CENTRALIZED UNDERGROUND CABLE FAULT DETECTION SYSTEM

By

2111CS050105	**G. KALYAN SAI GOUD**
2111CS050087	**BALTHU VARSHITH**
2111CS050092	**RAJAM PAVAN YADAV**
2111CS050079	**M. SHIVA RAJ**

Under the Guidance of

Mrs. D. Supriya

Assistant Professor

Department of Internet Of Things

School of Engineering

Malla Reddy University

Hyderabad, Telangana, INDIA

ABSTRACT

Underground cables are prone to a wide variety of faults due to underground conditions, wear and tear, rodents etc. Diagnosing fault source is difficult and entire cable should be taken out from the ground to check and fix faults. The project work is intended to detect the location of fault in underground cable lines from the base station in km using a controller. To locate a fault in the cable, the cable must be tested for faults. Whenever the fault occurs in underground cable it is difficult to detect the exact location of the fault for process of repairing that particular cable. The proposed system finds the exact location of the fault. The prototype is modeled with a set of resistors representing cable length in km and fault creation is made by a set of switches at every known distance to cross check the accuracy of the same. The fault occurring distance, phase, and time is displayed on a 16X2 LCD interfaced with the microcontroller

1. Introduction

1.1 Problem Definition & Description

- **Problem specification**
  - The existing system is unable to locate the exact fault position.
  - Functionality issues
  - Implementation cost
- **Problem description**
  - Faults in underground cables can lead to service disruptions, safety hazards, and costly repairs.
  - Detecting and locating faults in underground cables is challenging due to their inaccessible nature

1. Introduction

1.2 Objectives of the Project

- **Aim of the project**
  - The aim of this project is to develop a centralized underground cable fault detection system that can detect faults in underground cables and accurately locate them.
  - The system will improve the reliability and safety of underground cable networks by minimizing downtime for repairs and ensuring quick and precise fault detection.

- **Tasks and Deliverables**
  - Task 1: System Design
  - Task 2: Sensor Integration
  - Task 3: Communication System Development
  - Task 4: Fault Detection Algorithm
  - Task 5: System Intergration
  - Task 6: User Interface
  - Task 7: Documentation and Training
  - Task 8: System Development

1. Introduction

1.3 Scope of the project

- **Determining goals**
  - Fault Detection Accuracy
  - Fault Location Accuracy
  - Real-time Monitoring
  - Cost-effectiveness
  - Safety
- **Data Requirements**
  - Sensor Data
  - Cable Information
  - Environmental Data
  - Maintenance Records

1. Introduction

1.3 Scope of the project

- **Constraints**
  - Cost
  - Power Supply
  - Communication Range
  - Environmental Factors
  - Maintenance
  - Data Security

1. Introduction

1.3 Scope of the project

- **Workflow management strategies**
  - Requirement gatherings
  - Task Prioritization
  - Cross-Functional Teams
  - Continuous improvement
  - Regular Maintenance and Testing
  - Establish Communication Protocols

2. System Analysis

2.1 Existing System

➢ Background

- Existing system used in the past was the "thump test," where a high-voltage pulse was applied to the cable and the sound of the discharge was listened to with a ground microphone. This method could help locate faults by listening for the sound of the discharge at the fault location. Another method was the "time delay method," where the time taken for a signal to travel from one end of the cable to the other was measured. Any deviation from the expected time could indicate a fault. These methods were rudimentary compared to modern systems and relied heavily on manual labor and interpretation of results. They were also less accurate and efficient compared to modern sensor-based systems.

2. System Analysis

2.1 Existing System

➢ Literature Survey

- **"Fault Identification In Underground Cables Using IOT "**
- **"Presented Analysis of Underground Cable Fault Distance Locator"**
- **"Presented Underground Cable Fault Detector Using GSM"**

2. System Analysis

2.1 Existing System

➢ Limitations of Existing System

- Cost
- Complex installation
- Integration
- Device Compatibility
- Customization
- Lower efficiency

2. System Analysis

2.2 Proposed System

➢ Advantages of Proposed System

- Less maintenance
- It has higher efficiency
- Less fault occurs in underground cables
- Cost efficient
- This method is applicable to all types of cables ranging from 1kv to 500kv
- Operating range is large

2. System Analysis

2.3 Software & Hardware Requirements

➢ Software Requirements

- Programming languages like c, c++
- Development environment
- IFTT

➢ Hardware Requirements

- Arduino UNO
- LCD Display
- Relay
- Resistors
- Switches
- Buzzer
- IC
- Connecting Wires

2. System Analysis

2.4 Feasibility Study

- **Technical Feasibility**
  - Fault Detection Algorithm
  - Hardware and Infrastructure
  - Communication Systems
  - Sensor Technology
- **Economic Feasibility**
  - Cost-Benefit Analysis:
  - Return on Investment (ROI)
  - Cost Reduction Opportunities
  - Maintenance Costs

3. Architectural Design

3.1 Module Design

3.1.1 Module 1 (Sensors & Actuators)

- sensors for measuring parameters such as voltage, current, temperature, and vibration.

3.1.2 Module 2 (Central Hub Module)

- Receives data from sensors and processes it for fault detection.
- Contains fault detection algorithms for analyzing sensor data.

3.1.3 Module 3 (Microcontroller & Power)

- Yeah, here we are using Arduino UNO has a microcontroller and power bank as a power source

3. Architectural Design

3.2 Method & Algorithm Design

3.2.1 Fault Detection :

3.2.1.1 Method

- Detecting a cable Fault and giving a Buzzer sound

3.2.1.2 Algorithm

- 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

3. Architectural Design

3.2 Method & Algorithm Design

3.2.2 Fault Tracking:

3.2.2.1 Method

- Finding the exact location of the cable fault

3.2.2.2 Algorithm

- It uses resistors, and fault switches to locate faults in underground cables, the distance to the fault can be estimated using a method called impedance-based fault location

3. Architectural Design

3.2 Method & Algorithm Design

3.2.3 Alert System:

3.2.3.1 Method

- Sends an Alert to the base station

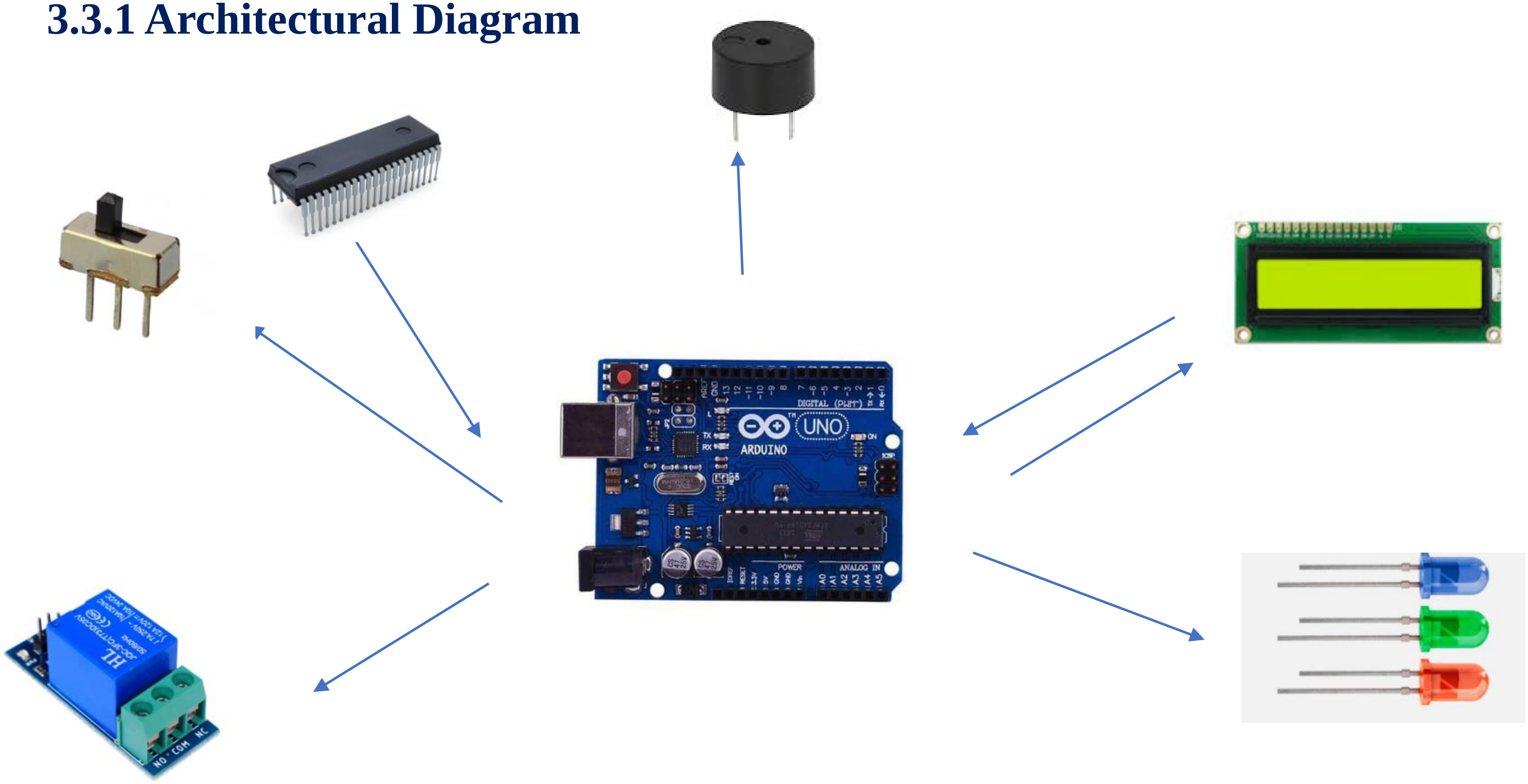
3.2.3.2 Algorithm

- When there is a fault in the cable, It sends an alert message to the base station
- And it sends the location of the fault

3. Architectural Design

3.3 Project Architecture

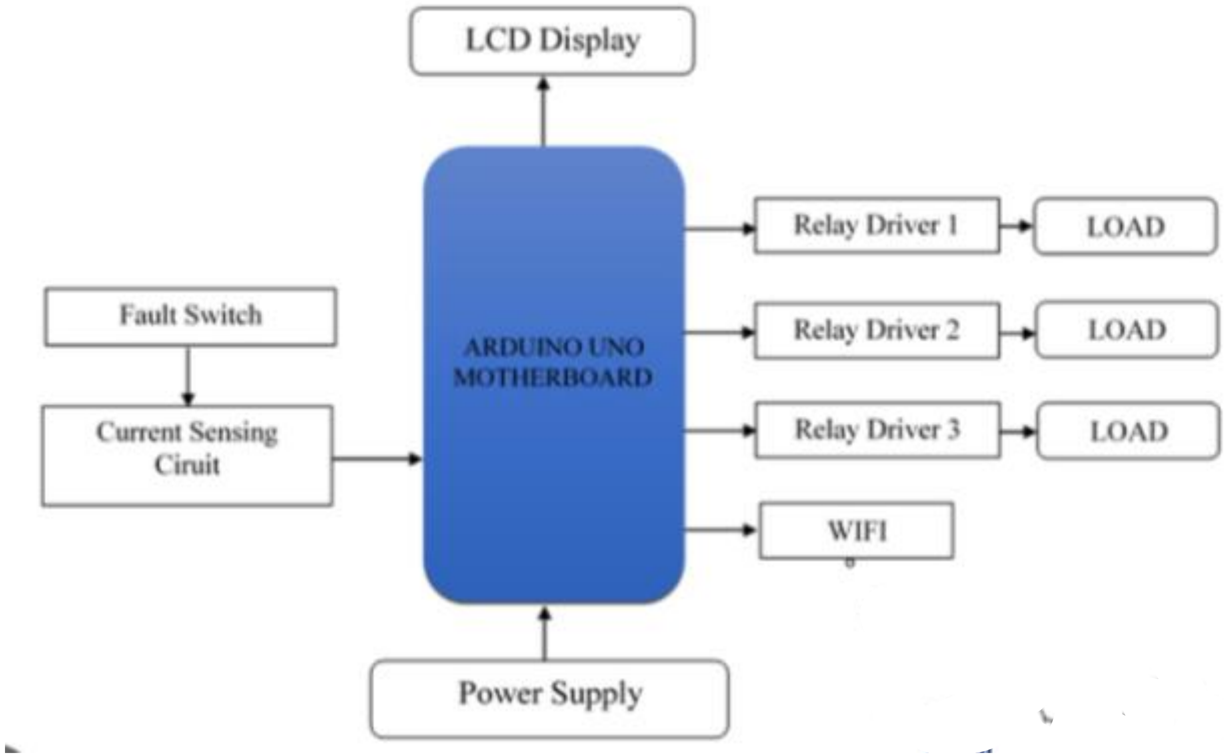
3.3.1 Architectural Diagram



3. Architectural Design

3.3 Project Architecture

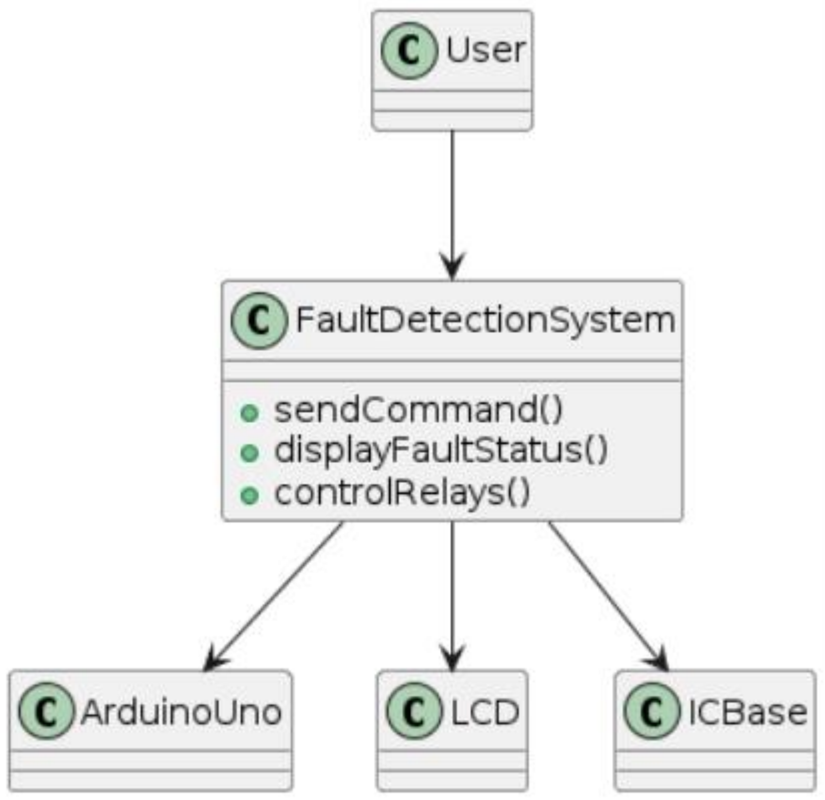
3.3.2 Data Flow Diagram



3. Architectural Design

3.3 Project Architecture

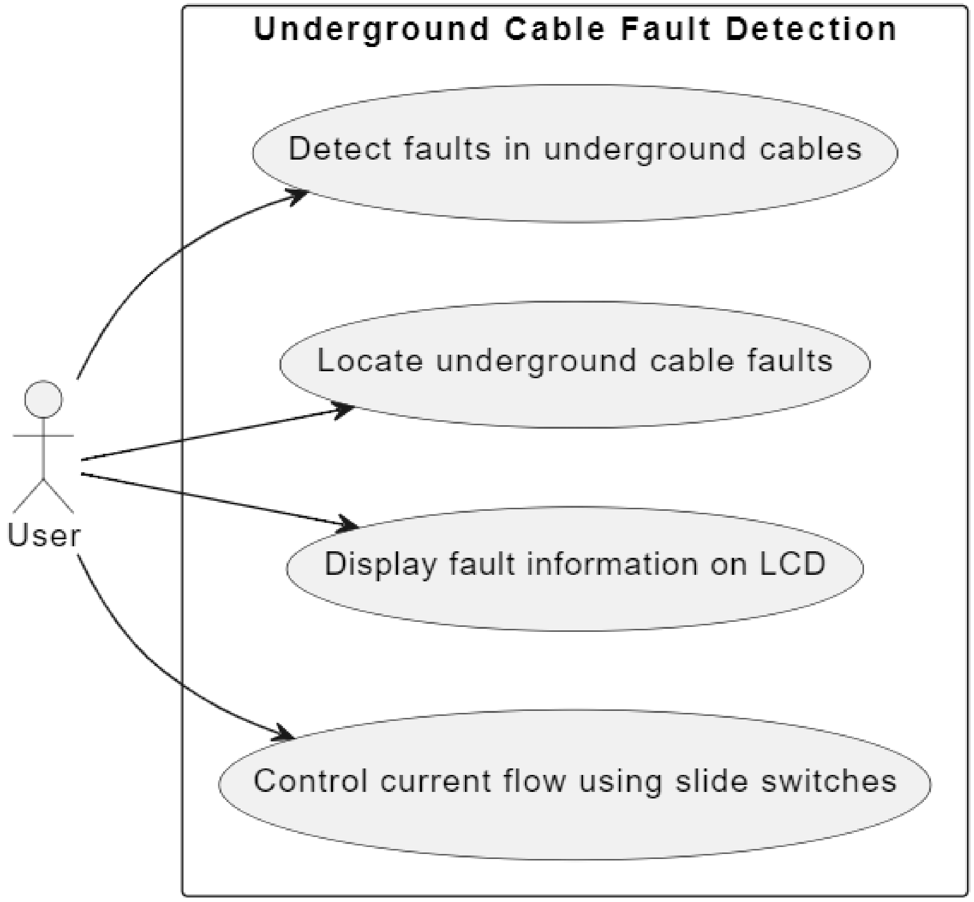
3.3.3 Class Diagram



3. Architectural Design

3.3 Project Architecture

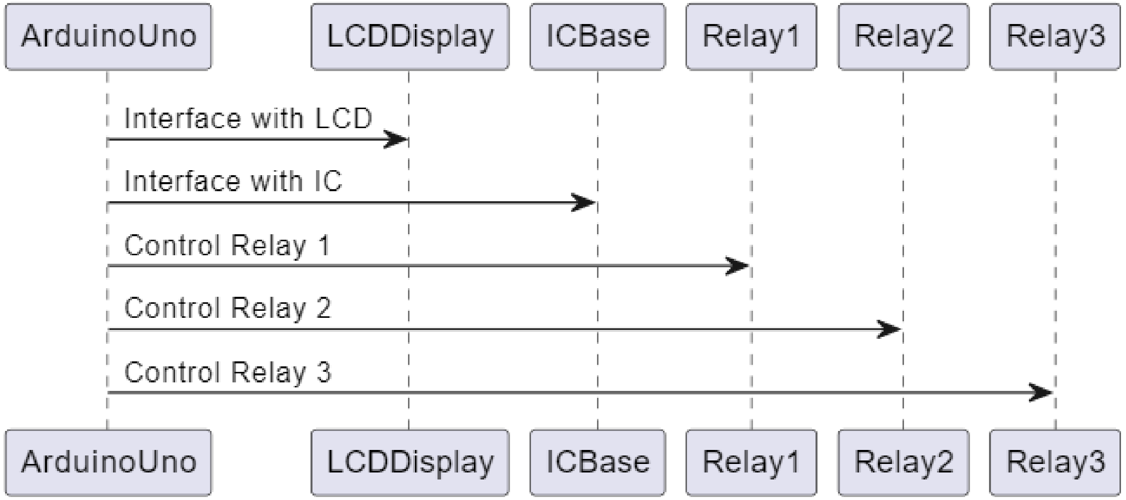
3.3.4 Use case Diagram



3. Architectural Design

3.3 Project Architecture

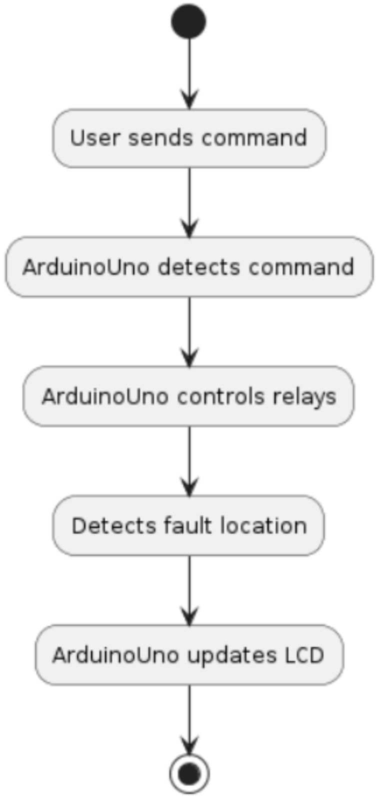
3.3.5 Sequence Diagram



3. Architectural Design

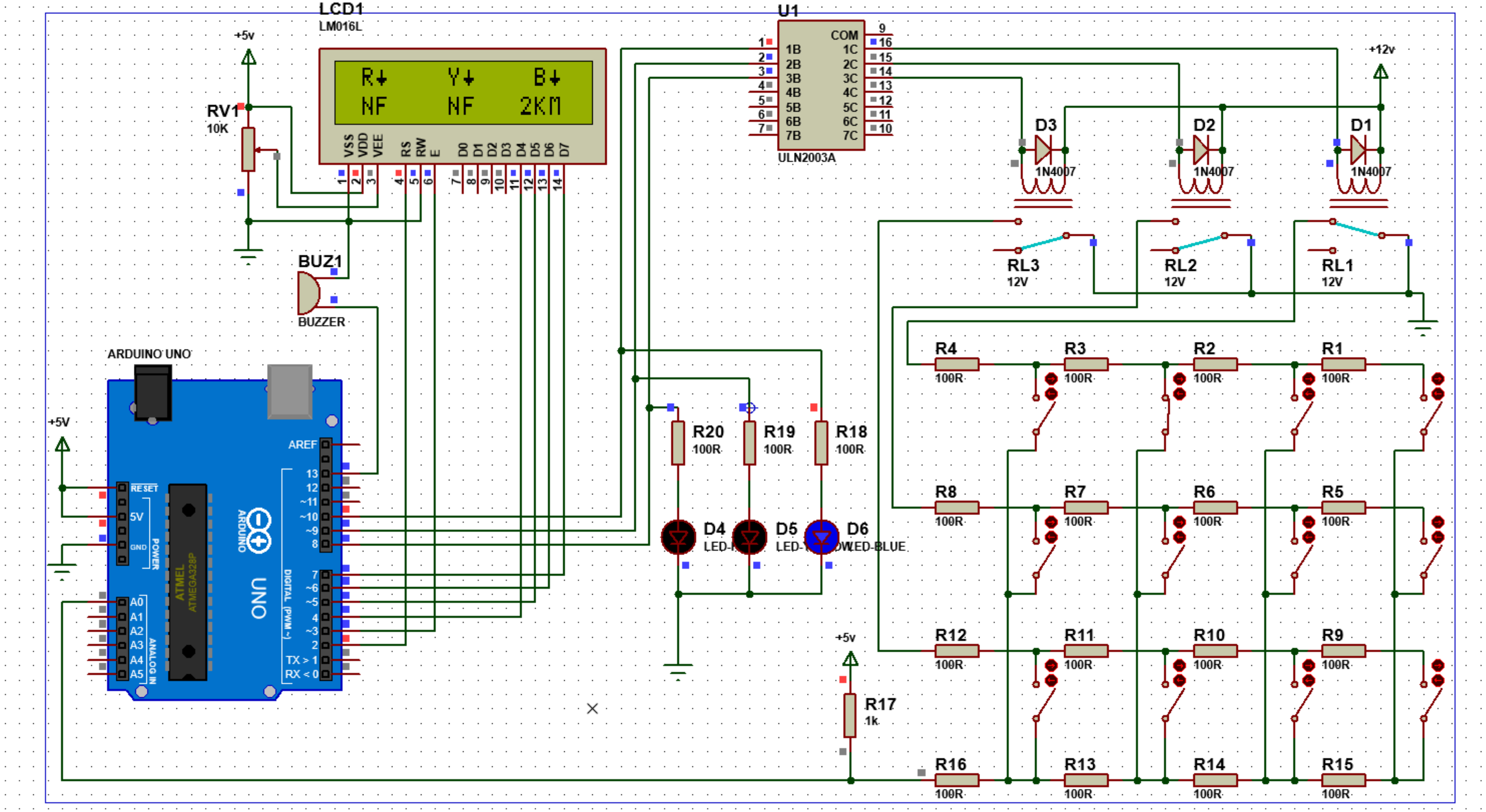
3.3 Project Architecture

3.3.6 Activity Diagram



3. Architectural Design

3.4 Circuit Diagram



4. Implementation & Testing

4.1 Coding Blocks

4.1.1 Libraries

4.1.2 Defining pins

4.1.3 Object Instantiation

4.1.4 Functions for obstacle detection

4.1.5 Main Loop

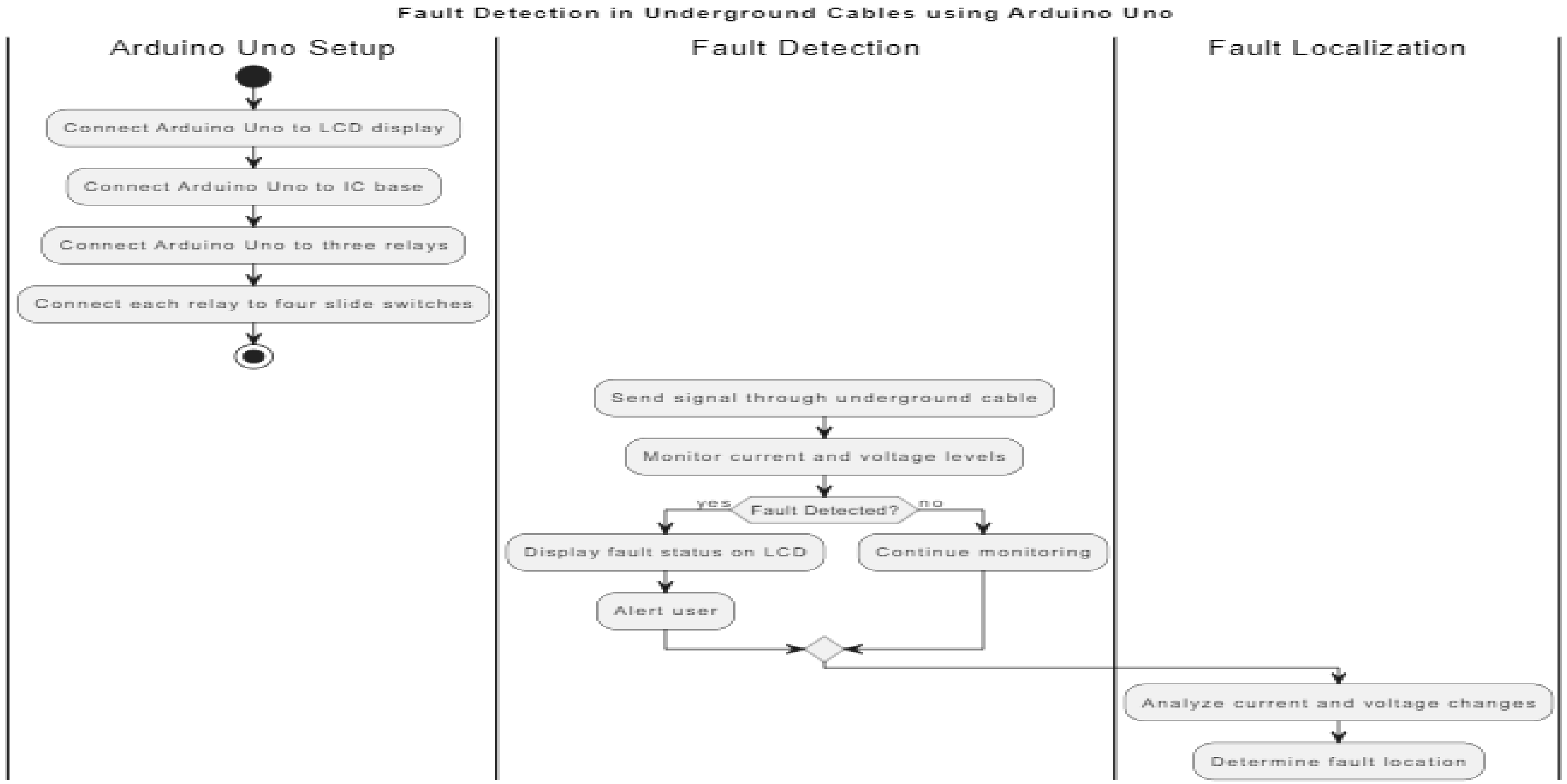
3. Architectural Design

4.2 Sample 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. Architectural Design

4.3 Execution Flow



4. Implementation & Testing

4.4 Testing

4.4.1 Fault Detection Test

1. **Test Setup:** Simulate a fault in the underground cable by toggling the appropriate slide switch.
2. **Test Steps:**

   Turn on the system and ensure that it is in a normal state.

   Simulate the fault by toggling the slide switch.

   Observe the LCD display for the fault detection message.
3. **Expected Result:** The system should accurately detect the fault and display a corresponding message on the LCD.

4. Implementation & Testing

4.4 Testing

4.4.2 Data Processing Test

1.Test Setup: Ensure that the system is in a normal state with no faults detected.

2.Test Steps:

1. Toggle the relays to simulate different fault conditions.
2. Observe the system's response and data processing.

3.Expected Result: The system should process the data correctly, detecting faults when present and displaying the appropriate messages on the LCD.

4. Implementation & Testing

4.4 Testing

4.4.3 Alert Message Test

1. **Test Setup:** Simulate a fault in the underground cable.
2. **Test Steps:**
   1. Turn on the system and ensure that it is in a normal state.
   2. Simulate the fault by toggling the slide switch.
   3. Observe the LCD display for the alert message.
3. **Expected Result:** The system should display a clear and informative alert message indicating the type and location of the fault.

4. Implementation & Testing

4.4 Testing

4.4.4 Integration Test

1.Test Setup: Ensure that all components (Arduino Uno, LCD display, IC base, relays, slide switches) are properly connected and functioning.

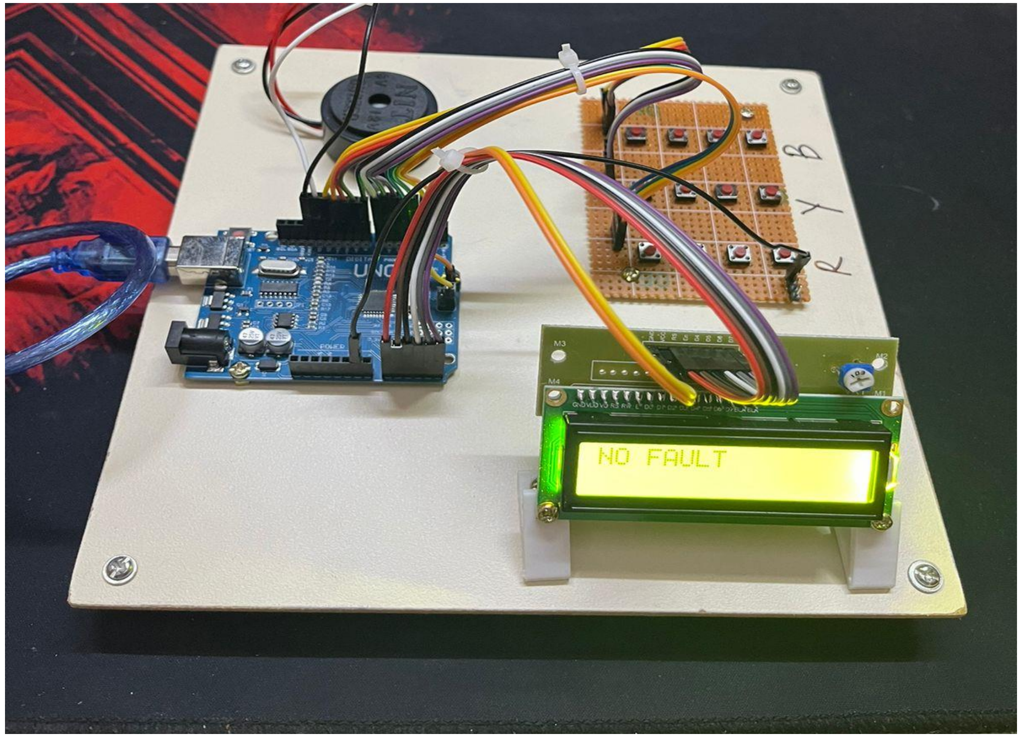
2.Test Steps:

1. Perform the Fault Detection, Data Processing, and Alert Message test cases in sequence.
2. Ensure that the system responds correctly to faults and displays the appropriate messages.

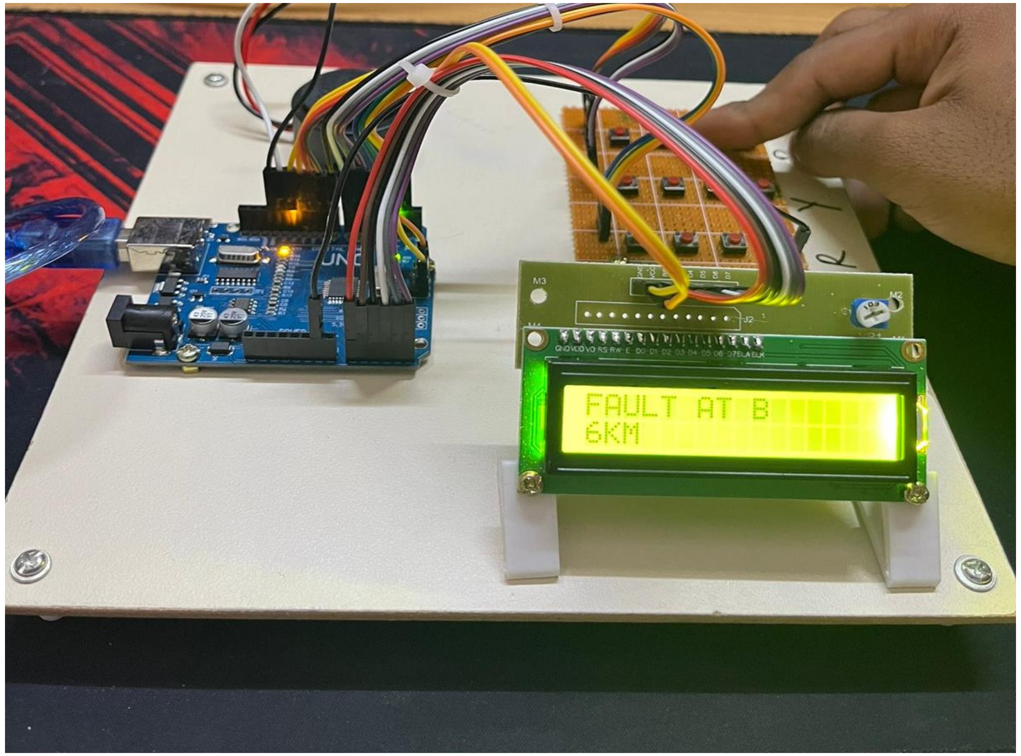
3.Expected Result: The system should demonstrate seamless integration of all components, accurately detecting faults and displaying the appropriate messages on the LCD.

5. RESULT

Case1: if there is no fault in underground cables the display unit will display line normal.



with in the range then the system will alert the user and display the fault, and respective phase relay goes switch off which is indicated by LED.



Here the fault is detected at B line at 6km so user will get alert by showing kms and line.

Thank You