ANALYSIS OF PROFIBUS COMMUNICATION USING PROCESS AUTOMATION AND DECENTRALISED PERIPHERY AGAINST CONVENTIONAL (4-20MA)

BY

BLESSING ZITHULELE TEFSA NTSHANGASE
STUDENT NO: 4727 0411

SUBMITTED IN ACCORDANCE WITH THE REQUIREMENTS FOR THE PARTIAL FULFILMENT FOR THE

DEGREE OF

MASTER OF TECHNOLOGY IN ELECTRICAL ENGINEERING

UNIVERSITY OF SOUTH AFRICA

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

25 October 2016

DECLARATION

I declare that this is my own, unaided work. It is being submitted for the Magister Technologiae in Electrical Engineering in the Department of Electrical Engineering at university of South Africa. It has not been submitted before for any degree, diploma or other examination at any other education institution.

____

B.Z.T Ntshangase

At___Durban_____ on this __25__day of __October___2016

ACKNOWLEDGEMENT

I would like to thank Almighty God for providing me with the courage and endurance to complete my research study. Following, I would like to thank my family. Especially my wife Mpumelelo, the toughest pillar of my life without whom nothing would have been possible. Not to mention my beautiful daughter Anelisa and my Son Akwande who were responsible for injecting fresh energy every time I am feeling exhausted. I could not have been able to complete this thesis without the expert advice and support from my supervisors Prof Naidoo from Mangosuthu University of Technology and Prof. MO Ohanga from Vaal University of Technology. They have been there to help me and permanently given their best guidance to complete the study. I would like thank my co-supervisor: Prof. Naidoo Mangosuthu University of Technology (MUT) for his tremendous contribution with regards to the research, ensuring that the research lab facilities were available when required and adequate analysis on critical aspects for the analysis of PROFIBUS Communication using Process Automation and Decentralised Periphery against Conventional (4-20mA). In addition, more details about how each component were expectable to operate and how they were functioning together, as it was the most heavily addressed. This research would have been impossible without the support of my supervisor: Prof. MO Ohanga Vaal University of Technology as well as for his excellent supervision and mentoring experience including his availability and guidance. I am obliged for the opportunity and privilege to work on this research. Special thanks go to Mangosuthu University of Technology for allowing me access to the research facilities. I will also like to thank Shane Govender from Siemens for assisting with special tools when experimental tests were conducted. In addition I would also like to thank University of South Africa for approving this research study and concluding research successfully. Finally I would like to thank the Electrical Engineering Department at University of South Africa for the privilege to register and study in the department.

ABSTRACT

The research Analyses, PROFIBUS Communication using Process Automation and Decentralized Periphery against Conventional (4-20mA) was based on the process plant constructed in 2012 by staff and students in the Department of Instrumentation and control at Mangosuthu University of Technology. Further work was not done including configuring the entire process plant, testing all devices and display the image of the process plant on human machine interface. The process plant operation was never tested and the research topic has not been attempted before. The research was conducted to improve the existing plant to full operation or functional project. The research study was conducted in March 2013 by BZT Ntshangase, it was established that both DP flowmeter and PA flowmeter were not tested before, not calibrated correctly and both drives which are (MM440 and MM420) were not setup to control both pump1 and pump 2 in manual or auto mode. The Programmable Logic Controller (PLC) was not configured to communicate with all devices on the network. PROFIBUS-DP and PROFIBUS-PA devices were not set up; input measurements were not scaled to read within certain limits. All devices used in the process plant were not assigned unique addresses for the network and DSG files were not installed so that Programmable logic controller (PLC) would identify all devices on the PROFIBUS network. The research was based on analysis of PROFIBUS communication using process automation and decentralized periphery against conventional (4-20mA), where one flowmeter was communicating with a PLC via PROFIBUS-DP and the other flowmeter was communicating through PLC via ET-200M to a DP/PA coupler. Research objective was to examine the time response between the two signals, data transmission, network configurations and their communication protocols and including transmission rate for both networks. Process plant components used in the research were tested for linear scaling, reliability, generalizability and validity. The reason for performing these tests was to produce consistent results and to checks how similar results are if the research was repeated under similar circumstances. During testing, respectable results were achieved. All simulated results were compared with the real-time results and then a conclusion drawn based on the obtained information and facts. Project design, implementation, test procedures and test results were achieved because prototype performed as per design and research objectives were achieved. Simulation tests were conducted and the obtained results analysed. The achieved results showed that the proposed solution or the prototype system performed as per design. The experimental results could be useful to other researchers in the future. At the end of the study conclusion and some recommendations for further studies are discussed efficiently to utilize resources in the process plant verification and validation.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF TERMINOLOGIES

PROFIBUS	Process field bus
PROFIBUS-DP	Decentralized Periphery
FMS	Field bus Message Specification
PROFIBUS-PA	Process Automation
PLC	Programmable Logic Controller
VSD	Variable Speed Drives
NCS	Network Control Systems
SDN	Send Data with Number
P&ID	Process and Identification
I/O’s	Inputs and Outputs
FIC	Flow Indicator Controller
LT	Level Transmitter
ADD1&2	Additives 1 and 2
VSP1&2	Variable Speeds Pump 1 and 2
FT	Flow Transmitter
S7-300	Siemens Type PLC
PNC	Process near Component
FDL	Fieldbus data link
FISCO model	Fieldbus Intrinsically Safe Concept
RAM	Random Access Memory
STL	Program instructions
LAD	Ladder Logic
DSG	General Station Description
LT	Level Transmitter
EEPROM	Electronically erasable programmable read only memory
UVEPROM	Ultraviolet erasable programmable read only memory
OSI	Open System Interconnection
M1	DP master (Class 1)
IEC	International standard
AUX	Auxiliary

TFT	Thin Film Transistor
OB	Organization blocks
SFB	System function blocks
SFC	System functions
DB	Data block
FC	Functional block
FM	Siemens Flow Meter Verificator
IGBT	Insulated Gate Bipolar Transistor
AOP	The Advanced operator panel
V/F	Voltage/ Frequency
DC	Direct current
AC	Alternately current
DCS	Distributed control system
FC	Functions block
ASCII	American standard code for information interchange

CHAPTER ONE: INTRODUCTION

Process automation control is the application of control theory for regulation of processes without direct human intervention or is the control systems for operating equipment such as machinery, processes in factories, boilers and heat treating ovens, switching on telephone networks, navigation and stabilization of ships, aeroplane and other applications with minimal or reduced human intervention. Process automation is the control systems for operating equipment such as machinery, processes in factories, boilers and heat treating ovens, switching on telephone networks, navigation and stabilization of ships, aeroplane and other applications. Process automation involves a wide range of technologies and sciences, and they are used in an unprecedented number of applications. Examples range from the control of heating, cooling, and hot water systems in homes and offices to chemical and automotive instrumentation and process control. Process automation focused on the principle and operation of measuring instruments and process control that are used in design and configuration of automated systems in electrical. Process automation systems have used a combination of proprietary and open digital network to provide improved information availability and increased throughput and performance. Process automation comprises set of devices that manages commands, regulates the behaviour of systems through special communication [1] and [2].

The importance of process automation technology continues to increase in the process industries and information between field devices and controllers through communication and network technology. In this research the digital communication field bus selected was called PROFIBUS. PROFIBUS network was used for connecting/linking all PROFIBUS-DP and PROFIBUS-PA devices to the main controller in this case programmable logic control was used and all data was digitally encoded and transmitted. PROFIBUS is a smart, field-bus technology for linking process control and plant automation modules. Instead of running individual cables from a main controller to each sensor and actuator, it is also used to connect all devices, with high speed, bi-directional, serial messaging used for transfers of devices information .Devices on the PROFIBUS network connect to a central line and once connected, these devices can communicate information in an efficient manner, but can go beyond automation communications [3] and [4].

The research study was proposed to cover all aspects of process automation, such as sensing flowmeter, the transmission of data and exchange of information between filed devices and the programmable logic controller and recording of the sensed signal, drives, pumps, HMI, for signal evaluation, and the control of the process plant for a quality and uniform product, in this case water was used as the product. The experimental process plant used in the research was commissioned by Mangosuthu University of Technology students around 2012 but no comparison between PROFIBUS and conventional 4-20mA analysis was conducted. The main objective of the research was to analyse this comparison of PROFIBUS communication using process automation decentralized periphery against conventional (4-20mA). As seen in figure 1, flow and level was controlled by using a Programmable logic controller (PLC). Three tanks contain process mediums. Additive 1 and additive 2 tanks contain two different mediums and the third tank was a mixer of the two mediums but in this case water was used. Pumps were used to pump water from additive 1 and additive 2 to the mix tank and both pumps were controlled by variable speed drives (VSD's). Both flow transmitters then measures a process fluid passing at a specific point which was then transmitted to a flow indicator controller to regulate flow to a desired flow rate. The level transmitter sends a signal to a level indicator controller which was a set point to flow indicator controllers. The type of programmable logic controller that was used in the research to control the process plant was called a Siemens S7-300 PLC. All input and output readings for the process plant were observed and monitored using human interface machine (HMI), as shown in 2.

Figure 1: Installed process plant

Flow indication control 1 and flow indication control 2 controls the respective process by measuring flow rates through flowmeter 1 and flowmeter 2 and regulating them via variable speed drive 1 and variable speed drive 2. Level transmitter measures the level of mix tank and when level reaches the desired level, the PLC stop both pumps to prevent over flow. ET200M was a field interface used as the connection medium for the bus cable between the hardwired flowmeter and the programmable logic controller. The two flow transmitters were used to measure a process liquid passing at a specific point in dissimilar loops which was then transmitted to a flow indicator controller to regulate flow to a desired flow rate. Programmable logic control generates a set point value according to its program and sends a set point to the drives to increase or decrease the speed on the pumps. A Siemens S7-300 PLC was used in the research. Both drives' have a built-in PID controller in the control board. When the PID control was activated, a process reference (set point) was connected to the drive instead of a speed reference.

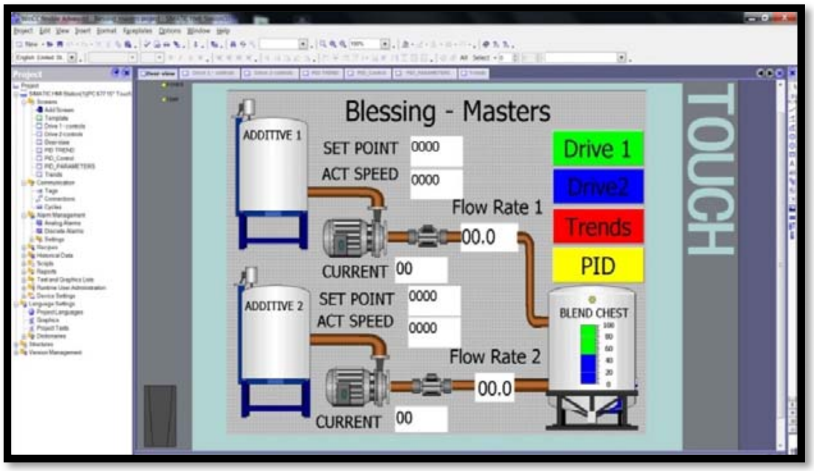


Figure 2: Process plant Display Page

An actual value (process feedback) was also brought back to the drive. The process PID control adjusts the drive speed in order to keep the measured process quantity (actual value) at the desired level (reference) by regulating two pumps to vary the flow rate of two different process variables under the same process condition. The research analysed variations between the two communication protocols. The objective of the research was to compare conventional 4-20mA and updated PROFIBUS communication protocols. In order to fulfil the process plant design specification, the important factors were noted and all experimental tests were also done to verified the prototype if met the design requirement. The detailed analysis of the test results proved that the prototype performed as designed. The process plant linear scaling results are shown in table 1. The researcher successfully captured real-time data communication results and concluded detailed analysis.

Table 1: linear scaling and value range

Device Name	Range and linear scaling	Units
Flowmeter 1	0-6.5 l/m	L/m
Flowmeter 2	0-6.5 l/m	L/m
Drive 1	1-10V; 0-50 Hz	Hz
Drive 2	1-10V; 0-50 Hz	Hz
Pump1	0-1481rpm	rpm
Pump 2	0-1481rpm	rpm
PLC	0-6384	Double word
HMI	0-100%	Percentage

Further detailed of the dissertation chapters are as follows; Chapter 1 provided a brief introduction of the study, outline process automation and stating the results achieved. Chapter 2 was a short discussion on the process automation but, serious evaluation/investigation of the other works and approaching solution(s) different from the others with valid reasons. Chapter 3 discusses the literature review and analyses all necessary information that was used in the research and further discusses the background and statement of the problem, research objectives and expectations and proposition. Chapter 4 discusses the design of the proposed solution, problem to be solved, evaluation of various concepts, user requirements and preferred solution. Chapter 5 provides a short discussion about profibus PA and DP and also clarify the profibus technologies. Chapter 6 discusses various type of measuring devices used in the research and highlights the important aspects in detailed design. Chapter 7 focuses on the introduction of the HMI-panel control and discussed main concepts in hardware, software installation, system configurations and network structure. Chapter 8 discusses the various aspects of PROFIBUS communication protocols. Chapter 9 explain how implementation and set-up procedures were conducted for each component used in the research and additional discusses the experiment evaluations were an interesting problem was addressed and detailed calibration procedures, installation procedures, test procedures, and other procedures were achieved. Chapter 10 provided various tests analysis of all process plant equipment’s including simulation tests, experimental tests and results achieved. Chapter 11 explains the analysis of test results and defines if the prototype performed as designed after testing according to specifications. Chapter 12 discusses the valuable lessons learned from this dissertation and including conclusion and recommendations. The primary reason for writing this dissertation was that the researcher felt that there was no clear, concise and understanding of PROFIBUS communication protocol between PROFIBUS DP and PROFIBUS PA. Every effort has been prepared to confirm that the dissertation was accurate, simply readable and understandable.

CHAPTER TWO: PROCESS AUTOMATION

2.1 Introduction

Process automation control is the application of control theory for regulation of processes without direct human intervention or is the control systems for operating equipment such as machinery, processes in factories, boilers and heat treating ovens, switching on telephone networks, navigation and stabilization of ships, aeroplane and other applications with minimal or reduced human intervention.

As industrial process automation systems grow more sophisticated in terms of the knowledge and algorithms they use, as they cover larger areas of operation comprising several units in factories and also involves significant amount of hardware technologies, related to Instrumentation and sensing, actuation and drives, electronics for signal conditioning, communication (transmission technology and communications protocol) and display, including mechanical, hydraulic and pneumatics systems. There are two common classes of control systems, open loop control systems and closed loop control systems. In open loop control systems output is generated based on inputs. In closed loop control systems current output is taken into consideration and corrections are made based on feedback. A closed loop system is also called a feedback control system.

In the simplest type of a process automation control, a controller compares a measured value of a process with a desired set value, and the processes resulting error signal to change some input to the process, in such a way that the process stays at its set point despite disturbances. The mathematical basis of control theory was begun in the 18th century, and advanced rapidly in the 20th. In today's fast-moving, highly competitive industrial world, a company must be flexible, cost effective and efficient if it wishes to survive. In the process automation and manufacturing industries, this has resulted in a great demand for industrial control systems/ process automation in order to modernise operations in terms of speed, reliability, quality and product output. Automation plays an increasingly important role in the world economy and in daily experience [5] and [6].

The biggest benefit of process automation is that it saves labour, however, it is also used to save energy and materials and to improve quality, accuracy and precision and it can be achieved through fast industrial digital communication systems. There are many types of fieldbus in widespread which are Device Net, Interbus-S, Foundation Fieldbus, PROFIBUS and many others). In this research

PROFIBUS was used as main communication protocol for fast data exchange between all PROFIBUS devices and the main controller (PLC) [7].

2.2 PROFIBUS

PROFIBUS is an international fieldbus network communications standard for linking process control and plant automation modules. It allows simple linking of devices from many different vendors and, reduces installation, commissioning, fault-finding and associated plant downtime costs [7]. The PROFIBUS-DP system is used to handle the communication at the control level and communicate with all PROFIBUS-DP devices similar in this research PLC, HMI, drives, remote I/O's including PROFIBUS–PA devices are communicating using PROFIBUS technology. It was also possible to connect externally powered field devices to this level, like the PA flow meter. PROFIBUS-DP ensures that data are quickly exchanged, whereby in mixed PROFIBUS-DP/PA systems the baud rate supported by the segment coupler is often the limiting factor.

In PROFIBUS-PA systems for Application process automation, a PROFIBUS-DP system is used at the control level for quick transmission of the data and a variant of PROFIBUS-DP, DPV1 is used [8]. The recommended baud rate for this research was 187.5kb/s due to that it was suitable for all nodes on the PROFIBUS network. In addition to the cyclic exchange of data with a PLC, this allows the field devices to be configured via acyclic services. The main objective on this research was to focus on analysis of PROFIBUS communication using process automation and decentralized periphery against conventional (4-20mA) where one flow meter was communicating with a PLC via PROFIBUS-DP and the other flow meter was communicating through PLC via ET-200 to a DP/PA coupler. The purpose of the research was to examine the time response between the two signals, data transmission, network configurations and their communication protocols and including transmission rate for both networks. Achieved results were captured in simulation tests chapter.

Due to the use of serial messaging for transfer of sensor information to a controller and return of any required actuator commands over one common bus link, special tools is necessary to view what is happening on this very important communications line [8]. In this research a PROFIBUS tester 4 tool was used to test the PROFIBUS network communication fault before experimental test was conducted. This tool was needed to verify the bus reliability, the communication waveform shapes and content quality of the signals from each separate device for verification of optimal operation and time response and location of weaknesses and faults. Correct tests procedure were applied at the right location to successfully understand and interpret the results achieved, so allowing rapid and accurate identification of errors, and their correction.

The difference between conventional and PROFIBUS-PA was that the wiring from the field to the junction box was roughly the same: if the measuring points are widely distributed, however, the fieldbus requires decidedly less cable because PROFIBUS-PA devices receives information from controlling stations and send information to controlling stations via two way communication [9]. For conventional wiring, every signal line must be continued from the junction box to the process-near component, e.g. a programmable logic controller, where it terminates in an I/O module.

Every device a separate power supply is required, where necessary, suitable for use with devices in hazardous areas. In contrast, the fieldbus requires a single cable only to carry all information. The bus terminates in a bus coupler that communicates directly with the process near components. Not only cable, but also I/O modules are saved [10]. Since the bus for the research process plant devices was powered from a single intrinsically safe power unit, there was no need for individual isolators and barriers. Communication of the devices with the PLC was done by using PROFIBUS-DP (decentralized periphery). The figure 3 shows the difference between the wiring of a conventional 4-20 mA control system and a fieldbus PA.

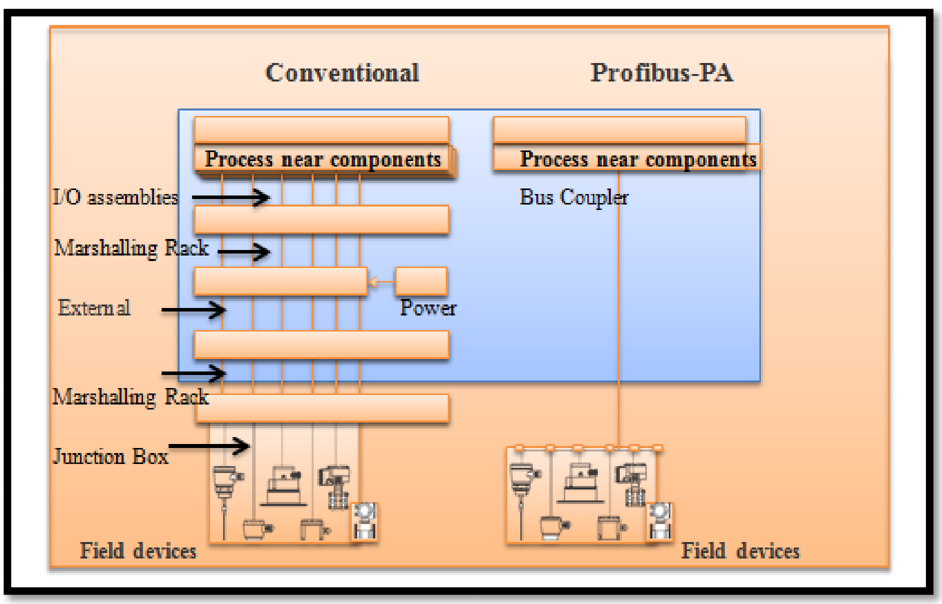


Figure 3: The difference between conventional and PROFIBUS-PA [10]

PROFIBUS was designed to automate an entire plant, regardless of its size or whether the plant is factory automation (composed of discrete input/output) or process automation (made up of analog input/output). It also does not matter if all the sections are local or remote: PROFIBUS can handle it all well. PROFIBUS is an open field bus standard for a wide range of applications in manufacturing and process automation or is a digital communication system for field devices. Fieldbus makes it possible to connect many devices together into a network, often using a single cable. PROFIBUS allows communication between devices of different manufacturers without any special interface adjustment. PROFIBUS can also be used for both high-speed time critical applications and complex communication tasks [10].

PROFIBUS was born out of a combined push by the German government, German companies, and other industry leaders in the late 1980s. Their effort created an automation solution that was not only still viable today, but has led to further solutions. The proud heritage of PROFIBUS allows for many European customers to turn to automation specific to their needs. In 1987, 21 companies and institutions in Germany joined forces to create a new protocol. Their goal was to create a bit-serial Fieldbus system. In order for the system to be viable, they needed to standardize the field device interface. The group, which had taken the name Central Association for the Electrical Industry, completed its goal with the creation of PROFIBUS FMS (Fieldbus Message Specification) [11].

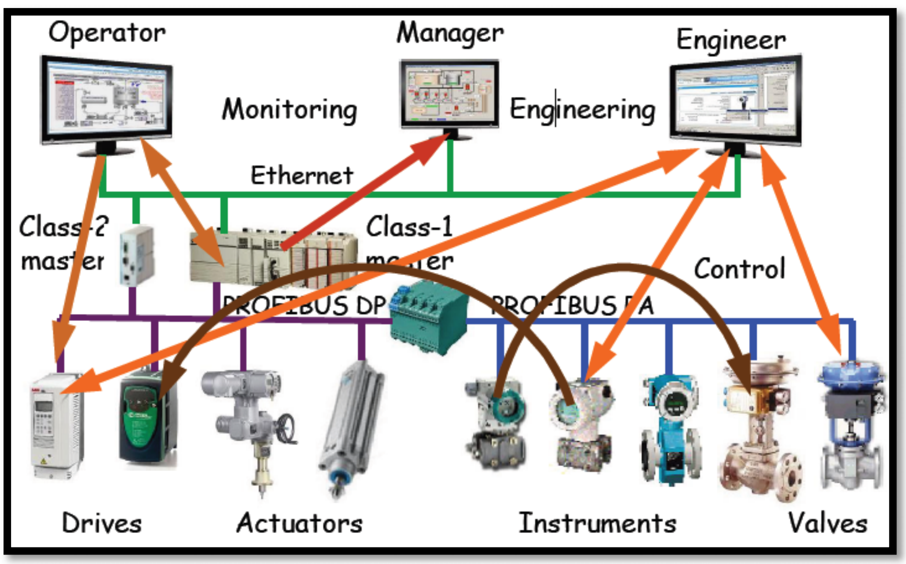


Figure 4: PROFIBUS communication [11]

As seen in figure 4, this new protocol satisfied standardization of Industrial automation through a protocol capable of sending complex communications. In 1993, the group introduced a new standard, PROFIBUS DP (Decentralized Periphery). This new version featured more simplicity, including easier configuration and faster messaging. Through its continuing further technical developments, PROFIBUS network is still the industrial communication system for the future because it’s two-way communication so that devices can receive information from controlling stations and also can send information to controlling stations. The data can be simple – i.e. digital on/off signals or can be complex such from and to sophisticated instruments such on-line signal converters or sensors. Fieldbus cable replaces individual, point-to-point device wiring. PROFIBUS family consists of three compatible versions offering very high integrity and a capability appropriate to the need, which are [11]:

(i) PROFIBUS DP- Decentralized Periphery, which is a low-cost, high speed field level communications. About 90% of current PROFIBUS applications are DP. It is designed and optimized especially for communication between automation systems and decentralized filed devices, such as sensors and actuators via a centralized controller in production (factory) automation applications. PROFIBUS-DP allowed only one master that communicates via the master-slave method. The extended version DPV1 (decentralised peripherally version 1) allows up to 127 participants including up to32 masters. It is usually runs over violet sheathed, two core screened cable, at speeds from 9.6 kbps up to 12 Mbps, and using RS 485 balanced transmission.

(ii) PROFIBUS FMS- Fieldbus Message Specification was designed to communicate between programmable controllers and PCs, sending complex information between them and high-end applications level communications. Used at cell or controller level. Provides object oriented transmission of structured data, loading and control of programs, alarm services, etc. This variant provides the user with a variety of functions which, however, makes it more complex to other variants, hence is no longer used extensively [12].

(iii) PROFIBUS-PA is a protocol designed for process automation and also is a type of PROFIBUS-DP Application profile. PROFIBUS PA standardizes the process of transmitting measured data and was developed specifically for the process industry, cost effective two-wire connection carrying both power and data. It is used to monitor measuring equipment via a process control system in process automation applications. It is particularly used in explosion/ hazardous areas because it allows power to be delivered over the bus to filed instruments, while limiting current flows so that explosive conditions are not created, even if malfunction occurs. The number of devices attached to a PA segment was limited by this feature. PA has a data transmission of 31.25kbit/s [12].

CHAPTER THREE: PROFIBUS COMMUNICATION USING AUTOMATION AND DECENTRALISED PERIPHERY AGAINST CONVENTIONAL (4-20 mA)

3.1 Introduction

In order to produce a product with consistently high quality, tight process control was necessary and understanding of what was expected from the author. The purpose of this project was to conduct a logical, in-depth analysis of the process plant set-ups covered by all situations during the stating phase of the project, building upon the detailed documentation of issues, variations, barriers, solutions and implementation plans, to identify and choose best solution. The methodological approach used by the author to achieve the objectives and complete this project consisted of core steps.

3.2 Background and Problem Statement

When a research was carried out, it was found out that many industries are experiencing problems with increased on down time, product quality issues, difficult to upgrade or expand the plants and the technical drawings of the plants are complicated for engineering team because of complex wirings in Junction boxes. Also found out that the number of injuries reported at industries increases every year and the main cause was incorrect functioning of critical instruments in hazardous areas due to limited device information.

Most industries that have upgraded to PROFIBUS networks system have fewer problems because PROFIBUS is a single cable that carries all devices information for plant or section of the plant and makes it easy to eliminate the problem before it occurs. According to Pierson, Lynda L "*Broader Fieldbus Standards Will Improve System Functionality Control Engineering*" PROFIBUS has the largest portfolio of products in the fieldbus world; in 2008 it reached 2400 products and services. Many industries that have upgraded to PROFIBUS network has many advantages over conventional point-to-point wiring because PROFIBUS communication has additional information such as calibration and configuration data, diagnostic and test information, device documentation such as device tag numbers; serial numbers and service history can be communicated over the network. Equipment maintenance and servicing become more centralised and it is not expensive to install, not complex to maintain and easy to fault-finding. PROFIBUS system provides more information for each device for operators and engineers to control and monitor. Devices are self-monitored by the bus master and status of each device can be checked in the control room. PROFIBUS transmission protocol facilitates very fast cyclical data exchange between the control system CPU and the distributed I/O devices. There is a need for a fast reliable communication system for hazardous areas like petroleum industry, pulp and paper and nuclear power plant to protect human and machine [12].

The process plant was built and installed by S4 students in the department of Electrical Engineering at Mangosuthu University of Technology, The comparison of two signals was not tried and it was not put in to practical. The motivation for the research was because the research topic:
Analysis of PROFIBUS communication using process automation and decentralized periphery against conventional (4-20mA). The research study was not attempted before; this was revealed at the early research stage. There were no records of any information or work done related to this research. The aim for this research was to improve the existing plant to full operation. At this stage the plant was not automated. When research was conducted, It was established that both flowmeters (PROFIBUS-DP and hardwired flowmeter or PA flowmeter) were not tested before, not configured correctly and both drives were also not setup to control both pump1 and pump2 in manual or auto mode. PLC was not configured to communicate with all devices. All devices were not assigned unique address to be recognised by programmable logic controller on the network.

(i) Two control loops are to be created, tuned and able to switch either Auto/Man.
(ii) Configure PROFIBUS network and assign address for all nodes on the network.
(iii) Create software/program for the plant and work according to proposed design and configure PROFIBUS DP and PROFIBUS PA.
(iv) Design graphics and link with PLC program and all reading to display on HMI.
(v) Install dsg files for all field devices used on the project in order for the PLC to recognize their standards.
(vi) Test system communication and including network and design structure of the network and test field devices.
(vii) Set both drives for PROFIBUS mode and transfer graphics to HMI via MPI cable and Install two flow meters, calibrate, scaling and configure them PROFIBUS network.

Since the research focused on analysis of PROFIBUS Communication using Process Automation and Decentralised Periphery against Conventional (4-20mA), the following was to be achieved so that the process plant operates as designed:

(i) One Flow meter was connected directly to the PROFIBUS DP network and other one was connected to PROFIBUS PA via PROFIBUS DP/PA coupler, but both flowmeters were not assigned unique address to communicate with the PLC.
(ii) Variable speed drivers was set up for PROFIBUS and configured to the PLC.
(iii) Correct DSG files for the right device were downloaded to the proper PLC directory in order for each device to support transmission rates, length of the input/output data to be exchanged and device to be recognised by the system.
(iv) Evaluate two network protocols, PROFIBUS DP and PROFIBUS PA associated with conversion of the transmission rate from 187.5 Kbit/s to 31.25 Kbit/s and time response.
(v) ET200 In HWConfig must give an address to each slave and Configure a second PROFIBUS subnet in the existing project with the CP 342-5 as DP master and the existing ET 200 component as slaves.
(vi) Configure an output to an ET 200 using the DP-SEND/DP-RECV blocks.
(vii) To be able to transfer variables due to a write or read request in the neutral FMS format, format information must be created and stored on the PROFIBUS CP.
(viii) Write a PLC program so that the project can be in full operation.

Once the above problems were resolved, the research task was started whose main objective was to perform a comparative study between PROFIBUS DP (decentralized periphery) and conventional 4-20mA analogue communication in terms of fast data exchange between PLC and devices in PROFIBUS DP and PROFIBUS DP/PA. In addition, to focussing on communication structures of PROFIBUS DP/PA-Interface transitions and analyse effect of PLC scan cycle time when signal was transmitted from device via DP/PA network, MPI interface card to PLC-CPU. Also carried out a hardware configuration of the modules in PROFIBUS DP and PROFIBUS PA and programming devices and looked at effect of data transition when more devices were connected to one PROFIBUS DP network.

Also compared the handling of functional blocks between PROFIBUS DP and conventional 4-20mA, convert numbers from decimal to binary, BCD, and hexadecimal etc. All results were captured by using especial PROFIBUS tool. Simulated, conducted tests using different methods and Analysed results based on the proposed solution and verify if performed as design. After the proposed solution was analysed, it was established that for the process plant to perform as designed, the core components of the proposed solution had to be listed as follows:

(i) 2 flow meters- are devices that were used detect the flow.

(ii) 2drives, 2 motors- were used to control speed of the motors

(iii) DP/PA coupler- it was used as the interface for PROFIBUS-PA field devices.

(iv) PLC, S7 300- Main control used for monitoring and controlling plant.

(v) HMI and PG- Display all variables of the process plant.

(vi) ET200s, ET200M (Module) - interface all PROFIBUS process automation devices.

(vii) Tanks and level transmitter- 3 tanks store/contains water and level transmitter to indication of level.

(viii) PROFIBUS cable, I/O's- used for transmitting all devices information and for fast data exchange.

3.3 Research objectives

Objective of this research was to:

Conduct an analysis of data transmitted using a conventional 4-20mA and PROFIBUS DP communication systems as they are both incorporated in the research. There was a loop based on PROFIBUS DP and another one based on conventional 4-20mA communication. Simulations were conducted and the simulated results were compared and analysed. Then a conclusion was drawn based on the obtained information, to establish which of the two methods was the best and give motivation why, based on facts.

Evaluate communication methods, including conversion of the PROFIBUS DP network transmission system from RS485 (bit coding with asynchronous code) to IEC 1158-2. Examine how data was transferred to the SIMATIC PCS 7 control system via PROFIBUS-DP at up to 187.5kb/s without significant loss of time (approximately 1ms) and the exchange of data with each field device lasts approximately 10ms. Discuss how the profile used the internationally recognized function block model to describe the device functions and parameters in addition to the application specific function blocks and two function blocks that were available for device-specific characteristics (physical block and transducer block). Also analysed how the input and output parameters of the function blocks were connected via the bus and linked to a process engineering application. `Further discuss the field device blocks that provided the interface to the hardware, including functional verification [13].

3.4 Assumptions and hypothesis

The necessary hardware and software tools required to conduct the research were available at Mangosuthu University of Technology and fully functional. The Programmable logic controller (PLC) communicates with all field instruments via PROFIBUS DP network and conventional 4-20mA. Parameterization was accomplished via PROFIBUS to facilitate communication between the PLC and field devices. It will be easy to modify or to upgrade due to latest technology like wireless devices or remote access with a better quality of signal. If there is a problem, the system generates an error message to help the user to quickly find the cause of the problem, and large data storage for system backups. The proposed system will have less equipment's failure because all devices are new and supplier spares will be available for the next 5 years. Researcher will gain acknowledged on latest technology and MUT will keep contemporary process plant for student demonstration.

3.5 Delimitation of the research

The main reason for research study was to:

(i) Establish a sound comparison between PROFIBUS and conventional 4-20mA based on obtains simulated results.

(ii) Conduct a comparison between simulated results and real-time results.

(iii) Examine how the devices data was transmitted via the PD/PA coupler.

(iv) Analyse PROFIBUS addressing and network configuration.

(v) Differentiate technical specifications between PROFIBUS DP and PROFIBUS DP/PA Coupler.

(vi) Evaluate field bus networks communication standards and configuration procedure.

CHAPTER FOUR: DESIGN OF THE PROPOSED SOLUTION

4.1 Introduction

This chapter reports the design concepts in this subsection for the proposed solutions and implementation plan aimed at addressing variations in each component to use. The purpose of the detail design for the proposed solution was to define the clarification in detail along with the approved project strategy and program. This work includes creating a functional specification, developing the solution design and preparing testing procedures and detailed individual components, such as the testing strategy, as well as schedules for all aspects of the process plant. Each subsection used different technology validation-complete document that tests the key features in the solution with a proof of concept. Discuss all process plant potential issues and their resolution, contains individual procedures for various roles. Discuss a conclusion as per the functional specification that outline requirements of all the components of the proposed solution and provides quantitative specifications about performance of the design.

4.2 Process plant flow diagram

Flowcharts are model for communicating a step-by-step process to others or diagram that represents a process or a procedure. The reason for this subsection was to visual representation a sequence of operation that was to be performed in order to get the solution to a problem. After a flowchart was drawn, it became comparatively easy to write the PLC program of the process plant and to clarify and to map out steps that lead to the concrete solution. The steps were represented by a series of boxes, and then connected with arrows. The author decided to draw a flowchart first and it assisted in analysis of the proposed solution and process plant to meet design specification [13].

Push button (start) was pressed, then both drive one and drive two started running and then pump 1 and pump2 begin to pump water from two tanks to mix tanks. PROFIBUS-DP flowmeter and PROFIBUS-PA flowmeter read the current flow. Level transmitter measures the level of tank 3 and all variables are displayed on the HMI. The procedure start again once the first cycle was completed. Figure 5 shows the flow chart diagram process of the process plant.

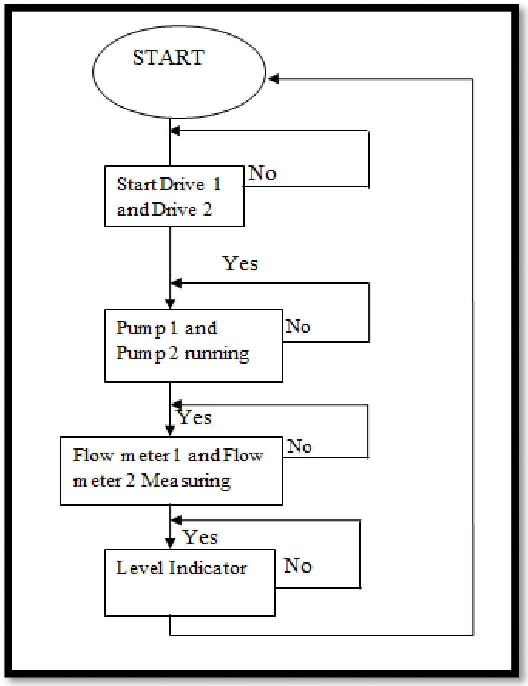


Figure 5: Flow chart of the process plant

4.3 Programmable Logic Controller

Programmable Logic Controllers (PLCs), also referred to as programmable controller, are in the computer family. They are used in commercial and industrial applications. A PLC was used in the research to monitor inputs, makes decisions based on its program, and controls outputs to automate a process plant. Type of the PLC used in this research was S7-300 (Siemens version 7- 300). Figure 6 shows the overview of the PLC S7-300 that was used as the main controller for the proposed solution. The function of the PLC for the proposed solution was to monitor signals from field devices and take the necessary action to keep the process within specified limits according to a predefined program by activating and controlling the necessary output devices so that process plant perform as designed.

Figure 6: Programmable Logic Controller (PLC) - S7 300

4.3.1 PLC basic operation

Main structure of the PLC S7-300 consists of input modules or points, a Central Processing Unit (CPU), and output modules or points and power supply as shown in figure 7. An input accepts a variety of digital or analog signals from various field devices (sensors) and converts them into a logic signal that can be used by the CPU. The CPU makes decisions and executes control instructions based on program instructions in memory. Output modules convert control instructions from the CPU into a digital or analogue signal that can be used to control various field devices.

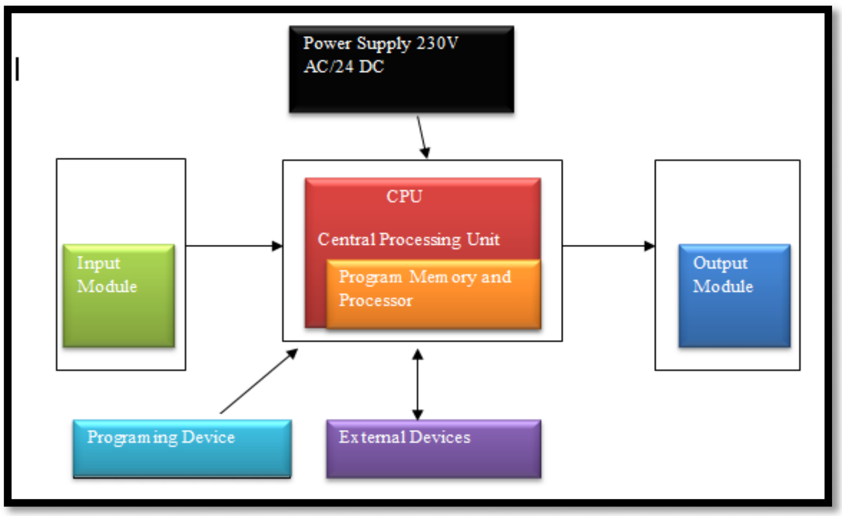


Figure 7: Main Structure of the PLC

A programming device was used to input the desired instructions. These instructions determine what the PLC will do for a specific input. An operator interface (HMI) device allowed process information to be displayed and new control parameters to be entered. The PLC program was executed as part of a repetitive process referred to as a scan. A PLC scan starts with the CPU reading the status of inputs. The application program was executed using the status of the inputs.

Once the program was completed, the CPU performs internal diagnostics and communication tasks. The scan cycle ends by updating the outputs, and then starts over. The cycle time depends on the size of the program, the number of I/Os, and the amount of communication required. Operating states of the PLC and their functions were used during configuration, downloading program and testing stage. This states had to be highlighted because were important and to ensure that the process plant perform as designed.

(i) Operate: Cyclic transmission of input and output data

(ii) Clear: Inputs are read, outputs remain in secure state

(iii) Stop: Diagnostics and parameterization, no user data transmission

Following are just a few of the advantages of PLCs:

(i) Smaller physical size than hard-wire solutions.

(ii) Easier and faster to make changes.

(iii) PLCs have integrated diagnostics and override functions.

(iv) Diagnostics are centrally available.

(v) Applications can be immediately documented.

(vi) Applications can be duplicated faster and less expensively.

Mode Selector For manual mode selection:

(i) STOP =Stop mode; the program is not executed.

(ii) TERM =Program execution, read/write access possible from PG.

(iii) RUN = Program execution, read-only access possible from PG.

(iv) Status Indicators SF =Group error; internal CPU error

(v) (LEDs) RUN =Run mode; green

(vi) STOP =Stop mode; yellow

(vii) DP=Distributed I/O (only CPU 313C)

The memory of the S7-300 was divided into three areas: program space, data space, and configurable parameter space. Program space stores the ladder logic (LAD) or statement list (STL) program instructions. This area of memory controls the way data space and I/O points are used. LAD or STL instructions are written using a programming device such as a PC, then loaded into program memory of the PLC.

Data space was used as a working area, and includes memory locations for calculations, temporary storage of intermediate results and constants. Data space includes memory locations for devices such as timers, counters, high-speed counters, and analogue inputs and outputs. Data space was accessed under program control. Configurable parameter space, or memory, stores either the default or modified configuration parameters [14].

There are four basic steps in the operation of all PLCs as shown in figure 8; Input Scan, Program Scan, Output Scan, and Housekeeping. These steps continually take place in a repeating loop. Input Scan (detects the state of all input devices that are connected to the PLC). Program Scan (executes the user created program logic). Output Scan (Energizes or de-energize all output devices that are connected to the PLC). Housekeeping (his step includes communications with programming terminals, internal diagnostics, etc...)

4.3.2 Central processing unit

The central processor unit (CPU) 313C-2 is a microprocessor system that contains the system memory called a RAM and was the PLC decision making unit. The CPU function in this research was monitor the inputs and makes decisions based on instructions held in the program memory. The CPU performs relay, counting, timing, data comparison, and sequential operations. In this research PLC S7-300 CPU-313C-2 required RAM with a capacity of: 64KByte, 256Kbyte.

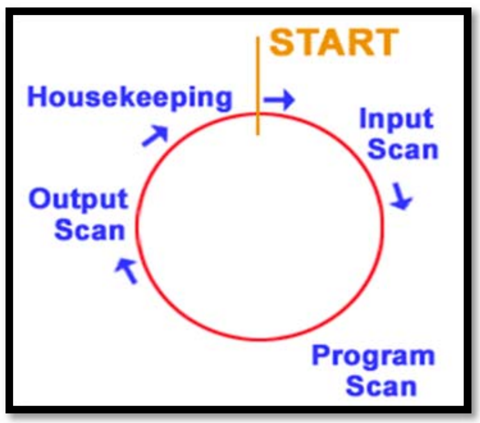


Figure 8: PLC continually processed in a loop

4.3.3 Program of the PLC

A program generally consists of an interconnection of function blocks, each of which may be written in any of the IEC languages. A function block or program is also called a program organization unit. In addition to the function blocks, the program contains declarations of physical inputs/outputs and any variables local to the program. A program read and writes to I/O channels, global variables, and communicates with other programs. Access paths provide the means to transfer information between configurations. Tasks control the execution of the program or parts of a program. The execution of a program implies that all of the function blocks in the program are processed once.

The execution of a function block implies that all of the software elements of the function block are processed once. There are no implied mechanisms for program execution. In order for a program to be executed, it must be assigned to a task, and the task was configured for process plant to perform as design. A software program was required in order to tell the PLC what instructions it must follow. Programming software was typically PLC specific and the program was stored in (non-volatile) memory, meaning that the program will not be lost if power was removed. The S7-300 used a Windows based software program called STEP 7-Micro/WINCC. The HMI (human machine interface) has STEP 7 software pre-installed. Micro/WIN32 was installed on a personal computer in a similar manner to any other computer software.

CHAPTER FIVE: PROFIBUS TECHNOLOGIES

5.1 Introduction

This section describes the important aspects of PROFIBUS and takes into account the level of technology and provides a complete functional description of the PROFIBUS system for the process plant. PROFIBUS is the world's leading fieldbus system and approximately 40 million devices installed, roughly 3,000 products from over 300 different suppliers. PROFIBUS has been around for almost 25 years and is a smart, field-bus technology or communications standard for linking process control and plant automation components. Instead of running individual cables from a main controller to each measuring device and actuator, but PROFIBUS communicating with all device in the network and a controller using single pair cable. The purpose of PROFIBUS in this research was to connect all devices, with high speed, bi-directional, serial messaging used for transfers of information [14].

Devices on the system connect to a central line. Once connected, these devices were communicate information in an efficient manner, but went beyond automation messages. PROFIBUS was also used for connecting/linking all PROFIBUS DP and PROFIBUS PA devices and main controller (PLC). Only the node with the token can communicate. Figure 9 below shows the PROFIBUS network structure, unique address for each device connected and main controller called programmable logic controller.

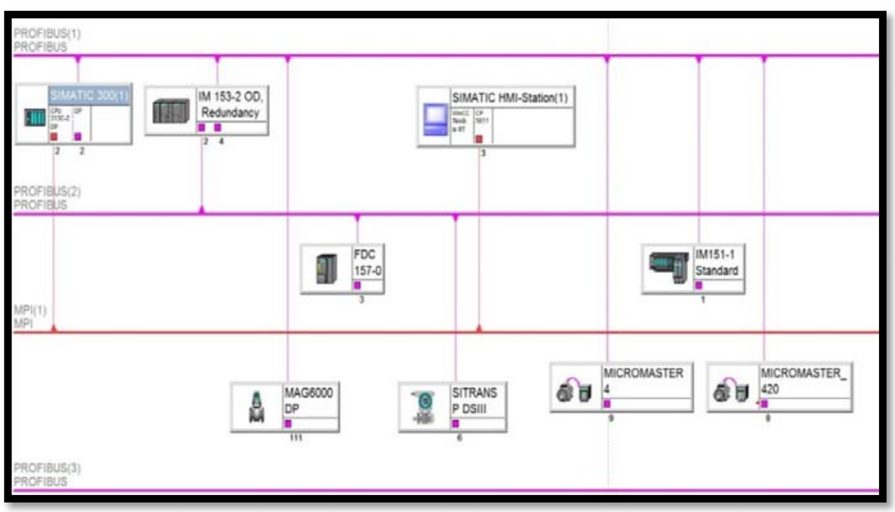


Figure 9: PROFIBUS network structure

PROFIBUS devices can also participate in self-diagnosis and connection diagnosis. At the most basic level, PROFIBUS benefits from superior design of its OSI layers and basic topology. PROFIBUS also includes token passing, a system in which a "token" signal was passed between nodes. There are multiple versions of PROFIBUS that handle different types of messaging at the application layer. Some of the types of messaging PROFIBUS supports include cyclic and acyclic data exchange, diagnosis, alarm-handling, and isochronous messaging. PROFIBUS does not define layers three through six. It does, however, define the data link and physical layers, layers one and two.

The data link layer was completed through a Field bus Data Link, or FDL. The FDL system combines two common schemes, master-slave methodology and token passing. In a master-slave (PLC) network, the master was the programmable logic controller that sends requests to slaves, sensors which in this study sensor were two flow meters and actuators were referred to as drivers. The slaves respond accordingly. PROFIBUS defines a physical layer, though it leaves room for flexibility. PROFIBUS systems can have three types of media. But On this research the standard twisted-pair wiring system was used and called RS485.

In this model every transmission layer handles precisely defined tasks. Layer 1 (physical Layer) defines the physical transmission characteristics. Layer 2 (data link layer) defines the bus access protocol. Layer 7 (application layer) defines the application functions. PROFIBUS has advanced through a handful of revisions. In this case, RS 485 transmission was the transmission technology most used by PROFIBUS. The application area includes all areas in which high transmission speed and inexpensive installations are required. Twisted pair shielded copper cable with one conductor pair was used [15].

The bus structure permits addition and removal of stations or step-by step commissioning of the system without influencing the other stations. Later expansions had no effect on stations which were already in operation. Transmission speeds between PROFIBUS-DP and PROFIBUS-PA 187 Kbits/sec are selected. 187.5 Kbits/sec transmission a speed that was selected for all devices on the bus when the system was commissioned due to the speed rate was suitable for bus. The bus was terminated by an active bus terminator at the beginning and end of each segment to ensure error-free operation, both bus terminators were powered. The bus terminator was switched in the devices and cable length specifications were based on type-A cable as shown in figure 10. In degree of protection IP20, the use of a 9-pin D sub connector was preferable for PROFIBUS networks using RS 485 transmission technology.

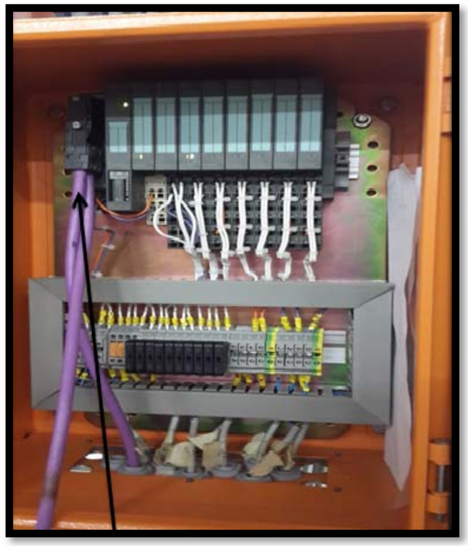


Figure 10: PROFIBUS cable

5.2 Decentralised Periphery (DP)

PROFIBUS-DP (decentralised periphery) is an extension of the original PROFIBUS standard, see Fig. 11. An extension contains a subset of the functionality of the original standard and is targeted at a specific area of application. PROFIBUS communication Profiles define how users transmit their data serially via the common transmission medium. DP is the most frequently used communication profile. It is optimized for speed, efficiency and low connection costs and was designed especially for communication between automation systems and distributed peripherals.

The PROFIBUS-DP protocol was used for fast exchange of data at the sensor/actuator level. At this level central control unit was a PLC, communicate with distributed input and output devices via a high-speed serial connection. Data exchange with these distributed devices was mainly cyclic; see Fig 12 and Fig 13. The central controller (PLC) reads the input data from the slaves and writes the output information to the slaves. The bus cycle time was shorter than the PLC scan time. DP was suitable as a replacement for conventional, parallel signal transmission with 24 volts in manufacturing automation as well as for analogue signal transmission with 4 ... 20 mA.

In the original version, PROFIBUS-DP allowed only one master that communicated via the master-slave method. PROFIBUS-DP was premeditated to provide high data transmission rates (up to 187.5 kbit/s) for this process plant and short response times (up to 1 ms) and was particularly suitable for direct control. One (MAG 5000) DP flow meter, two drives (M440 and M420), DP/PA coupler and intelligent field devices on the PROFIBUS-PA were connected via a DP/PA coupler to the Remote I/O device in this case was ET200M.

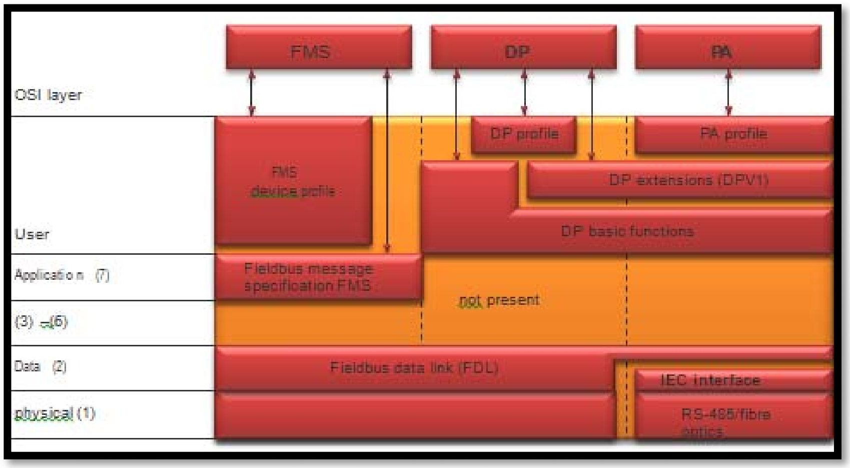


Figure 11: PROFIBUS versions and functions

PROFIBUS-DP was used as master/slave bus system were the master function was assumed by a PLC (master class 1) or HMI was (master class 2). Master class 1, in which the automation functions (closed-loop) has full access to the field devices via cyclic and acyclic messages. Master class 2 was required, for exchange data via acyclic messages with master class 1 (upload/download, master diagnostic read) and exchange data with the field devices (measured value read, slave diagnostic read, parameter write).

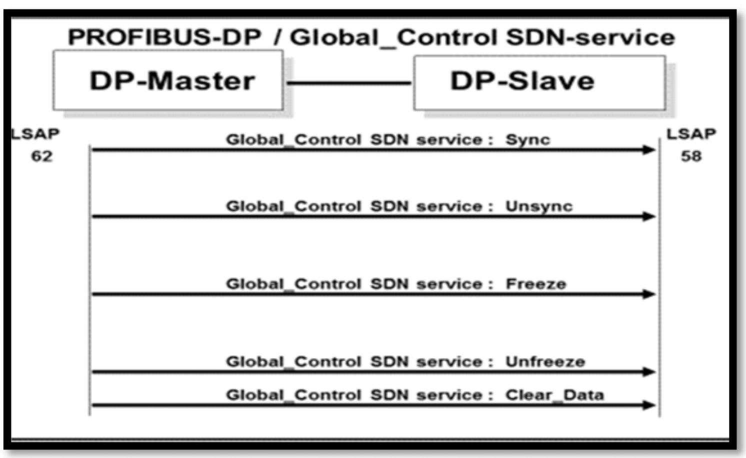


Figure 12: PROFIBUS DP and DP Slave

Current field devices such as Drives, DP flow meter had in addition to the measured value or manipulated variable, many parameters were changed during start up and, to some extent also during operation in order to utilize the "intelligence" of these field devices such as preventive maintenance or optimization of the interface to the sensors. On account of the different time-related demands for data access of the master (PLC), PROFIBUS-DP offers cyclic and acyclic services.

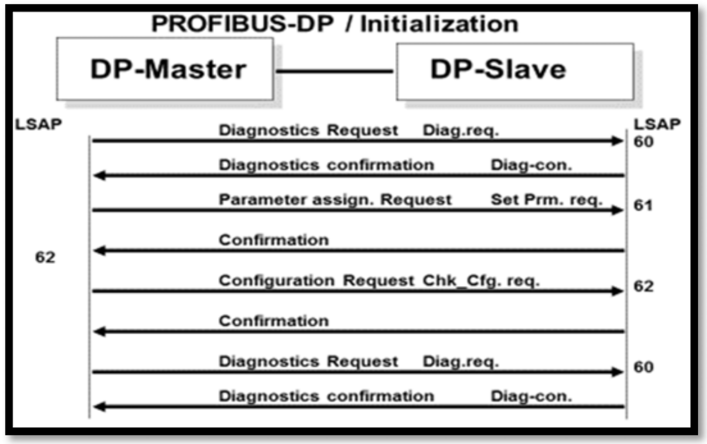


Figure 13: PROFIBUS DP initialization

All output values (control commands) were written to the field devices and all input values (measured values) were read out of the field devices in one cycle. Subsequently, an acyclic data interchange can take place with a particular field device. Settings of the field devices can be read or parameters can be modified. With the facility for supplementing each transmission cycle with precisely one single acyclic message, short, deterministic cycle times are ensured as the basis for software control in the PLC.

5.3 PROFIBUS Process Automation (PROFIBUS-PA)

PROFIBUS-PA (process automation) is an extension of PROFIBUS-DP for process automation. It has two specialities: firstly, participants can draw safe power from the bus; secondly, the data transfer is handled according to the international standard IEC 61158-2. A maximum of 32 participants can be connected to a PROFIBUS-PA segment but in this research only one participant was used and called PROFIBUS-DP flowmeter (MAG 6000). PROFIBUS-PA was used at field level and the segment coupler serves both as interface to the PROFIBUS-DP system and as power supply for the PROFIBUS-PA field devices.

Function of PROFIBUS-PA in this proposed solution was to permits simultaneous transmission of digital data and powering of the bus by means of a two-wire cable for the hardwired flow meter and it was labelled as node 4, See Fig 14. The suitable transmission rate on a PROFIBUS-PA segment was set at 187.5 Kbit/s so that it matches the PROFIBUS-DP transmission rate since two signals were going to be compared. The typical communication time of a transmitter was only approx.10ms. This means that practically all typical applications in the process plant were implemented with minimal constraints on segment device density [16].

PROFIBUS-PA was also used as linear structure and the DP/PA couplers/link was used for the supply of power and for adaptation of the data transfer rate from PROFIBUS-DP to PROFIBUS-PA. Conversion of the PROFIBUS-DP transmission system from RS 485 (bit coding with asynchronous code) to IEC 1158-2 (bit coding with synchronous code). PROFIBUS-PA takes place via the DP/P coupler and exchange the data with each field device lasts approximately 10 ms (outgoing and return message), thus the cycle time on a DP line with 3 field devices was about 6 x 0.1 ms = 0.6 ms, i.e. The measured values were read into the CPU or the manipulated variables can be read out 10 times per second.

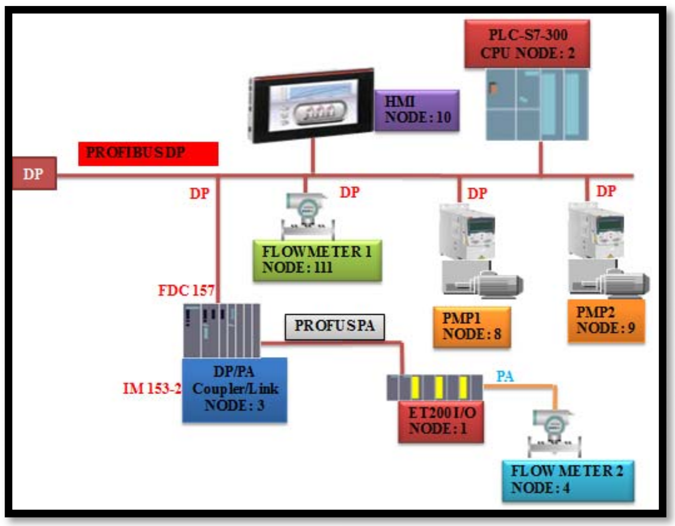


Figure 14: PROFIBUS DP/PA network

5.4 DP/PA Coupler/link

The DP/PA link functions as a slave on the PROFIBUS-DP and as a master on the PROFIBUS-PA. From the viewpoint of the host controller, the DP/PA link is a modular slave whose modules are the devices connected on the PROFIBUS PA. Addressing of these devices was carried out indirectly via the DP/PA link which itself only requires one DP address. Segment coupler/link converts a signal that adapts the RS-485 signals to the IEC 1158-2 signal level. From the point of view of the bus protocol coupler/link was transparent.

The baud rate used in this process plant was 187.5Kbit/s for DP because was suitable for segment couplers and there was no limit to the baud rate in the RS-485 segment when using coupler/links. DP/PA coupler/link was shown in Fig 15. PROFIBUS-PA was used in conjunction with a supervisory PROFIBUS-DP control system, since the protocols, physical layer and transmission rates of

PROFIBUS-DP and PROFIBUS-PA were different. The PROFIBUS-PA segment (fixed at 31.5Kbit/s) was connected to the PROFIBUS-DP system via a segment coupler/link (IM153-2. The data for hardwired flowmeter was transferred through the PD/PA link to the Main Controller (PLC).

Figure 15: PROFIBUS DP/PA Coupler Module

The comparison of the two signals was considered based on the communication speed rate and number of PA devices per PROFIBUS master (PLC). The host PROFIBUS master scanned hardwired flowmeter connected to the DP/PA link at once. The speed rate on the PROFIBUS-DP was not influenced by the subordinate PROFIBUS-PA, and DP devices therefore were operated together with the PA devices on the same segment without a loss in performance.

DP/PA coupler was linked with ET200M via FIELDBUS connectors. FIELDBUS normally have marked cable entries for (incoming) and (outgoing) cables. This distinction was important in (isolating connectors) where the outgoing cable was isolated when the termination was switched “in”. Such isolating connectors were useful for commissioning and testing the bus network. It was also useful for maintenance; when allowed sections of a segment to be isolated whilst retaining correct termination. Figure 16 and 17; shows the cable connection for first and last on the segment and connections for all other stations on the segment.

When any such connector has the termination switched (in) all the stations/devices on the outgoing side are disconnected, leaving a properly terminated sub-segment on the ingoing cable. For this reason the first and last connector were only used on a segment for the ingoing entry, as shown in figure 17. Field bus structures with PROFIBUS-PA have considerably lower fault potential than conventional cabling. If, however, a fault occurs, it can be very quickly located and corrected on account of the simple structure

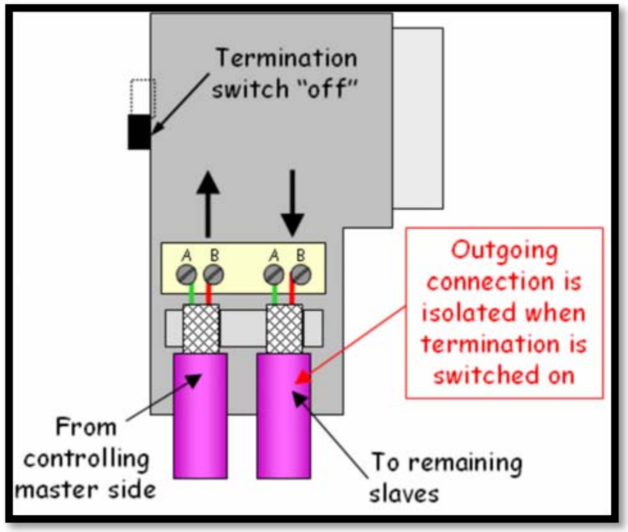


Figure 16: Connections for all other stations on the segment [17]

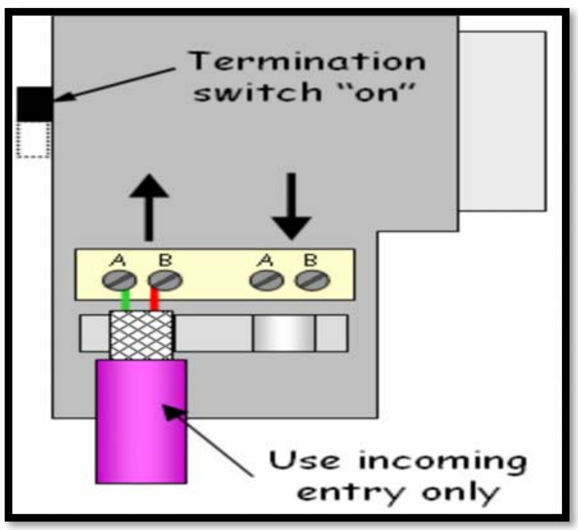


Figure 17: Connection for first and last stations on the segment [17]

CHAPTER SIX: FLOW MEASUREMENT INSTRUMENTS AND VARAIABLE SPEED DRIVES

6.1 Introduction

Flow measurement is important in many industries such as the oil, power, chemical, food, water, and waste treatment industries. These industries require the determination of the quantity of a fluid, either gas, liquid, or steam, that passes through a check point, either a closed pipe or an open channel, in their daily processing or operating.. Many different types of sensors can be used for flow measurements. The choice of any particular device for a specific application depends on a number of factors such as-reliability, cost, accuracy, pressure range, temperature, wear and erosion, energy loss, ease of replacement, particulates and viscosity. Function of a flowmeters in this research was to measure the flow rate/quantity of liquid in a pipe. The quantity to be determined may be volume flow rate, mass flow rate, flow velocity but in this study flow rate was selected [18].

The main reason for this chapter was to discuss the measurement of the flow of water in process plant, standards, formulas, and laws associated with flow rates. Instrument used in the measurement of flow rates are described, as well as considerations in instrument selection for flow measurements are discussed. Outline the required flow meter range by identifying minimum and maximum flows (mass or volumetric) that was measured. After that, the required flow measurement accuracy was determined. Specify the accuracy in percentage of actual reading, in percentage of calibrated span and in percentage of full scale units.

Electromagnetic flowmeter was used only for measuring flow liquids and it consists of two electrodes mounted in the liquid on opposite sides of the pipe. Magnetic fields were generated across the pipe perpendicular to the electrodes. The conducting fluids flowed through the magnetic fields and generated a voltage between the electrodes, which were measured to give the rate of flow. The meter gave an accurate linear output voltage with flow rate. There was no insertion loss and the readings are independent of the fluid characteristics. Clarify the calibration procedure of flowmeter by using a known flow rate so that the appropriate correct readings are obtained. A spot check of the flowmeter readings was periodically to check for instrument drift that may be caused by the instrument going out of calibration, particulate build up, or erosion.

6.2 DP flowmeter

DP flow meter is a measuring instrument that was used for signals comparison on this research. The flowmeter was connected on the PROFIBUS DP network and communicates direct with the Master controller (PLC). DP flowmeter communicate direct with the programmable logic control and it was also assigned unique address. Mag 5000 flowmeter address was transmitting a data of 2 bytes for inputs/outputs with a speed rate of 187.5kbit/s baud rate. Figure 18 shows the DP flowmeter and the details of the flowmeter as follows: A SITRANS F M MAG 5000 flow meter system includes:

(i) Transmitter type: SITRANS F M MAG 5000.

(ii) Sensor type: SITRANS F MAG 5100W.

(iii) Communication module type: PROFIBUS PA/DP.

(iv) SENSORPROM: memory unit.

(v) The transmitter installed: compact on the sensor.

(vi) Types of power supply: 115…... 230 V AC switch mode type [19].

The transmitter has a microprocessor-based with a built-in alphanumeric display in several languages. The flow measuring principle is based on Faraday's law of electromagnetic induction. Magnet coils mounted diametrically on the measuring pipe generate a pulsed electromagnetic field. The liquid flowing through this electromagnetic field induces a voltage. The transmitters evaluate the signals from the associated electromagnetic sensors, convert the signals into appropriate standard signals such as 4 ... 20 mA, and also fulfil the task of a power supply unit providing the magnet coils with a constant current.

Figure 18: DP Flowmeter (MAG 5000)

6.3 MAG 5000 cable and signal connection

DP flow meter consist of a memory called sensor PROM memory unit which stores sensor calibration data as well as transmitter settings for the lifetime of the product. The factory settings matching the sensor are stored in the sensor PROM unit. Specified settings were downloaded to the sensor PROM unit. If the transmitter be replaced, the new transmitter will upload all previous settings and resume measurement without any need for re-programming. Figure 19 shows the sensor connections terminals are 91 and 97 and the transmitter terminals are 56 and 58 for digital outputs and the supply voltage is 115—230V AC.

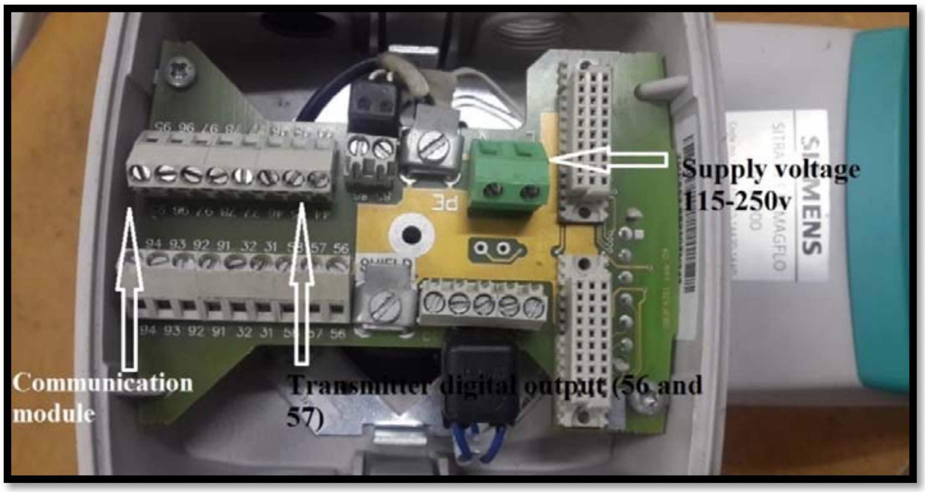


Figure 19: MAG 5000 connection

6.4 ET200 I/O module and flow meter (hard-wired)

The ET 200 distributed I/O system with the PROFIBUS-DP protocol was used for connecting distributed I/O's to a PLC-Central Processing Unit. ET 200 I/O was characterized by fast reaction times because only a few data (bytes) are transferred for MAG 6000 (hardwired) and was based on standard EN 50170. ET 200 I/O operates according to the master-slave principle. The ET200M consists of an IM151-1 interface module that was connected to an ET200S/IM151-1-PROFIBUS master, see Fig 20. PLC S7-CPU 313-2C modules addressed via the PROFIBUS bus inserted in the ET 200 I/O. Mag 200 I/O was also called conventional flowmeter because the sensor generates analog signal and converted to digital signal and then transmitted to the DP/PA coupler.

Mag 6000 flowmeter address space ET 200M was 5 bytes for inputs/outputs with a speed rate of 187.5kbit/s baud rate. The connection system was characterized by the following points: The 24 V DC supply voltage was connected in the front connector by means of screw terminals. The MAG 6000 transmitter (hardwired) measures the flow in the process pipe line and process 4 to 20 mA signal.

ET200M was used to interface Mag flow 6000 measurement to the PLC. The ET200I/O (IM151-1) module connecting a rated supply voltage to the BaseUnit higher than the one given in the technical specifications that may lead to dangerous situations in the process plant or cause defects in ET 200I/O components. Therefore, only connect the rated supply voltage given in the technical specifications to the BaseUnit. The MAG 6000 power was supplied and connected to the 2AI -2 wire input module. The 2AI analog module was configured in the ET200 I/O and BusUnit shown on Fig 21.

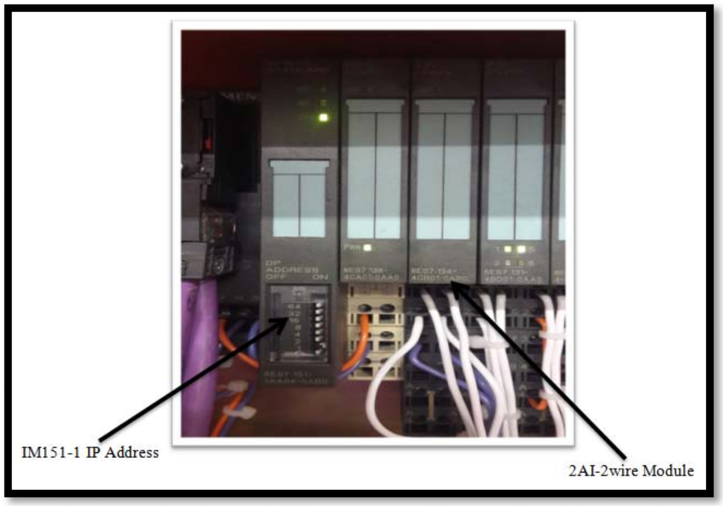


Figure 20: ET200 Module for PA (2AI-2wire)

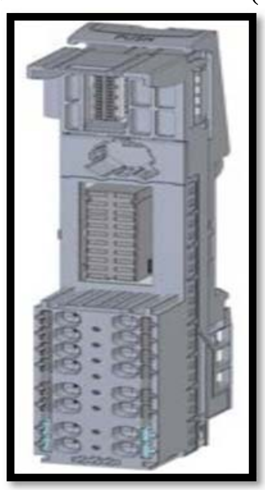


Fig 21: BusUnit for 2AI-2 wire (6ES7 134-4GB01-OAB0)

BaseUnit was suitable for all I/O modules of the BaseUnit type (B0/B1) and was recognized by the last two digits of the order number (6ES7 134-4GB01-DAB0) and the current-carrying capacity per process terminal (terminal 1 to 12): Max. 5A. The Base unit conducts the potential group further and the power busbars P1, P2 and the AUX busbar are connected to the IM151-2 module (Base unit) on the left. The access to the AUX busbar via terminals 12 terminals to the process (assignment with the

I/O module) connection technology in the form of push-in terminals [20]. The connection of the 2AI-2wire Block diagram is shown on Fig 22 and detail of each section:

(i) Backplane bus,

(ii) I/O module and

(iii) Terminals with connection to the I/O module

(iv) Connected power bus bars without connection to the terminals

(v) Connected AUX bus bar connected to the terminals

Current-carrying capacity per process terminal was (terminal 1 to 12). The power busbars P1, P2 and the AUX busbar are disconnected from the adjacent module on the left (BaseUnit, interface module).

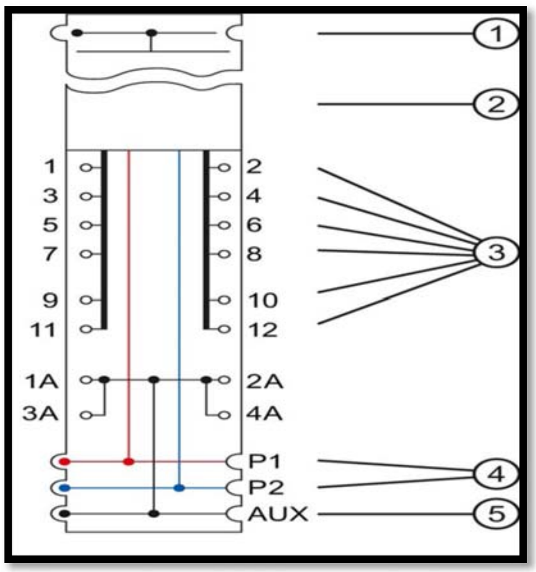


Fig 22: Block diagram (6ES7 134-4GB01-DAB0)

This type of magnetic flowmeter used in this research was called MAG 6000. This flowmeter had a microprocessor-based flow transmitter engineered for high performance. The main task of the flowmeter transducer was to generate sinusoidal excitation voltage for the coils and measure the level of the voltage coming from the electrodes. Theoretically the electrodes derive sinusoidal voltage proportional to the flowrate, and the voltage level measurement was simply the determination of the amplitude of the received sinusoidal waveform.

Circuit board generated voltage was gained and transformed into current, so the magnetic induction was only nearly constant and the waveform was distorted. Digital signal processor converts the analog flow signal to a digital signal and suppresses electrode noise through a digital filter. The analog to digital conversion takes place in an ultra-low noise with 23 bit signal resolution. The transmitter operates internal via an internal control area network communication bus. Signals were transferred to/from a signal conditioner to the display module, internal/external option modules and the module.

6.5 Variable speed drive

The word drive can have many meanings. In this chapter, the definition of drive is limited to any intelligent electronic equipment that provides adjustable speed control for a motor or the variable-frequency drive is the electronic controller specifically designed to change the frequency of voltage supplied to the motor. Drives provide many functions for the motor. Most often the drives have some form of rectification that takes place in addition to providing intelligent control for the motor. There are two types of drives which are AC (alternating current) drive and DC (direct current) drive. Variable-frequency drives (VFD) for AC motors have been the innovation that has brought the use of AC drive back into prominence. The AC-drives can change the speed by changing the frequency of the voltage used to power it.

This means that if the voltage applied to an AC motor is 60 Hz, the motor will run at its rated speed. If the frequency is increased above 60 Hz, the motor will run faster than its rated speed, and if the frequency of the supply voltage is less than 60 Hz, the motor will run slower than its rated speed. In the 1960s, the frequency drives had rather small solid-state components that limited the amount of current the drive could supply to the motor. This usually limited the size of the motor that could be controlled by a frequency and they were not commonly used. When larger transistors became available in the 1980s, variable-frequency drives allowed the largest motors to have their speed controlled. The earliest drives used linear amplifiers to control all aspects of the drive. Variable DC drives have been used to control DC motors longer than variable frequency drives have been used to control AC motors.

The first motor-speed control used DC motors because of the simplicity of controlling the voltage to the armature and field of a DC motor. The main obstacle in using DC motors is the increased level of maintenance involved because the DC motor has brushes and a commutator. Early speed control for DC motors consisted of large resistors that were switched in the motor circuit to reduce the amount of voltage supplied to it. The resistors created problems because of the heat build-up. In this research Siemens AC variable speed drive were used and the type of drive were called MICROMASTER 420 and MICROMASTER 440 and both drives were used for three phase application only. This section discusses the required speed rate, configuration and analysis of the detail design of the proposed solution.

6.6 DRIVE MM440 and MM420

There are two types of drives used on this research which are MICROMASTER 440 and 420. Both drives are suitable for a variety of variable speed applications, such as pumps and as shown in figure 23. In this case both drives were controlling pumps to their desired set point according to system application. Both pumps were used to pump water or dissimilar products from different tanks to a mix tank. The MICROMASTER 440 and 420 were controlled by a programmable digital microprocessor and were characterized by ease of setup and use. The MICROMASTER 440 and 420 are available in three fame sizes with power ratings from 1/6 HP to 15 HP**.** On this research, Drive 440 and 420 used frame size (200-240 VAC, 1/3 phi). Drives were primarily used for speed control of motors, by changing the frequency the drive ensures that the motor has the full supply voltage on it at all times and thus keeps the motor at full torque for a variety of different speed, F*60 = rpm.

Figure 23: MM440 and MM420 drives

The MICROMASTER has an RS485 serial interface that allowed communication with programmable logic controllers (PLC) through PROFIBUS communication method. The protocol was PROFIBUS and was programmable for 187.5 kb/s baud rate. The AC voltage comes into the drive; it gets "chopped" into a control frequency which was then fed into the control loop. The drive scales the external reference so that the signal minimum and maximum values correspond to a speed other than the minimum and maximum speed limits [20].

Control wiring, such as speed reference and start/stop control was connected to the control terminals as shown in figure 24. Power wiring, such as the AC supply and the motor, was connected to the power terminals. The drive form a reference out of an analogue input signal and a signal received through a PROFIBUS communication interface by using mathematical functions: addition and multiplication.

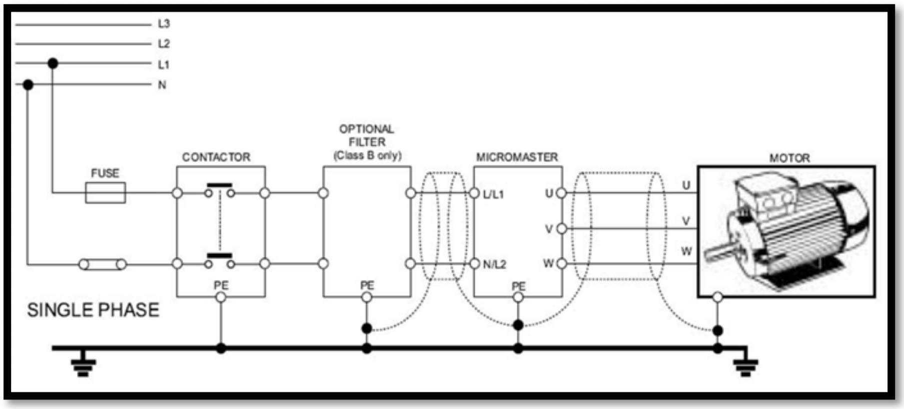


Figure 24: Micromaster Drive wiring [21]

The Drive MM 440 and MM420 had an operator panel and a PROFIBUS module for Parameters, such as ramp times, minimum and maximum frequencies and modes of operation were easily set. Toggle the display between a parameter number and the value of the parameter. The up and down pushbuttons scroll through parameters and were used to set a parameter value. In the event of a failure the inverter switches off and a fault code appears in the display. MICROMASTER 440 and MICROMASTER 420 have the following features:

(i) Flux Current Control (FCC)
(ii) Linear V/Hz Control
(iii) Quadratic V/Hz Control
(iv) Flying Restart
(v) Slip Compensation
(vi) Automatic Restart
(vii) PI Feedback for Process Control
(viii) Programmable Acceleration/Deceleration
(ix) Ramp Smoothing
(x) Fast Current Limit (FCL)
(xi) Compound Braking

Drive M440 and M420 has a PI-controllers are commonly used in drive technology. In this research, the desired speed and actual speed were input to a summation point. The two signals were opposite in polarity. When the actual speed was equal to the desired speed the deviation, which was the input into the PI-controller, was zero (0). Whenever desired speed and actual speed were different there was a deviation. Changes in load on the motor, for example, can affect motor speed. A sudden increase in load would cause the motor to slow down. This would decrease the feedback from actual speed and the deviation would become more positive. It was also possible that the application may require the motor to slow down or speed up. Until the motor reached the new desired speed there was a deviation. Drives PI controller parameters are shown below.

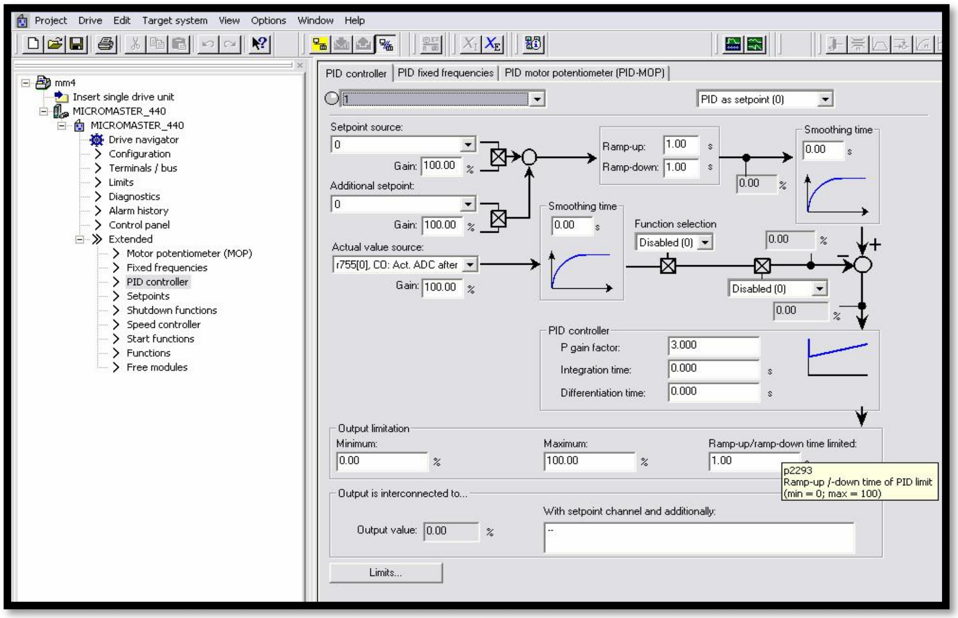


Figure 25: Drive PI controller

The PI-controller makes speed corrections quickly a minimal amount of overshoot and oscillation. Parameter P235 (gain) and parameter P240 (time) were used to tune the PI controller performance. The integral state was separately controlled, corresponding to whether the PI controller output was saturated or not. The experimental results showed that the speed response has much improved performance, such as small overshoot and fast settling time, over the conventional technique. Although the operating speed command was changed, similar control performance was obtained by using the PI gains selected in the linear state. The amplified error signal and the integrated error signal are then recombined at the output via a simple addition operation.

The integrator was included to drive the steady-state error of the system to zero, since any non-zero steady-state error was theoretically result in an unlimited integrator output. In order to prove that both drives were running at the same speed and settings were identical, the validity of the auto-tuning method was performed and results achieved as shown in figure 25. At the end of the result fast response time with about a 43% initial overshoot. The motor was settle in to the new desired speed. Connectors and binectors were elements used to exchange signals between function blocks. Connectors were used to store analog values. Analog values were stored in the form that was represented by 16 bit or 32 bit words.

Main characteristics of the Drive

(i) Easy, guided start-up
(ii) Modular construction allows maximum configuration flexibility
(iii) Six programmable isolated digital inputs
(iv) Two scalable analog inputs (0 V to 10 V, 0 mA to 20 mA) can also be used as digital input
(v) Two programmable analog outputs (0 mA to 20 mA)
(vi) Three programmable relay outputs (30 V DC/5 A resistive load; 250 V AC/2A inductive load)
(vii) Low-noise motor operation and high pulse frequencies.
(viii) Complete protection for motor and inverter.

Parameter P0700 was used to select control panel and the terminals one (1) through four (4) were used to provide a reference that controls the speed of the motor from 0 to 100%. Terminal one (1) was a +10 VDC power supply that was internal to the drive. Terminal two (2) was the return path, or ground, for the 10 Volt supply. An adjustable resistor was connected between terminals one and two. Terminal three (3) was the positive (+) analog input to the drive and the programmed to accept 4 to 20 mA speed reference signal.

These signals were typically supplied to the drive by the programmable logic controller (PLC). A jumper was connected between terminals two (2) and four (4). An analog output was used to monitor output frequency, and motor RPM. The control mode was linear voltage/frequency and the MM drives were programmed and operated by the SIMATIC HMI device and parameters, such as ramp times, minimum and maximum frequencies, and modes of operation were easily set. The changeover key (P) toggles the display between a parameter number and the value of the parameter. The up and down pushbuttons scroll through parameters and were used to select a parameter value, once the (P) key sets the parameter. The HMI used a touch-sensitive screen for control and monitoring.

CHAPTER SEVEN: THE HMI- PANEL CONTROL

7.1 Introduction

Human Machine Interfaces (HMIs) broadly refer to any graphical device that allows a user to interact with a machine's real-time control system and have been prevalent in manufacturing applications for decades. Modern HMIs typically include a touchscreen interface, a display with multiple pages, electronic storage for data logging, and a method of displaying alarms. Recently, HMIs have been introduced in automotive applications as a means of transmitting large amounts of disparate information to the driver. An HMI is used for three primary roles: a pushbutton replacer, data handler, and overseer. The pushbutton replacer takes the place of LEDs, on/ off buttons, switches or any mechanical device that performs a control function. The elimination of these mechanical devices is possible because the HMI can provide a visual representation of all these devices on its LCD screen, while performing all the same functions. The Data Handler is used for applications that require constant feedback and monitoring. Often these Data Handlers come equipped with large capacity memories.

The HMI is used throughout various industries including manufacturing plants, vending machines, food and beverage, pharmaceuticals, and utilities, just to name a few. HMIs along with PLCs are typically the backbone of the production line in these industries. HMI products originated from the need to make machinery easier to operate, while producing optimal outputs. Predecessors of HMI include the Batch Interface (1945-1968), Command-Line User Interface (1969-Present), and the Graphical User Interface (1981-Present) [21].

7.2 The HMI- panel control

An HMI is the interface method between the human and the machine. A human–machine interface involves peripheral hardware for the inputs and for the output and there is an additional component implemented in software, like e.g. a graphical user interface. An HMI provides a visual representation of a control system and provides real time data acquisition. In this research manufacture was Siemens and the type of HMI was used Panel PC 677 V2. The device consists of the control unit with a touch panel and the computer unit as shown in figure 26. Colour display with backlighting technology with 800 x 600 pixels resolution provided adequate cable diameter when setting up PROFIBUS DP networks and MPI networks.

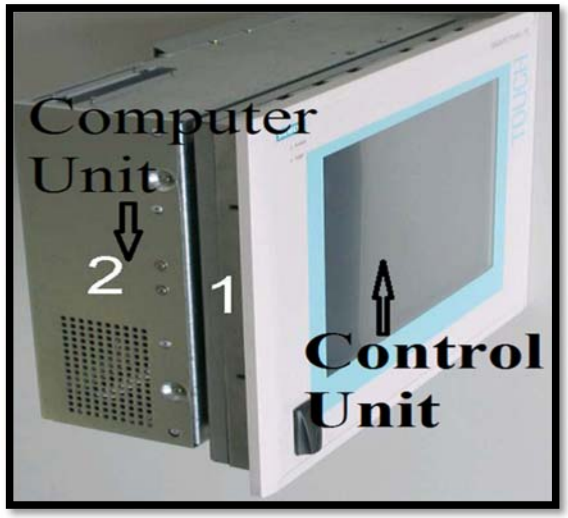


Figure 26: Control unit and computer unit

The user interacts with the computer over this software interface using the given input and output (I/O) hardware. Software and hardware was matched, so that the processing of the user input was fast enough, the invisibility of the computer output was not disturbing to the workflow as shown in figure 27.

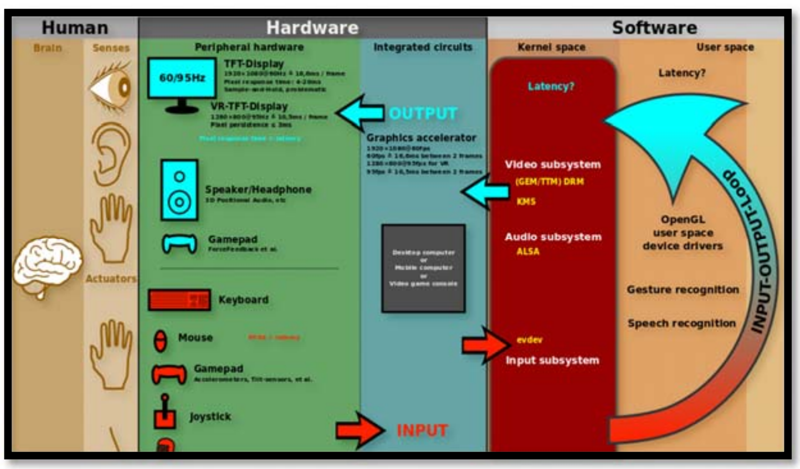


Figure 27: HMI user interaction [11]

PC-based control unit was for monitoring system for visualizing and operating processes, all inputs and outputs values and all parts of the plant. The HMI computer unit has a processor unit on-board with industry-standard functions for signalling and acknowledging events, archiving of messages and measured values, logging of all process and configuration data, user administration and visualization as shown in figure 27. Configuring was easy and efficient using object libraries, modular systems, tools for mass data processing and online loading of changes. Graphical interface contains colour coding that allows for easy identification and pictures allow for fast recognition. Generally, the goal of user interface design was to produce a user interface which makes it easy (self-explanatory), efficient, and enjoyable (user friendly) to operate a machine in the way which produces the desired result.

This generally means that the user needs to provide minimal input to achieve the desired output, and also that the machine minimizes undesired outputs to the human. The suitability that comes with an HMI was extremely valuable; the functionality achieved with digitizing a system with an HMI was unbeatable. HMI combines all the control features that were typically created throughout the automation line and places them in one centralized location, eliminating the need to run to a red pushbutton that will stop the process plant in an emergency. With remote access, the user does not need to be anywhere near the process plant to start/stop or monitor production. The user has all the same features, on centralized unit in a smaller compact form. With an HMI the user have a centralized unit of control, which allows for decisions based on real time events in a visual manner. The HMI was in constant communication with a PLC so the HMI can receive real time data from PLC. The HMI accesses specific registers on the PLC and makes decisions based on the state of those registers.

The PLC was used in this research to monitors inputs, makes decisions based on its program, and controls outputs to automate a process plant, to achieve good results and the HMI was required to display all inputs, output values and entire process plants so that inputs signals are monitored. The main objective of using HMI on this process plant was to allow effective operation and control of the plant from the human end, whilst the plant simultaneously feeds back information that aids the author to decision making process. HMI/PLC Block diagram overview is shown on figure 28. It offered the possibility to do interactive tests, giving access to all simulated operating conditions. This module allows quick implementation and linking it to the system during simulation. Parameters and animation rules were attached to each graphical object and component in order to produce the required visual effects as a result of the simulation.

The combined HMI+PLC unit has physical size benefits as it was always take up less panel space than separate units. The combined HMI+PLC unit also require one cable for communication that was made between the HMI and the PLC, a single power source powers both the HMI and PLC. HMI & PLC systems provide strong external device integration, I/O, and communications options and provide a full range of analog, discrete, motion control and specialty modules to meet the needs of the research study. These modules are typically part of a local or remote I/O chassis. In general, the HMI focuses on three different starting points in order to ensure safety, health, efficiency, and productivity: the human being, the machine, and the environment. At the same time, these are also identified as risk sources which may endanger safety and productivity at work.

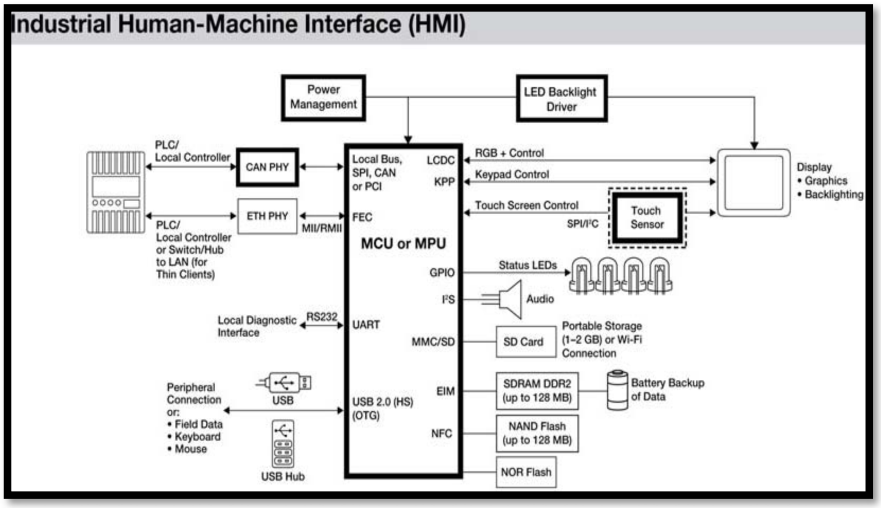


Figure 28: HMI/PLC block diagram

This module was completely integrated with all other libraries, making it possible to simulate systems while monitoring your machine in signals. Therefore, it was easy to test and to verify the functionality of the application and to accommodate developments in the controller protocol and hardware, drivers are continuously updated. Push buttons were grouped to make the panel easy to operation during testing. Colour codes were selected with reference to the standard industrial codes. Navigation to and from the HMI touch panel was easy and figured.

Programmable logic controller (PLC) that received the data from the input/output sensors, and converts the data into logical combinations and the HMI allowed control and data acquisition in the entire system. All the registers of the PLC were known in order to program an HMI to operate a PLC properly. HMI communicate with the PLC via PROFIBUS protocol using built in standard software. A 187.5kb /s baud rate was synchronising, so that there was no miscommunication occurred. A PLC was connected to the HM to communicate through a communication processor. Such a communication processor was processing a certain number of telegrams during a time unit. The property of a communication system was implemented in 187.5kb/s rate. Based on the value and a certain telegram length, the approximate number of telegrams per time unit was determined. With an increasing transmittal rate, the maximum number of telegrams per time unit was also increase.

The HMI accesses specific registers on the PLC and makes decisions based on the state of those registers. A technique activates data transmission of consecutive registers, and supports downloading data from the memory of the recipe card to the PLC. The HMI has a centralized unit of control, which allows for decisions based on real time events in a visual manner. The HMI must be in constant communication with a controller so the HMI can receive real time data from controller. The advantage of using an HMI over the PLC was that the PLC does not provide any real-time feedback or update during execution.

7.3 Software and program structure

The STEP 7 (Siemens software version 7) is the standard software package used for configuring and programming SIMATIC programmable logic controllers. It is part of the SIMATIC industry software. There are number versions of the STEP 7 Standard package, but on this research version STEP 7 V5.4 Simatic 300(1), CPU 313C-2DP and other software were used on this research as follows:

(i) Siemens Step7, version 5.4

(ii) Wcc Flexible advance V2008 for HMI and Verificator software V 5.0

(iii) Wcc trend analyzer tool v 2-8-0 and Wcc service pack 3 graphics

(iv) Window 7 operating system and GSD Revision 5.1.

The S7 (software version 7) programmable controller consists of a power supply unit, a CPU, input and output modules. The programmable logic controller (PLC) monitors and controls the process plant with the S7 program. The I/O modules are addressed in the S7 program via the addresses. Before the program was written, the hardware was configure first and then program the blocks because the advantage of this was that STEP 7 displays the possible addresses in the Hardware Configuration Editor as shown in figure 29. The project structure was used to store and arrange all the data and programs in order in a hierarchical structure and was available at any time. A program structure in a CPU principally runs with two different programs which were operating system and user program.

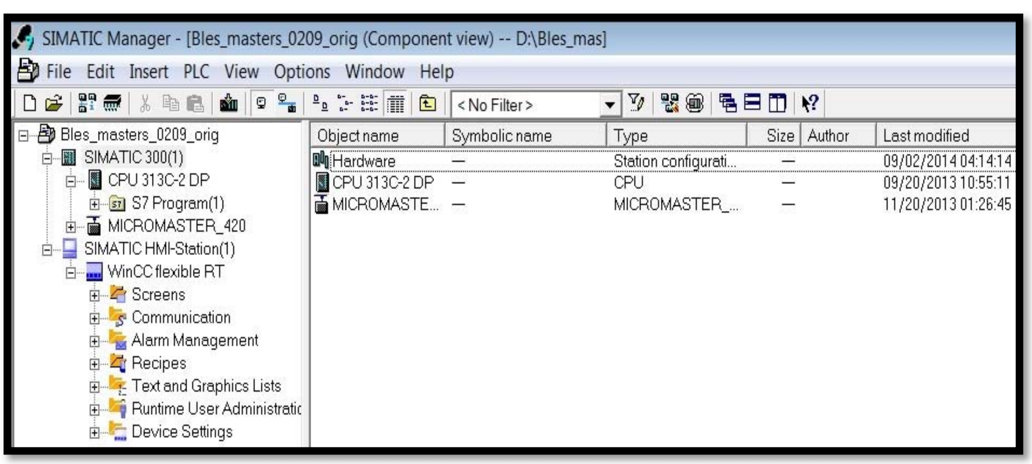


Figure 29: Project structure

Operating system organizes all CPU functions and sequences not associated with a specific control task for example (update of the process image table of the inputs and output of the process image table of the outputs). The user program contains all the functions required to process specific automation task for example (processing process data, generating logical links of binary signals, fetching and evaluating analog signal and also specifying binary signals for output, output of analog values). After a project was created, all other tasks were executed in this research [21] and [22].

The properties of the CPU 313 have a special significance for the performance of the system. In the dialog boxes for a CPU, the following properties as start-up characteristics, local data areas and priorities for interrupts, memory areas, retentive behaviour, clock memory and protection level. The data type which was previously added automatically to the symbol table determines the type of the signal to be processed for the CPU. In this way, assign symbolic names to all the absolute addresses of the inputs and outputs which were program requires. In the symbol table, you assign a symbolic name and the data type to all the absolute addresses which are address in the program.

The reason to configure the hardware was to specify in a configuration table which modules to use for process plant solution and which addresses were to be used to access the modules from the user program, the symbolic structure is shown on figure 30. The properties of the modules were assigned using parameters. Only one symbol table was created per S7 program, regardless of which programming language was selected. Setting symbolic names and comments for the process signals (inputs/outputs), bit memory, and blocks was providing an overview of the status of the programmable logic controller (PLC) and an also overview display symbols to show whether every module has a fault or not.

In STEP 7 SIMATIC manager, Organization Blocks (OBs) were the interface between the operating system of the CPU and the user program. OB1 was processed cyclically by the CPU. The CPU reads line by line and executes the program commands. When the CPU returns to the first program line, it has completed exactly one cycle. The time required for this was called as the scan cycle time. When OB1 has been executed, the operating system starts it again. Cyclic execution of OB1 was started after the start-up has been completed. STEP 7 calls other function blocks (FBs, SFBs, FCs, DBs and SFCs) in OB1 as shown in figure 31.

When OB1 has been executed, the operating system sends global data. Before restarting OB1, the operating system writes the process-image output table to the output modules, updates the process image input table and receives any global data for the CPU. S7 monitors the maximum scan time, ensuring a maximum response time. The value for the maximum scan time was pre-set to 10 ms. The operating system was delaying the start of a new cycle (writing of the process image output table to the output modules) until the minimum scan time has been reached.

Symbol Editor - [S7 Program(1) (Symbols) -- Bles_masters_0209_orig\SIMATIC 300(1)\CPU 313C-2 DP]

Symbol Table Edit Insert View Options Window Help

All Symbols

	Status	Symbol /	Address	Data type	Comment
1		COMM_FLT	OB 87	OB 87	Communication Fault
2		COMPLETE RESTART	OB 100	OB 100	Complete Restart
3		CONT_C	FB 41	FB 41	Continuous Control
4		CYC_INT5	OB 35	OB 35	Cyclic Interrupt 5
5		DPRD_DAT	SFC 14	SFC 14	Read Consistent Data of a Standard DP Slave
6		DPWR_DAT	SFC 15	SFC 15	Write Consistent Data to a Standard DP Slave
7		DRIVE 1 START	M 10.0	BOOL	Drive 1 - HMI Start
8		DRIVE 1 STOP	M 10.1	BOOL	Drive 1 - HMI Stop
9		DRIVE 2 START	M 10.2	BOOL	Drive 2 - HMI Start
10		DRIVE 2 STOP	M 10.3	BOOL	Drive 2 - HMI Stop
11		DRIVE NO:1 CONTROL	FC 2	FC 2	
12		DRIVE NO:1 CONTROL WORD	DB 10	DB 10	CONTROL WORD 1
13		DRIVE NO:1 STATUS WORD	DB 20	DB 20	STATUS WORD 1
14		DRIVE NO:2 CONTROL	FC 3	FC 3	
15		DRIVE NO:2 CONTROL WORD	DB 30	DB 30	CONTROL WORD
16		DRIVE NO:2 STATUS WORD	DB 40	DB 40	STATUS WORD
17		Flow-meter 1 _PA	PIW 256	WORD	
18		Flow-meter 1_PA	MW 104	WORD	
19		Flow-meter 2_4-20MA	PIW 261	WORD	Flow 2_4-20MA
20		Flow 2_4-20MA	MW 102	WORD	
21		LEVEL_PA	PIW 275	WORD	
22		LEVEL_TRANS_PA	MW 106	WORD	
23		main pid reference	MD 350	DWORD	main setpoint of pid controller
24		MOD_ERR	OB 122	OB 122	Module Access Error
25		OBNL_FLT	OB 85	OB 85	OB Not Loaded Fault
26		PID - OFF	M 40.0	BOOL	PID - DISABLE
27		PID - ON	M 60.0	BOOL	PID - ENABLE
28		PID SWITCHED ON/OFF	M 70.0	BOOL	PID ON/OFF OPERATION
29		PROG_ERR	OB 121	OB 121	Programming Error
30		RACK_FLT	OB 86	OB 86	Loss of Rack Fault
31		RESET DRIVE FAULTS	M 30.0	BOOL	Reset drives.
32		TCONT_S	FB 59	FB 59	temperature PID step controller

Figure 30: Program symbolic structure

When OB1 has been executed, the operating system sends global data. Before restarting OB1, the operating system writes the process-image output table to the output modules, updates the process image input table and receives any global data for the CPU. S7 monitors the maximum scan time, ensuring a maximum response time. The value for the maximum scan time was pre-set to 10 ms. The operating system was delaying the start of a new cycle (writing of the process image output table to the output modules) until the minimum scan time has been reached.

A function block (FB) is a block with a built in memory and contains a program that was always executed when the FB was called by a different logic block. The parameters that were transferred to FB41 and FB59 and the static variables are saved in the data block. Temporary variables are saved in the local data stack. Data saved in the instance data block were not lost when execution of the FB was complete. Data saved in the local data stack are, however, lost when execution of the FB was completed. In the figure 31, the calling FB was FB41 and FB59 (Drive1 and Drive2) and assign initial values to the formal parameters in the declaration section of the FB. These values are written into the instance data block associated with the FB. The data block associated with the FB when it was called to determine which drive was controlled.

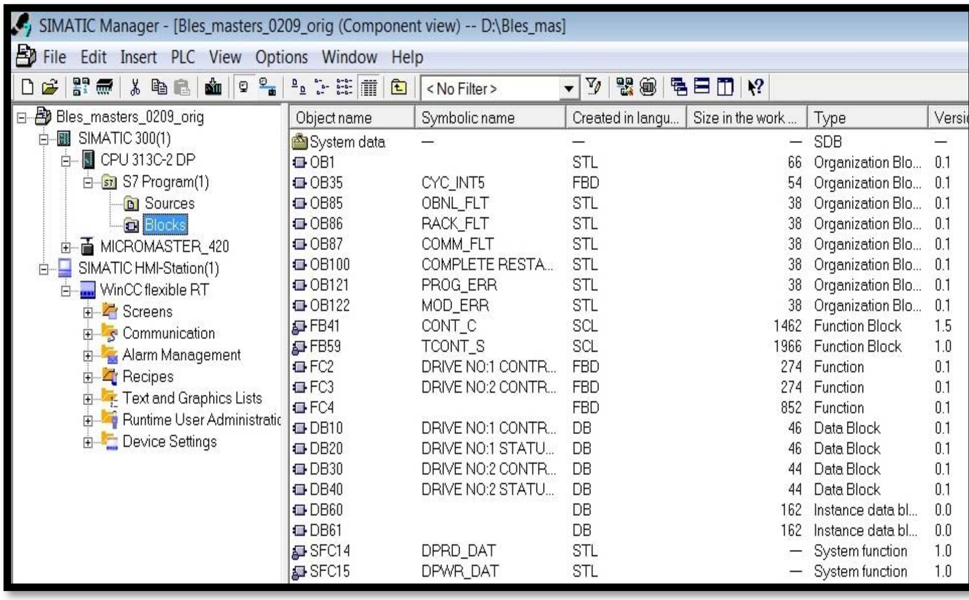
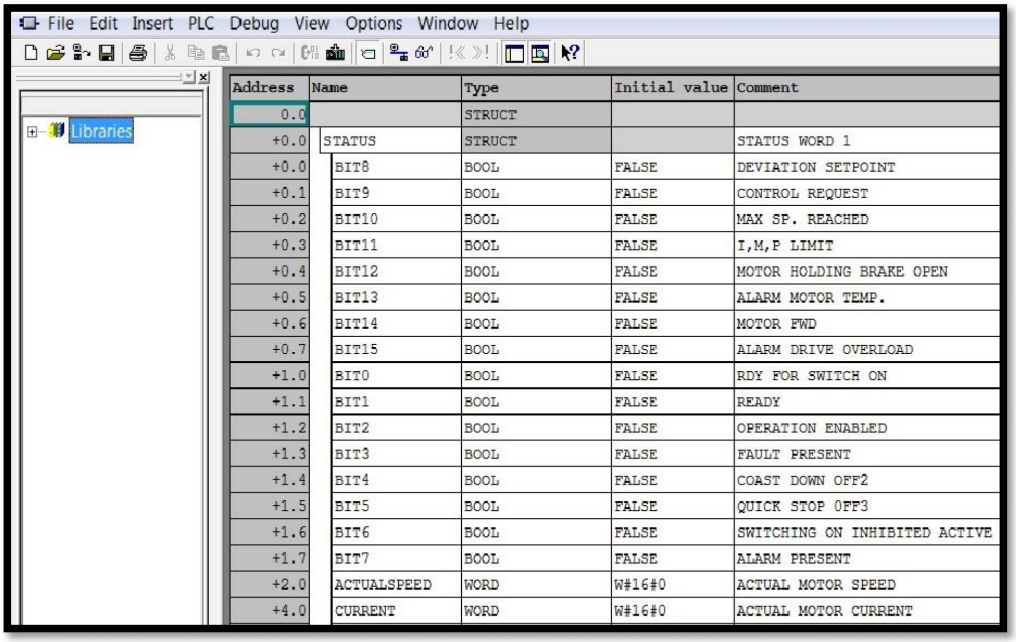


Figure 31: Overview of the Organization Blocks

Functions (FC) belong to the blocks that are on the program. A function is a logic block without memory. Temporary variables belonging to the FC are saved in the local data stack. This data is then lost when the FC has been executed. An FC contains a program section that was always executed when the FC was called by a different logic block. Since an FC does not have any memory of its own, actual parameters are specified for it. Actual parameters replace the formal parameters when the function was called. Actual parameters to the formal parameters of an FC2 FC3 and FC4 as shown in figure 30 are for (Drive 1 and Drive 2 "Control Start").

The input, output and in/out parameters used by the FC2, FC3 and FC4 were saved as pointers to the actual parameters of the logic block that called the functions. Function call (FC2 and FC3) were used for control start word and status word of Drive 1 and Drive 2, as shown figure 32 and figure 33. Data Block (DB) were used to store user data, in other words, data blocks contain variable data with which the user program works. A data block is assigned to every function block call that transfers parameters. The actual parameters and the static data of the function block are saved in the instance Data Block. The variables declared in the function block determined the structure of the instance data block. A data blocks store user data that can be accessed by all other block variables with the data type of the called function block for each individual instance in the declaration section of the calling function block. The call within the function block does not then require an instance data block, only the symbolic name of the variable.

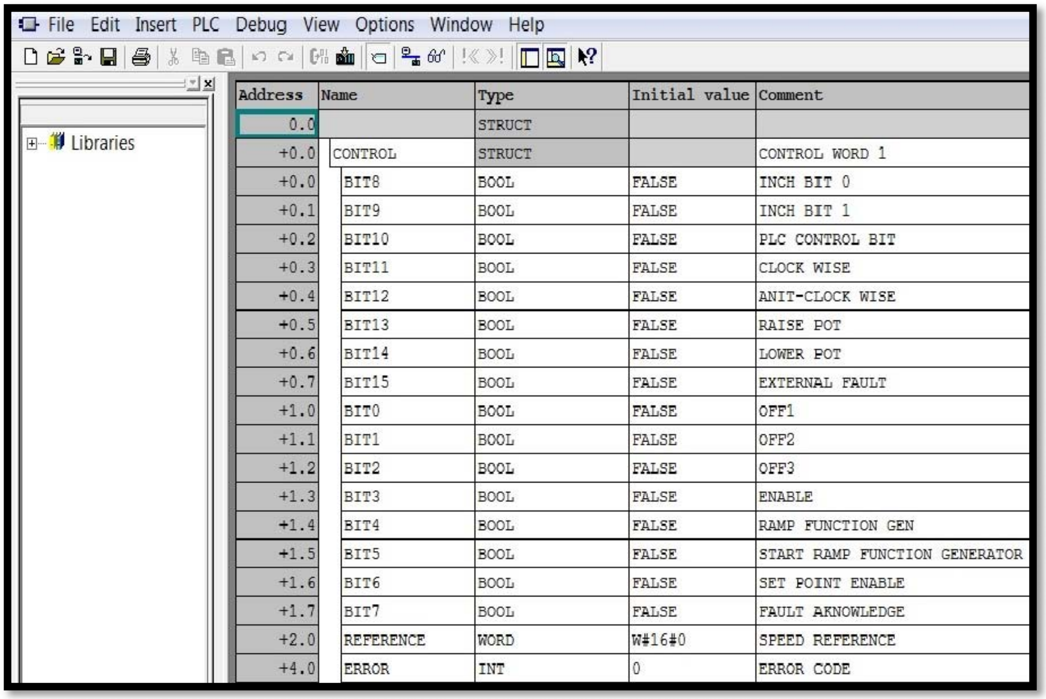


Address	Name	Type	Initial value	Comment
0.0		STRUCT		
+0.0	STATUS	STRUCT		STATUS WORD 1
+0.0	BIT8	BOOL	FALSE	DEVIATION SETPOINT
+0.1	BIT9	BOOL	FALSE	CONTROL REQUEST
+0.2	BIT10	BOOL	FALSE	MAX SP. REACHED
+0.3	BIT11	BOOL	FALSE	I,M,P LIMIT
+0.4	BIT12	BOOL	FALSE	MOTOR HOLDING BRAKE OPEN
+0.5	BIT13	BOOL	FALSE	ALARM MOTOR TEMP.
+0.6	BIT14	BOOL	FALSE	MOTOR FWD
+0.7	BIT15	BOOL	FALSE	ALARM DRIVE OVERLOAD
+1.0	BIT0	BOOL	FALSE	RDY FOR SWITCH ON
+1.1	BIT1	BOOL	FALSE	READY
+1.2	BIT2	BOOL	FALSE	OPERATION ENABLED
+1.3	BIT3	BOOL	FALSE	FAULT PRESENT
+1.4	BIT4	BOOL	FALSE	COAST DOWN OFF2
+1.5	BIT5	BOOL	FALSE	QUICK STOP OFF3
+1.6	BIT6	BOOL	FALSE	SWITCHING ON INHIBITED ACTIVE
+1.7	BIT7	BOOL	FALSE	ALARM PRESENT
+2.0	ACTUALSPEED	WORD	W#16#0	ACTUAL MOTOR SPEED
+4.0	CURRENT	WORD	W#16#0	ACTUAL MOTOR CURRENT

Figure 32: FC2 Status word for Drive1 and 2

Figure 31 shows that DB10, DB20, DB30, DB40, DB60 and DB61 were saved in different data blocks for each specific drive data. A system function (SFC) is a pre-programmed function that was integrated on the S7 (software version 7) CPU. SFC14 and SFC15 are part of the operating system and are not loaded as part of the program as shown in figure 33. S7 (software version 7) CPU provides SFC14 and SFC15 are for the following functions:

(i) Copying and block functions
(ii) Checking the program
(iii) Handling the clock and run-time meters
(iv) Transferring data sets
(v) Transferring events from a CPU to all other CPUs in multi-computing mode
(vi) Handling time-of-day and time-delay interrupts
(vii) Handling synchronous errors, interrupts, and asynchronous errors
(viii) Information on static and dynamic system data, for example, diagnostics
(ix) Process image updating and bit field processing
(x) Addressing modules
(xi) Distributed I/O
(xii) Global data communication
(xiii) Communication via non-configured connections
(xiv) Generating block-related messages



File Edit Insert PLC Debug View Options Window Help

Libraries

Address	Name	Type	Initial value	Comment
0.0		STRUCT		
+0.0	CONTROL	STRUCT		CONTROL WORD 1
+0.0	BIT8	BOOL	FALSE	INCH BIT 0
+0.1	BIT9	BOOL	FALSE	INCH BIT 1
+0.2	BIT10	BOOL	FALSE	PLC CONTROL BIT
+0.3	BIT11	BOOL	FALSE	CLOCK WISE
+0.4	BIT12	BOOL	FALSE	ANIT-CLOCK WISE
+0.5	BIT13	BOOL	FALSE	RAISE POT
+0.6	BIT14	BOOL	FALSE	LOWER POT
+0.7	BIT15	BOOL	FALSE	EXTERNAL FAULT
+1.0	BIT0	BOOL	FALSE	OFF1
+1.1	BIT1	BOOL	FALSE	OFF2
+1.2	BIT2	BOOL	FALSE	OFF3
+1.3	BIT3	BOOL	FALSE	ENABLE
+1.4	BIT4	BOOL	FALSE	RAMP FUNCTION GEN
+1.5	BIT5	BOOL	FALSE	START RAMP FUNCTION GENERATOR
+1.6	BIT6	BOOL	FALSE	SET POINT ENABLE
+1.7	BIT7	BOOL	FALSE	FAULT AKNOWLEDGE
+2.0	REFERENCE	WORD	W#16#0	SPEED REFERENCE
+4.0	ERROR	INT	0	ERROR CODE

Figure 33: FC3 control word for Drive1 and 2

7.4 System configuration and network structure

In order for a process plant to be automated, it was required to configure a central structure and configure the distributed I/O. The term (configuring) refers to the arranging of racks, modules, distributed I/O (DP) racks, and interface submodules in a hardware configuration window. Racks were characterized by a configuration table that permits a specified address of modules to be inserted. As Figure 34 shows, the hardware labels each slot and all input/output modules assigned with their unique addresses.

Input/output (I/O) addresses are required in order to read inputs and set outputs in the user program. It was very important to configure the selected hardware in the hardware configuration of (PCS7) and check it for stability. Main settings were parameterized so that the process control system (PCS7), using this information, independently sets up all blocks needed in the automation stations for communication between 2 drives, HMI, ET200S, flowmeters, level transmitter and the control level. The figure 34 shows how a real structure was converted into a configuration table.

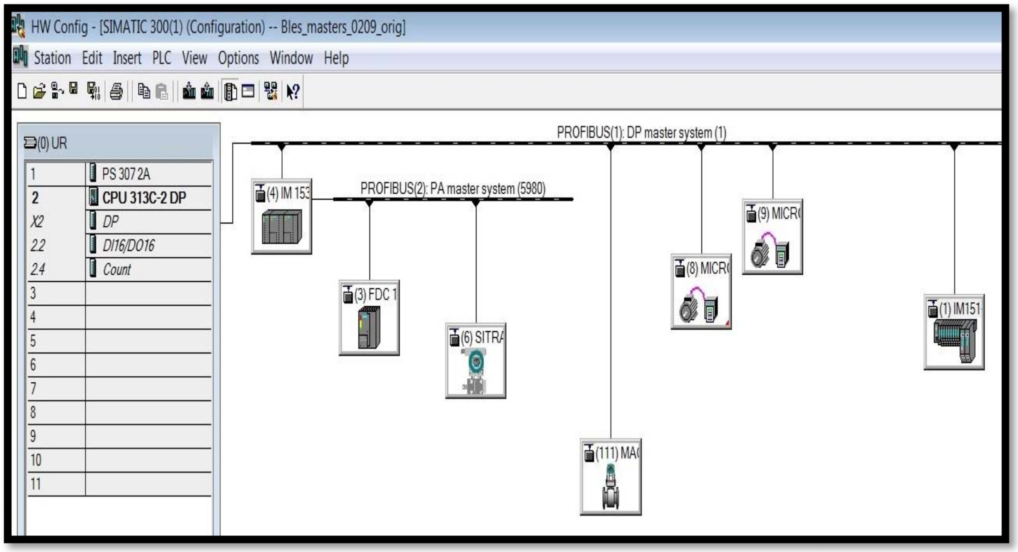


Figure 34: hardware and communication connection with S7

The system hardware used PROFIBUS communication and each station had its own unique addresses specified and also baud transmission rate at 187.5kbit/s. Configuring a programmable logic controller involves the use of two windows: The station window was placed in the rack for the station structure and the (Hardware Catalog) window for selecting the required hardware components, similar to rack, modules, and interface submodules. STEP 7 assigns input and output addresses when modules are placed in the configuration table. This means every module has a start address (address of the first channel); the lower part of the station window shows a detailed view of the inserted/selected rack.

DP master (CPU 313-2DP), STEP 7 draws a link that represents the master system. At the end of the link, place the DP slaves that are assigned to this DP master using the (Hardware Catalog) window under PROFIBUS-DP. As a DP master system was always connected to a PROFIBUS subnet, STEP 7 automatically displays dialog boxes for setting the subnet properties and the PROFIBUS address when the DP components are placed. The slave structure (DP identifiers and modules/submodules) and I/O addresses are displayed in the detailed view of the station window as shown in figure 35. The distributed I/O refers to master systems, comprising DP (distributed I/O) master and DP slaves that are connected via a bus cable and communicate with each other via the PROFIBUS-DP protocol. In order to configure a CPU 313-2 DP as an intelligent DP slave, continue as DP Slaves instructions [23].

To configure the ET200M as field device on the PROFIBUS network, select the matching interface module from the catalog in the folder PROFIBUS-DP, ET200M, to the CPU’s master system. This was done by dragging these modules to the respective slot within the ET200M. The IO addresses of the individual modules were set in their properties, as shown in figure 35. Once the configuration was finished, accept it with the button to save and compile. , PROFIBUS-DP, ET200M, IM 151-1HF, AI-300 these modules indicated in the footer of the hardware catalog after selecting a component. The figure 36 shown the tool NetPro provides a good overview of the components and networks of the process plant. All stations were connected to each other by means of the PROFIBUS, and the ET200M was interfaced by means of PROFIBUS with the SIMATIC300.

(1) IM151-1 Standard

Slot	Module	Order number	I Address	Q Address
1	PM-E DC24V	6ES7 138-4CA01-0AA0		
2	2AI I 2WIRE ST	6ES7 134-4GB00-0AB0	261...264	
3	4DI DC24V ST	6ES7 131-4BD01-0AA0	0.0...0.3	
4	2RO NO/NC 24..230V/	6ES7 132-4HB10-0AB0		0.0...0.1
5	2RO NO/NC 24..230V/	6ES7 132-4HB10-0AB0		1.0...1.1
6	2RO NO/NC 24..230V/	6ES7 132-4HB10-0AB0		2.0...2.1
7	2RO NO/NC 24..230V/	6ES7 132-4HB10-0AB0		3.0...3.1
8	2RO NO/NC 24..230V/	6ES7 132-4HB10-0AB0		4.0...4.1

Figure 35: ET200 interface address of the modules

The PROFIBUS protocol was designed for high-speed communications with distributed I/O devices. PROFIBUS networks typically have one master and several slave I/O devices. The master device was configured to know what types of I/O slaves were connected and at what addresses. The master initializes the network and verifies that the slave devices on the network match the S7-300: Master and ET 200M Slave configuration. The master continuously writes output data to the slaves and reads input data from them. Figure 36 shows PROFIBUS Network. When a DP master configures a slave device successfully, it then communicate with all slave device at the siutable transmision baud rates of 187.5 kbaud.

There are two reasons for configuring a PROFIBUS-PA device:

(i) The adjustment of the operating parameters of the device to calibrate it for the application at hand and in this case the corresponding operating instructions were used.

(ii) The adjustment of the profile parameters of the device in order to scale or simulate the cyclic measured value output to the PLC.

The operating parameters were set using the local operating elements of the device and these parameters were also adjusted by the acyclic services of the PROFIBUS-DP system with the operating and display program. Profile parameters were accessible only through the cyclic services of the PROFIBUS-DP system. The standard model of the PROFIBUS-PA device was subdivided as follows:

(i) PROFIBUS-PA Block model

(ii) Device management

(iii) Physical Block

(iv) Transducer Block

(v) Function Block

(vi) Operating program

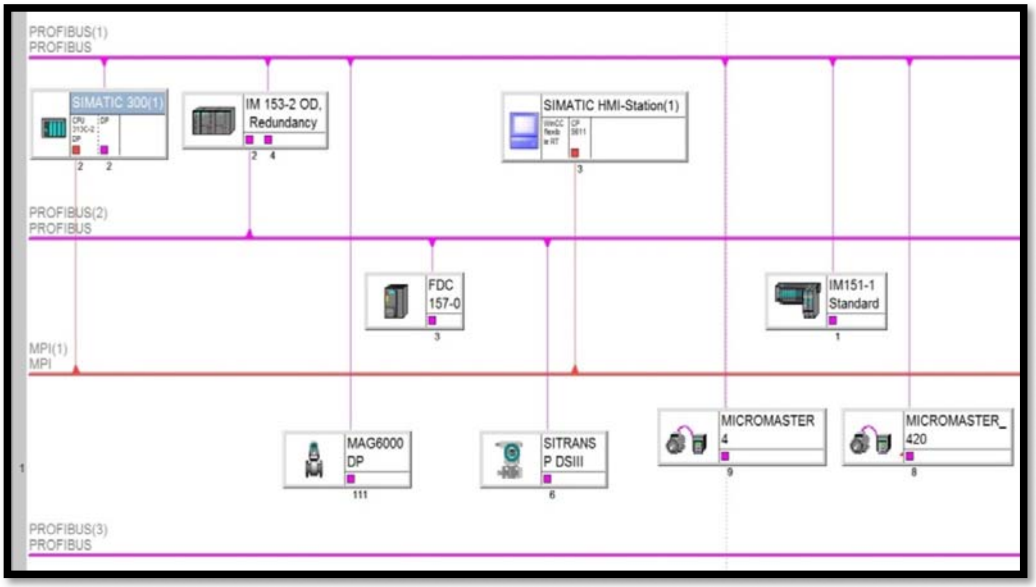


Figure 36: Process plant network structure

The PROFIBUS-PA Block model described several parameters that were used to realise a device and required parameters always be present, Optional parameters were only present when required, manufacturer-specific parameters used to realise device functions that were not in the standard profile. A manufacturer's operating tool or a device description was required for the operation. In the case of PROFIBUS-PA device that conform to the standard and these parameters were managed in Block objects. Within the Blocks, the individual parameters were managed using relative files. The sensor signal was converted to a measured value by the transducer Block and transmitted to the function Block. Here the measured value was scaled or limited before it was made available as the output value to the cyclic services of the PLC. As shown in figure 37, the Block model comprises four Blocks:

(i) Device management

(ii) Physical Block

(iii) Transducer block

(iv) function Block

Described the detail in the following sections.

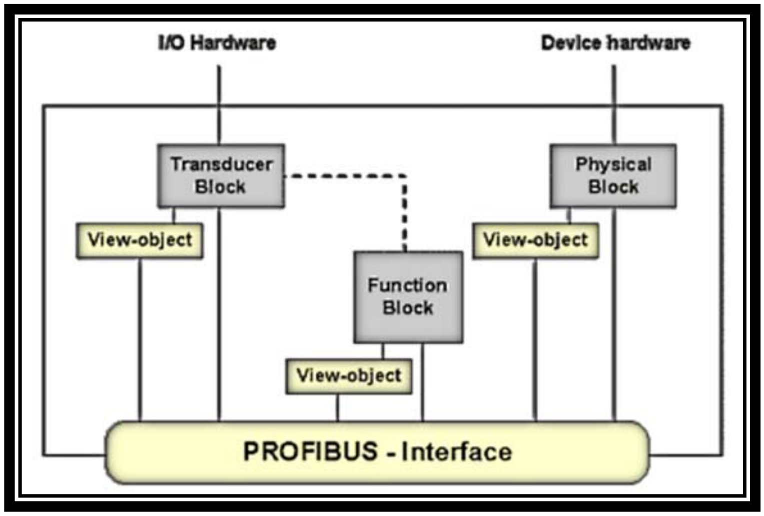


Figure 37: PROFIBUS-PA blocks model of a PA flowmeter

The PLC outputs a set point value that serves as the input value to the PA device. After any scaling, the set point value was transmitted to the transducer block as the output value of the function Block. It processes the value and outputs a signal that drives the device to the desired position. The parameters on the Block structure assigned to the individual Blocks for the data structures and data formats that were specified in the PROFIBUS standard. The structures were designed such that the data that was stored and transmitted in an ordered and interpretable manner.

The physical Block contains the properties of the field device. These are device parameters and functions that are not dependent upon the measurement method. This ensured that the function and transducer Blocks were independent of the hardware. Transducer Blocks stand as separating elements between the sensor and the function Block. They process the signal from the sensor and output a value that was transmitted via a device-independent interface to the function Block. The transducer blocks reflect the measurement principles. Moreover, Blocks also exist for devices with a binary input or output signal. The function Blocks contain the basic automation functions. Since the application program demands that a cyclic value always behaves in the same manner, the Blocks were designed to be as independent as possible from the sensor and the fieldbus. The analog input Block was fed by the transducer Block of a particular PA flowmeter.

The first function in the processing chain allows the measured value to be replaced by a simulated value when required. Then the input value was normalised to a value between 0 and 1. Normally, the lower and upper range values of the transducer Block were used for scaling. No limits were set on the scaling values, and values beyond the end-values are correctly scaled. The totalizer Block was used when a process variable was summed over a period of time. This was the case for flowmeter, whereby for PA flowmeter totalises was activated for both volume and mass measurements. The Block was fed by the transducer Block of a PA flowmeter, which provides a measured value and status. The following documents provided a clear clarification about PLC program, system settings, network structure, and details of each device and configuration of the system. The reason for this information or data was to show the reader details of the entire process plant structure and functions.

CHAPTER EIGHT: PROFIBUS COMMUNICATION PROTOCOL

8.1 Introduction

PROFIBUS communication protocol is a reliable and globally acknowledged protocol used for digital communication between the host and smart devices and enables powerful control and monitoring system for the user. In simple words PROFIBUS provides two dimensional communication and data access. An early version of PROFIBUS was PROFIBUS FMS, for (Fieldbus Message Specification). PROFIBUS FMS was intended to interface between programmable logic controllers and PLCs, sending complex data information between them [24] and [25]. PROFIBUS are generally used in process automation and have two types PROFIBUS which are PROFIBUS DP and PROFIBUS PA. DP and PA contain the same protocols and both were linked using a coupler device. In this proposed solution, PROFIBUS DP (Decentralized Peripherals) was used for data exchange with field devices like sensors and actuators through a programmable logic controller in process plant application and PROFIBUS PA (Process Automation) was used for interface measuring instruments through a process control system in process plant application. Also discusses the implementation plans for PROFIBUS communication and analyse its function in the process plant. Clarify the important aspects of PROFIBUS communication as follows:

(i) PROFIBUS wire polarity (A=green, B=red) and maximum number of devices per segment.
(ii) PROFIBUS cable, connectors and adequate baud rate with respect to cable length.
(iii) Assign of unique address for each node and protection of cable from short circuit and breaks.
(iv) Data transmission speed and amount of data per device.

8.2 PROFIBUS data transmission

PROFIBUS allows communication between devices of different manufacturers without any special Interface adjustment and was used for both high-speed time critical applications and complex communication tasks. PROFIBUS distinguishes between the following types of devices:

(i) Master devices determine the data communication on the bus. A master sends messages without an external request when it holds the bus access rights (the token).
(ii) Slave devices are peripherals such as I/O devices, drives, flowmeter and level transmitter. They do not have bus access rights and they only acknowledge received messages or send messages to the master when requested to do so as shown in figure 38. Slaves are called passive stations. Every DP master and DP slave in the PROFIBUS network must be assigned a unique PROFIBUS address in the range 0 through 125.

In this communication, data exchange takes place between the DP master and DP slaves, (I/O modules) by means of the DP master. The DP master polls each configured DP slave, one after the other, in the master's polling list within the DP master system and transmits the output data or receives the input data from the slaves. Only when all devices in a subnet have unique addresses and actual structure matches the network configuration and then the settings can be load via the remote engineering PC.

In this PROFIBUS communication, input data from DP slaves was read very rapidly by intelligent DP master on the same physical PROFIBUS-DP subnet. PROFIBUS application profiles describe the interaction of the communications protocol with the transmission technology being used. PROFIBUS used RS 485 transmission technology and the application area includes all areas in which high transmission speed and simple, twisted pair shielded copper cable with one conductor pair was used. PROFIBUS one unique transmission speed selected is 187.5kbit/sec for all devices on the bus system.

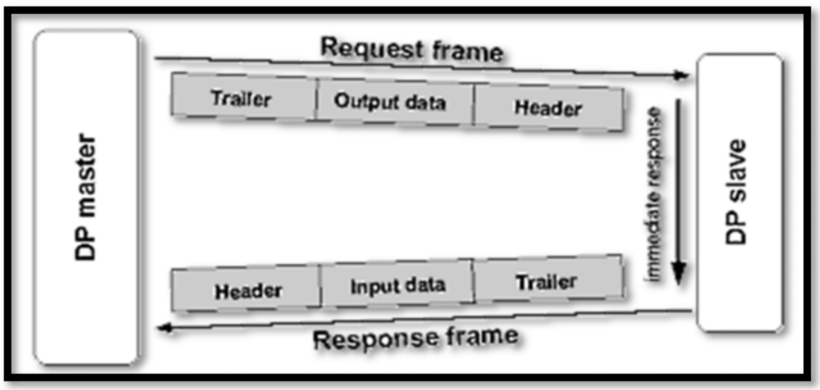


Figure 38: Cyclic user data transmission in DP [25]

During communication between complex automation systems (masters), it was ensured that each of these stations gets sufficient time to perform its communication tasks within a precisely defined time interval and communication between a programmable logic controller and its assigned peripherals (slaves), cyclic, real time data transmission needs were implemented as fast, shown in figure 38. Data exchange with the distributed devices was mainly cyclic and also offers extended acyclic communication services for the parameterization, operation, monitoring and alarm handling of intelligent field devices.

The central controller (PLC) cyclically reads the input information from the slaves and cyclically writes the output information to the slaves. The bus cycle time was shorter than the program cycle time of the central automation system, for this application it was approximately 10 msec. The amount of input and output information depends on the device type. A maximum of 244 bytes of input data and

244 bytes of output data was permitted. The programmable controller was the central control component and the slaves are decentrally linked to the PLC via the PROFIBUS transmission.

Data transmission between the DPMaster (PLC) and the slaves was divided into three phases: parameterization, configuration and data transfer. Before a DP slave enters the data transfer phase, in the parameterization and configuration phase it was checked, whether the planned configuration matches the actual device configuration. In the course of this check, the device type, format and length information as well as the number of inputs and outputs was agreed. These tests provided the user reliable protection against parameterization errors. In addition to the user data transfer, which was executed automatically by the DPMaster, new parameterization data were sent to the slaves at the request of the user.

A transmission data frame is shown in Figure 39. The frame begins with a start and this was then followed with a message identifier. For device network this was a 5 bit address code and a 6 bit command code. The ready to receive it bit was set by the receiving machine and both the sender and listener share the same wire. If the receiving machine does not set this bit the remainder of the message was aborted, and the message was resent later. While sending the first few bits, the sender monitors the bits to ensure that the bits send are heard the same way. If the bits do not agree, then another node on the network has tried to write a message at the same time – if there was a collision.

The two devices then wait a period of time, based on their identifier and then start to resend. The second node was then detects the message, and waits until it was done. The next 6 bits indicate the number of bytes to be sent, from 0 to 8. This was followed by two sets of bits for CRC (Cyclic Redundancy Check) error checking, this was a checksum of earlier bits. The next bit ACK slot was set by the receiving node if the data was received correctly. If there was a CRC error this bit would not be set, and the message would be resent. The remaining bits end the transmission. The ends of frame bits were equivalent to stop bits. There was a delay of at least 3 bits before the next message begins.

1 bit	start of frame	
11 bits	identifier	arbitration field
1 bit	ready to receive it	
6 bits	control field - contains number of data bytes	
0-8 bytes	data - the information to be passed	
15 bits	CRC sequence	
1 bit	CRC delimiter	
1 bit	ACK slot - other listeners turn this on to indicate frame received	
1 bit	ACK delimiter	
7 bits	end of frame	
>= 3 bits	delay before next frame	

Figure 39: Cyclic transmission data frame

A device with several measured values transmits correspondingly more bytes. In this study of the DP flowmeter, a cyclic telegram of 2 bytes (1 bytes input and 1 byte output data) was transmitted at maximum configuration. By using the data exchange service, a PLC can transmit its output data to the DP flowmeter, see table 2 and read the input data from the response telegram, see table 3. The cyclic data telegram for the maximum configuration of the Flowmeter has the following structure; In order to integrate the field devices into the bus system, the PROFIBUS-DP system required a description of the device parameters such as output data, input data, data format, data length and the transmission rates supported. The devices data were contained in the device database file (the so-called GSD file), which was required by the PROFIBUS-DP master to be able recognised then on the network.

Table 2: Output data PLC

Byte	Data	Access	Data format
0	Control 0 ⇒ 1: Reset totalisor 1 0 ⇒ 2: Reset totalisor 2 0 ⇒ 3: Reset totalisor 1 + 2 0 ⇒ 4: Zero point calibration 0 ⇒ 5: Positive zero return on 0 ⇒ 6: Positive zero return off	Write	Integer8, The control command is triggered by a change in the input data of the cyclic services to another value. A change from any bit pattern has no effect.

Table 3: Input data of the DP flowmeter

Byte	Data	Access	Data format	Unit
0 – 3	Mass flow	Read	32-bit floating point number (IEEE	kg/s
4	Status mass flow	Read	80h = OK	–
5 – 8	Totalisor 1	Read	32-bit floating point number (IEEE	kg
9	Status totalisor 1	Read	80h = OK	–
10 – 13	Density	Read	32-bit floating point number (IEEE	kg/m^3
14	Status density	Read	80h = OK	–
15 – 18	Temperature	Read	32-Bit floating point number (IEEE	K
19	Status temperature	Read	80h = OK	–
20 – 23	Totalisor 2	Read	32-bit floating point number (IEEE	off
24	Status totalisor 2	Read	80h = OK	–
25 – 28	Volumetric flow	Read	32-bit floating point number (IEEE	l/s
29	Status volumetric flow	Read	80h = OK	–
30 – 33	Standard volumetric flow	Read	32-bit floating point number (IEEE	Nl/s
34	Status standard volumetric	Read	80h = OK	–
35 – 38	Target medium flow	Read	32-bit floating point number (IEEE	kg/s; l/s
39	Status target medium flow	Read	80h = OK	–
40 – 43	Carrier medium flow	Read	32-bit floating point number (IEEE	kg/s; l/s
44	Status carrier medium flow	Read	80h = OK	–
45 – 48	Calculated density	Read	32-bit floating point number (IEEE	%
49	Status calculated density	Read	80h = OK	–

All participants in a PROFIBUS-DP system are governing by 187.5 Kbit/s transmission rate. The speed of data exchange was determined by the slowest participant and all active participants on the PROFIBUS are operating with the same bus parameters. Since PROFIBUS-PA segment was used to extend the PROFIBUS-DP and to interface hardwired flowmeter, to obtain the highest possible transmission rate, a DP/PA coupler was preferred as an interface. PROFIBUS-PA has been designed to satisfy the requirements of process plant and the device was powered over the bus cable and data are transferred via the IEC 61158-2 physical layer, which allows great freedom in the PROFIBUS topology. Since the protocols, physical layer and transmission rates of PROFIBUS-DP and PROFIBUS-PA were different; the PROFIBUS-PA segment was connected to the PROFIBUS-DP system via a segment coupler.

The SIMATIC ET200M was selected as the distributed IO device. The IO modules of the proven automation system S7-300 are connected to an interface module (IM 153-2) that ensured the communication to the automation station (AS). To the left of interface module IM 151-1 several digital and analog input and output modules were connected. The signal coming from the flowmeter was sent directly to the routing level that was placed below the input and output modules. The SIMATIC ET200M was interfaced with a PROFIBUS DP line of the automation station (AS). The ET200M slots were assigned input and output modules.

The hardwired flowmeter on the PROFIBUS-PA side was no longer being directly polled using the cyclic services. Instead, the DP/PA coupler link collected the device data in the interface module IM 151-1 which was read cyclically by the PLC. Hence a DP/PA coupler link was mapped in the PLC. On the PROFIBUS-PA side, DP/PA coupler link acts as the bus master, as shown in figure 40. It polls the field device data cyclically and stores them in a buffer. PA flowmeter was assigned a PROFIBUS-PA address that was unique for the DP/PA coupler link.

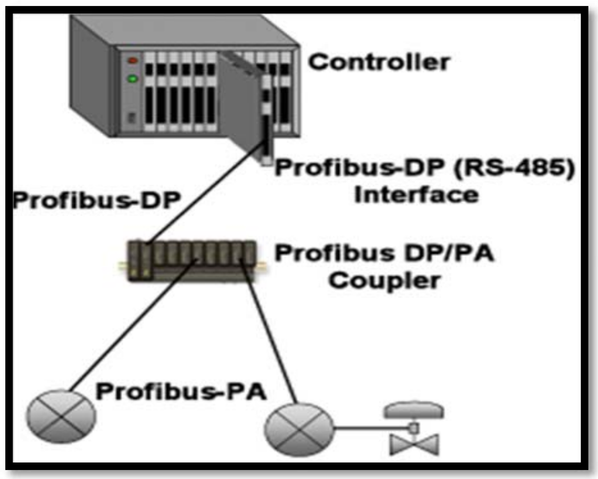


Figure 40: DP/PA coupler link

The majority of PROFIBUS-PA devices transmit measured value and status in 5 bytes, but PA flowmeter, cyclic telegram of 2 bytes (1 bytes measured value and 1 byte status) was transmitted at maximum configuration. See table 3; Input data of the DP flowmeter and table 4; Output data PLC.
A device database file contained a description of the properties of the PROFIBUS-PA device, e.g. the supported transmission rates, the type and format of the digital information output to the PLC. The bitmap files also belong to the gsd files.

These allowed the measuring point to be represented by an icon. The device database file and corresponding bitmaps were required by the network design tool of the PROFIBUS-DP network. By referring to figure 41 using the data exchange service, a PLC transmits its output data to a field device and read the input data from the response telegram. The output data was not evaluated by all devices. Analog input data contains analogue measured values; these are transmitted in 2 bytes to the PLC.

Table 4, summarise the measured values that was transmitted by flowmeter. The measured value was transmitted as an IEEE 754 floating point number calculation sample [26].

Table 4: Measured value as IEEE 754 floating point number

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
Sign	Exponent (E)								Fraction (F)						
	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}	2^{-7}
Fraction (F)															
2^{-8}	2^{-9}	2^{-10}	2^{-11}	2^{-12}	2^{-13}	2^{-14}	2^{-15}	2^{-16}	2^{-17}	2^{-18}	2^{-19}	2^{-20}	2^{-21}	2^{-22}	2^{-23}

The measured value was transmitted as an IEEE 754 floating point number calculation sample given as:

Measured value = (–1) X Expo Sign X 2Espo (E – 127) X (1 + F)

40 F0 00 00 hex = 0100 0000 1111 0000 0000 0000 0000 0000 binary

Value = (–1)0 x 2(129 – 127) x (1 + 2–1 + 2–2 +2–3)

= 1 x 22 x (1 + 0.5 + 0.25 + 0.125)

= 1 x 4 x 1.875 =

=7.5 measured value

Devices	DP Address	Number of input bytes	Number of output bytes
DP Flowmeter	111	1	1
ET200M-PA FLOWMETER	1	4	1
Drive 1	8	2	2
Drive 2	9	2	2
Level Transmitter	6	1	1
	Total	**10**	**7**

Figure 41: length of data in the PROFIBUS network

PROFIBUS DP according to IEC 61158 and PROFIBUS DP support many possibilities to implement data exchange between bus master and the connected slaves. The IEC (International Electro technical Commission) is a worldwide organization for standardization comprising all national electro technical committees (IEC National Committees) [27] and [28]. The objective of the IEC was to promote international co-operation on all questions concerning standardization in the electrical and electronic fields. IEC 61158 title or represent digital data communications for measurement and control. PROFIBUS DP master carry out data exchange, for that reason, field devices with PROFIBUS DP was connection optimally adapted to the respective automation task. Device descriptions (DSG) and bitmaps required by the master, in order that a device was recognised as a bus participant, as shown in 42. The device database files were downloaded during the commissioning of the communication system.

PROFIBUS DP slaves properties were specified by GSD files (device database files) that were located in the Hardware Catalog (Other Field Devices) folder and also in the (CP 313-2C as DP-Master) folder. Those DP Slaves were identified by specifying the GSD file or type file in the DP Slave properties. SD files were created as ASCII (American standard code for information interchange) files with an ASCII Editor by describing each feature of the field device with a standardized key word. When the system was powered up, the bus master sends a parameter assignment message to the configured slave. The data length in both directions, specified during configuring, and was monitored by the PLC as well as the slave at every data exchange. If there was a deviation, the data exchange was cancelled and a diagnostic message was issued.

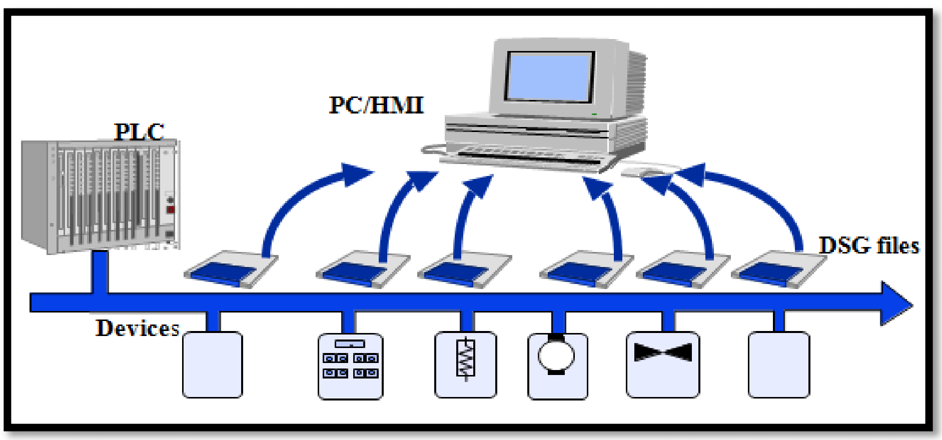


Figure 42: Generic Station Description File

After the configuration and process control phases, the HMI device was used to operate and monitor tasks in process plant and trend all measurements. The process plant screens on the HMI device was also used to provide a clear overview of the active processes and the process data was displayed on the HMI device and plant components were controlled by this HMI. The screens enable the user to observe, for example, operational states, current process data and faults in the process plant. The process displayed the process plant contain displays for values and messages which provide information about process statuses. Another requirement for process plant was that the HMI device be connected online to a Programmable logic controller, as shown in figure 43.

The HMI device provided operation feedback as soon as it detects that an operating element were selected. This operation feedback was independent of any communication with the PLC. Therefore, this operation feedback does not indicate whether the relevant action was actually executed or not. The update time was the sum of the acquisition cycle, the transmission time and the processing time. The tag address was entered according to the address structure of the PROFIBUS protocol. The tags are accessed through the selected connection listed in the (List of HMI Tags) area as shown in figure 44 and 45.

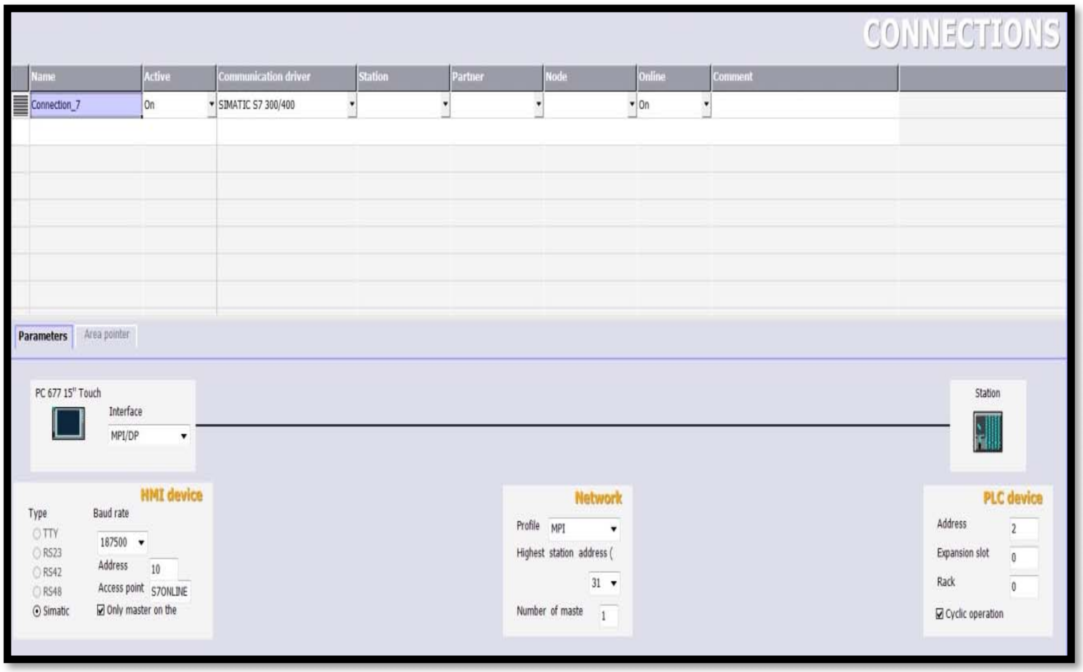
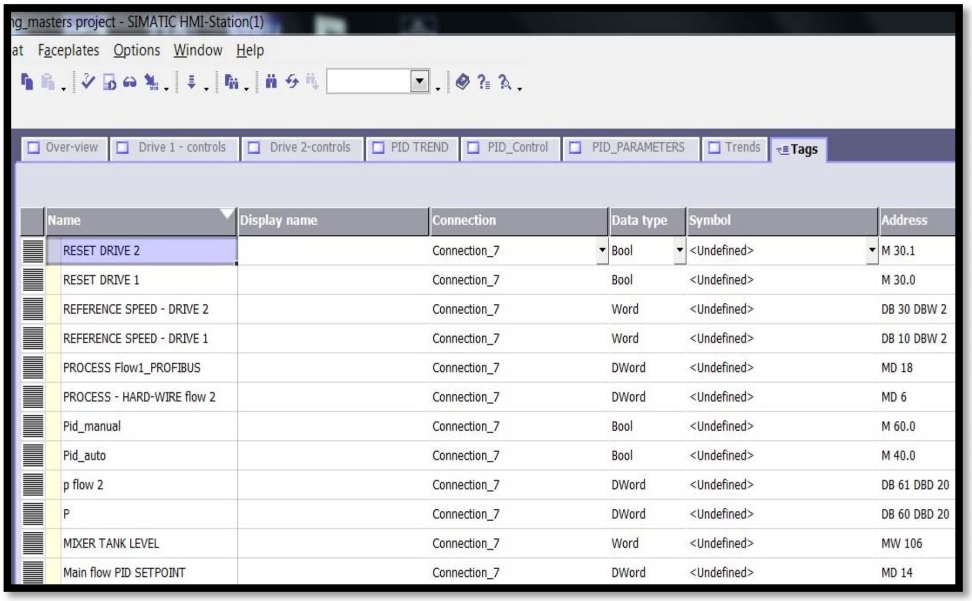


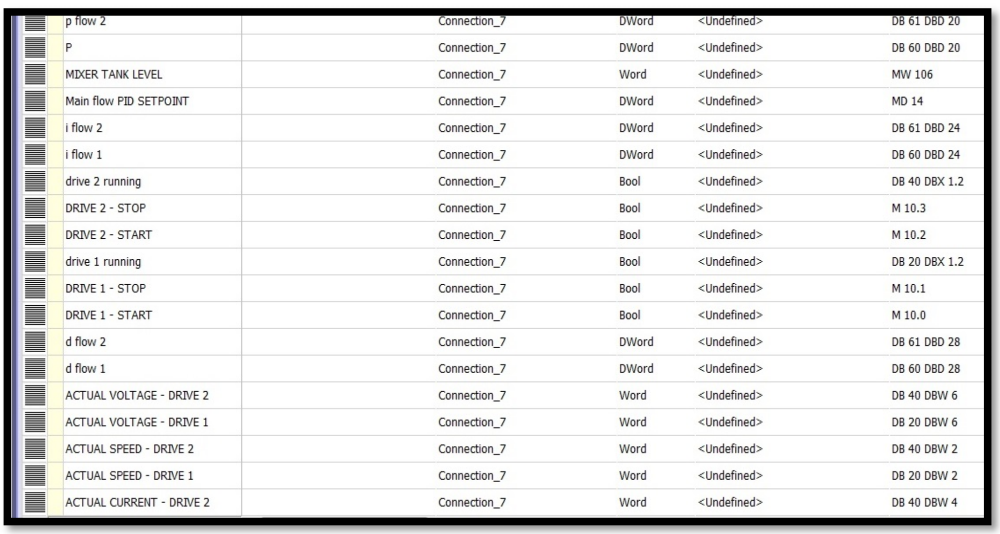
Figure 43: HMI and PLC online connection

WinCC software manages its tags centrally using so-called tag management. All data and tags created in the project and stored in the project database were captured and management in the runtime by WCC. Reading/writing tags via PROFIBUS, using the FDL protocol, was done using request and response telegrams. The request telegram was sent to the automation device from HMI. Programmable logic controller program was accessed through the interface with the communication calls. Tags with values supplied by the process were referred to as external tags in WinCC.



Name	Display name	Connection	Data type	Symbol	Address
RESET DRIVE 2		Connection_7	Bool	<Undefined>	M 30.1
RESET DRIVE 1		Connection_7	Bool	<Undefined>	M 30.0
REFERENCE SPEED - DRIVE 2		Connection_7	Word	<Undefined>	DB 30 DBW 2
REFERENCE SPEED - DRIVE 1		Connection_7	Word	<Undefined>	DB 10 DBW 2
PROCESS Flow1_PROFIBUS		Connection_7	DWord	<Undefined>	MD 18
PROCESS - HARD-WIRE flow 2		Connection_7	DWord	<Undefined>	MD 6
Pid_manual		Connection_7	Bool	<Undefined>	M 60.0
Pid_auto		Connection_7	Bool	<Undefined>	M 40.0
p flow 2		Connection_7	DWord	<Undefined>	DB 61 DBD 20
P		Connection_7	DWord	<Undefined>	DB 60 DBD 20
MIXER TANK LEVEL		Connection_7	Word	<Undefined>	MW 106
Main flow PID SETPOINT		Connection_7	DWord	<Undefined>	MD 14

Figure 44: HMI address list



p flow 2		Connection_7	DWord	<Undefined>	DB 61 DBD 20
P		Connection_7	DWord	<Undefined>	DB 60 DBD 20
MIXER TANK LEVEL		Connection_7	Word	<Undefined>	MW 106
Main flow PID SETPOINT		Connection_7	DWord	<Undefined>	MD 14
i flow 2		Connection_7	DWord	<Undefined>	DB 61 DBD 24
i flow 1		Connection_7	DWord	<Undefined>	DB 60 DBD 24
drive 2 running		Connection_7	Bool	<Undefined>	DB 40 DBX 1.2
DRIVE 2 - STOP		Connection_7	Bool	<Undefined>	M 10.3
DRIVE 2 - START		Connection_7	Bool	<Undefined>	M 10.2
drive 1 running		Connection_7	Bool	<Undefined>	DB 20 DBX 1.2
DRIVE 1 - STOP		Connection_7	Bool	<Undefined>	M 10.1
DRIVE 1 - START		Connection_7	Bool	<Undefined>	M 10.0
d flow 2		Connection_7	DWord	<Undefined>	DB 61 DBD 28
d flow 1		Connection_7	DWord	<Undefined>	DB 60 DBD 28
ACTUAL VOLTAGE - DRIVE 2		Connection_7	Word	<Undefined>	DB 40 DBW 6
ACTUAL VOLTAGE - DRIVE 1		Connection_7	Word	<Undefined>	DB 20 DBW 6
ACTUAL SPEED - DRIVE 2		Connection_7	Word	<Undefined>	DB 40 DBW 2
ACTUAL SPEED - DRIVE 1		Connection_7	Word	<Undefined>	DB 20 DBW 2
ACTUAL CURRENT - DRIVE 2		Connection_7	Word	<Undefined>	DB 40 DBW 4

Figure 45: HMI WCC Tags structure

In the case of process tags, tag management determines the communication driver by means of which WinCC was connected to the automation system and how the exchange of data took place. The associated tags are created in the directory structure of this communication driver as shown in figure 46. All tags assigned to its components in various displays and also trend display was created to view two flowmeter signals in order to analyse and compare both measurements.

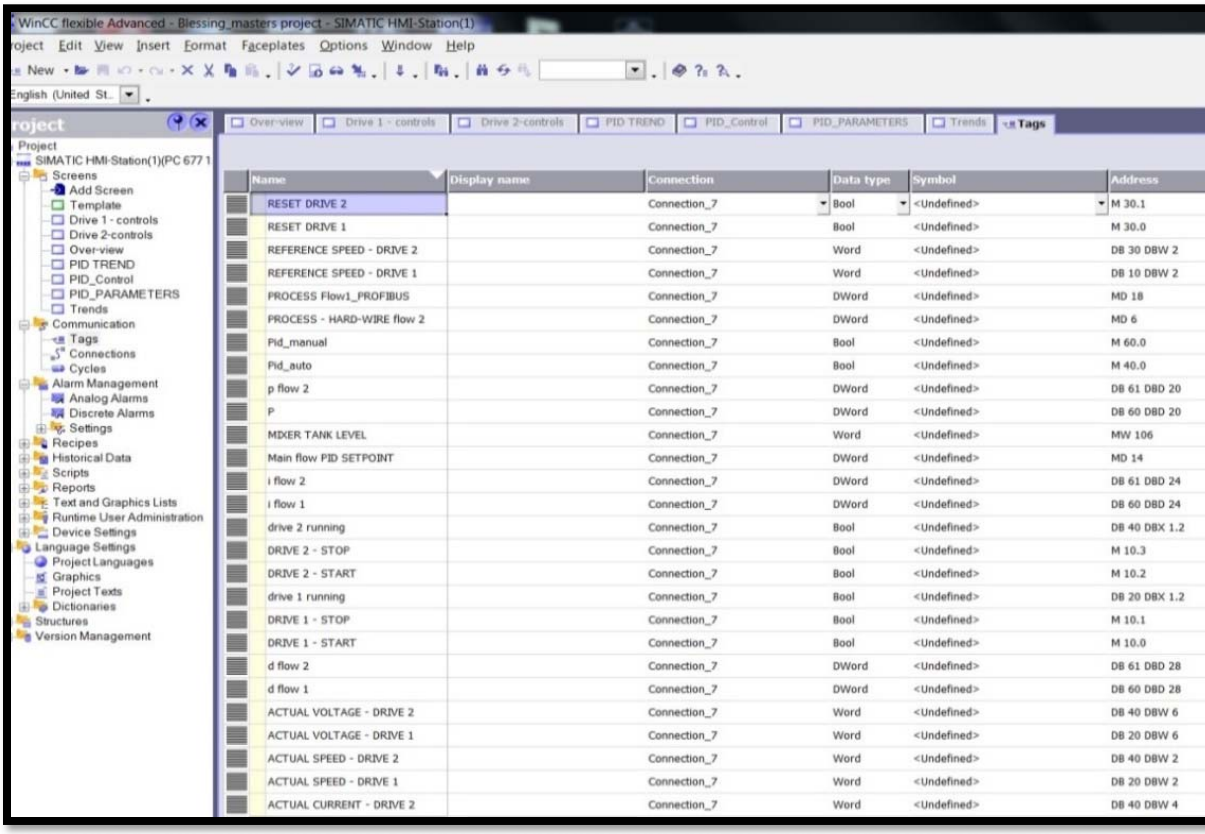


Figure 46: HMI address list

CHAPTER NINE: IMPLEMENTATION AND SET-UP PROCEDURES

9.1 Introduction

Simulation test was carried in order to verify the objectives of the research and find out if the prototype performed as per design. During simulation and testing stage, all technical aspects were analysed in various conditions before results were logged. All procedures, simulation tests were done to verify if the implementation (prototype) meets the design specifications and to check functionality of each component used in the process plant. In order to fulfil process plant design specification, important factors were noted like scaling, signal ranging and signal limits. Each component/ equipment used on this research was tested independently to detect if there was any problem in particular equipment before the entire process plant was fully practiced, the reason for that was to verify if all equipment's performed as expected in order to meet design requirement.

Another main reason to test all equipment' individually was to note functional status using required calibration procedures, methods , testing conditions and to reduce the risks of problems with the process plant during the operational process by means of intensive testing. Test activities including the used equipment were connected according to the graphic display as shown in figure 57 and were specified according compatible hardware. Software simulation was done for safety purposes and to reduce risk, detect an error early and remove. The quality of the software process plant was increased, while errors were found and corrected before the operational test.

9.2 Flowmeter calibration using Siemens fm verificator tool

Both DP Flowmeter and PA flowmeter was calibrated and verified using Siemens flowmeter (FM verificator tool). The SITRANS F M (Flow Meter) Verificator is an external tool designed to calibrate MAG 5000 and MAG 6000 sensors and to verify the entire product, installation and the application. The FM verificator checks the general operation conditions of the flowmeter and also check if the flowmeter was within required ranges. The SITRANS F M Verificator is highly advanced and carries out the complex verification and performance check of the entire flowmeter system, according to unique SIEMENS original principles [29]. The whole verification test was automated and there was no human error or influence.

The aim of calibration/verification was to validate that the calibration parameters and measuring range if were correctly set and that the calibration parameters were within limits. Both flowmeters, (MAG5000 and MAG 6000) were calibrated, verified and settings were made the same due to that both output signals were going to be used for results analyses in the research This process was dedicated reference measuring means, mainly applicable for diagnostics. The FM (flow meter) verificator tool checks the general operation conditions of the flowmeter and to check if the flowmeters were within specification range including the sensor's magnetic integrity. A fully automatic verification test took only 15 minutes after connection and consists of three steps:

(i) Transmitter test

(ii) Flowmeter insulation test

(iii) Sensor Magnetism test

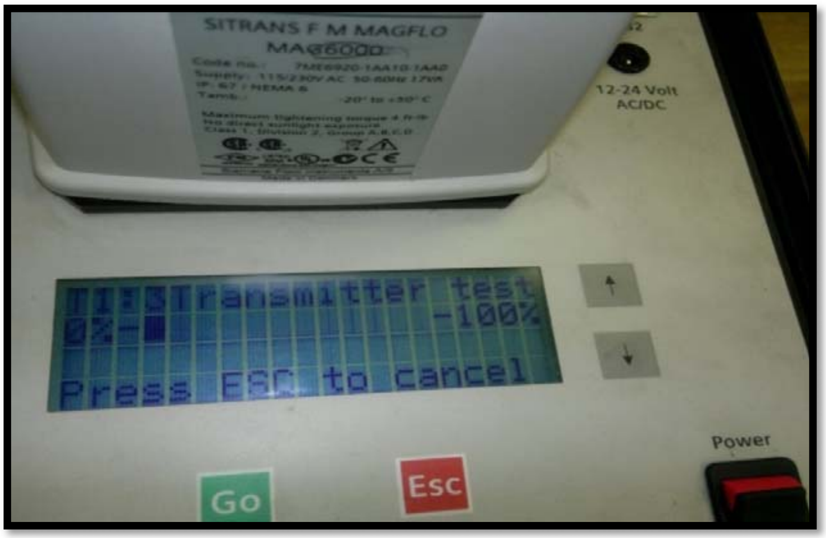


Figure 47: Transmitter Test range

Transmitter test; the transmitter verification checks the whole electronic system from signal input to output. Using a traceable calibrated precision network, the Verificator simulates flow signals to the transmitter input. Measuring the transmitter outputs and calculates its accuracy against defined factory values. Signal function from signal input to output signal processing – gain, offset, and linearity test of analog output. Figure 47 shows results obtained after transmitter test was successfully completed and ranges was (0-100%).

Table 5: Transmitter test results

Test	Results
Auto calibrating transmitter	4………..20ma
HMI range	0%...........100%

As seen in table 5, results obtained after 15 minutes of auto calibration was completed. These results shows that when transmitter was reading 4 ma and then HMI was display 0% for lower range and when transmitter was reading 20ma and then on the HMI it was display 100% for upper range. The reason for this test was to check if the transmitter was within acceptable range so that the author knows the range prevent transmitter from transferring values that are out of range and to generate incorrect signal. Transmitter insulation test; figure 48 showed verification test of the entire flowmeter and installation, which ensured that the flow signal generated in the sensor was not affected by any external influences. By generating dynamic disturbances close coupled to the flow signal, the flowmeter was tested for noise immunity to a maximum level. Other reasons for the flowmeter insulation test as follows:

(i) Influence on the flow signal.

(ii) Moisture in sensor, connection and terminal box.

(iii) Non-conductive deposit coating the electrodes within the sensor.

(iv) Missing or poor grounding, shielding and cable connection.

The tests carried out were very important to ensure that measurement generated by the sensors or measuring equipment's used on the research were set within acceptable ranges. These tests were done to improve the quality of signals and to obtain accurate results.

Figure 48: F M Verificator tool connected on the MG 5000/6000 sensor

Sensor magnetism test; the sensor magnetism test was done to ensure that the magnetic behaviour was unchanged. The current sensor magnetism was checked and determined during initial calibration and stored in the SENSOR PROM memory unit, as shown in 49.

(i) Changes in dynamic magnetic behaviour.

(ii) Magnetic influence inside and outside the sensor.

(iv) Missing or poor coil wire and cable connection.

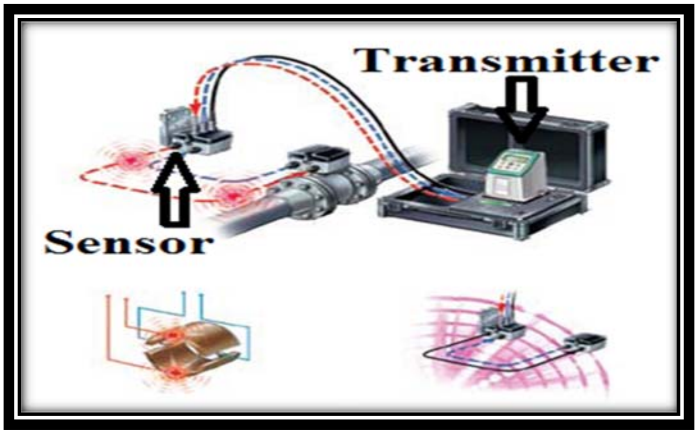


Figure 49: Sensor magnetism test

The verification process took only 15 minutes for fully calibration procedure. A verification report confirmed flowmeter performance was according to quality standard ISO 9001. The calibration of the flowmeters ranges was (0-100%), shown in figure 48, (0-16384), shown in figure 50 and linear scaled from (4….20ma). The result in figure 50 shows the linear scaling from the PLC (0-16384) and HMI (0-6.5l/m) displayed during operation test

The Both MAG 6000 and MAG 5000 flowmeters was measuring within (0-6.5l/m) because the minimum range was 0l/m and the maximum range was 6.5l/m after calibration as shown in figure 50. The flowmeter range was obtained by running both pumps at the maximum speed for 5 minutes and results indicated the set measuring range for both flowmeters was (0-6.5l/m) = (0-100%). Figure 50 and 51 shows the range of both flowmeters after there were scaled.

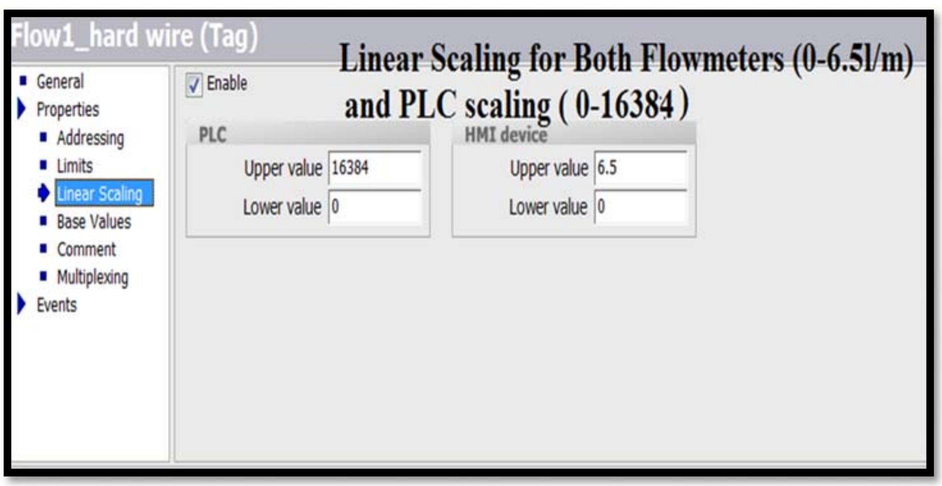


Figure 50: Flowmeters linear scaling

9.3 Experimental set-up

Preparing an experimental set-up correctly is very important in order to ensure that the right type of data and a sufficient sample size. The objective of experimental set-up was to improve the accuracy of the results in order to examine the research hypotheses. Experimental set-up refers to testing activity that examines component along with its design, generates component tests, identifies component faults and evaluates component reliability. Understanding how the relevant variables fit into the proposed solution and appropriate data was collected in a way that permits an objective analysis that leads to valid inferences with respect to the stated problem. This chapter introduces the reader to the setup used in the experimental set-up. The aim of this chapter was to evaluate each component used in this research and the important and unique aspects of the setup are described

Figure 51: Transmitter connection test

This was necessary to make comparisons and to determine whether any changes occur in the experimental set-up. Since it is always very hard to get an experimental procedure right the first time, experimental methods were repeated to ensure that the results were correct and accurate so that process plant performed as design. Discuss the proposed experimental strategy in detail along with its working and also discusses some suggestions that were helpful to improve the testing methods. There are many experimental strategies that were helpful in components based setting-up because these were useful to generate information about the components and help to evaluate the validity of the component for the process plant. This chapter also discuss the basic concepts of experimental set up, data collection, and data analysis. The following steps summarize the many decisions that need to be made at each stage of the planning process for the experiment and a brief explanation of each component, which helped to clarify the decisions that were made during each stage.

9.3.1 Micro-master 420 and 440 drives set-ups

The Micro-master 420 and 440 are a range of frequency inverters for controlling the speed of three/single phase AC motors. There is various models available range from the 120 W single phase input to the 11 kW three phase input. The inverters are microprocessor-controlled and use Insulated Gate BipoIar Transistor (IGBT) technology. This makes them reliable and versatile. A special pulse-width modulation method with selectable pulse frequency permits quiet motor operation [30] and [31]. There are five main important structures of the drives that were considered during experimental set-up:

(i) Control modes

(ii) System

(iii) Parameter structure

In this research Micro-master 420 and 440 drives were used control both pumps at the required speed set by the user; In this case both drives are controlling pumps to their desired set point according to system application. Both pumps are used to pump water or dissimilar products from different tanks to a mix tank (blend chest). All control modes and parameter settings of the Micro-master 420 and 440 are based on a V/Hz (voltage/frequency) control. Micro-master 420 and 440 parameters were changed and set using the advanced operator panel (AOP) to adjust the desired properties of the inverter, such as ramp times, minimum and maximum frequencies.

The parameter numbers selected and the setting of the parameter values were indicated on the optional five-digit LCD display. The Advanced operator panel (AOP) used to configure inverter for network operation; the following parameters were set as:

(i) P0700 = 6, used PROFIBUS control

(ii) P1000= 6, PROFIBUS reference both drives

(iii) P918 for drive 1 and P818, for Drive 2 = Unique number for each drive on the network.

These set up for the drive so that it communicates with PLC and HMI and to be able to start a motor. These settings were important for the drive to be recognised on the network. On confirmation of the parameter settings, only the editable parameters were copied to the new location - read only parameters were called from the original location,(P0700 = 6). AOP has been set as the I/O control (P0700= 5); to prevent unexpected drive operation, (P2014) was set to 5000 for removal of the AOP from the inverter in this mode and trip in 5 seconds. The AOP was set for (P2012) length to 4 when connected to an inverter. These parameter settings were used to restart the inverter automatically after an input power failure and motor parameters were accurately configured and to operate correctly. Pre-setting's of the inverter were compatible with the following motor data:

(i) Rated motor power

(ii) Motor voltage

(iii) Rated motor current

(iv) Rated motor frequency

The AOP used the status word to monitor and control connected inverters. Parameter P2016 and P2019 were changed from their default values and also an AOP used as the master control panel to prevent unexpected command source if as may occur. Once the inverters were set on the AOP to control speed using parameter (P700 = 5). The drive control buttons start, stop, jog and reverse control the drive at all times. The reason for setting drive parameters was to ensure that the motor was configured for the correct supply voltage: single / three-phase and both drives were able to communicate with the PLC via PROFIBUS network.

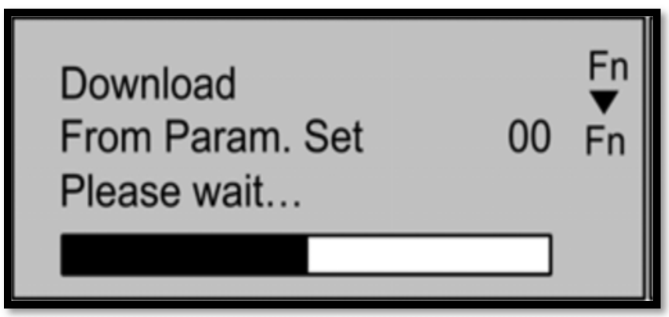


Figure 52: MM drive parameters

Micro-master 420 and 440 communications was successfully established and all parameters were downloading, as shown in 52. In the offline mode it was possible only to start and stop the both motors directly or simultaneously. Both drives were tested manually by pressing increase/decrease bottom and both pump1 and pump2 speeds was changing following drive command. Both drives were communicating using baud rate set at 187.5kbit/s for PROFIBUS network. Linear voltage/frequency for both drives was scaled as (0 - 10V= 0 - 50Hz) and the following conditions were met:

(i) Linear V/F motor speed controlled by an analog potentiometer.

(ii) Maximum speed (0-1481 rpm) at (0-50 Hz).

(iii) Controlled by a potentiometer via the analog inputs of the inverter.

(iv) Ramp acceleration time/ramp deceleration time = 10s.

Drives parameter settings were configured to prevent the failure of inappropriate operation or external frequency setting signal source from outputting frequency too high or too low and to prevent it being damaged. Parameter settings were also needed to restrict the rise rate of the frequency in acceleration during the electric motor start period in order to prevent over-current, and limit the frequency decrease rate in order to prevent over-voltage during deceleration period.

9.3.2 The pumps connection

Pump is one of the simplest pieces of equipment and its purpose is to convert energy of an electric motor into velocity or kinetic energy and then into pressure of a fluid that is being pumped. Liquid enters the pump suction and then the eye of the impeller. When the impeller rotates, it spins the liquid sitting in the cavities between the vanes outward and imparts centrifugal acceleration [32] and [33]. As the liquid leaves the eye of the impeller a low pressure area is created at the eye allowing more liquid to enter the pump inlet. In this proposed solution both pumps were used to pump water from tank 1 and tank 2 to the mix tank (blend chest).

The pumps connection is shown in figure 53. Both motors ranges were set at (0 - 1481rpm). Drive one and drives two were configured to control the speed of motor 1 and motor 2 to the desired speeds. In this research since two signals were compared so linear scaling had to be checked clearly and confirmed that the ranges were identical between two motors, two drives as follows:

(i) Drives range = (0-10V/0-50Hz)

(ii) Motors range = (0-1481rpm)

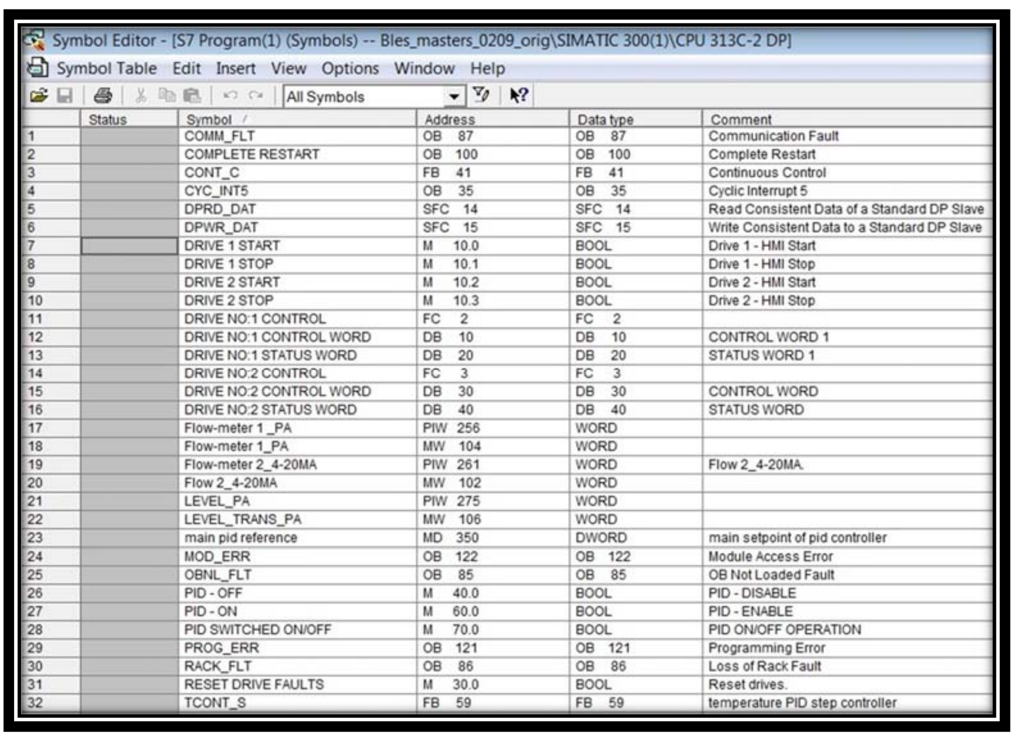
The ranges identical for both pumps to make sure that when test was conducted both pumps to run at the same speed at all times and to protect motors from overload and to obtain correct results.

Figure 53: Pump 1 and 2 connection

9.3.3 The PLC controller functional test

The PLC program was tested by entering the value manually on the program and signals statuses were viewed on the HMI page. Program test was done to ensure the program met design specifications; online monitoring/ forcing variables and HMI pages displayed the actual values. A function quick test via on- and off-line simulation was done to check if all nodes assigned on the network were communicating and addressing was accurate. Formal verification of the program was proved by checking whether a formal ideal of the system satisfies the requirement and the correctness.

Many authors have arrived to the conclusion that most of the failures in a system are due to incomplete and unambiguous requirements specification [34] and [35]. The program test was done to verify the correct operation safety of the process plant and to clear the description of the test cases. Figure 54 shows program symbol list used in the design and their comments. Input tests aimed to change an input seen at the entrance of a PLC and check that the modification has been perceived at the HMI. Output tests aimed to send commands to the process equipment, either from the HMI or from PLC. New states, new set-points and new actual values of the process equipment's were verified by being read as inputs and viewed on HMI. The reason of using formal methods was to try to increase the level of correctness before process plant was fully tested.


Symbol Editor - [S7 Program(1) (Symbols) -- Bles_masters_0209_orig\SIMATIC 300(1)\CPU 313C-2 DP]

Symbol Table Edit Insert View Options Window Help

All Symbols

	Status	Symbol /	Address	Data type	Comment
1		COMM_FLT	OB 87	OB 87	Communication Fault
2		COMPLETE RESTART	OB 100	OB 100	Complete Restart
3		CONT_C	FB 41	FB 41	Continuous Control
4		CYC_INT5	OB 35	OB 35	Cyclic Interrupt 5
5		DPRD_DAT	SFC 14	SFC 14	Read Consistent Data of a Standard DP Slave
6		DPWR_DAT	SFC 15	SFC 15	Write Consistent Data to a Standard DP Slave
7		DRIVE 1 START	M 10.0	BOOL	Drive 1 - HMI Start
8		DRIVE 1 STOP	M 10.1	BOOL	Drive 1 - HMI Stop
9		DRIVE 2 START	M 10.2	BOOL	Drive 2 - HMI Start
10		DRIVE 2 STOP	M 10.3	BOOL	Drive 2 - HMI Stop
11		DRIVE NO:1 CONTROL	FC 2	FC 2	
12		DRIVE NO:1 CONTROL WORD	DB 10	DB 10	CONTROL WORD 1
13		DRIVE NO:1 STATUS WORD	DB 20	DB 20	STATUS WORD 1
14		DRIVE NO:2 CONTROL	FC 3	FC 3	
15		DRIVE NO:2 CONTROL WORD	DB 30	DB 30	CONTROL WORD
16		DRIVE NO:2 STATUS WORD	DB 40	DB 40	STATUS WORD
17		Flow-meter 1 _PA	PIW 256	WORD	
18		Flow-meter 1_PA	MW 104	WORD	
19		Flow-meter 2_4-20MA	PIW 261	WORD	Flow 2_4-20MA.
20		Flow 2_4-20MA	MW 102	WORD	
21		LEVEL_PA	PIW 275	WORD	
22		LEVEL_TRANS_PA	MW 106	WORD	
23		main pid reference	MD 350	DWORD	main setpoint of pid controller
24		MOD_ERR	OB 122	OB 122	Module Access Error
25		OBNL_FLT	OB 85	OB 85	OB Not Loaded Fault
26		PID - OFF	M 40.0	BOOL	PID - DISABLE
27		PID - ON	M 60.0	BOOL	PID - ENABLE
28		PID SWITCHED ON/OFF	M 70.0	BOOL	PID ON/OFF OPERATION
29		PROG_ERR	OB 121	OB 121	Programming Error
30		RACK_FLT	OB 86	OB 86	Loss of Rack Fault
31		RESET DRIVE FAULTS	M 30.0	BOOL	Reset drives.
32		TCONT_S	FB 59	FB 59	temperature PID step controller

Figure 54: S7 program symbol list

PLC was communicated correctly and all devices were manipulated and their reaction was observed on the human machine interface. All test instructions were performed and tests results were corresponding.

9.3.4 Human machine interface (HMI)

The HMI - an adequate cable diameter was provided when setting up PROFIBUS DP network and MPI network. PLC connections was established through the internal MPI-/PROFIBUS DP interface only when all the modules in a subnet have unique addresses and when the actual structure matches the network configuration that has been created. DP master (PLC) and DP slave in the PROFIBUS network were assigned a unique and PROFIBUS addresses ranges from 0 -125 as shown in figure 55.

The PLC was used in this research to monitors inputs, makes decisions based on its program, and controls outputs to automate a process plant. Protocol drivers were binding the communication interface to the protocol, as shown in figure 55. HMI was required to display all inputs, output values and entire process plant so that inputs/outputs variables were monitored as shown in figure 56. The objective of using HMI on this process plant was to allow effective operation and the control of the process plant from the human end, whilst the plant simultaneously feeds back information that aids the author to decision making process and to analyse achieved results.

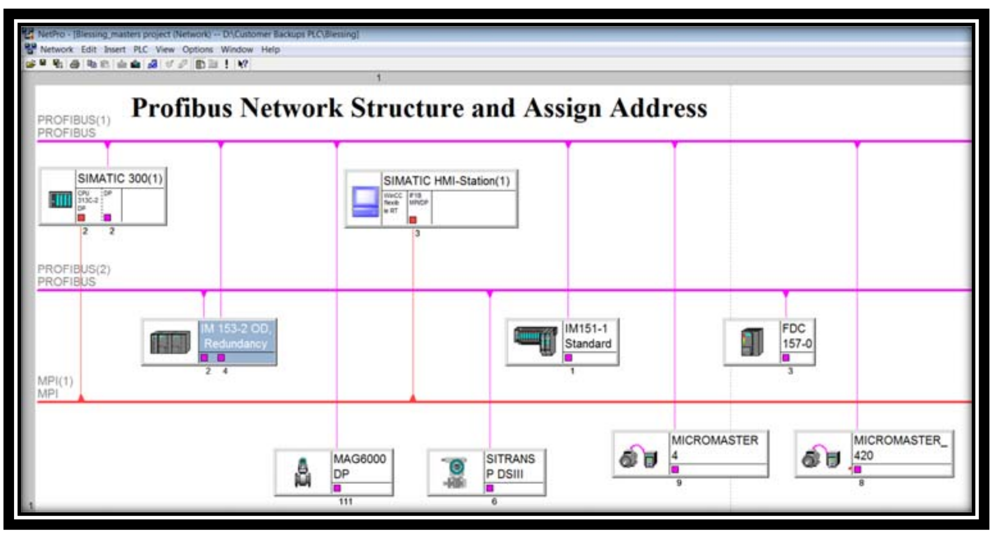


Figure 55: PROFIBUS DP network structure and assigned address range

The requirements for data exchanges between a PLC and an HMI station are basically: a physical connection that was available to both, such as a RS-485 connection, along with the adequate ports on both [36] and [37]. A common protocol called PROFIBUS was selected as a physical connection for Programmable logic control and human machine interface. Once the HMI program was develop, each screen created the graphic interfaces that reflect and send commands to the PLC program through the configured protocol. Every input/output field, alarm display and status icon generated was linked to a tag, interconnected to an address of a register in the PLC. When the output value changed in the PLC, the HMI received the value to be displayed in the output field that was generated in the active field. The HMI prepared a list of tags it needs to write to the PLC tag addresses, and values it has to read in the CPU memory every time it becomes active (when it has control of the communication link), using the tag addresses configured in both PLC and HMI programs to connect the two as shown in 57.

The HMI displays were tested by entering a value (1400rpm) on the PLC program and same value displayed on the screen (1400rpm). The reason for this test was to make sure if scaling was correct to all variables and to test if PLC program was matching with HMI tags. Communication from HMI to the output devices was tested by entering a set point on the screen and responses or measurements coming through from devise shown in figure 58. All graphics pages/displays on the HMI were tested by calling them differently; all input/output values required for that particular screen were requested from the PLC, the latest values were available and displayed on the screen as shown in 58. Aim of output testing was to test the correct wiring and routing of all the commands towards PLC and HMI or actuators (i.e. ET200M I/O cards or Drives).

Figure 56: HMI and MPI network addresses

Figure 57: PLC and MPI network addresses

9.3.5 PROFIBUS settings and addressing procedures

PROFIBUS DP is a network made up of two types of devices connected to the bus: master devices and slave devices. It is a bi-directional network, meaning that one device, a master, sends a request to slaves, and the slave's responds to that request [38]. Thus, the bus contention was not a problem in this research because only one master (PLC) controlled the bus at all times, and all slave devices must respond immediately to a request from the master. Since a request from a PLC to slave devices were heard by all devices attached to the bus, some mechanism were occurred for a slave device to recognized that a message was designated for it and then responded to the sender. This means that field module will recognise if the message send by the PLC contain information for the device connected on it.

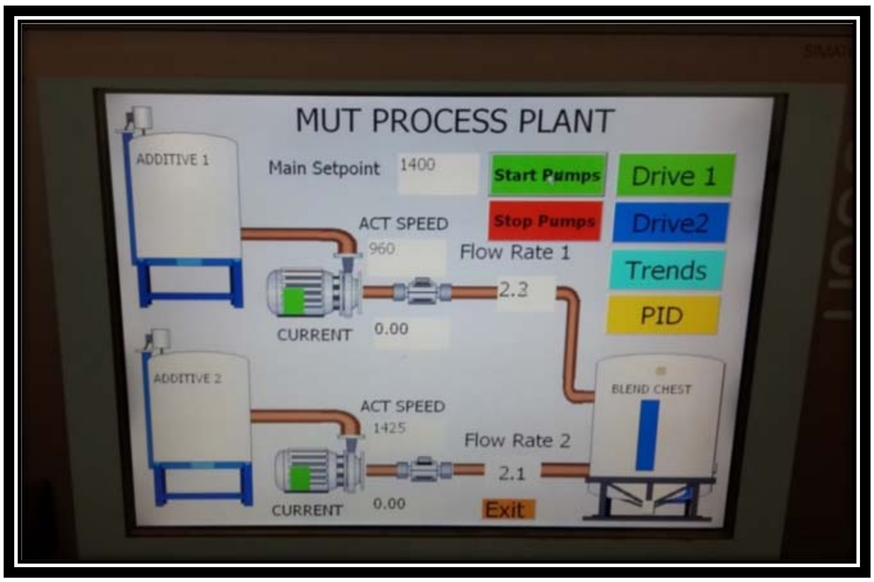


Figure 58: HMI graphic display page

All devices on a PROFIBUS network were assigned a unique address so that were recognised by the PLC. Figure 55 shows all devices on the network with assigned addresses. Each bus node received a PROFIBUS address so that were identified by unique address on PROFIBUS-DP network. The PROFIBUS address was set separately for both PROFIBUS DP networks directly on the DIL switches of the DP/PA Coupler. Below are steps that were followed during PROFIBUS settings and addressing procedures as shown in 59:

(i) Valid PROFIBUS address range: 1 to 125.

(ii) All PROFIBUS addresses assigned were within the DP master system.

(iii) Set the ADDR DIL switch was left (ON) position. Result: The PROFIBUS address was fetched from the non–volatile memory area in the DP/PA Coupler.

(iv) Switch on the 24 V DC power supply to the DP/PA Coupler. Result: The LEDs ON1 and/or ON2 are lit.

(v) Connect the PG/PC via PG cable to the DP network and Open STEP 7(hardware) HW Config and configure the DP/DP Coupler and in (hardware) HW Config, select the configured DP/DP Coupler.

(vi) Go to PLC > PROFIBUS > Assign PROFIBUS address. Result: The (PROFIBUS address) view appears.

(vii) Enter the configured PROFIBUS address and confirm with (OK). Result: The PROFIBUS address was transferred to the non–volatile memory of the DP/PA Coupler.

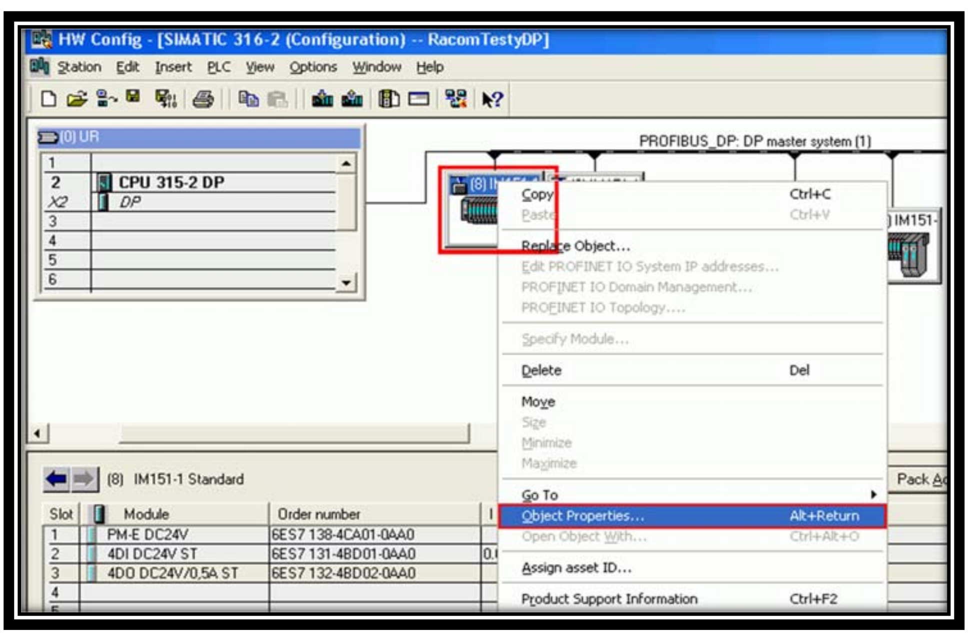


Figure 59: Assign PROFIBUS address

9.3.6 Calibration procedure for HMI touch

Human machine interface was used to operate and monitor tasks in process plant and display measurements. The process plant components were controlled by this HMI and screens on the HMI device were used to provide a clear overview of the active processes and process trend. The hardware of the HMI has an operator control called Touch screen. The Touch objects are basically operated in the same way as mechanical keys but touch-sensitivity had to be calibrated and configured so that the HMI provides an operation feedback as soon as it detects that an operating element has been selected, as shown in 66. The HMI touch screen step by step calibration and configuration procedure shown as:

(i) (Touch Ware) symbol on the desktop

(ii) Start menu (Start), command (Settings > Control panel > HMI Touch Touchscreen)

(iii) Start the touch ware settings and switch to the (Touch settings) tab control.

(iv) The following dialog will appear: touch screen while making settings in touch ware or if the screen saver was active, the SIMATIC process visualization software, (Pro Tool/Pro), will carry out the functions which happen to be behind it.

(v) Adjust the touch ware as desired. Switch to the (Tools) tab control and click on the (Options) button.

(vi) The (MHI Touchscreen Options) dialog will appear. Click on the (Advanced) button.

(vii) The following dialog will appear; Click on (2 point).

(viii) Activate the control box in the (Software settings) and lick (OK) to close all open dialogs. Switch to the (Calibrate) tab control.

(ix) The following dialog will appear; Click on the (Calibrate) button and the dialog for touch calibration will appear.

(x) Touch the cross hairs on the touch screen for at least 3 seconds and no longer than 10 Seconds. The (Calibration complete) dialog will appear and exit the calibration by clicking on the (Done) button.

(xi) The reason for calibrating touch screen was to prevent incorrect operations and undefined states of the application program and to make sure that no malfunctions were triggered in the application program as shown in figure 60.

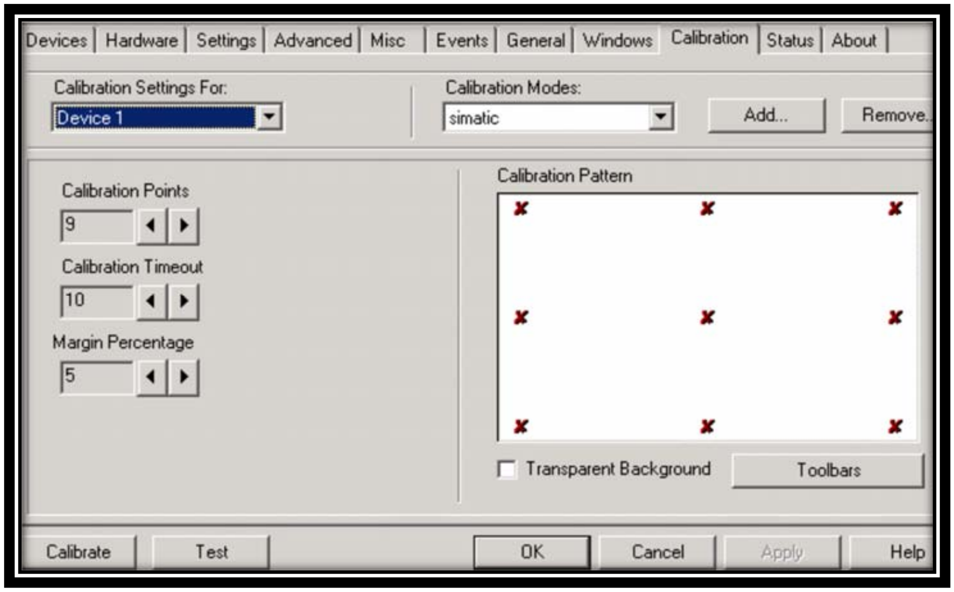


Figure 60: Touch Calibration screen [15]

9.3.7 Procedure to load dsg files to the subdirectory

Device database files (dsg) is a database file contains a description of the properties of the PROFIBUS devices [39] and [40]. This file completely describes the PROFIBUS functionality of the device, baud rates supported, input (basic of PROFIBUS operation), output data configurations, and start up parameter. The supported transmission rate (187.5kb/s) and the type and format of the digital information output to the PLC. The device database file and corresponding bitmaps were required by the network design tool of the PROFIBUS-DP network.

Before the dsg files were loaded, the configuration database was generated so that all device files were stored into the PLC. Database files were downloaded into the PLC via a serial port and files were loaded into a flash memory card of the PLC. A selected slave device was assigned the PROFIBUS address, identifies the I/O to be exchanged, and selected the appropriate parameters for the desired operation of the device. Downloading steps were repeated until all selected slave devices and the entire bus configuration was described or completed. Figure 61 shows the functionality of a PROFIBUS configuration tool.

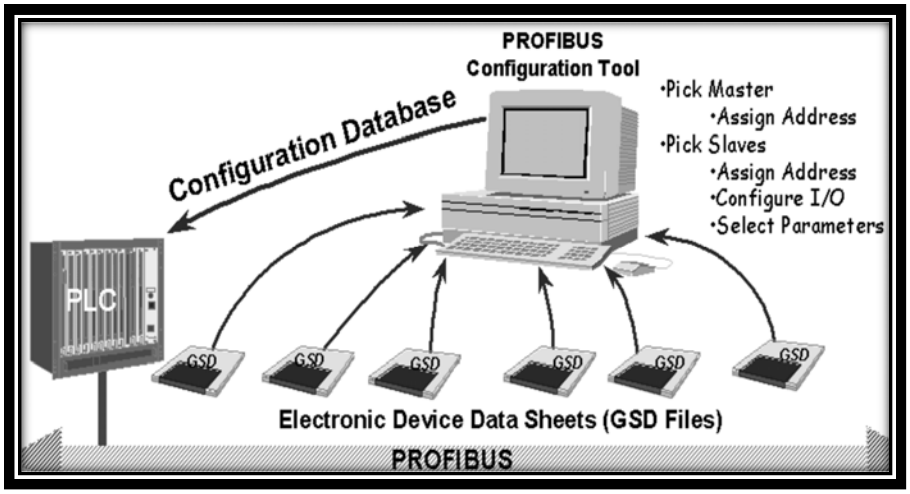


Figure 61: PROFIBUS configuration tool

After download, the PLC had the information necessary to start up all the devices in its configuration and this information was stored in retentive memory. After the bus system was started up, the normal communication between a PLC and each of its assigned slaves was to exchange I/O data. The PLC sends output data to slave devices in its configuration. The addressed slaves immediately respond with its input data. The baud rate of 187.5kbit/s was suitable for bus network and data exchange cycles was normally repeated many times during a single control-logic scan. Below are the reasons why the process GSD (device description) files were downloaded into the PLC:

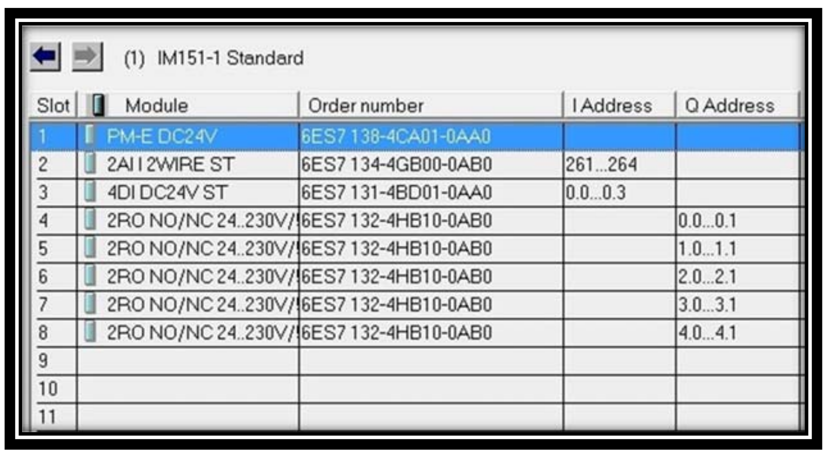
(i) To maintain a hardware catalog of devices configured on the bus.
(i) Allow the PROFIBUS device address to be specified and all selected slaves to be recognised by the PLC.
(ii) Allowed the specification of the input and output data to be transferred between PLC and its slaves
(iii) Allowed start up parameters to be selected in order to activate specific operating modes.
(iv) Allowed selection of the system baud rate and generated the database file that was used by the PLC.
(v) Performed validity checks so that the configured data was structured logically correct.

Every device was allocated an identity code by the PROFIBUS User Organisation (PNO). This appeared in the device data base file name (gsd) shown in table 14. The information contained in the dsg file and the full set of device data base files for Siemens devices were obtained as in table 6.

Table 6: Device data base file

Ident Number: (M)	0x00A2
Identifies the device type of the DP device.	187.5_supp: (G)
The DP device supports the transmission rate	187.5 kBaud.
Type: Unsigned16	Boolean (1: True)
MaxTsdr	187.5 kbit/s: (G) (Value= 60)
Time base	Bit time
GSD-File	ET 200M 8AI-2 DP SIEMENS AG
MLFB:	6ES7 141-1BF01-0XB0 <order number
Version:	18.05.98 SX
File	SI__803D.GSG

On SIMATIC S7 PA field devices are integrated via their GSD files as standard slaves in COM PROFIBUS (V3.1) [41],[42] and [43]. GSD file were copied into the IM 153-2 of the COM PROFIBUS directory (COMPBxx\GSD). After all files were downloaded, COM PROFIBUS files were selected and read in GSD file menu command. Result obtained after start up, shows the slaves of ET200M (IM 151-1) displayed in the hardware catalog. Figure 62 shows the ET200 I/O module and devices connected. In this case PA flow meter (hard-wired) was connected in slot 2 and its module (2AI 2WIRE ST) and the analog input bus address was (261….264).


(1) IM151-1 Standard

Slot	Module	Order number	I Address	Q Address
1	PM-E DC24V	6ES7 138-4CA01-0AA0		
2	2AI I 2WIRE ST	6ES7 134-4GB00-0AB0	261...264	
3	4DI DC24V ST	6ES7 131-4BD01-0AA0	0.0...0.3	
4	2RO NO/NC 24..230V/	6ES7 132-4HB10-0AB0		0.0...0.1
5	2RO NO/NC 24..230V/	6ES7 132-4HB10-0AB0		1.0...1.1
6	2RO NO/NC 24..230V/	6ES7 132-4HB10-0AB0		2.0...2.1
7	2RO NO/NC 24..230V/	6ES7 132-4HB10-0AB0		3.0...3.1
8	2RO NO/NC 24..230V/	6ES7 132-4HB10-0AB0		4.0...4.1
9				
10				
11				

Figure 62: ET200M (IM 151) - hardware catalog

9.3.8 Simulation tests using PROFIBUS tester

The communication status of each DP slave was very important for the proper functionality on the PROFIBUS network and intermittently failing DP slave's decreases the network throughput and lead to a complete system or plant breakdown [44]. In this research all devices connected on the PROFIBUS network had to be verified, checked for errors, short circuit and detect the current baud rate so that design PROFIBUS network meet design specification. The reason for a bus test was to automatically detect the baud rate or open circuit voltage immediately after connection to the PROFIBUS and provides much additional information on executing, analysing, and managing bus tests.

The PROFIBUS testers 4 tools was used for analysis and to allow a complete PROFIBUS network check, starting with the physical measurement of the PROFIBUS line up to detailed protocol analysis. There were test procedures that were defined for master and slave's devices on the protocol. These tests consist of self-test, isolation test, hardware test, and functional test, conformity and interoperability tests. All tests used a uniform testing procedure and each test was explained below. These tests were done to avoid poor process plant functionality and DP communication during final test. PROFIBUS DP nodes' RS 485 interfaces with the PROFIBUS Tester 4. Listings of addressable slaves connected on the bus were detected, even PLC on the PROFIBUS DP.

(i) Self-test: These checks were done to ensure that all components are functioning properly. The boot sequence was interrupted immediately and all error messages that occur were displayed on the screen [44].

(ii) Isolation test: Signal protection against disturbance and bad installation during commission.

(iii) Hardware test: examined the electronics of the PROFIBUS interface. The interface was checked for compliance with the RS 485 specification. The electrical characteristics (e.g. terminating resistors, bus interface and line level) were tested. In addition, dsg file were checked against the parameters of the device, as shown in 63.

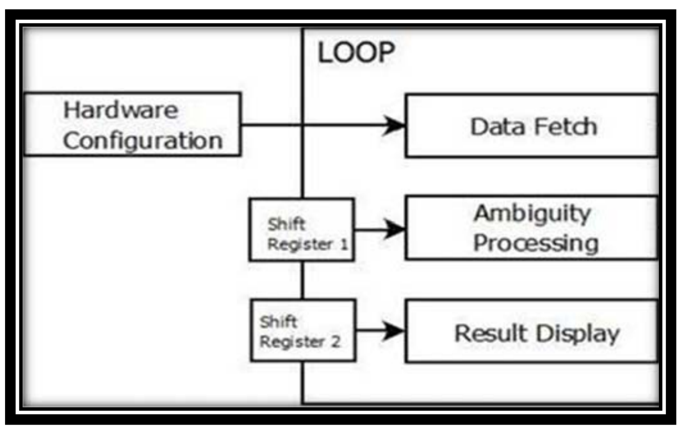


Figure 63: hardware test [21]

(iv) The function test: examined the bus access and transmission protocol, as well as the functionality of the devices. The dsg files used for parameterize and were test. This procedure required knowledge of the internal structure of the implementation. The reactions generated by the device under test was monitored on the bus and recorded via the bus monitor. The outputs of the devices were monitored.

(v) The conformity test: was the main part of the tests. Procedure implementation was checked for conformity with the PROFIBUS standard. The desired behaviour was combined to form test sequences which was adapted to the device under test. The token passing procedure ensured that the bus access right (the token) was assigned to the master within a precisely defined timeframe (less than 5mins). The token message, a special telegram for passing the token from one master was within a (configurable) maximum token rotation time. In PROFIBUS the token passing procedure was only used for communication between Master and all nodes on the network.

Figure 64: PROFIBUS tester 4 results.

This permitted advance function testing of single bus segment, which reduced troubleshooting times. During functional test, all faults were localized and rectified. The test results were displayed on the screen tester displays shown in table 7. PROFIBUS communication results are shown in figure 64. This tool was used to check if there the PROFIBUS cable and to test quality of the signal.

Table 7: Simulation results

Test voltage range	±4.89V
Sampling rate	192 samples/s
Resolution	40 mV
Signal details	1600 sampled points
Oscillogram analysis	1192 sampled points
Baud range	187.5kb/s
Input data length	10 bytes
Output data length	7 bytes
PD slaves on the Network	5 slaves

This was very important because two signals were going to be compared and the author was minimising risk of capturing incorrect samples. As seen in table 14A that voltage on the PROFIBUS cable was +/-4.89V, speed of data called baud rate was 187.5kb/s as expected. The amount of input/out data to be carried by the cable was 17 bytes in total on the table including the number DP slaves connected on the PROFIBUS network was 5 as expected.

CHAPTER TEN: SIMULATION TEST AND TEST RESULTS

10.1 Introduction

Generally, a simulation test is about validating various conditions of test procedures and accuracy of the proposed solution. It was very important for this research that the scope of the simulation test method was clearly defined, and any phases included in the range were to be accurate and repeatable through validation. The main reason of using these simulation test methods was to try to increase the level of correctness since two signals were going to be analysed based on many aspects. Using these testing approaches, it was possible to test all the combinations and problems were resolved by using formal verification and procedures. The bugs were detected before the process plant was implemented and to create a formal model of the real system and check its correctness against formal requirements describing the expected performance of the process plant. Simulation tests and experimental test were divided according to their level of standard as follows:

(i) Components testing: where each component used in the process plant was tested separately. Reason for this test was to quickly identify if all components that were within design specification and to check if all ranges were acceptable.

(ii) Combination testing: where all individual components were put together to be checked altogether. Main aim for this test was to verify if all components were communicating with the programmable logic control and to check if all devices had correct assigned addresses on the HMI including a communication between PLC and HMI.

(iii) Process plant testing: where the complete process plant was tested as a whole application. This was the main and important part of the thesis and it was done because all captured results were analysed base the proposed solution and be clear explained based on figures and table forms. It was also done to verify functionality of the prototype.

10.2 Experimental tests

Testing was applied to process control applications as a black box test of function blocks or modules in combination with input simulations and visualization of the results by the HMI trend page [44]. The HMI had been configured in a compliant way with I/O standards and it was tested in a way that there was no hidden data and all the HMI functions used. The normal coding procedure for serial interfaces used was called Universal Asynchronous Receiver Transmitter (UART). UART coding telegram has a number of UART characters; every bit was defined with a static signal level. Start = 0; Parity = EVEN; Stop = 1

With EVEN parity, the number of (1) values in data was rounded up to an even number with the parity. Therefore, in order to transmit 8 bits of data, for example **one** byte was =11 bits that always be sent. Individual telegrams begin with a SYN interval for a request telegram and a min T_{SDR} interval for a Response telegram. The following rules of transmission were applied:

(i) The quiescent state on the line corresponds to the logical (1) level.

(ii) Before each request telegram, a quiescent time should be maintained of at least 33 bits (SYN).

(iii) Between the individual characters in a telegram no quiescent times are allowed.

(iv) The receiver checks per character: start bit, stop bit and parity bit.

An identifier for a compact format module carries up to 51 bytes or 51 words. One word was made up of 16 bits and the DP master (PLC) processed data as a whole and it was the task of the DP-PLC to ensure the required consistency [45]. Table 8 shows the compact format module coding: Are 0x11 = 2 byte inputs - consistency per byte. The compact format was always described with one byte. Byte coding for the compact format as follows:

Table 8: Compact format

7	6	5	4	3	2	1	0	Configuration: compact format
		0	1					Input
		1	0					Output
		1	1					Input & Output
				0-15 0-48				Length of data 00 = 1 byte or word 15 = 16 bytes or words 48= 51 bytes or word
	1							Word(s) of 16 bits
	0							Bytes of 8 bits
1								Consistency across the entire module
0								Consistency across one byte or one word

The PLC cyclically reads the input information from the slaves and cyclically writes the output information to the slaves at 244 bytes (total data of telegram, request and respond). The two equivalent length bytes in the telegram header of the variable format contain the number of information bytes in the body of the telegram (input 10 bytes and output 7 bytes). The byte range was from 4 to 244 transferred to the FDL (*Field bus Data Link*) of a telegram as shown In table 9, but in this research total of 17 bytes were transferred to enable the class 1 master (PLC) to communicate efficiently with the slaves, it is known that 10 bytes the PLC sent to the slave as outputs and 7 bytes was received as inputs. In this study the total number of input bytes was 10 bytes and the output bytes are 7 bytes and equal to 17 bytes shown in table 8.

Table 9: Total number of the telegram

7	6	5	4	3	2	1	0	LE and LEr: length byte
1 to 17 bytes								Length byte

As a rule, the bus cycle time in a PLC was response time relatively small. In a PROFIBUS-DP network it was possible to differentiate between three different cycle times:

(i) PLC application cycle was the time the DP master (PLC) processes its control program in a specific cycle time. Before each cycle, input data was read and after each cycle, new output data was available.

(ii) Slave application cycle was the time used by the DP slave to take the output data, processes its device functions and makes input data available within a defined cycle time.

(iii) Bus cycle was the time PROFIBUS copies output data cyclically from DP master to DP slave and input data from DP slave to DP master.

The cycle's time of the PLC, PROFIBUS and the DP slave were not coupled to each other. In order for a system to function, the cycle time of the DP slave was shorter than the cycle time of the PROFIBUS which, in turn, shorter than the cycle time of the PLC [46]. The cycle time of the PROFIBUS-DP depends between the two other cycle times. Before we focused on analysing the DP and PA transmission time, we needed to calculate the total bus cycle time and calculate a PLC cycle time. To simplify the estimation of PROFIBUS-DP bus cycle time, findings set out in the calculation below, as shown in figure 65.

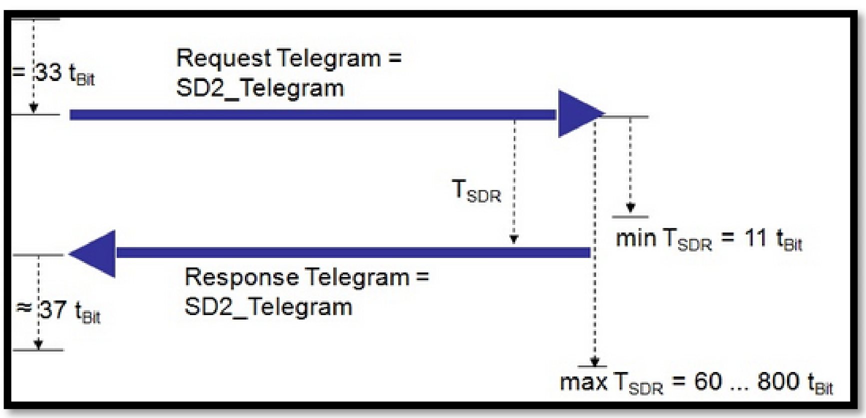


Figure 65: Bus cycle time frame

Response time is the interval expiring between the time data from a DP-PLC are written to the process image of outputs unit and the time until these can be read in the process image of the second. In this research the total response time or bus cycle time was 5ms. Response time was calculated using PROFIBUS standard formulas and calculation steps are shown below. The time cycle of the DP/PA Coupler was independent of the currently set data transfer rate and the following applies: The DPA/PA update time was calculated using PROFIBUS standard formula. This was done to ensure that time taken during data transmission was known since one signal was transmitted via DP/PA coupler. Before the response time was calculated, the list of certain system information was compulsory as listed and the importance of using the profibus standards formulae was to estimate and to understand variable required to perform the proposed solution and based on the following criteria:

(i) CPU 313-2 DP
(ii) Bus cycle time = 5 ms
(iii) DP/PA coupler (1ms)
(iv) Data length (n) 1 ... 17 bytes = (10 input and 7 output)
(iv) Data transfer rate of 187.5kb/s

DP/PA update time = 1 ms + n x 7s = 1ms+5x7s = 1.053ms =1ms [47], the results obtain was the time taken to transfer data between PD/PA coupler from the hardwired flowmeter. This was done because two signals were being compared, so hard wired signal was transmitted via DP/PA coupler to the PLC, whereas the other flowmeter signal was transmitted direct to the PLC. The time taken had to be calculated and it was required for calculation of total response time. The time taken to transmit 17 bytes of 5 devices in a bus cycle had to be calculated and verified. Table 10 shows the details of total bus data. The standard PROFIBUS network formulas were used to calculate the bus cycle time given following elements [48]:

(i) Number of DP slaves= 5 devices
(ii) Transmission rate= 187.5kbit/s
(iii) Data volume =(17 bytes)

Define the cycle time for a DP message cycle below:

(i) TID1 = (Idle Time) until the master is ready for the next message cycle
(ii) TID1 ≈ 37 tBit (for masters based on ASPC2 ASICs)
(iii) TMessage cycle = TSYN + 2 x SD1_telegram + typeTSDR + TID1
(iv) TMessage cycle = 33 + 2x (bytes x 11 + 99) + 32 + 37
(v) TMessage cycle ≈ 300 + bytes x 11 tBit
(vi) tBit = duration of one bit, corresponds to the inverse of the PROFIBUS bit rate.
(vii) Bytes = number of bytes for all inputs and all outputs added together.
(viii) Slaves = number of DP slave stations in the network.

The bus cycle time abbreviation and meanings:

(i) TBCycle= total bus cycle, TToken= total token, TGAP= total gap, Tmessage= Total message to be send, tBit= duration of one bit transmitted.
(ii) TBCycle = (TToken + TGAP + Slave x TMessage cycle) tBit
(iii) T_{BCycle} = (317 + slave x 300 + bytes x 11)/bit rate + 75 µs

(iv) Slave = 5, bytes = 17, bit rate = 187.5 kBit/s i.e. tBit = 0, 08337 μs

We obtain an estimated bus cycle time of PROFIBUS DP:

TBCycle = (317 + 5 x 300 + 17 x 11) / 0.8337 + 75 = 4917 μs ≈ 4.9 ms= 5ms

Table 10: Detail of DP devices

Devices	DP Address	Number of input bytes	Number of output bytes
DP Flowmeter	111	1	1
ET200M	1	4	1
Drive 1	8	2	2
Drive 2	9	2	2
Level Transmitter	6	1	1
	Total	**10**	**7**

Since PROFIBUS PA was using different baud rate, the PA cycle time was calculated as shown below. The formula was further simplified for PROFIBUS-PA and data was transferred at a fixed bit rate of 31.25 kbit/s and the process value (PV) 5 bytes was transferred. The time required to transfer such a process value therefore lies approximately around (transmission DP/PA link 1.1 ms (= 5bytes x8 bits) [48].

For a bit rate = 187.5 kbit/s the following applies:

$T_{BCycle} \approx 21.3 + (14.5 * \text{slaves}) + (1.1 * \text{PV})$ ms

$= 21.3 + (14.5*1 \text{ slave}) + (1.1*561)$ ms

$= 6.7$ ms

According to the estimated calculations it was evident that PROFIBUS-DP was faster that a PROFIBUS-PA due to that cycling time for PROFIBUS-DP was 5ms and for PROFIBUS PA was 6.7ms and difference between two networks was +/- 1.8ms. These estimation formulae are relatively accurate in practical measurements. After all calibration tests were completed, process plant actual performance was analysed and compared with the desired performance. The results are shown in figure 66 in a bus cycle time of 5ms, that the PLC can read/write 17 bytes in one cycle. It was also proven that 5 DP devices with a total length of 17 bytes were transferred within 5ms.

The PLC sends output data to a slave device in its configuration. The addressed slave immediately responds with its input data. This cyclic (repeated) I/O data exchange takes place asynchronously to the control logic scan and was repeated as quickly as possible. Data exchange took place every cycle for every slave in a master configuration. At the most commonly used baud rate of 187.5kbits/s, data exchange cycles are normally repeated many times during a single control logic scan, as shown in figure 67.

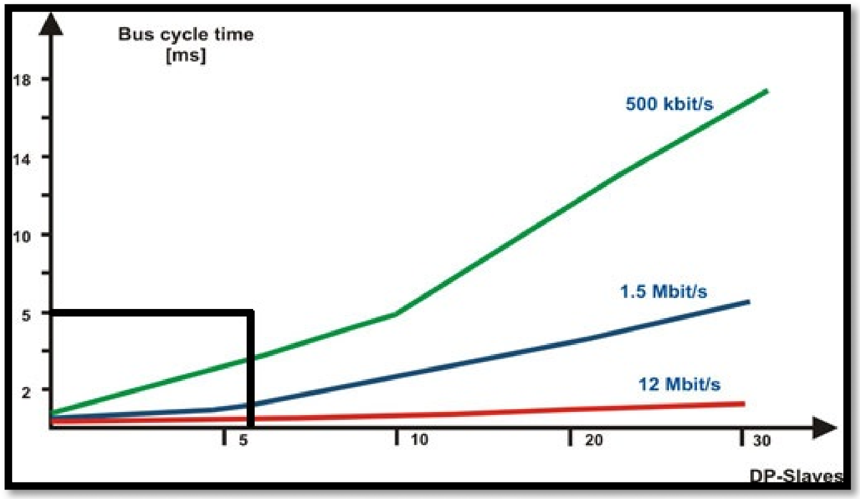


Figure 66: PROFIBUS DP transmission time

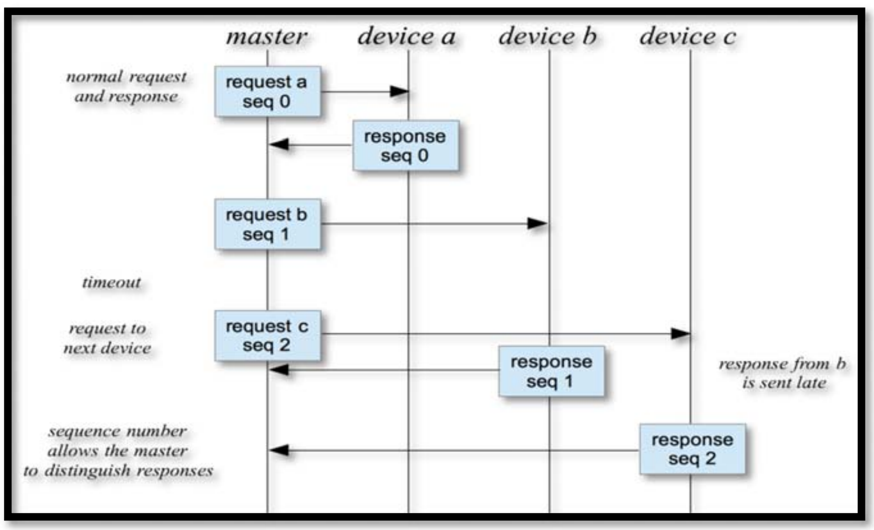


Figure 67: Single control logic scan

The master (PLC) was given control of the bus for a short time (5ms), and during this time, it exchanges I/O data with each of its assigned slaves. It then passed control to the master (PLC) on the bus, via a short message called a (token), and that master exchanges I/O data with each of its slaves. The PLC was holding the token that allowed initiating communication to its slaves. The reason for the process trend page (referring in figure 68) was to capture the process plant response during practical tests and analyse whether the response meets the design requirements and the results of trend names or tags are shown in figure 68.

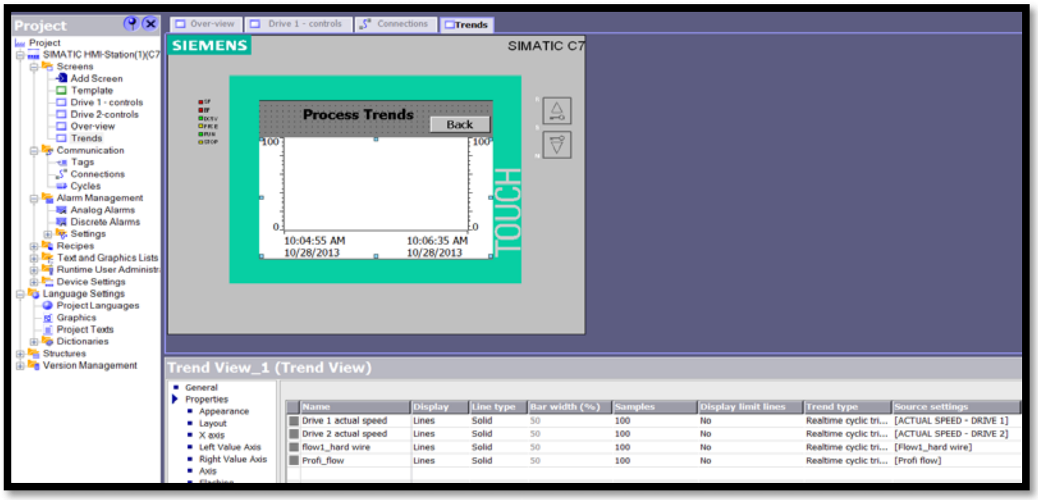


Figure 68: Process Plant trend view

Process plant components used in the research were tested for linear scaling, reliability, generalizability and validity as stated above. The reason for performing these tests was to produce consistent results and to checks how similar results were, if the research was repeated under similar circumstances. Each component has its own valid linear scaling and sets range. Detailed analyses of the test result were done to determine if the process plant performed as per the design and to evaluate if all measurements meet the expectations and were and measurements were very consistent.

Table 11: linear scaling and value range

Device Name	Range and linear scaling	Units
Flowmeter 1	0-6.5 l/m	L/m
Flowmeter 2	0-6.5 l/m	L/m
Drive 1	1-10V; 0-50 Hz	Hz
Drive 2	1-10V; 0-50 Hz	Hz
Pump1	0-1481rpm	rpm
Pump 2	0-1481rpm	rpm
PLC	0-6384	Double word
HMI	0-100%	Percentage

All equipment's values were valid, reliable and the results are shown on the figure 69. The linear scaling and the value range results are shown in table 11. These results were obtained when all components were calibrated and tested and table below, shows linear scaling for components that were used on the research. The relation between table 11 and figure 69 was that both shows the same range but in figure 69 all devices names and their ranges, whereas table 10 shows results in a table form.

After the HMI graphics design were completed, the tests set was containing both the requirements based test cases were done. The reason for the tests was to be confident that it was clearly understood how it was performing prior to it being implemented on the hardware. The performance of the hardware was never checked before because this research was not once attempted and the system was never configured before. The individual component test was done to make sure that all devices were within acceptable range and if there was a device that did not meet specification it was going to be noticed and its offset or error be included when capturing results to avoid misinterpreting the results. This was to note if there was component that did not meet requires and be clear marked before implementation and to prevent capturing erratic results. All the HMI pages were interacted with the process plant enabling rapid test case generation to ensure the design meets the requirements and the design intent with the inputs/outputs mapped to the program in the PLC, during simulation, pressing the buttons on the HMI caused the states to execute the appropriate transition for each button press.

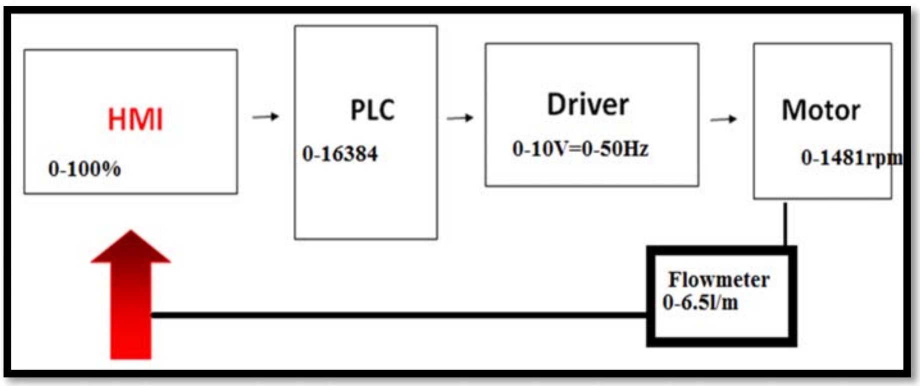


Figure 69: linear scaling of the process plant

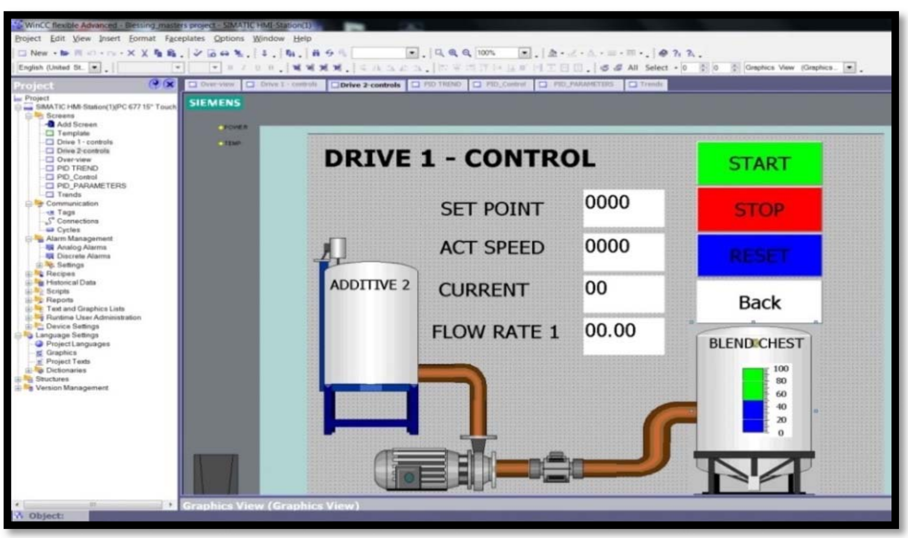


Figure 70: Drive 1 control display

Technically, this was enough functionality to exercise the model and perform interactive testing of the model while visualizing state transitions in State flow. The reason for testing the HMI displays was to identify the inputs and outputs of the process plant that the HMI was controlling and displayed. Another reason for testing all pages was to ensure that it was fully functioning as expected and meeting the design requirements. All the HMI display pages were created and tested independently as shown in figure 70, 71 and 72 by entering a value on the speed set point that was within the limits and press the start button, pumps started running and current values were displayed according to their units or displayed on their blocks. Two control loops were created, tested and able to switch either back to main menu by pressing back button as shown in figure 72.

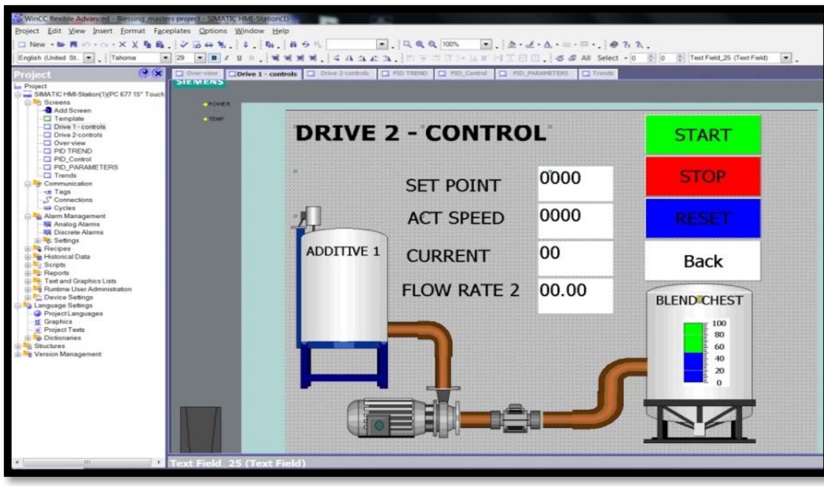


Figure 71: Drive 2 control display

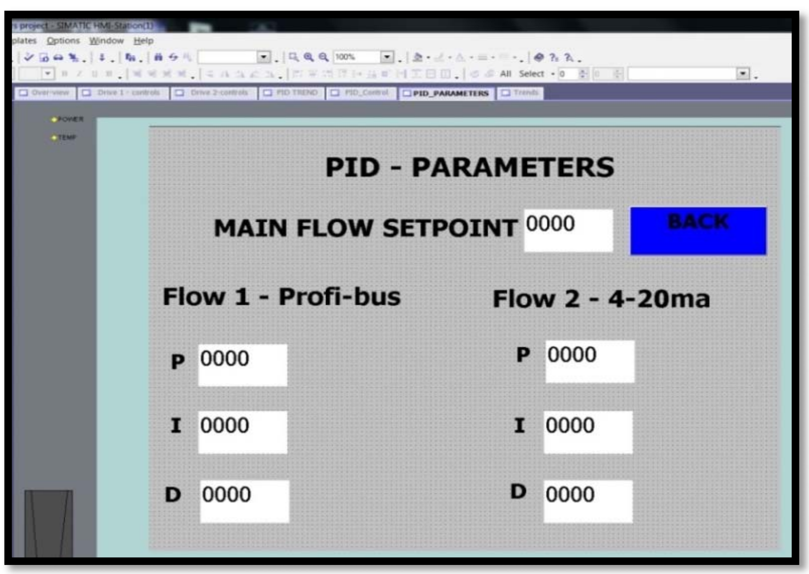


Figure 72: PID control display

The process plant design and operation of all equipment's used met design specification. Since the objective of the research was to analyze communication of PROFIBUS using process automation decentralized periphery against conventional (4-20mA).

The tests performed are shown in figure 73 as follows:

flow rate 1 represent the DP flow meter and flow rate 2 represent PA flow meter); the first test was conducted by entering a speed set point of 1400 rpm on the main set point and pressed (start pumps button) to run both pump 1 and 2 at the same speed. As displayed, the actual speed for pump 1 has not yet reached the set point but the DP flow meter was already reading 1.7 l/m as indicated. A speed of pump 2 was 1421 rpm but flow rate 2 still shows no measurement. The test result 1 shown that PA flow meter data transmission time was slower compare to DP flow meter. Table 12 shows the practical results.

Table 12: Test and results 1

Test item	flow rate 1 representing the DP flow meter	flow rate 2 representing the PA flow meter
Speed set point 1400 rpm	773	1421
flow rate	1.7 l/m	0 l/m

As seen in table 12 that actual speed of pump 1 was 773rpm and pump 2 was 1421 but flow rate 1(DP flowmeter) was already measuring 1.7 l/m and flow rate two (conventional flowmeter) was reading 0.0 l/m. In detail when pump 2 was running at 1421rpm, there was flow or water passing the flowmeter but because of data transmission delay it showed that there was still no measurement. It was caused by the time delay from the sensor to the field interface module to the PLC.

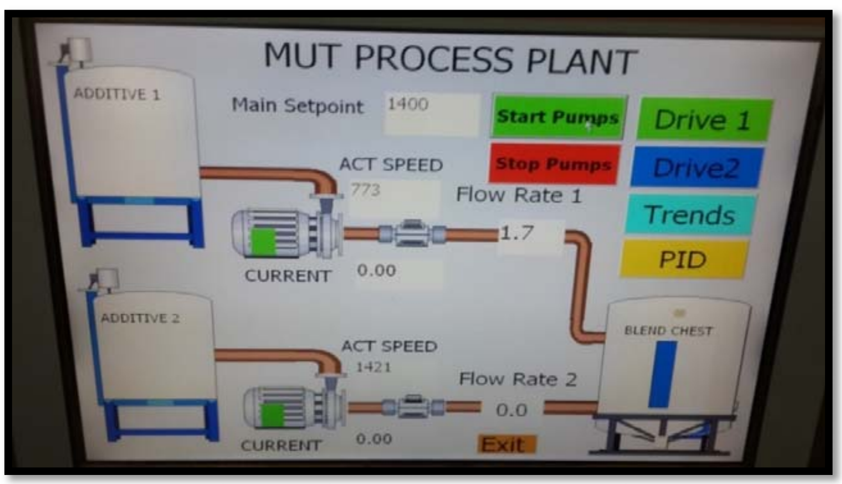


Figure 73: Result 1 (Process plant practical test)

In figure 74, the actual speed for pump 1 was 1423 and for pump 2 was 1425. This means that both pumps have reached the desired set point of (1400 rpm) and are running at the same speed. When both signals were compared, the second test result revealed that flow rate 1was reading (0.2 l/m) higher than flow rate 2 as seen in table 13. This clearly showed that DP flow meter detect flow and transmit its generated signal faster than conventional flow meter (PA flowmeter). This was also caused by that the DP flowmeter transmit 2 bytes for input and output whereas the PA flowmeter was transmitting 5 bytes in total, 3 bytes for inputs and 2 bytes for output. In short PA flowmeter was transmitting 3 bytes more that DP flowmeter as the same baud rate of 187.5kb/s. This proved that in profibus network the longer amount data to be transferred cannot be compared than a lower data at the same speed.

Table 13: Test and results 2

Test item	flow rate 1 representing the DP flow meter	flow rate 2 representing the PA flow meter
Speed set point 1400 rpm	1423	1425
flow rate	4.4 l/m	4.2 l/m

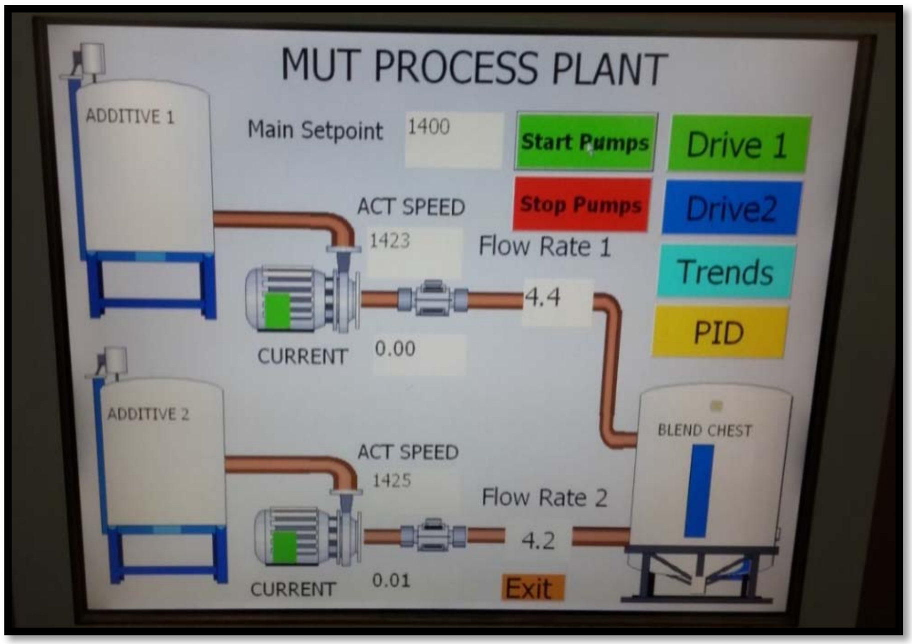


Figure 74: Result 2 (flowmeters update time)

The trend page in figure 75, shows that during testing, PA flowmeter signal was updating slower that PD flowmeter and PA also showed a delay update due to its slow data transmissions. Red trend represent main set point speed, black trend line represent DP flowmeter and green trend line represent PA flow meter. Further tests were done to verify if the results achieved were accurate by decreasing main set point from 1400 to 100 rpm

Result 4 achieved shown in figure 76, it was confirmed that the DP flow meter value (black trend) started dropping when there was a change in the speed set point but PA flowmeter value (green trend) took few milliseconds before a change in flow was detected. PA flowmeter updating value was slower than a DP flow meter. It was also tried by adding a delay time on program for DP flow line and entered a set point 1400rpm.

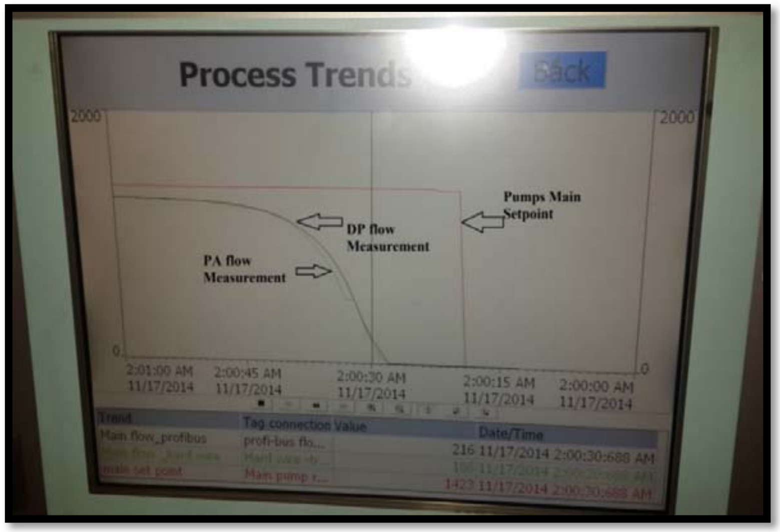


Figure 75: Result 3 (Process Trends)

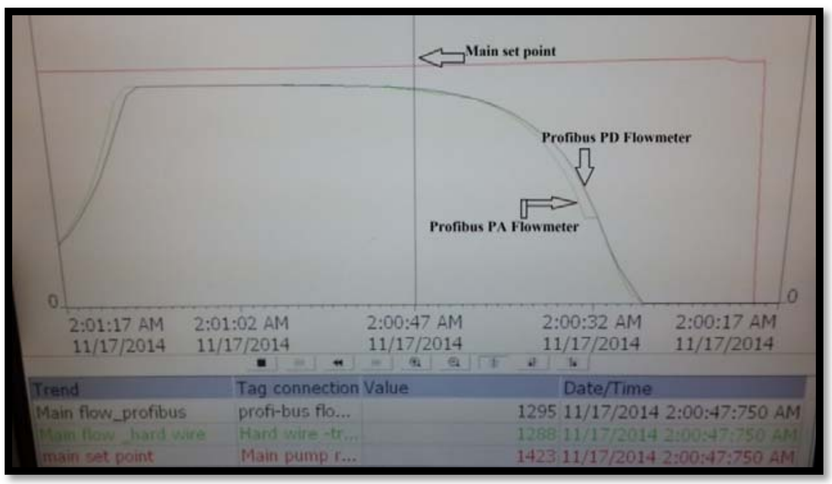


Figure 76: Result 4 (Flowmeters measurements)

As shown in figure 77, after a (start pumps button) was pressed, pump 2 started running after few milliseconds compare to pump 1. The reason for putting a delay timer was to check if PA flow meter will start to measure before the DP flow meter. Results obtained indicated that flow rate 1 started displaying some values before the PA flowmeter even if the timer was put on the PD flow line to delay pump 2 not starting simultaneously with pump 1. The result 4 proved that the DP flow meter was updating faster than PA flow meter.

Last test was conducted by entering a speed set point value of 1400 rpm. The result 6 as shown in figure 78 that flow rate 1 was reading faster that flow rate 2. Both pumps were configured to show running status as soon as pump start. The reason was to view the status of each pump. This was done to avoid getting incorrect results for example if pump stop due to a fault or other reason, we could able to clear the error and start a test again. As shown in figure 78, each pump shows running status by green status on it. Detailed analysis of the test results concluded that the prototype performed as designed because all equipment's were tested independently and successfully results were achieved.

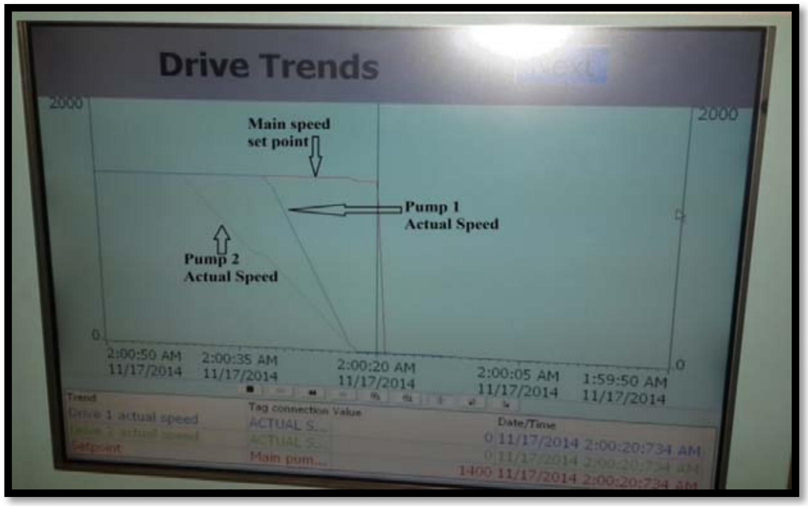


Figure 77: Result 5 (Drive Trends)

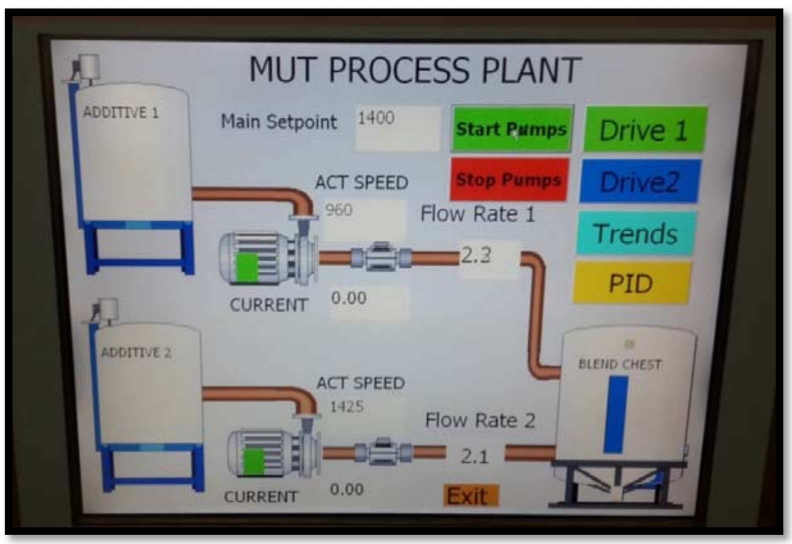


Figure 78: Result 6 (Process Plant measured values)

Referring to figure 78, further analysis showed that when a set point of 1400rpm was entered, the actual speed for pump 1 was captured at 960rpm whereas the actual speed for pump 2 was taken at 1425rpm. The actual flow rate 1 for DP flowmeter was reading 2.3 l/m and actual flow rate 2 for PA flowmeter (conventional Flowmeter) was reading lower by 2.1 l/m. Even though pump 1 was running at lower speed than pump 2, but DP flowmeter was reading high that PA flowmeter due to that DP flowmeter was communicating direct with the PLC and because DP flowmeter was connected on the PROFIBUS network. PA flowmeter was reading a bit lower that DP flowmeter because time taken for PLC to reading its data was delayed due to the signal passes to field module interface (ET200M) and to DP/PA coupler where time was taken to convert one language to another before data was read by the PLC. As mentioned earlier that PA devices do not have a privilege to communicate with a PLC direct and DP/PA coupler was used to interface the PA flowmeter data at the PROFIBUS network level.

The entire process plant tests results proved and verified that it met the design requirements and all kinds of inputs responded correctly to perform its functions within an acceptable time. The range of all procedures was completed successful to fulfill the design specification. Results validated that DP flowmeter was updating faster that a PA flowmeter and results in table14. This chapter also detailed the observed process of testing and its results recording. After performing the data analysis, it was revealed that the proposed solution met design requirement.

Table 14: Captured results after tests

Test	Test description	Test Results		
		DP flowmeter results	Hardwired (PA) flowmeter results	DP Flowmeter and PA Flowmeter response variances
Test 1	Comparison of two proposed signals and capture real results	1.7 l/m	0.0 l/m	1.7 l/m
Test 2	Comparison of two proposed signals and capture real results	4.4 l/m	4.2 l/m	0.2 l/m
Test 3	Comparison of two proposed signals and capture real results	0.8 l/m	0.2 l/m	0.6 l/m
Test 4	Comparison of two proposed signals and capture real results	5.2 l/m	4.4 l/m	1.2 l/m
Test 5	Comparison of two proposed signals and capture real results	4.8 l/m	4.1 l/m	0.7 l/m
Test 6	Comparison of two proposed signals and capture real results	6.2 l/m	5.6 l/m	1.4 l/m
Test 7	Comparison of two proposed signals and capture real results	1.4l /m	0.0 l/m	1.4 l/m
Test 8	Comparison of two proposed signals and capture real results	2.2 l/m	2.1 l/m	0.1 l/m
Test 9	Comparison of two proposed signals and capture real results	4.3 l/m	4.0 l/m	0.3 l/m
Test 10	Comparison of two proposed signals and capture real results	3.6 l/m	3.2 l/m	0.4 l/m
Test 11	Comparison of two proposed signals and capture real results	4.6 l/m	4.4 l/m	0.2 l/m

Figure 79 shows the difference between the DP flowmeter and PA flowmeter based on captured results for each tests. The main reason for generating this graph was too focused on analysis of the data collected during the experiment tests and compared two signals and conclude based on facts. As shown in figure 79, the process plant captured results from test 1 to test 11 shows that when both pumps were ran at the same speed and by entering different speed set points on the pumps main set point from 0 to1481rpm. Results obtained reveals that from test 1, conventional (hardwired) flow meter was taking longer to read similar value as the DP flowmeter hence both bumps were running as the same speed.

The test results show that from test 2 and test 3 conventional flowmeter was taking few milliseconds to measure the same reading as the DP flowmeter because when the pump main set point was changed from high to low, it was possible to detect the flow change and conventional flowmeter was able sense.

This was because conventional (PA flow meter) flowmeter was already measuring flow and it was easy to sense the flow variation.

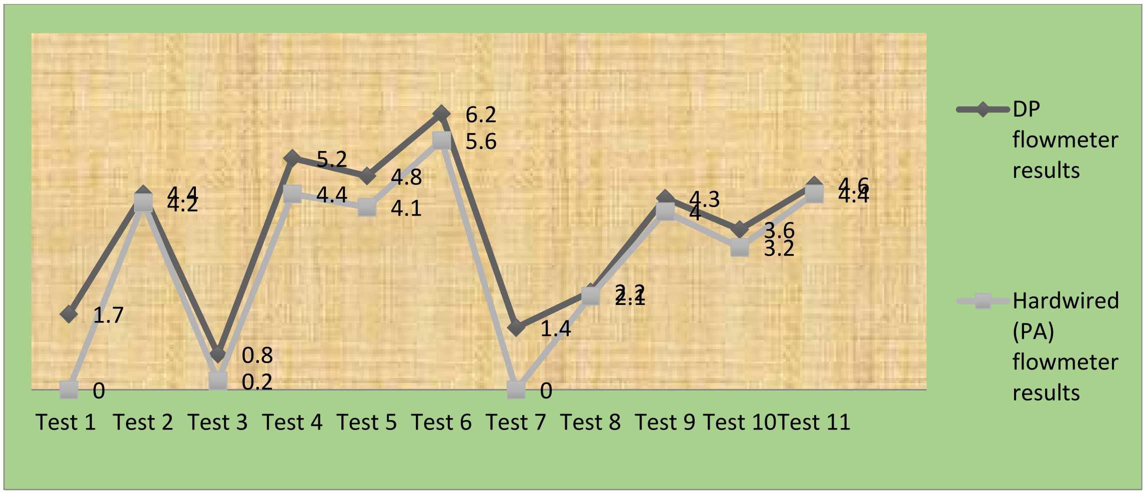


Figure 79: Process plant captured results

Table 15: Captured results of MM440 and MM420 drive

Test	Test description	Test result		
		Main Drive set point	MM440 Drive1 Actual speeds	MM420 Drive 2 Actual speeds
Test 1	Performing real- time test and capturing real results	1400rpm	773rpm	1331rpm
Test 2	Performing comparison of real results	1400rpm	1388rpm	1398rpm
Test 3	Performing real- time test and capturing real results	100rpm	98rpm	98.2rpm
Test 4	Performing comparison of real results	1000rpm	998rpm	997rpm
Test 5	Run both pumps at the same speed	1200rpm	1201rpm	1200rpm
Test 6	Performing real- time test and capturing real results	1400rpm	1398rpm	1390rpm
Test 7	Performing comparison of real results	1000rpm	459rpm	845rpm
Test 8	Performing real- time test and capturing real results	1400rpm	960rpm	1425rpm
Test 9	Performing comparison of real results	1400rpm	1423rpm	1424rpm
Test 10	Run both pumps at the same speed	1400rpm	1097rpm	1423rpm
Test 11	Run both pumps at the same speed	1200rpm	1240rpm	1246rpm

As shown from test 4 to test 11 different set points were entered on the main pumps set point and results captured proved that DP flowmeter was updating faster than a conventional (hard wired) flowmeter. In short the proposed solution met the requirement of the system and process plant components performed as expected. Table 15 shows the results captured when both drives were controlled from one main set point both drives were running. This was done and the two signals were compared on condition both drive run at the same speed all the time by entering one set point. As seen in table 15, all captured results are shown from test 1 to test 11.Both variable speed drives, DP flowmeter and PA flowmeter results were captured during tests. Results achieved proved that proposed solution met the design specifications and detailed analysis of real-time results determined that the prototype performed as designed. As shown from test 4 to test 11 different set points were entered on the main pumps set point and results captured proved that DP flowmeter was updating faster than a conventional (hard wired) flowmeter. In short the proposed solution met the requirement of the system and process plant components performed as expected.

Table 16 shows the results captured when both drives were controlled one main set point both drives were running. This was done for the reason that since two signals were compared on condition both drive run at the same speed all the time by entering one set point. As seen in table 16, all captured results are shown from test 1 to test 11 for variable speed drives, DP flowmeter and PA flowmeter tests. Results achieved proved that proposed solution met the design specifications and detailed analysis of real-time results determined that the prototype performed as designed.

Table 16: Process Plant results

Item	Main Drive set point	MM440 Drive1	MM420 Drive 2	Achieved Results	DP flowmeter results	Hardwired (PA) flowmeter results	DP Flowmeter and PA Flowmeter response variances
Test 1	1400rpm	773rpm	1331rpm	Result 1	1.7 l/m	0.0 l/m	1.7 l/m
Test 2	1400rpm	1388rpm	1398rpm	Result 2	4.4 l/m	4.2 l/m	0.2 l/m
Test 3	100rpm	98rpm	98.2rpm	Result 3	0.8 l/m	0.2 l/m	0.6 l/m
Test 4	1000rpm	998rpm	997rpm	Result 4	5.2 l/m	4.4 l/m	1.2 l/m
Test 5	1200rpm	1201rpm	1200rpm	Result 5	4.8 l/m	4.1 l/m	0.7 l/m
Test 6	1400rpm	1398rpm	1390rpm	Result 6	6.2 l/m	5.6 l/m	1.4 l/m
Test 7	1000rpm	459rpm	845rpm	Result 7	1.4l /m	0.0 l/m	1.4 l/m
Test 8	1400rpm	960rpm	1425rpm	Result 8	2.2 l/m	2.1 l/m	0.1 l/m
Test 9	1400rpm	1423rpm	1424rpm	Result 9	4.3 l/m	4.0 l/m	0.3 l/m
Test 10	1400rpm	1097rpm	1423rpm	Result 10	3.6 l/m	3.2 l/m	0.4 l/m
Test 11	1200rpm	1240rpm	1246rpm	Result 11	4.6 l/m	4.4 l/m	0.2 l/m

CHAPTER ELEVEN: ANALYSIS OF TEST RESULTS

11.1 Introduction

The objective of the research was to analyse communication of PROFIBUS using process automation decentralized periphery against conventional (4-20mA) and also conduct an analysis of data transmitted using a conventional 4-20mA and PROFIBUS DP communication systems as they are both incorporated in the research. All simulated results were compared with the real-time results and then a conclusion was drawn based on the obtained information and facts [48].

The research objectives were achieved since the prototype performed as designed and research met the design specification. All tests were conducted using different methods and also analysed results based on the proposed solution and verified that system performed as per the design. Evaluated communication methods, including conversion of the PROFIBUS –DP network transmission system from RS485 (bit coding with asynchronous code) to IEC 1158-2 and examined how data was transferred to the SIMATIC PCS 7 control system via PROFIBUS-DP at 187.5 Kbit/s without significant loss of time (approximately 1ms). Further analysed how the input and output parameters of the function blocks were connected via the bus and linked to a process engineering application. These field device blocks provided the interface to the hardware, including functionality verification.

11.2 Practical tests and analysis of the achieved results

The purpose of this research was to explain the performance of reports, table and graphs that were generated during process plant tests, to enable researcher to quickly and easily interpret the results from them and also to verify if prototype performed as per the design. This assisted as a guide to interpreting performance reports and expected results. An important part of this research was to compare and analyse PROFIBUS communication using process automation and decentralised periphery against conventional (4-20mA). Meaningful results were achieved from the accurate tests, the following table and graphs allowed researcher to analyse each tests condition. The summaries of captured results are attached on the table in 14, 15 and 16.

From the tables and graphs, it was possible to identify which signal was responding in time or faster during all tests and then continued to analyse the response times of individual test stage. This table/graph helped the researcher to evaluate the amount of results generate in terms of process plant output. When compared, the response time of these two signals, it was discovered that DP flowmeter was updating faster than PA flowmeter. Before the process plant was put in practical some

mathematical calculation were done as the part of technical analysis to verify which of the two signals was expected to update faster than the other. This calculation formed the part of simulation test based on PROFIBUS technology .The PROFIBUS-DP and PA systems required a description of the device parameters such as output data, input data, data format, data length and the transmission rates supported [48] and [49]. The DP PROFIBUS calculation was done as follows:
Calculations of the DP flowmeter cycle times as follows;

(i) 111 is an assigned network address for DP flowmeter.
(ii) The transmission rate is (187.5 Kbit/s).
(iii) Input data (2 bytes and output 1 byte) = 3 bytes for the DP flowmeter.

Cycle time = Σ (cycle time of the devices) + PLC cycle time (cal. 10 ms)
= 1 x 10 ms + 10 ms + (187.5 kbit/s)
= 5ms

Calculations of the PA flowmeter total cycle time as follows;

PA total cycle time = (n x 10 ms) + (PLC program run time)

(i) Cycle time = Σ (cycle time of the devices) + (cycle time per link) + (PLC-cycle time)
(ii) PROFIBUS-PA segment 1= 1 x 10 ms + 1 x 1.1 ms + 10 ms + 31.25kb/s
(iii) = 6.7 ms,

Analysis of mathematical calculated results attained using standard PROFIBUS formulas showed that PD flowmeter response time was 5ms and PA flowmeter response time was 6.7ms. From the obtained results it was clear that PA flowmeter update time was slower by 1,7ms where compared with the DP flowmeter. The cycling time for each signal include input data, output data, data format, data length and the transmission rates supported for both signals. The reason for this calculation was to identify which signal was expected to respond faster. Calculated results evidenced that DP flowmeter was going to update faster than PA flowmeter. Referring from Table 15 shows all results that were captured during simulation tests and to show the reader results achieved and to further generates figures that gave more clarity about the research in a graph form.

Data were collected from two flowmeters and two drivers, PA flowmeter, DP flowmeter, MM440 and MM420, as summarized in Table 15. Referring to test1, 1400rpm set point was entered to the main set point to start both drives at the same time and keep then running at the same speed. The Speed indicated that MM440 drive was running at 773 rpm and MM420 drive was running at 1421rpm. Before both drives reached desired set point, a DP flow meter was reading 1,7l/m and a PA flowmeter had no reading while MM420 drive was already running at 1421rpm at that time. Test 2 shows that when both drives were running at the same speed, a DP flowmeter was reading 4,4l/m and a PA flowmeter was reading 4,2l/m.

Results from test 1 showed that a PA flowmeter had no reading and test 2 results showed that a PA flowmeter starting measuring and it was reading 0.2l/m than a DP flowmeter. Test 3, 4 and 5 were implemented by entering a main set point from 1400rpm to 100rpm, then 1200rpm and 1400rpm. Test 3, 4 and 5 results showed that a PA flowmeter was still updating slower than a DP flower meter, all results are shown in table 15. Further tests were done so that the researcher has a fully understanding of the study and to verify results achieved were accurate. Referring to test 6, main speed drive was set to 1400rpm and speed from MM440 was 1398rpm and MM420 was 1390rpm. Data collected after 5 seconds indicated that even when both pump were running at the same speed but a PA flowmeter was reading 5,6l/m while a DP flowmeter was measuring 6.2l/m. A PA flowmeter only measured matching reading as a DP flowmeter after 5 seconds.

Further experimental tests were conducted from tests 7, 8 and 9 by entering difference set points as shown in table 15 and both drivers were running at the same speeds and results indicated that a PA flowmeter was measuring +/- 1.4l/m lower that DP flowmeter. All PA flowmeter results were updating slower than a DP flowmeter by almost 5 seconds. In order to prove that a DP flowmeter was updating faster than a PA flowmeter, researcher decided put a delay timer on a DP flowmeter loop on the MM440. The reason for this change was to check if MM440 drive was delayed by 10 seconds before starting while MM420 running. Obtained results from test 10 showed that MM440 drive was reading 1097rpm and MM420 was measuring 1423rpm. A PA flowmeter measurement continued to update slower by 0.4l/m but update time improved from 5 seconds to 2.5 seconds.

Final test was conducted by entering a set point of 1200rpm to the main drive set point, both drives ran at the speed as shown in table 15 and a PA flowmeter took approximately 5 seconds before measurement was close to a DP flowmeter. Results showed that a PA flowmeter was reading 4.4l/m and a DP flowmeter was measuring 4.6l/m and it was evident from mathematical calculation that a DP flowmeter was updating faster than a PA flowmeter. Additional practical experiments revealed that when both flowmeters were compared, it was verified that a DP flowmeter was the suitable flowmeter to be selected for critical conditions. The delay timer was added to the MM440 drive to delay pump 1 from starting at the same time with the MM420.

This was done to check if a PA flowmeter was able to update faster than a DP flowmeter. Results captured showed that a DP flowmeter was still the better measuring flowmeter to be used for fast data exchange and for safety process plant purposes. I was noted that a DP flowmeter was able to detect flow in the pipe line and update on HMI before PA flowmeter. A comparison trends were generated during experiment tests and results obtained as displayed in figure 80. On the graph there are two continuous signals that were recorded and each was identified with different colours and one extra trend was created from variances of the two signals.

As shown that from test 1 to test 11 and results achieved shows that a DP flowmeter was always reaching set point before a PA flowmeter. The reason for creating the graph was to understand and analyse each and every test that was conducted. The comparison of two results shows that the design specification was achieved. Figure 80 shows the comparison of DP flowmeter and PA flowmeter results including response variation that was created when both signals were analysed. Figure 81 shows a main drive set point, MM440 current speed andMM420 actuals speed.

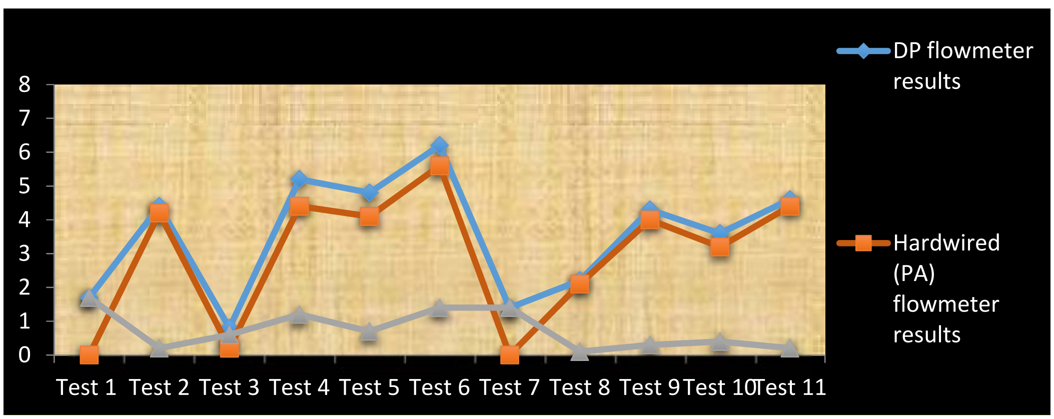


Figure 80: Analysis of DP and PA flowmeter results

Each measurement was identified by its colour and the data shown was captured during experimental test. Since the objectives of this research was to analyse the DP PROFIBUS communication using decentralised peripherals against conversion of 4…. 20ma, this was achieved because both drive were running at the same speed all the time by following the main set point speed. Another reason was to verify if both drives had same parameters and configuration so that good results were achieved.

The results achieved confirmed that both drives were running at the same speeds all the time so that both flowmeters measurements were analysed. This concludes that the researcher has a better understanding of the tests performance and report analysis. Tested the operational of the entire process plant and the accuracy of the results proved that process plant performed as designed. Estimated calculation also evidenced that DP flow meter was updating faster than PA flow meter and even when tried to add delay time on DP drive signal but DP flow meter was still updating faster that PA flow meter as shown in figure 80.

Figure 81: Analysis of Main Drive set point speed, MM440 and MM420 drive

CHAPTER TWELVE: CONCLUSION AND RECOMMENDATIONS

12.1 Introduction

The research was challenging with lots of sophisticated equipment's but was successfully executed and the process plant performed as per the design. With regards to design, hardware, software's installation and configuration were extremely important. All process plant equipment used in the research were checked for functionality, calibration, tested according to their standards after the PLC software was loaded and the system configuration completed. The process plant was completely tested and the design met the specified requirements.

Since the main objective of the research was to perform a comparative study between PROFIBUS DP (decentralized periphery) and conventional 4-20mA analogue communication in terms of fast data exchange between PLC and devices in PROFIBUS DP and PROFIBUS DP/PA. Detailed results showed that the objectives of the research were achieved because the prototype met design specifications and accurately data was captured and analysed in many ways including graphs. During simulation and testing, the following was also taken to account in order to obtain perfect results; communication structures of PROFIBUS DP/PA-Interface transitions and analyse the effect of PLC scan cycle time when signal was transmitted from device via DP/PA network. Also carried out a hardware configuration of the modules in PROFIBUS DP and PROFIBUS PA and analyse the effect of data transition when more devices are connected to one PROFIBUS DP network.

Referring in table 15, according to test 1 to test 11 and results achieved from result 1 to result 11 revealed DP flowmeter was updating faster than a PA flow meter (hardwired) in all conditions. When one set point was entered and both pumps were running at the same speed either by increasing and decreasing main speed set point. The different speed set points were entered on the HMI and results showed that both met design specification because both pumps were running at the same speed as shown in table 16. Results attained proved that PA flowmeter was updating slower than a DP flowmeter. Several tests were done to check if all condition were within required specification and obtained results indicated system design met the proposed solution.

As seen in figure 79, 80 and 81 that when both signals were plotted on the graph and compared. It was cleared DP flowmeter was the better flow meter that be used on critical plants for rapid response when information for the plant is required. Main set point, drive one signal and drive two signal were put onto a graph and displayed that both pumps were within required requirement. Analysis of mathematical calculation were used to calculate PROFIBUS time response for each flowmeter and proved that DP flowmeter was expected to update faster than a PA flowmeter. On experimental chapter it was explained that since PA flowmeter was a field device and its data was transmitted via field module interface (ET200M) and then transferred to DP/PA coupler were it was then converted to PROFIBUS DP so that PLC can recognise the PA flowmeter data.

Other objectives of the research was to analyse the handling of functional blocks between PROFIBUS DP and comparison of PROFIBUS PA (4-20mA) including a conversion of numbers from decimal to binary, BCD, and hexadecimal. The results achieved were observed on the HMI colour screen displays and on HMI trend pages. The tests resulted in a better understanding of the DP PROFIBUS communication and against conversion of 4…20ma using process automation. Future data analysis included further analysis of the PROFIBUS communication and additional comparisons of the results obtained during experimentation and advance analysis of the outcome revealed that prototype performed as designed.

12.2 Limitation of the present work

Process Plant was still not fully automated, because the 3rd pump was controlled manually to empty water when Mix tank was full and level transmitter was added on the PROFIBUS network during the research and display some measurement as mix tank fill ups. It was not calibrated and also some rework was still required because the level transmitter was not attached to the tank. Law of instrumentation does not support the way level transmitter was connected.

Firstly level transmitted needs to be mounted on the tank to measure correctly and also calibrated. When pump one and pump two was pumping water to tank 3, there was no level indication and pump 3 was run manually. It was suggest that the level transmitter must be included on the PLC program and the same measurement be used as the interlock for low level and high level to start or force pump 3 from starting if water is full or empty. The reason for these interlocks was to protect pump 3 from running dry. On/off valve must open when a high level signal is received and close if low level signal is active. This will protect pump 3 from pumping water while outlet valve is closed. Tank 1 and Tank 2 has a low and high limit witches of which they can be used to control both pumps to start if the level is high and stop if level is low.

This recommendation will minimise risk of running pumps dry because at this stage you have to visual see if there is enough what before pumps start. When the button is pressed on the HMI, pumps must start if high signal is active. There is nothing notify the operator/user if there is water in tank 1 and 2. Pump 1 and 2 must not start if level in tank 1 and tank 2 is low.

Also level transmitter for tank 3 must indicate the level and start up 3 and open on/off valve if the level is high. When the level is low pump 3 must not start. This recommendation will prevent process plant equipment's form unnecessary failures. HMI colour screen is required because at the moment HMI graphic design is black and white and it's difficult to view clearly plant pages. Due to that screen is too small it is difficult to select and control pages. I had to hire a colour screen HMI from Siemens for this research in order to view all signals in different colours during simulation, analyses and testing. All mentioned recommendation will protect equipment's from getting damaged and also the researcher will understand the importance of protection and interlocks.

12.3 Application of the research

The following assumption and benefits for MUT have been considered:

(i) Advantage of the process plant is user friendly and easy.

(ii) The HMI system will provide better plant overview with addition different pages and MUT will have less components failure since all components are new. The process plant can be accessible to local area network (LAN) to view remotely while plant is running.

(iii) Process Plant user/operator will have warnings on the displays to quick finding the root cause of the fault. The recommendation with regards this research is that every fault and alarm/status signal is coded/logged as an individual error code. In most cases, it is possible to reset the fault message like if there is a drive fault.

The process plant will be reliable and require low maintenance for the period of at least additional 10 years before the process plant equipment's reaches the obsolete phase. A considerable amount of work and design went into a research even though it was simple process of calibrating, configuring and testing. MUT students will able to identify different between the equipment's used on the research and understand the option of network communication. Siemens spares and competence are available throughout the entire lifecycle. Another important criterion is that the capability of the expansion of existing process plant section without having to modify the current plant and bus system will can be characterized in that many stations to communicate with each other using a low number of cabling. The recommendation for the future work on the comparison of PROFIBUS to 4-20ma is to compare the PROFIBUS DP against wireless PROFIBUS using process automation devices. The future work will able to simplify the differences in terms of data transmission, cycle time, amount of data per devices and input/output data including switching technology (simultaneous sending/receiving) and communications paths.

12.4 Implication of the research

Fieldbus networks are becoming a popular communication protocol in industry because of its convenience and efficiency. Production, quality and efficiency are the greatest concerns in the industry because the main aim is to obtain maximum quality product with less overheads and great reliability. A recent trend in distributed process control systems is to interconnect the distributed elements by means of a multipoint broadcast network such using PROFIBUS, instead of using a traditional point-to point links as in a 4-20mA.

Within industrial communication systems fieldbus networks are specifically intended for the interconnection of process controllers, sensors and actuators at the lower levels of the factory automation hierarchy. However, these hierarchical levels have dissimilar message flows, in terms of required response times, amount of data to be transferred, required reliability and message rates, that is how frequent messages must be transferred. All this makes PROFIBUS communication a better choice over conventional 4-20mA because of its ability of using one communication cable allowing the PLC to communicate with other devices without connecting each one to the PLC, less cabling. When comparing PROFIBUS DP against 4-20mA communication, the following are advantages:

(i) Significant reduction in the amount of wiring.

(ii) Centralised commissioning of devices, from the control room.

(iii) More extensive data to and from devices that is diagnostic information and device parameters, etc.

(iv) Quality signal with highest resolution and easy maintenance with fewer cables and hence less cost.

(v) Greater flexibility in system layout and design, ease of future expansion and modification.

(vi) Reduce downtimes because of quick repair and troubleshooting.

(vii) Easy to trace the fault and device has more data for engineering level.

(viii) Data is speedily exchanged and can be installed in critical areas or hazardous areas

(ix) Recent developments mean that PROFIBUS is also applicable to:

(x) High reliability, safety critical systems (PROFISafe) and Integration with management IT systems

(xi) (PROFINet).

The operating procedures will be available for problem tracing, maintenance, testing and modification including the designed drawing and software back-up. All documents will be filed accurately and stored for future alterations. The benefits of MUT and future researchers with the process plant as follows:

(i) Electrical Department will keep a process plant that is functional and they will able to demonstrate for final year students. Students will gain knowledge and basic understand of what is expected from them when they leave MUT.

(ii) Students will able to observe a live process plant and control the plant from the HMI screen.

(iii) Process plant screen will locate faults according to their categories to facilitate the corrective action to be implemented.

The dissertation was validated by data obtained from real PROFIBUS protocols and the results were satisfactory, proving the great strength and versatility that intelligent PROFIBUS systems have when applied to the outlined determinations in this work. Results were assessed from technical specifications of both protocols and practical experiments for data collection and conclusion showed that the PROFIBUS-DP was faster than PROFIBUS-PA. There is still future work on the comparison of communication speeds of PROFIBUS-DP against PROFIBUS wireless because in nowadays, PROFIBUS is often used for information transfer in the field level. Due to timing level requirements, which have to be strictly observed in an automation process, the application in the field level controllers require cyclic transport functions, which transmit source of information at the regular intervals. The data representation must be short as possible in order to reduce message transfer on the bus.

REFERENCES

[1] Lipták, Béla G; (2003), Instrumentation Engineers' Handbook. Process Measurement and Analysis. CRC Press. HB. ISBN 0-8493-1083-0.

[2] Gregory K. McMillan, Douglas M. Considine; (1999), Process/Industrial Instruments and Controls Handbook Fifth Edition, McGraw-Hill, ISBN 0-07-012582-1 Section 3 Controllers.

[3] T.Bangemann, S. Karnouskos, R. Camp, O. Carlsson, M. Riedl, S.McLeod, R.Harrison, A W. Colombo and P.Stluka (2014). State of the Art in Industrial Automation.

[4] J. Weigmann, G. Kilian: (No Date), Decentralization with PROFIBUS DP/DPV1, ISBN 978-3-89578-218-3.

[5] C.Yue-ting and H Fang; (2007), "Analysis of PROFIBUS Field Technology and Evolution Trends.

[6] S Vitturi; (2004), On the effects of the cyclic traffic on the PROFIBUS DP network.

[7] Zheng-shi Chen, J.z Song and Jian-fang Shi (2013), Real-time property analysis of single master DP/PA hybrid network control system.

[8] W. S. Nagel, G. M. Clayton and K. Leang (2016), Master-slave control with hysteresis inversion for dual-stage nanopositioning systems.

[9] Babb, Michael; (1994), another Fieldbus Delay, Schneider's DPV, and Open Systems Control Engineering.

[10] Prof M. Felser; (July 09, 2013), PROFIBUS, Networks assembled, Edition 1.2.3.

[11] http://en.wikipedia.org/wiki/Human%E2%80%93computer interaction.

[12] Mossin, E.A., Brandão, D., Sestito, G.S. et al and J Control Autom (2016). Automatic Diagnosis for Profibus Networks.

[13] Y.Zhou ,Xiao-kun Li and Ming-shan Liu (2016), Application prospect of 4G/LTE technology in PROFIBUS.

[14] Brown S, Vranesic Z & Mc Graw Hill; (2009), Fundamentals of Digital Logic with VHDL Design. 3rd Ed. New York.

[15] Jason C Hung, Neil Y. Yen and Kuan-Ching Li (2016), CPP Control System Design of Ship Based on Siemens PLC.

[16] M. Roda; L. Mendes, P. Gomes, G. Pigny and S. Faria (2015), A new Profibus-DP interface for CERN's sputter ion pump controllers.

[17] http:/www.sci-eng.mmu.ac.uk/ascent/literature/documents/installationguide.pdf.

[18] G.Sou-Liang, W.Yin-qiang and F. Tang-li, (2011), Application of PROFIBUS and MPI inpaper Manufactrue communication network.

[19] Liju G Chacko and Hugh Jack; (No Date), PLC programming languages Sequential Function Chart Important rules for SFC evolution [IEC 61131-3].

[20] Fraden, Jacob; (2010). Handbook of Modern Sensors: Physics, Designs, and Applications. Springer. ISBN 978-1441964656.

[21] Y.Zhou, D.Chai, M. Liu, F. Lin, W. Shang and Liang Wang (2014). Research on the security mechanism for interconnection between PROFIBUS and Internet.

[22] L. Gevorkov, V.Vodovozov, T.Lehtla and I. Bakman (2015), PLC-based flow rate control system for centrifugal pumps. International Scientific Conference.

[23] H. B. Mokadem; (2010), verification of a Timed Multitask System with UPPAAL, IEEE Transactions on Automations Science and Engineering.

[24] M. Masters, L. Osborn, N.Thakor (2015), Real-time arm tracking for HMI applications.
[25] http://www.seonam.seoul.kr/auto/Profibus.htm.
[26] Drury, Bill; (2009), Control Techniques Drives and Controls Handbook (PDF) (2nd ed.), Institution of Engineering and Technology.
[27] ANSI/ISA 62381; (No Date), Factory Acceptance Test (FAT), Site Acceptance Test (SAT), and Sit Integration Test (SIT), Automation Systems in the Process Industry.
[28] J. O. Blech and S. O. Biha. (2011), Verification of PLC Properties Based on Formal Semantics in Coq, 9th SEFM, Montevideo (Uruguay).
[29] B. Fern´andez; (No Date), model-based Automated Testing of Critical PLC Programs, 11th INDIN. Bochum (Germany).
[30] P.Kolaitis; (2005), (Schema mappings, data exchange, and metadata management), Proceedings of the twenty- fourth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems.
[31] L.Guoxin, Z.Wanli, W.Lirong (2015). Application of OPC to Realize the Communications between WinCC and Master-Slave PLC in the PROFIBUS Network.
[32] N.P. Mahalik; P.R. Moore; (1997) Fieldbus technology based, distributed control in process industries: a case study with Lon Works Technology.
[33] ARC Advisory Group; (2008) Foundation Fieldbus Safety Instrumented Functions Forge the Future of Process Safety.
[34] C. Fillyaw, J. Friedman, Sameer M. Prabhu; (No date), Creating Human Machine Interface (HMI).
[35] T. Sedran, T.Wendt; (September 2007) the Math Works Inc.
[36] Raskin, Jef;(2000), The human Machine interface: new directions for designing interactive systems: Addison Wesley. ISBN 0-201-37937-6.
[37] STROUD, K.A; (1995), Engineering Mathematic. Fourth Edition.
[38] M. Silva, F. Pereira, F. Soares, C. P. Leão, J. Machado and V. Carvalho (2014). An Overview of Industrial Communication Networks.
[39] Roger C. Baker; (No date), Flow Measurement Handbook, Cambridge university press. ISBN 978-0-521-0176.
[40] Hans Berger; (No Date), Automating with SIMATIC S7-1200 - ISBN 978-3-89578-356-2.
[41] S.Singh V. M Chary, P. Abdul Rahman (2014), Dual Redundant Profibus Network Architecture in hot standby fault tolerant control systems.
[42] W.S. Nagel, G.M. Clayton and K. K. Leang (2016), Master-slave control with hysteresis inversion for dual-stage nanopositioning systems.
[43] Z.Lin,S Chuanjum and Z. Dawei,(2010), Multiphase flow control system based on PLC and WINCC.
[44] M. Paulweber and K. Lebert (2016),Powertrain Instrumentation and Test Systems.
[45] C.Jinliang, C. Enzhi, L.Hai and W.Zhaoju, (2013), Fault Diagnosis of PROFIBUS-DP Network in Industrial Automation.
[46] Z. Botero, J. Gonzalo, H.Cuartas, J. Aurelio and S. Garces, S. Ignacio (2014),Gateway for Profibus communication between a PLC and embedded systems.

[47] Z.Botero, J. Gonzalo, H. Cuartas, J. Aurelio and S. Garces (2015), Gateway for Profibus communication between a PLC and embedded system.
[48] Sachin Karadgi (2014). Real-Time Performance Measurement in Manufacturing.
[49] A. Luis Dias, G. Serpa Sestito and D. Brandao (2016), Performance Analysis of Profibus DP and Profinet in a Motion Control Application.

PLC program, Profibus network and network addressing

OB1 - <offline>

""

Name: **Family:**
Author: **Version:** 0.1
Block version: 2
Time stamp Code: 09/02/2014 10:14:01 PM
Interface: 02/15/1996 04:51:12 PM
Lengths (block/logic/data): 00142 00030 00022

Name	Data Type	Address	Comment
TEMP		0.0	
TEMP0	Byte	0.0	
TEMP1	Byte	1.0	
TEMP2	Byte	2.0	
TEMP3	Byte	3.0	
TEMP4	Byte	4.0	
TEMP5	Byte	5.0	
TEMP6	Int	6.0	
TEMP7	Int	8.0	
TEMP8	Int	10.0	
TEMP9	Date_And_Time	12.0	

Block: OB1

Network: 1

```
CALL  "DRIVE NO:1 CONTROL"  FC2
CALL  "DRIVE NO:2 CONTROL"  FC3
```

SIMATIC 300(1)

UR - Rack (0)

Short description:	UR
Order no.:	6ES7 390-1???0-0AA0
Designation:	UR

Rack (0), Slot 1

Short description:	PS 307 2A
Order no.:	6ES7 307-1BA01-0AA0
Designation:	PS 307 2A
Width:	1
Comment:	- - -

Rack (0), Slot 2

Short description:	CPU 313C-2 DP
Firmware version:	V2.6
Order no.:	6ES7 313-6CF03-0AB0
Designation:	CPU 313C-2 DP
Width:	1
MPI address:	2
Highest MPI address:	31
Baud rate:	187.5 Kbps
Comment:	
- - -	

Rack (0), Slot 2, Interface X2

Short description:	DP
Order no.:	- - -
Designation:	DP
Width:	1
PROFIBUS address:	2
Highest PROFIBUS address:	126
Baud rate:	187.5 Kbps
Comment:	- - -

Addresses
Inputs

Start:	1023
End:	1023

Synchronization mode:	None
Time interval:	None

Operating mode:	DP Master

Rack (0), Slot 2, Interface 2

Short description:	DI16/DO16
Order no.:	- - -
Designation:	DI16/DO16
Digital channels:	16 Inputs 16 Outputs
Width:	1
Comment:	- - -

Addresses
Inputs

Start:	124
End:	125

Outputs

Start:	124

End: 125

Rack (0), Slot 2, Interface 4

Short description:	Count
Order no.:	- - -
Designation:	Count
Width:	1
Comment:	- - -

Addresses
Inputs
Start: 768
End: 783
Outputs
Start: 768
End: 783

DP master system:

Assigned master:

Short description	DP
Order no.	
Designation	DP
Location	
Station	SIMATIC 300(1)
Rack	0
Slot	2
Receptacle for interface module	1
PROFIBUS address	2

Group: 1
Comment:
The group supports SYNC.
The group supports FREEZE.

Group: 2
Comment:
The group supports SYNC.
The group supports FREEZE.

Group: 3
Comment:
The group supports SYNC.
The group supports FREEZE.

Group: 4
Comment:
The group supports SYNC.
The group supports FREEZE.

Group: 5
Comment:
The group supports SYNC.
The group supports FREEZE.

Group: 6
Comment:
The group supports SYNC.
The group supports FREEZE.

Group: 7
Comment:
The group supports SYNC.
The group supports FREEZE.

Group: 8
Comment:
The group supports SYNC.
The group supports FREEZE.

Slave : IM151-1 Standard PROFIBUS address: 1
Assigned to the groups:
Slave : IM 153-2 OD, Redundancy PROFIBUS address: 4
Assigned to the groups:
Slave : MICROMASTER_420 PROFIBUS address: 8
Assigned to the groups:
Slave : MICROMASTER 4 PROFIBUS address: 9
Assigned to the groups:
Slave : MAG6000 DP PROFIBUS address: 111
Assigned to the groups:

Slave (4) **IM 153-2 OD, Redundancy**

Order Number:	6ES7 153-2BA82-0XB0
Family:	DP/PA-LINK
DP slave type:	IM 153
Comment:	
PROFIBUS address :	4
Diagnostic address:	1022
SYNC:	No
FREEZE:	No
Watchdog:	Turned on
Plant designation:	- - -
Installation date:	11/07/2013
Additional information:	- - -
Location ID:	- - -
Startup only if setpoint equals actual value	No

The slave is connected with ...

PROFIBUS(1): DP master system (1):	Yes

Slave (4), Slot 2

Short description:	IM153-2 OD
Order no.:	6ES7 153-2BA82-0XB0
Designation:	IM153-2 OD
Width:	1
PROFIBUS address:	4
Highest PROFIBUS address:	126
Baud rate:	187.5 Kbps
Comment:	
- - -	
Diagnostic address:	1022

Slave (4), Slot 4

Short description:	PA master
Order no.:	6ES7 153-2BA82-0XB0
Designation:	IM 153 PA-Master
Width:	1
PROFIBUS address:	2
Highest PROFIBUS address:	2
Baud rate:	45.45 (31.25) Kbps
Comment:	- - -

Addresses

Synchronization mode:	None
Time interval:	None

Slave (1) **IM151-1 Standard**

Order Number:	6ES7 151-1AA04-0AB0
Family:	ET200S
DP slave type:	ET200S
Comment:	
PROFIBUS address :	1
Diagnostic address:	1020
SYNC:	Yes
FREEZE:	Yes
Watchdog:	Turned on
Plant designation:	- - -
Installation date:	- - -
Additional information:	- - -

Slave (1), Slot 1

Order no.	6ES7 138-4CA01-0AA0
Short description	PM-E DC24V
Designation	PM-E DC24V

Slave (1), Slot 2

Order no.	6ES7 134-4GB00-0AB0
Short description	2AI I 2WIRE ST
Designation	2AI I 2WIRE ST
I Address	261...264

Slave (1), Slot 3

Order no.	6ES7 131-4BD01-0AA0
Short description	4DI DC24V ST
Designation	4DI DC24V ST
I Address	0.0...0.3

Slave (1), Slot 4

Order no.	6ES7 132-4HB10-0AB0
Short description	2RO NO/NC 24..230V/5A
Designation	2RO NO/NC 24..230V/5A
Q Address	0.0...0.1

Slave (1), Slot 5

Order no.	6ES7 132-4HB10-0AB0
Short description	2RO NO/NC 24..230V/5A
Designation	2RO NO/NC 24..230V/5A
Q Address	1.0...1.1

Slave (1), Slot 6

Order no.	6ES7 132-4HB10-0AB0
Short description	2RO NO/NC 24..230V/5A
Designation	2RO NO/NC 24..230V/5A
Q Address	2.0...2.1

Slave (1), Slot 7

Order no.	6ES7 132-4HB10-0AB0
Short description	2RO NO/NC 24..230V/5A
Designation	2RO NO/NC 24..230V/5A
Q Address	3.0...3.1

Slave (1), Slot 8

Order no.	6ES7 132-4HB10-0AB0
Short description	2RO NO/NC 24..230V/5A
Designation	2RO NO/NC 24..230V/5A
Q Address	4.0...4.1

Slave (111) **MAG6000 DP**

Order number:	
Family:	PA
DP slave type:	MAG6000 DP
Manufacturer:	Siemens Flow Instruments
GSD file name:	SI018129.GSE
GSD revision:	5
Identification number:	0x8129
Revision of the DP slave:	2.00
Hardware release:	Rev C
Software revision level:	2.00
Comment:	
PROFIBUS address :	111
Diagnostic address:	1019
SYNC:	No
FREEZE:	Yes
Watchdog:	Turned on

Parameter assignment in ascending byte sequence (hexadecimal specification) :
80,00,00,05,20,00,00,01

DP ID:	**148**
Module description:	AI (Volumeflow)
Input address:	265
DP ID:	**193**
Module description:	TOTAL, SET_TOT, MODE_TOT
Input address:	270
Output address:	256
User-specific comment:	85
DP ID:	**0**
Module description:	EMPTY_MODULE

Slave (9) **MICROMASTER 4**

Order number:	6SE640X-1PB00-0AA0
Family:	Drives
DP slave type:	MICROMASTER 4
Manufacturer:	Siemens AG A&D
GSD file name:	SI0280B5.GSE
GSD revision:	3
Identification number:	0x80b5
Revision of the DP slave:	B01
Hardware release:	B01
Software revision level:	V1.0
Comment:	
PROFIBUS address :	9
Diagnostic address:	1017
SYNC:	Yes
FREEZE:	Yes
Watchdog:	Turned on

Parameter assignment in ascending byte sequence (hexadecimal specification) :
C4,00,00

DP ID:	**185**
Module description:	Universal module
Input address:	300
Output address:	300
Slave(8):	**MICROMASTER_420**
Station:	SIMATIC 300(1)
Subnet:	PROFIBUS(1)
PROFIBUS address:	8
Diagnostics address:	1015 ...
Order number:	6SE6400-1PB00-0AA0 (MM4B)
Family:	SIMOVERT
Device type:	MICROMASTER 420
Comment:	
SYNC capability:	Yes
FREEZE capability:	Yes
Response monitoring:	Yes
Preset configuration:	None
Slot:	4
Drive:	
Type:	No PIV
Address:	
PROFIBUS partner:	
Type:	
PROFIBUS address:	
I/O address:	
Length:	
Unit:	
Consistency:	
Comment:	
Slot:	5
Drive:	
Type:	Actual value
Address:	PZD 1
PROFIBUS partner:	
Type:	Input
PROFIBUS address:	2
I/O address:	400
Length:	4
Unit:	Word
Consistency:	Total length
Comment:	
Slot:	6
Drive:	
Type:	Setpoint
Address:	PZD 1
PROFIBUS partner:	
Type:	Output
PROFIBUS address:	2
I/O address:	400
Length:	4
Unit:	Word
Consistency:	Total length
Comment:	
Required optional package:	Drive ES / SIMOTION
Required optional package:	Drive ES / SIMOTION
Required optional package:	Drive ES / SIMOTION
Required optional package:	Drive ES / SIMOTION

DP master system:

Assigned master:

Short description	PA master
Order no.	6ES7 153-2BA82-0XB0
Designation	IM 153 PA-Master
Location	
Station	SIMATIC 300(1)
Rack	0
Slot	4
Receptacle for interface module	0
PROFIBUS address	02

Group: 1
Comment:
The group supports SYNC.
The group supports FREEZE.

Group: 2
Comment:
The group supports SYNC.
The group supports FREEZE.

Group: 3
Comment:
The group supports SYNC.
The group supports FREEZE.

Group: 4
Comment:
The group supports SYNC.
The group supports FREEZE.

Group: 5
Comment:
The group supports SYNC.
The group supports FREEZE.

Group: 6
Comment:
The group supports SYNC.
The group supports FREEZE.

Group: 7
Comment:
The group supports SYNC.
The group supports FREEZE.

Group: 8
Comment:
The group supports SYNC.

The group supports FREEZE.

Slave : FDC 157-0 PROFIBUS address: 3
Assigned to the groups:
Slave : SITRANS P DSIII PROFIBUS address: 6
Assigned to the groups:

Slave (3)	**FDC 157-0**
Order number:	6ES7 157-0AC83-0XA0
Family:	PA
DP slave type:	FDC 157-0
Manufacturer:	SIEMENS
GSD file name:	SI028131.GSE
GSD revision:	5
Identification number:	0x8131
Revision of the DP slave:	V0.0
Hardware release:	A1.0
Software revision level:	Z1.0
Comment:	
PROFIBUS address :	3
Diagnostic address:	1021
SYNC:	Yes
FREEZE:	Yes
Watchdog:	Turned on
Identification	
Plant designation:	
Location designation:	
Installation date:	
Additional information:	

Parameter assignment in ascending byte sequence (hexadecimal specification) :
C4,00,08,08,81,00,00,60,04,04,00

DP ID:	**148**
Module description:	current
Input address:	256
Slave (6)	**SITRANS P DSIII**
Order number:	
Family:	PA
DP slave type:	SITRANS P DSIII
Manufacturer:	SIEMENS AG
GSD file name:	SIEM80A6.GSE
GSD revision:	3
Identification number:	0x80a6
Revision of the DP slave:	3.1.1
Hardware release:	A01
Software revision level:	0300.01.05
Comment:	
PROFIBUS address :	6
Diagnostic address:	1018
SYNC:	No
FREEZE:	No
Watchdog:	Turned on

Parameter assignment in ascending byte sequence (hexadecimal specification) :
00,00,00

DP ID:	**148**
Module description:	AI Output
Input address:	275
DP ID:	**0**
Module description:	Not in cyclic data transfer

DB20 - <offline> - Declaration view

"DRIVE NO:1 STATUS WORD" STATUS WORD 1
Global data block DB 20
Name: **Family:**
Author: **Version:** 0.1
Block version: 2
Time stamp Code: 10/02/2013 11:17:47 AM
Interface: 10/02/2013 11:17:47 AM
Lengths (block/logic/data): 00142 00010 00000

Block: DB20

Address	Name	Type	Initial value	Comment
0.0		STRUCT		
+0.0	STATUS	STRUCT		STATUS WORD 1
+0.0	BIT8	BOOL	FALSE	DEVIATION SETPOINT
+0.1	BIT9	BOOL	FALSE	CONTROL REQUEST
+0.2	BIT10	BOOL	FALSE	MAX SP. REACHED
+0.3	BIT11	BOOL	FALSE	I,M,P LIMIT
+0.4	BIT12	BOOL	FALSE	MOTOR HOLDING BRAKE OPEN
+0.5	BIT13	BOOL	FALSE	ALARM MOTOR TEMP.
+0.6	BIT14	BOOL	FALSE	MOTOR FWD
+0.7	BIT15	BOOL	FALSE	ALARM DRIVE OVERLOAD
+1.0	BIT0	BOOL	FALSE	RDY FOR SWITCH ON
+1.1	BIT1	BOOL	FALSE	READY
+1.2	BIT2	BOOL	FALSE	OPERATION ENABLED
+1.3	BIT3	BOOL	FALSE	FAULT PRESENT
+1.4	BIT4	BOOL	FALSE	COAST DOWN OFF2
+1.5	BIT5	BOOL	FALSE	QUICK STOP OFF3
+1.6	BIT6	BOOL	FALSE	SWITCHING ON INHIBITED ACTIVE
+1.7	BIT7	BOOL	FALSE	ALARM PRESENT
+2.0	ACTUALSPEED	WORD	W#16#0	ACTUAL MOTOR SPEED
+4.0	CURRENT	WORD	W#16#0	ACTUAL MOTOR CURRENT
+6.0	VOLTAGE	WORD	W#16#0	ACTUAL VOLTAGE
+8.0	ERROR	INT	0	ERROR CODE
=10.0		END_STRUCT		
=10.0		END_STRUCT		

DB40 - <offline> - Declaration view

"DRIVE NO:2 STATUS WORD" STATUS WORD
Global data block DB 40
Name: **Family:**
Author: **Version:** 0.1
Block version: 2
Time stamp Code: 11/07/2013 06:03:03 PM
Interface: 11/07/2013 06:03:03 PM
Lengths (block/logic/data): 00138 00008 00000

Block: DB40

Address	Name	Type	Initial value	Comment
0.0		STRUCT		
+0.0	STATUS	STRUCT		STATUS WORD 1
+0.0	BIT8	BOOL	FALSE	DEVIATION SETPOINT
+0.1	BIT9	BOOL	FALSE	CONTROL REQUEST
+0.2	BIT10	BOOL	FALSE	MAX SP. REACHED
+0.3	BIT11	BOOL	FALSE	I,M,P LIMIT
+0.4	BIT12	BOOL	FALSE	MOTOR HOLDING BRAKE OPEN
+0.5	BIT13	BOOL	FALSE	ALARM MOTOR TEMP.
+0.6	BIT14	BOOL	FALSE	MOTOR FWD
+0.7	BIT15	BOOL	FALSE	ALARM DRIVE OVERLOAD
+1.0	BIT0	BOOL	FALSE	RDY FOR SWITCH ON
+1.1	BIT1	BOOL	FALSE	READY
+1.2	BIT2	BOOL	FALSE	OPERATION ENABLED
+1.3	BIT3	BOOL	FALSE	FAULT PRESENT
+1.4	BIT4	BOOL	FALSE	COAST DOWN OFF2
+1.5	BIT5	BOOL	FALSE	QUICK STOP OFF3
+1.6	BIT6	BOOL	FALSE	SWITCHING ON INHIBITED ACTIVE
+1.7	BIT7	BOOL	FALSE	ALARM PRESENT
+2.0	ACTUALSPEED	WORD	W#16#0	ACTUAL MOTOR SPEED
+4.0	CURRENT	WORD	W#16#0	ACTUAL MOTOR CURRENT
+6.0	ERROR	INT	0	ERROR CODE
=8.0		END_STRUCT		
=8.0		END_STRUCT		

FC2 - <offline>

"DRIVE NO:1 CONTROL"

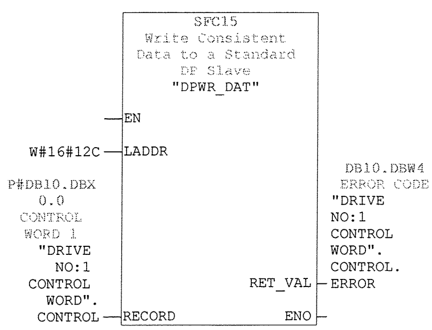
Name: **Family:**
Author: **Version:** 0.1
Block version: 2
Time stamp Code: 11/16/2014 10:28:42 PM
Interface: 10/01/2013 01:26:49 PM
Lengths (block/logic/data): 00404 00292 00016

Name	Data Type	Address	Comment
IN		0.0	
OUT		0.0	
IN_OUT		0.0	
TEMP		0.0	
RETURN		0.0	
RET_VAL		0.0	

Block: FC2 DRIVE NO:1 - SEND DATA TO DRIVE

CONTROL WORD 1 / STATUS WORD 1 - SEND /RECEIVE DATA TO/FROM DRIVE NO:1

Network: 1 DRIVE NO:1 - CONTROL WORD SEND DATA.

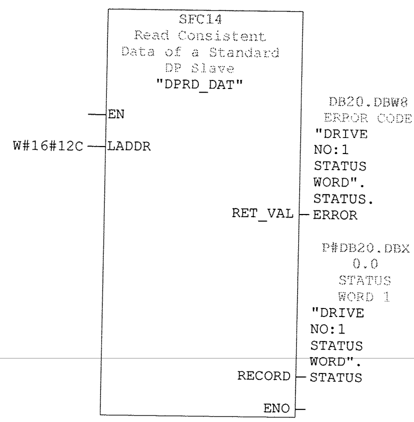


SIMATIC

Bless_final\
SIMATIC 300(1)\CPU 313C-2 DP\...\FC2 - <offline>

11/19/2014 09:56:01 PM

Network: 2 DRIVE NO:1 - STATUS WORD RECEIVE DATA FROM DRIVE NO:1

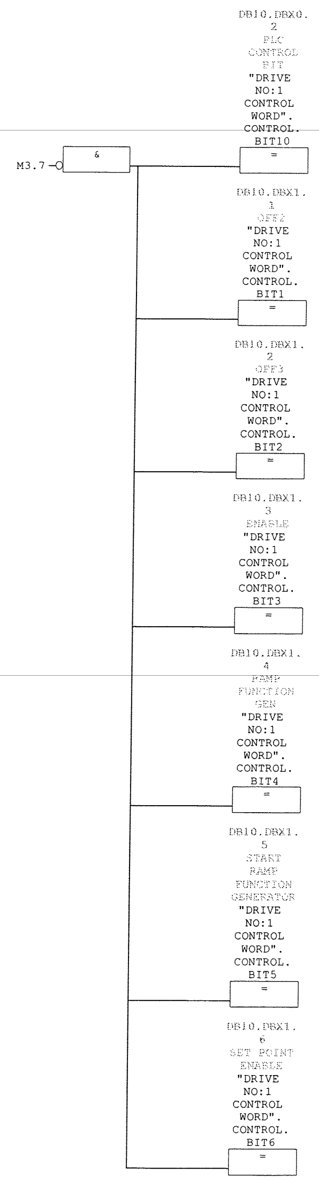


SIMATIC

Bless_final\
SIMATIC 300(1)\CPU 313C-2 DP\...\FC2 - <offline>

11/19/2014 09:56:01 PM

Network: 3   Initialize Drive NO:1



DP ID:	**185**
Module description:	Universal module
Input address:	300
Output address:	300
Slave(8):	**MICROMASTER_420**
Station:	SIMATIC 300(1)
Subnet:	PROFIBUS(1)
PROFIBUS address:	8
Diagnostics address:	1015 ...
Order number:	6SE6400-1PB00-0AA0 (MM4B)
Family:	SIMOVERT
Device type:	MICROMASTER 420
Comment:	
SYNC capability:	Yes
FREEZE capability:	Yes
Response monitoring:	Yes
Preset configuration:	None
Slot:	4
Drive:	
Type:	No PIV
Address:	
PROFIBUS partner:	
Type:	
PROFIBUS address:	
I/O address:	
Length:	
Unit:	
Consistency:	
Comment:	
Slot:	5
Drive:	
Type:	Actual value
Address:	PZD 1
PROFIBUS partner:	
Type:	Input
PROFIBUS address:	2
I/O address:	400
Length:	4
Unit:	Word
Consistency:	Total length
Comment:	
Slot:	6
Drive:	
Type:	Setpoint
Address:	PZD 1
PROFIBUS partner:	
Type:	Output
PROFIBUS address:	2
I/O address:	400
Length:	4
Unit:	Word
Consistency:	Total length
Comment:	
Required optional package:	Drive ES / SIMOTION
Required optional package:	Drive ES / SIMOTION
Required optional package:	Drive ES / SIMOTION
Required optional package:	Drive ES / SIMOTION

DP master system:

Assigned master:

Short description	PA master
Order no.	6ES7 153-2BA82-0XB0
Designation	IM 153 PA-Master
Location	
Station	SIMATIC 300(1)
Rack	0
Slot	4
Receptacle for interface module	0
PROFIBUS address	02

Group: 1
Comment:
The group supports SYNC.
The group supports FREEZE.

Group: 2
Comment:
The group supports SYNC.
The group supports FREEZE.

Group: 3
Comment:
The group supports SYNC.
The group supports FREEZE.

Group: 4
Comment:
The group supports SYNC.
The group supports FREEZE.

Group: 5
Comment:
The group supports SYNC.
The group supports FREEZE.

Group: 6
Comment:
The group supports SYNC.
The group supports FREEZE.

Group: 7
Comment:
The group supports SYNC.
The group supports FREEZE.

Group: 8
Comment:
The group supports SYNC.

The group supports FREEZE.

Slave : FDC 157-0 PROFIBUS address: 3
Assigned to the groups:
Slave : SITRANS P DSIII PROFIBUS address: 6
Assigned to the groups:

Slave (3)	**FDC 157-0**
Order number:	6ES7 157-0AC83-0XA0
Family:	PA
DP slave type:	FDC 157-0
Manufacturer:	SIEMENS
GSD file name:	SI028131.GSE
GSD revision:	5
Identification number:	0x8131
Revision of the DP slave:	V0.0
Hardware release:	A1.0
Software revision level:	Z1.0
Comment:	
PROFIBUS address :	3
Diagnostic address:	1021
SYNC:	Yes
FREEZE:	Yes
Watchdog:	Turned on
Identification	
Plant designation:	
Location designation:	
Installation date:	
Additional information:	

Parameter assignment in ascending byte sequence (hexadecimal specification) :
C4,00,08,08,81,00,00,60,04,04,00

DP ID:	**148**
Module description:	current
Input address:	256

Slave (6)	**SITRANS P DSIII**
Order number:	
Family:	PA
DP slave type:	SITRANS P DSIII
Manufacturer:	SIEMENS AG
GSD file name:	SIEM80A6.GSE
GSD revision:	3
Identification number:	0x80a6
Revision of the DP slave:	3.1.1
Hardware release:	A01
Software revision level:	0300.01.05
Comment:	
PROFIBUS address :	6
Diagnostic address:	1018
SYNC:	No
FREEZE:	No
Watchdog:	Turned on

Parameter assignment in ascending byte sequence (hexadecimal specification) :
00,00,00

DP ID:	**148**
Module description:	AI Output
Input address:	275
DP ID:	**0**
Module description:	Not in cyclic data transfer

DB20 - <offline> - Declaration view

"DRIVE NO:1 STATUS WORD" STATUS WORD 1
Global data block DB 20
Name: **Family:**
Author: **Version:** 0.1
Block version: 2
Time stamp Code: 10/02/2013 11:17:47 AM
Interface: 10/02/2013 11:17:47 AM
Lengths (block/logic/data): 00142 00010 00000

Block: DB20

Address	Name	Type	Initial value	Comment
0.0		STRUCT		
+0.0	STATUS	STRUCT		STATUS WORD 1
+0.0	BIT8	BOOL	FALSE	DEVIATION SETPOINT
+0.1	BIT9	BOOL	FALSE	CONTROL REQUEST
+0.2	BIT10	BOOL	FALSE	MAX SP. REACHED
+0.3	BIT11	BOOL	FALSE	I,M,P LIMIT
+0.4	BIT12	BOOL	FALSE	MOTOR HOLDING BRAKE OPEN
+0.5	BIT13	BOOL	FALSE	ALARM MOTOR TEMP.
+0.6	BIT14	BOOL	FALSE	MOTOR FWD
+0.7	BIT15	BOOL	FALSE	ALARM DRIVE OVERLOAD
+1.0	BIT0	BOOL	FALSE	RDY FOR SWITCH ON
+1.1	BIT1	BOOL	FALSE	READY
+1.2	BIT2	BOOL	FALSE	OPERATION ENABLED
+1.3	BIT3	BOOL	FALSE	FAULT PRESENT
+1.4	BIT4	BOOL	FALSE	COAST DOWN OFF2
+1.5	BIT5	BOOL	FALSE	QUICK STOP OFF3
+1.6	BIT6	BOOL	FALSE	SWITCHING ON INHIBITED ACTIVE
+1.7	BIT7	BOOL	FALSE	ALARM PRESENT
+2.0	ACTUALSPEED	WORD	W#16#0	ACTUAL MOTOR SPEED
+4.0	CURRENT	WORD	W#16#0	ACTUAL MOTOR CURRENT
+6.0	VOLTAGE	WORD	W#16#0	ACTUAL VOLTAGE
+8.0	ERROR	INT	0	ERROR CODE
=10.0		END_STRUCT		
=10.0		END_STRUCT		

DB40 - <offline> - Declaration view

"DRIVE NO:2 STATUS WORD" STATUS WORD
Global data block DB 40
Name: **Family:**
Author: **Version:** 0.1
Block version: 2
Time stamp Code: 11/07/2013 06:03:03 PM
Interface: 11/07/2013 06:03:03 PM
Lengths (block/logic/data): 00138 00008 00000

Block: DB40

Address	Name	Type	Initial value	Comment
0.0		STRUCT		
+0.0	STATUS	STRUCT		STATUS WORD 1
+0.0	BIT8	BOOL	FALSE	DEVIATION SETPOINT
+0.1	BIT9	BOOL	FALSE	CONTROL REQUEST
+0.2	BIT10	BOOL	FALSE	MAX SP. REACHED
+0.3	BIT11	BOOL	FALSE	I,M,P LIMIT
+0.4	BIT12	BOOL	FALSE	MOTOR HOLDING BRAKE OPEN
+0.5	BIT13	BOOL	FALSE	ALARM MOTOR TEMP.
+0.6	BIT14	BOOL	FALSE	MOTOR FWD
+0.7	BIT15	BOOL	FALSE	ALARM DRIVE OVERLOAD
+1.0	BIT0	BOOL	FALSE	RDY FOR SWITCH ON
+1.1	BIT1	BOOL	FALSE	READY
+1.2	BIT2	BOOL	FALSE	OPERATION ENABLED
+1.3	BIT3	BOOL	FALSE	FAULT PRESENT
+1.4	BIT4	BOOL	FALSE	COAST DOWN OFF2
+1.5	BIT5	BOOL	FALSE	QUICK STOP OFF3
+1.6	BIT6	BOOL	FALSE	SWITCHING ON INHIBITED ACTIVE
+1.7	BIT7	BOOL	FALSE	ALARM PRESENT
+2.0	ACTUALSPEED	WORD	W#16#0	ACTUAL MOTOR SPEED
+4.0	CURRENT	WORD	W#16#0	ACTUAL MOTOR CURRENT
+6.0	ERROR	INT	0	ERROR CODE
=8.0		END_STRUCT		
=8.0		END_STRUCT		

FC2 - <offline>

"DRIVE NO:1 CONTROL"

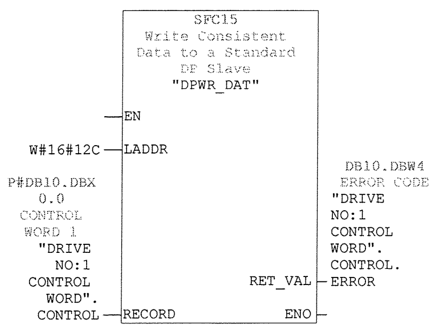
Name: **Family:**
Author: **Version:** 0.1
Block version: 2
Time stamp Code: 11/16/2014 10:28:42 PM
Interface: 10/01/2013 01:26:49 PM
Lengths (block/logic/data): 00404 00292 00016

Name	Data Type	Address	Comment
IN		0.0	
OUT		0.0	
IN_OUT		0.0	
TEMP		0.0	
RETURN		0.0	
RET_VAL		0.0	

Block: FC2 DRIVE NO:1 - SEND DATA TO DRIVE

CONTROL WORD 1 / STATUS WORD 1 - SEND /RECEIVE DATA TO/FROM DRIVE NO:1

Network: 1 DRIVE NO:1 - CONTROL WORD SEND DATA.

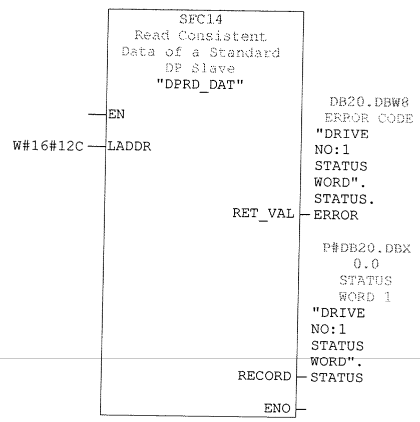


SIMATIC

Bless_final\
SIMATIC 300(1)\CPU 313C-2 DP\...\FC2 - <offline>

11/19/2014 09:56:01 PM

Network: 2 DRIVE NO:1 - STATUS WORD RECEIVE DATA FROM DRIVE NO:1



Network: 3 Initialize Drive NO:1