Sudan University of Science and Technology

Collage of Graduate Studies

Design of Digital Fault Recorder Using Microcontroller for Electrical Distribution Company Network

تصميم مسجل الاعطال الرقمي باستخدام المعالج الدقيق لشبكة شركة توزيع الكهرباء

A Thesis Submitted in Partial Fulfilment to the Requirements for the Award of the Degree of M.Sc. in Electrical Engineering (Microprocessor and Control)

Prepared By:

Khogali Abdulkareem Khogali Alameen

Supervised By:

Dr. Mudathir Abdallah Osman Fagiri

January 2019

الآيـــة

بسم الله الرحمن الرحيم

قـال تـعـالـى :

{ قالوا سُبحانكَ لا علِمَ لنا الاّ ماعلّمتَنا إنّـكَ أنتَ العَليمُ الحَكيمْ }

صدق الله العظيم

(سورة البقرة : الاية 32)

DEDICATION

Dedication to my mother ....

With warmth and faith.

Dedication to my father ....

With love and respect.

Dedication to my friends ....

Whom we cherish their friendship.

Dedication to my special people ....

Whom mean so much to me.

Dedication to all my teachers ....

Whom I believe so much.

ACKNOLOGEMENT

I thank ALLA almighty for his guidance during these important years.

Also, I extend my thanks to all who stood with me to achieves this research which it come because of grace of ALLAH and reconcile.

I would like to give special thanks to my supervisor Doctor: Mudathir Fagiri

For his great help and supports. He will always be remembered as a great teacher, counselor and father.

And

My teachers whose gave me information and all staff in Sudan University of science and technology.

ABSTRACT

It is obvious that our modern life has become very dependent on the constant availability of electricity. Therefore, the efficiency of electrical energy is of the utmost importance in our lives and this increase in the sensitivity of electrical appliances used both on the domestic and practical side. This efficiency is achieved consistently values of voltage, current, frequency, and any turbulent changes in the values of these elements may cause damage to users. The thesis aims to monitor the main electrical elements continuously and record mistakes if they occur and then send these data to the specialized control center so that operators can analyze the situation and take appropriate measures to reduce the effects of these errors on the electrical grid and users. The exact controller is used as the controlled mind of this system. The readings of the current values of the current network are taken from the current transformers with the main switches and then sent directly to a computer in the control center and received and displayed using a specific programs.

المستخلص

لا يخفى على احد ان حياتنا المعاصرة اصبحت تعتمد اعتمادا كبيرا على توفر الطاقة الكهربائية بشكل ثابت ومستمر. ولذلك اصبحت كفاءه الطاقة الكهربائية ذات اهمية قصوى في حياتنا وهذا نسبة لزياده حساسية الاجهزه الكهربائية المستخدمة سواءا على الجانب المنزلي او العملي.هذه الكفاءه تتحقق بثبات قيم كل من الجهد والتيار والتردد , واي تغيرات مضطربة في قيم هذه العناصر قد تؤدي لوقوع اضرار بالنسبة للمستخدمين . ومن ذلك يهدف المشروع لمراقبة اهم العناصر الكهربائية بصورة مستمرة وتسجيل الاخطاء اذا حدثت ومن ثم ارسال هذه البيانات الى مركز التحكم المتخصص حتى يتمكن المشغلون من تحليل الوضع واتخاذ الاجراءات المناسبة للتقليل من اثار هذه االخطاء على الشبكة الكهربائية وعلى المستخدمين. ويستخدم المتحكم الدقيق كعقل مدبر لهذا النظام حيث تتم اخذ قراءات قيم التيار في الشبكه الماخوذه من محولات التيار الموضوعه مع المفاتيح الرئيسية وبعد ذلك ارسالها مباشرة الى جهاز حاسب بمركز التحكم واستقبالها وعرضها باستخدام برنامج معين . فتسجيل الاخطاء وعرضها يوفر قاعدة بيانات تساعد في تحليل الاخطاء ودراستها والبحث عن حلول لتلافى الوقوع فيها في المستقبل.

Table of Contents

Chapter Three

Methodology

Chapter Four

Results and Discussion

Chapter Five

Conclusion and Recommendations

List of Figures

List of Abbreviations

DFRs	Digital Fault Recorders
MATLAB	matrix laboratory
AC	Alternating Current
HV	High Voltage
EHV	Extra High Voltage
SCADA	Supervisory control and data acquisition
UART	Universal Asynchronous Receiver - Transmitter
EIA	Electronic Industries Association
DDFR	Distributed Digital Fault Recorder
CT	Current transformer
A/D	Analog to Digital
ICs	Integrated Circuits
LED	Light Emitted Diode
I/O	Digital Input/Output
DC	Direct Current
EEPROM	Electrically Erasable Programmable Read Only Memory
USB	Universal Serial Bus
ROM	Read Only Memory
RAM	Random Access Memory
MCU	Microcontroller Unit
AVRs	Audio/Video Receivers
PC	Personal Computer
IDE	Integrated Development Environment
GUI	Graphical User Interface
ATM	Automated Teller Machines
POS	Point Of Sale

CHAPTER ONE

INTRODUCTION

1.1 Overview

Fault records are one of the most important pieces of evidence that event analysts can have during system event investigations. They can provide the reasons for premature equipment failure and status of equipment behavior during an event and give necessary information to perform post-fault event analysis. Proper use and interpretation of event records can lead to corrective action for a given system problem resulting in improved performance and reliability of any generation, transmission, and distribution system. Fault records are now captured by microprocessor relays but records are limited to sampling rate and record length. Some use digital filters that do not reflect the real captured waveform [1].

Digital fault recorders offer specialized, specific, and dedicated microprocessor equipment with far superior sampling rates, record lengths, and unfiltered recording abilities. Utility engineers have to make balanced decisions as to what equipment is better to use for pre- and post-event analysis. Regardless of the equipment employed, both come at some economic cost. Nevertheless, as expected maximum use of their recording capabilities assures maximum return in their investment.

It is important to understand the meaning of fault analysis and recoding, and this guide offers the newcomer some practical explanations and applications. Digital Fault Recorders (DFRs) and microprocessor-based relays offer recording capabilities in the form of waveforms and sequences of events. However, these two differ in the sampling rate processing power, type of record they can capture, lengths of records, and the ability to record wide system response. The important factor is to know the characteristics that both pieces of equipment offer and determine which one offers the best information for event analysis [1].

1.2 Problem Statement

Typical power system contains several thousands of transmission lines. Installation of recording devices at each transmission line is very expensive and this approach cannot be found in practice. It is common that (DFRs) are placed in critical substations and record voltages, currents and breaker contacts associated with several transmission lines connected to that substation. Protective relays are spread all over the system, but some of them are still electromechanical and they do not have capability to record measurements. In some cases, it can happen that there are no recordings at all available close to a fault.

1.3 Objectives

- Clarifications that the records are one of the most important pieces of evidence that analysts can have during system event investigations.
- Provide the reasons for premature equipment failure, supply waveforms and status of equipment behavior during an event and give necessary information of event records.
- Give a corrective action for given system problem resulting in improved performance and reliability of any generation, transmission, and distribution system. DFR offer specialized, specific, and dedicated microprocessor equipment with far superior sampling rates, records lengths, and unfiltered recording abilities.
- Provide permanent archives of data to give support at any time to operations staff in having the desired information on the plant status very quickly. However, the analogue waveforms of currents and circuit breakers, which form a plant – wide data base to provide the

necessary where withal for accurate analysis of network disturbances, are captured only by fault recording system [2].

1.4 Methodology

The concept of this thesis is to try to simulate the digital fault recorder uses simpler and cheaper components so that it can be used in different places and under many situations, and then facilitates future development.

1- Successes reading voltage or current parameters from the line using analog to digital converter (ADC) that provided in AVR microcontroller (Arduino).

2- The PC needs to be equipped directly connected with (Arduino) over serial communication link (RS-232 Protocol). This would enable short downloading time and local processing for all events recorded by DFR. The PC needs to communicate the event classification results.

3- Using MATLAB program to display the fundamental signal as it is. Then analysis this signal using voltage or current parameters to determine type of fault. So, operator can observe and control the signal easily and do processors and action in a safe manner.

4- The results of the analysis need to be stored in a central database and made available to the users' Final goal is to record the fault and its properties whenever there are, using MATLAB. And save it in an M-file and (guide Function) for easy reference later.

1.5 Thesis Layout:

Chapter two: reviews most of the Faults types, the purpose, principles and the importance of DFR in power plants. The description, structure and components of the Arduino are discussed.

Chapter three: describes the general structure and operations of DFR. All hardware and software design steps including equipments which used in DFR design, physical equipments algorithms, and circuits block diagram, flow chart are discussed.

Chapter four: consist of simulation results and their discussions.

Chapter five: Contains the general discussion of results obtained in Chapter 4, conclusions and recommended future work.

CHAPTER TWO

BACK GROUND AND LITERTURE REVIEW

2.1 Introduction

This chapter provides an overview of common faults occur in transmission lines in power systems which we aim to detect and control it by design a simple digital fault recorder. So many types of faults and their effects with analysis a must be taken into account when designing the DFR to suit all needs. Also, here we discuss the definition and purpose of the digital fault recorder and simple comparison between DFR and the digital relay.

2.2 Transmission and Distribution Lines

Transmission lines constitute a major part of the power system. Transmission and distribution lines are vital links between the generating unit and consumers to achieve the continuity of electric supply. To economically transfer large blocks of power between systems and from remote generating sites, High Voltage (HV) and Extra High Voltage (EHV) overhead transmission systems are being used [3].

Transmission lines also form a link in interconnected system operation for bidirectional flow of power. It is run over hundreds of kilometers to supply electrical power to the consumers. They are exposed to atmosphere, hence chances of occurrence of faults in transmission line is very high, which has to be immediately taken care of in order to minimize damage caused by it. It will also facilitate quicker repair, improve system availability and performance, reduce operating cost and save time and effort of maintenance crew searching in sometimes in harsh environmental conditions. It has always been an interest for engineers to detect and locate the faults in may lead to blackouts [3].

Power is usually transmitted as Alternating Current (AC) through overhead power lines. Underground power transmission is used only in densely populated areas because of its higher cost of installation and maintenance when compared with overhead wires, and the difficulty of voltage control on long cables. An overhead power line is an electric power transmission line suspended by towers or poles. Since most of the insulation is provided by air, overhead power lines are generally the lowest-cost method of transmission for large quantities of electric power.

Any fault, if not detected and isolated quickly will cascade into a system wide disturbance causing widespread outages for a tightly interconnected system operating close to its limits. Transmission protection systems are designed to identify the type and location of faults and isolate only the faulted section. The key challenge to the transmission line protection lies in reliably detecting and isolating faults compromising the security of the system [3].

2.3 The Fault

A fault is any abnormal condition in a power system. The steady state operating mode of a power system is balanced 3-phase A C. However, due to sudden external or internal changes in the system, this condition is disrupted. When the insulation of the system fails at one or more points or a conducting object comes into contact with a live point, a short circuit or a fault occurs [3].

2.3.1 Causes of faults

The fault of transmission line occurs for many causes some of it is natural and other are made by humans here is some cases we have seen:

(a) Nature causes

- Lightning strikes near line.
- Fires (farmers burn weeds out from under a line - the fire ionizes the air, and with the added particulates in the air create a path to ground).
- Snow storms/icing on lines (can cause lines to sag).

- High winds (can cause lines to blow out, possibly making contact with trees, or can cause enough stress to break poles). Winds can also blow stuff into the lines.

(b) Animals causes

- Creeping animals sticking between phases.
- Eagle or other large birds swooping down between conductors.

(c) People causes

- Farmers touching power lines with harvesting equipment.
- Crop dusters attempting to fly under lines.
- People crashing cars into poles.

Faults usually occur in a power system due to insulation failure, flashover, and short- circuits to earth or between live conductors or may be caused by broken conductors in one or more phases. Sometimes simultaneous faults may occur involving both short-circuit and broken conductor faults (also known as open-circuit faults) [4].

2.3.2 Types of Faults and their Effects

It is not practical to design and build electrical equipment or networks so as to completely eliminate the possibility of failure in service. It is therefore an everyday fact of life that different types of faults occur on electrical systems, however infrequently, and at random locations. Faults can be broadly classified into two main areas which have been designated: active and passive [4].

(a) Active Faults

The active fault is when actual current flows from one phase conductor to another (phase-to-phase) or alternatively from one phase conductor to earth (Phase-to-Earth) [4].

(b) Passive Faults

Passive faults are not real faults in the true sense of the word but are rather conditions that are stressing the system beyond its design capacity, so that ultimately active faults will occur. Typical examples are:

- Overloading - leading to overheating of insulation (deteriorating quality, reduced life and ultimate failure).
- Overvoltage - stressing the insulation beyond its limits.
- Under frequency - causing plant to behave incorrectly.

Power swings – generators going out-of-step or synchronism with each other. It is very necessary to also protect against these conditions. Other causes can typically be a high-resistance joint or contact, alternatively pollution of insulators causing tracking across their surface. Once tracking occurs, any surrounding air will ionize which then behaves like a solid conductor consequently creating a "solid" fault [5].

(c) Symmetric fault

A symmetric or balanced fault affects each of the three phases equally. In transmission line faults, roughly 5% are symmetric. In practice, most faults in power systems are unbalanced. With this in mind, symmetric faults can be viewed as somewhat of an abstraction; however, as asymmetric faults are difficult to analyze, analysis of asymmetric faults is built up from a thorough understanding of symmetric faults [5].

(d) Asymmetric fault

An asymmetric or unbalanced fault does not affect each of the three phases equally. Common types of asymmetric faults, and their causes:

1- Line to Ground Fault (L — G faults)

The single line to ground fault can occur in any of the three phases. However, it is sufficient to analysis only one of the cases. Looking at the symmetry of the symmetrical component matrix, it is seen that the simplest to analyses would be the phase a. Consider an L-G fault with zero fault impedance as shown in figure 2.1 [5].

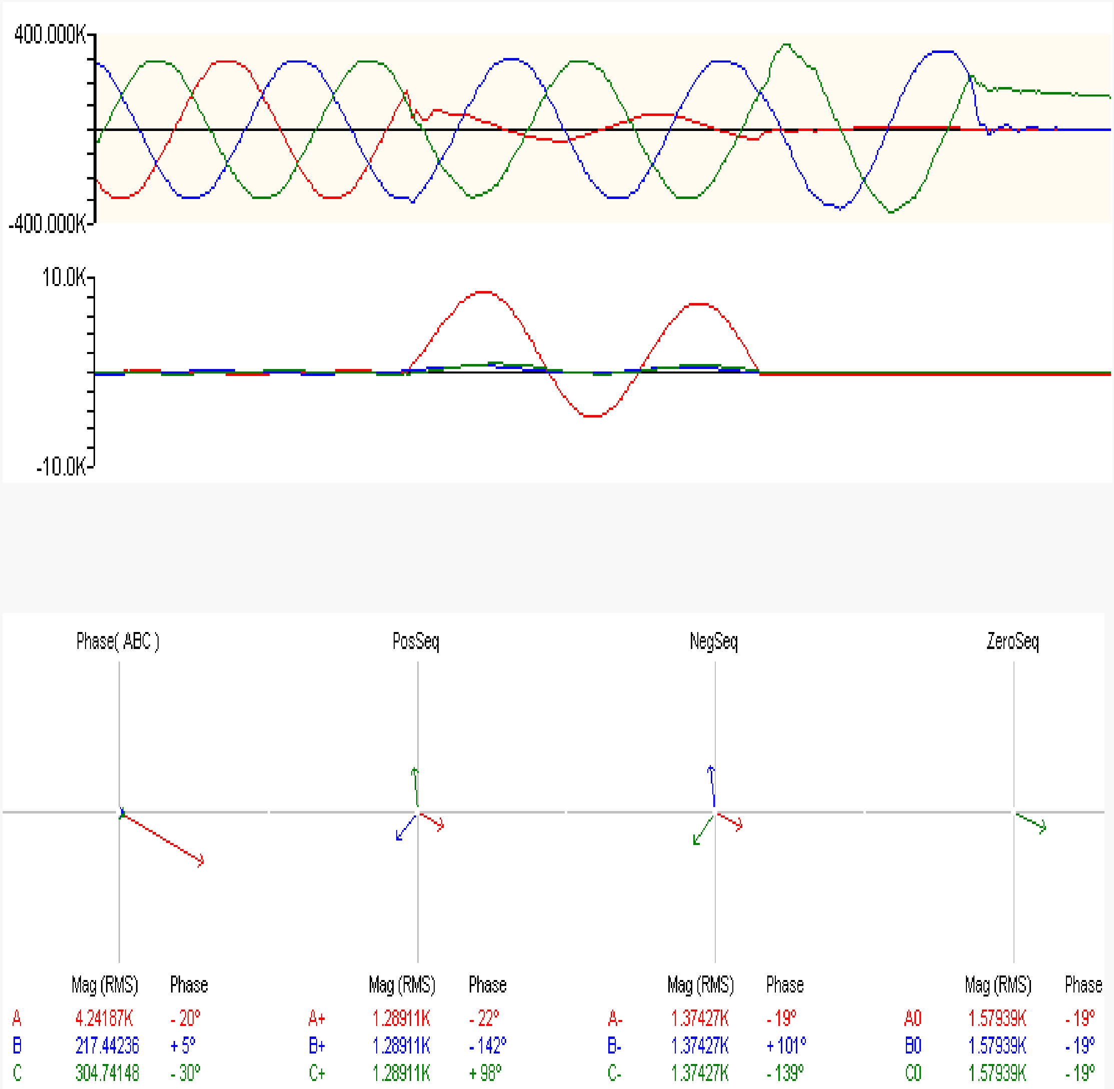


Figure 2.1: Phase A to ground fault.

Thus, for a single line to ground fault (L-G fault) with no fault impedance (Zf), the sequence networks must be connected in series and short circuited. So, for a line to ground fault we expect to see the positive, negative, and zero voltage and current components in MATLAB using DFR. We also expect to see a depression on the faulted phase voltage and a sharp increase in phase and residual current. These concepts are visualized in Figure 2.1 for an A phase to ground fault which calculated above. Notice the sharp increase in a phase current in the first phase diagram taken from the second trace [5].

2- Line to line fault (L-L faults)

A line-to-line fault is another type of unsymmetrical fault in which only positive and negative sequence components should be anticipated. For the phases involved in a phase-to-phase fault, a rapid increase in current and a voltage depression on each phase should be observed. In addition, both currents should be 180 degrees apart. This principle is illustrated in Figure 2.2 for an A-C fault.

3- Double-line to ground

A similar scenario occurs during a line-to-line-to-ground (or double-line) fault where the two involved phases will see an increase in current and a decrease in voltages. The difference is that a zero-sequence component will be present. This can be appreciated in the phase diagram in Figure 2.3.

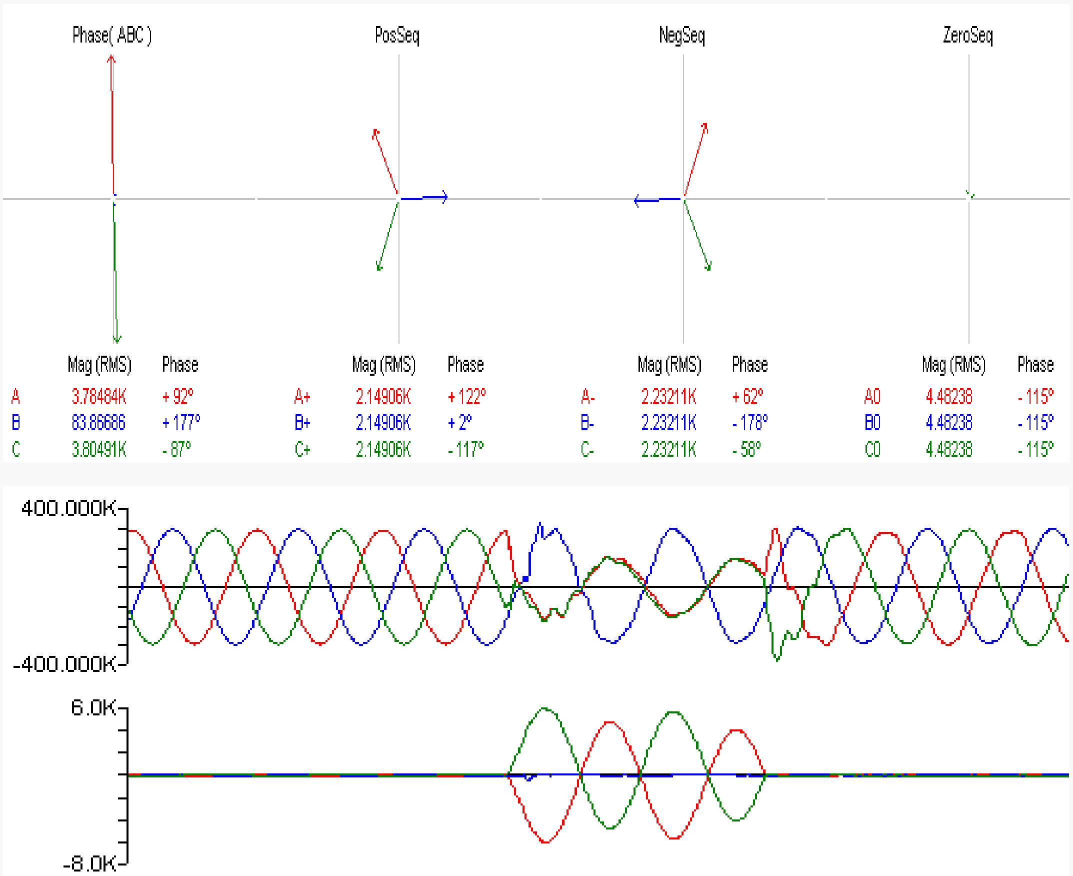


Figure 2.2: A-C Phase Fault

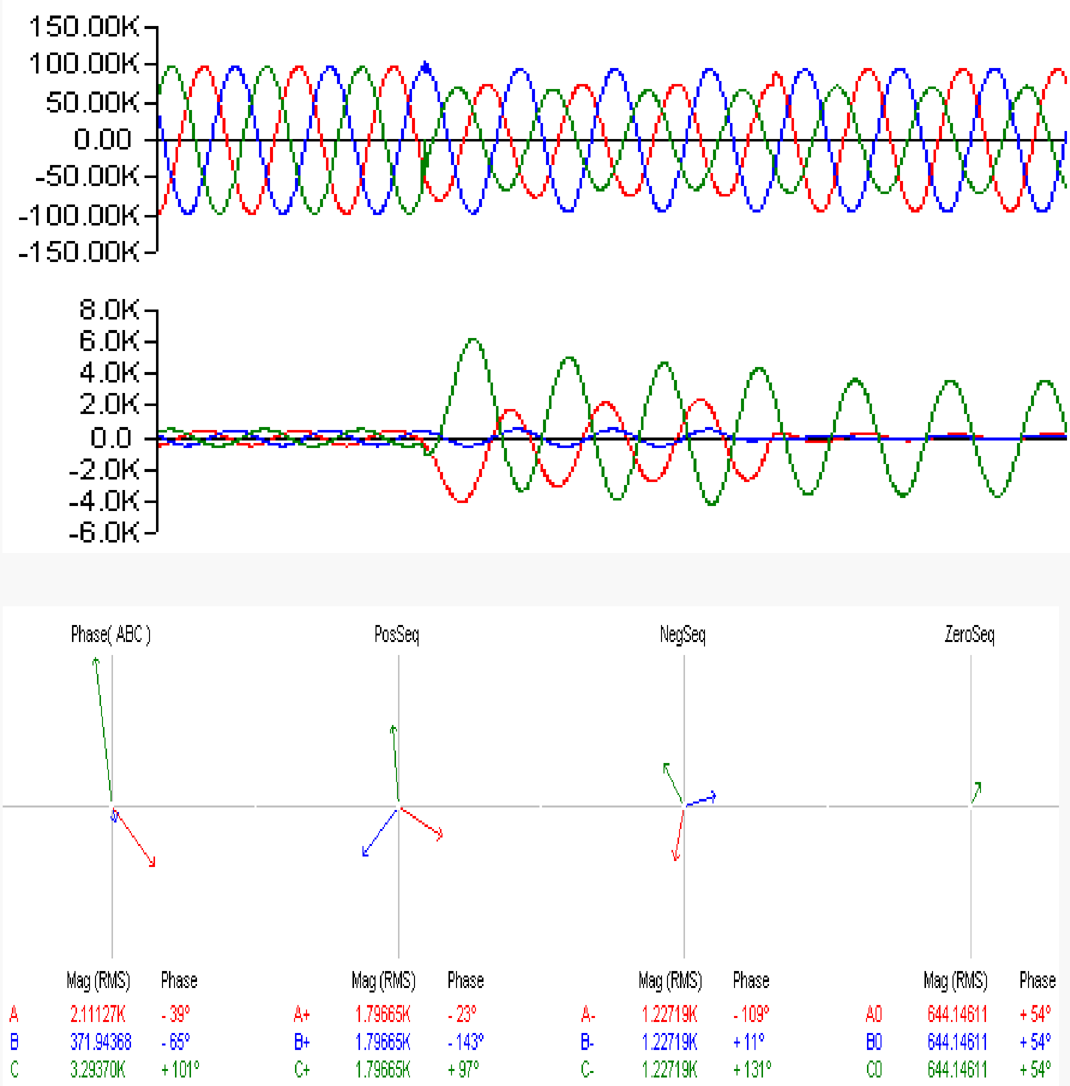


Figure 2.3: A-C-Ground Fault

4- Three Phase Fault

A three-phase fault will be reflected by a high, sharp increase in all three phase currents, and all three voltages should collapse. A three-phase fault can be seen in Figure 2.4. Upon inspection, we notice the rapid increase in all three phase currents and a depression in all three voltages. Since three phase faults are considered symmetrical, there should be no presence, or very small values, of negative and zero sequence components as seen below in the phase diagrams. In theory, the negative and zero sequence components should be zero; however, it could take some time for all three phases to become involved, and some zero and negative sequence quantities might be seen [5].

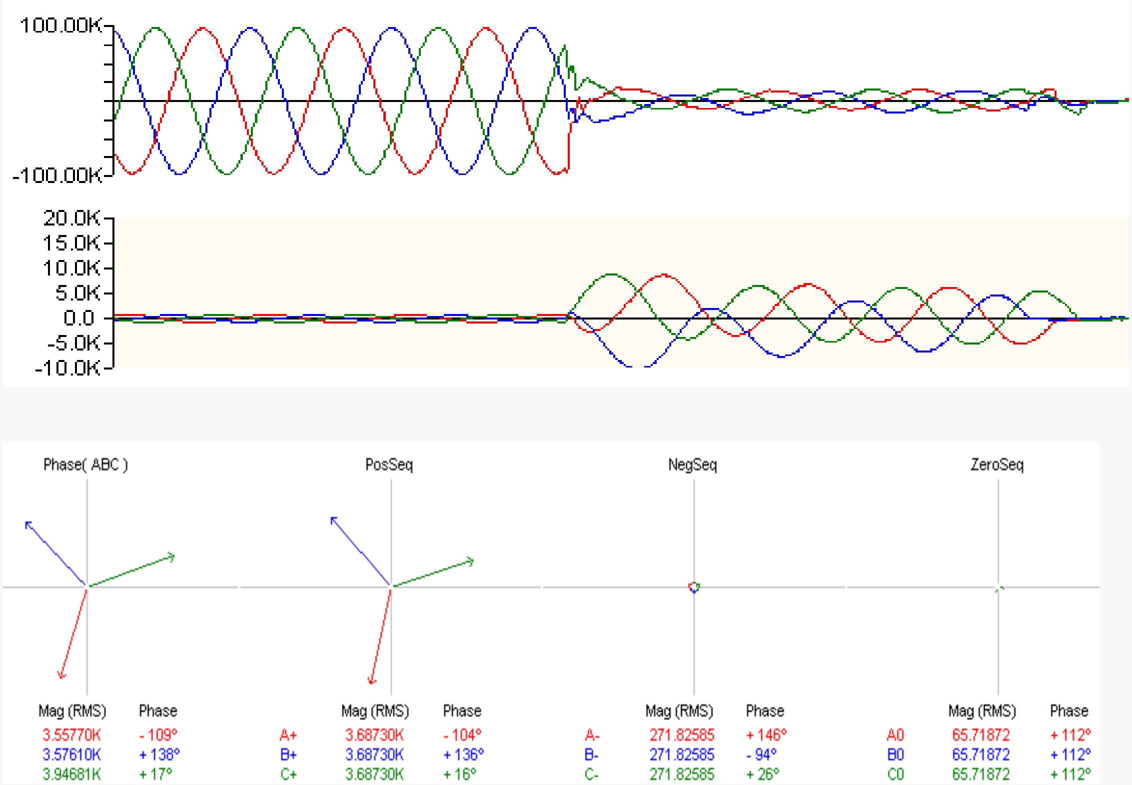


Figure 2.4: Three Phase Fault

(e) Transient fault

A transient fault is a fault that is no longer present if power is disconnected for a short time. Many faults in overhead power lines are transient in nature. At the occurrence of a fault power system protection operates to isolate area of the fault.

A transient fault will then clear and the power line can be returned to service. Typical examples of transient faults include:

- Momentary tree contact.
- Bird or another animal contact.
- Lightning strike.
- Conductor clash [6].

2.3.3 Purpose of fault recording

Fault records are one of the most important pieces of evidence that event analysts can have during system event investigations.

They can provide the reasons for premature equipment failure, supply waveforms and status of equipment behavior during an event and give necessary information to perform post-fault event analysis. Proper use and interpretation of event records can lead to corrective action for a given system problem resulting in improved performance and reliability of any generation, transmission, and distribution system. Utility engineers have to make balanced decisions as to what equipment is better to use for pre- and post-event analysis. Fault recording has been used for decades now, and it is generally used for two main purposes [9].

(a) Recording of System Events

Recording of system events can be classified as:

1- Fast transient recordings:

Relays and recorders are capable of recording fast system events such as power system faults, lighting strikes, switching events, insulator flashing, etc. These types of transient events are usually short-lived and fast; therefore, they do not require long record lengths unless faults have cascaded into multiple system elements or a fault has remained in the system longer than normal. In these cases, longer transient records are needed to capture the entire event. These types of records let the analyst know the current and voltage magnitudes, time, and duration that were observed during the course of the event. This information can then be analyzed and dissected to look for potential problems in the timing and current and voltage magnitudes. Investigation of current magnitudes can also be used to determine the deviation of actual fault values vs. calculated values from software. Short circuit databases, due to their large composition, can contain errors that yield misleading fault values. Comparing actual and calculated values is a good practice to check for possible inconsistencies. Transient records can further improve the analysis of such events mentioned above by providing the symmetrical component quantities of the current and voltage during steady state and fault conditions. The positive, negative, and zero sequence components can be used to determine their individual magnitudes during the transient event.

They can also be used to verify the type of fault. This process is further expanded in the section below about deciphering power system faults. Another type of system event is incipient faults such as early signs of insulator failure. Such conditions require longer record lengths to capture the early development of the event and are better handled by DFRs because of their record length capabilities [9].

2- Slow swing recordings:

They are designed to capture the power system's response in RMS values following a power swing or disturbance. These records can usually help to determine how well the system is designed. These types of recorders can capture the response of generators, power swings on transmission lines, load variations caused by voltage and frequency fluctuations, and transient phase angle changes. Since these records measure system response, swing recorders are required in specific spots and under different owners of an interconnected system. Swing records do not have the fast rise or sharp current changes that transient records have since they are sampled at very slow rates.

Therefore, accurate time stamps are needed to analyze system event records from many pieces of recording equipment. The records themselves need to cover a much longer period than transient records. There are some microprocessor relays capable of swing recording data, but they are limited by record length. DFRs have swing recording as part of their design and can capture incredibly long records. It is recommended to use maximum record length [11].

(b) Monitoring power system performance

Fault recording devices have proven to be invaluable assets in identifying proper as well as improper behavior of system protection schemes and associated equipment. The ability to record protection system performance such as relays, circuit breakers, and control systems has resulted in design improvements and corrections of the power system consequently.

Companies have prevented future equipment damage and failure, generating economic savings and improving the overall performance of the power system. Triggering of records should be sensitive enough to capture all local faults independent of relay response. Most importantly, the goal is to trigger for many events without resulting in local tripping so power system response can be reviewed. Secondly, it is important to capture a record of a local fault accompanied by a relay failure. Fault recorders have an advantage over recording relays in this regard [12].

2.3.4 Digital fault recorder vs. digital relays

Many people have questioned the value of digital fault recorders in an era of digital relays with recording capabilities. Nevertheless, digital fault recorders offer far advanced recording capabilities which results in better analysis of system problems and economic savings.

The main advantages of faults recorders are:

- They are independent of a failed or partially failed relay that a DFR maybe monitoring.
- They do not filter analog signals as many digital relays do.
- They offer more memory capacity, enabling longer records.
- They have faster sampling rates.
- They have broader frequency response.
- They are designed with more triggering options.
- They can monitor many power system components simultaneously.
- They can be used to monitor power quality issues, especially with connections with wind farms, static VAR generators, arc furnaces, and variable frequency drives.
- They are useful in studying problems associated with current inrush where large autotransformers are applied in parallel combinations.

• They offer a wide spectrum of system responses during faults [11].

2.4 Oscillography and Components

The analysis of oscillographic data requires an excellent understanding of the recording equipment, protection system schemes, and behavior of power system elements. Understanding some of the recording equipment's oscillography characteristics can prove to be helpful when interpreting fault records. Concepts such pre-post triggering characteristics, time frames, fundamental RMS versus true RMS, sampling rates, and time synchronization are given below [13].

2.4.1 Pre-post Initiation of fault data

The power system operates under steady state conditions when equal amounts of power are generated and consumed. This assumes that the system is working under its voltage, current, and element limits. We can also deduce that due to vast geographical exposure of the power system, this normal or steady state operation can be interrupted in certain parts of the system. No matter how well the system is designed, faults are going to occur for many reasons such as equipment failure, acts of nature, operator or technician errors, etc. The vital evidence that tells what happens during the fault is found in the pre-trigger, event duration, and post fault information of the captured record. Figure 2.5 shows these characteristics.

There are other pieces of information such as Supervisory Control and Data Acquisition (SCADA) logs that can also aid in the analysis of an event [13].

2.4.2 Pre-triggering of the Fault data

Pre-triggering of the record's fault section is identified by all the voltages, currents, and sequence of events that existed during the steady state conditions before the inception of the fault. The system voltages and currents should reflect a balanced system except for normal unbalances caused by changes in load demand. The pre-trigger information is critical during analysis of an event since any loss of waveform data can delay event investigations.

Therefore, it is recommended to use the recording equipment's maximum available pre-trigger settings.

Post-triggering indicates the beginning of the event duration. It is triggered immediately after an event detector such as over-current, under-voltage, or impedance, has indicated a fault inception. For proper fault analysis, the record should capture plenty of pre-fault data, total fault duration, clearing time of the fault, the magnitude of the fault voltages and currents, type of fault, and digital signals [13]. For transmission line faults it is desired to capture the fault from its inception through its clearing time and reclosing for about45 cycles.

The duration characteristics of the pre-trigger, duration, and post-trigger of an event record depend on the type of record and application for the analysis.

• They are useful in studying problems associated with current inrush where large auto transformers are applied in parallel combinations.

• They offer a wide spectrum of system responses during faults [13].

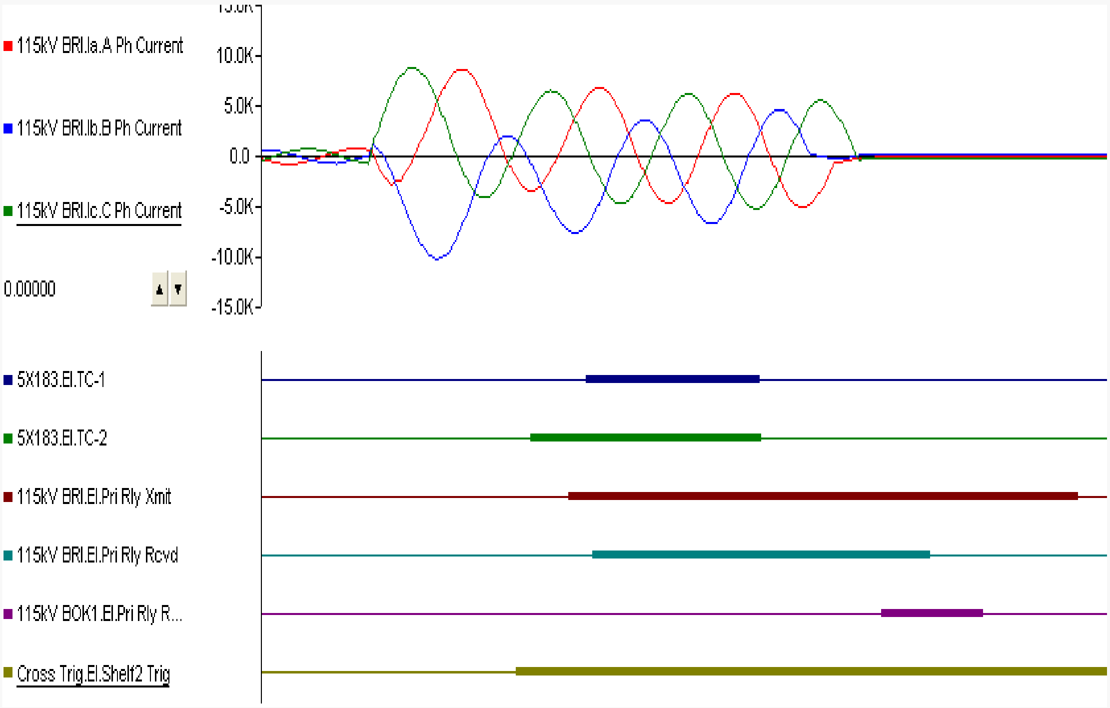


Figure 2.5: Characteristics of an oscillographic record.

2.5 Serial Communication

In telecommunication and computer science, serial communication is the process of sending data one bit at a time, sequentially, over a communication channel or computer bus. This is in contrast to parallel communication, where several bits are sent as a whole, on a link with several parallel channels. Serial communication is used for all long-haul communication and most computer networks, where the cost of cable and synchronization difficulties make parallel communication impractical. Serial computer buses are becoming more common even at shorter distances, as improved signal integrity and transmission speeds in newer serial technologies have begun to outweigh the parallel bus's advantage of simplicity [14].

2.5.1 Sampling rate

The sampling rate, sample rate, or sampling frequency (fs) defines the number of samples per unit of time (usually seconds) taken from a continuous signal to make a discrete signal. For time-domain signals, the unit for sampling rate is hertz (inverse seconds, I/s, s-1), sometimes noted as Sa/s (samples per second). The inverse of the sampling frequency is the sampling period or sampling interval, which is the time between samples [14].

2.5.2 Baud rate

Traditionally, a baud rate represents the number of bits that are actually being sent over the media, the baud count includes the overhead bits start, stop and parity that are generated by the sending a Universal Asynchronous Receiver - Transmitter (UART) and removed by the receiving UART. This means that seven-bit words of data actually take 10 bits to be completely transmitted. Therefore, a modem capable of moving 300 bits per second from one place to another can normally only move 30 7-bit words if parity is used and one start and stop bit are present. Most typical standard baud rates are: 300, 1200, 2400, 9600, 19200, etc. However, any baud rate can be used. This parameter affects both the receiver and the transmitter [10].

2.5.3 RS-232 Protocol

It is a standard interface approved by the Electronic Industries Association (EIA) for connecting serial devices. In other words, RS-232 is a long-established standard that describes the physical interface and protocol for relatively low-speed serial data communication between computers and related devices. The RS-232 standard supports two types of communication protocols Synchronous and asynchronous.

Using the synchronous protocol, all transmitted bits are synchronized to a common clock signal. The two devices initially synchronize themselves to each other, and then continually send characters to stay synchronized. Even when actual data is not really being sent, a constant flow of bits allows each device to know where the other is at any given time. That is, each bit that is sent is either actual data or an idle character. Synchronous communications allow faster data transfer rates than asynchronous methods, because additional bits to mark the beginning and end of each data byte are not required. Using the asynchronous protocol, each device uses its own internal clock resulting in bytes that are transferred at arbitrary times. So, instead of using time as a way to synchronize the bits, the data format is used.

In particular, the data transmission is synchronized using the start bit of the word, while one or more stop bits indicate the end of the word. The requirement to send these additional bits causes asynchronous communications to be slightly slower than synchronous. However, it has the advantage that the processor does not have to deal with the additional idle characters. Most serial ports operate asynchronously. Figure 2.6 shows the RS-232 connector 9 pins.

RS232 used to interface our microcontroller with PC through serial communication. But The RS-232 standard specifies that the maximum length of cable between the transmitter and receiver should not exceed 100 feet, although in practice many systems are used in which the distance between transmitter and receiver exceeds this rather low figure.

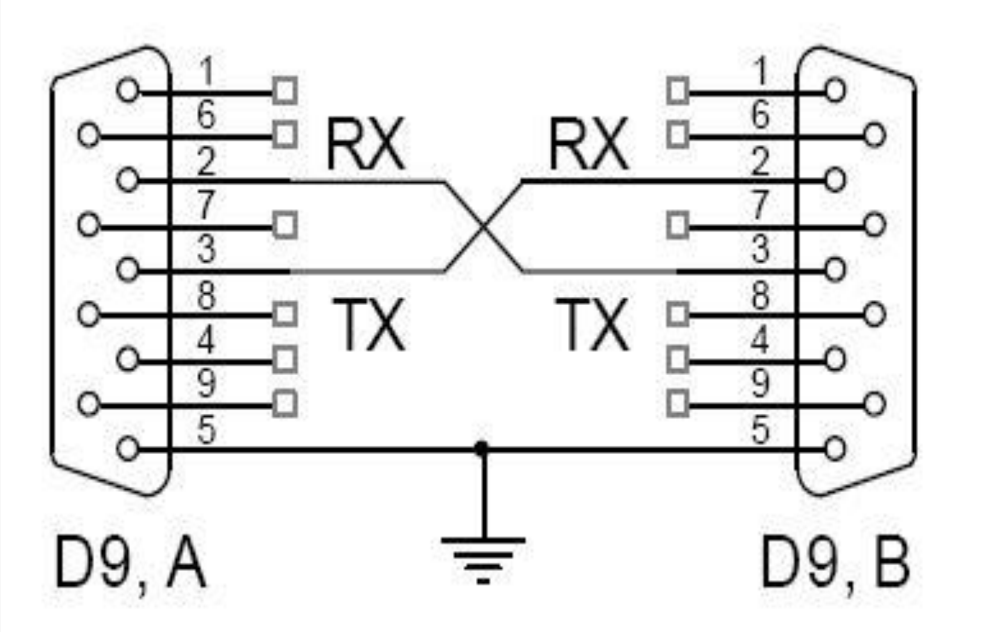


Figure 2.6: RS-232 connector 9 pins

The limited range of the RS-232 standard is one of its major shortcomings compared with other standards which offer greater ranges within their specifications. [10]. Figure 2.7 shows the RS-232 interface.

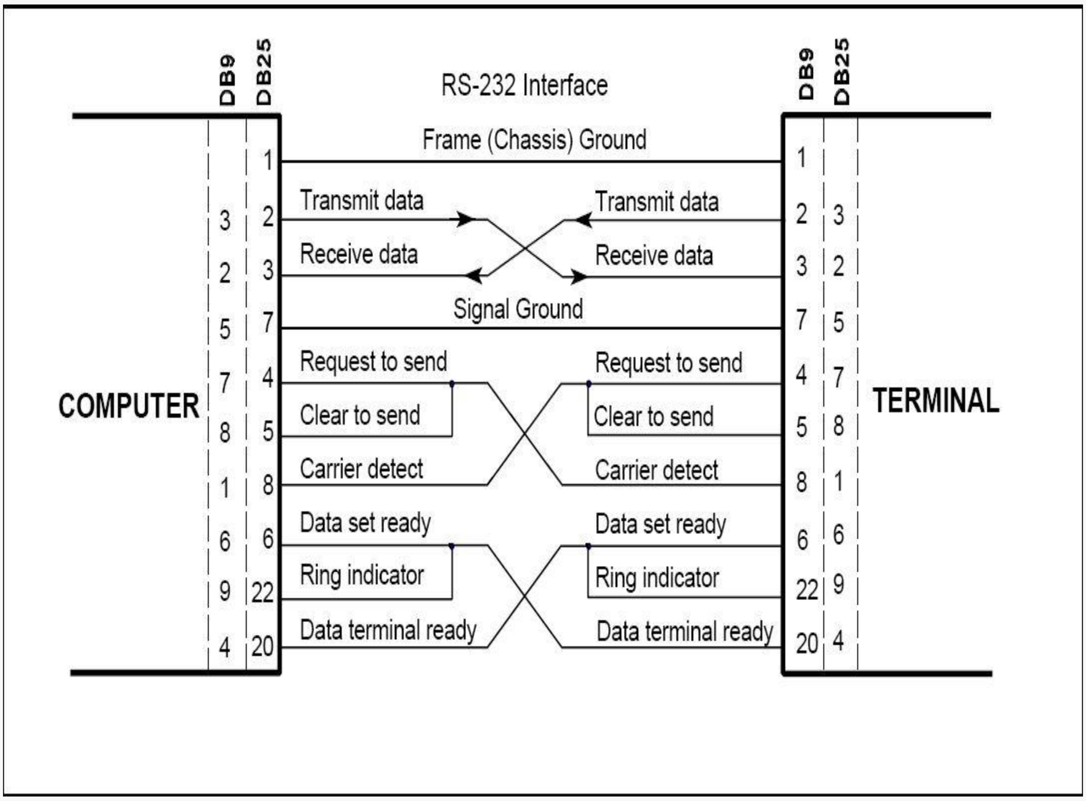


Figure 2.7: Rs-232 interface

2.6 Distributed Digital Fault Recorder

The Distributed Digital Fault Recorder (DDFR) is a digital fault and disturbance recording system that uses communications to retrieve Fault, disturbance, and sequence of event records that are captured by the protection relays distributed throughout a substation. The DDFR internally stores this critical substation information for local station troubleshooting as well as archives this data in a remote enterprise network location for permanent storage and analysis [10].

2.6.1 Data recording

The DDFR will detect that new information has been recorded in a relay and automatically retrieve and archive this data. Information that will be archived from protection relays include:

- Sequence of event records.
- Transient fault records.
- Disturbance records.

Figure 2.8 shows the disturbance digital recording system.

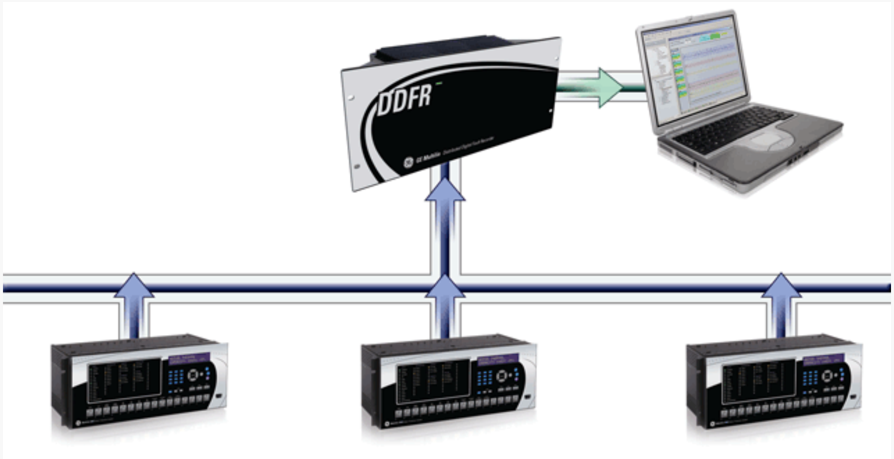


Figure 2.8: Disturbance Digital recording

The DDFR is a transient fault, sequence of event, and disturbance record archiving system that will seamlessly make available all of the recorded power system information necessary for performing post mortem fault and disturbance analysis [10].

1- Sequence of event record

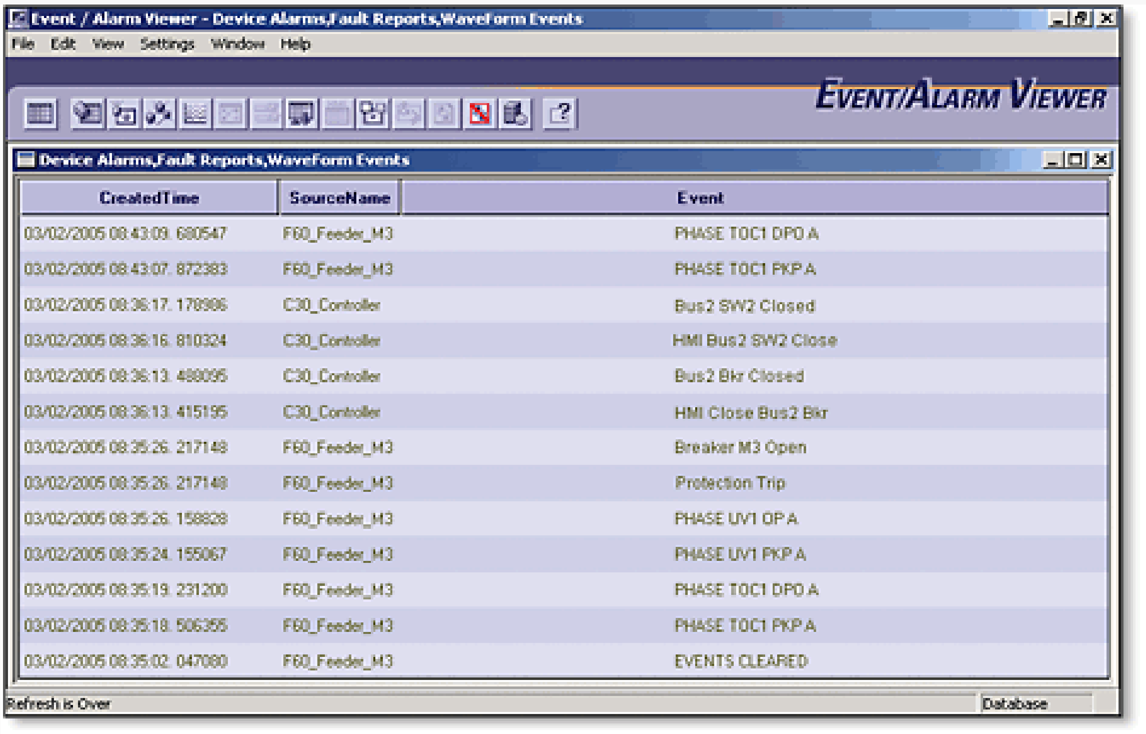
The DDFR will retrieve the Event Records stored in protection relays distributed across a substation and merge all of these Events into a Station-Wide Sequence of Event Record. Connecting the protection relays to an IRIG-B time clock will synchronize the relay internal clocks with 1 millisecond accuracy and enable the DDFR to create a station-wide event record that will be an exact representation of the sequence of operations that occurred in the substation [10].

2- Transient fault records:

Transient fault records (also known as oscillography records) that are recorded in protection relays will be retrieve by the DDFR and stored in its internal memory space. All transient fault records will be archived using a naming convention that will make it easy for users to relate the transient fault records with a particular substation fault. Each record will be stored with a name that includes the exact date and time that the record was initially triggered. If the transient fault record was originally recorded in the relay as a COMTRADE file, the DDFR will store this record in its native format; if the protection relay stores its record in another format, such as a CSV file, the DDFR will automatically convert this file into the COMTRADE-1999 format so that the record can be analyzed using a standard COMTRADE viewer [14].

3- Disturbance records:

Protections relays that are capable of recording disturbance records that are used for analyzing extended or evolving power system disturbances such as voltage sags or swells will be retrieved and archived be the DDRF. All disturbance records will be archived using a naming convention that will make it easy for users to relate the disturbance records with a particular power system disturbance. Each record will be stored with a name that includes the exact date and time that the record was initially triggered. Disturbance records will be stored in COMTRADE format and if needed, first convert this file to a COMTRADE file from its native format. Figure 2.9 shows the event alarm viewer and fault reports.



Figured 2.9: event alarm viewer and fault reports

In fact, major power system failures during a transient disturbance are more likely to be caused by unnecessary protective relay tripping rather than by the failure of a relay to take action. In other words, the performance of protective relay or system is very important to be known especially in smart power grid. New relay designs based on microprocessors are currently being developed and marketed. Some of these relays have the ability to extract and store relay input signals in digital form and transmit this data to another device. For performing this function, the relay is like digital fault recorder, except that the nature of the recorded data may be affected by the needs of the relaying algorithm. As the digital fault recorders, record format and other parameters have been standardized [14].

2.6.2 Electromechanical Relays

Electromechanical protective relays are operated by magnetic induction. In this an electromagnet is formed by a coil of wire wound around an iron core which will move an armature that is connected to the switch of the controlled circuit. If a relay is not energized its armature won't control the switch. When current exceeds the set current in the coil the armature will move and control the switch as long as it is energized [17].

The main disadvantages of electromechanical relay are:

1) Electromechanical Relay uses mechanical parts that makes it bulky and larger in size. Flag system is used to tell whether the relay is activated or not.
2) It is not flexible as we cannot modify its characteristics and functional operations unlike in software supported numerical relay.
3) It is not as reliable as the numerical relay.
4) It does not provide multifunctional operations to control various features related to fault.
5) Auto-resetting is not possible in these relays.
6) It does not have memory to record fault related data [17].

2.6.3 Static relays

It uses electronic amplifiers like vacuum tube amplifiers. It has no moving mechanical parts unlike in electromechanical relay. It uses analogue electronic devices instead of magnetic coils and mechanical parts to obtain the relay characteristics [17].

2.6.4 Numerical Relays

Numerical relays are microprocessor and microcontroller-based relays having its own memory numeric relays take the input analog quantities and convert them to numeric values. Electromechanical and static relays are not multifunctional unlike numerical relay [17].

The main Advantages of Numerical Relays are:

1) Compact Size: Numerical relay is compact in size and uses LCD to indicate relay activation. It requires less wiring so it is not having complex architecture.

2) Flexibility: We can modify its functional operation by changing codes in software.

3) Reliability: It is more reliable because of less interring, use of less components and reduced component failures.

4) Multi-Functional Capability: Displaying results and data in LCD, recording fault related data etc. makes it a multi-functional in its operation.

5) Different Types of Relay Characteristics: We can get Definite Time Characteristics of different time values and Indefinite Time Characteristics of various values from it as they are stored in the microcontroller memory.

6) Digital Communication Capabilities: It is easily interfaced with different digital equipment’s.

7) Sensitivity: It has high sensitivity and pickup ratio.

8) Speed: It has the highest speed of operation among other relays.

9) Data History: It has memory of its own so it can record the various details of faults like nature, magnitude and duration of fault.

10) Auto Resetting and Self Diagnosis: It can decide whether normal condition has arrived after the fault [17].

CHAPTER THREE

MOTHODOLOGY

3.1 Introduction

Fault records are one of the most important pieces of evidence that event analysts can have during system event investigations. They can provide the reasons for premature equipment failure, supply waveforms and status of equipment behavior during an event and give necessary information to perform post-fault event analysis. Proper use and interpretation of event records can lead to corrective action for a given system problem resulting in improved performance and reliability of any generation, transmission, and distribution system. In this chapter review the flow charts and circuits of the design and what we do in particle side using the hardware components and software programs that implementation our simple DFR [14]. Figure 3.1 shows the overall view of design DFR.

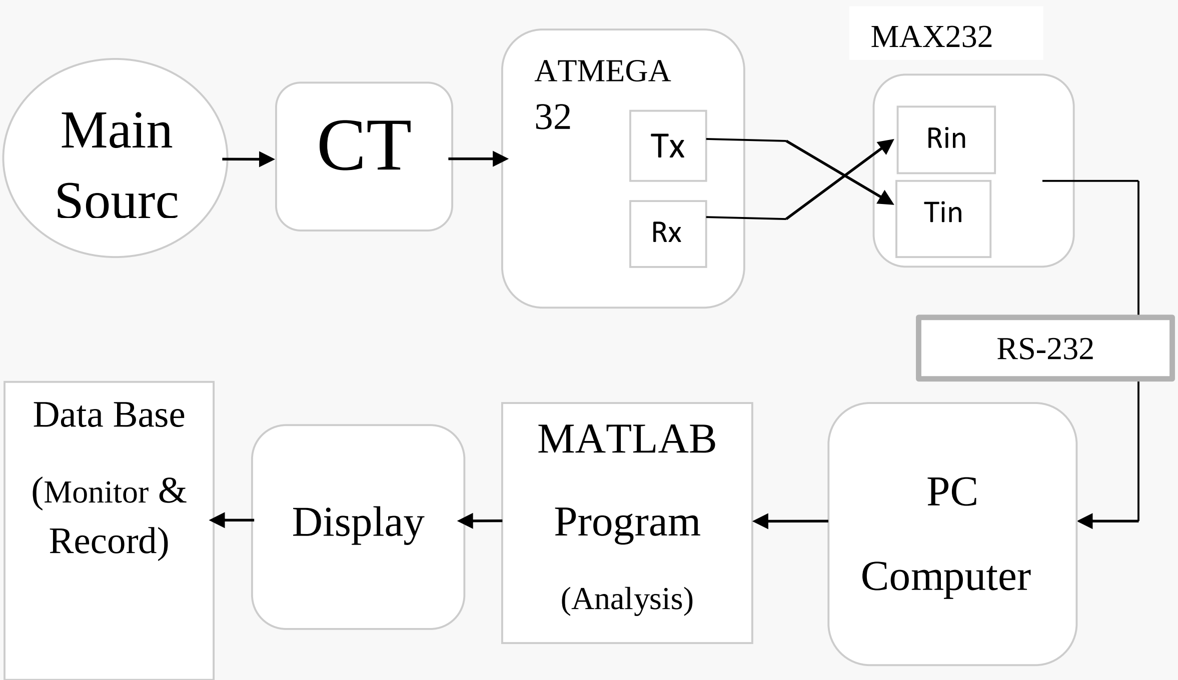


Figure 3.1: overall view of design DFR

3.2 Deal with Main Currents and Voltages from Source

Most of the 33/11 kV transformers are up to 20MVA. That would give you around 1kA at bus section. Now this bus section feeds many feeders (generally 4 to 6). Assuming 4 such feeders being supplied the current would be around 250A for a single feeder. But that’s not strict and depends upon the load being supplied by the feeder [27].

3.2.1 Current transformer

Because the microcontroller does not accept more than 5 volt, so the current is taken from the transmission li ne which its voltage is normally equal to 240 volt must be step down to amount that acceptable to microcontroller. Here used current transformer (100 A to 5A). The stepped current 5A converted to voltage parameter. This done by putting resistance before the analog signal enters to microcontroller to make Analog to Digital (A/D) conversion as shown in Figure 3.3 [15].

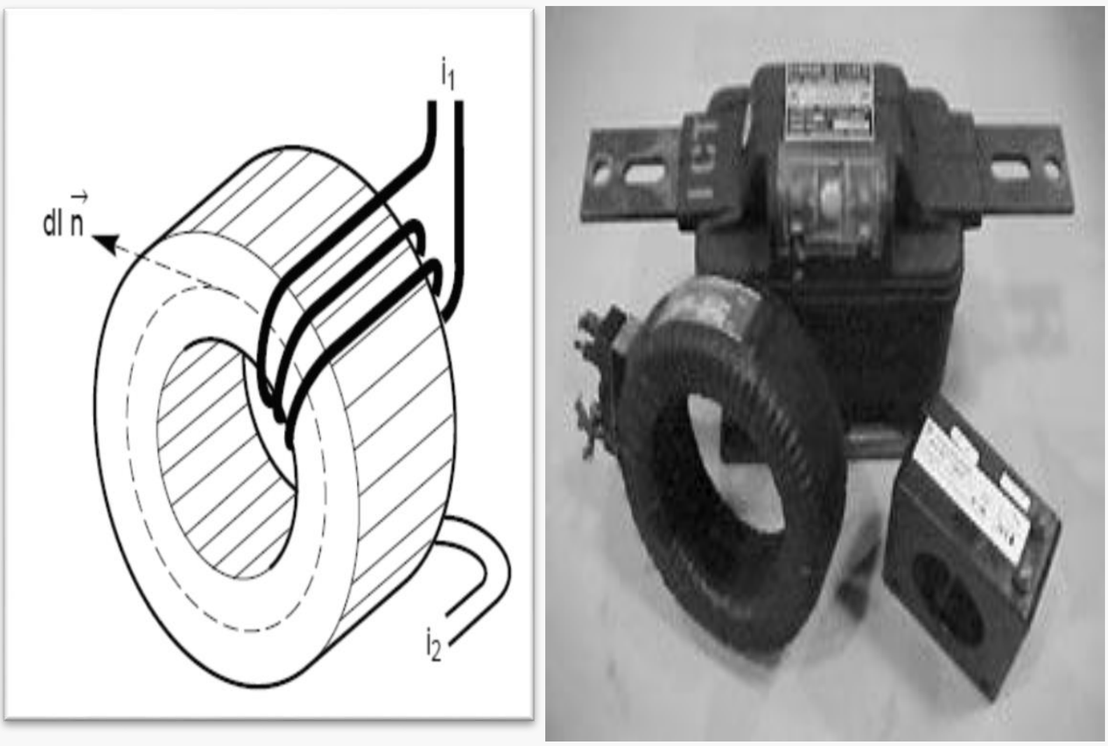


Figure 3.2: Current transformer

3.2.2 AC voltage shifting

A microcontroller is one of the Integrated Circuits (ICs) that in requiring a uni-polar AC voltage signal, the signal may be delivered from a bi-polar voltage amplifier/generator and repositioned relative to a zero-volt reference (an earth ground) using a voltage level shifting circuit. Figure 3.3 shows the AC voltage level shifting.

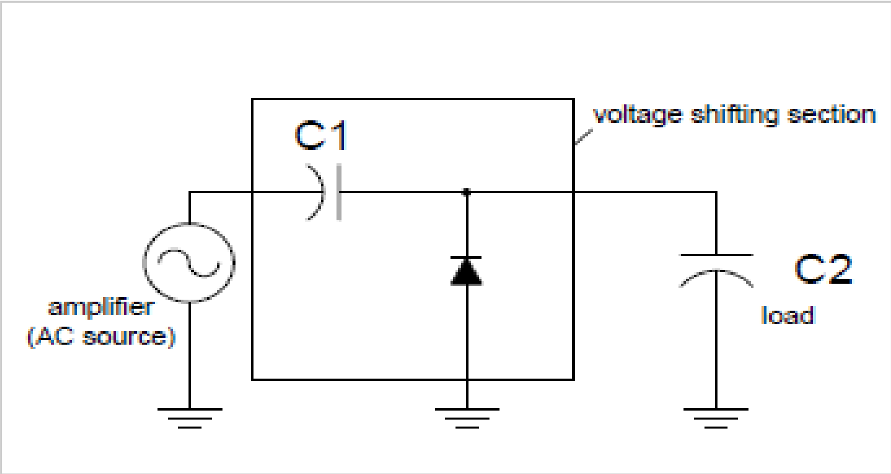


Figure 3.3: AC voltage level shifting

An example of an AC voltage shifting circuit diagram used in this process (Villard circuit) is shown in Figure 3.4 this Figure shows the input and the output voltage of the circuit [15].

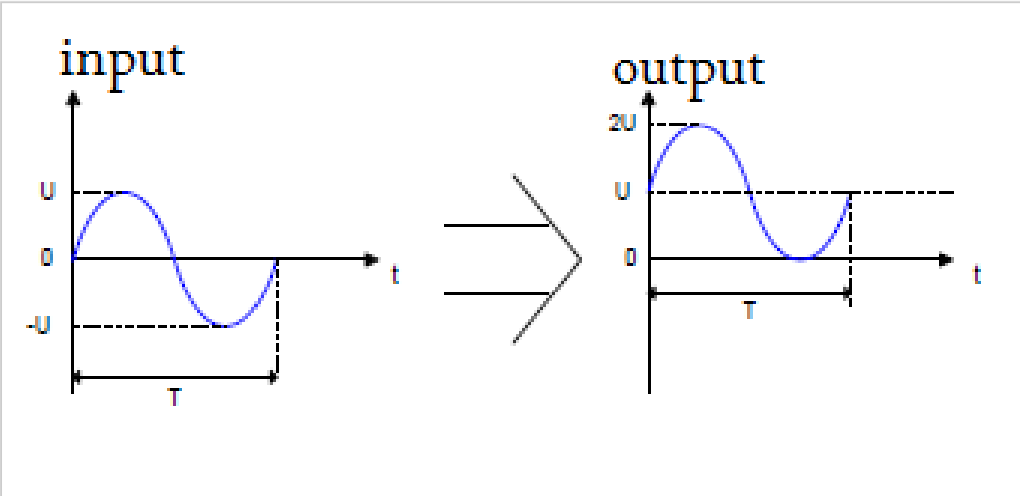


Figure 3.4: input and output AC signal through shifter

3.3 The Arduino

Arduino is a single-board microcontroller to make using electronics in multidisciplinary projects more accessible. It is an open-source electronics platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on a Light Emitted Diode (LED), publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on Processing. All Arduino boards are completely open-source, empowering users to build them independently and eventually adapt them to their particular needs. The software, too, is open-source, and it is growing through the contributions of users worldwide [24]. Figure 3.5 shows the hardware of Arduino.

Figure 3.5: Hardware of Arduino

3.3.1 Technical specifications

- Microcontroller: ATMEGA32.
- Operating voltage: 5V.
- Input voltage (recommended): 7-12V.
- Input voltage (limits): 6-20V.
- Digital Input/Output (I/O) pins: 14.
- Analog input pins: 6.
- Direct Current (DC) per I/O pin: 40 mA.
- DC for 3.3V pin: 50 mA.
- Flash memory: 32KB.
- Electrically Erasable Programmable Read Only Memory (EEPROM): 1KB.

3.3.2 Power

The Arduino Uno can be powered via the Universal Serial Bus (USB) connection or with an external power supply. The power source is selected automatically. External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. The board can operate on an external supply of 6 to 20V. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12V [24].

The power pins are as follows:

VIN: The input voltage to the Arduino board when it's using an external power source (as opposed to 5V from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.

5V: The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator or be supplied by USB or another regulated 5V supply.

3V3: A 3.3-volt supply generated by the on-board regulator. Maximum current draw is 50 mA.

GND: Ground pins.

3.3.4 The input and output pins

Figure 3.6 shows the electrical inputs and outputs pins. Each of the 14 digital pins on the Uno can be used as an input or output, using pinMode(), digitalWrite(), and digitalRead() functions. They operate at 5V. Each pin can provide or receive a maximum of 40mA and has an internal pull-up resistor (disconnected by default) of 20-50 KΩ. In addition, some pins have specialized functions:

- **Serial: 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 ATmega8U2 USB-to-TTL Serial chip.
- **External Interrupts: 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.
- **LED: 13:** There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

There are a couple of other pins on the board:

- **AREF:** Reference voltage for the analog inputs. Used with analogReference().
- **Reset:** Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board [22].

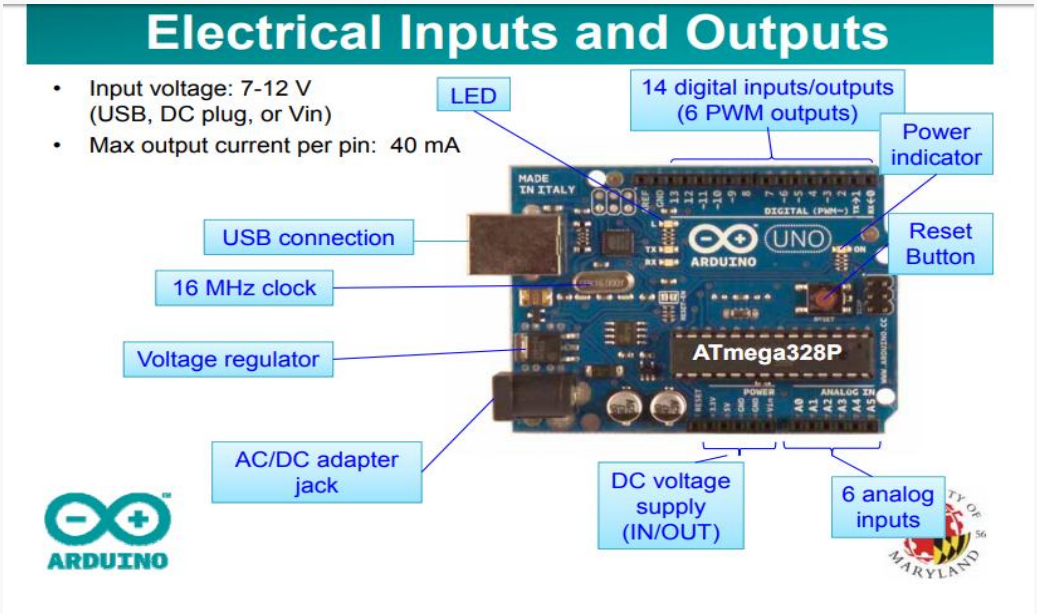


Figure (3.6): Arduino structure

3.4 Microcontroller

The microcontroller incorporates all the features that are found in microprocessor. The microcontroller has built in Read Only Memory (ROM), Random Access Memory (RAM), I/O ports, serial port, timers, interrupts and clock circuit. A microcontroller is an entire computer manufactured on a single chip. Microcontrollers are usually dedicated device is embedded within an application. For example, microcontrollers are used as engine controllers in automobiles and as exposure and focus controllers in cameras [20].

3.4.1 Uses of microcontroller

Microcontrollers are used in a wide number of electronic systems such as:

- Control systems in industries.
- Electronic measurement instruments.
- Printers.
- Mobile phones.
- Security systems.
- Hearing aids.
- TV, Radio and CD players. Figure 3.7 shows the ATMEGA 32 structure.

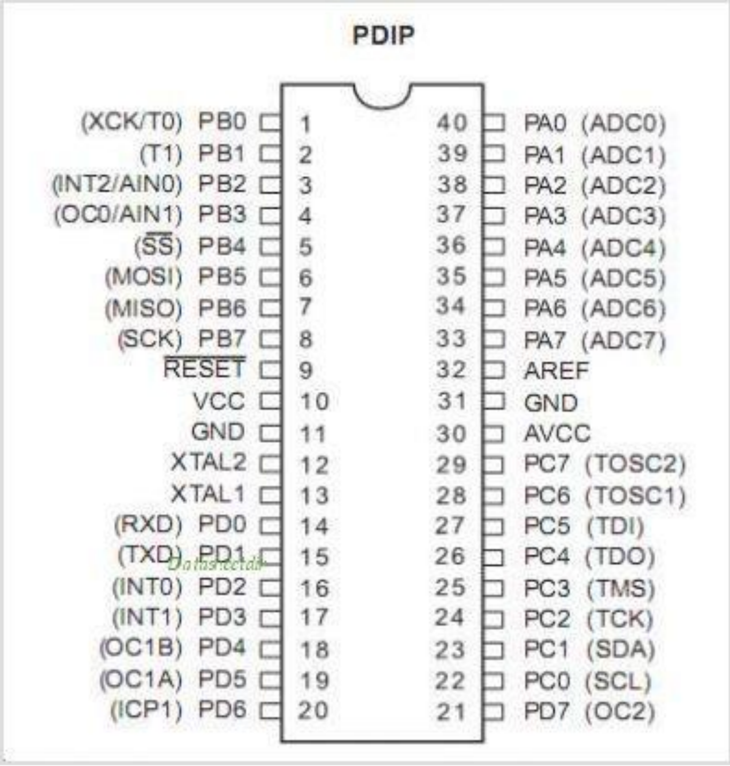


Figure (3.7): ATMEGA 32 structure

3.4.2 Features

• 10-bit resolution.

• ±2 LSB absolute accuracy.

• 13µs - 260µs conversion time.

• 8 multiplexed single ended input channels.

• Optional left adjustment for ADC result readout.

• 0 - VCC ADC input voltage range.

• Selectable 2.56V ADC reference voltage.

• Free running or single conversion mode.

• ADC start conversion by auto triggering on interrupt sources.

• Sleep mode noise canceler [20].

3.4.3 Analog to digital converter

Most of the physical quantities around us are continuous. By continuous we mean that the quantity can take any value between two extremes. To monitor, control and perform any action to the system equipments we have this analogue signal into digital form. For this an ADC or analog to digital converter is needed. Most modern Microcontroller Unit (MCU) including an Audio/Video Receivers (AVRs) has an ADC on chip, so from here ATMEG32 is used for this purpose. After initiates ATMEGA 32 by setting its clock 4MHz, selects the reference to AVCC with external source 5 volt and decoupled by an external capacitor at the AVCC pin to improve noise immunity. Setting ADMUX and ADCSRA registers to start conversion [20].

3.4.4 Voltage converter:

The +5V to ±10V conversion is performed by dual charge-pump voltage converters. The first charge-pump converter uses capacitor C1 to double the +5V into +10V, storing the +10V on the output filter capacitor, C3. The second uses C2 to invert the +10V into -10V, storing the -10V on the V- output filter capacitor, C4. In shutdown mode, V+ is internally connected to VCC by a 1k pull-down resistor, and V- is internally connected to ground by a 1kΩ pull up resistor [1]. Figure 3.8 show the Flow chart of sending data to Personal Computer (PC).

3.5 Serial Communication Setting:

- Baud Rate 9600 (bits per second).
- Parity: none.
- Data bit: 8bit.
- With 4 MHz clock.
- Communication port: COM1.
- If ADC complete, send the converted value directly to PC.

3.6 Max232

The MAX23 shown in Figure 3.9 is an integrated circuit, first created by Maxim Integrated Products, that converts signals from an RS-232 serial port to signals suitable for use in TTL compatible digital logic circuits. The MAX232 is a dual driver/receiver and typically converts the RX, TX, CTS and RTS signals.

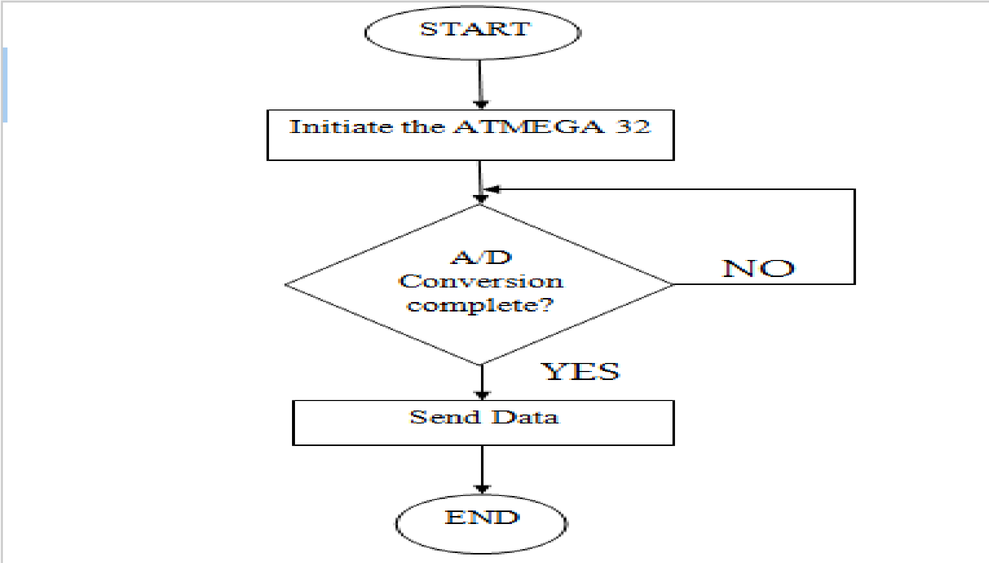


Figure 3.8: Flow chart of sending data to PC

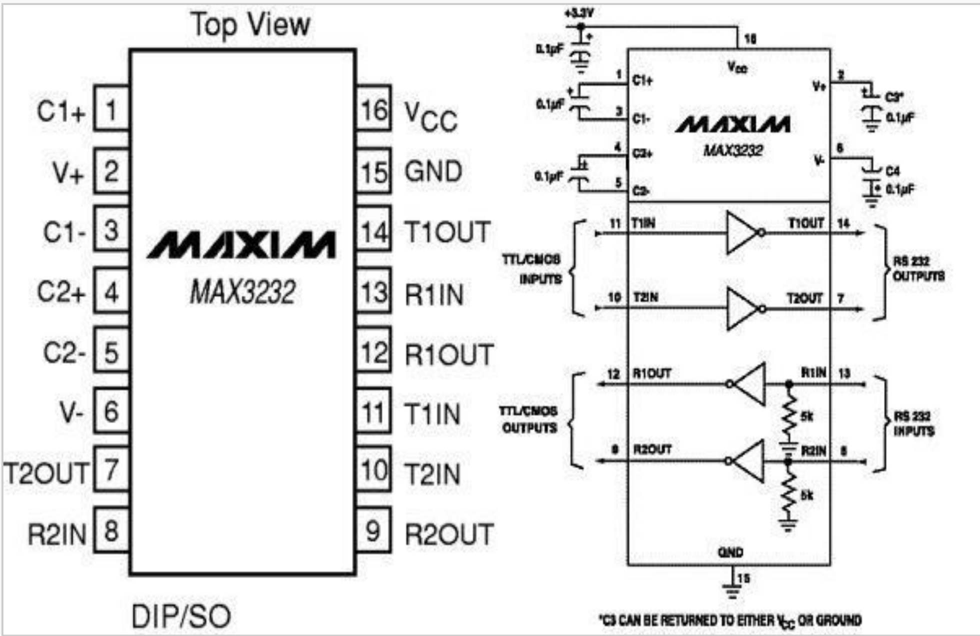


Figure 3.9: MAX232 pins description

The drivers provide RS-232 voltage level outputs (approx. ± 7.5V) from a single + 5V supply via on-chip charge pumps and external capacitors. This makes it useful for implementing RS-232 in devices that otherwise do not need any voltages outside the 0V to + 5V range, as power supply design does not need to be made more complicated just for driving the RS-232 in this case [21].

The later MAX23A is backwards compatible with the original MAX232 but may operate at higher baud rates and can use smaller external capacitors – 0.1μF in place of the 1.0μF capacitors used with the original device [1]. The newer MAX3232 is also backwards compatible, but operates at a broader voltage range, from 3 to 5.5V [21].

Figure 3.10 represents interface ATMEGA32 with PC through RS232 protocol and MAX232.

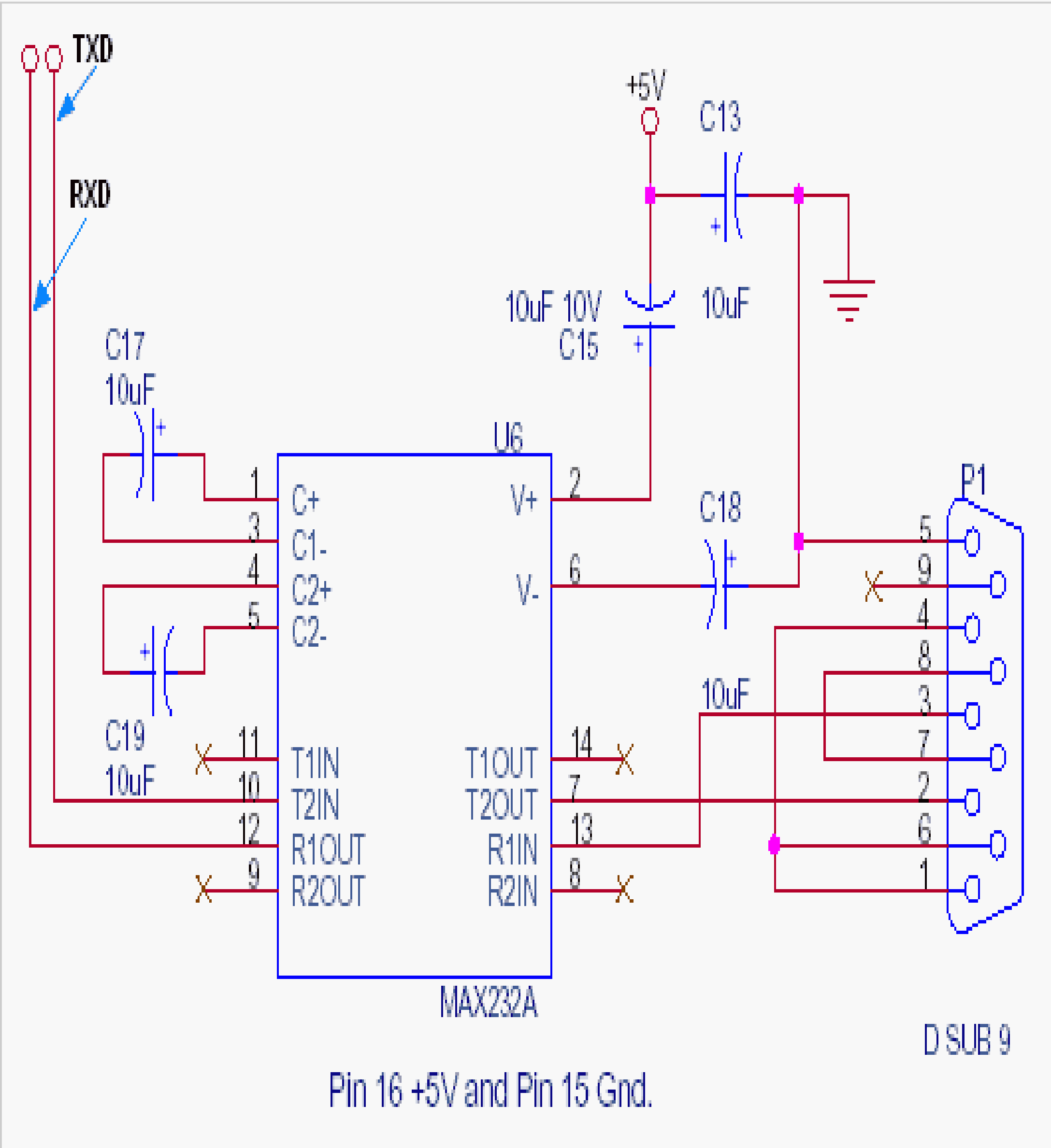


Figure 3.10: interface ATMEGA32 with PC through RS232 protocol and MAX232

3.7 Software Tools

The main software tools are:

3.7.1 Arduino compiler

The Arduino Integrated Development Environment (IDE) which is shows in Figure 3.11 used to create, open, and change sketches (Arduino calls programs "sketches".) Sketches define what the board will do [19].

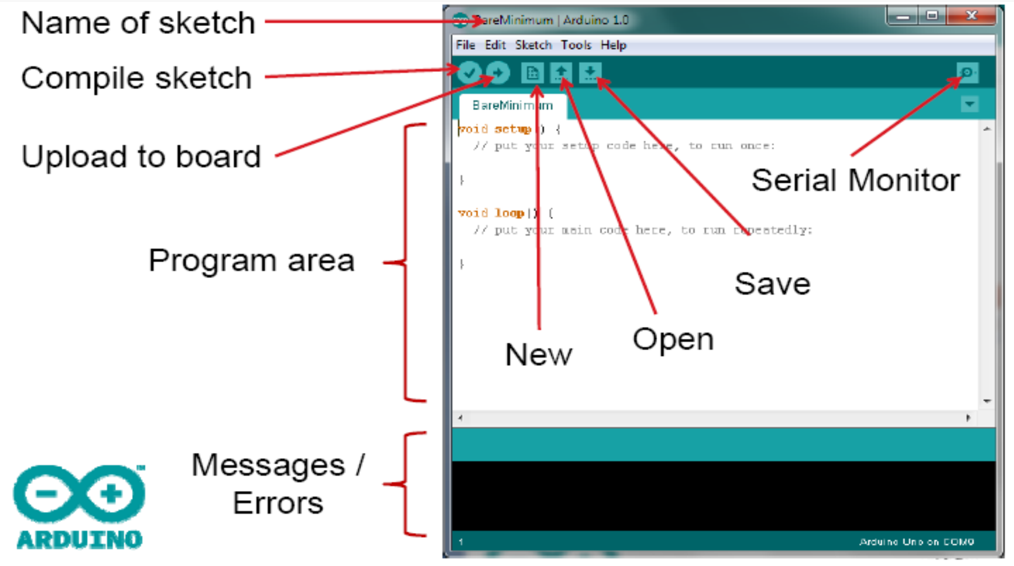


Figure 3.11: Integrated Development Environment

3.7.2 Elements of the IDE

- **Compile sketch:** Before the program "code" can be sent to the board, it need to be converted into instructions that the board understands. This process is called compiling.
- **Create new sketch:** This opens a new window to create a new sketch.
- **Open existing sketch:** This loads a sketch from a file on your computer.
- **Save sketch:** This saves the changes to the sketch you are working on.
- **Upload to board:** This compile and then transmits over the USB cable to your board.
- **Tab button:** This lets you create multiple files in your sketch. This is for more advanced programming than we will do in this class.

- **Sketch editor:** This is where you write or edit sketches.
- **Text console:** This shows you what the IDE is currently doing and is also where error messages display if you make a mistake in typing your program (Often called a syntax error).
- **Line number:** This shows you what line number your cursor is on. It is useful since the compiler gives error messages with a line number [22].

3.7.3 Micro C code

The Arduino (IDE) is a cross application for (Windows, macOS, Linux) that is written in the programming language Java. It is used to write and upload programs to Arduino board. 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. 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 [24]. Figure 3.12 shows the Arduino software IDE.

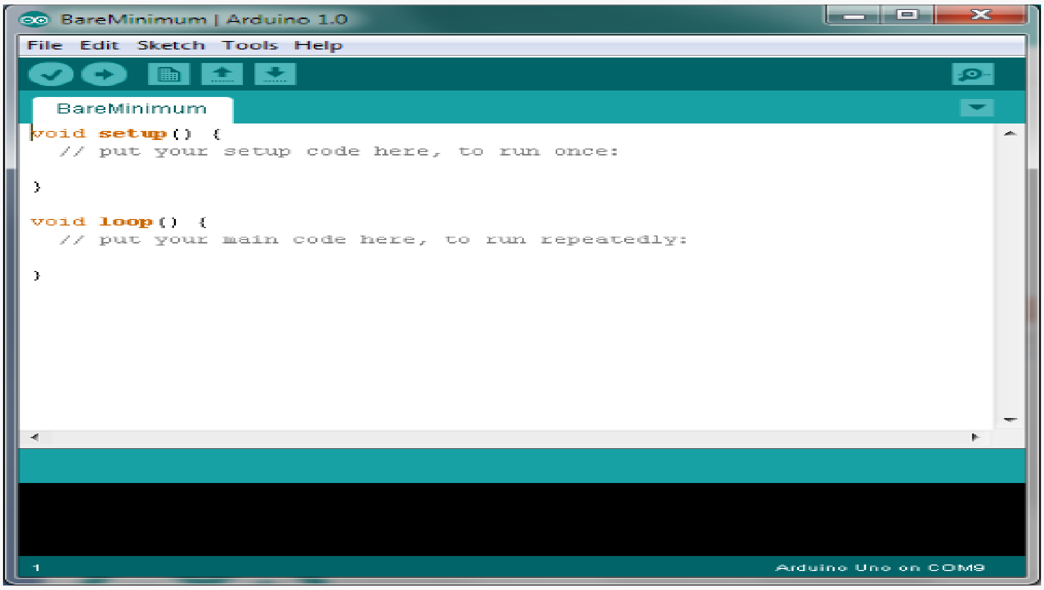


Figure 3.12: Arduino software IDE

3.7.4 Proteus design suite

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. It was developed in Yorkshire, England by Labcenter Electronics Ltd and is available in English, French, Spanish and Chinese languages [23]. Figure 3.13 shows the Proteus design suit window.

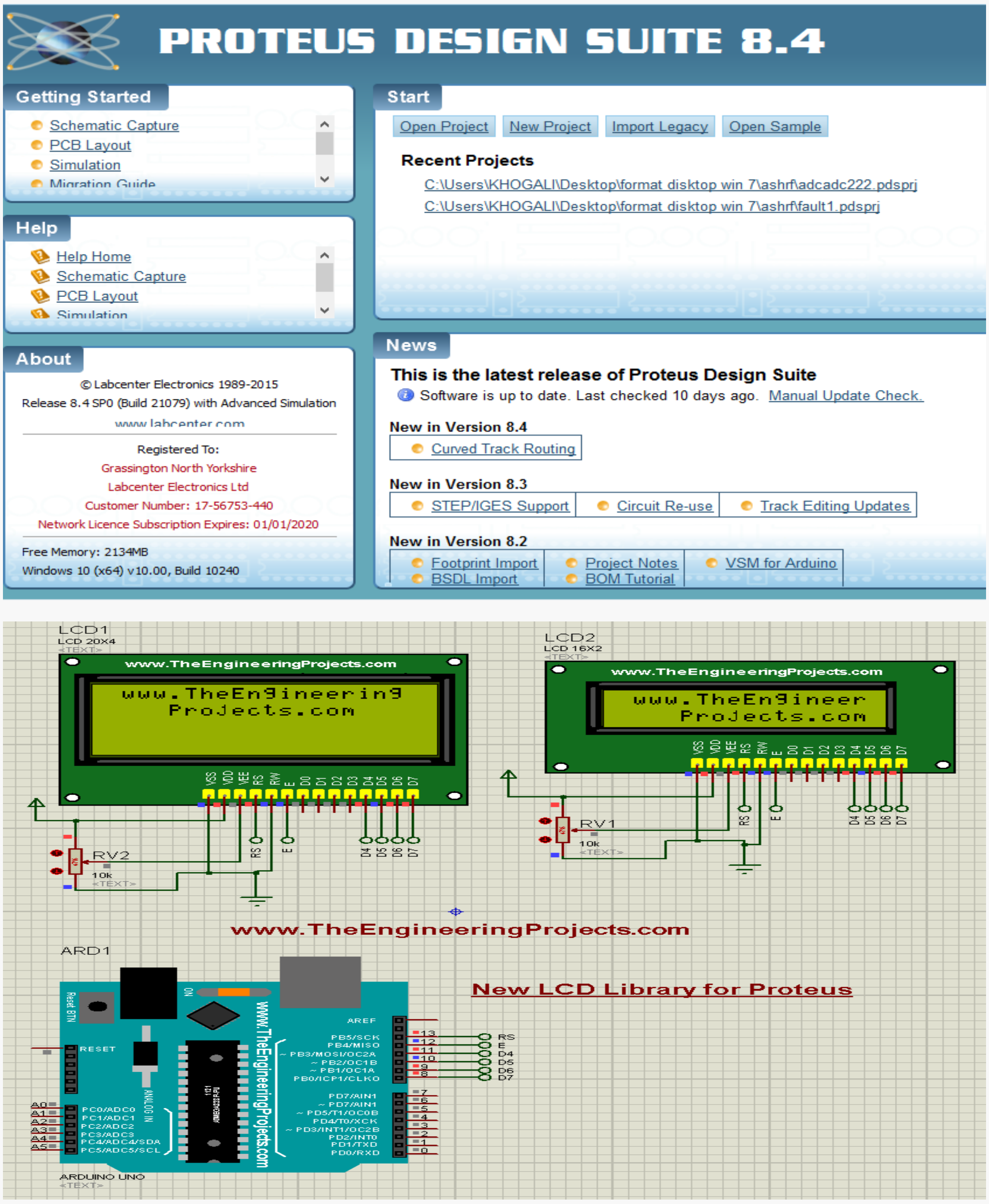


Figure 3.13: Proteus design suit window

3.7.5 MATLAB program

Cleve Moler, the chairman of the computer science department at the University of New Mexico, started developing MATLAB in the late 1970s. It soon spread to other universities and found a strong audience within the applied mathematics community [25]. Jack Little, an engineer, was exposed to it during a visit Moler made to Stanford University in 1983. Recognizing its commercial potential, he joined with Moler and Steve Bangert. They rewrote MATLAB in C and founded Math Works in 1984 to continue its development. These rewritten libraries were known as JACKPAC. In 2000, MATLAB was rewritten to use a newer set of libraries for matrix manipulation. The MATLAB application is built around the MATLAB scripting language. Common usage of the MATLAB application involves using the Command Window as an interactive mathematical shell or executing text files containing MATLAB code [25]. Figure 3.14 shows the main window of the MATLAB software.

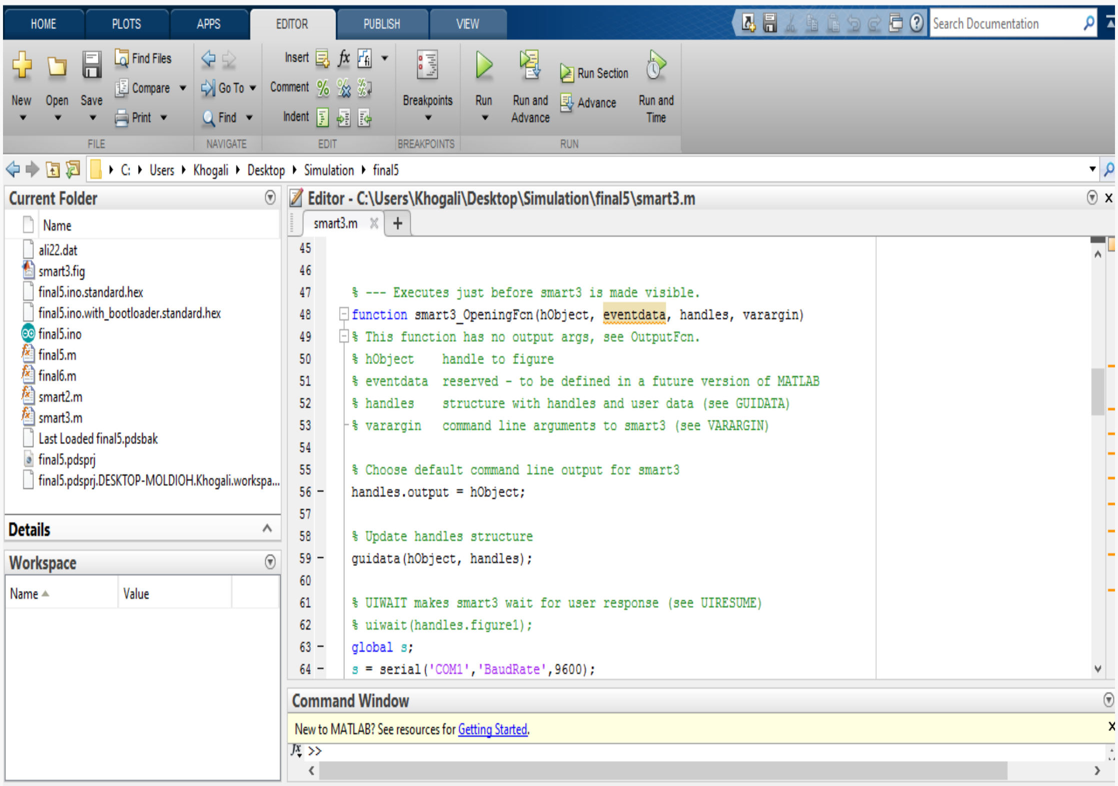


Figure 3.14: MATLAB program main window

3.7.6 Graphics user interface programming

The Graphical User Interface (GUI), is a type of user interface that allows users to interact with electronic devices through graphical icons and visual indicators such as secondary notation, instead of text-based user interfaces, typed command labels or text navigation. The actions in a GUI are usually performed through direct manipulation of the graphical elements. GUIs are used in many handheld mobile devices such as MP3 players, portable media players, gaming devices, smart phones and smaller household, office and industrial controls [26].

Designing the visual composition and temporal behavior of a GUI is an important part of software application programming in the area of human–computer interaction. A GUI may be designed for the requirements of a vertical market as application-specific graphical user interfaces. Examples include Automated Teller Machines (ATM), Point Of Sale (POS), touch screens at restaurants, etc… [26]. Figure 3.16 shows the general basic window of the graphical user interface.

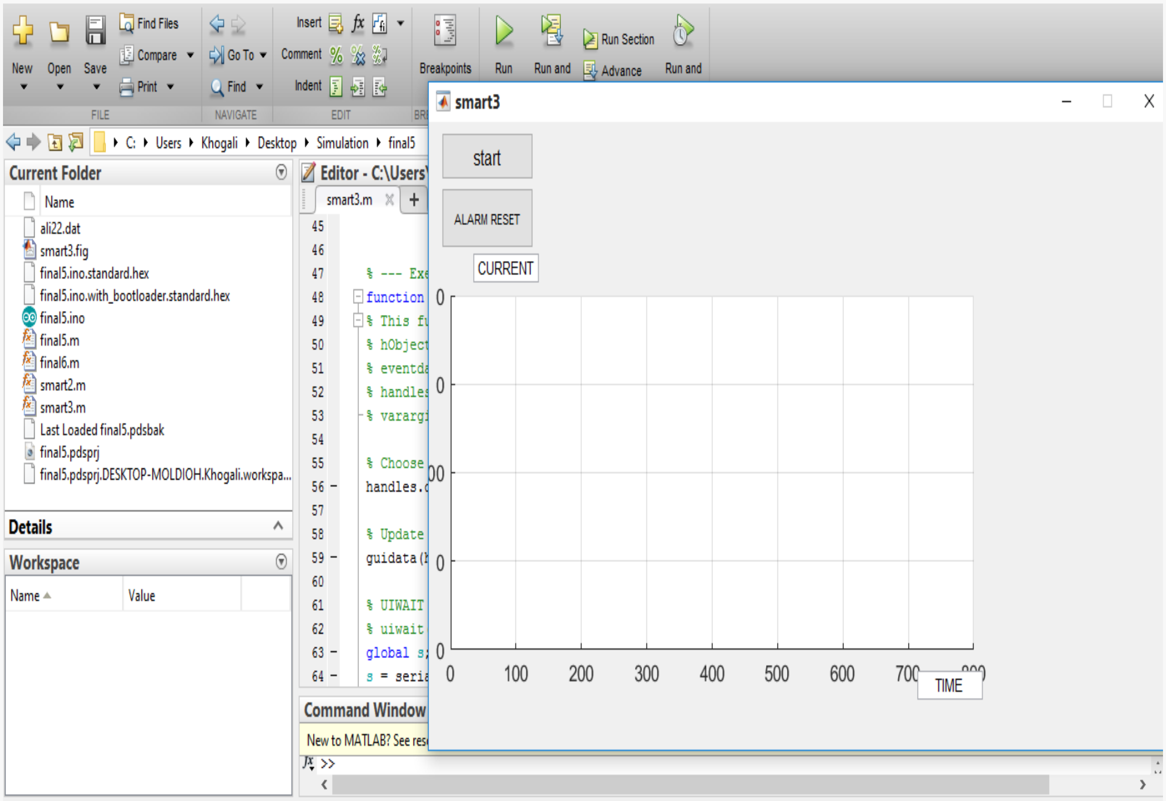


Figure 3.15: Graphical user interface

CHAPTER FOUR

RESULTS AND DISCUSSION

4.1 Introduction

The following chapter represent the results and discussion of the fault recorder simulation. This simulation was made using MATLAB, PROTEUS, VISUAL SERIEAL PORTS and ARDUINO programs. The result was taken from MATLAB by using Graphical User Interface (GUI) which draw the incoming signal from the microcontroller in sinewave signal.

Also, the microcontroller which is based in Arduion chip programed in C code by Arduino's program,

It is clear from the overall design of digital fault recorder that the signal was transmit from the microcontroller to the serial RS232 in PROTEUS program then it must be transfer to read it in the MATLAB. The SERIAL PORT program makes it so ease to transfer the output signals from Arduino to MATLAB and it make the two programs are connected which each other.

4.2 Results Scenarios

Fault recorder has big capability to analysis the signal which it is recorded. It has been taken into account that the cases study here are the most important faults in medium voltage transmission lines

4.2.1 The normal case

This case means the situation of the load current value which is in the range of the main relay setting. Each main relay of distribution power lines has certain setting of current values. Most of medium voltage (11KV to 33KV) power lines designed in limit of 400A maximum. And that for many reasons like conductor size, junction points, cables specifications … etc.

Figure 4.1 shows the input signal to the microcontroller, the yellow one represents the main supply or the main load current which is modulated by Current Transformers (CT’s) multiply by the (CT) ratio. The second signal in blue color explain the output sending from the Arduino chip to the serial port, and it’s clear that it is a digital signal. Last signal which in red color obtain the result of neutral current if it’s increased or reached of certain value. This value setting in the main relay according to the breaker, isolator and busbar specifications.

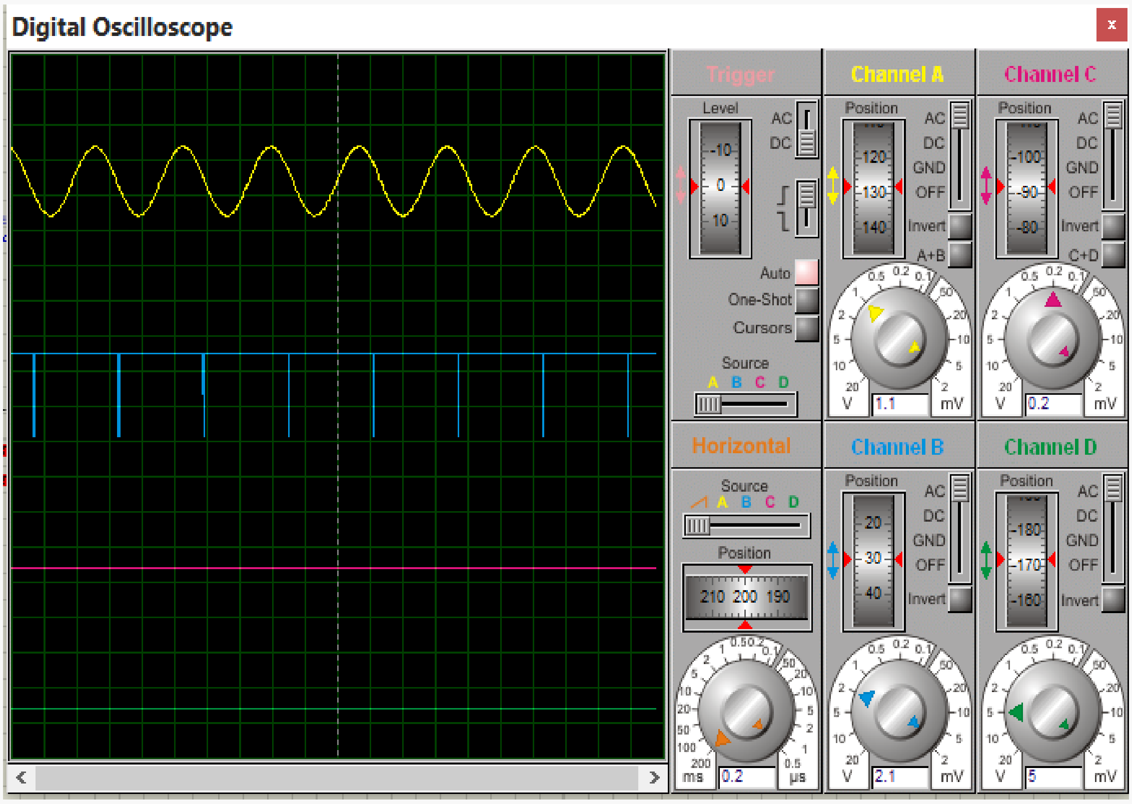


Figure 4.1: input signals in normal case

In Figure 4.2 the Y axis represent the magnitude of the output current value which is reading in MATLAB, and the X axis is for the time. The signal view only in the positive part of the sine wave signal because it converted from the digital signal. The output signal converted by the analog to digital converter before it delivered into microcontroller, for that the micro replace all the negative values by zeros. Table 4.1 shows the record data file in MATLAB.

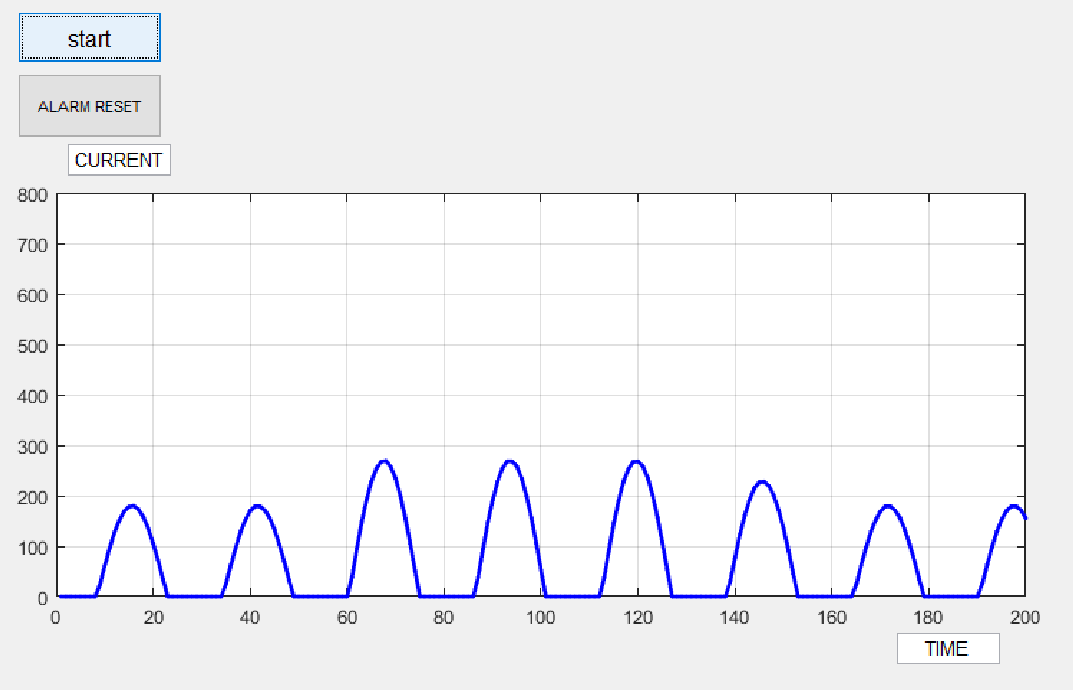


Figure 4.2: Output current signal vs. the time

Table 4.1 shows the record data file in MATLAB in the normal case

Serial	Time (second)	Current (A)
1	0	0
2	8	26
3	10	63
4	12	120
5	14	153
6	16	179
7	18	155
8	20	93
9	22	73
10	24	0
11	26	0
12	28	0
13	30	0

4.2.2 The Earth fault case

This case happened when one or more phases touch any peace of the ground. That makes the fault currant come through the resistance which is fixed in the ground near the substation. The CT detected the earth fault current when the current in the neutral (In) is bigger than the setting value. In most cases of earth fault in medium voltage power lines, neutral current could reach about triple time of the rated current. The change of the input signal is very clear in the digital oscilloscope in Figure 4.3, it viewed in the red line which increase in large values in a few time and return to zero when the breaker shut down or to normal values after it starting again.

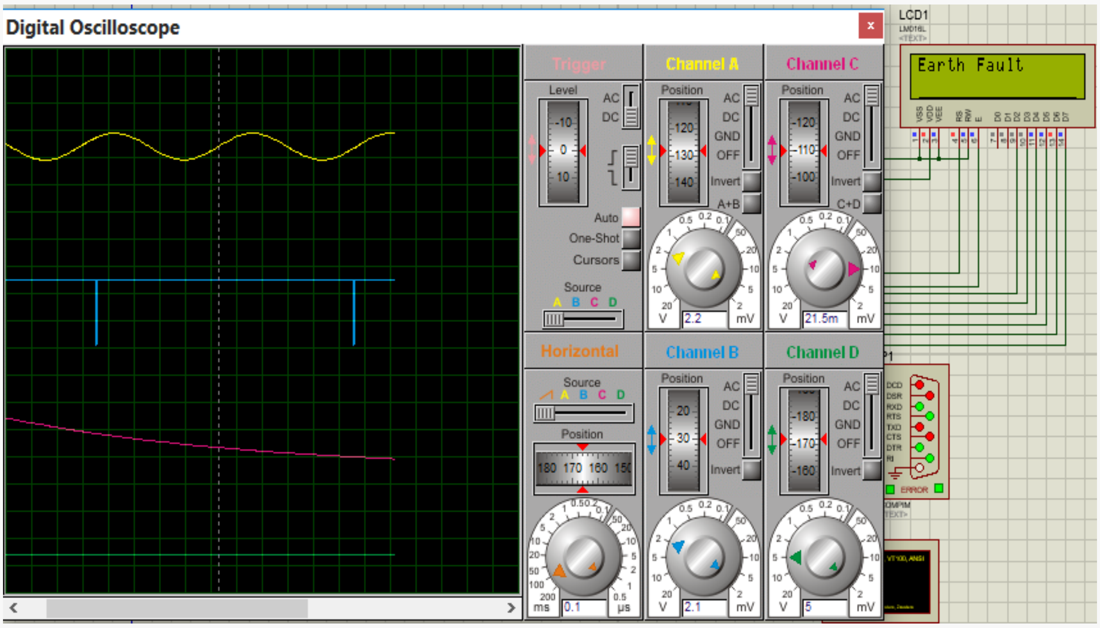


Figure 4.3: Input signal change in Earth fault case

The positive part of the signal which is shown in the Figure 4.4 represents the magnitude of the output current value. In this case the recorder takes the neutral current value because it is bigger than the rated one. And that because of the earth resister which the earth current come throw it is too small (Often between 6Ω to 12Ω). Table 4.2 shows the record data file in MATLAB in the earth fault case.

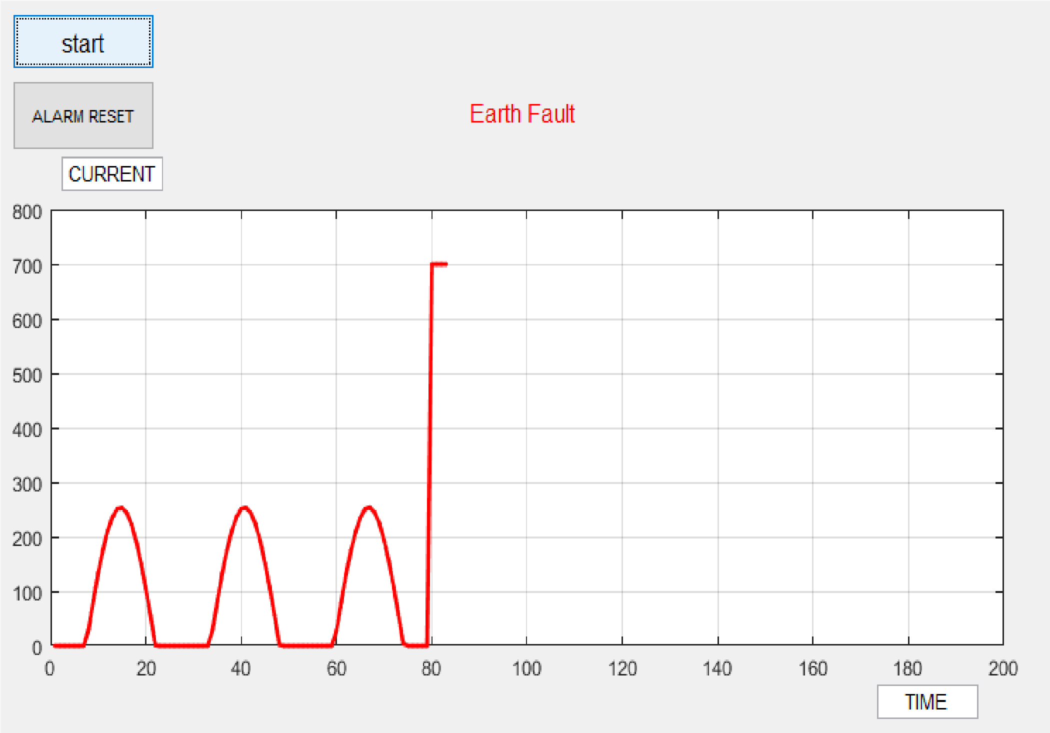


Figure 4.4: Output signal registered in Earth fault case

Table 4.2 shows the record data file in MATLAB in the earth fault case

Serial	Time (second)	Current (A)
1	0	0
2	10	125
3	20	75
4	30	0
5	40	265
6	50	25
7	60	0
8	70	225
9	80	700
10	90	700
11	100	700
12	110	700

4.2.3 The over current case:

This case can be separated in two types according to load current percentage. The first type (called critical over current) happens when the load current is greater than 85% of the rated current value and there is no any current signed in neutral phase. That is similar like the normal case but in high magnitude. It is very helpful when this type happened and the systems have a warning alarm to operators to make a necessaries process to avoided the loaded troubles. The Figure 4.5 shows the load current when it is reach the critical of the rated setting value.

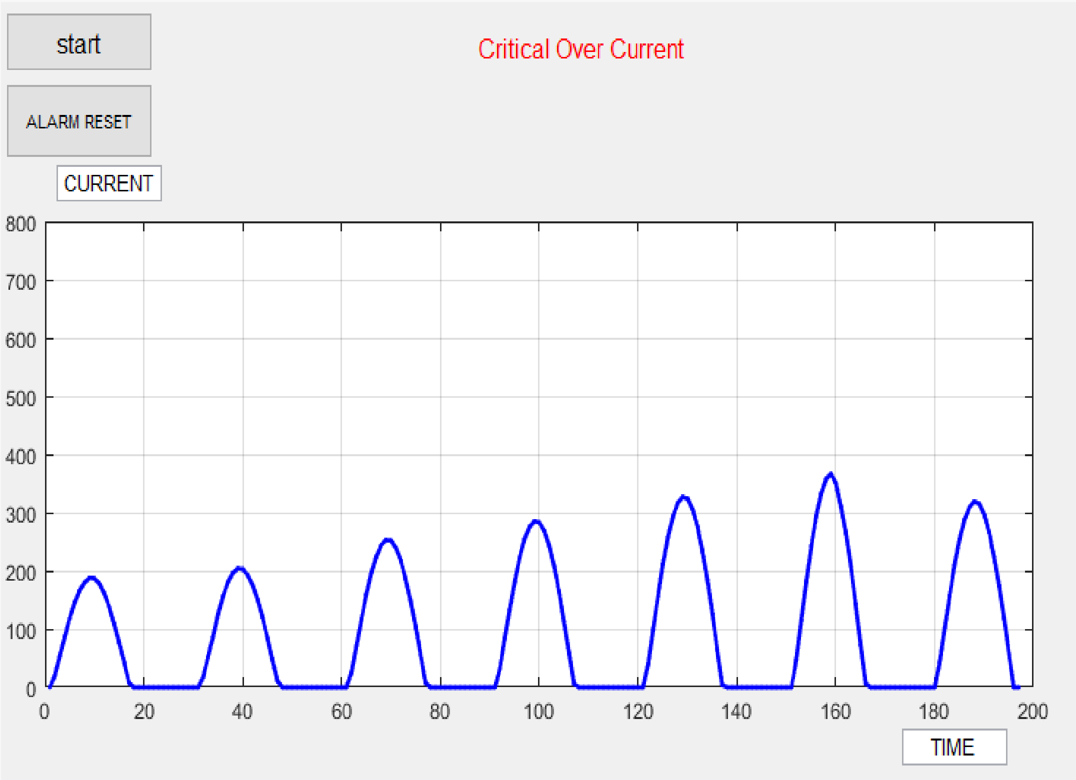


Figure 4.5: Critical value of load current

If the load continues increase the current value and pass the critical value without doing any treatment processes, the load current might reach to the maximum value and then the relay sends a signal to the tripping coil to turn the breaker off. This second type called Max Over current and it's shown in the Figure 4.6. From both types of over current case, the input signal does not different than the normal case, except the load current got a large value which may be defected in the power lines equipment.

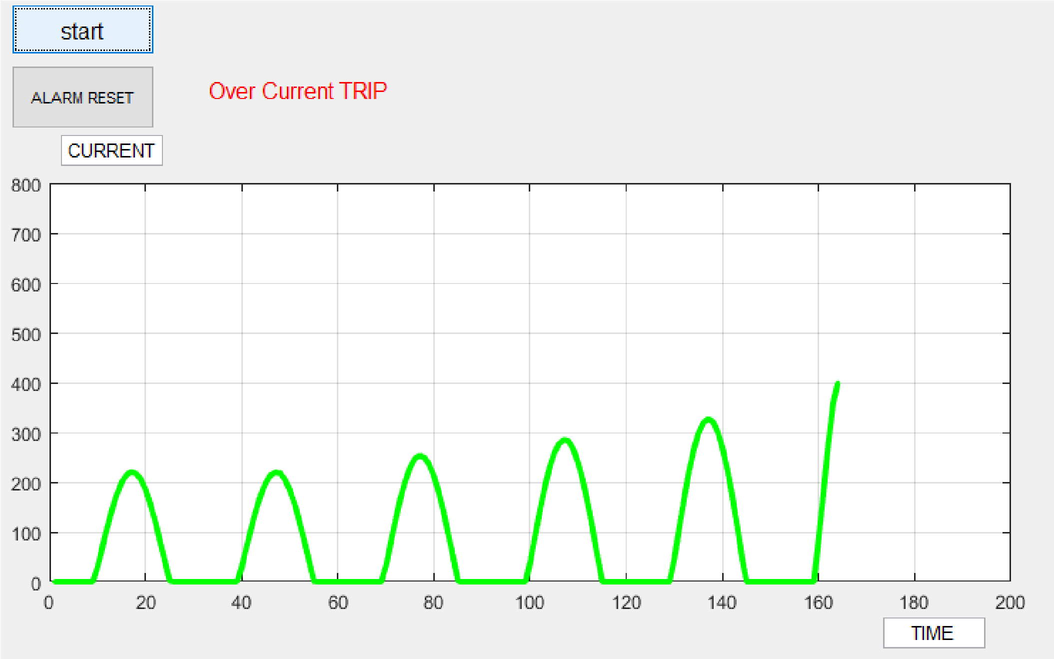


Figure 4.6: maximum over current

Figure 4.7 shows approximately the load current change in the input signal in the two types of over current case.

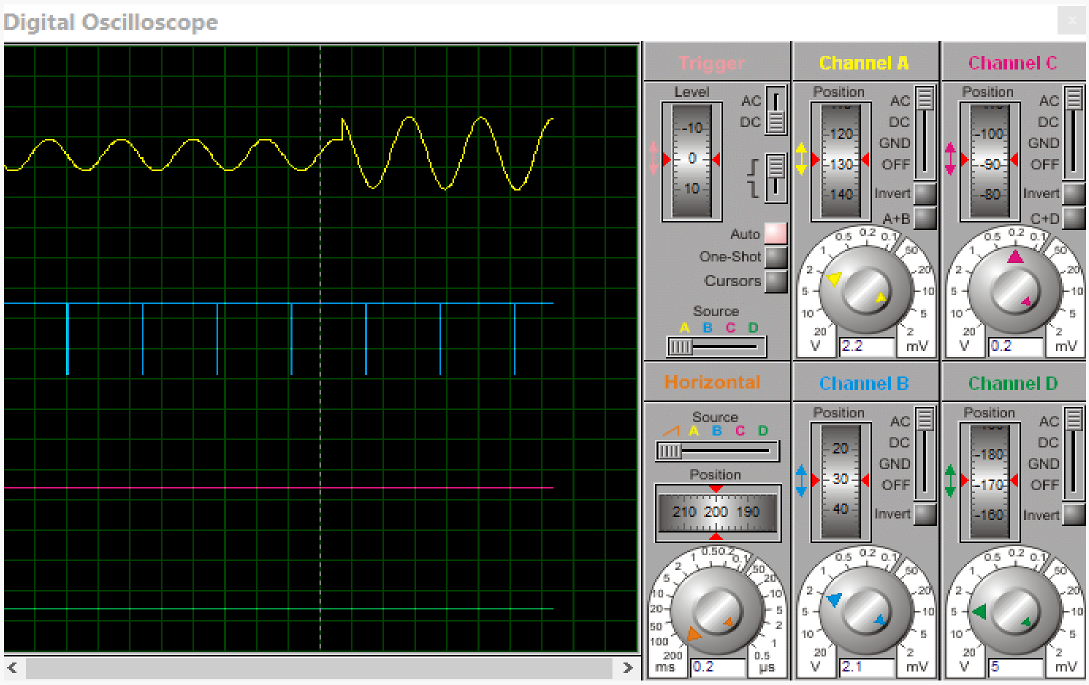


Figure 4.7: maximum over current input signal

CHAPTER FIVE

CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

The digital fault recorder design is providing permanent archives of data to give support at any time to operation staff in having the desire information on the plant status very quickly. This design makes sure all the data from the transmission lines had been captured into a software design MATLAB, record all the data from the microcontroller, displaying all this data and monitor the variation of the current signal amplitude. But it didn't reach the overall goal of this thesis because of recording only the absolute value instead of RMS value.

Conventional faults detectors response is slow in nature and even they might be not able to detect the faults due to the low level of fault current. However, designed network could perform well even in the presence of considerable amount of fault resistance. From this point digital fault recorders are more accurate, reliable and confidence than the conventional systems. Also, it is minimizing the energy consumption of distributed sensors which are participating in fault management. It is a good point that companies which interested in the design of the protection devices or any equivalents one went to developing of the digital fault recorders and even competition in this area because of their conviction of the importance of these devices.

The analysis of oscillograms has proven to be invaluable in improving the reliability of the power system. At the same time, it has resulted in great economic savings for utilities. When the proper people and recording equipment are used for system event investigations, the positive results can be immeasurable.

5.2 Recommendation

1) The future work has been taking from the missing issues in this thesis. These missing works represent in recording the RMS and the voltage values.
2) There are other faults never take chances to be consider in these devices, because of the RMS value are not recording like the deferential fault, over voltage, under voltage, restricted earth faults, and etc...
3) It’s better to use using high performance software technique interfacing because using MATLAB causes some delay time for reaching the data in same time when sending it.
4) It will be more helpful to determine the location of a fault on transmission line between a master station and a remote station by using this model if it has been developed.
5) Interfacing between Fault recorders and another control system make it is more easy and helpful for the operators to make decisions.
6) Any decisions could made very dangerous effects if it’s make without analysis and studies. The fault recorders help operators to analysis faster and reliable in same time.

REFERENCES

[1] Joy Perez, Ed, "A guide to Digital Fault Recording Event Analysis". Wining MB, R3Y1G4 Canada: ERL Phase Power Technology, 2010.

[2] Wenling Zhang , Yanjie Yang , "The Design of Digital Fault Recorder Module Based on S12XD". Monday, November 5, ending 5 pm ET on November 6, 2011.

[3] F. Dawalibi, George B. Niles, "Measurements and Computations of Fault Current Distribution on Overhead Transmission Lines", IEEE Transactions on Power Apparatus and Systems,Volume:PAS-103 , Issue3 , March 1984 .

[4] JR.LCA., Ed., EE 423- "power system analysis: Faults". JOURNAL NAME: Smart Grid and Renewable Energy, Vol.4 No.9, December 31, 2013.

[5] Steve Mackay, "Electrical Power System Protection -Faults", August 2006-June 2008.

[6] CHarles A. Gross, "Different Types of Faults in Power System", AIEE Trans. PAS-79.323, 1960. New York, 1983.

[7] California Environmental Quality Act (CEQA), (Association for Protection of Environmental Values in Ukiah v. City of Ukiah. Retrieved November 13, 2014.

[8] Curtis Michael, ED, "System Protection and Control Sustain Program Asset Management Strategy". March 2012.

[9] Rasheek Rifaat, "Power System Protective Relays: Principles and Practices", IEEE Southern Alberta Section PES/IAS Joint Chapter Technical Seminar, November 2016.

[10] Hug Davila, ED., “Software Tools for Record Faults Analysis in Power System”, June 2011.

[11] Institute of Electrical and Electronic Engineers. 1989. Summary of the special publication: application of fault and Disturbance recording devices for protection system analysis. IEEE Transactions on Power Delivery 4(3):1625-1630.

[12] Y. Li, D. C. Liu, “Study on the Electrical Power Fault Recorder Integrated Analysis & Application System”, X. W. Du and Q. F. Liao, November 2010.

[13] "Use of Oscillograph Records to Analyze System Performance," John R. Boyle, Section 9.6 of The Electric Power Engineering Handbook, L. L. Grigsby, Editor, CRC Press in Cooperation with IEEE Press, 1998.

[14] Mladen Kezunovic, Igor Rikalo. “Advanced System Solutions for Automated Fault Analysis Using DFR Files”,Test Laboratories International Inc ,and Texas A&M University, Power Research Institute under EPRI Project No. RP3192-01.

[15] Miguel Moreto and Jacqueline G. Rolim, “Automated Analysis of Digital Fault Recorder Data in Power Generating Plants”, Federal University of Santa Catarina Florianopolis, Brazil, August 2010 with 96 Reads DOI: 10.1590/S0103-17592010000400003.

[16] Alan G. Smith, “Introduction to Arduino _A piece of cake! _”, ISBN: 1463698348_ Colossians 3:23, September 30, 2011.

[17] ATMEGA 32 datasheet, ATMEL Corporation, 2503N–AVR–06/08, 2008 Atmel Corporation.

[18] Max232 datasheet, 19-4323; Rev 7b; 11/97, 1997 Maxim Integrated Products.

[19] D Cai and HP Zhang, “Hands-on Research in Complex Systems”, Advanced Study Institute ---17-29 June 2012, Shanghai Jiao Tong University (Shanghai, People’s Republic of China).

[20]. Hongqian Lu, Chunmei Duan,"The application of Proteus in teaching of microcomputer principles" 3rd International Conference on Science and Social Research (ICSSR 2014), IEEE Explore White Paper, June 2011.

[21] Castro, Jorge R. “Building a Home Security System with Arduino”, Design, build, and maintain a home security system with Arduino Uno. Birmingham, UK. p. 15. ISBN 9781785283802. OCLC 922588951.

[22] Yair Altman, "MATLAB Documentation", Western Technology Services (WTS), Automated Trader magazine (2011, 2012), Math Works. Retrieved 14 August2013.

[23] Smith, S. T. (2006). MATLAB: “Advanced GUI Development”, Dog Ear Publishing. ISBN 978-1-59858-181-2.

[24] Roshan Chhetri, Tshewang Lhendup,” Voltage Profile in Distribution System”, Department of Electrical Engineering, College of Science and Technology Royal University of Bhutan, Phuentsholing Bhutan.

APPENDIX A

ARDUINO CODE

```
#include <SoftwareSerial.h>

#include <LiquidCrystal.h>

// initialize the library with the numbers of the interface pins

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

void setup() {

  Serial.begin(9600);

   // set up the LCD's number of columns and rows:

  //lcd.begin(16, 2);

  //lcd.print("Fault recorder");

  delay(500);}

void loop()

{Serial.begin(9600);

delay(50);

 int adc = analogRead(A0);

 int adc2= analogRead(A1);

 delay(50);

if (adc2>=40)

  { delay(100);
```

```
  lcd.clear();
   // Print a message to the LCD.
  lcd.setCursor(0,1);
  lcd.print("Earth Fault");
  delay(200);
  Serial.println(700);
  delay(1000);
  Serial.println(700);
  delay(1000);
  Serial.println(700);
  delay(500);  }
  if (adc2<=40)
{lcd.clear();
lcd.setCursor(0,1);
lcd.print(adc);
 if ((adc>=350)&&(adc<400))
 { // Print a message to the LCD.
lcd.clear();
lcd.setCursor(0,1);
lcd.print("Critical value");
delay(100);
```

```
// Print a value to the LCD.
lcd.clear();
lcd.setCursor(0,1);
lcd.print(adc);
 Serial.println(adc);
 delay(80);
 lcd.print(adc);
  delay(80);}
if(adc<=349)
 { delay(10);
  Serial.println(adc);
  lcd.clear();
 // Print a message to the LCD.
  lcd.setCursor(0,1);
  lcd.print("TheValue= ");
  delay(100);
 lcd.print(adc);
  delay(150);}
if((adc>=400)&&(adc<=600))
{lcd.clear();
 // Print a message to the LCD.
```

```
  lcd.setCursor(0,1);
  lcd.print("Over Current");
  delay(200);
Serial.println(400);
delay(2000);
Serial.println(400);
delay(2000);
Serial.println(400);
delay(2000);}}
delay (100);}
```

APPENDIX B

MATLAB CODE

```
function smart3_OpeningFcn(hObject, eventdata, handles, varargin)

% This function has no output args, see OutputFcn.

% hObject    handle to figure

% eventdata  reserved - to be defined in a future version of MATLAB

% handles    structure with handles and user data (see GUIDATA)

% varargin   command line arguments to smart3 (see VARARGIN)

% Choose default command line output for smart3

handles.output = hObject;

 % Update handles structure

guidata(hObject, handles);

 % UIWAIT makes smart3 wait for user response (see UIRESUME)

% uiwait(handles.figure1);

global s;

s = serial('COM1','BaudRate',9600);

fopen(s);

% --- Outputs from this function are returned to the command line.

function varargout = smart3_OutputFcn(hObject, ~, handles)

% varargout  cell array for returning output args (see VARARGOUT);

% hObject    handle to figure
```

```
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
 % Get default command line output from handles structure
varargout{1} = handles.output;
 % --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global s;
i = 0;
while (i<200)
    i = i+1;
    adc(i)=fscanf(s,'%d');
    save ali22.dat adc
    drawnow;
    if (adc(i)<=349)
     set(handles.reset,'string',' ')
     set(handles.octrip,'string',' ')
     pause(0.2);
     set(handles.EF,'string',' ')
```

```
 pause(0.2);

 %set(handles.ALARM,'string','')

 %pause(0.3);

axes(handles.axes1);

plot(adc,'b.-','lineWidth',2);

grid on;

axis([0 200 0 800]);

pause(0.3);

end

if ((adc(i)>=350)&&(adc(i)<398))

 set(handles.reset,'string',' ')

 set(handles.octrip,'string',' ')

 set(handles.EF,'string',' ')

 set(handles.ALARM,'string','Critical Over Current')

 pause(0.3);

 axes(handles.axes1);

plot(adc,'b.-','lineWidth',2);

grid on;

axis([0 200 0 800]);

pause(0.5);

end
```

```
  if ((adc(i)>=400)&&(adc(i)<=600))
   set(handles.EF,'string',' ')
   set(handles.reset,'string',' ')
   set(handles.ALARM,'string',' ')
   set(handles.octrip,'string','Over Current TRIP')
   pause(0.3);
   axes(handles.axes1);
  plot(adc,'g.-','lineWidth',3);
  grid on;
  axis([0 200 0 800]);
pause(2);
  end
  if (adc(i)>=650)
     pause(0.5);
     set(handles.ALARM,'string',' ')
     set(handles.octrip,'string',' ')
     set(handles.EF,'string','Earth Fault')
     axes(handles.axes1);
  plot(adc,'r.-','lineWidth',2);
  grid on;
  axis([0 200 0 800]);
```

```
  pause(0.3);

  end

  pause(0.3);

  end

fclose(s);

    Delete(instrfindall);

      % --- Executes on button press in pushbutton2.

    function pushbutton2_Callback(hObject, eventdata, handles)

    % hObject    handle to pushbutton2 (see GCBO)

    % eventdata  reserved - to be defined in a future version of MATLAB

    % handles    structure with handles and user data (see GUIDATA)

    global s;

          set(handles.ALARM,'string',' ')

         set(handles.octrip,'string',' ')

         set(handles.EF,'string',' ')

        set(handles.reset,'string','system reset')

        pause(0.8);

        set(handles.reset,'string',' ')

    pause(0.2);

    Delete(instrfindall);

    fclose(s);
```