

**PC-BASED AND WEB-BASED SYSTEM  
FOR CONTROLLING XY ROBOT**

By

FATIN AWINA BINTI AWIS

FINAL PROJECT REPORT

Submitted to the Electrical & Electronics Engineering Programme  
in Partial Fulfillment of the Requirements  
for the Degree  
Bachelor of Engineering (Hons)  
(Electrical & Electronics Engineering)

Universiti Teknologi PETRONAS  
Bandar Seri Iskandar  
31750 Tronoh  
Perak Darul Ridzuan

© Copyright 2004

by

Fatin Awina Binti Awis, 2004

# **CERTIFICATION OF APPROVAL**

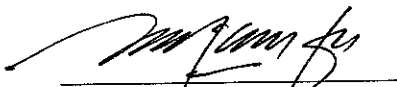
## **PC-BASED AND WEB-BASED SYSTEM FOR CONTROLLING XY ROBOT**

by

Fatin Awina Binti Awis

A project dissertation submitted to the  
Electrical and Electronics Engineering Programme  
Universiti Teknologi PETRONAS  
in partial fulfilment of the requirement for the  
BACHELOR OF ENGINEERING (Hons)  
(ELECTRICAL AND ELECTRONICS ENGINEERING)

Approved by,

---

(Mr. Rosdiazli Ibrahim)  
Project Supervisor

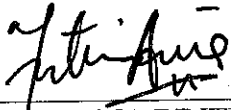
**UNIVERSITI TEKNOLOGI PETRONAS**

**TRONOH, PERAK**

**June 2004**

## CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledgements, and that the original work contained herein have not been undertaken or done by unspecified sources or persons.



---

FATINA AWINA BINTI AWIS

## ABSTRACT

This dissertation presents the development of PC-based system, Web-based system and Human Machine Interface for the project 'PC-Based and Web-Based System for Controlling XY Robot '. The objective of this project is to create systems that are capable of controlling an XY Robot through Human Machine Interface (HMI). For that matter, the project is being divided into three sections; the development of PC-based system, the development of Web-based system and the development of Human Machine Interface. PC-based system is aim to control the robot through HMI through serial link communication. As for Web-based system, its aim is to control the robot through HMI over the Ethernet connection. For this system, its HMI is being made into an interactive web page. Users can access its server's web site for the purpose of controlling the robot. The devices used in this project are mainly the OMRON Factory Automation devices. Therefore, the software used come from OMRON as well. The software essential for this project are ; Visual Basic 6.0, ONC Active X Control, FA Components, Data Agent and SYSMAC Compolet for HMI and web application development, FinsGateway for device communication and CX Programmer for robot system programming. Both systems are developed by designing its' respective system architecture and followed by devices settings and configuration. The systems are deem successful when communication is established between the end user and PLC which contains the program of the robot.

## **ACKNOWLEDGEMENT**

I would like to express my heartfelt gratitude to my supervisor; Mr. Rosdiazli Ibrahim for his guidance, support and endless patience while assisting me with this project. Special thanks to Miss Chin Lee Yong of OMRON for the technical advices and assistance. This project may not come this far without her help. Not to forget, Mr. Azhar and Mr. Musa; the UTP E&E lab technicians for their support in making this project possible. Last but not least, to my parents and my friends; for the encouragement and moral support. Thank you all.

## TABLE OF CONTENTS

<b>CERTIFICATION OF APPROVAL</b>	. . . . .	i
<b>CERTIFICATION OF ORIGINALITY</b>	. . . . .	ii
<b>ABSTRACT</b>	. . . . .	iii
<b>ACKNOWLEDGEMENT</b>	. . . . .	iv
<b>TABLE OF CONTENTS</b>	. . . . .	v
<b>LIST OF FIGURES</b>	. . . . .	vii
<b>LIST OF TABLES.</b>	. . . . .	viii
<b>CHAPTER 1:</b>	<b>INTRODUCTION</b>	1
	1.1 Background	1
	1.2 Problem Statement	1
	1.3 Objectives of the Project	2
	1.4 Scope of Study	3
<b>CHAPTER 2:</b>	<b>LITERATURE REVIEW</b>	4
	2.1 Teleoperation and Telerobotics	4
	2.1.1 Introduction	4
	2.1.2 General Components	5
	2.2 LAN Networking Principles	6
	2.3 TCP/IP and Internet Protocol Fundamentals	7
<b>CHAPTER 3:</b>	<b>METHODOLOGY</b>	10
	3.1 Procedures Identification	10
	3.2 Project Timeline	11
	3.3 Tools Requirement	11
<b>CHAPTER 4:</b>	<b>RESULTS AND DISCUSSION</b>	13
	4.1 Development of PC-Based System	13
	4.1.1 System Architecture	13
	4.1.2 Settings Parameter	14
	4.1.3 Discussion	17

4.2	Development of Human Machine Interface (HMI)	. . .	18
4.2.1	Procedures	. . .	18
4.2.2	Discussion	. . .	21
4.3	Development of Web-Based System	. . .	23
4.3.1	System Architecture	. . .	23
4.3.2	Device Installation and Settings Parameter	. . .	24
	4.3.2.1 Ethernet Unit (CSIW-ETN 11)	. . .	24
	4.3.2.2 Open Network Controller (ONC ITNC-ELX01)	. . .	32
4.3.3	Web Application Development	. . .	35
4.3.4	Discussion	. . .	36
	4.3.4.1 Ethernet Unit (CSIW-ETN 11)	. . .	36
	4.3.4.2 Open Network Controller (ONC ITNC-ELX01)	. . .	38
	4.3.4.3 Web Application Development	. . .	39
<b>CHAPTER 5:</b>	<b>CONCLUSION AND RECOMMENDATION</b>	. . .	40
<b>REFERENCES</b>	. . . . .	. . . . .	41
<b>APPENDICES</b>	. . . . .	. . . . .	42

## **LIST OF FIGURES**

- Figure 2.1: TCP/IP Protocol Architecture
- Figure 2.2: IP Address Format
- Figure 2.3: Example of Subnetworking
- Figure 4.1: PC-Based System Architecture
- Figure 4.2: Service Control Manager
- Figure 4.3: Node Setting Parameter for COM1 Serial Unit Properties
- Figure 4.4: Network Setting Parameter for COM1 Serial Unit Properties
- Figure 4.5: FINSGateway Network Tester
- Figure 4.6: SYSMAC C Icon for C Series PLC Control on Form
- Figure 4.7: Property Pages
- Figure 4.8: Mock XY Robot Program
- Figure 4.9: Mock HMI
- Figure 4.10: Compolet Coding for Mock HMI
- Figure 4.11: Web-Based System Architecture
- Figure 4.12: ETN Unit
- Figure 4.13: ETN Unit CPU Bus Unit
- Figure 4.14: Network Configuration Tools
- Figure 4.15: Routing Table Setup
- Figure 4.16: Routing Table
- Figure 4.17: Network Settings
- Figure 4.18: Driver Settings
- Figure 4.19: Open Network Controller
- Figure 4.20: Pin Layout for PC-ONC Communication
- Figure 4.21: Pin Layout for ONC-PLC Communication
- Figure 4.22: Hyper Terminal
- Figure 4.23: Accessing ONC via Ethernet
- Figure 4.24: Logging onto ONC via Ethernet



Figure 4.25: Ping on PLC CS1

Figure 4.26: FINSGateway Network Tester on PLC CS1

Figure 4.26: Ping on ONC

## **LIST OF TABLES**

Table 3.1: Project Software Requirement

Table 3.2: Project Hardware Requirement

Table 3.3: LAN Specifications

Table 4.1: Compolet Series Component

Table 4.2: IP Address Conversion

# CHAPTER 1

## INTRODUCTION

### 1.1 Background

In our modern society, robot is being manipulated in almost every field and area, so much so that it has become an integral part of our life. For example, robots are being use in manufacturing industry and service sectors; to either fulfill the increasing demand of productivity or to help deliver quality services.

Before the millennia, the operations of robotic system in these areas have remained separate and independent (Hu and Brandy, 1997). In other words, components of robotic systems are isolated from one another by different environment and have no effective way to communicate. However, with the advancement of today's computer and Internet technology, it is possible to develop an integrated network environment to cater the vast application of the robotic system. This include of having a standard network protocol and an interactive human –machine interface. Using a Web browser alongside with its intuitive user interface, a remote operator can control the robot via the Internet.

This dissertation describes the author's work in developing a PC-based and Web-based system that is capable of controlling an XY Robot through Human Machine Interface, via the Local Area Network.

### 1.2 Problem Statement

Robot applications are vast and in some applications, it works in inhospitable environment. Such application would be handling radioactive materials and perform salvage work in the ocean depths. To be able to control the robot without the

presence of its operator nearby, a mean of communication for the purpose of remote controlling is required. The concept of controlling robots at a distance is called 'teleoperation'.

However, the concept evolves as the usage of robot is expanding. Although its initial purpose of remote controlling remains the same; its applications does not only directed to working in hostile environment. Instead, it has varies to suit the demand of many sectors. In manufacturing sectors, remote controlling would be beneficial for a robot system placed separately from its operator due to the constraint or condition of the factory. For example, a robot system might be placed in a hot environment which is unsuitable for its operator to be around. In other case, a factory might have several locations where the robot systems are located far away from its operator or controller.

By utilizing the capabilities of today's computer and Internet technology, an integrated network environment for robotic system can be created. Hence, the mentioned problems can be solved. Furthermore, the effectiveness of the robot usage is boost to its utmost level.

### **1.3 Objectives Of The Project**

1. To develop a PC- based system that is capable of controlling an XY Robot locally.
2. To develop a Web-based system that is capable of controlling an XY Robot remotely.
3. To create a Human Machine Interface (HMI) for the identified XY Robot application using Visual Basic 6.0(VB 6.0).

#### **1.4 Scope Of Study**

To build the systems, OMRON devices are used. The devices used are its Programmable Logic Controller (PLC) of C200HE Series and CS1 Series, Open Network Controller (ONC) of ITNC-EIX01 model and Ethernet Unit (ETN Unit) of CS1W-ETN 11 model. The software used are VB 6.0, SYSMAC C Compolet and CX Programmer. Therefore, to achieve the mentioned objectives; knowledge on the devices installation and settings, programming of the mentioned software and data networking are required.

In this project, PLC is the 'brain' of the XY Robot system. In other words, programs for the movement of the robot are being done and stored in PLC. Hence, knowledge on programming the PLC is vital. To program OMRON PLCs, CX Programmer is being used.

In creating the HMI for the identified XY Robot function, knowledge on VB 6.0 programming; particularly regarding the SYSMAC Compolet coding is needed. SYSMAC Compolet is the software that allows OMRON PLCs series control in the VB 6.0 environment.

To develop both PC-based and Web-based system, basic knowledge on data communications and networking is needed. This includes understanding in LAN technology, network protocols and fundamentals of TCP/IP. Example of TCP/IP fundamentals would be client/server programming and the concept of domain names and IP addresses.

As the OMRON devices are being used, emphasis is given on procedures of setting up the devices and its settings. This is to ensure communication exist with one another over the network.

## **CHAPTER 2**

### **LITERATURE REVIEW**

This chapter reviews the written materials related to this project and also the area of study required in developing the Web-based system. It gives the reader an overview of previous researches and projects done by other scholars.

#### **2.1 Teleoperation and Telerobotics (Internet Robotics)**

##### **2.1.1 Introduction**

Teleoperation is the basic concept employed for this project. The system allow user to control robot remotely, i.e., without having the operator to be physically present at the same place with the robot. Teleoperated robots were developed over 30 years ago. The main objective was to develop a system capable of working in inhospitable environments (e.g. radiation sites). Along the way, the concept has expanded; where more intelligent devices such as surveillance camera and manipulators are embedded and Internet technology is being use as the medium of communication. For now, remotely controlled robots are being use for space exploration, perform salvage work in the ocean depths and clean up spills of hazardous substances. Robotics control is also being use in operating rooms, where surgeons can perform operation at distant locations.

There are distinct different between teleoperation and telerobotics (or Internet robotics). Basically, both imply remote controlling, although the latter was the evolution of the former. Internet robotics involves controlling a robot or device from a Web browser. Among the different aspect between both would be the delay, where delay and through put of the Internet are likely unpredictable, as compared to traditional teleoperation system; where the interfaces have fixed and guaranteed delay. Another aspect would be the level of tolerance; where Web-based telerobotics system has a high degree of

possible data-package loss due to packet discard when there is no existing remedy. As for teleoperation system, a dedicated wire connecting the robot and controlling system resulted in little data loss over transmission.

Internet robotics started in 1994. The first generation of Internet robots revolves around simple robotic arms controlled by human operators. These telerobots operate within well structured environment with little uncertainty and little intelligence. The second generation focus on autonomous mobile robot that navigate in dynamic and uncertain environment e.g. the Xavier—an office exploring robot at Carnegie-Mellon University <sup>[1]</sup>.

In the papers written on telerobotics projects; it's working principles, framework and software/hardware components involved were mentioned. Through reading, basic idea of how the system would be can be obtained. The papers also highlighted the problems encountered during project development. By this, the problem that is yet to be encountered can be anticipated. Better still, improvement can be done at the designing level. Among the problems mentioned is the restricted bandwidth and arbitrarily large transmission delay, which due to the performance of Internet <sup>[1]</sup>. Another problem highlighted is the network traffic. Adding camera vision features to the controlling websites complicates matter even further due to the limitation of HTML/HTTP environment <sup>[2]</sup>.

### **2.1.2 General Components**

There are various strategies that can be implemented as the client –server system for controlling robots over the World Wide Web. Generally most system, regardless of the networked appliances, consists of the following;

1. The controlled object or robot.
2. The microprocessor as either part of the controlled body or as a separate devices that interface the object to a personal computer or directly to the network.
3. The operator-PC computer as a client on the computer network with a GUI interface for controlling purposes or as a server being an interface between the controlled object and internet.

The software components depend on the hardware architecture and usually include:

1. The firmware for the embedded microprocessor.
2. The software that interfaces a personal computer to the controller object or to the embedded system.
3. The user interface for performing the control.
4. The internet server in case if the object need to be controlled via the Internet.

## **2.2 LAN Networking Principles**

Local Area Network (LAN) consists of a shared transmission medium and a set of hardware and software for interfacing devices to the medium. It allows devices to exchange data over the shared transmission medium but imposes certain rule in regulating the orderly access to the medium.

LAN is being use by many organizations for the convenience of data exchange. This includes data exchange between personal workstations and each other and also to a centralized facility. This is to fulfill the demand of today's working nature which requires information sharing.

The common protocol architecture would be the IEEE 802 reference model. Protocol architecture cites how blocks data is transmitted over the network. Important features of LAN would be its topologies and Medium Access Control (MAC) technique. LAN topology describes the physical layout of the interconnecting devices while MAC technique describes the method of determining which device in the network has access to the transmission medium at any given time.

Basic knowledge on LAN is required as LAN connection will be use for developing the Web-based system.

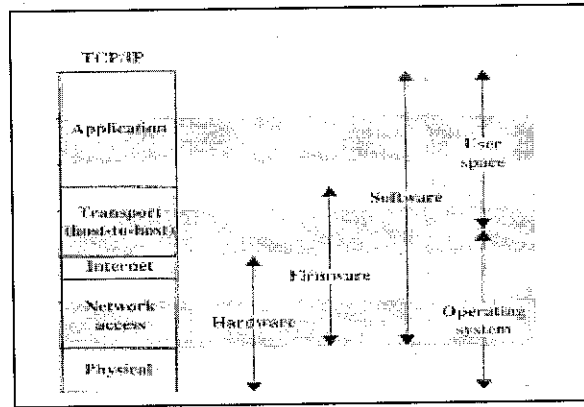
### 2.3 TCP/IP and Internet Protocol Fundamentals

When computers, terminals, and other data processing devices exchange data, it needs to agree on the same *protocols* and *protocol architecture*. *Protocol* is a convention or set of rules use for communication between entities in different system. An example of entity would be a user application program whilst an example of system would be computers or terminals. For the two entities to communicate, both must speak ‘the same language’. The key elements of *protocols*; which determine how two entities communicate, what is communicated and when it is communicated are *syntax*, *semantics* and *timing*. *Syntax* includes matter like data format and signal level, *semantic* is control information for coordination and error handling and *timing* controls speed matching and sequencing.

*Protocol architecture* is a structured set of modules that implements the communication functions. For example, each data frame transmitted or file transferred is unique; such as transmitting password, file command and records. .To ensure its sound transmission, there is a certain kind of reliability required. The same sort of reliability requirement can be used by variety of applications such as e-mail and document transfer. Therefore these requirements are met by a separate communication service module that can be use by a variety of application.

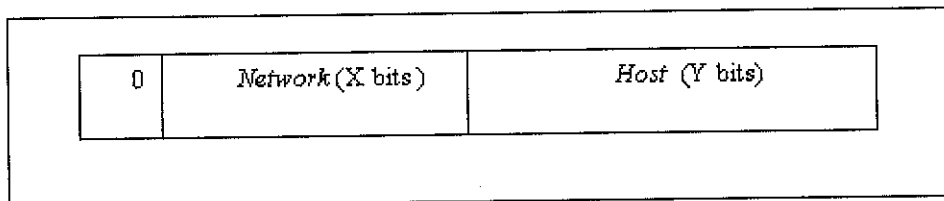
The most popular architecture protocol would be TCP/IP model. TCP/IP layers is depicted in Figure 2.1 .Each layer execute specific task in attending to a data frame reliability requirement.





**Figure 2.1: TCP/IP Protocol architecture**

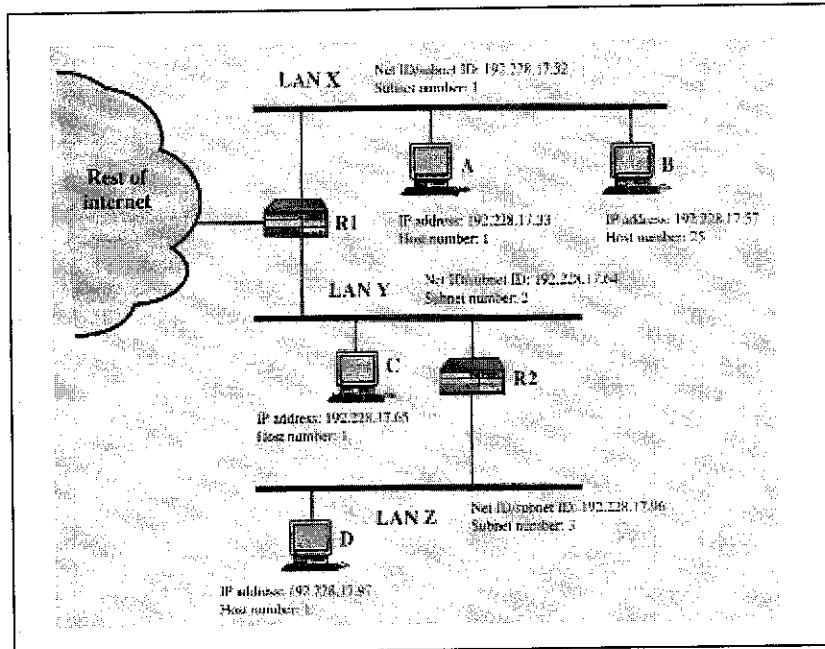
Internet Protocol (IP) is a layer of TCP/IP suite. It is the most widely used internetworking protocol. Basically, IP manages data that is being send between two systems over the internet. The most important matter when discussing IP would be the IP addresses. As internet involve multiple networks, the source and destination address in IP header appends at a data frame must consist of a network identifier and a host identifier as well. Figure 2.2 depicts IP address frame format. IP addresses are written in *dotted decimal notation*, where a decimal number representing each of the octets of the 32-bit address. For example, the IP address 11000000 11100100 00010001 00111001 is written as 192.228.17.57



**Figure 2.2: IP Address Format**

To simplify addressing and routing, the concept called *Subnets* and *Subnet Masks* are being used. Basically, a single network number termed *subnet number* is assigned to all of the LANs at a site. This is to allow the routers within the site to function properly. The *host* portion of the IP address is partitioned into a *subnet number* and a *host number* to accommodate this new level of addressing.

Within the subnetted number, the local routers must route on the basis of an extended network number consisting of the *network* portion of the IP address and the *subnet number*. The bit positions containing this extended *network number* are indicated by the *address mask*. The use of the *address mask* allows the host to determine whether an outgoing datagram is destined for the host on the same LAN (send directly) or another LAN (send datagram to router). Figure 2.3 illustrate and example of the use of subnetting.



**Figure 2.3: Example of Subnetworking**

In developing both systems, it is vital to have at least basic knowledge regarding data communication. Two automation devices; ONC and ETN Unit will be use to realize the communication between the components of the project; the PLC and PCs. To get it functioning, the correct configuration and parameter settings are required. Therefore, in order to do that, understanding pertaining data communication and its terminology is very much needed.

## **CHAPTER 3**

### **METHODOLOGY**

#### **3.1 Procedures Identification**

This chapter informs about the project's working procedure, which acts as a guidance. Working procedure is presented in a flowchart; which can be observed as Appendix 1. The first step is the preliminary research and literature review. The related area are being identified and studied. This is to obtain insights regarding the project; as well as what have been developed in the past, what is being done in the present and the potential area that can be develop for further improvement of the project.

The next step is the project scope and working procedures identification. Based on insights obtained earlier, project scope and stages are defined. With this, working procedures can be identified.

The third step is the identification of project hardware and software components. This covers the requirement for both PC-based and Web-based system. Emphasis is given on how to utilize them. This includes device installation and software settings.

The next step is the development of PC-based system and it's HMI. This is followed by Web-based system development. Between each development stages, there is system testing. This is to see the workability of each system .Upon failure; necessary corrective action will be taken.

Finally would be the whole system integration stage, where both systems are being integrated with the actual XY Robot system.

### 3.2 Project Timeline

Project timeline was made in order to ensure that the project can be completed within its duration of two semesters. Please refer Appendix 2 and Appendix 3 for the Project Gantt Chart for Semester 1 and 2 respectively. The key activities have been identified and the time allocated for each are thought to be sufficient. The project is feasible within the given duration if there is no additional key activity and the system development goes smoothly.

### 3.3 Tools Requirement

Project's tools requirement consist of software and hardware requirement. The tools that have been identified are tabulated as Table 3.1 and Table 3.2. The details regarding LAN of UTP are listed in Table 3.3.

**Table 3.1: Project Software Requirement**

Items	Software
User Interface	Visual Basic 6.0, ONC Active X Control, FA Components and Data Agent
Application program	SYSMAC Compolet Version 2 (for OMRON PLC ),
Communication software	FinsGateway
Programming software	CX Programmer

**Table 3.2: Project Hardware Requirement**

Items	Hardware
Robot PLC	OMRON SYSMAC C/CS1/CV –series and C200HE-series PLC
System housing	<ul style="list-style-type: none"><li>• Personal Computers (PC) with the mentioned software</li><li>• The necessary cables</li></ul>
Gateway device	ONC ITNC-EIX01 with Flash Card QNX Format
Information system network device	Ethernet Unit (ETN Unit) CS1W-ETN 11

**Table 3.3: LAN Specifications**

<b>Items</b>	<b>Details</b>
LAN	<ul style="list-style-type: none"><li>• Class B network (subnet mask : 255.255.252.0)</li><li>• IP addresses use : 160.0.44.5, 160.0.44.6, 160.0.44.7, 160.0.44.8</li></ul>

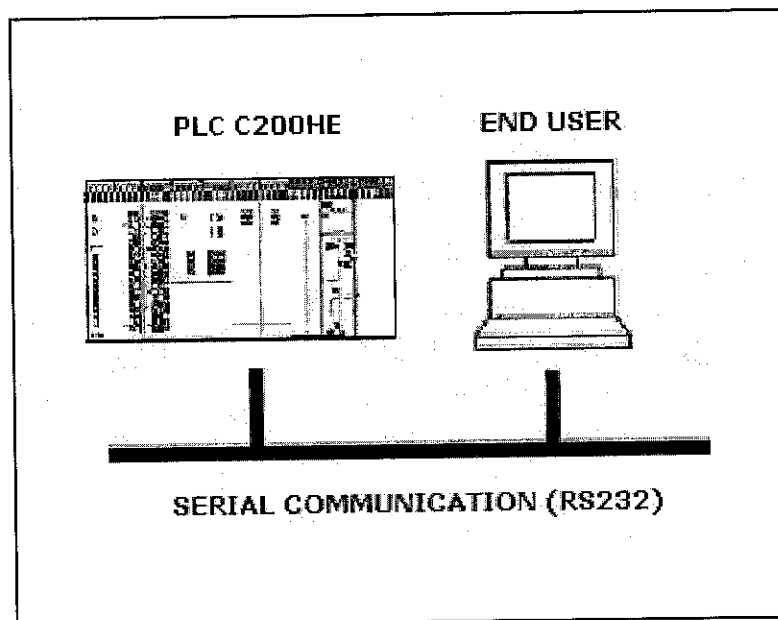
## CHAPTER 4

### RESULTS AND DISCUSSION

This chapter presents the results of work done in developing both systems. The chapter is divided into three sections: development of PC-based system, development of Human Machine Interface and development of Web-based system.

#### 4.1 Development of PC-Based System

##### 4.1.1 System Architecture



**Figure 4.1: PC-Based System Architecture**

The PC-Based system consists of an OMRON C200HE PLC and a PC with Window 2000 as its Operating System. PLC and PC communication is done through serial

communication port or COM port terminal using RS 232 cable. Basically, the system will work in a manner where a HMI at the PC will allow user to start a program in the PLC.

To ensure a successful PC–PLC communication, the system requires FINSGateway; OMRON standard communication software. This software allows the interchange of message and data between Microsoft Window Operating System and OMRON PLC; such as the C/CS1/CV Series PLC. The system also requires important software called SYSMAC Compolet (SYSMAC C). This software treats OMRON PLCs as an *Object Linking and Embedding (OLE)* control. With this, PLC can be ‘embedded’ onto VB 6.0 application such as HMI; thus allowing remote controlling of PLC.

#### 4.1.2 Settings Parameter

To establish PC-PLC connection, activate the **FINSGateway Service Manager**. Under **Service Control Manager**, start the **CPU\_UNIT** and **SerialUnit** by clicking ‘Start’ button at ‘Status’. Upon activation, ‘Status’ will cite ‘Running’. The window display can be viewed as Figure 4.2.

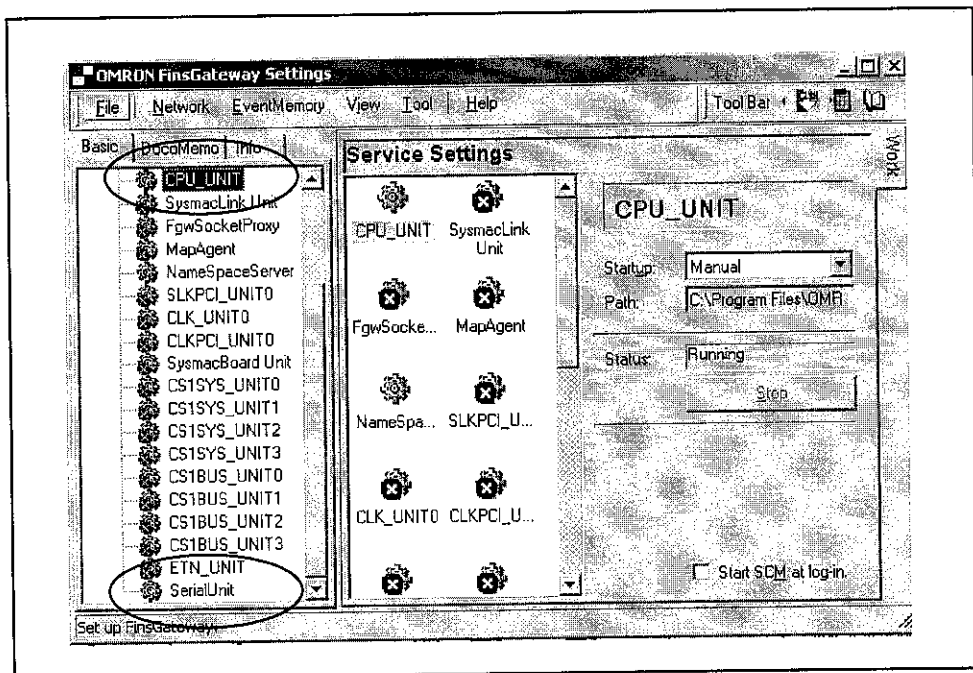
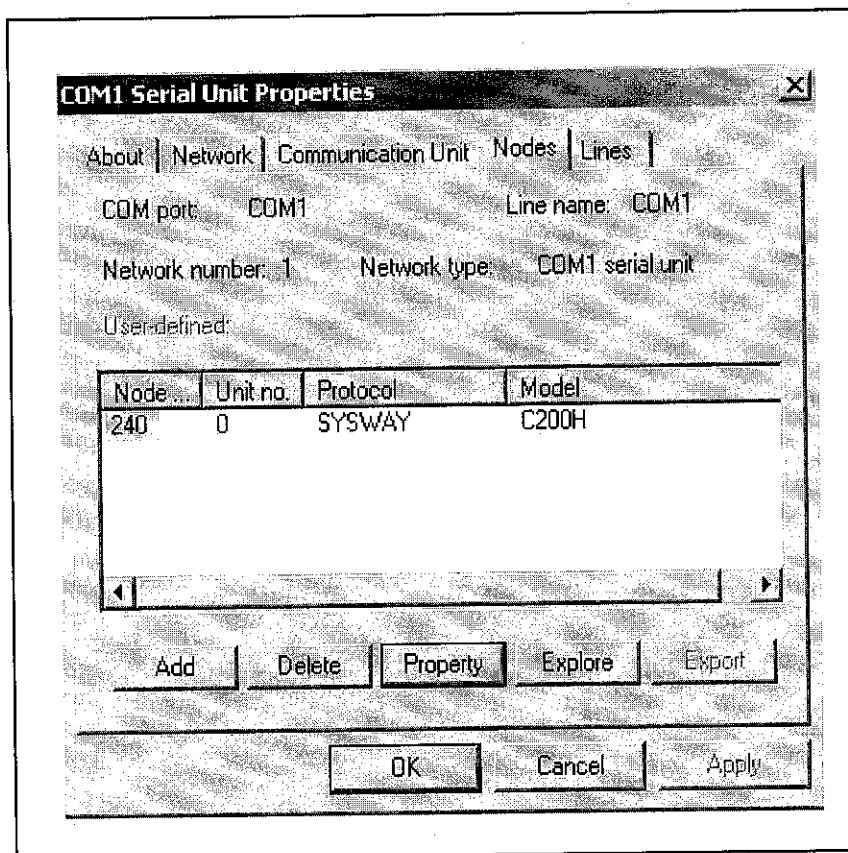


Figure 4.2: Service Control Manager

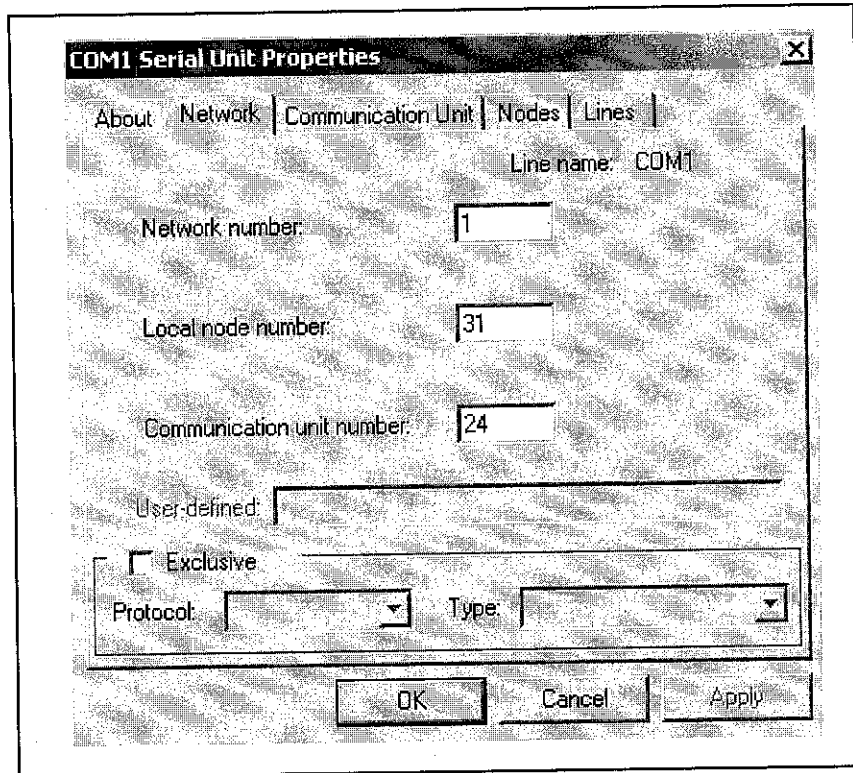
Settings parameter normally appears automatically once PC-PLC communication is established. To verify whether communication does exist, check its COM port properties. The serial communication port use is **COM 1**. The screen flow to check its properties is 'Basic'- 'Services'- 'Networks'- 'Local Network 1'- 'Properties'. Its properties or settings parameter can be observed from Figure 4.3 and 4.4.



**Figure 4.3: Node Setting Parameter for COM 1 Serial Unit Properties**

Figure 4.3 informs the property of the node or device connected to the network. The device detected is C200H, which is the exact model of PLC use. The **Unit** number of the node is recognized as '0' and **Node** number is recognized as '240'. The protocol use is 'SYSWAY'. SYSWAY is the type of protocol used when the communication is of the serial-type.





**Figure 4.4: Network Setting Parameter for COM 1 Serial Unit Properties**

Based on Figure 4.4, 'Network number: 1' indicates the number of network that is currently being use. Number of network can be added e.g., 'Network number: 2' and so forth if the PC has more than one COM port. 'Local node number' cites the number of devices that can be attached to the network. For this case, 31 nodes allowed. 'Communication unit number' indicates the type of network use. **FINSGateway** allow data transfer via many type of network: Ethernet, Serial and Controller Link; to name a few. Each type of network has been allocated with its own communication unit; whereby data will be transferred through this communication unit and onto the network, before it reach the intended node. Communication unit '24' has been allocated for serial -type network, and therefore means that 'Network number: 1' is serial communication.

Another way to verify communication is by executing **FINSGateway Network Tester**. Insert settings parameter, which are **Network** address, **Node** address and **Unit** address (N.n.u). Click 'Start' and a window as depicts in Figure 4.5 will appear. Encircled information proved that **FINSGateway** detects the connected PLC.

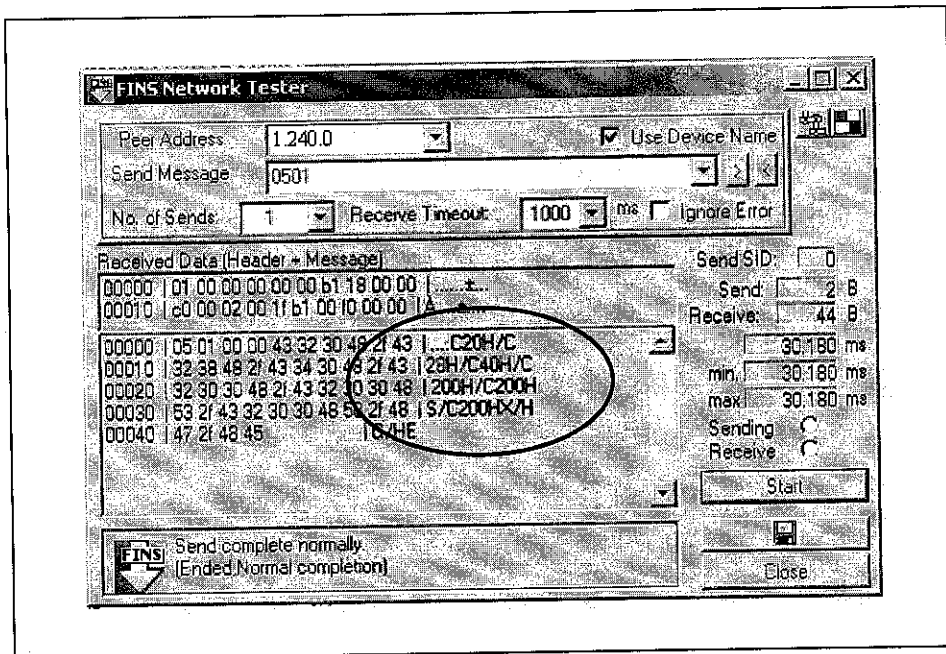


Figure 4.5: FINSGateway Network Tester

#### 4.1.3 Discussion

The objective of this system is to establish PLC-PC communication. Through **FINSGateway Service Manager** activation, the communication is successful.

The main important part of the system is to identify its **Network**, **Node** and **Unit** component, and also its addresses. The components of **Network** are PC and PLC; recognized by FINSGateway as **CPU\_UNIT** and **C200HE** respectively. For this PC-based system, Network address has been set to '1'.

**Node** is referring to the device connected to the PC. For this, only PLC C200HE is being used and recognize as node. PLC is connected to PC through **COM 1**. Its **Node** address is fixed at '240' by FINSGateway configuration. It is the address allocated for PLC connected to PC in the serial communication manner.

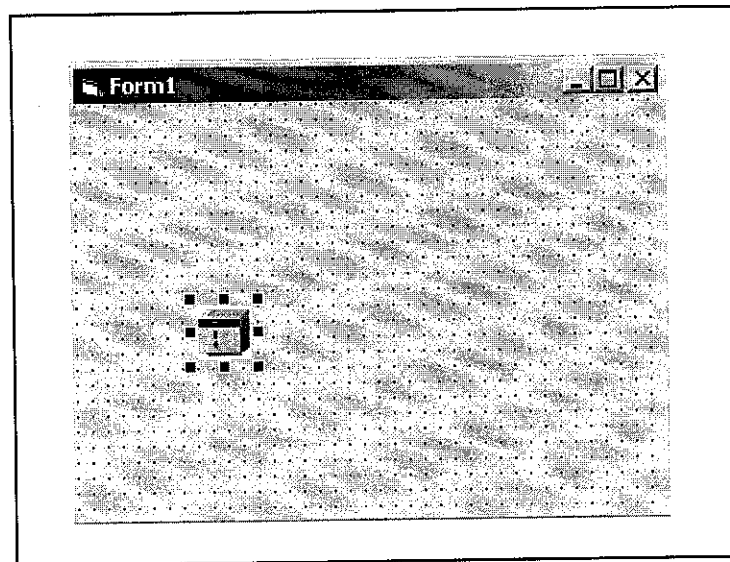
**Unit** is referring to the operating CPU unit in the network. In this system, only one PC is used and as mentioned, is recognized by FINSGateway as **CPU\_Unit**. **Unit** address for **CPU\_Unit** is fixed at '0' by FINSGateway.

## 4.2 Development of Human Machine Interface (HMI)

To design the HMI, VB 6.0 is used as it is widely accepted and compatible with OMRON PLCs. To allow PLC to be controlled from VB 6.0 program, SYSMAC Compolet (SYSMAC C) is required. SYSMAC C is an intermediate software containing the control of C/CS1/CV –series OMRON PLC.

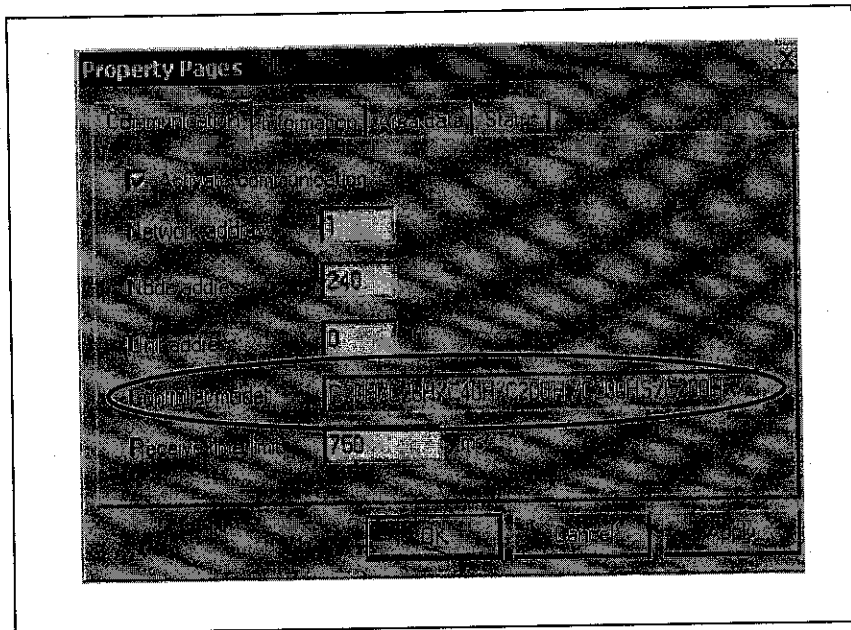
### 4.2.1 Procedures

Open VB 6.0 and choose 'Standard EXE' under 'New Project'. A form will appear. Choose OMRON SYSMAC C Control from 'Project Component'. SYSMAC C icon will appear and is placed on form. This is illustrated in Figure 4.6



**Figure 4.6: SYSMAC C Icon for C Series PLC Control on Form**

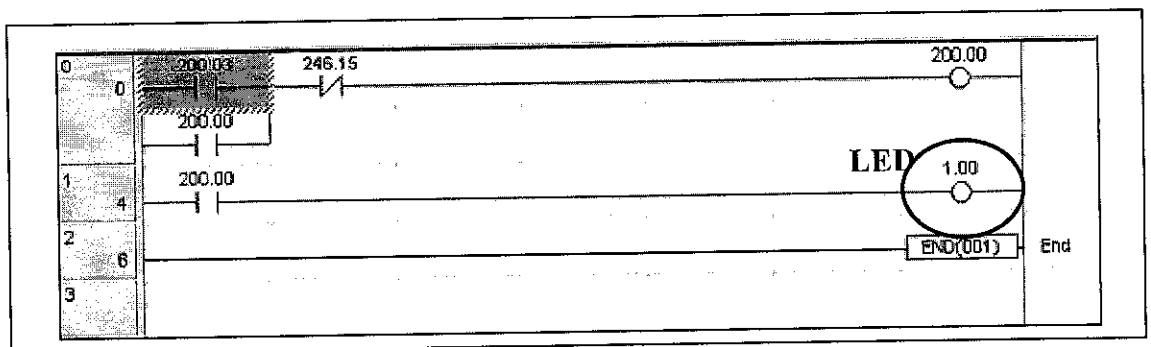
To allow the SYSMAC C control to function, double clicked the icon and it's 'PropertyPages' will appear. Insert the same **Network**, **Node** and **Unit** address (N.n.u); which is 1.240.0. If VB 6.0 detects PC- PLC communication, the 'Controller Model' will cite the PLC model that is currently being use. This can be viewed as Figure 4.7.



**Figure 4.7: Property Pages**

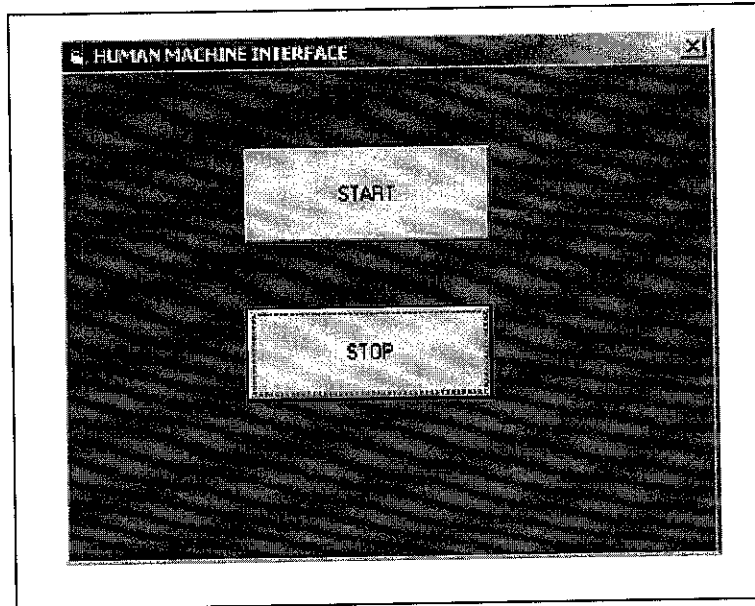
After the settings are done, HMI can be created by adding controls on the form. VB 6.0's commonly use control would be the CommandButton, which allow user to initiate program by clicking onto it. After placing the necessary CommandButton, programmer needs to write some code so that it will execute certain action.

To simplify work, the XY Robot program is being replaced by a very simple PLC program (Figure 4.8). The PLC program is written to switch on an LED when address 200.03 is triggered and switch off LED when address 246.15 is activated. This program was written using CX Programmer.



**Figure 4.8: Mock XY Robot Program**

A mock HMI is created based on this program. This can be viewed as Figure 4.9. Through button 'Start' and 'Stop', LED can be switch on and off. Beforehand, the CommandButton 'Start' and 'Stop' must be coded. Compolet coding can be viewed as Figure 4.10.



**Figure 4.9: Mock HMI**

```
Private Sub stopbutton_Click()  
SYSMAC_C1.writeArea plcAreaCIO, 246,1,"8000"  
SYSMAC_C1.writeArea plcAreaCIO, 200,1,"0000"  
End Sub  
  
Private Sub startbutton_Click()  
SYSMAC_C1.writeArea plcAreaCIO, 200,1,"0008"  
SYSMAC_C1.writeArea plcAreaCIO, 246,1,"0000"  
End Sub
```

**Figure 4.10: Compolet Coding for Mock HMI**

### 4.2.2 Discussion

The attempt to create a working user interface for this simple mock program was successful. Thus, HMI for the actual XY Robot program can be developed by following the same procedure. This is because, as long as there is Normally Open or Normally Close contact in a PLC program (which they usually have), it is possible to construct its HMI following the same method.

What is most important in developing the HMI is the usage of internal relay (IR) addresses to the contacts in PLC program. IR addresses allow PLC program to be invoked internally, i.e., through communication link.

Based on Figure 4.8, IR addresses are being used in constructing the PLC program or ladder diagram. The address 246.03 here is an example of IR address, assigned to Normally Open contact. In this program, address 246.03 needs to be invoked in order to switch on the LED output. Range of IR addresses can be obtained from the PLC Programming Manual and is different from one series of PLC to another.

The next important step is to know how to invoke these IR addresses from SYSMAC C control. From Figure 4.10, the coding for 'Start' button is **SYSMAC\_C1. writeArea plcArea CIO, 200, 1,"0008"**. '246' represents IR 246 and "0008" is citing the IR address in full (for this case; 246.03). '1' means sending the value of 1 to the address. In other words, it triggers the address. To convert the last two decimal of every IR addresses into SYSMAC C code, we need to refer to the Compolet Series Component Table. This can be viewed as Table 4.1

**Table 4.1: Compolet Series Component**

0				0				0				8			
15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
$2^3$	$2^2$	$2^1$	$2^0$	$2^3$	$2^2$	$2^1$	$2^0$	$2^3$	$2^2$	$2^1$	$2^0$	$2^3$	$2^2$	$2^1$	$2^0$

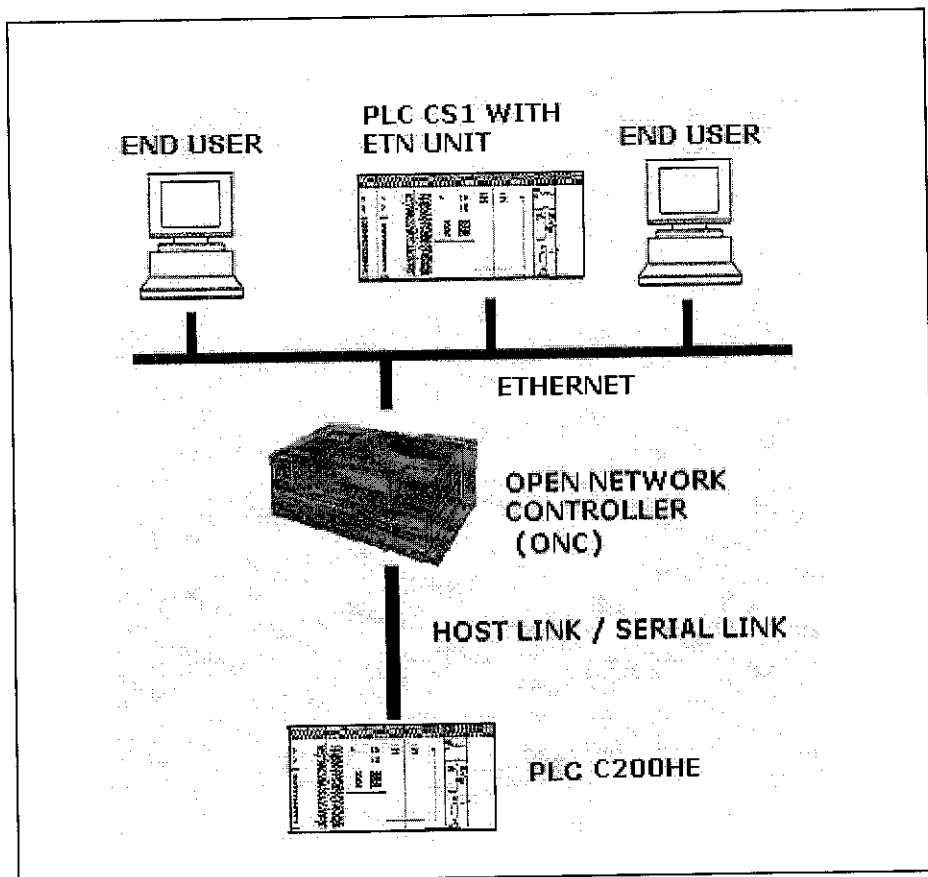
As an example of conversion, take the IR address 246.03. From the table, decimal '03' corresponds to '2<sup>3</sup>', which is; by the normal to-the-power-of-two conversion is '8'. Therefore, its SYSMAC C coding would be "0008". If the IR address is 200.15, the SYSMAC C coding would be "8000". From the table, we can see that 16 sub addresses assigned to an IR address; depicts by its last two decimal numbers.

During the development of PC-based system and HMI, the author discovers that the CX-Programmer and VB 6.0 cannot be use simultaneously. Using both at the same time will cause the failure of either one in detecting the PLC; hence failure to communicate. Therefore, it is important to transfer the ladder diagram to the PLC first before HMI is activated. Below is the summary of step-by step procedure to start a PC-based system with HMI;

1. Download ladder diagram to the PLC. Let the PLC to be in the Monitor mode and exit CX Programming.
2. Activate FINSGateway from **FINS Gateway Service Manager**.
3. Activate HMI and PLC is ready to be controlled from the HMI.

## 4.3 Development of Web-Based System

### 4.3.1 System Architecture



**Figure 4.11: Web –Based System Architecture**

The Web-based system architecture is depicted in Figure 4.11. It consists of a C200HE PLC and a CS1 PLC with Ethernet Unit (ETN Unit), an Open Network Controller (ONC), and two PCs.

Two PLCs are required as the Motion Controller Card is only compatible with C200H series PLC, whilst the ETN Unit is only compatible with CS1 series PLC. The Motion Controller Card is normally used with any robotic system, e.g. the XY Robot system. The ONC acts as a web server and the ETN Unit is placed in order to enable the PLC to support a wide range of protocols via Ethernet, including TCP/IP and FINS message communication.

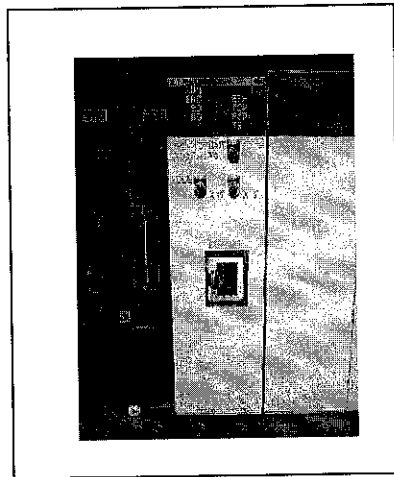


C200HE is connected to ONC through Host Link or serial communication. ONC, CS1 with ETN Unit and end users are connected to each other via the Ethernet or LAN.

The system works this way; a robotic system or XY Robot system for this matter, is being programmed into PLC C200HE. Its HMI is being launched as a web application and stored in the ONC. Whenever an end user needs to monitor or control the XY Robot system, the user just need to log onto the server (ONC) web site. This is because ONC has web server function that allows interactive monitoring from a web browser. Any end user with a standard web browser like Microsoft Internet Explorer can access the web site.

### **4.3.2 Device Installation and Settings Parameter**

#### **4.3.2.1 Ethernet Unit (CS1W-ETN 11)**



**Figure 4.12: ETN Unit**

The function of ETN Unit is to enable PLC supports a wide range of protocols via Ethernet. The protocols that it can support are TCP/IP, UDP/IP socket services, FINS command execution, FTP file transfers and SMTP message communication. This allows the communication or data transfer between two PLCs or PLC with end users over the Ethernet connection.

In order to use the ETN Unit, it must be connected to an Ethernet Connection or LAN. Beforehand, a few settings and installation must be done onto the module. Its procedures are summarize as below;

1. Determine the local IP address and the address conversion method.
2. Set the unit number of the ETN Unit as a CPU Bus Unit.
3. Set the node address.
4. Set the local IP address.
5. Mount the Unit to the PLC CPU Unit's Backplane.
6. Turn ON power supply to the CPU Unit.
7. Assume PLC communicates with PC through COM port using RS 232 cable.  
Create the I/O table using CX-Programmer.
8. Create the routing table with CX-Net.
9. Make other necessary settings in CX-Programmer such as changing the type of network use to communicate between PC and PLC.
10. Pull out RS 232 cable and plug in RJ 45 connector cable into its Ethernet connector.

• *ETN Unit's Settings Parameter*

The **Unit No.** is use to identify individual CPU Bus Unit when more than one CPU Bus Unit is mounted to the same PLC. Since there are no other network module mounted on the same PLC for this project, the **Unit No.** can be of any number. The ETN module **Unit No.** is set to '0'.

The **Node No.** identifies the devices attached to the same network. Its switches allow node number to be set from 01 to 7E hexadecimal or 1 to 126 decimal. For this project, since the automatic generation method is used for IP address conversion, the **Node No.** has to be the same with its Remote IP address **Host No.** Therefore, ETN Unit's **Node No.** is set to 05. (Further explanation regarding automatic generation method will be done in the following section)

- ***ETN Unit's IP Address Setting***

The nodes on LAN are identified by IP addresses. Before the ETN unit is connected to LAN, it is necessary to determine its IP address.

There are three methods that can be use to do this;

1. Automatic Address Generation
2. IP Address Table
3. Combined method ( using both automatic generation and I/O address table)

These methods basically convert FINS addresses into IP addresses. As FINS communication is being used as the communication function, it is necessary to specify the nodes according to the FINS address system. However, in this project, data must be sent and received on LAN using IP addresses. Hence, the IP addresses need to be converted from FINS addresses.

Based on the project's system architecture, whereby not many devices (or nodes) connected to the LAN; the most suitable method would be Automatic Address Generation. This method is suitable for network with 127 or less nodes connected to it. The conversion formula is shown as below;

$$\text{Remote IP address} = [\text{Local IP address AND Subnet mask}]$$

*OR remote FINS Node No.*

The ETN Unit's Remote IP address is achieved through the following calculation;

Local IP address :160.0.44.8  
 Subnet mask : 255.255.252.0  
 Remote FINS Node No. : 5

$$\begin{array}{r}
 160. 0. 44. 8 \\
 \text{AND } 255.255.252.0 \\
 \hline
 160. 0. 44.0 \\
 \text{OR } \qquad \qquad \qquad 5 \\
 \hline
 160. 0. 44. 5 \text{ (IP address)}
 \end{array}$$

When using this method, the ETN Unit's **Host No.** (of its IP address) and its **Node No.** must be set to the same value.

Therefore,

**Host No.**

IP address : 160.0.0.5

↑  
Becomes 0

The eight rotary switches at the back panel of the module are used to set its local IP (160.0.44.5). However, the rotary switches can only accept hexadecimal numbers. Therefore, the IP address needs to be converted into hexadecimal first. Table 4.2 shows the converted digits.

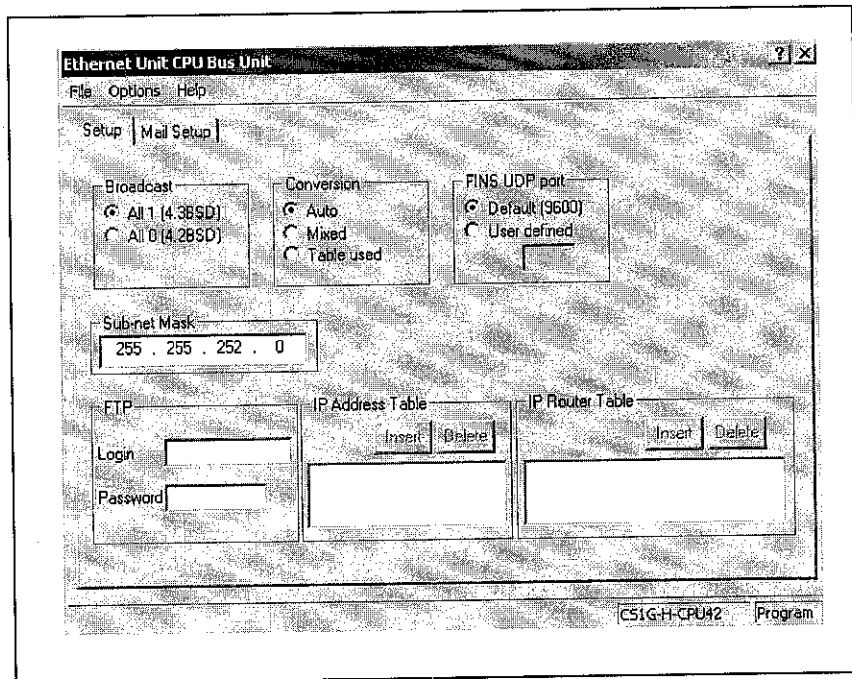
**Table 4.2: IP address conversion**

IP address in decimal	IP address in hexadecimal
160.0.44.5	A0. 0. 2C. 5

• **ETN Unit's I/O Table Setting**

To create an I/O table, create a new project in the CX Programmer. Before that, terminate **FINS Gateway Service Units; ETNUnit and SerialUnit**. If this is not being done, CX Programmer is unable to detect the connected PLC.

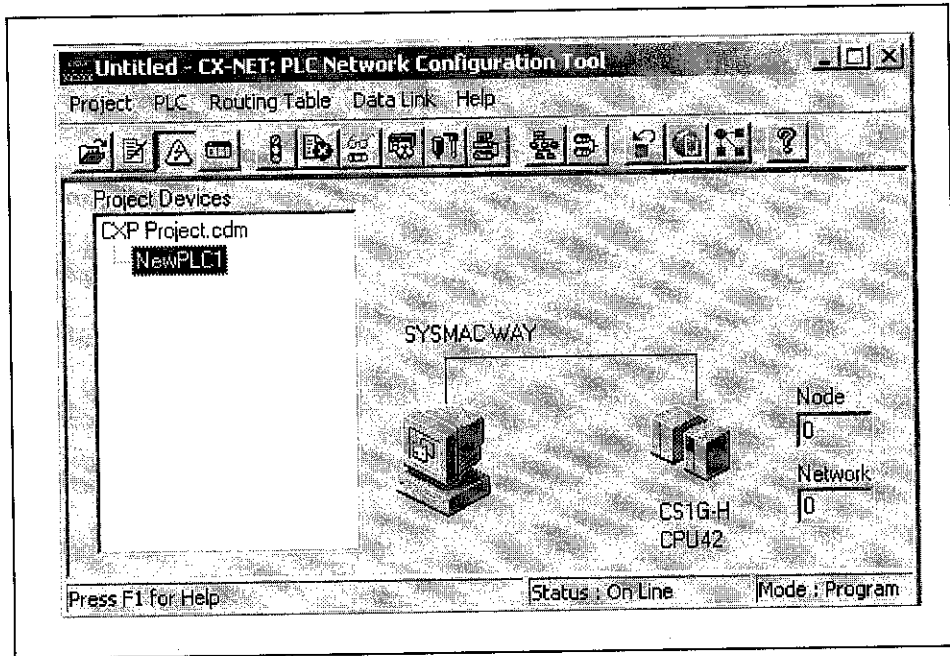
Right click at the I/O Table icon on the left side of the screen to give a pop up. Under 'Option' on its menu, choose 'Create'. Click at 'Main Rack' and next at 'ETN UNIT'. The window as shown in Figure 4.13 will appear. Insert the required settings such as its subnet mask. Transfer the settings to PLC.



**Figure 4.13: ETN Unit CPU Bus Unit**

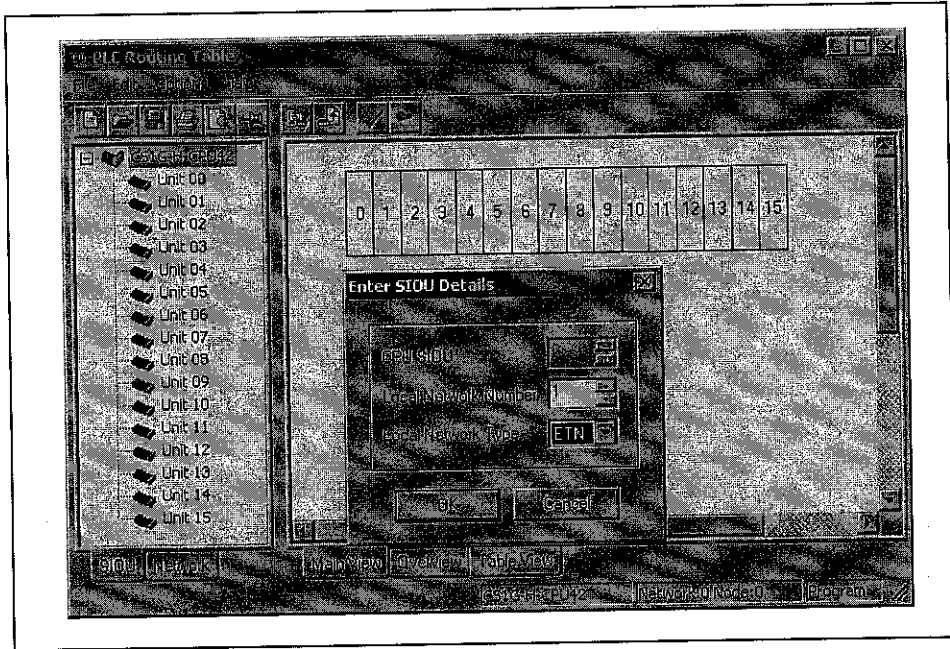
• **ETN Unit's Routing Table Setting**

After I/O table is downloaded to PLC, go to 'Tools' on the menu at the main page and select 'Network Configuration Tools'. A window as shown in Figure 4.14 will appear. Select 'New PLC' and click the online icon. The red line connecting PC to PLC as depicts in Figure 4.14 will turn to green.

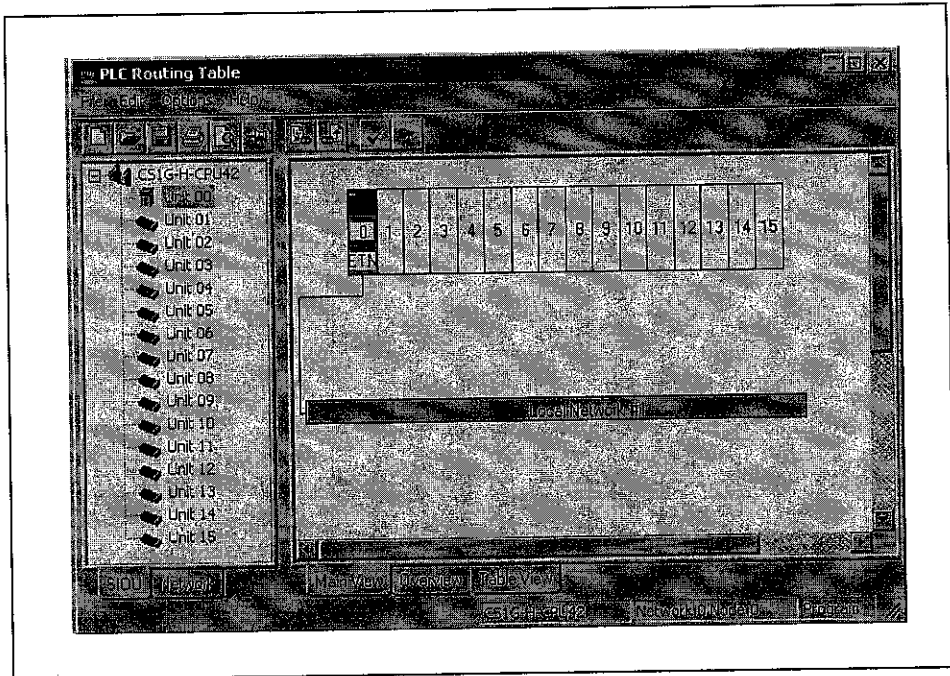


**Figure 4.14: Network Configuration Tools**

Next, choose 'Routing Table' and select 'Setup'. A window as Figure 4.15 will appear. Right click on column depicts '0' (the unit number of the ETN Unit), and a pop up will ask to insert 'Network number' and 'Type of network'; as depicts in Figure 4.15. Key in '1' and 'ETN' respectively. If it matched with the settings on ETN Unit module, the result will be as Figure 4.16. Download routing table to PLC.



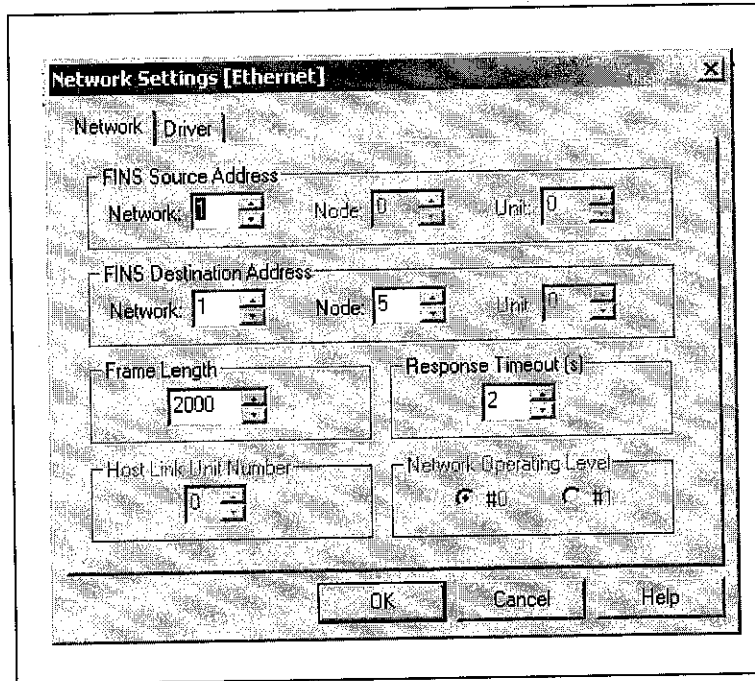
**Figure 4.15: Routing Table Setup**



**Figure 4.16 Routing Table**

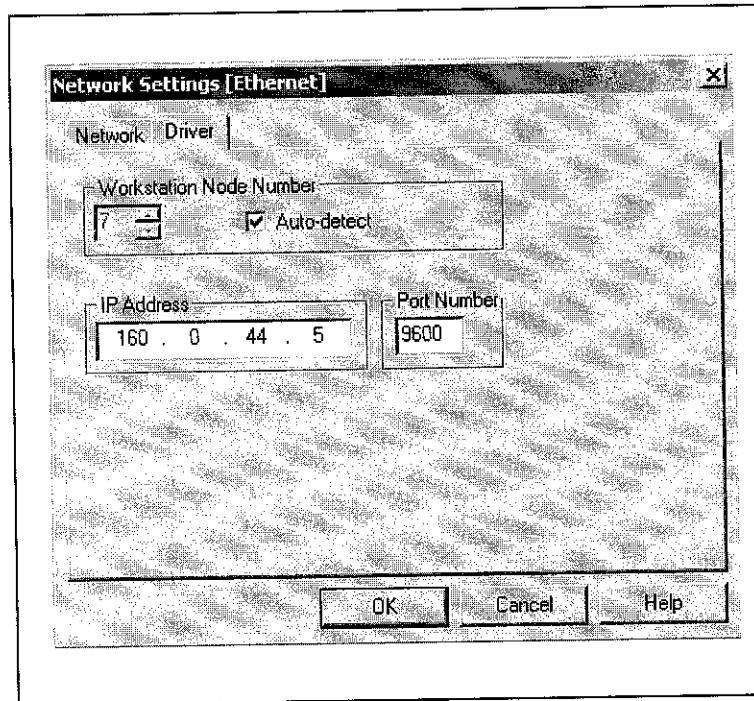
- **Changing Serial Communication to LAN Connection**

The next step is to change the RS 232 cable to RJ 45 cable. Make the PLC 'Offline' and pull out the RS 232 cable. Change the PLC 'Network Settings' by clicking onto PLC icon. Change 'Network' from 'SYSMAC WAY' to 'Ethernet'. Next, select 'Details'. Under 'Network', do the settings as Figure 4.17 and under 'Driver', do the settings as Figure 4.18.



**Figure 4.17: Network Settings**

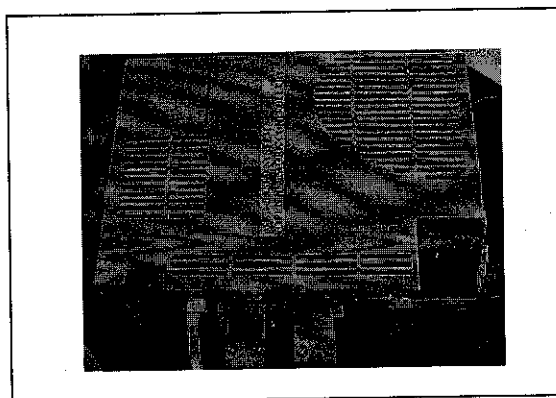




**Figure 4.18: Driver Settings**

Place the RJ 45 cable into the Ethernet connector on the ETN unit. Make the PLC in 'Online' mode. The process is successful if the CX Programmer is capable to detect the CS1 PLC via LAN or Ethernet.

#### ***4.3.2.2 Open Network Controller (ONC ITNC-EIX01)***



**Figure 4.19: Open Network Controller**

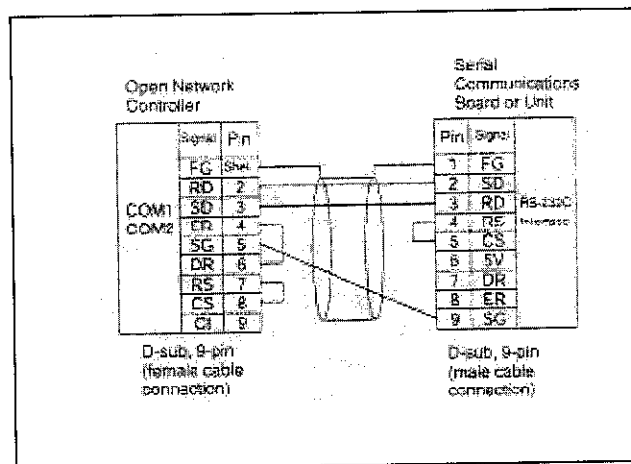
Open Network Controller (ONC) provides a gateway for FINS message communication between the host PC and CS1 PLC on the LAN to C200HE PLC via Host Link in this

project (Figure 4.11). Its web server function allows remote controlling or monitoring of a system from a web browser. The objective is to let any user who has access to the website to control a system from distance.

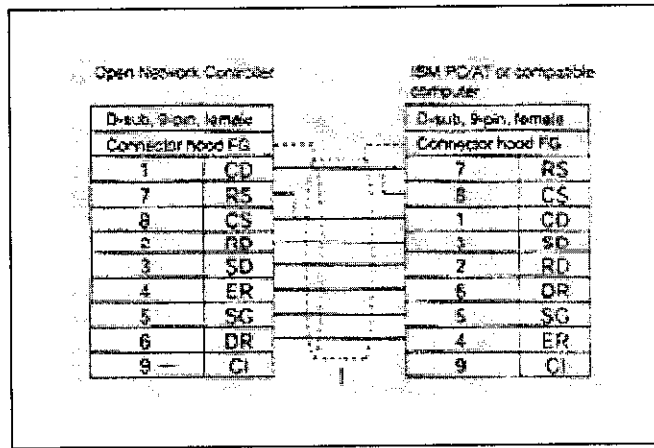
The ONC need to be configured and that is done by doing some files editing. The files that need to be edited are its settings files. The list of files can be viewed as Appendix 4. Basically, there are two methods that can be use to edit this file; using the **ONC\_wizard** setting tool or **vi editor**. **ONC\_wizard** allows files to be edited in the conversational window display whilst **vi editor** make settings changes in the settings files' temporary file. For this project, ONC was configured using the **ONC\_wizard** setting tools method.

Beforehand, two serial link cables need to be constructed. For PC –ONC communication (connection use while editing settings files), a cable of pin layout as shown in Figure 4.20 is needed.

For ONC- PLC C200HE communication (Host Link connection), a cable of pin layout as shown in Figure 4.21 is needed.

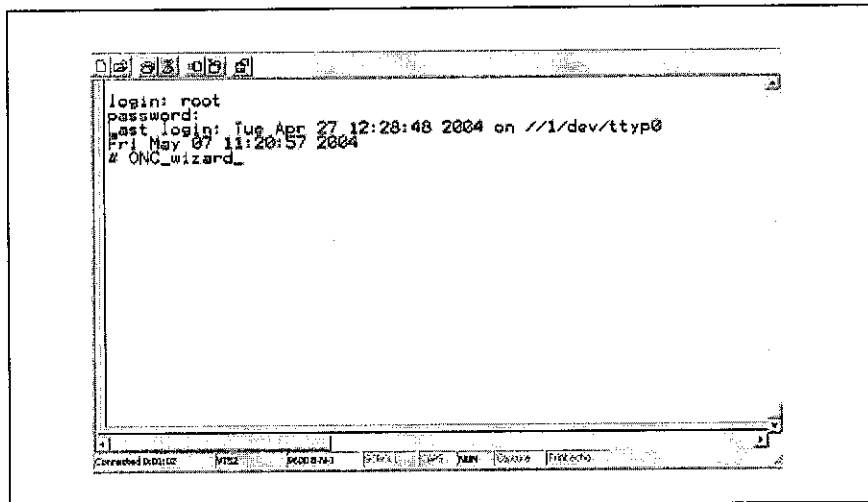


**Figure 4.20: Pin Layout for PC-ONC Communication**



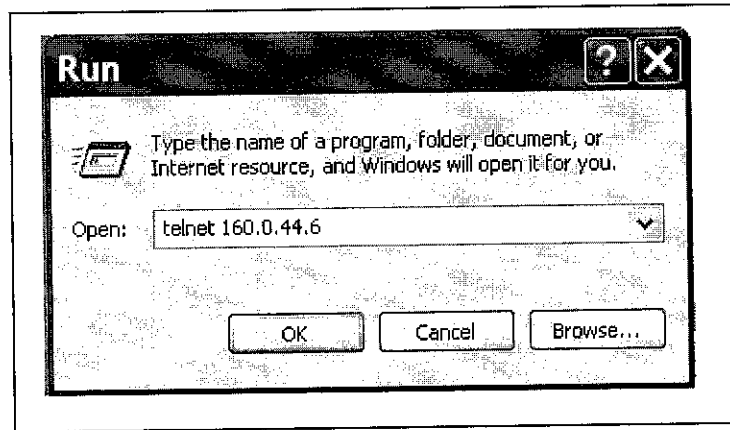
**Figure 4.21: Pin Layout for ONC- PLC C200HE Communication**

File editing can be done by logging onto the ONC by using Hyper Terminal connected to **COM 1**. If PC- ONC communication is successful, a window as depicts in Figure 4.22 will appear. Log onto the ONC as 'root' and default password 'OMRON'. A # key will appear. Type 'ONC\_wizard' to access settings files.

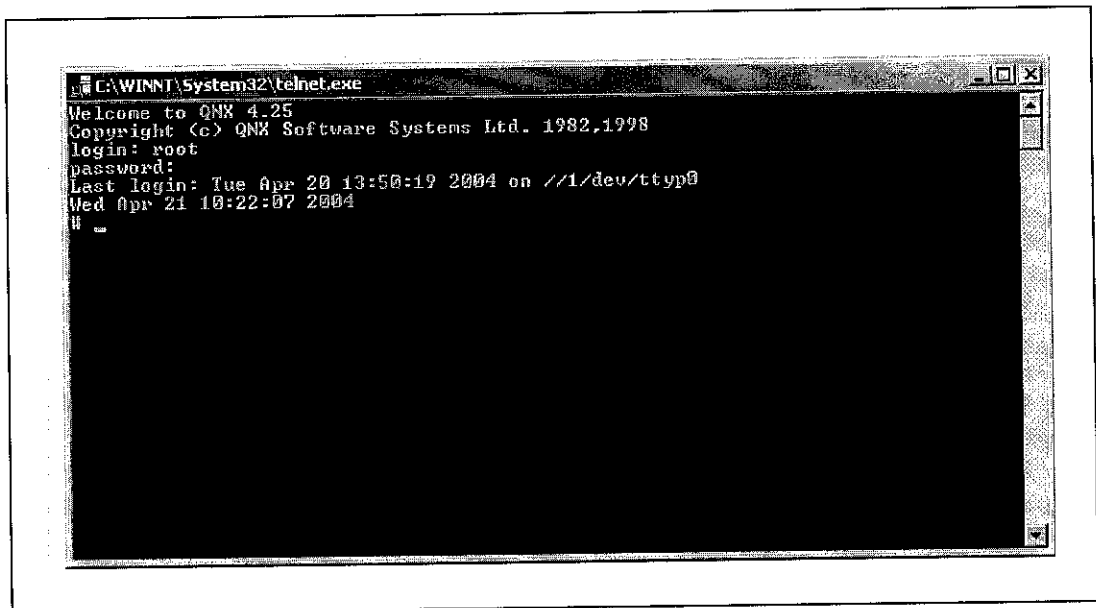


**Figure 4.22: Hyper Terminal**

After some editing done, especially it's IP address at SYSTEM SETUP; ONC can be accessed by connecting it to Ethernet port using Telnet (Figure 4.23). The IP address of ONC has been set to 160.0.44.6. If PC- ONC communication via Ethernet is successful, a window as depicts in Figure 4.24 will appear. Log onto the ONC as 'root' and default password 'OMRON'.



**Figure 4.23: Accessing ONC via Ethernet**



**Figure 4.24: Logging onto ONC via Ethernet**

### **4.3.3 Web Application Development**

Since ONC is used as the server, OMRON's web application development software; ONC Active X Control and FA Components (Active X Control) are required. These software cannot be used other than applications communicating with an ONC. ONC Active X Control contains VB 6.0 OCX component, which can be pasted onto VB 6.0's form to enable use. Combining FA components with VB 6.0 components allow the development of a HMI that can be operated as a web application.

To enable ONC to use the ONC Active X components, a software called ONC Data Agent must be installed. Please refer Appendix 5 for the mentioned software installation.

#### 4.3.4 Discussion

##### 4.3.4.1 Ethernet Unit (CSIW-ETN 11)

Basically, ETN Unit installation was successful. This is based on communication established between the PC and PLC CS1. This is verified when PING command was done (Figure 4.25). The IP address of the PLC CS1 with ETN Unit is 160.0.44.5.

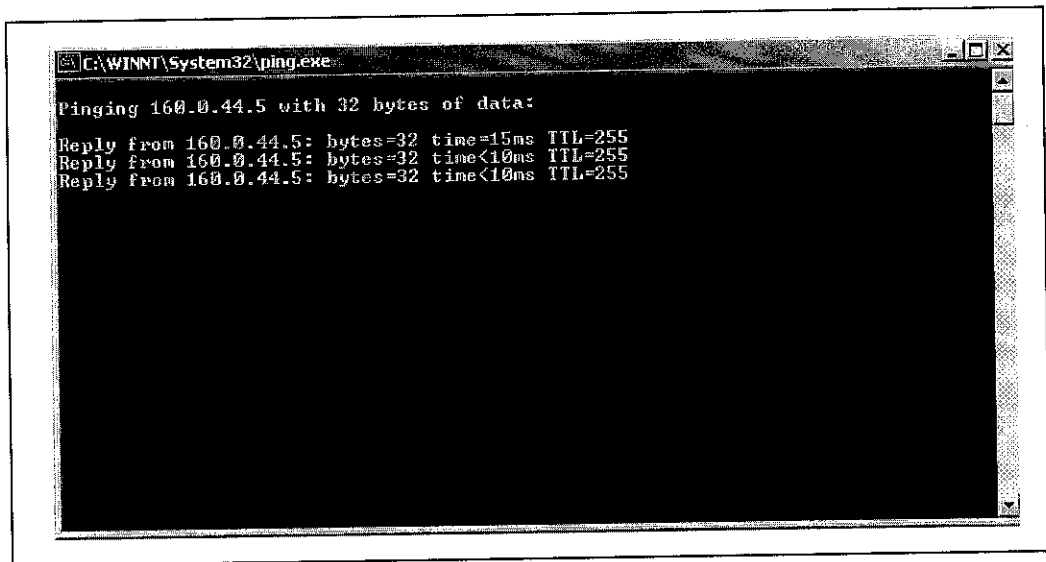


Figure 4.25: Ping on PLC CS1

Another way to verify communication is by executing the **FINSGateway Network Tester**. The **Network**, **Node** and **Unit No.** of the ETN Unit was inserted and the following result, as depicts in Figure 4.26 was obtained. The encircled information proved that FINSGateway detects PLC. The **Unit No.** for Ethernet network has been set to '17' by FINSGateway configuration.

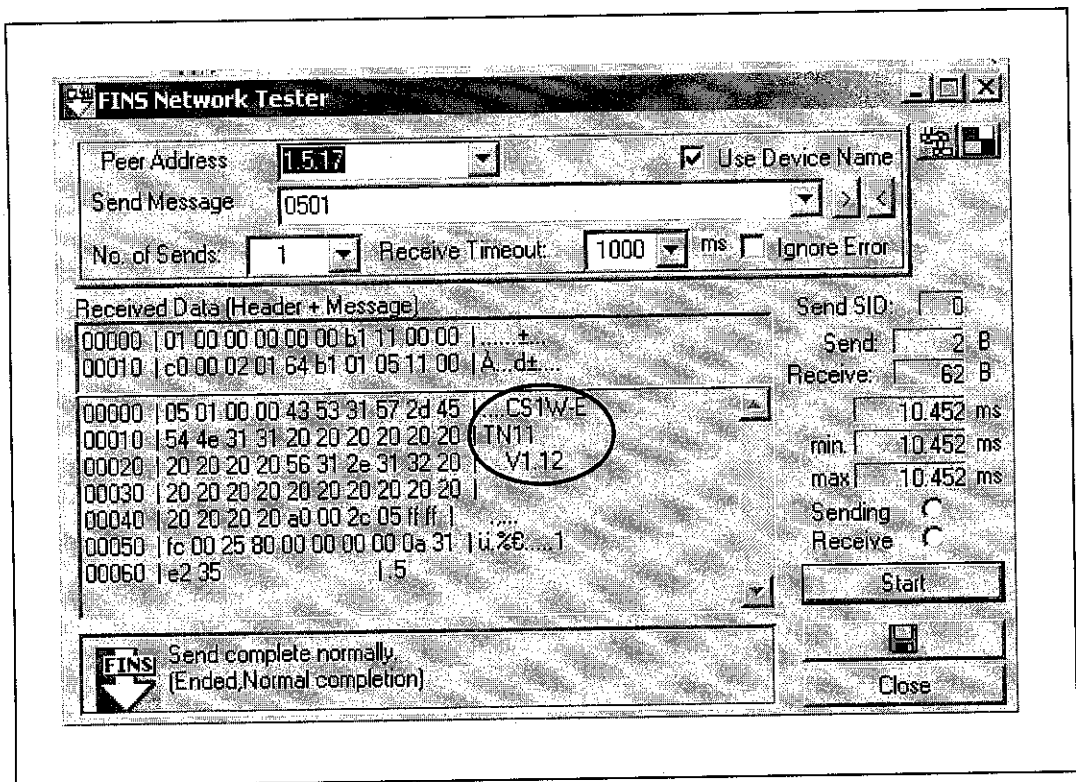


Figure 4.26: FINSGateway Network Tester on PLC CS1

With the successful communication, ETN Unit can be use and therefore, will enable data transfer to and from PLC CS1 over the Ethernet network.

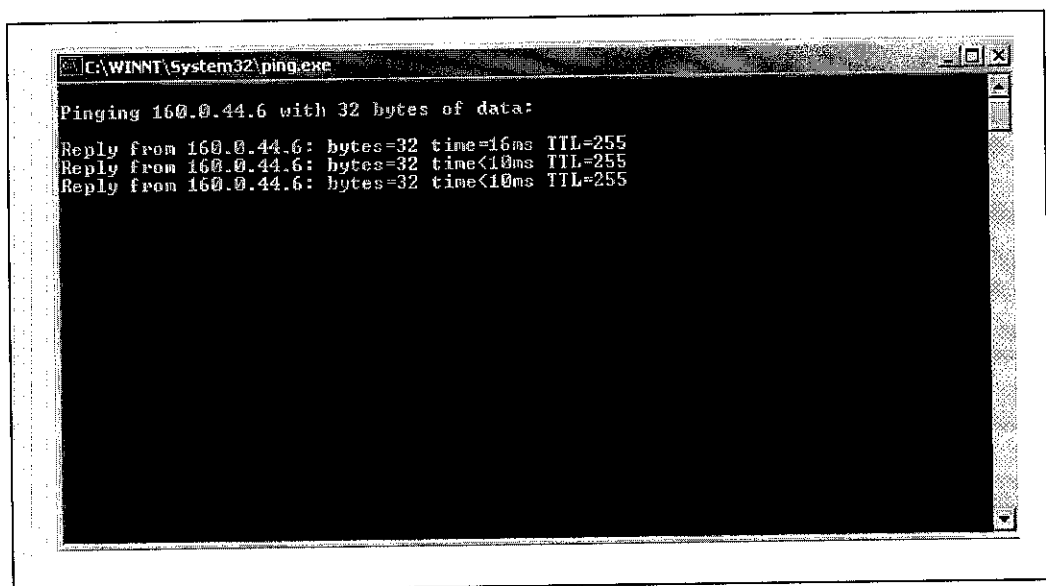
According to its manual, ETN Unit can transfer two types of data; FINS commands or any data. To transfer FINS command, **FINS communication** is use. To transfer any data, **Socket services** is use. Basically, both are ETN Unit's communication function. For **FINS communication** service, data is sent and received as UDP packets. As for **Socket services**, data is sent and received as either UDP or TCP protocol. This means **Socket services** have wider application.

The ETN Unit configuration done above was meant for FINS command and **FINS communication** service was used. This is because the Web-based system is a simple network configuration; whereby the network is small and only integrate two kinds of network; Ethernet and Host Link. For a small network, data is transferred using only a few protocol; SYSWAY protocol for Host Link and UDP protocol for Ethernet.

The system can be expanded into a larger scale network and integrate many type of OMRON Factory Automation Network; Controller Link, Host Link, Device Net and more. With this, a lot of data will be transferred and **Socket services** communication function will come in handy. To use its function, the ETN Unit installation and settings will be different a bit. To do so, reference to its manual is required.

#### **4.3.4.2 Open Network Controller (ONC ITNC-EIX01)**

Basically, ONC installation was successful. Settings files were edited by inserting the necessary parameter. To view the settings files, please refer Appendix 4. As the result, ONC can be accessed via Ethernet (Figure 4.24). This is further verified when PING command was done (Figure 4.27). The IP address of ONC is set to 160.0.44.6.



**Figure 4.27: Ping on ONC**

However, ONC-PLC C200HE communication, or Host Link network cannot be verified. The cable use was constructed according to its specification and it should be able to detect PLC C200HE. The possible problem could be the wrong settings in its settings file; HLK SETUP (Appendix 4). The author is still working on this.

Data Agent, however, cannot be installed as the software is not available. The software must be purchased from OMRON.

#### ***4.3.4.3 Web Application Development***

The author has managed to understand the procedure of creating a web application of HMI although did not manage to build an actual HMI. This is because web application development requires software (ONC Active X Control and FA Components) that are not available in UTP and must be purchased from OMRON.

Given the actual XY Robot System, its HMI can be created using the mentioned software. To build a more sophisticated HMI, much knowledge in programming the VB 6.0 must be acquired.

Overall, the settings of important devices are successful and ready to be use. The matter that hindered its positive progress is the unavailability of software required. These software must be use in developing the web application of HMI.

For future development, the Web-based system can be expanded into a larger network. This means, more nodes can be added and more types of network can be integrated. However, it is better to have a dedicated LAN connection. This is to reduce time delay due to network congestion.



## **CHAPTER 5**

### **CONCLUSION AND RECOMMENDATION**

With reference to its Gantt chart, the development of PC-based and Web-based system is successful; however HMI cannot be created fully due to a few problems. Although the author took a longer period to configure both ETN Unit and ONC, it did complete on time though. Technical guidance from OMRON personnel helps a lot and the readings from its manual do facilitate the author in understanding its step by step procedures explained by the personnel. Hence, these contribute to the success of developing both PC-based and Web-based system. As for the development of HMI, the author did not make a HMI controlling XY robot, but manage to grasp with the concept of creating a HMI. This is proven with the success of lighting an LED from a simple VB program. Therefore, given the extra time, software required and the XY Robot System program, there will not be much of a problem of creating the actual HMI. Basically it can be concluded that objective number 1 and 2 is achieved and close at realizing objective number 3.

This project is viable but it requires more than literature understanding. It requires professional guidance as well. For example, to install and configure the ETN unit alone, the author finds it difficult by just referring to its manual and not getting technical assistance from the OMRON personnel. The steps layout in the manual are somehow not direct, requires understanding of other device or software, reference to other manual and are often left with so many choices of methods to do it. This is because the project itself consists of many software and devices. Therefore, professional help is required in clarifying doubts. It is recommended that, in a project such as this; where it involves unfamiliar devices, a briefing or demonstration from the manufacturer is highly needed. Technical advice on tools needed; especially regarding its software requirement, is also necessary.

## REFERENCES

1. Hu,H., Yu,Lixiang., Pui,WT. and Quan,Zhou.2001. *Internet –based Robotic Systems for Teleoperation*. IEEE Robotics and Automation Magazine , July, pp.143-151.
2. Goldberg ,Ken. and Mascha, Micheal. 1994  
<<http://www.usc.edu/dept/raiders/paper.>>
3. Thayer, Robert.1999, *Visual Basic 6 Unleashed Professional Reference Edition*, Indianapolis, SAMS .
4. Smythe, Robert. 1996, *Visual Basic Quick Tutor*, Massachusetts, Boyd & Fraser Publishing Company.
5. Stallings,Williams.1996, *Data and Computers Communications*, New Jersey, Prentice Hall.
6. Noor Meliza Mohd. Kamal . 2002 , *Web-based and PC-based System for Controlling Automatic Sorting System* , Final Year Thesis , University Teknologi Petronas , Malaysia.
7. “SYSMAC Compolet Version 2 User Manual”, a manual by OMRON Corporation.
8. “SYSMAC CS1 Series CS1W –ETN 11 Ethernet Units Operation Manual”, a manual by OMRON Corporation.
9. “Open Network Controller ITNC-EIX01 Operation Manual“, a manual by OMRON Corporation.
10. Malinowski , Aleksander 2001 , *Controlling Robots via Internet* , Idaho; University of Idaho.
11. <[www.faweb.net/us/opc/finsgateway.html](http://www.faweb.net/us/opc/finsgateway.html) >
12. <[www.plcsoft.ne.jp/soft/Eng/Support/FinsGateway/FGDefaultSE.shtml](http://www.plcsoft.ne.jp/soft/Eng/Support/FinsGateway/FGDefaultSE.shtml)>
13. <[www.micromech.co.uk/dir\\_products/pdf/omron/plc\\_cs1.pdf](http://www.micromech.co.uk/dir_products/pdf/omron/plc_cs1.pdf)>
14. <[www.idom.ru/files/Omron/FA/Manuals/Software/FinsGateway%20V2%20Manual%20Runtime%20Eg.pdf](http://www.idom.ru/files/Omron/FA/Manuals/Software/FinsGateway%20V2%20Manual%20Runtime%20Eg.pdf)>
15. <[www.idom.ru/files/Omron/FA/Manuals/Software/SYSMAC%20Compolet%20Version2%20EG.pdf](http://www.idom.ru/files/Omron/FA/Manuals/Software/SYSMAC%20Compolet%20Version2%20EG.pdf)>

## **APPENDICES**

APPENDIX 1: FLOW CHART OF WORKING PROCEDURES FOR THE PROJECT

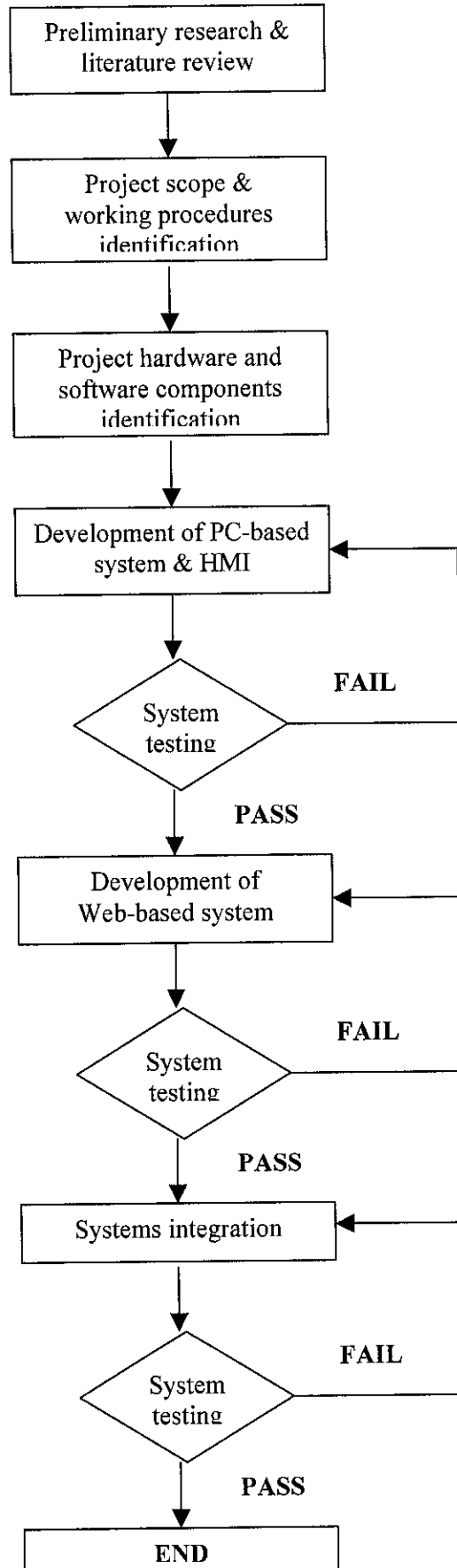
APPENDIX 2: PROJECT GANTT CHART FOR SEMESTER 1

APPENDIX 3: PROJECT GANTT CHART FOR SEMESTER 2

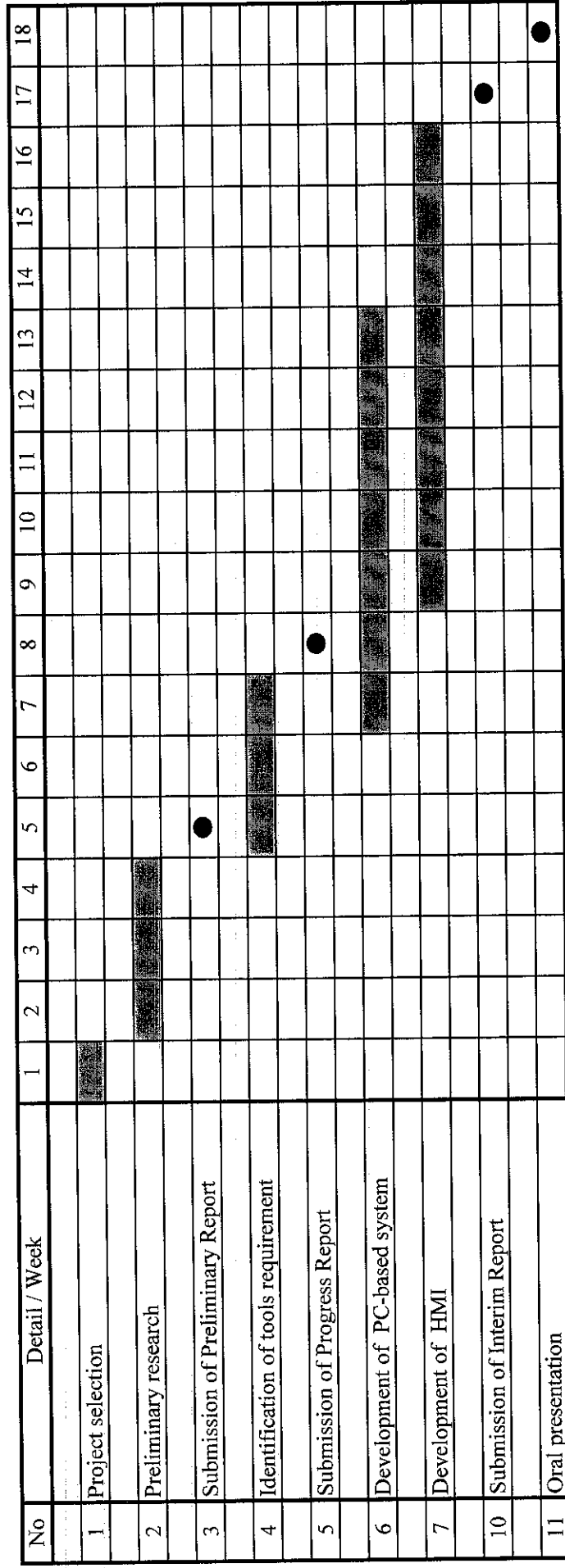
APPENDIX 4: ONC\_WIZARD SETTINGS FILES

APPENDIX 5: WEB APPLICATION SOFTWARE INSTALLATION PROCEDURES

**APPENDIX 1: FLOW CHART OF WORKING PROCEDURES FOR THE PROJECT**



APPENDIX 2: PROJECT GANTT CHART FOR SEMESTER 1 (JULY 2003-NOVEMBER 2003)

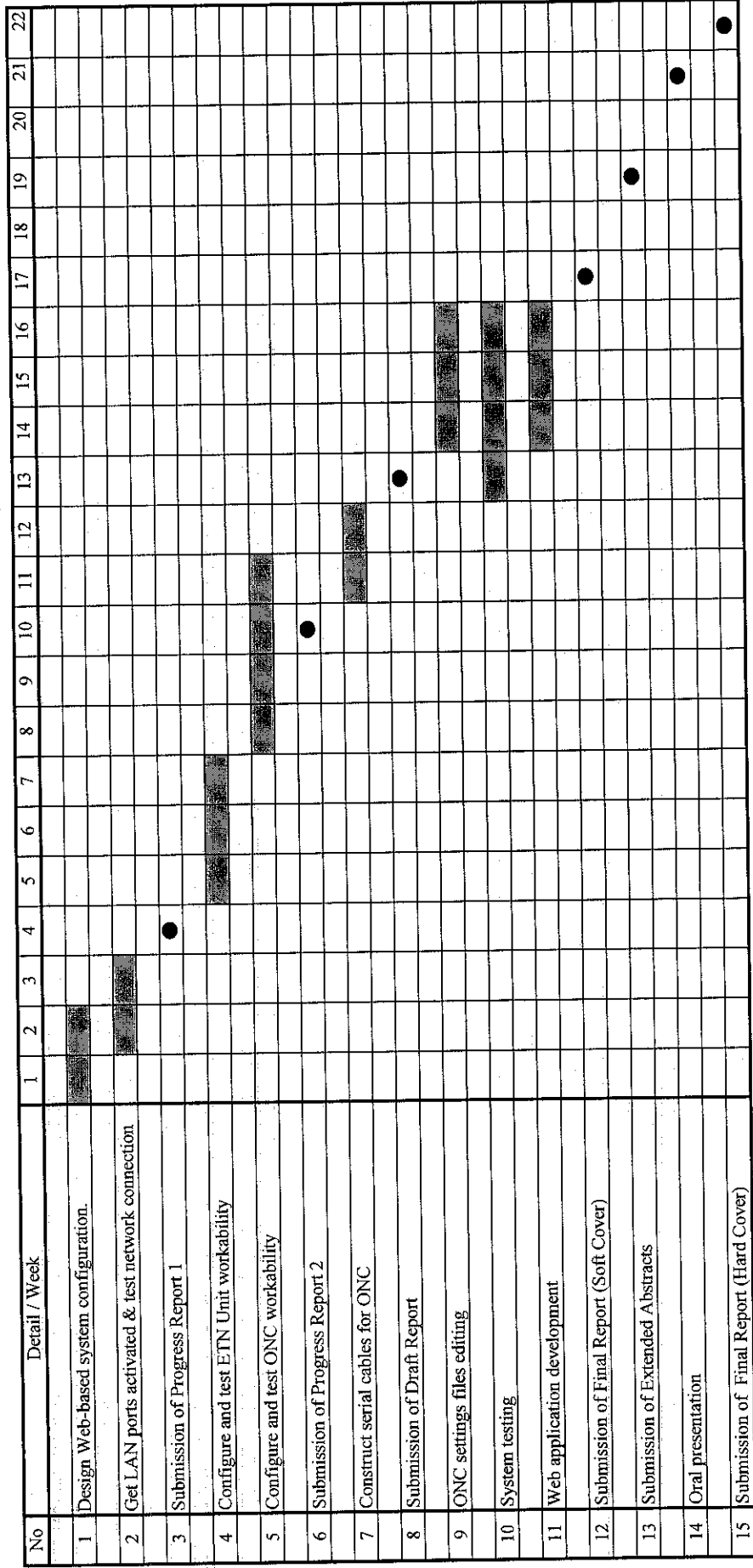


Process



Submission / Presentation

APPENDIX 3: PROJECT GANTT CHART FOR SEMESTER 2 (JAN 2004-MAY 2004)



█ Process

● Submission / Presentation

## APPENDIX 4: ONC\_WIZARD SETTINGS FILES

Main Menu	Usage	Menu item	Settings	Settings file
1. SYSTEM SETUP	Ethernet settings	1. HostName	Local node host	/etc/netstart
		2. SubnetMask	Subnet mask	
		3. IpRoute	IP router addresses	/iproute
		4. Host File	Host IP addresses	/etc/hosts
		5. exit	Goes to Main Menu.	---
2. RESOURCE SETUP	COM port settings	1. COM PROCESS COUNT	Number of COM ports	/etc/FgwQnx/FgwQnx.ini
		2. COM1 PROCESS	NP allocation to COM1	
		3. COM2 PROCESS	NP allocation to COM2	
		4. COM3 PROCESS	NP allocation to COM3	
		5. exit	Goes to Main Menu.	---
3. FinsGateway SETUP	FinsGate-way settings	1. StartService	Startup services	/etc/FgwQnx/FgwQnx.ini
		2. UnitID	Unit addresses	
		3. LocalNetworkTable	Local network table	
		4. RelayNetworkTable	Relay network table	
		5. MemoryAliases_DM	DM area settings	
		6. MemoryAliases_CIO	CIO area settings	
		7. exit	Goes to Main Menu.	---
4. ETN SETUP	Ethernet NP settings	1. NodeID	FINS node address	/etc/FgwQnx/FgwQnxEtn.ini
		2. IpTable	FINS node-IP address table	
		3. exit	Goes to Main Menu.	---
5. CLK SETUP	Controller Link NP settings	1. NodeID	Node addresses	/etc/FgwQnx/FgwQnxClk.ini
		2. BaseAddress	Base address display	
		3. IRQ	IRQ display	
		4. SystemSw	Baud rate	
		5. exit	Goes to Main Menu.	---
6. SYSMAC SETUP	SYSMAC Board NP settings	1. NodeID	Node address	/etc/FgwQnx/ FgwQnxSysmac.ini
		2. Startmode	Startup mode	
		3. SysmacDriver SETUP	I/O port address display and mailbox interval	/etc/FgwQnx/ FgwQnxSysmacDriver.ini
		4. SysmacMapping	Event memory allocations	/etc/FgwQnx/ FgwQnxSysmacMapping.ini
		5. exit	Goes to Main Menu.	---

Main Menu	Usage	Menu item	Settings	Settings file	
7. DeviceNet SETUP	DeviceNet NP settings	1. NodeID	Local MAC ID	/etc/FgwQnx/FgwQnxDrm.ini	
		2. Baudrate	Baud rate		
		3. Scanlist	Scan list	/etc/FgwQnx/scanlist.ini	
		4. ScanCycleTime	Scan cycle time	/etc/FgwQnx/FgwQnxDrm.ini	
		5. StartScanMode	Startup scan mode		
		6. ComErrorStop	Remote I/O communications handling for communications errors		
		7. EventCycleTime	Event memory refresh interval		
		8. InAreaName1	Input area 1 area		
		9. InAreaOffset1	Input area 1 first word		
		10. InAreaSize1	Input area 1 size		
		11. InAreaName2	Input area 2 area		
		12. InAreaOffset2	Input area 2 first word		
		13. InAreaSize2	Input area 2 size		
		14. OutAreaName1	Output area 1 area		
		15. OutAreaOffset1	Output area 1 first word		
		16. OutAreaSize1	Output area 1 size		
		17. OutAreaName2	Output area 2 area		
		18. OutAreaOffset2	Output area 2 first word		
		19. OutAreaSize2	Output area 2 size		
		20. StatusAreaName	Area for status area		
		21. StatusAreaOffset	First word of status area		
		22. StatusAreaSize	Size of status area		
		23. exit		Goes to Main Menu.	--
8. HLK SETUP	C-series Host Link, CV-series Host Link, and CompoWay NP settings	1. Set Com1	1. NodeID	Node address	/etc/FgwQnx/FgwQnxHlk.ini
			2. ComSet	Communications parameter	
			3. TimeOut	Timeout time	
			4. RetryCnt	Number of retries	
		3. Set Com3	5. NetTblSet	Host Link address table	/etc/FgwQnx/HlkNetTbl.ini
		4. exit		Goes to previous level.	--
9. RUT SETUP	Reserved for future development. Cannot be accessed.				
10. exit	Accesses Exit Menu.	1. Discard Changes & exit			
		2. Save Changes & exit			
		3. Make Temporary File & exit			
		4. MainMenu	Goes to Main Menu.	--	



# 4-2 Data Agent

## Data Agents

To use the ONC ActiveX controls and ONC Java Beans, it is necessary to install a Data Agent to the Open Network Controller.

The Data Agent is the same for the ONC ActiveX controls and the ONC Java Beans, but is different for the HTTP and Socket types. Install the Data Agent for the protocol used to the Open Network Controller.

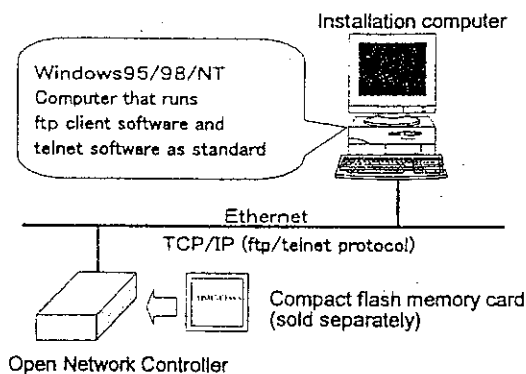
Installation File	Description
CD-ROM\DataAgents\webhttp101.tgz	HTTP protocol Data Agent
CD-ROM\DataAgents\websock100.tgz	Socket protocol Data Agent

The final 3 characters of the file name may change in various releases, and might not match with that shown here.

Transfer the appropriate file as shown above to the Open Network Controller, and install it.

## Before Installing the Data Agent

To install the Data Agent to the Open Network Controller, the following hardware and software is required. Confirm these conditions before attempting to install the Data Agent.



## Installation Computer

Item	Specifications
Computer, OS	Computer with Microsoft Windows95/98/NT
Network	Ethernet
Software	ftp client and telnet software

## Open Network Controller

A QNX format compact flash card must be inserted. Refer to the Open Network Controller manual (SBSB-504), 3-9, for details about formatting and inserting the compact flash card.

## Network Environment

The installation computer and the Open Network Controller must be connected by Ethernet, and ftp and telnet must be available on the computer. If an Ethernet connection is not already established, refer to the Open Network Controller manual (SBSB-504), Chapter 3 for details regarding the Ethernet connections and settings.

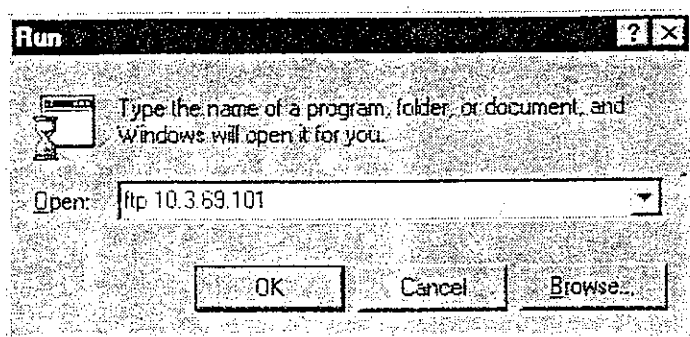
## Installing the Data Agent

Install the Data Agent to the Open Network Controller as follows (the  indicates the Enter key):

### Transferring the Installation File

- To install the Data Agent, first, transfer the installation file (webhttp101.tgz or websock100.tgz) from the computer to the compact flash card (/hd) in the Open Network Controller. Use a commercially available ftp client or the standard Windows ftp client to transfer the file to the Open Network Controller.
- In this example, the standard Windows ftp client is used to transfer the file (websock100.tgz).
- If using another ftp client, be sure to send the file in **binary mode**. Then, refer to the Installation explanation that follows.

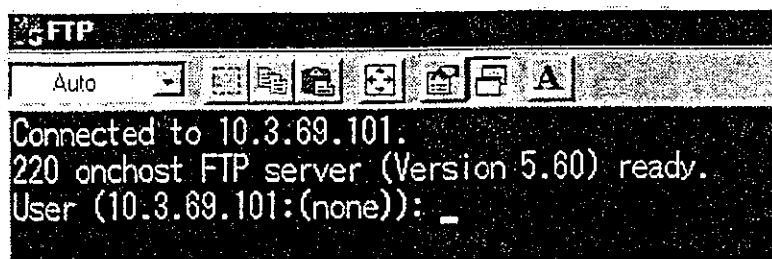
1 On the installation computer, select Run from the Windows Start menu, and enter the following:



Set the IP address of the Open Network Controller (default IP address: 10.0.0.1).

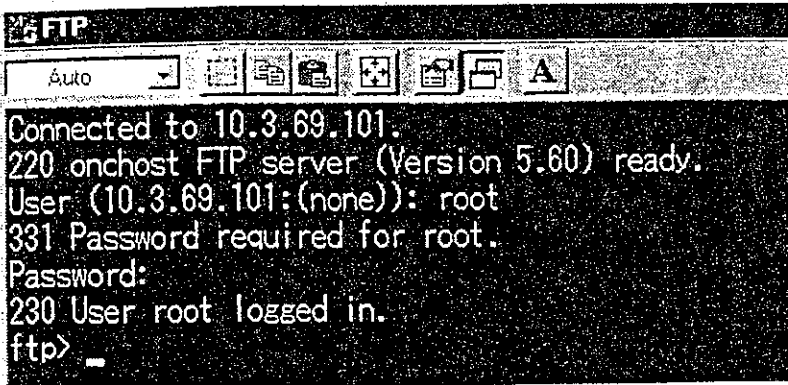
2 Click OK.

3 The Windows ftp client will be booted as shown below:



4 Log in as root. Enter root, and then the password. The default password is OMRON.

If the login is successful, the following will be displayed:



5 Transfer the installation file to the Open Network Controller compact flash card (/hd) as follows:

ftp>cd /hd

6 Set ftp to binary mode as follows:

ftp>binary

7 Transfer the installation file as follows (put CD-ROM drive\DataAgents\file name):

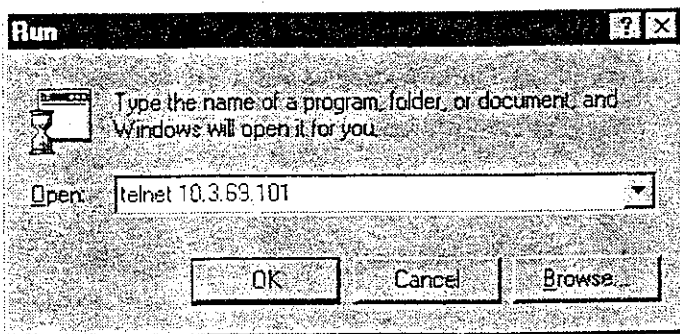
Example: ftp>put E:\DataAgents\websock100.tgz

This completes the transfer of the installation file. Refer to the Installation explanation next to complete the installation.

### Installation

First, connect from the installation computer to the Open Network Controller using telnet. Then, install the Data Agent. In this example, the Socket protocol Data Agent is being installed.

1 On the installation computer, select Run from the Windows Start menu, and enter the following:



Set the IP address of the Open Network Controller (default IP address: 10.0.0.1).

2 Click OK.

- 3 Log in as root. Enter `root`, and then the password. The default password is **OMRON**.  
If the login is successful, the following will be displayed:

```
Telnet - 10.3.69.101
Connect Edit Terminal Help
Welcome to QNX 4.25
Copyright (c) QNX Software Systems Ltd. 1982,1998
login: root
password:
Last login: Thu Sep 07 13:02:08 2000 on //1/dev/tty0
Fri Sep 22 15:21:45 2000
# █
```

- 4 Move to the directory with the `websock100.tgz` file (`/hd`):

```
#cd /hd
```

(For the HTTP Data Agent the file name is `webhttp101.tgz`.)

- 5 Unzip the `websock100.tgz` file to obtain the tar file:

```
#gunzip websock100.tgz
```

The `websock100.tar` will be unzipped.

- 6 Extract the `websock100.tar` file:

```
#tar -xvf websock100.tar
```

The `/hd/websock` directory will be created, and the files copied into it. (The HTTP Data Agent directory is `/hd/webhttp`.)

- 7 Move to the `/hd/websock` directory.

```
#cd /hd/websock
```

- 8 Begin the installation as follows:

```
#./setup
```

If the installation completes normally, the Data Agent files are all copied to the appropriate directories.

This completes the Socket protocol Data Agent installation.

After the installation is complete, move to the `/hd` directory and delete the `websock100.tar` file, as follows:

```
#cd ..
```

```
#rm websock100.tar
```

## Uninstalling the Data Agent

To uninstall the Data Agent, connect to the Open Network Controller from the computer, using telnet, and follow the procedure below:

- 1 Move to the `/hd/webhttp` or `/hd/websock` directory:

```
#cd /hd/websock
```

- 2 Uninstall as follows:

```
#./uninst
```

## 4-3 ONC ActiveX Controls

This section explains the installation of the ONC ActiveX controls.

Two types of ONC ActiveX controls are available: HTTP and Socket:

Installation File	Description
CD-ROM\ActiveXCtl\inst_http101.exe	HTTP ONC ActiveX controls
CD-ROM\ActiveXCtl\inst_socket100.exe	Socket ONC ActiveX controls
CD-ROM\ActiveXCtl\sample_http100.exe	HTTP ONC ActiveX controls sample project
CD-ROM\ActiveXCtl\sample_socket100.exe	Socket ONC ActiveX controls sample project

The final 3 characters of the file name may change in various releases, and might not match with that shown here.

Copy the above installation to the computer hard disk to install it.

Refer to the online documentation for details about using the ONC ActiveX controls.

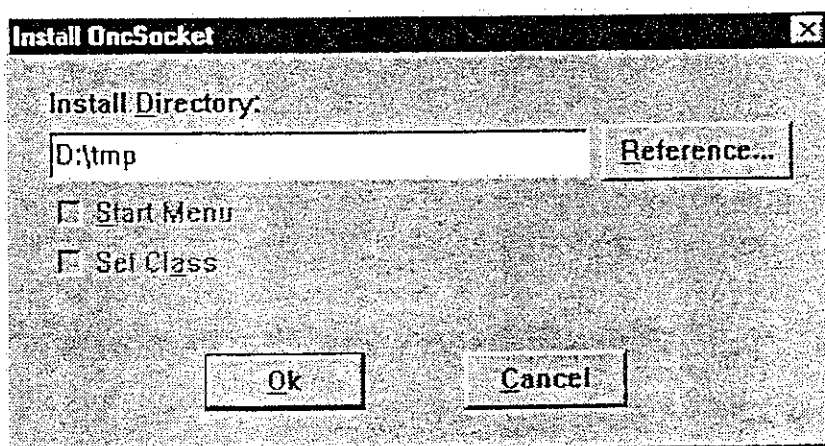
### Installing the ONC ActiveX Controls

The Socket ONC ActiveX controls are installed in this example. Install the HTTP ONC ActiveX controls or the sample projects in the same manner.

To use the ONC ActiveX controls, The Data Agent for the protocol to be used (HTTP or Socket) must be installed in the Open Network Controller. Refer to 4-2 for details.

- 1 Copy the inst\_socket100.exe file from the CD-ROM ActiveXCtl folder to the computer (in this example, D:\tmp).
- 2 Double-click the copied inst\_socket100.exe file.

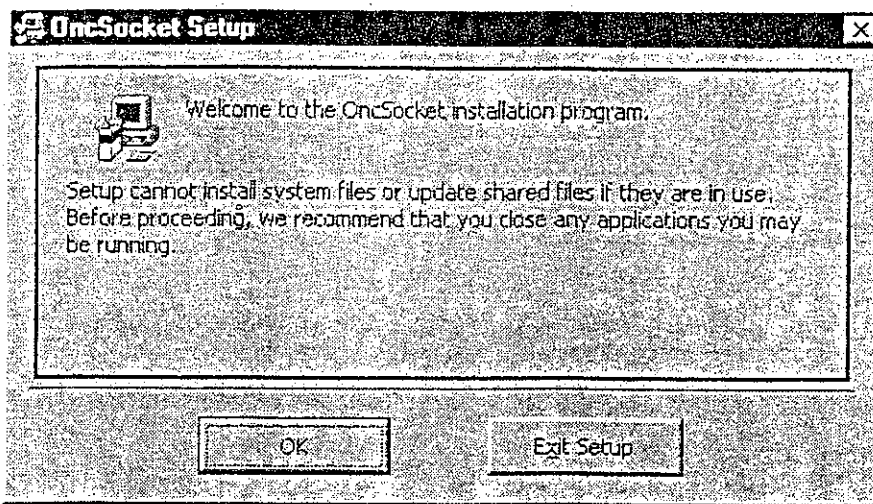
The following dialog will be displayed:



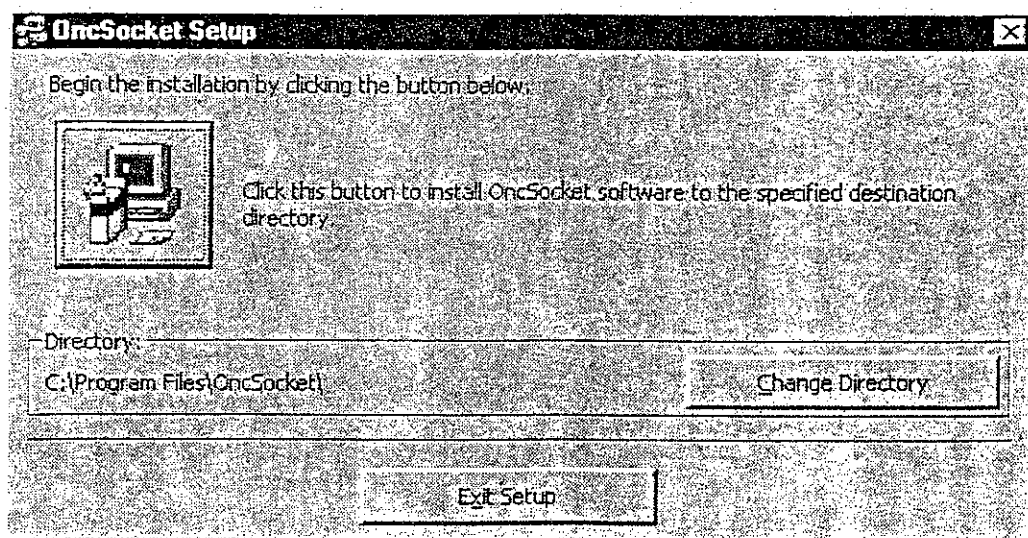
- 3 Click OK.

When the file has been extracted, the following installation dialog will be displayed.

Inside the folder where the inst\_socket100.exe file was executed, an onsocket directory (HTTP: onchttp) will be created. This directory is not necessary, and can be deleted.



- 4 Click OK.



- 5 Click the icon button in the upper left to start the installation.

## Uninstalling the ONC ActiveX Controls

This explanation uninstalls the Socket ONC ActiveX controls as an example.

- 1 Select Settings | Control Panel | Add/Remove Applications from the start menu. The Add/Remove Applications Properties dialog will be displayed.
- 2 Select Oncsocket.
- 3 Click the Add/Remove button.
- 4 Follow the instructions on the screen to remove the software.