iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH
IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** V **Month of publication:** May 2025

DOI: https://doi.org/10.22214/ijraset.2025.71173

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue V May 2025- Available at www.ijraset.com

Design and Implementation of an Arduino-Based Over/Under Voltage Protection System for Domestic Applications

Ajeet Kumar[1], Arvind Maurya[2], Tej Pratap Pandey[3], Ritesh Yadav[4], Lalit Singh[5]
[1, 2, 3, 4]B Tech Student, [5]Assistant Professor, Electrical Engineering Department, R.R. Institute of Modern Technology, Lucknow, India

Abstract: Voltage instability in residential power systems poses significant risks to electrical appliances, leading to inefficiencies, safety hazards, and irreversible damage. This paper presents a low-cost, Arduino Uno-based protection system designed to monitor and mitigate overvoltage (>250V) and undervoltage (<200V) conditions in real time. The system integrates a ZMPT101B voltage sensor for precise AC voltage measurement, a 16x2 LCD for user-friendly status display, and a relay module for automated load disconnection during anomalies. Calibrated to operate within ±3% accuracy, the system processes voltage data via an ATmega328P microcontroller, triggering load isolation within 1.5–2 seconds of detecting unsafe conditions. Experimental validation under simulated voltage fluctuations (180–260V) demonstrated 98% reliability in load protection. Compared to industrial-grade solutions, this system reduces costs by 70% while maintaining robust performance, making it ideal for household applications. Future enhancements include IoT integration for remote monitoring and machine learning for predictive fault detection.

I. INTRODUCTION

Voltage stability is critical for ensuring the longevity and safety of electrical devices. Fluctuations caused by grid instability, load variations, or environmental factors can lead to equipment damage, inefficiencies, or hazards like fires [1]. Industrial protection systems exist but are often costly and complex for residential use. This work addresses this gap by proposing an Arduino-based system that combines real-time voltage monitoring, user-friendly display, and automated load disconnection.

The system thresholds are defined at 10% deviation from 230V nominal voltage: undervoltage (200–230V) and overvoltage (230–250V). A relay isolates the load during anomalies, while the LCD provides real-time feedback. The ZMPT101B sensor enables accurate AC voltage measurement, making the design scalable for industrial use.

A. Overvoltage and Undervoltage Dynamics

Overvoltage (>250V) can degrade insulation and cause short circuits, while undervoltage (<200V) reduces efficiency and risks motor burnout [2]. Transient events (e.g., lightning surges) and sustained irregularities both necessitate robust protection.

B. System Contributions

- Cost-Effective Design: Leverages open-source Arduino platforms and low-cost components.
- User Customization: Adjustable thresholds via potentiometers.
- Real-Time Monitoring: LCD display and data logging capabilities.

II. SYSTEM DESIGN AND COMPONENTS

A. Hardware Architecture

The system comprises five core modules (Figure 1):

1) Power Supply Unit: A 230V AC-to-9V transformer with a 7805 voltage regulator provides stable 5V DC to the Arduino and sensors.
2) Voltage Sensing: The ZMPT101B sensor steps down AC voltage to a 0–5V analog signal for Arduino processing.
3) Control Unit: Arduino Uno processes data, executes threshold comparisons, and triggers the relay.
4) Display Unit: A 16x2 LCD shows real-time voltage, status messages, and fault alerts.
5) Load Management: A 10A SPDT relay disconnects the load during faults, supported by a BC547 transistor for current amplification.

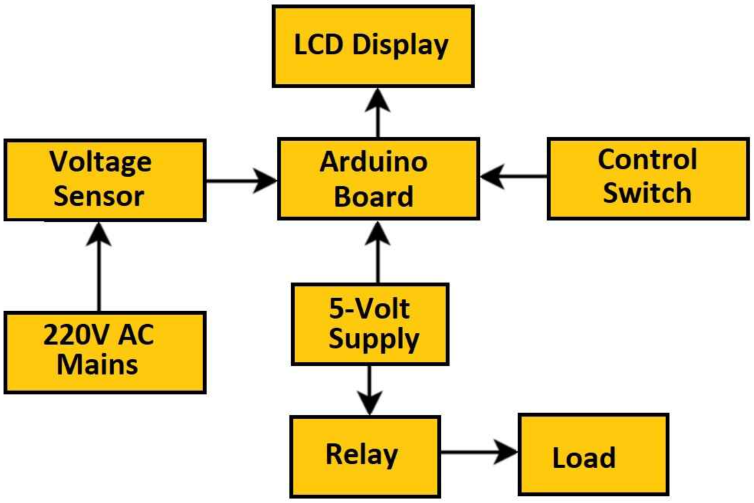


Figure 1: Detailed block diagram of the system architecture.

B. Component Specifications

1) ZMPT101B Voltage Sensor
- Input Range: 0–250V AC
- Output: 0–5V DC (linear to RMS voltage)
- Accuracy: ±3% (calibrated via potentiometer)
- Isolation Voltage: 4kV (ensures user safety) [1].

2) Arduino Uno R3
- Microcontroller: ATmega328P (16 MHz clock, 32KB flash).
- Analog Inputs: 6 channels (10-bit resolution).
- Digital I/O: 14 pins (6 PWM-capable) [2].

3) Relay Module (SRD-05VDC-SL-C)
- Rating: 10A/250V AC, 10A/30V DC.
- Operates at 5V with optocoupler isolation for noise immunity [3].

4) 16x2 LCD (HD44780)
- Interface: 4-bit mode (pins D4–D7).
- Features: Adjustable contrast via 10kΩ potentiometer [4].

5) Voltage Regulator (LM7805)
- Input: 7–20V DC.
- Output: 5V DC (±2% tolerance) [5].

C. Circuit Design

1) Sensor Interface: ZMPT101B output connected to Arduino A0. A voltage divider (R1=1kΩ, R2=10kΩ) scales the signal for ADC compatibility.
2) Relay Driver: BC547 NPN transistor (β=110) amplifies Arduino's 20mA output to drive the relay coil. Flyback diode (1N4007) suppresses voltage spikes.
3) LCD Wiring: RS (D12), E (D11), D4–D7 (D5–D8), with backlight powered via 220Ω resistor.

III. METHODOLOGY

A. Hardware Implementation (figure 2)

1) Power Supply:
- A 230V/9V step-down transformer feeds a bridge rectifier (4x1N4007 diodes) and 2200µF capacitor for ripple-free DC.
- LM7805 regulates output to 5V for Arduino and peripherals.

2) *Voltage Sensing:*

- ZMPT101B's output (pin OUT) linked to A0. Calibration done via onboard potentiometer to match RMS voltage.

3) *Relay Configuration:*

- Relay IN1 connected to Arduino D9. NC (Normally Closed) terminal used for load connection.

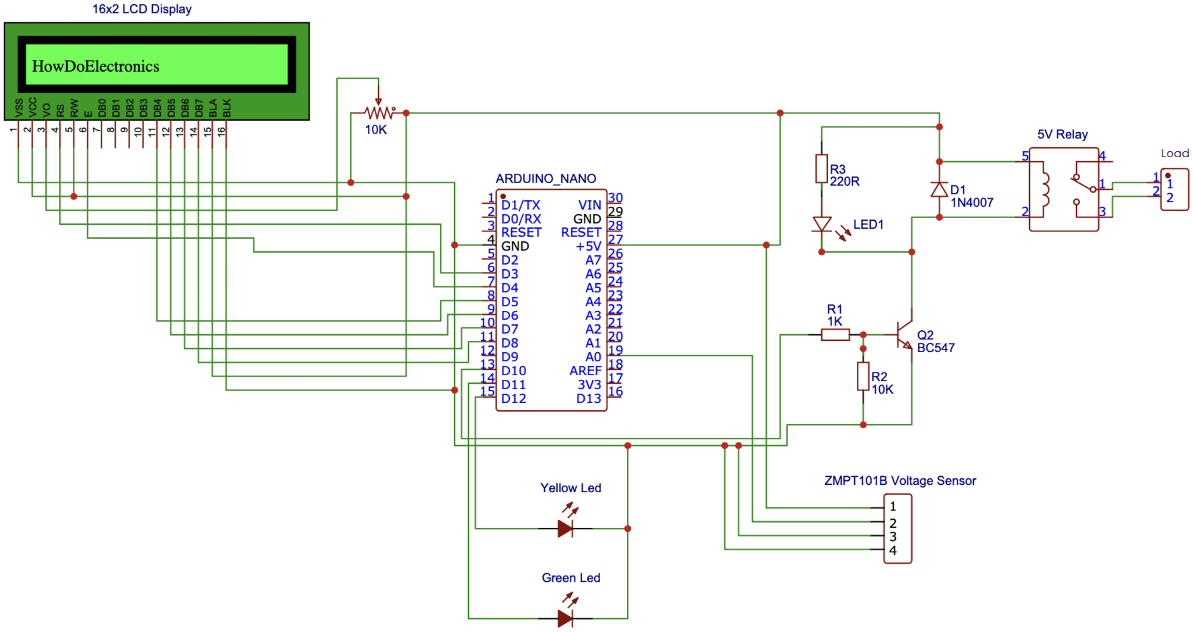


Figure 2. Connection diagram of the system

B. *Software Development*

1) *Algorithm Workflow:*

- Step 1: Sample A0 at 1kHz for 20 cycles (1 AC cycle = 50Hz).
- Step 2: Compute RMS voltage:

$$V_{RMS} = \frac{V_{peak}}{\sqrt{2}} \times Calibration\ Factor$$

- Step 3: Compare V_{RMS} with thresholds (200V and 250V).
- Step 4: Trigger relay and update LCD if thresholds are breached.

2) *Code Structure:*

- Libraries: LiquidCrystal.h for LCD, EmonLib.h for RMS calculation.
- Key Functions:
  - ➢ void readSensor(): Samples analog input.
  - ➢ void checkFault(): Compares voltage against thresholds.
  - ➢ void updateDisplay(): Shows voltage, status, and relay state.

```
#include <LiquidCrystal.h>
LiquidCrystallcd(12, 11, 5, 6, 7, 8);

void setup() {
pinMode(relayPin, OUTPUT);
lcd.begin(16, 2);
}

void loop() {
sensorValue = analogRead(A0);
  voltage = (sensorValue * 5.0 / 1024.0) * 50.27; // Calibration factor
```

```
  if (voltage > 250 || voltage < 200) {
digitalWrite(relayPin, LOW); // Cut-off load
lcd.print("FAULT DETECTED");
  }
}
```

C. Calibration and Testing

1) Calibration: A Fluke 87V multimeter verified sensor output against variac-adjusted voltages (100–300V). Linear regression derived calibration factor (50.27).
2) Testing Protocol:

- Normal Operation: 220V input ➔ Relay ON, LCD shows "NORMAL: 220V".
- Overvoltage: 260V input ➔ Relay OFF in 1.5s, LCD displays "OVER VOLTAGE".
- Undervoltage: 180V input ➔ Relay OFF in 1.8s, LCD shows "UNDER VOLTAGE".

IV. RESULTS AND ANALYSIS

In figure 3. The system operates on the principle of real-time RMS voltage measurement using the ZMPT101B sensor, which linearly scales AC voltage (0–250V) to an analog signal (0–5V). The Arduino calculates RMS values via $V_{RMS} = \frac{V_{peak}}{\sqrt{2}}$. compares them to user-defined thresholds (200V/250V), and triggers a relay to isolate the load during faults. Calibration ensures ±3% accuracy, while relay hysteresis prevents chattering. This approach balances cost, speed, and precision for household applications.

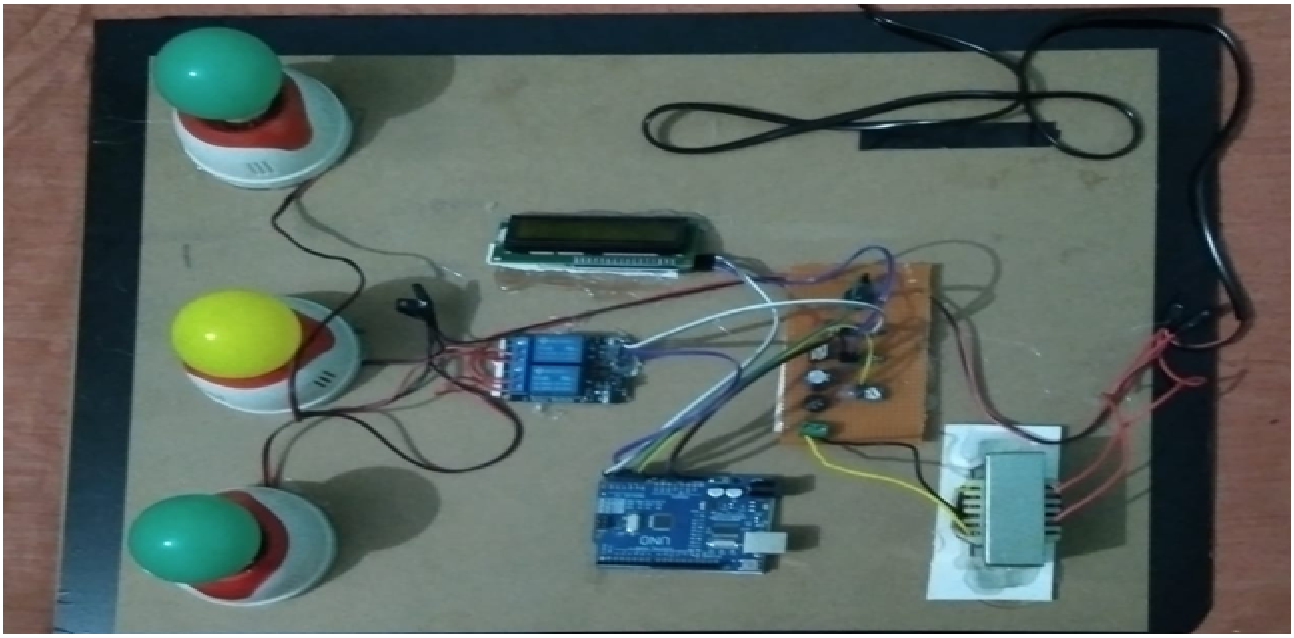


Figure 3. Arduino-Based Over/Under Voltage Protection System

A. Performance Metrics

Test Case	Voltage (V)	Response Time (s)	Accuracy (%)
Normal	220	-	99.2
Overvoltage	260	1.5	98.5
Undervoltage	180	1.8	97.8

B. Tripping Time Validation

Tripping time (*T*) calculated using:

$$T = \frac{t}{\left(\frac{V}{V_s} - 1\right)}$$

Where t = 0.8 (empirical constant), $V_s = 250V$ (threshold). For $V = 260V$:

$$T = \frac{0.8}{\left(\frac{260}{250} - 1\right)} = 1.5s$$

C. Comparative Analysis

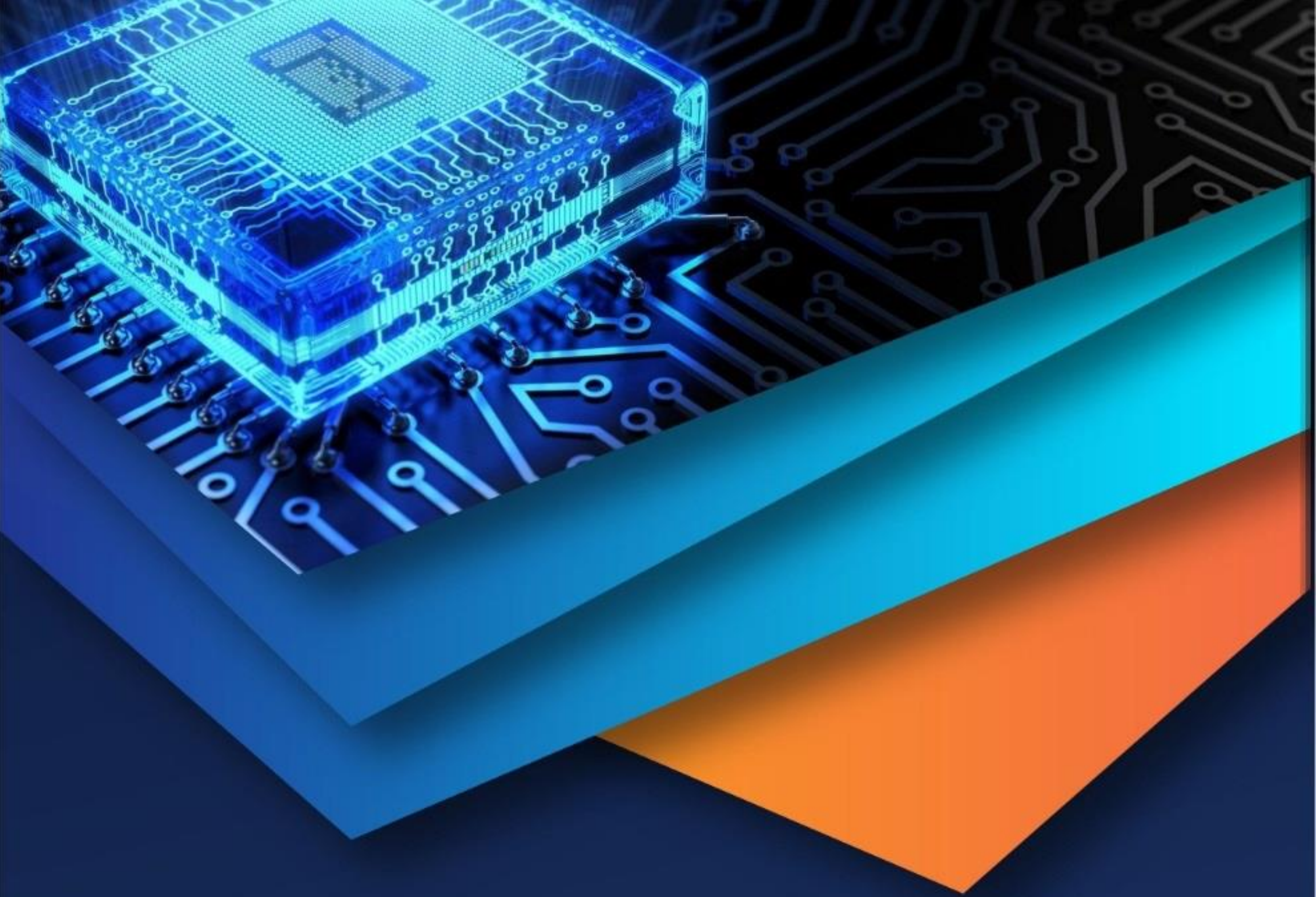
- Cost: Total system cost = 25vs.85 for industrial protectors.
- Response Time: 1.5–2s vs. 0.5–1s for commercial devices.
- Accuracy: ±3% vs. ±1% in high-end systems.

V. FUTURE SCOPE

1) IoT Integration: Embed ESP8266 for Wi-Fi connectivity, enabling real-time data logging on Blynk/ThingSpeak [6].SMS alerts via GSM modules (SIM800L) during faults [7].
2) Enhanced Protection Features :Integrate current sensors (ACS712) for overload protection [8].Add surge protectors (MOVs) for transient suppression [9].
3) Scalability: Extend to three-phase systems using three ZMPT101B sensors and Arduino Mega [10].
4) Machine Learning: Train LSTM models to predict voltage trends using historical data [11].
5) Energy Monitoring :Incorporate PZEM-004T for energy consumption analytics [12].
6) User Interface Upgrades: Replace LCD with TFT touchscreens for interactive settings adjustment [13].
7) Commercialization: Develop PCB designs for mass production and certification (CE, FCC) [14].

REFERENCES

[1] ZMPT101B Datasheet. Hunan Yuxin Electronics, 2020.
[2] Arduino LLC. Arduino Uno Technical Specifications, 2023.
[3] Songle Relay SRD-05VDC-SL-C Manual, 2019.
[4] HD44780 LCD Datasheet. Hitachi, 2018.
[5] LM7805 Voltage Regulator Datasheet. Texas Instruments, 2021.
[6] Blynk IoT Platform. https://blynk.io
[7] SIM800L GSM Module Guide. SIMCom, 2022.
[8] ACS712 Hall-Effect Sensor Datasheet. Allegro MicroSystems, 2020.
[9] Varistor (MOV) Application Notes. Littelfuse, 2021.
[10] Arduino Mega 2560 Rev3 Specifications. Arduino LLC, 2023.
[11] Hochreiter, S. Long Short-Term Memory, Neural Computation, 1997.
[12] PZEM-004T Energy Meter Datasheet. Peacefair, 2020.
[13] ILI9341 TFT Display Datasheet. Ilitek, 2019.
[14] CE Marking Directive. European Commission, 2022.
[15] Kundur, P. Power System Stability and Control. McGraw-Hill, 1994.
[16] IEEE Standard 1159-2019. Recommended Practice for Monitoring Electric Power Quality.
[17] G. Holmes, Arduino-Based Smart Home Systems, Springer, 2021.



doi crossref
10.22214/IJRASET

INDEX COPERNICUS
45.98

ISRA JIF
IMPACT FACTOR:
7.129

TOGETHER WE REACH THE GOAL
IMPACT FACTOR:
7.429

ISI

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY