UNIVERSITI TUN HUSSEIN ONN MALAYSIA

PENGESAHAN STATUS LAPORAN PROJEK SARJANA

A Training Model of An Automated Storage and Retrieval System (AS/RS) With Customized Warehouse Management System (WMS)

SESI PENGAJIAN : 2006/2007

Saya **TEE KIAN SEK** mengaku membenarkan laporan projek sarjana ini disimpan di Perpustakaan dengan syarat-syarat kegunaan seperti berikut:

1. Tesis adalah hakmilik Universiti Tun Hussein Onn Malaysia.
2. Perpustakaan dibenarkan membuat salinan untuk tujuan pengajian sahaja.
3. Perpustakaan dibenarkan membuat salinan tesis ini sebagai bahan pertukaran antara institusi pengajian tinggi.
4. **Sila tandakan (√)

□ **SULIT** (Mengandungi maklumat yang berdarjah keselamatan atau kepentingan Malaysia seperti yang termaktub di dalam AKTA RAHSIA RASMI 1972)

□ **TERHAD** (Mengandungi maklumat TERHAD yang telah ditentukan oleh organisasi/badan di mana penyelidikan dijalankan)

☑ **TIDAK TERHAD**

Disahkan oleh:

(TANDATANGAN PENULIS)

(TANDATANGAN PENYELIA)

Alamat Tetap:

**51, JALAN PERDANA 2/30, TAMAN
BUKIT PERDANA 2, 83000 BATU
PAHAT, JOHOR.**

PROF. MADYA DR. ZAINAL ALAM BIN HARON
(Nama Penyelia)

Tarikh: **28 MEI 2007**

Tarikh: **28 MEI 2007**

CATATAN:

** Jika tesis ini SULIT atau TERHAD, sila lampirkan surat daripada pihak berkuasa/organisasi berkenaan dengan menyatakan sekali tempoh tesis ini perlu dikelaskan sebagai SULIT atau TERHAD.

"Saya~~/Kami~~ akui bahawa saya~~/kami~~ telah membaca karya ini dan pada pandagan saya~~/kami~~ karya ini adalah memadai dari segi skop dan kualiti untuk tujuan penganugerahan Ijazah Sarjana Kejuruteraan (Elektrikal)."

Tandatangan :

Nama Penyelia I : Prof. Madya Dr. Zainal Alam Bin Haron

Tarikh : 28 MAY 2007

A Training Model of an Automated Storage and Retrieval System (AS/RS)
With Customized Warehouse Management System (WMS)

TEE KIAN SEK

A thesis submitted
In fulfillment of the requirements for the adward of the
Degree of Master of Engineering (Electrical)

Faculty of Electrical and Electronic Engineering
Universiti Tun Hussein Onn Malaysia

May, 2007

"No part of the study was covered by copyright. References of infomation obtained from other source are specially quoted, otherwise the rest of the infomation presented through this study is the sole work and experimentation carried out by the author"

Signature : ....

Author : TEE KIAN SEK

Date : 9 May 2007

For my parents, wife and my newborn daughter.
Life becomes merrier with my baby, Chloe.
Wish all luck and happiness to them!

ACKNOWLEDGEMENT

It is a blessing that I can finish my study. All the way, I have been receiving warmness help and guide from the people around me. Special thank to P.M. Dr. Zainal Alam Bin Haron, whom as my academic adviser, has guided and shown me the wonder world in the research, with his friendly attitude, always encouraging me to approach him on many questions, not only limited to my study. Many thank to Mr. Fadzil Esa and Mr. Rosli Omar, whom have given me the convenience to approach all facilities available in the robotic laboratory. Lastly, I must thank my wife, Soon Chin Fhong, for her support and encouragement.

ABSTRACT

AS/RS is a key industrial automation system that has drastically reduced the workforce needed to run a warehouse. Via a computer-controlled system, many intensive labour jobs are taken over by the system, including tediously moving and sorting heavy load from the minute of receving until shipping to customers, intensive paperwork to record goods receiving and order receipts. Somehow, in real business, the system is always complex in the perspective of engineering considerations, depending on the nature of the business, tending to upgrading and modification from time to time. It is desirable that the engineering training curve would provide an engineer perspective in industry design concepts and contemporary technologies to the students, not in the operator prospective. This project is intended to develop a training model of AS/RS for the engineering students. The learning curves are provided through three levels in the system integration. The device level illustrates basic input and output devices that are carefully choosen. The controller level processes all input information from the input devices and host computer. The supervisory level implements graphic user interface for system monitoring and control for the operator. The training model also emphasizes in three design concepts, flexibility, expandability and modularity. Flexibility will allow a broad spectrum of application environments and extend application life. Expandability will allow application in areas not yet defined. Modularity will enhance modification and maintenance.

ABSTRAK

AS/RS merupakan satu sistem automasi yang penting dan mengurangkan tenaga pekerja yang ramai untuk beroperasi sebuah gudang. Dengan menggunakan kawalan komputer, banyak kerja buruh telah diambilalih, termasuk kerja-kerja pemindahan barang-barang dari saat penerimaan hingga penghantaran ke pelanggan serta mengurangkan beban kertas kerja penerimaan dan penghantaran. Namun, pada industri yang sebenar, sistem ini adalah rumit pada perpektif kejuruteraan dan amat bergantung kepada fungsi niaga. Dari masa ke masa, ia juga perlu dinaik-taraf dan diubah-suia. Pembelajaran ilmu kejuruteraan perlu dipandang di perspektif jurutera pada konsep-konsep rekabentuk dan teknologi terkini, bukannya di perspektif seorang operator. Di project ini, satu sistem pembelajaran AS/RS dibangunkan untuk pelajar-pelajar aliran kejuruteraan. Pembelajarannya dibentang dalam tiga peringkat. Peringkat peralatan menunjukan kegunaan dan pemilihan alat-alat perangsan dan aktuator. Peringkat kawalan akan memproses semua data daripada alat-alat perangsan dan komputer. Peringkat pengawasan menggunakan perantaraan muka grafik pengguna untuk kegunaan pengawasan dan kawalan di sisi operator. Sistem pembelajaran ini menekankan tiga konsep rekabentuk, iaitu kebolehlenturan, kebolehkembangan dan modulariti. Kebolehlenturan mempelbagaikan aplikasi dan memanjangkan hayat kebolehgunaan applikasi. Kebolehkembangan pula membenarkan aplikasi pada bidang yang belum ditentukan. Modulariti menggalakan pengubahsuaian dan penyenglenggaraan.

CONTENTS

CHAPTER	ITEM			PAGE
	ACKNOWLEDGEMENT			**iv**
	ABSTRACT			**v**
	ABSTRAK			**vi**
	CONTENTS			**vii**
	LIST OF FIGURES			**ix**
	LIST OF APPENDICES			**xii**
1	**THE INTRODUCTION**			**1**
	1.1	Objective		3
	1.2	Research Scope		3
2	**LITERATURE REVIEW**			**4**
3	**METHODOLOGY**			**9**
	3.1	The System Flow		9
	3.2	Modular Design		12
	3.3	The Racking System And The WMS		13
		3.3.1	The Coordinate System of The Racking System	19

	3.3.2	Software Planning of The Racking System	24
	3.4	Material Handling System (MHS)	52
	3.5	The Sorter Robot	53
	3.6	Communication	56
4	**RESULT AND DISCUSSION**		**58**
	4.1	The Device Level	59
	4.2	The Controller Level	60
	4.3	The Supervisory Level	61
5	**CONCLUSION AND SUGGESTION**		**70**
	5.1	Future Works	71
REFERENCE			**73**
APPENDIX A			**75**
APPENDIX B			**82**
APPENDIX C			**91**

LIST OF FIGURES

FIGURE	ITEM	PAGE
2.1	Plan layout of a multi-product AS/RS serviced by RGVs [2]	6
3.1	The System Layout	10
3.2	The tray, the product and the product in the tray	14
3.3	The Rack in row and column	14
3.4	The Racking and SRM	15
3.5	X-Axis and Z-Axis	15
3.6	The SRM pneumatic end-effectors for storing/retrieving	16
3.7	The SRM: Home position and (X, Z) direction	16
3.8	The PLC and Pulse IO Board [12]	17
3.9	Pulse Output Wiring Diagram [12]	18
3.10	The Racking System Panel	18
3.11	The Rack Coordinate System: (X, Z)	19
3.12	The Coordinate System Table	19
3.13	Z-axis Drive and Gear Ratio	20
3.14	X-Axis Drive ad Gear Ratio	21
3.15	The Timing Belt	21
3.16	Electronic Gear [15]	22
3.17	The Coordinate System in pulses number	24
3.18	ID Layer , Seq_No Layer and the data search sequence	26
3.19	Job-List	27
3.20	Memory Monitoring and Management	28

3.21	FinsGateway of Omron	29
3.22	Manual Mode	29
3.23	Manual Mode: X-Axis	30
3.24	Manual Mode: Z-Axis	31
3.25	Home Mode	32
3.26	Coordinate Teaching: X-axis	33
3.27	Coordinate Teaching: Z-Axis	34
3.28	Absolute Position Counter, X_abs and Z_abs	35
3.29	SRM Action Block, Job-List Block and Memory Block	37
3.30	The Timing Diagram of the SRM Action Block	37
3.31	Substances of SRM Action Block	39
3.32	Flow Chart of "Bridge" and "Storage"	40
3.33	Flow Chart of "Retrieval"	41
3.34	Flow Chart of Loading Sub	42
3.35	Flow Chart of Unloading Sub	43
3.36	Flow Chart of Store Sub	44
3.37	Flow Chart of Retrieve Sub	44
3.38	Job-List Block	45
3.39	Job-List Timing Chart	45
3.40	Memory Block	46
3.41	Timing Chart of Memory Block	46
3.42	Flow Chart of Bridge Mode in Memory Block	47
3.43	Flow Chart of Storage Mode in Memory Block	48
3.44	Flow Chart of Search_ID subroutine in Memory Block	49
3.45	Flow Chart of Retrieval Mode in Memory Block	50
3.46	Flow Chart of Memory_Manage in Memory Block	51
3.47	Function Blocks Interconnected	52
3.48	Material Handling Conveyor	53
3.49	Sorter Robot Interfacing with the MHS	54
3.50	Picking Point and Sorting Points A, B	54
3.51	Sorter Robot Program	55
3.52	Communication Layout	56
4.1	The WMS	62

4.2 Communication Address Setup 63

4.3 FinsGateway Service Manager 63

4.4 Manual and Home 65

4.5 Teaching Step Size 66

4.6 Action Mode 67

4.7 The Job-List 68

4.8 Memory Management 68

LIST OF APPENDICES

APPENDIX	ITEM	PAGE
A	INPUT/OUTPUT ASSIGNMENT OF THE RACKING SYSTEM	75
B	INPUT/OUTPUT ASSIGNMENT OF THE MHS SYSTEM	82
C	FINSGATEWAY (FGW) SETUP PROCEDURE	91

CHAPTER 1

INTRODUCTION

1.0 The Introduction

Computer Integrated Manufacturing (CIM) system is well-known as 1. Group Technology (GT), 2. Computer Aided Design and Manufacturing (CAD/CAM), 3. Flexible Manufacturing System (FMS), 4. Industrial Robot, 5. Automatic Warehouse [4]. Automated Storage and Retrieval System (AS/RS) is a computer-controlled system for depositing and retrieving goods from defined storage locations. AS/RS is importance to improve the efficiency of operation of a warehouse or a distribution centre. Automatic warehouse has drastically reduced the workforce required to run the business. Minimum labor workers are needed for tasks input via a computerized warehouse management system. These tasks include goods receiving, retrieving and dispatch processing. On goods arrival, the automation system is notified and the goods are properly identified using an identity device such as a barcode scanner or a magnetic tag. Thereby, the goods are taken by a material handling system (MHS), sortation system and automated cranes to an assigned storage location. Upon receipt of orders, the automation system is able to re-locate the goods immediately via a computer and retrieve the goods to a pick location. The automation will combine all order information and assign picked goods into dispatch units. By sortation systems and the MHS, these dispatch units are move to outgoing trailers.

Typical AS/RS involve in goods receiving together with goods identifying process, storing and retrieving, sortation system, dispatching, a warehouse management system and personnel [1]. Technically, it can be seen that the system is an integration of multiple computer-controlled automations. Each automation serves for an assigned purpose, which may vary depending on the goods and the business. In general, it is a complex design involving modular system designs and integration system designs. The technologies applied for the system will evolve as new devices are invented, such as radio frequency identification (RFID). Consequently, for engineering instructors and students, AS/RS is too complex and too business nature dependant for teaching and learning purpose.

The engineering students are not supposed to learn in operator perspective but an engineer perspective in industry design concepts and contemporary technologies. The design concepts - Flexibility, Expandability and Modularity, are stressed in this paper. Via an automatic warehouse which integrates both a supervisory level and controller level via the PLC and computer network, this system demonstrates the design concepts and technologies applied in the integration. By inventory policies and the user friendly WMS software, the integration gives the user various information on the stored/retrieved items, the item searching mode and the status of the system. The integration highlights the concept of the supervisory level , the controller level and the device level. The supervisory level provides large amount of information meaningful to human, through an user-friendly graphic interface program. Whereas the controller level defines large information critical to the controllers themselves, both the logic and the communication amongst the controllers. The device level states the bottom level of the integration on variious types of input and output devices.

1.1 Objective

This project is intended to develop a training model for the engineering students. The training model is equipped with these objectives for learning.

1 To learn design concepts that apply flexibility, expandability and modularity in the integration;

2 To understand the supervisory level that implements an industry communication protocol for networking PLCs and a host computer with a customized application software.

3 To understand the controller level that enhance systematical sequential programming methods;

4 To understand the device level that tells the students to choose a sensory device for input and an actuator for output based on the application;

1.2 Research Scope

The scope of the project includes:

1. Design network connection between the supervisory level and the controller level using Omron Compolet, Omron CX-Programmer
2. Customized Storage and Retrieval Management software using Microsoft VB.net
3. Create monitoring and control, database of WMS.
4. Programming the controller level consisting of Omron PLC, including the SRM, the material handling, the receiving station, the picking and sorting station, the labeling and packaging station.
5. Specifying the devices used in the system.

CHAPTER 2

LITERATURE REVIEW

2.0 Literature Review

In real industry world, the development of automation is fast and the technology in system evolves as new solutions are recommended in the market from time to time. Engineering education must match with the high-speed automatic development of the factory, so it may not be fall behind in manpower training [6]. The training system is designed to contain all of the automation mechanic part, the control system, and in open structure. Somehow the overall design philosophy is based on three interrelated objectives, namely flexibility, modularity and expandability. We-Min Chow [9] had stated in his paper that flexibility will not only allow a broad spectrum of application environments but is also a major contribution factor in extending application life; expandability is closely coupled with flexibility and will allow application in areas not yet defined. Finally, modification and maintenance are greatly enhanced if the system is modularized in a meaning manner. As new technologies emerge, these three objectives are still valid for all automation applications.

There are some constraints in manpower training. Firstly, the system does not reflect the technologies used in industry or the technologies lagged behind. Secondly, the system does not review the real application in industry. Thirdly, real industry application is too complex.

Computer Integrated Manufacturing (CIM) system is well known as follows: [5] [6]

1. Group Technology (GT)
2. Computer Aided Design and Manufacturing (CAD/CAM)
3. Flexible Manufacturing System (FMS)
4. Industrial Robot
5. Automatic Warehouse

Automatic warehouse is one of the major applications of the CIM. An Auto Storage and Retrieval System (AS/RS) can be defined as an automatic warehouse.AS/RS has been an essential business operation system since the introduction of CIM In general, the automatic warehouse has the functions such as receiving, material handling, storage, picking and sortation, shipping, labeling and packing, the warehouse management and personnel [1]. In real industry application, Dotoli, M.; Fanti, M.P.; Iacobellis, G. (2004) [2], have stated that:

> *"A typical AS/RS comprises several aisles with storage racks on either side, each serviced by an automated stacker crane, operating storage and retrieval of the parts.Cranes move in three directions: along the aisle to perform transfers, sideways between the aisle and the racks, and vertically to reach the Storage/Retrieval (S/R) location. Each aisle is also serviced by a storage and by a retrieval conveyor. Moreover, the AS/RS may include Rail Guided Vehicles (RGVs), transporting parts. Finally, several input (storage) and output (retrieval) buffer stations, where the RGVs load or deposit pallets, are located in the system."*

Graphically, the definition of a large scale AS/RS [2] is shown in Figure 2.1.

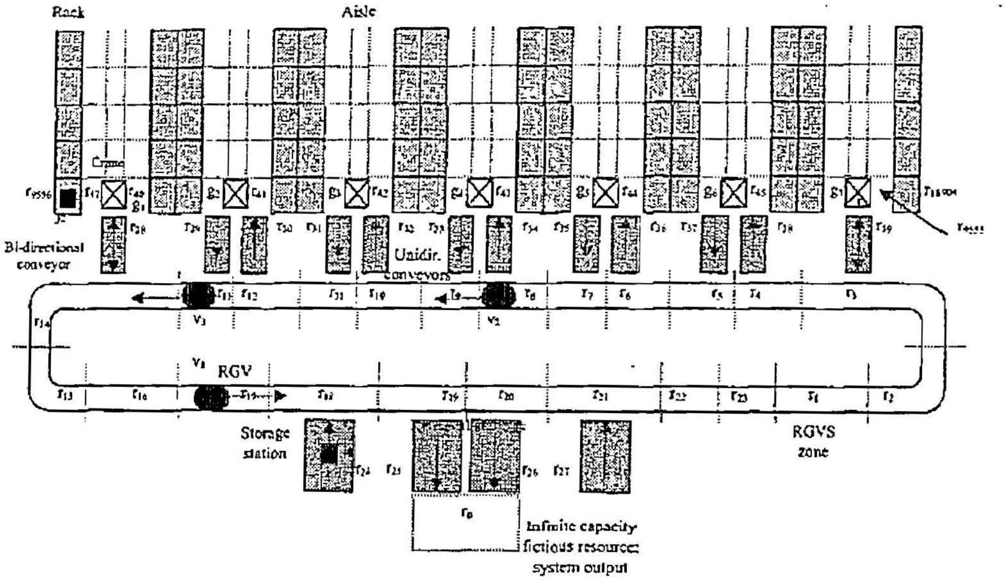


Figure 2.1: Plan layout of a multi-product AS/RS serviced by RGVs [2]

Automated warehouses represent a tremendous financial investment and play a critical role in the manufacturing and distribution process [10]. Especially in logistic business and distribution store, an AS/RS is so essential to automatically handling large amount of different items, flowing in and out according to the order, with minimum labor and human error.

According to Frazelle, E [10], to design an AS/RS, three physical configurations are to be considered carefully during design. Firstly, what is the appropriate size and shape of the warehouse? The question involves of minimizing total system cost with constraints such as storage requirement throughput. Secondly, how many input/output (IO) points should be designed into the system? The question involves the physical size of the system, which would affect the performance. The performance might be evaluated via simulation, queuing theory and statistical analysis. Thirdly, what is the appropriate material handling system to interface with the warehouse? The question involves the layout of the conveyor in a loop where trays can be delivered to workstations along the loop. The performance is affected as the length of the loop increases as the trays traverse along. Besides, the inherent loop control become complex. He did highlighted four major operation strategy design problems which need to be balanced. There are:

1. item classification
2. system balancing
3. storage location assignment
4. man-machine balancing

Beside Frazelle, E [10], there are few papers viewing AS/RS design in a whole picture rather most papers review on certain facets of the AS/RS issues. Suesut, T. and his research team [4] had investigated the purpose of inventory management to reduce the total cost of material stocks.

Serafini, P. and Ukovich, W [8], had recommended an optimum algorithm for the shortest storage and retrieval cycle time. The algorithm depends on the structure and scale of an AS/RS, and the nature of the items. Somehow, Ya-Hong Hu and his team [3] recommend pre-sorts the loads to specified locations to minimize the response time of retrieval, with a new type of AS/RS namely split-platform AS/RS. Soeman Takakuwa [7] had introduced a method of modeling large-scale AS/RS on examining storing/retrieving policy from the efficient standpoint. Seng-Yuh Liou and his team [6] had introduced an education AS/RS. The education model does not incorporate industrial package in integrating the supervisory level (computer) with the controller level (PLC) with graphical user interface (GUI). In his study, design philosophy objectives such as flexibility, expandability and modularity, are not emphasized throughout the design.

Thus, the design of an AS/RS is very business nature dependant and complex for a real industry application. Hence, it would be sensory overload if it is to teach or introduce engineering students a complex industry AS/RS in all design aspects.

There are few papers highlight the education model of AS/RS on:

- System design philosophy on flexibility, expandability and modularity
- The basic techniques in driving two axis servo motor for storing/retrieving

- The product identity data (ID) scheme which are crucial for data searching and matching purposes
- The PLC program method, namely function block programming method which increase repeatability in program and ease for debugging
- Communication protocol between a computer and a PLC, between PLCs, a PLC with a robot.
- the powerful computing capability of the PLC for the algorithms on the Storage and Retrieval Machine (SRM) using servo drives control, storage and retrieval decision scheme based on First-In-First-Out (FIFO) or Last-In-First-Out (LIFO), picking and sorting scheme, the communication and data transfer scheme to the supervisory level namely the Warehouse Management System (WMS)
- Stand-alone and simple WMS

CHAPTER 3

METHODOLOGY

3.0 Methodology

This section starts with a description of the system flow. The system flow describes the requirements in the perspective of an user, implying the technologies and methods that should be applied. It also explains what is expected the system should behave. Based on the expectations, we can design the expectation accordingly.

3.1 The System Flow

The AS/RS is based on layout configuration as shown in Figure 3.1. Once power on, the Storage/Retrieval Machine (SRM) needs to be triggered manually to an original position called "Home". Using the Warehouse Management System (WMS), the user must specify the communication address before triggering any buttons. Once the communication has established, the WMS will prompt its status in text.

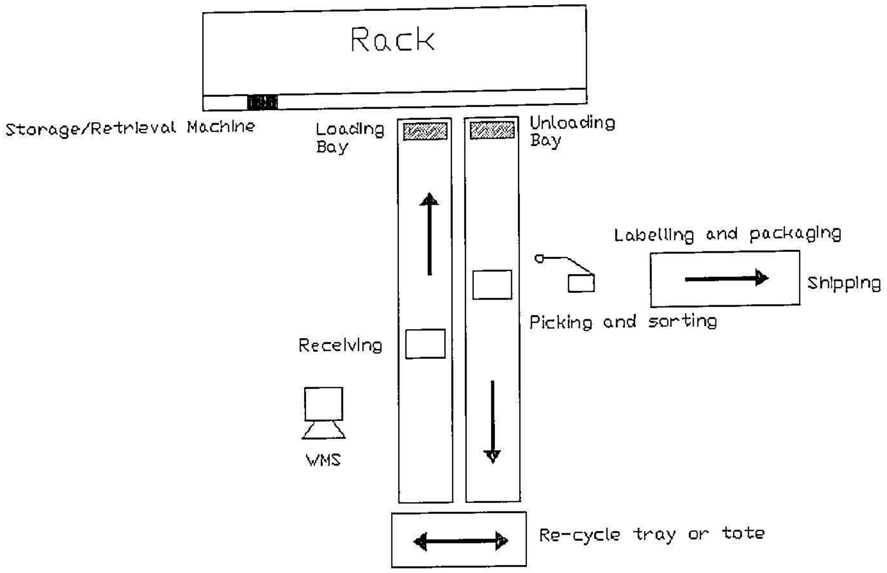


Figure 3.1: The System Layout

Once communication is setup, the user is required to trigger the SRM to an original position called "Home". Both axes consist of servo drive without any encoder feedback. Once "Home", a text message will prompt the user the status.

The WMS provides other options for critical position teaching as well. "Home" position is not equal to the coordinate (0, 0). Based on the system layout, there are three critical positions to be taught, namely coordinate (0, 0), the loading bay and the unloading bay. At teaching mode, the WMS provides options for inching step size for both X ad Z axes. The user is required to trigger the X or Z axis accordingly by selecting the step size. The user is required to monitor visually the precision of the taught positions. For example, teaching coordinate (0, 0). At near coordinate (0, 0), the user can inch X or Z axis at small step, simulating the storage/retrieval action manually for smooth action. Once the position is confirmed, the user can save the position into the controller memory.

The coordinate (0, 0) is the starting position of the rack coordinate of X-axis and Z-axis. This position is relative to a "Home" position which is defined by a proximity sensor for each axis. Other coordinate points (X, Z) are then relatively referred to coordinate (0, 0) at rack cell positioning. The loading bay and unloading bay positions are also relative to "Home" position. These three positions are critical positions to the system. Once these positions are saved, the user can still view these positions (X, Z) and modify them at any time. The taught positions will saved into the system memory. No teaching is required once the positions are satisfied. It is worthy to note that improper teaching positions would result in undesirable response at the following action mode.

The WMS also provides manual operation option for the racking system. The manual operation includes X and Z axis jogging and SRM pneumatic grippers. Every single pneumatic cylinder can be manually driven via the WMS. The WMS provides another option for action. This option provides fully automatically actions, data searching and data matching process as well as communicating with other modules. It is the desired design that the system is supposed to response. The WMS will list down all action jobs being entered by the user with all details regarding the products such as the action, the identity (ID), the sequence number and the rack cell coordinate. The user can enter the jobs without waiting previous job to be completed. Meanwhile, the user can cancel all listed jobs if mistakenly entering an action call.

With proper setup, the system would work or perform normally as described below:

1. At the Receiving station (along the material handling conveyor), an operator loads a part on a tray and enters a relevant product ID and action mode via WMS. A job is automatically generated at the WMS and command the SRM to perform accordingly.
2. Then, he/she presses a button to release the tray with predefined products to the loading bay. The tray is then release toward the loading bay. Multiple trays can be lined up in sequence.

3. If the action mode is "Storage", the job is "Storage". The SRM would fetch the tray in the loading bay and load into a pre-defined rack cell.
4. If the action mode is "Bridge", the job is 'Bridge". The SRM would fetch the tray in the loading bay and unload into the unloading bay.
5. Besides, the operator also can enter a product ID and check the action mode: "Retrieval", together with its criteria, FIFO or LIFO.
6. If the action mode is "Retrieval", the job is "Retrieval". The system would auto figure out the coordinate location based on input ID and criteria. The SRM would retrieve the tray from the rack cell and unload it onto the unloading bay.
7. At the Picking and Sorting station, the sorter robot would sort the product based on the ID passed over from the SRM during 'Retrieval" completed. If the ID is "0124", then sort the product to the right else to the left. If the ID is "0000", no sorting is needed.
8. The empty tray would auto recycle back to the Receiving station. The process repeats.

The WMS will provide a data management facility, so that the user can read or check or modify the items in the rack cells according to the coordinate assignment. The information will provides identity (ID) and the sequence of each products being stored.

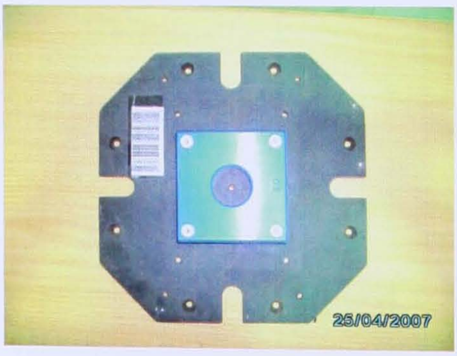
3.2 Modular Design

There are three design concepts to be achieved, namely flexibility, expandability and modularity. The system layout can be divided into multiple modules in a meaning way. Each module has its controllers and is programmable. The controllers have provided flexibility to the designer to reconfigure the logic sequence or formula without modifying tediously the wires. Besides, since the design is modular, the system can be expanded as more modules are added into. The benefit of modular design also facilitates system setup as well as maintenance because other modules will not be affected directly when any single module is setup or under maintenance.

The system is designed in module and integrated later. The system can be divided into four major modules, the Racking System and the WMS, the Material Handling System (MHS), the Sorter Robot, and lastly the Communication.

3.3 The Racking System and the WMS

The racking system is a standalone system in CIM. It is mainly to store/retrieve trays (Figure 3.2) from the rack. Each tray carries a product which is coded, and recorded by the WMS. The SRM is a two-axis servo driven mechanical machine with a multi-cylinder end-effectors. It moves at X and Z directions so that the end effectors can store/retrieve trays as commanded by the user. The racking system controller will communicate with other controllers and a host computer at the integration stage.



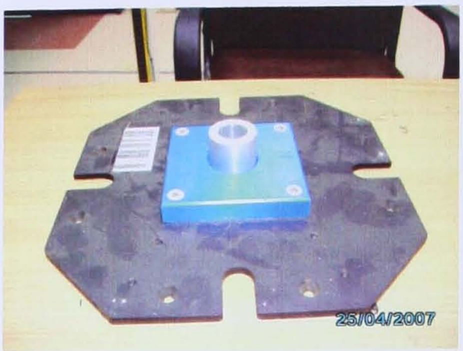


Figure 3.2: The tray, the product and the product in the tray

The racking system consists of a rack of eight rows times' nine columns as Figure 3.3 shown. Each cell is 283mm in width and 151mm in height except row3 to row4 where the height is 228mm.

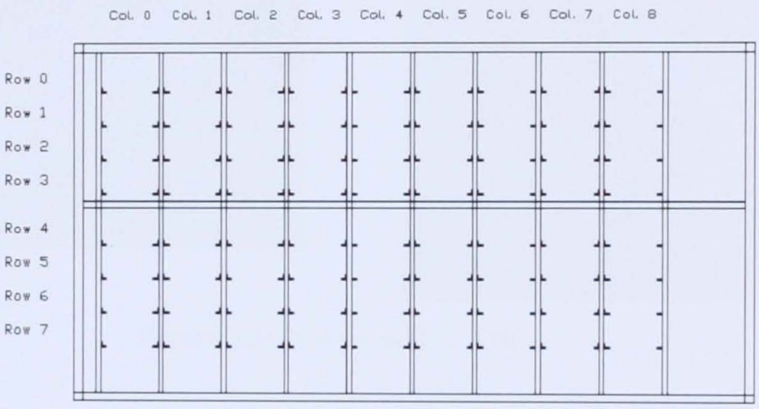


Figure 3.3: The Rack in row and column

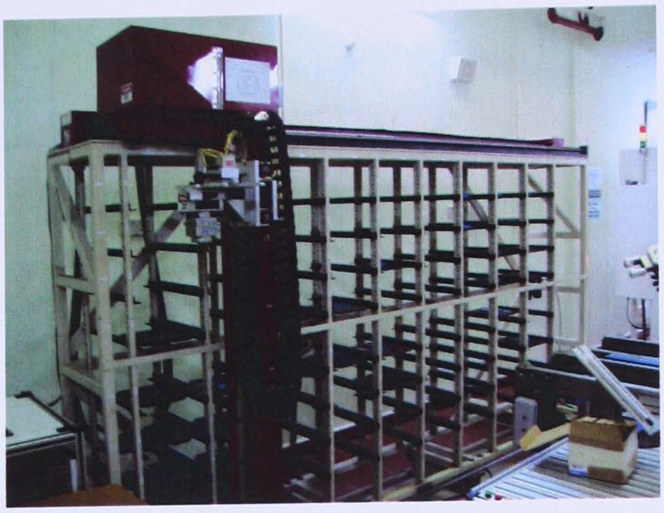


Figure 3.4: The Racking and SRM

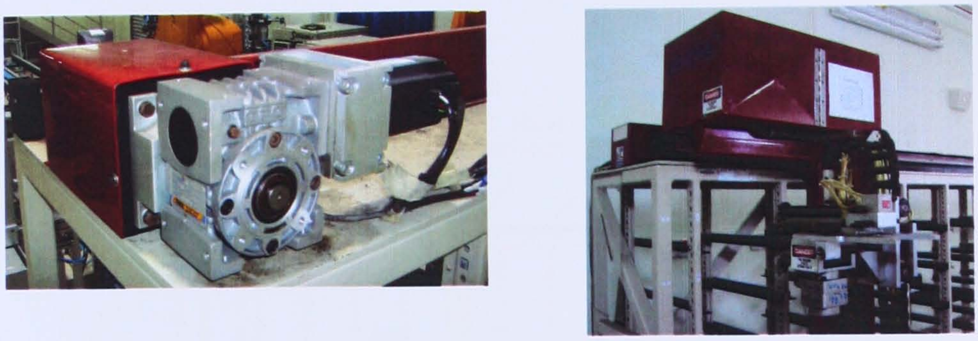


Figure 3.5: X-Axis and Z-Axis

In the view of the device level, as Figure 3.4 and Figure 3.5 shown, the SRM has two axes of movement, namely X-Axis and Z-Axis and a set of pneumatic end-effectors for storing/retrieving. Both axes are driven by a servomotor to achieve position exactly as command. The pneumatic end-effectors will work at two directions. One direction is facing to the rack for storing/retrieving; other direction is facing the material handling conveyor for loading/unloading. At both directions, a pneumatic gripper will be triggered to securely grip the tray (Figure 3.6).

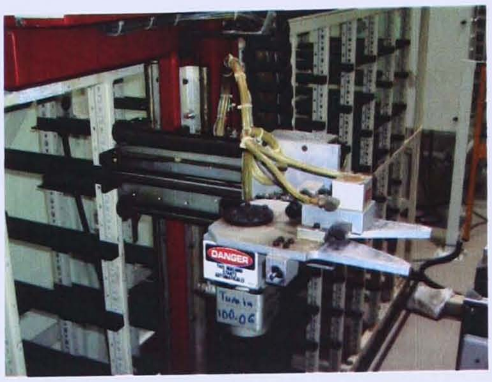


Figure 3.6: The SRM pneumatic end-effectors for storing/retrieving

The SRM multiple cylinders end-effectors contain two linear cylinders and a rotary cylinder. The linear cylinders is named SRM_In (SRM_Out) and SRM_Grip; while the rotary cylinder is named SRM_Turn In (SRM_Turn Out).

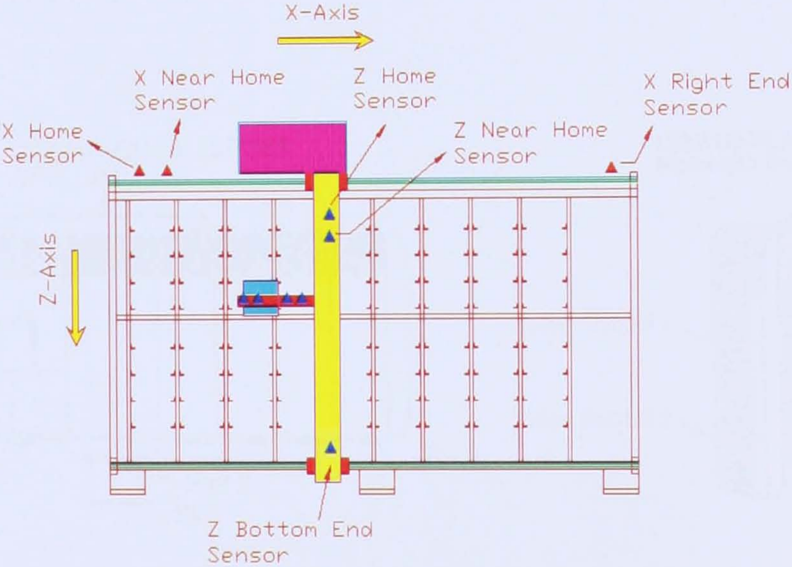


Figure 3.7: The SRM: Home position and (X, Z) direction

Figure 3.7 shows the X and Z axis sensor locations. The "Home" position is defined at left of X axis and at top of Z axis. The "Near Home" sensor would slow down the X and Z drives as the drives are approaching toward "Home" sensor. The "X Right End" sensor and "Z Bottom End" sensor prevent the drives to overrun.

Both axis of the SRM do not use rotary increment encoders to feedback the position rather the control of the position is of semi-closed type. By default, the servomotor is built in with an encoder which is feedback to the servo amplifier. The PLC does not read the encoder feedback from the servomotor. The servo amplifier would step the servomotor according to the number of pulses output by the programmable logic controller (PLC). Once at stop (power is ON), the servomotor is locked at both clockwise and anti-clockwise direction, providing an absolute position hold. In the view of the controller level, the PLC is CQM1H-CPU51 of Omron, with an inner board of pulse I/O that supports two pulse inputs and two open collector type outputs. The pulse I/O board is type CQM1H-PLB21 (Figure 3.8). The pulse output can output at maximum 50 KHz. Figure 3.9 shows the wiring example of a pulse output to a servo driver. The details of Pulse I/O board can be found in instruction manual of "Sysmac CQM1H Series" [12] as well as the PLC instruction to output pulses [13].

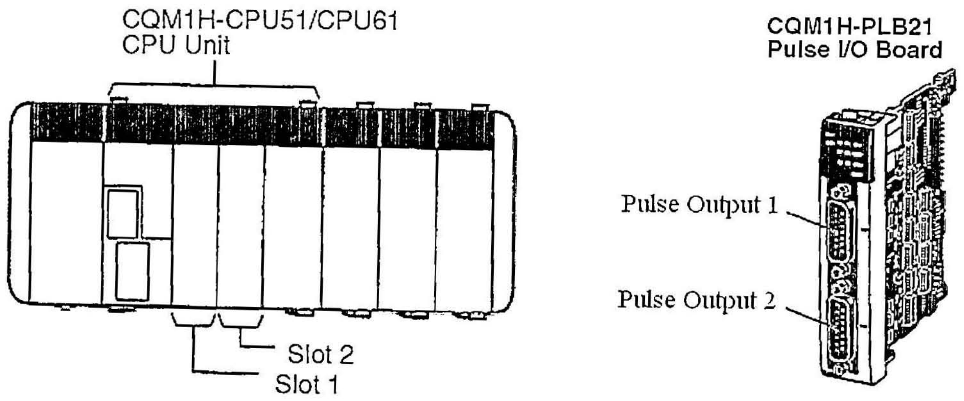


Figure 3.8: The PLC and Pulse IO Board [12]

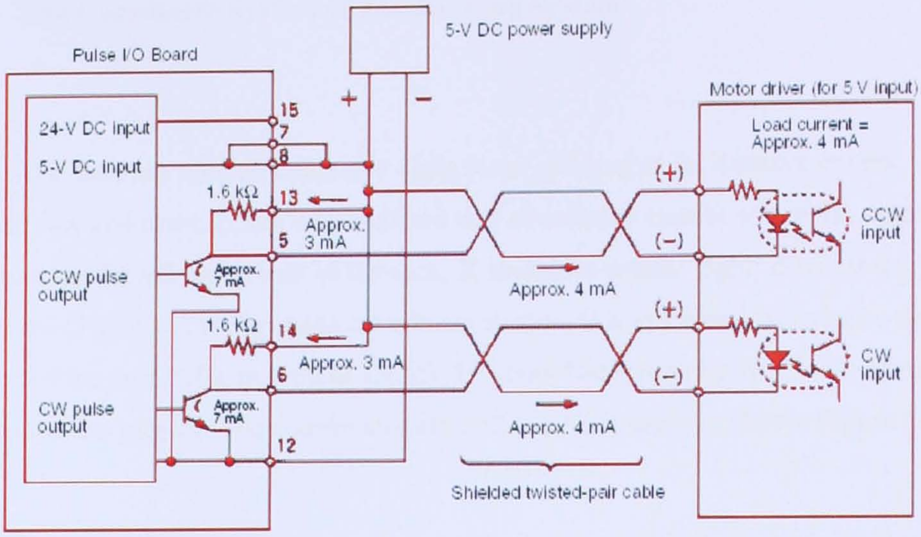


Figure 3.9: Pulse Output Wiring Diagram [12]

The X and Z servomotors are of Mitsubishi, type: HC-PQ43 with servo amplifier MR-C40A. The detail of the servo amplifier settings can be found in “Servo Motors and Servo Amplifiers Instruction Manual – MR-C” [15]. The racking system control panel is shown in Figure 3.10. More details of the IO assignment of the PLC of the racking system can be found in Appendix A.

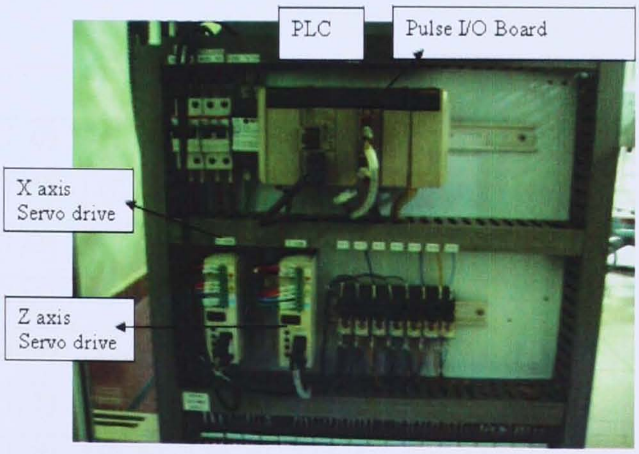


Figure 3.10: The Racking System Panel

3.3.1 The Coordinate System of The Racking System

The rack has nine columns and eight rows, pitching at fix distance except between row and row4. It can be visualized as a coordinate system where the origin start from the far left top corner of the rack; X increases toward right; Z increases downward (Figure 3.11). By using coordinate system, it is systematical to name the position of the rack cells, in term of (X, Z). The coordinate naming will be consistent through out the project. The coordinates are collected in a table as shown Figure 3.12.

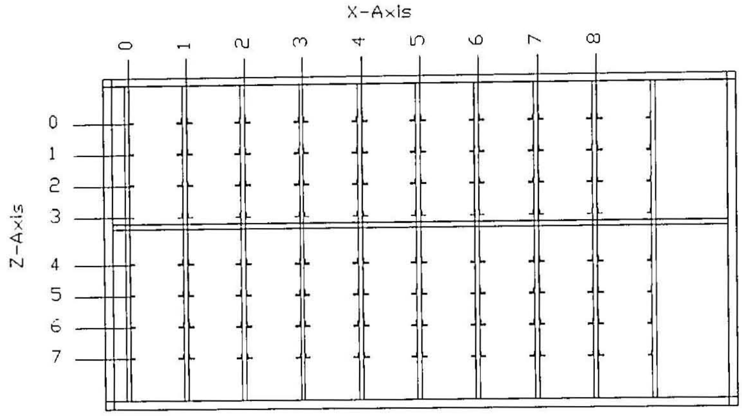


Figure 3.11: The Rack Coordinate System: (X, Z)

	X →								
	(0, 0)	(1, 0)	(2, 0)	(3, 0)	(4, 0)	(5, 0)	(6, 0)	(7, 0)	(8, 0)
	(0, 1)	(1, 1)	(2, 1)	(3, 1)	(4, 1)	(5, 1)	(6, 1)	(7, 1)	(8, 1)
	(0, 2)	(1, 2)	(2, 2)	(3, 2)	(4, 2)	(5, 2)	(6, 2)	(7, 2)	(8, 2)
	(0, 3)	(1, 3)	(2, 3)	(3, 3)	(4, 3)	(5, 3)	(6, 3)	(7, 3)	(8, 3)
Z	(0, 4)	(1, 4)	(2, 4)	(3, 4)	(4, 4)	(5, 4)	(6, 4)	(7, 4)	(8, 4)
↓	(0, 5)	(1, 5)	(2, 5)	(3, 5)	(4, 5)	(5, 5)	(6, 5)	(7, 5)	(8, 5)
	(0, 6)	(1, 6)	(2, 6)	(3, 6)	(4, 6)	(5, 6)	(6, 6)	(7, 6)	(8, 6)
	(0, 7)	(1, 7)	(2, 7)	(3, 7)	(4, 7)	(5, 7)	(6, 7)	(7, 7)	(8, 7)

Figure 3.12: The Coordinate System Table

Since the servomotor will step according to the number of pulses, the coordinate system must be converted into pulse number format as to locate the (X, Z) position. There is a need to find out how many pulses to move X or Z 1mm linearly or what is the measurement in mm for one pulse. Studies of both X and Z drive mechanical structure is necessary as well as the characteristic of the servomotor.

Both X and Z axis dynamic drives are studied as shown. It is worth to note that both X and Z axis use timing belt for linear movement; the revolution movement of the servomotors are changed to linear movement via gear and timing pulleys, with pre-defined ratio. Figure 3.14 and Figure 3.14 show both Z and X axis drives structure. Figure 3.15 shows the side view of a timing belt. Both X and Z axes use the same type of timing belt.The servo drive and servo motor is of Mitsubishi as listed below:

Servo Amplifier	MR-C40A
Servo Motor	HC-PQ43

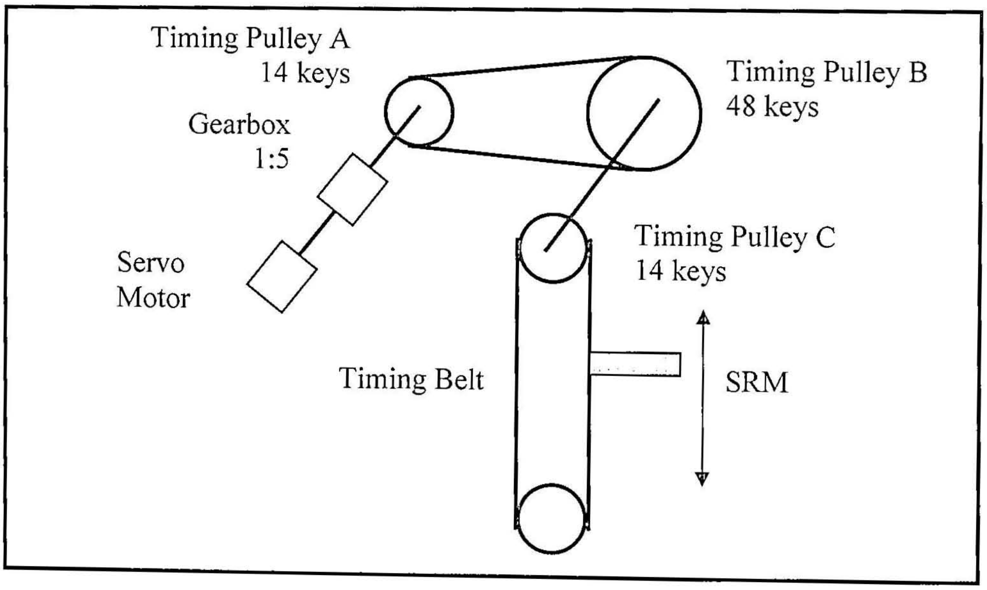


Figure 3.13: Z-axis Drive and Gear Ratio

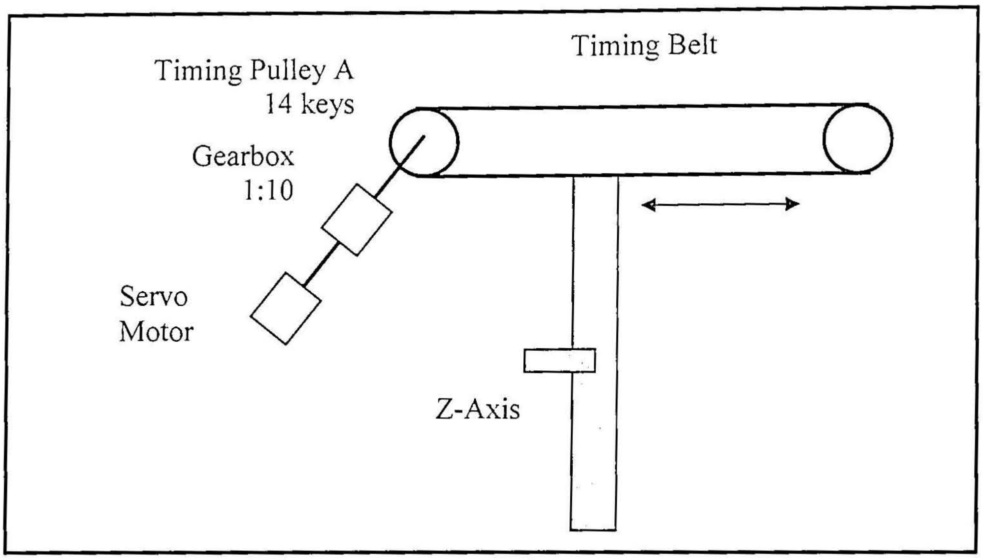


Figure 3.14: X-Axis Drive ad Gear Ratio

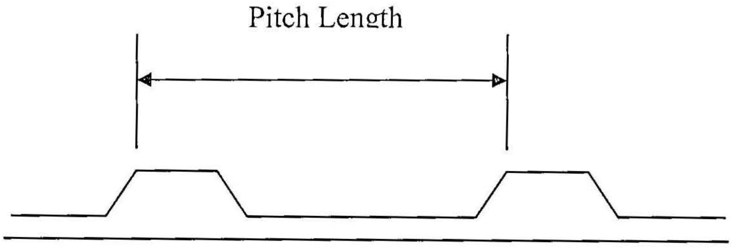


Figure 3.15: The Timing Belt

Via the relationship of the gear, we can calculate the linear movement in term of millimeters if a certain amount of pulses is fed into the servomotor. Before we actually drive the servomotor, there is a key function of the servo amplifier that would affect the precision. The function is the electronic gear, as shown in Figure 3.16.

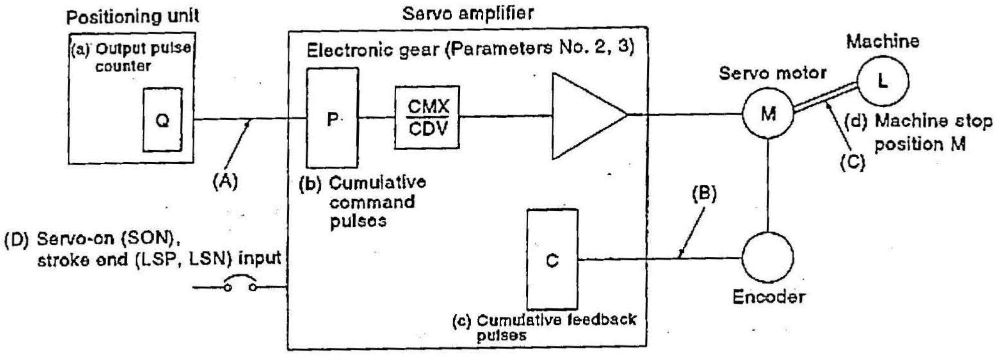

Positioning unit
(a) Output pulse counter
Q
(A)
(D) Servo-on (SON), stroke end (LSP, LSN) input
Servo amplifier
Electronic gear (Parameters No. 2, 3)
P
CMX
CDV
(b) Cumulative command pulses
C
(c) Cumulative feedback pulses
(B)
Servo motor
M
Encoder
Machine
L
(d) Machine stop position M
(C)

Figure 3.16: Electronic Gear [15]

By default, the electronic gear $\frac{CMX}{CDV}$ is at ratio 1:1 and the servomotor would revolute 360° if 10000 pulses are fed into the servo amplifier. If $\frac{CMX}{CDV}=\frac{2}{1}$, then the servo amplifier needs only 5000 pulses to drive the servo motor for 360°. If $\frac{CMX}{CDV}=\frac{1}{2}$, then the servo amplifier needs 20000 pulses to drive the servo motor for 360^0. It is a linear form of relationship. In that sense, the electronic gear amplifies or reduces the pulse number. Somehow, if the electronic gear is set above 1, the servomotor is said to lose its precision. While, if the electronic gear is set below 1, the servomotor would take more pulses to make a 360° revolution. This would take more time to complete a revolution. Through out the project, both electronic gear settings for X and Z axis are set at 1:1, which means 10000 pulses to make a 360° revolution.

Based on the mechanical drive structure and the servo amplifier characteristic, we need to formulate a mathematical relationship so that we can feed in a certain amount of pulses to the servo amplifier and to move X and Z-axis linearly as planned. It is a linear formula since all the gear ratio and the electronic gear all in linear form. X is pitching in 283mm while Z in 151mm. Hence, it is necessary to find out the number of pulses to drive X axis so that it will move linearly 283mm. So does the Z axis. After that, the rack coordinate system can be converted in pulses number format, of which is necessary in the application.

There is no detail mechanical design specification, regarding the rack dimension and the drive system. Thus, the drive precision per rotation cannot be calculated through the mechanical drive ratio. It is necessary to map the coordinate system into a coordinate system made-up of servomotor pulses number, both X and Z axis. The servomotor has the two main characters. Firstly, it can revolve as according to the number of pulses. Secondly, it always maintains at the specified position once stop. These characteristic can be used to formulate a linear mathematical equation. The result is retrieved through an experiment. The experiment is performed to retrieve the pulses precision. By default, the servomotor is capable of revolving 360^0 per 10000 pulses. This experiment does not need to consider the gear ratio of the mechanical structure. Both X and Z servomotors are fed with 40000 pulses separately, making four proper revolution. The traveling distance of the SRM in X and Z axis are measured. Pulse frequency is not important.
The experiment results show that:

Z-axis:	pulses : mm = 40000 : 104
X-Axis:	pulses : mm = 40000 : 177.875

Finally, we can conclude the experiment results into a mathematic equation for each axis, using the relation as below.

$$pulses\_numbers = \frac{Desire\_Length(mm)}{Length\_after\_40000\_pulses} \times 40000$$

The Z-axis: $$P\_Z = \frac{L\_Z}{104} \times 40000$$

The X-axis: $$P\_X = \frac{L\_X}{177.875} \times 40000$$

In X axis, the rack cell is pitched at 283mm which is equal to 63640 pulses. In Z axis, the rack cell is pitched at 151mm which is equal to 58077 pulses. Figure 3.17 shows the coordinate system in term of pulse number.

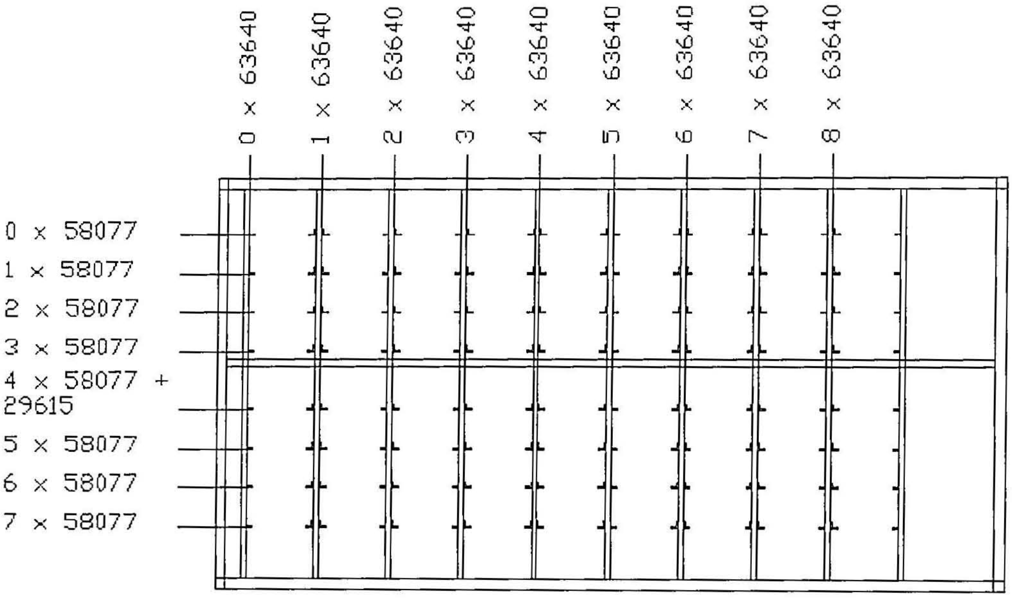


Figure 3.17: The Coordinate System in pulses number

3.3.2 Software Planning of The Racking System

This section explains the structure of the racking system software and functions. A host computer is communicating with the PLC. The host computer displays graphic user interface (GUI) which acts as the function of the racking system local control as well as a simple WMS. For the purpose of local system control, the GUI has the following functions, namely Manual Operation, "Home" Operation, Coordinate Teaching Operation and Text Status.

The Manual Operation will provide a GUI for command the SRM when the operation mode is manual. It includes individually commanding X and Z axis and the multiple cylinder end-effectors. An operation selection must be checked. Each single output can be triggered manually through the buttons designed in GUI. Logic interlocking is necessary to prevent disastrous collisions. The "Home" operation allows the user to call the SRM to a "Home" position upon power on or reset. A "Home" selection must be checked.

Coordinate Teaching Operation gives the user the convenience to teach three critical positions, namely Coordinate (0, 0), the loading bay and the unloading bay. Coordinate (0, 0) defines the starting point of the rack coordinate system; the loading bay specifies the location of the loading bay as referred to the "Home" position; as does the unloading bay. After taught, these positions will be saved in PLC memories.

The text status always gives useful information about the operation mode checked, the SRM status and the current positions of both X and Z axis in pulses number. All these operations provide a local control of the racking system.

Meanwhile, the GUI also performs as a simple WMS with the following functions:

1. Action Mode: "Bridge", "Storage", "Retrieval" with First-In-First-Out (FIFO) or Last-In-First-Out (LIFO) criteria selection.
2. Job lists
3. Memory monitoring and management

The Action Mode lists out the actions provided by the system. It is valid after the "Home" position has been performed. Throughout the project, simplicity is stressed. The user needs to check minimum selection with some help messages. In the Action Mode, three things need to be performed by a user; check an action mode; check an ID; click to confirm. This is a design concept that requires minimum skill requirement for the operator level.

The Action Mode has three selections. The 'Bridge" is defined as: fetching a tray at the loading bay and dropping it onto the unloading bay. This mode is useful if the user needs to send an item directly from the receiving station to the shipping station without the process of storing/retrieving. The "Storage" is defined as: fetching a tray with a pre-defined product and storing it into a pre-defined rack cell. While the "Retrieval" is defined as: fetching out a tray with a pre-defined product and dropping it onto the unloading bay. The "Retrieval" is accompanied with two data searching criteria, FIFO and LIFO. This searching scheme is based on two layer data search. There are totally 72 memories allocation for each layer (Figure 3.18) because the rack is nine columns and eight rows, a total number of 72 cells.

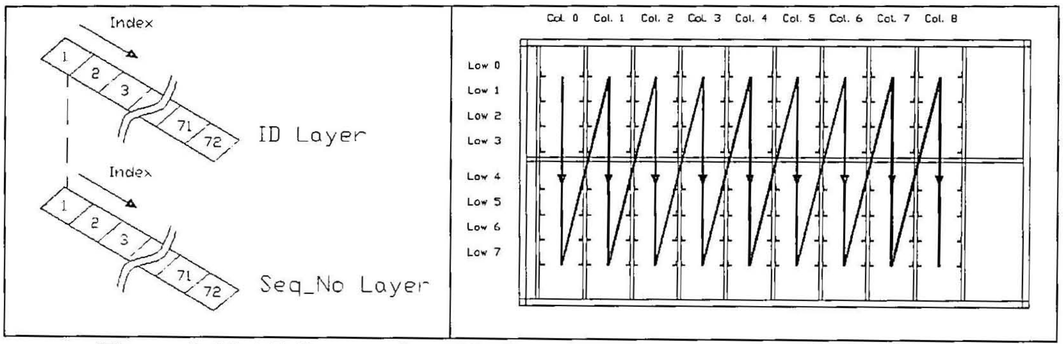
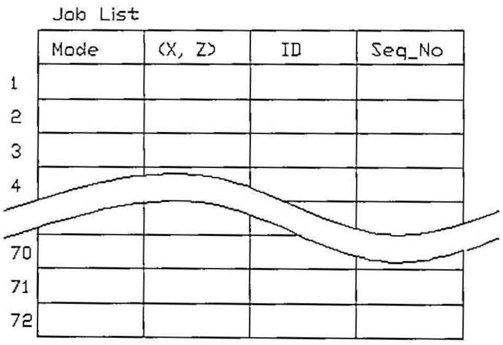


Figure 3.18: ID Layer , Seq_No Layer and the data search sequence

When the criterion is FIFO, the data searching scheme would search for matched ID at the ID layer with the current Seq_No equal to 73. The searching sequence is matched to the rack cells as indicated in Figure 3.18. The index would increment once at a time until 72. Once the ID is matched, the scheme would compare the Seq_No at the index number; if the Seq_No in the Seq_No layer is less than current Seq_No, then the index number is memorized and the current Seq_No is updated, equal to the compared one; if the Seq_No is bigger than the current one, then the ID searching proceeds. If the ID does not match at the end of the search, then it can be concluded that no such ID in the memory list. In short, FIFO would match the ID and look for the lowest score of the sequence number which means the first item of the same ID. The index number which matches both the ID matching and the lowest score of the Seq_No, is the rack cell locations. The index number is in the BCD format. It is divided by eight; the quotient is the X coordinate while the remainder is the Z coordinate. For example, if the index number is 10. If divided by 8, the quotient is 1 while the remainder is 2. The coordinate is then grouped as (1, 2). The result can be compared visually.

LIFO applies the same searching scheme expect LIFO would match the ID and look for the highest score of the sequence number which means the last item of the same ID.

The Job Lists is actually a multiple data memory stack (Figure 3.19). There are a total of 72 stack allocations reserved for the tasks. The SRM uses First-Come-First-Serve principle to perform the job. The first stack of the Job-List is performed. When the SRM has completed its job, the Job List would shift one stack upward; the first stack is over-written by its subsequent stacks.



Job List

	Mode	(X, Z)	ID	Seq_No
1				
2				
3				
4				
70				
71				
72				

Figure 3.19: Job-List

The GUI provides the convenience for memory monitoring and management. Figure 3.20 shows the layout of the GUI. The layout is matching with the rack cells; each cell is accompanied by ID and Seq_No. All ID and Seq_No can be monitored and even be manually modified by a user. No password or priority level is required.

As a standalone module, the racking system is capable of performing the functions and jobs as listed above. Meanwhile, the racking system is still a module of the CIM. It is designed to communicate with others modules using an industry communication protocol, the serial communication.

The GUI is designed using Microsoft Visual Basic.Net for its capability in graphical user interface design. FinsGateway (FGW) and Compolet of Omron (Figure 3.21) are used to provide a background communication platform between a host computer and a PLC. FGW provides a hidden background communication platform while the Compolet can be easily inserted as a component in Microsoft Visual Basic.Net (MSVB.Net).

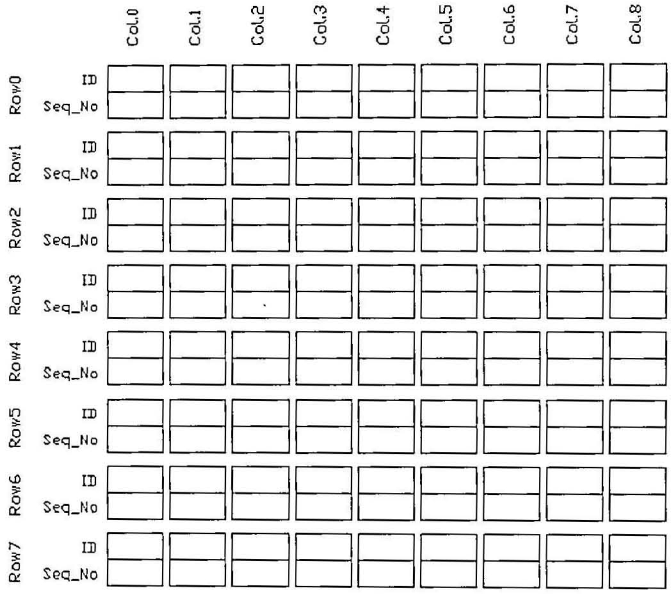


Figure 3.20: Memory Monitoring and Management

A programmer can easily apply the Compolet as an object in MSVB.net, without much knowledge of the communication format and protocol between PLC. Any type of PLC data can be read/written using Compolet in MSVB.Net. FWG supports various communication protocols. The project uses a serial communication protocol. Using the FGW together with MSVB.net, we can create a user-friendly applet that provides the features as stated above as well as the WMS. A serial communication is used because it is the cheapest yet widely used in the industry. Refer to Appendix C for the procedure to setup FGW and test with a sample applet.

All the sequential and logical jobs stated above are performed by the PLC. The host computer does provide a graphical user interface, which is called the supervisory level. PLC is the controller level. For the racking system, the PLC is assigned to perform the tasks such as Manual, Home, Coordinate teaching, Status,Action Mode, Job lists and Memory monitoring and management.

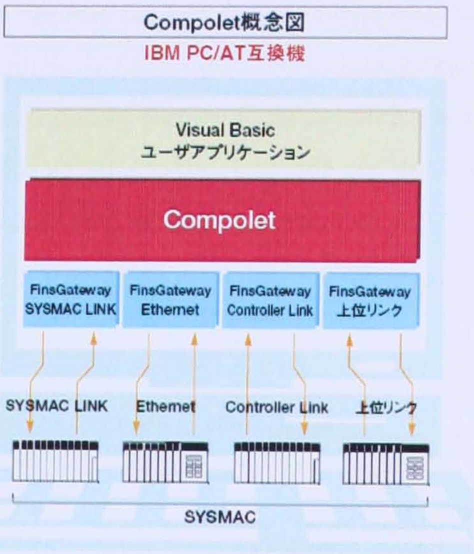


Figure 3.21: FinsGateway of Omron

Figure 3.22 shows the flow chart of the Manual mode. When the system is in Home mode, then it cannot be in Manual mode and perform the manual operation for X and Z axis (Figure 3.23 and Figure 3.24).

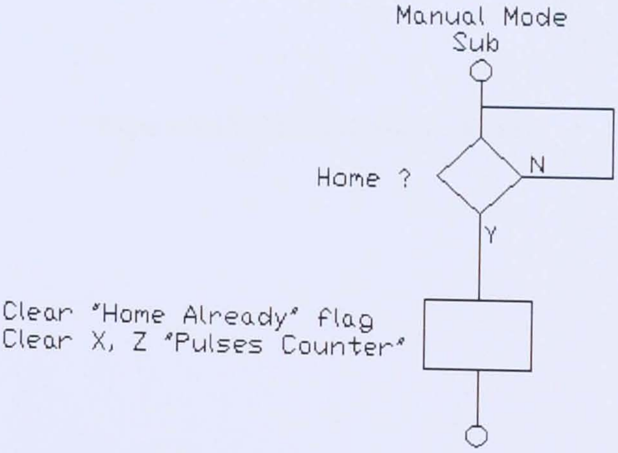


Figure 3.22: Manual Mode

When in Manual Mode, the servomotors are operating in "Continuous Mode"(Figure 3.23 and Figure 3.24) where pulse numbers are not recorded for positioning purpose. The operations for X-axis (Left or Right) and Z-axis (Up or Down) are manually triggered via buttons designed in the WMS. In Manual Mode, no complex logic is required but some logic interlocking to prevent confusion. For example X-axis, when a Left button is clicked, the SRM would move to left until Stop button is clicked. Whenever the Left action is in action, clicking a Right button would not stop nor trigger the SRM to right. For both axis, the action would stop if the end limit sensor is on, meaning that the SRM is moving toward the limit. For example, when the SRM is moving to the right side and the X Right End sensor is on, the X-axis servomotor would stop the action immediately.

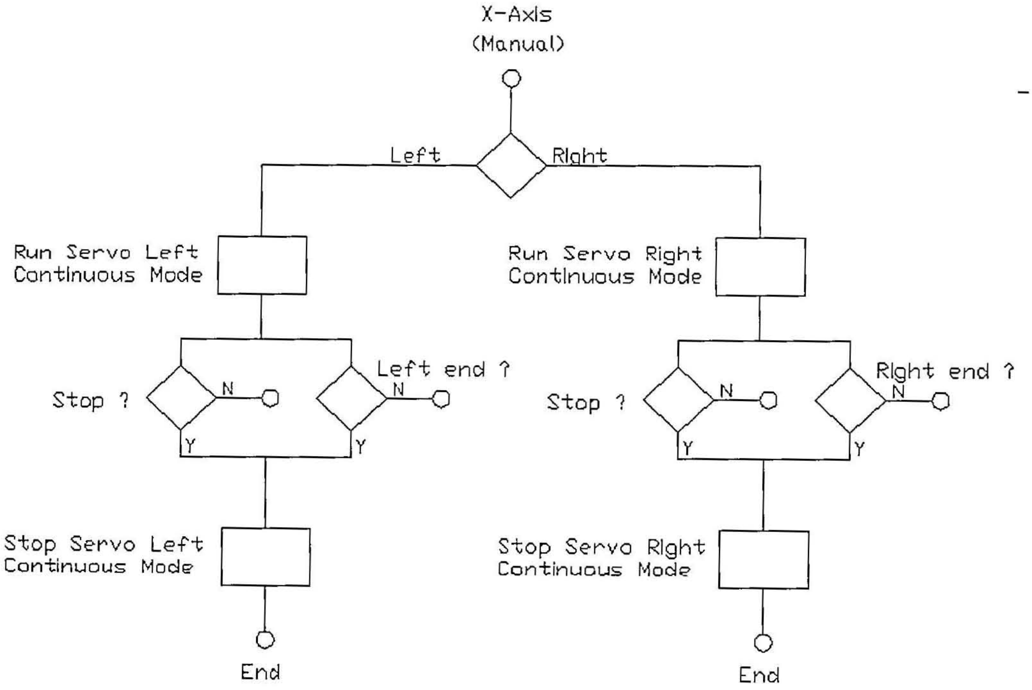

X-Axis
(Manual)
Left
Right
Run Servo Left Continuous Mode
Run Servo Right Continuous Mode
Stop ?
N
Y
Left end ?
N
Y
Stop ?
N
Y
Right end ?
N
Y
Stop Servo Left Continuous Mode
Stop Servo Right Continuous Mode
End
End

Figure 3.23: Manual Mode: X-Axis

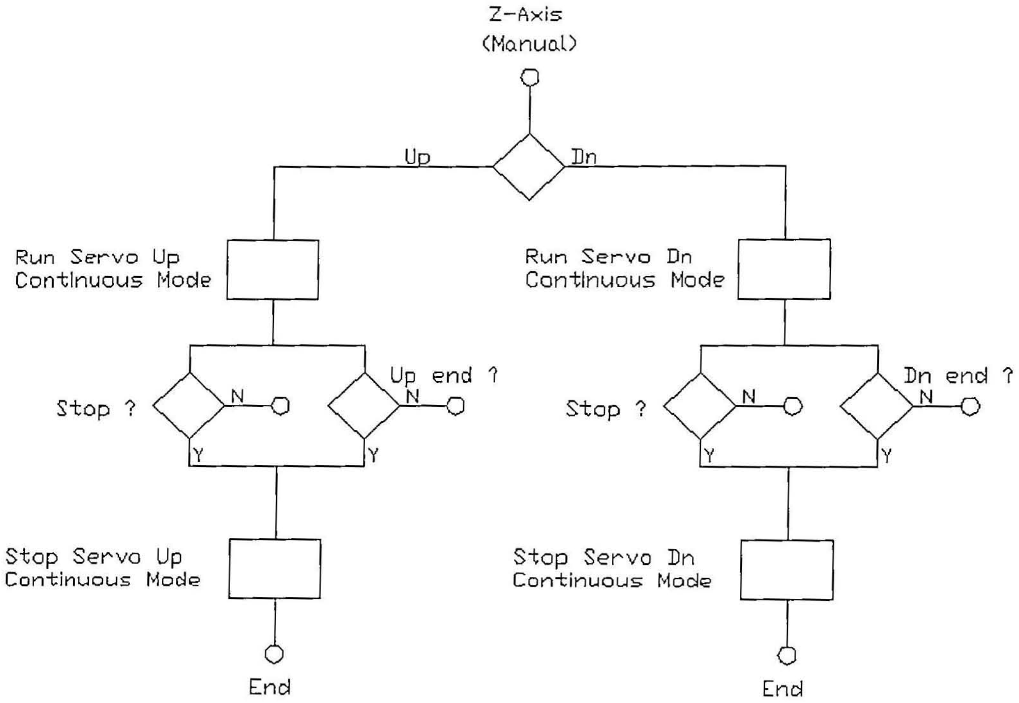


Figure 3.24: Manual Mode: Z-Axis

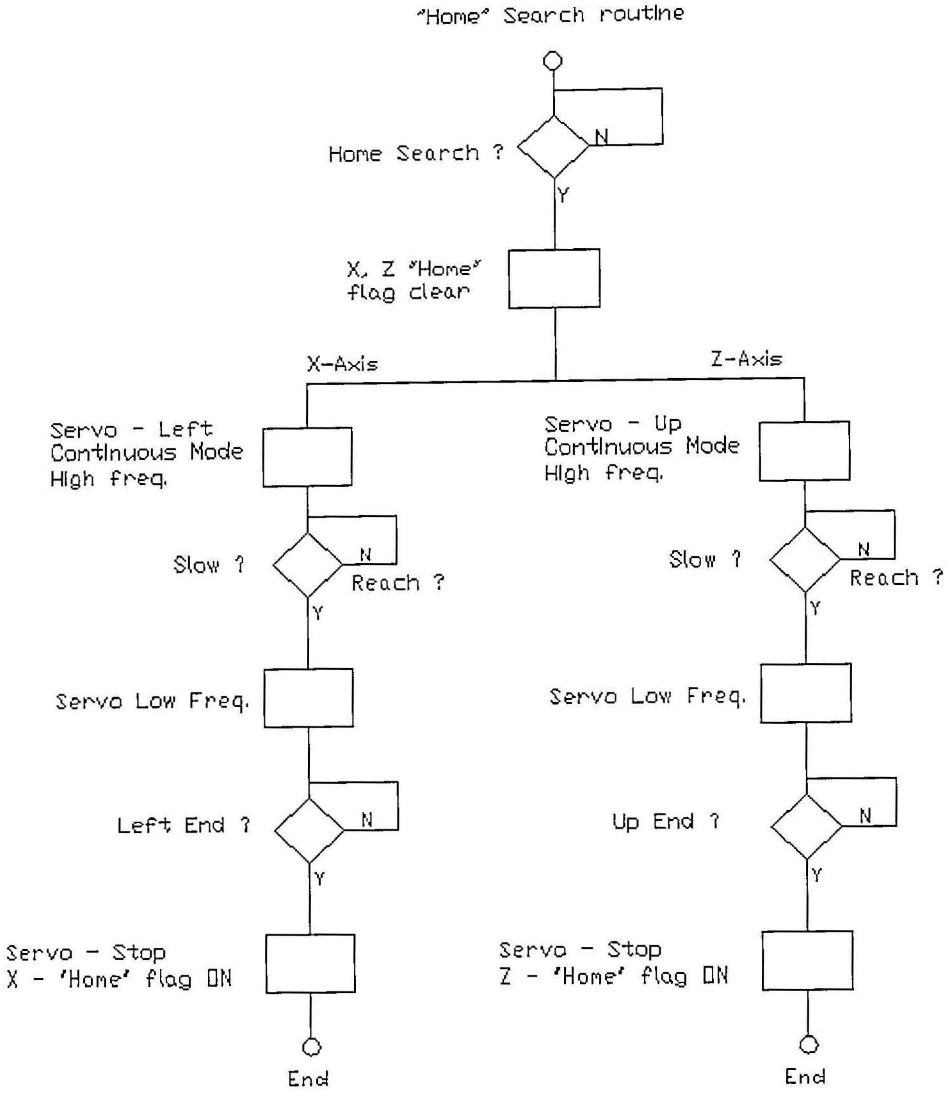

"Home" Search routine
Home Search ?
N
Y
X, Z "Home" flag clear
X-Axis
Z-Axis
Servo - Left Continuous Mode High freq.
Servo - Up Continuous Mode High freq.
Slow ?
N
Reach ?
Y
Servo Low Freq.
Left End ?
Up End ?
Servo - Stop X - 'Home' flag ON
Servo - Stop Z - 'Home' flag ON
End
End

Figure 3.25: Home Mode

"Home" is defined as left of X-axis and top of Z-axis by X Home sensor and Z Home sensor. While approaching toward these sensors, X Near Home sensor and Z Near Home sensor would reduce the servomotor speed. Figure 3.25 shows the flow chart of "Home" sequence for both X and Z axis. Both servomotors are operating in "Continuous Mode" and do not needs to specify any pulse number to move. Refer to "Sysmac CQM1H Series: Programming Manual" [13] for detail on pulse output command. Once at "Home" position, the absolute position counters, X_abs and Z_abs are reset to zero value. The absolute position counters is crucial to provide an absolute reference reading for the (X, Z) coordinate.

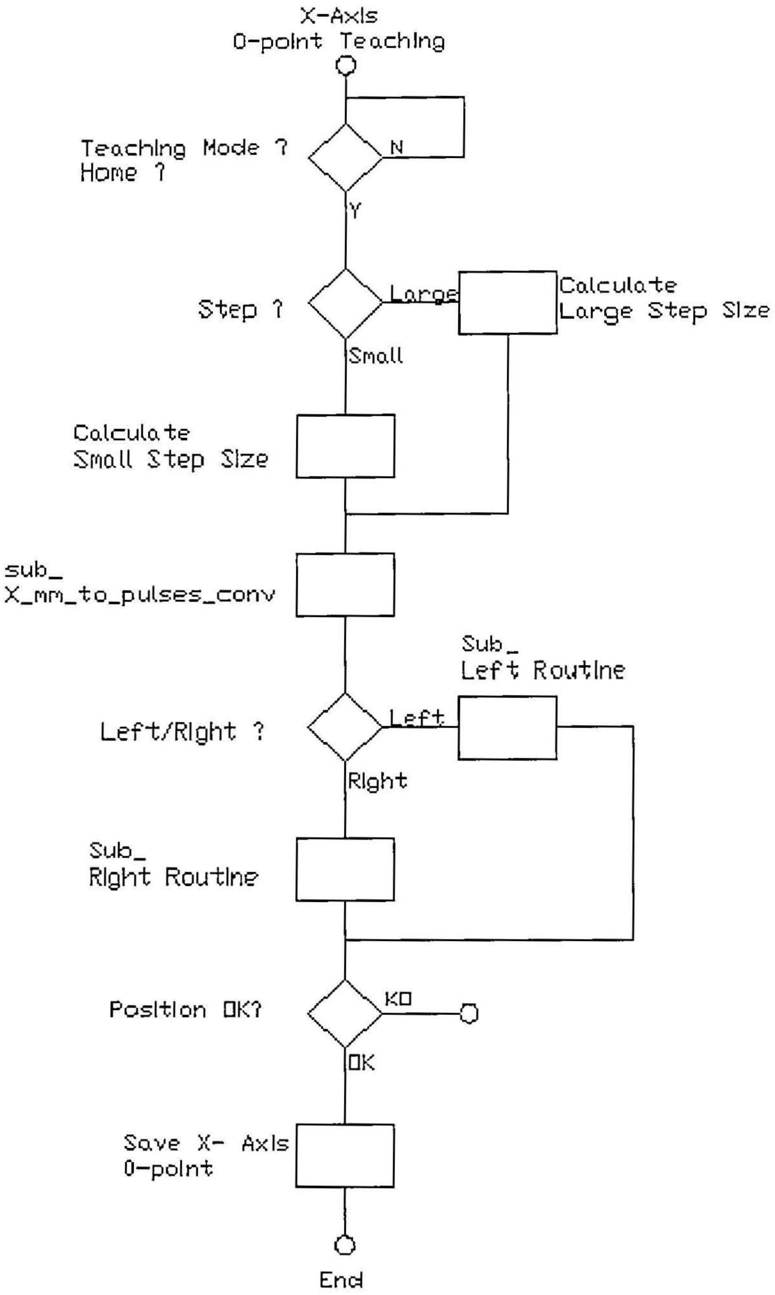


Figure 3.26: Coordinate Teaching: X-axis

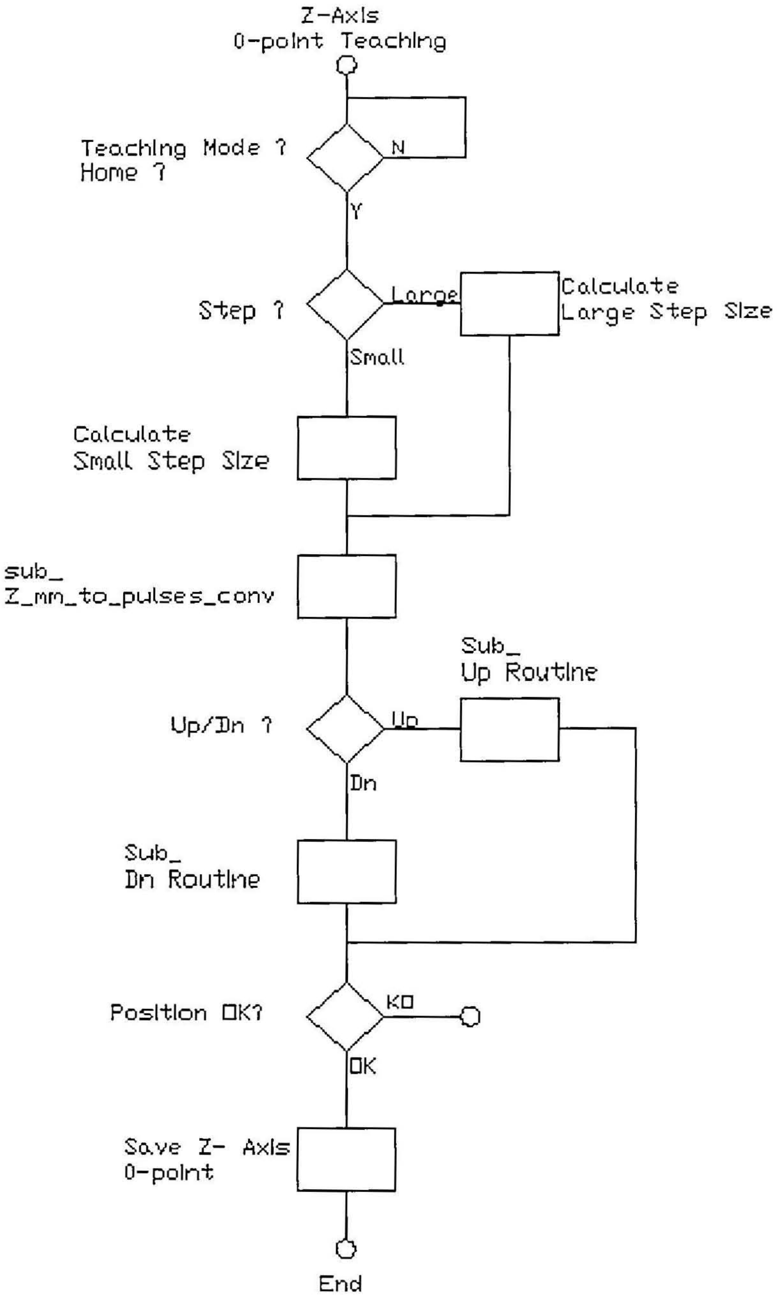


Figure 3.27: Coordinate Teaching: Z-Axis

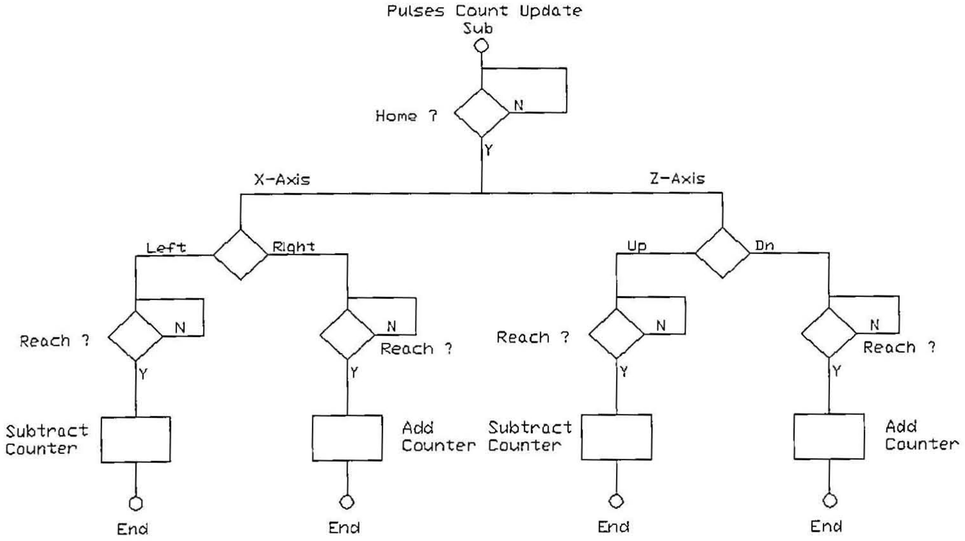


Figure 3.28: Absolute Position Counter, X_abs and Z_abs

Coordinate teaching consists of three critical positions, namely the Coordinate (0, 0), the loading bay and the unloading bay. In common, all these positions involve in X and Z axis movement in terms of pulses number. Figure 3.26 and Figure 3.27 show the flow logic of coordinate teaching for both the X and Z axis. There are two conditions before coordinate teaching can be performed. Firstly, the SRM must already achieve the status of "Home". Logically, all positions taught are relatively referred to the "Home" position. No coordinate teaching is allowed if not "Home". Secondly, the Teaching Mode must be checked with a choice amongst "Coordinate (0, 0)", "The Loading Bay" and "The Unloading Bay". It is worth to note that the pulse output command is in "Independent Mode", which means the number of pulses needs to be specified. There will be three step sizes for the coordinate teaching purpose. They are Small (1mm), Medium (10mm) and Large (50mm). These step sizes are automatically converted into specified pulses number internally once the conversion routine is called. For X-axis, the WMS will display Left and Right buttons while for Z-axis, the Up and Down buttons. It is designed that clicking the buttons once, the SRM would step at the specified step size. Once the step movement is completed, all the movements X and Z would be automatically recorded in the absolute position counter, X_abs and Z_abs as shown in Figure 3.28. For X-axis, movement to the right would add the counter once per movement; for Z-axis, movement to down would add its counter once per movement. To better enhance the teaching feature, the SRM multiple cylinder end-effectors can be triggered to act individually by clicking buttons in the WMS. A "Save Position" is designed in the WMS to save corresponding position X and Z coordinate position in term of pulse number into the allocated PLC memories. The taught positions can be reviewed later if not satisfied. These taught data are displayed in the WMS as well.

The PLC program is designed in modules form. Single functions and actions are grouped as a subroutine. Figure 3.27 and Figure 3.28 show that multiples subroutines are called for the purpose of coordinate teaching such as the step size conversion and the up/down routine. The absolute position counter is classified as a subroutine. There are a few advantages of designing the program into modules form. Via the concept 'Divide and Conquer", a complex solution can be divided into smaller portions; the small portions can be even sub-divided if too complex. It is much easier to handle a small portion of solution at a time and easier for debugging. The sub-routine can be called and used repetitively. The status message can tap at any part of the program. These status messages include "Home", "Home Already", "Not Home", "Manual", "Coordinate Teaching" and etc.

For "Action Mode", it is designed to have three main function blocks, namely the SRM action function block, the Job-List function block, and the Memory function block. Each block contains multiples subroutines which will be described in detail.The function block is a concept to group a complex task under a block, which can be used later. The left side of the block is the input port while the right side is the output port. The inputs and outputs ports can be both memory registers and just bit flags. The function block also increases repeatability as the inputs and outputs are assignment with difference memory register address and bit address. Virtually, the function block is a standalone function. Figure 3.29 shows the function blocks designed for the Action Mode for the project. There are SRM Action Block, Job-List Block and Memory Block.

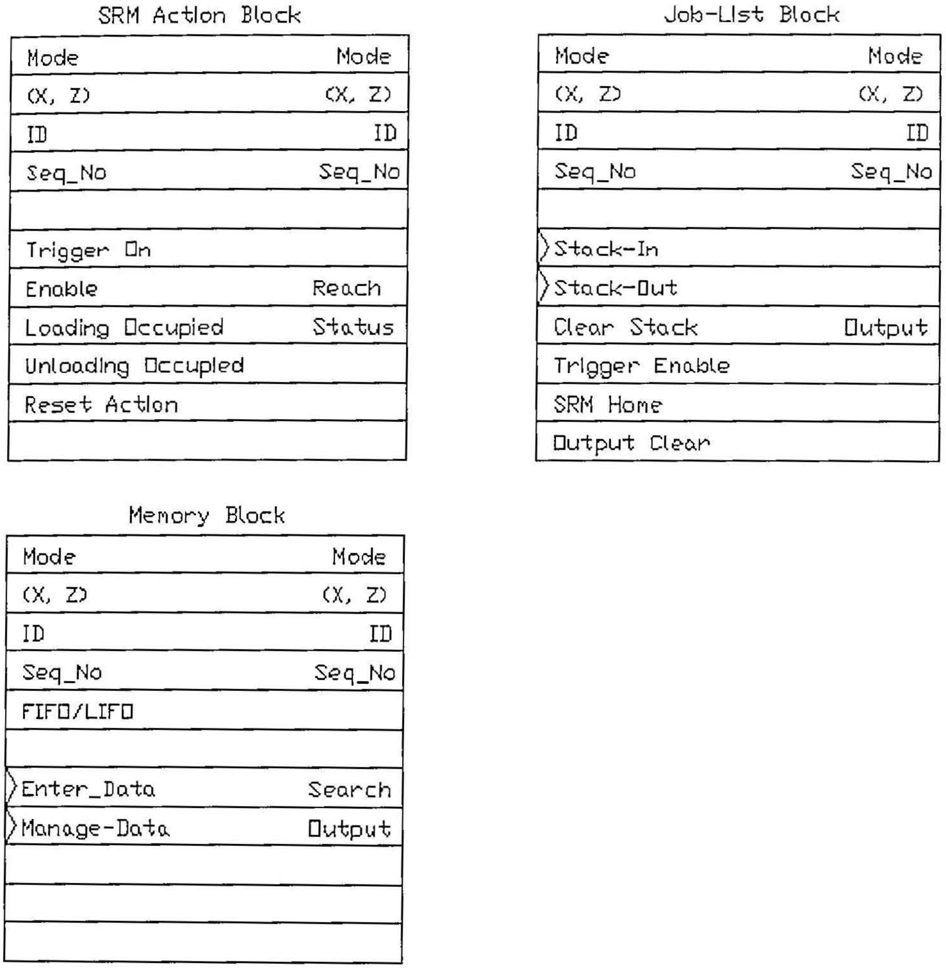


Figure 3.29: SRM Action Block, Job-List Block and Memory Block

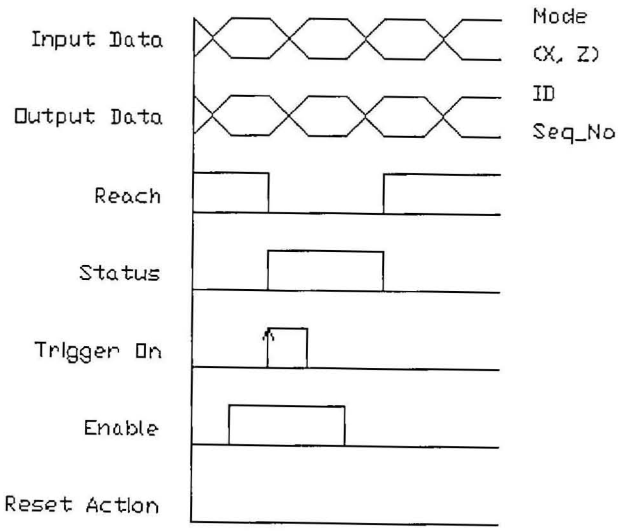


Figure 3.30: The Timing Diagram of the SRM Action Block

The SRM Action block is defined to perform tasks such as “Bridge”, ‘Storage” and “Retrieval” whenever commanded. “Mode” means the tasks either “Bridge”, “Storage” or “Retrieval”; “Bridge” data is assigned as hexadecimal number “000F”; “Storage” as “00FF”; “Retrieval” as “0FFF”. “(X, Z)” specifies the location of the rack cell in binary coded decimal (BCD); the upper byte is assigned as X location; the lower byte as Z location. “ID” specifies the identity of a product in BCD format where lower three digit is allowed; the most significant digit is reserved for memory booking purpose in the memory block. “Seq_No” specifies the sequence number according to the specified product in BCD format. Figure 3.30 reveals the timing diagram of the input and output ports. The block must be enabled under the condition of “Home”. Once enable, the block is ready to use. The block operates in a pre-defined manner. The input data line must be fed with data. Then, whenever “Trigger On” is triggered on at the rising edge, the block function would check the input data and perform the job accordingly. Two output flags, namely “Reach” and “Status”, would change switch their conditions upon “Trigger On”. The block would perform its job accordingly. Once completed, “Reach” would on and “Status” would off, indicating the job completed. The Output data is updated. The loading bay and unloading bay are checked by inductive proximity sensors. These sensors status represent “Loading Occupied’ and “Unloading Occupied” inputs of the block. If “Loading Occupied” is on, the SRM cannot perform tray fetching at the position of the loading bay; while if “Unloading Occupied” is on, the SRM cannot perform tray unloading at the position of the unloading bay.

The SRM Action Block is designed from bottom to up style as in Figure 3.31. The bottom consists of multiples subroutines of which consist of other subroutines in the subsequent layer. For example, the “Bridge” Mode would call the Loading Sequence and Unloading Sequence, which define the action sequence to fetch a tray from the loading bay and put it onto the unloading bay. Both sequences would further call Sub_X_Command and Sub_X_Command for direction decision and speed decision. Furthermore, both subroutines would call X_servo_Command and Z_servo_Command for direction and speed decision before driving the servomotors. The (X, Z)_absolute Counter would update the coordinate position when the servomotors have completed their movements. The logic of the SRM Action block is further elaborated by the help of flow charts.

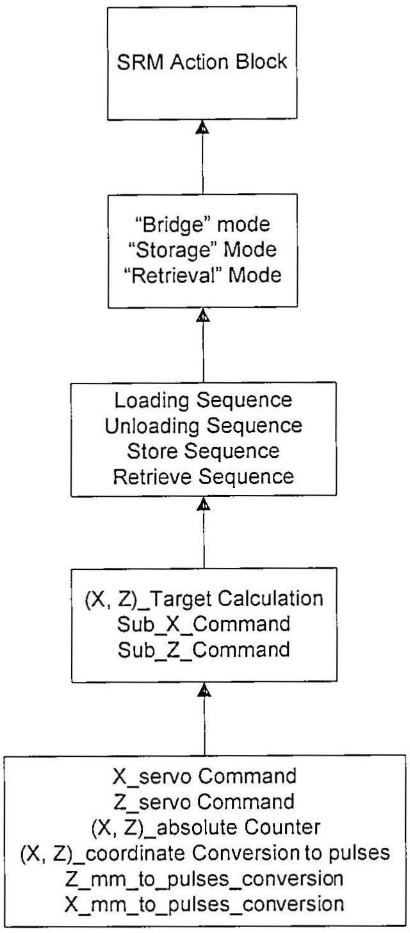


Figure 3.31: Substances of SRM Action Block

Figure 3.32 and Figure 3.33 reveal the flow logic of the three mode of the SRM Action block. The logic flows are simple and direct. In that sense, without much consideration on the detail, the sequence is easier to design. "Reach" and "Status" are the output flags of the block. Both flags are meant for status check when connected with other blocks. Loading Sequence is a series of fetching a tray at loading bay; Unloading Sequence is a series of unloading a tray onto the unloading bay; Store Sequence is a series of storing action the target (X, Z) position; Retrieve Sequence is a series of retrieving action at the target (X, Z) position. These sequences are later designed individually later. By dividing into sub-sequences, the flow charts are designed into a single line yet simple style.

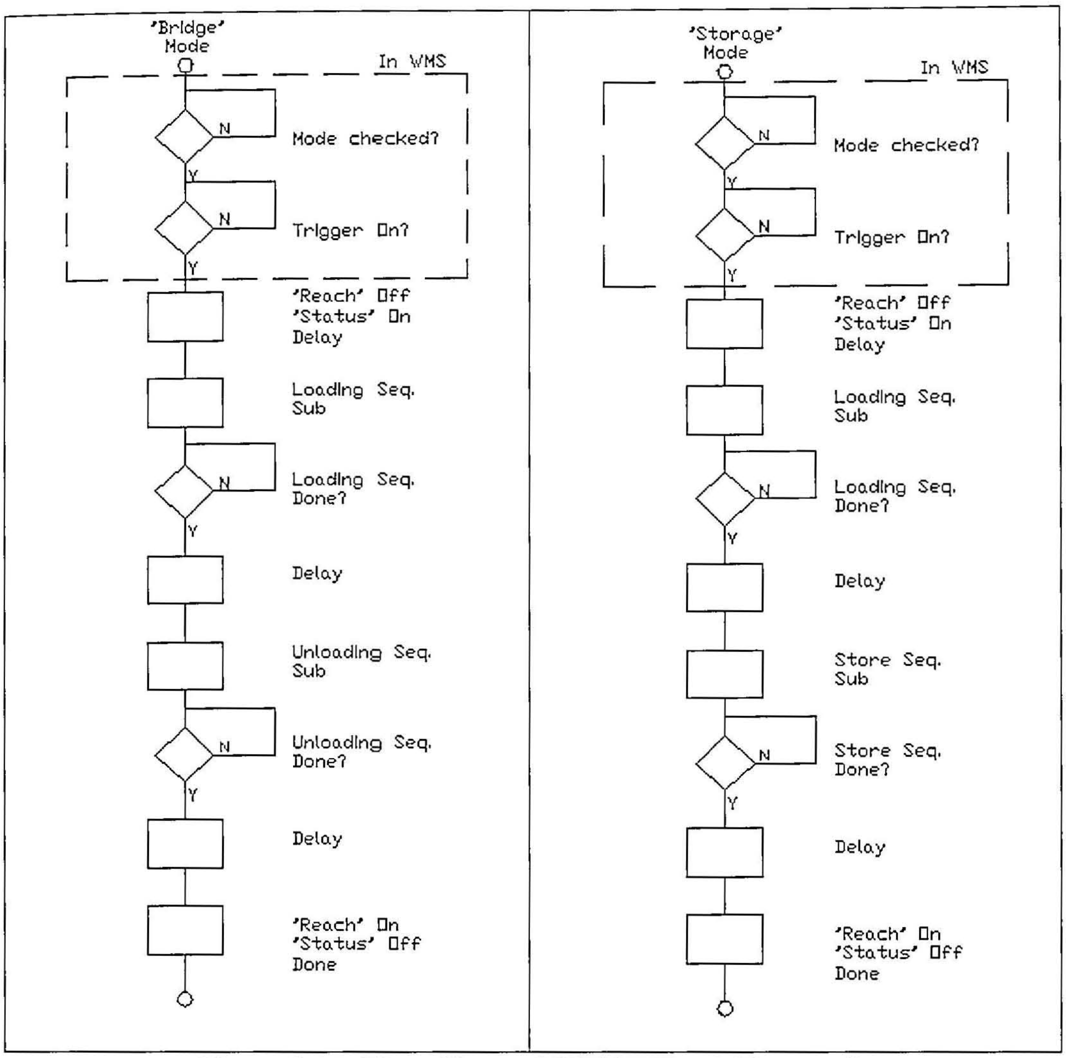


Figure 3.32: Flow Chart of "Bridge" and "Storage"

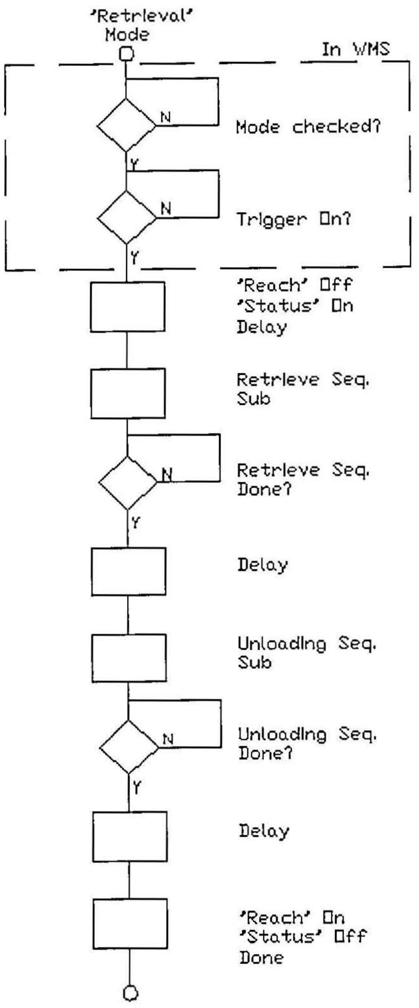


Figure 3.33: Flow Chart of "Retrieval"

Figure 3.34 to Figure 3.37 reveal the sub-sequences of the actions. The SRM can move in X and Z axis when the SRM is in "Sit" position. "Sit" position is a condition purposely designed as SRM_Out, SRM_Turn In and SRM_Grip Off. At these conditions, no extruding mechanical parts are in the middle of the X and Z moving paths. The "Sit" position provides smooth X and Z movement without the possibility of collisions. Loading sub-sequence and Unloading sub-sequence perform tray loading and unloading at the target position of the loading bay and the unloading bay respectively. Both the loading bay and the unloading bay position in pulses numbers are decided via the teaching mode. Somehow, while in Store and Retrieve sub-sequences, the rack target in (X, Z) coordinate is converted to pulses numbers respectively. At the targets positions, Z axis movement can be re-targeted by adding or subtracting an offset value of the original Z target; adding means move lower than the target while subtracting means move upper than the target. These offset values are relative to the target and can be altered freely via program. The sub-sequences do not perform the conversion and direction as well speed decisions but rather call corresponding sub-routines to do the jobs. Again, the flow charts of these sub-sequences are designed in simple and single line style.

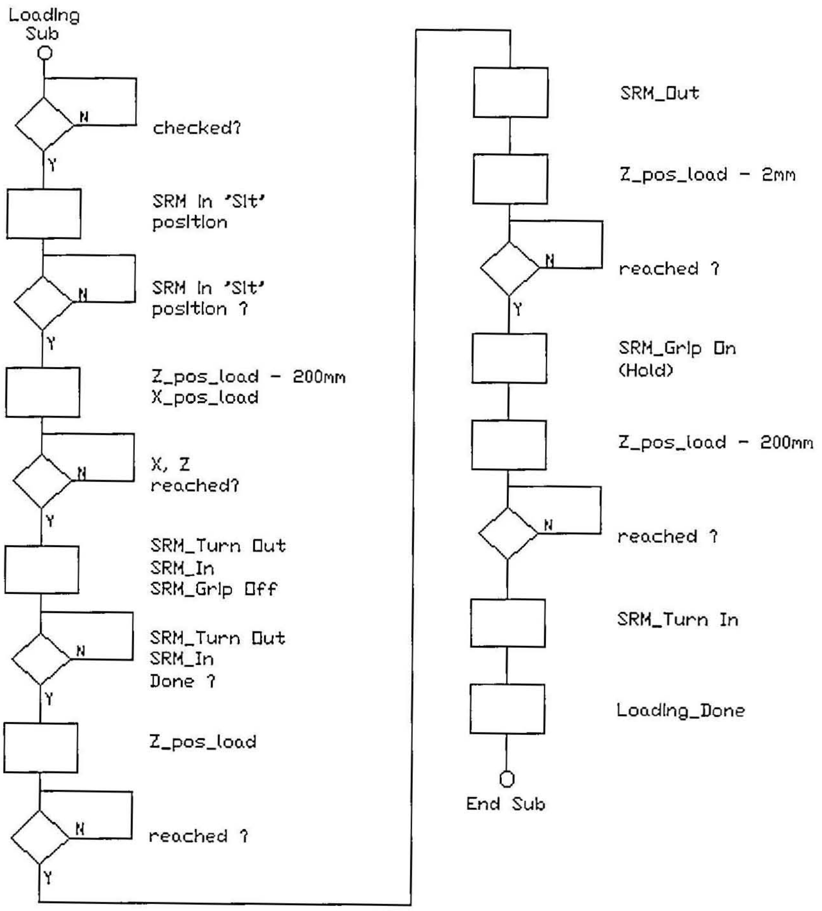


Figure 3.34: Flow Chart of Loading Sub

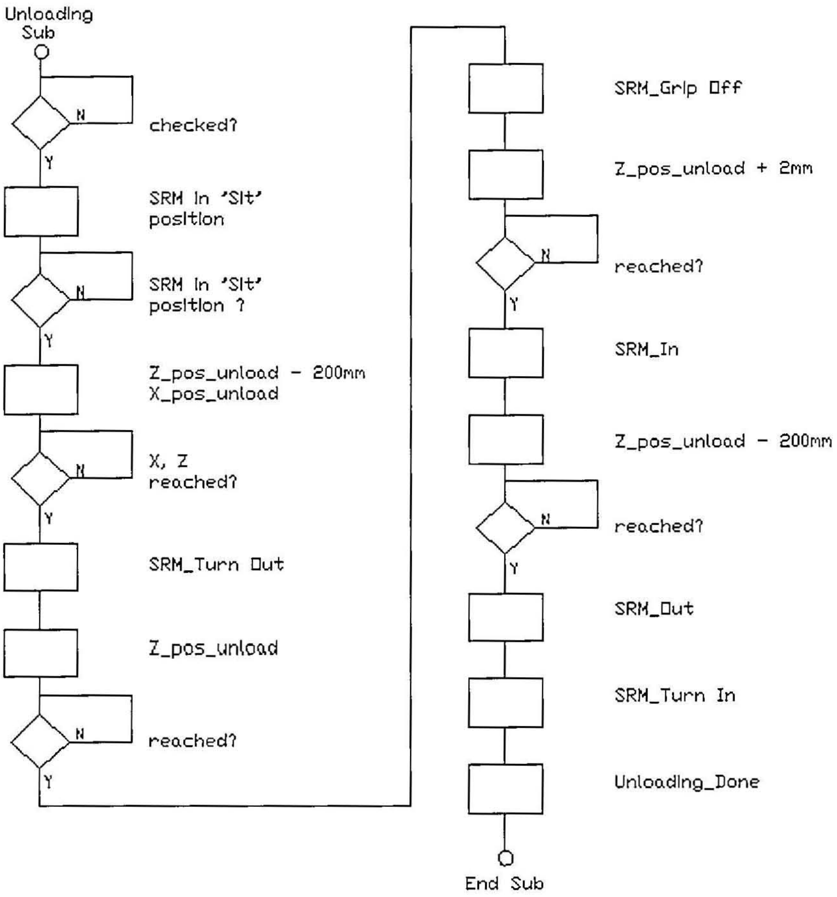


Figure 3.35: Flow Chart of Unloading Sub

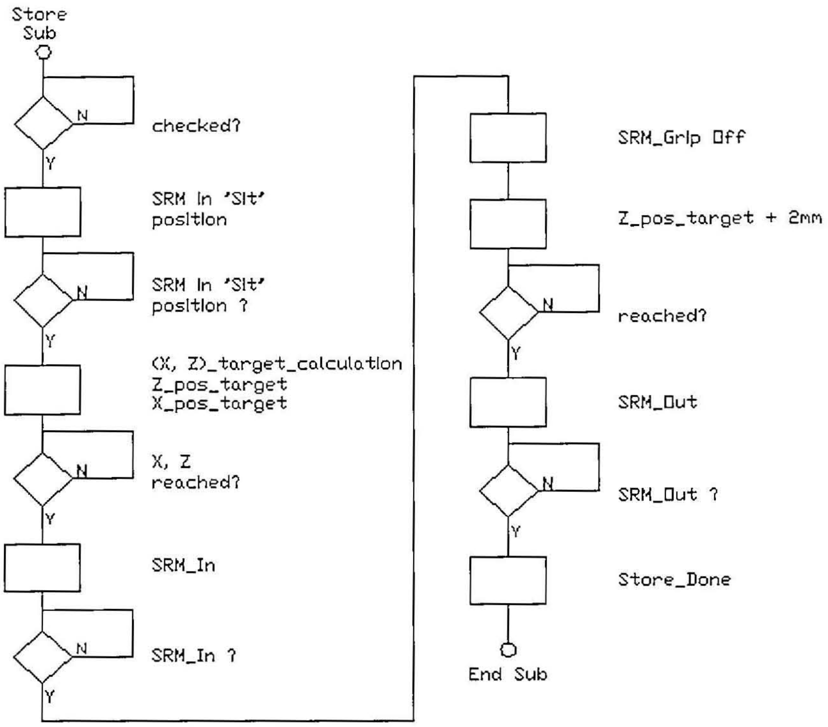


Figure 3.36: Flow Chart of Store Sub

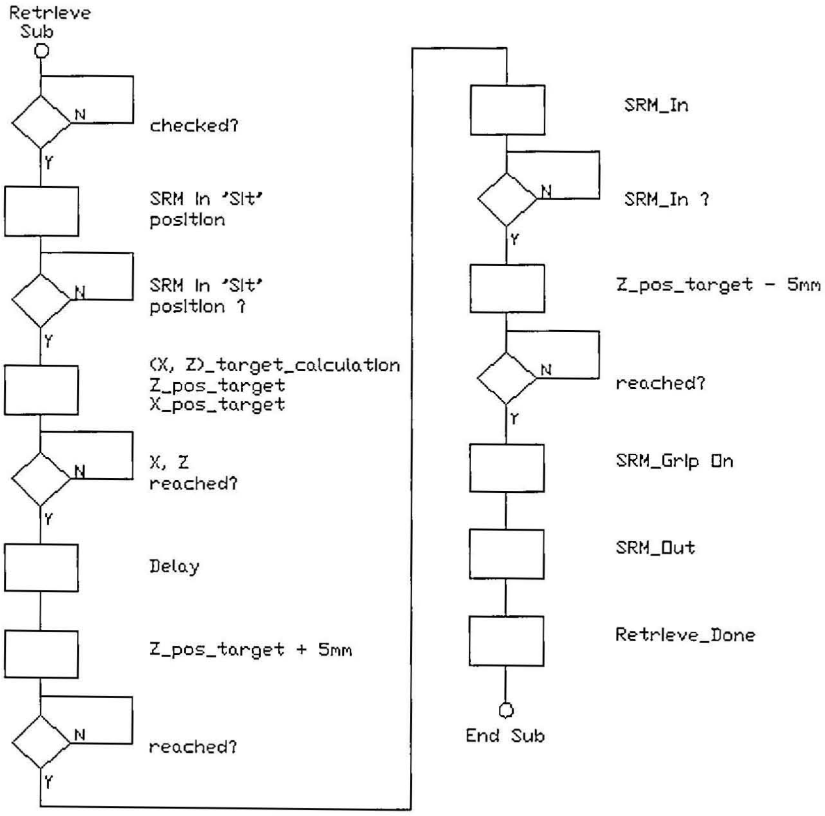


Figure 3.37: Flow Chart of Retrieve Sub

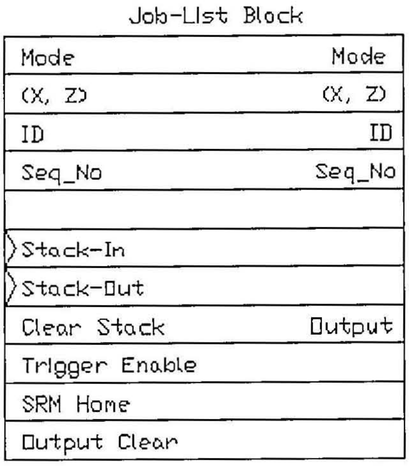


Figure 3.38: Job-List Block

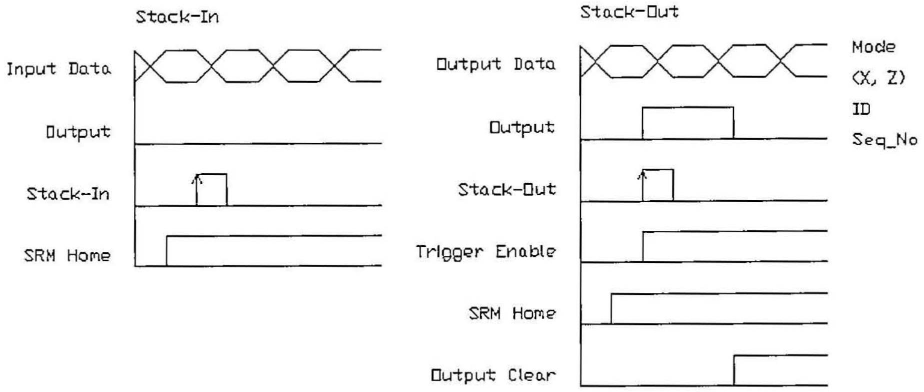


Figure 3.39: Job-List Timing Chart

The Job-List block (Figure 3.38) is designed to have three functions, namely Clear Stack, Stack Input and Stack output. The stack has four data types namely Mode, (X, Z), ID and Seq_No. Each data type has 72 memories allocations. Figure 3.39 shows the timing chart of the Job-List block. Clear Stack will reset all data types into zero regardless all conditions. When SRM Home is on, the Job-List block is enabled. Stack In will save the input data into the stack one after another by a stack pointer. Stack Out will shift upward the stack one at a time. The stack is based on First-Come-First-Serve principle, which means the very first data on top of the stack will be send to Output Data. A rising edge of Stack-In while Trigger Enable is on would automatically stack Input Data into the stack until the stack is full. The Output

Data is the first data of the stack. A rising edge of the Stack-Out would trigger on the Output flag. The Output flag will be cleared when Output Clear is on.

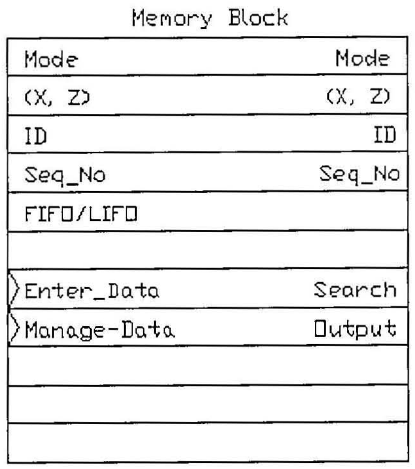


Figure 3.40: Memory Block

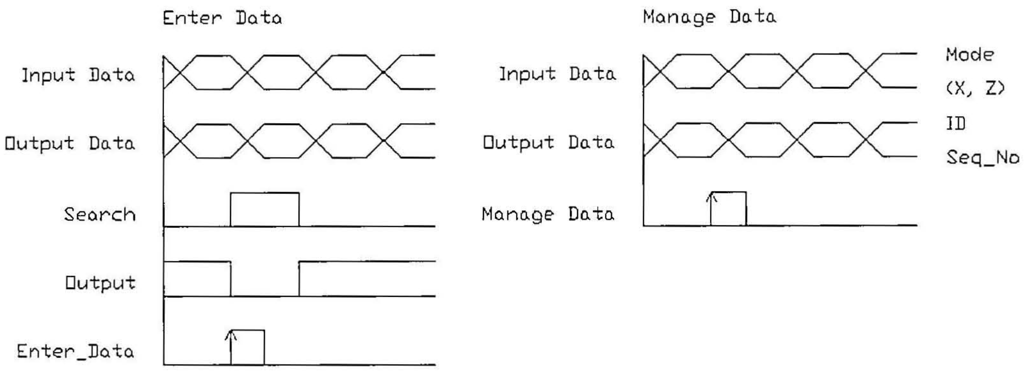


Figure 3.41: Timing Chart of Memory Block

The Memory block (Figure 3.40) operates based on the modes. Its main function is to allocate the coordinate location and rack cell booking by searching for match ID and Seq_No through a series of logic operation. As Figure 3.41 reveals, there are two trigger inputs, namely Enter_Data and Manage_Data. Enter_Data is mainly for coordinate allocation and booking by matching ID and Seq_No; Manage_Data is mainly to update the ID layer and Seq_No layer via a specified index and input data.

The modes are Bridge, Storage and Retrieval. Whenever Enter_Data is triggered, the rising edge is checked. The mode will be compared. When in Bridge Mode, the memory block does not perform any data search and matching through the ID layer and Seq_No layer. The Output Data are written zero except Mode is equal to #000F (Figure 3.42). The output flags, Search and Output are updated.

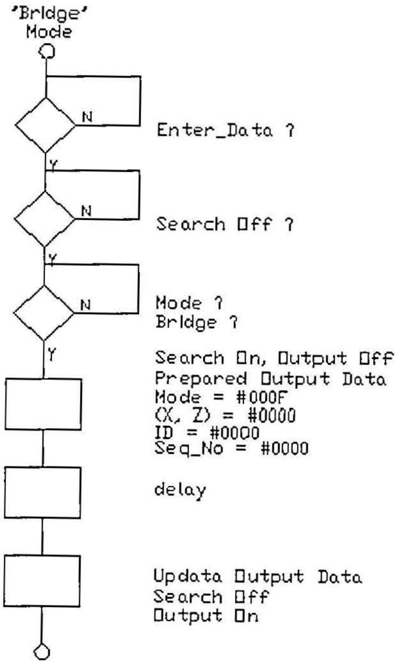


Figure 3.42: Flow Chart of Bridge Mode in Memory Block

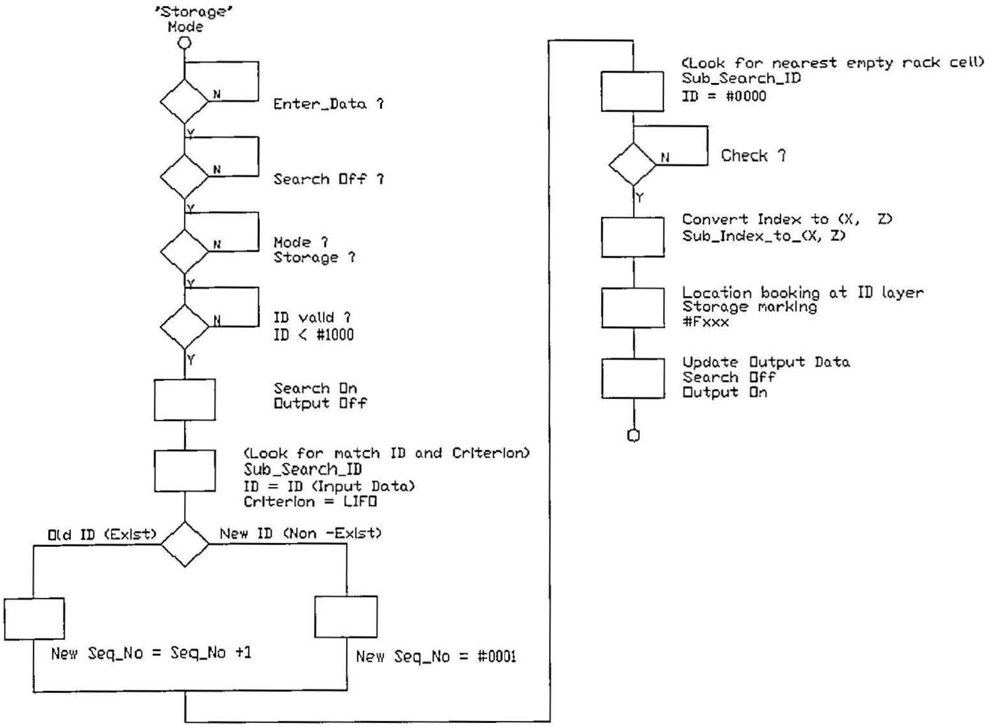


Figure 3.43: Flow Chart of Storage Mode in Memory Block

While in Storage Mode (Figure 3.43), the Memory block would perform a looping check on both the ID layer and the Seq_No layer based on the LIFO criterion. Initially, by LIFO, the initial Seq_No is equal to zero and is added one per loop. The initial matched Index equals to zero. Figure 3.44 shows the flow chart of the Search-ID subroutine. Both ID and Criterion must be fed into the subroutine. Once ID is matched in the ID layer, the Seq_No is compared with Seq_No in the Seq_No layer, named Seq_DM. If the $Seq\_DM > Seq\_No$, then Seq_No is updated as the Seq_DM and the matched Index equals to Seq_Index. The searching and matching will loop until the ID layer has been thoroughly searched. The result of the Seach_ID tells either the ID exists or not. If the ID exists, it implies that a product with same ID already in the rack. Or else, it is a totally new product with a new ID. Both ID and matched Seq_No are preserved for next operations. The Memory block would allocate an empty rack cell by searching with ID layer with data zero. The match Index tells the location. The allocated empty rack cell is booked by store marks. Both the ID layer and the Seq_No layer are marked with the store marks. The store mark in the ID layer highlights the most significant digit with hexadecimal code, F and the ID. For example, the store mark is #F124. (# means hexadecimal number.)

While, the store mark in the Seq_No layer is written with the match Seq_No from previous operation. Once booked, the marked location will be ignored during the memory searching and matching.

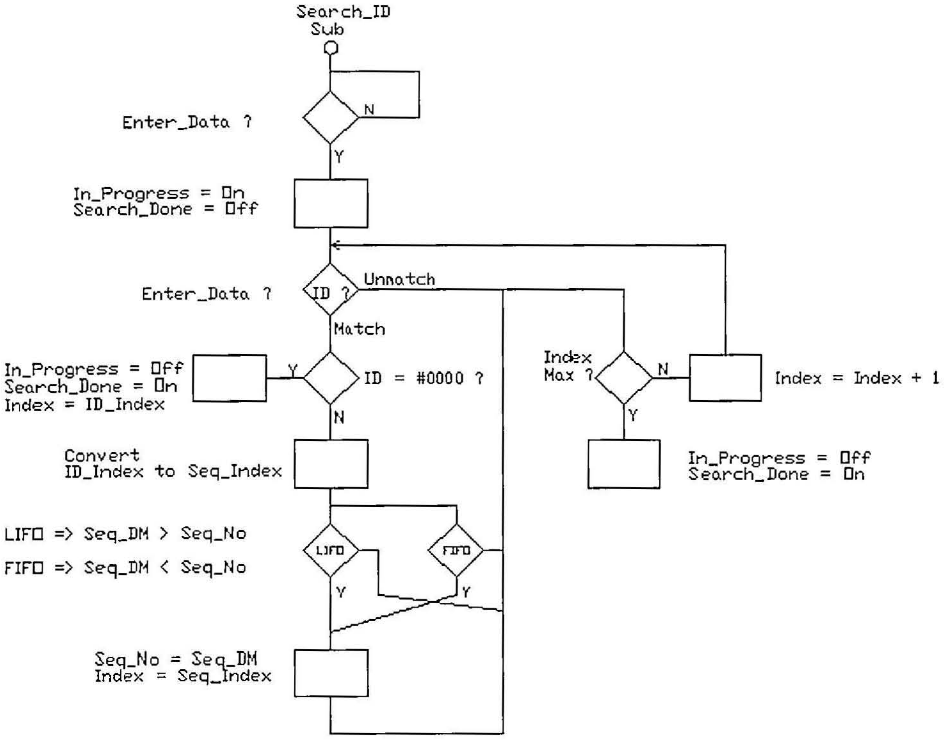


Figure 3.44: Flow Chart of Search_ID subroutine in Memory Block

Figure 3.45 describes the logic operation of the Memory block during the Retrieval Mode. During Retrieval Mode, either criterion, FIFO or LIFO, must be checked. FIFO checks the smallest score of the sequence number (Seq_No) for a matched ID, while LIFO checks the biggest score of the sequence number (Seq_No) for a matched ID. The Retrieval Mode undergoes the operation of loop searching and matching as the Storage Mode. Via the Retrieval Mode, ID layer and Seq_No layer are booked with retrieval marks. Unlike Storage Mode, the ID is the combination of hexadecimal number, E and matched ID. For example, #E124. Once booked, the marked location will be ignored during the memory searching and matching.

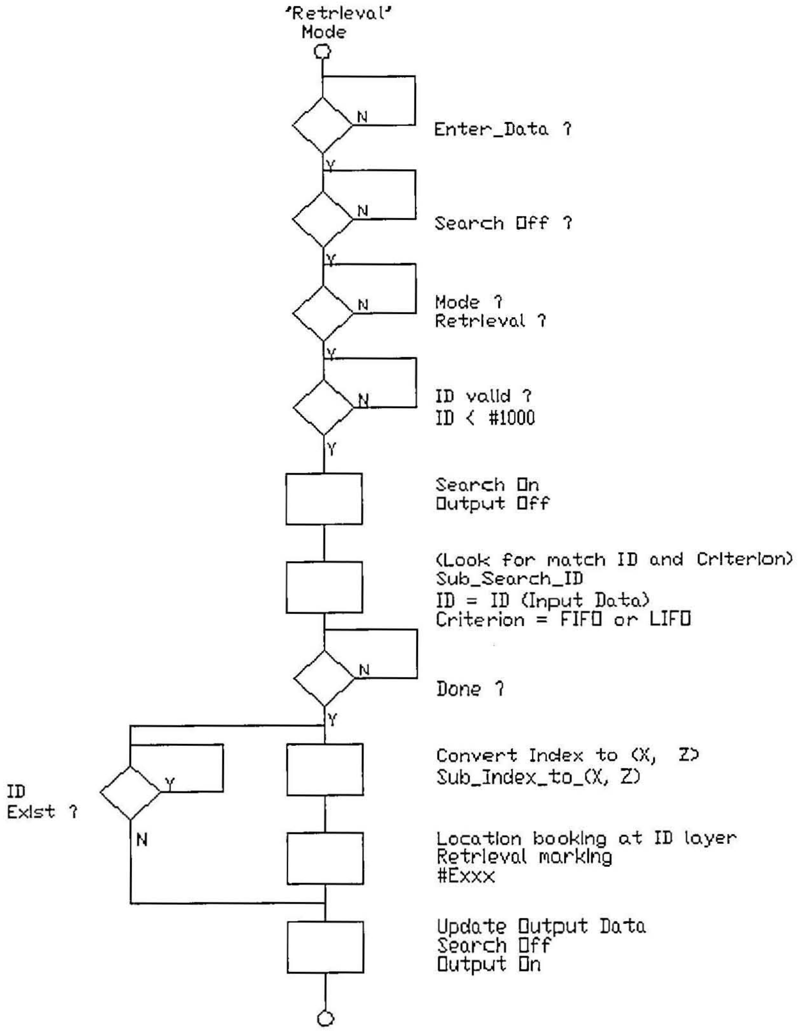


Figure 3.45: Flow Chart of Retrieval Mode in Memory Block

While Storage and Retrieval Mode perform ID searching, matching and booking, the Manage_Memory updates the memories when the SRM has completed a job, either Retrieval or Storage. Figure 3.46 describes the updating operation. The Memory block will do nothing during Bridge mode. The updating operation is performed during Storage or Retrieval Mode is checked. From Figure 3.41, when Manage_Data flag is triggered on, the Input Data are read. When in Storage Mode, ID and Seq_No, based on the index transformed from (X, Z), are written to the memories coordinate. It updates the memories when a product is successfully stored in the rack. When in Retrieval Mode, the ID and Seq_No, based on the index transformed from (X, Z), are reset to zero.

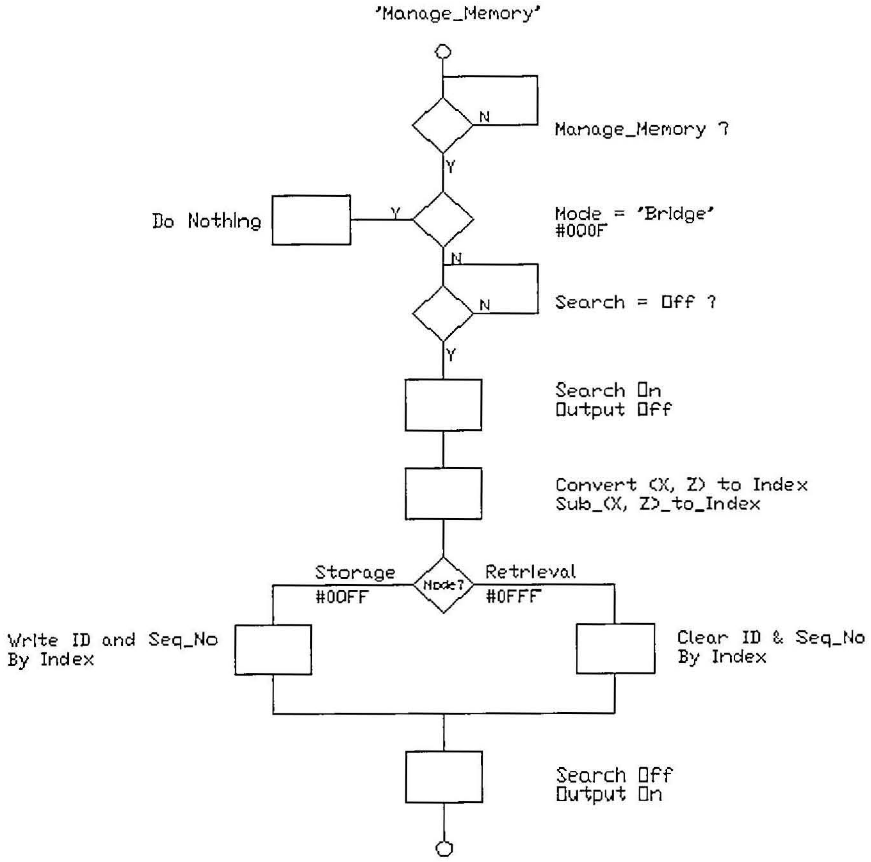


Figure 3.46: Flow Chart of Memory_Manage in Memory Block

The Action Mode will perform properly as expected when these three function blocks are interconnected in a meaningful way. The proposed connection is shown in Figure 3.47. The Memory block will perform memory searching, matching and booking when the WMS triggers the request as well as memory updating when the SRM Action block has completed its task. The data are output to the Job-List block to be stacked. Meanwhile, the SRM Action block will look for the first valid data of the stack. It will perform the job accordingly. The Job-List will update its stack data when the SRM Action block has completed a task. These three blocks are interconnected and interrelated.

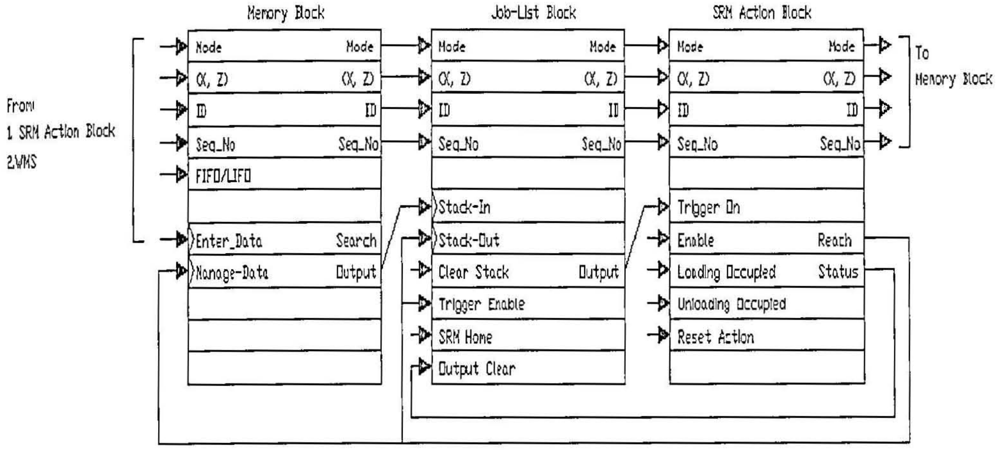


Figure 3.47: Function Blocks Interconnected

3.4 Material Handling System (MHS)

Appendix B shows the inputs and outputs of the PLC of the material handling conveyor (MHS).The MHS consists of two sets of chain conveyors and a transfer table. The MHS is a module by itself. It is communicating with the racking system and the WMS via the system network. As Figure 3.48 reveals, the left chain conveyor is named the Loading conveyor while the right chain conveyor is named the Unloading conveyor. The stations as well as the flow direction are defined and named as shown in the Figure 3.48. At unloading bay, the SRM would unload a tray onto the unloading conveyor. When the SRM has completed unloading action, the product ID is transferred to the MHS for stack purpose Via PLC network. The trays are buffered in a buffer space between the unloading bay and the picking and sorting station. The stack represents the ID carried on each tray as they are buffered. At the picking and sorting station, the tray is matched with the first data of the stack. The data represents the ID carried and is compared with a pre-defined data #0124. When the data is #0124, the MHS would output a right flag to the sorter robot while not #0124, the MHS would output a left flag to the sorter robot. The sorter robot is designed to perform accordingly. The empty trays are re-cycled back to the receiving station and queued. At the receiving station, an operator can load a product onto the tray and define each product with an ID via the WMS. A pushbutton is pressed to release a tray at the receiving station, one at a time.

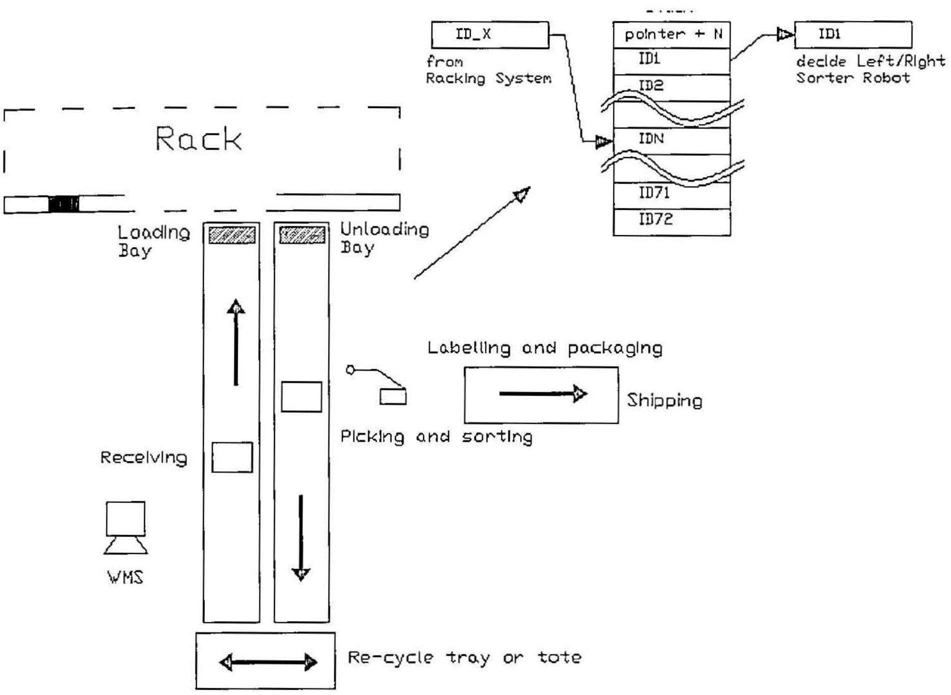


Figure 3.48: Material Handling Conveyor

3.5 The Sorter Robot

The sorter robot is designed to pick a product from a taught picking point and place it to another point. The robot is a Selective Compliant Articulated Robot Arm (SCARA) of Yamaha, YK600X SCARA robot. Refer to [16] and [17] for both the controller RCX40 series and YK series robot instruction manual. The RCX40 controller provides digital inputs and outputs (IO); some of the IO are defined as communication flags with the MHS, as shown in Figure 3.49. Many points can be taught and saved in memory. These points are named and can be called in the robot program.As illustrated in Figure 3.50, three points are defined and taught. The Pick point defines the picking point; A point defines the placing location if a right input flag, DI2(1), is triggered on; B point defines the placing location if a left input flag, DI2(2), is triggered on. The robot is idle at Pick point. Once an input flag either left or right, is triggered on, the robot end-effectors would lower down and grip the product in a tray. The end-effectors are then moving upward. At the upward position, the robot generates an output flag, DO2(2), to the MHS, indicating that a picking process has completed. It performs a placing task accordingly by itself then. The sorter robot program is shown in Figure 3.51.

PLC IO

	Input		Output		
DO2(2) →	2.10	Confirm Pick	Servo ON	102.05	→ DI0(1)
			Interlock ON	102.06	→ DI1(1)
			Prog Start	102.07	→ DI1(2)
			Auto Mode	102.08	→ DI1(3)
			RIght	102.12	→ DI2(1)
			Left	102.13	→ DI2(2)

Figure 3.49: Sorter Robot Interfacing with the MHS

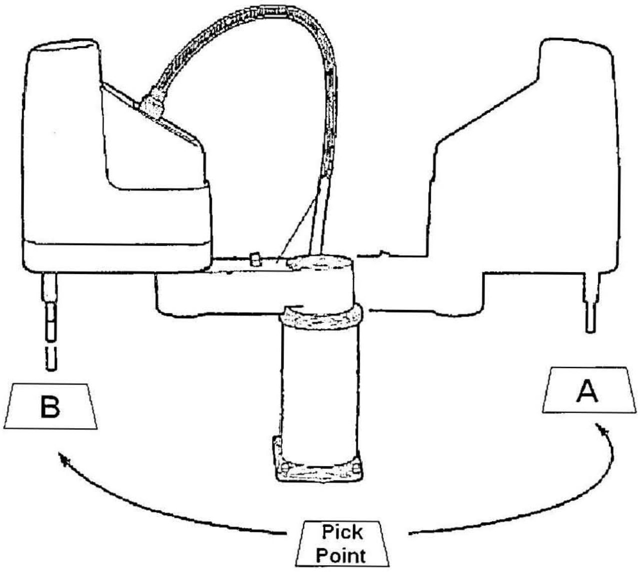


Figure 3.50: Picking Point and Sorting Points A, B

```
PROGR:SORTER ROBOT              'program title
SPEED=70                        'speed setting
*HOMING:                        'Homing routine
MOVE P,P2046,Z=30
IF DI(21)=1 THEN                'If product is type A, then sort for A.
GOSUB *SORTA                    'If A, then go to routine A
ELSEIF DI(22)=1 THEN            'If product is not type A, then sort for B.
GOSUB *SORTB                    'If B, then go to routine B.
ELSE
GOTO *HOMING
ENDIF
GOTO *HOMING
*FIN:
HALT
*SORTA:                         'Routine A
MOVE P,P2026,Z=30
DO(24)=1                        'The gripper on
DO(22)=1                        'Confirm pick is triggered on
DELAY 3000
DO(22)=0                        'Confirm pick is triggered off
MOVE P,P2020,Z=30
DO(24)=0                        'The gripper off
RETURN
*SORTB:                         'Routine B
MOVE P,P2026,Z=30
DO(24)=1                        'The gripper on
DO(22)=1                        'Confirm pick is triggered on
DELAY 3000
DO(22)=0                        'Confirm pick is triggered off
MOVE P,P2057,Z=30
DO(24)=0
RETURN
```

Figure 3.51: Sorter Robot Program

3.6 Communication

The integration involves four individual parts, the Racking System, the MHS, the Sorter Robot and the WMS (Figure 3.52). Serial communication RS232C is applied in between PLC and the host computer. The PLCs are networked using 1:1 Data Link protocol as provided by Omron. This protocol has limited read/write data number but promise satisfactory performance. For 1:1 Data Link, both the PLCs are set to 9600 bps.

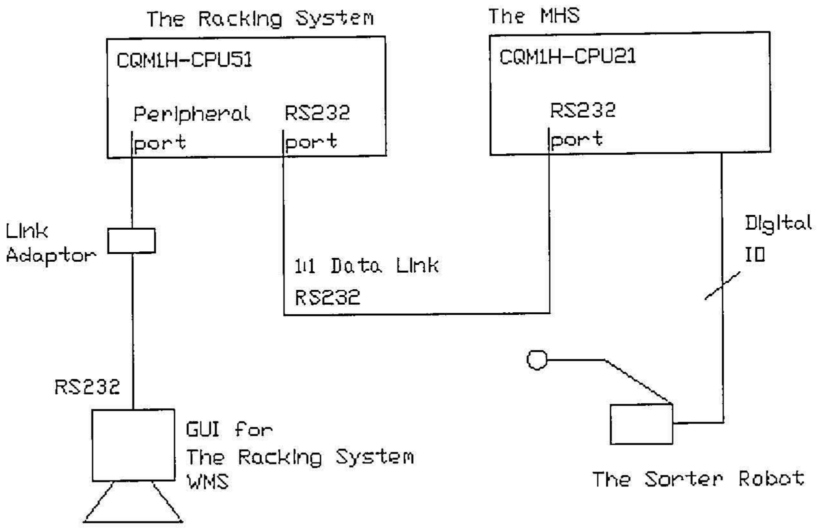


Figure 3.52: Communication Layout

The WMS displays plenty messages and data of the racking system. The data are read/write between the host computer and the PLC of the racking system with 19200 bps. The communication is based on serial communication standard provided by Omron FinsGateway and Omron Compolet, which is a communication driver for Omron PLC and a object operated in Microsoft VB.net environment. The supervisory level is totally a Microsoft VB.net environment; the storage and retrieval management system is programmed in VB.net, for functionality such as auto batch storage/retrieval, manual operating, items database, system status monitoring.

Serial communication RS232C is the cheapest communication available yet common in industry control. It has some limitations. Its maximum communication cable length is 15m and maximum communication speed is limited to 19200 bps because the PLC has only two options, 9600 bps or 19200 bps. The Sorter Robot is communicating with the MHS using digital IO.

CHAPTER 4

RESULT AND DISCUSSION

4.0 Result and Discussion

This project stresses on the design concepts on flexibility, expandability and modularity. The system is divided into two sub-systems, the Racking System and the MHS System. Both sub-systems are individually designed and has standalone controller. Combined with all pre-defined communication flags and register, the system is integrated and able to perform as expected. Flexibility allows a broad spectrum of application environments but is also a major contribution factor in extending application life. Adding new devices will make the system compatible for new challenges. Expandability is closely coupled with flexibility and will allow application in areas not yet defined. Via the application of an AS/RS, the techniques and devices used are not limited to this system. Finally, modification and maintenance are greatly enhanced if the system is modularized in a meaning manner. Generally, the control of system can be subdivided into three levels, namely the device level, control level and the supervisory level. By looking into these levels with an application, the system will provide a systematical training model to the engineering students.

Serial communication is the cheapest and common in industry as a communication protocol. The system has demonstrated the system integration using serial communication. Although, this protocol has its advantage on the cost but it has limited the communication distance and the data size. The maximum communication distance is 15m. Communication delay becomes significant as the data size increases.

4.1 The Device Level

The device level is the lowest level of the system control and is the main concern for a system designer.It mainly involves with the input devices and the output devices. Through out the system, both input and output devices are properly selected and cleverly placed into the system. The input devices are typically electrical transducers that detect the presence of an object using analytical physic properties such as magnet and light. The Racking System has demonstrated the usage of both devices (Appendix A). Both applied devices are of discrete type. The input devices use inductive proximity sensors for (X, Z) servo-axis home and end-limit detection and limit switches for position detection of the SRM multi-cylinder end-effectors. The output devices manipulate two servomotor for (X, Z) axis, pneumatic valves that drive linear and rotary cylinders as the SRM multi-cylinder end-effectors. The MHS (Appendix B) uses same devices except the inductive motor for the servomotor.

Through the application, the device level shows how each device serves for a specified purpose. The inductive proximity sensor is capable of sensing a metal object in a proximity distance and it provides reliable reading even in dusty and dirty environment. The diffused type photo sensor is capable of sensing an object using reflected light and is capable of detecting an object in a proximity distance.

Somehow, it is vulnerable to dirt and dust. Combined with linear and rotary cylinders, the SRM end-effectors can perform tray fetching and placing at the other direction. At the device level, the servomotors are chosen due to it capability in precision positioning. The conversion of rotary action to linear action gives the learning idea of how to drive the servomotor precisely using electrical pulses.

4.2 The Controller Level

The controller level consists of the individual PLC and the network of PLCs. It is the concern of a system designer and programmer. This level provides much information and skills to be learned for the engineering students. The system is designed in modular form. Each sub-system has a PLC that perform its sequential tasks while by properly defined the communication flags, the PLC can be networked and exchange massive meaning data amongst them. At this level, supporting instruction manuals are extremely neeeded as shown in [12] to [17], covering all the controllers and programming tools applied in this project. Technical supports from the original maker will be helpful when there are technical doubts and mis-understanding. "Divide and conquer" method for programming is stressed through out the programming planning. Appendix A and Appendix B show the wiring diagram of both the PLC of the Racking System and the MHS.

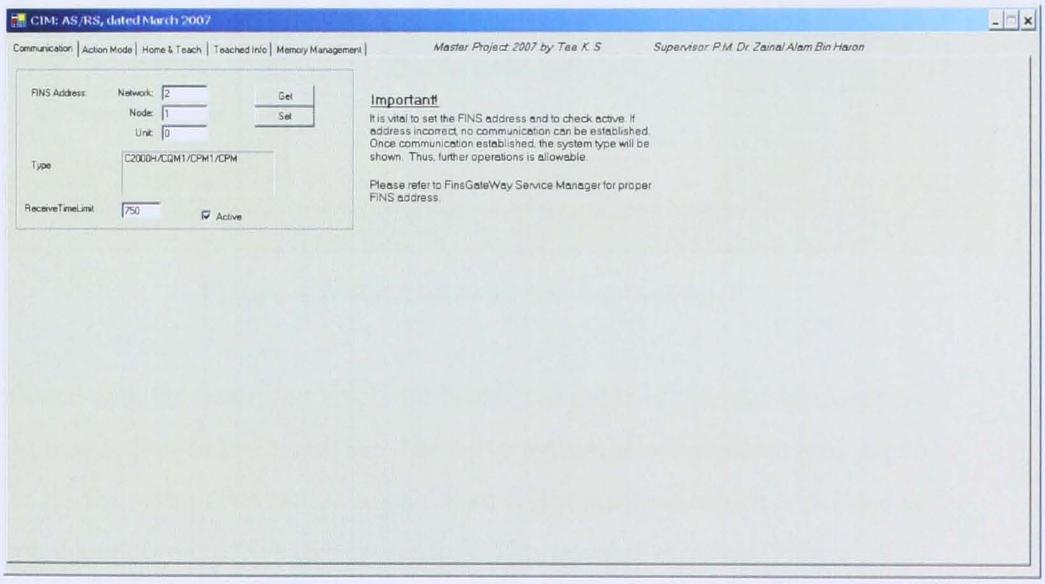
The Racking System has demonstrated that the program is divided and planned according to the functions defined such as Manual Operation, Home Operation, Coordinate Teaching Operation, Action Mode Operation, Job-List and Memory Monitoring and Management. Each function is further divided into sub-routines if the task is too complex. Flow chart as a program planning tool is introduced. Logic flow simplicity is stressed out so that little branching is needed. Via the flow chart, the logic flow can be planned in linguistic format that is easier for understanding rather than in symbol format. Too much tasks in a single flow will complicated the logic plan, resulting in unnecessary confusion in ideas and difficulties in reading the logic flow as it gets lengthy occupying multi pages.

The function blocks introduce a method to increase program repeatability and reusability. This method has concluded a function into a black box with input and output ports. Once each single function block is working on its own and tested for its tasks, it can be virtually treated as a single object. The internal program of the function block can be ignored as it is called and used. The only concern on function block usage is the predefined input and output port response based on it timing chart, ignoring how and whatever its internal program. This enhances repeatability and reusability for third party usage. Three function blocks are demonstrated, namely the SRM action function block, the Job-List function block, and the Memory function block. Inter-connected these blocks feature the tasks expected for the system. It is worth to note that as long as the input and output timing responses are maintained on its timing chart, each block can be modified and re-programmed without giving concern to the other block. Hence, modification and maintenance are made easier.

4.3 The Supervisory level

This level focuses on the application on a host computer, providing monitoring and supervising purposes for the system via a serial communication protocol. A customized WMS incorporates the graphic user interface (GUI), using VB.net [11], Compolet and FinsGateway (products of Omron Ltd.) [14]. By proper design and logic interlocking, the WMS is able to provide operator level operations with graphic and brief text guide. This level demonstrates in the view of an operator on the system control interface. Simplicity and user-friendly concepts are emphasized. The user must feel comfortable and easy to operate the GUI. Text guide are provided through the applet. The applet is explained in detail as followed.

The GUI as shown in Figure 4.1, consists of tabs of Communication, Action Mode, Home & Teach, Teach Info and Memory Management. Before communication between the GUI and the PLC, FinsGateway Service Manager of Omron (Figure 4.3) must be running in the background, with the setting of serial communication. It provides communication protocols which are defined as objects and methods, to be used in VB.net. An object oriented programming. In the GUI, the FINS address is needed as shown in Figure 4.2. The FINS address is a standard of Omron Ltd, for linking a PLC network with a host computer. The system uses a serial communication standard. The FINS address is in the format of Network. Node. Unit (N.n.u). The serial communication protocol is set at 19200 bps, 7 data length, 2 stop bits and even parity.


CIM: AS/RS, dated March 2007
Communication | Action Mode | Home & Teach | Teached Info | Memory Management
Get
Set

Master Project 2007 by Tee K S

Supervisor P.M. Dr. Zainal Alam Bin Haron

FINS Address: Network: 2

Node: 1

Unit: 0

Type: C2000H/CQM1/CPM1/CPM

ReceiveTimeLimit: 750

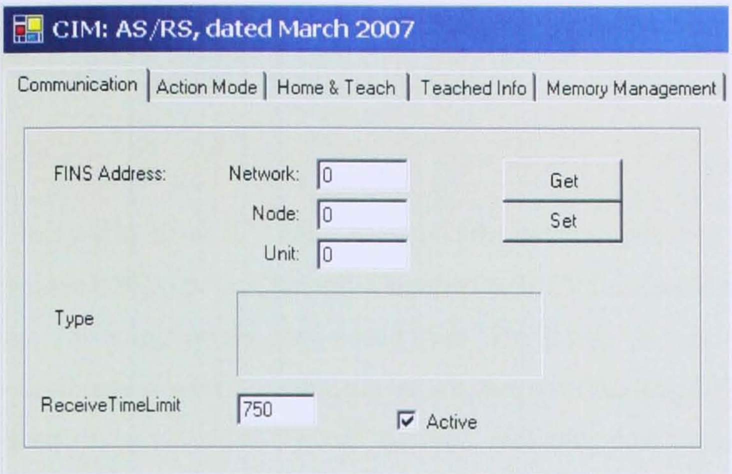
☑ Active

Important!

It is vital to set the FINS address and to check active. If address incorrect, no communication can be established. Once communication established, the system type will be shown. Thus, further operations is allowable.

Please refer to FinsGateWay Service Manager for proper FINS address.

Figure 4.1: The WMS



FINS Address: Network: 0

Node: 0

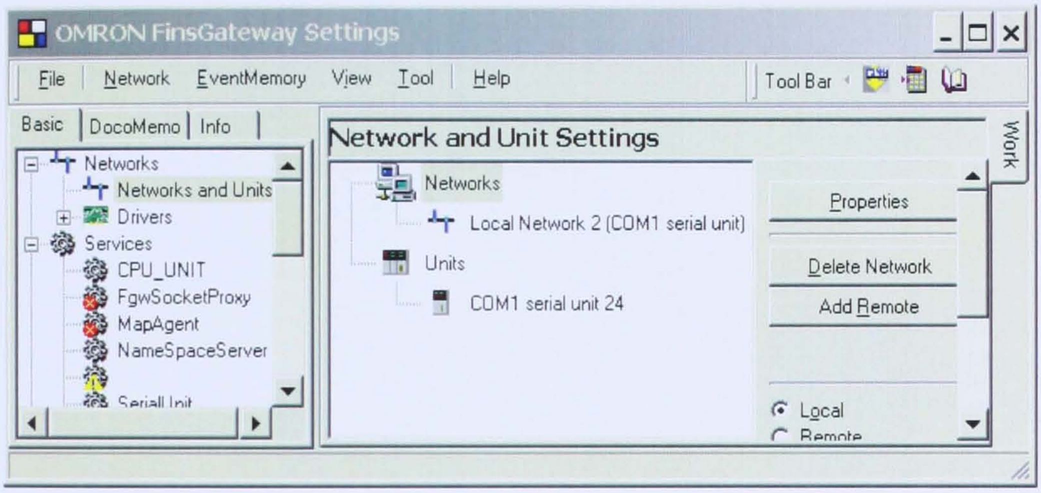
Unit: 0

Type

ReceiveTimeLimit 750

☑ Active

Figure 4.2: Communication Address Setup



Network and Unit Settings

Networks

Local Network 2 (COM1 serial unit)

Units

COM1 serial unit 24

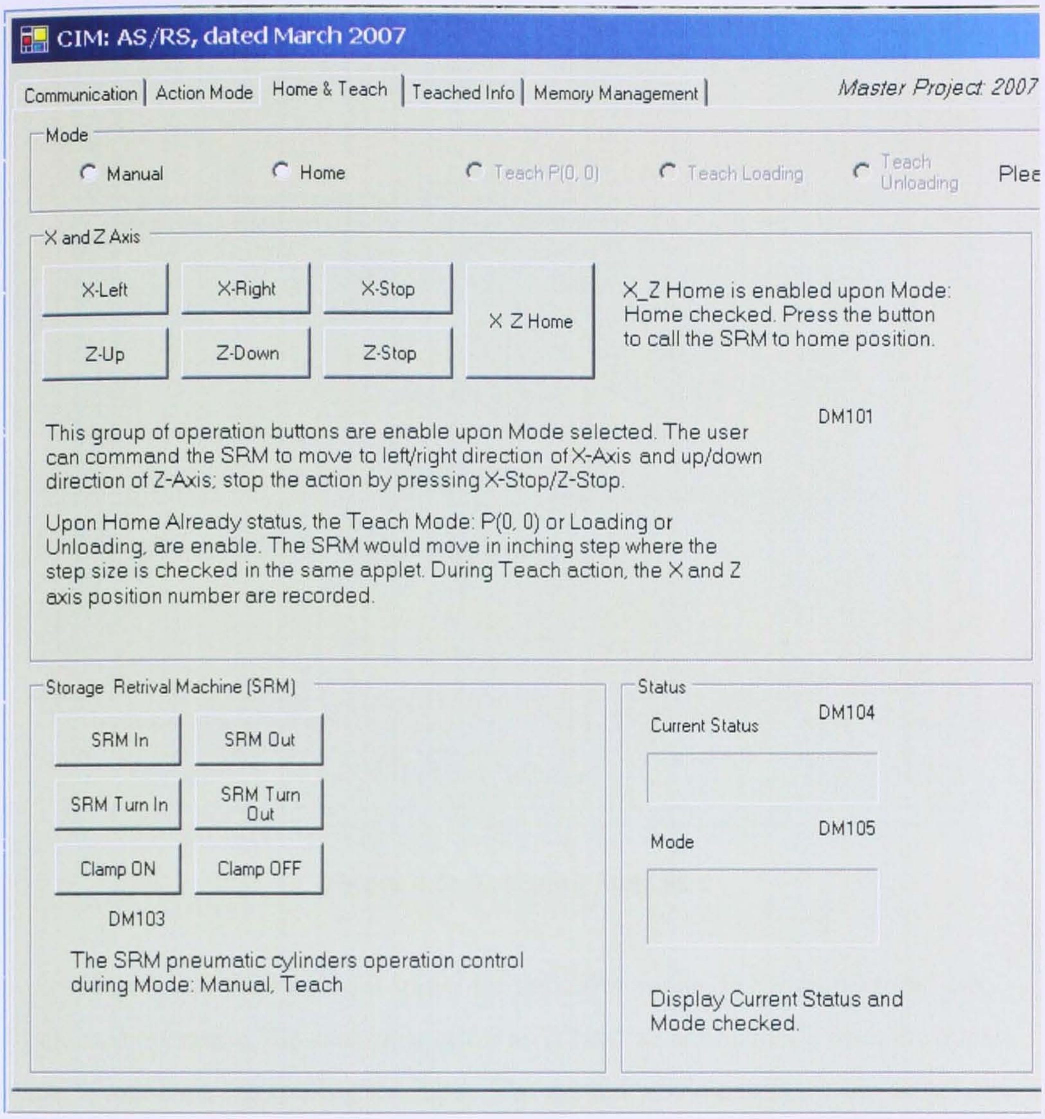
● Local

○ Remote

Figure 4.3: FinsGateway Service Manager

Combined with the controller level, the WMS is capable of monitoring status and giving instructions to the controller. The servo system is not absolute type. Upon power on, the system is required to call to an origin position, which is defined as the far left, upper corner of the rack by sensors. This position is called "Home". It is worth to note that the "Home" position is not equal to the coordinate (0, 0). At "Home" position, both X and Z pulses number is reset to zero. Every position is referred to the "Home" position relatively. The user needs to check the "Home" radio button and clicks "Home" button as shown in Figure 4.4. By checking the 'Manual" radio button, the user shifts the control mode into manual mode; the X, Z and the SRM can be manually triggered by clicking the buttons.

Once the system status is “Home Already”, the user can perform “Coordinate Teaching” for three critical points, namely Coordinate (0, 0), Loading Bay and Unloading Bay. The taught points can be read from ‘Teach Info” tab. It is worth to note that the coordinate system designed above is offset with the taught Coordinate (0, 0). Every position is relative to “Home” position, including coordinate (0, 0), Loading Bay and Unloading Bay. As shown in Figure 4.5, in teaching mode, the X and Z axis can inch at three steps, namely small step, medium step and large step. The point teaching has the advantage of reconfiguring new points if there is any mechanical structure change. Say, the loading bay and the unloading bay are shifted to left by 100mm. The loading bay and unloading bay must be different as before. Hence, the teaching mode provides a facility to the user to reconfigure new points without the need to access the PLC and program.



X_Z Home is enabled upon Mode: Home checked. Press the button to call the SRM to home position.

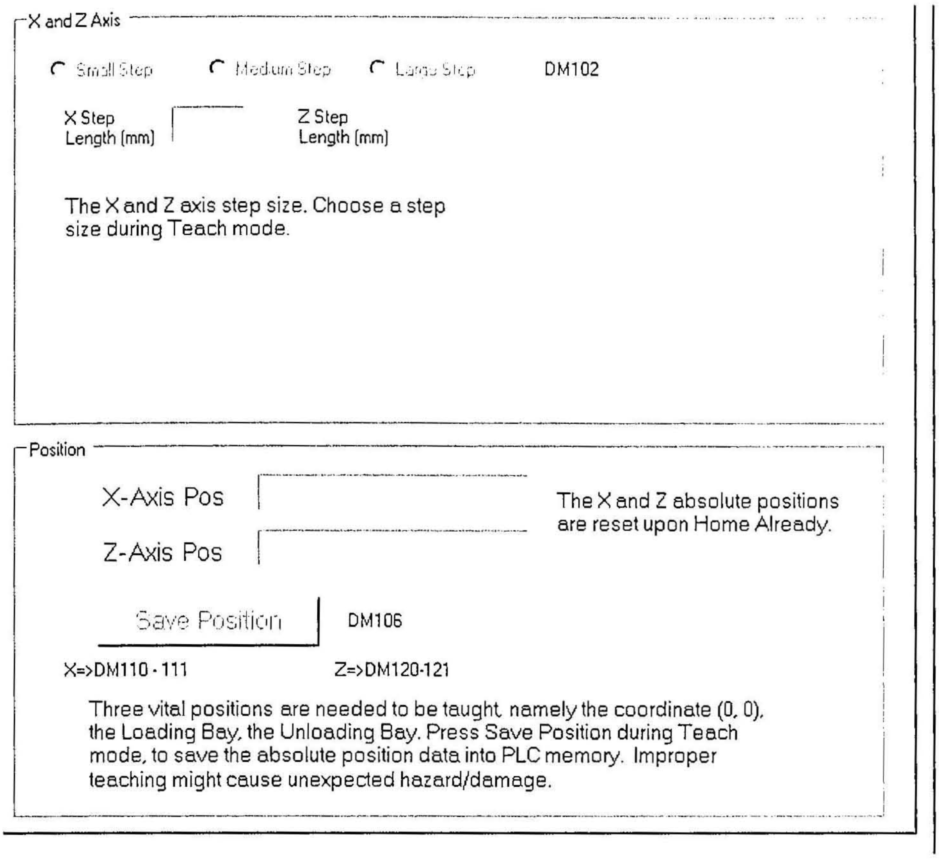
This group of operation buttons are enable upon Mode selected. The user can command the SRM to move to left/right direction of X-Axis and up/down direction of Z-Axis; stop the action by pressing X-Stop/Z-Stop.

Upon Home Already status, the Teach Mode: P(0, 0) or Loading or Unloading, are enable. The SRM would move in inching step where the step size is checked in the same applet. During Teach action, the X and Z axis position number are recorded.

The SRM pneumatic cylinders operation control during Mode: Manual, Teach

Display Current Status and Mode checked.

Figure 4.4: Manual and Home



The X and Z axis step size. Choose a step size during Teach mode.

The X and Z absolute positions are reset upon Home Already.

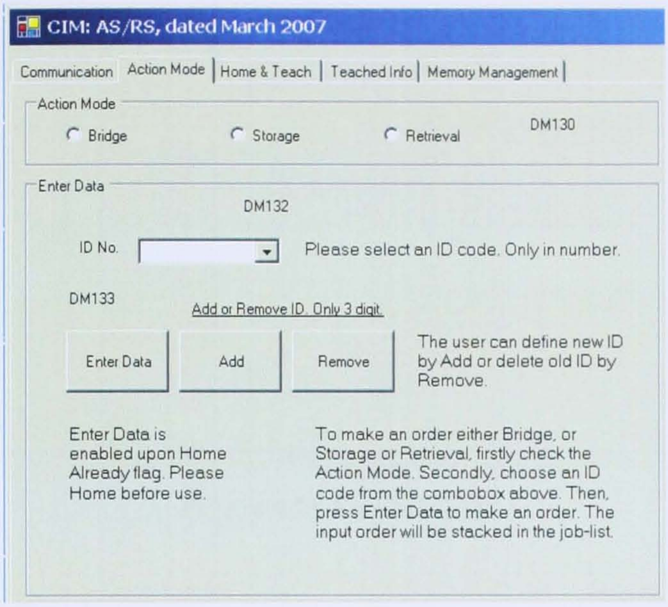
Three vital positions are needed to be taught, namely the coordinate (0, 0), the Loading Bay, the Unloading Bay. Press Save Position during Teach mode, to save the absolute position data into PLC memory. Improper teaching might cause unexpected hazard/damage.

Figure 4.5: Teaching Step Size

Figure 4.6 shows partial applet for the action mode. In "Action Mode" tab, for normal operation, the user must select an ID and an action mode from the applet. Press "Enter Data" to confirm the input. The system would automatically search the entire memory for identical product ID and automatically generate a job-list as shown in Figure 4.7. There are 72 allocations for job queue. The SRM would perform the action according to the first job listed in the job-list and delete the job once completed. There are two retrieval criteria for selection, FIFO and LIFO respectively. First-In-First-Out (FIFO) criteria tells the system the coordinate of a matched product with the smallest sequence number; while Last-In-First-Out criteria tells the system the coordinate of a matched product with the largest sequence number.

The system allows multiple jobs to be queued up regardless the SRM has not completed a job in process. All jobs are queued in the job-list as shown below. The

Job-List has for column of data, namely Mode, (X, Z), ID and Seq. Mode is coded; #000F means Bridge mode; #00FF means Storage mode; #0FFF means Retrieval. (X, Z) displays the target coordinate. ID displays the identity code of the product. Seq displays the sequence number of the product.



Please select an ID code. Only in number.

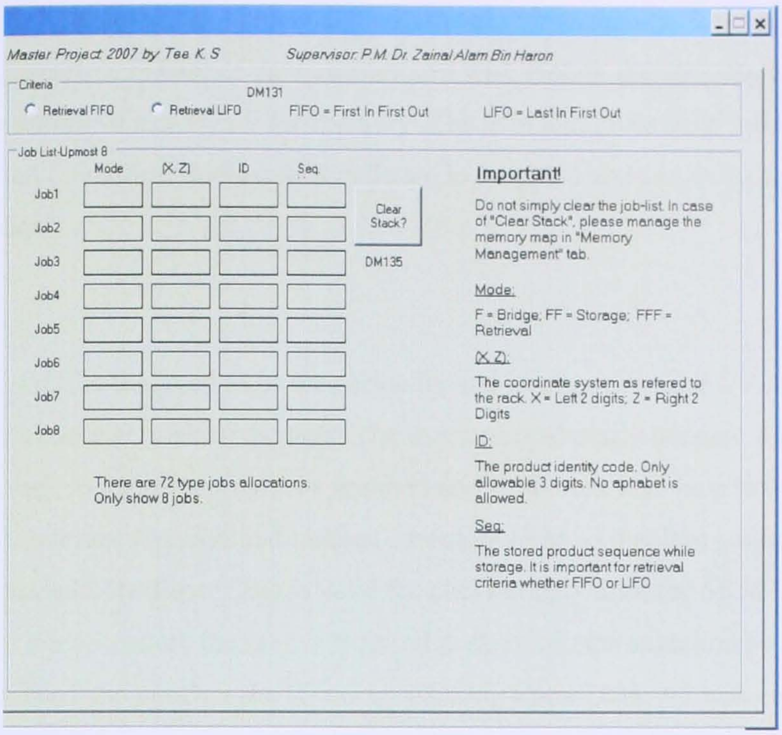
Add or Remove ID. Only 3 digit.

The user can define new ID by Add or delete old ID by Remove.

Enter Data is enabled upon Home Already flag. Please Home before use.

To make an order either Bridge, or Storage or Retrieval, firstly check the Action Mode. Secondly, choose an ID code from the combobox above. Then, press Enter Data to make an order. The input order will be stacked in the job-list.

Figure 4.6: Action Mode



Master Project 2007 by Tee K S *Supervisor: P.M. Dr. Zainal Alam Bin Haron*

Criteria

DM131

○ Retrieval FIFO ○ Retrieval LIFO FIFO = First In First Out LIFO = Last In First Out

Job List-Upmost 8

	Mode	(X, Z)	ID	Seq.
Job1				
Job2				
Job3				
Job4				
Job5				
Job6				
Job7				
Job8				

There are 72 type jobs allocations. Only show 8 jobs.

Important!

Do not simply clear the job-list. In case of "Clear Stack", please manage the memory map in "Memory Management" tab.

Mode:

F = Bridge; FF = Storage; FFF = Retrieval

(X, Z):

The coordinate system as refered to the rack. X = Left 2 digits; Z = Right 2 Digits

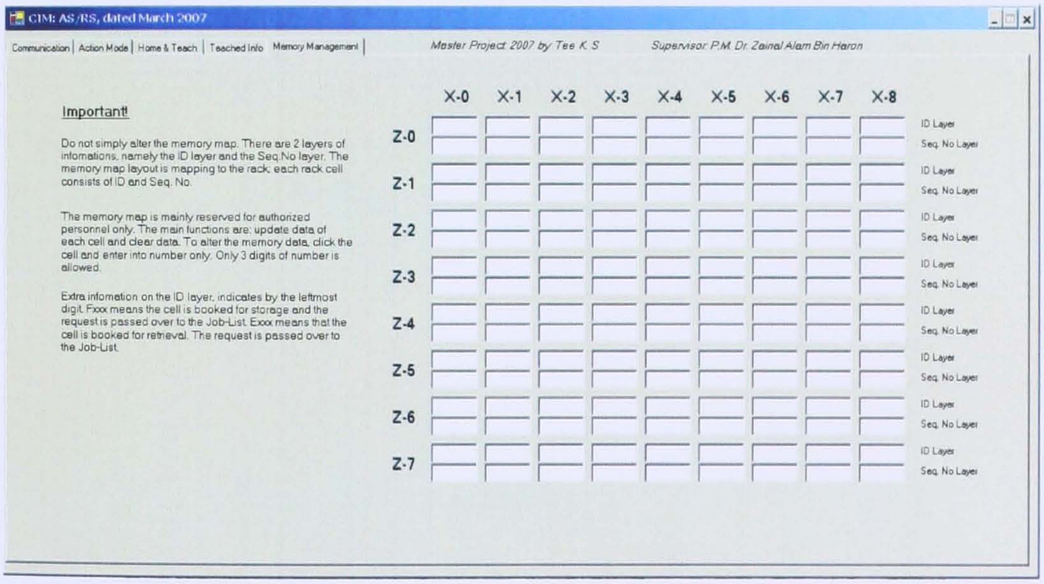
ID:

The product identity code. Only allowable 3 digits. No aphabet is allowed.

Seq:

The stored product sequence while storage. It is important for retrieval criteria whether FIFO or LIFO

Figure 4.7: The Job-List



Master Project 2007 by Tee K S *Supervisor: P.M. Dr. Zainal Alam Bin Haron*

Important!

Do not simply alter the memory map. There are 2 layers of infomations, namely the ID layer and the Seq.No layer. The memory map layout is mapping to the rack; each rack cell consists of ID and Seq. No.

The memory map is mainly reserved for authorized personnel only. The main functions are: update data of each cell and clear data. To alter the memory data, click the cell and enter into number only. Only 3 digits of number is allowed.

Extra infomation on the ID layer, indicates by the leftmost digit. Fxxx means the cell is booked for storage and the request is passed over to the Job-List. Exxx means that the cell is booked for retrieval. The request is passed over to the Job-List.

	X-0	X-1	X-2	X-3	X-4	X-5	X-6	X-7	X-8	
Z-0										ID Layer
										Seq. No Layer
Z-1										ID Layer
										Seq. No Layer
Z-2										ID Layer
										Seq. No Layer
Z-3										ID Layer
										Seq. No Layer
Z-4										ID Layer
										Seq. No Layer
Z-5										ID Layer
										Seq. No Layer
Z-6										ID Layer
										Seq. No Layer
Z-7										ID Layer
										Seq. No Layer

Figure 4.8: Memory Management

The Memory Management applet (Figure 4.8) allows the user to monitor as well as to edit the memories which are mapping to the rack. The memories consist of two layers, the ID layer and the sequence number layer respectively. This applet provides a facility for a user to manually edit the rack information. It is not password-protected. In case of data mismatched with record, the user can manually clean the memory or just to edit the memory. The help text gives brief information of the usage and operation. Booked rack cells are highlighted and can be easily seen from the map.

The GUI is designed to be user-friendly and informative. The VB.net applet must not confuse and mislead the user. The user can read easily because the text size is large enough to read; functions are grouped and supported with help texts. Meanwhile, relevant function and buttons are interlocked so that less confusion to the user. For example, the Enter Data is valid for clicking only after the SRM is home. Throughout the operation, the user is required to do minimum selection by clicking the radio buttons and confirm the ID no. by clicking Enter Data. All tasks entered are queued in the Job-List.

CHAPTER 5

CONCLUSION AND SUGGESTION

5.0 Conclusion And Suggestion

As definition, the AS/RS consists of the receiving, material handling, storage, picking and sortation, shipping, labeling and packing, the warehouse management and personnel [1]. Combined with all levels, the device level, the controller level and the supervisory level, the system is integrated and operates as the definition of an AS/RS. This project can provide as a training model for the engineering students. This model will convey design concepts and techniques involved through the AR/RS. These design concepts, flexibility, expandability and modularity, are crucial of giving the direction of a system design. They are always valid and evolve as new techniques emerge. Flexibility and expandability allow a broad spectrum of application, not limited within the AS/RS itself but other application of CIM system and future possible application. Modularity concept emphasizes the need for modification and maintenance in an industrial application.

The engineering students benefit from the AS/RS, learning the system from the ground. In general, the techniques are classified into three levels. The device level tells the selection of devices based on their characteristic and property. The controller level has emphasized the importance of supporting technical instruction manuals and elaborates in detail the PLC in the program planning and communication network amongst PLCs and a host computer. At this level, the AS/RS complex tasks are "divided and conquered", into subsequent routines. Systematical programming planning using flow chart method demonstrates the concept of simplicity and the benefit in linguistic format. Lastly, the introduction of function block has increased repeatability and reusability in program. The supervisory level has emphasized the design of the GUI applet of a host computer, from the perspective of the end user of the system in term of user-friendly and simplicity. These levels are integrated via the serial communication protocol. It is the cheapest industrial standard although maximum communication distance is 15m and the speed drops as the data size increases.

5.1 Future Works

The system can be upgraded in both hardware and software. Firstly, to allow massive data transfer between the controller and host computer without sacrificing the communication speed, an Ethernet protocol can be introduced to the existing system. It allows the usage of TCP/IP as the protocol, making an integration into a Local Area Network (LAN).

Secondly, artificial intelligence can be applied in the data mining procedure and jobs queuing criteria for minimum duty cycle and efficiency. It involves the AS/RS rack information to a computer terminal and artificial intelligence software in the supervisory level. A computer is capable of storing and managing massive data. In a computer, it is much easier to do data sorting, data mining as well as applying artificial intelligence to the solution.

Thirdly, Radio-frequency identification (RFID) can be implemented into the AS/RS. The RFID is expected to enhance the AS/RS with meaningful information attached to each trays or even products. The system is expected to be informative.

REFERENCE

[1] Qiu, R.G., Sr. and Sangwan, R.S. (2005) 'An approach to relieving warehouse pain points', Networking, Sensing and Control, 2005. Proceedings. 2005 IEEE, pp.197 - 201

[2] Dotoli, M.; Fanti, M.P.; Iacobellis, G. (2004) 'Comparing deadlock detection and avoidance policies in automated storage and retrieval systems' Systems, Man and Cybernetics, 2004 IEEE International Conference on 10-13 Oct. 2004, Vol. 2, pp.1607 - 1612

[3] Ya-Hong Hu, Wen-Jing Hsu and Xiang Xu (2004) 'Efficient algorithms for load shuffling in split-platform AS/RS', Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference,
Volume 3, pp.2717 - 2722

[4] Suesut, T.; Gulphanich, S.; Nilas, P.; Roengruen, P.; Tirasesth, K (2004) 'Demand forecasting approach inventory control for warehouse automation', TENCON 2004. 2004 IEEE Region 10 Conference.

[5] Suesut, T., Tipsuwanporn, V., Gulphanich, S., Rodcumtui, J. and Sukprasert, P. (2002) 'A design of automatic warehouse for Internet based system', Industrial Technology, 2002. IEEE ICIT '02. 2002 IEEE International Conference on

[6] Seng-Yuh Liou, Yuau-Tay Chen, Chao-Wei Chang and Meng-Jiun Wu (1995) 'Integration automatic warehouse and network with two-ways communication in CIM system', Industrial Automation and Control: Emerging Technologies, 1995., International IEEE/IAS Conference, pp:262 - 266

[7] Takakuwa, S. (1994) 'Precise modeling and analysis of large-scale AS/RS',

Simulation Conference Proceedings, 1994. Winter Dec. 1994, pp.1001 - 1007

[8] Serafini, P. and Ukovich, W (1988) 'Operating an automated storage and retrieval system', Computer Integrated Manufacturing, 1988., International Conference, pp.29 - 34

[9] Author: We-Min Chow (1986) 'Development of an automated storage and retrieval system for manufacturing assembly lines', Robotics and Automation. Proceedings. IEEE International Conference, Vol. 3, pp.490 - 495

[10] Frazelle, E. (1986) 'Design problems in automated warehousing', Robotics and Automation. Proceedings. 1986 IEEE International Conference, vol. 3, Apr 1986, pp. 486 - 489

[11] Nitin Pandey, Yesh Singhal, Mridula Parihar. (2002) 'Visual studio.NET all-in-one desk reference for dummies', New York .

[12] Sysmac CQM1H Series: Operation Manual (2005), W363-E1-07. Omron

[13] Sysmac CQM1H Series: Programming Manual (2005), W364-E1-04. Omron

[14] Sysmac Compolet Online Help. Omron

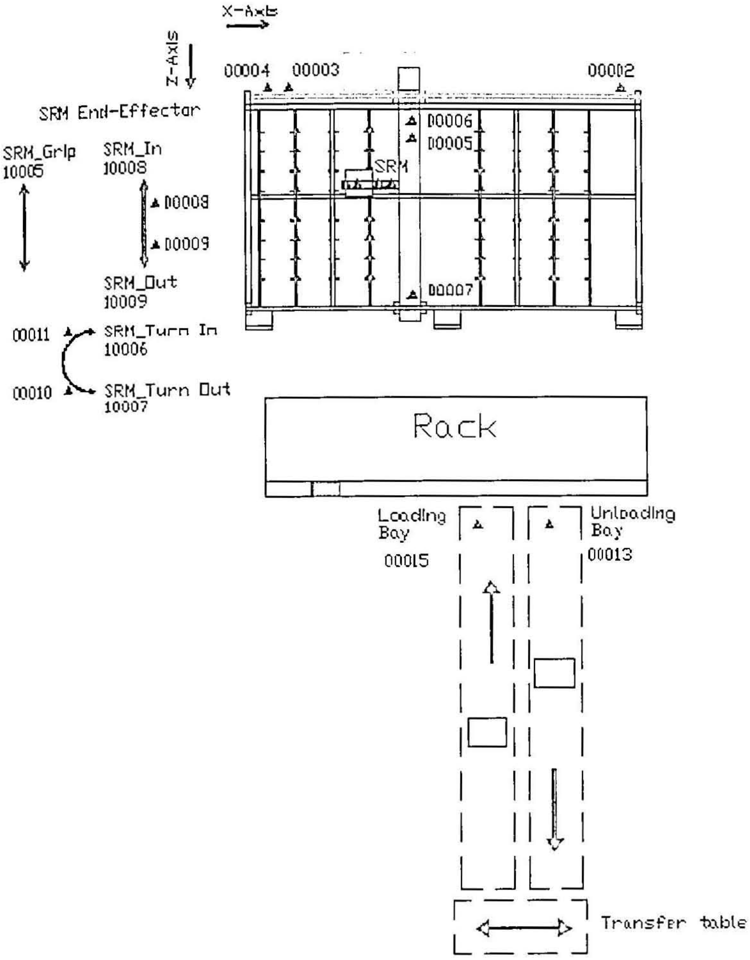
[15] Servo Motors and Servo Amplifiers Instruction Manual – MR-C, Mitsubishi Electric, Art. No.:127749, 2001 04 04, SH3167-C

[16] RCX40 Series Programming. Yamaha Motor Co., Ltd., 2004

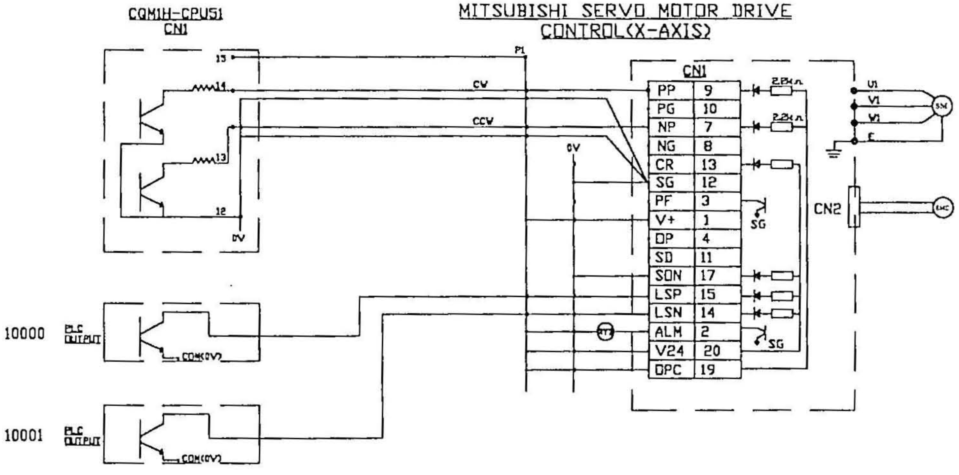
[17] Yamaha Scara Robots Yk-X Series Owner's Manual, Yamaha Motor Co., Ltd., May. 2004, Ver. 5.12

APPENDIX A

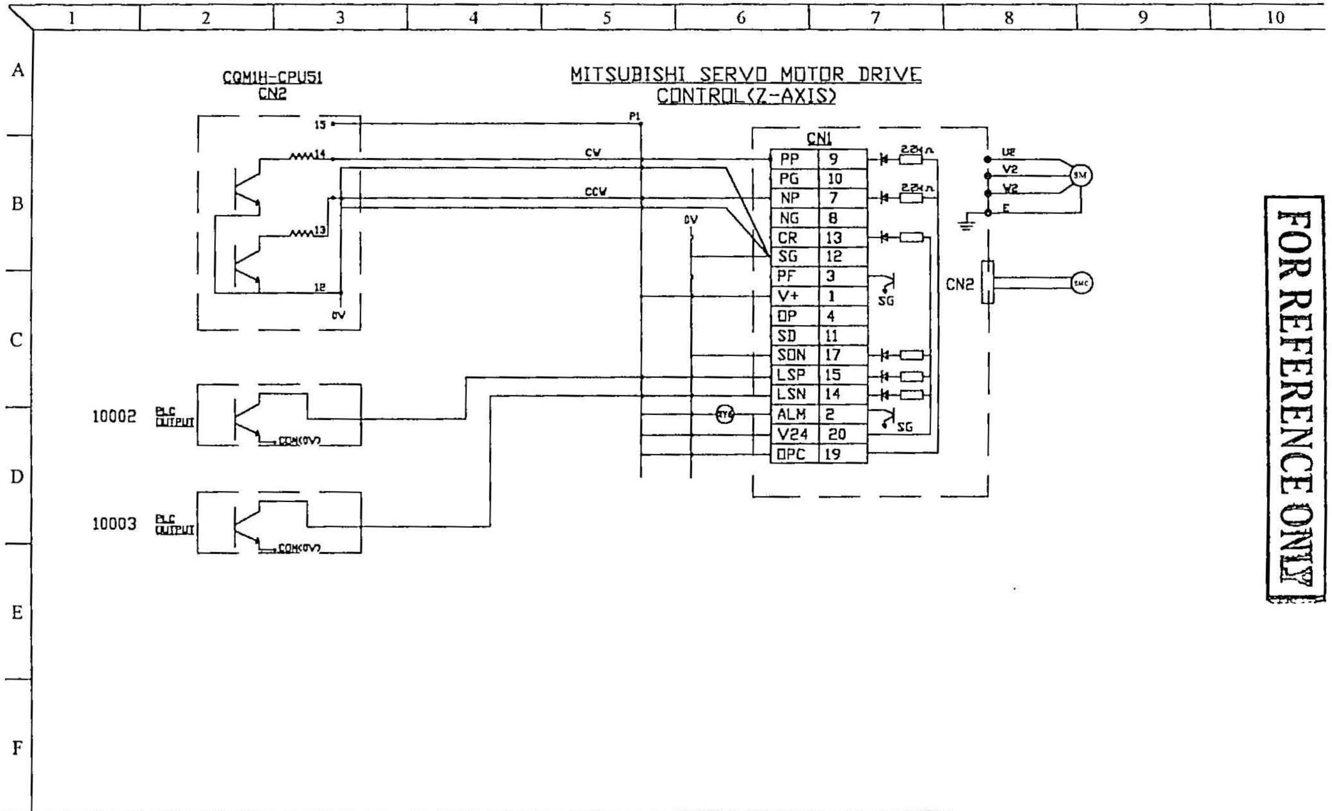
INPUT/OUTPUT ASSIGNMENT OF THE RACKING SYSTEM


X-Axis
Z-Axis
00004
00003
00002
SRM End-Effector
SRM_Grip
10005
SRM_In
10008
00008
00009
SRM_Out
10009
00011
SRM_Turn In
10006
00010
SRM_Turn Out
10007
00006
00005
SRM
00007
Rack
Loading Bay
00015
Unloading Bay
00013
Transfer table

MITSUBISHI SERVO MOTOR DRIVE CONTROL(X-AXIS)

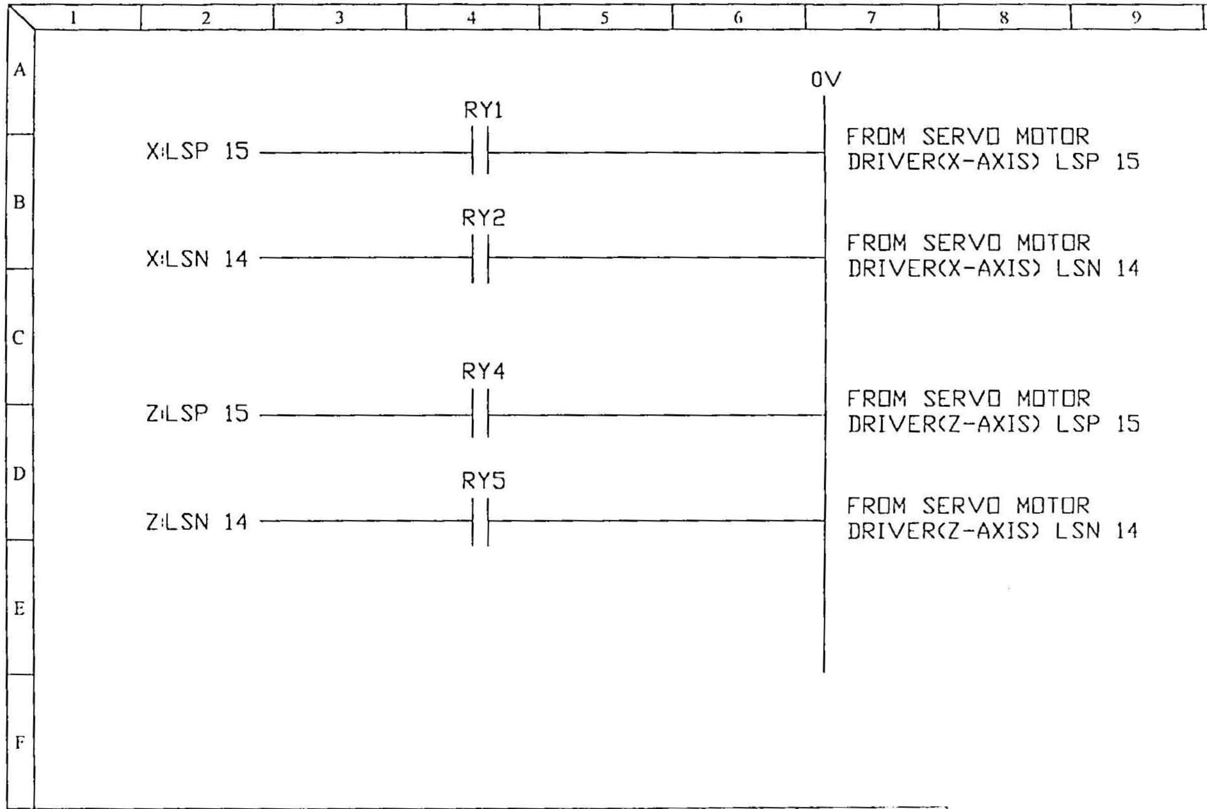


PAGE :	4	REV : 2	DRAWN BY :	YP YONG	CLIENT : KUITTHO		TITLE : AS/RS ROBOT-SERVO MOTOR DRIVE CONTROL (X-AXIS)
DATE :	10/5/2002		CHECKED BY :	DORIS			
JOB NO :	AFA3005		APPROVED BY :	Phua Kh 31 JUL 02 APPROVED	AFA	AFA SYSTEMS SDN BHD	



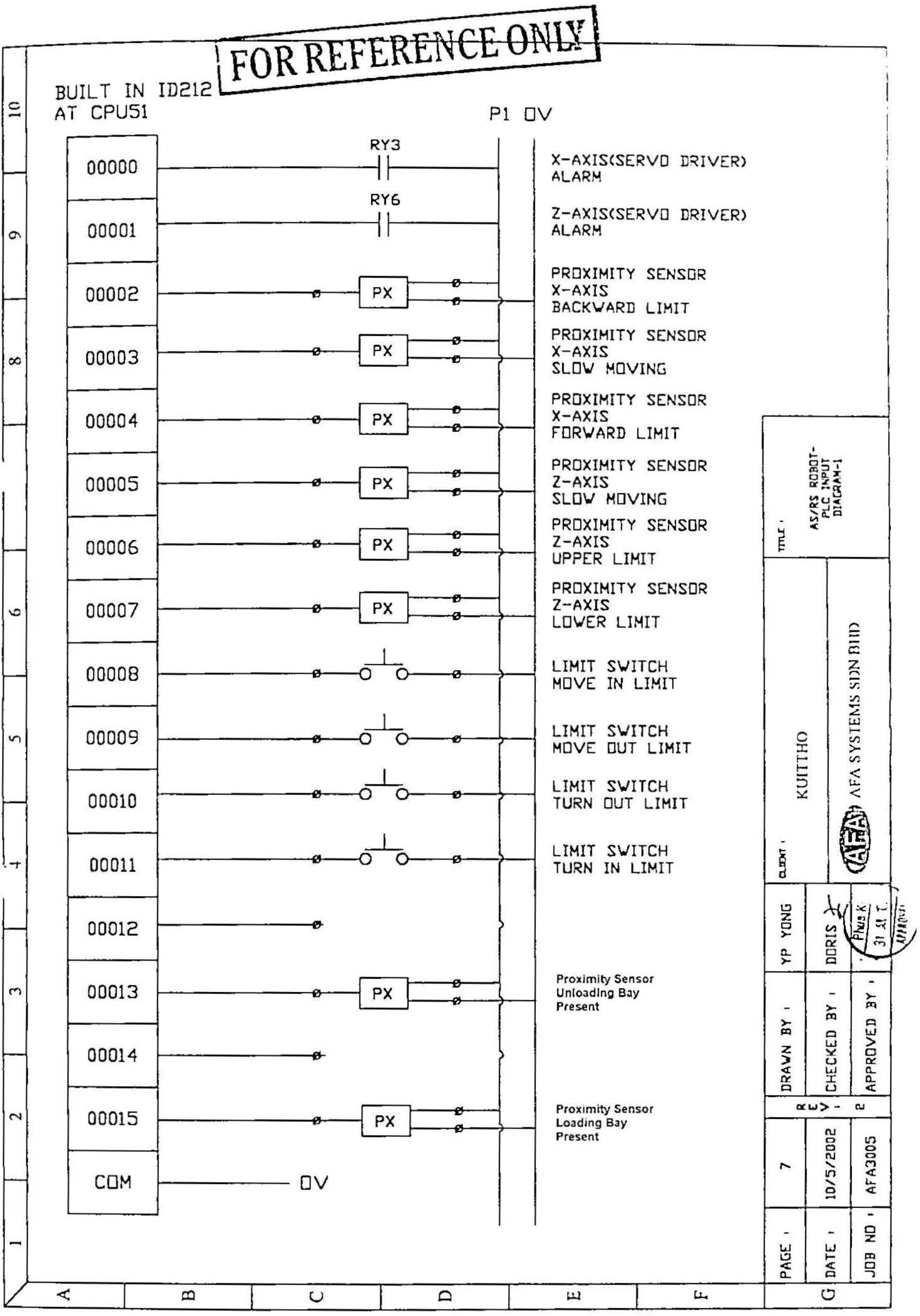
CN1	
PP	9
PG	10
NP	7
NG	8
CR	13
SG	12
PF	3
V+	1
OP	4
SD	11
SON	17
LSP	15
LSN	14
ALM	2
V24	20
OPC	19

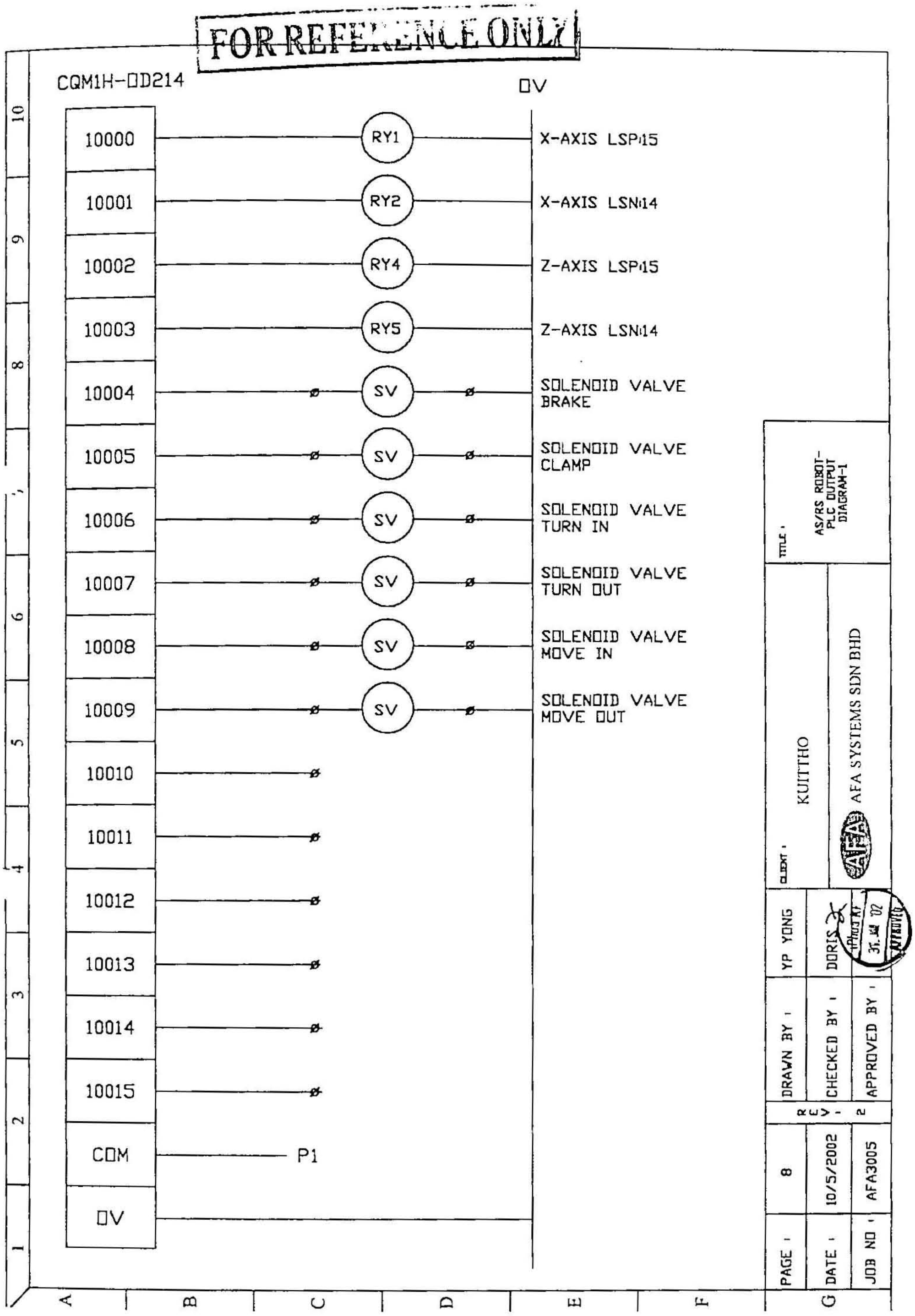
G	PAGE :	5	REV : 2	DRAWN BY :	YP YONG	CLIENT : KUITTHO	TITLE : AS/RS ROBOT-SERVO MOTOR DRIVE CONTROL (Z-AXIS)
	DATE :	10/5/2002		CHECKED BY :	DORIS	AFA SYSTEMS SDN BHD	
	JOB NO :	AFA3005		APPROVED BY :	Phua K. 31. JUL '02 APPROVED		


1
2
3
4
5
6
7
8
9
10
A
B
C
D
E
F
0V
RY1
X:LSP 15
FROM SERVO MOTOR DRIVER(X-AXIS) LSP 15
RY2
X:LSN 14
FROM SERVO MOTOR DRIVER(X-AXIS) LSN 14
RY4
Z:LSP 15
FROM SERVO MOTOR DRIVER(Z-AXIS) LSP 15
RY5
Z:LSN 14
FROM SERVO MOTOR DRIVER(Z-AXIS) LSN 14

G	PAGE :	6	REV : 2	DRAWN BY :	YP YONG	CLIENT : KUITTHO	TITLE AS/RS ROBOT RELAY CONTROL CIRCUIT DIAGRAM
	DATE :	10/5/2002		CHECKED BY :	DORIS	AFA SYSTEMS SDN BHD	
	JOB NO :	AFA3005		APPROVED BY :			

FOR REFERENCE ONLY


BUILT IN ID212
AT CPU51
P1 0V
00000
RY3
X-AXIS(SERVO DRIVER) ALARM
00001
RY6
Z-AXIS(SERVO DRIVER) ALARM
00002
PX
PROXIMITY SENSOR X-AXIS BACKWARD LIMIT
00003
PX
PROXIMITY SENSOR X-AXIS SLOW MOVING
00004
PX
PROXIMITY SENSOR X-AXIS FORWARD LIMIT
00005
PX
PROXIMITY SENSOR Z-AXIS SLOW MOVING
00006
PX
PROXIMITY SENSOR Z-AXIS UPPER LIMIT
00007
PX
PROXIMITY SENSOR Z-AXIS LOWER LIMIT
00008
LIMIT SWITCH MOVE IN LIMIT
00009
LIMIT SWITCH MOVE OUT LIMIT
00010
LIMIT SWITCH TURN OUT LIMIT
00011
LIMIT SWITCH TURN IN LIMIT
00012
00013
PX
Proximity Sensor Unloading Bay Present
00014
00015
PX
Proximity Sensor Loading Bay Present
COM
0V
TITLE : AS/RS ROBOT-PLC INPUT DIAGRAM-1
CLIENT : KUITTHO
AFA SYSTEMS SDN BHD
DRAWN BY : YP YONG
CHECKED BY : DORIS
APPROVED BY :
REV : 2
PAGE : 7
DATE : 10/5/2002
JOB NO : AFA3005
10
9
8
6
5
4
3
2
1
A
B
C
D
E
F
G



TITLE : AS/RS ROBOT-PLC OUTPUT DIAGRAM-1

CLIENT : KUITTHO

AFA SYSTEMS SDN BHD

DRAWN BY : YP YONG

CHECKED BY : DORIS

APPROVED BY :

REV : 2

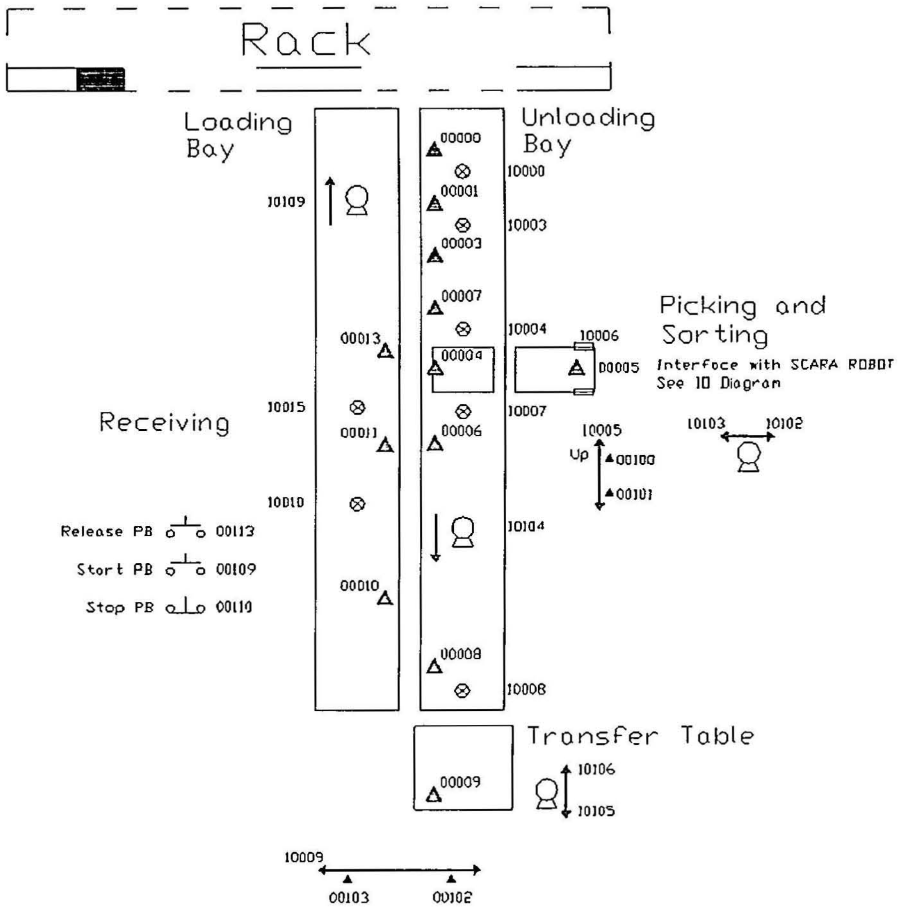
PAGE : 8

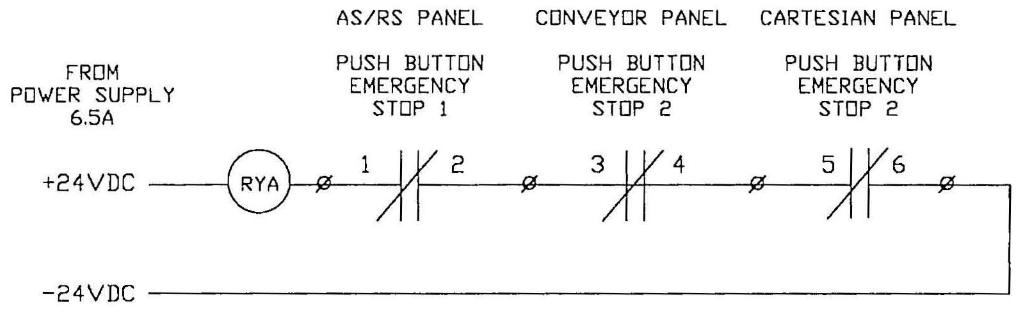
DATE : 10/5/2002

JOB NO : AFA3005

APPENDIX B

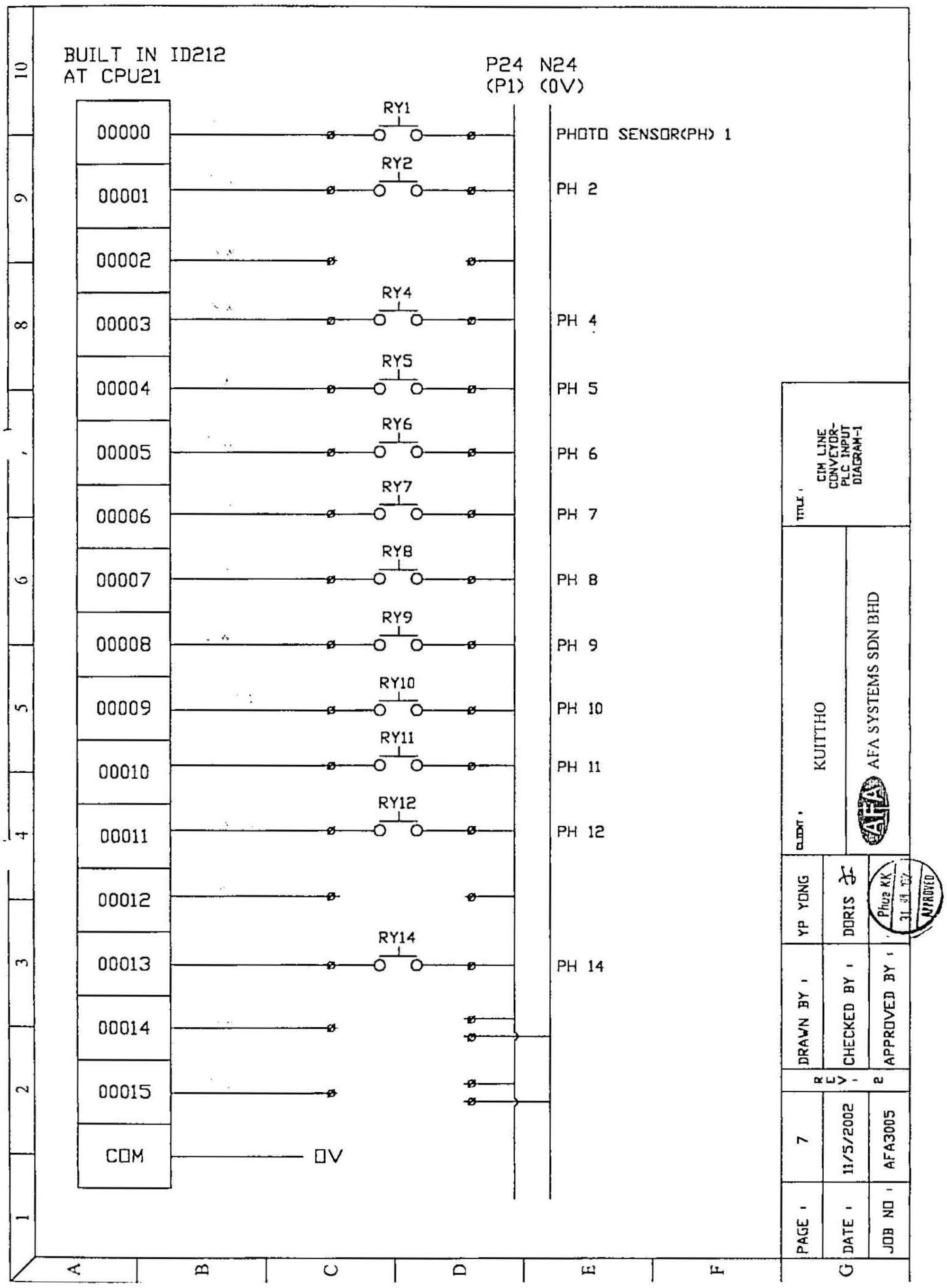
INPUT/OUTPUT ASSIGNMENT OF THE MHS SYSTEM

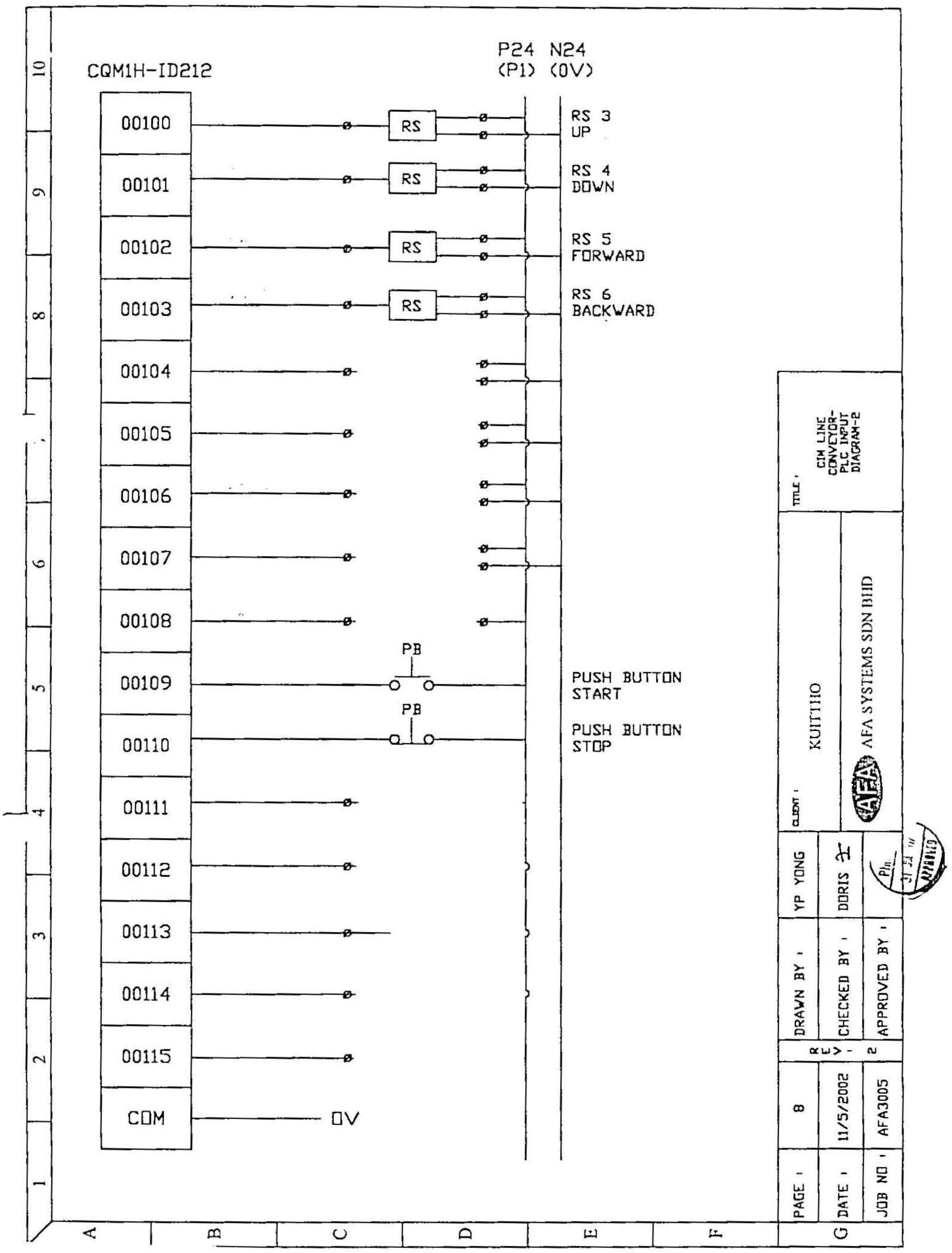

Rack
Loading Bay
Unloading Bay
I0109
00000
I0000
00001
I0003
00003
00007
I0004
I0006
Picking and Sorting
Interface with SCARA ROBOT
See IO Diagram
00013
00004
00005
I0015
Receiving
00011
I0007
00006
I0005
Up
00100
00101
I0103
I0102
I0010
Release PB
00113
Start PB
00109
Stop PB
00110
I0104
00010
00008
I0008
Transfer Table
00009
I0106
I0105
I0009
00103
00102

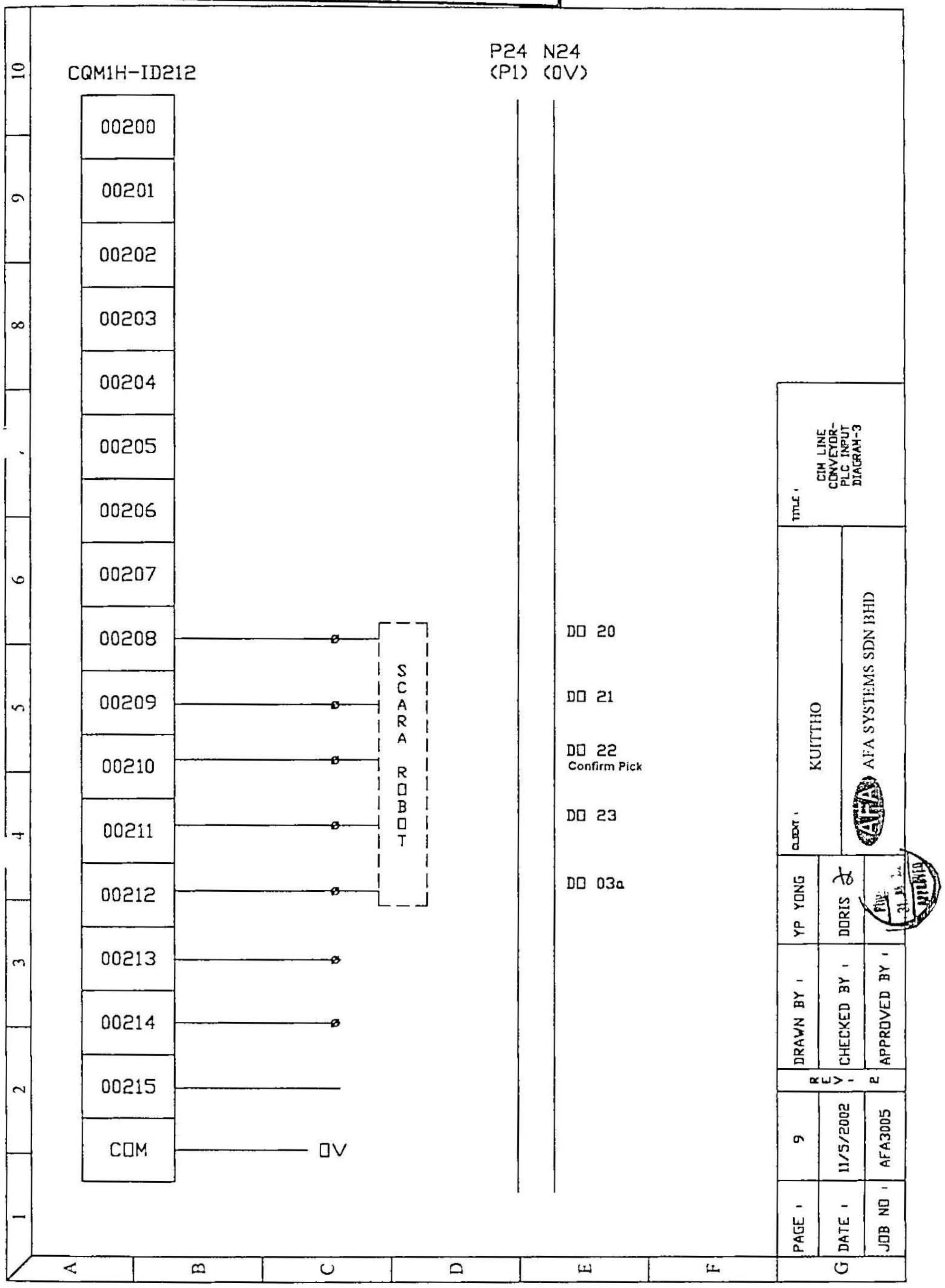

AS/RS PANEL
PUSH BUTTON EMERGENCY STOP 1
CONVEYOR PANEL
PUSH BUTTON EMERGENCY STOP 2
CARTESIAN PANEL
PUSH BUTTON EMERGENCY STOP 2
FROM POWER SUPPLY 6.5A
+24VDC
RYA
1
2
3
4
5
6
-24VDC

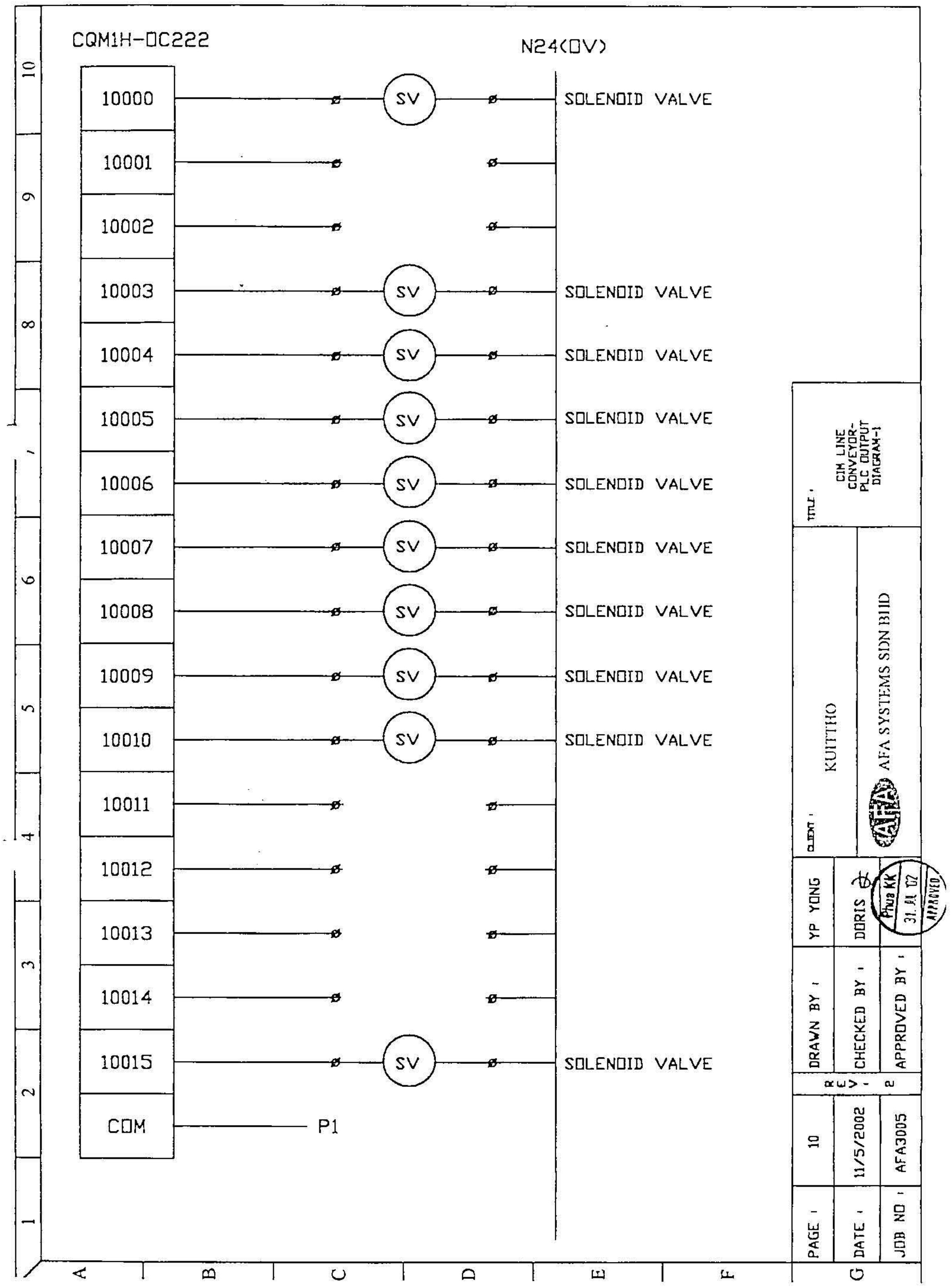
REFER PAGE 3

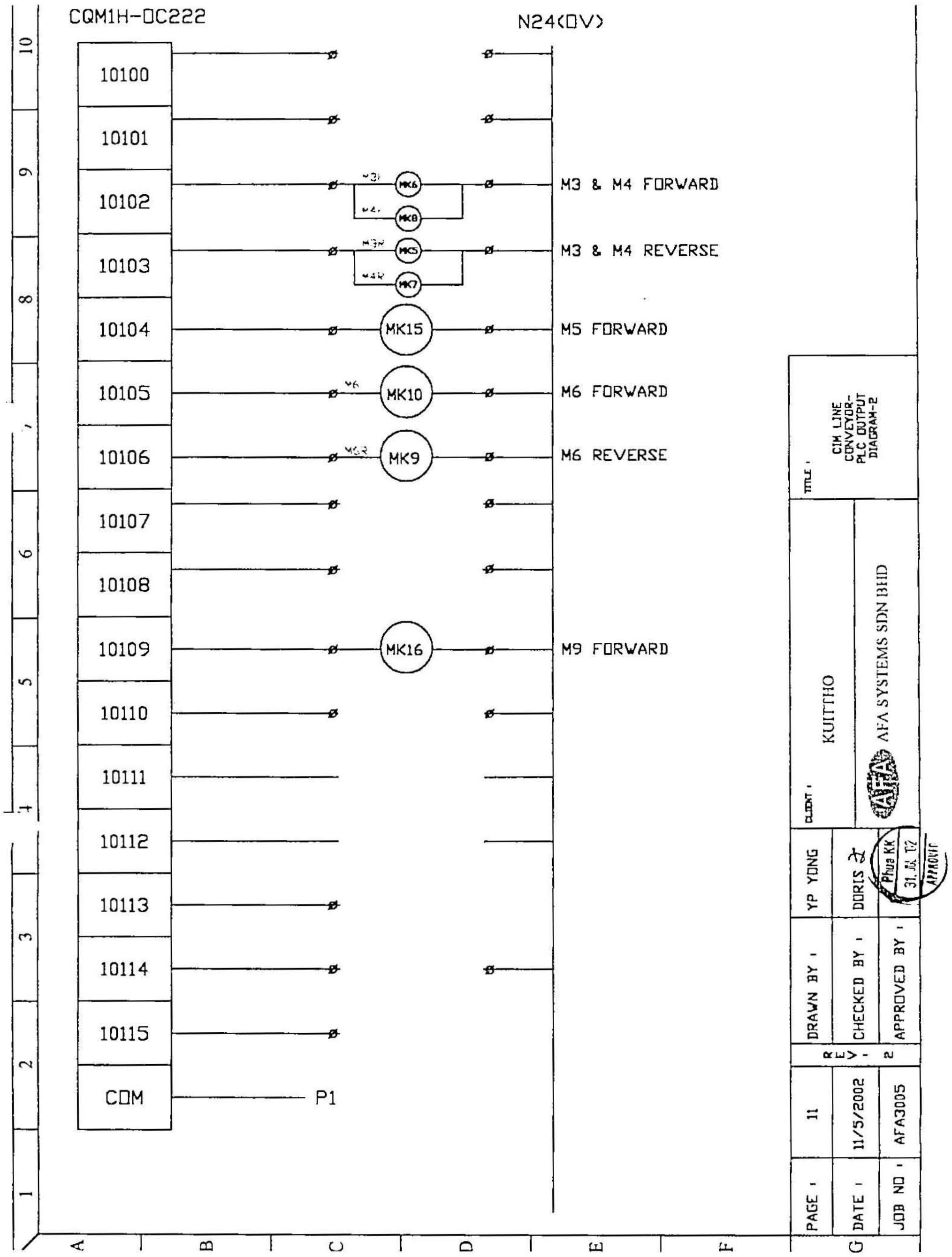
PAGE :	4	REV : 2	DRAWN BY :	YP YONG	CLIENT : KUITTHO	TITLE : CIM LINE CONVEYOR- POWER CONTROL CIRCUIT DIAGRAM 2 (EMERGENCY STOP)
DATE :	10/5/2002		CHECKED BY :	DORIS	AFA SYSTEMS SDN BHD	
JOB NO :	AFA3005		APPROVED BY :	Phua k 31 ... APPROVE		

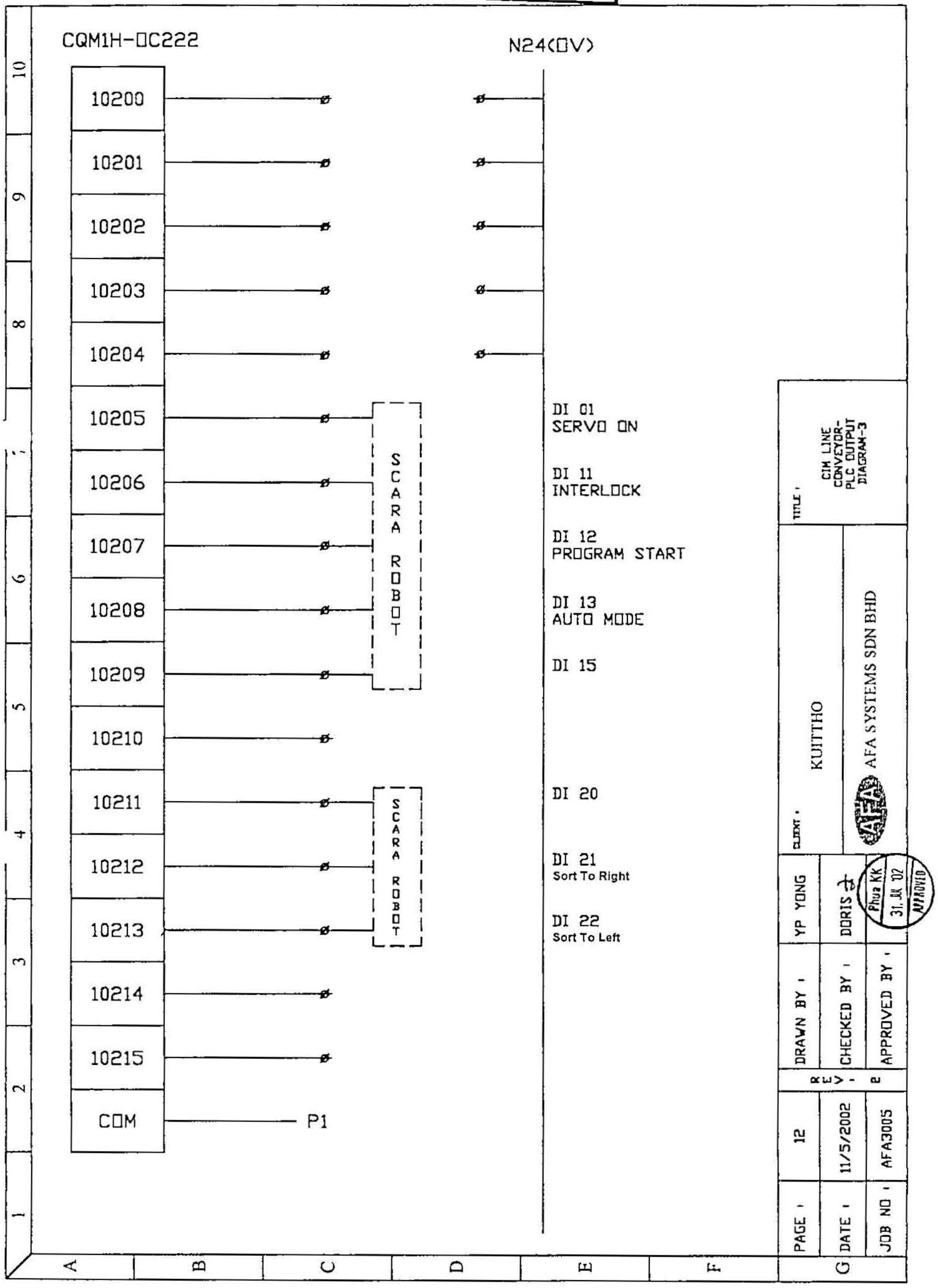

BUILT IN ID212
AT CPU21
P24 N24
(P1) (0V)
00000
RY1
PHOTO SENSOR(PH) 1
00001
RY2
PH 2
00002
00003
RY4
PH 4
00004
RY5
PH 5
00005
RY6
PH 6
00006
RY7
PH 7
00007
RY8
PH 8
00008
RY9
PH 9
00009
RY10
PH 10
00010
RY11
PH 11
00011
RY12
PH 12
00012
00013
RY14
PH 14
00014
00015
COM
0V
10
9
8
7
6
5
4
3
2
1
A
B
C
D
E
F
G
TITLE :
CIM LINE
CONVEYOR-
PLC INPUT
DIAGRAM-1
CLIENT :
KUITTHO
AFA
AFA SYSTEMS SDN BHD
DRAWN BY :
YP YONG
CHECKED BY :
DORIS
APPROVED BY :
Phua KK
31.05.02
APPROVED
REV :
B
PAGE :
7
DATE :
11/5/2002
JOB NO :
AFA3005


CQM1H-ID212
P24 (P1)
N24 (0V)
00100
00101
00102
00103
00104
00105
00106
00107
00108
00109
00110
00111
00112
00113
00114
00115
COM
0V
RS
RS 3 UP
RS 4 DOWN
RS 5 FORWARD
RS 6 BACKWARD
PB
PUSH BUTTON START
PUSH BUTTON STOP
TITLE : CIM LINE CONVEYOR-PLC INPUT DIAGRAM-2
CLIENT : KUITTHO
AFA SYSTEMS SDN BHD
DRAWN BY : YP YONG
CHECKED BY : DORIS
APPROVED BY :
REV : 2
PAGE : 8
DATE : 11/5/2002
JOB NO : AFA3005
A
B
C
D
E
F
G
1
2
3
4
5
6
7
8
9
10


CQM1H-ID212
P24 (P1)
N24 (0V)
00200
00201
00202
00203
00204
00205
00206
00207
00208
00209
00210
00211
00212
00213
00214
00215
COM
0V
SCARA ROBOT
DO 20
DO 21
DO 22
Confirm Pick
DO 23
DO 03a
10
9
8
7
6
5
4
3
2
1
A
B
C
D
E
F
G
TITLE :
CIM LINE CONVEYOR-PLC INPUT DIAGRAM-3
CLIENT :
KUITTHO
AFA
AFA SYSTEMS SDN BHD
DRAWN BY :
YP YONG
CHECKED BY :
DORIS
APPROVED BY :
PAGE :
9
DATE :
11/5/2002
JOB NO :
AFA3005
REV :
2


CQM1H-OC222
N24(0V)
10000
SV
SOLENOID VALVE
10001
10002
10003
SV
SOLENOID VALVE
10004
SV
SOLENOID VALVE
10005
SV
SOLENOID VALVE
10006
SV
SOLENOID VALVE
10007
SV
SOLENOID VALVE
10008
SV
SOLENOID VALVE
10009
SV
SOLENOID VALVE
10010
SV
SOLENOID VALVE
10011
10012
10013
10014
10015
SV
SOLENOID VALVE
COM
P1
1
2
3
4
5
6
7
8
9
10
A
B
C
D
E
F
G
TITLE :
CIM LINE CONVEYOR-PLC OUTPUT DIAGRAM-1
CLIENT :
KUITTHO
AFA SYSTEMS SDN BHD
DRAWN BY :
YP YONG
CHECKED BY :
DORIS
APPROVED BY :
31.5.02
APPROVED
REV :
2
PAGE :
10
DATE :
11/5/2002
JOB NO :
AFA3005


CQM1H-OC222
N24(0V)
10100
10101
10102
M3F
MK6
M4F
MK8
M3 & M4 FORWARD
10103
M3R
MK5
M4R
MK7
M3 & M4 REVERSE
10104
MK15
M5 FORWARD
10105
M6
MK10
M6 FORWARD
10106
M6R
MK9
M6 REVERSE
10107
10108
10109
MK16
M9 FORWARD
10110
10111
10112
10113
10114
10115
COM
P1
1
2
3
4
5
6
7
8
9
10
A
B
C
D
E
F
G
TITLE : CIM LINE CONVEYOR-PLC OUTPUT DIAGRAM-2
CLIENT : KUITTHO
AFA SYSTEMS SDN BHD
DRAWN BY : YP YONG
CHECKED BY : DORIS
APPROVED BY :
Phua KK
31 JUL '02
APPROVE
PAGE : 11
DATE : 11/5/2002
JOB NO : AFA3005
REV : 2


CQM1H-OC222
N24(0V)
10200
10201
10202
10203
10204
10205
10206
10207
10208
10209
10210
10211
10212
10213
10214
10215
COM
P1
SCARA ROBOT
SCARA ROBOT
DI 01
SERVO ON
DI 11
INTERLOCK
DI 12
PROGRAM START
DI 13
AUTO MODE
DI 15
DI 20
DI 21
Sort To Right
DI 22
Sort To Left
10
9
8
7
6
5
4
3
2
1
A
B
C
D
E
F
G
TITLE :
CIM LINE
CONVEYOR-
PLC OUTPUT
DIAGRAM-3
CLIENT :
KUITTHO
AFA SYSTEMS SDN BHD
DRAWN BY :
YP YONG
CHECKED BY :
DORIS
APPROVED BY :
Phua KK
31. JUL '02
APPROVED
PAGE :
12
DATE :
11/5/2002
JOB NO :
AFA3005
REV :
0

APPENDIX C

FINSGATEWAY (FGW) SETUP PROCEDURE

Establishing Communications

To perform communications with FinsGateway, it is necessary to set the communication unit(s), and start the communication service(s).

The following are the type of communication available in FGW.

Setting Up Serial Communications

Setting Up Ethernet

Setting Up Controller Link

Setting Up Controller Link (PCI)

Setting Up SYSMAC LINK

Setting Up SYSMAC LINK (PCI)

Setting Up a SYSMAC Board

CS1 Bus Interface Setup

Setting Up Serial Communications

FinsGateway serial communications use an RS-232C, or other serial line to communicate with a PLC or another device.

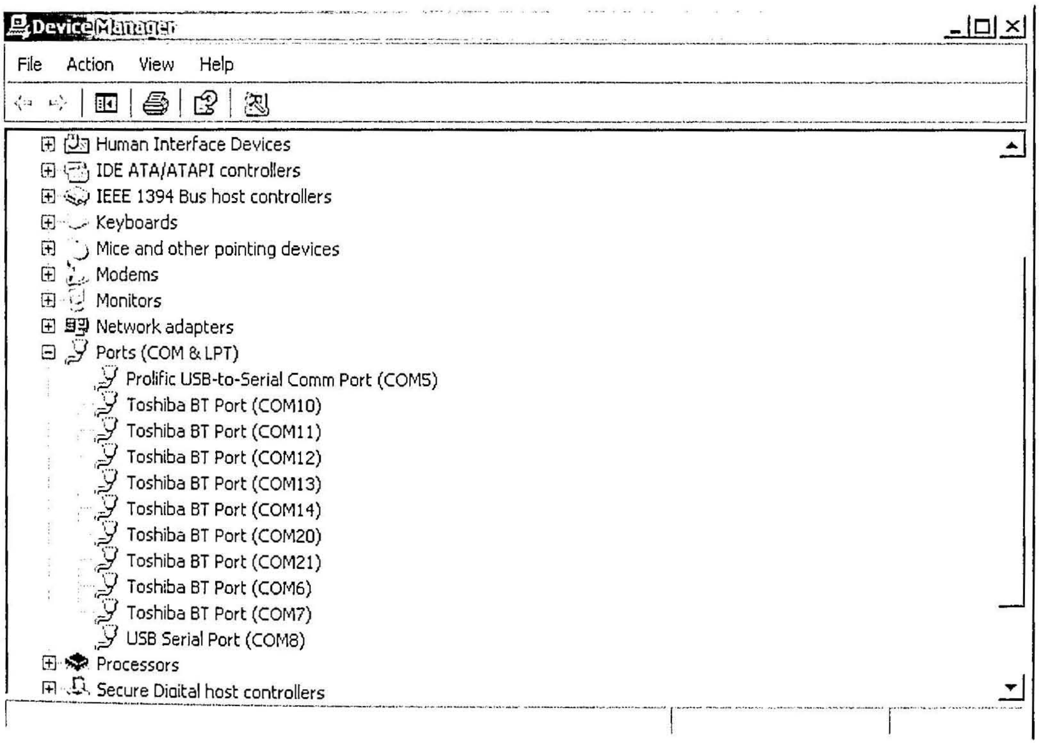
FinsGateway serial communications already provide the following protocols:

Protocol	Description
SYSWAY	Supported by the SYSMAC C-series and CV-series PLCs; the transfer command character string is @00RD, etc. This protocol is often called the Host Link, C-mode protocol.
SYSWAY-CV	Supported by the SYSMAC CV-series PLCs; the transfer command character string is @00FA, etc. This protocol is often called the Host Link, FINS protocol.
CompoWay/F	Used in many Omron control devices.
Toolbus-C	Used to communicate with SYSMAC C-series PLCs through the peripheral port. For setting details, refer to the \Online Document\English\ToolbusManual_E.pdf file on the distribution CD-ROM.

Toolbus-CV	Used to communicate with SYSMAC CV-series PLCs through the peripheral port. For setting details, refer to the \Online Document\English\ToolbusManual_E.pdf file on the distribution CD-ROM.
ToolbusCS/CJ	Used to communicate with SYSMAC CS/CJ-series PLCs through the peripheral port. For setting details, refer to the \Online Document\English\ToolbusCS1Manual_E.pdf file on the distribution CD-ROM.

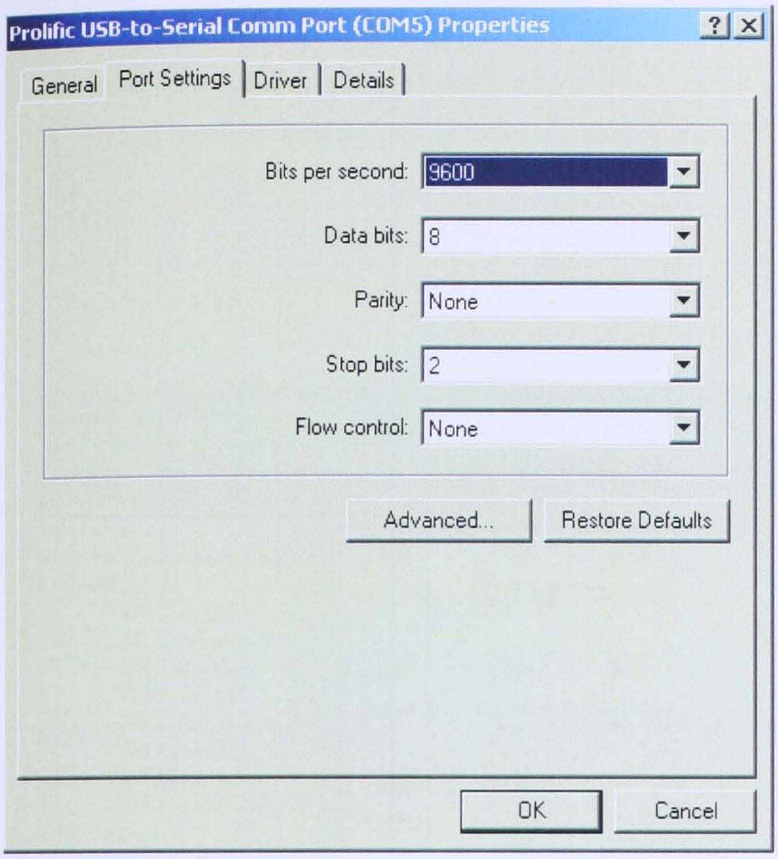
Serial communications use the serial port (COM port) of the computer. If the computer already has a serial port, there is no need to purchase a separate board.

The COM ports available to the computer can be confirmed in the Windows Device Manager, under Ports (COM & LPT).



```
⊞ Human Interface Devices
⊞ IDE ATA/ATAPI controllers
⊞ IEEE 1394 Bus host controllers
⊞ Keyboards
⊞ Mice and other pointing devices
⊞ Modems
⊞ Monitors
⊞ Network adapters
⊟ Ports (COM & LPT)
    Prolific USB-to-Serial Comm Port (COM5)
    Toshiba BT Port (COM10)
    Toshiba BT Port (COM11)
    Toshiba BT Port (COM12)
    Toshiba BT Port (COM13)
    Toshiba BT Port (COM14)
    Toshiba BT Port (COM20)
    Toshiba BT Port (COM21)
    Toshiba BT Port (COM6)
    Toshiba BT Port (COM7)
    USB Serial Port (COM8)
⊞ Processors
⊞ Secure Digital host controllers
```

Right click the selected COM port to configure the property.



Bits per second: 9600

Data bits: 8

Parity: None

Stop bits: 2

Flow control: None

Sample: Setup procedure for Serial communication use the serial port (COM ports) of the computer via a USB-Serial Converter.

This sample applies one COM port. Use a USB-Serial-Converter.

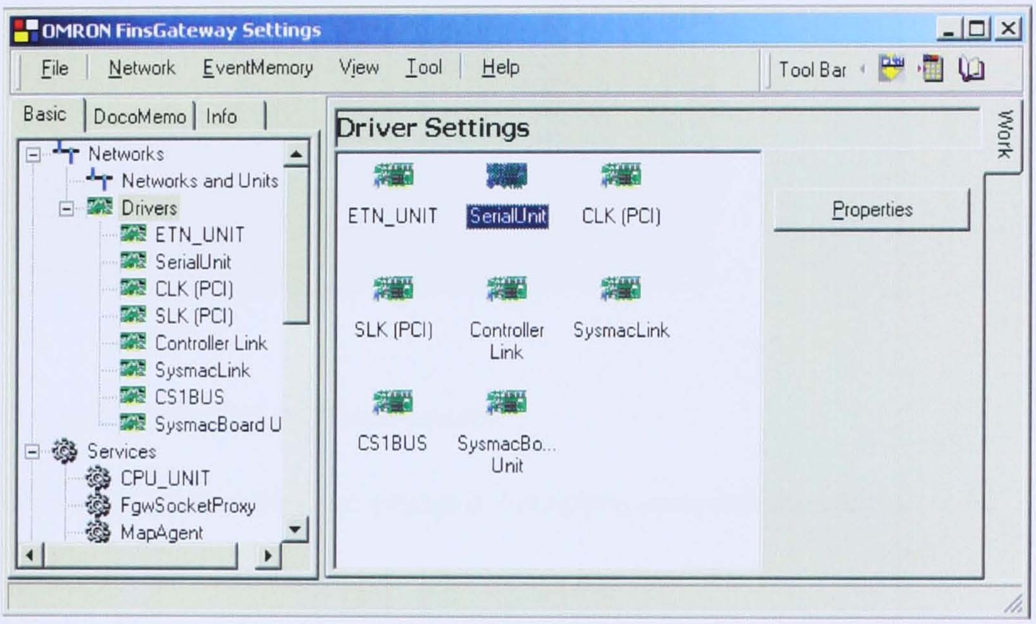
In Device Manager, try to configure the port accordingly.
(More detail on setting up in FinsGateway help file)

Specifying the Communications Path

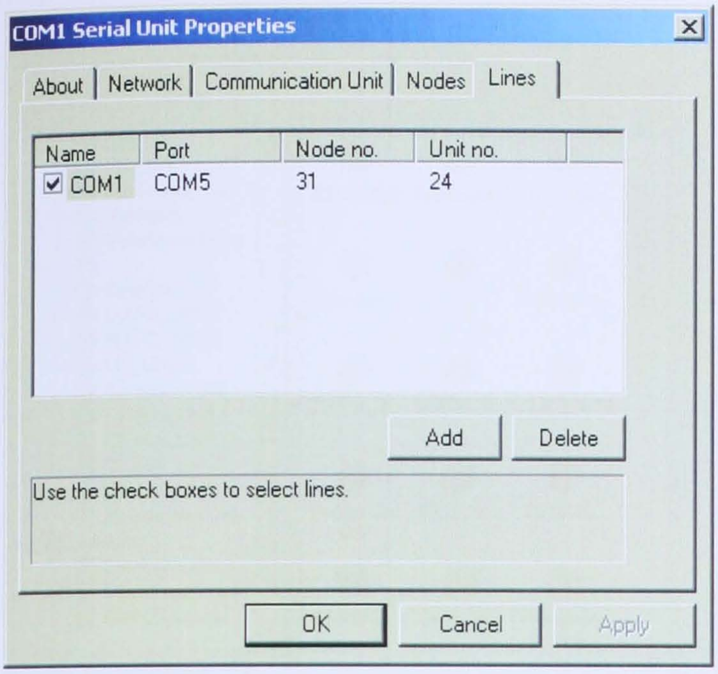
FinsGateway serial communications handle the COM port as a network.
FinsGateway must be set to use a specific COM port. More than one COM port can be set.

Follow the steps below:

1. In the FinsGateway Configuration Basic tab, select the Drivers icon.
2. Select the SerialUnit icon in the display.
3. Select the Communication Lines tab, and check the COM ports to use.



Click SerialUnit and Properties.



Name	Port	Node no.	Unit no.
☑ COM1	COM5	31	24

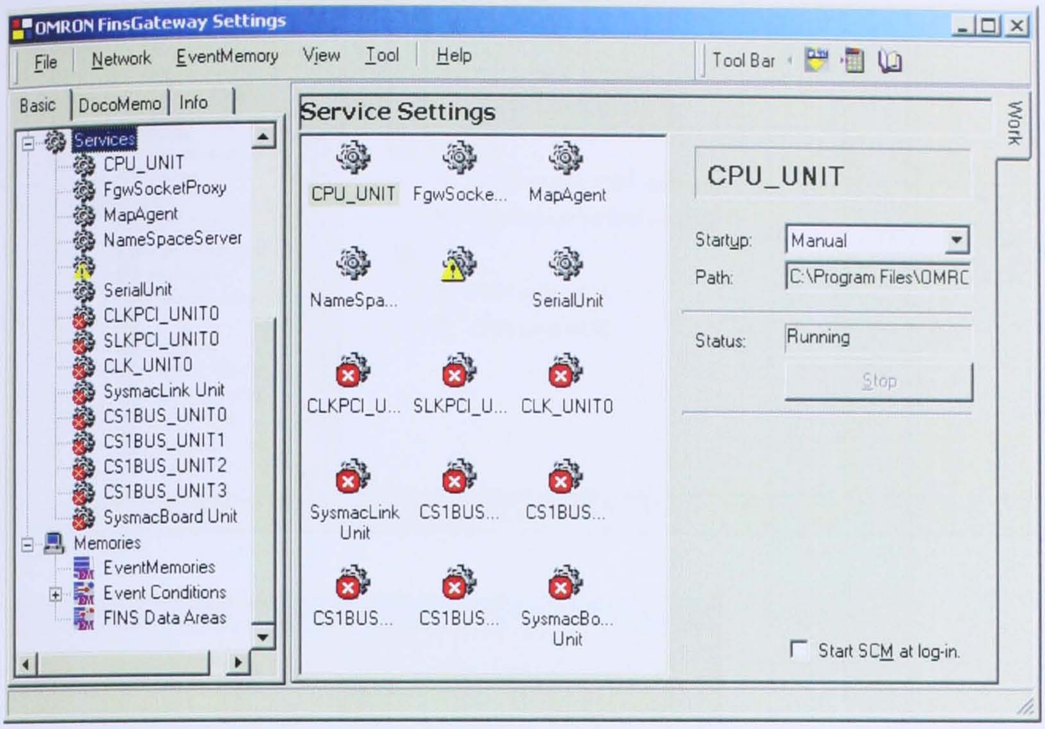
Use the check boxes to select lines.

Starting/Stopping Serial Communications

After the communications line settings are complete, serial communications can be started.

Start serial communications as follows:

1. In the FinsGateway Configuration Basic tab, select the Services icon.
2. Select the SerialUnit icon in the display.
3. Click Start.



CPU_UNIT

Startup: Manual

Path: C:\Program Files\OMRC

Status: Running

□ Start SCM at log-in.

To stop the serial communications, use the following procedure:

In the FinsGateway Configuration Basic tab, select the Services icon.

Select the SerialUnit icon in the display.

Click Stop.

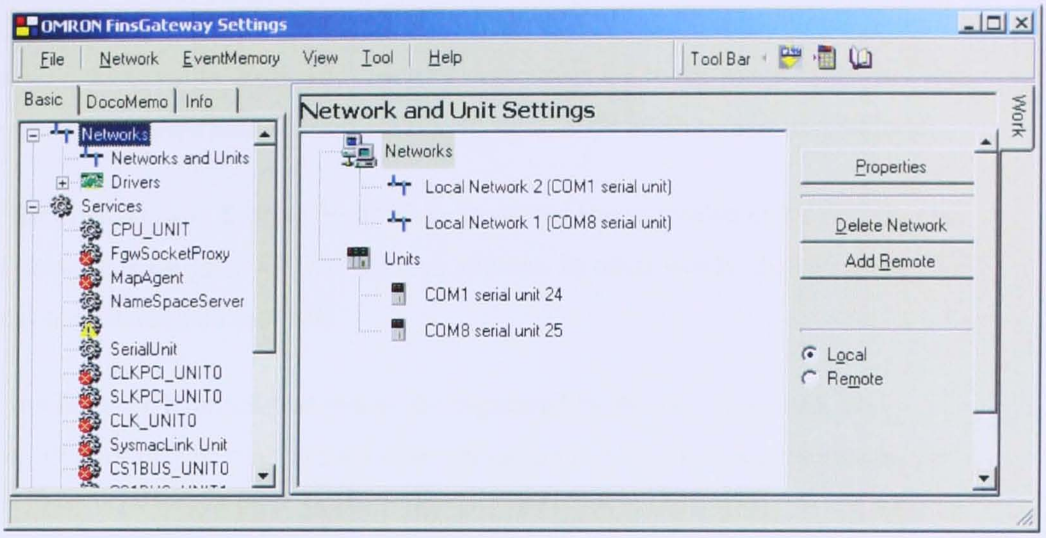
Serial Communications Settings

Before setting the network, record the following information:

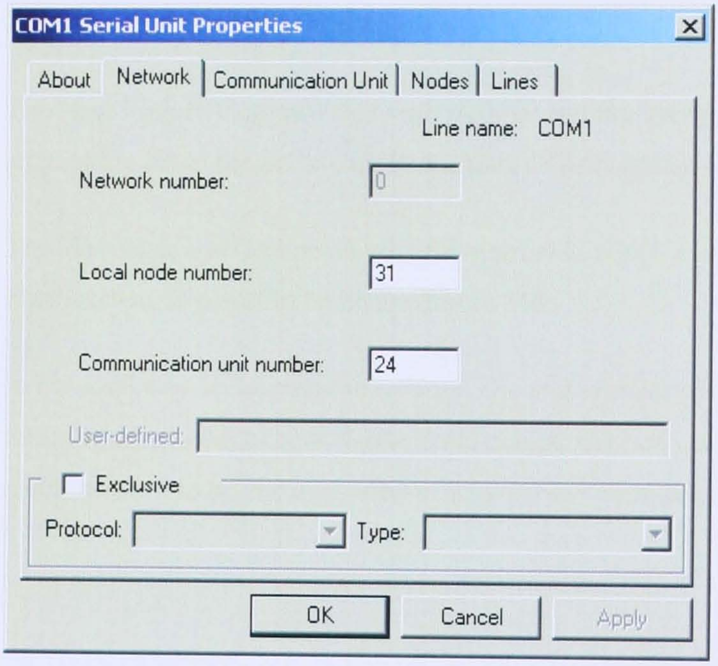
- The COM ports available to the system (COM5 etc.)
- Communication conditions of the target device: transfer speed, data length, stop bit setting, parity, etc.
- Protocol and unit number of the target device

To change the settings for serial communications that are already running, use the following procedure:

1. Select the Networks and Units icon in the Basic tab of the control pane.
2. Select the SerialUnit-COMx icon in the display.
3. Click Properties. The following dialog will be displayed:



● Local
○ Remote



Line name: COM1

Network number: 0

Local node number: 31

Communication unit number: 24

User-defined:

□ Exclusive

Protocol:

Type:

Serial Communications and FINS Network

The communication functions that FinsGateway provides are based on the FINS network model. This section explains how FinsGateway communicates with serial devices based on the FINS network model.

Unit Number and FINS Network

For Host Link (SYSMAC WAY, SYSMAC WAY-CV) and CompoWay/F, a unit number is assigned to distinguish each device connected to the serial cable. In FINS, communications are handled by network address, node address, and unit address.

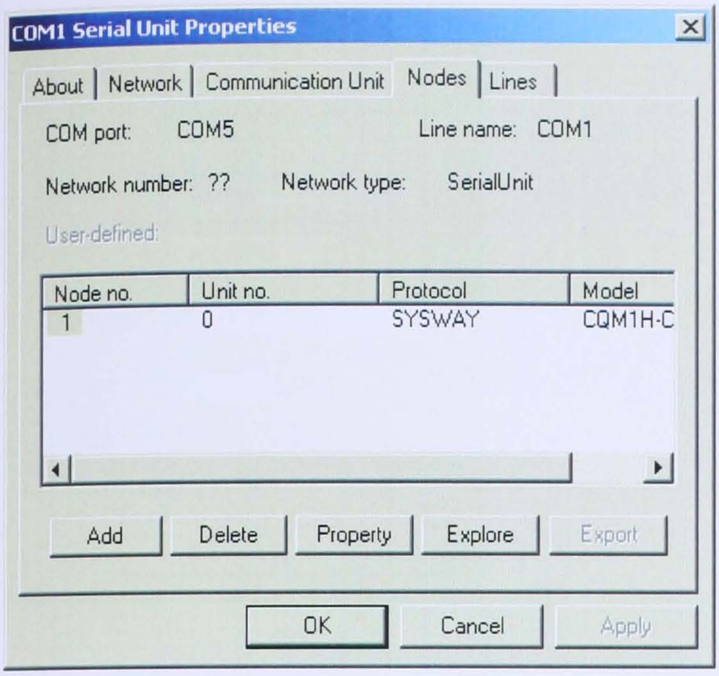
For the FinsGateway SysmacBoard Unit, the serial line is treated as a network, and that network is assigned a FINS network address. In other words, the computer COM port is considered one network.

In FinsGateway, a device that can be distinguished by its own Host Link or CompoWay/F unit number is treated as a node in the FINS network. However, the Host Link or CompoWay/F unit number, and the FINS node address do not have to be the same. The Host Link or CompoWay/F unit number and the FINS node address can be set independently of each other.

The Host Link or CompoWay/F unit number and the corresponding FINS node address are set in the SerialUnit Properties, COMx dialog, in the Node tab.

The Host Link and CompoWay/F unit number 0, which cannot be used in FinsGateway is preset to be node address 240.

In FinsGateway serial communications, the unit number of the FINS address has no meaning. However, to be consistent and compatible with other networks, it is recommended to set the unit number 0 for serial communications.



COM port: COM5

Line name: COM1

Network number: ??

Network type: SerialUnit

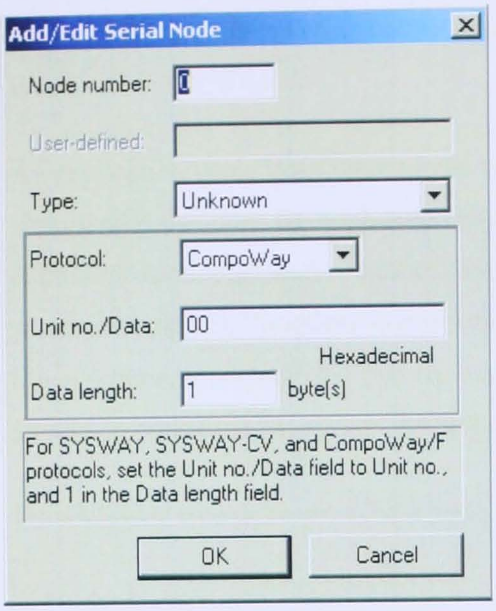
User-defined:

Node no.	Unit no.	Protocol	Model
1	0	SYSWAY	CQM1H-C

Adding a Node

Use the following procedure to add a node to the serial communication network:

1. Click the Node tab in the SerialUnit Properties dialog.
2. Click the Add button. The following dialog will be displayed:
3. Enter the FINS node address into the Node Number field of the Add/Edit Node (Serial) dialog.
4. Select the target device from the Model Name list box. To use CompoWay/F, set the Model Name to Other.
5. If the Protocol is set to SYSMAC WAY, SYSMAC WAY-CV, or CompoWay, specify the Host Link or CompoWay/F unit number for Unit No./Data. For Data Size, set 1. This is where the FINS node address and Host Link unit number relationship is set.



Add/Edit Serial Node

Node number: 0

User-defined:

Type: Unknown

Protocol: CompoWay

Unit no./Data: 00

Hexadecimal

Data length: 1 byte(s)

For SYSWAY, SYSWAY-CV, and CompoWay/F protocols, set the Unit no./Data field to Unit no., and 1 in the Data length field.

Node Communications Protocol

At present, serial communication protocols are many and varied, depending on the device.

FinsGateway is designed with the intention of enabling various protocols to be used simultaneously. This requires that for the FinsGateway communications settings, each node (device) have its own protocol setting.

Message Transfers

Differing from SYSMAC LINK, etc., the same serial communications cable (physical line) cannot carry communications for various devices simultaneously. The way communications for Host Link and CompoWay/F are performed is that the master first sends a command to a specific slave. The slave then sends the response for that command back to the master.

FinsGateway provides the master function of this arrangement. Therefore, the serial communications must always be initiated by an application using FinsGateway. The slave device then responds to the application through FinsGateway, and one communication set is completed. From the time that the command is sent, until the response is received, no other communications can take place on that same physical line.

An application using FinsGateway can execute multiple serial communications simultaneously (from the application user's perspective), but internally each communication must wait to be executed in order, one at a time. This means that when one command/response combination takes a little longer, it can adversely affect other communications that are waiting to use that same physical line. This should be considered in system development.

FinsGateway cannot be set to be a slave.

Transfer Speed

Serial communications are much slower than SYSMAC LINK, or other networks. Be sure to consider the following points when designing the system:

- The FinsGateway timeout value must be determined with consideration for the number of data that will typically be sent.
- Consider the length of the FINS messages for each protocol.
  The protocols supported by the SerialUnit (SYSWAY, SYSWAY-CV, CompoWay/F) all convert the FINS messages sent by the applications to the various protocols. The maximum FINS message length that can be converted is 1000 bytes.

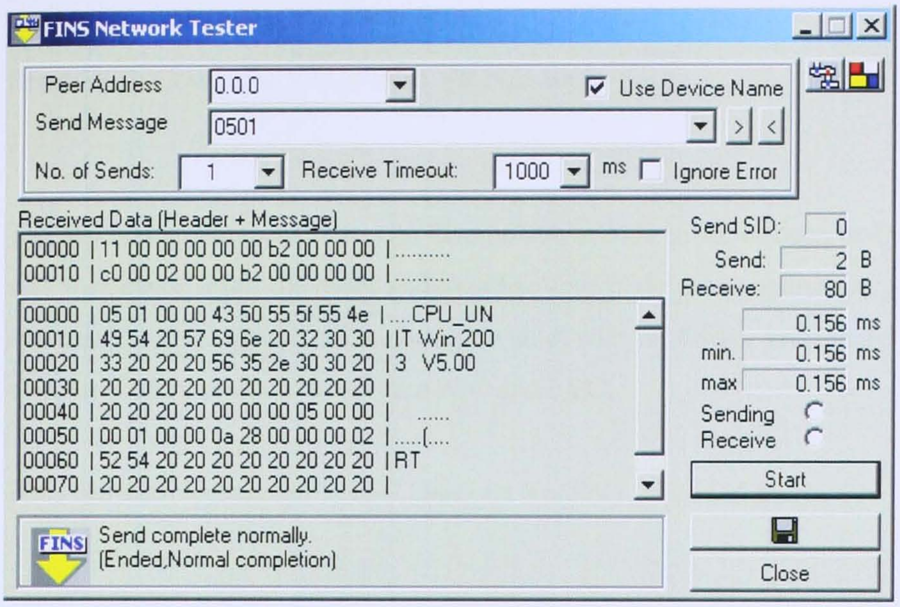
Device Communication Confirmation

The serial unit setup above has the address: 0.1.0 (Network.node.unit)
Notice that COM port is Network, node is node setup, unit is the PLC unit number.
Step 1: Start the ETN_UNIT.

1. From the FinsGateway Configuration Basic tab, click the Services icon. Click the ETN_UNIT icon in the display.
2. Click the Start button.

Step 2: Confirm that the application can communicate with the FinsGateway CPU_UNIT.

1. Start the FINS Network Tester program, which is included in FinsGateway as a general communications program. Select the FinsGateway Configuration program Network menu | FINS Network Tester option.
2. Enter CPU_UNIT for the Target Device. The CPU_UNIT is the default device prepared by the FinsGateway setup. For details about the devices, refer to the FinsGateway Configuration help.
3. Click the send button.
4. The FinsGateway CPU_UNIT data, and a message indicating normal completion should be displayed in the Receive Data display area. If that message is displayed, the communication was successful.



Peer Address: 0.0.0
☑ Use Device Name
Send Message: 0501
No. of Sends: 1
Receive Timeout: 1000 ms
□ Ignore Error

Received Data (Header + Message)

```
00000  | 11 00 00 00 00 00 b2 00 00 00  | ..........
00010  | c0 00 02 00 00 b2 00 00 00 00  | ..........
```

```
00000  | 05 01 00 00 43 50 55 5f 55 4e  | ....CPU_UN
00010  | 49 54 20 57 69 6e 20 32 30 30  | IT Win 200
00020  | 33 20 20 20 56 35 2e 30 30 20  | 3   V5.00
00030  | 20 20 20 20 20 20 20 20 20 20  |
00040  | 20 20 20 20 00 00 00 05 00 00  |     ......
00050  | 00 01 00 00 0a 28 00 00 00 02  | .....(....
00060  | 52 54 20 20 20 20 20 20 20 20  | RT
00070  | 20 20 20 20 20 20 20 20 20 20  |
```

Send SID: 0
Send: 2 B
Receive: 80 B
0.156 ms
min. 0.156 ms
max 0.156 ms
Sending ○
Receive ○

Send complete normally.
(Ended,Normal completion)

Troubleshooting

If the communication fails, check the following:

- Installation failure. If an installation failure warning is displayed at installation, re-install FinsGateway. When re-installing, always uninstall, first. Reboot the system. Then re-install.

Step 3: Confirm that the application can communicate with the FinsGateway communication unit (SerialUnit).

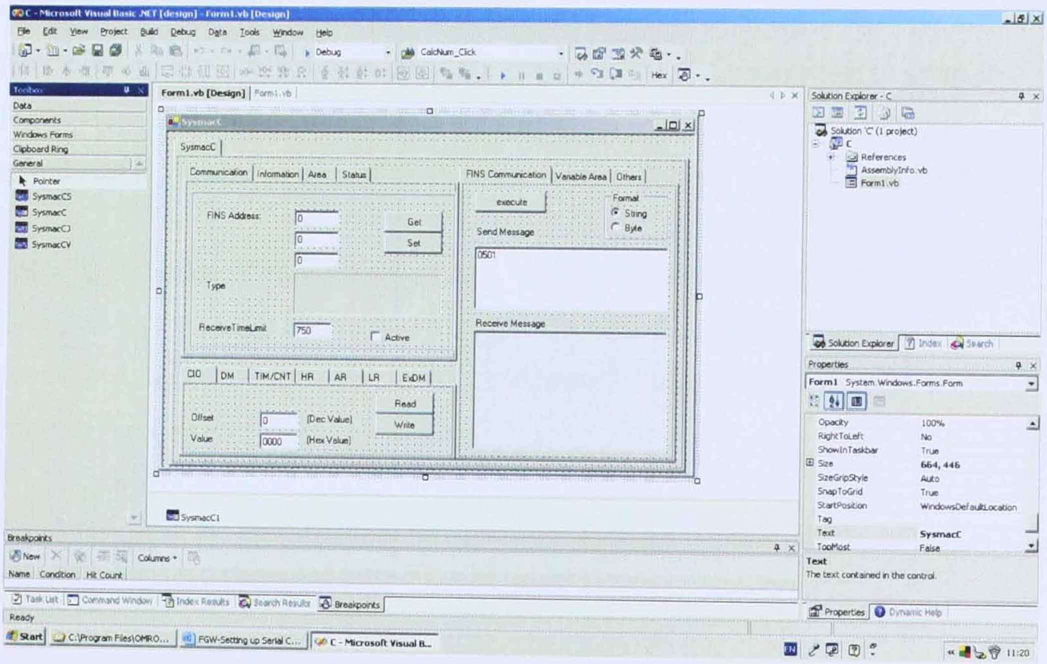
1. Enter 0.1.0 or the Target Device.
2. Click the Send button
3. The FinsGateway CPU_UNIT data, and a message indicating normal completion should be displayed in the Receive Data display area. If that message is displayed, the communication was successful.

Start With Compolet Ver2003 In VB.Net 2003

The VB.Net applet must start with Compolet Ver2003 in the background, with by itself is an class. Insert the class, and together with FGW setup before, the programmer can easily use the object and method provided, without further understand the communication protocol between a PLC and a PC.

Here is a sample attached in the Compolet Ver2003 , founded inside:

"C:\Program Files\Omron\FinsServerNT\Sample\dotnet\Compolet\Sysmac\SysmacC\C.sln"



FINS Address: 0 0 0

ReceiveTimeLimit 750 □ Active

Offset 0 (Dec Value)

Value 0000 (Hex Value)

Format ● String ○ Byte

Send Message 0501

Properties

Form1 System.Windows.Forms.Form

Opacity	100%
RightToLeft	No
ShowInTaskbar	True
⊞ Size	664, 446
SizeGripStyle	Auto
SnapToGrid	True
StartPosition	WindowsDefaultLocation
Tag	
Text	SysmacC
TopMost	False

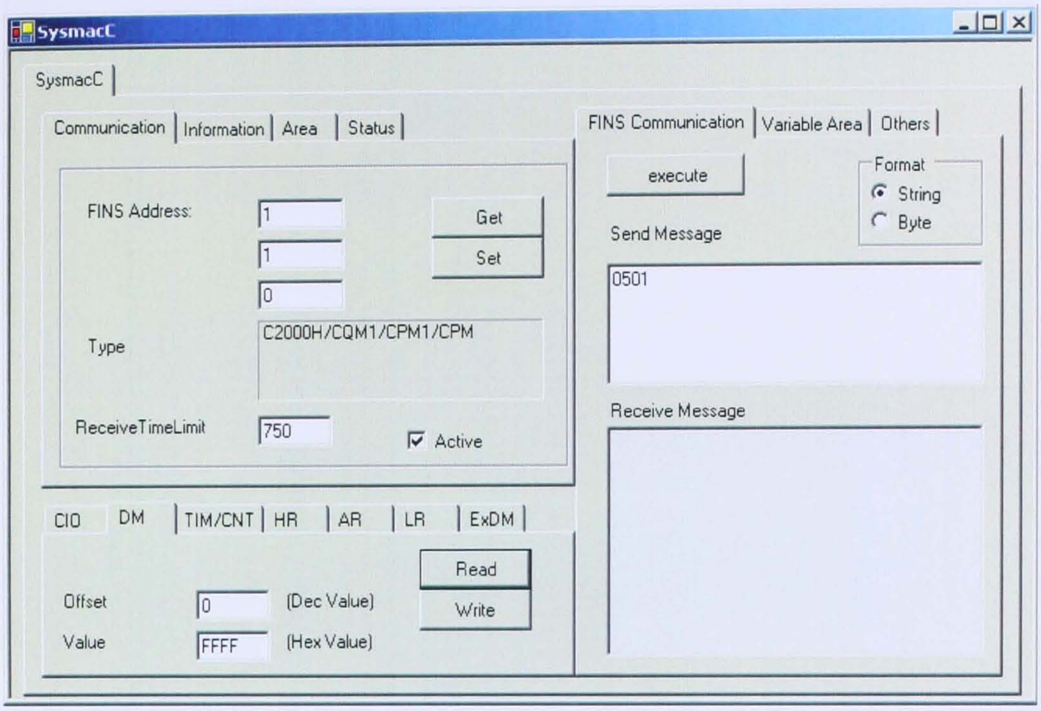
Text
The text contained in the control.

Double click the file. It should look like the figure shown above.

Notice that,a class object should be drag and drop in the form, such as shown.

Each class object corresponding to the PLC series used.

Once the FGW properly set, run the program. It should look like this:



FINS Address: 1 1 0

Type C2000H/CQM1/CPM1/CPM

ReceiveTimeLimit 750

☑ Active

Offset 0 (Dec Value)

Value FFFF (Hex Value)

Format ● String ○ Byte

Send Message 0501

In ‘Communication” tab, key in the FINS address of the PLC, according to setup in FGW. Here, the FINS adddress is 1.1.0. Then press “Set” to write the address to acknowledge which PLC is on-line. And, check box “Active” must be clicked ON.

Press “Get” button. A type display shown. “C200H/CQM1/CPM1/CPM”.
Once, the type message shown the PLC type, it means the communication is established. Other wise, the type shows “Unknown”. Then, re-recomfirm the adress setup in FGW again.

Afterward, once the communication is established, do try to other tab.
All read/write should work good.

A little trick to check and comfirm what is read is what inside the PLC memory.
Use a CX-programmer or a programming console.
Write some hex code into the memory, say #FFFF into DM0000. Then, read the info from VB.Net appplet. DM0000 should display #FFFF.
It means the VB.Net really work!