NATIONAL INSTITUTE OF TECHNOLOGY AGARTALA

<u>Power Electronics Lab Project</u>

Topic: Underground Cable Fault Detection

Year: 3rd, Semester: 5th

Submitted By:

1. Debraj Bhattacharjee (21UEE014)
2. Rahul Banik (21UEE006)
3. Tanmay Das (21UEE011)
4. Swarnila Karmakar (21UEE015)

Submitted to: Mr. Debabrata Bhattacharjee

Certificate

This is to certify that the project titled:

"Underground Cable Fault Detection System"

has been successfully completed by the following students of 3rd Year / 5th Semester. This project involved the design, development, and implementation of an innovative Underground Fault Detection System using a hardware-based approach. The project incorporated Arduino microcontroller to monitor and detect faults in underground infrastructures.

I certify that this project is up to my expectations and as per the guidelines. I have gone through their report and it is assessed that no work has been performed on this topic so far. They have worked hard towards the completion of this project and I wish them success.

Signature (Project Supervisor)

ACKNOWLEDGEMENT

We are very grateful to Mr. Debabrata Bhattacharjee Sir of Electrical Engineering Department of NIT Agartala, for providing us academic inputs, guidance and encouragement throughout the working period of our project.

We would also like to express our heartfelt gratitude to Dr. Bikram Das Sir without whose permission and wise counsel we wouldn’t be able to pursue out work in this manner.

We would also like to thank all others who have directly or indirectly helped us in making this project a success.

Thank You.

INTRODUCTION:

In an era defined by technological innovation and infrastructure expansion, the effective monitoring and maintenance of critical underground networks have emerged as paramount concerns. Underground infrastructure, including pipelines, electrical cables, and water supply systems, often remains hidden from view, making the timely detection of faults and anomalies challenging. To address this, a hardware-based approach employing Arduino microcontrollers, relays, 16-bit displays, and resistive sensors has become a promising solution.

This report delves into the theoretical framework and working principles of such a system, designed to revolutionize the way we safeguard and manage our underground assets. By harnessing the power of hardware components and innovative sensor technology, this underground fault detection system offers a proactive and responsive approach to identifying faults, minimizing downtime, enhancing safety, and optimizing maintenance efforts.

In the following sections, we explore the intricacies of this hardware-driven system, highlighting its advantages, disadvantages, and potential applications in various sectors. As the demand for resilient and efficient underground infrastructure continues to grow, this system represents a crucial step forward in ensuring the reliability and longevity of these vital networks.

Aim:

Underground Cable Fault Detection System.

Materials Required:

Sl.No.	Name of Apparatus	Specifications	Quantity
1.	Veroboard	18*30	1
2.	Arduino Uno	-	1
3.	Relay	12v5 pin, 30Amps	3
4.	Rectifier Diode	1N4007	3
5.	IC Base	16 pin	1
6.	ULN2003	-	1
7.	Slide Switch	-	12
8.	Variable Resistor	10k	1
9.	Resistor	100	16
10.	Buzzer	12V	1
11.	LEDs	Blue, Yellow, Red	3
12.	Header	Male & Female	1
13.	Power Adaptor	12V, 12Amps	1
14.	Connecting wires	Single stranded	As required
15.	LCD Display	16*2	1

Circuit Diagram:

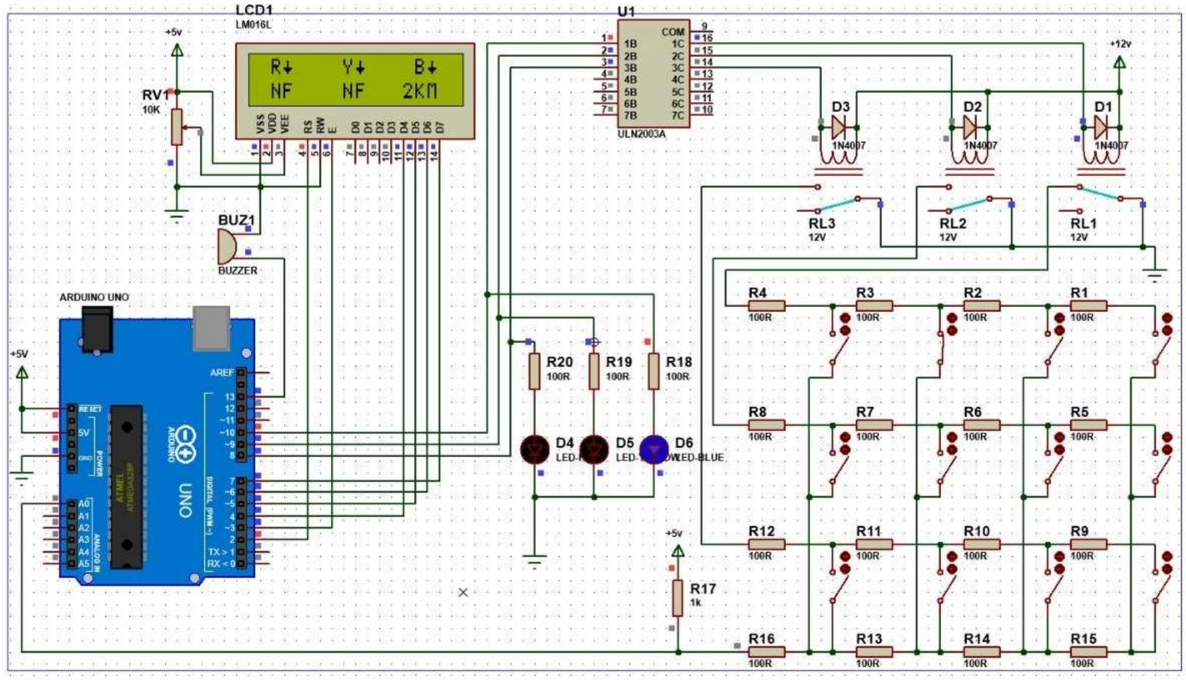
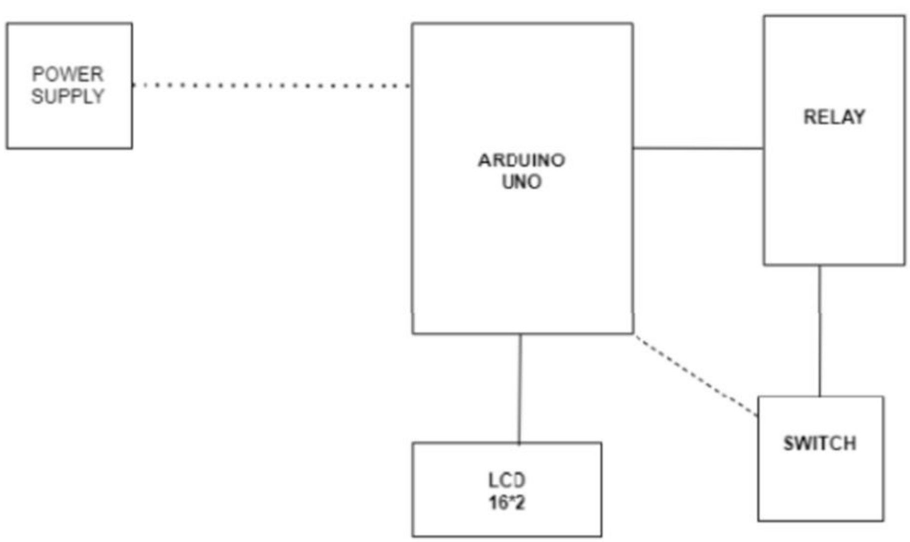


Fig.: Diagrammatic representation of an Underground Cable Fault Detection System In Proteus Application (Practically, we have used 117Ω)

Block Diagram:



Practical Operation:

In the project, Arduino Uno is acting as brain, it contains the program which is used to run the 16*2 LCD screen (each character is displayed in a 5×7pixel matrix), Relays ,Relay Driver and miniature the transmission line(Each phase line R,Y,B is represented by 4*117Ω resistor, the first resistor represents 7km and 2nd resistor represents 8km and 3rd resistor represent 9km). The relay is protected by diode. And the relay continuous ly switches for finding the fault in the transmission line and relay driver is used for the amplification of current . There is a senor wire connected to A0 pin of the Arduino for the fault detection.

Arduino Code:

```
#include <LiquidCrystal.h>
LiquidCrystal lcd(2,3,4,5,6,7);
// we have defined the relay pin and buzzer pin and sensor pin and variables
#define sensor A0
#define relay1 8
#define relay2 9
#define relay3 10
#define buzzer 13
int read_ADC;
int distance;
// custom characters are used for LCD screen
bytesymbol[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);
 }}
```

Working Principle:

1. **Resistive Sensors:** The system utilizes resistive sensors strategically placed along the underground infrastructure. These sensors monitor specific parameters like temperature, pressure, or conductivity, depending on the application. Resistance changes in these sensors correspond to variations in the monitored parameter.

2. **Data Acquisition:** The analog resistance values from the sensors are continuously collected and fed into analog input pins on Arduino microcontrollers. Arduino is chosen for its ease of interfacing with resistive sensors through analog-to-digital conversion.

3. **Threshold Comparison:** Within the Arduino microcontroller, acquired analog data is compared to predefined threshold values determined based on normal operating conditions. Deviations beyond these thresholds indicate potential faults or anomalies.

4. **Relay Activation:** Upon detecting a fault or anomaly, the Arduino triggers relay modules. These relays can be configured to execute specific actions like cutting off power, closing valves, or activating alarms. The action taken is determined by the type and location of the detected fault.

5. **16-bit Displays:** Real-time visual feedback is provided through 16-bit displays. These displays offer high-resolution graphical representations of the underground infrastructure, highlighting fault locations and types in fine detail. This resolution aids in precise fault localization.

6. **User Interface:** The 16-bit displays double as a user interface, allowing operators and maintenance personnel to configure system parameters, view historical data, and acknowledge fault alerts.

7. **Communication:** For remote monitoring and rapid response, the system can integrate communication modules (e.g., Wi-Fi, GSM, or LoRa) to transmit fault alerts to a central monitoring station or mobile devices.

8. **Data Logging:** Arduino microcontrollers can log fault data, including timestamps, sensor readings, and fault types. This data is valuable for post-analysis, maintenance planning, and record-keeping.

Component Description:

1. Veroboard:

A board of strip board and preformed circuit board material of copper strips on an on insulating bonded paper board.

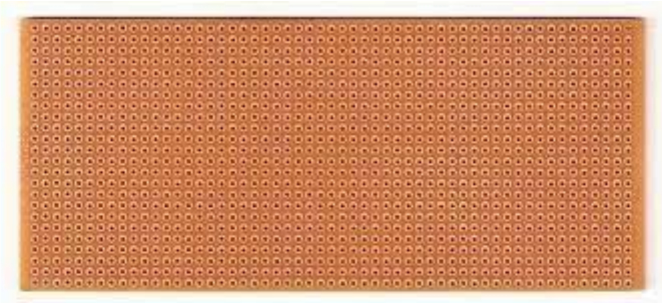


Figure 1-Veroboard

2. 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 ceramic resonator, a USB connection, a power jack, an ICSP header and a reset button.

Figure 2 -Veroboard

3. Relay:

A relay is an electrically operated switch that opens and closes circuits. Relays work by using a small electric current to turn on or off a larger electric current. The electromagnet in a relay is a coil of wire that becomes a temporary magnet when electricity flows through it

Figure 3-Relay

4. Rectifier Diode:

A rectifier diode is a semiconductor diode that is used in the rectifier bridge application to convert AC (alternating current) to DC (direct current).

Figure 4-Rectifier diode

5. IC Base:

This is used as a place holder for 40-pin IC chips as IC chips may get damaged from heat due to soldering.

Figure 5-IC BASE

6. ULN2003:

The ULN2003 is high-voltage high-current Darlington transistor arrays each containing seven open collector common emitter pairs. Each pair is rated at 500mA.

Figure 6-ULN2003

7. Slide Switch:

A slide switch is a mechanical switch that slides from the open (off) position to the closed (on) position and allows control of a circuit's current flow without having to manually splice or cut wire.

Figure 7-Slide Switch

8. Variable Resistor:

A variable resistor is a resistor of which the electric resistance value can be adjusted. A variable resistor is in essence an electro-mechanical transducer and normally works by sliding a contact (wiper) over a resistive element.

Figure 8-Variable Resistor

9. Resistor:

A passive electrical component with two terminals that are used for either limiting or regulating the flow of electric current in electrical circuits. The main purpose of resistor is to reduce the current flow and to lower the voltage in any particular portion of the circuit.

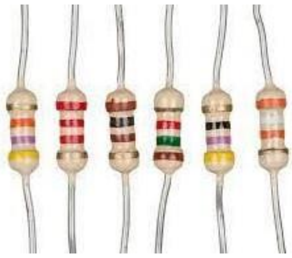


Figure 9-Resistor

10. Buzzer:

The piezo, also known as the buzzer, is a component that is used for generating sound. It is a digital component that can be connected to digital outputs, and emits a tone when the output is HIGH. Alternatively, it can be connected to an analog pulse-width modulation output to generate various tones and effects.

Figure 10-Buzzer

11. LEDs:

LED (Light Emitting Diode) is an electronic device, which emits light when the current passes through its terminals. LED's are used in various applications. It is also used as an ON/OFF indicator in different electronic devices.

Figure- LEDs

12. Header:

The name of the two lines of sockets that line up with the edges of the circuit board. These thin sockets allow you to plug wires into them.

Figure 12-Header

13. Power Adaptor:

The Adaptor must be DC (Direct Current) not AC (Alternating Current) The barrel plug must be centre positive (The middle pin of the plug has to be positive).

Figure 13-Power Adaptor

14. Connection Wire:

Jumper wires are used for making connections between items on your breadboard and your Arduino's header pins.

Figure 15-Connection Wire

15. LCD Display:

The LCDs have a parallel interface, meaning that the microcontroller has to manipulate several interface pins at once to control the display. The one we are using is a 16-bit display.

Figure 16-LCD Display

Benefits and Applications:

1. **Early Fault Detection**: Continuous monitoring detects faults in their early stages, minimizing downtime and potential hazards.

2. **Efficient Maintenance:** Real-time fault localization and data logging enable swift responses, reducing repair costs.

3. **Safety Enhancement:** Automated shutdowns and alarms enhance safety by preventing catastrophic failures and minimizing environmental risks.

4. **Cost Savings:** Proactive maintenance reduces overall repair expenses and extends the lifespan of underground assets.

Disadvantages:

1. **Complexity:** Building and configuring such a system can be complex, requiring expertise in electronics and programming. Maintenance personnel may need specialized training to operate and maintain the system effectively.

2. **Calibration and Maintenance:** Resistive sensors can require regular calibration to maintain accuracy. Additionally, sensors and other components may need periodic maintenance or replacement, adding to operational costs.

3. **Power Consumption:** The continuous operation of the system can consume a significant amount of power, especially if it's monitoring a large underground network 24/7. This may require a robust power supply or even backup power sources.

4. **Limited Scalability:** Expanding the system to monitor larger or more extensive underground networks may be challenging and costly. Adding more sensors and components can increase complexity and maintenance requirements.

Conclusion:

The hardware-based underground fault detection system described here, using Arduino microcontrollers, relays, 16-bit displays, and resistive sensors, offers a comprehensive solution for monitoring and maintaining underground infrastructure. Its ability to detect faults early, enhance safety, and improve maintenance efficiency makes it a valuable tool in various applications. The use of resistive sensors and careful threshold setting ensures precise fault detection, even in varying environmental conditions. As technology advances, further refinements in fault detection hardware are expected, making underground systems even more robust and sustainable.