficient utilization of energy resources. To address these challenges, the authors propose an MPC framework that optimizes the operation of PV, wind, and battery systems by predicting future system states and adjusting control actions in real time.

The proposed MPC approach aims to achieve three primary objectives:

Maximizing Renewable Energy Utilization: The controller prioritizes energy from PV and wind sources to minimize dependency on the grid and reduce fossil fuel consumption.

Minimizing Battery Degradation: The MPC strategy optimally schedules battery charging and discharging cycles to prolong battery lifespan and prevent overcharging or deep discharging.

Ensuring Grid Stability: The system dynamically regulates power flow between the hybrid system and the grid to maintain voltage and frequency stability under varying load and generation conditions.

CHAPTER 3
SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

The conventional systems used for wind energy generation are typically based on standalone wind turbines that operate without intelligent control or feedback mechanisms. These systems often lack the ability to adapt to varying environmental conditions such as wind speed and temperature, resulting in inefficient power generation. In most cases, the energy produced is either directly supplied to the load or stored in batteries without any real-time monitoring of voltage levels or charging status. Overcharging or deep discharging of batteries in such setups can reduce battery life and compromise system safety. Furthermore, the absence of automation and sensor-based monitoring in these systems means that any maintenance or fault detection relies heavily on manual observation, which is not only time-consuming but also increases the risk of unexpected failures.

Traditional wind energy setups also do not include communication capabilities, limiting their usability in remote or unattended locations. These systems generally operate in isolation, without integration into a smart grid or IoT-based platform. Users are unable to access live performance data or control the system remotely, which poses significant limitations, especially in rural or offshore installations where physical access may be challenging. As a result, these systems are less responsive to fluctuating energy demands and environmental changes. Additionally, power regulation in these traditional systems is either absent or based on basic mechanical systems, making the energy output inconsistent and sometimes unsuitable for sensitive electronic appliances.

Another major drawback of existing systems is the inefficient load management. Loads are typically connected directly to the turbine or battery, and their operation cannot be optimized based on available energy. This leads to energy wastage or load disruption during low wind conditions. Moreover, the absence of inverters or advanced power electronics limits the usability of the generated energy, as many traditional setups deliver only DC power which is not compatible with standard AC appliances used in households or industries. The lack of a proper energy management unit or user interface in such systems further restricts the users from understanding the performance, optimizing the load schedule, or analyzing energy usage trends. These limitations highlight the need for an advanced, intelligent wind energy system that incorporates automation, IoT, and real-time monitoring capabilities for effective and sustainable energy generation.

DISADVANTAGE:

Inefficient Power Regulation: Traditional systems lack advanced power regulation mechanisms, leading to unstable energy output. Fluctuations in wind speed cause inconsistent power generation. This makes them unsuitable for powering sensitive devices.

Manual Operation: Existing systems often require manual intervention for operation and maintenance. This increases the risk of human error and delays in addressing faults. Automation is typically not integrated, reducing system efficiency.

Limited Remote Monitoring: Most conventional wind energy systems do not feature remote monitoring. This makes it difficult for users to track performance or control the system without being physically present. Lack of connectivity also hinders timely fault detection.

Battery Management Issues: In traditional systems, there is limited monitoring of battery health and charge levels. Overcharging or deep discharging can shorten battery life. The absence of intelligent control leads to inefficient energy storage.

Lack of Adaptability: These systems do not adapt to varying environmental conditions such as wind speed or temperature. Consequently, they operate suboptimally during fluctuations, leading to wasted energy or equipment damage.

Energy Wastage: Without load management capabilities, traditional systems often fail to optimize energy usage. This results in excess energy being wasted when demand is low. It also leads to the overuse of resources during peak production times.

3.2 PROPOSED SYSTEM

The proposed system aims to modernize traditional wind energy generation setups by incorporating advanced features such as real-time monitoring, remote control, intelligent power regulation, and automated load management. By integrating an Arduino Uno microcontroller with IoT connectivity, the system enables efficient energy generation and storage while ensuring optimal performance under varying environmental conditions. The system's core components include a wind turbine, battery storage, voltage sensors, a temperature sensor, an inverter, a relay, and a user interface that allows both local and remote monitoring. This comprehensive design maximizes energy generation, improves system reliability, and reduces energy wastage, all while being cost-effective and easy to maintain.

The wind turbine in the proposed system is designed to harness wind energy efficiently and convert it into electrical power. The system's performance is optimized by continuously measuring key environmental parameters such as wind speed, temperature, and battery voltage. A temperature sensor monitors the wind conditions, ensuring the turbine operates under optimal conditions. By adjusting the turbine's operation based on real-time data, the system maximizes energy production even during fluctuating wind speeds. Voltage sensors on the wind turbine and the battery ensure that the system's energy storage is effectively managed, preventing overcharging or deep discharging of the battery. This continuous monitoring ensures the longevity of the battery and improves overall system efficiency.

Another essential feature of the proposed system is its power regulation mechanism, which guarantees stable energy output despite variations in wind speed. This mechanism adjusts the wind turbine's operation and energy storage capacity in real time, ensuring that the energy supplied to the load is steady and consistent. Additionally, the inclusion of an inverter in the system is a significant upgrade. The inverter converts the DC power generated by the wind turbine into AC power, which is suitable for household and industrial applications. This feature enhances the system's versatility and allows seamless integration into existing electrical grids or stand-alone load systems that require AC power. The inverter also ensures that energy is efficiently utilized by converting surplus power for storage or supplying it directly to the load.

The system also integrates a relay to automate both the wind turbine's operation and the load management. The relay allows the system to be more energy-efficient by turning the wind turbine on or off based on real-time data, ensuring that energy is only generated when it is needed. This prevents unnecessary operation during unfavorable wind conditions and optimizes the energy consumption of the load by powering it only when required. The relay also enhances safety by allowing the system to operate autonomously, reducing the need for human intervention and manual control.

To provide greater control and oversight, the proposed system features a user-friendly interface that displays key parameters such as wind energy, battery status, and overall system functionality. This interface is designed to help users make informed decisions regarding energy usage and management. By providing real-time data, the interface allows users to monitor the health of the system, identify potential issues, and adjust settings to optimize performance. The system also features remote control capabilities, which enable users to control the wind turbine and load operation from anywhere, making it convenient for users who cannot always be on-site. This feature significantly improves the system's usability and helps users manage energy consumption more efficiently, even in remote or off-grid locations.

A crucial component of the proposed system is the integration of IoT connectivity, which enables the transmission of real-time data to a cloud-based platform. This feature allows users and relevant authorities to access system data remotely, providing the ability to monitor the performance of multiple systems in different locations. With IoT connectivity, the system's performance can be continuously tracked, and any irregularities or faults can be quickly detected and addressed. Additionally, the cloud-based platform provides a secure and centralized repository for system data, facilitating data analysis, trend monitoring, and predictive maintenance. By integrating IoT, the system can also send automated notifications and alerts, ensuring users are informed of any system irregularities or required maintenance actions.

Furthermore, the proposed system's modular design allows for scalability and easy adaptation to different locations and requirements. The system can be tailored to suit the specific needs of different wind conditions, energy requirements, and environmental factors. Whether for residential, industrial, or agricultural applications, the system can be customized to meet energy demands while optimizing wind energy production. This adaptability ensures that the system remains viable in diverse geographical areas, offering a cost-effective solution for a wide range of users. The open-source nature of the Arduino platform also encourages further innovation and improvements by researchers and developers, ensuring the system can evolve over time.

The proposed wind energy system's main advantages lie in its intelligent energy management, remote monitoring capabilities, and scalability. By incorporating smart technologies, the system offers real-time insights into energy generation and usage, ensuring the system operates at peak efficiency. Moreover, the integration of IoT ensures that the system can be remotely managed, offering convenience and enhancing user experience. These features make the system suitable for both urban and rural applications, where conventional wind energy setups may not be as effective due to limited manual control or monitoring. The proposed system addresses the main limitations of traditional wind energy systems, providing an efficient, reliable, and user-friendly solution for modern wind energy applications.

ADVANTAGE:

Efficient Energy Management: The system continuously monitors and regulates the wind turbine's output and battery storage, ensuring optimal energy utilization. By preventing overcharging and undercharging, it prolongs the battery's lifespan. This results in cost-effective and sustainable energy use.

Remote Monitoring and Control: Through IoT connectivity, users can access real-time system data from anywhere, allowing for remote monitoring and control. This feature enhances convenience and enables quick responses to system changes. Users can optimize energy use even when not on-site.

Automated Load Management: The relay system automates turbine operation based on real-time data, ensuring energy is only used when needed. This prevents wasteful operation during unfavorable wind conditions. It optimizes both energy generation and consumption for better overall efficiency.

Seamless Integration: The inverter converts generated DC power into AC, making the system compatible with standard household or industrial electrical grids. This integration enables the use of wind energy for a wide range of applications. It eliminates the need for additional power conversion equipment.

Scalability and Customization: The modular design of the system allows for easy adaptation to different locations and energy needs. It can be tailored to suit various environments, making it suitable for urban, rural, or off-grid applications. This flexibility ensures the system's broad applicability and future-proofing.

Real-Time Data Visualization: The user-friendly interface provides detailed insights into key parameters such as battery voltage, wind energy, and system status. This data empowers users to make informed decisions regarding energy usage. Real-time monitoring ensures that the system operates efficiently, even in fluctuating conditions.

BLOCK DIAGRAM:

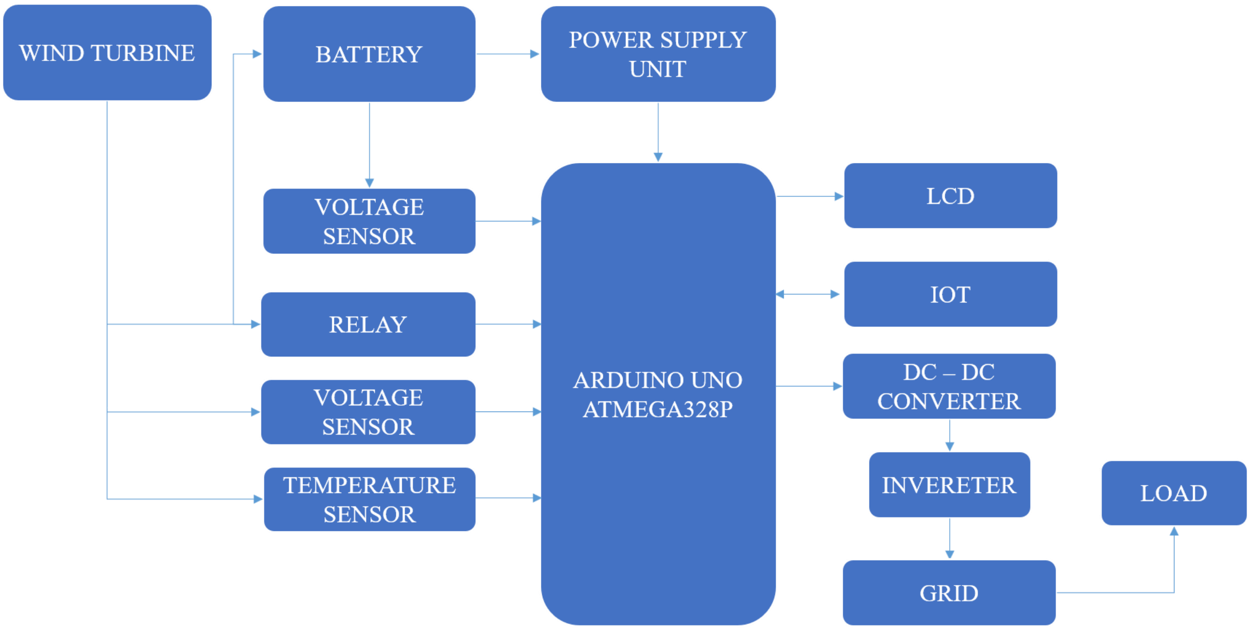

Fig 3.1 Hardware Block Diagram

The block diagram of the proposed IoT-based intelligent renewable energy management system illustrates the integration of key components for optimizing wind energy generation, battery storage, and grid voltage regulation. The system comprises a wind turbine that generates electricity, which is monitored by voltage sensors to track energy levels. A temperature sensor monitors environmental conditions to ensure optimal operating limits. The generated direct current (DC) power is regulated by a DC-DC converter, maintaining consistent voltage levels for efficient battery charging. The battery storage unit stores excess energy generated during peak wind periods and supplies power to the load when wind energy is insufficient. A DC-AC inverter converts the stored DC power into alternating current (AC), enabling seamless operation of household or industrial loads. The Arduino Uno microcontroller acts as the central control unit, processing real-time data from the sensors and dynamically adjusting system parameters to optimize power flow. An IoT interface provides remote monitoring and control, enabling users to track energy levels, system status, and environmental conditions. A relay controls the connection between the wind turbine, battery, and load, ensuring safe power distribution. The LCD display shows real-time system data, while the buzzer alerts users in case of system faults or abnormalities.

CIRCUIT DIAGRAM

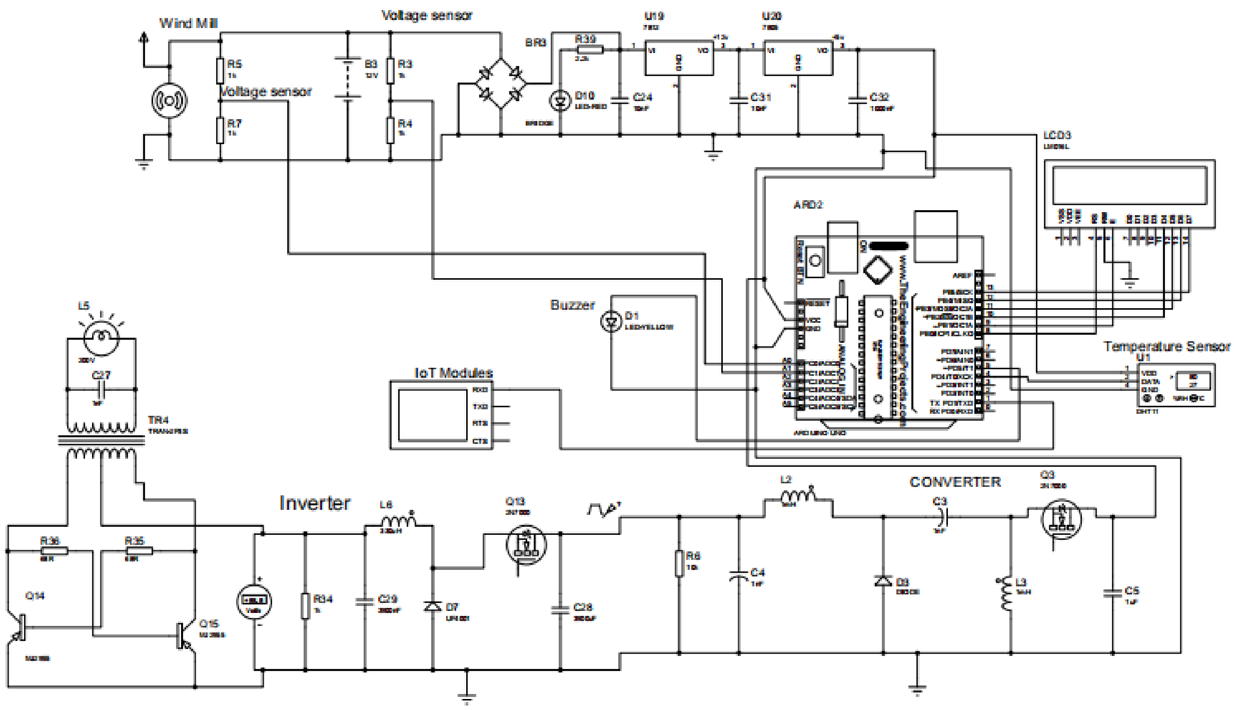

Fig 3.2 Hardware Circuit Diagram

The Arduino Uno serves as the central microcontroller in this system. It manages all the inputs and outputs to control the different components, process the sensor data, and communicate with the IoT module. The LCD display is connected to the Arduino to provide real-time data on the energy generation, battery voltage, and wind conditions. The pins (8, 9, 10, 11, 12, and 13) of the Arduino are used to control the LCD screen, displaying various parameters such as wind energy, battery charge level, and the operational status of the wind turbine. The LCD helps users monitor the system's real-time performance without the need for additional interfaces. It ensures a user-friendly experience by showing information directly related to the system's current conditions.

The wind model refers to a wind turbine or wind energy generation model in the system. A voltage sensor connected to A0 monitors the voltage output from the wind turbine. This sensor plays a critical role in measuring the power generated by the wind turbine. The Arduino reads the voltage levels from this sensor through analog input pin A0. The system can then process this voltage to calculate the available power and monitor how efficiently the wind turbine is generating energy. By continuously monitoring the wind turbine's output, the system ensures that energy is generated optimally, especially under varying wind conditions. The data obtained from the sensor helps the system adjust the operation of the wind turbine accordingly.

The battery voltage sensor, connected to A2, is crucial for monitoring the state of charge of the battery storage. The system keeps track of the battery voltage to ensure the battery is not overcharged or undercharged, thus preventing damage and optimizing its lifespan. The Arduino reads the voltage data through analog input pin A2 and compares it against predefined thresholds. If the voltage crosses a certain limit (either high or low), the system triggers alerts or adjusts the turbine's operation accordingly to ensure safe charging or discharging of the battery. This battery management feature is essential for ensuring that the energy stored is utilized efficiently and safely.

The inverter converter in the system converts the DC (Direct Current) generated by the wind turbine (and stored in the battery) into AC (Alternating Current). This is essential for integrating renewable energy with conventional AC-powered devices or the grid. The inverter ensures that the energy produced by the wind turbine is usable for everyday applications, including powering household or industrial loads. The inverter operates by taking the DC power from the battery and converting it into AC power, making it compatible with standard electrical appliances. The Arduino controls this inverter by switching it on and off based on real-time data from the sensors, optimizing energy output and ensuring smooth power conversion.

The DC to DC converter is used to step up or step down the DC voltage from the battery to the required level to power specific components in the system or even for direct use in applications. The DC to DC converter ensures that the system operates at a stable voltage level, regardless of the fluctuations in the battery's charge. It is also crucial in managing energy distribution to other subsystems within the renewable energy system. For example, if the system requires 5V or 12V to operate other components like sensors or microcontrollers, the DC to DC converter adjusts the voltage accordingly, ensuring that the components receive the required voltage without overloading or damaging them. It helps in improving the overall efficiency and adaptability of the system.

The IoT module (often based on Wi-Fi, such as ESP8266 or ESP32) enables remote monitoring and control of the entire system. This module connects the Arduino to the internet, allowing real-time data to be transmitted to a cloud platform or mobile app. With the IoT module, users can monitor the wind energy system from anywhere using a smartphone or computer, making it a smart, connected energy management system. Additionally, the IoT module allows the system to receive remote commands for operation adjustments, such as turning the wind turbine or the inverter on or off based on current wind conditions or energy storage levels. This real-time remote monitoring is especially useful for users in remote or off-grid locations, ensuring continuous operation without direct physical intervention.

CHAPTER-4

SOFTWARE REQUIREMENTS

4.1 ARDUINO IDE

The Arduino integrated development environment (IDE) is a cross-platform application (for Windows, macOS, Linux) that is written in the programming language Java. It is used to write and upload programs to Arduino board. The source code for the IDE is released under the GNU General Public License, version 2. The Arduino IDE supports the languages C and C++ using special rules of code structuring. The Arduino IDE supplies a software library from the Wiring project, which provides many common input and output procedures. User-written code only requires two basic functions, for starting the sketch and the main program loop, that are compiled and linked with a program stub main() into an executable cyclic executive program with the GNU tool chain, also included with the IDE distribution. The Arduino IDE employs the program argued to convert the executable code into a text file in hexadecimal encoding that is loaded into the Arduino board by a loader program in the board's firmware. Arduino is an open-source electronics platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board

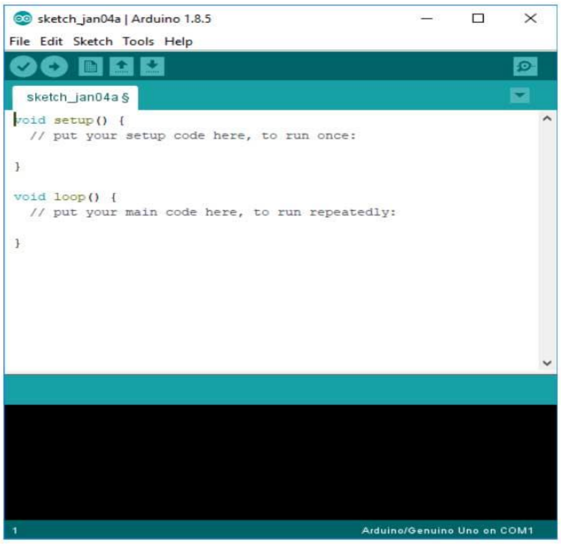

Fig4.1 Arduino IDE

The Arduino IDE is incredibly minimalistic, yet it provides a near-complete environment for most Arduino-based projects. The top menu bar has the standard options, including “File” (new, load save, etc.), “Edit” (font, copy, paste, etc.), “Sketch” (for compiling and programming), “Tools” (useful options for testing projects), and “Help”. The middle section of the IDE is a simple text editor that where you can enter the program code. The bottom section of the IDE is dedicated to an output window that is used to see the status of the compilation, how much memory has been used, any errors that were found in the program, and various other useful messages. Projects made using the Arduino are called sketches, and such sketches are usually written in a cut-down version of C++ (a number of C++ features are not included). Because programming a microcontroller is somewhat different from programming a computer, there are a number of device-specific libraries (e.g., changing pin modes, output data on pins, reading analog values, and timers). This sometimes confuses users who think Arduino is programmed in an “Arduino language.” However, the Arduino is, in fact, programmed in C++. It just uses unique libraries for the device. The Arduino Integrated Development Environment - or Arduino Software (IDE) - contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions and a series of menus. It connects to the Arduino and Genuine hardware to upload programs and communicate with them. Programs written using Arduino Software (IDE) are called **sketches**. These sketches are written in the text editor and are saved with the file extension .ino. The editor has features for cutting/pasting and for searching/replacing text. The message area gives feedback while saving and exporting and also displays errors. The console displays text output by the Arduino Software (IDE), including complete error messages and other information. The bottom right hand corner of the window displays the configured board and serial port. The toolbar buttons allow you to verify and upload programs, create, open, and save sketches, and open the serial monitor.

LIBRARIES

Libraries provide extra functionality for use in sketches, e.g. working with hardware or manipulating data. To use a library in a sketch, select it from the **Sketch > Import Library** menu. This will insert one or more **#include** statements at the top of the sketch and compile the library with your sketch. Because libraries are uploaded to the board with your sketch, they increase the amount of space it takes up. If a sketch no longer needs a library, simply delete its **#include** statements from the top of your code. There is a list of libraries in the reference. Some libraries are included with the Arduino software. Others can be downloaded from a variety of sources or through the Library Manager. Starting with version 1.0.5 of the IDE, you do can import a library from a zip file and use it in an open sketch.

CONNECTING THE ARDUINO

Connecting an Arduino board to your PC is quite simple On Windows:

1. Plug in the USB cable - one end to the PC, and one end to the Arduino board.
2. When prompted, select "Browse my computer for driver" and then select the folder to which you extracted your original Arduino IDE download.
3. You may receive an error that the board is not a Microsoft certified device - select “Install anyway.”
4. Your bord should now be ready for programming.

When programming your Arduino board it is important to know what COM port the Arduino is using on your PC. On Windows, navigate to Start->Devices and Printers, and look for the Arduino. The COM port will be displayed underneath. Alternatively, the message telling you that the Arduino has been connected successfully in the lower-left hand corner of your screen usually specifies the COM port is it using.

PREPARING THE BOARD

Before loading any code to your Arduino board, you must first open the IDE. Double click the Arduino .exe file that you downloaded earlier. A blank program, or "sketch," should open. The Blink example is the easiest way to test any Arduino board. Within the Arduino window, it can be found under `File->Examples->Basics->Blink`.

Before the code can be uploaded to your board, two important steps are required.

1. Select your Arduino from the list under `Tools->Board`. The standard board used in RBE 1001, 2001, and 2002 is the Arduino Mega 2560, so select the "Arduino Mega 2560 or Mega ADK" option in the dropdown.

2. Select the communication port, or COM port, by going to `Tools->Serial Port`.

If you noted the COM port your Arduino board is using, it should be listed in the dropdown menu. If not, your board has not finished installing or needs to be reconnected.

LOADING CODE

The upper left of the Arduino window has two buttons: A checkmark to Verify your code, and a right-facing arrow to upload it. Press the right arrow button to compile and upload the Blink example to your Arduino board. The black bar at the bottom of the Arduino window is reserved for messages indicating the success or failure of code uploading. A "Completed Successfully" message should appear once the code is done uploading to your board. If an error message appears instead, check that you selected the correct board and COM port in the Tools menu, and check your physical connections. If uploaded successfully, the LED on your board should blink on/off once every second. Most Arduino boards have an LED prewired to pin 13.

It is very important that you do not use pins 0 or 1 while loading code. It is recommended that you do not use those pins ever. Arduino code is loaded over a serial port to the controller. Older models use an FTDI chip which deals with all the USB specifics. Newer models have either a small AVR that mimics the FTDI chip or a built-in USB-to-serial port on the AVR micro-controller itself.

4.2 PROTEUS

The Proteus Design Suite is a proprietary software tool suite used primarily for electronic design automation. The software is used mainly by electronic design engineers and technicians to create schematics and electronic prints for manufacturing printed circuit boards. Proteus is design software developed by Lab center Electronics for electronic circuit simulation, schematic capture and PCB design. Its simplicity and user friendly design made it popular among electronics hobbyists. Proteus is commonly used for digital simulations such as microcontrollers and microprocessors. It can simulate LED, LDR, USB Communication. Proteus is a simulation and design software tool developed by Lab center Electronics for Electrical and Electronic circuit design. It also possess 2D CAD drawing feature. It deserves to bear the tagline “From concept to completion”.

ABOUT PROTEUS

- It is a software suite containing schematic, simulation as well as PCB designing.
- ISIS is the software used to draw schematics and simulate the circuits in real time. The simulation allows human access during run time, thus providing real time simulation.
- ARES is used for PCB designing. It has the feature of viewing output in 3D view of the designed PCB along with components.
- The designer can also develop 2D drawings for the product.

FEATURES

ISIS has wide range of components in its library. It has sources, signal generators, measurement and analysis tools like oscilloscope, voltmeter, ammeter etc., probes for real time monitoring of the parameters of the circuit, switches, displays, loads like motors and lamps, discrete components like resistors, capacitors, inductors, transformers, digital and analog Integrated circuits, semi-conductor switches, relays, microcontrollers, processors, sensors etc.

HISTORY

The first version of what is now the Proteus Design Suite was called PC-B and was written by the company chairman, John Jameson, for DOS in 1988. Schematic Capture support followed in 1990, with a port to the Windows environment shortly thereafter. Mixed mode SPICE Simulation was first integrated into Proteus in 1996 and microcontroller simulation then arrived in Proteus in 1998. Shape based auto routing was added in 2002 and 2006 saw another major product update with 3D Board Visualization. More recently, a dedicated IDE for simulation was added in 2011 and MCAD import/export was included in 2015. Support for high speed design was added in 2017. Feature led product releases are typically biannual, while maintenance based service packs are released as required.

PRODUCT MODULES

The Proteus Design Suite is a Windows application for schematic capture, simulation, and PCB (Printed Circuit Board) layout design. It can be purchased in many configurations, depending on the size of designs being produced and the requirements for microcontroller simulation. All PCB Design products include an auto-router and basic mixed mode SPICE simulation capabilities.

SCHEMATIC CAPTURE

Schematic capture in the Proteus Design Suite is used for both the simulation of designs and as the design phase of a PCB layout project. It is therefore a core component and is included with all product configurations.

MICROCONTROLLER SIMULATION

The micro-controller simulation in Proteus works by applying either a hex file or a debug file to the microcontroller part on the schematic. It is then co-simulated along with any analog and digital electronics connected to it. This enables its use in a broad spectrum of project prototyping in areas such as motor control, temperature control and user interface design. It also finds use in the general hobbyist community and, since no hardware is required, is convenient to use as a training or teaching tool.

SUPPORT IS AVAILABLE FOR CO-SIMULATION OF

- Microchip Technologies PIC10, PIC12, PIC16, PIC18, PIC24, dsPIC33 Microcontrollers.
- Atmel AVR (and Arduino), 8051 and ARM Cortex-M3 Microcontrollers
- NXP 8051, ARM7, ARM Cortex-M0 and ARM Cortex-M3 Microcontrollers.
- Texas Instruments MSP430, PICCOLO DSP and ARM Cortex-M3 Microcontrollers.
- Parallax Basic Stamp, Free scale HC11, 8086 Microcontrollers.

PCB Design

The PCB Layout module is automatically given connectivity information in the form of a net list from the schematic capture module. It applies this information, together with the user specified design rules and various design automation tools, to assist with error free board design. PCB's of up to 16 copper layers can be produced with design size limited by product configuration.

3D Verification

The 3D Viewer module allows the board under development to be viewed in 3D together with a semi-transparent height plane that represents the boards enclosure. STEP output can then be used to transfer to mechanical CAD software such as Solid works or Autodesk for accurate mounting and positioning of the board.

PROTEUS SIMULATIONS

Proteus's simulation feature. Many of the components in Proteus can be simulated. There are two options for simulating: Run simulator and advance frame by frame. The "Run simulator" option simulates the circuit in a normal speed (If the circuit is not heavy). "Advance frame by frame" option advances to next frame and waits till you click this button for the next time. This can be useful for debugging digital circuits.

You can also simulate microcontrollers. The microcontrollers which can be simulated include PIC24, dsPIC33, 8051, Arduino, ARM7 based microcontrollers. You can download the compilers for Proteus or use different compiler and dump the hex files in the microcontroller in Proteus. You can even interact in real-time with the simulation using switches, resistors, LDRs, etc. There are even virtual voltmeter, ammeter, oscilloscope, logic analyser, etc.

ADVANTAGES OF PROTEUS ISIS PROFESSIONAL

1. It gives the proper idea and implementation of your code and circuit before implementing on hardware.
2. It reduces the time on creating hardware and testing your errors directly on hardware. You can analyses your circuit and code both on Proteus and find the errors encountering before implementing on hardware.
3. Reduces project cost and software dependency.

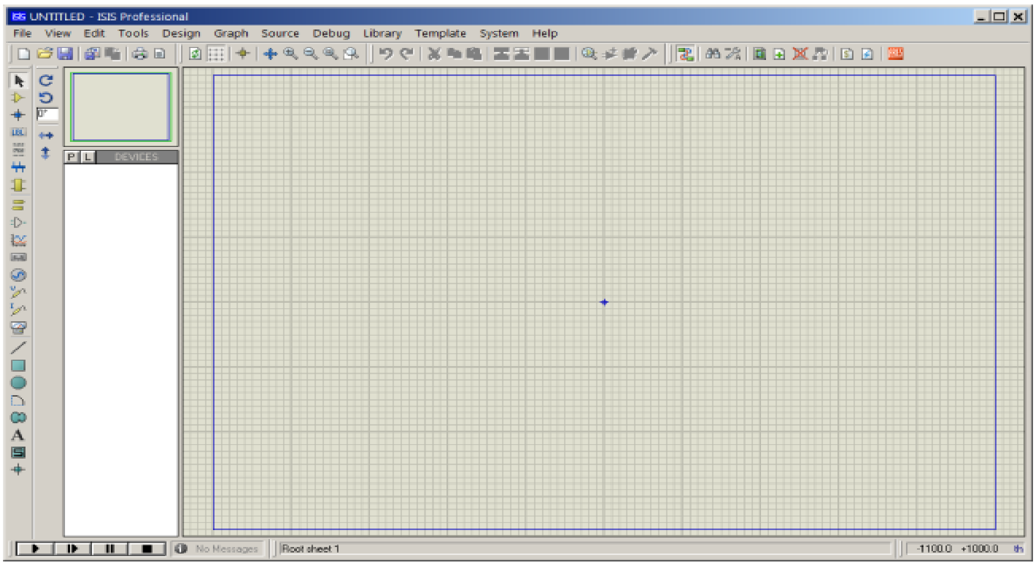

Fig 4.2 Proteus

CHAPTER 5

HARDWARE REQUIREMENTS

5.1 ARDUINO UNO

The Arduino UNO is an open-source microcontroller board based on the Microchip ATmega328P microcontroller and developed by Arduino.cc. The board is equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards (shields) and other circuits. The board has 14 Digital pins, 6 Analog pins, and programmable with the Arduino IDE (Integrated Development Environment) via a type B USB cable. It can be powered by a USB cable or by an external 9 volt battery, though it accepts voltages between 7 and 20 volts. It is also similar to the Arduino Nano and Leonardo. The hardware reference design is distributed under a Creative Commons Attribution Share-Alike 2.5 license and is available on the Arduino website. Layout and production files for some versions of the hardware are also available. "Uno" means one in Italian and was chosen to mark the release of Arduino Software (IDE) 1.0. The Uno board and version 1.0 of Arduino Software (IDE) were the reference versions of Arduino, now evolved to newer releases. The Uno board is the first in a series of USB Arduino boards, and the reference model for the Arduino platform. The ATmega328 on the Arduino Uno comes preprogrammed with a boot loader that allows uploading new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol. The Uno also differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it uses the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial converter

Fig 5.1 Arduino UNO

The Arduino project started at the Interaction Design Institute Ivrea (IDII) in Ivrea, Italy. At that time, the students used a BASIC Stamp microcontroller at a cost of $100, a considerable expense for many students. In 2003 Hernando Barragán created the development platform Wiring as a Master's thesis project at IDII, under the supervision of Massimo Banzi and Casey Reas, who are known for work on the Processing language. The project goal was to create simple, low-cost tools for creating digital projects by non-engineers. The Wiring platform consisted of a printed circuit board (PCB) with an ATmega168 microcontroller, an IDE based on Processing and library functions to easily program the microcontroller. In 2003, Massimo Banzi, with David Mellis, another IDII student, and David Cuartielles, added support for the cheaper ATmega8 microcontroller to Wiring. But instead of continuing the work on Wiring, they forked the project and renamed it Arduino. Early arduino boards used the FTDI USB-to-serial driver chip and an ATmega168. The Uno differed from all preceding boards by featuring the ATmega328P microcontroller and an ATmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial converter.

SPECIFICATION

1. Microcontroller: Microchip ATmega328P
2. Operating Voltage: 5 Volt
3. Input Voltage: 7 to 20 Volts

4. Digital I/O Pins: 14 (of which 6 provide PWM output)
5. Analog Input Pins: 6
6. DC Current per I/O Pin: 20 mA
7. DC Current for 3.3V Pin: 50 mA
8. Flash Memory: 32 KB of which 0.5 KB used by boot loader
9. SRAM: 2 KB
10. EEPROM: 1 KB
11. Clock Speed: 16 MHz
12. Length: 68.6 mm
13. Width: 53.4 mm
14. Weight: 25 g

COMMUNICATION

The Arduino/Genuino Uno has a number of facilities for communicating with a computer, another Arduino/Genuino board, or other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An ATmega16U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The 16U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, on Windows, a .inf file is required. The Arduino Software (IDE) includes a serial monitor which allows simple textual data to be sent to and from the board. The RX and TX LEDs on the board will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer (but not for serial communication on pins 0 and 1). A Software Serial library allows serial communication on any of the Uno's digital pins

PINS General Pin functions

- **LED:** There is a built-in LED driven by digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.
- **VIN:** The input voltage to the Arduino/Genuino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- **5V:** This pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7 - 20V), the USB connector (5V), or the VIN pin of the board (7-20V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage the board.
- **3V3:** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- **GND:** Ground pins.
- **IOREF:** This pin on the Arduino/Genuino board provides the voltage reference with which the microcontroller operates. A properly configured shield can read the IOREF pin voltage and select the appropriate power source or enable voltage translators on the outputs to work with the 5V or 3.3V.
- **Reset:** Typically used to add a reset button to shields which block the one on the board.

SPECIAL PIN FUNCTIONS

Each of the 14 digital pins and 6 Analog pins on the Uno can be used as an input or output, using pinMode (), digitalWrite(), and digitalRead() functions. They operate at 5 volts. Each pin can provide or receive 20 mA as recommended operating condition and has an internal pull-up resistor (disconnected by default) of 20-50k ohm. A maximum of 40mA is the value that must not be exceeded on any I/O pin to avoid permanent damage to the microcontroller. The Uno has 6 analog inputs, labeled A0 through A5, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and the analogReference() function.

IN ADDITION, SOME PINS HAVE SPECIALIZED FUNCTIONS:

- **Serial / UART:** pins 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.
- **External Interrupts:** pins 2 and 3. These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value.
- **PWM (Pulse Width Modulation):** 3, 5, 6, 9, 10, and 11 Can provide 8-bit PWM output with the analogWrite () function.
- **SPI (Serial Peripheral Interface):** 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support SPI communication using the SPI library.
- **TWI (Two Wire Interface) / I²C:** A4 or SDA pin and A5 or SCL pin. Support TWI communication using the Wire library.
- **AREF (Analog REFerence):** Reference voltage for the analog inputs

5.2 NODEMCU (ESP8266)

The ESP8266 is a low-cost, highly integrated Wi-Fi chip that has become one of the most popular choices for Internet of Things (IoT) applications. Developed by Espressif Systems, it offers Wi-Fi capabilities to embedded systems at an affordable price, making it an excellent choice for hobbyists, makers, and developers building connected devices. Whether it’s a simple home automation system, a weather station, or an industrial IoT device, the ESP8266 can handle a variety of tasks and is supported by a large community of developers.

In this guide, we'll go over a detailed explanation of the ESP8266 chip, its specifications, pin configuration, and common applications, giving you a comprehensive understanding of this powerful IoT tool.

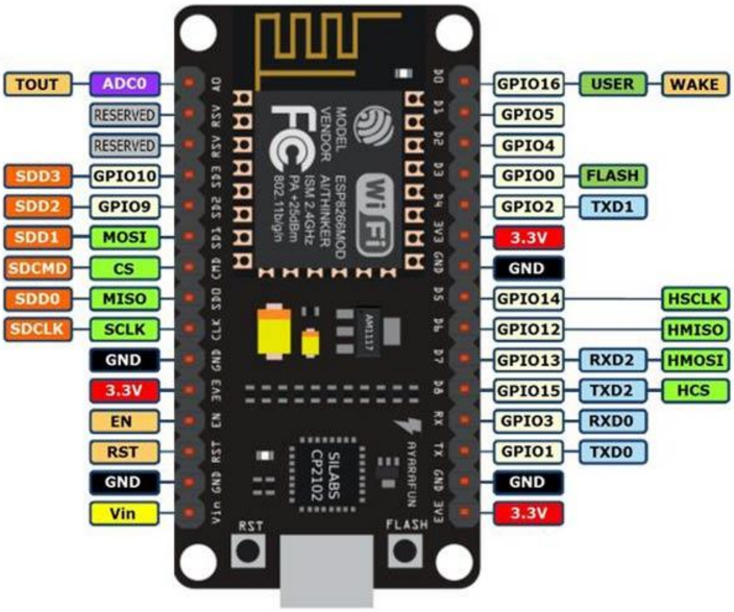

Fig 5.2 NODEMCU (ESP8266)

INTRODUCTION TO ESP8266

The ESP8266 is a 32-bit RISC-based microcontroller with an integrated Wi-Fi module. Espressif designed this chip with the goal of providing a low-cost yet highly functional platform for embedding wireless internet connectivity into everyday devices. This chip offers a combination of Wi-Fi communication, processing power, and a flexible input/output interface, all packed into a small form factor, making it ideal for embedded systems and IoT devices.

Originally, the ESP8266 was used primarily as a Wi-Fi module that could connect to a host microcontroller, but with the introduction of the ESP-12 and ESP-01 modules, it became a standalone microcontroller capable of running user applications. It operates in client mode (connecting to a Wi-Fi network) as well as access point mode (creating its own Wi-Fi network).

The versatility and functionality of the ESP8266, coupled with its low cost and ease of use, have made it a go-to solution for a wide range of applications. It is supported by numerous development environments and libraries, making it beginner-friendly while still being powerful enough for advanced projects.

SPECIFICATIONS OF THE ESP8266

The ESP8266 is a highly capable microcontroller with a set of impressive specifications, making it suitable for IoT applications. Below is a detailed breakdown of the ESP8266's specifications.

PROCESSOR AND PERFORMANCE

- **Architecture:** 32-bit RISC processor.
- **Clock Speed:** Typically runs at 80 MHz, though some models support up to 160 MHz.
- **Flash Memory:** The ESP8266 typically comes with 1 MB to 16 MB of Flash memory, depending on the variant.
- **RAM:** The chip contains 64 KB of instruction RAM and 96 KB of data RAM for executing applications.
- **CPU:** The ESP8266 features a Tensilica L106 microprocessor.

WIRELESS CONNECTIVITY

- **Wi-Fi Standard:** 802.11 b/g/n with support for both Station (STA) and Access Point (AP) modes.
- **Wi-Fi Speed:** It supports speeds up to 72.2 Mbps in 802.11n mode.
- **Security:** Supports WEP, WPA, and WPA2 security protocols.
- **Frequency Range:** Operates in the 2.4 GHz band.

I/O CAPABILITIES

- **GPIO Pins:** Depending on the module (e.g., ESP-12, NodeMCU), the ESP8266 has up to 17 GPIO pins.
- These pins can be configured for digital input/output, PWM, I2C, SPI, ADC (Analog-to-Digital Conversion), and interrupts.
- **ADC:** The chip has a 10-bit ADC (Analog to Digital Converter) with a maximum voltage range of 1V.
- **UART:** The ESP8266 supports two UART interfaces for serial communication.
- **PWM:** Pulse-width modulation (PWM) can be used for controlling motor speed or brightness of LEDs.
- **SPI/I2C:** The chip also supports SPI and I2C communication protocols, allowing easy interfacing with sensors and peripherals.

POWER CONSUMPTION

Operating Voltage: The ESP8266 operates at 3.3V, and care must be taken not to apply voltages higher than 3.3V to any of its pins.

Power Consumption: It is designed to be energy-efficient, with deep sleep and modem sleep modes available, which reduce power consumption significantly.

Typical Current Consumption: Active Mode: 80mA - 200mA (depending on the Wi-Fi activity).

Deep Sleep: ~20µA.

PROGRAMMING AND DEVELOPMENT

Development Platforms: The ESP8266 is most commonly programmed using the Arduino IDE, which offers a wide range of libraries and examples to get started quickly. The chip can also be programmed using other development environments such as MicroPython, NodeMCU (Lua), or PlatformIO.

Built-in Flash: The chip typically comes with 1MB to 16MB flash memory, enabling storage for both the application and Wi-Fi firmware.

ESP8266 PINOUT AND GPIO CONFIGURATION

The pinout for the ESP8266 can vary depending on the specific module, such as the ESP-01, ESP-12, or NodeMCU development board. The following is a general pin description for the ESP-12 module, one of the most popular variants.

ESP8266 Pin Description (ESP-12)

- GPIO0 (D3): Used as a digital I/O pin or for boot mode selection.
- GPIO1 (TX): UART Transmit (TX) pin for serial communication.
- GPIO2 (D4): Digital I/O pin, also used for PWM output.
- GPIO3 (RX): UART Receive (RX) pin for serial communication.
- GPIO4 (D2): Digital I/O pin, can be used with I2C or PWM.
- GPIO5 (D1): Another digital I/O pin, typically used for I2C or SPI.
- GPIO6: Flash Memory pin, typically not used for general I/O.
- GPIO7: Digital I/O pin, used for general purposes or in SPI communication.
- GPIO8: Similar to GPIO6 and 7, used for Flash Memory.
- GPIO9: Often used for Flash purposes, typically not used for general I/O.
- GPIO10: Same as GPIO9, used for Flash Memory.
- GPIO11: Flash Control, not usually available for general-purpose I/O.
- VCC: 3.3V power supply pin.
- GND: Ground pin.
- RST: Reset pin, used to reset the chip.

- CH_PD (EN): Chip enable pin, needs to be pulled high to enable the chip.
- ADC (A0): Analog-to-Digital Converter input pin (max 1V).

APPLICATIONS OF THE ESP8266

The ESP8266 has revolutionized IoT applications due to its flexibility, low cost, and Wi-Fi connectivity. Below are some of the most common applications of the ESP8266:

Home Automation

The ESP8266 is ideal for home automation systems, where it can be used to control appliances like lights, fans, thermostats, and even security systems. With Wi-Fi capabilities, the ESP8266 allows users to control their devices remotely through smartphones or web-based interfaces. Integration with popular platforms like Google Home or Amazon Alexa is also possible, allowing voice control of the devices.

Smart Sensors and Monitoring Systems

One of the most common uses of the ESP8266 is in sensor-based systems. It can interface with various sensors such as temperature and humidity sensors, gas detectors, motion sensors, and light sensors. Data from these sensors can be transmitted via Wi-Fi to a central server, cloud platform, or mobile app, allowing real-time monitoring and analysis.

Weather Stations

ESP8266 is widely used in creating wireless weather stations, where it can collect data from sensors like temperature, humidity, barometric pressure, and rainfall. This data can be uploaded to a cloud service like Thing Speak, allowing users to visualize the data online or via apps.

Industrial IoT (IIoT)

The ESP8266 is also employed in industrial applications, where it can be used to monitor equipment, detect faults, and control machinery remotely. This allows businesses to streamline operations, perform predictive maintenance, and enhance operational efficiency.

Wearables and Health Devices

The chip can be used in wearable devices to collect health metrics like heart rate, temperature, and activity levels. These devices can transmit data to smartphones or cloud servers, providing real-time insights into the user's health.

Security Systems

Security systems such as smart locks, surveillance cameras, and motion sensors frequently utilize the ESP8266 for remote control and monitoring. The Wi-Fi connectivity allows the user to access the system through mobile apps or web interfaces, providing security solutions for homes and businesses.

5.3 LIQUID CRYSTAL DISPLAY

A liquid crystal display (LCD) is a flat panel display, electronic visual display, or video display that uses the light modulating properties of liquid crystals. Liquid crystals do not emit light directly. LCDs are available to display arbitrary images (as in a general-purpose computer display) or fixed images which can be displayed or hidden, such as preset words, digits, and 7-segment displays as in a digital clock. They use the same basic technology, except that arbitrary images are made up of a large number of small pixels, while other displays have larger elements. An LCD is a small low cost display. It is easy to interface with a micro-controller because of an embedded controller (the black blob on the back of the board). This controller is standard across many displays (HD 44780) which means many micro-controllers (including the

Arduino) have libraries that make displaying messages as easy as a single line of code.

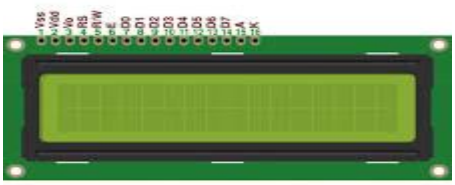

Fig 5.3 LCD display unit

LCDs are used in a wide range of applications including computer monitors, televisions, instrument panels, aircraft cockpit displays, and signage. They are common in consumer devices such as video players, gaming devices, clocks, watches, calculators, and telephones, and have replaced cathode ray tube (CRT) displays in most applications. They are available in a wider range of screen sizes than CRT and plasma displays, and since they do not use phosphors, they do not suffer image burn-in. LCDs are, however, susceptible to image persistence.

16X2 LCD SPECIFICATIONS

- Display Format: 16 characters per line, 2 lines total.
- Character Size: 5x8 pixels for standard characters.
- Dimensions: Approximately 80mm x 36mm x 13mm.
- Interface: Parallel (4-bit or 8-bit mode).
- Supply Voltage: Typically 5V DC.
- Current Consumption: Around 1.5 mA at 5V.
- Backlight: LED backlight (3.3V to 5V).
- Temperature Range: 0°C to 70°C operating, -20°C to 80°C storage.
- Response Time: Under 10 ms.
- Mounting: PCB or breadboard compatible.
- Character Set: Standard ASCII with custom character support.4

5.4 VOLTAGE SENSOR

A voltage sensor is a device used to measure and monitor the electrical potential difference (voltage) in a circuit. These sensors are crucial in various applications, including power systems, industrial automation, and electronic devices, as they help ensure the proper functioning and safety of electrical systems.

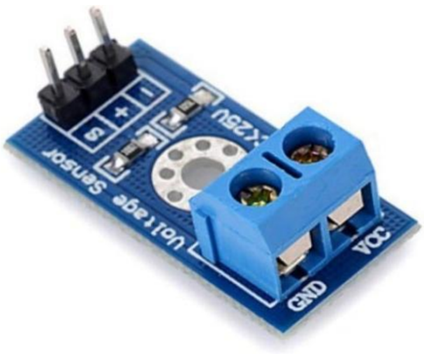

Fig 5.4 Voltage Sensor

TYPES OF VOLTAGE SENSORS

- **Analog Voltage Sensors:** Provide a continuous output signal proportional to the measured voltage. They are commonly used in simple applications.
- **Digital Voltage Sensors:** Convert the measured voltage into a digital signal for precise measurement and easy interfacing with digital systems.
- **AC Voltage Sensors:** Specifically designed to measure alternating current (AC) voltage.
- **DC Voltage Sensors:** Specifically designed to measure direct current (DC) voltage.

WORKING PRINCIPLE

Voltage sensors work based on various principles, including resistive dividers, capacitive dividers, and inductive coupling. The most common method involves a resistive divider, where the sensor reduces the voltage to a measurable level, which can then be processed by a microcontroller or other monitoring device.

VOLTAGE SENSOR PIN CONFIGURATION

- **VCC (Power Supply Pin):** Connect this pin to the positive voltage supply. For most sensors, this is typically 5V or 3.3V, depending on your sensor's requirements.
- **GND (Ground Pin):** Connect this pin to the ground (GND) of your microcontroller or development board.
- **VOUT (Output Voltage Pin):** This pin provides the output voltage corresponding to the measured input voltage. Connect this pin to an analog input pin on your microcontroller to read the voltage value.

APPLICATIONS

- **Power Monitoring:** To monitor voltage levels in power systems and ensure stable power supply.
- **Battery Management:** To measure and monitor battery voltage in electric vehicles and portable electronics.
- **Industrial Automation:** To monitor voltage levels in machinery and control systems.
- **Renewable Energy Systems:** To monitor and control voltage levels in solar and wind power systems.

5.5 DHT11 Sensor

The DHT11 is a basic, low-cost digital sensor used for measuring temperature and humidity. It is commonly used in various applications due to its simplicity, ease of use, and affordability.

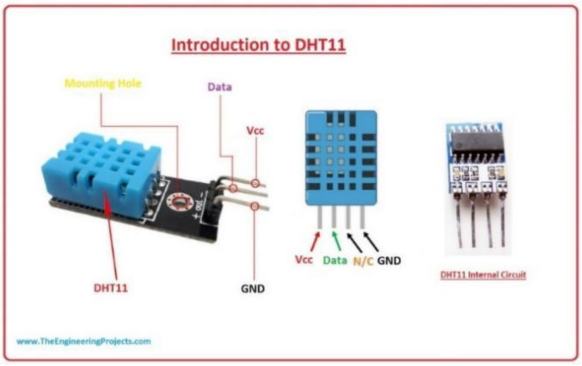

Fig 5.5 DHT11 Sensor

Features

- **Output:** Provides a calibrated digital output that is easy to interface with microcontrollers and other digital systems.
- **Cost and Availability:** Affordable and widely available, making it a popular choice for hobbyists and educational projects.
- **Interface:** Communicates using a single-wire serial interface, simplifying connections and wiring.

Pin Configuration

VCC	+3.3V to +5V
GND	Ground of the power supply
DATA	Send temperature and humidity data to a microcontroller
Range(T):	0 to 50°C
Accuracy	±2°C
Range(H):	20% to 90% RH
Accuracy:	±5% RH

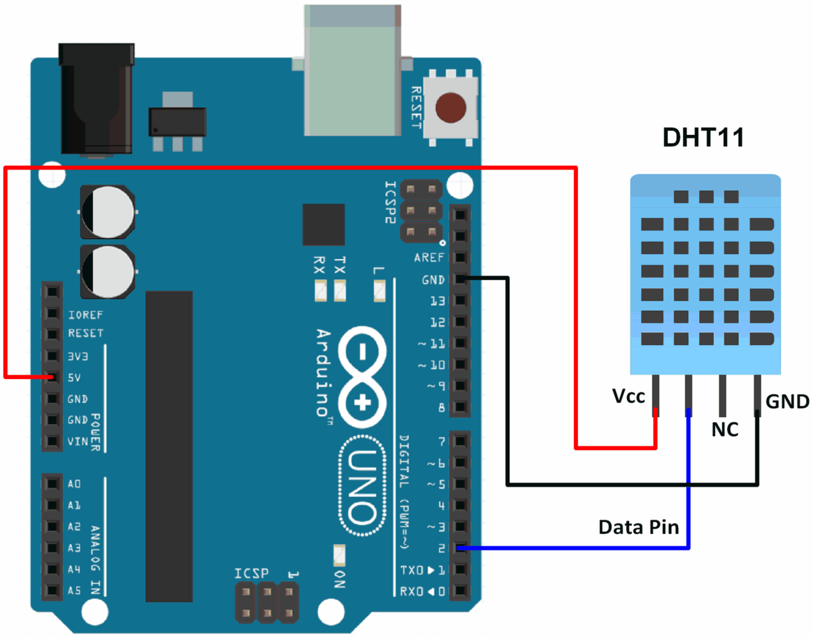

Fig 5.6 DHT-11 Digital Temperature And Humidity Sensor

APPLICATIONS

The DHT11 sensor is suitable for various applications, including:

- **Weather Stations:** Monitoring environmental temperature and humidity.
- **HVAC Systems:** Climate control in heating, ventilation, and air conditioning systems.
- **Greenhouses:** Maintaining optimal environmental conditions for plant growth.
- **Home Automation:** Smart home systems for monitoring and controlling indoor climate.
- **Industrial Monitoring:** Ensuring appropriate environmental conditions in industrial settings.

5.6 RELAY

A relay is an electromechanical device used to control high-power electrical devices with low-power signals, such as those from a microcontroller like Arduino. It works by using a small current to control a larger current, enabling you to switch circuits on and off safely.

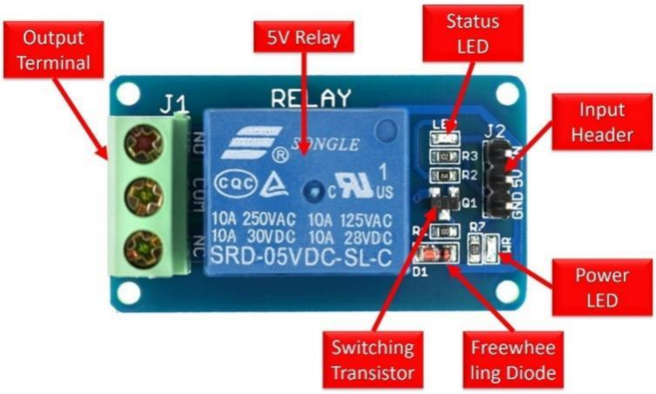

Fig 5.7 Relay

5.7 WIRING

Here's a general wiring setup using a relay module with an Arduino:

- **VCC and GND:** Connect the VCC and GND pins of the relay module to the 5V and GND pins on the Arduino, respectively.
- **Control Signal:** Connect the control pin (often labeled as IN or SIG) of the relay module to a digital output pin on the Arduino (e.g., pin 7).

5.8 LOAD AND RELAY CONTACTS

- Connect one terminal of your high-power device (load) to the power supply (usually mains voltage).
- Connect the other terminal of the load to the Common (COM) terminal of the relay.

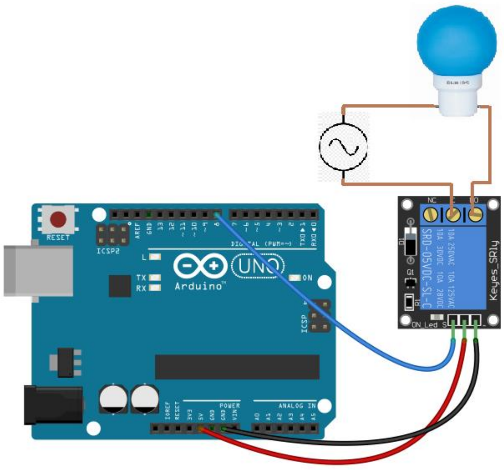

Fig 5.8 Load And Relay

- Connect the Normally Open (NO) terminal of the relay to the other side of the power supply.
- If needed, connect the Normally Closed (NC) terminal to the other side of the load for alternative switching configurations.

5.9 POWER SUPPLY

The 7812 and 7805 voltage regulators are commonly used components to provide stable DC voltage outputs of +12V and +5V, respectively, from a higher input voltage source.

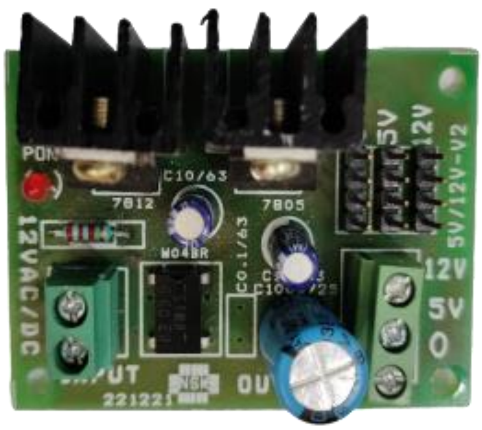

Fig 5.9 Power Supply

7812 VOLTAGE REGULATOR

Input Voltage: Typically requires an input voltage slightly higher than 12V (usually around 14-16V) to regulate effectively.

Output Voltage: Provides a stable +12V DC output.

Capacitors:

- **C1000/25:** This likely refers to a capacitor with a capacitance of 1000µF and a voltage rating of 25V. This capacitor is typically placed on the input side (between input and ground) to stabilize the input voltage, reducing noise and providing a reservoir of charge to handle transient spikes.
- **C10/63:** This could refer to a capacitor with a capacitance of 10µF and a voltage rating of 63V. This capacitor is usually placed on the output side (between output and ground) to stabilize the output voltage, filtering out any remaining noise and improving regulation.

Resistor: A resistor isn't typically used directly with the 7812 regulator in the same way as capacitors are, but it can be part of the circuit design for specific applications, such as in voltage dividers or as part of a feedback loop for stability.

7805 VOLTAGE REGULATOR

Input Voltage: Requires an input voltage typically around 7-25V (ideal 7-20V) to regulate effectively.

Output Voltage: Provides a stable +5V DC output.

Capacitors:

- **C1000/25:** As with the 7812, this capacitor stabilizes the input voltage to the regulator.

- **C10/63:** This capacitor stabilizes the output voltage of the regulator.

Resistor: Similar to the 7812, resistors are not directly part of the typical configuration but can be used in specific applications.

CIRCUIT CONSIDERATIONS

Decoupling Capacitors: These capacitors (like C1000/25 and C10/63) are crucial for filtering out noise and stabilizing the voltage levels, ensuring reliable operation of your circuit.

Heat Dissipation: Both regulators can generate heat, especially when dropping significant voltage. Adequate heat sinking may be necessary depending on the current drawn and the input-output voltage differential.

Current Requirements: Ensure that the regulators can supply enough current for your application. If higher currents are required, additional heat sinking and possibly parallel regulators may be needed.

In summary, the 7812 and 7805 voltage regulators, along with capacitors like C1000/25 and C10/63, form a basic yet effective setup for providing stable +12V and +5V outputs in electronic circuits, suitable for a wide range of applications from powering microcontrollers to analog circuits.

5.10 12V 5Ah BATTERY

BATTERY CHARGER

The rechargeable backup battery provides power to Finger Tec terminals when the primary source of power is unavailable. With the right backup battery, your system won't have to be interrupted during a power failure. 12V1.5Ah Backup Battery Access Control System: The external Rechargeable Backup Batteries are almost always used in an access control system. The backup battery prevents intruders from disabling the access control by turning off power to the building, and continues locking the doors secured by the system.

Time & Attendance System: For Time and Attendance System that records clocking-in and out data for employees, power failure might cause discrepancies in the payroll system. Thus, external rechargeable backup batteries are often used in Time & Attendance terminals as a backup power

A battery charger is a device used to put energy into a cell or (rechargeable) battery by forcing an electric current through it. Lead-acid battery chargers typically have two tasks to accomplish. The first is to restore capacity, often as quickly as practical. The second is to maintain capacity by compensating for self-discharge. In both instances optimum operation requires accurate sensing of battery voltage. When a typical lead-acid cell is charged, lead sulphate is converted to lead on the battery’s negative plate and lead dioxide on the positive plate. Over-charge reactions begin when the majority of lead sulphate has been converted, typically resulting in the generation of hydrogen and oxygen gas. At moderate charge rates, most of the hydrogen and oxygen will recombine in sealed batteries. In unsealed batteries however, dehydration will occur.

Size this is pretty straight forward, how big are the batteries? Lead acid batteries don't get much smaller than C-cell batteries. Coin cells don't get much larger than a quarter. There are also standard sizes, such as AA and 9V which may be desirable. Weight and power density This is a performance issue: higher quality (and more expensive) batteries will have a higher power density. If weight is an important part of your project, you will want to go with a lighter, high-density battery. Often this is expressed in Watts-hours per Kilogram. Price Price is pretty much proportional to power-density (you pay more for higher density) and proportional to power capacity (you pay more for more capacity). The more power you want in a smaller, lighter package the more you will have to pay. Voltage The voltage of a battery cell is determined by the chemistry used inside. For example, all Alkaline cells are 1.5V, all lead-acid's are 2V, and lithiums are 3V. Batteries can be made of multiple cells, so for example, you'll rarely see a 2V lead-acid battery. Usually they are connected together inside to make a 6V, 12V or 24V battery. Likewise, most electronics use multiple alkalines to generate the voltage they need to run. Don't forget that voltage is a 'nominal' measurement, a "1.5V" AA battery actually starts out at 1.6V and then quickly drops down to 1.5 and then slowly drifts down to 1.0V at which point the battery is considered 'dead'. Re-usability Some batteries are rechargable, usually they can be recharged 100's of times

POWER CAPACITY AND POWER CAPABILITY

Power capacity is how much energy is stored in the battery. This power is often expressed in Watt-hours (the symbol Wh). A Watt-hour is the voltage (V) that the battery provides multiplied by how much current (Amps) the battery can provide for some amount of time (generally in hours). Voltage * Amps * hours = Wh. Since voltage is pretty much fixed for a battery type due to its internal chemistry (alkaline, lithium, lead acid, etc), often only the Amps*hour measurement is printed on the side, expressed in Ah or mAh (1000mAh = 1Ah). To get Wh, multiply the Ah by the nominal voltage. For example, lets say we have a 3V nominal battery with 1Amp-hour capacity, therefore it has 3 Wh of capacity. 1 Ah means that in theory we can draw 1 Amp of current for one hour, or 0.1A for 10 hours, or 0.01A (also known as 10 mA) for 100 hours. However, the amount of current we can really draw (the power capability) from a battery is often limited. For example, a coin cell that is rated for 1 Ah can't actually provide 1 Amp of current for an hour, in fact it cant even provide 0.1 Amp without overextending itself. Its like saying a human has the capability to travel up to 30 miles: of course running 30 miles is a lot different than walking! Likewise, a 1Ah coin cell has no problem providing a 1mA for 1000 hours but if you try to draw 100mA from it, it'll last a lot less than 10 hours.

OPERATING INSTRUCTIONS

Once the connection instructions have been followed, plug-in AC power cord, the “POWER” Red (LED) will be on, the charger will begin charging automatically and the “CHARGING” Yellow (LED) will be on during charging. When the battery is fully charged the “CHARGING” Yellow (LED) will be off and the “FULL/FLOAT” Green (LED) will be on. Float Mode allows the charger to effectively be left connected to your batteries, over the course of a season, without overcharging your batteries and maintains your battery's full charge.

SPECIFICATIONS

Input Voltage: 120Vac 50/60Hz 0.4A Max. 9.2 Charging starting conditions: Battery not less than 5.5V 9.3 Rating output: 12Vdc 1.5A 9.4 Battery type: Lead-acid battery 9.5 Maximum charging voltage: 14.4V 9.6 Maintenance charging voltage: 13.2V~14.0V 9.7 Operating Environmental: -10~40℃, 90% RH Maximum 9.8 Weight: 0.62Lbs (0.28kg) approx. 9.9 Dimensions: L4.65” x W1.18” x H2.83” (L118 x W30 x H72mm)

Reverse Battery / Output Protect Condition: The charger has reverse battery and output short circuit protection. If a reverse battery charger condition exists (“FAULT” Red L.E.D.) solid, while output leads are connected backwards), simply unplug charger from AC power and properly remake the connections as described in this manual.

5.11 DC – DC CONVERTOR

A DC-DC converter is an essential power electronics device used to convert one direct current (DC) voltage level to another. In the context of electric vehicle (EV) charging systems, DC-DC converters play a crucial role in regulating voltage and ensuring that the correct voltage is supplied to the battery, enhancing system efficiency and protecting critical components from power fluctuations. The primary objective of a DC-DC converter is to ensure a stable and reliable voltage output, which is vital for the safe operation and longevity of the battery and other connected devices. DC-DC converters operate by taking input voltage from a power source, such as a solar panel or battery, and converting it to the desired output voltage through a series of switching and filtering processes. Depending on the application, the converter can either step up (boost) or step down (buck) the voltage level. A boost converter increases the input voltage to a higher output voltage, whereas a buck converter reduces the input voltage to a lower output voltage. Some DC-DC converters, known as buck-boost converters, can perform both functions, offering greater flexibility and adaptability in dynamic energy environments. In the proposed EV charging system, the DC-DC converter ensures that power from solar panels and battery storage is delivered at the optimal voltage required for charging the EV. As solar energy output varies depending on sunlight intensity, and battery voltage fluctuates during discharge and recharge cycles, the DC-DC converter stabilizes the voltage levels to prevent inconsistencies and damage. By maintaining consistent voltage output, the converter protects the battery from overvoltage, which can cause excessive heat, reduce battery life, or lead to catastrophic failure. The operation of a DC-DC converter involves high-frequency switching using metal-oxide-semiconductor field-effect transistors (MOSFETs) or insulated-gate bipolar transistors (IGBTs). These switches alternate between ON and OFF states, rapidly transferring energy to an inductor or capacitor, which stores and releases energy as required to achieve the desired output voltage. The switching frequency, typically ranging from a few kilohertz to several megahertz, is carefully controlled to optimize energy conversion efficiency and minimize power losses. The efficiency of a DC-DC converter is a critical factor in ensuring overall system performance. Modern converters are designed to achieve efficiencies of 90% or higher, minimizing energy losses and ensuring that maximum power is delivered to the load. High-efficiency converters reduce heat generation and improve the reliability and durability of the system, making them ideal for EV fast-charging applications where consistent power delivery is essential. In addition to voltage regulation, DC-DC converters offer protection mechanisms such as overvoltage protection, overcurrent protection, and thermal shutdown. These safeguards ensure that the converter operates within safe limits, preventing damage to connected devices and maintaining the stability of the charging system. a DC-DC converter is a vital component of an EV charging system, ensuring efficient power management and voltage regulation.

5.12 INVERTOR

An inverter plays a critical role in the proposed IoT-based intelligent renewable energy management system by converting the direct current (DC) generated by the wind turbine and stored in the battery into alternating current (AC), which is required to power household appliances, industrial equipment, and other electrical loads. The inverter ensures that the generated energy is compatible with the standard electrical grid and load requirements, enhancing the system's versatility and efficiency. The inverter operates by utilizing semiconductor devices such as Insulated Gate Bipolar Transistors (IGBTs) or Metal-Oxide-Semiconductor Field-Effect Transistors (MOSFETs) to rapidly switch the DC input on and off, creating a pulse-width modulated (PWM) signal. This PWM signal is then filtered to produce a smooth AC waveform that replicates the characteristics of the grid supply. The output frequency and voltage of the inverter are regulated to match the requirements of the connected load, ensuring stable and consistent performance. In the context of the proposed system, the inverter functions as an essential component that bridges the gap between renewable energy generation and practical energy utilization. The wind turbine generates DC power, which is either used to charge the battery or directly supplied to the load when demand is high. During periods of low wind energy generation or peak load demand, the stored DC energy from the battery is converted to AC through the inverter to meet the load requirements. This process ensures a continuous and reliable power supply, even under fluctuating environmental and load conditions. The inverter used in the proposed system incorporates advanced control algorithms that enhance its performance and efficiency. These algorithms dynamically adjust the pulse width and switching frequency to optimize power conversion and minimize energy losses. Additionally, the inverter is equipped with protection mechanisms such as overvoltage, undervoltage, and short-circuit protection, safeguarding the system against potential damage caused by abnormal operating conditions. Another notable feature of the inverter is its ability to perform grid synchronization. In grid-connected applications, the inverter ensures that the generated AC power is synchronized with the grid frequency and phase, allowing for seamless integration with the existing power infrastructure. This synchronization capability is essential for maintaining grid stability and preventing disturbances that could affect other connected systems. The inverter also supports bidirectional energy flow, enabling it to feed excess energy back to the grid when generation exceeds demand. This feature enhances the system's efficiency by allowing surplus energy to be utilized effectively rather than being wasted. During periods of low generation, the inverter draws power from the battery to maintain a stable power supply, ensuring that the load remains unaffected. To further enhance system reliability and user convenience, the inverter is integrated with the IoT interface, allowing users to monitor and control its performance remotely. Real-time data on voltage, current, frequency, and power output can be accessed through the IoT platform, providing users with valuable insights into system operations. This remote monitoring capability facilitates predictive maintenance and ensures that any potential issues with the inverter can be identified and addressed promptly. The inverter in the proposed system is a vital component that ensures efficient power conversion, grid compatibility, and reliable energy distribution. By integrating advanced control algorithms, protection mechanisms, and IoT-enabled monitoring, the inverter enhances the overall performance and resilience of the system, making it a robust and scalable solution for sustainable energy management.

5.13 DC MOTOR

A DC motor is a mechanically commutated electric motor powered from direct current (DC).The stator is stationary in space by definition and therefore the current in the rotor is switched by the commutator to also be stationary in space. This is how the relative angle between the stator and rotor magnetic flux is maintained near 90 degrees, which generates the maximum torque.DC motors have a rotating armature winding (winding in which a voltage is induced) but non-rotating armature magnetic field and a static field winding (winding that produce the main magnetic flux) or permanent magnet. Different connections of the field and armature winding provide different inherent speed/torque regulation characteristics. The speed of a DC motor can be controlled by changing the voltage applied to the armature or by changing the field current. The introduction of variable resistance in the armature circuit or field circuit allowed speed control. Modern DC motors are often controlled by power electronics systems called DC drives.

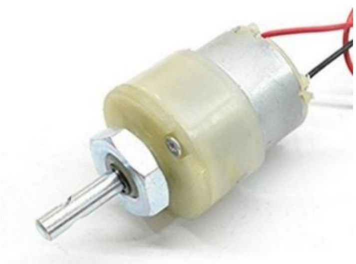

Fig 5.10 DC Motor

An Electric DC motor is a machine which converts electric energy into mechanical energy. The working of DC motor is based on the principle that when a current-carrying conductor is placed in a magnetic field, it experiences a mechanical force. The direction of mechanical force is given by **Fleming's Left-hand Rule** and its magnitude is given by

$$F = BIL \text{ Newton.}$$

CHAPTER 6

RESULT AND DISCUSSION

In this project, an intelligent and IoT-enabled wind energy management system was successfully developed and implemented to overcome the limitations of traditional standalone wind energy setups. The system incorporated key components such as a wind turbine, battery storage, inverter, voltage and temperature sensors, relays, and an Arduino Uno microcontroller, all coordinated to ensure optimal energy generation, monitoring, and distribution. One of the core outcomes was the system's ability to adapt dynamically to varying wind and environmental conditions using sensor feedback, thereby optimizing energy production and minimizing mechanical strain. Voltage sensors were placed on both the turbine and the battery to prevent overcharging and deep discharging, which effectively enhanced battery longevity and overall system efficiency.

A temperature sensor monitored ambient conditions and informed decisions about turbine operation, allowing the system to regulate itself even in fluctuating weather. The generated DC power was efficiently stored and later converted into AC via an inverter, enabling compatibility with standard domestic and industrial electrical appliances. The relay-based control mechanism automated the turbine and load operation based on real-time parameters, reducing manual intervention, preventing energy loss during low wind periods, and ensuring user safety by disconnecting loads during unsafe conditions.

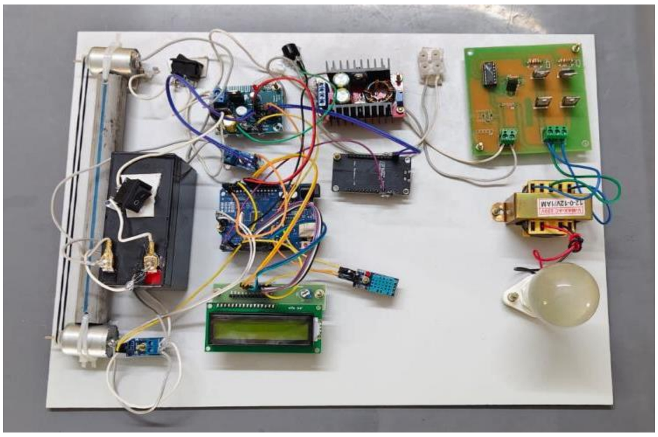

Fig 6.1 Iot Kit Image (Off)

Additionally, the integration of the ESP8266 Wi-Fi module added a powerful layer of remote monitoring and control. Real-time data such as wind energy output, battery voltage, and temperature were transmitted to cloud platforms like Blynk and ThingSpeak. This allowed users to visualize data through a mobile dashboard and receive alerts regarding system performance or fault conditions. Historical data storage on these platforms enabled trend analysis, predictive maintenance, and performance optimization over time.

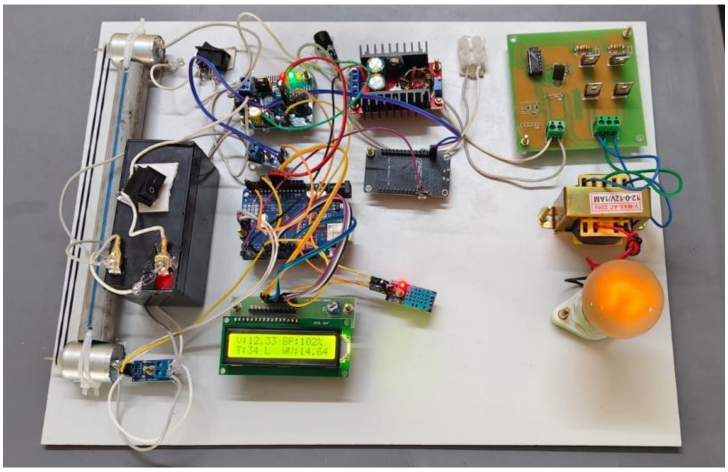

Fig 6.2 Iot Kit Image (On)

The system was first tested and verified through simulations in the Proteus Design Suite, which allowed for debugging and validation of circuit functionality before physical implementation. Programming was carried out using Arduino IDE, where custom logic was developed to handle sensor inputs, system thresholds, and communication protocols. The final hardware implementation aligned well with simulation results, confirming the accuracy of the control logic and sensor readings.

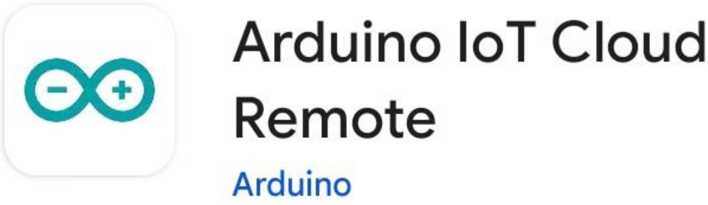

Fig 6.3 Iot Android Application

The system's modularity allows it to be easily scaled or customized based on specific energy demands or geographic conditions, making it highly adaptable for rural electrification, off-grid homes, agricultural fields, and disaster-prone areas. One of the standout features is its seamless user experience: the real-time LCD display provided on-site monitoring, while the IoT interface enabled full control from anywhere in the world. Furthermore, automated alerts and control through the app reduced the need for continuous physical supervision, making the system highly reliable and low-maintenance. Overall, the project achieved its objectives of smart energy generation, real-time feedback, automation, and user-friendly monitoring. It offers a cost-effective, energy-efficient, and sustainable solution for modern renewable energy challenges, serving as a practical model for future implementations in smart grids and remote electrification.

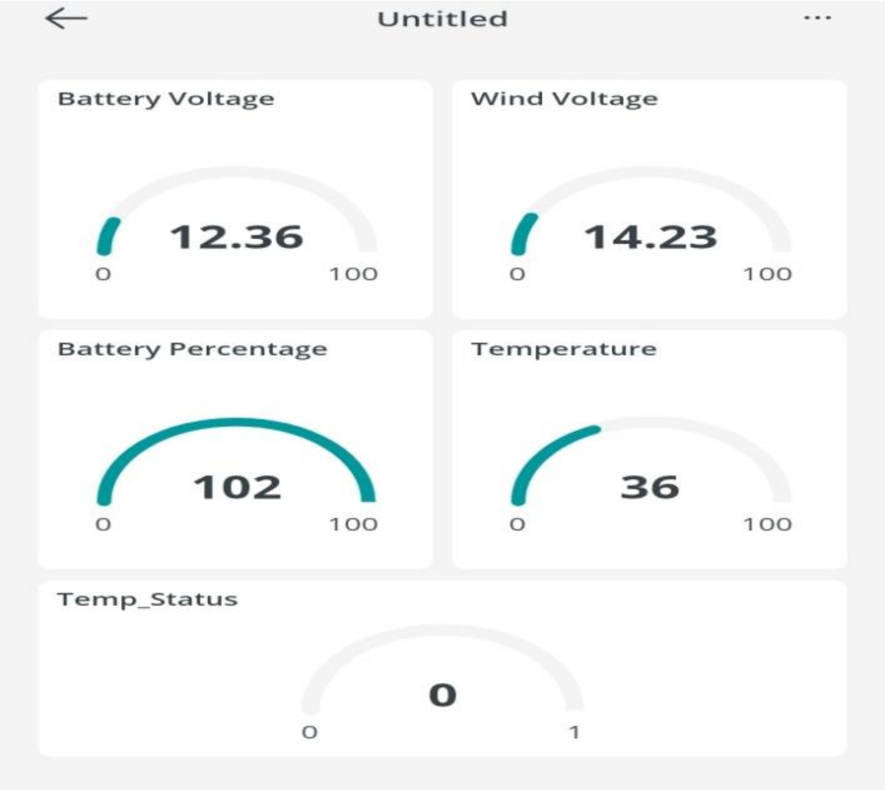

Fig 6.4 Iot Application Monitoring Interface

CHAPTER 7

7.1 CONCLUSION

This project successfully demonstrates the design and implementation of an intelligent wind energy generation and management system using Arduino and IoT technologies. By integrating voltage sensors, a temperature sensor, an inverter, a DC-DC converter, and IoT modules, the system effectively monitors, regulates, and controls energy flow from wind generation to battery storage and AC load delivery. The use of real-time sensor data ensures that the system operates efficiently and adapts to changing environmental conditions, such as fluctuating wind speeds, to maintain stable power output.

The inclusion of an IoT module enhances the system's functionality by enabling users to remotely monitor and control key parameters such as wind voltage, battery status, and load operation. This remote accessibility not only improves user convenience but also ensures better energy management in off-grid or hard-to-reach locations. The LCD interface offers local real-time feedback, while the IoT dashboard allows users to make smart, data-driven decisions from anywhere.

Overall, the hardware implementation validated the system's reliability, accuracy, and energy efficiency. With the successful demonstration of remote control, automated relay switching, and stable power conversion, this project serves as a scalable and eco-friendly solution for smart energy applications. It paves the way for sustainable energy management systems that combine renewable energy and smart technology for future-ready power solutions.

Note : Through this project work, PO1 to PO12 is attained and also PSO1, PSO2 and PSO3 is attained.

7.2 FUTURE ENHANCEMENT

The intelligent wind energy management system developed in this project has significant potential for future upgrades and scalability. One of the key enhancements would be the integration of machine learning algorithms to analyze historical wind patterns and predict optimal operating conditions. This would allow the system to anticipate energy generation potential and adapt its operation proactively for better efficiency.

Additionally, the system can be expanded to include hybrid energy sources, such as solar panels, creating a more robust and reliable renewable energy solution. By combining solar and wind energy, the system can ensure uninterrupted power supply even when one source is unavailable due to weather changes. Furthermore, the battery management system can be improved by implementing smart battery charging algorithms that increase battery life and efficiency.

On the IoT side, the platform can be enhanced with real-time data analytics dashboards and mobile app integration to provide better accessibility and more detailed insights for users. Cloud-based data storage and analytics could enable performance tracking, fault detection, and automated maintenance alerts.

Another promising enhancement involves grid integration, where excess energy generated can be fed back into the grid. This would not only support local energy demands but also contribute to sustainable energy development. Finally, the implementation of security features, such as encryption and authentication for IoT communication, would ensure that the system is protected against unauthorized access and data breaches.

These future enhancements aim to transform the system into a smart, scalable, and sustainable energy solution suitable for residential, commercial, and industrial applications.

SOURCE CODE

AURDINO CODE

```
#include <LiquidCrystal.h>
LiquidCrystal lcd(13, 12, 11, 10, 9, 8);
#include <Adafruit_Sensor.h>
#include "DHT.h"
#define DHTPIN 7   // what pin we're connected to
#define DHTTYPE DHT11   // DHT 11
int t;
DHT dht(DHTPIN, DHTTYPE);
/////////////////////////////voltage1///////////////
float correctionfactor1 = 0;
int analogInput1 = A5; //voltage
float vout1 = 0.0;
float vin1 = 0.0;
// two resistors 30K and 7.5k ohm
float R1_1 = 30000;  //
float R2_1 = 7500; //
float value1 = 0.0;
///////////////////////////////////////////////////////
///////////////////////battery voltage/////////////////
float correctionfactor2 = 0;
int analogInput2 = A1;
float vout2 = 0.0;
float vin2 = 0.0;
// two resistors 30K and 7.5k ohm
float R1_2 = 30000;  //
```

```
float R2_2 = 7500; //
int value2 = 0;
int x,y,z;
int R;
int Z2,Z1;
int vin1_1;
int s,f;
int F;
void setup()
{
dht.begin();
    /***********/
     Serial.begin(9600);
     lcd.begin (16,2);
     /***********/
     lcd.setCursor(0,0);
     lcd.print("Wind ");
     lcd.setCursor(0,1);
     lcd.print(" ");
     delay(3000);
     lcd.clear();
     /****************/
   pinMode(A1,INPUT);  //volt 1
  pinMode(A5,INPUT);  // voltage
pinMode(7,INPUT);// temp
pinMode(3,OUTPUT);//buzzer
}
/**********************/
void loop()
```

```
{
 dht11();
voltage1();
voltage2();
   Serial.println("T");
    Serial.println(t);
    delay(200);
    Serial.println("S");
   Serial.println(s);
   delay(200);
    Serial.println("a");
    Serial.println(vin1);
   delay(200);
      Serial.println("b");
   Serial.println(vin1_1);
   delay(200);
    Serial.println("w");
   Serial.println(vin2);
   delay(200);
}
void voltage1()
{
    value1 = analogRead(analogInput1);
    vout1 = (value1 * 5) / 1023.0; // see text
    vin1 = vout1 / (R2_1/(R1_1+R2_1));
    vin1 = vin1+0.7 - correctionfactor1;
    lcd.setCursor(0,0);
    lcd.print("V:");
    lcd.setCursor(2,0);
```

```
   lcd.print(vin1);
vin1_1=((vin1/12)*100);
  lcd.setCursor(8,0);
   lcd.print("BP:");
   lcd.print(vin1_1);
   lcd.print("% ");
}
void voltage2()
{
   value2 = analogRead(analogInput2);
   vout2 = (value2 * 7) / 1023.0; // see text
   vin2 = vout2 / (R2_2/(R1_2+R2_2));
   vin2 = vin2 - correctionfactor2;
   lcd.setCursor(8,1);
 lcd.print("WV:");
 lcd.setCursor(11,1);
lcd.print(vin2);
delay(2000);
}
 void dht11()
{
 int h = dht.readHumidity();
        t = dht.readTemperature();
       float f = dht.readTemperature(true);
       if (isnan(h) || isnan(t) || isnan(f))
       {
         //Serial.println("Failed to read from DHT sensor!");
         return;
       }
```

```
lcd.setCursor(0,1);
lcd.print("T:");        lcd.setCursor(2,1);
lcd.print(t);
   if(t > 36)   /////temp
  {
lcd.setCursor(5,1);
lcd.print("H");
s=1;
digitalWrite (3,HIGH);
  }
  else
  {
        lcd.setCursor(5,1);
lcd.print("L");
s=0;
digitalWrite (3,LOW);
  }
}
```

IOT APP CODE

```
//Your username is windsystem
//projectiot2025
// Code generated by Arduino IoT Cloud, DO NOT EDIT.
#include <ArduinoIoTCloud.h>
#include <Arduino_ConnectionHandler.h>
const char DEVICE_LOGIN_NAME[] = "4fae01a1-176e-4ee9-b969-
e7ecc93c0bb3";
```

```
const char SSID[] = "projectiot";    // Network SSID (name)
const char PASS[] = "123456789";    // Network password (use for WPA, or use as key for WEP)
const char DEVICE_KEY[] = "6lrt3v7##0mbIysAtSysrpLHC";    // Secret device password
void onBvChange();
void onWvChange();
void onBpChange();
void onTempChange();
void onTsChange();
float bv;
float wv;
int bp;
int temp;
int ts;
unsigned long lastUpdateTime = 0;
const unsigned long updateInterval = 1000; // 1 second
void initProperties(){
  ArduinoCloud.setBoardId(DEVICE_LOGIN_NAME);
  ArduinoCloud.setSecretDeviceKey(DEVICE_KEY);
  ArduinoCloud.addProperty(bv, READWRITE, ON_CHANGE, onBvChange);
  ArduinoCloud.addProperty(wv, READWRITE, ON_CHANGE, onWvChange);
  ArduinoCloud.addProperty(bp, READWRITE, ON_CHANGE, onBpChange);
  ArduinoCloud.addProperty(temp, READWRITE, ON_CHANGE, onTempChange);
```

```
  ArduinoCloud.addProperty(ts, READWRITE, ON_CHANGE, onTsChange);
}
WiFiConnectionHandler ArduinoIoTPreferredConnection(SSID, PASS);
void setup() {
  Serial.begin(9600);
  initProperties();
  ArduinoCloud.begin(ArduinoIoTPreferredConnection);
  setDebugMessageLevel(2);  // Enable Debug Messages
  ArduinoCloud.printDebugInfo();
}
void loop() {
  unsigned long currentMillis = millis();
  // Update cloud at set intervals
  if (currentMillis - lastUpdateTime >= updateInterval) {
    ArduinoCloud.update();
    lastUpdateTime = currentMillis;
  }
  // Handle Serial Input and Update Variables
  if (Serial.available() > 0) {
    char identifier = Serial.read();
    switch (identifier) {
      case 'T': temp = Serial.parseInt();
break;
      case 'S': ts = Serial.parseINT(); break;
      case 'a': bv = Serial.parseFloat(); break;
      case 'b': bp = Serial.parseInt(); break;
      case 'w': wv = Serial.parseFloat(); break;
    }
```

```
  }
}
void onBvChange()  {Serial.println("Status changed to: " + String(bv));}
void onTempChange()   {Serial.println("Status changed to: " + String(temp));}
void onWvChange()  {Serial.println("Status changed to: " + String(wv));}
void onBpChange()  {Serial.println("Status changed to: " + String(bp));}
 void onTsChange()  {Serial.println("Status changed to: " + String(ts));}
```

REFERENCE

1. Alhamrouni, M. A. Hairullah, N. S. Omar, M. Salem, A. Jusoh and T. Sutikno, "Modelling and design of PID controller for voltage control of AC hybrid micro-grid", International Journal of Power Electronics and Drive Systems, vol. 10, no. 1, pp. 151.
2. D. Kumar and K. Chatterjee, "A review of conventional and advanced MPPT algorithms for wind energy systems", Renewable and sustainable energy reviews, vol. 55, pp. 957-970, 2016.
3. E. Abioye, M. S. Abidin, M. S. Mahmud, S. Buyamin, O. O. Mohammed, A. O. Otuoze, et al., "Model based predictive control strategy for water saving drip irrigation", Smart Agricultural Technology, vol. 4, pp. 100179, 2023.
4. F. Garcia-Torres, A. Zafra-Cabeza, C. Silva, S. Grieu, T. Darure and A. Estanqueiro, "Model predictive control for microgrid functionalities: Review and fuure challenges", Energies, vol. 14, no. 5, pp. 1296, 2021.
5. J. Hu, Y. Li and J. Zhu, "Multi-objective model predictive control of doubly-fed induction generators for wind energy conversion", IET Generation Transmission & Distribution, vol. 13, no. 1, pp. 21-29, 2019.
6. M. J. Ali, A. Mondal and P. Dutta, "Intelligent monitoring and control of wind turbine prototype using Internet of Things (IoT)", IEEE International IOT Electronics and Mechatronics Conference, pp. 1-6, 2022.
7. M. Paul and D.V. Kumar, "Model Based Predictive Control Strategy For Grid-Connected Wind Energy System", International Journal of Advances in Signal and Image Sciences, vol. 8, no. 1, pp. 1-8, 2022.

8. N. T. Mbungu, R. Naidoo, R. C. Bansal and M. Bipath, "Optimisation of grid connected hybrid photovoltaic–wind–battety system using model predictive control design", IET Renewable Power Generation, vol. 11, no. 14, pp. 1760-1768, 2017.
9. S. Ghosh, "Neuro-Fuzzy-Based IoT assisted power monitoring system for smart grid", IEEE Access, vol. 9, pp. 168587-168599, 2021.
10. S. K. Sarkar, F. R. Badal and S. K. Das, "A comparative study of high performance robust PID controller for grid voltage control of islanded microgrid", International Journal of Dynamics and Control, vol. 6, pp. 1207-1217, 2018.
11. S. Karad and R. Thakur, "Efficient monitoring and control of wind energy conversion systems using Internet of things (IoT): a comprehensive review", Environment development and sustainability, vol. 23, pp. 14197-14214, 2021.
12. S. L. Ullo and G. R. Sinha, Advances in smart environment monitoring systems using IoT and sensors Sensors, vol. 20, no. 11, pp. 3113, 2020.
13. W. H. Lio, J. A. Rossiter and B. L. Jones, "A review on applications of model predictive control to wind turbines", International conference on control (control), pp. 673-678, 2014.
14. W. S. Kim, W. S. Lee and Y. J. Kim, "A review of the applications of the internet of things (IoT) for agricultural automation", Journal of BiosystemsEngineering, vol. 45, pp. 385-400, 2020.
15. Y. Xie, L. Liu, Q. Wu and Q. Zhou, "Robust model predictive control based voltage regulation method for a distribution system with renewable energy sources and energy storage systems", International Journal of Electrical Power & Energy Systems, vol. 118, pp. 105749, 2020.

BIBLIOGRAPHY

I have published Indian Patent Entitled " **REAL-TIME DATA-DRIVEN INTELLIGENT IOT-BASED WIND ENERGY MANAGEMENT SYSTEM".**

PATENT PUBLISHED PROOF

(12) PATENT APPLICATION PUBLICATION	(21) Application No.202541010328 A
(19) INDIA	
(22) Date of filing of Application :07/02/2025	(43) Publication Date : 21/02/2025

(54) Title of the invention : IMPLEMENTATION & DESIGN OF E-AUTHENTICATION SYSTEM FOR VEHICLE OPERATION

(51) International classification	:H02J0003380000, G06Q0050060000, H02J0013000000, F03D0007040000, H02J0003000000
(86) International Application No Filing Date	:NA :NA
(87) International Publication No	: NA
(61) Patent of Addition to Application Number Filing Date	:NA :NA
(62) Divisional to Application Number Filing Date	:NA :NA

(71)**Name of Applicant :**
1)Dr Karthi S
Address of Applicant :V.S.B. Engineering College, Karur ----------- -----------
2)G.YASIKA
3)PRANAV G
4)JAYARAJ V
5)JANARTHANAN S
Name of Applicant : NA
Address of Applicant : NA
(72)**Name of Inventor :**
1)G.YASIKA
Address of Applicant :V.S.B. ENGINEERING COLLEGE, KARUR KARUR ----------- -----------
2)PRANAV G
Address of Applicant :V.S.B. ENGINEERING COLLEGE, KARUR KARUR ----------- -----------
3)JAYARAJ V
Address of Applicant :V.S.B. ENGINEERING COLLEGE, KARUR KARUR ----------- -----------
4)JANARTHANAN S
Address of Applicant :V.S.B. ENGINEERING COLLEGE, KARUR KARUR ----------- -----------

(57) Abstract :
This project aims to develop an intelligent renewable energy system that utilizes wind energy for power generation and a battery for energy storage. The system is designed to monitor and control key parameters such as wind energy, battery voltage, and temperature, ensuring optimal energy management. Voltage sensors track the voltage levels of both the wind generator and the battery, while a temperature sensor monitors wind conditions for efficient energy production. The system includes a power regulation mechanism to ensure stable energy output and an inverter to convert DC to AC for powering connected loads. A relay is used to control the operation of both the wind turbine and the load. The system also integrates an interface that provides real-time data on wind energy, battery status, and system operations, ensuring that users can easily monitor and manage the system. With remote monitoring and control capabilities, users can turn the wind turbine and load on or off based on the data, optimizing energy efficiency.

No. of Pages : 6 No. of Claims : 7

The Patent Office Journal No. 08/2025 Dated 21/02/2025 18917