Underground Cable Fault Detection Using Internet of Things

1*.Mr.China Babu.V 2.Talari.Sanjana 3.Kanithi .Sowmya
1 Assistant professor 2 BTech student 3 BTech student

1* Department of electronics and communication engineering
1*Rajiv Gandhi University Of Knowledge and Technologies,Ongole Campus ,Andhrapradesh

1. ABSTRACT

This project Underground Cable Fault Distance Locator is used for identifying faults in the underground cable. 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 Arduino microcontroller. To locate a fault in the cable, the cable must be tested for faults. This prototype uses the concept of Ohms law. The current would vary depending upon the length of fault of the cable. In the urban areas, the electrical cable run in underground instead of overhead lines. Whenever the fault occurs in underground cable it is difficult to detect the exact location of the fault. The prototype is modelled with a set of wires 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. 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 16X2 LCD interfaced with microcontroller

2. INTRODUCTION

The underground cable system is a more common practice followed in many urban areas. While 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 cable fault. The conventional method of finding the fault needs manpower and it requires high cost. The method which we are using at present are Tracer method and Terminal method. The fault occurs in the cable due to various reasons such as

- Inconsistency
- Weakness or non- homogeneity that affects
- Current is diverted from the intended path
- Caused by breaking of conductors & failure of insulation

TYPES

1. OPEN FAULT
2. SHORT FAULT

3. APPLICATIONS

It is used in industrial applications to detect underground cable fault where the cable is laid underground instead of overhead lines. The method which is used at present, requires a manpower and machineries to dig the entire underground cable to know the location of the fault. By employing this project, the distance of the fault created will be accurate and it will be easy to repair the fault. This project work is also intended to detect the location of fault in underground cable lines from the base station in km using a Arduino microcontroller since the process of locating the fault in the underground cable is very tedious. This is a cost effective project, and the process of setting up the project is also easy. The distance the fault occurred will be shown in the LCD display.

4.OBJECTIVES

The main objective of this project is to detect abnormalities occurring in underground cables using Arduino. When the fault occurs the process of repairing related to that particular cable is very difficult.

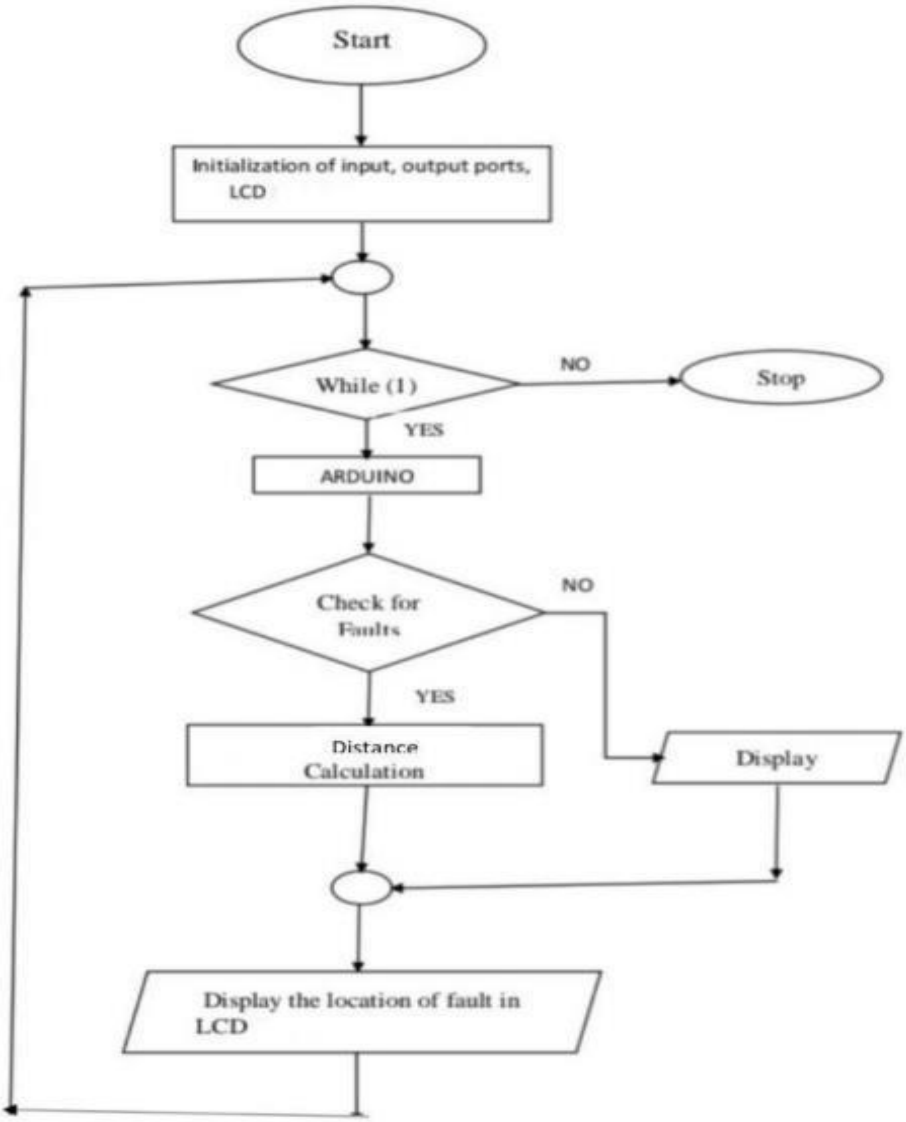
The fault of the main cable occurs due to various reasons, they are: inconsistent, any defect, weakness of the cable, insulation failure, and breaking of the conductor.

To overcome this problem, here is a project namely an underground cable fault distance locator, used to find the location of the fault for underground cable.

Using this project, we can accurately find the exact location of the fault in the underground cables.

This project can be done by using Arduino microcontroller coded with the program which is connected to LCD where the exact location of the fault will be shown

5.FLOW CHART



6.SOFTWARE IMPLEMENTATION

The Software used are

1.Proteus Professional
2.ArduinoIDE

PROTEUS PROFESSIONAL:

The Proteus Design Suite is a proprietary software tool suite used primarily for electronic design automation. The software is used mainly by electronic design engineers and technicians to create schematics and electronic prints for manufacturing printed circuit boards.

Schematic Capture

Schematic capture in the Proteus Design Suite is used for both the simulation of designs and as the design phase of a layout project. It is therefore a core component and is included with all product configurations.

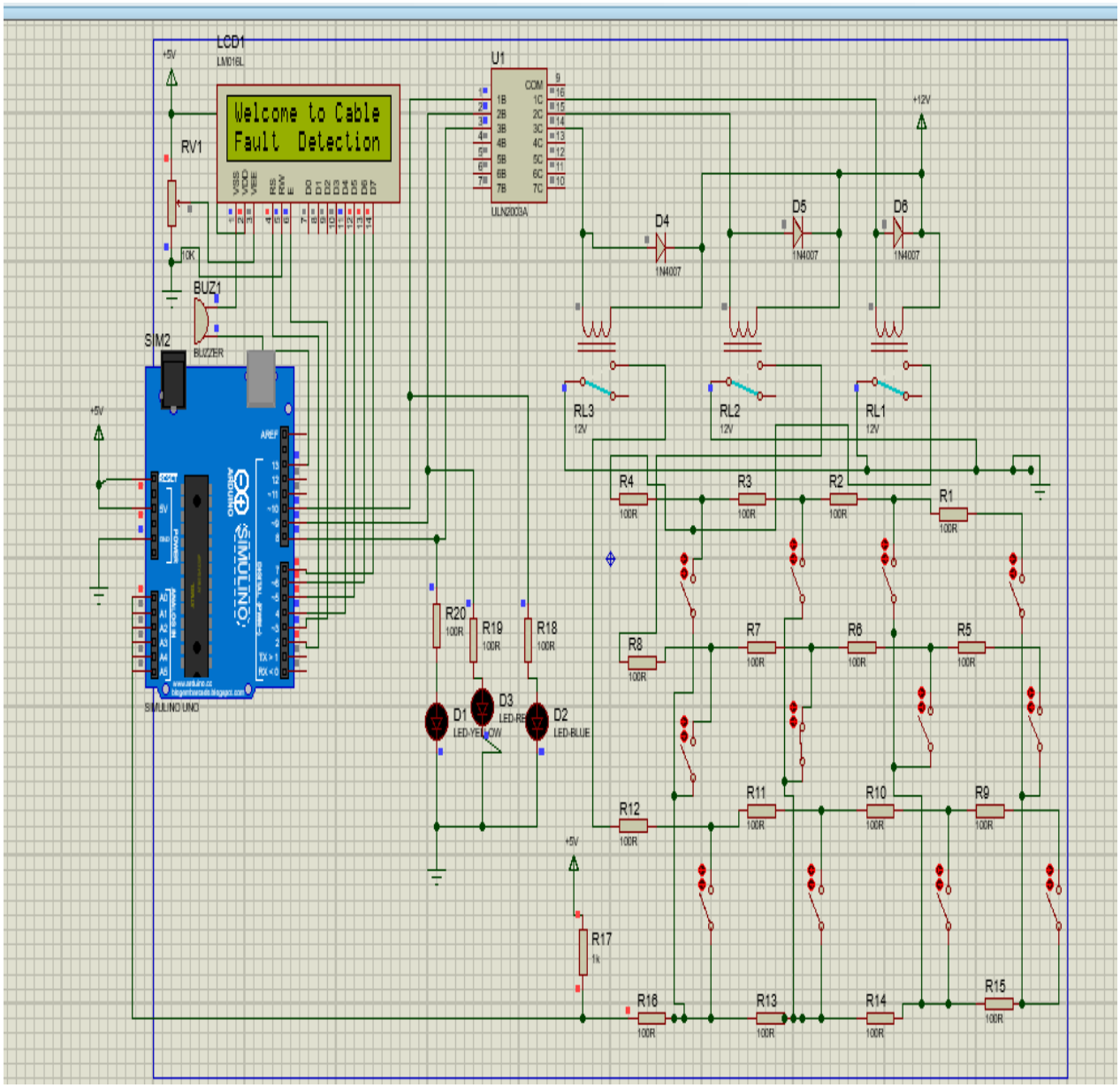
Microcontroller simulation

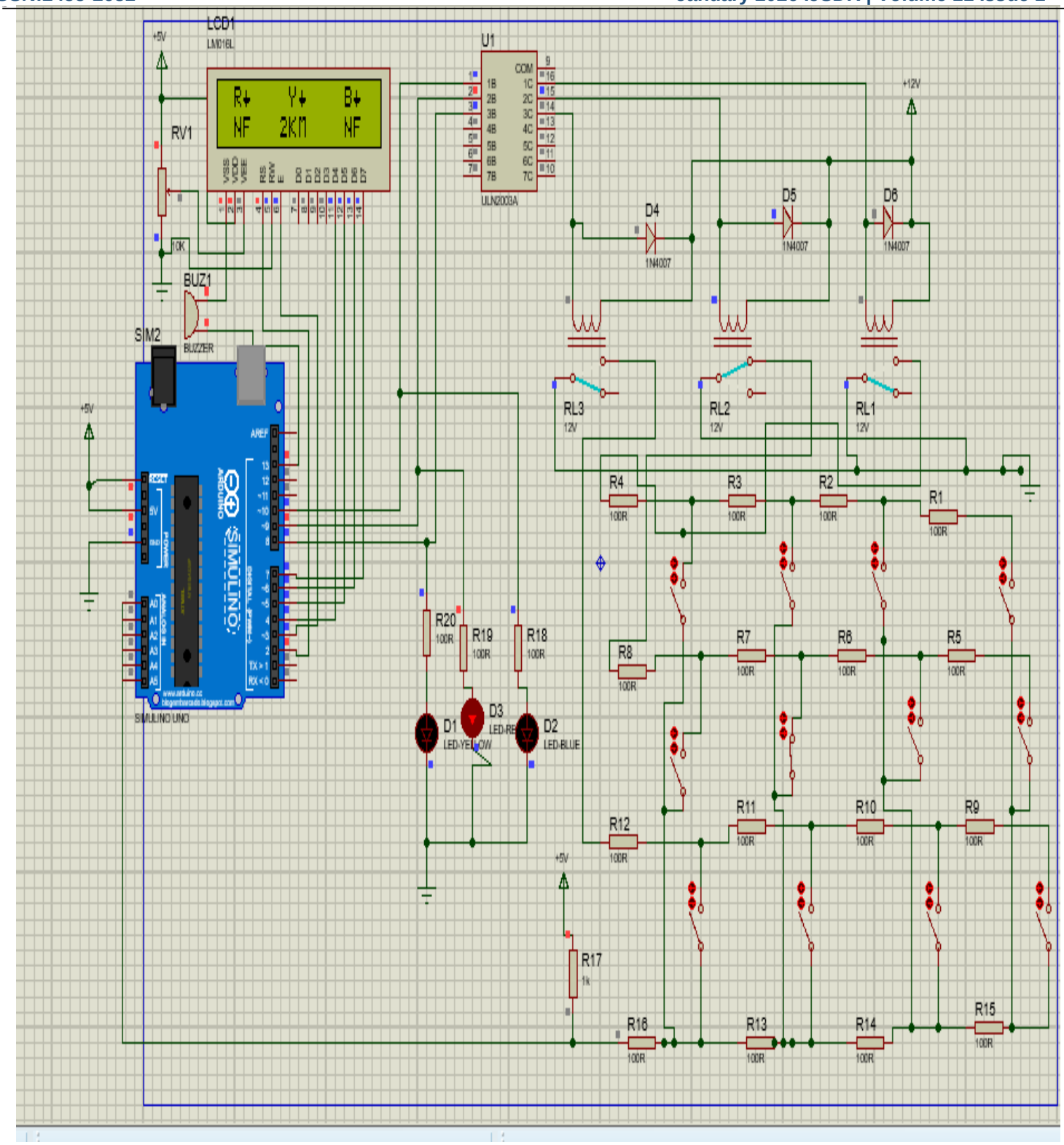
The micro-controller simulation in Proteus works by applying either a hex file or a debug file to the microcontroller part on the schematic. It is then co-simulated along with any analog and digital electronics connected to it. This enables its use in a broad spectrum of project prototyping in areas such as motor control, temperature control and user interface design. It also finds use in the general hobbyist community and, since no hardware is required, is convenient to use as a training or teaching tool.
Full feature (ISIS) schematic capture with support for hierarchical design bus pins configurable bill of materials and much more, ISIS lies at the heart of the Proteus system, and is far more than just another schematics package. It combines a powerful design environment with the ability to define most aspects of the drawing appearance. Whether your requirement is the rapid entry of complex designs for simulation and PCB layout, or the creation of attractive schematics for publication, ISIS is the tool for the job. ISIS supports Object oriented editor with automatic wire routing and dot placement or removal.

ARDUINO IDE:

The Arduino Integrated Development Environment (IDE) is a cross-platform application (for Windows, macOS, Linux) that is written in functions from C and C++. It is used to write and upload programs to Arduino compatible boards, but also, with the help of third-party cores, other vendor development boards.
The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. This software can be used with any Arduino board.
The source code for the IDE is released under the GNU General Public License. The Arduino IDE supports the languages C and C++ using special rules of code structuring. The Arduino IDE supplies a software library from the Wiring project, which provides many common input and output procedures. User-written code only requires two basic functions, for starting the sketch and the main program loop, that are compiled and linked with a program stub main() into an executable cyclic executive program with the GNU toolchain, also included with the IDE distribution. The Arduino IDE employs the program to convert the executable code into a text file in hexadecimal encoding that is loaded into the Arduino board by a loader program in the board's firmware.
With the rising popularity of Arduino as a software platform, other vendors started to implement custom open-source compilers and tools (cores) that can build and upload sketches to other microcontrollers that are not supported by Arduino's official line of microcontrollers.

7. SIMULATION AND OUTPUT





8.SIMULATION 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(200
 0);
 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,L
OW);
digitalWrite(relay3,L
OW);
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,L
OW);
digitalWrite(relay2,L
OW);
digitalWrite(relay3,H
IGH); 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);}
 }
```

9.Conclusion

Thus, the project on Underground cable fault detection using Arduino was done and the distance of the fault was displayed on the LCD for the three lines R,Y and B. When the fault switches are operated to fault condition then the phase corresponding to that particular switch is considered as the faulty phase, and the distance of the fault will be shown in the liquid crystal display using Arduino microcontroller.

10.REFERENCES

1.Abhishek Pandey, Nicolas. H, You nan, " Underground Cable Fault Detection and Identification via Fourier Analysis", 2010 IEEE.

2.Anurag.D.Bork hade (2014) „Transmission Line Fault Detection Using Wavelet Transform"- International Journal on Recent and Innovation Trends in Computing andCommunicationVolume.2

3.A.Ngaopitakkul, C.Pothisarn, M.Leelajindakrairerk, "Study of Characteristics for Simultaneous Faults in Distribution Underground Cable using DWT", 2011 IEEE.