

SENSOR NETWORKS


YASER ISSAM

SEPTEMBER, 2012


Title of the Thesis : **SENSOR NETWORKS**

Submitted by **YASER ISSAM**

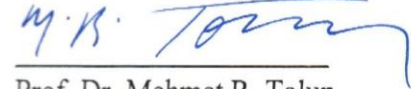
Approval of the Graduate School of Natural and Applied Sciences, Çankaya University.


Prof. Dr. Taner Altunok
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Computer Engineering.


Asst. Prof. Dr. Murat Saran
Acting Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree Master of Computer Engineering.


Prof. Dr. Mehmet R. Tolun
Supervisor

Examination Date : 28.09.2012

Examining Committee Members

Prof. Dr. Mehmet R. TOLUN

(TED Univ.)



Assist. Prof. Dr. Fahd JARAD

(Çankaya Univ.)



Asst. Prof. Dr. Reza HASSANPOUR (Çankaya Univ.)



STATEMENT OF NON PLAGIARISM

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name : YASER HAMODI

Signature : 

Date : 28.09.2012

ABSTRACT

SENSOR NETWORKS

HAMODİ, YASER

M.Sc., Department of Computer Engineering

Supervisor: Prof.Dr. Mehmet R. Tolun

September 2012, 65 Pages

Over the past two decades, the emergence of numerous new communication technologies has been witnessed. Among these, networking, optics, infrared, bluetooth are some. Many researchers have been focusing on this matter recently because nowadays, these technologies are used in a lot of fields in some emergency cases, especially to enable the monitoring of physical and environmental parameters for a variety of applications such as those used at homes, in offices, clinics, factories, vehicles, global environment and fire stations. These applications could monitor security and safety in the future in many private and local sectors. This research focuses on sensor network technology used in driving smaller devices that use less power with greater functionality. It includes low – power microcontrollers with networking communication and various digital and analog sensors, a battery operated network of sensor modules to acquire a wide range of data and the real – time operating system to address the priorities of such a sensor network using low power, hard real – time constraints, and robust communication. This thesis is the first step to monitoring fire and keeping it under control in conventional buildings and factories, and considers feasibility and reliability of sensor networks.

KeyWords: Sensor Networks, LM75, AT89c52, Server and Client, Monitoring.

ÖZET

ALGILAYICI AĞLARI

HAMODİ, YASER

Bilgisayar Mühendisliği Bölümü, Yüksek Lisans Tezi

Danışman: Prof. Dr. Mehmet R. Tolun

Eylül 2012, 65 Sayfa

Yirmi yıldan uzun süredir, pek çok yeni iletişim teknolojisinin ortaya çıkışına şahit olunmaktadır. Bunlar arasında, ağ ,optik, kızılötesi ve bluetooth örnek gösterilebilir. Pekçok araştırmacı bu konuya , bu teknolojilerin acil durumlarda kullanılması ile alanlara, özellikle de ev, ofis, klinik, fabrika, ulaşım araçları, küresel çevre ve yangın istasyonları gibi fiziksel ve çevresel parametrelerin ele alınabileceğinden ötürü, bugünler de odaklanmaktadır. Bu uygulamalar gelecekte özel ve bölgesel sektörde emniyetin ve güvenliğin gözlemlenmesinde kullanılabilir. Bu araştırma algılayıcı ağları teknolojisinin daha az enerji tüketimini daha fazla işlevsellikle sağlayan küçük cihazlarda kullanımına odaklanmaktadır. Bu araştırma ağ iletişimini ve pekçok sayısal ve analog algılayıcı içeren düşük güç mikrokontrolcilerini, geniş alandan veri toplamayı sağlayan pille çalışan algıyıcı modüllerini içeren ağları ve düşük enerji kullanımı veya gerçek zamanlı işlem veya sorunsuz iletişim gibi kısıtlara göre gerçek zamanlı kararları destekleyen işletim sistemini ele almaktadır. Bu tez geleneksel, tarihi(conventional) ve fabrikalarda yangın gözleme ve yangını kontrol altında tutma ve de algılayıcı ağlarının uygulanabilirliği ve güvenilebilirliği konularında bir ilk adımdır.

Anahtar Kelimeler: Algılayıcı ağları, LM75, AT89c52, Sunucu ve İstemci, Gözleme.

ACKNOWLEDGMENTS

In the name of God's, the Merciful, praise be to God's and pray, peace be on his prophet Mohammed (God's blessing), and peace be on his relatives. Praise is to God's, who created us and gave us the ability and mentality to think and work for the benefit of human beings.

The author wishes to express his deepest gratitude to his supervisor **Prof. Dr. Mehmet R. Tolun** for the guidance, advice, criticism, encouragement and insight throughout the research.

I would like to express my deep gratitude to **Prof. Dr. Taner Altunok**, who supported me and helped me a lot in the academic phase in study.

I would like to express my deep gratitude to **Assit. Prof. Dr. Fahd Jarad**, who has supported me a lot during this study.

Finally, I offer my sincere thanks to my parents, my brothers, and special my wife and my son for supporting me in this study.

TABLE OF CONTENTS

STATEMENT OF NON-PLAGIARISM.....	iii
ABSTRACT.....	iv
ÖZEL.....	v
TABLE OF CONTENTS.....	vii
LIST OF TABLES.....	ix
LIST OF FIGURES.....	x

CHAPTERS:

INTRODUCTION

1.1. RELATED RESEARCH.....	1
1.2. OBJECTIVES	2
1.3. THESIS LAYOUT.....	2

FOUNDAMENTALS OF SENSOR NETWORKS

2.1. DATA COMMUNICATION.....	3
2.2. NETWORKS.....	4
2.3. DISTRIBUTED PROCESSING.....	5
2.4. NETWORK CRITERIA.....	5
2.4.1. Performance.....	5
2.4.2. Reliability.....	5
2.4.3. Security.....	5
2.5. SENSOR NETWORK.....	6
2.6. DSN – DISTRIBUTED SENSOR NETWORK.....	7
2.7. SENSORS.....	8
2.8. SYSTEM OF SENSOR NETWORKS COMMUNICATION.....	9
2.9. DIJSTRA'S ALGORITHM.....	10
2.9.1. Example of Dijkstra's Algorithm	10

2.10. COMMUNICATION CONTROL IN VISUAL BASIC.....	13
2.10.1. Receiving And Transmitting Data.....	14
2.11. DATA BASE.....	15
DESIGN AND APPLICATION OF PROPOSED WORK	
3.1. OVERVIEW.....	16
3.2. BLOCK DIAGRAM DESCRIPTION.....	16
3.3. STEPS OF PROCESSING	17
3.4. CLIENT SIDE SYSTEM.....	20
3.5. SERVER SIDE SYSTEM.....	23
3.6. HARDWARE DESIGN.....	24
3.6.1. Electronic Circuit Description In Client Side.....	24
EXPERIMENTAL RESULTS	
CONCLUSION AND FUTURE WORK	
5.1. CONCLUSION.....	34
5.2. FUTURE WORK.....	34
REFERENCES.....	35
APPENDIX	
A- CLIENT APPLICATION – VISUAL BASIC 6.0.....	36
B- SERVER APPLICATION – VISUAL BASIC 6.0.....	40
C- BASCOM IDE SOURCE CODE.....	63
D- GENUS UNIVERSAL PROGRAMMABLE – G540.....	64
E- CURRICULUM VITAE.....	65

LIST OF TABLES

Table 4.1. Temperature Sensor Measurements on Various Environments.....	31
Table 4.2. Connections and lengths of The Designed Topology.....	32
Table 4.3. All connections of The Designed Topology.....	33

LIST OF FIGURES

Figure 2.1 Work of Sensor Networks	4
Figure 2.2. Sensor Networks Representation	7
Figure 2.3. Networking structure of a distributed sensor network.....	9
Figure 2.4. System of Sensor Networks	10
Figure 2.5. Example of Dijkstra's algorithm Topology.....	11
Figure 2.6. "AC" check path.....	11
Figure 2.7. "AB" Path.....	11
Figure 2.8. "CE" path.....	12
Figure 2.9. "BD" path.....	12
Figure 2.10. "EF" path	13
Figure 3.1. Main Block Diagram Of Proposed Work.....	16
Figure 3.2. Serial Port Pin Configuration between Microcontroller and Serial Port.....	19
Figure 3.3. Client Main Component.....	21
Figure 3.4. Client Flowchart on Network Data Transceiver.....	22
Figure 3.5. Device Interfaced on Client Side.....	23
Figure 3.6. Server Main Component.....	24
Figure 3.7. Schematic Circuit in Client Side.....	25
Figure 3.8. Interface Application Design in Client Side.....	26
Figure 3.9. Interface of Server Side Application.....	26
Figure 3.10. Schematic Design of Server Side.....	27
Figure 4.1. Pin Connection of I2C.....	28
Figure 4.2. Electronic Circuit Interfaced to Client Side.....	29
Figure 4.3. Electronic Circuit Interfaced To Server Side.....	30
Figure 4.4. Temperature Changes Over Time in Various Environments.....	31
Figure 4.5. Shortest Path Drawn from Source to Node (E).....	33

GCPRIS

CHAPTER I

INTRODUCTION

Every environment has different kinds of sensors are in use, and these sensors are used in various fields such as buildings automation to controlling the lights, access allowance control, and refrigeration control, besides that, some other sensors used to measurement such as temperature, pressure, level sensing and machinery monitoring. Also, in power, there are sensors implemented as meters or for power distribution diagnostics. Nowadays, there are several different wire-based or actuators network products can be found in various fields. The wired sensors are expensive to install, inflexible once installed, limited in size, in complexity, and in functionality compared with wireless sensor networks are not restricted by these limitations [1].

1.1. RELATED RESEARCH

In [2], the research considered the fire as major concern to those who work in underground coal mines. It is focused on important issue which has a warning system that is capable of detecting fire and generating an alarm. Besides that, and it is explanation of the uses of Wireless Sensor Networks (WSN) to detect the specified fire location and also providing the fire prevention system to stop the spread of fire to save the natural resources and the mining personnel from fire [2].

1.2. OBJECTIVES

1. To study the sensor network and wireless sensor network environments, the different range of applications used and to look into the technology, the software, the hardware, the protocol, the network operating environment .
2. To study the sensor network communication protocols stack.
3. To study a microcontroller (8051).
4. To design and develop a suitable network and control protocol for the sensor nodes with highly efficient.
5. To measure and test the performance of the network and operation of sensor.

1.3. THESIS LAYOUT

In chapter two, will discuss the fundamentals of communication, the sensor networks and shortest path theory and fundamentals.

In chapter three, will discuss proposed system in both software and hardware design. Besides that, will discussed fundamentals of implemented algorithm (Dijkstra) to find the shortest path.

In chapter four, an experimental result will discuss and focused on hardware process in both client and server, in addition, analysis the Dijkstra algorithm in server side.

In chapter five, a conclusion and future work has been provided.

CHAPTER II

FOUNDAMENTALS OF SENSOR NETWORKS

2.1. DATA COMMUNICATION

Data communications are the exchange of data between two devices via some form of transmission medium such as a wire cable. For data communications to occur, the communicating devices must be part of a communication system made up of a combination of hardware (physical equipment) and software (programs). The effectiveness of a data communications system depends on four fundamental characteristics: delivery, accuracy, timeliness, and jitter [4].

1. **Delivery.** The system must deliver data to the correct destination. Data must be received by the intended device or user and only by that device or user [4].
2. **Accuracy.** The system must deliver the data accurately. Data that have been altered in transmission and left uncorrected are unusable [4].
3. **Timeliness.** The system must deliver data in a timely manner. Data delivered late are useless. In the case of video and audio, timely delivery means delivering data as they are produced, in the same order that they are produced, and without significant delay. This kind of delivery is called real-time transmission [4].
4. **Jitter.** Jitter refers to the variation in the packet arrival time. It is the uneven delay in the delivery of audio or video packets. For example, let us assume that video packets are sent every 3D-ms. If some of the packets arrive with 3D-ms delay and others with 4D-ms delay, an uneven quality in the video is the result [4].

2.2. NETWORKS

A network is a number of clients connected with one server work with some protocols and it has set of devices and sometimes considered as nodes, and these nodes connected to each other by communication links. A node can be any device such as computer, printer, or any other device has ability to send and receive data via network communication. And the figure 2.1. Explain how to networks of sensors work in the software and hardware tools used in the system and explain the programs used to completed the all software applications and in the same time in hardware side divided to two part the first one of communication tools included the RS232 cable connections and network cable TCP/IP and the second is circuit design divided in two part client circuit and server circuit that showing in the Figure 2.1. [4].

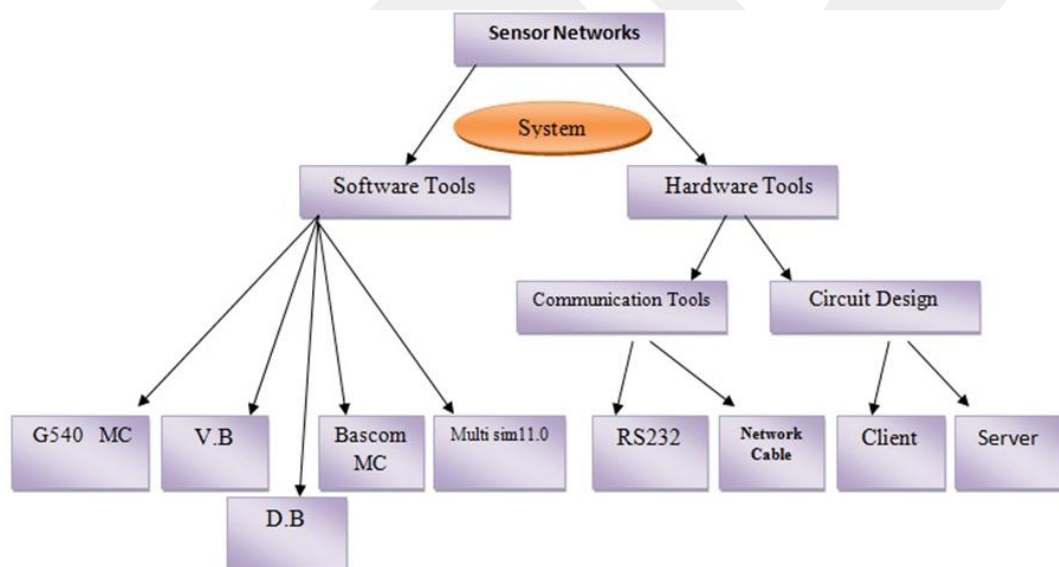


Figure 2.1. Work of Sensor Networks.

2.3. DISTRIBUTED PROCESSING

Distributed processing is a task among multi network devices, besides that, there is one machine tasks of process, and handling computers separation [4].

2.5. NETWORK CRITERIA

In network communication, there are criteria should be takes in consideration such as (performance, reliability, and security), and will discuss each of these criteria [4].

2.4.1. Performance

These criteria can be measured by transit and response time. The transit time means the amount of time which required when a message start is send from a device to another. The response time is the elapsed time between an inquiry and a response. In addition to that, the performance of a network could be based on a number of factors, instead number of users, transmission medium, the capabilities of the connected hardware, and the efficiency of the software [4].

2.4.2. Reliability

Other criteria, is reliability of network, and it one of important criteria of network which measure the failures of frequency, and how much time the link takes to recover from a failure [4].

2.4.3. Security

- The network security includes protecting data from unauthorized access, protecting data from damage and development, and implementing policies and procedures for recovery from breaches and data losses and when application the security roles of designer for all fields software and hardware systems in the data set used it we found the perfect result in this fields we

take it more of Security in software application and full Security for hardware components such as the Micro Controller and Security in D.B design and Security between connection client and server [4].

2.5. SENSOR NETWORK

Sensor networks represent a technology which execute a complex issues many inexpensive electromechanical micro-devices. These sensors could be attached in computational devices and/or embedded devices. In addition to that, these sensors used to be monitor changes in the operational environment. The sensor network devices could be found in battlefield, these devices may self-organize to act as numerous eyes and ears of soldiers surveying the field from a safe distance. Recently, the sensors found in embedded in unmanned air vehicles, they may monitor bio/chemical plumes in the atmosphere or handle hazardous materials on the ground. Also, the sensor network has been used as a mobile robots with embedded sensor systems which used as explorer the surface of planets; and found in systems of undersea robots are being designed to hunt for mines in shallow water and to develop high fidelity now casts and forecasts of the ocean through time-space coordinated sampling and in Figure 2.2. Explain the system of work between the server and client and included the type of connection between computers in the client side or the server side all of system control by the software applications with the same protocols show in the Figure 2.2. [5].

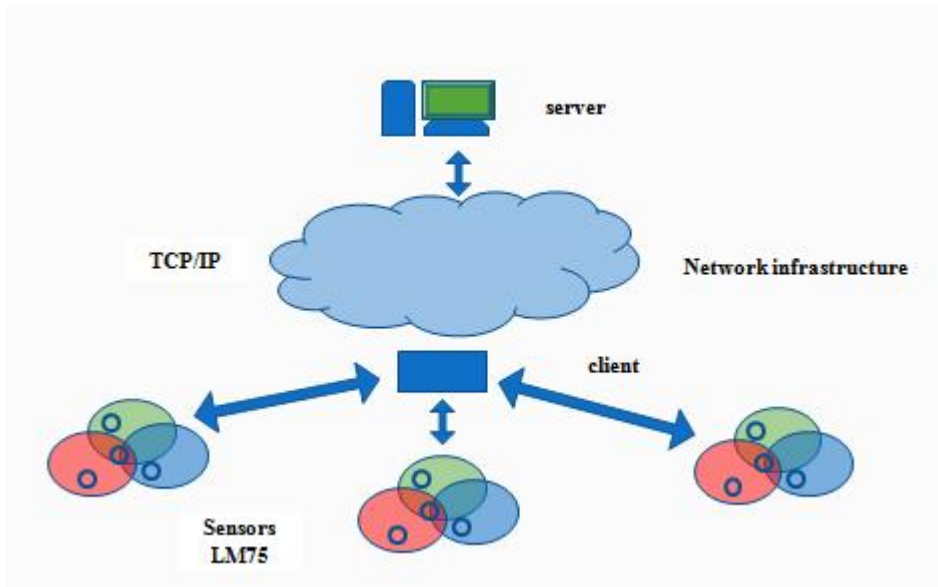


Figure 2.2. Sensor Networks Representation

2.6. DSN – DISTRIBUTED SENSOR NETWORK

A distributed sensor network (DSN) is a collection of sensors as Temperature sensor used to distributed the pointer location to used it or geographical location specified on most of over an environment in order to collect data of temperature reading to send it to the server controller and save it to large data base. And the distributed computing used to solve the error reading and error when transmission data in the middle Transmission user and distributed problem solving are used in the distributed sensor network in the order to take the result information of data as Relevant information collected and save it in data base after processing it and appropriate inferences. And when used the DSN in this kind of data fusion o reduce the failure to in the individual devices action in this case, is working on a collection of number of results and show results with best accuracy and selected from among the group [6].

most signals can the thermal network sensing multiple I mean sensing temperature and use of remote in multiple contexts and returned consists of several devices are connected with each other to show the best results, which is available including wired and wireless and by restrictions in computing power, telecommunications, and energy reserves necessary. This means that all sensors consist of devices with restricted resources and strongly, for example, commonly used in military research and communication devices wired. This hardware sensor has the ability reasonable and more importantly, may not be energy limited or restricted communication abilities [6].

2.7. SENSORS

Typically a sensor is composed of components that sense the environment, process the data, and communicate with other sensors/computers and control the output result by server and save all the data in data base designed by access 2003. A sensor responds to a physical stimulus, such as heat, light, sound, or pressure, and produces a measurable electrical signal. Thus a sensor with its own sensing device, a memory, and a processor can typically be programmed with a high-level programming language in recent years it was observed taking most of the focus on an important aspect in the maintenance of electronic devices and systems as especially sensitive and preserve appropriate temperature and have control of the high and low temperatures develop sensors and an example as we used the LM75 sensor who has the ability to feel the high and low temperature as a very high level and can work with used control it in 8 Data bus in the same bus used [6].

The sensing devices can range from nano-sensors to micro- and mega-sensors and all of them sensors work in the same speed when connected with controller devices in the baud rate is 9600 bps. And could refer to the sensor system such as a mote, which may have more than one physical sensor, its memory, processor, and other associated circuitry. Figure 2.3 shows distributed sensor architecture and various components [6].

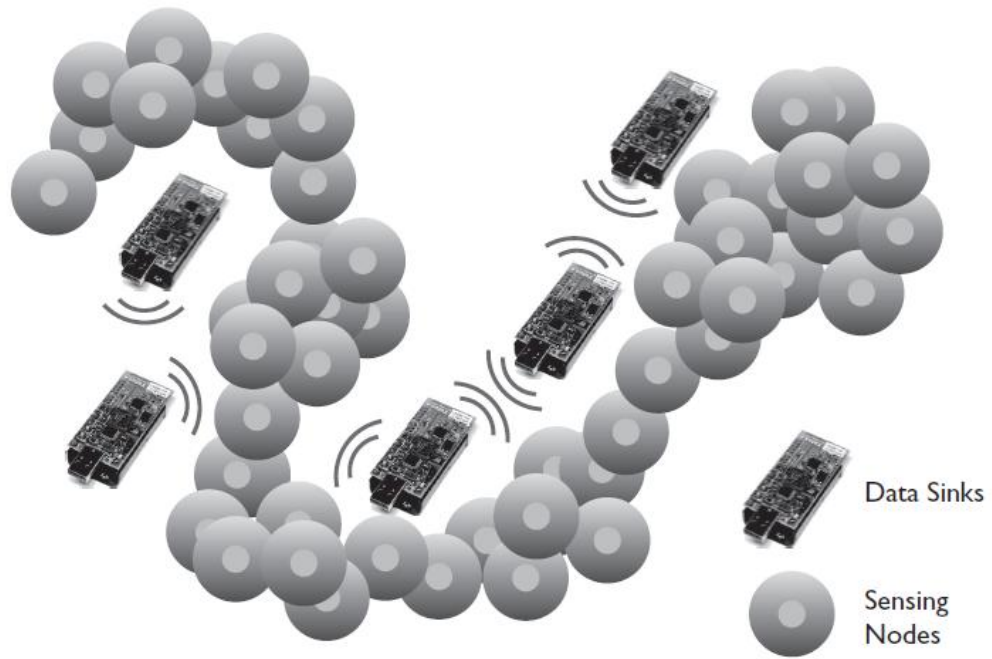


Figure 2.3. Networking structure of a distributed sensor network.

2.8. SYSTEM OF SENSOR NETWORKS COMMUNICATION

Figure 2.4 explains how the system works in the (software and hardware) side for all of them the server PC and client PC.

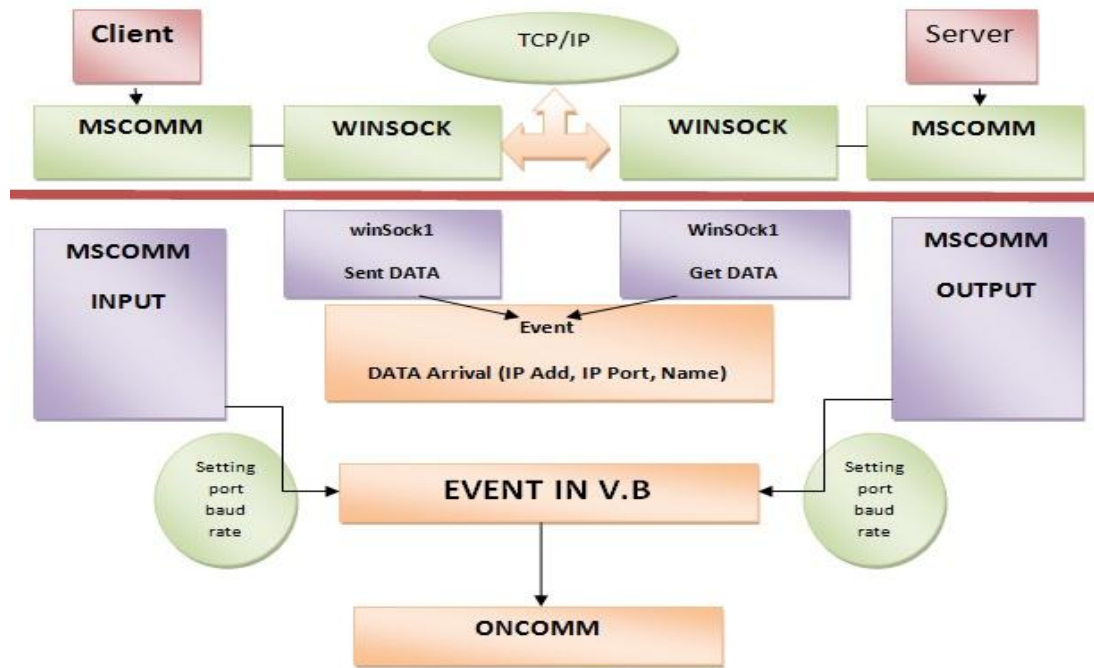


Figure 2.4. System of Sensor Networks

2.9. DIJSTRA'S ALGORITHM

When used this algorithm in the server side after applying that in Visual Basic program to take the best path as Shortest Path between the server and client. And see that in the example.

2.9.1. Example of Dijkstra's Algorithm

Dijkstra's algorithm can be carried out more easily if keep track of the necessary information on tables. These illustrate in this graph constructing an optimal A-rooted spanning tree in this graph as shown in Figure 2.5. [8].

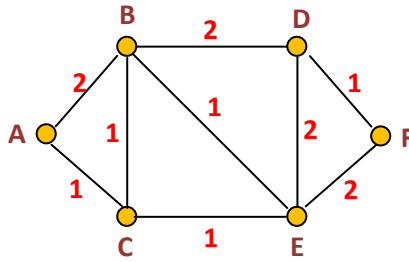


Figure 2.5.Example of Dijkstra's algorithm Topology.

Clearly the first edge is AC. The first tree, with vertex labels, is this one shown in Figure 2.6. [8].

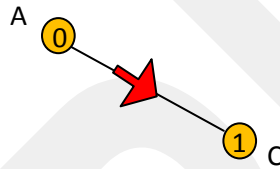


Figure 2.6. "AC" check path.

In which the first column shows the vertex labels at vertices already reached, and the remaining columns show the weights of all edges that are candidates for inclusion on the next step. By adding each vertex label to all weights in the same row, and obtain the values that have to be compared in order to determine which edge to add. This shows that any one of the edges "AB", "CB", or "CE" can be added on the next step as shown in Figure 2.7. [8].

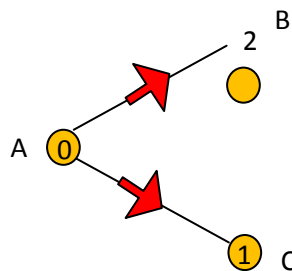


Figure 2.7. "AB" Path.

The remaining steps of the algorithm are shown in Figure 2.8, with the circled entry indicating the selected edge and the new vertex label on each step [8].

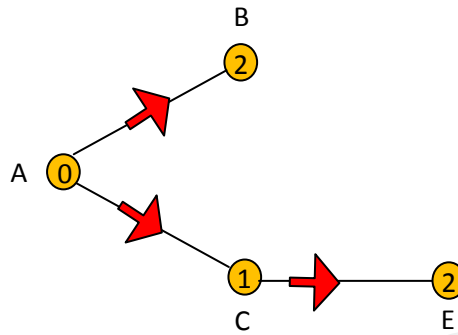


Figure 2.8. "CE" path.

As constructed in table above, the selected node from "B" is "D", and be shown in Figure 2.9. [8].

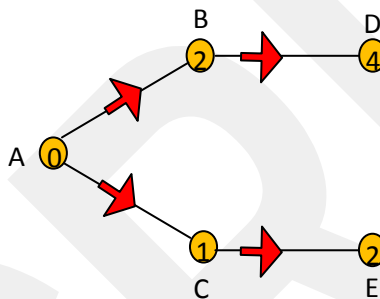


Figure 2.9. The "BD" path.

Finally, the shortest path from node "A" to "F" is get through "C" and then "E" and will be finally "A, C, D, and F" as shown in Figure 2.10 [8].

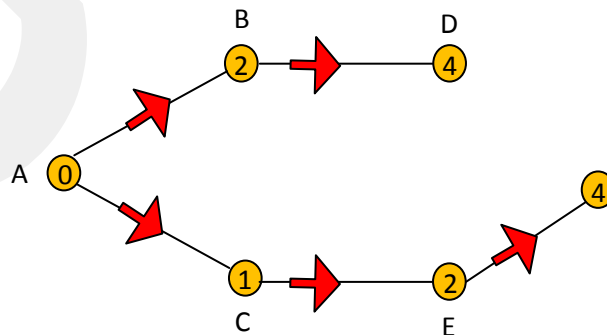


Figure 2.10. The "EF" path.

2.10. COMMUNICATION CONTROL IN VISUAL BASIC

The Microsoft Communications (MSComm) Control allows to control devices by using Visual Basic programs to transmit and receive data across a serial port or modem (also known as a COM port), much like a terminal emulation application. Before using the control in a program, should add the Mscomm object to the Toolbox [7].

To set up the MS Comm Control for use with modem, need to set these properties [7]:

* **CommPort.** An integer that specifies the COM port to which modem is attached—for example, 2 for COM2.

* **Settings.** A string that specifies the baud rate and parity settings.

These two properties can be set at design time or at the beginning of program in the Load event or Sub Main procedure. If have a modem on COM3, for example, the statements read as follow [7]:

```
Form1.MSComm1.CommPort = 3  
Form1.MSComm1.Settings = "9600,N,8,1"
```

2.10.1. Receiving and Transmitting Data

When a serial port is opened, the program creates receive and transmit buffers. To work with these buffers, the communications control supports a number of properties that can be set at design time using the control's property pages[7].

The InBufferSize and OutBufferSize properties hold the size of the input and output buffers, and the RThreshold and SThreshold properties set or return the number of characters that are received into the receive and transmit buffers before the OnComm event is fired (this event is used to monitor changes in communications states). To establish a connection, should first set the communications control's CommPort property to the serial port's number, the Settings property to the protocol

settings (for example, "9600,N,8,1"), and set the Port Open property to True. To start dialing, send the appropriate commands to modem [7].

Sending Data

To actually send data, using the Output property. That can either send data to modem or to the other computer [3].

```
MSComm1.Output = "hello world" & vbCr
```

Reading Data

To read data from an OnComm event occurs. In the OnComm event handler, can use the CommEvent property to determine what happened. For example, when CommEvent is equal to comEvReceive, a receiving data and can use the Input property. Here we fill a buffer with data that the communications control has received [3]:

```
Private Static Sub MSComm1_OnComm()  
Select Case MSComm1.CommEvent  
Case comEvReceive  
Dim Buffer As Variant  
Buffer = MSComm1.Input
```

Setting the InputLen property to some value means get that number of bytes when use the Input property (if those bytes are available). Setting InputLen to 0 makes the communications control read the entire contents of the receive buffer when use Input. The EOFEnable property is used to indicate when an End Of File (EOF) character is found in the data input. If set this property to True, it makes data input stop (and the OnComm event fire) when the EOF is encountered [3].

2.11. DATA BASE

— Data base work in MDB file and building by access 2003 and connection with the devices by two components in Visual Basic program :

1- Microsoft ADO Data Control 6.0.

2-Microsoft Data Grid Control 6.0.

GCCRIS

Chapter III

DESIGN AND APPLICATION OF PROPOSED WORK

3.1. OVERVIEW

The purpose of system is to collect and store environmental data for later processing by the server and send these data to client for monitoring. This is a mix of both PC and embedded system software, as well as embedded system hardware.

This system proposed contains both software and hardware design. It relied on networking infrastructure to provide communications between the Client and Server components. Besides that, the design application analysis the shortest path among nodes based on Dijkstra's algorithm as given in chapter 2. The focus of this system was to selectively present the collected environmental data to the end user in a graphical manner by using Visual Basic 6.0.

3.2. BLOCK DIAGRAM DESCRIPTION

There are four major phases implemented in our proposed system, as shown in Figure 3.1.

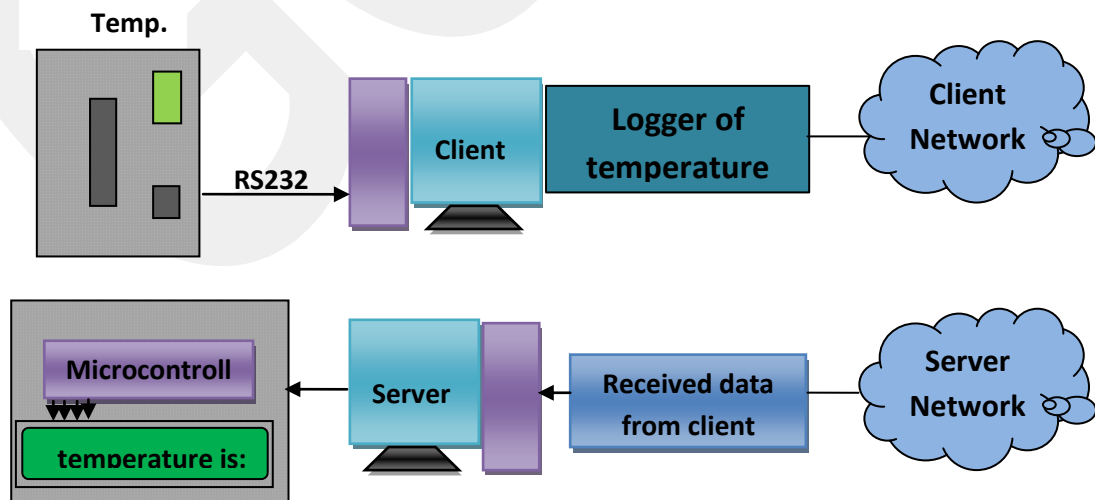


Figure 3.1. Main Block Diagram Of Proposed Work.

3.3. STEPS OF PROCESSING

The proposed design system has much functionality, and these functionalities will be described as following:

Step 1: Network Setup

First of all, the communication among the stations and server (monitoring base) must be setup. We've considered determined the both of client and server as a PCs and the connection between them using TCP/IP socket. The IP address and IP port has been signed, in client side settings of TCP/IP is (IP address: 192.168.1.2, IP Port No. 200) and in server side the settings are (IP address: 192.168.1.1, IP Port No. 200). These settings have been set by programming language, in our system we've used Visual Basic 6.0, and the components added to open the network socket is called (WinSock.ocx). The following statements used to set the winsock [7].

Winsock1.RemoteHost = IP address

Winsock1.LocalPort = IP port

Step 2: Network Connection

The communication network model used in our system is client/server model. This model refers to use two processes, one of the processes is called client and connect to other process called server. The establishment of connection needs the client to set the IP address and port no. of the server.

The socket on server side has to be ready in listening mode for any connection request from client(s). Any client send request to server asks to connection, the server will send an acceptance (if need to) to the client.

Step 3: Send and Receive Data in Network

After the connection has been established, the data is ready to be sending and received in both sides (client and server). When the data be sent from client side, on server side, the same data arrival on socket, which mean the data has been sent by sender, will save this data as a message as the statement below [7]:

Winsock1.GetData strData

strData is a variable contains the data which is sent it be client(s).

Step 4: Circuit Description Interfaced to Client

The microcontroller AT89C51 has an UART built-in which offered a serial communication. A serial port used as a physical interface to establish data transfer between computer and an external hardware or device. The type of transferring data is come to be send and receive data as bit by bit.

The voltage level of RS232 is not compatible with TTL logic to communicate and transceiver data between serial port and microcontroller. Therefore, voltage converter needed to communicate an RS232 to microcontroller, and the component used here is MAX232.

The simplest connection between a PC and microcontroller requires a minimum of three pins, RxD (receiver, pin2), TxD (transmitter, pin3) and ground (pin5) of the serial port of computer.

TxD pin of serial port connects to RxD pin of controller via MAX232. And similarly, RxD pin of serial port connects to the TxD pin of controller through MAX232.

MAX232 has two sets of line drivers for transferring and receiving data. The line drivers used for transmission are called T1, where as the line drivers for receiver are designated as R1. The connection of MAX232 with computer and the controller of circuit diagram shown in the Figure 3.2.

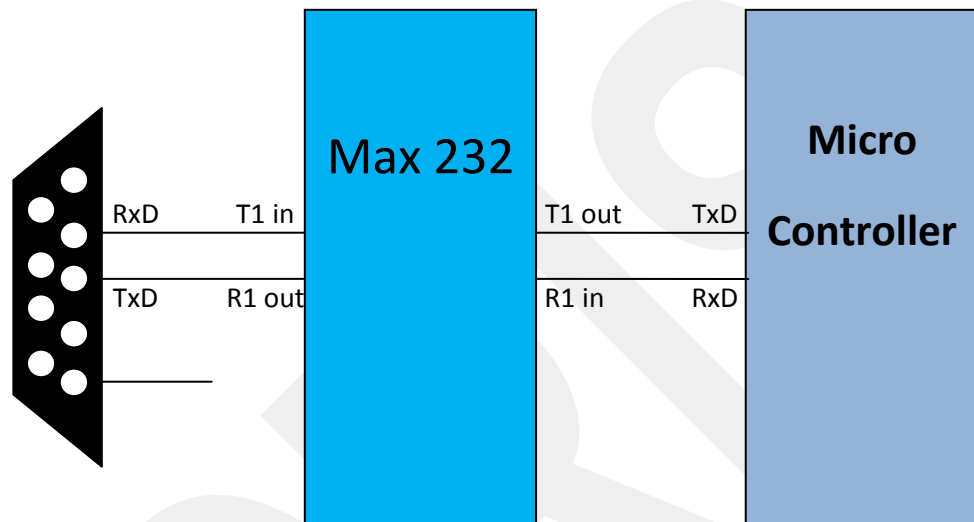


Figure 3.2. Serial Port Pin Configuration between Microcontroller and Serial Port.

An important parameter considered while interfacing serial port is the Baud rate which is the speed at which data is transmitted serially. It is defined as number of bits transmitted or received per second. It is generally expressed in bps (bits per second). AT89C51 microcontroller can be set to transfer and receive serial data at different baud rates using software instructions. In this project baud rate 9600bps is used. One of the important components in circuit design of client is temperature sensor, and the type of temperature sensor proposed in our design is LM75 which has 8bit and digital signal. The communication with the LM75 is through the I2C interface which is considered as a 2-wire communication protocol used to communicate serially.

Step 5: Circuit Description Interfaced to Server

This circuit design is different from the circuit interfaced to client that because this circuit design working only to received data from server PC. As discussed in step (4) regarding the serial port its will be same in server side. One of the components in this circuit is LCD, and the type of the LCD has a parallel interface. The interface consists of the following pins:

- **Register Select (RS):** this pin controls the LCD's memory to read/write data to and from. .
- **Read/Write (R/W):** Ability to change mode to reading mode or writing mode.
- **Enable (EN):** Enables writing to the registers.
- **8 data pins (D0 -D7):** These pins has two states, high or low state which determined the bits will be writing or read to/from register.

There's also a display contrast pin (Vo), power supply pins (+5V and Gnd).The process of controlling the display involves putting the data that form the image of needed to be display into the data registers, then putting instructions in the instruction register.

When data be transferred from PC via serial port to electronic circuit, the microcontroller will received this data, and same time, till pass this data to LCD to be displayed.

3.4. CLIENT SIDE SYSTEM

The Client side system contents of component which are necessary external to the development. That is, any computer with a network could be a Client. It served

only as a user interface to the data system. The system consist application and hardware. The application has been designed using Visual Basic 6.0 to control the socket of network using TCP/IP method and control the interface circuit design via serial port shown in Figure 3.3.

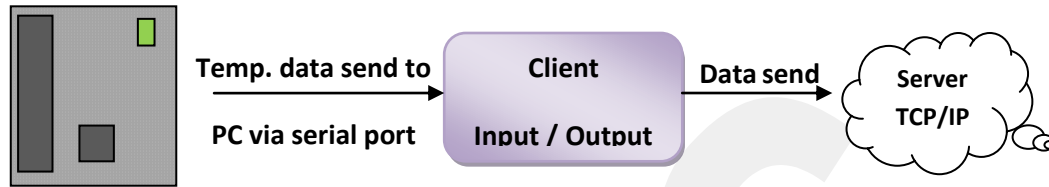


Figure 3.3. Client main component.

In client side initialization process of communication be taken, whereas the server responds to incoming client requests. The protocol used is TCP/IP that can be used for the communication link. When a client connects to listening sockets, the server receives a notification from Winsock, accepts the connection, and begins to dispatch and intercept messages to and from the new client.

The setting of client must know the Server's name, IP address, and the port number on which the server will be listening, as shown in flowchart in Figure 3.4.

In client side, the hardware design connect to client PC and device used to receive data from temperature sensors and to for post process analysis. The device consists of temperature sensor(s), microcontroller as processor of data, voltage regulator, and TTL/Serial converter, as shown in Figure 3.5.

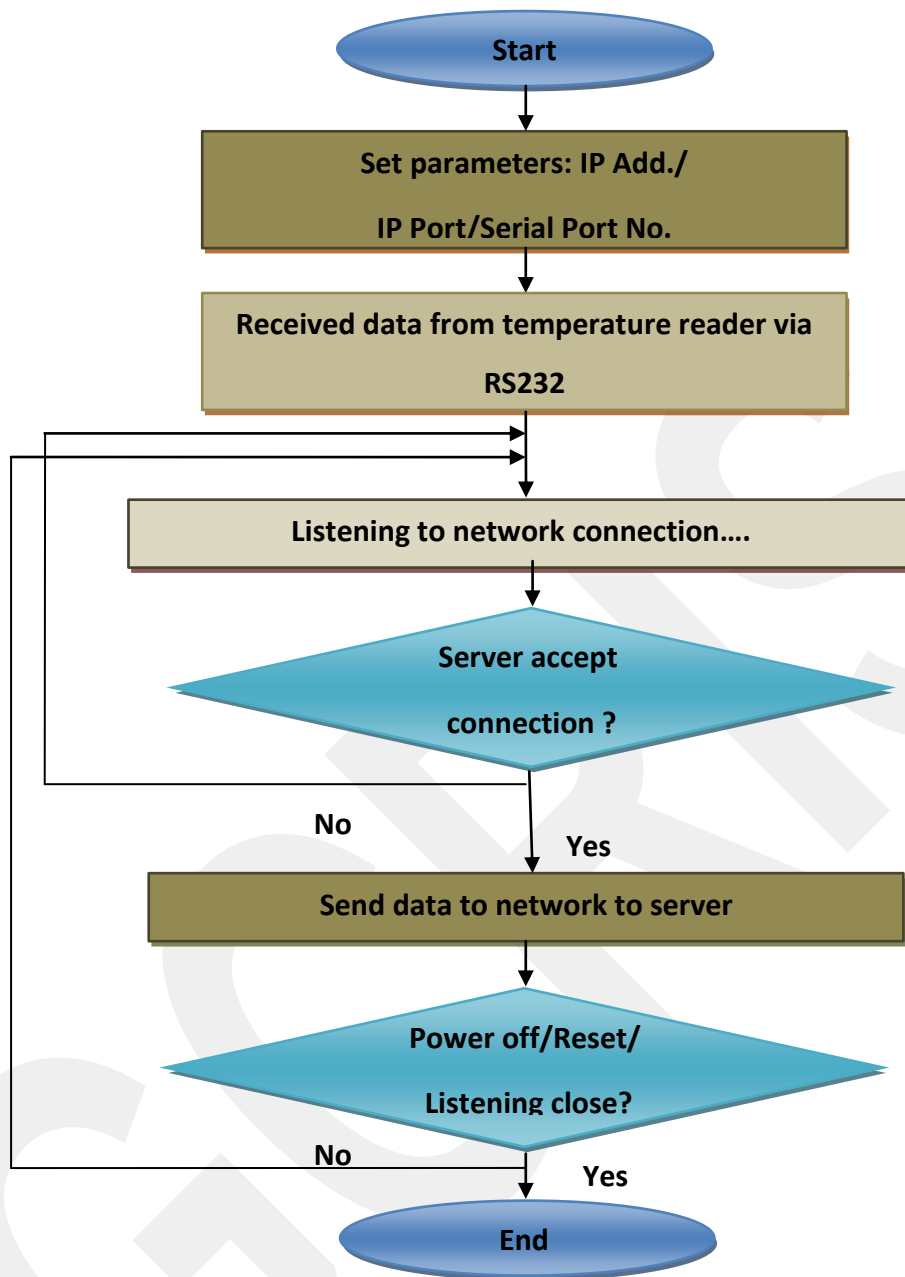


Figure 3.4. Client Flowchart on Network Data Transceiver.

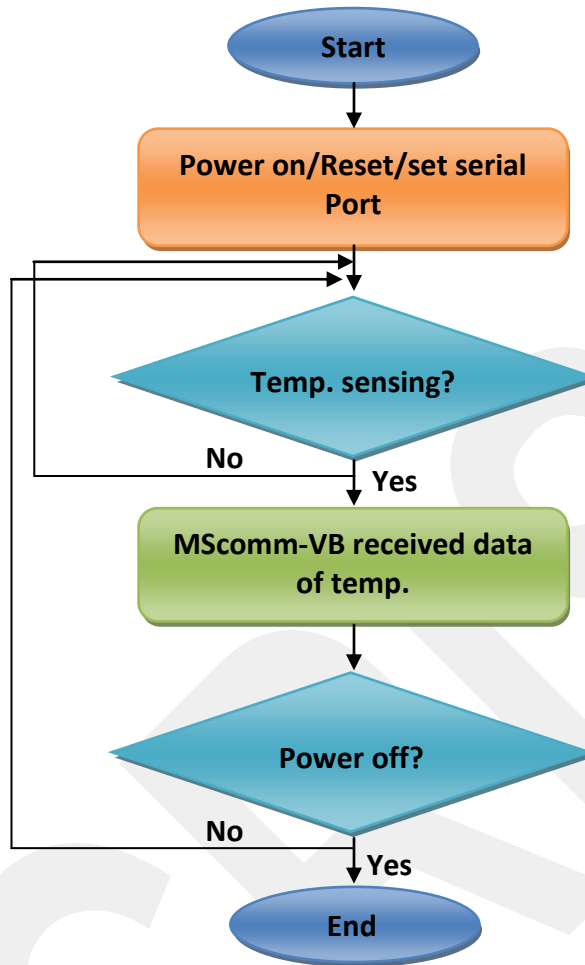


Figure 3.5. Device Interfaced on Client Side.

3.5. SERVER SIDE SYSTEM

The server side system has been designed to monitor the events of temperature on client(s) side. This system consists of application and hardware design. The application has two main processes, first control the socket of network and other process is a control the interface circuit design. When application request came in, the network server executes the application, which retrieved data from the database, processes it, and returns a data that the server transmitted to the Client as shown in Figure 3.6.

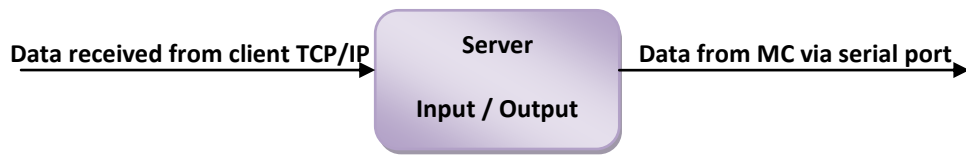


Figure 3.6. Server Main Component.

3.6. HARDWARE DESIGN

In this work, we've designed two electronic circuits, one of them belong client side, and other one belong to the server side. Below will discussed each of them starting from the client side.

3.6.1. Electronic Circuit Description in Client Side

As discussed in this chapter, there are two electronic circuits in both client and server system. The Figure 3.7. Shown the circuit schematic of client.

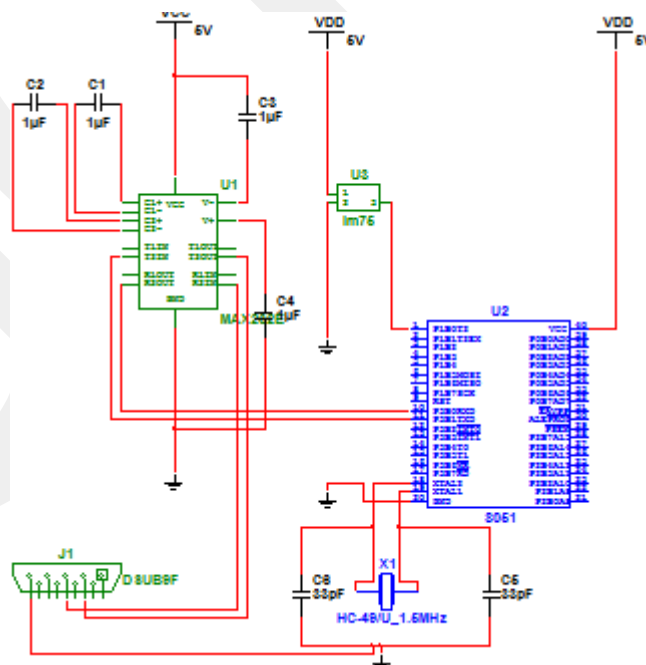


Figure 3.7.Schematic Circuit in Client Side.

The 2-wire communications protocol used to communicate serially with various types of devices similarly configured. The LM75, designed as a ‘slave’ device, can be configured through the I2C interface to alert, through various methods, the PC system that an over temperature condition has happened.

To control the LM75, first, need to read which address to set and let the value to and from. Second, write the value of address to memory to be read, finally start reading the value from memory.

The value will be a temperature, and this value will be ready to send from a microcontroller to serial port through TTL/Serial converter (MAX232). In same time, the program whom control the data on COM port (Visual Basic), will received the data, and display it on TEXTBOX as shown in Figure 3.8.

Finally the data will be send to server side via network using TCP/IP socket.

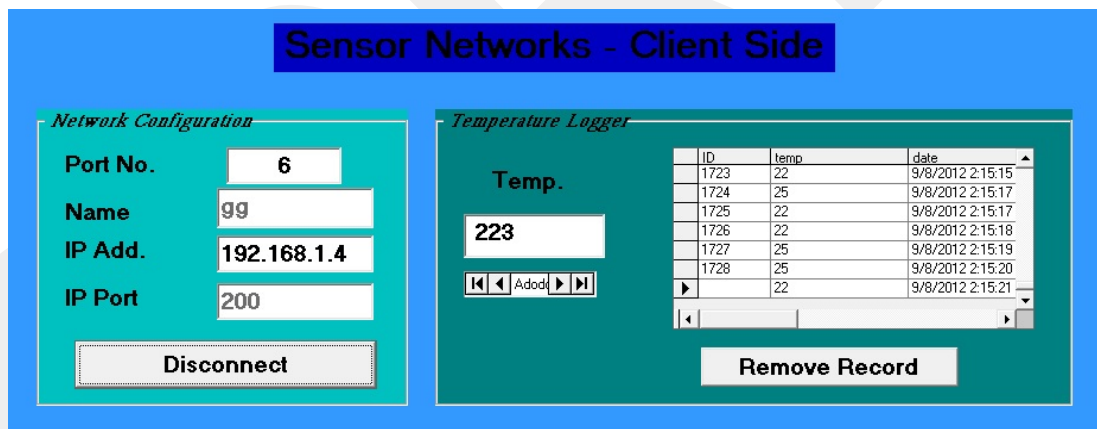


Figure 3.8.Interface Application Design in Client Side.

In server side, the design of electronic circuit has been modeled to received the data from client via TCP/IP socket, the application received the data that determined the temperature and forward this data to electronic circuit that interfaced to server PC as shown in Figure 3.9.

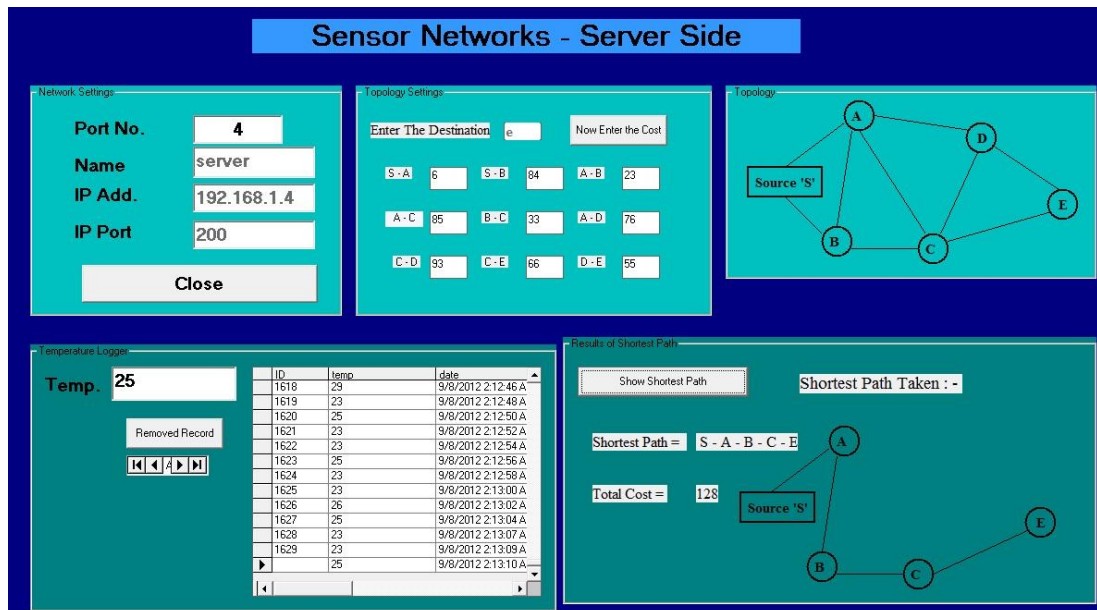


Figure 3.9. The User Interface of the Server Side Application.

The microcontroller will received the data from PC, and send it directly to LCD to be displayed as shown in schematic design in Figure 3.10.

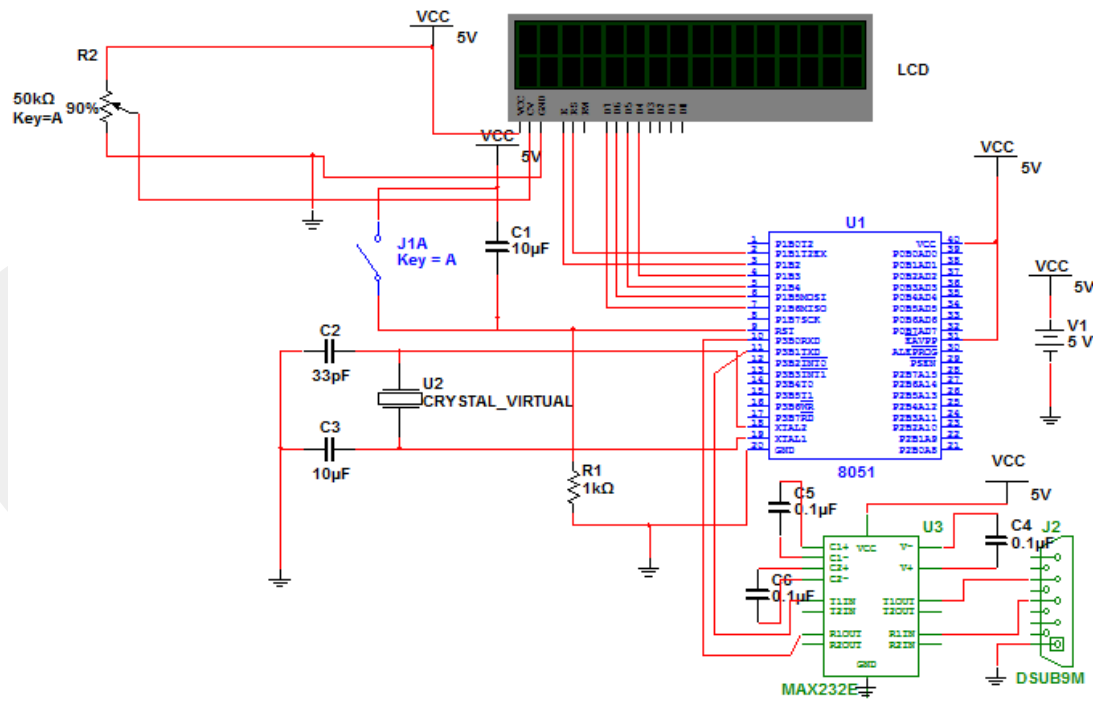


Figure 3.10. Schematic Design of Server Side.

CHAPTER IV

EXPERIMENTAL RESULTS

We've purposed and present the design system of a wireless sensor network for monitoring the temperature status. The sensor network consists of client and server nodes for temperature sensing. These nodes communicate by used a TCP/IP network.

The temperature sensor (LM75) is connected to the ATML 89c51 microcontroller on input pins numbered 0.1 and 0.2 as shown in Figure 4.1.

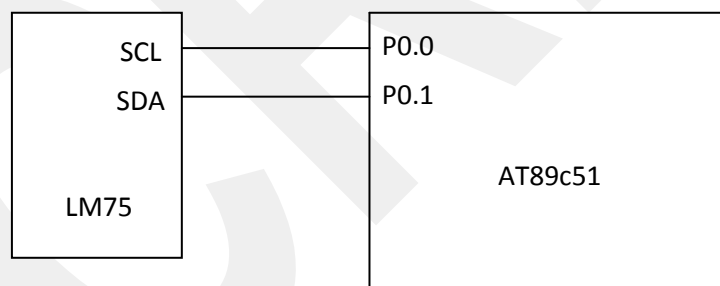


Figure 4.1. Pin Connection of I2C.

The electronic circuit interfaced to client PC start reads temperature through temperature sensor. The LM75 registers are accessible through the I2C port. This port is comprised of Pin 'SDA' (Serial Data) and Pin 'SCL' (Serial Clock). The interfaced electronic circuit has been snap shooting as shown in Figure 4.2.

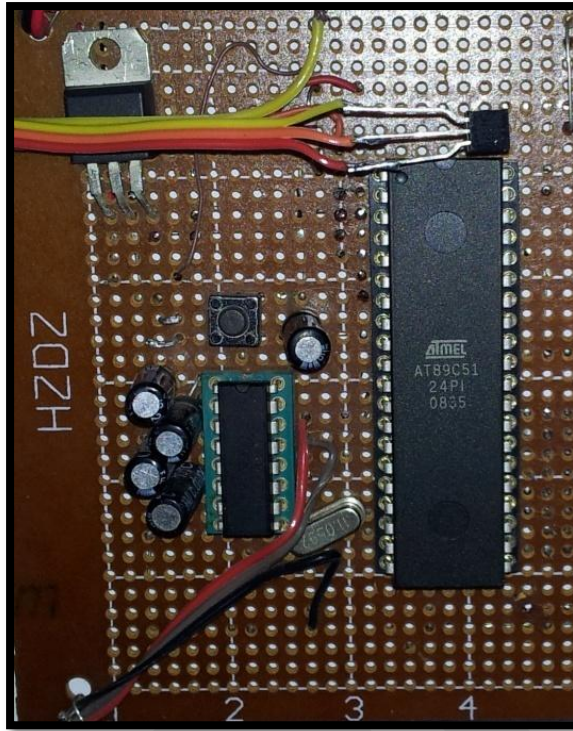


Figure 4.2. Electronic Circuit Interfaced to Client Side.

Other electronic circuit interfaced to server PC which implemented to get data from server PC via serial COM port and display the data directly to LCD as shown in Figure 4.3.



Figure 4.3. Electronic Circuit Interfaced To Server Side.

We deployed one sensor node far away from server (monitor). We monitor the temperature readings via a network communicate to monitor which is Server and receive the data from sensor and then monitor. The temperature sensor electronic circuit reads a temperature in current environment as shown in Table 4.1.

Table 4.1. Temperature Sensor Measurements on Various Environments

Environment	Normal Temp.	Time
Room	37	9:00
Room	38	9:30
Room	38.5	10:00
Office	36	9:00
Office	36.5	11:00
Office	37.5	02:00
Factory	38	10:00
Factory	38.5	11:00

Figure 4.4. Shows curves belong to the table above which shows temperature for each environment changed on time.

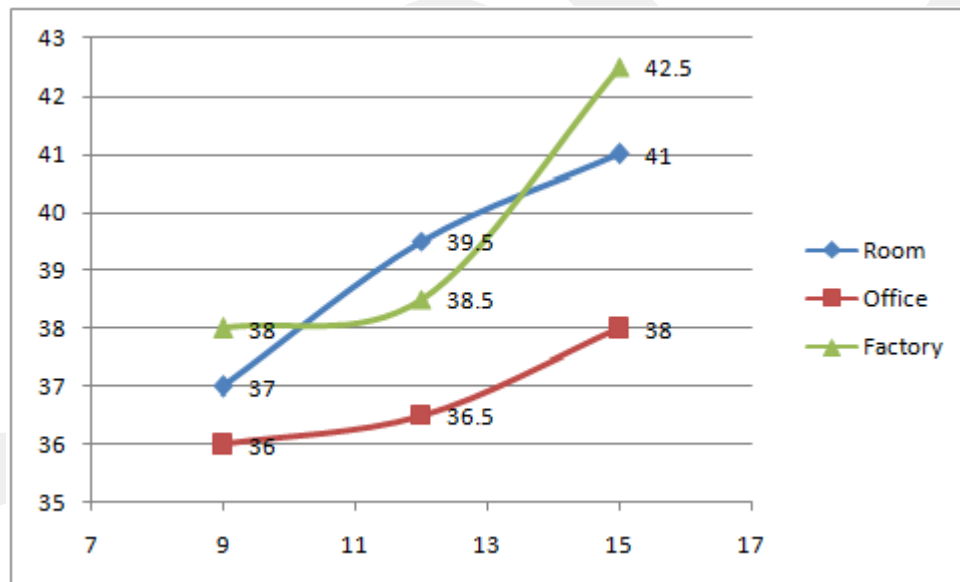


Figure 4.4. Temperature Changes Over Time in Various Environments.

We performed a test for temperature monitoring between 9am to 1pm. The system collected the temperature readings at specific sensor and got the temperature value, in addition to that, the value(s) of temperature directly saved in database file (.MDB), the database has been designed using Microsoft Access 2007, which contains two fields (temperature and date/time). In application design, we've connect to the database using ADODC component and the provider of database connection

was OLE JET 4.0. Finally, the data has been showed up in the form design using DATAGRID component and this one bring up the data via ADODC.

Another case study has been simulated in this research which focuses on issue how to find shortest path between source and destination node. For the same topology design we've specified the length and connection among nodes as describe in Table 4.2.

Table 4.2. Connections and lengths of The Designed Topology

Link	Length
(A, source)	10
(B,Source)	20
(Source, A)	10
(B,A)	15
(C,A)	10
(D,A)	4
(Source,B)	20
(A,B)	15
(C,B)	2
(A,C)	12
(B,C)	2
(D,C)	10
(E,C)	10
(A,D)	4
(C,D)	10
(E,D)	12
(C,E)	10
(D,E)	12

And all paths in the topology are shown Table 4.3.

Table 4.3. All Connections of The Designed Topology

	Source	A	B	C	D	E
Source	0	10	20	22	14	26
A	10	0	14	12	4	16
B	20	12	0	2	12	12
C	20	10	2	0	10	10
D	14	4	12	10	0	12
E	26	16	12	10	12	0

The purposed algorithm to find shortest path in this research is called Dijkstra’s algorithm. When try to find the shortest path from Source to destination node (E), we’ve found the following:

Total length is 26, and the shortest path is: (Source→ A→ D → E), and the path has been drawn as shown in Figure 4.5.

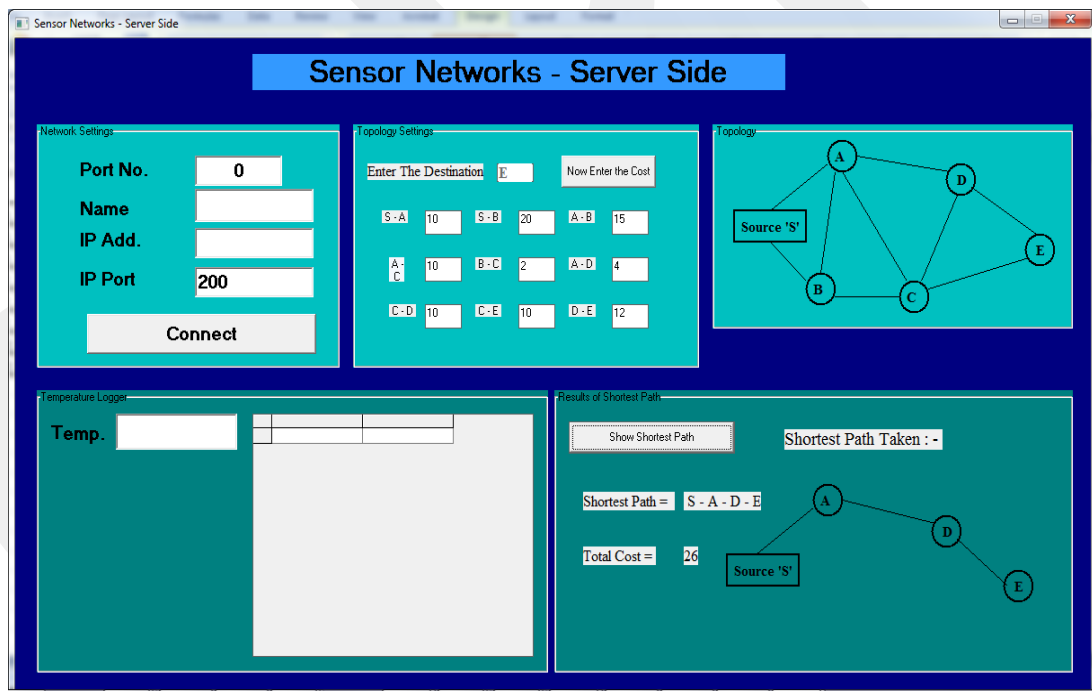


Figure 4.5. Shortest Path Drawn from Source to Node (E) .

CHAPTER V

CONCLUSION AND FUTURE WORK

5.1. CONCLUSION

In modern fire stations, several measurement points are taken in different parts of the station to make the fire station system work properly. Networking would make the measurement system easier. Furthermore, the system proposed is controlling one or several sensors are an attractive and cost efficient option to build the required measurement system. In this work, we developed a network sensor node for fire station monitoring by integrating a sensor with commercial sensors capable to measure temperature values. The system proposed was tested by deploying a simple sensor network into Fire station in Baghdad. We collected data to evaluate the network reliability and its ability to detect the fire and temperature, which typically exist in the client. We were also able to monitoring the sensors situations and show that the network can detect the differences temperatures caused by various disturbances, such as direct heat near the sensor or heat halted in sensor(s). In sensor network, it's an important to monitor the events on other nodes (clients), and one of these events, it's the temperature which considered as an critical issue especially on manufacturing, production line, secured places, and other places and sectors.

5.2. FUTURE WORK

We recommended for future work the following:

- Develop the system by sending the data of temperature to mobile based on GSM/SMS.
- Design a web site to display a temperature data for each sensor site/location.
- Design a robot to exploring sites has a high temperature.

REFERENCES

- 1- Jorge Chang, Amarnath Kolla, and Somya Kapoor; "**Security Threats in Wireless Sensor Network**"; 2006.
- 2- ."**Fire monitoring in coal mines using Wireless Sensor Networks**; Roy, P.; Bhattacharjee, S.; Ghosh, S.; Misra, S.; Obaidat, M.S 2011”.
- 3- Jan Axelson; "**Serial Port Complete: Programming and Circuits for Rs-232**"; lakeview research; 1998.
- 4- Behrouz A. Forouzan; "**Data Communication and Networking**"; McGraw-Hill, Forouzan Networking Series; 4th Edition, 2007.
- 5- Shashi Phoha, Thomas LaPorta, Christopher Griffin; "**Sensor Network Operations**"; IEEE Press; 2006.
- 6- S. Sitharaa Iyengar, Nandan Parameshwaran, Vir V. Phoha, N. Balakrishnan, Chuka D. Okoye; "**Fundamentals of Sensor Network Programming, Applications and Technology**"; IEEE Press-Wiley; 2011.
- 7- Brian Siler and Jeff Spotts; "**Using Visual Basic 6.0, Special Edition**"; Que publishing; 1998.
- 8- Thomas H. Cormen, Charles E. Leiserson, Ronld L. Rivest, Clifford Stein; "**Introduction to Algorithms**"; 3rd ed; MIT Press; 2009.

APPENDIX A

CLIENT APPLICATION – VISUAL BASIC 6.0

```
Dim s As Integer
```

```
Private Sub Command1_Click()  
Adodc1.Recordset.Delete  
End Sub
```

```
Private Sub Form_Load()  
Call optConnect_Click(0)  
MSComm1.CommPort = 1  
MSComm1.Settings = "9600,N,8,1"  
txtNick.Text = ""  
txtIP.Text = ""  
txtIP.Enabled = True  
cmdConnect.Caption = "Connect"
```

```
End Sub
```

```
Private Sub MSComm1_OnComm()
```

```
ms
```

```
End Sub
```

```
Sub ms()
```

```
s = s + 1
```

```
Dim a
```

```
On Error Resume Next
```

```
a = Asc(MSComm1.Input)
```

```
If a >= 150 Or a <= 100 Then Exit Sub
```

```
' txtMessage.Text = a
```

```
' If s = 4 Then
```

```
txtMessage.Text = a - 100
```

```
' End If
```

```
If s = 8 Then s = 0
```

```
End Sub
```

```
Private Sub txtMessage_Change()
```

```

Adodc1.Recordset.AddNew

Adodc1.Recordset.Fields(1).Value = txtMessage
Adodc1.Recordset.Fields(2).Value = Now
For I = 0 To 30000000
Next
    txtStatus.Text = ""
    If Winsock1.State = sckClosed Then
        MsgBox "Please connect first.", vbCritical
        txtMessage.Text = ""
        txtMessage.SetFocus
        Exit Sub
    End If

    If txtMessage.Text <> "" Then

        Winsock1.SendData txtMessage.Text

        txtStatus.Text = txtMessage.Text

    End If

End Sub

Private Sub Winsock1_Close()
    cmdConnect_Click
End Sub

Private Sub Winsock1_ConnectionRequest(ByVal requestID As Long)
    If Winsock1.State <> sckClosed Then Winsock1.Close

    Winsock1.Accept requestID

End Sub

```

```
Private Sub Winsock1_DataArrival(ByVal bytesTotal As Long)
    Dim strData As String
```

```
Winsock1.GetData strData
```

```
txtStatus.Text = strData
```

```
End Sub
```

```
Private Sub cmdConnect_Click()
```

```
If cmdConnect.Caption = "Connect" Then ' Guest
```

```
txtPort.Enabled = False
```

```
txtNick.Enabled = False
```

```
optConnect(0).Enabled = False
```

```
optConnect(1).Enabled = False
```

```
cmdConnect.Caption = "Disconnect"
```

```
Else ' Disconnect or Close
```

```
txtPort.Enabled = True
```

```
txtNick.Enabled = True
```

```
optConnect(0).Enabled = True
```

```
optConnect(1).Enabled = True
```

```
cmdConnect.Caption = "Connect"
```

```
txtStatus.Text = txtNick
```

```
Winsock1.Close
```

```
Exit Sub
```

```
End If
```

```
Winsock1.RemoteHost = txtIP.Text
```

```
Winsock1.RemotePort = txtPort.Text
```

```
Winsock1.Connect
    txtStatus.Text = txtNick

MSComm1.CommPort = Text1

MSComm1.PortOpen = True
End Sub

Private Sub optConnect_Click(Index As Integer)
    Select Case Index
    Case 0:
        txtNick.Text = "Host"
        txtIP.Text = Winsock1.LocalIP
        txtIP.Enabled = False
        cmdConnect.Caption = "Listen"
    Case 1:
        txtNick.Text = ""
        txtIP.Text = ""
        txtIP.Enabled = True
        cmdConnect.Caption = "Connect"
    End Select
End Sub
```

APPENDIX B

SERVER APPLICATION – VISUAL BASIC 6.0

```
Option Explicit
Private Sub Command1_Click()
Text11.Enabled = False
Dim reply
If Text11.Text = "" Then
reply = MsgBox(" Kindly Enter The Exact Destination Node... ", vbExclamation, "
Wrong Entry !!! ") = vbOK
Text11.Text = ""
Text11.Enabled = True
Text11.SetFocus
End If
If Text11.Text = "a" Or Text11.Text = "A" Then
'text11.Text = "A" Or
Label38.Visible = True
Label38.Caption = " S - A "
Label39.Visible = True
Label39.Caption = " S - B "
Label40.Visible = True
Label40.Caption = " B - A "
Text2.Visible = True
Text3.Visible = True
Text4.Visible = True
Command3.Enabled = True
End If
If Text11.Text = "b" Or Text11.Text = "B" Then
'text11.Text = "B" Or
Label38.Visible = True
Label38.Caption = " S - B "
Label39.Visible = True
Label39.Caption = " S - A "
Label40.Visible = True
Label40.Caption = " A - B "
Text2.Visible = True
Text3.Visible = True
```

```
Text4.Visible = True
Command3.Enabled = True
End If
If Text11.Text = "c" Or Text11.Text = "C" Then
'text11.Text = "C" Or
Label38.Visible = True
Label38.Caption = " S - A "
Label39.Visible = True
Label39.Caption = " S - B "
Label40.Visible = True
Label40.Caption = " A - B "
Label11.Visible = True
Label11.Caption = " A - C "
Label2.Visible = True
Label7.Visible = True
Label2.Caption = " B - C "
Text2.Visible = True
Text3.Visible = True
Text4.Visible = True
Text5.Visible = True
Text6.Visible = True
Command3.Enabled = True
End If
If Text11.Text = "d" Or Text11.Text = "D" Then
'text11.Text = "D" Or
Label38.Visible = True
Label38.Caption = " S - A "
Label39.Visible = True
Label39.Caption = " S - B "
Label40.Visible = True
Label40.Caption = " A - B "
Label11.Visible = True
Label11.Caption = " A - C "
Label2.Visible = True
Label7.Visible = True
Label2.Caption = " B - C "
Label2.Visible = True
Label3.Caption = " A - D "
```

```
Label3.Visible = True
Label4.Caption = " C - D "
```

```
Label4.Visible = True
Text2.Visible = True
Text3.Visible = True
Text4.Visible = True
Text5.Visible = True
Text6.Visible = True
Text7.Visible = True
Text8.Visible = True
Command3.Enabled = True
End If
If Text11.Text = "e" Or Text11.Text = "E" Then
'text11.Text = "E" Or
Label38.Visible = True
Label38.Caption = " S - A "
Label39.Visible = True
Label39.Caption = " S - B "
Label40.Visible = True
Label40.Caption = " A - B "
Label1.Visible = True
Label1.Caption = " A - C "
Label2.Visible = True
Label7.Visible = True
Label2.Caption = " B - C "
Label2.Visible = True
Label7.Visible = True
Label3.Caption = " A - D "
Label3.Visible = True
Label4.Caption = " C - D "
Label4.Visible = True
Label5.Caption = " C - E "
Label5.Visible = True
Label6.Caption = " D - E "
Label6.Visible = True
Text2.Visible = True
Text3.Visible = True
```

```
Text4.Visible = True
Text5.Visible = True
Text6.Visible = True
Text7.Visible = True
Text8.Visible = True
Text9.Visible = True
Text10.Visible = True
Command3.Enabled = True
End If
End Sub
```

```
Private Sub Command2_Click()
Form1.Refresh
Form1.No_Node
Text11.Enabled = True
Text11.Text = ""
Text11.SetFocus
'Call No_Node
End Sub
```

```
Private Sub Command3_Click()
If Text11.Text = "A" Or Text11.Text = "a" Then
Call Node_A
End If
If Text11.Text = "B" Or Text11.Text = "b" Then
Call Node_B
End If
If Text11.Text = "C" Or Text11.Text = "c" Then
Call Node_C
End If
If Text11.Text = "D" Or Text11.Text = "d" Then
Call Node_D
End If
If Text11.Text = "E" Or Text11.Text = "e" Then
Call Node_E
End If
End Sub
```

```
Private Sub Command4_Click()  
Unload Me  
End Sub
```

```
Private Sub Text1_KeyPress(KeyAscii As Integer)  
Dim result  
If KeyAscii >= 48 And KeyAscii <= 59 Then  
result = MsgBox(" Kindly Enter only Characters ", vbExclamation, " Wrong Entry  
!!! ") = vbOK  
KeyAscii = 0  
End If  
End Sub
```

```
Public Function Node_A()
```

```
If Text11.Text = "A" Or Text11.Text = "a" Then  
If Val(Text2.Text) <= (Val(Text3.Text) + Val(Text4.Text)) Then  
Label32.Visible = True  
Label33.Visible = True  
Label34.Caption = Label38.Caption  
Label35.Caption = Val(Text2.Text)  
Label19.Visible = True  
Line16.Visible = True  
Shape3.Visible = True  
Shape10.Visible = True  
Label26.Visible = True
```

```
Label27.Visible = True  
Else  
Label32.Visible = True  
Label33.Visible = True  
Label34.Caption = " S - B - A "  
Label35.Caption = Val(Text3.Text) + Val(Text4.Text)  
Label19.Visible = True  
Line15.Visible = True  
Line18.Visible = True  
Shape3.Visible = True  
Shape9.Visible = True
```

```

Shape10.Visible = True
Label26.Visible = True
Label27.Visible = True
Label28.Visible = True
End If
End If
End Function
Public Function Node_B()
If Text11.Text = "B" Or Text11.Text = "b" Then
If Val(Text2.Text) <= (Val(Text3.Text) + Val(Text4.Text)) Then
Label32.Visible = True
Label33.Visible = True
Label34.Caption = Label38.Caption
Label35.Caption = Val(Text2.Text)
Label19.Visible = True
Line15.Visible = True
Shape10.Visible = True
Shape9.Visible = True
Label26.Visible = True
Label28.Visible = True
Else
Label32.Visible = True
Label33.Visible = True
Label34.Caption = " S - A - B "
Label35.Caption = Val(Text3.Text) + Val(Text4.Text)
Label19.Visible = True
Line16.Visible = True
Line18.Visible = True
Shape3.Visible = True
Shape9.Visible = True
Shape10.Visible = True
Label26.Visible = True
Label27.Visible = True
Label28.Visible = True
End If
End If
End Function
Public Function Node_C()

```

```

If Text11.Text = "C" Or Text11.Text = "c" Then
Dim a1, b1, c1, d1
a1 = Val(Text2.Text) + Val(Text5.Text)
b1 = Val(Text3.Text) + Val(Text6.Text)
c1 = Val(Text3.Text) + Val(Text4.Text) + Val(Text5.Text)
d1 = Val(Text2.Text) + Val(Text4.Text) + Val(Text6.Text)
If a1 <= b1 And a1 <= c1 And a1 <= d1 Then
MsgBox ("s-a-c")
Label32.Visible = True
Label33.Visible = True
Label34.Caption = " S - A - C "
Label35.Caption = (Val(Text2.Text) + Val(Text5.Text))
Label19.Visible = True
Line14.Visible = True
Line16.Visible = True
Shape10.Visible = True
Shape1.Visible = True
Shape3.Visible = True
Label26.Visible = True
Label27.Visible = True
Label29.Visible = True
Else
If b1 < a1 And b1 <= c1 And b1 <= d1 Then
MsgBox ("s-b-c")
Label32.Visible = True
Label33.Visible = True
Label34.Caption = " S - B - C "
Label35.Caption = (Val(Text3.Text) + Val(Text6.Text))
Label19.Visible = True
Line13.Visible = True
Line15.Visible = True
Shape10.Visible = True
Shape1.Visible = True
Shape9.Visible = True
Label26.Visible = True
Label28.Visible = True
Label29.Visible = True
Else

```

```
If c1 < a1 And c1 < b1 And c1 <= d1 Then
MsgBox ("s-b-a-c")
Label32.Visible = True
Label33.Visible = True
Label34.Caption = " S - B - A - C "
Label35.Caption = (Val(Text3.Text) + Val(Text4.Text) + Val(Text5.Text))
Label19.Visible = True
Line14.Visible = True
Line15.Visible = True
Line18.Visible = True
Shape3.Visible = True
Shape1.Visible = True
Shape9.Visible = True
Shape10.Visible = True
Label26.Visible = True
Label27.Visible = True
Label28.Visible = True
Label29.Visible = True
Else
If d1 < a1 And d1 < b1 And d1 < c1 Then
MsgBox ("s-a-b-c")
Label32.Visible = True
Label33.Visible = True
Label34.Caption = " S - A - B - C "
Label35.Caption = (Val(Text2.Text) + Val(Text4.Text) + Val(Text6.Text))
Label19.Visible = True
Line13.Visible = True
Line16.Visible = True
Line18.Visible = True
Shape3.Visible = True
Shape1.Visible = True
Shape9.Visible = True
Shape10.Visible = True
Label26.Visible = True
Label27.Visible = True
Label28.Visible = True
Label29.Visible = True
End If
```

```
End If
End If
End If
End If
```

```
End Function
```

```
Private Sub Timer3_Timer()
```

```
Dim k
```

```
Dim t4
```

```
Dim rll
```

```
Dim sv
```

```
For k = 0 To 10
```

```
    If (t4 Mod 10 = k.ipaddr) Then
```

```
        MsgBox "nod(k).ipaddr" & nod(k).ipaddr
```

```
        MsgBox "nod(sv).ipaddr" & nod(sv).ipaddr
```

```
        MsgBox rll
```

```
        MsgBox sv
```

```
        k.Visible = True
```

```
        k.X1 = k.posy + 200
```

```
        MsgBox "rln(rll).X1" & rln(rll).X1
```

```
        k.Y1 = k.posx + 200
```

```
        MsgBox "rln(rll).y1" & rln(rll).Y1
```

```
        k.X2 = k.posy + 200
```

```
        MsgBox "rln(rll).X2" & rln(rll).X2
```

```
        k.Y2 = k.posx + 200
```

```
        MsgBox "rln(rll).y2" & rln(rll).Y2
```

```
        rll = k + 1
```

```
        sv = k
```

```
    End If
```

```
Next
```

```
End Sub
```

```
Public Function Node_D()
```

```

If Text11.Text = "D" Or Text11.Text = "d" Then
Dim a2, b2, c2, d2
a2 = Val(Text2.Text) + Val(Text7.Text)
b2 = Val(Text3.Text) + Val(Text6.Text) + Val(Text8.Text)
c2 = Val(Text2.Text) + Val(Text5.Text) + Val(Text8.Text)
d2 = Val(Text3.Text) + Val(Text4.Text) + Val(Text7.Text)
If a2 <= b2 And a2 <= c2 And a2 <= d2 Then
Label32.Visible = True
Label33.Visible = True
Label34.Caption = " S - A - D "
Label35.Caption = (Val(Text2.Text) + Val(Text7.Text))
Label19.Visible = True
Line11.Visible = True
Line16.Visible = True
Shape10.Visible = True
Shape7.Visible = True
Shape3.Visible = True
Label26.Visible = True
Label27.Visible = True
Label30.Visible = True
Else
If b2 < a2 And b2 <= c2 And b2 <= d2 Then
Label32.Visible = True
Label33.Visible = True
Label34.Caption = " S - B - C - D "
Label35.Caption = (Val(Text3.Text) + Val(Text6.Text) + Val(Text8.Text))
Label19.Visible = True
Line13.Visible = True
Line15.Visible = True
Line10.Visible = True
Shape10.Visible = True
Shape7.Visible = True
Shape9.Visible = True
Shape1.Visible = True
Label26.Visible = True
Label28.Visible = True
Label29.Visible = True
Label30.Visible = True

```

```

Else
If c2 < a2 And c2 < b2 And c2 <= d2 Then
Label32.Visible = True
Label33.Visible = True
Label34.Caption = " S - A - C - D "
Label35.Caption = (Val(Text2.Text) + Val(Text5.Text) + Val(Text8.Text))
Label19.Visible = True
Line14.Visible = True
Line16.Visible = True
Line10.Visible = True
Shape10.Visible = True
Shape7.Visible = True
Shape3.Visible = True
Shape1.Visible = True
Label26.Visible = True
Label27.Visible = True
Label29.Visible = True
Label30.Visible = True
Else
If d2 < a2 And d2 < b2 And d2 < c2 Then
Label32.Visible = True
Label33.Visible = True
Label34.Caption = " S - B - A - D "
Label35.Caption = (Val(Text3.Text) + Val(Text4.Text) + Val(Text7.Text))
Label19.Visible = True
Line11.Visible = True
Line15.Visible = True
Line18.Visible = True
Shape10.Visible = True
Shape7.Visible = True
Shape3.Visible = True
Shape9.Visible = True
Label26.Visible = True
Label27.Visible = True
Label28.Visible = True
Label30.Visible = True
End If
End If

```

```
End If
End If
End If
```

```
End Function
```

```
Private Sub Timer4_Timer()
```

```
' LAR Routing
```

```
Dim k
```

```
Dim t4
```

```
Dim rll
```

```
Dim sv
```

```
For k = 0 To 10
```

```
    If (t4 Mod 10 = k.ipaddr) Then
```

```
        "msgbox "nod(k).ipaddr" & nod(k).ipaddr
```

```
        "msgbox "nod(sv).ipaddr" & nod(sv).ipaddr
```

```
        "msgbox rll
```

```
        "msgbox sv
```

```
        k.Visible = True
```

```
        k.X1 = k.posy + 200
```

```
        "msgbox "rln(rll).X1" & rln(rll).X1
```

```
        k.Y1 = k.posx + 200
```

```
        "msgbox "rln(rll).y1" & rln(rll).Y1
```

```
        k.X2 = k.posy + 200
```

```
        "msgbox "rln(rll).X2" & rln(rll).X2
```

```
        k.Y2 = k.posx + 200
```

```
        "msgbox "rln(rll).y2" & rln(rll).Y2
```

```
        rll = k + 1
```

```
        sv = k
```

```
    End If
```

```
Next
```

```
End Sub
```

```
Public Function Node_E()
```

```
If Text11.Text = "E" Or Text11.Text = "e" Then
```

```

Dim a3, b3, c3, d3, e3, f3
a3 = Val(Text2.Text) + Val(Text7.Text) + Val(Text10.Text)
b3 = Val(Text3.Text) + Val(Text6.Text) + Val(Text9.Text)
c3 = Val(Text2.Text) + Val(Text5.Text) + Val(Text8.Text) + Val(Text10.Text)
d3 = Val(Text3.Text) + Val(Text6.Text) + Val(Text8.Text) + Val(Text10.Text)
e3 = Val(Text3.Text) + Val(Text4.Text) + Val(Text7.Text) + Val(Text10.Text)
f3 = Val(Text2.Text) + Val(Text4.Text) + Val(Text6.Text) + Val(Text9.Text)

```

```

If a3 <= b3 And a3 <= c3 And a3 <= d3 And a3 <= e3 And a3 <= f3 Then
Label32.Visible = True
Label33.Visible = True
Label34.Caption = " S - A - D - E"
Label35.Caption = (Val(Text2.Text) + Val(Text7.Text) + Val(Text10.Text))
Label19.Visible = True
Line11.Visible = True
Line12.Visible = True
Line16.Visible = True
Shape8.Visible = True
Shape10.Visible = True
Shape7.Visible = True
Shape3.Visible = True
Label26.Visible = True
Label27.Visible = True
Label30.Visible = True
Label31.Visible = True
Else
If b3 < a3 And b3 <= c3 And b3 <= d3 And b3 <= e3 And b3 <= f3 Then
Label32.Visible = True
Label33.Visible = True
Label34.Caption = " S - B - C - E"
Label35.Caption = (Val(Text3.Text) + Val(Text6.Text) + Val(Text9.Text))
Label19.Visible = True
Line13.Visible = True
Line15.Visible = True
Line9.Visible = True
Shape8.Visible = True
Shape10.Visible = True
Shape9.Visible = True

```

```

Shape1.Visible = True
Label26.Visible = True
Label28.Visible = True
Label29.Visible = True
Label31.Visible = True
Else
If c3 < a3 And c3 < b3 And c3 <= d3 And c3 <= e3 And c3 <= f3 Then
Label32.Visible = True
Label33.Visible = True
Label34.Caption = " S - A - C - D - E"
Label35.Caption = (Val(Text2.Text) + Val(Text5.Text) + Val(Text8.Text) +
Val(Text10.Text))
Label19.Visible = True
Line10.Visible = True
Line12.Visible = True
Line14.Visible = True
Line16.Visible = True
Shape8.Visible = True
Shape10.Visible = True
Shape3.Visible = True
Shape1.Visible = True
Shape7.Visible = True
Label26.Visible = True
Label27.Visible = True
Label29.Visible = True
Label30.Visible = True
Label31.Visible = True
Else
If d3 < a3 And d3 < b3 And d3 < c3 And d3 <= e3 And d3 <= f3 Then
Label32.Visible = True
Label33.Visible = True
Label34.Caption = " S - B - C - D - E"
Label35.Caption = (Val(Text3.Text) + Val(Text6.Text) + Val(Text8.Text) +
Val(Text10.Text))
Label19.Visible = True
Line10.Visible = True
Line12.Visible = True
Line15.Visible = True

```

```

Line13.Visible = True
Shape9.Visible = True
Shape10.Visible = True
Shape1.Visible = True
Shape8.Visible = True
Shape7.Visible = True
Label26.Visible = True
Label29.Visible = True
Label28.Visible = True
Label30.Visible = True
Label31.Visible = True
Else
If e3 < a3 And e3 < b3 And e3 < c3 And e3 < d3 And e3 <= f3 Then
Label32.Visible = True
Label33.Visible = True
Label34.Caption = " S - B - A - D - E"
Label35.Caption = (Val(Text3.Text) + Val(Text4.Text) + Val(Text7.Text) +
Val(Text10.Text))
Label19.Visible = True
Line11.Visible = True
Line12.Visible = True
Line15.Visible = True
Line18.Visible = True
Shape9.Visible = True
Shape10.Visible = True
Shape3.Visible = True
Shape8.Visible = True
Shape7.Visible = True
Label26.Visible = True
Label27.Visible = True
Label28.Visible = True
Label30.Visible = True
Label31.Visible = True
Else
If f3 < a3 And f3 < b3 And f3 < c3 And f3 < d3 And f3 < e3 Then
Label32.Visible = True
Label33.Visible = True
Label34.Caption = " S - A - B - C - E"

```

```
Label35.Caption = (Val(Text2.Text) + Val(Text4.Text) + Val(Text6.Text) +  
Val(Text9.Text))  
Label19.Visible = True  
Line9.Visible = True  
Line13.Visible = True  
Line16.Visible = True  
Line18.Visible = True  
Shape9.Visible = True  
Shape10.Visible = True  
Shape3.Visible = True  
Shape8.Visible = True  
Shape1.Visible = True  
Label26.Visible = True  
Label27.Visible = True  
Label28.Visible = True  
Label29.Visible = True  
Label31.Visible = True  
End If  
End If  
End If  
End If  
End If  
End If  
End If
```

```
End Function
```

```
Public Function No_Node()  
Form1.Refresh  
Command3.Enabled = False  
Label32.Visible = False  
Label33.Visible = False  
Label19.Visible = False  
Line9.Visible = False  
Line13.Visible = False  
Line15.Visible = False  
Line11.Visible = False  
Line18.Visible = False
```

Shape1.Visible = False
Shape8.Visible = False
Shape9.Visible = False
Shape10.Visible = False
Label28.Visible = False
Label29.Visible = False
Label31.Visible = False
Label32.Visible = False
Label33.Visible = False
Label38.Visible = False
Label39.Visible = False
Label40.Visible = False
Label34.Caption = ""
Label35.Caption = ""
Line10.Visible = False
Line14.Visible = False
Line16.Visible = False
Line12.Visible = False
Shape3.Visible = False
Shape7.Visible = False
Label26.Visible = False
Label27.Visible = False
Label30.Visible = False
Label1.Visible = False
Label2.Visible = False
Label7.Visible = False
Label3.Visible = False
Label4.Visible = False
Label5.Visible = False
Label6.Visible = False
Text2.Visible = False
Text3.Visible = False
Text4.Visible = False
Text5.Visible = False
Text6.Visible = False
Text7.Visible = False
Text8.Visible = False
Text9.Visible = False

```
Text10.Visible = False
Text2.Text = ""
Text3.Text = ""
Text4.Text = ""
Text5.Text = ""
Text6.Text = ""
Text7.Text = ""
Text8.Text = ""
Text9.Text = ""
Text10.Text = ""
End Function
```

```
Private Sub Text2_KeyPress(KeyAscii As Integer)
Dim result1
If KeyAscii < 48 And KeyAscii > 57 Then
result1 = MsgBox(" Kindly Enter only Numerals ", vbExclamation, " Wrong Entry
!!! ") = vbOK
'MsgBox KeyAscii
KeyAscii = 0
End If
End Sub
```

```
Private Sub Timer2_Timer()
Dim k
Dim t4
Dim rll
Dim sv
For k = 0 To 10

If (t4 Mod 10 = k.ipaddr) Then

"msgbox "nod(k).ipaddr" & nod(k).ipaddr
"msgbox "nod(sv).ipaddr" & nod(sv).ipaddr
"msgbox rll
"msgbox sv
k.Visible = True
```

```

    k.X1 = k.posy + 200
    "msgbox "rln(rl).X1" & rln(rl).X1
    k.Y1 = k.posx + 200
    "msgbox "rln(rl).y1" & rln(rl).Y1
    k.X2 = k.posy + 200
    "msgbox "rln(rl).X2" & rln(rl).X2
    k.Y2 = k.posx + 200
    "msgbox "rln(rl).y2" & rln(rl).Y2
    rll = k + 1

    sv = k

    End If
  Next
End Sub

' Programmed by, Arvind. N, Reg No: 04MCS057, II Year, M.Tech [CSE]

Private Sub cmdSend_Click()

End Sub

Private Sub Form_Load()

    optConnect_Click (1)

    MSComm1.CommPort = 1
    MSComm1.Settings = "9600,N,8,1"
End Sub

Private Sub MSComm1_OnComm()
    If MSComm1.CommEvent = comEvReceive Then
        ms
    End If
End Sub
Sub ms()
Dim a
    a = Asc(MSComm1.Input)

```

```

If a = 48 Then Exit Sub
If a = 44 Then Exit Sub

If a = Empty Then Exit Sub
txtMessage.Text = a - 15
End Sub

Private Sub txtMessage_Change()

For i = 0 To 30000000
Next
    txtStatus.Text = ""
    If Winsock1.State = sckClosed Then
        MsgBox "Please connect first.", vbCritical
        txtMessage.Text = ""
        txtMessage.SetFocus
        Exit Sub
    End If

    If txtMessage.Text <> "" Then

        Winsock1.SendData txtMessage.Text

        txtStatus.Text = txtMessage.Text

    End If

End Sub

Private Sub Winsock1_Close()
    cmdConnect_Click
End Sub

Private Sub Winsock1_ConnectionRequest(ByVal requestID As Long)
    If Winsock1.State <> sckClosed Then Winsock1.Close

```

```
Winsock1.Accept requestID
```

```
End Sub
```

```
Private Sub Winsock1_DataArrival(ByVal bytesTotal As Long)
```

```
    Dim strData As String
```

```
    Winsock1.GetData strData
```

```
    txtStatus.Text = strData
```

```
End Sub
```

```
Private Sub cmdConnect_Click()
```

```
    If cmdConnect.Caption = "Listen" Then ' Host
```

```
        txtPort.Enabled = False
```

```
        txtNick.Enabled = False
```

```
        optConnect(0).Enabled = False
```

```
        optConnect(1).Enabled = False
```

```
        cmdConnect.Caption = "Close"
```

```
    ElseIf cmdConnect.Caption = "Connect" Then ' Guest
```

```
        txtPort.Enabled = False
```

```
        txtNick.Enabled = False
```

```
        optConnect(0).Enabled = False
```

```
        optConnect(1).Enabled = False
```

```
        cmdConnect.Caption = "Disconnect"
```

```
    Else ' Disconnect or Close
```

```
        txtPort.Enabled = True
```

```
        txtNick.Enabled = True
```

```
        optConnect(0).Enabled = True
```

```
        optConnect(1).Enabled = True
```

```
    If optConnect(0).Value = True Then
```

```

        cmdConnect.Caption = "Listen"
        txtStatus.Text = txtStatus.Text & txtIP & ":" & txtPort & " closed
connection." & vbCrLf
    Else
        cmdConnect.Caption = "Connect"
        txtStatus.Text = txtStatus.Text & txtNick & " disconnected form " & txtIP &
":" & txtPort & vbCrLf
    End If

    Winsock1.Close

Exit Sub
End If

Select Case optConnect(0).Value
Case True:
    Winsock1.LocalPort = txtPort.Text
    Winsock1.Listen
    txtStatus.Text = txtIP & ":" & txtPort & " is waiting for connection ..." &
vbCrLf
Case False:
    Winsock1.RemoteHost = txtIP.Text
    Winsock1.RemotePort = txtPort.Text
    Winsock1.Connect
    txtStatus.Text = txtStatus.Text & txtNick & " is connecting to " & txtIP & ":" &
txtPort & vbCrLf
End Select

MSComm1.CommPort = Text1

MSComm1.PortOpen = True
End Sub

Private Sub optConnect_Click(Index As Integer)
    Select Case Index
    Case 0:
        txtNick.Text = "Host"

```

```
txtIP.Text = Winsock1.LocalIP
txtIP.Enabled = False
cmdConnect.Caption = "Listen"
Case 1:
txtNick.Text = ""
txtIP.Text = ""
txtIP.Enabled = True
cmdConnect.Caption = "Connect"
End Select
End Sub
```

GCCRIS

APPENDIX C

BASCOM IDE SOURCE CODE

```
Dim Vv As Byte
Ff:
Input "hellow" , Vv
Lcd "Sensor Networks"
Lowerline
Lcd "temp.= "
Lcd Vv
Waitms 250
Cls
Goto Ff
```

APPENDIX D

GENUS UNIVERSAL PROGRAMMABLE – G540



Figure – G540 Programmable

APPENDIX E

CURRICULUM VITAE

PERSONAL INFORMATION

Surname, Name: Yaser, Hamodi

Nationality: Iraq (IR)

Date and Place of Birth: 28/04/1983, IRAQ / BAGHDAD

Marital Status: Married

Phone: +90 535 38 04 459, +964 770 58 37 615

Email: yaserissam@yahoo.com, c1071503@student.cankaya.edu.tr

EDUCATION

Degree	Institution	Year of Graduation
MS	Çankaya Univ.Computer Engineering	2012
BS	College Of Electrical And Electronic Techniques	2004
High School	Al-iraq Al-jadeed	2000

WORK EXPERIENCE

Year	Place	Enrollment
2004	MOHE Of Iraq	Admin of Server
2006	AL- Nahrain University	Maintaining Computers
2007	STS. Company In Jordan	Admin Of Unix Server
2012	Iraq Culture Attache	Web Page Designing.

FOREIGN LANGUAGES

English, Fluent Arabic

PUBLICATIONS

1. Research published in the fifth conference in Journal of Çankaya University titled "Rule Extraction in Diagnosis of Vertebral Column Disease" in 2012.

HOBBIES

Security Programming, Hardware Circuit Maintenance, Football, Car Sports.