Development of a WinCC and Profibus Network for Teaching Purposes

Bachelor of Engineering Thesis Project

Prepared by: James Wiggins
Supervisor: A/Prof Graeme Cole
Date: Friday, 13th November 2015
**Declaration**

I James Wiggins, declare that this report contains only work completed by myself except for information obtained in a legitimate way from literature, company or university sources. All information from these other sources has been duly referenced and acknowledged in accordance with Murdoch University’s Policy on Plagiarism.
Abstract

The overall outcome for this thesis project is to establish the use of WinCC, Profibus-DP, and Variable Speed Drives (VSD’s) to the Industrial Computer Systems Laboratory (ISCE Lab) for future students to learn from. A number of students have attempted this project in the past, however due to updated software and loss of programs the project was reattempted with a different approach for current software and student needs.

The project can be separated into three distinct aims:

1. Configure Profibus-DP communication to all key devices used in the ICSE lab.
2. Set-up the WinCC SCADA system to simulate and control a process using all the equipment in the ICSE lab.
3. Create simple instructional documentation for future students to easily be able to connect Profibus devices and set-up WinCC SCADA control systems in the lab.

A total of five unique devices are available in the ICSE lab and were configured which include PLCs, control relays, VSDs, HMIs and a PC with WinCC. Very little documentation was available that detailed how the devices used in this project could be connected together, therefore it was strongly encouraged to focus on developing instructional manuals for future students to learn from, which was the primary aim of this project.

To achieve this task, the connectivity and operation of each device and the Profibus-DP protocol had to be thoroughly understood. From there, each individual connection was made, documented and tested before developing a larger network that incorporated all of the devices simultaneously.

The outcome of this project resulted in the setup of a Profibus-DP network that utilised most of the components, devices and connections that were available in the ICSE laboratory. The final network succeeded in communicating with all the connected devices and provided basic SCADA functionality such as alarm management, data logging, and HMI interfacing. Also achieved was the creation of five
instructional manuals which can allow future students to learn the steps involved in configuring each device to a Profibus-DP network.
# Table of Contents

Abstract .................................................................................................................. i

List of Figures ......................................................................................................... viii

List of Tables .......................................................................................................... x

Acknowledgements ............................................................................................... xi

Glossary .................................................................................................................... xii

Chapter 1 Introduction .............................................................................................. 1

1.1 Objectives .......................................................................................................... 2

Chapter 2 Review of Profibus .................................................................................. 4

2.1 Profibus Overview .............................................................................................. 4

2.1.1 The Fieldbus Protocol .................................................................................. 4

2.1.2 Benefits of Profibus ...................................................................................... 5

2.2 Types of Profibus .............................................................................................. 5

2.2.1 Profibus FMS ............................................................................................... 5

2.2.2 Profibus-DP ................................................................................................. 6

2.2.3 Profibus-PA ................................................................................................. 6

2.2.4 PROFldrive .................................................................................................. 6

2.2.5 PROFIsafe .................................................................................................. 6

2.3 Profibus-DP ....................................................................................................... 6

2.3.1 Network Topology ....................................................................................... 7

2.3.2 OSI Model ................................................................................................... 9

2.3.3 Cabling ....................................................................................................... 10
2.3.4 GSD Files........................................................................................................... 13
2.3.5 Versions................................................................................................................ 13
2.3.6 Profibus-DP Cyclic Data Exchange Function Blocks........................................... 14
2.4 Siemens S7 Protocol................................................................................................. 15
  2.4.1 PG Communication ......................................................................................... 15
  2.4.2 OP Communication ......................................................................................... 15
  2.4.3 Routing ........................................................................................................... 15

Chapter 3 Project Planning & Design ........................................................................... 17
  3.1 Project Hardware ................................................................................................. 18
    3.1.1 Siemens S7-300 PLC .................................................................................... 18
    3.1.2 Siemens RS-485 Repeater ............................................................................ 18
    3.1.3 Siemens HMI Touch Panel ........................................................................... 19
    3.1.4 Moeller Control Relay .................................................................................. 19
    3.1.5 SEW VSD .................................................................................................... 19
  3.2 Project Software ..................................................................................................... 20
    3.2.1 Siemens Totally Integrated Automation Portal (TIA) .................................... 20
    3.2.2 SEW MOVITOOLS-MotionStudio ............................................................. 20
    3.2.3 Siemens WinCC .......................................................................................... 21
  3.3 Network Arrangements ......................................................................................... 21
    3.3.1 Design Arrangement 1 – Single Panel Layout ............................................ 21
    3.3.2 Design Arrangement 2 – Dual Panel Layout ............................................... 22
    3.3.3 Design Arrangement 3 – WinCC Layout ...................................................... 23
### Chapter 4 Siemens SIMATIC S7-300 PLC

#### 4.1 Siemens PLC Overview

<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1.1</td>
<td>CPU</td>
</tr>
<tr>
<td>4.1.2</td>
<td>Digital and Analog I/O Modules</td>
</tr>
<tr>
<td>4.1.3</td>
<td>Communication Modules</td>
</tr>
</tbody>
</table>

#### 4.2 Connecting the PC to the Siemens PLC

<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.2.1</td>
<td>Connecting using MPI Communication</td>
</tr>
<tr>
<td>4.2.2</td>
<td>Connecting using Profibus-DP Communication</td>
</tr>
</tbody>
</table>

#### 4.3 Using Siemens TIA Portal with the PLC

<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.3.1</td>
<td>Device Configuration</td>
</tr>
<tr>
<td>4.3.2</td>
<td>PLC Programming</td>
</tr>
<tr>
<td>4.3.3</td>
<td>PLC Tags</td>
</tr>
<tr>
<td>4.3.4</td>
<td>Watch and Force Tables</td>
</tr>
</tbody>
</table>

#### 4.4 Profibus-DP Organisation Blocks

<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.4.1</td>
<td>Rack Failure [OB 86]</td>
</tr>
<tr>
<td>4.4.2</td>
<td>Diagnostic Interrupt [OB 82]</td>
</tr>
<tr>
<td>4.4.3</td>
<td>Warm Restart [OB 100]</td>
</tr>
</tbody>
</table>

#### 4.5 Configuring the PLC to the Profibus-DP Network

<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.5.1</td>
<td>As a Class 1 Profibus-DP Master</td>
</tr>
<tr>
<td>4.5.2</td>
<td>As a Profibus-DP Intelligent Slave</td>
</tr>
</tbody>
</table>

### Chapter 5 Moeller EASY719-DC-RC Control Relay

#### 5.1 Moeller Control Relay Overview
Chapter 9 Testing and Results ......................................................... 58

9.1 Testing the Connections ......................................................... 59

9.2 The Final Network ................................................................. 61

Chapter 10 Proposed Future Work .................................................. 68

10.1 Profibus-DP-V1 ................................................................. 68

10.1.1 DP-V1 Function Blocks .................................................. 68

10.2 Danfoss HVAC FC-102 VSD ................................................. 69

Chapter 11 Conclusion ................................................................. 71

Appendices .................................................................................. 72

Works Cited .................................................................................. 73

List of Figures

Figure 2.1: Profibus-DP Master-Slave Bus Topology ................................ 7

Figure 2.2: Centralised vs. Decentralised Network Configuration .............. 9

Figure 2.3: OSI Model for Profibus-DP ........................................... 9

Figure 2.4: Profibus-DP Type A Shielded Twisted Pair Cable .................... 11

Figure 2.5: Profibus 9-Pole D-type Plug Connector ................................ 12

Figure 2.6: Routing Subnets for S7 Devices ........................................ 16

Figure 3.1: Profibus-DP Test Panel ................................................ 17

Figure 3.2: Single Panel Layout Design ........................................... 22

Figure 3.3: Dual Panel Layout Design ............................................. 23

Figure 3.4: WinCC Layout Design .................................................. 24

Figure 4.1: Siemens S7 CPU314C-2 DP PLC .................................... 25

Figure 4.2: USB to MPI Adapter ................................................... 27
Figure 5.1: Moeller EASY719-DC-RC

Figure 5.2: Moeller EASY204-DP Profibus-DP Gateway

Figure 5.3: Moeller EASY-PC-CAB Cable

Figure 6.1: SEW MOVITRAC B VSD

Figure 6.2: SEW R17 DR63M4 Motor

Figure 6.3: Wiring Diagram for the SBus Network between SEW Devices

Figure 6.4: SEW DFP21B Profibus-DP Gateway

Figure 7.1: Siemens TP 177B 6" PN/DP HMI Touch Panel

Figure 7.2: TP 177B 6" PN/DP Communication Ports

Figure 7.3: Root Screen Template for HMI Touch Panel

Figure 8.1: WinCC RT Professional Online Trend Control

Figure 9.1: Final network created in Siemens TIA Portal

Figure 9.2: Watch table to monitor the connection between the master PLC and Moeller control relay

Figure 9.3: Watch table to monitor the connection between the master PLC and SEW VSD

Figure 9.4: Screen created on the Siemens HMI touch panel to control/monitor the Moeller’s operating mode

Figure 9.5: Screen created on the Siemens HMI touch panel to control/monitor the Moeller’s “S” and “R” inputs and outputs

Figure 9.6: Screen created on the PC using Siemens’ WinCC RT Professional to control Moeller’s basic functions

Figure 9.7: HMI screen created using Siemens’ WinCC RT Professional to control the SEW VSD’s basic functions

Figure 9.8: HMI screen created using Siemens WinCC RT Professional to control/monitor the simulated process

Figure 9.9: Alarm created using WinCC RT Professional to trigger when the simulated tank level exceeded 80%

Figure 9.10: Data log created using WinCC RT Professional to monitor the simulated process

Figure 10.1: Danfoss HVAC FC-102 VSD
List of Tables

Table 2.1: Technical Specifications of the Profibus-DP Type A Cable [16] [20] ___________________________ 11
Table 2.2: Maximum Length of Type A Profibus-DP Cable for Configured Baud Rates [20] ________________ 12
Table 5.1: Modules Configured to EASY719-DC-RC [35]_________________________________________ 37
Table 6.1: SEW R17 DR63M4 Technical Specifications _____________________________________________ 41
Table 6.2: FSC11B SBus Terminal Description [39] _______________________________________________ 42
Table 6.3: Configured SBus Parameters in MOVITOOLS___________________________________________ 44
Table 6.4: Modules Configured to SEW MOVITRAC B [40]_______________________________________ 46
Acknowledgements

For their assistance in this project, acknowledgements are given to the following individuals:

Project Supervisor: Associate Professor Grame R Cole,

Lecturer, Murdoch University

Mr Lafeta ‘Jeff’ Laava,

Technical Officer, Murdoch University

Mr Will Stirling,

Technical Officer, Murdoch University
Glossary

AC – Alternating Current

CAN – Controller Area Network

CPU – Central Processing Unit

DB – Data Block

DB-9 – Dsub 9 pin communication port

DC – Direct Current

DIN Rail – Metal rail used for mounting industrial control equipment

FB – Function Block

FBD – Function Block Diagram

FC – Function Call

GSD – General Station Description File

HMI – Human Machine Interface

Hz – Hertz

ICSE – Industrial Computer System Engineering

I/O – Input/Output

I-Slave – Intelligent Slave

Kbps – Kilobits per second

LAD – Ladder Diagram

LED – Light Emitting Diode
MB – Mega-Byte

Mbps – Megabits per second

MPI – Multi-Point Interface

OB – Organization Block

OP Communication – Operator Control and Monitoring Communication

PC – Personal Computer

PCI – Peripheral Component Interconnect

PG Communication – Programming Device Communication

PLC – Programmable Logic Controller

Profibus – Process Fieldbus

Profibus-DP – Profibus Decentralized Peripherals

Profibus-PA – Profibus Process Automation

Profinet – Process Ethernet

RPM – Revolutions per minute

SBus – System Bus

SCADA – Supervisory Control and Data Acquisition

SCL – Structured Control Language

SIMATIC – Siemens automatic

STL – Statement List

TIA Portal – Totally Integrated Automation Portal
USB – Universal Serial Bus

VSD – Variable Speed Drive

WinCC RT – WinCC Runtime
Chapter 1 Introduction

This report documents the progress made on the development and integration of Siemens’ WinCC and Profibus systems into the Industrial Computer Systems Facility as part of the Bachelor of Engineering thesis project. The main aspects of this project are:

- Setting up a Profibus network.
- Configuring various devices such as Programmable Logic Controllers (PLCs), Human Machine Interfaces (HMIs) and Variable Speed Drives (VSDs) to the network.
- Simultaneously control and monitor the devices on the network.
- Setup and configure a WinCC Supervisory Control and Data Acquisition (SCADA) system to the network.
- Construct guides to teach inexperienced users the basics of setting up and configuring Profibus and WinCC systems.

The completion of the guides was the primary aim of this project as the overall outcome will be to incorporate Profibus and WinCC SCADA systems into the facility for teaching and demonstration purposes. The progress made towards this will be discussed for each device that was implemented in the project. Configuring the devices to the network required in-depth research into the functionality of the devices, including the interfaces that allowed Profibus to interact with the device. The following sections of this report documents the technical review of Profibus, the overall design and planning of the project, the description and configuration of each device used in the network, the completed network, and also the future work that can be added to the project.
1.1 Objectives

Before commencing, time was taken at the beginning of the project to deconstruct the project so that all the tasks and requirements could be identified and understood in order to correctly plan an approach to the project. Developing a set of objectives allowed a rough estimate of what was needed to complete the project, and also establish what would be a reasonable time frame for each objective to be completed. After meeting with the project supervisor and revising previous student’s work on the project, the following objectives were established:

- Understand and research Profibus-DP communication such as protocols and cabling.
- Using Totally Integrated Automation (TIA) Portal, configure a Siemens S7-300 PLC as a master device on a Profibus network.
- Understand and research operation of the Siemens TP177B HMI touch panel and configure to the Profibus network.
- Locate and install the required General Station Description (GSD) files for the Profibus interface modules used in the project.
- Understand and research operation of the Moeller Easy719 and the Easy204-DP and configure to the Profibus network as a slave device.
- Understand and research operation of the SEW MOVITRAC B VSD and the DFP21B interface module and configure to the Profibus network as a slave device.
- Understand and research Siemens SIMATIC WinCC SCADA software and configure a simple server to the Profibus network.
- Understand and research operation of the RS485 Repeater and configure to the Profibus network.
- Develop a program that incorporates all of the required devices on a single Profibus network and simulates a realistic plant environment for demonstration purposes.
- Construct configuration guides to assist inexperienced users in configuring the Profibus and WinCC systems for teaching purposes.
For the project to be completed successfully, all the above objectives must be completed to a sufficiently high standard and finalised with a detailed demonstration, presentation and report.
Chapter 2  Review of Profibus

This chapter contains information about the Profibus protocol which was used to configure industrial automation devices in a network as part of the project. Prior to commencing the network configuration, some background and useful information about how the protocol works was first discovered. The following sections of this chapter document a brief overview about Profibus in general, the types of Profibus available, and an in-depth look at the Profibus-DP protocol which was primarily used for the project.

2.1 Profibus Overview

Profibus is a data communication protocol that originated from a group of German companies in the late 1980’s [1]. The name is derived from Process Fieldbus [3], which is due to the protocol being based on Fieldbus technology and was primarily developed for process automation applications. Compared to other technologies used in process automation, Profibus is very fast, easier to configure, and uses bus topology that can reduce the amount of wiring on a single network which led to the success and popularity of Profibus [1]. Over 1400 companies are now members of the largest automation community, PROFIBUS & PROFINET International (PI) [4], which has promoted the expansion and development of the protocol and made it a part of the international standard IEC 61158 for industrial communication networks [5].

2.1.1 The Fieldbus Protocol

The Fieldbus protocol was developed for the purpose of simplifying industrial communication networks. Prior to Fieldbus, each field device had to be individually connected to the controller [8]. Fieldbus allowed multiple field devices to be connected to the controller along a single cable thereby significantly reducing the amount of cable in the network [8]. Data is transmitted in small packets through serial communication which also is able to transmit over greater distances than point-to-point or parallel communication [8]. Fieldbus was internationally standardised as IEC 61158 and its core structure has since been used in a wide range of protocols including Profibus [9].
2.1.2 Benefits of Profibus

There are many fieldbus technologies available, however Profibus was selected for this project due to the following facts:

- It is the most widely used Fieldbus communication protocol worldwide therefore has the most support, availability and product choice [3] [16].
- It can support hybrid communication between factory, process automation, motion control and safety tasks from the same bus, reducing the cost of cabling and commissioning [2] [16].
- Devices can be swapped without bringing down the whole network [6]. Reduces the amount of plant downtime.
- Easy to integrate with other protocols such as HART and Profinet expanding the capabilities of the network [6].
- It is designed to transmit diagnostic messages about the status of the devices on the bus to assist with fault finding, commissioning and maintenance [6].

2.2 Types of Profibus

Different plant operations sometimes require different communication methods. The Profibus protocol has been designed to incorporate different applications through compatible versions of Profibus. The following briefly describes each type of Profibus available for unique applications.

2.2.1 Profibus FMS

Profibus Fieldbus Message Specification (FMS) was the first version of Profibus to be developed [1]. The purpose of Profibus FMS was to transfer advanced data from PCs to controllers [1]. Profibus FMS can still be found in older networks but advancements in the Profibus protocol allowed for a more robust version to be developed [1].
2.2.2 Profibus-DP
Profibus Decentralised Periphery (DP) is the most popular Profibus protocol in the market today [1]. This protocol is used for high speed discrete communication such as the data transfer between controllers and distributed I/O [3] [10]. Profibus-DP is the fastest Fieldbus protocol able to transmit 244 bytes of data at 9.6 Kbps up to 12 Mbps which is suitable for most applications [3].

2.2.3 Profibus-PA
Profibus Process Automation (PA) used the same protocol as Profibus-DP but over Manchester Encoded Bus Powered (MBP) technology [3]. Profibus-PA is used for continuous process automation applications such as sensors and actuators and is safer for use in hazardous areas. The transmission rate of Profibus-PA is always 31.25 kbps.

2.2.4 PROFIdrive
PROFIdrive is an expansion to the Profibus-DP protocol that has been optimised for motion control applications [11]. Typical applications include high speed communication from the controller to Variable Speed Drives (VSD's) and positioning devices such as servo motors [11]. PROFIdrive has been standardised in IEC 61800-7 making it available for a majority of motion control applications and vendors [11].

2.2.5 PROFIsafe
Similar to PROFIdrive, PROFIsafe is an expansion to the existing Profibus protocol [12]. PROFIsafe is designed to communicate messages through the safety systems within the network, without disrupting the standard network communications [12]. It has been defined as an international standard in IEC 61784-3-3 and integrates safety with process communication on the same cable [13].

2.3 Profibus-DP
As mentioned, Profibus-DP is used for high speed discrete communication between controllers such as PLCs and distributed I/O devices such as HMIs, control relays and VSDs, hence Profibus-DP was the protocol chosen for this project. For this project, Profibus-DP was the protocol used to setup and
configure a network containing multiple PLCs, HMIs, Control Relays and VSDs. The first stage of the project was to gain an understanding of the protocol and how it is used as a network for industrial computer systems. The following sections describe the Profibus-DP topology, OSI model, cabling, GSD files and versions.

2.3.1 Network Topology
Profibus-DP uses a master-slave bus topology [14]. The bus is a single cable that runs from one end of the network to the other with each device connected in parallel to the bus, as seen in Figure 2.1. Since all slave devices are connected to the bus, if a slave device fails the other devices on the bus are unaffected. Every device on the Profibus-DP network must have a unique address from 0 to 126 for the device to receive communication [14]. Address 126 has a special function called “Set Slave Address” which is used by slaves that use software address settings [14]. There is also an address 127 on the bus, but is reserved as a broadcast address for use by a bus monitor [14]. Slave devices on the Profibus-DP network can be “hot swapped”, meaning as long as the device is configured in the master PLC, the slave devices can be added or removed while the network is powered [14]. When the master PLC detects that a message does not reach its intended slave, the message is repeated [7].

![Figure 2.1: Profibus-DP Master-Slave Bus Topology](image-url)
The Profibus-DP network requires a Class 1 master to send and receive commands to the slave devices [14]. A bus on the network can only have one Class 1 master configured [15]. In addition to the Class 1 master is the Class 2 master. If a Class 2 master is also configured to the bus, “token” communication is performed between the masters [15]. This means that once the Class 1 master has performed its scan cycle, it passes a token to the Class 2 master giving it temporary control over the bus [15]. For Profibus-DP, there can never be two masters in control at the same time [15]. Class 2 masters are generally configured as supervisors for the network; however multiple masters will slow down the overall scan sequence due to only one being able to perform at a time [15] [16].

The scan cycle refers to the execution of the program stored on the PLC. The code is read in a particular order by the PLC (the scan), usually from top to bottom and left to right. Once the scan is executed, the program is repeated (the cycle) with updated values from the previous scan. This procedure is hence called the scan cycle.

Profibus-DP is a decentralised network, meaning each node on the network can perform functions that are not controlled by the master [17]. A decentralised network is made up of multiple subsystems collecting and sending only specific data that benefits the network as a whole [17]. A centralised network has all the nodes communicating to the master on the network [17]. A comparison of the network configurations can be seen in Figure 2.2 with the white circles representing the network masters, and the black circles representing the network slaves. Decentralised networks benefit by simplifying the programming on the master, and separating the programs to specific areas or processes [17]. Decentralised networks are also more expandable, and offer higher productivity in control systems [17].
2.3.2 OSI Model

The Open System Interconnection (OSI) model represents the framework of the communication protocol [18]. There are seven layers in the OSI model [18], Profibus-DP makes use of three of these layers [1]. A representation of the model is shown in Figure 2.3.

Layer 1 represents the physical layer of the protocol [18]. In other words, it is the hardware used to transmit data for the protocol. Profibus-DP generally uses the RS-485 physical layer but can also be transmitted through Fiber-Optic cable for long distances [1] [14]. For this project RS-485 was used. RS-485 supports true multi-point communication, meaning multiple nodes can send and receive data on the same bus [19]. RS-485 and Profibus-DP can have up to 32 devices on a single bus segment.
[19], which can be extended to a maximum of 126 devices with the use of repeaters [15]. To prevent “data collisions”, Profibus-DP and RS-485 use a “half-duplex” communication system [19]. This is where all slave devices remain in “receive mode” until addressed by the master which triggers the selected slave device to transmit a response back to the master [19]. It is for this reason only one device can transmit at a time.

Layer 2 of the OSI model represents the data link of the protocol [18]. The data link layer establishes the permissions for data to transmit, synchronisation, and error checking [18] [20]. This is where the Fieldbus protocol lies in the Profibus structure. The data link for the Profibus protocol is called the Field Bus Data Link (FDL) [1]. FDL ensures that only one device can transmit at a time [20]. This is done through master-slave and token passing communication [20]. For the master-slave network, the addressed slave can only transmit a response when requested by the master [20]. For networks with more than one master, the token passing principle is used. This involves a special portion of data (the token) which is sent between masters to allow temporary control over the bus [20]. The token is passed in a logical ring sequence, which is in ascending order of address from the first master to the last master and repeated [20].

Layer 7 of the OSI model represents the application of the protocol [18] and is the final layer defined by Profibus-DP [1]. Profibus supports a wide range of applications, some of which include: cyclic data exchange, acyclic data exchange, isochronous data exchange, alarm-handling, diagnostics, PROFIdrive and PROFIsafe [1] [14].

2.3.3 Cabling

Profibus has a standardised cable for transmitting data in a Profibus-DP network. For this project, the RS-485 physical layer was used. Two RS-485 Profibus-DP cables have been developed, Type A and Type B [20]. Type B was the first cable released but after developments in the protocol, it was upgraded to the new Type A cable. The technical specifications of the Profibus-DP Type A cable are shown in Table 2.1 which was the cable that was used in this project.
Table 2.1: Technical Specifications of the Profibus-DP Type A Cable [16] [20]

<table>
<thead>
<tr>
<th>Specification</th>
<th>Specification Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Impedance</td>
<td>35 to 165 Ohm at frequencies from 3 to 20 MHz</td>
</tr>
<tr>
<td>Capacitance</td>
<td>&lt; 30 pF per metre</td>
</tr>
<tr>
<td>Core Diameter</td>
<td>0.64 mm</td>
</tr>
<tr>
<td>Core Cross Section</td>
<td>&gt; 0.34 mm$^2$</td>
</tr>
<tr>
<td>Resistance</td>
<td>&lt;110 Ohm per km</td>
</tr>
<tr>
<td>Signal Attenuation</td>
<td>Max 9 dB over total length of line section</td>
</tr>
<tr>
<td>Shielding</td>
<td>Copper braid and foil shield</td>
</tr>
</tbody>
</table>

The standard RS-485 Profibus-DP Type A cable is a shielded twisted pair cable which outer casing is violet in colour, as shown in Figure 2.4 [20]. The two copper wires within the cable are called wire A (green) and wire B (red) which transmit the Profibus-DP protocol [20]. Profibus cables are designed for industrial environments therefore are highly insulated and shielded. It is important to also ground the shielding to optimally prevent any electromagnetic interference that may cause signal errors [16].

![Figure 2.4: Profibus-DP Type A Shielded Twisted Pair Cable](image)

It is important to have bus termination resistors at either end of the bus to prevent the occurrence of reflections. Reflections are more likely to occur from high transfer rates and distances, and can disrupt data signals resulting in communication errors [20]. Ideally a resistor that is equivalent to the wave impedance will cancel out any reflections [20]. For this project, Profibus 9-pole D-type plug connectors as shown in Figure 2.5 were used in the installation of the network, and had in-built line termination resistors. The connectors allow two cables to be attached, one through the A1/B1...
terminals and another through the A2/B2 terminals. It is important to note however, that for these connectors only the A1/B1 terminals can be terminated. Turning the bus termination on will prevent any signals travelling through the A2/B2 terminals [20].

Profibus-DP support baud rates from 9.6 kbps to 12 Mbps [20]. The baud rate also determines the maximum length of cable the signal can travel. Low baud rates are capable of travelling greater distances than high baud rates. Table 2.2 shows the maximum length of cable a signal can travel at the configured baud rate. A repeater can be installed to transmit greater distances [20].

Table 2.2: Maximum Length of Type A Profibus-DP Cable for Configured Baud Rates [20]

<table>
<thead>
<tr>
<th>Baud Rate in Kbps</th>
<th>Max Length of Type A Cable</th>
</tr>
</thead>
<tbody>
<tr>
<td>9.6</td>
<td>1200 m</td>
</tr>
<tr>
<td>19.2</td>
<td>1200 m</td>
</tr>
<tr>
<td>45.45</td>
<td>1200 m</td>
</tr>
<tr>
<td>93.75</td>
<td>1200 m</td>
</tr>
<tr>
<td>187.5</td>
<td>1000 m</td>
</tr>
<tr>
<td>500</td>
<td>400 m</td>
</tr>
<tr>
<td>1500</td>
<td>200 m</td>
</tr>
<tr>
<td>3000</td>
<td>100 m</td>
</tr>
<tr>
<td>6000</td>
<td>100 m</td>
</tr>
<tr>
<td>12000</td>
<td>100 m</td>
</tr>
</tbody>
</table>
2.3.4 GSD Files

The General Station Description (GSD) file is required for any device connected to the Profibus network [20]. If a device supports Profibus communication then there must also be a GSD file for the selected device [16]. GSD files are created by the device manufacturer and significantly assist with the configuration of the device. The GSD file contains parameter and configuration data for use with the device on a Profibus network such as data length, transmission speed, I/O data, address allocation, and bus parameters [15] [20]. This data is used by the Profibus master and is configured automatically during start-up to ensure correct transmission and data procedures [20]. The GSD file will usually come with the device or can be downloaded from the manufacturer’s website [23].

2.3.5 Versions

Through the popularity and development of Profibus-DP, different versions of the protocol have arisen. There have currently been three versions of Profibus-DP, each adding a new function and therefore expanding the capabilities of the protocol. Each version is backwards compatible, so new Profibus-DP configuration software will install the latest version to the network. The following versions of Profibus-DP and its functions have been developed:

**DP-V0:** This was the first version of Profibus-DP developed and uses cyclic data exchange to perform the main functions of the device. During start-up the master performs an initiation stage which sends the default parameters to the slave devices, and configures the length of data to be transmitted as defined by the slave’s GSD file. The master uses cyclic data exchange to send and receive the I/O process data to each slave and also check diagnostic data. The cycle time of the exchange depends on the bit rate and number of slaves on the network.

**DP-V1:** Version 1 of Profibus-DP added the ability to perform acyclic data exchange. Acyclic data exchange is mainly used for adjusting a device’s parameters [20]. Acyclic data exchanges whole data records between the master and slave, up to a total of 240 bytes per record [21]. The exchange takes
place only when the master has requested data from the slave instead of being continuously polled like in cyclic exchange [21].

**DP-V2**: Version 2 is the latest version of Profibus-DP. DP-V2 adds the ability to perform isochronous data exchange [22]. This is used for time critical applications such as motion control, slave-to-slave communication and time synchronisation [22].

### 2.3.6 Profibus-DP Cyclic Data Exchange Function Blocks

The PLC programming format used in this project conformed to the methods presented in the “Profibus Guidelines for Profibus Communication and Proxy Function Blocks according to IEC 61131-3” [36]. The guidelines referred to two standardised function blocks used to send and receive cyclic data between a Profibus-DP master and slave. The blocks that were used were called GETIO and SETIO which were available in the TIA function block library. The blocks were configured to send and receive the data transmitted across the configured addresses for each module. The function blocks store the data using the PLC’s process image [36]. The two methods of transferring data to remote I/O from the PLC are through the process image or through peripheral addressing. The process image is an area of memory located in the PLC’s CPU that updates at the start of every scan cycle [36]. With peripheral addressing the data is sent directly to the bus [37]. The advantage of using the process image is that the data can be accessed much faster than peripheral addressing. On the contrary, the amount of memory allocated to the process image for the CPU 314C-2 DP is only 128 bytes compared to the total address area of 1024 bytes [28].

Using the GETIO and SETIO process image, the data sent and received is stored in local memory on the PLC by configuring the function blocks to point at chosen markers. The PLC’s main scan cycle and the cyclic exchange of data through Profibus-DP are occurring at different rates. The GETIO and SETIO function blocks can be executed in OB 1 on the PLC due to the actual transfer of data to the modules occurring through the cyclic Profibus-DP transmission. In addition to the transfer of the
main process data, the GETIO and SETIO function blocks also receive diagnostic information to identify errors that may occur.

2.4 Siemens S7 Protocol

Communication has been simplified for exchanging data between Siemens S7 devices and PC’s with Siemens software due to the in-built S7 protocol. The S7 protocol is designed in the application layer of the OSI model and is designed for use with Profibus and Multi-Point Interface (MPI) communication protocols [24]. MPI is a protocol designed by Siemens for use with S7-300/400 PLC’s [25]. With the use of a PC to MPI adapter, the PLC can connect to a PC more simply than other communication method such as Profibus. MPI uses a similar protocol to Profibus, allowing the two to be compatible on the same network. The PC to MPI adapter has a fixed baud rate of 187.5 Kbps and uses a master-slave system where the highest address is 126.

2.4.1 PG Communication

Programming device (PG) communication is used for data exchange between engineering stations such as PCs and Siemens SIMATIC PLCs [28]. Functions include downloading programs and configuration data to the PLC, and processing diagnostic information [28]. PG communication can be transmitted over both MPI and Profibus-DP protocols [28].

2.4.2 OP Communication

Operator control and monitoring (OP) communication is used for data exchange between operator stations and Siemens SIMATIC PLCs [28]. Functions include online monitoring and control of a PLC from a PC [28]. OP communication can be transmitted over both MPI and Profibus-DP protocols [28].

2.4.3 Routing

The S7 protocol uses PG/PC communication to connect the Personal Computer (PC) to a SIMATIC programming device (PG) over different subnets [28]. For example, a PC can connect to a SIMATIC
device on a Profibus-DP subnet even though the PC is using an MPI subnet, provided both PC and device are connected to a SIMATIC PLC [28]. Programs and configuration data can be downloaded from the PC to the device through the PLC as a gateway as shown in Figure 2.6 [28]. The PLC must be routing compatible to act as a gateway between subnets [28].

Figure 2.6: Routing Subnets for S7 Devices
Chapter 3 Project Planning & Design

The project required the connection of various devices of different vendors to the Profibus-DP network. A total of five different devices were to be configured to the network, each with unique configurations and functionality. A project design had to be developed that would fit the time constraints of the project. It was therefore decided to prioritise focus on the primary functions of each device and their configuration. To assist with the project development and network testing, a test panel as shown in Figure 3.1 was commissioned in the Mechatronics Laboratory (room PS2.033 at Murdoch University). The test panel was used to develop the hardware and software configurations as required by the project. The following sections identify the hardware and software that was used throughout the project and the Profibus-DP network arrangements that were designed to be implemented to the Industrial Computer Systems Engineering Laboratory (room PS2.027 at Murdoch University).

![Figure 3.1: Profibus-DP Test Panel](image-url)
3.1 Project Hardware

One of the main project objectives was to discover Profibus-DP configurations for the devices used in the project. The types of devices used include PLCs, control relays, HMI, VSDs, gateways and repeaters. Each device is used for industrial automation purposes and when networked together, can be used to simulate a process plant or factory with SCADA functionality. The overall Profibus-DP network was to be configured using the following devices:

- Siemens S7-300 PLC
- Siemens RS-485 Repeater
- Siemens HMI Touch Panel
- Moeller EASY719-DC-RC Control Relay
- Moeller EASY204-DP Profibus-DP Gateway
- SEW Single Phase MOVITRAC B VSD
- SEW DFP21B Profibus-DP Gateway

3.1.1 Siemens S7-300 PLC

The PLC used in this project was the Siemens S7-314C-2 DP with article number 6ES7 314-6CG03 [26]. This PLC was configured as the Profibus-DP master. All other devices were connected to the PLC via the Profibus-DP protocol as slave devices. Communication between the PC and the PLC was able to be obtained via the Profibus-DP or MPI protocols. An additional S7-314C-2 DP PLC was also configured as a Profibus-DP intelligent slave (I-slave).

3.1.2 Siemens RS-485 Repeater

The repeater used in this project was the Siemens RS-485 Repeater, model number 6ES7972-0AA02-0XA0 [31]. A repeater is required if the Profibus-DP cable gets too long or has more than 32 device connected. Repeaters are able to extend the transmission distance and also the amount of devices on the bus. Another useful feature of the repeater is bus segmentation. Segmentation can reduce the amount of cable used by creating an extra branch within the network instead of a single linear line. If
a problem occurs, segmentation also allows bus segments to be terminated at the repeater instead of disabling the entire network.

3.1.3 Siemens HMI Touch Panel

The HMI touch panel used in this project was the Siemens TP 177B 6” PN/DP [30]. HMIs are useful in creating a user interface for a control system. Data collected by the PLC is difficult to monitor alone. The HMI adds graphical representation of the data to make it more identifiable to the user. Touch panels can be mounted in industrial environments where a PC may not be suitable.

3.1.4 Moeller Control Relay

The control relay used in this project was the Moeller EASY719-DC-RC [34]. This device was not originally designed to communicate using Profibus-DP. To connect the control relay to the Profibus-DP network, the Moeller EASY204-DP Profibus-DP gateway [35] was also required. Gateways are used to convert between different communication protocols. The control relay uses ladder logic programming to make use of inputs, outputs, timers and counters.

3.1.5 SEW VSD

The VSD used in this project was the SEW Eurodrive MOVITRAC B [39]. The SEW MOVITRAC B VSD was originally designed to communicate using a CAN based SBus protocol. To connect the VSD to the Profibus-DP network, the SEW DFP21B Profibus-DP gateway [40] was also required. The DFP21B device works as a gateway between the Profibus and the SBus networks. The SBus topology is very similar to Profibus, as it allows up to eight MOVITRAC B VSDs to be daisy-chained to a single DFP21B module. Like Profibus needing to be configured with unique addresses, the VSDs connected to the SBus network also need their own addresses configured. The gateway allows data to be sent from the PLC using Profibus-DP, to access the VSDs functions. Available functions include starting, stopping and changing the set-point speed of the VSD.
3.2 Project Software

In order to configure the hardware to the network, important software was also required. The majority of programs and software configurations were developed using Siemens’ Totally Integrated Automation (TIA) Portal. To configure the slave devices, the GSD files were required. The following GSD files were obtained via the manufacturers’ websites:

- Moe4d10.gsd was used to configure the EASY204-DP with the EASY719-DC-RC.
- SEW_6009.gsd was used to configure the DFP21B with the MOVITRAC B.

3.2.1 Siemens Totally Integrated Automation Portal (TIA)

TIA portal is a programming and configuration software package designed for Siemens SIMATIC devices [32]. As the name suggests, the software was aimed to totally integrate as many automation technologies into one programming environment [32]. The software is designed using an engineering framework that integrates controllers, HMIs, motion control, drives, distributed I/O and motor management application profiles [32]. TIA Portal also consists of a large function and device library capable of performing the majority of procedures used in automation applications [32]. Most of the functions are in a standardised form, allowing more familiarity for users performing engineering tasks and conformity between programs [32].

The GSD files are installed in TIA Portal, which allows the devices to be identified. Only two GSD files were required due to the devices being different manufacturers, the Siemens software was already able to identify other Siemens devices. The Profibus-DP network configurations, addresses, slave process data and PLC code was all developed in TIA Portal and downloaded to the master PLC.

3.2.2 SEW MOVITOOLS-MotionStudio

For the SEW MOVITRAC B VSD, the parameters of the drive were first configured using SEW MOVITOOLS-MotionStudio software. Communication between the PC and VSD was made using a USB to RS-485 adapter. Configuration of the SBus parameters allowed the MOVITRAC B to connect to the DFP21B Profibus-DP gateway.
3.2.3 Siemens WinCC

To configure the PC as a HMI with data logging capabilities, Siemens WinCC Runtime Professional software was used. WinCC RT Professional is a standalone software package that integrates into TIA Portal. Programs for the HMI touch panel were also developed using a different version of WinCC that was preinstalled on the HMI itself.

3.3 Network Arrangements

One of the project goals was to implement Profibus-DP communications to the Industrial Computer Systems Engineering (ICSE) Laboratory (room PS2.027 at Murdoch University) for teaching and demonstration purposes. Due to the layout of the laboratory, it is possible to implement different network arrangements. The laboratory has eight identical panels which are used as workstations by students. After discussions with the supervisor, three design arrangements were created at the beginning of the project. These arrangements were important to the development of the project, as each arrangement was an expansion of the last and required additional network configurations.

3.3.1 Design Arrangement 1 - Single Panel Layout

Once sufficient testing and research has been achieved on the test rig, the developed configurations and programs will need to be carried over to the ICSE laboratory during non-occupied time periods. The first arrangement is for users to operate the components on their individual panels. There are a total of eight individual panels in the ICSE lab which are configured identically. For each single panel, communication from the PC to the Siemens PLC is achieved through Multi-Point Interface (MPI) protocols which are made through the use of a Universal Serial Bus (USB) to MPI adapter. Thereafter the remaining components are connected through Profibus-DP communication as shown in Figure 3.2.
3.3.2 Design Arrangement 2 – Dual Panel Layout

The second design option is to have two panels connected and operated from a single computer. This involves expanding the previous design to incorporate a second Siemens PLC and the components from the second panel. Due to limitations on the number of SEW DFP21B modules, only one can be used for every two panels operated. In this design, the computer communicates with the master Siemens PLC using MPI, and all components thereafter are connected via Profibus-DP communication. The SEW VSDs communicate with SBus protocol which allows up to eight MOVITRAC B VSDs to connect to a single DFP21B Profibus-DP interface module. This layout is shown in Figure 3.3.
3.3.3 Design Arrangement 3 – WinCC Layout

The final design option for this project was to expand the single or dual panel configurations to incorporate an additional computer programmed with Siemens SIMATIC WinCC SCADA software.

Figure 3.4 shows the designed floor plan of the ICSE laboratory incorporating the WinCC system with the dual panel layout as four identical networks. In this design, the computers on the side of the room where the entry and exits are located will be configured with WinCC and are connected to multiple panels using the Profibus cables that run through the ceiling. WinCC has a programmable HMI format that allows the process to be visualised based on the sent and received data from physical hardware. Although all three design arrangements are required in order to complete the project, it is this final design that incorporates the majority of components in the lab and will be used for the final demonstration of the project.
Figure 3.4: WinCC Layout Design
Chapter 4 Siemens SIMATIC S7-300 PLC

This chapter aims to provide information about the Siemens S7-300 PLC that was configured to the Profibus-DP network as shown in Figure 4.1. The S7-300 PLC had two configurations; as a class 1 DP master, and as an intelligent slave (I-slave). For the I-slave configuration, an additional S7-300 PLC was added to the network. To implement the PLC within the Profibus-DP network, both hardware and software configurations had to be developed. The following sections in this chapter aim to describe the purpose of each configuration step.

For the complete list of steps taken to configure the Siemens S7-300 PLC to the Profibus-DP network, please refer to Appendix A: Siemens S7-300 PLC & RS-485 Repeater Profibus Configuration Guide.

Figure 4.1: Siemens S7 CPU314C-2 DP PLC
4.1 Siemens PLC Overview

The S7-300 series by Siemens is a mini PLC suitable for low to mid performance requirements [26]. The PLC has a compact modular design with a wide range of CPUs with different technical specifications [26]. The PLC can also be expanded with attachable signal, function and communication modules such as digital and analog I/O, high speed counters and timers, and communication processors for connection with different bus systems [27].

4.1.1 CPU

The CPU used with the PLC for this project was the CPU 314C-2 DP. This CPU has integrated digital and analog I/O, and communication interfaces for MPI and Profibus-DP networks [26]. The CPU requires a SIMATIC micro-memory card and a nominal supply voltage of 24V DC to operate [26]. The memory card is used to store programs created in TIA Portal.

4.1.2 Digital and Analog I/O Modules

The CPU 314C-2 DP also has integrated digital and analog I/O modules. The digital and analog I/O module has eight digital inputs, five analog inputs, and two analog outputs that are rated for ± 10 V DC [28]. The digital I/O module has sixteen inputs and sixteen outputs which require and produce +24 V DC respectively [26].

4.1.3 Communication Modules

The CPU 314C-2 DP has two communication interfaces X1 and X2. Both interfaces use RS-485 9-pole D-type connection ports [28]. The X1 interface is used for MPI communication and the X2 interface is used for Profibus-DP communication [28]. The CPU 314C-2 DP supports routing between the interfaces for Siemens SIMATIC devices [28].
4.2 Connecting the PC to the Siemens PLC

There were two possible methods for connecting the PC to the PLC. The PLC allows two types of communication protocols to connect through the in-built X1 and X2 interfaces. Due to the different design layouts previously mentioned in Chapter 3, the PC would have to be connected using both MPI and Profibus-DP. Connecting the PLC to the PC was achieved for both protocols through the following configurations.

4.2.1 Connecting using MPI Communication

To connect the PC to the PLC using MPI communication, the hardware had to first be correctly installed. Wiring the PLC to the PC was achieved using a USB to MPI adapter as shown in Figure 4.2. The correct hardware installation required the PC and PLC both to be powered on, and the USB to MPI adapter connected from the USB port on the PC to the X1 port on the PLC. On the PC, the USB to MPI adapter required a driver to be installed for the data to be transmitted.

![Figure 4.2: USB to MPI Adapter](image)
4.2.2 Connecting using Profibus-DP Communication

To connect the PC to the PLC with Profibus-DP communication, the PC requires the CP-5611 PCI card. This card supports both Profibus-DP and MPI communication. The Profibus-DP cable must be connected from the X2 port on the PLC to the CP-5611 card at the back of the PC using Profibus-DP 9-pole D-type connectors with correct bus termination. On the PC, the CP-5611 card must be configured as a communication module for Profibus-DP in TIA Portal.

4.3 Using Siemens TIA Portal with the PLC

For this project, all programs and configurations developed for the PLC were made using TIA Portal. The configuration data is available for the majority of Siemens devices which are located in the device catalogue within TIA. Adding the PLC to the TIA project opens up a range of programming and configuration possibilities. Some programming and configuration options include: device configuration, PLC programming, PLC tags, watch and force tables. The following sections describe some of the programming and configurations that can be configured for the PLC.

4.3.1 Device Configuration

The device configuration window displays most of the parameters that can be adjusted for the PLC. The device overview lists all the configured modules that are accessible by the PLC program and are given unique addresses. The number of configuration possibilities is extensive, but the main parameters configured for this project were for the MPI and Profibus-DP interfaces. These parameters include configuring the subnet, address and whether the PLC is a master or I-slave on the DP network.

4.3.2 PLC Programming

The program blocks folder in TIA is where the user-made programs and code can be developed. The programming developer can choose up to four different PLC programming languages depending on their preference including: Ladder Diagram (LAD), Function Block Diagram (FBD), Statement List (STL), and Structured Control Language (SCL). For this project, all programs were written in LAD for
consistency. In addition to the language, there are four types of program blocks that can be programmed to the PLC, they are: Organisation Block (OB), Function Block (FB), Function Call (FC), and Data Block (DB). The OB that must always be configured to the PLC is the Main [OB 1]. This OB performs the main scan cycle sequence for the PLC and is executed continuously unless interrupted by special functions. The main programs for the PLCs process data or control system are developed here.

4.3.3 PLC Tags
Any memory location used in the PLC program is defined as a tag. In the PLC tags folder in TIA, a list of all the tags defined in the current project can be modified or displayed. Creating a tag table assists with organising the tags for specific functions or sections of the program. For the PLC, a tag can refer to five types of addresses, they are: Local Input (I), Local Output (Q), Local Memory (M), Timer (T), or Counter (C). In addition to the address, the tag also indicates a data type. There are fourteen different data types available, some of which include: Boolean (Bool), Byte, Integer (Int), Real, and Word. The data type is also added to the last part of the address, for example: bit 4 from memory byte 1 has the address %M1.4 which is Boolean, local input byte 9 has the address %IB9 which is configured to a data type with the length of a byte.

4.3.4 Watch and Force Tables
Watch and force tables present an online feature to PLC programming. The tables are best used for debugging and testing the program during its development. The address or tag to be monitored or controlled is first entered in the table. In force tables, the values of the tags are monitored and can be forced to a new value. When a tag is forced using a force table, it is locked to the new value until the forcing is stopped. In watch tables, the tag values are monitored, but can only be modified for one instance. Once a tag is modified in the watch table, it takes the new value momentarily and can be overwritten by any updates made to the tag in the main PLC program.
4.4 Profibus-DP Organisation Blocks

Organisation blocks perform specific functions that are executed by the PLC’s CPU [33]. The main cyclic execution of the CPU is performed by OB 1. Other OBs perform various functions that usually alter the CPU scan sequence or are triggered by diagnostic data. These functions include timing interrupts, fault interrupts, hardware interrupts, alarm interrupts, and start-up procedures. The following OB’s are used with Profibus-DP communication.

4.4.1 Rack Failure [OB 86]

This OB initiates a fault interrupt to the PLC’s CPU if a communication error between the Profibus-DP master and slave is detected [33]. By configuring OB 86, when a fault is detected the CPU initiates the OB and diagnostics of the fault instead of stopping the CPU. This is an important feature especially for automation and factory control systems where stopping the PLC can cause significant problems. With OB 86, a user-defined program can be executed to handle the event of a Profibus-DP slave failure.

4.4.2 Diagnostic Interrupt [OB 82]

It occurred that when a Profibus-DP network with a master and I-slave PLC was configured, neither PLC could enter run mode. It was discovered that OB 82 had to be configured to each PLC in order for the PLCs to run. OB 82 is triggered by a diagnostic interrupt caused by the exchange of diagnostic data between the master and I-slave PLCs [33]. When configured to both PLCs, OB 82 collects diagnostic data from the other PLC without stopping the CPU.

4.4.3 Warm Restart [OB 100]

This OB is not required for Profibus-DP communication, but is recommended for any network involving industrial and automation devices. The warm restart OB executes once whenever the PLC is switched from stop to run mode. The purpose of this OB is to initialise process variables used by the PLC to correct starting values before running the main OB 1. Since some memory in the PLC can be retentive, it is best to initialise each variable to ensure the network starts correctly.
4.5 Configuring the PLC to the Profibus-DP Network

The Siemens PLC was connected in two distinct configurations for the project. The first configuration was as a class 1 Profibus-DP master which was required for all networks. The second was as a Profibus-DP intelligent slave exchanging data with a PLC master.

4.5.1 As a Class 1 Profibus-DP Master

All Profibus-DP networks must have at least one class 1 master PLC active. The Siemens PLC is capable of being configured as a DP master from TIA Portal. The configuration can be downloaded using either MPI or Profibus-DP communication from the PC and the PLC. The first steps are to add the PLC to the TIA project and apply the organisation blocks used with Profibus-DP. Setting up the PLC as a master in TIA is quite simple. The Profibus-DP parameters for the PLC are located in the DP interface [X2] properties in the device configuration window of TIA. From here the Profibus address and operating mode parameters can be altered.

To add the PLC to the Profibus-DP network, the PLC must be configured to a Profibus-DP subnet. Here a new Profibus-DP subnet can also be created to setup a new network. The Profibus-DP address for the PLC must be unique compared to any other devices on the network. The default address for the Siemens PLC is address 2. It is recommended to set the Profibus-DP address using a MPI connection from the PC. This is because the PLC cannot be found by the Profibus-DP network if the new address does not match the existing address previously configured. For MPI however, the USB to MPI adapter ensures that the connection is point-to-point so the PLC is the only device the PC will detect. The final parameter in configuring the Profibus-DP master PLC is selecting the operating mode as DP master. Downloading these parameters to the PLC will now make it a class 1 Profibus-DP master for the selected Profibus-DP subnet.
4.5.2 As a Profibus-DP Intelligent Slave

Additional Siemens PLCs added to the Profibus-DP network were configured as I-slaves for this project. An intelligent slave is a slave on the Profibus-DP network that has a CPU capable of performing advanced processes. To set up the Siemens PLC I-slave, the PLC had to be connected to an existing Profibus-DP network with a Siemens master PLC. The Profibus address of the I-slave had to be unique compared to other devices on the network. It was found to be easier to configure the I-slave using MPI communication from the PC. This was because the default PLC addresses on all the Siemens PLCs were initial set as 2, meaning the Profibus-DP network was unable to locate the devices due to addressing clashes. MPI connected directly to the PLC to be configured so all the parameters could be transferred without being confused with other devices sharing the same address on the network.

Also in the Profibus-DP parameter configuration window in TIA, the operating mode has the option to be configured as a DP slave. All Profibus-DP slaves must be assigned to a master. When the master is assigned, TIA automatically creates a master system connection between the master and I-slave.

When configured as an I-slave, the parameter window expands with the option to configure the I-slave communication. Communication is performed using defined transfer areas. Adding transfer areas defines input and output addresses on both the master and slave PLCs, including the length and direction of the message. It was found that, when downloading to the devices, it was best to download the configurations to the I-slave first to initiate the transfer areas, then to the master to accept the communication.
Chapter 5 Moeller EASY719-DC-RC Control Relay

This chapter aims to provide information about the Moeller EASY719-DC-RC control relay [34] that was configured to the Profibus-DP network as shown in Figure 5.1. The communication interface on the EASY719 was not initially developed for Profibus-DP communication, therefore a gateway device was required. The EASY719 can be connected to the Moeller EASY204-DP [35] as shown in Figure 5.2, which is used as a gateway between the original EASY719 communication protocol and the Profibus-DP protocol. The EASY719 was only configurable as a Profibus-DP slave, which was achievable with the use of the GSD file developed for the EASY204-DP, as mentioned in chapter 3.2 of this report. The following sections in this chapter aim to describe the purpose of each configuration.

For the complete list of steps used to configure the EASY719-DC-RC and EASY204-DP to the Profibus-DP network, please refer to Appendix B: Moeller EASY719-DC-RC Control Relay Profibus Configuration Guide.

![Figure 5.1: Moeller EASY719-DC-RC](image-url)
5.1 Moeller Control Relay Overview

The EASY719-DC-RC is part of the EASY700 basic unit series developed by Moeller. It is an electronic control relay that is capable of performing multiple control-based functions such as digital I/O logic functions, timing functions, counters, and measuring analog inputs [34]. The EASY719 device includes an LCD display with buttons for status and configuration, digital and analog inputs, output relays, and a port for expansion devices. The device requires a nominal supply voltage of 24V DC to operate.

5.1.1 LCD Display

The EASY719-DC-RC has an in-built LCD screen and buttons for displaying and configuring functions and status of the device. The main screen shows the operating mode of the device, and any inputs and outputs that are currently active. Pressing the OK button will change to the next screen and ESC will return to the previous screen. Ladder logic diagrams can be developed using the buttons and LCD screen. Some other functions include starting and stopping the control relay program, and adjusting the in-built clock. The system menu can be opened by simultaneously pressing the DEL and ALT keys. From the system menu, password protection, operating options, language, and expansion device options can be configured.

5.1.2 Inputs and Outputs

The EASY719-DC-RC has 12 digital inputs, 4 of which can also be configured as analog inputs. The digital inputs require 24V DC to trigger the Moeller program. The analog inputs can read voltages anywhere between 0 and 10V DC. The analog input voltages correspond to a 10-bit value, for example; 0V DC is the digital value 0, 5V DC is the digital value 512, and 10V DC is the digital value 1023. There are 6 output relay coils that can be controlled by the program. The coils are completely isolated, therefore can support up to 230V AC and 8 Amps.
5.1.3 Expansion Units

The EASY700 basic units have the ability to attach expansion units to add extra functionality to the device. Expansions include providing extra I/O or bus gateway connectivity. For this project, the Moeller EASY204-DP Profibus-DP slave gateway [35] as shown in Figure 5.2 was used to connect the EASY719-DC-RC to the Profibus-DP network. The expansion unit can be connected to the basic unit through the EASY-LINK port on both devices. The EASY204-DP also requires a nominal power supply of 24V DC to operate.

![EASY204-DP Profibus-DP Gateway](image)

5.2 Creating Programs on the Control Relay

The EASY719-DC-RC is a distributed I/O system capable of performing ladder logic-style programming tasks. Functions include local I/O, expansion I/O, timers and counters. Programs can be created on
the EASY719 via a PC or using the LCD and buttons of the device. For PC connection, the EASY-PC-CAB cable as shown in Figure 5.3 was required. The Moeller control relay programming PC software is called EASY-SOFT-BASIC and can be used to create logic programs on the PC. For this project, simple programs were created using the LCD and buttons on the device. The programs were kept simple to demonstrate the Profibus-DP communication between the control relay and PLC which was part of the primary aim of the project.

Figure 5.3: Moeller EASY-PC-CAB Cable

5.3 Using Siemens TIA Portal with the Control Relay

5.3.1 Installing the GSD File

The Moeller control relay was able to be configured as a Profibus-DP slave that was controlled by a Siemens S7-300 PLC master. The Profibus-DP communication between the control relay and master PLC were configured in TIA Portal. The control relay could be added to the TIA project by installing the EASY204-DP’s GSD file. TIA allows GSD files to be added or removed from the library in the Manage General Station Description files window under the options menu. The GSD file Moe4d10.gsd was the file that set up initial configuration for the control relay, and allowed the device to be configured in TIA. The TIA library was updated to include the EASY204-DP device, which was located under switching devices in the other field devices hardware catalogue.
5.3.2 Configuring the EASY204-DP Profibus-DP Parameters

Once the GSD file is installed, the EASY204-DP device can be added to the network view of TIA. The EASY204-DP is a slave device therefore it must to be assigned to a Profibus-DP master PLC. Assigning the master adds the device to the DP-Mastersystem and the Profibus-DP parameters could be set in the properties window. The Profibus address is automatically selected as the next available address but can be changed in the Profibus address menu.

5.3.3 Configuring Communication between the Master PLC and Control Relay

The GSD file also updates the device library with the programmable modules that can be configured with the EASY204-DP. These modules can be found and configured in the device view window for the EASY204-DP. Specific modules had to be configured for the EAY204-DP for use with the EASY700 unit. Table 5.1 lists the modules used with the EASY700 unit.

<table>
<thead>
<tr>
<th>Module Name</th>
<th>Functions</th>
</tr>
</thead>
</table>
| EASY 700/800 Control Commands | • Read/write date and time  
• Read/write image data  
• Read/write function block data |
| Input 3 Byte | • Scan the operating mode  
• Scan status of the easy output S1 to S8 |
| Output 3 Byte | • Set operating mode  
• Set/reset the easy inputs R9 to R16  
• Set/reset the easy inputs R1 to R8 |

Table 5.1: Modules Configured to EASY719-DC-RC [35]

The input and output modules perform the normal cyclic data exchange for the expansion (R and S) I/O programmed on the control relay. Since the local inputs and outputs on the control relay are already defined as I and Q respectively, the expansion inputs and outputs accessed through Profibus are defined as R and S by the control relay respectively. The control commands module also uses
cyclic data exchange, however the PLC must toggle a byte in order to exchange data [35]. The image
data stores the status of the other functions performed by the control relay such as local I/O, timers
and counters [35].

When the modules are added to the device they are given unique addresses for communicating with
the PLC. All three modules configured to the control relay use cyclic data exchange (DP-V0) with the
master.

5.4 Configuring the Control Relay Hardware for Profibus-DP

When the configurations were made using TIA portal, the project could then be downloaded to the
master PLC. Once the download was complete, the master PLC would then start polling the control
relay slave. To connect the EASY719 to the Profibus-DP network, the address of the control relay
slave must match the address configured in TIA. To change the address on the EASY719, the control
relay must be in stop mode and the system menu opened by simultaneously pressing the DEL and
ALT buttons. Selecting the configurator displays the configuration window for the Profibus-DP
address on the unit. Matching the address to the one configured in TIA Portal will allow the PLC to
find the device with correct address, and the data exchange can occur. When the EASY204-DP is
connected successfully to the bus, the bus LED will turn on and the GW symbol on the EASY719 LCD
will stop flashing.
Chapter 6 SEW Eurodrive MOVITRAC B VSD

This chapter aims to provide information about the SEW MOVITRAC B VSD [39] that was configured to the Profibus-DP network as shown in Figure 6.1. The communication interface on the MOVITRAC B was not initially developed for Profibus-DP communication, therefore a gateway device was required. The MOVITRAC B can be connected to the SEW DFP21B Profibus-DP-V1 fieldbus interface [40] as shown in Figure 6.4, which is used as a gateway between the original SEW SBus communication protocol and the Profibus-DP protocol. The MOVITRAC B was only configurable as a DP slave, which was achievable with the use of the GSD file developed for the SEW DFP21B, as mentioned in chapter 3.2 of this report. The following sections in this chapter aim to describe the purpose of each configuration.

For the complete list of steps used to configure the SEW MOVITRAC B and SEW DFP21B to the Profibus-DP network, please refer to Appendix C: SEW Eurodrive MOVITRAC B VSD Profibus Configuration Guide.
6.1 SEW MOVITRAC B VSD Overview

Variable speed drives are used alongside electric motors in industrial applications to regulate the speed of the motor [38]. The SEW MOVITRAC B VSD used in this project uses a single phase AC supply to operate a three phase motor. The primary function of the VSD in this project was to control the set-point speed of the motor in RPM. The VSD could be operated manually via a front keypad, or from a network with the use of a front communication module and gateway. The VSD is powered by 230V AC at 50Hz from a wall outlet to operate the motor.

6.1.1 3-Phase Motor

The motor used with the MOVITRAC B for this project was the SEW R17 DR63M4 as shown in Figure 6.2. The motor is a helical gear motor with two axles. Table 6.1 shows some of the technical specifications for the motor. The rotational relationship of the axles is 1320/30. In other words, when
the small axle is rotating at 1320 RPM, the larger axle is rotating at 30 RPM. On the motor shown in figure 6.2, the small axle is located within the housing on the left side and the large axle can be seen exposed from the housing on the right side.

### Table 6.1: SEW R17 DR63M4 Technical Specifications

<table>
<thead>
<tr>
<th>Specification</th>
<th>Rated Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dual Axis Rotational Relationship</td>
<td>1320/30</td>
</tr>
<tr>
<td>Voltage Rating</td>
<td>220-240Δ/380-415Υ at 50Hz</td>
</tr>
<tr>
<td>Max Frequency/Speed</td>
<td>50Hz/3000 RPM</td>
</tr>
</tbody>
</table>

![Figure 6.2: SEW R17 DR63M4 Motor](image)

6.1.2 FBG11B Keypad

The keypad installed on the MOVITRAC B is a FBG11B Front option keypad [39]. The keypad allows for the direct manual control over the MOVITRAC B functions. The frequency of the VSD output can be adjusted manually by using the arrow keys and selecting the dial symbol. The mode is selected by pressing the enter key, and the VSD is started by pressing run. The output frequency can then be adjusted by turning the dial. The operating frequency is displayed in RPM on the LCD screen. The VSD can be disabled again by pressing the stop key.
6.1.3 FSC11B Communication Module

The FSC11B front communication module [39] allows the MOVITRAC B to communicate with a PC or SBus network as previously mentioned in chapter 3.1.5. The MOVITRAC B can be connected to the PC via a SEW USB11A adapter. The adapter connects to the X44 port of the FSC11B module using the RS-485 connector, and to the PC using a USB connection. The X46 port on the FSC11B module is used for SBus communication. The SBus terminal connections can be seen in Table 6.2 [39].

Table 6.2: FSC11B SBus Terminal Description [39]

<table>
<thead>
<tr>
<th>Terminal</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>X46.1</td>
<td>SC11: SBus high</td>
</tr>
<tr>
<td>X46.2</td>
<td>SC12: SBus low</td>
</tr>
<tr>
<td>X46.3</td>
<td>Ground</td>
</tr>
<tr>
<td>X46.4</td>
<td>SC21: SBus high</td>
</tr>
<tr>
<td>X46.5</td>
<td>SC22: SBus low</td>
</tr>
<tr>
<td>X46.6</td>
<td>Ground</td>
</tr>
<tr>
<td>X46.7</td>
<td>24V DC</td>
</tr>
</tbody>
</table>

6.2 Connecting the MOVITRAC B to the DFP21B

The MOVITRAC B had to be connected to the DFP21B in order to connect to the Profibus-DP network. The communication between the VSD and Profibus-DP gateway is SBus. SBus is a simple fieldbus protocol that uses 4 wires to transmit the high and low data signals and power supply voltage. To supply the voltage to the SBus network, pins 8 and 9 on the X12 port located behind the FSC11B was wired to a 24V DC power supply. The SBus network was connected to the X46 port on the FSC11B communication module and the X26 port on the DFP21B. Figure 6.3 shows the wiring diagram between the two SBus ports for the network.
6.3 Configuring the MOVITRAC B using MOVITOOLS

MOVITOOLS-MotionStudio [44] was used to initialise the MOVITRAC B for Profibus-DP and SBus communication. The PC was connected to the MOVITRAC B by using a USB11A adapter to connect the USB port from the PC to the X44 RS-485 port on the FSC11B communication module. In MOVITOOLS, the MOVITRAC B can be located by searching the Serial USB network. Once found, the start-up wizard was initiated to configure the MOVITRAC B basic parameters such as unit information, application types, power ratings and limits.

6.3.1 Manual Override Switch

To establish connection to the SBus network, further parameters needed to be configured through the parameter tree in MOVITOOLS. Firstly as a safety feature, a 24V DC switch was required to be wired to one of the X12 digital inputs located behind the FSC11B module. Once the switch was connected, the binary inputs parameter settings for this digital input were set to the function CW/Stop. All the other digital inputs were then configured as no function. From this configuration the switch can now be used to change the VSD’s operating mode from manual mode to networked mode. The connection of this switch is a failsafe requirement for the MOVITRAC B VSD, without it the VSD cannot be controlled by an external controller.
6.3.2 Process Data

The next step was to control the type of data that would be sent to and from the VSD to an external controller. This data could be modified in the MOVITOOLS parameter tree from the Process Data Parameter Assignment folder. The MOVITRAC B VSD is capable of sending three process output data words, and receiving three process input data words to and from a controller. In order for an external controller to control the VSD’s primary functions, Control Word 1 and Set-point Speed must be configured as process output data, and Status Word 1 and Actual Speed configured as process input data. These process data words allow a controller to start and stop the VSD, change the RPM of the motor, read the motor’s RPM, and read the status of the VSD.

6.3.3 SBus Communication

The last parameter to be set was the SBus network parameters. Table 6.3 displays the SBus parameters were configured to achieve successful communication:

<table>
<thead>
<tr>
<th>SBus Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Protocol</td>
<td>MoviLink</td>
</tr>
<tr>
<td>Address</td>
<td>1</td>
</tr>
<tr>
<td>Group Address</td>
<td>0</td>
</tr>
<tr>
<td>Timeout</td>
<td>0.1</td>
</tr>
<tr>
<td>Baud Rate</td>
<td>500 kbaud</td>
</tr>
<tr>
<td>CANopen address</td>
<td>2</td>
</tr>
</tbody>
</table>

Similar to Profibus-DP networks, the address for each device on the SBus network had to be unique. By default the DFP21B is configured as address 0. The Baud rate also had to be the same for all devices on the network. The SBus timeout was recommended for any time between 0.05 and 0.2 seconds. The feature CANopen was not used, therefore the address did not need to be configured.
6.3.4 Upload Parameter Tree from PC to VSD

Once the start-up wizard, manual override, process data and SBus parameters had been set, the new parameters had to be transferred to the VSD. This was done by right-clicking the connected VSD in the MOVITOOLS and selecting upload. When the new parameters were installed, connection between the DFP21B and MOVITRAC B was established by toggling the AS dip switch on the DFP21B off, then on again. The AS dip switch stands for auto-setup, that automatically searches the SBus network for configured VSDs. If the auto-setup is successful, the H1 LED on the DFP21B will turn off and the MOVITRAC B will be connected to the DFP21B.

6.4 Using Siemens TIA Portal with the SEW VSD

6.4.1 Installing the GD File

The SEW VSD was configured as a Profibus-DP slave that was controlled by a Siemens S7-300 PLC master. The Profibus-DP communication between the VSD and master PLC were configured in TIA Portal. The VSD could be added to the TIA project by installing the SEW DFP21B GSD file. The GSD file SEW_6009.gsd was the file that set up initial configuration with the VSD, and allowed the device to be configured in TIA. The TIA library was updated to include the DFP21B device, which was located under drives in the other field devices hardware catalogue.

6.4.2 Configuring the SEW DFP21B Profibus-DP Parameters

Once the GSD file is installed, the SEW DFP21B device can be added in the network view of TIA Portal. The DFP21B is a slave device, therefore required to be assigned to a Profibus-DP master PLC. Assigning the master adds the device to the DP-Mastersystem allowing the Profibus-DP parameters to be configured in the properties window. The Profibus address is automatically selected to the next available address but can be changed in the Profibus address menu within TIA Portal.
6.4.3 Configuring Communication between the Master PLC and VSD

The GSD file also updates the device library, adding the programmable modules that can be configured with the DFP21B. These modules can be applied to the DFP21B in the device view window in TIA Portal. Specific modules had to be configured for the DFP21B for use with the MOVITRAC B VSD. Table 6.4 lists the modules used with the MOVITRAC B VSD.

Table 6.4: Modules Configured to SEW MOVITRAC B [40]

<table>
<thead>
<tr>
<th>Module Name</th>
<th>Functions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter Channel (8 byte)</td>
<td>• Read/write MOVITRAC B parameter data</td>
</tr>
<tr>
<td></td>
<td>• Read/write DFP21B parameter data</td>
</tr>
<tr>
<td>AS 1 Drive (3 PD)</td>
<td>• Read Process Input Data</td>
</tr>
<tr>
<td></td>
<td>• Write Process Output Data</td>
</tr>
</tbody>
</table>

When the DFP21B device view is opened in TIA, there are initially two reserved slots named PROFIsafe and Param-Channel. The Parameter Channel (8 byte) module must be added to the Param-Channel slot to ensure the correct address is configured between the PLC and DFP21B gateway. The PROFIsafe functions were not used in this project, therefore the PROFIsafe slot was left empty. For this project, only one SEW VSD was used, therefore the AS 1 Drive (3 PD) module was added to configure the addresses between the PLC and DFP21B that enable the functions of one drive only.

The Profibus-DP parameter channel module is only able to access the DFP21B Profibus-DP gateway parameters and not the parameters of the MOVITRAC B VSD. This made it possible to change some of the DFP21B’s settings such as baud rate, directly from the PLC while connected using Profibus-DP. The settings for the MOVITRAC B VSD had to be adjusted by connecting to a PC with MOVITOOLS installed as mentioned in chapter 6.3 regardless if connected to the PLC through the DFP21B gateway. This is because the Profibus-DP and SBus protocols are incompatible, therefore the PLC was unable to access the MOVITRAC B directly [40]. The parameter channel module uses cyclic data exchange from the PLC to the DFP21B through the Profibus-DP network meaning data is continuously
transferred, however it is designed in the DFP21B that the PLC must toggle a specific byte in order for the DFP21B to accept that data [40].

The VSD uses cyclic transmission to exchange the process data sent and received by the drive. Each drive sends three words (6 bytes) of process output data, and reads three words (6 byte) of process input data. The meaning of that data is configured in the process data parameter assignments assigned in MOVITOOLS as mentioned in chapter 6.3.2. Process data is used to start and stop the drive, change the speed set-point, read the drive status, and read the drive actual speed.

6.5 Configuring the DFP21B Hardware for Profibus-DP

When the configurations were made in TIA portal, the project could then be downloaded to the master PLC. Once the download was complete, the master PLC would then start polling the VSD slave. To connect the MOVITRAC B and DFP21B to the Profibus-DP network, the address of the DFP21B slave must match the address configured in TIA. The dip-switches on the front of the DFP21B as shown on figure 6.4, are used to configure the Profibus-DP address. The address is equal to the value labelled next to the switch. If multiple dip-switches are active, the DP address is the sum of the active values. For example the Profibus-DP address for 4 will be the switch labelled $2^2$. If the switch labelled $2^2$ and $2^0$ are active, the address will be 5. Matching the address to the one configured in TIA Portal would allow the PLC to find the device with correct address, and the data exchange could occur. When the DFP21B is connected successfully to the bus, the bus fault LED will turn off.
Figure 6.4: SEW DFP21B Profibus-DP Gateway
Chapter 7 Siemens HMI Touch Panel

This chapter aims to provide information about the Siemens TP 177B 6” PN/DP HMI touch panel that was configured to the Profibus-DP network as shown in Figure 7.1. The touch panel was configured as a HMI for the network by reading and writing data to and from the master PLC with the use of graphics. To integrate the HMI touch panel into the Profibus-DP network, both hardware and software configurations had to be developed. The following sections in this chapter aim to describe the purpose of each configuration step. For the complete list of steps taken to configure the Siemens TP 177B 6” PN/DP HMI touch panel, please refer to Appendix D: Siemens HMI Touch Panel Profibus Configuration Guide.

7.1 Siemens HMI Touch Panel Overview

The HMI touch panel is a user interface suitable for deploying in the field where a PC may not be suitable. The HMI gives the user a graphical representation of the data processed by a system. Due to its small size, the touch panel is compact and well suited to small scale sections of a process.
For this project, the Siemens HMI touch panel was used to monitor and control data exchanged from the Moeller control relay and DP master PLC. The HMI allows buttons and indicators to identify the state of variables instead of numbers and programming blocks that are found in PLC programming.

7.1.1 Display Screen
The touch panel has a six inch LCD screen which is suitable for simple HMI applications. The screen is touch sensitive and can perform tasks based on pressing, releasing, double tapping and dragging.

7.1.2 Communication Ports
The TP 177B touch panel had multiple ports allowing for various methods of communication as shown in Figure 7.2. The ports were physically designed for serial, USB, and Ethernet communication. The name PN/DP means that it supports both Profinet and Profibus-DP protocols. For this project, the serial port was used to connect the DB-9 Profibus-DP connector. Also on the back of the panel, are a set of dip-switches that can be used to select the protocol sent through the serial DB-9 port. These dip-switches were set to allow the port to send and receive the Profibus-DP protocol.

Figure 7.2: TP 177B 6” PN/DP Communication Ports
7.1.3 In-built Software

The HMI touch panel was provided with pre-installed software. The software that was configured was a Windows CE operating system, and a HMI version of WinCC Flexible [30]. The in-built operating system allows the settings and communication to be configured, while WinCC runs the HMI programs and communicates with the PLC.

7.2 Configuring the Touch Panel’s Operating System

The operating system that was pre-installed on the panel was used to configure the Profibus-DP communication between the PC and PLC. When the panel was powered on, the operating system control panel could be accessed, and the transfer settings were adjusted. In the transfer settings screen, the channel was configured for MPI/Profibus communication. This included the Profibus-DP address, time-out, transmission speed, highest address station, and profile. For Profibus-DP communication to be successful, the protocol settings on the touch panel operating system and the configured PLC must be the same. Once the settings are correctly adjusted and the panel is physically connected to the network, the operating system can be triggered for transfer mode. This mode triggers the panel to accept new network configurations and HMI programs sent by the PC.

7.3 Configuring the Touch Panel in TIA Portal

Since both TIA and the touch panel were developed by Siemens, the device configuration data was already included in the TIA Portal hardware catalogue. The WinCC software has been integrated into TIA Portal, therefore the touch panel HMI programs could be developed in TIA. To simplify the setup, WinCC had a HMI wizard that would automatically set the touch panel with a configured template. This provided the basic screen layout, navigation, alarm windows for the HMI. Similarly to the other devices, the touch panel was added to the Profibus-DP network from the device and networks window in TIA. This included a unique address that also needed to match the address configured on the touch panel’s operating system. In addition to the Profibus-DP connection, a HMI connection between the touch panel and master PLC also had to be established. Through the HMI connection,
the tags on the PLC and HMI can be shared, allowing tags used in the program to be monitored or controlled from either device. The HMI can control or monitor tags assigned to graphical objects using the events or animation controls.

7.3.1 WinCC HMI Programming

The HMI programs were made using Siemens WinCC [41]. A version of WinCC Runtime came pre-installed on the touch panel [30], which allowed programs to be developed in TIA Portal and downloaded to the device. For this project, the HMI wizard was used to create a basic template as seen in Figure 7.3 which could be built upon for the controlled purpose. The types of programs developed included screen navigation, event, animations, and alarm management.

![Figure 7.3: Root Screen Template for HMI Touch Panel](image)

7.3.2 Screen Navigation

After completing the HMI wizard setup, a drop box is created and assigned to a tag called Tag_ScreenNumber. This is created to change the screen that is displayed while the HMI is running. Each screen that is created in WinCC is defined by a number. A special event in WinCC allows the
screen that is displayed to be selected by that number. The number assigned to the screen number tag is selected by using a text list called TextList_ScreenNames. The text list contains the options that can be selected by the drop box, and the screen number that is displayed for each option.

7.3.3 Events

Events are used to alter a PLC tag or perform a special function on the HMI. There are a range of specific objects that can be added to the HMI screen that support events such as buttons, sliders and menus. The types of actions that can be selected to perform an event include click, press, release, activate, deactivate and change. The type of action selected depends on if it is user or process controlled. An example of a user controlled event is a human operator pressing a button on the HMI to set a tag in the PLC. An example of a process controlled event is a tag altered by the PLC causing the screen displayed on the HMI to change. Typical events include changing a PLC tag, navigating to a selected screen, acknowledging an alarm, or closing the HMI application.

7.3.4 Animations

Animations are used to monitor the status of a PLC tag or function on the HMI. There are a range of specific objects that can be added to the HMI screen that support animations such as shapes, text, dials, bars, and I/O fields. The types of animations triggered by a selected tag or function include appearance, visibility or movements. These animations can be used to inform the user about the state of the PLC’s variables. An example of an animation made by the HMI is changing the colour of an object on the screen to notify the user an alarm has been triggered, or a piece of machinery is active. Another example of an animation could be displaying the last value on the screen that was recorded by a sensor connected to the PLC. Typical animations include changing the colour of an object, causing an object to be visible or invisible, displaying the numerical value of a tag, filling a bar graph or moving a dial.
7.3.5 Alarm Management

The HMI alarms window allows various types of alarms to be triggered or displayed on the HMI. The types of alarms include discrete and analog alarms, controller alarms, system events, alarm classes and alarm groups. The alarms are triggered by selected conditions made by the PLC program, and can be configured to be displayed on the HMI. The alarms are displayed with text describing the type and cause of the alarm. It can also be configured to adjust an acknowledgement tag on the PLC to perform specified code related to the alarm.
Chapter 8 Siemens WinCC Runtime Professional

This chapter aims to provide information about the Siemens WinCC RT Professional software that was used to communicate with the Profibus-DP network. WinCC RT Professional has HMI capabilities similar to the touch panel, but also includes SCADA data logging functionality. WinCC RT Professional was installed on a PC connected to the Profibus-DP network to provide HMI control and monitoring, along with data logging, trends and alarm management. To integrate the PC with WinCC RT Professional to the Profibus-DP network, both hardware and software configurations had to be developed. The following sections in this chapter aim to describe the purpose of each configuration step.

For the complete list of steps used to configure the PC with WinCC RT Professional, please refer to Appendix E: WinCC Runtime Professional Profibus-DP Configuration Guide.

8.1 WinCC Runtime Overview

WinCC Runtime (RT) is a PC-based software package that is used for HMI operator control and monitoring of a process [41]. There are three versions of WinCC RT that can be integrated into TIA Portal, they are: WinCC RT Advanced, WinCC RT Professional, and WinCC RT Client. The version used in this project is WinCC RT Professional which provides industry standard visualisation, reporting and logging of a process [41]. The PC must be installed with both the WinCC RT Profesional software and the WinCC Server for Runtime Professional to operate. WinCC RT Professional can also be setup as a multi-user system by using the WinCC RT Client to connect multiple PCs to a WinCC RT Professional Server [41].

8.2 Connecting the PC to the Profibus-DP Network using TIA Portal

As mentioned in chapter 4, the PC requires a CP 5611 PCI card to be physically connected to the Profibus-DP network. Once connected, the PC with WinCC RT Professional software can be added the TIA project from the “add new device” window. The SIMATIC WinCC RT icon should also be present
in the PC’s system tray, located next to the date and time at the bottom left of the windows PC desktop. When the PC is added to the networks screen, the communication module must also be configured. For this project, the communication module is the Profibus CP 5611 (A2) card. Similar to the touch panel, the HMI wizard for WinCC RT Professional can then be run, which sets up the basic template for HMI navigation. Like the touch panel, the PC must also be connected to the Profibus-DP subnet, and a HMI connection must be made between the PC and the PLC. The final step to configure the connection between the PC and PLC was in the connections screen. From this screen is the option to change the HMI connection access point from the default CP_L2_1 to CP5611.PROFIBUS.1, which allowed the PC to connect to the PLC through Profibus-DP.

8.3 WinCC RT Professional HMI and SCADA Functions

After completing the WinCC HMI wizard and the basic screens template was created, the screens could then be customised to include various HMI and SCADA functionality. HMI programming was identical to that performed on the HMI touch panel in chapter 7. However, the PC-based WinCC RT Professional included higher level SCADA functions that could not be achieved on the touch panel. The main SCADA functions performed by the PC with WinCC in this project were data logging and trending.

8.3.1 Data Logging

Data logging can be performed by the WinCC RT Professional PC by configuring data logs in the historical data configuration menu. When the data log has been created, the tags that are logged can be configured to the database. This includes a wide range of configuration options for each tag logged including acquisition mode, logging rate, start and stop tags, and high and low limits. As long as the PC is running, the tags from the PLC or device will be continuously logged and stored in the database.
8.3.2 Trends

Trends are used to plot process variables in real time. They can be used to monitor variables or used for data analysis. WinCC RT Professional allows trends to created and monitored as a HMI screen. The editor in TIA Portal has a range of properties for configuring the trend appearance, layout, axis, and process variables. To add a trend to the plot, the process variable must first be assigned as a tag logged in a database. The trend is then added to the trend properties and the logged tag is defined as the data source. Other properties include the style of the line and limits of the data. The WinCC trend view allows online controls, meaning some properties of the trend can be altered while the program is running. The trend can also be paused at any time, and historical values can be looked back at. The timestamp can be assigned from the PC itself or imported from the PLC. The trend view can be seen in Figure 8.1.

![WinCC Online Trend Control](image)

Figure 8.1: WinCC RT Professional Online Trend Control

8.3.3 Alarm Management

Alarms in WinCC RT Professional can be created similarly to the HMI touch panel. The alarm control in WinCC RT Professional however allows online control. This includes the ability to filter types of alarms, show current or historical alarms, and alarm statistics can also be viewed that show the number of times and alarm has been triggered and its duration.
Chapter 9 Testing and Results

Once the connectivity and operation of the devices and Profibus-DP protocol were better understood, a larger network could be accomplished. As previously mentioned in chapter 3.3, it was desired to complete specific design arrangements for use in the engineering laboratory. During this project it was difficult to acquire space in the lab due to constant use by other students, therefore the test panel previously shown in Figure 3.2 was used. This allowed a small scale version of the desired network to be accomplished, which utilises all the necessary components and connections used by all three of the previously discussed design arrangements. Figure 9.1 shows the final network created for the project which is a small scale version of design arrangement three. This arrangement incorporates a PC using Siemens’ WinCC RT Professional along with all the devices used by two panels in the laboratory such as:

- A Siemens S7-300 PLC configured as a Profibus-DP master
- A Siemens S7-300 PLC configured as an intelligent slave
- Two Moeller EASY719-DC-RC Control Relays and EASY204-DP gateways both configured as Profibus-DP slaves
- A SEW MOVITRAC B VSD and DFP21B gateway configured as a Profibus-DP slave
- A Siemens TP 177B 6" PN/DP HMI Touch Panel configured as a Profibus-DP HMI

The complete list of steps taken to configure each of the above connections as well as the PC with WinCC has been documented in the appendices of this report, which were designed as guides for new users to follow. The guides were the main objective of this project and the information documented allowed this final network to be achieved.
Testing the Connections

Initially the connections between the master PLC and slave devices such as the Moeller control relay and SEW VSD, were tested using watch tables as previously described in chapter 4. The Moeller and SEW devices were connected to the master PLC as documented in appendix B and C respectively. Figure 9.2 and 9.3 show the watch tables that were created for the Moeller control relay and SEW VSD. The watch tables contain all the tags used by the PLC to communicate with the devices. In this case, Slave_1 refers to the Moeller EASY204-DP gateway and Slave_2 refers to the SEW DFP21B gateway. The watch table shows the assigned memory address of each tag used by the PLC. When online, the last recorded value of each tag is shown in the Monitor value column. The values of the tags can be modified by the user in the Modify value column. This method was used to test the connections for the first time, for example in Figure 9.2, modifying the value of the “Slave_1 Set
Operating Mode” tag to 34 in hexadecimal format would cause the Moeller to enter “run mode”.
This was confirmed by the “Slave_1 Scan Operating Mode” tag to monitor the hexadecimal value of 21 to indicate the Moeller is now in “run mode without debounce”. A complete list of commands for the Moeller and SEW devices have been documented in appendix B and C respectively.

Figure 9.2: Watch table to monitor the connection between the master PLC and Moeller control relay
For the network shown in Figure 9.1, a program was created that tested the connections of all the components used. The network was developed to have full SCADA functionality such as data logging and HMI support. The master PLC performed the communication between all the devices on the network simultaneously while the touch panel and WinCC HMI interfaces were developed to trigger specific functions for the devices and display the results of those functions.

Figure 9.4 and 9.5 show the screens created for the touch panel HMI to control the basic functions of the Moeller control relay. The basic functions involve: setting the operating mode, scanning the operating mode, setting and resetting the Moeller’s “R” inputs, and scanning the status of the Moeller’s “S” outputs. The first HMI screen as shown in Figure 9.4, is the program developed to control the Moeller control relay’s operating mode. This was achieved by creating two buttons linked to the PLC tag used to run or stop the Moeller. An indicator was also created to identify the operating
mode from the HMI which can be seen as the grey circle labelled “Run/Stop” in Figure 9.4. The colour grey was chosen to represent the stop mode, while when in run mode the circle will flash between grey and green at approximately 1 hertz. The “Sent” and “Received” displays were created for the purpose of debugging. The value displayed in these boxes was the value stored in the PLC tags that were used to control and monitor the Moeller’s operating mode.

![Figure 9.4: Screen created on the Siemens HMI touch panel to control/monitor the Moeller’s operating mode](image)

Figure 10.5 shows the second screen created to control the Moeller control relays other basic functions. On this screen, eight buttons were created to set and reset the Moeller’s “R” inputs which were labelled R1 to R8. Pressing any of these buttons would alter the corresponding PLC tag to activate or deactivate the input on the Moeller. An additional program was created on the Moeller control relay to activate the “S” outputs when the “R” inputs were set. The eight indicators displayed as white circles in Figure 9.5, indicated the state of the Moeller’s “S” outputs, with the outputs labelled from S1 to S8. The colour white was chosen to indicate the output was the “off” state, while green was chosen to indicate the output was in the “on” state. The “S Output” and “R Input” boxes
were used for debugging purposes to display the last recorded value stored in the corresponding PLC tags. The tags were represented in binary format to show the status of all eight “S” outputs and all eight “R” inputs. Also on this screen was the indicator used also on the first screen shown in Figure 9.4, to show the operating mode of the Moeller control relay.

![Screen created on the Siemens HMI touch panel to control/monitor the Moeller’s “S” and “R” inputs and outputs](image)

Figure 9.5: Screen created on the Siemens HMI touch panel to control/monitor the Moeller’s “S” and “R” inputs and outputs

A screen was also created for the Moeller control relay on the PC with Siemens’ WinCC RT Professional as shown in Figure 9.6. This screen simultaneously controlled/monitored all the functions performed by the HMI touch panel, but also included an additional function to read the local “Q” outputs that are on the Moeller. For this to be tested, the program on the Moeller was expanded to set the “Q” outputs on when the “R” inputs were active. Reading the “Q” outputs on the Moeller was much more complex than the “S” inputs because it required additional information such as the message command, length, type and index. When the correct values were sent, a toggle byte must also be modified to complete the data transfer between the Moeller and the PLC. When the correct information is sent, the Moeller responds to the PLC with the requested information, in this
case the state of the Moeller’s “Q” outputs. A complete list of commands that can be requested and the corresponding values for the Moeller control relay have been documented in appendix B. For this operation to work, a PLC program was developed so that when the “Read Q Outputs” button as shown in Figure 9.6 was pressed, the PLC would then toggle the byte along with the appropriate information. This would then cause the Moeller to return the state of the “Q” outputs and display the results on the HMI indicators labelled Q1 to Q8. The only downside to this information was that it did not update in real-time. For example if the status of the “Q” outputs changed, it would not be displayed on the HMI until the “Read Q Outputs” button was pressed.

Figure 9.6: Screen created on the PC using Siemens’ WinCC RT Professional to control Moeller’s basic functions

Similar to the Moeller, HMI and PLC programs were developed to control the basic functions of the SEW MOVITRAC B VSD. Figure 9.7 shows the screen created using Siemens’ WinCC RT Professional to: start or stop the VSD, set the speed of the VSD, read the speed of the VSD, and read the output
current generated by the VSD. This screen made use of the HMI’s sliding bars and dials to display the information. The sliding bar was used to select the speed of the VSD which was in RPM, while the dials monitored the actual speed in RPM and output current which was recorded as a percentage by default. The “RUN/STOP” button worked similar to the buttons used with the Moeller. PLC code was designed to send the hexadecimal value 6 to the tag which controlled the operating mode of the VSD, when the “Run” button on the HMI was pressed. The debugging section on the HMI displayed the actual values stored in the tags used in communication between the PLC and VSD. PLC code also had to be designed to scale the desired values shown on the HMI, to the actual values sent to the VSD. The relationship for speed was the RPM in decimal format, multiplied by 5 which gave the value to be sent to the VSD. The relationship for output current was the value in decimal format, divided by 10 which resulted in the value received by the VSD. A complete list of commands and corresponding values for the SEW VSD has been documented in appendix C.

Figure 9.7: HMI screen created using Siemens’ WinCC RT Professional to control the SEW VSD’s basic functions
To incorporate the second PLC configured as the intelligent slave, a simulated process of a tank filled with liquid was designed. A PLC program was created on the I-slave PLC which simulated the process. Information was sent between the two PLCs such as: the flow of liquid entering the tank, the level of liquid inside the tank and the flow of liquid exiting the tank. Figure 9.8 show the HMI interface of the simulated process created on Siemens WinCC RT Professional.

![Figure 9.8: HMI screen created using Siemens WinCC RT Professional to control/monitor the simulated process](image)

The SEW VSD was used to simulate the speed of the pump sending liquid to the tank and was control by the master PLC. This simulated process also was used to demonstrate WinCC RT Professional’s SCADA functions such as alarms and data logging. An alarm was created for when the tank level exceeded 80% as shown in Figure 9.9. The alarm can be customised to send an acknowledgement byte to the PLC which can be used for many purposes, for example slowing down the pump so the level decreases. In this program, the acknowledgement byte was used to cause the warning symbol
shown in Figure 9.8 to become visible. The bar on the side of the tank in Figure 9.8 also changed colour depending on the tanks level. When the high limit of 80% was reached the bar would turn red and flash, indicating the limit was exceeded.

![Figure 9.9: Alarm created using WinCC RT Professional to trigger when the simulated tank level exceeded 80%](image)

The last element added to the program was the WinCC data logging capabilities. Figure 9.10 shows the database that was created. Two PLC tags were selected to be logged. The tags were the simulated tank level and the speed of the VSD. The values stored in these tags were selected to be logged once every two seconds and could be viewed in real-time using the trend control screen previously mentioned in chapter 8.

![Figure 9.10: Data log created using WinCC RT Professional to monitor the simulated process](image)
Chapter 10 Proposed Future Work

This chapter documents the possible work that was not able to be achieved within the time frame of this project. The current progress made to the network only covers the primary and basic functions of each device, and of the Profibus protocol. The next stage would be to look at additional functions that the protocol and each of the devices are capable of. From the knowledge gained from this project, the following are some ideas for future work that may be implemented with some proposed suggestions.

10.1 Profibus-DP-V1

DP-V1 adds the ability to transfer acyclic data as an extension to the normal cyclic data transfer used by DP-V0. All communication to the devices’ primary functions that were configured during this project used cyclic data transfer. It was included in the device manuals however, that additional functions could be made using the acyclic transfer of data made possible by DP-V1. A recommended future work proposal would be to discover the acyclic transfer of data between the devices using DP-V1 capabilities. The following details about acyclic transfer were discovered.

10.1.1 DP-V1 Function Blocks

Acyclic data transfers whole data blocks in a single message, rather than separate bytes in periodic packages. As documented in the Profibus Guidelines, cyclic transfer was performed by the PLC using the function block GETIO and SETIO. The guidelines also document the function blocks used for acyclic transfer, RDREC and WRREC [36].

RDREC is the function block used by the PLC to read a process data record from the selected slave [36]. There are four inputs to the block that need to be configured. The REQ input initiates data transfer when it is equal to 1 [36]. The ID parameter is the address of the module configured by the PLC. INDEX is the data record from the slave device that is to be read [36]. MLEN specifies the length of data that will be read, the maximum length of acyclic data transferred 240 bytes [36]. RECORD is
the PLC data block that is to store the new information [36]. The output parameters on the block indicate and diagnose if the transfer is successful.

WRREC is the function block used by the PLC to write a process data record to a selected slave [36]. There are also four inputs to the block that need to be configured. The REQ input initiate data transfer when equal to 1 [36]. The ID parameter is the address of the module configured by the PLC. INDEX indicates the data record to be written [36]. LEN specifies the length of data to be written: the length of acyclic data transferred is from 1 to 240 bytes [36]. RECORD is the PLC data block that is to be sent [36]. The output parameters on the block indicate and diagnose if the transfer is successful.

10.2 Danfoss HVAC FC-102 VSD

The final device configuration to be achieved was for the Danfoss HVAC FC-102 VSD [42] as shown in Figure 10.1. The Danfoss VSD requires a three-phase AC power supply to operate. This was the reason that the SEW VSD was chosen for use since only one three phase supply source was available in the ICSE lab. Consulting the HVAC FC-102 manual should provide information on how to setup the drive for Profibus-DP communication. There is also a programming guide with the description of parameters that can be sent and received through Profibus communication [42]. To add the Danfoss VSD to the network in TIA Portal, the correct GSD file will have to be found on the Danfoss website for Profibus files [43]. Once the file is downloaded and installed into TIA Portal, the device can be used and configured on the Profibus network. Referring to the manual, the correct modules are added and configured to the master PLC, using either cyclic or acyclic transfer depending on the module's programming.
Figure 10.1: Danfoss HVAC FC-102 VSD
Chapter 11 Conclusion

Profibus is a fast fieldbus technology used in industrial automation applications [1]. The aim of the project was to configure various devices using Profibus-DP and WinCC systems, and develop manuals to assist new users with the configuration of the selected devices. The project consisted of a review of the Profibus protocol, the design arrangements that were to be commissioned, an overview and configuration of each device used in the project, and the development of the configuration manuals for the devices.

This resulted in three design arrangements being developed where each arrangement required a different configuration. The first device studied was the Siemens S7-300 PLC which was configured as a master and I-slave for the Profibus-DP network. The installation of the RS-485 repeater was also configured and the network could be split into segments.

The GSD files for the Moeller control relay and SEW VSD were installed into TIA Portal, which allowed the devices to be configured to the master PLC. Both slave devices required a Profibus-DP gateway which would convert the device’s original communication into the Profibus-DP protocol, allowing them to communicate with the rest of the network.

The HMIs were configured last and were used to create graphical controls and indicators for each device by editing tags in the PLC. The Siemens WinCC Runtime Professional allowed PLC data to be logged on the PC, and also display trends and alarm management.

The final outcome resulted in basic operations of the devices being configured and documented as manuals for inexperienced users. The next proposed development of the project would be to look at more advanced communication between the master PLC and slave devices using Profibus-DP-V1 for acyclic data exchange. Also the configuration of the Danfoss HVAC FC-102 VSD should be attempted to have all available devices communicating on the network.
Appendices

The appendices of this thesis report contain the configuration manuals that were developed as the main goal for this project. The appendices are available on the DVD provided labelled as “James Wiggins Thesis Appendices 2015”. The appendices are provided as follows:

Appendix A: Siemens S7-300 PLC & RS-485 Repeater Profibus Configuration Guide

Appendix B: Moeller EASY719-DC-RC Control Relay Profibus Configuration Guide

Appendix C: SEW Eurodrive MOVITRAC B VSD Profibus Configuration Guide

Appendix D: Siemens HMI Touch Panel Profibus Configuration Guide

Appendix E: WinCC Runtime Professional Profibus Configuration Guide
Works Cited


