Industrial IR Based Instrumentation Area Network

By Tshikalaha Takalani Raymond

Thesis presented in partial fulfillment of the requirements for the degree of **Masters of Science in Engineering Sciences** at the University of Stellenbosch.

Supervisor: Dr R. Wolhuter

Department of Electrical and Electronics Engineering
University of Stellenbosch

October 2004

Declaration

I, the undersigned hereby declare that the work contained in this thesis is my own original work and has not previously in its entirety or in part been submitted at any university for a degree.

Signature: Date:

Abstract

Wireless Area Network technology for industrial and factory applications is important for satisfying inflexible (safety-critical) real-time requirements in sometimes harsh environments. Many of these applications involve mobile subsystems and could benefit from recent Wireless LAN technologies replacing the current cable-based systems. An immediate question is how this technology can be used for wireless Area Network systems? An important aspect of this question is the development of time-variable wireless links with good real-time performance. This project will attempt to answer some aspects of this question. The main objective of this thesis is to create a wireless area network for instrumentation purposes, interconnecting various monitoring and control transducers to a central master station.

This project focuses on three transmission technologies used for wireless LANs with low power consumption; capable of close range positioning, indoors as well as outdoors. These transmission technologies are Infrared LAN (IrDA), Spread Spectrum LAN and Narrowband Microwave LAN. As a result of the evaluation of the three technologies, an Infrared LAN (IrDA) system was implemented as an area network, utilising an IrLAP protocol (Master and Slave) as a communication protocol. The Master is enabled to monitor and control all slaves interfaced to it.

Opsomming

Draadlose netwerktegnologie vir industrietoepassings, is nodig om aan te pas by spesifieke veiligheids- en omgewingstoestande. Baie van hierdie toepassings het betrekking op mobiele substelsels en kan baat by vervanging van bekabeling met onlangse draadlose netwerktegnologie. Die ontwikkeling van sulke netwerke met goeie tydreaksie, is hier belangrik. Die hoofdoel van hierdie tesis is om 'n draadlose areanetwerk te skep vir instrumentasiedoeleindes, wat verskeie monitor-en beheeromsetters aan 'n sentrale meesterstasie sal verbind.

Hierdie projek fokus op 3 sulke benaderings, nl. Infrarooi AN (IrDA), Spreispektrum AN en Nouband Mikrogolf AN. Na ondersoek is 'n stelsel gebaseer op IrDA, geimplementeer as areanetwerk, met behulp van die IrLAP protokol. Die meester beheer alle kommunikasie met- en beheeraksies van die buitestasies.

Acknowledgement

Firstly, I would like to thank the Almighty God for the strength and ability to complete the project, my study leader for his support during the project. Dr R. Wolhuter, without your advice I would have gotten lost on the way to completion.

I would also like to thank my mother Vho-Phophi Elisa Tshikalaha, my sisters (Livhuwani, Litshani, Tshifhambano and Mpho), for their undivided support during all those trying times and for believing in me. You gave me the reason to finish the things that I would never have finished without your confidence in me.

I would also like to thank my friend (Portia Nyadzani Masevhe) for her encouragement she gave me, and everyone who contributed to the progress of the project.

I further extend my appreciation and thanks giving to the South African National Department of Communication, through their initiative the Institute for Satellite and Software Application (ISSA), for the financial support.

Contents

List of Figures

List of Tables

List of abbrevations and acronyms

ACK	Acknowledge
ADDR	Address
BOF	Begining Of Frame
BPSK	Binary Phase Shift Keying
Cnt	Control
CPU	Central Processor Unit
CRC	Cyclic Redundancy Check
DDC	Direct Digital Control
DSSS	Direct Sequencing Spread Spectrum
EOF	End Of Frame
FCS	Frames Check Sequence
FHSS	Frequency Hopping Spread Spectrum
FIR	Fast Infrared
FSK	Frequency Shift Keying
HDLC	High-Level Data Link Control
IAP	Information Access Protocol
IAS	Information Access Services
ICASA	Independent Communications Authority of South Africa
IEEE	Institute of Electrical and Electronic Engineers
ISM	Industrial, Scientific, and Medical Band
I/O	Input/Output
IR	Infrared
IrCOMM	Infrared Communications Protocol
IrDA	Infrared Data Association
IrLAN	Infrared Local Area Network
IrLAP	Infrared Link Access Protocol
IrLMP	Infrared Link Management Protocol
IrOBEX	Infrared Object Exchange Protocol
IrPHY	Infrared Physical Layer
Kbps	Kilobits per second
LAN	Local Area Network
LM-IAS	Link Management Information Access Service
LM-MUX	Link Management Multiplexer
LMP	Link Management Protocol
LSAP-SEL	Link Service Access Point Selector
MAP	Manufacturing Automation Protocol
Mbps	Megabits per second
MIR	Medium Infrared
NC	Numerically Controlled
NDM	Normal Disconnect Mode
NRM	Normal Response Mode
OFDM	Orthogonal Frequecy Division Multiplexing
OSI	Open Systems Interconnection

PDA	Personal Digital Assistant
PLC	Programmable Logic Controller
PN	Pseudonoise
PPM	Pulse Position Modulation
QPSK	Quadrature Phase Shift Keying
RF	Radio Frequency
RLL	Run Length Limited
RZI	Return to Zero Inverted
Rx	Receive
SAR	Segmentation And Reassembly
SDLC	Synchronous Data Link Control
SIR	Serial Infrared
SS	Spread Spectrum
Tiny TP	Tiny Transport Protocol
Tx	Transmit
UART	Universal Asynchronous Receiver Transceiver
USB	Universal Serial Bus
VFIR	Very Fast Infrared
WAN	Wide Area Network
WLAN	Wireless Local Area Network
WMAN	Wireless Metropolitan Area Network
WPAN	Wireless Personal Area Network
WWAN	Wireless Wide Area Network
XID	Exchange Identification

Chapter 1

Introduction and Purpose

1.1 Introduction

Early, from the 1950's, many industrial communication systems were developed for control applications. These were proprietary networks using analog technology, and used to link the central processor to peripherals and terminals. Peripherals typically used parallel, multi-wire cables, and serial interfaces such as the RS232c 20mA current loop at low transmission rates. At the beginning of the 1960's, a digital computer was for the first time really applied as a digital controller. The term Direct Digital Control was used to emphasize that the computer directly controls the process.

In the 1960s, the application of a minicomputer was still a fairly expensive solution for many control problems. In the meantime, PLC's were developed and it replaced the conventional, relay-based controller, with relatively limited control functions. In addition, many technologies were developed for machine tools and discrete production processes. The Numerically Controlled machine tool was controlled by computers and the robot was developed in this period. The high capacity, low cost communication means, offered by LAN's have made distributed computing a reality, as well as many automation schemes.

Industrial automation systems are often implemented as an open distributed architecture with communication over digital communication networks. It is now common for users connected to a LAN to communicate with computers or automation devices on other local area networks via gateways linked by a WAN. As the industrial automation systems becomes large and the number of automation devices increases, it has become very important for industrial automation to provide standards which make it possible to interconnect many different automation devices in a standard way. Considerable international standardization efforts have been made in the area of LANs.

The OSI standards permit any pair of automation devices to communicate reliably regardless of the manufacturer. At the lower level communication networks for industrial automation, the industrial local area network solutions such as Manufacturing Automation Protocol (MAP) are too expensive and/or do not achieve the required short response times, depending on the application. Traditional wireless information networks, which include cordless and cellular telephones, paging systems, mobile data networks, mobile satellite systems, and IrDA devices have experienced enormous growth over last decade. New concepts of personal communication systems, WLANs, and mobile computing have appeared in industry. After more than a century of reliance on analog-based technology for telecommunications, nowadays people live in a mixed analog and digital world and are rapidly moving toward all digital networks.

The wireless communications industry is one of many that will continue to benefit from the introduction of digital technology. As the demand for communication services continues to increase, manufacturers and services providers are looking towards digital implementations for increased capacity and a wider offering of services to their users.

1.2 Purpose of the Project

Since the world is rapidly moving towards digital network technologies, this project concentrates on the development of short-range real time Wireless Local Area Network for instrumentation purposes, interconnecting various monitoring and control transducers to a central master station. The use and application of IrDA protocol as a communication protocol that can virtually eliminate all instrumentation plant wiring between the Primary (or Master Station) and Secondary (or Slave Station) in appropriate locations is investigated and developed.

The proposed wireless area network will facilitate point-to-point communication between electronic devices (e.g. Controlling Computer and Instrumentation Peripherals) using half-duplex serial infrared communication links through free space. This will increase mobility, flexible installations, enable easy scalability, reduce costs, facilitate and boost efficiency in the industrial applications.

A further objective of the project, was to enable the practical use of this protocol. This was achieved by developing a feasible user interface and related infrastructure.

A means to actually configure and run hardware and software based field installation, was provided and practically demonstrated.

1.3 Applications of Project

The project could be used in industry for monitoring and controlling. For example, in a winery where there are many tanks for storing wine, one will just sit down in the office and be able to monitor and control all those tanks using a wireless LAN link, instead of going to each tank to monitor and control temperature, pressure, and volume every time. In each and every slave station there could be an I/O board with digital and analog I/O. Each and every tank will be connected to the slave station I/O via cable. Each of the slave stations is connected to the master station via wireless link (Infrared) as shown in Figure 1.1.

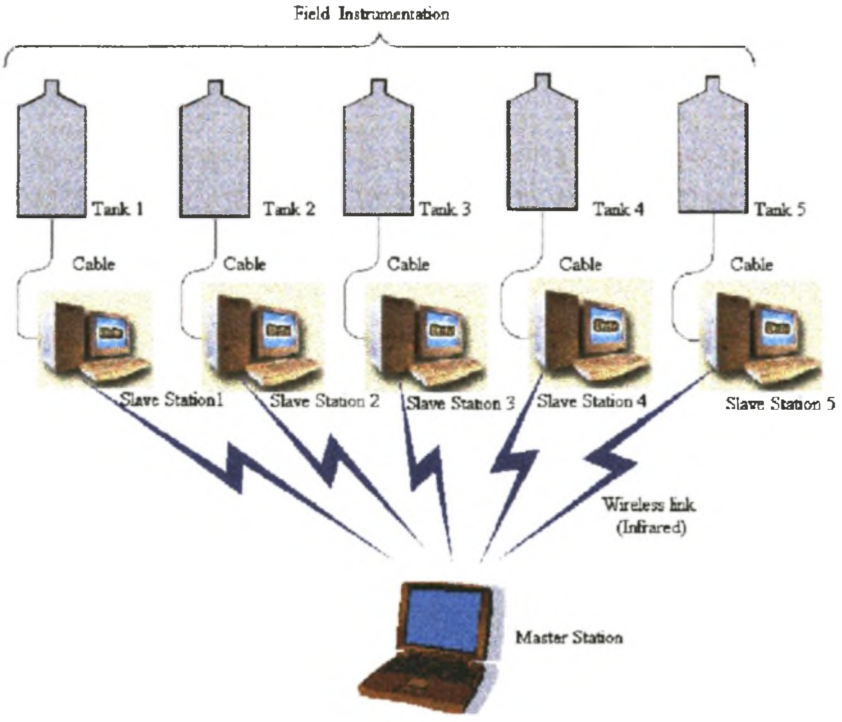


Figure 1.1: Wireless Communication Links

This is just one of many typical layouts, where this development and technology, could be applied.

1.4 Thesis Overview

The thesis is structured as follows:

- Chapter 1 - **Introduction and Purpose**
  This chapter introduces the industrial communication and automation systems. It also gives a description of why this work was done, including the purpose and application of the project.

- Chapter 2 - **Background Theory**
  Wireless data communication, which includes Wireless LANs will be discussed in this chapter. The purpose of this chapter is to give a general overview of wireless transmission technologies and transmission techniques.

- Chapter 3 - **Comparative WLAN Technologies**
  The different aspects that are of importance for wireless transmission technologies are discussed. A comparison is made between Infrared LANs, Spread Spectrum LANs and Narrowband Microwave LANs and the most appropriate transmission technology is selected.

- Chapter 4 - **Implementation of IrDA Protocol**
  The design and the implementation of IrLAP protocol (master-slave protocol) and implementation of Delphi 7 to access an I/O Board will be discussed in this chapter.

- Chapter 5 - **Evaluations and Results**
  The general reliability of system, performance, testing and evaluation of general functionality of system and the results found were discussed in this chapter.

- Chapter 6 - **Application Layer Implementation**
  This chapter discusses the Graphic User Interface (GUI) and Database Interface.

- Chapter 7 - **Conclusions and Recommendations**
  This chapter concludes the project by summarising the results and areas for future research, that could be done to enhance the performance and /or functionality, is recommended.

Chapter 2

Background Theory

2.1 Wireless Data Communication

Wireless data communication, as the term implies, allows information to be exchanged between two devices without the use of wire or cable.

Wireless data communications can take on many forms using a variety of technologies. Communications can occur using satellite, microwave, spread spectrum, ham radio (or an amateur radio station), cellular, infrared, and laser technologies. Each of these techniques is in use for voice as well as data communications, and there are numerous vendor offerings to accompany each communication method.

For example, satellite technology is used to beam television, telephone and data signals around the world to cover thousands of miles between sites. Microwave is used for relaying telephone, television and data signals between communities. Laser technologies can transmit signals between devices up to a 2km apart, such as between two corporate offices in the same city. Infrared technology can enable two devices to communicate with each other across a room, such as between a computer and a printer, or between a remote control and a television set. Spread spectrum technology is used for wireless connection between peripheral devices, and for data communications over a few kilometers. Bluetooth is the newest technology to enter the arena and offers new levels of local connectivity[19].

Wireless technologies can be applied to data communications for Wide Area Networking (WAN) as well as for Local Area Networking (LAN). Each has its own advantages and disadvantages, including the price/performance aspects of the system.

Standards for wireless networking

IEEE 802.11 is the original wireless LAN standard that specifies the slowest data transfer rates in both spread spectrum RF and infrared transmission technologies.

Currently there are four major wireless-networking standards.

1. **802.11b** - is the corporate standard and has a suitably wide range for use in big office spaces. It operates in the unlicensed 2.4GHz - 2.5GHz Industrial, Scientific, and Medical (ISM) frequency band using a direct sequence spread-spectrum technology. It permits transmission speeds of up to 11 Mbps.
2. **802.11a** - offers bigger bandwidth and fewer interference problems but a shorter range. It operates in the licensed 5 GHz band using Orthogonal Frequecy Division Multiplexing (OFDM) technology. Currently some manufacturers are modifying their equipment to handle 22 Mbps or more using this standard.
3. **802.11g** - is a new upcoming standard, an extension of the 802.11b standard, which means that old 802.11b equipment will work with the new 802.11g equipment. Current connection speeds of up to 54 Mbps, are available
4. **Bluetooth** - is meant for short-range, temporary (ad-hoc) networking in conference rooms, schools, or homes.

2.1.1 Introduction to Wireless LAN technologies

What is a Wireless Local Area Network?

A Wireless Local Area Network (WLAN) is a flexible data communication system implemented as an extension to, or as an alternative for, a wired LAN within a building or campus, using electromagnetic waves (typically infrared or radio), to enable communication between the devices in a limited area.

WLANs transmit and receive data over the air, minimizing the need for wired connections. Thus, WLANs combine data connectivity with user mobility, and, through simplified configuration, enable movable LANs.

Unlike Bluetooth, WLANs provide continuous coverage for devices in the network. As the devices may roam freely within the coverage areas, these coverage areas remain fixed.

Three transmission technologies used for WLANs are

1. Infrared LANs,
2. Spread Spectrum LANs, and
3. Narrowband Microwave LANs.

Each of these transmission technologies is in use for voice as well as data communications, and there are numerous vendor offerings to accompany each communication method[17].

2.1.2 Advantages of WLAN's

1. Enable communications in areas where wired networks are difficult to install (e.g. historic building, firewalls).
2. Reduce network installation costs.
3. Provide access anywhere (mobile computing).
4. Very flexible within the reception area
5. Enhance data access.
6. Ad-hoc networks without previous planning possible

2.1.3 Disadvantages of WLAN's

1. Proprietary solutions: slow standardization procedures lead to many proprietary solutions only working in a homogeneous environment (e.g. IEEE 802.11).
2. Safety and security: using radio waves for data transmission might interfere with other high-tech equipment.
3. Lower bandwidth due to limitations in radio transmission (1-10 Mbps) and higher error rates due to interference.

2.1.4 Benefits of WLAN's

1. Mobility improves productivity and service - Wireless LAN systems can provide LAN users with access to real-time information anywhere in their organization. This mobility supports productivity and service opportunities not possible with wired networks.
2. Installation Speed and Simplicity - Installing a wireless LAN system can be fast and easy and can eliminate the need to pull cable through walls and ceilings.
3. Installation Flexibility - Wireless technology allows the network to go where wire cannot go.
4. Reduced Cost-of-Ownership - While the initial investment required for wireless LAN hardware can be higher than the cost of wired LAN hardware, overall installation expenses and life-cycle costs can be significantly lower. Long-term cost benefits are greatest in dynamic environments requiring frequent moves, adds, and changes.
5. Scalability - Wireless LAN systems can be configured in a variety of topologies to meet the needs of specific applications and installations.

2.2 Infrared LANs Technology

Infrared systems use infrared emission to carry information and are used by IEEE 802.11R standard.

Two alternatives of infrared LANs transmission techniques:

1. Diffuse-beam and
2. Direct-beam.

Diffuse-beam Infrared LANs

Diffuse beam infrared LANs do not require line-of-sight directly between two devices. The receivers could be located anywhere in the cells where the transmitted beams could be reached.

Diffuse-beam is divided into two techniques

1. Omnidirectional - a single base station mounted on a ceiling within a line of sight of all other stations on the LAN. Ceiling transmitter broadcasts omnidirectional signals, which can be received by other IR transceivers. Figure 2.1 shows an omnidirectional diffused beam setup.

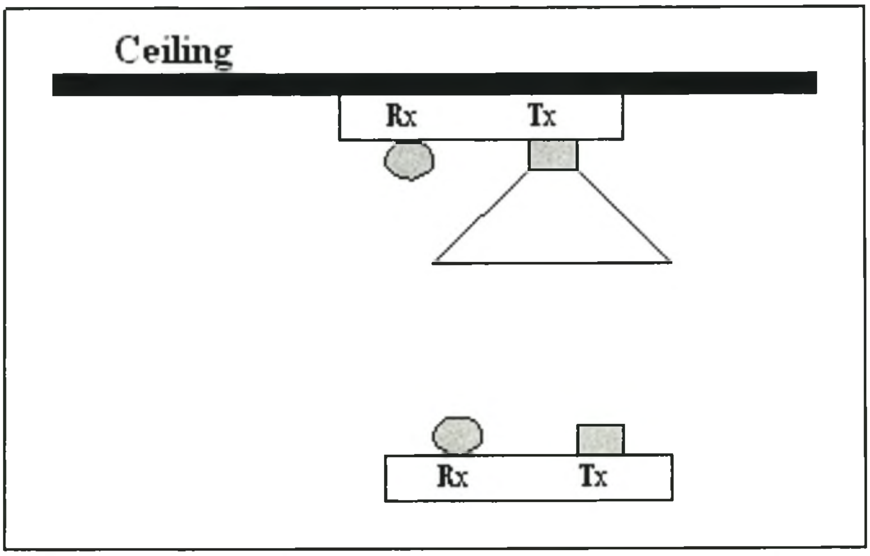


Figure 2.1: Omnidirectional Diffused Beam (Line of Sight).

2. Diffused - all IR transceivers are focused on a diffusely reflecting ceiling. Infrared radiation striking the ceiling is reradiated omnidirectionally and picked up by all receivers. Figure 2.2 shows the diffuse reflections.

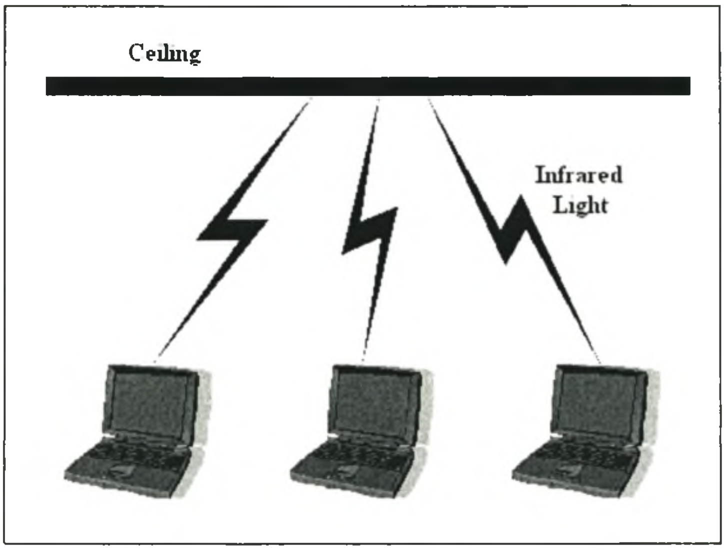


Figure 2.2: Diffuse Reflections of Infrared Light.

Directed Beam (point-to-point) Infrared LANs

With direct beam infrared systems, line-of-sight is required. The receiver is aligned with the sender unit. The infrared light is then transmitted directly to the receiver. The range depends on the emitted power and on the degree of focusing. A focused IR data link can have a range of numerous meters, and such ranges are not needed for constructing indoor WLANs. One indoor use of point-to-point IR links is to setup a token ring LAN[1].

IR transceiver can be positioned so that data circulate around them in a ring configuration. Each transceiver supports a workstation or hub station, with the hub providing bridging function. Figure 2.3 shows a directed beam Infrared (Token Ring).

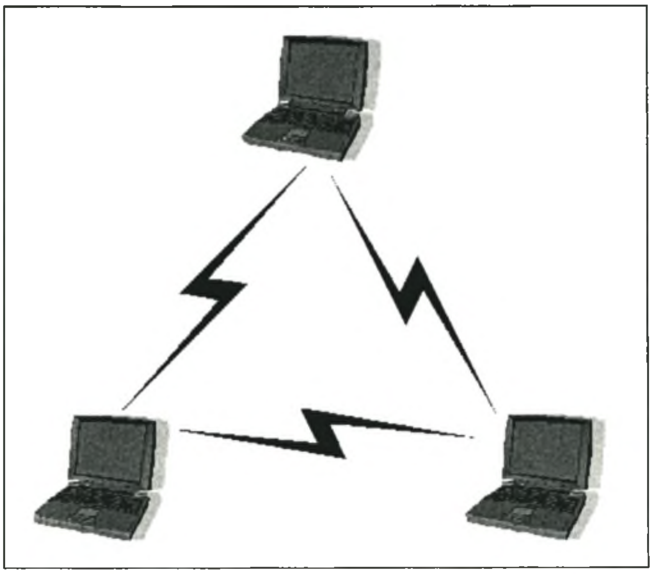


Figure 2.3: Directed Beam Infrared (Token Ring)

2.2.1 Introduction to IrDA

Infrared Data Association (IrDA) is a communication system based on infrared emission. It specifies a way to wirelessly transfer data via infrared radiation. IrDA devices communicate using infrared LED's. The wavelength used is ±875nm and production tolerance is around 30nm. It is commonly used in mobile devices for low cost, and point-to-point communication. Digital cameras, mobile phones and laptops are just a few examples of devices that often use IrDA for wireless communication[20].

2.2.1 (a) IrDA Protocols

IrDA Protocols consist of a mandatory set of protocols and a set of optional protocols. Figure 2.4 below shows how the IrDA protocol stack is layered[6].

The most important protocols are of course the mandatory protocols: IrPHY (Infrared Physical Layer), IrLAP (Infrared Link Access Protocol) and IrLMP (Infrared Link Management Protocol).

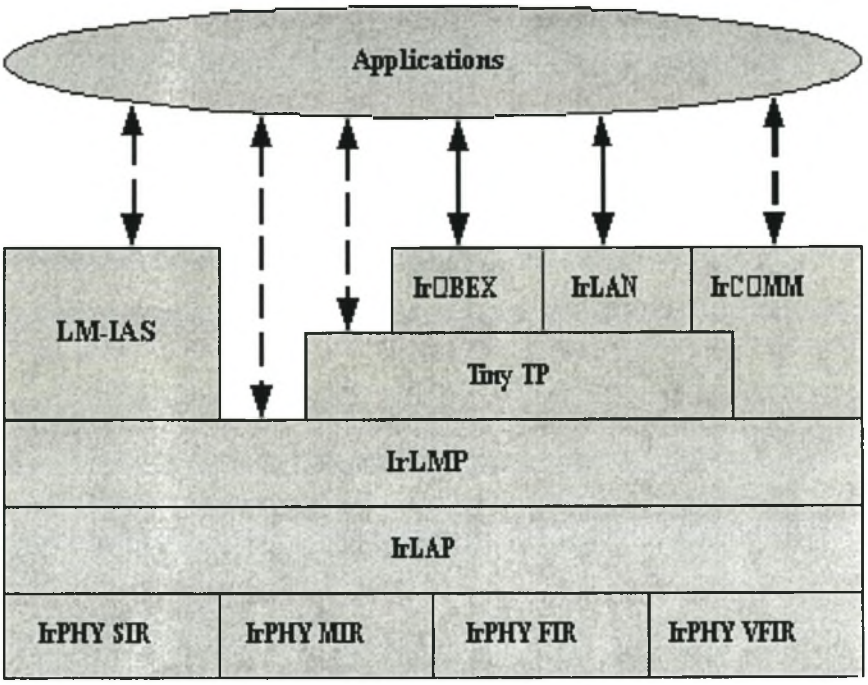


Figure 2.4: The IrDA Protocol Architecture

The optional protocols are Tiny TP (Tiny Transport Protocol), IrOBEX (Infrared Object Exchange Protocol), IrLAN (Infrared Local Area Network), and IrCOMM (Infrared Communications Protocol). The use of optional protocols depends upon the particular application. A brief description of these protocols can be found in Appendix A.

2.2.1 (b) IrDA Physical Layer (IrPHY)

The physical layer contains the actual Infrared transducer module. It is responsible for transmitting and receiving Infrared signals and also encode/decode these signals for the IrLAP layer, and some framing data, such as begin and end of frame flag (BOFs and EOFs) and Cyclic Redundancy Check (CRCs). Its primary responsibility is to accept incoming frames from the hardware and present them to the Link Access Protocol layer (IrLAP). This includes accepting the outgoing frames and doing whatever is necessary to send them.

Figure 2.5 below shows the infrared transducer module, the electrical signals to the left of the Encoder/Decoder at "**A**" are serial bit streams, for data rates up to 1.152 Mbps. The optical signals at "**C**" are bit streams with a "0" being a pulse, and a "1" is a bit period with no pulse. For 4 Mbps, a 4PPM (Pulse Position Modulation) encoding scheme is used with a "1" being a pulse and a "0" being a chip with no pulse.

The electrical signals at "**B**" are the electrical analogs of the optical signals at "**C**", for data rate up to 115.2 Kbps. In addition to encoding, the signals at "**B**" are organized into frames, each byte asynchronous, with a start bit, 8 data bits, and a stop bit. For data rates above 115.2 kbps, the data is sent in synchronous frames consisting of many data bytes.

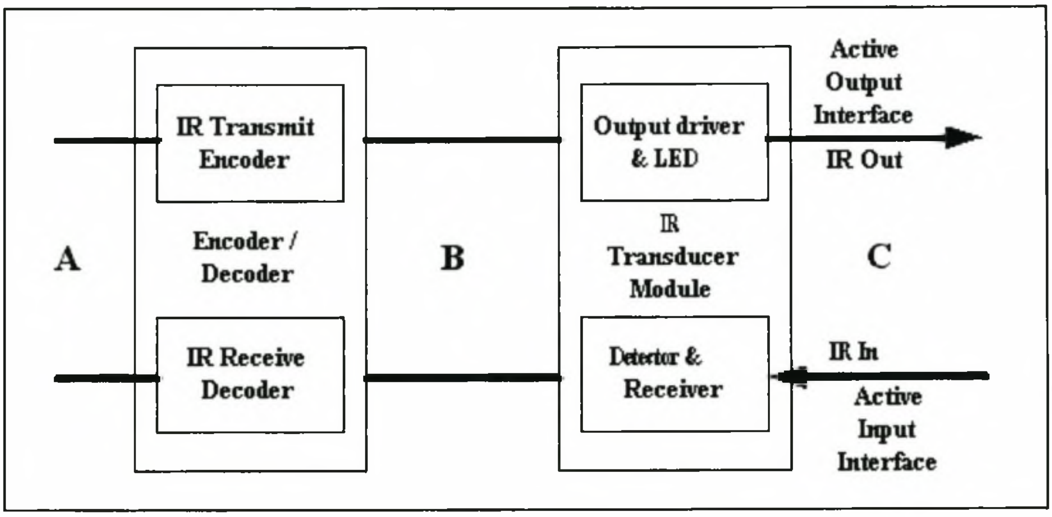


Figure 2.5: Infrared Transducer Module

2.2.1 (c) Physical aspects of the Infrared Physical Layer

When Infrared is transmitted there are several limitations in range and angle that other systems, (e.g. radio links), do not have. These limitations consist of limited range, line of sight and limited viewing angles.

Range of an IrDA device

Indoors, range varies from few meters (e.g. palmtop computer-to-portable printer) up to sixty meters (point-to-point link between fixed nodes). Outdoors, rooftop-to-rooftop infrared links may reach distances greater than 1 km, but are subjected to occasional weather related outages [4].

Optical angle limitations of an IrDA device

The infrared transmission is directional within a 15° half angle in order to minimize interference with surrounding devices. The optical signals are limited by angles in the transmitter and receiver. The transmitter has a typical limitation of 15° to 30° from the optical axis, also called half angle. The receiver is limited to 15° half angle or just above.

Power consumption

IrDA SIR is designed to be power efficient so that it will not be a drain on the batteries of portable devices like notebook computers, PDAs, mobile phones and other handheld IrDA devices. As IrDA devices are intended for short range, point-to-point communications, the technology will display an advantage over diffuse IR technologies (wide area coverage devices) since it uses very low power when transmitting. IrDA has low power consumption.

2.2.1 (d) The capacity and formats of the Infrared Physical Layer

The IrDA physical layer is split into four distinct data rate ranges: 2400bps to 115,200bps, 1.152 Mbps, 4 Mbps and 16 Mbps. A first protocol negotiation takes place at 9600 bps, making this data rate compulsory. All other data rates are optional and can be added if a device requires a higher data rate.

Infrared receivers contain a low-pass filter to remove background daylight. This low-pass filter forces the use of encoding on the link to ensure that long strings of zeros or ones are not lost in transmission. The actual format of packets that pass through on the infrared media can vary, according to the speed at which those packets are transmitted and received. The coding of packets and BOF, EOF, and FCS varies depending on the operating speed of the infrared media.

Serial Infrared (SIR) Link

The SIR defines a short-range infrared asynchronous serial transmission mode with one start bit, eight data bits, and one stop bit. The maximum data rate is 115.2Kbps (half duplex). This SIR coding scheme is called Return-to-Zero-Inverted (RZI). The BOF flag for SIR speeds is defined as 0xC0. The EOF value is defined as 0xC1.

Medium Infrared (MIR) Link

MIR link support 115,200 bps up to 1.152 Mbps data rate. At speeds above 115,200 bps, packet framing, CRC generation and checking become a significant burden to the host processor. At 1.152 Mbps, these tasks are performed in hardware by a packet framer. Higher-level protocols are less processor intensive than packet framing or CRC generation and are still implemented in software on the host processor. MIR uses a 1/4 bit period RZI modulation and Synchronous framing. For MIR link speeds, BOF and EOF values are the same; both BOF and EOF are defined as 0x7E. Two BOF flags are required on every frame.

Fast Infrared (FIR) Link

FIR Links supports a 4.0 Mbps data rate. As in the MIR link, packet framing, CRC generation and checking are performed in hardware to ease the load on the host processor, while higher-level protocols are implemented in software on the host processor. Pulse Position Modulation (PPM) framing is used and defines special flags for BOF and EOF. The FIR link uses a new encoding scheme and a new, more robust packet structure. A phase-locked loop replaces edge detection as the means of recovering the sampling clock from the received signal.

Very Fast Infrared (VFIR) Link

VFIR Link supports a 16.0 Mbps data rate. It uses the dedicated H HHH (1,13) encoding, and a rate 2/3 (1—13,5) RLL (Run Length Limited) scheme. The letters HHH that represent this coding scheme are the initials of the three researchers who invented it. The HHH (1,13) coding scheme should always be implemented in hardware.

2.2.1 (e) Interference on the Infrared Physical Layer

Environment light and electromagnetic fields are two factors that may interfere with the Infrared Physical Layer. There are basically four ambient interference conditions, which the receiver is to handle correctly. The conditions are to be applied separately.

1. Electromagnetic field: 3 V/m maximum
2. Sunlight: 10 kilolux maximum at the optical port
3. Incandescent Lighting: 1000 lux maximum
4. Fluorescent Lighting: 1000 lux maximum

There is also the aspect of distance between transmitter and receiver, which has been discussed earlier.

The interference can be seen with the Bit Error Rate (BER), which is the number of errors received, or expected divided by the total number of transmitted bits. The BER should be not greater than 10^{-8}, because too high a BER may indicate that a slower data rate would actually improve overall transmission time for a given amount of transmitted data since the BER might be reduced, lowering the number of packets that must resent.

2.2.2 Infrared Link Access Protocol (IrLAP)

The IrLAP protocol specification corresponds to the OSI layer 2 (Data Link Protocol), and is a mandatory layer for IrDA protocols. IrLAP is based on the pre-existing HDLC and Synchronous Data Link Control (SDLC) half duplex protocols, with some modifications to provide to the unique features and requirements of infrared communications.

The purpose of the IrLAP layer is to establish connection between IrDA devices. In doing so the IrLAP layer must deal with discovering hidden nodes, address conflicts and handling requests and confirmations to upper layers. The IrLAP layer is located right on top of the physical layer and the framer in the protocol stack.

There are basically two states of the IrLAP: Primary (master) and Secondary (slave). The master is the one telling all connected devices which one is allowed to send at the moment. Only one device is allowed to send at a time, and thus the master play an important role in making sure this is obeyed by all secondary devices [11].

2.2.2 (a) Discovering of other IrDA devices

There are three discovering services: request, indication and confirm. The "Request" is used to find out what, if any, devices are within communication range and if they are available for connection. "Confirm," returns a list with all available devices. Finally, the "Indication" is used to send information about the device that sends a request, to other devices.

2.2.2 (b) Connection of IrDA devices

A device that wants to broadcast its desire to connect may do so by using a procedure called sniffing, which is a power conservative procedure. A device that wants to connect and approaches a network of Infrared devices is called a hidden node. This device needs to listen and wait until spoken to, before it can connect to the network. This procedure is also a part of the sniffing procedure.

The basic procedure of the Sniffing device

1. A sniffing device wakes up and listens for a short period of time. If it hears traffic it goes back to sleep.

2. If it does not hear traffic it transmits an Exchange Identification (XID) response frame with a special value unique to the sniffing procedure. This XID indicates that the device desires to be connected as a slave.

3. The device then waits a short period for a message directed to it. If such a message arrives the device can connect.

4. If no frames are sent to it, the Sniffing device goes to sleep (usually 2 - 3 seconds) and starts the procedure again. If it hears traffic not directed to it, it is assumed to be connection traffic and the device cannot connect.

Modes for connection

IrLAP is built around two modes of operation, corresponding to whether or not a connection exists.

1. Normal Disconnect Mode (NDM) - NDM is also known as the contention state, and is the default state of disconnected devices. In order to connect from this state the device must first listen for a time greater than 500 milliseconds. If no traffic is detected during this time then the media is considered to be available for establishment of a connection.
2. Normal Response Mode (NRM) - NRM is the mode of operation for connected devices. Once both sides are talking using the best possible communication parameters (established during NDM), higher stack layers use normal command and response frames to exchange information.

2.2.2 (c) Address conflicts

The address conflict services are used to resolve device address conflicts. If the discovery log contains entries for more than one device with the same device address, the address conflicts service may be invoked in order to cause the IrLAP layers of the conflicting devices to select new non-conflicting device addresses. The IrLAP addresses are 32-bit randomly selected addresses. On an address collision a new random address is selected.

2.2.2 (d) IrLAP Services

Once the connection has been established, the IrLAP starts to work as a kind of message handling service for the upper layers. As help, the IrLAP have four generic types of service primitive:

1. Request: Passed from the Upper Layer to invoke a service.
2. Indication: Passed from IrLAP to the Upper Layer to indicate an event or notify the Upper Layer of an IrLAP initiated action.
3. Response: Passed from the Upper Layer to acknowledge some procedure invoked by an indication primitive.

4. Passed from IrLAP to the Upper Layer to convey the result of the previous service request.

Figure 2.6 shows the graphical representation of how these primitives are related to each other.

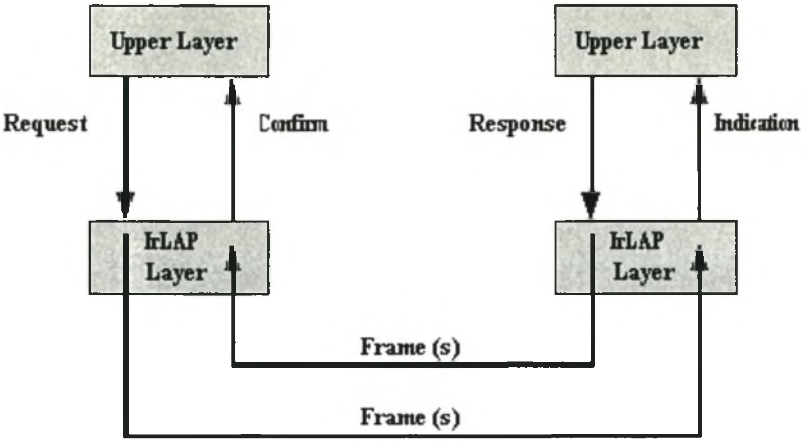


Figure 2.6: IrLAP Services

2.2.3 Infrared Link Management Protocol (IrLMP)

The IrLMP protocol is a layer that sits above the IrLAP layer. It provides services to both the Transport layer and directly to the application layer. IrLMP consists of two components, LM-IAS (Information Access Service) and LM-MUX (Link Management Multiplexer).

2.2.3 (a) Link Management Multiplexer (LM-MUX)

The IrLMP multiplexer, LM-MUX, makes it possible for several clients to connect to the IrLAP connection thus relieving the client entity of the requirement of coordinating access to the single IrLAP connection. In order to do this, LM-MUX uses several of the IrLAP services such as discovery, link control and data transfer. When several clients are connected to the IrLAP protocol by using the LM-MUX, the LM-MUX is called being in Multiplexed Mode.

Some protocols and applications may require special control of a service access point in order to achieve a reduced, dependable latency and/or control the link turnaround through their use of the link. This special case is called Exclusive Mode.

2.2.3 (b) Information Access Service (LM-IAS)

The Information Access Service (IAS) acts as the "yellow pages" for a device. A full IAS implementation consists of client and server components. The client is the component that makes inquiries about services on the other device using the Information Access Protocol (IAP, used only within the IAS). The server is the component that knows how to respond to inquiries from an IAS client. The server uses an information base of objects supplied by the local services/applications.

The LM-IAS Information Base

The IAS Information Base is a collection of objects that describes the services available for incoming connections. It consists of a class name and one or more attributes. They are quite similar to entries in the yellow pages of a phone book.

The class name is equivalent to the business name in the phone book; it is the official published name of the service or application. IAS clients will inquire about a service using this name. The attributes contain information, which can be compared to the phone number, address or other characteristics of a business found in the yellow pages.

One important attribute is the Link Service Access Point Selector Address (LSAP-SEL address or service address), which is required in order to make a Link Management protocol (LMP) connection to the service.

Getting information using the LM-IAS

There are a number of IAS operations defined in the IrLMP standard, but the most used and only required one is the one used to get values by providing class (GetValueByClass) from the IAS service. The procedure might be as follow[13]:

IAS Query arguments:

1. Class Name Length

2. Class Name
3. Attribute Name Length
4. Attribute Name

Results:

1. Return code:
   - 0: Success, results follow.
   - 1: No such class, no results follow.
   - 2: No such attribute, no results follow.

If the result code indicates success, the call returns the following information:

1. List Length
2. List of results:
   - Object Identifier
   - Attribute value

2.2.4 Security of IrDA

IrDA contains no encryption or other means of security. Still, IrDA is considered secure because of the limited range and the fact that it requires line of sight. Someone wanting to overhear communication needs to be in the direct vicinity of the communicating devices and on top of that be within the angle limitations.

2.3 Overview of Spread Spectrum LAN Technology

Spread Spectrum is the art of secure digital communications that is now being exploited for commercial and industrial purposes. Most wireless LAN systems use spread spectrum technology. Spread-spectrum transmission takes a digital signal and expands or spreads it so as to make it appear more like random background noise rather than a digital data signal transmission. Spread spectrum is designed to trade off bandwidth efficiency for reliability, integrity, and security.

Spread spectrum can be described as a transmission technique in which a pseudonoise code, independent of the information data, is employed as a modulation waveform to "***spread***" the signal energy over a bandwidth much greater than the signal information bandwidth. At the receiver the signal is "***despread***" using a synchronized model of the pseudo-noise code, in DSSS systems.

Spread Spectrum WLANs designed for use in the 2 GHz and 5 GHz ISM bands can be operated without the need for ICASA licensing under certain conditions. These systems are limited to power levels less than 1 watt and are typically intended to provide signal coverage up to about 200m. This is generally larger coverage than is provided by either IR or Microwave LANs. The use of spread spectrum transmission, combined with effective multi-user access protocols such as CSMA, make it feasible to deploy multiple systems in the same general area, even though signal coverage is overlapping. Also, these systems can be deployed with special directional antennas, allowing longer distances to be spanned, such as links between buildings on a campus or in an office park.

Spread Spectrum LANs operates at transmission rates of up to 11 Mbps in the 2.4-2.485 GHz ISM (Industrial, Scientific, and Medical) band. The IEEE 802.11 standard specifies the data rates for both frequency-hopped and direct sequence spread spectrum (FHSS and DSSS) radio transmission.

The spread spectrum LANs are well suited for small business applications where a few terminals are distributed over several floors of a building and can be served by a single system. The access method employed in spread spectrum is CSMA/CA with exponential back off.

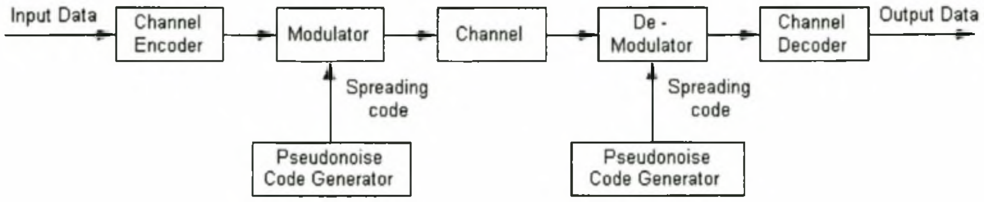


Figure 2.7: General Model of Spread Spectrum

From Figure 2.7, which shows the general model of spread spectrum digital communication system. The Input Data is fed into channel encoder, which produces an analog signal with a relatively narrow bandwidth around some centre frequency. This signal is further modulated using a sequence of digits known as a spreading code or chip sequence. A pseudonoise generator generates the spreading code. The effect of this modulation is to increase the bandwidth significantly (spread the Spectrum) of the signal to be transmitted. On the receiving end, the same chip sequence is used to demodulate the spread spectrum signal and then the signal fed into a channel decoder to recover the data[1]. Interference between legal users are avoided by orthogonality of the PN codes.

Two types of Spread Spectrum radio

1. Direct Sequence Spread Spectrum (DSSS)
2. Frequency Hopping Spread Spectrum (FHSS)

2.3.1 Direct Sequence Spread Spectrum (DSSS)

Direct-sequence spread-spectrum is a transmission technology used in WLAN transmissions where a data signal at the sending station is combined with a higher data rate bit sequence, or chipping code, that divides the user data according to a spreading ratio.

DSSS uses an 11-bit Barker Sequence to spread the data before it is transmitted. This bit pattern is called the chipping code. The chipping code is a redundant bit pattern for each bit that is transmitted, which increases the signal's resistance to interference. For each bit to be transmitted, a chipping code is assigned to represent logic 1 and 0 data bits. For example, the transmission of a data bit equal to 1 would result in the sequence 00010011100 being sent. This process spreads the RF energy across a wider bandwidth than it would be required to transmit the raw data.

Figure 2.8 shows the basic principle of DSSS for Binary Phase Shift Keying modulation, the input data (Binary data d_t with symbol rate $R_s = 1/T_s$ which is equal to bit rate R_b for BPSK) and Pseudo-noise code pn_t with a chip rate $R_c = 1/T_c$ which is an integer multiple of R_s.

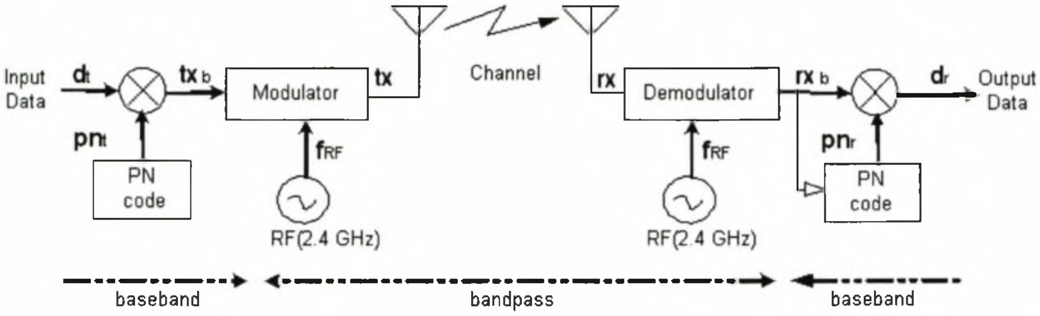


Figure 2.8: Basic Principle of DSSS

In the transmitter (spreading), the binary data d_t (for BPSK, I and Q for QPSK) is directly multiplied with the PN code pn_t, which is independent of the binary data, to produce the transmitted baseband signal tx_b:

$$tx_b = d_t \ . \ pn_t$$

The result of multiplication of d_t with a PN code pn_t is to spread the baseband bandwidth R_s of d_t to a baseband bandwidth of R_c.

In the receiver, the received baseband rx_b is multiplied with the PN code pn_r. If pn_t $= pn_r$ and synchronized to the PN code in the received data, then the recovered binary data is produced on d_r. The result of multiplication of the spread spectrum signal rx_b with a PN code pn_t used in the transmitter is to despread the bandwidth rx_b to R_s. If the receiver does not know the PN code of the transmitter, it cannot reproduce the transmitted data[18].

2.3.2 Frequency Hopping Spread Spectrum (FHSS)

Frequency hopping spread spectrum is a transmission technology used in WLAN transmissions where the data signal is modulated with a narrowband carrier signal that hops in a random but predictable sequence from frequency to frequency as a function of time over a wide band of frequencies. The signal energy is spread in time domain rather than chopping each bit into small pieces in the frequency domain. This technique reduces interference because a signal from a narrowband system will only affect the spread spectrum signal if both are transmitting at the same frequency at the same time. If synchronized properly, a single logical channel is maintained.

The transmission frequencies are determined by a spreading, or hopping, code. The receiver must be set to the same hopping code and must listen to the incoming signal at the right time and correct frequency in order to properly receive the signal. To an unintended receiver, FHSS appears to be short-duration impulse noise.

The FHSS physical layer has 22 hop patterns to choose from. The frequency hop physical layer is required to hop across the 2.4 GHz ISM band covering 79 channels. Each channel occupies 1MHz of bandwidth and must hop at the minimum rate specified by the regulatory bodies of the intended country. A minimum hop rate of 2.5 hops per second or maximum 400 ms dwell time is specified for the United States.

2.4 Overview of Narrowband Microwave LANs

The term narrowband microwave refers to the use of a microwave radio frequency band for signal transmission, with a relatively narrow bandwidth, (i.e. a narrowband radio system transmits and receives user information on a specific radio frequency). Narrowband radio keeps the radio signal frequency as narrow as possible just to pass the information. Unwanted crosstalk between communications channels is avoided by carefully coordinating different users on different channel frequencies.

A private telephone line is much like a dedicated radio channel. When each home in a neighborhood has its own private telephone line, people in one home cannot listen to calls made to other homes. In a radio system, privacy and noninterference are accomplished by the use of separate radio frequencies. The radio receiver filters out all radio signals except the ones on its designated frequency.

2.4.1 Licensed Narrowband RF

Microwave radio frequencies usable for voice, data, and video transmission are licensed and coordinated within specific geographic areas to avoid potential interference between systems. ICASA controls the licensing of narrowband radio frequencies in South Africa.

2.4.2 Unlicensed Narrowband RF

The first vendor to introduce a narrowband wireless LAN using unlicensed ISM spectrum was RadioLAN. This spectrum can be used for narrowband transmission at low power (0.5 watts or less). The RadioLANs operates at 10 Mbps or more in the 5.8 GHz band, and has a range of 50 m in a semiopen office and 100 m in open office. RadioLAN makes use of a peer-to-peer configuration.

2.5 Transmission Techniques

The transmission of a stream of bits from one device to another across a transmission link involves a great deal of cooperation and agreement between the two devices (transmitter and receiver). One of the most basic requirements is synchronization. The receiver must know the rate at which bits are being received so that it can sample the line at suitable intervals to determine the value of each received bit.

2.5.1 Synchronization Techniques

There are two most common synchronization techniques namely:

1. Asynchronous transmission and
2. Synchronous transmission.

Asynchronous transmission

Asynchronous transmission sends individual characters (one at a time) that are framed by a start bit and 1 or 2 stop bits. Figure 2.9 shows the character format of Asynchronous transmission technique.

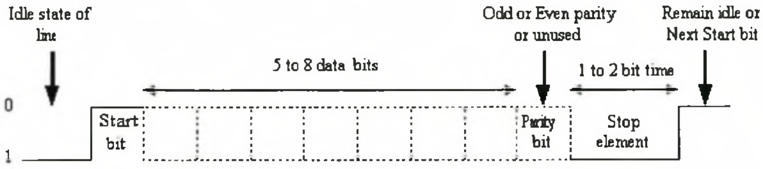


Figure 2.9: Asynchronous Transmission Format

Start and Stop bit

The purpose of the Start bit is to notify the receiving station of a new arriving character. Typically, Bitstreams are generally interpreted / read from left to right as shown in Figure 2.10. The MSB (Most Significant Bit) is sent first and the LSB (Least Significant Bit) is sent last.

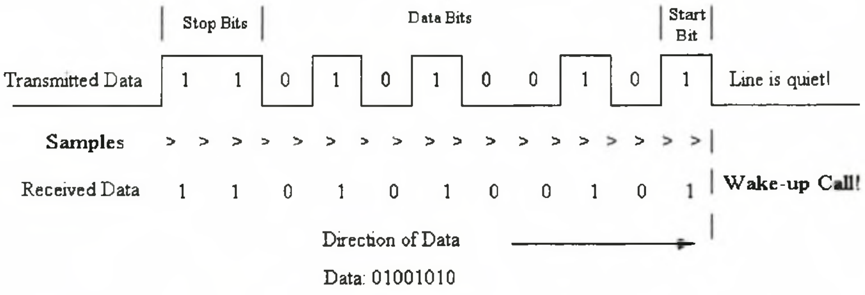


Figure 2.10: Asynchronous Transmission

The purpose of the Stop bits is to indicate the end of data. There could be 1 or 2 stop bits, with 1 being the typical number of stop bits that are used today. In Asynchronous transmission, the characters are sent individually with a quiet period in between (quiet meaning 0 bit level). Asynchronous communications requires the transmitting station and the receiving station to operating at the same fundamental clock.

The receive station starts checking for data after the Start bit is received (the Start bit is a wake up call!) as shown in Figure 2.10 above.

The receive station samples the transmitted data in the middle of each data bit. The samples are evenly spaced. They match the transmitted data because both transmit and receive clocks are operating at the same frequency.

If the receive station's clock is higher in frequency than the transmit frequency, then the samples will be spaced closer together (higher frequency - shorter period). In the above example, we transmitted the following data: 0100 1010, but we received the data: 0100 0101. The samples are out of synchronization with the transmitting data. Therefore, we would have an error in receiving data.

If the receiving station's clock is lower in frequency than the transmitted frequency, then the samples become further apart (lower frequency - wider period). Again, the samples will be out of synchronization with the transmitted data! The transmitted data is 0100 1010, but the receive data is 0101 0101! We would again have received data errors.

This is a basic problem with asynchronous transmission: both transmitter and receiver require a same clock to work properly. At high frequencies (which result in high transfer rates), clock constancy is critical and asynchronous transmission is very difficult to accomplish. Because of this problem, asynchronous transmission is generally used at low frequency/slow transfer rates, but not always.

Synchronous transmission

Synchronous transmission sends packets of data continuously. Each packet of data is formatted as a frame that includes a Starting and an Ending flag. Figure 2.11 below shows the synchronous transmission frame format.

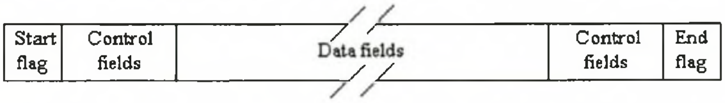


Figure 2.11: Synchronous Transmission Frame Format

The Starting flag is used to tell the receiving station that a new packet of characters is arriving, and also to synchronize the receiving station's internal clock. The End flag indicate the end of the packet. The packet can contain up to 64000 bits (or 8000 bytes), depending on the protocol. Both the Start and End flag have a special bit sequence that recognised by receiving station can recognizes to indicate the start and end of a packet. The Starting flag may consist of 1 or 2 bytes.

Synchronous transmission is more efficient than asynchronous (character transmission). For example, only 3 bytes (two Start flag and one Stop Framing bytes) are required to transmit up to 8000 bytes.

The channel efficiency is the number of bits of useful information passed through the channel per second. It does not include starting flag, ending flag, and error-detecting bits that may be added to the information bits before a message is transmitted, and will always be less than one.

Channel Efficiency = Number of data bytes / Total number of bytes transmitted

Synchronous transmission is used for high data rate transmission (100 Kbps to 100 Mbps). The timing is generated by sending a separate clock signal, or embedding the timing information into the transmission. This information is used to synchronize the receiver circuitry to the transmitter clock. Because the clocks are synchronized, much longer block lengths are possible.

When using synchronous transmission, special procedures must be adopted to permit the unique identification of the frame. This is achieved by setting the frame header to a predetermined pattern. On transmission, any repeat of this pattern in the data is destroyed by the addition of binary zeros that are removed again on reception. This process is known as **bit-stuffing**.

2.5.2 Error Detection

All communication systems try to make sure that the transmitted messages reach the destination without any difficulty. But environmental interference and physical defects in the communication medium can cause random bits errors during transmission. To avoid any difficulty, they intend to implement different error detection techniques in order to satisfy this requirement.

The aim of an error detection technique is to enable the receiver of message transmitted through a noisy channel to determine whether the message has been received correctly or not. To do this, the transmitter constructs a value called a checksum that is a function of the message, and appends it to the message. The receiver can then use the same function to calculate the checksum of the received message and compare it with the appended checksum to see if the message was correctly received.

The most common error detection techniques are:

1. Parity bit, and
2. Cyclic Redundancy Check (CRC) or Polynomial codes.

Parity bit

A parity bit is a check bit appended to an array of binary digits to make the sum of all binary digits, including the check bit, always odd (Odd parity) or always even (Even parity). In a parity system, the transmitter unit calculates the state of the parity bit and appends it to the character during transmission. The receiving unit calculates the state of the parity bit and compares the calculated value to the actual received. If they disagree, the receiver knows that a bit has been received in error.

Even Parity

Even Parity counts the number of 1s in the data to see if the total is an even number. If the number of 1s is an even number, then the Parity bit is set to 0. If the number of 1s is an odd number, the Parity bit is set to 1. This makes the total number of 1s an even number. The Even Parity Bit is used to make the total number of 1s equal to an even number.

Even Parity Checking

Even parity checking occurs when a data with even parity is received. The number of 1s in both the data and the parity bit are counted. If the number of 1s is an even number, then the data is not corrupted; if it is an odd number, then the data is corrupted.

Odd Parity

Odd Parity is the opposite of Even Parity. Odd Parity counts the number of 1s in the data to see if the total is an odd number. If the number of 1s is an odd number, then the Parity bit is set to 0. If the number of 1s is an even number, then the Parity bit is set to 1: this makes the total number of 1s an odd number. The Odd Parity Bit is used to make the total number of 1s equal to an odd number.

Odd Parity Checking

Odd parity checking occurs when data with odd parity has been received. The number of 1s in both the data and the parity bit are counted. If the number of 1s is an odd number, then the data is not corrupted; if it is an even number, then the data is corrupted. Both receive and transmit stations must agree on the type of parity checking that's used before transmitting. The parity bit is added in the asynchronous bit stream just before the stop bits (and adds to the overhead for asynchronous transmission).

Cyclic Redundancy Codes or Polynomial codes

Cyclic Redundancy Codes (CRC), or polynomial codes are the most common and most powerful error detection codes that are very popular in digital communications. Polynomial codes are based upon treating bit strings as polynomials, where the coefficients are only 0, or 1.

The CRC performs a mathematical calculation on a block of data and returns information (number) about the contents and organization of that data. So the resultant number uniquely identifies that block of data. This unique number can be used to check the validity of data or to compare two blocks.

The main idea of CRC is to treat the message as binary numbers, and divide it by a fixed binary number. The remainder from this division is considered the checksum.

The recipient of the message performs the same division and compares the remainder with the "checksum" (transmitted remainder). CRC is done with modulo arithmetic based on mod 2.

The CRC algorithm uses the term polynomial to perform all of its calculations. This polynomial is the same concept as the traditional arithmetic polynomials. The divisor, dividend, quotient, and remainder that are represented by numbers are represented as polynomials with binary coefficients. For example, the string of 8 bits: 10100111, can be represented an eight term polynomial with coefficients 1,0,1,0,0,1,1 and 1 i.e. $x^7 + x^5 + x^2 + x + 1$.

In order to do the CRC calculation; a divisor must be selected, which can be any one. This divisor is called the generator polynomial. One of the most used terms in CRC is the width of the polynomial. This width is represented by the order of the highest power in the polynomial. The width of the polynomial in the previous example is 7, which has 8 bits in its binary representation.

Since CRC is used to detect errors, a suitable generator polynomial must be selected for each application. This is because each polynomial has different error detection capabilities.

CRC algorithms are commonly called after the generator polynomial width, for example CRC-16 uses a generator polynomial of width 15 and 16-bit register and CRC-32 uses polynomial width of 31 and 32-bit register.

Three polynomials that are in common use are:

1. CRC-16 $= x^{16} + x^{15} + x^2 + 1$ (used in HDLC)
2. CRC-CCITT $= x^{16} + x^{12} + x^5 + 1$
3. CRC-32 $= x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$ (used in Ethernet)

CRC coding performs a mathematical calculation on a block of data as follows: The transmitter and the receiver must agree in advance on a generator polynomial G(x). In order to calculate a frame check sequence (FCS) of n bits, represented by the polynomial N(x), the transmitter divide N(x) by G(x) and append the remainder (or FCS) to the end of the frame. When the receiver receives the combined frame, it separate the message (Nx) and remainder (or FCS), and then divide the polynomial N(x) by generator polynomial G(x) and compare the received FCS with the calculated, if the FCS are the same it assume that there is no error.

2.5.3 Data Link Configuration

There are two characteristics that distinguish various data link configurations namely:

1. Topology
2. Communications Channels

The topology of data link refers to the physical arrangement of stations on a transmission medium.

The topology of data link is divided into two categories namely

1. Point to point link - A point-to-point link refers to the situation where there are only two stations (e.g. two computer or terminal and a computer).
2. Multipoint link - Multipoint link refers to the situation where there are more than two stations e.g. a Primary station (Master station) and a set of Secondary stations (Slave stations)

Figure 2.12 shows the multipoint configuration of wireless communication.

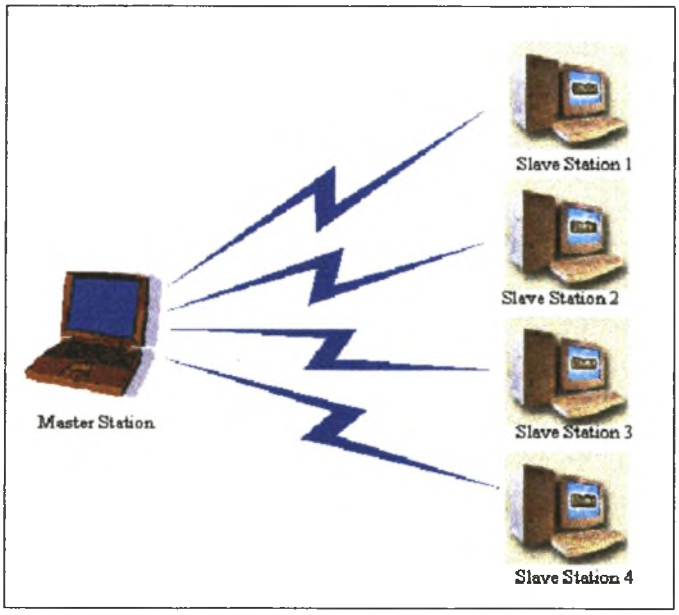


Figure 2.12: Multipoint Configuration of Wireless Link

Communications Channels

A communications channel is a pathway over which information can be conveyed. It may be defined by a physical wire that connects communicating devices, or by a radio, Infrared or other radiated energy source that has no physical presence. Information sent through a communications channel has a source (or Transmitter) from which the information originates, and a destination (or Receiver) to which the information is delivered. Although information originates from a single source, there may be more than one destination, depending upon how many receive stations are linked to the channel and how much energy the transmitted signal possesses.

In a digital data communications channel, the information is represented by data bits, which may be sum up into multiple of bits message units. A byte, which consists of eight bits, is an example of a message unit that may be transmitted through a digital communications channel. A collection of bytes may be grouped into a frame or other higher-level message unit. Such multiple of bits message units facilitate the handling of messages in a complex data communications network.

Any communications channel has a direction associated with it:

There are three types of communications channel

1. Simplex Channel - A Simplex channel is a channel whose direction of transmission is unchanging. For example, a radio station is a simplex channel because it always transmits the signal to its listeners and never allows them to transmit back.
2. Half-Duplex Channel - A Half-Duplex channel is a single physical channel in which the direction may be reversed. Messages may flow in two directions, but never at the same time, in a half-duplex system. In a telephone call, one party speaks while the other listens. After a pause, the other party speaks and the first party listens. Speaking simultaneously results in garbled sound that cannot be understood.
3. Full-Duplex Channel - A full-duplex channel allows simultaneous message exchange in both directions. It really consists of two simplex channels, a forward channel and a reverse channel, linking the same points. The transmission rate of the reverse channel may be slower if it is used only for flow control of the forward channel.

Figure 2.13 shows different communications channels.

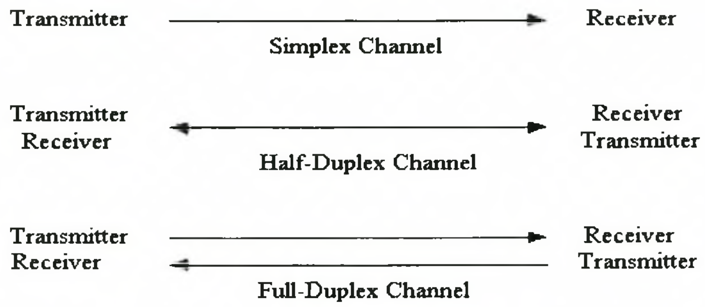


Figure 2.13: Different Communications Channnels

2.5.4 Medium Access Control

In industrial networks, several stations share the same communication media in order to save wiring costs. However, since the medium is shared, not all devices can communicate simultaneously. Therefore, there must be rules to govern who gains access to the medium and those rules called Medium Access Control (MAC).

There are several Medium Access Control implementations, but they basically fall into two main categories;

1. Carrier Sense Multiple Access / Collision Detection (CSMA/CD) and
2. Token-passing.

2.5.4 (a) CSMA/CD

CSMA/CD is the most popular method of gaining access to network.

In CSMA, there is no scheduled time for any station to transmit; station transmissions are ordered randomly. When the station needs to transmit data, it first listens to the channel to determine whether it is busy or not. If found channel not busy, it transmits data immediately. If channel busy, then it waits until the channel is not busy. When station is transmitting, it listens to the channel. If detects the collision it stops transmitting and wait a random period of time before retransmitting [1]. This type of medium access control is very common and is basis for shared Ethernet.

2.5.4 (b) Token-passing

In Token-passing, each station to the network is guaranteed some time to transmit data on a permission basis. This permission occurs when a station receives the one token that exists in the network. The token is passed from one station to another station in a circular logical ring. Once a station receives the token, the station must initiate a transmission or pass the token to the next station in orderly fashion. Each station is assigned station address.

Token-passing is deterministic in the industrial networks when events occurs in a timely manner, because all stations have equal access to the network and network collisions are avoided by restricting a transmission to one and only one station.

2.6 Summary

Wireless data communication, which includes Wireless LANs were discussed in the chapter. This includes the general overview of wireless transmission technologies and transmission techniques.

Chapter 3

Comparative WLAN Technologies

We have looked at three different transmissions technologies for WLAN. Our purpose for this has been to select the most suitable transmission technology to use for the application as intended in a WLAN. In this chapter we will discuss advantages and disadvantages of each WLAN transmission technologies in relevant areas.

3.1 Comparison Between Transmission Technologies for WLAN's

3.1.1 Infrared LANs versus Spread Spectrum and Narrowband Microwave LAN

The main advantages of Infrared LAN technology

1. Ability to carry a high bandwidth.
2. Offer higher degrees of security and performance - directionality of the beam helps ensure that data isn't leaked or spilled to nearby devices as it's transmitted.
3. Infrared emission does not penetrate opaque objects - higher degrees of security and performance than microwave. Separate infrared installations can be operated in the same building without interference.

4. Few international regulatory constraints: IrDA (Infrared Data Association) functional devices will ideally be usable by international travelers, no matter where they will be.

5. High noise immunity: not as likely to have interference from signals from other devices.

6. In IR receivers' detection of the amplitude of the optical signals is needed only (detection of frequency and phase is required for microwave receivers).

The main disadvantages of Infrared LAN technology

1. Line of sight: transmitters and receivers must be almost directly aligned (i.e. able to see each other) to communicate.

2. The range is limited with respect to Radio LANs (Spread Spectrum and Narrowband Microwave).

The main advantages of Spread Spectrum LAN technology

1. Has the ability to eliminate or alleviate the effects of multi-path interference.

2. Can share the same frequency band (overlay) with other users.

3. Provide privacy due to unknown random codes.

4. Involves low power spectral density since signal is spread over a large frequency band.

The main disadvantages of Spread Spectrum LAN technology

1. Bandwidth is sometimes insufficient.

2. Implementation is somewhat complex

3. The code generator used for generating pseudonoise sequence should match the speed of the information signal for modulation; hence a fast code generator is required.

The main advantages of Narrowband Microwave LAN technology

1. One advantage of licensed narrowband LAN, is that it guarantees interference free communication, unlike unlicensed spectrum, such as ISM. Licensed spectrum gives the license holder a legal right to an interference free data communications channel. Users of an ISM band LAN are at risk of interference disrupting their communications, for which they may not have a right of removing anything undesirable.

The main Disadvantages of Narrowband Microwave LAN technology

1. One major disadvantage to the use of Narrowband microwave LAN technology is that the frequency band used requires licensing by the ICASA. Once a license is granted for a particular location, that frequency band cannot be licensed to anyone else, for any purpose, within a 28.164 Km radius.

3.1.2 Selection of Transmission Technology for WLAN's

There are a number of things to take into consideration when selecting a transmission technology to use for WLAN. Such aspects can be range, transmission power, data rate and security[21]. As seen in the description of the transmission technologies, each system has a different range depending of various factors such as power and frequency use. Table 3.1 below summarizes the comparison of Wireless LAN technologies.

	Infrared LAN	**Spread Spectrum LAN**	**Narrowband Microwave LAN**

Wavelength / Frequency	$3 \times 10^{14} Hz$;
	λ : 800 to 900
		902 to 928 MHz
		2.4 to 2.4385 GHz
		5.725 to 5.825 GHz
			902 to 928 MHz;
			5.2 to 5.775 GHz;
			18.82 -19.205 GHz

Range (m)	15 to 60	30 to 200	10 to 40
Line of sight required	Yes	Almost	Yes
Transmit power	N/A	Less than 1 W	25 mW
License required	No	No	Yes unless ISM
Interbuilding use	Possible	Possible with Antenna	Conditional
Data Rate (Mbps)	1 to 16	1 to 50	10 to 20
Modulation technique	ASK	FSK / QPSK	FS / QPSK
Access method	Token ring, CSMA	CSMA/CD	ALOHA, CSMA/CD

Table 3.1: Comparisons of Wireless LAN Transmission Techniques

Infrared emission does not penetrate walls, resulting in a considerable higher degree of security and performance than microwave by confinement of the transmission within an office or other work area. The only way for Infrared signals to be detected outside the installation area is through windows, which can easily be covered by curtains or shades. The confinement of Infrared signals by walls also allows concurrent usage of similar systems in neighboring office without mutual interference.

Spread Spectrum LANs are license free and as a result, they are the most popular variety of radio LANs and are used where the ranges required are significantly higher (compared to Infrared LANs), though setting up a radio LAN is in itself an expensive business sometimes.

Narrowband Microwave LANs offer higher data rates than Infrared LANs but have to be licensed and coordinated within geographic areas to prevent interference between systems. Microwave LANs are best suited for areas like an open office where there are very few obstructions like concrete walls and floors.

Based on the requirements given in Chapter 1, Section 1.3 and some important issues seen in this section, we summarize our conclusion for the theoretical part of this project:

1. All three transmission technologies are used for short-range real time Wireless LAN's.
2. Infrared LAN fulfills the requirements for this project as it concerned. Infrared LAN is chosen because of the following reasons:

- Spectrum for infrared virtually unlimited: possibility of high data rates
- Infrared spectrum unregulated.
- Equipment inexpensive and simple compared to Spread Spectrum and Narrowband Microwave LAN.
- Reflected by light-colored objects: ceiling reflection for entire room coverage.
- Doesn't penetrate walls: More easily secured against eavesdropping and more importantly, less interference between different rooms, or instrumentation areas

Spread Spectrum and Narrowband Microwave LAN's does have higher speed and range than Infrared LAN's, but Spread Spectrum and Narrowband Microwave LAN's are only permitted in certain frequency bands, radiated power is limited and very limited ranges of license-free bands are available. They are not the same in all countries either. Their cost could also burden the typical instrumentation installation in industrial applications where production cost is always important.

3.2 Summary

The different aspects that are of importance for wireless transmission technologies were discussed. A comparison was made between Infrared LANs, Spread Spectrum LANs and Narrowband Microwave LANs. The most appropriate transmission technology (i.e. Infrared LANs) was selected to be implemented in the project.

Chapter 4

Implementation of IrDA Protocol

4.1 Environment

The controlling software for this Wireless Local Area Network for instrumentation was developed in Delphi 7.

The hardware used was two IrDA adapters from Bafo Technologies, connected to the computers through a Universal Serial Bus (USB) port. An I/O board (Data Acquisition and Process Control board) with 3 digital I/O ports, 4 analog outputs channels and 16 single ended, or 8 differential mode inputs from Eagle Technology, was also used.

4.2 Design

The goal with the design was to make it simple, flexible, and easily scaleable, meaning that users could be enabled to access real-time information anywhere in their particular Industry. To achieve the aim, the IrDA protocol (IrLAP protocol layer) was developed and implemented in this project, which is an Infrared Link Access (Master and Slave) Protocol. The design of the IrLAP protocol includes the packet frame formatting, as will be discussed in more detailed later in this chapter, in Section 4.2.2 and Section 4.3.

There are four key design aspects that were important in the design of a network, namely

1. Addressing
2. Communications channel
3. Error detection and error correction and
4. Flow control

Since a network might have many computers, some of which run multiple processes, a means is needed for a process on one machine to specify with whom it wants to communicate. As a consequence of having multiple destinations, some form of Addressing is needed in order to specify a specific destination. In this project, each slave station has its own address and the master station was designed to connect up to sixteen Slave stations.

Another set of design decisions concerns the rules for data transfer. In some systems, data only travel in one direction; in others, data can travel in both ways. The protocol must also determine how many logical channels the connection corresponds to and what their priorities are. Many networks provide at least two logical channels per connection, one for normal data and one for urgent data. In this project the half-duplex channels were used, meaning that only one station is allowed to send data at a time.

Because physical communication circuits are not perfect, the use of error control was implemented in the design. The design of error control will be discussed in more detail later in this chapter, in Section 4.3.1.

We also use flow control to ensure that the source (or transmitter) does not overwhelm the destination (receiver) by sending data faster than can be processed and absorbed.

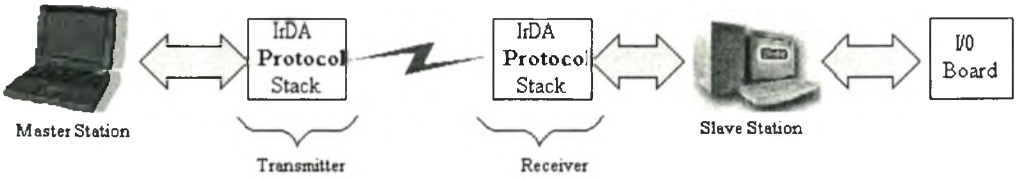


Figure 4.1: Network Design Configuration

Figure 4.1 above shows the network configuration of the project.

4.2.1 IrDA Protocol Stack

The Figure 4.2 shows an integrated of IrDA protocol stack in an embedded system [10].

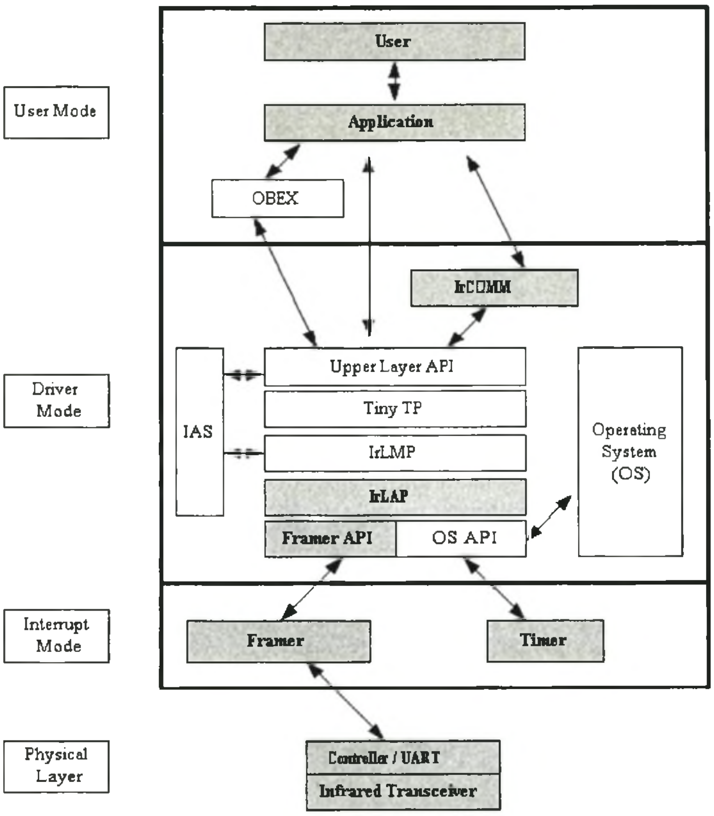


Figure 4.2: Integration of IrDA Protocol Stack into an Embedded System

Note: Shaded areas shows software developed under this project, to be embedded with standard routines and of systems

4.2.1 (a) Physical Layer

The IrDA physical layer provides half duplex point-to-point communication through the IR medium and provides services to the upper IrLAP layer. The encoding of the data bits and framing of data, such as: begin and end of frame flag (BOFs and EOFs) and cyclic redundancy check (CRCs), are performed by the physical layer, although this is generally implemented in software. A more detailed description of the physical layer has been included in Chapter 2, Section 2.2.1.

4.2.1 (b) Interrupt Mode

A software layer called Framer is created in order to isolate the remainder of the stack from over burdening the hardware layer. The main responsibility of Framer is to accept incoming frames from the hardware and present them to the Link Access Protocol layer. This includes accepting outgoing frames and doing whatever is necessary to send them. In addition, the Framer is responsible for changing hardware speeds at the bidding of the IrLAP.

4.2.1 (c) Drivers Mode

Link Access Protocols

Immediately above the Framer we encounter the IrLAP layer. The IrLAP layer is characterized by a half duplex connection with master and slave station roles. There can be multiple slave stations in the link but only one master station. A more detailed description of the IrLAP layer has been included in Chapter 2, Section 2.2.2.

Link Management Protocol

The IrLMP provides two distinct different types of services.

- Firstly, it provides a level of connection oriented multiplexing (LM-MUX) on top of IrLAP. The LM-MUX provides a simple level of switching over the top of an IrLAP connection. It also hides the master/slave nature of IrLAP from the application and provides a symmetrical set of services to IrLMP clients.
- Secondly it provides an Information Base that holds detail of the application entities present in the local station that is current offer services to other IrDA devices. Objects in this information base carry the essential addressing information necessary to establish communication with the corresponding application entities. An Information Access Server and a corresponding Client provide access to this Information Base. Together the Information Base, the Server and the Client provide an Information Access Service (LM-IAS). Both LM-IAS Client and Server entities are LM-MUX clients. A more detailed description of the IrLMP layer has been included in Chapter 2, Section 2.2.3

IrCOMM

An IrCOMM implementation generally takes the form of a system-installable serial port driver. To develop the IrCOMM was to convert USB serial line state change into protocol messages that are communicated to the peer application through the native serial API. A more detailed description of the IrCOMM layer has been included in Appendix A.

4.2.2 Design of the IrLAP Protocol

4.2.2 (a) IrLAP Frame Format Design

The IrLAP frame format shown in Figure 4.3 was designed and implemented in this project. This IrLAP frame format is sent and received on the infrared media for 1.152 Mbps data rate (half duplex).

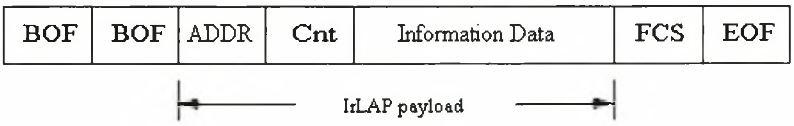


Figure 4.3: IrLAP Frame Format

The designed frame format consists of the following elements:

- Two Beginning of Frame (BOF) flags that mark the beginning of the frame. The size of each of the BOF is 8 bits long. Infrared transceivers synchronize with the infrared signal while they receive Beginning of Frame (BOF) flags at the beginning of each incoming frame. BOF is defined as 0x7E (Hex) or 01111110 (binary).

- Address (ADDR) - Address identifies the slave station connection address. The address is 8 bits long. The first bit of the address is used as the command and response (C/R) bit. If this bit is set, the packet specifies that it is being sent from the master station to the secondary station. The master station initiate connections to slave stations and run the timers that keep those connections active. The least-significant 7 bits of the address specify the Link Access Protocol (LAP) address.

- Control (Cnt) - Control specifies the function of the particular frame. The control field is eight bits long.

- Optional information - Information contains the information data.
- Frames check sequence (FCS) - Frames check sequence is an error detection code calculated from the remaining bits of the frame, exclusive of flags (BOF and EOF). It allows the receiving station to check the transmission accuracy of the frame. The FCS is 16 bits or cyclic redundancy check (CRC-16). FCS is calculated over Address (ADDR), Control (Cnt), and information Data on an IrLAP frame as the packet's CRC-16.
- End of frame (EOF) - EOF signals the end of the frame. The size of the EOF is 8 bits long and is defined as 0x7E (Hex) or 01111110 (binary).

4.3 Implementation of Link Access Protocol

4.3.1 Error Control

Cyclic Redundancy Check (CRC-16)

To make sure that the transmitted message from either Transmitting Station or Receiving Station reach the destination without any problem, CRC-16 algorithms was implemented to detect possible corrupted messages.

The transmitter calculates the CRC-16 of the message, which then appends the remainder to the message as FCS. The receiver recalculate the CRC-16 of the message, and compares the calculated value to the actual value it received in the CRC field, if they don't match, the receiver will then send back a NAK message, which means that there will be a retransmission until the message is received with no error.

CRC-16 checking was implemented in both the master and the slave stations. Figure 4.4 shows the flow chart of CRC-16, as implemented and would be described as follows:

A byte count is set for data to be sent, and then initialise the 16-bit remainder (CRC-16) register to all zeros. XOR the first 8-bit data byte with the high order byte of the CRC-16 register, and the result would be the current CRC-16. Initialise the shift counter to 0 (j = 0), and the current CRC-16 register is shifted by 1 bit to the right. Check if there is a carry. If there is a carry, then XOR the generating polynomial (Hex 18005) with the current CRC-16 and increment the shift counter by 1, or else, increment the shift counter by 1. Check if the shift counter is greater than 7. If is not greater than 7, then shift the current CRC-16 register 1 bit to the right, or else increment the byte count. Check if the byte count is greater than the data length. If the byte count is not greater than the data length, then XOR the next 8-bit data byte with the current CRC-16 and initialise the shift counter to 0 (j = 0), or else add current CRC-16 to the end of data message for transmission and exit.

Note: On the receiver side, the CRC-16 would be calculated as shown in Figure 4.4, but the only difference is that, the remainder would not be added to the data message, however, it would be compared to the received CRC-16.

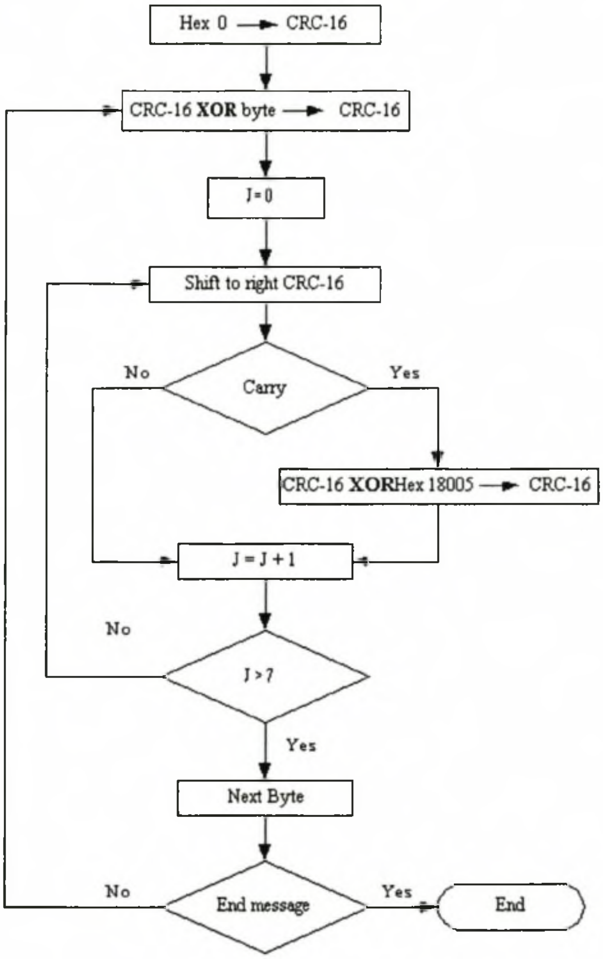


Figure 4.4: CRC-16 Flow Chart

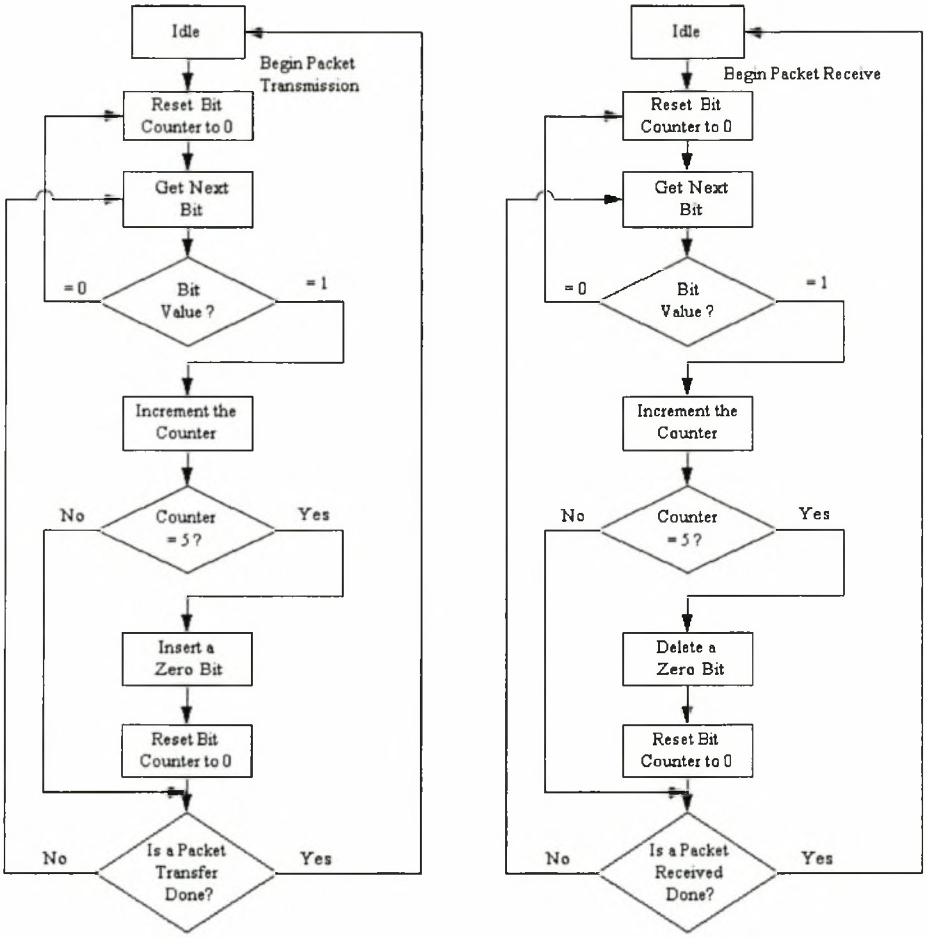
4.3.2 Bit Stuffing

To avoid uncertainty in the frame with BOF and EOF, a procedure known as Bit Stuffing was implemented as shown in Figure 4.5(a). Between the transmission of the starting and ending flags (BOF and EOF), the transmitter always inserts a zero after every five consecutive ones in the data stream. Figure 4.5(a) shows the Bit Stuffing employed by the transmitting station when it is beginning to transmit the packet.

After detecting a starting flag, the receiver monitors the data streams. When a pattern of five consecutive ones in the data streams appears, the sixth bits is examined, if this bit is 0, it is deleted. This procedure is known as Bit Destuffed. Figure 4.5(b) shows the bit stuffing flow chart on the receiver.

Figure 4.5 shows the Bit Stuffing and Bit Destuffing flow charts, as implemented and would be described as follows:

If there is a packet transfer, the bit counter would be initialised to zero, and then get the first bit from a packet transfer. Compare if the bit value is 1 or 0. If the bit value is 0, then reset the bit counter to zero, or else, increment the bit counter by 1. Compare if the bit counter is equal to 5. If bit counter is equal to 5, then insert a zero bit, and reset counter to zero, or else, compare if a packet transfer is done, if not , get another packet transfer, or else get the next bit value.



(a) Bit Stuffing flow chart

(b) Bit Destuffing flow chart

Figure 4.5: Bit Stuffing and Bit Destuffing Chart

4.3.3 Master Station and Slave Station Negotiation

The IrLAP layer specifies seven parameters that stations must negotiate before data transfer may commence. The parameters oversee the size of the packets, the speed at which they are sent, and the timing of their transmission.

Negotiation parameters are divided into two groups: type 0 and type 1. Master and Slave station must agree on the same value for type 1 parameters but may use different values for type 0 parameters.

Baud rate and link disconnect/threshold time are the only type 0 parameters. The type 1 parameters include minimum and maximum turnaround time, data and window size, and the number of additional beginning of frame (XBOF) bytes. Table 4.1 shows the seven IrDA negotiation parameters, with their types and permissible values.

Negotiation Parameter	**Type**	**Permissible Values**
Baud Rate	0	9600 bps, 19.2 Kbps, 38.4 Kbps, 57.6 Kbps, 115.2 Kbps, 576 Kbps, 1.152 Mbps, 4 Mbps 16 Mbps
Link Disconnect/Threshold Time	0	3 s, 8 s, 12 s, 16 s, 20 s, 25 s, 30 s, 40 s
Maximum Turnaround Time	1	500 ms, 250 ms, 100 ms, 50 ms
Data Size	1	64 B, 128 B, 256 B, 512 B, 1024 B, 2048 B
Window Size	1	1-7 frames
Additional BOFs	1	Below 115.2 Kbps: some fraction of value below 115.2 Kbps: 48, 24, 12, 5, 3, 2, 1, 0 576 Kbps, 1.152 Mbps: 2, 0 4 Mbps: 0
Minimum Turnaround Time	1	10 ms, 5 ms, 1 ms, 0.5 ms, 0.1 ms, 0.05 ms, 0.01 ms, 0 ms

Table 4.1: IrDA Negotiation Parameters as Implemented

The link disconnect/threshold time determines the length of time the Master station and Slave station will maintain their link if invalid data (or no data) is received. Each time a station sends data, it starts a timer that counts down to the link disconnect time. Once it receives a response from the other station, it resets the timer. If a station receives no response, it will continue to send data and wait for a response until the timer reaches the disconnect time. If the two stations are unable to communicate for the duration of the link disconnect/threshold time, both sender and receiver will close the link simultaneously.

The minimum and maximum turnaround times control the timing of IrDA transmissions. IrDA data transmission is half-duplex, meaning that only one station may transmit data at a time. To ensure reliable data transmission, the protocol requires that the station sending data stop transmission to receive acknowledgements from the receiving station. Since no data is transmitted during these breaks, the length of time required receiving the acknowledgement affects throughput significantly. To maximize throughput, stations would ideally send as much data as possible before pausing, and then wait for as little time as possible. Due to physical limitations, however, a longer pause is sometimes necessary. While a station is sending data, its transceiver blinds its own receiver such that it cannot notice remote infrared pulses. After an infrared transceiver finishes sending data, it waits to recover from sending before it can receive from the other station.

The minimum turnaround time allows a station to specify the length of time the other must wait, before it begins to transmit, thus providing itself sufficient time to recover from data transmission. Each station specifies its own required minimum turnaround time. Higher quality transceivers will require less time to recover from data transmission than lower quality transceivers.

The maximum turnaround time specifies the maximum amount of time a station may transmit before it must stop and wait to receive an acknowledgement. At baud rates up to 115.2 Kbps, 500 ms is used. At higher baud rates, the maximum turnaround time is shorter.

The data size specifies the number of bytes that may be contained in one frame. The window size is the number of contiguous frames that may be transmitted before an acknowledgement must be received. Smaller window sizes enable stations with limited memory to process incoming data without overflowing their buffers and losing data. The IrLAP specification states that the maximum turnaround time has priority over the window and frame sizes in determining when link turnaround must occur. If a station decides data and window sizes such that the total number of bytes in a window cannot be transmitted in the time allotted by the maximum turnaround time, the data and window sizes must be adjusted accordingly.

Figure 4.6 shows the data transmission between master station and slave station with window size equal to 1. The master station sends one frame to slave station and waits for acknowledgement from the slave station. For example, the master station is sending the maximum of 1024 byte packets at 1.152 Mbps. Each packet requires 4.1 ms to transmit.

For a latency of 5 ms, the total transmission time is calculated as follows:

$$\begin{aligned}\text{Total transmission time} &= (T_{pf} + T_l) \times N_f \\ &= (4.1 \text{ ms} + 5 \text{ ms}) \text{ x } 7 \\ &= 63 \text{ ms}\end{aligned}$$

Where T_{pf} is a time per frame, T_l is latency time and N_f is number of frames.

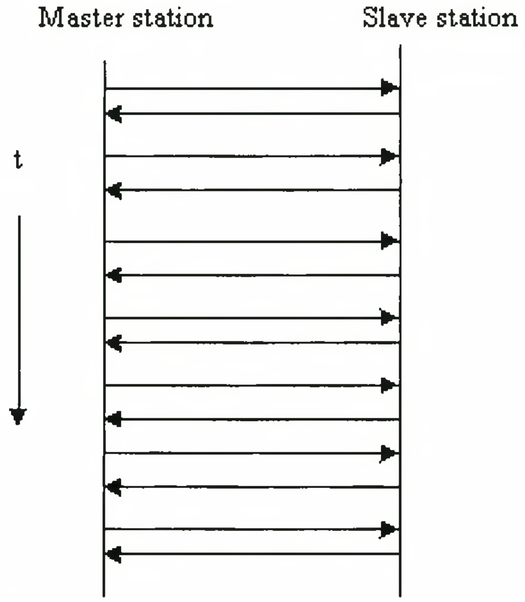


Figure 4.6: Timing Diagram of Window Size

Transmitting Station

The transmitting station generates the data message and sends it to receiving stations as shown as follows:

The transmitting station converts the data message to binary (i.e. ones and zeros), and computes CRC-16. Therefore, the bit stuffing would be performed, and put the data message in a frame format by adding the BOFs and EOF to the binary message (i.e. two BOFs would be added at the begin and one EOF would be added to the end of binary message). After frame formatting, the frame would be sent to receiving station.

Figure 4.7 below shows the transmitter flow chart.

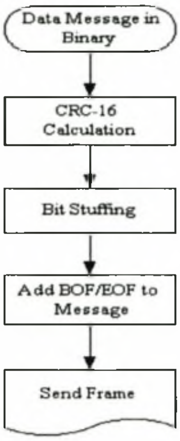


Figure 4.7: Transmitter Flow Chart

Receiving Station

The receiving station processes the received data message as follows:

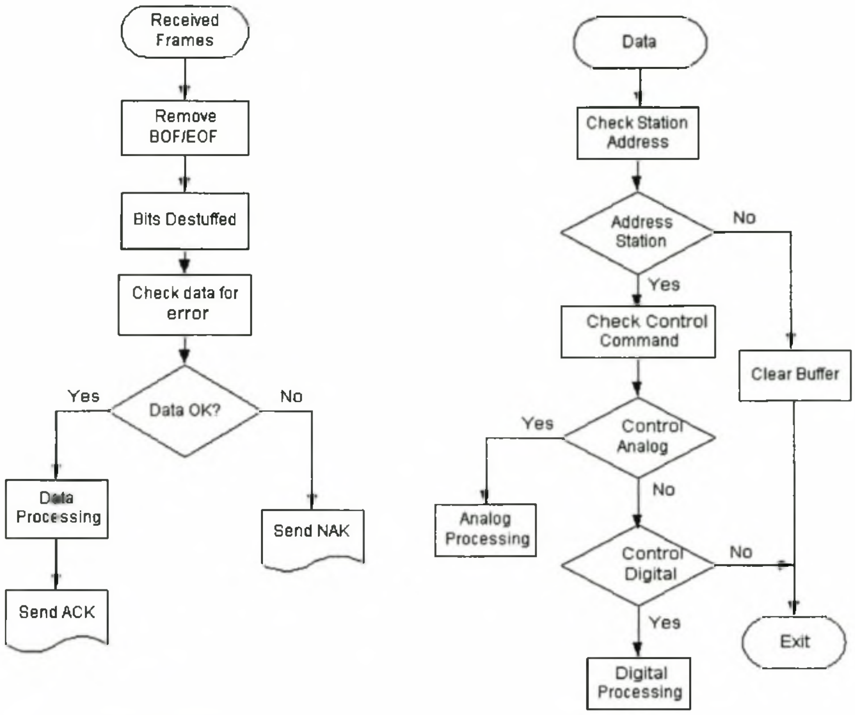
The receiving station removes the BOFs and EOF from received frame before bit destuffing and CRC-16 computation, respectively. Check the data message for error, if there is error, send the Negative Acknowledge (NAK) to the transmitting station, else it process the data message as shown in Figure 4.8 (b) and send the positive Acknowledgement (ACK) to the transmitting station.

Figure 4.8 (b) shows the data processing by the receiving station. The receiving station checks an address. If the address is incorrect, it clears the buffer and ignores the message, or else it checks the control command. If the control command is analog control, it process the data message as shown in Figure 4.9 (a), or else it process the data message as shown in Figure 4.9 (b).

Figure 4.9 shows analog and digital data message processing flow chart.

Figure 4.9 (a) shows the analog data processing by receiving station. The receiving station checks if the analog control is read or write control and process it as, will be described later in Chapter 5, Section 5.1.2 (b).

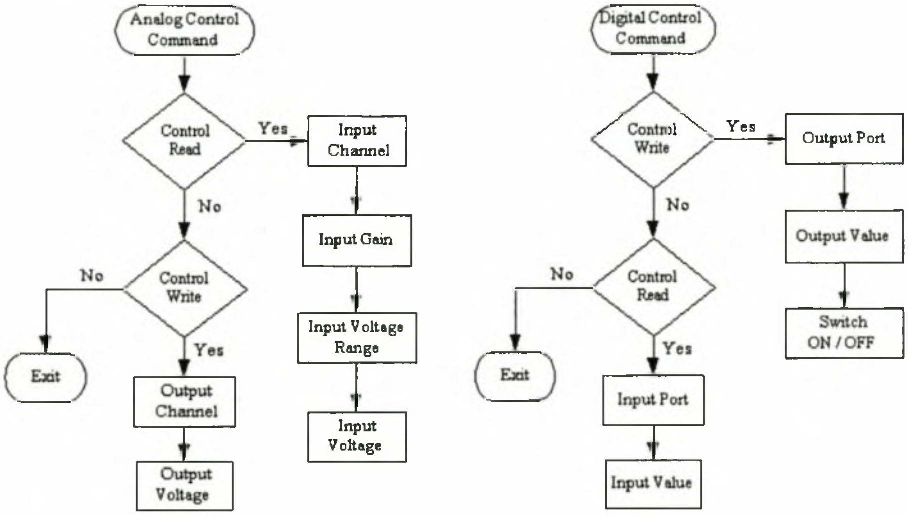
Figure 4.9 (b) shows the digital data processing by receiving station. The receiving station checks if the digital control is read or write control and process it as, will be described later in Chapter 5, Section 5.1.2 (a).



(a) Receiver flow chart

(b) Packets frame Processing

Figure 4.8: Receiver Flow Chart



(a) Analog Processing Flow Chart

(b) Digital Processing Flow Chart

Figure 4.9: Analog I/O and Digital I/O Flow Chart

4.4 Implementation of Delphi 7 to Access an I/O Board

Figure 4.10 shows the PCI-730 I/O board. The PCI-730 I/O board is a high performance data acquisition board for the PCI-bus. It is multi-function I/O board, which feature both analog and digital I/O on the same board. The PCI-730 board has industry standard connectors.

Extremely compact, the PCI-730 features 16 single-ended or 8 differential 14-bit analog input channels with an overall sampling speed of 100KHz, and four 14-bit analog output channels. It is ideal for laboratory use as voltage references.

Typical applications include analog input streaming, voltage measurements, voltage reference outputs, analog and digital data logging, digital I/O for control of relays, frequency counter for event logging, etc. It has an Internal Cable for digital I/O (IDC40 to DB37)

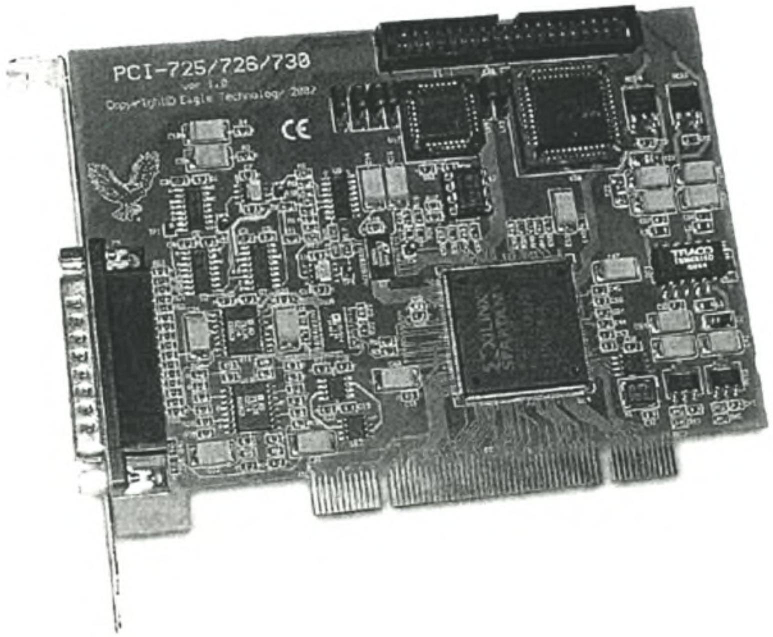


Figure 4.10: PCI-730 I/O Board

4.4.0 (a) Programming the I/O Board

Traditionally, measurements are done on stand-alone instruments of various types, i.e oscilloscopes, multi meters, counters, process measurement, etc. However, the need to record the measurements and process of the collected data for visualization has become increasingly important.

There are several ways in which the data can be exchanged between instruments and a computer. Many instruments have a serial port, which can exchange data to and from a computer or another instrument. As an example, use of a GPIB interface board (General Purpose Instrumentation Bus) allows instruments to transfer data in a parallel format and gives each instrument an identity among a network of instruments. Other instruments have a simple 4 - 20mA output for process measurement.

Another way to measure signals and transfer the data into a computer is by using a Data Acquisition (DAQ) board. A typical commercial DAQ board contains Analog to Digital conversion (ADC) and Digital to Analog Conversion (DAC) that allows input and output of analog and digital signals in addition to digital input/output channels.

Since the project is concentrated on the development of a WLAN for instrumentation purposes, a programmable I/O board was used as an Instrumentation Bus.

A programmable I/O board (Data Acquisition Board) from Eagle Technology was used for Analog and Digital I/O. The I/O board, which contains 3 digital I/O ports and 16 single ended or 8 differential mode inputs, is assembled in the slave stations. Delphi 7 was used for programming and configuring the I/O board.

Various aspects of the I/O board.

1. **Sampling**

   An Analog to Digital Converter uses a process called sampling. Sampling an analog signal involves taking a sample of the signal at discrete times. The rate at which the signal is sampled is known as the sampling frequency. The sampling frequency determines the quality of the analog signal that is converted. A higher sampling frequency achieves better conversion of the Analog Signals.

2. **Analog to Digital Conversion (ADC)**

   Once the signal has been sampled, one needs to convert the analog samples into a digital code. This process is called analog to digital conversion.

   The I/O board that is used in this project has 14-bit analog input resolution. Analog input ranges of $\pm 2.5V$, $\pm 5V$ and $\pm 10V$. Maximum analogue input sampling rate of 200 kS/s. It has four analogue output channels, each with a resolution of 14 bits and full-scale range of $\pm 10V$ at 5 mA. It has three eight bit Digital I/O channels, three user counter timers and 16 channel analog input. This makes it possible to acquire up to 16 analog signals in parallel (however, the sampling frequency will be divided by the number of parallel channels).

   - **Resolution**

     The resolution is an important characteristic of the I/O board. Accuracy of the analog input signal converted into digital format is dependent upon the number of bits the ADC uses. The resolution of the converted signal is obviously a function of the number of bits the ADC uses to represents the digital data. An 8-bit ADC gives 256 levels (2^8).

   - **Settling Time**

     On the I/O board, the analog signal is first selected by a multiplexer, then amplified before the ADC converts it. The amplifier used between multiplexer and ADC must be able to track the output of the multiplexer, otherwise the ADC will convert the signal that is still in transition from the previous channel value to the current channel value. Poor settling time is a major problem because it changes with sampling rate and the gain of the I/O board.

3. **Digital to Analog Conversion (DAC)**

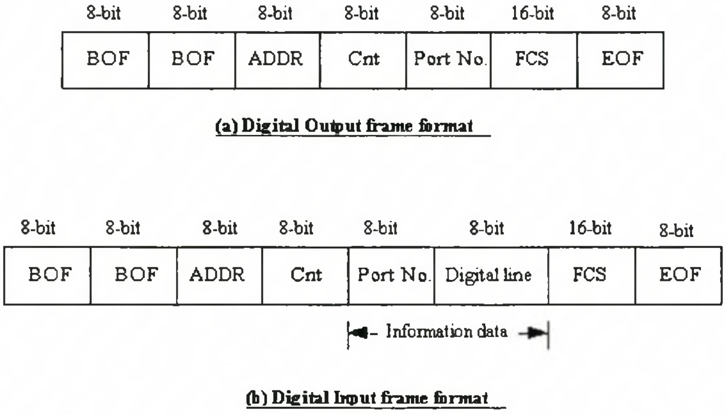
   The I/O board has digital to analog converters (DAC). A DAC can generate an analog output from a digital input. This allows the board to generate analog signals, both direct current (dc) and alternate current (ac) voltages. Like the ADC, the number of samples it can process and the number of bits that is used in converting the digital code into an analog signal limit the DAC's performance.

4.4.1 Digital Input/Output

Digital I/O sections have digital I/O circuits that interface to on/off sensors such as pushbutton and limit switches; and on/off actuators such as motor starters, pilot lights, and annunciators. These outputs are directly controlled by the state of corresponding processor data bits. These inputs directly control the state of corresponding processor data bits. The I/O board has got 3 digital ports and each port is only 8-bit wide.

To demonstrate and test digital I/O functions, eight states switches were built on the slave station, and linked to the programmable I/O board parallel port on the slave station. The master station sends the digital I/O controls to control and monitor the switches that are on the slave station (i.e. digital input control was used to read the status of the slave station while digital output control was used to write to the slave station).

The packet frame for digital I/O is formatted as shown in Figure 4.11:



8-bit	8-bit	8-bit	8-bit	8-bit	16-bit	8-bit
BOF	BOF	ADDR	Cnt	Port No.	FCS	EOF

(a) Digital Output frame format

8-bit	8-bit	8-bit	8-bit	8-bit	8-bit	16-bit	8-bit
BOF	BOF	ADDR	Cnt	Port No.	Digital line	FCS	EOF

(b) Digital Input frame format

Figure 4.11: Digital I/O Packet Frame Format

The Control (Cnt) can either be a digital input control or digital output control.

For digital input control, the Information Data consists of digital port number to read to as shown in Figure 4.11 (a).

For digital output control, the Information Data field consists of a digital port number to write to and digital line on that port to write to as shown in Figure 4.11 (b).

The master station controls the switches connected to I/O board on the slave station as follows:

It sends the digital I/O packet to the addressed slave station. The slave station will receive the packet and process it, and then check if the packet was received successful or not. If the packet was received successfully, then it first checks if the Control (Cnt) is for digital input or digital output. If for digital output, then the digital I/O port to write to is checked, as well as the digital I/O line on that port that the user wants to write to. A Digital to Analog Conversion (DAC) is then performed and those values written to the corresponding digital port.

If the value to write to digital port is 0, then all the data pins of the digital port will be set to low level, which means that all switches connected to these data pins will turn OFF, while if the data value written to the digital port is 255, then all the data pins of the digital port will set to high level, which means that all switches connected to these data pins will turn ON. If the data value to write to the digital port is 1, then the data pin1 (PA0) will set to high level and all others to low level, which means that only a switch connected to data pin1 will turn ON, whereas others are OFF. Figure 4.9 (b) above shows the Digital I/O control flow chart.

Calculating the Analog value to write to the digital port.

The Master Station was designed with eight states of light to indicate whether the line on this port is high or low. When the switch is ON, it takes the corresponding value of the switch as 1 and if the switch is OFF, it takes the corresponding value of the switch as 0. All the values of these switches will then be added up to 8-bit.

For example if the switch 1 is ON and all others switches are OFF, the corresponding values in binary to all the switches will be 00000001 (or 1 in decimal). If all the switches are ON, their corresponding value in binary will be 11111111 (or 255 in decimal). Every bit of the binary number controls one output bit. Table 4.2 shows the relationship of the bits, digital port output pins (data pins) and the value of those bits.

Pin	Bit	Analog Value
1	PA0	1
20	PA1	2
2	PA2	4
21	PA3	8
3	PA4	16
22	PA5	32
4	PA6	64
23	PA7	128

Table 4.2: Relation of Bits, Digital port and Analog values of those bits

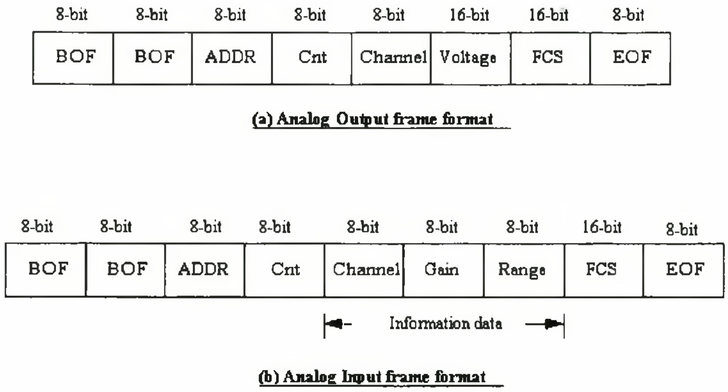
For example, if the user wants to set pins 1 and 4 to high level (logic 1 or ON), then he needs to output analog value 1 + 64 = 65. If user wants to set pins 1, 2, 3 and 20 to high level (logic 1 or ON), then he needs to output value 1 + 4 + 16 + 2 = 23.

4.4.2 Analog Input/Output

Analog I/O modules perform the required A/D and D/A conversions to directly interface analog signals to the processor, with 14-bit resolution. Analog I/O can be user-configured for the desired fault-response state in the event that I/O communication is disrupted. This feature provides a safe reaction/response in case of a fault.

For analog I/O, 16 analog input channels, 4 analog output channels, were built on the slave station and linked to the programmable I/O board parallel port on the slave station. The master station sends analog I/O controls to control and monitor the voltage (i.e. analog input control was used to read the status of the slave station while analog output control was used to write to the slave station).

The packet frame for analog I/O was formatted as shown in Figure 4.12



8-bit	8-bit	8-bit	8-bit	8-bit	16-bit	16-bit	8-bit
BOF	BOF	ADDR	Cnt	Channel	Voltage	FCS	EOF

(a) Analog Output frame format

8-bit	8-bit	8-bit	8-bit	8-bit	8-bit	8-bit	16-bit	8-bit
BOF	BOF	ADDR	Cnt	Channel	Gain	Range	FCS	EOF

(b) Analog Input frame format

Figure 4.12: Analog I/O Packet Frame Format

Analog I/O controls (Cnt) can either be an analog input control or analog output control. For an analog input control, Information Data consists of the channel number to sample, Gain at which the sampling should take place and voltage range as shown in Figure 4.12 (b). The analog input function can be configured for number of gain settings. Using different gain settings will give varying degrees of accuracy.

For analog output control, Information Data consist of the output voltage to be written and the channel to which to output this voltage as shown in Figure 4.12 (a).

The master station controls the output voltage of I/O board interfaced to the slave station as follows:

It sends the analog I/O packet to the addressed slave station. The slave station will receive the packet and process it, and then check if the packet was received successful or not. If the packet was received successfully, then it first checks if the analog I/O control is for analog input or analog output. If for analog output, then the output voltage to be written is checked and the channel to which to output this voltage. Digital to Analog Conversion (DAC) is then performed and those values written to the corresponding analog channel. Figure 4.9 (a) above shows the Analog I/O control flow chart.

4.5 Engineering Units Conversion

Scaling to engineering units makes incoming analog signals easier to interpret. Three types of engineering units were implemented in this project:

1. Temperature in degrees Celsius
2. Pressure in bar
3. Flowrate in litres per second.

It is, obviously, possible to include others as required.

The analog values (or voltage) were converted to engineering units as follows:

Temperature conversion

Temperature can be used for controlling temperature measurements, such as room temperature, tank temperatures, etc. For Temperature conversion, the following formula was used:

Temperature (T) = T_f (V - V_i) / V_f

Where T_f is the maximum temperature in degree Celsius, V_f is the maximum voltage (or analog value), V is the voltage to be converted to temperature and V_i is the cut-off voltage (or analog value).

For example, assuming that the maximum temperature we can read is 100 degrees Celsius, cut-off voltage is 1.5 V, and maximum voltage to be read on the I/O board is 5 V, then the Temperature value (T) will be 40 degree Celsius at a voltage of 3.5 V.

Pressure conversion

Pressure control is critical in many applications,i.e steam vessels, water pipelines, etc.

For Pressure conversion, we use the following formula:

Pressure (P) = P_f (V - V_i) / V_f

Where P_f is the maximum temperature in degree Celsius, V_f is the maximum voltage (or analog value), V is the voltage to be converted to pressure and V_i is the cut-off voltage (or analog value).

Flowrate Conversion

Flowrates are used for controlling and measuring flows of practically all fluids. e.g. dairy products, wine, beer, soft drinks, etc.

For Flow conversion, we use the following formula:

Flowmeter (F) = F_f (V - V_i) / V_f

Where F_f is the maximum Flowmeter in degree Celsius, V_f is the maximum voltage (or analog value), V is the voltage to be converted to flowrate and V_i is the cut-off voltage (or analog value).

4.6 Summary

The design and the implementation of IrLAP protocol (master-slave protocol) and implementation of Delphi 7 to access an I/O Board were discussed in this chapter. This includes the system design, error control and Engineering Units Conversion.

Chapter 5

Evaluations and Results

5.1 Evaluations

After, but also during the implementation of system, different aspects i.e. reliability of systems, performance and also general function under more realistic circumstances tested and evaluated.

5.1.1 General Reliability of System

To test the reliability of the system as a whole, both the master station and the slave stations were made to run non-stop for a long period. The aim being to check if no error will occur.

After running for several hours the master station was still running without a flaw, however, the slave stations had crashed. When checking the log, it was found that the machines had run out of memory and froze. The master station, not receiving any response from the slave station, continued to send data and waited for a response until the timer reached the disconnect time. When both the master station and the slave station are unable to communicate for the duration of the link disconnect/threshold time, both close the links simultaneously. The problem was traced to a CPU speed problem, as the PC's utilised initially were of a fairly old generation. Using more state of the art hardware solved the problem.

5.1.2 Testing and Measurements

5.1.2 (a) Testing and Controlling of Digital I/O

Relay Switch

A relay is an electrically operated switch and the coil current can be ON or OFF. Relays have two switch positions and they are normally double throw (changeover) switches. Relays allow one circuit to switch a second circuit, which can be completely separate from the first. For example a low voltage battery circuit can use a relay to switch a 230V AC mains circuit.

The relay switch circuit shown in Figure 5.1(a) below was constructed for the purpose of testing and controlling digital input/output, and it was connected to the parallel digital I/O port as shown below in Figure 5.1(b) so that one end of the circuit goes to the datapin (for controlling the relay switch) and the other one goes to the ground pin.

One circuit was connected to each datapin. In this way eight software controllable relay switches were connected to the port as shown in Figure 5.1 (b). Figure 5.1(c) below shows the pin out of the parallel digital I/O port. The full description of parallel digital I/O port is shown in Appendix B (DB37)

The relay switch circuits were controlled as follows:

When the user sends out a 1 to the datapin where the relay switch is connected, the relay switch will go ON. When user sends 0 to the same datapin, the relay switch will go OFF.

The following programs are an examples of how to control and monitor the parallel digital I/O port datapins from the software.

1. Writing to the digital input/output port

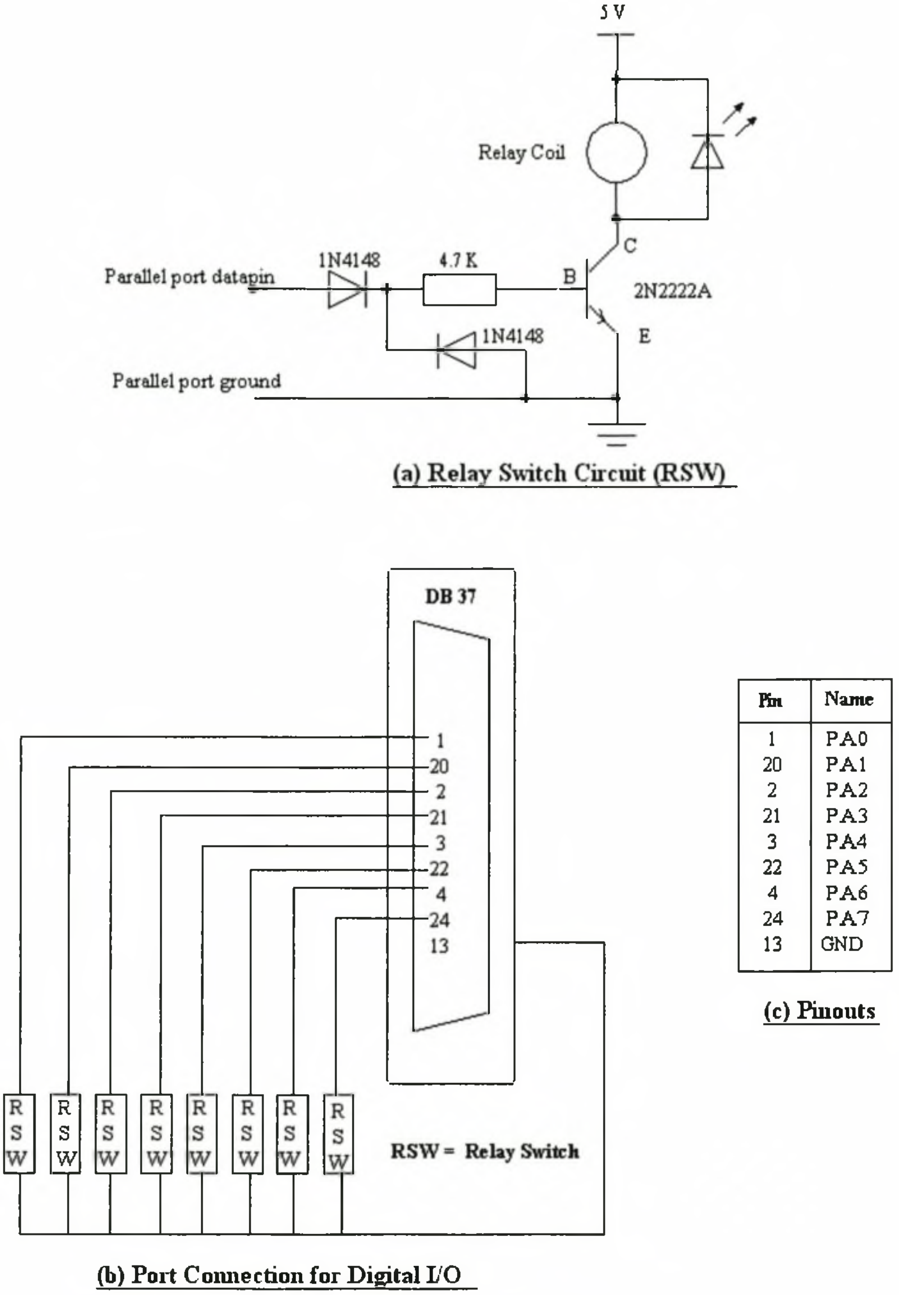
   ```
   procedure TForm1.btnDigitWriteClick(Sender: TObject);
   var
      dioPort, dioLine : Integer;
   begin
      EDREDioX.Write(dioPort, dioLine);
   end;
   ```

   The program above takes the dioPort (digital input/output port) to write to and dioLine (digital input/output line) on that port the user wants to write to and loops through every 300ms, writing the bit value as specified to the specific port and line.

2. Reading from the digital input/output port

   ```
   procedure TForm1.btnDigitReadClick(Sender: TObject);
   var
      dioPort : Integer;
   begin
      EDREDioX.Read(dioPort);
   end;
   ```

   The program above takes the dioPort (digital input/output port) to read to and then loops through every 300ms reading the specific port, to indicate whether the lines in this port are high or low.



(a) Relay Switch Circuit (RSW)

Pin	Name
1	PA0
20	PA1
2	PA2
21	PA3
3	PA4
22	PA5
4	PA6
24	PA7
13	GND

(c) Pinouts

(b) Port Connection for Digital I/O

Figure 5.1: Digital I/O Connection

5.1.2 (b) Measuring and Controlling of Analog I/O

Voltage Regulator Circuit

A voltage regulator circuit shown in Figure 5.2 below was constructed for the purpose of controlling and measuring the analog inputs/output, and connected to the parallel analog I/O port as follows:

The input voltage (l2V) was connected on the input of voltage regulator (Vin). The analog output pin (DAC0) of parallel analog inputs/outputs port was connected on the adjustment (1), to control the output voltage (or to give adjustable output voltage range). The analog input channel (for example channel 0) and the LEDs were connected on the output of voltage regulator (Vout). The analog input channel was connected so that the user can be able to measure (or read) the controlled output voltage, while LEDs were connected as a load so that user can be able to control their brightness. The brightness of LEDs varies depending on the voltage. The full description of parallel analog I/O port is shown in Appendix B (DB25).

The following program is an example of how to control and measure the parallel analog I/O port datapins from the software.

1. Writing to the analog input/output port

```
procedure TForm1.btnAnalogWriteClick(Sender: TObject);
var
   OutputChannel : Integer;
   OutputVoltage : Double;
begin
   EDREDaX.SingleWrite ((OutputVoltage*1000000), OutputChannel);
end;
```

The program above takes an analog output voltage (OutputVoltage) to be written and the analog output channel (OutputChannel) to which to output this voltage. The output voltage will be converted to μV from V by multiply it by 10^6 and will then be written out to the DA channel.

2. Reading from the analog input/output port

```
procedure TForm1.btn_RdAnalogClick(Sender: TObject);
var
  uVolt, InputVolt, InputGain, InputRange, InputChannel : Integer;
begin
  EDREAdX.Gain := InputGain;
  EDREAdX.Range := InputRange;
  uVolt := EDREAdX.SingleRead(InputChannel);
  InputVolt :=   (uVolt / 1000000);
end;
```

The program above takes the gain (InputGain), input voltage range (InputRange) to use for sampling and an input channel (InputChannel) from which to sample (or read) to. The input voltage (InputVolt) will be converted from μV to V by dividing it by 10^6.

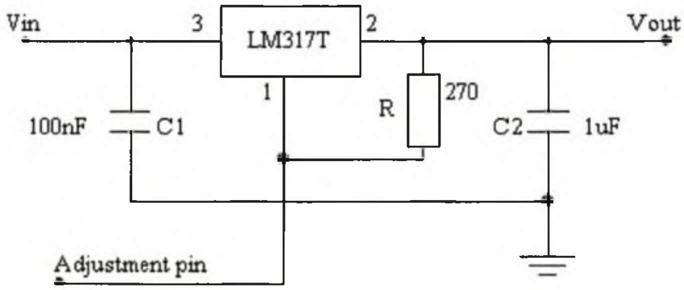


Figure 5.2: Voltage Regulator Circuit

Temperature Sensor

Temperature sensing is one of the most important data collection functions, which may be used for a wide variety of applications, including oven controllers, fan control and remote temperature sensing.

A temperature sensor circuit shown in Figure 5.3 was constructed for the purpose of measuring the analog inputs (Temperature), and connected to the parallel analog I/O port. The circuit uses an LM35 temperature sensor to develop 4 - 20mA current. The LM35 centigrade temperature sensor outputs a voltage that is linearly proportional to the ambient temperature. Every degree Celsius is represented by 10mV. For example, a temperature of 23 degree Celsius would read 0.23V or 230mV.

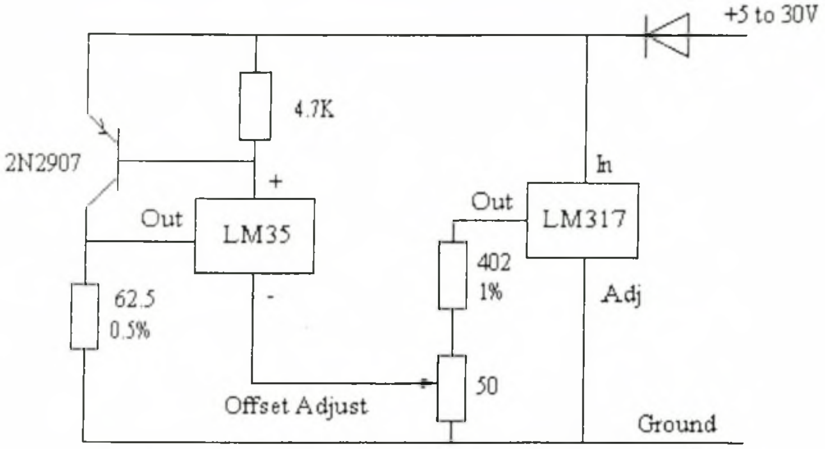


Figure 5.3: Temperature Sensor Circuit (4 - 20mA)

The circuit shown in Figure 5.3 was connected to the parallel analog I/O port as follows: The analog input channel (for example channel 0) was connected on the output pin (Out) of LM35. The analog input channel was connected so that user can be able to measure (or read) the output voltage.

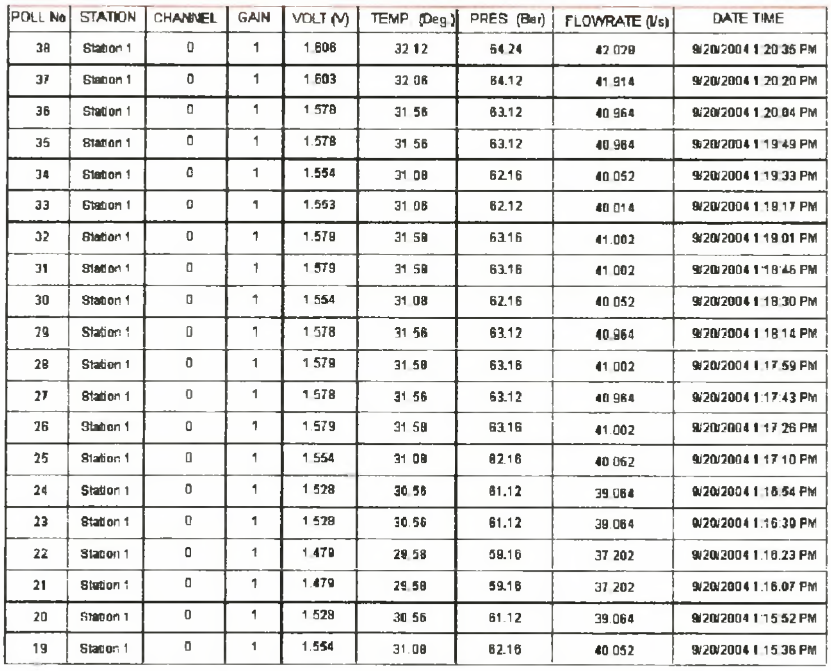
5.2 Results

The results shown below were taken during testing the different functions of the system.

5.2.1 Testing the Engineering Units Conversion

For the purpose of simulations, the output voltage measured from the voltage regulator circuit was converted to engineering units (i.e. pressure, temperature and flowrate) as explained in Chapter 4, Section 4.5. All different measurements were recorded in the database table as shown in Figure 5.4.

The records shown in Figure 5.4 can also be shown in a simple plot of analog variables (i.e. temperature, pressure and flowrate) versus date/time or represented in a graphical form as shown in Figure 5.5. This enables the user to analyse data.



POLL No	STATION	CHANNEL	GAIN	VOLT (V)	TEMP (Deg)	PRES (Bar)	FLOWRATE (l/s)	DATE TIME
38	Station 1	0	1	1.606	32.12	64.24	42.028	9/20/2004 1:20:35 PM
37	Station 1	0	1	1.603	32.06	64.12	41.914	9/20/2004 1:20:20 PM
36	Station 1	0	1	1.578	31.56	63.12	40.964	9/20/2004 1:20:04 PM
35	Station 1	0	1	1.578	31.56	63.12	40.964	9/20/2004 1:19:49 PM
34	Station 1	0	1	1.554	31.08	62.16	40.052	9/20/2004 1:19:33 PM
33	Station 1	0	1	1.553	31.06	62.12	40.014	9/20/2004 1:19:17 PM
32	Station 1	0	1	1.579	31.58	63.16	41.002	9/20/2004 1:19:01 PM
31	Station 1	0	1	1.579	31.58	63.16	41.002	9/20/2004 1:18:46 PM
30	Station 1	0	1	1.554	31.08	62.16	40.052	9/20/2004 1:18:30 PM
29	Station 1	0	1	1.578	31.56	63.12	40.964	9/20/2004 1:18:14 PM
28	Station 1	0	1	1.579	31.58	63.16	41.002	9/20/2004 1:17:59 PM
27	Station 1	0	1	1.578	31.56	63.12	40.964	9/20/2004 1:17:43 PM
26	Station 1	0	1	1.579	31.58	63.16	41.002	9/20/2004 1:17:26 PM
25	Station 1	0	1	1.554	31.08	62.16	40.062	9/20/2004 1:17:10 PM
24	Station 1	0	1	1.528	30.56	61.12	39.064	9/20/2004 1:16:54 PM
23	Station 1	0	1	1.528	30.56	61.12	39.064	9/20/2004 1:16:39 PM
22	Station 1	0	1	1.479	29.58	59.16	37.202	9/20/2004 1:16:23 PM
21	Station 1	0	1	1.479	29.58	59.16	37.202	9/20/2004 1:16:07 PM
20	Station 1	0	1	1.528	30.56	61.12	39.064	9/20/2004 1:15:52 PM
19	Station 1	0	1	1.554	31.08	62.16	40.052	9/20/2004 1:15:36 PM

Figure 5.4: Engineering Conversion Record

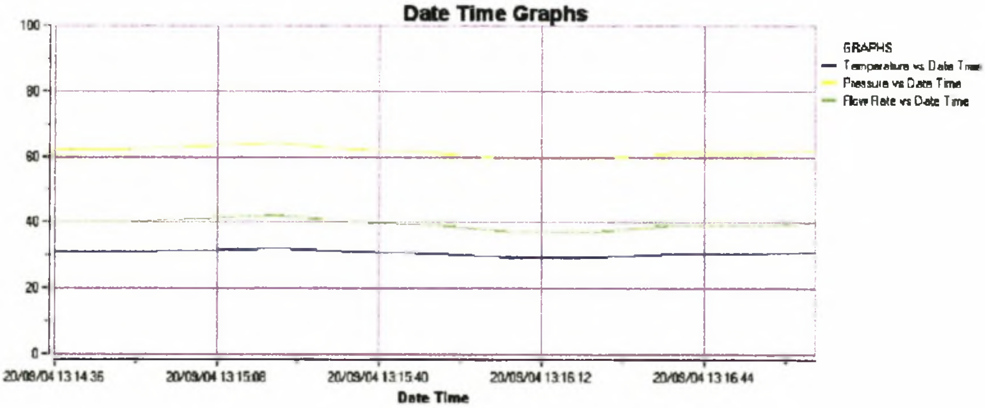


Figure 5.5: Different Engineering Conversions versus Date Time

The first plot in Figure 5.5 shows the temperature against date/time. By plotting temperature against date/time, the user would be able to see the consistency of the temperature for a specific date and time. Practically, assume that the user was measuring and controlling the tank temperature in an industry application, and afterwards, the user could plot the measurements to be able to see the constancy of the tank temperature at a particular date and time. The system was designed in such away that the user could select a date/time over which he wishes to analyse data (or to see the consistency of the tank temperature). Zooming in or zooming out is also supported.

The second plot shows pressure against date/time, and the user could see how the pressure varies with reference to date/time. The same applies to the third plot of flow rate versus date/time.

Figure 5.6 shows the plotting of the analog variables (i.e. temperature, pressure and flowrate) versus the voltage. This could enable the user to analyse what voltage is needed to output for the analog variables. Figure 5.6 shows that the analog variables (i.e. temperature, pressure and flowrate) are directly proportional to voltage.

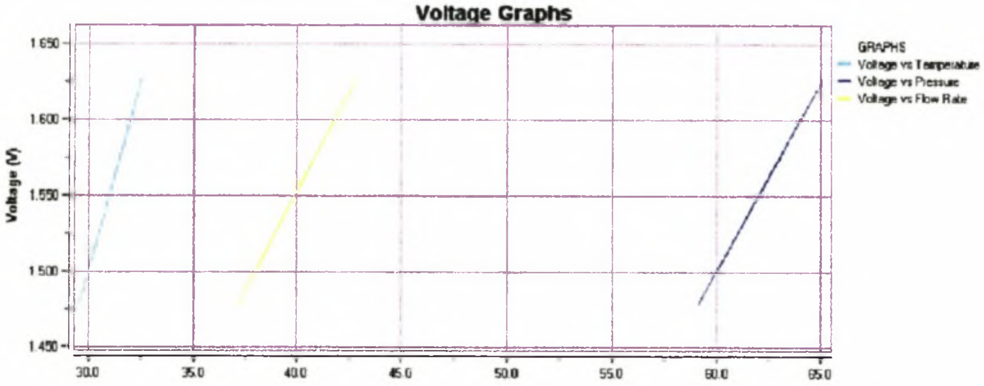


Figure 5.6: Analog Variables versus Voltage

5.2.2 Realistic Testing of Engineering Units and Results

The temperature sensor circuit was used for practical testing of the system. The temperature sensor circuit was placed in the DSP Lab (Department of Electrical and Electronics Engineering) at University of Stellenbosch on the 20 September 2004 for the purpose of monitoring the temperature of the laboratory, and all data was stored in the database. After the monitoring processes, the data was analysed by plotting of the graphs as shown in Figure 5.7 and 5.8.

Figure 5.7 shows the plotting of voltage in (mV) versus temperature in (degree Celsius). The plot shows that voltage is linearly proportional to the ambient temperature.

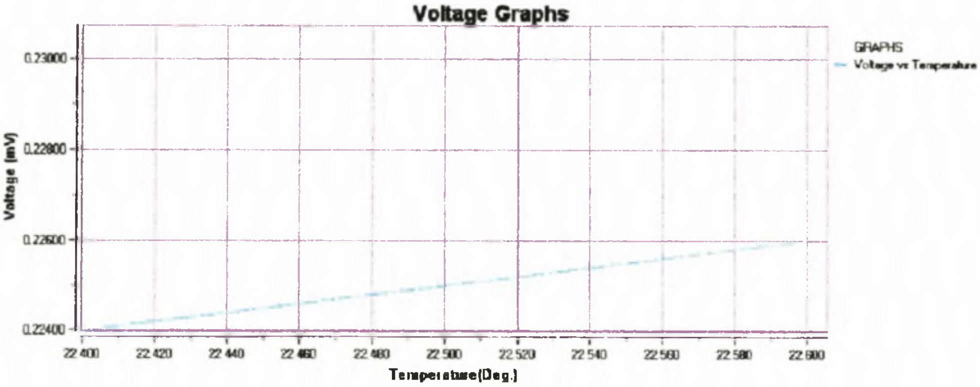


Figure 5.7: Plot of Voltage versus Temperature

The plot shown in Figures 5.8, 5.9, and 5.10 shows the temperature in degree Celsius versus date/time. The plot shows that the temperature was around 22 degree Celsius, but it was somewhat inconsistent. This is clearly seen by zooming in, as shown in Figures 5.9 and 5.10, that the temperature was inconsistent. By plotting temperature against date/time, it enables the user to control temperature easily, because when the user sees that the temperature is above the normal set temperature, the fun speed would be increased,and vice versa.

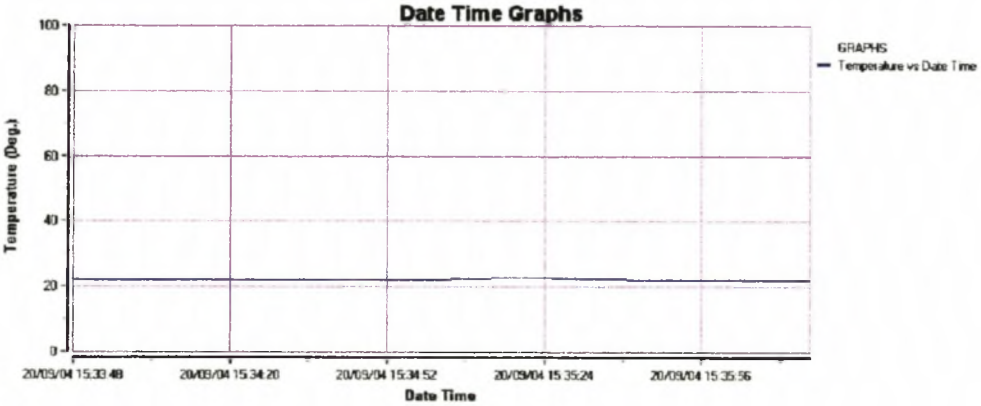


Figure 5.8: Temperature versus Date Time

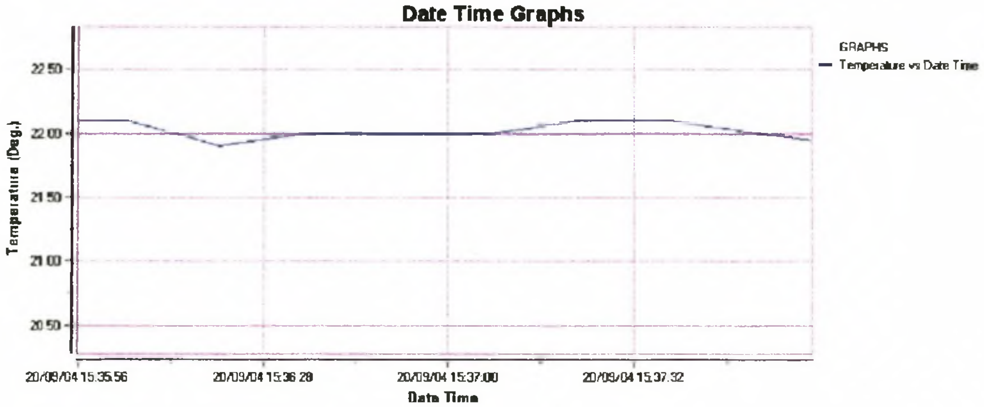


Figure 5.9: Temperature versus Date Time (Zooming in)

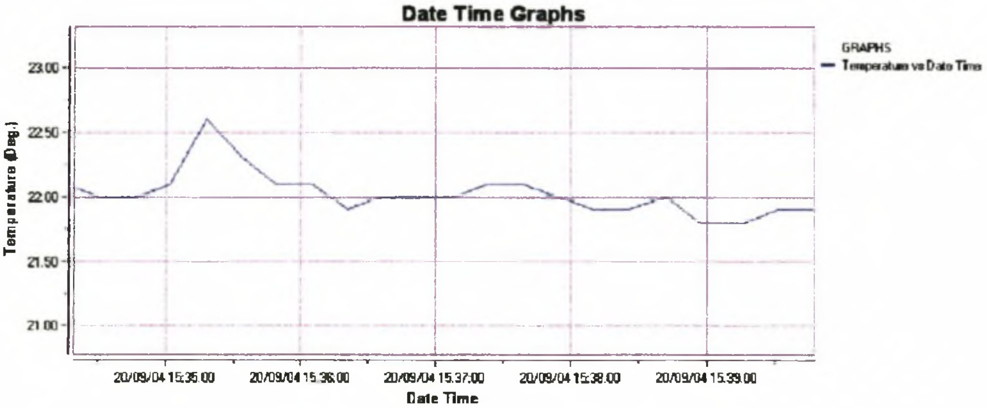


Figure 5.10: Temperature versus Date Time (Zooming in)

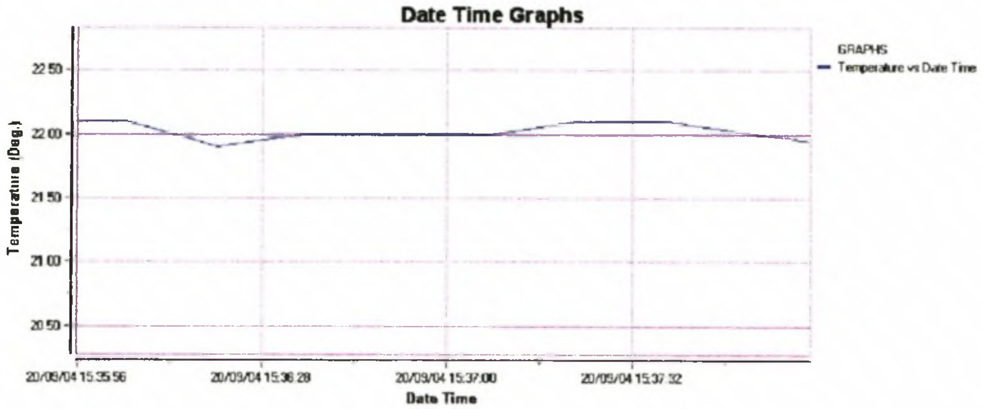


Figure 5.9: Temperature versus Date Time (Zooming in)

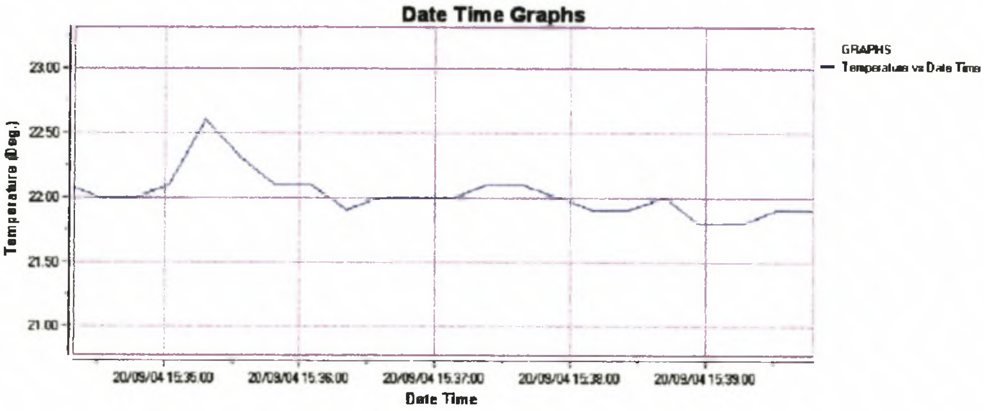


Figure 5.10: Temperature versus Date Time (Zooming in)

5.3 Summary

The general reliability of system, performance, testing and evaluation of general functions of system and the results found were discussed in this chapter.

Chapter 6

Application Layer Implementation

6.1 Application Layer

The application layer is closest to the end user, which means that the user interacts directly with the software application. The application layer includes network software that directly serves the user, providing such things as the graphic user interface, database interface and application features. The Application layer is usually made available by using an Application Programmer Interface (API). This layer interacts with software applications that implement a communicating component.

6.1.1 Graphic User Interface (GUI)

6.1.1 (a) Master Station

The design is based on a master-slave principle where the parties have different responsibilities.

The role of the master station is as follows:

- The master station sends command frames initiating the connections and transfers and,
- It is also responsible for organizing and controlling data flows.

- It oversees system configuration, data organisation and user interfacing on the application layer

Description of Master Station Components

Station addresses

The master station consists of up to 16 different station addresses (i.e. Slave station 1 up to slave station 16). This enables the user to select the slave station, which he/she wants to monitor and control. This could be obviously, be expanded if is required.

Control Settings

Three different control settings were constructed on the master station. This enables the user to select the controls he/she wants to perform. The control settings are as follows: The user can select whether he/she wants to read or write analog inputs/outputs, or digital input/output, or to just confirm the availability of slave stations.

Polling controls

The master station was constructed with two different polling controls namely: Manual and Automatic. This enables the user to select whether he/she wants to monitor or control the slave station manually or automatically. Manual operating strategies would require the dedicated and undivided attention of knowledgeable person to continually monitor and control field equipment, whereas automated systems would require the control strategies programmed into the process control system.

Port Settings

Port settings on the master station enables the user to select the port to which the device (IrDA adaptor) is connected. It is essential to open the port, so that the station can be able to transmit and receive data.

Digital I/O Controls

The master station was constructed with two digital I/O controls (write and read), three 8-bit wide digital I/O ports (Port 0, Port 1 and Port 2) and eight digital switches (SW1 to SW8), which corresponds to an 8-bit pattern as shown in Figure 6.1.

Write control (Digital Output)

The user has to specify, which digital I/O port to write to and the bit pattern that should be written. The bit pattern can be selected by switching the digital switches to either "ON" or "OFF". For example, if the user wants to switch SW1 "ON", then the corresponding bit pattern will be "1" and when is "OFF" bit pattern will be "0". By pressing the write button on the master station, the command will immediately be sent to the directed slave station and the corresponding switches will be set to either high or low depending on the bit pattern.

Read control (Digital Input)

The user has to specify the digital I/O port to read to. By pressing the read button on the master station, the command will immediately be sent to the directed slave station and read the specified port. The slave station will then send back the state of the digital input/output port to master station.

Analog I/O Controls

The master station was constructed with two analog I/O controls (write and read), four analog output channel (DAC0 to DAC3), sixteen analog input channels (Channel 0 to Channel 15), three gain levels (0.25, 0.5 and 1), two voltage range levels (Single mode and Differential mode) and four engineering units conversion utilities (Voltage, Temperature, Pressure and Flow rate) as shown in Figure 6.1.

Engineering units conversion

Engineering units allows the user to select either the output voltage or pressure or flow rate or temperature he/she wants to write (control). For example, if the user wants to control temperature, the selected value of temperature will be converted to voltage. The voltage will be used to control the speed of the funs or motors.

Writing control (Analog output)

The user has to specify the analog output channel to write to and either the output voltage (or engineering units value in volt) to write. The output voltage could be selected by adjusting the sliding bar. By pressing the write button on the master station, the command will immediately be sent to the directed slave station and the slave station will output the exact voltage on the on the particular channel.

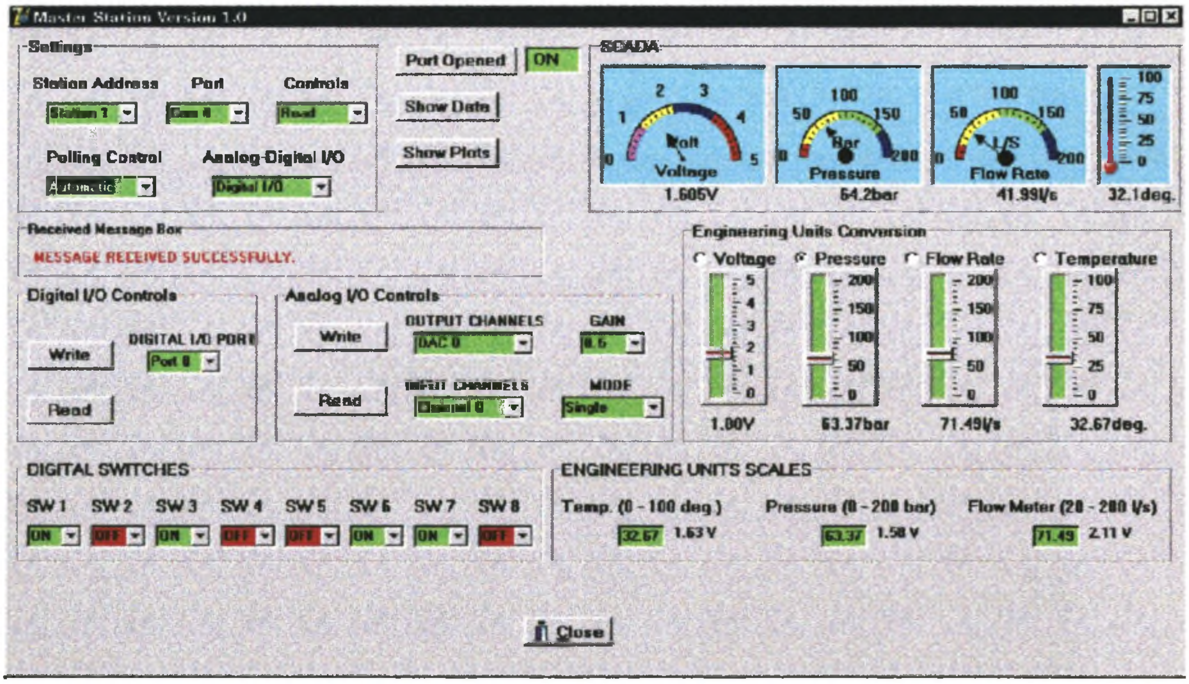
Reading control (Analog input)

The user has to select the analog input channel to read to, the gain and the input voltage range to use for sampling. By pressing the read button on the master station, the command will immediately be sent to the directed slave station and read the particular input channel. The slave station will then send back the value read to master station.

SCADA

The small SCADA shown in Figure 6.1 is designed to allow the user to view the up to date analog inputs (i.e. temperature, voltage, pressure and flow rate) from slave stations.

Figure 6.1 below shows the graphical representation of the master station.



Station Address: Station 1
Port: Com 0
Controls: Read
Polling Control: Automatic
Analog-Digital I/O: Digital I/O
Port Opened: ON

MESSAGE RECEIVED SUCCESSFULLY.

DIGITAL I/O PORT: Port 0
OUTPUT CHANNELS: DAC 0
GAIN: 0.5
INPUT CHANNELS: Channel 0
MODE: Single

SW 1	SW 2	SW 3	SW 4	SW 5	SW 6	SW 7	SW 8
ON	OFF	ON	OFF	OFF	ON	ON	OFF

1.605V 64.2bar 41.99l/s 32.1deg.

1.00V 63.37bar 71.49l/s 32.67deg.

Temp. (0 - 100 deg.)	Pressure (0 - 200 bar)	Flow Meter (20 - 200 l/s)
32.67 1.63 V	63.37 1.58 V	71.49 2.11 V

Figure 6.1: The Graphical Representation of Master Station

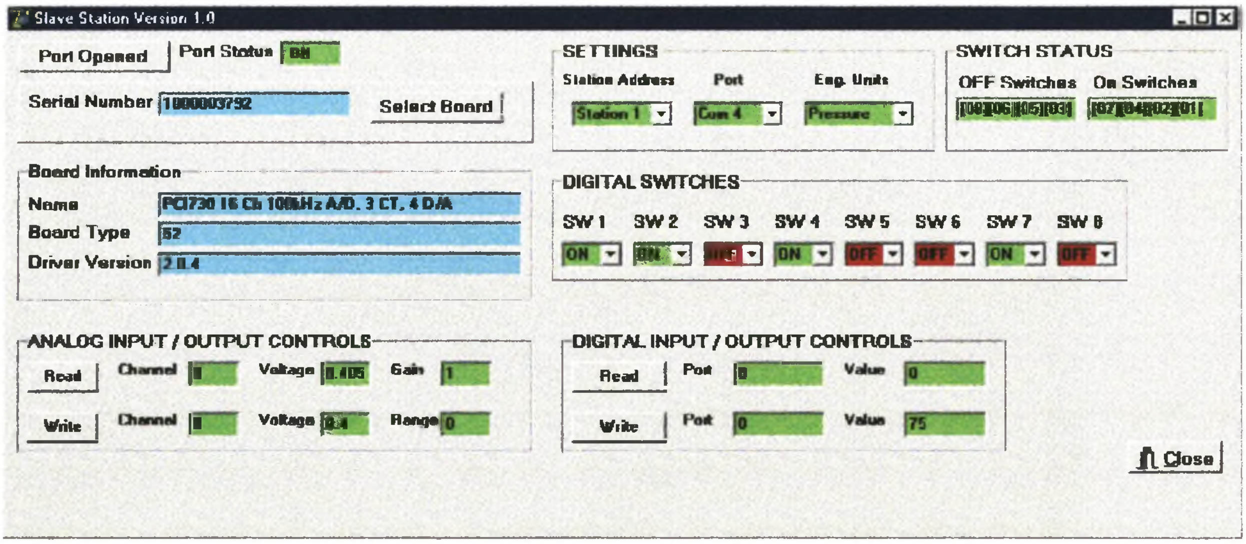
6.1.1 (b) Slave Station

A slave station only sends the response frames when the master station requests it.
The port status on the slave stations shown in Figure 6.2 shows whether the port is opened or closed, if the port (to which the device is connected) is opened, the port status would be ON, while if is closed, the port status would be OFF. It is essential to open the port,

so that the station can be able to transmit and receive data.

The slave station is constructed in such a way that the user could be able to select serial number of the I/O board embedded on the slave station. When the I/O board is selected, the board information (i.e. Board Name, Board Type and Board Version) will be displayed as shown in Figure 6.2.

It is also constructed in such away that the user could also change the station address. For example, the user could be able to change the slave station address 1 to slave station address 8. By doing that, when master station wants send the message to slave station 1, slave station 8 would receive the message instead of slave station 1.
It is also constructed with analog and digital I/O controls. This enables the user to read from or write to the I/O board, when he/she is in the slave station.
Eight state switches (SW1- SW8) are constructed on slave station to indicate whether the lines on the I/O port are high or low. Figure 6.2 shows the graphical representation of the slave station.



Port Status: ON
Serial Number: 1000003792

Station Address: Station 1
Port: Com 4
Eng. Units: Pressure

OFF Switches: [08][06][05][03]
On Switches: [07][04][02][01]

Name: PCI730 16 Ch 100kHz A/D, 3 CT, 4 D/A
Board Type: 52
Driver Version: 2.0.4

SW 1: ON
SW 2: ON
SW 3: OFF
SW 4: ON
SW 5: OFF
SW 6: OFF
SW 7: ON
SW 8: OFF

Read — Channel: 0, Voltage: 0.405, Gain: 1
Write — Channel: 0, Voltage: 2.4, Range: 0

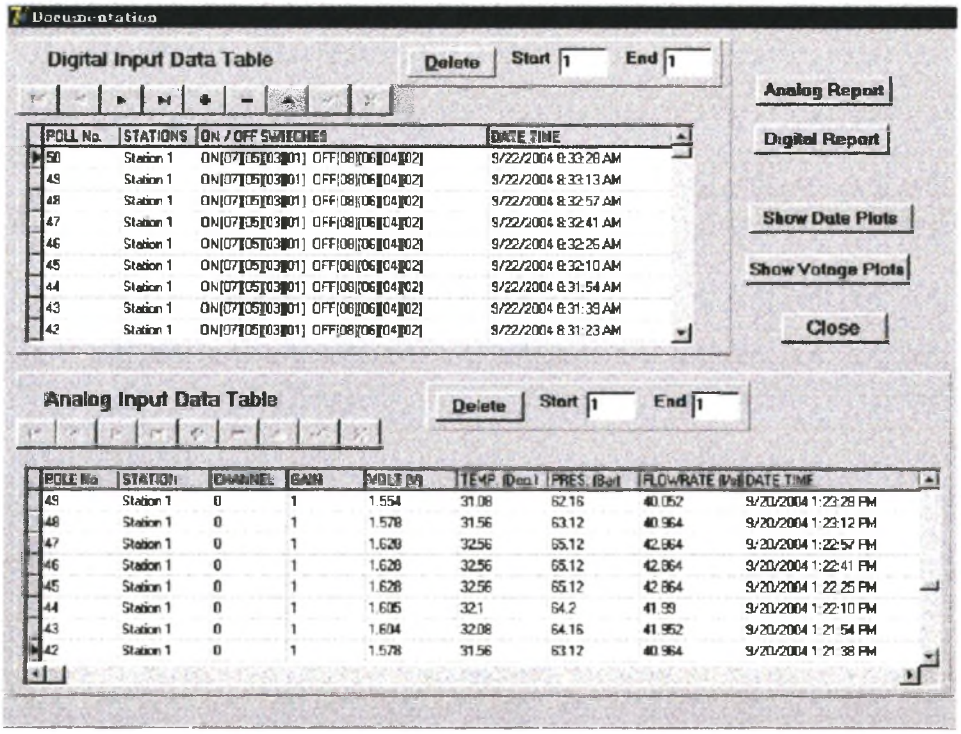
Read — Port: 0, Value: 0
Write — Port: 0, Value: 75

Figure 6.2: The Graphical Representation of Slave Station

6.1.2 Database Interface

Database is used for storing data for later analyses.

The master station was designed with two databases tables for recording all the measurements. By pressing the Show Data button on the master station, it will take the user to the database tables as shown in Figure 6.3. The first database is for recording the measured digital inputs and the second one for recording the measured analog inputs.



Digital Input Data Table

POLL No.	STATIONS	ON / OFF SWITCHES	DATE TIME
50	Station 1	ON[07][05][03][01] OFF[08][06][04][02]	9/22/2004 8:33:28 AM
49	Station 1	ON[07][05][03][01] OFF[08][06][04][02]	9/22/2004 8:33:13 AM
48	Station 1	ON[07][05][03][01] OFF[08][06][04][02]	9/22/2004 8:32:57 AM
47	Station 1	ON[07][05][03][01] OFF[08][06][04][02]	9/22/2004 8:32:41 AM
46	Station 1	ON[07][05][03][01] OFF[08][06][04][02]	9/22/2004 8:32:26 AM
45	Station 1	ON[07][05][03][01] OFF[08][06][04][02]	9/22/2004 8:32:10 AM
44	Station 1	ON[07][05][03][01] OFF[08][06][04][02]	9/22/2004 8:31:54 AM
43	Station 1	ON[07][05][03][01] OFF[08][06][04][02]	9/22/2004 8:31:39 AM
42	Station 1	ON[07][05][03][01] OFF[08][06][04][02]	9/22/2004 8:31:23 AM

Analog Input Data Table

POLE No	STATION	CHANNEL	GAIN	VOLT (V)	TEMP. (Deg.)	PRES. (Bar)	FLOWRATE (l/s)	DATE TIME
49	Station 1	0	1	1.554	31.08	62.16	40.052	9/20/2004 1:23:28 PM
48	Station 1	0	1	1.578	31.56	63.12	40.964	9/20/2004 1:23:12 PM
47	Station 1	0	1	1.628	32.56	65.12	42.864	9/20/2004 1:22:57 PM
46	Station 1	0	1	1.628	32.56	65.12	42.864	9/20/2004 1:22:41 PM
45	Station 1	0	1	1.628	32.56	65.12	42.864	9/20/2004 1:22:25 PM
44	Station 1	0	1	1.605	32.1	64.2	41.99	9/20/2004 1:22:10 PM
43	Station 1	0	1	1.604	32.08	64.16	41.952	9/20/2004 1:21:54 PM
42	Station 1	0	1	1.578	31.56	63.12	40.964	9/20/2004 1:21:38 PM

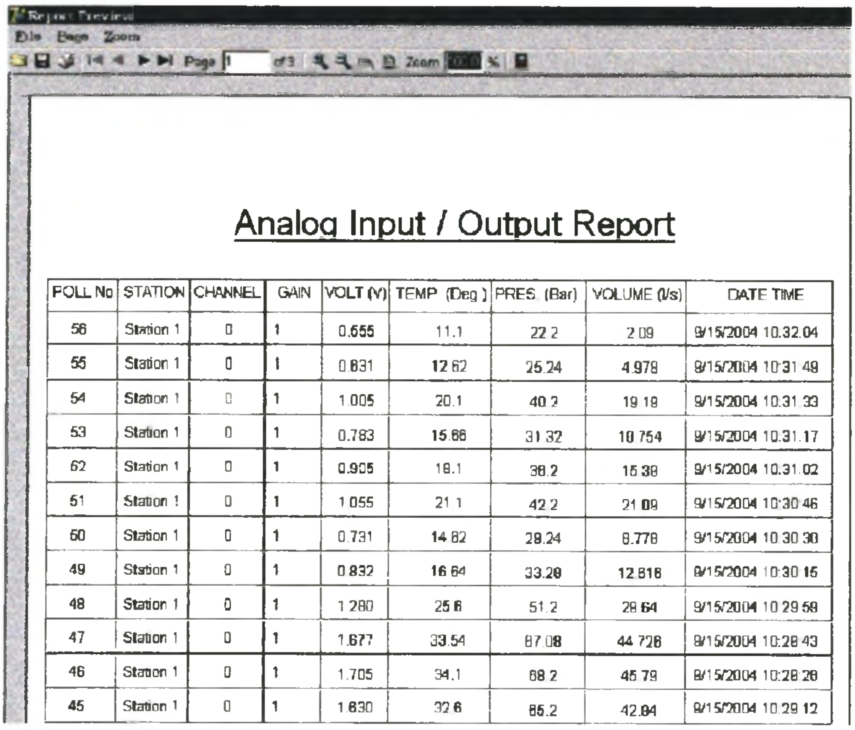
Figure 6.3: Master Station Database Tables.

All database tables were designed in such a way that the user could be able to generate the report and be able to compare all the records by plotting graphs.

By pressing the Analog Report button on the database tables shown in Figure 6.3, it will generate the report as shown in Figure 6.4.

Figure 6.4 shows an example of report generated from analog input database table. On the report generated the user can be able to save and print.

Figure 6.5 shows an example of plotting of different analog inputs from analog input database table.



Analog Input / Output Report

POLL No	STATION	CHANNEL	GAIN	VOLT (V)	TEMP (Deg)	PRES (Bar)	VOLUME (l/s)	DATE TIME
56	Station 1	0	1	0.655	11.1	22.2	2.09	9/15/2004 10.32.04
55	Station 1	0	1	0.631	12.62	25.24	4.978	9/15/2004 10:31:49
54	Station 1	0	1	1.005	20.1	40.2	19.18	9/15/2004 10:31:33
53	Station 1	0	1	0.783	15.66	31.32	10.754	9/15/2004 10.31.17
62	Station 1	0	1	0.905	18.1	38.2	15.38	9/15/2004 10.31.02
51	Station 1	0	1	1.055	21.1	42.2	21.09	9/15/2004 10:30:46
50	Station 1	0	1	0.731	14.62	28.24	8.778	9/15/2004 10:30:30
49	Station 1	0	1	0.832	16.64	33.28	12.816	9/15/2004 10:30:15
48	Station 1	0	1	1.280	25.6	51.2	28.64	9/15/2004 10:29:59
47	Station 1	0	1	1.677	33.54	67.08	44.726	9/15/2004 10:28:43
46	Station 1	0	1	1.705	34.1	68.2	45.79	9/15/2004 10:28:28
45	Station 1	0	1	1.630	32.6	65.2	42.84	9/15/2004 10:29:12

Figure 6.4: Report generated from the Analog Input Database Table.

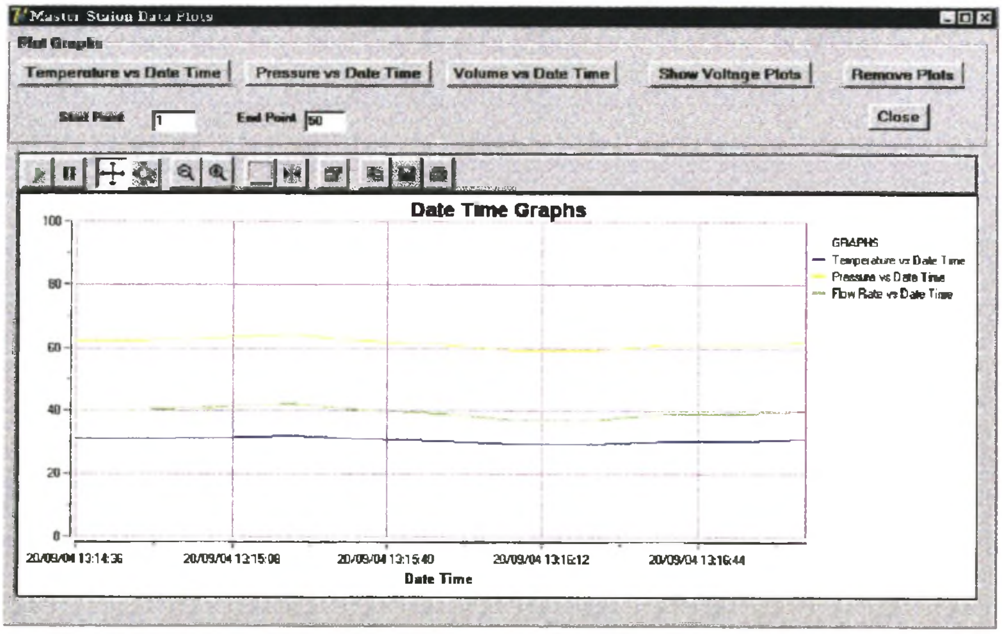


Figure 6.5: Measured Analog Inputs.

6.2 Summary

The chapter discusses the Graphic User Interface (GUI), which includes responsibilities of both the master station and the slave station. It also discussed Database Interface.

Chapter 7

Conclusions and Recommendations

7.1 Conclusions and Discussions

Many different aspects should be taken into account when choosing wireless transmission technologies. The user has to be careful, when selecting transmission technologies, whether he is mainly interested in price, transmission power and range, data rate or security.

In this report, different selection criteria were discussed and the most appropriate technology (Infrared LAN's) selected because is of its price, security and the specific application. If the main interest was range and data rate, Spread Spectrum LAN's would have been considered as the best.

The purpose of the project was the development of a short-range, real time, Wireless Local Area Network for instrumentation purposes, interconnecting various monitoring and control transducers to a central master station.

Further to the initial investigations and after selection of an appropriate communications technology, this project then set off to develop a full IrDA embedded master-slave protocol, for instrumentation and process control applications.

This facility was then used as a platform for the further development of application layer software, enabling overall system configuration, convenient data extraction and management, as well as fully realisable user interaction and practical applicability.

The benefits obtainable by having such a system available in process control and industrial instrumentation installations are many and well known. Potential problems can be identified at an early stage and interactive control actions are much simplified, resulting in increased plant efficiency.

The reduction in plant wiring due to the use of IR communication is obviously, a clear and great advantage. It does not only offer a significant cost reduction, but also reduces restrictions on placement of monitoring and control equipment. These are, in such an implementation. not bound by available cable routes and related infrastructure.

The developed solution similarly offers a reduction in maintenance cost, as it is a known fact that many operational problems are due to varying interconnection and connector integrity. This solution eliminates a good percentage of such potential defects.

It can be said in conclusion, that the project was a success and the analysis showed that the implemented overall design is feasible and presents a cost effective situation. A functional implementation of the overall network will assist in the management of systems, as intended for the application.

7.2 Recommendations

The success of the implementation as set out, suggests some further development, such as the following:

1. Implementation of a basic front end user web based interface for report generation and basic system settings. This will reduce cost of deployment and maintenance in the automation control systems and will also enable convenient remote access.
2. Investigate the possibility and the potential benefits of integrating Internet technologies into the conventional process control structure.
3. Investigate methods of improving and optimising the polling process.
4. Development of a more complete Supervisory Control And Data Acquisition (SCADA) system to monitor plant process and minimize human intervention. This should minimise all encompassing as existing full house SCADA packages, but still fully suitable for this type of application, with all facilities normally required.

Appendix A

Optional IrDA Protocol

There are several protocols above the Link Management Protocol. Some of these are more important than others. The most important of these optional protocols are the Tiny TP, IrCOMM, and IrOBEX, which are used in most communication[9].

Some mobile devices need a specification on how to communicate with each other. Digital Image captures devices/cameras and mobile telephony and communication devices are such examples. For these there are separate protocols: IrTran-P for image exchange and IrMC for information exchange in telephony and communication devices.

Smaller mobile devices with limited bandwidth and capacity may need a specialized set of IrDA instructions. For this there is a protocol called IrDA Lite. It provides methods of reducing the size of IrDA code while maintaining compatibility with full implementations.

Tiny TP (TTP)

TinyTP is an optional IrDA layer, although it is so important that it should generally be considered a required layer (except in the case of current printing solutions). TinyTP provides two functions:

- Flow control on a per-LMP-connection (per-channel) basis
- Segmentation and reassembly (SAR)

Flow control

Per-channel flow control is currently the most important use of TTP. The IrLAP protocol does offer flow control but in the case of multiplexing, another flow control is needed in order to make it work efficiently. The flow control in TTP is a credit based scheme and it works as follows:

- At connection, some credit is extended by each side. One credit corresponds with permission to send one LMP packet. If one side sends a credit, it must be able to accept a maximum sized packet. The number of credits one side can send depends entirely on how much buffer space that is available.
- Sending data causes credit to be used up (one unit of credit per packet sent).
- Periodically, the receiver issues more credit.
- If a sender has no credit, no data movement can occur, except a credit-only packet, which can always be sent; it is not a subject to flow control.

Although this description talks about the sender and receiver as if those roles were fixed, it is common for both sides of a LMP connection to send and receive.

Segmentation and Reassembly

Some devices with low capacity might not be able to accept full size packages. To solve this problem there is function called "segmentation and reassembly" (SAR) that chops large data packages into pieces, sends the pieces and then puts the data back together on the other side. The entire piece of data being chopped up and reconstituted is called an SDU, or Service Data Unit. The maximum SDU size is negotiated when the Tiny TP/LMP connection is first made[9].

IrCOMM

IrCOMM is a family of protocols that run on top of the mandatory IrDA protocol group. IrCOMM supports the emulation of a peer device connected using a serial or parallel cable. This emulation is from the perspective of applications that are accessing serial and parallel ports through the operating system API.

IrCOMM provides four service types or classes:

- 3-Wire Raw (parallel and serial emulation): sends data only, no non-data circuit information and hence no control channel. Runs directly on IrLMP
- 3-Wire (parallel and serial emulation): minimal use of control channel. Uses TinyTP
- 9-Wire (serial emulation only): uses control channel for status of standard RS-232 non-data circuits. Uses TinyTP
- Centronics (parallel emulation only): uses control channel for status of Centronics non-data circuits. Uses TinyTP

These services falls into two types, raw and cooked.

Three wired raw provides a data channel only, and utilizes the IrLAP flow control, while the cooked types supports a control channel and employs TinyTP flow control. In the cooked mode, the control signal (CTS and RTS for example) are encoded and transmitted serially as commands. The IrCOMM layer in the receiving devices decodes the commands and report them to the next higher layer[9].

IrOBEX

IrOBEX is an optional application layer protocol designed to enable systems of all sizes and types to exchange a wide variety of data and commands in a resource-sensitive standardized fashion. It addresses one of the most common applications on either PCs or embedded systems: take an arbitrary data object (a file, for instance), and send it to whomever the infrared device is pointing to. It also provides some tools to enable the object to be recognized and handled intelligently on the receiving side.

The potential range of objects is wide, encompassing not only traditional files, but also pages, phone messages, digital images, electronic business cards, database records, hand-held instrument results, or diagnostics and programming. The common thread is that the application doesn't need or want to get involved in managing connections or dealing with the communications process at all. Just take the object and ship it to the other side with the least fuss possible. It is very similar to the role that HTTP serves in the Internet protocol suite, although HTTP is very "pull" -oriented in its fundamental design, while OBEX is more evenly balanced[9].

OBEX was created to "package" an IrDA communications transaction as completely as possible and thereby dramatically simplify the development of communications-enabled applications. It was further designed to meet the following criteria:

- Simple: Supports most-needed operations/applications
- Compact: Under 1K code on small system
- Flexible: Supports data handling for both industry standard and custom types
- Works on IrDA, but is transport independent

Appendix B

Connector Pin Assignments of PCI730 Board

Pin	Name	Pin	Name
1	PA0	20	PA1
2	PA2	21	PA3
3	PA4	22	PA5
4	PA6	23	PA7
5	PB0	24	PB1
6	PB2	25	PB3
7	PB4	26	PB5
8	PB6	27	PB7
9	PC0	28	PC1
10	PC2	29	PC3
11	PC4	30	PC5
12	PC6	31	PC7
13	DGND	32	NOT USED
14	CLK0	33	NOT USED
15	COUT0	34	GATE0
16	GATE1	35	CLK1
17	CLK2	36	COUT1
18	COUT2	37	GATE2
19	+5V		

Table B.1: Pinouts for PCI730 (Internal Connector-DB25)

Pin	Name	Pin	Name
1	CHO	14	CH1
2	CH2	15	CH3
3	CH4	16	CH5
4	CH6	17	CH7
5	CH8	18	CH9
6	CH10	19	CH11
7	CH12	20	CH13
8	CH14	21	CH15
9	AGND	22	DAC0
10	DAC1	23	DAC2
11	DAC3	24	+VDD
12	-VDD	25	EXT_TRIGGER
13	NOT USED		

Table B.2: Pinouts for PCI730 (External Connector-DB25)

Signal	Description
CH0-15	Analog Inputs
DAC0-3	Analog Outputs
+VDD	+12V_Fused Output
-VDD	-12V_Fused Output
AGND	Analog Ground
CLK	Counter Timer External Clock Input
COUT	Counter Timer Output
GATE	Counter Timer External Gate Input
PA0-7	Digital Inputs/Outputs Port A
PB0-7	Digital Inputs/Outputs Port B
PC0-7	Digital Inputs/Outputs Port C
+5V	Power Output
DGND	Digital Ground

Table B.3: Signal Definitions

Appendix C

The Source Code

The accompanying disc contains the source code of the master and slave stations written in Delphi 7.

Some of the function used, which are written in Delphi 7 are as follows:

- Integer To Binary Conversion
- Binary To Characters Conversion
- Cyclic Redundancy Checking (CRC-16).
- Bit Stuffing and Destuffing

```
//********************* Converting Integer To Binary Function *****************
function IntToBin(Value: LongInt; Digits : Integer): String;
Var
   i : Integer;
Begin
   Result := '';
   For i := Digits downto 0 do
     if Value and (1 shl i) <> 0 then
        Result := Result + '1'
     else
        Result := Result + '0';
end;
```

```
//****************** End of Converting Integer To Binary Function **************

//******************* Converting Binary To Characters Function *****************

Function BinToChar(M : String):String;
var
   i,j : Integer;
begin
   J := 0;
   for i := 1 to Length(M) do
      begin
          if M[i] = '0' then
             begin
                 j := 2*j + 0;
             end
          else begin
              j := 2*j + 1
          end
      end;
   Result := IntToStr(j);
end;
//***************** End of Converting Binary To Characters Function *************

//*************** Bit Stuffing  and Bit Destuffing Function *********************

Function Bit_Stuffing(S : String): String;
begin
    Result := StringReplace(S,'11111','111110',[rfReplaceAll]);
end;

Function Bit_DeStuff(S :String) : String ;
begin
    Result := StringReplace(S,'111110','11111',[rfReplaceAll]);
end;
//************** End of Bit Stuffing and Bit Destuffing Function ****************
```

```
//*********************** Calculating CRC 16 Function ***************************
Function CRC16(Str: String) : Word;
 Const
     GenPoly = $18005;
 Var
     CRC     : Word;
     Index1, Index2  : Byte;
 begin
     CRC := 0;
     For Index1 := 1 to length(Str) do
          begin
              CRC := (CRC xor (ord(Str[Index1]) SHL 8));
              For Index2 := 1 to 8 do
                  if ((CRC and $8000) <> 0) then
                      CRC := ((CRC SHL 1) xor GenPoly)
             else
                 CRC := (CRC SHL 1)
          end;
      CRC16 := (CRC and $FFFF)
end;
//********************* End of Calculating CRC 16 Function **********************
```

Bibliography

[1] William Stallings, "Data and Computer Communications", Seventh Edition, Prentice Hall 2004.

[2] Andrew S. Tanonbaum, "Computer Network", Fourth Edition, Prentice Hall 2003.

[3] Kaven Pahlavan and Allen H. Levesque, "Wireless Information Networks", John Wiley & Sons, INC.

[4] Ira Brodsky, "Wireless: The Revolution in Personal Telecommunications", Mobile Communications Series, Artech House, INC, 1995

[5] Norihiko Morinaga, Ryuji Kohno and Seiichi Sapmpei, "Wireless Communication Technologies": New Multimedia Systems, The Kluwer International Series in Engineering and Computer Science.

[6] IrDA, "Infrared Data Association Serial Infrared Physical Layer Specification - version 1.4," Infrared Data Association, May 2001. http:www.irda.org/

[7] IrDA, "Infrared Data Association Serial Infrared Physical Layer Specification - version 1.2", Infrared Data Association, November 10, 1997.

[8] AMP Incorporated SystemSoft Corporation, "Universal Serial Bus IrDA Bridge Device Definition - Revision 1.0", March 23, 2000.

[9] Peter Barker, Anthony C. Boucouvalas. "Performance Modeling of the IrDA Protocol for Infrared Wireless Communications." IEEE Communications Magazine. December 1998.

[10] Patrick J. Megowan, David W.Suvak, Charles D Knutson, "IrDA Infrared Communications: An Overview", Counterpoint Systems Foundry, INC, http://www.web-ee.com/primers/files/irda.pdf/;

[11] IrDA, "Serial Infrared Link Access Protocol (IrLAP), Version 1.1", Apple Computer, Inc. Counterpoint Systems Foundry, Inc. June 16, 1996, http://www.irxon.com/download/IrLAP11.pdf

[12] Mike Rodbell, Communication System Design, "Infrared Wireless Link Access Protocol." http://www.commsdesign.com/main/9802art2.htm, February 1998.

[13] Michael G. Robertson, Scott V. Hansen, Franklin E. Sorenson, Charles D. Knutson, "Modeling IrDA Performance: The Effect of IrLAP Negotiation Parameters on Throughput", Dept. Computer Science, Brigham Young University, Provo, UT 84602, U.S.A, michaelr@cs.byu.edu

[14] J. Meel, "Spread Spectrum Introduction", De Nayer Institute, Sirius Publications, Belgium.

[15] Universal Serial Bus IrDA Bridge Device Definition, AMP Incorporated SystemSoft Corporation, Revision 1.0, March 23, 2000

[16] Marco Cantu, "Mastering Delphi 7", SYBEX, San Francisco, London.

[17] Guide to Wireless LAN Technologies, Intermec, Technologies Corporation, 1998, http://www.intermec.com

[18] Harinath Ganesan, "Wireless Local Area Networks: Spread Spectrum communication and its Security", Arizona State University East.

[19] Wireless networking Primer: Introduction to Network, http://www.pdamd.com/vertical/features/wireless_2.xml

[20] Agilent IrDA Data Link Design Guide, Agilent Technologies, Semiconductor Products Group, http://www.agilent.com/view/ir

[21] Kay Townes, "Wireless LANs Comparison and Selection Guide", September 2002