THE UNIVERSITY OF ZAMBIA

SCHOOL OF ENGINEERING

DEPARTMENT OF ELECTRICAL AND ELECTRONIC ENGINEERING

PROGRAMMEABLE LOGIC CONTROLLERS [PLC] INTERFACE TO THE ROBOT.

BY

CHILESHE MARTIN

"Report submitted in partial fulfillment of the requirements for the degree of Bachelor of Engineering, University of Zambia."

3RD APRIL 2009

DEDICATION

To my mother Justina Mumba Chileshe, my girl friend Mutinta Hangoma and the rest of my family members who gave me support during my studies.

ACKNOWLEDGEMENTS

A study and compilation of this kind would not be a reality without reference to the work of others and without the help of my God. During the study of this project, I have had to consult constantly numerous encyclopedias, textbooks and key people on a variety of subjects; and I hereby pass my acknowledgement all them.

I wish to record my thanks and gratitude to my project supervisor, Mr. V. NYIRENDA of the Department of Electrical and Electronic Engineering, at the University of Zambia, for his rendered assistance, support and encouragement in the preparation of this work.

I am also deeply indebted to Electrical and Electronic Engineering lecturers for their guidance which they were always ready to give during the oral presentations of the project.

Special thanks go to Electrical Engineering class of 2008/9 for having contributed in one way or the other to my work. To my family for being supportive during my times on campus, I say 'Thank You'.

And last but not the least to my girl friend, Mutinta Mangoma, thank you for being there to support and encourage me though and though.

Table of Contents

LIST OF FIGURES

LIST OF TABLES

NOMENCLATURE OR NOTATION

A	Amperes
bps	Bits per second
BNC	Broadband Network Cable
CANbus	Controller Area Network bus
CSMA/BA	Collision Sense Multiple Asses/Bitwise Arbitration
CSMA/CD	Collision Sense Multiple Asses/Collision Detection
CTDMA	Concurrent Time Domain Multiple Access
DC	Direct Current
FSK	Frequency Shift Key
HMI	Human Machine Interface
I/O	Input/output
LAN	Local Area Network
LCD	Liquid Crystal Display
LED	Light Emitting Diode
MI	Memory Integer
npn	Negative Positive Negative junction
NSI	Network System Integer
OPLC	Operating panel Programmable Logic Controllers
OSI	Open System Interconnection
PC	Personal Computer
PLC	Programmable Logic Controller
pnp	positive negative positive junction
R	Repeater
SDW	System Double Word
SI	System Integer
V	Volts
WAN	Wide Area Network

SUMMARY

Every process of industry, in developed countries, from power generation, automobile painting to food packaging uses programmable logic controllers to expand and enhance production. There is need to introduce PLCs and networking in our Zambian industries dealing with food and beverage packaging.

This project aims at observing the food and beverage packaging system in Zambian industries using PLCs and appropriate hardware interface. The system should automatically detect the data like availability of parts in the pick-up fixture, weight of the container after loading, weight of each part and then relay this information to a central control centre for controlling purposes.

The system uses Devicenet for fast data and instruction transmission between the PLC and the Robot or any other machine in this regard.

CHAPTER 1

INTRODUCTION

1.1 Report Organization

This report which is divided into five chapters is a final report whose aim is to describe the work done in EE590 final year project. The first part of the report i.e. part before chapter 1 contains the dedications, acknowledgements, table of contents, list of figures, nomenclature and summary. The reference is at thc end of the report after the recommendations.

Chapter one gives a general overview of the subject i.e. problem statement, rationale, objectives and methodology. A general layout of the LOADING system and its operation is outlined herein.

Chapter two gives a detailed description of the structure of a Networking. This includes items like the classification of Networks, basic control systems and the types of communication and networking to be used.
It also explains PLC control system and specifically describing the Unitronics OPLC, its functions and the kinds of communication features it offers.

Chapter three gives the planning of the works, configurations done on the PLC module. It further gives a description of the communication methods selected and the overall system integration.

The discussion is then given in chapter four outlining the data analysis in relation to the work done and the cost of the project. This is then followed by conclusions and recommendations in chapter five.

1.2 Background

Every aspect of industry, in developed countries, from power generation, automobile painting to food packaging uses programmable logic controllers to expand and enhance production.
There is need to introduce PLCs and networking in our Zambian industries dealing with food and beverage packaging.

This project aims at observing the food and beverage packaging system in Zambian industries using PLCs and appropriate hardware interface. The system should automatically detect the data like availability of parts in the pick-up fixture, weight of the container after loading, weight of each part and then relay this information to a central control centre for controlling purposes.

The purpose of this case study, to the consumer of food and beverages, is to deliver rightful amount of specific food or beverage packed in the right containers and with the right prescribed weight. To the producer, this would help to increase production of the food and beverages, reduce on the error in packing (less cost on man power and machinery, since it will be using Device net, thus more profits) and reduce on the down-time of the plant in terms of faults in the machinery.ie since detections of faults will be easy to follow with PLC programming.

1.3 PROBLEM STATEMENT

The current status of food and beverage packaging in our industries exhibits high unreliability and inefficiency. Production personnel need to control this process of packaging through an automated network from a remote computer.
The robot will be loading parts i.e. food or beverages from the part Pick-Up fixture into a container until it reaches a prescribed weight. PLCs will be used to control the parameters such as weight of the container when it has filled to the right volume, they will also be used to feed parts in the Pick-Up fixture when it is empty. Apart from that, the PLC will be used to detect when the part is ready for pick up in the part pick-up fixture by detecting its weight.
PLCs will then be used with the proper interface to communicate to the robot when to pick up the part and load it in the container.
The main aim is to design and create a PLC ladder logic program that will be used to control a robot in the plant from a remote computer and ensure that the End Effect of this robot does not

change with time and restrict it to a specific work space. The devicenet network will then be used for fast data transmission between the robot and the PLC

1.4 RATIONALE

In light of the prevailing growth of beverage industry, with packaging hardships, considerations of technologies that would make the system better and fast in terms of packaging of products must not be ignored.

A continuous and defined control system is essential for efficient performance of this section of packaging industry. Therefore there is need to have a definite system of controlling the packaging parameters in order to address the above problems. This is to mean that the weight of the container on loading is to be closely monitored and therefore avoid such things as over loading or under loading of food and/or beverage in container which has become rampant in most companies.

1.5 OBJECTIVES

This project aims at:

- (i) Designing and fabricating a system to remotely control the food and beverage packaging system using PLCs and appropriate hardware interfaces.
- (ii) Use devicenet for data transmission between the robot and the PLC
- (iii) Testing the prototype of the control system.

The parameter considered for controlling is weight of the container when loaded, presence of parts in the Pick-Up fixture and the weight of each part. Therefore the design based on the problems initially defined in the project as well as field study results. The testing of the system has done in order to verify the design specifications. This involves transduction, processing the input data by the PLC application program, communication and encompassing devicenet network to transmit instructions to and from the robot.

1.6 METHODOLOGY

The following approach was taken in order to reach the aims of the project:

(i) Literature review of the project.

(ii) The study of the software tool (VisiLogic) used to create control projects for vision controllers was carried out.

(iii) Method of transduction to be used for the parameter to be controlled.

(iv) Particular attention is paid to the weight sensing for the box since an understanding of the techniques used on this one can easily be extended to that of each part. Laboratory tests are to be carried out on the PLC in order to study the output characteristics and condition the output signal accordingly. The conditioning of the input signal was done in order to produce correct PLC inputs.

(v) Visited the Zambia Breweries and Zambia Bottlers companies for consultations on the methods of packaging that are currently existing in order understand the system used in controlling their distributed system.

The figure below shows the system lay out, all its operation for proper packaging and control;

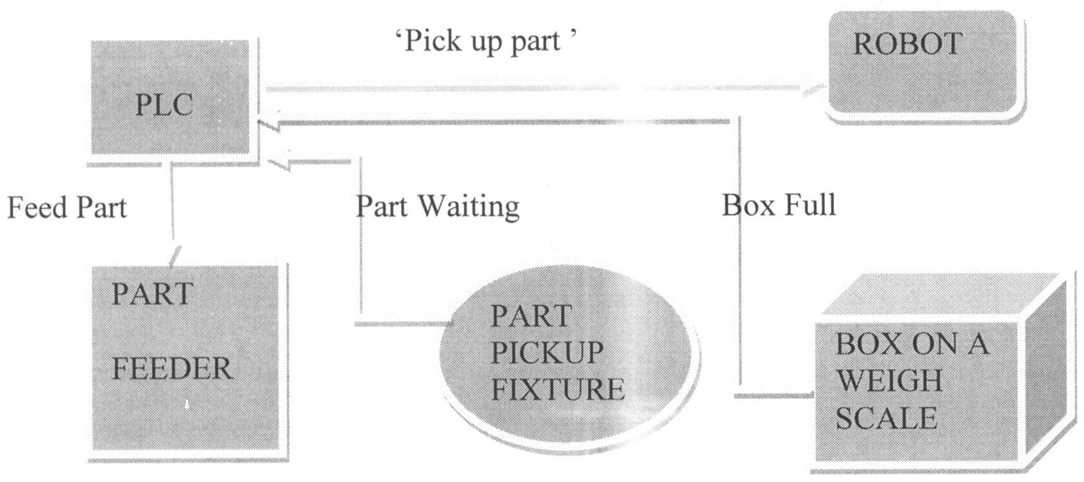


Figure1.1: System Lay out

CHAPTER 2

NETWORKING AND CONTROLING SYSTEMS

2.1. Background

Networking is defined as the process by which data is transmitted between systems. Typical advantages of networks include resource sharing and ease of communication.

Small networks are often called Local Area Networks (LANs). These may connect a few hundred computers within a distance of hundreds of meters. These networks are inexpensive. Data can be transmitted at rates of millions of bits per second. Many controls system are using networks to communicate with other controllers and computers. Typical applications include;

- taking quality readings with a PLC and sending the data to a database computer
- distributing recipes or special orders to batch processing equipment
- Remote monitoring of equipment.

Larger Wide Area Networks (WANs) are used for communicating over long distances between LANs. These are not common in controls applications, but might be needed for a very large scale process. An example might be an oil pipeline control system that is spread over thousands of miles.

2.2 Topology

The structure of a network is called the topology. Figure 2.1 shows the basic network topologies. The *Bus* and *Ring* topologies both share the same network wire. In the *Star* configuration each computer has a single wire that connects it to a central hub.

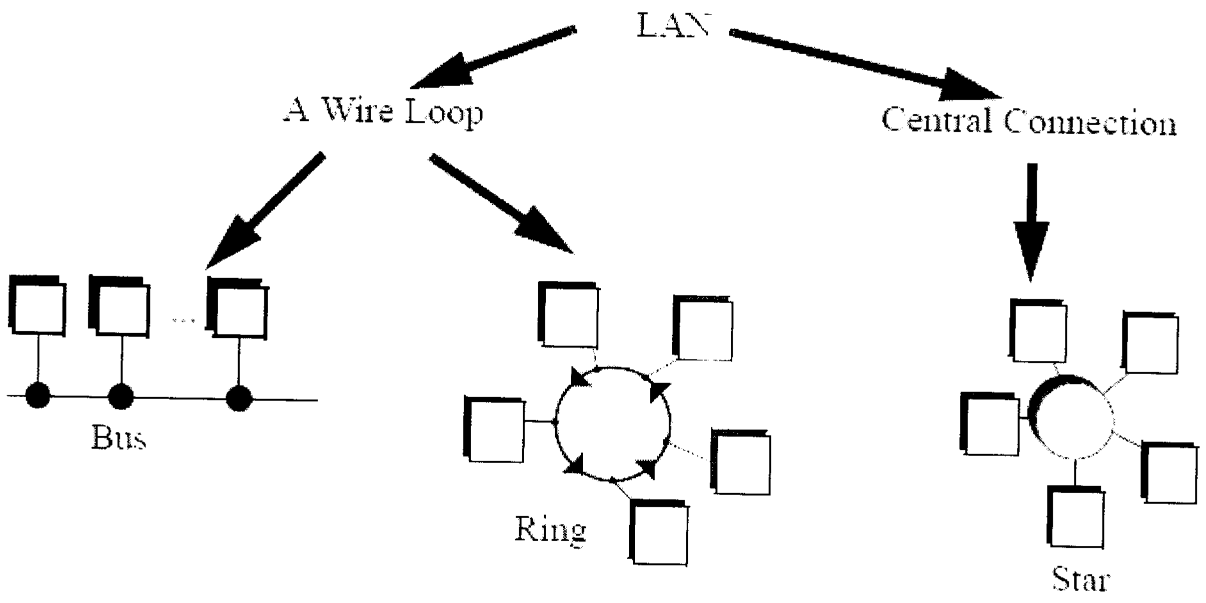


Figure 2.1: Network Topologies

In the *Ring* and *Bus* topologies the network control is distributed between all of the computers on the network. The wiring only uses a single loop or run of wire. But, because there is only one wire, the network will slow down significantly as traffic increases. This also requires more sophisticated network interfaces that can determine when a computer is allowed to transmit messages. It is also possible for a problem on the network wires to halt the entire network.

The *Star* topology requires more wire overall to connect each computer to an intelligent hub. But, the network interfaces in the computer become simpler, and the network becomes more reliable. Thus it is deterministic, meaning that performance can be predicted. This can be important in critical applications.

For a factory environment the bus topology is popular. The large number of wires required for a star configuration can be expensive and confusing. The loop of wire required for a ring topology is also difficult to connect, and it can lead to ground loop problems. Figure 2.2 shows a tree topology that is constructed out of smaller bus networks.

Repeaters are used to boost the signal strength and allow the network to be larger.

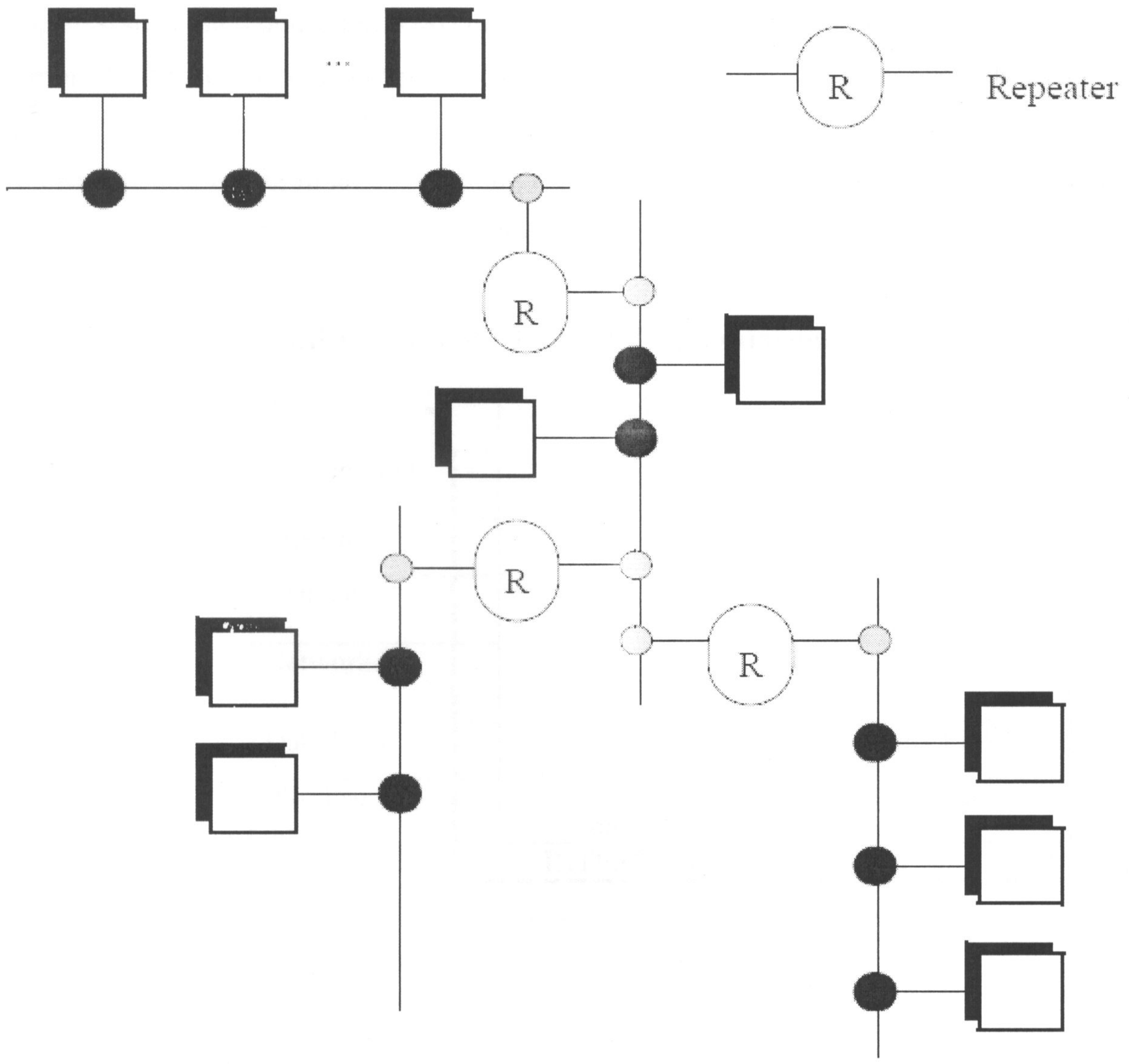


Figure2.2 Tree Topology

.3 OSI Network Model

he Open System Interconnection (OSI) model in Figure 2.3 was developed as a tool to describe e various hardware and software parts found in a network system. The model contains seven ayers, with the hardware at the bottom, and the software at the top. The darkened arrow shows hat a message originating in an application program in computer #1 must travel through all of he layers in both computers to arrive at the application in computer #2.

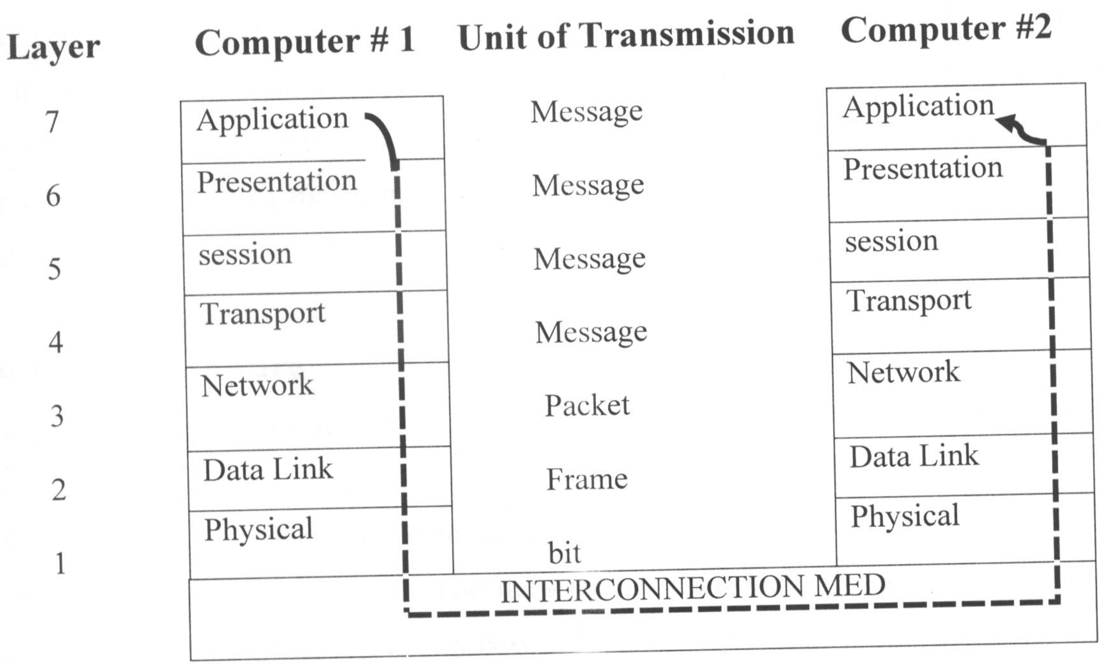


Figure 2.3: OSI Network Model [ ]

Application	- This is high level software on the computer.
Presentation	- Translates application requests into network operations.
Session	- This deals with multiple interactions between computers.
Transport	- Breaks up and recombines data to small packets.
Network	- Network addresses and routing added to make frame.
Data Link	- The encryption for many bits, including error correction added to a frame.
Physical	- The voltage and timing for a single bit in a frame.

Interconnecting Medium - The wires or transmission medium of the network.

The *Physical* layer describes items such as voltage levels and timing for the transmission of single bits. The *Data Link* layer deals with sending a small amount of data, such as a byte, and error correction. Together, these two layers would describe the serial byte shown in the previous chapter. The *Network* layer determines how to move the message through the network. If this were for an internet connection this layer would be responsible for adding the correct network address. The *Transport* layer will divide small amounts of data into smaller packets, or recombine them into one larger piece. This layer also checks for data integrity, often with a checksum. The *Session* layer will deal with issues that go beyond a single block of data. In particular it will deal with resuming transmission if it is interrupted or corrupted. The *Session* layer will often make long term connections to the remote machine. The *Presentation* layer acts as an application interface so that syntax, formats and codes are consistent between the two networked machines.

2.4 Networking Hardware

The following is a description of most of the hardware that will be needed in the design of networks.

• Computer (or network enabled equipment)

• Network Interface Hardware - The network interface may already be built into the computer/PLC/sensor/etc. These may cost ZMK150, 000 to over ZMK1, 000, 000.

• The Media - The physical network connection between network nodes. 10baseT (twisted pair) is the most popular. It is a pair of twisted copper wires terminated with an RJ-45 connector.

10base2 (thin wire) is thin shielded coaxial cable with BNC connectors

10baseF (fiber optic) is costly, but signal transmission and noise properties are very good.

• Repeaters (Physical Layer) - These accept signals and retransmit them so that longer networks can be built.

• Hub/Concentrator - A central connection point that network wires will be connected. It will pass network packets to local computers or to remote networks if they are available.

• Router (Network Layer) - Will isolate different networks, but redirect traffic to other LANs.

• Bridges (Data link layer) - These are intelligent devices that can convert data on one type of network, to data on another type of network. These can also be used to isolate two networks.

• Gateway (Application Layer) - A Gateway is a full computer that will direct traffic to different networks, and possibly screen packets. These are often used to create firewalls for security.

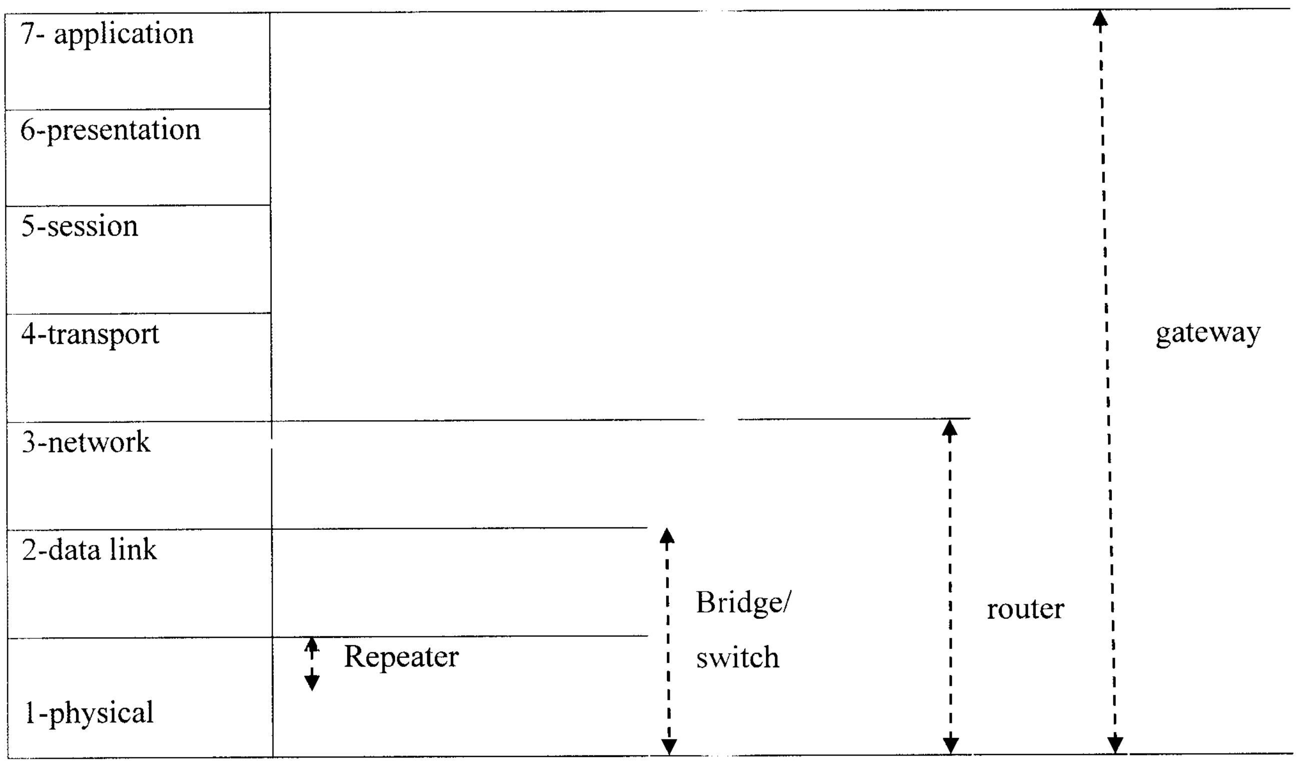


Figure 2.4: Network Device and the OSI Model

2.5 NETWORK STANDARDS

2.5.1 Devicenet

Devicenet has become one of the most widely supported control networks. It is an open standard, so components from a variety of manufacturers can be used together in the same control system. This network has advantages over the other network because it has been designed to be noise resistant and robust. One major change for the control engineer is that the PLC chassis can be eliminated and the network can be connected directly to the sensors and actuators reducing the total amount of wiring by moving I/O points closer to the application point. This can also simplify the connection of complex devices, such as HMIs. Two way communications inputs and outputs allow diagnosis of network problems from the main controller.

Devicenet covers all seven layers of the OSI standard. The protocol has a limited number of network addresses, with very small data packets and this helps limit network traffic and ensure responsiveness. The length of the network cables will limit the maximum speed of the network. The basic features of are listed below;

- A single bus cable that delivers data and power.
- Up to 64 nodes on the network.
- Data packet size of 0-8 bytes.
- Lengths of 500m/250m/100m for speeds of 125kbps/250kbps/500kbps respectively.
- Devices can be added or removed while power is on.
- Based on the CANbus (Controller Area Network) protocol for OSI levels 1 and2.
- Addressing includes peer-to-peer, multicast, master/slave, polling or change of state.

An example of a Devicenet network is shown in Figure 2.5. The dark black lines are the network cable. Terminators are required at the ends of the network cable to reduce electrical noise. In this case the PC would probably be running some sort of software based PLC program. The computer would have a card that can communicate with Devicenet devices. The *FlexIO rack* is a miniature rack that can hold various types of input and output modules. Power taps (or tees) split the signal to small side branches. In this case one of the taps connects a power supply, to provide the 24Vdc supply to the network. Another two taps are used to connect a *smart sensor* and another *FlexIO rack*.

The *Smart sensor* uses power from the network, and contains enough logic so that it is one node on the network. The network uses *thin trunk line* and *thick trunk line* which may limit network performance.

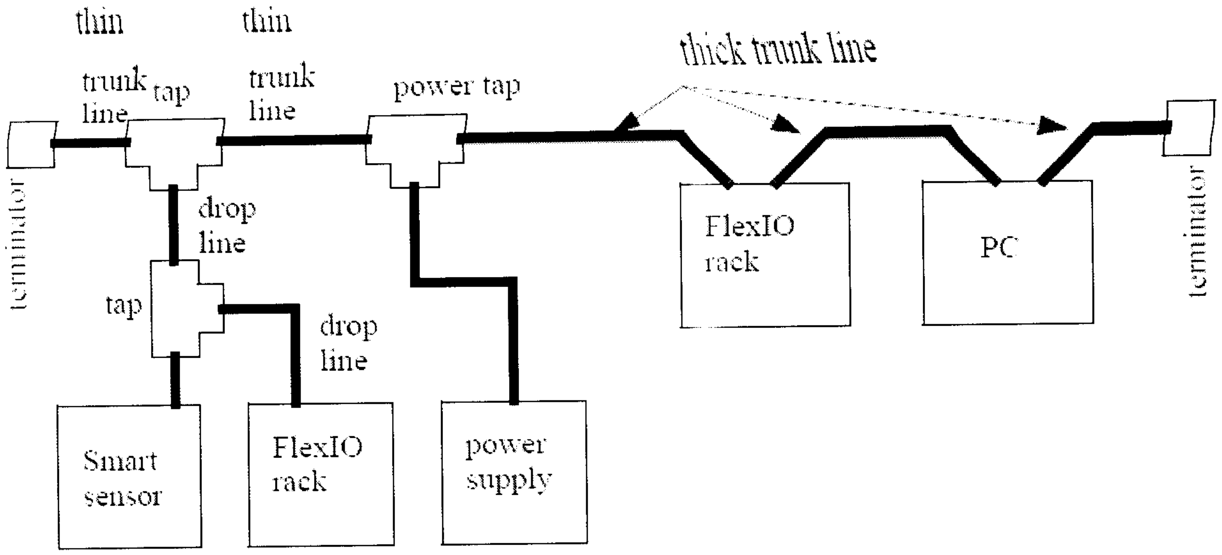
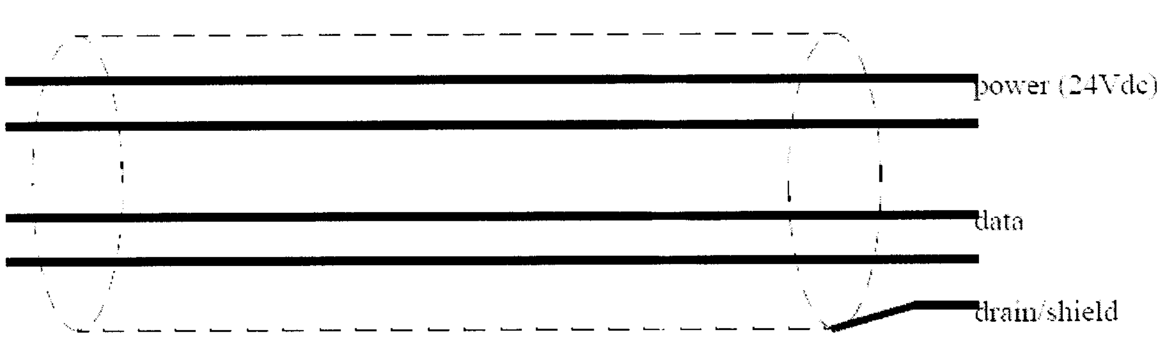


Figure 2.5: Devicenet Network©. [12] ref.

The network cable is important for delivering power and data. Figure 2.6 shows a basic cable with two wires for data and two wires for the power. The cable is also shielded to reduce the effects of electrical noise. The two basic types are thick and thin trunk line. The cables may come with a variety of connections to devices.

- bare wires
- unsealed screw connector
- sealed mini connector
- sealed micro connector
- vampire taps



Thick trunk - carries up to 8A for power up to 500m

Thin trunk - up to 3A for power up to 100m

Figure 2.6: shielded Network Cable ©.

Some of the design issues for this network include;

- Power supplies are directly connected to the network power lines.
- Length to speed is 156m/78m/39m to 125Kbps/250Kbps/500Kbps respectively.
- A single drop is limited to 6m.
- Each node on the network will have its own address between 0 and 63.

2.5.2 CANbus

The CANbus (Controller Area Network bus) standard is part of the Devicenet standard. Integrated circuits are now sold by many of the major vendors (Motorola, Intel, etc.) that support some, or all, of the standard on a single chip.

CANbus covers the first two layers of the OSI model. The network has a bus topology and uses bit wise resolution for collisions on the network (i.e., the lower the network identifier, the higher the priority for sending). A data frame is shown in Figure 2.7. The frame begins with a start bit. This is then followed with a message identifier. For devicenet this is a 5 bit address code (for up to 64 nodes) and a 6 bit command code. The *ready to receive it* bit will be set by the receiving machine. (Note: both the sender and listener share the same wire.) If the receiving machine does not set this bit the remainder of the message is aborted, and the message is resent later. While sending the first few bits, the sender monitors the bits to ensure that the bits send are heard the same way. If the bits do not agree, then another node on the network has tried to write a message at the same time - there was a collision. The two devices then wait a period of time, based on their identifier and then start to resend. The second node will then detect the message, and wait until it is done. The next 6 bits indicate the number of bytes to be sent, from 0 to 8. This is followed by two sets of bits for CRC (Cyclic Redundancy Check) error checking, this is a checksum of earlier bits. The next bit *ACK slot* is set by the receiving node if the data was received correctly. If there was a CRC error this bit would not be set, and the message would be resent. The remaining bits end the transmission. The *end of frame* bits is equivalent to stop bits. There must be a delay of at least 3 bits before the next message begins.

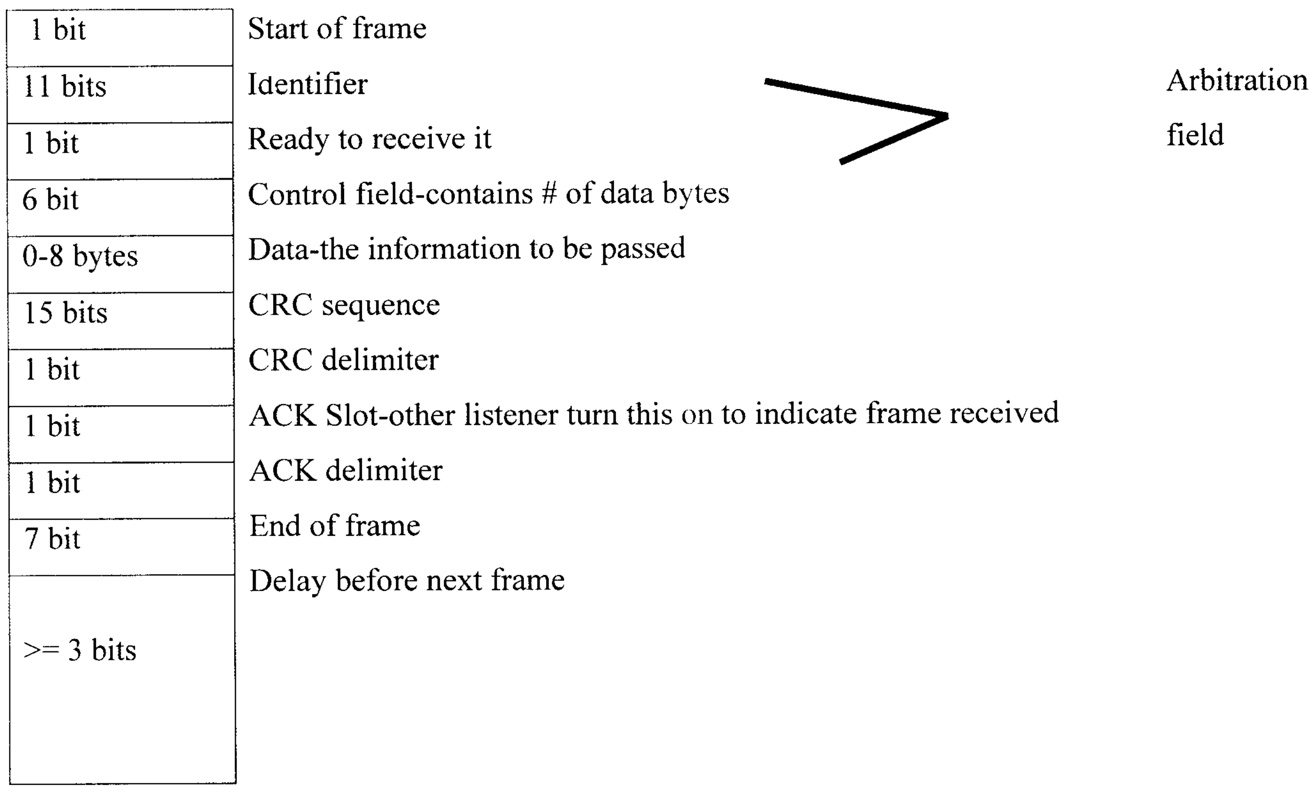


Figure 2.7: CANbus Data frame

Because of the bitwise arbitration, the address with the lowest identifier will get the highest priority, and be able to send messages faster when there is a conflict. As a result the controller is normally put at address *0*. And, lower priority devices are put near the end of the address range.

2.5.3 Controlnet

Controlnet is compliments Devicenet. The standard is designed for communication between controllers and permits more complex messages than Devicenet. It is not suitable for communication with individual sensors and actuators, or with devices off the factory floor. Controlnet is more complicated method than Devicenet. Some of the key features of this network include,

- Multiple controllers and I/O on one network
- Deterministic
- Data rates up to 5Mbps
- Multiple topologies (bus, star, tree)
- Multiple media (coax, fiber, etc.)
- Up to 99 nodes with addresses, up to 48 without a repeater
- Data packets up to 510 bytes
- Unlimited I/O points
- Maximum length examples
    - 1000m with coax at 5Mbps - 2 nodes
    - 250m with coax at 5Mbps - 48 nodes
    - 5000m with coax at 5Mbps with repeaters
    - 3000m with fiber at 5Mbps
    - 30Km with fiber at 5Mbps and repeaters
- 5 repeaters in series, 48 segments in parallel
- Devices powered individually (no network power)
- Devices can be removed while network is active.

This control network is unique because it supports a real-time messaging scheme called Concurrent Time Domain Multiple Access (CTDMA). The network has a scheduled (high priority) and unscheduled (low priority) update. When collisions are detected, the system will wait a time of at least 2ms, for unscheduled messages. But, scheduled messages will be passed sooner, during a special time window.

2.5.4 Ethernet

Ethernet has become a predominant networking format. Most modern Ethernet cards will support different types of frames.

The Ethernet frame is shown in Figure 2.8. The first six bytes are the destination address for the message. If all of the bits in the bytes are set then any computer that receives the message will read it. The first three bytes of the address are specific to the card manufacturer, and the remaining bytes specify the remote address. The address is common for all versions of Ethernet. The source address specifies the message sender. The first three bytes are specific to the card manufacturer. The remaining bytes include the source address. This is also identical in all versions of Ethernet. The *Ethernet type* identifies the frame as a Version II Ethernet packet if the value is greater than 05DChex. The other Ethernet types use these to bytes to indicate the data length. The *data* can be between 46 to 1500 bytes in length. The frame concludes with a *checksum* that will be used to verify that the data has been transmitted correctly. When the end of the transmission is detected, the last four bytes are then used to verify that the frame was received correctly.

6 bytes	Destination address
6 bytes	Source address
2 bytes	Ethernet
46-1500 bytes	Data
4 bytes	Check sum

Figure 2.8: Ethernet Data Frame

2.5.5 Profibus

Another control network that is popular and also available worldwide. General features include;

- A token passing between up to three masters
- Maximum of 126 nodes
- Straight bus topology
- Length from 9600m/9.6Kbps with 7 repeaters to 500m/12Mbps with 4 repeaters
- With fiber optic cable lengths can be over 80Km
- 2 data lines and shield
- Power needed at each station
- Uses RS-485, Ethernet, fiber optics, etc.
- 2048 bits of I/O per network frame

2.6 SERIAL COMMUNICATION

2.6.1 INTRODUCTION

Multiple control systems will be used for complex processes. These control systems may be PLCs, but other controllers include robots, data terminals and computers. For these controllers to work together, they must communicate.

Whether the controlling system is a remote one or local, a method of conveying the data from the data acquisition system to the observable tool has to exist. In this dissertation, the final outcome is to display information on a central terminal (PC). It is therefore important that a method of communication between the PLC(s) and the PC be designed as well as communication between several PLCs in different locations.

The simplest form of communication is a direct connection between two computers. A network will simultaneously connect a large number of computers on a network. Data can be transmitted one bit at a time in series, this is called serial communication. Data bits can also be sent in parallel. The transmission rate will often be limited to some maximum value, from a few bits per second, to billions of bits per second. The communications often have limited distances, from a few feet to thousands of miles/kilometers. Serial communications and networks are both very important in modern control applications. An example of a networked control system is shown in

Figure 2.9. The computer and PLC are connected with an RS-232 (serial data) connection. This connection can only connect two devices. Devicenet is used by the Computer to communicate with various actuators and sensors. Devicenet can support up to 63 actuators and sensors. The PLC inputs and outputs are connected as normal to the process.

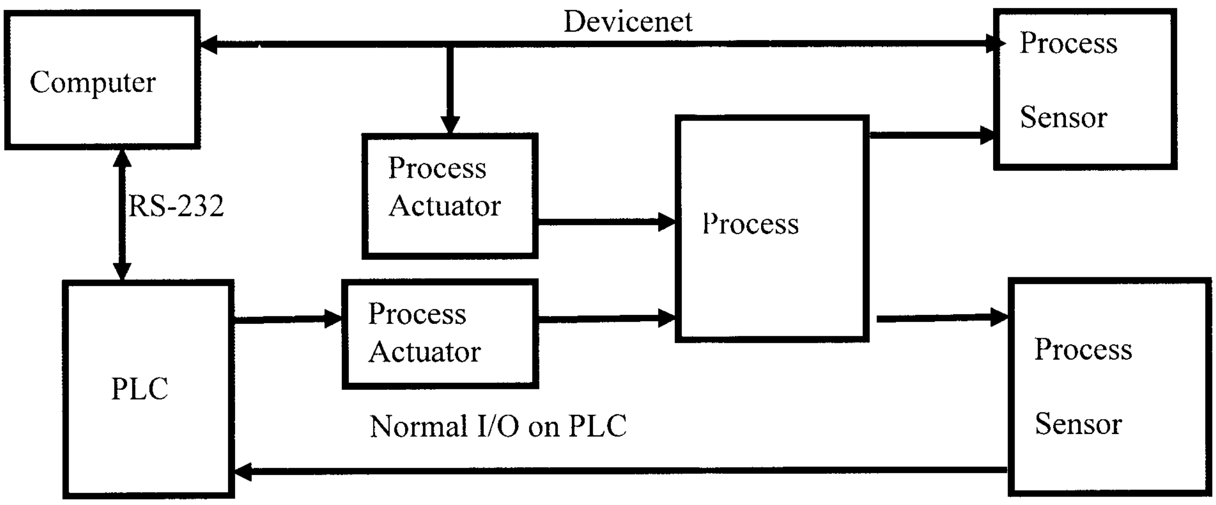


Figure 2.9: Communication System. [ ]

Serial communications send a single bit at a time between computers. This only requires a single communication channel, as opposed to 8 channels to send a byte. With only one channel the costs are lower, but the communication rates are slower. The communication channels are often wire based, but they may also be an optical and radio. Figure2.10 shows some of the standard electrical connections. RS-232c is the most common standard that is based on a voltage change levels. At the sending computer an input will either be true or false. The line driver will convert a false value into a *Txd* voltage between +3V to +15V, true will be between -3V to -15V. A cable connects the *Txd* and *com* on the sending computer to the *Rxd* and *com* inputs on the receiving computer. The receiver converts the positive and negative voltages back to logic voltage levels in the receiving computer. The cable length is limited to 50 feet to reduce the effects of electrical noise. When RS-232 is used on the factory floor, care is required to reduce the effects of electrical noise - careful grounding and shielded cables are often used.

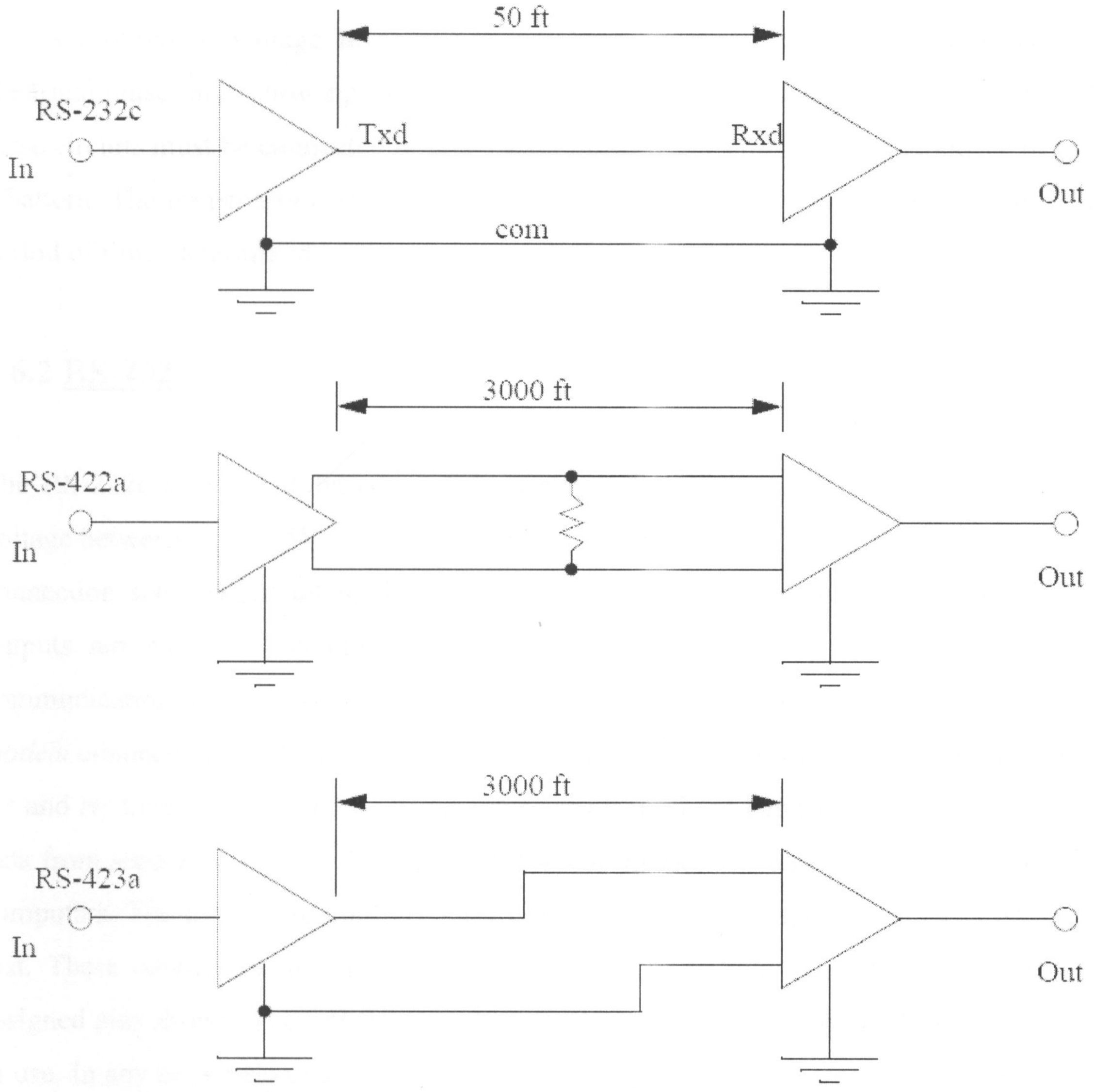


Figure 2.10: Serial Data standard

The RS-422a cable uses a 20mA current loop instead of voltage levels. This makes the systems more immune to electrical noise, so the cable can be up to 3000 feet long. The RS-423a standard uses a differential voltage level across two lines, also making the system more immune to electrical noise, thus allowing longer cables. To provide serial communication in two directions these circuits must be connected in both directions. To transmit data, the sequence of bits follows a pattern. The transmission starts at the left hand side. Each bit will be true or false for a fixed period of time, determined by the transmission speed.

2.6.2 RS-232

The RS-232c standard is based on a low/false voltage between +3 to +15V, and a high/true voltage between -3 to -15V (+/-12V is commonly used). Figure 2.11 shows some of the common connection schemes. In all methods the *txd* and *rxd* lines are crossed so that the sending *txd* outputs are into the listening *rxd* inputs when communicating between computers. When communicating with a communication device (modem), these lines are not crossed. In the *modem* connection the *dsr* and *dtr* lines are used to control the flow of data. In the *computer* the *cts* and *rts* lines are connected. These lines are all used for handshaking, to control the flow of data from sender to receiver. The *null-modem* configuration simplifies the handshaking between computers. The three wire configuration is a crude way to connect to devices and data can be lost. These connectors are either male (with pins) or female (with holes), and often use the assigned pins shown. The DB-9 connector is more common now, but the DB-25 connector is still in use. In any connection the *RXD* and *TXD* pins must be used to transmit and receive data. The *COM* must be connected to give a common voltage reference. All of the remaining pins are used for *handshaking*.

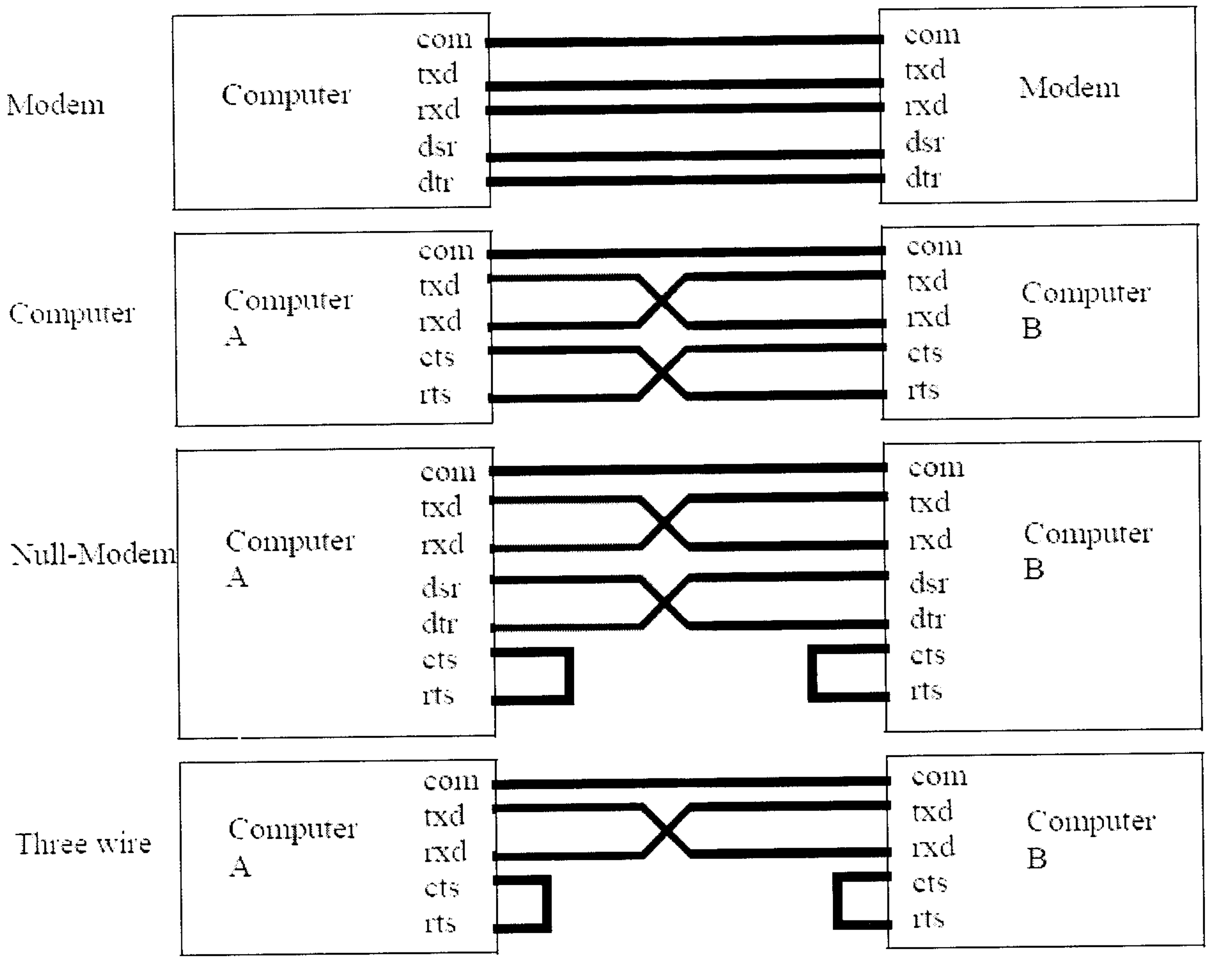


Figure 2.11: Common RS 232 connection schemes

The *handshaking* lines are to be used to detect the status of the sender and receiver, and to regulate the flow of data. It would be unusual for most of these pins to be connected in any one application.

2.7 PROGRAMMABLE LOGIC CONTROLLER SYSTEMS

2.7.1 Programmable Logic Controllers

Control engineering has evolved over time. In the past humans were the main methods of supervising and controlling systems. **Programmable logic controllers**, also called programmable controllers or PLC*s*, are **solid-state** members of the computer family, using integrated circuits instead of electromechanical devices to implement control functions. They are capable of storing instructions, such as sequencing, timing, counting, arithmetic, data manipulation, and communication, to control industrial machines and processes. Figure 2.12 illustrates a conceptual diagram of a PLC application. PLCs can be thought of in simple terms as industrial computers with specially designed architecture in both their central units (the PLC itself) and their interfacing circuitry to field devices (input/output connections to the real world).

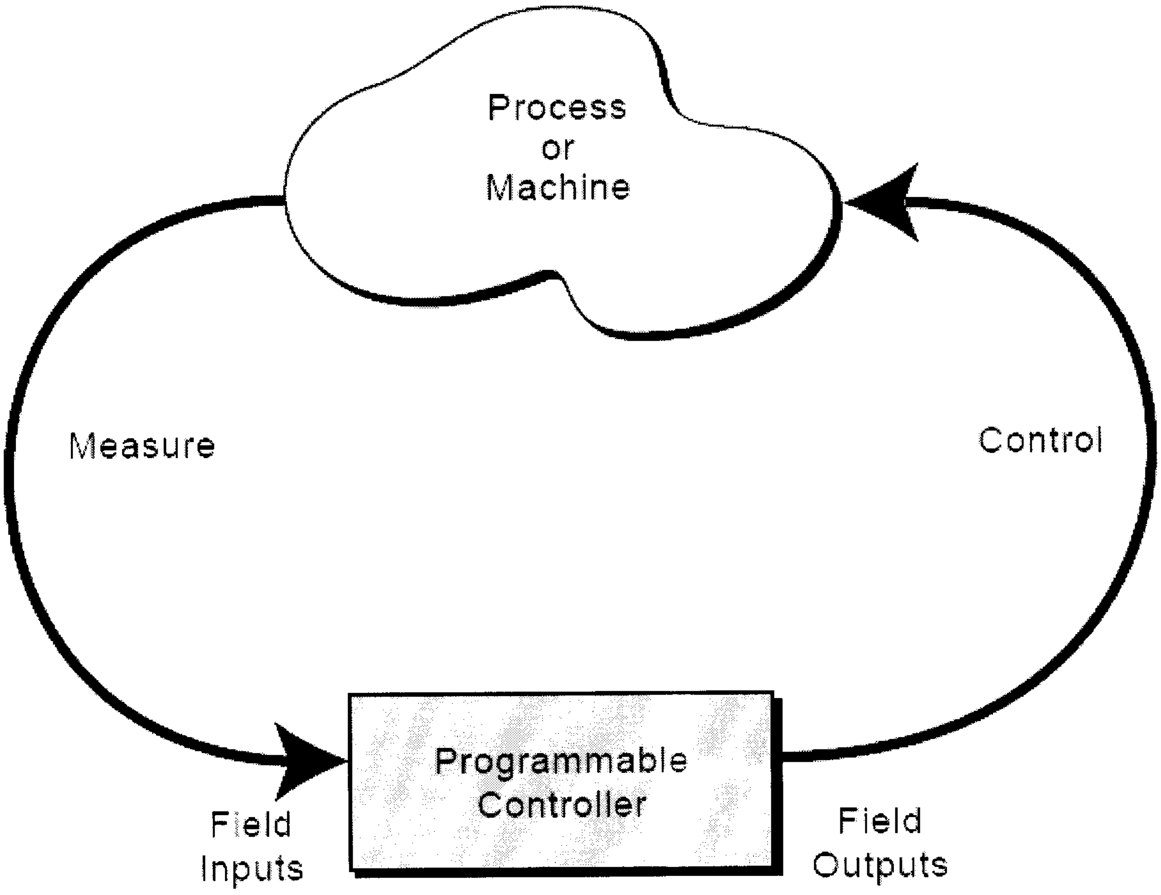


Figure 2.12: PLC Conceptual Application Diagram

More recently electricity has been used for supervisory and control applications. Early electrical control was based on relays. These relays allowed power to be switched ON and OFF without a mechanical switch by commonly making simple logical control decisions. The development of the low cost computer has brought the most recent evolution, the Programmable Logic Controller (PLC). A PLC is a microcontroller used to automate industrial and domestic processes. The use of PLC's begun in the 1970s.

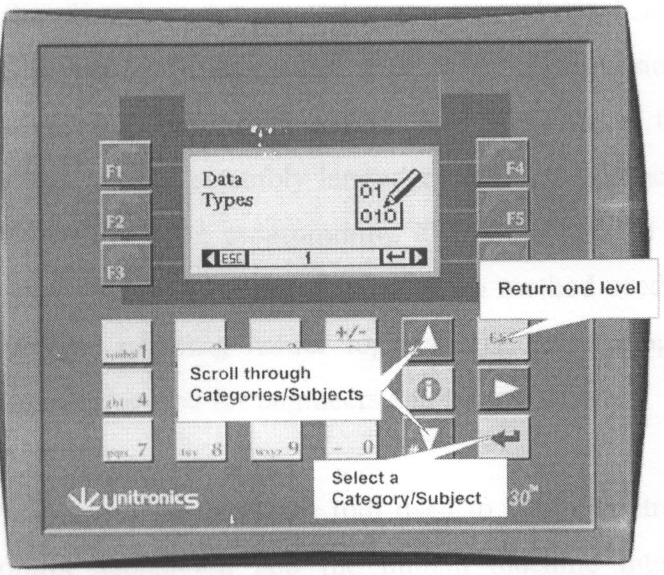


Figure 2.13: A vision controller (Source: Unitronics User Manual)

PLCs were chosen for use in this project due to the following advantages that they offer:

- Cost effective for controlling complex systems.
- Flexible and can be reapplied to control other systems quickly and easily.
- Reliable components make them likely to operate for years before failure.
- The programming is easier since it just involves blocks which are representatives of assembly language.

Unitronics is one of the companies that manufacture several models of PLCs and in this project; one model of their Vision controllers was selected for use.

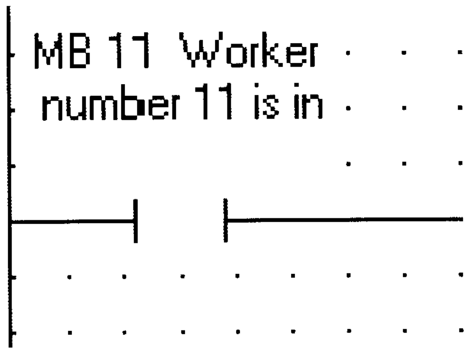
This is the V120 OPLC (Operator Panel Programmable Logic Controller). It is called operator panel because all Vision controllers offer an integrated human machine interface (HMI) operating panel that includes an LCD screen and a keypad. The screen size, type and keypad vary. A figure of a Unitronics OPLC is shown in figure 2.13 above. Besides Unitronics, other PLC manufacturers are Siemens.

2.7.2 Unitronics OPLC

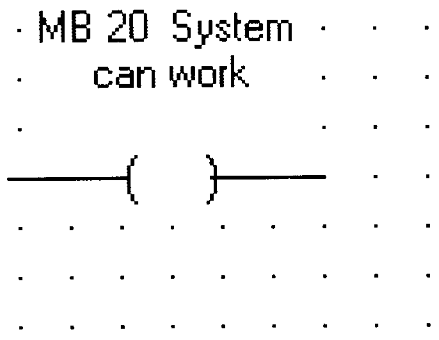
After planning the control task, control applications have to be written, debugged and downloaded into the controller i.e. the PLC application is the automation application. There are several methods for programming PLCs. One of the earliest techniques involved mnemonic instructions i.e. assembly language. Sequential Function Charts (SFCs) have been developed to accommodate the programming of more advanced systems. These are similar to flow charts. Ladder logic is the main programming method used for PLCs. Ladder logic mimics relay logic and thus selecting ladder logic as the main programming method reduces the amount of retraining needed for engineers and trades people.

VisiLogic is the software tool used to create control projects for Vision controllers. Both the control application and the human machine interface (HMI) are developed in the same programming environment thus increased time efficiency. The PLC application is written using the ladder editor. The ladder diagrams are composed of contacts, coils and function block elements. The ladder diagram contains a left and right rail. Between these rails, the control application is arranged in nets. A net contains a row of ladder elements that drive a coil. Power flows through the ladder elements in a net from left to right. Each net must contain only one rung. This is why the first ladder element in the net must touch the left ladder rail [4]. All the elements in a net must be connected to allow power flow and the last element on the right does need to be connected to the right side of the ladder in each net.

The contact in a ladder program represents the input conditions. Power is allowed to flow from the left rail o the right through the contacts. Output instructions or results or expressions of an instruction are represented by coils. Diagrams of the contact and coil representations in VisiLogic are shown in figure 2.14 below. Once an element is placed in a net, the ***select operand and address box*** opens and allows the selection of an operand. An element's operand is the form in which information is stored and operated on in the ladder program. The ladder elements and functions are linked to operands. Operands contain data and the ladder program determines the way in which the operand data is used in a program.



(a) representation of a contact in VisiLogic



(b) representation of a coil in VisiLogic

Figure 2.14: Contact and coil (Source: Unitronics User Manual)

2.7.3 Operand types

There are different kinds of operand types and some of the commonly used are described below:

(i) Inputs (I) – inputs are bit operands (1 or 0). The number of the inputs varies according to snap-in I/O modules and I/O expansion modules integrated into the system. An input is an actual hardwired input connection into the controller. A V120-22-R1 OPLC has a total of ten (10) digital inputs and one (1) analogue input. All the ten inputs can be set to pnp (source) or npn (sink) via a single jumper and appropriate wiring. An input is an actual hardwired input connection into the controller.

(ii) Outputs (O) – outputs are bit operands (1 or 0). The number of outputs varies according to snap-in I/O modules and I/O expansion modules integrated into the system.

(iii) Timers (T) – there are three types of timers each with three variables; timer bit value, timer preset value and timer current value.

- Timer Bit Value: A timer is scanned as a bit data type (scan for OFF, scan for ON). The result of the scan is dependent on the timer type.
- Timer Preset Value. A running timer always decrements (counts down) from the Preset Value. The Preset Values are loaded for all timers at power up. The Preset Value is also loaded into the Current Value when the timer is reset.
- Timer Current Value. The current value of the timer is dependent on the timer type.

All timer types are activated by a rising transition edge, OFF to ON. Timers can further be divided into three types and these are:

(a) **TD Timer: On Delay** - When the timer's Start & Run Condition is OFF, the timer's Bit Value is also OFF. When the timer's Start & Run

Condition rises, the timer's Preset Value is loaded into the timer's Current Value. The timer begins to run. Note that the timer's Bit Value is OFF. If the timer's Start & Run Condition remains ON during subsequent PLC cycles, the Current Value of the timer continues to decrement. When the timer has decremented to 0, and the timer's Start & Run Condition is still ON, the timer's Bit Value turns ON. Note that when the timer has finished running, its Current Value is 0. If the timer's Start & Run Condition falls while the timer is decrementing, the timer stops running. The current value of the timer remains. Timer Reset takes precedence over the timer's Start & Run Condition. When the timer' Reset Condition rises, the timer's Bit Value turns OFF. The timer's Preset Value is loaded into the Current Value, and the timer's Start & Run Condition cannot activate the timer as long as Reset is ON. When the timer's Reset Condition falls while the timer's Start & Run Condition is ON, the timer begins to run, exactly the same as when the timer's Start & Run Condition rises.

(b) **TA Timer: Accumulated** - When the timer's Run Enable Condition rises, the timer's Preset Value is loaded into the timer's Current Value. The timer begins to run. Note that the timer's Bit Value is OFF. When the timer's Run Enable Condition remains ON during subsequent PLC cycles, the Current Value of the timer continues to decrement. When the timer has decremented to 0, and the timer's Start & Run Condition is still ON, the timer's Bit Value turns ON. Note that when the timer has finished running, its Current Value is 0. If the timer's Run Enable Condition falls while the timer is running, the timer stops running, but the current value of the timer is retained. When the timer is reactivated, it begins decrementing from the retained value. Timer Reset takes precedence over the timer's Run Enable Condition. When the timer' Reset Condition rises, the timer's Bit Value turns OFF. The timer's

Preset Value is loaded into the Current Value, and the timer's Run Enable Condition cannot activate the timer as long as Reset is ON. When the timer's Reset Condition falls while the timer's Start & Run Condition is ON, the timer begins to run, exactly the same as when the timer's Run Enable Condition rises.

(c) **TE Timer: Extended Pulse** - When the timer's Start Condition rises, and the Bit Value is OFF, the timer's Preset Value is loaded into the timer's Current Value. The timer begins to run and the Bit Value turns ON. If the timer's Start Condition remains ON during subsequent PLC cycles, the Current Value of the timer continues to decrement. However, if the timer's Start Condition rises before the timer has decremented to its Preset Value, the timer reloads the Preset Value into the Current Value, and again begins to decrement. Note that a falling Start condition does not affect the timer. When the timer has decremented to 0 the timer's Bit Value turns OFF. Note that when the timer has finished running, its Current Value is 0. Timer Reset takes precedence over the timer's Start Condition. When the timer' Reset Condition rises, the timer's Bit Value turns OFF. The timer's Preset Value is loaded into the Current Value, and the timer's Start Condition cannot activate the timer as long as Reset is ON. When the timer's Reset Condition falls while the timer's Start Condition is ON, the timer stops. When the Start condition rises, the timer begins to run, counting down from the Preset Value, exactly the same as when the timer's Start Condition rises

(iv) Counters (C) – these are ladder operands that count rising edge pulses (transition from logic zero to logic one). VisiLogic offers 24 built-in counters, represented by the symbol C. To use an Up Counter in a program, an Increment function is placed in a net and then C is selected in the *select operand* box. To use a Down Counter in a program, a Decrement function is used. A counter's Preset Value can be assigned either in the Select Operand box or in the Output Window.

(v) Memory Bit (MB) – are bit operands (0 or 1). There are 4096 memory bits, i.e. addresses MB0 to MB4095.

(vi) Memory Integers (MI) – are 16 bit integer operands that may be signed or unsigned. The range is -32768 to 32768. There are 1024 memory integers i.e. addresses MI 0 to MI 1023.

(vii) Constant value – is an integer number signed or unsigned and is created by the programmer. It is symbolized by the pound sign (#).

Some operands, called system operands are connected to certain functions and to values in the controller's operating system. System Operands are used by the controller's operating system to manage certain functions and values. Many System Operands are linked to fixed parameters and are read-only, such as SB 2 Power-up bit, which turns ON for a single cycle whenever the controller powers up.

Other System Operands can be written to by the program, or via INFO Mode. System Operands have preset descriptions that describe their function.

These include some of:

i) System Bit (SB) – bit operand (0 or 1).

ii) System Integers (SI) – these are 16-bit integer operands signed or unsigned from -32768 to 32768.

iii) System Double Word (SDW) – these are 32-bit unsigned integer operands with maximum = 4 294 967296.

iv) System Long – these are 32-bit integer operands, signed or unsigned.

If the controller is networked, the following operands are accessible to other controllers:

- Network System Bit (NSB) – SB200 to SB207
- Network Input (NI) – bit operands I0 to I16
- Network System Integer (NSI) – these are 16-bit operands i.e. SI200 to SI201.

2.7.4 <u>Ladder elements and functions</u>.

Ladder elements are placed in a net and are linked to operands. Besides the contacts and coils, some of the commonly used elements and functions are compare, math, clock, store, vector, linearization, logic etc.

2.7.4.1 Contacts

A contact represents an action or condition. You can link it to any of the above stated operands and can take on various forms. The most commonly used forms of contacts are:

a. Direct contact – A door buzzer is an example of a Direct Contact. When you push the buzzer, power flows through the circuit and the buzzer sounds. When you release the buzzer, the sound stops.

   During the system scan, the processor evaluates the program elements net by net.

   If the Direct Contact bit operand (the door buzzer) is OFF (logic 0): power will not flow through the Direct Contact. The door buzzer is silent.

   If the Direct Contact address (the door buzzer) is ON (logic 1): power will flow through the Direct Contact. The door buzzer sounds.

b. Inverted contact – is normally closed (NC) and opens when the contact bit operand is OFF. An Inverted Contact condition can be from an external input device (for example: a push button) or from an internal input system element (for example: SB 50 Key +/- is pressed).

   An emergency light contains an example of an Inverted Contact.

   Normally, there is power flow through the emergency light's Inverted Coil and the light stays off.

During an electric power outage, the power flow through the Inverted Coil stops and the emergency light comes on.

During the system scan, the processor evaluates the program elements net by net.

If the Inverted Contact address (power supply) is ON (logic 1): power **will not** flow through the Inverted Contact. The emergency light will stay off.

If the Inverted Contact address (power supply) is OFF (logic 0): power **will** flow through the Inverted Contact. The emergency light turns on.

If the power outage ends and power flow is returned to the Inverted Contact, it will close and the emergency light will again turn off.

c. Positive transition (Rise) contact – gives a single one shot pulse when its reference address rises from logic 0 to logic1. A cellular phone keypad key is an example of a Positive Transition Contact. When you push a key a number is displayed on the screen. It does not matter if you push the key quickly or hold it down for several seconds. The number will only appear once on the screen.

   The cellular phone registers the transition from key NOT pressed to key pressed. The length of time the key is pressed is not relevant. You must release the key and press it again to repeat the number on the cellular phone screen.

   During the system scan, a Positive Transition Contact address is evaluated for a transition from OFF to ON. A transition allows power to flow through the Positive Transition Contact for one scan.

   At the end of a scan, the Positive Transition Contact is reset to ON (logic 1). The Positive Transition Contact is re-activated when the linked signal turns from OFF to ON.

d. Negative transition contact – gives a single one shot pulse when the bit operand it is linked to falls from logic 1 to logic 0. A computer ON/OFF button is an example of a Negative Transition Contact. The computer is ON.

   If you push the ON/OFF button in without releasing it, the computer will not shut down. But when you release the button, the system registers a change in status from ON to OFF. The computer then shuts down.

   During the system scan, a Negative Transition Contact address is evaluated for a transition from ON to OFF. A transition allows power to flow through the Negative Transition Contact for one scan.

   At the end of a scan, the Negative Transition Contact is reset to OFF (logic 0). The Negative Transition Contact can only be re-activated when the triggering signal again changes from ON to Off.

2.7.4.2 Coils

A Coil represents a result or expression of an action. A coil turns ON when the preceding net conditions are ON, allowing power flow to reach the coil from the net. If the preceding net conditions are OFF, a coil turns OFF. A coil can be linked to any operand a VisiLogic coil types include:

a. Direct Coil - A Direct Coil turns ON when the preceding net conditions are ON, allowing power flow to reach the coil from the net. If the preceding net conditions are OFF, a direct coil turns OFF. The coil can represent an external output device (for example: alarm bell) or to an internal system element, as for example, SB 41, which is key #1 on the controller's keyboard.
b. Inverted Coil - An Inverted Coil turns OFF when the preceding net conditions are ON, allowing power flow to reach the coil from the net. If the preceding net conditions are OFF, an inverted coil turns ON.
c. Set Coil - A set coil separates the coil from the action or condition that energized the coil. Once energized, a set coil's result is no longer dependant on the action that energized it.

A set coil stays energized (latched) until its condition is reset (unlatched) by a reset coil. An example of a set coil is an overhead light. When you turn on a light, it stays lit until you turn it off (reset or unlatch it) or the light bulb burns out. You do not have to hold the light switch to keep the light on.

An example of a coil that is **not** set (unlatched) is a car horn. It is expected to toot only when the horn button is pressed and is expect to stop when the horn button is released. A set coil is never used without a condition to reset it.

d. Reset Coil - A reset coil turns a set coil OFF (unlatches), when the preceding net conditions are ON, allowing power flow to reach the reset coil from the net.
e. Toggle coil - A toggle coil changes its state when it is activated. An example of a toggled coil is a light switch. When you turn on a light, it stays lit until you toggle it; it then turns off. The light stays off until you toggle it back on.

It is important to note that a coil should never be energized more than once in a program.

2.7.4.3 Compare Functions

These compare the values (usually two) according to the selected type of function. If the comparison is true (logic 1), power then flows through the block. If the comparison is false (logic 0), then power does not flow through the block. There are six compare functions and these are namely greater than (>), less than(<), greater than or equal to (≥), less than or equal to (≤), equal to (=) and not equal to (≠). Compare functions can compare memory integers (MI), memory long integer (ML), double word (DW), system operands like system integers (SI), system long (SL), system double word (SDW), network system integers (NSI) and constant value (#).

2.7.4.4 Math Functions

Mathematical functions are performed by placing math functions in a net. These functions are provided for:

a. Increment/decrement functions – these elements increment or decrement the value of the selected operand by one.

b. Add function – this function adds up to eight input values of operands that include MI, ML, DW, SI, SL, SDW, NSI and constant value. With the exception of the constant value, all the other operands may be used to contain the output value.

c. Subtract function - The math function Subtract is executed by the Subtract function block. Just like the add function, the input values to the subtract function block may be MI, ML, DW, SI, SL, SDW, NSI or a constant value.

d. Multiply function - The math function multiply is executed by the multiply function block. It also works on input values of operands like those of add and subtract functions.

e. Divide function – this performs the math function Divide and still with the same operand inputs like add and subtract functions.

f. Linearization function - The Linearization functions, located on the Math menu, facilitates the conversion of values. It is used, for example, to convert analog input values to values in degrees Celsius. There are two variations in the types of linearization *functions. One of the types linearizes a single source value, and then stores it in the target* register. The other function linearizes a vector.

g. Floating point function - Float function blocks enable you to use Memory Float (MF) values in your program. Floating point values include sign. These elements basically perform mathematical operations of addition. subtraction, compare etc on float point numbers.

2.7.5 The Human Machine Interface.

All Vision controllers offer an integrated human machine interface (HMI) operating panel that includes an LCD screen and a keypad. The screen size, type and keypad vary. In order to have interaction with the PLC, an interactive user interface needs to be created. This requires the use of display screens, variables, list variables and display jumps. This can be used by the operators of the controller to set certain parameters about the controlled variable by using the V120 keypad. The operators do this in local mode i.e. when the operators are closer to the PLC.

To change between Displays, jumps from one display to the next should be set. A jump contains a Jump condition, which is linked to a bit operand, and a destination Display. You can also load a Display by placing a function in a Ladder net. When an HMI keypad entry variable is active, and the Enter key is pressed on the controller keypad, SB 30 (*HMI Keypad Entries Completed)* turns ON. This can be used as a Jump condition. The HMI also has graphic images in display and simple geometric shapes can be drawn on a display. Graphic displays can therefore be imported from the Image Library, drawn on a display or created with a program such as Microsoft Paint and then imported. The two kinds of images found on HMI are:

(a) Fixed graphic images this type of image stays on the screen and does not change until a different Display is loaded by the program.

(b) Variable graphic images Variable images change according to the value of a linked operand. Binary Image Variables are linked to bit operand status (MB, SB, I, T, O). List Image Variables are linked to integers (MI, SI, ML, and SL).

It is important to note however that:

(i) Although an imported image can be resized, resizing may result in some degree of distortion. To avoid this, images that are created to match the required size are used.

(ii) The HMI display uses a grid which separates the lines eight (8) pixels apart. To optimize displays and shorten the PLC cycle time, images and variables should be aligned to grid.

2.7.6 Program Sequencing: Modules, Subroutines & Jumps

A module is a container of subroutines. Modules and subroutines are used to divide an application into program blocks. You can then run these program blocks conditionally, from any point in your control application. Within the program tree, elements are presented alphabetically. This does not affect the order in which the program runs.

Ladder Modules and subroutines can be moved via drag-and-drop, as can HMI Modules and Displays. Again, moving elements does not affect the order in which they run.
The Main Ladder Module, Main Subroutine, Start-up HMI Module and the Start-up HMI Display cannot be moved via drag-and-drop or erased. For easy identification, they are always marked in orange.
There are two main functions used to invoke subroutines and HMI displays these are:

(i) Call Subroutine - This function causes a subroutine to run in response to a Ladder Condition. To control the Ladder program flow sequence and avoid loops, use the Call Subroutine function to conditionally call subroutines. Within a subroutine, you control the sequence by conditionally skipping over nets using Labels and Jump to Label functions. This enables you to shorten the program scan time. A new VisiLogic project contains the main module and subroutine for the program. Each new subroutine contains a default number of nets and a Subroutine Return function. Subroutines do not run if they are not called by Call Subroutine. If no Call Subroutine commands are included in the first subroutine of the main module, the program runs until it reaches the Subroutine Return function, and then jumps back to the beginning of the first subroutine.

(ii) Call HMI Display - These Ladder functions call HMI Displays. Use these functions to initially load the Display, and then to refresh it when your application requires, as, for example, when you want to update variable display. They are located on the Ladder toolbar, under the HMI menu.

2.7.7 Hardware Configuration

Hardware configuration is the adjustment of certain hardware parameters through programming or through actual equipment rearrangement. In Vision controllers, hardware configuration is important because this is when power requirements are selected i.e. either 12V or 24V and electrical current consumptions. This selection can be done through a single jumper. Hardware configuration also allows the setting of input and output requirements. The digital inputs of Vision controllers can either be set to pnp (source) or npn (sink) through a single jumper and appropriate wiring. In pnp digital inputs, the positive terminal (12V or 24V) of the power supply is the one that is connected to all the digital input ports and the negative to the zero volts input. In npn digital input configuration, the positive terminal of the power supply is taken to the zero volts port and the negative terminal of the power supply is connected to all the digital inputs. Through hardware configuration, the type of analogue input is selected to be either a voltage input or current input. All these hardware configurations are done through several I/O jumper settings within the controller and therefore the controller has to be opened in order to access the jumpers.

CHAPTER 3

SYSTEM DESIGN

3.1 Planning the work

The control and networking system is specified such that the robot should be able to load parts in the box until it reaches a prescribed weight, which is the main parameter to be controlled. This weight is then converted to an electrical signal. This electrical signal then becomes the input to the PLC and is processed by the PLC application to produce an output. This output should then be relayed to centrally location and displayed on the screen. The description above therefore entails the need for:

- Input section – data acquisition and input to the PLC.
- Control section – the PLC application program.
- Output section – communications and displays.

A block diagram of these blocks interconnected is shown in figure 3.1 below.

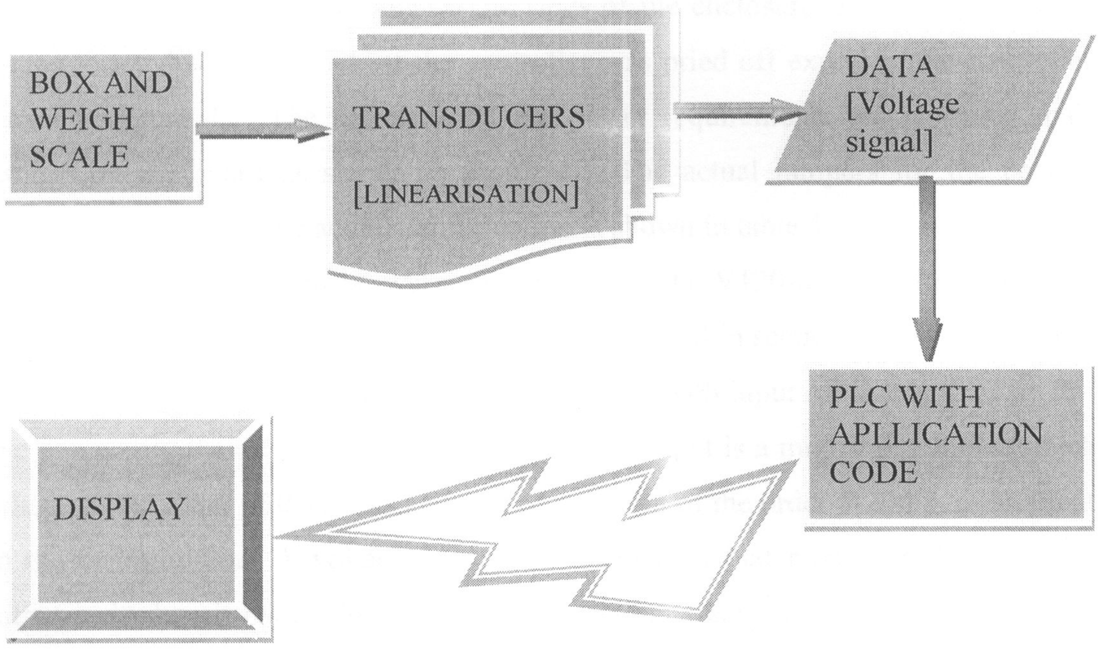


Figure 3.1: System layout

The input section includes the signals from the digital weigh scale and how they are conditioned before they enter the PLC. It can also on a broad scale cover the power which biases the PLC.
The control section is the actual software application that is designed in ladder and HMI. It interprets the signals from the inputs and causes the certain actions at the output section. The development of the ladder application and the HMI is done in a same programming environment.
The output section covers both the control outputs i.e. the relay output as well as the displays and the signals which can be in form of LEDs. These various sections were designed and configured as stated in the subsequent sections for the project to be successfully complete.

3.2 Hardware Architecture and Configuration

The first consideration of all was the power requirements for the PLC. The V120-22-R1 OPLC has a power supply requirement of 12VDC or 24VDC with a permissible range from 10.2VDC to 28.8VDC with less than 10% ripple. A 12VDC power supply was chosen with a consideration that it would be easy to attenuate if a higher DC voltage is available instead. The PLC was first opened by locating the four slots on the sides of the enclosure. And using the blade of a flat-bladed screw driver, the back of the controller was pried off exposing the controller's board as shown in figure 4.2. The selection for the power requirements was done by making jumper settings on the board as shown in figure 4.3. The actual jumpers for the power supply are jumpers 5 and 6 and their actual configuration is shown in table 4.
The next configurations to be were on the inputs. The V120-22-R1 OPLC has ten (10) digital inputs that can be configured as pnp or npn as explained in section 3.2.6 and one analogue input. The input impedance for these digital inputs is 3kΩ with input cable length of up to 100 meters. The pnp configuration was selected. The analogue input is a multi-range input which can take a voltage of the order of 0-10V as an input or a current of the order 0-20mA as an input depending on the configuration. A voltage input was selected in that most transducers have voltage as output and when it comes to connectivity, a current transducer requires that the voltage be in series with the PLC input unlike a voltage transducer which can just be applied across the PLC input and thus easy to manipulate. The settings for the jumpers in order to come up with the above mentioned configurations are as shown in table 5 and the jumpers for the digital and analogue inputs are jumpers 1 and 3 respectively.

The digital outputs did not require any configuration and were used directly. Next to configure were the communication ports. The V120-22-R1 OPLC has two communication ports and both the ports can be configured either as RS232 or as RS485. The ports were configured to RS232 and the panel was closed again. The next in the design was the control application and is explained in the next section.

3.3 <u>Ladder and HMI design</u>

A well known concept from system design is that for a system to work effectively each subsystem must perform its functions well and its goal must not conflict with the goals of the other subsystems. With this project not being an exception, thorough study of the three main subsystems (data input, control application and communication & display) was done in order to finally come up with a working software code.
The application for this project has got only the Main modules and six (6) HMI displays. This is main module explained below in the way that the system will be operating. The HMI displays are shown in appendix in the appendices and are described below.

3.3.1 Ladder

The ladder as mentioned above has only the Main module and it is all explained below. The explanation just follows them in order of appearance in the program and not order of execution.

(a) Part 1– this is named analogue weight. It starts with the linearization function. This function converts values from the analogue input to memory integers (MI). The desired prescribed weight of the box to be 250 grams and this is linearized to optima voltage signal of 5V.This is store in the MI 0 and MI 1. These memory integers are referred to later in the report. Since also the weight of the parts to be packed in the box must be of a specific weight, the same linearization function is to be used only that this time one part has a weight of 50grams and is linearized to 1V and these are stored in the MI2 and MI3. Using these memory integers are now used to develop the remaining code.

(b) Part 2 – this is named Comparison stage. It contains compare functions and coils. With the assumption that the weight of the box when it is empty is negligible, if the memory integer, MI1 is 'A' and the integer # 250grammes is 'B', when A < B, it will simply mean that the signal that was being received from the weigh scale is less than 250grammes.This will imply that the parts must be fed and the box will not move from the filling position until 'A = B' and this is when packing will stop. Thus on the display of HMI it will show the screen reading "Green Light On". It should be understood that this display could be the LED in the practical situation.

When the MI1 which is A, integer # 250grammes is B are compared. If it is found that A > B, on the HMI display it will be show the screen reading" Red Light On". This will imply that there is an error in the packing because the weight of the box is greater 250grammes which is the prescribed weight. Theses will be the boxes that will be removed from the system.

In the same line with the same assumption for A and B, when the two values are equal, on the HMI display, it will show the screen reading "Orange Light On". This will imply that there is normal packaging and the box has reached a prescribed weight. It should again be noted that all these three displays can be in form of LEDs in practical sense.

On the other hand, for the part to be fed, if we assume the MI3 is 'A' and the integer #50 is 'B'. When these are compared and "A > = B", it will imply that there is a part waiting in the Part Pickup fixture. On the HMI display it will show a screen reading "Part waiting." This could be a LED as well.

Part 3 - Execution stage. When MI3 >= #50grammes and MI1 < 250grammes it will simply mean that the PLC is ready to execute the 'feed part' function and it will send this instruction through the devicenet network to the robot to pick up the part from the Part Pickup fixture because it is available and the box is on the weigh scale and it has not yet reached the prescribed weight. This is explained by the conditions above.

On the other hand when MI1 < 250grammes but with MI3 >= #50grammes it will simply mean that the PLC is ready to execute the instruction 'Stop feeding Part' and it will send this instruction through the Devicenet network to the robot to stop feeding the parts to the box since the condition above tells that the box has reached the prescribed weight. The other condition explains that there could be parts ready for pick up in the part pickup fixture but these will not be loaded into the box.

The design and operation of the system has been explained in parts but the overall understanding can only be gotten if the code is executed.

3.3.2 Assumptions

The following assumptions were used during the planning process:

i. Since no information was obtained stating the characteristics of the box, i.e. actual shape box was assumed to be linear. This is to mean that the amount of parts in the box varied linearly with amount that was loaded in.

ii. This system assumes that only one parameter get into abnormal state frequently but at least one at a time..

iii. The weight of the box when it is empty was also assumed to be negligible.

iv. The transducers where not to be designed but obtained in their finished form.

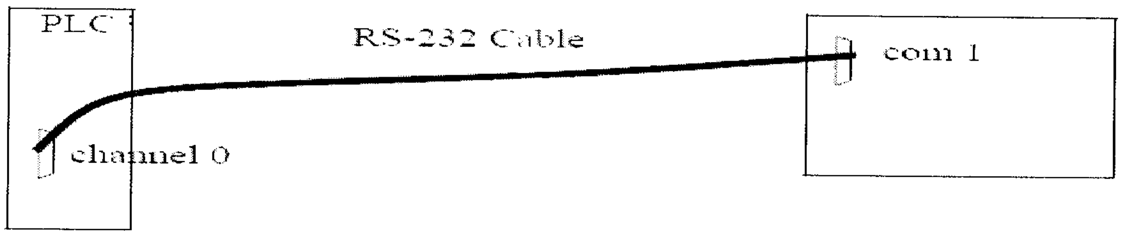
v. Only the outputs were to be examined for compatibility with PLC inputs.

3.4. Devicenet networking

On the networking part, when a PLC is ready to receive data it will set the *CTS* bit, the remote machine which is the robot in this case will notice this on the *RTS* pin. The *DSR* pin is similar in that it indicates the modem is ready to transmit data. *XON* and *XOFF* characters are used for a software only flow control scheme.

Many PLC processors have an RS-232 port that is normally used for programming the PLC. In this project, this same port is to be used for implementations. Figure 3.2 shows an example of how the PLC can be connected to a robot with a Null-Modem line. It is connected to the *channel 0* serial connector on the PLC processor, and to the *com 1* port on the robot. In this example the *terminal* could be a robot packing parts in the box program. The ladder logic below will send a signal to the robot to pick up a part from the part pickup fixture and load it in the box when all conditions are met. It should be noted that;

1. Serial communications pass data one bit at a time.
2. RS-232 communications use voltage levels for short distances.



(a): RS 232 connection between the PLC and the Robot

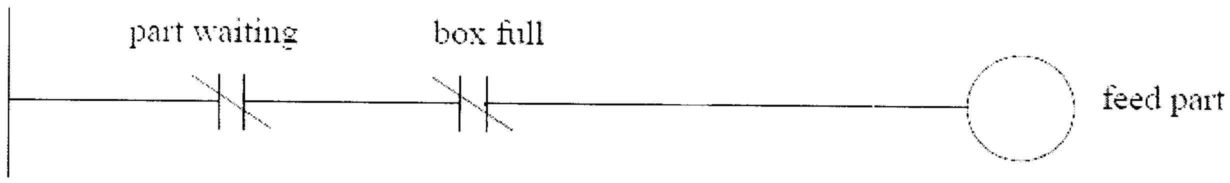


Figure 3.2(b): Part of the ladder logic

Thus using the RS 232, the PLC will be able to communicate with the robot. And the robot is instructed when to pick up a part from the pickup fixture and load it in the box. Through the same network it is instructed when to stop loading.

3.4.1 Control Network Problems

A wide variety of networks are commercially available, and each has particular strengths and weaknesses. The differences arise from their basic designs. One simple issue is the use of the network to deliver power to the nodes. Some control networks will also supply enough power to drive some sensors and simple devices. This can eliminate separate power supplies, but it can reduce the data transmission rates on the network. The use of network taps or tees to connect to the network cable is also important. Some taps or tees are simple *passive* electrical connections, but others involve sophisticated *active* tees that are more costly, but allow longer networks.

The transmission type determines the communication speed and noise immunity. The simplest transmission method is baseband, where voltages are switched off and on to signal bit states. This method is subject to noise, and must operate at lower speeds. RS-232 is an example of baseband transmission. Carrier band transmission uses FSK (Frequency Shift Keying) that will switch a signal between two frequencies to indicate a true or false bit. This technique is very similar to FM (Frequency Modulation) radio where the frequency of the audio wave is transmitted by changing the frequency of a carrier frequency about 100MHz. This method allows higher transmission speeds, with reduced noise effects. Broadband networks transmit data over more than one channel by using multiple carrier frequencies on the same wire. This is similar to sending many cable television channels over the same wire. These networks can achieve very large transmission speeds, and can also be used to guarantee real time network access. The bus network topology only uses a single transmission wire for all nodes. If all of the nodes decide to send messages simultaneously, the messages would be corrupted (a collision occurs).

There are a variety of methods for dealing with network collisions, and arbitration.

I. CSMA/CD (Collision Sense Multiple Access/Collision Detection) - if two nodes start talking and detect a collision then they will stop, wait a random time, and then start again.

II. CSMA/BA (Collision Sense Multiple Access/Bitwise Arbitration) - if two nodes start talking at the same time the will stop and use their node addresses to determine which one goes first.

III. Master-Slave - one device one the network is the master and is the only one that may start communication. Slave devices will only respond to requests from the master.

IV. Token Passing - A token, or permission to talk, is passed sequentially around a network so that only one station may talk at a time. The token passing method is deterministic, but it may require that a node with an urgent message wait to receive the token. The master-slave method will put a single machine in charge of sending and receiving. This can be restrictive if multiple controllers are to exist on the same network. The CSMA/CD and CSMA/BA methods will both allow nodes to talk when needed. But, as the number of collisions increase the network performance degrades quickly.

CHAPTER 4

DISCUSSION

4.1 Data Analysis

In this project, the study or data collection was divided into four. These four areas were collecting data from the related companies and see the methods of packing system, studying the vision controller i.e. both hardware and software development, comparing with existing systems in both costs and reliability.

The project approach began with understanding beverage and packaging reticulation system. In the project description, a statement of the parameters to be controlled was given from a general perception. Therefore in order to scale the scope of controlling to one that seriously affects the utilities, it was necessary to visit Zambia breweries and Zambia bottlers for information. From the visits, it was found that the main effects on the network utilities have got a lot of failures and errors among them are; they are using timers and these are not a accurate system because it relays so much on the efficiency of the motor and incase of any failure they are a lot of losses.

The system also considers less on the weight, thus less consideration on the amount of the product packed. Therefore this parameter was selected for the project and set for simulation and the networking part is explained very well. Therefore the system could only be simulated but still examine the feasibility of such a packaging system.

The second task was a study of the Vision Controllers and a specific study of the V120-22-R1 was taken. This study included a review of the codes that came with the controller. Both the HMI and the ladder were studied. Though the V120 controller has the same versatility as the other vision controllers, the number of keys or keypad functions it offers are far much less than the other controllers and as such introduces a limitation on the functionality. The V120 display is relatively small and therefore even the kinds of images that can be imported are limited. This was seen when distortions appeared on the screen with an increased complexity in the images used. These distortions were due to the small number of pixels for the V120. Therefore a more visually high-grade system can be designed with an advanced Vision controller. Nevertheless, in moments of limited resources, a V120 can still offer a high grade of service and still perform the remote controlling functions.

The third task involve the study of the different networks that are available and much attention was paid on the Devicenet network

4.2 Work Done

The work done in this project is outlined below:

i) The selection of the parameter was arrived at after studying the packaging systems that exist in the current system. And as it was stated in the project description, weight was more convenient for this study.

ii) The design was done using a Vision Controller (V120) and was mainly focused on the software development i.e. the application that will be controlling the whole system. The design of the code is shown in the Appendices.

iii) The software development was done using VisiLogic and this required a study of VisiLogic from the manual for the controller and the help menu in the ladder editor. The codes that came with the controller software were studied and these include; keypad entry code, mathematics functions code, basic coils and contacts codes, timer codes, linearization codes, displaying floating point code, displaying real time clock code and codes connecting between subroutines and displays.

iv) The simulations were done in the electronics laboratory and the outputs observed.

v) Knowledge on the use of programmable logic controllers was gained and their programming using ladder. Knowledge on the basic requirements in control and packaging system was also gained.

4.3 Cost of Project

The cost is one of the variables that need consideration during the design of a controlling system, in this case one that uses programmable logic controllers. In this design, PLCs are the most expensive and form an indispensable part of the system. The cost of a V120-22-R1 controller is approximately ZMK 5,000,000. Since the Vision controllers are not locally found, the freight charges are somewhere around ZMK600, 000. For a more efficient control system, an enfora GSM modem, if the mode of control will be notifying through SMS, has a rough cost of ZMK 850, 000. The transducers are not very expensive and have their lumped cost estimated at ZMK 650 000. In this case the main transducer to be used is the Digital weigh scale. The prices were obtained from Unitronics Israel in October 2008. The project would therefore require a huge capital investment of about ZMK 7,100,000

TABLE 1: Summary of the cost of the project.

Type Of Material	Cost
V120-22-R1	ZMK 5,000,000
GSM Modem	ZMK 850,000
Digital Weigh scale	ZMK 650,000
Freight Cost	ZMK 600,000
TOTAL	ZMK 7,100,000

CHAPTER 5

CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

The aims of the experiment were to design a system that would use PLC to feed parts into a pickup fixture when it is empty and sends a signal to the robot when to pick up a part and load it into the box until it reaches the prescribed weight. It was also required that devicenet is to be used to connect the PLC to the robot for data exchange. The aims of the project were achieved. The system was designed and tested in the laboratory. In order to meet the design requirements, a modular approach to the design was taken. This required that the whole system be broken down into modules both at hardware level and software level. The modules at software level required that the functions dealing with the parameter be developed and then integrated using appropriate software logic.

The hardware modular design was such that the input, output, and power supply were classified as separate modules. The power supply was designed independently as compared to the inputs and outputs which were dependent on the application.

This project on packaging and controlling system has brought out so many factors considered when designing a control system using programmable logic controller. In order to increase the performance of a system and resource utilisation, a proper system which takes into account the technical and economical factors must be formulated.

The method of communication between the data acquisition subsystem and the display or information section is among the factors that need to be noted to ensure that the network is efficient in its function as a control tool. In addition, coordination of the various modules should not be overlooked. This involves selecting appropriate transducers, arranging the elements such that they are activated in a logical order and that there are no conflicts between the systems.

These aspects contribute to the smooth running of a remote controlling system because not only is monitoring a must but also the management of losses is evident.

With the information contained in this report for the pricing and easy installation of a PLC monitoring system, we can therefore compare with the systems that Zambia breweries uses for monitoring their system for functionality (being one of the places visited during field visits). Even if this PLC system would not offer the same kind of versatility that their current system can offer, it is obvious that it would still work for their basic alarms and at a lower cost. The diagnostics in cases of any problems can be done locally unlike taking the monitoring tool abroad.

5.2 Recommendations

This project on food and beverage packaging is a guide to proper system design not only applicable to food and beverage but also to most distributed control processes. The following are the recommendations:

1. In order to increase reliability of the controlling system, all the pickup fixtures should have parts readily available for loading so that controlling and processing can be done in one place.
2. The main the pickup fixture should not be far from the loading machine and the container box, this is to allow a computer network cable should be run with only one repeater in between. This will facilitate the login by the operator from any point in the plant.
3. To make the system even more diverse and reliable, more advanced versions of the vision controllers should be used in the actual implementation e.g. the V530 controller.
4. The weight parameter in food and beverage industry is a very important parameter which if neglected can lead to catastrophic occurrences like under loading or over loading thus losses to the company. In this line, a further final year project could be done but this time testing the functionality of the weight measuring unit and not a simulation.

REFERENCES:

1. VisiLogic Help Menu.
2. Unitronics User Manual.
3. "Telemetry". *Microsoft Student* 2006[DVD]. Redmond, WA: Microsoft Cooperation, 2005.
4. Bartelt, T, "Industrial Control Electronics", 2nd Edition, Delmer Thomson Learning, New York, USA, 2002, Pp 257 to Pp 661 & Pp199.
5. Hugh Jack, "Automating Manufacturing System with PLC", Version 5.0, 4th May 2007, New York, USA. Pp 439 to Pp 452 also Pp 650 to Pp 672
6. L.A. & E.A, Bryan, "Programmable Controllers, Theory and Implementations", 2nd Edition, 1997 Atlanta, Georgia USA. Pp130, Pp 203 to Pp230,Pp 533 to Pp 554.
7. The Electrical Engineering Handbook-CRC Press 2000, Electronic Circuit Guide encyclopedia.
8. Langhton, Say, "Electrical Engineer's Reference Book", 14th edition.
9. J. David Irwin, "Basic Engineering Circuit Analysis", 5th Edition, 2007 New York, USA. Pp143 to 165.
10. www.plclink.co.uk, "PLC programming using ladder logic", 2008.
11. www.claymore.engneer.gvsu.edu, "Engineering journal on Automating Manufacturing system with PLC", 2008.

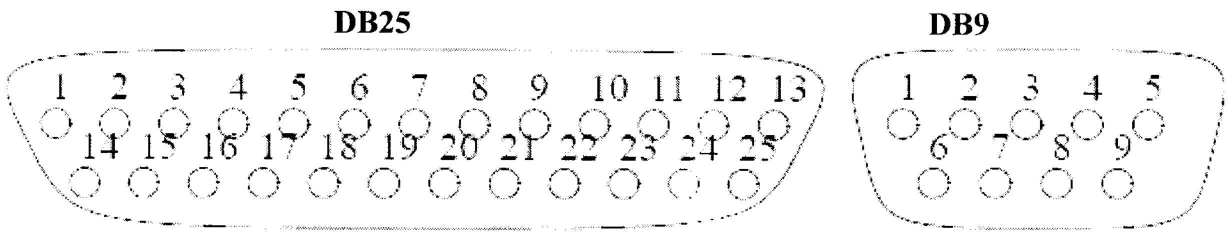
APPENDICES

APPENDIX A: COMPARISON OF NETWORK TYPES

TABLE 2

Network	topology	addresses	length	speed	packet size
Bluetooth	wireless	8	10	64Kbps	continuous
CANopen	bus	127	25m-1000m	1Mbps-10Kbps	8 bytes
ControlNet	bus or star	99	250m-1000m wire, 3-30km fiber	5Mbps	0-510 bytes
Devicenet	bus	64	500m	125-500Kbps	8 bytes
Ethernet	bus, star	1024	85m coax, 100m twisted pair, 400m-50km fiber	10-1000Gbps	46-1500bytes
Foundation Fieldbus	star	unlimited	100m twisted pair, 2km fiber	100Mbps	<=1500 bytes
Interbus	bus	512	12.8km with 400m segments	500-2000 Kbps	0-246 bytes
Lonworks	bus, ring, star	32,000	<=2km	78Kbps-1.25Mbps	228 bytes
Modbus	bus, star	250	350m	300bps-38.4Kbps	0-254 bytes
Profibus	bus, star, ring	126	100-1900m	9.6Kbps-12Mbps	0-244bytes
Sercos	rings	254	800m	2-16Mbps	32bits
USB	star	127	5m	>100Mbps	1-1000bytes

APPENDIX B: TYPICAL RS 232 PIN ASSIGNMENT AND NAMES



Commonly used pins

For DB 25:
1 - GND (chassis ground)
2 - TXD (transmit data)
3 - RXD (receive data)
4 - RTS (request to send)
5 - CTS (clear to send)
6 - DSR (data set ready)
7 - COM (common)
8 - DCD (Data Carrier Detect)
20 - DTR (data terminal ready)

For DB 9:
1 - DCD
2 - RXD
3 - TXD
4 - DTR
5 – COM
6 - DSR
7 - RTS
8 - CTS
9 - RI

Other pins

9 - Positive Voltage
10 - Negative Voltage
11 - Not used
12 - Secondary Received Line Signal Detector
13 - Secondary Clear to Send
14 - Secondary Transmitted Data
15 - Transmission Signal Element Timing (DCE)
16 - Secondary Received Data
17 - Receiver Signal Element Timing (DCE)
18 - Not used
19 - Secondary Request to Send
21 - Signal Quality Detector
22 - Ring Indicator (RI)

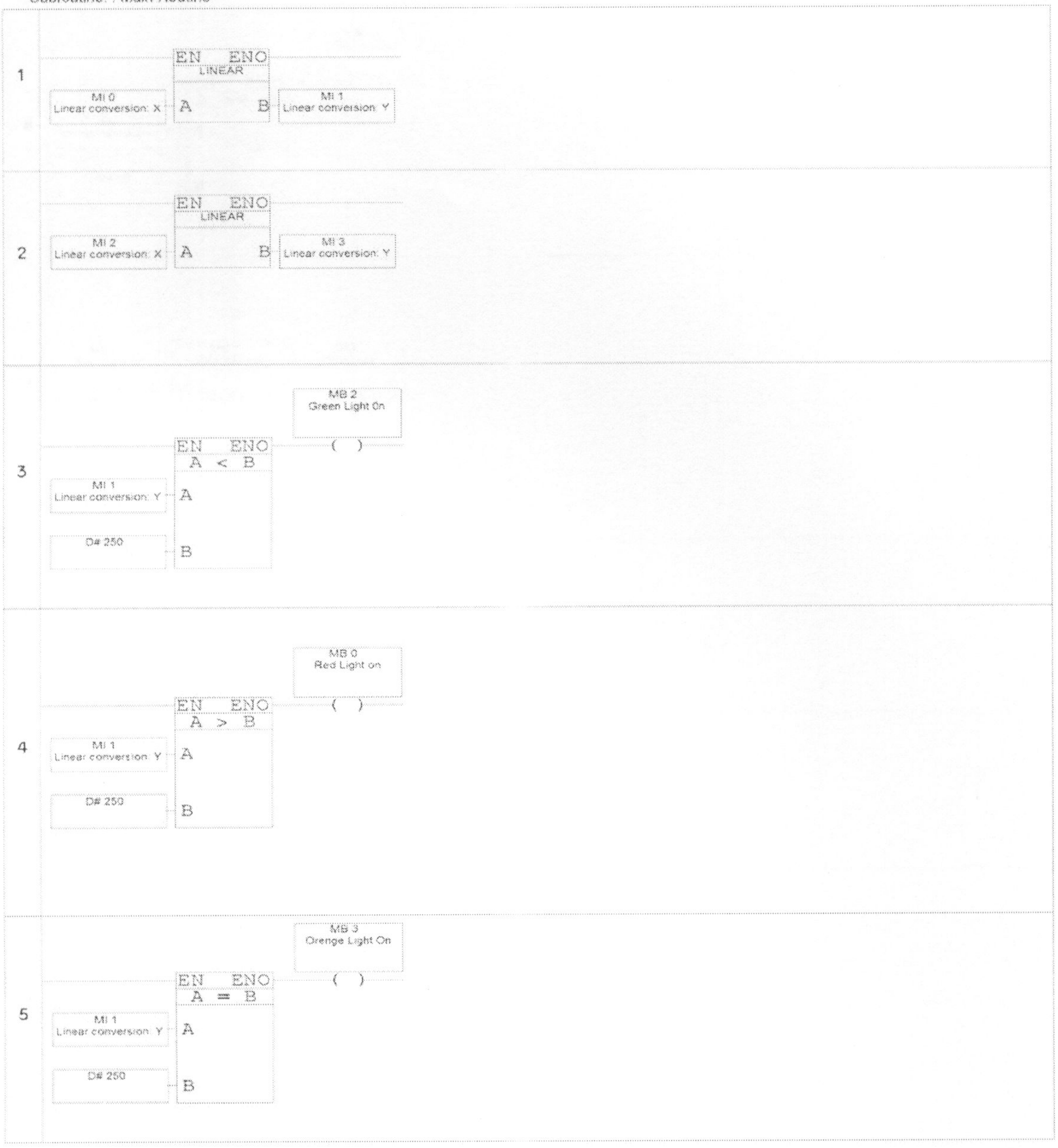
APPENDIX C: HMI Operands

TABLE 3:

SB 40	Key: # 0
SB 41	Key: # 1
SB 42	Key: # 2
SB 43	Key: # 3
SB 44	Key: # 4
SB 45	Key: # 5
SB 46	Key: # 6
SB 47	Key: # 7
SB 48	Key: # 8
SB 49	Key: # 9
SB 50	Plus/Minus
SB 51	Left Arrow
SB 52	Right Arrow
SB 53	ENTER
SB 54	Key <i> (ON when in Info mode, may also be turned ON in order to enter Info mode, via Remote Access or user program)

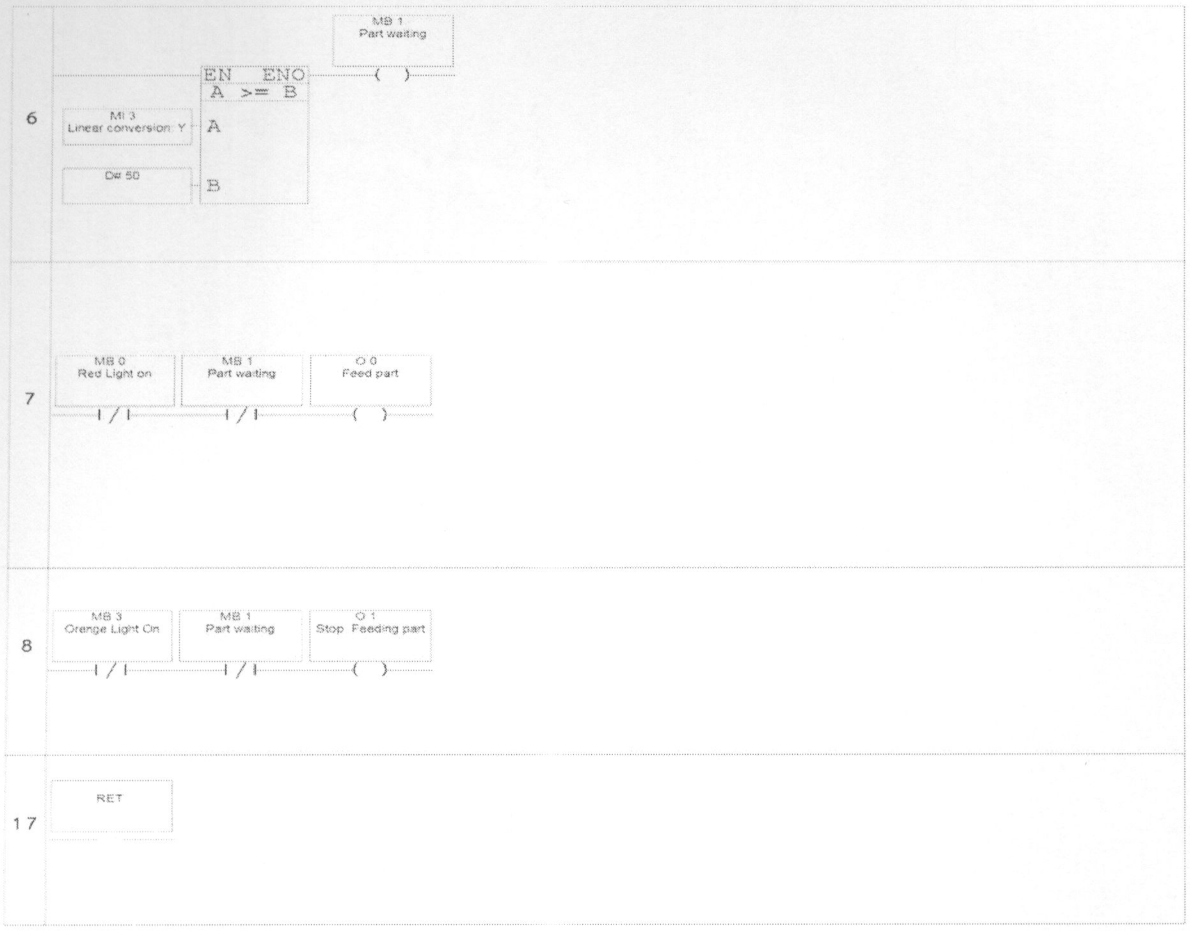
APPENDIX D: LADDER LOGIC CODE

Module: ! Main Module
Subroutine: ! Main Routine



ic_C\Examples\BW\Advanced\HMI\HMI General\martin final report.vlp Page 1 of 2 Thursday, April 02, 2009

APPENDIX D: Contd.


6
MB 1
Part waiting
EN ENO
A >= B
MI 3
Linear conversion: Y
A
D# 50
B
7
MB 0
Red Light on
MB 1
Part waiting
O 0
Feed part
8
MB 3
Orange Light On
MB 1
Part waiting
O 1
Stop Feeding part
17
RET