Sudan University of Science and Technology
College of Graduate Studies

An Embedded Remote Monitoring System for Cathodic Protection

نظام المراقبة عن بعد المدمج للحماية الكاثودية

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

Prepared by:

Ehab Ali Hassan Alhaj

Supervised by:

Dr. Omer Abdel Razag Sharif Abubaker

August 2018

الآية

بسم الله الرحمن الرحيم

﴿ اقْرَأْ بِاسْمِ رَبِّكَ الَّذِي خَلَق (١) خَلَقَ الإِنسَانَ مِنْ عَلَقٍ(٢) اقْرَأْ وَرَبُّكَ الأَكْرَمُ(٣) الَّذِي عَلَّمَ بِالْقَلَمِ (٤) عَلَّمَ الإِنسَانَ مَا لَمْ يَعْلَمْ(٥)﴾ سورة العلق

﴿ يَرْفَعِ اللَّهُ الَّذِينَ آمَنُوا مِنكُمْ وَالَّذِينَ أُوتُوا الْعِلْمَ دَرَجَاتٍ وَاللَّهُ بِمَا تَعْمَلونَ خبِير(١١)﴾ سورة المجادلة

DEDICATION

This research is dedicated to:

- The sake of ALLAH, my creator and my master.
- My great teacher and messenger, Mohammed (May ALLAH bless and great him).
- My great parents, who never stop giving of themselves in countless ways.
- My dearest wife, who leads me through the valley of darkness with light of hope and support.
- My beloved brothers and sisters, who stand by me when things look bleak.
- My friends for thier encourage and support.
- All the people in my life who touch my heart.

ACKNOWLEDGMENT

Thanks to merciful ALLAH for all the countless gifts he has offered me, and thanks to my family for their love and support.

It is a great pleasure to acknowledge my deepest thanks and gratitude to Dr. Omer Abdel Razag Sharif, for his kind supervision, encouragement, and comprehensive advice until this work came to existence. It is a great honor to work under his supervision.

I would like to express my extreme sincere gratitude and appreciation to Eng. Muaaz A. Rahim, for his kind endless help, generous advice and support during the study.

All thanks due to my family members especially my father, mother, wife, sisters, and brother for their prayers, patient and encourage which have helped me in becoming what I am today.

ABSTRACT

The cathodic protection remote monitoring system is a circuit senses the measured voltage used to protect metallic structures against corrosion and display it locally at the field and then transfers it to monitoring room. In this research, two microcontroller-based subsystems have been designed: the field and monitoring sides. These two subsystems are connected together by two wireless Xbee modules.

To design and simulate this proposed system, a Proteus ISIS simulator and ATMEL studio software have been used. The monitoring side is equipped with alarming Light Emitting Diodes (LEDs) to monitor the in-range, over and under protection ranges according to NACE standard. The simulated and implemented results were coincided, display and wireless communication were done in high accuracy and the alarm has been activated when the sensed value was out of protection range.

المستخلص

نظام مراقبة الحماية الكاثودية عن بعد هو دائرة تقوم بقراءة الجهد المستخدم في حماية الاجسام المعدنية ضد التآكل وعرضه محلياً في الحقل ثم نقله الى غرفة المراقبة. في هذا البحث تم إقتراح تصميم لهذا النظام يتكون من وحدتين من أنظمة المتحكمات الدقيقة بكل من موقع الجسم المعدني (مستودعات خام البترول) و غرفة المراقبة البعيدة. تم ربط الوحدتين بواسطة جهازي ارسال واستقبال لاسلكيين. لتصميم ومحاكاة النظام المقترح، تم إستخدام برنامج المحاكاة بروتوس وبرنامج آتمل ستوديو. تم تجهيز غرفة المراقبة يوحدة تنبيه تتكون من ثنائيين ضوئيين لمراقبة مستوى الحماية وفقاً لمعيار الرابطة الوطنية لمهندسي التآكل. نتائج المحاكاة و التنفيذ للنظام المقترح كانت متطابقة، حيث أن العرض والاتصال اللاسلكي تم بدقة عالية وتم تفعيل الانذار عندما كانت قيمة الفولتية المقاسة خارج مدى الحماية.

TABLE OF CONTENTS

CHAPTER FOUR: RESULTS AND DISCUSSION

CHAPTER FIVE: CONCLUSION AND RECOMMENDATIONS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

AC	Alternating Current
ADC	Analogue to Digital Convertor
ARM	Advanced RISC Machine
AVO meter	Ampere Volt Ohm meter
AVR	Alf and Vegard's RISC
BS	Base Station
CP	Cathodic Protection
CPF	Central Processing Facility
CPU	Central Processing Unit
CSE	Copper sulfate reference electrodes
DC	Direct Current
DDR	Data Direction Register
EEPROM	Erasable Electrical Programmable Memory
GND	Ground
GPRS	General Packet Radio Service
GPS	Global Positioning System
GSM	Global System for Mobile
IC	Integrated Circuit
ICCP	Impressed Current Cathodic Protection
IDE	Integrated Development Environment
IEEE	Institute of Electrical and Electronics Engineers
I/O	Input/ Output
LCD	Liquid Crystal Display
LED	Light Emitting Diode
MC	Micro controller
PC	Personal Computer
PCB	Printed Circuit Board
PDOC	Petro Dar Operating Company
PIC	Peripheral Interface Controller
PLC	Programmable Logic Controller
RAM	Random Access Memory
RMS	Remote Monitoring System

ROM	Read Only Memory
RTU	Remote Terminal Units
RXD	Receiving Data
SCADA	Supervisory Control and Data Acquisition
SPICE	Simulation Program with Integrated Circuit Emphasis
TXD	Transmit Data
U.S	United States
UART	Universal Asynchronous Receiver-Transmitter
UBBR	USART Baud Rate Register
UDR	USART Data Register
USART	Universal Synchronous Asynchronous Receiver-Transmitter
USB	Universal Serial Bus
WS	Wireless Sensor
WSN	Wireless Sensor Network

CHAPTER ONE

INTRODUCTION

CHAPTER ONE
INTRODUCTION

1.1 Preface

Corrosion is the result of an electrochemical reaction driven by a potential difference between two electrodes; an anode and a cathode, connected by an electronic path and immersed in the same electrolyte (see Figure 1.1). In the case of uniform corrosion, a multitude of microscopic anodic and cathodic sites exist on the surface of the metal structure [1].
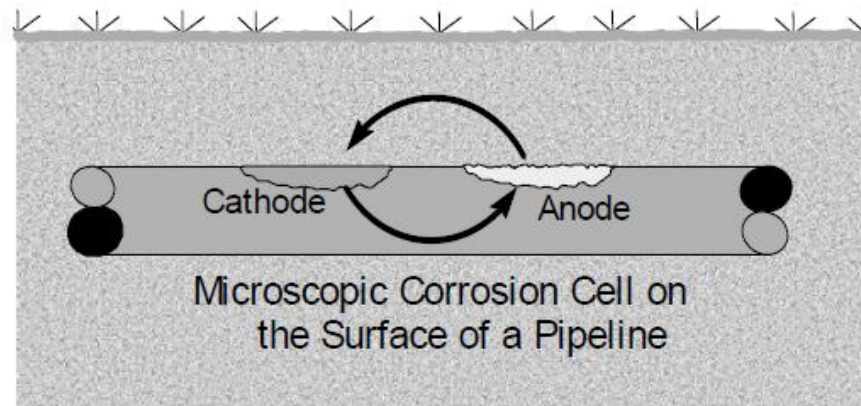
Figure 1.1: Microscopic corrosion cell

Cathodic Protection (CP) is a technique to control the corrosion of a metal surface by making it the cathode of an electrochemical cell [2]. It is most commonly used to protect steel of water or fuel pipelines, storage tanks, steel pier piles, water-based vessels including yachts and powerboats, offshore oil platforms and onshore oil well casings [3].

1.2 Problem Statement

Petro Dar Operating Company (PDOC) CP system for Central Processing Unit (CPF) Crude tanks is monitored and controlled by CP team who are located in Khartoum (PDOC did not dedicate a CP team for the CPF due to high operation

cost). Accordingly, any failure in the system will not be recognized on time but after a while (up to a month) and that could compromise the tanks structure to a severe corrosion and badly affecting all the processing facility. In addition, the measurement done by technicians is susceptible to human measurement errors. Moreover, due to harsh environment, technician may not be fully motivated to conduct a regular and reliable measurement.

So, Does PDOC CPF tanks protected all times? And if it is not, how the protection system could be improved? Could we apply the concept of remote monitoring and control on facility CP? And what are the effects on the operation perspectives? What kind of remote monitoring and automatic control could be utilized on such system?

1.3   Research Objectives

The objectives of this research is to provide an embedded low cost remote monitoring and control of PDOC CPF crude tanks CP system to protect tanks against corrosion and to save the facility against any sudden shutdown resulting from corrosion.

1.4   Methodology

To achieve the previous objective, the following methodology can be presented in Figure 1.2.

1.5   Thesis Layout

Chapter two contains an introduction about CP concepts and techniques in addition to presenting of related literature which addresses solutions to monitor CP system remotely. In chapter three, the CP monitoring system design and implementation has been discussed in details and given a brief description of each device used and the simulation counterpart for it. In chapter four, the results have been reviewed and discussed in details and compared the simulation to the

implementation circuit. In chapter five, conclusion of work done in this research and recommendations for the results and future improvements.
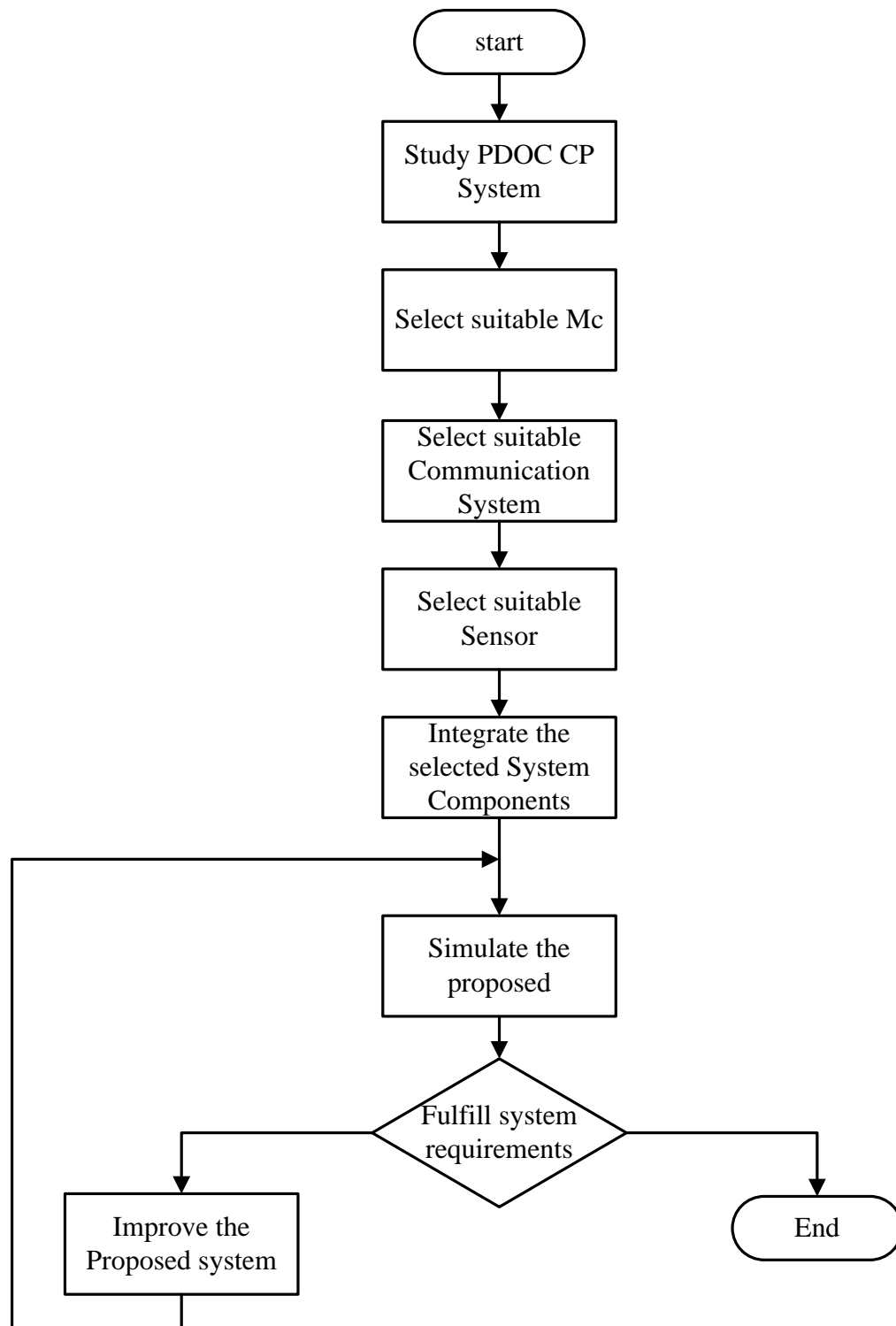
Figure 1.2: CP monitoring system methodology

CHAPTER TWO

CATHODIC PROTECTION TECHNIQUES

CHAPTER TWO
CATHODIC PROTECTION TECHNIQUES

2.1 Introduction

In this chapter, a brief introduction about corrosion, its type and how it can be mitigated were given. In addition, CP definition and concept as well as its protection techniques were discussed.

2.2 Corrosion

Corrosion is one of the most important problems encountered by the owners and operators of underground, offshore, submerged and other metallic structures exposed to an electrolyte. If corrosion is not controlled, it can lead to large costs in repairs or facility replacement. Even greater costs can be incurred from environmental damage, injuries and fatalities. Corrosion is defined as the deterioration of a substance or its properties as a result of an undesirable reaction with the environment. This includes metals, plastics, wood, concrete, and virtually all other materials. Current estimates show that the cost of corrosion in the U.S. exceeds $500 billion per year amounting to more than $2,000 per person [1].

There are several forms of corrosion, such as erosion, fretting, nuclear, high temperature, and electrochemical. The most common form of corrosion that you will encounter is electrochemical corrosion. On a corroding surface, there are hundreds of local, or microscopic, corrosion cells. Figure 2.1 illustrates such a cell on the surface of a pipeline. There is a potential, or voltage, difference between the anode and cathodes of these cells; this potential difference drives the corrosion current [1].
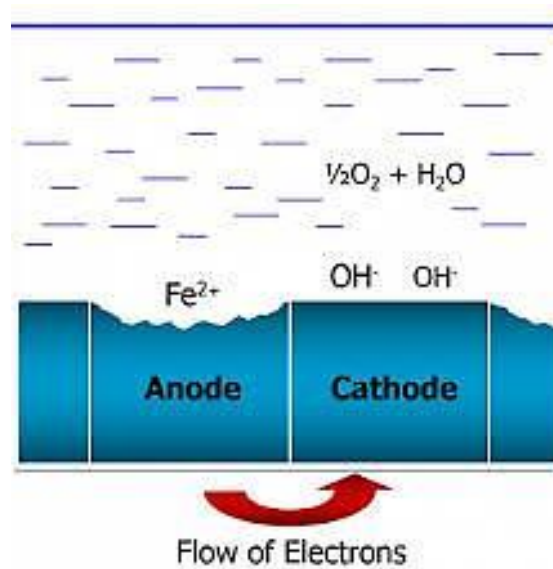
Figure 2.1: Microscopic corrosion cell

To mitigate corrosion, all existing cathode sites must be electro negatively polarized to a potential equal to the open circuit potential of the most active anode potential existing on the structure. Polarization of a structure is accomplished by applying external current, the magnitude of which depends on the cathodic polarization behavior. Figure 2.2 represents the polarization of the cathode to the open circuit potential of the anode of a corrosion cell on a structure using CP [2].
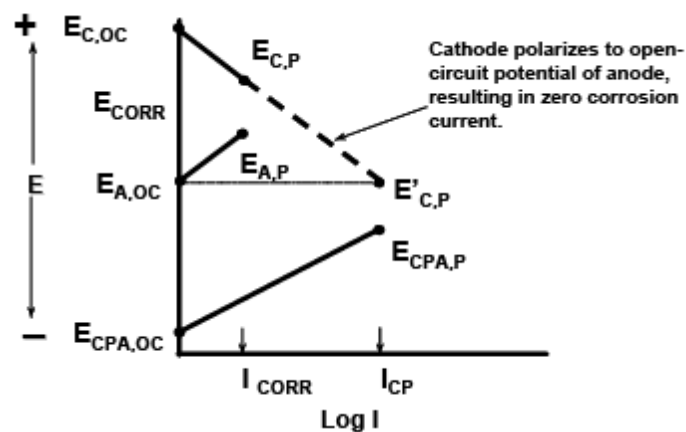
Figure 2.2: Cathode polarized to the open circuit potential of the anode

2.3 Cathodic Protection

The concept of CP involves reducing the potential difference between the local anodic and cathodic sites to zero, resulting in zero corrosion current flow. This can be accomplished by impressing current onto the structure from an external electrode and polarizing the cathodic sites in an electronegative direction. As the potentials of the cathodic sites polarize toward the potentials of the anodic sites, corrosion current is reduced. When the potentials of all cathodic sites reach the open circuit potential of the most active anodic sites, corrosion is eliminated on the structure. The structure is now the cathode of an intentional macroscopic corrosion cell. Corrosion of the metal will cease once the applied CP current equals or exceeds the corrosion current [2].

There are two main types of CP systems:

a) Sacrificial anode CP (also called galvanic CP) consists of magnesium or zinc anodes connected to the metal tank or piping (see Figure 2.3a). With this arrangement, the anodes will corrode over time instead of the tank or piping.

b) Impressed Current CP (ICCP) also uses anodes to prevent tanks or piping from corroding by applying a weak electrical current to strengthen the protective effect (see Figure 2.3b). A rectifier converts Alternating Current (AC) from a power source to Direct Current (DC) and sends it to the anodes. ICCP is typically used at larger sites, and sites with soil conditions where more current is needed than the anodes themselves can produce.

CP is often used in conjunction with other corrosion control methods including protective coatings and electrical isolation. It does not actually eliminate corrosion. Instead, it transfers it from the structure to be protected to the Cathodic Protection anode(s). The structure is now the cathode of an intentional corrosion cell. The simplest method to apply CP is by connecting

the metal to be protected with another more easily corroded metal to act as the anode of the electrochemical cell [1].
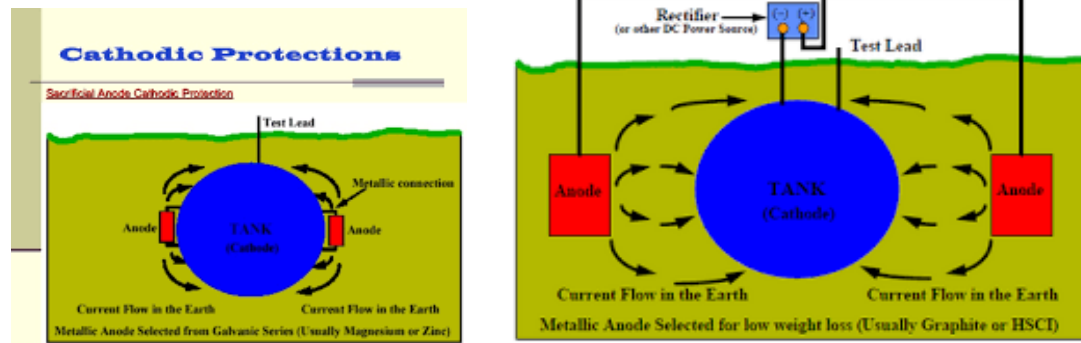
(a) CP with sacrificial anode   (b) Impressed Current CP (ICCP)

Figure 2.3: Two types of CP system

To determine that a CP system is operating properly and protecting a structure from corrosion, the following data are recorded on a routine basis:

• Structure-to-electrolyte potential.

• Rectifier voltage and current outputs.

• Current output of galvanic anodes.

• Magnitude and direction of current through mitigation bonds.

• Resistance of ground beds.

• Integrity of rectifiers, isolating joints, electrical bonds, and other physical features associated with the corrosion control system.

Measurement of structure-to-electrolyte potentials is the only method of determining when adequate CP is achieved [1].

The most obvious reason to monitor CP is to make sure corrosion is under control. When a structure corrodes, leaks may occur, product may be lost, and structural damage may occur. There is also concern over public safety and environmental damage. For this reason, regulations have been enacted in many industries and countries to make sure structures containing hazardous products are adequately protected to reduce the risk to the public and the environment [1].

9

Researchers have tried lately to apply remote monitoring in many aspects of life to ease and decrease the cost of monitoring such as in security, tracking vehicles, control as well as CP. Following are some researches tried to apply remote monitoring concepts on CP.

2.4 CP Related Literature

In [4], a remote monitoring for CP system for gas pipeline accompanied with modeling tool to predict the situation of the pipeline was designed. The remote monitoring is based on cellular phones, antennas and Global Positioning System (GPS) devices which are used to assess readings on remote sensors. A series of sensors mounted in monitoring points at different locations in the field collects soil to pipe potential readings. Also the status of the anode beds and rectifiers are collected and transmitted wirelessly to the network. The information is collected by servers across the network where the data is processed. The process involves updating a database which contains the historical data, as well as comparing the measured values against results coming from the simulation software. Finally, a series of reports on the status of the system become available in the network, so that any client terminal can access it; and in case of failures, or anomalies different warning signals are sent to mobile devices.

In [5], researcher presented study and simulation for connecting CP Remote Monitoring System (RMS) with Supervisory Control and Data Acquisition (SCADA) system. Researcher simulated CP RMS controller with Rockwell Automation software package and the virtual Global System for Mobile (GSM) modem with windows internal communication service and SCADA software using Wonder ware™ - Intouch™. The software design demonstration revealed that the possibility of imitating and developing the home made version of the RMS program satisfied with using the existing available of small- scale Programmable

Logic Controller (PLC's), communication modems, SCADA software and database management applications.

In [6], cathodic protection monitoring system accomplished in this paper demonstrates a solution for protecting long distance transportation pipeline for a long-term. This system integrates three kinds of networks, Wireless Sensor Network (WSN): monitors pipeline's electric potential information, the General Packet Radio Service (GPRS) network: responsible for the long-distance transmission, Internet: in charge of issuing and analyzing monitoring information. Practical running data of this system optimized with reliable transmission protocol and energy saving mechanism indicate monitoring system can meet requirement of tracking status of cathodic protection system in station.

In [7], the proposed system integrates the technology of WSN in order to collect potential data and to realize remote data transmission. Three cases had been studied; the normal situation when there is not find any problem in oil pipeline, in the event of a malfunction in the values of voltage by a simple increase or decrease and in the event of a serious defect in the pipe and be output either smuggling operation or malfunction in the anode. Each one of the three studied cases uses nine WS's (MDA100CB mote with IRIS mote), three Remote Terminal Units (RTU's) (MIB 520 mote with IRIS mod) and one Base Station (BS) Personal Computer (PC). LabVIEW 2010 program was used as the tool to build the simulation environment. The simulation results of this technique showed that it has least time delay, high speed, low power, and the corrected location of the WS and the corrosion location in the pipeline determined. In the other side, the system is complicated and relatively high.

In [8], ICCP technology is a common method for protecting ships and bridges from corrosion. A color touch screen is connected to ARM LPC2138 on the ICCP master control panel through serial ports. Communications between ICCP systems and the ARM processor of remote terminals are realized using

modbus protocol. The touch screen is utilized for real-time online monitoring and a real-time database is built using RS485 communication interface for user's convenience. In receiving the response frames, predictions are strictly checked with the frames received to avoid any fault. The result shows the efficiency of the system.

CHAPTER THREE

DESIGN AND IMPLEMENTATION OF PROPOSED CP MONITORING SYSTEM

DESIGN AND IMPLEMENTATION OF PROPOSED CP MONITORING SYSTEM

3.1 Introduction

The CP monitoring system has been divided into three sub categories: Field Side, Wireless Communication Unit and Monitoring Side as presented in Figure 3.1. For the purpose of power supply, an external adapter with 5V/0.8-1.0A can be used at each side. In this Chapter, the components of the proposed system are briefly discussed, connected and tested to be ready for complete simulation and implementation in the next chapter. The Proteus ISIS simulation simulator is used to draw the schematic and run the simulation.
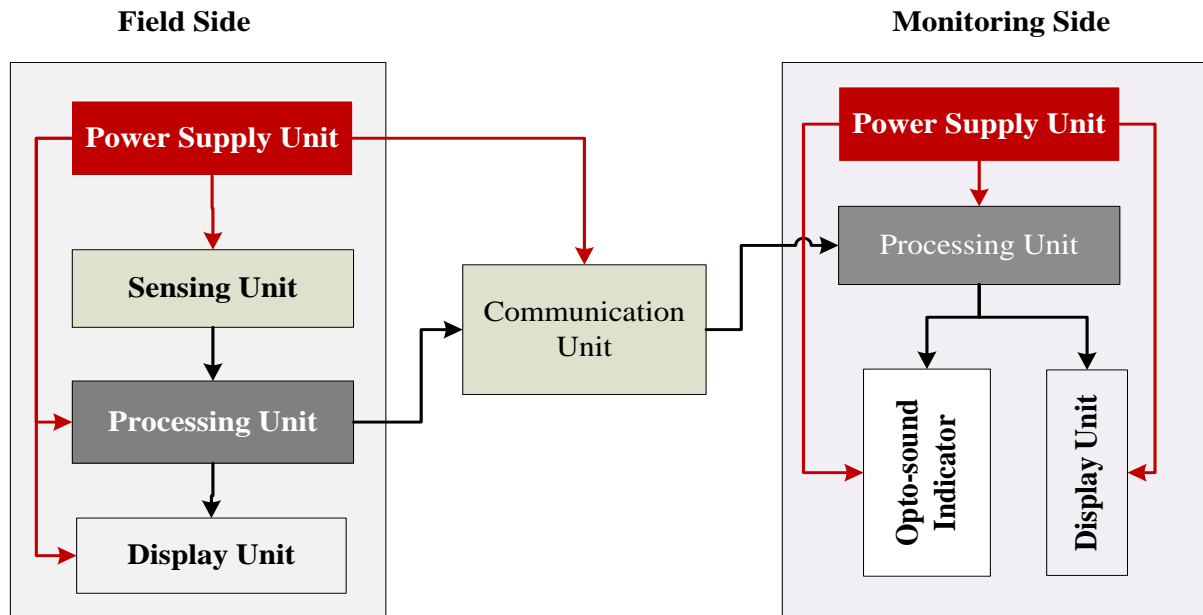
Figure 3.1: The proposed CP monitoring system Block Diagram

3.2 Field Side

The field side composes of the microcontroller, voltage sensor, power adapter and Liquid Crystal Display (LCD) as display unit. To simulate the output voltage of the sensor (i.e. reference electrode), a variable resistor (i.e. voltage divider) is used as shown in Figure 3.2.
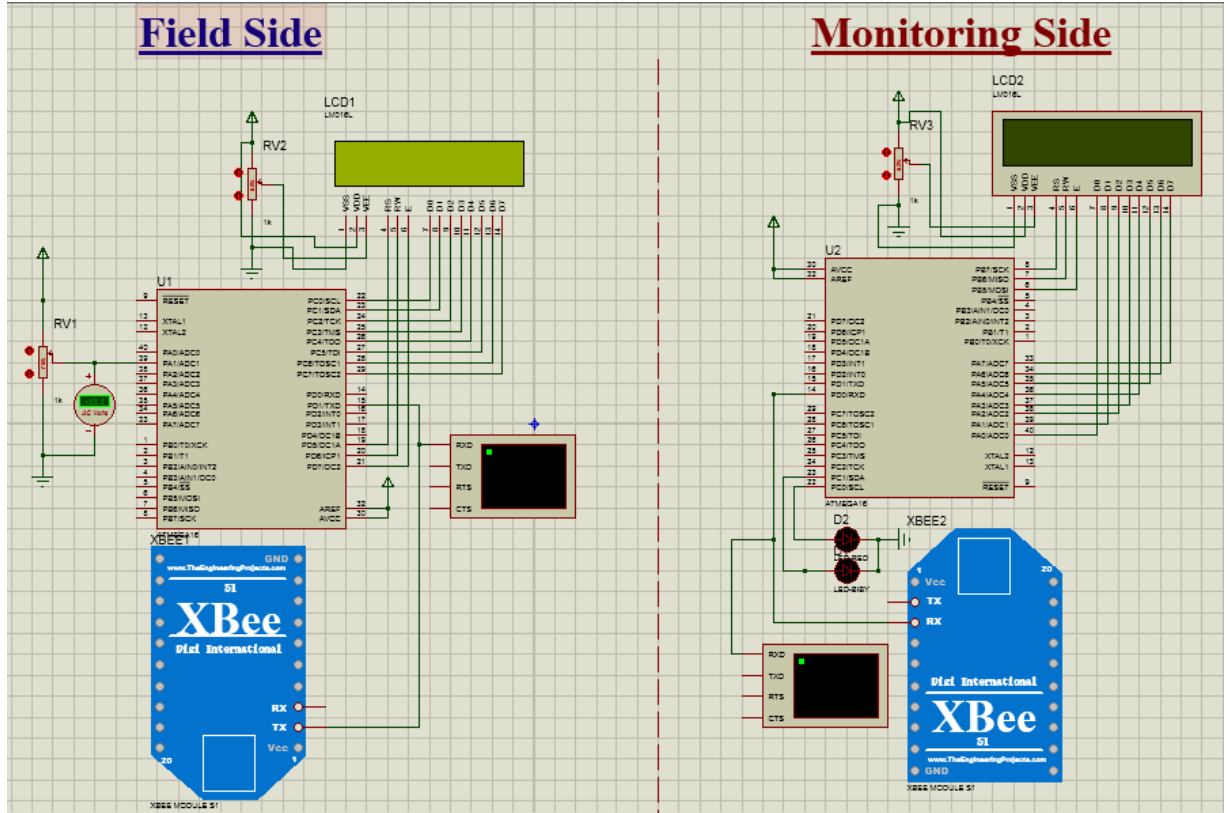
Figure 3.2: The proposed CP monitoring system schematic

3.2.1 Processing Unit (ATmega16/32):

Products using microprocessors generally fall into two categories as in Figure 3.3. The first category uses high-performance microprocessors such as the Pentium in applications where system performance is critical. In the second category of applications, performance is secondary; issues of cost, space, power, and rapid development are more critical than raw processing power. The microprocessor for this category is often called a microcontroller [10].

A microcontroller has a Central Processing Unit (CPU) (a microprocessor) in addition to a fixed amount of Random Access Memory (RAM), Read Only Memory (ROM), I/O ports, and a timer all on a single chip. In other words, the processor, RAM, ROM, I/O ports, and timer are all embedded together on one chip; therefore, the designer cannot add any external memory, I/O, or timer to it.

It is interesting to note that many microcontroller manufacturers have gone as far as integrating an Analog-to-Digital Converter (ADC) and other peripherals into the microcontroller [10].

(a) General purpose microprocessor system   (b) Microcontroller
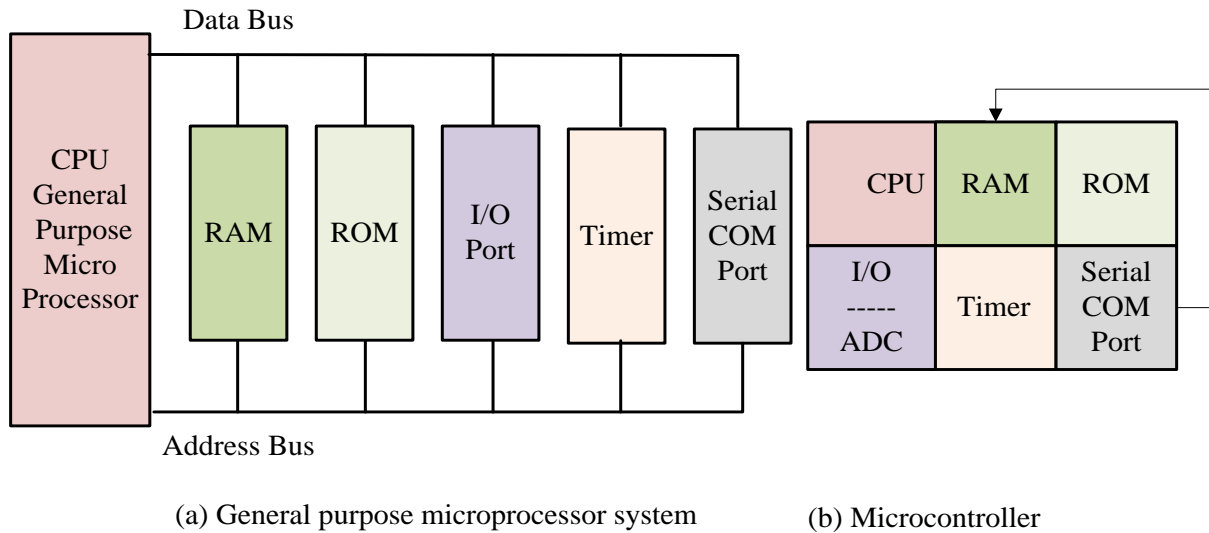
Figure 3.3: Microprocessor system contrasted with microcontroller system

There are five major 8-bit microcontrollers' manufacturers. They are: Freescale Semiconductor's (formerly Motorola) 68HC08/68HC11, Intel's 8051, Atmel's AVR, Zilog's Z8, and Peripheral Interface Controller (PIC) from Microchip Technology. Each of the above microcontrollers has a unique instruction set and register set; therefore, they are not compatible with each other. There are three criteria in choosing microcontrollers. The first one is meeting the computing needs of the task at hand efficiently and cost effectively. Secondly, availability of software and hardware development tools such as compilers, assemblers, debuggers, and emulators; and thirdly, wide availability and reliable sources of the microcontroller [10]. Table 3.1 shows a comparison between selected families.

The basic architecture of AVR was designed by two students of Norwegian Institute of Technology, Alf-Egil Bogen and Vegard Wollan, and then was bought

and developed by Atmel in 1996. There are many kinds of AVR microcontroller with different properties. Except for AVR32, which is a 32-bit microcontroller, AVRs are all 8-bit microprocessors, meaning that the CPU can work on only 8 bits of data at a time. Data larger than 8 bits has to be broken into 8-bit pieces to be processed by the CPU. AVRs are generally classified into four broad groups: Mega, Tiny, special purpose, and classic. The AVR microcontroller from Atmel is one of the most widely used 8-bit microcontrollers in the world [10].

Table 3.1: Comparison of 8051, PIC18 family, and AVR (40-pin package)

Feature	8052	PIC18F452	ATmega16/32
Program ROM	8.0K	32.0K	32.0K
Data RAM	256 bytes	2.0K	2.0K
EEPROM	0bytes	256 bytes	1.0K
Timers	3	4	3
I/O pins	32	35	32

Mega AVR (ATmegaxxxx) are powerful microcontrollers with have more than 120 instructions and lots of different peripheral capabilities, which can be used in different designs. Some of their characteristics, which presented in Table 3.2, are as follows:

- Program memory: 4K to 256K bytes.
- Package: 28 to 100 pins (see Figure 3.4).
- Extensive peripheral set.
- Extended instruction set: They have rich instruction sets.
- ADCs are among the most widely used devices for data acquisition. ADC converter is used to translate the analog signals to digital numbers so that the microcontroller can read and process them.

- The ADC has n-bit resolution; where n can be 8, 10, 12, 16, or even 24 bits. Higher-resolution ADCs provide a smaller step size, where step size is the smallest change that can be discerned by an ADC.

Table 3.2: Features of some members of the ATmega family

Part No.	Code ROM	Data RAM	Data EEPROM	I/O Pins	ADC	Timers	Pin No. and Package
ATmega8	8K	IK	0.5 K	23	8	3	TQFP32, PDIP28
ATmegal6	16K	IK	0.5 K	32	8	3	TQFP44 PDIP40
ATmega16/32	32K	2K	IK	32	8	3	TQFP44, PDIP40
ATmega64	64K	4K	2K	54	8	4	TQFP64, MLF64
ATmegal280	128K	8K	4K	86	16	6	TQFP100, CBGA

```
(XCK/T0) PB0   1         40   PA0 (ADC0)
   (T1) PB1    2         39   PA1 (ADC1)
(INT2/AIN0) PB2 3        38   PA2 (ADC2)
(OC0/AIN1) PB3  4 MEGA32 37   PA3 (ADC3)
    (SS) PB4   5         36   PA4 (ADC4)
  (MOSI) PB5   6         35   PA5 (ADC5)
  (MISO) PB6   7         34   PA6 (ADC6)
   (SCK) PB7   8         33   PA7 (ADC7)
       RESET   9         32   AREF
         VCC   10        31   AGND
         GND   11        30   AVCC
       XTAL2   12        29   PC7 (TOSC2)
       XTAL1   13        28   PC6 (TOSC1)
   (RXD) PD0   14        27   PC5 (TDI)
   (TXD) PD1   15        26   PC4 (TDO)
  (INT0) PD2   16        25   PC3 (TMS)
  (INT1) PD3   17        24   PC2 (TCK)
 (OC1B) PD4    18        23   PC1 (SDA)
 (OC1A) PD5    19        22   PC0 (SCL)
   (ICP) PD6   20        21   PD7 (OC2)
```

Figure 3.4: ATmega16/32 pin diagram

Table 3.3: Resolution versus step size for ADC (Vref = 5 V).

n-bit	Number of Steps	Step Size (mV)
8	256	5/256 = 19.53

10	1024	5/1024 = 4.88
12	4096	5/4096 = 1.2
16	65,536	5/65,536 = 0.076

Notes: Vee= 5.0VDC, and Step size (resolution) is the smallest change that can be discerned by an ADC.

In an 8-bit ADC we have an 8-bit digital data output of DO-D7, while in the 10-bit ADC the data output is DO-D9. To calculate the output voltage, the following formula:

$$D_{out} = \frac{V_{in}}{step\ size} \tag{3.1}$$

Where D_{out} is the digital data output (in decimal), V_{in} is the analog input voltage, and step size (resolution) is the smallest change, which is, for example, Vref/256 for an 8-bit ADC. Figure 3.5 shows the block diagram of an 8- bit ADC.
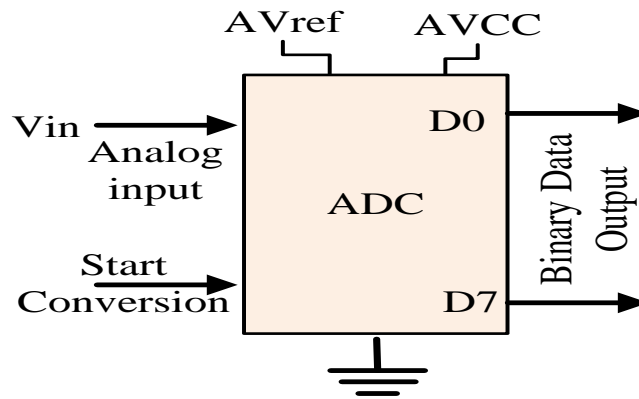
Figure 3.5: An 8-bit ADC block diagram

The ADC peripheral of the ATmega16/32 has the following characteristics:
(a) It is a 10-bit ADC.
(b) It has 8 analog input channels, 7 differential input channels, and 2 differential input channels with optional gain of 10x and 200x.

(c) The converted output binary data is held by two special function registers called ADCL (A/D Result Low) and ADCH (A/D Result High).

To program the ADC of the AVR, the following steps must be taken:

1. Make the pin for the selected ADC channel an input pin.

2. Turn on the ADC module of the AVR because it is disabled upon power-on reset to save power.

3. Select the conversion speed. The registers ADPS2:0 is used to select the conversion speed.

4. Select voltage reference and ADC input channels. The REFS0 and REFS1 bits in the ADMUX register can be used to select voltage reference and the MUX4:0 bits in ADMUX to select the ADC input channel.

5. Activate the start conversion bit by writing a one to the ADSC bit of ADCSRA.

6. Wait for the conversion to be completed by polling the ADIF bit in the ADCSRA register.

7. After the ADIF bit has gone HIGH, read the ADCL and ADCH registers to get the digital data output. Notice that you have to read ADCL before ADCH; otherwise, the result will not be valid.

8. If you want to read the selected channel again, go back to step 5.

9. If you want to select another Vref source or input channel, go back to step 4 [10].

The output of the reference electrode has been connected to ATmega16/32 via PA1/ADC1 port (pin #39). For simulation and implementation purposes, the ADC power ports AREF and AVCC are connected to VCC. In addition, the RESET port is connected to VCC through resistor (10.0K Ohm) to limit the power supply current. The external frequency crystals XTAL1 and XTAL2 have been left not connected and the internal crystal is used instead as can be shown in Figure 3.1.

3.2.2 Sensing Unit

Reference electrodes, or half-cells, are important devices that permit measuring the potential of a metal surface exposed to an electrolyte. A structure-to-electrolyte potential is actually the potential difference between the structure and a reference electrode. The electrolyte itself has no potential value against which the potential of a structure can be measured independently of the potential of the reference electrode used [1].

A copper-copper sulfate reference electrode has been used in this thesis to measure the CP protection voltage. Copper Sulfate Electrodes (CSE) are the most commonly used reference electrode for measuring potentials of underground structures and also for those exposed to fresh water. It is not suitable for use in a chloride electrolyte as the chloride ions will migrate through the porous plug and contaminate the CSE. The electrode is composed of a copper rod, immersed in a saturated solution of copper sulfate, held in a non-conducting cylinder with a porous plug at the bottom, as shown in Figure 3.6. The copper ions in the saturated solution prevent corrosion of the copper rod and stabilize the reference electrode [1]. For simulation and implementation purposes, a potentiometer (voltage divider) is used as sensing unit. Moreover, an Ampere Volt Ohm (AVO) meter is used to read the potential at the analog input of the microcontroller as in Figure 3.1.
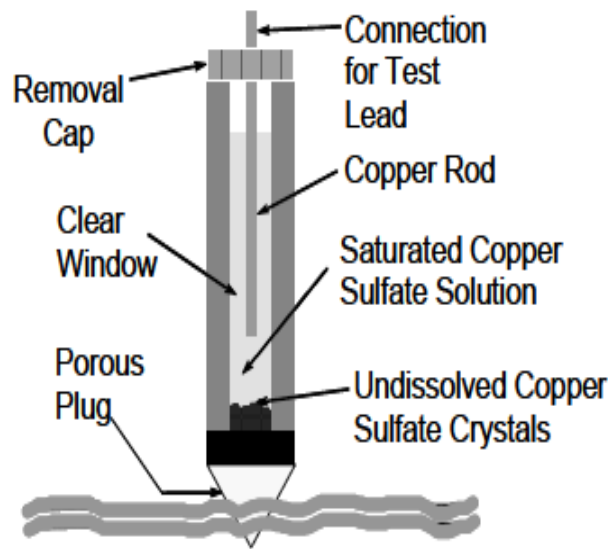
Figure 3.6: Copper- copper sulfate reference electrode in contact with Earth

3.2.3 Display unit

To display the extracted electrode voltage in milli-volt (mv), an LCD unit is used. In recent years the LCD is finding widespread use replacing LEDs (seven segment LEDs or other multi-segment LEDs). This is due to the following reasons:

1. The declining prices of LCDs.

2. The ability to display numbers, characters, and graphics. This is in contrast to LEDs, which are limited to numbers and a few characters.

3. Incorporation of a refreshing controller into the LCD, thereby relieving the CPU of the task of refreshing the LCD. In contrast, the LED must be refreshed by the CPU (or in some other way) to keep displaying the data.

4. Ease of programming for characters and graphics.

The LCD used in this thesis has 14 pins. The function of each pin is given in Table 3.5.

Figure 3.7 shows the block diagram of LCD connection with microcontroller, where Table 3.5 presents the related LCD command code.

Table 3.4: Pin descriptions for LCD

Pin	Symbol	I/O	Description
1	Vss	-	Ground
2	Vcc	-	+5 V power supply
3	VEE	-	Power supply to control contrast
4	RS	I	RS = 0 to select command register, RS = 1 to select data register
5	R/W	I	R/W = 0 for write, R/W = 1 for read
6	E	I/O	Enable
7	DB0	I/O	The 8-bit data bus
8	DB1	I/O	The 8-bit data bus
9	DB2	I/O	The 8-bit data bus
10	DB3	I/O	The 8-bit data bus
11	DB4	I/O	The 8-bit data bus
12	DB5	I/O	The 8-bit data bus
13	DB6	I/O	The 8-bit data bus
14	DB7	I/O	The 8-bit data bus

To send data and commands to LCD you should do the following steps. Notice that steps 2 and 3 can be repeated many times:

1. Initialize the LCD.
2. Send any of the commands from Table 3.5 to LCD.
3. Send the character to be shown on LCD.

There are two ways to interface an LCD to AVR; an 8-bit data or 4-bit data options can be used. The 8-bit data interfacing is easier to program but uses 4 more pins. In this thesis, an 8-bit data option is used.
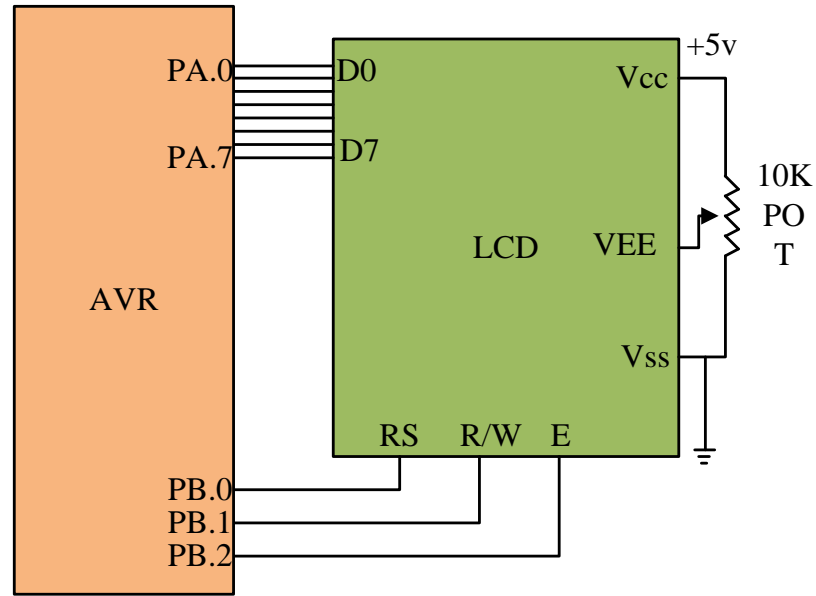
Figure 3.7: LCD connections for 8-bit data

LCD control signal i.e. RS, RW, and E is connected to the controller through PD5/OC1A, PD6/ICP1 and PD7/OC2 ports respectively. LCD data ports (D0 to D7) are connected to Atemega32 through port C. LCD power ports VEE is connected to a variable resistor connected to the adapter, VDD is connected to adapter and VSS is connected to the ground (for simulation purpose, adapter is being simulated by VCC (i.e. 5VDC).

Table 3.5: LCD command codes

Code (Hex)	Command to LCD Instruction Register
1	Clear display screen
2	Return home
4	Decrement cursor (shift cursor to left)
6	Increment cursor (shift cursor to right)
5	Shift display right
7	Shift display left

8	Display off, cursor off
A	Display off, cursor on
C	Display on, cursor off
E	Display on, cursor blinking
F	Display on, cursor blinking
10	Shift cursor position to left
14	Shift cursor position to right
18	Shift the entire display to the left
1C	Shift the entire display to the right
80	Force cursor to beginning of 1st line
C0	Force cursor to beginning of 2nd line
28	2 lines and 5 x 7 matrix (D4-D7, 4-bit)
38	2 lines and 5 x 7 matrix (D0-D7, 8-bit)

After ADC process completion, the 8 digits will be sent to LCD for local display in the field side. The digital value will be prepared to be sent serially digit by digit through the communication medium to the far monitoring and control room as illustrated in Figure 3.1. Moreover, the extracted analog electrode value which is displayed at LCD can be transmitted through the wireless communication unit to the monitoring side.

3.3 Communication Unit (Serial Transmission)

Computers transfer data in two ways: parallel or serial. In parallel data transfer, often eight or more lines (wire conductors) are used to transfer data to a device that is only a few feet away. Although a lot of data can be transferred in a short amount of time by using many wires in parallel, the distance cannot be great. To transfer to a device located many meters away, the serial method is used. In serial communication, the data is sent one bit at a time, in contrast to parallel

communication, in which the data is sent a byte or more at a time. The AVR has serial communication capability built into it. Figure 3.8 diagrams serial versus parallel data transfers.

Parallel Transfer

Serial Transfer

| Sender | → | Reciever |

| Sender | D0 ⇒ ⇒ ⇒ ⇒ ⇒ ⇒ ⇒ ⇒ D7 | Reviever |
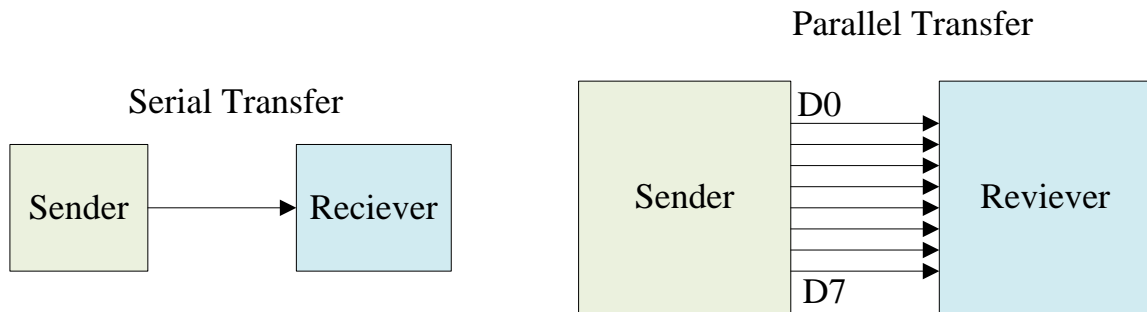
Figure 3.8: Serial versus parallel data transfer

For serial data communication to work the byte of data must be converted to serial bits using a parallel-in-serial-out shift register; then it can be transmitted over a single data line. This also means that at the receiving end there must be a serial-in-parallel-out shift register to receive the serial data and pack them into a byte. Serial data communication uses two methods, synchronous and asynchronous. The synchronous method transfers a block of data (characters) at a time, whereas the asynchronous method transfers a single byte at a time. Special Integrated Circuit (IC) chips are made by many manufacturers for serial data communications. These chips are commonly referred to as Universal Asynchronous Receiver-Transmitter (UART) and Universal Synchronous Asynchronous Receiver-Transmitter (USART).

The AVR chip has a built-in USART. The rate of data transfer in serial data communication is stated in bits per second (bps). Another widely used terminology for bps is baud rate. The baud rate is a modem terminology and is defined as the number of signal changes per second. The data transfer rate of a given computer system depends on communication ports incorporated into that

system. The ATmega16/32 has two pins that are used specifically for transferring and receiving data serially. These two pins are called TX and RX and are part of the port D group (PD0 and PD1) of the 40-pin package. Pin 15 of the ATmega16/32 is assigned to TX and pin 14 is designated as RX.

In the AVR microcontroller, five registers are associated with the USART that we deal with. They are UDR (USART Data Register), UCSRA, UCSRB, UCSRC (USART Control Status Register), and UBRR (USART Baud Rate Register). The relation between the value loaded into UBBR and the Fosc (Frequency of oscillator connected to the XTAL1 andXTAL2 pins) is dictated by the following formula:

$$DesiredBaudRate = \frac{Fosc}{16(X+1)} \tag{3.2}$$

Where *X* is the value that is loaded into the UBRR register.

In programming the AVR to transfer character bytes serially, the following steps must be taken:

1. The UCSRB register is loaded with the value 08H, enabling the USART transmitter. The transmitter will override normal port operation for the TXD pin when enabled.
2. The UCSRC register is loaded with the value 06H, indicating asynchronous mode with 8-bit data frame, no parity, and one stop bit.
3. The UBRR is loaded with one of the values in Table 3.6 (if Fosc = 8MHz) to set the baud rate for serial data transfer.
4. The character byte to be transmitted serially is written into the UDR register.
5. Monitor the UDRE bit of the UCSRA register to make sure UDR is ready for the next byte.
6. To transmit the next character, go to step 4.

In programming the AVR to receive character bytes serially, the following steps must be taken:

1. The UCSRB register is loaded with the value 10H, enabling the USART receiver. The receiver will override normal port operation for the RXD pin when enabled.

2. The UCSRC register is loaded with the value 06H, indicating asynchronous mode with 8-bit data frame, no parity, and one stop bit.

3. The UBRR is loaded with one of the values in Table 3.6 (if Fosc = 8MHz) to set the baud rate for serial data transfer.

4. The RXC flag bit of the UCSRA register is monitored for a HIGH to see if an entire character has been received yet.

5. When RXC is raised, the UDR register has the byte. Its contents are moved into a safe place.

6. To receive the next character, go to step 5.

Table 3.6: UBRR values for various baud rates (Fosc = 8 MHz, U2X = 0)

Baud Rate	UBRR (Decimal Value)	UBRR (Hex Value)
38400	12	C
19200	25	19
9600	51	33
4800	103	67
2400	207	CF
1200	415	19F

Zigbee is an open global standard built on the IEEE 802.15.4 MAC/PHY. Zigbee defines a network layer above the 802.15.4 layers to support advanced mesh routing capabilities. The Zigbee specification is developed by a growing consortium of companies that make up the Zigbee Alliance. The Alliance is made up of over 300 members, including semiconductor, module, stack, and software developers (see Figure 3.9) [11].
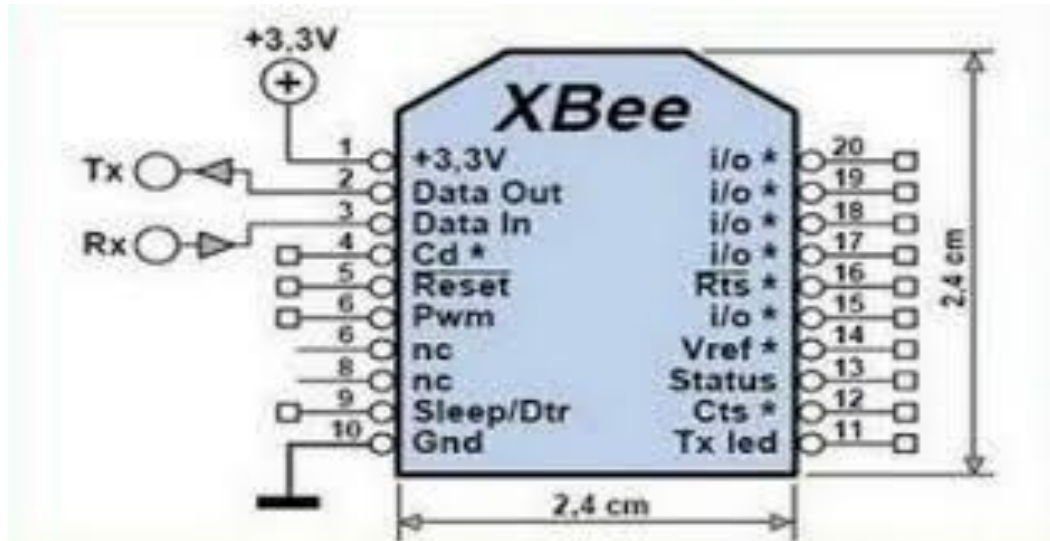
Figure 3.9: The Xbee Series 2 Module Pin diagram

XBees are hugely popular wireless transceivers for a number of reasons. They're flexible, they send and receive data over a serial port, which means they're compatible with both computers and microcontrollers (like Arduino). They are highly configurable you can have meshed networks with dozens of XBees, or just a pair swapping data. You can use them to remotely control your robot, or arrange them all over your house to monitor temperatures or lighting conditions in every room [12].

The XBee and XBee-PRO RF modules were engineered to meet IEEE 802.15.4 standards and support the unique needs of low-cost, low-power wireless sensor networks. The modules require minimal power and provide reliable delivery of data between devices. The modules operate within the ISM 2.4GHz frequency band and are pin-for-pin compatible with each other [13]. The XBee-PRO features are:

- Indoor/Urban: up to 300' (90m), 200' (60m) for international variant.
- Outdoor line-of-sight: up to 1 mile (1600m), 2500' (750m) for International variant.

- Transmit power: 63mW (18dBm), 10mW (10dBm) for international variant.
- Receiver sensitivity: 100 dBm RF data rate: 250,000bps [13].

Range figure estimates are based on free-air terrain with limited sources of interference. Actual range will vary based on transmitting power, orientation of transmitter and receiver, height of transmitting antenna, height of receiving antenna, weather conditions, interference sources in the area, and terrain between receiver and transmitter, including indoor and outdoor structures such as walls, trees, buildings, hills, and mountains [11].

3.4 Monitoring Side

This side consists of: processing unit, display unit and alarming unit as in Figure 3.1. The processing unit is to be ATmega16/32, which is connected as discussed previously. It is worthy to note that the transmitted signal from the field side is received by the RF-based wireless receiver which is connected to the microcontroller through port PC0/RXD. The received signal shall be stored in the UDR register at the ATmega16/32 where the processed data are sent to the monitoring LCD through Port A to be displayed (i.e. in milli volt). Then the value will be compared to the minimum protection voltage (850mv) and the maximum protection voltage (3,000mv) and activate the alarming unit (i.e. LED and buzzer) if the value is out of the range.

3.5 Simulation Platform

For the purpose of simulation, Proteus 8 simulator is used to simulate the circuit and ATmel studio version 6 to write, verify and generate the hex file which is executed by the simulator and the real implemented system.

3.5.1 Proteus 8 simulator

Proteus is a virtual system modeling and circuit simulation application. The suite combines mixed mode SPICE circuit simulation, animated components and

microprocessor models to facilitate co-simulation of complete microcontroller based designs (see Figure 3.10). Proteus also has the ability to simulate the interaction between software running on a microcontroller and any analog or digital electronics connected to it. It simulates I/O ports, interrupts, timers, USARTs and all other peripherals present on each supported processor.
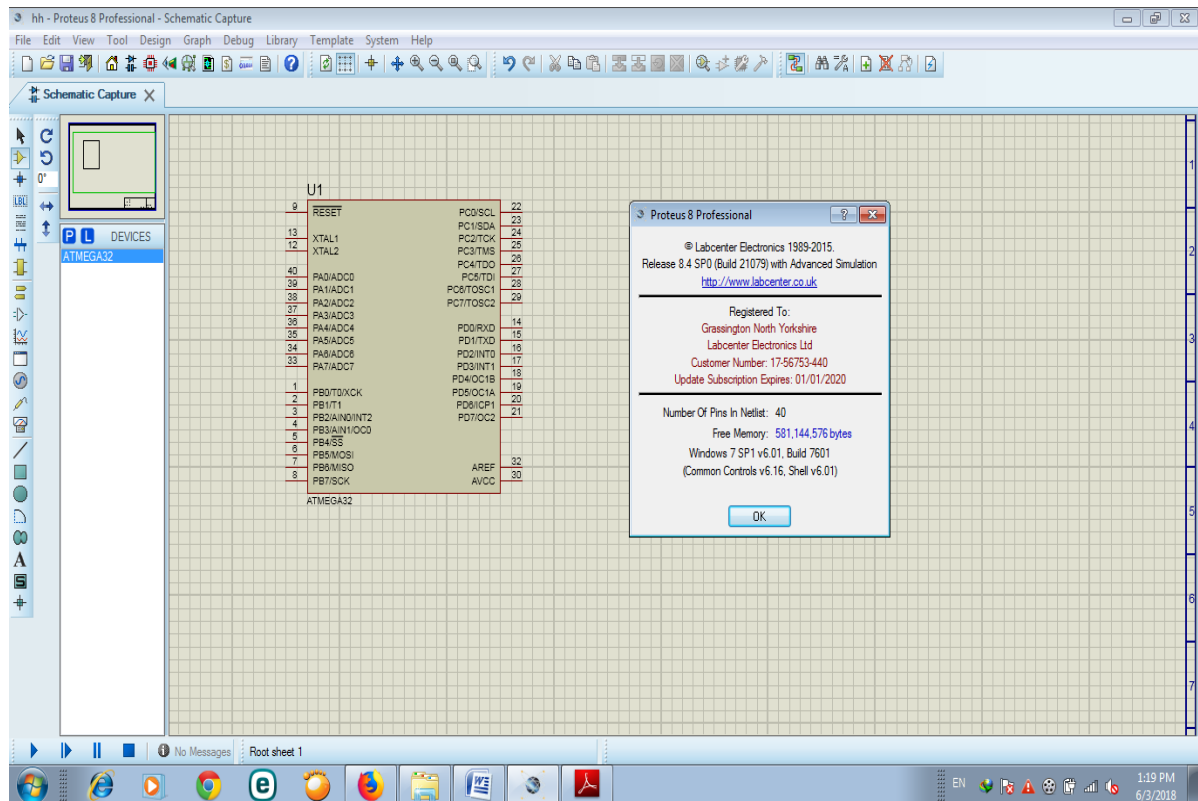
Figure 3.10: Screenshot of Proteus ISIS simulator environment

For the first time ever it is possible to draw a complete circuit for a microcontroller based system and then test it interactively, all from within the same piece of software. ISIS provides the development environment for Proteus VSM. This product combines mixed mode circuit simulation, microprocessor models and interactive component models to allow the simulation of complete microcontroller based designs. ISIS provides the means to enter the design in the first place, the architecture for real time interactive simulation and a system for managing the source and object code associated with each project. In addition, a

number of graph objects can be placed on the schematic to enable conventional time, frequency and swept variable simulation to be performed. Major features of Proteus VSM include:

1. True mixed mode simulation based on Berkeley SPICE3F5 with extensions for digital simulation and true mixed mode operation.
2. Support for both interactive and graph based simulation.
3. CPU Models available for popular microcontrollers such as the PIC and 8051 series.
4. Interactive peripheral models include LED and LCD displays, a universal matrix keypad, an RS232 terminal and a whole library of switches, pots, lamps, LEDs etc.
5. Virtual instruments include voltmeters, ammeters, a dual beam oscilloscope and a 24 channel logic analyzer.
6. On-screen graphing - the graphs are placed directly on the schematic just like any other object. Graphs can be maximized to a full screen mode for cursor based measurement and so forth.
7. Graph based analysis types include transient, frequency, noise, distortion, AC and DC sweeps and Fourier transform. An audio graph allows playback of simulated waveforms.
8. Direct support for analogue component models in SPICE format.
9. Open architecture for 'plug in' component models coded in C++ or other languages. These can be electrical, graphical or a combination of the two.
10. Digital simulator includes a BASIC-like programming language for modeling and test vector generation.
11. A design created for simulation can also be used to generate a netlist for creating a PCB; there is no need to enter the design a second time [9].

3.5.2 AVR ATmel studio 6

Atmel Studio 6 is Atmel's official Integrated Development Environment (IDE), used for writing and debugging AVR applications on the Windows platform. Atmel Studio 6 is the new professional IDE for writing and debugging AVR applications in Windows environments [10].
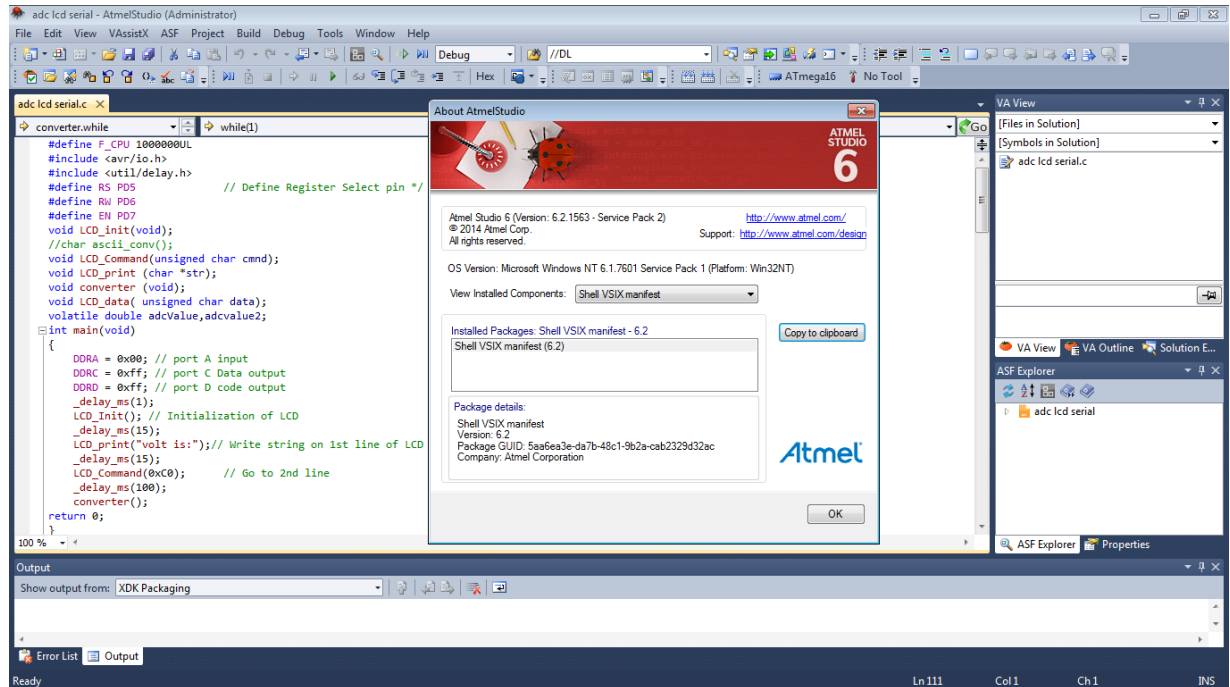
Figure 3.11: Screen shot of Atmel studio Ver.6 environment

AVR Studio is a development tool for the AVR family of microcontrollers. AVR Studio enables the user to fully control execution of programs on the AVR In-Circuit Emulator. AVR studio supports source level execution of Assembly programs assembled with the Atmel Corporation's AVR Assembler and C programs compiled with IAR systems' ICCA90 C compiler for the AVR microcontrollers.

To develop an operating code for the proposed system in its both sub-systems at filed and monitoring sides, the flow charts in Figure 3.12 and Figure 3.13 were utilized using C language in Atmel studio Ver.6 environment (See Appendix A and Appendix B). However, the developed system which its

33

schematic was presented in Figure 3.2 was implemented as in Figure 3.14 where

the simulation and implementation results were compared in the next chapter.

Figure 3.12: The flow chart of operation of field side sub-system

```
                    ┌─────────────┐
                    │    Start    │
                    └─────────────┘
                           │
                           ▼
            ┌──────────────────────────────┐
            │  Microcontroller initialization │
            │  and define Ports A,C,B as    │
            │  outpus and Port D as input   │
            └──────────────────────────────┘
                           │
                           ▼
            ┌──────────────────────────────┐
            │       LCD Initialization      │
            └──────────────────────────────┘
                           │
                           ▼
            ┌──────────────────────────────┐
            │  Print "Volt is (mV)" on  LCD │
            └──────────────────────────────┘
                           │
                           ▼
            ┌──────────────────────────────┐
            │   Set Parameters for serial   │
            │  communication and Baud rate  │
            │            for MC             │
            └──────────────────────────────┘
                           │
                           ▼
            ┌──────────────────────────────┐
            │  Activate serial communication │
            └──────────────────────────────┘
                           │
                           ▼
            ┌──────────────────────────────┐
            │   Transfer the sent value from │◄────┐
            │  UDR register to variable x[3] │     │
            └──────────────────────────────┘     │
                           │                       │
                           ▼                       │
            ┌──────────────────────────────┐     │
            │   display the sensed voltage on │     │
            │        Monitoring LCD         │     │
            └──────────────────────────────┘     │
                           │                       │
                           ▼                       │
            ┌──────────────────────────────┐     │
            │     Compare the voltage to the │     │
            │   acceptable range and activate │     │
            │  the alarming system if it is in or │     │
            │          out of ranges        │     │
            └──────────────────────────────┘     │
                           │                       │
                           └───────────────────────┘
```
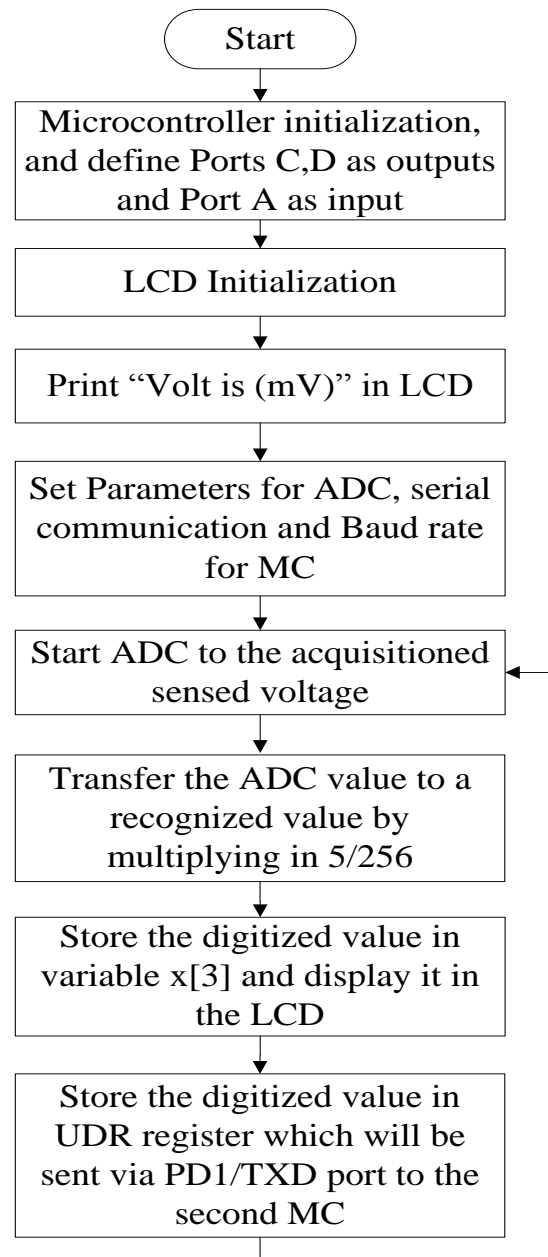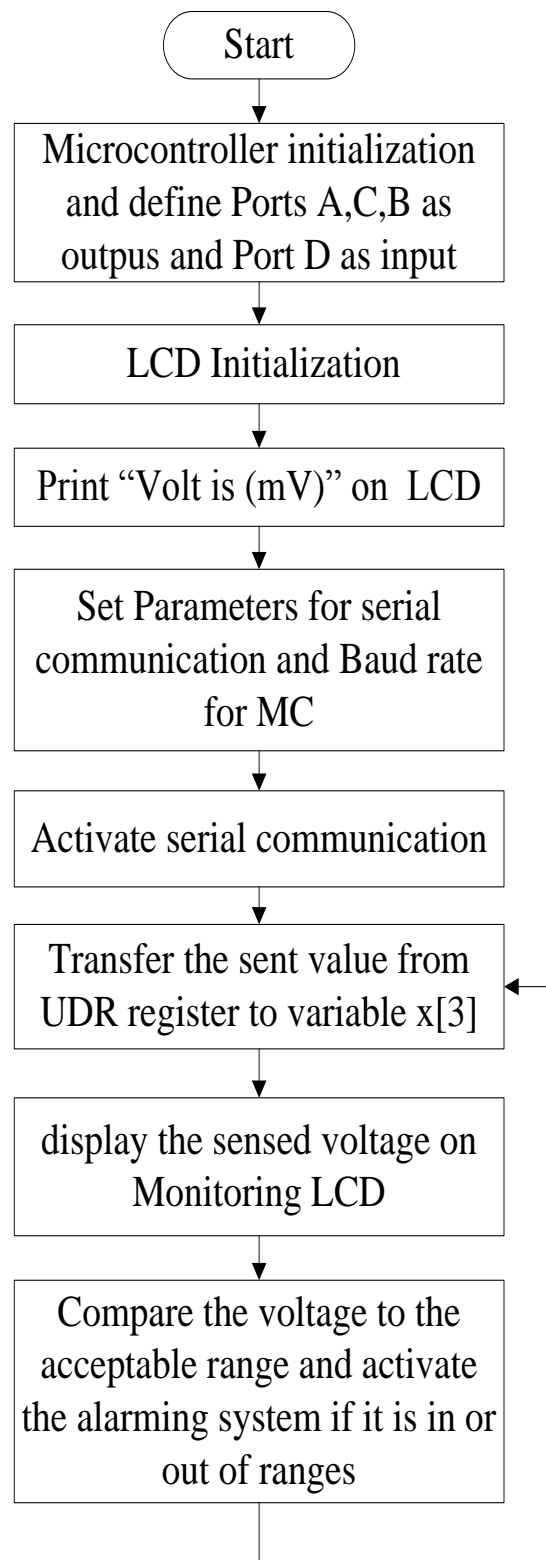
Figure 3.13: The flow chart of operation of monitoring side sub-system

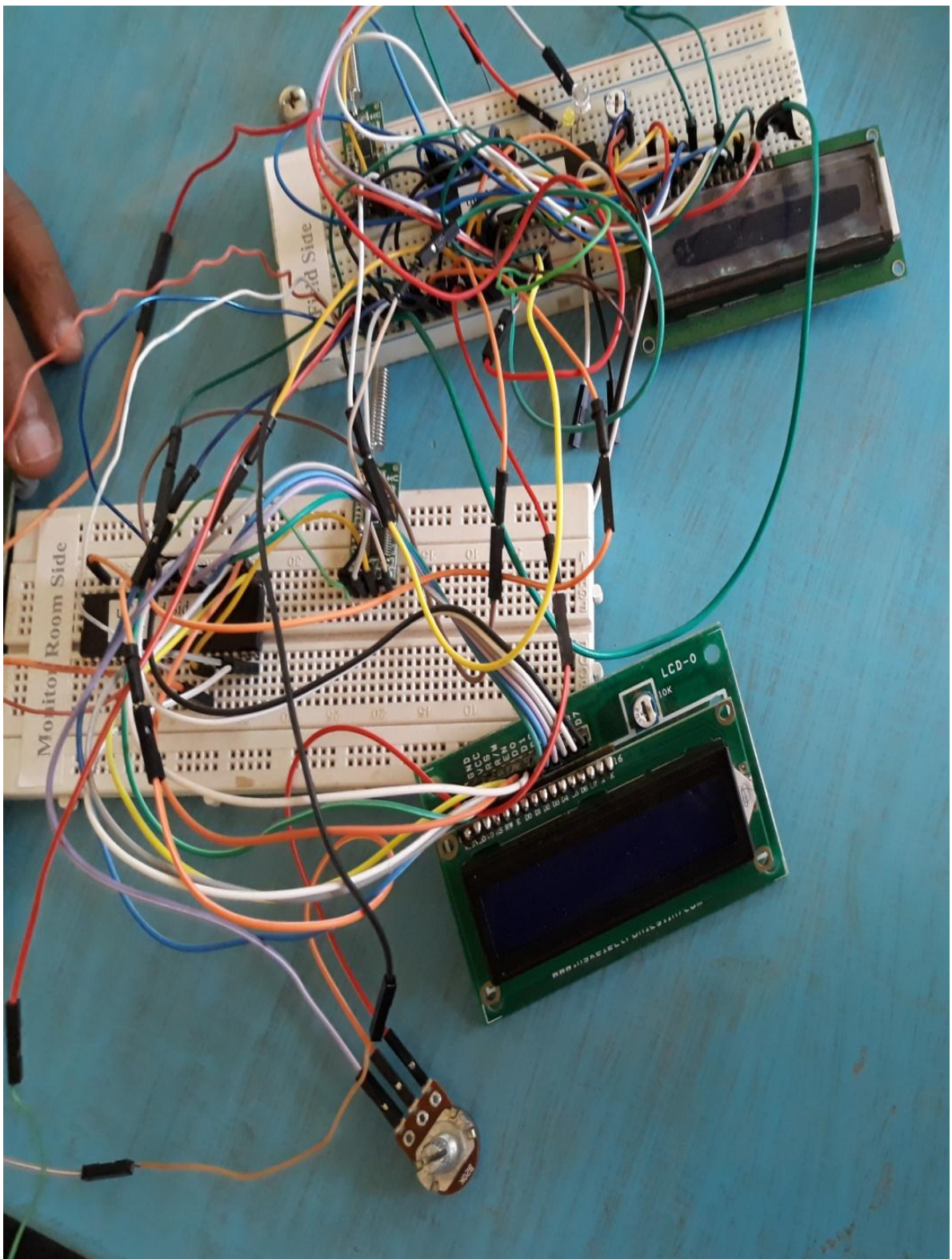Figure 3.14: The implementation of the proposed CP monitoring system

CHAPTER FOUR

RESULTS AND DISCUSSION

CHAPTER FOUR

RESULTS AND DISCUSSION

4.1 Introduction

In this chapter, the designed proposed CP sensing and monitoring system is simulated and implemented. The obtained results of the field side, wireless communication unit and monitoring side are presented where the findings are commented, discussed and recognized carefully to draw the conclusions.

4.2 Results of Field Side

To set the simulation parameters for the microcontroller at the field side, the Fosc frequency has been adjusted to 1.0 Mega Hertz to activate 1,200 Baud rate as in Figure 4.1.
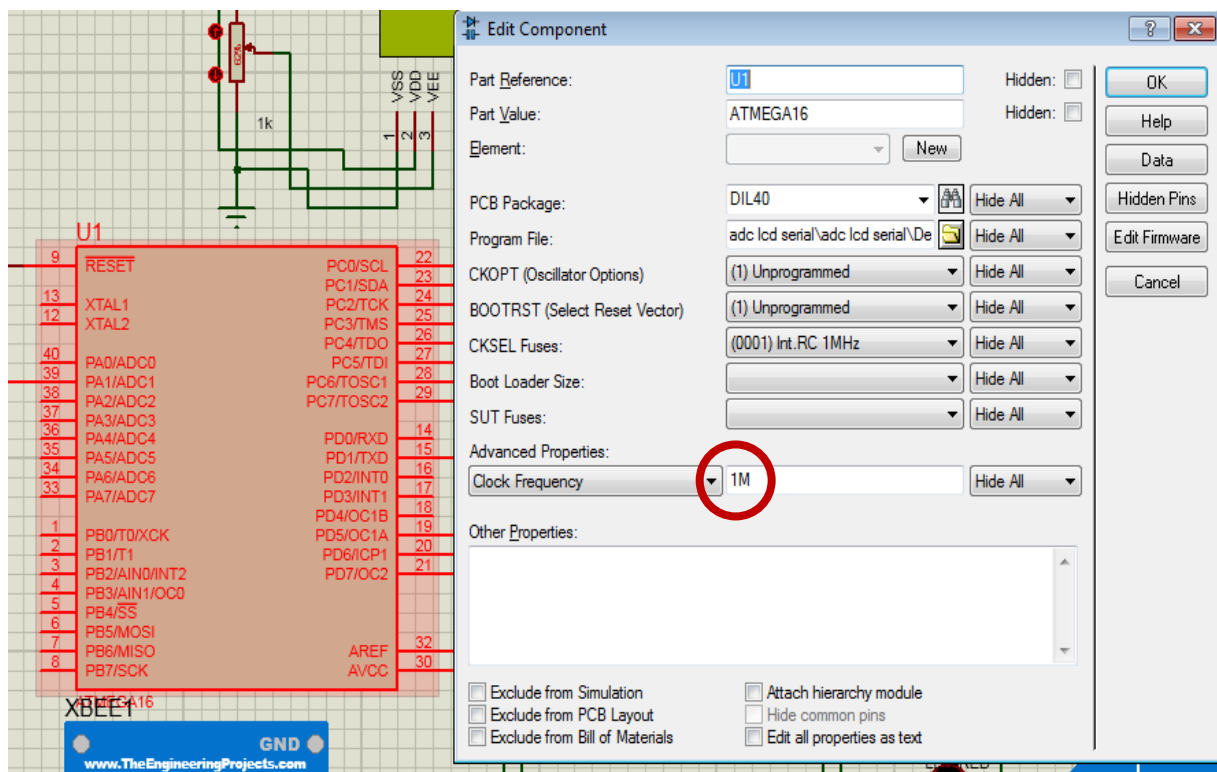

Figure 4.1: Screenshot of microcontroller Fosc set to be 1.0MHz

By running the simulation (see Figure 4.2), the potentiometer (i.e. the electrode) was adjusted to 78%, the output read in AVO meter is +3.90 (=5V*0.78=3.90V). This simulated electrode value has been received at the ADC i.e. PA1/ADC1 of the field side MC.
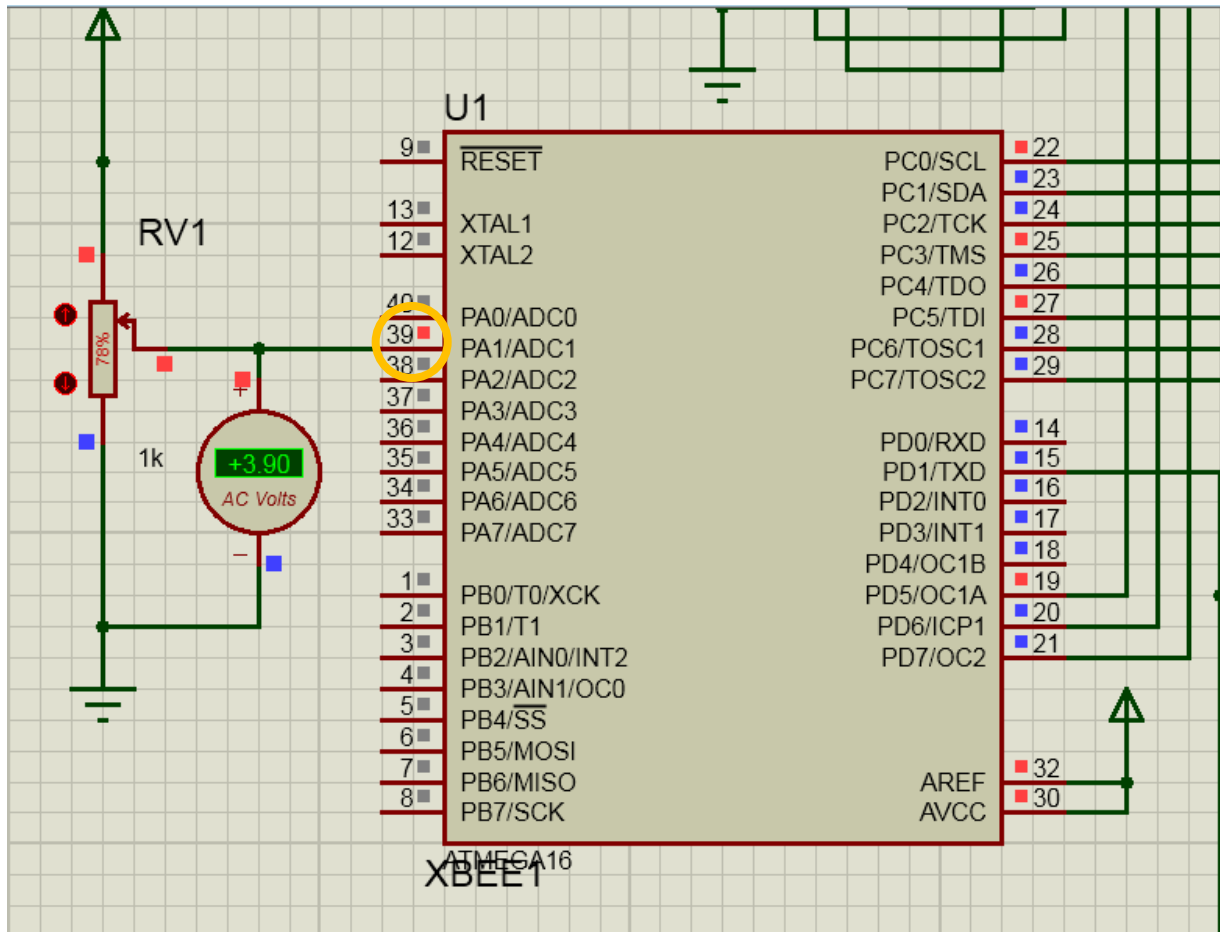

Figure 4.2: Screenshot of simulated electrode voltage using potentiometer

The electrode voltage value has been transformed to a digital form via ADC, and its digital value is sent to LCD for displaying as in Figure 4.3. As can be noticed from Figure 4.3, the digitized electrode voltage value which is 2.3V is displayed in the LCD as 2285mV where the resolution of the measured electrode value is set to be in millivolts.
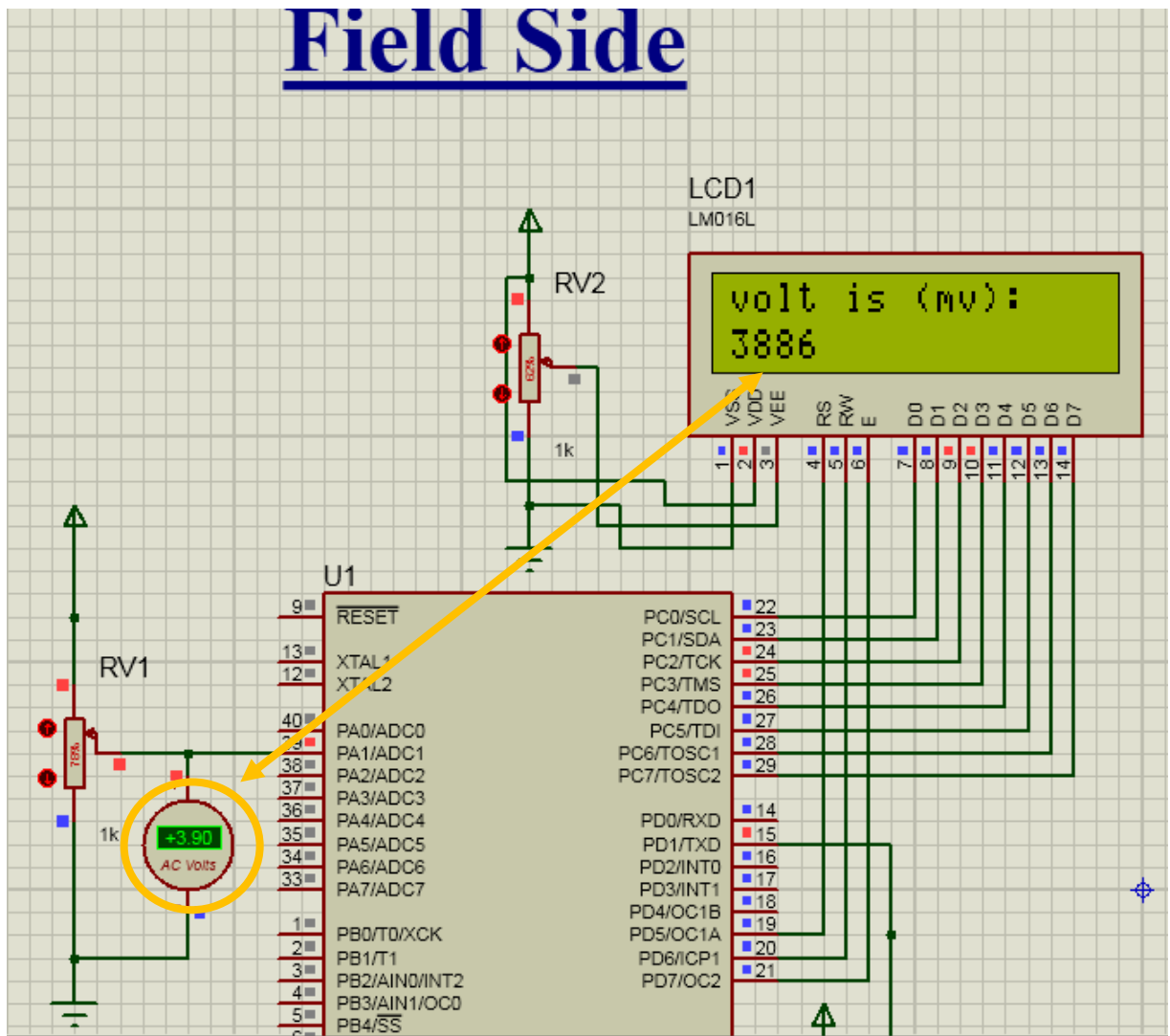
Figure 4.3: Screenshot of displaying the digitized electrode voltage on LCD at field site in milli volt

When the simulating electrode value was successfully acquisitioned, digitized and displayed in the LCD, the implementation of this part can be seem in Figures 4.4 and 4.5 respectively. In Figure 4.4, a potentiometer (10.0K Ohm) can simulate easily the electrode value which is digitized by the microcontroller ADC. This digitized value was displayed by the 16x2 LCD in mV as can be seen in Figure 4.5. It is worthy to note that the acquisitioned value and the displayed one are coincided.
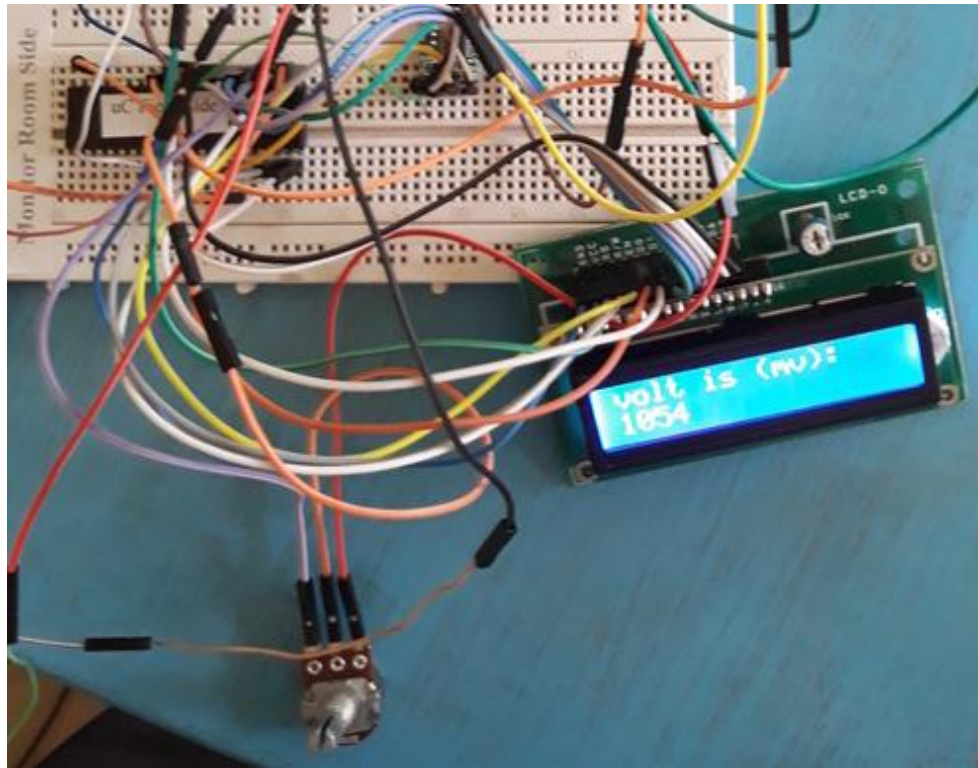
Figure 4.4: Screenshot of implemented potentiometer (electrode) output value which is received by the ADC at MC
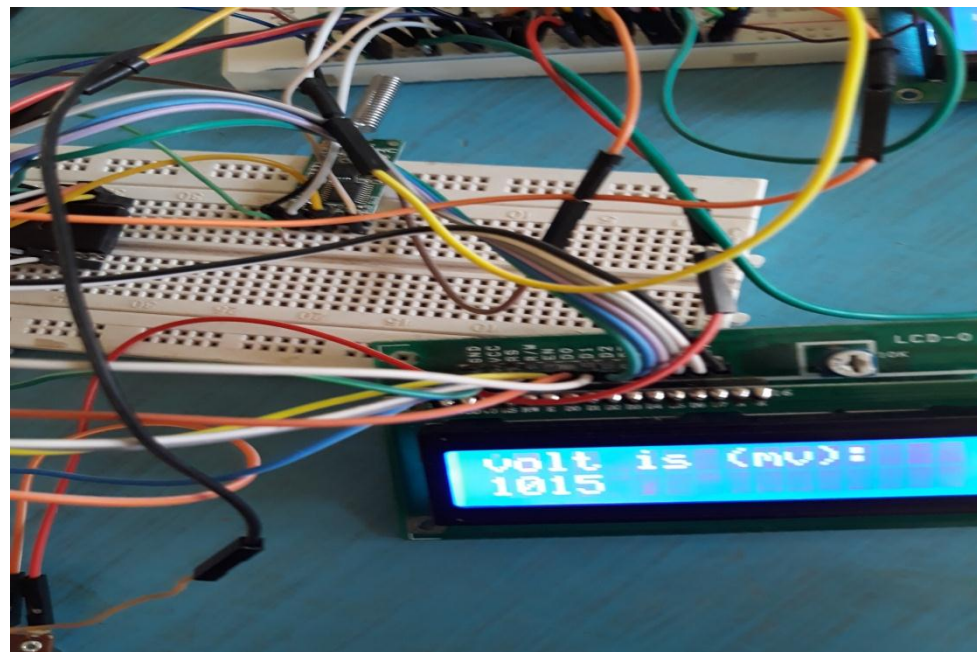

Figure 4.5: Screenshot of the implementation of displaying the digitized electrode value by 16x2 LCD in mV

4.3 Results of Wireless Communication

To simulate the wireless communication unit using Xbee module, a virtual serial port is needed. Virtual Serial Port Driver 8.0 software was used to provide these serial communications between the two XBee ports: COM3 and COM4 at the field side and monitoring side, respectively as in Figure 4.6. In addition, to configure the XBee, the virtual baud rate has been set to 1200 and virtual data bits to 8 as in Figure 4.7.
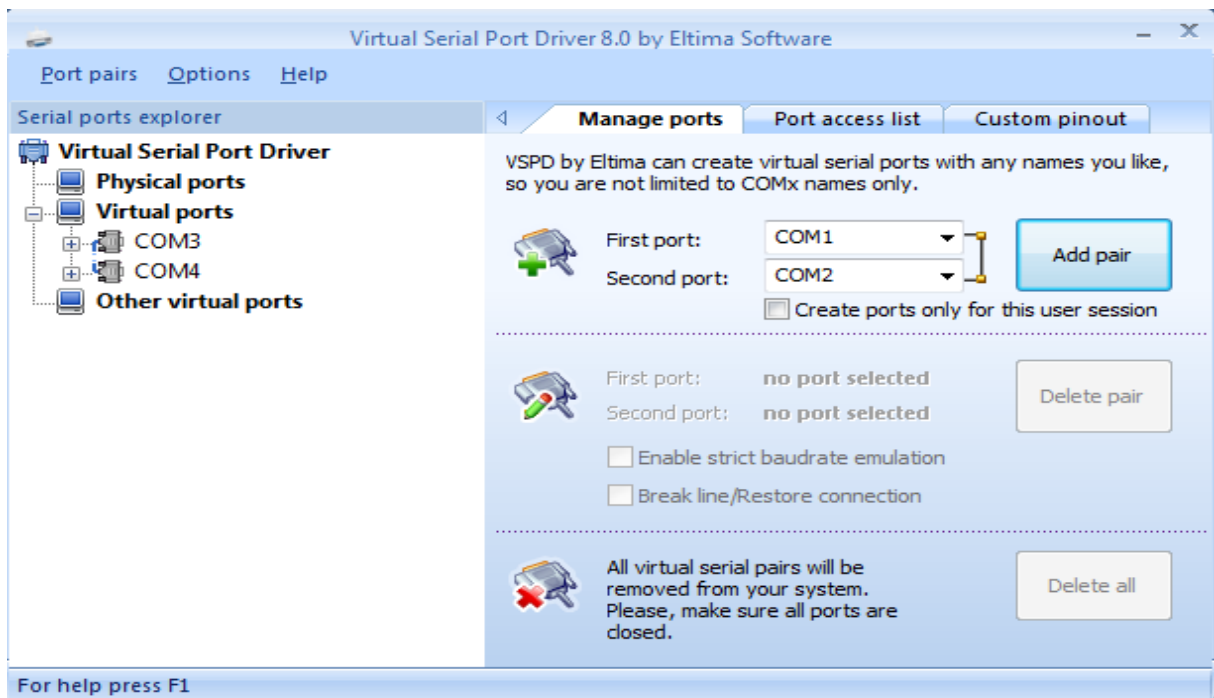

Figure 4.6: Screenshot of creating virtual ports between COM3 and COM4 using Virtual Serial Port Driver 8.0 software

By running the Proteus ISIS simulation with the wireless communication module Xbee at both field and monitoring side, the digitized electrode value can be serially transmitted to the Xbee at the field side and then wirelessly to the receiving Xbee at the monitoring side which is connected serially to the monitoring MC as can be seen in Figure 4.8.
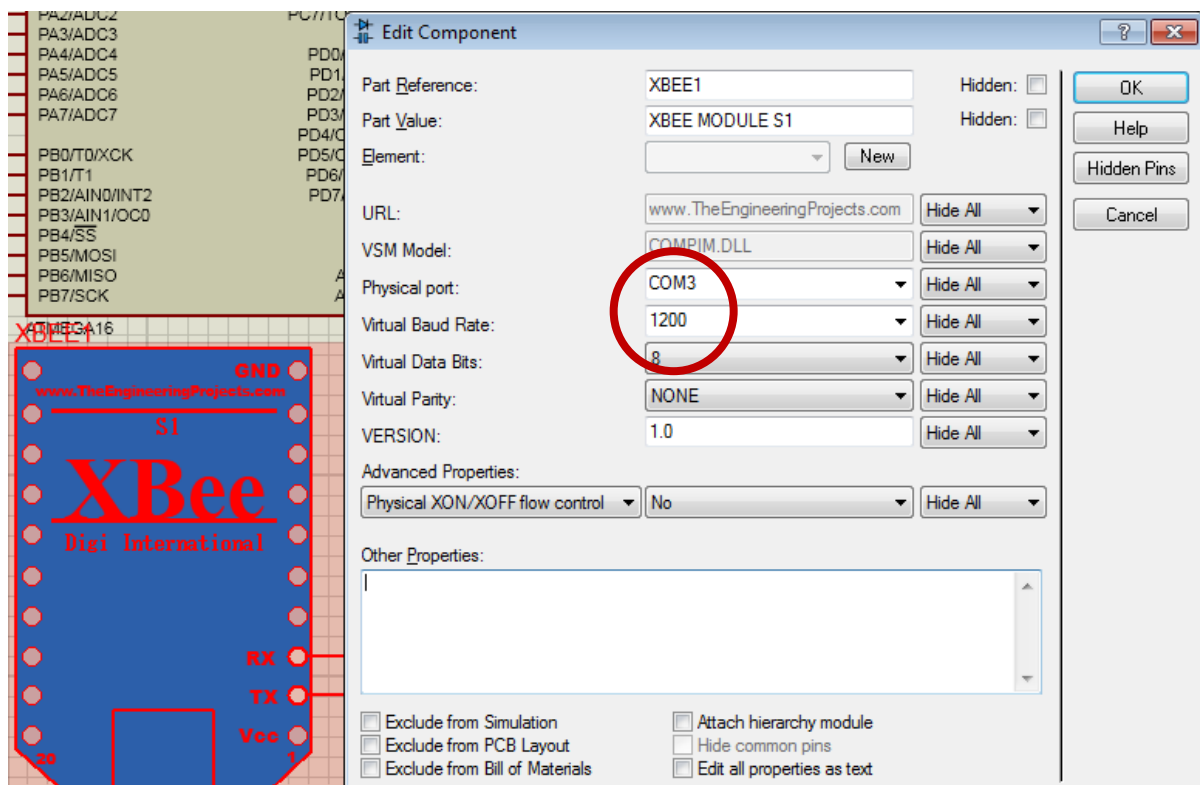
Figure 4.7: XBee module configuration for both field and monitoring sides

It is worthy to note that the exchanged electrode value between the two sides is coincided as can be recognized from their LCDs as in Figure 4.9.
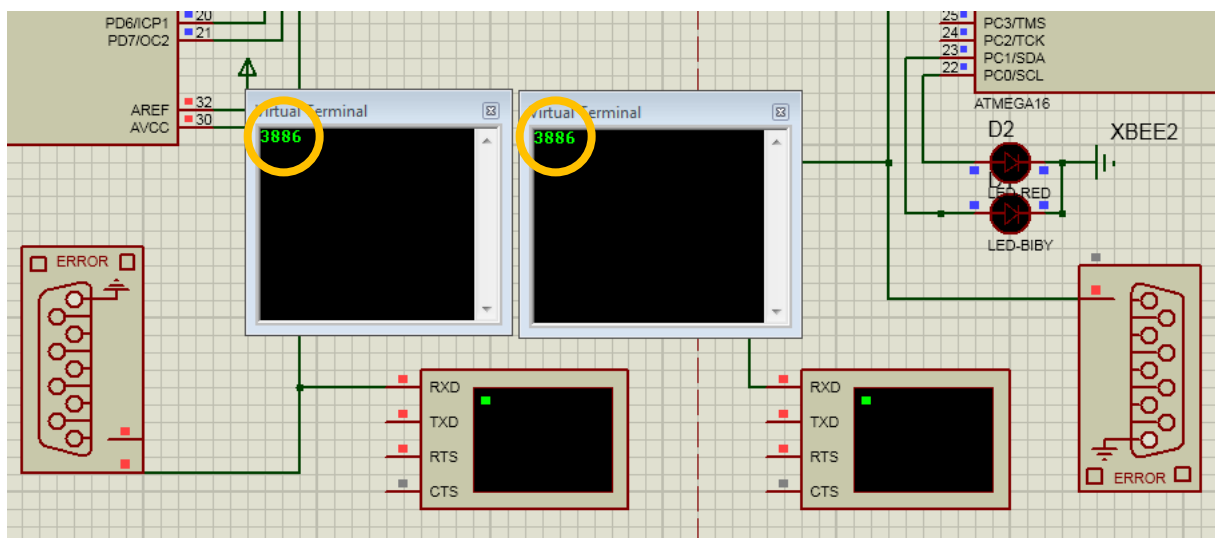

Figure 4.8: The transmission and receiving of the digitized electrode value serially using Xbee modules
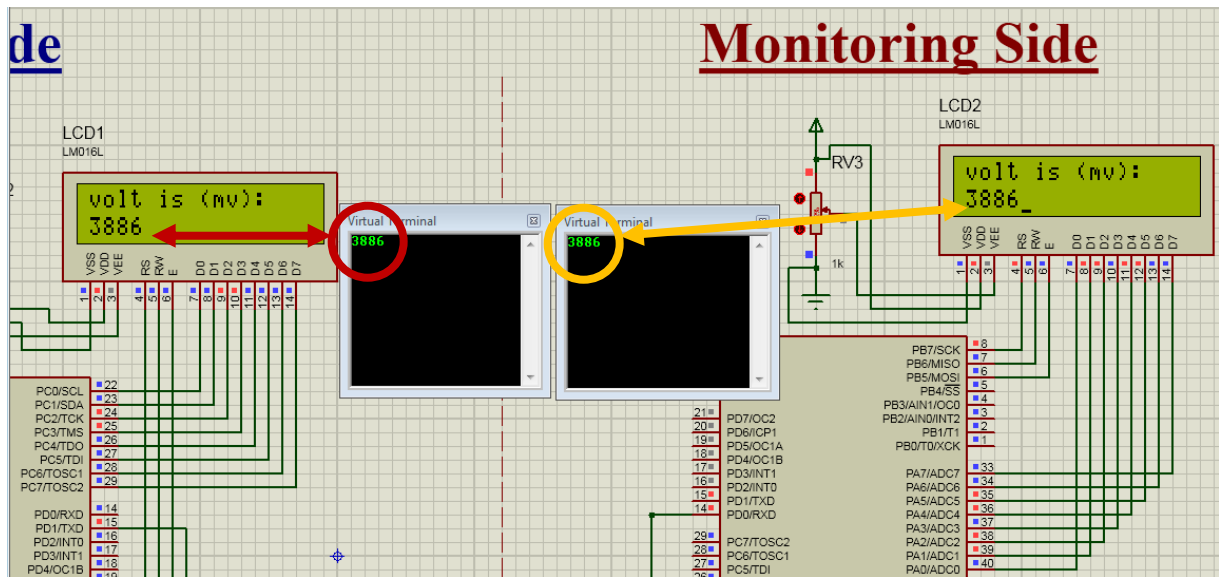
Figure 4.9: The coincidence of the transmission and receiving of the digitized electrode value serially using Xbee modules with the displayed values on LCDs

4.4 Results of Monitoring Side

As in Figure 4.9 and Figure 4.10, the digitized sensed electrode value is received accurately (in mV) at the monitoring side.
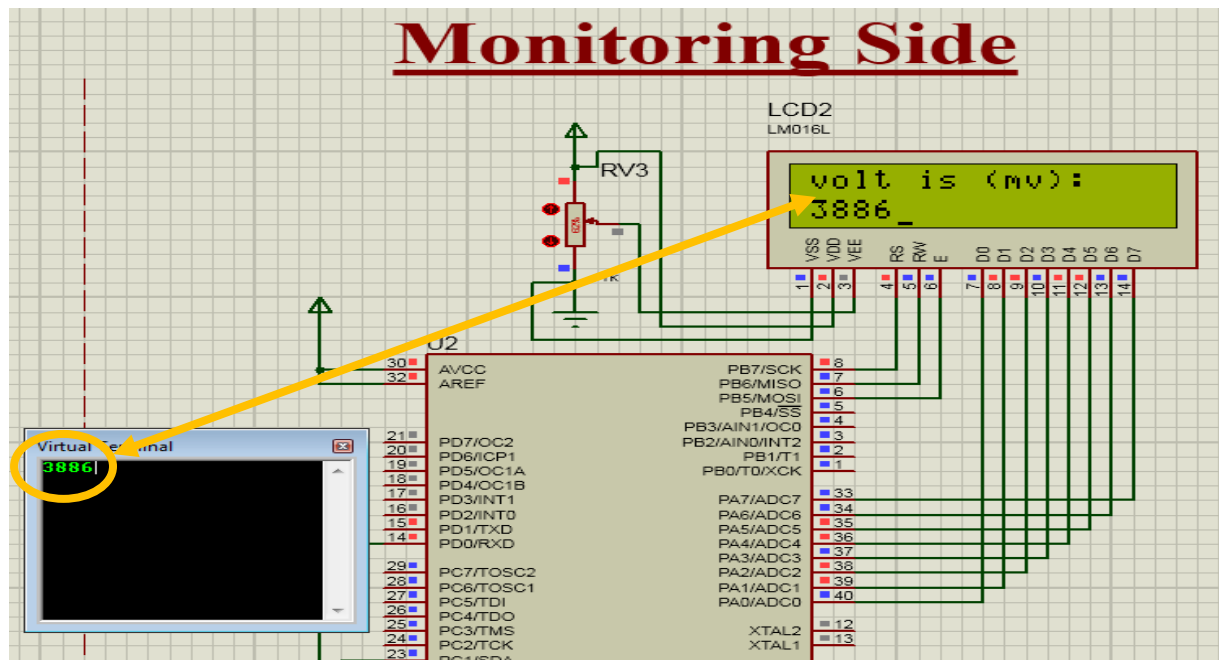

Figure 4.10: Display sensed voltage in the second LCD in mV

44

According to NACE standard, to protect the steel from corrosion, the protecting voltage of CP system should be within 850mV to 3000mV. The proposed system considers this criterion in the developed program. To test this criterion, three scenarios were simulated. In Figure 4.11, the potentiometer is set to read 2.15V which was digitized to 2148mV and received accurately at the monitoring side. It is recognized that this value is within the standard protecting voltage, therefore an alarm LEDs were not activated.
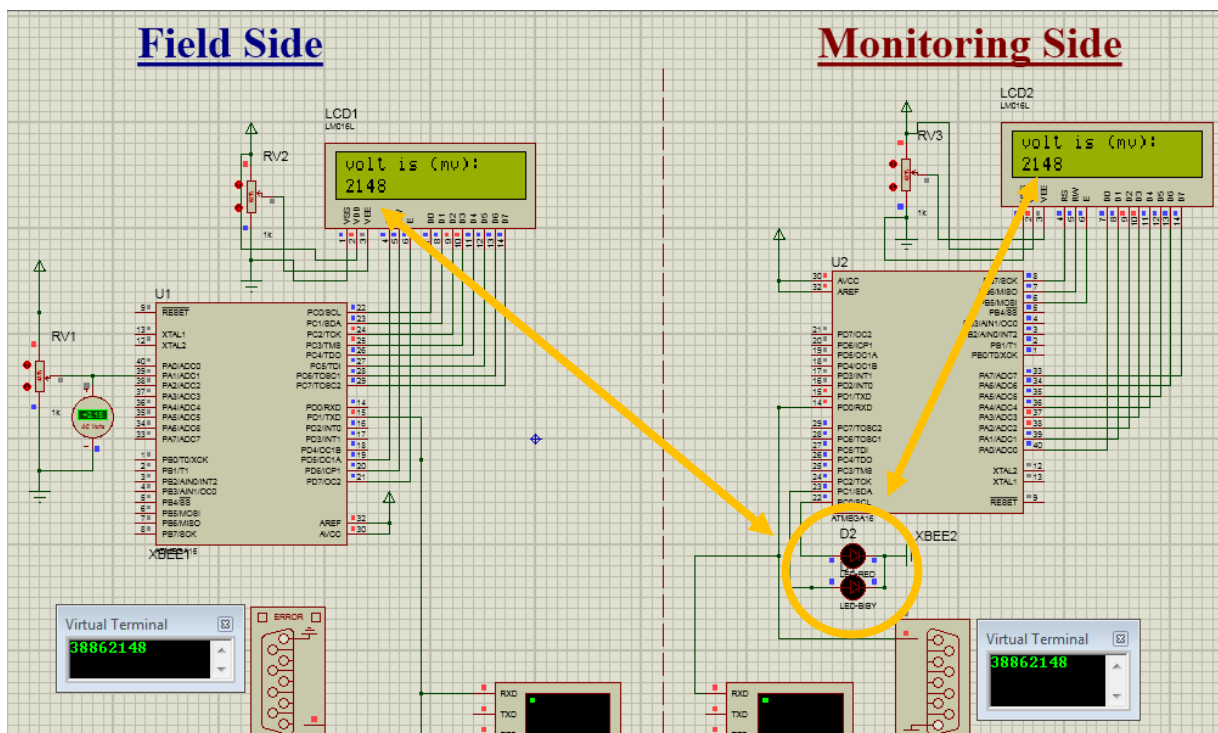
Figure 4.11: Warning diodes were not activated for the acceptable range of sensed voltage

To explore the response of the CP system when the CP voltage is greater than the maximum acceptable CP level, the potentiometer has been set to read 3.9 Volt. The red alarming LED at monitoring side is turned ON as in Figure 4.12. This red LED alarm always indicates that the bottom of the tank at the filed side is susceptible to coating disbondment as per NACE standard, therefore the

protecting level should be decreased accordingly be within the acceptable protection range (850 to 3,000 m Volt). In addition, the proposed system also can alarm the value less than the minimum protecting level (i.e. 850mV) as in Figure 4.13. In this Figure 4.13, the electrode reads 488mV which is less than 850mV; therefore, a yellow LED was turned ON indicating that the protection level provided is less than the required one to cease the corrosion current if it is present.
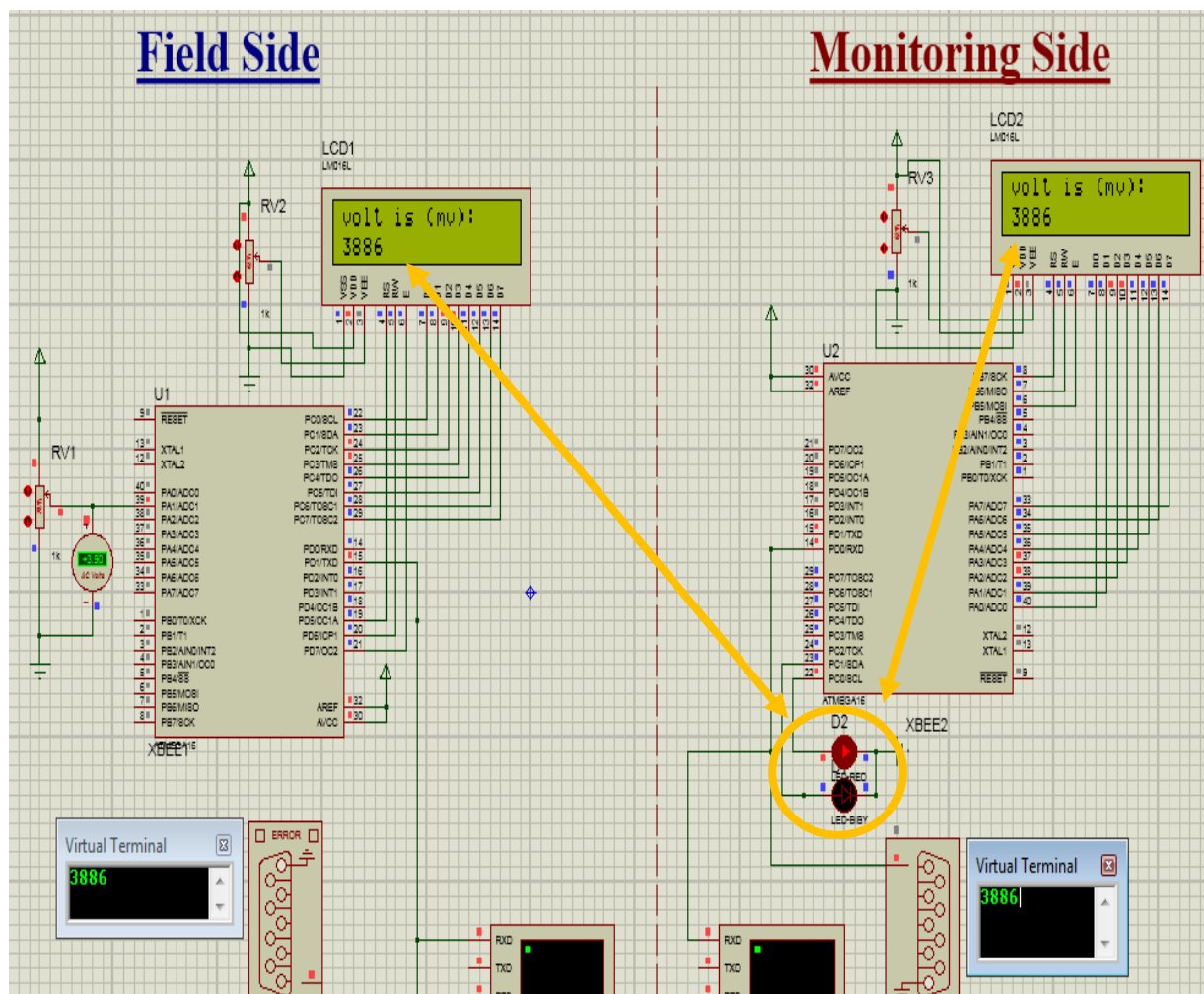
Figure 4.12: Warning red LED is activated for over protection range of sensed voltage
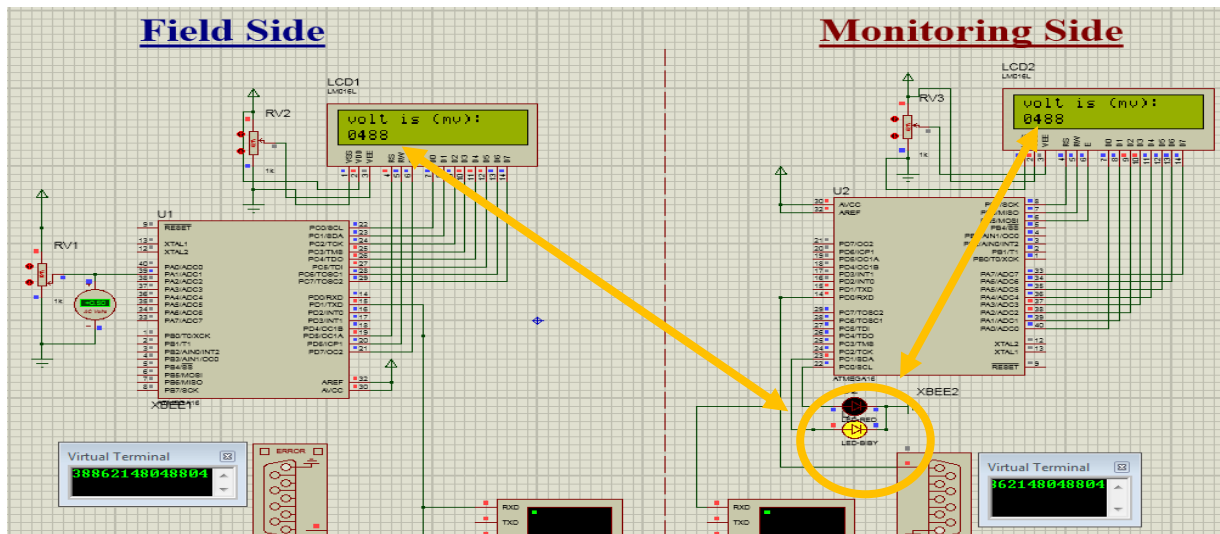
Figure 4.13: Warning yellow LED is activated for lower protection range of sensed voltage

Moreover, theses simulated three scenarios were also implemented in Figure 4.14, Figure 4.15 and Figure 4.16, respectively. Figure 4.14 represents the implementation of the scenario of accepted protection range i.e.850-3000mV where the LCDs display 2441mV as sensed voltage at the field side; therefore, both alarming LEDs were not activated.
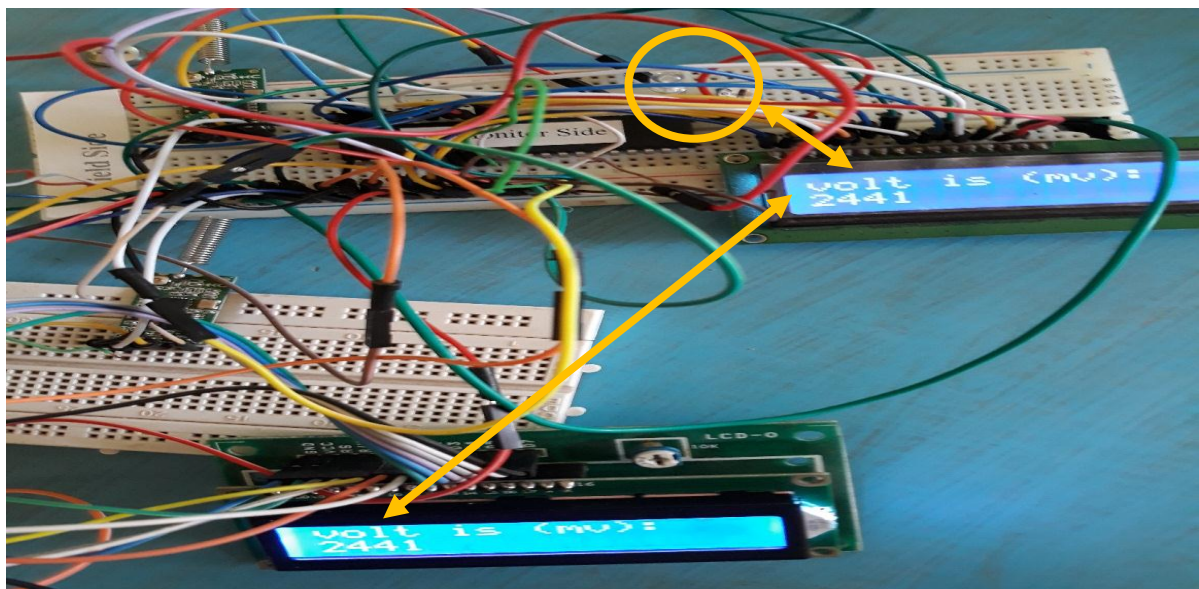
Figure 4.14: Implementation of warning diodes which were not activated for the acceptable range of sensed voltage

Figure 4.15 represents the implementation of the scenario of over protection range>3000mV where the LCDs display 3457 milli Volt as sensed voltage from the reference electrode at the field side; therefore, the green alarming LED was activated which means that the bottom tank coating may be damaged or disbanded from the steel. Figure 4.16 represents under protection range<850mV where the LCDs display 566 milli Volt as sensed voltage from the reference electrode at the field side; therefore, the yellow alarming LED was activated which means that the voltage is not enough to protect structure against corrosion and the feeding current shall be increased.
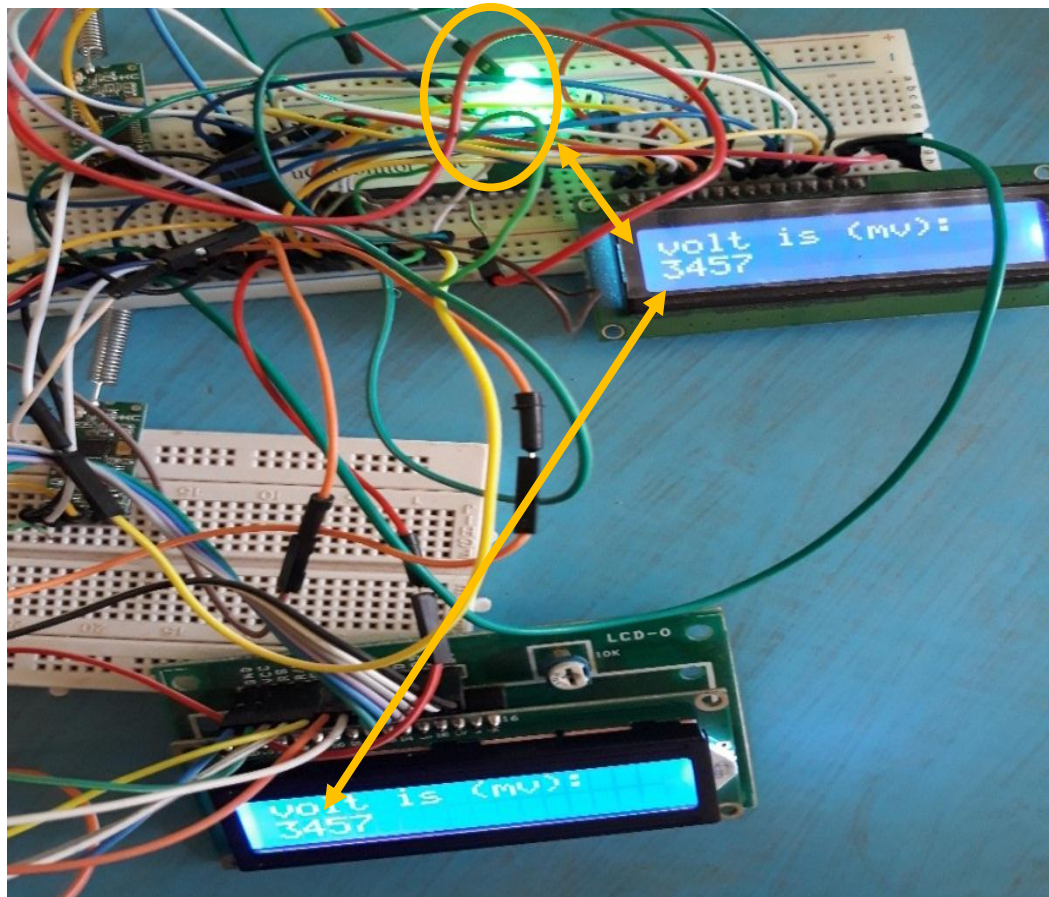
Figure 4.15: Implementation of green warning diode which was activated for the overprotection range of sensed voltage
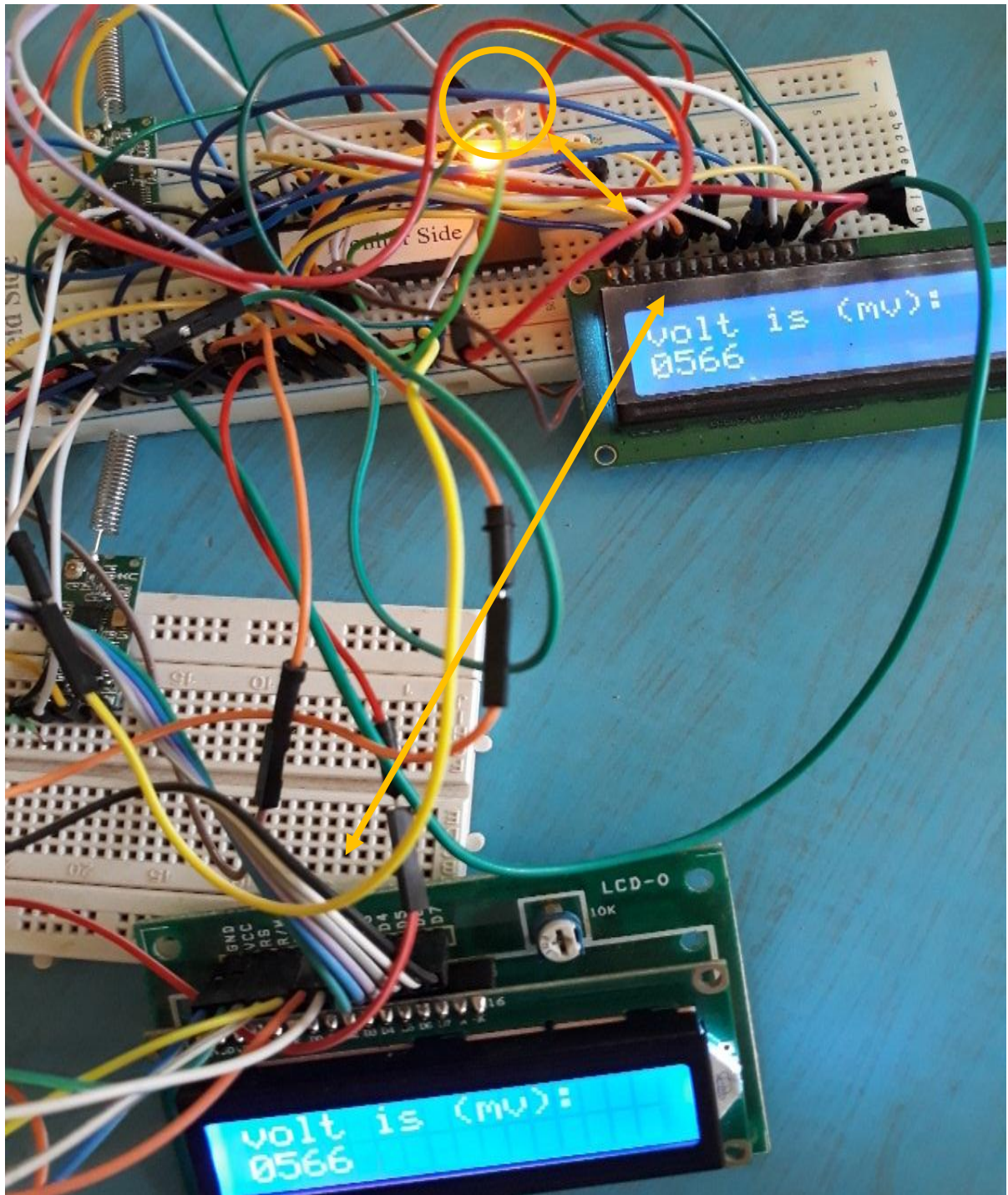

Figure 4.16: Implementation of yellow warning diode which was activated for the under protection range of sensed voltage

CHAPTER FIVE

CONCLUSION AND RECOMMENDATIONS

CHAPTER FIVE
CONCLUSION AND RECOMMENDATIONS

5.1. Conclusion

The proposed CP monitoring system has been designed and simulated using Proteus 8 simulator and Atmel Studio 6. The voltage data out of the sensor (Reference Electrode) has been transformed to digital form and displayed in the LCD at the field side successfully. Then the digitized date has been wirelessly transmitted using serial protocol to the monitoring side. The Xbee modules were used to provide the wireless communication. In addition, the monitoring side was equipped with alarming LEDs to detect over, under and acceptance range of protection referring to NACE standard. Moreover, results of the simulated and implemented CP monitoring system were coincided.

To implement this system at CPF at Jabalain at White Nile State, Sudan, the supplied power can be fed from the junction box at the crude tanks. In addition, to avoid the exposure of the sun light, heat, dust and other environment impairments, a suitable case with cooling fan as well as non-explosion roof can be used at the field side.

5.2. Recommendations and Future Work

To improve the proposed CP monitoring system, the following recommendations and future work can be conducted:

- The proposed system can be integrated with SCADA system where whole transportation system including CPF utilizes SCADA system. Therefore, a unified system can be recognized.

- The proposed system is simulated and implemented for single CP monitoring system; a network based system can be designed where the utilized Xbee module is the core unit for any wireless sensor network.

- This proposed system is designed for a deep well anode CP system, therefore, it is recommend improving this system to work on anode flex loops tank CP system by adding multiplexing and synchronization algorithm for the multi-anode-nodes in the tank.

- The proposed CP monitoring system can be monitoring and control system by adding a feedback loop from the monitoring side to control the TR at the field side. This can be achieved by utilizing the wireless Xbee module to be transceiver unsteadily.

- The system can be doubled easily for backup in case of system failure.

REFERENCES

1- NACE Cathodic Protection subcommittee," CP 2–Cathodic Protection Technician ", United States, February 2005.

2- NACE Cathodic Protection subcommittee," CP 1–Cathodic Protection Tester ", United States, February 2005.

3- RanaM.Hussain. "Cathodic Protection of Storage Tank", 2012.

4- A. Peratta, J. Baynham, and R. Adey, Gervasio F. Pimenta "Intelligent Remote Monitoring System for Cathodic Protection of transmission pipelines", 2009

5- Moh. Islam GamarEldeen "Interface between Cathodic   Protection and Scada System((Case Study: PETRODAR Operating Company)), 2011

6- Pingting Liu1*, Zhiyu Huang1, Shihong Duan2, Zikang Wang2 and Jie He2 "Optimization for Remote Monitoring Terrestrial Petroleum Pipeline Cathode Protection System Using Graded Network", 2015

7- Prof. Dr. mohammedzeki Al−faiz, Eng. Liqaasaadimezher " Sacrificial Anode Cathodic Protection Remote Monitoring And Control Using Labview And Micaz Motes", 2014.

8- Xiaohua Wang, Meijiao Hu, Yao Huang, Jun Gu, ShouweiGao, Zhonghua Miao" Design and Implementation of the Remote Control of ICCP Systems", 2014

9- Labcenter Electronics, "Proteus All-in-One Manual v6[1]_3", November 2003

10- Muhammad Ali Mazidi, SarmadNaimi, SepehrNaimi , "The AVR microcontroller and embedded systems: using Assembly and C", 2011

11- DIGI, "Zigbee RF Modules XBEE2, XBEEPRO2, PRO S2B User Guide", 2018

12-   Digi International, "Product Manual v1.xEx - 802.15.4 Protocol For RF Module Part Numbers: XB24-A...-001, XBP24-A...-001 IEEE® 802.15.4 RF Modules", 2009

APPENDIX A

Field side MC Code Using C Language

```c
#define F_CPU 2000000UL
#include <avr/io.h>
#include <util/delay.h>
#define RS PD5                          // Define Register Select pin */
#define RW PD6
#define EN PD7
void LCD_init(void);
void LCD_Command(unsigned char cmnd);
void LCD_print (char *str);
void ADConverter (void);
void LCD_data( unsigned char data);
void USART_init(void);
void USART_send(unsigned char data);
volatile double adcValue;
volatile unsigned char x[4];
int main(void)
{
DDRA = 0x00; // port A input
DDRC = 0xff; // port C Data output
DDRD = 0xff; // port D code output
_delay_ms(1);
LCD_init(); // Initialization of LCD
_delay_ms(15);
LCD_print("CP voltage(mv)");// Write string on 1st line of LCD
_delay_ms(15);
USART_init();
while(1)
{
x[0]=122;
ADConverter();
LCD_Command(0xC0);          // Go to 2nd line
USART_send(x[0]);
USART_send(x[1]);
USART_send(x[2]);
USART_send(x[3]);
```

```c
USART_send(x[4]);
LCD_data(x[1]);
LCD_data(x[2]);
LCD_data(x[3]);
LCD_data(x[4]);
_delay_ms(250);
}
}
void LCD_init (void)
{
LCD_Command(0x38);
_delay_ms(200);
LCD_Command(0x0E);  // Send for 4 bit initialization of LCD
_delay_ms(200);
LCD_Command(0x01);  // 2 line, 5*7 matrix in 4-bit mode
_delay_ms(200);
}
void LCD_Command(unsigned char cmnd )
{
PORTD &= ~ (1<<RS);
PORTD &= ~ (1<<RW);          // RS=0,RW=0, command reg.
PORTC =cmnd;
PORTD |= (1<<EN);            // Enable pulse
_delay_us(1);
PORTD &= ~ (1<<EN);
_delay_us(200);
}
void LCD_print (char *str)                // Send string to LCD function
{
int i;
for(i=0;str[i]!=0;i++)                    // Send each char of string till the NULL
{
LCD_data (str[i]);
}
}
void LCD_data( unsigned char data )
{
PORTD |= (1<<RS);  // RS=1, data reg.
PORTD &=~ (1<<RW); // RW=0, data reg.
PORTC = data; // Send data to data port
PORTD |= (1<<EN);
```

```c
_delay_us(100);
PORTD &=~ (1<<EN);
_delay_ms(100);
}
void ADConverter(void)
{
int adcvalue1;
ADCSRA |= (1 << ADEN); //activate adc
ADCSRA |=(1<< ADPS0)|(1<< ADPS1)|(0<<ADPS2);//CK division factor(128)
ADMUX |= (1 << ADLAR); //activate 8 bit (ADCH)
ADMUX |= (1 << MUX0); //choose terminal to read input from (00001 =ADC1)
ADCSRA |= (1 << ADSC); //start conversion
while(ADCSRA & (1<<ADSC)); //wait till finish conversion
adcValue = ADCH;
adcvalue1=(adcValue*5/256)*1000;  // analogue value 4.9
x[1]=(adcvalue1/1000)+48;
x[2]=((adcvalue1%1000)/100)+48;
x[3]=(((adcvalue1%1000)%100)/10)+48;
x[4]=((((adcvalue1%1000)%100)%10)/1)+48;
}
//***************USART initialization*************
void USART_init(void)
{
UCSRC = (1 << URSEL ) | (1 << UCSZ0 ) | (1 << UCSZ1 );
UBRRL = 0x0C;
UBRRH = 0x00;
UCSRB = (1 << RXEN ) | (1 << TXEN );
}
//*************** USART Send*************

void USART_send(unsigned char data)
{
while ((UCSRA & (1<<UDRE)) == 0x00){};
UDR = data;
}
```

APPENDIX B

Monitoring side MC Code Using C Language

```c
#define F_CPU 2000000UL
#include <avr/io.h>
#include <util/delay.h>
#define RS PD5   // Define PB7 as RS
#define RW PB6  // Define PB6 as RS
#define EN PD7   // Define PB5 as RS
void LCD_init(void);
void LCD_Command(unsigned char cmnd);
void LCD_print (char *str);
void LCD_data( unsigned char data);
void USART_init(void);
unsigned char USART_receive(void);
unsigned char x[4],y[3];
int main(void)
{
    DDRD = 0b11100000;
    DDRB = 0xff; // Define port B as output
    DDRC = 0xff;
    _delay_ms(1);
    LCD_init(); // Initialization of LCD
    _delay_ms(15);
    LCD_print("CP voltage(mv)"); // Write string on 1st line of LCD
    LCD_Command(0xC0); // Go to 2nd line
    _delay_ms(15);
    USART_init();
    while(1)
        {
            x[0]=USART_receive();
            if(x[0]=122)
                {
                x[1]=USART_receive();
                LCD_data(x[1]);
                x[2]=USART_receive();
                LCD_data(x[2]);
                x[3]=USART_receive();
                LCD_data(x[3]);
```

```c
                                    x[4]=USART_receive();
                                    LCD_data(x[4]);
                                    LCD_Command(0xC0); // Go to 2nd line   3 0 0 0
                                    if(x[1]>50)
                                    PORTB=0x01;
                                    else
                                            {
                                                    if(x[1]<49)
                                                    {
                                                            if(x[2]<56)
                                                            {
                                                                    PORTB=0x02;
                                                            }
                                                            else
                                                            PORTB=0x00;
                                                    }
                                                    else
                                                    PORTB=0x00;
                                            }

                                    }
                            }
}

//***************LCD Initialize function*******************
void LCD_init (void)
{
        LCD_Command(0x38); //Set LCD: 2 rows and 5x7 matrix (DO-D7, 8-bit)
        _delay_ms(200);
        LCD_Command(0x0E);//Initialization of LCD (Disp on, cursor blinking)
        _delay_ms(200);
        LCD_Command(0x01); // Clear display screen
        _delay_ms(200);
}
//-----------------------------------------------------
void LCD_Command(unsigned char cmnd )
{
        PORTD &= ~ (1<<RS);
        PORTD &= ~ (1<<RW);              // RS=0,RW=0, command reg.
        PORTC =cmnd;
        PORTD |= (1<<EN);                // Enable pulse
```

```c
        _delay_us(1);
        PORTD &= ~ (1<<EN);
        _delay_us(200);
}
//************Print a String*************
void LCD_print (char *str)
{
        int i;
        for(i=0;str[i]!=0;i++)              // Send each char of string till the NULL
        {
                LCD_data (str[i]);
        }
}
//**************Print a Character*************
void LCD_data( unsigned char data )
{
        PORTD |= (1<<RS);  // RS=1
        PORTD &=~ (1<<RW); // RW=0
        PORTC = data;
        PORTD |= (1<<EN); // EN=1
        _delay_us(100);
        PORTD &=~ (1<<EN); // EN=0
        _delay_ms(100);
}
//**************USART initialization*************

void USART_init(void)
{
        UCSRC = (1 << URSEL ) | (1 << UCSZ0 ) | (1 << UCSZ1 );
        UBRRH = 0x00;
        UBRRL = 0x0C;
        UCSRB = (1 << RXEN ) | (1 << TXEN );
}
//************** USART recieve*************

unsigned char USART_receive(void)
{
        while ((UCSRA & (1<<RXC)) == 0x00){};
        return (UDR);
}
```