

Electricity Anti-Theft System - Over Loading Detection and Automatic Shut Down

Electricity theft is a common problem. Due to electricity theft, not only there are losses to the revenue of the government but over loading and damage to the transformers is an unwanted consequence of the theft as well. This project is an attempt to resolve electricity theft problem. The project detects the theft by detecting over loading at the transformer. Once the over loading is detected, a relay circuit is employed to shut down the power supply from the respective transformer. Definitely, power shut down on doing electricity theft will discourage the mischievous elements.

The project demonstrated here is to show the mechanism of electricity anti-theft system. A pack of parallel connected bulbs is used as load while the programmable circuit operating the supply controlling relay is based on Arduino. The principle used behind detecting overloading is voltage comparison and an additional buzzer circuit is attached to the project to alert electricity theft in the locality. The buzzer mechanism has been added to alert people of electricity theft and demoralize the mischievous elements.

The project has the following execution cycle:

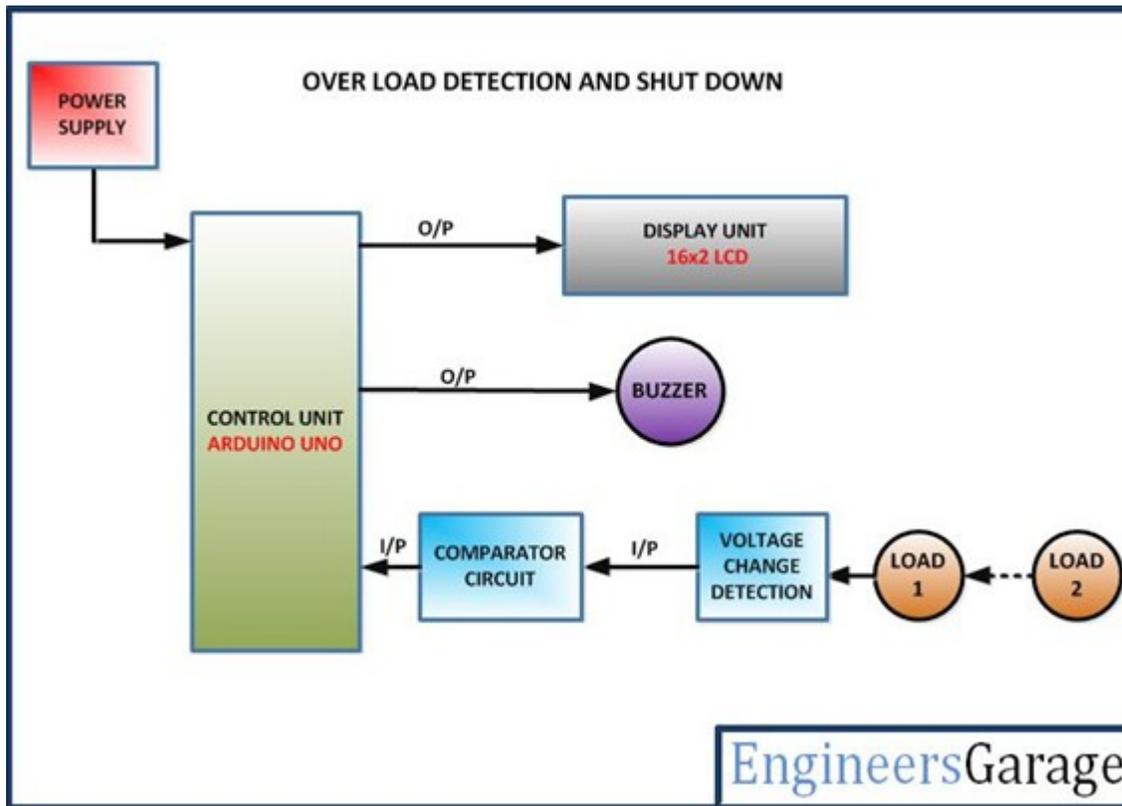
- 1) The project will be connected at the primary coil of an additional transformer which will have its secondary coil connected between the main line and loads while the loads (a pack of bulbs is used as load in the demonstration) will be connected to the main line via relay interfaced to main line through secondary coil of the additional transformer. This additional transformer shall work like a voltage sensor in the circuit.
- 2) The Arduino based circuit will show the status of the main line as overloaded or not on an LCD display. The LCD display section is however optional and added for testing and demonstration purpose only.
- 3) When system detects over load condition which will be transferred as a HIGH signal to the voltage comparator interfaced pin of Arduino, the board will output signals to trigger the relay circuit.
- 4) The relay will cut off the power supply from the main line and the whole system will get shut down.
- 5) After a shut down the system will wait for conditions to normalize and will resume the power supply as unauthorized loads are detached and relevant voltage is detected at the primary of transformer-based sensor circuit

Components Required:

- 1) Arduino UNO
- 2) Transformer 12 - 0v
- 3) Buzzer
- 4) 16x2 LCD
- 5) 12v Relay
- 6) BC 547 transistor

- 7) 1K ohm resistors -7
- 8) Voltage regulator -7805
- 9) LED -5mm RED- 2
- 10) Wire to carry 2A, 230v AC
- 11) Two pin Plug
- 12) Bulb holder

BLOCK DIAGRAM



CIRCUIT DIAGRAM

- 1) Power Supply: While the loads will be running on AC supply, the project circuit will require 12V and 5V DC supply. The 12V supply will be provided via a 12V DC Battery and will be provided to 12V relay and 7805 voltage regulator. The other components will need a 5V DC supply which will be provided via 7805 voltage regulator that is converting the input 12V supply to 5V DC.

- 2) Transformer as Sensor: It is well known that transformer works on the principle of mutual induction. The loads (bulbs in this case) are connected through the secondary coil of the transformer with the main line and are parallel connected. This is similar to wiring system of houses in which all the power sockets have parallel connections with the main supply. Therefore, in any house all the electrical appliances are basically connected in parallel. When the loads (bulbs) starts conducting current, the current also flows the secondary coil added transformer and due to mutual induction, some voltage is induced at the primary coil of the added transformer. The voltage induced increases as the more number of loads are connected to the main line. The voltage gain is feed to the comparator circuit. The comparator circuit compares the change in voltage and sends a HIGH logic to the connected pin of the Arduino board.

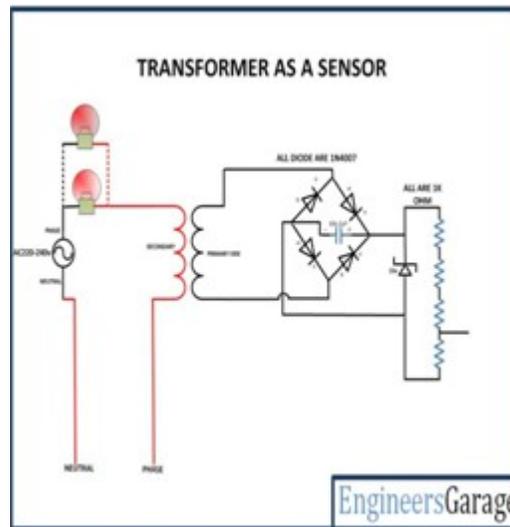


Figure: Using Transformer as Overload Detector

- 3) LCD Display: The 16X2 LCD display is connected to the Arduino board by connecting its data pins to pins 3 to 6 of the Arduino board. The RS and RW pin of LCD is connected to pins 12 and 13 of the Arduino UNO respectively. The E pin of the LCD is grounded.

| LCD | ARDUINO UNO |
|-------------|----------------------|
| RS | 12 |
| RW | 13 |
| E | GRND |
| D7,D6,D5,D4 | 3,4,5,6 Respectively |

The standard code library for interfacing Arduino UNO and Arduino Pro Mini are used in the project to program LCD with the board.

- 4) Relay Circuit: The loads are connected to the main line via a 12V relay. The relay is connected to the pin A0 of Arduino board through BC547 transistor circuit which is in a common emitter configuration.

How the Circuit Works

The circuit works in following stages - :

- 1) Initialization of the circuit

As the circuit is switched on Arduino loads the imported libraries and initialize LCD display and the relay circuit. First Initial messages are flashed on the display screen showing no overloading condition and data from voltage comparator is fetched from the A1 pin of Arduino UNO board. A LOW signal is set at the pin 10 of Arduino UNO connected to the relay which switches the relay to an ON state and keeps the power supply through main line ON.

- 2) Switching Relay

While continuous polling logic from the voltage comparator, at an instance, voltage at A1 pin of Arduino UNO gets greater than the reference voltage. This prompts the Arduino board to set a HIGH signal at pin 10. An LED connected parallel to the relay lights on to alert that the overloading condition. The relay is connected to the pin 10 via transistor BC547. The BC547 is configured in common emitter configuration. The base of the transistor is connected to pin 10 through a 1K ohm resistor and collector of the transistor is connected to the relay coil. The emitter is grounded. When a HIGH signal is received from pin 10, current starts flowing from base to emitter and collector becomes grounded. The other end of the relay coil is connected to 12V supply, thus as collector is grounded current starts flowing through the relay coil and the device is cut short of the power supply.

Programming Guide

The code utilizes standard open source libraries of the Arduino UNO. Therefore, first of all the standard library for LCD is imported and a variable 'relay' is declared which is assigned to pin A2 of the Arduino board.



```
overload_shut_down | Arduino 1.6.12
File Edit Sketch Tools Help
overload_shut_down
#include <LiquidCrystal.h>//import the LCD library
LiquidCrystal lcd(13, 12, 6, 5, 4, 3);// Pins used for RS,E,D4,D5,D6,D7
int x;
#define relay 7
#define buzzer 10
```

A setup() function is called where the relay and buzzer connected pins are set to digital output using pinMode() function and a LOW is passed to the respective pins through digitalWrite() function. The lcd is initialized and initial messages are flashed on it.

The image shows a screenshot of the Arduino IDE interface. The title bar reads "overload_shut_down | Arduino 1.6.12". The menu bar includes "File", "Edit", "Sketch", "Tools", and "Help". Below the menu bar is a toolbar with icons for saving, undo, redo, and other functions. The main workspace shows the code for the setup() function. The code is as follows:

```
void setup() {  
  lcd.begin(16,2);//LCD 16x2 initialization  
  pinMode(relay, OUTPUT);  
  pinMode(buzzer, OUTPUT);  
  digitalWrite(buzzer, LOW);  
  digitalWrite(relay, LOW);  
  lcd.setCursor(0,0); //Initially set the cursor position of LCD to 1st Columb 1st row.  
  lcd.print("Engineers Garage");//After initialising print data  
  lcd.setCursor(0,1); //Initially set the cursor position of LCD to 1st Columb 2nd row.  
  lcd.print("          "); //print blank to clear all the data on LCD  
  delay(3000);  
  lcd.setCursor(0,0);  
  lcd.print(" LOAD DETECTION ");  
  lcd.setCursor(0,1);  
  lcd.print(" AND SHUTDOWN ");  
  delay(3000);  
  lcd.setCursor(0,0);  
  lcd.print("          ");  
  lcd.setCursor(0,1);  
  lcd.print("          ");  
}
```

A loop() function is called where analog value of the voltage from voltage comparator is fetched and compared to a standard calibrated voltage range and if there is overloading detected, relay is operated to cut off the supply line, buzzer is set to alert and overloading message is displayed on the screen.



```
overload_shut_down | Arduino 1.6.12
File Edit Sketch Tools Help
overload_shut_down
void loop() {
  x=analogRead(A0)*(5.0/1023.0)*100;
  lcd.setCursor(0,0);
  lcd.print("  OVER LOAD      ");
  lcd.setCursor(0,1);
  lcd.print("  DETECTION  ");
  lcd.setCursor(13,1);
  lcd.print(x);
  if(x>385 && x <470){
    digitalWrite(relay, HIGH);
    lcd.setCursor(0,0);
    lcd.print("  LOAD DETECTED ");
    lcd.setCursor(0,1);
    lcd.print("                ");
    for(int i=0;i<3;i++){
      digitalWrite(buzzer,HIGH);
      delay(500);
      digitalWrite(buzzer,LOW);
      delay(500);
    }
  }
  else if(x<280){
    digitalWrite(relay, LOW);
  }
  else;
} // end loop
```

PROGRAMMING CODE

```
#include <LiquidCrystal.h>//import the LCD library

LiquidCrystal lcd(13, 12, 6, 5, 4, 3);// Pins used for RS,E,D4,D5,D6,D7

int x;

#define relay 7

#define buzzer 10
```

```

void setup() {
    lcd.begin(16,2);//LCD 16x2 initialization
    pinMode(relay, OUTPUT);
    pinMode(buzzer, OUTPUT);
    digitalWrite(buzzer, LOW);
    digitalWrite(relay, LOW);
    lcd.setCursor(0,0); //Initially set the cursor position of LCD to 1st Column 1st
row.
    lcd.print("Engineers Garage");//After initialising print data
    lcd.setCursor(0,1); //Initially set the cursor position of LCD to 1st Column 2nd
row.
    lcd.print("                "); //print blank to clear all the data on LCD
    delay(3000);
    lcd.setCursor(0,0);
    lcd.print(" LOAD DETECTION ");
    lcd.setCursor(0,1);
    lcd.print(" AND SHUTDOWN ");
    delay(3000);
    lcd.setCursor(0,0);
    lcd.print("                ");
    lcd.setCursor(0,1);
    lcd.print("                ");
}

void loop() {
    x=analogRead(A0) * (5.0/1023.0) *100;
    lcd.setCursor(0,0);
    lcd.print(" OVER LOAD ");
    lcd.setCursor(0,1);
    lcd.print(" DETECTION ");
    lcd.setCursor(13,1);
    lcd.print(x);
    if(x>385 && x <470){

```

```
digitalWrite(relay, HIGH);  
lcd.setCursor(0,0);  
lcd.print("  LOAD DETECTED ");  
lcd.setCursor(0,1);  
lcd.print("                ");  
for(int i=0;i<3;i++){  
digitalWrite(buzzer,HIGH);  
delay(500);  
digitalWrite(buzzer,LOW);  
delay(500);  
}  
}  
else if(x<280){  
digitalWrite(relay, LOW);  
}  
else;  
  
} // end loop
```