Khulna University of Engineering & Technology

Department of Electrical & Electronic Engineering

Sessional on Electrical and electronic project design

EE-3200

Project Title: Arduino Based Electrical Power Distribution Board for Home Appliances

Supervised by:

Dr. Md. Selim Hossain
Professor
Department of Electrical & Electronic Engineering (EEE),
Khulna University of Engineering & Technology (KUET), Khulna-9203, Bangladesh.

Submitted To:

Dr. Kalyan Kumar Halder
Professor
Department of Electrical & Electronic Engineering (EEE),
Khulna University of Engineering & Technology (KUET), Khulna-9203, Bangladesh.

Submitted By:

Shetu Mohanto
Roll: 1703018
3rd year, 2nd semester

Contents:

Abstract:

The problem with traditional electromechanical circuit breakers is that they can't be controlled digitally, and their current ratings are set by electromechanical design. They also don't provide voltage protection. An electrical power distribution board has been built in this project, which can provide overcurrent protection up to 7 amps and current protection for varied current ratings using the same circuit design and microcontroller program modification. It can also defend against overvoltage and undervoltage. It may measure the connected load's power factor, the frequency of the provided supply, the wattage of the load, and the energy consumption in kWh.

Introduction:

Block diagram of the project has been shown in fig-01 and the picture of the project in the fig-03.

In this project the main challenge was the conversion of 220-2240V ac supply and up to 7amp ac current to measurable level for Arduino board. This was achieved using potential transformer and current transformer with necessary circuit for conditioning the voltage for the Arduino. Current measuring transformer has been shown in fig-02. The given current sensor can withstand up to 100amp current but considering household load it has been conditioned to 7amp. If the current exceeds 7amp (rms) value, then load is disconnected from the supply by turning off the relay.

The voltage sensor section is designed for measuring voltage up to 245V (rms), program was done so that in case of voltage under 220V (rms) and for voltage more than 240V (rms) load is disconnected from the supply.

For either case overcurrent and over voltage or under voltage user can’t turn on the load without solving the use, in case of over current load must be reduced before turning on the supply otherwise program won’t give permission to turn on the load. Same way for the under or over voltage user can’t turn on the load, only when voltage is 220-240V (rms) user can turn on the relay.

List of equipment:

SL no.	Name	Model / Rating	Quantity	Cost (per quantity) BDT.
I.	Arduino UNO	R3	01	650
II.	4 Channel 5V Relay Module	2PH63083A, DC 30V/10A, AC 250V/10A, 15-20mA driver current	01	300
III.	Potential Transformer	I/P 220V-50Hz, O/P 12Vx2, 3000mA, Center Tap	01	110
IV.	Potential Transformer	I/P 220V-50Hz, O/P 12Vx2, 1000mA, Center Tap	01	100
V.	Current Transformer	SCT-013 1000A/50mA	01	1050
VI.	1.3" OLED Module White	Operating Voltage 3.3V-5V DC, Interface type: IIC interface, Display Color: White, Resolution: 128 x 64 Driver IC: SH1106	01	550
VII.	LED	(Green, 2V, 20mA) x01 (Red, 2V, 20mA) x01	02	8
VIII.	Incandescent Bulb	40W, 220-240V	02	60
IX.	Buck Converter	LM2596 DC-DC	01	98
X.	Resistors	-	-	-
XI.	Capacitors	-	-	-
XII.	Buzzer module	-	-	-
XIII.	Crocodile clip	-	-	-
XIV.	Jumper wire	-	-	-
XV.	Breadboard	-	03	260
XVI.	Push Button	-	-	-

Table-01: Table listed equipment used to implement the project.

Circuit Block Diagram:

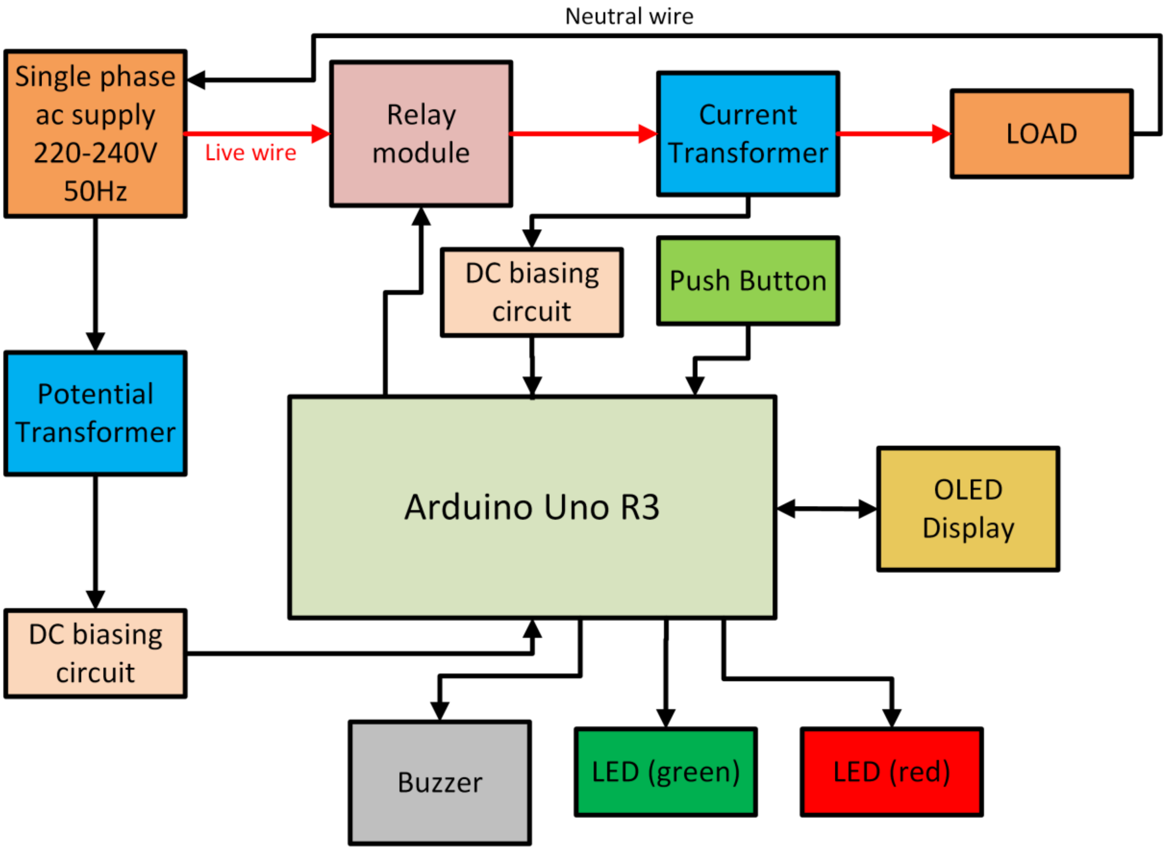

Fig-01: Block of the circuit.

Working principle:

At first when the circuit is powered on load is kept off, when user press on the push button relay connect the load to the main supply. Meanwhile voltage sensor and current sensor continuously feed their signal to the Arduino and Arduino process the data and decide whether voltage and current in given limit. If current or voltage goes out of their limit Arduino immediately disconnect the load from the supply and red led as well as buzzer is turned on.

Current Sensor step down the current from primary to the secondary with ratio of 2000 that means for 100A in primary 50mA flows in the secondary. This ac signal is then brought into 0-5V dc range using the biasing circuit and feed to the Arduino's analog pin. In the case of voltage sensor voltage is step down to 12 V ac when in primary it is 220V ac. Then using same dc biasing technique this range is brought to 0-5V range and feed to Arduino. Principle of the current sensor has been shown in fig-02.

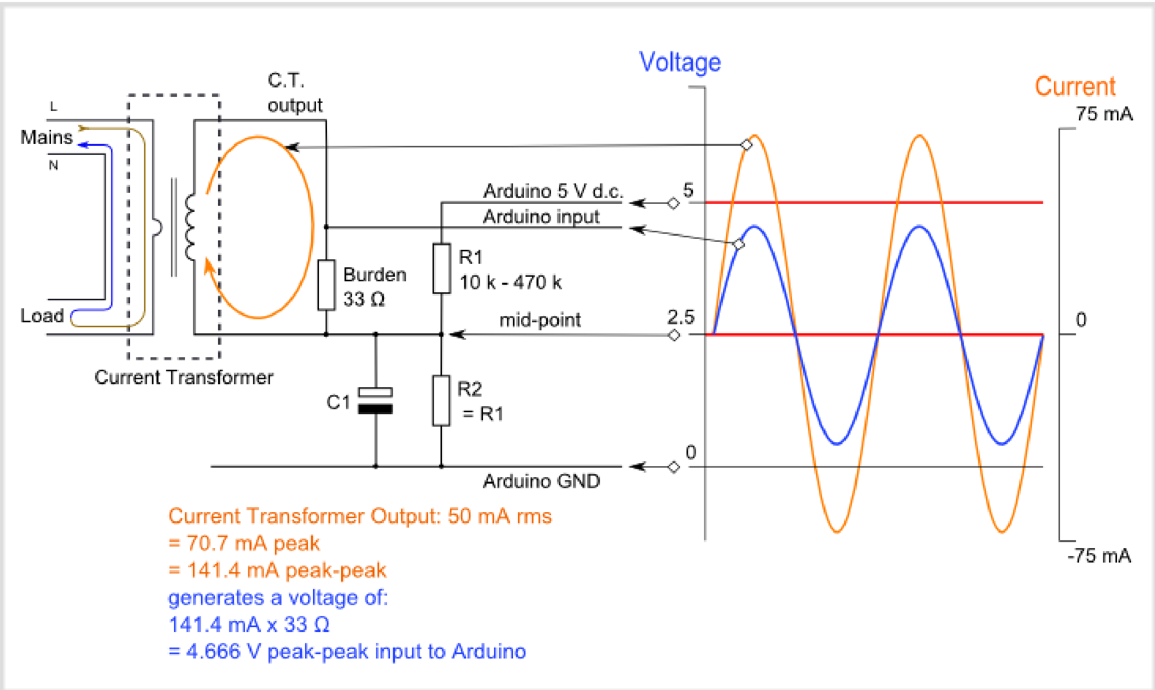

Fig-02: Current sensor circuit diagram and working principle demonstration.

Code:

```
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

#define sample 50 //number of instantaneous measurement in one loop
#define voltage_sensor_pin_num A1 //voltage sensor connected pin number
#define current_sensor_pin_num A2 //current sensor connected pin number
#define freq 50 //operating frequency of the applied voltage
#define voltage_offset 2.49
#define tune_parameter 8.9 //voltmeter
#define current_resistance 480
#define local_relay 8
#define button_01 7
#define green_led 4
#define red_led 3
#define buzzer 5
#define max_current_limit 0.23 //Ampere ac rms
#define max_voltage_limit 240.0 //Voltage ac rms
#define min_voltage_limit 190.0 //Voltage ac rms
#define over_current_stay 100 //milli second
int T_voltage = 1000 * 1000 / freq; //period of applied sine voltage


//double inst_vol, sqre, sum;
//double rms_vol, t1, t2, val;
//float n = 1, avg = 0;

#define SCREEN_WIDTH 128 // OLED display width, in pixels
#define SCREEN_HEIGHT 64 // OLED display height, in pixels

// Declaration for an SSD1306 display connected to I2C (SDA, SCL pins)
#define OLED_RESET     -1 // Reset pin # (or -1 if sharing Arduino reset pin)
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire,
OLED_RESET);

int digitalReadOutputPin(uint8_t pin)
{
  uint8_t bit = digitalPinToBitMask(pin);
  uint8_t port = digitalPinToPort(pin);
```

```
  if (port == NOT_A_PIN)
    return LOW;

  return (*portOutputRegister(port) & bit) ? HIGH : LOW;
}

void setup() {
  // put your setup code here, to run once:
  //analogReference(EXTERNAL);
  pinMode(voltage_sensor_pin_num, INPUT);
  pinMode(current_sensor_pin_num, INPUT);
  pinMode(local_relay, OUTPUT);
  digitalWrite(local_relay, HIGH);
  pinMode(green_led, OUTPUT);
  pinMode(red_led, OUTPUT);
  pinMode(buzzer, OUTPUT);
  digitalWrite(local_relay, HIGH);
  pinMode(button_01, INPUT);
  Serial.begin(9600);

  if (!display.begin(SSD1306_SWITCHCAPVCC, 0x3C))
    Serial.println(F("SSD1306 allocation failed"));

}

bool button_last_state = digitalRead(button_01); //false means switch is turned
off, true means on
bool button_current_state, overcurrent_voltage = false;
void loop() {
  button_current_state = digitalRead(button_01);
  if (button_current_state != button_last_state) {
    button_last_state = button_current_state;
    if (button_current_state == true) {
      digitalWrite(local_relay, !digitalReadOutputPin(local_relay));
      overcurrent_voltage = false;
      digitalWrite(red_led, LOW);
      digitalWrite(buzzer, LOW);
    }
  }
  digitalWrite(green_led, !digitalReadOutputPin(local_relay));
```

```
  power_mea();
  //  delay(500);
}

void power_mea() {
  unsigned int crossCount = 0;
  unsigned long int numberOfSamples = 0;
  float startV, timeout = 50, lastFilteredV, filteredV = 0, filteredI = 0, sampleV,
sampleI, sqV, sumV = 0, sqI, sumI = 0, power, sumP = 0;
  float power_factor, phaseShiftedV;
  boolean lastVCross, checkVCross;
  int crossings = 2, PHASECAL = 1.5;

  // 1) Waits for the waveform to be close to 'zero' (mid-scale adc) part in sin
curve.

  unsigned long start1 = millis();
  while (1)
  {
    startV = analogRead(voltage_sensor_pin_num);
    if ((startV < (1023 * 0.55)) && (startV > (1023 * 0.45))) break; //check its within
range
    if ((millis() - start1) > timeout) break;
  }

  // 2) Main measurement loop

  start1 = millis();

  while ((crossCount < crossings) && ((millis() - start1) < timeout))
  {
    numberOfSamples++;                    //Count number of times looped.
    lastFilteredV = filteredV;            //Used for delay/phase compensation

    //-------------------------------------------------------------------------
    // A) Read in raw voltage and current samples
    //-------------------------------------------------------------------------
    sampleV = analogRead(voltage_sensor_pin_num);                  //Read in raw
voltage signal
```

```
    sampleI = analogRead(current_sensor_pin_num);                    //Read in raw
current signal

    filteredV = (sampleV * 5.0 / 1023.0) - voltage_offset;
    filteredV = (filteredV * tune_parameter) * (223.00 / 14.00);

    filteredI = ((((5.0 * sampleI) / 1023.0) - voltage_offset) * 2000.0) /
current_resistance;
    // C) Root-mean-square method voltage

    sqV = filteredV * filteredV;              //1) square voltage values
    sumV += sqV;                              //2) sum

    sqI = filteredI * filteredI;              //1) square current values
    sumI += sqI;                              //2) sum

    phaseShiftedV = lastFilteredV + PHASECAL * (filteredV - lastFilteredV);
    power = phaseShiftedV * filteredI;
    //accumulating sum of the voltage
    sumP += power;

    lastVCross = checkVCross;
    if (sampleV > startV) checkVCross = true;
    else checkVCross = false;
    if (numberOfSamples == 1) lastVCross = checkVCross;

    if (lastVCross != checkVCross) crossCount++;
  }
  filteredV = sqrt(sumV / numberOfSamples);
  filteredI = sqrt(sumI / numberOfSamples);
  power = sumP / numberOfSamples;
  if(power<0.1){
    power = 0.1;
  }

  if (!digitalReadOutputPin(local_relay)) {
    if (filteredI > max_current_limit) {
      unsigned long overcurrent_time = millis();
      while (1) {
        if ((millis() - overcurrent_time) > over_current_stay) {
```

```
        digitalWrite(local_relay, HIGH);

        display.clearDisplay(); // Clear the Screen
        display.setTextSize(2); // Set text Size
        display.setTextColor(WHITE); // set LCD Colour
        display.setCursor(30, 0); // Set Cursor Position
        display.print("Over_current_detected");
        display.display();

        digitalWrite(red_led, HIGH);
        digitalWrite(buzzer, HIGH);
        overcurrent_voltage = true;

      }
      if (current_mea2() < max_current_limit) {
        break;
      }
    }
  }
}

if (!digitalReadOutputPin(local_relay)) {
  if (filteredV > max_voltage_limit) {
    digitalWrite(local_relay, HIGH);

    display.clearDisplay(); // Clear the Screen
    display.setTextSize(2); // Set text Size
    display.setTextColor(WHITE); // set LCD Colour
    display.setCursor(30, 0); // Set Cursor Position
    display.print("Over_Voltage_detected");
    display.display();

    digitalWrite(red_led, HIGH);
    digitalWrite(buzzer, HIGH);
    overcurrent_voltage = true;
  }

  if (filteredV < min_voltage_limit) {
    digitalWrite(local_relay, HIGH);
```

```
      display.clearDisplay(); // Clear the Screen
      display.setTextSize(2); // Set text Size
      display.setTextColor(WHITE); // set LCD Colour
      display.setCursor(30, 0); // Set Cursor Position
      display.print("Lower_Voltage_detected");
      display.display();

      digitalWrite(red_led, HIGH);
      digitalWrite(buzzer, HIGH);
      overcurrent_voltage = true;
    }
  }

  if (!overcurrent_voltage) {
    display.clearDisplay(); // Clear the Screen
    display.setTextSize(2); // Set text Size
    display.setTextColor(WHITE); // set LCD Colour
    display.setCursor(30, 0); // Set Cursor Position
    display.print(filteredV);
    display.print("V ");// Print the this Text
    display.print(filteredI);
    display.print("A ");
    display.print(power);
    display.print(" W");
    display.display(); // Update the Display
  }
}
```

Real life preview:

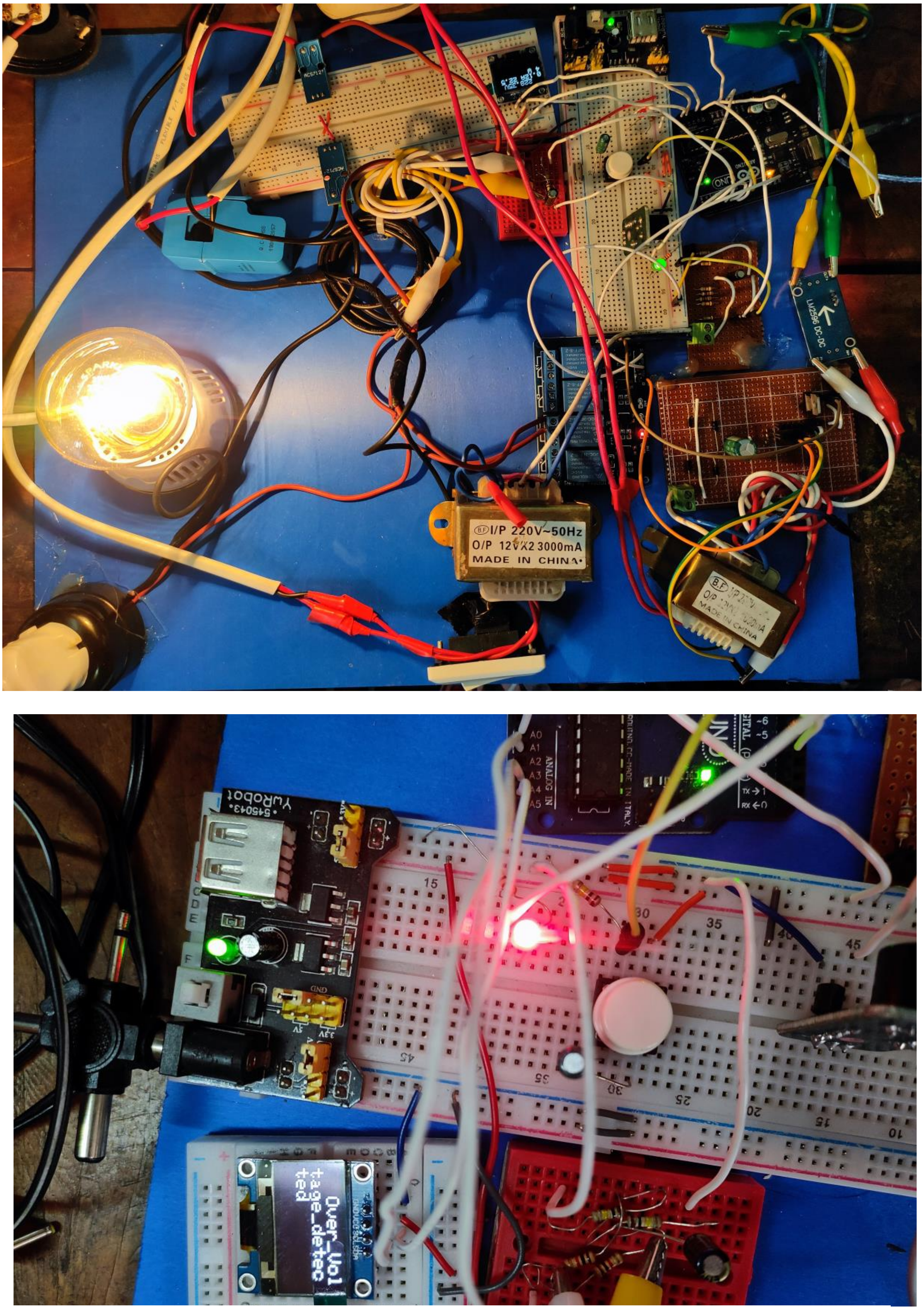

Fig-03: Real life implementation of the project

Feature of the system:

- Overcurrent protection up to 7amp (rms).
- Over voltage protection up to 240V (rms).
- Under voltage protection for voltage less than 220V (rms).
- Single push button to control the load on and off.

Conclusion:

This project can provide safety against overcurrent, overvoltage, undervoltage to household appliances that insure the safety of the expensive equipment. The circuit response very fast with the fault occur in the load or supply and with simple code using same circuit current and voltage rating for protection can also be varied. This project was tested up to 6.5amp with 1300W power delivery and it worked fluently as described and designed. In future IoT can be integrated with the existed circuit and frequency protection can also be added with a minor modification in the code.