MODELLING AND REAL-TIME IMPLEMENTATION OF WIRELESS COMMUNICATION ON A TYPICAL INDUSTRIAL PROCESS

by

WILSON MABALANA NDLOVU

submitted in accordance with the requirements
for the degree of

MAGISTER TECHNOLOGIAE

in the subject

ELECTRICAL ENGINEERING

at the

UNIVERSITY OF SOUTH AFRICA

SUPERVISOR: Prof. MO OHANGA
CO-SUPERVISOR: Prof. P NAIDOO

June 2016

DECLARATION

I, Wilson Mabalana Ndlovu, student number 37943936, hereby declare that the dissertation entitled Modelling and Real-time implementation of Wireless Communication on a Typical Industrial Process is the result of my own research and presents my own work except where stated in reference text.

Name: ______________________

Signed: _____________________

Date: _______________________

ACKNOWLEDGEMENTS

I would like to express my gratitude to my supervisors Prof. MO Ohanga of Vaal University of Technology and Prof. P Naidoo of Mangosuthu University of Technology for their commitment in the research. I would like to thank them for their continued support, guidance and their invaluable input towards the completion of this research. My gratitude also goes to Z P Ngubane, my sincere student, who has been of great help from the day of commencement to completion of this research work. She has been adept in tasks that needed dexterity and she kept on giving support and encouragement even after she has completed her studies. I would also like to thank Warren Scott, the director of ABACUS Solutions for his contribution in the early stages of the research process and spending his time to assist with the establishment of the PROFIBUS PA section of the network. I would also like to thank Dhiren Naidoo from ABACUS Solutions for his contribution. I would also like to thank my ex-student Treasure Mshololo for his continued support. My sincere thanks to Mangosuthu University of Technology for providing the research facility and funding.

ABSTRACT

Communication amongst field devices, control unit and programming unit in industrial automation networks is essential for bulk production, but largely consists of wired networks that can sometimes be bulky and substantially lack mobility as at times there can arise a need for a field device to be moved either for maintenance purposes or for rearrangement. There was therefore a need for wireless communication and PROFIBUS networks that can provide the minimum movement to field devices or the programing computer. Although wireless communication technology has penetrated the commercial network, it is still inadequately utilised in industrial settings due to electromagnetic induction and other forms of interferences due to industrial machinery. This dissertation introduced wireless communication in a PROFIBUS network where the MPI section was replaced with the wireless link. The PROFIBUS network technology is a hybrid of protocols where the PROFIBUS DP employs RS485 technology with a transmission rate of 45 kbps and above while the PROFIBUS PA employs Manchester Encoded Bus Powered (MBP) technology at a fixed rate of 31.25 kbps. In RS485 technology, data is transmitted as a voltage difference between the two wires while in MBP data is transmitted as transitions in current signal and data and power are transmitted on the same conductors. The PROFIBUS data is also transmitted in the form of telegrams which further puts a strain on any form of intermediate processing and hence the need for high speed processing. In this research task the PROFIBUS PA level transmitter measures the pressure of the fluid in the Blend Chest and sends it to the PLC. The level transmitter was installed and wired to the PROFIBUS DP/PA coupler. The PROFIBUS network, consisting of the PLC, variable speed drives, variable speed pumps, delivery pump and level transmitter, was configured and commissioned for controlling and monitoring from the programing computer. The program for the PLC was written using Siemens Step-7, compiled and downloaded to the PLC. The control and monitoring was done using the variable table. The wireless communication channel was then simulated using Matlab and Simulink. The wireless devices were then integrated into the PROFIBUS network and the MPI cable linking the programing computer and the PLC was then replaced by the wireless channel and the network was controlled and monitored from the programing computer over the wireless channel. On successful completion of this research task the research plant at MUT was controlled and monitored from the programing computer over the wireless channel and the researchers and demonstrators can now access the PLC and the PROFIBUS network using the wireless communication.

TABLE OF CONTENTS

CHAPTER FOUR: MODELLING OF WIRELESS COMMUNICATION CHANNEL

CHAPTER FIVE: DESIGN AND IMPLEMENTATION OF WIRELESS NETWORK

CHAPTER SIX: SIMULATION TEST AND ANALYSIS OF TEST RESULTS

CHAPTER SEVEN: CONCLUSION

APPENDIX A

APPENDIX B

APPENDIX C

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

A/D	Analog-to-digital converter
AM	Amplitude modulation
ASK	Amplitude shift keying
AWGN	Additive white Gaussian noise
BER	Bit error rate
BFSK	Binary frequency shift keying
BOP	Basic operator panel
CPU	Central processing unit
DP	Decentralised peripheral
FC	Function block
FM	Frequency modulation
FSK	Frequency shift keying
FT	Flow transmitter
GFSK	Gaussian frequency shift keying
GHz	Giga hertz
GSD	Generic station description
HLS	high level sensor
I2S	Inter-IC sound
ISM	Industrial, scientific and medical radio band
LLS	Low level sensor
LT	Level transmitter
M	Marker
MATLAB	Matrix laboratory
MBP	Manchester encoded bus powered
MFSK	M-array frequency shift keying
MIPS	Microcontroller without interlocked pipeline stages
MPI	Multipoint interface
MUT	Mangosuthu university of technology
OB	Operating block
OFDM	Orthogonal frequency division multiplex
PA	Process automation
PC	Personal computer

PHY	Physical layer
PLC	Programmable logic controller
PM	Phase modulation
PROFIBUS	Process field bus
QAM	Quadrature amplitude modulation
QPSK	Quadrature phase shift keying
RAM	Random access memory
RF	Radio frequency
RX	Receiver
SNR	Signal to noise ratio
SPI	Serial peripheral interface
SV	Supply valve
TDM	Time division multiplex
TTL	Transistor transistor logic
TX	Transmitter
UART	Universal asynchronous receiver transmitter
VSD	Variable speed drive
VSP	Variable speed pump

CHAPTER ONE: INTRODUCTION

In Process Automation and Control Engineering Process Control is defined as an engineering discipline dealing with all equipment and algorithms. The equipment and algorithms are used to regulate and control the variables of the process in order to maintain the output within the set requirements. Precise control and monitoring of variables such as temperature, pressure, and flow is vital in many process applications in order to ensure the quality of the end product and the safety of equipment and humans in the work floor. Process automation is the tool used to keep the operation of the plant within expected limits for optimum production that does not compromise safety while increasing production by maintaining the precise ratio of ingredients for the end product. Process control is achieved by using a control loop that measures the output of the process and compares it to the setpoint and adjusts the ingredients accordingly. If the output increases above the setpoint the control loop will reduce input ingredients until the desired output is realised. If the output decreases below the setpoint the control loop will increase the ingredients until the desired output is realised. The setpoint is the value of the process variable that must be maintained for a desired end product. The sensors are used to measure the ingredient variables and send the measured values to the control unit that will compare the measured values to the setpoint [4].

The process control equipment comprises of all the hardware used in process automation including sensors, pipes, gauges, valves, valve actuators, pumps, cables, PLCs and etc. The cables may be power cables, signal cables or PROFIBUS cables. In process control environment using signal cables there is a cable from the control device to each field device in the network. In process control environment using PROFIBUS, a single 2-wire cable can link up to 32 field devices, each field device having a unique address for identification. In this setup each device is referred to as the node and the control device sends and receives data to and from the nodes by means of datagrams. In Networking and Digital Communication a datagram is defined as a packet of data containing a source and destination information for routing through the network plus the message transmitted. The control system that requires a human being to operate the control device in order to make an adjustment or to control the process is called manual control system while the control system that does not require the human being to operate the device is called automatic control system.

The process control algorithm refers to the mathematical expression of a control function. The program is installed into a device to perform control operation. An automatic control system in which a process variable is measured and compared to the setpoint in order to determine the action to be taken in the control process is known as the closed loop control system. The result of the comparison between the measured variable and the setpoint is called an error signal and is proportional to the deviation of the output product from the expected value and it is used to adjust the input variable. An automatic control system that does not require the comparison in order to take control action is known as an open loop system. In open loop systems the control action is taken according to pre-set conditions regardless of the current status of variables.

1.1 Background of research and review of the work done

Modelling and simulation is based on a wireless communication channel to predict the behaviour of the system when implemented in real time. Real-time implementation entails building a real-world physical system that uses wireless communication for interconnection and sharing of data between the devices or between the controller and the devices. For this research, modelling and simulation was accomplished using Matlab while real-time implementation was commissioned on the experimental plant in the research lab at Mangosuthu University of Technology (MUT).

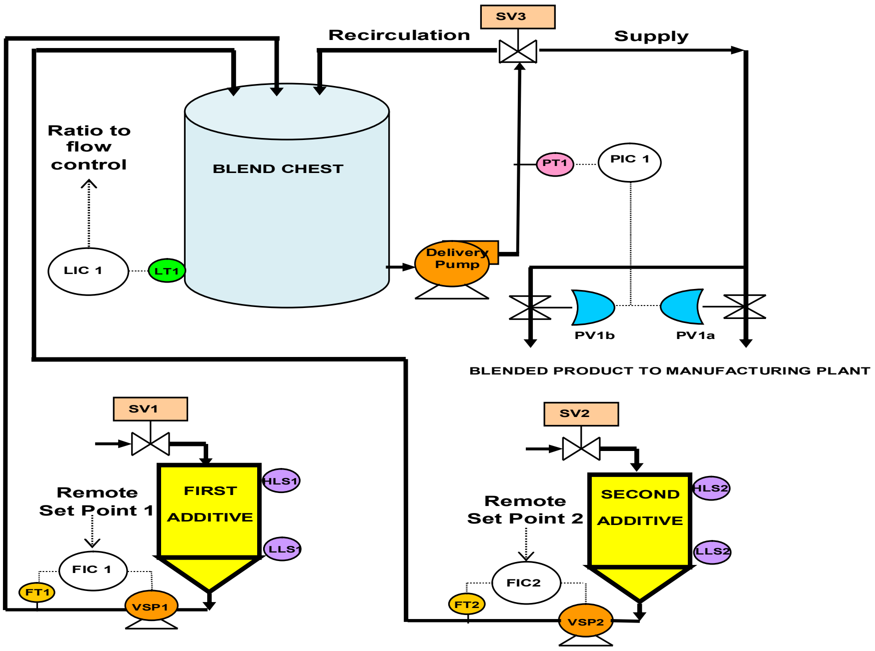


Figure 1.1: Block diagram of Experimental setup (P & ID) of Research Project No. Eng/01/2008, MUT

Figure 1.1 illustrates the interconnection of various components of the research plant and their distinguishing characteristics. The colours are used for identification purposes and do not impart any information about the components. Figure 1.2 gives the pictorial view of the research plant.

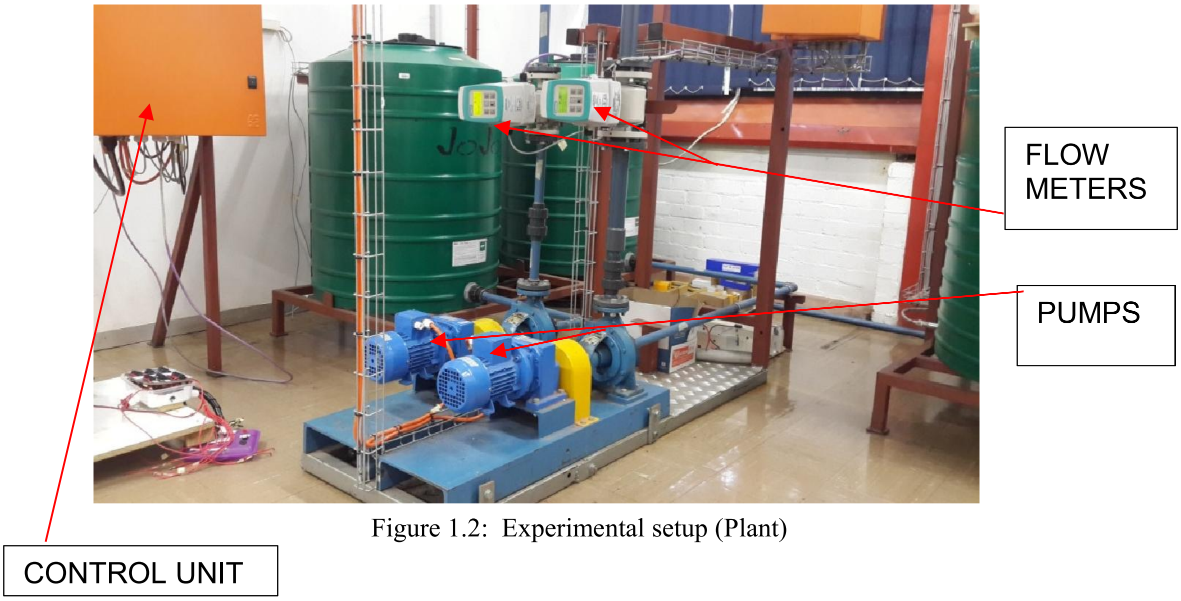


Figure 1.2: Experimental setup (Plant)

The purpose of this research was to design and implement a model for wireless communication between the programming computer and the PLC in the PROFIBUS network in order to eliminate the problem caused by fixed hardwiring of devices in the typical work floor. The network consists of the programming computer, PLC, PROFIBUS DP network, PROFIBUS DP/PA coupler, PROFIBUS PA field level transmitter, two variable speed drives, two variable speed pumps, two flow meters, and a delivery pump as shown in Figure 1.1 and Figure 1.2. All these devices were hard wired and fixed in their positions. The purpose of this research was to introduce minimum flexibility within the working floor by introducing wireless communication between the programming computer and the PLC thereby introducing wireless communication between the PLC and field devices for experimental purposes.

The research laboratory at MUT was constructed in 2008 under the leadership of Prof P Naidoo, Electrical department at MUT. The supply valves, SV1 and SV2, supply the liquid to additive tank1 and additive tank 2 respectively. The level sensors LLS1 and LLS2 monitor the low levels in tank1 and tank 2 respectively while HLS1 and HLS2 monitor the high level. The variable speed pumps, VSP1 and VSP2, pump the liquid from the additive tanks to the batch tank, Blend Chest. The pumps are controlled by the variable speed drives, VSD1 and VSD2. The delivery pump delivers the liquid from the batch tank back to the additive tanks for experimental purposes in the network shown in Figures 1.1 and 1.2. The PROFIBUS communication network provides communication between the control unit and the field devices. The capacitive level sensors measure the levels in the additive tanks. The flow transmitters, FT1 and FT2, send the rate of flow of the two pumps to the control unit. The supply valve, SV3, can recirculate the liquid to the blend chest or direct it to the additive tanks. The level transmitter, LT1 is the PROFIBUS PA field device. It senses the pressure of the fluid in the blend chest and sends the reading to the control unit. The interface module and the PROFIBUS DP/PA coupler interface the PROFIBUS DP and the PROFIBUS PA technologies. PROFIBUS DP uses RS485 protocol at the rate of 9.6 kb/s to 1200 kb/s while the PROFIBUS PA employs Manchester coded Bus Powered (MBP) protocol at the rate of 31.25 kb/s. Communication between the programming computer, PLC and field devices was hardwired and fixed.

Extensive research has been done in the field of wireless communication with specific focus in the development of commercial systems which has resulted in the development of systems like GSM, WI-FI, ZigBee, etc. but very little has been done in industrial setting due to electromagnetic interference and other forms of noise introduced by machinery. This research was done in order to introduce a wireless communication between a programming computer and the Programmable Logic Controller (PLC) and use the results to lay the ground for introducing wireless communication between the PLC and unit field devices in the MUT research lab for experimental and educational purposes [6].

The Communication infrastructure of current process control and factory automation systems is based on fieldbus networks. These networks provide adequate levels of performance, dependability, timeliness, maintainability and cost [1, 2], however, they lack in minimum mobility of field devices within the working floor that can enhance maintainability without affecting performance. Hybrid wired/wireless PRFOFIBUS (PROcess FIeld BUS) networks exist based on cellular technology, but this study focuses on communication channel adaptation possibilities and mobility of the programming computer within acceptable radius of the plant area.

This research task started by installing the PROFIBUS PA level transmitter, configuring the PROFIBUS network hardware, commissioning and testing the network, modelling the communication channel using Matlab and Simulink software then integrating and testing the wireless.

Due to successful implementation of wireless communication between the programming computer and the PLC in the PROFIBUS network as the result of this research, the assumption for replacing other parts of the network with wireless channel was made. There is a need for wireless communication in industrial automation setting to eliminate physical wiring between devices in order to enable access to devices in hazardous areas, eliminate electromagnetic interferences, and to allow acceptable movement of field devices as well. This need was evidenced by a leak in one of the pumps that results in the water being accumulated under the pump stand which cannot be moved because of hard wiring. The wireless communication was interfaced to existing automation settings to minimise the need for specialised training for staff.

Modelling and simulation was achieved by using Matlab to model and simulate the channel and deduce the nominal bit error rate (BER) plot to be used to forecast the behaviour and operation of the wireless channel. The characteristics of different wireless devices were compared and analysed and the AR9331, Wi-Fi system-on-chip (SoC) for wireless communication manufactured by Atheros, was found suitable to meet the basic requirements because it had built in radio frequency interfacing circuitry. The AR9331 wireless communication device is the heart of the MR3020 transceiver/router manufactured by TP-LINK. The PROFIBUS experimental plant hardware was configured and the components allocated addresses for identification by the PLC. The hardware configuration and allocation of addresses was achieved by using the programming computer and Siemens Step-7 software. After the hardware configuration was completed the program was written, compiled and downloaded to the PLC using the MPI cable. The PROFIBUS system was then tested for functionality and the variable tables were used to monitor and control different components of the network. After the PROFIBUS network was tested, the wireless communication device was installed and integrated into the network and tested for functionality by downloading the same hardware structure and program that was used to test the hardwired section. The same variable tables were used to monitor and control various components various components of the network and similar behaviour was observed. The Wireshark software was used to analyse the functionality of the wireless network

The challenges of using wireless communication in an industrial environment include radio interference, harmonic interference and other noises from different machinery and multipath fading. The radio interference is the disturbance that affects the electrical communication circuit due to electromagnetic radiation while harmonic interference is due to the harmonics in radio transmission. Multipath fading is interference to radio signals due to reflected waves. These disturbances were excluded in this research since the environment was experimental only and there was no heavy duty machinery to induce such interferences thereby eliminating the need for specialised testing equipment such as Wi-Spy channel analyser.

1.2 Review of work already done under this topic

Extensive research on wireless communication has been carried out although it still remains a challenge in industrial applications. The four main groups of the production industry (Agriculture, Mining, Manufacturing and Electricity, Gas and Water Supply) as classified by the South African Standard Industrial Classification [1] rely on research outputs in automation techniques for rapid and bulk production. Automation and control systems apply the use of microcontrollers and programmable logic controllers (PLCs) for data acquisition from the transducers and process control in the control devices. The extension of a wired factory-floor PROFIBUS communication network with wireless capability was proposed by [2], however, this proposal focused on the methodology to compute values for some parameters that enable optimal and bounded duration for handoff procedure and mobility management mechanism based on cellular technology. The electronics for linking a controller with a transducer was proposed by [3] in the designs for machine condition monitoring in order to predict failure before it occurs, amongst other applications. The most successful fieldbus technology is PROFIBUS and its variants, PROFIBUS DP (Decentralised Peripherals) and PROFIBUS PA (Process Automation) for reducing communication to one two-wire cable [4, 5].

A secure time division multiplex (TDM) based wireless technology was proposed by [6] where each slot is $10\,ms$ and the link schedule determines the next slot to be serviced and applies the mesh communication technology where each field device can forward data packets on behalf of other field devices. The approach that may be used for fabrication of devices for wireless pressure sensing was proposed by [7] where the thick-film was employed and wide range of materials were combined in order to produce a device with required mechanical properties. [8] Concurs with [2] in the electronics for linking the devices in an automated process and includes digital compensation algorithms to compensate for variations in sensor response.

A system that can acquire data in real-time from a sensor through a wireless system is described by [9] where each terminal is composed of sensor, signal conditioning circuit, analog-to-digital converter, microcontroller, transceiver and power supply. The current consumption is reduced by applying the ShockBurst mode where data packets are handled at a low level by the microcontroller and the transceiver. The design of pressure test system based on wireless communication technology is realized in [10] and the working principle, system composition and software design is introduced. The system structure and hardware design of low power wireless communication system was described by [11] based on 16 bit MSP430F1611 microcontroller and nRF2401 transceiver module where the RS232 interface was employed between the microcontroller and the control PC in real time data acquisition. The wireless communication between the PLC and the control computer (PC) was analysed by [12] where the PLC was linked by the microcontroller to the transceiver module on the PLC side and the PC was linked by the microcontroller via the MAX485 transceiver chip to the transceiver module on the PC side. The design employed the 6N137 opto-isolator for electrical isolation and electrical matching between the PLC and the microcontroller and between the PC and the microcontroller. The design also employed the double-port RAM (IDT7132 module) in order to avoid data congestion on the port.

With continuing development in industrial communication automation has become part of a network that covers service, maintenance, warehousing, and data acquisition for automation systems [13]. Because PROFIBUS DP allowed the integration of specific requirements into application profiles, it stood out from other fieldbus system and can service up to 32 nodes per segment. PROFIBUS DP is the high speed solution fieldbus that has been designed and optimised for communication between the automation systems and decentralised field devices via cyclic data traffic. The PROFIBUS PA network was specifically designed for the process control industries and petro-chemical industries due to their hazardous nature. PROFIBUS PA transmits data and power on the same two wires using Manchester encoding technology (MBP) [14] at the fixed rate of 31.25kbits/s. Because of different technologies employed in PROFIBUS DP and PROFIBUS PA variants of PROFIBUS, there was a need for the PROFIBUS DP/PA coupler which is a physical link between PROFIBUS DP and PROFIBUS PA [15]. The PROFIBUS PA communication is implemented as a partial system embedded in a higher-level PROFIBUS DP communication system [16] therefore it is of utmost importance to keep to the rules and regulations regarding the installation and commissioning of PROFIBUS

PA field devices [17] and to ensure specific Generic Station Description (GSD) files are properly installed. The pumps are controlled by the MICROMASTER 440 variable speed drives that employ the sophisticated vector control system that ensures uniformly high quality drive and dynamic response based on proper commissioning and correct system parameter settings [18].

Keeping an explicit representation of the design process, its components and interactions can enhance the design process and result in improved consistency maintenance during the design process and enhance documentation and modification of the design process [19]. According to [20] the interesting strategy for developing new applications (design process) is the development of object-oriented application framework that allows the reuse of previously developed objects to enhance problem solving at design and code levels. New applications have generated new requirements for integrated radio frequency (RF) structures in order to minimize losses and parasitic effects and improve electrical performance [21].

For wireless communication on the PROFIBUS DP/PA network, a set of transceivers are required because of two way data traffic, however, a careful choice of modules was made to meet the strict interface requirements. On the PC to PLC side of the network the RS485 technology is employed while on the PROFIBUS PA side Manchester encoding technology is employed. The WSN802G transceiver module is an ideal solution for sensor networks [21] while the nRF24L01 transceiver module is designed to work in the world wide ISM frequency band and it operates through a serial peripheral interface (SPI) using Gaussian frequency shift keying (GFSK) modulation scheme [22]. The HM-TR module is half duplex with automatic change over between transmit and receive modes employs FSK technology and is suitable for long range wireless transmission [23] while the Laird Technologies' fifth generation LT2510 transceiver modules set the standard for industrial RF communication by providing an extremely reliable communication link with the throughput of up to 280 kb/s [24]. The Linx Technologies (LT) series transceiver module is ideal for bidirectional wireless transfer of serial data in the 260 – 470 MHz band [25]. While the MRF89XA PICTAIL was designed for experimental purposes using the explorer development board and daughter board [26], the YS-1020L transceiver module by Yishi Electronics was designed for professional wireless data transmission systems in short range and can interface directly with RS485 devices and other UART components with RS232 or TTL interface [27]. Serial data transmission can be realized using the serial peripheral interface (SPI) in which data in a byte can synchronously be shifted in or out one bit at a time in order to communicate with another serial peripheral device [28].

These communication components are realizable in embedded systems where technology advancements are providing more cost effective devices for integrating computational processing and wireless communication and other functionalities [29]. Communication in a microcontroller can be achieved using the universal asynchronous receiver transmitter (UART) or serial peripheral interface (SPI) technology, however, the PIC16F84a (used in this application) does not have hardware UART or SPI hence the technology is implemented in software [30].

A computer model of signal and data processing was created to verify a wireless communication channel based on time differences of signal arrivals [31] where Simulink is utilised for the signal processor model and Matlab software is utilised for mathematical evaluations and for determination of initial conditions. This is the model that was explored and utilised in this research exercise. The implementation was realised using the TL-MR3020 transceiver manufactured by TPLINK [32] because it uses the AR9331 chip [33] as its core which has built in wireless interface supporting IEEE802.11standard.

Once the wireless network is finalised, it can contain several sensing nodes. It will be desired that these nodes be cheap and energy-efficient for quality results [37] while hoping to integrate wireless networking with consumer electronics [38]. The wireless sensor networks are used to extend the functionality of protocols and simulation results analysis [39]. It is of importance to guard against unauthorised intrusion on the wireless network [40], however, this will not be necessary in this research because it is for experimental purposes.

1.3 Dissertation overview

Chapter one introduces the dissertation project and its vision. It outlines the problem of the research and the solution. It gives the description of the dissertation document. The simulation modelling was done using MATLAB and the real time implementation was done in the laboratory. It also outlines the literature review where the related text is reviewed and analysed. This provides the source of acquired knowledge and forms the reference of the documentation. Chapter two outlines process automation giving the equipment used for processing, communication and interfacing. It also outlines the characteristics and advantages of process automation. Chapter three outlines the PROFIBUS network and its technologies and analyse various components utilised in the implementation of the network. It also deals with hardware configuration and software development for the PROFIBUS network implementation and validation. It concludes with the commissioning of the plant and testing from the BOP and from the control station. Chapter four outlines the modelling of the channel and expands on Matlab and Simulink simulation of BPSK, QPSK and 16-QAM digital modulation schemes over the AWGN channel. The performance of these modulation schemes is finally analysed for noise performance using the BER plots. Chapter five outlines the design and implementation of wireless network by illustrating the integration of wireless communication devices onto the existing PROFIBUS network. Chapter six analyses the results for Matlab and Simulink simulation of different modulation schemes and the use of BER plots to measure the noise performance of the channel. It also analyses the results of the PROFIBUS network and the results of the wireless network. It uses ProfiTrace tool to analyse the PROFIBUS network and the Wireshark network analyser to analyse the wireless network. Chapter seven concludes the research and it summarises and reviews all the work done, discuss the implications of the findings and offer suggestions for future research.

1.3.1 Aim

The aim of this research was to setup a wireless communication network to test its performance in an environment of industrial sensors, final correcting elements and controller with a PROFIBUS coupler.

1.3.2 Objective

In this research the Siemens PLC (CPU-313C-DP) and the Siemens SIMATIC DP/PA coupler (FDC 157-0) comprised the hardware of the system, with the PROFIBUS network interfaced with field devices. The SIMATIC Step-7 version 5.5 software formed the control and monitoring algorithm. The TP-Link wireless transceiver module (TL-MR3020) was integrated into the communication link between the programming and operating unit, to replace the MPI cable. This was a facility to incorporate remote control and programming to derive some comparative statistical analysis between MPI and wireless communication protocols.

1.3.3 Hypothesis

The research intended to set up a Siemens PROFITRACE analyser to process the data on the PROFIBUS network between the programing unit or operator station, and the plant via the Siemens MPI protocol. An alternate architecture was set up using the Wireshark network analyser to process the data on the PROFIBUS network between the programing unit or operator station, and the plant via the TP-Link wireless transceiver module (TL-MR3020).

1.4 Conclusion

This chapter defined process control and highlighted the background of the research. The work already done under the same topic was reviewed. The aim, objectives and hypothesis were highlighted. The next chapter will deal with process automation.

CHAPTER TWO: PROCESS AUTOMATION

2.1 Introduction

In process automation and control engineering, process automation is defined as the use of computer equipment, involving computer, microcontrollers and programmable logic controllers (PLCs) to increase the efficiency in the production plant by using control equipment to monitor and control the process instead of human beings. Where there is no process automation, human beings have to manually operate the equipment according to the given settings on which to run the plant and try to monitor the performance and the output of the plant. Where there is process automation the sensors are located at various stages of the process where they measure the process variable and send the reading to the controlling device for control purposes [4].

2.2 The computer

The computer is used to write the program that regulates the efficient running of the plant. The program can be written in various programming languages depending on the type of PLC being used. The program uses measurements and settings to monitor how the plant is working and to simulate different modes of operation and control process variables in order to ensure consistent quality of the output product. The program is then tested, compiled and downloaded to the PLC for automatic controlling of the plant. The computer may then be used to view and monitor the performance of the plant.

2.3 The microcontroller

The microcontroller is a digital controlling device embedded in digital sensors to convert the measured process variable to digital signals and send them to the controlling device (PLC). The microcontroller also receives and decodes control signals from the controlling device depending on the type of the sensor. The embedded microcontroller contains the CPU, ROM, RAM, I/O ports and other peripherals in a single chip and the control program is burnt into the memory by the manufacturer. The microcontrollers are available in various sizes ranging from 8-pin to 80-pin devices with various hardware peripherals built into them like hardware timers, analog-to-digital converter (ADC), pulse width modulation (PWM), etc. and can be used for control purposes, but the theory and the technique of using the microcontroller is a course of study on its own.

2.4 The PLC

The PLC is explained in detail in section 3.2. It provides for connection of input and output devices and it is used in process automation to store the control program and to run the algorithm in order to monitor and control the process. The PLC reads the status of the inputs and outputs at regular intervals and compare with the set points in order to determine the form of control required to achieve optimum production. It is more efficient than human beings, but operators must be able to manually override the automatic control when necessary.

There are various manufacturers of automation equipment but this research uses the Siemens Step-7 PLC because it already exists in the research plant at MUT. Automation is necessary because the throughput is increased when controlling and maintaining the ratio of the ingredients required for manufacturing the end product by using the feedback control loop which maintains the ratio set at the beginning of the process in order to meet the specifications of the end product, thereby minimises wastage of ingredients. Process automation is characterised by slow procedures and longer system service life [13]. Various technologies are employed in process automation, but this research uses the PROFIBUS network.

2.5 Communication in process automation

Process automation connects automation systems and process control systems with field devices using suitable technology such as 4 to 20 mA technology, field bus technology, etc. [13]. A wide range of products, from different manufacturers, may be interconnected with separate power cables and data cables; analog or digital, resulting in bulky cabling, however, other communication technologies were developed in order to reduce the number of cables in process automation. The communication function in process automation is standardised by the open system interconnection model (OSI model) where the communication process is partitioned over seven layers according to IEC 61158 standard. In Data Communication and Networking, the seven layers of the OSI model are defined as:

(i) The physical layer deals with the transfer of data bits over transmission media such as copper wires, coaxial cable or optical fibre.

(ii) The data link layer provides for the transfer of frames of data across a transmission link that directly connects two nodes.

(iii) The network layer provides for the transfer of data packets across a communication network.

(iv) The transport layer for the end-to-end transfer of information from the process in the source machine to the process in the destination machine.

(v) The session layer is used to control the manner in which data are exchanged like Full-duplex, Half-duplex, synchronization and error detection and correction.

(vi) The presentation layer provides for universal data representation so that the different machines at the source and destination can interpret the message correctly and

(vii) The application layer uses protocol to transfer files of data and to maintain the communication network.

In some applications all the layers of the OSI model are used while in other applications fewer layers are used, for example, in fieldbus technology the physical layer, the data link layer and the application layer are utilised [13]. For reliable data transfer, various transmission technologies such as 4 to 20 mA, RS485, MBP and optical transmission [13] are employed. In 4 to 20 mA, a reading of 0 is represented by a current 4 mA while the maximum reading is represented by 20 mA; other values are scaled between these readings. In RS485 technology, data is transmitted as the voltage difference between two lines, MBP uses changing current to transmit data [14] while optical transmission employs light pulses in an optical fibre cable. Caution must be exercised to integrate different technologies in the same process automation setup and also the distance must be considered because electromagnetic interference may impair the reliability of the system if the input or output distance is extensive [15]. In other systems, several components may be connected on the same communication link and have unique identifying codes used by the control system to identify them, thereby extensively reducing the number of data cables and reducing the cost.

2.6 Process automation characteristic

Process automation consists of a control device, field devices and interconnection network and is characterised by the type of processing in the control device, properties of field devices and the properties of the communication network.

(i) The control device is usually a digital processing system with or without memory. If the signals to be processed are analog, they must be converted to digital before processing and converted to analog after processing.

(ii) The field devices (digital or analog) may be a transducer, sensor or actuator.

- The digital devices mostly have an embedded microcontroller that will condition the signal before sending to the control device or receive the signal from the control device.
- The analog device, connected to the input of the control unit, generates a continuous signal and there must be a form of analog to digital conversion before the signal reaches the processing device while that connected to the output must have a digital to analog converter interface.

(iii) The interconnection network may consist of twisted wire cables or PROFIBUS cables:

- The pairs of insulated conductors are twisted and bounded to form a cable that may be covered by sheathing material to protect the conductors against lightening or it may not be sheathed. A pair of conductors is used for data communication between the control device and each field device.
- The PROFIBUS cable consists of two insulated conductors, A and B, that are covered by an insulation cladding and sheathed by a multistrand wire mesh that is used as grounding terminal. A single PROFIBUS cable may connect up to 32 nodes, each having a unique identification address as explained in chapter 3.

2.7 Advantages of process automation

The advantages of process automation include increased efficiency, increased production, and increased quality of the end product, reduction in error and reduction in cost. The benefits of process automation are:

(i) Where no process automation is used, human beings control the process and they sometimes have to switch off the plant during break times, resulting in under- utilization of the equipment whereas with process automation, the system runs efficiently according to specification all the time.

(ii) In process automation the control system maintains the ratio of the ingredients according to the required output and minimises the wastage thereby increasing the production and maintaining the quality of the end product.

(iii) The control device executes the program controlling the system and maintains the status of the plant eliminating the human error.

(iv) There are process automation systems, like PROFIBUS, that utilise fewer cables thereby reducing the installation cost and their reliability also reduces maintenance cost.

The research plant illustrated in Figures 1.1 and 1.2 consists of the batch tank for mixing the ingredients from the two additive tanks. Process automation employing PROFIBUS technology was utilised to control the pumps and to measure and monitor the levels of the fluids in the tanks. This research replaced the multi-point interface (MPI) cable with the wireless link between the programing unit and the controlling device (PLC) thereby providing movability of the programing unit within the working floor.

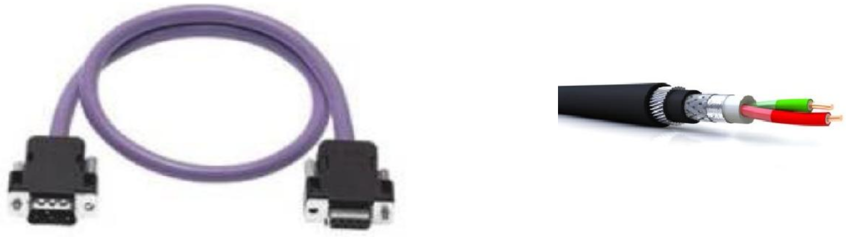
2.8 Conclusion

This chapter defined process automation and highlighted its application in an industrial environment. It also highlighted the components used for process communication in the industrial environment. The next chapter will explain the PROFIBUS network and its configuration.

CHAPTER THREE: HARDWARE CONFIGURATION ASPECTS OF THE PROFIBUS NETWORK

3.1 Introduction

The PROFIBUS network consists of the control PC, PLC, PROFIBUS DP/PA coupler and the PPROFIBUS PA field device. The MPI cable connects the programming computer to the PLC. The PLC is connected to the PROFIBUS DP/PA coupler via a PROFIBUS DP cable. The PROFIBUS DP cable is shown in Figure 3.1(a) while the PROFIBUS PA cable is shown in Figure 3.1(b). The PROFIBUS DP/PA coupler is connected to the PROFIBUS PA field device via a PROFIBUS PA cable. PROFIBUS is field bus communication standard for linking process control and plant automation modules. Instead of running individual cables from a main controller to each sensor and actuator, a single multi-drop cable is used to connect all devices, with a high speed, bidirectional, serial messaging used for transfer of information. The PROFIBUS DP transmits at 45.5 kb/s while the PROFIBUS PA transmits at 31.25 kb/s [5]. These parts will be allocated addresses in the experimental program setup for the PLC to be able to identify them.



(a) (b)

Figure 3.1: The PROFIBUS DP cable [36].

3.2 The PLC: CPU-313C-DP

A programmable logic controller (PLC) in Figure 3.2 is a solid state user programmable control system with functions to control logic, sequencing, timing, arithmetic data manipulation and counting capabilities. It is an industrial computer that has a central processing unit, memory, input/output interface and a programming device. The central processing unit (CPU) provides the intelligence of the controller. It accepts data (status information from various sensing devices), executes the user control program stored in the memory and gives appropriate output commands to field devices. Input/output interface is the communication link between the controller and field devices. The field devices are wired to the I/O interfaces. Through these interfaces the processor can sense and measure physical quantities regarding a machine or process, such as, proximity, position, motion, level, temperature, pressure, etc. Based on the status sensed, the CPU issues command to output devices such as valves, motors, alarms, etc.

The programing unit is used to write the program, compile it and download it to the PLC for execution and to monitor the variables.

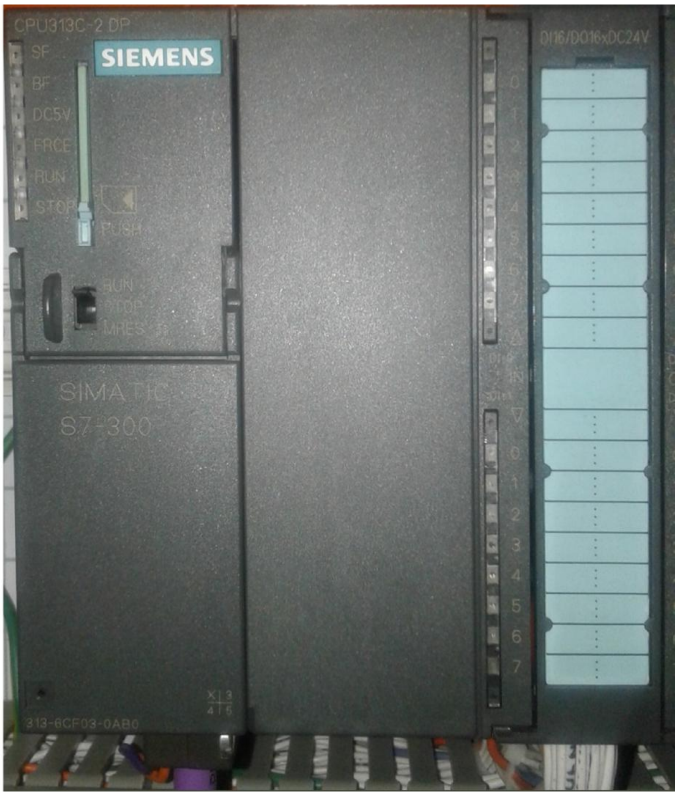


Figure 3.2: The PLC.

3.3 The PROFIBUS DP/PA coupler

The PROFIBUS DP/PA coupler is the FDC 157-0AC83-0XA0 interface to convert between the PROFIBUS DP technology and the PROFIBUS PA technology. The PROFIBUS DP technology runs at a speed from 9.6 kbps up to 12 Mbps using RS485 balanced transmission while the PROFIBUS PA technology runs at 31.25 kbps Manchester Encoded Bus Powered (MBP) [5, 15].

3.4 The PROFIBUS PA network

The PROFIBUS PA network connects the PROFIBUS PA field device to the PLC using the PROFIBUS PA cable via the PROFIBUS DP/PA coupler. The PROFIBUS PA signal is Manchester encoded [13, 14, 16]. In Manchester coding there is always transition in the middle of the bit period as demonstrated in Figure 3.3. In this coding a digit “0” is transmitted as logic level 1 followed by a logic level 0 while a digit “1” is transmitted as a logic level 0 followed by a logic level 1. The data is therefore transmitted as current transitions instead of voltage levels. These transitions are useful in clock recovery.

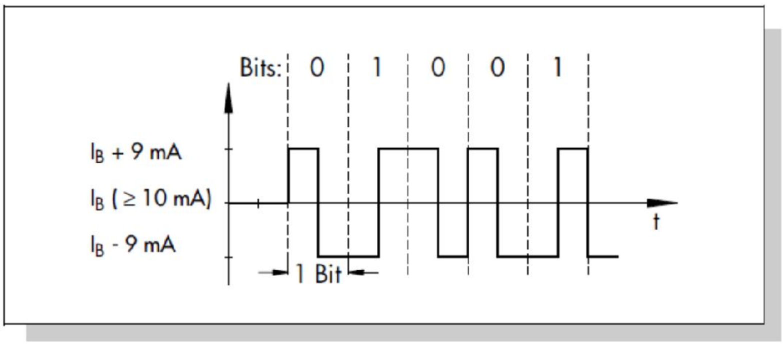


Figure 3.3: Manchester coding [16].

3.5 Hardware configuration

The basic instruments are interconnected by baseplate and physical wires and each instrument has its unique specifications and product/part number useful for hardware configuration and commissioning. The hardware interconnection is displayed in Figure 3.4. PROFIBUS PA cable is normally blue but at the time of installation of the project the contractor used an equivalent as the required cable was not available.

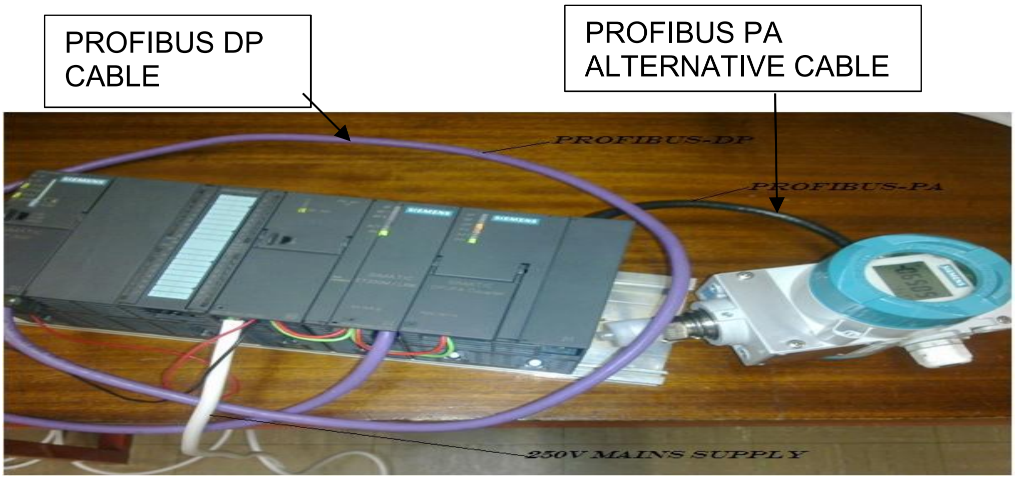


Figure 3.4: The interconnection of components.

The components of hardware configuration are:

(i) PLC: CPU_313-DP. The PLC is independently powered by the 24 V DC power supply. The DC power supply steps down and rectifies the 230 V AC supply and produce the 24 V DC that is used by the PLC and other modules and field devices [15].

(ii) IM 153-2BA82-0XB0. The interface module provides internal circuitry for the PROFIBUS DP/PA coupler [15].

(iii) FDC 157-0AC83-0XA0. The PROFIBUS DP/PA coupler communicates the PROFIBUS PA signal to the controller as a PROFIBUS DP signal and the PROFIBUS DP signal from the controller to the PROFIBUS PA field device. It provides the transition between the RS485 technology used by the PROFIBUS DP and the Manchester encoded bus powered (MBP) technology used by the PROFIBUS PA [15]. The interface module and the PROFIBUS DP/PA coupler are shown in Figure 3.5.

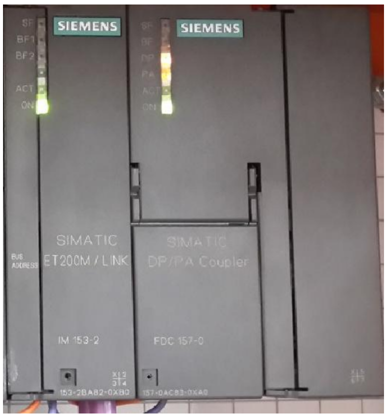


Figure 3.5: The interface module and the coupler.

(iv) 7MF 4034-7BA00-1AB6-Z. The level transmitter in Figure 3.6 provides the transducer that converts the pressure of the liquid in the blending chest to proportional electrical pulses that display the pressure measurement on the display and convey these pulses to the PLC over the PROFIBUS PA cable for displaying on the control PC and for control purposes [17].

There are specific working conditions that must be observed when connecting this loop:

(i) The PROFIBUS cable must be at least one meter in length to avoid reflection.

(ii) Insulation must not be removed from the bus connector heads with monkey-grip.

(iii) All connections must be tight enough to minimise signal loss.

(iv) The PROFIBUS-PA screen wire is terminated on the coupler side and on the meter side for signal transmission. A maximum of 32 field devices may be connected in one PROFIBUS network segment.

3.6 Software configuration

This research project utilised the Siemens SIMATIC Manager S7 (Step7) software for device configuration, monitoring, process measurement and control. The hardware configuration, address allocation, operating blocks, function blocks and program form a project. The Step7 utilises the project wizard to guide the programmer through the steps required to create a project and to select the units required to complete the PLC application. Figure 3.7 indicates the project wizard window for selection of the CPU and its address. The CPU is allocated address 2 because address 1 is reserved for the power supply on the rack.

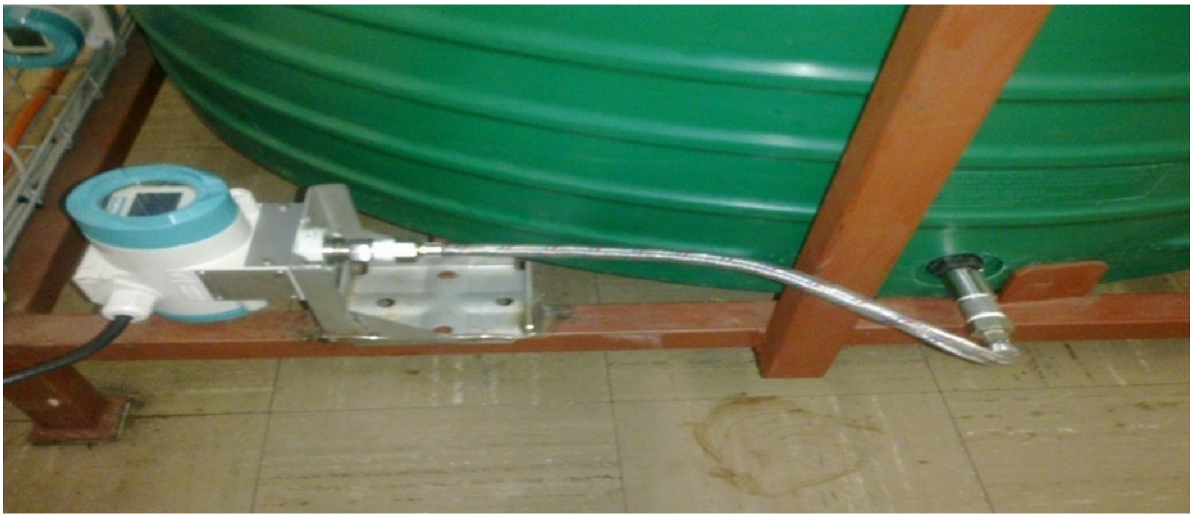
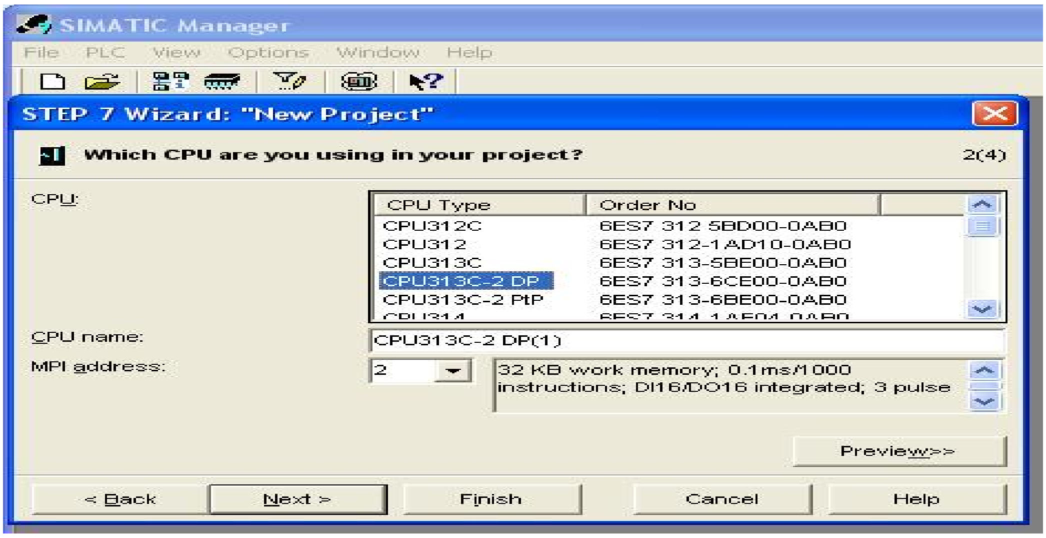


Figure 3.6: The level transmitter is connected as shown in the actual plant.



SIMATIC Manager
File PLC View Options Window Help
STEP 7 Wizard: "New Project"
< Back
Next >
Finish
Cancel
Help
Preview>>

Which CPU are you using in your project? 2(4)

CPU:

CPU Type	Order No
CPU312C	6ES7 312 5BD00-0AB0
CPU312	6ES7 312-1AD10-0AB0
CPU313C	6ES7 313-5BE00-0AB0
CPU313C-2 DP	6ES7 313-6CE00-0AB0
CPU313C-2 PtP	6ES7 313-6BE00-0AB0
CPU314	6ES7 314-1AE04-0AB0

CPU name: CPU313C-2 DP(1)

MPI address: 2

32 KB work memory; 0.1ms/1000 instructions; DI16/DO16 integrated; 3 pulse

Figure 3.7: The new project wizard.

The creation of the project on the project wizard also provides for inserting the necessary blocks, function blocks (FC) and operating blocks (OB), the selection of the programming language (statement list or ladder) and the project name as shown in the layout in Figure 3.8. The function blocks hold the logic of the application in ladder diagram or statement list while the operating blocks provide interface between operating system of the controller and user program.

Figure 3.8 illustrates the insertion of the station (SIMATIC 300 Station) on the project where the station refers to the PLC to be used

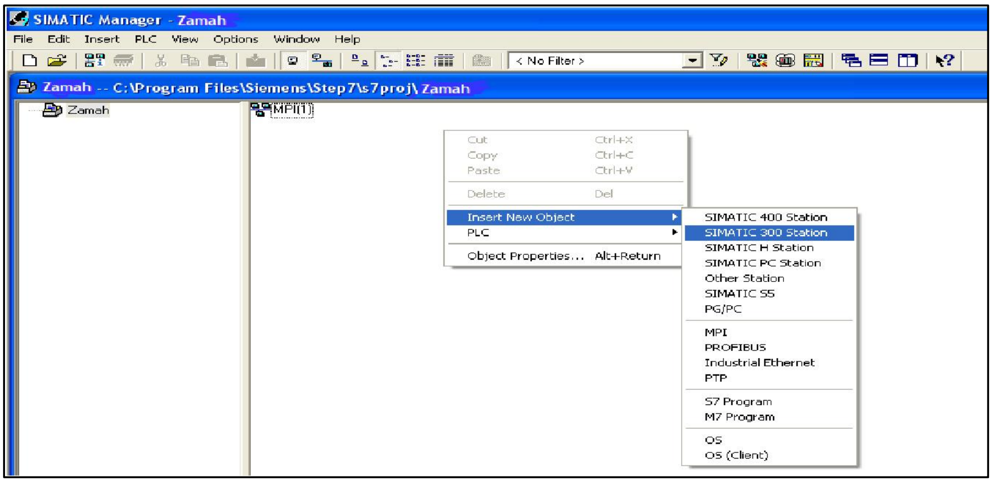


Figure 3.8: Project creation using the new project wizard.

After the project has been created on the PLC, the station to be used (SIMATIC 300) is inserted. The hardware of the chosen station is opened by clicking on the station icon and the actual network is created and the component catalogue is opened for device selection. The network is created following the sequence:

(i) Insert the rail.

(ii) Insert the station on the rail (dragging).

(iii) Insert the DP master system.

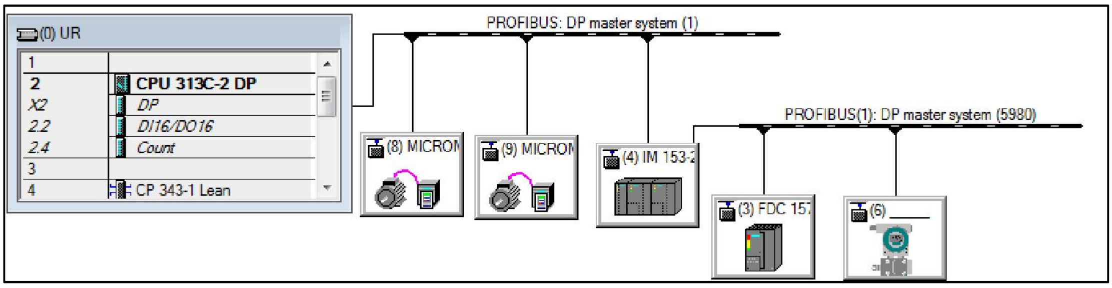
(iv) Insert the interface module (IM 153-2).

(v) Insert the PROFIBUS DP/PA coupler (FDC 157) and

(vi) Insert the level transmitter (7MF 4034).

The network is created consisting of the CPU, variable speed drives (MICROMASTER 1 & 2), and the interface module IM 153-2 on the PROFIBUS DP master system while the PROFIBUS DP/PA coupler FDC 157 and the PROFIBUS PA field device SITRANS P are connected on the PROFIBUS PA master system as illustrated in Figure 3.9. The CPU is allocated address 2, the PROFIBUS DP/PA coupler is allocated address 3, the interface module is allocated address 4, and the level transmitter is allocated address 6 while the variable speed drives are allocated addresses 8 and 9. These addresses are used by the system to monitor each component in the network. Figure 3.9 illustrates the hardware configuration layout and the allocated addresses of devices.

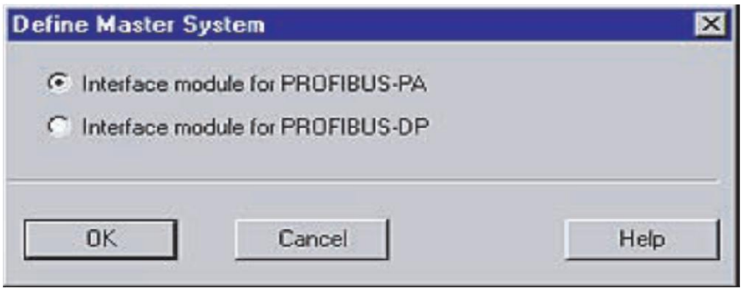
Each component on the network has its own configuration window used to set it up according to specification of the operation. The configuration window is opened by right clicking on the component and scrolling down to its properties. It is in this configuration window where the IM 153-2 interface module is configured for PROFIBUS DP or PROFIBUS PA interfacing as shown in Figure 3.10. The functionality of the completed network relies on the installation of the specific Generic Station Description (GSD) files that define the operation of each device in the network [17].


(0) UR

1	
2	CPU 313C-2 DP
X2	DP
2.2	DI16/DO16
2.4	Count
3	
4	CP 343-1 Lean

PROFIBUS: DP master system (1)
(8) MICROM
(9) MICROM
(4) IM 153-2
PROFIBUS(1): DP master system (5980)
(3) FDC 157
(6)

Figure 3.9: PROFIBUS network layout.


Define Master System

● Interface module for PROFIBUS-PA
○ Interface module for PROFIBUS-DP

Figure 3.10: The configuration window for the PROFIBUS master system.

3.7 Program setup

After the network configuration was completed, the program to control the operation was written using the statement list. The ladder diagram or the combination of the two languages could also be used. The program writing procedure commenced with the insertion of the blocks. The organisation block OB1 was inserted first and an algorithm *call FC1* was inserted in it as illustrated in Figure 3.11. The OB block determines the structure of the program. The simulator icon must be opened in order to view the inputs and the outputs. It is advisable to save and download the created modules of the program at regular intervals.

The function block (FC) contains the program routines. The main program was written in the FC1 window using statement list and it was saved and downloaded to the PLC as illustrated in Figure 3.12. This algorithm is always executed whenever an FC is called by another logic block. The FC1 block for this research application consists of two networks. Network 1 has the algorithm:

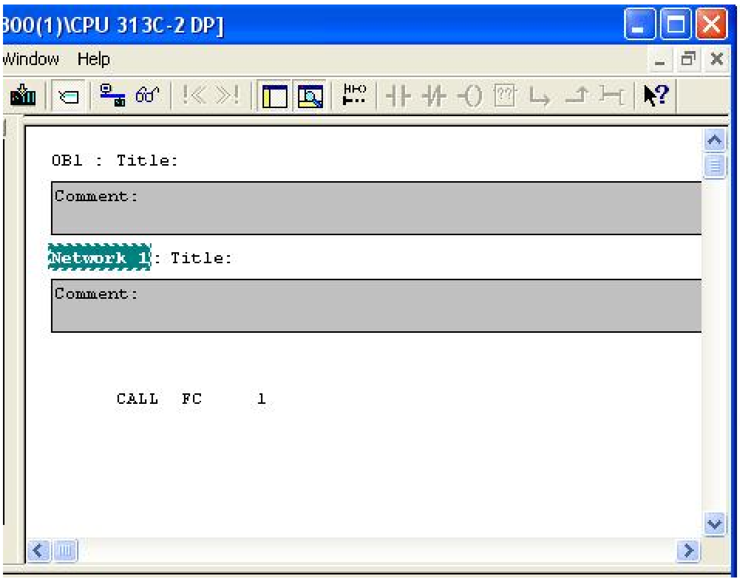
```
L    PIW   277
T    MW    0
```

The first line of this algorithm reads the value on the level transmitter at input word address 277, then line two stores the reading on memory location 0.

Network 2 has the algorithm:

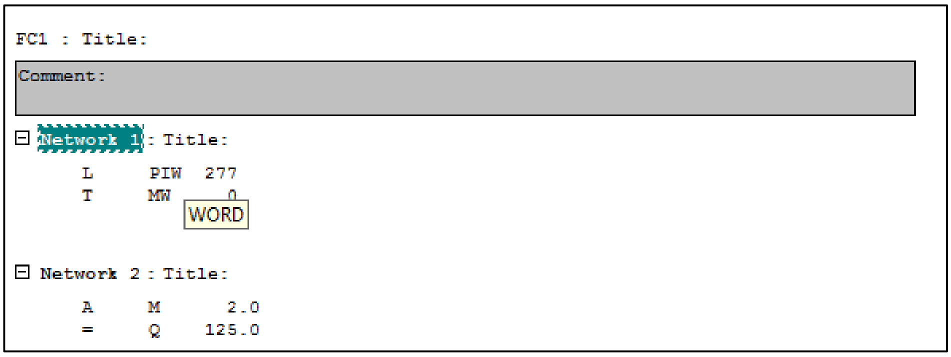
```
A    M    2.0
=    Q    125.0
```

The first line of this algorithm monitors marker M 2.0 which controls the delivery pump. When the marker M 2.0 is closed, it activates the output Q 125.0 which starts the delivery pump and when the marker M 2.0 is open, it deactivates the output Q 125.0 which switches off the delivery pump. The marker operation is Boolean in nature.



```
OB1 : Title:
Comment:

Network 1: Title:
Comment:

      CALL  FC     1
```

Figure 3.11: The OB block and its algorithm.



```
FC1 : Title:
Comment:

Network 1: Title:
      L      PIW   277
      T      MW      0
                WORD

Network 2 : Title:
      A      M       2.0
      =      Q     125.0
```

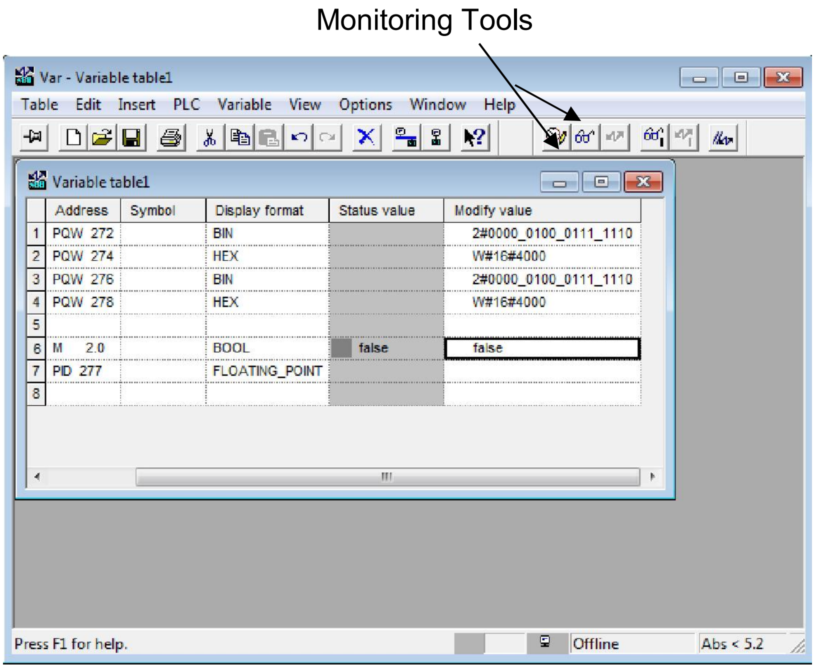
Figure 3.12: The program in the FC1 block.

When the network was successfully configured, it was saved and compiled then downloaded to the PLC for execution. After downloading the program to the PLC, the simulator was monitored and the results observed on the variable table. The variable table is illustrated in Figure 3.13 where the different variables are monitored. The variable table is opened by selecting 'modify variable' on the PLC menu and the variables to be monitored are listed with their data types. The control word is 47EH in hexadecimal or 010001111110 in binary. The first digit on the right is a start/stop bit used to start or stop the pump. If it is 1 the pump runs, but if it is 0 the pump stops. Line 1 of the variable table contains address PQW 272 which is the output word for variable speed drive 1 of binary format controlled by the control word 47EH. Line 3 contains address POW 276 which is the output word for variable speed drive 2 of binary format controlled by the control word 47EH. The control word 47EH will be explained in detail in the commissioning section. Line 2 and line 4 contain the setpoints for pump 1 and pump 2 respectively. Line 6 contains the address M 2.0 which is the marker for controlling the delivery pump and is of Boolean format. The modify value of 'false' stops the delivery pump while the modify value of 'true' starts the delivery pump. Line 7 contains the address PID 277 of the floating-point format which displays the reading of the level transmitter in bars. The variable table is illustrated in Figure 3.13.

The monitoring of variables is activated by clicking the goggles of the monitoring tool in the variable table of Figure 3.13. The monitored variables are displayed on the table and the device can be controlled by modifying its controlling value and clicking the modify button in the monitoring tool.

3.8 Commissioning

The plant network layout consists of the programming computer, PLC, variable speed drives (MICROMASTRER 440), interface module (IM 153-2), PROFIBUS DP/PA coupler (FDC 157), the PROFIBUS PA field device (SITRANS P level transmitter), two variable-speed pumps, two flowmeters and one delivery pump with the supply control valve. These devices needed to be commissioned in order to ensure optimum functionality and diagnostics in order to detect, classify and evaluate errors and fault messages when monitoring the functions that run automatically while the plant is in operation [29]. After proper installation and instrument configuration was completed, the parameters were set using the MICROMASTER 440 BOP device for commissioning [29] and they are sequentially listed in Table 3.1.



	Address	Symbol	Display format	Status value	Modify value
1	PQW 272		BIN		2#0000_0100_0111_1110
2	PQW 274		HEX		W#16#4000
3	PQW 276		BIN		2#0000_0100_0111_1110
4	PQW 278		HEX		W#16#4000
5					
6	M 2.0		BOOL	false	false
7	PID 277		FLOATING_POINT		
8					

Press F1 for help. Offline Abs < 5.2

Figure 3.13: The variable table

The parameter can be defined as a numerical factor that defines the operation of a system. Parameter P003 defines user access levels and determines access parameters where 1= Standard, 2 = Extended, 3 = Expert and 4 = Service. The user has access to certain parameters depending on access level parameter selected, therefore, selecting access level 3 as an expert, the user has access to most important parameters required for commissioning and can define other parameters when necessary. The first setting was P003 = 3, P010 = 30, P0970 = 1 in order to reset all parameters to factory default settings. The reset process takes approximately 10 seconds [28] to complete thereafter quick commissioning was performed. At the end of commissioning the parameters were set as indicated in Table 3.1, (values within brackets are for pump 2). Parameter P0700 = 1 was set in order to select the basic operation panel (BOP) for operating the field devices using the panel while parameter P1910 allowed the system to identify the motor data according to specifications set on parameters. Parameter P3900 = 3 completed the commissioning process. The devices were operated and verified using the BOP control unit, and when all devices functioned according to set parameters, then parameter P0700 = 2 was set to transfer control to the control terminal and the program was loaded to the PLC using the programing unit and the plant was operated using the status word (control word) on the variable table.

Table 3.1: Commissioning parameter list

Parameter Setting	Description
P003 = 3	Defines user access level to parameter sets
P010 = 30	Reset all parameters to factory default.
P0970 = 1	Reset complete.
P003 = 3	
P010 = 1	Quick commissioning.
P0300 = 1	Asynchronous motor indication.
P0304 = 380 V (230 V)	Voltage rating pump1, pump2
P0305 = 0.94 A (1.63 A)	Current rating pump1, pump2
P0307 = 0.37 Kw (0.37 Kw)	Power rating pump1, pump2
P0308 = 0.77 (0.77)	Power factor pump1, pump2
P0310 = 50 Hz (50 Hz)	Frequency rating pump1, pump2
P0311 = 1674 rpm (1674 rpm)	Rated motor speed pump1, pump2
P0700 = 1	BOP (Basic Operating Panel)
P1000 = 1	(MOP) Frequency set point
P1910 = 1	Select motor data identification
P3900 = 3	End commissioning, carry out necessary calculations, clear unused parameters.

3.9 Basic Operator Panel (BOP)

The basic operator panel (BOP) is used to set the parameters during commissioning and the seven segment display displays the parameters and the parameter values. The Fn button, Figure 3.14, is used to activate the parameters and the up/down buttons are used to scroll through the set values. When the desired setting is reached, the P button is pressed to accept the parameter setting.

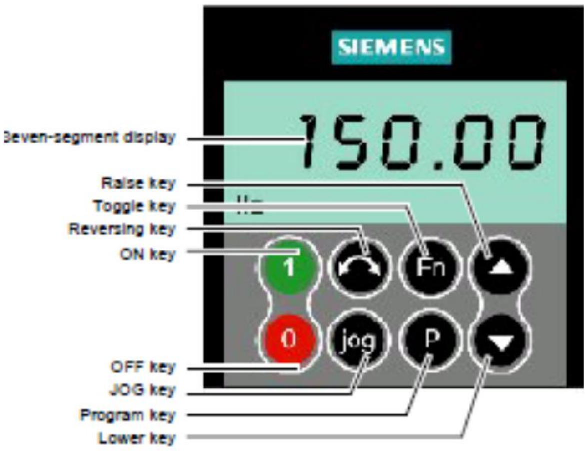


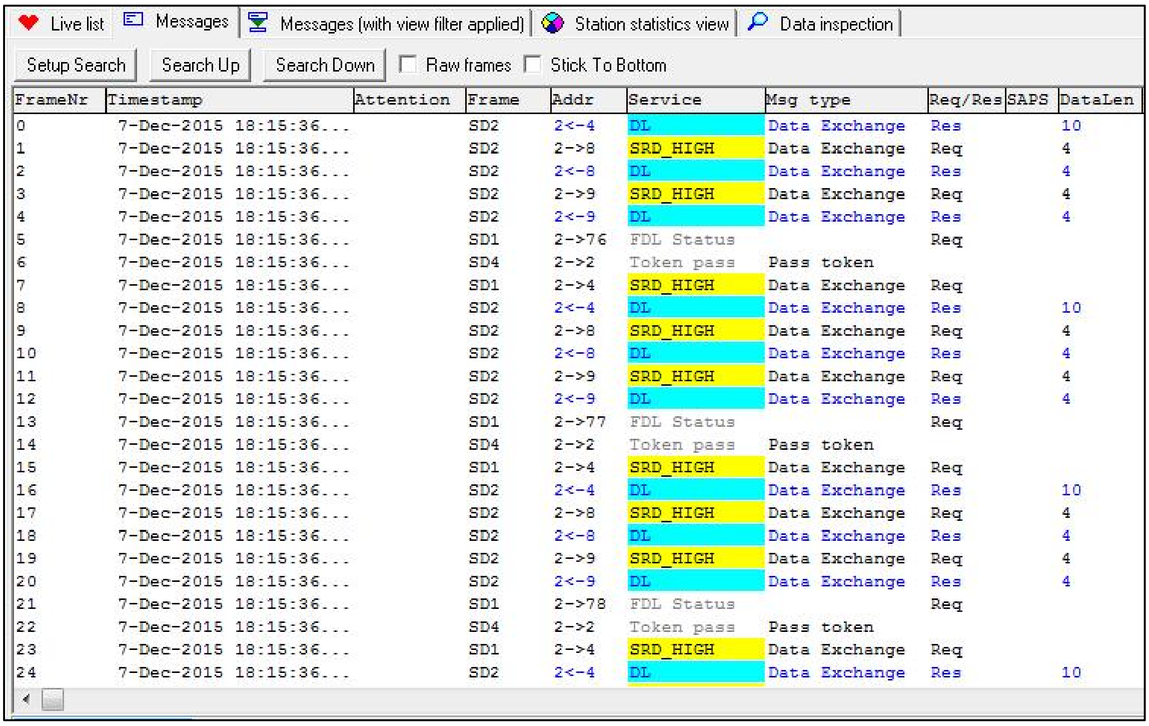
Figure 3.14: The BOP device for setting the parameters [29].

The ↶ button is used to reverse the settings while the jog key is used to deviate from the set point to the left or to the right. The green button (1) is used to manually start the operation while the red button (0) is used to stop the operation. When the parameter setting is finalized, the operation is transferred from the BOP to the control station where operation is monitored and controlled using the control word and variable tables. ProfiTrace PROFIBUS analyser was used to analyse the PROFIBUS network and the results are illustrated in Table 3.2. The results show that the average data transmitted under various load conditions is fairly constant. An example of the ProfiTrace capture is illustrated in Figure 3.15. In Figure 3.15, the FrameNR header is used to specify the sequence between message samples while the Timestamp header specifies the date and time the message was captured and recorded. It can be seen that the messaged frames were of SD1, SD2 and SD4 type. The SD1 message type does not contain the user data, the SD2 message contains the user data and the SD4 message type is a token sent from the master to another master. In this setup there is no other master, therefore the SD4 type message was sent by the master to itself. The Addr header specifies the address of the source and destination of the message and the $->$ indicates the direction of the message. The service header specifies the type of service of the message as follows:

(i) DL – Data response, low priority: It occurs when the station acknowledges that the message received was correct.
(ii) SRD_HIGH – Send and receive data, with high priority.
(iii) FDL Status – This message is used to identify stations on the bus.

The DataLen header specifies the length of the message. There is a lot of functions integrated into the ProfiTrace Tool that were not shown in this capture because they were not useful to the required analysis

Five tests were carried out and the results were recorded in Table 3.2. These tests were carried out in order to keep the record of the performance of the PROFIBUS network under various load conditions. This record will be compared with the performance of the wireless network in order to validate the replacement of the PROFIBUS network with the wireless network. Column two in Table 3.2 displays the statistical summary VSP1. Column three displays the statistical summary for VSP2. Column four displays the summary for delivery pump. Column five displays summary for VSP1+VSP2 while column six displays summary for when all the pumps are running where VSP1 and VSP2 represent the variable speed pumps. The statistical items: packets, time elapsed, average, bytes, etc. had to be physically counted from each capture because the PofiTrace tool does not show the statistical summary like other network analysis tools.



FrameNr	Timestamp	Attention	Frame	Addr	Service	Msg type	Req/Res	SAPS	DataLen
0	7-Dec-2015 18:15:36...		SD2	2<-4	DL	Data Exchange	Res		10
1	7-Dec-2015 18:15:36...		SD2	2->8	SRD_HIGH	Data Exchange	Req		4
2	7-Dec-2015 18:15:36...		SD2	2<-8	DL	Data Exchange	Res		4
3	7-Dec-2015 18:15:36...		SD2	2->9	SRD_HIGH	Data Exchange	Req		4
4	7-Dec-2015 18:15:36...		SD2	2<-9	DL	Data Exchange	Res		4
5	7-Dec-2015 18:15:36...		SD1	2->76	FDL Status		Req		
6	7-Dec-2015 18:15:36...		SD4	2->2	Token pass	Pass token			
7	7-Dec-2015 18:15:36...		SD1	2->4	SRD_HIGH	Data Exchange	Req		
8	7-Dec-2015 18:15:36...		SD2	2<-4	DL	Data Exchange	Res		10
9	7-Dec-2015 18:15:36...		SD2	2->8	SRD_HIGH	Data Exchange	Req		4
10	7-Dec-2015 18:15:36...		SD2	2<-8	DL	Data Exchange	Res		4
11	7-Dec-2015 18:15:36...		SD2	2->9	SRD_HIGH	Data Exchange	Req		4
12	7-Dec-2015 18:15:36...		SD2	2<-9	DL	Data Exchange	Res		4
13	7-Dec-2015 18:15:36...		SD1	2->77	FDL Status		Req		
14	7-Dec-2015 18:15:36...		SD4	2->2	Token pass	Pass token			
15	7-Dec-2015 18:15:36...		SD1	2->4	SRD_HIGH	Data Exchange	Req		
16	7-Dec-2015 18:15:36...		SD2	2<-4	DL	Data Exchange	Res		10
17	7-Dec-2015 18:15:36...		SD2	2->8	SRD_HIGH	Data Exchange	Req		4
18	7-Dec-2015 18:15:36...		SD2	2<-8	DL	Data Exchange	Res		4
19	7-Dec-2015 18:15:36...		SD2	2->9	SRD_HIGH	Data Exchange	Req		4
20	7-Dec-2015 18:15:36...		SD2	2<-9	DL	Data Exchange	Res		4
21	7-Dec-2015 18:15:36...		SD1	2->78	FDL Status		Req		
22	7-Dec-2015 18:15:36...		SD4	2->2	Token pass	Pass token			
23	7-Dec-2015 18:15:36...		SD1	2->4	SRD_HIGH	Data Exchange	Req		
24	7-Dec-2015 18:15:36...		SD2	2<-4	DL	Data Exchange	Res		10

Figure 3.15: ProfiTrace capture

Table 3.2: Test results for PROFIBUS network

	VSP1	VSP2	Delivery Pump	VSP1 + VSP2	All Pumps
Bytes captured	357240	647920	967890	1140096	887380
Time elapsed between first and last capture	13 sec	44 sec	34sec	52 sec	38 sec
Average bytes/sec	27480	14725	28467	21925	23352
Average MBit/sec	0.22	0.12	0.22	0.18	0.19

The ProfiTrace capture results recorded in Table 3.2 were filtered for successfully transmitted user data only, excluding error messages, tokens and request messages. The average of 0.186 Mbit/sec was obtained which concurs with the PROFIBUS system which transmits 1.5 Mbit/sec in real time. The values plotted for VSP1, VSP2, Delivery Pump, VSP1+VSP2 and All Pumps in Table 3.2 were obtained from the Profitrace capture illustrated in appendix B1 to B5.

3.10 Conclusion

This chapter introduced the PROFIBUS network components and technologies employed. It also introduced software development and commissioning of existing plant. The next chapter will highlight the modelling of a wireless communication channel using Matlab.

CHAPTER FOUR: MODELLING OF WIRELESS COMMUNICATION CHANNEL

4.1 Introduction

This chapter provides the design of an alternative communication technology in an automated process plant by integrating a wireless link in the PROFIBUS network in order to enhance the production and maintenance of the plant by introducing mobility of control devices and field devices. It is important to keep an explicit representation of the design process in order to give easier access to information for both the designer and the user resulting in improved consistency maintenance during the design process and enhance documentation of the design process [18]. Major problems may arise from the complexity of the design process. The design process should develop a technique for representation and construction of integrated system architectures [18].

The design process for this research consists of two phases, MATLAB simulation and hardware implementation.

4.2 Wireless channel simulation using MATLAB

The PROFIBUS data is transmitted as datagrams at a rate of 45 kbps in RS485 format. This rate is very low to transmit over a wireless channel. A very high carrier frequency is used to carry the data over a wireless channel. Because data is binary in nature, a binary modulation scheme was used to modulate the carrier frequency with data to be transmitted and a binary demodulation scheme was used to recover the data from the carrier frequency at the receiving end. The target wireless module employs the AR9331 chip that supports the IEEE802.11b standard which includes binary phase shift keying (BPSK), quadrature phase shift keying (QPSK) and quadrature amplitude modulation (16-QAM) digital modulation schemes [33]. In this research these modulation schemes were simulated using Matlab and Simulink transmitting over the additive white Gaussian noise (AWGN) channel.

4.2.1 AWGN channel

In communication the channel is the medium used to transfer data from the source (Tx) to the destination (Rx). The performance of the communication channel is characterised by the noise content the channel injects into the signal. Additive white Gaussian noise (AWGN) in the channel comes from different sources like vibration of atoms, radiation from other objects, etc. [34]. The important characteristic of the AWGN channel is that the amplitude frequency response is flat and the phase response is linear for all frequencies [34] therefore the signal will pass without amplitude and phase distortion, but it will suffer only from noise which can be filtered at the receiver. The received signal can be modelled by the function:

$$r(t) = x(t) + n(t) \ [34] \qquad (1)$$

Where $r(t)$ is the received signal, $x(t)$ is the transmitted signal and $n(t)$ is the noise power injected by the channel to the signal passing through it. Equation (1) indicates that the channel introduces no amplitude loss and no phase distortion, but the only distortion is introduced by the AWGN [34].

4.2.2 Binary phase shift keying (BPSK)

In Communication BPSK is a form of phase shift keying (PSK) that switches between two phases and transmits one phase to represent a "1" and another phase to transmit a "0". Because the data is in binary form this modulation scheme is known as binary phase shift keying (BPSK). Figure 4.1 illustrates the BPSK modulation for the data message"10110'. The phase transition is noticed at the end of the bit "1" to mark the beginning of the bit "0" and the phase is continuous where a "1" is followed by a "1" similarly where a "0" is followed by a "0". The Matlab program for plotting Figure 4.1 is listed in appendix A1. The lower portion of figure 4.1 represents the switching of frequency at different angles to illustrate phase transitions.

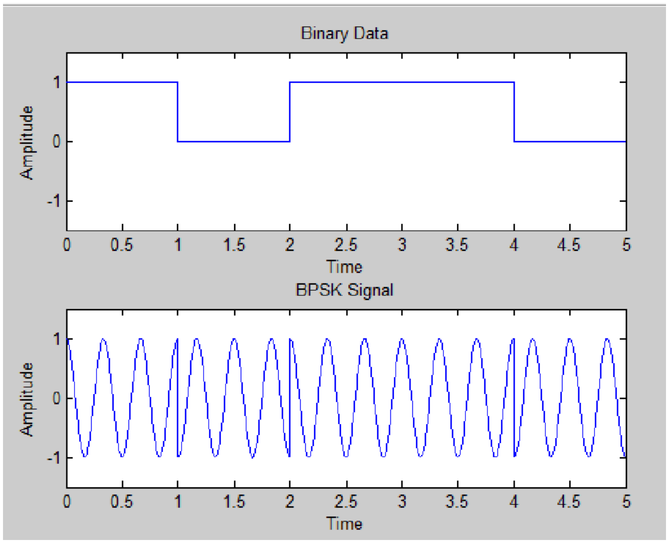


Figure 4.1: BPSK modulation

4.2.3 Quadrature phase shift keying (QPSK)

In Digital Communication Quadrature phase shift keying (QPSK) is defined as the digital modulation scheme in which each symbol consists of two bits of information. The signal shifts between the states are separated by 90° implying four possible states of the carrier. In QPSK the modulator takes two independent signal components, one in-phase (I) and the other 90° out of phase (Q) and modulates and transmits the sum of the two signals as a single composite signal. The receiver demodulates the incoming composite signal and separates I and Q components. The QPSK modulation scheme is illustrated in figure 4.2 where the plot is subdivided into four slots. The slot in row 1: column 1 indicates the discrete binary data to be modulated. The slot in row 1: column 2 represents the I component while the slot in row 2: column 2 represents the Q component. The slot in row 2: column 1 represents the modulated signal representing 11, 01, 10 and 00. The Matlab program for plotting Figure 4.2 is listed in appendix A 2.

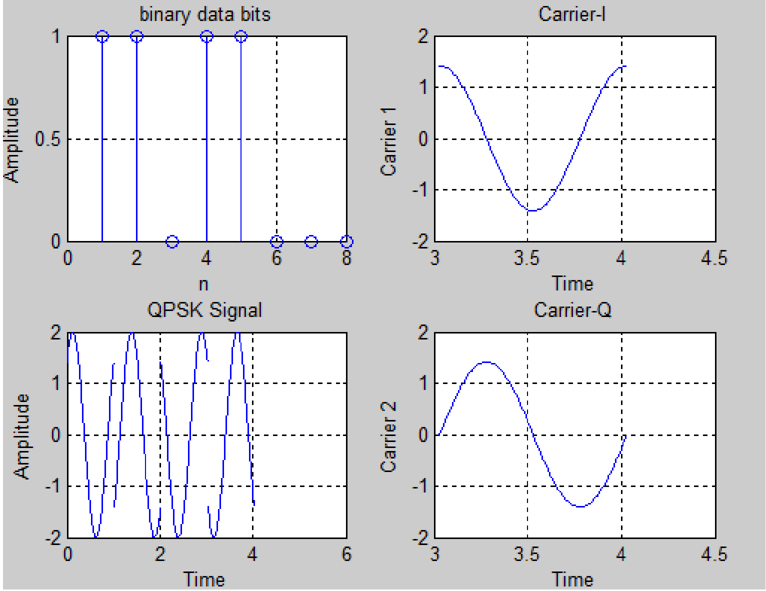


Figure 4.2: QPSK modulation

4.2.4 16-Quadrature amplitude modulation (16-QAM)

In radio engineering quadrature amplitude modulation (QAM) is the modulation scheme that combines the amplitude shift keying with phase shift keying by changing the amplitude and the phase of the carrier waves. The commonly used QAM scheme is 16-QAM where each symbol represents four bits of data. QAM can be applied in both analog and digital systems. In general the QAM is the same as QPSK except that the amplitude is also modified in QAM while in QPSK the amplitude remains constant.

4.3 Analysis of the different modulation schemes using Simulink

Matlab Simulink was used to model and simulate the communication channel consisting of the transmitter, AWGN channel and receiver. At the transmitter, Bernoulli random data generator block from communication toolbox was used to generate random data stream to be transmitted over the channel. The Bernoulli block generates data with a probability (ρ) of zero as 0.5 at a rate of one sample per second. The model was simulated with signal to noise ratio (Eb/No) ranging from -13 dB to 3 dB and the results were recorded in Table 4.1 and the BER rate was plotted using the Matlab semilogy function in order to make the y-axis logarithmic. The theoretical BPSK plot was plotted and compared positively to the simulated plot. This was done in order to simulate the behaviour of the channel when data is transmitted through it.

4.3.1 The BPSK model

The BPSK model consists of the BPSK modulator that modulates the binary data from the Bernoulli generator in BPSK format for transmission over the AWGN channel on the transmitter side. It consists of the BPSK demodulator on the receiver side that decodes the received signal and recover the data. The received signal may be corrupted by the noise injected by the channel and some data bits may be received in error. The error is calculated by comparing the received signal with the transmitted data. The BPSK model is illustrated in Figure 4.3. The first row on the display indicates the calculated BER, the second row indicates the number of errors detected during comparison while the third column indicates the number of comparisons made.

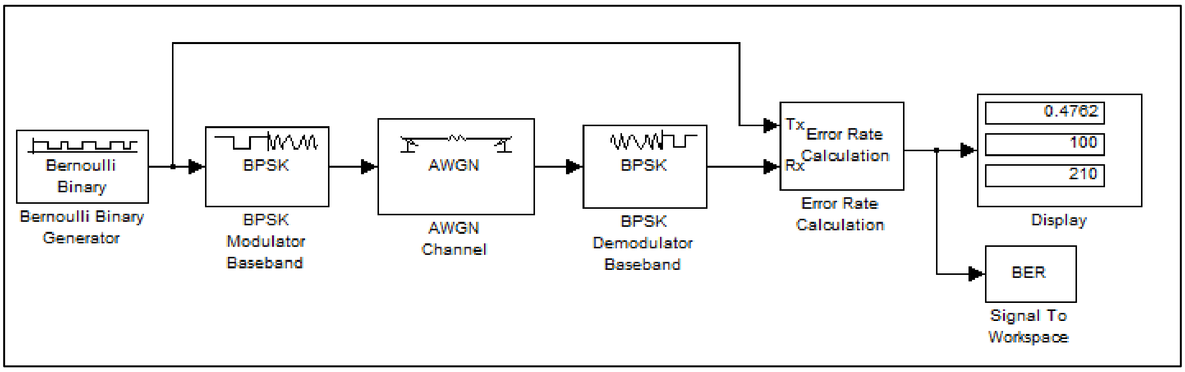


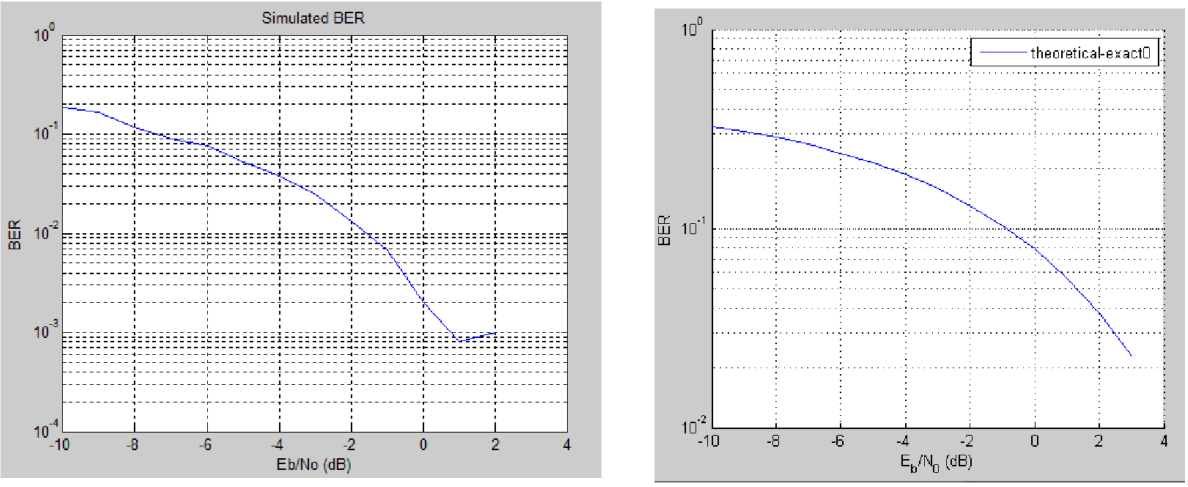
Figure 4.3: BPSK model

During the simulation of the model in Figure 4.3 the signal to noise ratio Eb/No was varied from -13 dB to 3 dB and the number of comparisons, number of errors and the BER values were read off from the display and recorded in Table 4.1. Column 1 and column 4 were used to plot the BER and it compares positively with the theoretical BPSK BER plot that was plotted using Simulink Monte Carlo and indicates that the errors decrease as the signal to noise ratio is increased. The error is determined by comparing the received data stream with the transmitted data stream and the number of comparisons that were made during each test is recorded in column 2 while number of errors is recorded in column 3. Table 4.1 also illustrates that as the signal to noise ratio (Eb/No) of the channel is increased, the number of comparisons increases while the number of errors is reduced.

Table 4.1: Test results for BPSK simulation

Eb/No (dB)	No of Comparisons	No of Errors	BER
-10	533	100	0.1876
-9	606	100	0.165
-8	859	100	0.1164
-7	1111	100	0.09001
-6	1298	100	0.07704
-5	1911	100	0.05233
-4	2672	100	0.03743
-3	4024	100	0.02458
-2	7496	100	0.01334
-1	1.00E+04	68	0.006799
0	1.00E+04	20	0.002
1	1.00E+04	8	0.0007999
2	1.00E+04	1	1.00E-04
3	1.00E+04	0	0

The simulated BPSK BER is plotted in Figure 4.4(a) while the theoretical BER is plotted in Figure 4.4 (b) indicating the close relationship between the simulated and theoretical plots. The Matlab program for plotting Figure 4.4 is listed in appendix A 3.



(a)

(b)

Figure 4.4: Simulated and theoretical BPSK BER

4.3.2 The QPSK model

The QPSK model uses the QPSK modulator on the transmitter side to modulate the information signal from the Random integer generator and send it to the AWGN cannel. On the receiver side the QPSK demodulator demodulates the received signal and extracts the message. The AWGN channel injects noise on the signal and transmits it to the receiver. The received signal is compared with the transmitted data in order to calculate the data bits received in error by the error rate and the statistic is displayed on the display. The QPSK model where each symbol represents 2 bits (4PSK) is illustrated in Figure 4.5.

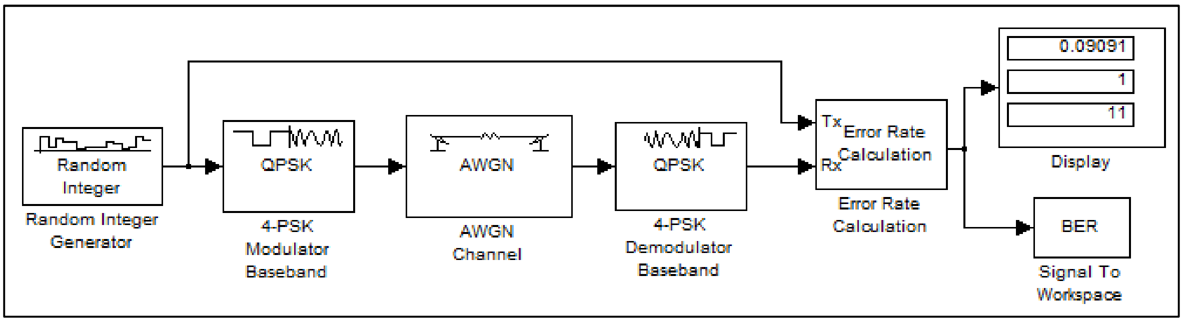


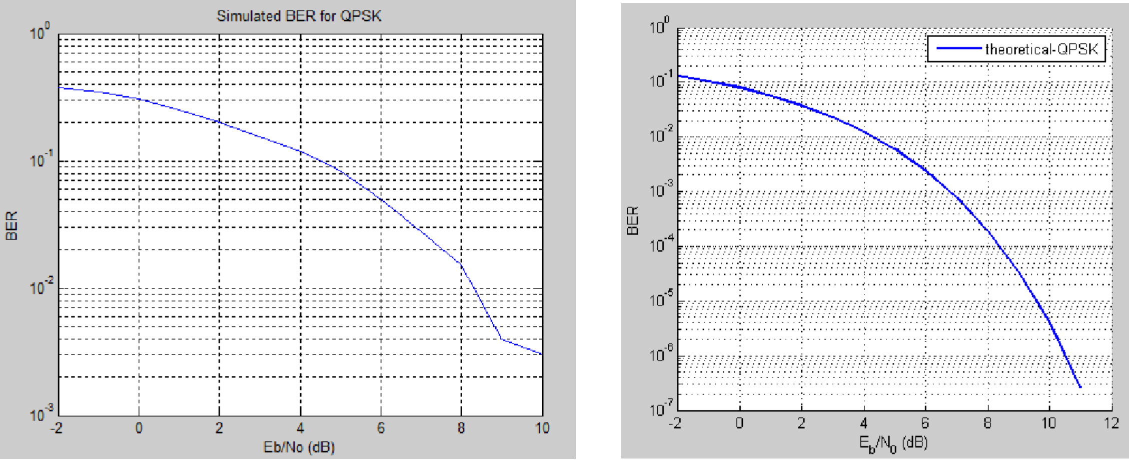
Figure 4.5: QPSK model

During the simulation of the model in Figure 4.5 the signal to noise ratio was varied from -2 dB to 11 dB and the results plotted in Table 4.2 and used to plot the simulated BER.

Table 4.2: Test results for QPSK simulation

Eb/No (dB)	No of Comparisons	No of Errors	BER
-2	1001	378	0.3778
-1	1001	344	0.3437
0	1001	304	0.307
1	1001	250	0.2498
2	1001	197	0.1988
3	1001	155	0.1548
4	1001	119	0.1189
5	1001	82	0.08192
6	1001	50	0.04995
7	1001	28	0.02797
8	1001	15	0.01499
9	1001	4	0.003996
10	1001	3	0.002997
11	1001	0	0

The simulated QPSK BER is plotted in Figure 4.6 (a) while the theoretical BER is plotted in Figure 4.6 (b) indicating the close relationship between the simulated and the theoretical plots. Table 4.2 indicates that number of comparisons stays the same for this modulation scheme with the increase in signal to noise ratio while the number of errors decreases. The Matlab program for plotting Figure 4.6 is listed in appendix A 4.



(a) (b)

Figure 4.6: Simulated and theoretical QPSK BER

The constellation diagram could be used to illustrate the relationship between the in-phase (I) and quadrature (Q) components of the signal, but the BER was sufficient to define the performance of the QPSK model.

4.3.3 The 16-QAM model

In the 16-QAM model on the transmitter side, the 16-Array rectangular QAM modulator block is used on the input side to modulate the data from the random integer generator and send it to the AWGN channel. The AWGN channel injects noise onto the signal and sends it to the receiver. The 16-Array rectangular QAM demodulator is used on the receiver side to extract the received data from the AWGN channel. The received signal is compared with the transmitted signal in order to detect the data received in error and the results are displayed on the display. In 16-QAM scheme each symbol is represented by 4 bits. The 16-QAM model is illustrated in Figure 4.7.

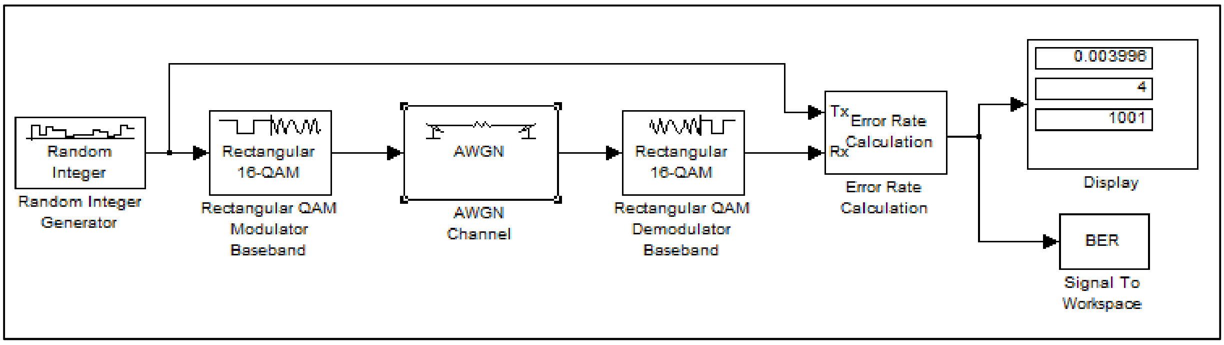


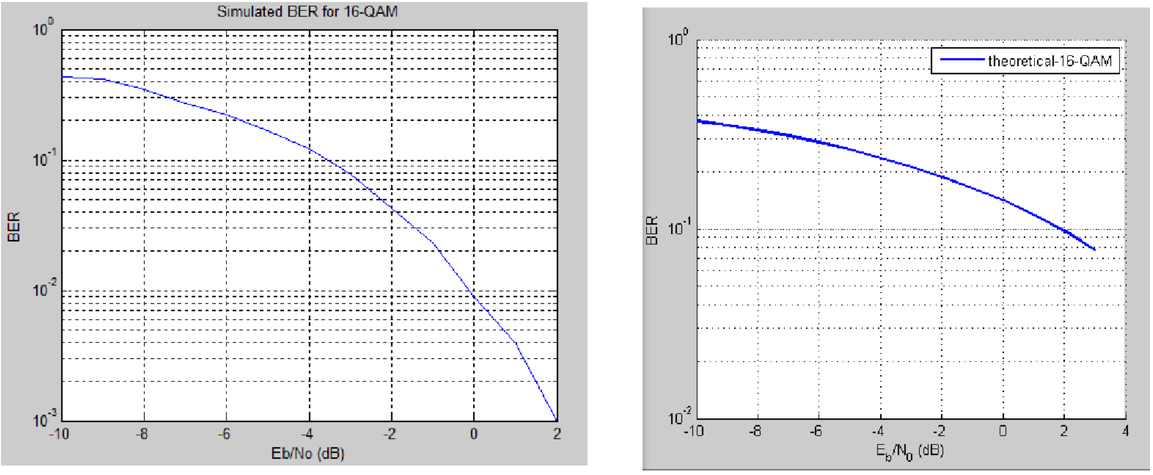
Figure 4.7: 16-QAM model

During the 16-QAM simulation of the model in Figure 3.7 the signal to noise ratio was varied from -10 dB to 3 dB and the results were plotted in Table 4.3.

Table 4.3: Test results for 16-QAM simulation

Eb/No (dB)	No of Comparisons	No of Errors	BER
-10	1001	493	0.4925
-9	1001	418	0.4176
-8	1001	345	0.3447
-7	1001	274	0.2737
-6	1001	224	0.2238
-5	1001	109	0.1688
-4	1001	121	0.1209
-3	1001	78	0.07792
-2	1001	42	0.04196
-1	1001	23	0.02298
0	1001	9	0.008991
1	1001	4	0.003996
2	1001	1	0.000999
3	1001	0	0

The results illustrated in Table 4.3 were used to plot the BER for the 16-QAM model and compare it with the theoretical BER that was plotted using the BERTool and Monte Carlo simulation. The simulated BER and theoretical BER are illustrated in Figure 4.8. The close relationship between the simulated BER and the theoretical BER for 16-QAM is evident in Figure 4.8, however, where Figure 4.8 (a) illustrates the simulated BER plot while Figure 4.8 (b) illustrates the theoretical BER plot. The Matlab program for plotting Figure 4.8 is listed in appendix A 5.



(a) (b)

Figure 4.8: Simulated and theoretical BER for 16-QAM

Figure 4.8 illustrates the close relationship of the simulated BER and theoretical BER for 16-QAM modulation scheme.

4.4 Discussion of simulation results

In Information Theory the probability of error in a transmitted bit can be calculated using the error function tables and is modelled by the function in equation (2):

$$P_b = \tfrac{1}{2} erfc\left(\sqrt{\gamma_b}\right) \tag{2}$$

Where P_b represents the probability of the bit to be transmitted and $\gamma_b = E_b / N_o$ represents the ratio of the power carried by the bit to noise power of the channel. The AWGN channel selected for this research injects white noise with a Gaussian distribution to the signal, but in this research calculations were not done, instead Matlab Simulink was used to perform the calculations and estimate the usable BER.

The BER plots for BPSK, QPSK and 16-QAM modulation schemes are indicated in Figures 4.4, 4.6 and 4.8 respectively. The Eb/No range was selected by first identifying the ratio that gives 0 errors and then simulating for 14 levels which seemed to give enough information to be able to plot the curves. From the tables and the BER plots it is seen that when the number of bits per symbol in modulation schemes is increased the bit error rate also increases and as a result, the performance of the system with more bits per symbol is degraded. Although the number of levels per symbol was not simulated for each modulation scheme used in this research, the BER plots indicates enough evidence to make assumption that the BPSK modulation scheme will perform better than the QPSK and the 16-QAM schemes in terms of noise performance, but the other schemes will perform better in terms of volume of data transmitted per given time.

These communication schemes were chosen and analysed in this research because they are employed in the target device, AR9331, to implement a digital communication platform that handles data and transform it for wireless transmission and decode the received data from the wireless signals and transform it to the format applicable for the other parts of the automation system. These tests were carried out in order to model the behaviour of the target device when handling data stream and to have an overview of the channel performance.

4.5 Conclusion

This chapter highlighted the simulation of a wireless communication channel using different models. The bit error rate was used to analyse the performance of the channel. The next chapter will focus on the design and implementation of a wireless communication network.

CHAPTER FIVE: DESIGN AND IMPLEMENTATION OF WIRELESS NETWORK

5.1 Introduction

The wireless network was integrated into the existing PROFIBUS network after configuring the hardware to monitor the level of the fluid in the batch tank and control the variable speed pumps and the delivery pump in the MUT research plant. The purpose of this research was to control and monitor the research plant with no physical connection between the monitoring station (programing unit) and the control station (PLC), by using the wireless communication network and to make assumptions for introducing wireless communication between the PLC and field devices. This chapter illustrates the layout of the existing network and the integration of the wireless devices.

5.2 The background of the existing plant

The existing research plant was built by the students at MUT as the initiative that was led by Prof P. Naidoo. Since the completion of the research plant it has been used for demonstration of process automation and carrying out experiments for educational purposes. The plant layout is illustrated in Figures 1.1 and 1.2. The research plant mixes the contents of the two additive tanks in the blend chest. The supply valves SV1 and SV2 allow the fluids into the additive tanks, tank1 and tank 2 respectively while the level sensors LLS1, LLS2, HLS1 and HLS2 measure the low levels and high levels in the additive tanks. The variable speed pumps, VSP1 and VSP2 pump the contents of the additive tanks into the blend chest for mixing. The speed of the variable speed pumps can be varied according to the ratio of the additives required in the blend chest while the flowmeters FT1 and FT2 measure the rate of flow from each pump and send it to the control device for observation and control. The level transmitter, LT1, on the blend chest is the transducer that measures the pressure of the fluid in the blend chest and converts it to proportional electrical impulses and sends these impulses to the control unit for monitoring and controlling the operation of the plant. The delivery pump on the blend chest is used to pump the end product away from the blend chest, but at the moment it is still used for experimental purposes to pump the fluid back to the additive tanks. After completion of the hardware configuration these components were controlled and monitored from the programing unit.

5.3 Integrating the wireless communication devices

The block diagram of the electrical connections of the plant layout is illustrated in figure 5.1 where the level transmitter is connected to the PROFIBUS DP network by the PROFIBUS PA cable via the PROFIBUS DP/PA coupler which interfaces between the two technologies. The variable speed pumps are connected to the variable speed drives while the delivery pump is the Boolean device and it connects direct to the controller. The PRFOFIBUS DP cable connects to the rest of the field devices to the control station, the PLC. The PLC is connected to the programing unit via the multipoint interface (MPI) cable.

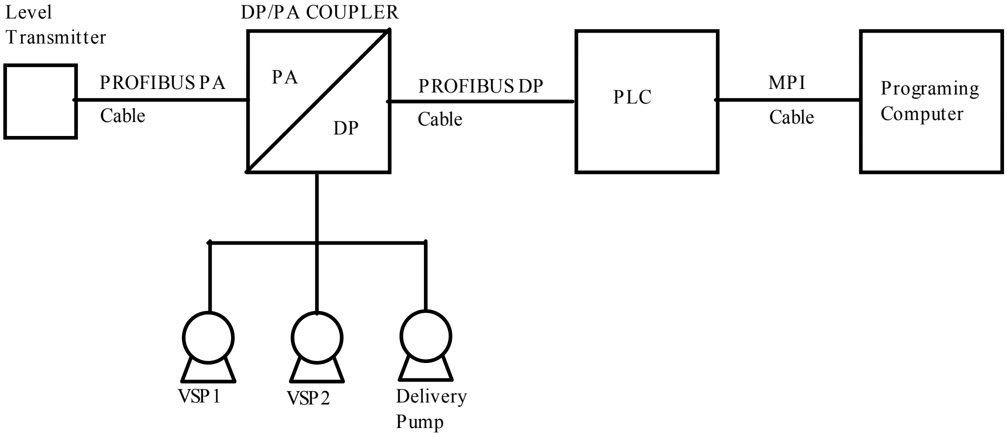


Figure 5.1: Electrical connections of the existing plant

The integration of the wireless communication devices to the existing network replaced the MPI cable in Figure 5.1 and made room for wireless communication between the programing unit and the PLC. The wireless network was integrated to the existing research plant by inserting the Siemens SIMATIC NET, CP 343 lean communication processor for connecting SIMATIC S7-300 to Industrial Ethernet via TCP/IP, UDP and multicast. Because the PLC exchanges data with other devices by means of datagrams, this communication processor card interfaces between the PLC and other wireless devices. The integrated block diagram is indicated in Figure 5.2.

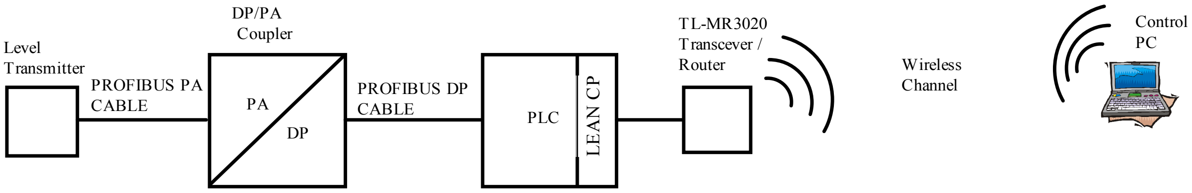


Figure 5.2: Insertion the Siemens communication processor to the PLC

Figure 5.2 illustrates the insertion of the Siemens communication processor to the PLC and the connection of the suitable wireless device to communicate data between the PLC and the programing unit over the wireless communication channel and it makes the wireless device transparent to the PLC system, that means the PLC will still behave as if it is connected to the original MPI cable network without any interference. The wireless device was used because it has the built in wireless interface and no additional circuitry was required for it to be integrated into the network.

5.4 Selecting the suitable wireless transceiver

The hardware design suitable to implement the wireless communication was supposed to meet the specifications stipulated in IEC 61158-2 standard [16] and IEEE 802.11 standard [32, 33]. Some of these specifications are:

(i)	Interface	: RS-485 / UART
(ii)	Modulation Type	: QPSK, QAM, PSK
(iii)	Data transmission rate	: IEEE 802.11b
(iv)	Maximum power	: 15-20 dBm
(v)	Maximum current	: 300 mA

Various transceiver modules were analysed in order to select the most suitable module. The various transceiver modules studied include:

(i) WSN802G transceiver module made by RFM which is a robust solution for sensor networks. The module includes analog, digital, serial and SPI I/O providing versatility needed to serve a wide range of sensor network applications [21].

(ii) nRF24L01 transceiver module by Nordic Semiconductor with an embedded baseband protocol engine operating at 2.4 – 2.4385 GHz band using serial peripheral interface (SPI) [22].

(iii) HM-TR transparent wireless transceiver module by Hope Microelectronics with high data rate and long transmission distance and self-controlled communication protocol completely transparent to the user interface [23].

(iv) LT2510 transceiver module by Laird Technologies which is embedded with server-client protocol permitting an unlimited number of clients to synchronize to a single server for low latency communication [24].

(v) LT series transceiver module by Linx Technologies which is ideal for bidirectional wireless transfer of serial data in the 260-470 MHz band [25]

(vi) MRF89XAM PICTail module by Microchip designed for experimental demonstrations on the explorer development board [26].

(vii) YS-1020UA transceiver module by Yishi Electronics which is designed for professional wireless data transmission systems on ISM frequency band with half duplex receiving and transmitting and can connect directly with processor, PC, RS485 devices, UART components with RS232 and TTL [27].

(viii) TL-MR3020 transceiver module by TP-LINK Technologies which is based on the Wireless N network giving the freedom to set up a stable and high speed network [33]. This transceiver uses the 802.11 standard for mobile wireless networking.

All these transceiver modules share common characteristics as listed in table 5.1 for comparison and selection of the most suitable device for the application. Although the YS-1020L has a unique transmission technology, RS-485 interface, that is employed by PROFIBUS, its half-duplex characteristic renders it unsuitable because the PLC is sensitive to latency. The TL-MR3020 transceiver, on the other hand, employs the MIPS 2k technology which makes it transparent to the PLC and therefore suitable for this application. Figure 5.3 shows the TL-MR3020 data transceiver. The TL-MR3020 employs the AR9331 chip which has built in wireless interface and it can be configured as the transceiver for standard two way communications or as the router packet transmission wireless networks which made it more suitable for this research task.

The TL-MR3020 module manufactured by TP-LINK is illustrated in Figure 5.3 and it is used for its superior properties as illustrated in Table 5.1.

Figure 5.3: The TL-MR3020 transceiver module [33]

Table 5.1: Characteristics of various data transceivers

	WSN802G	nRF24L01	HM-TR	LT2510	LT Series	YS-1020U	TL-MR3020
Interface	SPI	SPI	UART, TTL, RS232	UART	Direct serial	TTL, RS232, RS485	SPI, I^{2}S, UART
Frequency range	2401 to 2474 MHz	2.4 to 2.335 GHz	310.24 to 929.27 MHz	2400 to 2483.5 MHz	260 to 470 MHz	433 / 450 / 868 MHz	2.4 GHz
Modulation	QPSK	GFSK	FSK	PWM	AM and OOK	GFSK	BPSK, QPSK, QAM
Data rate	1 to 11 Mbps	1 to 2 Mbps	Wide range	280 to 500 kbps	Up to 10000 bps	1200 to 38400 bps	150 Mbps
RF Power	10 mW	4 dBm	Frequency dependent	+30 dBm	4 dBm	10 dBm	16 – 18 dBm
Voltage range	3 to 3.63 Vdc	-3 to 3.6 Vdc	5 Vdc	3.3 Vdc	-0.3 to 4 Vdc	3.3 to 5 Vdc	-0.3 to 4.0 V
Temperature range	-40 to 85 °C	-40 to 85 °C	-35 to 80 °C	-40 to 85 °C	-40 to 85 °C	-35 to 75 °C	-65 to 150 ºC
Transmission mode	Bidirectional	Bidirectional	Half duplex	Bidirectional	Bidirectional	Half duplex	Bidirectional

In digital systems the universal asynchronous receiver/transmitter (UART) is defined as the technology for translating data between parallel and serial form in the transceiver module while the serial peripheral interface (SPI) is a synchronous serial data link standard in which data is shifted one bit at a time. This device supports the BPSK, QPSK and QAM digital modulation schemes which make it suitable to handle packet data transmission, thereby making it suitable to transmit datagrams used by the PLC without any additional interface circuitry. Because the TL-MR3020 employs the AR9331 chip which has built in wireless interface, it can instantly establish a wireless network and yet remain transparent to the existing network in order to efficiently enable various connections to the existing network without affecting the functionality of the network [32, 33]. From Table 5.1, and the properties discussed, the TL-MR3020 was therefore an obvious choice for the implementation of the wireless communication. The functional block of the AR9331 chip is illustrated in Figure 5.4. The integrated CPU of the AR9331 is of the MIPS 24K processor family and can run up to 4MHz and has internal RAM [33]. The digital physical layer (PHY) block meets the 802.11n wireless networking standard and provides the transmit and receive path for wireless communication. It is essentially a half-duplex orthogonal frequency division multiplexing (OFDM) encoding scheme [33]. In communication OFDM is described as a form of modulation that divides a data rate modulating stream and place them on many narrow band subcarriers. This technique makes data less sensitive to frequency fading. This is critical because some frames of data are used for signal detection, automatic gain control, symbol timing and channel estimation [33]. There is an integrated radio block and RF front end. In microwave technology an RF front end refers to all the wireless communication circuitry from the antenna up to the mixer. In the AR9331 chip the radio block consists of the transmit (Tx) chain and the receive (Rx) chain

[33]. The transmitter converts the baseband signal to 2.4 GHz RF for transmission using integrated up-conversion architecture comprising of mixers and power amplifiers and send it to the antenna using an RF switch [33]. The receiver converts the received 2.4 GHz RF signal to baseband implementing direct conversion architecture comprising of amplifiers, mixers and low pass filters [33]. The TL-MR3020 manufactured by TP-LINK is a complete wireless system implementing the IEEE 802.11n/g/b and IEEE 802.3/3u standards. The 802.11n standard employs multiple-inputs multiple-outputs transmitter and receiver antennas to allow for increased data throughput [32]. The 802.11g standard is a specification for 54 megabits per second (MBPs) using OFDM and operating in an unlicensed radio band of 2.4 GHz and using encryption for transmission [32] while 802.11b specifies wireless networking at 11 MBPs. The 802.3/3u specification for Ethernet is a method of connection in a network [32]. The compatibility of the AR9331 with these standards make the TL-MR3020 module versatile in that it can be configured as transceiver, Ethernet node or a complete Wi-Fi system on its own.

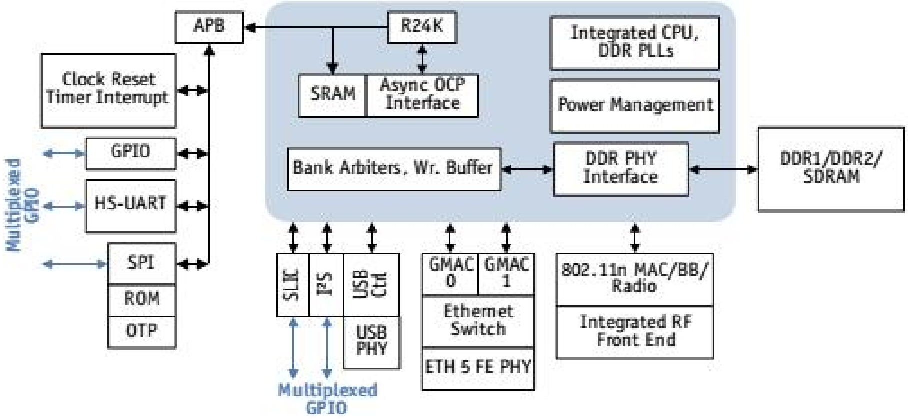


Figure 5.4: The functional block of the AR9331 chip [33]

The wireless module was then integrated into the system as illustrated in Figure 5.5 and the different components are indicated.

The communication processor card, Lean card, in Figure 5.6 was connected to the PLC via the back plane and it has a slot for a cable to network with the wireless device. The wireless device, TL-MR3020 module, has a separate power source and it interfaces the PLC with the wireless devices. The TL-MR3020 was then connected to the internet and its software was downloaded from the manufacturer's website and the AP mode was configured.

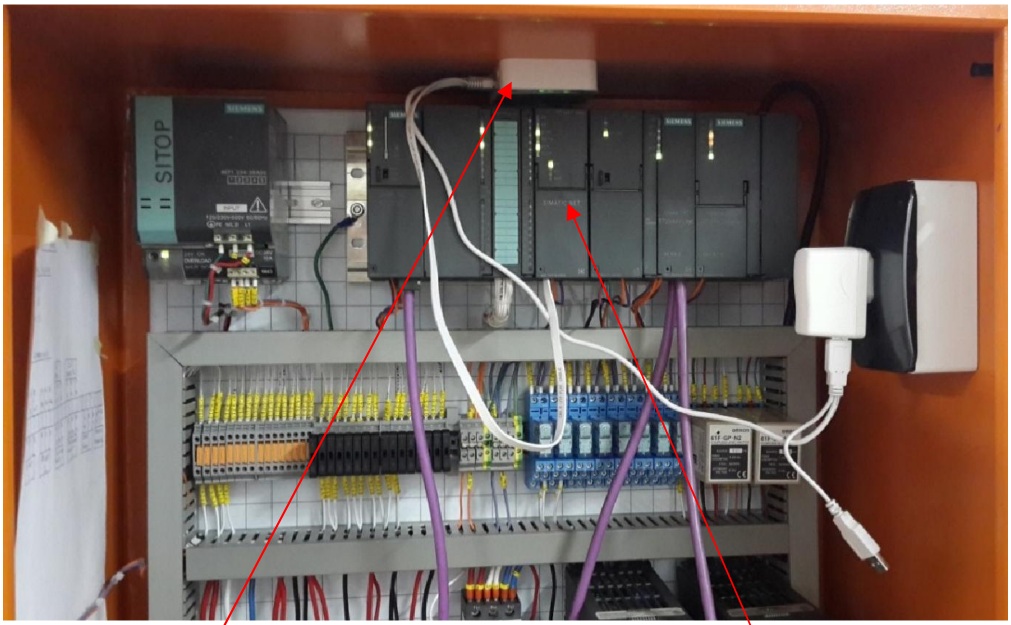


Figure 5.5: Integration of wireless communication devices into existing network

In AP mode the module can act as a wireless access point supporting four modes, Access Point mode, Repeater mode, Bridge mode and Client mode [32]:

(i) In Access Point mode the module acts as a wireless central hub for wireless LAN clients, giving a wireless extension for current wired network.

(ii) In repeater mode the module extends the coverage of another wireless Access Point.

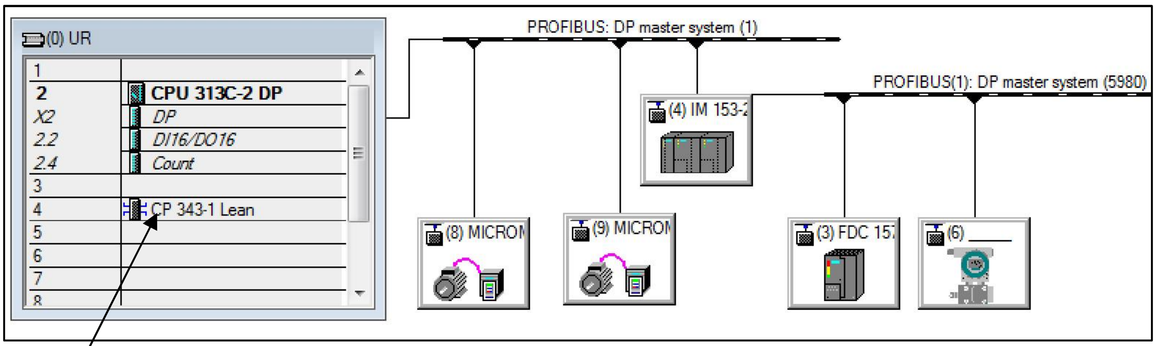
(iii) In Bridge mode the module wirelessly connects two remote LANs together and

(iv) In Client mode the module acts as a wireless card to connect with wireless network [32].

The Access Point mode was therefore selected so that the module can provide wireless extension for the existing wired network. After the insertion of the wireless devices, Lean card and TL-MR3020 module, the network was then configured and tested.

5.5 Wireless network configuration

While the programing computer was still connected to the PLC via the MPI cable, the Siemens communication processor card was installed on the rack and connected to the PLC via the backplane and then its GSD files were installed using the programing unit. The hardware configuration on the programing unit was updated and the CP-343-1 Lean card was inserted in address 4 on the SIMATIC project and it was linked to S7-300 CPU as illustrated in Figure 5.6.



(0) UR

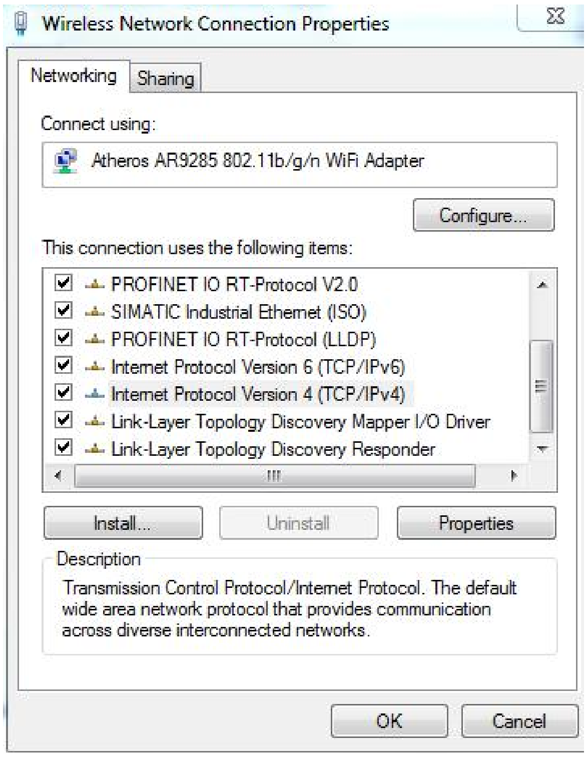
1	
2	CPU 313C-2 DP
X2	DP
2.2	DI16/DO16
2.4	Count
3	
4	CP 343-1 Lean
5	
6	
7	
8	

PROFIBUS: DP master system (1)
PROFIBUS(1): DP master system (5980)
(4) IM 153-2
(8) MICROM
(9) MICROM
(3) FDC 157
(6)
Lean card

Figure 5.6: Integration of Lean card to SIMATIC project

The project was then compiled and downloaded to the PLC and then configured for wireless operation by selecting the network button in the programing unit and creating a new network. The new network was named wilson-WIFI and the Internet Protocol Version (TC/IPv4) was configured. In Networking IPv4 is defined as a connectionless protocol used in packet switching to provide logical connection of devices by providing identification of each device known as IP address. The IP address is a 32 bit number used to uniquely identify the device in the network, it is represented by four numbers separated by a dot. Each number can range from 0 to 255. For private network the numbers can be allocated at random. The PLC was then allocated the IP address: 192.168.1.100 and the wireless module was allocated the IP address: 192.168.1.251. In this case the first three numbers identify the network and the last number identifies the device connected in the wireless network. The network was then activated and the PLC communicated with the TL-MR3020 and the connection was verified by indicating the device name as illustrated in Figure 5.7 as: Atheros AR9285 802.11b/g/n WiFi adapter. This verifies that the AR9331 chip is IEEE802.11b/g/n compliant as previously explained.

The programing computer was then configured to transfer the wireless operation to the PLC by setting the PG/PC interface in the control panel. The PG/PC interface is the programing device that is used to program the Siemens PLCs, it contains the commissioning software and it is installed on the computer when installing the Siemens software. The device name for the wireless module appeared on the PG/PC interface and it was activated to transfer the communication with the PLC from MPI cable to wireless.



Wireless Network Connection Properties

Connect using:

Atheros AR9285 802.11b/g/n WiFi Adapter

This connection uses the following items:

- ☑ PROFINET IO RT-Protocol V2.0
- ☑ SIMATIC Industrial Ethernet (ISO)
- ☑ PROFINET IO RT-Protocol (LLDP)
- ☑ Internet Protocol Version 6 (TCP/IPv6)
- ☑ Internet Protocol Version 4 (TCP/IPv4)
- ☑ Link-Layer Topology Discovery Mapper I/O Driver
- ☑ Link-Layer Topology Discovery Responder

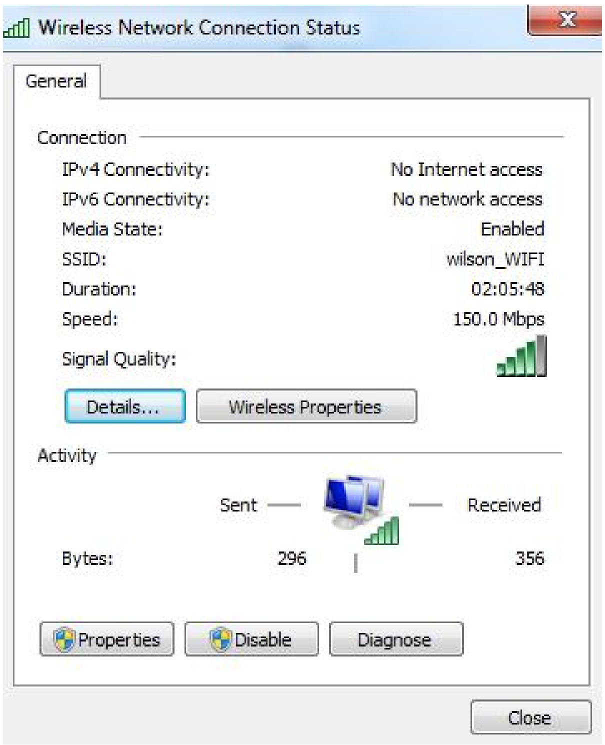
Description

Transmission Control Protocol/Internet Protocol. The default wide area network protocol that provides communication across diverse interconnected networks.

Figure 5.7: Wireless network connection

The laptop with wireless facility was then configured to take the place for the programing unit and communicated with the PLC via the wireless module. The new programing unit was allocated the IP address: 192.168.1.10 and the communication was establish with the PLC via the wireless module and the connection was confirmed as illustrated in the wireless network connection status of Figure 5.8.

The wireless communication was then tested using the Ping utility from the command prompt. The Ping utility is used in networking to check the reachability of a distant station, and test how long it takes to send and receive data between the two stations. The testing of the PLC yielded the response in Figure 5.9 where the IP address of the PLC is indicated from the programing unit. In Figure 5.9 it is illustrated that 32 bytes of data were sent to the PLC and it sent back 4 packets of 32 bytes. The first packet took 8 ms to receive, the second packet took 4 ms, the third packet took 4 ms and fourth packet took 3 ms. The first packet took long because it included the time to send and receive, but the average transmission time was 4 ms



Connection

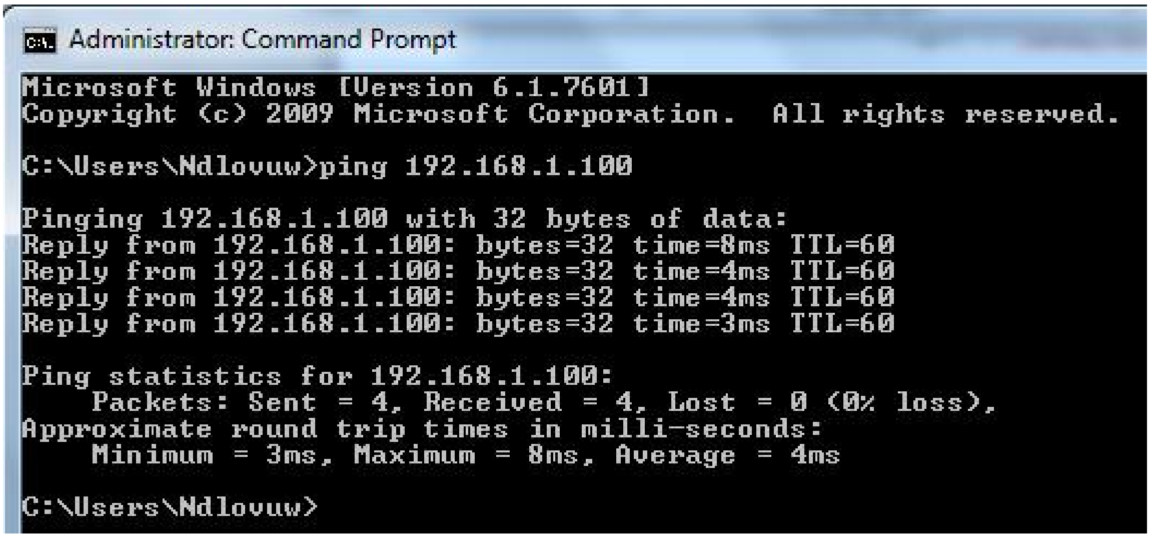
IPv4 Connectivity: No Internet access
IPv6 Connectivity: No network access
Media State: Enabled
SSID: wilson_WIFI
Duration: 02:05:48
Speed: 150.0 Mbps
Signal Quality:

Activity

Sent — Received

Bytes: 296 | 356

Figure 5.8: wilson_WIFI wireless network connection status

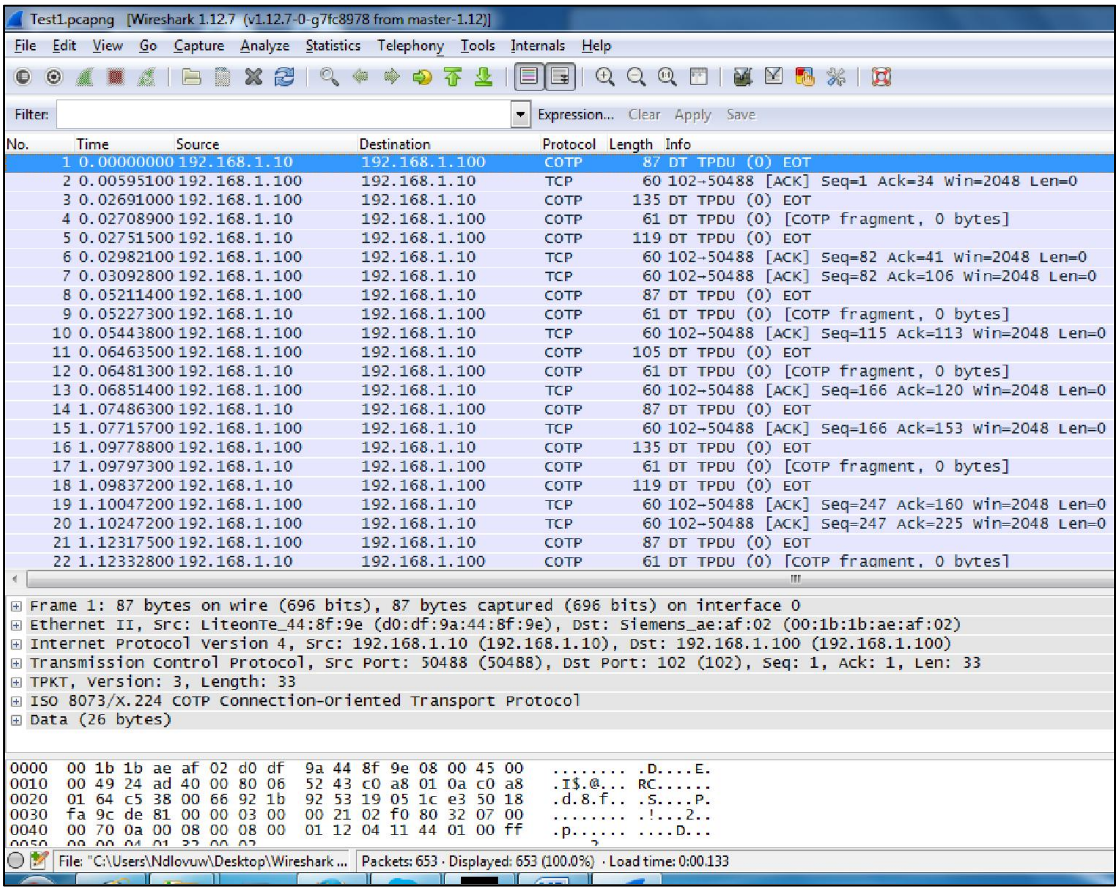


Administrator: Command Prompt

```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\Ndlovuw>ping 192.168.1.100

Pinging 192.168.1.100 with 32 bytes of data:
Reply from 192.168.1.100: bytes=32 time=8ms TTL=60
Reply from 192.168.1.100: bytes=32 time=4ms TTL=60
Reply from 192.168.1.100: bytes=32 time=4ms TTL=60
Reply from 192.168.1.100: bytes=32 time=3ms TTL=60

Ping statistics for 192.168.1.100:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 3ms, Maximum = 8ms, Average = 4ms

C:\Users\Ndlovuw>
```

Figure 5.9: Ping utility testing reachability of the PLC

After successful communication was established, the program was then downloaded from the laptop programing unit to the PLC and control and monitoring of the research plant was done over the wireless communication link. The functionality was tested and evaluated by controlling the pumps from the same variable table as when using the MPI cable. Since the network was wireless transmitting data in packets format, it was necessary to use the network analyser to capture the packets for different operations and analyse the performance of the network. The Wireshark network analyser was downloaded and used to analyse the network and the results were recorded in Table 5.2. The example of the Wireshark capture is illustrated in Figure 5.10. The Wireshark capture contains information about the data sent and received over a specified time of the observation. The first column displays the time when the packet was captured. The second column displays the IP address of the source of data, in this case 192.168.1.10 if the packet is from the programing unit. The third column displays the IP address for the data destination, in this case 192.168.1.100 if the packet was going to the PLC. The fourth column displays the protocol. In this network only two protocols were detected, the COTP and TCP. In Networking and Information Theory, the Connection Oriented Transmission Protocol (COPT) is used to transport packets of data from one user without altering the boundaries set by the transmitter. The Transmission Control Protocol (TCP) transports a stream of data to the receiver and another protocol is required to add the boundaries. The latest Networking strategies, however, have discontinued the use of COTP and instead couple another protocol on top of TCP. The fifth column displays the length of the packet while the sixth column displays the information about the packet such as end of transmission (EOT) and acknowledgement (ACK). The Wireshark network analyser also displays the frame information and the contents of the frame.

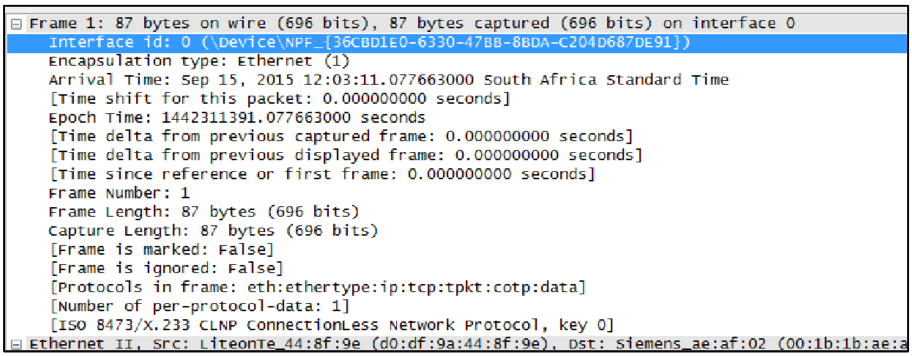


```
No.   Time       Source        Destination    Protocol Length Info
    1 0.00000000 192.168.1.10  192.168.1.100  COTP       87 DT TPDU (0) EOT
    2 0.00595100 192.168.1.100 192.168.1.10   TCP        60 102→50488 [ACK] Seq=1 Ack=34 Win=2048 Len=0
    3 0.02691000 192.168.1.100 192.168.1.10   COTP      135 DT TPDU (0) EOT
    4 0.02708900 192.168.1.10  192.168.1.100  COTP       61 DT TPDU (0) [COTP fragment, 0 bytes]
    5 0.02751500 192.168.1.10  192.168.1.100  COTP      119 DT TPDU (0) EOT
    6 0.02982100 192.168.1.100 192.168.1.10   TCP        60 102→50488 [ACK] Seq=82 Ack=41 Win=2048 Len=0
    7 0.03092800 192.168.1.100 192.168.1.10   TCP        60 102→50488 [ACK] Seq=82 Ack=106 Win=2048 Len=0
    8 0.05211400 192.168.1.100 192.168.1.10   COTP       87 DT TPDU (0) EOT
    9 0.05227300 192.168.1.10  192.168.1.100  COTP       61 DT TPDU (0) [COTP fragment, 0 bytes]
   10 0.05443800 192.168.1.100 192.168.1.10   TCP        60 102→50488 [ACK] Seq=115 Ack=113 Win=2048 Len=0
   11 0.06463500 192.168.1.100 192.168.1.10   COTP      105 DT TPDU (0) EOT
   12 0.06481300 192.168.1.10  192.168.1.100  COTP       61 DT TPDU (0) [COTP fragment, 0 bytes]
   13 0.06851400 192.168.1.100 192.168.1.10   TCP        60 102→50488 [ACK] Seq=166 Ack=120 Win=2048 Len=0
   14 1.07486300 192.168.1.10  192.168.1.100  COTP       87 DT TPDU (0) EOT
   15 1.07715700 192.168.1.100 192.168.1.10   TCP        60 102→50488 [ACK] Seq=166 Ack=153 Win=2048 Len=0
   16 1.09778800 192.168.1.100 192.168.1.10   COTP      135 DT TPDU (0) EOT
   17 1.09797300 192.168.1.10  192.168.1.100  COTP       61 DT TPDU (0) [COTP fragment, 0 bytes]
   18 1.09837200 192.168.1.10  192.168.1.100  COTP      119 DT TPDU (0) EOT
   19 1.10047200 192.168.1.100 192.168.1.10   TCP        60 102→50488 [ACK] Seq=247 Ack=160 Win=2048 Len=0
   20 1.10247200 192.168.1.100 192.168.1.10   TCP        60 102→50488 [ACK] Seq=247 Ack=225 Win=2048 Len=0
   21 1.12317500 192.168.1.100 192.168.1.10   COTP       87 DT TPDU (0) EOT
   22 1.12332800 192.168.1.10  192.168.1.100  COTP       61 DT TPDU (0) [COTP fragment, 0 bytes]

⊞ Frame 1: 87 bytes on wire (696 bits), 87 bytes captured (696 bits) on interface 0
⊞ Ethernet II, Src: LiteonTe_44:8f:9e (d0:df:9a:44:8f:9e), Dst: Siemens_ae:af:02 (00:1b:1b:ae:af:02)
⊞ Internet Protocol Version 4, Src: 192.168.1.10 (192.168.1.10), Dst: 192.168.1.100 (192.168.1.100)
⊞ Transmission Control Protocol, Src Port: 50488 (50488), Dst Port: 102 (102), Seq: 1, Ack: 1, Len: 33
⊞ TPKT, Version: 3, Length: 33
⊞ ISO 8073/X.224 COTP Connection-Oriented Transport Protocol
⊞ Data (26 bytes)

0000  00 1b 1b ae af 02 d0 df  9a 44 8f 9e 08 00 45 00   ........ .D....E.
0010  00 49 24 ad 40 00 80 06  52 43 c0 a8 01 0a c0 a8   .I$.@... RC......
0020  01 64 c5 38 00 66 92 1b  92 53 19 05 1c e3 50 18   .d.8.f.. .S....P.
0030  fa 9c de 81 00 00 03 00  00 21 02 f0 80 32 07 00   ........ .!...2..
0040  00 70 0a 00 08 00 08 00  01 12 04 11 44 01 00 ff   .p...... ....D...
0050  00 00 04 01 32 00 02
```

File: "C:\Users\Ndlovuw\Desktop\Wireshark ... Packets: 653 · Displayed: 653 (100.0%) · Load time: 0:00.133

Figure 5.10: Wireshark capture

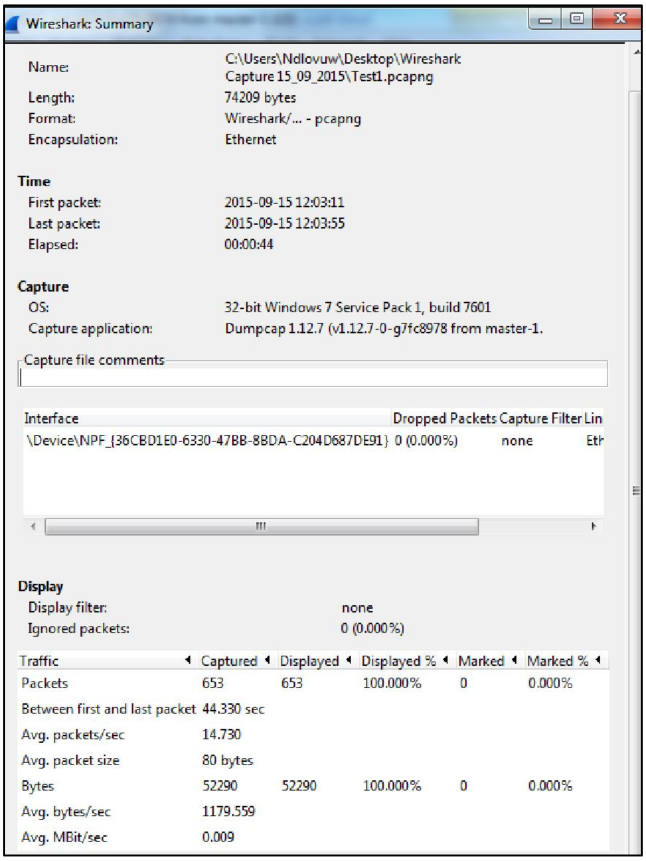
The information displayed about the frame includes the size in bytes, the type of network, the protocol and etc. The contents of the frame can be observed by clicking on the frame while the structure of the frame can be observed by clicking the + sign in front of the frame as shown in Figure 5.11.



```
⊟ Frame 1: 87 bytes on wire (696 bits), 87 bytes captured (696 bits) on interface 0
    Interface id: 0 (\Device\NPF_{36CBD1E0-6330-47BB-8BDA-C204D687DE91})
    Encapsulation type: Ethernet (1)
    Arrival Time: Sep 15, 2015 12:03:11.077663000 South Africa Standard Time
    [Time shift for this packet: 0.000000000 seconds]
    Epoch Time: 1442311391.077663000 seconds
    [Time delta from previous captured frame: 0.000000000 seconds]
    [Time delta from previous displayed frame: 0.000000000 seconds]
    [Time since reference or first frame: 0.000000000 seconds]
    Frame Number: 1
    Frame Length: 87 bytes (696 bits)
    Capture Length: 87 bytes (696 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: eth:ethertype:ip:tcp:tpkt:cotp:data]
    [Number of per-protocol-data: 1]
    [ISO 8473/X.233 CLNP ConnectionLess Network Protocol, key 0]
⊟ Ethernet II, Src: LiteonTe_44:8f:9e (d0:df:9a:44:8f:9e), Dst: Siemens_ae:af:02 (00:1b:1b:ae:a
```

Figure 5.11: Wireshark frame contents

The Wireshark network analyser also has the Analyser tool that can filter data of the same protocol, but it was not used in this analysis because the data transmitted was of the same type, however, the statistics tool was useful in summarising each test and the statistics summary for Test 1 is illustrated in Figure 5.12.



Wireshark: Summary

Name:	C:\Users\Ndlovuw\Desktop\Wireshark Capture 15_09_2015\Test1.pcapng
Length:	74209 bytes
Format:	Wireshark/... - pcapng
Encapsulation:	Ethernet
Time	
First packet:	2015-09-15 12:03:11
Last packet:	2015-09-15 12:03:55
Elapsed:	00:00:44
Capture	
OS:	32-bit Windows 7 Service Pack 1, build 7601
Capture application:	Dumpcap 1.12.7 (v1.12.7-0-g7fc8978 from master-1.

Capture file comments

Interface	Dropped Packets	Capture Filter	Lin
\Device\NPF_{36CBD1E0-6330-47BB-8BDA-C204D687DE91}	0 (0.000%)	none	Eth

Display	
Display filter:	none
Ignored packets:	0 (0.000%)

Traffic	Captured	Displayed	Displayed %	Marked	Marked %
Packets	653	653	100.000%	0	0.000%
Between first and last packet	44.330 sec				
Avg. packets/sec	14.730				
Avg. packet size	80 bytes				
Bytes	52290	52290	100.000%	0	0.000%
Avg. bytes/sec	1179.559				
Avg. MBit/sec	0.009				

Figure 5.12: Wireshark statistics summary

The Wireshark statistics summary of Figure 5.12 displays the statistics of packet capturing and this is the information displayed in Table 5.2 for all the tests. Column two in Table 5.2 displays the statistical summary for VSP1. Column three displays the statistical summary for VSP2. Column four displays the summary for delivery pump. Column five displays summary for VSP1+VSP2 while column six displays summary for when all the pumps are running. The values plotted for VSP1, VSP2, Delivery Pump, VSP1+VSP2 and all pumps in Table 5.2 were obtained from the Wireshark captures illustrated in appendix C1 to C5.

Table 5.2: Test results for wireless network

	VSP1	VSP2	Delivery Pump	VSP1 + VSP2	All Pumps
Packets captured	1568	1238	1346	3101	2898
Time elapsed between first and last packet	10.287sec	8.255sec	9.003sec	20.823sec	19.333sec
Average packets/sec	152.432	149.974	149.498	149.924	149.899
Average packet size	77 bytes	77 bytes	77bytes	77 bytes	77 bytes
Bytes	120438	95017	103404	238427	222787
Average bytes/sec	11708.326	11510.529	11484.887	11450.374	11523.668
Average MBit/sec	0.094	0.092	0.092	0.092	0.092

Table 5.2 summarises the performance of the wireless network under various load conditions where the loads are different pump combinations running at the time of the test. These tests were carried out in order to measure the performance of the wireless network. The performance of the PROFIBUS network is recorded in Table 3.2. The performance of the PROFIBUS network and that of the wireless network are compared in Table 6.2 in order to validate the reliability of the wireless network to replace the PROFIBUS network.

5.6 Conclusion

This chapter discussed the integration of wireless communication devices into the existing plant. The performance of the wireless communication network was tested and verified against the performance of the existing plant. In the next chapter the simulation results are tested for reliability.

CHAPTER SIX: SIMULATION TEST AND ANALYSIS OF TEST RESULTS

6.1 Introduction

The aim of the research was to model and implement in real time the wireless communication between the programing unit and the PLC in a research plant at MUT and make an assumption for implementing wireless communication between the PLC and some field devices and generalise the use of wireless communication of Process and Control equipment in an industrial setting within the work floor. This was achieved by using Matlab and Simulink for modelling the digital communication channel and by using the TL-MR3020 wireless module for real time implementation of wireless communication between the programing unit and the PLC.

6.2 Modelling using Matlab and Simulink

After the evaluation and analysis of various wireless communication devices, the suitable one was found to support BPSK, QPSK and 16-QAM digital modulation schemes and therefore the necessity to model these schemes in Matlab and Simulink was established. The Simulink models consisted of data source, transmitter, AWGN channel, receiver and error calculation and display. The AWGN channel was selected as the suitable medium to simulate the noisy transmission channel because it injects white Gaussian noise to the signal passing through it. The performance of the wireless channel is better measured by its ability to transmit and receive under noisy conditions and this was realized by using the bit error rate (BER) calculator and plotting the test results for various modulation schemes simulated under test. The results of the various schemes tested are summarised in Table 6.1. The values in Table 6.1 were recorded for one-bit per symbol for BPSK, two-bits per symbol for QPSK and four-bits per symbol for 16-QAM. Because of the discrepancy in number of bits per symbol, for each modulation scheme the Eb/No ratio where the BER was 0 was first determined in order to determine the range of the Eb/No range for testing the model and recording the BER. For each modulation scheme the simulated values were plotted and compared to the theoretical plot obtained by using the BERTool and Monte Carlo Simulink simulation. The simulated plots are illustrated in Figures 6.1, 6.2 and 6.3 for comparison, but each plot was closely related to the theoretical plot of its format.

Table 6.1: Bit Error Rate (BER) for BPSK, QPSK and 16-QAM

BPSK		QPSK		16-QAM	
Eb/No	BER	Eb/No	BER	Eb/No	BER
-10	0.1876	-2	0.3778	-10	0.4925
-9	0.165	-1	0.3437	-9	0.4176
-8	0.1164	0	0.307	-8	0.3447
-7	0.09001	1	0.2498	-7	0.2737
-6	0.07704	2	0.1988	-6	0.2238
-5	0.05233	3	0.1548	-5	0.1688
-4	0.03743	4	0.1189	-4	0.1209
-3	0.02458	5	0.08192	-3	0.07792
-2	0.01334	6	0.04995	-2	0.04196
-1	0.006799	7	0.02797	-1	0.02298
0	0.002	8	0.01499	0	0.008991
1	0.0007999	9	0.003996	1	0.003996
2	1.00E-04	10	0.002997	2	0.000999
3	0	11	0	3	0

It is evident from Table 6.1 that for BPSK the Eb/No ratio had to be reduced to obtain the comparable performance with the other modulation schemes, therefore, if the number of bits per symbol is increased, the BER increased resulting in the decrease in system performance. Although there is discrepancy in the performance of the different schemes, it can still be deduced from Table 6.1 that the signal power to noise power ratio (Eb/No) was well within the acceptable limits. There were 0 errors at a ratio of 3 dB for BPSK and 16-QAM while there were 0 errors at a ratio of 11 dB for QPSK. This means that BPSK and 16-QAM perform better than QPSK under noisy conditions. According to Digital Communication and Data Communication the wireless access systems operate at a signal to noise ratio of ± 30 dBm closer to the source.

In Communication and Radio Electronics the bit error rate (BER) is estimated using the ratio of the bits received or processed in error to the total number of bits transmitted over the channel and is expressed as:

$$BER = \frac{Number\ of\ bits\ in\ error}{Total\ number\ of\ bits\ sent} \tag{3}$$

Equation 3 is used to estimate the probability of error due to signal to noise ratio (SNR). In the plot the signal to noise ratio is represented by E_b/N_o, where E_b represents the energy of each bit and N_o represents the noise power. The plots of the simulated results for the channel model is illustrated in Figure 6.1 for BPSK, Figure 6.2 for QPSK and Figure 6.3 for 16-QAM modulation schemes.

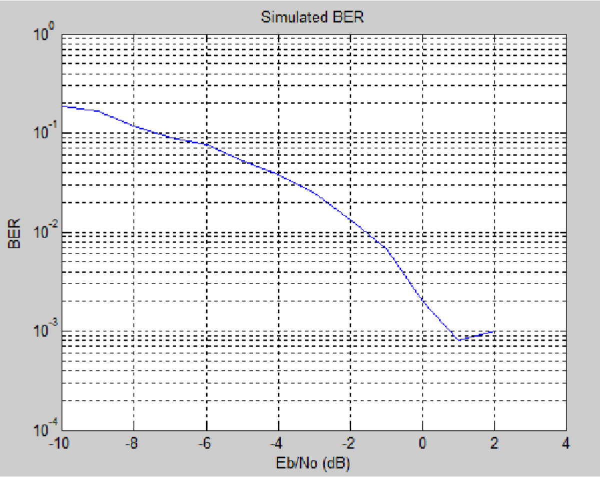


Figure 6.1: BER for BPSK

Figure 6.1 is the BER plot for BPSK modulation. A logarithmic scale was used in the BER axis to accommodate the very small values of the ratio but the plot verifies the values listed in Table 6.1. The tail in the range of 1 to 2 dB is due to the very low error rate that the difference is almost non-existent. Figure 6.2 illustrates the BER plot for QPSK modulation.

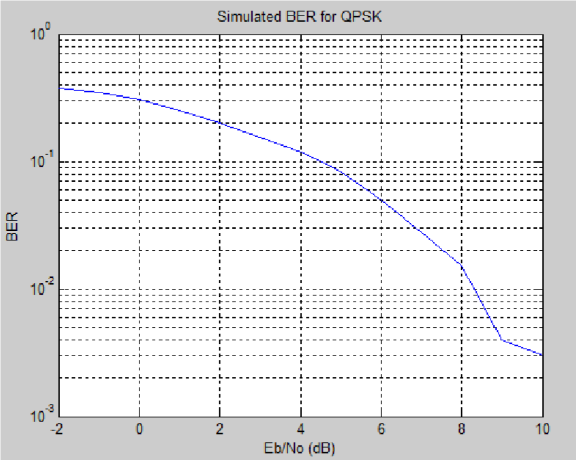


Figure 6.2: BER for QPSK

Close comparison of the Eb/No axes of Figures 6.1 and 6.2 illustrates that the ratio for the QPSK modulation has to be increased to obtain the noise performance similar to that of the BPSK modulation. Figure 6.3 illustrates the BER for the 16-QAM modulation.

When observing the Eb/No axis of Figure 6.3 it is again evident that the noise performance of the BPSK modulation is better than the noise performance of the 16-QAM modulation. These findings confirm that the noise performance of digital modulation schemes deteriorate when the number of bits per symbol is increased, however, it does not mean they should not be used.

There are techniques for minimising noise in the digital system and the schemes with high number of bits per symbol are useful in handling high volume of data to be transmitted per given time.

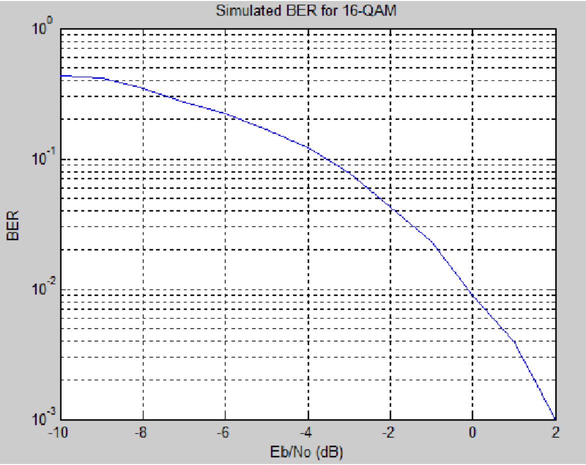


Figure 6.3: BER for 16-QAM

6.2.1 Analysis of simulation results

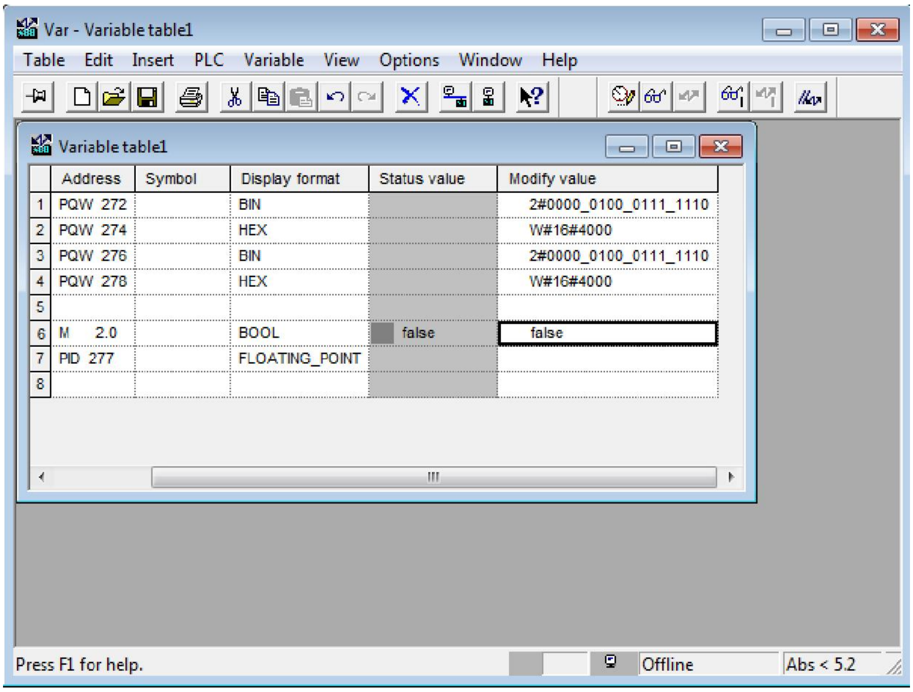
The AWGN channel was used to model BPSK, QPSK and 16-QAM digital modulation schemes because these are the modulation schemes employed by the target wireless device. The AWGN channel was used because it adds white noise with Gaussian distribution to the signal passing through it. The simulation results were recorded in a table and the bit error rate (BER) was plotted for each modulation scheme. These plots were then compared to analyse the noise performance for each modulation scheme. The analysis indicated that the BPSK performed better than the other schemes because it uses only one bit per symbol while the other schemes use more bits per symbol, however, the overall performance of all these schemes was generally good because the maximum Eb/No ratio tested was 11 dB while the real time wireless application operates at the Eb/No ratio of 30 dBm. Therefore these modulation schemes were suitable for the real time implementation of wireless communication.

6.3 Implementation of wireless communication

The aim of this research was to implement the wireless communication in real time in an industrial communication channel. This was achieved by implementing the wireless communication between the programming computer and the PLC in the research plant at MUT and assumptions were made to implement wireless communication between the PLC and some field devices.

6.3.1 Results for PROFIBUS network

The implementation was achieved by first configuring the existing PROFIBUS plant to control and monitor the field devices using the programming computer. The Step-7 project was created in the programing unit, then compiled and downloaded to the PLC via the MPI cable. The various components of the plant were allocated address to be used by the PLC to identify them. These addresses were also used to monitor and control the plant using the variable table in Figure 6.4.



	Address	Symbol	Display format	Status value	Modify value
1	PQW 272		BIN		2#0000_0100_0111_1110
2	PQW 274		HEX		W#16#4000
3	PQW 276		BIN		2#0000_0100_0111_1110
4	PQW 278		HEX		W#16#4000
5					
6	M 2.0		BOOL	false	false
7	PID 277		FLOATING_POINT		
8					

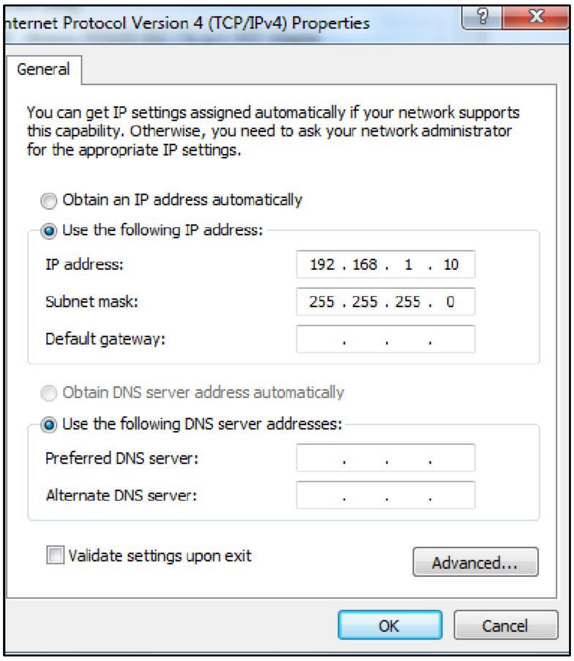
Press F1 for help.
Offline
Abs < 5.2

Figure 6.4: Variable table monitoring and controlling the plant

In Figure 6.4 the modify value is the control word, some call it status word, was used to control the various devices. The first bit on the right in the control word on line 1 and line 3 is the start stop bit for the VSP1 and VSP2 respectively. A “0” stops the pump and a “1” runs the pump. The delivery pump is a start stop device and is controlled by the Boolean word in line 6 where false stops the pump and true runs the pump. Line 7 displays the reading on the level transmitter. The modify values in line 2 and line 4 are the setpoints for the variable speed pumps. The controlling and monitoring of the plant using the variable table was an indication that the PROFIBUS network was fully functional and this functionality was used to validate the functionality of the wireless network. The results showing the functioning of the plant are illustrated in Table 6.2

6.3.2 Results for wireless network

The Institute of Electrical and Electronic Engineers (IEEE) and the International Electrotechnical Commission (IEC) laid standards and regulations governing the use of wireless communication devices in order to preserve the bandwidth and to prevent interference with commercial wireless business by those experimenting with wireless devices. For that reason the wireless module selected for this research was IEEE802.11b/g/n compliant. The wireless network was implemented using the experimental setup illustrated in Figure 6.5 to setup the remote programing unit for wireless communication with the PLC. The programing unit, laptop, was allocated the IP address: 192.168.10 in the established wilson_WIFI network.


nternet Protocol Version 4 (TCP/IPv4) Properties

General

You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.

○ Obtain an IP address automatically

● Use the following IP address:

IP address: 192 . 168 . 1 . 10

Subnet mask: 255 . 255 . 255 . 0

Default gateway: . . .

○ Obtain DNS server address automatically

● Use the following DNS server addresses:

Preferred DNS server: . . .

Alternate DNS server: . . .

□ Validate settings upon exit

Figure 6.5: IP address configuration

On setting up the wireless network parameters the programing unit was interfaced to the PLC via the TL-MR3020 module attached to the S7-300 PLC and the connection was validated using the Ping utility illustrated in Figure 5.9. The same program that was used to test the PROFIBUS network was then compiled and downloaded to the PLC via the wireless link. The same variable table illustrated in Figure 6.4 was then opened to control and monitor the plant wirelessly. To analyse data signals on Wireshark the following filtering protocols were available:

- IP: protocol enabling the transmission of packets between devices
- TCP: protocol ensuring that the packet of data is received in the same order in which it was sent

- COTP: protocol controlling the transportation of data between two points
- DATA: actual information shared between two points in the network

For this research, TCP and COTP were filtered in order to analyse the handling of packet data transmission over a wireless network. Five tests were run and the Wireshark network analyser was used to monitor the performance of the network and the results were summarised in Table 6.2. In test 1 VSP 1 was run and the Wireshark network analyser was used to capture the packets of data exchanged between the programing unit and the PLC. In test 2 VSP was run, in Test 3 the delivery pump was run, in Test 4 the VSP 1 and VSP 2 were run concurrently while in Test 5 all pumps were running. The results for Test 1 to Test 5 were captured and recorded in Table 6.2.

Table 6.2: Summary of PROFIBUS and wireless networks

		TEST 1	TEST 2	TEST 3	TEST 4	TEST 5
Bytes captured	PROFIBUS network	357240	647920	967890	1140096	887380
	Wireless network	120438	95017	103404	238427	222787
Average bytes/sec	PROFIBUS network	27480	14725	28467	21925	23352
	Wireless network	11708.326	11510.529	11484.887	11450.374	11523.668
Average MBit/sec	PROFIBUS network	0.22	0.12	0.22	0.18	0.19
	Wireless network	0.094	0.092	0.092	0.092	0.092

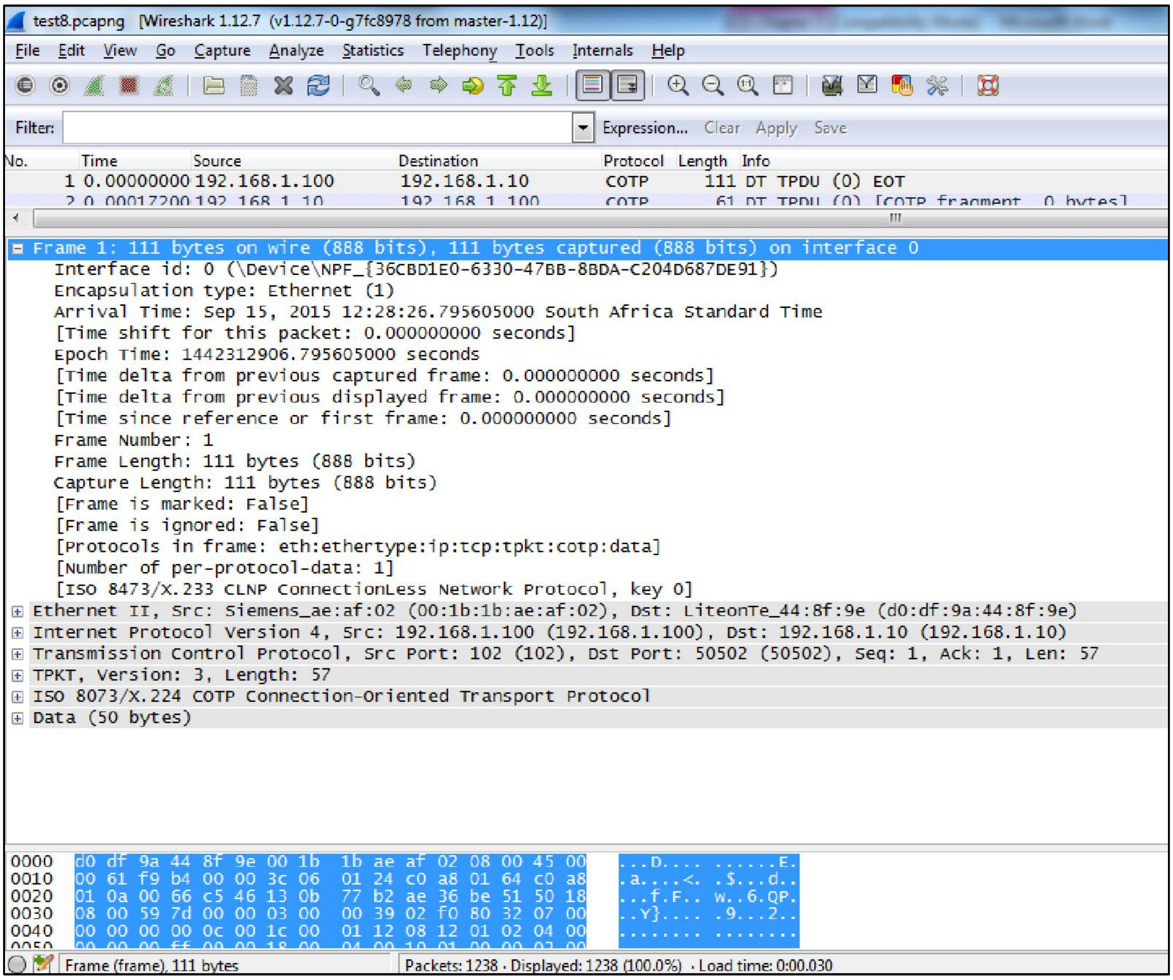
The amount of data sent and received for the PROFIBUS network was obtained from the Profitrace network analyser. The amount of data sent and received for the wireless network was obtained from the Wireshark network analyser. The other entries were obtained from the statistics summary of both analysers. The analysis of Table 6.2 illustrates that the wireless network performance was almost uniform under various test conditions. This assumption was validated by the average packet size and the average Mbit/sec information which is constant for all tests. On average the data volume of 92000 bits per second was passed between the programing unit and the PLC. No errors or lost data were recorded by the analyser. The performance analysis of the wireless network indicates that the network can be trusted to perform with a degree of reliability and can be regarded as successful. The information analysed was obtained from the Wireshark network analyser and the sample of the capture is illustrated in Figure 6.6 where breakdown of the frame of data is illustrated. In this frame 111 data bytes (888 bits) were captured and carried information about Interface id, Encapsulation type, Arrival time, Epoch time, Frame number, Frame length and Capture length. According to IEEE802, the Interface identifier is a 48 bit address. The number of packets sent and received was obtained from the Wireshark network analyser captured data by selecting Conversations from the Statistics menu. The other entries were obtained from the Statistics

Summary. The analysis of Table 6.2 illustrates that the wireless network performance was almost uniform under different test conditions. The discrepancy in the average values of the PROFIBUS network was due to the complex structure of the datagrams generated by the PLC, however, the communication processor matched the datagram structure to the packet structure. If the structure of the datagram captured with the Profitrace can be stripped of the additional information it can be shown that the results captured in Table 3.2 were similar to the results captured in Table 5.2. On average the data volume of 92000 bits per second was passed between the programing unit and the PLC. No errors or lost data were recorded by the analyser. Although the wireless network under test was handling less volume of data, the performance analysis indicates that the network can be trusted to perform with a degree of reliability and can be regarded as successful. The information analysed was obtained from the Wireshark network analyser and the sample of the capture is illustrated in Figure 6.6 where breakdown of the frame of data is illustrated. In this frame 111 data bytes (888 bits) were captured and carried information about Interface id, Encapsulation type, Arrival time, Epoch time, Frame number, Frame length and Capture length. According to IEEE802, the Interface identifier is a 48 bit address made up of 24 bit manufacturer ID and 24 bit of product ID. In Networking, encapsulation is the hiding of information within high level structures. The Wireshark network analyser capture also contains information about the header checksum that is used to protect the data packets against corruption. The Wireshark capture for this test procedure could not be used to generate BER because there was very little traffic and the data was almost of the same type.

6.4 Conclusion

Wireless communication was implemented in real time by integrating the wireless hardware devices in the existing PROFIBUS hard wired network. The wireless devices were the Siemens Communication Processor known as the Lean card and the TL-MR3020 wireless module. The Lean card was connected to the S7-300 PLC using the backplane and the TL-MR3020 was connected using the network cable. The GSD files for the Lean card were installed and the drivers for the wireless module were downloaded from the manufacturer's website and installed into the device and the AP operation was configured for the device to operate as the access point to the PLC. The new network (wilson_WIFI) to control the communication was then created. The programming computer was then configured by setting the PG/PC interface and transfer communication between the programing unit and the PLC from MPI to wireless.

The PLC was then allocated the IP address: 192.168.1.100. Another computer with wireless communication capability, laptop in this case, was then used as the programing unit.


test8.pcapng [Wireshark 1.12.7 (v1.12.7-0-g7fc8978 from master-1.12)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter:
Expression... Clear Apply Save

```
No.   Time        Source          Destination     Protocol  Length  Info
    1 0.00000000  192.168.1.100   192.168.1.10    COTP         111  DT TPDU (0) EOT
    2 0.00017200  192.168.1.10    192.168.1.100   COTP          61  DT TPDU (0) [COTP fragment, 0 bytes]
```

```
⊟ Frame 1: 111 bytes on wire (888 bits), 111 bytes captured (888 bits) on interface 0
    Interface id: 0 (\Device\NPF_{36CBD1E0-6330-47BB-8BDA-C204D687DE91})
    Encapsulation type: Ethernet (1)
    Arrival Time: Sep 15, 2015 12:28:26.795605000 South Africa Standard Time
    [Time shift for this packet: 0.000000000 seconds]
    Epoch Time: 1442312906.795605000 seconds
    [Time delta from previous captured frame: 0.000000000 seconds]
    [Time delta from previous displayed frame: 0.000000000 seconds]
    [Time since reference or first frame: 0.000000000 seconds]
    Frame Number: 1
    Frame Length: 111 bytes (888 bits)
    Capture Length: 111 bytes (888 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: eth:ethertype:ip:tcp:tpkt:cotp:data]
    [Number of per-protocol-data: 1]
    [ISO 8473/X.233 CLNP ConnectionLess Network Protocol, key 0]
⊞ Ethernet II, Src: Siemens_ae:af:02 (00:1b:1b:ae:af:02), Dst: LiteonTe_44:8f:9e (d0:df:9a:44:8f:9e)
⊞ Internet Protocol Version 4, Src: 192.168.1.100 (192.168.1.100), Dst: 192.168.1.10 (192.168.1.10)
⊞ Transmission Control Protocol, Src Port: 102 (102), Dst Port: 50502 (50502), Seq: 1, Ack: 1, Len: 57
⊞ TPKT, Version: 3, Length: 57
⊞ ISO 8073/X.224 COTP Connection-Oriented Transport Protocol
⊞ Data (50 bytes)
```

```
0000  d0 df 9a 44 8f 9e 00 1b  1b ae af 02 08 00 45 00   ...D.... ......E.
0010  00 61 f9 b4 00 00 3c 06  01 24 c0 a8 01 64 c0 a8   .a....<. .$...d..
0020  01 0a 00 66 c5 46 13 0b  77 b2 ae 36 be 51 50 18   ...f.F.. w..6.QP.
0030  08 00 59 7d 00 00 03 00  00 39 02 f0 80 32 07 00   ..Y}.... .9...2..
0040  00 00 00 00 0c 00 1c 00  01 12 08 12 01 02 04 00   ........ ........
0050  00 00 00 ff 00 00 18 00  04 00 10 01 00 00 02 00
```

Frame (frame), 111 bytes   Packets: 1238 · Displayed: 1238 (100.0%) · Load time: 0:00.030

Figure 6.6: Wireshark frame capture analysis

This computer established connection with the wilson_WIFI network and was allocated the IP address: 192.168.1.10 and the communication with the PLC was tested using the Ping utility illustrated in Figure 5.9. The Ping results illustrated in Figure 5.9 indicated that there was wireless communication between the programing unit and the PLC and that 32 bytes of data were sent from the programing unit to the PLC and the PLC responded by sending 32 bytes of data to the programing unit four times. The average transmission time between the two stations was 4 ms, no packets lost, and this indicated that the wireless communication channel was reliable to handle data. The same program that was used to test the PROFIBUS network was then transferred to the wireless programing unit, compiled and downloaded to the PLC via the wireless communication channel. The plant was then controlled and monitored from the programing unit over the wireless channel using the variable table illustrated in Figure 6.4. The Wireshark network analyser was then utilised to monitor the network and captured data for five test runs. For Test 1 VSP1 was running, for Test 2 VSP2 was running, for Test 3 the delivery pump was running, for Test 4 VSP1 and VSP2 were running while for Test 5 all pumps were running. The Wireshark network analyser was then used to analyse the captured data and the results indicated that:

(i)	Total packets captured over 5 tests	2030
(ii)	Total time elapsed between first and last packet	13.354 sec
(iii)	Average packets/sec	150.354
(iv)	Average packet size	77 bytes
(v)	Bytes	156014
(vi)	Average bytes/sec	11535.557

It can be deducted from the capture results that average of 2030 packets of data were transmitted over a period of 13.354 seconds. The average transmission was 150 packets were transmitted each packet carrying 77 bytes. This translates to an average 150x77x8 = 92400 bits transmitted per second with no error which is estimated to 0.092 Mbit/sec. This statistic is enough to deduce that the wireless network is a good and reliable network.

The comparison of Tables 3.2, 5.2 and 6.2 illustrate that the PROFIBUS network was transmitting an average of 0.186 Mbit/sec of user data without error while the wireless network was transmitting an average of 0.092 Mbit/sec without error. It was noted that the datagram structure for the PROFIBUS network is different from the packet structure of the wireless network but the processor of the LEAN card explained in Chapter 5 interfaced between the PROFIBUS network and the wireless network and matched the datagram structure to the packet structure and the average transmission rate. The wireless network therefore successfully replaced the PROFIBUS protocol between the programming computer and the PLC.

CHAPTER SEVEN: CONCLUSION

7.1 Introduction

Process automation measures and regulates a system by means of an appropriate control algorithm to maintain a desired process condition without human interference to meet plant specifications. An efficient and updated communication protocol in the automation field is PROFIBUS. The structure is configured by means of two conductors to transmit and receive data on a single pair up to 32 nodes in the network on multiplexing/de-multiplexing platform. Each device is allocated a unique address for identification. The data shared between the control point and the field devices is in the form of datagrams which is the lower level of data packet transmission. The program to control and monitor the process is written in the programing unit, compiled and downloaded to the controller, the PLC, but control and monitoring is done on the programing unit by means of variable tables.

The aim of this research was to model the wireless communication in industrial setting and implement it in real time. Modelling was done using Matlab and Simulink while real time implementation was carried out in the research plant at MUT by replacing the physical MPI cable between the programming computer and the PLC. The success of this implementation was the employment of wireless communication between the PLC and some field devices. Extensive research has been done in the field of wireless communication and these were reviewed to gather information to be used in this research and also the datasheets for various devices were consulted to gather sufficient information about the devices and to be able to choose the most suitable device to be used in order to achieve the desired results. The properties of these devices were mapped in the form of the table and analysed and compared to the IEEE 802.11 standard that governs wireless communication and the TL-MR3020 transceiver/router manufactured by TP-LINK was chosen. This wireless module employs the AR9331 chip which has the wireless front end integrated into it and is therefore a complete wireless network controller in one device.

The existing research plant was then configured for controlling the variable speed pumps, VSP1 and VSP2, delivery pump and to monitor the level in the blend chest. The application software program was written in statement list, Siemens Step-7, compiled and downloaded to the S7-313C-2DP PLC. The wireless hardware devices were integrated into the existing network, configured and tested for functionality.

7.2 Implications of the research

Since 2008 the technology in the research plant at MUT has been used for teaching, testing and demonstration of process automation and control using hard wired communication between the programing unit and the PLC and the work had to be loaded on the computer that is connected to the MPI cable before downloading. This created a bottleneck problem when many learners wanted to access the PLC. Another problem was experienced when the computer connected to MPI cable was faulty. The implementation of wireless communication between the programing unit and PLC in 2015 as the result of this research will allow other users and learners to access the PLC and control the network from various positions within the work floor if they have the computer with wireless facility and the Siemens Step-7 program installed. These findings can be used in industry in order to eliminate the problems encountered when there is a problem with the programing unit or the link between the computer and the PLC and to provide access to the PLC from various control points. The research plant is configured with sensors, final correcting devices and control software and hardware that currently exists in industry. The difference from an industrial application to the research facility is that all piping and tanks were brought down to scale physically. To implement the research into an industrial application it will entail reconfiguring the software and reprogramming all field devices. The control hardware and software remains the same.

7.3 Application of the research

The learners and other users of the research lab will be able to access the research plant from any position in the laboratory building, especially for technical support staff members who can remotely prepare student work in real-time, from their offices. The monitoring and control of various processes in the research plant can be carried out from various points in the plant. This facility will enable the supervisor to monitor the activities of the plant without interacting directly with the plant. The research project can be used as a test facility where industry considers implementing wireless communication. Since the PROFIBUS communication averaged a 0.186 Mbit/sec rate using ProfiTrace and the wireless network averaged a 0.092 Mbit/sec rate using Wireshark, the wireless network can be assumed as the preferred protocol to use. However, the following considerations must be taken into account:

(i) ProfiTrace analyses the entire communication data signal whilst Wireshark analyses using a filtering system as discussed in page 56

(ii) The research was not conducted in an ideal industrial environment

7.4 Limitations of the research

The research outcomes were not tested for long range wireless communication and in a typical industrial environment where electromagnetic induction (EMI) is common. The strength of the radio signal was not tested to establish specifications on bandwidth.

7.5 Suggestions for future research

The work done on this research was limited to controlling the pumps from the remote programing unit and monitoring the blend chest level over the wireless communication channel. There is a need for future research to implement wireless communication between the PLC and other field devices. This facility will test the effect of EMI on the communication protocol especially in an industrial environment where heavy electrical machinery exist. Wireless communication in the process and automation industry is currently used for signal transmission only, for process measurement, due to interferences like EMI that can be measured resulting in an incorrect process variable [35]. This could impact negatively on the control algorithm as it affects the stability of the control system which can be tested further.

REFERENCES

[1] http://www.statssa.gov.za/additional_services/siccoder/default.aspx.

[2] Mário Alves, Eduado Tovar, “Real-time communications over wired/wireless PROFIBUS networks supporting inter-cell mobility”. 2007.

[3] J. Higino Correia, E. Cretu, M. Bartek and R.F Wolffenbuttel, ”A Microinstrumentation System for Industrial Applications”. 1997.

[4] http://www.smar.com/PDFs/Catalogues/Introduction_PROFIBUS.PDF.

[5] Andy Vewer, “Introduction to PROFIBUS and PROFINET”. 2012.

[6] Jianping Song, Aloysius K. Mok, Deji Chen, Mike Lucas, Mark Nixon, “WirelessHART: Applying Wireless Technology in Real- Time Industrial Process Control”. 2008.

[7] Khalil I. Arshak, Deidre Morris, Arousian Arshak, Olga Koronstyska, and Essa Jafer, “Development of a Wireless Pressure Measurement System Using Interdigitated Capacitors”. 2006.

[8] Andrew Mason, Navid Yazdi, Khalil Najafi, and D. Wise, “A Low-power Wireless Micro-Instrumentation System for Environmental Monitoring”. 1995.

[9] Hong Zhang, Yong Zheng, and Xing-xin Sun, “Design of a Wireless Data Acquisition System Based on nRF24E1”. 2009.

[10] Xu Bai, and Yongjie Fu, “Design of Pressure Test System Based on Wireless Communication Technology”. 2009.

[11] Zhan Yanbing, “Design of Low-Power Wireless Communication System Based on MSP430 and nRF2401”. 2010.

[12] Hui Guo, Cheng-hua Fu, Hao Wu, Shu-chuan Gun, and Chang-zhong Chen, “Research of Wireless Communication between PLC and Computer Based on nRF2401”. 2010.

[13] PI PROFIBUS.PROFINET, PROFIBUS System Description, Technology and Application. PROFIBUS Nutzerorganisation. 2010.

[14] Grant Weyman, “PROFIBUS PA”, PROFIBUS Competence Centre, Australia. 2009.

[15] SIEMENS SIMATIC, Bus links, DP/PA Coupler, 10/2006, A5E00193841-16.

[16] SAMSON Technical Information, PROFIBUS PA, SAMSON AG. 2012.

[17] SIEMENS SITRANS, Pressure transmitter, SITRANS P, Series DS III with PROFIBUS PA communication, 09/2008, A5E00053276-05

[18] R. Bañares-Alcantara and H.M.S. Lababidi, “Design Support Systems for Process Engineering-II. KBDS: An Experimental Prototype”. 1994.

[19] Marcus Fontoura, “A Framework Design and Instantiation Method”. 1998.

[20] S. Perrot, C. Person, L. Corré, H. Lattard, and M. Ney, “Proposed design Solutions for Low-Cost Flip-Chip Membrane Devices for Millimeterwave Applications”. IEEE MTT-S Digest. 1999.

[21] RF Monolithics, Inc, “802.11g Wireless Sensor Network Module”. 2009-2010.

[22] Nordic Semiconductor, “nRF24L01 Single Chip 2.4 GHz Transceiver Product Specification”. 2007.

[23] Hope RF, HM-TR Transparent Wireless Data Link Module”. 2008.

[24] Laird Technologies, “2.4 GHz Wireless Module”. 2011.

[25] Linx Technologies, “LT Series Transceiver Module”. 2012.

[26] Microchip, “MRF89XAMxA PICTail Daughter Board”. 2011.

[27] Shen Zhen Yishi Electronic Technology Development Co. “YS Ultra low power wireless module”. 2008.

[28] http://www.rpi.edu/dept/ecse/mps/SPI.pdf

[29] SIEMENS MICROMASTER 440, Parameter List User Documentation, 6SE6400-5BB00-0BP0. 2002.

[30] http://www.kresttechnology.com. “Wireless Energy Meter using RF Communication”.

[31] Z. Nemec, R. Dolesec and Z. Silar. “The Model of Communication Channel in the 802.11b Standard wireless Network”. 2008.

[32] TL-MR3020 User Guide. 2011.

[33] AR9331 Data Sheet. 2010.

[34] Md. Golam Sadeque, Bit Error Rate (BER) Comparison of AWGN Channels for Different Types of Digital Modulation using Matlab Simulink, 2015.

[35] Honeywell, XYR 6000 Wireless Universal I/O Transmitter, 34-XY-16U-58. 2016.

[36] https://www.google.co.za/search?q=Profibus+cables

[37] Nuray AT and Daraghma SM. “A New Energy Efficient Clustering-based Protocol for Heterogeneous Wireless Sensor networks”. 2015.

[38] Sabzpoushan SH, Maleki A and Miri F. “A Novel Application of Sensor Networks in Biomedical Engineering”. 2015.

[39] Farrukh S. “Extending the functionality of Paymote: Low Level Protocols and Simulation Results Analysis”. 2015.

[40] Rajagopal D and Thilakavalli K. “Monitoring Internet Access along with Usage of Bandwidth using Intrusion Detection System”. 2015.

APPENDIX A

1. PROGRAM FOR SKETCHING FIGURE 4.1

```
% Used in one of Signal Processing Labs at MUT
d=[1 0 1 1 0 ];         % Data sequence
b=2*d-1;                % Convert unipolar to bipolar
T=1;                    % Bit duration
Eb=T/2;                 % This will result in unit amplitude waveforms
fc=3/T;                 % Carrier frequency
t=linspace(0,5,1000); % Discrete time sequence between 0 and 5*T (1000 samples)
N=length(t);            % Number of samples
Nsb=N/length(d); % Number of samples per bit
dd=repmat(d',1,Nsb); % replicate each bit Nsb times
bb=repmat(b',1,Nsb); dw=dd'; % Transpose the rows and columns
dw=dw(:)';
% Convert dw to a column vector (colum by column) and convert to a row vector
bw=bb';
bw=bw(:)'; % Data sequence samples
w=sqrt(2*Eb/T)*cos(2*pi*fc*t); % carrier waveform
bpsk_w=bw.*w; % modulated waveform

% Plotting commands

subplot(2,1,1);
plot(t,dw); axis([0 5 -1.5 1.5])
xlabel('Time'); ylabel('Amplitude');
title('Binary Data');

subplot(2,1,2);
plot(t,bpsk_w); axis([0 5 -1.5 1.5])
xlabel('Time'); ylabel('Amplitude');
title('BPSK Signal')
```

2. PROGRAM FOR SKETCHING FIGURE 4.2

```
% QPSK Modulation
% Used in one of Signal Processing Labs at MUT

clc;
clear all;
close all;
Tb=1;t=0:(Tb/100):Tb;fc=1;     % GENERATE QUADRATURE CARRIER SIGNAL
c1=sqrt(2/Tb)*cos(2*pi*fc*t);
c2=sqrt(2/Tb)*sin(2*pi*fc*t);
N=8;m=rand(1,N);                  % Generate message signal
t1=0;t2=Tb
for i=1:2:(N-1)
     t=[t1:(Tb/100):t2]
   if m(i)>0.5
      m(i)=1;
      m_s=ones(1,length(t));
   else
      m(i)=0;
      m_s=-1*ones(1,length(t));
   end
   odd_sig(i,:)=c1.*m_s;      % Odd bits modulated signal

   if m(i+1)>0.5
       m(i+1)=1;
      m_s=ones(1,length(t));
   else
      m(i+1)=0;
      m_s=-1*ones(1,length(t));
   end

   even_sig(i,:)=c2.*m_s;    % Even bits modulated signal
   %qpsk signal
   qpsk=odd_sig+even_sig;
   subplot(2,2,3);plot(t,qpsk(i,:));        % Plot the QPSK modulated signal
   title('QPSK Signal');xlabel('Time');ylabel('Amplitude');grid on; hold on;
   t1=t1+(Tb+.01); t2=t2+(Tb+.01);
 end
hold off
subplot(2,2,1);stem(m);      %Plot the binary data bits and carrier signal
title('binary data bits');xlabel('n');ylabel('Amplitude');grid on;
subplot(2,2,2);plot(t,c1);
title('Carrier-I');xlabel('Time');ylabel('Carrier 1');grid on;
subplot(2,2,4);plot(t,c2);
title('Carrier-Q');xlabel('Time');ylabel('Carrier 2');grid on;
```

3. PROGRAM FOR PLOTTING FIGURE 4.4

```
x = -10:1:3; % Range on x-axis
y = [0.1876 0.165 0.1164 0.09001 0.07704 0.05223 0.03743 0.02458
   --> 0.01334 0.006779 0.002 0.000799 0.001 0]; %BER values
semilogy(x,y,'d')
xlabel('Eb/No (dB)');
ylabel('BER');
title('Simulated BER');
grid on;
```

4. PROGRAM FOR PLOTTING FIGURE 4.6

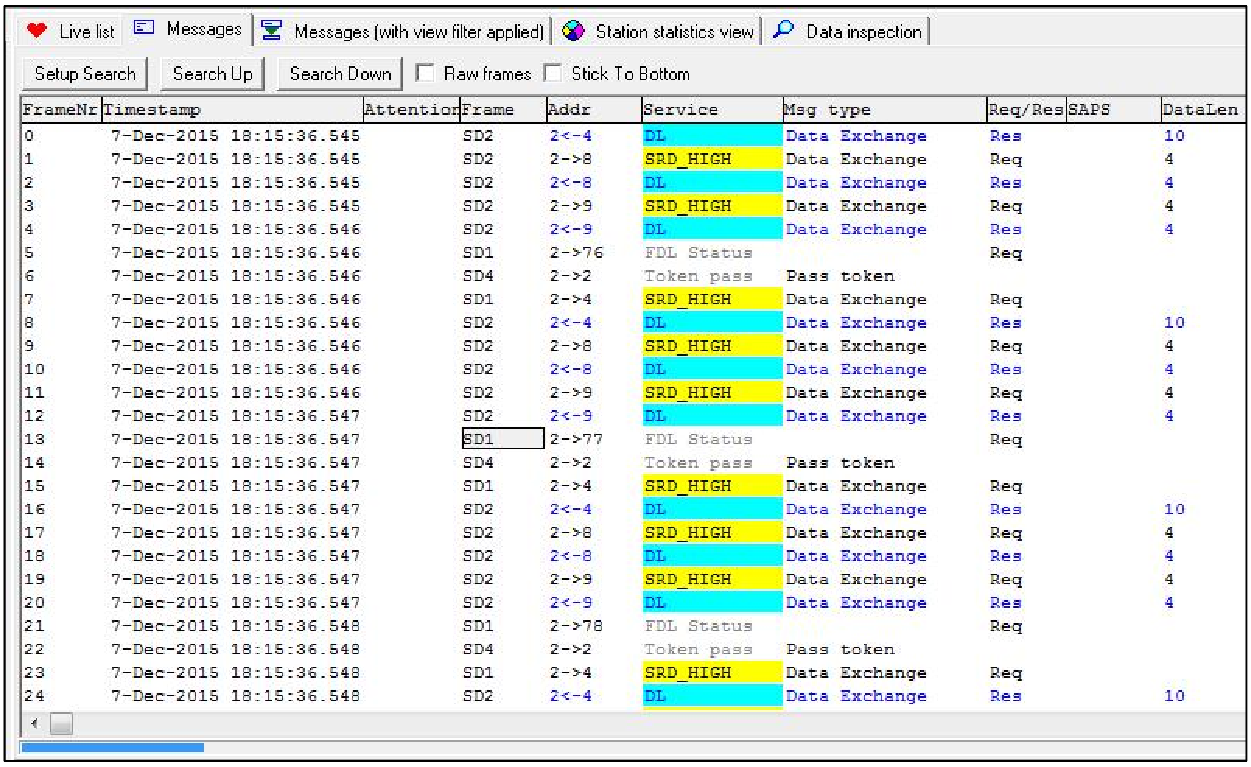
```
x = -2:1:11;  % Range on x-axis
y = [0.3778 0.3437 0.307 0.2498 0.1988 0.1548 0.1189 0.08192
   --> 0.0499 0.02797 0.01499 0.003996 0.002997 0]; % BER values
semilogy(x,y)
xlabel('Eb/No (dB)');
ylabel('BER');
title('Simulated BER for QPSK');
grid on;
```

5. PROGRAM FOR PLOTTING FIGURE 4.8

```
x = -10:1:3; % Range on x-axis
y = [0.429 0.4176 0.3447 0.2737 0.2238 0.1688 0.1209 0.07792
   --> 0.04196 0.02298 0.008991 0.003996 0.000999 0]; % BER values
semilogy(x,y)
xlabel('Eb/No (dB)');
ylabel('BER');
title('Simulated BER for 16-QAM');
grid on
```

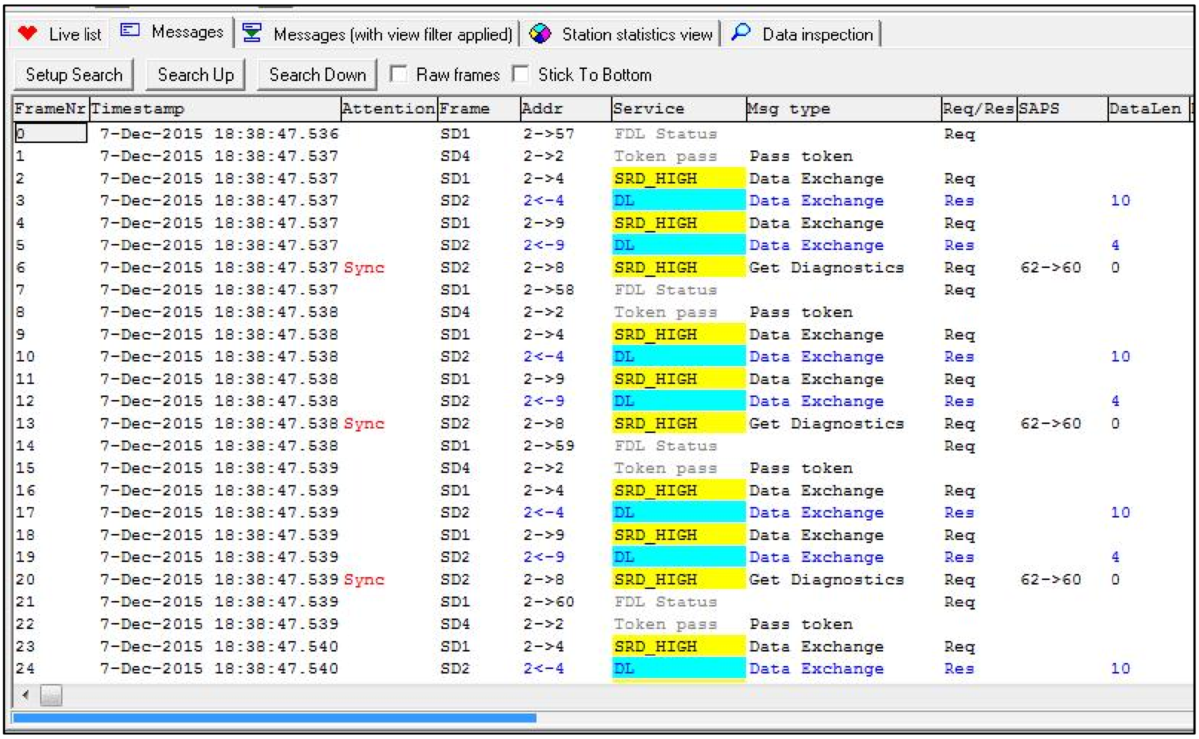
APPENDIX B

1. PROFITRACE CAPTURE FOR VSP1 IN TABLE 3.2



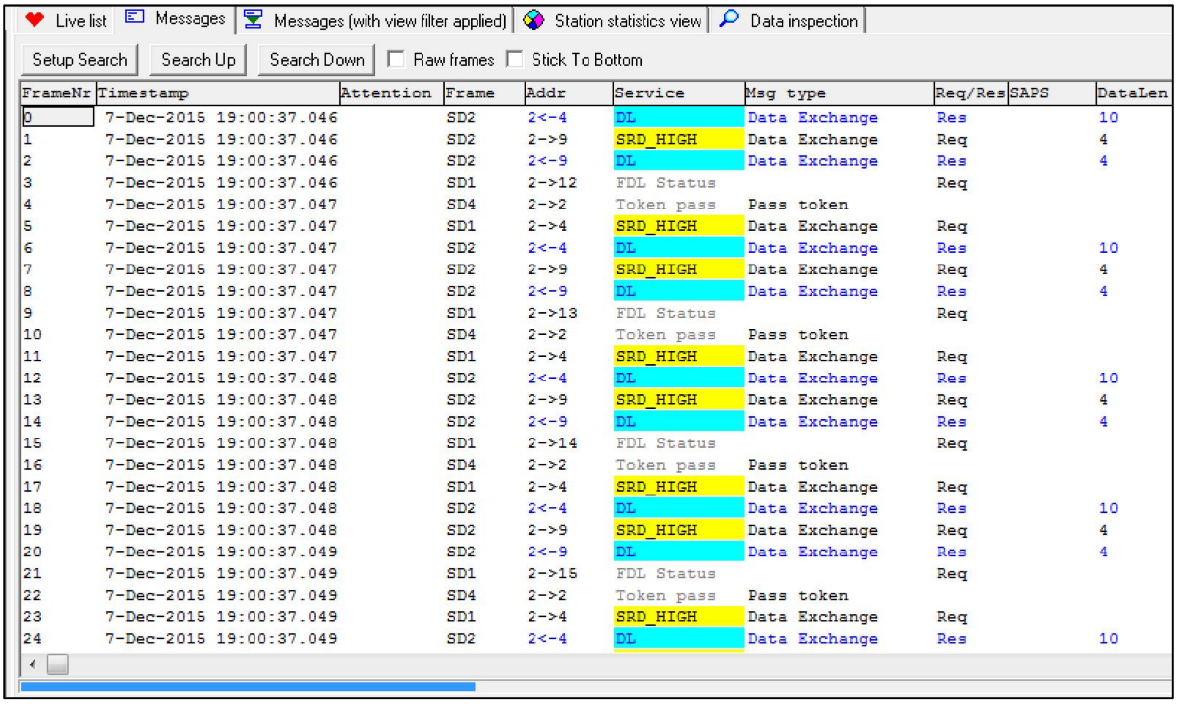
FrameNr	Timestamp	Attention	Frame	Addr	Service	Msg type	Req/Res	SAPS	DataLen
0	7-Dec-2015 18:15:36.545		SD2	2<-4	DL	Data Exchange	Res		10
1	7-Dec-2015 18:15:36.545		SD2	2->8	SRD_HIGH	Data Exchange	Req		4
2	7-Dec-2015 18:15:36.545		SD2	2<-8	DL	Data Exchange	Res		4
3	7-Dec-2015 18:15:36.545		SD2	2->9	SRD_HIGH	Data Exchange	Req		4
4	7-Dec-2015 18:15:36.546		SD2	2<-9	DL	Data Exchange	Res		4
5	7-Dec-2015 18:15:36.546		SD1	2->76	FDL Status		Req		
6	7-Dec-2015 18:15:36.546		SD4	2->2	Token pass	Pass token			
7	7-Dec-2015 18:15:36.546		SD1	2->4	SRD_HIGH	Data Exchange	Req		
8	7-Dec-2015 18:15:36.546		SD2	2<-4	DL	Data Exchange	Res		10
9	7-Dec-2015 18:15:36.546		SD2	2->8	SRD_HIGH	Data Exchange	Req		4
10	7-Dec-2015 18:15:36.546		SD2	2<-8	DL	Data Exchange	Res		4
11	7-Dec-2015 18:15:36.546		SD2	2->9	SRD_HIGH	Data Exchange	Req		4
12	7-Dec-2015 18:15:36.547		SD2	2<-9	DL	Data Exchange	Res		4
13	7-Dec-2015 18:15:36.547		SD1	2->77	FDL Status		Req		
14	7-Dec-2015 18:15:36.547		SD4	2->2	Token pass	Pass token			
15	7-Dec-2015 18:15:36.547		SD1	2->4	SRD_HIGH	Data Exchange	Req		
16	7-Dec-2015 18:15:36.547		SD2	2<-4	DL	Data Exchange	Res		10
17	7-Dec-2015 18:15:36.547		SD2	2->8	SRD_HIGH	Data Exchange	Req		4
18	7-Dec-2015 18:15:36.547		SD2	2<-8	DL	Data Exchange	Res		4
19	7-Dec-2015 18:15:36.547		SD2	2->9	SRD_HIGH	Data Exchange	Req		4
20	7-Dec-2015 18:15:36.547		SD2	2<-9	DL	Data Exchange	Res		4
21	7-Dec-2015 18:15:36.548		SD1	2->78	FDL Status		Req		
22	7-Dec-2015 18:15:36.548		SD4	2->2	Token pass	Pass token			
23	7-Dec-2015 18:15:36.548		SD1	2->4	SRD_HIGH	Data Exchange	Req		
24	7-Dec-2015 18:15:36.548		SD2	2<-4	DL	Data Exchange	Res		10

2. PROFITRACE CAPTURE FOR VSP2 IN TABLE 3.2



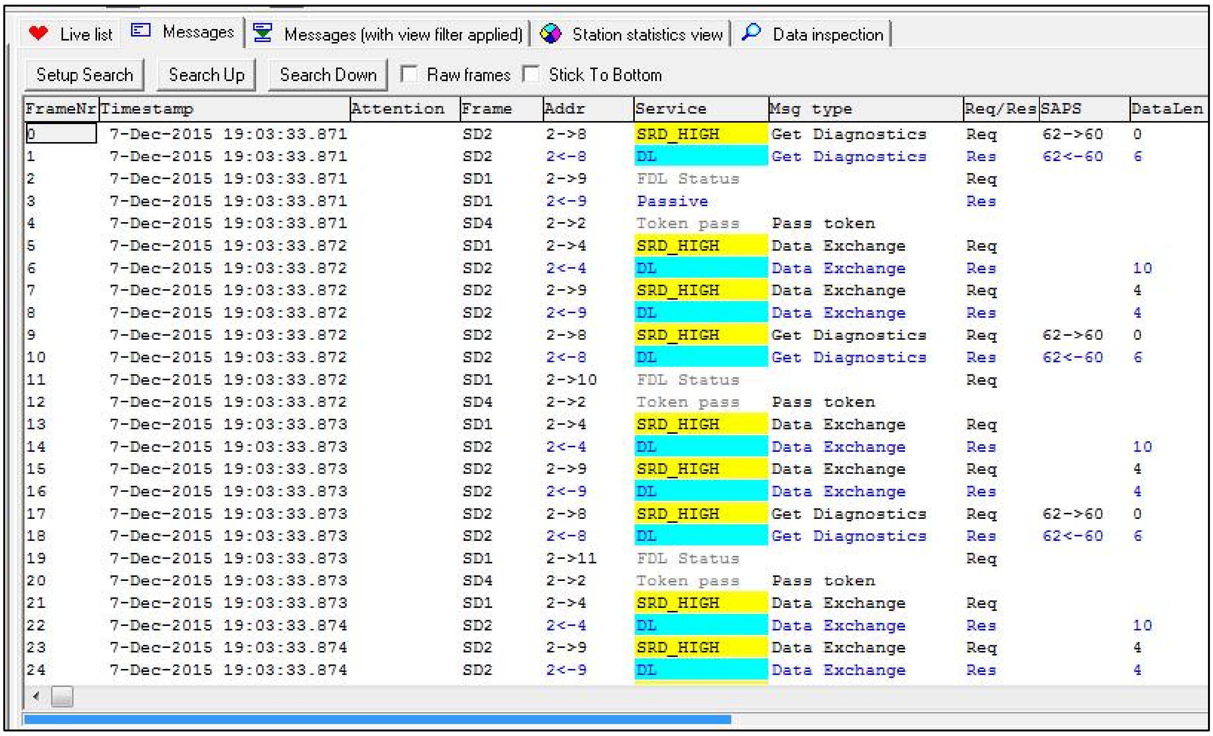
FrameNr	Timestamp	Attention	Frame	Addr	Service	Msg type	Req/Res	SAPS	DataLen
0	7-Dec-2015 18:38:47.536		SD1	2->57	FDL Status		Req		
1	7-Dec-2015 18:38:47.537		SD4	2->2	Token pass	Pass token			
2	7-Dec-2015 18:38:47.537		SD1	2->4	SRD_HIGH	Data Exchange	Req		
3	7-Dec-2015 18:38:47.537		SD2	2<-4	DL	Data Exchange	Res		10
4	7-Dec-2015 18:38:47.537		SD1	2->9	SRD_HIGH	Data Exchange	Req		
5	7-Dec-2015 18:38:47.537		SD2	2<-9	DL	Data Exchange	Res		4
6	7-Dec-2015 18:38:47.537	Sync	SD2	2->8	SRD_HIGH	Get Diagnostics	Req	62->60	0
7	7-Dec-2015 18:38:47.537		SD1	2->58	FDL Status		Req		
8	7-Dec-2015 18:38:47.538		SD4	2->2	Token pass	Pass token			
9	7-Dec-2015 18:38:47.538		SD1	2->4	SRD_HIGH	Data Exchange	Req		
10	7-Dec-2015 18:38:47.538		SD2	2<-4	DL	Data Exchange	Res		10
11	7-Dec-2015 18:38:47.538		SD1	2->9	SRD_HIGH	Data Exchange	Req		
12	7-Dec-2015 18:38:47.538		SD2	2<-9	DL	Data Exchange	Res		4
13	7-Dec-2015 18:38:47.538	Sync	SD2	2->8	SRD_HIGH	Get Diagnostics	Req	62->60	0
14	7-Dec-2015 18:38:47.538		SD1	2->59	FDL Status		Req		
15	7-Dec-2015 18:38:47.539		SD4	2->2	Token pass	Pass token			
16	7-Dec-2015 18:38:47.539		SD1	2->4	SRD_HIGH	Data Exchange	Req		
17	7-Dec-2015 18:38:47.539		SD2	2<-4	DL	Data Exchange	Res		10
18	7-Dec-2015 18:38:47.539		SD1	2->9	SRD_HIGH	Data Exchange	Req		
19	7-Dec-2015 18:38:47.539		SD2	2<-9	DL	Data Exchange	Res		4
20	7-Dec-2015 18:38:47.539	Sync	SD2	2->8	SRD_HIGH	Get Diagnostics	Req	62->60	0
21	7-Dec-2015 18:38:47.539		SD1	2->60	FDL Status		Req		
22	7-Dec-2015 18:38:47.539		SD4	2->2	Token pass	Pass token			
23	7-Dec-2015 18:38:47.540		SD1	2->4	SRD_HIGH	Data Exchange	Req		
24	7-Dec-2015 18:38:47.540		SD2	2<-4	DL	Data Exchange	Res		10

3. PROFITRACE CAPTURE FOR VSP1+VSP2 IN TABLE 3.2



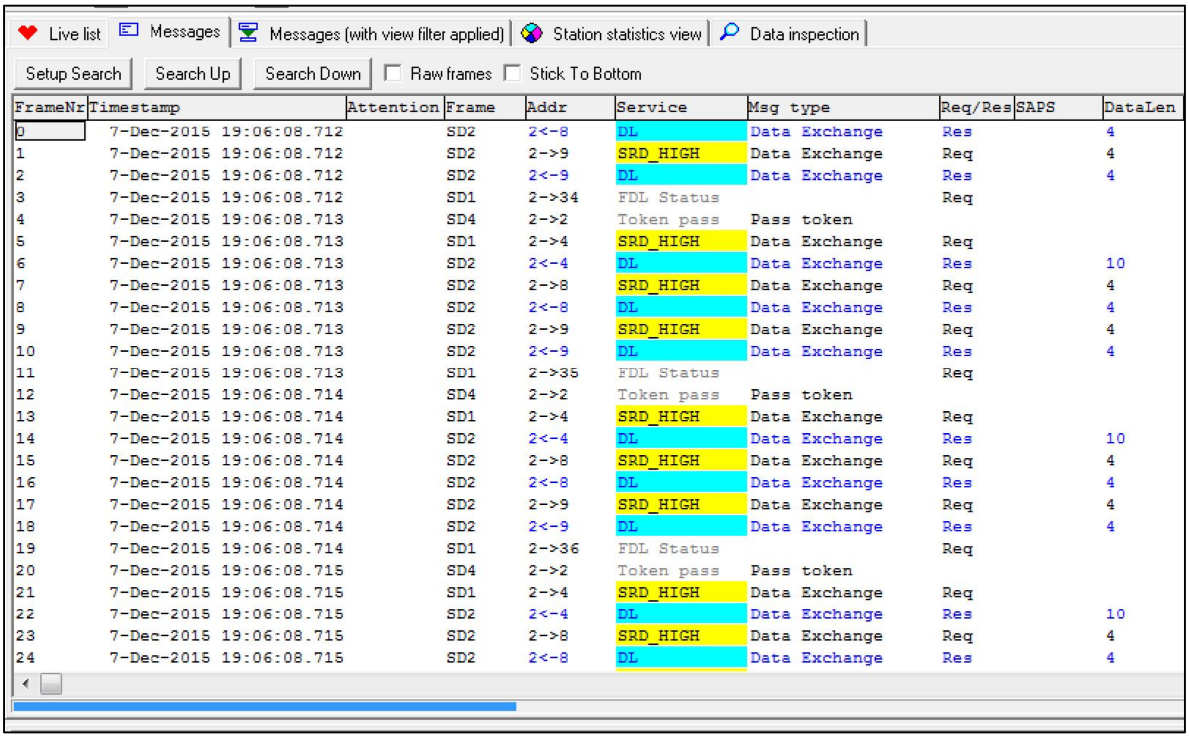
FrameNr	Timestamp	Attention	Frame	Addr	Service	Msg type	Req/Res	SAPS	DataLen
0	7-Dec-2015 19:00:37.046		SD2	2<-4	DL	Data Exchange	Res		10
1	7-Dec-2015 19:00:37.046		SD2	2->9	SRD_HIGH	Data Exchange	Req		4
2	7-Dec-2015 19:00:37.046		SD2	2<-9	DL	Data Exchange	Res		4
3	7-Dec-2015 19:00:37.046		SD1	2->12	FDL Status		Req		
4	7-Dec-2015 19:00:37.047		SD4	2->2	Token pass	Pass token			
5	7-Dec-2015 19:00:37.047		SD1	2->4	SRD_HIGH	Data Exchange	Req		
6	7-Dec-2015 19:00:37.047		SD2	2<-4	DL	Data Exchange	Res		10
7	7-Dec-2015 19:00:37.047		SD2	2->9	SRD_HIGH	Data Exchange	Req		4
8	7-Dec-2015 19:00:37.047		SD2	2<-9	DL	Data Exchange	Res		4
9	7-Dec-2015 19:00:37.047		SD1	2->13	FDL Status		Req		
10	7-Dec-2015 19:00:37.047		SD4	2->2	Token pass	Pass token			
11	7-Dec-2015 19:00:37.047		SD1	2->4	SRD_HIGH	Data Exchange	Req		
12	7-Dec-2015 19:00:37.048		SD2	2<-4	DL	Data Exchange	Res		10
13	7-Dec-2015 19:00:37.048		SD2	2->9	SRD_HIGH	Data Exchange	Req		4
14	7-Dec-2015 19:00:37.048		SD2	2<-9	DL	Data Exchange	Res		4
15	7-Dec-2015 19:00:37.048		SD1	2->14	FDL Status		Req		
16	7-Dec-2015 19:00:37.048		SD4	2->2	Token pass	Pass token			
17	7-Dec-2015 19:00:37.048		SD1	2->4	SRD_HIGH	Data Exchange	Req		
18	7-Dec-2015 19:00:37.048		SD2	2<-4	DL	Data Exchange	Res		10
19	7-Dec-2015 19:00:37.048		SD2	2->9	SRD_HIGH	Data Exchange	Req		4
20	7-Dec-2015 19:00:37.049		SD2	2<-9	DL	Data Exchange	Res		4
21	7-Dec-2015 19:00:37.049		SD1	2->15	FDL Status		Req		
22	7-Dec-2015 19:00:37.049		SD4	2->2	Token pass	Pass token			
23	7-Dec-2015 19:00:37.049		SD1	2->4	SRD_HIGH	Data Exchange	Req		
24	7-Dec-2015 19:00:37.049		SD2	2<-4	DL	Data Exchange	Res		10

4. PROFITRACE CAPTURE FOR DELIVERY PUMP IN TABLE 3.2



FrameNr	Timestamp	Attention	Frame	Addr	Service	Msg type	Req/Res	SAPS	DataLen
0	7-Dec-2015 19:03:33.871		SD2	2->8	SRD_HIGH	Get Diagnostics	Req	62->60	0
1	7-Dec-2015 19:03:33.871		SD2	2<-8	DL	Get Diagnostics	Res	62<-60	6
2	7-Dec-2015 19:03:33.871		SD1	2->9	FDL Status		Req		
3	7-Dec-2015 19:03:33.871		SD1	2<-9	Passive		Res		
4	7-Dec-2015 19:03:33.871		SD4	2->2	Token pass	Pass token			
5	7-Dec-2015 19:03:33.872		SD1	2->4	SRD_HIGH	Data Exchange	Req		
6	7-Dec-2015 19:03:33.872		SD2	2<-4	DL	Data Exchange	Res		10
7	7-Dec-2015 19:03:33.872		SD2	2->9	SRD_HIGH	Data Exchange	Req		4
8	7-Dec-2015 19:03:33.872		SD2	2<-9	DL	Data Exchange	Res		4
9	7-Dec-2015 19:03:33.872		SD2	2->8	SRD_HIGH	Get Diagnostics	Req	62->60	0
10	7-Dec-2015 19:03:33.872		SD2	2<-8	DL	Get Diagnostics	Res	62<-60	6
11	7-Dec-2015 19:03:33.872		SD1	2->10	FDL Status		Req		
12	7-Dec-2015 19:03:33.872		SD4	2->2	Token pass	Pass token			
13	7-Dec-2015 19:03:33.873		SD1	2->4	SRD_HIGH	Data Exchange	Req		
14	7-Dec-2015 19:03:33.873		SD2	2<-4	DL	Data Exchange	Res		10
15	7-Dec-2015 19:03:33.873		SD2	2->9	SRD_HIGH	Data Exchange	Req		4
16	7-Dec-2015 19:03:33.873		SD2	2<-9	DL	Data Exchange	Res		4
17	7-Dec-2015 19:03:33.873		SD2	2->8	SRD_HIGH	Get Diagnostics	Req	62->60	0
18	7-Dec-2015 19:03:33.873		SD2	2<-8	DL	Get Diagnostics	Res	62<-60	6
19	7-Dec-2015 19:03:33.873		SD1	2->11	FDL Status		Req		
20	7-Dec-2015 19:03:33.873		SD4	2->2	Token pass	Pass token			
21	7-Dec-2015 19:03:33.873		SD1	2->4	SRD_HIGH	Data Exchange	Req		
22	7-Dec-2015 19:03:33.874		SD2	2<-4	DL	Data Exchange	Res		10
23	7-Dec-2015 19:03:33.874		SD2	2->9	SRD_HIGH	Data Exchange	Req		4
24	7-Dec-2015 19:03:33.874		SD2	2<-9	DL	Data Exchange	Res		4

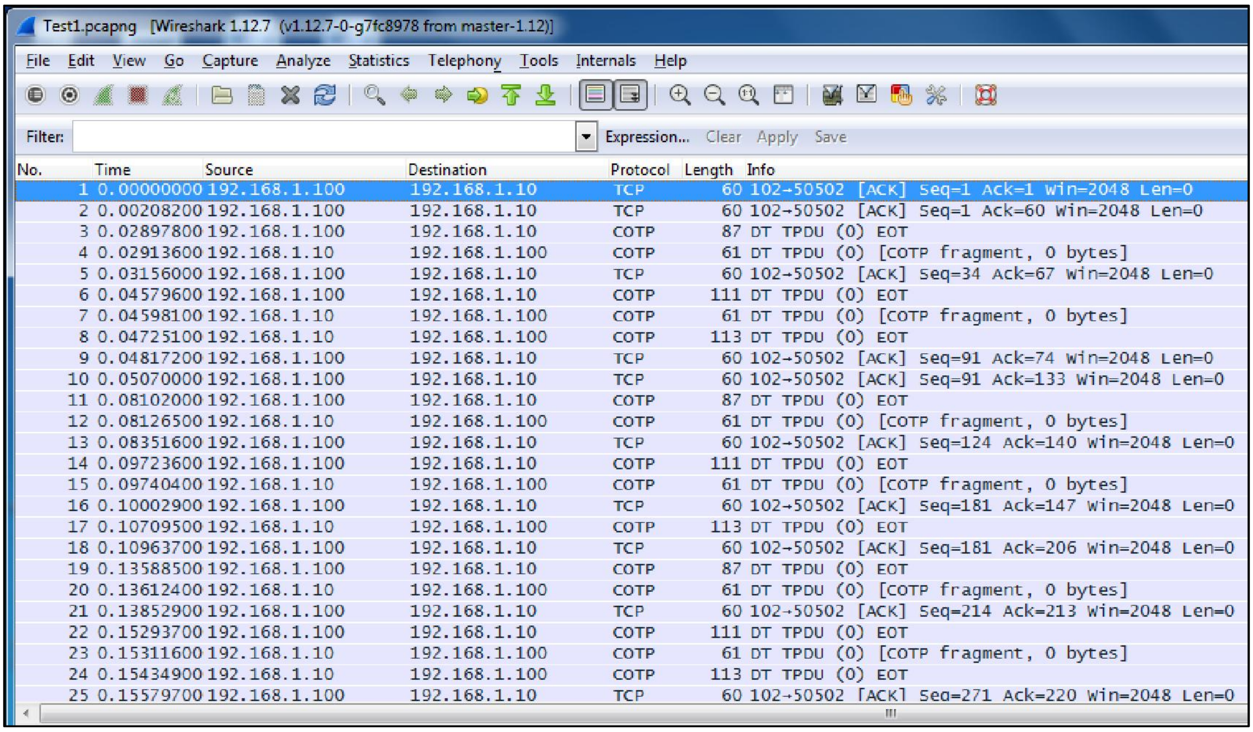
5. PROFITRACE CAPTURE FOR ALL PUMPS IN TABLE 3.2



FrameNr	Timestamp	Attention	Frame	Addr	Service	Msg type	Req/Res	SAPS	DataLen
0	7-Dec-2015 19:06:08.712		SD2	2<-8	DL	Data Exchange	Res		4
1	7-Dec-2015 19:06:08.712		SD2	2->9	SRD_HIGH	Data Exchange	Req		4
2	7-Dec-2015 19:06:08.712		SD2	2<-9	DL	Data Exchange	Res		4
3	7-Dec-2015 19:06:08.712		SD1	2->34	FDL Status		Req		
4	7-Dec-2015 19:06:08.713		SD4	2->2	Token pass	Pass token			
5	7-Dec-2015 19:06:08.713		SD1	2->4	SRD_HIGH	Data Exchange	Req		
6	7-Dec-2015 19:06:08.713		SD2	2<-4	DL	Data Exchange	Res		10
7	7-Dec-2015 19:06:08.713		SD2	2->8	SRD_HIGH	Data Exchange	Req		4
8	7-Dec-2015 19:06:08.713		SD2	2<-8	DL	Data Exchange	Res		4
9	7-Dec-2015 19:06:08.713		SD2	2->9	SRD_HIGH	Data Exchange	Req		4
10	7-Dec-2015 19:06:08.713		SD2	2<-9	DL	Data Exchange	Res		4
11	7-Dec-2015 19:06:08.713		SD1	2->35	FDL Status		Req		
12	7-Dec-2015 19:06:08.714		SD4	2->2	Token pass	Pass token			
13	7-Dec-2015 19:06:08.714		SD1	2->4	SRD_HIGH	Data Exchange	Req		
14	7-Dec-2015 19:06:08.714		SD2	2<-4	DL	Data Exchange	Res		10
15	7-Dec-2015 19:06:08.714		SD2	2->8	SRD_HIGH	Data Exchange	Req		4
16	7-Dec-2015 19:06:08.714		SD2	2<-8	DL	Data Exchange	Res		4
17	7-Dec-2015 19:06:08.714		SD2	2->9	SRD_HIGH	Data Exchange	Req		4
18	7-Dec-2015 19:06:08.714		SD2	2<-9	DL	Data Exchange	Res		4
19	7-Dec-2015 19:06:08.714		SD1	2->36	FDL Status		Req		
20	7-Dec-2015 19:06:08.715		SD4	2->2	Token pass	Pass token			
21	7-Dec-2015 19:06:08.715		SD1	2->4	SRD_HIGH	Data Exchange	Req		
22	7-Dec-2015 19:06:08.715		SD2	2<-4	DL	Data Exchange	Res		10
23	7-Dec-2015 19:06:08.715		SD2	2->8	SRD_HIGH	Data Exchange	Req		4
24	7-Dec-2015 19:06:08.715		SD2	2<-8	DL	Data Exchange	Res		4

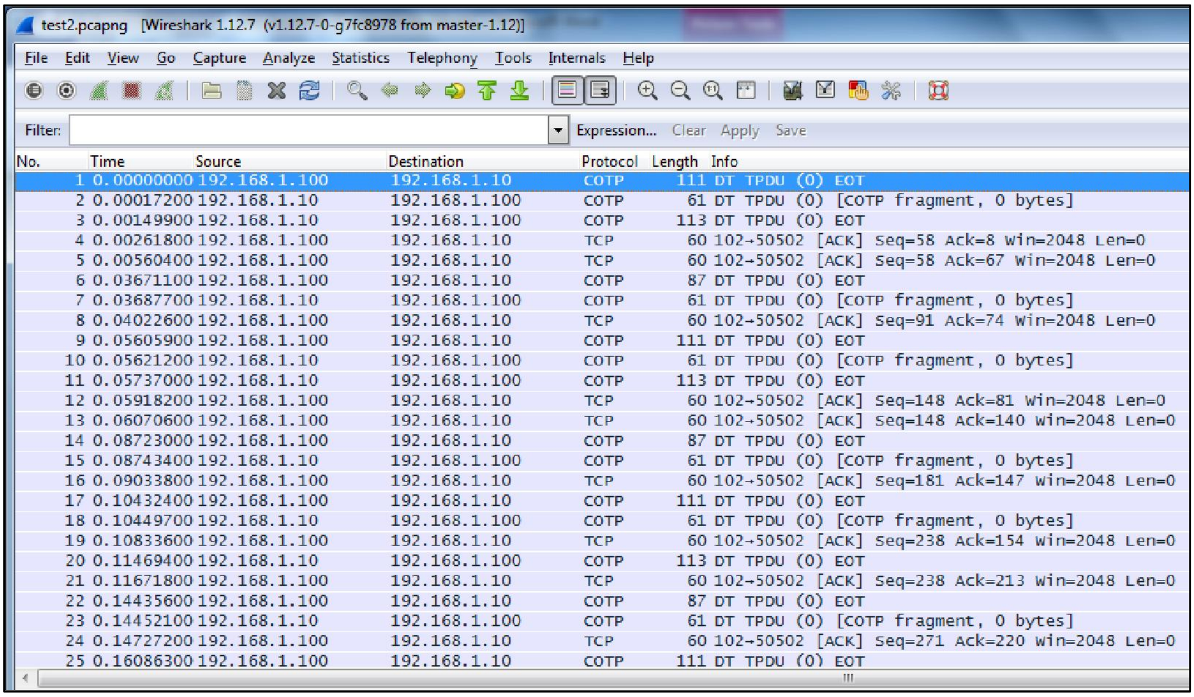
APPENDIX C

1. WIRESHARK CAPTURE FOR TEST 1


Test1.pcapng [Wireshark 1.12.7 (v1.12.7-0-g7fc8978 from master-1.12)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter:
Expression... Clear Apply Save

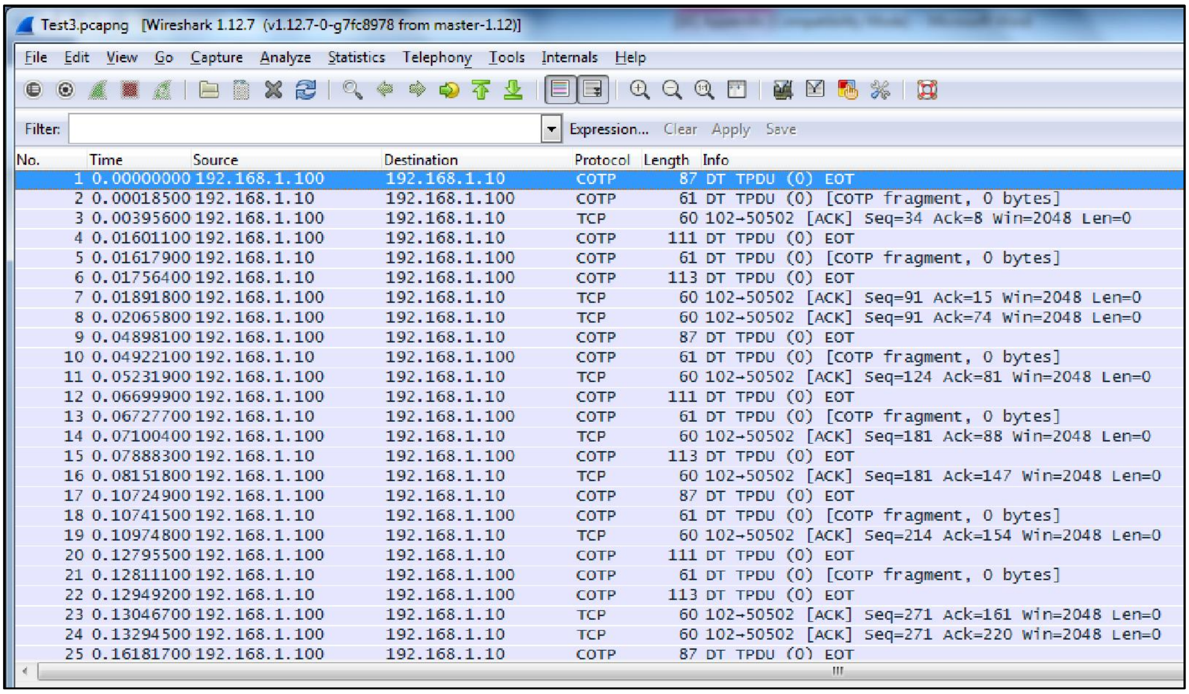
No.	Time	Source	Destination	Protocol	Length	Info
1	0.00000000	192.168.1.100	192.168.1.10	TCP	60	102→50502 [ACK] Seq=1 Ack=1 Win=2048 Len=0
2	0.00208200	192.168.1.100	192.168.1.10	TCP	60	102→50502 [ACK] Seq=1 Ack=60 Win=2048 Len=0
3	0.02897800	192.168.1.100	192.168.1.10	COTP	87	DT TPDU (0) EOT
4	0.02913600	192.168.1.10	192.168.1.100	COTP	61	DT TPDU (0) [COTP fragment, 0 bytes]
5	0.03156000	192.168.1.100	192.168.1.10	TCP	60	102→50502 [ACK] Seq=34 Ack=67 Win=2048 Len=0
6	0.04579600	192.168.1.100	192.168.1.10	COTP	111	DT TPDU (0) EOT
7	0.04598100	192.168.1.10	192.168.1.100	COTP	61	DT TPDU (0) [COTP fragment, 0 bytes]
8	0.04725100	192.168.1.10	192.168.1.100	COTP	113	DT TPDU (0) EOT
9	0.04817200	192.168.1.100	192.168.1.10	TCP	60	102→50502 [ACK] Seq=91 Ack=74 Win=2048 Len=0
10	0.05070000	192.168.1.100	192.168.1.10	TCP	60	102→50502 [ACK] Seq=91 Ack=133 Win=2048 Len=0
11	0.08102000	192.168.1.100	192.168.1.10	COTP	87	DT TPDU (0) EOT
12	0.08126500	192.168.1.10	192.168.1.100	COTP	61	DT TPDU (0) [COTP fragment, 0 bytes]
13	0.08351600	192.168.1.100	192.168.1.10	TCP	60	102→50502 [ACK] Seq=124 Ack=140 Win=2048 Len=0
14	0.09723600	192.168.1.100	192.168.1.10	COTP	111	DT TPDU (0) EOT
15	0.09740400	192.168.1.10	192.168.1.100	COTP	61	DT TPDU (0) [COTP fragment, 0 bytes]
16	0.10002900	192.168.1.100	192.168.1.10	TCP	60	102→50502 [ACK] Seq=181 Ack=147 Win=2048 Len=0
17	0.10709500	192.168.1.10	192.168.1.100	COTP	113	DT TPDU (0) EOT
18	0.10963700	192.168.1.100	192.168.1.10	TCP	60	102→50502 [ACK] Seq=181 Ack=206 Win=2048 Len=0
19	0.13588500	192.168.1.100	192.168.1.10	COTP	87	DT TPDU (0) EOT
20	0.13612400	192.168.1.10	192.168.1.100	COTP	61	DT TPDU (0) [COTP fragment, 0 bytes]
21	0.13852900	192.168.1.100	192.168.1.10	TCP	60	102→50502 [ACK] Seq=214 Ack=213 Win=2048 Len=0
22	0.15293700	192.168.1.100	192.168.1.10	COTP	111	DT TPDU (0) EOT
23	0.15311600	192.168.1.10	192.168.1.100	COTP	61	DT TPDU (0) [COTP fragment, 0 bytes]
24	0.15434900	192.168.1.10	192.168.1.100	COTP	113	DT TPDU (0) EOT
25	0.15579700	192.168.1.100	192.168.1.10	TCP	60	102→50502 [ACK] Seq=271 Ack=220 Win=2048 Len=0

2. WIRESHARK CAPTURE FOR TEST 2


test2.pcapng [Wireshark 1.12.7 (v1.12.7-0-g7fc8978 from master-1.12)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter:
Expression... Clear Apply Save

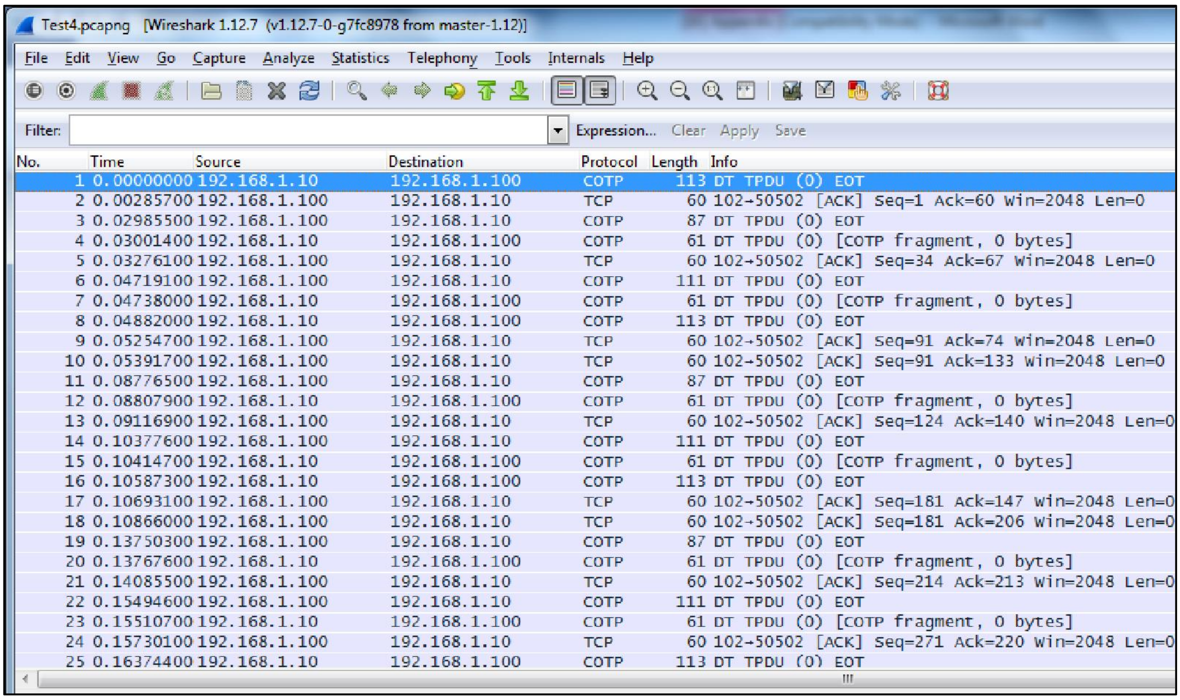
No.	Time	Source	Destination	Protocol	Length	Info
1	0.00000000	192.168.1.100	192.168.1.10	COTP	111	DT TPDU (0) EOT
2	0.00017200	192.168.1.10	192.168.1.100	COTP	61	DT TPDU (0) [COTP fragment, 0 bytes]
3	0.00149900	192.168.1.10	192.168.1.100	COTP	113	DT TPDU (0) EOT
4	0.00261800	192.168.1.100	192.168.1.10	TCP	60	102→50502 [ACK] Seq=58 Ack=8 Win=2048 Len=0
5	0.00560400	192.168.1.100	192.168.1.10	TCP	60	102→50502 [ACK] Seq=58 Ack=67 Win=2048 Len=0
6	0.03671100	192.168.1.100	192.168.1.10	COTP	87	DT TPDU (0) EOT
7	0.03687700	192.168.1.10	192.168.1.100	COTP	61	DT TPDU (0) [COTP fragment, 0 bytes]
8	0.04022600	192.168.1.100	192.168.1.10	TCP	60	102→50502 [ACK] Seq=91 Ack=74 Win=2048 Len=0
9	0.05605900	192.168.1.100	192.168.1.10	COTP	111	DT TPDU (0) EOT
10	0.05621200	192.168.1.10	192.168.1.100	COTP	61	DT TPDU (0) [COTP fragment, 0 bytes]
11	0.05737000	192.168.1.10	192.168.1.100	COTP	113	DT TPDU (0) EOT
12	0.05918200	192.168.1.100	192.168.1.10	TCP	60	102→50502 [ACK] Seq=148 Ack=81 Win=2048 Len=0
13	0.06070600	192.168.1.100	192.168.1.10	TCP	60	102→50502 [ACK] Seq=148 Ack=140 Win=2048 Len=0
14	0.08723000	192.168.1.100	192.168.1.10	COTP	87	DT TPDU (0) EOT
15	0.08743400	192.168.1.10	192.168.1.100	COTP	61	DT TPDU (0) [COTP fragment, 0 bytes]
16	0.09033800	192.168.1.100	192.168.1.10	TCP	60	102→50502 [ACK] Seq=181 Ack=147 Win=2048 Len=0
17	0.10432400	192.168.1.100	192.168.1.10	COTP	111	DT TPDU (0) EOT
18	0.10449700	192.168.1.10	192.168.1.100	COTP	61	DT TPDU (0) [COTP fragment, 0 bytes]
19	0.10833600	192.168.1.100	192.168.1.10	TCP	60	102→50502 [ACK] Seq=238 Ack=154 Win=2048 Len=0
20	0.11469400	192.168.1.10	192.168.1.100	COTP	113	DT TPDU (0) EOT
21	0.11671800	192.168.1.100	192.168.1.10	TCP	60	102→50502 [ACK] Seq=238 Ack=213 Win=2048 Len=0
22	0.14435600	192.168.1.100	192.168.1.10	COTP	87	DT TPDU (0) EOT
23	0.14452100	192.168.1.10	192.168.1.100	COTP	61	DT TPDU (0) [COTP fragment, 0 bytes]
24	0.14727200	192.168.1.100	192.168.1.10	TCP	60	102→50502 [ACK] Seq=271 Ack=220 Win=2048 Len=0
25	0.16086300	192.168.1.100	192.168.1.10	COTP	111	DT TPDU (0) EOT

3. WIRESHARK CAPTURE FOR TEST 3


Test3.pcapng [Wireshark 1.12.7 (v1.12.7-0-g7fc8978 from master-1.12)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter:
Expression... Clear Apply Save

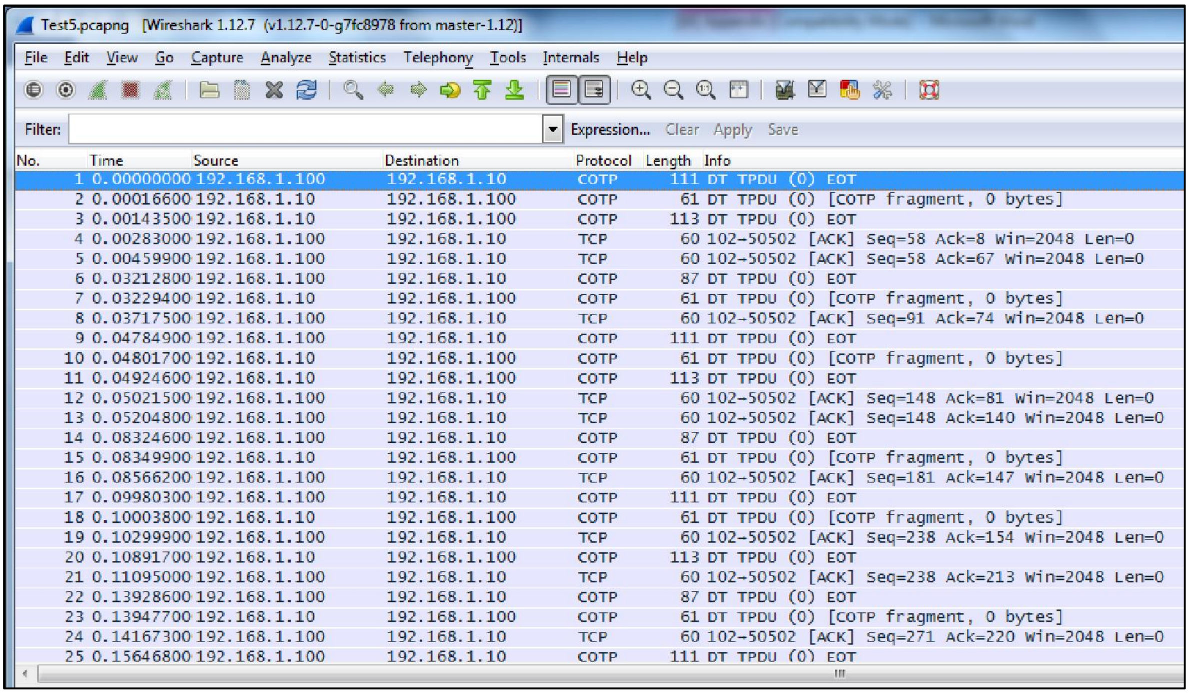
```
No.  Time        Source         Destination    Protocol Length Info
   1 0.00000000 192.168.1.100  192.168.1.10   COTP      87 DT TPDU (0) EOT
   2 0.00018500 192.168.1.10   192.168.1.100  COTP      61 DT TPDU (0) [COTP fragment, 0 bytes]
   3 0.00395600 192.168.1.100  192.168.1.10   TCP       60 102→50502 [ACK] Seq=34 Ack=8 Win=2048 Len=0
   4 0.01601100 192.168.1.100  192.168.1.10   COTP     111 DT TPDU (0) EOT
   5 0.01617900 192.168.1.10   192.168.1.100  COTP      61 DT TPDU (0) [COTP fragment, 0 bytes]
   6 0.01756400 192.168.1.10   192.168.1.100  COTP     113 DT TPDU (0) EOT
   7 0.01891800 192.168.1.100  192.168.1.10   TCP       60 102→50502 [ACK] Seq=91 Ack=15 Win=2048 Len=0
   8 0.02065800 192.168.1.100  192.168.1.10   TCP       60 102→50502 [ACK] Seq=91 Ack=74 Win=2048 Len=0
   9 0.04898100 192.168.1.100  192.168.1.10   COTP      87 DT TPDU (0) EOT
  10 0.04922100 192.168.1.10   192.168.1.100  COTP      61 DT TPDU (0) [COTP fragment, 0 bytes]
  11 0.05231900 192.168.1.100  192.168.1.10   TCP       60 102→50502 [ACK] Seq=124 Ack=81 Win=2048 Len=0
  12 0.06699900 192.168.1.100  192.168.1.10   COTP     111 DT TPDU (0) EOT
  13 0.06727700 192.168.1.10   192.168.1.100  COTP      61 DT TPDU (0) [COTP fragment, 0 bytes]
  14 0.07100400 192.168.1.100  192.168.1.10   TCP       60 102→50502 [ACK] Seq=181 Ack=88 Win=2048 Len=0
  15 0.07888300 192.168.1.10   192.168.1.100  COTP     113 DT TPDU (0) EOT
  16 0.08151800 192.168.1.100  192.168.1.10   TCP       60 102→50502 [ACK] Seq=181 Ack=147 Win=2048 Len=0
  17 0.10724900 192.168.1.100  192.168.1.10   COTP      87 DT TPDU (0) EOT
  18 0.10741500 192.168.1.10   192.168.1.100  COTP      61 DT TPDU (0) [COTP fragment, 0 bytes]
  19 0.10974800 192.168.1.100  192.168.1.10   TCP       60 102→50502 [ACK] Seq=214 Ack=154 Win=2048 Len=0
  20 0.12795500 192.168.1.100  192.168.1.10   COTP     111 DT TPDU (0) EOT
  21 0.12811100 192.168.1.10   192.168.1.100  COTP      61 DT TPDU (0) [COTP fragment, 0 bytes]
  22 0.12949200 192.168.1.10   192.168.1.100  COTP     113 DT TPDU (0) EOT
  23 0.13046700 192.168.1.100  192.168.1.10   TCP       60 102→50502 [ACK] Seq=271 Ack=161 Win=2048 Len=0
  24 0.13294500 192.168.1.100  192.168.1.10   TCP       60 102→50502 [ACK] Seq=271 Ack=220 Win=2048 Len=0
  25 0.16181700 192.168.1.100  192.168.1.10   COTP      87 DT TPDU (0) EOT
```

4. WIRESHARK CAPTURE FOR TEST 4


Test4.pcapng [Wireshark 1.12.7 (v1.12.7-0-g7fc8978 from master-1.12)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter:
Expression... Clear Apply Save

```
No.  Time        Source         Destination    Protocol Length Info
   1 0.00000000 192.168.1.10   192.168.1.100  COTP     113 DT TPDU (0) EOT
   2 0.00285700 192.168.1.100  192.168.1.10   TCP       60 102→50502 [ACK] Seq=1 Ack=60 Win=2048 Len=0
   3 0.02985500 192.168.1.100  192.168.1.10   COTP      87 DT TPDU (0) EOT
   4 0.03001400 192.168.1.10   192.168.1.100  COTP      61 DT TPDU (0) [COTP fragment, 0 bytes]
   5 0.03276100 192.168.1.100  192.168.1.10   TCP       60 102→50502 [ACK] Seq=34 Ack=67 Win=2048 Len=0
   6 0.04719100 192.168.1.100  192.168.1.10   COTP     111 DT TPDU (0) EOT
   7 0.04738000 192.168.1.10   192.168.1.100  COTP      61 DT TPDU (0) [COTP fragment, 0 bytes]
   8 0.04882000 192.168.1.10   192.168.1.100  COTP     113 DT TPDU (0) EOT
   9 0.05254700 192.168.1.100  192.168.1.10   TCP       60 102→50502 [ACK] Seq=91 Ack=74 Win=2048 Len=0
  10 0.05391700 192.168.1.100  192.168.1.10   TCP       60 102→50502 [ACK] Seq=91 Ack=133 Win=2048 Len=0
  11 0.08776500 192.168.1.100  192.168.1.10   COTP      87 DT TPDU (0) EOT
  12 0.08807900 192.168.1.10   192.168.1.100  COTP      61 DT TPDU (0) [COTP fragment, 0 bytes]
  13 0.09116900 192.168.1.100  192.168.1.10   TCP       60 102→50502 [ACK] Seq=124 Ack=140 Win=2048 Len=0
  14 0.10377600 192.168.1.100  192.168.1.10   COTP     111 DT TPDU (0) EOT
  15 0.10414700 192.168.1.10   192.168.1.100  COTP      61 DT TPDU (0) [COTP fragment, 0 bytes]
  16 0.10587300 192.168.1.10   192.168.1.100  COTP     113 DT TPDU (0) EOT
  17 0.10693100 192.168.1.100  192.168.1.10   TCP       60 102→50502 [ACK] Seq=181 Ack=147 Win=2048 Len=0
  18 0.10866000 192.168.1.100  192.168.1.10   TCP       60 102→50502 [ACK] Seq=181 Ack=206 Win=2048 Len=0
  19 0.13750300 192.168.1.100  192.168.1.10   COTP      87 DT TPDU (0) EOT
  20 0.13767600 192.168.1.10   192.168.1.100  COTP      61 DT TPDU (0) [COTP fragment, 0 bytes]
  21 0.14085500 192.168.1.100  192.168.1.10   TCP       60 102→50502 [ACK] Seq=214 Ack=213 Win=2048 Len=0
  22 0.15494600 192.168.1.100  192.168.1.10   COTP     111 DT TPDU (0) EOT
  23 0.15510700 192.168.1.10   192.168.1.100  COTP      61 DT TPDU (0) [COTP fragment, 0 bytes]
  24 0.15730100 192.168.1.100  192.168.1.10   TCP       60 102→50502 [ACK] Seq=271 Ack=220 Win=2048 Len=0
  25 0.16374400 192.168.1.10   192.168.1.100  COTP     113 DT TPDU (0) EOT
```

5. WIRESHARK CAPTURE FOR TEST 5



```
No.  Time       Source         Destination    Protocol Length Info
   1 0.00000000 192.168.1.100  192.168.1.10   COTP     111 DT TPDU (0) EOT
   2 0.00016600 192.168.1.10   192.168.1.100  COTP      61 DT TPDU (0) [COTP fragment, 0 bytes]
   3 0.00143500 192.168.1.10   192.168.1.100  COTP     113 DT TPDU (0) EOT
   4 0.00283000 192.168.1.100  192.168.1.10   TCP       60 102→50502 [ACK] Seq=58 Ack=8 Win=2048 Len=0
   5 0.00459900 192.168.1.100  192.168.1.10   TCP       60 102→50502 [ACK] Seq=58 Ack=67 Win=2048 Len=0
   6 0.03212800 192.168.1.100  192.168.1.10   COTP      87 DT TPDU (0) EOT
   7 0.03229400 192.168.1.10   192.168.1.100  COTP      61 DT TPDU (0) [COTP fragment, 0 bytes]
   8 0.03717500 192.168.1.100  192.168.1.10   TCP       60 102→50502 [ACK] Seq=91 Ack=74 Win=2048 Len=0
   9 0.04784900 192.168.1.100  192.168.1.10   COTP     111 DT TPDU (0) EOT
  10 0.04801700 192.168.1.10   192.168.1.100  COTP      61 DT TPDU (0) [COTP fragment, 0 bytes]
  11 0.04924600 192.168.1.10   192.168.1.100  COTP     113 DT TPDU (0) EOT
  12 0.05021500 192.168.1.100  192.168.1.10   TCP       60 102→50502 [ACK] Seq=148 Ack=81 Win=2048 Len=0
  13 0.05204800 192.168.1.100  192.168.1.10   TCP       60 102→50502 [ACK] Seq=148 Ack=140 Win=2048 Len=0
  14 0.08324600 192.168.1.100  192.168.1.10   COTP      87 DT TPDU (0) EOT
  15 0.08349900 192.168.1.10   192.168.1.100  COTP      61 DT TPDU (0) [COTP fragment, 0 bytes]
  16 0.08566200 192.168.1.100  192.168.1.10   TCP       60 102→50502 [ACK] Seq=181 Ack=147 Win=2048 Len=0
  17 0.09980300 192.168.1.100  192.168.1.10   COTP     111 DT TPDU (0) EOT
  18 0.10003800 192.168.1.10   192.168.1.100  COTP      61 DT TPDU (0) [COTP fragment, 0 bytes]
  19 0.10299900 192.168.1.100  192.168.1.10   TCP       60 102→50502 [ACK] Seq=238 Ack=154 Win=2048 Len=0
  20 0.10891700 192.168.1.10   192.168.1.100  COTP     113 DT TPDU (0) EOT
  21 0.11095000 192.168.1.100  192.168.1.10   TCP       60 102→50502 [ACK] Seq=238 Ack=213 Win=2048 Len=0
  22 0.13928600 192.168.1.100  192.168.1.10   COTP      87 DT TPDU (0) EOT
  23 0.13947700 192.168.1.10   192.168.1.100  COTP      61 DT TPDU (0) [COTP fragment, 0 bytes]
  24 0.14167300 192.168.1.100  192.168.1.10   TCP       60 102→50502 [ACK] Seq=271 Ack=220 Win=2048 Len=0
  25 0.15646800 192.168.1.100  192.168.1.10   COTP     111 DT TPDU (0) EOT
```