IoT Based Underground Cable Fault

D. G. Wankhade[1] S. A. Gomase[2] S. G. Dharmale[3]
[1,2,3]Assistant Professor
[1,2,3]Department of Electronics and Telecommunication Engineering
[1,2,3]DRGITR, Amravati, India

Abstract— Fault location model for underground power cable using NodeMCU on IOT with google firebase as backend. The aim of project to get the alert of the fault occurrence and determine the distance of underground cable fault from base station in Km on an anroid application over the intrest in real time.

Key words: IoT, NodeMCU, Cable Fault

I. INTRODUCTION

Day by day, the world is becoming digitized & smart, so the project is proposed to find the location of the fault in digital way and in real time. When the fault occurs, the process of repairing related to that particular cable is very difficult. The fault of the cable mainly occurs due to many reasons. They are: inconsistent, any defect weakness of the cable, insulation failure and breaking of the conductor. To overcome this problem, here is project namely underground cable fault distance locator, used to find the location of the fault for underground cable. Fault in cable is represented as: Any defect, Inconsistency, Weakness or non-homogeneity that affect performance of cable, Current is diverted from the intended path, Caused by breaking of conductor& failure of insulation.

Moreover, the project locates the fault in cable and notifies the authorizations over the internet in real time with a mobile application which is powered by a real time database from Google Firebase. Google Firebase is the future of backend services. It includes the real time database which is the future of the databases. Any type of transfer of data occurred in real time. It is free to use. It reduces the tasks of developers to code for backend and managing databases.

II. INTERNET OF THINGS

The evaluation of IOT in the electrical Power Industry transformed the way things performed in usual manner. IOT increased the use of wireless technology to connect power industry assets and infrastructure in order to lower the power consumption and cost. The applications of IOT are not limited to particular fields, but span a wide range of applications such as energy systems, homes, industries, cities, logistics, heath, agriculture and so on. Since 1881, the overall power grid system has been built up over more than 13 decades, meeting the ever increasing demand for energy. Power grids are now been considered to be one of the vital components of infrastructure on which the modern society depends. It is essential to provide uninterrupted power without outages or losses. It is quiet hard to digest the fact that power generated is not equal to the power consumed at the end point due to various losses. It is even harder to imagine the after effects without power for a minute. Power outages occur as result of short circuits. This is a costly event as it influences the industrial production, commercial activities and consumer lifestyle.

Fig. 1. IoT Visualization

The evaluation of IOT in the electrical Power Industry transformed the way things performed in usual manner. IOT increased the use of wireless technology to connect power industry assets and infrastructure in order to lower the power consumption and cost. The applications of IOT are not limited to particular fields, but span a wide range of applications such as energy systems, homes, industries, cities, logistics, heath, agriculture and so on. Since 1881, the overall power grid system has been built up over more than 13 decades, meeting the ever increasing demand for energy. Power grids are now been considered to be one of the vital components of infrastructure on which the modern society depends. It is essential to provide uninterrupted power without outages or losses. It is quiet hard to digest the fact that power generated is not equal to the power consumed at the end point due to various losses. It is even harder to imagine the after effects without power for a minute. Power outages occur as result of short circuits. This is a costly event as it influences the industrial production, commercial activities and consumer lifestyle.

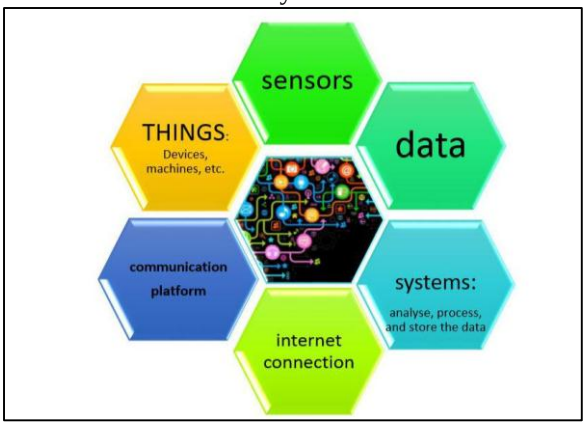

Fig.2. Components of Internet of Things

Government & independent power providers are continuously exploring solutions to ensure good power quality, maximize grid uptime, reduce power consumption, increase the efficiency of grid operations and eradicate outages, power loss & theft. Most importantly, the solution should provide a real-time visibility to customers on every penny paid for their energy. There is an increasing need of a centralized management solution for more reliable, scalable,

and manageable operations while also being cost effective, secure, and interoperable. In addition, the solution should enable power providers and utilities to perform effective demand forecasting and energy planning to address the growing need for uninterrupted quality power The goal of IOT is not just only connecting things such as machines, devices and appliances, but also allowing the things to communicate, exchanging control data and other necessary information while executing applications. It consists of IOT devices that have unique identities and are capable of performing remote sensing, monitoring and actuating tasks. These devices are capable of interacting with one another directly or indirectly. Data collection is performed locally or remotely via centralized servers or cloud based applications. These devices may be data collection devices to which various sensors are attached such as temperature, humidity, light, etc., or they may be data actuating devices to which actuators are connected, such as relays. IOT system is composed of three layers: the perception layer, the network layer, and the application layer.

The perception layer includes a group of Internet-enabled devices that can percept, detect objects, collect systems information, and exchange information with other devices through the Internet communication networks. Sensors, Global Positioning Systems (GPS), cameras, and Radio Frequency Identification Devices (RFID) are examples of devices that exist at perception layer. The network layer is responsible of forwarding data from perception layer to the application layer under the constraints of devices' capabilities, network limitation and the applications' constraints. IOT systems use a combination of Internet and short-range networks based on the communicated parties. Short-range communication technologies such as Bluetooth and ZigBee are used to carry the information from perception devices to a nearby gateway. Other technologies such as Wi-Fi, 2G, 3G, 4G, and Power line Communication (PLC) carry the information for long distances based on the application. The upper layer is the application layer, where incoming information is processed to induce insights for better power's distribution design and management strategies.

III. SYSTEM DESIGN

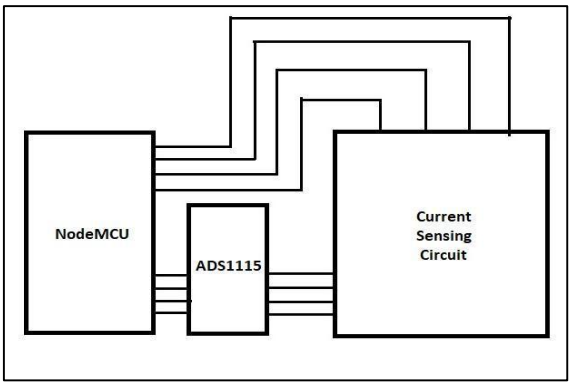

Fig.3. Block Diagram of IOT Based Underground Cable Fault Detection

The operation of the system states that when the current flows through the fault sensing circuit module the current would vary depending upon the length of the cable from the place of fault that occurred if there is any short circuit fault with the Single Line to ground fault, or double line to ground fault, or three phase to ground fault. The voltage drops across the series resistors changes accordingly and then the fault signal goes to internal ADC of the microcontroller to develop digital data. Then microcontroller will process the digital data and the output is being displayed in the LCD connected to the microcontroller in kilometres and phase as per the fault conditions. This Output is also displayed in the app through the IOT Ethernet Shield connected to the system.

The project uses the simple concept of OHM's law where a low voltage DC is applied at the feeder end through a series resistor. The current would vary depending upon the length of fault of the cable in case there is a short circuit. The series resistor voltage drop changes accordingly which is then fed to an ADC (Analog pins of Arduino Mega) to develop precise digital data which the programmed controller would display the same in kilo meters.

The project is assembled with a set of resistors representing the cable length in kilometers and fault creation is made by a set of switches at every known kilometer to cross check the accuracy of same. When we turn off the particular switch, the current in the particular cable will change accordingly. The controller will read the analog value of current and pre-programmed controller will convert it to the kilometers. The converted value & the cable id are then sent to the Firebase Database instantly. As soon as the firebase receives the data, it will instantly send that data to the mobile application. On mobile screen, the name of the particular cable & location of the fault in that cable will display. And it will remain as it is till the fault is corrected.

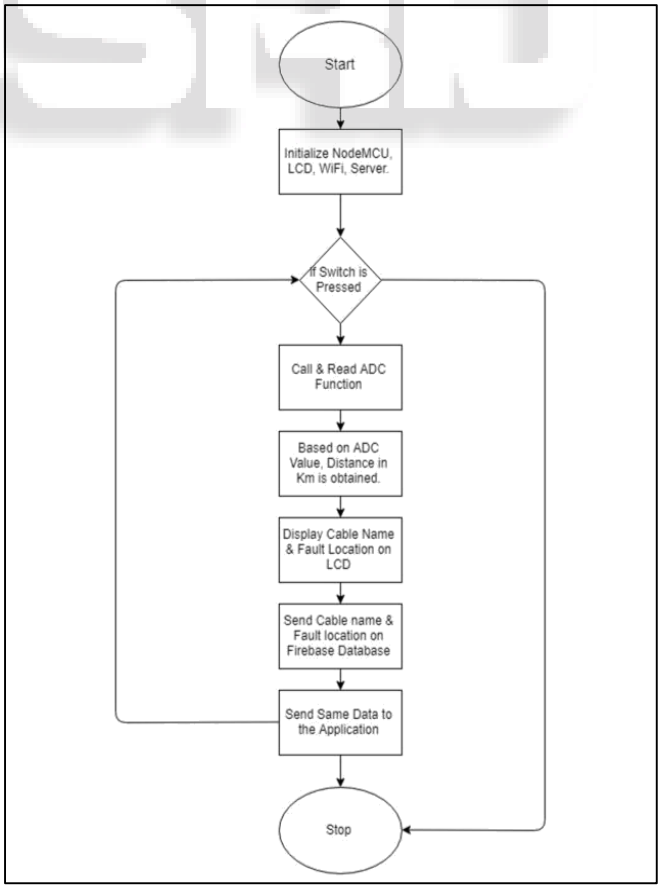

Fig.4. Flowchart of IOT Based Underground Cable Fault Detection

A. NodeMCU

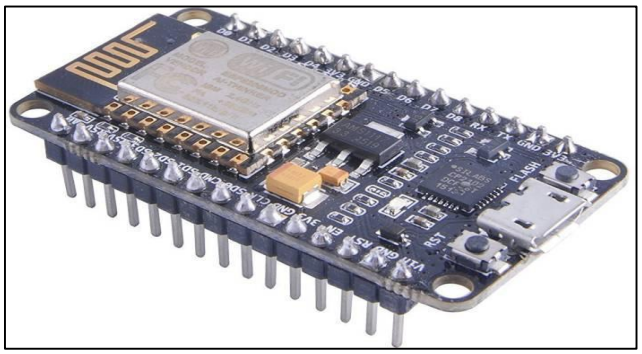

Fig. 5. NodeMCU

NodeMCU is an open source IoT platform. It includes firmware which runs on the ESP8266 Wi-Fi SoC from Espressif Systems, and hardware which is based on the ESP-12 module. The term "NodeMCU" by default refers to the firmware rather than the DevKit. The firmware uses the Lua scripting language. It is based on the eLua project, and built on the Espressif Non-OS SDK for ESP8266. It uses many open source projects, such as lua-cjson, and spiffs.

B. ESP8266 Arduino Core

As Arduino.cc began developing new MCU boards based on non-AVR processors like the ARM/SAM MCU and used in the Arduino Due, they needed to modify the Arduino IDE so that it would be relatively easy to change the IDE to support alternate toolchains to allow Arduino C/C++ to be compiled for these new processors. They did this with the introduction of the Board Manager and the SAM Core. A "core" is the collection of software components required by the Board Manager and the Arduino IDE to compile an Arduino C/C++ source file for the target MCU's machine language. Some ESP8266 enthusiasts developed an Arduino core for the ESP8266 WiFi SoC, popularly called the "ESP8266 Core for the Arduino IDE".[16] This has become a leading software development platform for the various ESP8266-based modules and development boards, including NodeMCUs.

C. Pins

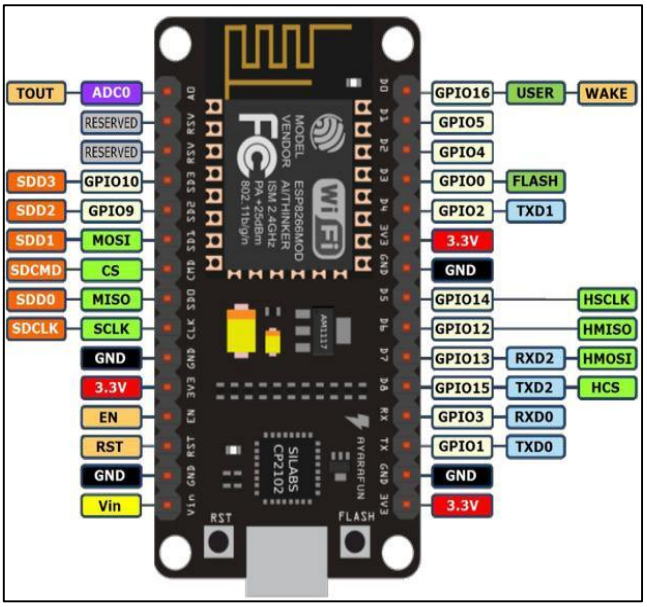

Fig. 6. NodeMCU Pinouts

NodeMCU provides access to the GPIO (General Purpose Input/Output) and a pin mapping table is part of the API documentation.

I/O Index	Esp8266 Pins	I/O Index	Esp8266 Pins
0	GPIO 16	7	GPIO 13
1	GPIO 5	8	GPIO 15
2	GPIO 4	9	GPIO 3
3	GPIO 0	10	GPIO 1
4	GPIO 2	11	GPIO 9
5	GPIO 14	12	GPIO 10
6	GPIO 12		

TABLE I PINS OF NodeMCU

D. Android Studio

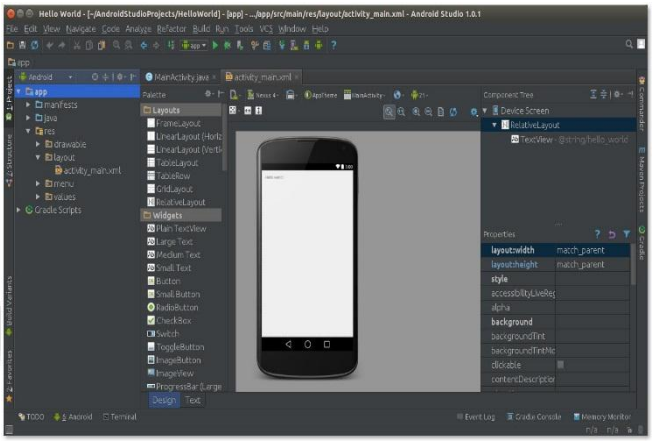

Fig .6. Android Studio

Android Studio is the official integrated development environment (IDE) for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development. It is available for download on Windows, macOS and Linux based operating systems. It is a replacement for the Eclipse Android Development Tools (ADT) as the primary IDE for native Android application development.

Android Studio was announced on May 16, 2013 at the Google I/O conference. It was in early access preview stage starting from version 0.1 in May 2013, then entered beta stage starting from version 0.8 which was released in June 2014. The first stable build was released in December 2014, starting from version 1.0. The current stable version is 3.3, which was released in January 2019.

Android Studio supports all the same programming languages of IntelliJ (and CLion) e.g. Java, C++, and more with extensions, such as Go; and Android Studio 3.0 or later supports Kotlin and "Java 7 language features and a subset of Java 8 language features that vary by platform version." External projects backport some Java 9 features. While IntelliJ that Android Studio is built on supports all released Java versions, and Java 12, it's not clear to what level Android

Studio supports Java versions up to Java 12 (the documentation mentions partial Java 8 support). At least some new language features up to Java 12 are usable in Android.

1) Visual Layout Editor

Create complex layouts with constrain layout by adding constraints from each view to other views and guidelines. Then preview your layout on any screen size by selecting one of various devices configurations or by simply resizing the priview window.

2) APK Analyzer

Find apportunities to reduce your android app size inspecting the contents of your app APK file, even if it wasn't built with android studio. Inspect the manifest files, resources files, DEX files. Compare two APKs to see how your app size changed between app versions.

3) App Emulator

Install and run your apps faster than with physical devices and simulate different configurations and features, including ARCore, Google's platform for building augumented reality experiences.

4) Intelligent Code Editor

Write better code, work faster, and be more productive with an intelligent code editor that provides code completion for Kotlin, Java, C/C++ languages.

5) Flexible Build Systems

Powered by Gradle, Android Studio's build system allows customize your build to generate multiple build varients for different devices in a single project.

6) Realtime Profilers

The built in profile tools provides realtime staticstics for your App's CPU, memory and network activity. Identify performance bottlenecks by recording method traces, inspecting the head and allocation, and see incoming and outgoing payloads.

IV. ANDROID APPLICATION

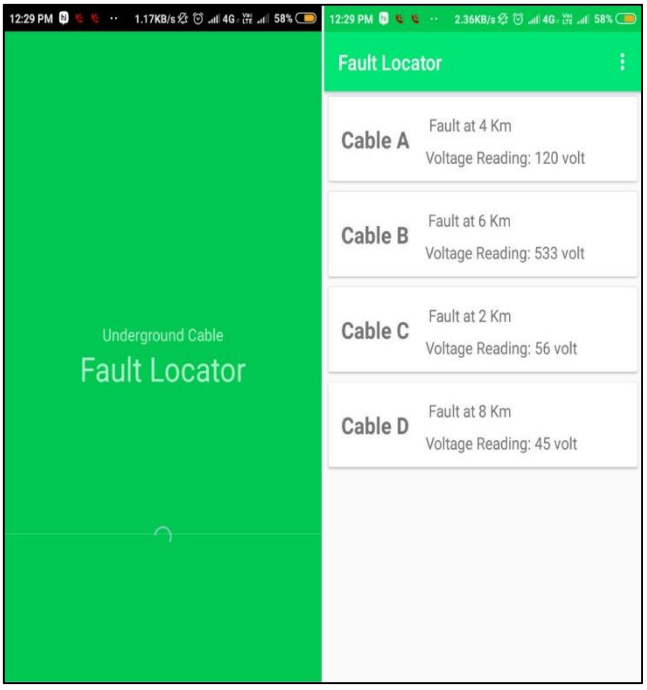

Fig. 7. Screenshots of Android Application

"Fault Locator" App is developed in Android Studio. When we click on the app, the app starts with a poster showing the name of the app "Underground Cable Fault Locator". After few seconds, the main screen of the app appears as shown in figure 7.1. The app shows the name of the cable, fault location of the cable in kilometers, and the voltage reading of the cable whenever the fault occurs. Also when fault occurs, the android application alert the user with an alert sound even if phone is on silent or vibration mode.

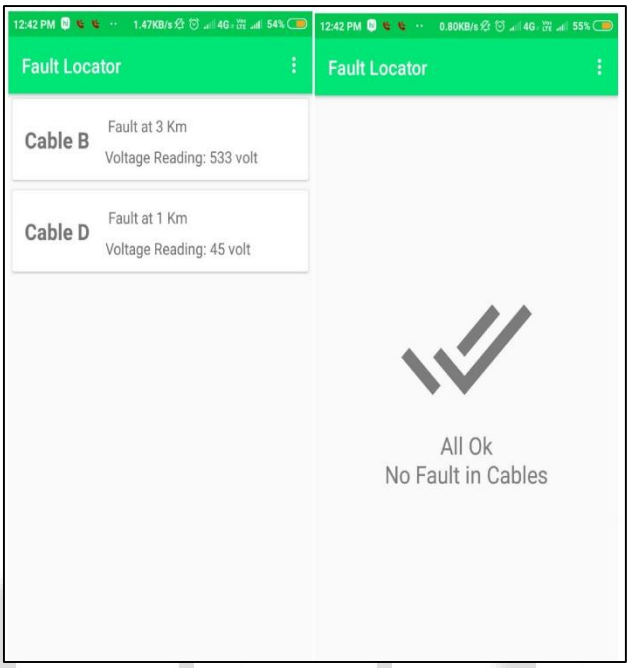

Fig .8.More screenshots of the Android Application

The app shows the name of the only cables where is there fault. If there are no fault in cable, it will not be shown in the list. And if there are no any fault in any cable, then the app will show that everything is ok, no fault in cable as shown in fig 8

V. WORKING

The prototype uses resistors to represent the cable length. The resistors R11 to R13 represents R1 phase of the cable. Similarly R21 to R23 and R31 to R33 represent R2 and R3 phase of the cable. RN1. To represent the occurrence of fault in underground cables switches are used. Each phase is connected with a power pin of Microcontroller, and analog pins connected to ADC pins of the ADS1115.

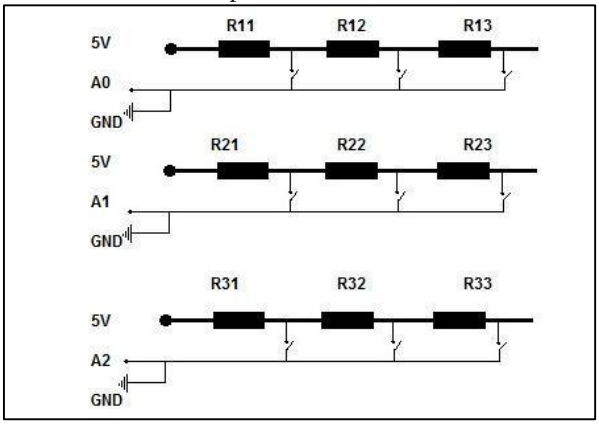

Fig. 9. Circuit Diagram

When a switch connected to a particular phase is closed. The resistance connected to that particular phase adds up and the voltage drop thus generated is given to Port A of the Microcontroller. The voltage drop is converted to distance and is displayed in the LCD. The cable side circuit diagram is shown in the Figure 7.1. The data is then sent to the firebase server on their respected place.

When data is get changes on the database, the firebase database detects the change in values and automatically send the changed data to the app with calling for it in real time. The app will play buzzer when any fault occurs in any cable.

VI. Conclusion

The short circuit fault at a particular distance in the underground cable is located to rectify the fault efficiently using simple concepts of Ohms law. The work automatically displays the phase, distance and time of occurrence of fault with the help of PIC 16F877A and ESP8266 Wi - Fi module in a webpage. The benefits of accurate location of fault are fast repair to revive back the power system, it improves the system performance, it reduce the operating expense and the time to locate the faults in the field.

References

[1] Abishek Pandey, Nicolas H. Younan Underground cable fault detection and identification using Fourier analysis, 2010 IEEE Conference
[2] C.Bharatiraja, S.Jeevananthan, J.L. Munda, A Timing Correction Algorihm based extended SVM for three level Neutral point clamped MLI in Over Modulation Zone IEEE Journal of Emerging and Selected topics in Power Electronics.
[3] Dhivya Dharani. A and Sowmya. T, Development of a Prototype of Underground Cable Fault Detector, International Journal Electrical, Electronics, and Computer Systems, Volume-2, 2014.
[4] Gilbert Cheung, Yuan Tian, Tobias Neier, Technics of Locating Underground Cable Faults inside conduits, International Conference on Condition Monitoring and Diagnosis IEEE (CMD 2016)
[5] M.Fonseca_Badillo, L. Negrete_Navarrete, A. Gonzalez_parada, A. Castaneda_Miranda, Simulation and analysis of underground power
[6] Manar Jaradat, Moath Jarrah, Abdel Kader Bousselham, Yaser Jararweh, Mahmoud AlAyyoub The Internet of Energy: Smart Sensor Networks and Big Data Management for Smart Grid, Procedia Computer Science Elsevier, July 2015.
[7] Md. Fakhrul Islam, Amanullah M T O, Salahuddin. A. Azad, Locating Underground Cable Faults: A Review and Guideline for New Development, 2013 IEEE Conference
[8] Nikhil Kumar Sain, Rajesh Kajla, and Mr.Vikas Kumar, Underground Cable Fault Distance Conveyed Over GSM, International Organization of Scientific Research Journal of Electrical and Electronics Engineering, Volume 11, Issue 2, Mar-April 2016.
[9] Xiaoning Kang; Xiuda Ma; Shuai Jiang; Xiaoyun Qu, Chao Zhang; Xiaoning Kang; Xiuda Ma; Shuai Jiang; Xiaoyun Qu 2016 IEEE PES Asia-Pacific Power and Energy Engineering Conference (APPEEC)

