IMPROVED ROBOTIC DEVICE DRIVE SYSTEM – FLEXIBLE MANUFACTURING SYSTEM (FMS)

By

SITI SALWA ABDULLAH

Dissertation 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 2009
by
Siti Salwa Abdullah, 2009

CERTIFICATION OF APPROVAL

Improved Robotic Device Drive System- Flexible Manufacturing System (FMS)

by

Siti Salwa bt Abdullah

Final report submitted to the
Electrical & Electronics Engineering Programme
Universiti Teknologi PETRONAS
in partial fulfilment of the requirement for the
Bachelor of Engineering (Hons)
(Electrical & Electronics Engineering)

Approved:

(Dr. Rosdiazli b. Ibrahim)

UNIVERSITI TEKNOLOGI PETRONAS
TRONOH, PERAK

June 2009

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.

Siti Salwa bt Abdullah

ABSTRACT

With today's industry and new technology development, there are more demands on efficient and flexible system in the manufacturing system in order to have a productive and qualitative production. Plus, robotic and automation is the main element in the industry especially for the manufacturing factories. The innovation of flexible manufacturing system has been developed rapidly in the industry as the market become more intense and competitive. This report presents the improved robotic device drive system-flexible manufacturing system (FMS). Fundamentally, this project is an improvement of the previous project. The existing system is an integrated system of KUKA KR3 Robotic, OMRON C200HE PLC, sensors, and conveyors. The improvement is focused on the two main parts which are PLC programming and human machine interface of the system. The system developed is an application of robotic and PLC in the real manufacturing industry which is a pick-and-place system. It is a small scale FMS where the robotic arm will respond to the PLC command which has been programmed. By using the application of HMI, the user could control and monitor the pick-and-place system from PC. In order to understand and develop this project a thorough study of the robotic, PLC, FMS, and HMI as well as the hardware and software has been done. Then, the development of programming for PLC and HMI had been developed by using CX-programmer and Visual Basic 6.0 software.

ACKNOWLEDGEMENT

I would like to express my gratitude to those who involved in completing my final year project. Firstly, I would like to thanks to the project's supervisor Dr. Rosdiazli Ibrahim for his guidance, idea, and time. I also appreciate the helps and cooperation from the technicians Mr. Azhar Zainal Abidin, Miss Siti Hajjar Mustaffa, and Mr. Khairul Anuar. Lastly, I would like to express my grateful to my family and friends for their supports in this project.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

Flexible Manufacturing System	-	FMS
Programmable Logic Controller	-	PLC
Human Machine Interface	-	HMI

CHAPTER 1
INTRODUCTION

1.1 Background of Study

The concept of FMS evolved during the 1960s when robots, programmable controllers, and computerized numerical controls brought a controlled environment to the factory floor in the form of numerically-controlled and direct-numerically-controlled machines. FMS is defined as a group of numerically-controlled machine tools, interconnected by a central control system. The operational flexibility is enhanced by the ability to execute all manufacturing tasks on numerous product designs in small quantities and with faster delivery. It has been described as an automated job shop and as a miniature automated factory [2].

FMS also can be defined as a "reprogrammable" system that variety of products could be produced automatically. There are a few factors of FMS introduced to the industry; firstly to increase the production of products and continuous without human interaction. By implementing it in the industry, this system will make use of maximum effort to produce at optimum level. Secondly, to reduce cost of manpower so that it is easier to control the operating cost [1].

As industries move closer to implementing agile manufacturing concepts the need for automatic and reprogrammable controllers will increase rapidly. The productivity of FMSs in such industries will be measured in terms of (i) device flexibility - use of reconfigurable and reprogrammable machines for part production and robotic manipulators for part transfer and (ii) system flexibility - use of a supervisory controller to reprogram the operation of the FMS in order to accommodate alternate production routes when needed [3].

In the manufacturing industry, the competence among the manufacturers is high. Therefore, it is important to increase the high productivity of productions in order to have strong position in the market. Basically, in order to increase the productivity, the number of labour also must be increased. But, the cost will be increased as well. Therefore, with the advance of robotic technology nowadays, the use of robot arms has become very popular and important in the industry. The study of the robotic and flexible manufacturing system requires knowledge in the PLC, programming and the manufacturing system in the industry.

1.2 Problem Statement

1.2.1 Problem Identification

The existing system is an integrated system of robotic arm, PLC, conveyors and sensors. The previous student had been developed PLC programming with the Ladder Diagram logic programming. There is no user friendly interface for this system as it is controlled by the robot control panel and requires time and understanding to use it. It is hard for non experienced user to control and monitor the system's operation. Besides, the system programmed for the FMS has no flexibility. The system's operation is about to pick two objects and then placed in the box. When the user need to increase the number of objects picked up by the robot, the program must be reconstruct to achieve that. Therefore, it is very important to construct a flexible program for the system in order to have a FMS.

1.2.2 Significant of the Project

Since manufacturing industry have been developed rapidly as the demand is increasing, it is important to come out with the reliable system. The significant of this project is to design and implement HMI for the monitoring as well as controlling the system in automatic and manual mode. It is also to have an efficient and flexible pick and place system.

1.3 Problem Objective and Scope of Study

1.3.1 Objectives

The main objective of this project is to improve the existing system which leads to two identified goals:

- To create HMI for monitoring and controlling the system automatically and manually. The HMI will give more reliable and feasible system to the user or operator.
- To establish a communication link between PLC with PC-based
- To reconstruct PLC programming

1.3.2 Scope of Study

The scope of study for this project is limited to the development of the small scale FMS and HMI. The main task of the system is pick and place. Once the sensors sense the existing of object on the conveyor, the robot will pick and then place it into the box. The software that will be used is CX- Programmer which is where the programming for the PLC will be developed. For the HMI, Visual Basics software will be used which is integrated with the Compolet series in order to have the communications between HMI (PC) and PLC; FinsGateway will be act as a communication server.

CHAPTER 2
LITERATURE REVIEW

2.1 Robotics and Automation

The term robots are defined by machines that can replace human beings as regards to physical work and decision making. Meanwhile the study of robots is classified as robotics [19]. In Europe, robotics is defined as "the science of robotology," and robotology is defined as "the means by which robot machines are put together and made to work." Many people think of robotics as a single area of technology but in fact robotics encompasses such diverse areas of technology as mechanical, electrical and electronic systems; computer hardware and software [20].

Most of industrial robots are stationary and generally has a single manipulator somewhat similar to a human arm and hand. In most applications, robots do not work as fast as humans, but they are more reliable than human in some applications. They are intended to take over work currently done by humans in areas that are dull, dirty, dangerous, or difficult. A hazardous atmosphere in a workplace requires expensive protective devices for human workers and even these may not completely protect the human from harm. Since the emphasis in robot development is on industrial robots, factories are the place where robots mostly are used. Industrial robot can be used for tasks that need to be done repeatedly, but not often enough to justify the use of automated equipment. It has been found many applications in industry: die casting, forging, machine tool loading and unloading, parts transferring, spray painting, small parts of assembling, finishing, plastic molding, welding, and inspecting. For parts transferring, many manufacturing jobs involve removing parts from pallets and placing them in bins or conveyor belts-or removing parts from bins and conveyor belts and placing them on pallets. This type of work is also known as pick-and-place. Once the robot taught where to locate each part on pallet, it can load or unload the parts properly and consistently from the on. Picking up parts from a conveyor belt is more of a problem for the robot since it must receive some kind of timing signal from the conveyor system to tell it when part is ready to be picked up or it must be equipped with the sensor of its own to see of find the part. An industrial robot has the parts and characteristics: hand, wrist, arm, base, lifting power, repeatability, manual control, automatic control, and memory, library of programs, safety interlocks, reliability, and easy maintenance. The hand of robot is known as a gripper, an end effector, an actuator, or end-of-arm tooling. It consists of the driven mechanical device(s) attached to the end of manipulator, by which objects can be grasped or acted upon. The manipulator is the part of the robot that physically performs the task. Attached to it is the gripper or hand which actually contacts the parts or materials being processed. Without the manipulator the robot not is able to produce motion. The wrist of the robot is used to aim the hand at any part of the work piece. The arm is used to move the hand within reach of a part or work piece. The base of the robot which serves to support the arm is called shoulder [20].

Figure 1: The KUKA robotic arm

For this project, the robotic arm used is KUKA KR 3 type and it is an industrial robot. It has six-axis variants and designed for the light payload applications that required articulated motion in the horizontal and vertical planes. The combination of high-speed flexible motion, reliability and easy usage make this robotic arm powerful. Besides, the position for arm can be maintained for a long time even though the controller is disconnected and re-mastering only required after prolonged shelf-storage of the system. There are two main parts of this robotic arm which are KR C3 Controller Unit (CPU) and KUKA Control Panel. The control panel is used for the teaching and route defined

Robotics has rapidly moved from theory to applications and from the research labs to industries over the last 20 or so years. Robotics is going to be a prominent component in manufacturing industry, which will affect human labour at all levels, from managers of productions to shop floor unskilled workers. Robots are also finding many applications outside of the industry, in research, hospitals, space, supermarkets, service sector, farmhouses, and even in home as pets. The current-day applications of robots can be categorized into two broad areas: industrial applications and non-industrial applications. In today's economy, industry needs to be efficient to cope with the competition. Installing robots in the industry is often a step to be a more competitive because robots can do certain tasks more efficiently than humans. Robots offer an excellent means of utilizing technology to make a given manufacturing operation more profitable and competitive. The main advantage offered for the industrial needs is the improved productivity and quality offered by the robots. The most basic robot applications is one which the robot is required to pick a part of material or other material from one location and place it at another location. Many tasks performed by a robot require this basic pick-and-place operation. [19]

The robot technology is advancing rapidly. The industry is moving from the current state of automation to robotization to increase productivity and to deliver uniform quality. One type of robot commonly used in the industry is a robotic manipulator or simply a manipulator or a robotic arm [19]. But the robot task such as pick-and- place generally require some initial teaching operation. Teaching operations are indispensable for industrial robot arms to be brought into use, and a huge amount of time is consumed in this operation. This comes from the fact that only the positioning repeatability of a robot arm is available for the practical application. In fact most of today's robot arms have worse absolute positioning accuracy than repeatability. For example, the position of the end effector of an articulated robot arm is usually calculated based on the joint angles which are detected by rotary encoders. This calculation decreases the absolute accuracy. One way to overcome such a problem is to utilize sensors which can directly detect the position of the end effector. If sensors can detect the position of the end effector directly, the accuracy of positioning can be increased, and the teaching operation can be done easily. When the sensory feedback is considered in the conventional control algorithm, the sensory information in the task coordinate system is transformed into the joint coordinates of robot arms using inverse kinematics, and the servo loop is closed in the joint coordinate system. But it is difficult to have sufficient accuracy because of the errors caused by the coordinate transformation. In order to take advantage of the sensory feedback, a control algorithm, which has an explicit closed loop in the task coordinate system, is preferable. Some articles have proposed control schemes in the task coordinates [4].

The Internet connects factory automation systems to customers and providers in order to build the so-called "Virtual Factory;" "Web Robotics" are a new fascinating frontier for research and entertainment. Most of these new trends have been made possible by the evolution of the personal computer (PC) (in terms of cost, power, and robustness) and the Internet (in terms of security, speed, and reliability). This evolution has dramatically influenced the way robotics and automation systems are conceived and developed today. In the last few years, big companies like General Motors have initiated radical moves to replace PLCs in manufacturing with PC-based systems running PLC emulation software [5].

Robot arms are used in various fields of industries for many purposes, and the rigidity of robots must be adaptive to a purpose of works. For instance, in case of cutting arid welding, the rigidity of robots must be kept high, because high accuracy of the locus of a tip of the robot arm is required. On the other hand, in case of pull-out-work, assembly or grinding, the rigidity of robots must be kept low, because the flexible motion by external force is required. As almost all industrial articulated robot arms however are designed with the high rigidity for convenience of positioning and contouring, those robot arms are not so suitable for pullout-work [6].

2.2 Flexible Manufacturing System

Flexible manufacturing (FM) is a term used to describe a manufacturing activity and facility that is computer controlled so that it can automatically process a number of different products in any desired quantity and priority. The flexibility of a FM facility comes from the fact that it is [7]:

1) an automated system in which the majority of the resources employed are programmable;
2) fully tooled and programmed to process a variety of dissimilar products-as opposed to a dedicated automation line which normally produces only one or a restricted family of like products;
3) Controlled by a supervising computer which interprets the overall production schedule and creates a system schedule of product type, priority and quantity. It will then control the processes, materials handling equipment and tooling to produce to the system schedule
4) Capable of producing product in any priority or quantity.

FM facilities take three major forms: cells (FMC), systems (FMS), and flexible transfer lines. The main distinctions between these forms are the number of part varieties handled, the degree of integration with other external facilities, overall size, the extent of the transport systems needed and capabilities of the computer control system and software. FMS is a system is designed to fully schedule and controls the automatic processing of a variety of products through a number of cells and other production facilities, the total operation being controlled by a central computer. A FMS may have a large amount of equipment providing extensive processing, materials handling, inspection and tooling management capabilities. The central (host) computer is networked to the facilities to provide either direct control or to communicate with the autonomous controls of the equipment [7].

In industrial world, automation is very important in efficiently conducting certain tasks for instance material transfer, machine loading and unloading, welding, primary metal work, spray painting, processing and fabrications operations, assembly, products packaging inspections, and even medicine [8] .While industrial automation has long been associated with the increase of manufacturing productivity, much of the current motivation behind the development of advanced automation techniques is driven by the need to enhance product quality, increase the safety of the operational environment, and quickly respond to changing consumer demands for new products [9].

The industrial community has become increasingly aware of the critical importance of efficient manufacturing techniques in maintaining a competitive stance in world markets. In response to this competition various new manufacturing strategies have been developed, one being FMS applied in the batch manufacturing environment. The benefits in comparison to conventional approaches can be substantial and include minimum batch sizes, improved product quality, reduced manufacturing lead time, reduced labour requirements, and improved equipment utilization [10]. The variety and complexity of processes required to be controlled in modem manufacturing places heavy demands on PLCs [11].

2.3 Programming Logic Controller

PLC is a specialized computer used to control machines and process. It uses a programmable memory to store instructions and execute specific functions that include on/off control, timing, counting, sequencing, arithmetic and data handling. Basically, the PLC is an assembly of solid-state digital logic elements designed to make logical design and provide outputs [8,12,13,21]. The operation of a controller is most simply understood by envisioning that it repeatedly performs three steps [12]:

i. Reads inputs from input modules
ii. Solves preprogrammed control logic
iii. Generates outputs to output modules based on the control logic solution

A typical PLC can be divided into four main parts which are central processing unit (CPU), Input / Output (I/O) section, power supply and programming device. The term architecture of PLC can be referred to PLC hardware to PLC software or to a combination of both. An open architecture design allows the system to be connected easily to devices and programs made by other manufacturers (Figure 2). Generally, there are five classes of PLCs in the industry today which are; nano, micro, small, medium, and large. The criteria used in categorizing PLCs include functionality, number of inputs and outputs, cost, and physical size. PLCs have the great advantage that the same basic controller can be used with a wide range of control systems over conventional relay type of control [21]:

a) It has eliminated much of the hardwiring associated with conventional relay control circuit. Relays have to be hardwired to perform a specific function. When the system requirements change, relay wiring has to be changed or modified. In extreme cases, such as in the auto industry, complete control panels had to be replaced since it was not economically feasible to rewire the old panels with each model changeover.

b) It is small and inexpensive compared to the equivalent relay-based process control systems. Lower cost/cost effective

c) Increased reliability

d) More flexibility

e) Communications capability – a PLC can communicate with other controllers or computer equipment to perform such functions as supervisory control, data gathering, monitoring devices and process parameters, and download and upload programs.

f) Faster response time

g) Easier to troubleshoot

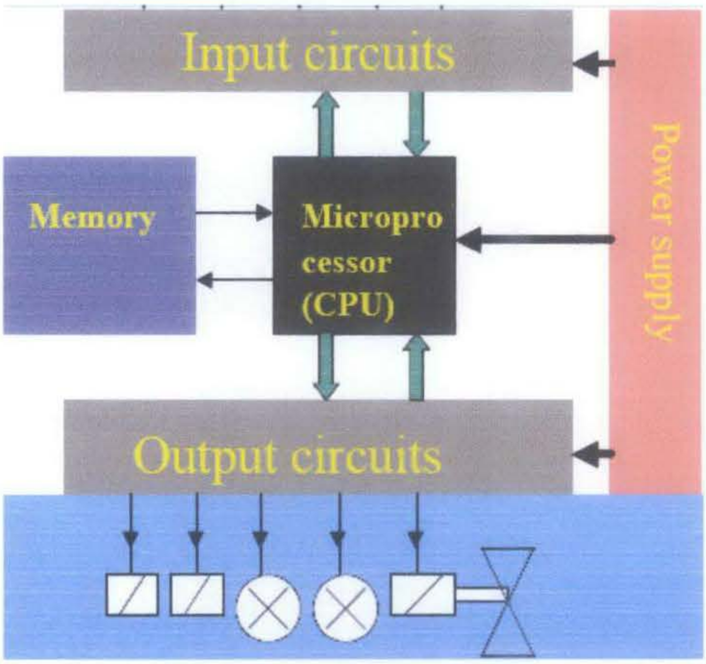


Figure 2: The PLC architecture

The variety and complexity of processes required to be controlled in modern manufacturing, places heavy demands on PLCs. The emergence of the programmable controllers programming language standard IEC 1131-3 facilitates structured software development. IEC 1131-3 provides PLC users with a programming language based around familiar PLC concepts, but with constructs similar to those used in the current generation of PC programming languages. Improvements and standardization in PLC programming brings greater opportunities for structured software development [11].

The main advantages of IEC 1131-3 are that the software can be designed for reuse as program organization units (POUs) and encourages hierarchical design decomposition. The IEC 1131-3 standard languages consist of sequential function chart (SFC), instruction list (IL), structured text (ST), ladder diagram (LD) and function block diagram (FBD). The PLC translates these languages into executable codes and executes them on the CPU module to control a plant [17].

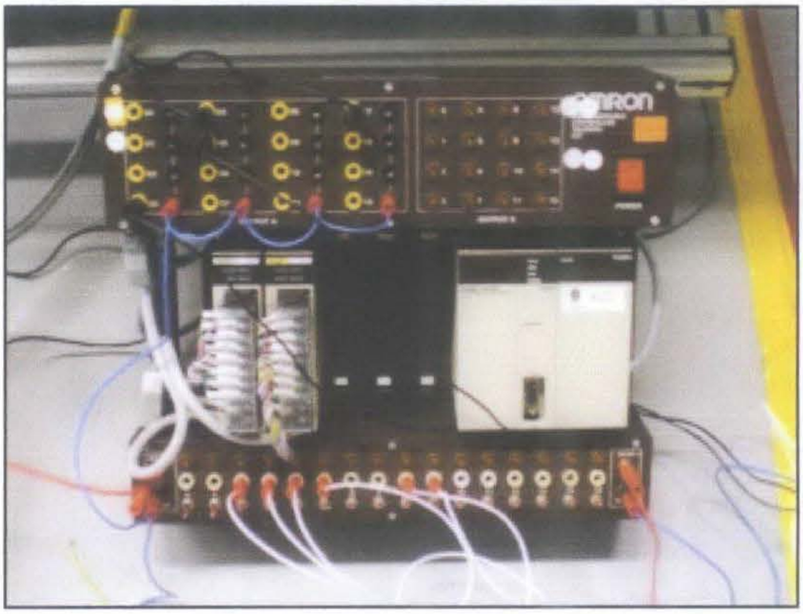


Figure 3: OMRON C200HE PLC

OMRON C200HE as shown in Figure 3 is a series of OMRON PLC that can control a variety of automation applications in the industry. C200HE is high reliable, easy to communicate, effective, and low cost compared to the others. It comes with an expansion slot unit which places the input/output (I/O) cards. The input card, C200H-ID212 comes with 16 inputs per card also same with the output card, C200H-OC225. The I/O cards can be added based on the requirement. For the communication between PLC and PC, RS-232 cable is used.

2.4 Human Machine Interface

There is a large body of human-machine interface design literature, which has at its heart three main goals – effectiveness (helping operators achieve their intentions) efficiency (reducing time taken and the incidence of operator errors) and satisfaction (providing human factors research for safety has focused specifically on reducing the incidence of operator error). However, no method of user interface design can completely mitigate the fact that operators make mistakes. They make, amongst others: slips, lapses, substitutions, sequence errors, post-completion errors, rule misapplications, and even errors of intention. Automated systems also may have errors. They contain, amongst others: requirements errors, specification errors, design errors, and simple typographic bugs. When operator mistakes and computer errors manifest, often the last line of defense is the operators themselves. Safe system design involves not just reducing the likelihood of hazardous failures, but mitigating their consequences as well. This requires HMI design that facilitates the operator acting to mitigate errors, whatever their source [14].

Information display is a core element of human–machine interfaces in complex process control systems. Many studies have reported that traditional information displays, which focus on providing physical information representing state variables of each component and subsystems of the work domains, burdened human operators with great cognitive loads, thereby degrading task performance. The insufficient display design has been one of the primary causes of severe safety problems in complex systems like nuclear power plants. For this reason, there has been a great deal of research works for designing more effective information display, using advanced information technologies. Introduction of new information technologies like computer graphics has increased designers' degree of freedom and thus made it possible to present diverse kinds of processed information in various graphical formats. However, advanced display technologies do not ensure a display design that reduces cognitive loads efficiently and thus allows human operators to effectively deal with complexity and unanticipated events [15].

Traditional production assembly or workstation design related to human factors starts by diagnosing the system and identifying deficiencies in the existing human-system interactions, followed by implementation of a solution through design, training, and selection. Design, including equipment design, task design, and environment design, is a critical step that requires a great amount of effort since the redesign procedure usually leads to loss of significant amount of money and time. With advances in computer technology and development of digital human modeling methodology, it is possible to predict risks of potential injuries during manufacturing design prior to production, and make proactive ergonomic design for manufacturing assembly workstations or work-cells involving human-machine interfaces [16].

The purpose of an HMI is to provide the operator with at least as much functionality as the control board. In fact, until recent years, the HMI package was limited to displaying buttons, lights, numeric displays, and other simple graphic elements that had a direct relationship to the electronics on the control board. Those elements are still used today. But, it is also possible to provide a picture of the process just as it appears to the operator and to animate that picture to reflect process conditions. Today's operator interface is likely to look like a process flow diagram with three dimensional graphics, fully animated and interactive with the operator. But, graphics are just pictures until they are linked to the outside world. To configure a GUI is to establish links between the graphic elements on the screen and calculations, analog or discrete values pulled from the PLC, or other graphic elements [18].

CHAPTER 3
METHODOLOGY

3.1 Research Methodology

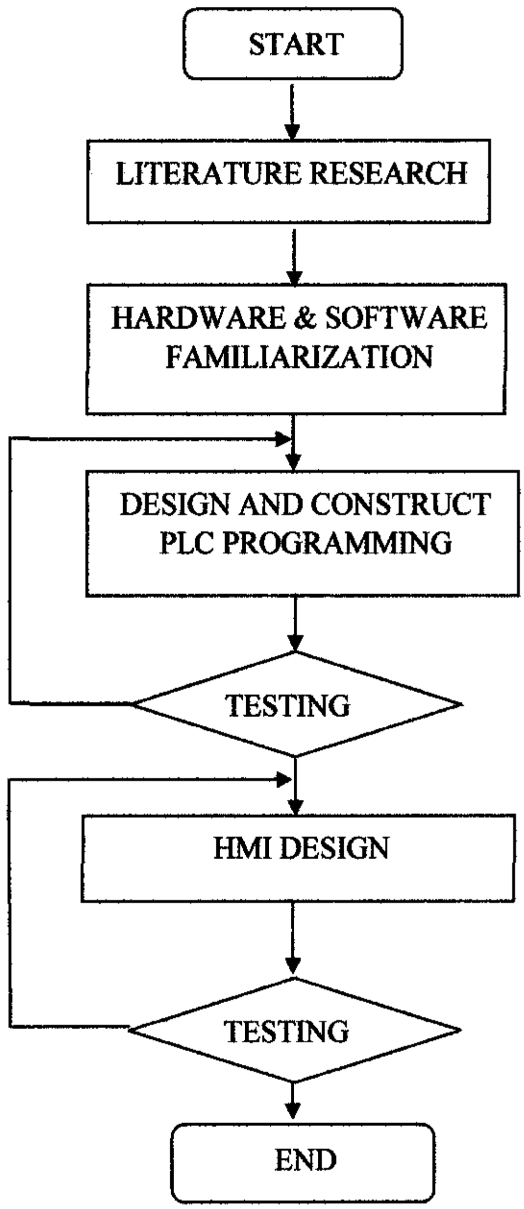


Figure 4: Flow chart of the project

Literature Research

A study has been done on robotic and automation, PLC, FMS, and HMI generally. This is important in order to have a proper view, idea and understanding of the project. The four main topics are related to each other. As in the manufacturing industry, PLC and robotic arm are widely used. Besides understanding the topics generally, a focus must be done on the project hardware used in the project as well as the software. A good knowledge on the hardware and software are really important.

Hardware and Software Familiarization

Since this is an enhancement project from the previous project, it is important to understand the existing system first. There are two main parts that need to be concentrated on which are hardware and software part. The hardware's used in this project are OMRON C200HE PLC and KUKA robotic arm. Thorough understandings of this hardware are necessary. First of all, the KUKA robotic arm operation must be familiarized by learning how to handle it by using the KUKA Control Panel (KCP). Plus, the KCP familiarization must be done in order to control and program the robot. The robot movement is programmed based on the PLC programming. For the C200HE PLC, wiring for the I/O must be troubleshooting in order to verify the correct wiring since the sensors had been removed. Then the system testing of the previous project is done in order to observe the sequence of the system as well as the functionality.

Besides, the software use has been focused for this phase. The PLC is programmed by CX-programmer and for the HMI development, Visual Basic 6.0 is used. CX-programmer is a PLC programming tool for the creation, testing, and maintenance of programs associated with the OMRON PLC series. It provides facilities for the support of PLC device, address information and for communications with PLC as well as PLC's supported network types. There is also other software used in this project i.e. SYSMAC Compolet Series which is for ActiveX control and PC-PLC communication.

Design and Construct PLC Programming and HMI

The programming for the system is divided into two parts which are PLC and HMI programming. For the system, there are manual and automatic programming mode needs to be developed. The system is controlled by using OMRON C200HE PLC. The PLC programming is programmed by using CX-programmer software and the language used for the programming is Ladder Diagram (LD).

From the LD programming, the HMI programming can be developed. Basically, the faceplates of the HMI can be designed parallel with the LD programming. According to the address assigned in the LD, the HMI programming developed by using Compolet coding. For this HMI development, the communication between PC (HMI) and PLC must be established first. The communication server, FinsGateway has to be configured to allow the communication. Comoplet series component application is to be the PLC ActiveX communication for programming support. With this component, a program could be created for communications between PC and the PLC through the communication server. After that, the

Testing

The objective for the testing is to ensure the system is working as per required. The testing was done to the FMS in the laboratory in which the full integration system between the robotic arm and PLC. For the existing FMS testing, the sensors have been installed first since the sensors of the system have been removed before. Then the sensor testing is done to check the functionality and the address of the sensors based on the PLC program. After that, the testing has been conducted for the system as to understand the sequence of the existing system and to troubleshoot the I/O wiring of the PLC. While for the improved programming with HMI, the testing must be conducted to check the functionality of them. Since it is the improved one, it must be tested whether it works and the sequence is correct. First of all, the new programming is downloaded to the PLC and tested with the whole system. The HMI check will be done then. The system will be activated from the HMI and the system will be running.

3.2 Tools and Software Required

Tools required for the project include hardware and software. It is important to identify and use the correct equipments for the project. The detail for the project's tool is simplified in the Table 1 below.

Table 1: Equipments details

No.	Equipments Hardware	Details	Unit
1	KUKA Robotic Arm	• KR C3 CPU • KUKA Control Panel (KCP) • Power supply	1
2	OMRON C200HE PLC	• CPU 42 • Power supply PA204 • Input card C200H-ID212 • Output card C200H-OC225	1
3	Personal Computer	• Windows XP/2000/98	1
4	Cable	• RS-232 • I/O terminal	1
	Software		
5	CX-programmer	• For PLC programming	-
6	SYSMAC Compolet Full Version	• For PC-PLC communication	-
7	Visual Basic 6.0	• For HMI development	-

CHAPTER 4
RESULT AND DISCUSSION

4.1 PLC Programming

4.1.1 Findings

For the PLC programming development, a lot of modifications need to be done in terms of addressing and also the ladder diagram. Since the current programming for the existing system was assigned with the physical addresses, then the addresses must be reassigned to the internal address (virtual address) but only applied to the inputs which are used in the HMI. The physical address means that, the address can be triggered with the physical input such as push button, switch, or sensor. Therefore, in order to trigger the programming internally which is from PC through the HMI, internal address use is important. Basically, internal address refers to internal relay (IR) and Special Relay (SR) address. First of all, the addresses that need to be reassigned must be identified. Then, the changed address is tested and working successfully with the HMI coding.

For the programming, other than reassigned the address, TIM instruction is used. Its function is to prevent the conveyor moves before object is placed on it. This is because as the sensor detects the presence of the object, the conveyor will move immediately. Therefore the conveyor would move before the object is placed properly and affect its position on the conveyor. Besides, the STOP instruction had been added in the program as it is not exist in the previous project's program. By using this STOP instruction, the user could stop the system from the HMI instead of using the KUKA control panel. The manual part had been developed by adding by COUNTER instruction where the desired count for the numbers of objects could be chosen by the operator or user. The complete PLC programming is attached in Appendix C.

4.1.2 Discussion

The modifications of the current programming have been done by reassigning certain physical address to the internal address. For the addressing, it should be identified which address could be used. This is because some of the addresses are not possible for the programming use. Table 2 below shows the address area which is could be used for reassigning address of the program. The details of the address are available in the Appendix B. Then, STOP instruction had been added to the programming. This STOP instruction function is to stop the system from the HMI by the user. For timer instruction, its prefix is TIM, decrementing ON-delay timer instruction which required number of set value (SV) and timer number. As shown in Figure 5, the timer number is 003 and the set value is 30 seconds. It means that, the timer would decrement from 30 seconds and then activate the next operands. For the manual part, the user could choose the numbers of objects to be picked up by the robot. The use of counter is being applied for this part.

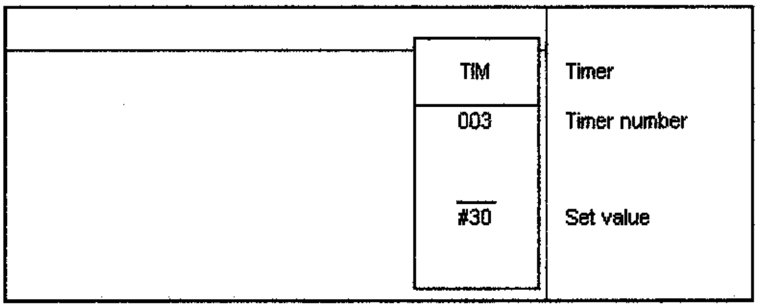


Figure 5: TIM symbol

Table 2 : IR and SR address range

AREA	SIZE	RANGE
IR Area 1	3776 bits	IR 000 to IR 235
SR Area 1	312 bits	SR 236 to SR 255
SR Area 2	704 bits	SR 256 to SR 299
IR Area 2	3392 bits	IR 300 to IR 511

4.2 PC-PLC Communication Establishment

4.2.1 Findings

For the HMI development, communication between PC and PLC has been established. This is to ensure that PC can read/write data from/to PLC. SYSMAC Compolet software provides Active X components that enable the development of user applications to operate SYSMAC C-series PLCs as for this project C200HE PLC is used, connected to the computer. SYSMAC Compolet provides ready-made functionality for accessing and utilizing most of the PLC operations from the computer. It makes it easier to customize HMI for the control of PLC. FINSGateway is required to enable the communication for PC-PLC since it acts as communication server. For the FINSGateway setting parameter, the procedure is shown as below, first by activating FINSGateway Service Manager.

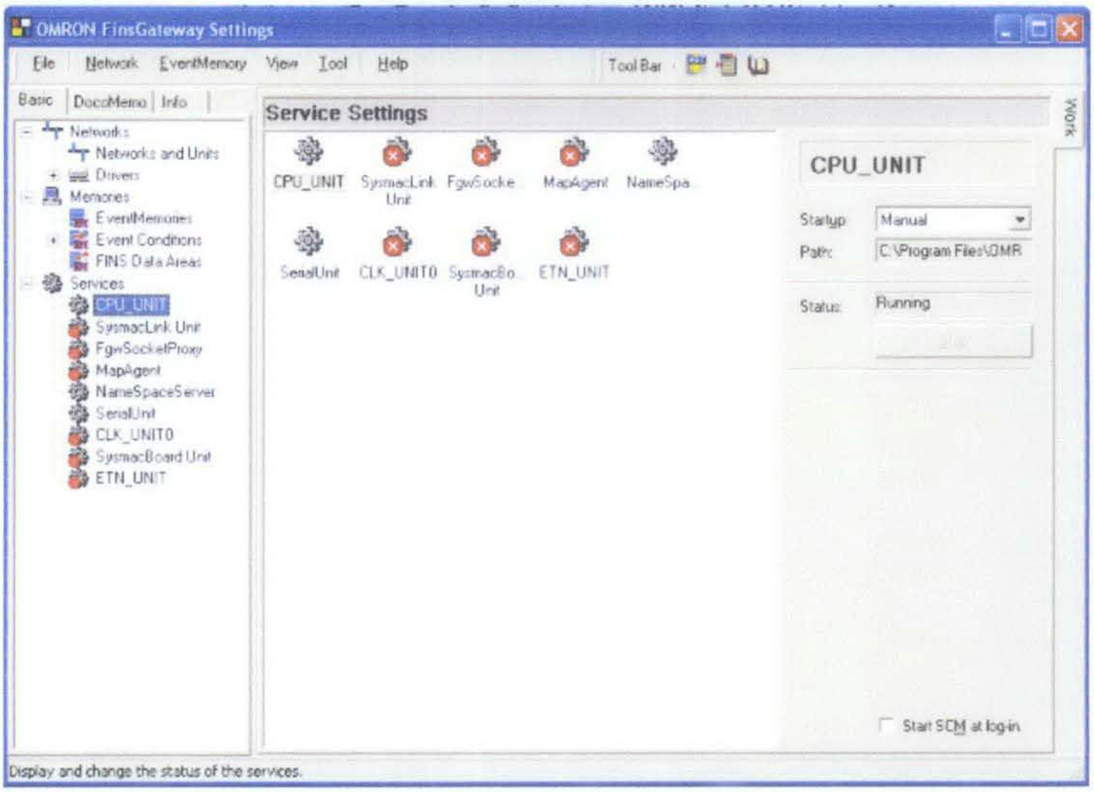


Figure 6: Service Control Manager

Figure 6 shows the service control manager for the FINSGateway settings. Under the service, CPU_UNIT and SerialUnit were started by clicking Start button at the right. Upon activation, the Running status would appear. Both services must be in the running status simultaneously.

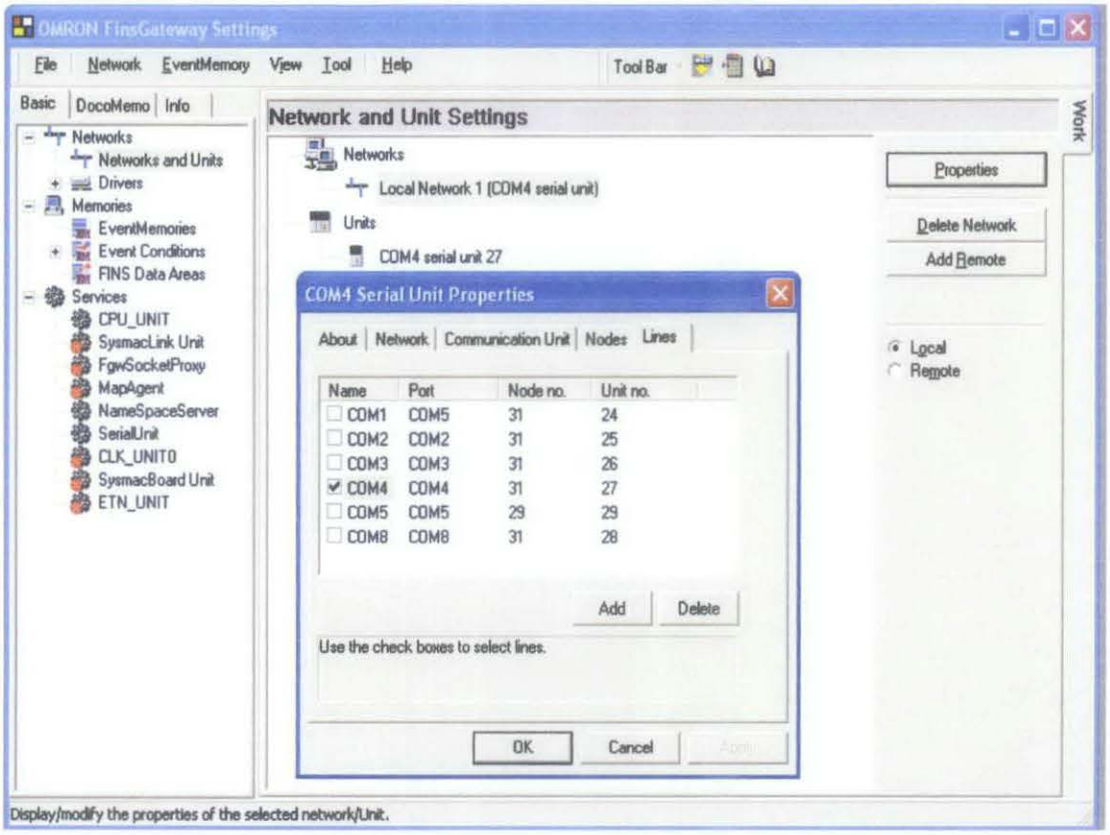


Figure 7: Network and Unit Settings

Normally, setting parameters appeared automatically once the PC-PLC communication has been established. But sometimes the user has to check and set the setting manually. To check the Network and Unit setting, COM port properties is viewed by clicking the Properties button at the right and go to Lines tab as displayed in Figure 7. The COM port used must be same as the PLC setting in the CX-programmer setting. If the PLC setting used COM1, then in the Network and Unit setting the COM1 must be ticked. In the Nodes tab, node setting parameter shows the property of the node or device connected to the network as shown in Figure 8. The device selected is C200H which is representing the PLC used i.e. C200HE PLC. The Unit number of the node is recognized as '0' and Node number is '240'. The protocol is SYSWAY which is the type of protocol used when the communication is of the serial type. The Network number is recognized as '1'

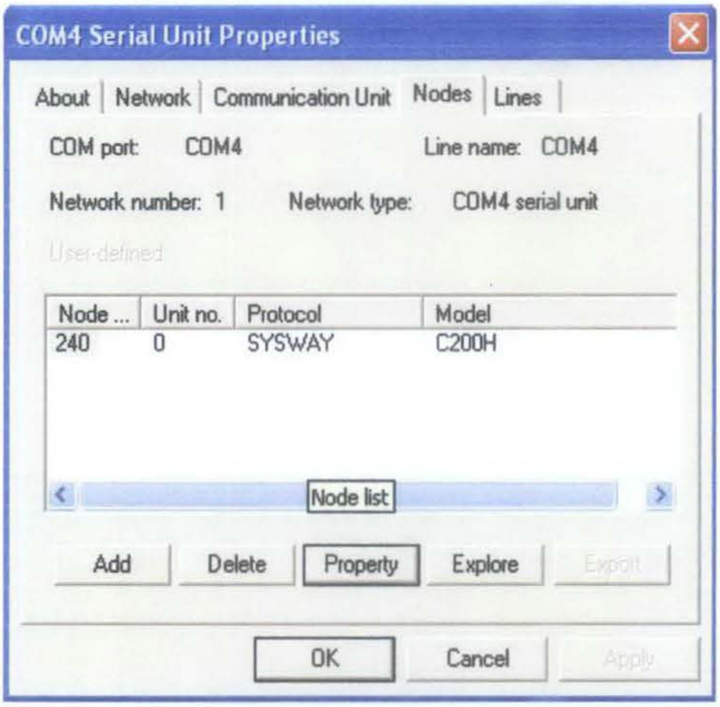


Figure 8: The Serial Unit properties

Then, to verify the communication was established successfully, FINSGateway Network Tester was executed. This network tester was executed by clicking the Network tabs at the Toolbar and display as Figure 9 appeared. At the peer address, the address is according to the **Network.Node.Unit** number. Therefore, the address should be **1.240.0** based on the setting in the Serial Unit properties in Figure 8.

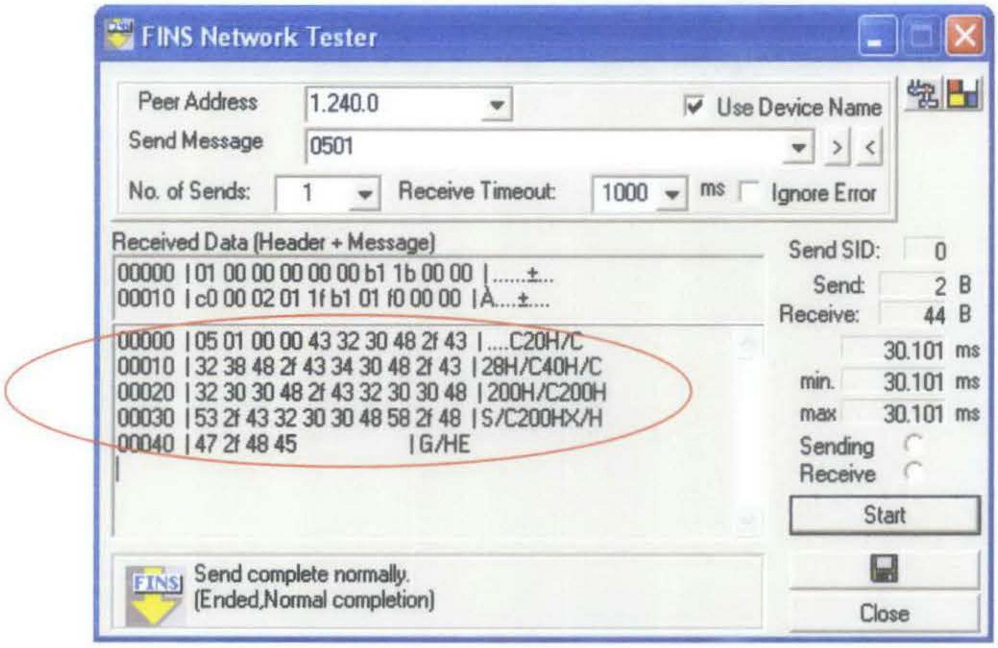


Figure 9: FINSGateway Network Tester

Once the peer address is keyed in and Start button is clicked, the message will come out as in the red circle of Figure 9. It means that communication is successful which is the information displayed shows the connected and used PLC i.e. C200H type.

4.2.2 Discussion

For the communication PC-PLC establishment, the objective is to create communication between PC and PLC so that PLC can be controlled and monitored from PC. The PC-PLC communication has been established by activating FINSGateway Servie Manager. The important setting is to identify the Network, Node, and Unit component and its addresses. The component of Network for PC and PLC are recognized by the FINSGateway as CPU_UNIT and C200HE respectively. For this PC-based system, the Network address has been set to 1. Node is referring to the device connected to the PC and only one PLC is being used i.e. C200HE PLC. The Node address is fixed at '240' by FINSGateway configuration. It is the address allocated for PLC connected to the PC in the serial communication manner. Unit is referring to the operating CPU unit in the network and recognized as CPU_UNIT. The Unit address is fixed by FINSGateway at '0'.

4.3 HMI Development

4.3.1 Findings

As the communication between PC and PLC had been established successfully, it is more convenient to develop the HMI. For HMI design, VB 6.0 is used as it is compatible with the OMRON C200HE PLC and the SYSMAC Compolet series software. In the VB 6.0, in order to allow the function of SYSMAC C control, SYSMAC C control must be ticked from 'Project Component' on the toolbar. Then the SYSMAC C icon will appear at the left pane and by double click the icon will appear in the form. For the setting, right click the icon and choose Properties, Property Pages displayed as in Figure 10. The Network, Node, and Unit addresses are inserted with 1,240, and 0 respectively.

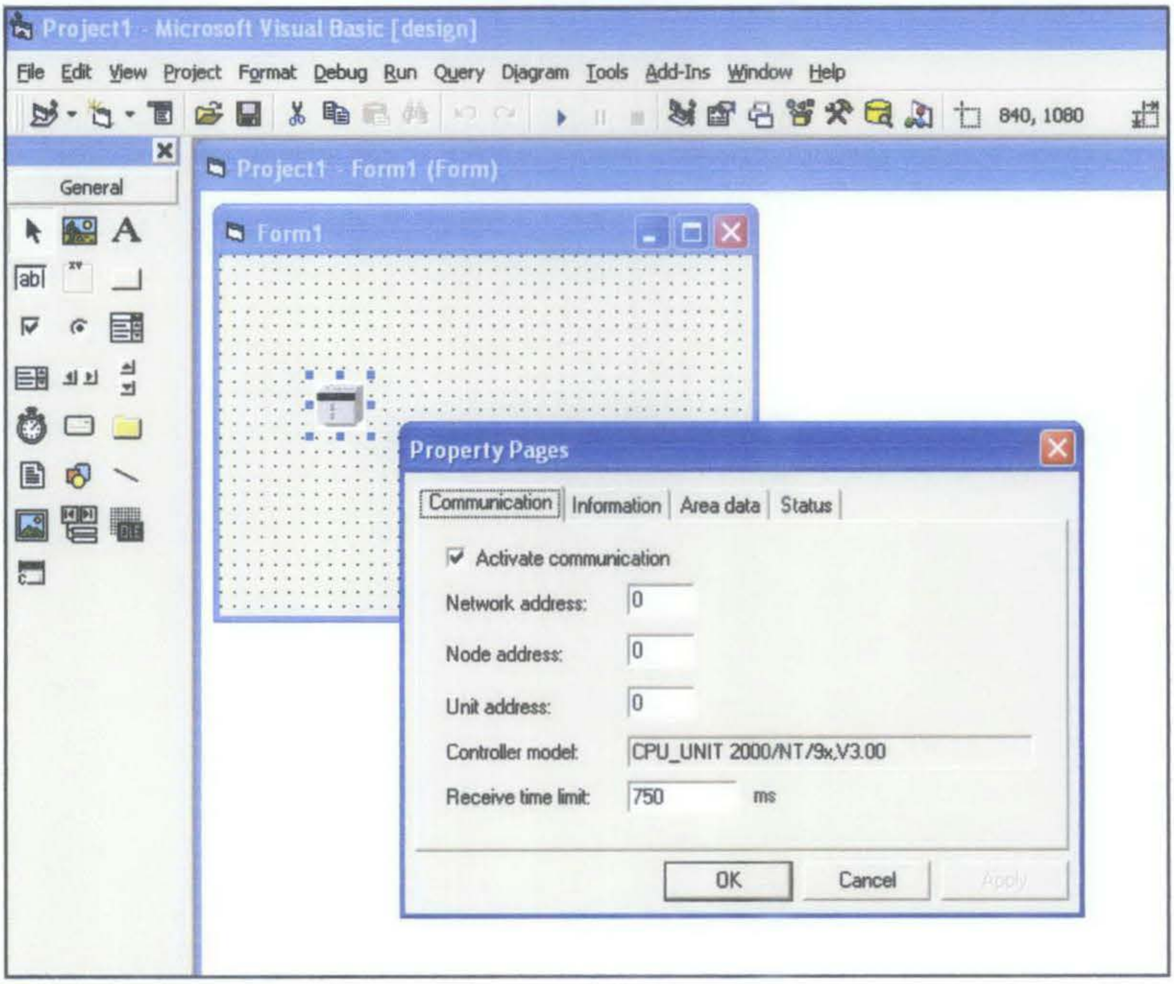


Figure 10: SYSMAC C icon and its Property Pages

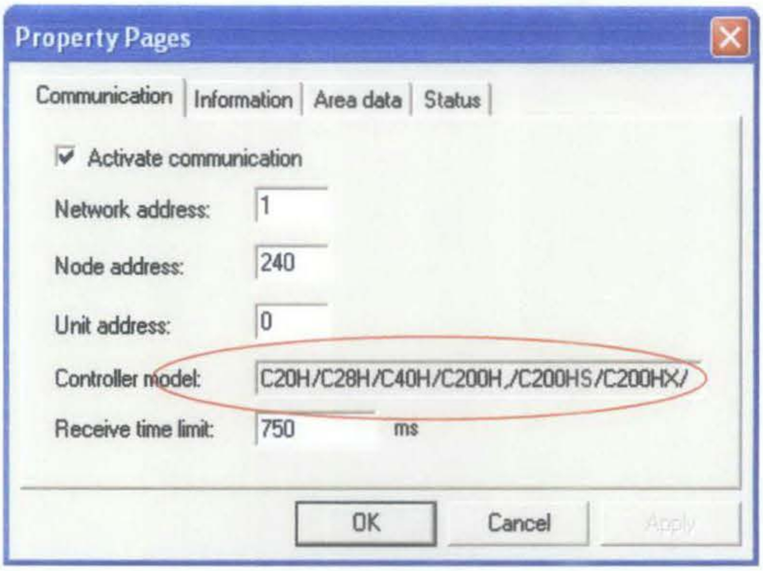


Figure 11: The Property Pages of SYSMAC C icon

If VB 6.0 detects the PC-PLC communications, the Controller Model will show the PLC model that is currently being used as per Figure 11. The controller model C200H represents the C200HE controller. This is showing that the communication is established between the PC and PLC.

Figure 12: Form1-Main page

Figure 12 shows the Form 1 which is the Main page of the HMI design. On the page, there are Activate, Stop, Exit and Run buttons. From this page, the user can activate as well as to stop the system. The Run button is used to run the program of the system by diverting to the Form 2 showed in Figure 13. Figure 13 shows the program option where the user could choose automatic or manual mode for the system.

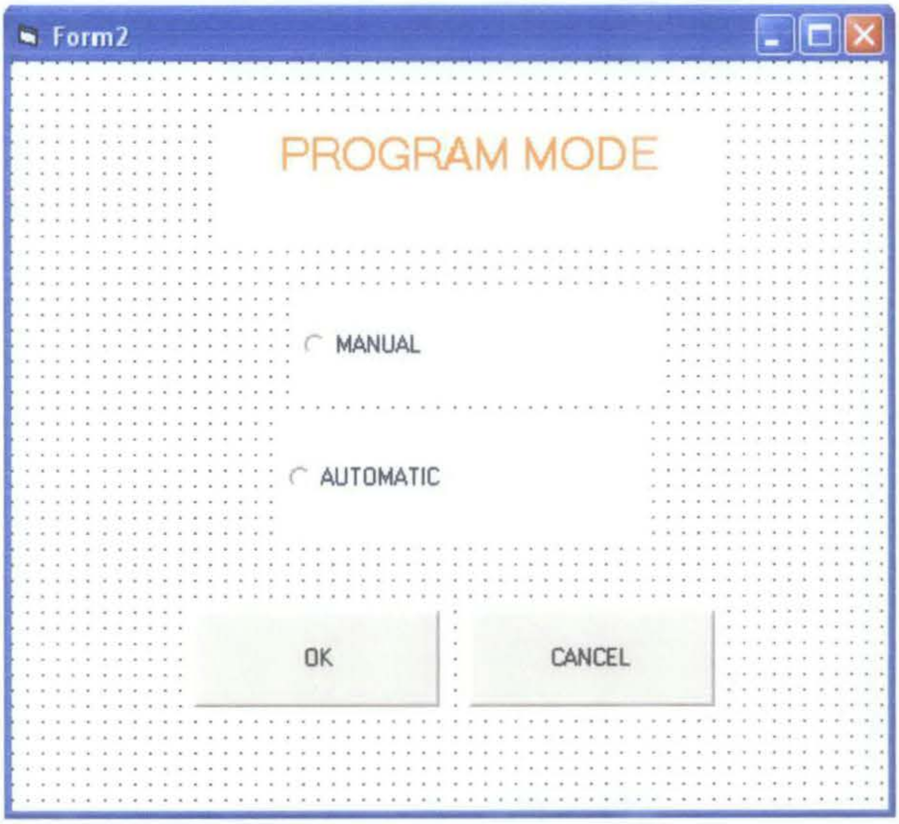


Figure 13: Form 2- optional program mode page

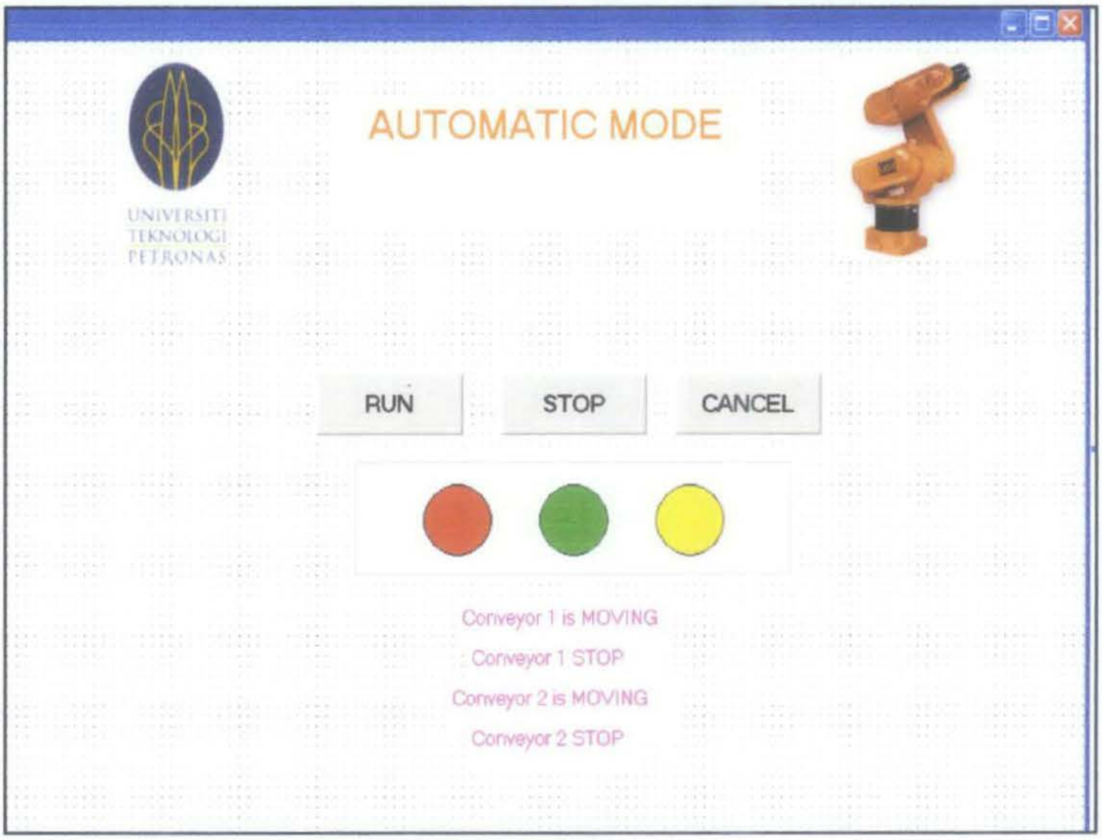


Figure 14: Form 3- automatic mode page

If the user choose automatic mode, form in Figure 14 would pop out. On the Automatic Mode page, once the user run the program the circle button will display the current status of the system either running or stop by showing the green colour for running status and red colour for the stop. The yellow colour of circle indicates the idle state of the system before run the system. The manual mode form in Figure 15 emphasises on the numbers of object can be picking up by the robot. Thus for this form, the user can choose the numbers of the object based on the user's need. The complete VB programming for HMI is shown in Appendix D.

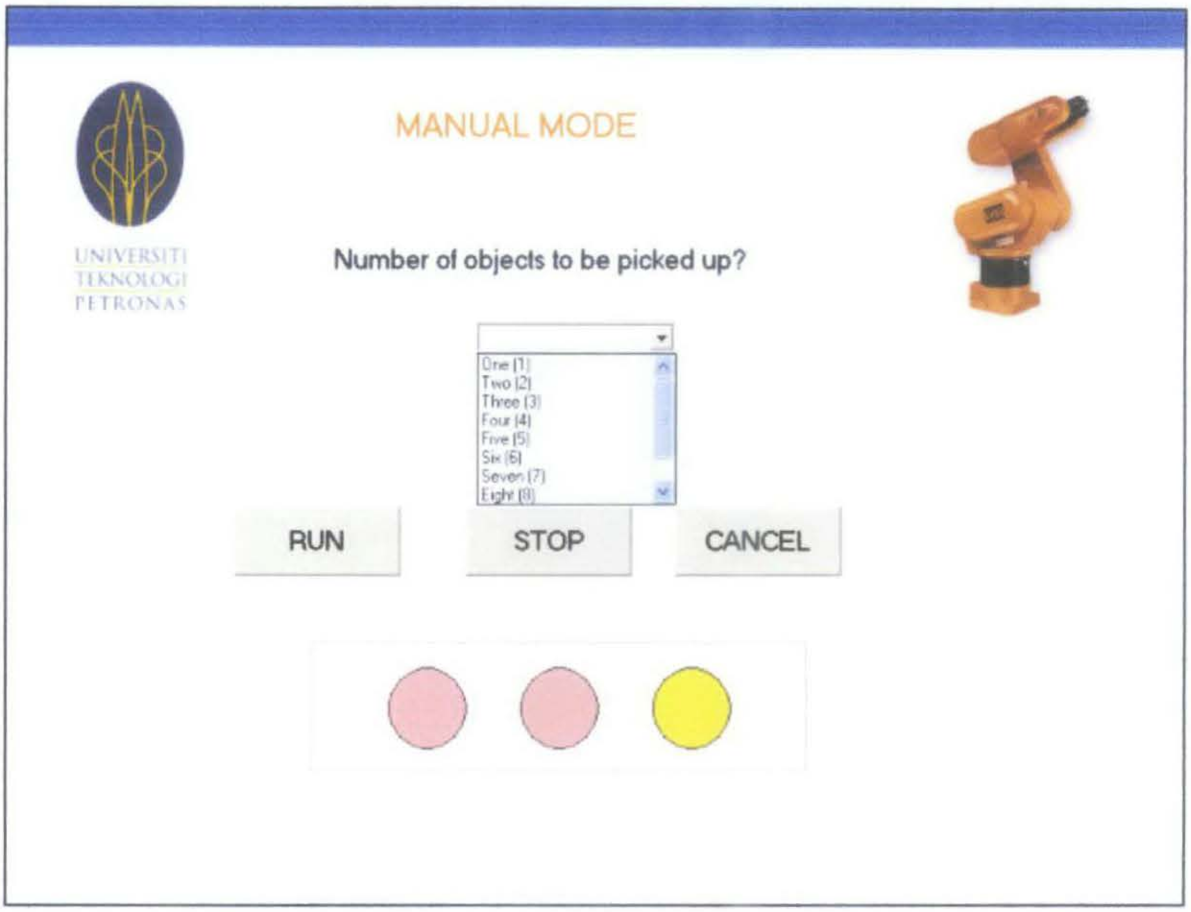


Figure 15: Form 4 - Manual mode form

4.3.2 Discussion

The HMI development, the address of the PLC program is important in order to be used in the VB coding. The addresses allow the PLC program to be invoked internally which is through communication link. Then during the development, there is some problem occurred and discovered such as failure communication establishment. This is because the CX-programmer software and VB 6.0 are being used simultaneously. Therefore, the ladder diagram programming must be transferred to the PLC first and let the PLC in the Monitor mode. Then exit the CX-programmer software. After that, the FINSGateway Service Manager is activated followed by the activation of HMI by using VB 6.0.

The important part for the HMI development is developing the VB coding or programming. As the faceplates design for the system are done, the next step is to program it as per required. Therefore, for the triggering the system locally via HMI, the addresses of the PLC programming is assigned accordingly. But before that, the address must be converted to Compolet Series Component coding address as shown in the Table 3.

Table 3 : The conversion table for the PLC address

PLC add.	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	**00**
Conversion	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	$\mathbf{2^0}$
Compolet	**0**				**0**				**0**				**1**			

As an example, if the PLC programming IR address is 246.**00**. The bit **00** is only considered to be converted.

PLC address: **246.00**

Compolet address: **246.0001**

The Compolet coding is written like this; SYSMAC_C1. writeArea plcArea CIO, **246, 1, 0001** where the **'246'** represents the IR address 246, the **'1'** means the value 1 is sending to the address i.e. the force bit is 1 in order to turn ON the specified command. For this address, the aim is to turn ACTIVATE/ON the system. The rest of the reassigned addresses are converted as the same way.

4.4 System Testing

4.4.1 Findings

Before the testing conducted, there are four sensors installed to the conveyors. The existing program has been tested by using KUKA control panel. During the testing, the PLC program has been monitored through the CX-programmer software. From that, the student could understand the flow of the program. There are two inputs from the PLC act as switches. These two switches are needed to activate and run the system. After that, the testing for the PLC program and HMI had been done with the FMS. From the testing, the robot movement is according to the PLC program. For the HMI testing, the automatic mode is working successfully while there is a problem for the manual mode. This is because the automatic external mode of the robot is not working. Because of this problem, the robot controller could not receive the commands from the HMI and did not response to it since the internal/virtual addresses are used.

4.4.2 Discussion

At the initial testing, there are minor problems to be solved. Since the sensors have been removed before and also the input wiring to the PLC has been changed, the addresses of the sensors must be verified first and then assigned the input wiring to the correct input address on the PLC. Besides, the positions of the sensors are not synchronizing with the robot arm movement. A repetitive testing has been done to ensure the correct position of the sensors. Then, for the PLC program and HMI design testing, a few problems occurred such as automatic external mode of the robot is not working. This is due to the automatic external mode configuration is not configured for the robot. The configuration is quite complex and could be done by the expert only. In addition it will take time to configure it. Thus for this project the robot's automatic mode is used. The robot also has its own controller and I/O circuit. For the robot's automatic mode, it should have the hard wiring from the PLC to the robot's I/O and the physical inputs such as sensor. Therefore, it is discovered that the robot's movement is based on the physical contact. In this system, the physical contacts are the sensors where upon its activation, the robot will move. As a result, the system could not be fully controlled from the HMI. Since the manual mode for the HMI is programmed using the internal/virtual addresses, the physical contact do not exist.

CHAPTER 5
CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

As a conclusion, this project is feasible with respect to the time, hardware and software capability. However due to some circumstances, there are a few limitations came up during the process. For example the robot's configuration for the automatic external mode is not available and must be done by the expert. Because of this, some features could not be used in this project. Therefore, the available option that could be used is automatic mode.

For the first phase of the project, the student had done a literature research of the robotic and automation, FMS, PLC and HMI. Then, a thorough understanding and learning process of hardware and software had been carried out. At this stage, the OMRON C200HE PLC and the KUKA robot had been familiarized and learnt together with the software. For the second phase, there are programming designs for PLC and HMI. The communication between PC and PLC is successfully established by using Compolet Series Component. The FinsGateway acts as a communication server for PC-PLC.

Lastly, the testing for HMI and PLC program showed that the PLC as well as the robot could be controlled and monitored from the PC via HMI. However, due to the robot's automatic external mode is not configured; the system could not be fully controlled from the HMI. There is only HMI automatic mode has been working with the system successfully. The manual mode of the HMI is not working properly. A consistent study about the configuration must be done and understand how to manipulate the automatic external mode in order to control the robot from the HMI.

5.2 Recommendations

For this project, as a small scale FMS has been developed with the HMI. HMI is one of the monitoring and controlling systems that have been practiced in the industry for the operator use. Even though there are improvement has been made to the existing system of the previous project, but more research and study could be done to improve the system.

For the HMI part, the student could add more features to control and monitor the system's operation. More graphics could be added in the forms such as the conveyor graphic. It would animate when the conveyor is moving and vice versa. Therefore, the user could monitor the operation from the HMI only instead of supervise the real system.

Then, the system sequence part can be improved. Essentially, a lot of things can be done to the robot sequence instead of just pick and then place the objects on the conveyor. But the student must do a thorough study on the robot programming and as well as its configuration. The robot and the whole system capability must be taken into consideration.

Since internet and technology is growing rapidly nowadays, there are many applications of internet had been applied to the industry as well as in the manufacturing industry. Internet based system can be implemented to this project. By this implementation, it will be more reliable and flexible system as the operator or user can control the system remotely anywhere via internet connection or LAN.

REFERENCES

[1] Ezri Ezani bin Abu, 2008, *A Robotic Device Drive System- Flexible Manufacturing System (FMS),* B. ENG. (HONS) Thesis, Universiti Teknologi PETRONAS

[2] R. Anthony Inman, August 11 2008, *Flexible Manufacturing System* <http://www.referenceforbusiness.com/management/Ex-Gov/Flexible-Manufacturing.html>

[3] Tarek M Sobh, Beno Benhabib, *Applications of Discrete Event and Hybrid Systems in Robotics and Automation of the IEEE Robotics and Automation Magazine.*

[4] Katsuhisa Furuta, Kazuhiro Kosuge, Nobuhiko Mukai, *Control of Articulated Robot Arm with Sensory Feedback: Laser Beam Tracking System,* IEEE Transactions On Industrial Electronics, Vol. 35, No. 1, February 1988.

[5] Davide Brugali, Mohamed E. Fayad, *Distributed Computing in Robotics and Automation,* IEEE Transactions On Robotics And Automation, Vol. 18, No. 4, August 2002.

[6] Daisuke Kushida, Masatoshi Nakamura, Satoru Goto, Nobuhiro Kyura, *Flexible Motion Realized by Forcefree Control: Pull-Out-Work by Articulated Robot Arm.*

[7] Koshal, D., 1993, *Manufacturing Engineer's Reference Book, Elsevier (pp. 328-348).*

[8] Naiza Binti Adnan, 2002, *Designing A Controller for X-Y Robot Using Programmable Logic Controller (PLC),* B. ENG. (HONS) Thesis, Universiti Teknologi PETRONAS

[9] Harry E.Stephanou, *Advanced Automation in Manufacturing and Service Industries*

[10] D. Saundgrs, M.F. Crockett, M.D. Novels, G.B. Hackwell, R.W. Parkinson, *Linking Graphical Simulation to The Control of Flexible Manufacturing Systems*

[11] C M Davidson, J McWhinne, *Engineering the Control Software Development Process*

[12] J. G. Gilberl, PE G. R. Diehl, PE, *Application Of Programmable Logic Controllers To Substation Control And Protection,* IEEE Transactions on Power Delivery, Vol. 9, No. 1, January 1994

[13] Bolton, W. *Programmable Logic Controllers, 4th Edition, Elsevier. (pp. 1-16).*

[14] Andrew Rae, *Helping the Operator in the Loop: Practical Human Machine Interface Principles for Safe Computer Controlled Systems*

[15] Dong-Han Ham, Wan Chul Yoon, Byoung-Tae Han, *Experimental study on the effects of visualized functionally abstracted information on process control tasks,* Reliability Engineering and System Safety 93 (2008) 254–270

[16] Qiuli Yum,Vincent Duffy,John McGinley,Zachary Rowland, *Productivity Simulation With Promodel For An Automotive Assembly Workstation Involving A Lift Assist Device*

[17] Hyung Seok Kim, Jae Young Lee, Wook Hyun Kwon, *A Compiler Design for IEC 1131-3 Standard Languages of Programmable Logic Controllers*

[18] Whitt, Michael, *Successful Instrumentation and Control Systems Design. (pp. 98-131). ISA.*

[19] R K Mittal, I J Nagrath, *Robotics and Control,2nd Edition 2003, Tata McGraw Hill (pp1-392)*

[20] James L. Fuller, *Robotic: Introduction, Programming, and Projects, 2nd Edition 1999, Prentice Hall (pp3-147)*

APPENDICES

APPENDIX A

GANNT CHART

Suggested Milestone for the First Semester of 2-Semester Final Year Project

No.	Detail/ Week	1	2	3	4	5	6	7	8	9	10		11	12	13	14
1	Selection of Project Topic	■	■									Mid-semester break				
2	Preliminary Research Work		■	■	■											
3	Submission of Preliminary Report				●											
4	Seminar 1 (optional)					■	■	■								
5	Project Work					■	■	■								
6	Submission of Progress Report								●							
7	Seminar 2 (compulsory)									■	■		■	■		
8	Project work continues								■	■	■		■	■		
9	Submission of Interim Report Final Draft														●	
10	Oral Presentation															●

● Suggested milestone

■ Process

Suggested Milestone for the Second Semester of 2-Semester Final Year Project

No.	Detail/ Week	1	2	3	4	5	6	7	8	9		10	11	12	13	14
1	Testing on the FMS, HMI development	■	■	■							Mid-semester break					
2	Submission of Progress Report 1				●											
3	HMI and PLC programming development				■	■	■	■								
4	Submission of Progress Report 2								●							
5	Seminar									●						
5	Project work continue - Testing								■	■		■	■			
6	Poster Exhibition													●		
7	Submission of Dissertation (soft bound)														●	
8	Oral Presentation															●
9	Submission of Project Dissertation (Hard Bound)															●

● Suggested milestone

■ Process

APPENDIX B

PLC DATA AREA

(Courtesy of Omron Operation Manual)

1. First take the absolute value (12345) and convert to unsigned b

ıumber	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
ıents	0	0	1	1	0	0	0	0	0	0	1	1	1	0	0	1

2. Next take the complement:

umber	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
ents	1	1	0	0	1	1	1	1	1	1	0	0	0	1	1	0

3. Finally add one:

ımber	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
ınts	1	1	0	0	1	1	1	1	1	1	0	0	0	1	1	1

Reverse the procedure to convert negative signed binary data to decimal.

IR (Internal Relay) Area

The IR area is used both as data to control I/O points, and as work bits to manipulate and store data internally. It is accessible both by bit and by word. In the C200HX/HG/HE PC, the IR area is comprised of words IR 000 to IR 235 (IR area 1) and IR 300 to IR 511 (IR area 2). Basic instructions have somewhat longer execution times when they access IR area 2 rather than IR area 1.

Words in the IR area that are used to control I/O points are called I/O words. Bits in I/O words are called I/O bits. Bits in the IR area which are not assigned as I/O bits can be used as work bits. IR area work bits are reset when power is interrupted or PC operation is stopped.

Area		Range
IR Area 1	I/O Area 1	IR 000 to IR 029
	Group-2 High-density I/O Unit Area 1 and B7A Interface Unit Area	IR 030 to IR 049
	SYSMAC BUS Area	IR 050 to IR 099
	Special I/O Unit Area 1	IR 100 to IR 199
	Optical I/O Unit and I/O Terminal Area	IR 200 to IR 231
	Work Area	IR 232 to IR 235
IR Area 2	I/O Area 2	IR 300 to IR 309
	Work Area	IR 310 to IR 329
	Group-2 High-density I/O Unit Area 2	IR 330 to IR 341
	Work Area	IR 342 to IR 349
	Special I/O Unit Area 2	IR 350 to IR 459
	Work Area	IR 460 to IR 511

ds

If a Unit brings inputs into the PC, the bit assigned to it is an input bit; if the Unit sends an output from the PC, the bit is an output bit. To turn on an output, the output bit assigned to it must be turned ON. When an input turns on, the input bit assigned to it also turns ON. These facts can be used in the program to access input status and control output status through I/O bits.

Usage

Input bits can be used to directly input external signals to the PC and can be used in any order in programming. Each input bit can also be used in as many instructions as required to achieve effective and proper control. They cannot be used in instructions that control bit status, e.g., the OUTPUT, DIFFERENTIATION UP, and KEEP instructions.

Output Bit Usage

Output bits are used to output program execution results and can be used in a order in programming. Because outputs are refreshed only once during ea cycle (i.e., once each time the program is executed), any output bit can be us in only one instruction that controls its status, including OUT, KEEP(1 DIFU(13), DIFD(14) and SFT(10). If an output bit is used in more than one su instruction, only the status determined by the last instruction will actually be o put from the PC.

See *5-15-1 Shift Register – SFT(10)* for an example that uses an output bit in t 'bit-control' instructions.

Word Allocation for Racks

I/O words are allocated to the CPU Rack and Expansion I/O Racks by slot pc tion. One I/O word is allocated to each slot, as shown in the following table. Si each slot is allocated only one I/O word, a 3-slot rack uses only the first 3 wor a 5-slot rack uses only the first 5 words, and an 8-slot rack uses only the fir words. Words that are allocated to unused or nonexistent slots are availabl work words.

← Left side of rack Right side of a 10-slot rack

Rack	Slot 1	Slot 2	Slot 3	Slot 4	Slot 5	Slot 6	Slot 7	Slot 8	Slot 9	Slot
CPU	IR 000	IR 001	IR 002	IR 003	IR 004	IR 005	IR 006	IR 007	IR 008	IR 00
1st Expansion	IR 010	IR 011	IR 012	IR 013	IR 014	IR 015	IR 016	IR 017	IR 018	IR 01
2nd Expansion	IR 020	IR 021	IR 022	IR 023	IR 024	IR 025	IR 026	IR 027	IR 028	IR 02
3rd Expansion	IR 300	IR 301	IR 302	IR 303	IR 304	IR 305	IR 306	IR 307	IR 308	IR 3C

Unused Words

Any words allocated to a Unit that does not use them can be used in progr ming as work words and bits. Units that do not used the words assigned tc slot they are mounted to include Link Units (e.g.. Host Link Units, PC Link U SYSMAC NET Link Units, etc.), Remote I/O Master Units, Special I/O U Group-2 High-density I/O Units, Group-2 B7A Interface Units, and Auxi Power Supply Units.

Allocation for Special I/O Units and Slave Racks

In most C200HX/HG/HE PCs, up to sixteen Special I/O Units may be mount any slot of the CPU Rack or Expansion I/O Racks. (A limited number of Sp I/O Units can be installed in Remote I/O Slave Racks, too.) Each Special I/C is allocated ten words based on its unit number (0 to F).

Up to ten Special I/O Units may be mounted in the C200HE-CPU□□-E C200HG/HX-CPU3□-E/4□-E PCs. Each Unit is allocated ten words base its unit number (0 to 9).

Unit number	I/O words	PC Restrictions
0	IR 100 to IR 109	None
1	IR 110 to IR 119	
2	IR 120 to IR 129	
3	IR 130 to IR 139	
4	IR 140 to IR 149	
5	IR 150 to IR 159	
6	IR 160 to IR 169	
7	IR 170 to IR 179	
8	IR 180 to IR 189	
9	IR 190 to IR 199	

Unit number	I/O words	PC Restrictions
A	IR 400 to IR 409	Not available in C200HE-CPU□□-E and C200HG/HX-CPU3□-E/4□-E PCs.
B	IR 410 to IR 419	
C	IR 420 to IR 429	
D	IR 430 to IR 439	
E	IR 440 to IR 449	
F	IR 450 to IR 459	

Note I/O words that aren't allocated to Special I/O Units can be used as work words.

Up to five Slave Racks may be used, whether one or two Masters are used. IR area words are allocated to Slave Racks by the unit number on the Unit, as shown in the following tables.

Unit number	I/O words
0	IR 050 to IR 059
1	IR 060 to IR 069
2	IR 070 to IR 079
3	IR 080 to IR 089
4	IR 090 to IR 099

The C500-RT001/002-(P)V1 Remote I/O Slave Rack may be used, but it requires 20 I/O words, not 10, and therefore occupies the I/O words allocated to 2 C200H Slave Racks, both the words allocated to the unit number set on the rack and the words allocated to the following unit number. When using a C200HX/HG/HE CPU, do not set the unit number on a C500 Slave Rack to 4, because there is no unit number 5. With the C500 Slave Rack, I/O words are allocated only to installed Units, from left to right, and not to slots as in the C200HX/HG/HE Racks.

tion for Optical I/O and I/O Terminals

I/O words between IR 200 and IR 231 are allocated to Optical I/O Units and I/O Terminals by unit number. The I/O word allocated to each Unit is IR 200+n, where n is the unit number set on the Unit.

tion for Remote I/O r and Link Units

Remote Master I/O Units and Host Link Units do not use I/O words, and the PC Link Units use the LR area, so words allocated to the slots in which these Units are mounted are available as work words.

location for I/O Units

An I/O Unit may require anywhere from 8 to 16 bits, depending on the model. With most I/O Units, any bits not used for input or output are available as work bits. Transistor Output Units C200H-OD213 and C200H-OD411, as well as Triac Output Unit C200H-OA221, however, uses bit 08 for the Blown Fuse Flag. Transistor Output Unit C200H-OD214 uses bits 08 to 11 for the Alarm Flag. Bits 08 to 15 of any word allocated to these Units, therefore, cannot be used as work bits.

location for Interrupt Units

The Interrupt Input Unit uses the 8 bits of the first I/O word allocated to its slot in the CPU Rack. (An Interrupt Input Unit will operate as a normal Input Unit when installed in an Expansion I/O Rack.) The other 24 bits allocated to its slot in the CPU Rack can be used as work bits.

Allocation for Group-2 High-density I/O Units and B7 Interface Units

Group-2 High-density I/O Units and B7A Interface Units are allocated words between IR 030 and IR 049 according to I/O number settings made on them and do not use the words allocated to the slots in which they are mounted. For 32-point Units, each Unit is allocated two words; for 64-point Units, each Unit is allocated four words. The words allocated for each I/O number are in the following tables. Any words or part of words not used for I/O can be used as work words or bits in programming:

32-point Units		64-point Units	
I/O number	Words	I/O number	Words
0	IR 30 to IR 31	0	IR 30 to IR 33
1	IR 32 to IR 33	1	IR 32 to IR 35
2	IR 34 to IR 35	2	IR 34 to IR 37
3	IR 36 to IR 37	3	IR 36 to IR 39
4	IR 38 to IR 39	4	IR 38 to IR 41
5	IR 40 to IR 41	5	IR 40 to IR 43
6	IR 42 to IR 43	6	IR 42 to IR 45
7	IR 44 to IR 45	7	IR 44 to IR 47
8	IR 46 to IR 47	8	IR 46 to IR 49
9	IR 48 to IR 49	9	Cannot be used.

When setting I/O numbers on the High-density I/O Units and B7A Interface Units, be sure that the settings will not cause the same words to be allocated to more than one Unit. For example, if I/O number 0 is allocated to a 64-point Unit, I/O number 1 cannot be used for any Unit in the system.

Group-2 High-density I/O Units and B7A Interface Units are not considered Special I/O Units and do not affect the limit to the number of Special I/O Units allowed in the System, regardless of the number used.

The words allocated to Group-2 High-density I/O Units correspond to the connectors on the Units as shown in the following table.

Unit	Word	Connector/row
32-point Units	First	Row A
	Second	Row B
64-point Units	First	CN1, row A
	Second	CN1, row B
	Third	CN2, row A
	Fourth	CN2, row B

Note Group-2 High-density I/O Units and B7A Interface Units cannot be mounted to Slave Racks.

3-4 SR (Special Relay) Area

The SR area contains flags and control bits used for monitoring PC operation, accessing clock pulses, and signalling errors. SR area word addresses range from 236 through 299; bit addresses, from 23600 through 29915.

The SR areas is divided into two sections. The first section ends at SR 255 and the second section begins at SR 256. When an SR area word is used as an operand in an instruction, the operand mustn't cross over this boundary. Basic instructions that access bits in the SR Area 2 have longer execution times.

Area	Range
SR Area 1	SR 23600 to SR 25507
SR Area 2	SR 25600 to SR 29915

The following table lists the functions of SR area flags and control bits. Most of these bits are described in more detail following the table. Descriptions are in order by bit number except that Link System bits are grouped together.

Unless otherwise stated, flags are OFF until the specified condition arises, when they are turned ON. Restart bits are usually OFF, but when the user turns one ON then OFF, the specified Link Unit will be restarted. Other control bits are OFF until set by the user.

Not all SR words and bits are writeable by the user. Be sure to check the function of a bit or word before attempting to use it in programming.

rd(s)	Bit(s)	Function
	00 to 07	Node loop status output area for operating level 0 of SYSMAC NET Link System
	08 to 15	Node loop status output area for operating level 1 of SYSMAC NET Link System
	00 to 07	Completion code output area for operating level 0 following execution of SEND(90)/RECV(98) SYSMAC LINK/SYSMAC NET Link System
	08 to 15	Completion code output area for operating level 1 following execution of SEND(90)/RECV(98) SYSMAC LINK/SYSMAC NET Link System
ıd 241	00 to 15	Data link status output area for operating level 0 of SYSMAC LINK or SYSMAC NET Link System
ıd 245	00 to 15	Data link status output area for operating level 1 of SYSMAC LINK or SYSMAC NET Link System
	00 to 15	Not used
ıd 248	00 to 07	PC Link Unit Run Flags for Units 16 through 31 or data link status for operating level 1
	08 to 15	PC Link Unit Error Flags for Units 16 through 31 or data link status for operating level 1
ıd 250	00 to 07	PC Link Unit Run Flags for Units 00 through 15 or data link status for operating level 0
	08 to 15	PC Link Unit Error Flags for Units 00 through 15 or data link status for operating level 0
	00	Remote I/O Error Read Bit
ıble	01 to 02	Not used
	03	Remote I/O Error Flag
	04 to 06	Slave Rack number and unit number of Remote I/O Unit, Optical I/O Unit, or I/O Terminal with error
	07	Not used
	08 to 15	Master's unit number and word allocated to Remote I/O Unit, Optical I/O Unit, or I/O Terminal with error (Hexadecimal)
	00	SEND(90)/RECV(98) Error Flag for operating level 0 of SYSMAC LINK or SYSMAC NET Link System
	01	SEND(90)/RECV(98) Enable Flag for operating level 0 of SYSMAC LINK or SYSMAC NET Link System
	02	Operating Level 0 Data Link Operating Flag
	03	SEND(90)/RECV(98) Error Flag for operating level 1 of SYSMAC LINK or SYSMAC NET Link System
	04	SEND(90)/RECV(98) Enable Flag for operating level 1 of SYSMAC LINK or SYSMAC NET Link System
	05	Operating Level 1 Data Link Operating Flag
	06	Rack-mounting Host Link Unit Level 1 Communications Error Flag
	07	Rack-mounting Host Link Unit Level 1 Restart Bit
	08	RS-232C Port Error Flag
	09	RS-232C Port Restart Bit
	10	PC Setup Clear Bit
	11	Forced Status Hold Bit
	12	Data Retention Control Bit
	13	Rack-mounting Host Link Unit Level 0 Restart Bit
	14	Not used.
	15	Output OFF Bit

Word(s)	Bit(s)	Function	
253	00 to 07	FAL number output area (see error information provided elsewhere)	
	08	Low Battery Flag	
	09	Cycle Time Error Flag	
	10	I/O Verification Error Flag	
	11	Rack-mounting Host Link Unit Level 0 Communications Error Flag	
	12	Remote I/O Error Flag	
	13	Always ON Flag	
	14	Always OFF Flag	
	15	First Cycle Flag	
254	00	1-minute clock pulse bit	
	01	0.02-second clock pulse bit	
	02	Negative (N) Flag	
	03	MTR Execution Flag	
	04	Overflow Flag (for signed binary calculations)	
	05	Underflow Flag (for signed binary calculations)	
	06	Differential Monitor End Flag	
	07	Step Flag	
	08	HKY Execution Flag	
	09	7SEG Execution Flag	
	10	DSW Execution Flag	
	11	Interrupt Input Unit Error Flag	
	12	First cycle flag	
	13	Interrupt Program Error Flag	
	14	Group-2 Error Flag	
	15	Special Unit Error Flag (includes Special I/O, PC Link, Host Link, Remote I/O Master Units)	
255	00	0.1-second clock pulse bit	
	01	0.2-second clock pulse bit	
	02	1.0-second clock pulse bit	
	03	Instruction Execution Error (ER) Flag	These flags are turned OFF when the END(01) instruction is executed, so their status can't be monitored from a Programming Console. Refer to Appendix C for a table showing which instructions affect these flags.
	04	Carry (CY) Flag	
	05	Greater Than (GR) Flag	
	06	Equals (EQ) Flag	
	07	Less Than (LE) Flag	
	08 to 15	Reserved by system (used for TR bits)	
256 to 261	00 to 15	Reserved by system	
262	00 to 15	Longest interrupt subroutine (action) execution time (0.1-ms units)	
263	00 to 15	Number of interrupt subroutine (action) with longest execution time. (8000 to 8255) (Bit 15 is the Interrupt Flag)	

rd(s)	Bit(s)	Function
	00 to 03	RS-232C Port Error Code 0: No error 1: Parity error 2: Framing error 3: Overrun error 4: FCS error 5: Timeout error 6: Checksum error 7: Command error
	04	RS-232C Port Communications Error
	05	RS-232C Port Send Ready Flag
	06	RS-232C Port Reception Completed Flag
	07	RS-232C Port Reception Overflow Flag
	08 to 11	Peripheral Port Error Code in General I/O Mode 0: No error 1: Parity error 2: Framing error 3: Overrun error 4: FCS error 5: Timeout error 6: Checksum error 7: Command error
	12	Peripheral Port Communications Error in General I/O Mode
	13	Peripheral Port Send Ready Flag in in General I/O Mode
	14	Peripheral Port Reception Completed Flag in General I/O Mode
	15	Peripheral Port Reception Overflow Flag in General I/O Mode
	00 to 15	NT Link (1:N) Mode Bits 00 to 07: Communicating with PT Flags for Units 0 to 7 Bits 08 to 15: Registering PT Priority Flags for Units 0 to 7 RS-232C Mode Bits 00 to 15: RS-232C Port Reception Counter
	00 to 15	Peripheral Reception Counter in RS-232C Mode
	00 to 04	Reserved by system (not accessible by user)
	05	Host Link Level 0 Send Ready Flag
	06 to 12	Reserved by system (not accessible by user)
	13	Host Link Level 1 Send Ready Flag
	14 to 15	Not used.
	00 to 15	Communications Board Error Information
	00 to 07	Memory Cassette Contents 00: Nothing; 01: UM; 02: IOM; 03: HIS
	08 to 10	Memory Cassette Capacity 0: 0 KW (no cassette); 2: 4 or 8 KW; 3: 16 KW; 4: 32 KW
	11 to 13	Reserved by system (not accessible by user)
	14	EEPROM Memory Cassette Protected or EPROM Memory Cassette Mounted Flag
	15	Memory Cassette Flag

Word(s)	Bit(s)	Function	
270	00	Save UM to Cassette Bit	Data transferred when the Bit is turned ON in PROGRAM mode. Bit will automatically turn OFF. A non-fatal error will occur if these bits are turned ON in RUN or MONITOR modes.
	01	Load UM from Cassette Bit	
	02	Compare UM to Cassette Bit	
	03	Comparison Results 0: Contents identical; 1: Contents differ or comparison not possible	
	04 to 10	Not used.	
	11	Transfer Error Flag: Transferring SYSMAC NET data link table on UM during active data link.	Data will not be transferred from UM to the Memory Cassette if an error occurs (except for Board Checksum Error). Detailed information on checksum errors occurring in the Memory Cassette will not be output to SR 272 because the information is not needed. Repeat the transmission if SR 27015 is ON.
	12	Transfer Error Flag: Not PROGRAM mode	
	13	Transfer Error Flag: Read Only	
	14	Transfer Error Flag: Insufficient Capacity or No UM	
	15	Transfer Error Flag: Board Checksum Error	
271	00 to 07	Ladder program size stored in Memory Cassette Ladder-only File: 04: 4 KW; 08: 8 KW; 12: 12 KW; ... (32: 32 KW) 00: No ladder program or a file other than a ladder program has been stored.	
	08 to 15	Ladder program size and type in CPU (Specifications are the same as for bits 00 to 07.)	
272	00 to 10	Not used.	
	11	Memory Error Flag: PC Setup Checksum Error	
	12	Memory Error Flag: Ladder Checksum Error	
	13	Memory Error Flag: Instruction Change Vector Area Checksum Error	
	14	Memory Error Flag: Memory Cassette Online Disconnection	
	15	Memory Error Flag: Autoboot Error	
273	00	Save IOM to Cassette Bit	Data transferred to Memory Cassette when Bit is turned ON in PROGRAM mode. Bit will automatically turn OFF. An error will be produced if turned ON in any other mode.
	01	Load IOM from Cassette Bit	
	02	Set this bit to 0.	
	03 to 07	Not used.	
	08 to 11	Contains the EM bank number when the Memory Cassette contains IOM data.	
	12	Transfer Error Flag: Not PROGRAM mode	Data will not be transferred from IOM to the Memory Cassette if an error occurs (except for Read Only Error).
	13	Transfer Error Flag: Read Only	
	14	Transfer Error Flag: Insufficient Capacity or No IOM	
	15	Always 0.	

rd(s)	Bit(s)	Function	
	00	Special I/O Unit #0 Restart Flag	These flags will turn ON during restart processing. These flags will not turn ON for Units on Slave Racks.
	01	Special I/O Unit #1 Restart Flag	
	02	Special I/O Unit #2 Restart Flag	
	03	Special I/O Unit #3 Restart Flag	
	04	Special I/O Unit #4 Restart Flag	
	05	Special I/O Unit #5 Restart Flag	
	06	Special I/O Unit #6 Restart Flag	
	07	Special I/O Unit #7 Restart Flag	
	08	Special I/O Unit #8 Restart Flag	
	09	Special I/O Unit #9 Restart Flag	
	10	Special I/O Unit #A Restart Flag	
	11	Special I/O Unit #B Restart Flag	
	12	Special I/O Unit #C Restart Flag	
	13	Special I/O Unit #D Restart Flag	
	14	Special I/O Unit #E Restart Flag	
	15	Special I/O Unit #F Restart Flag	
	00	PC Setup Error (DM 6600 to DM 6605)	
	01	PC Setup Error (DM 6613 to DM 6623)	
	02	PC Setup Error (DM 6635 to DM 6655)	
	03	Not used.	
	04	Changing RS-232C Setup Flag	
	05	Not used.	
	06 to 07	Reserved by system (not accessible by user)	
	08 to 15	Not used.	
	00 to 07	Minutes (00 to 59)	Indicates the current time in BCD.
	08 to 15	Hours (00 to 23)	
279	00 to 15	Used for keyboard mapping. See page 396.	
	00 to 15	Group-2 High-density I/O Unit Error Flags for Units 0 to F (AR 0205 to AR 0214 also function as Error Flags for Units 0 to 9.)	
	00 to 15	Special I/O Unit Restart Bits for Units 0 to F (Units 0 to 9 can also be restarted with Special I/O Unit Restart Bits AR 0100 to AR 0109.)	
	00 to 15	Special I/O Unit Error Flags for Units 0 to F (AR 0000 to AR 0009 also function as Error Flags for Units 0 to 9.)	
286	00 to 15	Communications Board monitoring area	
288	00 to 15	Communications Board interrupt data area	
	00 to 07	Communications Board general monitoring area	
	08	Communications Board Port A Instruction Execution Flag	
	09 to 10	Used by Communications Board Port A instructions	
	11	Communications Board Port A Instruction Abort Bit	
	12	Communications Board Port B Instruction Execution Flag	
	13 to 14	Used by Communications Board Port B instructions	
	15	Communications Board Port B Instruction Abort Bit	
o 293	00 to 15	Macro Area inputs.	
o 297	00 to 15	Macro Area outputs.	
o 299	00 to 15	Reserved by system (not accessible by user)	

3-4-1 SYSMAC NET/SYSMAC LINK System

Loop Status

SR 236 provides the local node loop status for SYSMAC NET Syste shown below.

—	Bit in SR 236						
Level 0	**07**	**06**	**05**	**04**	**03**	**02**	**01**
Level 1	**15**	**14**	**13**	**12**	**11**	**10**	**09**
Status/ Meaning	1	1	Central Power Supply 0: Connected 1: Not connected	1	Loop Status 11: Normal loop 10: Downstream backloop 01: Upstream backloop 00: Loop error		Reception Status 0: Reception enabled 1: Reception disabled

Completion Codes

SR 23700 to SR23707 provide the SEND/RECV completion code for op level 0 and SR 23708 to SR 23215 provide the SEND/RECV completion c operating level 1. The completion codes are as given in the following t

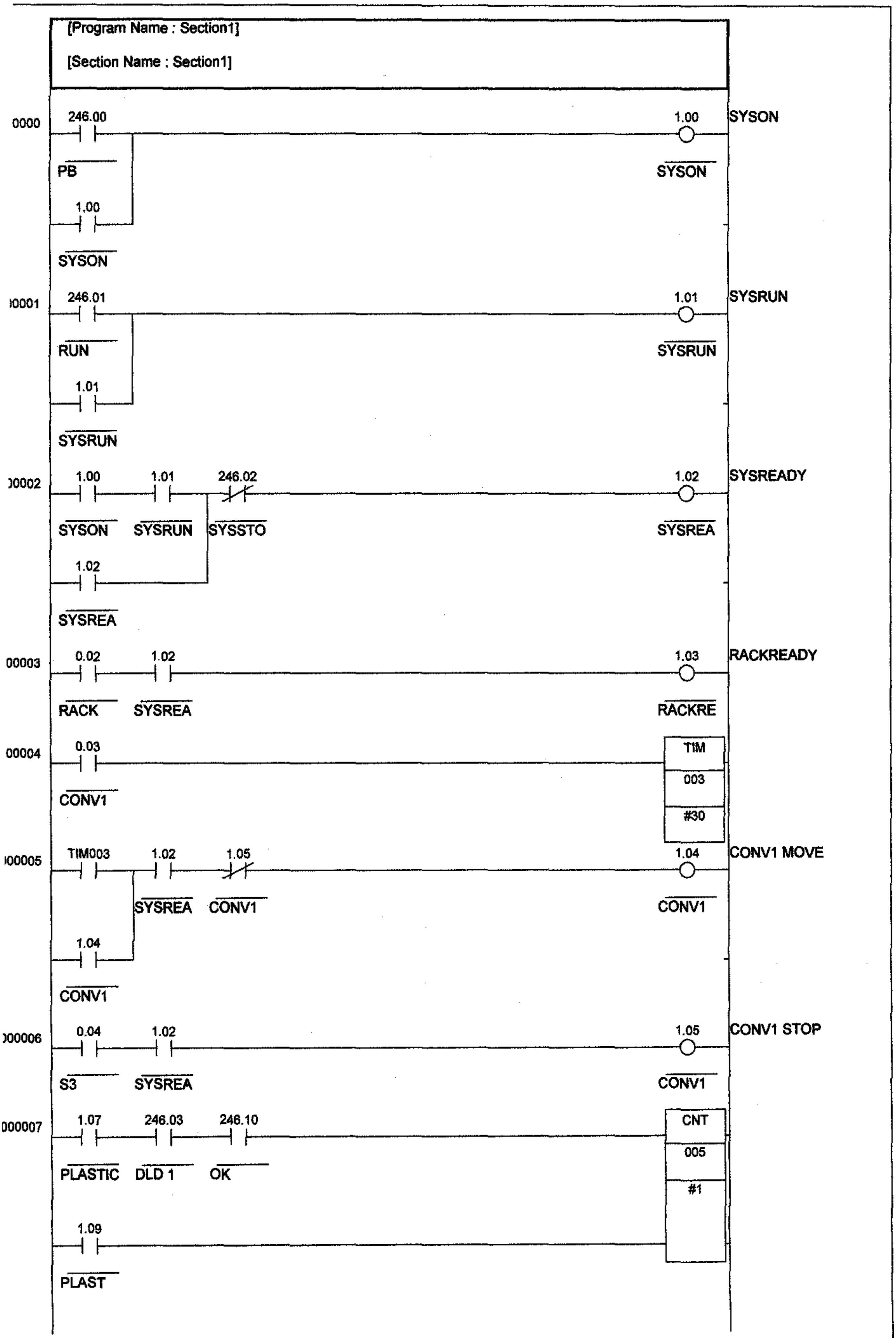
SYSMAC LINK

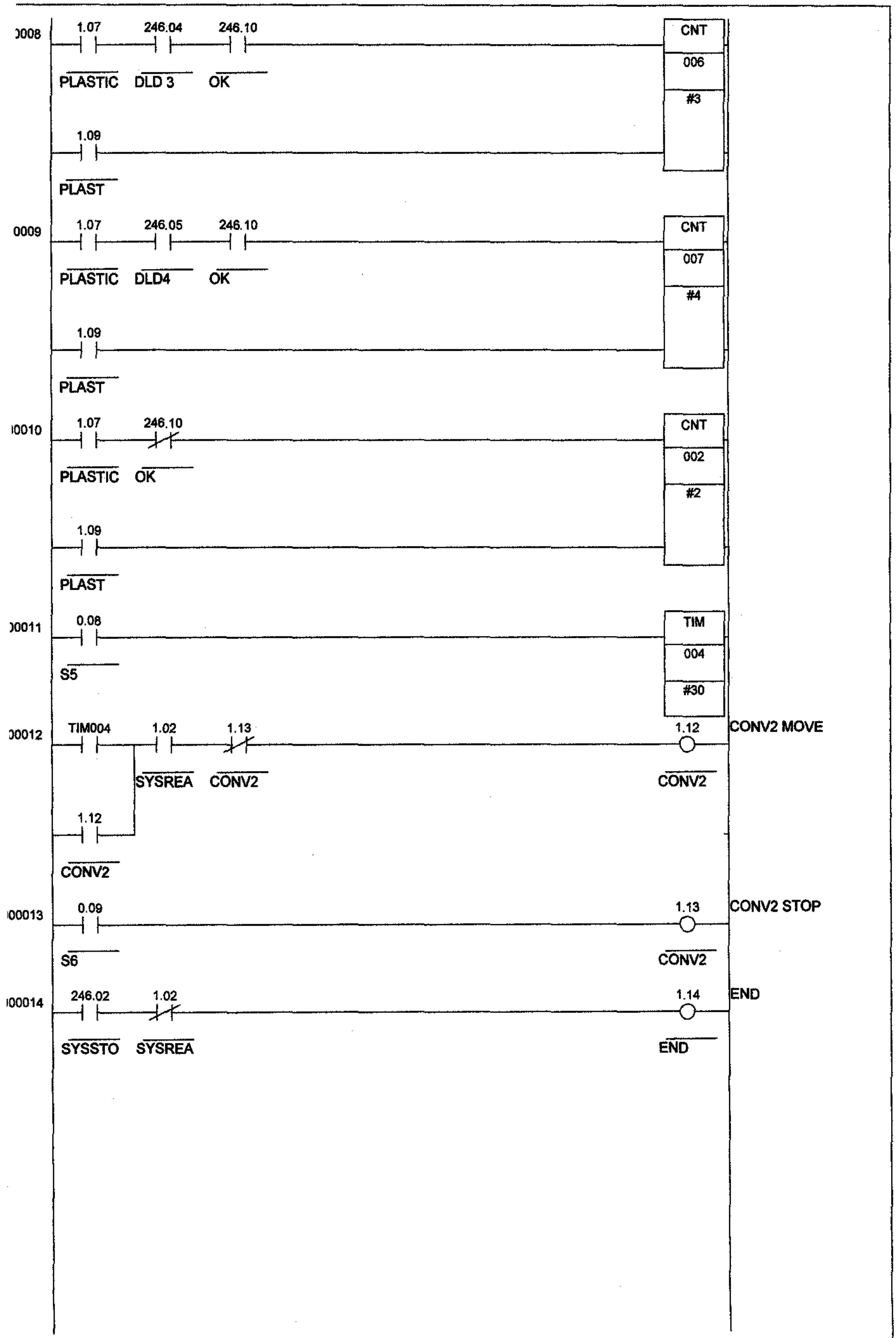
Code	Item	Meaning
00	Normal end	Processing ended normally.
01	Parameter error	Parameters for network communication instruc not within acceptable ranges.
02	Unable to send	Unit reset during command processing or local in not in network.
03	Destination not in network	Destination node is not in network.
04	Busy error	The destination node is processing data and c receive the command.
05	Response timeout	The response monitoring time was exceeded.
06	Response error	There was an error in the response received fr the destination node.
07	Communications controller error	An error occurred in the communications contr
08	Setting error	There is an error in the node address settings.
09	PC error	An error occurred in the CPU of the destinatior node.

SYSMAC NET

Code	Item	Meaning
00	Normal end	Processing ended normally.
01	Parameter error	Parameters for network communication instruc not within acceptable ranges.
02	Routing error	There is a mistake in the routing tables for connection to a remote network.
03	Busy error	The destination node is processing data and c receive the command.
04	Send error (token lost)	The token was not received from the Line Ser
05	Loop error	An error occurred in the communications loop.
06	No response	The destination node does not exist or the res monitoring time was exceeded.
07	Response error	There is an error in the response format.

APPENDIX C
PLC PROGRAMMING


[Program Name : Section1]
[Section Name : Section1]
0000
246.00
PB
1.00
SYSON
1.00
SYSON
SYSON
)0001
246.01
RUN
1.01
SYSRUN
1.01
SYSRUN
SYSRUN
)0002
1.00
SYSON
1.01
SYSRUN
246.02
SYSSTO
1.02
SYSREA
1.02
SYSREA
SYSREADY
00003
0.02
RACK
1.02
SYSREA
1.03
RACKRE
RACKREADY
00004
0.03
CONV1
TIM
003
#30
)00005
TIM003
1.02
SYSREA
1.05
CONV1
1.04
CONV1
1.04
CONV1
CONV1 MOVE
)00006
0.04
S3
1.02
SYSREA
1.05
CONV1
CONV1 STOP
000007
1.07
PLASTIC
246.03
DLD 1
246.10
OK
1.09
PLAST
CNT
005
#1



)008
1.07 246.04 246.10
PLASTIC DLD 3 OK
CNT
006
#3
1.09
PLAST
0009
1.07 246.05 246.10
PLASTIC DLD4 OK
CNT
007
#4
1.09
PLAST
)0010
1.07 246.10
PLASTIC OK
CNT
002
#2
1.09
PLAST
)0011
0.08
S5
TIM
004
#30
)0012
TIM004 1.02 1.13
SYSREA CONV2
1.12
CONV2
1.12
CONV2
CONV2 MOVE
)00013
0.09
S6
1.13
CONV2
CONV2 STOP
)00014
246.02 1.02
SYSSTO SYSREA
1.14
END
END

APPENDIX D

VB PROGRAMMING

VISUAL BASIC PROGRAMMING FOR HMI

FORM 1 – MAIN PAGE

```
Private Sub Activate_Click()

Dim intactivate As Integer
intactivate = MsgBox("Start Operation?", vbYesNo + vbQuestion, "Connecting to PLC")
If intactivate = vbYes Then
SYSMAC_C1.writeArea plcAreaCIO, 246, 1, "0001"

 activate.Enabled = False
 run.Enabled = True
 cancel.Enabled = True

 End If
End Sub

Private Sub cancel_Click()

Unload Me
Unload Form2
Unload Form3
Unload Form4

End Sub

Private Sub Form_Load()

SYSMAC_C1.RunMode = plcRunmodeMONITOR
  run.Enabled = False

End Sub
```

```
Private Sub run_Click()

Form2.Show
 Form1.Hide

 End Sub

Private Sub stop_Click()

SYSMAC_C1.writeArea plcAreaCIO, 246, 1, "0004"
 activate.Enabled = True
 run.Enabled = False
 cancel.Enabled = True

End Sub
```

FORM 2 – PROGRAM OPTION MODE

```
Private Sub ok_Click()

If Optmanual.Value = True Then

Form4.Show
Unload Me

End If

If Optauto.Value = True Then

Form3.Show
Unload Me

End If

End Sub
```

```
Private Sub cancel_Click()

Unload Me
Form1.Show

End Sub

Private Sub Form_Load()

Optmanual.Value = True

End Sub
```

FORM 3 – AUTOMATIC MODE

```
Private Sub Form_Load()
Shape1.FillColor = &HC0C0FF
Shape2.FillColor = &HC0C0FF
Shape3.FillColor = vbYellow

End Sub

Private Sub STOP1_Click()

SYSMAC_C1.writeArea plcAreaCIO, 246, 1, "0004"
Shape1.FillColor = &HC0C0FF
Shape2.FillColor = vbRed
Shape3.FillColor = &HC0C0FF

End Sub

Private Sub Run2_Click()

SYSMAC_C1.writeArea plcAreaCIO, 246, 1, "0002"
```

```
Private Sub ok1_Click()

If Combo1.Text = "One (1) " Then
 SYSMAC_C1.writeArea plcAreaCIO, 246, 1, "0008"

End If

If Combo1.Text = "Two (2) " Then
 'SYSMAC_C1.writeArea plcAreaCIO, 246, 1, "1010"

If Combo1.Text = "Three (3) " Then
 SYSMAC_C1.writeArea plcAreaCIO, 246, 1, "0010"

End If

If Combo1.Text = "Four (4) " Then
 SYSMAC_C1.writeArea plcAreaCIO, 246, 1, "0020"


End If
End Sub

Private Sub STOP3_Click()

SYSMAC_C1.writeArea plcAreaCIO, 246, 1, "0004"
Shape1.FillColor = &HC0C0FF
Shape2.FillColor = vbRed
Shape3.FillColor = &HC0C0FF
End Sub
```