Implementation of Fieldbus Networks in Factory Automation

Seddik Abujreda

A thesis submitted in candidature for the degree of Master of Philosophy of the University of Wales

October 1999

DECLARATION

This work has not previously been accepted in substance for any degree and is not being concurrently submitted in candidature for any degree.

Signed....................................

Date..22..10....1999....

This thesis is the result of my own investigations, except where otherwise stated. Other sources are acknowledged footnotes giving explicit references. A bibliography is appended.

Signed....................................

Date..22..10....1999...

I hereby give consent for my thesis, if accepted, to be available for photocopying and for inter-library loan, and for the title and summary to be made available to outside organisation.

Signed....................................

Date...22...10....1999.

ABSTRACT

Over the last few years the importance of communication between machines has greatly increased, and this trend will continue as more industrial companies strive for enhanced production and efficiency through linked automation systems. Many of these systems use large numbers of input/output devices and operate over long distances, which leads to increased wiring and installation cost, besides low flexibility for system modification and extension.

One way of overcoming these limitations, as well as reducing the cost, is to use a serial communication system which is known as **fieldbus**.

The work described in this thesis is a study of some of the commonly used industrial networks, in particular Highway Addressable Remote Transducers (HART), Controller Area Network (CAN), Foundation Fieldbus, and Process Fieldbus (Profibus). In addition, the properties and applications of the Actuator Sensor Interface (AS-Interface) networking system which is for automation at the level of sensors and actuators has also been considered.

Case studies of three applications of fieldbus have been considered in detail. These are for the University of Wales College Newport (UWCN) Profibus DP training Network, a complete control system for the Mechatronic Development Centre's Computer Integrated Manufacturing (CIM) system using Profibus and AS-Interface networks, and a suggested solution for control problems at a local manufacturing company Brohome Ltd.

PLCs performs an important role to control different industrial processes. Its functionality, reliability and cost effectiveness puts it at the top of the list of industrial controllers. On the other hand fieldbus networks are originally designed to link those controllers so they can integrate with one another to achieve production tasks. A training package for one of the modern generation of PLCs has been developed and is included in the thesis.

ACKNOWLEDGEMENT

I would like to express thanks to my supervisors Dr Eifion Breeze and Dr Hefin Rowlands for their encouragement and support.

Many thanks goes to Professor Geoff Roberts, Director of the Mechatronics Research Centre for his support, and to my colleagues in Mechatronics Research Centre for their corporation

Also, I would like to extend thanks to E&L Instrument Ltd. Wrexham (UK).

Many thanks goes to Dr Abdulwahab Abdusalam for his encouragement and support

A special thanks goes to my family for their encouragement.

Contents

CHAPTER 1
Introduction

CHAPTER 2
Basic Principles of Networking

CHAPTER 3
Industrial Networks

CHAPTER 4
UWCN Profibus DP Training Network

CHAPTER 5
Profibus and AS-Interface for the Mechatronic Development Centre Laboratory

List of Figures

List of Tables

CHAPTER 1

Introduction

1.1 Problem definition.

Modern production techniques covering the demand of flexible manufacturing, just in time production, environmental laws, quality controls and the economic use of raw materials and energy, require companies to make use of automation.

Automation of the shop floor can be defined as the integration of individual machines such as robots, Programmable Logic Controllers (PLCs) and machine tools, into a single production system, using a common medium over which the controllers of related machines can communicate with each other and with other higher level controllers. To achieve this aim, the Local Area Network (LAN) concept has been adopted by vendors of PLCs and by manufacturers of computers for shop floor data collection and control.

Semiconductor technology has delivered the components that allow programmability and communication capabilities to be integrated and applied at relatively low cost to the automation of industrial processes. Advances in control, integrated circuits, software and communication technology have largely supported the movement towards larger and more complex plant which is necessary to take advantage of economies of scale.

Due to the increasing demand in the industrial world for new control systems, new vendor independent, multidrop, bi-directional digital communication systems known as **fieldbuses** have been introduced for communication between devices at shop floor level. These systems are increasingly meeting with international acceptance in all areas of industrial applications, from the automotive industry through to process engineering, materials handling, food processing and so on.

There are around 200 fieldbus systems on the market today (D. P. Lane, 1997). Each of them has been developed at different times, by a different company (or companies) and for different purposes. Choosing the right fieldbus is a matter of meeting the requirements of the application. This project discusses the properties and applications of some of the more commonly used fieldbuses. The outcome should be that informed decisions can be made in order to match a particular fieldbus to a specified application.

1.2 Fieldbus

Fieldbus has its roots in the 1980s. It can be described as an "all-digital, serial, two-way communications system at the base level in the hierarchy of plant networks, serving as a LAN for instruments used in process control and manufacturing automation applications and having a built-in capacity to distribute the control application across the network." (Fieldbus Foundation, 1998). It represents a key element that ensures interoperability and interchangeability (see chapter 3) enhances field-level control and reduces installation costs.

Fieldbus networks offer a multitude of advantages for the user, some of these are:

- All devices on the fieldbus network are connected to the main controller via one cable, this means significant reduction in wiring when compared to traditional control systems(M. Ochsner & M. Schrier, 1998).
- There can be a 5:1 reduction in field wiring expenses (Rolf, 1998).
- Fieldbus networks require less time to install than conventional systems.
- Fewer system drawings will be needed in order to develop a fieldbus system.
- Fieldbus systems are less complex than conventional control systems, which means less need for maintenance.
- The operator can easily view all of the devices in the system, so he can discover any problem and carry out maintenance more quickly.
- Fieldbus systems enable on line diagnostics to be carried out on individual field devices.
- Control cabinets in fieldbus systems are smaller. Unlike traditional control systems where all inputs and outputs must be connected to the main controller, in fieldbus systems most of the inputs and outputs can be connected to individual field devices.
- In fieldbus systems some of the control operations and intelligence can reside in the individual field devices, reducing the complexity of the main controller.

- Fieldbus systems are open systems, which means that the end user can obtain devices from various vendors.
- Some fieldbus protocols provide power via the communication lines.
- Fieldbus networks can be implemented through various wiring topologies.
- Fieldbus systems achieve quick response times, which is the most important factor in real time applications.
- Fieldbus systems allow for greater distances than with traditional networks.
- Fieldbus enables the total integration of data from device level to management system. This leads to greater efficient through the control of production, stock control, warehousing and handling of orders.

However, the implementation of fieldbus systems needs some knowledge of PLCs, which are being applied at ever increasing rate in industry. PLCs are now widely used and extend from small, self-contained units for use with a limited number of digital inputs and outputs, to modular systems which can be used with large numbers of digital and analogue inputs/outputs. PLCs are of the greatest importance to manufacturing and the process industries. The main advantages of PLC control over older, traditional methods are:

1. The construction and the programming methods of PLC, which have similarities to the traditional forms of wired logic, help to make it a directly accessible tool.
2. The possibility of saving programmes on disk and also printing them out are particularly valuable and are not possible in traditional technology.
3. Associated auxiliary devices such as printers and memory cards offer further facilities to the PLC.
4. PLCs enable much preparatory work to carried out away from site.
5. Development work on hardware and software can be carried forward in parallel.
6. The hardware, which is modular, can be rapidly adapted to new demands.
7. In PLC systems, there are no physical connections between the field-input devices and output devices. The only connections are through the control program.
8. The simplicity of PLCs allows personnel to be trained, to a good standard in a short time.
9. In PLC systems, dialogue between the operator and the system can be optimised.

10. The stored program can be modified, new control features can be added or old ones changed with out rewiring input and output devices.
11. PLCs have been designed with ease of maintenance in mind, so that faulty modules can be replaced easily.
12. PLCs can be programmed to aid the identification of faults.
13. Installing PLCs is easy and cost effective. Its relatively small size allows the PLC to be located conveniently in a much smaller space than that required by an equivalent relay control panel.
14. A PLC can replace all the relays that would have been used to provide control logic. This result in much more reliable system
15. Because the logic is contained within the PLC software program, changes can be implemented much more easily, in addition, complicated control algorithms such as PID and mathematical manipulation are possible.
16. The availability of communication modules allows the connection PLCs to industrial networks, which facilitate data, interchange through out a plant.
17. The use of a PLC is a preliminary step which does not commit the decision - maker irrevocably, but which allows development and progressive integration to higher levels of automation.
18. The PLC can be considered as the first important step in the automation of complex processes, leading to the ultimate level of integrated management allowing production to be tied to the commercial and financial activity of the company.
19. As the cost of PLCs has reduced and their functionality and reliability have increased, they have taken over from relay as the most widely used means of controlling plant and machines.

Millions of units are in service throughout the world today to testify to their functionality, reliability, flexibility and cost effectiveness (see table 1.1).

Year	*Units (mil.)*	*Revenue (bil.)*	*Growth rate*
1994	8.62	$ 4.65	10.5 %
1995	9.85	$ 5.20	11.8 %
1996	11.32	$ 5.84	12.4 %
1997	12.92	$ 6.54	12.0 %
1998	14.56	$ 7.25	10.8 %
1999	16.20	$ 7.94	9.6 %
2000	17.78	$ 8.60	8.3 %

Table 1.1 World market analysis for PLC systems (Industrial Networking 1999)

1.3 Aims and Objectives

The main aim of this work is to study some of the commonly used fieldbus systems and to compare their properties and suitability for different applications.

Since end users are interested in how the implementation of fieldbus systems can be achieved, the project discusses three different network applications:

1. The University of Wales College Newport (UWCN) Profibus training Network.
2. A complete control system for the Mechatronic Development Centre's Computer Integrated Manufacturing (CIM) system using Profibus and AS-Interface networks.
3. A suggested solution for control problems at a local manufacturing company, Brohome Ltd.

Finally, the importance of training the end user in the use of PLCs, probably the most important control devices on the shop floor, is taken into account and the project provides a training package for the S7-200 Siemens PLC. The training package explains the concept of PLC control, introduces the principles of operation of a modern PLC and provides practical exercises.

The objectives of this work are to:

- Illustrate the principles of networks with an emphasis on industrial applications.
- Examine some commonly used fieldbuses and to carry out a comparative study.
- Use the results of the comparison in the design of a real practical application.

- Produce a PLC training package to support the implementation of a fully networked system.

1.4 Outline of thesis

The remainder of this thesis is organised into the following chapters:
Chapter 2 begins with an introduction to the concept of industrial networks. This is necessary in order to understand fieldbus systems.

Chapter 3 discusses the different techniques used for linking various devices in a factory or process automation plant, focussing again on fieldbus technologies. In order to choose a fieldbus system to be used in the case studies, a comparison between the studied fieldbuses is carried out at the end of this chapter.

Chapters 4, 5, and 6 use the results of chapter 3 to explain the design of Profibus networks in three different applications. In Chapter 4 the University of Wales College Newport (UWCN) Profibus DP training network is studied. This chapter describes network implementation and concentrates on the practical aspect of networking, which is important for end users. This chapter includes sufficient detail to enable the realisation of a Profibus DP network.

Chapter 5 studies the functions of the various stations of UWCN's Mechatronic Development Centre laboratory and the implementation of an integrated Profibus DP and AS-Interface to the system is considered. The description includes the way in which devices can communicate with each other by means of a Profibus network and how the link between Profibus DP and AS-Interface can be achieved. In addition, a list of required components together with costs is provided.

Chapter 6 proposes a fieldbus system for production control in a real manufacturing company, Brohome Ltd., situated in South Wales.

Conclusions are discussed in Chapter 7.

Appendix A illustrates the operation sequence of the Mechatronic Development Centre's Computer Integrated Manufactory (CIM) laboratory.

Appendix B lists the costs of the parts used to upgrade Mechatronic Development Centre's Computer Integrated Manufactory (CIM) laboratory and Brohome Ltd. control systems.

Appendix C presents a practical training programme, which provides a foundation course for students and technicians who are involved in studying and using PLCs.

Appendix D discuses the concept of the availability of the system.

CHAPTER 2

Basic Principles of Networking

2.1 Introduction

A network is a communication system which allows a number of stations to exchange data with each other. Industrial networks are designed to transmit information for control, and in many ways are different to traditional computer networks, since the amount of data is less and the time of response is shorter. Their characteristics are determined by the rules that control the flow of data within the system. The lowest levels of industrial networks, which deal with the field devices are known as fieldbus systems. Real time control is the main task of fieldbus systems. In order to provide an understanding of these systems this chapter discusses some important topics.

2.2 Communication Systems

A typical communication system consists of five main parts:

1. Source (or sources) of data.
2. Transmitter, which converts the data generated by the source into a form compatible with the transmission system.
3. The transmission system which transfers the source's data to the receiver.
4. Receiver, that accepts the data from the transmission system and converts it into a form suitable for the destination device.
5. Destination device.

The various parts of the communication system must be compatible with regard to many factors such as signal type, coding, and the rate of data flow. This means that both ends of the system must use the same rules for data manipulation. These rules apply to the hardware in which case they are known as interface standards and to the software when they are known as protocols (G. Waters, 1991).

2.3 Transmission Modes

In communication systems, data may be sent in one direction or both directions. There are three ways to transmit data which are known as Simplex, Half duplex, and Full duplex (A. S. Tanenbaum, 1996).

2.3.1 Simplex

A simplex system, Figure 2.1, is designed to send data in one direction only. Examples of simplex systems are radio and TV transmissions.

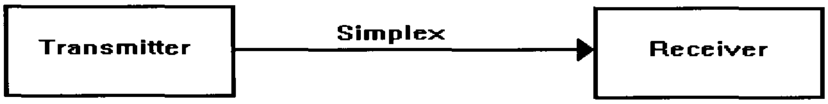


Figure 2.1 Simplex system

2.3.2 Half duplex

A system is said to be Half duplex when data can flow in both directions, but only in one direction at a time, Figure 2.2. This means that while one unit is talking the other must listen. After a time called the *turnaround time* they are able to reverse their roles. Two way radio, which is used by the police, is an example of a Half duplex system.

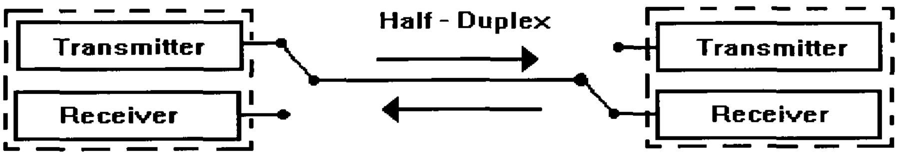


Figure 2.2 Half duplex system

2.3.3 Full duplex

With a Full duplex system, Figure 2.3, messages can flow in both directions simultaneously. The ordinary telephone system is an example of a full duplex system.

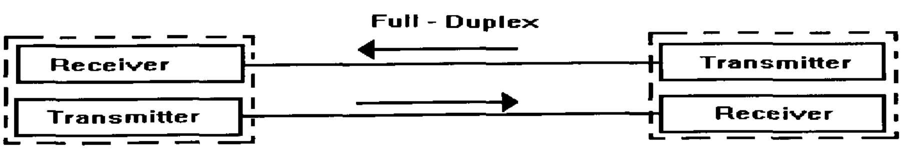


Figure 2.3 Full duplex system

2.4 Parallel and Serial communication

Information can be transferred as parallel or serial data (G. Held, 1997). In parallel communication, each bit of data is transferred from transmitter to receiver at the same time. Each bit requires its own conductor, and there must be at least one additional conductor to complete the circuit. Figure 2.4 shows that to transfer eight bits of data in parallel requires at least nine conductors, one for each bit and a common return (often called ground (GND)).

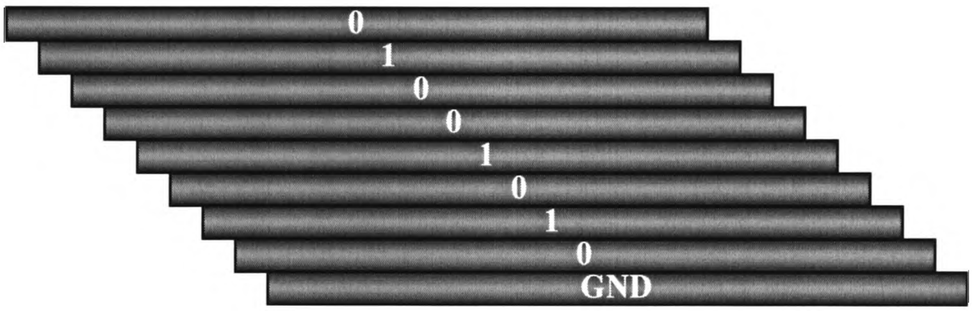


Figure 2.4 Parallel communication

In serial communication only one bit is sent at a time, with bits being sent one by one over the same line. Figure 2.5 illustrates that in this type of communication only 2 conductors are required.

Figure 2.5 Serial communication

Parallel communication is much faster than serial communication, but it is limited to short distances because of the large number of conductors required.

The main disadvantage of serial communication is its low speed. On the other hand, the main advantage of serial communication is that it requires fewer conductors than parallel communication.

2.5 Data Transmission Speed

The term data rate, used to describe a medium's capacity to transfer data from source to receiver, is measured in bits per second (bps). Data transfer speed depends upon:

- Communication medium.
- Distance between transmitter and receiver.
- The interface.
- The protocol.

The term baud rate (Maurice E. Baudot) is used to express the data transfer speed (J. Stenerson, 1993)

1 Baud = 1 bps

In practice, there are specific values of baud rate used as standards such as 50, 110, 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600, and 115200. Some protocols define other values such as 31.25 kbps, 1Mbps, and 2.5 Mbps which is used in Foundation fieldbus protocol.

Generally, the lower the data rate the less complex the requirements.

2.6 Asynchronous and Synchronous Transmission

Data communication systems can be asynchronous or synchronous (J. Quinn, 1995). **Asynchronous** means that the communication may occur at any time, and it is not tied to a clock with the result that there is no fixed timing relationship between one message and the next. Information from a computer keyboard is an example of asynchronous transmission, where the user may press the keys with different pausing times.

In these types of systems different sizes of messages can be transmitted. They can vary between a few characters up to hundreds of characters long, depending upon the protocol design.

Synchronous systems are generally tied to a common clock. These are used when high baud rates are required with a continuous stream of data. In these systems bigger frames are used, packed into Blocks. This allows more data bits to be sent and fewer *preambles* and *postamble* bits. The length of a Block is determined by the protocol, which needs to be more sophisticated to maintain a smooth flow of data.

2.7 Codes

A code is a method of representing data. There are many ways of coding data. Morse code has been used for many years in communication systems, where combinations of dots and dashes represent each number and letter. A dash is three times the duration of a dot. The main problem with Morse code is that it an open code, i.e. the number of pulses (dot & dash) which represents a letter is variable (for example _ represents T and _ _ . . _ _ represents a comma). This means that the receiver does not know where a character finishes.

2.7.1 Baudot Code

In 1874, a French telegraph engineer, Maurice Emile Baudot, introduced the first uniform length code (5-bit binary) which is based on Morse code (J. Quinn, 1995). A standard based on 5 bit binary code was adopted by the CCITT (Consultative Committee for International Telephone and Telegraph), and called Baudot Code.

In Baudot code all the characters consist of 5 bits, where the number of possible combinations is 32 ($2^5 = 32$), which is not enough to send text. For this reason it is not an ideal code for data communication. Baudot code solves this problem by using two shift keys for letters and figures, which doubles the number of characters that can be represented. This system was the forerunner of modern digital codes such as American Standard Code for Information Interchange (ASCII) and Extended Binary Coded Decimal Interchanging Code (EBCDIC) (see sections 2.7.3 and 2.7.4)

2.7.2 ASCII Code

The details of the ASCII code are specified in ANSI (American National Standard Institute) standard X3.4 - 1977, ITA # 5 (International Telegraphic Alphabet # 5) and ISO -

646 (International Standard Organisation - 646). There are two types of ASCII codes, namely 7bit code and 8bit code (J. Fitzgerald & A. Dennis, 1996). originally designed to be used with printers connected to computers. In addition to the printable code, which means the numbers, letters, punctuation marks or symbols on a printer or monitor, there are also non-printable codes, which are used to represent instructions or information to a device. Non printable codes, called control characters, can be accessed by pressing the control key on a PC keyboard.

ASCII code is the most commonly used code in data communication and control and most computers and PLCs are designed to accept it.

2.7.3 EBCDIC

Extended **B**inary **C**oded **D**ecimal **I**nterchanging **C**ode (EBCDIC) was developed by IBM for use in main frame computer systems(U. Black, 1993). In EBCDIC the bits are numbered in reverse order (bit 0 is the MSB and bit 7 is the LSB).

2.7.4 Gray Code

In ASCII code, a step from one value to the next, may involve the change of more than 1 bit. In Gray code only one bit changes each time a value increases by one which reduces the chances of errors.

This code is particularly suitable for some types of devices such as shaft position encoders, which give a code output of position in degrees.

2.7.5 Binary Coded Decimal (BCD) code

This is a 4 bit binary code for digits:

Digit	0	1	2	....	9
BCD	0000	0001	0010	....	1001

A decimal number, e.g. 22, in binary is 00010110

The same number in BCD is 0010 0010

BCD code is used in simple systems such as small instruments and digital panel meters, It is also used in larger, more complicated systems where interaction between the controllers and operator is required.

2.8 Data transmission

Most modern systems use digital signals to communicate with each other. There are several techniques that can be used to achieve the transmission of binary data, the simplest technique being shown in figure 2.6, where a lower voltage represents zero and one is represented by a higher voltage level.

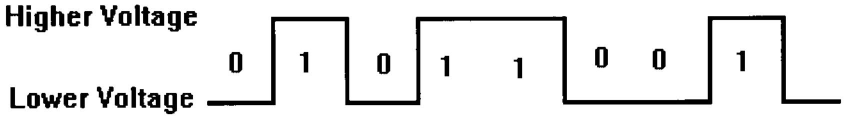


Figure 2.6 Data transmission

The following sections illustrate the operation of several different techniques.

2.8.1 Non Return to Zero (NRZ)

In this method the lower voltage is zero and the higher voltage is positive (normally +5 volt). It is called NRZ because the voltage does not return to zero between adjacent 1 bits (G. Held, 1997). This is suitable for use within devices where distances are short and paths are well shielded. However, because it is unipolar in nature it is unsuitable for long paths because of the presence of residual DC levels and the potential absence of enough signal transitions to allow reliable recovery of a clocking signal. To overcome this, signal conditioning devices are used to convert the unipolar waveform into a bipolar form to produce what is called **Polar NRZ**. Figure 2.7

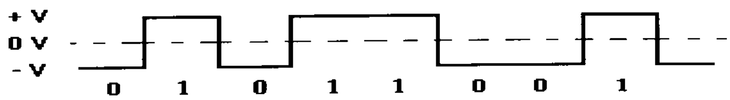


Figure 2.7 Polar NRZ

The main disadvantage of NRZ signals is the lack of synchronisation capability since for long strings of 0s or 1s, the voltage is constant over a long periods of time. This can lead to loss of synchronisation between the transmitter and the receiver if any variation of timing occurs between the two.

2.8.2 Bipolar AMI

In the case of bipolar **Alternate Mark Inversion** (AMI), 0s are represented by no voltage and 1s are represented by positive or negative pulses,(figure 2.8a). This technique will overcome the problem of synchronisation of 1s, although there is still a problem for long strings of 0s. Another advantage of this technique is that there is no net dc component because the 1s alternate in voltage from positive to negative. A similar technique which, instead of using no voltage to represent 0s, uses alternative positive and negative pulses is known as **Pseudoternary** and is shown in (figure 2.8 b)

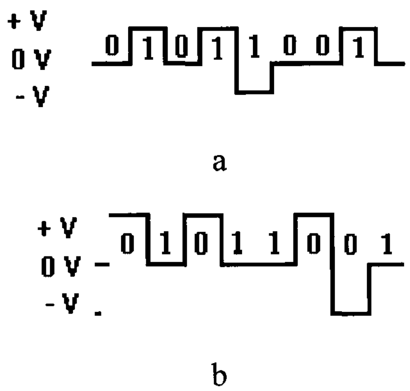


Figure 2.8 Bipolar AMI

2.8.3 Manchester coding

Manchester coding provides effective timing information by supplying a transition at the middle of each bit whether it be a zero or a one, figure 2.9. This serves both as a clocking mechanism and as data. There is no official standard to define which transitions define 0s and which define 1s. In Profibus PA and AS-Interface (as we shall see in chapter 3) a transition from low to high represents a 1, and a high to low transition represents a 0.

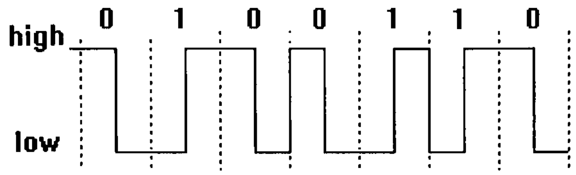


Figure 2.9 Manchester coding

2.9 Standards

Industrial devices and systems are becoming more and more complex, and the manufacturing of these devices and systems is more and more specialised. Each company has its own products and its proprietary solutions. This situation makes standardisation mandatory if devices of different origins are to work together. Vendors have to follow these standards if they do not want to lose their market positions.

There are several organisations involved in standards relating to industrial systems and networks. This section will focus on the most important organisations and their standards namely:-

International Standard **O**rganisation (**ISO**).

International **E**lectrotechnical **C**ommission (**IEC**).

Electronic **I**ndustrial **A**ssociation (**EIA**).

Institute of **E**lectrical and **E**lectronic **E**ngineering (**IEEE**)

2.9.1 ISO.

ISO was founded in 1946 (W. Stalling , 1997) and is one of the most important standard-making bodies. The membership is comprised of the national standards organisation of each ISO member country.

In 1984 ISO published a reference model for describing data communications architecture (International Standard # 7489) (J. Fitzgerald & A. Dennis, 1996). The name of the document is Open System Interconnection Reference Model (OSI/RM), simply called OSI or OSI model.

OSI is not itself a network standard but a framework to guide the implementation of network standards. In this model, data communication is broken down into the seven layers shown in (figure2.10)

Application
Presentation
Session
Transport
Network
Data Link
Physical

Figure 2.10 OSI model

Each layer communicates directly with the layers above and/or below it, where each layer has a defined purpose. For example, the application and presentation layers perform separate functions. The presentation layer provides a service to the application layer but the application layer has no knowledge of how the service is performed. To illustrate the principles of the ISO module we can think of the manager of a company asking his secretary to type a letter to be sent to another company. The secretary is the presentation layer and the manager is the application layer. The manager does not care how the secretary types the letter as long as it is done. This example applies to any two layers.

The model work as follows:

The Application layer, which is the highest layer, defines the application of the network. It works directly with the user.

The Presentation layer defines how applications can enter the network and represents the data in a format its user can understand. For example, if two different devices which use different formats are connected to a network the presentation layer translates data from one representation to another and insulates the user from such differences.

The Session layer is responsible for maintaining the data exchange session between the end users. A telephone call can be used as an example to illustrate the function of the session layer. Dialling the number represents the creation of a physical connection (layer 1). The session is then the dialogue.

The Transport layer defines the way of connecting the nodes, the protocol of the message delivery and the addressing of devices on the network.

The Network layer manages the routing strategies. Since networks may consist of many routes, this layer deals with which to choose. In other words it decides how information is routed around the network.

By means of the network access protocol the Data Link layer controls the flow of information between network nodes. It also specifies data format, breaking up large data blocks into several formatted frames. Each frame consists of a number of bytes.

The Physical layer carries the data over the network. It is concerned with physical aspects of communication, such as network topology, voltage and current values, signal modulation techniques, cables and connectors. This layer together with the Data Link layer constitutes the hardware layer.

2.9.2 International Electrotechnical commission (IEC).

IEC was founded in 1907 to develop standards in the field of electrotechnology, electronic appliances, radio communication, and transportation equipment.(J. R. Pimentel, 1990). In some cases, ISO and IEC have interests in the same areas. To reduce duplication of effort in the areas of microprocessor and information systems a joint committee was formed (J. R. Jordan, 1995).

2.9.3 Electronic Industrial Association (EIA)

EIA is a standards organisation representing the manufacturers of electronic equipment and components. The EIA produced several serial data interface standards, the best known of which is EIA-RS-232-C (EIA-Recommended Standard- standard No. 232 - Revision C). It is simply called RS 232.

RS 232 was developed for a single purpose as stated in its title (Interface between Data Termal Equipment (DTE) and Data Communication Equipment (DCE) employing serial binary data interchanging) (A. J. Crispin, 1997).

In RS 232 technology a 25-line cable using a 25-pin connecter connects DTE and DCE, each pin having a specific function (W. A. Shay, 1988). The main features of RS 232 are:

- Point to Point communication.
- Serial digital data communication, most data using ASCII code.
- Voltage levels -3 V to -25 V for logic 1and 3V to 25V for logic 0.
- Maximum communication distances about 15 m.
- Data rate 20 kbps.

Although RS 232 is still the most commonly used interface, with regard to modern requirements for data communication in industrial instrumentation and control it has several weaknesses, such as distance limitation, low data rate, not being suitable for connecting more than two devices and voltage levels not compatible with modern controllers. To overcome these limitations, other interfaces such as RS 423, RS 422, RS 485 and RS 449 are frequently used.

RS 423 standard defines an unbalanced data communication interface, which means that only one wire carries the voltage with reference to a common wire called ground.

RS 423 is similar to RS 232 but with some improvements to increase the maximum distance between devices, and the transmission rate. The main features of RS 423 are:

- Distances up to 1200m.
- Data rate up to 100 kbps.
- Up to 10 receivers.

RS 422 is an improvement on RS 423 and allows reliable serial data communication for distances up to 1200 m with data rate up to 10 Mbps. It defines a balanced or differential data communication interface, which requires two conductors to transmit the signal. The signal at the receiver is measured as the difference in voltage between the two conductors which means that it is better able to reject interference common to both lines.

The **RS 485** interface standard which is an extension of RS 422 has the same data speeds over the same lengths of cable but allows an increase in the number of devices on the line. RS 485 is very useful for control systems and fieldbus networks where several controllers (up to 32 devices) are connected together using a low cost medium which may be simply a twisted pair of conductors.

RS 485 defines the line voltages as:

-1.5 to -6 v for logic 1

1.5 to 6 v for logic 0

The largest improvement of RS 485 over RS 422 is that the driver can be operated in three-states, logic 0, logic 1 and high impedance.

In the high impedance state the line driver draws no current and appears not to be present on the line. This state is known as the disable state and can be initiated at any time by a control signal.

2.9.4 Institute of Electrical and Electronic Engineering (IEEE)

IEEE is a professional society in the United States, which focuses on local area network standards. The equivalent in Britain is the Institute of Electrical Engineers (IEE).

IEEE introduced the most important series of standards for local area networks known as IEEE 802. The IEEE standards committee administers this series through several working groups, each with its own section. Many of these standards have been superseded by ISO standards with an (8) prefixed.

IEEE 802.1 defines how the other IEEE 802 standards are related to each other and to the OSI model.

IEEE 802.2 divides the OSI model Data Link layer into two sublayers, Logic Link Control (LLC) layer and Media Access Control (MAC) layer. This standard defines the function of each sublayer.

IEEE 802.3 (Ethernet), defines the Carrier Sense Multiple Accessing with Collision Detection (CSMA/CD) protocol.

IEEE 802.4 (ISO 8802.4) defines the token passing bus access method. Physically it looks like a CSMA/CD network but with an operation like that of a token ring network. Stations see themselves as being arranged in a loop with each station assigned an address.

IEEE 802.5 defines the token ring access method when the ring is not only logical but also physical. The token is circulated on the ring and each station receives it, regenerates it, and puts it on the line again. Each station checks the message and if not intended for it, it simply sends it on along the network.

2.10 Network Topology

The way in which nodes are interconnected is known as network topology. The three most common topologies are Star, Ring or Loop, and Bus (or Multidrop)

In **Star topology**, figure 2.11, there is a central node and all the stations communicate back to it on separate links.

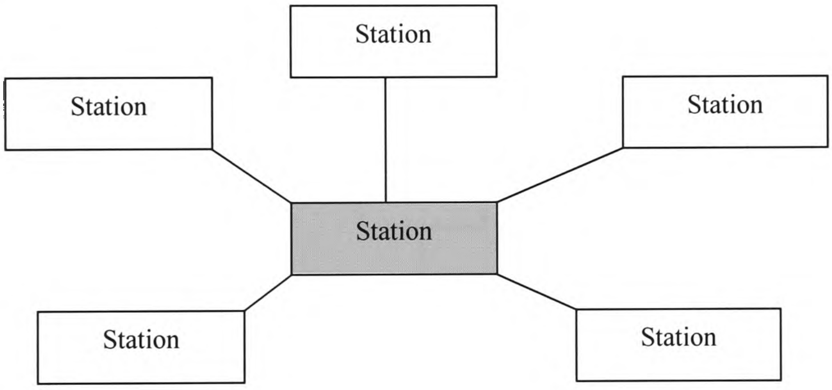


Figure 2.11 Star topology

The main advantage of star networks is that collisions are not possible, since each node is connected to the central node via its own cable.

The main disadvantage is that if the central node is disabled the entire system is inoperable.

In **Ring topology,** figure 2.12, nodes are connected together forming a loop, each node passing data to the next node and so on. The most commonly used system for these networks is token ring, originally developed by IBM in the 1980s. As stated, the specification of IEEE 802.5 defines this system.

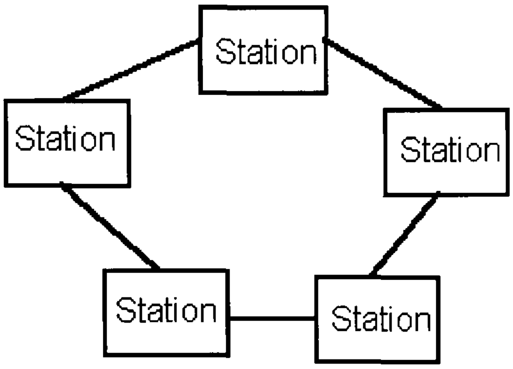


Figure 2.12 Ring topology

Bus topology consists of a communication path with nodes connected to it, figure 2.13. All the nodes can see data on the bus but only the destination node (or nodes) will read the data.

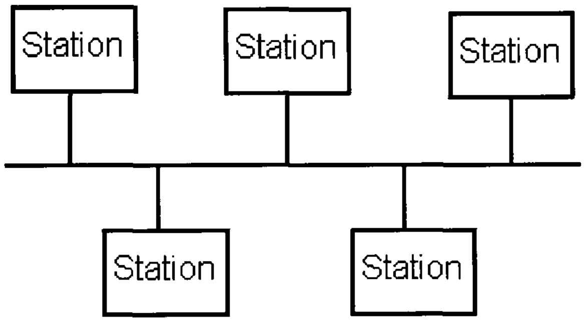


Figure 2.13 Bus topology

Unlike star and ring networks, in the event of a disabled node the network can still continue to operate as before.

2.11 Medium Access Control Methods

The two main techniques adopted to ensure that all nodes on the network can access it, are Carrier Sense Multiple Access with Collision Detection (**CSMA/CD**) and Token Control.

2.11.1 Carrier Sense Multiple Access with Collision Detection

This is used on bus networks where all nodes have access to the same cable. The transmitting node first encapsulates the data in a frame with the required destination node address. All nodes will read this frame and the one which identifies its own address in the message accepts the data and responds. However, it is possible for two nodes to try to transmit at the same time, which results in a collision. In order to minimise the chance of a collision, the source node first listens for a signal on the line before commencing transmission. It is also possible that two or more nodes will determine that there is no activity on the bus and both start to transmit at the same time. This means that a collision can still occur. The transmitting nodes sense this and enforce the collision by sending random bit patterns (Jam Sequence). The two or more transmitting nodes involved in the collision wait for a short random time interval before trying to transmit again. In this method there is no guarantee of how long a frame will take to reach its destination (G. Olsson & G. Piani , 1992)

2.11.2 Control token

A token is passed from node to node in a logical ring until it is received by a node wishing to transmit a frame. The transmission node then sends its frame to the next node in sequence. In a **token ring** a node wishing to send data must first wait for the token. When it receives the token it is able to transmit a frame containing a destination address. Each node repeats the frame until it reaches the destination node which retains a copy of the frame and indicates to the transmitting node that it has received the data by setting an appropriate bit in the frame. The **token bus**, which has been developed for use in factory automation and described in IEEE 802.4 is mostly used in industry (L. Mackenzie, 1998). Token bus stations wishing to transmit data, organise themselves into a logical ring for token-passing purposes. A device connected to the token bus can be active or passive. Active devices circulate the token and may transmit whenever they hold the token. Passive stations may issue messages only when the active stations address them. Their reply or acknowledgement must be immediate.

2.12 Error detection and correction

Noise spikes can change 1s to 0s and 0s to 1s during transmission. A number of techniques have been developed to detect and sometimes to correct errors. All of these methods involve the transmission of redundant data, which is not a part of the information. This section describes some of the more common forms of error detection and correction.

2.12.1 Parity.

This form of error detection, which is frequently used with ASCII code, adds a redundant bit, called a *parity* bit to the end of the character. The redundant bit is placed in the Most Significant Bit (MSB) position. Two types of parity can be used. **Even parity**, which means that every 8-bit data word in the message has an even number of binary 1s, and **odd parity** where every word contains an odd number of 1s. As an example for even parity, the letter V is encoded 0110101 using 7-bit ASCII. Since the number of 1s is 4, the parity bit in this case will be 0, yielding 00110101 as the transmitted message.

In a similar way, the letter W is 0001101. A 1 is added in the parity position to make the number of 1s even, so W becomes 10001101. The same type of parity must be used in the transmitting and the receiving terminals. If an error occurs the receiver will detect it, but there is no information about which bit was in error. Also if two bits are changed no error can be detected. To overcome these problems Longitudinal Redundancy Checking (LRC) has been developed.

2.12.2 Longitudinal Redundancy Checking (LRC).

To overcome the limitation of the Parity error detection, one additional character is added to the end of the message or packet of data. This character is called the **block check character (BCC).** In the same way as the parity bit the value of BCC is determined, but by counting longitudinally through the message, rather than by counting through each character. By counting the number of 1s in the first bits of all characters in the message the first bit of the LRC is determined, and set to 1 or 0 depending upon the parity used. The second bit of the BCC is determined by counting the number of 1s in the second bits of the characters in the message, and so on for all the bits of the BCC (J. Fitzgerald & A. Dennis, 1996). For example, the message (HART) can be sent using odd parity and odd LRC with 7-bit ASCII as follows:

	Letter	Parity bit
H	1001000	1
A	1000001	1
R	1010010	0
T	1010100	0
BCC	1110000	0

As the message arrives, the receiver checks the horizontal parity of each character, generates its own BCC and compares it with the check character received at the end of the message. The two should be identical.(J. Quinn, 1995).

2.12.3 Arithmetic Checksum

This is a simple sum of characters in the block. The transmitter sends the least significant byte of the sum as an extra character called the **checksum** at the end of a data block. By summing the data as it is received the receiver generates its own checksum and compares it with the checksum from the transmitter. If the two are identical, it is likely that no error has occurred.

2.12.4 Cyclic Redundancy Check (CRC)

This is one of the most powerful, error detecting methods, which adds 8, 16, 24, or 32 bits to the message (J. Fitzgerald & A. Dennis, 1996). For example, a communication protocol using a 16 bit CRC calculates a 16-bit number that is a function of all the data in the message. The transmitter adds this 16-bit number to the end of the message block. At the other end the receiver calculates its own 16 bit CRC as the block is received. If there is a difference, an error has occurred.

In this method the message is treated as one long binary polynomial, P. Before transmission, the data link layer divides P by a fixed binary polynomial, G, resulting in a new polynomial, Q and a remainder R/G.

$$\frac{P}{G} = Q + \frac{R}{G} \text{ ---------------- } 2.1$$

Before transmission, R is attached to the message, as a check sequence, k bits long. The receiving data link layer divides the received message by the same G, generates an R and checks the received R against the locally generated R.

2.13 Summary

In communication systems transmission of data may be, **simplex** (one way communication), **half duplex** (either way communication), and **full duplex** (both way communication).

Data can be sent in serial or in parallel mode. Serial communication is the transfer of information one bit at time and requires two conductors. Parallel communication is the transfer of several bits simultaneously and this requires one conductor for each bit, plus a common or ground conductor.

Any two entities, which are communicating, must follow a protocol, which is a system of rules. Protocols are defined in standards, defined by various national and international organisations. This chapter has focused on the most important organisations and their standards relating to the subject of industrial networks.

Protocols define the codes that are used in data communication. The most commonly used in data communications are Baudot, ASCII, EBCDIC, Gray, and BCD.

In addition, protocols define various error detecting methods. Parity uses an extra bit or bits added to each data character. A parity bit is set to ensure that each character contains either an even or odd number of 1s. A checksum is the arithmetical sum of all the binary characters transmitted in a block of data. Both the transmitter and the receiver calculate the checksum. The receiver compares the received checksum with its own checksum.

In Cyclic Redundancy Check (CRC). Identical circuits are used at the transmitter and receiver. At the end of a block of data the sending terminal transmits its CRC character and the receiver compares it with the CRC character that it has generated.

Data may be sent either synchronously or asynchronously. In synchronous data communications the clock signals of the transmitter and the receiver must have the same frequency.

Data communications use a number of different methods to represent digital information on the network. Commonly used methods in industrial networks such as NRZ, Bipolar AMI, and Manchester coding have been discussed in this chapter.

Media access control is required for serial networks. Different systems use different methods. The two main techniques adopted for industrial networks CSMA/CD and token control, are used for different topologies. The most commonly used types of topologies are Star, Bus, and Ring. A Star network has communication links radiating from a central terminal to its remote stations so that each remote station has its own dedicated point-to-point link that connects it to the master. In Bus topology, some-times called Multidrop, all terminals share a common communication link. Ring topology as its name implies connects all the terminals in the form of a circle or ring. The next chapter discusses factory automation protocols, which make use of the networking protocols and standards described in this chapter.

CHAPTER 3

Industrial Networks

3.1 Introduction

This chapter discusses the concepts of factory automation and, since Manufacturing Automation Protocol (MAP) represents the first step towards developing an open industrial network, the main features of this protocol are described. An explanation of fieldbus technology is presented followed by an examination of the main features of the most commonly used fieldbuses; Highway Addressable Remote Transducer (HART), Controller Area Network (CAN), Foundation Fieldbus, and Process Fieldbus (Profibus) together with a comparison of their properties and performance. Finally a brief study of Actuator Sensor - Interface (AS-Interface) technology is provided.

3.2 FACTORY AUTOMATION

The term *factory automation* refers to manufacturing facilities with a high degree of automation. Other terms are used to describe similar concepts such as *Flexible Manufacturing Systems,* **FMS** and *Computer Integrated Manufacturing* **CIM**.

Various definitions are given to the terms FMS and CIM (G. Olsson and G. Piani 1992), (P. G. Ranky, 1990), (Ian G. Warnock 1988). The following definition is that given by O. Grady and Ranky (J. R. Pimentel, 1990)

An automated manufacturing system is an interconnected system of material processing stations capable of automatically processing a wide variety of part types simultaneously and under computer control.

The system is not only interconnected by its material transport systems, but also by a communication network for integrating all feature of manufacturing. The system possesses flexibility in routing elements, part processing operations, co-ordination and control of part handling and the use of appropriate tooling.

3.2.1 Models for factory automation

Different models such as *SME CIM* by the Society of Manufacturing Engineering **(SME)**, figure 3.1, **Advanced Factory Management System (AFMS)** by Computer Aided Manufacturing Inc. and the **Automated Manufacturing Research Facility (AMRF)** of the National Bureau of Standards have been introduced to describe the term factory automation (J. R. Pimentel, 1990). SME CIM do not take into account the levels of control, which are responsible for achieving a completely automated manufacturing facility. Since this model has no consideration of the hierarchical organisation structure of most manufacturing facilities, it is not used to illustrate the concept of networks in manufacturing. On the other hand, **AFMS** and **AMRF** Models do show the different levels of control systems. These models are described in tables 3.1 and 3.2 respectively.

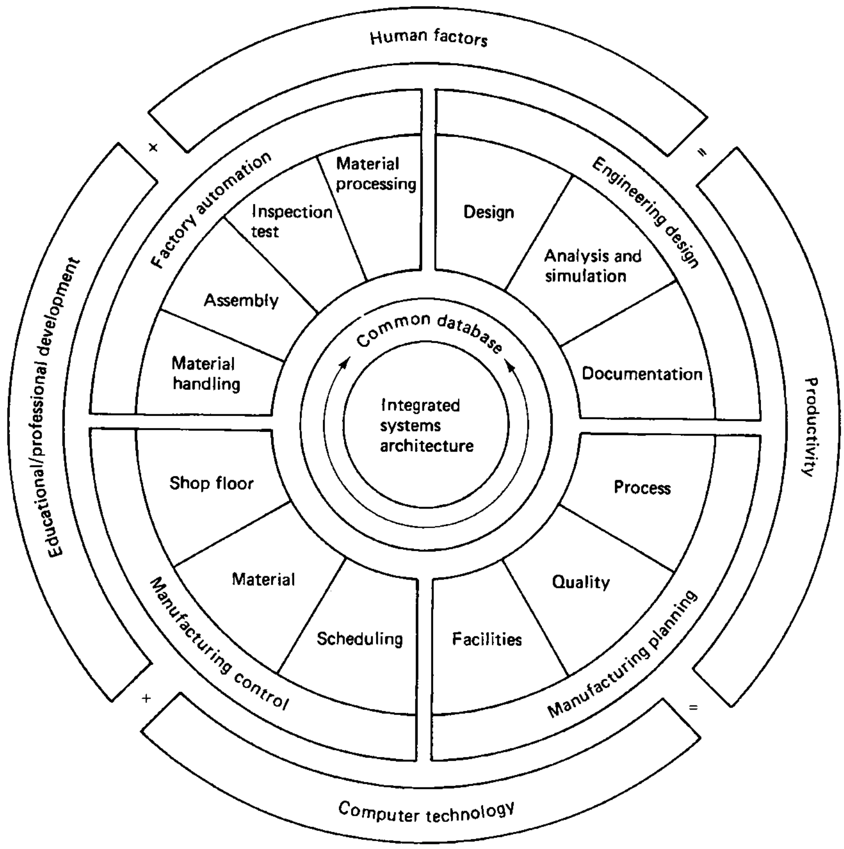


Figure 3.1 The SME model

The **Advanced Factory Management System (AFMS)** model consists of four levels as shown in table 3.1

Level	Function
Factory control system.	Determine end product requirements, aid product planning and determine individual capacities and capabilities.
Job shop	Takes commands from the upper level (Factory control system) to generate commands for the Work Centre Level.
Work Centre	Take orders from the job shop level and generate task requirements. Commands for these tasks are passed to the next lower level.
Unit / resource	Carry out the tasks.

Table 3.1 AFMS model.

The **Automated Manufacturing Research Facility (AMRF)** model is similar to the AFMS with an additional level. The hierarchical levels in the AMRF are shown in table 3.2.

Level	Function
Facility	Information management Production management Manufacturing Engineering
Shop	Task management Resource allocation
Cell	Batch management Scheduling
Workstation	Set-up, Equipment tasking Take down
Equipment	Machining Handling Monitoring

Table 3.2 AMRF model.

The various levels use different networks for connecting devices and cells. The connection between these different networks is achieved by using gateway devices. A gateway is a 'dual' architecture device, which on one side has all the layers of a protocol to suit a particular network and on the other side has all the protocol layers to suit a second network. In some cases not all control levels are required. For example, if a system contains only a small number of devices which are not connected to any other system. For example, in chapter 6 the proposed system for Brohome Ltd. uses only two layers. Figure 3.2 illustrates the networks that are commonly used at different levels (Schickhuber and Mccarthy 1997).

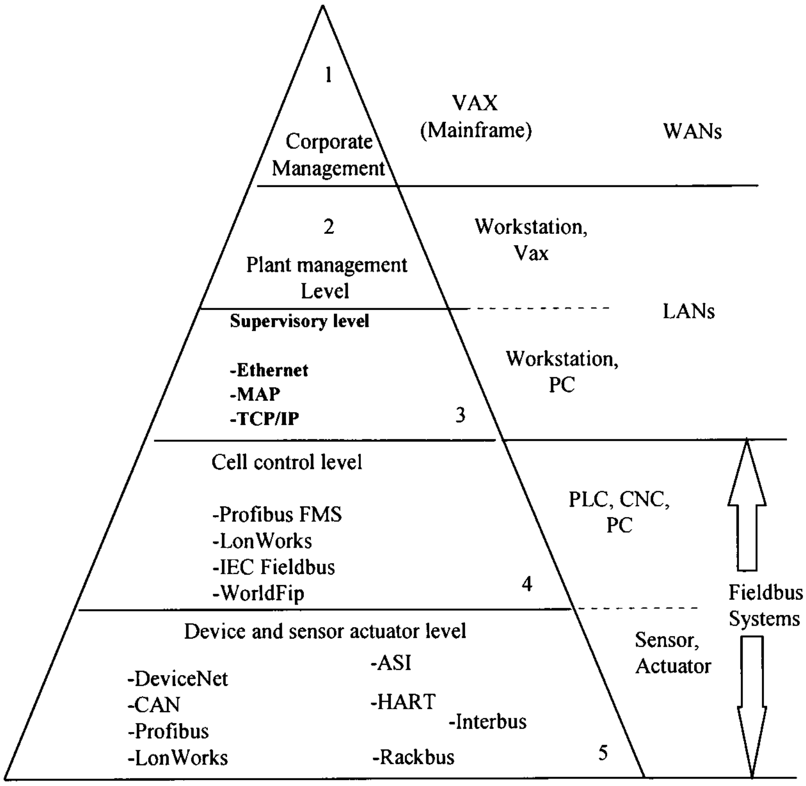


Figure 3.2 Example of CIM levels.

3.2.2 GMT 400

As an example of factory automation this section will present a project developed by the truck and bus group of the General Motor Corporation and referred to as GMT400 (J. R. Pimentel 1990). The target of the project was to improve productivity to reach an output of 60 trucks per hour. The system architecture consists of a hierarchical control structure having five control levels:

- Information processing Centre
- Plant host
- Area manager
- Plant floor
- Programmable device level

The main function of each level is as follows:

Information Processing Centre. This is the highest level in the control hierarchy. Its main function is to order materials, process customer orders, and carry out financial calculations.

Plant host. This is situated in each assembly plant. It supports plantwide applications such as flexible scheduling, rescheduling, and shipping control. It also controls the flow of unfinished parts through the plant, since unfinished products can follow several paths in the production process.

Area manager. The function of this level is to monitor vehicle information, manage the movement of components through the plant and carryout statistical process control.

Plant floor. This level receives its commands from the area manager level. Its main function is to monitor the control and dynamic configuration of automated equipment on a real time basis.

Programmable device. This level contains manufacturing devices such as robots, PLCs, automated guided vehicles, and weld controllers.

Network architecture

The GMT400 project has only one *Information Processing Centre (IPC)*, which supports many plants. The IPC and the plant host levels are interconnected by a *long haul network*, which is a type of WAN. Currently, IBM's System Network Architecture, SNA, which is a tree structured architecture with a mainframe host computer acting as the network control Centre (P. G. Ranky 1990) is used for communication between these levels (J. R. Pimentel 1990). In the area manager level, the numbers of controllers used are few, as shown in table 3.3, which illustrates that for every level there are different needs.

Level	Number of Devices
Information processing Centre	1
Plant host	2
Area manager	4
Plant floor	59
Programmable devices	6272

Table 3.3 Number of devices at assembly plant.

The interconnection at the area manager level and the plant host level is achieved by using a SNA gateway on a point to point basis.

A main requirement for communications at a level lower than the area manager level involves the use of terminals for providing access to the control hierarchy. This is accomplished by interconnecting the appropriate devices with the terminals by means of a local area network. A CSMA/CD (see chapter 2) based broadband network is used, because the same network medium is shared with two other local area networks, as discussed below.

A MAP 2.1 broadband network supports communication among the area manager, the cell controllers, and certain robot controllers.

Communication between the plant floor level and programmable device level is provided by a variety of networks. In most cases, the cell controller supports proprietary networks (e.g., Allen Bradley's Data Highway). Robots and PLCs are connected to these networks. In addition, networks based on CSMA/CD, and MAP are also used for communication between devices on the plant floor and programmable device level. Communication between the PLCs and their sensors and actuators is provided by the specific vendors, consisting typically of hardwire interconnections.

The reason for introducing (GMT 400) is to illustrate in practical terms the concept of factory automation, and to provide an understanding of the meaning of lower level control. Studying the higher levels is beyond the remit of this work.

3.3 Manufacturing Automation Protocol (MAP)

MAP is a communication protocol based on the full seven-layer OSI model (X. Jin 1994), which was proposed to solve the problem of connecting devices from different vendors.

At American car manufacturer **General Motors (GM),** management noticed that rewiring costs were related to retooling for every new car model. According to estimates made at the beginning of the 1980s, by 1990 there should have been about 100,000 different units such as robots and PLCs to interconnect at a typical GM plant, The cost for these interconnections would take a major share of the total company investments in automation. GM decided therefore to develop a comprehensive and standard approach to plant floor communication, which has led to what is known today as **Manufacturing Automation Protocol (MAP)**

3.3.1 MAP Objectives

MAP objectives were to provide different features such as (M.G. Rodd/F. Deravi, 1989):

- The capability of sending and receiving messages and commands between two or more control systems.
- A facility, to ensure that data arrives at a destination without having been corrupted.
- The ability to send information files between different systems.
- Common formats to support communication between shop floor devices.

3.3.2 MAP Specification

In 1984 the first specification (version 1.0) was published. International MAP activity was established and by 1987 version 3.0 of the MAP specification had been published (J. R. Jordan, 1995). MAP is the result of a concept to realise interconnections between different equipment at plant floor level and to higher planning and control systems to achieve so called open communication (G. Olsson & G. Piani , 1992). The principal goals of open communication are **interoperability** (*all information should be understood by the addressed units without the need for conversion programs*) and **interchangeability** (*a device replaced with another of a different model or manufacturer should be able to operate without changes in the rest of the connected system*).

3.3.3 MAP Layers

Since MAP is primarily intended to support the manufacturing environment, its layers must meet the requirements of manufacturing applications. OSI did not have appropriate application protocols, so MAP developed one called **Manufacturing Message Format Standard (MMFS)** which has evolved into a more sophisticated protocol called **Manufacturing Message Specification (MMS)** (see section 3.2.5). MAP layers are illustrated in table 3.4 (M.G. Rodd/F. Deravi, 1989)

Layer	Specifications			
Layer 7 Application	MAP network management based on: 1st DPs for ISO/DP 9595 Parts 1 & 2 ISO/DP 9596 Parts 1 & 2	Manufacturing message system (MMS) ISO/DIS 9506 Parts 1 & 2	File transfer access and management (FTAM) ISO 8571 Parts 1–4	Directory services ISO/DP 9594 Parts 1–8 Remote operations ISO/DIS 9072 Parts 1 & 2
	ISO 8649/2 : ACSE service ISO 8650/2 : ACSE protocol			
Layer 6 Presentation	ISO 8822 : Presentation service ISO 8823 : Presentation protocol ISO 8824 : ASN. 1 (abstract syntax one) definition ISO 8825 : ASN. 1 basic encoding rules			
Layer 5 Session	ISO 8326 : Session service ISO 8327 : Session protocol			
Layer 4 Transport	ISO 8072 : Transport service (Class 4) ISO 8073 : Transport protocol			
Layer 3 Network	ISO 8348 : Network layer service ISO 8348/AD1 : Connectionless mode transmission ISO 8473/AD1 : Network layer protocol ISO/DIS/9542 : ES to IS exchange protocol			
Layer 2 Data link	ANSI/IEE 802.2-1987 : Logical link control (Class 1) (Equivalent to ISO 8802/2 ISO8802/4 : Medium access control (token-passing bus)			
Layer 1 Physical	IEEE 802.4 Draft H : Broadband token bus IEE 802.4 Draft H : Carrier band token bus			

Definitions:

ISO : International Standards Organization
IEEE : Institute of Electrical & Electronic Engineers
IS/DIS/DP : Potential ISO standards are balloted 3 times via interested, representative parties:

1st Stage : Draft proposal (DP)
2nd Stage: Draft international standard (DIS)
3rd Stage : International standard (IS)

Table 3.4 MAP 3.0 layers specification (M.G. Rodd/F. Deravi, 1989)

MAP protocol was developed with a number of options for the physical layer. The **broadband** version of MAP has the highest capabilities where several types of communications can take place on the same cabling at the same time. On the other hand, because of the greater flexibility, a broadband system is more complex and more expensive to install. It needs modems and MAP interface equipment for each item (S. Soloman 1998). The main cable used for broadband is unwieldy (approximately 25 mm in diameter) and is suitable only for wiring very large factories. Multiple drop cables can be branched off the main cable for the different MAP nodes.

A different type of MAP network uses **carrier-band** technique, which uses slightly less expensive modems and interface units and does not need heavy-duty cable. For a small factory, a carrier-band version can be much more cost-effective.

To link a broadband factory-wide communication network to a local carrier-band network, devices called *bridges* can be used. The bridge transforms the message format presented on one side to that required on the other side.

Although MAP represents one of the first stages towards the target of open systems, it faces some restriction such as:

- Various versions of MAP have been produced. It is difficult to obtain an agreement between different countries and vendor groups on a particular standard. The resulting standards are so extensive that they have become very complex, making it hard to develop the hardware and software to implement the system and as a result driving up related expense.
- The compatibility between MAP equipment units, i. e. interoperability, has been a continuing difficulty because of complexity.
- Because of the cost, and disagreements on how it should be implemented the use of MAP has not grown as rapidly as was initially hoped by its proponents.
- It is difficult to assemble a complete set of documentation for MAP activity , since this requires the accumulation of a file of standards organisation reports, a number of industry organisation reports, and documentation from all the working committees associated with ISO.

3.3.4 Manufacturing Message Specification

Manufacturing Message Specification (**MMS**) is a set of commands used for the remote control of industrial equipment. MMS as an ISO standard (ISO/IEC 9506) consists of six parts, namely Service definition, Protocol definition, Robot control messages, Numerical control Messages, PLC messages and Process control messages. MMS is based on the object-oriented programming concept, where the operation that can be performed on an object and the class of the object is defined. The main concept of MMS is the **Virtual Manufacturing Device (VMD),** which is a collection of all possible commands for a particular device. A real device will react to VMD commands as predefined in the standard. MMS is based on the 'client - server' model, where the client requests a service from the server. The server executes the service and provides an answer to the client to specify the result of the operation. A VMD represents the function of a real device, but to the client there is only the virtual server. The VMD command sent to a virtual device will result in a different, real device taking the same action. In the case where the real device is not able to execute the commands it will respond with an operation result code stating the reason (G. Olsson and G. Piani 1992).

3.3.5 MAP/EPA

MAP/EPA (EPA stands for Enhanced Performance Architecture) is a dual system incorporating on the one hand a full- MAP protocol and on the other hand a reduced version (some times called a collapsed architecture), which consists of layers 1, 2, and 7.
A MAP/EPA station can switch to either of the two communication paths. The seven-layer side links it to the full MAP stations and the collapsed-architecture side provides high-speed access to other similar EPA stations. Figure 3.3 shows the schematic of MAP/EPA (M.G. Rodd/F. Deravi, 1989).

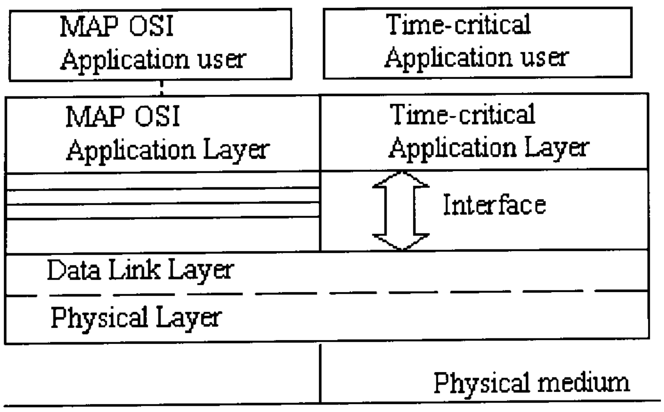


Figure 3.3 MAP/EPA

3.3.6 Mini-MAP

Mini-MAP is the reduced version of MAP/EPA. It can be used if compatibility with the ISO module is not important. The right hand side of figure 3.3 shows the architecture of mini-MAP. Other standards have been specified to support mini-MAP, which are non-MAP-compatible. These are low-cost, real time buses, which can be used to connect devices at the lowest level. These standards the so-called fieldbuses, are the main subject of the next section.

3.4 Fieldbus

Fieldbus is the name given to a communication system specifically designed for the networking of devices at the lowest levels in manufacturing systems or process control hierarchies. It is a bi-directional digital communication technology. Data exchange at fieldbus level is characterised by heavy traffic of short messages that must be securely transmitted with quick response time (M. Soufian, 1998). It was in the late 1970s that this technology began to attract industrial attention (M. G. Rodd *et al*, 1998)

3.4.1 Process and manufacturing automation

Fieldbus systems are used in various applications such as buildings, automobiles, transport systems, and factories. This project concentrates on using fieldbus systems in the manufacturing environment. The main two branches of industrial production are process automation, and factory automation.

Process automation deals with the automatic operation of continuous processes such as power plants, cement kilns and petrochemical plants. It has a very large material flow and often need greater safety requirements. Factory automation is concerned with the production of discrete objects. Initially, this involved isolated production islands for maximum quantity throughput. It was not until much later that these islands were networked to achieve flexible manufacturing, with short re-tooling times (J. R. Jordan 1995).

3.4.2 Fieldbus Devices

Devices interconnected by fieldbus networks can be any of the following:

Simple, dumb devices such as solenoid valves, motor starters, limit switches, relays, and simple sensors and actuators.

Simple intelligent devices, such as transmitters, receivers, intelligent actuators, and smart sensors.

Complex devices such as PLCs, CNC machine controllers, robot controllers, and PCs.

Compared with higher levels in the control hierarchy, the requirements of interconnecting these devices at the fieldbus level are different. The main features of fieldbus networking are:

- Quick response time. "*It is a common experience that in most manufacturing and automation systems, a guaranteed maximum delay time across a network of 5 ms appears to be reasonable*" (T.Karppinen, 1998).
- Short messages.
- Low cost.
- Good reliability and safety.
- Easy installation and maintenance.

The need for a quick response time is the most important feature of the above list. It distinguishes fieldbus networks from local area networks or other types of networks used at the higher levels.

To meet these requirements, fieldbus networks must be designed with the following considerations in mind:

1. The Fieldbus system should guarantee error-free messages between field devices and between these devices and higher level devices.
2. The fieldbus devices must be designed to guarantee safety requirements for operation in hazardous areas.
3. The fieldbus protocol must support cyclic transactions and event-driven acyclic transactions at low network interface costs.

4. The fieldbus protocol should be designed to permit the possibility of changing, adding, and removing devices without disturbing communications between other devices.

3.4.3 Fieldbus Layers

As was mentioned in the previous section, fieldbus is a form of computer networking which can be described in terms of the OSI reference model. However, to meet the requirements of the lowest levels of the factory automation hierarchy there is no need to use all seven layers of the OSI reference model (N. P. Mahalik & P. R. Moore, 1997).

To illustrate this, let us consider an example of a French company that wants to develop a production line by buying some equipment from a German Company. The two companies are in similar buildings, each consisting of seven floors, each floor linked directly to the floor below and above it, receiving services from the floor below and providing services to the floor above, (see figure 3.4).

The process starts when the French manager (application program) asks the employees on the seventh floor (Application layer) to prepare a list of the required equipment. This means that the seventh floor employees provide a service to the manager. To prepare the list various catalogues can be used. However, the two companies must use the same version of the catalogue (protocols). This means that there is a logical link between the employees on the seventh floor in both companies.

The list, after preparation is transferred to the sixth floor (Presentation layer), where the translation to German can be done. The employees on this floor use rules (protocols), which must be understood in Germany. A statement must be added to the list to specify the rules used. The list is then transferred to the fifth floor.

The employees on this floor (Session layer) use the rules of commercial co-operation between France and Germany to open a session with the German Company. They copy the list, and transfer the list to the fourth floor together with information about the rules.

On the fourth floor (Transport layer) the employees link with fourth floor employees in Germany (say by phone) who may or may not be busy. If the German company is too busy to deal with the whole order, they split the list into parts, and send them part by part to their third floor employees.

On the third floor (Network layer), the route that the list must follow to reach Germany is decided. Any congestion in the chosen route is noted so those alternatives can be considered. The list is then passed to the second floor.

On the second floor (data link layer) the employees collate the order, place it in an envelope and affix the stamps. Then transfer of the envelope to the first floor (Physical layer) takes place where the envelope can be sent to Germany using various transport methods, which are acceptable to both sender and receiver. This means that the two sides must use the same rules (same protocol) for the communication to be successful.

Layer	Activity	Examples
Application	Preparing the order, using catalogues and ordering numbers	Siemens Omron Allen Bradley
Presentation	Translate between the languages using dictionaries	Oxford Collins Websters
Session	Prepare the linking channel use the rules of communication. Make a copy of the order.	National laws European laws
Transport	Split the order into small units. Ensure no losses or duplication. Regulate the flow of units	Rules of splitting. Capacity of carrier
Network	Routing the units. Control the congestion	Maps. Port capacity
Data link	Pack the units. Address the envelopes	Numb. System Hex, Binary…
Physical	Means of transport, speed	Plane, Car,…, Walking

Figure 3.4 OSI example

This example outlines some of the principles involved in communication where the full 7 layer OSI model is needed.

In contrast, the simple situation for communications along industrial networks can be understood if we consider the following example.

Suppose that the French company wants to communicate with another French company on the same street as itself. The employees on the sixth floor are not needed. Similarly, since both the companies are in the same country there is no need for fifth floor employees because the employees on the seventh floor can deal with the local laws. The functions of the fourth floor employees can be carried out by the second floor employees and finally since the companies are in the same street, there is no need to worry about routing. List preparation (by the seventh floor), packaging and addressing (by second floor) and list transportation (by first floor employees) is all that is left. These processes are mandatory.

For industrial fieldbus networks, the conclusions are that:

- The network layer is not needed because the transmission of states associated with sensors and actuators across subnetworks can be avoided since fieldbus networks are situated in the same factory. This will contribute to a cost reduction in network interfaces.
- The transport layer can be eliminated because the data link and application layers can perform its most important functions (e.g., error control, and reliable data transfer with error recovery).
- The session layer is not needed because its basic function (e.g., process to process communication) can be performed by the application layer.
- The presentation layer is not needed because the fieldbus network components must be configured in advance.
- Only the remaining three layers, namely application, data link, and physical layers are required.

3.4.4 Fieldbus standards

Various national projects have already begun to define how the future standard will look. A final agreement has not yet been reached, but nobody wants to wait until a general standard is introduced. Some companies have already defined their products and are marketing them, and projects have been carried out in Europe and America to define national fieldbus standards.

In the next sections the main features of the fieldbuses HART, CAN, Foundation Fieldbus, and PROFIBUS, will be examined together with the AS-Interface networks.

3.5 HART

Highway **A**ddressable **R**emote **T**ransducer (HART) is a communication protocol used in the process sector of industry. It was developed by Rosemount Inc and has been available since the mid-1980s. Now HART Communication Foundation (HCF) which consists of companies providing products that use the HART protocol owns HART, maintains the standard and supports the protocol.

HART is a digital system which preserves the integrity of the 4-20 mA current loop signal (J. Brignell & N. White, 1996)

Before a discussion of the HART protocol can take a place some concepts must be described.

3.5.1 4-20 mA current loop

Current loop is a serial interface. The main reason for choosing current to transmit signals is that current magnitude remains constant along a cable, while voltage values decrease with distance due to the cable resistance.

The current is constrained between 4 and 20 mA, where 4 mA represents zero and 20 mA corresponds to full scale. Using 4 mA as a zero allows failure detection since 0 mA means that the circuit is broken or the transmitter is not sending a signal.

4-20 mA current loop has been used since 1955 (J. Quinn, 1995) as a communication standard for process automation equipment.

3.5.2 Frequency Shift Keying (FSK)

Frequency Shift Keying is a method of changing the frequency of signals to transmit 0's and 1's. Each 0 or 1 is represented by a different number of waves per second (i.e., a different frequency). In this case a certain frequency is defined to be 1, and a different frequency is defined to be 0 as shown in figure 3.5 (HCF 1999).

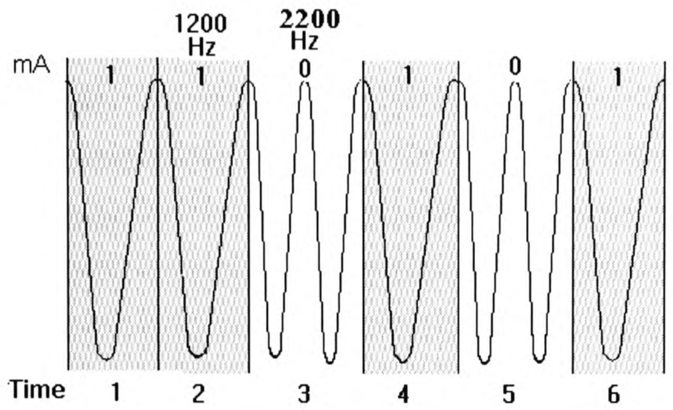


Figure 3.5 Frequency Shift Keying (FSK)

3.2.3 HART protocol

By superimposing a FSK signal on an analogue signal as shown in figure 3.6, HART extends the 4-20 mA standard in order to enhance communication with smart field instruments. This enables two-way field communication to take place. Figure 3.5 shows that logic 1 is represented by 1 cycle of 1200 Hz and logic 0 by two cycles of 2200 Hz which gives a data rate of 1.2 Kb/s (A. McFarlane, 1997).

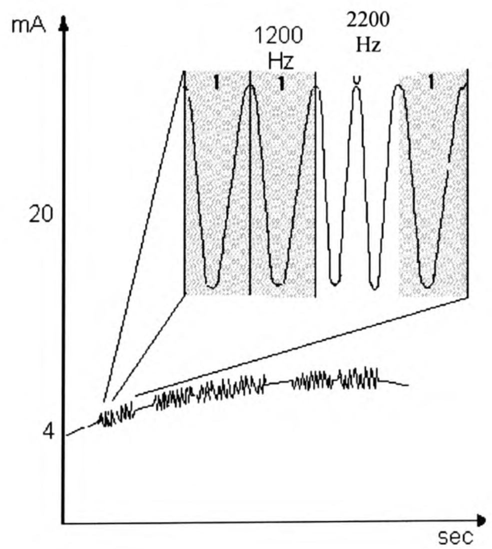


Figure 3.6 HART signals

HART is based on a reduced OSI reference model, which uses the physical layer, data link layer and application layer.

HART specification defines:

- The physical form of transmission.
- The transaction procedure.
- Message structure.
- Data formats.
- A set of commands.

The **Physical layer** defines how the connections are made (e.g. plugs, sockets, and cables) and the type of signal used (voltage, current, and frequency). HART protocol was designed to use 4-20 mA equipment operating with a superimposed modulated signal enabling digital communication to be established between a remote device and a host system. Table 3.5 shows some of the physical layer specifications.

Signal	0.5 V pk-pk
Coding	FSK, 1200/2200 Hz
Speed (bits/sec)	1.2 K
Distance	2000 m
Multidrop	15
Spurs	Yes
Power on the bus	Yes

Table 3.5 HART physical layer specification

The **Data link layer** defines the method of addressing the message, data coding and the rules of sending the message (master - slave, token passing, collision detection). HART protocol is a master/slave protocol which means that the slave device only responds when it receives an instruction from the master via the bus (R. Helson 1998). Each slave device has an unique address and a response occurs only when a slave receives a message containing its address. Each device has a 38-bit address consisting of the manufacturer ID code, device type code, and device-unique identifier. A unique address is encoded at the time of manufacture.

The **Application layer** defines the data type and the standard function of the message meaning.

A **User layer**, which is not a part of the OSI model, is added to the HART protocol to achieve interoperability. It contains the function blocks and the Device Description.

Device Description (DD)

DD includes all of the information needed by a host application to communicate with the field device. DD is written in **Device Description language (DDL)** and combines all the information in a single structured file. It identifies which common practice commands (see table 3.6) are supported as well as the format and the structure of all device-specific commands. HART suppliers have the option of producing DD for their products. If they choose to produce one, DD files are submitted to the HCF for registration in the HCF DD Library and for verification that there are no conflicts with DDs already registered.

Message structure

Figure 3.7 shows the basic message structure

PREAMBLE	START	ADDR	COMM	BCNT	STATUS	DATA	CHK

Figure 3.7 HART message structure

Every message is preceded by specific a number of characters (5-20 bytes) called PREAMBLE, which helps the receiver's signal detection circuit to synchronise to the character stream.

The START character (1 byte) defines the message type (master to slave, slave to master or burst message from slave) and the address format.

The **ADDR**ESS field (1-5 bytes) includes a master address, since in HART protocol there is the possibility of using two masters. These are for example, the controller and a hand-held interface which are known as the primary and secondary masters. A single bit is used to indicate the master address. This is 1 for the primary master and 0 for the secondary master.

The **COMM**AND byte contains the HART command. There are three groups of HART protocol commands, Universal, Common practice, and Device specific commands. Table 3.6 lists some typical commands.

Universal commands provide access to information normally used in plant operation. They must be carried out by all HART devices and provide interoperability of products from different suppliers. The common byte values lies in the range 0 - 30.

Common practice commands provide access to functions that can be carried out by some devices but not all. For these, the common byte values lies in the range 32 to 126.

Device specific commands deal with functions that are used in particular devices and common byte values range from 128 to 253.

Universal commands	Read manufacturer and device type. Read primary variable (PV) and unit. Read current output and percent of range. Read up to four pre-defined dynamic variables. Read or write 8-character tag, 16-character descriptor, and data. Read or write 32 character message. Read transmitter range, unit and damping time constant. Write polling address.
Common practice commands	Write damping time constant Write transmitter range Calibrate (set zero, set span) Set fixed output current. Perform self-test. Perform master reset. Trim PV zero. Trim DAC zero and gain. Write transfer function (square root/ linear). Write sensor serial number.
Device specific commands	Read or write low flow cut-off value. Start, stop or clear totalizer. Choose primary variable (mass flow or density). Read or write material or construction information. Time sensor calibration.

Table 3.6 HART commands

BYTE COUNT (BCNT), occupying 1 byte, indicates the number of bytes in the status and data bytes. Since there is no special 'end of message' character, the receiver uses this byte value to detect when the message is complete.

The **STATUS** field (also known as the response code) is two bytes in length. It only exists in a reply message, where the slave informs the master about communication errors in the outgoing message, the status of the received command, and the status of the device itself.

The **DATA** field may be present depending on the particular message. A maximum length of 25 bytes is recommended, to keep the whole message duration reasonably short.
The **CHECKSUM** byte provide a further check on all the bytes in the message. It contains longitudinal parity information for all previous bytes.

3.5.4 The future of HART

HART has been used widely and successfully for remote configuration, adjustment and diagnostics of smart field devices. However, in the future, as field devices become more sophisticated, holding more information about both themselves and the process that they are controlling, high data rates will become very important. Compared with digital fieldbuses, which operate at much higher communication rates, it seems likely that HART will become redundant. However, this transition will not take place as quickly as once believed (ARC Advisory Group 1999). HART will continue to grow until 2001. After 2001, the installed base of HART will begin showing signs of yielding to Foundation Fieldbus (section 3.7) and Profibus-PA (section 3.8). The transition to these two digital protocols will initially come from new installations and revamps of older instrumentation systems. The actual replacement of HART installations by these two protocols is unlikely to start occurring until at least 2003. This transition will prevail due to the cost advantages of digital communications, improved functionality of process field devices, and users gaining confidence in this new technology.

3.5.5 Summary

HART protocol is primarily used for Process automation. It superimposes a FSK signal on a 4-20 mA analogue signal. Table 3.7 lists the main features of HART protocol.

Layer	Protocol
User	Device Description
Application	Command Set
Data Link	Master-Slave 38-bit address Longitudinal, Exclusive-OR, parity
Physical	Bell 202-telephone communication standard. FSK Max. number of devices =15 Point-to-point, Multidrop. Shielded twisted pair cable. 2000 metre

Table 3.7 HART specification

3.6 CAN

CAN (**C**ontroller **A**rea **N**etwork) is a serial, multimaster communication protocol for use with interconnected control modules, sensors and actuators (K. Dimyati, 1996). It was developed by Robert Boch GmBH for use in time critical applications in the automotive industry (M. G. Rodd; K. Dimyati , and L. Mtus 1998) and has since been standardised in ISO 11898 and ISO 11519 (Schickhuber & Mccarthy, 1997). CAN uses Non Return to Zero (NRZ) format for data transmission. In this method of encoding data the synchronisation can be lost when a large number of identical bits is transmitted. To overcome this CAN employs bit stuffing with the stuff width specified at five bits (J. R. Jordan 1995). This means that after five consecutive identical bits the sender inserts into the bit steam a stuff bit with the complementary value. The receiver removes this bit.

3.6.1 CAN protocol

CAN protocol defines the physical layer and data link layer of the OSI reference model. The Application layer is specific to a system designer.

The scope of the **Physical Layer** involves the actual transfer of bits between the different nodes with respect to all electrical properties (Robert Bosch GmbH, 1991), and within a single network the physical layer has to be the same for all nodes. The Physical layer can be divided into three sublayers; **Physical Signal (PLS)**, which deals with bit encoding / decoding, bit timing and synchronisation, **Physical Medium Attachment (PMA)**, which describes the physical transceiver characteristics, and the **Medium Dependent Interface (MDI)** sublayer which specifies the cable and connector characteristics.

For bit timing, CAN uses a so-called time quantum (tq) which is the smallest discrete timing resolution used by a node. The length of tq is generated by a programmable divide of the CAN node's oscillator frequency. The bit time consists of four time segments, each constructed from an integer multiple of the tq. The number of quanta for each bit is limited to between 8-25 tq (figure 3.8).

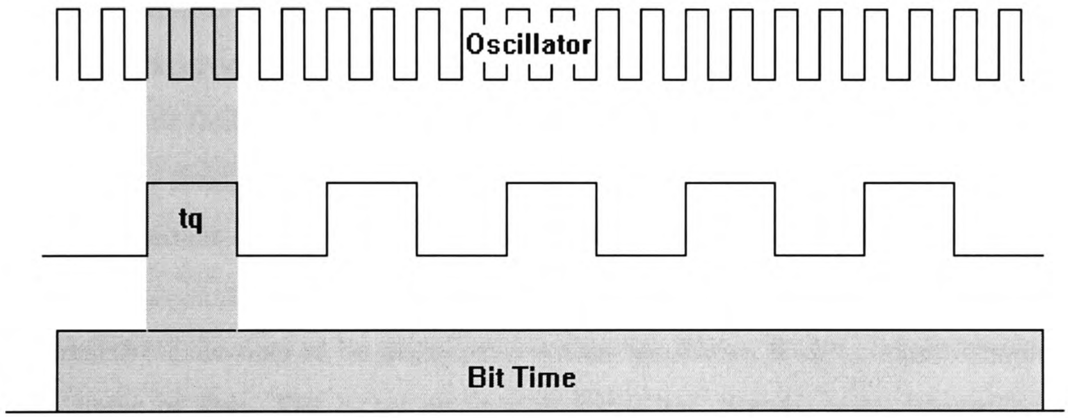


Figure 3.8 Bit timing

The Data Link Layer is divided into two sublayers, the Logic Link Control (LLC) and the Medium Access Control (MAC).

The main functions of the LLC sublayer is to provide a service for data transfer and remote data request, in order to decide which message is to be accepted.

The main scope of the MAC sublayer involves the transfer protocol. It must decide whether the bus is free to start transmission or whether other communications are taking place.

3.6.2 Message format

CAN uses a development of CSMA/CD protocol with non-destructive bit wise arbitration for bus contention (see section 3.6.4). When the bus is free, any connected unit may start to transmit data, which are sent in fixed format messages of various but limited length. There are four different message or frame types, **Data frame**, **Remote frame**, **Error frame** and **Overload frame**, of which Data frame is the most common. CAN protocol supports two types of frame format known as **standard format** and **extended format**.

Standard frame format

The standard frame format consists of seven data fields:

SOF (Start Of Frame) consists of a single dominant bit (0) used to synchronise receiving nodes.

ARB (arbitration field) consists of 12 bits. 11bits are used as an identifier and the RTR (Remote Transmission Request) bit indicates whether it is a data frame (carrying data from a transmitter to receivers) or a remote frame (requesting the transmission of data).

CTRL, the control field, consists of six bits. Four bits are used to determine the number of bytes in the data field, and are called the *Data Length Code*. One identifier extension bit (IDE) indicates either standard format or extended format, and the other bit is reserved for future extension.

DATA consists of the data to be transferred within the frame. It can contain between zero and eight bytes of data. The exact number of bytes has already been determined in the CTRL field.

The Cyclic Redundancy Check (CRC) field is sixteen bits long and is used as a frame security check for detecting bit errors. The first fifteen bits are used to check for errors in data, and one bit (called CRC delimiter) is transmitted recessive.

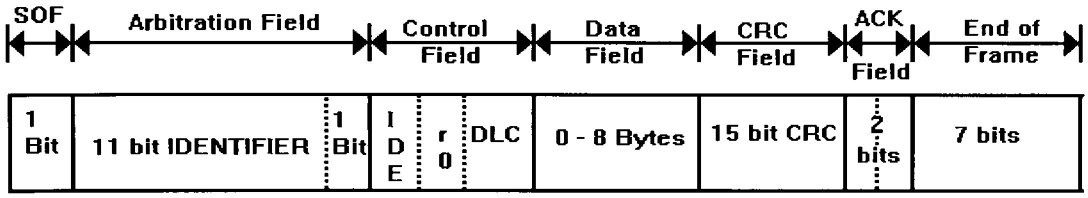
The ACK field is two bits long. The first bit is called ACK-slot and the second ACK-delimiter. The transmitter transmits both bits recessive. When a receiver correctly receives a message, it sends a dominant bit during the ACK slot.

EOF (End Of Frame) indicating the end of the message is a seven bit field.

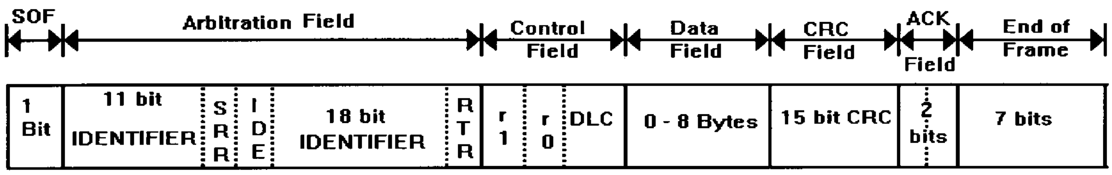
Extended data frame format

The CAN protocol was extended because the implementation of the protocol is easier when a longer identification field is used.

The extended data frame is similar to the standard data frame except that, the ARB field is extended to 32 bits instead of 12 bits. These bits are split into an 11-bit base identifier (ID) which is equivalent to the standard field, and an extended ID of 18 bits. An SRR (Substitute Remote Request) bit and the CTRL field's IDE bit separate the two IDs.
If the IDE bit is transmitted recessive, then this indicates the use of the extended frame.
The SRR bit is a recessive bit transmitted at the position of the RTR bit in a standard frame.
Figure 3.9 shows the standard and extended formats.



A- standard format



B- Extended format

Figure 3.9 Message format of CAN specification.

CAN controllers dealing with messages in extended format can also support messages in standard format. However controllers configured for standard format are not able to communicate using extended format.

Remote frame

Nodes requesting a data frame (figure 3.10) use this feature. It is similar to data frame except for two important differences:

1. RTR bit in the Arbitration field is recessive.
2. There is no data field.

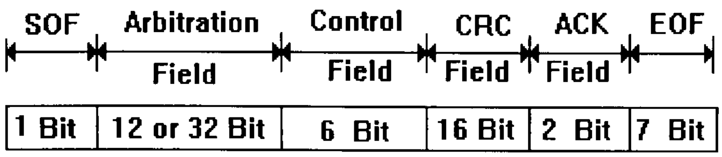


Figure 3.10 CAN remote frame

Error frame

An Error frame is transmitted when a node detects a fault. It consists of an error flag and error delimiter (figure 3.11).

Error flag	Error Delimiter
6 bits	8 bits

Figure 3.11 CAN Error frame

Overload frame

An Overload frame has the same format as an error frame. A busy node transmits this frame to indicate that it is not ready to start reception of the next message.

Data exchange

Instead of giving an address to a station, CAN uses the message identifier which is unique throughout the network to define the content and the priority of the message. The lower the identifier the higher the priority (M.M. Marcos et la., 1999). When a station place a message on the network all the other stations perform an acceptance test to decide whether to accept or ignore the message.

3.6.3 Non-destructive bitwise arbitration.

Any CAN node can start transmission when it detects a quiet bus. The result may be that two or more nodes will start to transmit at the same time. Each transmitting node monitors the bus while it is sending. If a node sends a recessive bit (1) and detects a dominant bit, it will immediately quit the arbitration process and become a receiver. For example, if nodes 1, 2, and 3 start sending at the same time (See figure 3.12) and at bit 3, nodes 2 and 3 send a dominant identifier bit while node 1 sends a recessive identifier bit, node 1 looses the bus arbitration and switches to listen only. At bit 7, node 3 looses the arbitration against node 2 because the message identifier of node 2 has a lower binary value and therefore a higher priority.

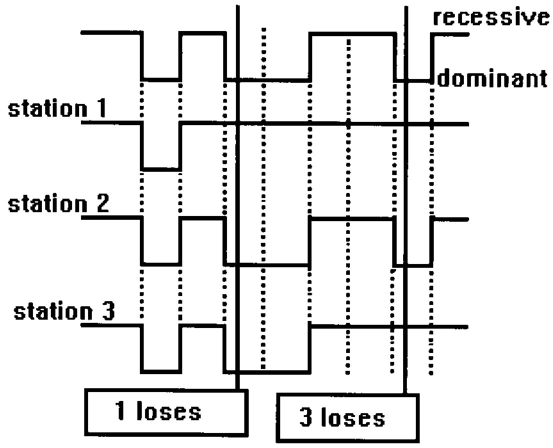


Figure 3.12 Bitwise arbitration

3.6.4 Data-Rate/ Bus length

The rate of data transmission depends on the total overall length of the bus. As mentioned earlier CAN was initially designed for automobiles and trucks so that a long bus was not the primary goal. Generally, data rates are in the range 200 Kbit/s to 1 Mbit/s. For all ISO 11898 compliant devices running at 1 Mbit/s, the maximum possible bus length is specified as 40 metres. To achieve longer bus lengths it is necessary to reduce the bit rate.

Because CAN is not a complete protocol, i.e. system designers may design CAN systems to suit their own requirements, some companies recognised the need for a higher layer protocol. Examples of these are DeviceNet from Honeywell, Smart Distribution System (SDS) from Allen-Bradley and CAN Kingdom from Kvaser.

To give some indication of data rate and bus length, table 3.8 lists values for DeviceNet.

Data rate	bus length
125 Kbit/s	500 m
250 Kbit/s	200 m
500 Kbit/s	100 m

Table 3.8 Data rate / bus length for Device Net

3.6.5 Basic CAN and Full CAN

The terms "Basic CAN" and "Full CAN" originate from the early days of CAN. (These should not to be confused with the terms standard and extended format). The Intel 82526 CAN controller provided an interface to the programmer, which was different to that developed by Philips. To distinguish between the two programming modes, the Intel technique was termed "Full CAN" and the Philips method "Basic CAN". Today, most CAN controllers allow both programming modes, so there is no reason to use the terms "Full CAN" and "Basic CAN"- in fact, these terms can cause confusion and should be avoided.

There are no compatibility problems between the two. A "Full CAN" controller can communicate with a "Basic CAN" controller and vice versa.

3.6.6 Summary

The Controller Area Network (CAN) is a serial communication protocol. It uses only two layers of the OSI model; Data Link Layer and Physical Layer. CAN data link layer is standardised in ISO 11898. Its services are implemented in the Logical Link Control (LLC) and Medium Access Control (MAC). LLC provides acceptance filtering, overload notification and recovery management. The MAC is responsible for data encapsulation (de-capsulation), frame coding (stuffing/de-stuffing), medium access management, error detection, error signalling, acknowledgement and serialisation (de- serialisation). The Physical Layer covers the aspects Physical data transmission between the nodes of the network. Table 3.9 shows the main features of CAN protocol.

Layer	Features
physical	media support: Twisted Pair
Data link	Media access: non-destructive bitwise arbitration. Error detection: CRC, frame check, ack error, monitoring and bit stuffing (NRZ). Message services: broad and multicast.
3 - 6	Explicit to application
application	e.g. SDS, DeviceNet, CAL, CAN-Kingdom

Table 3.9 CAN protocol

3.7 Foundation Fieldbus

The International Electrotechnical Commission (IEC) and Instrument Society of America (ISA) have been working together to provide an international standard for fieldbus. The IEC/ISA SP50 Fieldbus committee has decided to use four layers for a Fieldbus solution, namely Physical, Data Link, Application and User layers. The IEC Fieldbus standard consists of eight parts:

IEC DIS 1158 (pt. 1), Fieldbus - Introductionary Guide
IEC DIS 1158 (pt. 2), Fieldbus - Physical Layer Specification
IEC DIS 1158 (pt. 3), Fieldbus - Data Link Service Definition
IEC DIS 1158 (pt. 4), Fieldbus - Data Link Protocol Specification
IEC DIS 1158 (pt. 5), Fieldbus - Application Service Definition
IEC DIS 1158 (pt. 6), Fieldbus - Application Protocol Definition
IEC DIS 1158 (pt. 7), Fieldbus - Fieldbus Management
IEC DIS 1158 (pt. 8), Fieldbus - Conformance Testing

Unfortunately, movement toward the final standard has been very slow. As a result of this a number of suppliers have attempted to provide their own versions of the standard. In North America WorldFIP and ISP joined forces to become Fieldbus Foundation in 1994 (A.R. Dewey, 1996) to begin using the available parts of the IEC standards. Fieldbus Foundation defines a protocol called **Foundation Fieldbus.**

Where:	Fieldbus Foundation (FF) = Organisation
And	Foundation Fieldbus = Protocol

3.7.1 FOUNDATION Fieldbus protocol

Foundation fieldbus protocol follows the OSI model. It consists of three parts (Fieldbus Foundation, 1998); Physical Layer, Communication Stack and User Application Layer. The Physical layer is OSI layer 1. The Communication Stack is composed of **Data Link Layer** (DLL) *layer 2*, and the application layer which is divided into two sublayers the **Fieldbus Access Sublayer (FAS)** and the **Fieldbus Message Specification (FMS).**

This protocol does not use layers 3, 4, 5, and 6 of the OSI model, since FAS maps the FMS onto DLL. The OSI model does not define the user application layer. (See figure 3.13)

Physical Layer

As defined by IEC and ISA, this layer receives a message from the communication stack and converts it into physical signals on the transmission medium. The signals are encoded by using Manchester technique (see section 2.8.4) with logic 1 denoted by a negative going edge at midcycle and logic 0 by a positive going edge at the midcycle. Initially, FF focused on copper wiring media, although both fibre optics and radio communications can be used.

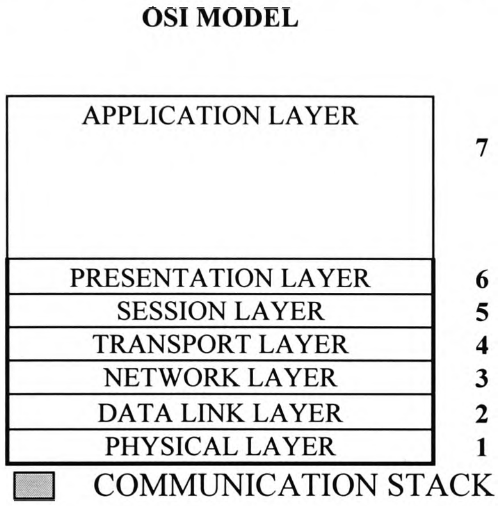


Figure 3.13 Foundation Fieldbus/ OSI module layers

The length of the bus depends on the quality of the cable and the bit rate. 31.25 Kbps (H1 fieldbus), 1 Mbps and 2.5 Mbps (H2 fieldbus) are all possible. FF standard enables 32 devices to be connected onto one bus. The total number of devices will vary depending on factors such as the power consumption of each device, type of cable, and use of repeaters.

H1 fieldbus allows the use of spurs, although the total spur length is limited according to the number of spurs and the number of devices per spur (see table 3.10).

Number of Devices	Maximum Spur Length
25 - 32	1 m
19 - 24	30 m
15 - 18	60 m
13 - 14	90 m
1 - 12	120 m

Table 3.10 Maximum spur length

H2 fieldbus supports only bus topology. Due to the high frequencies used, spurs are not allowed because of the danger that they may cause signal reflections resulting in signal distortion.

Data Link Layer (DLL)

DLL identifies two types of devices known as **basic**, and **link master** devices. The link master contains the **Link Active Scheduler (LAS)** which controls the scheduling of activities on the bus using token bus protocol. Without a LAS the bus will not function (EEMUA, 1997). Since the DLL manages access to the bus through LAS. During the configuration of the system the LAS is supplied with a list of all devices and the times when specific data are needed. Basic devices have no capability to become LAS. Any device configured to use the data is called a '**subscriber**'. When it is time for a device to send a message, the LAS orders the device to broadcast its data to all the devices on the bus. This applies to data that are required at regular intervals. For unscheduled data such as alarm messages, all devices have a chance to send messages between the transmissions of scheduled messages. In configuring LAS the time allowed for scheduled messages must be monitored to ensure enough spare time for unscheduled messages.

Application layer

The application layer is divided into two sections, the Fieldbus Access Sublayer (FAS) and the Fieldbus Message Specification (FMS).

The FAS uses features of the DLL to provide a number of services for the FMS. Virtual Communication Relationships (VCR) describes the types of service, (see table 3.11). There are different types of VCRs:

Client/ Server type

When a device receives the Pass Token (PT) from the LAS it may send a request message to another device. The requester is called the client and the receiver is called the server (figure 3.14). The server responses when it receives the PT.

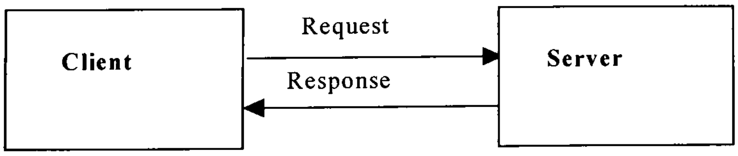


Figure 3.14 Client/ Server service

Report Distribution VCR type.

When a device with an event or a trend report receives the PT, it sends its message to a group of addresses defined for its VCR. Devices that are configured to listen on that VCR will receive the Report.

Publisher/Subscriber VCR.

When a device broadcasts its message to all devices on the bus, those that wish to receive the message are called Subscribers.

Client/Server	Report Distribution	Publisher/ subscriber
Used for operator messages	Used for event notification and trend reports	Used for publishing data
Setpoint changes. Mode changes. Tuning changes. Upload/download. Alarm management. Access display views. Remote diagnostics.	Send process alarm to operator console Send trend reports to data historian	Send transmitter process variable (PV) to PID control block and operator console.

Table 3.11 Virtual Communication Relationships

Fieldbus Message Specification (FMS)

Fieldbus Message Specification describes the communication service that allows user applications to send messages to each other across the bus. It describes the message format, and protocol behaviour. It contains an '**Object Dictionary**', which is a collection of messages. The messages are described by an 'Object Description'.

Each object description is identified by its index in the Object Dictionary. FMS uses a **'Virtual Field Device' VFD** to remotely view local device data described in the object dictionary. A typical device has at least two VFDs. The VFD provides access to address information, device tag, and schedules for function block operation. It also provides access to Virtual Communication Relationship (VCR), dynamic variables, statistics, and LAS if the device is the link master.

User Layer

The user layer, which is not part of the OSI model consists of three separate functions, network management, system management, and user application. Network management supports the configuration of the Link Active Scheduler and provides execution and error monitoring and configuration of other functions in the communication stack.

System management present access to assign device addresses, synchronises the application clock, locate tag devices and schedule function block operations (A. T. Bradshow, 1997).

To achieve interoperability, Foundation Fieldbus uses function blocks and Device Description Language (DDL) technology. **Function blocks** are used for purposes such as control, diagnostics, and safety, and have been developed by representatives from various manufacturers. In this way, they do not merely represent the requirements of single control system suppliers. A device can contain a number of function blocks (see figure3.15) (W. R. Hodson, 1998). For special features, a Device Description (DD) can be defined using Device Description Language. The device has also a **Resource Block** (RB) which contains parameters that relate to the physical device as a whole. Through it the user can observe manufactures ID, type of device and version, and memory size for example.

The **Transducer Block** (TB) decouples function blocks from the local input/output functions required for reading sensors and activating output hardware. TB contains information such as calibration data and sensor type. There is usually one transducer block for each input or output function block. This means that the function block is defined in terms of input and output connections, the parameters in the block, the response to events and the control mode selection. These items are called **Objects**.

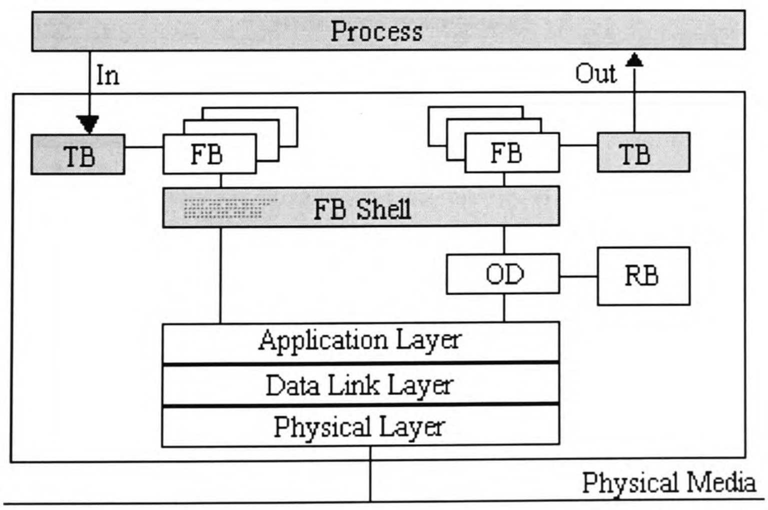


Figure 3.15 Foundation Fieldbus device

3.7.2 Summary

Foundation Fieldbus protocol consists of a Physical Layer, a Data Link Layer, an Application Layer, and a User Layer. The specification provides for standardised function blocks (FB). Which can be used for control and other purposes such as diagnostic, safety, and production accounting. A device can contain any number of function blocks. In addition each device contain resource block (RB), which contain parameters related to the physical device, where the user can characteristics such as manufacturer ID, type of device, and memory size. The transducer block (TB) groups the parameters the associated with the sensor or actuator.

Manufacturers can define their specific parameters by using device description languages (DDL). Table 3.12 lists the main specifications of Foundation Fieldbus technology

Feature	Definition
Technology developer	fieldbus foundation
Year introduced	1995
Guide standard	ISA SP50/IEC TC65
Network topology	Multidrop with bus powered device
Physical media	twisted pair
Max. devices	32
Max. Distance	1900 m@ 31.25 K 500m @ 2.5 M
Communication method	Client / server publisher/subscriber
Transmission speed	31.25 Kbps, 1 Mbps, and 2.5 Mbps
Error checking	16 bit CRC

Table 3.12 Main specifications of FF technology

3.8 PROFIBUS

PROFIBUS (**PRO**cess **FI**eld**BUS**) technology was developed in 1987 by 13 German industrial control and measurement vendors and five technical scientific institutes with part funding from the German Federal Government. It is now a German National standard, DIN 19245 parts 1 and 2 (A. Leach, 1994) and a European standard (European Norm) EN 50 170, which contains the whole of Profibus FMS & DP, WorldFIP, and P-NET. The

Profibus User Organisation (PNO) based in Germany has played a very important role in the growth of interest in Profibus which consists of three compatible versions (figure 3.16). Profibus **F**ieldbus **M**essage **S**pecification **(FMS)**, which is designed for communication at cell level where the application of a high degree of functionality is more important than reaction time, **Profibus D**ecentralised **P**eriphery (**DP**) for high - speed data communication at the device level, and **Profibus PA** for **P**rocess **A**utomation. This section describes the features of each version.

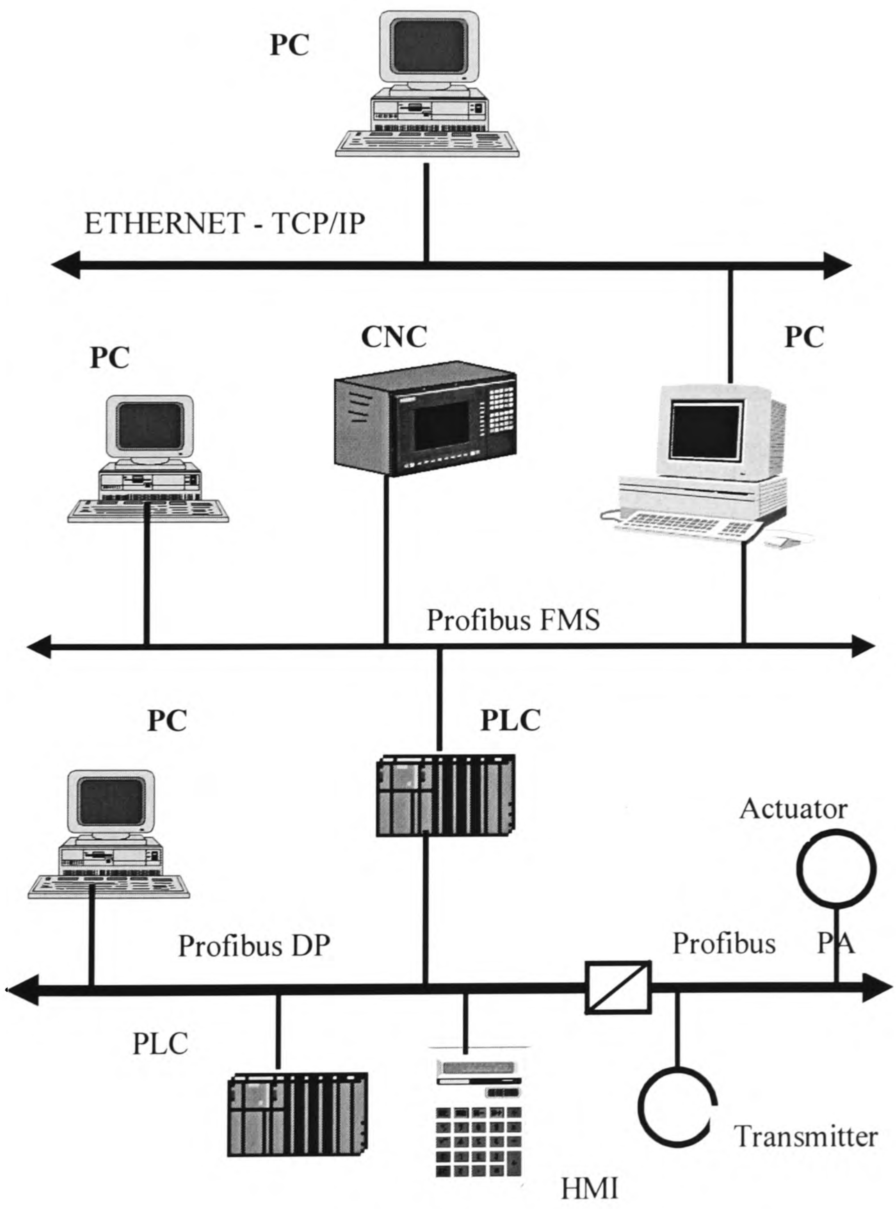


Figure 3.16 Profibus versions

3.8.1 Profibus FMS

Profibus FMS uses layers 1, 2, and 7 of the OSI model. Layers 3 to 6 are not used, since the functions of these layers are combined in the Lower Layer Interface, as shown in table 3.13.

OSI Layers	Profibus FMS layers
7 - Application	Fieldbus Message Specification (FMS)
	Lower Layer Interface (LLI)
3 - 6	Not used
2 - Data Link Layer	Fieldbus Data Link (FDL)
1 - Physical	Physical

Table 3.13 OSI/ Profibus FMS Layers

Profibus FMS implements layer 7 of the OSI model to provide the user with open data transfer between application processes. This layer has two components, **Fieldbus Message Specification (FMS)** and **Lower Layer Interface (LLI)**. The FMS describes communication objects, services and associated models from the point of view of the communication partner. To define the relationship between devices within an application, FMS adopts an object-oriented, client-server architecture.

FMS is designed for the transmission of large amounts of data between higher level devices. Cycle times are typically 60 ms (T.Karppinen, 1998). It is not essential for FMS to meet the time restriction of real time control where the fundamental principle is that the bus cycle time should be shorter than the program cycle time of the controller, typically of the order of 10 ms.

The concept of a **Virtual Field Device (VFD)** is used by FMS to define the aspects of the real device with which it is possible to communicate over the bus, figure 3.17. For communication to take place, LLI maps the functions of the VFD on the real device. These may be defined at the time of manufacturing or, in high level devices, set up over the network configuration.

VFD defines the communication activities. It is based on VFD object, which contains one Object Dictionary **OD**.

A real device may contain a number of VFD objects (figure 3.17), each addressed by its communication end points.

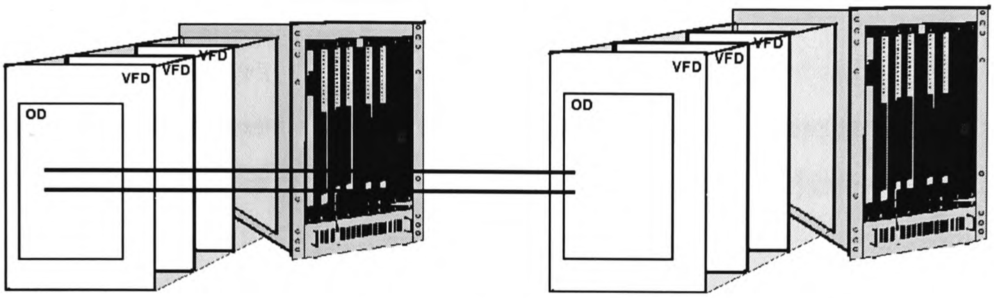


Figure 3.17 Communication between two Profibus FMS devices.

The transferred process objects between two devices, must be known to the communication system. This means that a list of process objects must scheduled as communication objects and entered in the local OD. This can be predefined or configured for the device.

Communication end-points are used by the application process to access the communication. Some communication endpoints are fixed and uniquely assigned to an application process by means of communication references, which are device specific and not defined by the communication itself. One or more communication relationships may be present between two application processes, each one having unique communication end points.

In order to allow an application process which performs the information processing required for a particular application (X. Jin, 1994) to communicate with an application process of another device, communication objects must be available. Particular communication objects (virtual objects) represent existing process objects that an application process has made visible and accessible to the communication. Communication objects can be accessed using the FMS services provided.

FMS Services

Several interactions called service primitives form a service sequence, which describes the execution of the FMS. Service primitives describe the interaction between requester and responder.

Confirmed services are only used for connection-oriented communication relationships. Figure 3.18 shows the execution of a Confirmed service.

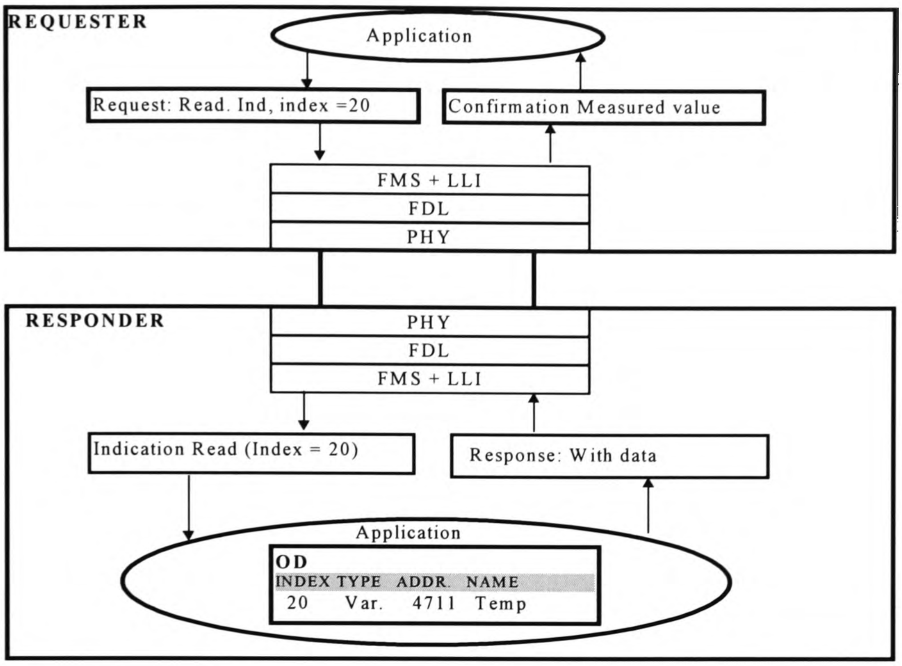


Figure 3.18 Profibus FMS service

Unconfirmed services can also be used on connectionless communication relationships (Broadcast and Multicast). They can be transmitted with high or low priority. An unconfirmed service is requested with a request service primitive. After transmission over the bus, an indication service primitive is issued to the application process of the receivers. Confirmation/response service primitives do not exist for unconfirmed services.

FMS services are divided into the following groups:

Context Management services establishment and release of logical connections and the rejection of nonpermissible services can be achieved using this service.

Variable Access services is used to access simple variables, records, arrays and variable lists.

Domain Management services transmit large memory areas. A Domain Management service is used where the user divides the data to be transferred into segments.

Program Invocation Management services are used for program control.

Event Management services. The transmission of alarm messages and events is carried out by this service. These messages can also be sent as Broadcast or Multicast transmissions.

VFD Support services identification and status polling is accessed by using these services. They can also be sent spontaneously at the request of a device as Multicast or Broadcast transmissions.

OD Management services are used to read and write-access the OD.

PROFIBUS-FMS has a large number of application services, which makes it possible to meet a wide selection of demands made by different devices. Only a few services are necessary. Selection depends on the application and is specified in profiles for specific application areas.

Lower Layer Interface (LLI)

Mapping layer 7 to layer 2 is managed by the LLI.

The LLI provides various types of communication relationships, which using logical channels enables the user to communicates with the other application processes. The communication relationships have different connection capabilities (i.e., monitoring, transmission and demands on the communication partners).

Connection-oriented communication relationships describe a logical peer-to-peer connection between two application processes. Before the connection can be used for data transmission it must first be established with an Initiate service. After being successfully established, the connection is protected against unauthorised access and is available for data transmission. When the connection is no longer needed, it can be disconnected with the Abort service. The LLI permits time-controlled connection monitoring for connection-oriented communication relationships.

The connection attributes "open" and "defined" are another characteristic feature of connection-oriented communication relationships.

In defined connections the communication partner is specified during configuration.

In open connections the communication partner is not specified until the connection establishment phase.

Using unconfirmed services, connectionless communication relationships permit one device to communicate simultaneously with several stations. In Broadcast communication relationships an unconfirmed FMS service is simultaneously sent to all other stations. In

Multicast communication relationships an unconfirmed FMS service is simultaneously sent to a predefined group of stations.

3.8.2 Profibus PA

Profibus PA has been designed especially for the process industries where intrinsic safety is an issue. It can be used to replace 4-20 mA technology with consequent savings of up to 40% in planning, cabling, commissioning and maintenance. PA uses a single certified bus coupler and power supply which results in a saving through the reduction of the documentation required for intrinsically safe certification. In addition, a Profibus PA system provides much greater functionality with the possibility of distributed intelligence carrying out local processing of data and self-diagnostics. For maintenance procedures, PA allows the connection or disconnection of devices during operation without affecting other stations even in hazardous areas.

PA uses the same Fieldbus Data Link (OSI Layer 2) as Profibus DP and FMS, which means that Profibus bus masters see no difference between PA and DP segments. Transmission of power and data takes place over two wires using a fixed rate of 31.25 kbps and Manchester Encoding. The lower transmission rate means that spur and star networks can be used in addition to a linear bus. End termination is required in the form of a simple resistance/capacitor circuit.

According to IEC 1158-2 up to 32 field devices can be connected to a bus segment but this may reduce depending on the power requirements and the degree of protection required. Maximum line lengths are of the order of several hundred metres.

At layer 7, Profibus-PA implement the same function as Profibus-FMS, allowing use of the special device profiles developed within the **I**nteroperable **S**ystem **P**roject (ISP) and based on the concept of function blocks and ISP **D**evice **D**escription **L**anguage (DDL).

3.8.3 Profibus DP

Profibus DP omits layer 7 of the OSI model and maps the user interface directly on to layer 2 using a **Direct Data Link Mapper (DDLM)** as shown in figure 3.19. The main purpose of DP is the fast cyclic exchange of data between powerful masters and a number of simple slaves (up to 126). Thus, this system uses mainly the master - slave type of communication services although token passing can be used between the masters. Also, Master -Master communication is possible by using a DP-DP gateway.

DP Basic Functions DDLM
layers 3 - 7 not used
FDL
PHYSICAL

Figure 3.19 Profibus DP layers

Profibus FMS and DP are based on the same layer 1 & 2, which means that FMS and DP can use the same bus. Also, message header and data length are identical. Some of the information relating to Profibus DP is listed in table 3.14.

		Layer
Transmission Technology	RS-485, twisted pair, two wire cable or fibre optics	PHY layer
Baud rates	from 9.6 kbit/sec to 12 Mbit/sec	PHY layer
Bus access	Token passing procedure between masters and master-slave procedure for slaves. Mono-master or multi-master systems possible Master and slave devices, maximum of 126 stations on one bus	FDL layer
Communication	Peer-to-peer (user data transmission) or Multicast (control commands). Cyclic master-slave user data transmission and acyclic master-master data transmission	FDL layer
Operating modes	Operate: Cyclic transmission of input and output data Clear: Inputs are read, and outputs are held in fail-safe status Stop: Only master-master data transmission is possible	
Synchronisation	Control commands permit synchronisation of the inputs and outputs Sync mode: Outputs are synchronised Freeze mode: Inputs are synchronised	
Functionality	Cyclic user data transmission between DP master and DP slave(s)· Dynamic activation or deactivation of individual DP slaves Check of DP slave configuration Powerful diagnostic functions, 3 hierarchical levels of diagnostic messages Synchronisation of the inputs and/or the outputs Address assignment for the DP slaves over the bus Configuration of the DP master (DPM1) over the bus Maximum of 244 bytes of input and output data per DP slave	
Security and protection functions	All messages are transmitted with Hamming distance HD = 4· Watchdog timer at the DP slave Access protection for the inputs/outputs of the DP slaves Monitoring of user data transmission with configurable monitoring timer at the master	FDL layer
Types of devices	Class-2 DP master (DPM2): programming/configuration/diagnostic devices Class-1 DP master (DPM1): central programmable controllers such as PLCs, PCs, etc DP slave: device with binary or analogue inputs/outputs, drives, valves, etc.	

Table 3.14 Profibus DP features (from Profibus user organis

3.8.4 Summary

As stated, Profibus consists of three different versions, FMS, DP, and PA, which means that it covers a wide range of industrial applications. Moreover the three versions can be integrated to provide full control of a factory. Figure 3.20 shows the integration between PA and DP on the new production line in a German company called Wacker, which produces a silicone oil and glue.

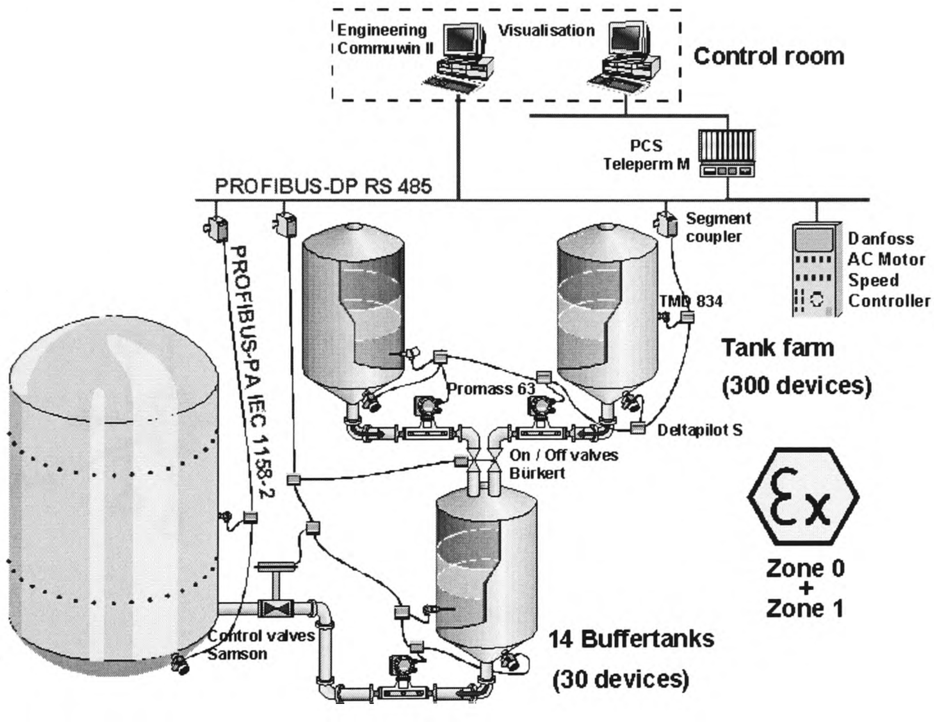


Figure 3.20 Profibus PA/DP implementation

For factory automation Profibus DP is the most widely installed fieldbus and it will remain so according to market studies until at least the year 2003. Since this project deals mainly with factory automation, the next chapter will concentrate on this technology.

3.9 AS-Interface

The AS-interface (Actuator Sensor - Interface) has been developed in 1990 by a consortium of European companies (AS-Interface UK, 1996) as a solution for connecting binary sensors and actuators to equipment at the controller level or field level. Figure 3. 21 (M. T. Hoske, 1998).

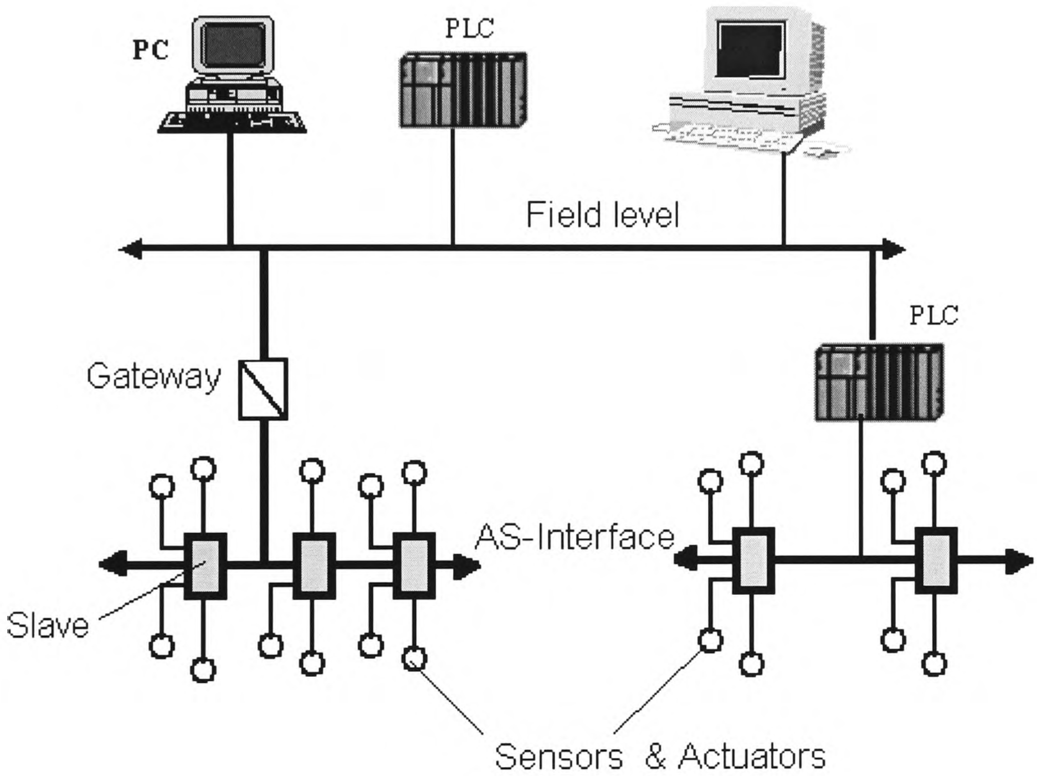


Figure 3.21 Control levels

3.9.1 System components

The basic components of an AS- Interface system are:

The AS - Interface master

The master links the network to the controller (e.g. PLC or PC). It can be a slave of a higher control level (e.g. Profibus).

AS- Interface slave

The slave links the sensors and actuators to the AS- Interface network. The AS-Interface slave chip is either built into a module to which sensors and actuators are connected, or it is built directly into the sensor or actuator itself.

AS- Interface power supply

The power supply feeds the AS-Interface network. If actuators with a high power requirement are used, a separate power supply must be presented.

The AS- Interface cable

The AS- Interface cable is an untwisted, unshielded, two-wire cable. It is used for transferring signals and supplying the sensors and actuators with power.

Other components include addressing devices for setting the slave addresses and repeaters to extend the network length. To ensure that AS-Interface devices of differing origins work together without problems, the AS-Interface Association certifies all devices and allocates the AS-Interface mark of conformity following conformance-testing (Hirschmann, 1997)

3.9.2 The AS - Interface master

AS-Interface is a single master system (Siemens, 1996). Master units are available in the form of a self-contained unit or as a rack-mount card for integration in a PLC or PC. The master device can be connected to the PLC in a similar way to connecting input and output modules, figure 3.22

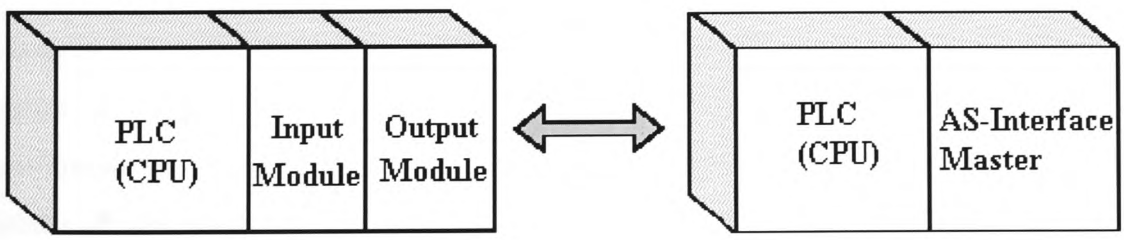


Figure 3.22 Connecting AS-I master to the host PLC

The master specification is classified into four layers, figure 3.23

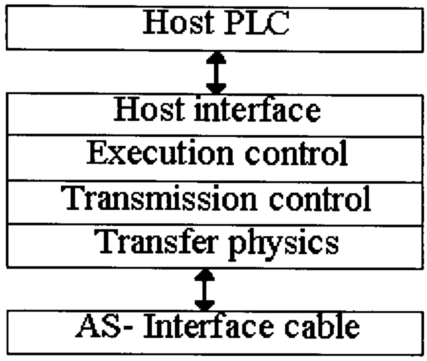


Figure 3.23 AS-Interface Master layers

- Transfer physics layer describes the electrical connection to the cable.
- The transmission control layer is responsible for exchanging individual telegrams with the slaves.
- Execution control is responsible for the initialisation, the start up and the normal data exchange on the network.
- The host interface consists of several functions that deliver the user data, control the network and set the configuration.

3.9.3 AS- Interface Slave

A slave is the connecting link between the cable and attached inputs / outputs. The slave supplies the sensors and the low current actuators with power and carries out the communications between them and the master. At the point of connection, the cable provides an operating voltage of between 26.5 and 31.6 volts D.C.

AS-Interface uses “modules” to connect conventional sensors and actuators to the network. Up to 31 slaves, each with its own address can be connected to one AS-Interface network, and a slave, as stated in section 3.9.2, can be one of two types:

1) Module type which allows the connection of standard 24 V dc sensors and actuators. Using these modules, up to 248 sensors and / or actuators can be connected to the network.

2) Dedicated type, which is more flexible than the standard modules. The sensors and actuators of this type can handle up to 4 input and output bits. Some of these slaves have changeable parameters and fault signals, each with its own address

Figure 3.24 shows schematically a 2I/2O module. It contains an AS-Interface IC which receives switching signals from (in this case) 2 sensors and sends switching signals to 2 actuators.

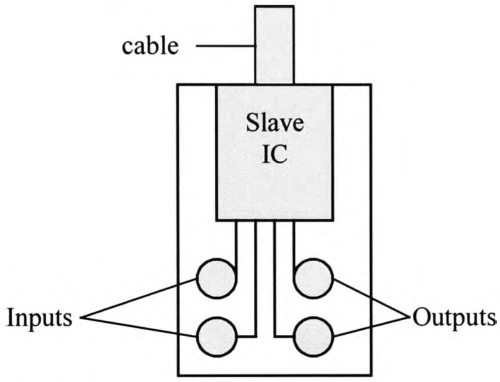


Figure 3.24 AS-Interface slave

3.9.4 Power supply

Since AS-Interface uses the same cable to exchange data and to provide power to the slaves, it is important to use a special power supply with data de-coupling.

The power supply provides a voltage of 29.5 … 31.6 volt D.C. to the slaves. Under normal operating conditions a current of up to 7 A can be drawn. The power supply has to be short circuit and over load protected and must be current limited.

3.9.5 AS Interface cable

The AS-Interface components are linked using AS-Interface cable. It consists of two wires with 2 x 1.5 mm^2 cross section.

Two different unshielded, two wire cables have been defined as the transmission medium for AS-Interface. The first is flexible power cable according to CEN 90, which is inexpensive and widely available. The second type is the flexible, mechanically coded, yellow cable, which is now used for standard cabling of all AS-Interface stations, Figure 3.25. The yellow cable is specially designed to assist faster commissioning.

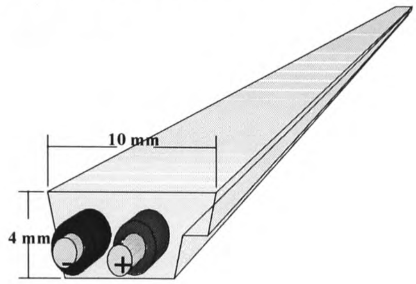


Figure 3.25 AS-Interface cable

AS-Interface can have different topologies (figure 2.26) (AS-International Association, 1998). This allows adjustment of the wiring to the needs of the machine or plant. The slaves do not have to be connected in line or in a ring and the cables do not need terminating resistors.

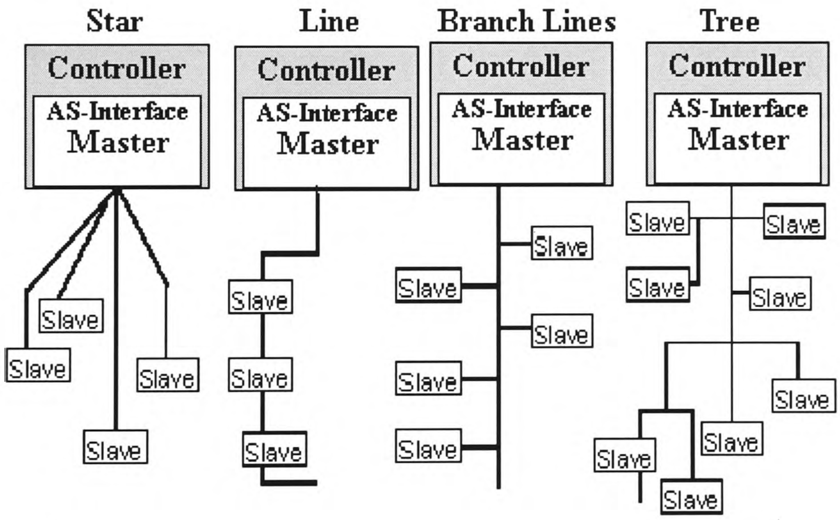


Figure 3.26 the AS-I topology

The maximum cable length is 100 m, including all pieces. Networks with more than 100 m can be built with repeaters. Up to 2 repeaters can be used in series to produce a line of 300 metre. Several parallel branches may use repeaters. Due to the possibility of a tree-structure and the use of individual addresses, new slaves or a complete new branch may be introduced into the system at any time.

3.9.6 Data exchange

The AS-Interface system contains one master and up to 31 slaves (A. McFarlane, 1997). In order for data exchange with the master to take place, every slave must be assigned a unique address between 1 and 31. Addressing the slaves can be done by using a function block of the PLC or by using the addressing unit. The address is stored in non-volatile memory of the slave which means that it remains stored even when power is removed. The addresses of the slaves can be changed at any time but it must be ensured that no two slaves on the network have the same address at the same time.

When the master powers up it interrogates all possible addresses and waits for answers from the slaves. If a reply is received the master records the slave in an internal table.

With the use of additional software the user can provide the master with a configuration list, which is used as a reference. The master compares this with its generated list to detect if there are errors such as incorrect addresses and reports these to the controller.

Communications between master and slaves run cyclically (see figure 3.27):

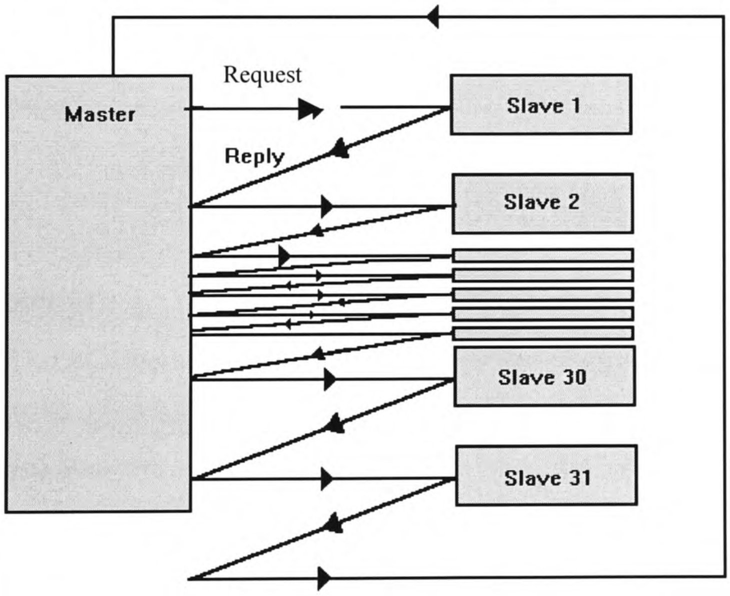


Figure 3.27 Data exchange.

First, the master sends a message (request) to a certain slave. The called slave answers immediately.

After a short pause the master calls up the slave with the next address. After slave 31 the cycle is complete and starts again at slave 1. Unregistered addresses are skipped. The master pause can occupy a minimum of 3 bit and a maximum of 10 bit intervals. If the slave is synchronised it can send a response after 3 bit Intervals. If it is not synchronised it needs a further 2 bit intervals, to monitor the master pause for possible further information before it finally accepts the request as valid. However, if the master has not received the start bit of the slave response after 10 bit intervals (Time out condition) it will assume that there will be no reply and it starts the next request. The slave pause should only be between one and two bit intervals long. A bit time unit corresponds to 6 μs.

The master sends messages to all the available addresses one after another with the message structure shown in figure 3.28. The slave message structure is shown in figure 3.29.

master interrogation														master break
0	SB	A4	A3	A2	A1	A0	I4	I3	I2	I1	I0	PB	1	
ST													EB	

Figure 3.28 Master message format

ST Start bit, always 0

SB Control bit 0 Data/ Parameter / address request

1 Command request

A4 ... A0 Address of interrogated slave (5 bits)

I4 Information bit

0 Data request

1 Parameter request

I3 ... I0 Data / Parameter bits (4 bits)

PB Parity bit

EB End bit, always 1

slave Reply							Slave break
0	I3	I2	I1	I0	PB	1	
SB						EB	

Figure 3.29 Slave message format

SB Start bit, Always 0

I3 ... I0 Data / Parameter bits

PB Parity bit

EB End bit, always 1

The time for one full cycle of master/slave communication is 5 ms for a network with 31 slaves including repeat calls which happen if the system detects an incorrect telegram.

The master organises the date transfer through firmware which means that there is no programming or configuring by the user. To the user, the master looks very much like a traditional I/O card handling input and output process images.

AS-Interface transmits 4 data bits for inputs and 4 for outputs to each slave in each cycle. In addition it is able to send 4 parameter bits to one slave in each cycle. The user can utilise these parameter bits, for example, to set remote values in intelligent slaves.

Analogue modules can also be connected to an AS-Interface network. In this case special software in the control unit has to be used. The use of analogue signals is limited to relatively slow varying quantities like temperature or pressure.

Normally, information and energy are transmitted on the same (yellow) cable. For circuits with emergency shut-off, or if a slave needs more power, an additional power supply with a separate cable (black for DC or red for AC) can be used. Both use the same simple connection technique.

Most systems need to know the true positions of the machine’s components (detected by sensors) in order to allow restart of the machine. With an additional power supply, only the main power to the actuators has to be switched off in an emergency via the black cable. The network itself and the sensor power supply need not be switched off.

3.10 Summary

Five different fieldbuses have been discussed in this chapter, HART, CAN, FF, Profibus, and AS-Interface.

Highway **A**ddressable **R**emote **T**ransducer **(HART)** was initially developed by Fisher-Rosemount to be used in **process automation**. It is now managed by a separate organisation, the HART Communication Foundation (HCF) and is particularly popular in the USA. Commercially available products include a range of slave sensors and both PLC and PC masters.

HART enables groups of up to 15 sensors and actuators to be jointed together in series as part of a 4 - 20 mA current loop. A bus master may then individually poll the sensors. 1200 bits/s FSK signalling at frequencies of 1200 Hz and 2200 Hz provides a data communication throughput over ranges of up to 2000 m, dependent on cable type.

Using the HART protocol, the analogue signal is still used to transmit the process value while a digital value is superimposed on top of the analogue value in order to exchange additional information with the device.

The HART signalling levels are chosen to be deliberately low. In a single sensor application, this enables the sensor to output data in both analogue and digital form. Where multiple sensors are employed, the data transfer must be purely digital. The signalling system is essentially the American Bell 202 standard (J. Quinn, 1995). Consequently, silicon support is available in the form of the modem and UART combinations.

The **Controller Area Network (CAN)** is a serial data communications bus for real-time applications. It was originally developed in the mid 1980s by the German company Robert Bosch for use in the car industry to provide a cost-effective communications bus for in-car electronics and as an alternative to expensive and cumbersome wiring looms. CAN is now an international standard and is documented in ISO 11898 and ISO 11519. At present, CAN is being used in many automation and **manufacturing** applications.

CAN supports broadcast and multicast transmissions using a content-oriented addressing scheme. A unique message identifier (ID) defines the type and priority of the message.

There are two types of identifiers available, standard format (11 bit) and extended format (29 bit ID). The maximum size of a data packet can be 8 bytes. The CAN bus system uses

CAN proprietary transceivers for TP communication. With CAN, the network management is very simple with each node reading all transmissions and filtering relevant messages depending on an Identifier. CAN is also based on the publisher-subscriber approach, but it uses smaller messages aimed at digital devices rather than analogue quantities as with the Foundation standards.

CAN defines only the Physical and Data Link layers. Many applications of CAN require services that are beyond the basic functionality specified by the Data-Link Layer but which may be implemented at the Application Layer. For example, the transmission or reception of data units longer than eight bytes. To meet this need several organisations have developed Application Layers. Such as CAL (CAN Application Layer), CAN open, PCAL (Portable CAN Application Layer), DeviceNet, SDS (Smart Distributed System), and CAN Kingdom

CAN in Automation (CiA), the international users and manufacturers group founded in March 1992, provides technical, product and marketing information with the aim of fostering CAN's image and providing a path for future developments of the CAN protocol.

These numbers of application layers make the search for a unique solution for fieldbus networks more difficulty, since the users have not only to decide which fieldbus is better but in this case which application of CAN fieldbus is better. At the moment Device Net seems to be the most popular solution.

Foundation Fieldbus is used in both **process** (H1 version) and **manufacturing** (H2 version) automation applications. The technology is controlled by the Fieldbus Foundation. It allows 32 devices to communicate over distance of 1900 metre at speed of 31.25 Kbps

The **Foundation Fieldbus** seems to be developing into an American project appropriated by the main DCS vendors who have never shown much aptitude for producing the common standards that would be useful to industry.

Foundation Fieldbus uses the publisher-subscriber approach where each parameter is broadcast at regular intervals onto the bus by the publisher device. The subscriber device then monitors the bus and copies this information into their database. In this protocol the messages consist of objects, where each message can contain 16.6 M objects.

Process fieldbus (Profibus) is the best-known fieldbus solution. This is due, in part, to the backing of Siemens, one of the major players in the PLC marketplace. Profibus spans a range of applications in industrial control.

Profibus FMS is targeted at higher level communication between cells within a plant. It is designed to operate at medium speed (cycle time is typically 60 ms) and provide a wide range of functionality and flexibility (see section 3.8.2). Profibus FMS (Multi-master/Peer to Peer). This protocol enables up to 126 devices to communicate with each other. It is used for transferring large amounts of data such as program downloading.

Profibus DP is a performance - optimised version, specifically dedicated to time - critical communications between automation systems and distributed peripherals. It is targeted at lower level tasks in **manufacturing automation**. Both DP and FMS protocols can operate over the same transmission medium. Profibus DP is a Master/Slave, which means that the master periodically requests the status of each node. This ensures that each device on the network (which can send up to 244 bytes of data per scan) is updated consistently and reliably, up to 126 devices can be linked to one network and the speed of data transfer is between 9.6 Kbps and 12 Mbps.

Profibus PA is the newest variant of the Profibus family. It is aimed specifically at the **process automation** market. Profibus PA which is intrinsically safe, is commonly found in Process Control. Transmission of power and data takes place over two wires using a fixed rate of 31.25 kbps and Manchester Encoding.

Profibus is the most widely accepted international networking standard, being nearly universal in Europe and also popular in North America, South America, and parts of Africa and Asia. Profibus can handle large amounts of data at high speed and serve the needs of large installations. The DP, FMS and PA versions collectively address the majority of automation applications. On the other hand Profibus has a high overhead to message ratio for small amounts of data, no power on the bus and a slightly higher cost than some other buses.

Profibus is the world's most widely installed open Fieldbus. Its substantial speed, distance and data handling capabilities make it ideal for many process control and data intensive applications.

Actuator Sensor **Interface (AS-Interface)** was developed by a consortium of European companies and is now promulgated by a separately established organisation, AS-International. It is targeted at the very lowest level of the industrial network hierarchy. Specifically, it is optimised for binary sensors and actuators in automation networks.

It may be used as a local I/O bus in a system managed by one of the higher-level fieldbuses. A key feature of the bus is that nodes connected to it receive power (24V) and data over the same pair of unshielded wires. The bus is most often associated with its characteristic yellow rectangular section cable with polarising bump (although a circular section cable is now available). Nodes may be attached to and removed from the cable as required.

AS-Interface is specified for single master operation only. The master polls up to 31 slave devices spread over a distance of up to 100m (300 m with repeaters). Each slave device may have up to four sensors and four actuators associated with it. With this number of slaves, each slave is polled every 5 ms. Faster scan rates are achieved with fewer slaves.

Comparing the five fieldbuses except AS-I, which is targeted on a lower level application, Profibus provides a complete solution for all industrial applications through the use of Profibus PA and Profibus DP which can be linked together. Also, Profibus DP is the fastest fieldbus at the present time. Table 3.15 lists the main features of the fieldbuses discussed in this chapter.

Protocol	HART	CAN	Foundation Fieldbus	Profibus	AS-Interface
Typical application	Process	manufacturing	Process	Manufacturing Process	manufacturing
Technology Developer	Rosemount	Bosh	Fieldbus Foundation	PNO	AS-I trade organisation
Organisation	HCF	CiA	FF	PNO	AS-I trade organisation
Year Introduced	mid 1980s	1980s	1995	FMS, DP1987 PA 1995	1993
Standard	IEC 1158	ISO 11898 ISO 11519	ISA SP 5O IEC TC 65	IEC 50170 IEC 1158	Submitted to IEC
Network Topology	Bus, tree	Bus	Multidrop	Star, ring, bus	Bus, tree, ring, star
Physical medium	Shielded TP	TP	TP & fiber	TP & fiber	Two wire cable
Max. devices	15	64	32	126 DP, 32 PA	31
Max. distance	2000 m	500 m @ 125 kbps	1900 @ 31.25 kbps	1200 @ 9.6 kbps	100 m
Communication method	Master/slave	Broad and multicast	Client/server	Master/ slave	Master/ slave
Transmission speed	1.2 kbps	125, 250, 500 kbps	31.25 kbps 1, 2.5 Mbits/s	12Mbits/s DP. 31.25 kbps PA	167 kbps
Data size	25 byte	8 byte	16.6 M object/ device	244 byte	8 bits
Error checking	Longitudinal Exclusive-OR, parity	CRC	16 bit CRC	HD 4	HD 2

Table 3.15 Fieldbuses main features

3.11 Conclusions

This chapter discusses the subject of fieldbus networks. Compared with higher level networks, fieldbus networks have the following features (see figure 3.30)

- Response times are shorter.
- The number of devices is greater.
- The data package length is smaller.
- The number of protocols is larger.

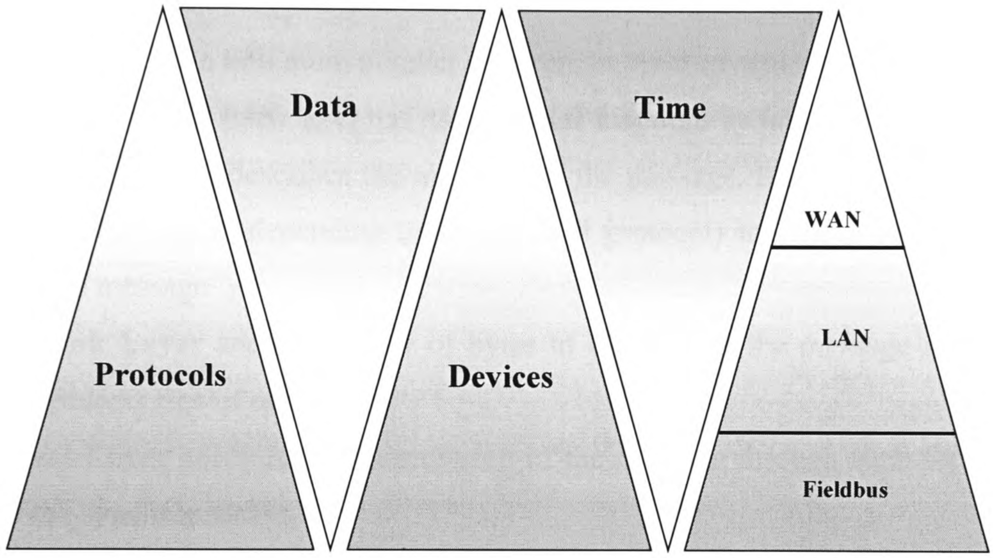


Figure 3.30 Fieldbus features

Fieldbus technology is based on the OSI reference model and MAP protocol (figure 3.31). Most fieldbus networks use the physical layer, data link layer, and application layer, which are described in the OSI reference model, and a user layer which originates in the idea of Manufacturing Message Specification (MMS) developed by MAP to meet the requirement of industrial applications. This leads to the conclusion that using 7 layers is not always necessary and that communication systems can work with a reduced number of layers.

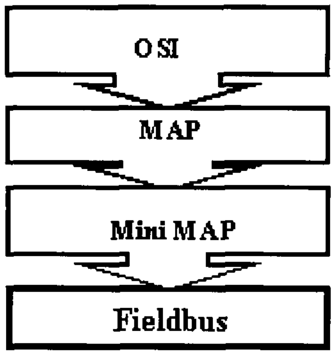


Figure 3.31 Fieldbus development

Fieldbus protocols use a minimum number of layers in order to achieve a fast response time as shown in figure 3.32. Each layer has its particular functions as follows:

The **Application Layer** describes the meaning of the message. It adds to the main data a number of bits or bytes (depending upon the used protocol) to enable other partners to understand the message.

The **Data Link Layer** adds a number of bytes to ensure that the message arrives at the destination address free of errors.

The **Physical Layer** controls the transmission of the message through such factors as the speed and the synchronisation.

At the destination, each layer removes the addition at its own level in the following sequence:

The Physical Layer checks the received message to ensure that the baud rate, coding, current and voltage are correct. Then it removes the additional bits belonging to this part and passes the message to the next layer (DLL)

The Data Link Layer receives the message, and checks the address and for the presence of any errors. If it detects any errors it does not allow the message to pass to the next upper layer and reports the error to the system.

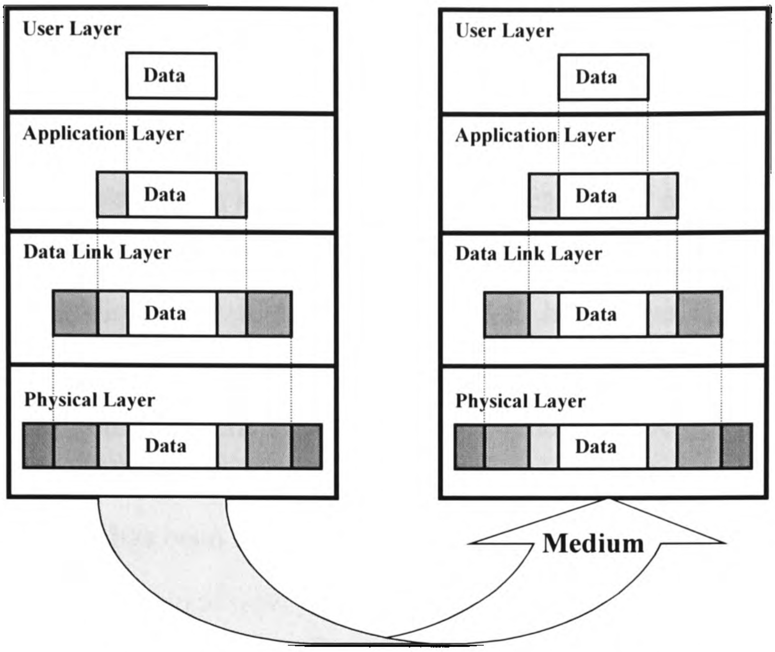


Figure 3.32 Fieldbus message propagation

The Application Layer receives the message from the DDL, removes the additional bytes and provides the user with data alone.

Some protocols such as Profibus DP omit layer 7 of the OSI model and map the user interface directly onto layer 2 (DLL). A number of profiles are defined to be used in the user layer so that the message is directly pointed to specific action.

The different protocols discussed in this chapter are targeted at two different applications. These are process automation and manufacturing (or discrete) automation. HART, FF, and Profibus PA are aimed at process automation while Profibus DP and CAN are aimed at discrete automation.

HART utilises a 4-20 mA signal and has been in use for some time. This protocol attempts to marry digital and analogue technologies. At the present time many companies still support HART, although Foundation Fieldbus (H1) and Profibus PA are tending to push HART out especially since the successes of implementing Foundation Fieldbus and Profibus PA in different applications have been demonstrated.

It is important to mention that gateways are available to link HART and Profibus DP networks together. However from the point of view of a user, it is probably better to use Profibus PA and Profibus DP since both are supported by the same organisation.

CAN which was designed for use in automobiles, has been widely adopted by industry. The main characteristic of CAN is that it only defines the 1^{st} and 2^{nd} layers of the OSI model and leaves the application layer to the user. This has led to the appearance of various systems based on CAN, the most well known system being DeviceNet, which is not as popular in Europe as in the US. The same is true for Foundation Fieldbus.

The aim of fieldbus foundation was to produce H1 for process automation and H2 for discrete automation. H1 has been implemented in various applications while H2 has not yet been issued.

Profibus PA corresponds closely to Foundation Fieldbus H1 except for the protocol implementation, and the resemblance is not coincidental since both originated in the work of the Interoperable System Project (ISP).

The H2 functions of Foundation Fieldbus are performed by Profibus FMS or by Profibus DP (P. Stude Baker, 1997) which are already available on the market and described in EN 50170.

In other words, Profibus with its three versions (FMS, DP, and PA) provides a complete solution for all industrial networking applications. However, the growth of industrial Ethernet on one hand and Profibus DP on the other hand is reducing the importance of Profibus FMS, figure 3.33.

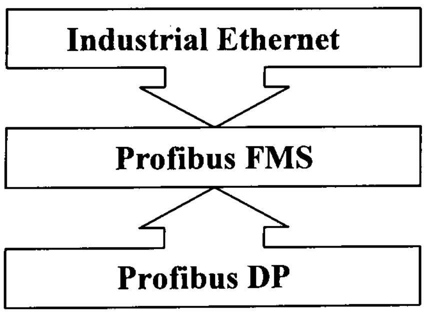


Figure 3.33 The future of Profibus FMS

Compared to Profibus, CAN is not a complete protocol. It always needs an application layer, which may be different in different systems (see section 3.10).
Fred Holger the automation technology president in Bosch, the company which developed CAN, said recently "Profibus is a key part of our long term strategy. It meets user's needs today and is well proven. We will continue to make, supply and support Profibus products".
In conclusion, all the signs indicate that to choose Profibus at the moment would be a good decision. Profibus is supported by PNO with 1000 members and 3,000,000 nodes and 300,000 applications have been installed world wide.

CHAPTER 4

UWCN Profibus DP Training Network

4.1 Introduction

In order for a company to achieve success in implementing new technology such as fieldbus systems, a number of issues must be addressed. One of these is the essential need to train engineers and technicians on how to deal with the new technology. Since training programmes are very expensive, this has the effect of increasing the cost of projects. However, seeking to minimise the amount of money to be spent on training programs below a certain threshold is likely to affect the efficiency of the engineers and technicians who are required to run the system.

The network described in this chapter has been established at University of Wales College Newport (UWCN). The Profibus DP Network is located in the Siemens S7 laboratory on the Allt-Yr-Yn Campus, in order to provide training programs on Profibus DP systems for industry. In offering reasonably priced courses for local companies the hope is that companies will be more willing to adopt the new technology and hence remain competitive in the global market.

This chapter describes the network in terms of practical applications, illustrates how to implement a Profibus DP system, and represents the main features of this type of fieldbus.

4.2 Network components

The network consists of a number of units as shown in figure 4.1. Each unit has a specific function (Siemens, 1998) on the network. In order to understand the operation of the network the following section describes the components and the function of each unit. Devices on the network can be one of the following:

Class 1 master (DPM1)

This is a central controller able to exchange data with connected I/O devices. A Class 1 Master (DPM1) handles the network token and determines the baudrate (up to 12 MBaud).

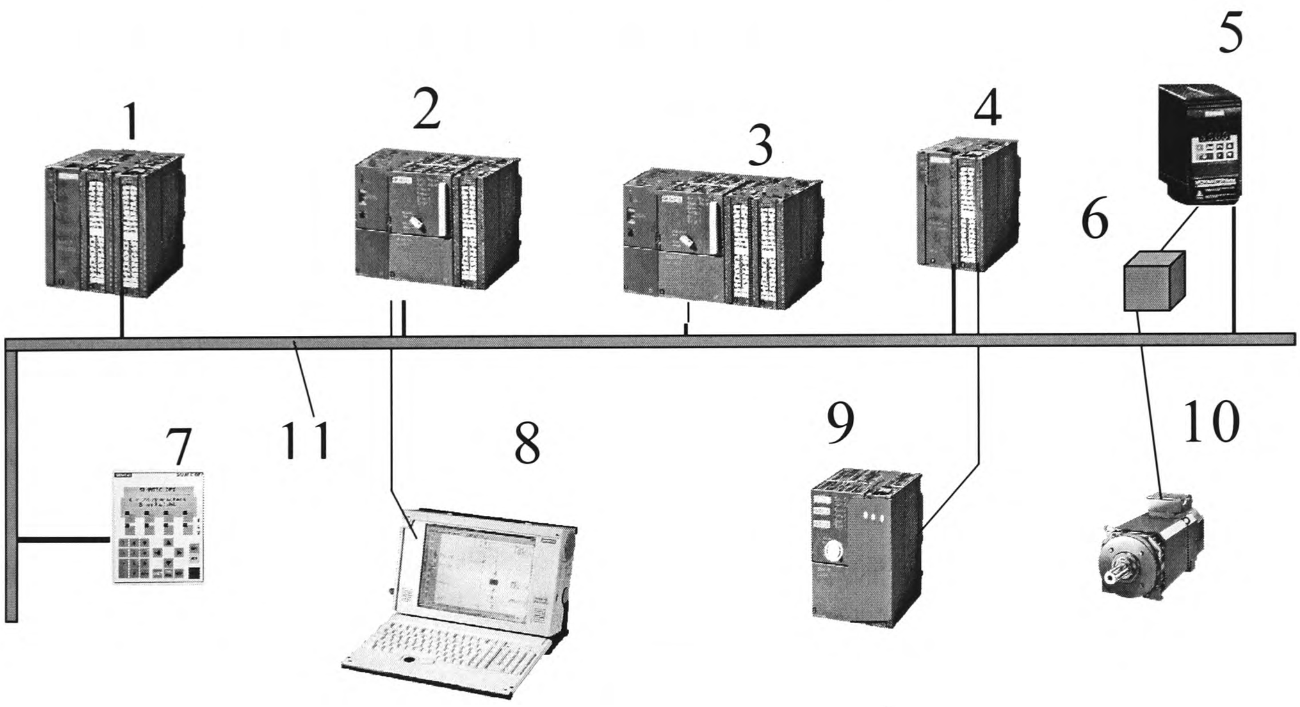
Typical DPM1 devices are PLCs and PCs which are known as Active Stations.

Class 2 Master (DPM2)

DPM2 are used as programming tools, configuration tools, and start-up and diagnostic tools, able to control one slave at a time.

Slave station

These devices also called *passive stations,* merely acknowledge messages or provide answers on request. Slaves only respond to a request from a master and therefore have no right to control the bus. Master devices alone can control the network (for a limited amount of time (*token hold time*) in the case of a multi master system).

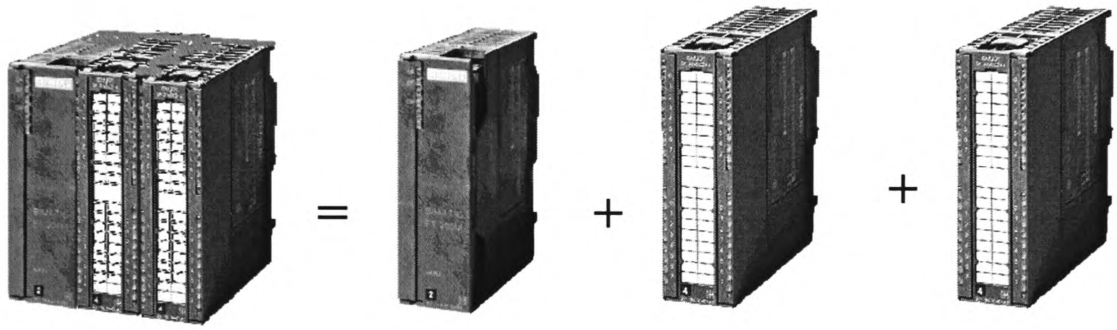


1) Remote Digital I/O
2) PLC S7 300 (Master)
3) PLC S7 300 (Slave)
4) Remote simulator
5) Micromaster Drive
6) Connection box
7) Operator panel (OP 7)
8) Programmer
9) Analogue power supply
10) Motor
11) Profibus Cable

Figure 4.1 UWCN Profibus DP network

4.2.1 Remote Digital I/O

In traditional manufacturing plants, input and output devices were connected to a centralised rack, leading to extensive wiring with high cabling costs and reduced flexibility for modifications and expansions. Linking the components via an open standardised fieldbus system provides a solution to these problems. In plants where the I/O modules can be installed in remote locations near the sensors and actuators, then these can be connected to the controller over a 2 core single cable by using a suitable interface, figure 4.2
In this case, the Siemens IM 153 serves as the interface module of the ET 200M I/O station, which connects the station to the Profibus DP network.



Unit 1 = Interface module + Digital Input module + Digital Output module

Figure 4.2 Unit 1 of UWCN Profibus DP network

Interface module. UWCN uses the ET 200M I/O station, which contains the IM 153 interface module. Up to 8 I/O modules from the S7-300 programmable controller's range can be connected to the interface.

The interface module and the required I/O modules are mounted on an S7-300 DIN rail. Bus connectors (Figure 4.3) are used to connect the I/O modules to each other and to the IM 153 interface module.

Various types of I/O modules can be used depending on system requirements.

The ET 200M is a passive station (slave) on the network. The IM 153 interface module provides the communications capability between the modular ET 200M I/O station and the higher-level master connected to the PROFIBUS-DP. This module needs a power supply of

24 V DC. Input and output addresses are assigned to the respective master during configuration.

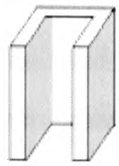


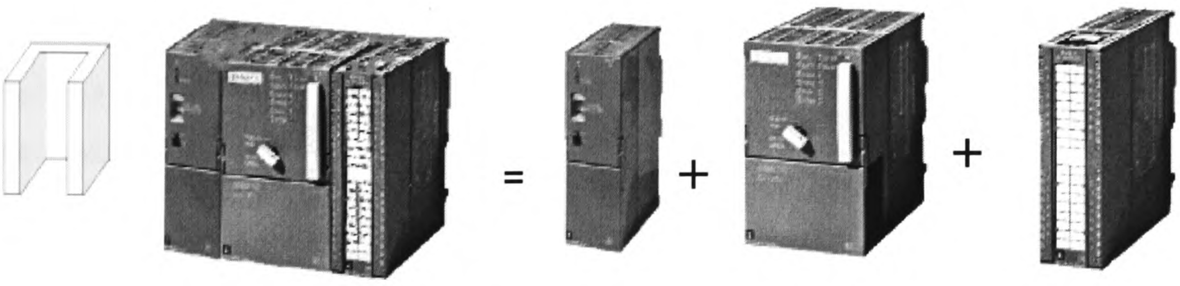
Figure 4.3. Bus connector

A Digital Input module converts the levels of the external digital signals transmitted from the process to the internal S7-300 signal levels. The modules are suitable for the connection of switches and 2-wire proximity switches.

A Digital Output Module converts the internal signal levels of the S7-300 to the external signal levels required for the process. This module is suitable for the direct connection of low current equipment such as solenoid valves, contactors, small motors, lamps and motor starters.

4.2.2 PLC S7 300 (Master)

Unit 2 of the UWCN network consists, of a power supply module, CPU unit and digital input module, as shown in figure 4.4.



Unit 2 = Power Supply + CPU + Simulator

Figure 4.4 Unit 2 of UWCN Profibus DP network

Power Supply Module

The S7-300 requires a 24 V DC power supply. A PS 307 load power supply module converts 230 V AC line voltage to the unit's 24 V DC operating voltage. It makes it possible to use line power to run both the SIMATIC S7-300 and the process sensors and low power actuators. The connection to the CPU is established using a power connector.

The **CPU** 315 - 2 DP *that is used in this network*, has an integrated connection for PROFIBUS DP which means that it can be used as a DP master or slave. The CPU 315-2 DP with PROFIBUS-DP master/slave interface enables distributed automation configurations with high speed and ease of configuration to be realised. From the user's point of view, distributed I/O is treated like centralised I/O (same configuring, addressing and programming).

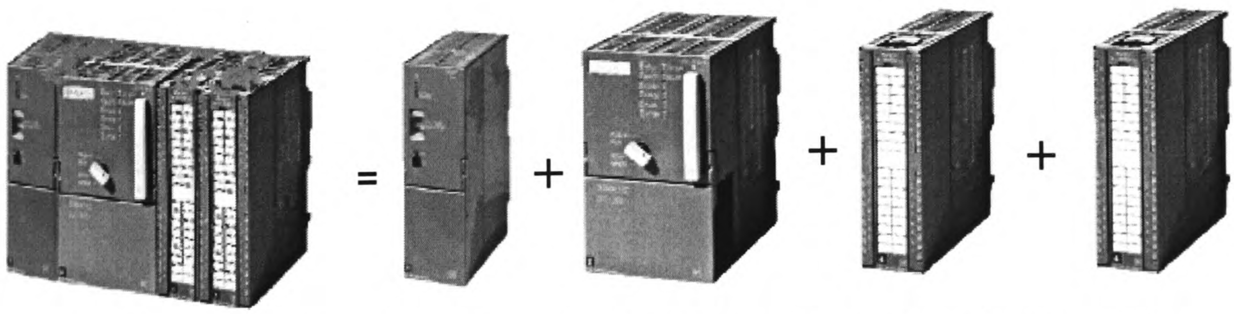
Simulator

The SM 374 **simulator** module provides a convenient means of testing programs during start-up and operation. It contains 16 switches, which allow simulation of input signals, and 16 LEDs indicate the signal status at the outputs.

The simulator module is plugged into the S7-300 in place of a digital input or output module. This allows control of program execution by setting the inputs. The CPU reads the input signal states set on the simulation module and processes them in the application program.

4.2.3 Slave PLC S7300

Unit 3 of the network consists of a power supply module, CPU linked to a simulator and an analogue input module, as shown in figure 4.5.



Unit 3 = Power Supply module + CPU + Simulator + Analogue Input module

Figure 4.5 Unit 3 of UWCN Profibus DP network

An Analogue input module converts analogue signals from the process to digital signals for internal processing within the S7-300. Voltage and current sensors, thermocouples, and resistance thermometers can be connected to the module.

4.2.4 Remote simulator

This unit consists of an interface module (see section 4.2.1) and Simulator (see section 4.2.2).

4.2.5 Micromaster drive

The Micromaster drive is used to control the speed of the motor (**Unit 10**) via the connection box (**Unit 6**). Unit 5 of UWCN Profibus network consists of a micromaster drive and a Profibus module (CB 15), which is a device allowing control of the drive over the network. CB 15 is connected to the front of the micromaster drive where it is powered directly from the drive. Transmission of the Information between the network master (Unit 2) and the micromaster drive take place through CB 15.

4.2.6 Operator Panel

The operator panel (OP) can be used for varying tasks and demands associated with machine operation and monitoring. It can be connected to the Profibus DP network. The operator panel is optimised for use with PROFIBUS-DP at data rates up to 1.5 Mbaud.

4.2.7 Programming Device

The network uses a PG 740 as a programming device. The PG 740 can be used for all SIMATIC automation systems in both online and offline operation. In this case it uses STEP 7 programming language to satisfy the requirements for hardware configuration, programming, testing, commissioning, as well as maintenance and servicing.

4.2.8 Analogue Power supply

This unit is a laboratory power supply used to provide an analogue signal in the range (0 to +10 V). This signal is connected to the analogue input module (unit 3) and is the set-point for motor speed.

4.2.9 Transmission medium

The Transmission medium consists of Bus **Cable** and Bus **Connectors**.

The PROFIBUS standard defines two variations of the bus cable for PROFIBUS - DP. Type A and type B

Specification of Type A Cable

Impedance:	135 up to 165 Ohm at a frequency of 3 to 20 MHz.
Cable capacity:	< 30 pF per Meter
Core diameter:	> 0,34 mm^2, corresponds to AWG 22
Cable type:	twisted pair cable. 1x2 or 2x2 or 1x4 lines
Resistance:	< 110 Ohm per km
Signal attenuation:	max. 9 dB over total length of line section
Shielding:	Copper shielding braid or shielding braid and shielding foil

Cable specification Type B

Impedance:	135 up to 165 Ohm at a frequency of > 100kHz
Cable capacity:	typ. < 60 pF per Meter
Core diameter:	> 0,22 mm^2, corresponds to AWG 24
Cable type:	twisted pair cable. 1x2 or 2x2 or 1x4 lines
Signal attenuation:	max. 9 dB over total length of line section
Shielding:	Copper shielding braid or shielding braid and shielding foil

Type A is especially recommended for high transmission speeds (> 500 kBaud) and permits doubling of the network distance in comparison to Type B. Type B should only be used at low baud rates and low requirements on the network distances. Type A is used for the network under discussion with the cable being connected to the network stations via bus connectors.

Bus connectors

Bus connectors are plugged directly into device PROFIBUS interfaces (9-pin sub D female connector). The incoming and outgoing cable is connected via four terminals in the connector. There are two types of bus connectors used in this network, (figure 4.6) the technical data of each of them are illustrated in the table 4.1.

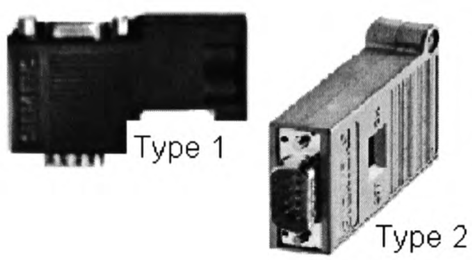


Figure 4.6 Bus connectors

Features	Type1	Type 2
Cable outlet	Vertical	Axial
Data transfer rate	9.6 kbit/s -- 12 Mbit/s	9.6 kbit/s -- 12 Mbit/s
Dimension (mm)	15.8 x 54 x 34	15 x 57 x 39
Weight	40 g	100 g
Use in PLC	Yes	No
Use with ET 200 M	Yes	No
Use in programming device	No	yes

Table 4.1 Bus connector characteristics

The terminating resistor integrated in the bus connector can be switched in using a slide switch. It is important to switch on the terminating resistors at both ends of the network to prevent signal reflections. This means that the segments are terminated at both ends figure 4.7, where the power to the termination is provided by the devices, figure 4.8. The termination switch must be closed whenever only one cable is connected to a device.

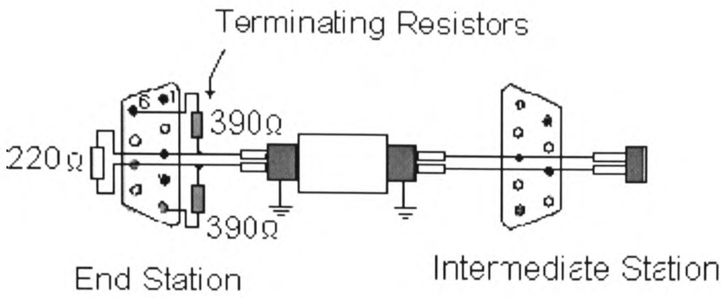


Figure 4.7 Terminal connection for bus connector

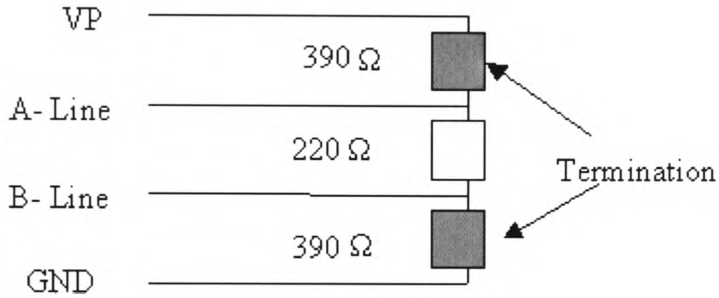


Figure 4.8 Termination Resistance

4.3 Network Assembly

The stations of the network are connected by the 2 core shielded copper cable via 9 pin Sub-D connectors.

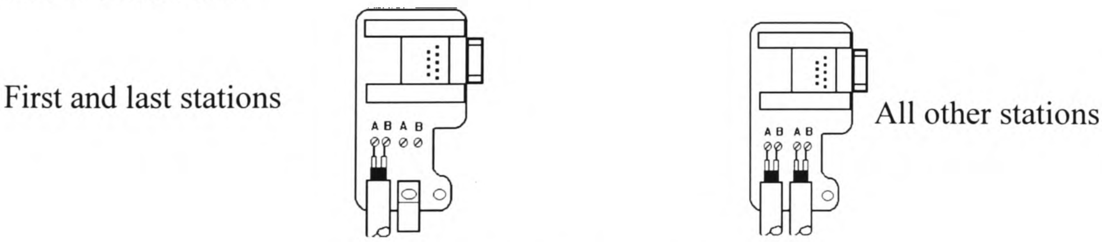


Figure 4.9 Bus Connectors

The two-wire cable is usually colour coded (the red wire for signal B and the green for signal A. The shield is connected to the housing figure 4.9.

4.4 Network operation

4.4.1 Device Description File (GSD)

After connecting all the stations, the network needs to be configured so that the master is aware of each station. Using **STEP 7** software (SIMATIC Manager) which provides a library or hardware catalogue of most of the devices that can be used in Profibus networks the configuration can be carried out. In Profibus DP, each slave or Master class1 device needs to have a **device description file** (**GSD**) figure 4.10. This file is created by the device vendor and describes the characteristics of the device. The description includes:

Vendor's name.

Device's name (this will be displayed in the configuration tool).

Identification number (each slave and master class1 needs to have a unique identification number).

Protocol identifier (Profibus DP or FMS)

Station type (master or slave)

Supported Baudrates

```
GSD_Revision=1
Vendor_Name="SIEMENS
Model_Name="S7-315-2DP-AF03
Revision="V1.4
Ident_Number=0x802F
Protocol_Ident=0
Station_Type= 1
Hardware_Release="A1.0
Software_Release="Z1.0
9.6_supp=1
19.2_supp=1
45.45_supp=1
93.75_supp=1
187.5_supp=1
500_supp=1
1.5M_supp=1
3M_supp=1
6M_supp=1
```

Figure 4.10 Part of the S7-315-2 DP GSD file

The configuration of the network has to start by choosing the master class1 of the network (in this case unit 2), The user must be sure that the device can be used as a master. A statement in the GSD file provides this as follows:

Station_ Type=0 Slave

Station_ Type=1 Master

4.4.2 Device Addressing

Using the Siemens software package known as SIMATIC Manager software, which is already installed on the programmer (unit 8), the user defines the master of the network to be CPU 315-2 DP with an appropriate Profibus address. Up to 126 devices can be connected to the network. A transmission rate of 1.5 Mbps is used in the network. The relationship between the segment length and the transmission rate is shown in table 4.2.

transmission rate	Max. segment length in metres
9.6 Kbps	1200
19.2 Kbps	1200
93.75 Kbps	1200
187.5 Kbps	1000
0.5 Mbps	400
1.5 Mbps	200
3 Mbps	100
6 Mbps	100
12 Mbps	100

Table 4.2 Segment length / transmission rate relationship

For baudrates greater than 1.5 Mbps spur lines are not allowed. The minimum distance between two stations is limited to 1 metre in the case of 12 Mbps. Most of the other stations are defined in the same way by using the hardware catalogue and defining the transmission rate. The user should define which stations are slaves and which are masters and should provide an address for each one (in some cases by using the **DIP switches** on the front of the device). After downloading the network configuration to the master, the master should recognise all the slaves as being present on the network.

4.4.3 Data Transmission

UWCN's network is a mono-master system, with only one master active on the bus during the operating phase of the network. Profibus uses the Master-Slave procedure for communication between master and slave.

The transmission of data between the master and the slaves is executed automatically by the master in a defined order. During the configuration, the user specifies the assignment of the slave to the master.

Data transmission between the master and the slaves is divided into three phases, Parameterisation, Configuration and Data transfer, figure 4.11.

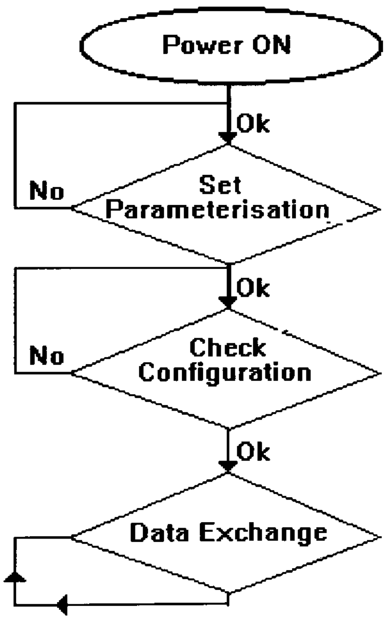


Figure 4.11 Data transmission phases

In parameterisation and configuration the slaves compare their real configurations and the configurations held at the master. Only if the actual and the real configurations match, will the slave be included in the user data transfer phase. Therefore, device type, device address, format and length information as well as the number of inputs and outputs must correspond to the real configuration. These tests provide the user with reliable protection against parameterisation errors. In addition to the user data transfer which is executed automatically by the master, new parameterisation data can be sent to the slaves.

Data exchange starts by sending a telegram from the master to the slave, (figure 4.12). The telegram consists of:

1. **Header**, which contains 11 bytes of information such as destination address and source address.
2. **Output data**. Up to 244 bytes can be send. Profibus offers four transmission services in cyclical mode:

SDN Send Data with No acknowledgement is used to broadcast from the master to all the slaves and acknowledgement is not appropriate.

SDA Send Data with Acknowledge.

RDR Request Data with Reply.

SRD Send and Request Data, where the slave responds in the same bus cycle as it receives a message from the master.

3. **Trailer** contains the end delimiter and frame check sequence.

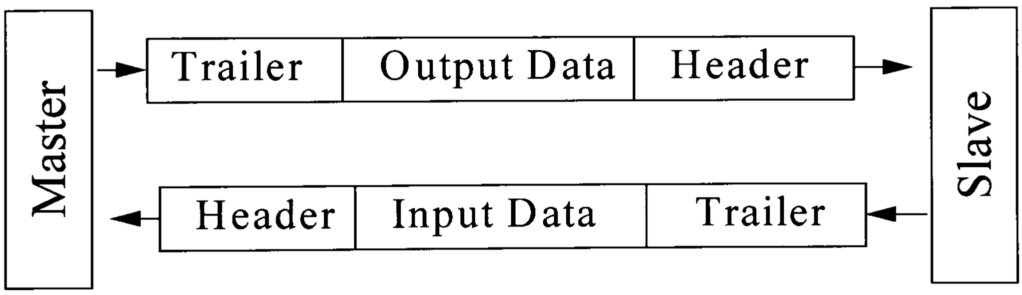


Figure 4.12 Profibus DP telegram

4.4.4 Master / Slave communication

Using the programming unit the user can write a program to pass data along the network. As the network uses the PLC of unit 3 as a slave, the CPU of this unit must be configured as a DP slave with defined address (4 in this case). When the CPU is configured as a slave it makes available an intermediate memory area to be used for the exchange of data between the slave itself and the master, figure 4.13. The user must configure (using STEP 7, figure 4.14) this area so slave and master can read or write to it.

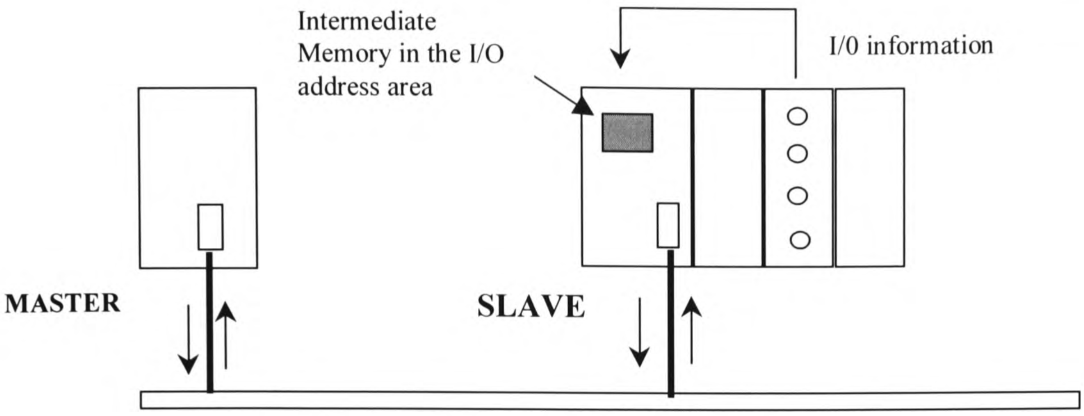


Figure 4.13 Master Slave communication

The most important aspect in configuring the intermediate memory area is that the input data for the slave is always output data for the master and the output data for the slave is always the input data for the master.

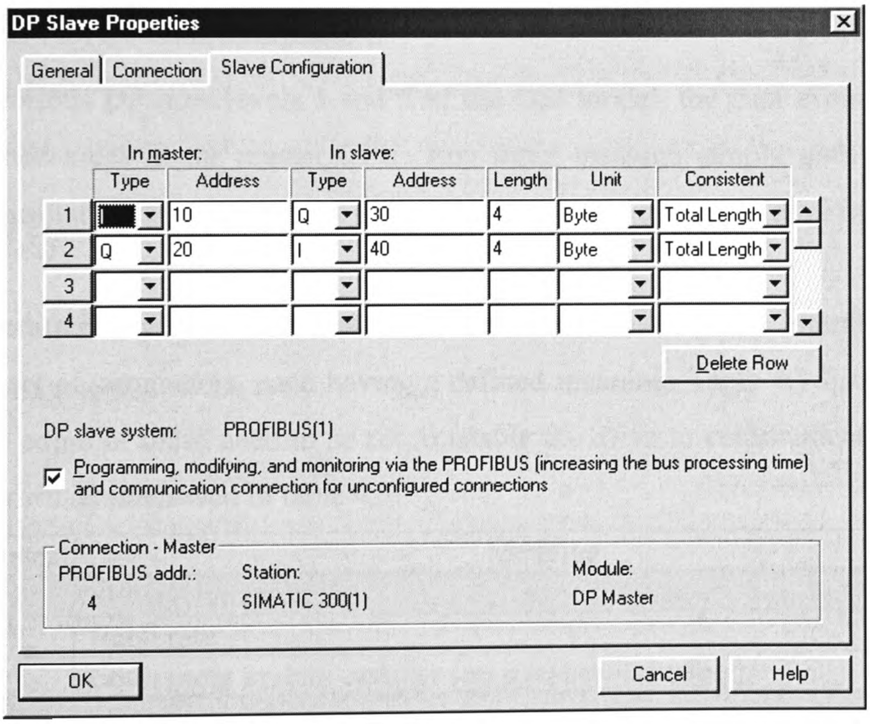


Figure 4.14 Slave configuration

For example, if the slave writes any value into its output Byte 30 (QB 30), the Master is able to read it from its input byte 10 (IB10), Figure 4.15.

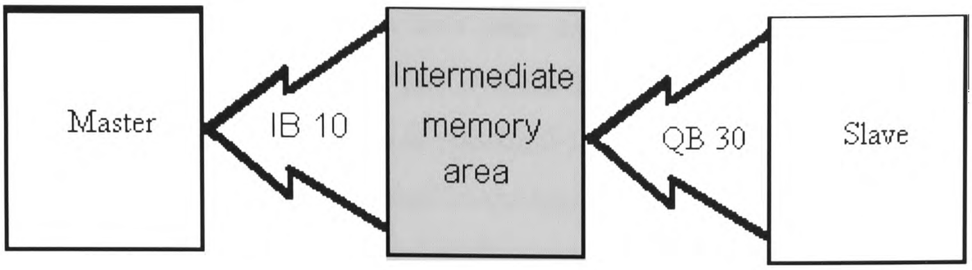


Figure 4.15 Master - slave communication

Example: The input signal I0.0 that is connected to the slave should give us an output signal on Q0.0 of the master.

To solve this task the user should move the Input byte 0 (which contains the input signal I0.0) of the slave into the intermediate memory area as output byte 30 so the master can read it as input byte 10. The state of I 10.0 is now available to control Q0.0.

4.4.5 Master / Drive Communication.

Because Profibus DP uses levels 1 and 2 of the OSI model, the data exchange maps onto predetermined areas in the master CPU. Any input message simply goes to the mapped area. In a similar manner, messages to the slaves are automatically read from the mapped area.

The micromaster drive communicates with the master via a CB15 interface module and provides a set of parameters, each having a defined meaning. Table 4.3 lists some of these parameters. some of these need to be set to enable the drive to communicate on a Profibus network shown as shadowed in table 4.3.

Parameter	Meaning
P918	Slave Address
P963	Baud rate
P701	Equipment system number (*no operational effect*)
P099	Optional model type
P928	local/ remote mode
P958	Warning code
P967	*shows the latest received control word*

Table 4.3 Micromaster driver parameters

The PROFIBUS-DP protocol defines how user data are to be transmitted between the stations over the bus. User data are not evaluated by the PROFIBUS-DP transmission protocol. The significance of the data is specified in the profiles. In addition, the profiles specify how PROFIBUS-DP is to be used in the application area. Users have the advantage of being able to exchange devices from different vendors. The profiles also significantly reduce engineering costs for the user since the meaning of application-related parameters is specified precisely. Utilisation of the profiles permits individual components from different vendors to be exchanged without the plant operator noticing a difference. One of these profiles which is used for the drives is illustrated in the next section.

4.4.6 Variable-Speed Drive Profile (3.072)

The PROFIDRIVE profile specifies how the drives are to be parameterised and how the set-points and actual values are to be transmitted. It has been defined by the leading manufacturers of drive technology. This idea permits drives from different vendors to be exchanged. The profile contains required specifications for speed control and positioning. It specifies the basic drive functions while leaving sufficient freedom for application-specific expansions and additional expansion. The profile describes the mapping of the application functions for DP or FMS.

The profile of Profibus variable speed drives defines five parameter process data objects PPO1 to PPO5. The drive that is used in UWCN's network only supports two of them, namely **PPO1** for process and parameter data areas and **PPO3** for process data area. These objects describe the structure of data in the message format. (Figure 4.16)

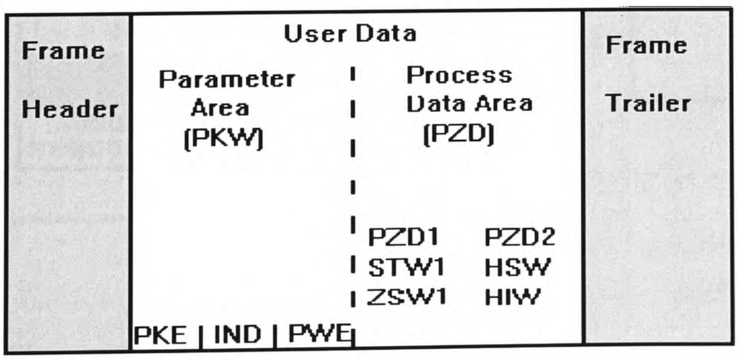


Figure 4.16 Message frame

PKW	Parameter identifier value	task or reply + parameter number
PZD	Process data	control + set-point (Master - drive)
		Status + actual value (drive- master)
PKE	Parameter identifier	
IND	Index	
PWE	Parameter value	
STW1	Control word 1	on/off, condition of operation, rotate direction
ZSW1	Status word 1	such as fault/no fault, ready to start
HSW	Main setpoint	the value 16384 corresponding to 100%
HIW	Main actual value	the actual frequency

The PPO type is defined in the master of the network during set-up. For this network, type PPO3 is used, which is a two word message format allowing the master to send control words and setpoints to the drive and allowing the drive to send status and actual values to the master. The addresses for writing to the drive and reading from it must be defined during network setup. In this case, input words 50 to 53 and output words 50 to 53 are used, (Figure 4.17).

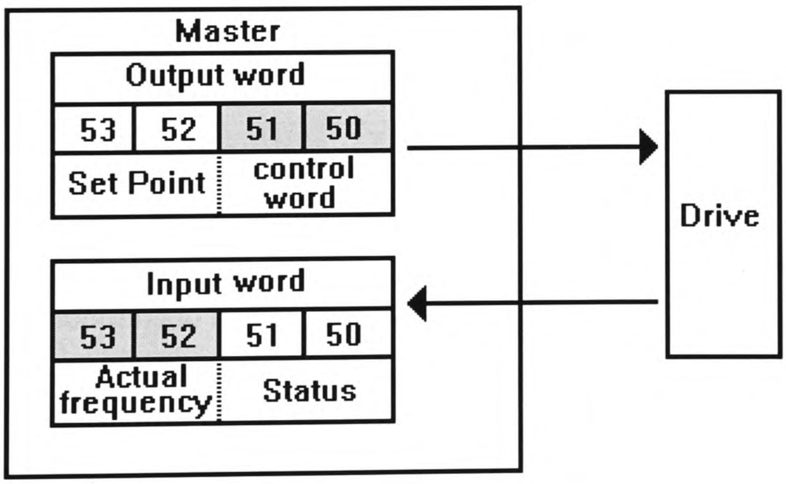


Figure 4.17 Drive addresses

4.4.7 Connecting the OP 7 to the network.

To connect the OP 7 to the network it first has to be configured using the Siemens ProTool Lite software Package, where parameters such as the Profibus address and the baud rate can be specified. On the screen of the OP7 there are two fields. In this network the upper field is used as an input to the slave PLC (in this case it is configured to write to Marker Byte 0 (MB0)) and the output from the slave is displayed in the lower field where the user can monitor the motor speed. The OP 7 takes the value from Marker Word 6 of the slave PLC (MW6).

4.4.8 Master / Analogue module Communication

Industrial applications often require the generation of analogue output signals as well as the ability to read analogue inputs. To meet these requirements, PLCs need analogue input and output modules.

The address of an analogue input or output channel is always a word address. The analogue address depends on the module start address. If the first analogue module is plugged into slot 4 it has the default start address 256, table 4.4. The start addresses of each further analogue module increases by 16 per slot.

Rack	Slot Number										
	1	2	3	4	5	6	7	8	9	10	11
0	PS	CPU	IM	256	272	288	304	320	336	352	368

Table 4.4 PLC Slots

The power supply (Unit 9) is connected to the analogue input module which is in slot 5 of the PLC (unit 3). According to table 4.12, the default address of this module is 272.

The value of this analogue signal controls the speed of the motor (unit 10). Since the raw value of this signal is not compatible with the set point that should be used to control the speed of the motor it requires some arithmetic operations using STEP 7 instructions which can be performed in the slave PLC. The sequence of motor speed control over the Profibus network is as follows:

1. The signal value from the power supply is read into the analogue input module of the slave PLC.

2. The slave PLC calculates the appropriate value and sends the result to the master.
3. The master uses the result to control the motor via the CB15 driver interface.

4.5 Summary

In this chapter the UWCN Profibus DP network has been discussed. Devices on the network can be masters class 1 or 2, and slaves. Profibus DP uses only layer 1 and 2 of the OSI model. For layer 1 it uses RS 485 technology where 32 devices can be linked together by using a two wire shielded cable. Devices are connected to the cable via bus connectors with only the first and last nodes on the network using terminating resistors. The speed of data transfer is set at 1.5 Mbps.

For the data link layer, Profibus DP uses a Master /Slave technique. The master polls the slaves in sequence and each slave replies in the same cycle. Only the master can control the bus, and communication between master and slave can take place after configuration of the network. Some slaves, such as slave PLCs need software configuration in addition to that carried out for the master. Other slaves need only some hardware configuration, such as addressing using DIP switches, or the setting of parameters on the micromaster drive.

A data telegram on the network can be up to 244 bytes long and checking procedures produce Hamming distance = 4.

Instead of using an application layer, Profibus DP uses a Direct Data Link Mapper to interface user data directly to the Data Link Layer. Manufacturers must provide profiles for devices. This chapter has illustrated the concept of using these profiles in the case of the micromaster drive.

Currently the network is used to deliver an introduction to Profibus DP course, which occupies two days. It is divided into four parts. The first half-day is a general introduction to fieldbuses and Profibus. The rest of the course is spent dealing with Profibus DP networks, where three different networks are implemented, faults are introduced and troubleshooting procedures are practised.

CHAPTER 5

Profibus and AS-Interface for the Mechatronic Development Centre Laboratory

5.1 Introduction

The Mechatronic Development Centre Laboratory is located in the Mechatronics Research Centre - Allt-Yr-Yn campus of the University of Wales College Newport (UWCN). The system consists of a series of stations connected by means of a conveyor as shown in figure 5.1. The stations are designed to work as an integrated production line, where each station has its own separate function. This chapter describes the design of a Profibus DP and an AS-Interface networks for this laboratory. Figure 5.2 shows the hierarchical structure of the two networks. The following sections describe the functions of each station, discuss the need for a fieldbus network and explain the design and implementation plan of the network.

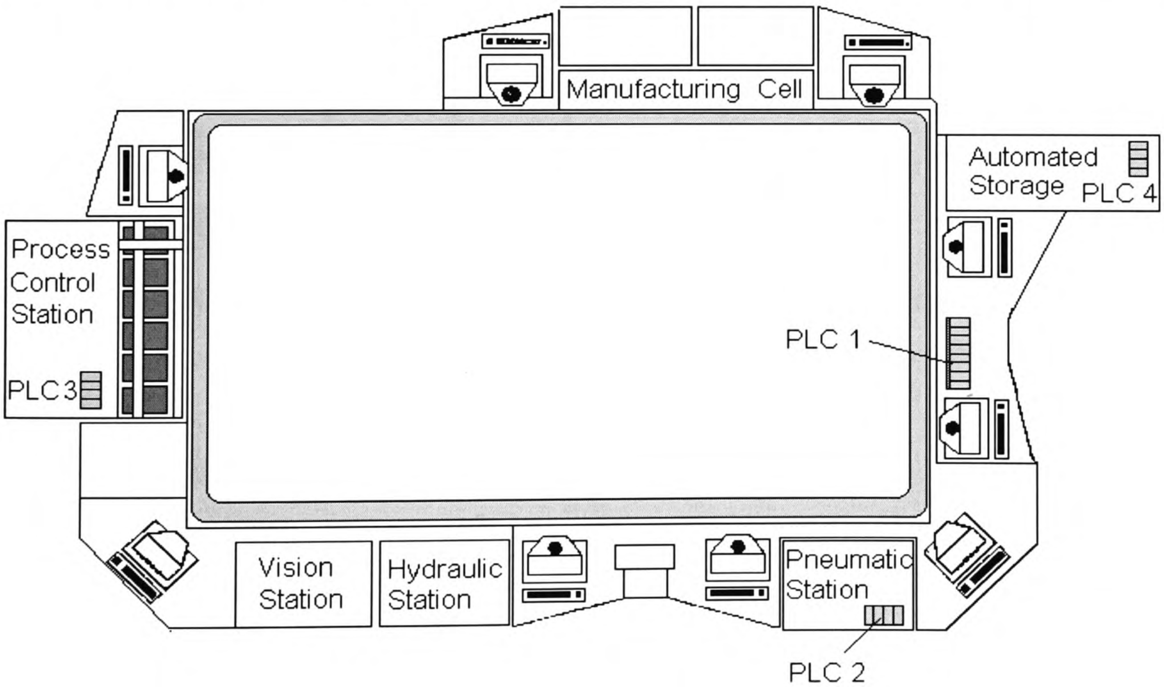


Figure 5.1 Mechatronic Development Centre Laboratory

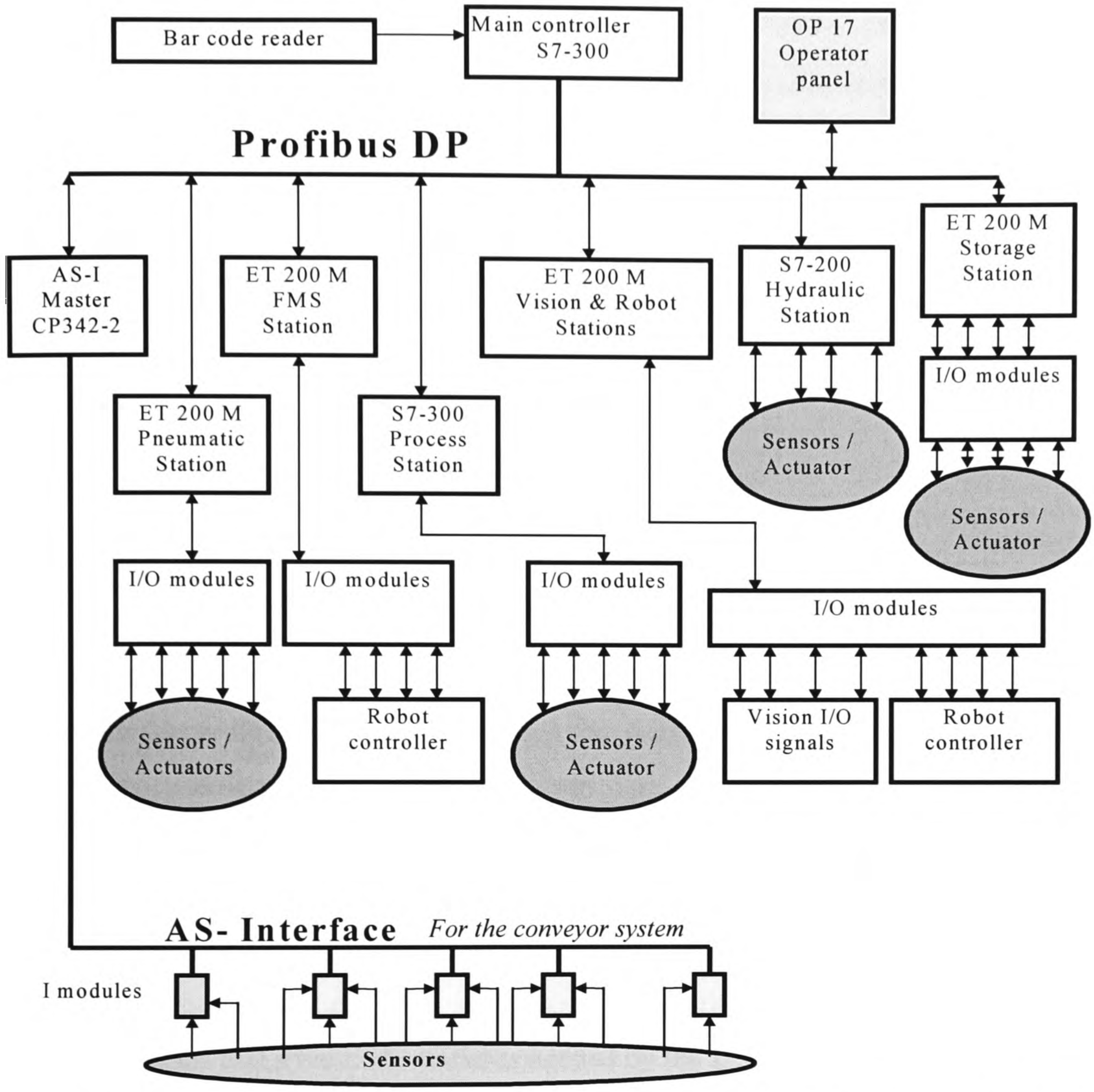


Figure 5.2 The hierarchical structure of Profibus DP and AS-Interface networks for Mechatronic Development Centre Laboratory.

5.2 System components

5.2.1 Conveyor belt

The conveyor transports wagons between the different stations of the system. The wagons pass over sets of sensors lying behind each station. The wagons are coded with inserted metal discs giving each an individual binary number, as shown in figure 5.3 According to the number of available places to insert the metal discs on the under side of a wagon, which in this case is 5, the total number of wagons can not exceed 32 i.e. (2^5). The sensors, which detect the metal discs, are connected to the main PLC which controls the conveyer.

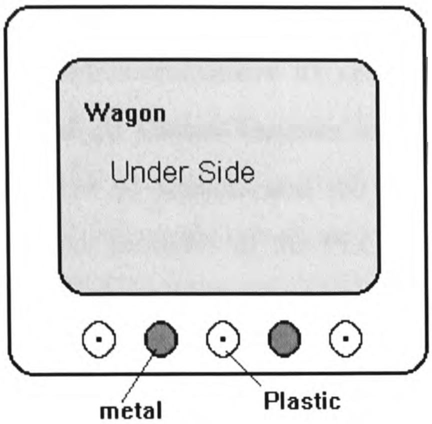


Figure 5.3 Under side of the wagon

When a wagon passes over a sensor set, the sensors detect the wagon number such that the controller (according to the user program) can decide to let the wagon pass or to stop it (using a pneumatic stopper) as shown in figure 5.4.

Another advantage of using coded wagons is that the main controller knows which wagons are carrying components and which are empty.

The wagons are able to carry components on bar-coded carriages. The carriages are designed so the stations can pick and place them easily. The use of the bar codes on the carriages means that a bar code reader is needed by the system.

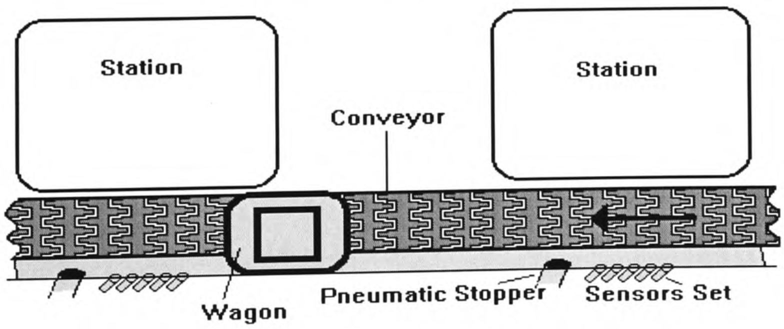


Figure 5.4 Conveyer belt

5.2.2 Pneumatic station

This station supplies carriages and raw materials in the form of aluminium cylinders and Perspex blocks to the production line.

The station uses pneumatic cylinders to achieve its tasks. A 'Modicon' PLC controls the station via 40 Digital Inputs and 20 Digital Outputs as shown in Table 5.1.. The inputs, which are generated by a number of sensors and tell the PLC what is going on in the station, are connected to the Input modules of the PLC, where the CPU can evaluate the status of the input devices. The task of the Input modules is to interface between the input devices and the CPU. Thus the CPU can read the input signals without having to tolerate the high plant voltages. Output devices such as lamps, relays and solenoids are connected to the output modules, which also are part of the PLC.

These output devices receive their signals from the CPU and provide isolation between the CPU and the output devices.

The PLC decides which materials have to be delivered to a wagon and through its sensors detects which materials are available and which are not.

Module No.	Function	Existing Number of I/O
DEP 216	Discrete Input Module	16 Inputs
NUL 200	Empty	
DEP 216	Discrete Input Module	16 Inputs
DAP 216	Discrete Output Module	16 Outputs
DAP 212	Combined I/O Module	4 Outputs/ 8 Inputs

Table 5.1 I/O for pneumatic station.

5.2.3 Lathe and Milling station

This station consists of three units:

1. Lathe unit
2. Milling unit
3. Robot

The station machines the raw materials into appropriately shaped components. The lathe unit deals with the aluminium cylinders while the milling unit machines the Perspex blocks. The robot picks up material from the carriages and places it in the appropriate machine. On completion of the machining operation it returns the component to the carriage.

Each unit has its own program located in a small, dedicated PLC. The main controller instructs the units with regard to which components are to be made.

The main controller controls this station via a number of inputs (output from the station), which inform the controller of the situation at the station. Appropriate output signals from the main PLC (which are inputs to the station) are generated to achieve the task. The number of inputs and outputs for each unit are given in table 5.2.

Unit	Digital Inputs	Digital Outputs
Lathe	4	5
Mill	4	5
Robot	16	16

Table 5.2 I/O for FMS station

5.2.4 Process station

This station plates the aluminium cylinders on completion of the machining process. It consists of seven process tanks which carry out the following operations:

Tank 1 pre-cleans the cylinder.

Tank 2 rinses the cylinder.

Tank 3 dips the cylinder into very dilute sulphuric acid.

Tank 4 rinses the cylinder.

Tank 5 coats the cylinder with 722 passivator.

Tank 6 rinses the cylinder.

Tank 7 dries the cylinder.

The plating process starts when the wagon which carries the cylinder reaches this station. The wagon is stopped by a pneumatic arm. A pneumatic/electrical arm picks up the cylinder and places it into the appropriate tank. The station is controlled by a "Modicon" PLC with 32 digital inputs, 32 digital Outputs, 4 Analogue Inputs and 2 analogue Outputs as shown table 5.3. The inputs inform the PLC on the situation of the station such as tank temperature, the position of the pneumatic/electrical arm and levels of tank contents. The outputs control the various actuators such as heaters, motors, and solenoid valves to achieve the task of the station.

At the end of the process, the component is returned to its carriage and wagon.

Module No.	Function	Existing Number of I/O
DEP 220	Discrete Input Module	16 Inputs
DEP 216	Discrete Input Module	16 Inputs
DAP 216	Discrete Output Module	16 Outputs
DAP 216	Discrete Output Module	16 Outputs
DAU 202	Analogue Output Module	2 Outputs
ADU 205	Analogue Input Module	4 Inputs

Table 5.3 I/O for the process station

5.2.5 Vision Station

A robot arm is used to transfer a component to this station which checks the component dimensions in two planes. The station has an inspection camera and compares the captured image to an image stored in its memory. If the component is within the prescribed tolerances it can be passed to the assembly unit.

5.2.6 Assembly station

This unit assemble the machined Perspex blocks and the passed cylinders. It uses a hydraulic arm to manipulate the components and a hydraulic cylinder to press the blocks and the cylinders together. A PC provided with a suitable card controls this station. The numbers of inputs and outputs in this station are 20 digital Inputs and 20 digital Outputs

5.2.7 Storage station

This station stores the components with their carriages. The components can be at any stage of the process, and this station will store and retrieve on request. It uses a bar code reader to index different carriages. A 'Modicon' PLC controls this station. Table 5.4 shows that the station has total number of 24 digital Inputs and 20 digital Outputs

Module No.	Function	Existing Number of I/O
DEP 220	Discrete Input Module	16 Inputs
DAP 216	Discrete Output Module	16 Outputs
DAP 212	Combined I/O Module	4 Outputs/ 8 Inputs

Table 5.4 Number of I/O for the storage station

5.2.8 Main Controller

The main controller in this laboratory is a 'Modicon' PLC, which has responsibilities as follows:

Control the movement of the conveyor and the pneumatic stoppers.

1. Read and store the code number of wagons.
2. Read and store the bar codes of carriages via the reader.
3. Inform the pneumatic station which material is needed.
4. Inform the robot of the Lathe and Milling station when to pick up a component.
5. Indicate to the Lathe and Milling station which machining processes are needed.
6. Tell the process station when to start.
7. Read the result of an inspection from the vision system.
8. Order the assembly unit to start its operation.
9. Perform the function of the emergency stop.

This station contains a total number of 64 digital inputs and 80 digital outputs as shown in Table 5.5. The outputs are connected to the various stations, and by using the control program located in the CPU, the main station can generate signals to control the whole system.

Model No.	Function	Existing Number of I/O
DEP 216	Discrete Input Module	16 Inputs
DEP 216	Discrete Input Module	16 Inputs
DEP 216	Discrete Input Module	16 Inputs
DEP 216	Discrete Input Module	16 Inputs
DAP 216	Discrete Output Module	16 Outputs
NUL 200	Empty	
DAP 216	Discrete Output Module	16 Outputs
DAP 216	Discrete Output Module	16 Outputs
DAP 216	Discrete Output Module	16 Outputs
DAP 216	Discrete Output Module	16 Outputs

Table 5.5 Input / Output of the main controller

Appendix A shows the complete sequence of operations of the system

All the stations receive their commands from the main controller. consequently if there was no network linking all these stations together, each station would work independently. For example, the pneumatic station might feed the system with wrong material. The FMS station would not know which manufacturing process is needed. The process station may pick up a block instead of picking up a cylinder. The vision system might try to compare the measurements of a cylinder with the measurements of a cube, and the hydraulic station may pick up rejected material and try to assemble it.

It is clear that the system must be networked to perform its task and there are many ways in which this can be done.

Currently the laboratory uses Modicon PLCs to control pneumatic, storage, process and the main station in addition to using PCs to control the rest of the stations. The PLCs are networking using a Modbus Plus Network, which is a local area network system for industrial control applications. It is supported by Modicon products and by other manufacturers. The PCs are networking using a LAN, which means that there are two networks on the system. This means that to maintain the system, experience and knowledge of both types of networks are required.

A study of this system showed that to network any system a number of procedures should be followed. The following sections illustrate these procedures.

5.3 Network Documentation.

This should include the manuals that describe the operation of the system, the purpose of each station or machine, and why networking is desirable or necessary. Also, the specification of the network should be described, the installation material, cable routing and labelling.

In general, for each station or machine the following steps must be followed to achieve a well-documented network.

1. Describe the function of the station on the system.
2. Describe Controller type (e. g. PLC, PC, CNC etc.) with full details of the control strategy. For example, if the controller were a PLC the documentation should contain

information about the type of the device, technical specification, programming language, full wiring description of the inputs and outputs, a copy of the program that is used to control the station, and details on where to obtain technical support.

3. Discuss the reasons for networking the system.
4. Describe the type of network and the functions of the devices on the network (active, passive). The document should contain the address, parameters, the way that the controller is connected to the network and how it communicates with the other nodes together with a diagram of network layout with stations.
5. List the installation requirements and a list of spare parts required.
6. Finally, the documents should describe the most probable faults and how to deal with them.

As an example, information on the pneumatic station of the Mechatronic Development Centre laboratory, with regard to the previous list, should contain the following:

1. A description of the station should list cylinder numbers and the function of each one, motion diagrams of the cylinders, the number of valves and the function of each valve and the timing diagram of the movement of valves and cylinders. Also, the description should contain pneumatic circuit diagrams illustrating the position of the sensors and the valves and the operating pressure.
2. A description of the Modicon PLC (which is used here) should lists PS, CPU and I/O modules, a list of the inputs and outputs with suitable comments to describe the function of each one and the addresses used on the PLC, a copy of the PLC program used to control this station and the address of the vendor of this PLC and how to obtain technical support.
3. The benefits of connecting this station to the network. As described previously this station provides the system with raw material. This means that the system needs to know if raw material is available, and the station needs to know when, and what type of material, to feed to the system.
4. The previous step dictates that the PLC has to be connected to a network. The type of network, which is Modbus Plus, should be described from the point of view of this

controller, the address of this PLC in the network, the parameters that are used and how this PLC communicates with other stations.

5. The hardware and space requirements for installing the PLCs, the cable routing and a list of the available spare parts.
6. Descriptions of the most probable faults, including mechanical faults and the steps necessary for corrective action.

5.4 Profibus network.

As a part of this study a fieldbus network is to be designed for this system. A Profibus DP network is the network of choice for the following reasons:

Two markets studies, namely "Fieldbus system 1998 - 2000" by Consultic & Industrieberating GmbH in Germany and "Global market and user needs for industrial Distributed Remote I/O" by Venture Development Corporation, have confirmed the market leadership position of Profibus. This indicates the need to study this technology and to implement research projects and training programs to support industry with this new technology.

Profibus is an open system (about 140 companies offer over 840 Profibus products), which means that there is a wide range of components to choose from.

There is already a simple Profibus network at the university (see chapter 4), which means that technical support is available.

The following sections describe how the Mechatronic Development Centre laboratory stations can be connected to a Profibus network. The suggested network consists of one master main controller, with all the other stations connected to the master via a Profibus cable. Each station is provided with a suitable slave.

5.4.1 Pneumatic station

This station consists of a number of pneumatic cylinders and storage places. Sensors indicate cylinder positions and the presence of material in storage. The station controller uses the sensor signals to control the movement of the cylinders via a set of electropneumatic valves. Currently a Modicon PLC controls the unit. With a Profibus network, remote I/O modules, figure 5.4, can be connected to the network as a slave, in other words another PLC is not required.

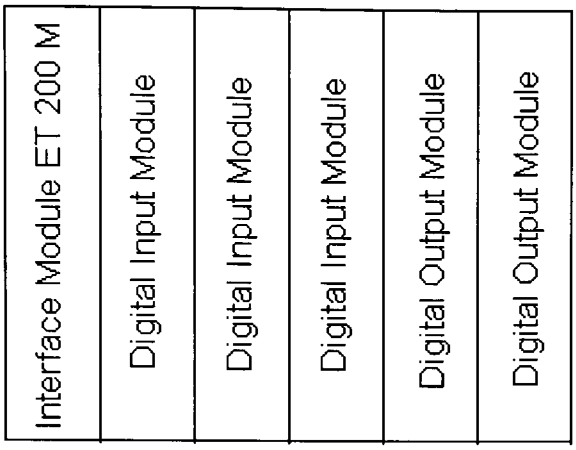


Figure 5.5 Pneumatic station slave

This slave has address 1 on the network and the program which controls this station will be created in function blocks in the master CPU. This will guarantee that the latest information from this station will reach the master without delay. The master will treat the remote I/O as if connected directly to itself. The Function blocks, which constitute the program, can each control separate operations on the station. Signals from sensors can be used to set alarm signals to the user. In this case the network will use an operator panel such as an OP17 and the user should program this panel so those messages appear to indicate the problems.

5.4.2 Flexible manufacturing station (station No 2)

A small, dedicated PLC controls this station, where the PLC receives orders from the master PLC. It then controls the operations of turning and milling. A choice is made between seven different programs for each machine. Each program carries out a different task. This means that when the user chooses to produce a certain item, the master informs all the stations so that all the operations lead to that product.

The robot controller sends digital signals as inputs to the master so that the master knows the state of this station. The master sends digital signals to instruct the robot which process is required.

For this station a remote I/O module can be used for the Profibus network as slave No 3, as shown in figure 5.6. This slave communicates directly with the master. The manufacturing process is completely controlled by the robot controller, the slave only being used to tell the controller what kind of raw material is on the wagon and what kind of manufacturing process is needed. On completion the slave tells the master that the operation is finished. In the main program of the master another function block can be used. When the station is ready, the master receives a signal to indicate that a new order can be sent to the station to start the next manufacturing operation.

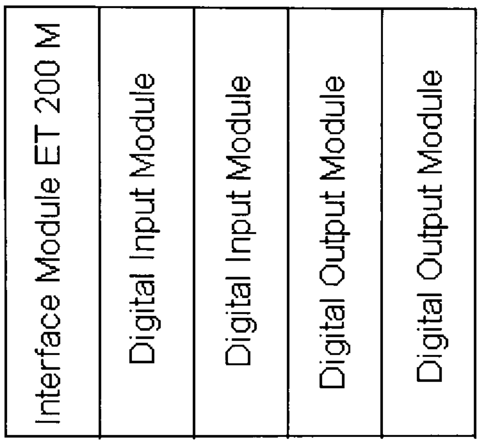


Figure 5.6 Flexible manufacturing station slave

5.4.3 Process station (station No 3)

At the moment the process station uses a Modicon PLC to control its operation. For the Profibus network the proposal is that a Siemens S7-300 PLC be used as slave. When a wagon carrying a cylinder arrives at this station, the master orders the pneumatic cylinder to stop the wagon. Then it orders the slave PLC to start the processing operation. After picking up the cylinder, the slave PLC informs the master that the wagon is empty. The master decides whether to move the wagon or not. On completion of the process operation the slave PLC inform the master and the master sends an empty wagon or orders the slave to put the processed cylinder on to the waiting wagon.

It is clear that the master does not control the processing operation. It only orders the slave to start the operation and receives information at the end of the process. The slave PLC is given address no 4 and it will use an intermediate memory area to communicate with the master.

5.4.4 Vision system

Currently, the controller of the vision system uses digital signals to inform the master if it is ready to take a measurement and to report the results. It also receives digital signals from the master telling it to start measuring. Remote digital I/O modules can be used here as slave No 5. The robot controller can be connected to the same slave with additional I/O modules so that the master can control operations.

5.4.5 Hydraulic station

A PC with I/O card controls the hydraulic station. The movements required of this station are not complicated. The intention would be to use an S7-200 PLC (CPU 215) to control this station. On the Profibus network this PLC will be slave No 6. It would operate in the same way as the process station, receiving an instruction from the master to start operations and informing the master when operations are finished.

5.4.6 Bar code reader

This can be connected to the master by means of a suitable interface card.

5.4.7 Storage station

The storage station will be controlled by the master as a remote I/O station acting as slave No 7. Figure 5.7 shows the arrangement of the intended Profibus network for this laboratory.

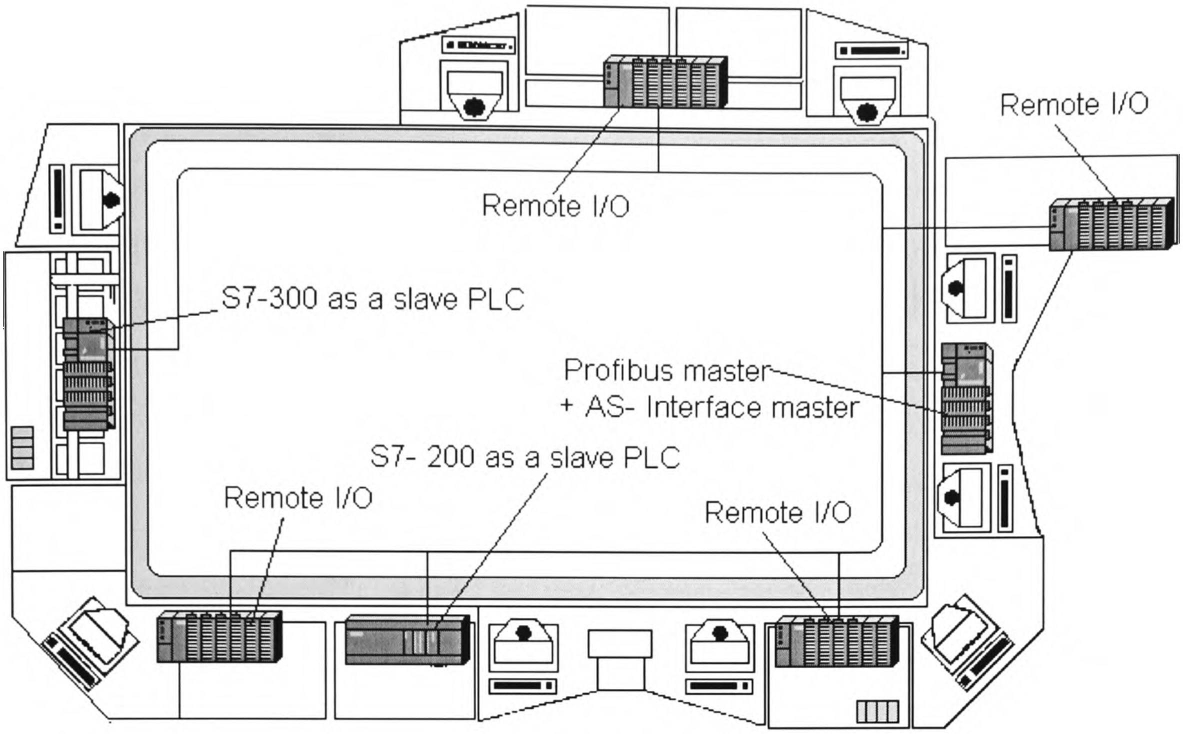


Figure 5.7 Profibus network for CIM laboratory.

5.5 AS-Interface network.

An AS-Interface network can be used in this laboratory to connect all the sensors around the conveyor over a single cable. Figure 5.8 shows a process diagram illustrating the planning of the network. Using the steps in figure 5.8 the AS-Interface network of the Mechatronic Development Centre laboratory can be designed. It is important to mention here that the Profibus network could be used to link the conveyor's sensors to the master, since a number of remote I/O slaves are available around the conveyor. However, in order to extend the facilities of the laboratory to illustrate the performance of different networks an AS-Interface network design which links with the Profibus DP network described in the previous section is explained. By having two different networks in this way, the flexibility of the system can be exploited for research, teaching and industrial demonstration purposes.

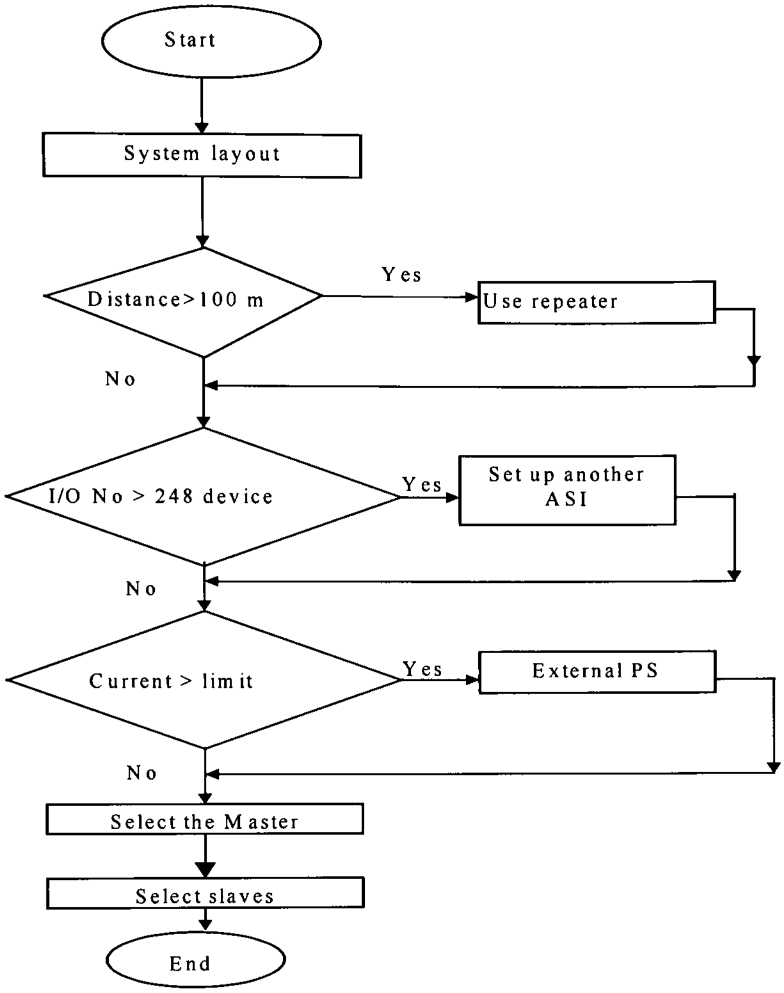


Figure 5.8 Planning AS-I network

5.5.1 System Layout.

One of the first considerations is the cable routing, keeping in mind that different types of topologies can be used. The network is required to connect to the sensors around the conveyor so here the topology can be as shown in figure 5.9.

Studying the system layout provides the network designer with knowledge of features such as cable layout, number of inputs (sensors) and outputs (actuators) and the required components.

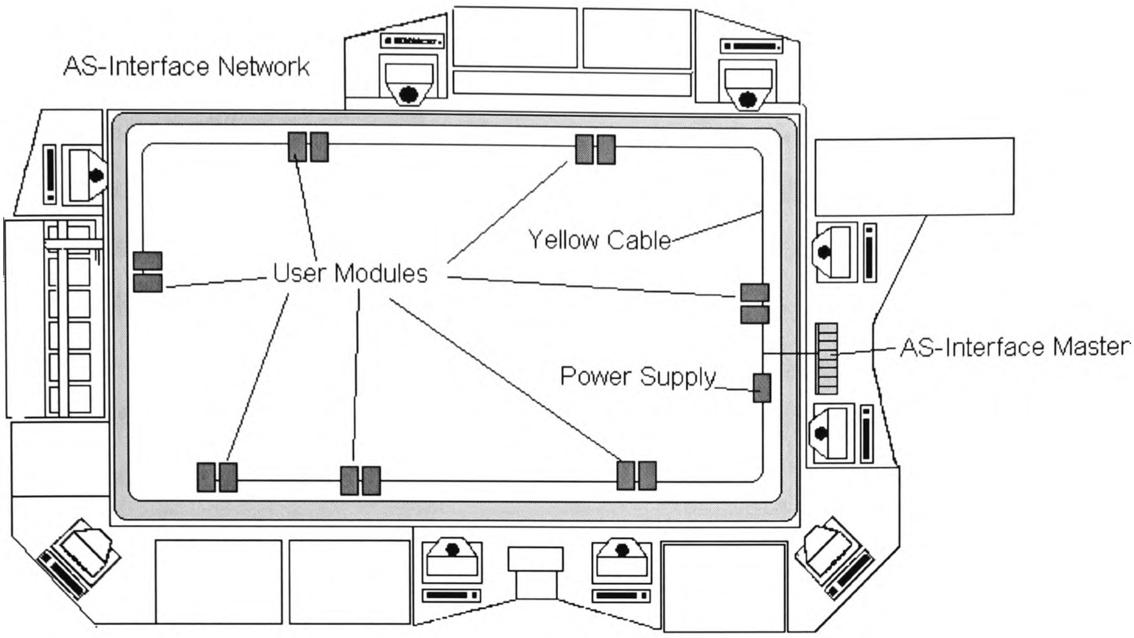


Figure 5.9 AS-Interface network for CIM system.

The maximum number of sensors and / or actuators that can be connected to a slave is 4 inputs and/or outputs, which means that the user has to be sure that he chooses the right modules. Generally, the modules consist of two parts, the **lower section** (coupling module) and the **upper section** (user module). The coupling module, figure 5.10, is responsible for the mechanical connection of the AS - Interface cable to the user module.

The user module contains all the electronics required for communication to the inputs and outputs. To ensure that the coupling modules and user modules fit together, the interfaces have been standardised. There are two types of interfaces:

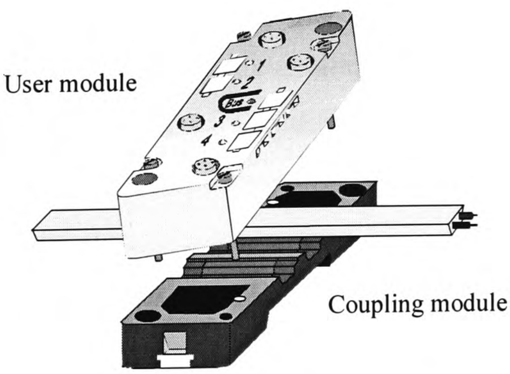


Figure 5.10 AS-I module

Electromechanical interface (EMI)

In this type, only the yellow signal cable is connected. Two cables can be joined together in order to allow the network to be branched from the module. This module has two electrical connections to the user module (ASi + and ASi -).

Extended Electromechanical Interface (EEMI)

The EEMI devices allow connection of the auxiliary 24 V d.c (black) or 240V a. c (red). supply used for driving outputs. It has two cable entries, one for the yellow cable and one for the (black or red) cable.

Each module is assigned a slave address between 1 and 31 by using a hand-held programmer.

Since the number of sensors for each station is 6, the network will use two types of modules for each set of sensors as follows:

1. 4 way input module which connects to standard sensors
2. 2I / 2O module. The outputs of the second module may be used for future expansion of the system.

5.5.2 Power supply

In choosing the power supply it is necessary to distinguish between the AS Interface power supply and a standard 24 V power supply.

An AS Interface power supply is needed for the supply of the AS-Interface system.

The standard power supply is used in case of high current consumption by field device.

The AS-Interface power supply provides power to both the I/O modules and the master. Power supplies ranging from 2.2 to 7 A are available to match the power requirement of the network. The current consumption of the different modules is between 15 to 60 mA, depending on the type. A current requirement of 100 mA has to be included in the power calculations to supply the master.

All the supplied modules are short circuit protected. Each input module is able to supply standard sensors with a total current of up to 200mA. If more current is consumed than is typical for the module, the module automatically cuts itself off from the network and a fault is reported to the master.

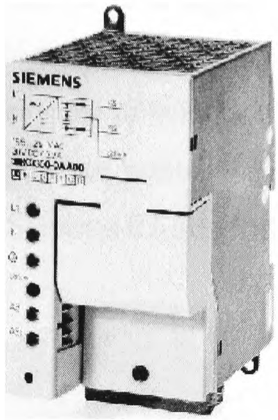


Figure 5.11 AS-I power supply

In the case of modules that take power for the inputs and outputs, the sensor supply is tapped internally on the module board and supplied to the outputs as a load voltage. The maximum current available for each module is 200 mA.

5.5.3 Cable length

As already described, the cable can be laid in any way through the plant or machine.
Since this particular network is located in a small area, the total length of the cable needed is about 18 meters, which means that there is no need for a repeater.
Repeaters are used to amplify the signals on the network, and can be placed in any location on the bus. The maximum number of allowed repeaters on one branch is 2. If more than two repeaters are used, then the delay of data transmission will be increased, which may lead to loss of some data.

5.5.4 Selecting the master

The range of masters available provides a choice of different structures. The master can be implemented centrally with a PLC or a PC, or in a distributed system using fieldbus systems (e.g. Profibus DP and DP/ASI link). The master with central I/O modules can be used in parallel with an AS-Interface network since they do not interfere with each other.

Each uses different address areas in the host memory. It is very important to ensure that duplicate addresses are avoided.

To make the AS-Interface data available to a higher level of the control hierarchy, the master must be interfaced to the higher level fieldbus. The host PLC that contains the master of the AS-Interface must be capable of being used as a slave for the higher fieldbus. As an example of a possible AS-Interface master we can consider the CP 342-2 can be considered which operates with the Siemens S7-300 PLCs.

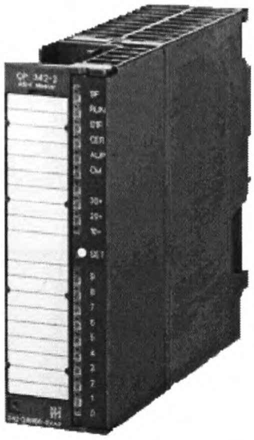


Figure 5.12 CP342-2 AS-Interface master

From the point of view of the CPU, the master behaves logically as two expansion modules.

The master occupies 16 input bytes and 16 output bytes in the I/O address area of the host (in this case the main controller of the Profibus network, S7-300). The start of this address area is decided by the slot number of the master. Figure 5.13 shows the starting addresses.

model	**PS**	**CPU**	**IM**	**CP**	**CP**	**CP**	**CP**	**CP**	**CP**	**CP**	**CP**
Slot No	1	2	3	4	5	6	7	8	9	10	11
Start address	1	2	3	256	272	288	304	320	336	352	368

Figure 5.13 I/O starting address

Each input or output byte address consists of two nibbles. The host PLC uses each nibble to define a slave. (See figure 5.14)The PLC program uses the AS-Interface data in the same way as I/O arriving from the central input and output modules.

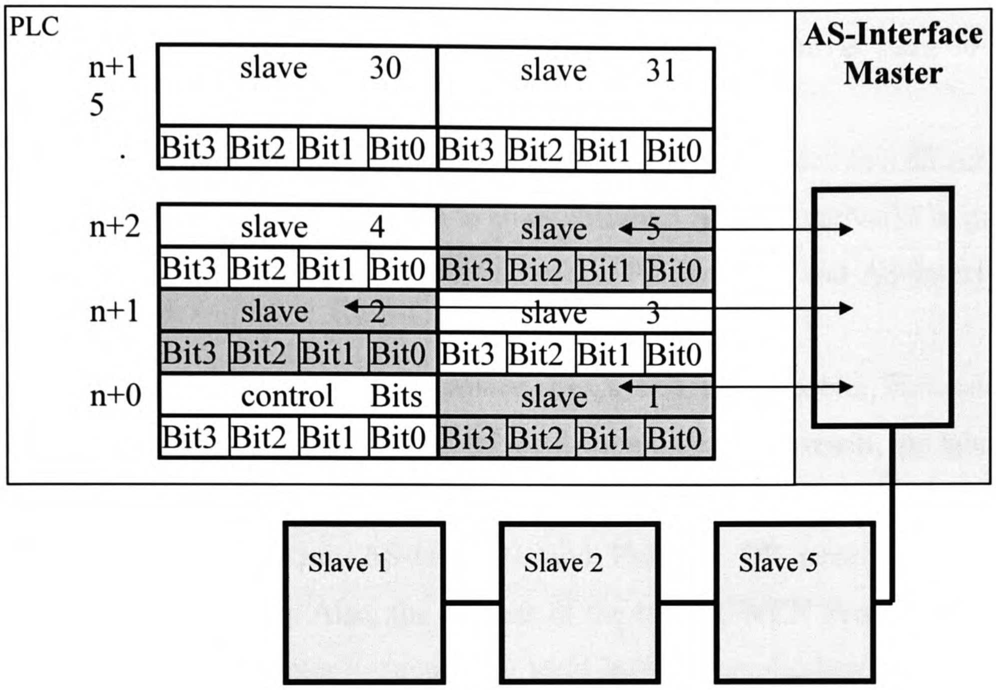


Figure 5.14 Data manipulation

5.6 Installation

After choosing the network components, installation can be done in the following order:

Installing the coupling modules.

Laying the cable from the master (control cabinet) to the modules.

Connecting the power supply.

Installing the user modules, and addressing them (the master can detect double addressing of slaves and indicate an error).

Before the master can begin cyclical data exchange, a List of Configured Slaves (LCS) must be created. This list contains the addresses of slaves on the network. At each start-up of the system the master creates a List of Recognised Slaves (LRS), and compares this list with the LCS. All the slaves that are successfully activated are then written into the List of Active Slaves (LAS).

5.7 Summary

This chapter has described the design and implementation of Profibus and AS-Interface networks for the devices in this laboratory, which consists of several stations; pneumatic, FMS, process, vision, hydraulic, and storage. A conveyor system is used to link the stations.

The main uses of this laboratory are to offer support to local companies as a didactic centre, and also provide a research base in which to study different types of networks to provide an opportunity to study two different fieldbus networks, Profibus DP and AS-Interface were implemented.

At the moment the laboratory uses Modicon PLCs and the Modbus Plus networking system. Neither is used to any great extent by local companies. As a result, the laboratory is not widely applicable to the needs of local industry. On the other hand Siemens PLCs are widely used, and implementing AS-Interface and Profibus DP networks more nearly matches current requirements. Also, the success of the basic UWCN Profibus DP network described in the previous chapter in supporting local industry emphasises the importance of developing this laboratory.

In this chapter the procedures that are needed to document a network for any system are described. Also, the planning of an AS-Interface network for any system has been discussed.

The Profibus network is used in the control of the different stations in addition to interfacing with an AS-Interface network, where the master of the AS-Interface is a slave on the Profibus network. Figure 5.2 shows the hierarchical structure of the two networks. The cost of the whole control system is listed in appendix B.

CHAPTER 6

Industrial Case Study at Brohome Ltd.

6.1 Introduction

Brohome is a manufacturer of cladding, insulation and roofing materials, which are used in building and agricultural units. The company has been in existence for 21 years and is located on the Dyffryn industrial estate near Ystrad Mynach in South Wales. It makes a range of products to order. All of the products are made from steel coils, which are purchased from a number of different suppliers.

6.2 Manufactured Products

The products are made on five semi-automated production lines shown in figure 6.1, as follows:

19 mm Corrugated cladding

26 mm Corrugated cladding

34 mm Corrugated cladding

34 AP (Architectural Profile) Corrugated cladding

Figure 6.1 Production lines

The fifth production line is used to manufacture "Z purlins", which are used to fix cladding sheets and to support roofing materials, as shown in Figure 6.2.

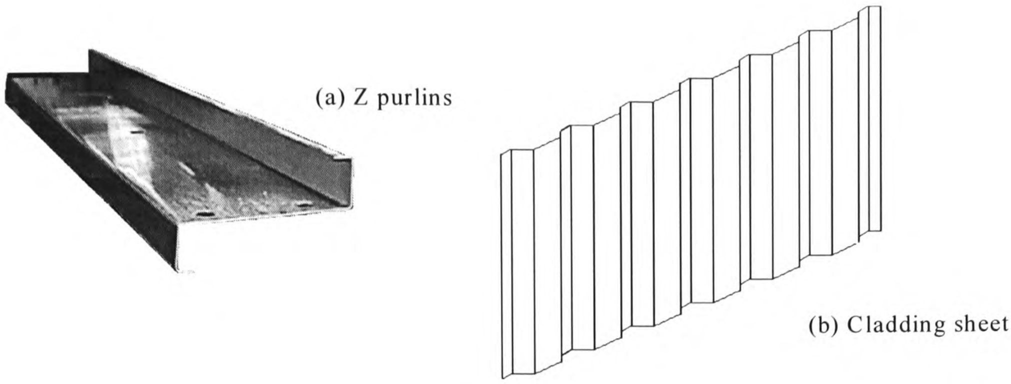


Figure 6.2 Brohome Products

6.3 Methods of manufacture

Steel coil is loaded on to mandrels and the start of the coil is fed onto a conveyor, (figure 6.3). Passing the steel sheeting through a series of heavy, shaped rollers, which press the corrugated shape into the steel, (figure 6.4), produces cladding. Once the desired length of cladding has passed through the rollers, a shearing device cuts the sheeting.

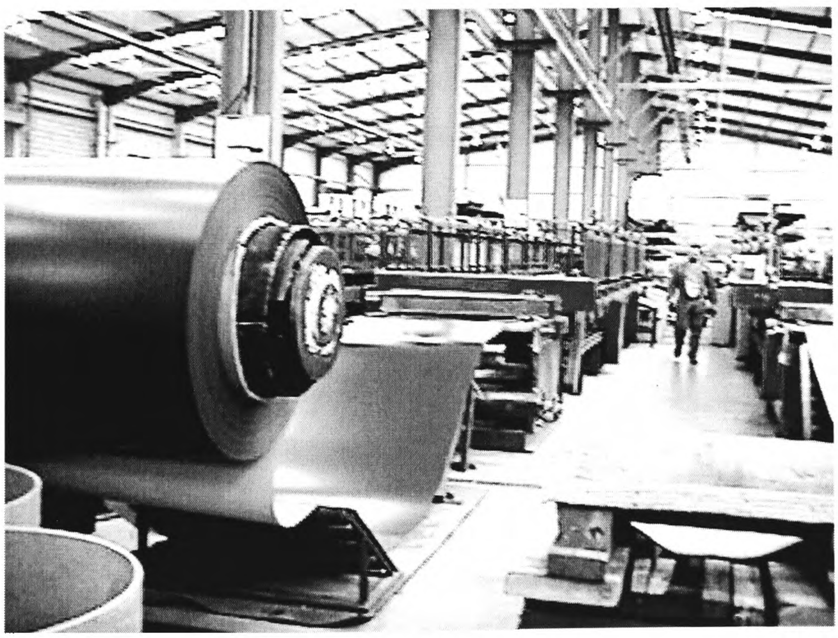


Figure 6.3 Conveyor

Figure 6.4 Profiled rollers

6.4 Production line controllers

Each production line is controlled by a PC, which is linked via coaxial cable to a Novell network version 3.12. The PC uses communication cards to obtain signals from the line, send signals to the line and to communicate with the file server, located in the administration office.

The production PCs run applications written in Turbo Basic operating in a DOS environment. They receive the works order information from the file server in the form of a text file. The appropriate PC references the information and shop floor operatives select the job to run from a menu option, (figure 6.5)

Figure 6.5 Works order screen

The main disadvantages of the system can be summarised as follows:

Tracing faults is difficult and time consuming, because the control cabinets are complicated and not well designed (see figures 6.6)

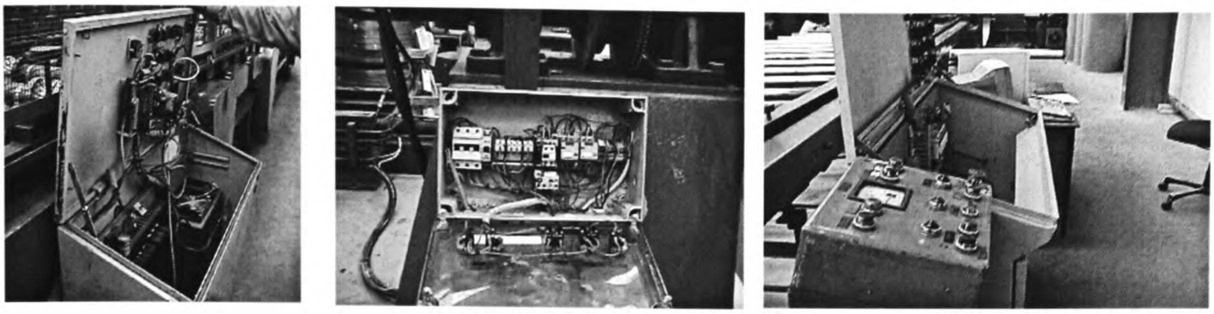


Figure 6.6 Different control cabinet

- The PC communication cards figure 6.7 are expensive to replace, take a long time to manufacture (by hand), and are sourced from a single company.

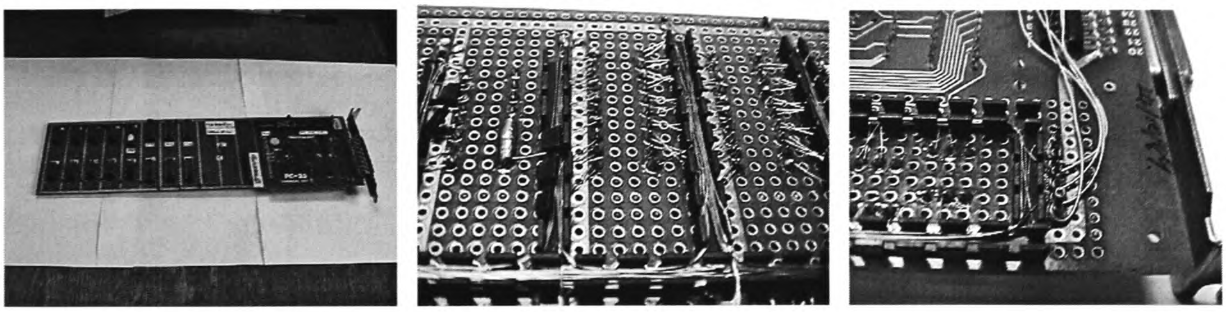


Figure 6.7 Hand made PC card

- The production PCs are rather old and prone to problems (They are not designed for industrial applications on a shop floor) (figure 6.8).

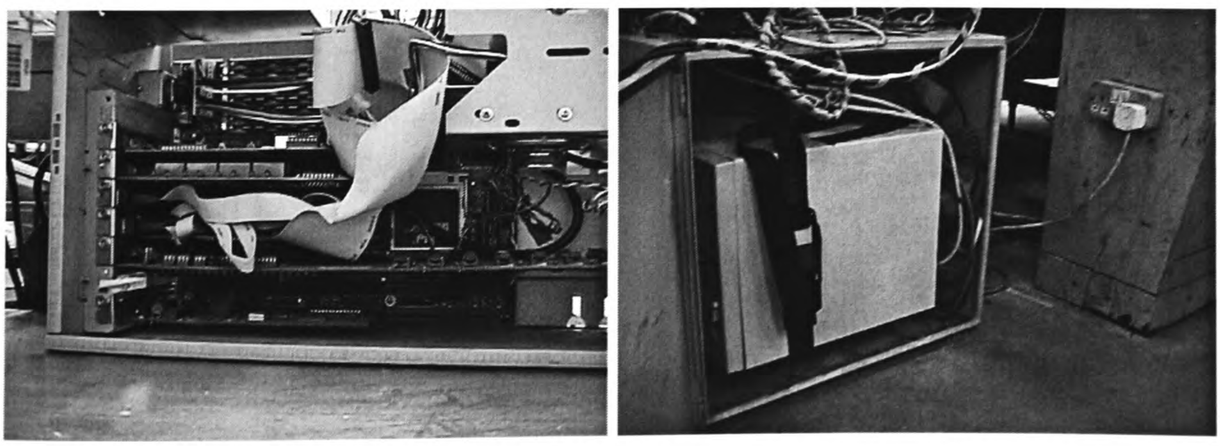


Figure 6.8 Shop floor PC

- Some devices, for example the operator panels, cannot be replaced since they are no longer manufactured. (Figure 6.9)

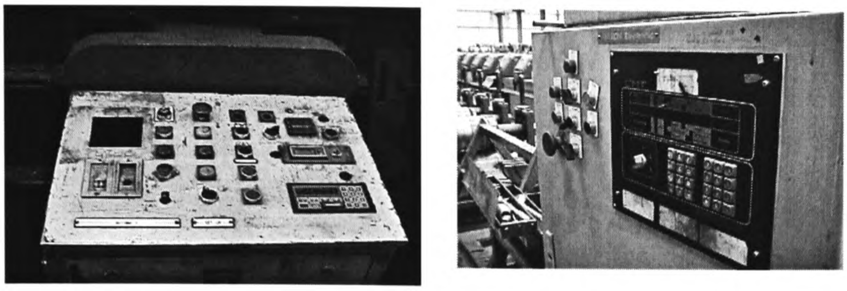


Figure 6. 9 Operator panels

6.5 Control development

To overcome these problems the control system of the factory should be upgraded or modified, since eventually the system will fail and parts will no longer be available. A study of the present situation (with future development in mind) leads to the conclusion that new control systems should be installed. Figure 6.10 shows a proposed structure for automating the factory.

Situated at the lowest level in the plant hierarchy are the machines and units that are directly in contact with the manufacturing process. At this level the machines are controlled by PLCs.

Choosing the protocol that should be used in a system depends upon the requirement of the system. In this particular case, HART is not suitable, since this protocol is mainly used for process automation, where the manufacturing process is continuous.

Choosing CAN protocol would not be a good decision, since this protocol is not optimised for data greater than 8 bytes in length. Also, different incompatible CAN based protocols are available in the market, which is an added complication. If the company were to decide to add some process production lines in the future they would need to study which protocols would be compatible with the chosen system.

Foundation Fieldbus protocol is designed to be used in both process and manufacturing automation. The version, which deals with process control, is known as H1, and more popular in North America than in Europe. The H2 version, which deals with manufacturing automation, is not yet available.

As discussed in Chapter 3, Profibus covers all industrial applications. It is more popular than any other fieldbus system in Europe, where it is supported by Siemens, one of the major players in the PLC marketplace. One of the advantages of using Profibus is that Brohome Engineers can obtain technical support from UWCN where a number of laboratories use this technology.

Choosing the right protocol not only depends upon technical specifications. The survival of the protocol over the next few years is also important. Two markets studies, namely "Fieldbus system 1998 - 2000" by Consultic & Industrieberating GmbH in Germany and "Global market and user needs for industrial Distributed Remote I/O" by Venture Development Corporation, have confirmed the market leadership position of Profibus. all these arguments tends to indicate that the choice of a Profibus network for this installation would be appropriate.

In cases where the lengths of the production lines are long and the number of the sensors and actuators is high, an AS-Interface fieldbus can be implemented to reduce cabling costs and maintenance time. However in this case, the number of inputs and outputs on the production line is small (see table 6.1). This means that there is no advantage in using an AS-Interface network. A comparison between the cost of a master and other network equipment and the cost of traditional wiring shows that the use of traditional wiring is cheaper. Table 6.2 compares the cost of equipment for the 34 AP production line which is shown schematically in Figure 6.11.

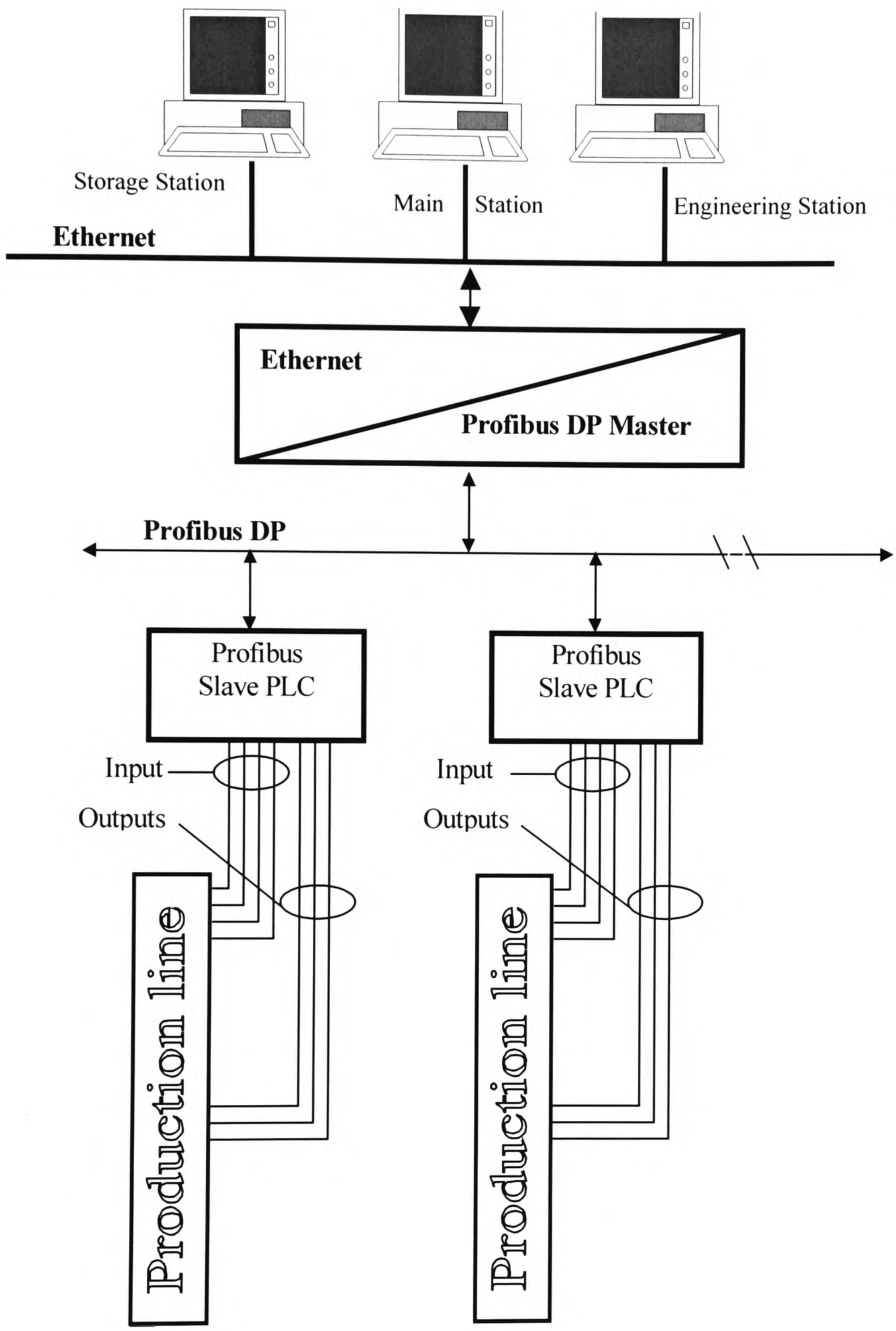


Figure 6.10 Proposed network

Inputs			Outputs		
Input 0	Coil sensing	1 D I	Output 0	Mandrel moving	1D O
Input 1	Emergency stop	2 D I	Output 1	Cutting device	1D O
Input 2	Guard sensors	4 D I	Output 2	Hydraulic cylinders	1D O
			Output 3	Conveyor motor	1D O

Table 6.1 No. of Inputs and Outputs for 34 AP production line.

AS - Interface		Traditional cabling	
Item	Price (£)	Item	Price (£)
AS-I cable (100 m)	112	Cable (2-core, 0.75 mm, 100m)	20.62
AS-I Master	228		
AS - I user module	5.78		
AS - I application models	65.66		

Table 6.2 Cost comparison AS-I/ Traditional for 34 AP production line.

Traditional cabling means that all the inputs and outputs are connected directly to the PLC with an analogue input module used for the measuring device.

In order to illustrate how the new system works, consider the processing of an order to produce 10 sheets of cladding that are 3 meters in length. The process starts at the main station, (figure 6.10), which receives the order from the manager. This station communicates with the engineering station to collect the required data such as raw material type, then it interrogates the storage station to cheek the availability of raw material. (These stations could be a single unit or separate units each having its own function). The order is transferred to the plant floor via a gateway, where the implementation can start. The line operator sees the order on his operator panel, and can decide to start or not. The main station monitors his actions. When production starts, 3-m cuts are made and the sheet count incremented. When the count reaches 10 the process stops.

The control communications architecture consists of two levels, namely Ethernet and fieldbus. The requirements of data exchange within each level vary considerably. At the bottom of the hierarchy, a real time system must process data at speeds faster than for levels higher-up in the hierarchy. To illustrate this, suppose that the measuring device sends a signal to the controller to indicate that the measured length is 3m. The controller should order the cutting device to act immediately otherwise the sheet is useless. At the Ethernet

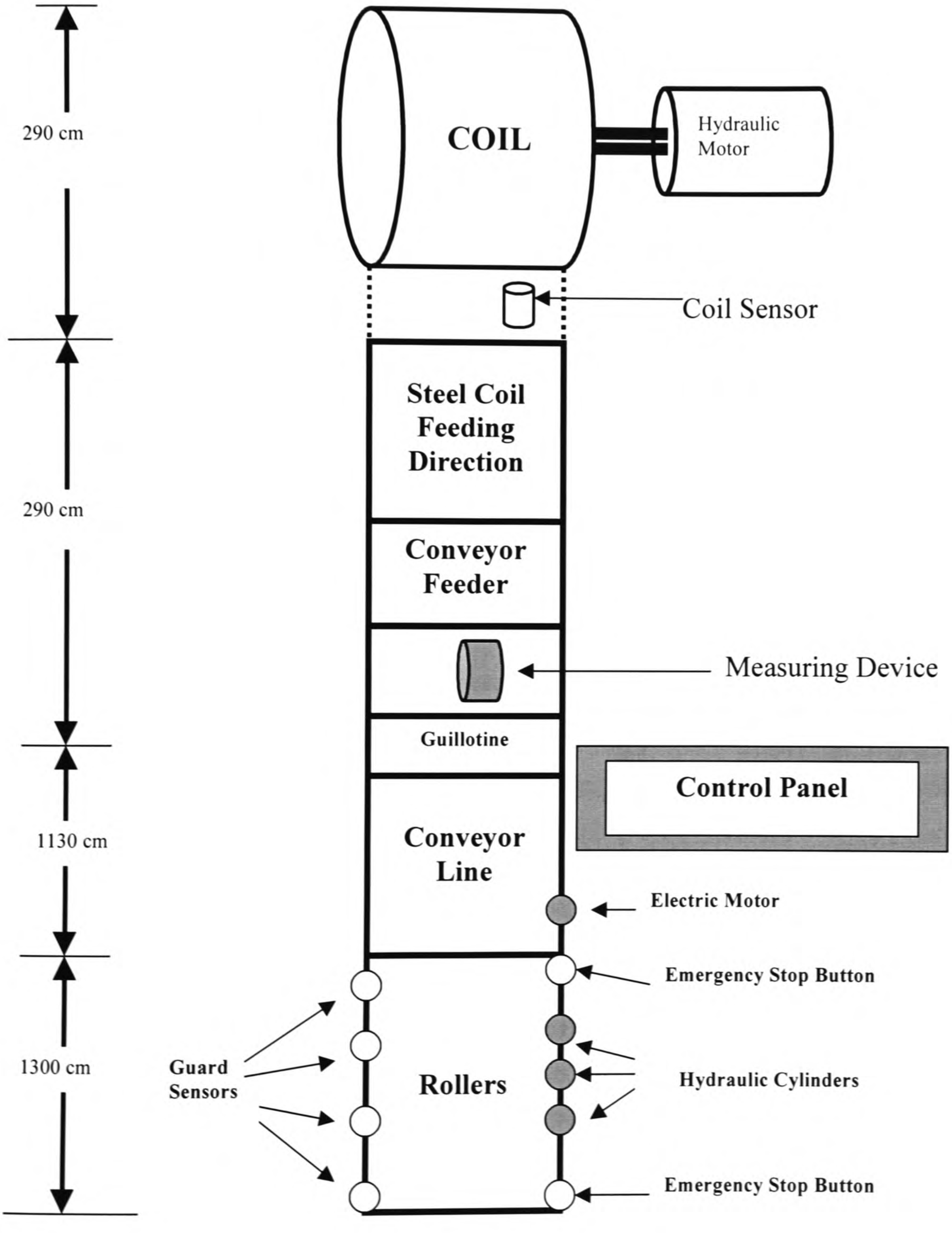


Figure 6.11 34 AP production line

level the reaction times can be longer. The precise time at which an order is sent over Ethernet to the production facilities is not important.

The fieldbus level consists of a Profibus DP network. Here, each production line can be controlled by a PLC connected to the Profibus network as a slave as discussed in the next section.

For the measuring device a "rotary position encoder" as shown in figure 6.12 can be used with a counter.

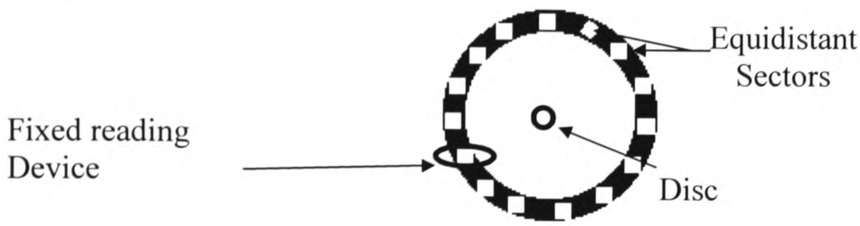


Figure 6.12 Rotary position encoder

6.6 Connecting the PLCs to Profibus DP.

A Profibus network is recommended to connect all the production line controllers to the main controller, as described in figure 6.10. Since Profibus is an open system, any make of PLC can be used as long as it supports Profibus DP. It is recommended that the conversion to Profibus to start with only one production line. The advantages of starting with one line are:

1. New equipment can be tested in a real application, and the weaknesses identified. Unexpected faults can be overcome before the development of other lines starts.
2. The line can be used to train maintenance engineers and operators, so that they become familiar with the new technology and then actively support the development of the other lines.
3. Initial investment by the company is kept to a minimum until tangible benefit can be realised.

Discussions with Brohome engineers have shown that production line 34AP suffers many failures, especially in the control and communication cards. For this reason the decision to start with an upgrade of this production line has been made.

The implementation procedures can be described as follows:

1. Choose the production line, which is to be developed, and define its functions.
2. Study the production line in order to define requirements for input and output modules. This will help in the choice of PLC to be used as a controller.
3. Consider the need to connect the controller to other devices in the plant, and decide on the type of protocol to be used for communications. For example, the S7-200 CPU 214 can control the line, but it has no facility to communicate with a Profibus network, instead, a CPU 215 should be chosen. In this case, the system is designed to use a Profibus DP network, and all the associated equipment (see appendix B) is chosen to support this protocol.
4. After choosing the equipment and the protocol for interconnection, which might be called the Hardware part, the implementation then considers the software. It is important to generate a document at each stage (see section 5.3). Finally these can be grouped to provide full document for the project.

It is important to remember that there are a number of important considerations for any manufacturer when installing a new production system. These relate to the Reliability, Risk Assessment and Failure Modes of the equipment (see appendix D).

6.7 New controller unit for 34 AP production line.

Figure 6.13 shows the control unit of the production line to consist of:

1. PLC with a Profibus DP interface (Siemens S7-200 CPU 215 DP). The PLC is provided with an analogue module.
2. Operator panel (TD 200 text display) linked to the PLC. The operator panel is connected to the PLC via the PLC's programming interface (PPI) and can be used to display messages, change set points (the length and the number of sheets), and to start/stop production.

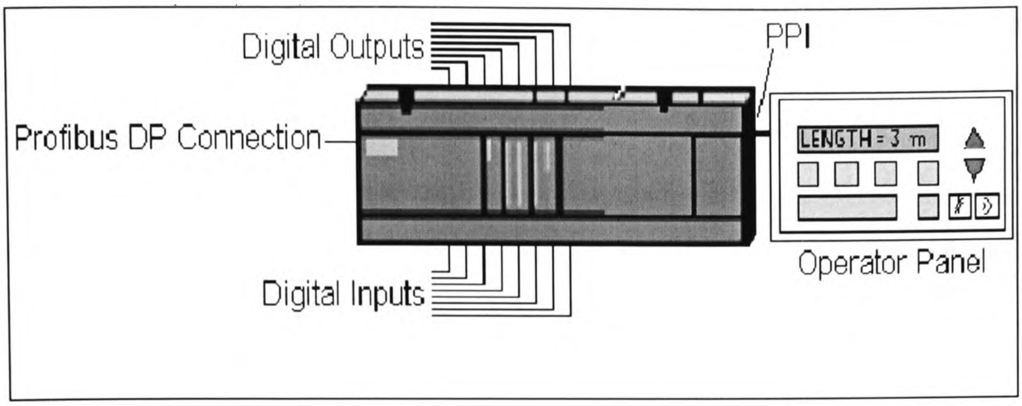


Figure 6.13 The new controller of 34 AP production line

The PLC S7-200 CPU 215 DP has the following features:

- It has two communications interfaces, PPI and PROFIBUS-DP.
- It can be used as a PROFIBUS-DP slave.
- It can be programmed via PROFIBUS-DP with STEP 7-Micro/Win V2.1.
- An Integrated 24 V sensor/load power supply, for direct connection of sensors. With its 400-mA output current it can also be used as a load power supply for small devices.
- Integrated digital inputs (14) and outputs (10).
- Easy expansion using digital and analogue modules.

Table 6.13 lists some of the S7-200 CPU 215 PLC specifications

Program memory	8 Kbyte /typ. 4 K statements on built-in EEPROM (non-volatile)
Data memory	2.5 K words
Programming language	STL and LAD
Counters	256
counting range	0 to 32767
Timers	256
digital inputs/outputs	Max. 62 inputs and 58 outputs (incl. on-board) inputs/outputs)
analogue inputs/outputs	12 inputs and 14 outputs
Max. configuration	7 expansion modules (digital and analogue)
Weight approx.	500 g

Table 6.3 PLC S7-200 CPU 215 DP Specification

The S7-200 series PLCs are programmed using STEP 7 micro/WIN software (see appendix A).

As already stated, Profibus DP will be used to network all the production line controllers. A Siemens S7-300 315 DP PLC is suggested as the master for the network, with an S7-200 PLCs to control the production line as slaves.

To ensure reliable communication between master and slave, certain settings have to be made on the participating devices.

6.8 Setting the master PLC

Setting the communication parameters and initialisation of the master is carried out using the STEP7 software package ver 3.1 or higher. Setting a Profibus address for communication between the programming unit or PC and the master and for communication between the master and the slave can be done during the hardware configuration within the software.

The master PLC is located in the main cabinet. A number of inputs such as the emergency stops, and outputs such as warning lamps can be connected directly to the master. For this purpose 1 input module and 1 output module are used in the main PLC.

6.9 Configuring the slave PLC

The configuration of the slave is carried out using STEP 7 software. Here, the address of the PLC as Profibus slave can be specified (0-126). The slave sends and receives data to and from the master via its variable memory area. Setting an offset configures the start of the receiving area in the slave. For example, if the offset is 0 the receiving area starts at VB0. The size of this area is defined in bytes. If the size is 16 bytes and the offset is 0, the receiving area starts at VB0 and ends at VB15. The sending area for this example is shown in figure 6.14, extending from VB16 to VB31.

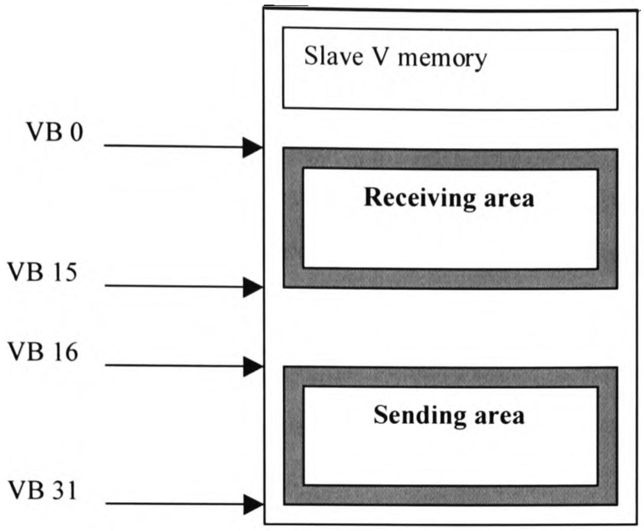


Figure 6.14 Slave communication areas

6.10 Data exchange

When the connection of the slave PLC to the master via the Profibus network is successfully accomplished, the slave will enter the data exchange mode, as shown in figure 6.15. Data from the master must be taken from the receiving area by the program in the slave. Likewise, the output data, which the slave wishes to send to the master, must be placed in the sending area.

For example, consider a simple case, where the operator presses a push button represented by input I0.0 on the master to test the alarm horn on production line 34AP.

The program in the master transfers this signal to the communication area in the master, say PQW 256, (the person programming the PLC must know the configuration setting in order to create the program). The signal, received in the V memory of the slave, is used by the slave to set the alarm.

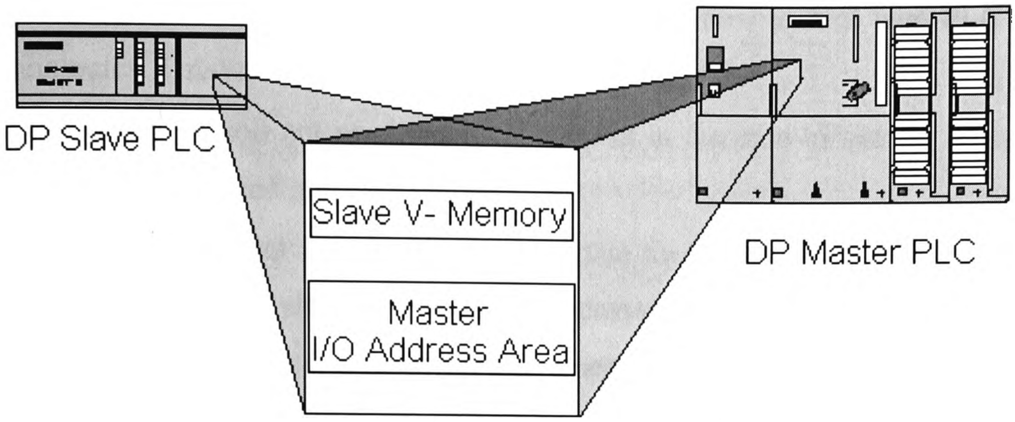


Figure 6.15 Data exchange

6.11 Communication with higher levels

A CP 343-1 module can be used to interface the DP master PLC to the Ethernet network. An Ethernet card in the main PC (such as a PC 1411 from Siemens) connects the PC to the Ethernet network, as shown in figure 6.16. The CP 1411 can be operated with the SOFTNET package for industrial Ethernet.

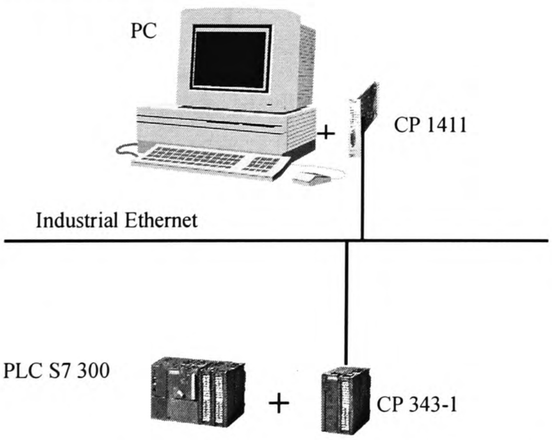


Figure 6.16 Connecting Profibus DP Master to Ethernet level

6.12 Conclusion

This study suggests that an up-dating of the Brohome production and control systems is necessary. The implementation of this development is to be carried out by Brohome engineers after attending suitable training courses on how to deal with PLC and Profibus technologies. This strategy has the following advantages:

- The company will not need technical support in the case of simple faults, resulting in savings of time and money.
- Brohome engineers will be able to continue to develop their production lines in the future. This, again, should result in major cost savings.

Figure 6.16 shows the sequence of the implementation

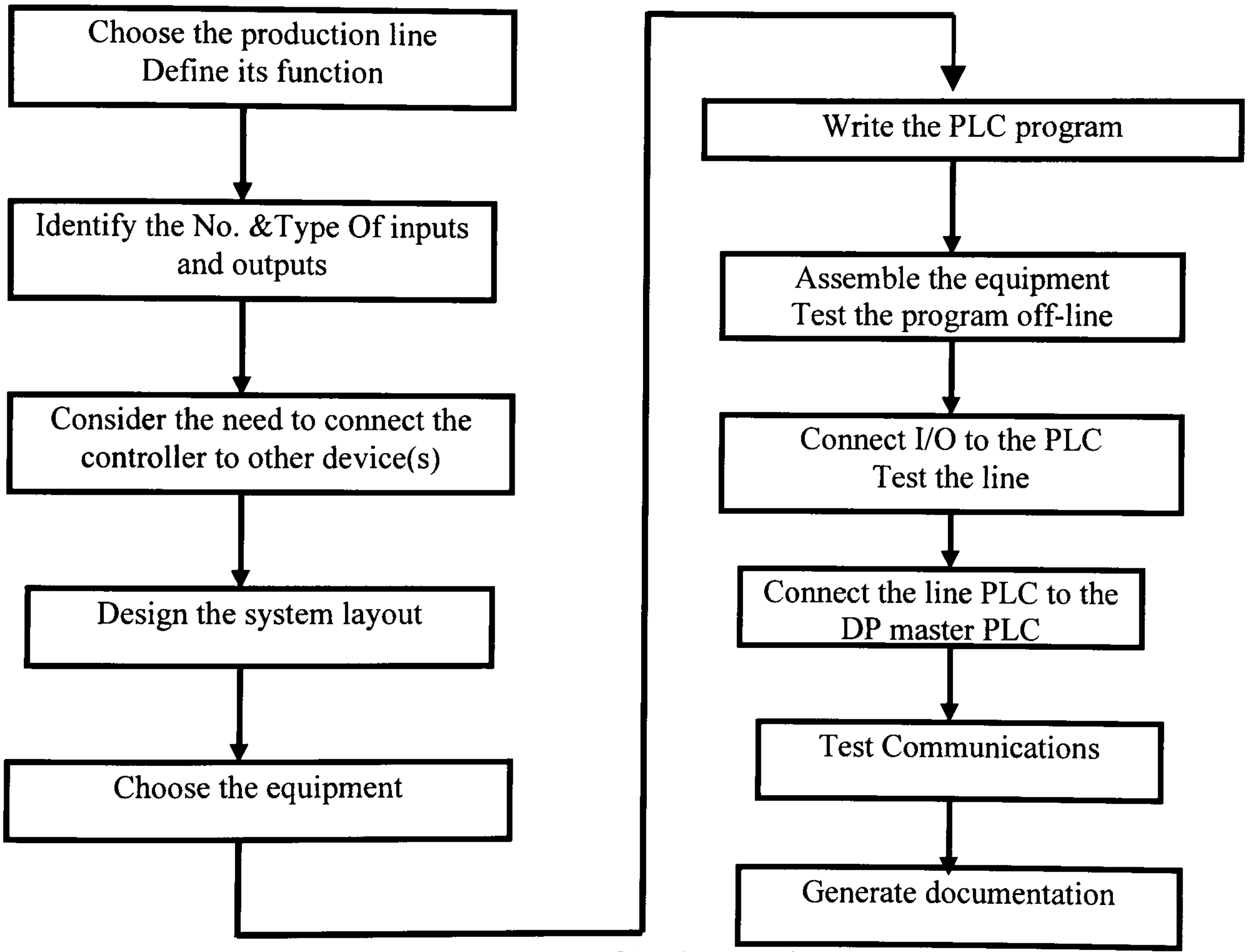


Figure 6.17 Sequence of implementation.

An important part of the pervious diagram concerns the writing of the PLC program. Brohome Ltd. needs to consider training some of their engineers on how to write a PLC program and how to link a PLC to a Profibus network. A suggested training schedule might look like:

1. Introduction to PLCs. (one week)
2. Advanced PLC programming. (one week)
3. Profibus DP network course. (one week)

Further points to consider when choosing fieldbus equipment, in addition to the cost are:

1. Access to technical support.
2. Availability of replacement equipment in case of breakdowns.

The implementation can start with only one production line. The advantages of starting with one line are, new equipment can be tested in a real application, and the weaknesses identified. Unexpected faults can be overcome before the development of other lines starts. The line can be used to train maintenance engineers and operators, so that they become familiar with the new technology and then actively support the development of the other lines. Initial investment by the company is kept to a minimum until tangible benefit can be realised. Discussions with Brohome engineers have shown that production line 34AP suffers many failures, especially in the control and communication cards. For this reason the decision to start with an upgrade of this production line has been made.

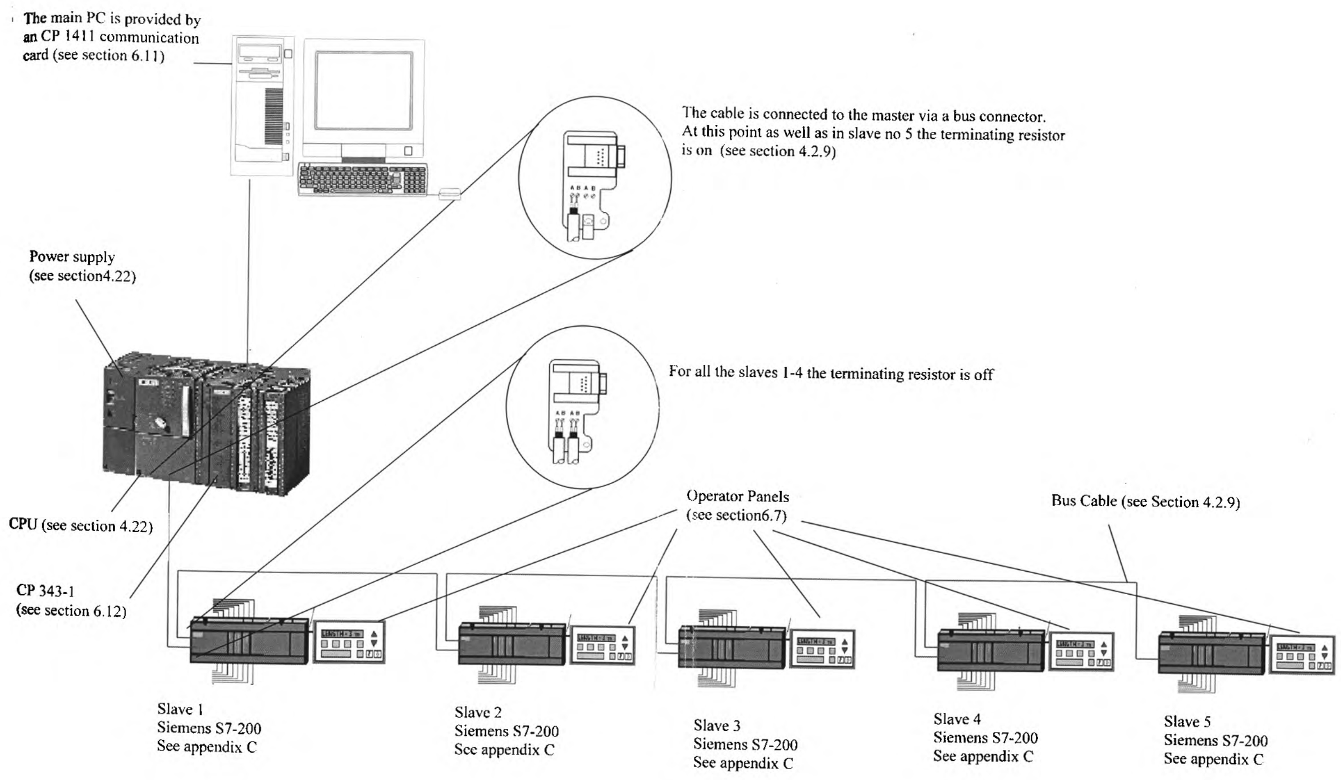


Figure 6.18 Brohome control system

CHAPTER 7

Conclusion

The various international standards organisations have tried and are still trying to provide a single solution to the problem of interconnecting devices at the shop floor level (see figure 7.1). However, each of the currently available solutions has powerful support from various companies each trying to dominate the market and make its preferred system the defacto standard. This is not surprising in view of the size of the market for this technology. In other words reaching a single solution is not difficult theoretically (or technically), but it is very difficult commercially. For a company to wait for that unique solution is not a sensible option, since the potential benefits to be obtained by using fieldbus technology in manufacturing and process control are many and various, and include:

- **Greatly reduced wiring costs**. All devices on the fieldbus network are connected to the main controller via one cable, in addition some fieldbus protocols provide power via the communication lines. Fieldbus networks can be implemented through various wiring topologies, where fieldbus systems allow for greater distances than with traditional networks. Using fieldbus systems there can be a 5:1 reduction in field wiring expenses (Rolf, 1998).
- **Reduced installation and start up time**. Fieldbus networks require less time to install than conventional systems, and fewer system drawings will be needed in order to develop a fieldbus system. Control cabinets in fieldbus systems are smaller. Unlike traditional control systems where all inputs and outputs must be connected to the main controller, in fieldbus systems most of the inputs and outputs can be connected to individual field devices, this means that the implementation can be done in various sites of the plant. In fieldbus systems some of the control operations and intelligence can reside in the individual field devices, reducing the complexity of the main controller, where the user can start different parts in the plant whenever they are ready.
- **Improved line monitoring and diagnostics**. Fieldbus enables the total integration of data from device level to management system. This leads to greater efficient through the

control of production, stock control, warehousing and handling of orders. The operator can easily view all of the devices in the system, so he can discover any problem and carry out maintenance more quickly. Also fieldbus systems allows on line diagnostics to be carried out on individual field devices.

- **Easier changes to the system**. Most vendors provide an interface cards or modules to their devices to be connected to various fieldbus protocols.
- **Improved interoperability between manufacturers equipment**. Fieldbus systems are open systems, which means that the end user can obtain devices from various vendors. That is various companies now provides equipment that can be connected to the same network

The possible areas of application of fieldbus cover a wide range of situations, including automotive systems, environmental control in buildings, manufacturing and process control. It seems highly unlikely that a single fieldbus would be able to meet the requirements of such diverse applications. More realistically, it is reasonable to expect that the very large number of different fieldbuses available at present will be reduced drastically to just a hand-full of preferred options. These fieldbuses should be able to communicate with one-another via well-defined gateways.

In this work various fieldbus protocols are studied **H**ighway **A**ddressable **R**emote **T**ransducer (**HART**), (see section 3.5) is primarily used for **process automation**. HART protocol is now managed by a separate organisation, the HART Communication Foundation (HCF) and is particularly popular in the USA. Commercially available products include a range of slave sensors and both PLC and PC masters.

HART enables groups of up to 15 sensors and actuators to be connected together in series as part of a 4 - 20 mA current loop. A bus master may then individually poll the sensors. 1200 bits/s FSK signalling at frequencies of 1200 Hz and 2400 Hz provides a data communication throughput over ranges of up to 2000 m, dependent on cable type.

The **Controller Area Network** (**CAN**) (see section 3.6) is used for real-time applications. It was originally developed for use in the automobile industry to provide a cost-effective communications bus for in-car electronics and as an alternative to expensive and cumbersome wiring loom CAN defines only Physical and Data Link layers. Many applications of CAN require services that are beyond the basic functionality specified by the Data-Link Layer but which may be implemented at the Application Layer. To meet this need several organisations have developed Application Layers. Such as CAL (CAN Application Layer), CAN open, PCAL (Portable CAN Application Layer), DeviceNet, SDS (Smart Distributed System), and CAN Kingdom. Each of them has its own features, and are incompatible with other CAN based protocol. However **CAN in Automation (CiA),** the international users and manufacturers group founded in March 1992, provides technical, product and marketing information with the aim of fostering CAN's image and providing a path for future developments of the CAN protocol.

Foundation Fieldbus, (see section 3.7), is a communication protocol used in both process and manufacturing automation applications. The technology is controlled by the **Fieldbus Foundation** The aim of fieldbus foundation was to produce H1 for process automation and H2 for discrete automation. H1 has been implemented in various applications while H2 has not yet been issued.

The **Foundation Fieldbus** seems to be developing into an American project appropriated by the main DCS vendors who have never shown much aptitude for producing the common standards that would be useful to industry. Foundation Fieldbus uses the publisher-subscriber approach where each parameter is broadcast at regular intervals onto the bus by the publisher device. 32 device can be linked in segment of 1900m long at speed of 31.25 kbps.

Profibus, (see section 3.8), spans a range of applications in industrial control. **Profibus FMS** is targeted at higher level communication between cells within a plant. It is designed to operate at medium speeds and provide a wide range of functionality and flexibility. **Profibus DP** is a performance - optimised version, specifically dedicated to time - critical communications between automation systems and distributed peripherals. It is targeted at lower level tasks. **Profibus PA** is the newest variant of the Profibus family. It is aimed specifically at the process automation market. Profibus Trade Organisation supports all Profibus DP (Master/Slave), Profibus FMS (Multi-master/Peer to Peer), and Profibus PA (intrinsically safe). Using Profibus DP up to 126 devices can be linked in one network over 1200 metre at speed of 9.6 Kbps.

Actuator Sensor **Interface (AS-Interface)** was developed by a group of European companies and is now promulgated by a separately established organisation, **AS-International**. It is targeted at the very lowest level of the industrial network hierarchy. Specifically, it is optimised for binary sensors and actuators in automation networks. Up to 31 slaves can be connected to AS-Interface network for a distance of 300 metre at speed of 167 Kbps.

A comparison of the features of five fieldbuses has been carried out with the exception of AS-I, which is targeted on a lower level application. As a result Profibus provide a complete solution for all industrial applications through the use of Profibus PA and Profibus DP which can be linked together. Also, Profibus DP is the fastest fieldbus at the present time and it is the most widely installed fieldbus and it will remain so according to market studies until at least the year 2003.

In the opinion of many people it may be possible to modify Ethernet in order for it to take over the functions of fieldbus. The main feature of industrial fieldbus compared with Ethernet is its ability to cope with real-time applications. At the moment, Ethernet uses the CSMA/CD technique, which does not guarantee real-time actions. This means that there is theoretically the possibility that a critical message sent from one station to another may not get to its destination quickly enough due to a collision caused by high traffic levels.

The recent advent of so-called "soft" PLCs is likely to have a significant effect on the adoption of fieldbus systems by industry. Software resident in a PC mimics the control action of a traditional PLC. However, there is still the need to connect field devices to the PC so that inputs can be monitored and outputs controlled. The easiest way in which this can be achieved is to provide the PC with a fieldbus card and to run a fieldbus cable from the PC to remote I/O stations, operator panels and motor drivers, for example. Due to cost considerations, it is anticipated that the use of soft PLCs in industry will grow rapidly over the next few years.

J. Pittman, Fieldbus Foundation
John Berra, Fisher Rosemount
W. Moss, ControlNet International
Steve Eisenbrown, Rockwell Automation
E. Küster, PROFIBUS Nutzerorganisation
Dr. Klaus Wucherer, Siemens AG

To: IEC 65C WG6 and the IEC National Committees

Memorandum of Understanding

Regarding the IEC Committee of Action (CA)
Decision 105/19 + 105/20 on IEC 61158 Part 3 ... 6

After a long period of misunderstandings during the development of the fieldbus specification IEC 61158, we welcome very much the decisions made during the CA meeting of June 15, 1999. The results offer the opportunity to successfully finalize this project and provide the benefits of following an open fieldbus strategy to users and manufacturers. To ensure this goal, additional profiles and protocols will be amended to the current IEC 61158 specification. In addition, profiles will be created within Working Group 1 to map the appropriate services and use of protocol to each network.

We will cooperate on the finalization of the requested specifications to represent consensus and we will do the work necessary in a spirit of mutual trust based on the challenging decisions of the CA.

We, representing many of the key participants in this project, would like to ask you for your support to complete this IEC fieldbus standard. Please approve the vote on the amended IEC 61158 specifications and agree to the open communication strategy.

July 16, 1999

J Pittman, FF | W. Moss, CI | E. Küster, PNO

John Berra, Fisher-Rosemount | Steve Eisenbrown, RA | Klaus Wucherer, SAG

Figure 7.1 Fieldbus strategy

At the present time Profibus, with its three versions FMS, DP, and PA provides a complete solution for most applications. It is leading the market. However, companies installing new equipment must make a judgement as to whether a fieldbus system associated with the project is really appropriate. They must be sure that they have not been influenced by the current interest in fieldbus to the extent that they become the owners of a system that they will have to maintain and fault- find and which may be technically beyond their capabilities. For example, the study undertaken in chapter 6 indicated that there is no need to implement an AS-Interface network in Brohome. Having decided that the installation of a fieldbus would provide significant benefits, there may then be the question of which fieldbus to choose. One of the main aims of this project has been to provide information and guidance in this area, and descriptions of some of the practical aspects of establishing a network have also been included in order to illustrate some of the principles involved.

As discussed, the introduction of any new technology generates a need to train engineers and technicians to use this technology. Training is very expensive but of great importance. The project has acknowledged this by providing a training package for one of the small, modern PLCs. (see appendix C)

The implementation of a Profibus DP network for UWCN's Mechatronic Development Centre laboratory as discussed, will provide a good environment in which to study this technology, and so support local industrial companies. The Centre will be able to advise companies on whether they need fieldbus or not, and which equipment would be most suitable for their application. Training programmes to explain the operation of the technology and to practice maintenance and trouble-shooting procedures will also be available.

The emphasis on products from Siemens throughout this project does not necessarily mean that they are the best or the cheapest. The reason lies in the fact that technical support is available at UWCN. It is quite likely that at a later stage, equipment from other vendors will be added to the system.

Although, fieldbus systems have been implemented in number of industrial areas, most of the available information dealing wit this technology can be one of two types, standards and company brochures.

The standards, which describe the different systems tend to be technical documents, and are targeted at specialists. For example, EN 50 170 standard is more than 2000 pages long and is very complicated for technicians or top managers who have no technical background to understand the terms. Another type of document which discusses fieldbus technology is provided by companies to support their products. However, these contain no theoretical explanation of the operation of their systems. In conclusion, technicians, engineers, top managers and decision-makers needs enough documentation that allows them to make informed decision. These documents should be bridge the gap between the standards and company brochures, explaining the new technology in simple terms that people in industry can understand.

The work in this thesis considers this point and provides this document, which describes fieldbus technology in practical terms, giving examples of how to implement different networks for different purposes. Also, the basic principles that are needed to understand the subject are explained in chapter two. As a result the reader is provided with the necessary background on fieldbus technology, which enables him to implement networks and to make the right decisions in choosing suitable fieldbus solution according to his requirements.

Industrial development and the adoption of new automation technology depends on the presentation of information and knowledge in the best possible way.

List of Acronyms

ASCII	American Standard Code of Information Interchange
AS-Interface	Actuator Sensor-Interface
BCC	Block Check Character
BCD	Binary Coded Decimal
CAN	Controller Area Network.
CIM	Computer Integrated Manufacturing
CSMA/CD	Carrier Sense Multiple Access with Collision Detection.
DD	Device Description
DDL	Device Description Language
DDLM	Direct Data Link Mapper
DLL	Data Link Layer.
DP	Decentralised Periphery
EBCDIC	Extended Binary Coded Decimal Interchanging Code
EIA	Electronic Industrial Association
EPA	Enhanced Performance Architecture
FB	Function Block
FMS	Fieldbus Message Specification
FSK	Frequency Shift Keying
GSD	Device Description File
HART	Highway Addressable Remote Transducer
IEC	International Electromechanical Commission
IEEE	International Electrical and Electronic Engineering
ISO	International Standard Organisation
ISP	Interpretable system Project
LSB	Least Significant Bit
LAS	Link Active Scheduler
LRC	Longitudinal Redundancy Checking
LLI	Lower Layer Interface

LAN	Local Area Network
MAP	Manufacturing Automation Protocol
MMS	Manufacturing Message Specification
MMFS	Manufacturing Message Format Standard
MSB	Most Signification Bit
NRZ	Non Return to Zero
OD	Object Dictionary
OSI	Open System Interconnection
PA	Process Automation
PLC	Programmable Logic Controller
Profibus	Process fieldbus
RB	Resource Block
RS	Recommended Standard
SDS	Smart Distribution System
SME	Society of Manufacturing Engineering
TB	Transducer Block
VCR	Virtual Communication Relationship
VFD	Virtual Field Device
VMD	Virtual Manufacturing Device
UWCN	University of Wales College Newport

References

1. A. J. Crispin 1997. *Progammable logic controller and their Engineering application.* England: McGraw Hill. 0077093178.

2. A. Leach 1994. Profibus : the German fieldbus standard. *Assemply Automation,* **14** (1), pp. 8-12.

3. A. McFarlane 1997. Fieldbus review. *Sensor Review,* **17** (3), pp. 204-210.

4. A.R. Dewey 1996. *FIELDBUS - HOW WILL I BE AFFECTED.* Energy week conference. 1996, Texas, USA.

5. A. S. Tanenbaum 1996. *Computer Networks.* 3rd ènd. Prentice- Hall, Inc. 0-13-394248-1.

6. A. T. Bradshow, Fieldbus Foundation or Foundation Fieldbus?, <http://www.iac.honywell.com>, 1/4/1997, (2/3/1999).

7. ARC Advisory Group, Popularity of HART Rveals the Hidden fieldbus Market, <http://www.arcweb.com>, JAN 1999, (AUG 1999).

8. AS-Interface UK, 1996. *Newsletter.*

9. AS-International Association, 1998. *FAQ.*

10. D. P. Lane 3/11/1997. Sensor Gateway to Fieldbus. *Sensor Review,* pp. 211-216.

11. EEMUA 1997. *A Guide to fieldbus application for the process industry.* The Engineering Equipment and Material User Association. 085931 0663.

12. Fieldbus Foundation Autumn 1998. DElivering the world's fieldbus The promis fulfiled. *Process Products,* pp. 6-9.

13. G. Held 1997. *Understanding data communication*. 4th end. USA: Prentice-Hall. 0-672-30501-1.

14. G. Olsson , & G. Piani 1992. *Computer Systems for Automation and Control.* U.K: Prentice Hall International. 0-13-457581-4.

15. G. Waters 1991. *Computer Communication Networks.* U. K: McGraw-Hill Book Company (UK) Limited. 0-07-707325-8.

16. Hirschmann. *AS-Interface*. Germany: Richard Hirschmann Gmbh & Co., 1997. 145355915

17. J. Brignell , & N. White 1996. *Intelligent Seneor systems.* U.K: Institute of Physics Publishing. 0750303891.

18. J. Fitzgerald , & A. Dennis 1996. *Business data communication and networking.* Canada: John Wily &sons, Inc. 0-471-12365.

19. J. Quinn 1995. *Digital Data Communication.* U S A: Prentic Hall Career & Technology. 0023972408.

20. J. R. Jordan 1995. *Serial Network Field Instrumentation* . 1 end. England: John Wiley & Sons Ltd. 0 471 953261.

21. J. R. Pimentel 1990. *Communication Networks for Manufacturing* . 1 end. USA: Prentice Hall. 0 13 168576 7.

22. J. Stenerson 1993. *Foundamentals of Programmable Logic Controllers, sensors & communication.* U.S.A: Regents / Prentic Hall. 013-726860-2.

23. K. Dimyati. "Real-time Clock Synchronisatin over a CAN-based fieldbus." wales, 1996.

24. L. Mackenzie 1998. *Communication and Networks.* 1st end. UK: McGraw-Hill Publishing Company.

25. M.G. Rodd/F. Deravi 1989. *Communication System for Industrial Automation.* 1 end. UK: Prentice Hall International. 0 13 152372 4.

26. M. G. Rodd,K. Dimyati , & L. Mtus 1998. The Design and analysis of low cost

real-time fieldbus systems. *Cntrol Engineering Practice,* **6**, pp. 83-91.

27. M.M. Marcos et la. 1999 Jul. *An integrated framework for development of real-time distributed control software based on CAN bus.* 14th World Congress of the International Federation of Automatic Control (IFAC) . 1999 Jul 5-1999 Jul 9, China.

28. M. Ochsner and M. Schrier, To fieldbus or not to fieldbus, <http://www.iac.honywell.com>, 24/1/1998, (2/11/1998).

29. M. Soufian. "Profibus Application: Introduction to fieldbus systems."Dec. 1998, *The UK Profibus competence centre,* The Manchester Metropolitian University.

30. M. T. Hoske 1998. Connect to the benefits of digital industrial networks. *Control Engineering,* (January)

31. N. P. Mahalik , & P. R. Moore 1997. Fieldbus technology based, distributed control in process industries: a case study with LonWorks technology. *Integrated Manufacturing system,* **8** (4), pp. 231-243.

32. P. G. Ranky 1990. *Computer Networks for World Class CIM Systems.* U.K: CIMware Limited. 1-872631-01-0.

33. P. Stude baker 1997. Fieldbus: Reaching the promised LAN. *Control Magazine,* **4**

34. R. Helson July 1998. The HART Protocol- A Solution Enabling Technology. *The HART Book,* (Issue 8), pp. 4-8.

35. Robert Bosch GmbH,.1991. *CAN Specification* . Version 2.0 ed.

36. Rolf, Advantages of fieldbus, <http://rolf.ece.curtine.edu.au>, 24/1/1998, (29/3/1999).

37. Schickhuber, G, & Mccarthy, O Feb 1997. Distributed fieldbus and control

network systems. *Computing and control engineering,* pp. 21-32.

38. Siemens, 1996. "The AS interface system." *Actuator Sensor Interface, System Description.* Siemens Aktiengesellschaft.

39. Siemens. *Components for Automation & Drivers*. UK 10/98. Siemens AG, 1998. CA 01

40. T.Karppinen, Industrial communication and fieldbus systems, <http://www2.hamkk.fi/karppinen/english/fieldb4.html>, 22/10/1998, (16/11/1998).

41. U. Black 1993. *Computer Networks Protocol, standards, and Interface.* 2nd end. USA: Prentice-Hall, Inc. 013-090861-4.

42. W. A. Shay 1988. *understanding data communcation and networking.* 053495054X.

43. W. R. Hodson, Exploting Non-control functions in field device with fieldbus, <http://www.iac.honywell.com>, 18/11/1998).

44. W. Stalling 1997. *data and computer communication.* 5th end. USA: Prentice-Hall, Inc. 0-13-571274-2.

45. X. Jin. "An Investigation into the Requirements of A Real-Time Fieldbus Messaging service." Wales , 1994.

Appendix A

Operation Sequences of

Mechatronic Development Centre Laboratory

Appendix A operation sequences of Mechatronic Development Centre Laboratory

Step	User action	Main controller	Conveyor	Unit 1 pneumatic	Unit 2 FMS	Unit 3 process	Unit 4 vision	Unit 5 Hydraulic	Unit 6 bar code reader	Unit 7 storage
1	Power ON	ON	ON	ON	ON	ON	ON	ON	ON	ON
2	–	check safety, emergency stop	–	–	–	–	–	–	–	–
3	–	check the stations ready or not	check the power and report to the main controller	check power, air pressure, raw materials	robot ready, lathe, mill	power, tank levels, temperature value	power, reference images available	power, oil level in reservoir	ready to read or not	available space
4	–	receive the station's reports and define the faults to the user	wait for the commands	report to main controller and wait for commands	report to main controller and wait for commands	report to main controller and wait for commands	report to main controller and wait for commands	report to main controller and wait for commands	report to main controller and wait for commands	report to main controller and wait for commands
5	correct any faults	–	–	–	–	–	–	–	–	–
6		if no faults, system ready message	–	–	–	–	–	–	–	–
7	select desired operation from menu	necessary instructions issued to stations	–	–	–	–	–	–	–	–
8	–	–	read the wagon numbers and locations and report to main controller	–	–	–	–	–	–	–

Appendix A Continuous

Step	User action	Main controller	Conveyor	Unit 1 pneumatic	Unit 2 FMS	Unit 3 process	Unit 4 vision	Unit 5 Hydraulic	Unit 6 bar code reader	Unit 7 storage
9	–	send a wagon to station No 1	execute order and report to main that the wagon is on position	–	–	–	–	–	–	–
10		order station No1 to pick up a carrier and a cylinder		–	–	–	–	–	–	–
11	–	–	report to main controller and waiting for the command	execute order and report to the main controller	–	–	–	–	–	–
12	–	order conveyor to carry the wagon to station No 2 and to bring another wagon to station No1		report to main controller and waiting for the command	–	–	–	–	–	–
13	–	–	execute the order of step 12	–	–	–	–	–	–	–
14	–	when wagon No1 reaches station No 2, order the station to execute a defined action and when a wagon No 2 reach station No1 ask for a block	report to main controller and wait for command	–	–	–	–	–	–	–

Appendix A Continuous

Step	User action	Main controller	Conveyor	Unit 1 pneumatic	Unit 2 FMS	Unit 3 process	Unit 4 vision	Unit 5 Hydraulic	Unit 6 bar code reader	Unit 7 storage
15	–	–	–	Pick up a carrier and transfer a block to the second wagon, report to main controller	execute action and report to main controller	–	–	–	–	–
16	–	–	report to the main controller that wagon No 1 is empty and carry wagon No 2 to station No2	report to main controller and waiting for the command	–	–	–	–	–	–
17	–	order station No 2 to pickup the block and to execute a manufacturing process	report to main controller and wait for commands	–	–	–	–	–	–	–
18	–	–	report to the main controller that wagon No 2 is empty	–	–	–	–	–	–	–
19	–	decide to move the wagon or to wait for the completion of one of manufacturing processes	–	–	–	–	–	–	–	–

Appendix A Continuous

Step	User action	Main controller	Conveyor	Unit 1 pneumatic	Unit 2 FMS	Unit 3 process	Unit 4 vision	Unit 5 Hydraulic	Unit 6 bar code reader	Unit 7 storage
20	–	–		report to main controller of the available row material	report to the main if any of manufacturing is finished	–	–	–	–	–
21	–	decide to send the cylinders to process station and the block to the vision station and to order a new raw material from station No 1	report to main controller and wait for commands	report to main controller and wait for command	–	–	–	–	–	–
22	–	–	–	load wagon No 3 by the ordered raw material	load the wagon with finished material	–	–	–	–	–
23	–	–	carry the manufactured material to the designation station and bring the raw material from the station No 1 to station No 2	report to main controller and waiting for the command	report to main controller and wait for command	–	–	–	–	–

Appendix A Continuous

Step	User action	Main controller	Conveyor	Unit 1 pneumatic	Unit 2 FMS	Unit 3 process	Unit 4 vision	Unit 5 Hydraulic	Unit 6 bar code reader	Unit 7 storage
24	–	order the process station to start processing \(in case of cylinders) or order the vision station to compare	report to controller if any wagon is empty	–	–	–	–	–	–	–
25	–	–	report to main controller and wait for command	report to main on available material and feed the wagon according to the main's order	report to the main if the lath or mill ready for the next work	if wagon carrying a cylinder arrives at this station, pick up and start processing	if wagon carrying a block arrives at robot, pick up, put in the measuring position. vision system measure it and report result to master	–	–	–
26	–	–	report to main which wagon is empty and the location of the different wagons	–	–	–	–	–	–	–

Appendix A Continuous

Step	User action	Main controller	Conveyor	Unit 1 pneumatic	Unit 2 FMS	Unit 3 process	Unit 4 vision	Unit 5 Hydraulic	Unit 6 bar code reader	Unit 7 storage
27	–	Read the reports of all the stations. For example, if station No2 finishes a cylinder and process station is bussy it may sent to store, or if the result from vision system is acceptable, item can be sent to hydraulic station	report to main controller and waiting for the command	–	–	–	–	–	–	–
28	–	–	carry wagons according to the main's order	report to main the available material and feed the system with a new raw materials	execute the main's orders and report the result to the main controller	execute the processing and report to the main controller	take measurements and report to the main controller	assemble the products and report to the main controller	read the bar code and report to the main controller	receive carriers, store them and report to the main controller of location and available space
29	–	Continue to follow previous steps and on request inform user of the state of the process	report to main controller and wait commands	report to main controller and wait commands	report to main controller and wait commands	report to main controller and wait commands	report to main controller and wait commands	report to main controller and wait commands	report to main controller and wait commands	report to main controller and wait commands

Appendix B

Cost of Profibus DP and AS-Interface networks

For Mechatronic Development Centre Laboratory

AND

Cost of Profibus DP

For Brohome Ltd.

Part list for project:PROFIBUS DP CIM

Part list for station:FMS STATION

Order number	Designation	Number	Unit price	Total price ($)
6ES7153-1AA02-0XB0	SIMATIC S7-300, ET 200 M, DISTRIBUTED I/O SYSTEM FOR MAX. 8 S7-300 MODULES	1	195.00	195.00
6ES7972-0BA11-0XA0	SIMATIC S5/S7,BUS CONNECTOR FOR PROFIBUS UP TO 12 MBIT/S 90 DEGREE ANGLE OUTGOING CABLE, TERMINAT. RESIST. WITH ISOLAT. FUNCTION, WITHOUT PG SOCKET	1	29.00	29.00
6ES7321-1BH01-0AA0	SIMATIC S7-300, DIGITAL INPUT SM 321, OPTICALLY ISOLATED 16DI, 24 V DC	1	90.00	90.00
6ES7322-1BH01-0AA0	SIMATIC S7-300, DIGITAL OUTPUT SM 322, OPTICALLY ISOLATED 16DO, 24 V DC, 0.5A	1	120.00	120.00
6ES7390-1AB60-0AA0	SIMATIC S7-300, RAIL L=160MM	1	12.00	12.00
Station value:				**446.00**

Part list for station:HYDRAULIC STATION

Order number	Designation	Number	Unit price	Total price ($)
6ES7215-2AD00-0XB0	SIMATIC S7-200, CPU 215 COMPACT UNIT, DC POWER SUPPLY 4K PRG/2,5K DW, 14DI DC/10DQ DC PROFIBUS-DP INTERFACE EAN: 4025515160151	1	399.00	399.00
6ES7291-8BA00-0XA0	SIMATIC S7-200, BATTERY MODULE F. LONG-TERM BACKUP OF DATA PLUGGABLE IN VERTICAL TRUNKING OF MEMORY SUBMODULE OF CPU 214 EAN: 4025515160748	1	19.00	19.00
6ES7298-8AE00-8YE0	SIMATIC S7-200, DOCUMENTATION ON CD-ROM, 5 LANGUAGES, INCL. HMI MANUALS TD 200 STEP7 MICRO DOS/WIN WITH ADOBE ACROBAT READER	1	20.00	20.00
6ES7298-8FA01-8BH0	SIMATIC S7-200, SYSTEM MANUAL HARDWARE A. SOFTWARE MANUAL FOR S7-200, ENGLISH EAN: 4025515161097	1	11.00	11.00
6ES7298-8CA00-8BH0	SIMATIC S7-200, OPERATIONS LIST (10 PIECES), ENGLISH EAN: 4025515160472	1	3.00	3.00
6ES7972-0BA11-0XA0	SIMATIC S5/S7,BUS CONNECTOR FOR PROFIBUS UP TO 12 MBIT/S 90 DEGREE ANGLE OUTGOING CABLE, TERMINAT. RESIST. WITH ISOLAT. FUNCTION, WITHOUT PG SOCKET	1	29.00	29.00
Station value:				**481.00**

Part list for station:MASTER

Order number	Designation	Number	Unit price	Total price ($)
6ES7315-2AF02-0AB0	SIMATIC S7-300, CPU 315-DP CPU WITH INTEGRATED 24 V DC POWER SUPPLY, 64 KBYTE WORKING MEMORY 2ND INTERFACE DP-MASTER/SLAVE	1	1,025.00	1,025.00
6ES7951-0KD00-0AA0	SIMATIC S7, MEMORY CARD MC 951, F. S7-300,SHORT VERSION 5V FLASH EPROM, 16 KBYTES	1	56.00	56.00
6ES7971-1AA00-0AA0	SIMATIC S7, BACKUP BATTERY (LI) 3,4V/1AH, FOR S7-300 (CPU 313, 314, 315) AND S5-90U	1	10.00	10.00
6ES7972-0BA11-0XA0	SIMATIC S5/S7,BUS CONNECTOR FOR PROFIBUS UP TO 12 MBIT/S 90 DEGREE ANGLE OUTGOING CABLE, TERMINAT. RESIST. WITH ISOLAT. FUNCTION, WITHOUT PG SOCKET	1	29.00	29.00
6ES7398-8AE00-8YE0	SIMATIC S7-300, DOCUMENTATION ON CD-ROM, 5 LANGUAGES, INCL. FM/CP/OP/TD, SIMATIC NET SIMATIC HMI, SIMATIC M7 WITH ADOBE ACROBAT READER	1	48.00	48.00
6ES7998-8XC00-8YE0	SIMATIC S7, REFERENCE DOCUMENT. ON CD-ROM FOR SIMATIC S7/M7/C7 INCL. OP, TD, FM, CP, NCM, 5 LANGUAGES (GER/EN/FR/SP/IT) CONTINUOUS UPDATES - PRICE APPLICABLE FOR ONE YEAR	1	199.00	199.00
6ES7321-1BH01-0AA0	SIMATIC S7-300, DIGITAL INPUT SM 321, OPTICALLY ISOLATED 16DI, 24 V DC	1	90.00	90.00
6ES7322-1BH01-0AA0	SIMATIC S7-300, DIGITAL OUTPUT SM 322, OPTICALLY ISOLATED 16DO, 24 V DC, 0.5A	1	120.00	120.00
6ES7392-1AJ00-0AA0	SIMATIC S7-300, FRONT CONNECTOR FOR SIGNAL MODULES WITH SCREW CONTACTS, 20-PIN	2	16.00	32.00
6ES7972-0BA11-0XA0	SIMATIC S5/S7,BUS CONNECTOR FOR PROFIBUS UP TO 12 MBIT/S 90 DEGREE ANGLE OUTGOING CABLE, TERMINAT. RESIST. WITH ISOLAT. FUNCTION, WITHOUT PG SOCKET	1	29.00	29.00
6XV1830-0AN30	SIMATIC NET, LAN CABLE FOR PROFIBUS TWO-CORE, SHIELDED 30 M	1	30.00	30.00
6GK1970-5CA10-0AA1	SIMATIC NET, MANUAL FOR FOR PROFIBUS NETWORKS, NETWORK ARCHITECTURE,COMPONENTS CONFIGURATION,INS. INSTRUCTIONS IN ENGLISH	1	39.00	39.00
6ES7307-1BA00-0AA0	SIMATIC S7-300, STABILIZED POWER SUPPLY, PS 307, 120/230 V AC; 24 V DC, 2 A	1	81.00	81.00

Part list for project:PROFIBUS DP CIM

Order number	Designation	Numb er	Unit price	Total price (£)
6ES7390-1AE80-0AA0	SIMATIC S7-300, RAIL L=480MM	1	19.00	19.00
Station value:				**1,807.00**

Part list for station:PNEUMATIC STATION

6ES7153-1AA02-0XB0	SIMATIC S7-300, ET 200 M, DISTRIBUTED I/O SYSTEM FOR MAX. 8 S7-300 MODULES	1	195.00	195.00
6ES7972-0BA11-0XA0	SIMATIC S5/S7,BUS CONNECTOR FOR PROFIBUS UP TO 12 MBIT/S 90 DEGREE ANGLE OUTGOING CABLE, TERMINAT. RESIST. WITH ISOLAT. FUNCTION, WITHOUT PG SOCKET	1	29.00	29.00
6ES7321-1BH01-0AA0	SIMATIC S7-300, DIGITAL INPUT SM 321, OPTICALLY ISOLATED 16DI, 24 V DC	3	90.00	270.00
6ES7392-1AJ00-0AA0	SIMATIC S7-300, FRONT CONNECTOR FOR SIGNAL MODULES WITH SCREW CONTACTS, 20-PIN	3	16.00	48.00
6ES7322-1BH01-0AA0	SIMATIC S7-300, DIGITAL OUTPUT SM 322, OPTICALLY ISOLATED 16DO, 24 V DC, 0.5A	2	120.00	240.00
6ES7392-1AJ00-0AA0	SIMATIC S7-300, FRONT CONNECTOR FOR SIGNAL MODULES WITH SCREW CONTACTS, 20-PIN	2	16.00	32.00
6ES7390-1AE80-0AA0	SIMATIC S7-300, RAIL L=480MM	1	19.00	19.00
6ES7153-1AA00-8BA0	SIMATIC S7, MANUAL FOR ET 200 M DISTRIBUTED I/O DEVICE, ENGLISH	1	11.00	11.00
Station value:				**844.00**

Part list for station:PROCESS CONTROLLER

6ES7315-2AF02-0AB0	SIMATIC S7-300, CPU 315-DP CPU WITH INTEGRATED 24 V DC POWER SUPPLY, 64 KBYTE WORKING MEMORY 2ND INTERFACE DP-MASTER/SLAVE	1	1,025.00	1,025.00
6ES7971-1AA00-0AA0	SIMATIC S7, BACKUP BATTERY (LI) 3,4V/1AH, FOR S7-300 (CPU 313, 314, 315) AND S5-90U	1	10.00	10.00
6ES7951-0KD00-0AA0	SIMATIC S7, MEMORY CARD MC 951, F. S7-300,SHORT VERSION 5V FLASH EPROM, 16 KBYTES	1	56.00	56.00
6ES7972-0BA11-0XA0	SIMATIC S5/S7,BUS CONNECTOR FOR PROFIBUS UP TO 12 MBIT/S 90 DEGREE ANGLE OUTGOING CABLE, TERMINAT. RESIST. WITH ISOLAT. FUNCTION, WITHOUT PG SOCKET	1	29.00	29.00
6ES7321-1BH01-0AA0	SIMATIC S7-300, DIGITAL INPUT SM 321, OPTICALLY ISOLATED 16DI, 24 V DC	3	90.00	270.00
6ES7392-1AJ00-0AA0	SIMATIC S7-300, FRONT CONNECTOR FOR SIGNAL MODULES WITH SCREW CONTACTS, 20-PIN	3	16.00	48.00
6ES7322-1BH01-0AA0	SIMATIC S7-300, DIGITAL OUTPUT SM 322, OPTICALLY ISOLATED 16DO, 24 V DC, 0.5A	2	120.00	240.00
6ES7392-1AJ00-0AA0	SIMATIC S7-300, FRONT CONNECTOR FOR SIGNAL MODULES WITH SCREW CONTACTS, 20-PIN	2	16.00	32.00
6ES7331-7KF01-0AB0	SIMATIC S7-300, ANALOG INPUT SM 331, OPTICALLY ISOLATED U/I/THERMOCOUPLE/RESISTANCE INTERRUPT, DIAGNOSTICS; RESOLUTION 9/12/14 BITS, 8AI REMOVE/INSERT W. BACKPLANE BUS	1	440.00	440.00
6ES7392-1AJ00-0AA0	SIMATIC S7-300, FRONT CONNECTOR FOR SIGNAL MODULES WITH SCREW CONTACTS, 20-PIN	1	16.00	16.00
6ES7332-5HD01-0AB0	SIMATIC S7-300, ANALOG OUTPUT SM 332, OPTICALLY ISOLATED U/I; DIAGNOSTICS; RESOLUTION 11/12 BITS 4AO, REMOVE/INSERT W. ACTIVE BACKPLANE BUS	1	336.00	336.00
6ES7392-1AJ00-0AA0	SIMATIC S7-300, FRONT CONNECTOR FOR SIGNAL MODULES WITH SCREW CONTACTS, 20-PIN	1	16.00	16.00
6ES7390-1AF30-0AA0	SIMATIC S7-300, RAIL L=530MM	1	23.00	23.00
Station value:				**2,541.00**

Part list for station:PROGRAMMING DEVICE

6ES7741-0AB00-0YA0	SIMATIC PG, PROGRAMMER PG740 PII, 13.3" TFT XGA, PENTIUM II 266,64MB RAM,CD-ROM, 5.2 GB HD, WIN95, STEP5 V7.01, STEP7 V4.02, MICRO/WIN V2.1	1	5,281.00	5,281.00
6ES7810-4CA03-8BA0	SIMATIC S7, STEP7, DOCUMENT. BASIC KNOWLEDGE, COMPRISES BEGINNER'S PRIMER, MANUAL PROGRAMM. MANUAL,CONVERTER MAN. FOR V4.0, ENGLISH	1	36.00	36.00
6ES7810-4CA02-8BR0	SIMATIC S7, STEP7, DOCUMENT. REFERENCE MANUAL, COMPRISES: MANUAL FOR STL/LAD/CSF REFERENCE MANUAL S7-300/400 FOR V3.X/V4.X, ENGLISH	1	65.00	65.00
Station value:				**5,382.00**

Part list for project:PROFIBUS DP CIM

Part list for station:STORAGE STATION

6ES7153-1AA02-0XB0	SIMATIC S7-300, ET 200 M, DISTRIBUTED I/O SYSTEM FOR MAX. 8 S7-300 MODULES	1	195.00	195.00
6ES7972-0BA11-0XA0	SIMATIC S5/S7,BUS CONNECTOR FOR PROFIBUS UP TO 12 MBIT/S 90 DEGREE ANGLE OUTGOING CABLE, TERMINAT. RESIST. WITH ISOLAT. FUNCTION, WITHOUT PG SOCKET	1	29.00	29.00
6ES7321-1BH01-0AA0	SIMATIC S7-300, DIGITAL INPUT SM 321, OPTICALLY ISOLATED 16DI, 24 V DC	2	90.00	180.00
6ES7392-1AJ00-0AA0	SIMATIC S7-300, FRONT CONNECTOR FOR SIGNAL MODULES WITH SCREW CONTACTS, 20-PIN	2	16.00	32.00
6ES7322-1BH01-0AA0	SIMATIC S7-300, DIGITAL OUTPUT SM 322, OPTICALLY ISOLATED 16DO, 24 V DC, 0.5A	2	120.00	240.00
6ES7392-1AJ00-0AA0	SIMATIC S7-300, FRONT CONNECTOR FOR SIGNAL MODULES WITH SCREW CONTACTS, 20-PIN	2	16.00	32.00
6ES7390-1AE80-0AA0	SIMATIC S7-300, RAIL L=480MM	1	19.00	19.00
Station value:				**727.00**

Part list for station:VISION + ROBOT

6ES7153-1AA02-0XB0	SIMATIC S7-300, ET 200 M, DISTRIBUTED I/O SYSTEM FOR MAX. 8 S7-300 MODULES	1	195.00	195.00
6ES7972-0BA11-0XA0	SIMATIC S5/S7,BUS CONNECTOR FOR PROFIBUS UP TO 12 MBIT/S 90 DEGREE ANGLE OUTGOING CABLE, TERMINAT. RESIST. WITH ISOLAT. FUNCTION, WITHOUT PG SOCKET	1	29.00	29.00
6ES7321-1BH01-0AA0	SIMATIC S7-300, DIGITAL INPUT SM 321, OPTICALLY ISOLATED 16DI, 24 V DC	2	90.00	180.00
6ES7322-1BH01-0AA0	SIMATIC S7-300, DIGITAL OUTPUT SM 322, OPTICALLY ISOLATED 16DO, 24 V DC, 0.5A	2	120.00	240.00
6ES7392-1AJ00-0AA0	SIMATIC S7-300, FRONT CONNECTOR FOR SIGNAL MODULES WITH SCREW CONTACTS, 20-PIN	4	16.00	64.00
6ES7390-1AF30-0AA0	SIMATIC S7-300, RAIL L=530MM	1	23.00	23.00
Station value:				**731.00**

Total value:				**12,959.00**

Part list for station:AS-Interface

Order number	Designation	Number	Unit price	Total price ($)
3RX9010-0AA00	AS-I CABLE,TRAPEZOIDAL YELLOW, RUBBER 2X1.5 QMM,100M 1 PIECE = 100M	1	112.94	112.94
6GK7342-2AH01-0XA0	SIMATIC NET, CP 342-2 COMMUNICATION PROCESSOR FOR CONNECTION OF SIMATIC S7-300 TO AS-INTERFACE (W/O FRONT CONNECTOR)	1	.00	.00
6ES7392-1AJ00-0AA0	SIMATIC S7-300, FRONT CONNECTOR FOR SIGNAL MODULES WITH SCREW CONTACTS, 20-PIN	1	16.00	16.00
6GK7342-2AH00-8BA0	SIMATIC NET, CP 342-2 MANUAL INCLUDING SW (FC) A. EXAMPLES ENGLISH	1	30.00	30.00
3RG9001-0AG00	ACTUATOR SENSOR INTERFACE 4I APPLICATION MODULE 4X1 INPUT 200 MA, PNP 4X M12 SOCKETS, EMI	12	71.44	857.28
3RG9010-0AA00	AS-INTERFACE FK-INTERFACE MODULE FOR 2 AS-I LINES, EMI	12	5.78	69.36
3RX9300-0AA00	AS-INTERFACE POWER SUPPLY UNIT WITH DATA DECOUPLING 2.4A;SIMATIC-DESIGN INPUT VOLTAGE AC230V	1	223.77	223.77
3RX9400-0AA00	AS-INTERFACE, ADDRESSING UNIT NO SERVICE FUNCTIONS WITH CHARGING UNIT AND ADAPTER	1	223.24	223.24
3RK1901-3HA01	ADRESSING LINE FOR AS-INTERFACE	1	10.40	10.40
3RX9801-0AA00	AS-I TRANSITIONAS-I LINE ON M12 SOCKET	36	7.14	257.04
3RX9802-0AA00	AS-I CAP PLUGS FOR UNUSED M12 SOCKETS OF THE APPLICATION MODULES 1 UNIT = 10 ITEMS (PIECES)	12	5.46	65.52
Total value:				**1,865.55**

Part list for project:brohome

Order number	Designation	Number	Unit price	Total price ($)

Part list for station:plc

6ES7215-2AD00-0XB0	SIMATIC S7-200, CPU 215 COMPACT UNIT, DC POWER SUPPLY 4K PRG/2,5K DW, 14DI DC/10DQ DC PROFIBUS-DP INTERFACE EAN: 4025515160151	1	399.00	399.00
6ES7291-8BA00-0XA0	SIMATIC S7-200, BATTERY MODULE F. LONG-TERM BACKUP OF DATA PLUGGABLE IN VERTICAL TRUNKING OF MEMORY SUBMODULE OF CPU 214 EAN: 4025515160748	1	19.00	19.00
6ES7972-0BA11-0XA0	SIMATIC S5/S7,BUS CONNECTOR FOR PROFIBUS UP TO 12 MBIT/S 90 DEGREE ANGLE OUTGOING CABLE, TERMINAT. RESIST. WITH ISOLAT. FUNCTION, WITHOUT PG SOCKET	1	29.00	29.00
6AV3017-1NE30-0AX0	TEXT DISPLAY TD17 LC DISPLAY, BACK-LIT LED, 4 LINES X 20 CHARACTERS FOR SIMATIC S5 ,505 ,S7 ,M7 NATIVE DRIVERS, PROFIBUS-DP12	1	527.00	527.00
6ES7901-0BF00-0AA0	SIMATIC S7, MPI CABLE FOR CONNECTING SIMATIC S7 AND PG VIA MPI 5M	1	24.00	24.00
6AV6580-3AX05-0BB0	CONFIGURING SOFTWARE PROTOOL/LITE V5.01, MULTILING. CD ROM, SINGLE LICENSE INSTRUCTIONS IN ENGLISH	1	265.00	265.00
Station value:				**1,263.00**

Part list for station:S7-300

6ES7315-2AF02-0AB0	SIMATIC S7-300, CPU 315-DP CPU WITH INTEGRATED 24 V DC POWER SUPPLY, 64 KBYTE WORKING MEMORY 2ND INTERFACE DP-MASTER/SLAVE	1	1,025.00	1,025.00
6ES7971-1AA00-0AA0	SIMATIC S7, BACKUP BATTERY (LI) 3,4V/1AH, FOR S7-300 (CPU 313, 314, 315) AND S5-90U	1	10.00	10.00
6ES7972-0BA11-0XA0	SIMATIC S5/S7,BUS CONNECTOR FOR PROFIBUS UP TO 12 MBIT/S 90 DEGREE ANGLE OUTGOING CABLE, TERMINAT. RESIST. WITH ISOLAT. FUNCTION, WITHOUT PG SOCKET	1	29.00	29.00
6ES7321-1BH01-0AA0	SIMATIC S7-300, DIGITAL INPUT SM 321, OPTICALLY ISOLATED 16DI, 24 V DC	1	90.00	90.00
6ES7392-1AJ00-0AA0	SIMATIC S7-300, FRONT CONNECTOR FOR SIGNAL MODULES WITH SCREW CONTACTS, 20-PIN	1	16.00	16.00
6ES7322-1BH01-0AA0	SIMATIC S7-300, DIGITAL OUTPUT SM 322, OPTICALLY ISOLATED 16DO, 24 V DC, 0.5A	1	120.00	120.00
6ES7392-1AJ00-0AA0	SIMATIC S7-300, FRONT CONNECTOR FOR SIGNAL MODULES WITH SCREW CONTACTS, 20-PIN	1	16.00	16.00
6GK7343-1BA00-0XE0	SIMATIC NET, CP 343-1 COMMUNICATIONS PROCESSOR FOR CONNECTING SIMATIC S7-300 TO INDUSTRIAL ETHERNET, FOR SEND/RECEIVE INTERFACE AND S7 FUNCTIONS	1	1,023.00	1,023.00
6ES5727-0AA11	SIMATIC NET, BUS CABLE 727-0 FOR INDUSTRIAL ETHERNET W/O COAXIAL CONNECTOR SOLD BY THE METER	1	8.00	8.00
6ES7307-1BA00-0AA0	SIMATIC S7-300, STABILIZED POWER SUPPLY, PS 307, 120/230 V AC; 24 V DC, 2 A	1	81.00	81.00
6ES7390-1AE80-0AA0	SIMATIC S7-300, RAIL L=480MM	1	19.00	19.00
6AV3607-1JC20-0AX1	OPERATOR PANEL OP7/DP LC-DISPLAY, LED ILLUMINATED 4-LINES, 20 CHARAC./ LINE, SIMATIC S7, M7, PROFIBUS-DP 1,5 MBAUD UL/CSA-LISTED	1	421.00	421.00
6GK1500-0EA00	SIMATIC NET, LAN CONNECTOR FOR PROFIBUS FOR INDUSTRIAL PC, SIMATIC OP, OLM, WITH AXIAL CABLE OUTLET DEGREE OF PROTECTION IP 20 TRANSMISSION RATE 12MBIT/S	1	46.00	46.00
6GK1500-0EA00	SIMATIC NET, LAN CONNECTOR FOR PROFIBUS FOR INDUSTRIAL PC, SIMATIC OP, OLM, WITH AXIAL CABLE OUTLET DEGREE OF PROTECTION IP 20 TRANSMISSION RATE 12MBIT/S	1	46.00	46.00
Station value:				**2,950.00**

Total value:				**4,213.00**

Appendix C
Training Course on
Siemens S7-200 PLC

PROGRAMMABLE LOGIC CONTROLLER

SIEMENS S7-200

TRAINING COURSE

Contents

Preface

PLCs (Programmable Logic Controllers) were first conceived in 1968. They are now widely used and extend from small, self-contained units for use with a limited number of digital inputs and outputs, to modular systems which can be used with large numbers of digital and analogue inputs/outputs and also for PID control.

With the increased rate of application of PLCs, millions of units are in service throughout the world today, testifying to functionality, reliability, flexibility and cost effectiveness.
The main difficulty raised by the advance of the PLC is one of education and training especially, amongst new users.

This course will provide a guide to students and technicians who are involved in studying and using PLCs.

After completing the training and carrying out all the practical exercises the trainee will be able to programme the SIEMENS S7-200 using STEP7 Ladder logic language, and use it to control simple tasks.

The prerequisites needed

In order to start this course we should have the following:

- A PC with 386 processor as minimum.
- Windows 3.1 or higher and some knowledge of how to work with it. (In this training package we shall be using the operating system known as **Windows NT**)

INTRODUCTION

This manual contains 18 sections as follows:

Section 1 provides a basic overview of the PLC, its history, structure and the wiring needed to connect it to external devices. The section introduces most of the basic terminology needed to start working with the PLC.

Section 2 introduces the SIEMENS S7-200 PLC and explains its memory structure.

Section 3 explains the installation of the programming language STEP7 software onto the PC.

Section 4 deals with the connection between PC and PLC.

Sections 5 deals with the principles of programming technique using STEP7, in terms of how to write, save and print a simple program.

Section 6 discusses the modes of PLC operation.

Section 7 explains how to downloading a program from PC to PLC.

Section 8 shows how to monitor the status of a program.

Sections 9 - 18 deal with more complicated programming instructions such as internal markers, timers, counter and arithmetic operations.

This manual also contains a number of examples and exercises.

As a result of working through this manual the trainee will have gained a good of understanding and be competence in the design, writing and downloading of PLC programs to a SIEMENS S7-200 PLC, using the ladder logic programming language provided by STEP 7 software.

1 PLC

This section introduces the basic concepts of the PLC, providing the principles of PLC constructions and defines the terms related to a PLC.

At the end of this section the trainee should know about PLC definition, history, programming languages, structures and be able to connect the PLC to external inputs and outputs.

1.1 What is a PLC?

A Programmable Logic Controller (**PLC**), figure 1, is an electronic device designed to perform control tasks by using microprocessor technology. PLCs receive input signals and generate output signals according to a set of instructions (called a program) telling it how to behave.

Figure 1 different types of PLCs

1.2 PLC History

Compared with many other electronic devices the PLC has a short history, which began in 1968 when a group of American automobile company engineers used a PLC to replace an electromechanical control system. Since that time PLCs have grown from simple devices to devices that are able to perform almost any type of control task.
Nowadays there are many PLC types installed around the world. They are of the greatest importance to manufacturing and the process industries.

1.3 Bit, Byte, Word

PLC as well as other digital systems operate by opening and closing tiny electrical switches programmed on a chip. All the switches are in one of two states: open or closed. Symbolically, we represent these switches by bits, which are generally grouped into entities known as Bytes and Words.

1.3.1 Bit

Bit (**B**inary Dig**it**) the smallest unit of information represents the digits 1 & 0 in the binary system. It can also represent the digital signal

ON = 1
OFF = 0

1.3.2 Byte, Word, Double word

A **Byte** consists of eight Bits (Binary digits). In PLCs we use bytes to define PLC memory locations. Bytes are used in a wide range of arithmetic operations.
In the S7-200 PLC we can also use words and double words to increase data size.

1 Word = 2 Bytes = 16 Bits
And a Double word = 2 Words = 4 Bytes = 32 Bits

Figure 2, shows a comparison of a Byte, Word and Doubleword access to the same address

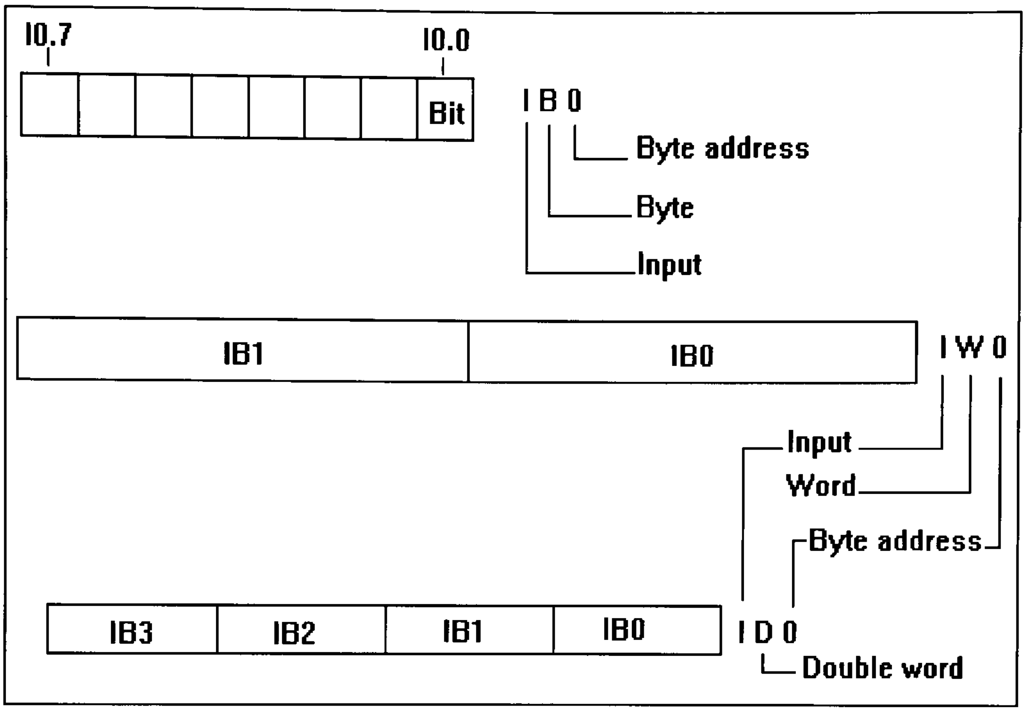


Figure 2 comparison of a Byte, Word and Doubleword

1.4 The program

This is a set of instructions that determine the PLC behaviour. It is created with a programming terminal that can also be used for other important functions such as documenting, modifying and monitoring the program. Most PLC manufacturers offer software packages that allow a standard personal computer to be used as a programming terminal. The STEP7 microWIN software package for the S7-200 PLC is an example of this.

1.5 Programming languages

There are several methods (languages) of writing a PLC program. The most common are statement list and ladder logic. Statement list takes the form of logical statements. Ladder logic is a graphical representation of the program. For an S7-200 PLC with STEP7 microWIN software both languages can be used.
Ladder logic (Ladder Diagram) is similar to circuit diagrams used by electricians to draw electric circuits. It has been the standard method of representing an electrical control circuit for some time and continues to be widely used in industry. When the first PLCs were introduced, programming followed a similar technique of ladder diagram so that technicians could learn how to program PLCs with the minimum of training. Thus, Ladder Diagram is the language that we will use in this manual.

1.6 Ladder diagram

Ladder diagram (also known as ladder logic) is a common programming format for most PLCs. It consists of rungs (called networks in STEP7) containing a number of contacts, coils and boxes. The boxes usually represent more complicated operations.
By convention the left-hand side of the diagram represents the live supply while the right hand side represents neutral.
For example the electrical circuit shown in figure 3 (a) can be represented in ladder format as illustrated in figure 3 (b).

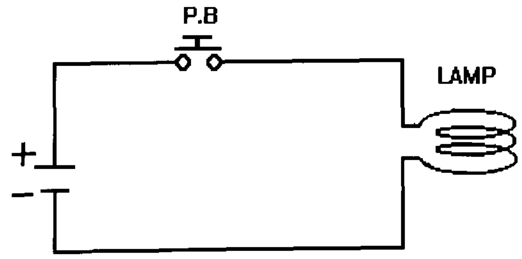


Figure 3 (a) traditional wiring

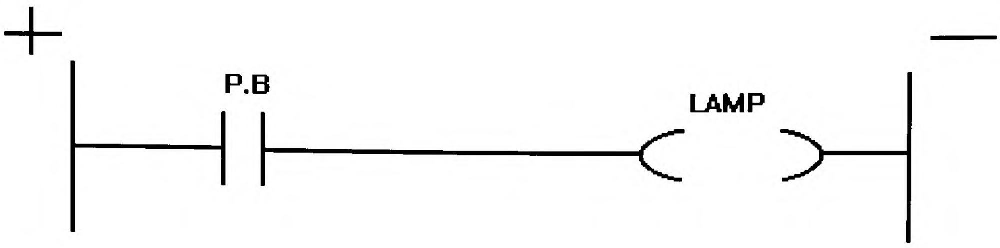


Figure 3 (b) ladder diagram

In figure 3(b) the push button is represented by the symbol (┤├) known as a **Normally open contact** and the lamp by ( ) which is the **coil** symbol.

1.7 PLC Structure

Figure 4 illustrates the structure of a typical PLC. The main part of the PLC is the **central processor unit** (**CPU**) which is the **"**decision maker**"** in the device. It makes the decisions based on the program instructions. The CPU is composed of two main sections the, **processor** and the **memory**.

The principle function of the processor is to command and govern all the activities of the entire system.

The memory of the PLC can be one of two types, volatile and non-volatile. Volatile memory loses its contents when power is removed whereas Non-volatile memory retains its content when power is removed. The most basic unit of the memory structure is a bit. Each bit stores information in binary in the form of 1 or 0. Memory size is measured in bytes or words. A byte typically represents one instruction.

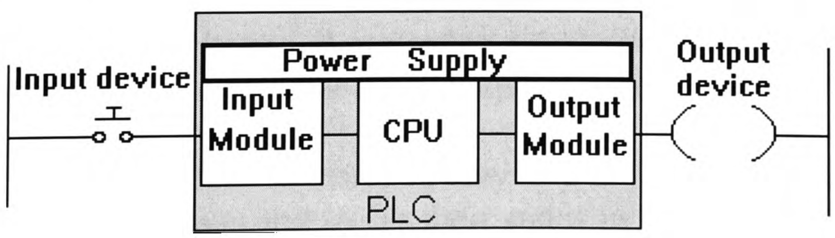


Figure 4 PLC structure

Input devices such as push buttons, limit switches and sensors which tell the PLC what is going on in the plant are connected to the **Input modules** of the PLC, where the CPU can evaluate the status of the input devices. The task of the Input modules is to interface between the input devices and the CPU. Thus the CPU can read the input signals without having to tolerate the high plant voltages.

Output devices such as lamps, relays and solenoids are connected to the **output modules**, which also are part of the PLC, figure 5.

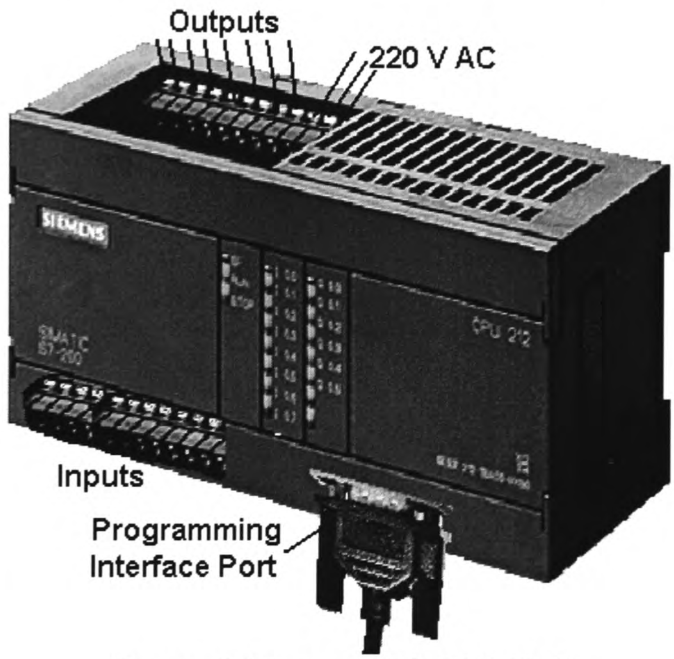


Figure 5 Siemens S7-200 PLC

These output devices receive their signals from the CPU and provide isolation between the CPU and the output devices.

The task of the **power supply** is to serve the various parts of PLC with the correct voltages. It may also supply the voltage for the input devices and sometimes to the output devices.

When we ask a PLC to perform any control task it starts by scanning all the inputs and stores their states in its memory. Then it goes on to process the user program and finally it updates the outputs (see figure 6). The time needed to complete one cycle is called the scan time. The PLC must repeat this cycle indefinitely and very quickly (in typical systems the scan time is in the range of 5-100 ms).

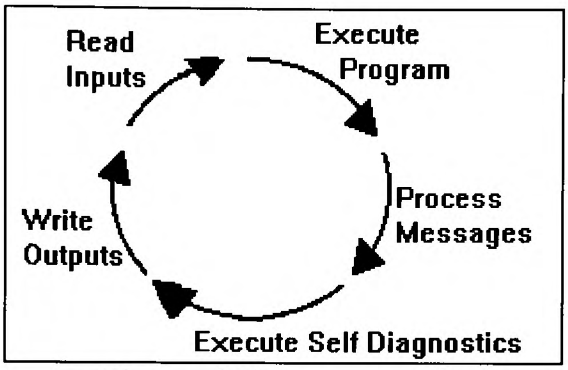


Figure 6 One scan cycle

The S7-200 PLC incorporates a protective mechanism (known as watchdog timer) which operates automatically. If the scan time is excessive, the program execution is stopped and a warning lamp on the front of the PLC is illuminated.

1.8 PLC wiring

Every PLC has at least three sets of terminals through which it is connected to the plant and to the power supply. These are the Power terminals, Input terminals, and Output terminals (figure 7).

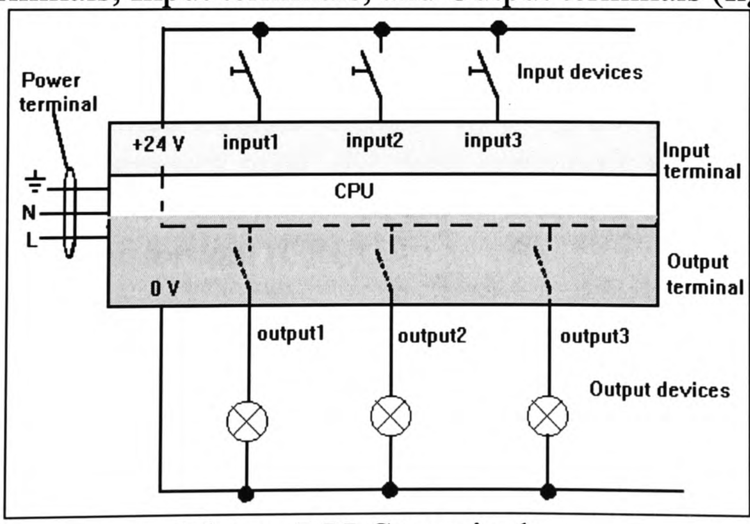


Figure 7 PLC terminals

The **Power terminals** connect to the mains supply (240V ~).

The **input terminals** are used for the connection of the switches and sensors. Each input device is individually connected to one input terminal on the PLC.

Similarly, the **output terminals** are used to connect the output devices to the PLC. Each Output device is individually connected to one Output terminal on the PLC

PLCs come in many forms and capabilities ranging from small PLCs with a limited number of inputs /outputs, to extremely powerful PLCs able to handle hundreds or thousands of inputs /outputs.
The small PLCs come as integrated packages complete with power supply, CPU and a fixed number of input and output terminals in a compact package. Some types of small PLCs are expandable with input and output modules that can be connected to the base unit. The user decides the mix of modules best suited to his requirements and assembles them accordingly. One of the smaller types PLCs, the Siemens S7-200, is the subject of this training programme.

Summary
Th PLC is an electronic device that is used to carry out control tasks by using a set of instructions (called a program) stored in its memory. The memory size, measured by Byte or Word, consists of a defined number of Bits. To program the PLC many different languages are used, the most commonly used language is ladder logic. The main parts of PLC structure are:

- *Central processor unit (CPU).*
- *Input modules.*
- *Output modules.*
- *Power supply.*

A PLC is connected to plant through sets of three terminals, which are the power terminals, input terminals, and output terminals.

2 S7-200 PLC

This section introduce the SIEMENS S7-200 PLC. It provides a specification of this type of PLC and explains its memory structure.

S7-200 is a family of PLCs manufactured by a German company called SIEMENS. The family contains different types of basic unit, where each unit has a particular specification. Table (1) lists some of these specifications.

Basic Unit	Number of Digital I / O	Number of Expansion Modules	Number of Timers	Number of Counters	Program Memory KB	Data Memory
CPU 210	4 I /4 O	----	4	2	0.25	----
CPU 212	8 I /6 O	2	64	64	1	512 word
CPU 214	14 I /10 O	7	128	128	4	2 K word
CPU 215	14 I /10 O	7	256	256	8	2.5 K word
CPU 216	24 I /16 O	7	256	256	8	2.5 K word

Table 1 Basic Units specification

2.1 S7-200 memory

The memory of an S7-200 PLC is divided into three areas:

- **Program space.**
- **Data space.**
- **Parameter space.**

2.1.1 Program space

Program space stores the user program instructions to be executed by the PLC.

2.1.2 Data space

The **data space** is used as a working area. It includes memory locations for constants that are used as control parameters and also includes special devices such as timers and counters.
It consists of data memory and data objects as shown in figure 8.

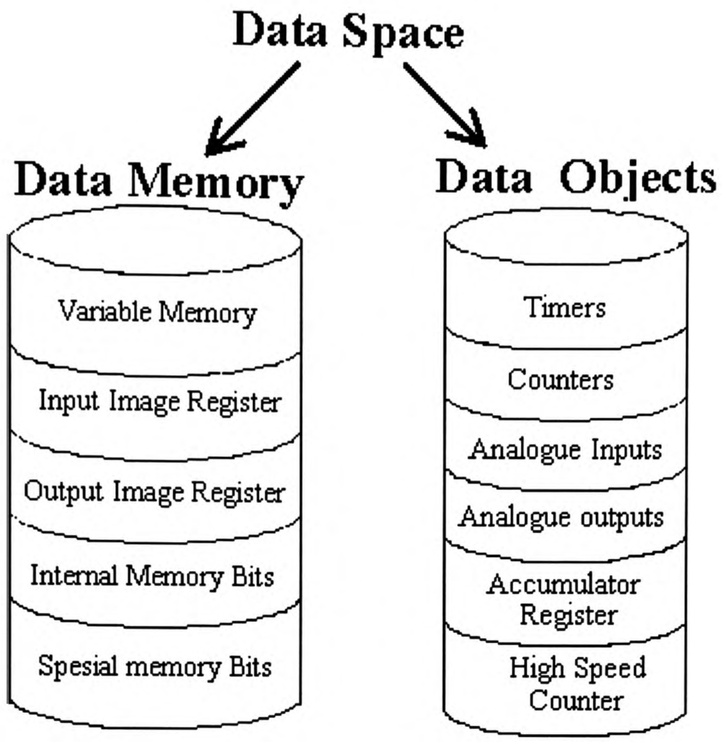


Figure 8 data space

2.1.3 Parameter space

The **parameter space** provides storage for a selection of **configurable parameters**, such as the password and station address, which define the way that the PLC and programming unit are connected and function together.

Summary

The S7-200 PLC is one of the S7 PLC family manufactured by a German company called SIEMENS. Its memory divided into three areas:

Program space, to store the user program.

Data space, which is used as a working area.

Parameter space, where the configuration parameters are stored.

3 Installing and setting up STEP7-micro/win

This section explains the installation of STEP 7 Software onto a PC.

The STEP7 software package is contained on a set of floppy disks or one compact disk (CD). STEP7 must be installed to allow the PC to be used as a programming terminal.

To install STEP 7 Software, follow these steps:

Insert the disk (disk 1 of the floppy disks) into the appropriate drive (floppy disk in drive A, in this case).

From the **Start** menu select **Run** as shown in figure 9.

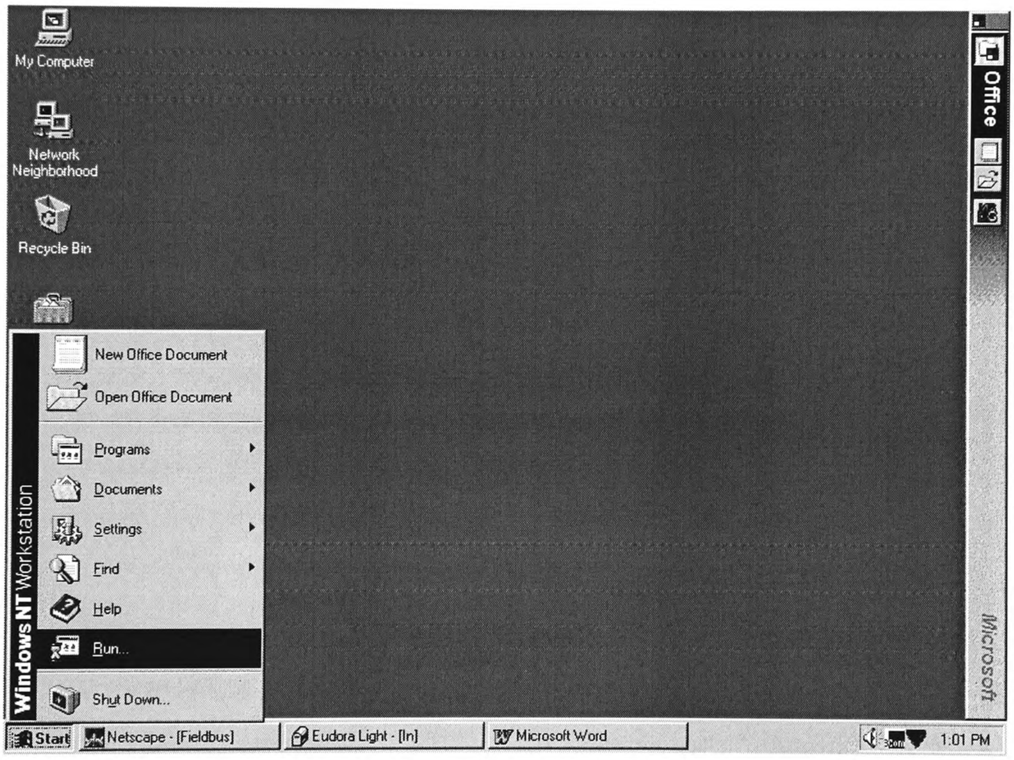


Figure 9 Start menu

Click on

In the **Run** dialogue box (Figure 10).

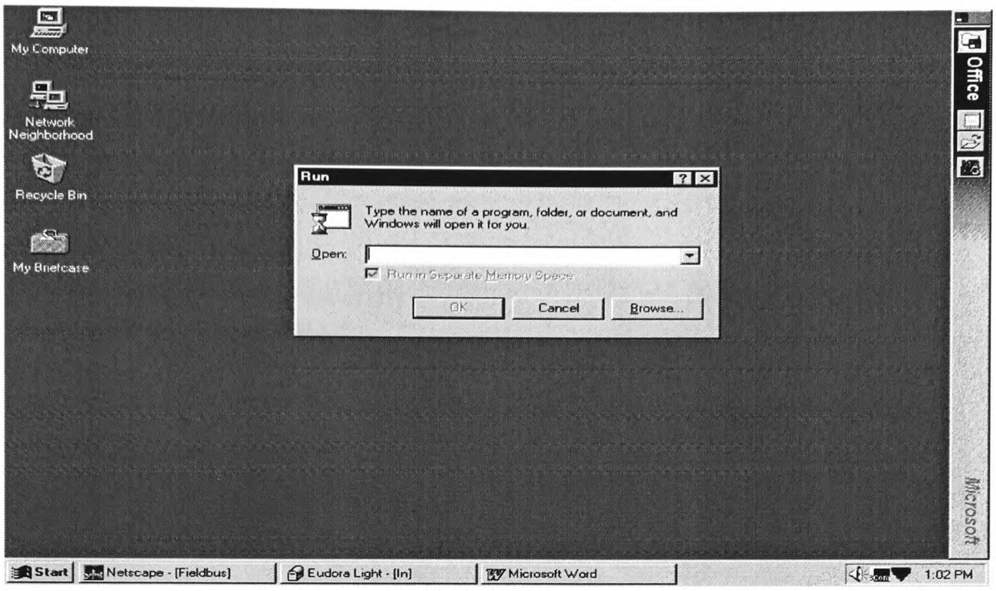


Figure 10 Run dialogue box

Select the appropriate drive (31/2 Floppy (A:)), Figure 11.

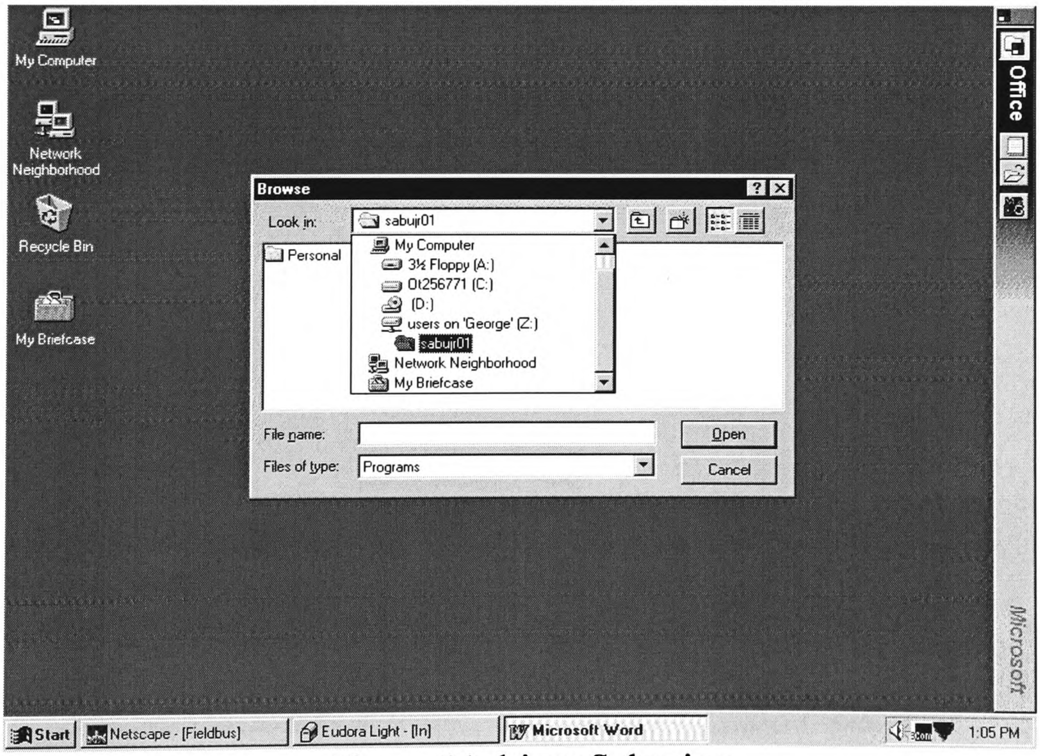


Figure 11 driver Selection

Select the **Setup** file. And click on **Open** (figure 12)

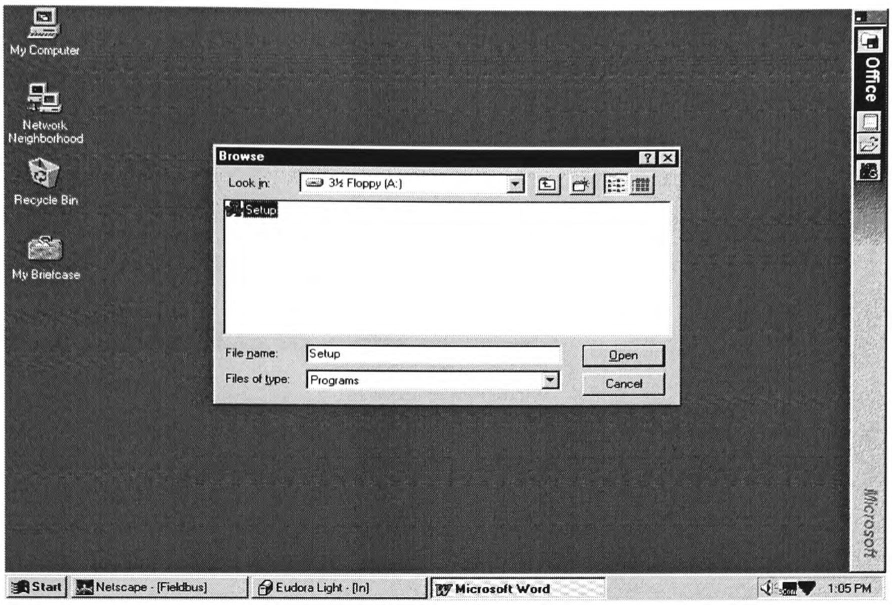


Figure 12 starting Set-up

Follow the instructions on the screen (figure 13) until the installation is complete.

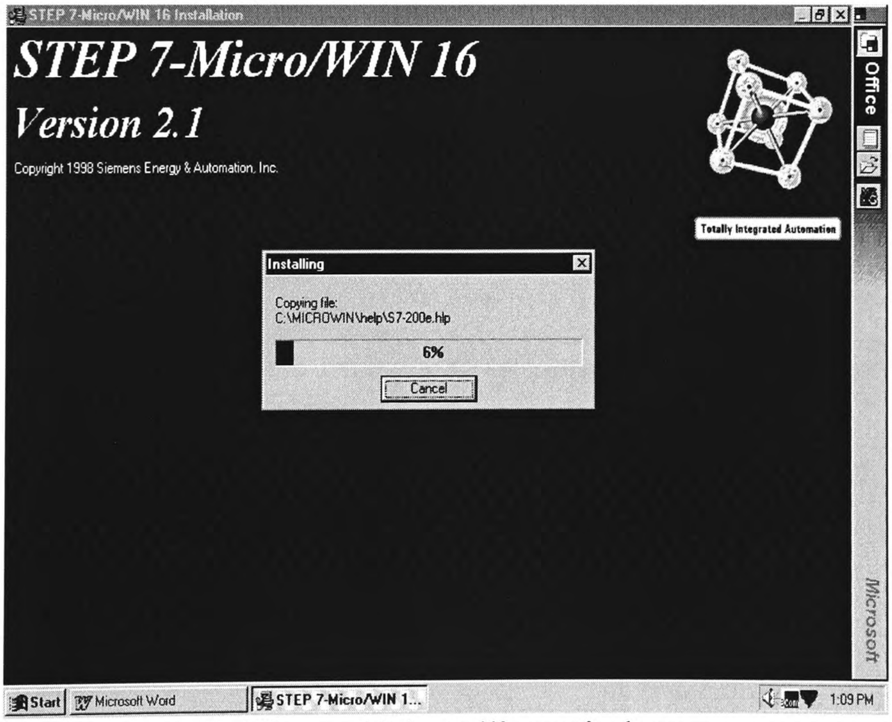


Figure 13 installing window

Installing STEP7 Software allows the PC to act as both a programming unit to the PLC, and as a monitor for an existing program. Monitoring can only be done when the PC and the PLC are connected.

Summary
To install STEP 7 Software insert the disk into the appropriate drive and follow the instruction on the PC screen.

4 Connecting an S7-200 PLC to a PC

This section deals with how to connect the PC to the PLC. At the end of this section the trainee should be able to connect the PC to the PLC and use it as programming unit.

The program is created with a programming terminal (in this case a **PC**) before being transferred to the PLC by means of an RS232 to RS485 multi-drop cable connected between a serial interface on the PC and the S7-200 communication interface, figure 14.

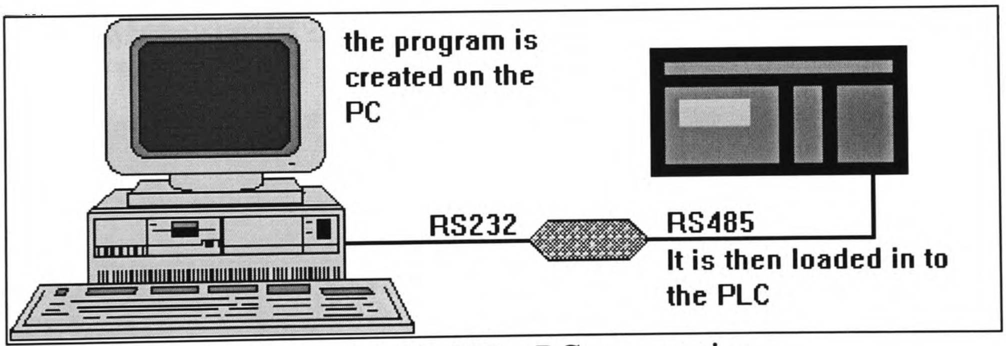


Figure 14 S7-200 - PC connection

5 Using STEP 7

This section deals with the basic programming instructions. It is the first step in using STEP7 as a programming tool.
In these sections the trainee will learn how to start a new project, how to edit a program, address the inputs and the outputs, document the program, compile the program, and save and print a project.

5.1 The STEP7 micro/WIN software package

STEP7 supports S7-200 PLCs, which means that it provides a set of features that enable us to:
Edit, find and correct errors in the program.
Communicate with the PLC and observe the status of the program.
Print the program.

5.1.1 Getting started

To start STEP7 carry out the following:
On the **Main** screen of the Windows software, double click on the **STEP7 Micro/WIN** icon (figure 15)

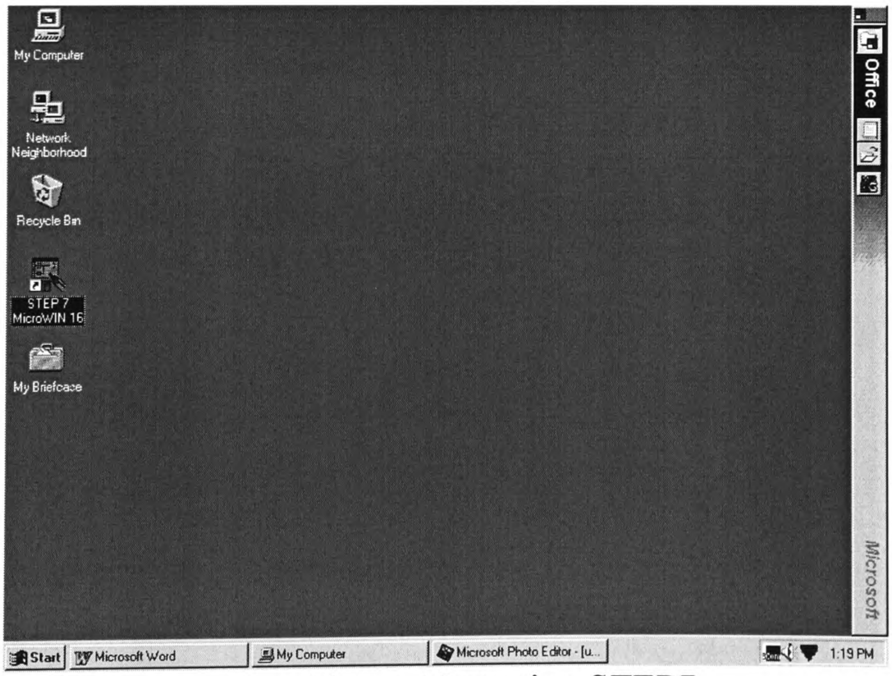


Figure 15 starting STEP7

The first window of STEP7 should appear on the PC screen as shown in figure 16.

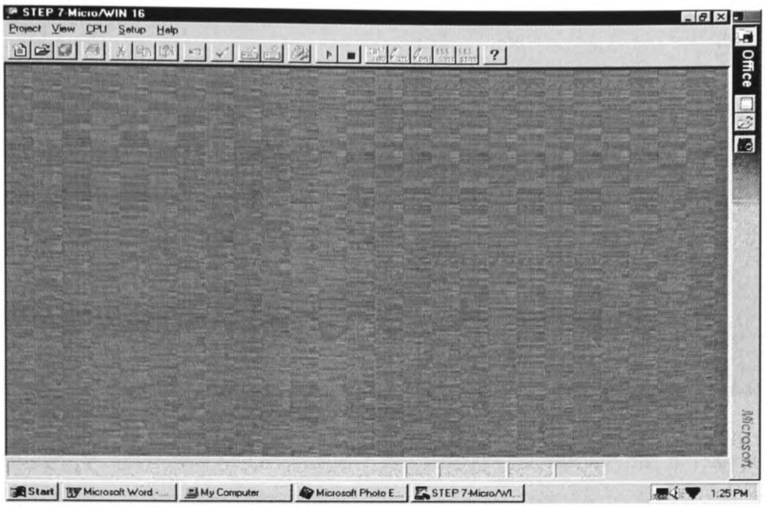


Figure 16 first window of STEP7

5.2 Starting a new project

In industrial applications as well as others, PLCs can be used to control many processes, each process requiring a different program. In order to choose between them we give the various programs different names.

To **start a new project** try the following:

1- From the initial screen, figure 17, select **New** from the **Project** menu or click on  in the toolbar.

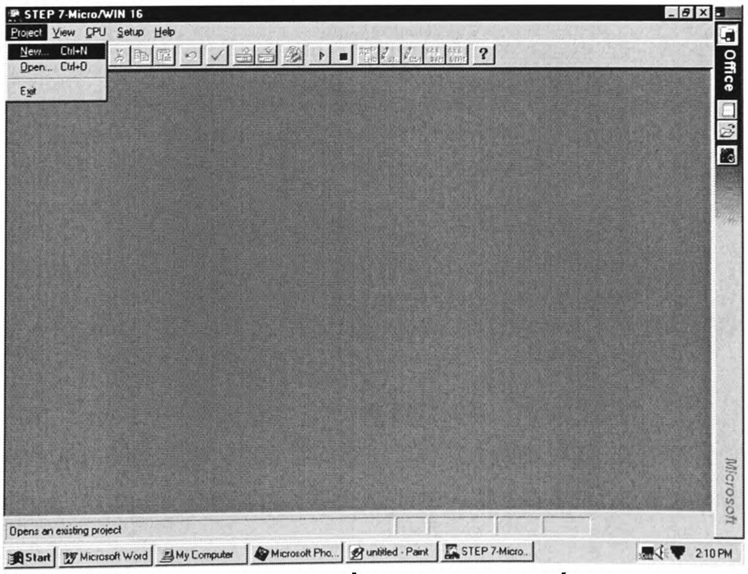


Figure 17 starting a new project

2- Choose the CPU type from the list, figure 18 (select CPU 212).

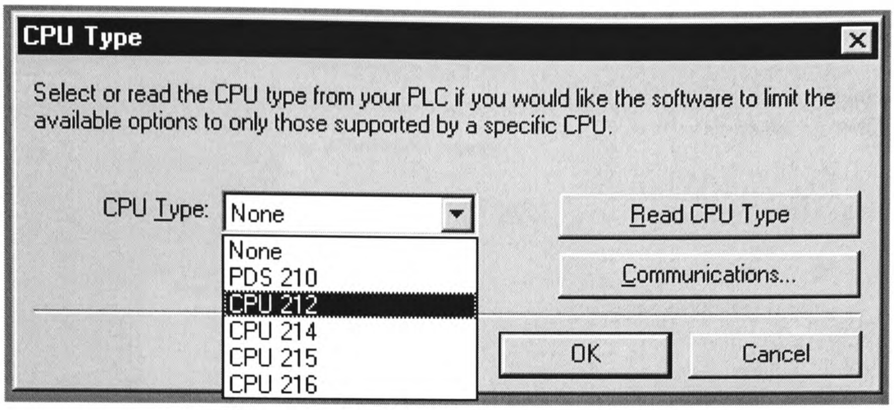


Figure 18 Choosing the CPU type

3- Click on the 

button.

5.2.1 Project editing

In practice we would not use a PLC to control the simple circuit which is shown in figure 19. However, as an introduction to some STEP7 instructions we are going to use our PLC to perform this simple task. We want to switch the lamp ON and OFF using the push button switch.

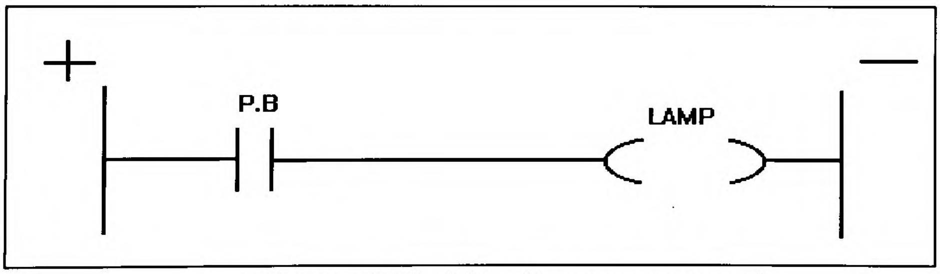


Figure 19 ladder diagram network

Select the editing language using one of these two methods:

1- From the **View** menu choose **Ladder**, figure 20.

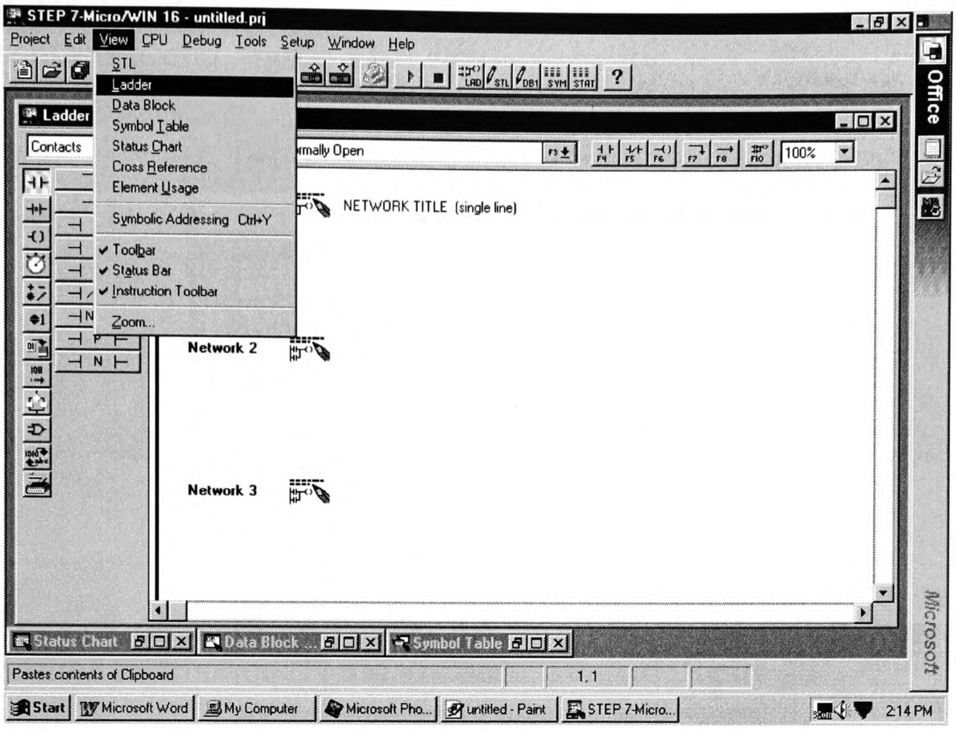
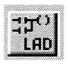


Figure 20 View menu

Or

2- click  on the toolbar.

Now to start editing the program do the following:

1- To enter the normally open contact, position the cursor at the first network, figure 21.

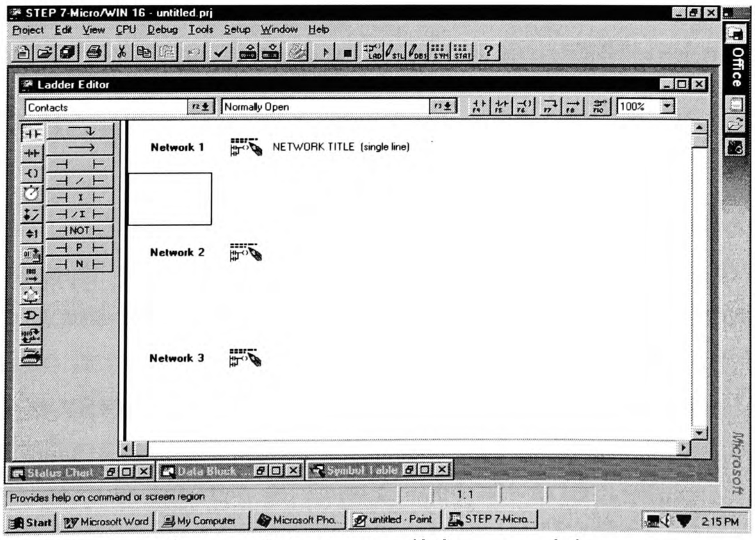
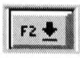


Figure 21 start editing position

2- Click  to access the list of instruction groups. We need the contacts group, figure 22.

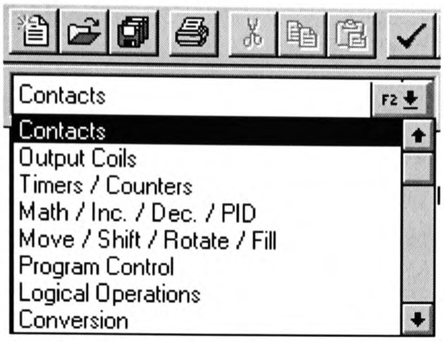
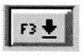


Figure 22 list of instruction groups

3- click on  to access the drop-down list and choose the desired instruction (Normally open contact), figure 23.

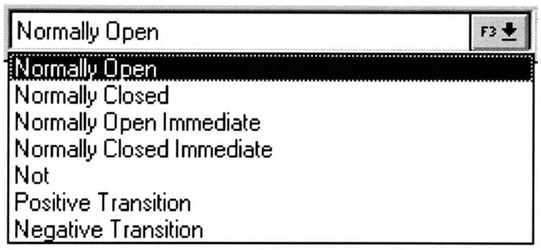
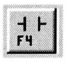


Figure 23 instruction list

We could also click on , or use the **F4** key on the keyboard. Or we can use the icons located on the left hand side of the screen, which all can enable us to edit a normally open contact.

5.2.2 Input/output Addressing

Each input and output module has a unique number of connecting points, which are designated by the PLC manufacturer. This means that we have to specify a particular input (screw terminal location) to access an input signal and a particular output (screw terminal location) to access an output signal. The specification of input locations on the input module and outputs locations on the output module is known as **input/ output addressing**.

For the S7-200, inputs are identified by the letter **I**, figure 24, and outputs by the letter **Q**.

In our example we should connect the push button switch (input) to one of the PLC's input terminals and the lamp (output) to an output terminal.
The push button, which is an Input Bit, will be connected to the PLC input terminal as part of a specific Byte, which contains this Bit together with seven other Bits. We must tell the processor exactly which Byte and which Bit and whether it is an input or Output. Figure 25 illustrates this addressing technique of the Input and in the same way the lamp should be addressed as output (**Q 0. 0**).

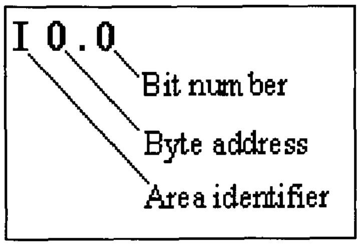


Figure 24 inputs identification

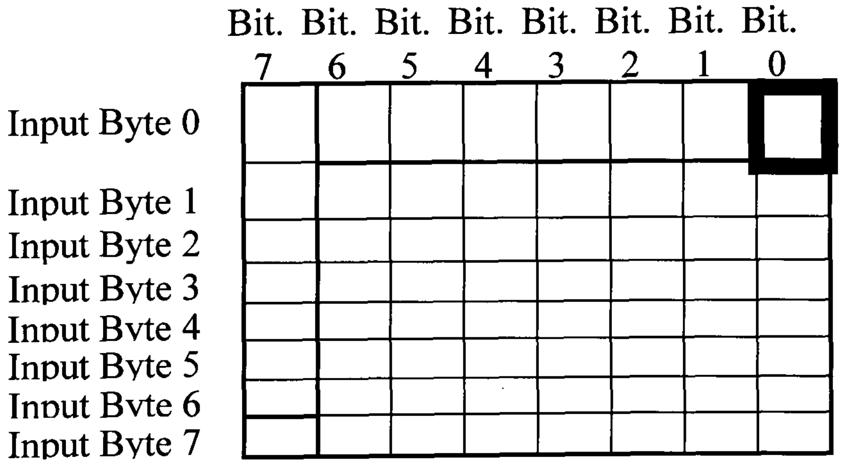


Figure 25 Bit, Byte addressing

Type **I0.0**, as the address for the input, as shown in Figure 26. Where

I Input
0 Byte number 0
0 the Bit number is 0

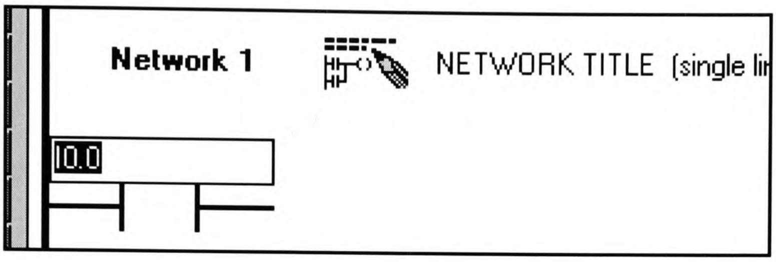


Figure 26 input addressing

Press Enter (↵) to confirm.

To enter the coil which represents the lamp:

Press **F6** or click on [F6 coil button] as shown in Figure 27.

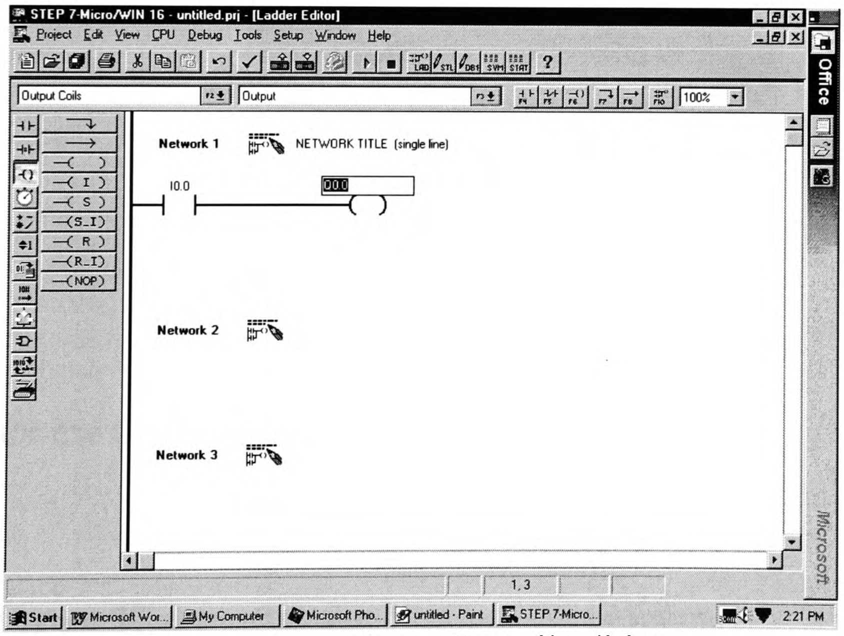
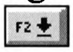


Figure 27 coil editing

Address the output in the same way as the input.

5.2.3 END

The network (known as a rung in some other PLC types) that we have on the screen represents our simple program. Now we have to tell the PLC processor that our program is finished.

Start a new network and click on  to access the drop-down list of instruction groups, figure 28. Click on **Program Control**.

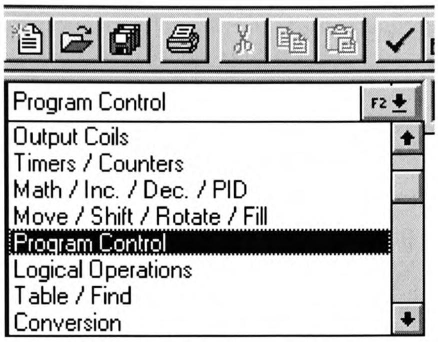
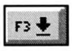


Figure 28 instruction groups list

Click on  to access the drop-down list and choose the desired instruction, in this case (**End**), Figure 29.

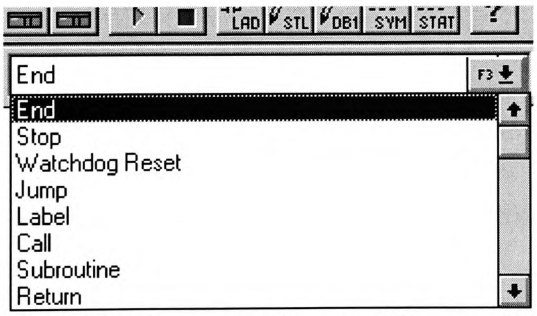


Figure 29 Program Control instruction

Or use the **Program control** icon:

Figure 30 shows the final form of our program.

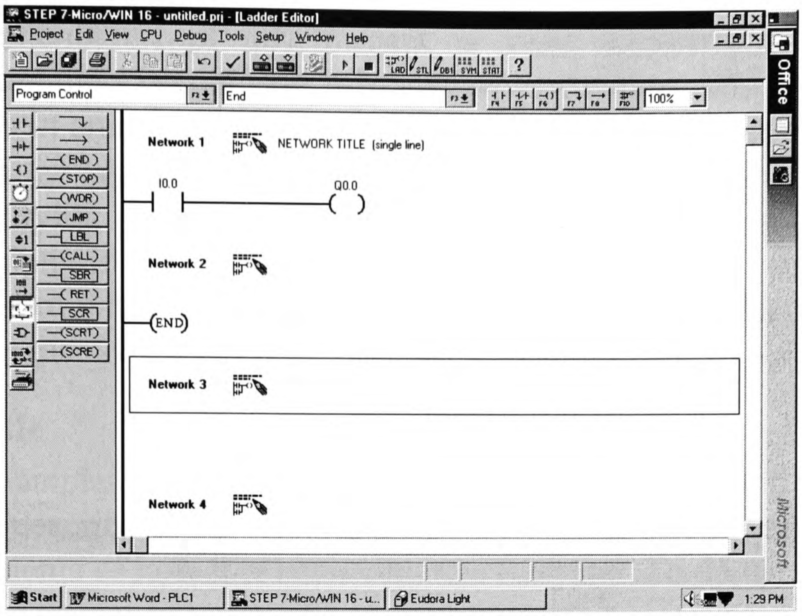


Figure 30 program's final form

5.2.4 Delete

Deleting a rung in STEP7 removes the rung from the current diagram.

To delete a rung select **Delete** from the **Edit** menu to open the Delete dialog box, where you can choose from the following options:

Row: Deletes the row where your cursor is positioned.
Column: Deletes the column where your cursor is positioned.
Network: Deletes the network where your cursor is positioned.
Vertical Line: Deletes the line you have selected.

Click on the OK button to perform the operation selected and close the dialog box.

Click on the Cancel button to cancel the operation and close the dialog box.

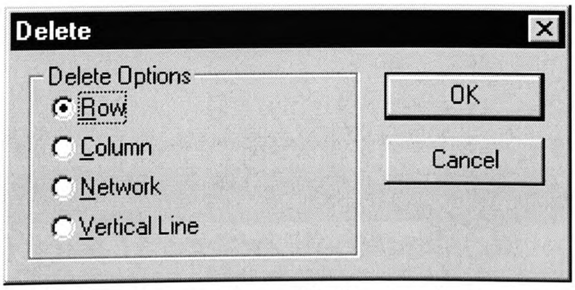


Figure 31 Delete dialogue box

5.3 Program Documentation

Documentation of any project is very important and many programmers do not give this sufficient attention. They think that the input and output addresses are sufficient. This is probably true while the program is still fresh in their minds, but for someone else trying to maintain the system, documentation is vital, especially if the project is large. Documentation is not only just a record of what we have done but also can be a maintenance tool where the complete project can be reviewed.

5.3.1 Network Title

For complete documentation every network requires a name or title describing its function in the project, and also some comments of how it is achieving this function. To do this:

Double click on

The screen shown in figure 32 appears:

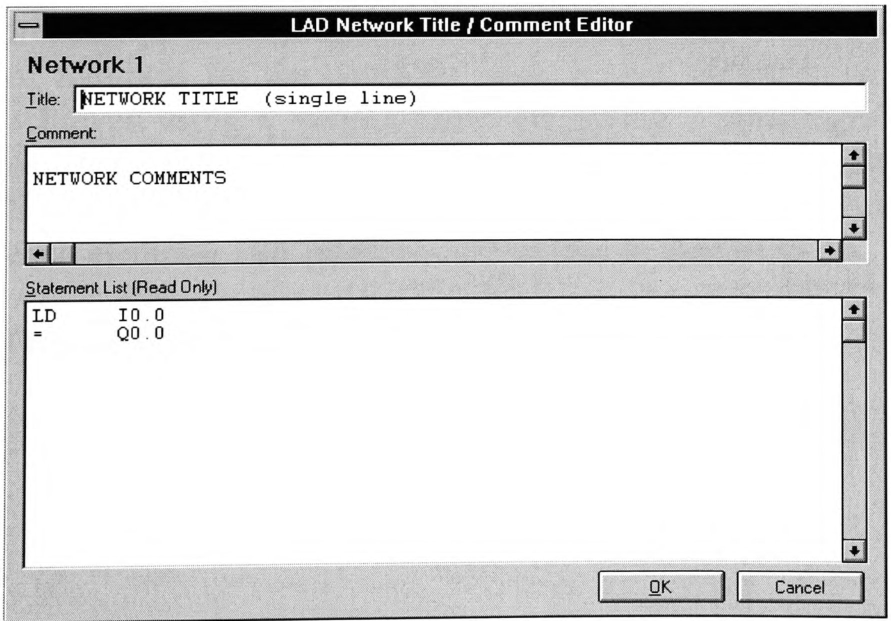


Figure 32 network documentation

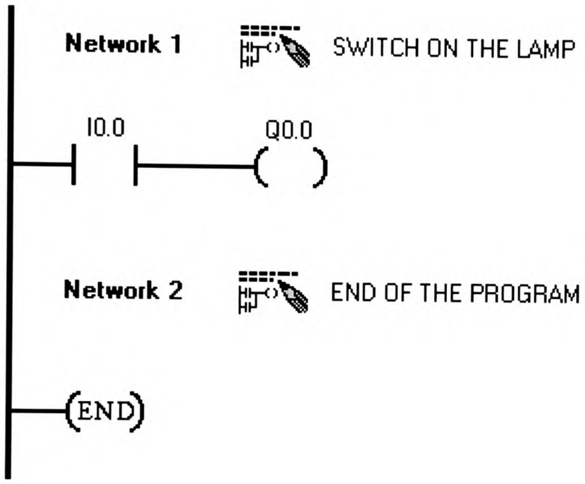
Type SWITCH ON THE LAMP as a title for the first network.

Click 

to confirm.

Type END OF THE PROGRAM as a title for NETWORK 2.

The program now takes the following form:



5.3.2 Symbol table

In addition to defining the function of each network, the provision of Symbols for the contacts, coils, timers, counters, etc. can be achieved using a symbol table. By giving a meaningful symbol name to every element appearing in the programming code, linked by suitable comments, we can describe the function of each element. This table is a useful tool in case of faults.

From the **View** list click on **Symbol Table**, Figure 33.

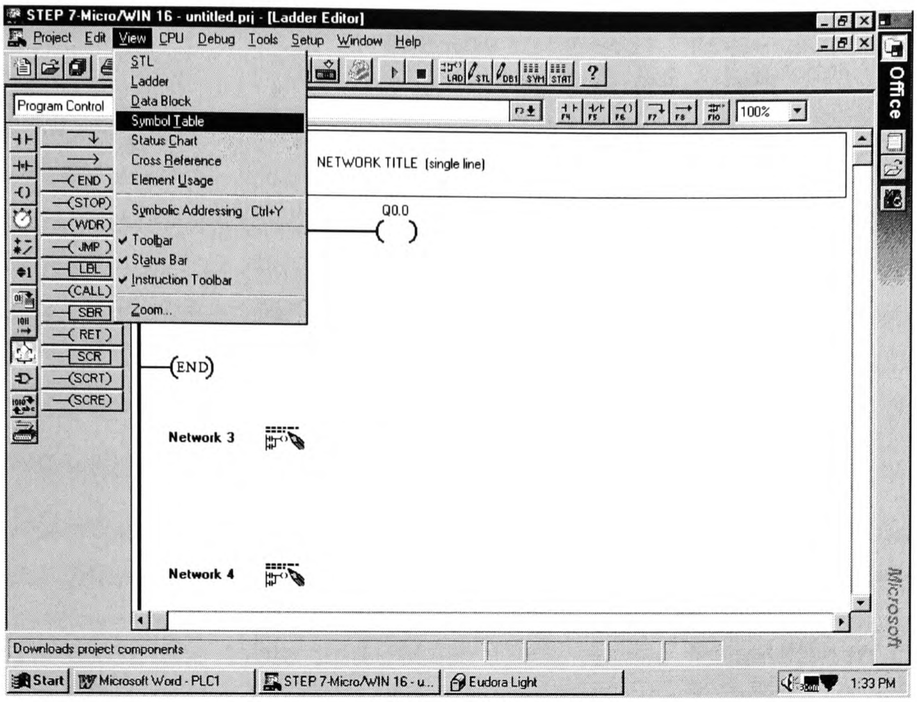


Figure 33 view list

Type into the table, Figure 34, a suitable symbol name for each element:

Switch 1 for **I0.0**

Lamp for **Q0.0**

Symbol Name	Address	Comment
switch 1	I0.0	

Figure 34 Symbol Table

To save the table click on **Save All** on the **Project** menu, Figure 35.

Figure 35 program saving

To close the table and return back to the main screen click on **Ladder** on the **View** menu, Figure 36.

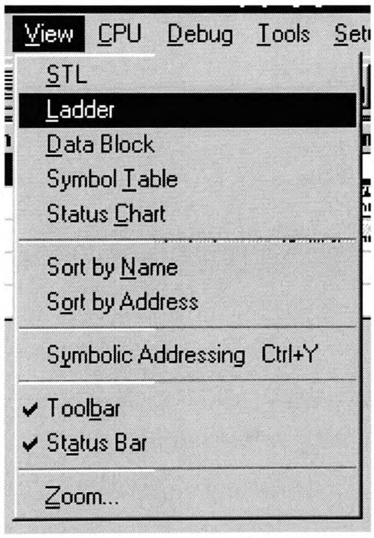


Figure 36 view list

5.4 Compiling a program

Compiling is the translation of the program code into machine language for execution by the CPU. In other words it checks the project and if any of the following errors exist the compile will fail:

1. A missing contact or output element in the network.
2. A short or open circuit.
3. In the case of reverse power flow.
4. Illegal placement of positive or negative transition or NOT instruction.
5. Invalid network.
6. Network too large.
7. Additional outputs or boxes with a counter box.
8. Interconnection between counters.

To compile your project select **Compile** from the **CPU** menu, Figure 37.

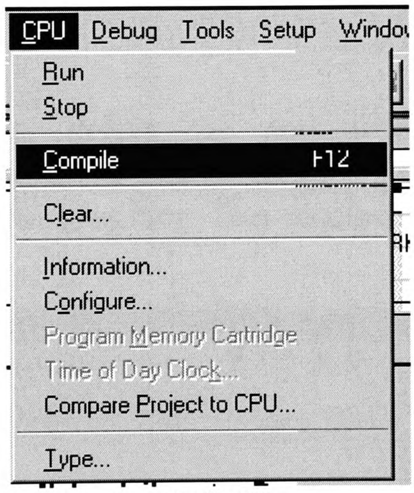


Figure 37 **CPU** men

Or click on  on the toolbar

When the compilation is completed you will receive the result in the form of a message as shown in Figure 38.

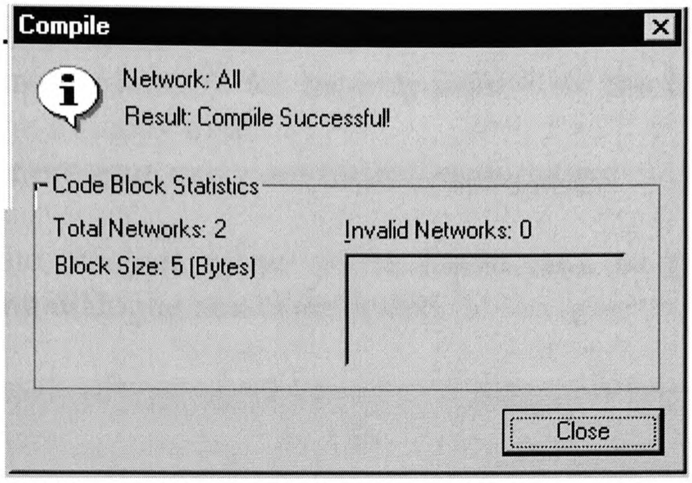


Figure 38 successful compilation result

Click [Close] to close the compile box.

If there are any errors a message is displayed indicating the network numbers.

To demonstrate this, delete the coil from network 1 by positioning the cursor on it and striking the Delete key on the keyboard. Compile the program. The result should be as shown in figure 39.

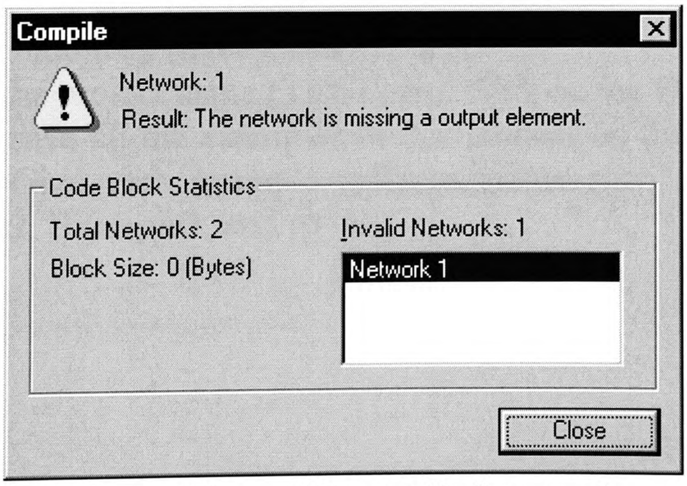


Figure 39 compilation result

5.5 Saving the Project

We may download the program directly to the PLC where it is stored in PLC memory. However, to avoid the danger of loosing the program accidentally we have to store it on the hard disk of our PC or on a floppy disk.
To save our program carry out the following steps:

1- From the **Project** menu select (**Save As**), to give us the following dialogue box (Figure 40).

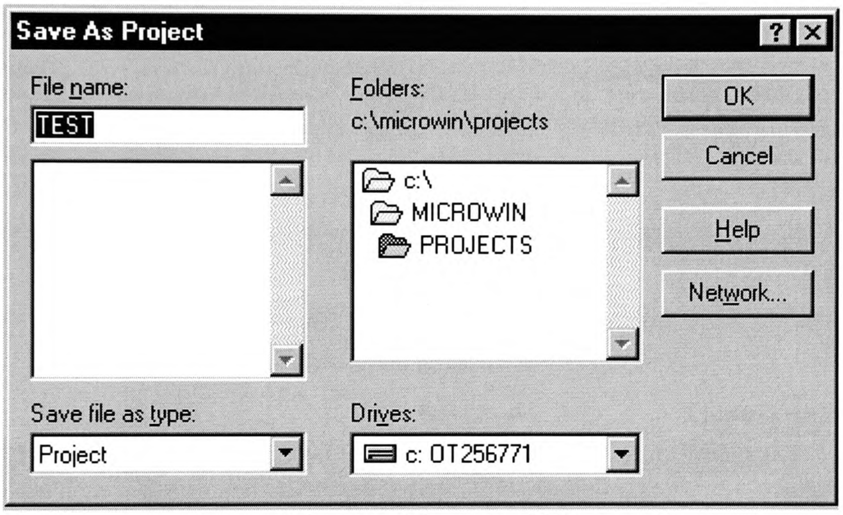


Figure 40 save dialogue box

If we want to save our project on to a floppy disk, change the path from the hard disk **drive C:** to **drive A**:
Type in the project name (**TEST**) and click on the **OK** button.
We will save all the examples in this manual on the hard disk of the PC. Give each example different name such as Example1, Example2, Example3, and so on.

5.6 Project Printing

In addition to saving our project on hard disk or floppy disk, a printed copy of the project can be obtained by using the print facilities (obviously a printer should be connected to the PC). The printed copy can be a useful tool for developing and maintaining our project. To print the project select **Print** from the **Project** menu.

The following dialogue box will appear on the screen (see Figure 41).

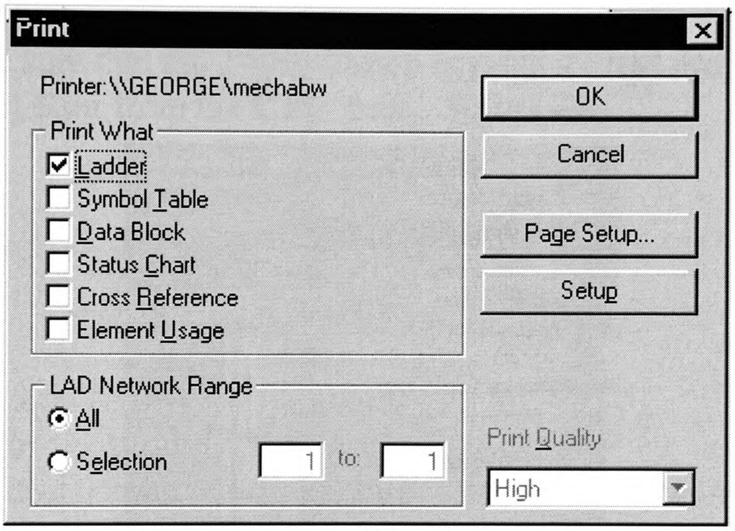


Figure 41 printing dialogue box

Select the desired box, (in this case **Ladder**) and click on **OK** button.

6 PLC MODES

This section defines the different modes of PLC operation. The trainee will be able to switch the PLC into different modes, so that he can Run or Stop program execution.

The S7-200 has two main modes of operation **STOP** and **RUN.** In addition, the 3-way switch (on the front panel of the PLC) has another position called **TERM**.

1. In **STOP** mode, the PLC does not performing any control functions. This mode enables us to download the program from the PC into the PLC memory.
2. **RUN** mode executes all the instructions contained in PLC memory.
3. In **TERM** mode we can change the mode of operation by sending a message from the programming device.

First of all, let's prove that we can change the operating mode from the PC.

Switch the PLC to RUN mode by following the steps:

1- Make sure that the 3-way switch is set to TERM mode.
2- Select **Run** from the **CPU** menu, Figure 42.

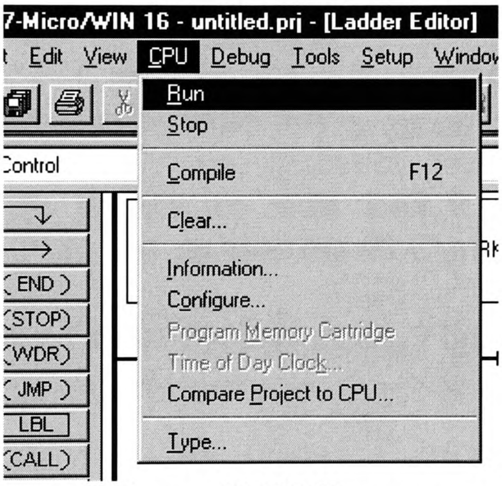


Figure 42 **CPU** menu

Or click  on the toolbar.

3- Click 

in the following box, Figure 43.

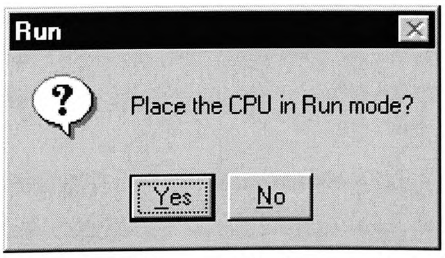
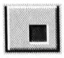


Figure 43 run dialogue box

In the same way we can switch the PLC to STOP mode by:

- Selecting **Stop** from **PLC** menu. Or
- Clicking  on the toolbar.

Summary
The S7-200 has three modes of operation:
Run: execute the program
Stop: do not perform any control function.
Term: the programming device is able to change the mode of operation.

7 Project Downloading

This section explains how to download the program into the PLC memory, and defines the different options of downloading. At the end of this section the trainee will be able to download the program into the PLC memory and understand the meaning of the different options in the download box.

After creating the project we must load it into the program memory of the PLC. To do this the PLC must be in the stop mode.

1- Select Download from **Project** menu, Figure 44, or click on  on the toolbar.

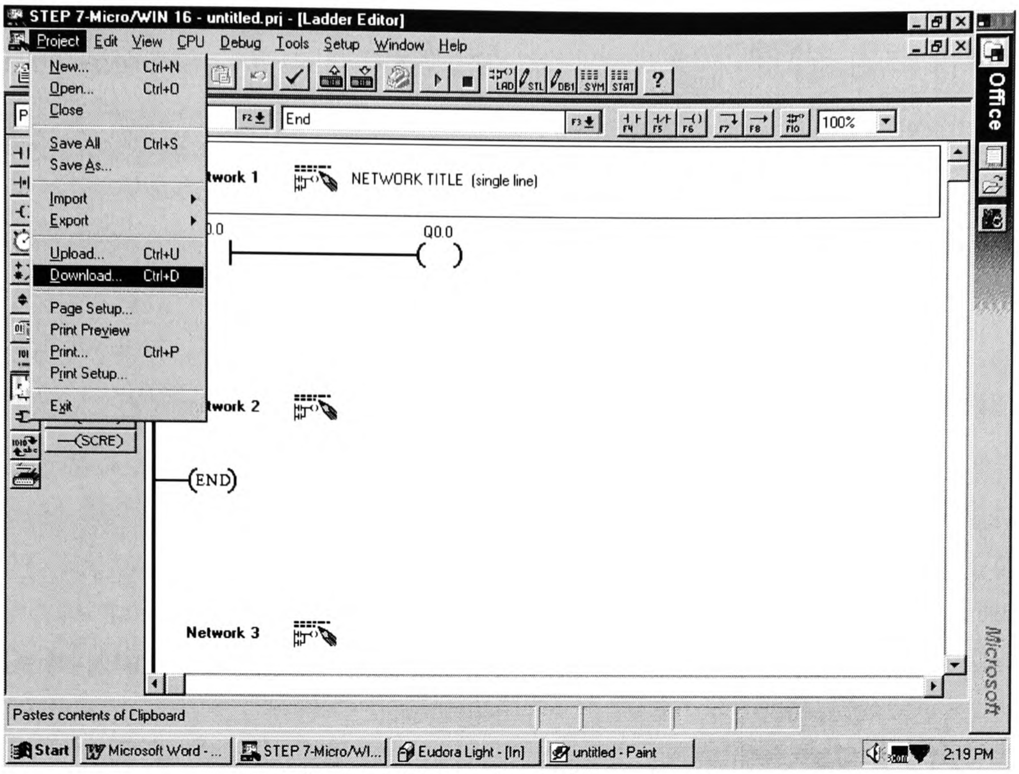


Figure 44 Project Download

STEP 7 gives us some options, Figure 45.

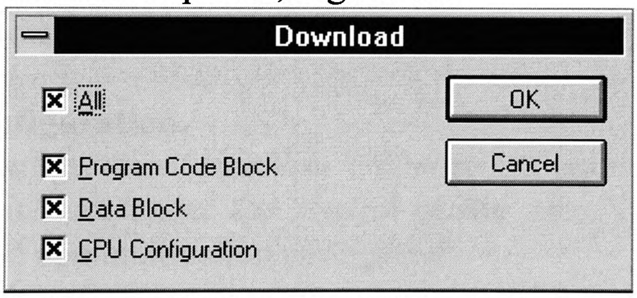


Figure 45 Download dialogue box

1- Program Code Block

These are the ladder logic program instructions.

2- Data Block

Data block is a selection of data such as calibration constants under the control of our program,. In cases where this data is constant we may download it in to the PLC memory. At the moment we have not created a Data Block.

Data Block is created by clicking on **Data Block** from **View** menu, Figure 46.

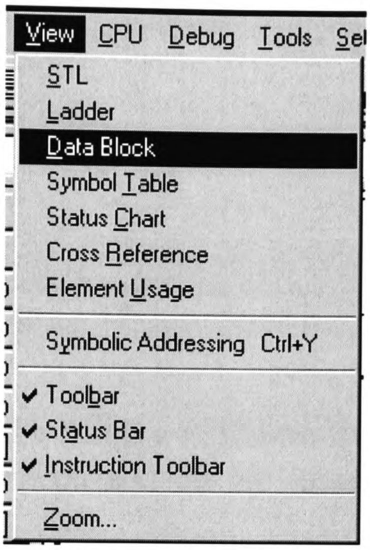


Figure 46 creating Data Block

Or by clicking  on tool bar.

To create a data block for your program, enter the data locations and values in the data block file. The data block editor is a free-form text editor; i.e., no specific fields are defined for particular types of information. However, a few rules must be followed to enable the data to compile correctly.

3- CPU Configuration.

The CPU configuration describes the basic attributes of the CPU, some of which are under the control of the user. These include password and communication parameters.

In general, the greater the number of selected boxes (figure 45) the more time is needed to transfer the project to the CPU.

Select the appropriate box (Program Code Block, in this case) and Click on  to confirm.

When the program successfully transfers to the PLC, the following message appears on the screen, Figure 47.

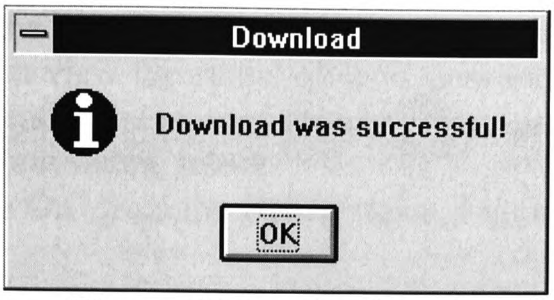


Figure 47 download result

Click on  and switch the PLC into RUN mode.
Try to switch the lamp on and off by pressing and releasing the push button.
If download is not successful a message appears on the screen, (see figure 48). The message indicates the reason and what to do to correct the fault. Figure 48 shows an example of downloading while the PLC is disconnected from the programming unit.

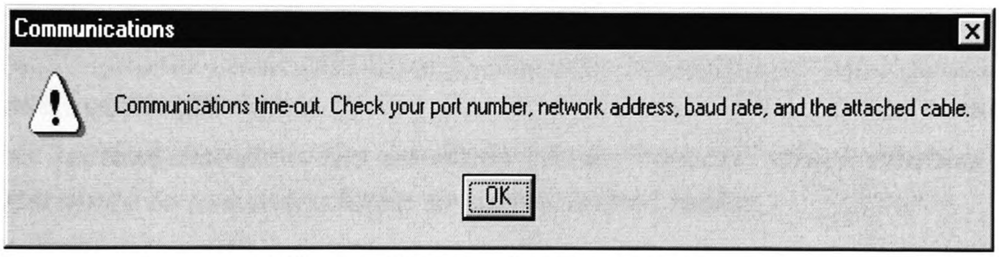


Figure 48 warning message

Summary

Downloading a program means transferring it from the programming unit into PLC memory. To do this the PLC must be in stop mode. The three options of downloading are:

- *Program Code Block: the ladder logic instructions.*
- *Data Block: constant data*
- *CPU Configuration: configurable parameters.*

8 Program Status

This section introduces a tool to monitor the status of the program while it runs. The trainee can trace many faults by using this tool.

By examining the status of contacts, coils, timers and counters for example, we can view the status of each network of our project, and possibly identify the reasons for any project errors.
To access program status, select
'Ladder Status On' from the **Debug** menu, Figure 49.

Figure 49 program monitoring

This screen will display the status of the different networks if the PLC is in **RUN** mode.

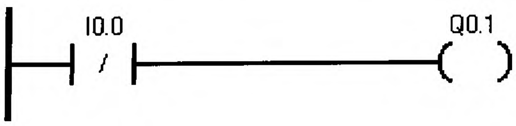
Watch the screen as you switch the lamp on and off.

9 Normally closed contacts

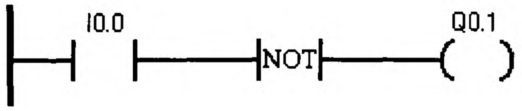
This section discusses the normally closed contact, which enables the trainee to use extra logic to solve control tasks.

In the previous example we were using a normally open contact (-| |-) to perform the task.

STEP7, as well as all other PLC programming software packages, also provides a **normally closed contact** (-|/|-). For a normally closed contact, power flows through it when its Bit value is 0.
Try the following program to understand the difference between normally open and normally closed contacts



We can also do the same thing by using -|NOT|- and -| |- as shown below:



Exercise 1

At the end of a production line, a supervisor tells the workers to reject any faulty packages by activating a light signal using a push button. Figure 50.

Write a program to perform this task using a Ladder Diagram.

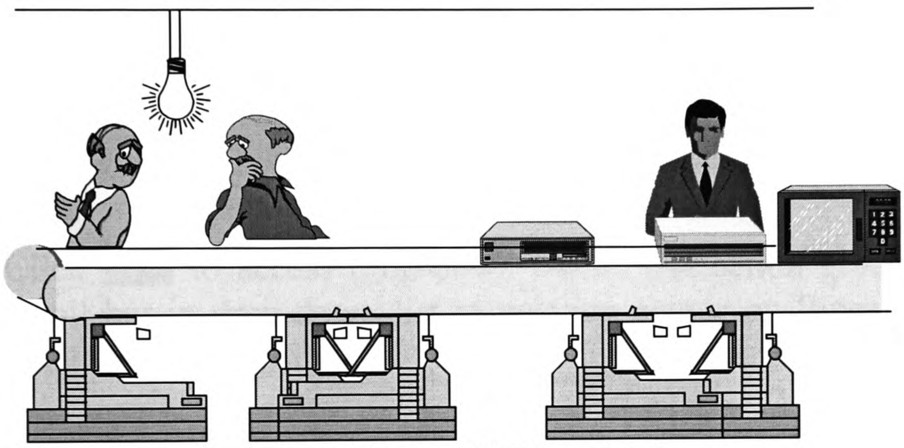


Figure 50 Exercise 1

Exercise 2

To confirm the rejection order the supervisor must use both his hands to press two push buttons.

He draws the diagram Figure (51) and the truth table of the task is as follows:

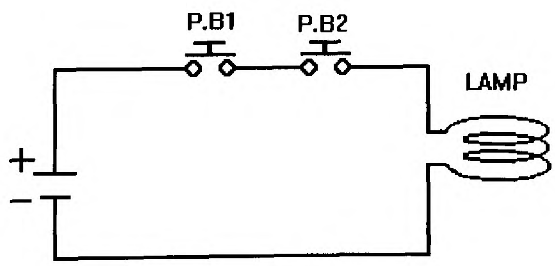


Figure 51 Exercise 2

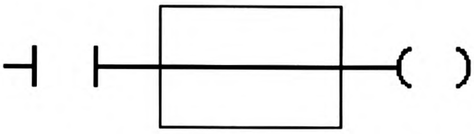
P.B1	P.B2	Lamp
0	0	0
0	1	0
1	0	0
1	1	1

Table 2 AND result

Write a program to achieve this, and execute it on your PLC.

Worked Example 1

Now we want to link in a horn sound to the light. Put the cursor at the left of the desired coil.



Click  to access drop-down list of instruction group. Click scroll bar in drop-down list to position cursor on 'lines', Figure 52.

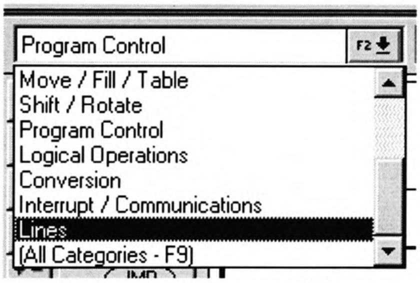


Figure 52 line editing

Click on , Figure 53.

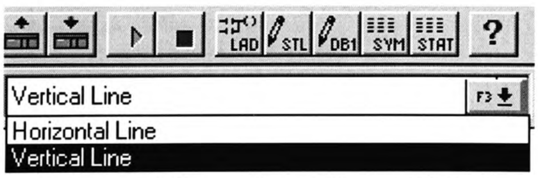
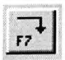
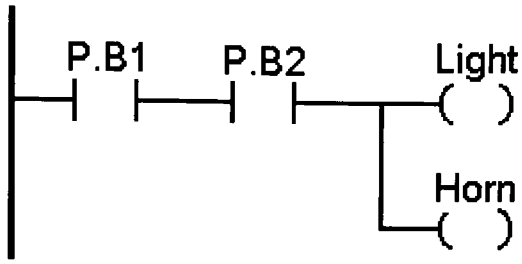


Figure 53 vertical line editing

Click on **Vertical line**

Or click on .

Add the coil which represents the horn, the network now takes the following form



Exercise 3

In industrial applications, different types of inputs are used such as Limit switches, Push buttons, Foot switches and Proximity sensors.

In this example an output is obtained, for example a horn, (Figure 54), when either of two input switches is actuated, table (3).

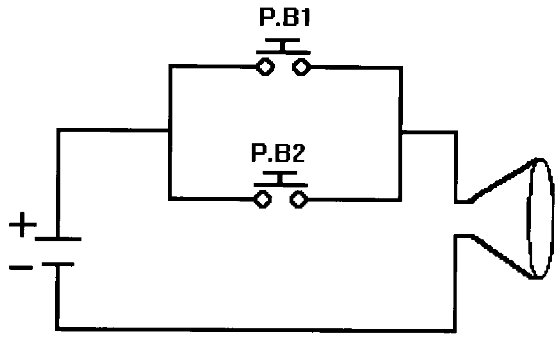
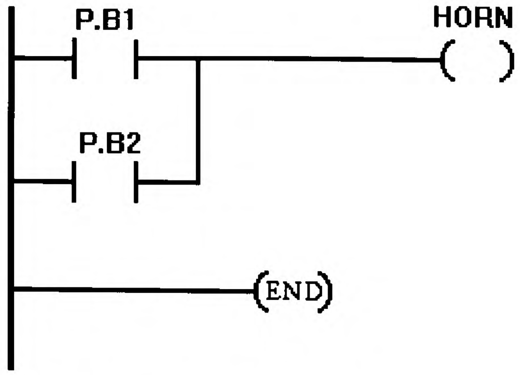


Figure 54 exercise 3

P.B1	P.B2	HORN
0	0	0
0	1	1
1	0	1
1	1	1

Table 3 OR result

Try to produce this program:



Load it into the PLC and execute it.

Exercise 4

The sizes of products are checked by 2 out of 3 switches, Figure 55. If the three switches SW1, SW2 and SW3 are actuated, this means that the product is too big and if only one switch is actuated it means that the product is too small.

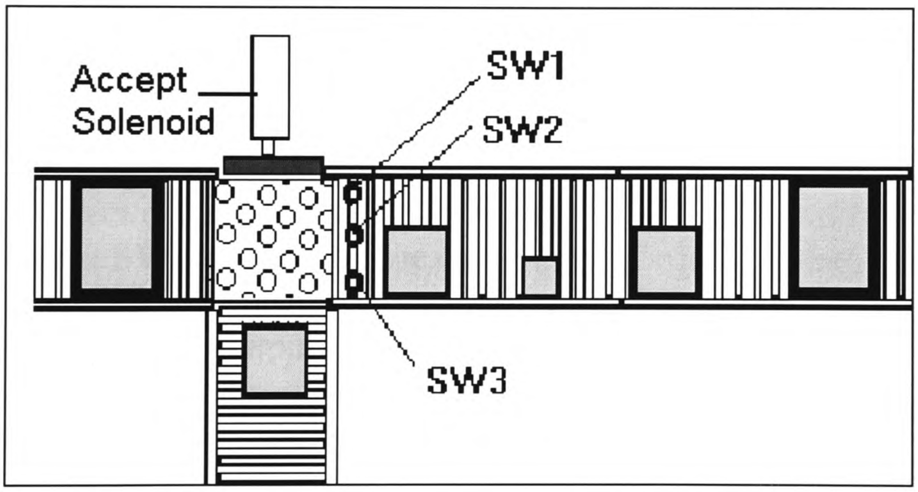


Figure 55exercise 4

Use table 4, to write a control program and execute it on the PLC.

SW1	SW2	SW3	Accept solenoid
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

Table 4 solenoid signal

Worked Example 2

A lamp should be switched **ON** by using a push button SW1. It should remain **ON** until push button SW2 is operated to switch it **OFF**.

Two different methods can be used to achieve this task:
1- Self latching technique.
2- Using **Set** & **Reset.**

1- Using self latching technique:

The network shown in figure (56), consists of a normally open contact SW1 push button in series with a normally closed contact SW2 (SW2 is also a normally open push button), a holding contact of the output (LAMP) is connected in parallel across the SW1 push button.

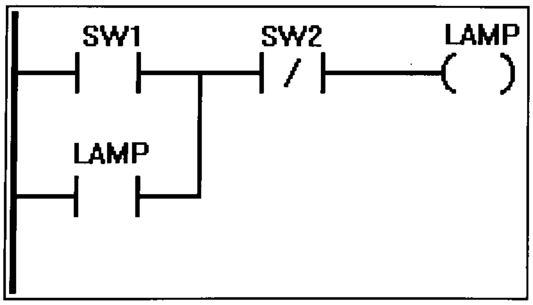
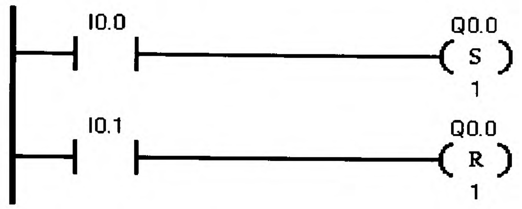


Figure 56 self latching

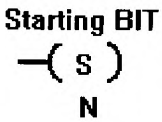
The operation of SW1 will allow power to the coil (LAMP), thus closing the lamp contact. When SW1 is released, the lamp will remain latched in the **ON** condition until the **OFF** push button (SW2) is operated to break the power flow.

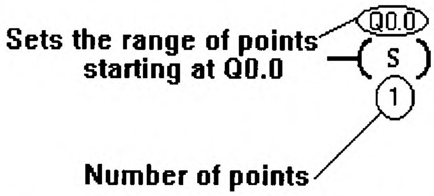
2- Using Set & Reset

In ladder logic, **Set** (Turn on)& **Reset** (Turn off) are used to hold and reset the state of the coil.



S & **R** coils set & reset a range of points starting at the defined **Bit** for the number of points specified by N as follows:





Use Set & Reset instructions, Figure 57, to achieve the previous example:

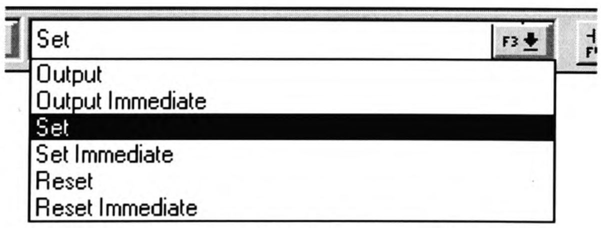


Figure 57 Set instructions

Summary

Normally Open and normally closed Contacts are used in similar ways in logic circuits. The normally closed contact allows power flow when an input signal is absent.

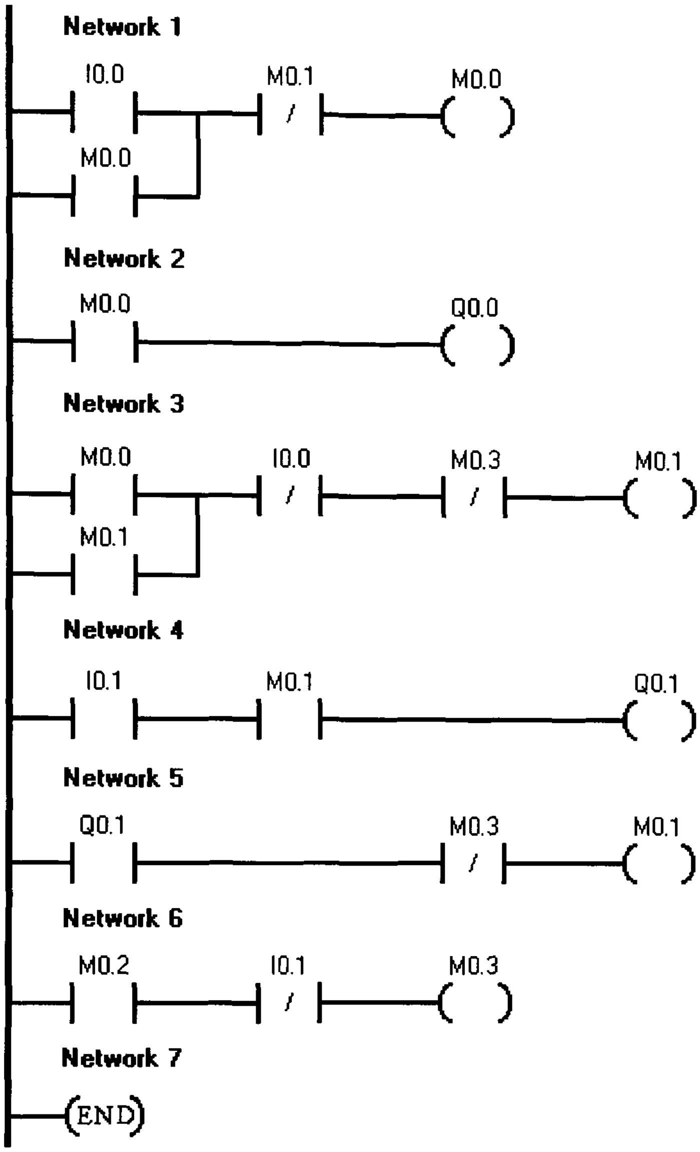
10 MARKERS

This section introduces internal outputs, which we can use to build the logic of a program.

STEP7 provides a group of Markers (known as internal Control relays or Flags in other types of PLCs), which can be used as pseudo outputs plus associated contacts and as general memory for the PLC. PLC systems usually have a much larger number of markers than real outputs.

Exercise 5

SW1 is used to switch LAMP1 on and off. Push button PB2 is used to switch LAMP2 on and off. But LAMP2 can only be switched on if LAMP1 is switched on and then off.
To solve this example we will use internal markers as shown in following program.



Summary
Markers are very useful tools in ladder logic programming. They allow us to improve the logic of programming techniques. Markers are used as internal contacts and/or coils in the program.

11 Status chart

This section explains how to use the ***Status chart*** *tool to monitor the status of program elements. The trainee will be able to trace faults within the program.*

The Ladder Status on tool that we used in section 8 can be used to monitor the status of the program and to identify faults. For one, two, or three networks this tool is sufficient, but to monitor a number of instructions, we need to use another powerful tool called **Status chart**. Figure 58.

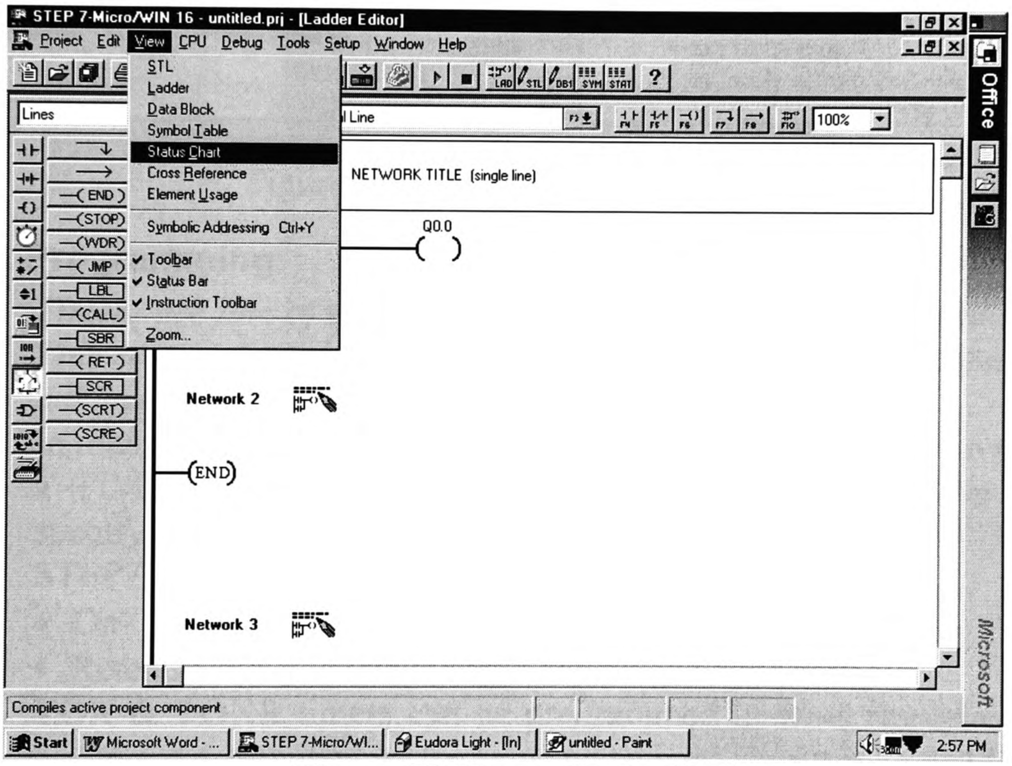


Figure 58 Status chart selection

Fill in the table shown in figure 59 by typing an address such as I0.0

Address	Format	Current Value	Change Value
	Signed		
	Signed		
	Signed		
	Signed		
	Signed		

Figure 59 chart table

To monitor the status of our project click on **Status chart** in **View** figure (60).
And
Click on **Continuous Read** in **Debug** list, figure 58.

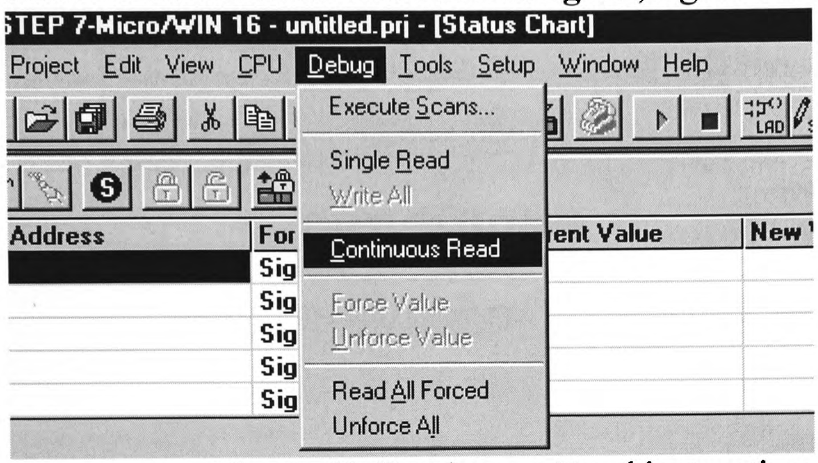


Figure 60 Continuous Read instruction

12 TIMER Programming:

Timers are one of the most powerful tools in PLC programming. This section introduces them, enabling the trainee to use them to achieve timing tasks.

In many types of control applications there is a requirement to use a TIMER. Commonly, in programming a timer we have to specify the timer period and the start event.

STEP7 provides two different timer instructions

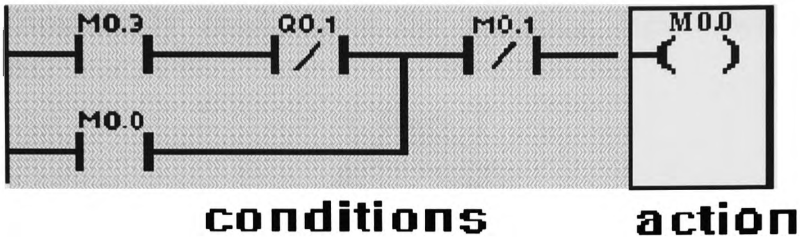
- On- Delay timer (**TON**).
- Retentive On-Delay timer (**TONR**).

TON & TONR timers can be programmed in three resolutions, which are determined by the timer number as shown in table (5).

Instruction	Timer Resolution	Timer number
TON	1 ms	T 32, T96
	10 ms	T33-T36, T97-T100
	100 ms	T 37-T63, T101-T127
TONR	1 ms	T 0, T64
	10 ms	T1-T4, T65-T68
	100 ms	T5-T31, T69-T95

Table 5 timers resolutions and addressing

Timers are represented by means of a box, with the function of the timer being executed when power flows to it.



11.1 On-Delay timer (TON)

TONs in STEP7 are controlled by means of an **Input Signal** (IN) and have a **Preset Value** (PT) as shown in figure 57. TON timers are automatically reset when the enabling input switches off, figure 61.

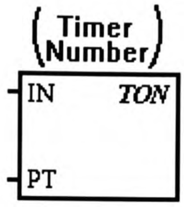


Figure 61 On-Delay timer box

In the next example we will produce a delay of 5 sec before the lamp turns on after the switch (I0.0) has been closed.

To obtain the timer click on  to access the drop-down list of instructions and click on (**Timers/Counters**), Figure 62.

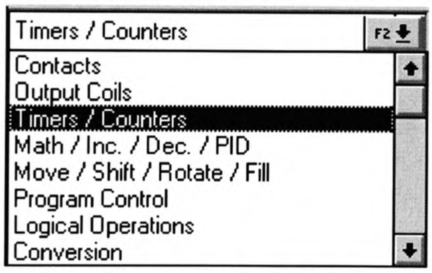
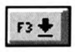


Figure 62 Timers/Counters selection

Click on  to access the drop- down list and choose the desired instruction (Timer-On delay), Figure 63.

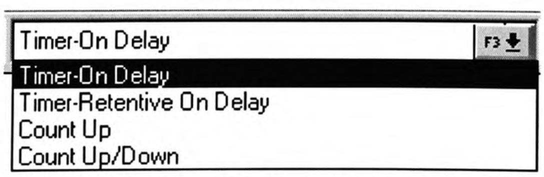


Figure 63 On-Delay timer selection

or use the icons at the left of the screen, figure 64.

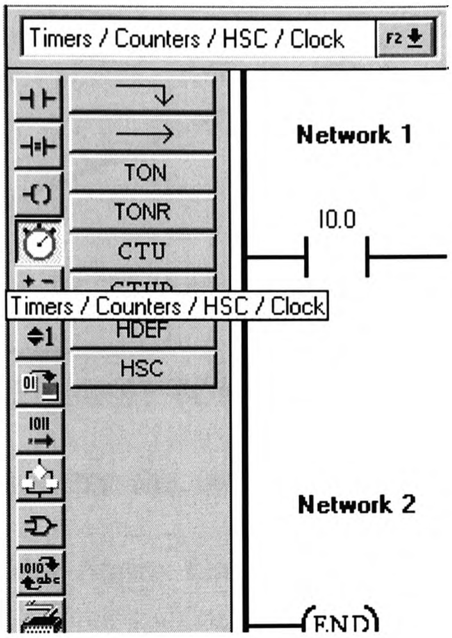


Figure 64 STEP 7 icons

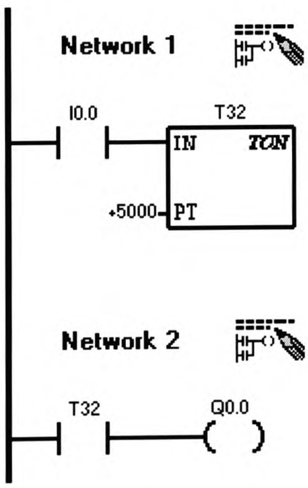


Figure 65 explains the behaviour of the TON. When the **Input Signal** turns on, a period of time later (defined by the preset value) the Timer Bit turns on. When the Input Signal turns off, the Timer Bit also turns off.

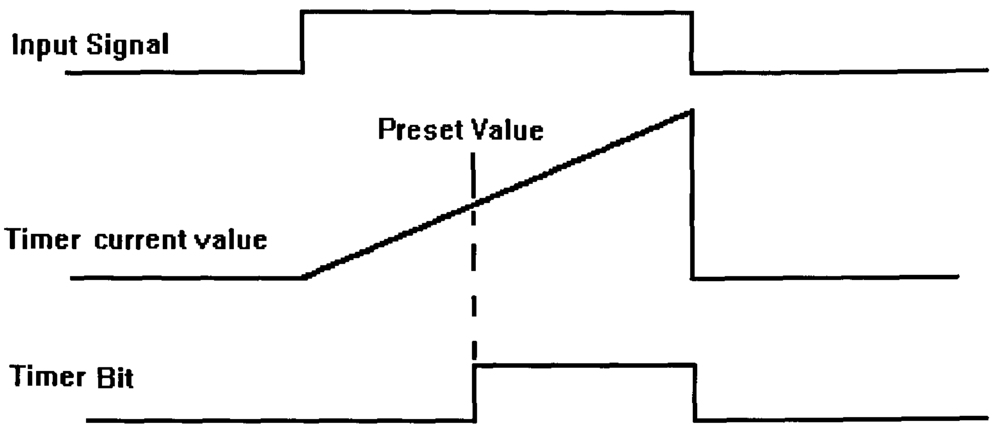


Figure 65 TON behaviour

Refer to table (5) and try the same program using timers T34, T38.
Use the option Ladder Status On to watch the current value in each case.

Write a program to switch the lamp On for a period of 5 sec after a switch SW3 is operated.

11.2 Retentive On-Delay timer (TONR)

A Retentive On-Delay timer (**TONR**), figure 66, is similar to a (TON). It introduces a delay between the instant of switching the **Enabling Input** on and turning the **Timer Bit** on. However, a (TONR) is only reset if a Reset Instruction is used, figure 67.

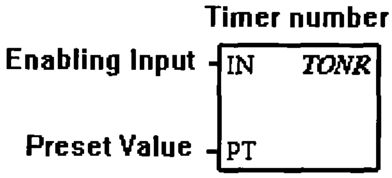


Figure 66 Retentive On-Delay timer box

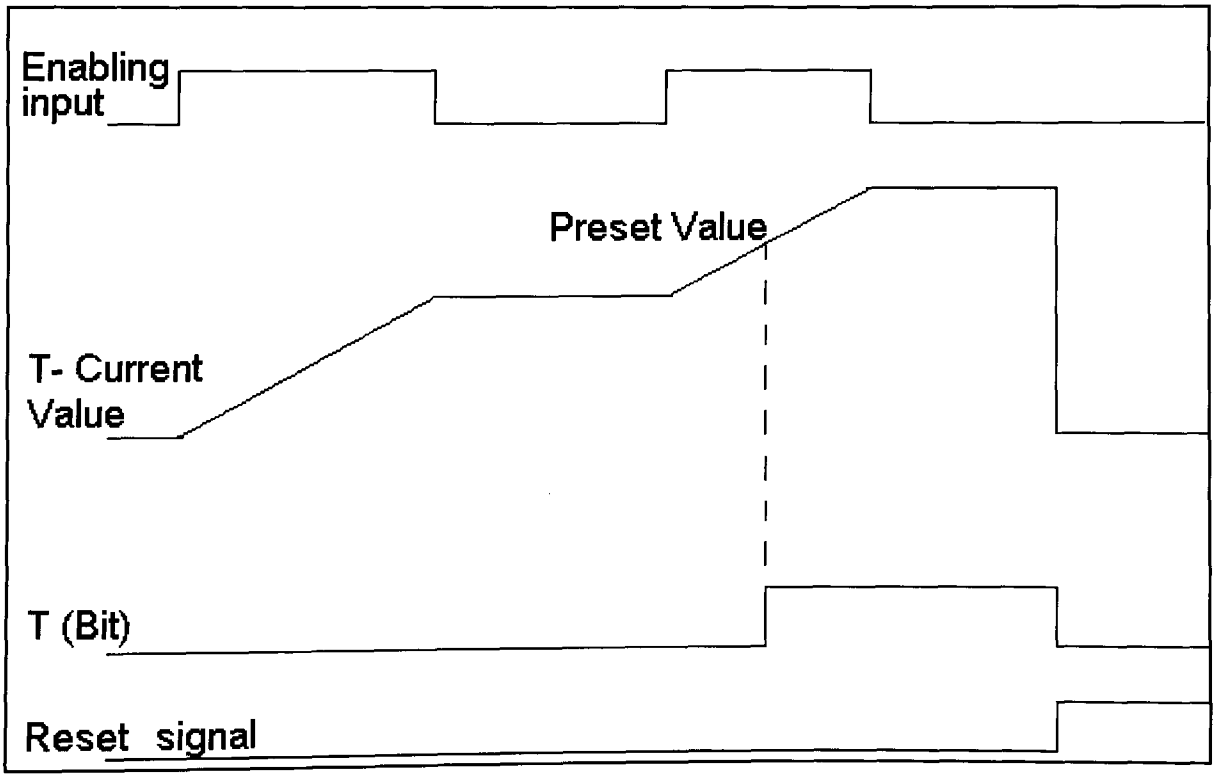
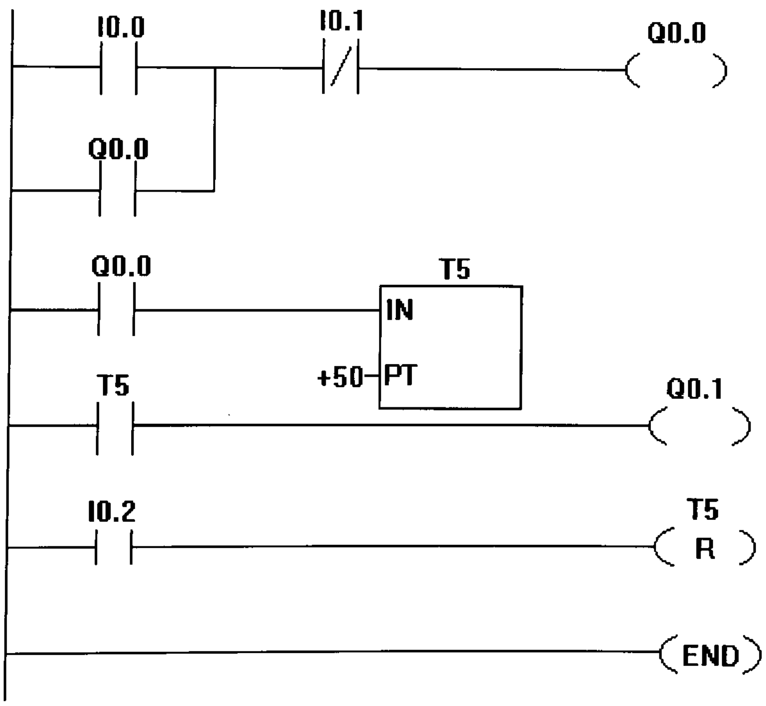


Figure 67 Retentive On-Delay behaviour

Exercise 6

In a control cabinet, a lamp is switched on whenever a fault occurs. After the fault has been repaired the maintenance engineer switches the lamp off. The lifetime of this lamp is defined as 20 hours, which means that the lamp should be changed after 20 hours of operation regardless of the number of times that the lamp has been in operation. After changing the lamp, the timer (which records the lighting time) should be reset.

In the next program **I0.0** represent the fault signal which causes the lamp (output **Q0.0**) to switch on and Timer **T5** records the lighting time of the lamp. After repairing the fault the maintenance engineer switches the lamp off by using a normally open push button **I0.1**. When the current value of the timer equals 20 hours, an output signal **Q0.1** gives a signal to the maintenance engineer to change the lamp. A normally open push button switch **I0.2** is used to reset the timer. Try the following program:



Summary

STEP 7 provides two timer instructions:
On-Delay timer (TON).
Retentive On-Delay timer (TONR)
Each can be programmed in three resolutions, 1ms, 10ms, 100ms. They have an input signal and a preset value. A Box placed in the output area of the ladder diagram represents the timer. (TON) timers are automatically reset, while (TONR) are only reset if a reset instruction is used.

12 COUNTERS

This section introduces a very important tool for programming. After this section the trainee should be able to write a program to achieve tasks which depend on counting events or actions.

Counters are used for counting the number of occurrences of a signal. The S7-200 PLC is able to perform both up and down counting. The programming of a counter requires an input pulse, the target value (preset value) and a reset for the counter.

12.1 Count Up (CTU)

Up counters count up to the preset value. When the preset value is reached the counter's contact changes state.

The S7-200 provides an **up counter** (**CTU**) which counts up each time the count up input switches from off to on and resets when the reset input turns on. CPU 212 for example, has 64 counters (see table 1).

Each CTU has a current value that holds the current count and a preset value (**PV**). When the current value becomes equal to or greater than the preset value the counter bit turns on. The range of each **CTU** is from 0 to 32767.

Exercise 7

A lamp should be switched on after a switch SW1 has been actuated 5 times.

To solve this example use a normally open contact to represent SW1, and a count up counter. Click on **Timers/Counters** as shown in figure 68.

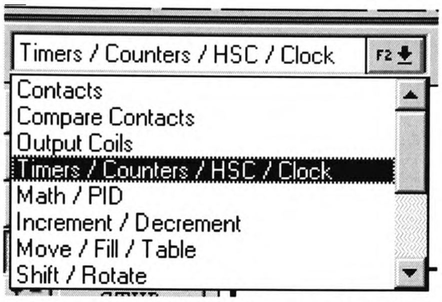


Figure 68 Counters selection

and click on count Up, Figure 69.

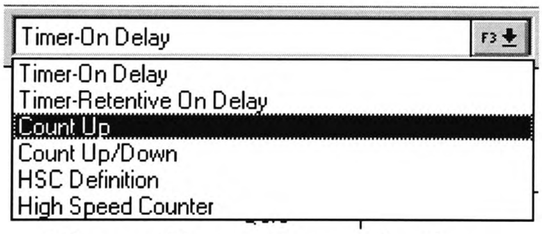
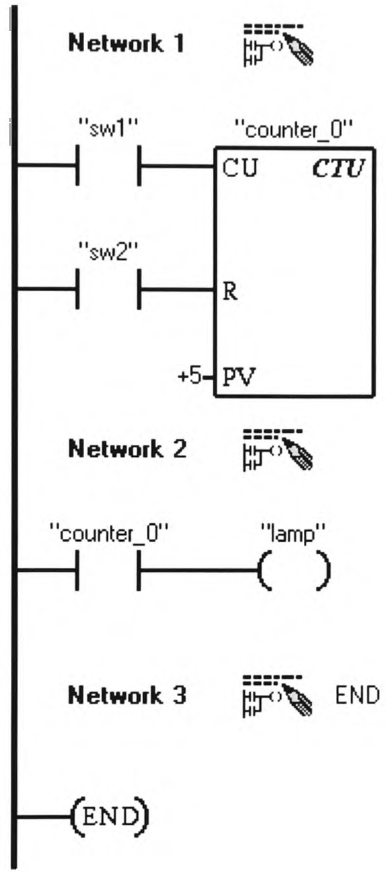


Figure 69 count up selection

To reset the counter use SW2, connected to **R** on the counter. Finally, the program should look like:



Referring to exercise (5), write a program to count the number of accepted items.

12.2 Counting up and down (CTUD)

In addition to **counting up** with CTU, the S7-200 PLC has Up/Down counters, **CTUD,** Figure 70. In this type, the counter counts down each time the count-down input is actuated and counts up when the count up signal is actuated, Figure 71.

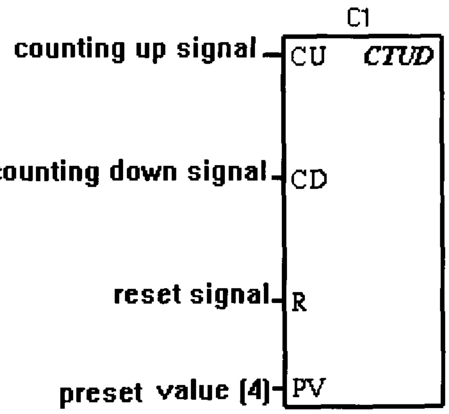


Figure 70 Counting up and down box

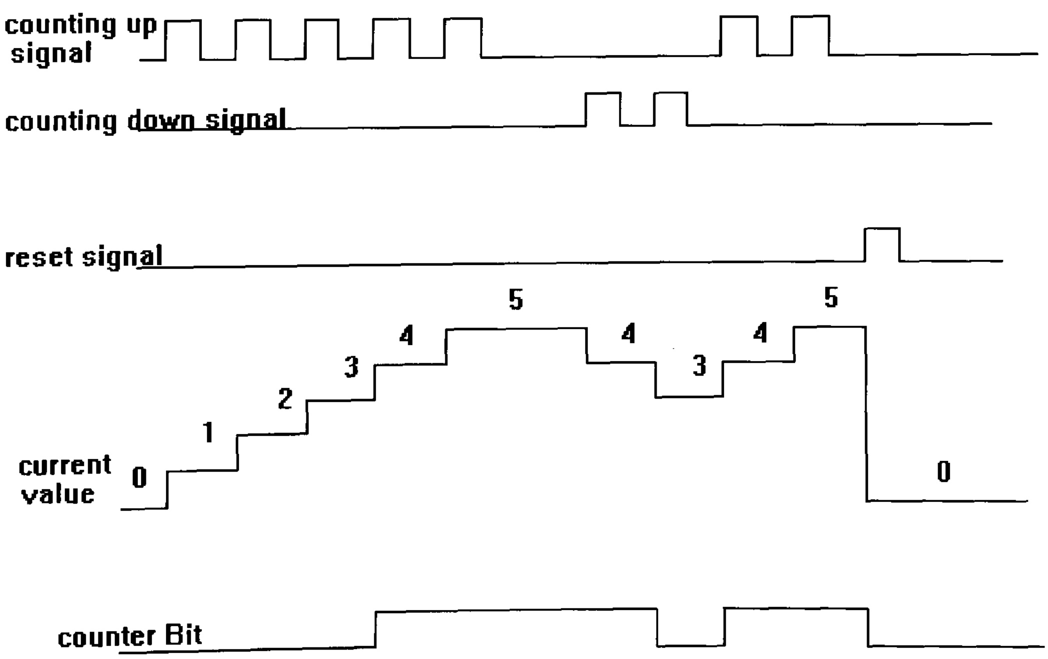
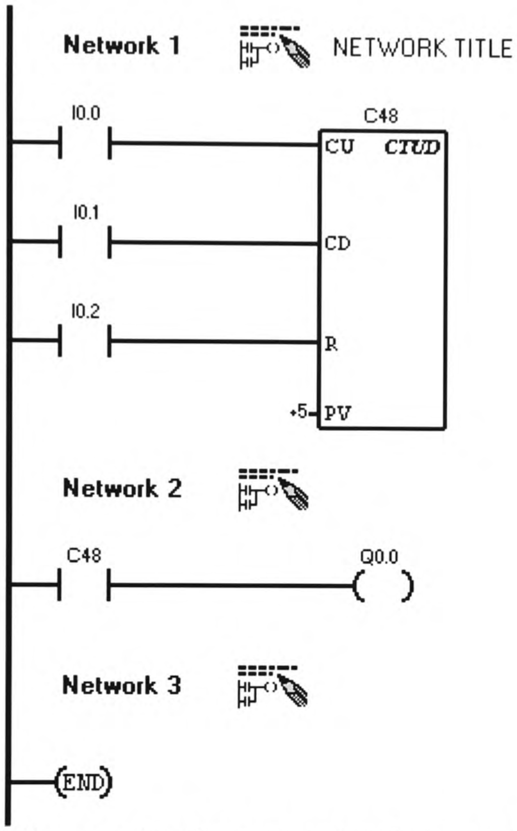


Figure 71 Counting up and down behaviour

Programming of CTUD is similar to CTU, try the following program:



Watch the changes in the current values on the PC screen.

Exercise 8

Write a program to switch a lamp on (for 2 sec) and off 3 times with an interval of 3 sec between each operation. At the end you should be able to start the program again by using a push button SW1 (Figure 72).

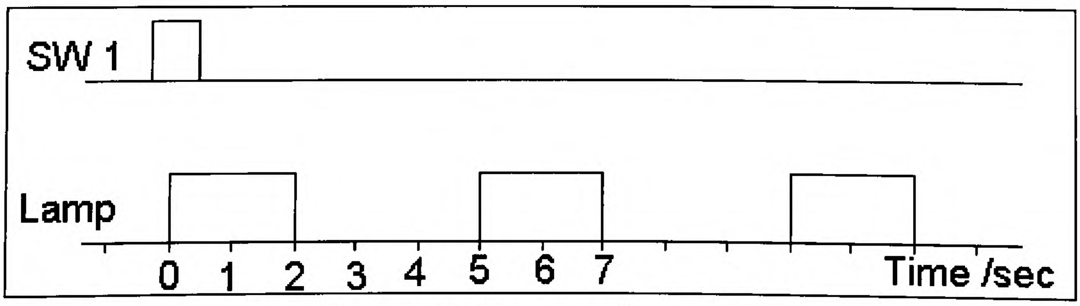


Figure 72 Exercise 8

Summary

STEP7 provides two types of counters:

- *Count Up (CTU).*
- *Count Up and Down (CTUD).*

Each of them has a current value and preset value. When the current value becomes equal to or greater than the preset value the counter Bit turns On.

(CTU) count up each time the count up input is switched from off to on and resets when reset input turns on.

(CTUD) in addition to counting Up, this type counts down when the countdown signal comes On.

13 MOVE BYTE

This section introduces an instruction that can be used in high level programming techniques. The trainee can use this instruction to save programming time or move whole bytes instead of moving separate bits. Also, this instruction can be used when working with arithmetic instructions.

Exercise 9

In a paint factory, 7 different colours must be mixed to produce a new colour, Figure 73. The different colours are stored in tanks and each tank has its own valve.

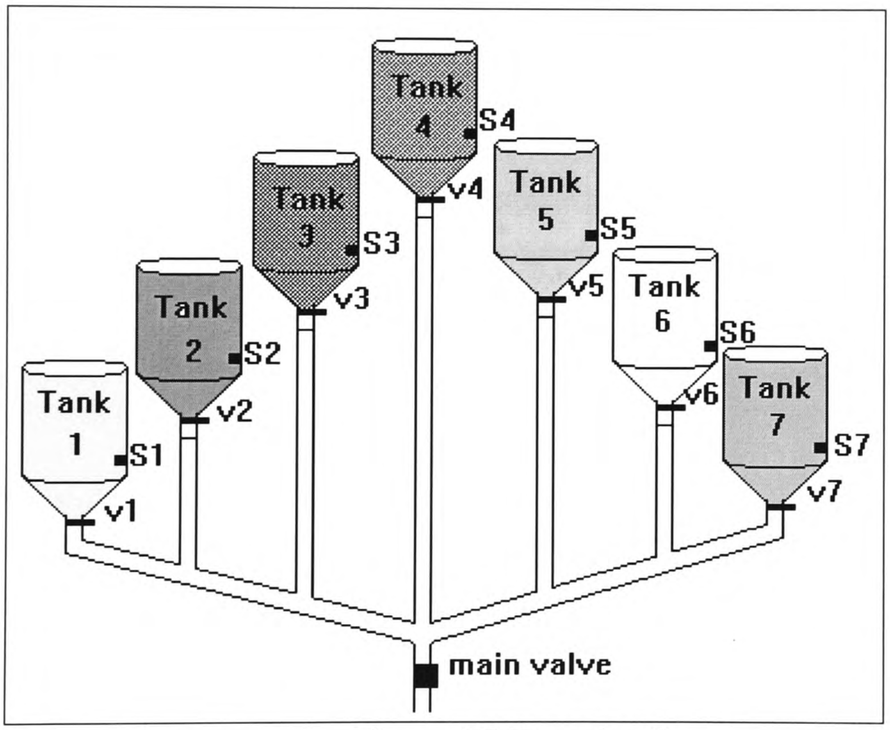
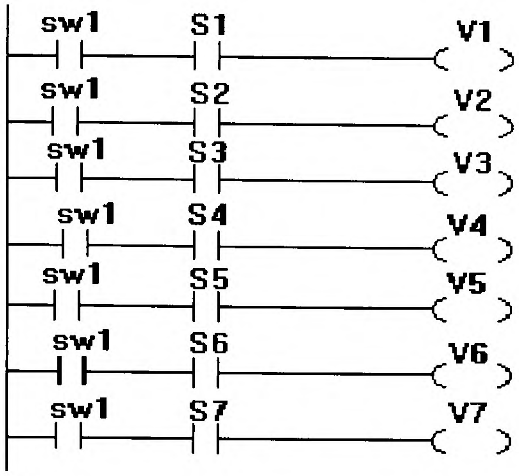
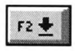


Figure 73 Exercise 9

The tanks are provided with sensors S1 to S7 to indicate the state of the tanks (Not Empty). To start the operation a switch SW1 enables the valves to open according to the state of the tank. S1 is addressed as I0.0, S2 as I0.1, S3 as I0.2, etc. and SW1 is I0.7. As shown in the following program:



Instead of this long program we can use a move instruction to achieve the same result.
A Move box can be obtained by:

Clicking on  to access the instruction list, figure 74.

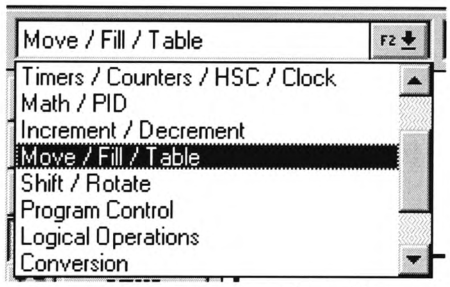
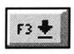


Figure 74 instruction list

And clicking on , Figure 75.

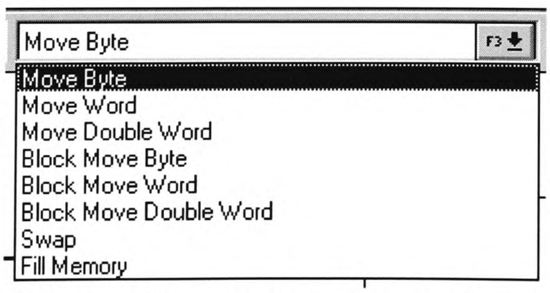
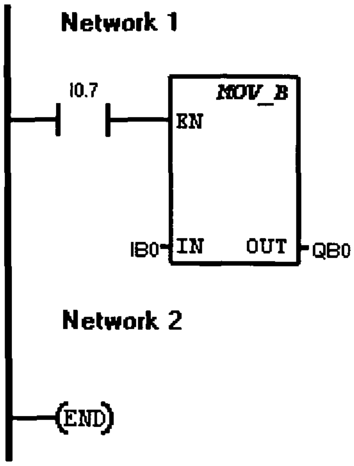


Figure 75 move byte instruction

Try the following program. Where the closure of the SW7 causes the state of the Input Byte 0 to Move to the Output Byte, QB0.



Where I0.0, I0.0,......, I0.6 are all contained in Byte 0.

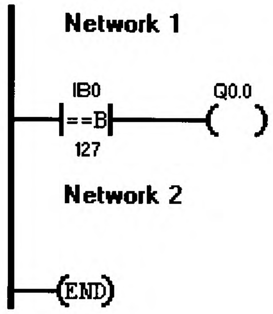
Summary

Move Byte is an important instruction for data handling. Using this instruction 8 Bits can be manipulated.

14 Compare Instructions

This section explains the use of compare instructions. The trainee uses this instruction in cases where comparison is necessary to achieve the control task.

Compare instructions are used to make a control decision based upon the result of a comparison. To clarify the use of the compare instructions load this program into your PLC.



Compare instructions can be obtained by choosing the compare contact group, Figure 76.

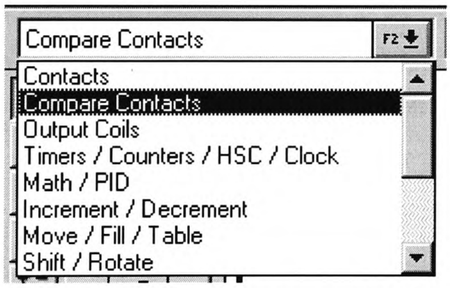


Figure 76 compare contact group

Or by clicking on the proper icon, figure 77

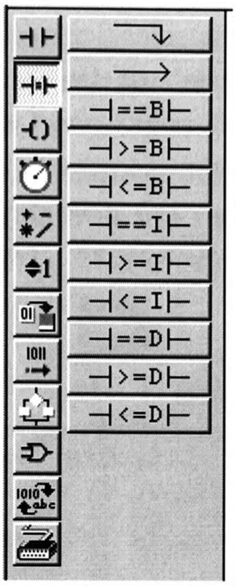


Figure 77 STEP7 icons

Switch the PLC into RUN mode and watch the changes on the screen. [**Ladder Status On**].
Compare contacts can be used to perform comparisons between Bytes, words and double words.
In STEP7 we can compare for less than or equal, equal, and greater than or equal.

Exercise 10

A car park, Figure 78 has three entrances and three exits, In each gate there is a sensor to count the number of cars entering or leaving the car park, which has a limited capacity.
Write a program to count the numbers of cars that enter and leave the park. Compare the number in the park with the park capacity so as to provide a signal at the entrances when the park is full.

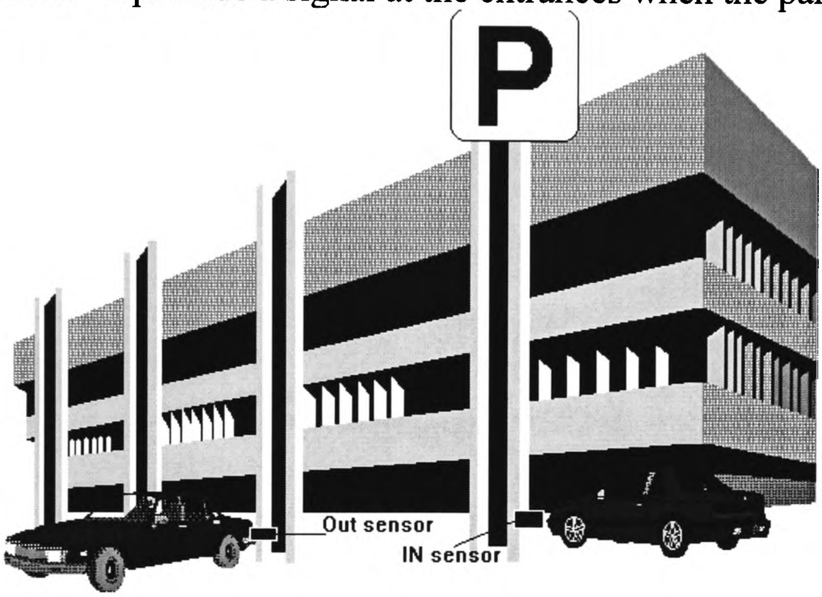


Figure 78 Exercise 10

There are a number of ways to solve this example. One way is to use the memory of your PLC to store the capacity of the park. We can also use **CTUD.**

Summary
Compare instructions can be used to compare the values stored in Bytes, Words, and Double Words. In STEP 7 we can compare for less than or equal, equal, and greater than or equal.

15 Math Instructions

This section introduces the applications of the math instructions. At the end of this section the trainee will be able to use math instructions to perform many tasks such as add, subtract, multiply and divide.

S7-200 has the capability to perform arithmetic operations. By using the Math Instructions, which are introduced in table 6, we can perform many tasks.

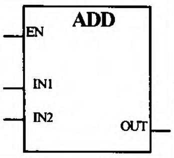
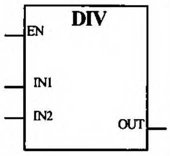
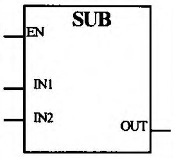
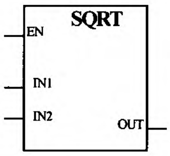
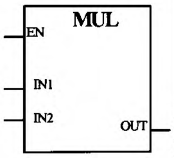
LAD Instruction	Description	**LAD** Instruction	Description
	Add When EN actuated IN1 + IN2 = OUT		Divide When EN actuated IN1/ IN2 = out
	Subtract When EN actuated IN1 - IN2 = OUT		Square root When EN actuated / IN = OUT
	Multiply When EN actuated IN1 * IN2 = OUT		

Table 6 STEP7 arithmetic operations

Math Instructions can be found in STEP 7 (see Figures 79 and 80)

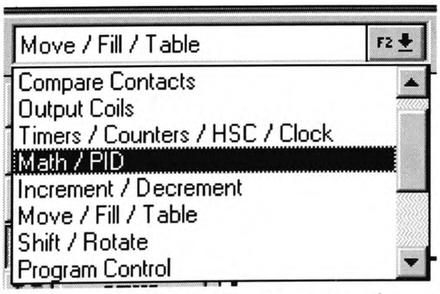


Figure 79 Math Instructions

And

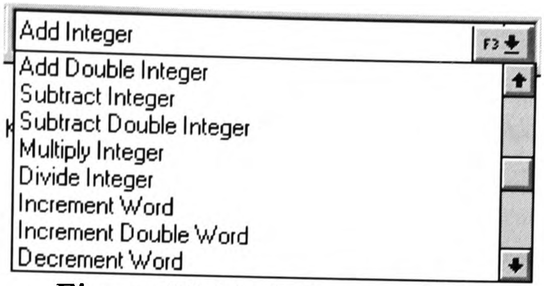
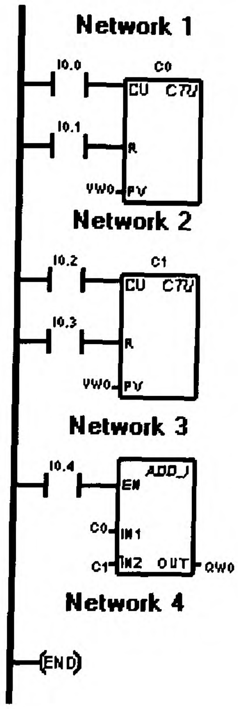


Figure 80 Math Instructions

Exercise 11

Execute the following program, where I0.0,..,I0.4 are normally open push buttons. Switch I0.4 causes the addition of the values in counter 1 and counter 2.



Fill in the status chart as shown in Figure 81.

Address	Format	Current Value	Change Value
C0	Signed		
C1	Signed		
QW0	Signed		
	Signed		
	Signed		

Figure 81 status chart

Monitor the changes on your screen using **Continuous Read**

Exercise 12

From exercise (4), calculate the percentage of the accepted items using this equation

$$\frac{C0}{C0 + C1} \times 100$$

Where
C0 = the number of accepted items.
C1 = the number of rejected items.

Summary

Math instructions can be used in STEP7 to perform different mathematical tasks. they are represented by Boxes where the OUT of the Box give the result of the operation that has been performed on the two inputs, IN1 and IN2, when the EN signal is actuated.

16 Special Memory (SM)

One of the tools of S7-200 is Special Memory. This section introduces some of these special memories.

The S7-200 is equipped with certain special memories that provide useful facilities for control functions. These facilities can be used in programs to create certain actions depending on their locations in the PLC memory. Special memory may be used in Bit, Byte, Word, or Double word format.

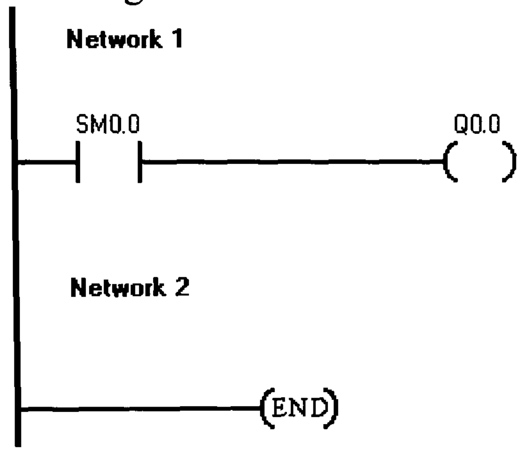
Table 7, lists some of these special memories.

Status Bit	Description
SM 0.0	Always ON
SM 0.1	ON for the first scan
SM 0.2	ON for one scan if the retentive data is lost
SM 0.3	ON for one scan when RUN mode is entered from a power up condition
SM 0.4	ON for 30 sec and OFF for 30 sec.
SM 0.5	ON for 0.5 sec and OFF for 0.5 sec
SM 0.6	ON for one scan and off for the next scan
SM 0.7	ON if the Mode switch in RUN position and OFF if it's in TERM position
SM 1.0	ON when the result of the operation is zero
SM 1.3	ON when division by zero is attempted

Table 7 Special memory

Exercise 13

Load the following program to the PLC. Watch output Q0.0 after switching the PLC to the **RUN** mode.



17 Subroutine Instructions

At the end of this section the trainee should be able to write a program incorporating a subroutine.

Whenever an operation is required several times in the main program, it can be written once at the end of the program and then called every time we want to use it. By using subroutines we can save programming time and memory usage.

Calling of subroutines is not limited to just the main program. A called subroutine may itself call another (to a depth of eight).

When we call a subroutine (by using **(CALL)** instruction) the execution moves from the main program to the subroutine. On completion, the instruction **(RET)** returns the execution to the main program at the instruction following the (CALL) instruction.

Exercise 14

Refer to exercise 4. Calculate the percentage of the accepted items for every 100 items. We need:

- Counter 0 **(C0)** counts the accepted items.
- Counter 1 **(C1)** counts the rejected items.
- The total number of items produced by using **ADD** instruction (C0+C1).
- The result stored at MB1.
- If MB1= 100, call the subroutine which calculates the percentage ratio.
- The value 100, which is used to calculate the ratio, should be loaded into MW2.

To achieve this task, program the following.

Select **Data Block** from the **View** list, Figure 82.

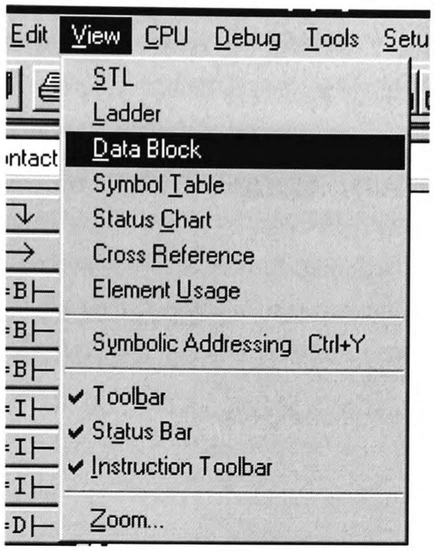


Figure 82 Data Block selection

Set the value of **VW0** as shown in Figure 83.

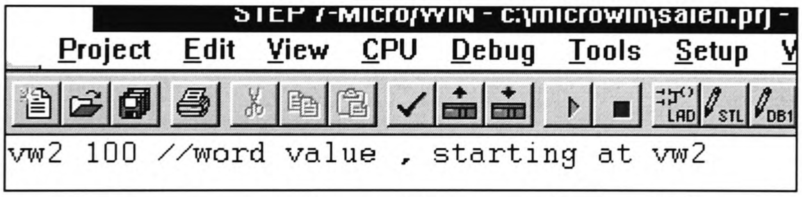


Figure 83 Data Block editing

Click on **Status Chart** from the **View** list, Figure 84, and type the following:

Address	Format	Current Value	Change Value
C0	Signed		
C1	Signed		
VW0	Signed		
VW2	Signed		
VW4	Signed		
VW6	Signed		
VW8	Signed		
VD4	Signed		
VD8	Signed		
VD12	Signed		

Figure 84 Status Chart

Load the project to the PLC. It is necessary here to download the **Data Block.** Switch the **Chart Status On,** Figure 85, and watch the changes while you using the switches I0.0, I0.1. For a quicker result you may change the value in VW2

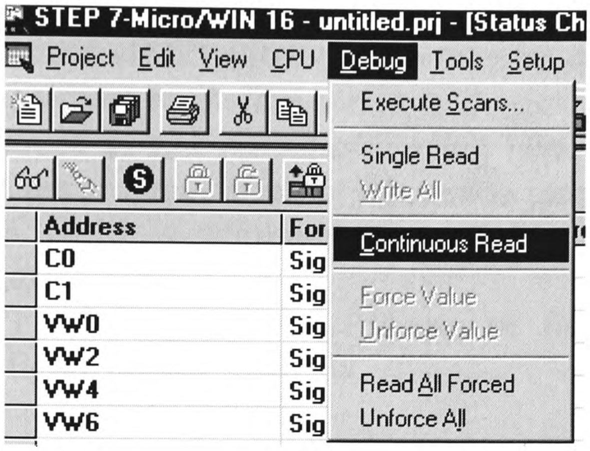


Figure 85 Switching **Chart Status On**

18 Jump Instructions

This section introduces an instruction which can be used to jump over some other instructions. The trainee will be able to write a program and skip some actions according to the state of a switch which actuates the jump coil.

Jump instructions (JMP) allow you to transfer control from one point of the program to another corresponding to a label (LBL).

A Jump instruction and its corresponding label must be in the same part of the program. It is not allowed to jump from the main program to a label in a subrotine or jump from a subrotine to the main program.

Jump (JMP) instructions and LBL can be found in program control list, figure 86 and figure 87.

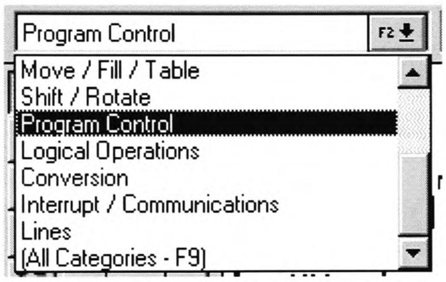


Figure 86 program control list

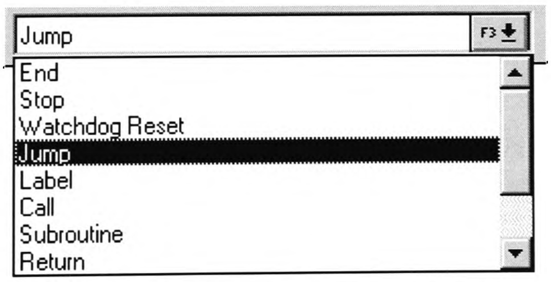
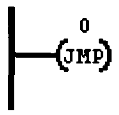
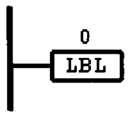


Figure 87 Jump instructions

The Jump coil (JMP) performs a branch to the specific label (n) within the program, figure 85. The label defines the location of the jump destination (n), where n = 0 to 63

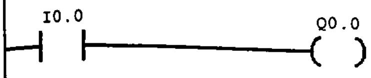




To understand the operation of jump instructions try the next program.

JUM INSTRUCTION

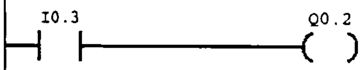
Network 1 I 0.0 actuate Q0.0



Network 2 I 0.1 actuate Q0.1

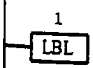
Network 3 If I0.2 actuated then Jump to LBL 1

Network 4 I 0.3 actuate Q0.2

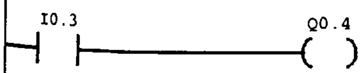


Network 5 I 0.4 actuate Q0.3

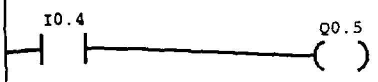
Network 6 LBL 1



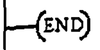
Network 7 I 0.3 actuate Q0.4



Network 8 I 0.4 actuate Q0.5



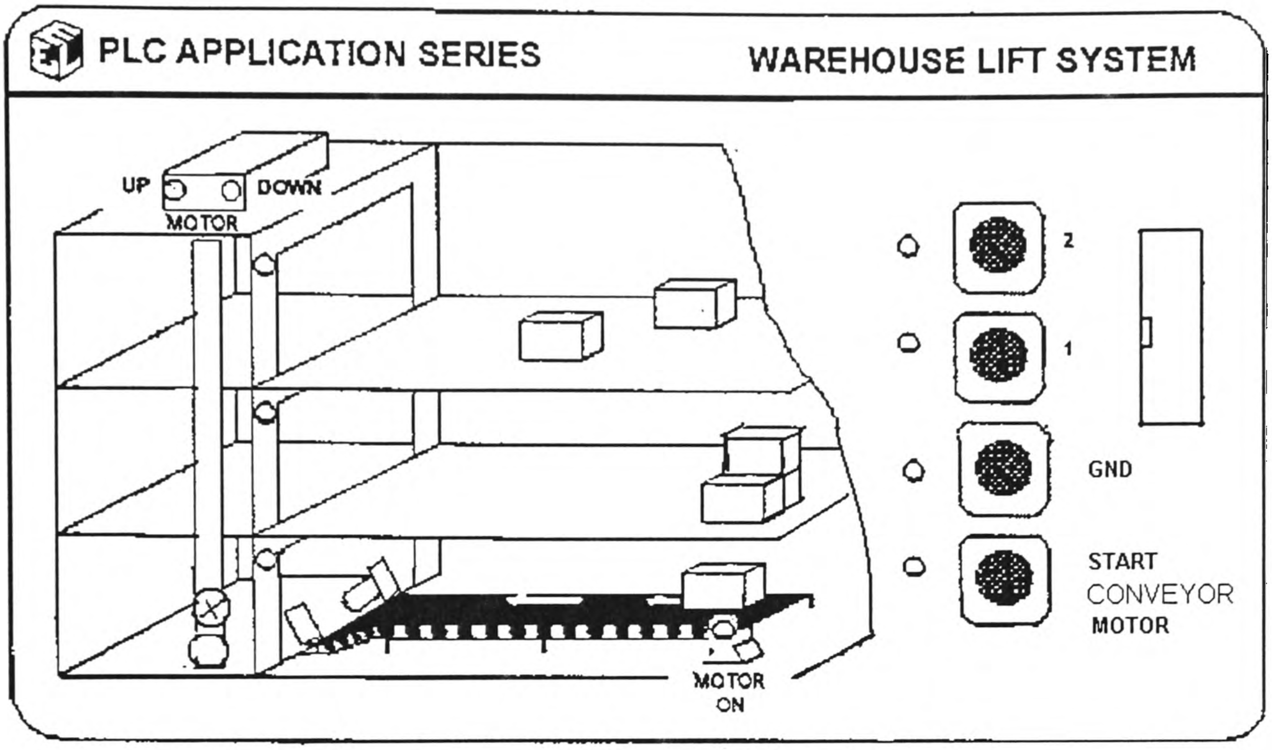
Network 9 End of the program



PLC APPLICATIONS

PLC APPLICATION

WAREHOUSE LIFT SYSTEM



Introduction
This simulator represents a warehouse lift system.
Correct operation of the system should be as follows.
On pressing the " **START CONVEYOR MOTOR** " push button the **MOTOR ON** LED will light. Goods passing along the conveyor into the lift are simulated by passing an object (e.g. a finger or piece of paper) between the **two sensors** on the end of the conveyor.
Now we can select a floor to which the goods are to be moved, by Pressing either **GND, 1, or 2.** An LED on each of the floors will light to indicate which floor has been selected. Once floor has been selected the **"UP"** motor (green LED) will light (by moving the small knob on the left hand side of the board up or down we simulate the movement of the lift. A switch attached to the knob operates as it passes each floor. When the lift reaches the requested floor, the LED will go out. The process is reversed to bring the lift back down to the ground floor.

Programming Commands required

In programming the warehouse Lift System you will need to use:
Normally Open Contacts, Normally Closed Contacts, Coils, Markers, Counter, Compare, Special Memory.

Programming Tasks

Before you start programming move the knob up and down and note the state of the input 0.0 on the front of the PLC in order to familiarise yourself with its operation.

Task1

Write a programme so that the (MOTOR ON) LED flashes on and off If the START **CONVEYOR MOTOR** button **(I0.7)** is pressed once. To stop the action of the LED the **GND** button (I0.1) must be pressed.

Task 2

Add to the program from **Task 1** so that when a specified quantity of cases is loaded into the lift (e.g. 5) as indicated by sensor I0.6, the motor stops.

Task 3

Write a programme, so that when the lift is on the ground floor the LED indicator (Q0.3) for that floor is lit. Do this for all floors (Q0.4, Q0.5).

Task 4

Add to the program from **Task 3** a counter that counts up to 2 when the lift is moved upwards and count back down to 0 when the lift is moved downwards.

List of Inputs and outputs

Input 0 (I0.0)	Lift detector (Normally closed switch)
Input 1 (I0.1)	Ground floor pushbutton (GND)
Input 2 (I0.2)	First floor Pushbutton (1)
Input 3 (I0.3)	Second floor pushbutton (2)
Input 6 (I0.6)	Goods detector
Input 7 (I0.7)	START CONVEYOR MOTOR
Output 0 (Q0.0)	Lift motor Upwards (Green LED)
Output 1 (Q0.1)	Lift motor Downwards (Red LED)
Output 2 (Q0.2)	Conveyor motor MOTOR **ON** (Red LED) and start conveyor motor (Green LED)
Output 3 (Q0.3)	Ground level (2 Red LED)
Output 4 (Q0.4)	First level (2 Red LED)
Output 5 (Q0.5)	Second level (2 Red LED)

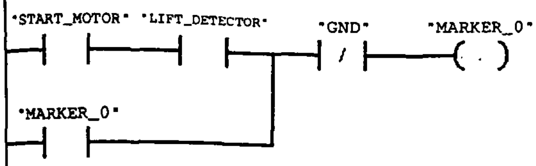
PLC APPLICATION

WAREHOUSE LIFT SYSTEM

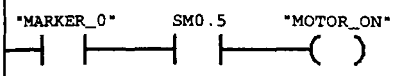
EXAMPLE PROGRAM FOR **TASKS 1- 4**

PLC APPLICATION
WAREHOUSE LIFT SYSTEM
TASK 1

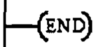
Network 1 STARTING THE MOTOR LED



Network 2 USING SM0.5 TO FLASH THE LED

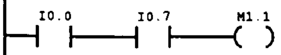


Network 3 END OF THE PROGRAMME

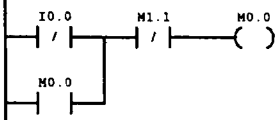


PLC APPLICATION
WAREHOUSE LIFT SYSTEM
TASK 3

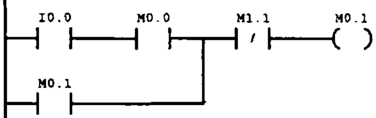
Network 1



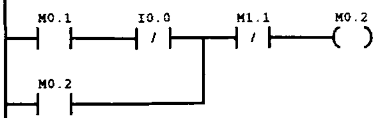
Network 2



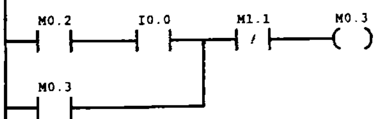
Network 3



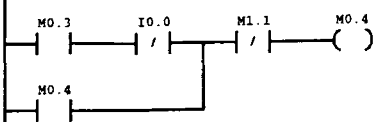
Network 4



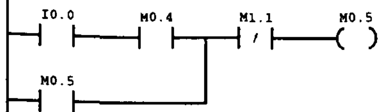
Network 5



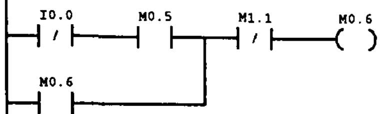
Network 6



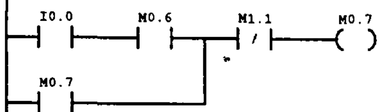
Network 7



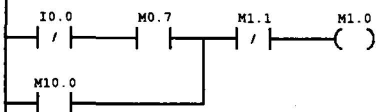
Network 8



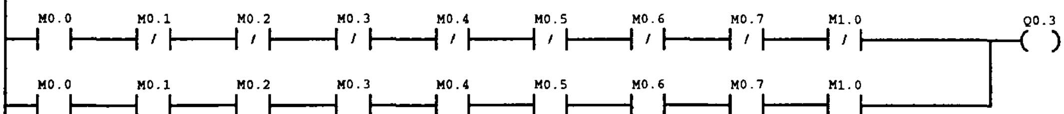
Network 9



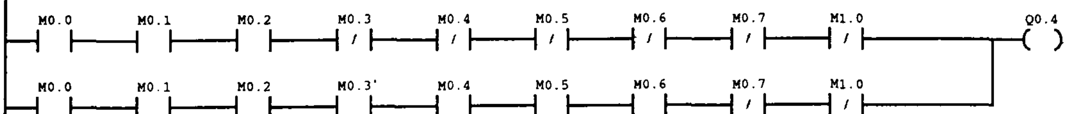
Network 10



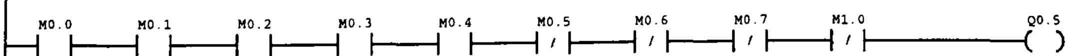
Network 11



Network 12



Network 13

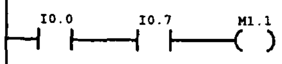


Network 14

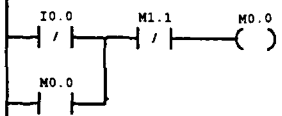
—(END)

PLC APPLICATION
WAREHOUSE LIFT SYSTEM
TASK4

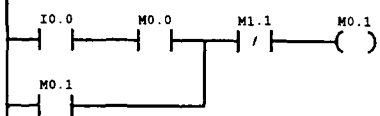
Network 1 MARKER 1.1 TO RESET MARKER 0.0



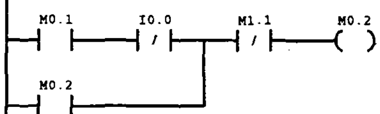
Network 2 THE LIFT ON THE GND FLOOR



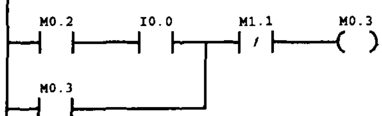
Network 3 THE LIFT IS MOVING TO THE FIRST FLOOR



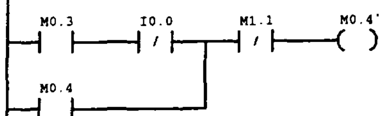
Network 4 THE LIFT ON THE FIRST FLOOR



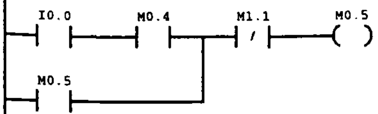
Network 5 THE MOVEMENT FROM FIRST TO SECOND FLOOR



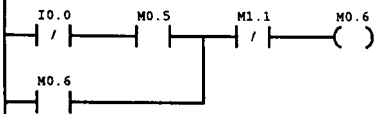
Network 6 THE LIFT ON THE SECOND FLOOR



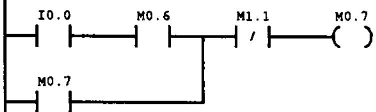
Network 7 MOVEMENT FROM SECOND FLOOR TO FIRST FLOOR



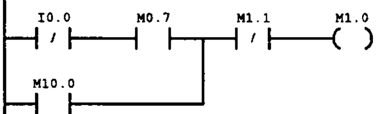
Network 8 THE LIFT ON THE FIRST FLOOR



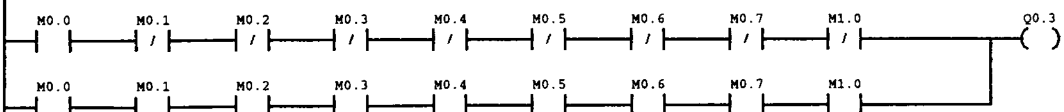
Network 9 MOVEMENT FROM THE FIRST FLOOR TO THE GND FLOOR



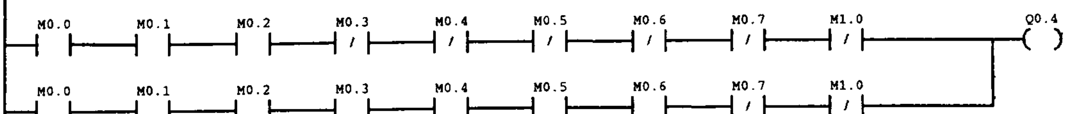
Network 10 THE LIFT ON THE GND FLOOR



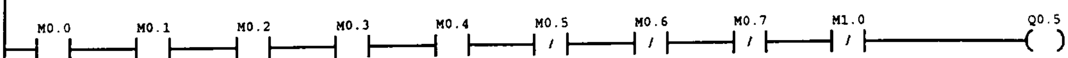
Network 11 Q3 INDICATE THAT THE LIFT ON THE GND FLOOR



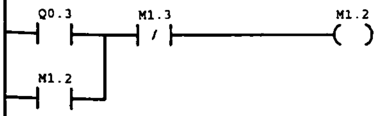
Network 12 Q4 INDICATE THAT THE LIFT ON THE FIRST FLOOR



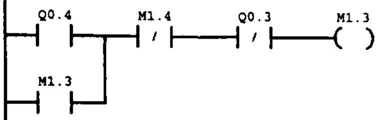
Network 13 Q5 INDICATE THAT THE LIFT ON THE SECOND FLOOR



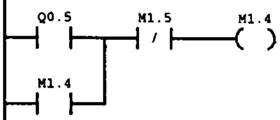
Network 14 MARKER 1.2 ACTUATED IF THE LIFT ON THE GND FLOOR



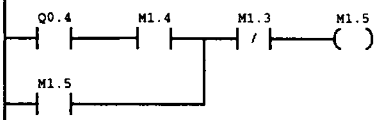
Network 15 MARKER 1.3 ACTUATED IF THE LIFT MOVED FROM GND TO FIRST FLOOR



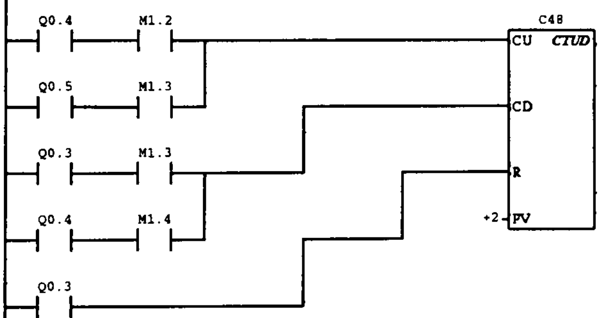
Network 16 MARKER 1.4 ACTUATED IF THE LIFT ON THE SECOND FLOOR



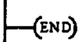
Network 17 MARKER 1.5 ACTUATED IF THE LIFT MOVED FROM SECOND TO FIRST FLOOR



Network 18 LIFT COUNTER

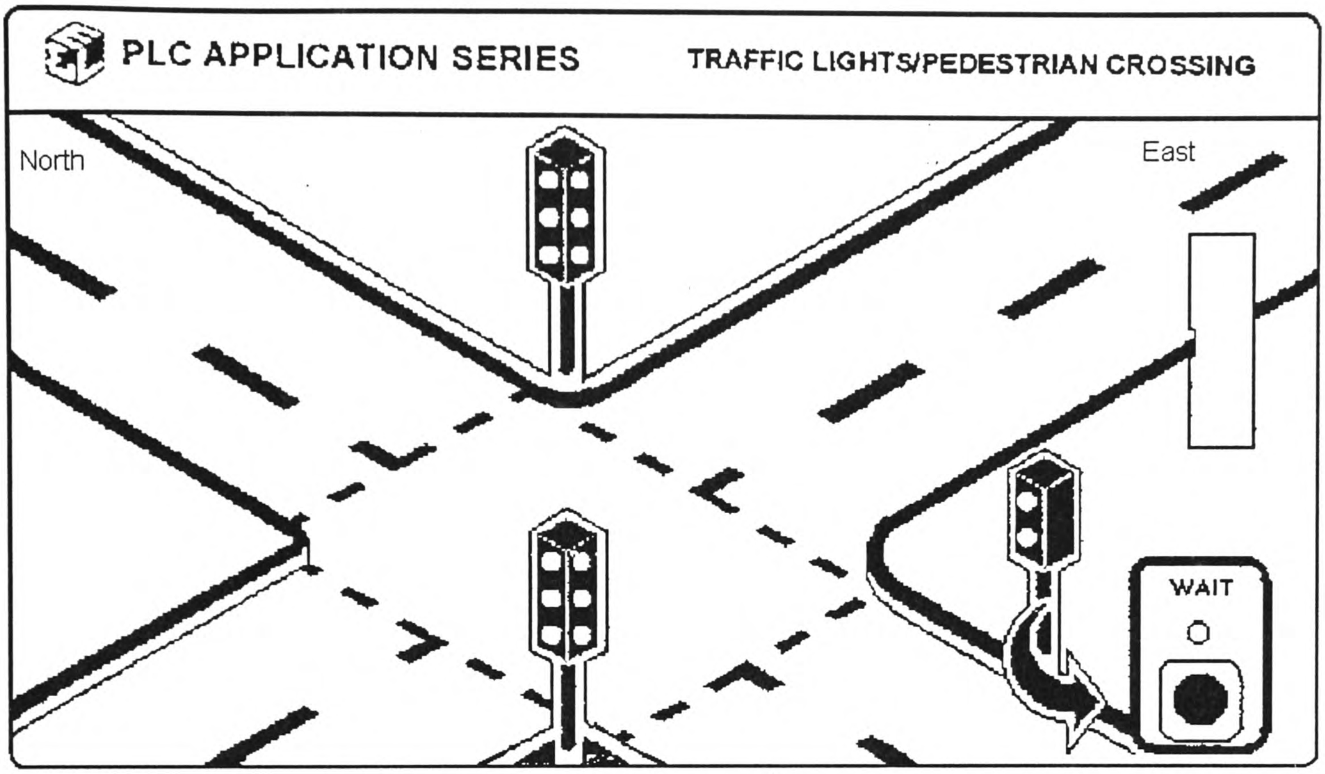


Network 19 END OF THE PROGRAMME



PLC APPLICATION

TRAFFIC LIGHT / PEDESTRIAN



Introduction
This board simulates a set of Traffic Lights at a crossroad with a pedestrian crossing on one of the roads.

Correct operation of this board should be as follows:
The sequence of the traffic lights is to be the same as in real life. Red, red and amber green, amber, red.
The crossroads are referred to as North way and East way. North way being from bottom right to top left and East way from bottom left to top right.
The pedestrian crossing consist of a red light for don't walk and a green light for walk.

Programming Commands
In programming the traffic lights we will need to use the following commands:
Normally open contact, normally closed contact, coil, marker, timer.

Programming tasks

Note that the following LED's are connected internally:

- Both Amber.(Q0.0)
- Red north and Green Pedestrian.(Q0.2)
- Green north and red pedestrian.(Q0.4)

Task 1

Write a program so that one set of the traffic lights works in correct sequence (Red - Red & Amber - Green - Amber).

Task 2

Add to the first program so that both sets of lights working in the correct sequence in opposition to each other and in a safe manner.

Task 3

Add the pedestrian crossing to the program, so that if the WAIT push button is pressed the WAIT LED comes on (for a period of 1 sec.), then the traffic light interrupted for 2 sec, so that the Green pedestrian comes on.

List of inputs and outputs

I0.6	Pedestrian Crossing Pushbutton.
Q0.0	Red North (*Green pedestrian, internally connected*)
Q0.1	Red East
Q0.2	Green North (*Red pedestrian, internally connected*)
Q0.3	Green East
Q0.4	Amber *East & North*
Q0.5	WAIT LED

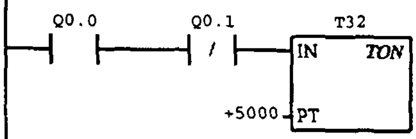
PLC APPLICATION

TRAFFIC LIGHT / PEDESTRIAN

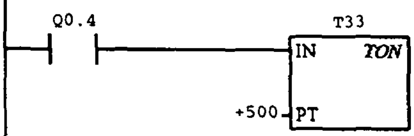
EXAMPLE PROGRAM FOR **TASK 1-2**

PLC APPLICATION
TRAFFIC LIGHT
TASK 1

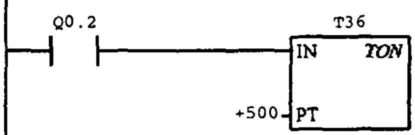
Network 1 RED TIME



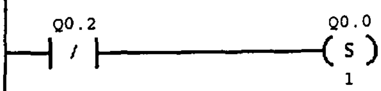
Network 2 AMBER TIME



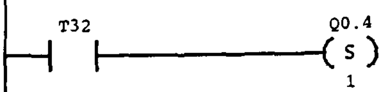
Network 3 GREEN TIME



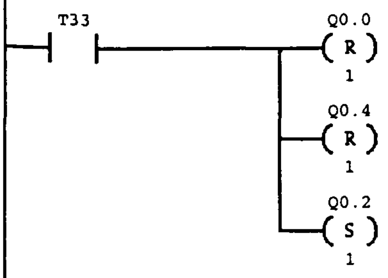
Network 4 SET RED NORTH



Network 5 SET AMBER EAST AND NORTH



Network 6 RSET RED NORTH, AMBER EAST & NORTH AND SET GREEN NORTH

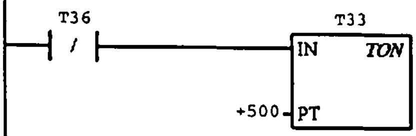


Network 7 RESET GREEN NORTH

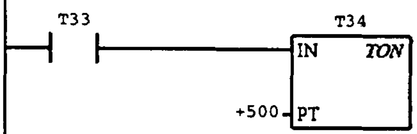
Network 8 END OF THE PROGRAM

PLC APPLICATION
TRAFFIC LIGHTS
TASK 2

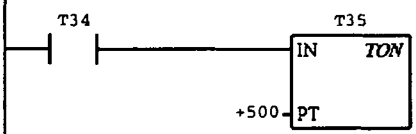
Network 1 AMBER SOUTH AND NORTH TIME



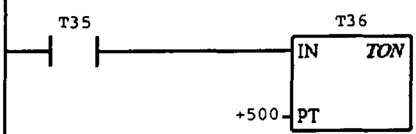
Network 2 GREEN SOUTH TIME



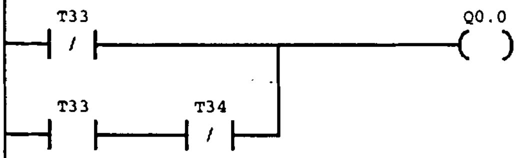
Network 3 RED NORTH 2nd STAGE TIME



Network 4

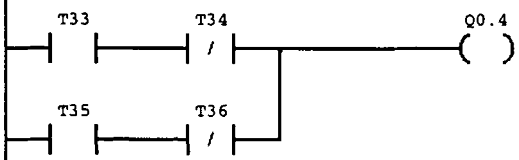


Network 5 RED SOUTH

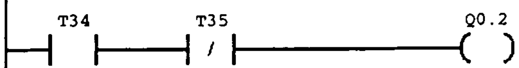


Network 6 GREEN NORTH

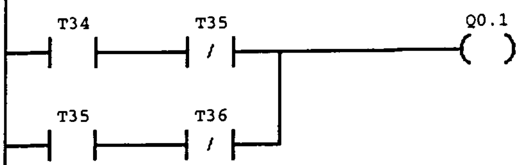
Network 7 AMBER SOUTH AND NORTH



Network 8 GREEN SOUTH



Network 9 RED NORTH

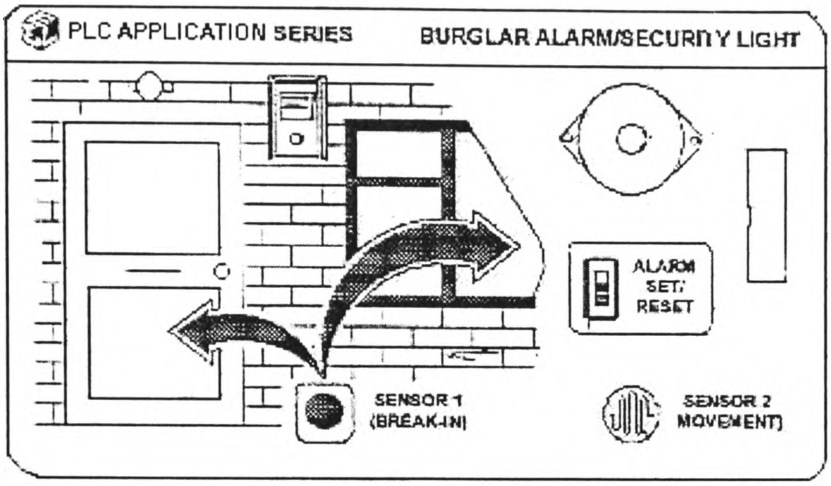


```
Network  10    END OF THE PROGRAM

|--(END)
```

PLC APPLICATION

BURGLAR ALARM / SECURITY LIGHT



Introduction

This board represents a burglar Alarm and Security Light.
Correct operation of this board should be as follows:
Moving the ALARM SET/ RESET switch into the SET position can activate the alarm. When this has been done this the alarm indicator (**Green LED**) will light to show that the alarm is active.
SENSOR 1 (BREAK-IN) being activated (this is done by pressing the push button) simulates a break-in through either the door or the window.
The **buzzer** will sound and the security light **(the lamp)** will flash on and off until the alarm is reset by moving the **ALARM SET/ RESET** switch into the **RESET** position.
SENSOR 1 (MOVEMENT) is activated when it senses any movement outside the building (this is done by covering the sensor with your finger). The security light (the lamp) will come on for a set period of time and then go out.

Programming commands

In programming the Burglar Alarm/Security light Application Board we will need some knowledge of the following commands:
Normally open contact, normally closed contact, coil, marker, timer.

Programming Tasks.

Before you start, ensure that the ALARM SET/RESET switch is in the set position (up).

Task One.

Write a program, so that pressing SENSOR 1 will cause the buzzer to sound until the ALARM SET/RESET switch is switched to the reset position (down). The buzzer will not re-activate when the switch is returned to the ALARM SET position.

Task Two.

Repeat task one, but this time use SENSOR 2 to make the alarm active and light the lamp. The alarm indicator (Green LED) should flash on and off.

Task Three.

Write a program so that if SENSOR 2 detects any movement, the lamp lights for 2 sec and then switches off.

Task Four.

Write a programme so that if movement is detected by SENSOR 2 the security lamp will light. Also, set the alarm so that if a break in occurs (SENSOR 1) the alarm will sound, the indicator will flash and the lamp (which should already be on) will begin to flash.

List of I/ O

Input 0.0	ALARM SET/RESET
Input 0.1	SENSOR 2 (MOVEMENT
Input 0.2	SENSOR 1 (BREAK-IN))
Output 0.0	Buzzer (Alarm)
Output 0.1	Alarm indicator (Green LED)
Output 0.2	Security light (lamp)

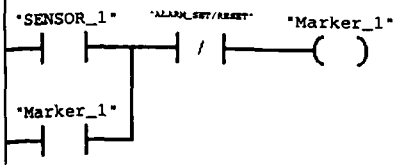
PLC APPLICATION

BURGLAR ALARM / SECURITY LIGHT

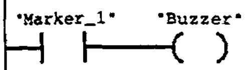
EXAMPLE PROGRAM FOR
TASK 1- 4

PLC APPLICATION
BURGLAR ALARM/SECURITY LIGHT
TASK 1

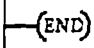
Network 1 Self Latching of Marker 1



Network 2 Marker 1 Switches the buzzer on

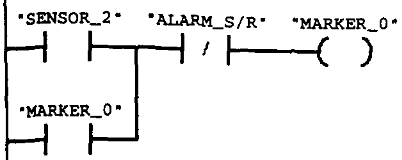


Network 3

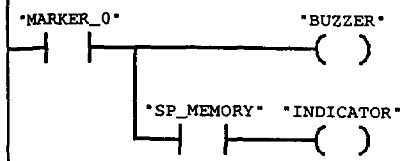


PLC APPLICATION
BURGLAR ALARM / SECURITY LIGHT
TASK2

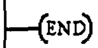
Network 1 USING SENSOR 2 TO ACTIVATE THE SYSTEM



Network 2 MARKER 0 ACTIVATE THE BUZZER AND FLASH THE INDICATOR

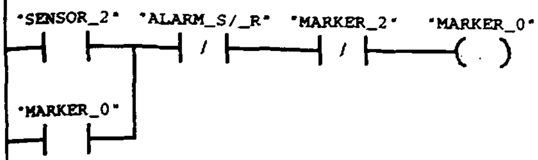


Network 3 END OF THE BROGRAMME

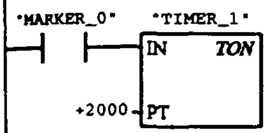


PLC APPLICATION
BURGLAR ALARM / SECURITY LIGHTS
TASK 3

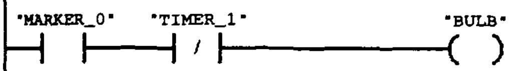
Network 1 SENSOR 2 ACTUATE THE SYSTEM



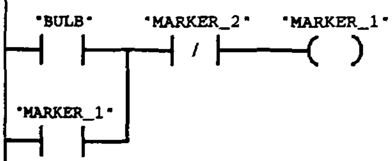
Network 2 Timer 32 Will Act After 2 Sec



Network 3 AFTER 2 Sec THE BULB WILL SWITCHED OFF



Network 4 Q0.2 ACTUATE THE INTERNAL MEMORY 0.1

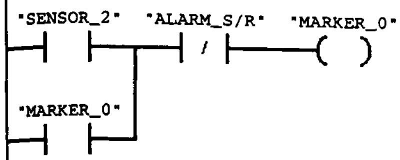


Network 5 AFTER 2 Sec INTERNAL MARKER 2 ACT AND RESTART THE SYSTEM

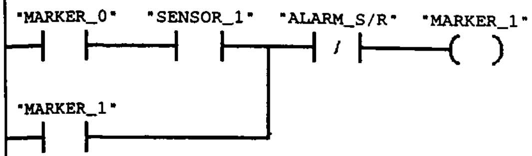
Network 6 END OF THE PROGRAM

PLC APPLICATION
BURGLAR ALARM / SECURITY LIGHTS
TASK 4

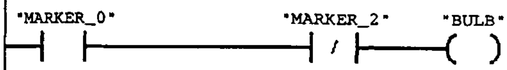
Network 1 SENSOR 2 SENSE THE MOVMENT AND SET THE INTERNAL MARKER 0.0



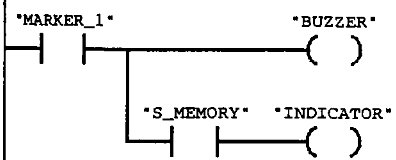
Network 2 SENSOR 1 SIMULATE THE BREAK AND SET THE INTERNAL MARKER 0.1



Network 3 THE BULB IS SWITCHED ON BY MARKER 0.0

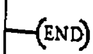


Network 4 WHEN A BREAK OCCURES THE BUZZER WILL SOUND, THE INDICATOR WILL FLACH



Network 5 INTERNAL MARKER 0.2 FLACHING, SO THE BULB WILL FLACH TOO

Network 6 END OF THE PROGRAMME



APPENDIX D

SYSTEM AVAILABILITY

INTRODUCTION

There are a number of important considerations for any manufacturer when installing a new production system. These relate to the Reliability, Risk Assessment and Failure Modes of the equipment. In the case of the proposed control system for Brohome Ltd., the actual production machines will not be replaced. In some cases, these machines are already being controlled by PLCs so that the Company should be familiar with the associated safety issues as outlined in the PES (Safety in Programmable Electronic Systems) publications of the Health and Safety Executive (HEALTH AND SAFETY EXECUTIVE 1987)as well as the likely failure modes and production availability. However, consideration must be given to these issues with regard to the introduction of new communications facilities to the plant.

COMPATIBILITY OF EQUIPMENT

As noted above, some of the production lines at Brohome are already being controlled by PLCs which are a number of years old, and which do not have the capability of being interfaced to an industrial fieldbus. The intention, as outlined in the case study in Chapter 6, is to replace these PLCs and where necessary install PLCs which will be compatible with the desired communication system. The suggested line controllers are the Siemens S7-200 which will communicated as Slaves to a Siemens S7-300 PLC as Master.

FAILURE MODES, RISK ASSESSMENT AND AVAILABILITY

In any networked system, the ways in which the communications system can fail and the effect that this will have on production are of interest. In common with other kinds of modern equipment based on microelectronic components, PLCs are very reliable devices. Faults in systems controlled by PLCs are usually associated with failures in the field wiring or the devices such as switches and actuators connected to the PLC. Where electronic devices fail, the so-called Bath-tub curve (M. Beasley 1991)(see figure D1) still represents the likelihood of failure at various stages of the devices age.

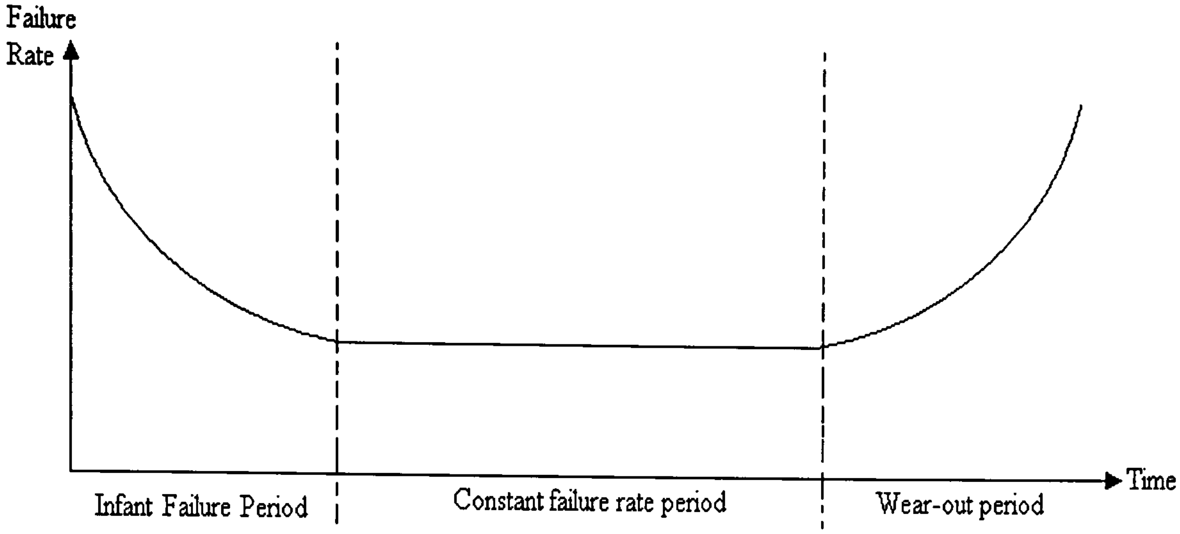


Figure D1 Bath-tub curve

For the purposes of this study, failures in the proposed Brohome installation (figure D2) can be represented as follows:

1. Failure of a Slave. This may occur through failure of the internal electronics or through the loss of its power supply. The result of such as failure will be that the line controlled by that particular slave will cease to function and the diagnostics associated with the Master should immediately pin-point the loss of that Slave from the network. Production on the other lines connected to the network will not be affected.

2. Failure of the Master. Loss of the Master will effectively cut communications between the factory floor and the Office. Production data will still be held in each Slave and this can be uploaded to the Office once communications are restored. However, production orders will not be able to be downloaded to the production area. Since the Slaves are still capable of controlling the lines, it should be possible to manually load production information into each Slave using a PLC programmer in order to maintain production. An alternative strategy would be to provide each line with an inexpensive operator panel to enable production data to be input to the PLC.

3. Failure of the Office System. In many ways this would cause similar problems to loss of the Master and production could be maintained as already explained by intervention through the use of a programmer or operator panel into the Master. The Master can hold production data until the problem has been resolved.

4. Network Cable failure. Breaking the network cable will not disable the Slave PLCs. Provided that the control program resides in each Slave as suggested, then production can continue as described above. The effect of this type of problem will be a loss of control from Office and Master and a delay in the passing of production data back to the Office. The exact nature of the problem will depend on the location where the failure occurs. The effects of breaks in the cable at locations A to D (figure D2) are self-evident.

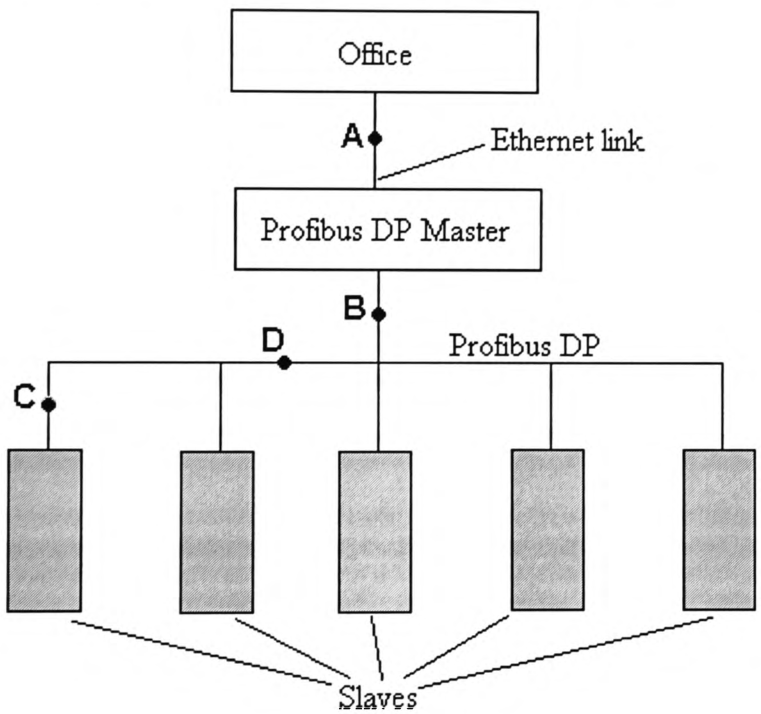


Figure D2 Brohome control system

REDUNDANCY

None of the control at Brohome is critical in nature, except with regard to the standard safety procedures outlined in the PES documents. It is therefore not thought necessary to provide Redundancy of equipment with all that this implies for extra cost and increased system complexity. The new equipment discussed for the Company revolves around the use of a fieldbus system. While it may appear that a single, two-core cable carrying all the information relating to the process makes the system vulnerable to damage, the above analysis indicates that in most circumstances production will be able to continue. Also, damage to a single, two-core cable can be repaired much more rapidly that damage inflicted on a large bundle of cables.

Reference List

1. HEALTH AND SAFETY EXECUTIVE. PROGRAMMABLE ELECTRONIC SYSTEMS IN SAFETY RELATED APPLICATIONS. UK: HEALTH AND SAFETY EXECUTIVE; 1987. ISBN: 0 11 883906 3.
2. M. Beasley. Reliability for Engineers. U.K: Macnillan Education Ltd.; 1991. ISBN: 0-333-54237-1.