Metropolia

Leevi Tiihonen

Testing of Variable Speed Drive's Remote Access

Metropolia University of Applied Sciences

Bachelor of Engineering

Degree program of Electrical and Automation Engineering

Bachelor's Thesis

28 April 2023

Abstract

Author:	Leevi Tiihonen
Title:	Testing of Variable Speed Drive’s Remote Access
Number of Pages:	40 pages
Date:	28 April 2023
Degree:	Bachelor of Engineering
Degree Programme:	Electrical and Automation Engineering
Professional Major:	Electrical Power Engineering
Supervisors:	Sunilkumar Raina, Product Owner Eero Kupila, Senior Lecturer

Purpose of this study was to test the behaviour of variable speed drive as part of a network environment and to create and automate test cases according to these results. With the future development and new products comes the need to explore possible ways of implementing remote accessibility for ABB variable speed drives.

This study includes common theory of the working principles of a variable speed drive and examples of variable speed drive applications and their requirements. Following the theory of variable speed drives, network structures and requirements for establishing a working connection are presented.

The goal for the thesis work was to build and automate a test setup for variable speed drive’s remote access tests. It was necessary to run manual tests to define test cases before test automation could be done.

Some of the features were left out of the initial test automation for scheduling reasons. The result is a working test setup for variable speed drive’s communication.

Keywords: variable speed drive, test setup

Tiivistelmä

Tekijä:	Leevi Tiihonen
Otsikko:	Taajuusmuuttajan etäkäytön testaus
Sivumäärä:	40 sivua
Aika:	28.4.2023
Tutkinto:	insinööri (AMK)
Tutkinto-ohjelma:	sähkö- ja automaatiotekniikan tutkinto-ohjelma
Ammatillinen pääaine:	sähkövoimatekniikka
Ohjaajat:	product owner Sunilkumar Raina lehtori Eero Kupila

Insinöörityön aiheena oli testata taajuusmuuttajan käyttäytymistä osana verkkoympäristöä sekä luoda ja automatisoida testitapaukset näiden tulosten perusteella. Tulevaisuuden kehitystyön ja uusien tuotteiden myötä on tullut tarve tutkia mahdollisia tapoja toteuttaa ABB:n taajuusmuuttajan etäkäyttö.

Insinöörityö sisältää yleistä teoriaa taajuusmuuttajan toimintaperiaatteista sekä esimerkkejä taajuusmuuttajasovelluksista ja niiden vaatimuksista. Taajuusmuuttajien teorian jälkeen käsitellään verkkorakenteita ja vaatimuksia toimivan yhteyden muodostamiseksi.

Insinöörityön tavoitteena oli rakentaa ja automatisoida testilaitteisto taajuusmuuttajan etäkäyttöä vastaaville testeille. Ennen testiautomaatiota tuli kuitenkin määrittää testitapaukset manuaalisten testien yhteydessä.

Osa alkuperäisen suunnitelman mukaisista testiautomaation ominaisuuksista jätettiin toteuttamatta aikataulusyistä. Tuloksena saatiin toimiva testilaitteisto taajuusmuuttajan kommunikointia varten.

Avainsanat: taajuusmuuttaja, testilaitteisto

Contents

List of Abbreviations

DDCS: Distributed drives communication system. Protocol used in optical fibre communication.

DSI: Display Serial Interface. Defines high-speed serial interface between a host processor and a display module.

GUI: Graphical User Interface. GUI-application is designed using graphical features that make it easier to use.

I/O: Input/Output. Communication between an information processing system.

IP: Internet protocol. Network layer communications protocol.

IOT: Internet of things. Describes physical objects sharing data over internet.

PC: Personal Computer. Multi-purpose microcomputer.

PLC: Programmable logic controller. Industrial computer used to control automation processes.

URL: Universe Resource Locator. Colloquially termed a web address.

USB: Universal Serial Bus. Industry Standard that establishes specifications for cables.

USB-C: USB Type-C. 24-pin USB connector system with a rotationally symmetrical connector.

1 Introduction

Purpose of this study was to build a test setup for variable speed drive’s remote access for ABB Oy, Drives unit. Work was carried out at ABB Drives’ Pitäjänmäki campus in Helsinki. ABB is a global company whose main line of business involves electric power- and automation technology, with headquarters locating in Zürich, Switzerland. The company operates in over 100 countries with 105 000 employees globally and around 5000 employees in Finland. The main factory operations in Finland include Helsinki, Vaasa, Porvoo and Hamina. (1.)

Goals for the work were to build a test setup and automate tests for variable speed drive’s remote access. Important part of the test setup is the possibility to simulate issues normally occurring in a network. Test setup consists of one ACS380-drive, an optional Ethernet switch, and a network simulator. Ethernet switch allows for additional devices to be added as part of the configuration. The network simulator will be used to simulate and test effects of typical communication issues. At the other end of the network, a test laptop is used to communicate and monitor data transmission between the test laptop and the drive.

Remote access will have an impact on commissioning and maintenance of the drive since physical connection via Ethernet or USB is no longer needed. It is also more efficient to access the drive remotely when performing a routine inspection and monitoring the drive.

Thesis begins with a brief introduction into different variable speed drive types, applications, network types and topologies, followed by connectivity options. After the theory, test specifications and the actual manual testing for the drive are presented. At the end of the work there is an introduction into the final test program and its phases.

2 Variable Speed Drive

Basic function of a variable speed drive is to control the flow of energy from the mains into the process through the motor shaft. The motor shaft can be described using two quantities, torque, and speed. (2, p.8.) More specifically variable speed drive is a power electronic device that is designed to control electric motor's speed, torque, and position by adjusting supply voltage and frequency. Variable speed drives offer their user full control of the process and are responsible for bringing valuable energy savings by adjusting the electric motors according to each specific task. Variable speed drives can reduce the load of the supplying network and mechanical stress on the machines during starting.

Variable speed drives are controlled by control boards. Control board's main functions include controlling of semiconductors, data transmission, status reports and monitoring of safety and limit values (3). Variable speed drives use variables or parameters associated with the operation of the drive. These parameters allow the drive to be customized to suit specific motor and process. Variable frequency drive (VFD) refers to AC drives only, while variable speed drive (VSD) refers to both AC- and DC drives.

2.1 Operating Principle of a Variable Speed Drive

Variable speed drives use power electronics to adjust supply voltage and frequency. The conversion can be done by using semiconductor composites, such as diodes, thyristors, and transistors. A VSI (Voltage Source Inverter) -type variable speed drive consists of three main circuits: rectifier, intermediate DC -circuit and inverter. Figure 1 shows an illustration of a VSI-type variable speed drive.

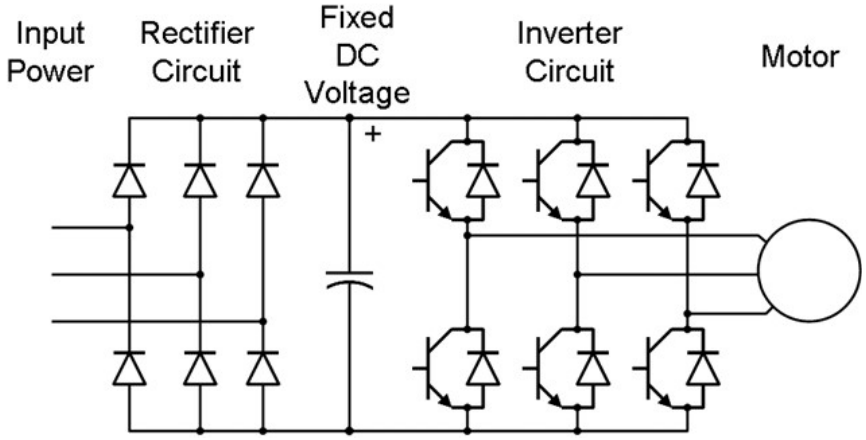


Figure 1. Variable speed drive with voltage intermediate circuit (VSI). Diodes on rectifier circuit and IGBT-modules on inverter circuit (3).

Full-wave rectifier consists of diodes and utilizes negative and positive half-cycles of the AC voltage. Diodes are passive components, and the diode bridge works without external control and has a good power factor. With diodes current only flows in one direction as there is no control option. Controlled rectifier bridges, such as half-wave rectifier consist of both diodes and thyristors and allow current and power flow in two directions. (4, p.44.)

The rectified DC voltage is conditioned in the intermediate circuit using capacitors or inductors depending on the type of drive. Most of the drives on the market use fixed DC voltage and therefore capacitors are used to smoothen out the ripple of the DC voltage as in figure 1.

Inverter circuit inverts the DC supply back to AC supply using active components, thyristors, and transistors. Typical way of inverting the DC supply is to use pulse width modulation (PWM), meaning the logic in which the inverter circuit's semiconductors are controlled to create the wanted output voltage for the motor. Nowadays IGBTs (Insulated-Gate Bipolar Transistors) are unmatched in terms of switching frequency, voltage duration and ease of control (4, p.45).

As mentioned, diodes in the rectifier circuit are passive components and do create harmonic current distortion into the supplying network. These diodes can also be replaced by active components to reduce harmonic current distortion and to regenerate the braking energy of the motor by returning it back to the drive through inverter circuit and into the supplying network through the rectifier circuit.

2.2 Drive Applications

Variable speed drives are used by various industries like water and wastewater, food and beverage, HVAC (heating, ventilation, air conditioning), mining, pulp and paper. Applications include example pumps, conveyors, vehicles, winches, and cranes.

By number most electric applications are based on transmission of liquid and gas. Pump and fan applications do not require good control or accuracy and are typically one motor applications with each motor having its own drive. These applications typically run on 40-100% of the nominal speed of the motor. The quality of the motor torque does not need to be perfect and small twitching is allowed excluding piston-type pumps. Conveyors also use variable speed drives and need breakaway torque depending on the material. (4, p. 60-61.)

For vehicle use the most important control variable is torque. Dimensioning is based on accelerating- and braking torque of a substantial inertial mass. Safety- and diagnostics requirements can also be strict when it comes to applications like trains and cars. (4, p. 65.)

Cranes and elevators require precise and smooth torque even in zero speeds, especially in crane applications where it is necessary to hold heavy loads still. Elevators require accurate positioning in order to stop and be levelled with the right floor. Applications responsible for human lives can be equipped with fail-safe options to guarantee safety. (4, p. 66.)

3 Categories of Networks

Computer network is an interconnected collection of autonomous computers. Computer network can satisfy broad range of purposes and requirements, allowing connectivity among users and processors regardless of the software. Computer networks enable sharing of resources and processing functions, alongside centralised control and management. (5, p.46.)

3.1 Area Networks

Depending on the operating geographical area, computer networks can be separated into three main categories: local-area network, metropolitan-area network, and wide-area network.

3.1.1 Local-Area Network (LAN)

A local-area network is generally a privately owned network within a single office or building. The main reason of designing a LAN is to be able to share resources and exchange information inside a defined network. LAN’s data rate is from 4 Mbps to hundreds of Mbps and it is typically used in a star or ring topology (5, p. 48-49.) Most widely used LAN technology is Ethernet.

Downside of using Ethernet is that it is primarily designed for smaller networks and even though it is possible to use longer cables, this can lead to unwanted coupling between signal paths (crosstalk). Crosstalk may lead into impairment in signals at the sender and receiver end. While generally Ethernet is a fast option for communication, this only applies to a smaller network. The speed in which data is transferred can slow down remarkably in comparison to traffic on the network. Troubleshooting in a network like this can also be challenging with so many different components, nodes, and links. (6.)

3.1.2 Metropolitan-Area Network (MAN)

Metropolitan-area network is designed to cover an entire city. It can be as simple as one network (cable television network) or multiple networks connected to each other to share resources (LAN-to-LAN) and is usually company owned. Typical LAN operates at a speed of 1.5 Mbps to 150 Mbps. (5, p. 49.)

3.1.3 Wide-Area Network (WAN)

Wide-area network is designed to interconnect computer systems over large geographical scope with speed ranging from 1.5 Mbps to 100 Gbps. WANs can utilize public, leased, or private communication devices and therefore expand unlimited miles. Internet is a good example of WAN reaching from continent to continent. (5, p. 50.)

3.1.4 Internetwork

When two or more networks are connected, they become internetwork. Individual networks can be connected into internetworks by using internetworking devices, for example routers and gateways. Figure 2 shows an illustration of individual networks connected to form an internetwork. (5, p. 49.)

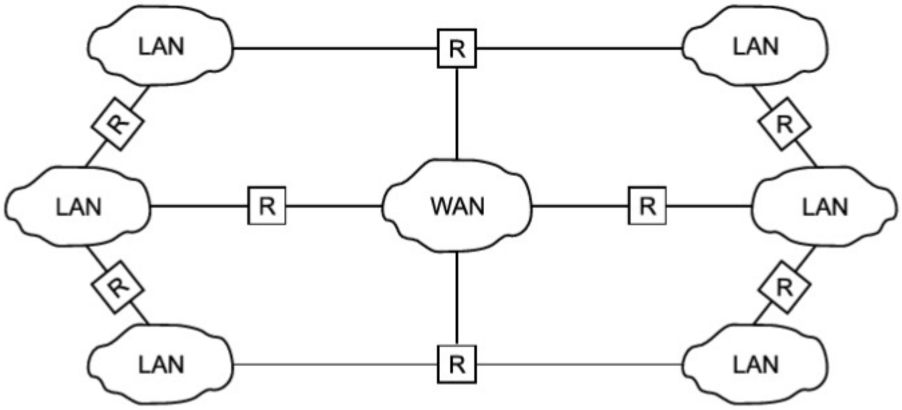


Figure 2. Internetwork (5, p. 49).

3.2 WLAN (Wireless Local-Area Network)

WLAN provides wireless network connectivity and allows devices to form a local-area network on a limited scale. WLAN relies on high-frequency radio waves rather than wired connections and often includes an Internet access point. WLAN is made to complement wired solutions and may establish an Internet connection through a gateway. Wireless Ethernet bridges are internetworking devices that allow devices connected to wired Ethernet to be connected to a wireless network. (7.)

Infrastructure is the most common type of WLAN. Infrastructure relies on the wireless devices to connect to WAP (Wireless Access Point) if they want to access the network. In purely infrastructure-less (ad hoc) network devices are connected peer-to-peer, communicating directly without a server. Wireless Distribution Systems (WDS) allow wireless interconnection of access points and allow the network to be expanded using multiple access points. (7.)

During the development of WLAN, other technical specifications have become available, including Bluetooth and Wi-Fi. Wi-Fi (wireless fidelity, 802.11b standard) offers fast transmission speed with long effective range and compatibility with various 802.11DSSS (direct-sequence spread spectrum) devices that are primarily used to reduce signal interference. In other words, Wi-Fi is included in WLAN and offers better signal strength and coverage. (7.)

3.3 Industrial Network

For a bigger scale of networking such as plants it is necessary for devices to work properly in any situation, send data to right place, right time, and in the right context. These industrial networks are built to monitor and control processes within a plant. Industrial networks can also be built using different approaches or levels depending on the needs of the specific plant.

Sensors are in the lowest level of these industrial networks. Designed to be inexpensive, sensors can only provide basic data from certain point of the process. Sensors can typically only provide position data from wherever they are attached without having the ability to act on it.

Fieldbus networks include all industrial networks that are designed for installation and are manufactured for where there is need of shared and distributed intelligence at each point of the network. In fieldbus networks there is either programmed device (PLC) or a separate microprocessor in charge of the process. (8.) Fieldbuses are designed to work in data transmission of the lowest level of factory automation. Fieldbuses are used to transmit data between smart sensors and devices. It is often necessarily to be able to add more advanced devices into the fieldbus such as I/O modules, programmable logics, and distributed control systems. I/O modules (Input/Output modules) act as mediators between the processor and input/output devices. Input/output devices can receive and send data. Drives have multiple options for fieldbus protocols depending on the type of drive and customer needs.

With approximately 20 of different standards existing, fieldbuses can be separated into sensor-, device-, and process buses. Sensor buses are applicable for data transmission produced by measurement sensors. Device buses are applicable for small device assembly such as motor drive groups' control. Process buses are used for larger scale operations controlling production lines and factories. (4, p. 279.) Each of these fieldbuses have multiple options for communication (fieldbus) protocols. Communication protocols provide real-time, closed-loop control between intelligent field devices and host systems.

Safety buses have similar features as regular fieldbus networks but are also designed to be faster and more reliable. Safety buses often require more components than regular fieldbus networks, with PLC specified for safety options. (8.)

4 Network Topology

In general, network topology is the arrangement of all devices and links of communication network. Network topologies can be separated into two sections, physical and logical topologies. Physical provides information of the physical construction of the network, layout of cables and devices. Logical topology provides information about the physical design of the network, communication mechanism for all the devices. (9.)

The simplest network topology is between two devices point-to-point connection. Star topology can be established by connecting more devices through bilateral connections to a central node. Downside of using star connection is cabling for every device in the network, also messages must always go through central device. Cabling problem can be solved by using just one cable or bus. (4, p. 272-273.)

When cable goes through device to device and closes to its original starting point, it is considered a ring topology. Disconnection in one point of the ring might not have any effect on the network’s normal function, this way ring topology can assure uninterrupted communication. (4, p. 272-273.)

It is often more efficient to separate bigger networks into smaller network segments, allowing the network to be expanded without the signal fading. Response time can be reduced by reducing the number of nodes in one segment. To reduce response time, it is also required that major part of the communication happens between nodes close to one another. Tree topology can be established by connecting multiple buses so that lower-level buses are connected to the upper level through a central device. (4, p. 273.) This way the network becomes more flexible and meets the requirements of the end-user. This can also save time and money during possible errors. Mesh topology can be used in more demanding networks, which implements all the topologies mentioned previously. Figure 3 shows an illustration of a mesh topology.

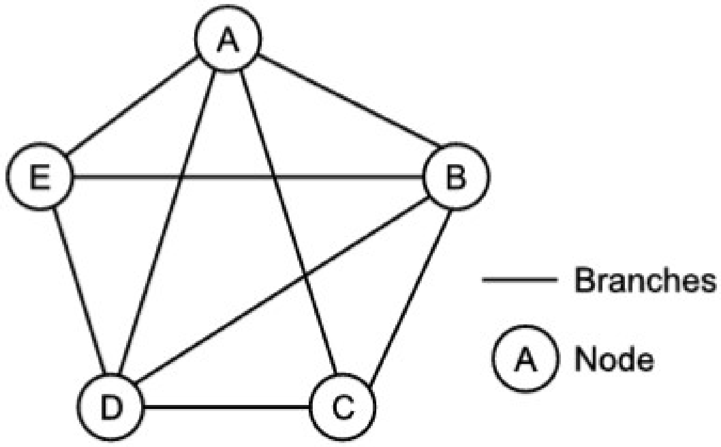


Figure 3. Mesh topology (5, p. 53).

In mesh topology devices have several links between each other to ensure communication speed, reliability, and security. Downside to this is limited ports of one device and cost of cabling. Fault identification and fault isolation becomes easier with point-to-point links. (5, p. 52-53.)

Cellular topology is applicable only in case of wireless media that does not require cable connection. In wireless media, each point transmits in a certain geographical area called a cell. Devices in specific cell communicate through a central hub and those hubs in different cells are interconnected and route data across the network. The data is transmitted in the CPDP-format (Cellular Digital Packet Data).

General troubleshooting is easy in cellular topology. In controversy hub-to-hub fault tracking can be difficult and in case a central hub fails, all the devices in the assigned range of cell will be affected. Figure 4 shows an illustration of a cellular topology.

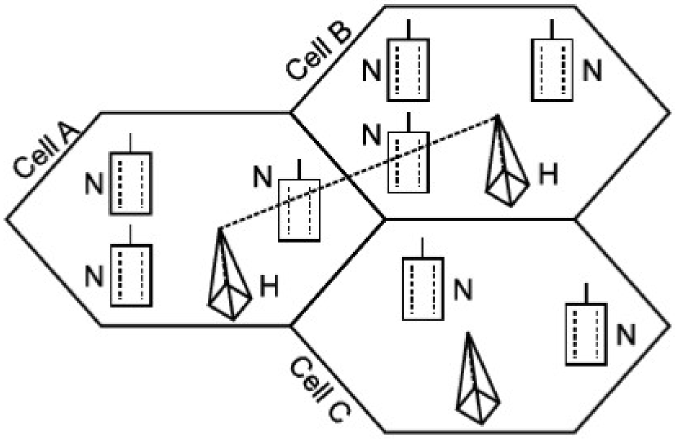


Figure 4. Cellular topology (5, p. 57).

5 Network Performance

Performance of a network includes variable factors: the volume of data successfully delivered, time it takes for the data packets to travel, variations in latency, number of packets failed to reach their destination, packets arriving in different order than what they were originally sent in.

There are two techniques for sharing the resources of communication networks, “circuit” and “packet” technique. Circuit technique consists of reserving the resources for the communication. In packet technique communications occur without any reservation, information being transferred in the form of independent packets. In short this is the main difference between (public switched) telephone network and IP-network. (10, p. 2.) Figure 5 shows an illustration of “circuit”- and “packet” techniques.

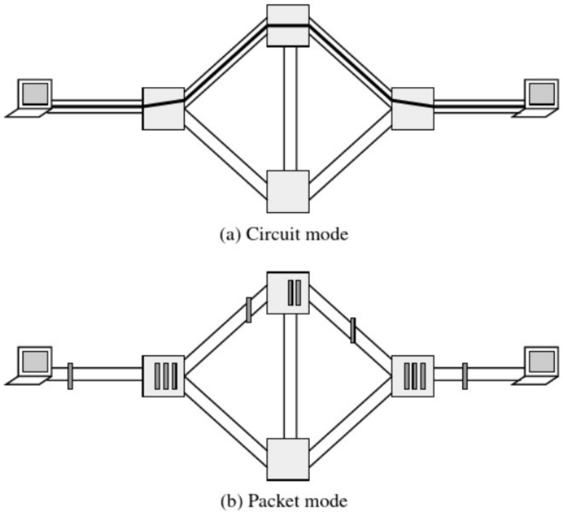


Figure 5. Communication techniques (10, p. 2).

Differences between the two communication techniques can be seen as the principle of bandwidth reservation versus that of bandwidth sharing. Questions of accessibility (call reject rate) against speed (bit rate) and integrity (packet delay, packet loss rate). (10, p. 2.)

5.1 Packet Delay and Packet Loss

In general packet delay, also referred to as latency describes the time delay of a data packet between two places. There are many factors contributing into packet delay like other packets which essentially lead to congestion in the network. For their control mechanics, wired connections have lower latency compared to satellite and wireless connections. Latency affects the Quality of Experience (QoE), for example in case of high-latency audio or video call the voices of each contestant can appear out of sync.

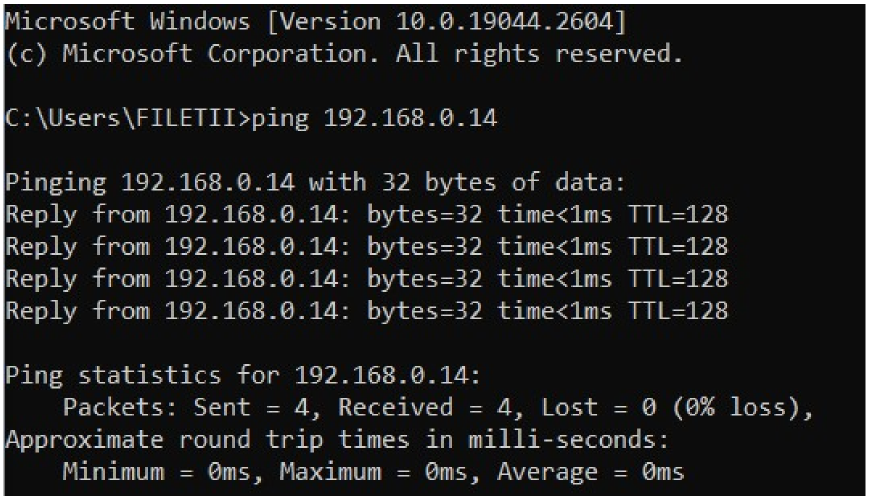
Network optimization and cabling can improve poor network experience caused by packet delay. Fibre cabling relies on light rather than electricity. Replacing the coaxial cabling with fibre cabling can improve network speed and handle higher amount of data transmitted in a given amount of time (bandwidth). With less bandwidth, there is a higher chance of latency. Another way of improving packet delay is to have the nodes closer to customer and reducing customer serving group sizes.

Packet loss describes lost packets of data that are not reaching their destination after being transmitted. Packet loss can occur due to multiple factors, network congestion, hardware issues, software bugs, overtaxed devices, security threats, or by deficient structure. Packet loss may have different impact on different applications. For example, when downloading data files, a 10% packet loss might not have a serious impact on the actual downloading time. Higher packet loss rate can cause delays to be worse and prevent packets from reaching their destination. (11.)

Networking monitoring is the easiest way to prevent packet loss. User can detect the packet loss by using traceroute command line -tool along with ping command that can assess network delays. Regularly scanning the network for possible defects and making sure routers can handle the capacity. (11.)

5.2 Network Quality Test Programs

There are several software programs available to test network quality in terms of packet delay and loss rate. Packet delay can be tested using ping command that pings the wanted device using its IP-address. Ping measures device’s accessibility in the network. The response shows the URL that the user is pinging, the IP-address associated with that URL and the size of the packets being sent. Figure 6 shows an example of ping command.



```
Microsoft Windows [Version 10.0.19044.2604]
(c) Microsoft Corporation. All rights reserved.

C:\Users\FILETII>ping 192.168.0.14

Pinging 192.168.0.14 with 32 bytes of data:
Reply from 192.168.0.14: bytes=32 time<1ms TTL=128
Reply from 192.168.0.14: bytes=32 time<1ms TTL=128
Reply from 192.168.0.14: bytes=32 time<1ms TTL=128
Reply from 192.168.0.14: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.0.14:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

Figure 6. Ping command.

The four lines of the ping command show the replies of each individual packet, including the time it took for the response in milliseconds and the time-to-live (TTL) of the packet, which describes the time that must pass before the packet is discarded.

Same way packet delay can be tested using the ping command, packet loss can be measured by pinging the wanted object for certain number of times and counting down the failed responses.

Traceroute command allows the user to determine the path between two connections. Often the connection must go through multiple routers. The traceroute command gives the IP-address of those routers.

In any case there appears to be problems with the connection, deep packet inspection (DPI) can be established to examine and manage the network traffic. Deep packet inspection locates, identifies, blocks, or re-routes specific packets. (11.)

Network simulation is a technique where a software program replicates the behaviour of a real network. Network simulators can be used to study the effects of a real computer network by interfering with the communications.

There are various open-source- and commercial network simulators to choose from, with each their own operating principles and objectives.

6 Variable Speed Drive's Connectivity

Variable speed drives are an essential part of any industrial process and have multiple options considering control location of the drive. These controls are either local or external. Local control instructs the drive to ignore external control and only abide by the keypad. Local control means user is within sight of the drive. Local control uses the drive's internal panel/keypad, or a separate laptop connected to the drive through a cable.

External control instructs the drive to ignore local control. External control means user is not within sight of the drive. External control uses external panel/keypad, I/O-terminal (digital- or analog inputs) or fieldbus. Digital inputs are used to interface the drive with devices, such as push buttons and selector switches. Analog inputs are used to interface the drive with an external signal, such as potentiometers. (12.) Figure 7 shows illustration of local and external control locations of ABB ACS380 machinery drive.

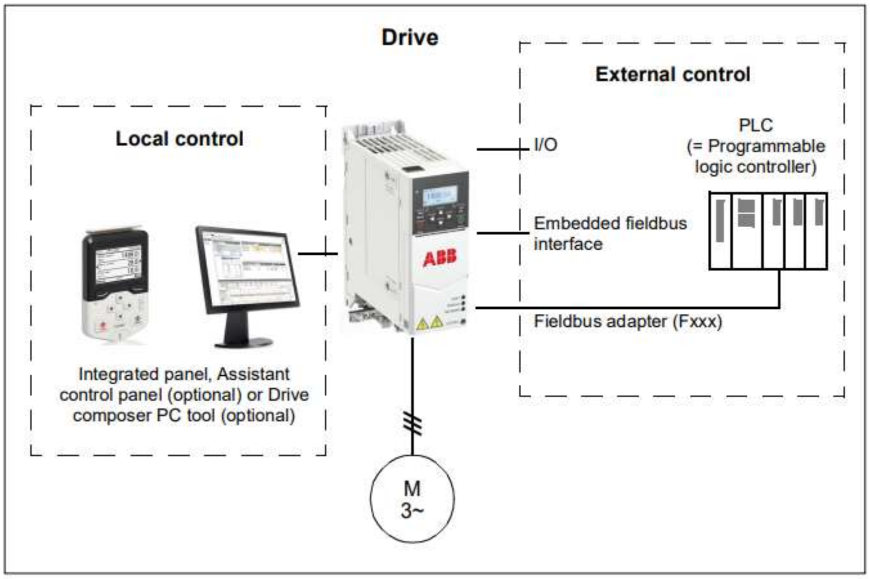


Figure 7. Control locations (ACS380) (13, p. 50).

ABB’s variable speed drives can be connected to a network for setup, operation, and monitoring means through multiple different options and protocols. In ACS380 drives local control mode’s control commands can be given from integrated panel, assistant control panel or by attaching PC to the drive through USB-cable and using ABB’s Drive Composer -PC tool. This type of connection is usually used in commissioning and maintenance of the drive. (13, p. 50.) Assistant control panel is also available with Bluetooth connectivity which makes it quick and easy to tune and monitor the drive once paired with specific mobile device.

External control mode can be established using various methods, such as I/O terminals or an external panel. Drive can be connected to an external control system through communication link using either fieldbus adapter module or embedded fieldbus interface. (13, p. 51.)

7 Test Specification

The work began by building a test setup using ABB ACS380 machinery drive and running manual tests for better understanding of possible issues and behaviour regarding the connection. Manual tests were done using Ethernet connection and a network simulator was used to simulate issues normally occurring in a network. After this it was needed to automate the tests using a test automation software. In the future this thesis can be used for further develop the drive’s communication and as a reference on creating test instructions for future drives. Figure 8 shows observation figure of the test setup and its connections.

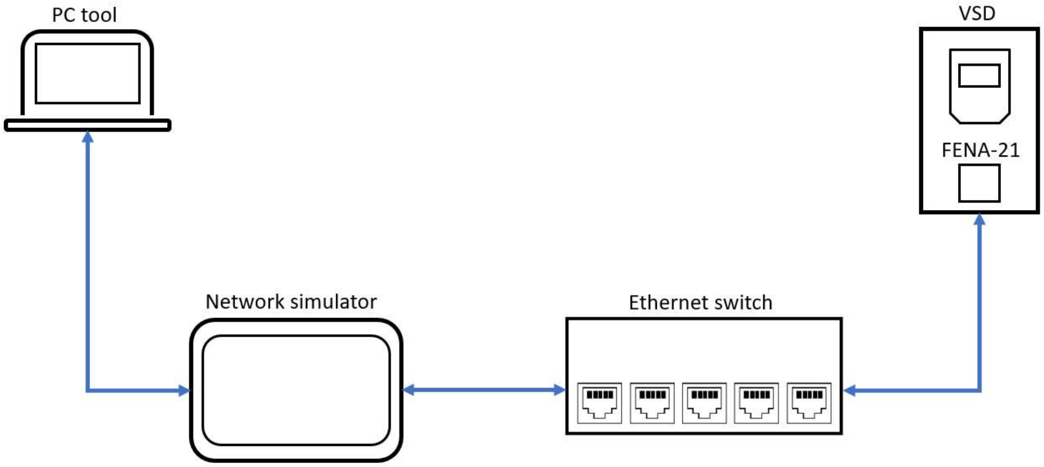


Figure 8. Test setup.

PC tool used in this case was ABB's own Drive Composer -software for its easy access to the drive's parameters. Network simulator was connected between the laptop and the drive to disturb the data transmission. Connection between network simulator and the drive goes through an Ethernet switch with 8 ports for optional devices like PLC to control the drive. The drive was connected to the Ethernet switch through a fieldbus adapter module FENA-21.

7.1 Hardware

Network simulator was used to simulate a real-life network environment. For test use this solution proved to be perfect since the network simulator could easily be operated from the display and allowed its user to demonstrate possible issues and communication behaviour of a real network. The test setup itself included one drive, fieldbus adapter module, Ethernet switch, network simulator, additional USB-Ethernet adapter, and a laptop.

Drive chosen for this work was ABB's ACS380-machinery drive. ACS380 is typically used in food and beverage, material handling and textile industry with applications such as mixers, conveyors, and tower cranes. The demo case used runs on the mains single phase 230 V supply and includes the drive (0.25 kW) and a 3-phase squirrel cage motor (0.09 kW).

In general fieldbus adapter modules enable communication between systems. In this work FENA-21 fieldbus adapter module was used for Ethernet connection. FENA-21 can be used for EtherNet/IP, MODBUS TCP and PROFINET I/O protocols. The fieldbus adapter module has two Ethernet ports and enables daisy chain wiring scheme, where two or multiple drives can be wired together and into same network. Fieldbus protocol used in this work was PROFINET. PROFINET is an industry technical standard for data communication over industrial Ethernet. Even though fieldbus is not used in its full capacity in this work and there is not a separate PLC controlling any process, it is still necessary to define the used fieldbus protocol and other relative parameters such as IP-address for the drive.

Ethernet switch used in this work was Zyxel’s 8-port desktop gigabit Ethernet switch. The switch was unmanaged, so there was no need to define any IP-address for the switch itself. This makes it easier for all users to configure the test setup. The switch allows additional PLC to be configured as part of the test setup in the future. In this case CAT6 (category 6) standardized twisted pair cable was used for the connections, but any of the ANSI/TIA 568 -standard defined Ethernet cabling could be used.

7.2 Network Simulator

The network simulator enables its user to define the time it takes for the packets to travel, and the number of packets failed to reach their destination. The program for the network simulator was already made in Kivy by ABB. Kivy is a free open-source Python framework and enables its user to build and distribute Python cross-platform GUI applications. The main function of the program was to influence packets sent from laptop to the drive by dropping the packets, delaying or re-arranging them and the other way around.

7.2.1 Raspberry Pi 4 Model B

The hardware used for the network simulator was Raspberry Pi 4 model B with 2 GB RAM. Raspberry Pi is a small, single board computer with dual display and allows its user to create and demonstrate basic programs. Raspberry Pi is widely used for educational purposes.

MicroSD card could be used to download any wanted operating system to the computer. Operating system in this case was Raspberry Pi OS, that could easily be downloaded using Raspberry Pi Imager. Figure 9 shows Raspberry Pi 4 model and its main components.

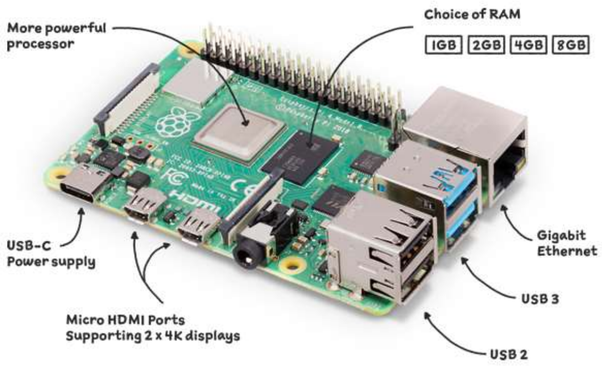


Figure 9: Raspberry Pi 4 model B (14).

7.2.2 Network Simulator’s Touchscreen Display

Raspberry Pi touchscreen display was added as part of the network simulator. The touchscreen display can be used for interactive projects and made it possible to set wanted changes on the network simulator while running. This solution saves time in manual testing. Figure 10 shows all the main components included in the Raspberry Pi 7’’ display order.

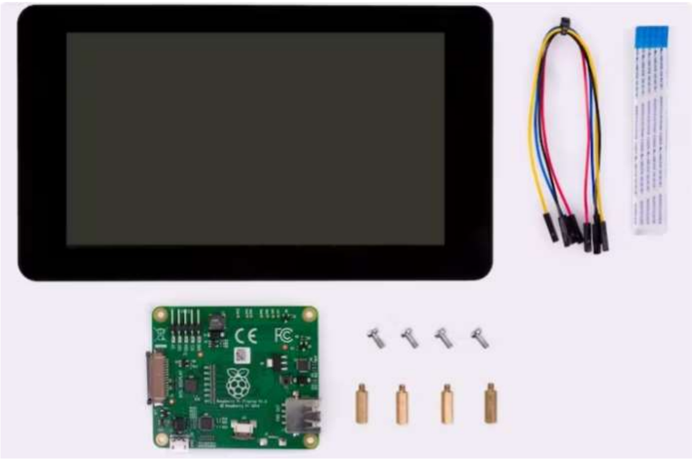


Figure 10: Raspberry Pi 7’’ touchscreen display, adapter board, mounting screws for Raspberry Pi, four pieces of coloured jumper wires, and a data (DSI ribbon) cable (15).

7.2.3 Raspberry Pi 7’’ Touchscreen Setup

The display can be connected to the Raspberry Pi using separate adapter board that handles power and signal conversion from the display to the serial (DSI) port of the Raspberry Pi. Raspberry Pi OS provides touchscreen drivers, giving the user access to full functionality of the display without the need of keyboard and mouse (15). Raspberry Pi’s GPIO pins could be found from the documentation of the board while the adapter board’s pin designations were printed into the board itself. Figure 11 shows close up of Raspberry Pi’s GPIO pins and their designations.

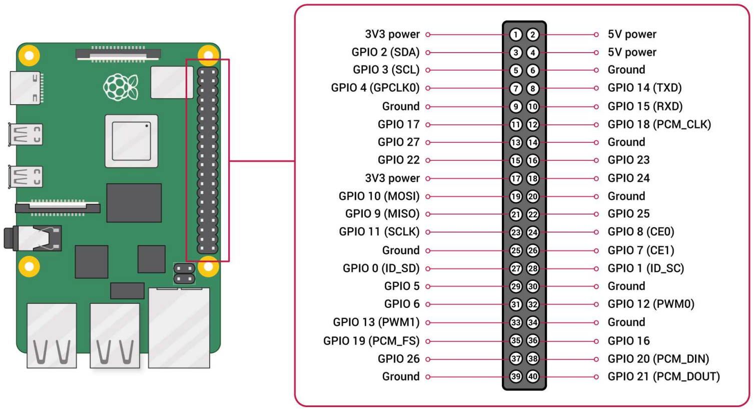


Figure 11. Raspberry Pi 4 GPIO pins (16).

For the display to work, DSI ribbon cable must be connected between the Raspberry Pi and the adapter board of the display. The white USB-C cable was the 15 W (5 V/3 A) power supply of the Raspberry Pi and could also be used to power the adapter board as described: red jumper wire connected between the 5 V supply and black jumper wire connected between the ground of both boards. Figure 12 shows all the connections between Raspberry Pi and the adapter board.

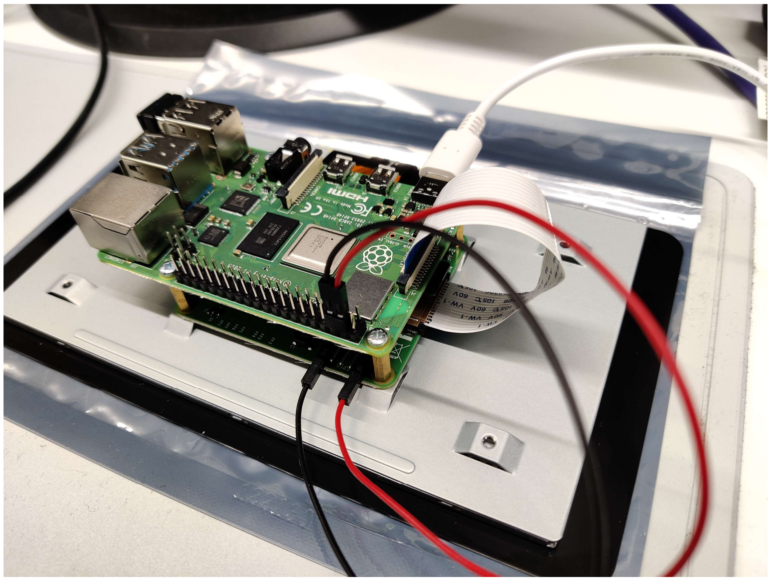


Figure 12: Raspberry Pi 4 Model B connected on top of Raspberry Pi 7’’ display.

7.2.4 Network Simulator Installation Box

An installation box was built for the network simulator to improve electrical safety and to avoid any unnecessary touches on the connections and on the Raspberry Pi. Touches without electrostatic discharge (ESD) safe clothing could end up damaging the board. Figure 13 shows the test setup built according to the observation figure shown earlier.

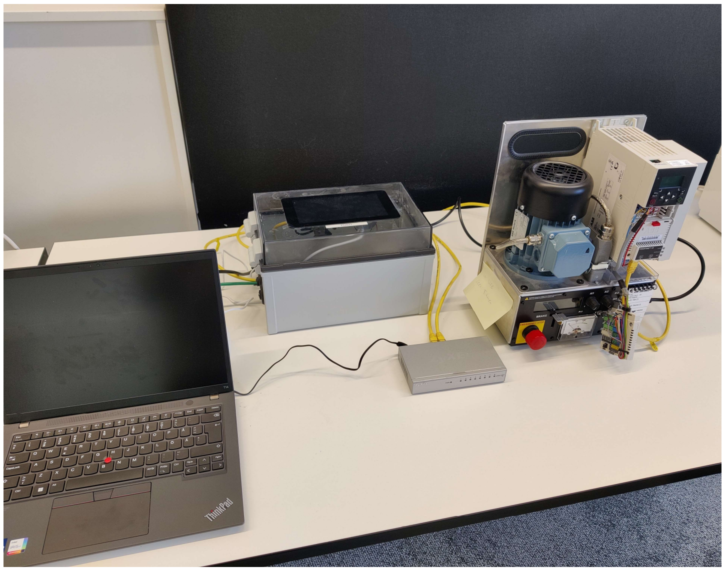
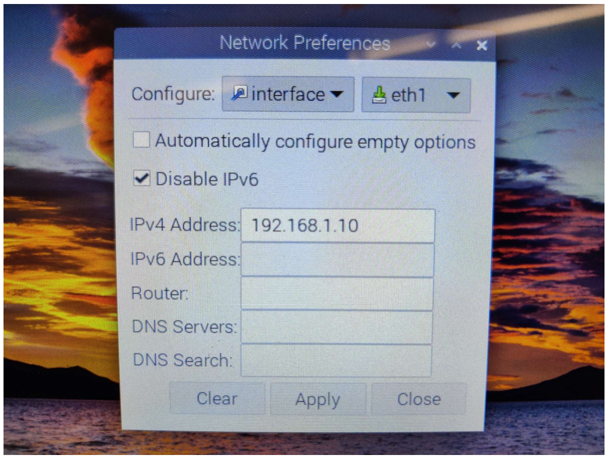


Figure 13. Test setup.

The original installation box for the network simulator was ordered from Fibox Oy (model CAB PC 203018 T 8113031) which is a company specialised in manufacturing housings and precision-pressed plastic parts for electrical equipment. The installation box measures in 200 x 300 x 180 millimetres and can fit instructions and other optional accessories. Installation box was modified to have three input holes for USB-C-, Ethernet- and USB cables. The installation box had a quadrilateral shaped hole cut into the cover to fit the network simulator and have the touchscreen visible for interaction. USB-Ethernet adapter was used to place the network simulator in between the laptop and the switch, since Raspberry Pi has only one Ethernet port available.

7.3 Network Communication Settings

Communication settings and ports needed to be configured for the network simulator to function and interfere the communication. These settings were done in both, Raspberry Pi, and Drive Composer. Only one port of the Raspberry Pi needed to be in the same network as the PC. In this case, used IP-addresses for the ports were 192.168.1.10 and 192.168.0.20. Both used ports' IP-address needed to be configured on the Raspberry Pi's wireless and wired network settings. IP-address for the PC was 192.168.0.14, which can be changed in network and internet settings of the PC on Windows 10. Figure 14 shows network preferences of the Raspberry Pi OS and the IP-address of one of Raspberry Pi's ports.



Configure: interface eth1

☐ Automatically configure empty options

☑ Disable IPv6

IPv4 Address: 192.168.1.10

IPv6 Address:

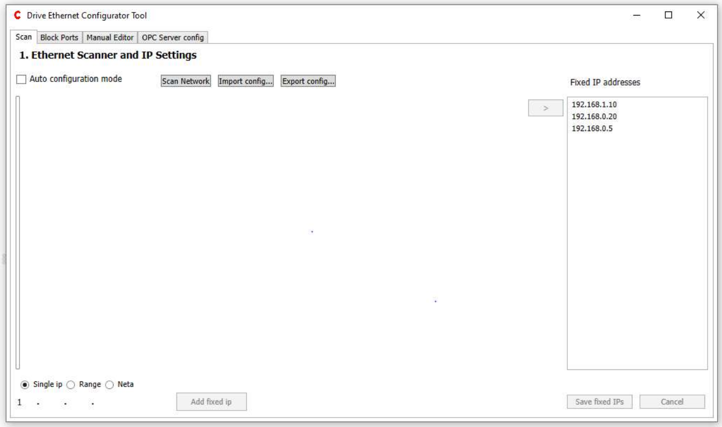
Router:

DNS Servers:

DNS Search:

Figure 14. Raspberry Pi OS wireless and wired network settings.

After setting the IP-addresses for the ports in Raspberry Pi, those same ports, including the drive's IP-address needed to be configured in the Drive Composer's communication settings on the welcome window. Figure 15 shows Ethernet and IP settings of Drive Composer pro v. 2.8.1 including fixed IP-addresses.



Drive Ethernet Configurator Tool

1. Ethernet Scanner and IP Settings

□ Auto configuration mode

Fixed IP addresses

192.168.1.10
192.168.0.20
192.168.0.5

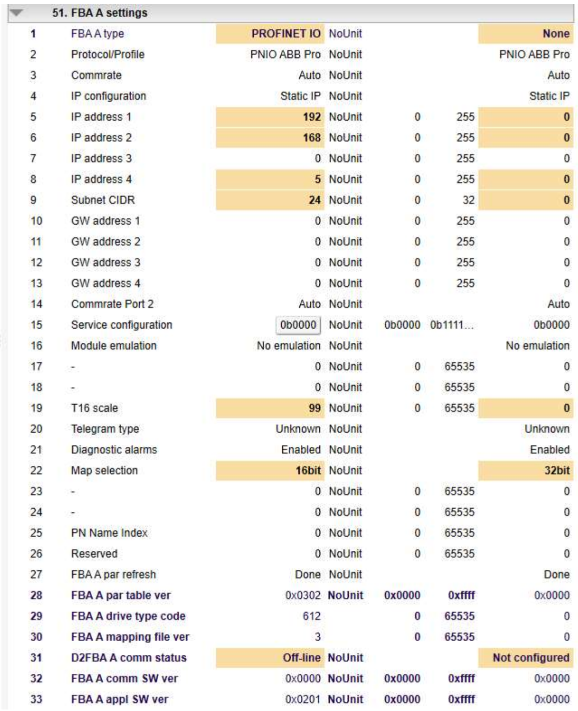
● Single ip ○ Range ○ Neta

1 . . .

Figure 15. Ethernet scanner and IP settings of Drive Composer pro v. 2.8.1.

The network simulator program itself can be executed using Raspberry Pi’s terminal by sudo (super user do) command. Sudo is a Linux command that allows programs to be executed as a super user, temporarily giving a user account root privilege.

In ACS380 parameter group 50 user must enable fieldbus adapter before any changes to fieldbus adapter (FBA) A settings in group 51. In these settings user can configure the drive’s fieldbus related parameters, for example fieldbus protocol, IP-address, and subnet CIDR (Classless Inter-Domain Routing). After configuring the drive in terms of fieldbus related parameters, user must update the changes from the same parameter group (parameter 51.27). Figure 16 shows view of drive’s parameters in group 51 in Drive Composer. Orange filling representing changed parameters.



	51. FBA A settings					
1	FBA A type	PROFINET IO	NoUnit			None
2	Protocol/Profile	PNIO ABB Pro	NoUnit			PNIO ABB Pro
3	Commrate	Auto	NoUnit			Auto
4	IP configuration	Static IP	NoUnit			Static IP
5	IP address 1	192	NoUnit	0	255	0
6	IP address 2	168	NoUnit	0	255	0
7	IP address 3	0	NoUnit	0	255	0
8	IP address 4	5	NoUnit	0	255	0
9	Subnet CIDR	24	NoUnit	0	32	0
10	GW address 1	0	NoUnit	0	255	0
11	GW address 2	0	NoUnit	0	255	0
12	GW address 3	0	NoUnit	0	255	0
13	GW address 4	0	NoUnit	0	255	0
14	Commrate Port 2	Auto	NoUnit			Auto
15	Service configuration	0b0000	NoUnit	0b0000	0b1111...	0b0000
16	Module emulation	No emulation	NoUnit			No emulation
17	-	0	NoUnit	0	65535	0
18	-	0	NoUnit	0	65535	0
19	T16 scale	99	NoUnit	0	65535	0
20	Telegram type	Unknown	NoUnit			Unknown
21	Diagnostic alarms	Enabled	NoUnit			Enabled
22	Map selection	16bit	NoUnit			32bit
23	-	0	NoUnit	0	65535	0
24	-	0	NoUnit	0	65535	0
25	PN Name Index	0	NoUnit	0	65535	0
26	Reserved	0	NoUnit	0	65535	0
27	FBA A par refresh	Done	NoUnit			Done
28	FBA A par table ver	0x0302	NoUnit	0x0000	0xffff	0x0000
29	FBA A drive type code	612		0	65535	0
30	FBA A mapping file ver	3		0	65535	0
31	D2FBA A comm status	Off-line	NoUnit			Not configured
32	FBA A comm SW ver	0x0000	NoUnit	0x0000	0xffff	0x0000
33	FBA A appl SW ver	0x0201	NoUnit	0x0000	0xffff	0x0000

Figure 16. Fieldbus adapter A settings in Drive Composer.

7.4 Software

Software used in this work can be separated into two main categories according to their purpose: communication and test automation. Communication software were used for establishing and monitoring communication between devices. Test automation software were used to automate the test cases according to manual test results.

7.4.1 Communication Software

Drive Composer pro v. 2.8.1 was used to connect between the laptop and the drive. Drive composer is ABB’s own software tool used for start-up and commissioning of the drive. Drive Composer enables parameter changes, monitoring of the drive and tune of the process. Laptop can be connected to the drive using Ethernet or USB cable. Overall Drive Composer gives a wider view of the drive compared to the control panel.

Windows Command Prompt was used to ping all the devices in the network. Windows Command Prompt is intended for running computer programs that do not have a graphical user interface. Wireshark could also be used to keep track of communication between laptop and other devices in the network when facing any issues. Wireshark is a free open-source network protocol analyzer widely used for troubleshooting and analyzing networks.

Communication tester is an executable program made by ABB used to keep time of how long the connection handshake and specific tasks take between the test laptop and the drive. Communication tester was run from Windows Command Prompt.

7.4.2 Test Automation Software

The final test automation was done using Robot Framework, which is a generic open-source automation framework that can be extended with multiple libraries to test virtually any system or device. Robot Framework can be used for test automation and robotic process automation (RPA). Robot Framework has an easy syntax, and its capabilities can be extended by libraries from Python, Java, and other programming languages. Robot Framework is used by multiple technology companies from component to software testing. (17.) ABB has made their own installation instructions for the Robot Framework.

The actual code was written in Visual Studio Code that is redefined and optimized for building and debugging modern web and cloud applications. Visual Studio Code enabled the code to be debugged in sections and allowed easy troubleshooting.

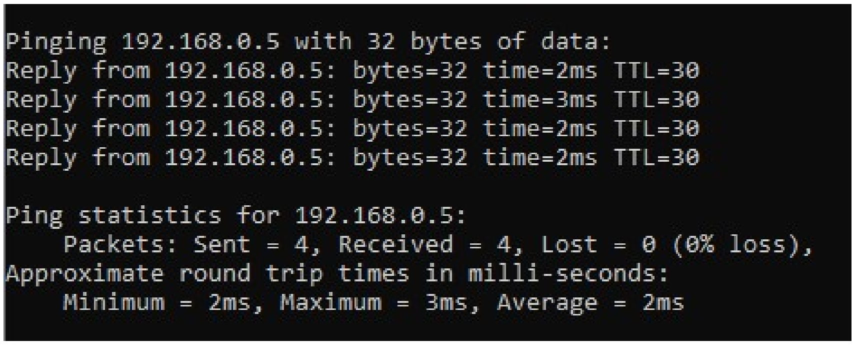
8 Manual Testing

In network planning it is common to come across network interference by multiple different reasons, such as environment in which the devices are located or by other surrounding devices. Way to approach this problem is to build the network to minimize and avoid interference. In most cases this interference between devices can be noted and is accepted to some extent. Interfering can lead to transmission issues and data loss. Ethernet cables' quality of the signal can degrade when placed near power cables. This is also why ABB has included recommendations for cabling in their manuals.

Manual tests were run to study the effects of packet delay and packet drop, but also to find out the minimum requirements for the Drive Composer to work and create connection between the PC and the drive. Packet delay and packet drop were first tested alone and then together, to find the point in which the PC and the drive fail to communicate sufficiently. Manual tests were done using ABB's premade software tool to figure out how the communication would behave while network simulator is interfering with the communication.

8.1 Packet Delay Demonstration

Since the network simulator did not have an option for choosing the way of interference, parameters chosen in network simulator affected both, upload (from PC to drive), and download (from drive to PC). Time delay set in network simulator would also be multiplied by this effect when pinging the drive. All the measurements were conducted three times and the average time stamps were added to an Excel table. The drive could be pinged from the PC using Command Prompt's ping command. This was used to quickly confirm the connection between devices. Figure 17 shows ping command with no delay.

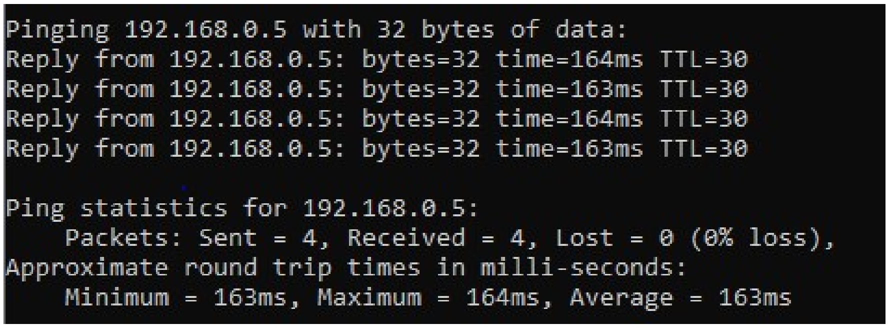


```
Pinging 192.168.0.5 with 32 bytes of data:
Reply from 192.168.0.5: bytes=32 time=2ms TTL=30
Reply from 192.168.0.5: bytes=32 time=3ms TTL=30
Reply from 192.168.0.5: bytes=32 time=2ms TTL=30
Reply from 192.168.0.5: bytes=32 time=2ms TTL=30

Ping statistics for 192.168.0.5:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 2ms, Maximum = 3ms, Average = 2ms
```

Figure 17. Ping command.

For the pinging, typical time it takes for the PC to ping the drive is around 2-3 milliseconds without any external interference. This value worked as a reference for different testing scenarios while using the network simulator. Figure 18 shows ping command with time delay of 81 ms affecting both, upload, and download (total 164 ms).



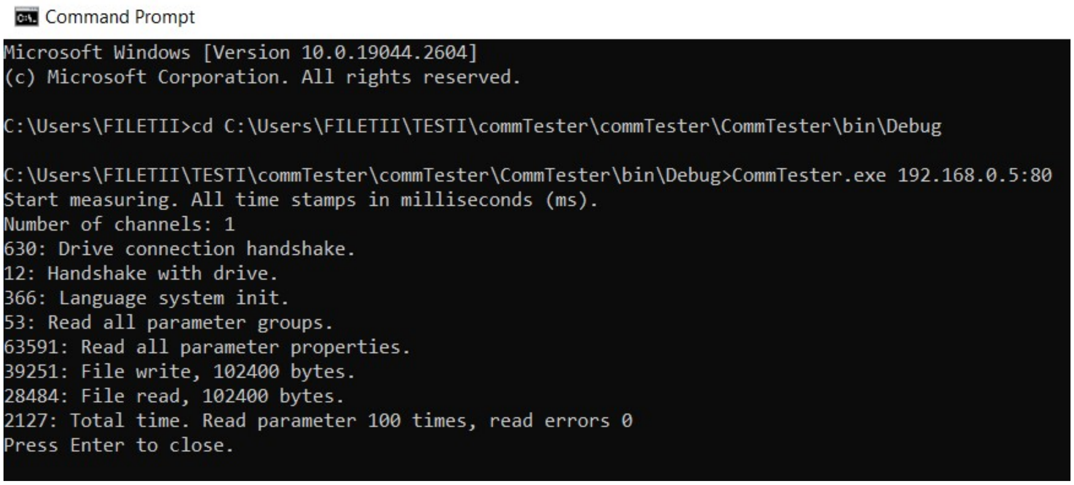
```
Pinging 192.168.0.5 with 32 bytes of data:
Reply from 192.168.0.5: bytes=32 time=164ms TTL=30
Reply from 192.168.0.5: bytes=32 time=163ms TTL=30
Reply from 192.168.0.5: bytes=32 time=164ms TTL=30
Reply from 192.168.0.5: bytes=32 time=163ms TTL=30

Ping statistics for 192.168.0.5:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 163ms, Maximum = 164ms, Average = 163ms
```

Figure 18. Ping command.

8.2 Communication Tester Demonstration

An ABB made communication tester was used to test communication between the test laptop and the drive. Communication tester is an executable program that performs eight tasks between the test laptop and the drive. These tasks included ‘connection handshake’ and ‘handshake with the drive’, which are an essential part of any automated process, and are used to authenticate the signal between the two devices. Following the handshakes were ‘language system initialization’ and ‘reading of all parameter groups’. The following tasks would take more time to run, including ‘reading of all parameter properties’, ‘file write’, ‘file read’ and ‘read parameter 100 times, read errors 0’. The time stamps in milliseconds represent the time it took for the test laptop and the drive to establish connection and perform specific tasks. The command with which the communication tester is run, must include the IP-address of the drive (192.168.0.5) and the specific port in use (:80). Figure 19 shows communication tester results with no interference.



```
Microsoft Windows [Version 10.0.19044.2604]
(c) Microsoft Corporation. All rights reserved.

C:\Users\FILETII>cd C:\Users\FILETII\TESTI\commTester\commTester\CommTester\bin\Debug

C:\Users\FILETII\TESTI\commTester\commTester\CommTester\bin\Debug>CommTester.exe 192.168.0.5:80
Start measuring. All time stamps in milliseconds (ms).
Number of channels: 1
630: Drive connection handshake.
12: Handshake with drive.
366: Language system init.
53: Read all parameter groups.
63591: Read all parameter properties.
39251: File write, 102400 bytes.
28484: File read, 102400 bytes.
2127: Total time. Read parameter 100 times, read errors 0
Press Enter to close.
```

Figure 19. Executed communication tester.

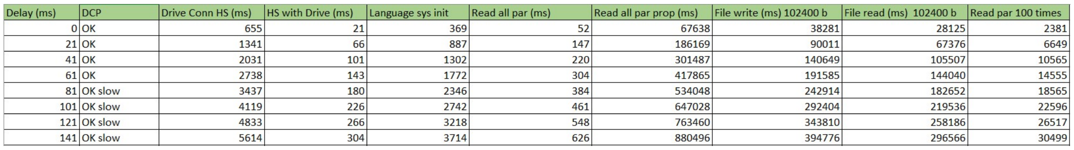
8.3 Test Results

Tests were taken separately between packet delay and packet drop to record effects of both variables on the communication. After this, both variables were tested simultaneously to figure out the minimum requirements for the connection between the drive and the test laptop.

8.3.1 Test Results for Packet Delay

The results were taken three times and put into an Excel table. Starting from no delay with 20 millisecond steps up to 141 milliseconds. Same thing was done with the packet drop percentage with one percentage point steps, from no packet drop up to 6 percentage packet drop. “DCP” in the Excel table stands for Drive Composer pro and describes the functionality of the program during different scenarios. Table 1 shows average time stamps for each communication tester task at certain point of packet delay.

Table 1. Communication tester tasks and time stamps.



Delay (ms)	DCP	Drive Conn HS (ms)	HS with Drive (ms)	Language sys init	Read all par (ms)	Read all par prop (ms)	File write (ms) 102400 b	File read (ms) 102400 b	Read par 100 times
0	OK	655	21	369	52	67638	38281	28125	2381
21	OK	1341	66	887	147	186169	90011	67376	6649
41	OK	2031	101	1302	220	301487	140649	105507	10565
61	OK	2738	143	1772	304	417865	191585	144040	14555
81	OK slow	3437	180	2346	384	534048	242914	182652	18565
101	OK slow	4119	226	2742	461	647028	292404	219536	22596
121	OK slow	4833	266	3218	548	763460	343810	258186	26517
141	OK slow	5614	304	3714	626	880496	394776	296566	30499

It took a lot of time for Drive Composer to open and change the parameters when the delay was 61 milliseconds, and at that point the connection handshake took over 2700 milliseconds. With the delay of 141 milliseconds, it took communication tester 30 minutes to run the tests.

Figures 20 and 21 show graphs of communication tester's results. First graph shows less time-consuming tasks performed at the beginning of the tester. For the second graph tasks, time stamps vary from 2000 ms up to 880 000 ms.

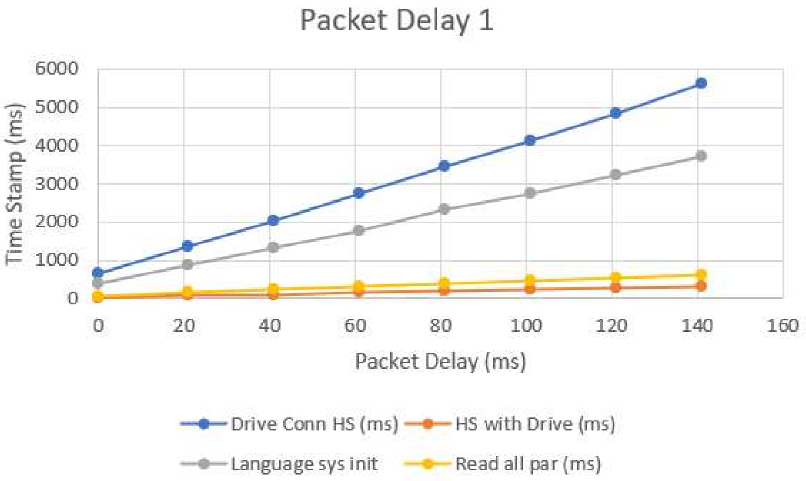


Figure 20. Graph of time stamps as a function of packet delay.

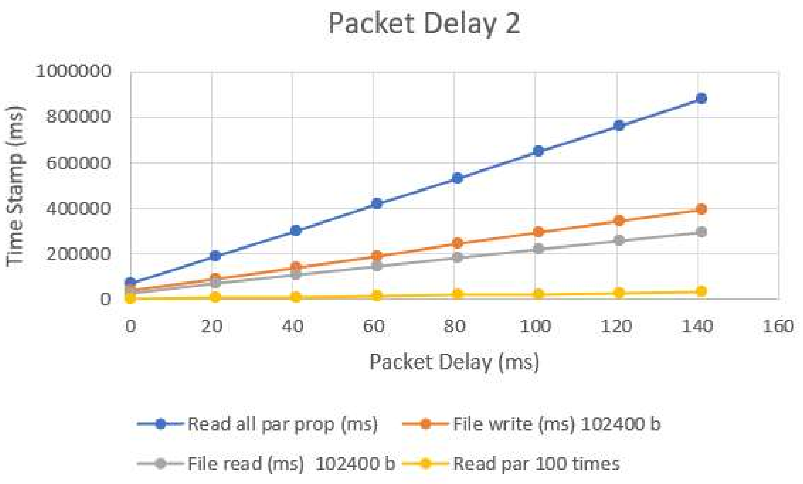


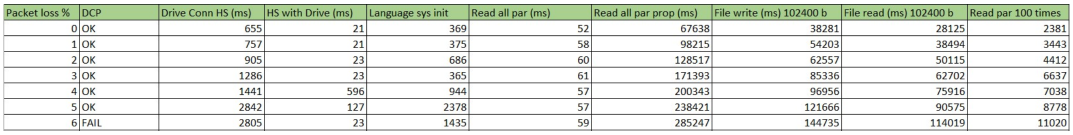
Figure 21. Graph of time stamps as a function of packet delay.

As expected, the packet delay had a straight negative effect on the time stamps of the communication tester. However, the time added into communication tester’s tasks was not directly proportional to the given time delay of the network simulator.

8.3.2 Test Results for Packet Drop

Different applications are affected different ways by packet drop. In this case packet drop had unpredictable effects on the communication tester’s tasks, which can be seen as variation on the time stamps. During handshakes the time stamps would experience some visible variation between 4 and 5 % packet drop. With the packet drop of 6 % Driver Composer would constantly disconnect from the drive or give a notification of change in configuration referring to disconnection. Table 2 shows average time stamps for each communication tester task at certain point of packet drop percentage.

Table 2. Packet drop's effect on the time stamps of each task performed by the communication tester.



Packet loss %	DCP	Drive Conn HS (ms)	HS with Drive (ms)	Language sys init	Read all par (ms)	Read all par prop (ms)	File write (ms) 102400 b	File read (ms) 102400 b	Read par 100 times
0	OK	655	21	369	52	67638	38281	28125	2381
1	OK	757	21	375	58	98215	54203	38494	3443
2	OK	905	23	686	60	128517	62557	50115	4412
3	OK	1286	23	365	61	171393	85336	62702	6637
4	OK	1441	596	944	57	200343	96956	75916	7038
5	OK	2842	127	2378	57	238421	121666	90575	8778
6	FAIL	2805	23	1435	59	285247	144735	114019	11020

Figures 22 and 23 show graphs of communication tester's results. First graph shows less time-consuming tasks performed at the beginning of the tester. For the second graph tasks, time stamps vary from 2000 ms up to 280 000 ms.

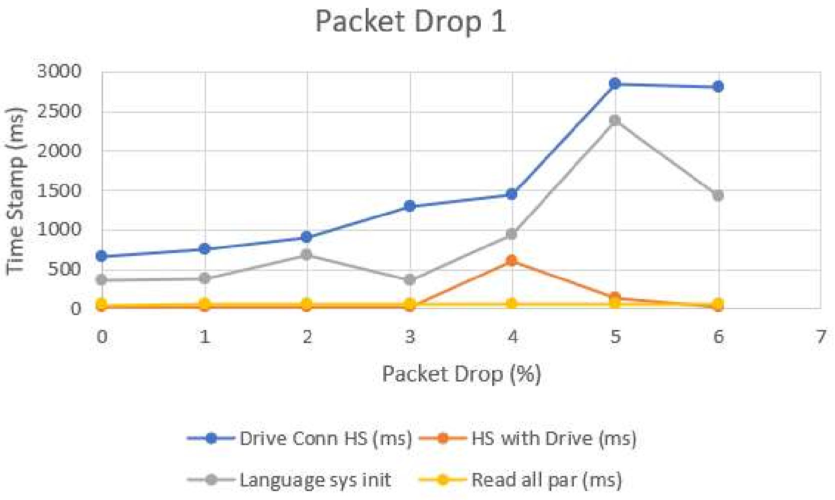


Figure 22. Graph of time stamps as a function of packet drop percentage.

As seen in figure 22 the effects of packet drop are hard to predict as there is no regularity. For example, in language system initialization the peak for measured time stamp was recorded during 5 % packet drop. Correspondingly the measured time stamp experiences a small drop in 6 % packet drop, which can be considered as the worst-case scenario in this testing.

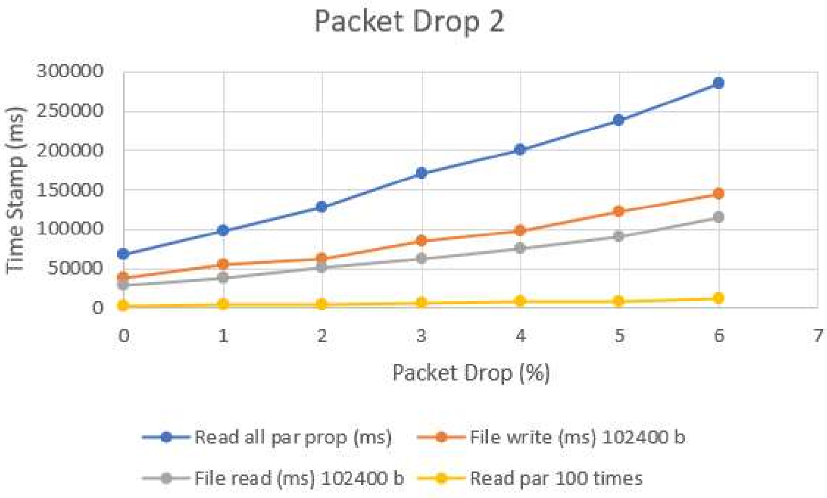
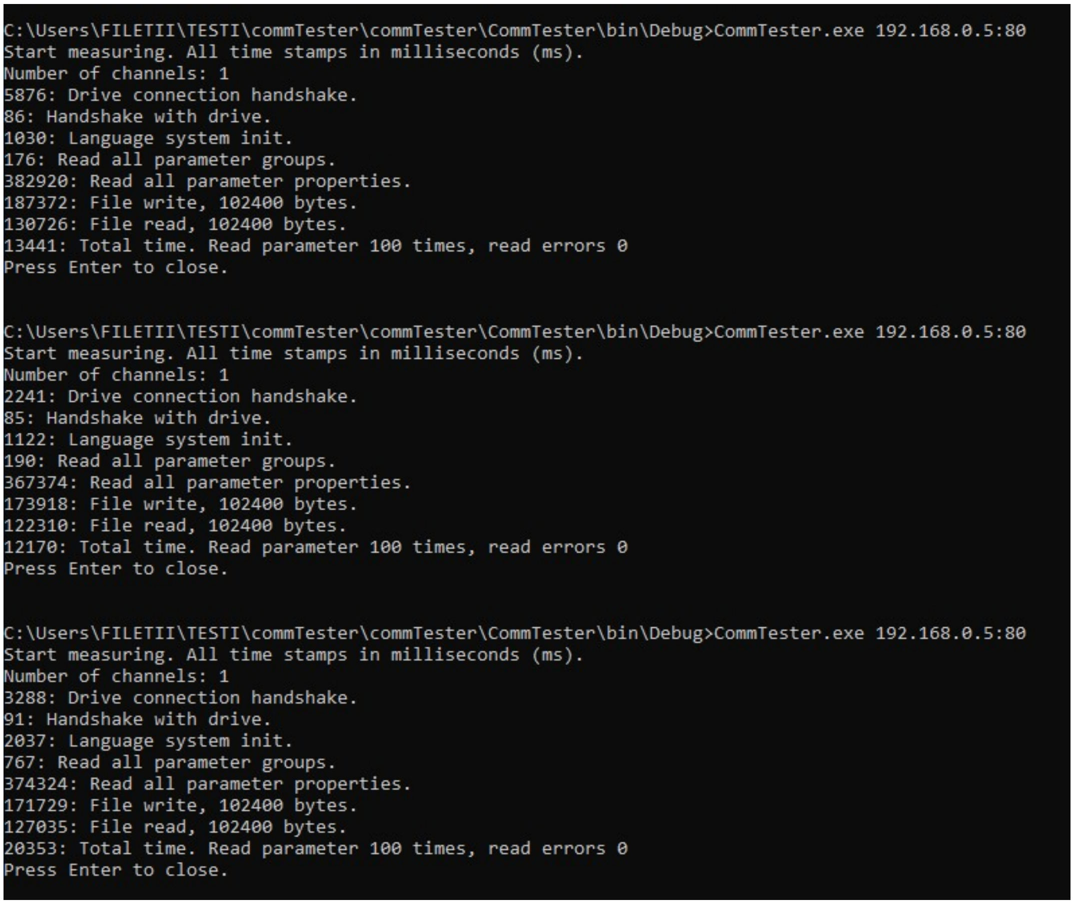


Figure 23. Graph of time stamps as a function of packet drop percentage.

8.3.3 Test Results for Packet Delay and Packet Drop

Reasonable packet delay and packet drop values proved to be 31 ms for the packet delay and 4 % for the packet drop. Even though packet drop of 2 % is considered as the limit in which the network performance can experience negative effects, 4 % packet drop still enables the user to monitor actual values in the parameter group 1 and change parameters with some delay. For the delay, Drive Composer's monitor window and signals can come up missing some data. In this case drive composer is still able to open, show parameters and actual values, and does not lose connection while operating. Figure 24 shows three consecutively executed communication tests with packet delay of 31 ms and packet drop of 4 %.



```
C:\Users\FILETII\TESTI\commTester\commTester\CommTester\bin\Debug>CommTester.exe 192.168.0.5:80
Start measuring. All time stamps in milliseconds (ms).
Number of channels: 1
5876: Drive connection handshake.
86: Handshake with drive.
1030: Language system init.
176: Read all parameter groups.
382920: Read all parameter properties.
187372: File write, 102400 bytes.
130726: File read, 102400 bytes.
13441: Total time. Read parameter 100 times, read errors 0
Press Enter to close.

C:\Users\FILETII\TESTI\commTester\commTester\CommTester\bin\Debug>CommTester.exe 192.168.0.5:80
Start measuring. All time stamps in milliseconds (ms).
Number of channels: 1
2241: Drive connection handshake.
85: Handshake with drive.
1122: Language system init.
190: Read all parameter groups.
367374: Read all parameter properties.
173918: File write, 102400 bytes.
122310: File read, 102400 bytes.
12170: Total time. Read parameter 100 times, read errors 0
Press Enter to close.

C:\Users\FILETII\TESTI\commTester\commTester\CommTester\bin\Debug>CommTester.exe 192.168.0.5:80
Start measuring. All time stamps in milliseconds (ms).
Number of channels: 1
3288: Drive connection handshake.
91: Handshake with drive.
2037: Language system init.
767: Read all parameter groups.
374324: Read all parameter properties.
171729: File write, 102400 bytes.
127035: File read, 102400 bytes.
20353: Total time. Read parameter 100 times, read errors 0
Press Enter to close.
```

Figure 24. Three consecutive communication tests.

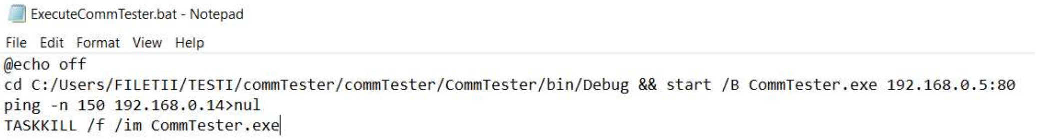
9 Test Automation

For the test automation, first step was to configure right IP-address for the drive in the parameter group mentioned earlier. After verifying the drive’s new IP-address, communication tester could be executed using the drive’s new IP-address.

The communication tester demands acknowledgement from the user after running the tests, that had to be bypassed. For this reason, a separate Batch-file was created in order to start the program using start /B -command and to add a timer for the communications tester. Start /B -command starts the program without creating a new window. To prevent all commands from displaying on the screen, @echo off -command is used. After the time limit is exceeded the batch file uses TASKKILL /f /im -command to force the communication tester to close.

The time delay for the program to close was built using ping. Ping uses less processor time than sleep- or TIMEOUT /T -commands. TIMEOUT /T -command would also interfere with the time stamps given from the communication tester by adding the remaining seconds into the recorded time stamps.

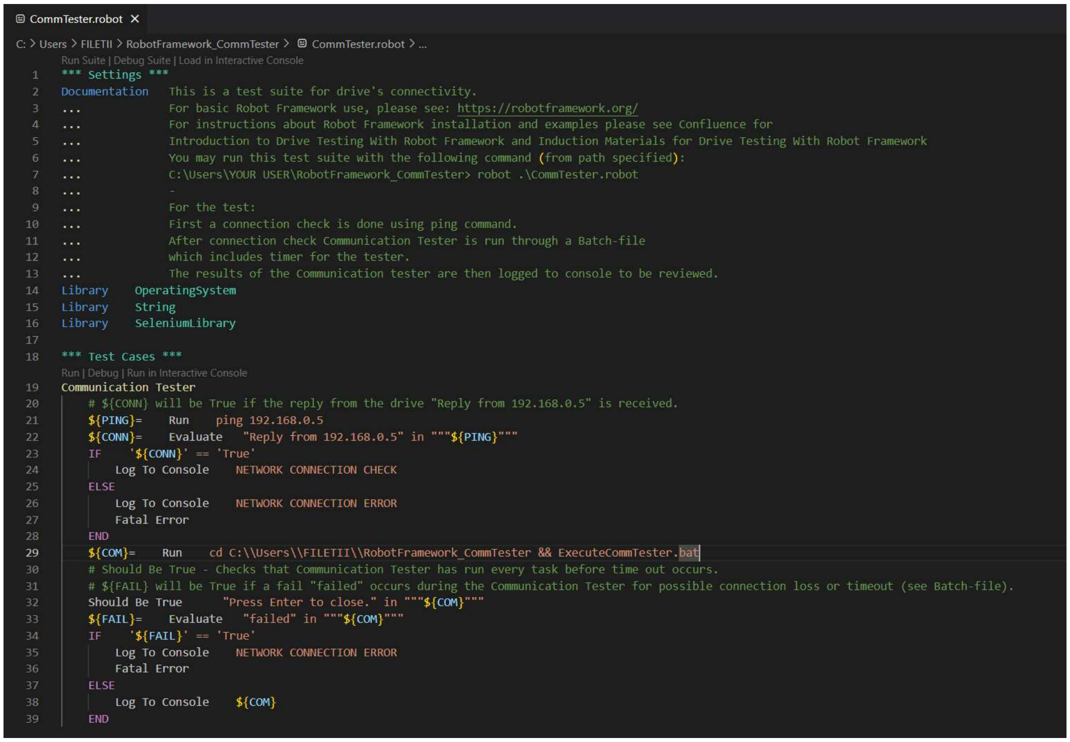
In case of connection error after the initial connection check in the beginning, the program gives notification in the end of the tests for connection error. The program does not have the ability to acknowledge connection errors while running commands, so it continues to run through the communication test according to the predetermined time. Figure 25 shows working Batch-file for communication tester.



```
@echo off
cd C:/Users/FILETII/TESTI/commTester/commTester/CommTester/bin/Debug && start /B CommTester.exe 192.168.0.5:80
ping -n 150 192.168.0.14>nul
TASKKILL /f /im CommTester.exe
```

Figure 25. Batch-file for communication tester.

While using the drive’s IP-address in the Batch-file during a possible connection loss after the first ping of the drive, the program would run endlessly. For this reason, the Batch-file used IP-address from the test laptop, since the timer needs a working connection in order to work properly. Figure 26 shows working program for communication tester in Visual Studio Code.

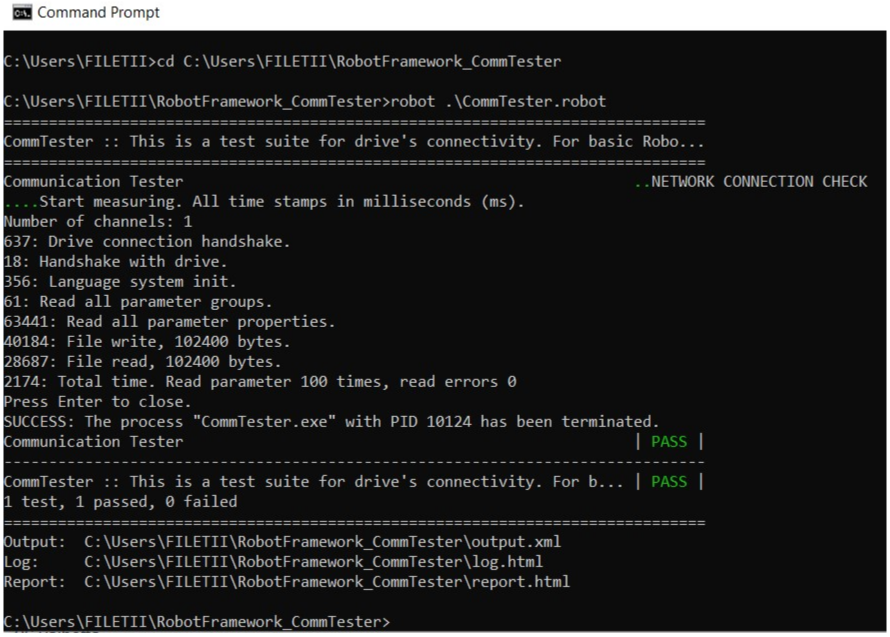

CommTester.robot ✕
C: > Users > FILETII > RobotFramework_CommTester > CommTester.robot > ...
Run Suite | Debug Suite | Load in Interactive Console
Run | Debug | Run in Interactive Console

```
*** Settings ***
Documentation   This is a test suite for drive's connectivity.
...             For basic Robot Framework use, please see: https://robotframework.org/
...             For instructions about Robot Framework installation and examples please see Confluence for
...             Introduction to Drive Testing With Robot Framework and Induction Materials for Drive Testing With Robot Framework
...             You may run this test suite with the following command (from path specified):
...             C:\Users\YOUR USER\RobotFramework_CommTester> robot .\CommTester.robot
...             -
...             For the test:
...             First a connection check is done using ping command.
...             After connection check Communication Tester is run through a Batch-file
...             which includes timer for the tester.
...             The results of the Communication tester are then logged to console to be reviewed.
Library    OperatingSystem
Library    String
Library    SeleniumLibrary

*** Test Cases ***
Communication Tester
    # ${CONN} will be True if the reply from the drive "Reply from 192.168.0.5" is received.
    ${PING}=    Run    ping 192.168.0.5
    ${CONN}=    Evaluate   "Reply from 192.168.0.5" in """${PING}"""
    IF    '${CONN}' == 'True'
        Log To Console    NETWORK CONNECTION CHECK
    ELSE
        Log To Console    NETWORK CONNECTION ERROR
        Fatal Error
    END
    ${COM}=    Run    cd C:\\Users\\FILETII\\RobotFramework_CommTester && ExecuteCommTester.bat
    # Should Be True - Checks that Communication Tester has run every task before time out occurs.
    # ${FAIL} will be True if a fail "failed" occurs during the Communication Tester for possible connection loss or timeout (see Batch-file).
    Should Be True      "Press Enter to close." in """${COM}"""
    ${FAIL}=    Evaluate   "failed" in """${COM}"""
    IF    '${FAIL}' == 'True'
        Log To Console    NETWORK CONNECTION ERROR
        Fatal Error
    ELSE
        Log To Console    ${COM}
    END
```

Figure 26. Communication tester program.

First the connection is checked using ping command. If connection fails, the program gives notification for connection error. If the connection is checked without any failure, the program continues to run the communication tester. Communication tester is then run using a Batch-file, which includes a timer for the tester and terminates it after the predetermined time. After the predetermined time, the program evaluates whether it was run without any failures and logs the measured time stamps into the console or in case of connection loss, gives notification for connection error. Figure 27 shows example of passed communication tester using Robot Framework.



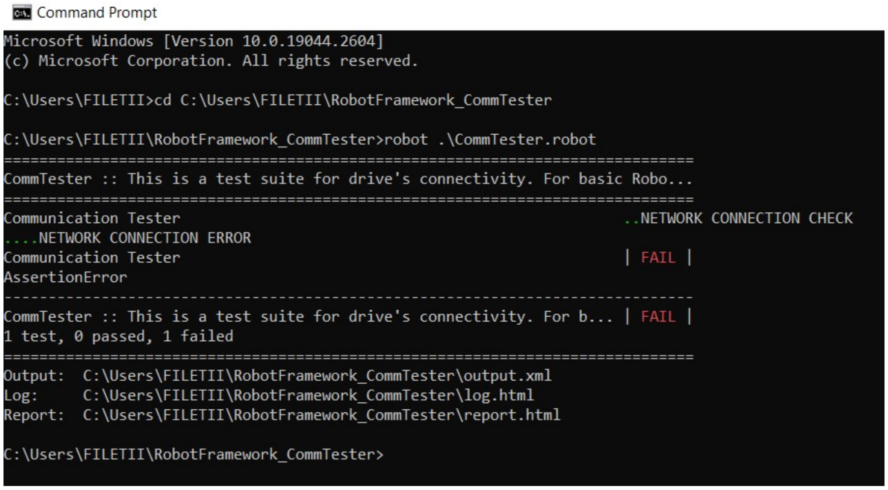
```
C:\Users\FILETII>cd C:\Users\FILETII\RobotFramework_CommTester

C:\Users\FILETII\RobotFramework_CommTester>robot .\CommTester.robot
==============================================================================
CommTester :: This is a test suite for drive's connectivity. For basic Robo...
==============================================================================
Communication Tester                                                  ..NETWORK CONNECTION CHECK
....Start measuring. All time stamps in milliseconds (ms).
Number of channels: 1
637: Drive connection handshake.
18: Handshake with drive.
356: Language system init.
61: Read all parameter groups.
63441: Read all parameter properties.
40184: File write, 102400 bytes.
28687: File read, 102400 bytes.
2174: Total time. Read parameter 100 times, read errors 0
Press Enter to close.
SUCCESS: The process "CommTester.exe" with PID 10124 has been terminated.
Communication Tester                                                  | PASS |
------------------------------------------------------------------------------
CommTester :: This is a test suite for drive's connectivity. For b... | PASS |
1 test, 1 passed, 0 failed
==============================================================================
Output:  C:\Users\FILETII\RobotFramework_CommTester\output.xml
Log:     C:\Users\FILETII\RobotFramework_CommTester\log.html
Report:  C:\Users\FILETII\RobotFramework_CommTester\report.html

C:\Users\FILETII\RobotFramework_CommTester>
```

Figure 27. Passed communication tester.

Successful communication tester checks the connection in the beginning using ping command and continues to run the communication tester. After the predetermined time the program is terminated, and the results are logged to the console. Figure 28 shows example of failed communication tester using Robot Framework.



```
Microsoft Windows [Version 10.0.19044.2604]
(c) Microsoft Corporation. All rights reserved.

C:\Users\FILETII>cd C:\Users\FILETII\RobotFramework_CommTester

C:\Users\FILETII\RobotFramework_CommTester>robot .\CommTester.robot
==============================================================================
CommTester :: This is a test suite for drive's connectivity. For basic Robo...
==============================================================================
Communication Tester                                                  ..NETWORK CONNECTION CHECK
....NETWORK CONNECTION ERROR
Communication Tester                                                  | FAIL |
AssertionError
------------------------------------------------------------------------------
CommTester :: This is a test suite for drive's connectivity. For b... | FAIL |
1 test, 0 passed, 1 failed
==============================================================================
Output:  C:\Users\FILETII\RobotFramework_CommTester\output.xml
Log:     C:\Users\FILETII\RobotFramework_CommTester\log.html
Report:  C:\Users\FILETII\RobotFramework_CommTester\report.html

C:\Users\FILETII\RobotFramework_CommTester>
```

Figure 28. Failed communication tester.

10 Conclusion

The goal for the thesis was to create a test setup and study variable speed drive's connectivity by manually running tests and simulating typical network issues and behaviour caused by the surrounding network. After the manual tests it was supposed to automate the test cases according to the manual tests.

At the beginning of the thesis there was a short representation of operating principle of a variable speed drive and examples of drive applications, the basic structure of a computer network, network topologies and connectivity regarding variable speed drives.

For the thesis the hardware responsible for the tests were either built or obtained and verified. Network simulator was built to simulate network behaviour and an installation box was built for electrical safety reasons to avoid any unwanted touches on the live wires and other electronic components.

Manual tests were run using the network simulator and ABB's own software to obtain knowledge about issues and behaviour regarding the connection between laptop and variable speed drive.

Communication tester used to test communication between the laptop and the drive was automated using Robot Framework. Unfortunately automating the whole process with possible parameter changes and re-using the data from communication tester would have taken a lot of time, considering the time already spent on the work in terms of getting the hardware and software to work for the manual tests. Automatization of the tests were confirmed to be possible but required some additional libraries and possibility to share data between hardware and software. The work included instructions for future development.

References

1 ABB Suomessa. (Internet). ABB Oy. Available at: <https://new.abb.com/fi/abb-lyhyesti/suomessa> (Accessed 12.28.2022).

2 ABB drives - Technical guide book. 6.6.2011. (Internet). ABB. Available at: <https://library.e.abb.com/public/df559f6df460420c8a0b14d07d109263/TechnicalGuideBook_EN_3AFE64514482_RevI.pdf> (Accessed 2.22.2023).

3 Taajuusmuuttajat. (Internet). SähköNet. Available at: <https://blogit.gradia.fi/sahkonet/sahko-ja-automaatioasennukset/oppimistehtavat/teollisuuden-sahkoasennukset/moottori-kaytot/taajuusmuuttajat/> (Accessed 1.26.2023).

4 Niiranen, Jouko. 1999. Sähkömoottorikäytön digitaalinen ohjaus. Helsinki: Otatieto.

5 Shinde, S.S. 2009. Computer network. New Delhi: New Age International (P) Ltd., Publishers.

6 Ethernet. 8.2021. (Internet). Available at: <https://www.techtarget.com/searchnetworking/definition/Ethernet> (Accessed 12.30.2022).

7 What is WLAN? (Internet). Sunny Valley Networks. Available at: <https://www.sunnyvalley.io/docs/network-basics/what-is-wlan> (Accessed 2.1.2023).

8 Industrial network architectures: Structure, Types and Examples. (Internet). Available at: <https://study.com/academy/lesson/industrial-network-architectures-structure-types-examples.html> (Accessed 1.4.2023).

9 What are the Different Types of Network Topologies? 11.29.2022. (Internet). Available at: <https://www.naukri.com/learning/articles/different-types-of-network-topology/> (Accessed 1.16.2023).

10 Bonald, Thomas; Feuillet, Mathieu. 2011. Network Performance Analysis. Great Britain and the United States: ISTE Ltd. and John Wiley & Sons, Inc.

11 What is network packet loss? (Internet). IR. Available at: <https://www.ir.com/guides/what-is-network-packet-loss> (Accessed 3.21.2023).

12 Variable Frequency Drive Digital Inputs. (Internet). VFDs. Available at: <http://www.vfds.org/variable-frequency-drive-digital-inputs-631312.html> (Accessed 4.5.2023).

13 ACS380 machinery control program firmware manual. (Internet). ABB. Available At: <https://library.e.abb.com/public/a6160c1b1d334aa988152ca497e8cc24/EN_ACS380_FW_F_A5.pdf> (Accessed 1.31.2023).

14 Raspberry Pi 4. (Internet). Raspberry Pi. Available at: <https://www.raspberrypi.com/products/raspberry-pi-4-model-b/> (Accessed 1.10.2023).

15 Raspberry Pi Touch Display. (Internet). Raspberry Pi. Available at: <https://www.raspberrypi.com/products/raspberry-pi-touch-display/> (Accessed 1.12.2023).

16 Raspberry Pi Documentation. (Internet). Raspberry Pi. Available at: <https://www.raspberrypi.com/documentation/computers/raspberry-pi.html> (Accessed 1.25.2023).

17 Robot Framework. (Internet). Robot Framework. Available at: <https://robotframework.org/> (Accessed 2.16.2023).