“AUTO FAULT DETECTON AND PROTECTION VIA TRIPPING OF DISTRIBUTION LINES AS WELL AS ALERT OVER THE INTERNET”

A

Project Report

Submitted to

JHARKHAND UNIVERSITY OF TECHNOLOGY, RANCHI

In Partial fulfillment of the Requirement for the Award of Degree of

BACHELOR OF TECHNOLOGY

IN

ELECTRICAL ENGINEERING

Submitted by

RAHUL KUMAR SAW	**18030450061**
PRABHAT AGARWAL	**18030455007**
VISHAL KUMBHKAR	**18030450100**
RAJU KUMAR YADAV	**18030450062**

Under the guidance of

Dr. K. SARWAGYA

Assistant Professor Electrical Engineering
BIT ,Sindri

DEPARTMENT OF ELECTRICAL ENGINEERINGBIT SINDRI, DHANBAD-828123 (JHARKHAND)

JUNE 2022

BIT SINDRI, DHANBAD-828123

Department of Higher & Technical Education, Government of Jharkhand

CANDIDATE'S DECLARATION

We hereby certify that the dissertation entitled "**AUTO FAULT DETECTON AND PROTECTION VIA TRIPPING OF DISTRIBUTION LINES AS WELL AS ALERT OVER THE INTERNET**" which is being submitted in the partial fulfillment of the requirement for the award of **Bachelor of Technology in Electrical Engineering**, is a record of our own work carried out under the supervision and guidance of "Dr. K. Sarwagya " , Assistant Professor, Department of Electrical Engineering, BIT Sindri.

All information in this document has been obtained and presented as per academic rules and ethical conduct.

To the best of my knowledge, the contents presented in this project report have not been submitted elsewhere for the award of any other degree/diploma.

Place - Jharkhand

Date - June 2022

Rahul Kumar Saw
(18030450061)

Prabhat Agarwal
(18030450071)

Vishal Kumbhkar
(18030450074)

Raju Kumar Yadav
(18030450093)

CERTIFICATE OF DECLARATION FROM THE SUPERVISOR

This is to certify that the work presented in the dissertation entitled "**AUTO FAULT DETECTON AND PROTECTION VIA TRIPPING OF DISTRIBUTION LINES AS WELL AS ALERT OVER THE INTERNET**" in partial fulfillment of the requirement for the award of Degree of **Bachelor of Technology in Electrical Engineering at BIT Sindri** is authentic work carried out under my supervision and guidance.

To the best of my knowledge, the content of this dissertation does not form a basis for the award of any previous Degree to anyone else.

Date:- June 2022

Signature of supervisor
Name : Dr. K. Sarwagya
Designation: Assistant Professor
Department Name: Electrical Engineering

CERTIFICATE OF APPROVAL FROM THE EXAMINERS

The foregoing dissertation entitled "**AUTO FAULT DETECTON AND PROTECTION VIA TRIPPING OF DISTRIBUTION LINES AS WELL AS ALERT OVER THE INTERNET**", is hereby approved as a creditable study of the research topic and has been presented satisfactorily to warrant its acceptance as a prerequisite to the degree for **Bachelor of Technology in Electrical Engineering at BIT Sindri**.

It is understood that by this approval, the undersigned do not necessarily endorse any conclusion drawn or opinion expressed therein, but approve the report for the purpose for which it is submitted.

Signature of Internal Examiner	Signature of External Examiner
Name:	Name :
Designation :	Designation :
Name of Department :	Name of Department :
Name of the Institute :	Name of the Institute/Organization :

Signature of HOD
Name: Dr. Md Abul Kalam
Head, Electrical Engineering

ACKNOWLEDGEMENT

We would like to articulate our deep gratitude to our project guide **Dr. K. Sarwagya**, Assistant Professor Department of Electrical Engineering who have always been source of motivation and firm support for carrying out the project. We also express our gratitude to him for his invaluable suggestion and constant encouragement allthrough the project work.

We would also like to convey our sincerest gratitude and indebtednessto all other faculty members and staff of Department of Electrical Engineering, BIT Sindri, who bestowed their great effort and guidanceat appropriate times without it, would have been very difficult on our project work.

An assemblage of this nature could have been attempted with our reference to and inspiration from the works of others whose details arementioned in references section. We acknowledge our indebtedness toall of them. Further, we would like to express our feeling towards our parents and god who directly or indirectly encouraged and motivated usduring this dissertation.

INDEX

TITLE	PAGE NO
Declaration	II
Certificate of Approval	IV
Acknowledgement	V
Index	VI
Abstract	VIII
List of Figures	X
List of tables	XII
CHAPTER 1 : INTRODUCTION	1
1.1 Overview	1
1.2 Objective of Report	2
CHAPTER 2 : COMPREHENSIVE SURVEY	3
2.1 Literature Review	3
2.2 Fault	4
2.3 The Causes of fault in there phase devices	5
CHAPTER 3 : METHODOLOGY OF THE WORK	6
3.1 Analysis of Project	6
3.1.1 Required Components	6
3.1.2 Block Diagram	10
3.1.3 Circuit Diagram	11
3.2 Schematic Representation of Work	12
3.2.1 Working Procedure	12
3.2.2 Required Software	12
3.3 Hardware Circuit Description	14
3.3.1 Power Supply Unit	14
3.3.2 Fault Creating Unit	14
3.3.3 Arduino Operating Unit	15
3.3.4 Display Unit	16
3.3.5 Relay Operating Unit	17
3.3.6 Flame Sensor Unit	18

3.3.7 Wireless Transmission Unit 18
3.3.8 Assembled Model 19

CHAPTER 4 : RESULTS AND DISCUSSION 20

4.1 Observation Table and Status 21

4.2 Results 22

CHAPTER 5 : CONCLUSION AND SCOPE OF THE FUTURE WORK 22

5.1 Conclusion 22

5.2 Scope of the future work 22

REFERENCES 23

APPENDIX 1 Arduino Code 24

APPENDIX 2 ESP8266 code 29

APPENDIX 3 Pin Diagram of Arduino 31

APPENDIX 4 Pin Diagram of Node MCU 33

ABSTRACT

In Electrical Power System, there are High Voltage Transmission Lines between generating station and substation as well as Low Voltage Distribution Lines between substation and load. There are a variety of faults like Line to Ground faults(LG), Line to Line faults(LL), Line to Line to Ground fault, etc. that can occur in these lines which causes a huge amount of current flow through these lines then it may cause heavy damage to the equipment that are connected throughout the Power System. Moreover, it may lead to public hazards.

Already there are protection systems which trip the lines during faulty situation, but this Project is designed in such a way that not only it automatically trips the Lines during fault but also alerts us about the fault in any place around the world as this communication will be over internet connection.We can get the location, current parameters and status of the faulty condition and remotely control various apparatus through internet. Also, the system automatically resets itself in case of temporary fault. Since it would be highly risky and dangerous to work directly on high voltage lines, the system model would use 220V AC domestic line. The system will consist of three single phase transformers which converts 220V into 12V which would be safe for small scale model. The output of three transformers would be configured as star connected 3-phase Line. Switches would be used to create manual faults and Arduino along with IoT would used for control and communication purpose. There would be relays acting as circuit breakers for tripping the line. Arduino would turn ON the supply after a short time interval in case of temporary fault and trips permanently otherwise. There would be LCD display which would show current parameters and a Wi-Fi module for common.

LIST OF FIGURES

Fig 1.1 Transmission Tower.....…….…….…03

Fig 2.1 Fault Creating Unit……...…..……..04

Fig 3.1 Schematic representation of work…….……..……..…..06

Fig 3.2 Arduino UNO…….....……...…..07

Fig 3.3 Push Button……..……..……07

Fig 3.4 Resistor ……...07

Fig 3.5 LCD Display…….……..……07

Fig 3.6 Relay…….……..07

Fig 3.7 Transformer…….08

Fig 3.8 Flame Sensor……..……..08

Fig 3.9 Capacitor…….08

Fig 3.10 Buzzer……08

Fig 3.11 Diode……...……..09

Fig 3.12 Zener Diode……...09

Fig 3.13 Block Diagram……..10

Fig 3.14 Circuit Diagram…… ……11

Fig 3.15 Proteus Software……..…13

Fig 3.16 Arduino IDE ……..…….……..13

Fig 3.17 Power Supply Unit……..…..……..14

Fig 3.18 Fault Creating Unit……15

Fig 3.19 Arduino Operating Unit…….……15

Fig 3.20 LCD 1……..…….……16

Fig 3.21 LCD 2……..……..16

Fig 3.22 Relay 1……..……..…17

Fig 3.23 Relay 2…….....…..……..17

Fig 3.24 Flame Sensor unit……..……...…..…18

Fig 3.25 Assembled model…….…….....19

Fig 4.1 Result …….……21

Fig AP.1 Arduino Pin Diagram……31

Fig AP.1 Node MCU ESP8266 Pin Diagram……...…33

LIST OF TABLES

Table 3.1 Components Ratings…….…….09

Table 4.1 Observation Table line fault status……20

Table 4.2 Observation Table Voltage Range……20

CHAPTER 1

INTRODUCTION

1.1 Overview

Fig 1.1

In Power System, the Transmission Lines provides a path for the electrical power to flow from generating station to the distribution unit. The generating station steps up the voltage of produced electricity using step-up transformer for transmission, Transmission lines carry these high voltages and then Distribution unit steps down voltage using step-down transformer and finally distribution lines supply power to the load.

In the overall electrical power system, more than 80% of transmission failure occurs. In this project the design and implementation of fault detection, classification and protection techniques of transmission lines are present.

1.2 Objective of Report

The main objectives of this project work are:

- To automatically detect fault in different phases and find the exact location of fault as well as phase

- To detect Fire due to fault at all data collection point in 3-phase distribution line and alert using buzzer.

- To detect fault due to line break in any phase and trip the line within seconds.

- To trip the line in any fault and display the information in LCD as well as to the authority over the internet

CHAPTER 2

COMPREHENSIVE SURVEY

2.1 Literature Review

During our previous semesters while going through industrial training, under BCCL substation we are exposed to a condition that really needs attention and improvement. In case of fault in the line, it takes time to find its location and there is more power cut as each time it needs to be manually rectified. It can be solved by IoT and Automation.

Many Journals and Research Papers have been published in this matter. In Open Access International Journal of Science and Engineering, Rishikesh Vadje, Punam Kshirsagar and others from SKN-Sinhgad Institute of Engineering and Technology proposed a model which informs the authority about fault location through Internet. In a research article of Journal of Engineering Research and Application, Dr. R.K. Dhatrak used ESP12 Based (ESP8266) advanced micro-controller to detect fault location in distribution line. In International Research Journal of Engineering and Technology (IRJET), Krushna Nikam and others said that there are many components in Power System and proposed a model where we can find the distance of fault from the transformer.

After familiarizing with scenario and techniques of Power System we have proposed a model based on above papers. This system proposed gives the continuous monitoring with the memory involved to save the data for present and future scope. Furthermore, when any fault occurs and the fault is not sustained then the system auto disconnects the faulty section with alarming facility and after a time of relaxation system auto recloses itself creating the self-healing capability and maintain reliability. If the fault is sustained then the system does not reclose itself and seeks for the manual support by informing the authority about fault location through message or mobile notification using WIFI Module.

2.2 Fault

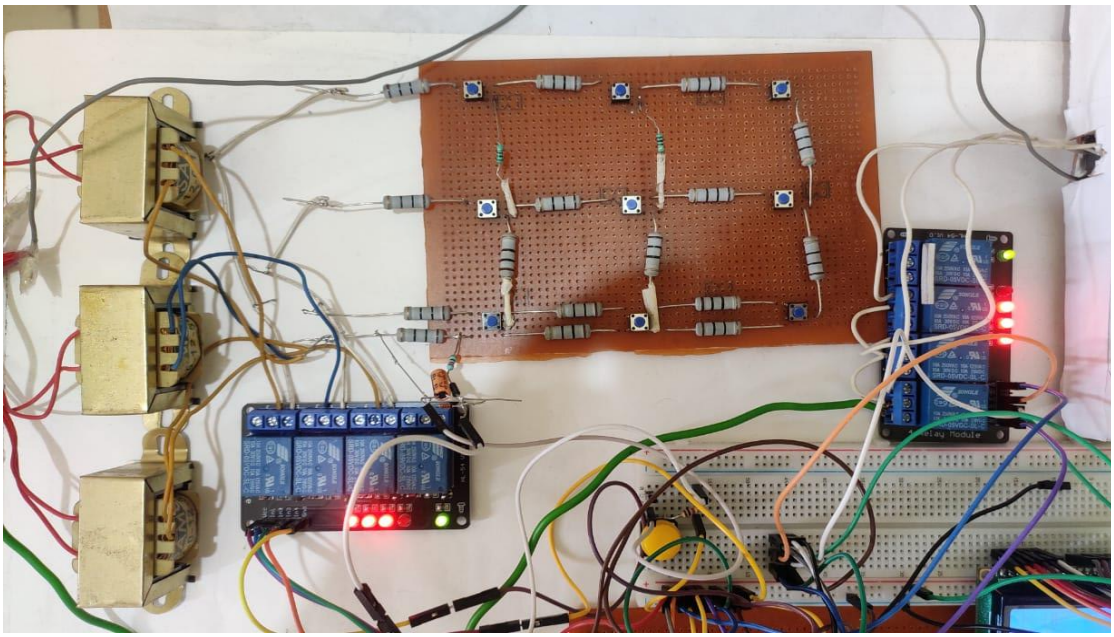


Fig 2.1

A fault, also known as a fault current, is an irregular electric current that passes along a line in an electrical power system. A fault in a three-phase system can occur between one or more phases and ground, or it can occur only between phases. Faults can be divided into two categories:

1.Balanced fault: This fault is also known as a symmetric fault. It has an equal impact on all three phases of the transmission line. In total transmission line faults, approximately 5% of faults are symmetric.
2.Unbalanced faults: It is an example of an asymmetric fault. Asymmetric faults have no impact on all three phases of the transmission line.

Unbalanced fault is further subdivided into three types:

Line-to-line fault: This fault happens when two lines come into direct contact with each other, resulting in a short circuit. Between them. Asymmetric L-L faults account for around 5% of all faults.
Line to ground fault: A short circuit between one line and the ground causes this form of fault. This occurs as a result of physical contact between the line and the ground conductor. as a result of storm damage and lightning, among other things. In transmission line faults,

this is the most common fault.

Double line-to-ground fault: A double line-to- ground fault occurs when two lines come into direct contact with each other and the ground conductor. This type of fault occurs 15-20% of the time.

A double line-to-ground: fault occurs when two lines come into direct contact with each other and the ground conductor. This type of fault occurs 15-20% of the time.

2.3 The causes of fault in a three-phase system: -

Overvoltage: When the device voltage exceeds 110 percent of the nominal voltage rating, it is called an overvoltage. Overvoltage can be triggered by a variety of factors, including abrupt load reductions, load switching, lightning strikes, failure of control equipment such as voltage regulators, and neutral displacement. The part connected to the supply can be damaged by this point of overvoltage.

Under-voltage: When the voltage supplying the drive is too low, this point arises. Under-voltage happens when the input voltage is less than the nominal voltage rating, such as when a 440 V system is operated by 220 V.

Overheating: Overheating occurs when the temperature of the equipment exceeds the nominal values. When the equipment is overloaded, it can overheat. Short circuits, single line to ground faults, L-L faults, and other issues may cause overheating. It has the potential to harm machinery windings as well as the electrical system.

CHAPTER 3

METHODOLOGY OF THE WORK

3.1 Analysis of Project

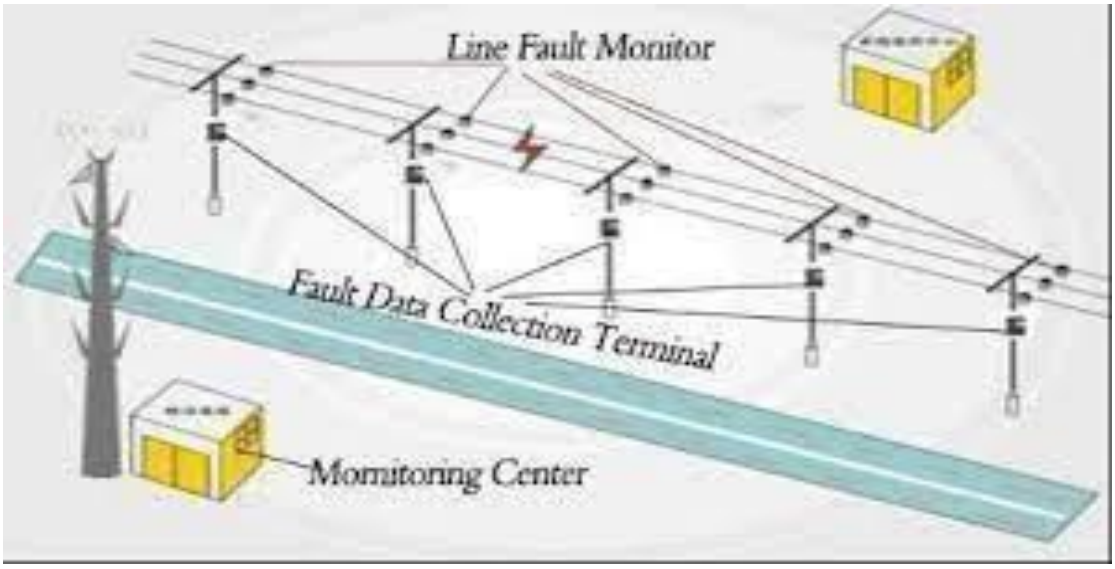


Fig 3.1

3.1.1 Required Components :-

- Arduino Uno
- Push Buttons
- 16*2 lcd display
- Step Down Transformer
- Relay
- Resistors
- Flame sensor
- Capacitor
- Buzzer
- Diode
- Zener diode (1N5918B)

Arduino UNO

Arduino Uno is a microcontroller board based on the ATmega328P. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button.
It can be programmed as per required by the user and we can get desired facilities or functions through the input and output pins.

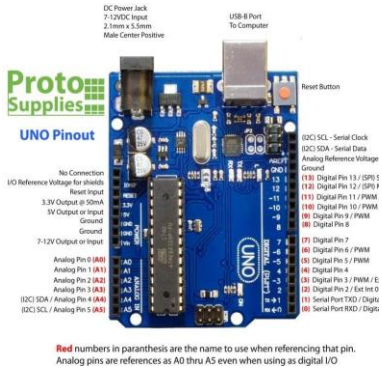


Fig 3.2

Push Button

There are four pins of these buttons which are shorted internally in pairs. When it is pressed, it establishes connection between those pairs as long as button is pressed and disconnects as soon as it is released.

Fig 3.3

Resistor

It is current limiting element as per the color code it is 10 ohm which is used in the project.

FIG 4.4

16*2 LCD Display

It is an alphanumeric display with 2 rows and 16 columns. In this LCD there are 16 pin connections as displayed above. Among these are positive pins, ground pins, brightness adjustment pin (V0), read/write, enable pin and 8 data pins.

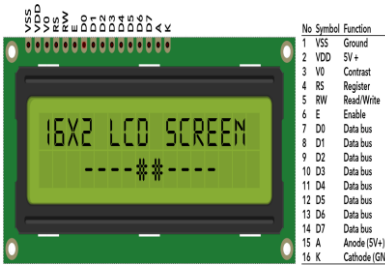


FIG 3.5

Relay

It is a switching device triggered by a signal to open/close a circuit. It has five terminals. Terminal 3 is

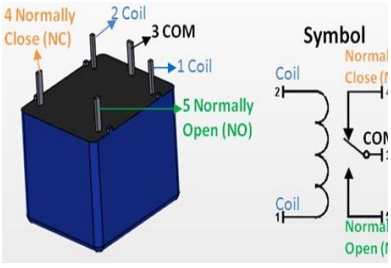


FIG 3.6

common and without triggering it is connected to terminal 4 called Normally Close (NC) and not connected to terminal 5 called Normally Open (NO). When triggering signal flows through the coil using pins 1 and 2, the coil becomes electromagnet and shifts movable contact from NC to NO as long as signal flows through the coil.

Transformer

Transformer transfers electrical energy between two electrically isolated circuits without a change in frequency. Here the Input Side is of 220V 50Hz and Output Side is of 12V 50Hz. Since it decreases the voltage it is called step down transformer.

FIG 3.7

Flame Sensor

Flame sensor works on the principle that when something burns it emits IR rays and these IR signals are sensed by the photodiode or IR sensor and provides digital output signal through output pin. It operates on voltage between 3V – 5.5V

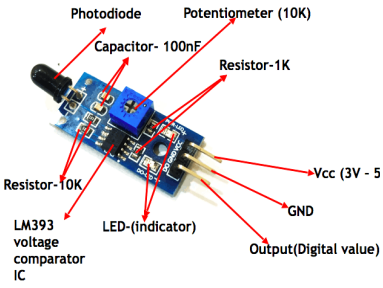


FIG 3.8

Capacitor

It is an electrical element that stores energy in the form of electrostatic charges. It resists a change in voltage applied across it. Its Capacitance is 1 uF.

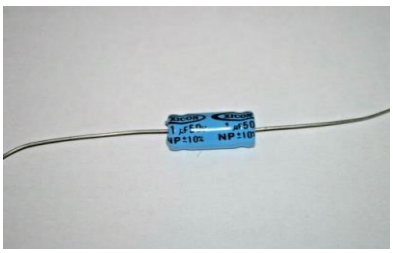


FIG 3.9

Buzzer

It is an electrical device that converts electrical energy into sound energy. The longer terminal in + side is positive terminal and other shorter one is negative terminal. For the project buzzer with 5V operating

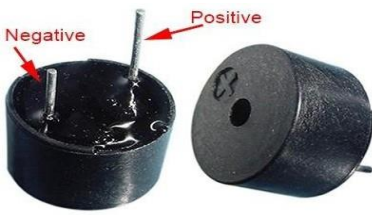


FIG 3.10

voltage is used.

Diode

It is an unidirectional device with two terminals. Anode is the p-side and cathode is the n-side. When p-side is connected to positive voltage compared to the n-side, then it is forward biased and conducts current from p-side to n-side. When polarity is reversed, it is called reverse biased and it offers very high resistance and acts as open circuit

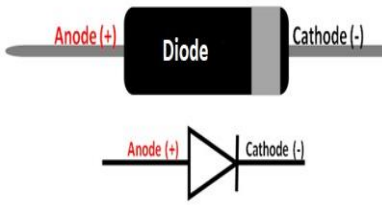


FIG 3.11

Zener Diode

It is also a p-n junction diode but it operates in reverse biased condition. It maintains a constant voltage across its terminals and hence used as a voltage regulator. In our project the Zener diode used is 1N5918B. Its operating voltage is 5.1V and it is of 1.5W.

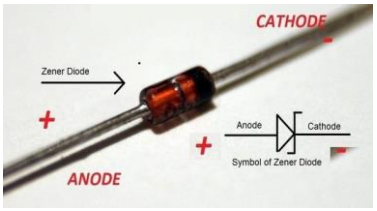


FIG 3.12

Component Rating

COMPONENT RATINGS	PARAMETERS
Diode	Si-1N4007
Capacitor	1 uf
Resistor	10ohm , 100ohm , 220ohm
LCD Display	5V
4-Channel Relay	5V
Buzzer	5V
PCB	6 X 4
Jumper Wires	20cm , 10cm , 2.54mm(mil)
Potentiometer	10K
Zener Diode	4.7V
Step Down Transformer	0-12 12V , 500mA
Flame sensor	5V

Table 3.1 Component Ratings / Parameters

3.1.2 Block Diagram

AC supply is stepped down to up to 12V using three step down transformers and then through relays in each phase, power is supplied to fault creating unit. Manual fault is created here and then the fault signal to sent to Arduino after being converted to DC by using rectifier. Arduino then processes the fault signal and display the fault status of each phase and distance of fault in LCD display. It also sends the tripping signal to the relay to trip the line and protect it from further damage.

Arduino also reads signal from flame sensor module and alerts using buzzer in case of fire. Also, it sends tripping signal to trip all the three phases.

Arduino sends information to the NodeMCU using Serial Communication which then sends the data to the authority using internet.

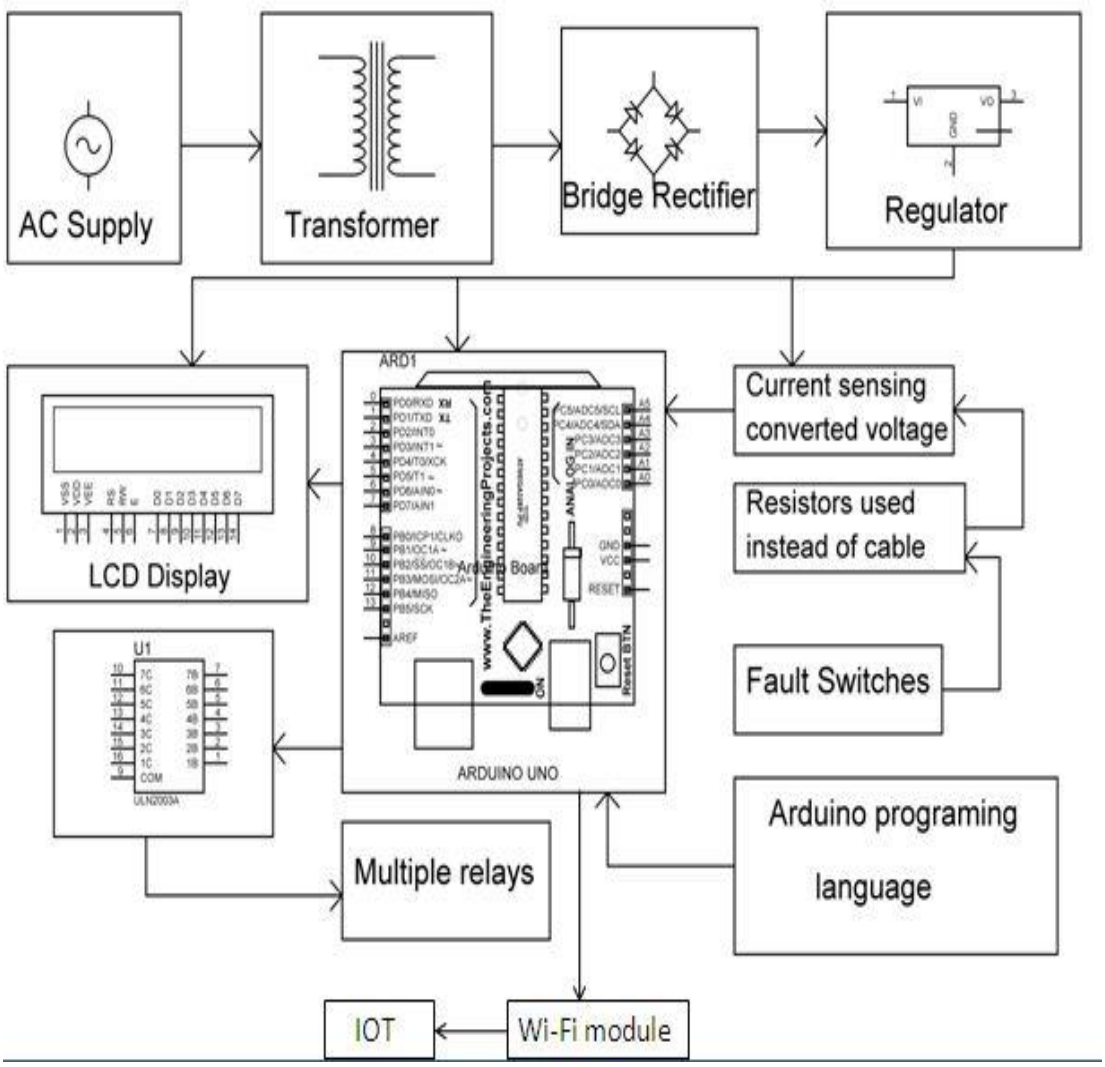


FIG 3.13 Block Diagram

3.1.3 Circuit Diagram

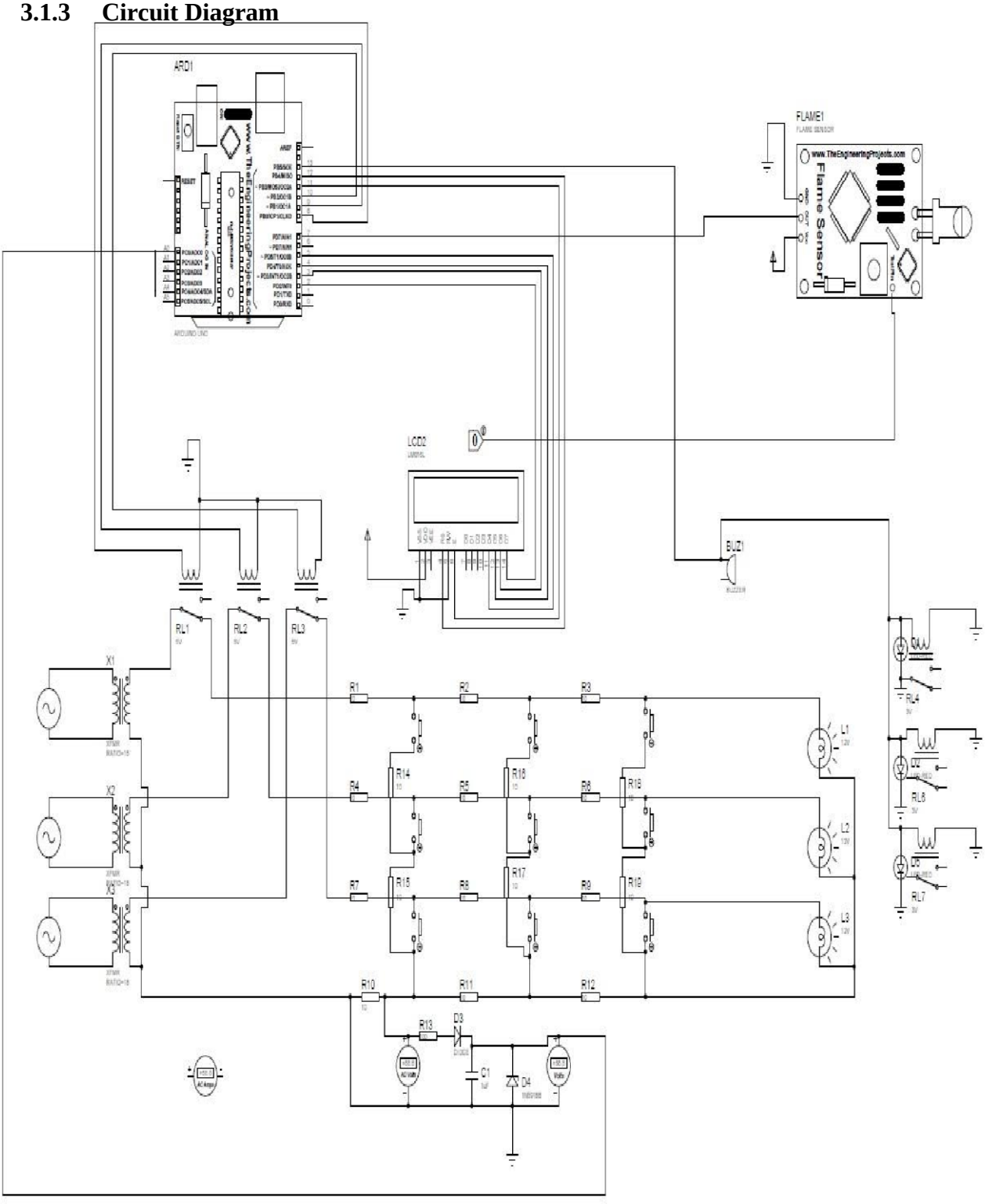


Fig 3.14 Circuit Diagram

3.2 Schematic Representation of Work

3.1.1 Working Procedure

1.Three parallel division RYB of single 220V AC Source and Transformer converts these into 12V.

2.Star Connection in secondary of Transformer with each live wire through NC of relay.

i. Lumped resistance of 10 ohm in each line divided into 3 sections each.

ii. Lamps are connected as load.

iii. Switches for manual fault.

3. Rectifier to convert AC to DC for Arduino Analog input.

4. Zener diode to limit maximum voltage of 5V as required by Arduino.

5. 16*2 LCD Display and previous relays connected to digital Output pins of Arduino.

6. LCD to see current status of Lines and Relays to trip in case of fault.

7. Flame Sensor is connected to digital input of Arduino.

8. Another 3 relays and a buzzer are connected to digital output pins of Arduino.

9. In case of fire trips all three lines.

3.1.2 Required Software

Proteus Professional 8.11

1. Proteus is used to simulate, design and drawing of electronic circuits. It was invented by the Lab center electronic.

2. By using proteus we can make two-dimensional circuits designs as well.

3. There are numerous benefits to simulate circuits on proteus before make them practically.

4. Designing of circuits on the proteus takes less time than practical construction of the circuit.

5. Using proteus we can find different parents of circuits such as current, a voltage value of any component and resistance at any instant which is very difficult in a practical circuit.

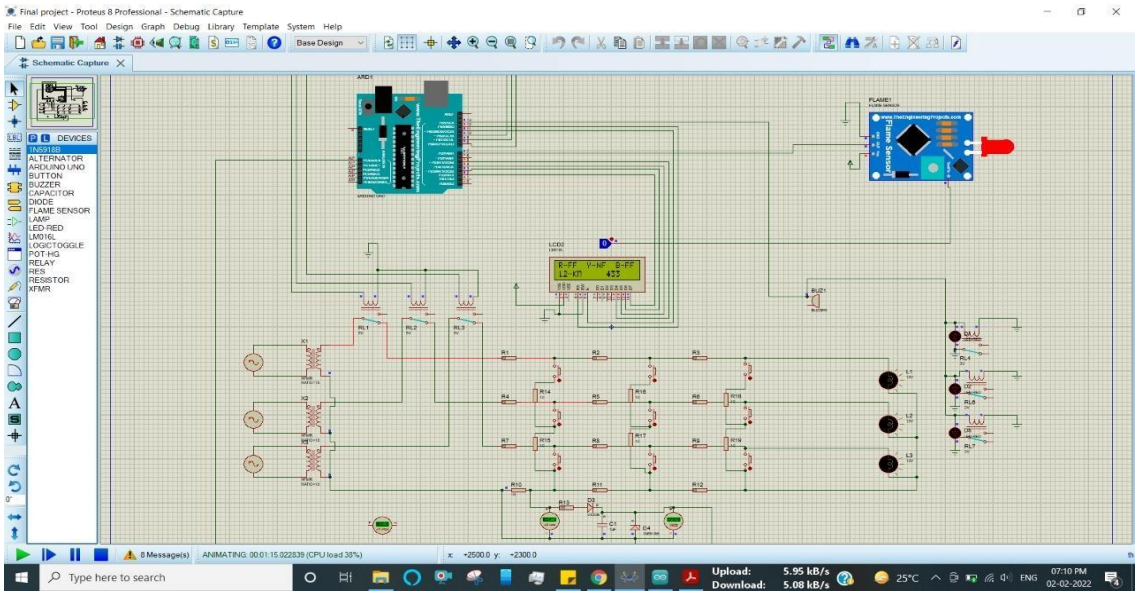


FIG 3.15 PROTESUS SOFTWARE

Arduino Integrated Development Environment (IDE)

1. Arduino IDE stands for Arduino Integrated Development Environment. It is the software for Arduino.
2. It is a text editor like a notepad with number of features.
3. It is used for writing codes, compiling the codes to check if there is any errors and then uploading the code to the Arduino.
4. It is used to connect the Arduino hardware to upload programs and communicate with them.

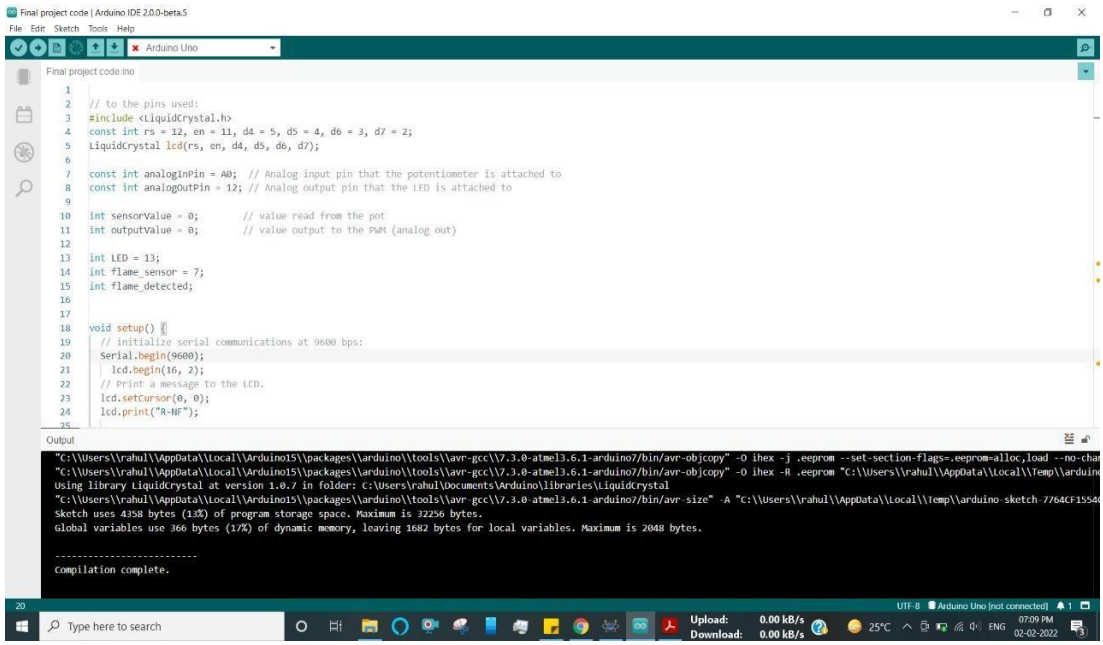


FIG 3.16 Arduino IDE

3.3 Hardware Circuit Description

3.3.1 Power Supply unit

1. Single Phase is distributed parallelly in 3-phases to create 3-phase distribution line.
2. 3-Step Down Transformer converts 230v AC to 12v AC which is safer to work on.
3. Secondary side of Transformers are connected as Star Connection which are connected to fault creating circuit through relay in each phases.
4. External 5v DC supply is provided to Arduino and relay operating units.

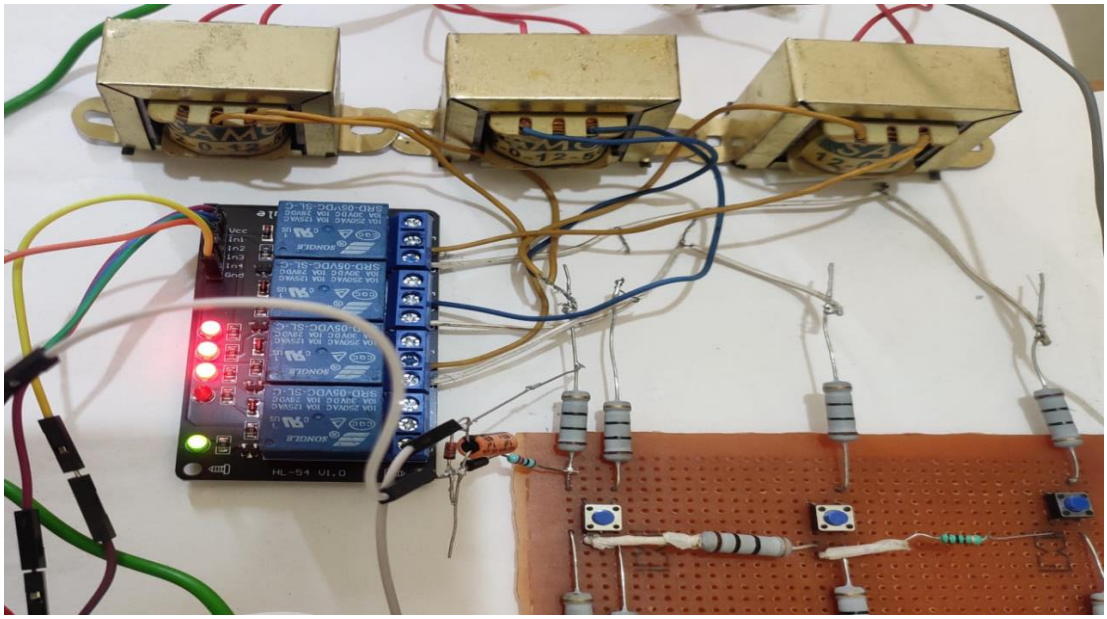


FIG 3.17 Power Supply Unit

3.3.2 Fault Creating Unit

1. 4 rows of resistors are created to demonstrate the distributed resistances of specific distance offered by 3 phases R, Y, B and neutral.
2. Push buttons are used to manually create the fault in the lines
3. Also, resistances are inserted between adjacent lines to distinguish fault in different phases.
4. Each resistance in rows is 10ohm and resistance between adjacent rows are 10 and 100ohms.
5. Rectifier circuit is used to convert AC voltage across a resistance in neutral line into DC so that Arduino can read from it. It demonstrates the distance relay in practical world.

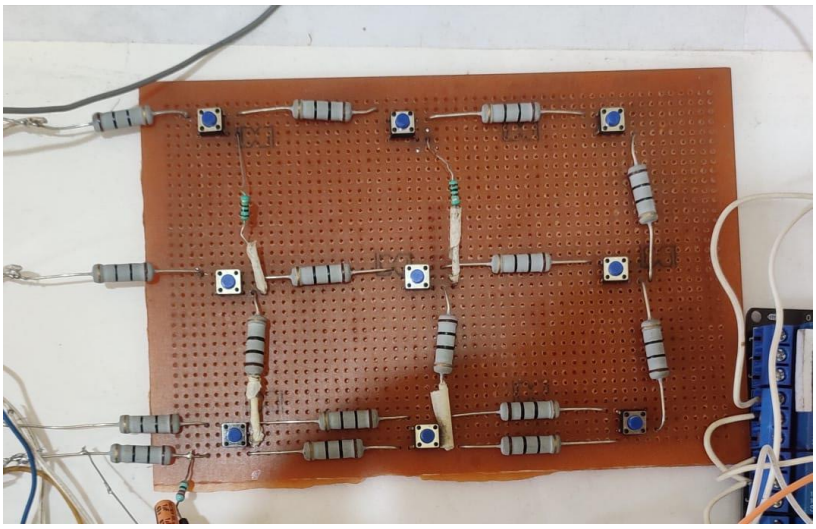


FIG 3.18 Fault Creating unit

4.3.3 Arduino Operating Unit

Arduino is the main component of our project. It's function include the following

1. Read DC voltage from fault creating circuit which is different when different push buttons are pressed and display the fault in display and fault distance in KM.
2. It sends signal to relay operating unit to trip the line in case of fault.
3. It receives signal from flame sensor unit and alert using buzzer.
4. It communicates with wireless transmission unit to transfer the fault status data via internet.

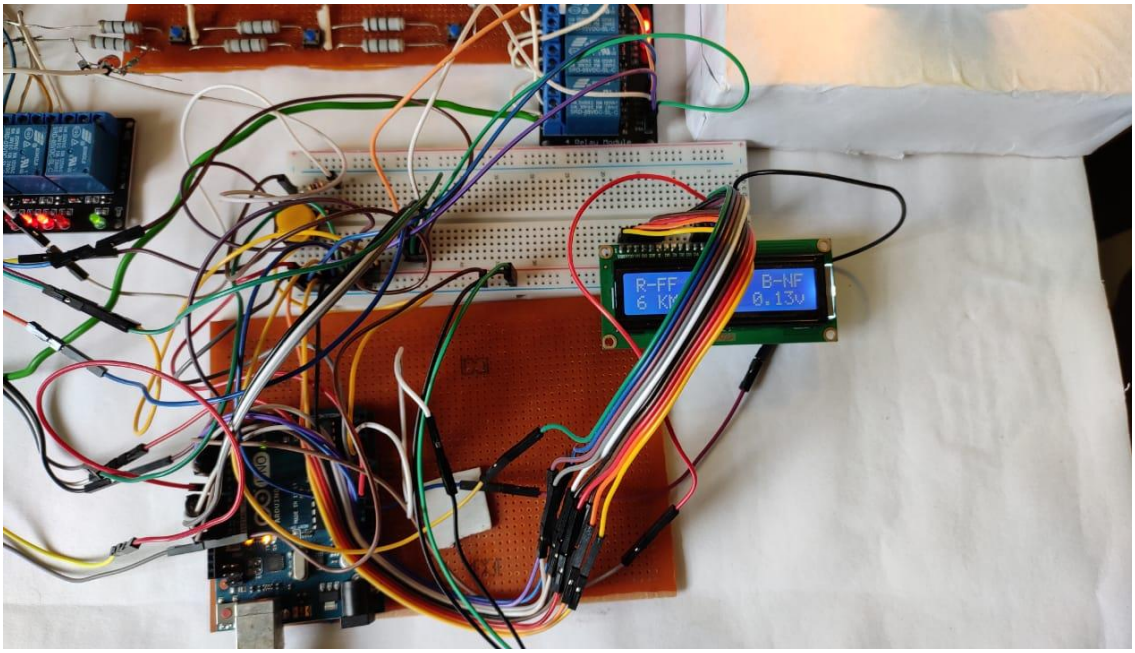


FIG 3.19 Arduino Operating Unit

4.3.4 Display Unit

1. When the fault occurred in any of the phases then the following information are displayed on the 16 X 2 LCD display.
    a. Fault status (in which phase fault has occurred).
    b. Distance from the source to the fault.
    c. DC voltage from creating unit (through rectifier).
2. NF- shows no fault, FF- shows Fault in the phase.
3. In case of fire detection, it displays “Fire Fire Fire”.

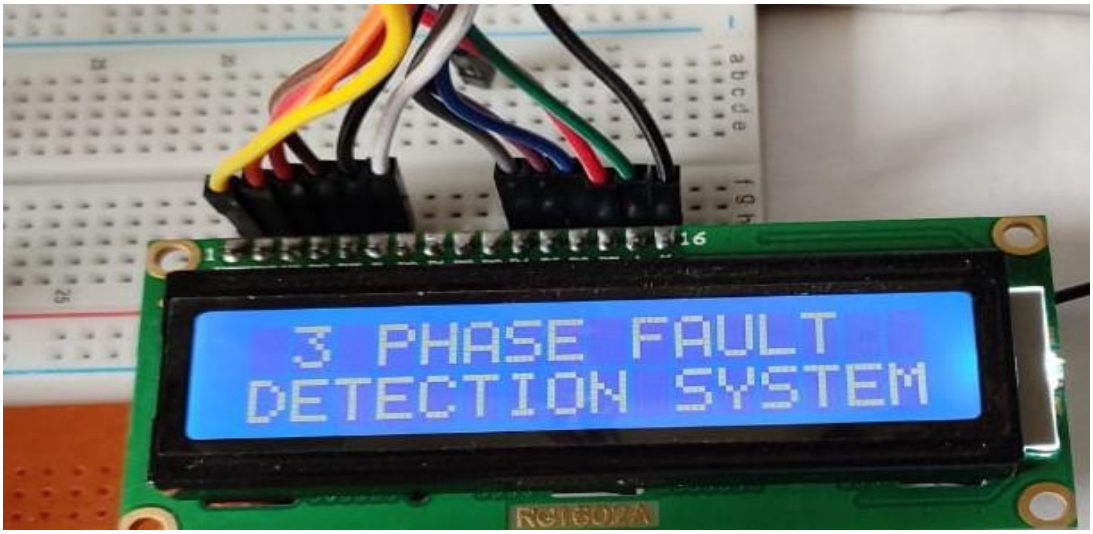


FIG 3.20 LCD 1

FIG 3.21 LCD 2

4.3.5 Relay operating Unit

1. Three relays are connected through the three phases R, Y, B to trip the line in case of fault or fire.
2. Each phase is connected to the NC of relays and tripping signal pins are connected to the Arduino.
3. Another three relays are connected to the three bulbs representing the loads of three phases.

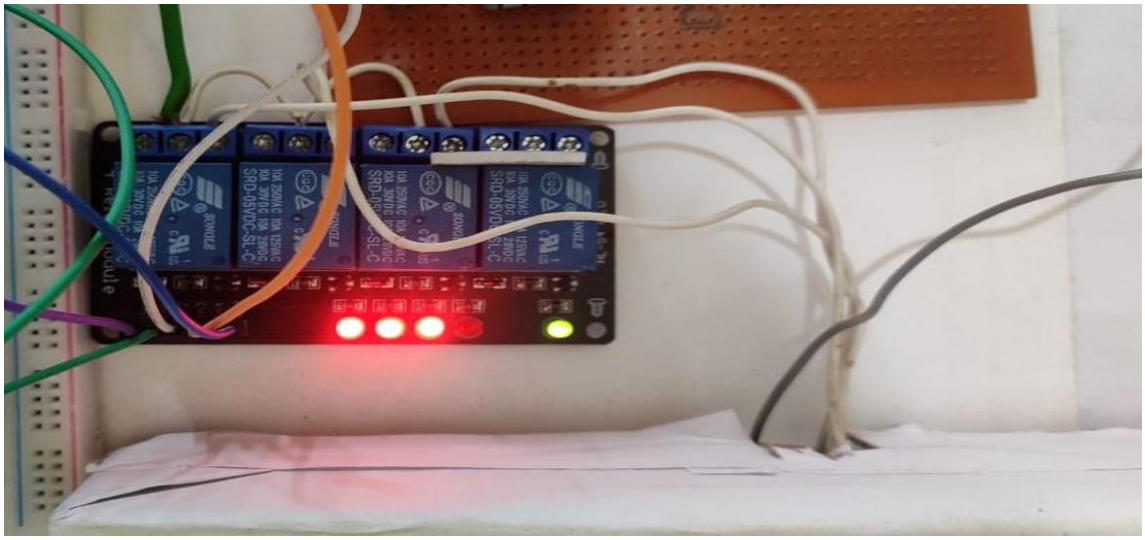


FIG 3.22 Relay 1

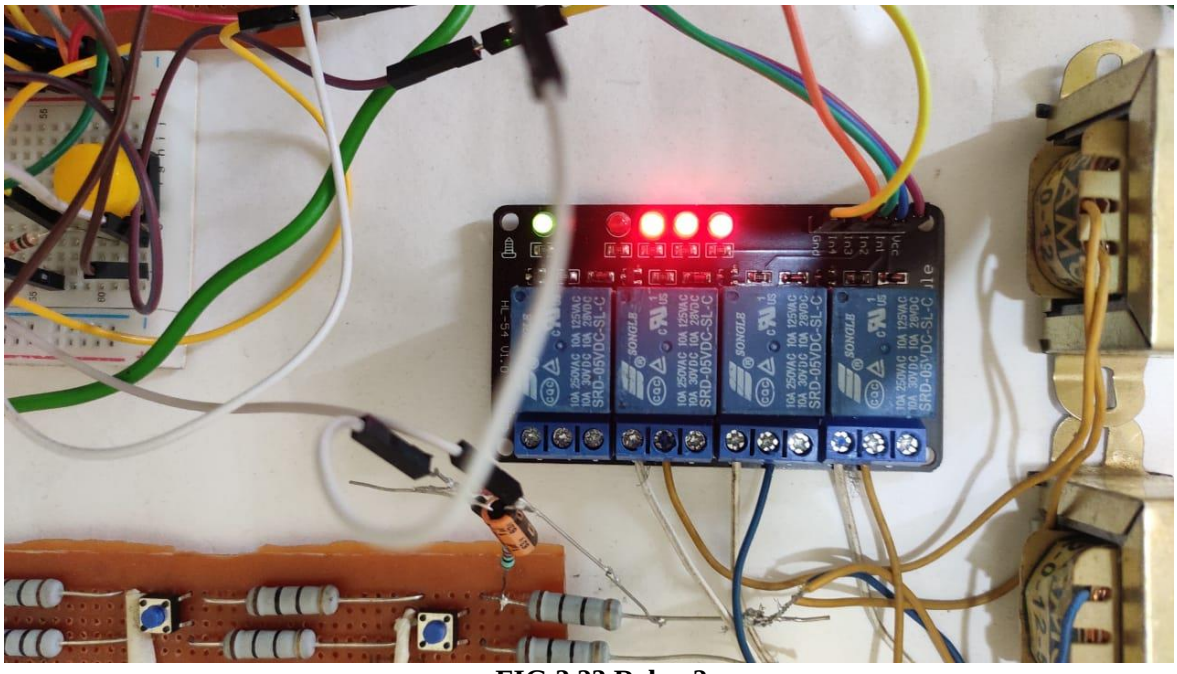


FIG 3.23 Relay 2

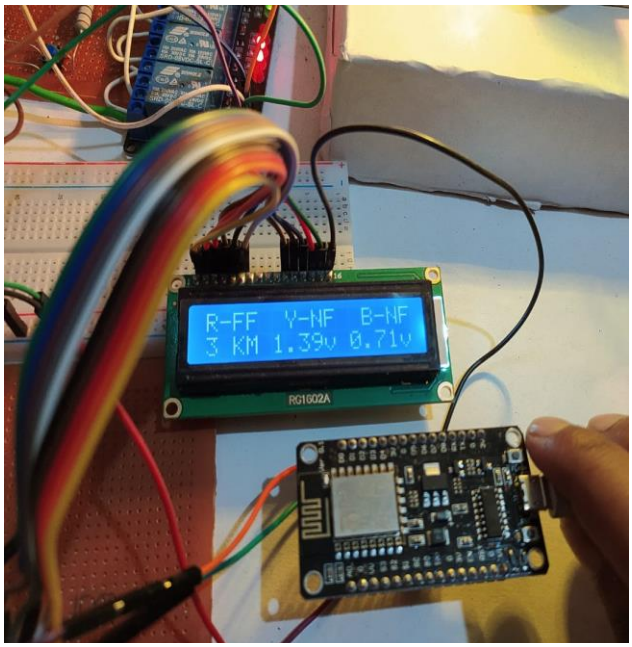
4.3.6 Flame Sensor Unit

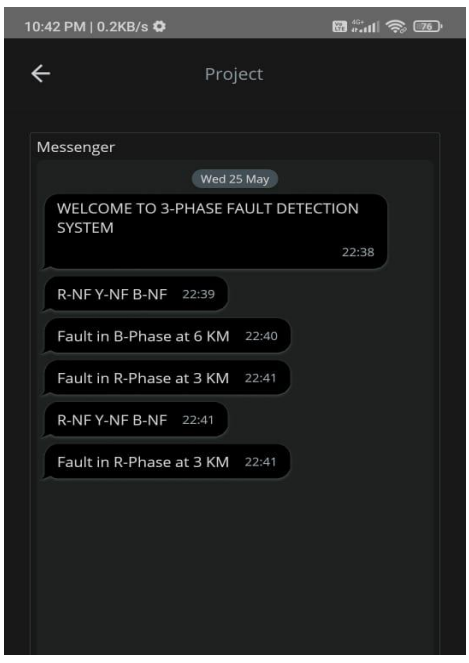
When the fault occurred in the system high value of current will flow as a result temperature of the lines increases and fire occurred and this will be sensed by the flame sensor. we would be alerted by the buzzer and simultaneously relay would be tripped.

FIG 3.24 Flame Sensor

4.3.7 Wireless Transmission unit

1. NodeMCU board which contains ESP8266 WIFI Module is used for wireless data transmission.
2. Serial Communication is set up between Arduino operating unit and NodeMCU and then data is transmitted to the cloud from NodeMCU through internet.





4.3.8 Assembled Model

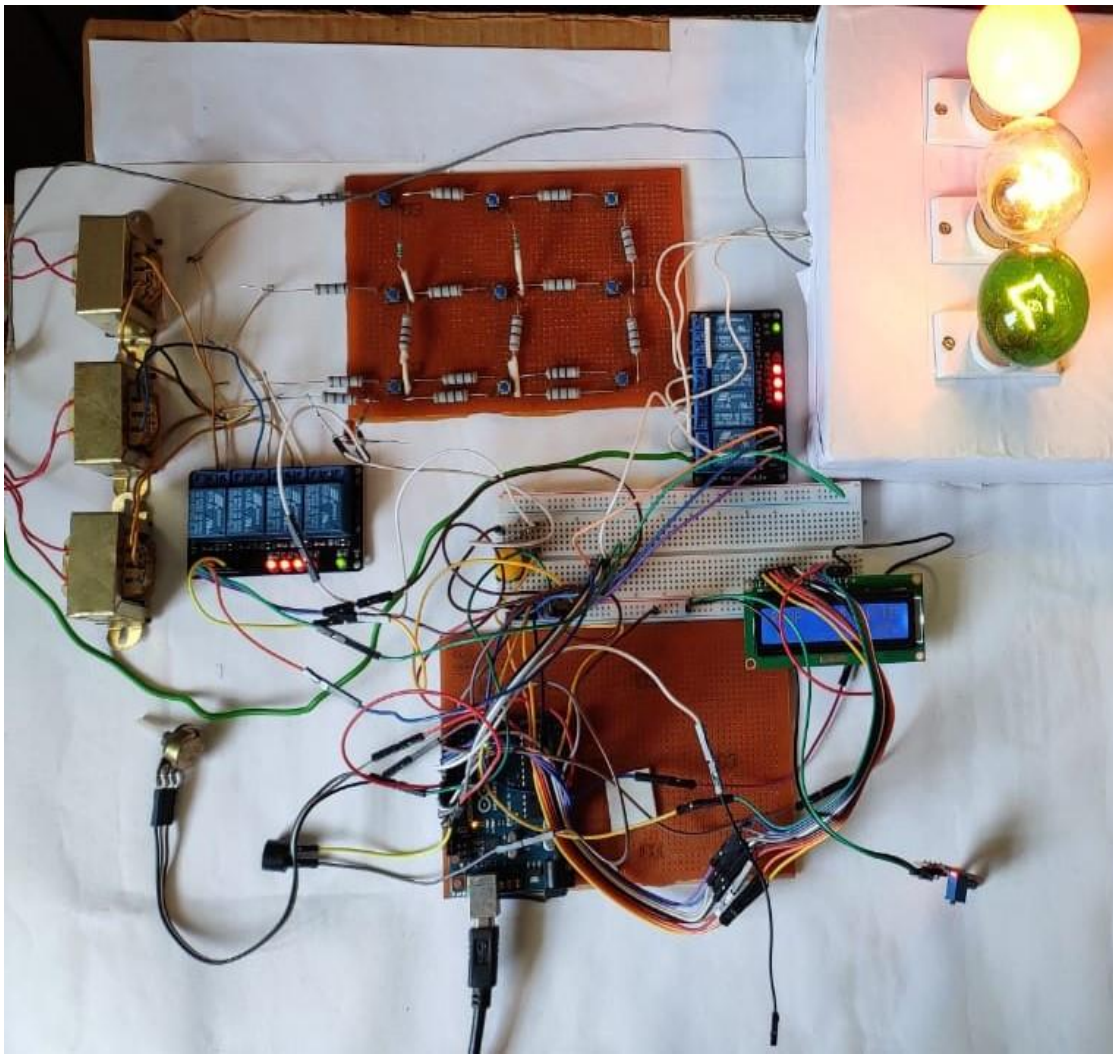


FIG 3.25 Assembled model

CHAPTER 4

RESULTS AND DISCUSSION

4.1 Observation Table and Status

SI.No	Line Fault Status			Fault Distance
	Status of R Phase	Status of Y phase	Status of B phase	
1	NF	NF	NF	
2	FF	NF	NF	3KM 6KM 9KM
3	NF	FF	NF	3KM 6KM 9KM
4	NF	NF	FF	3KM 6KM 9KM

Table 3.1 Observation Table Line Fault status

Voltage Range for fault Detection			
Distance / Fault Phase	3KM	6KM	9KM
R	1.28V-1.83V	0.9V-1.28V	1.83V-2.05V
Y	3.35V-3.8V	2.64V-2.95V	2.05V-2.35V
B	3.8V-5V	2.95V-3.35V	2.35V-2.64V

Table 3.2 Observation Table Voltage Range

4.2 Result

End result for 3-Phase fault detection circuit can be drawn as:

- So designed circuit for the detection of fault in distribution line are successfully able to detect the short circuit fault between the range of km in transmission line.
- Designed circuit is successfully able to detect the phase in which fault is occur
- Circuit is successfully able to provide information about the distance of fault from the source in the cable.
- In Case of any fire due to fault the flame sensor detects it and alerts us through Buzzer and Display the information in LCD.
- All the information about the type of fault and its distance from the source is successfully able to display on the LCD screen

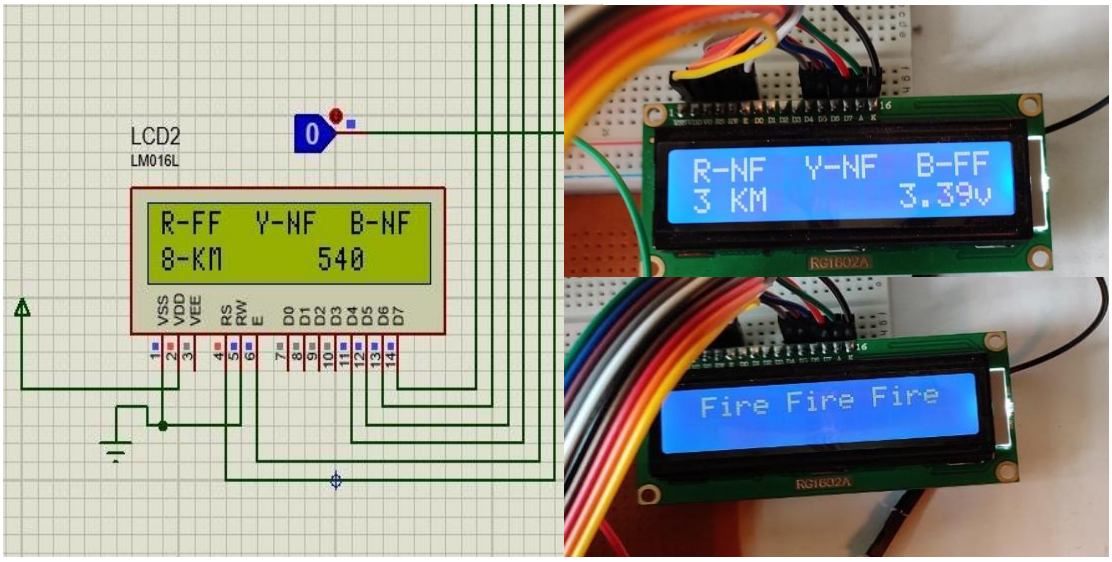


FIG 4.26 RESULT

CHAPTER 5

CONCLUSION & FUTURE SCOPE WORK

5.1 Conclusion

We have been able to successfully detect the fault by using Arduino and DC relays by taking both the input voltage as AC and DC sinusoidal. it is observed that when the current value obtained from the secondary current value of the current transformer is greater than the present value of the relay then fault is detected by the Arduino. The result is displayed on the LCD screen. We can set different trip time delay using definite time characteristics of the relay. By this project it can be ensured faster detection of faults than the electromechanical relay on the power lines and their advanced analysis can be studied from the recorded data by the Arduino programming. Also the method we followed to find out the RMS value of the voltage gave error of about 2-8% error of the actual calculated RMS value.

5.2 Future Work

1. This system provides precise information on the type of fault that occurred in the line, such as L-G, L-L, and so on.
2. Because of the WIFI Module system, which provides real-time status of the transmission system, we can easily track it from anywhere in the world. There is less upkeep.
3. This system is more adaptable than the current system, allowing it to quickly resolve the time taken to locate a fault in any area.
4. Because of its small size and light-weight, we can easily install the device on a pole.
5. Reduce human intervention.
6. More accuracy.
7. More accuracy.
8. Improved Safety & security.

REFERENCES

[1] https://www.arduino.cc/

[2] https://quartzcomponents.com/

[3] https://circuitglobe.com/

[4] http://www.oaijse.com/

[5] http://fazpublishing.com/jepes/index.php/jepes/article/view/16/14

[6]https://www.researchgate.net/deref/http%3A%2F%2Fwww.ijsrd.com%2Farticles%2FIJSRDV3I1094.pdf

[7] https://springerplus.springeropen.com/articles/10.1186/s40064-015-1080-x

[8] https://create.arduino.cc/cloud/things/169296e8-4259-4b5d-8d51-f559f08a48eb

[9] https://sqingredients.com/v0s9qhp/3-phase-detection-using-arduino.html

[10] https://www.arduino.coach/iot-under-ground-cable-fault-detection-using-arduino-3.html

[11]https://www.researchgate.net/publication/321097011_AUTOMATIC_FAULT_DETECTION_AND_PROTECTION_OF_THREE_PHASE_INDUCTION_MOTOR

APPENDIX 1

Ardunio Code:

```
#include<LiquidCrystal.h>  //To include the 1602 lcd header file
int rs=12, en=11, d4=5, d5=4, d6=3, d7=2; //LCD connection pins to arduino
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);  //Creating lcd object
int linePin = A0, lineValue;  //Analog input pin for line voltage and to store its values
float V;
int faultStatus = 0;
int R_relay_Pin = 8, Y_relay_Pin = 9, B_relay_Pin = 10;  //Realy Pins
int flame_sensor_Pin = A1, flame_sensor_value;  //Flame Sensor Pin and to store its value
int fire = 0, buzzerPin = 7, esp;
void setup() {
  Serial.begin(9600);
  lcd.begin(16,2);
  pinMode(linePin, INPUT);
  pinMode(R_relay_Pin, OUTPUT);
  pinMode(Y_relay_Pin, OUTPUT);
  pinMode(B_relay_Pin, OUTPUT);
  pinMode(flame_sensor_Pin, INPUT);
  pinMode(buzzerPin, OUTPUT);
  lcd.clear();
  lcd.setCursor(0,0); lcd.print(" 3 PHASE FAULT");
  lcd.setCursor(0,1); lcd.print("DETECTION SYSTEM");
  delay(1000);
  lcd.clear();
  lcd.setCursor(0,0); lcd.print("R-NF");
  lcd.setCursor(6,0); lcd.print("Y-NF");
  lcd.setCursor(12,0); lcd.print("B-NF");
  esp = 1;  //Default value
  Serial.println(esp);
```

```
}
void loop() {
  flame();
  lineValue = analogRead(linePin);
  V = (lineValue/1023.)*5.;
  lcd.setCursor(11,1); lcd.print(V); lcd.print("v");  //To pdisplay current DC voltage across
the resistor
  if(faultStatus==0) {
    faultCheck(V); //Calling function to check for faulty condition if faultStatus is 0
  }
  if(faultStatus==1){
    Serial.println(esp);
    faultStatus = 2;
  }
  delay(500);
}
void flame() {
  do {
    flame_sensor_value = analogRead(flame_sensor_Pin);
    if(flame_sensor_value<70) {
      if(fire==0){
        esp = 2; Serial.println(esp);  //To display Fire! Fire! Fire!
      }
      digitalWrite(buzzerPin, HIGH);
      lcd.clear();
      lcd.setCursor(0,0); lcd.print(" Fire Fire Fire ");
      fire = 1;
      delay(500);
    }
    else {
      digitalWrite(buzzerPin, LOW);
      if(fire == 1) {
        lcd.clear();
        lcd.setCursor(0,0); lcd.print("R-NF");
        lcd.setCursor(6,0); lcd.print("Y-NF");
        lcd.setCursor(12,0); lcd.print("B-NF");
```

```
      fire = 0;
     }
    }
  }while(fire);
}
void faultCheck(float V) {
  if(V>=3.80) { //Fault at 3KM in B phase
    lcd.setCursor(14,0); lcd.print("F");
    lcd.setCursor(0,1); lcd.print("3 KM");
    digitalWrite(B_relay_Pin, HIGH);
    lcd.setCursor(5,1); lcd.print(V); lcd.print("v");
    esp = 9;  //To display Fault in B-Phase at 3 KM
    faultStatus = 1;
  }
  else if(V>=3.35) {  //Fault at 3 KM in Y phase
    lcd.setCursor(8,0); lcd.print("F");
    lcd.setCursor(0,1); lcd.print("3 KM");
    digitalWrite(Y_relay_Pin, HIGH);
    lcd.setCursor(5,1); lcd.print(V); lcd.print("v");
    esp = 6;  //To display Fault in Y-Phase at 3 KM
    faultStatus = 1;
  }
  else if(V>=2.95) {  //Fault at 6 KM in B phase
    lcd.setCursor(14,0); lcd.print("F");
    lcd.setCursor(0,1); lcd.print("6 KM");
    digitalWrite(B_relay_Pin, HIGH);
    lcd.setCursor(5,1); lcd.print(V); lcd.print("v");
    esp = 10;  //To display Fault in B-Phase at 6 KM
    faultStatus = 1;
  }
  else if(V>=2.64) {  //Fault at 6 KM in Y phase
    lcd.setCursor(8,0); lcd.print("F");
    lcd.setCursor(0,1); lcd.print("6 KM");
    digitalWrite(Y_relay_Pin, HIGH);
    lcd.setCursor(5,1); lcd.print(V); lcd.print("v");
    esp = 7;  //To display Fault in Y-Phase at 6 KM
```

```
  faultStatus = 1;
}
else if(V>=2.35) {  //Fault at 9 KM in B phase
  lcd.setCursor(14,0); lcd.print("F");
  lcd.setCursor(0,1); lcd.print("9 KM");
  digitalWrite(B_relay_Pin, HIGH);
  lcd.setCursor(5,1); lcd.print(V); lcd.print("v");
  esp = 11;  //To display Fault in B-Phase at 9 KM
  faultStatus = 1;
}
else if(V>=2.05) {  //Fault at 9 KM in Y phase
  lcd.setCursor(8,0); lcd.print("F");
  lcd.setCursor(0,1); lcd.print("9 KM");
  digitalWrite(Y_relay_Pin, HIGH);
  lcd.setCursor(5,1); lcd.print(V); lcd.print("v");
  esp = 8;  //To display Fault in Y-Phase at 9 KM
  faultStatus = 1;
}
else if(V>=1.83) {  //Fault at 9 KM in R phase
  lcd.setCursor(2,0); lcd.print("F");
  lcd.setCursor(0,1); lcd.print("9 KM");
  digitalWrite(R_relay_Pin, HIGH);
  lcd.setCursor(5,1); lcd.print(V); lcd.print("v");
  esp = "R9";  //To display Fault in R-Phase at 9 KM
  faultStatus = 5;
}
else if(V>=1.28) {  //Fault at 3 KM in R phase
  lcd.setCursor(2,0); lcd.print("F");
  lcd.setCursor(0,1); lcd.print("3 KM");
  digitalWrite(R_relay_Pin, HIGH);
  lcd.setCursor(5,1); lcd.print(V); lcd.print("v");
  esp = 3;  //To display Fault in R-Phase at 3 KM
  faultStatus = 1;
}
else if(V>=0.90) {  //Fault at 6 KM in R phase
  lcd.setCursor(2,0); lcd.print("F");
```

```
  lcd.setCursor(0,1); lcd.print("6 KM");
  digitalWrite(R_relay_Pin, HIGH);
  lcd.setCursor(5,1); lcd.print(V); lcd.print("v");
  esp = 4;  //To display Fault in R-Phase at 6 KM
  faultStatus = 1;
 }
 else {  //NO fault case
  lcd.setCursor(2,0); lcd.print("N");
  lcd.setCursor(8,0); lcd.print("N");
  lcd.setCursor(14,0); lcd.print("N");
  lcd.setCursor(0,1); lcd.print("    ");
  digitalWrite(R_relay_Pin, LOW);
  digitalWrite(Y_relay_Pin, LOW);
  digitalWrite(B_relay_Pin, LOW);}}
```

APPENDIX 2

Node MCU ESP8266 Code:

```
#include "arduino_secrets.h"
/*
  Sketch generated by the Arduino IoT Cloud Thing "Project"
  https://create.arduino.cc/cloud/things/169296e8-4259-4b5d-8d51-f559f08a48eb

  Arduino IoT Cloud Variables description

  The following variables are automatically generated and updated when changes are made to the Thing

  String message;

Variables which are marked as READ/WRITE in the Cloud Thing will also have functions
  which are called when their values are changed from the Dashboard.
  These functions are generated with the Thing and added at the end of this sketch.
*/

#include "thingProperties.h"

void setup() {
  // Initialize serial and wait for port to open:
  Serial.begin(9600);
  // This delay gives the chance to wait for a Serial Monitor without blocking if none is found
  delay(1500);

  // Defined in thingProperties.h
  initProperties();

  // Connect to Arduino IoT Cloud
  ArduinoCloud.begin(ArduinoIoTPreferredConnection)

  /*
     The following function allows you to obtain more information
     related to the state of network and IoT Cloud connection and errors
```

```
    the higher number the more granular information you'll get.
    The default is 0 (only errors).
    Maximum is 4
 */
 setDebugMessageLevel(2);
 ArduinoCloud.printDebugInfo();
 message = "WELCOME TO 3-PHASE FAULT DETECTION SYSTEM";
}
void loop() {
 ArduinoCloud.update();
 // Your code here
 while(Serial.available()>0)
 {
 int msg = Serial.parseInt();
 Serial.println(msg);
 switch(msg){
   case 1: message = "R-NF Y-NF B-NF";
           break;
   case 2: message = "Fire! Fire! Fire!";
           break;
   case 3 : message = "Fault in R-Phase at 3 KM";
           break;
   case 4 : message = "Fault in R-Phase at 6 KM";
           break;
   case 5 : message = "Fault in R-Phase at 9 KM";
           break;
   case 6 : message = "Fault in Y-Phase at 3 KM";
           break;
   case 7 : message = "Fault in Y-Phase at 6 KM";
           break;
   case 8 : message = "Fault in Y-Phase at 9 KM";
           break;
   case 9 : message = "Fault in B-Phase at 3 KM";
           break;
   case 10 : message = "Fault in B-Phase at 6 KM";
           break;
```

```
   case 11 : message = "Fault in B-Phase at 9 KM";
             break;
  }
  Serial.println(message); }}
```

APPENDIX 3

Pin Diagram:

ARDUINO

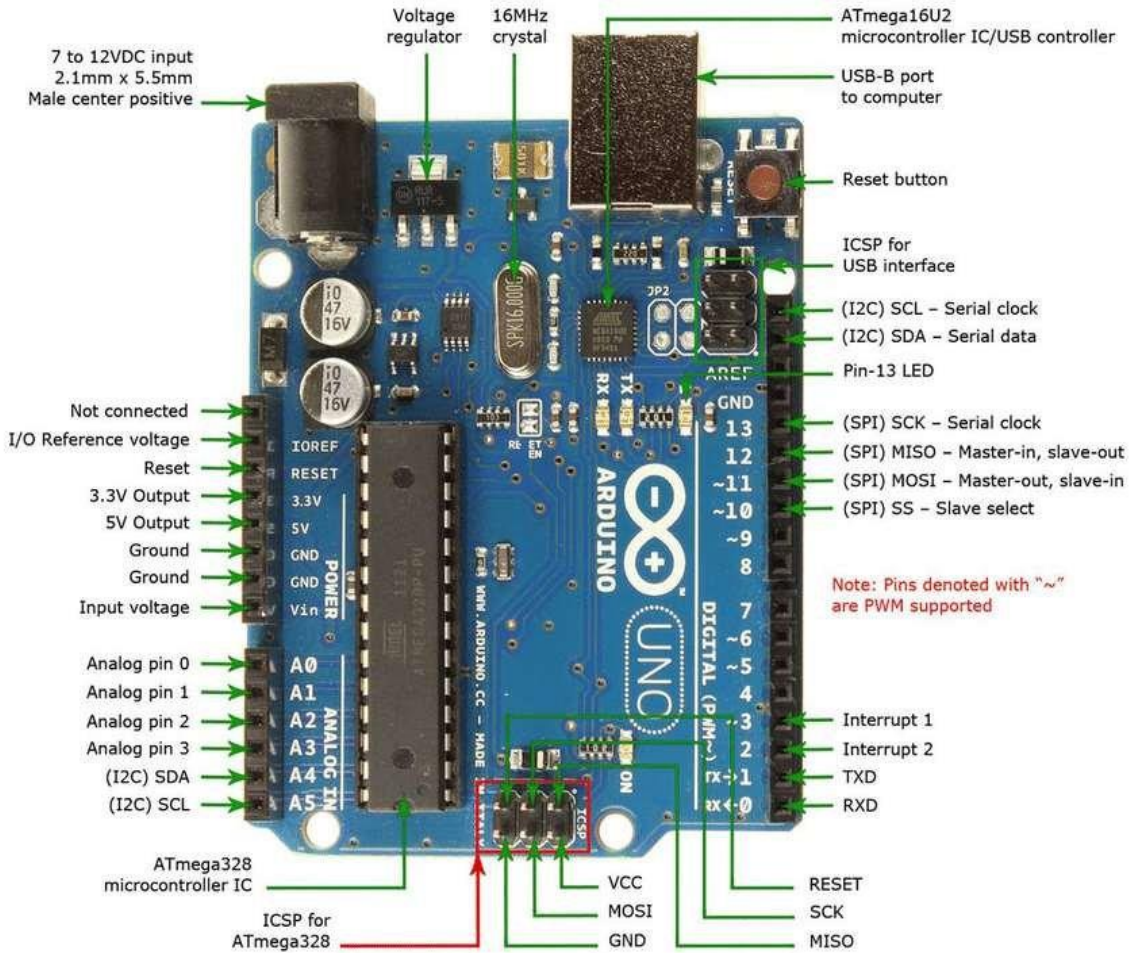


Fig AP.1 Ardunio Pin Diagram

Vin: This is the input voltage pin of the Arduino board used to provide input supply from an external power source.

5V: This pin of the Arduino board is used as a regulated power supply voltage and it is used to give supply to the board as well as onboard components.

3.3V: This pin of the board is used to provide a supply of 3.3V which is generated from a voltage regulator on the board.

GND: This pin of the board is used to ground the Arduino board.

Reset: This pin of the board is used to reset the microcontroller. It is used to Resets the microcontroller.

Analog Pins: The pins A0 to A5 are used as an analog input and it is in the range of 0-5V.

Digital Pins: The pins 0 to 13 are used as a digital input or output for the Arduino board.

Serial Pins: These pins are also known as a UART pin. It is used for communication between the Arduino board and a computer or other devices. The transmitter pin number 1 and receiver pin number 0 is used to transmit and receive the data resp.

External Interrupt Pins: This pin of the Arduino board is used to produce the External interrupt and it is done by pin numbers 2 and 3.

PWM Pins: This pin of the board is used to convert the digital signal into an analog by varying the width of the Pulse. The pin numbers 3,5,6,9,10 and 11 are used as a PWM pin.

SPI Pins: This is the Serial Peripheral Interface pin, it is used to maintain SPI communication with the help of the SPI library. SPI pins include:

1. SS: Pin number 10 is used as a Slave Select
2. MOSI: Pin number 11 is used as a Master Out Slave In
3. MISO: Pin number 12 is used as a Master In Slave Out
4. SCK: Pin number 13 is used as a Serial Clock

LED Pin: The board has an inbuilt LED using digital pin-13. The LED glows only when the digital pin becomes high.

AREF Pin: This is an analog reference pin of the Arduino board. It is used to provide a reference voltage from an external power supply.

APPENDIX 4

Pin Diagram

Node MCU ESP8266

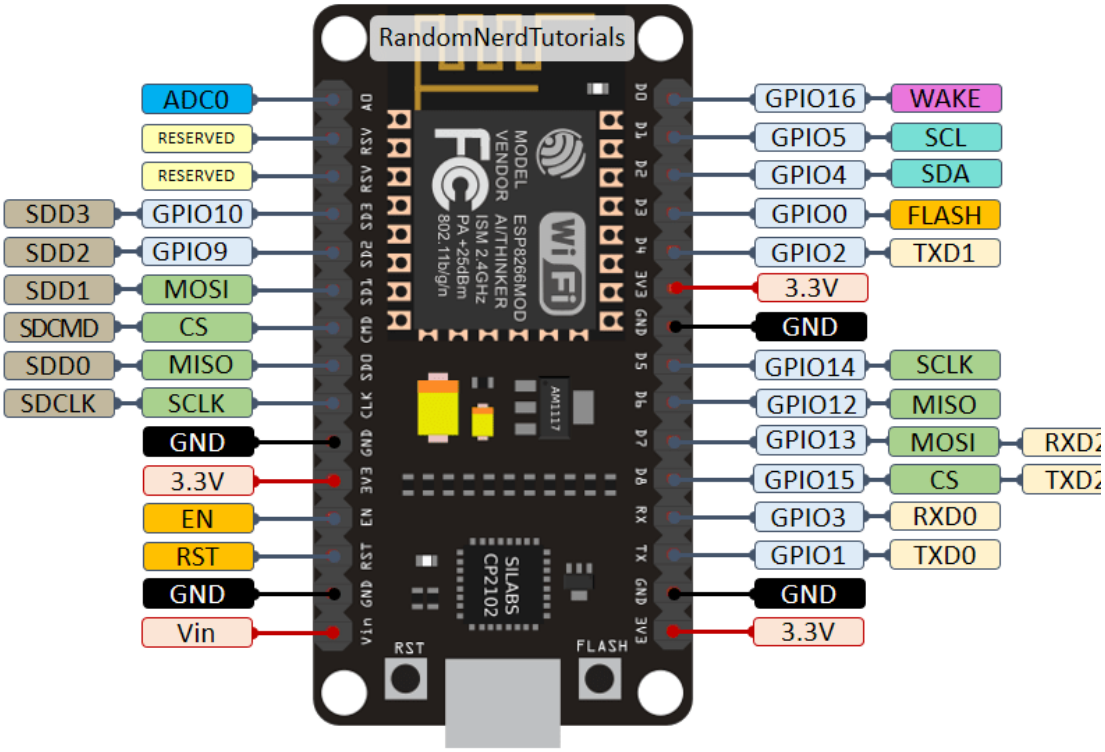


Fig AP.2 Nodemcu ESP8266 Pin Diagram

NodeMCU Development Board Pinout Configuration

Pin Category	Name	Description

Power	Micro-USB, 3.3V, GND, Vin	**Micro-USB:** NodeMCU can be powered through the USB port **3.3V:** Regulated 3.3V can be supplied to this pin to power the board **GND:** Ground pins **Vin:** External Power Supply
Control Pins	EN, RST	The pin and the button resets the microcontroller
Analog Pin	A0	Used to measure analogy voltage in the range of 0-3.3V
GPIO Pins	GPIO1 to GPIO16	NodeMCU has 16 general purpose input-output pins on its board
SPI Pins	SD1, CMD, SD0, CLK	NodeMCU has four pins available for SPI communication.
UART Pins	TXD0, RXD0, TXD2, RXD2	NodeMCU has two UART interfaces, UART0 (RXD0 & TXD0) and UART1 (RXD1 & TXD1). UART1 is used to upload the firmware/program.
I2C Pins		Node MCU has I2C functionality support but due to the internal functionality of these pins, you have to find which pin is I2C.