Ahsanullah University of Science and Technology

PROJECT REPORT

EEE 4154
Power System II Laboratory

Submitted By:

Group 2
ID:
20210205167
20210205169
20210205170
20210205171
20210205172
Section: C2

Fall 2024

Date Submitted:

September 21, 2025

1. Introduction

In power systems, load shedding is a common practice during generation shortages or peak demand. Manual load shedding is often unreliable, time-consuming, and prone to human error.
To address this issue, this project presents an **Arduino-based Load Shedding Time Management System** that automatically manages electrical loads (represented by bulbs) based on user-defined schedules. The system is designed to be reliable, simple, and low-cost, making it practical for households and small-scale power systems.

2. Objective

- To design and implement an automated load shedding system using Arduino.
- To control multiple loads with time-based scheduling and manual toggling.
- To provide real-time feedback through an LCD display.
- To ensure fair distribution of power during outages and improve energy efficiency.

3. Components and Specifications with Price

1. **Arduino Uno R3** – main microcontroller : **1000tk**
2. **3 × Relay Modules (5V, 10A)** – switching devices for loads : **280tk**
3. **4×4 Matrix Keypad** – user interface for selection and time entry : **70tk**
4. **16×2 I2C LCD Display** – for displaying load status and countdown : **450tk**
5. **Step-down Transformer (12V AC)** – initial power source : **250 tk**

6. **Bridge Rectifier** – converts AC to DC : **50tk**

7. **Buck Converter** – regulates 14V down to stable 12V for Arduino : **100tk**

- Input Capacitor – smooths input voltage and reduces fluctuations before regulation
- Buck IC (Controller Chip) – controls switching frequency to step down voltage
- Inductor – stores and releases energy, enabling efficient DC-DC step-down
- Output Capacitor – smooths the regulated DC output to provide a stable supply

8. **Bulbs (White, Blue, Yellow)** – representing electrical loads of 0.5W, 1W, 3W: **200 tk**

9. **Supporting components** – breadboard, wires : **150tk**

TOTAL ESTIMATED COST: 2550 BDT

4. Circuit Design

The circuit is divided into four main parts: **power supply**, **control unit**, **switching unit**, and **load unit**.

1. Power Supply Section

- The AC mains (220 V) is first stepped down to 12 V AC using a transformer.

- A **bridge rectifier** converts this AC into DC, which measures around 14–15 V after filtering.

- A **buck converter** then regulates the voltage down to a stable 12 V DC to power the Arduino Uno and the relay board.

- The buck converter uses input/output capacitors, an inductor, and a switching IC to provide smooth and efficient DC output.

2. Control Unit

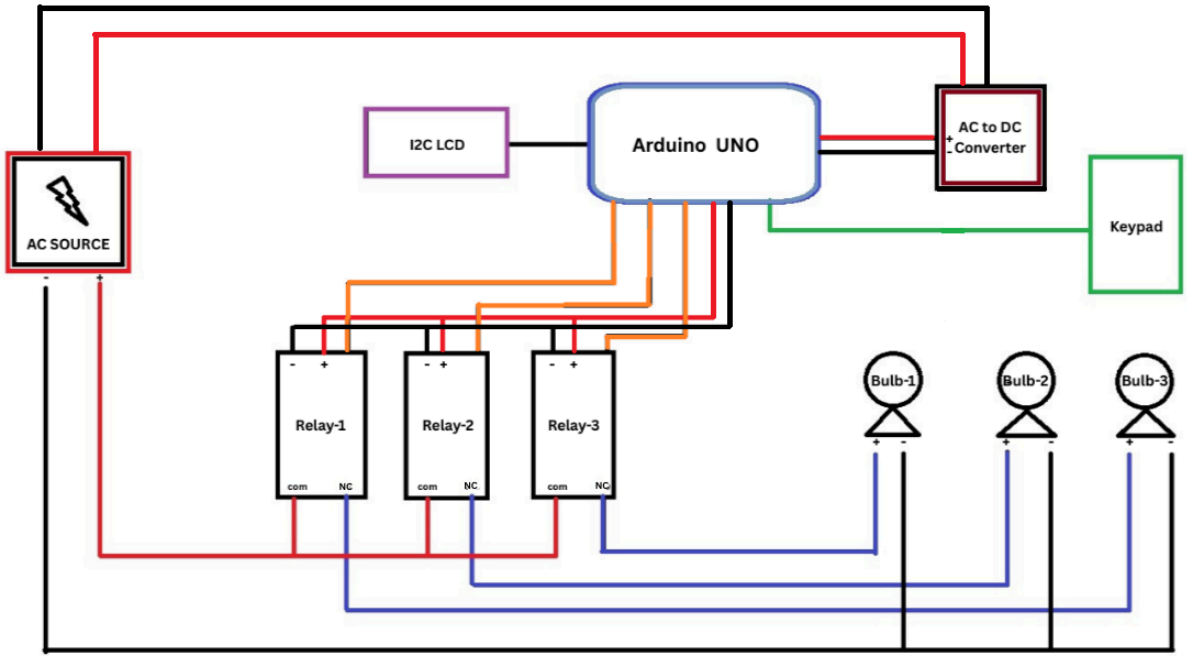
- The **Arduino Uno** acts as the brain of the system.
- It receives input from the **4×4 keypad**, which allows the user to select loads, enter countdown times, or toggle bulbs manually.
- The Arduino also controls the **16×2 I2C LCD display**, which shows the ON/OFF status of each load, the countdown timer, and user prompts.

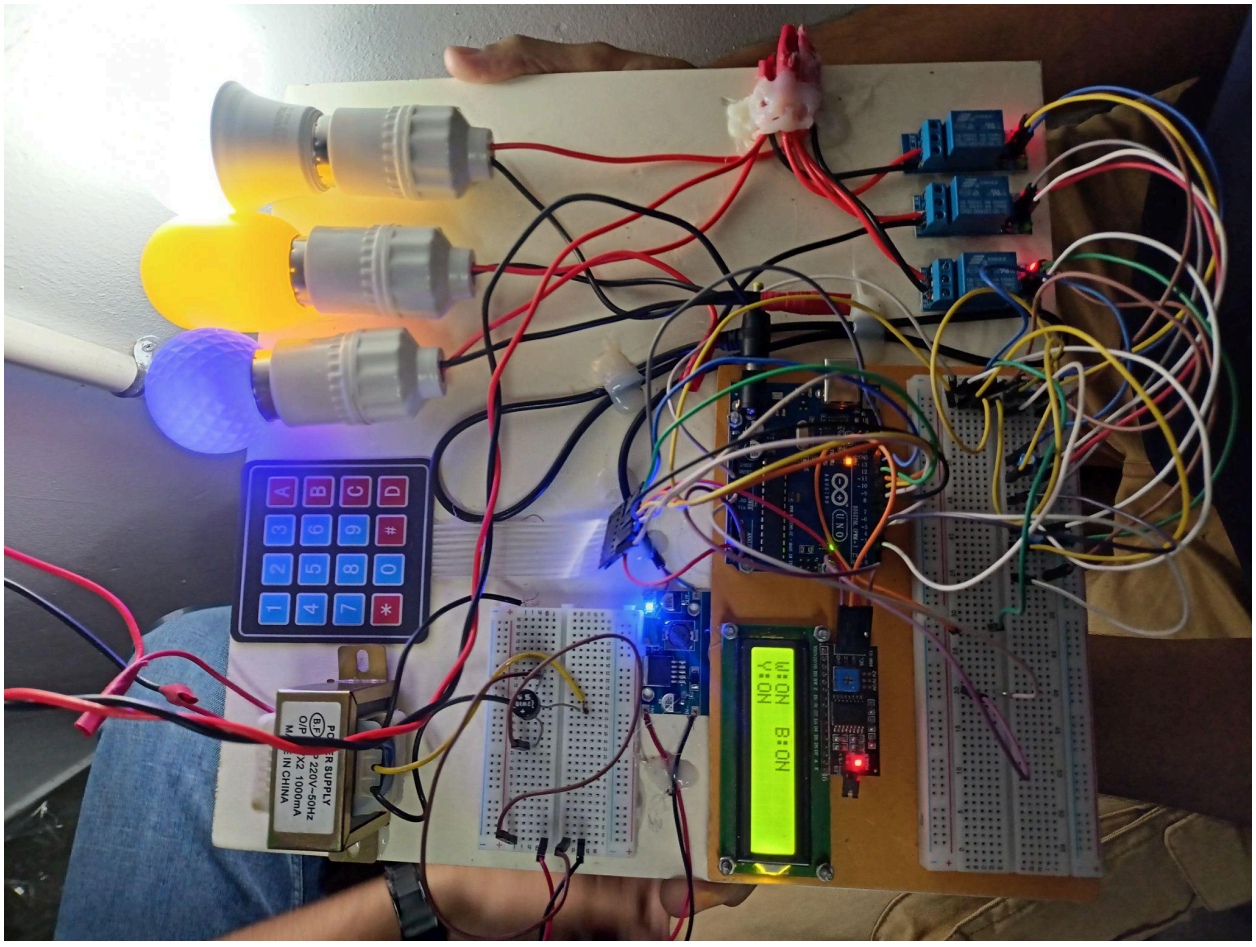
3. Switching Unit (Relay Module)

- Three **5 V relay modules** are used, each connected to a separate bulb (Relay-1 for Bulb-1, Relay-2 for Bulb-2, Relay-3 for Bulb-3).
- Each relay works as an electrically isolated switch, driven by the Arduino.
- The **COM terminal** of each relay is connected to the AC live line, while the Normally Open (NO) terminal goes to the bulb.
- When the Arduino outputs a HIGH signal, the relay energizes and allows current to flow to the respective bulb.

4. Load Unit

- The loads are represented by three bulbs (White, Blue, Yellow).
- The AC neutral line is connected directly to all bulbs, while the live line passes through each relay.
- This ensures that the Arduino can independently control the ON/OFF state of each bulb safely through the relay switching mechanism.


I2C LCD
Arduino UNO
AC to DC Converter
AC SOURCE
Keypad
Relay-1
Relay-2
Relay-3
com
NC
Bulb-1
Bulb-2
Bulb-3


W:ON
Y:ON
B:ON

5. Working Principle

1. System Initialization

- When the system is powered ON, the Arduino initializes the LCD display and relay outputs.
- By default, all three loads (White, Blue, Yellow bulbs) are switched ON, which is immediately reflected on the LCD as **ON status**.
- This ensures the system starts in a stable and known state.

2. Manual Load Control

- The user can manually toggle the status of each load directly using the keypad:
    - Pressing key 1 toggles the White bulb.
    - Pressing key 4 toggles the Blue bulb.
    - Pressing key 7 toggles the Yellow bulb.
- Each press immediately updates the relay status and the LCD display, giving real-time feedback.
- This mode provides the user with direct control in case of urgent or priority needs.

3. Load Selection for Countdown

- To automate load control, the user first selects which loads should be managed.

- This is done by pressing keys `A`, `B`, and `C` to choose White, Blue, or Yellow bulbs respectively.
- Multiple loads can be selected at once, and their short names appear on the LCD under “Chosen.”
- Key `*` is used to confirm the selection and proceed to time entry mode.
- If the user wants to clear the selection, pressing `C` cancels all chosen loads.

4. Countdown Time Entry

- Once loads are selected, the user enters the countdown duration in **seconds** using the numeric keys.
- The LCD shows “Time (sec):” and displays the entered digits in real time.
- After entering the desired countdown value, pressing key `D` confirms the entry.
- If the user presses `#` during entry, the process is canceled, and the system returns to the status screen.

5. Automated Countdown Operation

- Immediately after confirmation, the selected loads are switched OFF (relay set to LOW).
- The Arduino starts the countdown timer in the background while continuously updating the LCD with the remaining seconds for the currently active load.

- The LCD shows both the name of the load being managed and the time left before it will be restored.

- This process allows loads to be temporarily disconnected during peak demand or simulated load-shedding periods.

6. Automatic Restoration of Loads

- Once the countdown timer for a load expires, the Arduino automatically switches the corresponding relay back to HIGH, turning the load ON again.

- The LCD display updates to reflect the restored status.

- If multiple loads are under countdown, the system handles them sequentially and updates the display accordingly until all timers expire.

7. Cancel and Reset Options

- The user can interrupt any selection or time entry process by pressing key #.

- This immediately cancels the operation and resets the system back to the main status screen.

- This ensures flexibility and prevents accidental programming.

6. Advantages

- Simple and low-cost design
- Automatic and manual load control
- Real-time monitoring on LCD
- Safe switching using relays
- Portable and scalable system

7. Future Improvements

- Adding a **current sensor (ACS712/INA219)** for automatic overload-based load shedding
- Adding a **buzzer** for alerts
- Allowing user-defined current thresholds via keypad
- Integration of **RTC module (DS3231)** for real-time scheduling
- Remote control via Bluetooth, Wi-Fi, or IoT

8. Results

The Load Shedding Time Management System was successfully implemented and tested in the laboratory. The following observations were recorded:

1. **System Initialization**

- On powering the circuit, all three bulbs (White, Blue, Yellow) turned ON as designed.
- The LCD immediately displayed the ON/OFF status of each bulb, confirming proper initialization.

2. **Manual Operation**
   - Using keys `1`, `4`, and `7`, the bulbs could be toggled ON and OFF individually.
   - The relay response was instantaneous, and the LCD updated in real time without delay.
3. **Countdown Operation**
   - Bulbs selected using keys `A`, `B`, `C` successfully entered countdown mode.
   - After entering the time and pressing `D`, the selected bulbs switched OFF immediately, and the LCD displayed the countdown in seconds.
   - Once the timer expired, the bulbs automatically turned back ON, confirming correct countdown logic.
4. **System Stability**
   - The power supply (transformer + rectifier + buck converter) provided a stable 12 V DC to the Arduino, ensuring smooth and reliable performance.
   - No flickering or misoperation was observed during testing.
5. **Practical Verification**

- The system demonstrated both **manual control** and **automated scheduling** effectively.
- All components (Arduino, LCD, keypad, relays, buck converter) performed according to design specifications.

Final Outcome

The project achieved its objectives by automating load control through time scheduling and manual toggling, with real-time display feedback. The successful demonstration proved that the system is reliable, practical, and suitable for household-level load shedding management.

10. Conclusion

The Arduino-based Load Shedding Time Management System has proven to be an effective and reliable solution for automating load management in small-scale power systems. By integrating relays, a keypad interface, and an LCD display, the system successfully combines both manual and automated control, reducing dependency on human intervention and ensuring fair distribution of power during load shedding.

The use of separate live wires for each relay ensured safe and proper disconnection of loads, preventing leakage or unintended operation. From the power perspective, the 12 V transformer provided 12 V RMS (≈17 V peak), which after rectification and filtering measured about 15–16 V DC. To protect the Arduino and ensure stability, a buck converter was employed to regulate the supply to a steady 12 V DC.

Overall, the system demonstrated reliable performance during testing and fulfilled all project objectives. With minor improvements such as current sensing, real-time clock scheduling, and IoT-based remote control, this design can be adapted for practical applications in households, commercial facilities, and even small-scale industries.