A

PROJECT REPORT
ON

OVER VOLTAGE AND OVER CURRENT PROTECTION SYSTEM

Submitted By:
BOMMISETTY LOHITHA

Aim:

This project aims to design and implement an Over Voltage and Over Current Protection System using an Arduino microcontroller. The system is intended to monitor electrical parameters and disconnect the load when voltage or current exceeds predefined thresholds. This is crucial for protecting electrical devices from damage due to overvoltage or overcurrent conditions.

introduction:

Electrical devices are susceptible to damage from overvoltage and overcurrent conditions. Overvoltage can occur due to lightning strikes, power surges, or faulty wiring, while overcurrent can result from short circuits or excessive load. This project utilizes an Arduino microcontroller to monitor voltage and current levels and activate protective measures when necessary.

Objectives:

To design a system that continuously monitors voltage and current levels.

- To implement a mechanism to disconnect the load when overvoltage or overcurrent is detected.
- To provide visual and audible alerts for users when protection is activated.
- To ensure the system is user-friendly and easy to calibrate.

Components Required:

1. Arduino Uno

Description: The Arduino Uno is a microcontroller board based on the ATmega328P. It is widely used in electronics projects due to its ease of use, versatility, and extensive community support.

Function in the Project:

- Acts as the central processing unit (CPU) of the system.
- Reads analog signals from the voltage and current sensors.
- Processes the data to determine if the voltage or current exceeds predefined thresholds.
- Controls the relay to disconnect the load when necessary.
- Provides visual and audible alerts through LEDs and a buzzer.

2. Voltage Sensor Module

Description: A voltage sensor module is designed to measure the voltage of an electrical circuit. It typically consists of a voltage divider circuit that scales down the voltage to a level that can be safely read by the Arduino.

Function in the Project:

- Monitors the input voltage.
- Outputs an analog signal proportional to the measured voltage, which the Arduino reads and processes.
- Helps determine if the voltage exceeds the safe operating limit.

3. Current Sensor Module (e.g., ACS712)

Description: The ACS712 is a Hall-effect based current sensor that can measure both AC and DC currents. It provides an analog output that is proportional to the current flowing through it.

Function in the Project:

- Measures the current flowing to the load.
- Outputs an analog signal that the Arduino reads to determine the current level.
- Helps detect overcurrent conditions that could damage the load.

4. Relay Module

Description: A relay is an electrically operated switch that allows a low-power signal to control a high-power circuit. Relay modules typically include a relay and necessary circuitry to interface with microcontrollers.

Function in the Project:

- Controls the connection and disconnection of the load based on signals from the Arduino.
- When the Arduino detects overvoltage or overcurrent, it activates the relay to disconnect the load, preventing damage.

5. Buzzer

Description: A buzzer is an audio signaling device that produces sound when an electrical signal is applied. It can be either active (producing sound when powered) or passive (requiring a specific frequency to produce sound).

Function in the Project:

- Provides an audible alert when the system detects overvoltage or overcurrent conditions.
- Alerts users to take action, such as checking the system or resetting it.

6. LEDs

Description: Light Emitting Diodes (LEDs) are semiconductor devices that emit light when an electric current passes through them. They are commonly used for indicators and displays.

Function in the Project:

- Serve as visual indicators of the system's status.
- One LED can indicate normal operation (green), while another can indicate an alert condition (red) when overvoltage or overcurrent is detected.

7. Resistors

Description: Resistors are passive electrical components that limit the flow of electric current in a circuit. They are used to control voltage and current levels.

Function in the Project:

- Used in the voltage sensor circuit to create a voltage divider, ensuring that the voltage fed to the Arduino is within safe limits.
- May also be used to limit current to the LEDs.

8. Breadboard and Jumper Wires

Description: A breadboard is a reusable platform for prototyping electronic circuits without soldering. Jumper wires are used to make connections between components on the breadboard.

Function in the Project:

- Facilitate the assembly of the circuit for testing and prototyping.
- Allow for easy modifications and adjustments during the development phase.

9. Power Supply

Description: A power supply provides the necessary electrical power to the Arduino and other components in the circuit.

Function in the Project:

- Supplies power to the Arduino and sensors, ensuring they operate correctly.
- Can be a battery, USB power, or an external AC-DC adapter, depending on the project requirements.

Connections:

1. arduino

Power Supply: Connect the Arduino to a power source (USB or external power supply).

1. **Voltage Sensor Module**
   - **VCC**: Connect to the 5V pin on the Arduino.
   - **GND**: Connect to the GND pin on the Arduino.
   - **OUT**: Connect to an analog pin on the Arduino (e.g., A0).
2. **Current Sensor Module (ACS712)**
   - **VCC**: Connect to the 5V pin on the Arduino.
   - **GND**: Connect to the GND pin on the Arduino.
   - **OUT**: Connect to another analog pin on the Arduino (e.g., A1).
3. **Relay Module**
   - **VCC**: Connect to the 5V pin on the Arduino.
   - **GND**: Connect to the GND pin on the Arduino.
   - **IN**: Connect to a digital pin on the Arduino (e.g., D7).
   - **Common (COM)**: Connect to one terminal of the load (e.g., a lamp or motor).
   - **Normally Open (NO)**: Connect to the power supply (e.g., AC or DC source).
   - **Normally Closed (NC)**: Leave unconnected for this application.
4. **Buzzer**
   - **Positive (+)**: Connect to a digital pin on the Arduino (e.g., D8).
   - **Negative (-)**: Connect to the GND pin on the Arduino.
5. **LEDs**

- **Anode (+)**: Connect to a digital pin on the Arduino (e.g., D9 for the alert LED).
- **Cathode (-)**: Connect to a resistor (e.g., 220Ω) and then to the GND pin on the Arduino.

6. **Resistors**

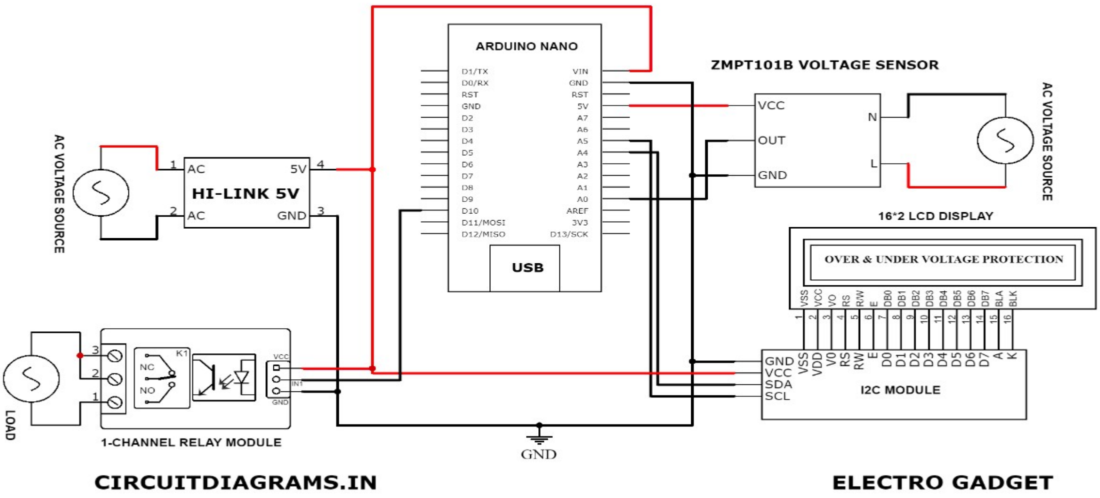
- Use a resistor (e.g., 220Ω) in series with the LED to limit the current and prevent damage to the LED.

Program:

```
#include <Arduino.h>

const int voltagePin = A0;

const int currentPin = A1;

const int relayPin = 7;

const int buzzerPin = 8;

const int ledPin = 9;

// Thresholds

const float voltageThreshold = 240.0;

const float currentThreshold = 10.0;

// Calibration Constants

const float voltageCalibration = 300.0 / 5.0;

const float currentCalibration = 30.0 / 5.0;

void setup() {

    pinMode(relayPin, OUTPUT);

    pinMode(buzzerPin, OUTPUT);

    pinMode(ledPin, OUTPUT);

    Serial.begin(9600);
```

```
    digitalWrite(relayPin, HIGH);
}
void loop() {
    float voltage = analogRead(voltagePin) * (5.0 / 1023.0) * voltageCalibration;
    float current = analogRead(currentPin) * (5.0 / 1023.0) * currentCalibration;
  // Print voltage and current to Serial Monitor
    Serial.print("Voltage: ");
    Serial.print(voltage);
    Serial.print(" V, Current: ");
    Serial.print(current);
    Serial.println(" A");
    // Check condition
    if (voltage > voltageThreshold || current > currentThreshold) {
        digitalWrite(relayPin, LOW); // Disconnect load
        digitalWrite(buzzerPin, HIGH); // Activate buzzer
        digitalWrite(ledPin, HIGH); // Turn on LED
    } else {
        digitalWrite(relayPin, HIGH); // Connect load
        digitalWrite(buzzerPin, LOW); // Deactivate buzzer
        digitalWrite(ledPin, LOW); // Turn off LED
    }
    delay(1000); // Delay for stability
}
```

SCHEMATIC:



Applications

- **Industrial Equipment**: Safeguarding industrial machinery from electrical faults that could lead to equipment failure or safety hazards.
- **Renewable Energy Systems**: Protecting solar inverters and battery systems from overvoltage and overcurrent conditions.

Future scope:

The future scope of the Over Voltage and Over Current Protection System using Arduino is vast and varied. By incorporating advanced features, enhancing monitoring capabilities, integrating with IoT, and improving user interfaces, the system can evolve into a comprehensive solution for electrical safety and efficiency. These enhancements can lead to broader applications in residential, commercial, and industrial settings, ultimately contributing to safer and more reliable electrical systems.

Conclusion:

In conclusion , this project serves as a foundational step toward developing smarter, safer, and more efficient electrical systems. As technology continues to advance, the integration of such protective systems will become increasingly vital in safeguarding our electrical infrastructure and devices.