FULLY AUTOMATIC PALLET TRANSFER SYSTEM FOR AUTOMOTIVE INDUSTRIES

By

MOHD RAZIK BIN MOHD AZNAM

FINAL PROJECT REPORT

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

Universiti Teknologi Petronas
Bandar Seri Iskandar
31750 Tronoh
Perak Darul Ridzuan

CERTIFICATION OF APPROVAL

FULLY AUTOMATIC PALLET TRANSFER SYSTEM FOR AUTOMOTIVE INDUSTRIES

by

Mohd Razik bin Mohd Aznam

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

Approved:

Ms Noor Hazrin Hany bt Mohamad Hanif
Project Supervisor

UNIVERSITI TEKNOLOGI PETRONAS
TRONOH, PERAK

June 2006

CERTIFICATION OF ORIGINALITY

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

Mohd Razik bin Mohd Aznam

ABSTRACT

This project is a collaboration project with Perusahaan Otomobil Nasional Berhad (PROTON), by which the department involved is Press & Tool Engineering Department, also known as Stamping Department. This project is focused on the fully automatic press machine called H-Line. As fork lifters are unable to perform many tasks simultaneously, unfilled and fully filled pallets become a problem. This project is done on a basis to minimize the workforce on the production line and to maximize the production quality. The project will cover the research on pallet transfer system and followed by simulation using Programmable Logic Controller (PLC). At the initial stage, the push button will be the input command of the PLC. Once successful, Visual Basic will be designed as the controller material that will monitor the process on the hardware. The specialty of the GUI is the system does not have to be operated at the line but instead only operates the system on a computer. Sensors and motors will control movements of every part in the Pallet Transfer System. External circuits will be added to the system to ensure the correct timing and accuracy of the movements. Mechanical parts also come in handy in designing gearings, shaft, and the roller conveyor. The significance of the project is it will be the latest design of Pallet Transfer System not just in PROTON, but also in Malaysian local automotive industries. The Pallet Transfer also will be replaced in PROTON's entire fully automatic press machine with this new design to increase the pallets management efficiency in PROTON.

ACKNOWLEDGEMENTS

First and foremost, I would like to praise God Almighty and thank Him for this countless blessing and guidance throughout the hard times I have endured through and through. Without Him, I am sure I would not have made it this far.

I would like to express my fullest appreciation and highest gratitude to my supervisor, Ms. Hazrin Hany bt Mohamad Hanif for her guidance and endless advice, the knowledge offered and the time spent in improving this project.

I would also like to express my fullest gratitude to Mr Yuswira Yussop for his patience, understanding and willingness to tutor me. Not to forget, the countless hours of Mechanical knowledge and Q&A sessions.

Last but not least warm thank you to my fellow friends, who helped contributed and shared their ideas, thus helped to enhance the project one way or other.

Thank you.

TABLE OF CONTENTS

LIST OF TABLES ix
LIST OF FIGURES x
LIST OF ABBREVIATIONS xi
CHAPTER 1 INTRODUCTION 1
1.1 Project Background 1
1.2 Problem Statement 1
1.2.1 Problem Identification 1
1.2.2 Significant of the project 2
1.3 Objective and Scope of Work 2
1.3.1 Objective of Project 2
1.3.2 Scope of Study/Work 3
CHAPTER 2 LITERATURE REVIEW 4
2.1 Description of Main Components 6
2.1.1 Zone 1: Crane Saver 6
2.1.2 Zone 2: Destack Station 8
2.1.3 Zone 3: Transfer Conveyor 10
2.1.4 Zone 4: Restack Station 12
2.2 Computer Interface Application 12
2.2.1 Microcontroller Application 12
2.2.2 PIC16F84a Microcontroller 13
2.2.3 Graphical User Interface 16
2.2.4 Microsoft Visual Basic Environment 17
2.2.5 PC Communication: Serial Port Interface 18
CHAPTER 3 METHODOLOGY/PROJECT WORK 22
3.1 Procedure Identification 22
3.1.1 Project Overview 23
3.1.2 Process Flow 24
3.1.3 PLC Ladder Diagram 26
3.1.4 Computer Interface Application 28
3.2 Tools Required 32

CHAPTER 4 DISCUSSION 33
4.1 Prototype build 33
4.1.1 Hardware Preparation 34
4.2 Input Interface (PIC16F84 Microcontroller Architecture) 36
4.2.1 PIC16F84 Microcontroller Circuitry 37
4.2.2 PIC16F84 Microcontroller Programming 37
4.3 Serial Communication Interface 38
4.4 GUI interface – Microsoft Visual Basic 41
4.5 Constraints and Problems Encountered 42
4.6 Comparison with other technology 43
CHAPTER 5 CONCLUSION 44
REFERENCES 45
APPENDICES 46
Appendix A Plc ladder diagram 47
Appendix B autocad drawing 51
Appendix C microcontroller circuitry 52
Appendix D microcontroller programming 53
Appendix E ascii codes 54
Appendix F rs232 circuitry 56
Appendix G GUI windows 57

LIST OF TABLES

Table 1 Trouble and breakdown cases of fully automatic press machine..... 5

LIST OF FIGURES

Figure 1 Layout of the Stamping Department 4
Figure 2 Crane Saver 7
Figure 3 Destack Station 10
Figure 4 Transfer Conveyor 11
Figure 5 Restack Station 12
Figure 6 PIC16F84 pin out 14
Figure 7 Components of a Microcontroller 15
Figure 8 Visual Basic Editor 17
Figure 9 Serial Port (9 pins) 19
Figure 10 Pin Diagrams for 16550, 16450 and 8250 UARTs 20
Figure 11 The process flow of Pallet Transfer System Project 22
Figure 12 Layout of the Pallet Transfer System 24
Figure 13 Latching the output 27
Figure 14 OMRON C200HE 27
Figure 15 Overall Block Diagram 28
Figure 16 Input Interface Details 29
Figure 17 Serial Interface Details 30
Figure 18 GUI interface details 31
Figure 19 Photo Sensor DT-PS-S-001 34
Figure 20 Photo Sensor wiring 34
Figure 21 OMRON C200HE 35
Figure 22 Simplified Block Diagram for PIC16F84 36
Figure 23 RS-232 Pin Out on a DB-9 Pin Used for Asynchronous Data 40
Figure 24 Pallet Transfer in Toyota ASSB 43

LIST OF ABBREVIATIONS

Proton	Perusahaan Otomobil Nasional Berhad
PLC	Programmable Logic Controller.
CMOS	Complimentary Metal Oxide Semiconductor
CPU	Central Processing Unit
DOS	Disk Operating System
EPROM	Erasable Programmable Read Only Memory
GUI	Graphical User Interface
I/O	Input/Output
IC	Integrated Circuit
LED	Light Emitting Diode
OLEDB	Object Link Embedded Database
PC	Personal Computer
PIC	Programmable Integrated Circuit
PWM	Pulse Width Modulation
QBE	Query By Example
RAM	Random Access Memory
RDBMS	Relational Database Management System
RISC	Reduced Instruction Set Computer
RXD	Receive
SCI	Serial Communication Interface
SQL	Structured Query Language
TTL	Transistor-Transistor Logic
TXD	Transmit
UART	Universal Asynchronous Receiver/Transmitter

CHAPTER 1
INTRODUCTION

1.1 Project Background

This project is a collaboration with PROTON. The project was proposed due to the lack of efficiency in pallet handling in the department. The project will be the first Automatic Pallet Transfer System in PROTON and it will be tested on one of the fully automatic press line, which is called H-Line. A study needs to be carried out to create a prototype of the system.

Different types of pallet racks require different ways of pallet transfer. It depends on the size, height, load consumptions, and stacking regulations in the department. For the time being, PROTON has three fully automatic press lines in Shah Alam and four more in Tanjung Malim. This project will be implemented in all of the fully automatic press machine.

1.2 Problem Statement

1.2.1 Problem Identification

Pallet racks are among the problems in Stamping Department as it comes in large quantity. Fork lifters (person responsible with the pallets) have to fully monitor the unfilled and filled pallets, as they are the only individual with transport in the factory. This leads to inefficient time management and leads to lack of production quality. The inappropriate pallets management in any department in PROTON can simply judge the inefficiency of production. It is desired that a pallet system can increase the efficiency of the production in the department.

1.2.2 Significant of the project

This project is significant in such a way that it will be automatic pallet transfer system in our local automotive industries especially industries dealing with stamping body parts such as PROTON and PERODUA. The term 'Pallet Transfer' in other automotive industries are quite common, but the system is simple whereas it can only transfer just one pallet through a roller conveyor from one end to the other end. The activity is done manually by the operators using push button to move the pallet and it is not controlled automatically.

1.3 Objectives and Scope of Work

1.3.1 Objectives of Project

The project is aimed to meet the following objectives:

- To construct PLC program that will be based on the flow of the system. Input and output of the PLC must be initiated to ensure the perfect working model of the system.
- To produce working prototype of scale 1:10 of the actual size. The prototype design is a demand by the collaborator to fabricate the real working system and to present the prototype to the Department Head.
- To design the interface between CPU and PLC using Visual Basic. One of the inputs from PLC is connected to a microcontroller that will send the information to computer through the serial port.

1.3.2 Scope of Study/Work

The scope of work in this project is the construction of pallet transfer system for automotive industries that:

- Consist of four zones that is Crane Saver, Destack Station, Transfer Conveyor, and Restack Station.
- The mechanism of the pallet transfer is controlled by PLC and microcontroller.
- Prototype designing of scale 1:10 of the real size.
- Use of Visual Basic to represent touch panel.

CHAPTER 2
LITERATURE REVIEW

Press & Tool Department or also known as Stamping Department in Proton functions as the body parts making section. The department consists of coil section area, blanking area and press machines. The layout of the stamping department in Proton is showed by the diagram below:

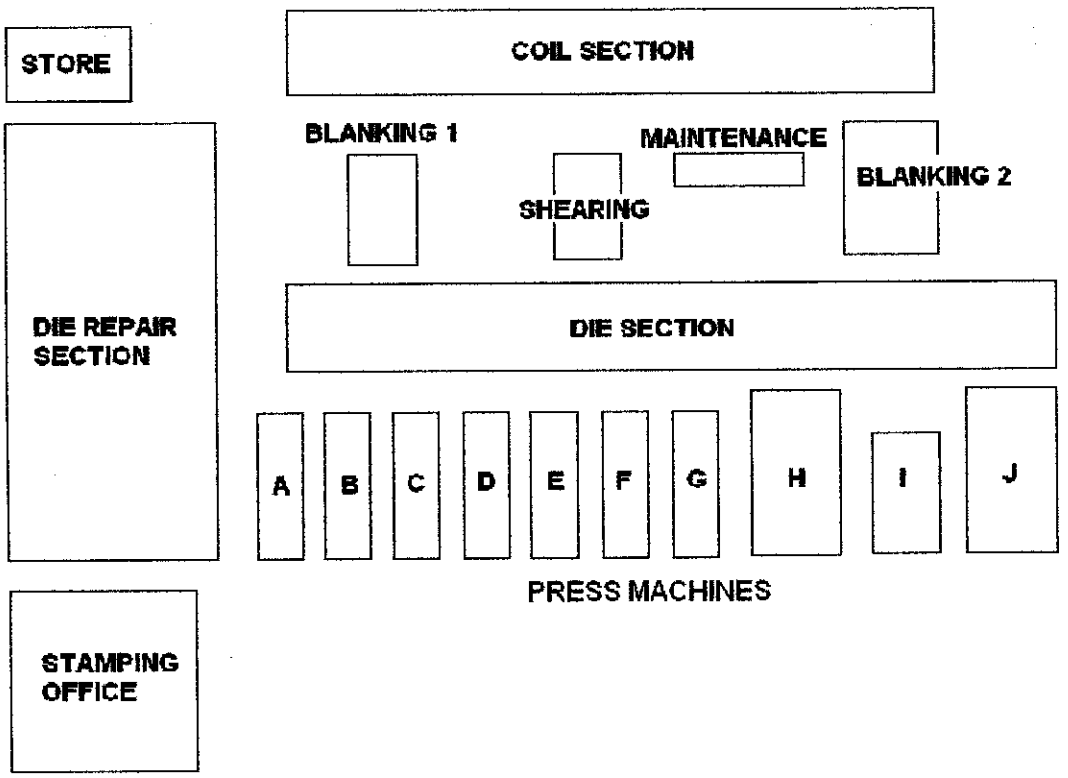


Figure 1 Layout of the Stamping Department

The focus of the project is on the press machine that is the main body parts making machines. In PROTON, there are ten press machines, in which three of the press machines are fully automatic (H-Line, I-Line and J-Line). The project focus is on the H-Line press machine. Among the three fully automatic press line, H-Line was the first press machine to be controlled fully automatic and the movements are controlled

by using sensors and PLC. H-Line recorded the highest number of trouble and breakdown as compared to the I-Line and J-line. The statistic below shows the comparison between the three fully automatic press machines in term of trouble and breakdown.

Table 1 Trouble and breakdown cases of fully automatic press machine

Period \Lines	H-Line	I-Line	J-Line
Jan 05 to July 05	273 cases	47 cases	127 cases

There are three main sections in H-Line press machine that is:

- Destack Feeder – process of taking one metal blank from the blank stacking to be transferred to Idle Stage.
- Idle Stage – process of stamping a blank with a die.
- Loading Area – pallets being placed at this area for operators to fill it with body parts that has been stamped.

All three fully automatic press line in the Stamping Department is controlled by PLC. The PLC used in the Stamping department is Mitsubishi type. The automatic movements are detected by sensors, which consist of electromagnetic sensors, transmit-receive light beam sensors, photo sensors, proximity sensors, and limit switch.

2.1 Description of Main Components

The project is an improvement on the Loading Area where fork lifters will load more than two stacks of pallets at one time at the finishing line and will collect the pallets also more than two stacks at a time. This improvement project is called Pallet Transfer System. The system is divided into four zones which is:

- Zone 1: Crane Saver – brings the stacked pallets into the Destack Station.
- Zone 2: Destack Station – destack the pallets in which one pallet will be transferred to the Transfer Conveyor.
- Zone 3: Transfer Conveyor – transfer the pallet to the middle of the Loading Area where operators will fill the body parts onto the pallet.
- Zone 4: Restack Station – restack the pallet into more than two stacks for fork lifters to transfer them to another place.

The flows of every zone will be discussed in the Methodology section.

2.1.1 Zone 1: Crane Saver

The Crane Saver is located in the beginning of the Pallet Transfer System. The main purpose of this unit is to store the empty pallets and transfer them to the Destack Station. This makes loading of the system more flexible and reduces waiting time for loading devices.

Crane Saver consists of two main parts:

a) Platform (main body of crane saver).

b) Wheel boxes and drive.

a) Platform

The platform is of fully welded construction. This frame is the base for all other parts. (ie. Wheel boxes, drive and locating devices) On the top of the frame is welded a metal sheet which creates the working platform of the crane saver. Several locating pins are welded on the platform to provide correct location of pallets. At

the bottom end of the frame there are welded plates for the attachment of the wheel box and geared motor.

b) Wheel boxes and drive

The crane saver runs on the four wheels. The wheel boxes incorporate the wheel and bearings as a separate sub-assembly for easy attachment to the main frame. The crane saver uses the same rail tracks used by the Transfer Conveyor. These tracks consist of two channels, welded together with rail and base plates. An electric geared motor is employed to drive the crane saver using the tracks. The crane saver is able to move from its initial loading position (situated at the end of the rail track) to the position in the Destack Station and then return to the home position. Movement of the crane saver is controlled by PLC.

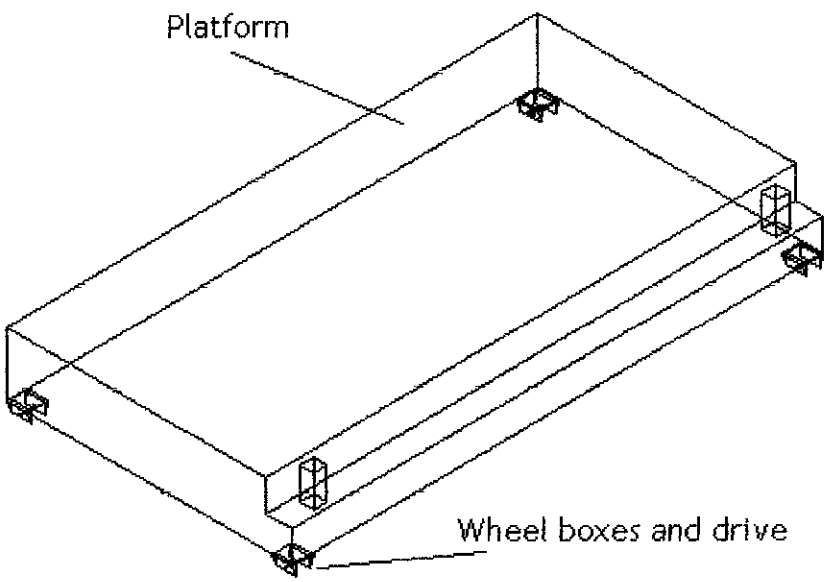


Figure 2 Crane Saver

2.1.2 Zone 2: Destack Station

The load station is the first machine of the Pallet Transfer System. The main purpose of the Destack Station is to hold the stack of empty pallets and destack them one-by-one on the Empty Racks Transfer Conveyor.

The Destack Station is divided into the following main parts:

a) Unmovable frame.

b) Movable frame.

c) Front lifting beam.

d) Rear lifting beam.

e) Lifting drives.

a) Unmovable frame

The unmovable frame consists of two stationary posts attached to the floor. Posts create the base for the linear bearing rails and accommodate the ball lifting drives. An electric motor for lifting drives attached to the left hand side post.

b) Movable frame

The movable frame consists of two posts. Each post runs on two wheels fitted with roller bearings. The rail track consists of two I-beams. The posts create a base for the linear bearing rails and accommodate the ball screw lifting drives. The ball screws of the lifting drives on each post on the movable frame are driven through spline shafts from an electric motor, attached to the unmovable frame through bevel gearboxes.

c) Front lifting beam

The front lifting beam connects the posts of the unmovable frame. It supported by linear bearing running blocks. It can move up and down using ball screw lifting

drives. There are a number of lifting pins welded to the beam which are used for holding and destacking the pallets.

d) Rear lifting beam

The rear lifting beam connects posts of the movable frame. It supported by linear bearing running blocks. It can move up and down using ball screw lifting drives. In addition the rear lifting the rear lifting beam can be rotated by its own electric motor around pivot point's on the ball screw lifting drives.

e) Lifting drives

Lifting drives are generally the same for all four (4) posts of both the movable and unmovable frames. They include a carriage with ball nut and linear bearing running blocks. Carriages mounted on the unmovable frame are connected together with the front lifting beam. Carriages mounted on the movable frame are connected together with the rear lifting beam with rotating arms. Movements of all components of the Destack Station can be controlled by PLC.

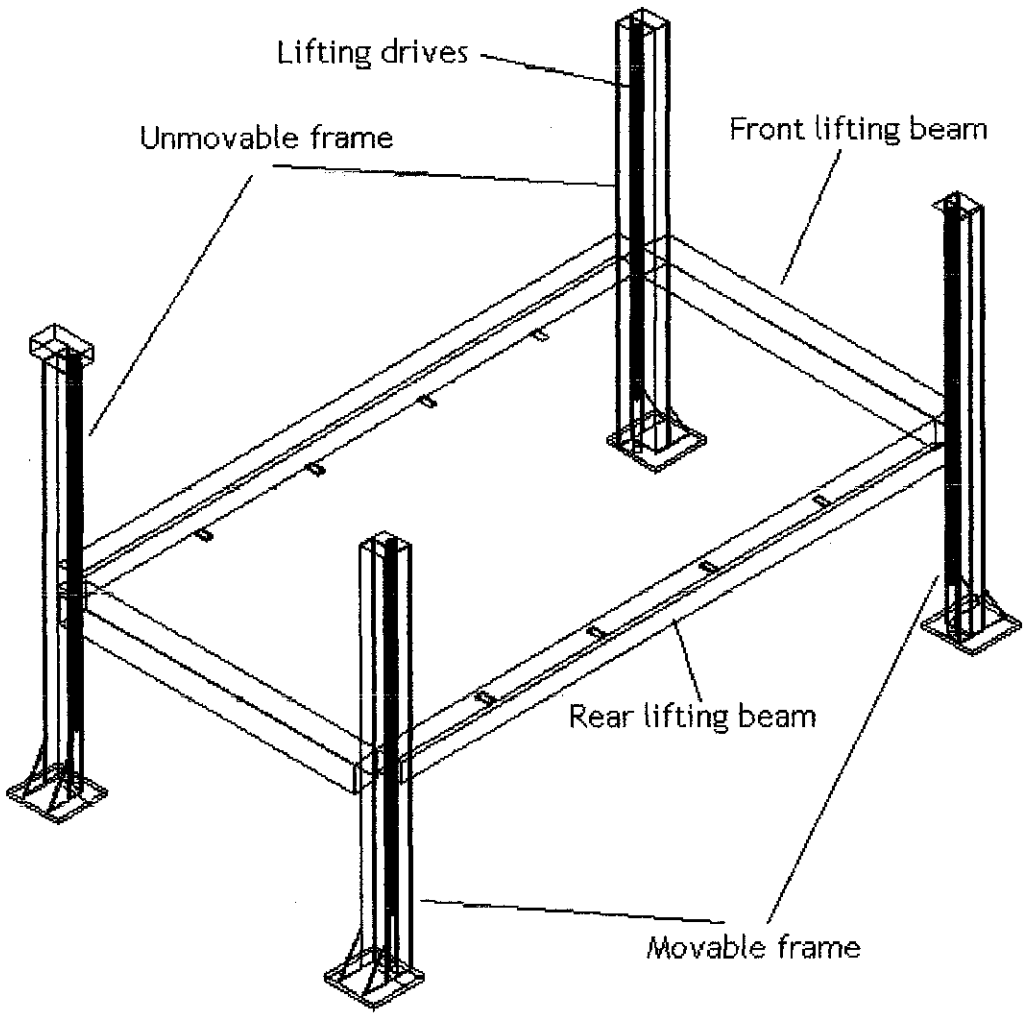


Figure 3 Destack Station

2.1.3 Zone 3: Transfer Conveyor

The Transfer Conveyor is the third machine of the Rack Transfer System.

The main purpose of the Transfer Conveyor is to move empty racks one-by-one from the Conveyor 1 to the packing area, hold the rack when it is filled by operators and transfer the Conveyor 3.

Transfer Conveyors consists of three main parts:

a) Trolley;

b) Slat Conveyor;

c) Operator's platform.

a) Trolley

The trolley is fully welded construction. It creates a base for the slat conveyor. The trolley runs on four wheels fitted with roller bearings. The rail track consists of two I-beams. The Transfer Conveyor 1 can move from the initial position next to the Transfer Conveyors 2 in the packing area to the Transfer Conveyor 3 and returns to the home position. Movement of the Transfer Conveyor is connected by PLC or manually.

b) Slat Conveyor

The slat conveyor is placed on the top of the trolley. It consists of an electric drive, two (2) pairs of sprockets and two (2) top chains connected with bolted slats.

c) Operator's Platform

The operator's platform is a light steel structure which is attached to the trolley of the Transfer Conveyor. The operator's platform provides space for the worker who erects the rack before moving to the packing area.

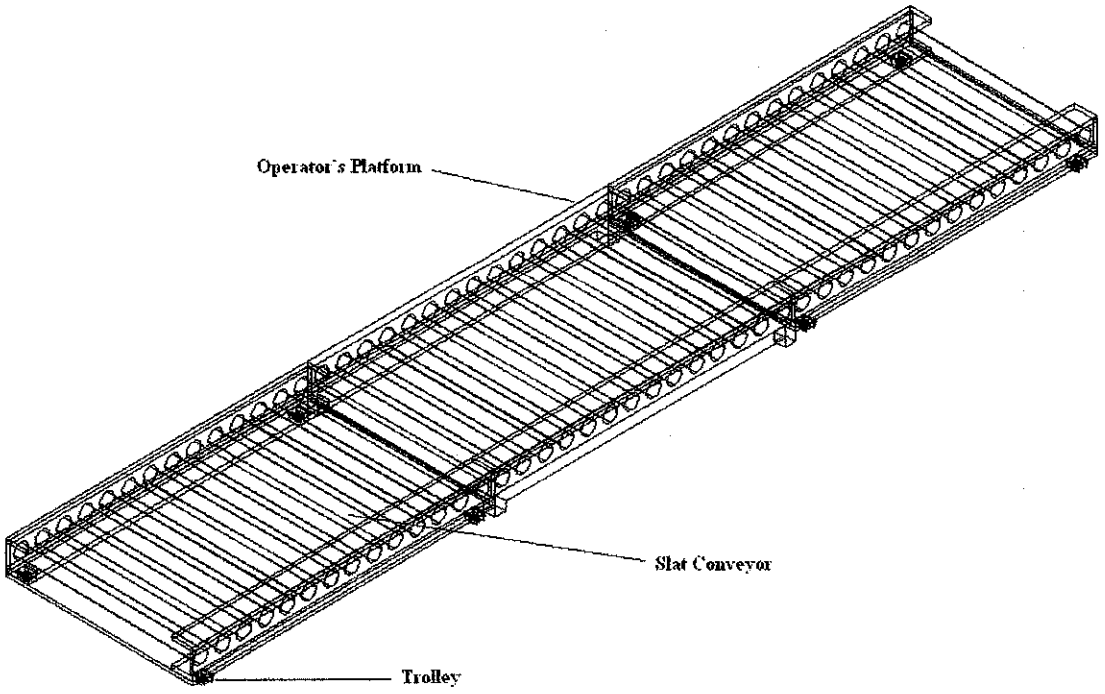


Figure 4 Transfer Conveyor

2.1.4 Zone 4: Restack Station

The Full Restack Station is the fourth machine of the Rack Transfer System. The main purpose of the Full Restack Station is to unload the Full Racks Transfer Conveyor, hold the full rack and stack on the Transfer Conveyor. The description the every parts in the Restack Station is the same with Destack Station. Destack Station will destack the pallets and Restack Station will stack them up together again for the fork lifters to take them away.

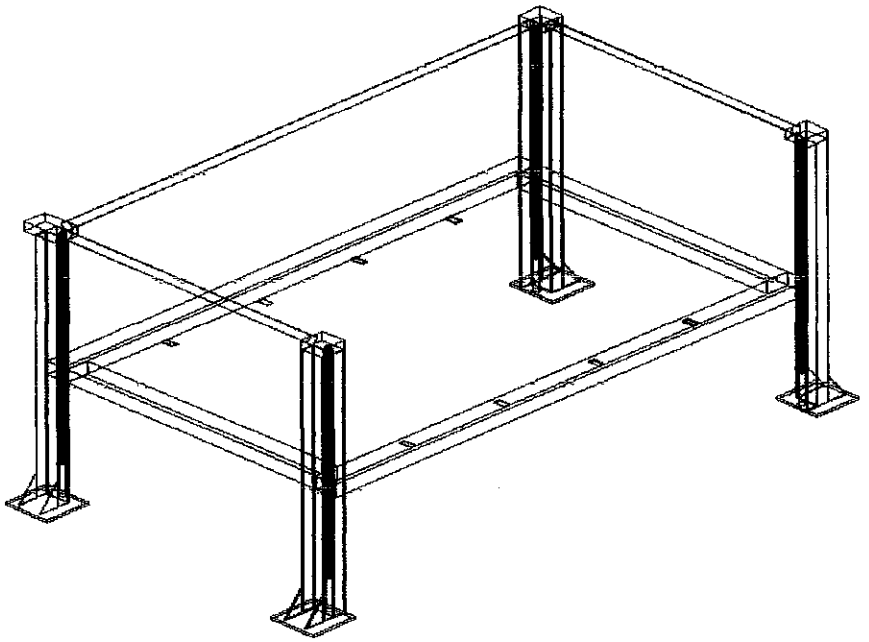


Figure 5 Restack Station

2.2 Computer Interface Application

2.2.1 Microcontroller Application

The evolution of the microprocessor technology is marked by a major fork. One branch of the evolutionary tree is represented by Pentium chips and Power-PC chips which form the heart of personal computers. This mentioned branch is mainly focused upon enhancing raw computers. Meanwhile, the other branch, which is 30 times larger in unit volume, is represented by microcontrollers. Basically, in microcontroller technology, the emphasis is upon the integration of many features

needed in a single chip so that the chip has versatility to sense inputs and control outputs in a device or instrument. Mainly, the 8-bit microcontroller market is dominated by Motorola and Microchip Technology. All these devices are optimized to meet stringent performance requirements in cost-effective applications. [1]

Among the main features that contribute to the wide popularity of microcontrollers are:

- Speed
- Instruction set simplicity
- Integration of operational features
- Programmable timer options
- Interrupt control
- Powerful output pin control
- I/O port expansion
- Serial programming via two pins
- EPROM support
- Mail-order support
- Free assembler and simulator

2.2.2 PIC16F84a Microcontroller

The PIC16F84 belongs to the family of low-cost, high-performance, CMOS, fully-static, 8-bit microcontrollers. All PICmicro™ microcontrollers employ an advanced RISC architecture. PIC16F84 have enhanced core features, eight-level deep stack, and multiple internal and external interrupt sources. The separate instruction and data buses of the Harvard architecture allow a 14-bit wide instruction word with a separate 8-bit wide data bus. The two stage instruction pipeline allows all instructions to execute in a single cycle, except for program branches (which require two cycles). A total of 35 instructions (reduced instruction set) are available. Additionally, a large register set is used to achieve a very high performance level. [3]

PIC16F84 microcontrollers typically achieve a 2:1 code compression and up to a 4:1 speed improvement (at 20 MHz) over other 8-bit microcontrollers in their class. The PIC16F84 has up to 68 bytes of RAM, 64 bytes of Data EEPROM memory, and 13 I/O pins. A timer/counter is also available. [3]

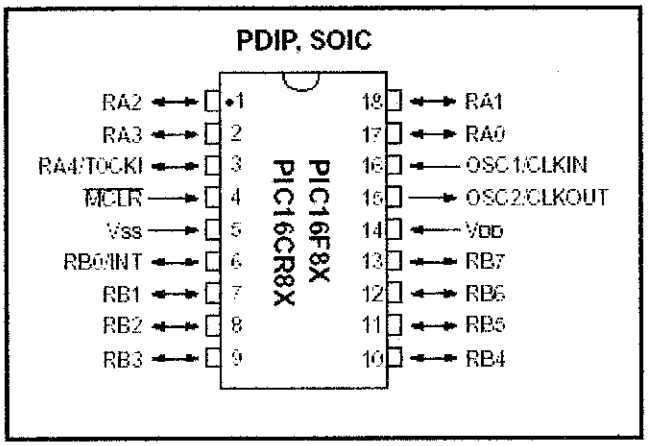


Figure 6 PIC16F84 pin out

There are four oscillator options, of which the single pin RC oscillator provides a low-cost solution, the LP oscillator minimizes power consumption, XT is a standard crystal, and the HS is for High Speed crystals.

The interesting features in the PIC16F84 are:

- The SLEEP (power-down) mode offers power saving. The user can wake the chip from sleep through several external and internal interrupts and resets.
- A highly reliable Watchdog Timer with its own on-chip RC oscillator provides protection against software lockup.

The devices with Flash program memory allow the same device package to be used for prototyping and production. In-circuit reprogrammability allows the code to be updated without the device being removed from the end application. This is useful in the development of many applications where the device may not be easily accessible, but the prototypes may require code updates. This is also useful for remote applications where the code may need to be updated (such as rate information). A simplified block diagram of the PIC16F84 is shown in Figure 7.

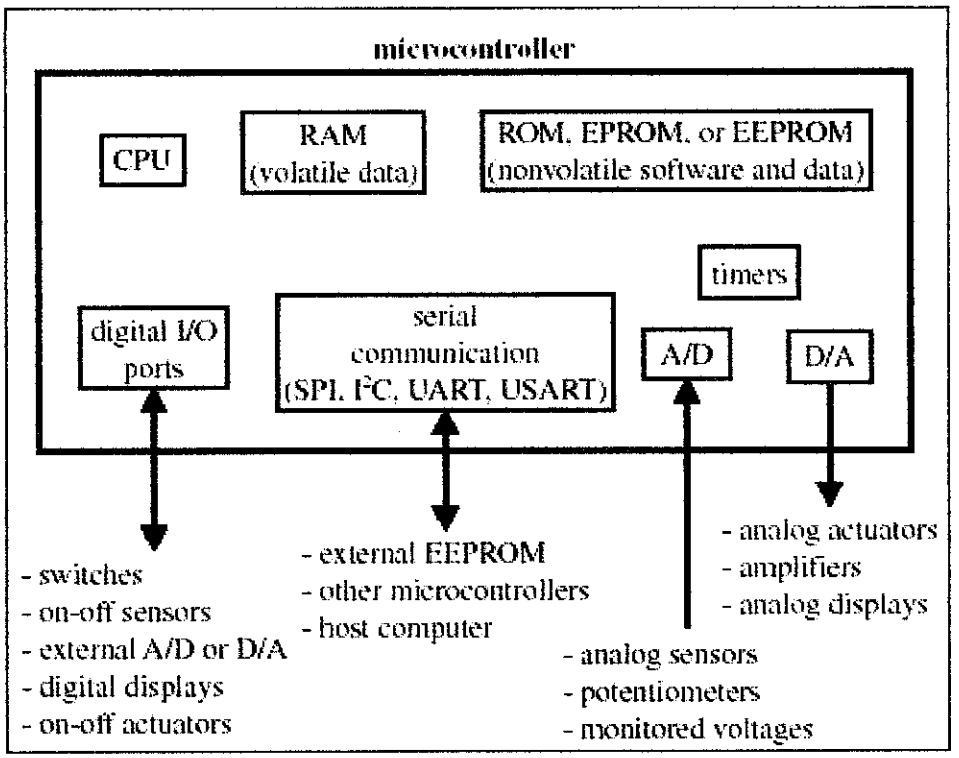


Figure 7 Components of a Microcontroller

The PIC16F84 fits perfectly in applications ranging from high speed automotive and appliance motor control to low-power remote sensors, electronic locks, security devices and smart cards. The Flash or EEPROM technology makes customization of application programs such as transmitter codes, motor speeds, receiver frequencies and security codes extremely fast and convenient. The small footprint packages make this microcontroller series perfect for all applications with space limitations. Low-cost, low-power, high performance, ease-of-use and I/O flexibility make the PIC16F84 very versatile even in areas where no microcontroller use has been considered before (example: timer functions; serial communication; capture, compare and PWM functions; and co-processor applications). [3]

The serial in-system programming feature (via two pins) offers flexibility of customizing the product after complete assembly and testing. This feature can be used to serialize a product, store calibration data, or program the device with the current firmware before shipping.

2.2.3 Graphical User Interface

A GUI is a graphical, (rather than purely textual) user interface to a computer. The term came into existence because the first interactive user interfaces to computers were not graphical; they were text-and-keyboard oriented and usually consisted of commands that we had to remember and computer responses that were infamously brief. The command interface of the DOS operating system is an example of the typical user-computer interface before GUIs arrived. An intermediate step in user interfaces between the command line interface and the GUI was the non-graphical menu-based interface, which allows interaction by using a mouse rather than by having to type in keyboard commands. [6]

Nowadays, major operating systems provide a graphical user interface. Applications typically use the elements of the GUI that come with the operating system and add their own graphical user interface elements and ideas. A GUI sometimes uses one or more metaphors for objects familiar in real life, such as the desktop, the view through a window, or the physical layout in a building. Elements of a GUI include such things as:

- windows
- pull-down menus
- buttons
- scroll bars
- iconic images
- wizards

With the increasing use of multimedia as part of the GUI, sound, voice, motion video, and virtual reality interfaces seem likely to become part of the GUI for many applications. A system's graphical user interface along with its input devices is sometimes referred to as its "look-and-feel." [6]

The GUI familiar to most of us today in either the Mac or the Windows operating systems and their applications originated at the Xerox Palo Alto Research Laboratory in the late 1970s. Apple used it in their first Macintosh computers. Later, Microsoft used many of the same ideas in their first version of the Windows operating system for IBM-compatible PCs. [6]

2.2.4 Microsoft Visual Basic Environment

The Microsoft Visual Basic is known to be among the most popular choice to create Windows GUI. In Visual Basic, new windows created are called forms. Elements (such as text boxes and buttons) that are placed inside a form are called controls. The Visual Basic allows event-driven programming, where the user's actions cause events, and each event in turn triggers a procedure that is associated with it.

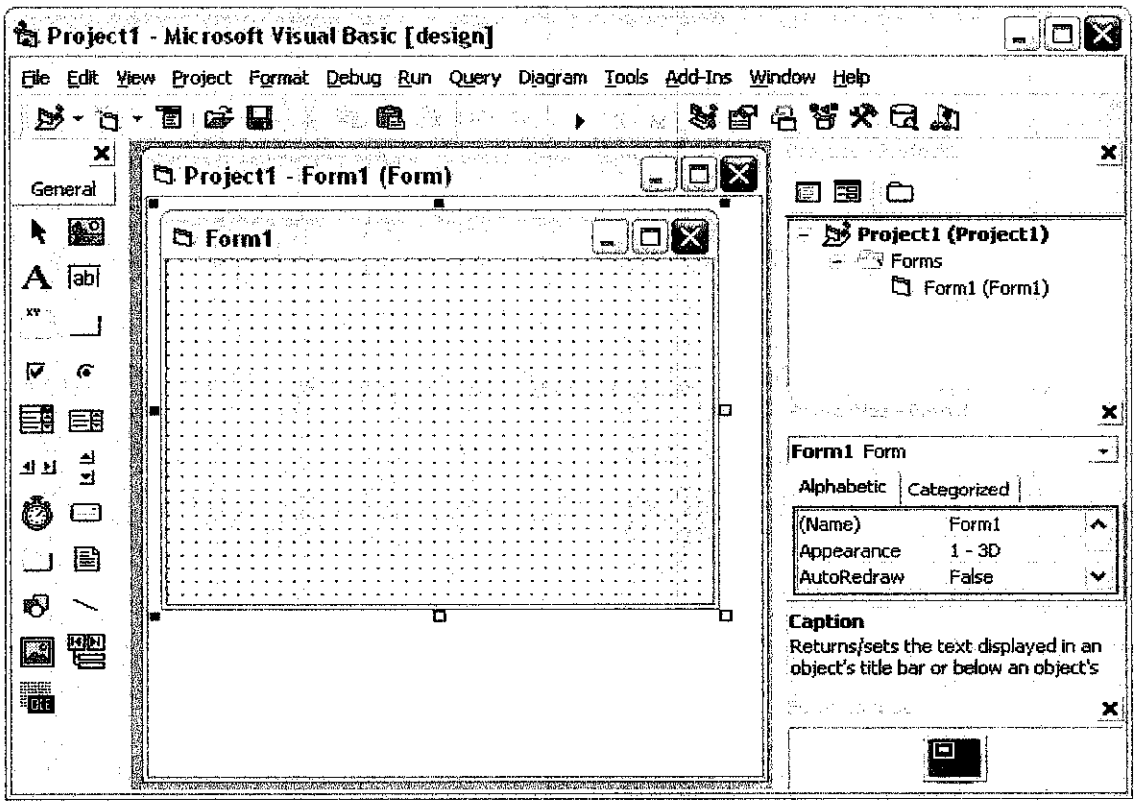


Figure 8 Visual Basic Editor

The properties of creating an object model in Visual Basic are:

- Objects have properties and methods
- Forms and controls are examples of objects
- Properties describe an object. Examples: name, color, size, or how they will behave
- Methods are actions associated with an object. Examples: move, clear and print

In order to write a proper Visual Basic project, there several important elements to learn and understand. The two vital steps are:

1. *Planning*

- Design the user interface
- Plan the properties
- Plan the Basic code - procedures are associated with the events, actions written in pseudocode

2. *Programming*

- Define the user interface - define objects
- Set the properties
- Write the Basic code

2.2.5 PC Communication: Serial Port Interface

The serial port is an I/O device. An I/O device is just a way to get data into and out of a computer. There are many types of I/O devices such as serial ports, parallel ports, disk drive controllers, Ethernet boards, universal serial buses and many others. Most PC's have one or two serial ports. Each has a 9-pin connector or sometimes 25-pin on the back of the computer. Computer programs can send data (bytes) to the transmit pin (output) and receive bytes from the receive pin (input). The other pins are for control purposes and ground. [14]

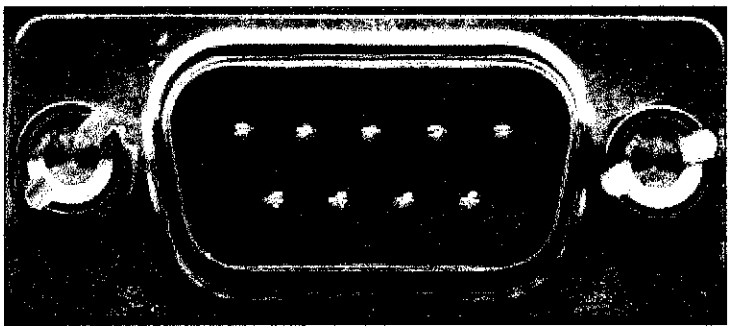


Figure 9 Serial Port (9 pins)

The serial port is much more than just a connector is. It converts the data from parallel to serial and changes the electrical representation of the data. Inside the computer, data bits flow in parallel, using many wires at the same time. Serial flow is a stream of bits over a single wire, such as on the Transmit or Receive pin of the serial connector. For the serial port to create such a flow, it must convert data from parallel inside the computer to serial on the transmit pin and conversely. [14]

Most of the electronics of the serial port is found in a computer chip or a part of a chip known as a UART. UART stands for Universal Asynchronous Receiver / Transmitter. It is the little box of tricks found on the serial card, which plays the little games with the modem or other connected devices. Most cards will have the UARTs integrated into other chips that may also control the parallel port, games port, floppy or hard disk drives and are typically surface mount devices. The 8250 series, which includes the 16450, 16550, 16650, & 16750 UARTs are the most commonly found type in your PC. [14]

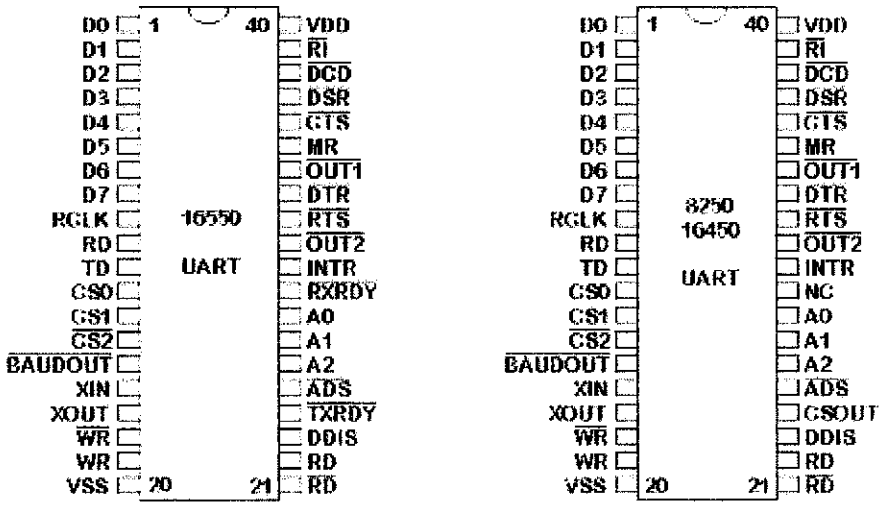


Figure 10 Pin Diagrams for 16550, 16450 and 8250 UARTs

The serial port is harder to interface than the parallel port. In most cases, any device connected to the serial port will need the serial transmission converted back to parallel so that it can be used. This can be done using a UART. On the software side of things, many more registers have to be attended to than on a standard parallel port.

The advantages of using serial data transfer rather than parallel are: [14]

- Serial cables can be longer than parallel cables. The serial port transmits a '1' as -3 to -25 volts and a '0' as +3 to +25 volts where as a parallel port transmits a '0' as 0v and a '1' as 5v. Therefore, the serial port can have a maximum swing of 50V compared to the parallel port, which has a maximum swing of 5 Volts. Therefore, cable loss is not going to be as much of a problem for serial cables as they are for parallel.

- Less wires than parallel transmission. If your device needs to be mounted a far distance away from the computer then 3 core cable (Null Modem Configuration) is going to be a lot cheaper that running 19 or 25 core cable. However, you must take into account the cost of the interfacing at each end.

- Infra Red devices have proven quite popular recently. It is common to see many electronic diaries and palmtop computers that have infrared capabilities build in. Imagine transmitting 8 bits of data at the one time across the room and being able to (from the devices point of view) decipher which bits are which? Therefore, serial transmission is used where one bit is sent at a time. IrDA-1, the first infrared specifications was capable of 115.2k baud and was interfaced into a UART. The pulse length however was cut down to 3/16th of a RS232 bit length to conserve power considering these devices are mainly used on diaries, laptops and palmtops.

- Microcontrollers have also proven to be quite popular recently. Many of these have in built SCI (Serial Communications Interfaces) which can be used to talk to the outside world. Serial communication reduces the pin count of these MPU's. Only two pins are commonly used, Transmit Data (TXD) and Receive Data (RXD) compared with at least eight pins using an 8 bit Parallel method. Furthermore, it may also require a Strobe.

CHAPTER 3
METHODOLOGY/PROJECT WORK

3.1 Procedure Identification

Several main procedures have been identified towards accomplishing the project. The following diagram summarizes on the task to be performed accordingly.

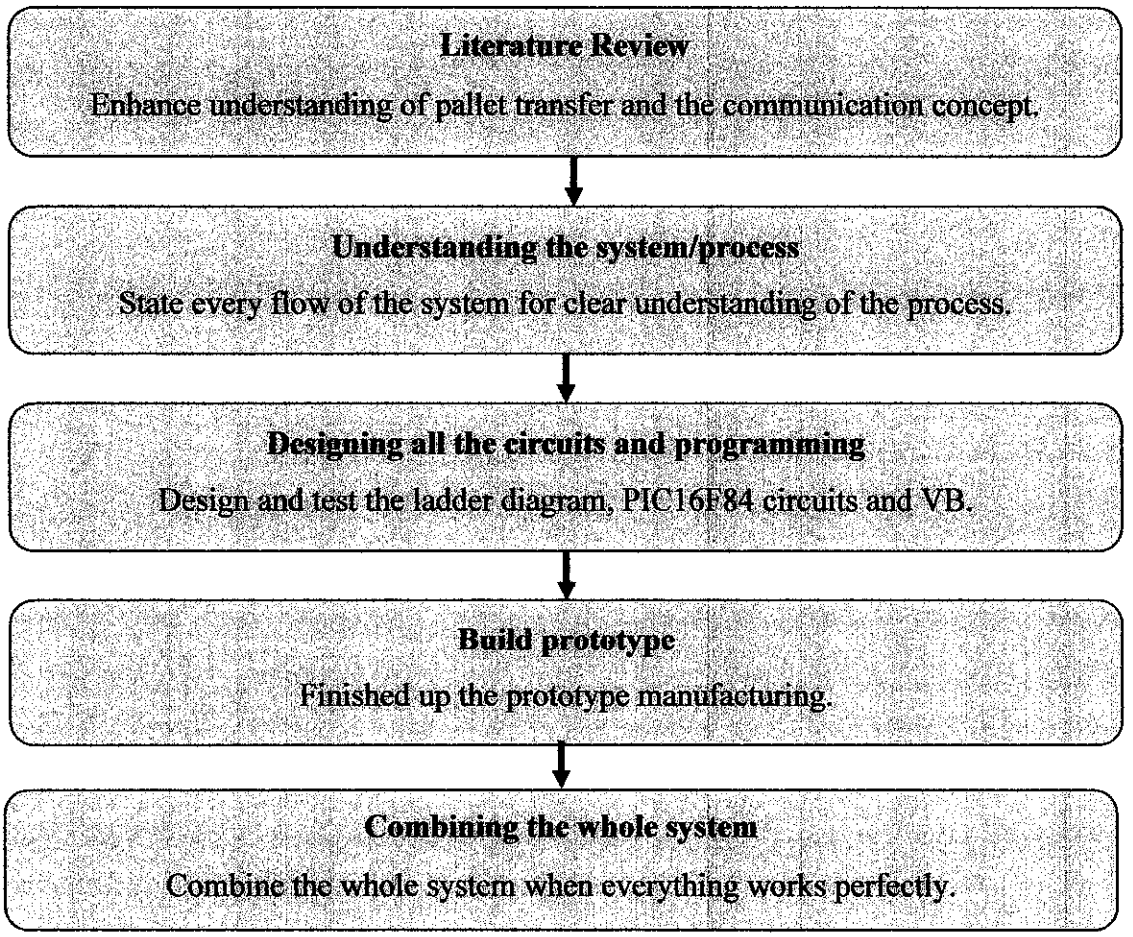


Figure 11 The process flow of Pallet Transfer System Project

3.1.1 Project Overview

The Pallet Transfer System is a group of machines at the end of a press line.

The major purpose of the Pallet Transfer System is to automatically supply empty racks from the loading point to the packing area at the end of press line and to then transfer full racks away to a disposal area, where they are picked up by a fork lift truck.

The design of the Pallet Transfer System allows the handling of different types of racks with wide variations in the dimensions and design. The design of the Pallet Transfer System allows the operator to load a stack of empty pallets.

The Pallet Transfer System destacks pallets and brings them to the packing area, where they are filled by operators. The Pallet Transfer System moves full Pallets to the stack station, where they are restacked and can be picked up by forklift truck.

The design of the Pallet Transfer System allows the operator to replace the full pallet with the next empty pallet without stoppage to the press line. At the same time, the pallet change operation is completely safe for all personnel within the vicinity of the packing area.

Defective parts are produced from time to time. Therefore, it is necessary to have another Pallet positioned in easy reach of the packing area so that these panels remain separated from the good quality panels. The Pallet Transfer System provides space for a rework platform that can be easily removed by forklift truck.

The Pallet Transfer System consists of four individual machines which through electronics and programming, work in unison to destack pallet, transfer it to the packing area, remove full pallet from the packing area and stack it for the forklift truck to pick it up.

The layout of the project is stated in Figure 12.

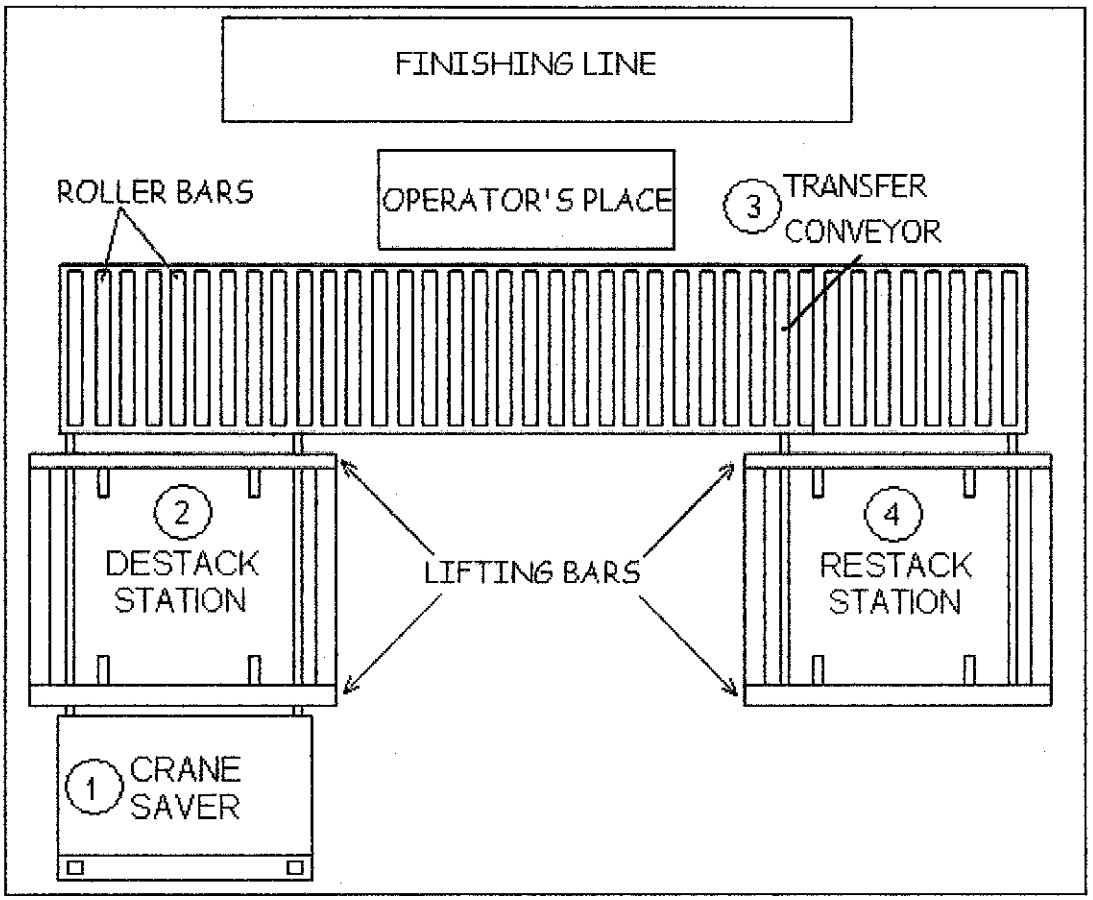


Figure 12 Layout of the Pallet Transfer System

The overall processes of the Pallet Transfer System are divided into four zones which mentioned earlier. The four zones are:

- Zone 1: Crane Saver.
- Zone 2: Destack Station.
- Zone 3: Transfer Conveyor.
- Zone 4: Restack Station.

3.1.2 Process Flow

Zone 1: Crane Saver.

1) Forklift loads empty pallets
2) Operator enters pallet identification.
3) Crane saver moves into destack position. Photosensor 1 detects crane saver and motor will slow down. Photosensor 2 will stop the crane saver.
4) Destack beams down to destack position. Proximity sensor 1 detects position.

5) Destack beam (press side) moves towards pallets. Photosensor 2 detects presence of pallets and stops.
6) Both destack beams move up lifting all racks clear of crane saver.
7) Crane saver moves back to pallet load position. Proximity sensor 2 detects this position.

Zone 2: Destack Station

1) Destack position. Proximity sensor detects position is made.
2) Destack beams move down placing all racks on transfer conveyor. Photosensor 2 detects position on transfer conveyor.
3) Destack beams move out to extremities.
4) Destack beam move up by a preset amount (one pallet) dependant on pallet identification and measured by encoder function.
5) Destack beams move in.
6) Destack beams move up by 50mm leaving one layer of racks on transfer conveyor.
7) Transfer conveyor moves pallet to original position.

Zone 3: Transfer Conveyor

1) First pallet transfers from conveyor 1 to conveyor 2.
2) Operator starts load the pallet with body parts.
3) Transfer conveyor 1 moves to Destack Station and receives another pallet. *(operation Zone 2 repeated)*
4) When the pallet is full, the system removes the full pallet to Transfer Conveyor 3 and receives an empty pallet from Transfer Conveyor 1.

Zone 4: Restack Station

1) Full pallet is transfered to Conveyor 3 with limit switch.
2) Conveyor 3 moves into restack position with photosensor detect its presence.
3) Restack beam lower down until proximity sensor detects the pallet's base.
4) Restack aisle side moves in and limit switch detects and identifies the rack position.
5) Restack beams lift pallet clear of conveyor 3 and lift a further rack height.

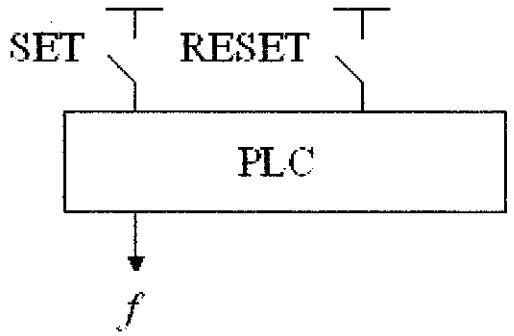
6) Conveyor 3 return to original position.
7) Next full pallet delivers into conveyor 3.
8) Conveyor 3 moves to restack position.
9) Restack beam lowers first pallet onto the second pallet which has just been delivered.
10) Restack beams move apart.
11) Restack beams move down past conveyor giving forklift access for removal of two layers of stacks.
12) Operator resets allowing restack beams to move out of the way and return conveyor 3 to home position.

3.1.3 PLC Ladder Diagram

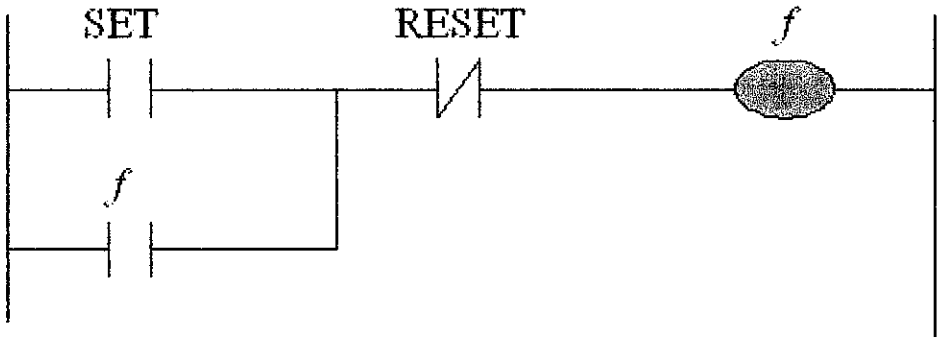
The PLC ladder diagram has been designed based on several steps.

The first ladder diagram is designed using GX-Developer that is meant for Mitsubishi PLC. Problem raised when the Mitsubishi PLC is not available in UTP. The most common PLC set in UTP is OMRON. So the next step is to design the ladder diagram into CX-Programmer that is on OMRON.

The second ladder diagram is designed using software CX-Programmer for OMRON's PLC. The design was failed with several errors and the most problem is with the output. The output (motor) only trigger on for about less than a second. This is because the output in the ladder diagram designed is not latched. That's why the output does not continuously ON for the desired period. The ladder diagram been modified by this theory shown in Figure 13.



Schematic layout diagram



Ladder diagram

Figure 13 Latching the output

The third ladder diagram designed with no errors and tested perfectly. The ladder diagram will be discussed in the Discussion section.

PLC chosen is OMRON PLC C200HE.

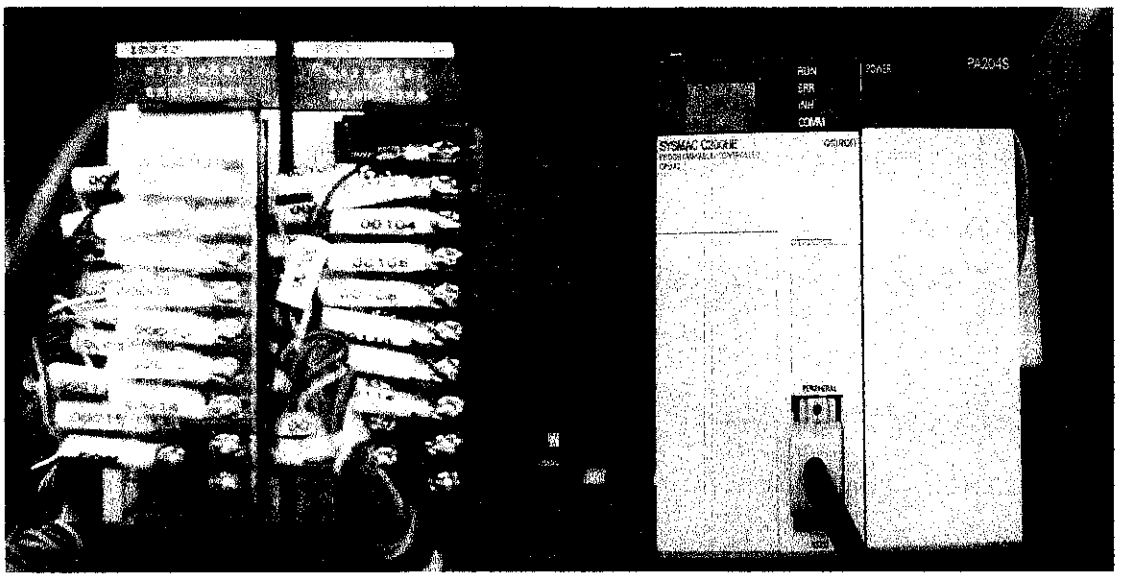


Figure 14 OMRON C200HE

3.1.4 Computer Interface Application

As the title suggests, the procedures involved for implementation throughout the project work are divided into several systematic steps. They are therefore based on the general block diagram which represents the overall attributes of the system. The block diagram is shown in Figure 15 below.

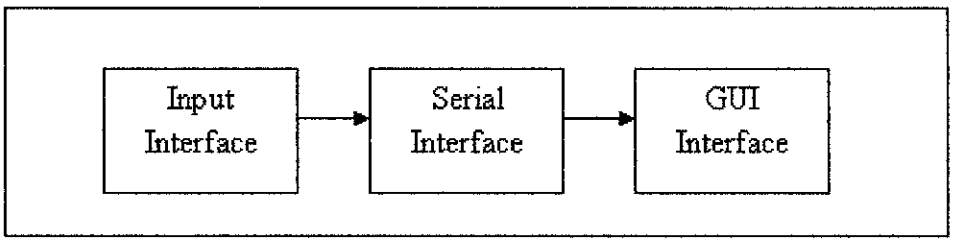


Figure 15 Overall Block Diagram

The basic structure of this system consists of:

1. An input interface which consists of the microcontroller circuitry that connects the sensors, microcontroller and output components.
2. A serial interface to gather the feedback from the microcontroller to transmit to the serial port of a PC.
3. A GUI interface to extract the input from the serial port of the PC while allowing the user to monitor.

1. Input Interface

Generally, the input interface is all about managing the microcontroller for the desired tasks. Figure 16 provides better understanding for this interface.

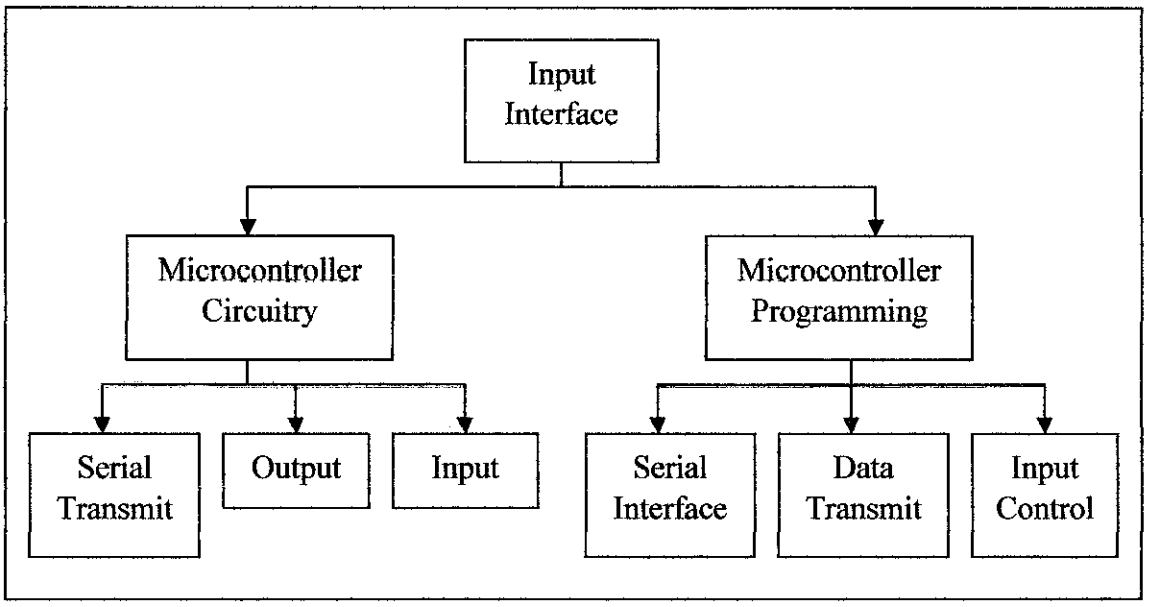


Figure 16 Input Interface Details

As shown, this interface is divided into two important sections:

- Microcontroller circuitry
- Microcontroller programming

In the microcontroller circuitry, it is important to specify in detail the pins involved for the input, output and serial transmit. This is to ensure that the microcontroller is utilized to its optimum performance. At this point, all the pins at PORT B of the PIC16F84 will be used for the outputs. This leaves PORT A, which will be used for the inputs from the sensors and serial transmit.

The C language is chosen to program the microcontroller. The programming must therefore coincide with all the pins that will be utilized. The four main considerations for the programming are:

- Serial interface
- Data transmit
- Input control
- Output control

The initializations for serial interface will allow the microcontroller to communicate with the serial port of the PC. The format of the data to be transmitted to the serial port must be defined to avoid misinterpretation of data. For simple and practical application, the ASCII codes are used for the data transmit. Lastly, the program must also include input control to allow the microcontroller to respond to any incoming input from the sensors.

2. Serial Interface

This interface comes in between the microcontroller circuitry and the serial port of the PC. The details are shown in the figure 17 below.

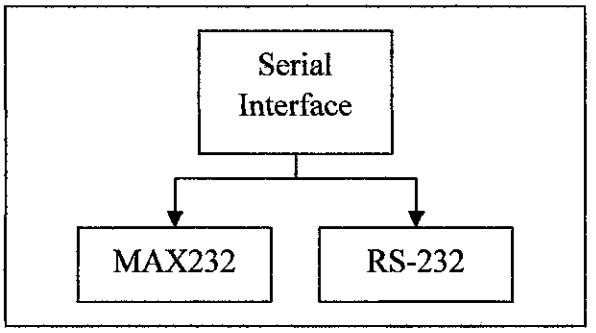


Figure 17 Serial Interface Details

Basically, the data transmitted from the microcontroller will be managed by two important components. They are:

- MAX232
- RS-232

The MAX232 is a component that converts TTL logic levels (0V to 5V) to swing between 12V to -12V. From here, it will go through RS-232 to complete the communication with the serial port of the PC.

3. GUI Interface

Microsoft Visual Basic is chosen to design the GUI interface for the system. Therefore, it must accomplish several important characteristics of the interface. The details are as in Figure 18 below.

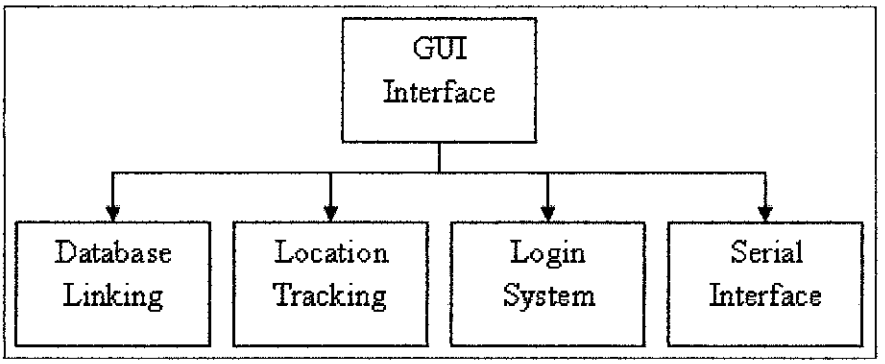


Figure 18 GUI interface details

As visualized in the figure, the main characteristics are:

- VB GUI interface which will alert the user and allow the user to respond to any input from the serial port.
- Sensor detects will represent the surveillance parameter for monitoring purposes.
- Login system to allow only its rightful user to use the system.
- Serial interface which will coincide with the serial settings at the PC and microcontroller

3.2 Tools Required

Hardware requirements:

1) Limit Switches
2) Photo Sensors
3) Proximity Sensors
4) RS-DC motors
5) A lot of L-shape aluminums
6) Cylindrical hollow aluminums
7) OMRON PLC (C200HE)

Software requirements:

1) CX-Programmer

3.2.2 Communication Application

Hardware requirements:

1) A PC equipped with at least Pentium III or equivalent processor, 128 MB RAM, Microsoft 2000/ME/XP operating system and D-9 serial port.
2) Microcontroller – PIC16F84
3) RS-232 port connector – female
4) MAX232 level converter IC

Software requirements:

1) CCS Compiler – Microchip PIC C programming software
2) Microsoft Visual Basic – Windows GUI programming software

CHAPTER 4
RESULTS & DISCUSSION

4.1 Prototype

The prototype of Pallet Transfer System is built to better illustrate the capability of the system. The illustration drawing of the prototype built is made available in Appendix B. Basically, the design and shape of the model coincides with the location map designed using Visual Basic. This will allow the demonstration of how the system surveys and tracks the desired objects.

In this prototype, six objects are specified as the secured objects for tracking. Each of these objects will be placed on the designated location on the prototype.

The mechanisms used to trigger the system are limit switches, photosensors, and proximity sensors. The connections from the switches to PLC are using mechanical relays. Same goes to the output of PLC to the motor and microcontroller.

This prototype enables the demonstration of real time surveillance and tracking of the system designed.

4.1.1 Hardware Preparation

1) Photo Sensor

Photo sensors chosen is Photo Sensor DT-PS-S-001.

- Built-in amplifiers accept wide supply voltage range.
- Slim, space saving construction measures only 50x50x17.4mm.
- Relay outputs with long life expectancy and high switching capacity.
- Polarized retroreflective type available for glossy or shiny object detection.

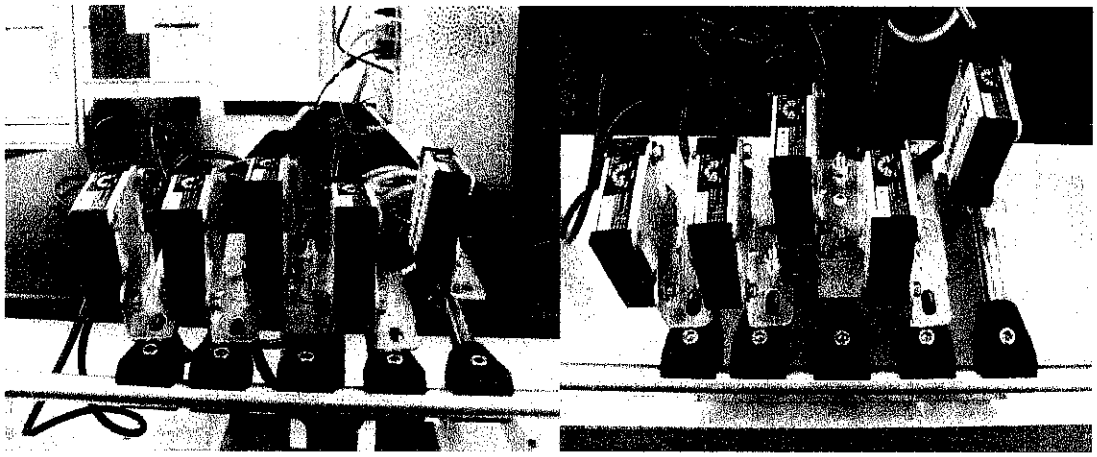


Figure 19 Photo Sensor DT-PS-S-001

The connection of the photo sensor is:

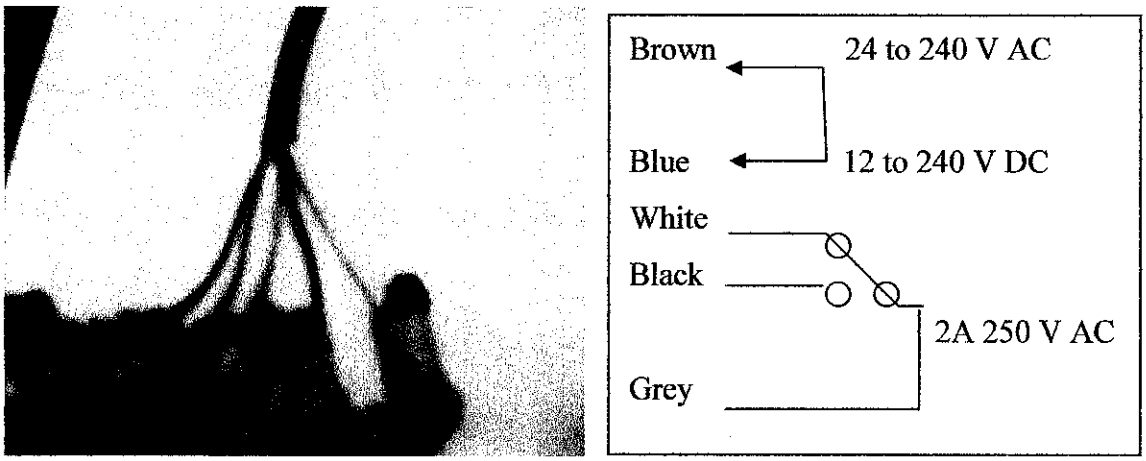


Figure 20 Photo Sensor wiring

2) Programmable Logic Control

PLC chosen is OMRON PLC C200HE.

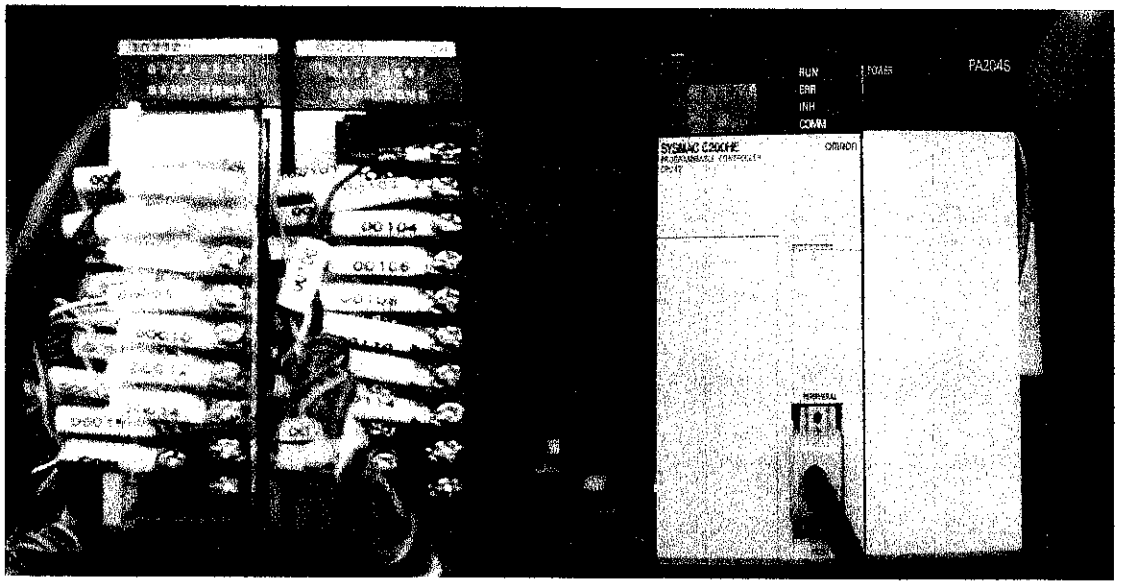


Figure 21 OMRON C200HE

3) Proximity Sensor

Proximity Sensor chosen is Proximity Sensor DT-CPS-S-001

- Senses both metallic and nonmetallic objects (glass, lumber, water, oil, plastic, etc.) without direct contact with objects.
- Indirectly senses objects buried in a nonmetallic wall of those placed in a nonmetallic container.
- Loads up to 200 mA can be switched at 90 to 250 VAC switching type and at 10 to 40 VDC switching type.
- Ideal for to sorting out various shapes workpieces.

4.2 Input Interface (PIC16F84 Microcontroller Architecture)

Generally, the high performance of the PIC16F84 can be attributed to a number of architectural features commonly found in RISC microprocessors. Figure 22 below shows the architecture of the PIC16F84 microcontroller.

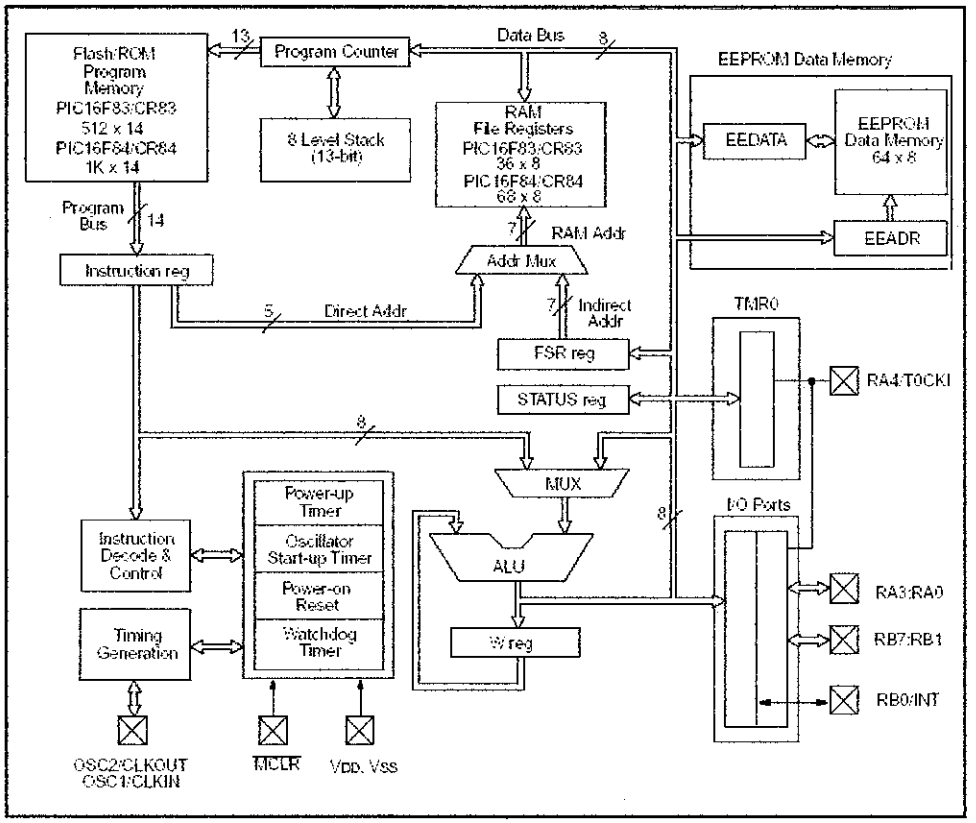


Figure 22 Simplified Block Diagram for PIC16F84

To begin with, the PIC16F84 uses Harvard architecture. This architecture has the program and data accessed from separate memories. So the device has a program memory bus and a data memory bus. This improves bandwidth over traditional von Neumann architecture where program and data are fetched from the same memory, which accesses over the same bus. Separating program and data memory further allows instructions to be sized differently than the 8-bit wide data word.

The PIC16F84 can directly or indirectly address its register files or data memory. All special function registers including the program counter are mapped in the data

memory. An orthogonal (symmetrical) instruction set makes it possible to carry out any operation on any register using any addressing mode. This symmetrical nature and lack of 'special optimal situations' makes programming with the PIC16F84 simple yet efficient.

4.2.1 PIC16F84 Microcontroller Circuitry

The microcontroller circuitry must be well designed for optimum usage. As mentioned earlier in Section 3.1, all the eight pins at PORT B are used for the inputs from the sensors. Meanwhile, the three pins at PORT A are used for the outputs and serial transmit pins. The details of the complete microcontroller circuitry are attached to Appendix C.

From this circuit, the two pins on the left of the PIC16F84 are for the input from PLC and output to PLC. The two pins on the right of the PIC16F84 are reserved for serial communication. The configuration is designed as such to provide a neat and tidy connection when constructed on the circuit board. This allows easier troubleshooting should there be any unexpected problems faced at any time.

4.2.2 PIC16F84 Microcontroller Programming

The C programming language is chosen to program the PIC16F84. The program will flow in such a way that the PIC16F84 will gather all the eight inputs into the PIC and produce an output that includes transmitting data output to the PC and vice versa. In order to run this program smoothly, several important initializations must be included. They are:

- PIC16F84 header file
- Declare the desired fuses
- Declare the usage of RS-232 which includes the baud rate, receive and transmit pins
- Define the delay or the clock oscillation speed

```
#include <16f84.h>
#fuses XT,NOPROTECT,NOWDT

#use delay(clock=4000000)
```

Initially, a flowchart of the whole operation is designed to guide the coding and the flow of the program. This flowchart shapes the programming codes in the C programming. The complete source codes are attached to Appendix D. Since the chosen data format for the data transmit are ASCII codes, the details of the universal ASCII codes are attached to Appendix E.

Before downloading a program into a PIC, the program must be verified and compiled with the C Compiler. The program will be converted into a hex file. Then, the EPROM of the PIC must be cleared and blanked. Once this stage is successful, the program can be downloaded into the PIC. This was done using the WARP-13 PIC programmer.

4.3 Serial Communication Interface

Naturally, the output from the PIC16F84 needs to be transmitted to the PC so that the user could be alerted of any unwanted incidents. Among the transmission medium available include parallel and serial communication. As explained in earlier Chapter 3, the serial port is chosen as it proves to be more feasible.

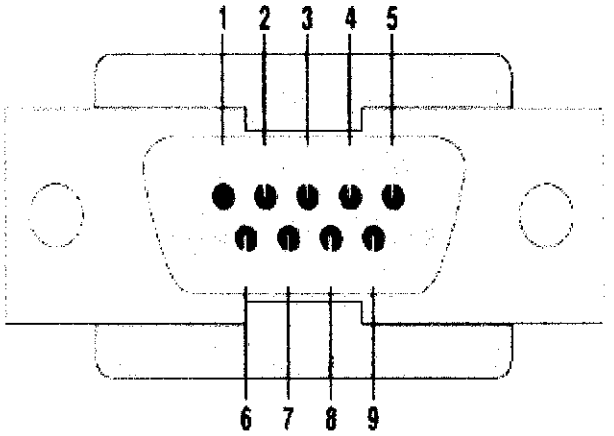
As far as interfacing between the PIC16F84 and PC is concerned, initialization is the first consideration by modifying the settings on both ends to accommodate each other. These settings include:

- Baud rate: 9600
- Parity bit: None
- Data bits: 8
- Stop bits: 1

These initializations will coincide with that of the PIC16F84. The two pins from PIC16F84 that interacts with the serial port are:

- Transmit pin (A0)
- Receive pin (A1)

Old PC's used 25 pin connectors but only about 9 pins were actually used so today most connectors are only 9-pin. Each of the 9 pins usually connects to a wire. Besides the two wires used for transmitting and receiving data, another pin (wire) is signal ground. The voltage on any wire is measured with respect to this ground. Thus the minimum number of wires to use for 2-way transmission of data is 3. Despite this method, it has also been known to work with no signal ground wire but with degraded performance and sometimes with errors.



Pin	Signal	Pin	Signal
1	Data Carrier Detect	6	Data Set Ready
2	Received Data	7	Request to Send
3	Transmitted Data	8	Clear to Send
4	Data Terminal Ready	9	Ring Indicator
5	Signal Ground		

Figure 23 RS-232 Pin Out on a DB-9 Pin Used for Asynchronous Data

Almost all digital devices used nowadays require either TTL or CMOS logic levels. Therefore the first step to connecting a device to the RS-232 port is to transform the RS-232 levels back into 0 and 5 Volts. This is done by RS-232 level converters. Two common RS-232 level converters are the 1488 RS-232 driver and the 1489 RS-232 receiver. Each package contains 4 inverters of the one type, either drivers or receivers. The driver requires two supply rails, +7.5V to +15V and -7.5V to -15V. As expected, this may pose a problem in many instances where only a single supply of +5V is present. However the advantages of these IC's are they are cheap.

Besides the RS-232, another device is the MAX-232. It includes a charge pump, which generates +10V and -10V from a single 5V supply. This IC also includes two receivers and two transmitters in the same package. This is handy in many cases when only the Transmit and Receive data Lines are needed. This eliminates the usage of two chips, one for the receive line and one for the transmit. However all this convenience comes at a price, but compared with the price of designing a new power supply it is very cheap. The details of the serial circuitry are attached to Appendix F.

4.4 GUI interface – Microsoft Visual Basic

Microsoft Visual Basic is a programming tool which is capable of both serial and parallel communications. Beside that, it is also has features that allow interfacing with other Microsoft Office applications such as Microsoft Access. Furthermore, there is also a component which monitors the surveillance operations. In order to initialize these features, Microsoft Comm Control 6.0 must be added to the VB.

Microsoft Comm Control 6.0

In order for Visual Basic to establish connection with the serial port, a component called the Microsoft Comm Control 6.0 is applied. This component is represented by an icon of a phone.

The portion of the source codes that initializes the serial port using the Microsoft Comm Control 6.0 is presented below:

```
Private Sub Form_Load()
   'Initialize
   Comm.InputMode = 0 'take ASCII codes as inputs
   Comm.CommPort = 1
   Comm.Settings = "9600,N,8,1"
   Comm.PortOpen = True 'open port
   Comm.InputLen = 1 'limitation for input
   Comm.RThreshold = 1

End Sub
```

The comm settings will coincide with the serial port settings which are:

- Baud rate: 9600
- Parity bits: None
- Data bits: 8
- Stop bit: 1

4.5 Constraints and Problems Encountered

Initially, the PIC programming was done using assembly language but there were some difficulties faced to transmit data into the MAX232 for serial communication. Instead, the C programming was designed and implemented, as it is known to be less complicated than the assembly language. Furthermore, the WARP-13 programmer available can be quite tricky. The process of burning the program into the PIC may take many attempts as the programmer is prone to errors.

The circuitries also have to go through a series of troubleshooting. Upon checking every connections and components, some components were found to be faulty. Another constraint faced was the fact that there is only one MAX232 chip available. Therefore, the chip had to be shared among the students using it.

.

4.6 Comparison with other technology

Toyota ASSB factory in Shah Alam did have pallet transfer in their factory and in their department called Body Shop (stamping). The technology of the pallet transfer is simple and beneficial to them. Figure 15 shows the mechanism of the pallet transfer.

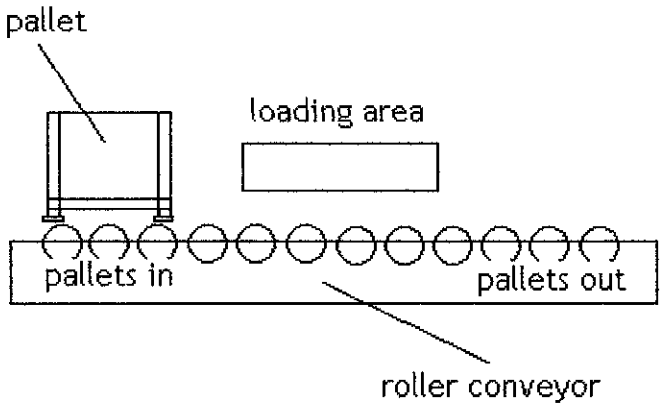


Figure 24 Pallet Transfer in Toyota ASSB

The concept of the diagram is to show the mechanism of the pallet transfer in Toyota ASSB. The pallet transfer is more towards conveyor function type. One pallet will be placed on the early section of the roller conveyor and will transfer to the center where the loading of body parts will take place. Fully filled pallet will transfer to the end of the roller conveyor where forklift truck will collect them. Although the system is simple, but it still advance from local automotive industries where could not have a conveyor at least.

The Pallet Transfer System which proposed will have more functions in term of mechanism and more efficient in pallets management and time consuming for the Proton's employees.

CHAPTER 5
CONCLUSION

This project at first looks easy as it consists of only PLC; it gets tougher along the way as many problems appear mainly regarding the availability of components, equipments, and materials. At the end, all of the problems faced were encounter and the project was brilliant and meets the collaborator's needs.

Looking from the economic perspective, this project is definitely cost effective, as it comprises of components that are generally low in terms of cost. Furthermore, they are also widely available in the market. This certainly proves to be a boost as this system is not only practical, but also competitive in performance and cost. Though this system is small scale at this stage, with some improvisations, it can adapt and be implemented in various domestic and industrial operations. For example, this system is applicable in monitoring the manufacturing machines should there be any faulty or damage to a particular section. A taste of change includes implementing this system to monitor a particular parameter, such as a closed room.

REFERENCES

1. John B. Peatman, *DESIGN with PIC MICROCONTROLLERS*, Prentice Hall, Upper Saddle River, New Jersey, 1998
2. Wallace Wang, *Visual Basic 6 for Dummies*, IDG Books Worldwide, 1998
3. Carl Bergquist, *Guide to PICMICRO Microcontrollers*, Prompt Publications, USA, 2001
4. Evangelos Petroutsos, *Database Programming with Visual Basic 6*, SYBEX Inc., USA, 2000
5. Mike Tooley, *PC-based Instrumentation and Control*; 2nd Edition, Newnes, Great Britain, 1991
6. Myke Predko, *PC Ph.D. Inside PC Interfacing*. McGraw Hill, Washington DC, 2000
7. David I. Schneider, *An Introduction to Programming Using Visual Basic 6.0*. Fourth edition, Prentice-Hall, 1995
8. PE Stamping layout design.
9. Devices Specification –Press & Tool Department 1995.
10. PLC for beginners – William Stallings
11. The Toyota Way – Jeffrey K. Liker – Mc Graw Hill
12. Mechanical Bearings – Singiresu S. Rad – Prentice Hall
13. *MSComm Control and Visual Basic*
    http://www.ontrak.net
14. *MAX232 and Serial Communications*
    http://www.microchipc.com/PIC16bootload

APPENDICES

APPENDIX A: PLC Ladder Diagram

APPENDIX B: AutoCAD Drawing

APPENDIX C: Microcontroller Circuitry

APPENDIX D: Microcontroller Programming

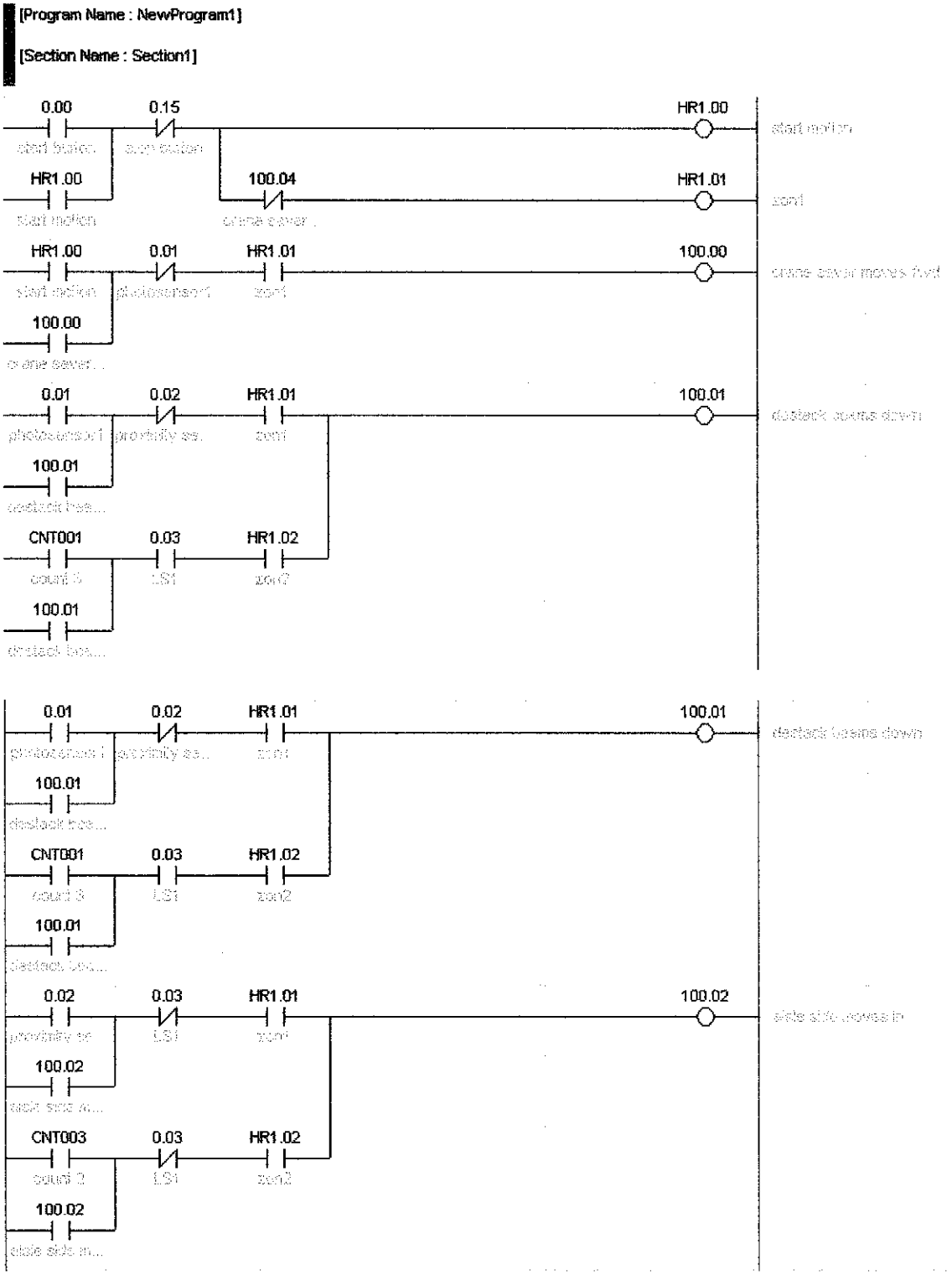
APPENDIX E: ASCII Code Table

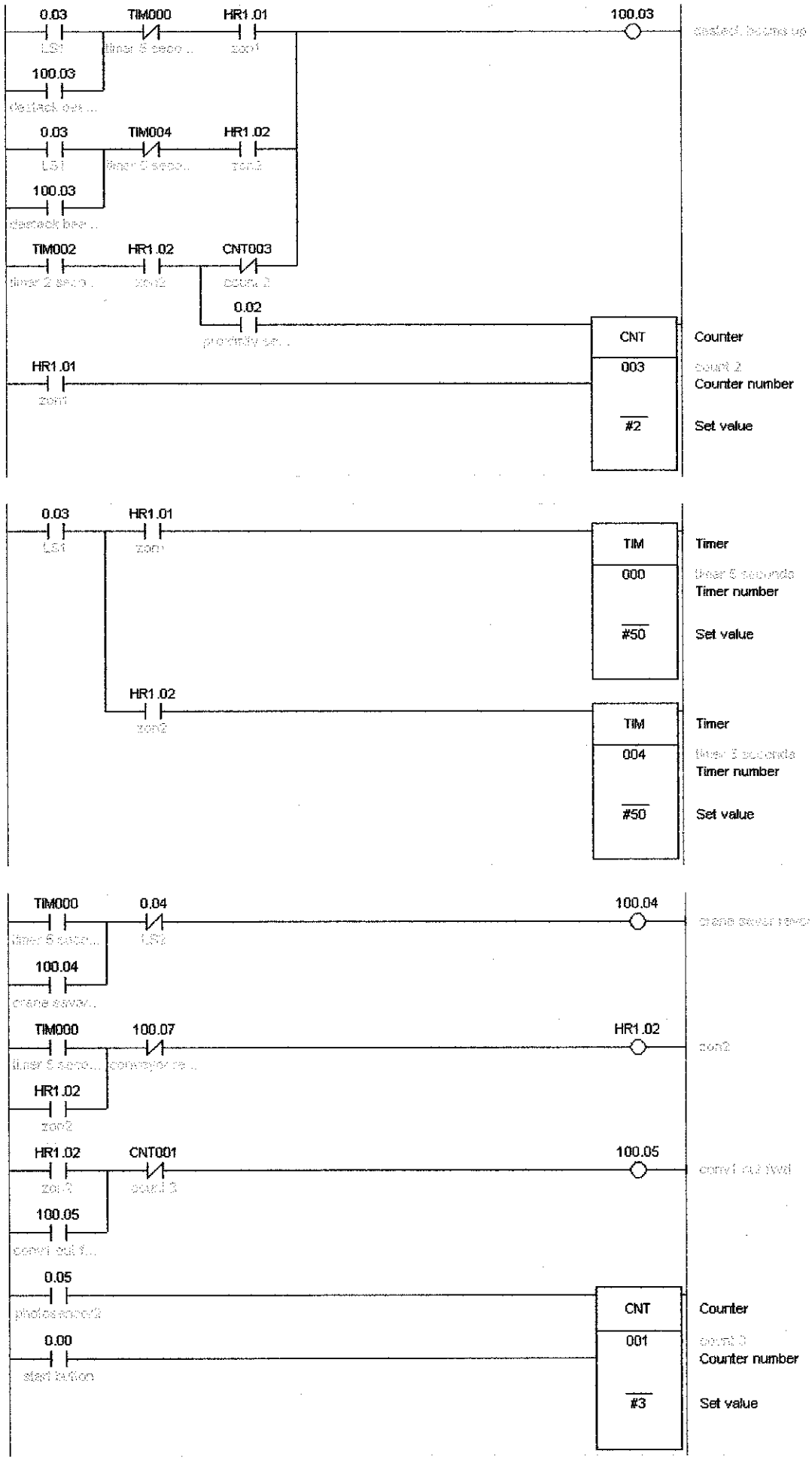
APPENDIX F: RS232 Circuitry

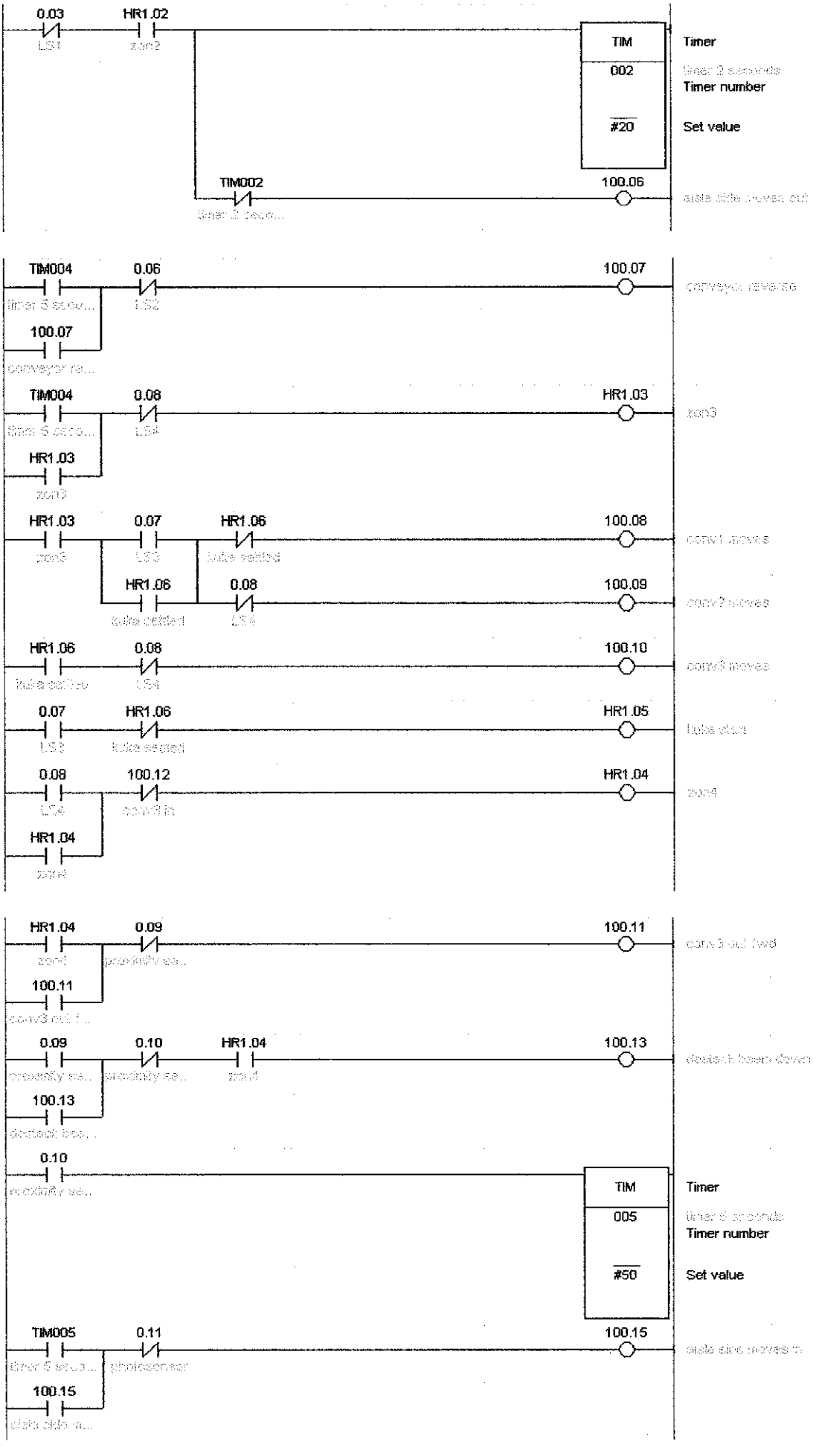
APPENDIX G: GUI Windows

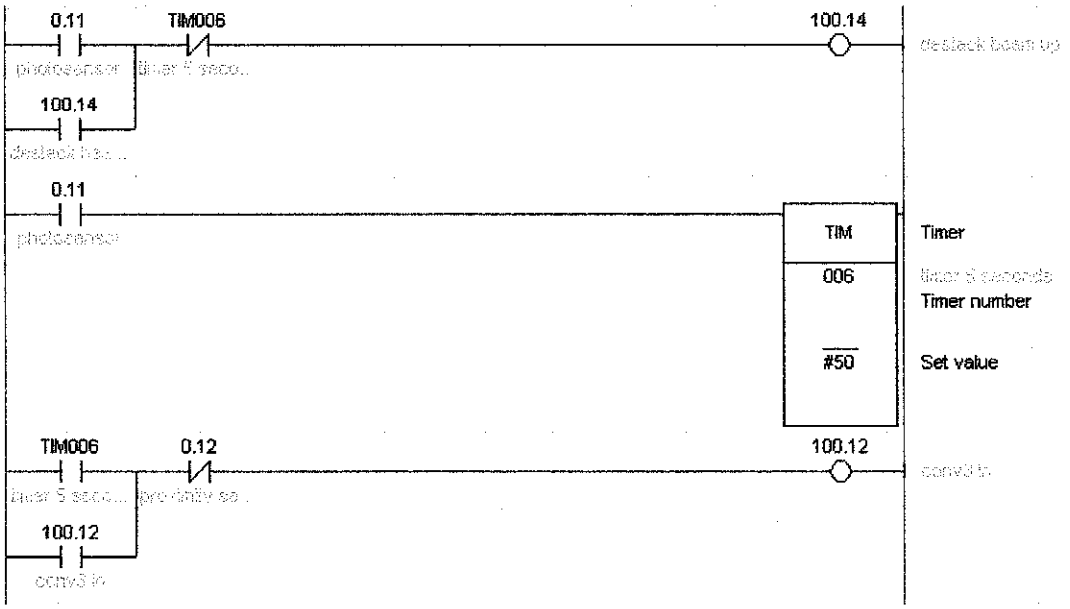
APPENDIX A

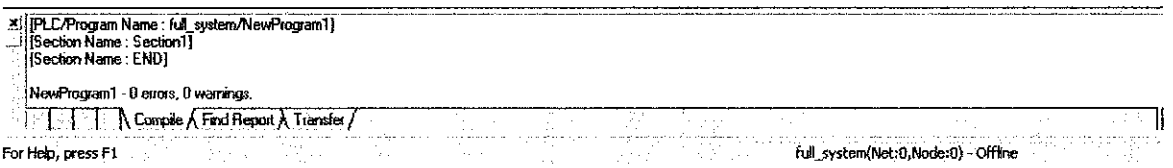
PLC LADDER DIAGRAM




0.03
TIM000
HR1.01
100.03
100.03
0.03
TIM004
HR1.02
100.03
TIM002
HR1.02
CNT003
0.02
CNT
Counter
003
Counter number
#2
Set value
HR1.01
0.03
HR1.01
TIM
Timer
000
Timer number
#50
Set value
HR1.02
TIM
Timer
004
Timer number
#50
Set value
TIM000
0.04
100.04
100.04
TIM000
100.07
HR1.02
HR1.02
HR1.02
CNT001
100.05
100.05
0.05
CNT
Counter
001
Counter number
#3
Set value
0.00

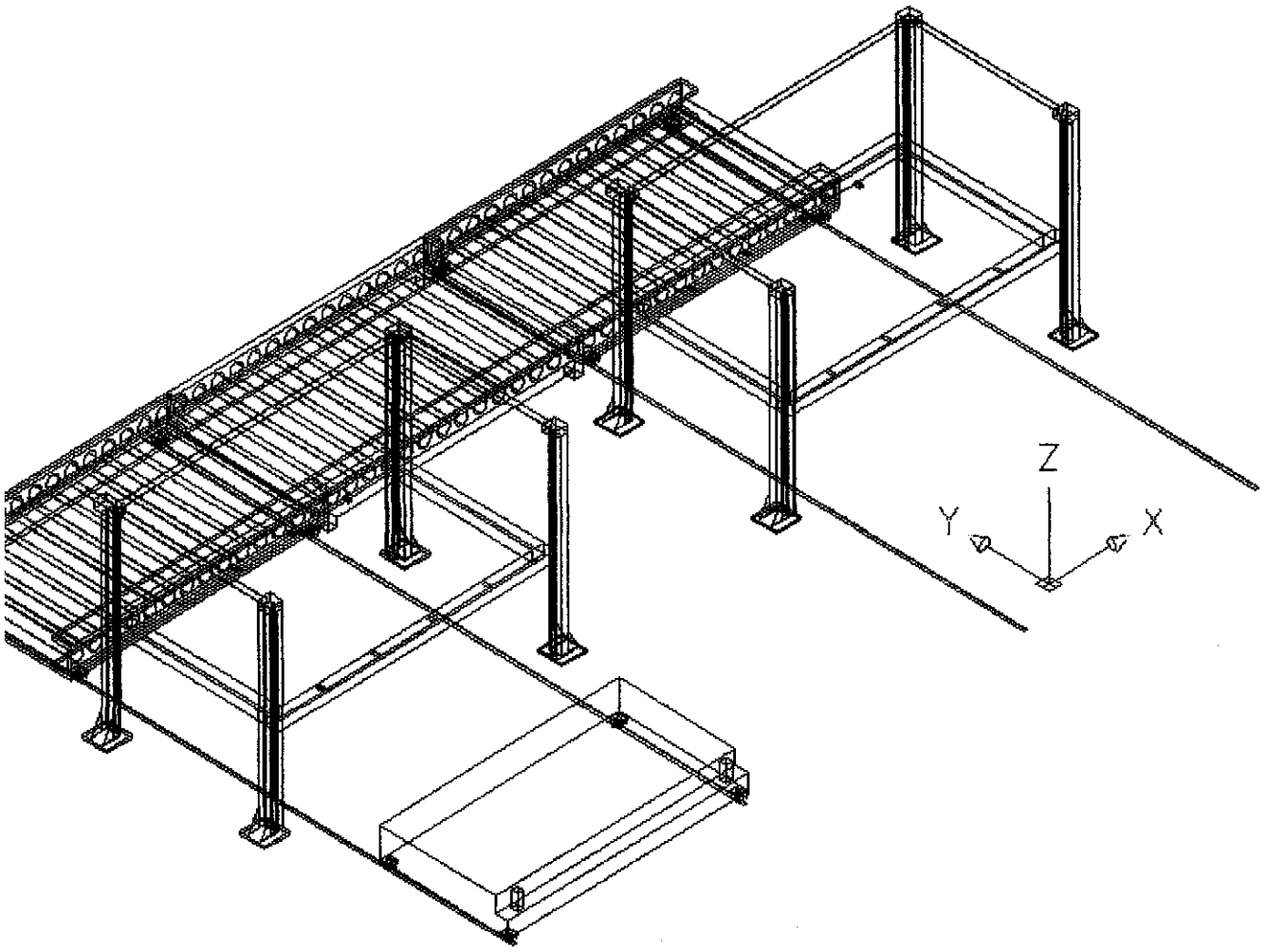

0.03
LS1
HR1.02
zon2
TIM
002
#20
Timer
Timer number
Set value
TIM002
100.06
TIM004
0.06
LS2
100.07
100.07
TIM004
0.08
LS4
HR1.03
HR1.03
HR1.03
0.07
LS3
HR1.06
100.08
HR1.06
0.08
100.09
HR1.06
0.08
100.10
0.07
LS3
HR1.06
HR1.05
0.08
LS4
100.12
HR1.04
HR1.04
HR1.04
0.09
100.11
100.11
0.09
0.10
HR1.04
100.13
100.13
0.10
TIM
005
#50
Timer
Timer number
Set value
TIM005
0.11
100.15
100.15


0.11
TIM006
100.14
100.14
0.11
TIM
Timer
006
Timer number
#50
Set value
TIM006
0.12
100.12
100.12


[PLC/Program Name : full_system/NewProgram1]
[Section Name : Section1]
[Section Name : END]
NewProgram1 - 0 errors, 0 warnings.
Compile
Find Report
Transfer
For Help, press F1
full_system(Net:0,Node:0) - Offline

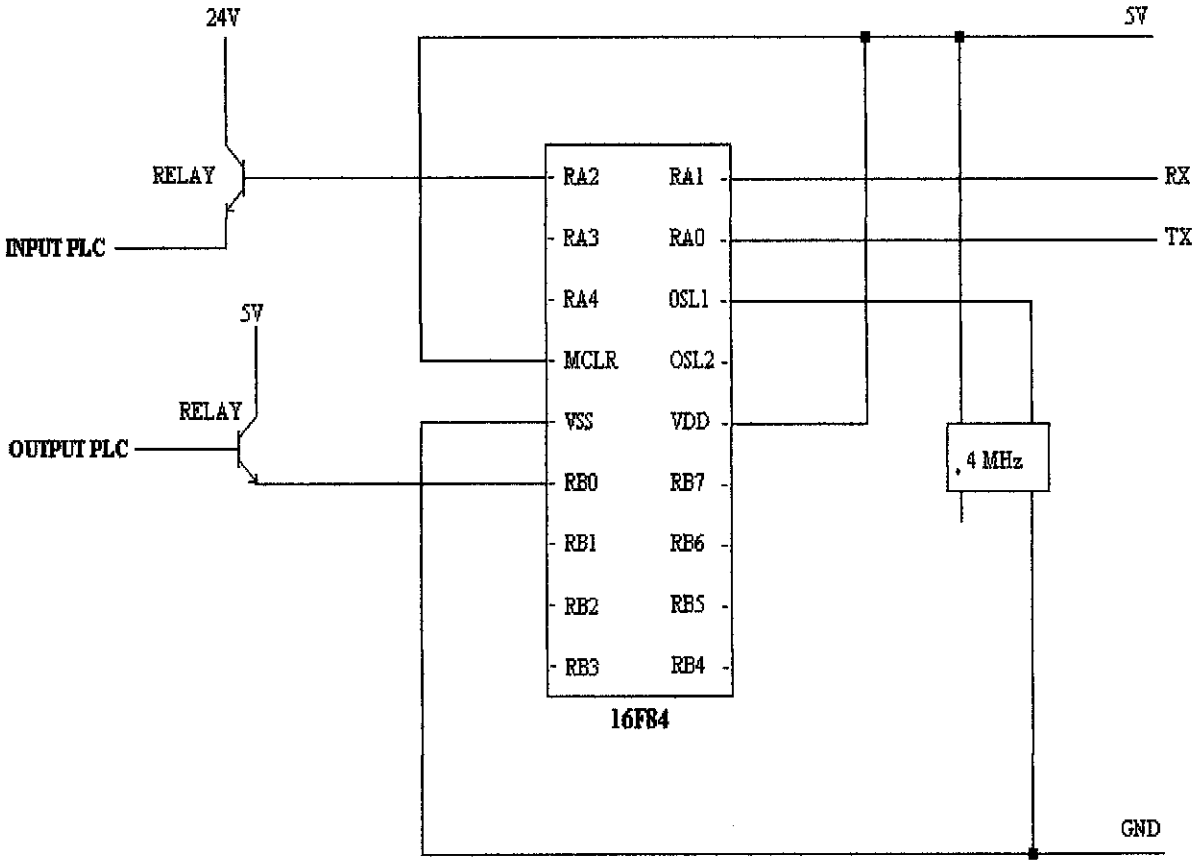
APPENDIX B

AUTOCAD DRAWING



APPENDIX C

MICROCONTROLLER CIRCUITRY



APPENDIX D

MICROCONTROLLER PROGRAMMING

```
#include <16F84a.H>
#use delay(clock= 4000000)
#use rs232(baud=9600, xmit=PIN_A0, rcv=PIN_A1)
#fuses XT,NOPROTECT,NOWDT


main()
{
char k;                    //assign integer 'k'
k = getch();               // character 'k' will act as get character value
  do
  {
  set_tris_a(0x00);
  set_tris_b(0x00);

  if (input(PIN_B0)!=0)    // if sensor detects obstacle
  {
  putchar(65);             // master will send character 'A' to CPU
  }
  else if (k==6)           // if master get character 'B'
  {
  output_high(PIN_A2);     // send high to the output
  }
  } while(1); }
```

APPENDIX E

ASCII CODES

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040		Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS (backspace)	40	28	050	(	(	72	48	110	H	H	104	68	150	h	h
9	9	011	TAB (horizontal tab)	41	29	051	)	)	73	49	111	I	I	105	69	151	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	;	91	5B	133	[	[	123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174			|
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135	]	]	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177	&#127;	DEL

Source: www.asciitable.com

ASCII Code Table

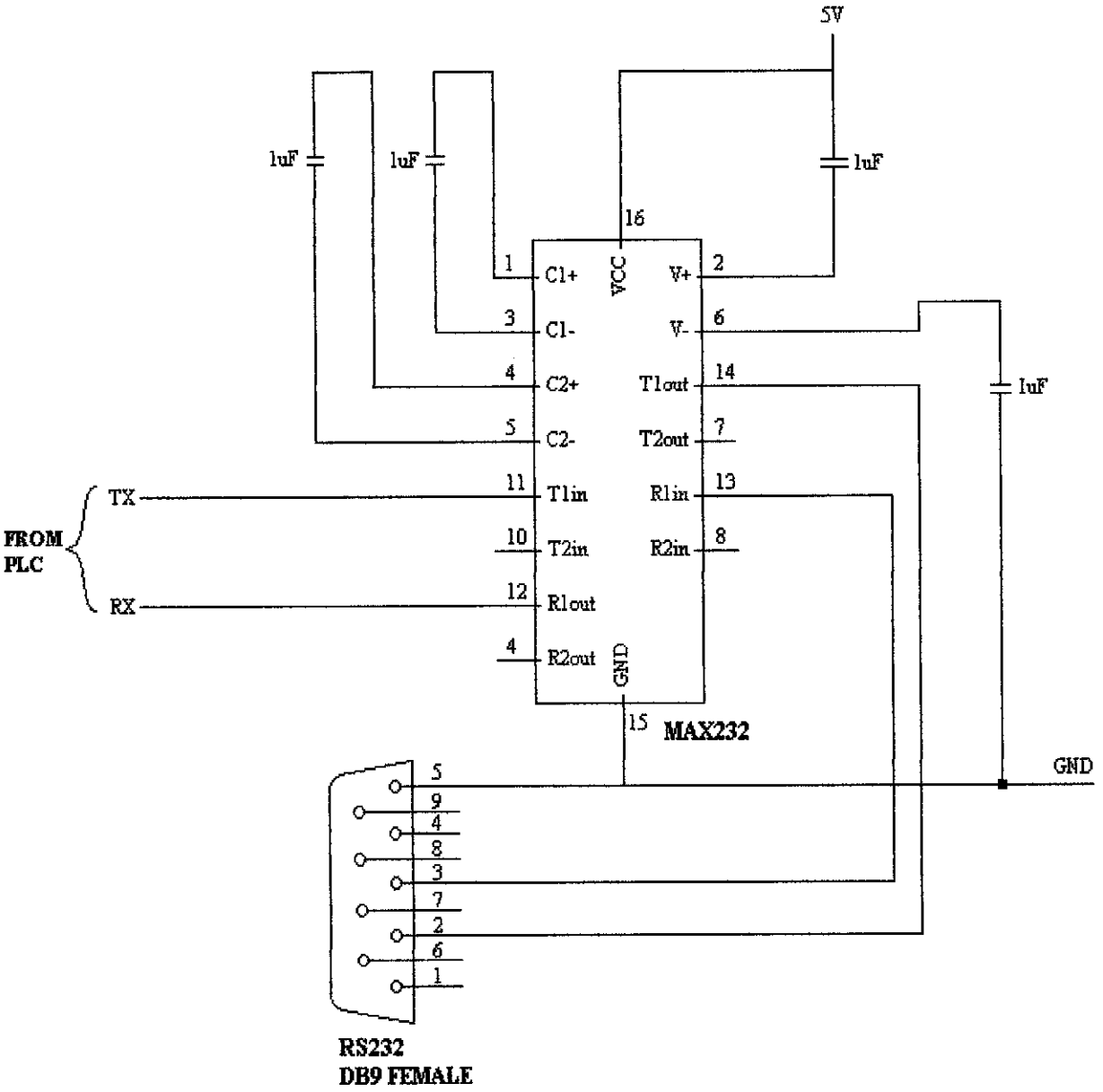
128	Ç	144	É	160	á	176	▒	193	┴	209	╤	225	ß	241	±
129	ü	145	æ	161	í	177	▓	194	┬	210	╥	226	Γ	242	≥
130	é	146	Æ	162	ó	178	▓	195	├	211	╙	227	π	243	≤
131	â	147	ô	163	ú	179	|	196	─	212	╘	228	Σ	244	⌠
132	ä	148	ö	164	ñ	180	┤	197	┼	213	╒	229	σ	245	⌡
133	à	149	ò	165	Ñ	181	╡	198	╞	214	╓	230	µ	246	÷
134	å	150	û	166	ª	182	╢	199	╟	215	╫	231	τ	247	≈
135	ç	151	ù	167	º	183	╖	200	╚	216	╪	232	Φ	248	°
136	ê	152	_	168	¿	184	╕	201	╔	217	┘	233	Θ	249	·
137	ë	153	Ö	169	_	185	╣	202	╩	218	┌	234	Ω	250	·
138	è	154	Ü	170	¬	186	║	203	╦	219	▉	235	δ	251	√
139	ï	156	£	171	½	187	╗	204	╠	220	▄	236	∞	252	_
140	î	157	¥	172	¼	188	╝	205	═	221	▌	237	ϕ	253	²
141	ì	158	_	173	¡	189	╜	206	╬	222	▐	238	ε	254	■
142	Ä	159	ƒ	174	«	190	╛	207	╧	223	▀	239	∩	255	
143	Å	192	└	175	»	191	┐	208	╨	224	α	240	≡		

Source: www.asciitable.com

ASCII Code Table (Extended List)

APPENDIX F

RS232 CIRCUITRY



APPENDIX G

GUI WINDOWS