International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

"PRIORITY BASED AUTOMATIC LOAD SHEDDING SYSTEM"

Prof. Shivalingaiah K L[1], Naveena[2], Poornima G N[3], Ramesh V K[4], Srikantha Badigera[5]

[1]Asst.Professor, Electrical & Electronics Engineering, RajaRajeswari College of Engineering
[2,3,4,5]UG Students, Electrical & Electronics Engineering, RajaRajeswari College of Engineering

A B S T R A C T :

This project is about building an automatic load shedding system using the ESP32 microcontroller. The system helps manage electricity when the power load is too high. It works by off less important devices to prevent overload and power failure. The ESP32 checks the automatically turning power usage in real-time and controls the devices based on priority. This system can be used in agriculture or industries to save energy, make smart power management easier. This project presents a simple priority-based automatic load-shedding system in which two resistive loads are controlled according to user-defined start and stop times. The system uses a microcontroller to automate the switching of loads based on a present schedule while maintaining a priority sequence. When the programmed time is reached, the controller activates the first load and then switches the second load only if sufficient supply is available. Similarly, if a load must be turned OFF at the stop time or if power must be reduced, the lower-priority load is disconnected before the primary load. Users can set the required start and stop times using a keypad, and the RTC module provides accurate time for timed operations. A display unit shows the current time, load status, and user settings. The relay driver and relays interface the controller with the two resistive loads, ensuring safe and reliable switching. This system helps automate energy usage, prevents unnecessary power consumption, and ensures important loads always receive priority during limited supply conditions. The design is simple, cost- effective, and ideal for small-scale timed load management application.

Keywords:

- Load shedding
- Priority-based control
- Automatic switching
- Overload protection
- Real-time power monitoring
- Energy management

Introduction

A Priority Based Automatic Load Shedding System is designed to manage electrical loads efficiently during situations of limited power availability or overload. In many households, industries, and commercial areas, the demand for electricity often exceeds the available supply, leading to voltage drops, equipment damage, or complete power failure. To avoid these problems, load shedding is used to temporarily disconnect certain electrical loads. In this system, loads are assigned priority levels based on their importance. The microcontroller (such as ESP32 or any controller used) continuously monitors the power status or time settings. When the load exceeds the predefined limit or when the programmed schedule is reached, the system automatically turns OFF lower-priority loads first while keeping essential loads active. Once normal conditions return, the system restores the disconnected loads. This approach ensures efficient power management, protects electrical devices, and maintains uninterrupted operation of critical equipment. It is especially useful in smart homes, industries, and energy-saving applications where controlled distribution of power is required.

Problem Statement

In many residential, commercial, and industrial areas, the demand for electrical power often exceeds the available supply, leading to overload conditions, voltage fluctuations, and unexpected power failures. Traditional load shedding methods are manual, inefficient, and disconnect all loads without considering their importance. This results in disruption of essential services, reduced productivity, and wastage of energy. There is a need for an automatic system that can intelligently monitor the power conditions or scheduled timings and disconnect only non-essential loads while keeping critical loads active. The system should prioritize loads based on importance and perform shedding in a controlled manner to prevent overload and ensure continuous operation of key equipment.

Literature review

Automated load-shedding systems have been widely studied as a solution to power instability, especially in distribution networks where demand frequently exceeds supply. Traditional methods such as under-frequency and under-voltage load shedding rely on relays that disconnect preset blocks of load when the system frequency or voltage drops, but research shows that these approaches often cause excessive or poorly targeted disconnections. To overcome these limitations, recent studies focus on smart, priority-based shedding using IoT technologies, particularly the ESP32 microcontroller. The ESP32 is frequently highlighted in literature because of its built-in Wi-Fi, low cost, and ability to perform real-time monitoring and control. Research prototypes typically use sensors to measure voltage and current, relay modules to switch loads, and cloud or local dashboards for remote monitoring. Many studies demonstrate that integrating ESP32 with IoT platforms improves decision-making by allowing loads to be shed based on priority levels, real-time consumption data, or demand-response signals rather than fixed thresholds. Some investigations incorporate optimization algorithms such as fuzzy logic, heuristic rules, or scheduling strategies to minimize consumer inconvenience while maintaining system stability. Although findings show that ESP32-based solutions enhance flexibility, responsiveness, and user control, the literature also notes limitations such as small-scale testing, limited cybersecurity, and a lack of field-level validation in real power systems. Overall, the review suggests that ESP32-enabled automated load shedding offers a promising, intelligent, and energy-efficient alternative to traditional protection-based methods, with potential for scalable deployment in smart grids and microgrids.

Existing System

In the existing load-shedding system used in many power distribution networks, the process is mostly manual or relay-based, where operators disconnect entire feeders or large blocks of load whenever the demand exceeds the available supply. These traditional systems use fixed under-frequency or under-voltage relays that automatically trip circuits when the grid becomes unstable. However, this method is often inefficient because it disconnects all loads together, without considering their importance or priority. As a result, even essential loads such as lighting, communication equipment, or hospital devices may get switched off unnecessarily. The existing system also lacks real-time monitoring, remote control, and consumer-level decision-making, causing delays and communication gaps between grid operators and users. Additionally, manual switching requires continuous human supervision, which increases response time and can lead to uneven power distribution. Due to these limitations, the existing system is unable to provide selective, flexible, and efficient load management—creating the need for an automated, intelligent solution like an ESP32-based load-shedding system.

Methodology

The methodology of the automated load-shedding system using an ESP32 involves a structured process of sensing, decision-making, and controlled switching. First, electrical parameters such as voltage, current, and load consumption are measured using sensors like current transformers or Hall-effect sensors connected to the ESP32. The ESP32 continuously processes this real-time data to compare the total load demand with the available power threshold set by the user or utility. Based on this comparison, the controller applies a priority-based algorithm, where each connected appliance is assigned a priority level such as critical, essential, or non-essential. When the demand exceeds the threshold, the system automatically disconnects lower-priority loads by activating relay modules, ensuring essential loads remain powered. Simultaneously, the ESP32 uses its built-in Wi-Fi to update the monitoring dashboard or mobile app, allowing users or administrators to view consumption data and the current shedding status. The system can also receive commands for remote control, scheduling, or manual override. This methodology ensures efficient energy management by combining real-time sensing, intelligent decision-making, and automated switching in a seamless and user-friendly IoT-based framework.

System Architecture

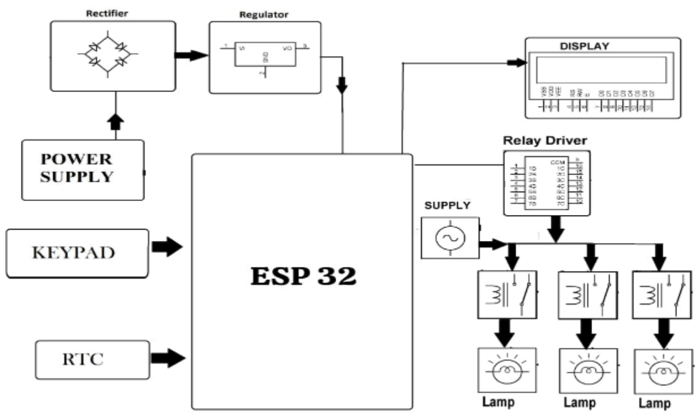

Fig: Block Diagram of Automatic Load Shedding System

The working of the automated load-shedding system using the ESP32 is based on continuous monitoring, intelligent decision-making, and automatic control of electrical loads. The system first measures real-time voltage and current using sensors connected to the ESP32, which processes this data to calculate the total load consumption. A predefined threshold limit is set, and whenever the connected load exceeds this limit, the ESP32 immediately identifies an overload condition. To prevent power failure or equipment damage, the controller uses a priority-based algorithm, where each load is assigned a priority level such as critical, essential, or non-essential. The ESP32 then disconnects only the low-priority loads through relay modules, ensuring that important appliances continue operating even during high demand. At the same time, the ESP32 updates the system status on a mobile app or dashboard through Wi-Fi, allowing users to track consumption and switching actions in real time. Once the load returns to safe levels, the system can reconnect the shed loads automatically or as per user command. This continuous sensing, evaluating, and switching process enables the system to maintain stable power distribution without human intervention, making the overall operation efficient, reliable, and fully automated.

The Proposed Priority Based Automatic Load Shedding System using the hardware and Software components Implementation

Hardware Implementation:

1. **Node MCU ESP32**

The ESP32 series employs either a variation Xtensa LX6 microprocessor in both dual-core and single-core variations, an Xtensa LX7 dual-core microprocessor, a includes built-in antenna switches, RF balun, power amplifier, low-noise receive amplifier, filters, and power management modules. a system on a chip that integrates the following features: Wi-Fi (2.4 GHz band) Bluetooth. Dual high performance Xtensa 32-bit LX6 CPU cores. Ultra Low Power co- processor. It is commonly found either on device-specific PCBs or on a range of development boards with GPIO pins and various connectors depending on the model and manufacture of the board.

2. **Keypad**

A 4×4 matrix keypad is a simple and commonly used input device in embedded systems. It consists of 16 keys arranged in four rows and four columns. Each key is connected at the intersection of a row and a column. When a skey is pressed, it creates an electrical connection between its corresponding row and column line. Microcontrollers such as Arduino or ESP32 scan these row–column combinations to detect which key is pressed. This type of keypad is widely used in security systems, password-based door locks, calculators, and menu navigation because it is inexpensive, easy to interface, and reliable for numeric and character input.

3. **RTC**

The DS3231 RTC (Real-Time Clock) module is a highly accurate timekeeping device used to maintain real-time information such as seconds, minutes, hours, date, month, and year. It includes an onboard battery that ensures the clock continues running even when external power is removed. The module communicates with microcontrollers through the I2C protocol using the SDA and SCL pins. Due to its stability, low power consumption, and built-in temperature-compensated crystal oscillator, it is widely used in data loggers, automation systems, clocks, and time-based control applications.

4. **16*2 LCD Display**

The term LCD stands for liquid crystal display. It is one kind of electronic display module used in an extensive range of applications like various circuits & devices like mobile phones, calculators, computers, TV sets, etc. These displays are mainly preferred for multi-segment light-emitting diodes and seven segments. The main benefits of using this module are inexpensive; simply programmable, animations, and there are no limitations for displaying custom characters, special and even animations.

5. **Two Channel Relay Module**

A two-channel relay module is an electromechanical switching device that allows a microcontroller such as the ESP32 or Arduino to control two independent high-voltage electrical loads. Each channel consists of a relay driver circuit and an electromechanical relay that can switch AC or DC loads safely while isolating the low-voltage control circuitry from the high-voltage output.

6. **LAMP**

A lamp is used in this project as a representative electrical load to demonstrate the functioning of the automated load-shedding system. It simulates typical household or industrial electrical appliances that are controlled during load-shedding operations.

7. **Jumper Wire**

Jumper wires are simply wires that have connector pins at each end, allowing them to be used to connect two points to each other without soldering. Jumper wires are typically used with breadboards and other prototyping tools to make it easy to change a circuit as needed. Fairly simple. In fact, it doesn't get much more basic than jumper wires. Though jumper wires come in a variety of colours, the colours don't mean anything. This means that a red jumper wire is technically the same as a black one. But the colours can be used to your advantage to differentiate between types of connections, such as ground or power.

Software Implementation:

1. **Introduction of Arduino IDE**

The ESP32 consists of both a physical programmable circuit board (often referred to as a microcontroller) and a piece of software, or IDE (Integrated Development Environment) that runs on your computer, used to write and upload computer code to the physical board.

2. **Programming language used**

First, the ESP32 compiler/IDE accepts C and C++ as-is. In fact, many of the libraries are written in C++. Much of the underlying system is not object oriented, but it could be. Thus, "The ESP32 language" is C++ or C.

3. **Getting started with Arduino IDE**

This is the Arduino IDE once it's been opened. It opens into a blank sketch where you can start programming immediately. First, we should configure the board and port settings to allow us to upload code. Connect your Arduino board to the PC via the USB cable.

4. **Arduino IDE: Board setup**

To set up the ESP32 module in Arduino IDE, install the ESP32 board package via Boards Manager, select the correct board and COM port, install necessary USB drivers if required, and verify the setup by uploading a basic Blink LED program.

5. **Arduino IDE com port setup**

To set up the COM port for ESP32 in Arduino IDE, first connect the ESP32 board to the computer using a USB cable. Open Arduino IDE, go to Tools → Board, and select ESP32 Dev Module. Then, navigate to Tools → Port and choose the correct COM port where the ESP32 is connected. If the board is not detected, install the necessary CP2102 or CH340 drivers. After selecting the correct port, the board type and COM number will appear in the bottom-right corner of the IDE. This confirms that the ESP32 is successfully recognized and ready for programming.

6. **Introduction of processing IDE**

Processing (programming language) Processing is an open-source computer programming language and integrated development environment (IDE) built for the electronic arts, new media art, and visual design communities with the purpose of teaching non-programmers the fundamentals of computer programming in a visual context.

7. Processing software

Word processing software is used to manipulate a text document, such as a resume or a report. You typically enter text by typing, and the software provides tools for copying, deleting and various types of formatting.

8. Getting started

- Open processing IDE.
- Go to the file menu and create a new page.
- Write the code.
- Run.

9. How processing IDE Communicate with Arduino IDE

The Arduino IDE and the Processing IDE will communicate with each other through serial communication. The Processing IDE has a serial library which makes it easy to communicate with the ESP32.

10. Circuit Diagram

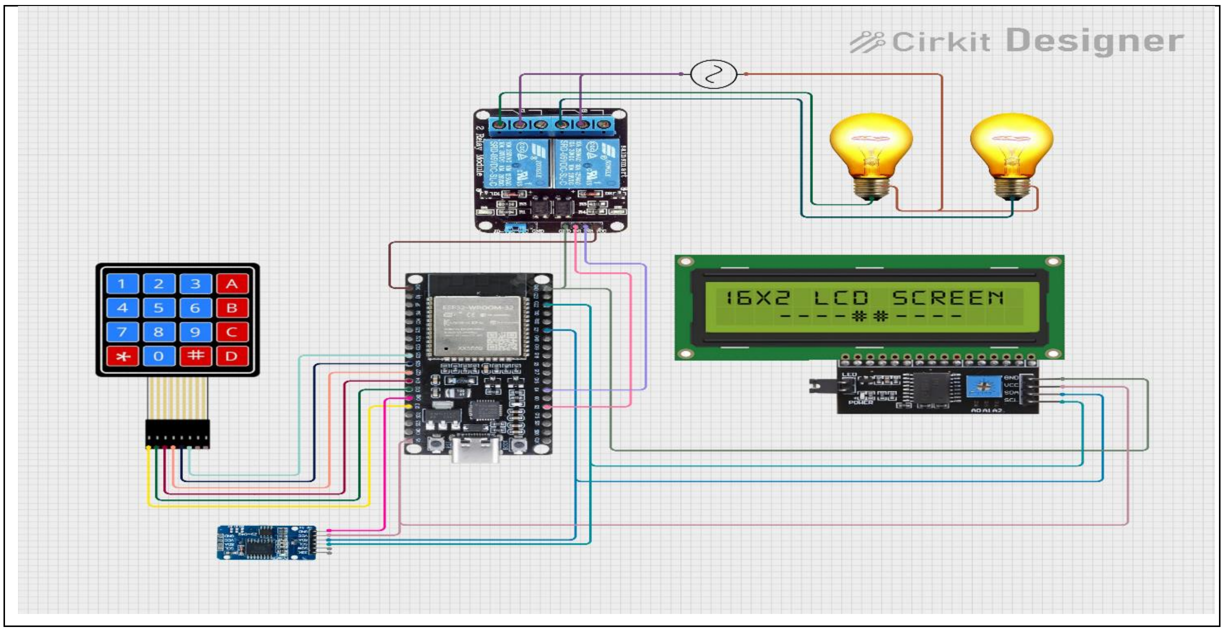

Fig: Circuit Diagram of Automatic Load Shedding System

Main Components in the Connection Diagram

ESP32 / Microcontroller: Controls the loads based on priority and timing.

Power Supply (230V AC → 5V/12V DC):

5V/12V for relay module

Main AC supply for loads

Relay Module (2-Channel / 4-Channel):

Each relay controls one load

Load 1 = High priority

Load 2 = Low priority

Resistive Loads:

For example: Bulbs, heaters, or test resistive elements

Current Sensor (optional):

Used if overload detection is required

Start/Stop Time Logic:

ESP32 switches loads ON/OFF based on user-set time

Results of Photographs

Fig:1 Load shedding system ready

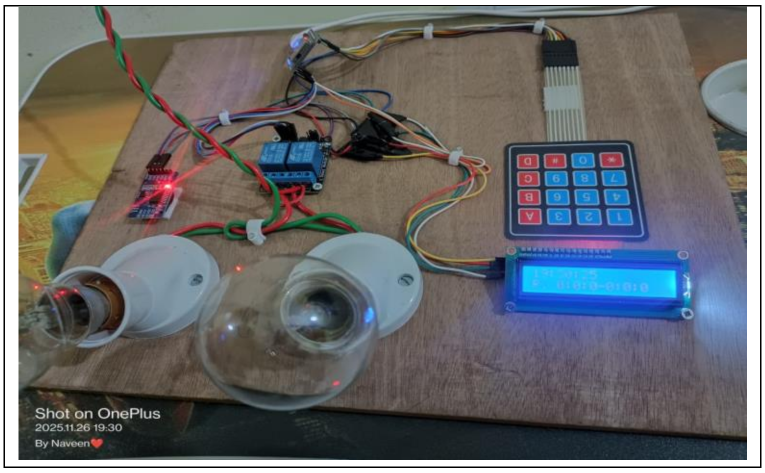

Fig:2 Load shedding System model

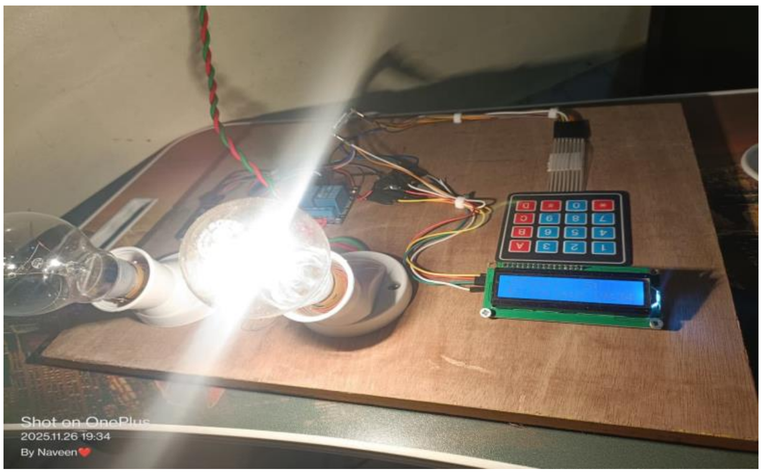

Fig:3 Load 1 Active

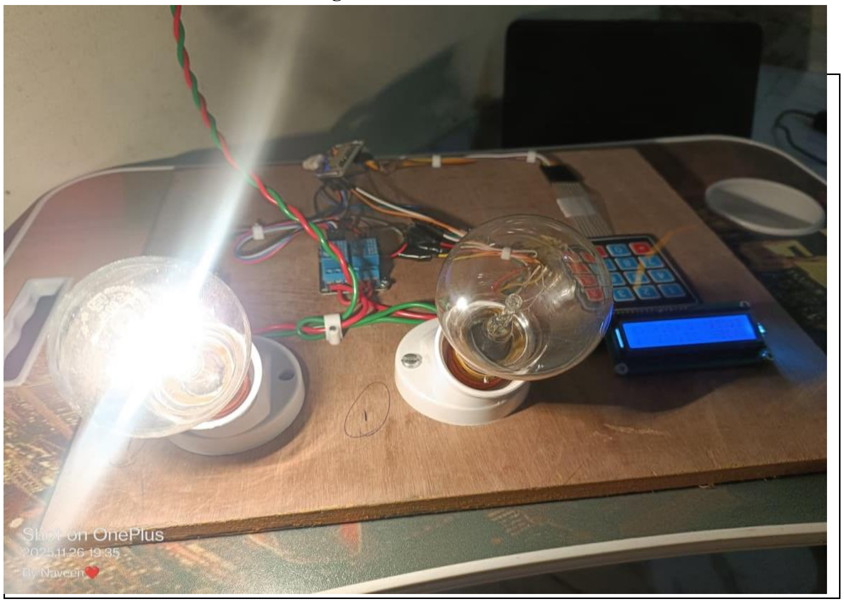

Fig:4 Load 2 Active

Discussion

The priority-based automatic load shedding system helps manage power efficiently by disconnecting lower-priority loads first during limited power conditions. In this project, two resistive loads are controlled using preset start and stop times, ensuring that essential loads continue to operate while non-essential ones are turned off automatically. This method reduces manual effort, prevents overload, and maintains system stability. Although the project uses only two loads, the same concept can be expanded for larger applications, making it a practical approach for agriculture and small industries.

Conclusion

The priority-based automatic load shedding system successfully demonstrates how power can be managed efficiently by automatically controlling loads based on their importance. By using preset start and stop times for two resistive loads, the system ensures that essential loads receive uninterrupted power while non-essential loads are turned off during limited supply or overload conditions. This reduces manual intervention, improves safety, and prevents complete power failure. Overall, the project provides a practical, reliable, and scalable solution for smart energy management in agriculture, laboratories, and small industrial setups.

Acknowledgment

We would like to express our sincere gratitude to **Prof. Shivalingaiah K L** for her continuous guidance, valuable suggestions, and encouragement throughout the project.
We also thank **Dr. Arul Jose** for his technical support and timely assistance, which greatly contributed to the successful completion of this work.
Finally, we appreciate the cooperation and contributions of all project team members who worked together to complete this research effectively.

REFERENCES

1]ESP32 Technical Reference Manual https://www.espressif.com/en/products/socs/esp32/resources – Espressif Systems

2] K. A. Raut, D. R. Kalbande, "Smart Load Shedding Using IoT", International Journal of Scientific Research in Engineering and Management (IJSREM), Vol. 4, Issue 6, 2020.

3]Arduino and ESP32 Documentation – Arduino.cchttps://www.arduino.cc

4]D. Patel, R. Sharma, "Automatic Load Management System Using IoT," International Journal of Engineering Research & Technology (IJERT), Vol. 8, Issue 12, 2019.

5]Tutorials and source code from Electronics Hub, Circuit Digest, and Instructables on ESP32 and IoT-based automation projects.

6]Priority based load shedding and automatic power factor correction—Kashid, Devkate & Kamble (2021) https://repo.ijiert.org/index.php/ijiert/article/view/1665

7] Optimization of Load Ranking and Load Shedding in a Power System Using the Improved AHPAlgorithm—Le et al. (2022) https://etasr.com/index.php/ETASR/article/view/4862

8] Arduino and ESP32 Documentation – Arduino.cchttps://www.arduino.cc

9] D. Patel, R. Sharma, "Automatic Load Management System Using IoT," International Journal of Engineering Research & Technology (IJERT), Vol. 8, Issue 12, 2019.

10] An adaptive load shedding methodology for renewable integrated power systems — Abrar, Masood&Alam(2024)https://www.sciencedirect.com/science/article/pii/S2405844024160744