بسم الله الرحمن الرحيم

Sudan University of Science & Technology

College of Graduate Studies

Design of Control System for Identification of Phase Failure and Detection of Unbalanced Phase Sequence of Induction Motor

تصميم نظام تحكم لتمييز الفشل في مرحلة إكتشاف عدم إتزان تركيب المرحلة للمحرك الحثي

A thesis Submitted for Partial Fulfilment for the Requirement of M.Sc. Degree in Mechatronics Engineering

Presented by:-

Hussam Aldeen Elyass ALamain

Supervisor:-

Dr.Ala Eldin Awouda

November 2018

الآيـة

بسم الله الرحمن الرحيم

قال تعالى:

﴿اقْرَأْ بِاسْمِ رَبِّكَ الَّذِي خَلَقَ ۞ خَلَقَ الْإِنْسَانَ مِنْ عَلَقٍ ۞ اقْرَأْ وَرَبُّكَ
الْأَكْرَمُ ۞ الَّذِي عَلَّمَ بِالْقَلَمِ ۞ عَلَّمَ الْإِنْسَانَ مَا لَمْ يَعْلَمْ ۞ ﴾

صدق الله العظيم

سورة العلق الآية(1-5)

Dedication

I dedicate this thesis to my parents and my family they are my life save me in good environment and support me to study.

I also dedicate this thesis to my teachers, friends, and colleagues who have supported me throughout the process; I will always appreciate all they have done.

Acknowledgement

First and foremost, praise be to ALLAH who gave me the power to complete this project.
Then I would like to express my heartily gratitude to my supervisor, Dr: Alaa Eldin Awouda, for the guidance and support given throughout the progress of this project.

My appreciation also goes to my family who has been so tolerant and supported me all these years. Thanks for their encouragement, love and emotional support that they had given to me.

I also would like to thank Major Eng. Asaad Musaab Ali Yousif for his unlimited support whenever I need, also great thanks to Eng. Muaaz Abdulrahem Ibrahim Ali for his valuable help during this project.

Abstract

The main purpose of this project is to design of protection for induction motors breakdown observation based on arduino microcontroller. This project proposes an approach to Good motor protection; this has to cover all possible problem areas. It must not be tripped before the motor is put at risk, if the motor is put at risk, the protection device has to operate before any damage occurs, If damage cannot be prevented, the protection device has to operate quickly in order to restrict the extent of the damage as much as possible. Protection and controls the motor from the Phase sequence, voltage unbalanced considered the major functions of most control systems. The experiment showed that the system for protection and controlling induction motor can be used stably and reliably.

المستخلص

الهدف الاساسي من هذا المشروع هو تصميم نظام حماية الموتور الحثي و مراقبة الانهيار باستخدام المتحكم أردوينو. هذا المشروع يعالج طرق الحماية الجيدة للموتور الحثي و يشمل كافة المشاكل الممكنة التي يجب العثور عليها قبل وضع الموتور تحت الخطر , اذا تعرض الموتور للخطر يجب ان يعمل جهاز الحماية قبل حدوث اي تلف , اذا لم يكن بالإمكان منع الضرر, يجب ان يعمل جهاز الحماية علي الحد من مدي الضرر قدر الإمكان.

الحماية و السيطرة علي المحرك من تسلسل الطور و عدم اتزان الجهد تعتبر من المهام الاساسية لمعظم انظمة الحماية. أظهرت التجربة انه يمكن استخدام نظام الحماية و التحكم في المحرك الحثي بثبات و موثوقية.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLE

LIST OFABBREVIATIONS

Abbreviation	Description
AC	Alternating Current
ADC	Analog to Digital Convertor
AREF	Analog Reference (voltage)
CT	Current Transformers
DC	Direct current
DTR	Data Terminal Ready
Emf	Electromagnetic Field
FLC	Ferroelectric Liquid Crystal
FTDI	Future Technology Devices International
GMMs	Gaussian Mixture Models
IDE	Integrated Development Environment
I/O	Input and Output
IM	Induction Motor
I-T	Inverse Time
LED	Light Emitting Diode
LCD	Liquid Crystal Display
PC	Personal Computer
PWM	Pulse width Modulation

RMF	Rotating Magnetic Field
RPS	Reconstructed Phase Spaces
RXD	Receive Data
TFT	Thin Film Transistor
TN	Twisted Nematic
TXD	Transformer Data
UART	A Universal Asynchronous Receiver Transmitter
UIP	User Interface Panel
USB	Universal Serial Bus

CHAPTER ONE

INTRODUCTION

CHAPTER ONE

Introduction

1.1 Overview

Induction motors are complex electro-mechanical devices utilized in most industrial applications for the conversion of power from electrical to mechanical form. Induction motors are used worldwide as the workhorse in industrial applications. Such motors are robust machines used not only for general purposes, but also in hazardous locations and severe environments. General purpose applications of induction motors include pumps, conveyors, machine tools, centrifugal machines, presses, elevators, and packaging equipment. On the other hand, applications in hazardous locations include petrochemical and natural gas plants, while severe environment applications for induction motors include grain elevators, shredders, and equipment for coal plants. Additionally, induction motors are highly reliable, require low maintenance, and have relatively high efficiency. Moreover, the wide range of power of induction motors, which is from hundreds of watts to megawatts, satisfies the production needs of most industrial processes.

However, induction motors are susceptible to many types of fault in industrial applications. A motor failure that is not identified in an initial stage may become catastrophic and the induction motor may suffer severe damage. Thus, undetected motor faults may cascade into motor failure, which in turn may cause production shutdowns. Such shutdowns are costly in terms of lost production time, maintenance costs, and wasted raw materials.

1.2 Problem Statement

For motor protection and control there are three main problems are raised which are Phase failure, unbalanced voltage and phase sequence. Phase failure occurs when power is lost from one of the power supplying. The motor will continue to operate but will draw an excessive amount of current. In this condition, the overload relay should cause the motor starter to disconnect the motor from the power line if the overload heaters have been sized correctly. Single phasing will cause the two phases that remain energized in a three-phase motor to increase current by an average of 173%. The seconded problem is unbalanced voltage:

When voltage unbalance exceeds more than 2 percent in three-phase systems can cause current unbalance among the windings. These, in turn can cause this an increase in winding temperature and an overheating problem that can be harmful to the motor. Lastly the Phase sequence:

When two of powers supplying lines are reversed, this will cause the motor to reverse the direction of rotation. Which will lead to serious problem with some types of equipment. Unintended reversal of direction can cause gear teeth to shear, chains to break, and the impeller of submersible pumps to unscrew off the end of the motor shaft. It can not only cause damage to equipment but also injury to operators or personnel in the vicinity of the machine.

1.3 Proposed Solutions

Design of control circuit for detection and identification phase sequence between three phases of induction motor and protection induction motor from over voltage and under voltage using microcontroller arduino and contactor, arduino control of the contactor to on or off. User interface panel and LCD are used to monitoring status of three phase power supply and showing result.

1.4 Research Aims and Objectives

- Design and implementation of induction motors breakdown observation based on arduino microcontroller.
- Protection and controls of induction motor from the Phase failure, unbalanced voltage and phase sequences.
- Simulation of proposed control circuit.
- Performance evaluation.

1.4 Methodology

The main objective of this system is to protect the motor from phase failure, phase unbalance and phase insequence which can be achieved using microcontroller arduino. All the manual operations are replaced by sending signals from the microcontroller to the main contractor. The working of the design can be classified into four phases:

[A]phase sequence identification

The system start converting the three phase input power supply into pluses which can be easy to identify if the phase in sequence or not.

[B]phase failure and phase unbalance identification

Through ADC pins of microcontroller the system also can read the value of voltage for the three line to identify the under voltage, over voltage and unbalance voltage between the three lines and also to detect if one line or more is failure.

[C]User Interface Panel

Visual basic is used to design the user interface panel for observation of motor behavior which will offer you the voltage of the three lines (L1, L2&L3), indication of the phase sequence, indication of under-over voltage, indication of unbalance voltage and phase failure indication continuously.

[D] Microcontroller processing

Microcontroller is used as main processing unit to read data from motor and starts processing, update information and send it to visual basic through RS232 or LCD directory from microcontroller.

1.6 Thesis layout

Thesis is summarized in five chapters:

Chapter 1: Introduction highlight problem statement proposed solution and objectives.

Chapter 2: Literature Review gives solid data about system parts and previous work.

Chapter 3: Methodology explains system design.

Chapter 4: Result and discussion highlight result from system simulation.

Chapter 5: Conclusion and Recommendation for future research.

CHAPTER TWO

LITERATURE REVIEW

CHAPTER TWO

Literature Review

2.1 Overview

This chapter is protection of induction motor (IM) previous case studies, these studies which have been done previously by other researchers. It is very essential to refer to the variety of sources in order to gain more knowledge and skills to complete this project. These sources include reference books, thesis, and papers.

2.2 Previous studies

PalashKundu and Arabinda Das proposed a prototype detection system based on 8085 microprocessor 8751 microcontroller has been developed for identification of phase sequence and detection of phase unbalance. Proposed microprocessor based detection system can generate unique codes for different probable permutations for supply system. System is capable to identify phase sequence of three phase supply system and can identify phase sequence during fluctuation of power frequency within recommended value. System can detect phase sequence within a cycle and can identify and protect system form any unbalance condition instantaneously [1].

Aderiano M. da Silva, B.S. proposed induction motor fault diagnosis is based on the analysis of the envelope of the three phase stator current. This diagnostic method can classify two types of induction motor faults: broken rotor bars and inter-turn short circuits in the stator windings. Experimental results show that the three phase current envelope is a powerful feature for motor fault classification. The envelope signal is extracted from the experimentally acquired stator current signals and is used in conjunction with machine learning techniques based on Gaussian Mixture Models (GMMs) and Reconstructed Phase Spaces (RPSs) to identify motor faults. The second method presented in this thesis is an induction motor fault monitoring technique based on the air gap torque profile analysis, associated with machine learning techniques to classify the operating condition of an induction motor as healthy or faulty. These machine learning techniques are based on GMMs and RPSs [2].

A.Selvanayakam, W.RajanBabu, S.K.Rajarathna Fault Detection in Three Phase Induction Motor. Proposed Induction motor faults can be detected in an initial stage in order to prevent the complete failure of an induction motor and unexpected production costs. Accordingly, this project presents three methods to detect induction motor faults. The first method is a motor fault diagnostic method that identifies the faults like inter-turn short circuits in stator windings, broken rotor bars that helps in identifying the motor fault's severity and air gap eccentricity. These types of faults represent 40 to 50% of all reported faults. In motor fault monitoring method that classifies the operating condition of an induction motor as healthy or faulty [3].

Harsha Jain, Surbhi Shrivastava proposed Modern Method for Protection of Induction Motor Using Microcontroller and Wi-Fi Technology. Proposed describe the system for monitoring and controlling of induction motor. A low cost system is proposed to monitor the parameters of the induction motor such as voltage, current, temperature of the windings, speed of the motor and power factor. Continuous monitoring is done for all these factors and a warning massage is sent to in charge person in case of exceed in set limited values. This provision in proposed system facilitates the action to be taken before actual fault occurs on motor, and hence improving the performance of the motor by avoiding its tripping followed by fault. In addition with the monitoring, the speed control of the motor is also performed. If an overload occurs i.e. current exceeds the maximum limit the relay circuit will turn on the buzzer [4].

Ms. Priya V. Kale, Mr. Amit M. Kale, Mr. Nikhil R. Kamdi, Prof. Ankita A. Yeotikar proposed Protection of Induction Motor Using Classical Method. Protection of an Induction Motor (IM) against possible problems such as single phasing, over voltage, over current, overload, dry run as well as under voltage occurring in the Induction Motor. The Electric Motor is a most crucial drive in a modern era of automation. Induction Motors can be protected using some components such as contactors, voltage measurement, current measurement and relays circuit. So, this method is named as the classical method. Classical method involves electrical as well as mechanical dynamic parts. The circuit was fully controlled by microcontroller and it will continuous monitored the voltage, current, speed of the three phases. If any problem come due to fault that normally created in the motor like a protection system which is implemented will monitored working of the induction motor during normal condition and abnormal condition. The motor will get protected from these faults by the operation of the implemented protection system and all the conditions are display by it over the Liquid Crystal Display (LCD) display [5].

2.3 Induction motor

The three-phase induction motor, also called an asynchronous motor, is the most commonly used type of motor in industrial applications. In particular, the squirrel- cage design is the most widely used electric motor in industrial applications [6].

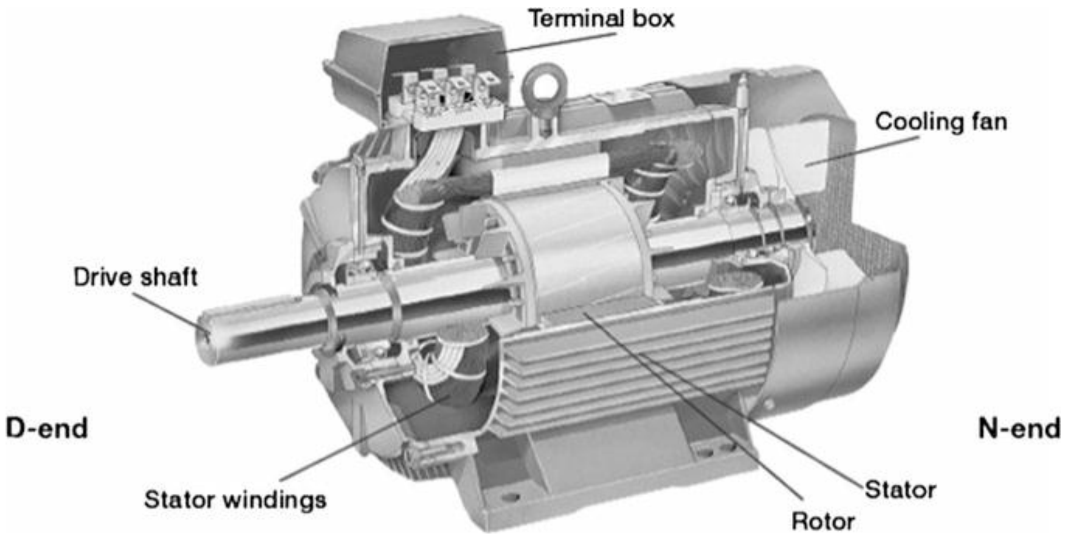

Figure (2.1) Three-phase induction motor

2.3.1 Advantages and Disadvantages of Induction Motor

Advantages:

AC induction motors have their own advantages. These include:

- Simplicity of design – simple & cheap to construct.
- Reliability – they have no brushes or commutator and there is little friction to wear parts away.
- They can be built to suit almost any industrial requirement.
- They are economical and efficient to run for most purposes.
- Poly-phase induction motors are self-starting

Disadvantages:

AC induction motors have some disadvantages as well. These include:

• They only work on AC.

• Their maximum speed is limited by the supply frequency (a 50 Hz supply limits the motor to about 3000 rpm).

• Starting torque is low – they do not get heavy loads moving very quickly.

• They are not as efficient as some other AC motors when used in heavy industrial applications

2.3.2 Types of induction motor

There are basically two types of induction motor that depend upon the input supply, single-phase induction motor and three-phase induction motor.

[A] Single phase induction motor:

The single-phase induction motor is not self-starting. When the motor is connected to a single-phase power supply, the main winding carries an alternating current. It is logical that the least expensive, most reduced up keep sort engine ought to be utilized most regularly. These are of different types based on their way of starting since these are of not self-starting. Those are split phase, shaded pole and capacitor motors. Again capacitor motors are capacitor start, capacitor run and permanent capacitor motors.

In these types of motors the start winding can have a series capacitor and/or a centrifugal switch. When the supply voltage is applied, current in the main winding lags the supply voltage because of the main winding impedance. And current in the start winding leads/lags the supply voltage depending on the starting mechanism impedance. The angel between the two windings is sufficient phase difference to provide a rotating magnitude field to produce a starting torque. The point when the motor reaches 70% to 80% of synchronous speed, a centrifugal switch on the motor shaft opens and disconnects the starting winding.

Applications of Single-Phase Induction Motor:

These are used in low power applications and widely used in domestic applications as well as industrial. And some of those are mentioned below

- Pumps
- Compressors
- Small fans

Single-Phase Induction Motor types:

- Split phase induction motor (with centrifugal switch)
- Capacitor start induction motor
- Capacitor start capacitor run induction motor
- Shaded pole induction motor

[B] Three-Phase Induction Motor:

These motors are self-starting and use no capacitor, start winding, centrifugal switch or other starting device. Three-phase AC induction motors are widely used in industrial and commercial applications. These are of two types, squirrel cage and slip ring motors. Squirrel cage motors are widely used due to their rugged construction and simple design. Slip ring motors require external resistors to have high starting torque. Induction motors are used in industry and domestic appliances because these are rugged in construction requiring hardly any maintenance, that they are comparatively cheap, and require supply only to the stator.

Applications of three-phase induction motor:

- Lifts
- Cranes
- Hoists
- Large capacity exhaust fans
- Driving lathe machines
- Crushers
- Oil extracting mills
- Textile and etc.

Three-Phase Induction Motor types:

- Squirrel cage induction motor
- Slip ring induction motor

2.3.3 Induction motor operation

In a DC motor, supply is needed to be given for the stator winding as well as the rotor winding. But in an induction motor only the stator winding is fed with an AC supply.

- Alternating flux is created around the stator winding due to AC supply. This alternating flux revolves with synchronous speed. The revolving flux is called as "Rotating Magnetic Field" (RMF).
- The relative speed between stator RMF and rotor conductors causes an induced emf in the rotor conductors, the Faraday's law of electromagnetic induction. The rotor conductors are short circuited; hence rotor current is produced due to induced emf. That is why such motors are called as induction motors. (This action is same as that occurs in transformers, hence induction motors can be called as rotating transformers.)
- The induced current in rotor will also create alternating flux around it. This rotor flux lags behind the stator flux. The direction of induced rotor current, according to Lenz's law, is such that it will tend to oppose the cause of its production.
- As the cause of production of rotor current is the relative velocity between rotating stator flux and the rotor, the rotor will try to catch up with the stator RMF. Thus the rotor rotates in the same direction as that of stator flux to minimize the relative velocity.

However, the rotor never succeeds in catching up the synchronous speed. This is the basic working principle of induction motor of either type, single phase or three phases.

2.3.4 Synchronous speed

The rotational speed of the rotating magnetic field is called as synchronous speed:

$$Ns = \frac{60*\mathrm{f}}{P} \tag{2.1}$$

Where, f = frequency of the supply

P = number of pole pairs

2.3.5 Slip

Rotor tries to catch up the synchronous speed of the stator field, and hence it rotates. But in practice, rotor never succeeds in catching up. If rotor catches up the stator speed, there won’t be any relative speed between the stator flux and the rotor, hence no induced rotor current and no torque production to maintain the rotation. However, this won't stop the motor, the rotor will slow down due to lost of torque, and the torque will again be exerted due to relative speed. That is why the rotor rotates at speed which is always less the synchronous speed.

The difference between the synchronous speed (Ns) and actual speed (N) of the rotor is called as slip [7].

$$\%slip\ s = \frac{Ns-N}{\mathrm{Ns}} * 100\% \tag{2.2}$$

2.4 Faults in Induction Motors

Three-phase induction motors are the most popularly used motors especially in industry because of their reliability and simplicity. These motors experience different types of faults during their operation [8]. These faults can be classified as internal and external faults. The external faults include:

1. Over loading
2. Single phasing
3. Unbalanced supply voltage
4. Locked rotor
5. Phase reversal
6. Ground fault
7. under voltage
8. over voltage

A brief description of these faults and their characteristics is given below. Protection of these motors is an important task which has been challenging to engineers. Protective relays were used to monitor these faults and disconnect the motor in case of a fault. Different relays used for protecting against these faults are also described below [8].

2.4.1 Overloading

Overload fault occurs when the mechanical torque exceeds the threshold point by applying mechanical load to the motor greater than full load rating. Overloading causes increase in phase currents, over heating the machine. In a traditional relay protection system, the over current relay trips the motor off-line when the current transformers (CT) encounter over current in the line.

2.4.2 Single phasing

Single phasing is one of the unbalanced cases of the motor. It occurs when one of the three lines are open. More current flows through the other two lines and more heat is generated in stator winding. In the traditional protection systems, a high-set instantaneous trip unit relay is used [8]. Single phasing also gives rise to negative sequence current. A negative sequence relay can also be used to protect against this fault.

2.4.3 Unbalanced

There are many causes of unbalance supply voltages such as unbalance loading, open delta transformers and unequal tap setting. This condition leads to reduction in motor efficiency, raises the motor temperature and excessive unbalanced full load current. Three-phase voltages and currents during an unbalanced supply are shown in Figure (2.2). The protection design should detect over current condition during unbalanced [8].

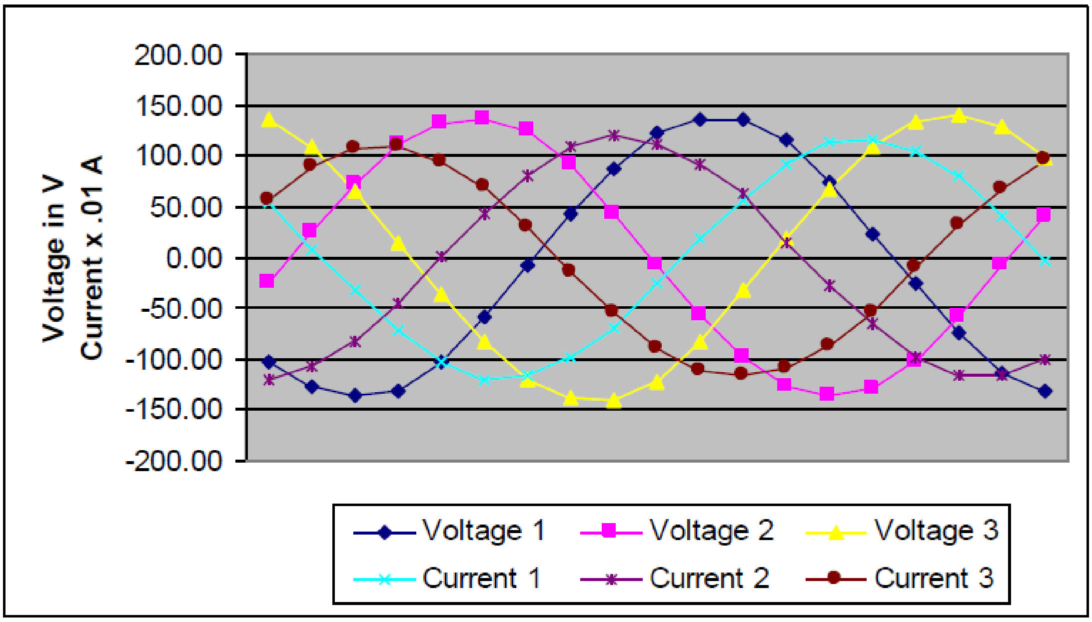

Figure (2.2) Three-phase voltages and currents during an unbalanced supply

2.4.4 Locked Rotor

Locked rotor occurs when the voltage is applied to a non-rotating motor. The stator current may be almost six times its rated value during this condition. There are many causes for this fault to occur for instance, if the rotor shaft is connected to heavy load the motor may experience locked rotor conditions. Locked rotor causes high current which leads to heating the rotor. Therefore locked rotor condition cannot be withstood for a long time. The allowed duration of the motor overloaded under locked rotor condition depends on the voltage applied to the motor terminals. Therefore, the protection system should be able to disconnect the motor when locked rotor condition exceeds the amount of allowed time. Traditional protective systems use over current relays with I-T characteristics [9].

2.4.5 Phase Reversal

Phase reversal occurs when any of the two phases are reversed from the normal sequence, which leads the motor to rotate in the opposite direction. When the motor starts to rotate in the opposite direction, it can cause intensive damage. Therefore, this condition should be corrected immediately. Reverse-phase relays and negative sequence relays are used for the protection [8].

2.4.6 Ground Fault

Ground faults occur when any of the phases touches the ground. Ground faults are more frequent in motors than any other power system, because of their violent condition and frequent starts. The effects of this fault are intensive such as causing hazards to human safety and interference with telecommunication. It can be detected by measuring the ground leakage current [8].

2.4.7 Under Voltage

Under voltage fault is reducing the supply voltage on the three phases by specific percentage, which makes the motor from attaining rated speed in specified time, increases the current and overheats the machine. Low voltage protection relays are used in traditional systems. However, in order to avoid unwanted relay shutdowns due to momentary voltage drops, the AC contacts need a delay mechanism which delays the under voltage protection for a time period. This additional mechanism needs high sensitive devices and involves calibrations [8].

2.4.8 Over Voltage

Over voltage occur if the three phase voltages are greater than rated voltage. The effect of this fault is increasing current flow which leads unacceptable stress on the motor insulation due to high heat dissipation. Conventional protection systems use the over voltage relays to protection the motor during this condition [8].

2.5 Arduino Microcontroller

Arduino is an open-source electronics prototyping platform based on flexible, easy-to use hardware and software. It's intended for artists, designers, hobbyists, and anyone interested in creating interactive objects or environments. Arduino can sense the environment by receiving input from a variety of sensors and can affect its surroundings by controlling lights, motors, and other actuators. The microcontroller on the board is programmed using the Arduino programming language and the Arduino Development Environment. Arduino projects can be stand-alone, or they can communicate with software running on a computer [10].

2.5.1 Arduino Feature

Arduino really simplifies the process of building projects on a microcontroller making it a great platform for amateurs. You can easily start working on one with no previous electronics experience.

Although there are many different types of Arduino boards available, this manual focuses on the Arduino Uno. This is the most popular Arduino board around. So what makes this thing tick? Here are the specifications:

• Processor: 16 MHz ATmega328

• Flash memory: 32 KB

• Ram: 2kb

• Operating Voltage: 5V

• Input Voltage: 7-12 V

• Number of analog inputs: 6

• Number of digital I/O: 14 (6 of them pwm*)*

The specs may seem meager compared to your desktop computer, but remember that the Arduino is an embedded device. Another wonderful feature of the Arduino is the ability to use what are called “Shields”.

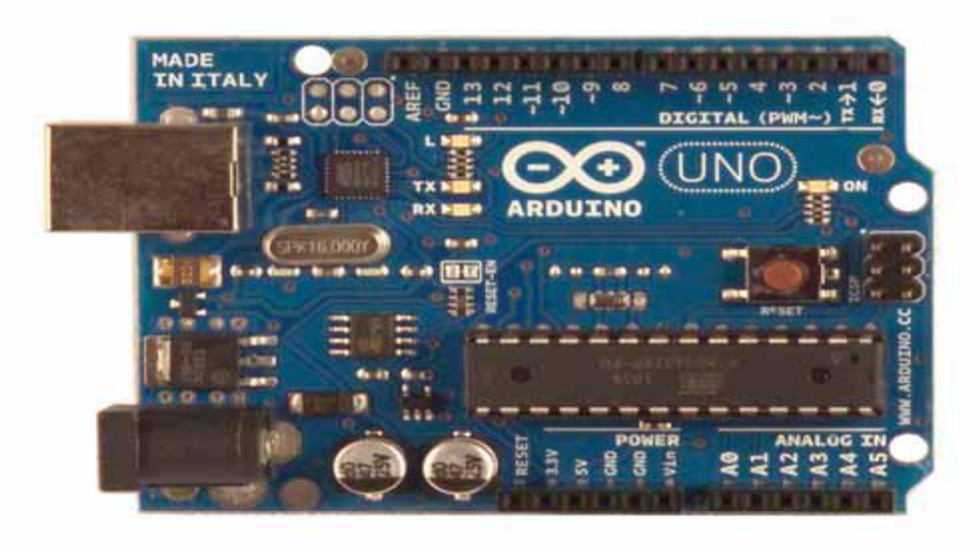

Figure (2.3) Arduino Hardware

The Arduino board is a small-form microcontroller circuit board. At the time of this writing, a number of Arduino boards exist:-

• Arduino Uno

• Arduino Leonardo

• Arduino Lily Pad

• Arduino Mega

• Arduino Nano

• Arduino Mini

• Arduino Mini Pro

• Arduino BT

Figure (2.4) Arduino Family

	Processor	Processor Voltage	Supply Voltage	Flash	SRAM	Digital I/O Pins	PWM Pins	Analog Inputs	Hardware Serial Ports	Dimensions	Shield Compatibility	Notes and Special Features
Uno	16MHz Atmega 328	5v	7-12v	32Kb	2Kb	14	6	6	1	2.1"x2.7" 53x75mm	Excellent (most will work)	
Uno Ethernet	16MHz Atmega 328	5v	7-12v	32Kb	2Kb	14	6	6	1	2.1"x2.7" 53x75mm	Very Good (some pin conflicts)	Has Ethernet Port. Requires FTDI cable to program.
Mega	16MHz Atmega 2560	5v	7-12v	256Kb	8Kb	54	14	16	4	2.1"x4" 53x102mm	Good (some pinout differences)	
Mega ADK	16MHz Atmega 2560	5v	7-12v	256Kb	8Kb	54	14	16	4	2.1"x4" 53x102mm	Good (some pinout differences)	Works with Android Development Kit.
Leonardo	16MHz Atmega 32U4	5v	7-12v	32Kb	2.5Kb	20*	7	12*	1	2.1"x2.7" 53x75mm	Fair (many Pinout Differences)	Native USB capabilities. USB Micro B programming port.
Due	84MHz ARM SAM3X8E	3.3v	7-12v	512Kb	96Kb	54	12	12	4	2.1"x4" 53x102mm	Poor (voltage and pinout differences)	Fastest processor. Most memory. 2-channel DAC. USB micro B programming port. Native micro AB port.
Micro	16MHz Atmega 32U4	5v	5v	32Kb	2.5Kb	20*	7	12*	1	0.7"x1.9" 18x49mm	N/A	Smallest board size. Native USB capabilities
Flora	8MHz Atmega 32U4	3.3v	3.5-16v	32Kb	2.5Kb	8*	4	4*	1	1.75" dia 44.5mm dia	N/A	Sewable Pads. Fabric-friendly design. Native USB Capabilities
DC Boarduino	16MHz Atmega 328	5v	7-12v	32Kb	2Kb	14	6	6	1	0.8"x3" 20.5x76mm	N/A	Can build without headers or sockets for smaller size. Requires FTDI cable for programming
USB Boarduino	16MHz Atmega 328	5v	5v (USB)	32Kb	2Kb	14	6	6	1	0.8"x3" 20.5x76mm	N/A	Can build without headers or sockets for smaller size. USB Mini B programming port.
Menta	16MHz Atmega 328	5v	7-12v	32Kb	2Kb	14	6	6	1	0.8"x3" 20.5x76mm	Excellent (most will work)	Mint-Tin Size and Prototyping Area. Requires FTDI cable for programming.

Table (2.5) Compare between the different types of Arduino

2.5.2 Arduino Software

Arduino microcontrollers are programmed using the Arduino IDE (Integrated Development Environment).Arduino programs, called “sketches”, are written in a programming language similar to C and C++. Every sketch must have a setup () function (executed just once) followed by a loop () function (potentially executed many times); add “comments” to code to make it easier to read. Many sensors and other hardware devices come with prewritten software line for sample code, libraries (of functions). Libraries are a collection of code that makes it easy for you to connect to a sensor, display, module, etc. For example, the built-in Liquid Crystal library makes it easy to talk to character LCD displays. There are hundreds of additional libraries available on the Internet for download [11].

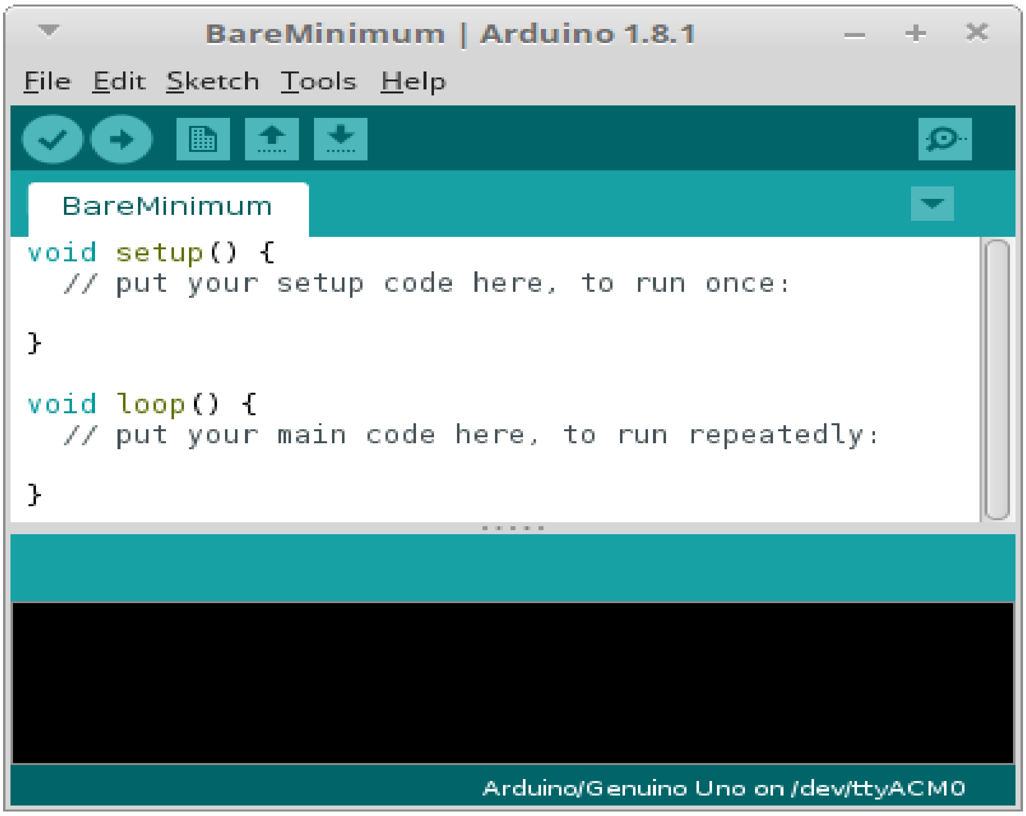

Figure (2.6) Arduino IDE

2.5.3 Arduino Uno

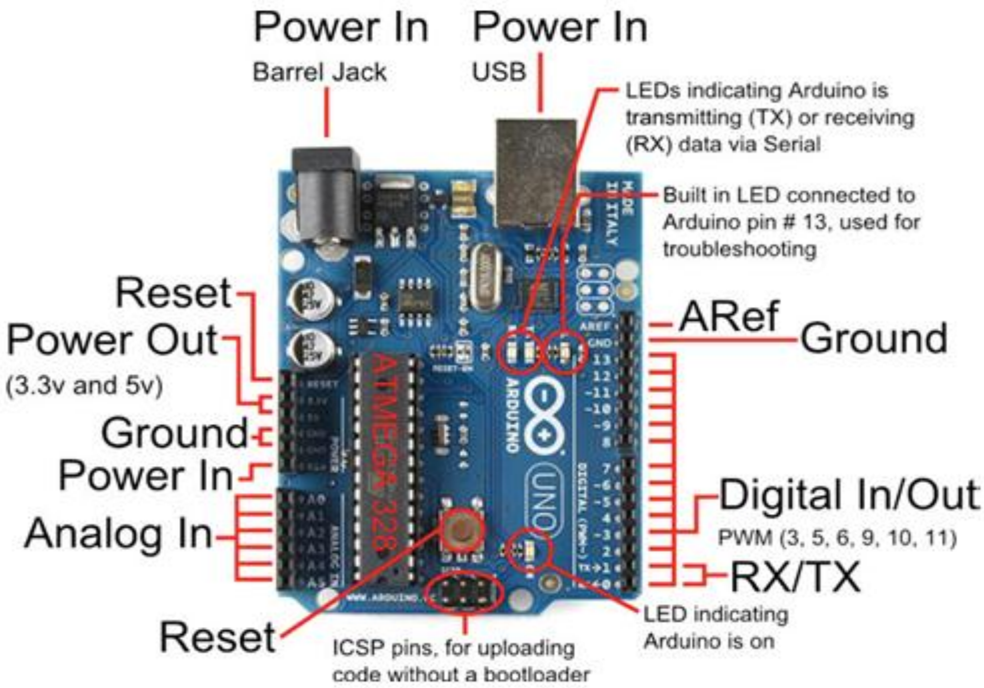

Figure (2.7) Arduino Uno

The Arduino Uno is a microcontroller board based on the ATmega328 (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial converter.

2.5.3.2 Power

The Arduino Uno can be powered via the 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. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. 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 12 volts. 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 5 volts 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.**This pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7 - 12V), the USB connector (5V), or the VIN pin of the board (7-12V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage your board. We don't advise it.
- **3V3.** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- **GND.** Ground pins.

2.4.3.3 Memory

The ATmega328 has 32 KB (with 0.5 KB used for the bootloader). It also has 2 KB of SRAM and 1 KB of EEPROM (which can be read and written with the EEPROM library).

2.5.3.4 Input and Output

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 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. 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. See the attachInterrupt() function for details.
- **PWM: 3, 5, 6, 9, 10, and 11.** Provide 8-bit PWM output with the analogWrite() function.
    - **SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK).** These pins support SPI communication using the SPI library.
    - **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.

The Uno has 6 analog inputs, labeled A0 through A5, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and the analogReference() function. Additionally, some pins have specialized functionality:

- **TWI: A4 or SDA pin and A5 or SCL pin.** Support TWI communication using the Wire library.

There are a couple of other pins on the board:

- **AREF.** Reference voltage for the analog inputs
- **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.

2.5.3.5 Communication

The Arduino Uno has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An ATmega16U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The '16U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, on Windows, a .inf file is required. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer (but not for serial communication on pins 0 and 1). A software serial library allows for serial communication on any of the Uno's digital pins. The ATmega328 also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; see the documentation for details. For SPI communication, use the SPI library.

2.5.3.6 Programming

The Arduino Uno can be programmed with the Arduino software (download). Select "Arduino Uno from the **Tools** > **Board** menu (according to the microcontroller on your board). For details, see the reference and tutorials. The ATmega328 on the Arduino Uno comes preburned with a bootloader that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol (reference, C header files). You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see these instructions for details. The ATmega16U2 (or 8U2 in the rev1 and rev2 boards) firmware source code is available. The ATmega16U2/8U2 is loaded with a DFU bootloader, which can be activated by:

- On Rev1 boards: connecting the solder jumper on the back of the board (near the map of Italy) and then resetting the 8U2.
- On Rev2 or later boards: there is a resistor that pulling the 8U2/16U2 HWB line to ground, making it easier to put into DFU mode.
- You can then use Atmel's FLIP software (Windows) or the DFU programmer (Mac OS X and Linux) to load a new firmware. Or you can use the ISP header with an external programmer (overwriting the DFU bootloader). See this user-contributed tutorial for more information.

2.5.3.7Automatic (Software) Reset

Rather than requiring a physical press of the reset button before an upload, the Arduino Uno is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2/16U2 is connected to the reset line of the ATmega328 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload. This setup has other implications. When the Uno is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Uno. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data. The Uno contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line; see this forum thread for details [12].

2.6 Liquid Crystal Displays (LCD)

LCD is a tool used for visual display of the output. The liquid-crystal display has the distinct advantage of having low power consumption than the LED. It is typically of the order of microwatts for the display in comparison to the some order of milli watts for LEDs. Low power consumption requirement has made it compatible with MOS integrated logic circuit. Its other advantages are its low cost, and good contrast. The main drawbacks of LCDs are additional requirement of light source, a limited temperature range of operation (between 0 and 60° C), low reliability, short operating life, poor visibility in low ambient lighting, slow speed and the need for an ac drive.

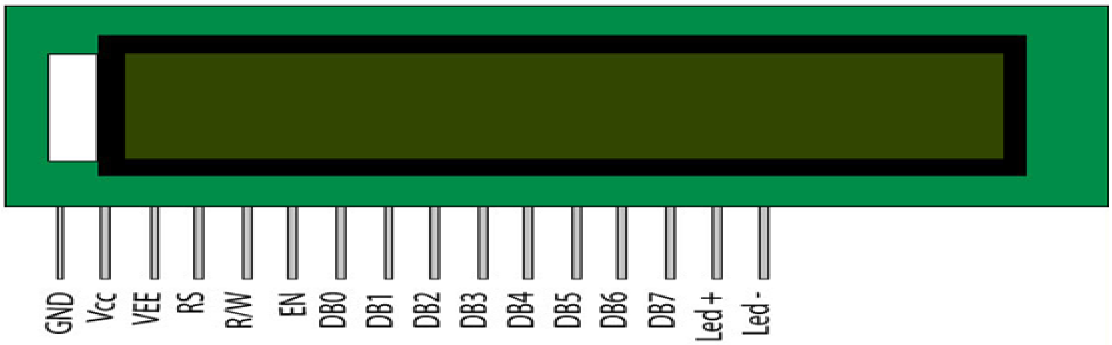

Figure (2.8) LCD

2.6.1 Types of LCDs

There are many types of LCDs.

- Dynamic Scattering: Higher voltage, higher power, less legible, now obsolete
- FLC (Ferroelectric Liquid Crystal) Bistable, faster switching times (~2MHz), can achieve
- Good grayscale by rubbing process.
- TN: Twisted Nematic
- STN: Super-twisted Nematic
- TFT: Thin Film Transistor Active Matrix TN
-

2.6.2 Power Requirements

The LCDs have minimal power requirements. Currently manufactured LCDs consume between 1 and 300 microwatts per square centimeter. This is the lowest power consumption of any display type now available. This very low power consumption allows most LCD products to be battery operated [13].

CHAPTER THREE

SYSTEM DESINGAND

CHAPTER THREE

System Design

3.1 Overview

This chapter is about describing the flow chart, block diagram and the description of the system

3.2 Block Diagram

The block diagram consists of three phase power supply, step down transformer, converting circuit AC to pulses, rectifier circuit, and LCD connected to arduino board.

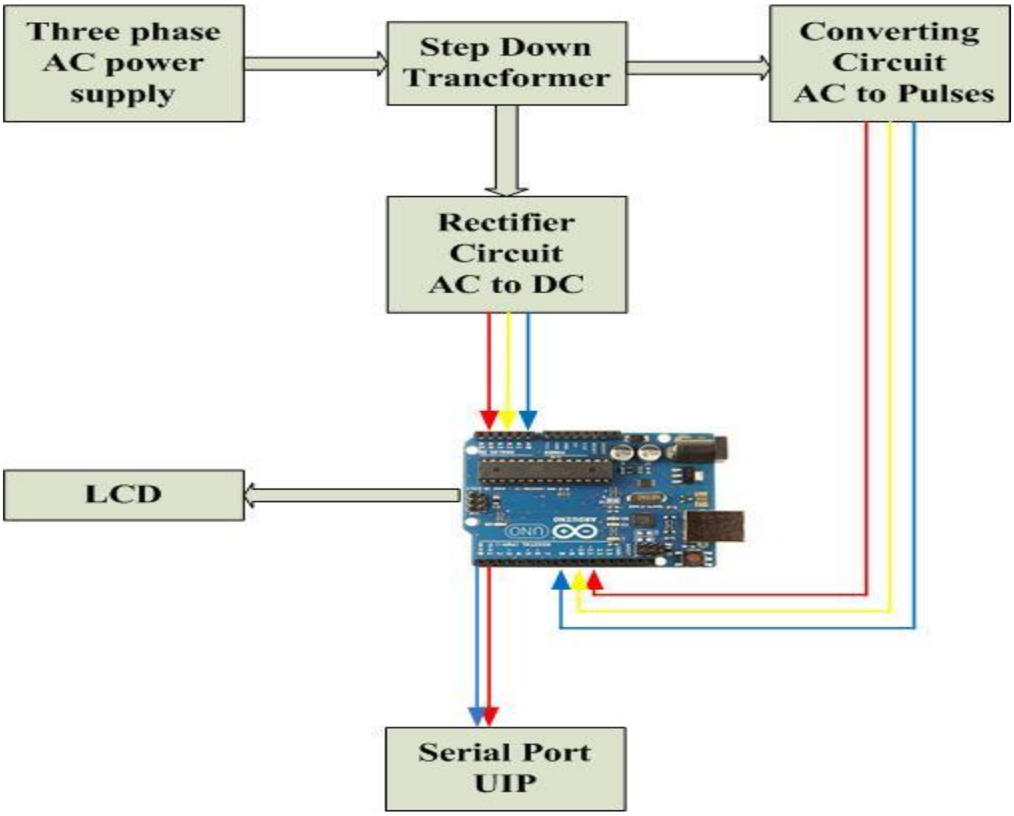

Figure (3.1) Block diagram

3.3 Flow chart

The microcontroller read the data from input pins and starts processing, update information and send it to visual basic through RS232 or LCD directory from microcontroller. Figure 3 shows the flowchart of processing.

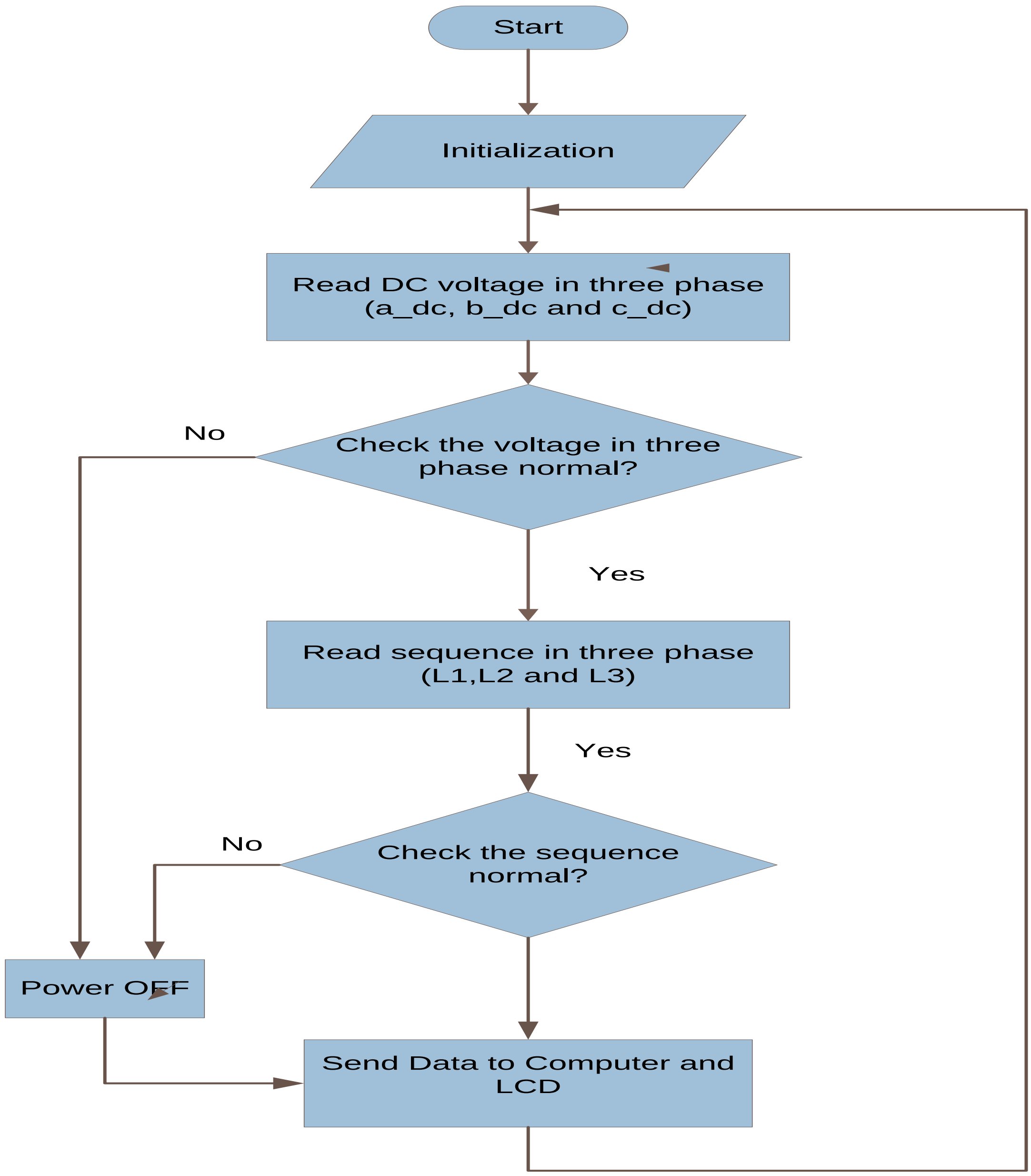

Figure (3.2) flowchart of processing

3.4 Signal follow

As always the AC power connected to the AC motor throw contactors which controlled by controller, also step down transformer used to step down the voltage which used in voltage measurement circuit and phase sequence detection circuit as shown in diagram.

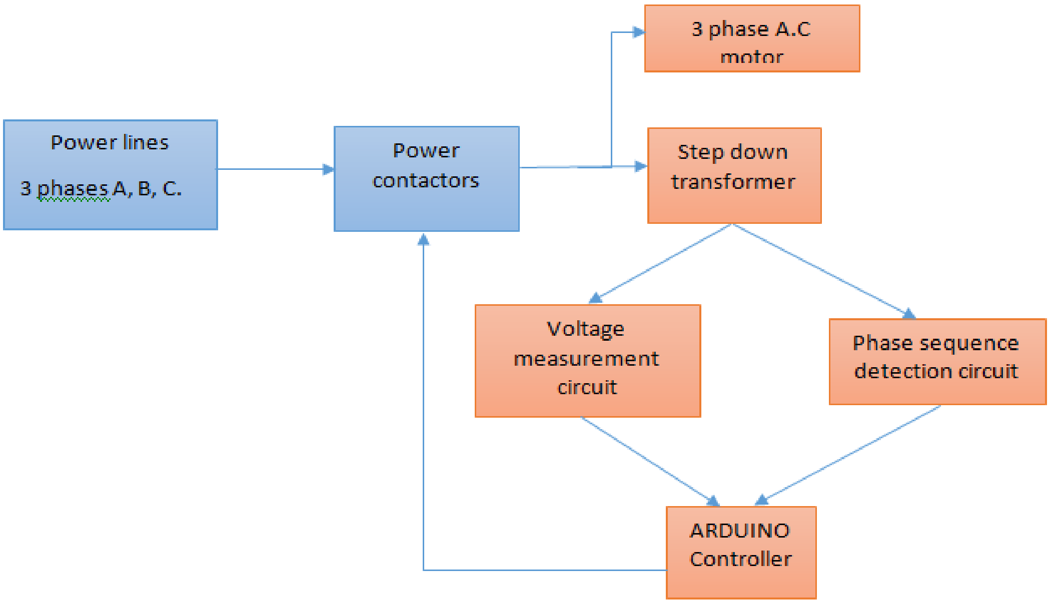

Figure (3.3) signal flow in the system

3.5 Description of the System

The system has being installed using the following components:

- Micro controller Arduino Uno.
- Three phase Power supply.
- Step down transformers.
- Rectifier circuit.
- Relay.
- LCD.
- User interface panel.

3.6 Signal movement

The system is designed as follow:

1. Three phase lines has been named as A_dc, B_dc and C_dc respectively connected to analog inputs of microcontroller pins A0,A1 and A2 respectively.
2. Phase sequence detector has been named as L1, L2 and L3 respectively connected to digital inputs of microcontroller pins (8, 9 and 10) respectively.
3. Led 16X2 (Rs,E,D4,D5,D6 and D7) has been connected to digital inputs of microcontroller pins (12,11,5 4,3 2)
4. Contactor of the induction motor has been named as RELAY and connected from microcontroller pins 6.
5. Power supply (pin 7)
6. Alarm massage has been used serial port from microcontroller pins (0, 1).

3.7 System Implemention

Control circuit has many parts three phase power supply, transformer step down, voltage divider, half wave rectifier and LCD.

There are two cases when implementing the system. The first case happen when phase failure and voltage is not in the range and second case happen when phase.

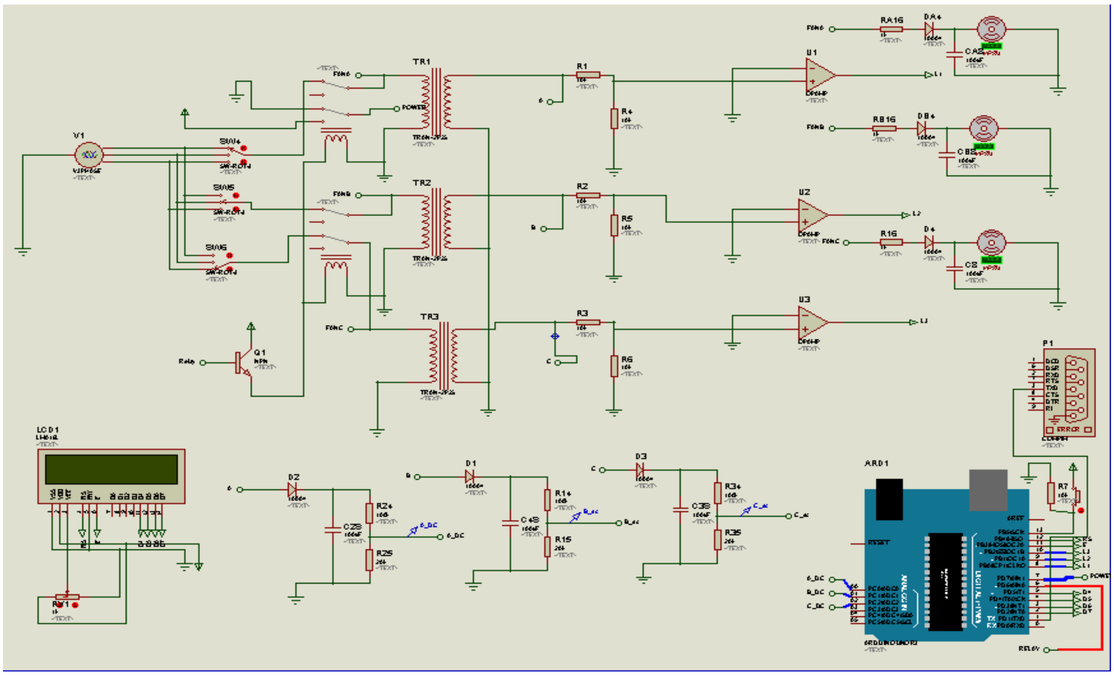

Figure (3.4) circuit diagram of the system

3.7.1 Phase detection circuit

To detect the phase sequence firstly operation amplifier was used to convert sine wave to squire wave form and compare it in software, the software program check the sequence and send the state of sequence as logic bit (0 or 1) to monitor system (PC) in case there are fault in phase sequence the controller cutoff the power throw dis energizing the contactors and LCD display message (Sequence abnormal).

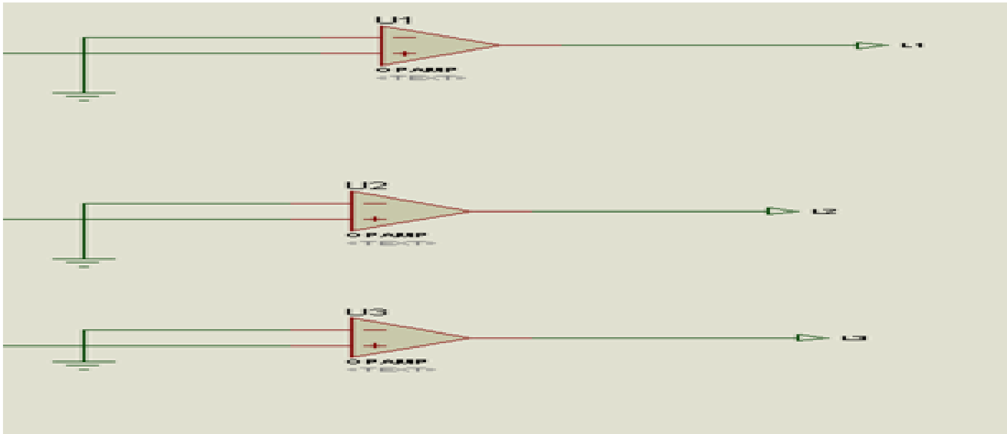

Figure (3.5) signal condition circuit

3.7.2 Voltage measurement circuit

The main contents in this circuit half wave rectifier and RC filter and voltage divider resistors, when AC voltage changed the DC voltage change also, so DC voltage used to indicate the AC voltage level, this circuit used for any phase individually. Inside the controller (ARDUINO UNO) analog to digital convertor (ADC) used to measure the DC voltage and compare it with normal levels and send messages throw LCD also send the values of voltage to monitor interface program (PC).

- Step down transformer with factor (0.075).
- Out voltage from transformer inserted to have wave rectifier and softening circuit stabilize the DC voltage.
- Voltage divider was used to reduce DC voltage to fits input range of Arduino ADC from (0-5) V.
- Maximum AC voltage was measured (400 V/50 Hz) to get out DC (5 v) and Minim AC voltage 0V-0VDC voltage.
- Equations using in circuit to convert AC voltage in three phase power supply to DC voltage into arduino:

Vdc=(Vac*Tfactor-Vdiode)*voltage divider (3.1)

Vdc=[[(Vac*0.075)-0.7]*(10k/(10k+50k))]- Loses

Loses= 0.449V

Vdc=[(Vac*0.075)-0.7]*(1/6)]- 0.1433

Decimal value from ADC = Vdc *(1023/5) (3.2)

Vac	Vdc	Decimal (ADC)
400V	4.43V	906
300V	3.31V	677
200V	2.19V	448
100	1.07	347
0V	0V	0V

Table (3.1) Comparing between AC and DC voltage

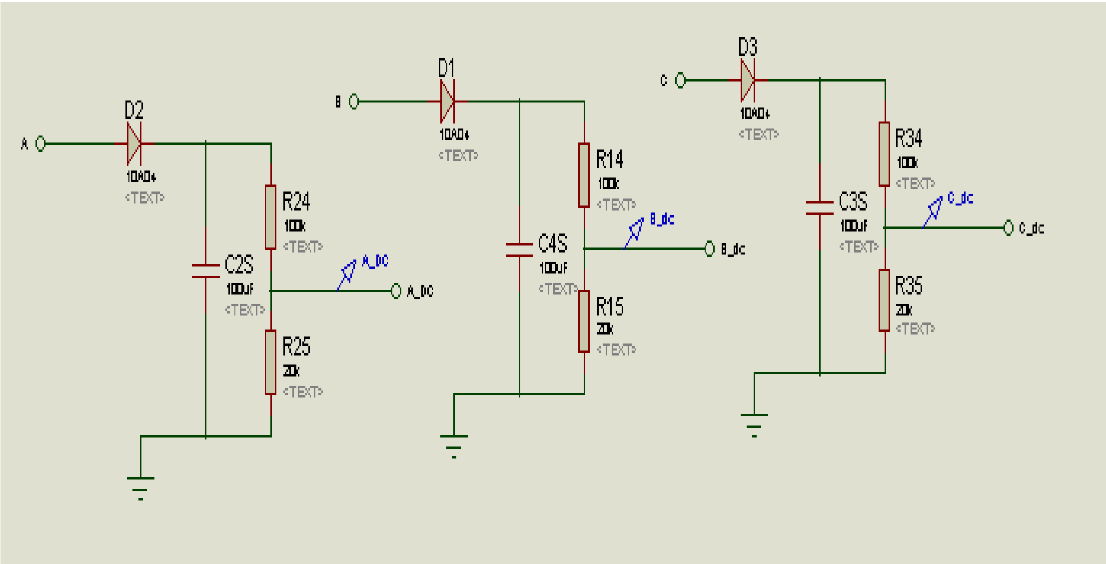

Figure (3.6) Voltage measurement circuit

CHAPTER FOUR

RESULT AND DISCUSSION

CHAPTER FOUR

RESULTS AND DISCUUSION

4.1 Operation Concept

The main objective of this system is to protect the motor from phase failure, phase unbalance and phase insequence which can be achieved using microcontroller arduino UNO. All the manual operations are replaced by sending signals from the microcontroller to the main contractor. As shown in figure below.

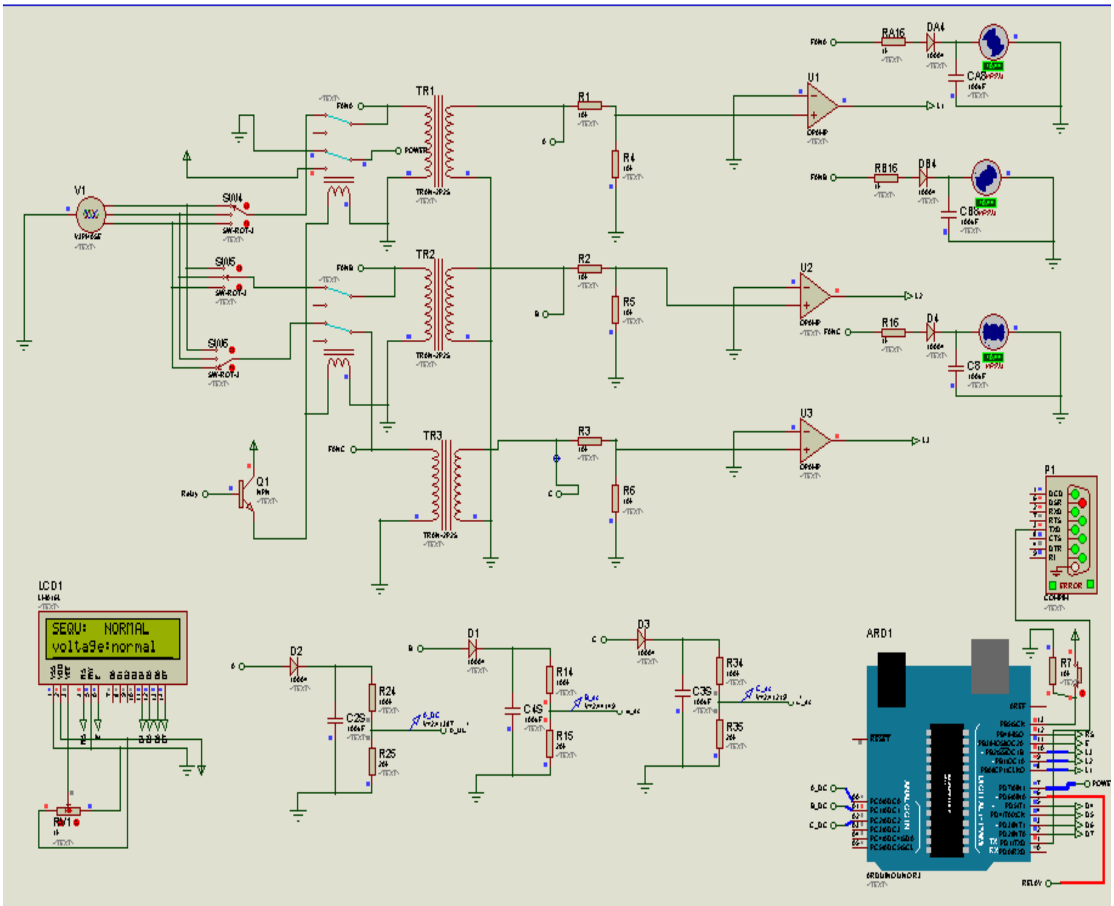

Figure (4.1) general schematic of design

4.2 Working Phases

The working of the design can be classified into three stages:

4.2.1 Phase sequence identification:

The system start converting the three phase input power supply into pluses which can be easy to identify the phase in sequence or not, Figures 1&2 below shows the schematic circuit and there signals. I used microcontroller to identify the shift between the three lines is normal or abnormal, I apply the pulses of line1 , line2 and line3 to the microcontroller pins to detect the sequence is correct and the angle shift is 120,240 respectively related to line1.

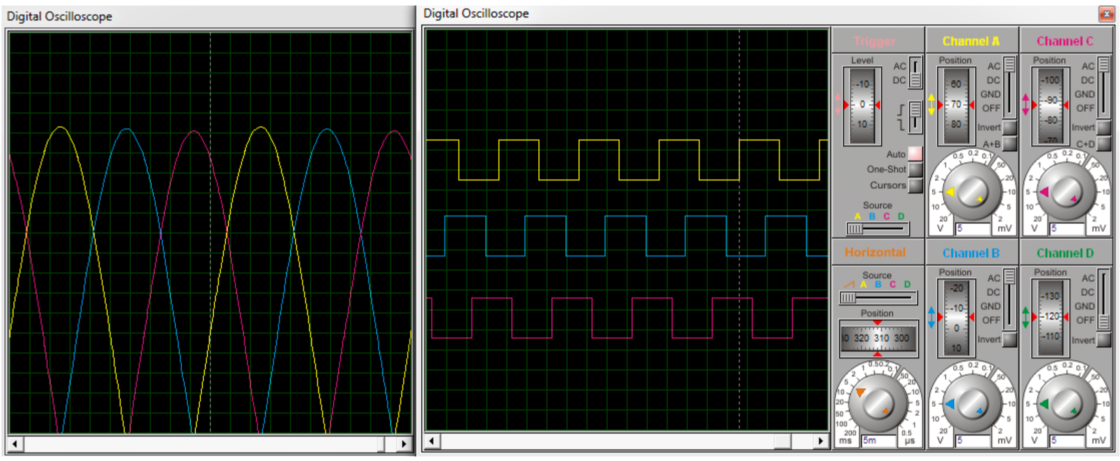

Figure (4.2) converting the three phase into pluses

4.2.2 Voltage identification

The system works normally for [200-240] volt Ac power supply, through ADC pins of microcontroller the system read the value of voltage for the three lines to identify the under voltage, over voltage and unbalance voltage between the three lines and also to detect if one line or more is failure. Figure 4.3 below shows that the schematic of converting three phase power supply into DC voltage.

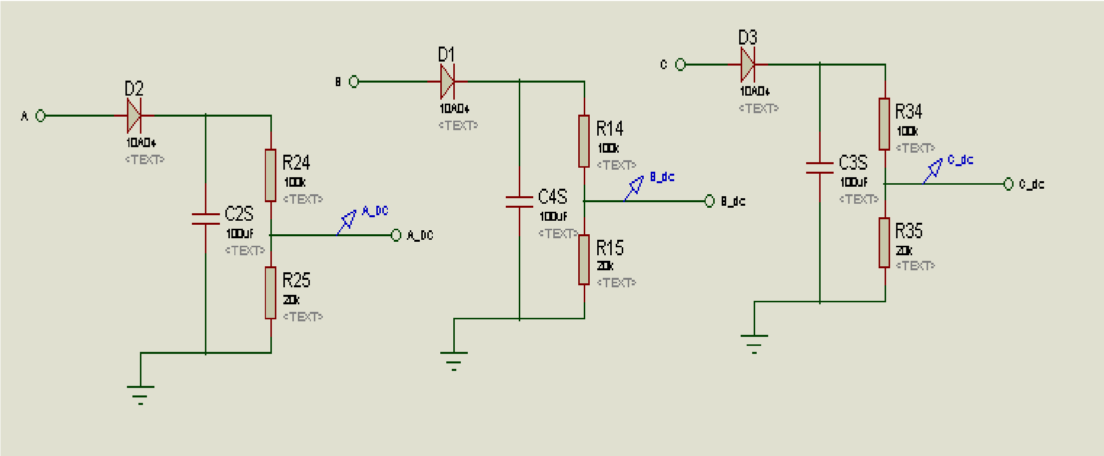

Figure (4.3) converting three phase power supply into DC voltage

3.2.3 User Interface Panel

Visual basic was used to design the UIP for observation of motor behavior it can offer you the voltage of three lines (L1, L2&L3), indication of the phase sequence, indication of under-over voltage, indication of unbalance voltage and phase failure indication continuously. Figure 4.4 shows the form of UIP.

Figure (4.4) UIP massage

4.3 Simulation Results

Schematic circuit in protuse simulation have two cases, the first case phase insequance, in this case arduino check the sequence in three phase. The second case unbalanced voltage, in this case arduino check the voltage if it range or not and send massage to relay and UIP.

4.3.1 Phase insequance case

In this case phase 1, phase 2 is reversed and phase 3 is normal. The system automatically cut off the power supply and at the same time send a message through serial port to the UIP to inform the local operator. Figure 4.5 below shows the schematic and UIP message.

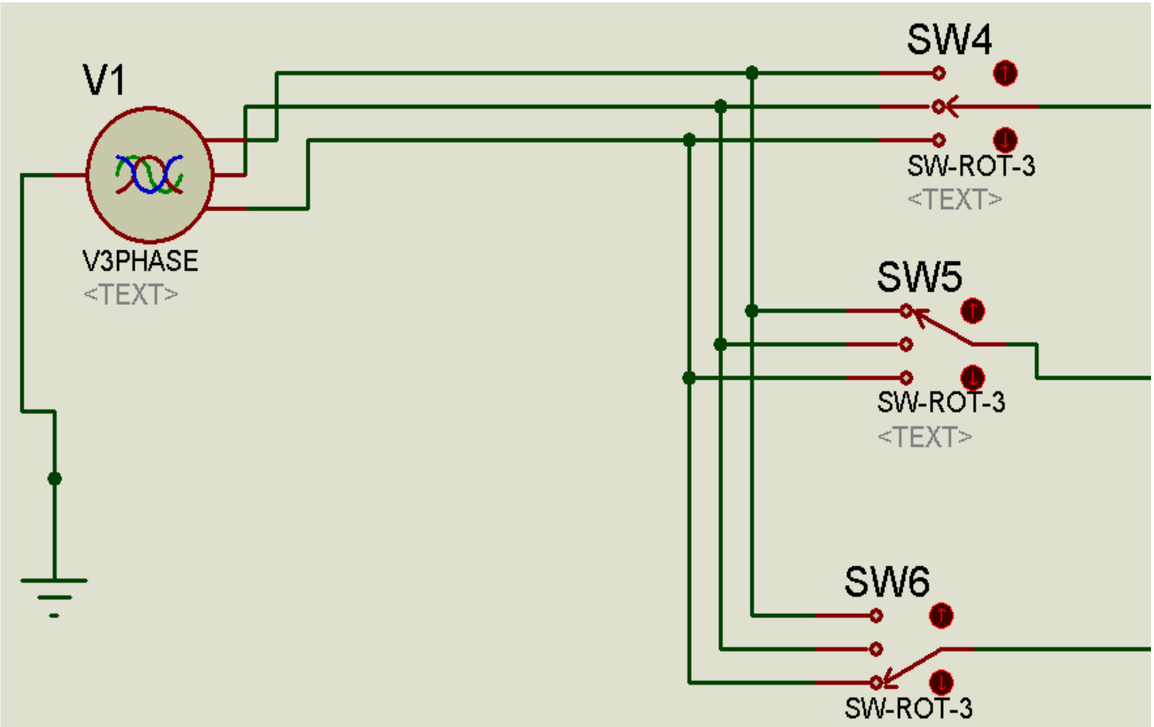

Figure (4.5 a) phase insequence

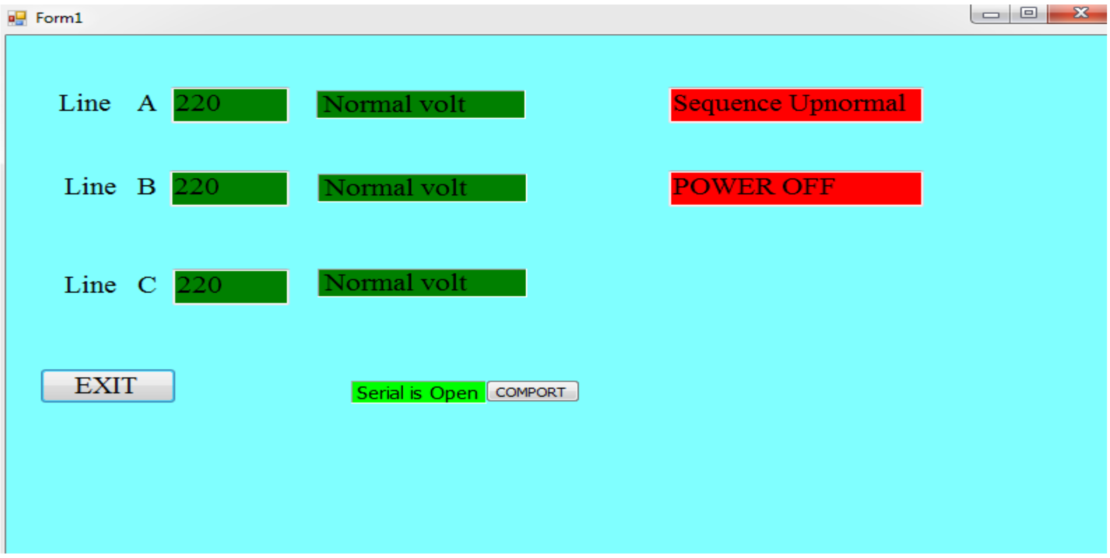

Figure (4.5 b) UIP phase insequence message

4.3.2 Over-Under Voltage Case

In this case the power supply out of rang [200-240] volt AC, 50 HZ. The system automatically cut off the power supply and at the same time send a message through serial port to the UIP to inform the local operator. If power supply less than 200 volt the message will be Under Voltage else the message will be Over Voltage .Figure 4.6 below shows the schematic and UIP message.

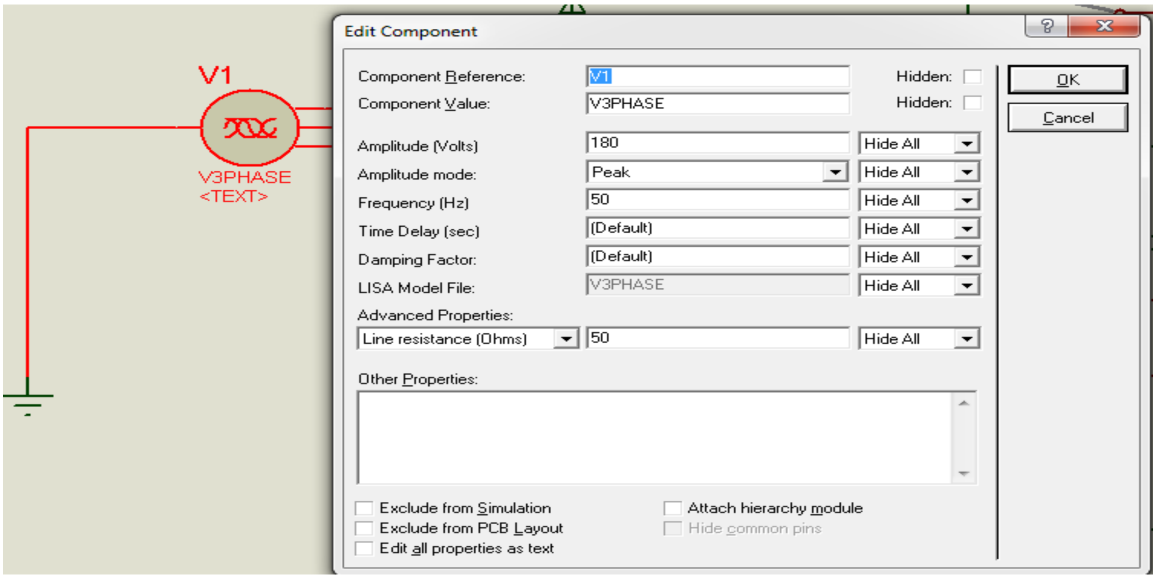

Figure (4.6 a) power supply edit setting panel

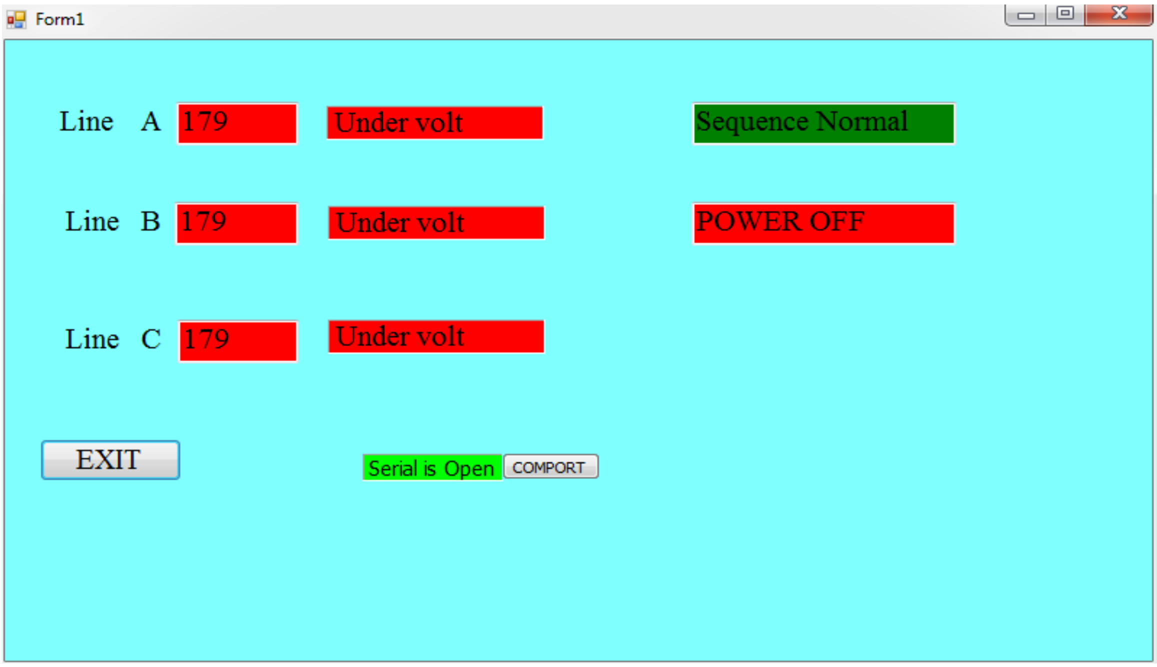

Figure (4.6 b) UIP under voltage message

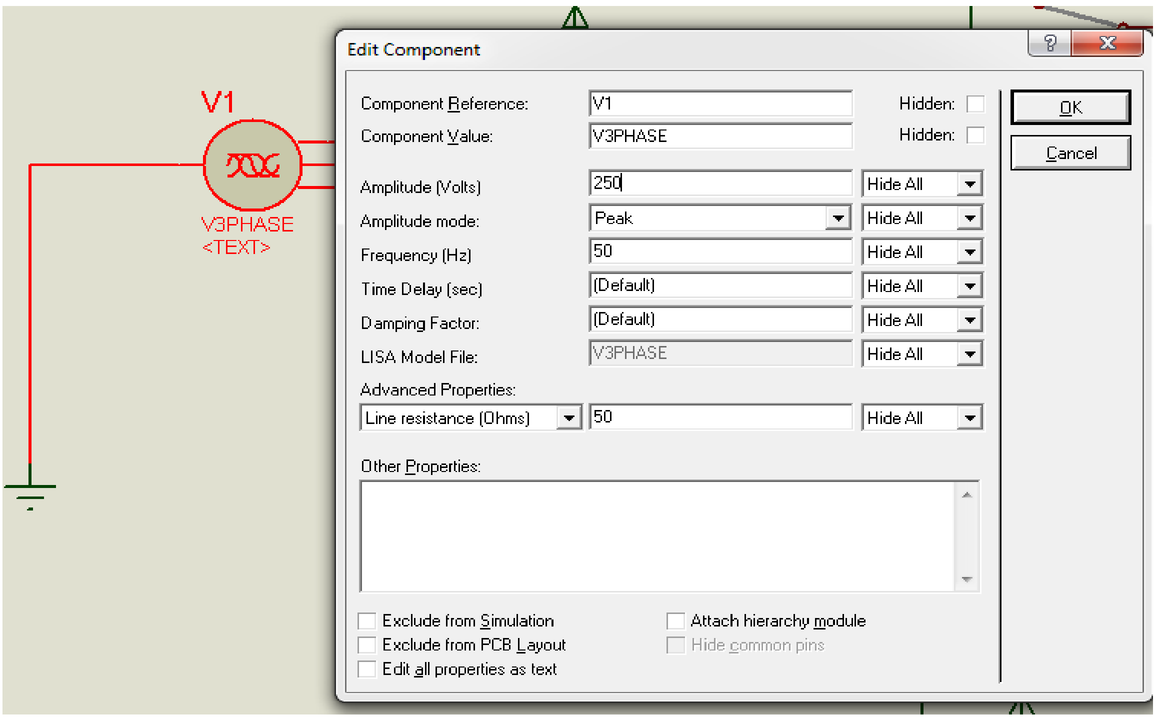

Figure (4.6 c) power supply edit setting panel

Figure (4.6 d) UIP overvoltage message

CHAPTER FIVE

CONCLUSION AND RECOMMENDATION

CHAPTER FIVE
Conclusion and Recommendations

5.1 Conclusions

In my thesis I wrote basic information about protection of induction motor. In the first I explained what the induction motor is, how it works and how it is constructed. I started theoretical principles and I explained what kind of damage there are and what methods of protection apply. Protection of induction motor is important for long use motor. This faults identifier detects three types of external faults experienced by the three-phase induction motor and also no fault condition. These faults are three phasing, unbalanced supply voltage, under voltage, over voltage and phase sequence.

In protues software first I designed schematic circuit for converting three phase power supply into pluses and then feed the microcontroller with this signal to detect the sequence is normal. Secondly I added the rectifier circuit to convert AC voltage to DC voltage to detect voltage profile is in the range. Finally I designed user interface panel to monitoring the induction motor status.

After I programmed the microcontroller and designed schematic circuit and configure the user interface panel with the protues software the system is work successfully for all the cases.

5.2 Recommendation for Future Work

This project still has many improvements that should be done to improve it accuracy and reliability. There are some suggestions for the future research and development:

- Would be to build the real-time implementation system for the three-phase induction motor faults identification based on the proposed microcontroller method in this project.
- Apply microcontroller based method for identifying faults in different power systems components such as synchronous generators and transformers.
- Build the real-time implementation system for the three-phase induction motor faults identification based on the proposed SVM method.
- Insert automatic control system in motor using control board.

REFRENCES

[1] PalashKundu and ArabindaDas”microprocessor–based system for identification of phase sequence and detection of phase unbalance of three –phase ac supply”journal of scientific &industrial research vol.68, ,pp.597-604 Electrical Engineering Department, Jadavpur University, Kolkata 700 032,India July 2009.

[2] Aderiano M. da Silva, B.S. “INDUCTION MOTOR FAULT DIAGNOSTIC AND MONITORING METHODS” Marquette University, Milwaukee, Wisconsin May 2006.

[3] A.Selvanayakam 1, W.Rajan Babu 2, S.K.Rajarathna 3 “ Fault Detection in Three Phase Induction Motor” International Journal of Innovative Research in Science, Engineering and Technology On 20th & 21st March Organized **by** Sri Krishna College of Engineering & Technology, Kuniamuthur, Coimbatore-641008, Tamilnadu, India.

[4] Harsha Jain, Surbhi Shrivastava “Modern Method for Protection of Induction Motor Using Microcontroller and Wi-Fi Technology” International Journal of Innovative Research in Computer and Communication Engineering Vol.4,Issue6,Wainganga College of Engineering, Nagpur, India June 2016.

[5] Ms. Priya V. Kale1, Mr. Amit M. Kale2, Mr. Nikhil R. Kamdi3, Prof. Ankita A. Yeotikar4 “ PROTECTION OF INDUCTION MOTOR USING CLASSICAL METHOD’’ International Research Journal of Engineering and Technology (IRJET) Volume: 04 Issue: 01 Bachelor Student, Electrical Engineering, DES’sCOET Dhamangaon (Rly), Maharashtra, India Jan -2017.

[6] Sprecher+Schuh AG Rockwell Automation, Aarau **“**Application basics of operation of three-phase induction motors “1996.

[7] Ibatullayev Aibek "Induction motor protection systems" Faculty of Electrical Engineering, Prague University May 2017.

[8] Walter A Elmore Protective Relaying Theory and Applications. New York, NY: Marcel 2004.

[9] Sudha, M, & Anbalagan, P. A Protection Scheme for Three-Phase Induction Motor from Incipient Faults Using Embedded Controller. Asian Journal of Scientific Research, 28-50 2009.

[10] RupinderSaini, NarinderRana, Comparison of Various Biometric Methods, March 2014.

[11] James Flaten, MN Space Grant Consortium, Christopher Gosch, Chad Serba, Introduction to Arduino.

[12] www.arduino.cc

[13] www.circuitstoday.com/liquid-crystal-display-lcd-working.

APPENDEX

Appendix: (A)

IDE program:

```
#include <LiquidCrystal.h>

#include <LiquidCrystal.h>


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

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


const int L1 = 8;

const int L2 = 9;

const int L3 = 10;

const int op= 6;

int a_dc,b_dc,c_dc,k=0,x,m=1;

int count=0;

 boolean L1State =1;

 boolean L2State = 1;

 boolean   L3State =1;

void setup() {

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

  lcd.begin(16, 2);

  Serial.begin(9600);

   pinMode(L1, INPUT);
```

```
   pinMode(L2, INPUT);
    pinMode(L3, INPUT);
    pinMode(op, OUTPUT);
    pinMode(13,INPUT);pinMode(7,INPUT);
     lcd.clear();  //lcd clear

}

void loop() {
 //------------------------- Read voltage values
    a_dc=analogRead(0);   //-----phase A
    delay(10);
    b_dc=analogRead(1);   //-----phase B
    delay(10);
    c_dc=analogRead(2);   //-----phase C
    delay(10);
  Serial.println(a_dc);
  delay(10);
  Serial.println(b_dc);
  delay(10);
  Serial.println(c_dc);
  delay(10);
     //-------------------------voltage condition
     if (a_dc>538){lcd.setCursor(0,1); lcd.print("over voltage:A") ;
delay(100);k=1; }
```

```
if(b_dc>538){lcd.setCursor(0,1); lcd.print("over voltage:B");delay(100);k=1; }

if(c_dc>538){lcd.setCursor(0,1); lcd.print("over voltage:C"); delay(100);  k=1; }

if(a_dc<448){lcd.setCursor(0,1); lcd.print("under volt : A"); delay(100); k=1; }

if(b_dc<448){lcd.setCursor(0,1); lcd.print("under volt : B"); delay(100);  k=1;}

if(c_dc<448){lcd.setCursor(0,1); lcd.print("under volt : C"); delay(100);  k=1;}

if(k==0){lcd.setCursor(0,1); lcd.print("voltage:normal ");}


//----------------------Sequence condition

count=0;

L1State = digitalRead(L1);

while(L1State==LOW){count=0;L1State = digitalRead(L1);};

while(L1State==HIGH){count=0;L1State = digitalRead(L1);};

L2State = digitalRead(L2);

while(L2State==HIGH){

delay(1);

count=count+1;

//if (count>20){k=1;break;}

L2State = digitalRead(L2);

}

if((count<=8)&&(count>5)){

  lcd.setCursor(0,0);
```

```
    lcd.print("SEQU:  NORMAL ");
   m=1;k=0;}
else {
    lcd.setCursor(0,0);
    lcd.print("SEQU: UPNORMAL");
    k=1;m=0;
  }
  //--------------------------------------
  if (m==1){
    Serial.println(49);
  }
  else {Serial.println(48);}
  delay(10);
   if(k==1){digitalWrite(op,HIGH);delay(1500);}
  else{delay(10);digitalWrite(op,LOW);}
    if (digitalRead(7)==HIGH){
    Serial.println(48);
  }
  else{
  Serial.println(49);}
  k=0;
```

Appendix: (B)

UIP Program:

```
Dim seq As Integer, POWER As Integer = 49
    Dim x As Byte
    Dim xa As Integer, xb As Integer, xc As Integer
    Dim a As String, b As String, c As String
    Dim factor As Single = 400 / 908.4
    Dim max As Integer = 538, min As Integer = 448
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button1.Click
        End
    End Sub
    Private Sub rec()
        If SerialPort1.IsOpen = True Then
            Label4.Text = "Serial is Open"
            Label4.BackColor = Color.Lime
            xa = CInt(a)
            xb = CInt(b)
            xc = CInt(c)
            If (xa > 0 And xb > 0 And xc > 0) Then
                TextBox1.Text = CInt(xa * factor) + 2
                TextBox2.Text = CInt(xb * factor) + 2
                TextBox3.Text = CInt(xc * factor) + 2
            Else
                TextBox1.Text = CInt(xa * factor)
                TextBox2.Text = CInt(xb * factor)
                TextBox3.Text = CInt(xc * factor)
            End If
            If seq = 48 Then
                TextBox5.Text = "Sequence Upnormal"
                TextBox5.BackColor = Color.Red
            Else
                TextBox5.Text = "Sequence Normal"
                TextBox5.BackColor = Color.Green
            End If
            If POWER = 48 Then
                TextBox4.Text = "POWER OFF"
```

```
    TextBox4.BackColor = Color.Red

Else
    TextBox4.Text = " POWER ON"
    TextBox4.BackColor = Color.Green
End If
If xa > max Then
    TextBox1.BackColor = Color.Red
    Label5.Text = "Over volt "
    Label5.BackColor = Color.Red

ElseIf xa < min Then
    TextBox1.BackColor = Color.Red
    Label5.BackColor = Color.Red
    Label5.Text = "Under volt"

Else
    TextBox1.BackColor = Color.Green
    Label5.BackColor = Color.Green
    Label5.Text = "Normal volt"

End If

If xb > max Then
    TextBox2.BackColor = Color.Red
    Label6.BackColor = Color.Red
    Label6.Text = "Over volt "

ElseIf xb < min Then
    Label6.BackColor = Color.Red
    TextBox2.BackColor = Color.Red
    Label6.Text = "Under volt "

Else
    TextBox2.BackColor = Color.Green
    Label6.BackColor = Color.Green
    Label6.Text = "Normal volt"

End If

If xc > max Then
    Label7.BackColor = Color.Red
    TextBox3.BackColor = Color.Red
```

```
            Label7.Text = "Over volt "

        ElseIf xc < min Then
            Label7.BackColor = Color.Red
            TextBox3.BackColor = Color.Red
            Label7.Text = "Under volt "



        Else
            TextBox3.BackColor = Color.Green
            Label7.BackColor = Color.Green
            Label7.Text = "Normal volt"

        End If
    Else
        Label4.Text = "Serial is closed"
        Label4.BackColor = Color.Red

    End If

End Sub
Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
    SerialPort1.Open()
    SerialPort1.ReceivedBytesThreshold = 1
    If SerialPort1.IsOpen = True Then
        TextBox4.BackColor = Color.Cyan
    Else
        TextBox4.BackColor = Color.DarkBlue
    End If

End Sub

Private Sub Timer1_Tick(ByVal sender As Object, ByVal e As
System.EventArgs) Handles Timer1.Tick
        rec()
End Sub

Private Sub SerialPort1_DataReceived(ByVal sender As Object,
ByVal e As System.IO.Ports.SerialDataReceivedEventArgs) Handles
SerialPort1.DataReceived

    a = SerialPort1.ReadLine()
```

```
        b = SerialPort1.ReadLine()
        c = SerialPort1.ReadLine()
        seq = SerialPort1.ReadLine()
        POWER = SerialPort1.ReadLine()
    End Sub

    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button2.Click
        SerialPort1.Open()
    End Sub
End Class
```