A Project Report

"ARDUINO-DRIVEN SYSTEM FOR UNDERGROUND CABLE FAULT IDENTIFICATION

submitted in partial fulfillment of the requirements for the award of the Degree

BACHELOR OF TECHNOLOGY

IN

ELECTRICAL AND ELECTRONICS ENGINEERING

Submitted by

G.Ghana Shyam (22B85A0207)

Ch.Siva Ram Sai (22B85A0203)

T.Mohan Raghu Varma (21B81A0267)

P.Soma Lakshmi (21B81A0257)

M.Samuyelu (22B85A0216)

Under the guidance of

Sri. S. Raghunath Sagar , M.Tech.
Associated Professor, Dept of EEE

Mrs K. Renuka Sowjanya ,M.Tech
Assistant Professor, Dept of EEE

DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

SIR C.R.REDDY COLLEGE OF ENGINEERING

Approved by AICTE - Accredited by NBA

(Affiliated to Jawaharlal Nehru Technological University, Kakinada)

Eluru-534007, Andhra Pradesh

DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

SIR C.R.REDDY COLLEGE OF ENGINEERING

Approved by AICTE - Accredited by NBA

(Affiliated to Jawaharlal Nehru Technological University, Kakinada)

Eluru-534007, Andhra Pradesh

(2024-2025)

CERTIFICATE

This is to certify that project report entitled "ARDUINO-DRIVEN SYSTEM FOR UNDERGROUND CABLE FAULT IDENTIFICATION" being submitted by

G.Ghana Shyam (22B85A0207)

T.Mohan Raghu Varma (21B81A0267)

M.Samuyelu (22B85A0216)

Ch.Siva ram Sai (22B85A0203)

P.Soma Lakshmi (21B81A0257)

in partial fulfillment for the award of the Degree of Bachelor of Technology in Electrical and Electronics Engineering to the Jawaharlal Nehru Technological University, KAKINADA is a record of bonafied work carried out under my guidance and supervision. The results embodied in this project report have not been submitted to any other University or Institute for the award of any Degree or diploma.

Sri. S.R. SAGAR ,M.tech	**Mrs K. RENUKA SOWJANYA,M.tech**	**Dr. A. SRINIVASA REDDY**
Associated Professor, Dept of EEE	Assistant Professor, Dept of EEE	HOD, Dept of EEE

ACKNOWLEDGEMENT

We wish to express our sincere thanks to various personalities who were responsible for the successful completion of this project.

We express our special gratitude to **Dr. K. VENKATESHWAR RAO**, Principal of Sir C.R.Reddy College of Engineering, who made this endeavour possible.

We are thankful to Sir. **Dr. A. SRINIVASA REDDY**, Head of Electrical and Electronics Engineering department, for providing the necessary facilities for completing the project in specified time.

We wish to express our deep gratitude to **Sri. S. RAGHUNATH SAGAR**, Associative Professor, for his mastery supervision through all the phases of our main project work with valuable suggestions and support. His friendly and informal talks helped us to accomplish our project successfully in time.

Our special thanks to **Smt. LAKSHMI KUMARI**, Librarian for providing necessary library facilities.

We would also thank all the teaching and non-teaching staff of Electrical and Electronics Engineering department who have extended their full cooperation.

Last but not least we are thankful to our friends clarified the doubts during the completion of our project.

Project associates

G.GHANA SHYAM	(22B85A0207)
M.SAMUYELU	(22B85A0216)
T.MOHAN RAGHU VARMA	(21B81A0267)
CH.SIVA RAM SAI	(22B85A0203)
P.SOMA LAKSHMI	(21B81A0257)

DECLARATION

We hereby declare that the project report entitled "**Arduino-Driven System For Underground Cable Fault Identification**" is submitted to JNTUK is partial fulfilment of the requirement for the award of the degree of Bachelor of Technology (B. Tech) is an original work carried out by us. The matter embodied in this project is a genuine work by the students and has not been submitted earlier to this university or any other university for award of any Degree.

Project associates

G.GHANA SHYAM	(22B85A0207)
M.SAMUYELU	(22B85A0216)
T.MOHAN RAGHU VARMA	(21B81A0267)
CH.SIVA RAM SAI	(22B85A0203)
P.SOMA LAKSHMI	(21B81A0257)

ABSTRACT

With demand increasing day-by-day, underground cables are used more and more than overhead lines in urban areas. Whenever a fault occurs in underground cables, it is difficult to detect the exact location of the fault. Locating the fault is a tedious process and there is always a probability of damaging the insulation while digging the cable. As a workaround to this problem,in this project, an easy and smart fault detection is proposed that can automate fault detection using an Arduino controller.

The system used in this project uses the basic concept of Ohm's law is applied at the cables end though a series resistor. In case of short circuit of phase to ground the current flowing in the faulty sections will vary depending on the length of the in underground cable lines from the base station in km using a Arduino microcontroller. To locate a fault in the cable, the cable must be tested for faults. The current would vary depending upon the length of fault of the cable. This model is modelled with a set of wires representing cable length in km and fault creation is made by a set of switches in reference with ground at every known distance to cross check the accuracy of the same. In case of fault, the voltage across series wires change accordingly, which is then fed to an ADC to develop precise digital data to a programmed Arduino that further displays fault location in distance. The fault occurring distance is displayed on a 16*2 LCD interfaced with the microcontroller and identified by Buzzer Sound,enabling remote monitoring and prompt corrective actions. This system ensures rapid detection and localization of faults, significantly reducing troubleshooting time

TABLE OF CONTENTS

	Page. No
CERTIFICATE	
ACKNOWLEDGEMENT	I
ABSTRACT	II
CHAPTER – 1	
INTRODUCTION	
1.1 Introduction	1
1.2 Construction of Underground Cables	2
1.3 Types of Underground Cables	3
1.4 Types of Faults	6
1.5 Fault detection methods	9
1.6 Objective	10
1.7 Problem Statement	11
1.8 Operating principle	11
CHAPTER-2	
PROPOSED METHODOLOGY & CALUCATIONS	
2.1 Proposed system	
2.1.1 Overview of proposed system	12
2.1.2 Components used	12
2.2 Methodology And Approach	
2.2.1 Module-1: Software design	12
2.2.2 Module – 2: Hardware Design	13
2.2.3 Block Diagram	13
2.3 Flow Chart	14
2.4 Single Line Diagram	14

2.5 Calucations

2.5.1 Operational Calculations 15

2.5.2 Theoretical Calculations 16

2.5.3 ADC Calucations 17

CHAPTER-3

MODULE-1: SOFTWARE DESIGN

3.1 Introduction 19

3.2 Arudino IDE

3.2.1 Programming 19

3.2.2 Funtions 20

3.2.3 Code 20

3.3 Proteus Software

3.3.1 Introduction 21

3.3.2 Steps To Design Proteus Circuit 22

3.3.3 Simulation Circuit 27

3.4 Proteus Working 27

CHAPTER-4

MODULE-2 HARDWARE IMPLEMENTATION AND DESIGN

4.1 Introduction 28

4.2 Hardware Description 28

4.2.1 ARDUINO UNO 28

4.2.1.1 OVERVIEW 28

4.2.1.2 Schematic Design of Arduino UNO 29

4.2.1.3 Summary 30

4.2.1.4 Input and Output Pins 30

4.2.1.5 Communications 31

4.2.1.6 Programming 31

4.2.2 Relays Module 32

4.2.2.1 Table: - Technical Data 32

4.2.2.2 Characteristics 32
4.2.2.3 Applications 33
4.2.3 ULN2003A 33
4.2.3.1 Table: - Technical Data 33
4.2.3.2 Characteristics 34
4.2.3.3 Applications 34
4.2.4 Trimpot Potentiometer 34
4.2.4.1 Table: - Technical Data 34
4.2.4.2 Characteristics 35
4.2.4.3 Applications 35
4.2.5 LCD DISPLAY 35
4.2.5.1 Specifications 36
4.2.5.2 Applications 36
4.2.6 LED Bulb 36
4.2.7 Buzzer 37
4.2.8 Jumper Wires 37
4.3 Circuit Design
4.3.1 Circuit Connections 38
4.3.2 Hardware Circuit Implementation 39
4.3.3 Working of Hardware 39

CHAPTER-5

RESULTS AND DESCRIPTIONS

5.1 Theoretical Calucations Vs Practical Calucations
5.1.1 Under no fault condition 40
5.1.2 Under fault condition 41
5.1.2.1 Sub case-1 41
5.1.2.2 Subcase-2 42
5.1.2.3 Subcase-3 43
5.2 Software Results 43

5.2.1 No-Fault Condition in Under ground cable fault detection Using Simulation:

5.2.1.1 Fault Condition in Under ground cable fault detection Using Simulation at R- phase: 44

5.2.1.2 Fault Condition in Under ground cable fault detection Using Simulation at Y-phase: 45

5.2.1.3 Fault Condition in Under ground cable fault detection Using Simulation at B-phase: 45

5.3 Hardware Results 46

5.3.1 No-Fault Condition in Under ground cable fault detection using Hardware design 46

5.3.1.1 Fault Condition in Under ground cable fault detection using Hardware design at R-phase 46

5.3.1.2 Under Fault Condition in Under ground cable fault detection using Hardware design at Y-phase 47

5.3.1.3 Under Fault Condition in Under ground cable fault detection using Hardware design at B-phase 48

5.4 Comparison of Results

5.4.1 Comparison Between Simulation And Hardware Results under no fault Condition 48

5.4.2 Comparision Between Simulation And Hardware Results at R -phase: 49

5.4.3 Comparision Between Simulation And Hardware Results at Y -Phase: 49

5.4.4 Comparision Between Simulation And Hardware Results at B-Phase: 50

Future scope 51

Conclusion 52

References 53

List of Figures

Page.No:

Fig 1:-Under ground Cables 1

Fig 2:-Layout of Under ground Cables 2

Fig 3:-Single core cables 3

Fig 4(a),(b) Three core cables 4

Fig:-5 Low tension cables 4

Fig:-6 Belted cables 5

Fig:-7 H-type cables 5

Fig:-8 SL-type cables 5

Fig:-9 Oil filled cables 6

Fig:-10 Gas Pressure cables 6

Fig:-11 Types of faults 7

Fig:-12 One Open Conductor Faults 7

Fig:-13 Three Open Conductor Faults 7

Fig:-14 3-Phase Short Circuit Faults 8

Fig:-15 3-Phase to Ground Faults 8

Fig:-16 Single Line to Ground Fault 8

Fig:-17 Line to Line Faults 8

Fig:-18 DoubleLine to Ground Faults 9

Fig:-19 Tracer 9

Fig:-20 Cable thumping method 10

Fig:-21 Time-domain reflectometer 10

Fig:-22 Block diagram 13

Fig:-23 Line Diagram 14

Fig:- 24 Flowchart 19

Fig:-25 Simulation Circuit 27

Fig:-26 Arduino Uno 29

Fig:-27 Schematic design of Arduino 29

Fig:-28 Relay Module 32
Fig:-29 ULN2003A 33
Fig:-30 Pin diagram of ULN2003A 33
Fig:-31 Trimpot Potentiometer 34
Fig:-32 16 * 2 LCD Display 35
Fig:-33 LED 36
Fig:-34 Buzzer 37
Fig:-35 Jumper Wires 37
Fig:-36 Circuit Connections 38
Fig:-37 Hardware Kit implementation 39
Fig:-38 Proteus Simulation Results under No-Fault Condition 44
Fig 39: - Proteus Simulation Results under Fault Condition 44
Fig 40: - Proteus Simulation Results under Fault Condition 45
Fig 41: - Proteus Simulation Results under Fault Condition 45
Fig:-42: - Hardware results under No-Fault Condition 46
Fig 43:- Hardware results under Under R Phase Fault Condition 47
Fig 43: - Hardware results under Under Y PhaseFault Condition 47
Fig 44: - Hardware results under Under B Phase Fault Condition 48

CHAPTER-1

INTRODUCTION

1.1 Introduction

Till last decades cables were made to lay overhead& currently it is lay to underground cable which is superior to earlier method. Because the underground cable are not affected by any extreme or unfavourable weather condition such as storm, snow, heavy rainfall as well as pollution. Underground cables are electrical power cables that are buried in the ground for the transmission and distribution of electricity. They are often used in urban areas where overhead power lines are not practical due to aesthetic or safety reasons. Underground cables are typically insulated with materials such as XLPE or EPR to protect against moisture, corrosion, and other environmental factors. They offer a number of advantages over overhead lines, including reduced visual impact, lower maintenance requirements, and reduced susceptibility to weather-related outages.

Fig 1:-Under ground Cables

As far as we know electricity can be distributed and transmitted using underground wires or overhead lines. In comparison to overhead systems, underground cables have a number of benefits including robust construction, higher service reliability, increased safety, fewer chances of faults, low maintenance costs, a better appearance, and less interference from outside disturbances like storms, lightning, ice, trees, etc. Even, compared to an equivalent overhead system, they have more expensive to installation costs and insulation concerns at high voltages. Hence, cables are mostly used in situations where using overhead lines is impractical

1.2 Construction of Underground Cables

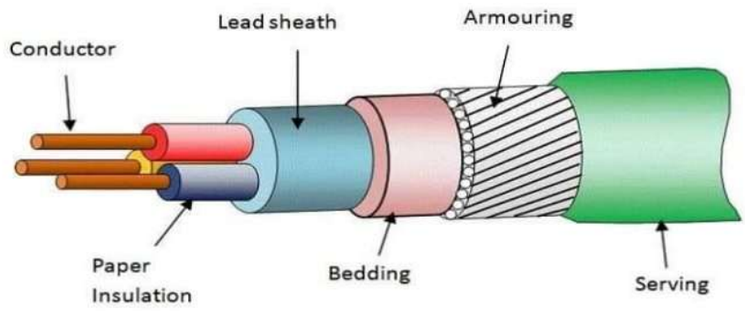


Fig 2:-Layout of Under ground Cables

Cores or Conductors: A cable may have one or more than one core (conductor) depending upon the type of service for which it is intended. For instance, the 3-conductor cable shown in Fig. 2 is used for 3-phase service. The conductors are made of tinned copper or aluminium and are usually stranded in order to provide flexibility to the cable.

Insulation: Each core or conductor is provided with a suitable thickness of insulation, the thickness of layer depending upon the voltage to be withstood by the cable. The commonly used materials for insulation are impregnated paper, varnished cambric or rubber mineral compound.

Metallic sheath: In order to protect the cable from moisture, Conductor gases or other damaging liquids (acids or alkalies) in the soil and atmosphere, a metallic sheath of lead or aluminium is provided over the insulation as shown in Fig. 2

Bedding: Over the metallic sheath is applied a layer of bedding which consists of a fibrous material like jute or hessian tape. The purpose of bedding is to protect the metallic sheath against corrosion and from mechanical injury due to armouring.

Armouring: Over the bedding, armouring is provided which consists of one or two layers of galvanised steel wire or steel tape. Its purpose is to protect the cable from mechanical injury while laying it and during the course of handling. Armouring may not be done in the case of some cables.

Serving: In order to protect armouring from atmospheric conditions, a layer of fibrous material (like jute) similar to bedding is provided over the armouring. This is known as serving.

1.3 Types of Underground Cables

Classification of underground cables

The classification of Underground cables can be done on the basis of several criteria includes:

1. Number of conductors in the cable
2. Voltage rating of the cable
3. Construction of cable
4. Type and thickness of insulation used

Classification based upon number of conductors in the cable

1. Single core cable
2. Three core cable

Single core cables: A single-core cable has a single conductor. It is insulated by layers of suitable materials. It is mainly used in applications needing one conductor, like simple electrical installations. These cables are favoured for their simplicity and easy installation. They are ideal where only a few conductors are needed. Single-core cables are often used in underground systems. They provide robust, reliable performance.

Fig 3:-Single core cables

Three core cable: A three-core cable contains three separate conductors, each insulated from the others. These cables are used in three-phase power systems. Each conductor carries one of the three phases. Underground cables are usually employed to deliver 3 phase power. A 3 cored cable is preferred up to 66 kV. Beyond that, insulation required for the cable is too much.

(a)

(b)

Fig 4(a),(b) Three core cables

For higher voltages, 3 cored constructions become too bulky, and hence, even with some limitations we employ single cored cables

Classification based upon voltage rating of the cable

1. Low tension cables (LT)
2. High tension cables (HT)
3. Super tension cables (ST)
4. Extra high tension cables (EHT)
5. Extra super voltage cables

Low tension cables: These have a maximum voltage handling capacity of 1000 V (1 kV).A cable may have one or more than one core depending upon the type of service for which it is intended. It may be (i) single-core (ii) two core (iii) three-core (iv) four-core etc.

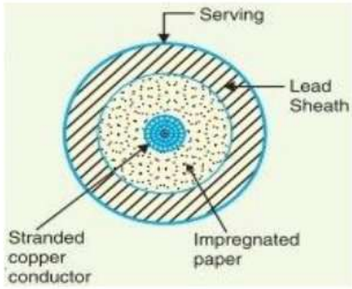


Fig:-5 Low tension cables

High Tension Cable: These are Three core cables i.e. multi core cables and are used for voltage level upto 66kV. It is of two types,

1. Belted cables – upto 11kv

2. Screened type cables. – upto 66kv
    i. H-type Cables
    ii. S-L (Separate Lead) type Cables.

1. **Belted cables**: These cables are used for voltage level up to 11kV. The cores are not circular in shape and insulation present is of impregnated paper and these cores are belted together with help of paper belt. Gaps are filled with fibrous material like jute and then it is covered with sheath and other things.

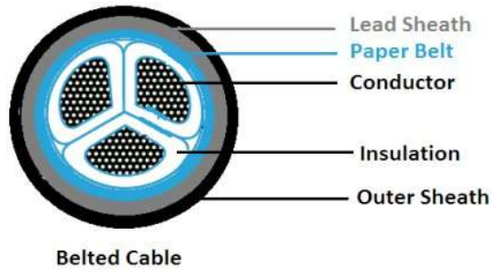


Fig:-6 Belted cables

2. **Screened type cables:** These cables are used for voltage levels upto 22kV – 33kV ,In H-type cable there is no paper belt used and each conductor is insulated with paper covered with metallic screen of aluminium foil. In SL- type, each core is first insulated with impregnated paper and then with separate lead sheath.

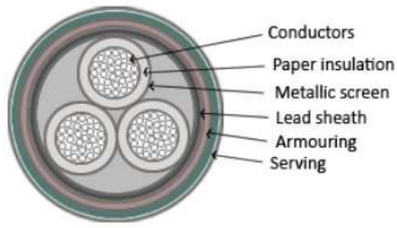


Fig:-7 H-type cables

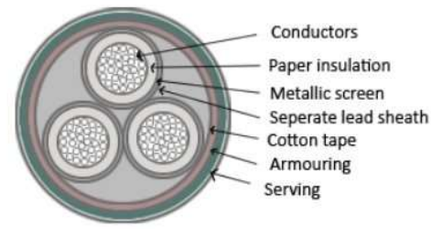


Fig:-8 SL-type cables

Super Tension Cable: These are used for voltage levels upto 132kV to 275kV. It is also of two types:

1. Oil filled Cable
2. Gas Pressure Cable

1. **Oil filled Cable**: In this the ducts are provided within or adjacent to the cores through which oil under pressure is circulated. It has concentric stranded conductor, built around a hollow cylindrical steel spiral core. This hollow core acts as channel for oil.

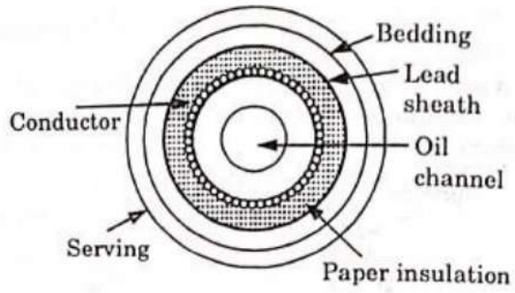


Fig:-9 Oil filled cables

2. **Gas Pressure Cable**: In this a inert gas like nitrogen at high pressure (12-15atm) is introduced. There is no bedding and serving in this cable. The working power factor of these type of cables are high.

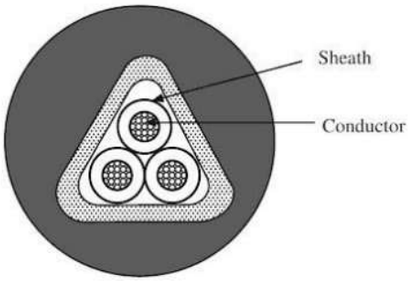


Fig:-10 Gas Pressure cables

Super tension cables: These have a maximum voltage handling capacity of 33 kV.

Extra high tension cables: These have a maximum voltage handling capacity of 66 kV.

Extra super voltage cables: These are used for applications with voltage requirement above 132 kV.

1.4 Types of Faults

The most common types of faults in underground cables are open circuit, short circuit, and earth faults

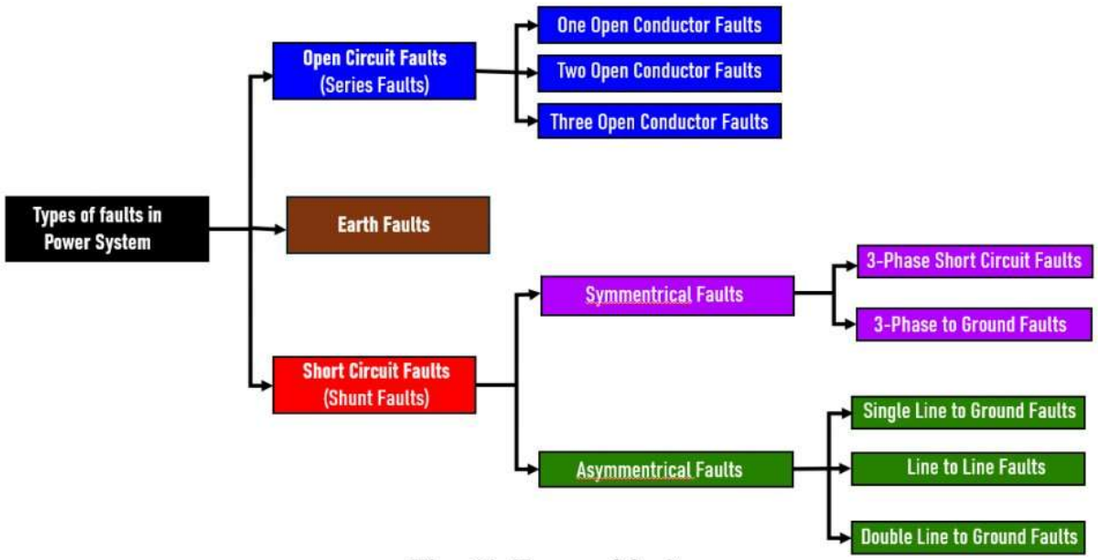


Fig:-11 Types of faults

Open circuit fault:

- A break in the conductor of a cable
- Can be detected using a megger
- A megger will indicate infinite resistance if a conductor is broken

Open Circuit Faults are classified into three types

i. One Open Conductor Faults

ii. Two Open Conductor Faults

iii. Three Open Conductor Faults

Fig:-12 One Open Conductor Faults & Two Open Conductor Faults

Fig:-13 Three Open Conductor Faults

Short circuit fault:

- When two conductors of a multi-core cable come into contact with each other due to insulation failure
- Can be detected using a megger
- A megger will indicate zero resistance if there is a short circuit

Short Circuit Faults are classified into two types

i. Symmentrical Faults
   - ✓ 3-Phase Short Circuit Faults(L-L-L)
   - ✓ 3-Phase to Ground Faults(L-L-L-G)

ii. Asymmentrical Faults
   - ✓ Single Line to Ground Fault(L-G)
   - ✓ Line to Line Faults(L-L)
   - ✓ Double Line to Ground Faults(L-L-G)

Symmentrical Faults

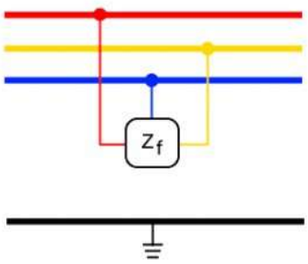


Fig:-14 3-Phase Short Circuit Faults

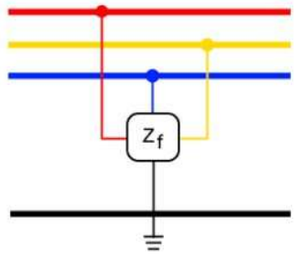


Fig:-15 3-Phase to Ground Faults

Asymmentrical Faults

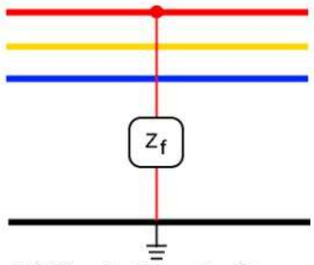


Fig:-16 Single Line to Ground Fault

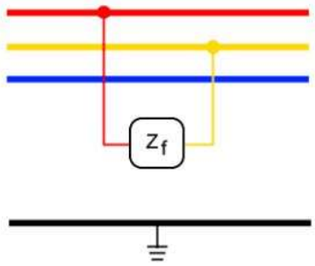


Fig:-17 Line to Line Faults

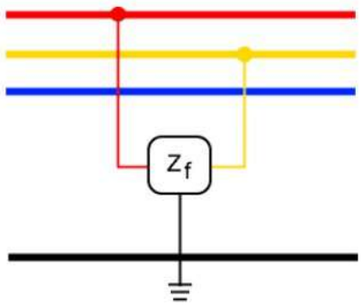


Fig:-18 DoubleLine to Ground Faults

Earth fault:

- When a cable's conductor comes into contact with the ground

1.5 Fault detection methods

There are various types of fault location methods it can be classified as follows:

1. **Online Method**: To determine the fault points this methodology utilizes & processes the sampled voltages & current.
2. **Offline Method** : This methodology uses special instrument to check out service of cable within the field.

i. **Tracer Method**

In this methodology fault point in the cable lines is determine by walking on ground. The fault point detected by using signal like as audible signal or electromagnetic signal. It is used to point out fault location very accurately.

Example: Tracing current method, Sheath coil method

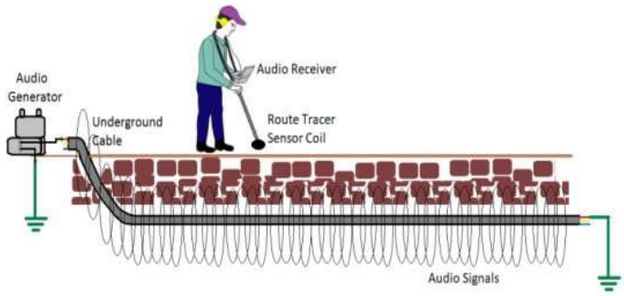


Fig:-19 Tracer

ii. **Terminal Method**

This method is used to determine fault location of cable from one or both ends without using tracing method.The general area of fault is located by the use of this method

Example: Murray loop method, Impulse current method

Special technique of fault location methods to find the exact location of the fault. The following are some of the methods used to locate faults in underground cables

Cable thumping method:

- The heat from this high current damages the insulation of the cable.
- This test damages can be reduced by limiting the power supply through the cable.
- However, sustained cable testing can deteriorate the insulation to an intolerable level
- This method cannot detect short circuit faults.

Fig:-20 Cable thumping method

Time-domain reflectometer(TDR) method:

- A time-domain reflectometer (TDR) is used to measure the cable length and to determine the location of joints, low-resistive faults and intermittent faults

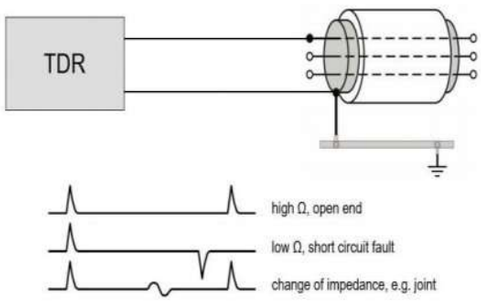


Fig:-21 Time-domain reflectometer

High voltage radar methods:

To overcome the limitations of TDR , the following high voltage radar methods are used.

- arc reflection method
- surge pulse reflection method
- voltage decay reflection method

1.6 Objective

The objective of this project is to determine the distance of underground cable fault from base station in kilometres. While a fault occurs for some reason, at that time the repairing process

related to that particular cable is difficult due to not knowing the exact location of the cable fault. The proposed system is to find the exact location of the fault. The project uses the standard concept of Ohms law i.e., when a low DC voltage is applied at the cable ends, then current would vary depending upon the location of fault in the cable. In case there is a short circuit (Line to Ground), the voltage across series resistors changes accordingly, which is then fed to inbuilt ADC of Arduino board to develop precise digital data for display in kilometre

1.7 Problem Statement

Underground power cables are essential for urban and industrial electricity distribution. However, identifying faults in these cables is challenging due to their hidden placement. Traditional methods for fault detection are time-consuming, expensive, and require extensive manpower.

This proposed system aims to develop an Arduino-based underground cable fault detection system that efficiently locates faults using a simple and cost-effective approach. The system will use Ohm's Law to measure voltage variations along the cable and determine the fault distance from the base station. By integrating sensors, an Arduino microcontroller, and a user-friendly interface, the system will provide real-time fault location data, reducing maintenance time and operational costs.

The proposed solution enhances the reliability of power distribution systems by allowing quicker fault identification and repair, minimizing power disruptions, and improving overall efficiency.

1.8 Operating principle

The underground cable protection using IoT based system works on principle of Ohm's Law, which states that the voltage across a conductor is directly proportional to the current flowing through it, provided the resistance remains constant. Mathematically, Ohm's Law is given by: $V=I*R$, where V is the voltage (volts), I is the current (amperes), and R is the resistance (ohms). In underground cables, when a fault occurs, such as a short circuit, open circuit, or insulation failure, the resistance of the cable changes, leading to variations in voltage and current. By continuously monitoring these electrical parameters, the system can detect faults and calculate the exact fault location based on the resistance per unit length of the cable.

CHAPTER-2

METHODOLOGY & CALCULATIONS

2.1 The System

2.1.1 Overview of the system

The system operates by continuously monitoring voltage levels across predefined cable sections. When a fault occurs (open circuit or short circuit), the voltage drops at the fault location, which is then analysed to calculate the distance of the fault from the main supply point.

2.1.2 Components used

Voltage Sensor (Resistors as Voltage Dividers): The circuit uses resistors to detect voltage drops across different cable segments. These resistors act as a voltage-sensing mechanism to identify faults.

Relay Module (12V Relays): Used to control and isolate faulty sections of the cable helps in switching between different cable segments.

ULN2003 IC (Relay Driver): Controls the relays based on fault detection signals from the Arduino.

Arduino Uno: Acts as the main microcontroller to process data from sensors and control outputs.

LED Indicators (Red, Yellow, Blue): Indicate the fault status of different cable sections (R+, Y+, B+).

LCD Display (16x2): Displays real-time fault location data

Buzzer:Provides an audio alert when a fault is detected.

2.2 Methodology And Approach

2.2.1 Module-1

Software Design:

The proteus software is used in implementing this project, The software design includes the design of electrical circuit in digital way and writing up the code in Arduino platform. For

simulating the circuit, we are utilizing the proteus software which includes different electrical components for designing. The IOT includes the work of simulation and software platform. For checking the conditions, the code is writeup on Arduino platform and is dumped in Arduino uno. Proteus is a versatile tool for electronic design and development.

2.2.2 Module – 2

Hardware Design

The hardware design includes the practical working of circuit after verified in software design. It can be achieved by connecting all the hardware components on a box structure and dumping the code in arudino uno and checking the results. The results of hardware design are similar to software design, but the difference is after verifying the results in simulation, the hardware is designed for practical applications.

2.2.3 Block Diagram:

This is the central processing unit of the system. It's responsible for coordinating all activities, including data collection from sensors, processing that data, and managing outputs.

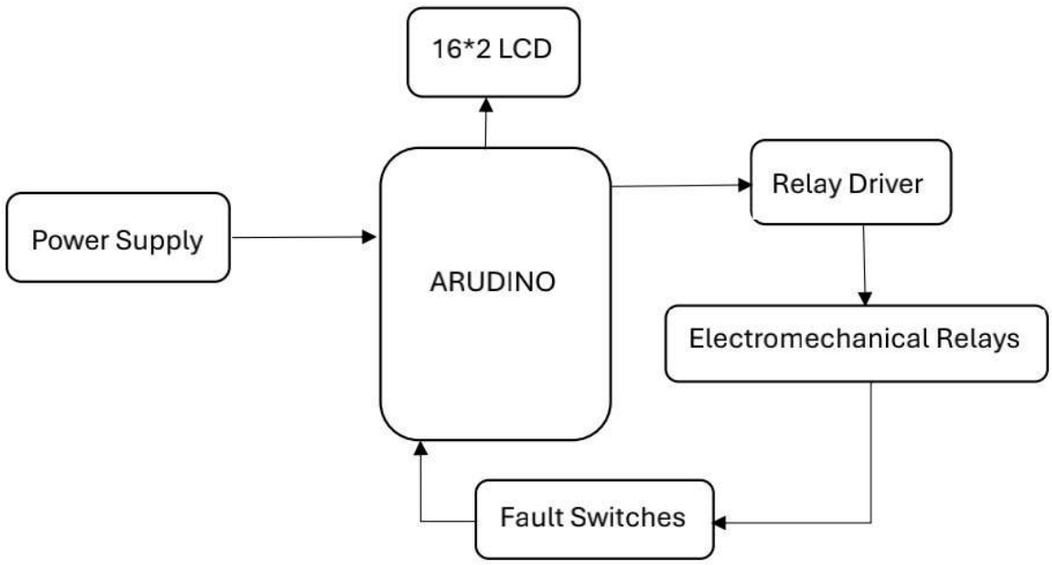


Fig:-22 Block diagram

Each component is designed to detect the fault location, providing comprehensive monitoring capabilities, The data processed by the Arduino Uno is displayed here, allowing for real-time monitoring of the fault at particular location

2.4 Line Diagram

The single line diagram illustrates the basic working of an underground cable fault detection system. The system begins with an AC power supply, which is converted into DC using a rectifier. This DC power is transmitted through underground cables to the connected load. To monitor the health of the cable, a detector unit is connected at a strategic point along the cable. In the event of a fault, such as a short circuit or open circuit, the detector unit identifies the fault by analyzing changes in voltage and current. These values are sensed using a voltage/current sensor, which sends the data to a control circuit. Based on the calculated resistance and standard cable properties, the system determines the approximate fault location. If a fault is detected, the relay trip unit can disconnect the faulty section to prevent further damage, while the alarm unit alerts the user through sound or light signals. Additionally, an optional digital display unit can be used to show the exact distance to the fault, enabling quick and precise maintenance action.

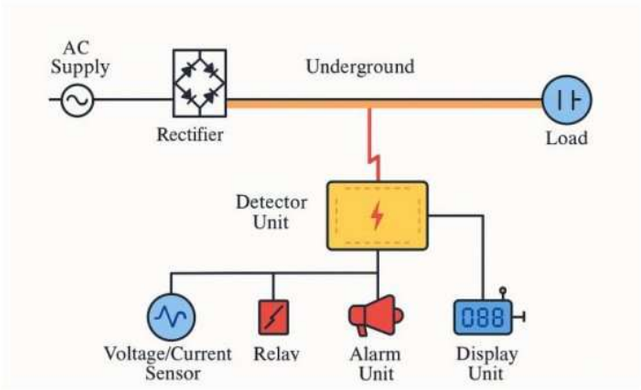


Fig:- 23 line diagram

2.5 Calculations

2.5.1 Sample Theory

The location of fault in the DC- cables can be determined by the following method in which the magnitude of the current is measured at the base station at the time of fault and a fixed voltage is applied at the base station. With the help of the current and voltage we can calculate the resistance in the cable at the time of fault by the ohm's law

$$V = I*R \text{ volts}$$

$$R = V/I \text{ ohms}$$

Let this be the resistance at the time of fault

$$R_{cal} = x \text{ ohms}$$

The standard resistance of the cable is given as the rating of the cable,

Let this be the standard resistance of the cable

R_{st} = y ohms/m

And by comparing both the calculated resistance (R_{cal}) and standard resistance (R_{cal}) we can calculate the fault location. This can be clearly understood by looking at the example below

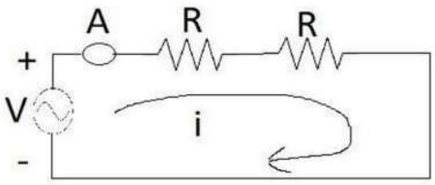


Fig: Circuitry representation of dc cable fault

For example:

The following values are represented for the explaining purposes

Pre fault voltage (applied voltage) = 5 v

When the fault occurs Voltage drop due to fault V_{drp} = 2.5 v

Current flowing through cable I_{cal} = 250 amp

Then the calculated resistance $R_{cal} = V_{drp}/I_{cal}$ = 0.01 ohm

As per cable rating

The standard resistance R_{st} = 0.1 ohm/km

But the obtained resistance R_{cal}= 0.01 ohm

The standard resistance is 0.1 ohm per 1 kilometre and by comparing R_{st} and R_{cal}, the approximate fault distance can be found out

1 km $\Longrightarrow$ 0.1 ohm

Distance $\Longrightarrow$ 0.01 ohm

Distance = (1 km * 0.01 ohm) / 0.1 ohm meters

Distance = 100 meters

So the distance of the DC - cable fault from the base station according to the above mentioned data is 100 meters away from the base station.

The above data taken from this reference ***”**IJRESM_V3_I3_105 Journal Paper of Underground Cable Fault Location Detection by Simple Calculations**”***

2.5.2 Theoretical Calculations

To determine the location of a fault in an underground cable, the following values were considered:

- Applied Voltage (V) = 5 Volts
- Voltage Drop due to Fault ,V_{drp} = 0.42 Volts
- Current ,I = 4.2mA = 0.0042A
- Standard Resistance of Cable , R_{st} = 100 Ohms per kilometer

Objective: To determine the distance of the fault when the voltage drop is 0.42 V

Let us assume the current I flowing through the cable remains the same as previous calculations (for comparative purposes), or is measured using sensors. But here, we aim to compute distance directly using:

$$R_{\text{cal}} = \frac{V_{\text{drp}}}{I}$$

distance directly from resistance, and we know:

$$\text{Distance (km)} = \frac{R_{\text{cal}}}{R_{\text{std}}}$$

We can rewrite as:

$$\text{Distance (km)} = \frac{V_{\text{drp}}}{I \times R_{\text{std}}}$$

If we assume current , $I = 4.2\ mA = 0.0042\text{A}$,
same as previous example, then:

$$R_{\text{cal}} = \frac{0.42}{0.0042} = 100\Omega$$

Then, the fault distance is:

$$\text{Distance(km)} = \frac{100}{100} = 1\text{km}$$

1 km $\Longrightarrow$ 10 ohm

Distance $\Longrightarrow$ 10 ohm

So the distance of the DC - cable fault from the base station according to the above mentioned data is 1kilo meters away from the base station.

2.5.3 ADC Calculations

Arduino uses a 10-bit ADC, which means:

- 0V → ADC value = 0
- 5V → ADC value = 1023

The resolution is:

$$\frac{5V}{1024} = 0.00488V (\text{or} 4.88\text{mVperstep})$$

So, the ADC value(digital reading) is calculated as:

$$\text{ADC Value} = \frac{\text{Input Voltage} \times 1023}{5}$$

Assuming each 1Km segment corresponds to a voltage drop

1km $\Longrightarrow$ 0.42 V

2km $\Longrightarrow$ 0.36 V

3km $\Longrightarrow$ 0.31 V

4km $\Longrightarrow$ 0.28 V

At 1km,

$$\text{ADC Value} = \frac{0.42 \times 1023}{5} = 85.93$$

At 2km,

$$\text{ADC Value} = \frac{0.36 \times 1023}{5} = 73.65$$

At 3km,

$$\text{ADC Value} = \frac{0.31 \times 1023}{5} = 63.42$$

At 4km,

$$\text{ADC Value} = \frac{0.28 \times 1023}{5} = 57.28$$

So, As the distance increases, both analog voltage and ADC values drops proportionally.

CHAPTER-3

MODULE-1: SOFTWARE DESIGN

3.1 Introduction

The Proteus Design is a software tool is used primarily for electronic design automation (EDA). It is a Windows application for Schematic capture, Simulation, and Printed Circuit Board (PCB) layout design. It can be found in many configurations, depending on the size of designs being produced and the microcontroller simulation. The software is used mainly by electronic design engineers and technicians to create schematics and electronic prints for manufacturing PCBs. It also has features that allows to virtually simulate IoT projects. Download the Proteus version 8.13 Pro software from the link below:

https://drive.google.com/file/d/10LhQ0JHKHTNStOKeo9wX3g_kR1uyl_8L/view?usp=drive_link

Unzip the files and move them to an appropriate folder.

3.2 Arudino IDE

The Arduino IDE is an open-source software, which is used to write and upload code to the Arduino boards. The IDE application is suitable for different operating systems such as Windows, Mac OS X, and Linux. It supports the programming languages C and C++. Here, IDE stands for Integrated Development Environment. The program or code written in the Arduino IDE is often called as sketching. We need to connect the Genuine and Arduino board with the IDE to upload the sketch written in the Arduino IDE software. Download the Proteus version 8.13 Pro software from the link below:

https://drive.google.com/file/d/1KXit4vZyiivEQVqe8PmymEEmx7Y4IQ9/view?usp=drive_link

3.2.1 Programming

Board Manager: If need to add a new board to the Arduino Software, Install the relate core and manage it.

Variables: Understand how to define and use variables in a Sketch.

Functions: Learn how to define and use functions in a Sketch.

Library: Using and installing Arduino Libraries.

3.2.2 Funtions

Digital Read (): Reads the value from a specified digital pin, either HIGH or LOW.

Digital Write (): Write a HIGH or a LOW value to a digital pin.

Pin Mode (): Configures the specified pin to behave either as an input or an output.

2.3 Flow Chart

This flowchart assumes a basic understanding of programming and sensor interfacing with Arduino.

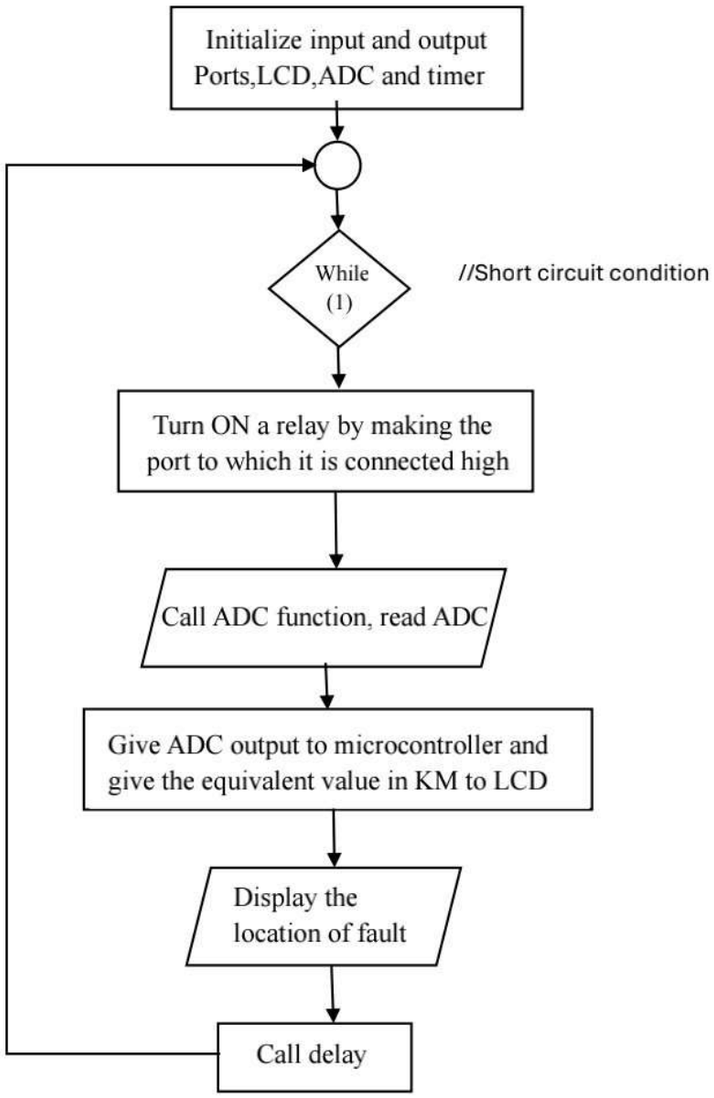


Fig:-24 Flowchart

3.2.3 Code

```
#include <LiquidCrystal.h>
LiquidCrystal lcd(2,3,4,5,6,7);
#define sensor A0
#define relay1 8
#define relay2 9
#define relay3 10
#define buzzer 13
int read_ADC;
int distance;
byte symbol[8] = {
      B00000,
      B00100,
      B00100,
      B00100,
      B11111,
      B01110,
      B00100,
      B00000};
void setup() {
pinMode(sensor,INPUT);
pinMode(relay1, OUTPUT);
pinMode(relay2, OUTPUT);
pinMode(relay3, OUTPUT);
pinMode(buzzer, OUTPUT);
lcd.createChar(1, symbol);
lcd.begin(16, 2);
lcd.clear();
lcd.setCursor(0, 0); // set the cursor to column 0, line 2
lcd.print("Welcome to Cable");
lcd.setCursor(0, 1); // set the cursor to column 0, line 2
lcd.print("Fault  Detection");
delay(2000);
lcd.clear();
}
void loop(){
lcd.setCursor(1,0);
lcd.print("R");
lcd.write(1);
lcd.setCursor(7,0);
lcd.print("Y");
lcd.write(1);
lcd.setCursor(13,0);
lcd.print("B");
lcd.write(1);

digitalWrite(relay1,HIGH);
digitalWrite(relay2,LOW);
digitalWrite(relay3,LOW);
delay(500);
data();
lcd.setCursor(0,1);
if(distance>0){lcd.print(distance); lcd.print("KM  ");}
```

```
else{lcd.print(" NF ");}
digitalWrite(relay1,LOW);
digitalWrite(relay2,HIGH);
digitalWrite(relay3,LOW);
delay(500);
data();
lcd.setCursor(6,1);
if(distance>0){lcd.print(distance); lcd.print("KM  ");}
else{lcd.print(" NF ");}
digitalWrite(relay1,LOW);
digitalWrite(relay2,LOW);
digitalWrite(relay3,HIGH);
delay(500);
data();
lcd.setCursor(12,1);
if(distance>0){lcd.print(distance); lcd.print("KM  ");}
else{lcd.print(" NF ");}
}
void data(){
read_ADC = analogRead(sensor);
distance = read_ADC/100;
if(distance>9)distance = 0;
if(distance>0){
digitalWrite(buzzer,HIGH);
delay(200);
digitalWrite(buzzer,LOW);
delay(200);
 }
}
```

3.3 PROTEUS SOFTWARE

3.3.1 Introduction

Proteus software, designed by Labcenter Electronics Ltd., is a comprehensive toolset primarily used for creating schematics, simulating electronics and embedded circuits, and designing PCB layouts. This module allows users to create schematics and simulate electronic circuits. It operates under ideal conditions, meaning it can be lenient in circuit design, such as not requiring pull-up resistors for simulations.

Used for designing PCB layouts, ARES helps in transitioning from a simulated circuit to a physical board design.

Proteus can simulate a wide range of electronic components and microcontrollers, including Arduino, PIC, and 8051, which is particularly useful for testing programming codes before hardware implementation. After ensuring the circuit works perfectly in the simulation, Proteus ARES can be used to design the PCB, which can then be manufactured.

It's widely used by engineering students and professionals for learning and designing electronic circuits due to its user-friendly interface and robust simulation capabilities. Proteus is valuable for its ability to test circuits in a controlled environment before moving to real-

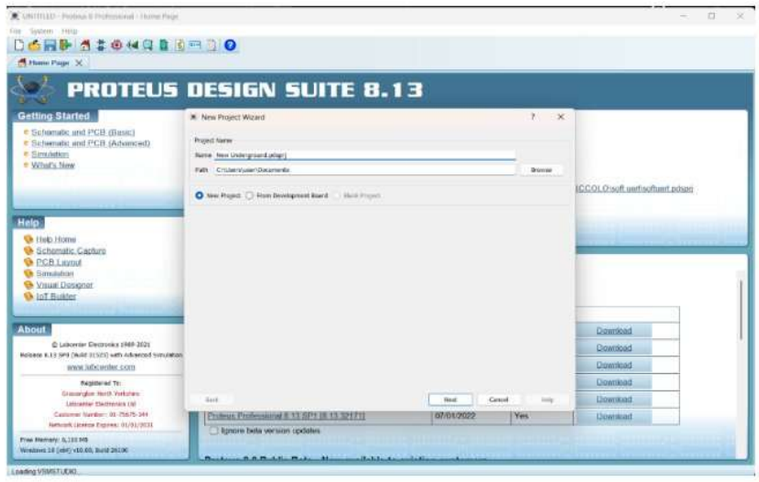
world applications, reducing the risk of errors and component damage during the development phase.

Proteus PCB Software combines Schematic Capture and PCB Layout modules to provide an affordable, powerful and easy to use suite of tools for professional PCB design. Proteus Virtual System Modelling (VSM) blends mixed-mode SPICE simulation with world leading fast microcontroller simulation. It enables rapid prototyping of both hardware and firmware designs, in software.

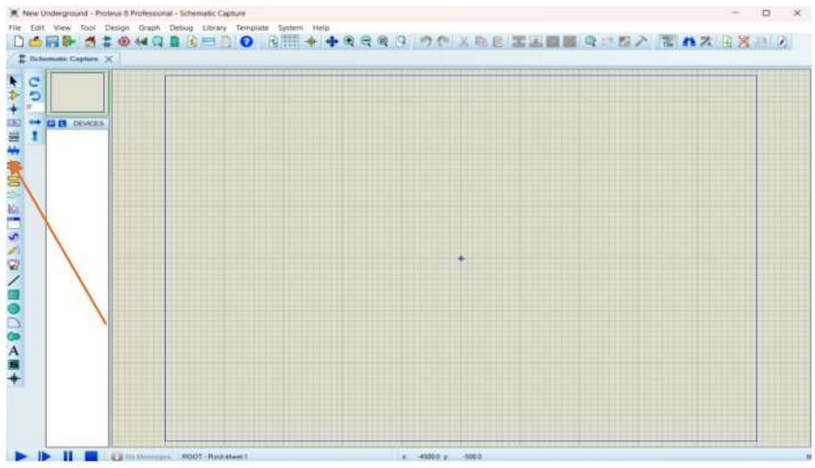
Proteus PCB Software combines Schematic Capture and PCB Layout modules to provide an affordable, powerful and easy to use suite of tools for professional PCB design. Proteus Virtual System Modelling (VSM) blends mixed-mode SPICE simulation with world leading fast microcontroller simulation. It enables rapid prototyping of both hardware and firmware designs, in software.

3.3.2 STEPS TO DESIGN PROTEUS CIRCUIT:

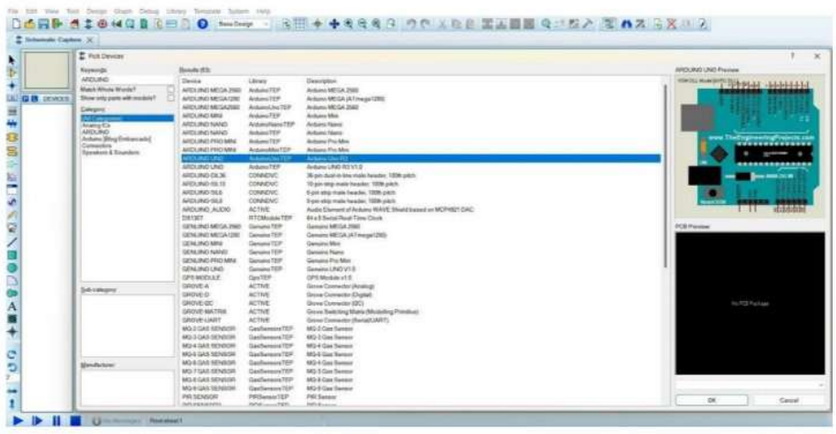
STEP-1: Create a new project in proteus with title "Underground Fault Detection using Arduino"



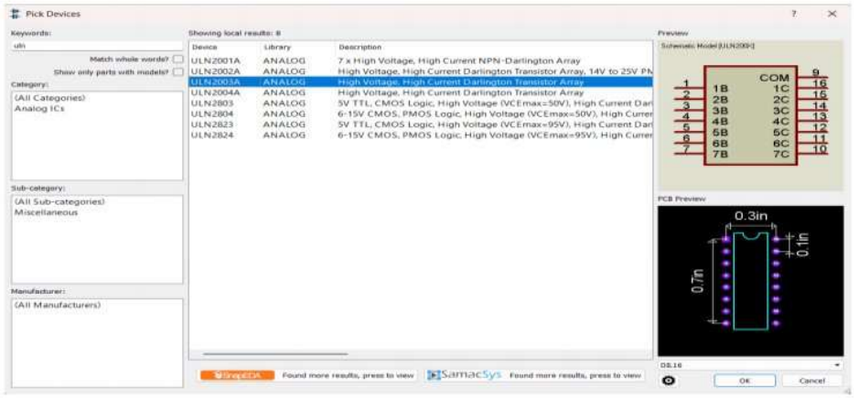
STEP-2: Included a part clicking on 'P' beside the toolbar



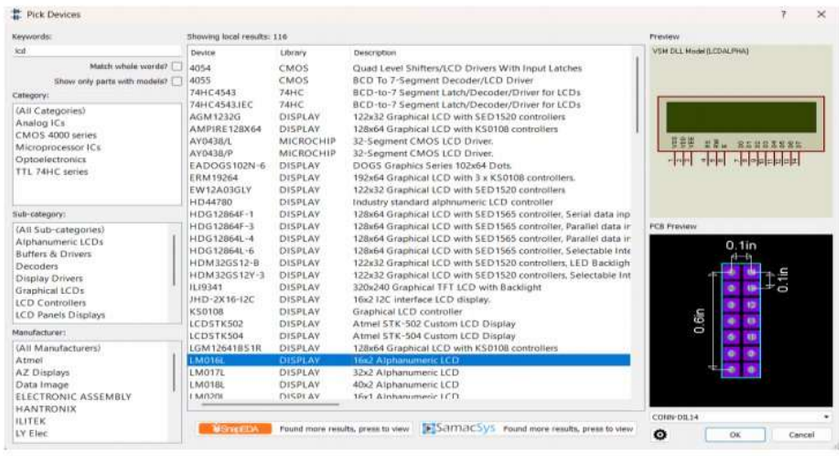
STEP-3: Added ‘Arduino Uno R3’ component as shown in below figure



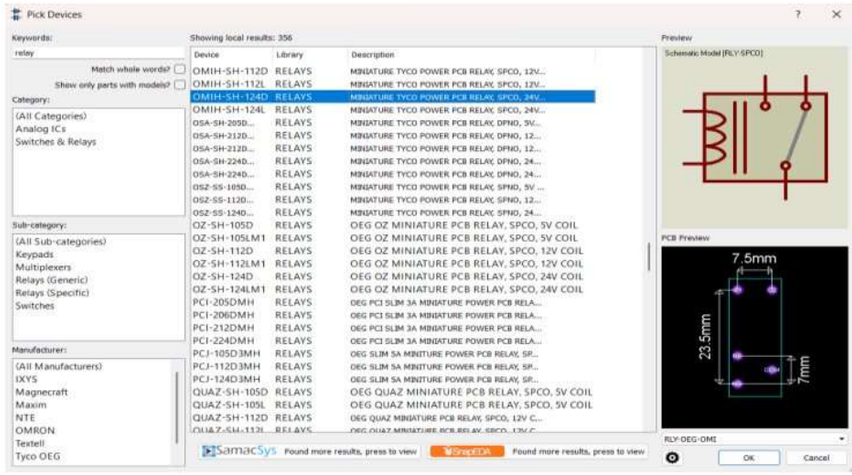
STEP-4: Added ‘ULN2003A’ as shown in below figure



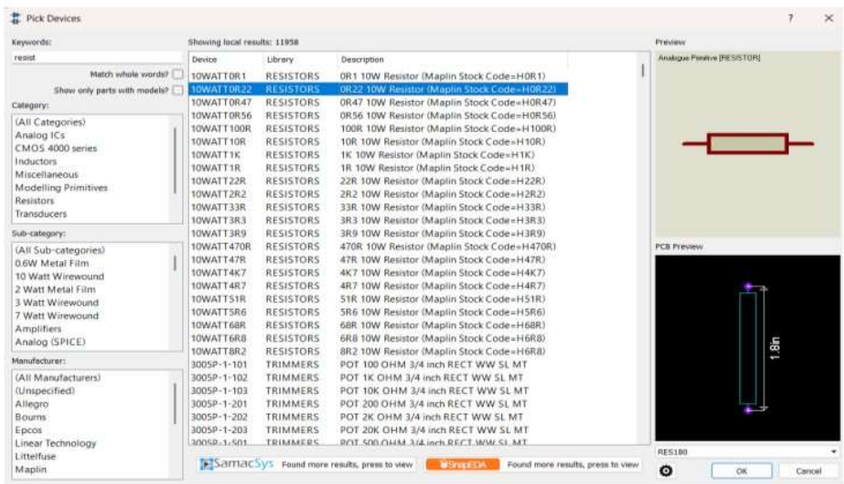
STEP-5: Added ‘16*2 LCD Display’ as shown in below figure



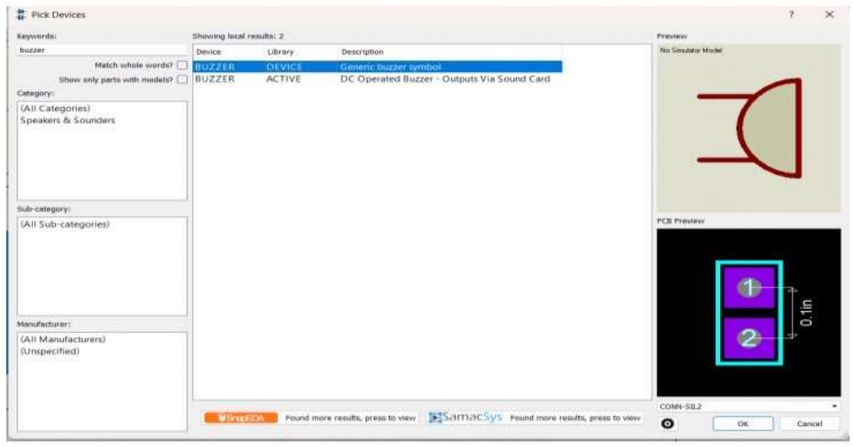
STEP-6: Add 'Relay' as shown in below figure



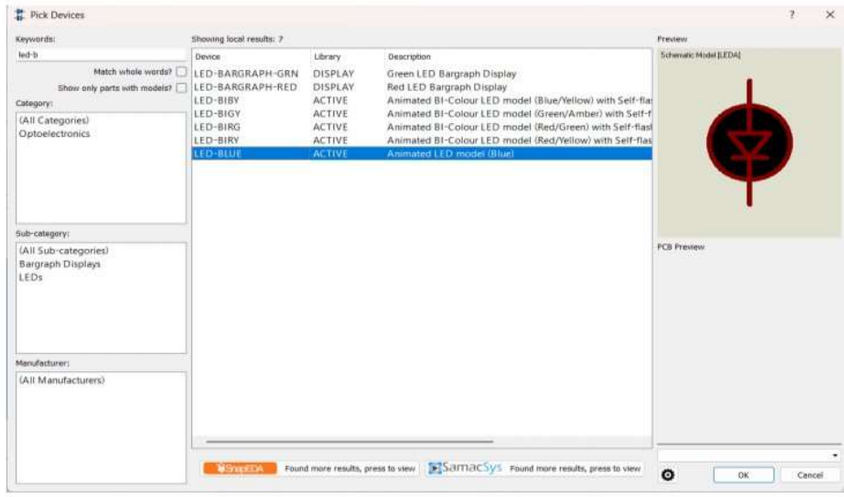
STEP-7: Add 'Resistors ' as shown in below figure



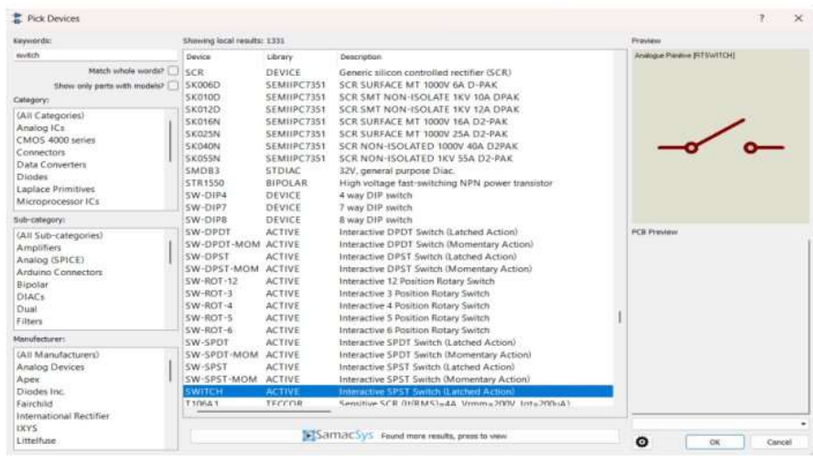
STEP-8: Add 'Buzzer' as shown in below figure



STEP-9: Add 'LED(blue, red, yellow)' as shown in below figure



STEP-10: Add 'Switch' as shown in below figure

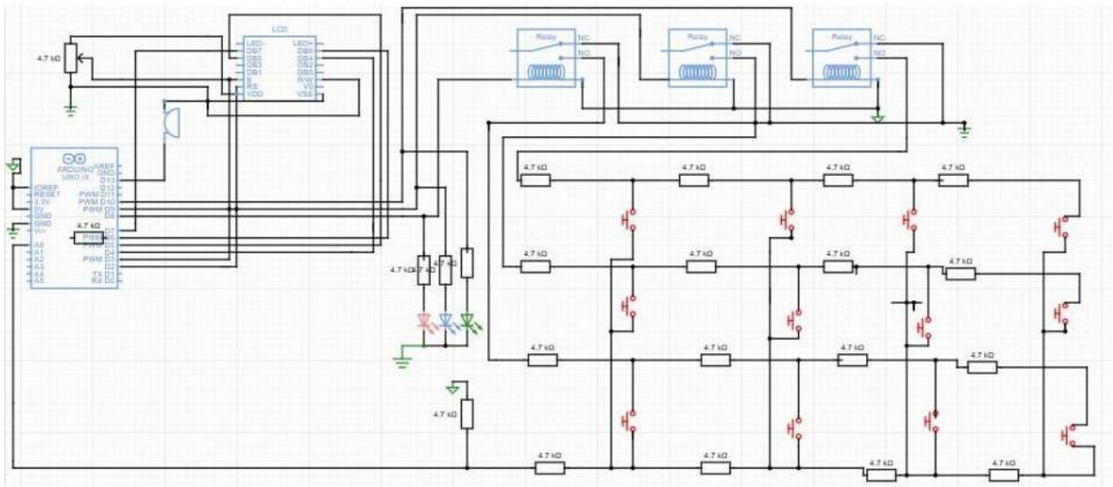


STEP-11: Open Arduino IDE, write the code given below to display the Fault status on LCD display.

STEP-12: Compile the code and copy path to the hex file, select the path for its hex file by clicking the Arduino UNO in proteus software

STEP-13: Finally run the simulation to observe the output of this project

3.4 Wiring Diagram



3.5 Simulation Circuit:

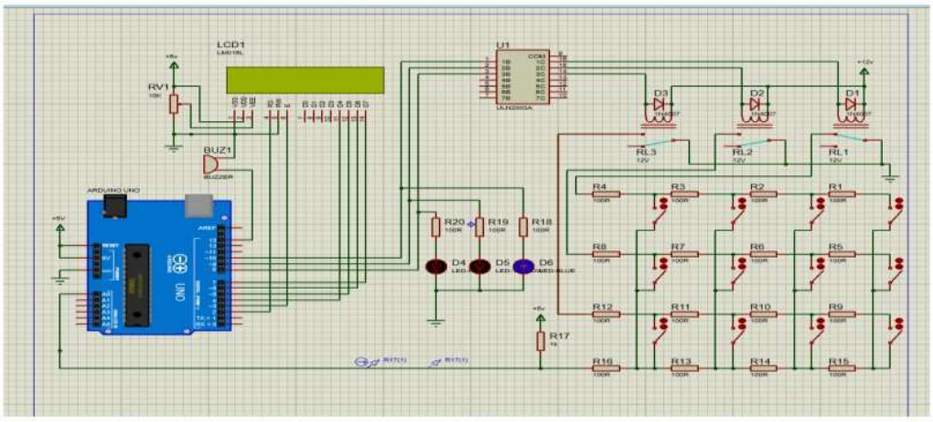


Fig:-25 Simulation Circuit

Download and install Proteus software. After installation, Refer to a block diagram for guidance and use the libraries to connect sensors, Arduino Uno, Relays ,ULN2003A, LED, Resistors, etc., correctly in simulation design. For simulating the circuit, we are utilizing the proteus software which includes different electrical components for designing. The IOT includes the work of simulation and software platform. For checking the conditions, the code is writeup on Arduino platform and is dumped in Arduino UNO. once simulation design is complete. Run the simulation in the Arduino IDE to check for the required output.

3.6 Proteus Working

The working principle of the underground fault detection system relies on continuous measurement of current flow through underground cables and detection of anomalies that indicate faults. The system operates as follows:

Normal Operation: In normal operating conditions, current flows uniformly through the underground cables with small variations.

Fault Occurrence: When a fault (e.g., short circuit or insulation failure) occurs, the current in the faulty portion varies significantly from normal readings..

Fault Location Identification: Based on the variations detected, the Arduino runs an algorithm that estimates the fault location by comparing measured and calculated calculated current readings at different points in the network.

Fault Indication and Display: When the fault is detected, the system energizes fault-indicating LEDs for the faulty section and displays the fault location on the LCD and virtual terminal in the Proteus simulation environment.

Alert Mechanism: When the data is processed, the Arduino triggers an alert mechanism if any anomalies are detected. The alert may be in the form of displaying the information on an I2C LCD Display, providing real-time Monitoring

Continuous Monitoring: This system is all about continuous monitoring, analysis, and interpretation of underground cable health parameters. Using real-time data transmission, the system enhances the reliability.

CHAPTER-4

MODULE-2 HARDWARE IMPLEMENTATION AND DESIGN

4.1 Introduction

Another way of execution of our project is Hardware implementation and design. This includes the brief introduction of hardware components and the working of the components.

By using the required components, we had constructed a circuit for the operation. The hardware circuit is designed by taking minimum values which for practical operation. For real time application the practical values can be multiplied according to the demands.

4.2 Hardware Description

- Arduino Uno
- Relay Module
- ULN2003A
- SIM800L Module
- Trimpot potentiometer
- 16*2 LCD display
- LED Bulbs
- Buzzer
- Jumper Wires

4.2.1 ARDUINO UNO

4.2.1.1 Overview

The Arduino Uno is an open-source microcontroller board based on the ATmega328P. It operates at 5V with a recommended input voltage range of 7-12V and features 14 digital input/output pins, 6 of which can be used as PWM outputs, alongside 6 analog inputs. This board offers a modest 32 KB of flash memory, 2 KB of SRAM, and 1 KB of EEPROM. Its programming is done through the Arduino IDE, employing a simplified version of C++, simply connect options include USB for programming and serial communication, along with AC-to-DC adapter or battery for external power supply.

Instead, it features the ATmega16U2 (ATmega8U2 up to version R2) programmed as a USB to serial converter.

Revision3 of the board has the following new features: pinout: added SDA and SCL. pins that are near to the AREF pin and two other new pins placed near to the RESET

Fig:-26 Arduino Uno

pin, the IOREF that allow the shields to adapt to the voltage provided from the board. In future, shields will be compatible both with the board that use the AVR, which operate with 5V and with the Arduino Due that operate with 3.3V. The second one is a not connected pin, that is reserved for future purposes.

At mega 16U2 replace the 8U2."Uno" means one in Italian and is named to mark the upcoming release of Arduino 1.0.

4.2.1.2 Schematic Design of Arduino UNO

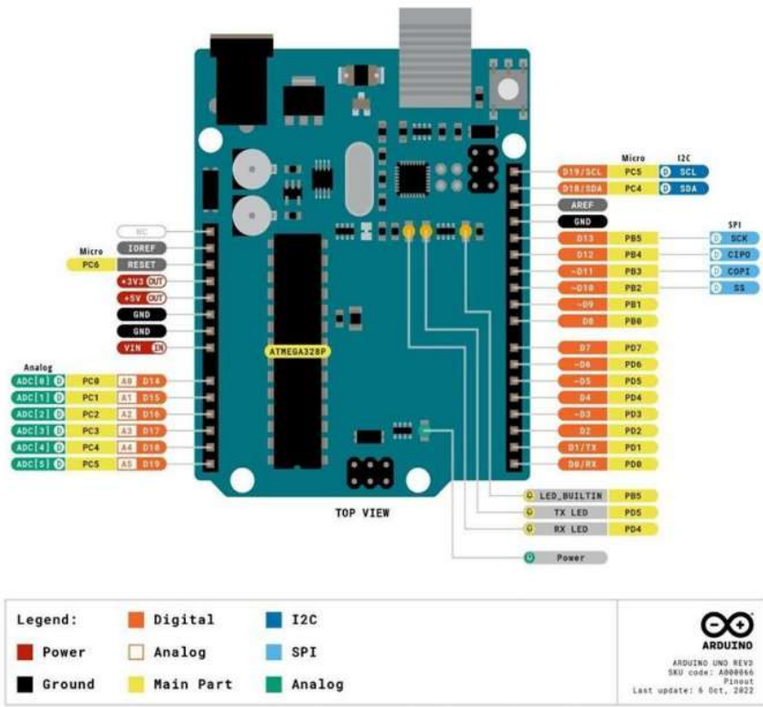


Fig:-27 Schematic design of Arduino

4.2.1.3 Summary

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
PWM Digital I/O Pins	6
Analog Input Pins	6
DC Current per I/O Pin	20 mA
Flash Memory	32 KB (ATmega328P) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Clock Speed	16 MHz
LED_BUILTIN	13

4.2.1.4 Input and Output Pins

Each of the 14 digital pins on the Uno can be used as an input or output, using pin Mode (), digital Write (), and digital Read () functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50k Ohms

In addition, some pins have specialized functions:

Analog Input Pins (A0 - A5):

The Arduino Uno has 6 analog input pins labelled A0 through A5. These pins can read analog voltage levels from sensors or other devices.

Digital Pins (D0 - D13):

There are 14 digital pins on the Arduino Uno labelled D0 through D13. These pins can be used for both input and output. Additionally, some of them support PWM (Pulse Width Modulation) for analog output.

Communication Pins:

Pins 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega81/2 USB-to-TTL Serial chip.

External interrupts:

Pins 2 and 3. These pins can be configured to trigger an interrupt on a low value, a

rising or falling edge, or a change in value. See the attach Interrupt () function for details.

PWM: Pins 3, 5, 6, 9, 10 and 11. Provide 8-bit PWM output with the analog Write () function

SPI: Pins 10 (SS). 11 (MOSI), 12 (MISO). 13 (SCK). These pins support SPI using the SPI library communication using the SPI library.

LEDs: There are built-in LEDs connected to pins 13 (built-in LED) and pin 9 (LED for TX/RX communication).

AREF: The analog reference voltage for the analog inputs

Power Pins:

-5V: This pin provides a regulated 5V output.

- 3.3V: This pin provides a regulated 3.3V output.

- Vin: This pin can be used to power the board from an external power source (7-12V).

RESET: Bring this line LOW to reset the microcontroller. Typically used to add reset button to shields which block the one on the board.

4.2.1.5 Communications

The Arduino Uno has several facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An ATmega16U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The '16U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, on Windows, a.inf file is required. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer (but not for serial communication on pins 0 and 1). A Software Serial library allows for serial communication on any of the Uno's digital pins. The ATmega328 also supports I2C (TWI) communication. The Arduino software includes a Wire library to simplify use of the 12C bus.

4.2.1.6 Programming:

The Arduino Uno is programmed using the Arduino IDE, where sketches (programs) are written in C/C++. These sketches are then compiled into machine-readable code. Through a USB connection, the compiled code is uploaded to the Arduino Uno's microcontroller, allowing it to execute the programmed instructions. This process enables users to create a variety of projects, from simple to complex, by controlling the board's inputs and outputs. The IDE simplifies coding with pre-defined functions and libraries tailored for the Arduino platform. The ATmega328 on the Arduino uno comes returned with a bootloader that allows to upload new code to it without the use of an external hardware programmer.

4.2.2 Relays Module:

A relay module is a circuit board that switches high-power circuits using low-power signals. A relay is an electrical switch that can be used to control devices and systems that use higher voltages. In the case of module relay, the mechanism is typically an electromagnet. The relay module input voltage is usually DC. However, the electrical load that a relay will control can be either AC or DC, but essentially within the limit levels that the relay is designed for. A relay module is available in an array of input voltage ratings: It can be a 3.2V or 5V relay module for low power switching, or it can be a 12 or 24V relay module for heavy-duty systems.

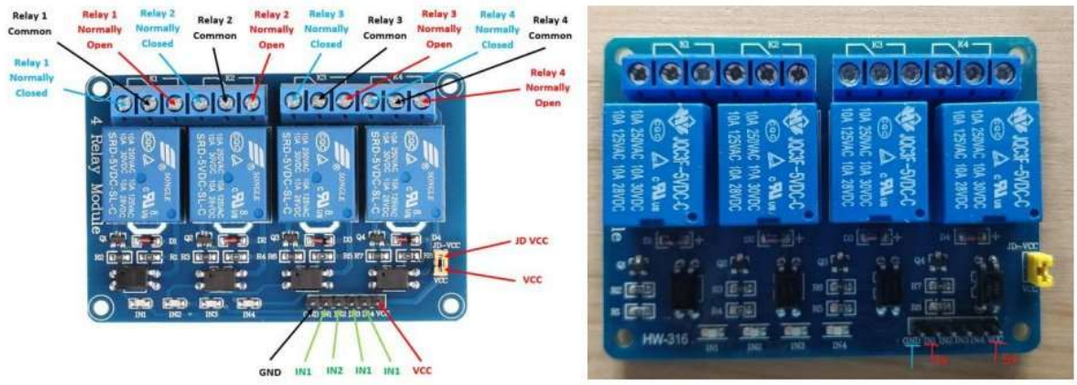


Fig:-28 Relay Module

The relay module information is normally printed on the surface of the device for ready reference. This includes the input voltage rating, switch voltage, and current limit

4.2.2.1 Table: - Technical Data

Supply voltage	3.75V to 6V	
Trigger Current	5mA	
Current when relay is active	~70mA (single)	~300mA (all four)
Relay maximum contact voltage	250VAC	30VDC
Relay maximum current	10A	
Size	73mm x 54mm x 19.5mm (LxWxH)	
Power Supply Voltage	5VDC, 12VDC	
Relay Type	Single Pole Double Throw (SPDT)	

4.2.2.2 Characteristics

- Its switching speed is Typically 5–10ms for mechanical relays.
- Each relay consumes ~70–100mA when active
- Uses optocouplers for electrical separation between the control circuit and the load

4.2.2.3 Applications

- Home Automation Systems
- Robotics and DIY Projects
- Internet of Things (IoT) Applications

4.2.3 ULN2003A

The ULN2003A devices are high-voltage, high-current Darlington transistor arrays. Each consists of seven NPN Darlington pairs that feature high-voltage outputs with common-cathode clamp diodes for switching inductive loads.

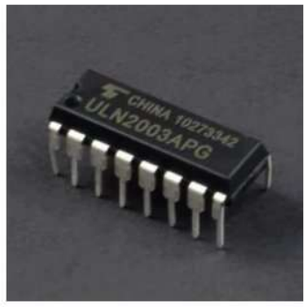


Fig:-29 ULN2003A

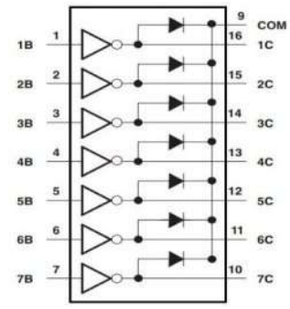


Fig:-30 Pin diagram of ULN2003A

ULN2003 is a 16 Pin IC, consisting of 7 Darlington pairs (each pair protected with suppression diode) and thus has the capability to handle a maximum of 7 loads(could be inductive). Each Darlington pair can handle a maximum 500mA load, while the peak value is 600mA.

4.2.3.1 Table: - Technical Data

Transistor Polarity	NPN
Collector- Emitter Voltage VCEO Max	50 V
Maximum DC Collector Current	500 mA
Mounting Style	Through Hole
Minimum Operating Temperature	- 20 C
Maximum Operating Temperature	+ 85 C
Product Type	Darlington Transistors
Height	4.59 mm
Length	20 mm
Width	7.1 mm
Unit Weight	1.600 g

4.2.3.2 Characteristics

- Voltage Rating: Can handle up to 50V

- Current Rating: Can sink up to 500mA per channel
- Darlington Pair Configuration: Provides high gain
- Output Clamp Diodes: Protect against voltage spikes from inductive loads

4.2.3.3 Applications

- Relay drivers
- Lamp drivers
- Stepper motor drivers
- Logic buffers
- Line drivers

4.2.4 Trimpot Potentiometer

A trimpot or trimmer potentiometer is a small potentiometer which is used for adjustment, tuning and calibration in circuits.

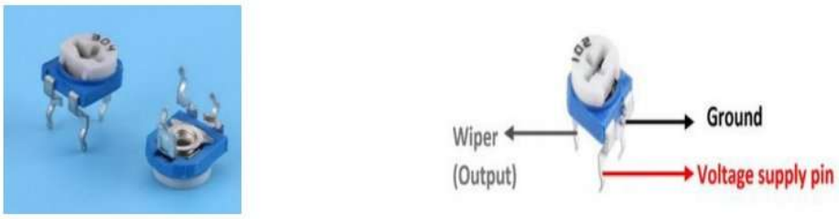


Fig:-31 Trimpot Potentiometer

When they are used as a variable resistance (wired as a rheostat) they are called preset resistors. Trimpots or presets are normally mounted on printed circuit boards and adjusted by using a screwdriver.

The material they use as a resistive track varies, but the most common is either carbon composition or cermet. Trimpots are designed for occasional adjustment and can often achieve a high resolution when using multi-turn setting screws.

4.2.4.1 Table: - Technical Data

Resistance:	10 Kohm
Power Rating:	500 mW (1/2 W)
Tolerance:	10 %
Number of Turns:	25
Temperature Coefficient:	100 PPM / C
Operating Temperature:	-55~125C
Dimensions:	9.53 x 4.83 x 10.03 mm
Rotational Life:	200 cycles
Number of Pins:	3
Adjustment:	Top Slot

4.2.4.2 Characteristics

- More precise voltage or current control in a circuit.
- Lower sensitivity, meaning a small rotation results in a relatively large change in resistance.
- Less abrupt changes in electrical parameters when adjusting the trimpot.

4.2.4.3 Applications

- Voltage & Current Adjustment
- Sensor Calibration
- Audio & Signal Processing
- Timing & Oscillator Circuits
- Motor & Servo Control
- Power Supply Regulation

4.2.5 LCD DISPLAY

16×2 LCD is an Alphanumeric display that can show up to 32 characters on single screen. You can display more characters by scrolling the texts one by one. The I2C Module is used to reduce the no. of pins needed for the display. The LCD Display can be used with all Microcontroller boards like 8051, AVR, Arduino, PIC, and ARM Microcontrollers. Most projects require an LCD display to communicate with the user in a better way. Some projects required to display warnings, errors, Sensor values, State of the input and output device, Selecting different modes of operations, Time and date display, Alert message and many more. This will give the project a better view and its operation in a more visual way.

Fig:-32 16 * 2 LCD Display

A 16×2 LCD means it can display 16 characters per line and there are 2 such lines. In this LCD each character is displayed in 5×7 Pixel matrix. This LCD has two registers, namely, Command and Data. The data register stores the data to be displayed on the LCD. The data is the ASCII value of the character to be displayed on the LCD

4.2.5.1 Specifications:

- Display capacity: 16 Character x 2 row

- Display colour: Blue backlit
- Character size: 2.95 mm wide x 4.35 mm high
- Character pixels: 5 W x 7 H
- Voltage requirements: 5 VDC +/- 0.5V
- Current requirements: 2 mA @ 5 VDC
- Connection: 4-pin male header with 0.1″: spacing
- Communication: I2C
- Overall dimensions: 3.15 x 1.42 x 0.51 in (80 x 36 x 13 mm)
- Operating temperature range: 32 to +131 °F (0 to +55 °C)

4.2.5.2 Applications:

- Show control information of temperature, time or numbers/special characters
- Access all custom characters stored in ROM
- Debugging for microcontroller projects

4.2.6 LED Bulb

The long lead is the positive pin and the short lead will be the negative pin. 3, If you hold the LED sideways on and view through the plastic, there will be two parts, a small straight pin and a fatter 'L' shaped pin. The small pin is the positive side and the fatter 'L' shaped pin is the negative side.

Fig:-33 LED

This figure represents the standard representation of a diode, where the anode represents the positive side of the diode and the cathode represents the negative side of the diode. These arrows are essential in the symbol as it means the emission of light that makes a difference between the diode and LED

4.2.7 Buzzer

A buzzer or beeper is an audio signalling device, which may be mechanical, electromechanical, or piezoelectric (piezo for short). It typically consists of a coil of wire wound around a core, an armature, and a contact mechanism.

Fig:-34 Buzzer

When the current flows through the coil, it creates a magnetic field that attracts the armature, causing it to move and make contact with the mechanism, which in turn produces the buzzing sound.

4.2.8 Jumper Wires

A jump wire is an electrical wire, or group of them in a cable, with a connector or pin at each end, which is normally used to interconnect the components of a breadboard or other prototype or test circuit, internally or with other equipment or components, without soldering.

Fig:-35 Jumper Wires

Though jumper wires come in a variety of colours, the colours don't actually mean anything. This means that a red jumper wire is technically the same as a black one. But the colours can be used to your advantage in order to differentiate between types of connections, such as ground or power.

4.3 Circuit Design

4.3.1 Circuit Connections:

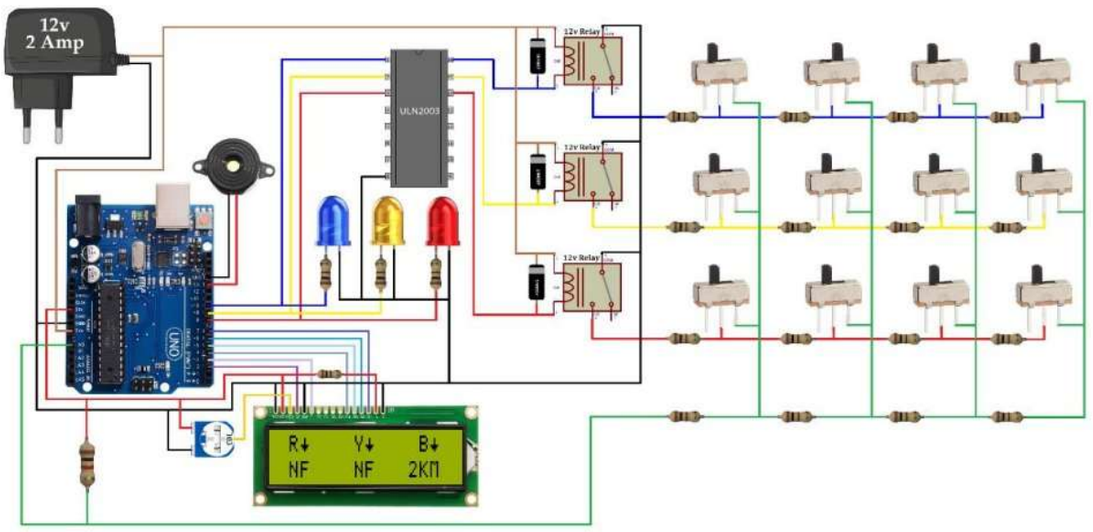


Fig:-36 Circuit Connections

- Connect your Arduino Uno board to your laptop using a USB cable.
- Open the Arduino IDE software on your laptop.
- In the Arduino IDE, go to the "Tools" menu and select "Board." Choose "Arduino Uno" from the list of available boards.
- Next, select the port to which your Arduino Uno board is connected. You can find this information in the "Tools" menu under the "Port" option. Select the appropriate port. Now, reset your Arduino Uno board.
- Connect your components to the Arduino Uno board according to the diagram provided. Compile the code provided in the previous response. Ensure there are no errors shown in the Arduino IDE.
- Upload the compiled code to your Arduino Uno board by clicking on the "Upload" button in the Arduino IDE
- After uploading the code, open the serial monitor in the Arduino IDE to check the output displayed. This will help you verify that the code is running correctly

4.3.2 Hardware Circuit Implementation:

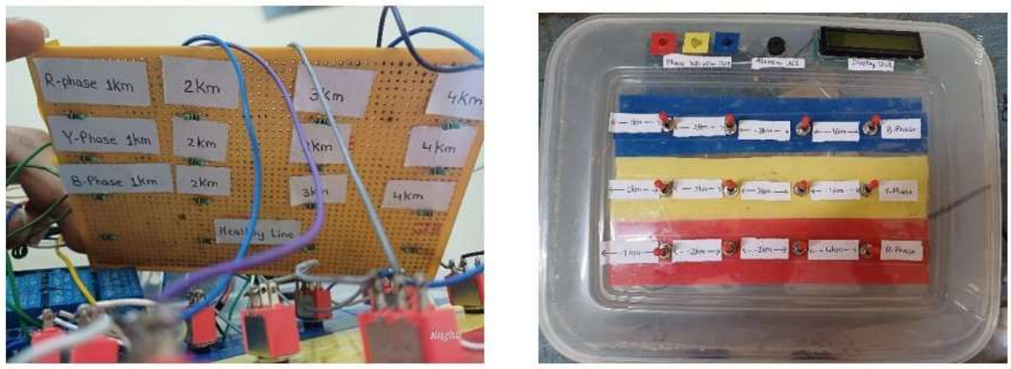


Fig:-37 Hardware Kit Implementation

4.3.3 Working of Hardware

The circuit connections are made by above circuit diagram with components Arduino, trimpot potentiometer, relays etc To set off faults manually withinside the kit, fault switches are used. About twelve fault switches are used which might be organized in four rows with every row having fault switches. The three rows constitute the three levels particularly R, Y and B. The fault switches: have 2 positions- No fault position (NF) and fault position (F).Each collection resistor represents the resistance of the underground cable for a selected distance and so right here four resistances in collection constitute 1-four kms. Value of every resistance is 10kΩ. One relay for every segment R, Y and B ,3 relays are used and the not unusual place factors of the relays are grounded. This 5V DC is used to deliver strength to the Arduino and the LCD. Power deliver to the LCD is given through Arduino. When fault is caused with the aid of using running any of the twelve switches, they impose situations like LG (Line To Ground), LL (Line To Line), LLG (Line To Line Ground) fault as in line with the transfer operation. As a end result of the fault, there's a voltage variations get created. This voltage variations are measured throughout the resistance is fed to the ADC of the Arduino. Using this variations, the Arduino computes the gap. Finally the gap of the fault from the bottom station is displayed in kilometres is LCD

CHAPTER-5

RESULTS AND DISCUSSION

The implementation of the underground cable fault detection system using Arduino yielded accurate and consistent results during testing. When a fault was introduced between cable segments, the system successfully identified the fault type (open or short circuit) and its approximate distance from the source. The resistance-based voltage drop method proved effective in locating faults with minimal error. Real-time data display on the LCD module enhanced usability and ensured quick fault recognition. The prototype was tested under varying conditions, and it consistently responded within a short time frame. Minor fluctuations were observed due to temperature variations and wire contact quality, but these did not significantly affect accuracy. The integration with Arduino allowed for flexible programming and easy modifications. Overall, the system demonstrated reliable performance, making it a promising low-cost solution for early fault detection in underground power cables

5.1 Theoretical Calculations Vs Practical Calculations

5.1.1 Under no fault condition

the following values were considered under the no fault condition:

- Applied Voltage (V) = 5 Volts
- Voltage Drop due to Fault ,V_{drp} = 0 Volts
- Current ,I = 4.2mA = 0.0042A
- Standard Resistance of Cable , R_{st} = 100 Ohms per kilometer

Objective: To determine the distance of the fault when the voltage drop is 0 V

Let us assume the current I flowing through the cable remains the same as previous calculations (for comparative purposes), or is measured using sensors. But here, we aim to compute distance directly using:

$$R_{\text{cal}} = \frac{V_{\text{drp}}}{I}$$

distance directly from resistance, and we know:

$$\text{Distance (km)} = \frac{R_{\text{cal}}}{R_{\text{std}}}$$

We can rewrite as:

$$\text{Distance (km)} = \frac{V_{\text{drp}}}{I \times R_{\text{std}}}$$

If we assume current , $I = 4.2\ mA = 0.0042\text{A}$

same as previous example, then

$$R_{\text{cal}} = \frac{0}{0.0042} = 0\Omega$$

Then, the fault distance is:

$$\text{Distance(km)} = \frac{0}{100} = 0\text{km}$$

5.1.2 Under fault condition

5.1.2.1 Sub case-1

At R-Phase

To determine the location of a fault in an underground cable, the following values were considered:

- Applied Voltage (V) = 5 Volts
- Voltage Drop due to Fault ,V_{drp} = 1.68 Volts
- Current ,I = 4.2mA = 0.0042A
- Standard Resistance of Cable , R_{st} = 200 Ohms per kilometer

Objective: To determine the distance of the fault when the voltage drop is 1.68 V

Let us assume the current I flowing through the cable remains the same as previous calculations (for comparative purposes), or is measured using sensors. But here, we aim to compute distance directly using:

$$R_{\text{cal}} = \frac{V_{\text{drp}}}{I}$$

distance directly from resistance, and we know:

$$\text{Distance (km)} = \frac{R_{\text{cal}}}{R_{\text{std}}}$$

We can rewrite as:

$$\text{Distance (km)} = \frac{V_{\text{drp}}}{I \times R_{\text{std}}}$$

If we assume current , $I = 4.2\ mA = 0.0042\text{A}$,
same as previous example, then:

$$R_{\text{cal}} = \frac{1.68}{0.0042} = 400\Omega$$

Then, the fault distance is:

$$\text{Distance(km)} = \frac{400}{200} = 2\text{km}$$

So the distance of the DC - cable fault from the base station according to the above mentioned data is 2 kilo meters away from the base station.

Theoretical Calculation	Practical Calculation
1.68	1.06

5.1.2.2 Subcase-2

At Y-Phase

To determine the location of a fault in an underground cable, the following values were considered:

- Applied Voltage (V) = 5 Volts
- Voltage Drop due to Fault ,V_{drp} = 1.68 Volts
- Current ,I = 4.2mA = 0.0042A
- Standard Resistance of Cable , R_{st} = 200 Ohms per kilometer

Objective: To determine the distance of the fault when the voltage drop is 1.68 V

Let us assume the current I flowing through the cable remains the same as previous calculations (for comparative purposes), or is measured using sensors. But here, we aim to compute distance directly using:

$$R_{\text{cal}} = \frac{V_{\text{drp}}}{I}$$

distance directly from resistance, and we know:

$$\text{Distance (km)} = \frac{R_{\text{cal}}}{R_{\text{std}}}$$

We can rewrite as:

$$\text{Distance (km)} = \frac{V_{\text{drp}}}{I \times R_{\text{std}}}$$

If we assume current , $I = 4.2\ mA = 0.0042\text{A}$,
same as previous example, then:

$$R_{\text{cal}} = \frac{6.32}{0.0042} = 1504.3\Omega$$

Then, the fault distance is:

$$\text{Distance(km)} = \frac{1504.3}{400} = 3.7\text{km}$$

So the distance of the DC - cable fault from the base station according to the above mentioned data is 2 kilo meters away from the base station.

Theoretical Calculation	Practical Calculation
6.32	6.28

5.1.2.3 Subcase-3

At B-Phase

To determine the location of a fault in an underground cable, the following values were considered:

- Applied Voltage (V) = 5 Volts
- Voltage Drop due to Fault ,V_{drp} = 3.78 Volts
- Current ,I = 4.2mA = 0.0042A
- Standard Resistance of Cable , R_{st} = 200 Ohms per kilometer

Objective: To determine the distance of the fault when the voltage drop is 3.78 V

Let us assume the current I flowing through the cable remains the same as previous calculations (for comparative purposes), or is measured using sensors. But here, we aim to compute distance directly using:

$$R_{\text{cal}} = \frac{V_{\text{drp}}}{I}$$

distance directly from resistance, and we know:

$$\text{Distance (km)} = \frac{R_{\text{cal}}}{R_{\text{std}}}$$

We can rewrite as:

$$\text{Distance (km)} = \frac{V_{\text{drp}}}{I \times R_{\text{std}}}$$

If we assume current , $I = 4.2\ mA = 0.0042\text{A}$,
same as previous example, then:

$$R_{\text{cal}} = \frac{3.78}{0.0042} = 900\Omega$$

Then, the fault distance is:

$$\text{Distance(km)} = \frac{900}{200} = 3\text{km}$$

So the distance of the DC - cable fault from the base station according to the above mentioned data is 2 kilo meters away from the base station.

Theoretical Calculation	Practical Calculation
3.78	3.53

5.2 Final Observation

The final observation based on the comparison of theoretical and practical voltage drop values in underground cable fault detection using IoT shows a strong alignment between expected and measured results. Theoretically calculated voltage drops, based on cable resistance and fault distance, closely match the practical readings obtained from the IoT-

based detection system. Slight discrepancies are observed, which can be attributed to real-world factors such as temperature variations, wire aging, and contact resistance at joints. Despite these minor differences, the system demonstrates high accuracy in identifying fault locations. This confirms that IoT-enabled monitoring provides a reliable and efficient approach to detect voltage drop anomalies and pinpoint faults in underground cables.

Cases	Theoretical Calculation	Practical Calculation	Distance
At R phase	1.68	1.06	2Km
At Y phase	6.32	6.28	4Km
At B phase	3.78	3.53	3Km

5.3 Software Results

5.3.1 No-Fault Condition in Under ground cable fault detection Using Simulation:

This output is taken from software design.

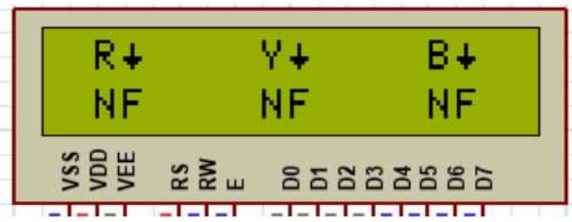


Fig 38: - Proteus Simulation Results under No-Fault Condition

Table: simulation results

S.No.	Phase	Buzzer	FAULT	Fault Distance
1.	R	No-response	No-fault	0 Km
2.	Y	No-response	No-fault	0 Km
3.	B	No-response	No-fault	0 Km

Subcase-1:

5.3.1.1 Fault Condition in Under ground cable fault detection Using Simulation at R-phase:

In this condition, a fault occurs in the underground cable, leading to a sudden change in voltage and resistance. The system detects the fault and determines the distance of the fault location

When phase R creates the fault at 2Km

This output is taken from software design.

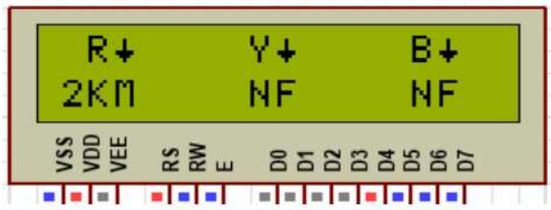


Fig 39: - Proteus Simulation Results under Fault Condition

Table: simulation results

S.No.	Phase	Buzzer	FAULT	Fault Distance
1.	R	On-State	Short circuit fault	2 Km
2.	Y	No-response	No-fault	0 Km
3.	B	No-response	No-fault	0 Km

Subcase-2

5.3.1.2 Fault Condition in Under ground cable fault detection Using Simulation at Y-phase:

When phase Y creates the fault at 4Km

This output is taken from software design.

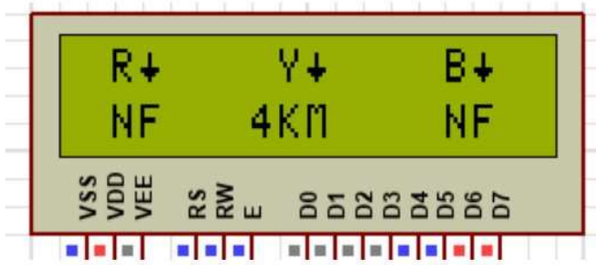


Fig 40: - Proteus Simulation Results under Fault Condition

Table: simulation results

S.No.	Phase	Buzzer	FAULT	Fault Distance
1.	R	No-response	No-fault	0 Km
2.	Y	On-State	Short circuit fault	4 Km
3.	B	No-response	No-fault	0 Km

Subcase-3

5.3.1.3 Fault Condition in Under ground cable fault detection Using Simulation at B-phase:

When phase B creates the fault at 3Km

This output is taken from software design.

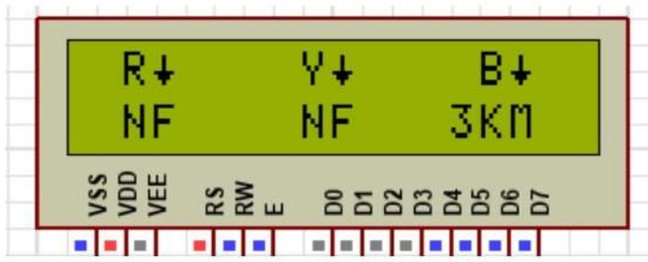


Fig 41: - Proteus Simulation Results under Fault Condition

Table: simulation results

S.No.	Phase	Buzzer	FAULT	Fault Distance
1.	R	No-response	No-fault	0 Km
2.	Y	No-response	No-fault	0 Km
3.	B	On-State	Short circuit fault	3 Km

5.4 Hardware Results

5.4.1 No-Fault Condition in Under ground cable fault detection using Hardware design

In this no-fault condition, indicates that the cable is operating normally without any breaks, short circuits. In this state, the voltage across the cable remains stable, and there is no sudden change in resistance. The Arduino microcontroller continuously monitors the voltage and displays "Cable Status: No Fault Detected" on an LCD screen. However, if a fault occurs, the system detects the voltage drop or resistance change and calculates the fault location using Ohm's Law, displaying the details on the LCD

Fig:-42: - Hardware results under No-Fault Condition

Subcase-1

5.4.1.1 Fault Condition in Under ground cable fault detection using Hardware design at R-phase

In a fault condition, the underground cable experiences a break, short circuit, or insulation failure, leading to an abnormal voltage drop or a sudden change in resistance. The Arduino microcontroller continuously monitors these parameters, and upon detecting a fault, it calculates the exact fault location using Ohm's Law (V = IR). The system then displays the fault details on an LCD screen, showing the fault distance, This ensures efficient fault detection, reducing downtime and improving maintenance accuracy.

Fig 43:- Hardware results under Under R Phase Fault Condition

Subcase-2

5.4.1.2 Under Fault Condition in Under ground cable fault detection using Hardware design at Y-phase

In a fault condition on the Y phase of an underground cable, issues such as a break, short circuit, or insulation failure cause a sudden drop in voltage or a significant change in resistance. These abnormalities are continuously monitored by the Arduino microcontroller, which reads the analog signals through sensors connected to the cable. When a fault is detected, the system calculates the exact distance of the fault using Ohm's Law (V = IR), based on the measured voltage drop and known resistance of the cable per kilometer. The calculated fault location is

then displayed on an LCD screen, indicating which phase is affected and how far the fault has occurred. This real-time monitoring enables technicians to act quickly and repair only the damaged section of the cable. By identifying the fault accurately on the Y phase, the system avoids unnecessary digging and saves both time and cost.. Overall, this method ensures fast, reliable, and efficient underground cable fault detection and enhances the reliability of power distribution systems.

Fig 43: - Hardware results under Under Y-Phase Fault Condition

Subcase-3

5.4.1.3 Under Fault Condition in Under ground cable fault detection using Hardware design at B-phase

In a fault condition on the B phase of an underground cable, a short circuit, break, or insulation failure can lead to an abnormal voltage drop or resistance fluctuation in the line. The system continuously monitors the electrical parameters of each phase using sensors connected to a microcontroller like Arduino. When a fault is detected on the B phase, the system calculates the fault distance using Ohm's Law (V = IR) by analysing the voltage drop and the known resistance per kilometer of the cable. This fault location is then shown on an LCD display, specifically indicating the B phase and the distance in kilometers where the fault occurred. The use of IoT technology allows this information to be transmitted to a remote server for real-time monitoring. This helps maintenance teams quickly identify and fix the exact point of damage without unnecessary delays or manual searching. Accurate detection on the B phase reduces power loss, downtime, and repair costs. Overall, the system provides a smart and effective solution for underground cable fault detection in modern power distribution networks.

Fig 44: - Hardware results under Under B Phase Fault Condition

5.5 Comparison of Results

5.5.1 Comparison Between Simulation And Hardware Results under no fault condition:

Table: No-Fault Condition in Underground cable fault detection Using both Simulation and Hardware Modules:

S.No.	Phases	Buzzer		Fault		Fault type
		Simulation	Hardware	Simulation	Hardware	
1.	R	Off	Off	NF	NF	--
2.	Y	Off	Off	NF	NF	--
3.	B	Off	Off	NF	NF	--

The above table presents a comparison of results from both simulation and actual hardware measurements for underground cable fault detection using IoT. These results include fault location, fault type, resistance, and voltage readings.

The resistance and voltage values recorded in both the simulation and hardware measurements show slight variations due to real-world factors such as wire resistance, environmental conditions, and sensor accuracy. The buzzer are triggered only when a fault is detected, confirming the effectiveness of the system. The comparison highlights the system's accuracy and practical implementation in detecting faults in underground cables

5.5.2 Comparision Between Simulation And Hardware Results at R - phase:

Table: Under Fault Condition in Underground cable fault detection Using both Simulation and Hardware Modules:

S.No.	Phases	Buzzer		Fault		Fault type
		Simulation	Hardware	Simulation	Hardware	
1.	R	On	On	2Km	2Km	Short circuit fault
2.	Y	Off	Off	NF	NF	--
3.	B	Off	Off	NF	NF	--

The above table presents a comparative analysis of simulation and hardware results for underground cable fault detection using IoT under fault conditions. This comparison highlights the system's accuracy and reliability, demonstrating its effectiveness in detecting and locating faults in underground cables.

5.5.3 Comparision Between Simulation And Hardware Results at Y - Phase:

Table: Under Fault Condition in Underground cable fault detection Using both Simulation and Hardware Modules:

S.No.	Phases	Buzzer		Fault		Fault type
		Simulation	Hardware	Simulation	Hardware	
1.	R	Off	Off	NF	NF	--
2.	Y	On	On	4Km	4Km	Short circuit fault
3.	B	Off	Off	NF	NF	--

In the case of a fault condition in the Y phase of an underground cable, both hardware and software components play crucial and interconnected roles in detecting and managing the fault. The hardware part consists of sensors, a microcontroller (like Arduino), relays, switches, buzzers, and an LCD display. When a fault such as a short circuit or break occurs in the Y phase, voltage and current sensors pick up abnormal signals like sudden voltage drops or resistance changes. The relays are triggered to isolate the faulty phase, and the buzzer sounds an alert to indicate the fault physically. Simultaneously, the LCD displays which phase is affected and the calculated distance of the fault.

On the software side, the program running on the microcontroller continuously reads data from the analog sensors. It uses Ohm's Law (V = IR) to calculate the fault distance based on voltage drop and known cable resistance. The software also controls the LCD output, manages the timing of relays and buzzers, and—if connected—sends data to cloud platforms for remote monitoring via IoT. In the case of a Y phase fault, the software logic identifies the problem phase and determines the precise fault location in kilometers, which helps technicians take fast and focused action.

5.5.4 Comparision Between Simulation And Hardware Results at B-Phase:

Table: Under Fault Condition in Underground cable fault detection Using both Simulation and Hardware Modules:

S.No.	Phases	Buzzer		Fault		Fault type
		Simulation	Hardware	Simulation	Hardware	
1.	R	Off	Off	NF	NF	--
2.	Y	Off	Off	NF	NF	--
3.	B	On	On	4Km	4Km	Short circuit fault

The comparison table between simulation and hardware results highlights the accuracy and reliability of the underground cable fault detection system under fault conditions. During testing, the R and Y phases showed no faults in both simulation and hardware, confirmed by the buzzer remaining off and the fault distance displaying "NF" (No Fault). This consistency verifies the system's ability to correctly identify healthy cable lines. However, in the B phase, both simulation and hardware detected a fault at a distance of 4Km, and the buzzer was activated in both cases, signalling an active fault condition. The type of fault identified was a short circuit fault, showcasing that the system successfully recognized the specific nature and location of the problem. The uniformity between simulation and hardware outputs demonstrates that the designed IoT-based system performs effectively under real-world conditions, validating its use for reliable underground cable monitoring.

6. Future scope

- AI-Based Fault Prediction – Machine learning can help predict faults before they occur, reducing failures.
- GPS-Enabled Location Tracking – More precise fault detection using GPS for accurate repair interventions.
- Wireless Sensor Networks (WSNs) – Eliminates wired connections for real-time fault monitoring.
- 5G and Edge Computing – Faster data transmission and improved system responsiveness.
- Self-Healing Smart Grids – Automatic rerouting of power to minimize downtime during faults.
- IoT and Cloud Integration – Remote monitoring and data analytics for better decision-making.

This represents opportunities for driving innovation, improving reliability, and optimizing asset performance in the electrical sector through the continued advancement using IoT technology.

7. Conclusion

This project presents The underground cable fault detection system using IoT is a crucial innovation for modern electrical infrastructure. By continuously monitoring cable parameters and detecting faults in real-time, this system significantly reduces downtime and maintenance costs. The integration of IoT allows remote monitoring, making fault detection more efficient and accessible. With further technological advancements, such as AI, GPS, and wireless networks, the system can become even more precise and reliable, ultimately improving the stability and efficiency of underground power distribution networks.

References

[1] Underground Cable Fault Location Detection by simple Calculations, International Journal of Research in Engineering,Science and Management Volume-3,Issue-3,March-2020 www.ijresm.com | ISSN (Online): 2581-5792

[2] Underground Cable Fault Detection Using IoT Based by Muskan Agrahari-2021, Journal of emerging technologies and innovative research

[3] Arduino - Uno Based Underground Cable Fault Detection System (AUCFDS) by S.R. Purohit , Sunilkumar M. Hattaraki, Soumya P.Hampangoudra, Rashmi Nimbaragi and Savita Mattihal

[4] Development of fault distance locator for underground cable detection To cite this article: FH Md Arifin et al 2020 J. Phys.: Conf. Ser. 1432 012014

[5] Underground Cable Fault Detector By Pratik Sutar 2021, International journal of engineering research and technology

[6] M.M. Amarnath, B. Ganesh Ram, Mr. V. Chandrasekaran, Mr. G.Mohan Ram, "IOT Based On Underground Cable Fault Identification Using Arduino" in April 2021

[7] Raushan Kumar Bhagat, Shrawan Suthar,Chandan Kumar, Sunil Kumar Mina, "Underground Cable Fault Detector Using Arduino And Gsm Modules" in July 2020.

[8] Locating Underground Cable Faults: A Review and Guideline for New Development Md. Fakhrul Islam, Amanullah M T Oo, Salah