Department of Electrical Engineering

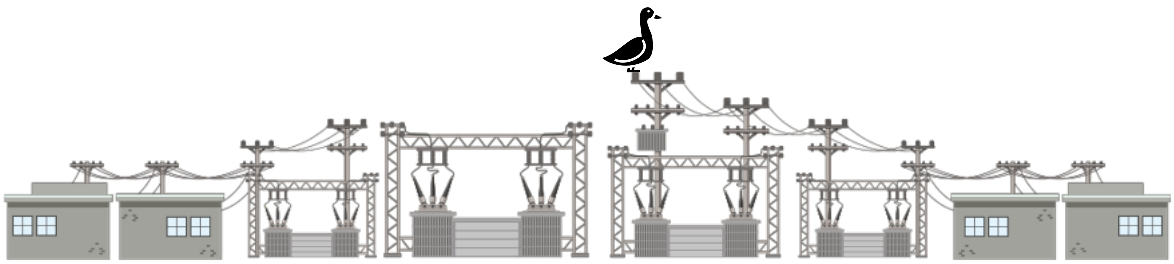
Msc Thesis
to obtain the degrees of:
MSc. Electrical Engineering at the Delft University of Technology,
MSc. Wind Energy at the Norwegian University of Science and Technology.

Automated Testing of Interlockings in Digital Substations

Thesis Committee:

Prof. Dr. Ir. Marjan Popov	Supervisor and chair of the committee, TU Delft
Prof. Hans Kristian Høidalen	Main supervisor and external committee member, NTNU
Dr. Jianning Dong	Core member of the committee, TU Delft

To be defended publicly on the 5th of July, 2024 at 13:00.



By:
Jens Kruse-Hansen

Table of Contents

Nomenclature

ASD	Application Specification Description
BAP	Basic Application Description
BCU	Bay Control Unit
CILO	Control Interlocking
CIT	Conventional Instrument Transformer
CT	Current Transformer
DA	Data Attribute
DANH	Doubly Attached Node with HSR
DANP	Doubly Attached Node with PRP
DO	Data Object
DUT	Device Under Test
EMU	Embedded Merging Unit
FAT	Factory Acceptance Test
FOC	Fibre Optic Cable
GOOSE	Generic Object Oriented Substation Event
HMI	Human Machine Interface
HSR	High-availability Seamless Redundancy
I/O	Input/Output
ICD	IED Configuration Description
IEC	International Electrotechnical Committee
IED	Intelligent Electronic Devices
IID	Instantiated IED Description
IT	Instrument Transformer
LAN	Local Area Network
LD	Logical Device
LN	Logical Node
LPIT	Low Power Instrument Transformer
MMS	Manufacturing Message Specification
MRP	Media Redundancy Protocol
MU	Merging Unit
NCIT	Non-Conventional Instrument Transformer
O&M	Operation and maintenance
OSS	Offshore Substation
PAC	Protection Automation & Control
PACS	Protection, Automation and Control Systems
PRP	Parallel Redundancy Protocol
PTP	Precision Time Protocol
Q01	Outgoing Feeder Bay
Q05	Bus Coupler Bay
RedBox	Redundancy Box
RSTP	Rapid Spanning Tree Protocol
SA	Substation Automation
SAMU	Stand-Alone Merging Unit
SAN	Singly Attached Node
SAT	Site Acceptance Test
SCADA	Supervisory Control and Data Acquisition
SCD	Substation Configuration Description
SCL	System Configuration Language
SCU	Substation Control Unit
SV	Sampled Values
TAL	Time Allowed To Live
TC	Technical Committee
VBA	Visual Basics for Applications
WG	Working Group

List of Figures

List of Tables

Acknowledgements

I want to express my gratitude towards those who have helped me during this master's thesis. I am deeply grateful to Professor Hans Kristian Høidalen, who not only served as my main supervisor but also enriched my work with his insightful feedback from our many meetings.

I also want to thank Professor Marjan Popov from TU Delft, who agreed to be my second supervisor at a pivotal point in his career due to his professorial inauguration. Both professors were immensely helpful in providing feedback on the thesis and the PAC World paper based on this work.

I gratefully acknowledge the TU Delft thesis committee as a whole, consisting of Prof. Marjan Popov, Dr. Jianning Dong, and Prof. Hans Kristian Høidalen, for taking the time to attend my thesis defence.

I would like to extend my thanks to Rizwan Rafique Syed, PhD candidate at NTNU, for his exceptional collaboration on the paper and in the laboratory and Burak Tahincioglu, whom I met during an IEC 61850 course in Oslo. This thesis would have been entirely different if it had not been for him sharing his previous paper with Rizwan and me. Burak, representing OMICRON, provided precious insights from within the industry and gave NTNU and me access to valuable tools for this thesis. His insights and contributions have indubitably enhanced my thesis work.

Finally, I want to express my gratitude towards my mother, my brothers, and my girlfriend for their unrelenting support during the final stages of my studies.

Jens Kruse-Hansen,
Trondheim, June 2024

Abstract

This thesis proposes an automated method for generating, executing, and assessing an interlocking test in a digital substation using a Python script designed for that specific purpose.

The goal is to expedite the process of performing the factory acceptance test (FAT) and the site acceptance test (SAT) of a substation automation system (SAS). This work requires a good understanding of the IEC 61850 standard, which is the international standard applicable to protection, automation and control systems (PACS). An account of the relevant parts from this series is therefore given, in addition to an overview of the benefits of digital substations in general.

The workflow for automatically generating a test case is based on previous work that made it possible to execute and assess an interlocking test automatically but not to generate a test case automatically for this purpose. Therefore, that is the main intention of the thesis.

It is done by having knowledge of the underlying interlocking logic of the system subject to test. From this logic, a test sequence can be created, where the position of the various switchgear is changed sequentially. This and the signal addresses for these devices are needed to generate a test case.

For additional robustness, the script can cross-check the signal addresses provided with the signal addresses in the substation configuration description (SCD) file and validate the final test case generated using a suitable schema. It is furthermore capable of generating a test case irrespective of the number of test steps, switching devices, and bays present in the substation.

A test file generated using this script is further validated by executing the test it describes in the SAS laboratory at the Norwegian University of Science and Technology (NTNU). This test was carried out remotely to showcase the possibilities of IEC 61850, which can be valuable for distant or offshore substations. Another benefit of this workflow is that it allows for the simulation of all devices of the test except for the device under test (DUT) . This is particularly useful during commissioning if all devices have not yet been delivered or installed. In this case, the missing equipment can be compensated for by simulating the signals expected from these devices.

The final assessment of a test case relies upon the presence of the IEC 61850 LN (Logical Node) CILO (Control Interlocking). The output of this LN controls the interlock status of the DUT. If the DUT is allowed to operate, it sends a release signal or, alternatively, a blocking signal. In addition to this, information is gained based on the position of the switchgear under test to check that the CILO signal is consistent with the actual switchgear control command. This control command is known as the AddCause in IEC 61850 and will provide additional information on whether or not the DUT is interlocked.

Finally, the thesis will describe ongoing work in the IEC 61850 that could lead to a more streamlined approach and touch on the utilities' attitude towards SAS.

Chapter 1

Introduction

1.1. Motivation for the Thesis

According to research carried out by McKinsey, the total capacity of installed offshore wind is expected to increase from 40 GW in 2020 to 630 GW in 2050, an increase of 590 GW. The capacity is projected to be able to increase further to 1000 GW by 2050 if following a target of limiting global warming to only 1.5 °C [1].

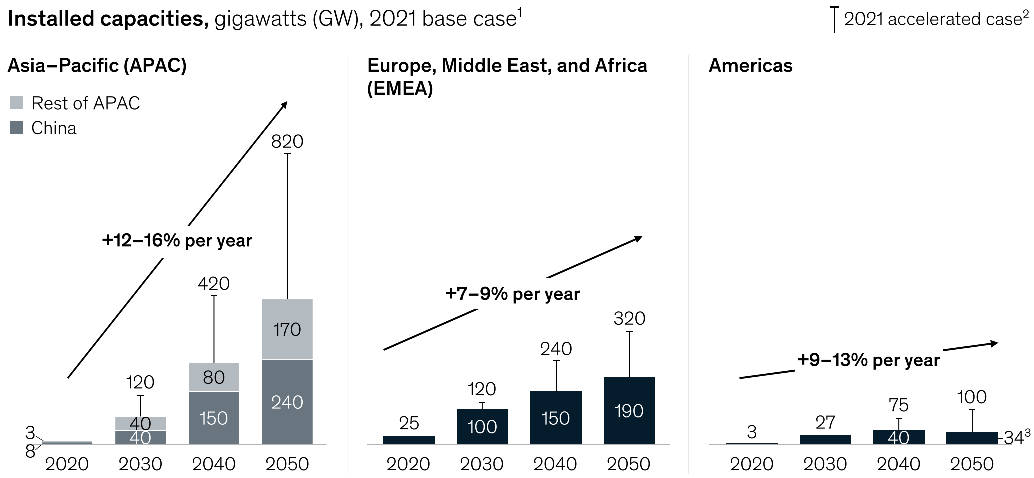


Figure 1.1.: Projected increase in installed offshore wind capacity from 2020 to 2050 in APAC, EMEA and the Americas. Note: APAC includes OECD Asia–Pacific and non-OECD Asia; EMEA includes OECD Europe, Eurasia, Middle East, and Africa; Americas includes OECD Americas and Latin America. [1]McKinsey's view on the current path of energy transition without major shifts in production and consumption compared to today. [2]McKinsey's view on an accelerated energy transition, including several conceivable shifts in production and consumption compared to today. [3]Capacity decrease due to forecasted decommission. This figure is from [1].

Given the projected growth in offshore wind, it is only natural that a growth in the number of offshore substations (OSS) will follow. Therefore, it is desirable to uncover ways to make offshore commissioning, operation, maintenance, and monitoring easier and safer. These are all factors where digital substations could have a significant advantage.

The amount of manual intervention in OSS should be kept to a minimum, as the price of carrying out any activity on an OSS compared to an onshore substation is approximately tenfold [2]. Furthermore, there are several security hazards that should be accounted for. A plethora of these hazards have to do with personnel being physically present at the OSS; these include transport to and from the OSS, emergency evacuation, exposed locations, restricted working area and electrical hazards when testing or operating [2].

This thesis will attempt to contribute to the development of digital substations. Specifically, it is desired to make a developmental step within substation automation (SA) of digital substations. A larger degree of automation within the offshore digital substations would decrease the amount of time that personnel need to spend on an OSS exposed to these various hazards, bringing down the immense costs associated with them.

1.2. Thesis Description

As part of the joint research project ProDig [3] at NTNU, it is the aim of this master's thesis to contribute to the core objective, which is *to develop and disseminate the knowledge required to securely utilize digital substations in the future power system* [3]. Furthermore, contributions will be made within work packages 5 and 6, which are *Test procedures for fast digital substation upgrade* and *Laboratories for digital substation testing, workforce development* respectively.

This master's thesis seeks to make use of the laboratory that NTNU and ProDig provide, where testing of digital substations is made possible. Concretely, it is desired to investigate whether automating the testing of interlockings for digital substations is possible. The thesis will reflect on the advantages and disadvantages of digitising and automating the tests. Furthermore, the IEC 61850 standard series, which is the international standard applicable to protection, automation and control systems (PACS) will be explored, and benefits and possible shortcomings will be addressed [4].

A substantial part of the thesis will, therefore, be dedicated to reviewing the literature on IEC 61850 and accounting for what it is and what it can and cannot provide. An overview of what can be expected from the IEC 61850 in the future will also be given.

To meet the objectives set forth, the following tasks should be performed:

- Address advantages of digital substations over conventional substations
- Investigate ongoing work related to the expansion of the SAS and logic testing-related functionalities in the IEC 61850 standard.
- Provide an account of relevant parts of the IEC 61850 that relate to testing of interlockings.
- Devise a way to automate the testing of interlockings in digital substations using IEC 61850.
- Demonstrate an automated interlocking test.
- Contribute to the NTNU ProDig laboratory.

Chapter 2

Theory

2.1. Substations

Substations play a crucial role in the functioning of the electrical power grid, serving as critical points of interconnection between different voltage levels and facilitating the transmission and distribution of electricity, whether conventional or digital. Importantly, while conventional substations are still predominant, there has been a significant transition towards digital substations in recent years [5–8]. In 2019, over 5000 digital substations were in operation in China alone [9]. This transition is driven by the adoption of the IEC 61850 standard, which provides a framework for implementing digital communication and control systems in substations. The IEC 61850 standard offers key concepts and protocols that enable interoperability and seamless communication between devices within a substation. The conventional, hard-wired communication will be superficially explained. In contrast, digital, IEC 61850-based communication will be covered in more extensive detail as it is a pivotal point for the thesis.

Before diving deep into the complexity of the IEC 61850, an overview of digital substations will be given.

2.1.1. Definition of a Digital Substation

The definition, according to IEEE fellow and distinguished member of CIGRE Alexander Apostolov, of a digital substation is:

> *An IEC 61850-based digital substation is a substation in which all interfaces between the primary equipment in the substation and the devices performing protection, automation, control, monitoring and recording are based on communications over the substation local area network using the models and services defined in the standard* [10].

He further states that they require merging units, either Stand-Alone Merging Units (SAMU) or Embedded Merging Units (EMU). The difference between these is that the former connects to the secondary of conventional current transformers (CTs) and voltage transformers (VTs), whereas the latter is incorporated into low power instrument transformers (LPITs).

Other sources focus on the presence of the process bus in addition to the more widely used station bus [11].

2.2. Substation Layout

The terms process, station, and bay level are fundamental to understanding the layout of a substation. Together, these levels form the complete system, which provides control and monitoring for the entire facility and ensures communication with connected systems, such as the utility company's control centre.

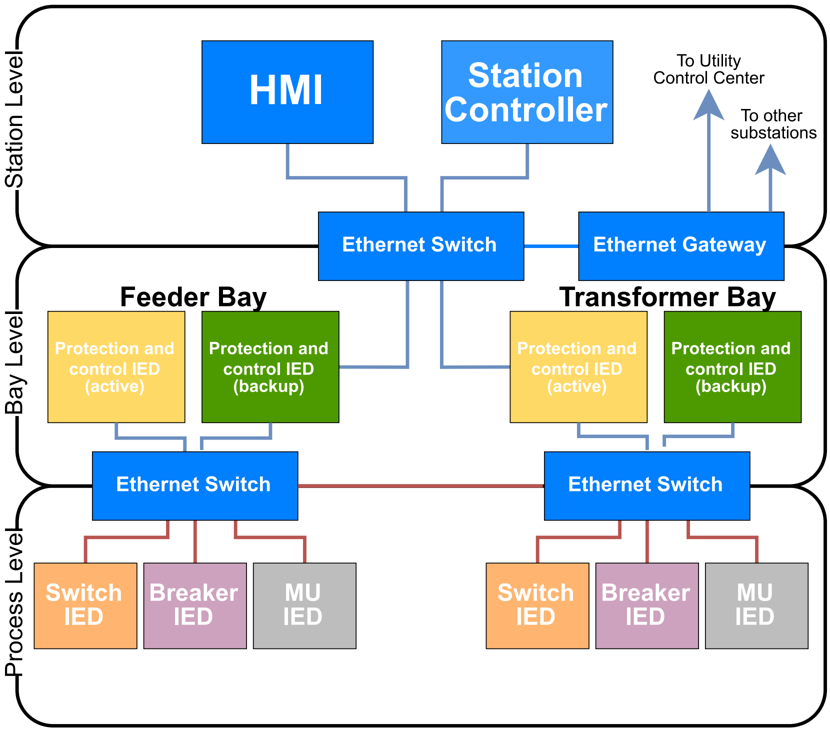


Figure 2.1.: The three levels of digital substations and what these levels are comprised of are shown. The figure is adapted from [12].

IEC 61850-2 [13] defines three "levels" of devices in an automation system:

1. *Station Level: station HMI (Human Machine Interface, gateway interface to SCADA (Supervisory Control and Data Acquisition) and station loggers.*
2. *Bay Level: protection relays, meters and controllers.*
3. *Process level: switchgear, transformers, sensors and input/output (I/O) interfaces.*

These three levels are further described below, and their contents are represented visually in Figure 2.1.

2.2.1. Station Level

On the top level of Figure 2.1 lies the station level. This level entails all the control and monitoring capabilities of the substation. All of the substation infrastructure can be overseen, and large parts of it can potentially be interacted with through the HMI. This is how the operators can manage alarms, inspect them, and make changes that affect the substation. It is also from here that the communication interfaces to the grid control centre [4]. Station level functions apply to the whole power system [13].

2.2.2. Bay Level

In the middle of Figure 2.1, the bay level is situated. Here, individual bays, busbars or feeders are controlled by means of the relays, meters and controllers [4]. Communication from here is established to the station level through Ethernet switches and to other substations and the utility control centre, where a system like SCADA could be in place.

Bay level functions are functions that mainly use the data of a single bay and primarily interact with that specific bay [13].

2.2.3. Process Level

Here, the primary equipment is found. This includes IEDs (intelligent electronic devices), transformers, instrument transformers (ITs), sensors and switchgear. [4]. Communication from here is established to the bay level through Ethernet switches.

Process level functions are all functions interfacing the process, i.e. binary and analogue input/output functions like data acquisition (including sampling) and issuing of commands, as opposed to process related station level functions that use the data of more than one bay and act on more equipment than that of just one bay [13].

2.3. Network Redundancy Architectures in Digital Substations

Not unlike other electrical infrastructure, digital substation design also employs redundancy architectures to ensure reliable, continuous operation. This ensures dependable communication and thus mitigates the adverse impacts of network failures. As this thesis deals with tripping functions, the communication signals here are of utmost importance, which is why zero network recovery time is the aim. Any higher network recovery time comes with the risk of having equipment operate too late or not at all, which may damage equipment, result in financial losses, downtime due to repair and maintenance and, at worst, endanger human life. Less critical applications might allow for a higher network recovery time, and to accommodate these various requirements, different network configurations may be used, depending on the application. Furthermore, one must select an appropriate protocol to accompany the network configuration.

The most common protocols will be briefly addressed for two different network topologies, namely single-ring and two-ring networks, focusing on protocols suitable for applications where critical tripping commands are found. This is not an exhaustive list. There are more protocols and configurations available, additionally it is possible to combine more of the same or even different protocols to tailor to one's needs [14].

2.3.1. Single-Ring Network

In this network configuration, LAN (local area network) switches are connected to each other in series. The LAN switch connected at the end will then be connected to the first one, forming a single ring. The various devices can then be connected to these LAN switches. The device can even be connected redundantly by using two ports, such that a failure on one port will not matter as the other port will take over the transmission [4].

2.3.1.1. Rapid Spanning Tree Protocol (RSTP)

RSTP can be used on most network configurations but works well on single-ring networks [4]. Although not advisable for applications with SV or tripping GOOSE signals, this protocol is recommended for SCADA, protection, automation and control systems applications. The protocol still deserves a mention, as it is very prevalent in both applications and literature. It is defined in the IEC 62439-2 standard. As the ring network gets expanded by adding more LAN switches, the recovery time using this protocol increases [4]. It works by means of an algorithm implemented on the switch, which can dictate the optimal path for a message to be routed. Consequently, if a path fails, the algorithm will reroute all network traffic via the remaining unimpaired links. This does, however, take time [15]. For a 50 Hz system, considering a SV stream of 80 S/period, there would be $\frac{80\,\mathrm{S/period}}{1/50\,\mathrm{Hz}} = 4000\,\mathrm{S/s}$, which is equivalent to a frame every 250 µs. If considering a recovery time of 60 ms, that would equate to dropping 240 packets. At 80 S/period, three cycles would be lost, which is simply too much for a system meant to trip reliably [4, 15].

2.3.1.2. Media Redundancy Protocol (MRP)

MRP, defined in IEC 62439-2 [16], is similar to RSTP, particularly suitable for single-ring networks. This differs from the aforementioned in that it is able to exhibit sufficiently fast recovery times to also be used for applications that require trip signals transmitted on GOOSE. MRP is able to do this even with an extensive network featuring numerous LAN switches [4]. It is imperative, however, that the worst-case recovery time is calculated if the protocol is to be used for critical operations. The downtimes can be calculated in accordance with IEC 62439-1 [17].

2.3.1.3. High-availability Seamless Redundancy (HSR)

HSR is also used on a single-ring topology and offers the advantage of a null recovery time. Moreover, it does not employ LAN switches and has a simplified installation, leading to a lower final cost. HSR is recommended for the same applications as RSTP, with the added inclusion of trips transmitted by GOOSE messages [4]. It is based on PRP (Parallel Redundancy Protocol), which will be introduced below for the two-ring networks. Both of these protocols are found in the IEC 62439-3 [14]. Due to the ring topology, there is no need for switches, rather the redundancy is embedded in the IEDs themselves, by sending duplicated frames in both the clockwise and the counterclockwise direction. Whenever another IED receives two frames, the duplicate one is simply discarded. In the event of a failure, the traffic will not be interrupted, as the transmission of communication can simply take place in the opposite direction around the ring.

A DANH is a Doubly Attached Node with HSR. These nodes have built-in redundancy in the form of the two ports they provide. A singly attached (SAN) node is, unlike a DANH, a device with only one port and, thus, no built-in redundancy. Therefore, these devices must be connected via a RedBox (redundancy box) in the HSR configuration [15].

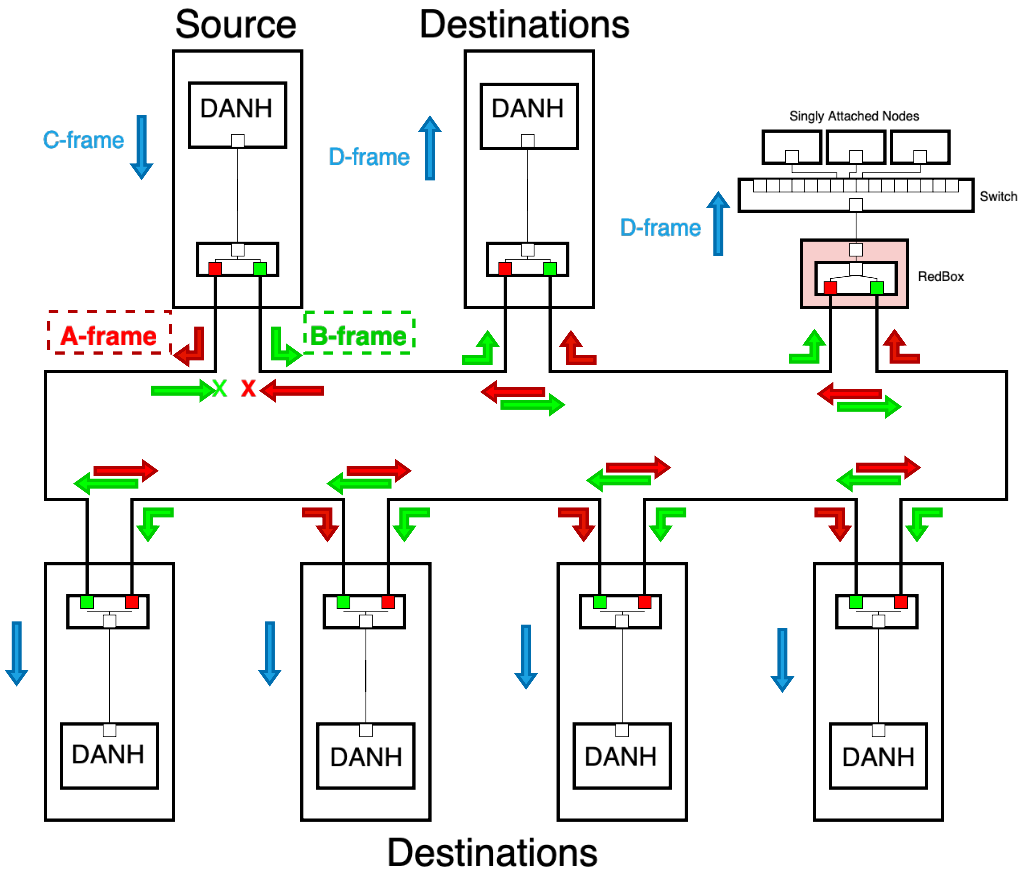


Figure 2.2.: HSR network. Adapted from [14]. Note how communication goes both clockwise and counterclockwise. Notice also the need for the RedBox when connecting SANs here. The C and D frames are non-HSR frames.

2.3.2. Two-Ring Network

This network configuration follows the exact same approach as the single-ring network, but another ring is added in parallel. These rings are completely independent. This results in a system where a failure on a LAN switch or on the connection between two switches only affects one ring, while the other redundant ring will remain in operation and ensure that the overall system is not affected.

2.3.2.1. Parallel Redundancy Protocol (PRP)

PRP is also described in IEC 62439-3 [14]. This protocol also ensures a null recovery time. It does this by means of a full duplication of the LAN network. This entails having a fully redundant LAN network ring. Similarly to the HSR, a source node creates duplicate frames and transmits them through both parallel paths. Once again, the destination nodes make sure to discard duplicate messages received. Again, this requires doubly attached nodes, this time with PRP instead of HSR. These nodes are, therefore, abbreviated DANP instead of DANH here. All of this comes together to provide an expected time delay of zero in case of a single failure since the two parallel paths are completely isolated. DANPs must be connected through switches (one for each ring), and SANs still require the extra interface, the RedBox, if they are to be redundant. The PRP can be used for anything that the HSR can, but it is implemented on two-ring networks. [4, 15]. An obvious disadvantage of the PRP protocol is that it is costly since it requires a full duplication of the original LAN network, including the switches.

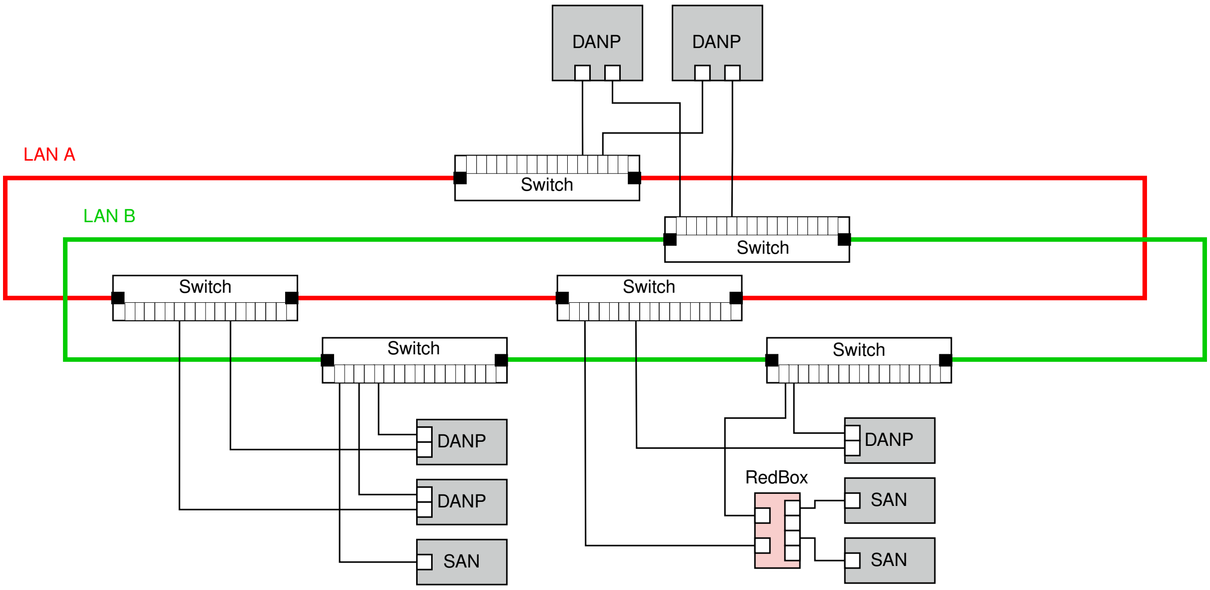


Figure 2.3.: PRP network. Connections to DANPs and how to interface a SAN with and without a RedBox are shown. Adapted from [14]. The LAN networks could also follow a linear bus topology instead of the ring network.

2.4. Advantages of Digital Substations (For Offshore Applications)

Digital substations offer copious advantages over conventional substations. There are significant benefits in weight, footprint, length of cabling for instrument transformers and sensors, safety, manufacturing hours and maintenance. Some of the critical differences are quantified in Table 2.1, and these, along with other notable benefits, will be elaborated upon in the following sections. Furthermore, it will be emphasised why these advantages are particularly beneficial for offshore substations.

	Conventional Switchgear	**Digital Switchgear**	**Reduction**
Length	5.94 m	5.28 m	11%
Estimated Weight	13 200 kg	11 400 kg	14%
IT/sensor wiring length	762 m	88.1 m	88%
IT/sensor terminations	910	76	92%
Manufacturing	135 h	24 h	82%
Number of shipping splits	3	2	33%

Table 2.1.: An example of the savings that can be achieved with digital switchgear adapted from [18].

2.4.1. Low-Power Instrument Transformers

Digital substations are safer than conventional substations, largely because of the employment of Low-Power Instrument Transformers (often referred to as Non-Conventional Instrument Transformers (NCITs), however in accordance with CIGRE, LPIT will be used henceforth [4]). These LPITs are, as the name suggests, characterised by low power output when compared to a CIT (Conventional Instrument Transformer). Furthermore, the lack of the ferromagnetic core that most CITs make use of eliminates the significant safety hazard of conventional CTs that can be caused by an unpremeditated opening of the secondary CT circuit, which will lead to an extreme voltage increase, which can lead to arcing and in worst case an explosion [4, 18–23]. Similarly, the LPIT variant of a VT does not include the aforementioned ferromagnetic core. This means a short circuit on the secondary side will have severely limited consequences compared to a CIT. Furthermore, the omission of the ferromagnetic core makes the sensor noninductive, which eliminates the risk of ferroresonance, which can result in overvoltages. To fully appreciate the advantages of LPITs, consider also that a CIT can provide somewhere in the vicinity of 25,000 to 100,000 times the power level of an LPIT on the secondary side. These advantages are summarised in Table 2.2 [18].

LPITs	**CITs**
These do not require additional measures for open-circuit safety for current measurement devices.	CTs require shorting blocks to avoid potentially fatal voltages if the transformer is energised with the secondary terminals open.
These do not require additional measures for short circuit safety for voltage-measurement devices.	Potential transformers present a hazardous condition if the secondary terminals are short circuited with the primary windings energised.
Voltage sensors are noninductive devices and are not subject to failure from ferroresonance events.	The inductance of potential transformers may resonate with the capacitance of the line and can result in an overvoltage failure due to ferroresonance.
Sensors are low-energy devices and generate negligible internal heating, resulting in higher reliability.	Traditional instrument transformers may experience thermal aging of the internal insulation system due to excessive load, which can lead to the device's premature failure.

Table 2.2.: Comparison between LPITs and CITs [18].

Another advantage that is most prevalent in offshore applications is that the fibre optic cables (FOC) that replace a vast amount of the copper cables in digital substations are more resistant to the corrosion that can occur in harsh environments with high humidity and salinity [19, 24]. Perhaps more importantly, the FOC have the advantage of being only a fraction of the weight of copper cables. Consider, for instance, for the instrument transformers, a reasonable $2.5\,\mathrm{mm}^2$ (14 AWG) copper cable [25]. These weigh 35 kg/km [26], whereas fibre optic cables weigh between 7.5 kg/km to 12 kg/km [27]. This puts the copper cables at a weight between approximately 2.9 to 4.6 times greater than the FOC. The weight difference between copper cables and FOC is not the principal reason for major weight reduction compared to conventional substations; rather, this is due to the overall reduction in the amount of copper cables used. Using FOC for communication, copper cabling can be reduced by up to 80% [19]. Moreover, research suggests that the speed of IEDs can be increased by up to 30% by the use of wires (commonly referred to as Ethernet cables) and FOC [4, 28]. Further specifications for Ethernet, including Ethernet over fibre, can be found in IEC 8802-3 [29].

Not only are the cables lighter in weight, but so are the LPITs themselves [20–22]. The weight of an OSS is directly related to the price of the structure [2]; consequently, possible ways to bring down the weight of the substation are of interest. The physical footprint naturally plays a role, too, as the size of an OSS platform is tied directly to costs [2]. This further promotes the use of LPITs as they are also smaller in footprint [20–23]. According to a white paper by

Hitachi Energy [19], the relay house space can be reduced by up to 60%. All these benefits can make LPITs less costly in comparison to their counterparts [23]

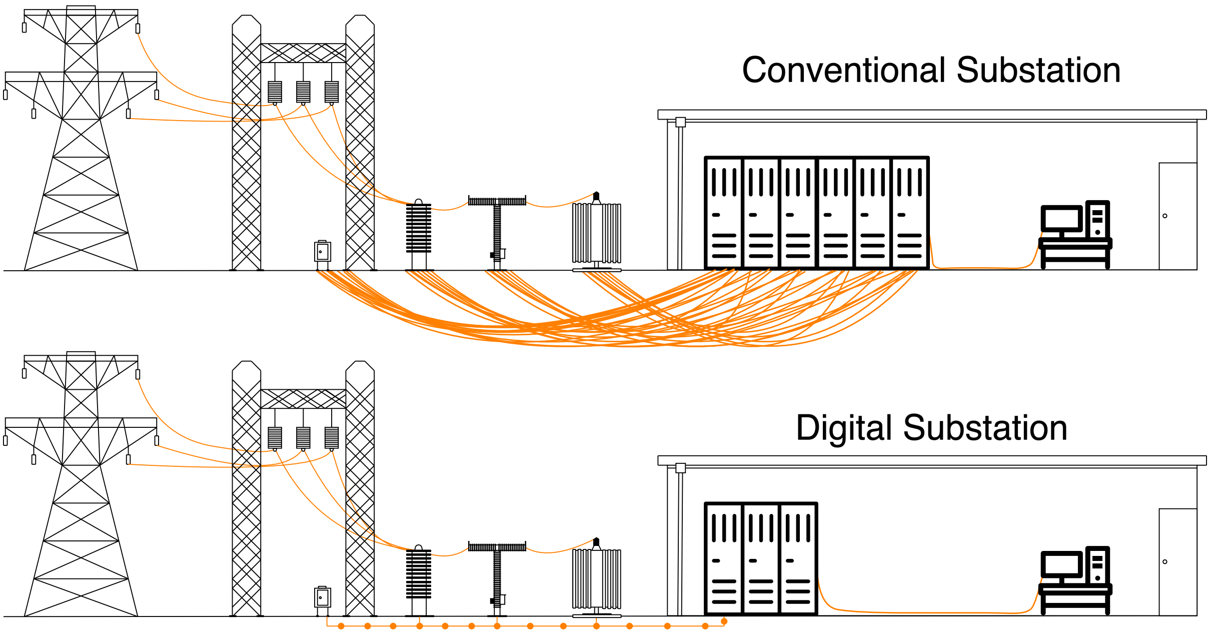


Figure 2.4.: Illustration showing that digital substations replace many point-to-point copper cables with a single fibre-optic process bus. The figure is adapted from [30].

2.4.2. Interoperability

Full interoperability between IEDs is possible if ensuring an IEC 61850-compliant integration of the digital substations. This comes with several advantages. One of these advantages is that it avoids common mode failures that might be specific to a particular vendor. Furthermore, one can reap the benefit of being able to select an IED from whichever vendor is at the forefront of the particular implementation case required. Another important case for supporting interoperability is that if a device becomes obsolete, it will be easier to replace it [4, 31]. It is, however, important to note that interoperability and interchangeability are not synonymous. Interchangeability signifies that an IED is replaceable with one from another vendor during operation and maintenance without any requirement for IED-specific re-engineering and testing. This cannot be expected, and one should anticipate having to configure any IED that the prior one was replaced with using a new ICD (IED Configuration Description) file, as IED interface tools are proprietary [4, 31].

Research shows that the interoperability of IEDs has not always been flawless. However, if the end-user clearly specifies requirements to the vendors regarding functionality before commissioning, interoperability can be achieved [6]. Newer studies suggest that the interoperability has improved in terms of communication recently, but the configuration tools are trailing slightly behind these developments [32]. Steinhauser [32] does, however, point out that even though the development is not as fast as one would like, the proprietary tools often have sufficient generic functions to, in the end, craft an adequate SCD file. Lastly, he also states that open-source solutions have emerged, which further plays into the multi-vendor approaches.

A CIGRE paper by LANDSNET and Omicron [5] found that the standards leave some ambiguity, which can result in inconsistencies in implementing functionalities in IEDS between vendors. Despite this, the paper also states that for full redundancy of the main control and protection

IEDs, IEDs from two different vendors must be incorporated. This avoids a single type of failure attributed to a specific vendor during operation.

2.4.3. Maintenance

Another great advantage that digital substations provide is improved maintenance, which can be achieved through all the information the IEDs provide. Usually, time-based maintenance is done to ensure correct operation. This means inspecting and testing system equipment at regular intervals. As IEDs and the communication in digital substations carry a lot of information, it is self-monitoring. This entails that an alarm is automatically set off when something is not functioning correctly, which can be reacted to immediately. In combination herewith, fault corrective maintenance is also carried out to find the root cause for the problems and prevent similar issues from occurring in the future [33].

Furthermore, due to the nature of digital substations providing vast amounts of data, combining these with asset performance management software, a more predictive maintenance approach can be adopted [19] in conjunction with the time-based and alarm-based maintenance. By utilising digital switchgear conjointly with low-power output sensors, catastrophic failure and electrical injuries on personnel can be kept to a minimum. Since heat/humidity and partial discharges are the main cursors for failure, monitoring and logging relevant data can help identify these problems before they lead to erroneous operation. This, in turn, guides the personnel to only the faulty parts of the system and thus lowers the number of hours needed for maintenance, which brings down costs for operation and maintenance (O&M). In addition, it will assist in keeping maintenance staff away from potentially hazardous components, thus significantly reducing the risk of electrical injury [18].

Even disregarding all the data gained from the IEDs, the installation, operation, and maintenance itself would still be cheaper and faster [4, 18, 28, 34].

2.4.3.1. Remote Testing & Maintenance

The things that lead to an improvement in maintenance will also be beneficial in conjunction with remote testing, maintenance, assessment and servicing. Implementing remote tests would decrease the time personnel need to spend on site and could contribute significantly to the overall safety of personnel. Another major motivation for implementing a system with remote maintenance capabilities is the significant savings it could bring. It can quickly become time-consuming for a crew to drive or fly to a remote or offshore location.

However, a notable drawback of remote testing and maintenance is the criticality of cyber security. A secure connection between the remote test computer and the server on the site must be ensured. It is important that only qualified people are granted access rights to this remote testing platform, such that issues stemming from lack of knowledge are avoided [35], and malicious actors are kept out of the system. Where possible, spare access ports used only for testing should be disabled during normal operation for cyber security reasons [36].

Not only should a secure connection be ensured in terms of cyber security, but it is also imperative that connection is not lost in the middle of a test execution. It is, therefore, advised that, where possible, the test is downloaded onto a local device at the site to be executed from there. This guarantees that the test will continue running, even if the communication link to the remote device is lost [36].

Although Apostolov claims that the benefits of improving the efficiency of testing are too significant to ignore [35], there is a lot of reluctance among the utility companies [36]. This is expressed through a survey done by CIGRE, where 23 employees of various utility companies and two system integrators were asked about using the remote testing capabilities for their SAS.

This is seen in Figures 2.5, 2.6 and 2.7. In Figure 2.5, 48% said they did not plan on implementing remote testing for functions of the SAS at all. Of the 20% that intended to use it for control functions, only 12% would also use it for protection functions. One participant in the survey commented: "There are no pressing needs for remote testing".

In Figure 2.7, 40% did not envision taking advantage of the monitoring capabilities supported by IEC 61850 between 2019, when the survey was done, and five years in advance. When asked about a later time horizon, the participation was greater, relatively speaking, where only 28% were dismissive of the idea, and 24% said they would take advantage.

It is important that the remote testing and maintenance are thoroughly tested and validated and that the operators have a good knowledge of the systems. If it is not possible to visually inspect the system and manually interface it during a test, extreme caution must be exercised.

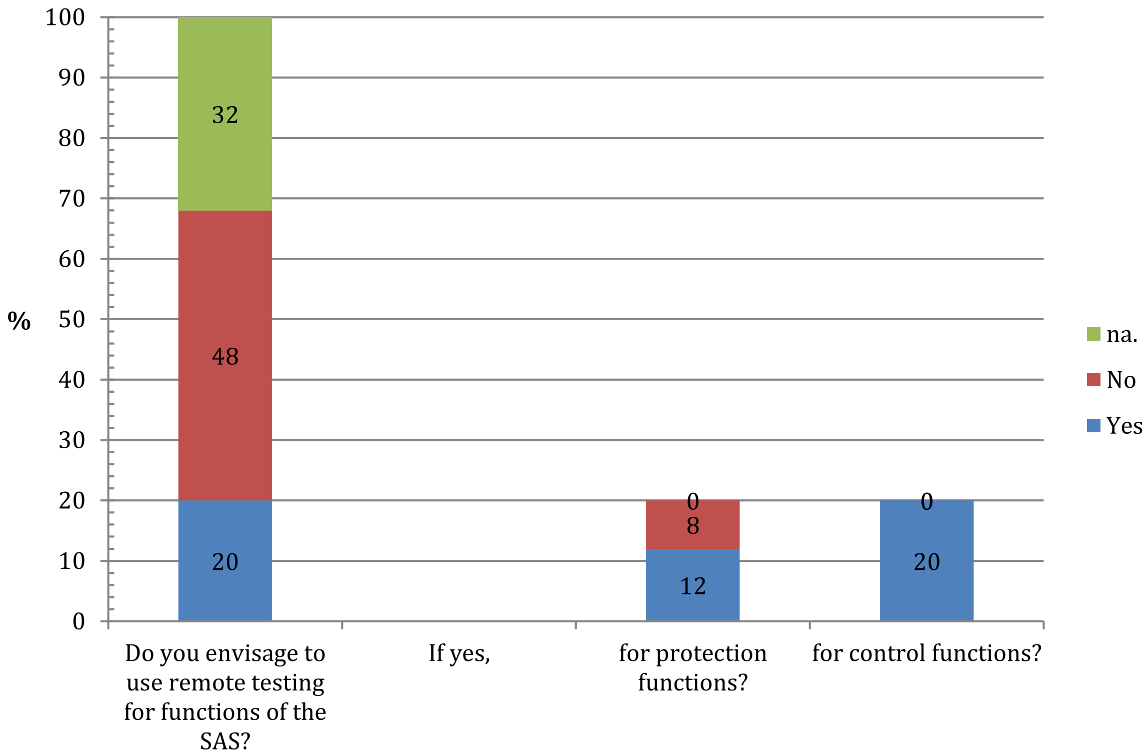


Figure 2.5.: Survey results for the question "Do you envisage to use remote testing for functions of the SAS?". The survey results are from [36].

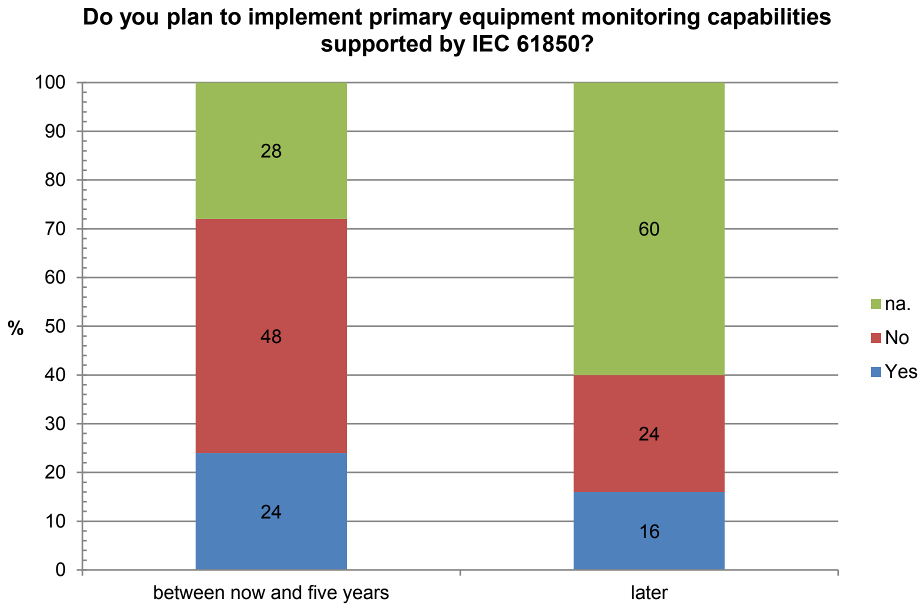


Figure 2.6.: Survey results for the question ”Do you plan to implement primary equipment monitoring capabilities supported by IEC 61850? The survey results are from [36].

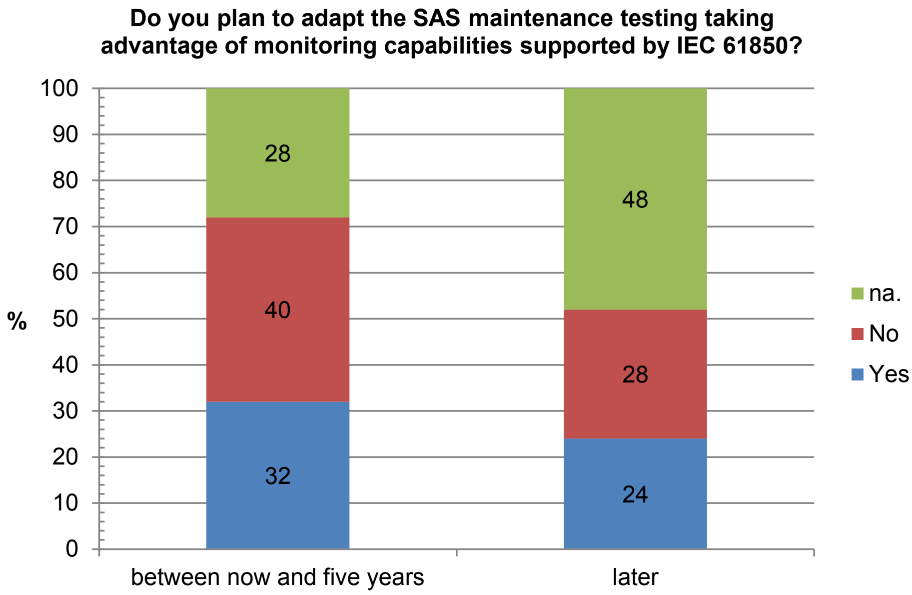


Figure 2.7.: Survey results for the question ”Do you plan to adapt the SAS maintenance testing taking advantage of monitoring capabilities supported by IEC 61850? The survey results are from [36].

2.5. IEC 61850 Standard

The IEC 61850 is a standard made by the International Electrotechnical Committee, which covers and serves to standardise communication protocols and object models for substations. Notably, it covers GOOSE (Generic Object Oriented Substation Event), which is specified in IEC 61850-8-1 [37], Sampled Values (SV) in IEC 61850-9-2 [38] and MMS (manufacturing message specification), which is first introduced in IEC 61850-7-2 [39], but also described in IEC 61850-8-1. Other than the communication structure, more general requirements and the engineering processes are also covered in the other parts of the standard. Part 5, for instance, serves to standardise the communication between IEDs and to detail the system requirements associated herewith [40], and part 6 covers the engineering language used in relation to substation automation, SCL (System Configuration Language). Furthermore, the standard covers the data model that is used by the IEC 61850 compliant devices and covers Logical Devices (LD), Logical Nodes (LN), Data Objects (DO) and Data Attributes (DA) [4]. This can be seen in Figure 2.9. The first 14 released parts of the standard are seen in Table 2.3, and how they relate to one another can be seen in Figure 2.8.

Part	Title (Edition 1)
-1	Technical Report Introduction and overview
-2	Technical Report Glossary
-3	Technical Specification General requirements
-4	International Standard System and project management
-5	International Standard Communication requirements for functions and device models
-6	International Standard Configuration description language for communication in electrical substations related to IEDs
-7-1	International Standard Basic communication structure for substation and feeder equipment—Principles and models
-7-2	International Standard Basic communication structure for substation and feeder equipment—Abstract communication service interface (ACSI)
-7-3	International Standard Basic communication structure for substation and feeder equipment—Common data classes
-7-4	International Standard Basic communication structure for substation and feeder equipment—Compatible logical node classes and data classes
-8-1	International Standard Specific Communication Service Mapping (SCSM)—Mappings to MMS (ISO 9506-1 and ISO 9506-2) and to ISO/IEC 8802-3
-9-1	International Standard Sampled values over serial unidirectional multidrop point to point link Part 9-1 was later withdrawn in favour of Part 9-2
-9-2	International Standard Specific Communication Service Mapping (SCSM)—Sampled values over ISO/IEC 8802-3
-10	International Standard Conformance testing

Table 2.3.: The initial 14 parts of the IEC 61850. Several additions have been made to expand the compatibility and provide information in more unique cases. Furthermore, several of these original parts have been revised, and a second edition has been released. The table is from [4].

In the IEC 61850, a hierarchical data structure is defined. The data model layers are shown in Figure 2.9. Here, it is clearly seen that the physical device is the top level (outermost on the figure). Inherent in this physical device, the logical devices that contain logical nodes (LNs) are found. These LNs have the properties contained in data objects (DO), which finally are comprised of data attributes (DA).

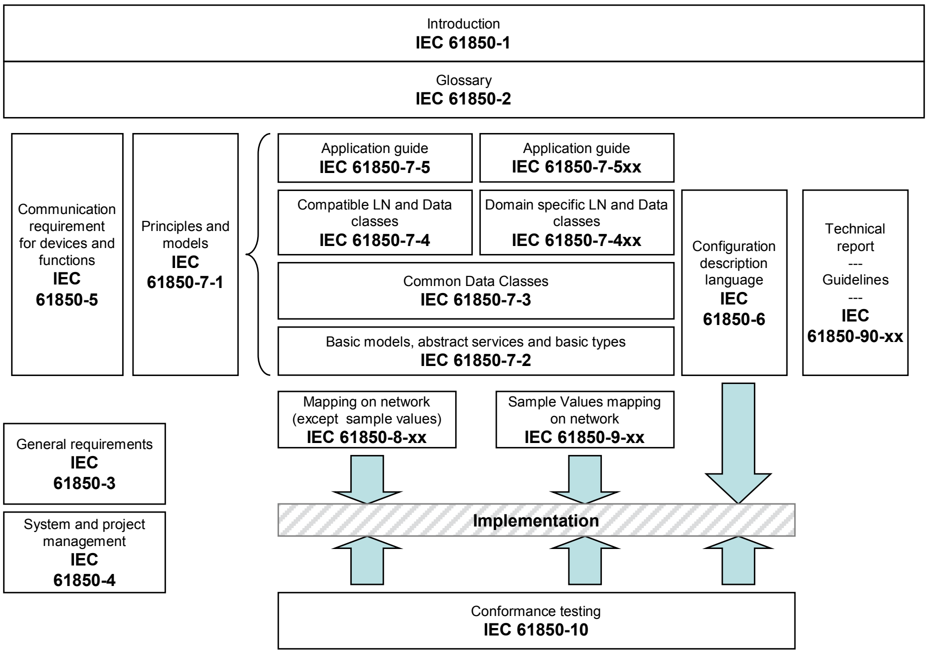


Figure 2.8.: Relationship between the different parts of the IEC 61850. SCL is represented on the right in -6, the principles and models in -7 in the centre and the mapping with GOOSE, SV or MMS in -8 and -9. Together, these are the backbone for getting to the implementation stage. The figure is from IEC 61850-7-1 [41].

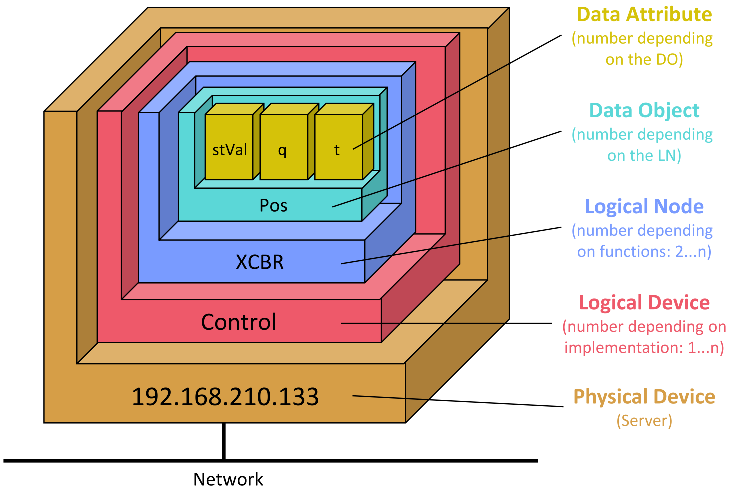


Figure 2.9.: Data model structure in IEC 61850. The figure is from [42].

2.5.1. Logical Nodes (LN)

The LNs are classified into different groups, which makes it easy to identify which LNs provide which functionalities. For instance, every LN prefixed with a P is related to protection functions, such as PTRC (Protection TRip Conditioning), which is the LN responsible for sending a trip signal to a breaker. The LNs prefixed C are related to control functions. A relevant example hereof is the CILO LN, which means ”Control Interlocking” and gives information about the interlocking status at the physical device related to the LN. All the group indicators are listed in IEC 61850-7-4 and are shown in Table 2.4.

Group Indicator	Logical Node Groups
A	Automatic control
B	Reserved
C	Supervisory control
D	Distributed energy resources
E	Reserved
F	Functional blocks
G	Generic function references
H	Hydropower
I	Interfacing and archiving
J	Reserved
K[a]	Mechanical and non-electrical primary equipment
L	System logical nodes
M	Metering and measurement
N	Reserved
O	Reserved
P	Protection functions
Q	Power quality events detection related
R	Protection-related functions
S[a]	Supervision and monitoring
T[a]	Instrument transformer and sensors
U	Reserved
V	Reserved
W	Wind power
X[a]	Switchgear
Y[a]	Power transformer and related functions
Z[a]	Further (power system) equipment

[a] LNs of this group exist in dedicated IEDs if a process bus is used. Without a process bus, LNs of this group are the I/Os in the hardwired IED one level higher (for example, in a bay unit), representing the external device by its inputs and outputs (process image).

Table 2.4.: List of logical node groups in IEC 61850 from IEC 61850-7-4 [43].

To better understand the operations with respect to LNs, consider Figure 2.10. Here, the information being sent to the HMI can be observed. Note that the prefix for the HMI is I since it has to do with interfacing and archiving. Signals from protection functions, both with the prefix P and R (protection-related functions, such as the synchrocheck), are sent to the HMI, as well as the response from the switch controller (CSWI) and the reason for the interlocking (CILO). The full explanation of the abbreviations seen in this figure can all be found in IEC 61850-5 [40]

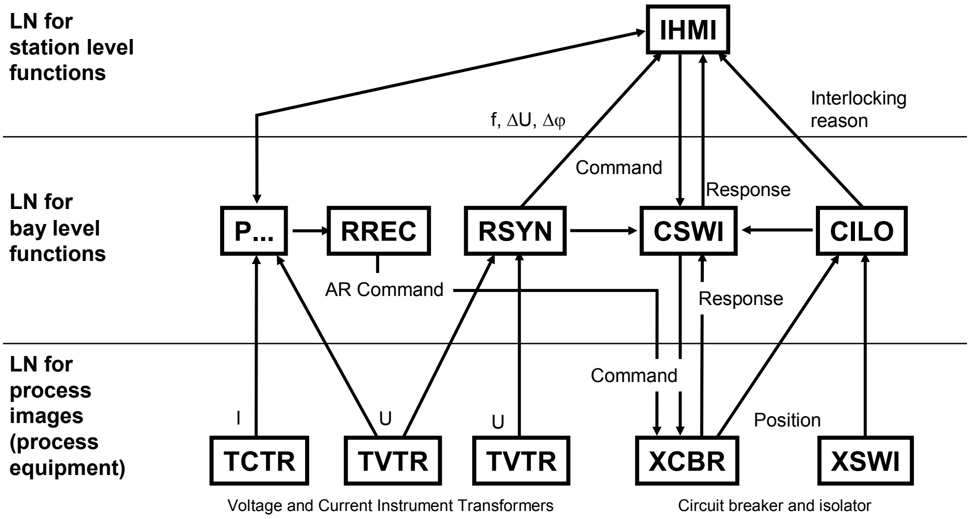


Figure 2.10.: Example for the interaction of LNs for switchgear control, interlocking, synchrocheck, autoreclosure and protection. Figure taken from IEC 61850-5 [40].

Furthermore, it is seen that the autorecloser (RREC), as well as the switch controller (CSWI), send a command to the circuit breaker (XCBR), which together with the isolator (XSWI) (which can be either a load breaker, disconnector, earthing switch or a high-speed earthing switch [40]), transmits their positions to the piece of equipment responsible for issuing a release or blocking (interlocking) command (CILO). Additionally, it is seen that the original trigger for these operations is the reading from the instrument transformers TCTR and the two TVTRs (current and voltage transformers, respectively). Note that this example follows the structure of station, bay, and process levels illustrated previously in Figure 2.1.

2.5.2. SCL

In the engineering process, a digital substation System Configuration Language (SCL) is used. The SCL requires that it must be possible to describe system specifications the same way a single line diagram would. Furthermore, it is necessary that the functionality is described. Thus, the logical nodes (LN) of all parts present must be included. Pre-configured IEDs and their connections to the client/server should also be represented [44]. These files follow XML syntax.

Although numerous IEC 61850-related file formats, which all can be found in IEC 61850-6, exist for different purposes, only the Substation Configuration Description (SCD) and the Instantiated IED Description (IID) files will be covered here, as these were the only files used for the work carried out in this thesis.

To create an SCD file, the workflow depicted in Figure 2.11 can be followed: IEDS are configured in an IED configuration tool, which can then output IID files that can be merged into a system configuration tool to finally produce an SCD file. IED configuration tools may allow for directly producing an SCD file from the IED configurations therein. The SCD can then be further enhanced in a system configuration tool if necessary.

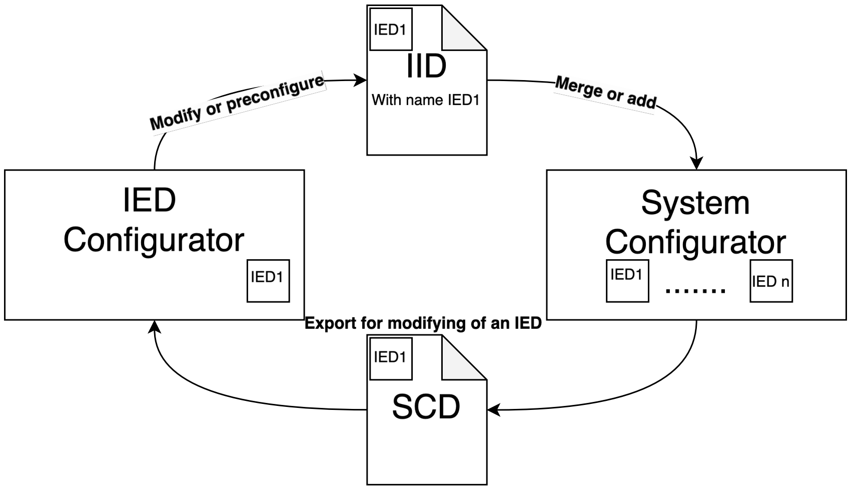


Figure 2.11.: Iterative process of engineering the IID and SCD file. Adapted from [44].

2.5.2.1. SCD

A Substation Configuration Description (SCD) file, conforming to the IEC 61850 standard for digital substations, comprises various sections, each serving a specific purpose. The substation section has information on voltage levels, power transformers, bays, LNs and conducting equipment.

Then, there is the communication section, which contains information on the IP addresses of equipment connected via the station bus and on the MAC addresses of the equipment. Moreover, there are several IED sections, each describing one IED in the system. These sections and the final section, "DataTypeTemplates," reveal how the IEDs are configured, including which LNs and DOs they contain. The IED configuration can be changed in an IED configuration tool, as can be seen from 2.11.

An example showing important elements at the beginning of the substation section in an SCD file is seen in Listing 1. It shows the hierarchical structure that all SCD files follow. At the beginning of the excerpt, the "Substation" group is seen. Like the other opening tags found in the SCD file, the "Substation" tag has attributes. In this case, a description, name, and x and y coordinates. Below it in the hierarchy lies the voltage level, which has the name attribute. Within the voltage level lies the actual voltage magnitude of this particular voltage level, here the high voltage side of 320 kV is shown. The voltage level also has different bays as its children, as there can be multiple bays within one voltage level. These bays, in turn, contain logical nodes and conducting equipment. Line 13 and line 20 are highlighted as these show the switching equipment QA1 and QB1 being a CB and a disconnector, respectively. Importantly, these pieces of equipment are the parents of various logical nodes. Here, "CILO", "XCBR", and "CSWI" are seen. The CILO LN is highly relevant and must exist for automatic interlocking to be feasible, as it provides the interlocking status of the physical device. By parsing the SCD file, all information it provides can be accessed.

2.5.2.2. IID

The IID describes only one IED and can be generated from an IED configuration tool. These tools are often proprietary and thus specific to the vendor of the given IED. For instance, ABB uses their own software called PCM600. From this software, the internal logic and configurations

```xml
1 <Substation desc="Munich" name="AA1" sxy:x="1" sxy:y="5">
2     <VoltageLevel name="D1">
3       <Voltage multiplier="k" unit="V">320</Voltage>
4       <Bay desc="TF1" name="Q01" sxy:x="1" sxy:y="2">
5         <LNode iedName="AA1D1Q01Q1" ldInst="PROTECTION" lnClass="PDIS" lnInst="1"/>
6         <LNode iedName="AA1D1Q01Q1" ldInst="PROTECTION" lnClass="PTOC" lnInst="1"/>
7         <LNode iedName="AA1D1Q01Q1" ldInst="T3S1S1" lnClass="ATCC" lnInst="1"/>
8         <LNode iedName="AA1D1Q01Q1" ldInst="T3T1P1" lnClass="YLTC" lnInst="1"/>
9         <LNode iedName="AA1D1Q01Q1" ldInst="T3T1P1" lnClass="YPTR" lnInst="1"/>
10        <LNode iedName="AA1D1Q01Q1" ldInst="PROTECTION" lnClass="PTRC" lnInst="1"/>
11        <LNode iedName="AA1D1Q01Q1" ldInst="PROTECTION" lnClass="RBRF" lnInst="1"/>
12        <LNode iedName="AA1D1Q01Q1" ldInst="MEASUREMENT" lnClass="MMXU" lnInst="1"/>
13        <ConductingEquipment name="QA1" sxy:y="5" type="CBR">
14          <LNode iedName="AA1D1Q01Q1" ldInst="QA1" lnClass="CILO" lnInst="1"/>
15          <LNode iedName="AA1D1Q01Q1" ldInst="QA1" lnClass="XCBR" lnInst="1"/>
16          <LNode iedName="AA1D1Q01Q1" ldInst="QA1" lnClass="CSWI" lnInst="1"/>
17          <Terminal bayName="Q01" cNodeName="L11" name="L11" substationName="AA1" voltageLevelName="D1" connectivityNode="AA1/D1/Q01/L11"/>
18          <Terminal bayName="Q01" cNodeName="L12" name="L12" substationName="AA1" voltageLevelName="D1" connectivityNode="AA1/D1/Q01/L12"/>
19        </ConductingEquipment>
20        <ConductingEquipment name="QB1" type="DIS">
21          <LNode iedName="AA1D1Q01Q1" ldInst="QB1" lnClass="CILO" lnInst="1"/>
22          <LNode iedName="AA1D1Q01Q1" ldInst="QB1" lnClass="XSWI" lnInst="1"/>
23          <LNode iedName="AA1D1Q01Q1" ldInst="QB1" lnClass="CSWI" lnInst="1"/>
24          <Terminal bayName="BB1" cNodeName="L1" name="L1" substationName="AA1" voltageLevelName="D1" connectivityNode="AA1/D1/BB1/L1"/>
25          <Terminal bayName="Q01" cNodeName="L1" name="L1" substationName="AA1" voltageLevelName="D1" connectivityNode="AA1/D1/Q01/L1"/>
26        </ConductingEquipment>
```

Listing 1: The beginning of the substation section of an SCD file. The start of the two conducting equipment sections that describe the circuit breaker and the disconnector of bay 1, respectively, are found on lines 13 and 20 and are highlighted in light blue.

of the IED can be created or modified. From this tool, it is then possible to export the IID, which, along with other IIDs, can be loaded into the system configuration tool and produce an SCD. However, from PCM600, it is also possible to create an SCD directly.

These IID files collectively define the configuration of the digital substation, providing a standardised and machine-readable representation of the substation's logical and physical structure, communication details, and associated devices. The detailed specifications within each IID enable interoperability between different IEDs and ensure consistent communication and control in a digital substation environment.

2.5.3. IEC 61850 Protocols

Due to the powerful nature of GOOSE and Sampled Values in IEC 61850, it is often mistakenly assumed that the IEC 61850 standard is merely a standard of protocols. It should be clear by now that it is much more than that. However, the protocols are powerful, and an account of GOOSE and Sampled Values, which are the protocols used for protection and control-related services [4], will be given. In addition, Manufacturing Message Specification (MMS), which is a third IEC 61850 protocol, will be explained in brief. This protocol is client/server based, unlike GOOSE and SV, which are publisher/subscriber based. This also means that MMS can leave the local network contrary to GOOSE and SV.

2.5.3.1. Manufacturing Message Specification (MMS)

This is an ordinary client/server protocol and is tied to IP addresses and can cross routers, meaning it is not restricted to the local area network. MMS is as seen in Figure 2.12 used for ”Vertical Traffic”. That implies that MMS is used to communicate between substation layers, for example, from station level to bay or process level. It is used extensively for messages related to SCADA functions, such as monitoring and event logging. The MMS protocol is not as fast as, for example, GOOSE and is therefore not used for protection and control (PAC) related functions, but rather for data acquisition [4, 9].

2.5.3.2. Generic Object-Oriented Station Event (GOOSE)

GOOSE is defined in IEC 61850-8-1 [37] and allows for communication between bays, which in Figure 2.12 is referred to as horizontal traffic and between process level (primary equipment) and bay level, which is indicated in the left-most bay in Figure 2.12. The signals are transmitted via station or process bus but cannot leave the LAN network. GOOSE signals are event-based, as the name indicates, which means that upon triggering an event configured in the IED, it transmits a burst of GOOSEs that carry information regarding that particular event. Immediately after the event, the GOOSE signals are sent in rapid succession, after which the time between bursts increases gradually. This is seen in Figure 2.13. In order to protect against loss of packet information, the GOOSE signals are always retransmitted at regular intervals to ensure correct receipt of the sent message. Furthermore, another fail-safe is implemented to verify that the connection is established. The device constantly transmits its current status, known as the heartbeat, if no event occurs. This status signal is transmitted every T_0 seconds, which is an application-specific time. If the receiver does not receive the transmitted message within a predefined time, usually $2T_0$, it is determined that the message is lost. This predefined time is known as the ”Time Allowed to Live” (TATL or TAL). If $4T_0$ elapses, then it signals that the GOOSE communication is lost entirely, and a broken link alarm signal is sent [9]. The messages

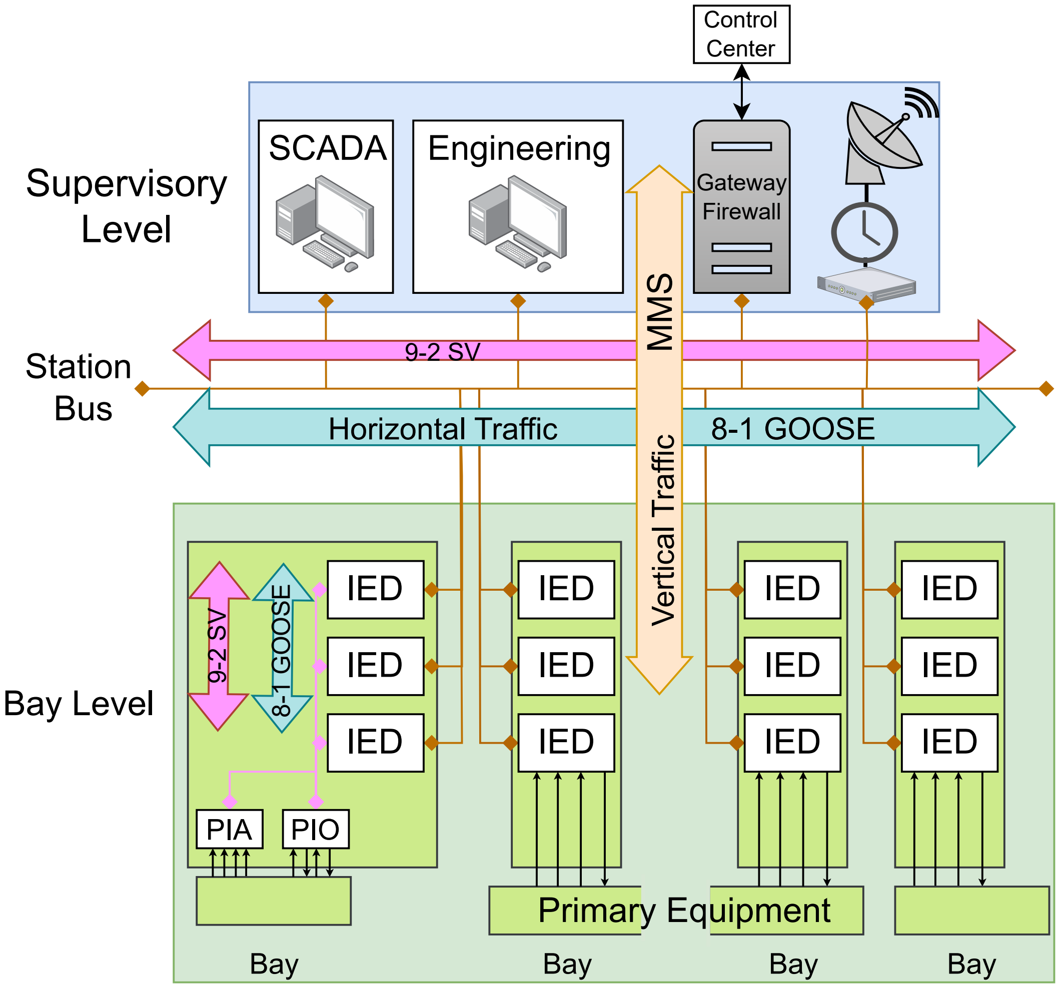


Figure 2.12.: Overview of communication in digital substations. The figure is adapted from [4].

are multi-cast, meaning they are broadcast to the entire network, but only devices subscribing to the GOOSE message will parse them [4].

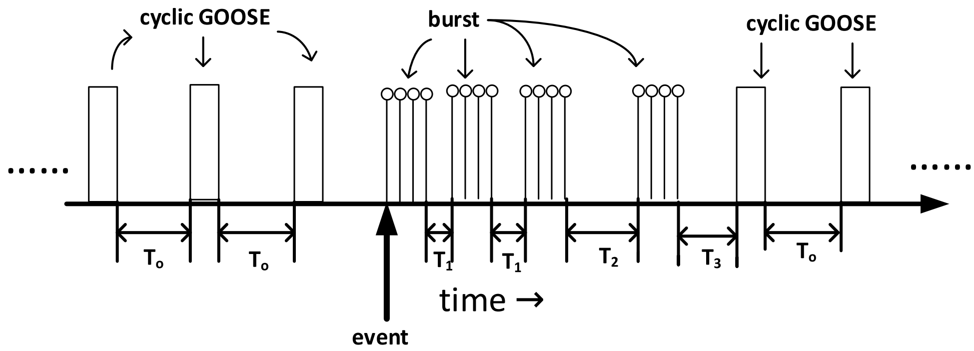


Figure 2.13.: GOOSe message retransmission in IEC 61850. The cyclic retransmission is shown with constant time intervals of T_0 between them, and the rapid bursts, after an event, are sent with gradually increasing transmission intervals, here shown as $T_1 ... T_3$. The figure is from [45].

GOOSE signals are used for sending tripping commands, and it is dictated that such a command has a transmission time from the protection IED to the MUs of no more than 3 ms [4]. This transmission time is also helped by the fact that multi-cast communication is used instead of serial, which is more time-efficient as all transmitted messages are broadcast simultaneously instead of sequentially [31]. These signals are, therefore, due to their reliability and relatively fast transmission times, suitable for implementing interlocking, tripping schemes and more, which can make hardwired schemes obsolete in digital substations. If implementing GOOSE in this way, it is critical that this is considered during the engineering stage, such that it is possible to simulate and monitor GOOSE signals on the station bus [33].

As the transmission times of GOOSE messages are so critical, it is difficult to afford much in terms of encryption. Therefore, it is pointed out in the IEC 61850-1 that the only security measure used in conjunction with GOOSE signals is authentication through a digital signature, which is described in IEC 62351-6 (Security for IEC 61850) [46, 47].

2.5.3.3. Sampled Values (SV)

The SV protocol is defined in IEC 61850-9-2 [38] and is predominantly used for transmitting analogue current and voltage values recorded in the sensors to the IEDs. This protocol does not include the same retransmission strategy as GOOSE offers. Here, the next sample is transmitted, overwriting the previous one. Another difference between SV and GOOSE is that the SV is always transmitted at the same interval. This interval should be 250 µs in a 50 Hz grid, equating to 4000 samples/s. As SV is also publisher/subscriber based, it is implied that the protocol is restricted to the LAN [4, 9].

2.5.4. Merging Units (MU)

As the name suggests, a merging unit (MU) is intended to merge several currents and voltages coming from CTs and VTs. MUs can be separate devices, known as stand-alone merging units (SAMU), or they can be embedded in LPITs (EMU). There are also merging units accepting analogue signals as inputs and merging units that require digital signal inputs. Whichever type of MU is chosen, their outputs should theoretically be indiscernible. In practice, however, the SAMU will often have lower accuracy than the EMU as the accuracy of the IT and the SAMU are cascaded [48]. Both EMU and SAMU will output digital values compliant with IEC 61850.

This can be either SV or GOOSE [49]. This data should be synchronised with a common external time source to ensure synchrony in data capture. This external time source is used for all MUs in the substation to ensure that when a relay receives information from several MUs, they can be correlated for correct operation. The synchronisation method should preferably rely on the Precision Time Protocol (PTP) as specified in IEC 61588, in which the MUs internal timing systems are synchronised to a station clock (grandmaster clock) that is synchronised to GPS or GLONASS [50].

While digital interfaces for ITs are covered in general in IEC 618969-9 [48], IEC 618969-13 [51] covers only SAMU in detail.

2.6. Interlockings

The overarching purpose of interlockings is to prevent switchgear mal-operation [31]; Interlocking exists to avoid energising or de-energising components using equipment not intended for this purpose, such as a disconnector. In essence, this means that a device not suited for making or breaking current must not operate if it is energised or the current is not commutable to a parallel branch. Similarly, interlocking ensures that earthing switches cannot close on an energised component [52, 53].

These requirements are put in place to ensure the safety of personnel, protect the equipment, and make the behaviour of the switchgear predictable [53].

The interlocking functionality is often divided into two groups for substation interlockings. These are bay interlockings and inter-bay (or cross-bay [54]) interlockings. The inter-bay interlockings cover the interlockings resulting from within one bay, whereas the cross-bay interlockings also consider other bays. This means that when considering the interlocking of a particular device in a bay, equipment may exist in another bay that influences the operation of that particular bay [31].

The interlocking logic can be processed either in a centralised or decentralised manner. A hybrid version is also possible. The characteristics of the three types are described below [55]:

- **Centralised** interlockings are all calculated in a single device, and the outcome is communicated to the respective bays. This has the advantage of being simple and well structured, however, it offers no redundancy, meaning that a failure of this central unit would be catastrophic as this would prevent interlockings from occurring.
- **Decentralised** interlockings do not lack redundancy, as the release signals are computed in each individual bay controller by processing the relevant switchgear positions. The drawback of this implementation is that it is more complex, and an expansion of the substation could require updating all bay devices.
- **Hybrid** interlockings are realised by implementing the decentralised scheme, but in addition, the bay devices send information about the switch position of their respective bays to a station controller, which can calculate the topological information of the entire substation and communicate this to the decentralised bay devices. From this information, the bay device can determine whether or not an interlock is required. This means that even if the central device fails, decentralised interlocks are still in place.

2.6.1. Interlocking in Digital Substations

The purpose of interlockings does not change whether they are implemented in a conventional or digital substation. The main difference is that the interlocking signals are usually hard-wired for conventional substations, whereas they are sent via GOOSE messages for digital substations. The implementation of the interlocking logic schemes lies within the IEDs themselves but can also lie within a centralised bay control unit (BCU).

Unfortunately, even though the interlocking logic is configured for the individual IEDs in an IED configuration tool, extracting this data along with the IID file is not yet possible. This capability is also not supported in the SCD file yet, making it virtually impossible to generate an interlocking test case from just an SCD file. However, advancements are being made within the IEC, where they are working on extensions for the standard that make it possible to include logic in the SCL environment.

2.6.1.1. Future Advancements Within SCL

The IEC 61850-90-11 [56] is a first step towards integrating logic specification into the IEC 61850 environment and addresses how to describe logic and how to link logic variables into the IEC 61850 data model. This part of the standard is available but does not see use [57, 58].

Brunner states that by employing the basic application profiles found in IEC 61850-7-6 [59] in conjunction with a new part that working group 10 of the IEC technical committee 57 (IEC TC57 WG10) is working on called IEC 61850-90-30, it will be possible to describe the behaviour of an application [58, 60]. An "application" as defined in IEC 61850-7-6 is a subset of the standard covering a specific application or function, such as substation interlocking [59]. The logic behaviour could, in this case, be described using IEC 61131. This would remedy the problems concerning the production of engineering files using IEC 61131. Currently, it is not specifically defined which file format should be used when specifying logic with IEC 61131. Although it could produce an XML file, it would not be immediately able to be integrated into an SCD file that could work with any vendor. Thus, it would not be possible to extract the logic configuration in the SCD for any vendor [31]. Providing recommendations to IEC TC57 WG10 is the European research project Optimal System-Mix Of Flexibility Solutions For European Electricity (OSMOSE) [61]. They also advise that an inclusion of standardised logic based on IEC61131 is made:

> Using standardised IEC61131 logics to describe system behaviour at specification and implementation levels improves the ability to automate testing and provide more rigorous simulation. At the implementation level, the IEC61850-90-11 provides ways to do this, and it is recommended for WG10 to also enable this at specification level [62].

2.6.2. Test Mode and Simulation Mode

Simulation and test modes play a significant role in testing within the framework of the IEC 61850 standard. These modes provide the ability to simulate various conditions and scenarios encountered in digital substations. This offers several benefits, including the ability to conduct comprehensive and controlled testing, reduced risks of equipment damage, and improved safety for personnel involved in the testing process. Importantly, it also allows for testing a function in isolation while the rest of the system is still in operation. Toggling the simulation flag allows for the simulation of data attributes without activating the output contacts to the substation control unit (SCU), for example. Understanding simulation and test modes and their benefits and limitations is crucial for conducting effective and reliable tests in digital substations [6].

2.6.2.1. Test Mode

Test mode relates to the different values that can be selected to put a function in a specific mode or follow a specific behaviour. These different values of the modes and behaviours (Mod and Beh) are described in Table 2.5. Mod and Beh can always be accessed even if the device is placed in the "off" mode. This is to guarantee that the device can always be put in a different mode despite all other signals not being processed [43].

An example is given in Figure 2.14, where some function, XXYZ1, has its behaviour set to test mode via the Beh.stVal=test as seen in the yellow box. This allows the function to process an incoming GOOSE signal that has its q (quality) value set to "test". The same is true for the control service signal, as this has its test parameter set to true [41]. If there is no match between the behaviour of the function receiving the data and the parameters and settings of the incoming signals, the data will be treated as invalid.

A more interesting example, perhaps, is to consider a function with its behaviour set to "on" as opposed to "test", as a logical node with Beh.stVal=on will reject all incoming data with the test flag set to true and treat it as invalid data. This provides an isolation point when testing, as the logical node in the normal operating mode is unaffected by the test [63].

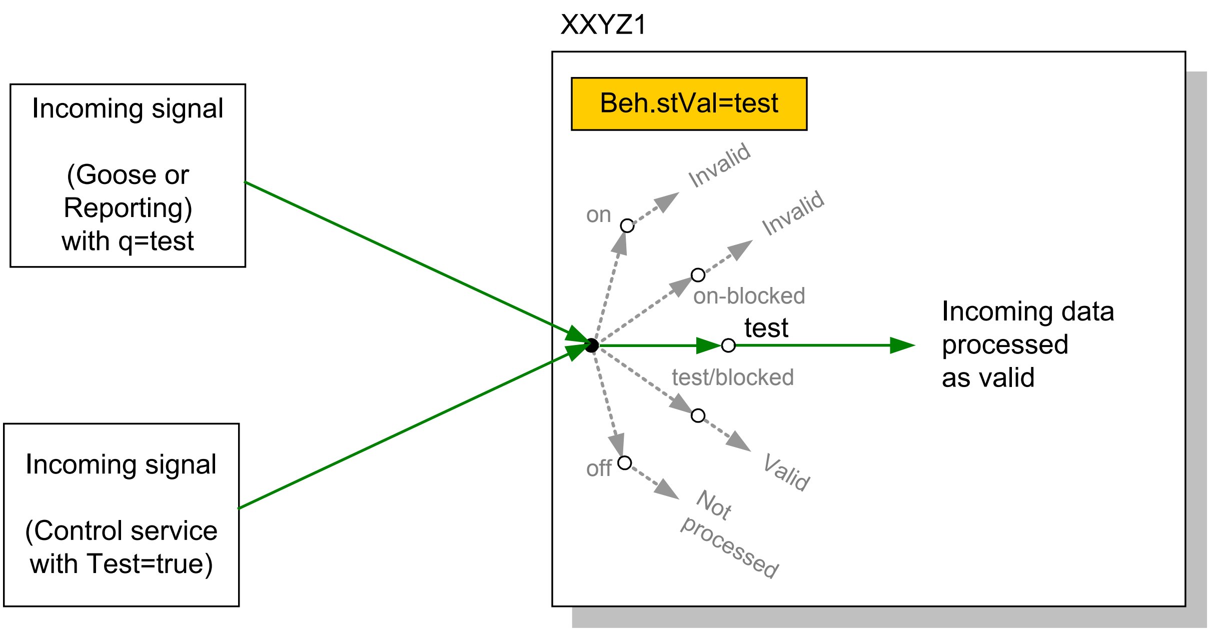


Figure 2.14.: Test mode example from [41]. The mode of the function (Beh.stVal) must be set to "test" or "test/blocked" to accept the incoming signals with the quality value set to true and the control service parameter set to true respectively [41].

2.6.2.2. Simulation Mode

Putting an IED in simulation mode is done by changing the Sim.stVal in the LN LPHD1 of the IED to true. This results in the IED prioritising simulated signals where they are available.

This means that a multi-cast GOOSE signal can still be transmitted to all devices that require it, but for the DUT, a simulated signal can be sent, which the IED would prioritise.

This is illustrated in Figure 2.15, where a GOOSE 1 message coming from the simulation device with the simulation bit set to true is selected and processed instead of a simultaneously incoming, almost identical, GOOSE 1 message from an actual device. The only difference is whether or not the simulation bit is true. In this example, the simulated signal is processed because the Sim.StVal data attribute of LN LPHD1 of the IED is set to true. Note that the IED continues to process other signals without their simulation bit set to true. It is also important to note that the two GOOSE signals at the bottom of Figure 2.15 would not be processed if, at any time, a new GOOSE 2 or GOOSE 3 with the simulation bit toggled is received by the IED. This holds true for SV as well [41].

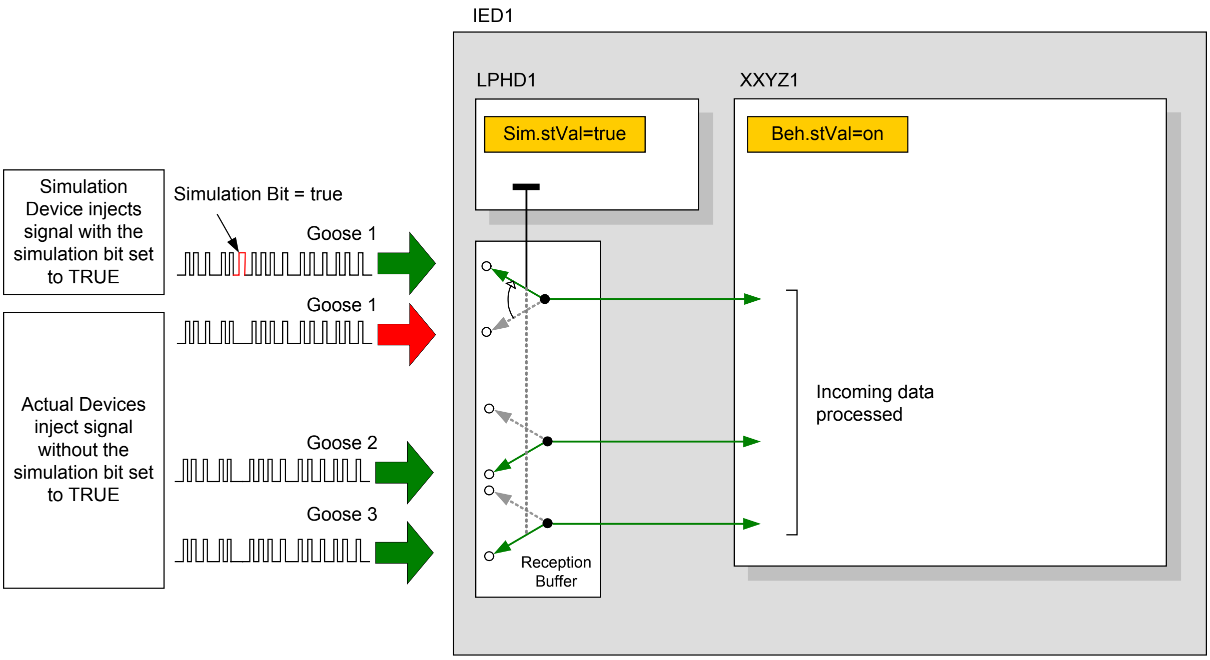


Figure 2.15.: Data used for receiving simulation signals. The selectivity associated with the simulation mode is shown. Two almost identical versions of the same GOOSE signal are transmitted, but as the IED is in simulation mode, it prefers the simulated version, where it is present. The figure is from [41].

2.6.3. Automated Workflow Based on StationScout

As the intent of this work is to devise a way of automatically testing interlockings in IEC 61850-based substations and with the current lack of support for logic-based applications in the SCD file format from IEC 61850, the method employed in this thesis will focus on providing a minimum amount of inputs to limit the manual interaction required instead of relying on any SCL files. In a recent study, a way of optimising the workflow when testing interlockings was found [55]. This method relies on test cases in the JSON file format, which is compatible with the Omicron SAS test software StationScout [64]. What Tahincioglu, Schossig and Heimisson did was to create a VBA (Visual Basics for Applications) script that would allow for importing an already existing JSON test case, created in StationScout, into Excel, where it could be reconfigured as desired and then exported as a JSON once again. A simple example in the Excel macro tool is shown in Figure 2.16.

Automating a FAT for a bay interlocking scheme has been shown to be more efficient than conventional methods despite the fact that the approach used is not fully optimised. Loenders,

Table 2.5.: Values of mode and behaviour from [43].

Value	Mode	
1	on	The application represented by the LN works. All communication services work and get updated values
2	on-blocked	The application represented by the LN works. No output data (digital by relays or analogue setting) will be issued to the process. All communication services work and get updated values. Data objects will be transmitted with quality “operatorBlocked”. Control commands will be rejected. See note below Table A.1.
3	test	The application represented by the LN works. All communication services work and get updated values. Data objects will be transmitted with quality “test”. Control commands with quality test will be accepted only by LNs in “test” or “test-blocked” mode. “Processed as valid” means that the application should react in the manner what is foreseen for “test”.
4	test/blocked	The application represented by the LN works. No output data (digital by relays or analogue setting) will be issued to the process. All communication services work and get updated values. Data objects will be transmitted with quality “test”. Control commands with quality test will be accepted only by LNs in TEST or TEST-Blocked mode.
5	off	The application represented by the LN doesn’t work. No process output is possible. No control command should be acknowledged (negative response). Only the data object Mod and Beh should be accessible by the services.
NOTE: The Mod=”blocked” from edition 1 is changed in edition 2 to “on-blocked”.		

Description		Test Step 1		Test Step 2	
		Value	Order	Value	Order
AA1E1Q01BCUDigSS_QB1_Disc/CSWI1.Pos.stVal	C	POS_OFF		POS_OFF	
AA1E1Q01BCUDigSS_QA1_CB/CSWI1.Pos.stVal	C	POS_OFF		POS_ON	
AA1E1Q01BCUDigSS_QC9_EarthSw/CSWI1.Pos.stVal	C	POS_OFF		POS_OFF	
AA1E1Q01BCUDigSS_QB2_Disc/CILO1.EnaOpn.stVal	A	TRUE		FALSE	
AA1E1Q01BCUDigSS_QB2_Disc/CILO1.EnaCls.stVal	A	TRUE		FALSE	

Figure 2.16.: A simple example of a test configuration in the Excel Macro tool from [55]. Note that the order is not filled out here.

Lisiecki and Yesil estimated that a time reduction of approximately 45% could be achieved. The time spent is segmented into different parts and shown in Table 2.6, where the automatic times are compared to the time the conventional method takes. The times for the conventional testing are agreed upon by an industrial consortium. Furthermore, it is mentioned how additional time gains could be made if it was possible to read or transfer the logic implemented in the IEDs to the SCL files [57].

Table 2.6.: Comparison of time taken for various activities in conventional and automatic methods. The digitalisation of the process bus will eliminate preparation steps (*) as a direct consequence of the reduction of copper wiring, simplifying electrical plans. Corrective measures (**) are less associated with incorrect wiring and more so with identifying input signals and requiring manual intervention [57].

Activity	**Time (min)**	
	Conventional	**Automatic**
Interpretation of electrical plans	60	0
Programming the interlocking diagram	60	60
Preparations to facilitate interlock testing (*)	130	30-120
Testing the interlocking schemes	30	1
Corrective measures (**)	1-60	1-60
Total time	281 - 340	92 - 241

The experimental work of this thesis will take a starting point in the method from Tahincioglu, Schossig and Heimisson. An attempt will be made to improve upon it and make it more accessible, automatic, safe and future-proof. For instance, the VBA script only runs on Windows-based operating systems as it is tied to the file explorer hereof, and the order must be set in the tool instead of automatically assigned. Furthermore, there are cyber security concerns when using VBA, as Excel can easily access outer resources in the computer system [65]. In general, a lot of care must be exhibited when interacting with macros from VBA, as they are a common way for people with malicious intent to initiate a cyber-attack [66]. It is even advised that macros be turned off in organisations where they are not used [67]. Methods of distinguishing malicious VBA macros from benign ones have been developed but do not offer a 100% success rate. The methods themselves are outside of the scope of this thesis.

Chapter 3

Test Case Generation

3.1. Test Case Prerequisites

A simple substation configuration is used to prove the concept of automatically generating a test case that can be used in the execution of an interlocking test. The configuration being studied is seen in Figure 3.1. It is a two-bay system, where Q01 is a feeder bay containing two disconnectors, QB1 and QB2 and one CB, QA1. It is assumed that the CT, VT and, by virtue, the MU record values that indicate that the feeder is not energised. This effectively means that QC9 is permanently in the open position, which is why it will not be considered going forward. Similarly, it is assumed that synchrochecks will never be a problem. The other bay is Q05, the bus coupler bay, which contains two disconnectors, QB1 and QB2, a CB, QA1, and finally, an earthing switch for each bus. These earthing switches are controlled simultaneously in this system and will, therefore, be given only one name, QC11. In this system, as a result of the simplifications implemented, only the switch positions and the associated logic are considered. This approach is used to efficiently prove or disprove the adequacy of the employed method.

The interlocking logic of these bays is shown with a logic circuit in Figure 3.2. Note that the diagram could be expressed in several ways, but how it is depicted here clearly shows the importance of the CB in bay one, QA1, and the earthing switch in bay five, QC11. This is apparent, as these both are inputs to the final AND gate, i.e. QB1 will always be interlocked if QA1 in bay one or QC11 in bay five are closed. Similarly, it becomes clear that all switchgear of the bus coupler must be closed for the first AND gate to produce a high output. When used in conjunction with Table 3.1, it is seen that this diagram holds true for all of the different topologies tested [65].

From this particular logic, a truth table can be created. It is, however, unnecessary to consider every permutation, and utility companies will only consider the most common and critical topologies as they are the ones who define the requirements and interlocks [57]. Different utilities might also conduct tests on an entirely different combination of test steps, although the interlocking logic of the systems is the same.

The seven steps shown in Table 3.1 are the ones considered for this particular bay system. Only one step has the bay one CB closed, and similarly, only one step has the bay five earthing switches, QC11, closed. This is due to the aforementioned fact that whenever these pieces of equipment are closed, the release signal sent from the CILO logical node will evaluate to false and thus result in interlocking, cf. Figure 3.2, and it is often desirable to test the interlocking function of such a step at least once.

It is, therefore, clear why steps one and six result in the interlocking of the bay one disconnector QB1, which is the DUT. Step two evaluates to true, as it is always permissible to close the DUT

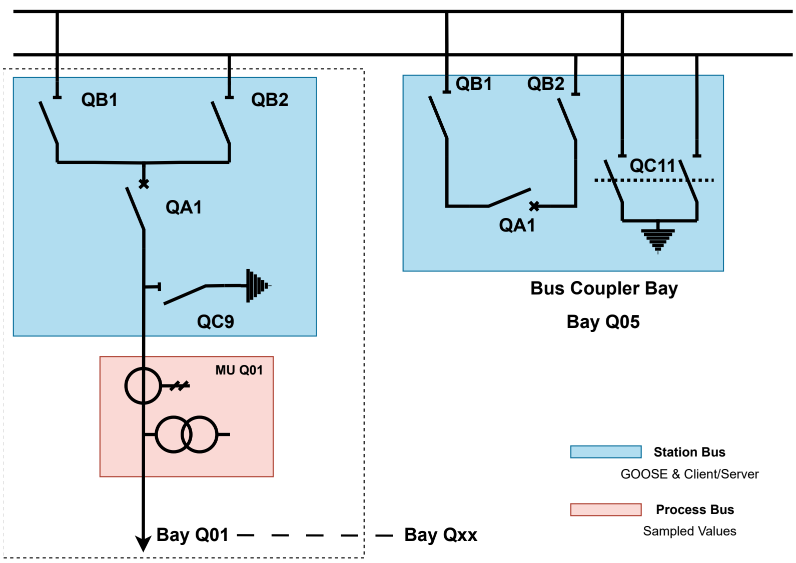


Figure 3.1.: Example of a double bus system with a bus coupling (BC) bay. Only one bay other than the BC bay is shown.

Table 3.1.: An example of a test sequence, where the DUT is QB1 of bay one. The first row is the number of test steps, and the second row is the assessment of the DUT. A "FALSE" assessment signifies that the disconnector is interlocked and thus not allowed to operate, and "TRUE" denotes that it is allowed to operate. The state changes from one test step to another are highlighted in yellow and bold and are found in rows three through eight. The final row is the AddCause [65].

Test Step Number	1	2	3	4	5	6	7
Q01 QB1	FALSE	TRUE	FALSE	TRUE	FALSE	FALSE	TRUE
Q01 QA1	**CLOSED**	**OPEN**	OPEN	OPEN	OPEN	OPEN	OPEN
Q01 QB2	OPEN	OPEN	**CLOSED**	CLOSED	CLOSED	**OPEN**	OPEN
Q05 QA1	OPEN	OPEN	OPEN	**CLOSED**	**OPEN**	OPEN	OPEN
Q05 QB1	OPEN	OPEN	OPEN	**CLOSED**	CLOSED	**OPEN**	OPEN
Q05 QB2	OPEN	OPEN	OPEN	**CLOSED**	CLOSED	**OPEN**	OPEN
Q05 QC11	OPEN	OPEN	OPEN	OPEN	OPEN	**CLOSED**	**OPEN**
Q01 QB1	CMD	CMD					

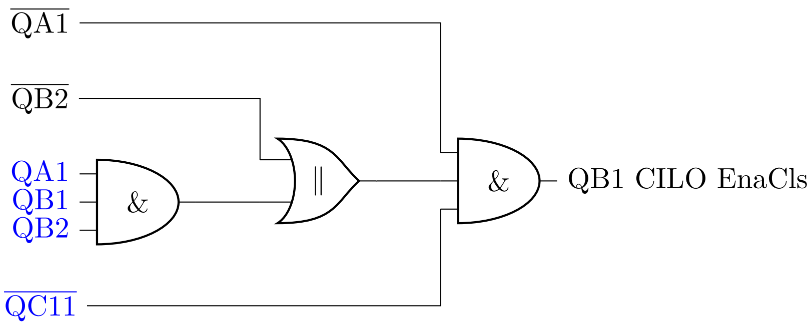


Figure 3.2.: Logic schematic of the interlocking scheme. Blue input signals denote that the signal is coming from bay five, whereas black inputs are sent from bay one. The default value for a logic high signal is closed. Note that the position signals of QA1 and QB2 of bay one, as well as QC11 of bay five, are inverted, meaning that if these switches are closed, it will result in a logic 0 instead of a 1.

switch when all other switches, CBs and earthing switches are opened, as that line segment will be isolated regardless of closing DUT. Step three results in interlocking as closing DUT after closing Q01 QB2 would result in short-circuiting the two buses. Step five also allows the operation of the DUT, as all the elements of the bus coupler, disregarding the earthing switch, are already activated. Therefore, closing the DUT would not have an impact in this configuration. This is unlike step five, where the CB of bay five is not closed. Here, the buses are independent, and closing the DUT would short-circuit them, which is not permissible. Finally, as step six is already covered, step seven is naturally also not limiting the operation of the DUT as it is identical to step two. This test step does not provide additional information, but leaving the switch gear in a safe (open) state is a good practice after running a test.

The ”CMD” at the bottom of columns 1 and 2 signify that an AddCause command is associated with these test steps. The control command, AddCause, gives information based on the position of the switchgear under test to check that the CILO signal is consistent with the actual switchgear control command [65]. This is only done twice, once for a false and once for a true assessment, as it is sufficient to verify that the expected AddCause in these scenarios works. In the remaining steps, the AddCause would be ”None” if going by the implemented AddCause descriptors of IEC 61850-7-2 [39] or in the case of using OMICRONS's software ”CAR-NO-OPERATION” due to the extension they have provided for positive AddCause messages. When the reply from the DUT is positive, a position change of the switchgear that the command is issued to results. Therefore, for these positive cases, the test system uses a special selection called: ”CAR_POSITION_CHANGED” to match the positive expectation with a position change of the switchgear, thus providing greater feedback than what is specified in IEC 61850-7-2 [65].

In addition to Table 3.1 containing the test steps, the signal addresses must also be known. These are essentially paths that point to the exact DA in the IEC 61850 data model. This could, for instance, be `AA1D1Q01QB1/CSWI1.Pos`, if one wanted to access the position of the ”QB1” disconnector in bay ”Q01” in the voltage level ”D1” in the substation ”AA1”. These names do not have to follow this convention, although doing so makes it easy to determine their origin and structure. An example of a table of signal addresses is seen in Table 3.2.

Finally, to make the script more general, the associated SCD file is imported within it. This also nicely sets the script up for the future when interlocking logic and single line diagrams become available in the SCD file itself.

Table 3.2.: Signal addresses used for control, assessment and reading the AddCause.

IED Name	Signal Address
AA1D1Q01Q1	AA1D1Q01Q1QB1/CILO1.EnaCls
AA1D1Q01Q1	AA1D1Q01Q1QB1/CSWI1.Pos
AA1D1Q01Q1	AA1D1Q01Q1QA1/CSWI1.Pos
AA1D1Q01Q1	AA1D1Q01Q1QB2/CSWI1.Pos
AA1D1Q05Q1	AA1D1Q05Q1CBSW/XCBR1.Pos.stVal
AA1D1Q05Q1	AA1D1Q05Q1CBSW/XSWI1.Pos.stVal
AA1D1Q05Q1	AA1D1Q05Q1CBSW/XSWI2.Pos.stVal
AA1D1Q05Q1	AA1D1Q05Q1CBSW/XSWI3.Pos.stVal

3.2. Information from SCD

While the full support for implementing logic applications in SCL is not there yet, extracting information from the SCD file may still be useful. It can be used to cross-reference the signal addresses and verify that they are all present in the substation and that their names are correct.

The package ElementTree [68] in Python is used for parsing XML files. For instance, in Listing 2, it is seen how easily the substation name and description can be extracted using ElementTree. At first, the substation tag is found using root.findall, where root is the entire XML file. The .get() method is applied to the entry that is found. This is used to retrieve the name and description (desc). For the excerpt of the SCD file shown in Listing 1, this function would return "AA1, Munich", as AA1 is the name attribute for the tag and Munich is the description. The same principle can be applied to an entire SCD file. It is possible to iterate through all the voltage level tags, for example, and, from there, iterate through all bay tags inside that voltage level and so forth. In Listing 3, it is seen how these tags are being iterated over. Using the `if bay.find(something) is not None:`, it is possible to check whether that something is included under the bay tag and access it if it exists. The .get() method is employed throughout.

```
def get_SS(root, namespaces):
    for component in root.findall('.//scl:Substation', namespaces):
        ss_name = component.get('name')
        ss_desc = component.get('desc')
    return f'{ss_name}, {ss_desc}'
```

Listing 2: Parsing of information from an SCD file. The name and description attribute from the "Substation" section is extracted.

```
for voltage_level in root.findall('.//scl:VoltageLevel', namespaces):
    for bay in voltage_level.findall('.//scl:Bay', namespaces):
        ce_data = []
        for ce in bay.findall('.//scl:ConductingEquipment', namespaces):
            name = ce.get('name')
            ce_type = ce.get('type')
        if bay.find('.//scl:ConductingEquipment', namespaces) is not None:
            bay_name = bay.get('name')
                bay_names_temp.add(bay_name)
                conducting_equipment_data[bay_name] = ce_data
```

Listing 3: Accessing nested information from an SCD file.

It is seen that accessing elements in the XML file using ElementTree is analogous to accessing a file on a computer. It works by specifying the path to that particular location. For instance, if one wants to access all LNs in the SCD, it can be done by looping through the hierarchical structure of it. In this case, it would be `VoltageLevel/Bay/LNode`, from where the various attributes can be accessed:

- the name of the IED, `iedName`
- the logical device instance, `ldInst`
  - i.e. whether it is a measurement, protection or control-related function
- The LN Class, `lnClass`
  - This could be PTOC, XCBR or CILO, for example
- A unique, numeric identifier, `lnInst`

This will be used to visualise the SCD file and cross-reference the signal addresses to it. It will be visualised using the tkinter package that Python provides [69]. This package offers a GUI toolkit.

3.3. The Test Script

With the prerequisites done, the test script can now be run. The most important parts are shown and explained in this section. The entire code can be found on GitHub via the link: `https://github.com/krusehansenjens/gen_test_case/tree/main`. An overview of the sequence of execution of the developed script is seen in Figure 3.3.

As the script relies on the signal addresses and test steps as inputs (see Table 3.2 and Table 3.1), these are kept in two different tabs of an Excel file for ease of editing and viewing. This builds upon the implementation of Tahincioglu, Schossig and Heimisson in [55].

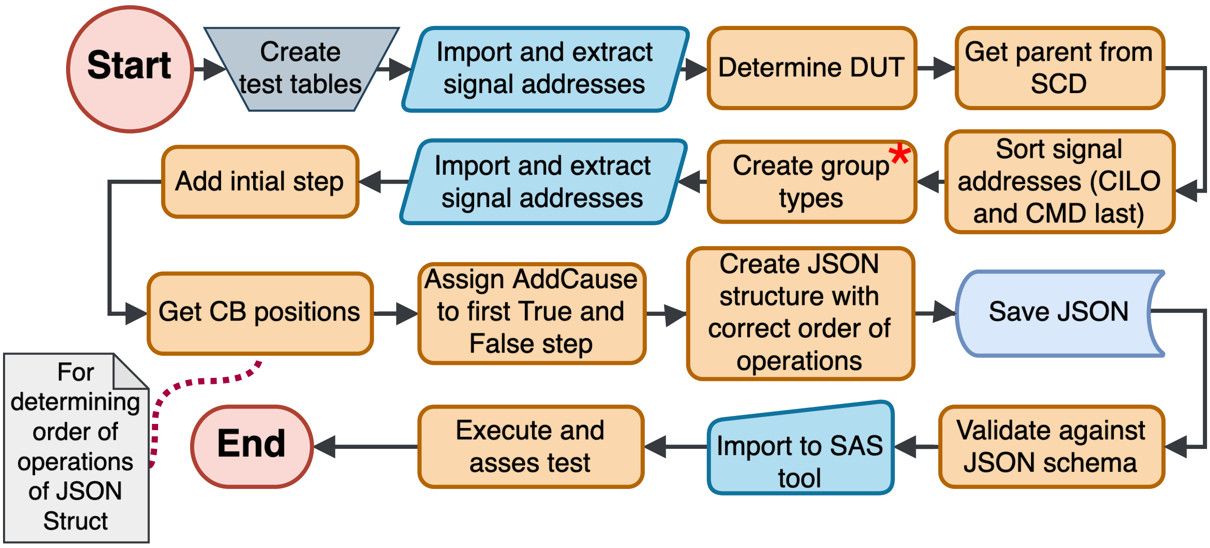


Figure 3.3.: Flowchart of the script generating the test file. The asterisk at "Create group types" signifies that this part is changed for the other version of the script [65].

The first step of the script is to retrieve the signal addresses from the Excel file. This is done by using the data analysis package Pandas from Python. This package allows for reading an Excel file and extracting rows and columns based on their indices. From here, it becomes possible to determine the DUT name by isolating the signal address which has the CILO LN in its name. After identifying this address, the resulting string is stripped such that only the DUT name remains. In the case of Table 3.2, `AA1D1Q01Q1QB1/CILO1.EnaCls` would first be found, and then "`/CILO1.EnaCls`" would be stripped from it, leaving only the name of the switchgear under test, `AA1D1Q01Q1QB1`.

When the address of the DUT is known, the parent, i.e. the IED that holds the DUT, can be found in a generic manner. This is shown in Listing 4. Here, the ElementTree [68] package of Python is used to go through the XML structure of the SCD file and access the IED section of the file. The .get() method is then applied to extract the names of all IEDs. As the DUT name has already been established, it is possible to iterate through the list of IEDs and compare them to the DUT. If the name of the IED is contained in the DUT, it must be the parent, which is then returned.

Next, the signal addresses are reordered to fit the JSON formatting. The goal is for the

```python
def get_parent(root, namespaces, DUT) -> Union[str, list]:
    ieds = []
    # Iterate through every IED
    for ied in root.findall('.//scl:IED', namespaces):
        # Retrieve the name of the IEDs and add them to a list
        ied_name = ied.get('name')
        ieds.append(ied_name)

    # Iterate through the IEDs and determine if the IED name is contained in
    ↪ the DUT name
    parent = next((ied for ied in ieds if ied in DUT))
    return parent, ieds
```

Listing 4: The function that returns a string containing the name of the parent.

assessment to be the penultimate operation and for the CMD providing the AddCause to be the last operation of any given test step. This is because these test steps are ordered and executed top-down, and all the position changes must be resolved before assessing the system state. The finished JSON formatting can be seen in Figure 3.4.

The reordering is shown in Listing 5. First, the imported list of signal addresses is reordered using the .sorting() method with the DUT name as a key. As False < True (0 < 1) in Python, the LNs containing the DUT will appear last in the resulting list. The list of addresses must be further rearranged to ensure that the assessment is in the penultimate position and not the AddCause command. This is done by retrieving the index of the address, which contains the string "CILO". The address with this index is then removed from the list using the .pop() method and reinserted at the second to last index, after which the function returns the list of successfully reordered addresses.

```python
def sort_signal_adresses(signal_addresses, dut) -> list:
    # Sort elements in list based on whether DUT is in the address
    # False < True in python, so dut will appear last as they will return true
    dut_last = sorted(signal_addresses, key=lambda address: dut in address)
    # get the index of the CILO LN
    index = [idx for idx, s in enumerate(dut_last) if 'CILO' in s][0]
    # Remove CILO from list and reinsert at second-to-last position, to fit
    ↪ with the JSON
    dut_last.insert(-1, dut_last.pop(index))
    sorted_addresses = dut_last
    return sorted_addresses
```

Listing 5: Sorting to place the CILO second to last for the assessment and the DUT position last for the command.

Following the sorting, the different group types are created. This is a branch of the JSON (see "signalGroups" in Figure 3.4) that serves to inform which signals belong to which group. The signal addresses are thus sorted into the "CONTROL", "ASSESS", or "COMMAND" group, where CILO and the DUT position will populate the last two groups, respectively, and the remaining switches will populate the "CONTROL" group. A sorting is again applied utilising the certainty that the string "CILO" and the DUT name will be in the name of the CILO signal address and that "CSWI" and the DUT name will be in the assessment signal. This is seen in Listing 6.

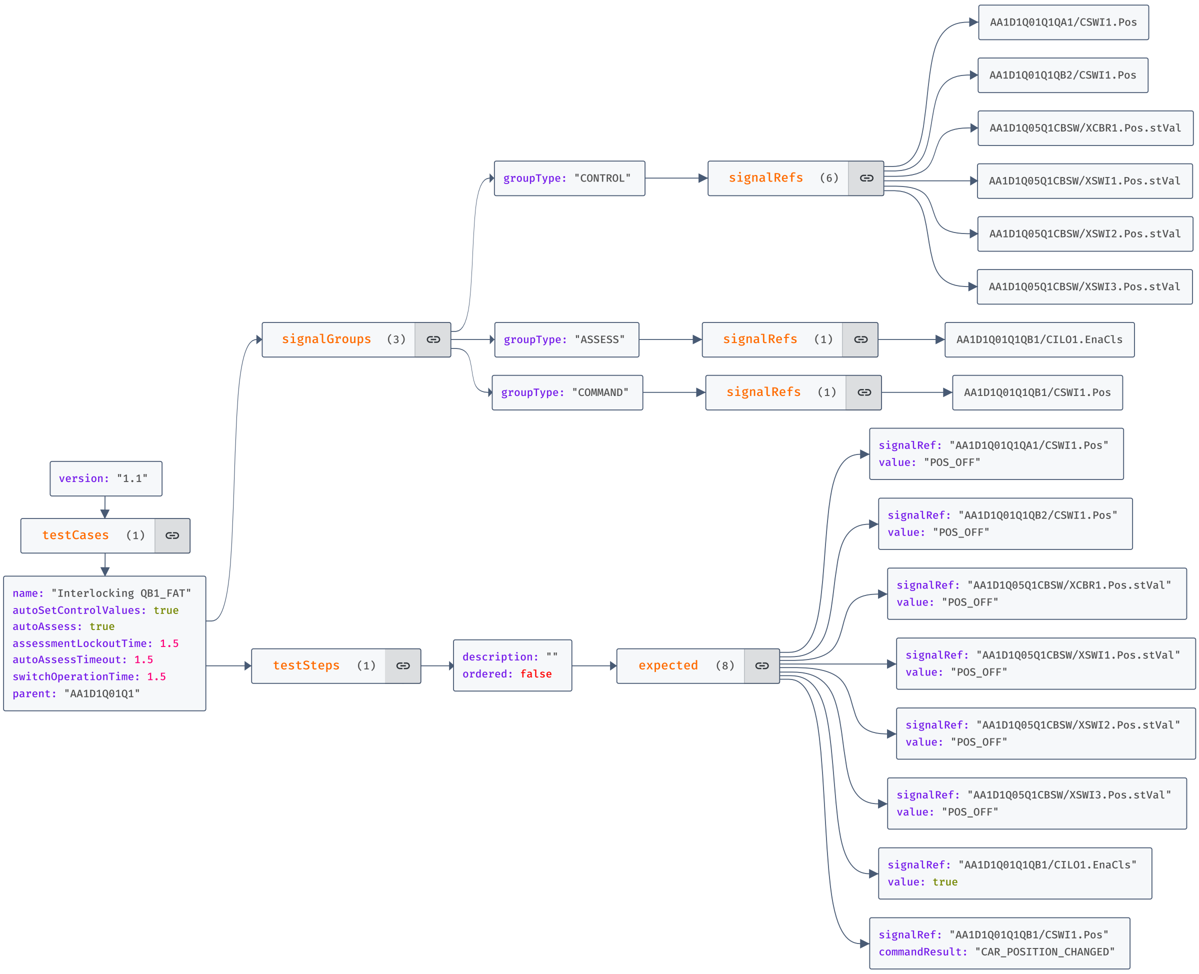


Figure 3.4.: Graphical representation of a JSON test file. Note that for compactness, only one of the eight test steps is shown [65].

```
group_types = {
# The controlled elements are all elements that are not the DUT
"CONTROL": [address for address in signal_addresses if DUT not in address],
# CILO is the assessment
"ASSESS": [address for address in signal_addresses if DUT in address and "CILO"
↪  in address],
# The DUT SWI is the command
"COMMAND": [address for address in signal_addresses if DUT in address and
↪  "CSWI" in address]
        }
```

Listing 6: Sorting of group types.

An alternative for the sorting of group types has also been created, which is indicated on the script overview diagram in Figure 3.3 by the red asterisk. This version of the script demands an extra column in the table of signal addresses. This extra column at the end of the table would contain the group type of each of the signal addresses. Concretely, this means that after the CILO signal of the DUT, the entry in this new column would read ”Assess”, and after the signal address pointing to the position of the DUT, it would read ”Command”, and finally, after the rest of the addresses it would read ”Control”. An example of this is seen in Table 3.3. While the original implementation is effective, it does require a standardised naming convention, where the DUT name is in the addresses. To handle cases that do not follow the naming convention in the original script, a flag, ”test_mode”, was created that allowed for hard coding the dictionary of group types. As hard coding is generally a bad practice the alternative method was developed instead to handle such cases. Although following the naming convention is best practice, the alternative method, where the extra column is added, gives more leniency in terms of naming without requiring a lot of extra effort, which may be desired. This is merely one possibility of modifying the script to one's liking.

The key difference of the alternative method is shown in Listing 7. Here, the column of group types is imported in addition to the signal addresses. An empty dictionary is then created, to which the names of the signal groups will be added as keys, and the signal addresses will be added as values under the corresponding key.

Table 3.3.: Expanded version of signal addresses including the group type that each individual signal address belongs to.

IED Name	Signal Address	Group Type
AA1D1Q01Q1	AA1D1Q01Q1QB1/CILO1.EnaCls	**Assess**
AA1D1Q01Q1	AA1D1Q01Q1QB1/CSWI1.Pos	**Command**
AA1D1Q01Q1	AA1D1Q01Q1QA1/CSWI1.Pos	**Control**
AA1D1Q01Q1	AA1D1Q01Q1QB2/CSWI1.Pos	**Control**
AA1D1Q05Q1	AA1D1Q05Q1CBSW/XCBR1.Pos.stVal	**Control**
AA1D1Q05Q1	AA1D1Q05Q1CBSW/XSWI1.Pos.stVal	**Control**
AA1D1Q05Q1	AA1D1Q05Q1CBSW/XSWI2.Pos.stVal	**Control**
AA1D1Q05Q1	AA1D1Q05Q1CBSW/XSWI3.Pos.stVal	**Control**

Next, the test steps are extracted from the Excel file. Further information is then harnessed from these test steps. These pieces of information are the various switch positions in each step, the total number of steps, as well as the expected assessments of each test step. The script expands the test steps given in the input file by adding an initial step that will be executed prior to the test sequence described by the input table. This is a step in which all switches are set to the open position. This is done as it will ensure that no unpremeditated interlockings will be encountered during the execution of the test. If one, for example, considers that the earthing switch of the substation system was closed prior to the execution of the test, it becomes clear that this could cause a problem if the next step involves closing the DUT, for example.

After having read both the switch positions and the signal addresses it is possible to determine the positions of the CBs in every step. These addresses can be determined, as the convention is to use QA followed by an integer (such as QA1) for the switchgear name and XCBR for the LN. The states of each of the two CBs are then found by retrieving the switch positions corresponding to the CB signal addresses, such that CB states can be compared between steps. This is important since if the CBs are opening, then these should operate before the disconnectors, and conversely, if the CBs are closing, the disconnectors should operate first. This is because the disconnectors are not rated to make or break the current. How extracting the signal addresses of the circuit

```
    # Extract the signal addresses & group types from the Excel file
def get_signal_addresses(excel_file: str, signal_addresses_sheet: str) ->
↪   pd.DataFrame:
    signal_addresses_tab = pd.read_excel(
        excel_file, sheet_name=signal_addresses_sheet, header=None)
    # Extract the signal addresses from the second column
    signal_addresses = signal_addresses_tab.iloc[1:, 1:2].values.flatten()
    # Extract the group types from the third column
    group_types = signal_addresses_tab.iloc[1:, 2:3].values.flatten()

    # Create an empty dict to assign pairs of group type and signal adresses
    group_type_dict = {}
    # Group up the signal adresses and group types
    for key, value in zip(group_types, signal_addresses):
        # If the group type is already in the new dictionary add the
        ↪   corresponding value to it
        if key.upper() in group_type_dict:
            group_type_dict[key.upper()].append(value)
        # If they key is not in the dictionary already add it as a new key
        ↪   with the corresponding value
        else:
            group_type_dict[key.upper()] = [value]
    return signal_addresses, group_type_dict
```

Listing 7: Alternative way of getting the group types and pairing them up with the corresponding addresses.

```
# The circuit breakers have the signal addresses that contain either QA+any
↪   digit or XCBR
circuit_breakers = [x for x in switch_positions.index if re.match(r"QA\d", x)
↪   or ("XCBR" in x)]

[...]

# Define scenarios, which affect the order of operations
cb_closing = (bay1_cb_states[step] == "POS_ON") & (
    bay1_cb_states[step-1] == "POS_OFF")
cb_opening = (bay1_cb_states[step] == "POS_OFF") & (
    bay1_cb_states[step-1] == "POS_ON")
cb_unchanged = bay1_cb_states[step] == bay1_cb_states[step-1]
```

Listing 8: The part of the code responsible for monitoring state changes of CBs across steps.

breakers based on the string patterns and how defining the CB positional changes is done can be seen in Listing 8. Here, the indices of the data frame "switch_positions" are the signal addresses.

```
def sort_switch_order(LNs_signal_dict: dict, cb_direction: str) -> dict:
    """Sort the dictionary based on the direction of the circuit breaker."""
    key = list(LNs_signal_dict.keys()
               )  # Get out keys (addresses) from the dictionary

    # If cb is opening between steps
    if cb_direction == ('opening' or 'opening'.upper()):

        # Sort the keys, excluding the last two. Here if XSWI is in the key it
        ↪  will be sorted last as true=1 and false=0
        sorted_keys = sorted(key[:-2], key=lambda x: ('XSWI' in x, x))

    elif cb_direction == ('closing' or 'closing'.upper()):
        # This sorting will put CB last
        sorted_keys = sorted(key[:-2], key=lambda x: ('XSWI' not in x, x))
    # Add the last two keys to the sorted keys
    sorted_keys += key[-2:]

    # Return a new dictionary with the sorted keys
    new_dict = {key: LNs_signal_dict[key] for key in sorted_keys}
    return new_dict
```

Listing 9: The function used for sorting the switch sequence based on the state changes of circuit breakers.

Before reorganising all the acquired data in a JSON-compatible format, the expected AddCauses must first be assigned to each step. The goal is to verify that the AddCause works for one step where the CILO release signal allows for the operation of the DUT and one where it does not. Therefore, a non-default AddCause is only tied to the first step that evaluates to true and the first step that evaluates to false. "CAR_POSITION_CHANGED" will be the command associated with the first true step, and "CAR_BLOCKED_BY_INTERLOCKING" will be associated with the first false step. The latter signifies that a control action is blocked due to interlocking of switching devices (in CILO attribute EnaOpn.stVal="FALSE" or EnaCls.stVal="FALSE") [39]. The "CAR_POSITION_CHANGED" is an addition made by Omicron which can be used with StationScout. This allows for positive feedback instead of opting for the default positive message specified in IEC 61850-7-2 "None", which means "Control action successfully executed". For the remainder of the steps, it is sufficient for the command to be "CAR_NO_OPERATION", as the other commands have been verified at this point [65].

With the knowledge of when a CB is opening or closing, the order of operations in those steps can be programmed. The approach is seen in Listing 9. As the assessment and commands are always the second to last and the last action, these do not need to be sorted based on the CB operation. Therefore, the last two indices of the sorted keys are omitted, as indicated by `key[:-2]`. The fact that Python can sort True and False values is once again used. A tuple is created that contains either True or False in the first position depending on whether or not the n^{th} element contains the string "XSWI". If it does, then the first element of the tuple is true, and the second will be the signal address. This will be applied to the entire list of signal addresses. Eventually, within the sorted() method, a list of tuples containing a boolean value and a signal address will exist. These are then sorted in numerically ascending order, such that the first signal addresses will be those of the disconnectors and vice versa. The same sorting key, but inverted, is applied when the CB is closing. Having the states ordered sequentially works as the test file will be executed top-down. This means that for a step where the CBs are closing, it is ensured that all disconnectors are operated prior to the CBs by sorting the signal addresses from the ”Control” group such that the CBs appear last.

Finally, the JSON structure can be created, and the different steps can be filled out with the correct information in the right sequence. This is seen in Listing 10. In the beginning, a variable is created which will have its contents expanded as the function loops through the different information contained in the script at this point. This variable is what will be converted to a JSON file at the end.

The interesting part is the ”testCases” list created within this variable. It is initialised as an empty list but immediately filled with a dictionary that contains information such as the test name, parent, and various test parameters, such as whether or not the control values should be set automatically. This and the auto assessment are naturally set to true, as the overarching goal is to automate the process.

Then, two further lists are initialised: ”signalGroups” and ”testSteps.” The first is filled by iterating through the items of ”group_types,” which stores the group types ”control,” ”assess,” and ”command” and the associated signal addresses. The list of ”testSteps” is filled by iterating through the total number of test steps, including the initial step added. Here, as described previously, it is checked whether the CB status changes and how. After knowing if the CB is opening or closing in the current step, it is checked whether the current step is the first step. If it is, no consideration of the CB states must be given as everything should be opened. In this initial step, the expected values are simply retrieved and added to the ”expected” entry of the dictionary within ”testSteps”, and their associated signal addresses are added under the key ”signalRef”. This structure is more easily comprehensible when seen in conjunction with Figure 3.4.

If the script arrives at a step where the CBs are closing, the switch order is sorted accordingly as described prior to this, and the rest of the procedure is identical to the case for the initial step. Listing 10 only shows an excerpt of the function, but it naturally continues from what is shown by adding the data to the same initial variable ”ilo_FAT” under the ”cb_closing” conditional. It then continues to handle the cases where the CBs are opening or where the states remain unchanged. Finally, the variable containing this data structure is returned and can be saved as a JSON file.

```python
def create_FAT_json(version: float, test_name: str, dut_name: str, group_types:
↪   dict) -> dict:
    """Create JSON Structure"""
    # Create version
    ilo_FAT = {"version": str(version), "testCases": []}

    # Create parameters for test
    ilo_FAT['testCases'].append({"name": str(test_name),
    "autoSetControlValues": True,
    "autoAssess": True,
    "assessmentLockoutTime": 1.5,
    "autoAssessTimeout": 1.5,
    "switchOperationTime": 1.5,
    "parent": str(dut_name),
    "signalGroups": [],
    "testSteps": []
```

```
})

# Create CONTROL, ASSESS and COMMAND signal groups and associated LNs
for key, value in group_types.items():
    ilo_FAT['testCases'][0]['signalGroups'].append(
        {"groupType": key, "signalRefs": value})

# Create test steps
# As everything in the JSON have now been filled out except the testSteps,
↪  we can now fill out the those
for step in range(num_test_steps):

    # Define scenarios, which affect the order of operations
    cb_closing = (bay1_cb_states[step] == "POS_ON") & (
        bay1_cb_states[step-1] == "POS_OFF")
    cb_opening = (bay1_cb_states[step] == "POS_OFF") & (
        bay1_cb_states[step-1] == "POS_ON")
    cb_unchanged = bay1_cb_states[step] == bay1_cb_states[step-1]

    # In the first step, order is not important, as there is no previous
    ↪  step
    if step == 0:
        step0_vals = get_expected_vals(LNs_signal)

        # We are appending the data to the testSteps list in the JSON.
        ↪  Description stays empty.
        # Everything is ordered, except for the first step
        # The expected values are added to the JSON. The last value is a
        ↪  commandResult, the second to last is the assessment
        ilo_FAT['testCases'][0]['testSteps'].append({"description": "",
        "ordered": (False if step <= 0 else True),
        "expected": [{"signalRef": ln, "value": step0_vals[step][j]} if j
        ↪  != len(signal_addresses)-1
        else {"signalRef": ln, "commandResult": step0_vals[step][j]}
        for j, ln in enumerate(signal_addresses)]})

    # If the circuit breaker is closing, the order of the switching
    ↪  operations, should ensure that disconnectors close before the
    ↪  circuit breaker
    elif cb_closing:
        print(f'{step} CLOSING')
        # the switching signals and the LNs are sorted appropriately
        LNs_signals_closing = sort_switch_order(LNs_signal, 'closing')
        # The dictionary is "transposed" to get all expected values for
        ↪  the current step for the different LNs
        closing_vals = get_expected_vals(LNs_signals_closing)

[...]
```

Listing 10: An excerpt of the part of the code responsible for creating the JSON structure and assigning the positions, expected assessments and commands to the test file. The function continues with code for handling cases where the circuit breaker opens, or the status remains unchanged, but this is not shown for brevity.

After having saved the JSON the validity of the file is checked. This is done by comparing it to a JSON schema tailor-made for this JSON format. It follows the same structure and dictates the data types that can be entered under the different parameters and, therefore, gives a good impression of how a finished test case would look. For example, it dictates that the version at the top of the JSON should be of the type string and that "autoAssess" should be a boolean value. It also specifies which parameters are required in the file. The JSON schema developed for this script is seen in Listing 11. Finally, this validation schema works irrespective of the number of test steps in the test case.

```
test_case_schema = {
    # Most recent json schema specification
    "$schema": "http://json-schema.org/draft/2020-12/schema",
    "type": "object",
    "properties": {
        "version": {
            "type": "string"
        },
        "testCases": {
            "type": "array",
            "items": {
                "type": "object",
                "properties": {
                    "name": {
                        "type": "string"
                    },
                    "autoSetControlValues": {
                        "type": "boolean"
                    },
                    "autoAssess": {
                        "type": "boolean"
                    },
                    "assessmentLockoutTime": {
                        "type": "number"
                    },
                    "autoAssessTimeout": {
                        "type": "number"
                    },
                    "switchOperationTime": {
                        "type": "number"
                    },
                    "parent": {
                        "type": "string"
                    },
                    "signalGroups": {
                        "type": "array",
                        "items": {
                            "type": "object",
                            "properties": {
```

```
                        "groupType": {
                            "type": "string",
                            "enum": ["CONTROL", "ASSESS", "COMMAND"]
                        },
                        "signalRefs": {
                            "type": "array",
                            "items": {
                                "type": "string"
                            }
                        }
                    },
                    "required": ["groupType", "signalRefs"]
                }
            },
            "testSteps": {
                "type": "array",
                "items": {
                    "type": "object",
                    "properties": {
                        "description": {
                            "type": "string"
                        },
                        "ordered": {
                            "type": "boolean"
                        },
                        "expected": {
                            "type": "array",
                            "items": {
                                "type": "object",
                                "properties": {
                                    "signalRef": {
                                        "type": "string"
                                    },
                                    "value": {
                                        "type": ["string", "boolean"]
                                    },
                                    "commandResult": {
                                        "type": "string"
                                    }
                                },
                                "required": ["signalRef"],
                                "oneOf": [
                                    {
                                        "required": ["value"]
                                    },
                                    {
                                        "required": ["commandResult"]
                                    }
                                ]
                            }
                        }
```

```
                        },
                        "required": ["ordered", "expected"]
                    }
                }
            },
            "required": [
                "name",
                "autoSetControlValues",
                "autoAssess",
                "assessmentLockoutTime",
                "autoAssessTimeout",
                "switchOperationTime",
                "parent",
                "signalGroups",
                "testSteps"
            ]
        }
    }
},
"required": ["version", "testCases"]
}
```

Listing 11: JSON schema used for validating the test case.

3.4. StationScout

The test cases in the JSON file format can be imported and run in StationScout [64], which was shown in [55]. This also applies to the test cases generated using the developed script. StationScout is a SAS test software developed by Omicron [64] and offers an intuitive GUI that allows the test operator to visualise an SCD file imported into the software.

This overview is seen in Figure 3.5. The SCD file shown here is different from the one used in the laboratory test in this thesis but serves to illustrate the capabilities of StationScout. Here two voltages levels of 320 kV and 33 kV, respectively, are seen. There are five bays on the high voltage level and two on the lower voltage level. Furthermore, it is seen that all bays have at least one IED. These are shown on the left side of each bay. The two IEDs in the low voltage levels are offline, as indicated by the disconnected cables across the IED symbol. Four of the IEDs and a disconnector in the high voltage level display a yellow warning triangle. This can signify that an expected GOOSE signal is not received. Finally, the quotation marks in a green circle over an IED imply that this IED is currently being simulated.

The communication paths are also shown when selecting an IED, as seen here. Q1 of Q03 is selected, which reveals GOOSE messages (indicated in purple) going to other IEDs and a busbar protection IED. A sampled value stream going from Q2 of Q05 to the highlighted IED is also seen. These SV signals are colour-coded red. Finally, the green signals going to the HMI and the RTU/Gateway are IEC 61850 reports, which can be used for data logging, for example.

If available in the SCD, the switching devices will also be displayed, as seen in this example.

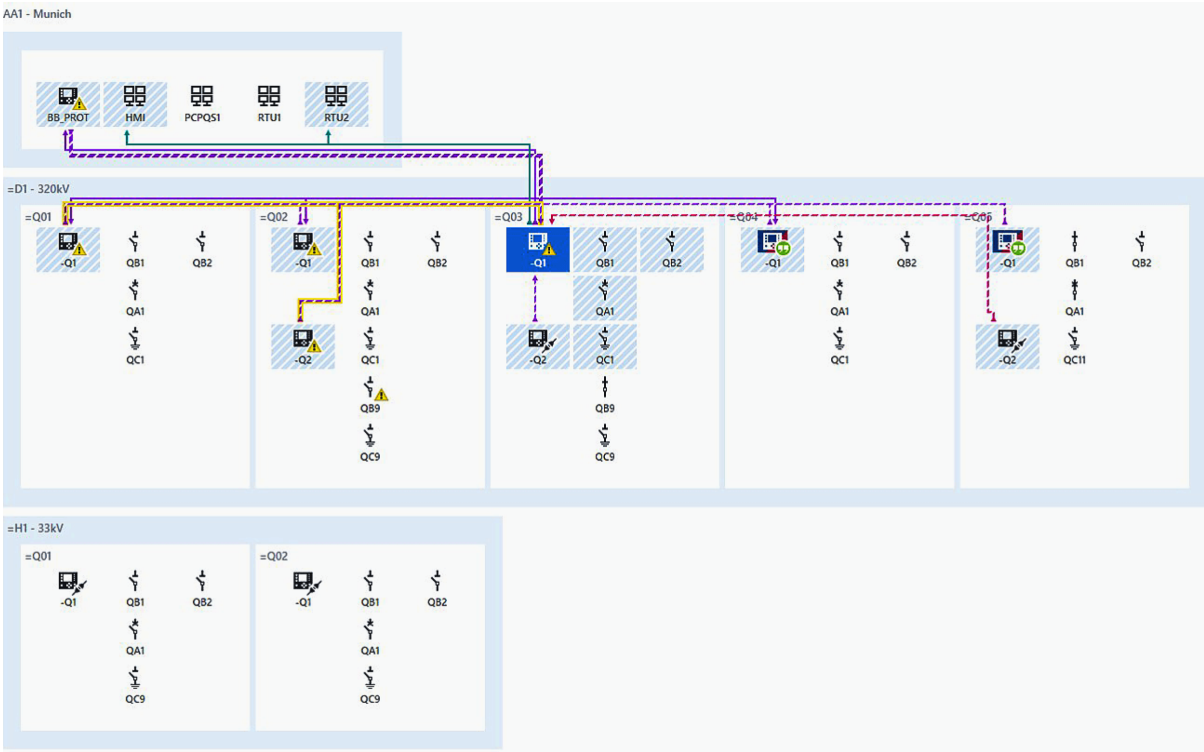


Figure 3.5.: Overview of a substation system as seen in StationScout [70].

3.4.1. Test Cases

StationScout is capable of executing logic tests, meaning it can compare an expected value to an observed one. This is useful for interlocking testing. The LNs used for controlling the switches (CSWI and XSWI) and the interlocking assessment LN, CILO, are available in the SCD, so it is possible to manipulate and read their statuses.

Figure 3.6 illustrates the creation of an interlocking test case in StationScout. The three group types, control, assessment, and command, are seen in Table 3.3 as "Control", "Signal assessment", and "Command assessment" and are in red, yellow, and blue, respectively. In this first part of the interlocking test, these groups must be populated with signals pointing to an address that controls a switch position, then a CILO LN that provides signal assessment and finally, the address to the position of the DUT in the command assessment group.

After the selection of the relevant signal addresses the individual test steps can be built. This is where the contents of Table 3.1 can be implemented. A control signal is configured to be issued to each of the switches selected for the control group. Then, an expected value is chosen for the CILO LN of the DUT, and subsequently, an expected command from the positional LN of the DUT is selected. This way of manually configuring test steps in StationScout is shown in Figure 3.7.

The created test case can be executed automatically or using manual interaction. In this case, the switch positions are changed manually from the test case interface in StationScout one at a time until they match the expected positions for that particular test step. If the test is designed to be run automatically, the order of operations must also be specified. This is done by assigning a number from 1 to n to each of the n switches controlled in the control stage of the test. When the test is executed, the assessment and command will also be automatically obtained. This automatic workflow is illustrated in Figure 3.8. Subsequent to having executed a test case a

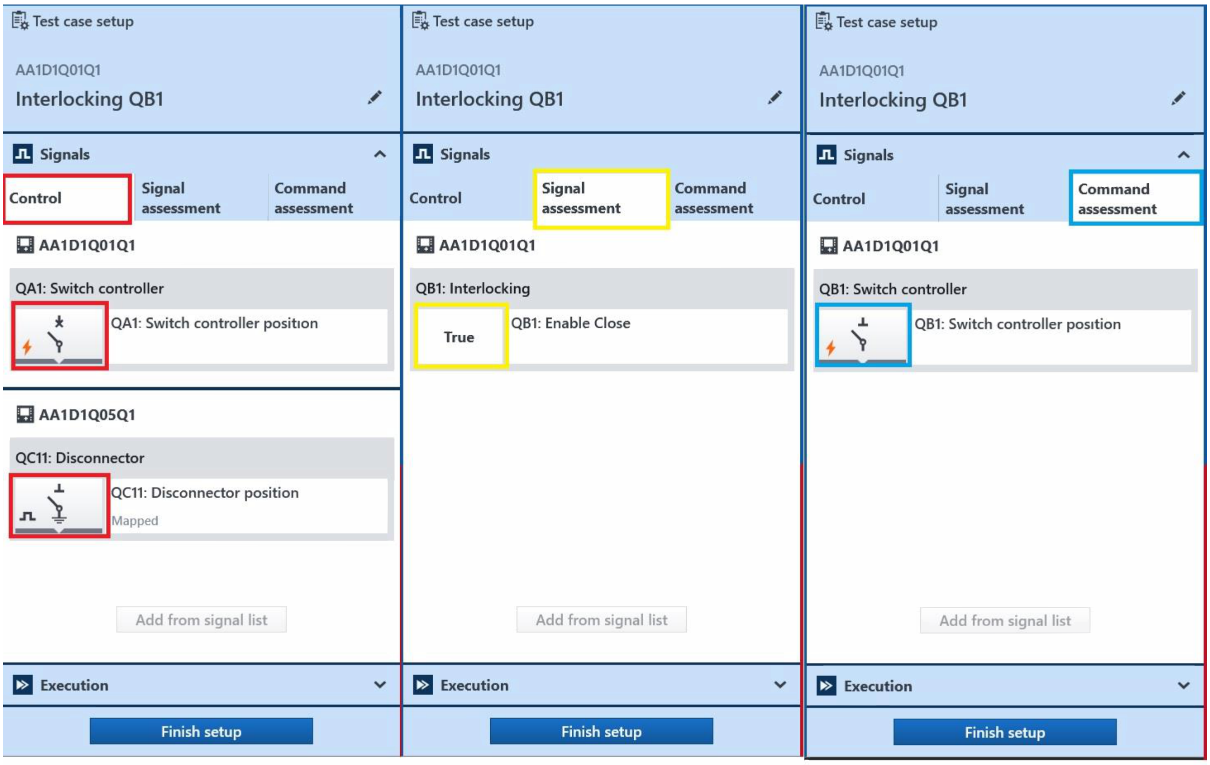


Figure 3.6.: Example of a test case in StationScout showing the different parts that make up a complete test step. The figure is from [55].

Word containing the test results can be exported. This file shows whether or not the test has passed, as well as the expected and actual values for each step in the test. It also provides time stamps for each operation undergone in the test.

The process of creating the test cases by navigating to each of the required LNs, adding them to the signal list, and selecting expected values for them for each step is part of the process eliminated by the developed script. This functionality also already exists if using the VBA macro developed by Tahincioglu, Schossig and Heimisson [55]. The VBA macro does not, however, take care of the order of operations automatically, and that must, therefore, be included in the data provided by the user as seen in Figure 2.16. Nor does it have any schema validation or interaction with the associated SCD file. The automated workflow in StationScout should, therefore, improve for interlocking testing when using the developed script.

3.4.2. VBX1

StationScout is meant to be used with MBX2 [71], a physical test set that enables the simulation of devices, facilitates communication and provides an interface with a high level of cybersecurity. The licensing of StationScout is based on the purchase of the MBX2, but during the collaboration with Omicron, access was granted to StationScout and its virtual equivalent VBX1 [72]. The VBX1 can be run through a virtual machine (VM) by installing the OVA (Open Virtual Appliance) file provided by Omicron on the VM.

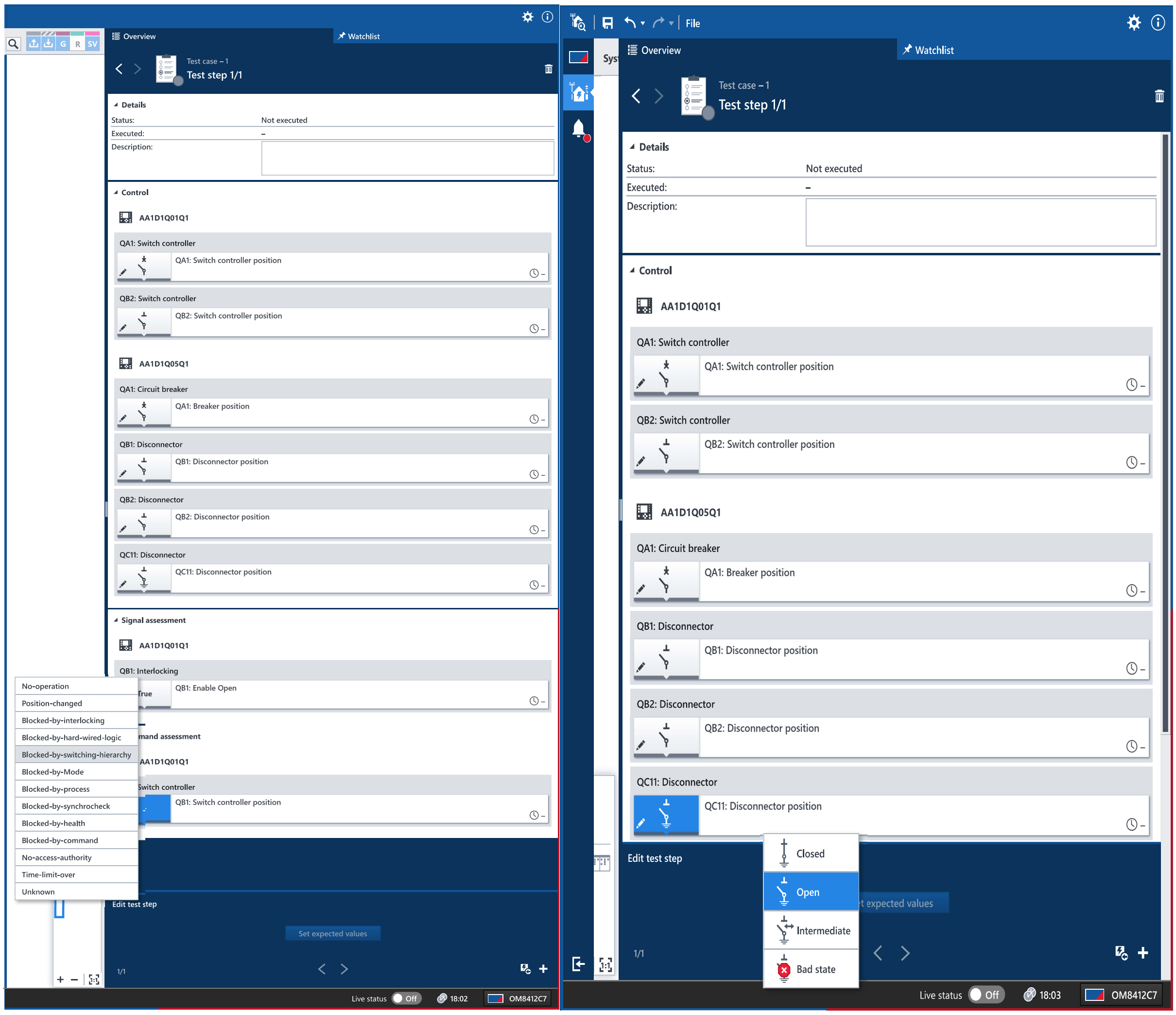


Figure 3.7.: Manual configuration of a single test step in StationScout.
Left: Configuration of switch positions, the possible states are shown in the open dialogue box at the bottom.
Right: At the bottom of it is seen that the assessment from the LN CILO is set to true, and the dialogue box for the command assessment is opened.

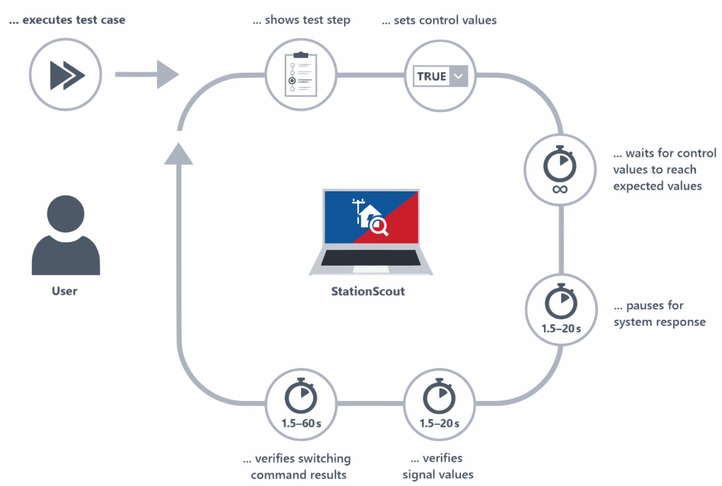


Figure 3.8.: Automatic test case workflow in StationScout. The figure is from [55].

Chapter 4

Laboratory Test

4.1. Equipment and Software

This study's laboratory setup and equipment used for testing interlockings in digital substations are crucial aspects. As the JSON test case is tailor-made for StationScout [64], the SAS test tool developed by Omicron, this software is naturally used. The VBX1 version used requires interfacing through a VM to access the equipment connected to their server. VMWare Workstation Player 17 [73] was used for this purpose. Furthermore, IEDScout [74], another Omicron tool used for interacting on an IED level instead of on a substation level, was used. Moreover, a piece of binary input/output (I/O) equipment is used in one test. This is the CMC356 from OMICRON [75], which can be seen in Figure 4.5.

To configure the IEDs, the IED Configuration Tool PCM600 [76] is used. This tool was developed by ABB and is used specifically to configure ABB IEDs. This tool can also generate an SCD file. The SCD file generated by PCM600 is then further configured using an open-source system configuration tool called OpenSCD [77]. This allows for ease of implementing conducting equipment that will show up in StationScout. Although this step is not strictly necessary, it is relatively low effort and adds to the completeness of the SCD file.

The IED used was a RED670 from ABB. This is a line differential protection IED that can be used to protect, control, and monitor overhead lines and cables. It supports IEC 61850 communication protocols and is customisable through PCM600 [78].

Although IEDScout is useful for viewing the communication going through the IEDs, WireShark [79] was also used to probe the communication and to discover the IP address needed to access the virtual version of StationScout.

4.2. IED Configuration

As the RED670 from ABB is used, the associated IED Configuration Tool, PCM600, was used to configure it. The HW configuration for a RED670 from the laboratory is seen in Appendix B. Both the IED of the outgoing feeder bay and the IED of the bus coupler bay will be configured. In the IED Configuration Tool, all the relevant logical nodes and the communication between IEDs and the logic must be configured. The configuration of both the feeder and the bus coupler IEDs will be described here. The application configuration for the IED in the feeder bay (Q01) is seen in Section D of the appendix, and the application configuration for the bus coupler bay (Q05) is seen in Section E of the appendix.

At first, Q01 was configured. This was done by adding three function blocks, one for each piece of switchgear that is being interacted with in the feeder bay. Two function blocks of the type SXSWI were selected for the disconnectors and once SXCBR for the CB. A binary input signal was assigned to the inputs of each of these blocks through a binary input signal matrix, such that the position change from the binary input could be split up into a regular signal and an inverted signal for the ”POSOPEN” and ”POSCLOSE” inputs. This allows for reading the position of the switch afterwards via the OPENPOS/CLOSEPOS variables that were created. These signals are later used for the interlocking scheme.

In the bus coupler IED, a binary input was linked to both disconnectors, the earthing switches (controlled by the same command), and the CB. Additionally, GOOSE commands were created to control the feeder IED from the bus coupler IED and thus allow for remote testing without using a physical test set. This was done by configuring three extra switches in the bus coupler IED. These are seen in Section E of the appendix under the blue label reading ”BINARY GOOSE COMMAND TO Q01”. The Q01 IED is configured to receive these GOOSE commands when it does, the associated output contact will be excited. This output contact is directly connected to another input contact of the same IED which can, therefore, be energised through here. These input contacts are what control the switch position signals of the feeder IED. This is illustrated on the left side of Figure 4.3, and the implementation in PCM600 is seen in Figure 4.1. In the latter, the red dashed lines represent the physical connection from the IED’s outputs to its inputs.

This configuration scheme allows for remote operation, as everything can be controlled via StationScout and VBX1. Remote control could also be possible using a physical I/O box, which would not require wiring the outputs of the IED with the DUT to its inputs but would require another piece of equipment that is connected to the inputs of the DUT.

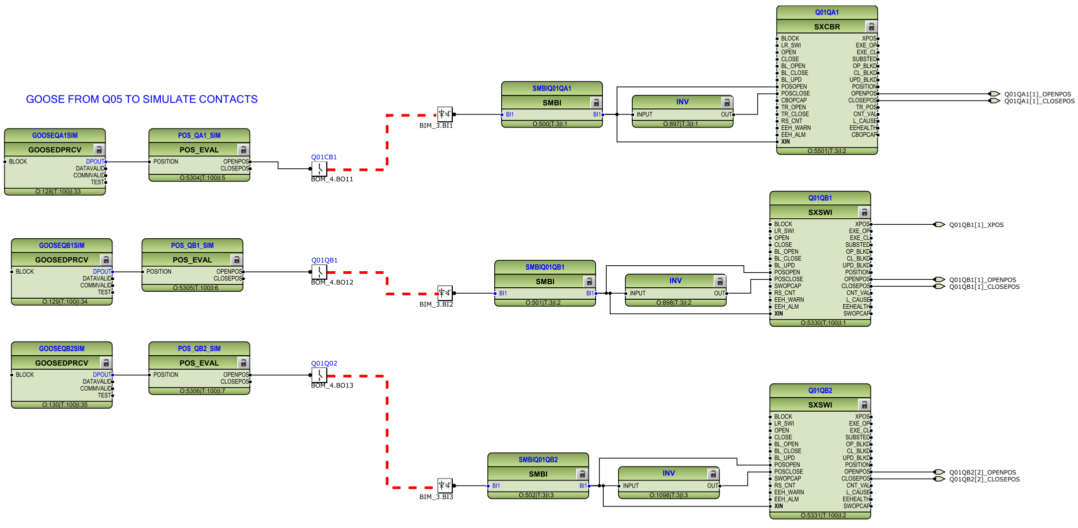


Figure 4.1.: The configuration that allows for the bus coupler IED to effectively be used as a virtual I/O box. When one of the three GOOSE commands shown here is received from the bus coupler IED, the associated binary output will be excited. As the binary output contacts are routed externally (indicated by the red, dashed lines) to the physical IED’s own inputs, the GOOSE command from the virtual IED can be used to control the physical one.

With all the position variables in place, the interlocking logic of the DUT could be configured. As PCM600 contains logic blocks, the logic could easily be implemented. Here, this is implemented as seen in Equation 4.1, where the default position is closed, Q05 denotes the bus coupler bay, Q01 denotes the feeder bay, QA is used for CBs, QB is used for disconnectors, and QC denotes the earthing switch. This equation is represented by the circuit diagram seen in Figure 3.2. The output of this equation is then fed to the SCILO block that controls the interlocking of a device, in this case, the feeder disconnector, Q01QB1, as that is the DUT. This interlocking is controlled via the "CILO enable open" and "CILO enable close" inputs to the SCSWI block.

$$((\text{Q05QA1} \cdot \text{Q05QB1} \cdot \text{Q05QB2}) + \overline{\text{Q01QB2}}) \cdot \overline{\text{Q01QA1}} \cdot \overline{\text{Q05QC11}} = \text{CILO Signal} \qquad (4.1)$$

A control block for the DUT is inserted and fed the position of the DUT switch as well as the enable close and enable open signals from the CILO LN, which will determine whether it can operate. This and the interlocking logic implementation in PCM600 are seen in Figure 4.2.

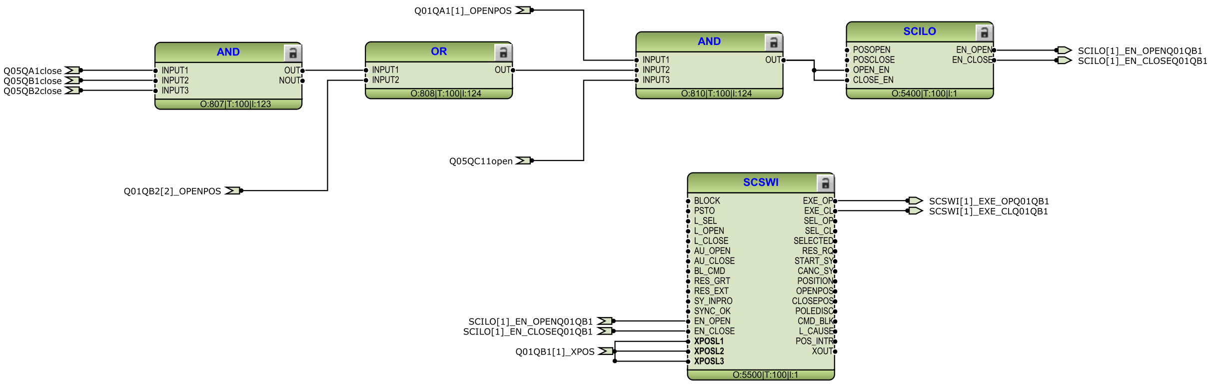


Figure 4.2.: Interlocking logic as implemented in PCM600. The logic is described by Equation 4.1 and Figure 3.2.

Finally, three LEDs were configured for the feeder IED to monitor the position of the switching equipment in this bay. The IED would light red if the switches were closed and green if they were open.

4.3. Test Setup

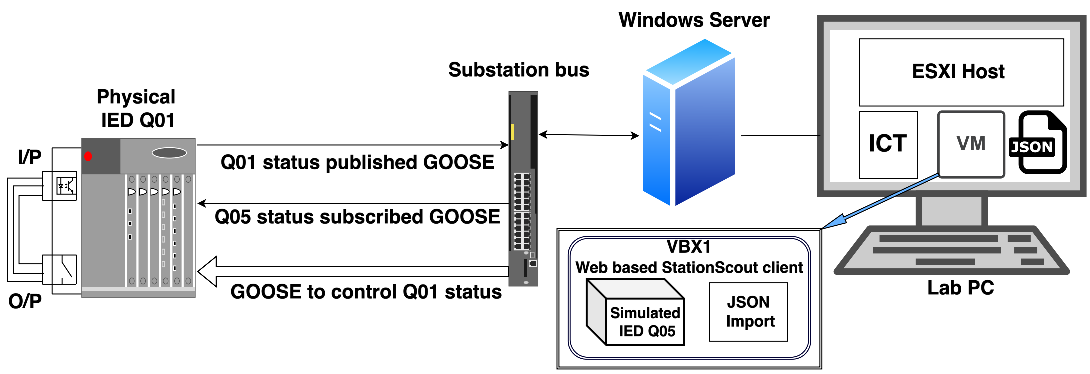


Figure 4.3.: Illustration of the laboratory setup. Q05 is simulated within the virtual edition of StationScout through the network, and the physical IED is connected to the substation bus to ensure communication between the two [65].

The test setup used for validating the generated test cases is shown graphically in Figure 4.3. It is seen how the laboratory PC runs a virtual machine that hosts a web server on which StationScout is run. This StationScout version utilises the virtual test set VBX1. Here, the SCD file produced from the IED configurations is uploaded, the bus coupler IED, Q05, is simulated, and a connection is established to the physical IED via StationScout.

The simulated IED communicates with the physical IED through the substation bus via the Windows server. This allows it to publish GOOSE signals that are received by the physical IED and used to excite its output contacts, which are routed to its own input contacts and used to trigger a positional change in the switches controlled by Q01. The external connections of the physical IED are seen on the left side of Figure 4.3. This is all possible due to the aforementioned IED configuration, where the simulated IED is able to act both as the bus coupler IED itself and as a virtual I/O box that can control the physical IED. This is illustrated in Figure 4.1. As the virtual IED was used for the control of the physical IED, the majority of the signal addresses used in this test case were from the virtual IED, as seen in Table 4.1. This table illustrates that signals from the bay five IED are used to control or read positions and statuses from the bay one IED.

Table 4.1.: Signal addresses used for control, assessment and reading the AddCause in the laboratory test. Note that the Q05 bay IED sends all the signals except for the interlocking signal, CILO, due to the hardware configuration described in chapter 4 [65].

Signal Addresses	**What it controls/reads**	
Q01CTRL/SCILO1.EnaCls.stVal	Q01QB1	CILO
Q05CTRL/SXSWI4.Pos.stVal	Q01QB1	AddCause
Q05CTRL/SXCBR2.Pos.stVal	Q01QA1	Position
Q05CTRL/SXSWI5.Pos.stVal	Q01QB2	Position
Q05CTRL/SXCBR1.Pos.stVal	Q05QA1	Position
Q05CTRL/SXSWI1.Pos.stVal	Q05QB1	Position
Q05CTRL/SXSWI2.Pos.stVal	Q05QB2	Position
Q05CTRL/SXSWI3.Pos.stVal	Q05QC11	Position

The physical setup is seen in Figure 4.4. The laboratory computer running the virtual version of the StationScout-hosted web server through the virtual machine is seen on the left, and the two IEDs used are seen at the top of the left rack.

Before testing the complete setup in an automated fashion with the generated test files, the configuration of the IEDs and the hardware connections were verified using the CMC356 seen in Figure 4.5. By using TestUniverse, which is the software used for controlling the CMC356, the position signals sent by the IEDs could be changed, and the states of the positions and resulting interlocking could be verified. This was done by sequentially applying a 48 VDC signal to the binary inputs of the IEDs. The changes could then be observed in PCM600 or IEDScout, and the implementation verified. The verification of one of the test steps is seen in Section F of the appendix.

After having verified the implementation, the generated test case was imported into StationScout, and a physical connection to Q01 and a virtual connection to Q05 were established. When the test case was executed, the signals transmitted from Q05 successfully energised the output contacts of Q01, which were fed to its own inputs and thus, the positions of the switches changed as intended.

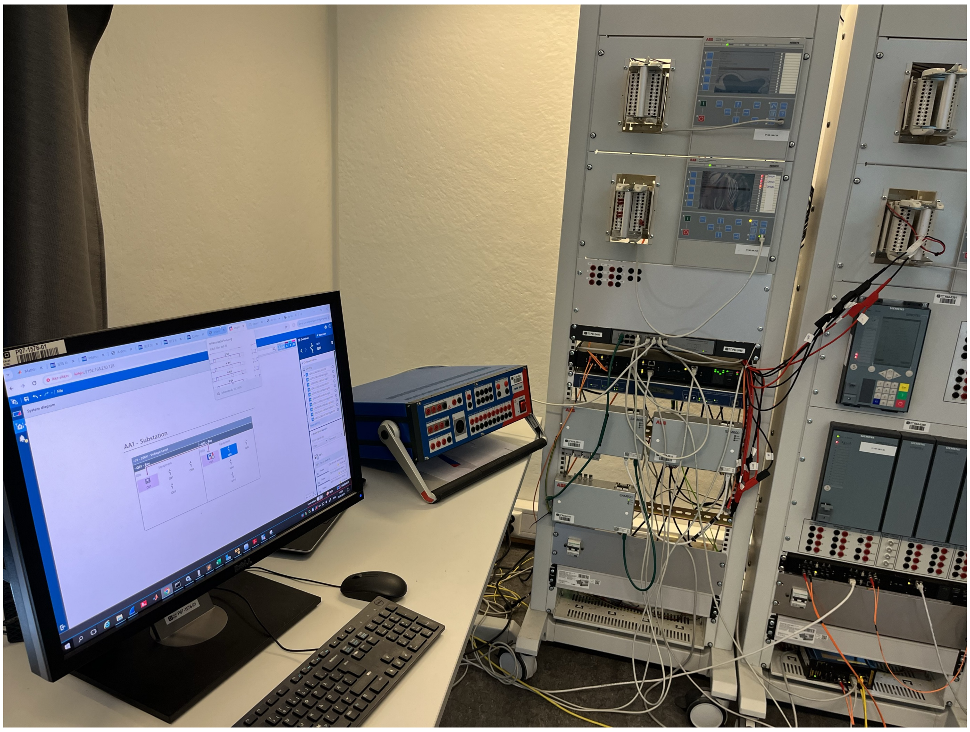


Figure 4.4.: ProDig Laboratory at NTNU. The laboratory computer shows StationScout, and the two configured IEDs are seen in the rack. The CMC356 is also shown but is not used here.

Figure 4.5.: CMC356 I/O test tool from OMICRON for GOOSE and SV [75].

Chapter 5

Results

Using the developed Python script, several JSON test files were generated. An excerpt of step two from the JSON test file used in the laboratory is seen in Listing 12. As this test follows the same interlocking logic that was shown throughout the thesis, this step can be directly related to the first step shown in Table 3.1, as it is seen that all switchgear are in "POS_OFF" i.e. the open position, except for `"Q05CTRL/SXCBR2.Pos.stVal"`, which is the address that controls the CB of bay one cf. Table 4.1. Only the order of operations is different, as the script ensures that CBs operate last when closing. This is the second step in the test file and not the first, as indicated by Table 3.1 due to the script prepending an initialisation step, where all switchgear are in the open position. It is furthermore seen that an interlocking condition is expected in this configuration, as indicated by the false value from the CILO LN and the CMD "CAR_BLOCKED_BY_INTERLOCKING" from the switch controller of the DUT. This is once again in alignment with what is seen in Table 3.1. The full test file is seen in Section G of the appendix, and the full test report is seen in Section I of the appendix. Step two from that test report is also shown in Table 5.1 for easy comparison with the test case snippet in Listing 12, and step two of Table 3.1.

```
{
      "description": "",
      "ordered": true,
      "expected": [
        {
          "signalRef": "Q05CTRL/SXSWI1.Pos.stVal",
          "value": "POS_OFF"
        },
        {
          "signalRef": "Q05CTRL/SXSWI2.Pos.stVal",
          "value": "POS_OFF"
        },
        {
          "signalRef": "Q05CTRL/SXSWI3.Pos.stVal",
          "value": "POS_OFF"
        },
        {
          "signalRef": "Q05CTRL/SXSWI5.Pos.stVal",
          "value": "POS_OFF"
        },
        {
          "signalRef": "Q05CTRL/SXCBR1.Pos.stVal",
          "value": "POS_OFF"
        },
        {
          "signalRef": "Q05CTRL/SXCBR2.Pos.stVal",
          "value": "POS_ON"
        },
        {
          "signalRef": "Q01CTRL/SCILO1.EnaCls.stVal",
          "value": false
        },
        {
          "signalRef": "Q05CTRL/SXSWI4.Pos.stVal",
          "commandResult": "CAR_BLOCKED_BY_INTERLOCKING"
        }
      ]
    },
```

Listing 12: Excerpt from JSON test file showing the first test step [65].

Table 5.1.: Step 2 from the test case results report exported from StationScout.

Step 2 – 2024-05-21 18:06:41.631+02:00 – Passed

Control

Signal	Timestamp	Value	Expected Value
Q05 QA1 – Q05CTRL/SXCBR1.Pos	2024-05-21 18:06:12.234+02:00	Open	Open
Command to Q01 QA1 – Q05CTRL/SXCBR2.Pos	2024-05-21 18:06:40.111+02:00	Closed	Closed
Q05 QB1 – Q05CTRL/SXSWI1.Pos	2024-05-21 18:06:13.795+02:00	Open	Open
Q05 QB2 – Q05CTRL/SXSWI2.Pos	2024-05-21 18:06:13.808+02:00	Open	Open
Q05 QC11 – Q05CTRL/SXSWI3.Pos	2024-05-21 18:06:15.375+02:00	Open	Open
Command to Q01 QB2 – Q05CTRL/SXSWI5.Pos	2024-05-21 18:06:13.791+02:00	Open	Open

Signal Assessment

Signal	Timestamp	Value	Expected Value
Close operation at open or interm. or bad pos. is enabled – Q01CTRL/SCILO1.EnaCls	2024-04-28 20:48:08.781+02:00	False	False

Command Assessment

Signal	Timestamp	Value	Expected Value
Command to Q01 QB1 – Q05CTRL/SXSWI4.Pos	2024-05-21 18:06:41.630+02:00	Blocked-by-interlocking	Blocked-by-interlocking

Regarding the validity of the JSON file that was generated for the hardware test, both methods of validation, that is, the built-in schema and the online validation tool, are in agreement that the JSON file was valid [65].

Figure 5.1 shows the view in StationScout immediately after executing the test. The substation configuration is seen on the left-hand side of the figure, and the test steps on the right, showing eight green checkmarks, signify that the expected and actual assessments of every test step are identical, which further validates the generated test case [65]. Moreover, it is seen from the figure that the entire test was executed in the span of approximately 11 seconds. For a more detailed look at every test step, refer to Section I of the appendix.

In order to show that the generation of test cases is general and not specific to this one case, an extended test case has been created using another switch and another test step. As the creation of the test case is only based on the data in the Excel tables, there is no practical upper limit for the number of test steps or the amount of switching operations per step. The extended test case is seen in the appendix in Section H.

Furthermore, a tool for visualising the SCD file in a non-proprietary manner has been developed. This is seen in Figure 5.2. This view provides an overview of the equipment present in the different bays similar to what is seen in StationScout, cf. Figure 5.1. It serves to give a better

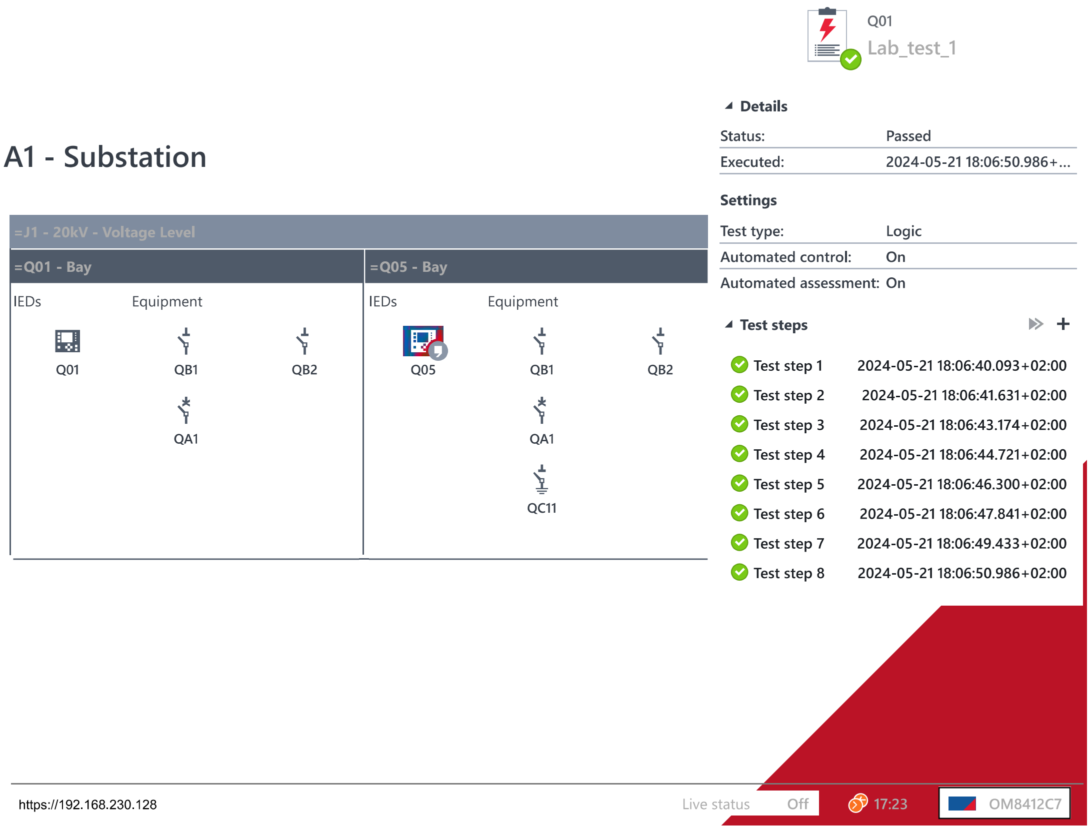


Figure 5.1.: Successful execution and assessment of generated interlocking logic test case in StationScout [65].

understanding of which parts of the substation are being present, and in conjunction with the generated test file, it becomes even clearer where the tests are performed. It shows the substation name and description at the top level. In this case, the substation is named ”AA1”, and the description is ”Munich”. Furthermore, it is seen that there are two voltage levels. There are five and two bays in the 320 kV and 33 kV voltage levels, respectively. The names of bays are seen in bold, white text at the top of each bay, and the IEDs present in the respective bays are seen in magenta. Finally, the conducting equipment and their functions are seen in white below the IEDs’ names. For instance, it is seen that bay Q05 on the high voltage side has a piece of equipment named QA1, which is a circuit breaker, as indicated in the description in the parenthesis (CBR). Voltage and current transformers (VTR and CTR) are also visible in this view.

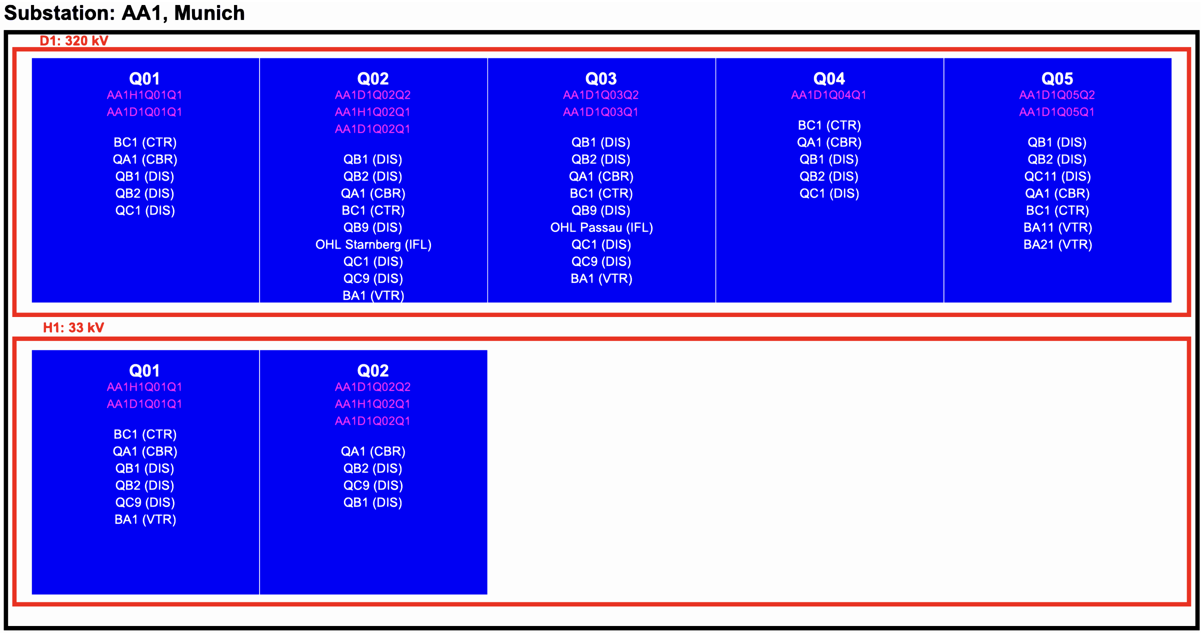


Figure 5.2.: Visualisation of SCD file using Python's ElementTree package to extract data and tkinter to show the graphics.

Chapter 6

Discussion

A test case was successfully generated and tested in the digital substation laboratory at NTNU using the setup described in chapter 4. This shows that the workflow from start to end works. This entails the generation, validation, execution and assessment of a test case in StationScout.

With the proof of concept that the test cases generated worked as intended, a more complex test case, including another earthing switch, was generated without any problems. Provided that an SCD file is created containing the same devices with the same names and signal addresses, the proof of concept implies that this more complex test case would also work. In any case, it proves that the test cases that can be generated from the Python tool are not limited to a certain number of test steps or switches.

Given the potential savings in terms of risk, time and money when performing tests on remote or offshore substations, it was shown that the interlocking test could be done remotely. Furthermore, by simulating and utilising the spare contacts of another IED as a virtual I/O box, the testing could be done remotely without tools like the CMC356 or a separate IED used only for this purpose. Testing remotely poses a challenge in terms of cybersecurity, which must be given consideration.

While it was not possible to generate, execute, and assess a test case from purely the SCD file, it has been shown that with only a list of signal addresses and a table of switch sequences, it was possible through the developed Python script. Furthermore, the Python script reads the SCD file for the added benefit of confirming the presence of the signal addresses used in the test in the substation.

The inability to include the logic implementation from the IID files in the final SCD file slightly hinders the complete automation of the workflow, as the operator still needs to create a table of the switching sequences and associated signal addresses that are subject to test.

Table 2.6 shows the estimated time taken to conduct an interlocking test using a method from a previous study [57]. The time taken for the method employed in this thesis is in general agreement with the estimates for automatic testing given there. As pointed out by Loenders, Lisiecki and Yesil, the preparations to facilitate the interlock testing can be reduced to zero in a fully digital system but took 30 minutes to 120 minutes in that study. With the method employed, there would still be preparations, although it was an interlocking test on a fully digital system, as input files, i.e. the Excel tables, would have to be made ready. However, this would only take a few minutes for an experienced operator familiar with the system. The testing itself took only approximately 11 seconds as opposed to the 1 minute outlined in [57], but this difference is inconsequential and would be smaller for a larger test case. During the test case executed here, there was no need for any corrective measures, but the sample size is too small to conclude that this would always be the case.

Overall, the two methods are comparable in the time taken to execute a test, and both results would be faster if the internal programmable logic of the IEDs was accessible in the SCL files [57]. The testing of the interlocking scheme itself is substantially faster ($\sim$11 s) when done automatically compared to the conventional way (30 min) [57].

A great benefit of automatic testing in StationScout based on JSON test cases is that it can easily be repeated after any changes to the substation. Furthermore, due to the simple input files, it is trivial to port a test case to another bay governed by the same interlocking logic. This could even be a bay in an entirely different substation. If the logic equations describing the two systems are identical, then the only thing to ensure is that the signal addresses are changed to fit the new bay.

Part of the work done in this thesis was about parsing the SCD file to verify the signal addresses and have the capabilities ready to extract logic information from an SCD when the IEC releases standards supporting this. This allowed for the generation of a view of the substation elements in each bay and voltage level. It would have been even more beneficial to have created a single line diagram from the SCD file. However, it is still not common engineering practice to include this primary topological information in the SCL files, as this is optional and most commonly only included if it is specifically required by the customer [65].

It is unclear whether this tool will see any use as there is a lot of reluctance towards automating logic testing as described in Section 2.4.3.1: Remote Testing and Maintenance. During the work, a PACS engineer from OMICRON and his colleague expressed interest in the script and would like to be granted access.

6.1. Future Work

In the future, if the implementation of interlocking logic specifications directly in the SCD files becomes commonplace, the input requirement to the script can be limited to only the SCD file, which could drastically reduce the engineering efforts required for using the script. Furthermore, if it becomes the rule rather than the exception to include enough primary topological information to produce a single-line diagram from the SCL files, doing so in the script would be a helpful addition to the operators [65].

Future studies should look at whether including system logic and/or BAPs from IEC 61850-90-30, 90-11, 7-6, and IEC 61131-3 would improve the efficiency of the interlocking testing. Moreover, it is expected that the Application Specification Description (ASD) files created from BAPs can be used to create an SCD file, which could include logic specifications [65], and the script could be updated accordingly. It is, however, not a high priority until utilities start embracing these possibilities.

There is abundant room for further progress in pinpointing why utilities are reluctant to use the monitoring and remote testing capabilities of IEC 61850. It could be of great interest to find out what could be done to make the transition towards a smarter substation go faster.

If further studies are undertaken that focus on the remote testing aspect of this thesis, the addition of cyber security measures is critical and should also be implemented.

An important expansion of the script would be to add a check for the energisation status of the lines, cables or busbars using the values from the associated MUs [65]. This could also be simulated by sending a virtual signal, either signifying that the line is energised or not, but whether or not it would pose a challenge remains to be elucidated. Furthermore, executing the expanded test case seen in Appendix H remains to be done, but it is hypothesised that, as long as the HW configuration matches the test case, the test case should run successfully. This particular test file would involve configuring another switch for the bay one IED in the IED configuration tool and making the required hardware connections.

The script has not yet been tested with any switches in the ”intermediate”, meaning neither closed nor open, position. However, it is expected that that would be trouble-free, as the script simply reads what expected position is put in the columns of the test sequence table and writes that in the corresponding location in the test case. Similarly, suppose an open or closed position is expected, but a switch enters the intermediate position during the test. In this case, a failed test step will result, which will be clear in StationScout after running the test.

It would also be interesting to run a test where the control of one of the switches is inverted while keeping the expected assessments the same. This should show that the generated test case produces test results that conclude the interlocking logic cannot be verified. By showing failed test cases and describing the cause of the failure, the successful test cases gain more credibility.

A version of the script could be implemented that creates a test case containing all 2^n permutations of the switch configurations (discounting intermediate positions), but it was decided that this was unnecessary for now. It would quickly take an extreme amount of time, and many of the switch configurations would not be necessary to test.

Finally, it would be interesting to test this script on IEDs interfacing with actual switchgear, validating the script in this setting rather than just examining the resulting position changes in the data model.

Chapter 7

Conclusion

It has been shown that the script developed can generate a valid test file that integrates well with the existing SAS testing software, and it works irrespective of whether IEDs are simulated or physically present in the substation, thanks to the possibilities within IEC 61850. The script is furthermore able to generate a test case for any number of IEDs in any number of bays and with any number of test steps.

For multiple reasons, the Python-based script is more versatile than the VBA script developed in [55]. It has the advantage that it can run on any operating system that can run Python, whereas the VBA macros are restricted to the Windows OS, as it is tied to that file explorer. In contrast to the VBA script from OMICRON [55], this script is completely non-proprietary. Furthermore, the Python script can, with relative ease, be expanded upon to include newer extensions from, e.g., IEC 61850-90-30, as SCD parsing is already part of it [65].

It was possible to make the script handle the order of operations between disconnectors and CBs by itself, thus eliminating the need for information regarding the order of the switching sequence in the various test steps. Another convenience implemented during this work was the automatic validation of the JSON using a predefined schema, which eliminates the need to carry out that operation manually. Lastly, by providing an alternative to the VBA script, the reliance on Microsoft Office and the threat of malicious macros are no longer factors to account for, making it safer and more accessible.

It was shown that an interlocking test containing eight steps was able to be executed in only approximately 11 seconds on a laboratory setup involving both a virtual and a physical IED. This is faster than what previous work on the topic has shown [57], although this is arguably inconsequential.

The advantages of digital substations have been addressed. It was found that the footprint, weight, safety and several aspects of the maintenance process, are improved compared to a conventional substation. Furthermore, digital substations are not reliant on the sheer amount of copper cables that are necessary for conventional substations. The standardised workflow that the IEC 61850 offers is extremely valuable and can allow for great interoperability.

Relevant parts of the IEC 61850 have been described, as well as the ongoing work to expand the standard series and provide a better platform for automation. These improvements are largely driven by the WG10 of TC57 in the IEC.

During the work, it was possible to contribute to the ProDig laboratory by developing a script that could potentially be used in future testing and by giving laboratory users access to VBX1.

Finally, the work done in this thesis was disseminated in a conference paper, which was presented at PAC World 2024 in Athens [65]. Here, utility personnel and TSOs showed interest in the work [80].

Bibliography

[1] F. Kühn, F. Liebach, T. Matthey, A. Schlosser and J. Zivansky, *How to Succeed in the Expanding Global Offshore Wind Market*, McKinsey & Company, Apr. 2022. [Online]. Available: https://www.mckinsey.com/industries/electric-power-and-natural-gas/our-insights/how-to-succeed-in-the-expanding-global-offshore-wind-market#.

[2] International Council on Large Electric Systems (CIGRE) Study Committee B3: Substations, *Substations* (CIGRE Green Books), T. Krieg and J. Finn, Eds. Paris, France: Springer, 2019, ISBN: 978-3-319-49573-6. DOI: 10.1007/978-3-319-49574-3.

[3] Norwegian University of Science and Technology (NTNU), *PRODIG: Protection and Diagnosis*, https://www.ntnu.edu/prodig, Accessed: 2024-05-29.

[4] P. Bishop and N.-K. C. Nair, *IEC 61850 Principles and Applications to Electric Power Systems* (CIGRE Green Books), 2nd. Springer, 2023, ISBN: 978-3-031-24566-4 (print). DOI: 10.1007/978-3-031-24d567-1.

[5] B. Heimisson, T. Jónsson, P. Mohapatra and F. Steinhauser, 'Large scale application of fully Digital Substations at LANDSNET, Iceland', in *CIGRE Session 2022*, B5-10641, LANDSNET, Iceland and OMICRON electronics GmbH, Austria, Iceland, 2022.

[6] L. Stensrud, B. Ohrn, R. Loken, N. Hurzuk and A. Apostolov, 'Testing of IED in a Digital substation', *The Journal of Engineering*, vol. 2018, Jul. 2018. DOI: 10.1049/joe.2018.0172.

[7] B. Heimisson, 'Landsnet's Road to Fully Digital Transmission System', *PAC World*, 066 Dec. 2023, Accessed: 2024-05-23. [Online]. Available: https://www.pacw.org/landsnets-road-to-fully-digital-transmission-system.

[8] PAC World, 'The digital substations market growth', *PAC World*, 058 Dec. 2021, Accessed: 2024-05-23. [Online]. Available: https://www.pacw.org/the-digital-substations-market-growth.

[9] Y. Yuan and Y. Yang, *IEC 61850-Based Smart Substations*. Academic Press, 2019. [Online]. Available: https://doi.org/10.1016/c2017-0-00584-0.

[10] A. Apostolov, 'Efficient maintenance testing in digital substations based on IEC 61850 edition 2', *Protection and Control of Modern Power Systems*, vol. 2, p. 37, Dec. 2017. DOI: 10.1186/s41601-017-0054-0.

[11] R. Hunt, B. Flynn and T. Smith, 'The Substation of the Future: Moving Toward a Digital Solution', *IEEE Power and Energy Magazine*, vol. 17, no. 4, pp. 47–55, 2019. DOI: 10.1109/MPE.2019.2908122.

[12] A. Chattopadhyay, A. Ukil, D. Jap and S. Bhasin, 'Toward Threat of Implementation Attacks on Substation Security: Case Study on Fault Detection and Isolation', *IEEE Transactions on Industrial Informatics*, vol. 14, pp. 2442–2451, Jun. 2018. DOI: 10.1109/TII.2017.2770096.

[13] International Electrotechnical Commission, 'IEC 61850-2:2019 – Communication networks and systems for power utility automation – Part 2: Glossary', Standard, Apr. 2019, Edition 2.0.

[14] International Electrotechnical Commission, 'Industrial communication networks - High availability automation networks - Part 3: Parallel Redundancy Protocol (PRP) and High-availability Seamless Redundancy (HSR)', Standard, Dec. 2021, Edition 4.0.

[15] R. C. Bernardino, C. M. Martins, P. S. Pereira, G. E. Lourenço and P. S. P. Junior, 'Link redundancy in the process bus according to IEC 61850 ED.2: experience with RSTP, PRP and HSR protocols', in *16th International Conference on Developments in Power System Protection (DPSP 2022)*, vol. 2022, 2022, pp. 164–169. DOI: 10.1049/icp.2022.0931.

[16] International Electrotechnical Commission, 'Industrial communication networks - High availability automation networks - Part 2: Media Redundancy Protocol (MRP)', Standard, Dec. 2021, Edition 3.0.

[17] International Electrotechnical Commission, 'IEC 62439-1:2010 – Industrial communication networks - High availability automation networks - Part 1: General concepts and calculation methods', Standard, Feb. 2010, Edition 1.0.

[18] H. Karandikar, T. Neighbours and R. Pate, 'Digital Switchgear: The Next Phase in the Evolution of Safety by Design', *IEEE Industry Applications Magazine*, vol. 27, no. 3, pp. 23–30, 2021. DOI: 10.1109/MIAS.2020.3024491.

[19] Hitachi Energy, 'Hybrid Communications, Cybersecurity and the Future of Offshore Wind', 2021.

[20] D. Xu, W. Sae-Kok, A. Vujanic, A. Motta, N. Powers and T. Neo, 'Fiber-optic Current Sensor (FOCS): Fully Digital Non-conventional Instrument Transformer', in *2019 IEEE PES GTD Grand International Conference and Exposition Asia (GTD Asia)*, 2019, pp. 982–987. DOI: 10.1109/GTDAsia.2019.8715989.

[21] R. Thomas et al., 'Non-conventional instrument transformers enabling digital substations for future grid', in *2016 IEEE/PES Transmission and Distribution Conference and Exposition (T&D)*, 2016, pp. 1–5. DOI: 10.1109/TDC.2016.7519930.

[22] H.-E. Olovsson, T. Werner and P. Rietmann, 'Next Generation Substations, Impact of the Process Bus', 19th Aug. 2010. [Online]. Available: https://library.abb.com/d/9AKK105152A5172.

[23] P. Shaub, D. J. Haywood, A. Kenwrick, D. Ingram and G. Dusha, 'Test and evaluation of Non Con- ventional Instrument Transformers and sampled value process bus on Powerlink's transmission network', in *South East Asia Protection and Control Conference (SEAPAC 2011)*, 2011.

[24] Control Engineering Staff, *Fiber Optic vs. Copper: 5 Reasons Fiber Wins*, Accessed: 2024-02-08, 29th Jan. 2008. [Online]. Available: https://www.controleng.com/articles/fiber-optic-vs-copper-5-reasons-fiber-wins/.

[25] S. Hennell, '36kV Outdoor Structure Mounted Instrument Transformers', National Grid, Engineering Specification EE SPEC: 184/1, Aug. 2020.

[26] MULTICOMP PRO, *Tri-Rated Flexible PVC Equipment Wire*, Datasheet, Type: PP001199, Wire, Tri Rated, PVC, Blue, 14 AWG, 2.5 mm², 328 ft, 100 m, 1st Apr. 2020.

[27] Genuine Modules, *How much does fiber optic cable weigh?*, https://www.genuinemodules.com/how-much-does-fiber-optic-cable-weight_a5511, Accessed: 2024-02-05, Jan. 2017.

[28] S. Kumar, A. Abu-Siada, N. Das and S. Islam, 'Review of the Legacy and Future of IEC 61850 Protocols Encompassing Substation Automation System', *Electronics*, vol. 12, no. 15, 2023, ISSN: 2079-9292. DOI: 10.3390/electronics12153345. [Online]. Available: https://www.mdpi.com/2079-9292/12/15/3345.

[29] International Electrotechnical Commission, 'Telecommunications and exchange between information technology systems – Requirements for local and metropolitan area networks – Part 3: Standard for Ethernet', Standard, 3rd Mar. 2021, Edition 3.0.

[30] M. Froese and S. A. Kunsman, *Going Digital: A Look at the Modern Substation*, Accessed: 2024-02-07, ABB, 17th Oct. 2017. [Online]. Available: https://www.windpowerengineering.com/going-digital-look-modern-substation/.

[31] CIGRE Working Group B5.36, 'Applications of IEC 61850 Standard to Protection Schemes', CIGRE, Technical Brochure TB540, Jun. 2013.

[32] F. Steinhauser, 'Embrace IEC 61850 Engineering - From Bottom-up to Top-Down', in *Proc. 2022 PAC World Conference*, Prague, Czech Republic: OMICRON electronics GmbH Austria, Jun. 2022.

[33] CIGRE Working Group B5, 'IEC 61850 based substation automation systems – Users expectations and stakeholders interactions', CIGRE, Technical Brochure 819, Nov. 2020.

[34] A. Apostolov, 'Impact of IEC 61850 on the interoperability and reliability of protection schemes', in *2013 IEEE Power & Energy Society General Meeting*, 2013, pp. 1–5. DOI: 10.1109/PESMG.2013.6673051.

[35] A. Apostolov, 'Remote testing of multifunctional protection IEDs — Benefits and challenges', in *13th International Conference on Development in Power System Protection 2016 (DPSP)*, 2016, pp. 1–6. DOI: 10.1049/cp.2016.0039.

[36] CIGRE Working Group B5.53, 'Test strategy for Protection, Automation and Control (PAC) functions in a fully digital substation based on IEC 61850 applications', CIGRE, Technical Brochure TB760, Mar. 2019.

[37] International Electrotechnical Commission, 'IEC 61850-8-1:2011 – Specific communication service mapping (SCSM) – Mappings to MMS (ISO 9506-1 and ISO 9506-2) and to ISO/IEC 8802-3', Standard, 2011, Edition 2.0.

[38] International Electrotechnical Commission, 'IEC 61850-9-2:2012 – Communication networks and systems for power utility automation – Part 9-2: Specific communication service mapping (SCSM) – Sampled values over ISO/IEC 8802-3', Standard, Feb. 2012, Edition 2.0.

[39] International Electrotechnical Commission, 'IEC 61850-7-2:2010 – Communication networks and systems for power utility automation – Part 7-2: Basic communication structure – Abstract communication service interface (ACSI) – Mappings to MMS (ISO 9506-1 and ISO 9506-2) and to ISO/IEC 8802-3', Standard, 9th Nov. 2010, Edition 2.0.

[40] International Electrotechnical Commission, 'IEC 61850-5:2013 – Communication networks and systems for power utility automation – Part 5: Communication requirements for functions and device models', Standard, 3rd Jun. 2013, Edition 2.0.

[41] International Electrotechnical Commission, 'IEC 61850-7-1:2011 – Communication networks and systems for power utility automation – Part 7-1: Basic communication structure – Principles and models', Standard, 25th Nov. 2011, Edition 2.0.

[42] Omicron Electronics, *IEC 61850 System Engineering Concept and Testing in DSS*, Obtained through e-mail correspondence with a Sales and Application Engineer from Omicron, Apr. 2024.

[43] International Electrotechnical Commission, 'IEC 61850-7-4:2010 – Communication networks and systems for power utility automation – Part 7-4: Basic communication structure – Compatible logical node classes and data object classes', Standard, 16th Jul. 2010, Edition 2.0.

[44] International Electrotechnical Commission, 'IEC 61850-6:2009 – Communication networks and systems for power utility automation - Part 6: Configuration description language for communication in electrical substations related to IEDs', Standard, 2009.

[45] M. A. Aftab, S. Hussain, S. Roostaee, I. Ali, S. Mehfuz and M. Thomas, 'Performance evaluation of iec 61850 goose based inter substation communication for accelerated distance protection scheme', *IET Generation, Transmission and Distribution*, vol. 12, Aug. 2018. DOI: 10.1049/iet-gtd.2018.5481.

[46] International Electrotechnical Commission, 'IEC 61850-1:2013 – Communication networks and systems for power utility automation – Part 1: Introduction and overview', Standard, Mar. 2013, Edition 2.0.

[47] International Electrotechnical Commission, 'IEC 62351-6:2020 – Power systems management and associated information exchange - Data and communications security - Part 6: Security for IEC 61850', Standard, Oct. 2020, Edition 1.0.

[48] International Electrotechnical Commission, 'Instrument transformers - Part 9: Digital interface for instrument transformers', Standard, 2019.

[49] OMICRON electronics, *Merging Unit Testing*, https://www.omicronenergy.com/en/solution/merging-unit-testing/, Accessed: 2024-06-04, 2024.

[50] International Electrotechnical Commission, 'IEC 61588:2021 – Precision Clock Synchronization Protocol for Networked Measurement and Control Systems', Standard, 14th Jun. 2021, Edition 3.0.

[51] International Electrotechnical Commission, 'Instrument transformers - Part 13: Stand-alone merging unit (SAMU)', Standard, 2021.

[52] Landsnet-Icegrid, *Requirements for Protection and Control Systems: Principles for the Icelandic Power Transmission Network*, version 05, 25th Mar. 2011. [Online]. Available: https://www.landsnet.is/library?itemid=29339c03-75e7-407c-b36e-10d325150a89.

[53] National Grid, 'National Grid Substation Interlocking Technical Schemes Specification', National Grid Company plc, Tech. Rep. NGTS 3.1.1, Oct. 1993.

[54] Y. Liu, R. Yang and H. Zhu, 'Research and development of centralized GOOSE interlocking device for intelligent substation', in *2024 IEEE 7th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, vol. 7, 2024, pp. 981–986. DOI: 10.1109/IAEAC59436.2024.10503625.

[55] B. Tahincioglu, T. Schossig and B. Heimisson, 'A Novel Approach for Fully Automated Testing of Interlockings', in *Proc. 2023 PAC World Conference*, Glasgow, Scotland: OMICRON electronics GmbH Austria and Landsnet Iceland, Jun. 2023.

[56] International Electrotechnical Commission, 'Communication networks and systems for power utility automation – Part 90-11: Methodologies for modelling of logics for IEC 61850 based applications', Standard, Sep. 2020, Edition 1.0.

[57] R. Loenders, T. Lisiecki and I. Yesil, 'Practical review and advancements in testing multivendor digital substations', in *27th International Conference on Electricity Distribution (CIRED 2023)*, IET, vol. 2023, 2023, pp. 1110–1114.

[58] C. Brunner, *E-mail to Jens Kruse-Hansen*, E-mail, See Appendix A for email contents., May 2024.

[59] International Electrotechnical Commission, 'IEC 61850-7-6:2019 – Communication networks and systems for power utility automation – Part 7-6: Guideline for definition of Basic Application Profiles (BAP) using IEC 61850', Standard, 2019, Edition 1.0.

[60] C. Brunner, *Latest Update on IEC 61850*, Accessed: 2024-06-04, Switzerland: it4power, Mar. 2023. [Online]. Available: `https://www.pacw.org/latest-update-on-iec-61850`.

[61] OSMOSE, *Osmose Project Overview*, `https://www.osmose-h2020.eu/project-overview/`, Accessed: 2024-06-05.

[62] OSMOSE, 'Recommendations for IEC61850 WG10 and the industry WP7.1 - Deliverable 7.3', Tech. Rep., Feb. 2022.

[63] D. Abetz, 'Optional Use of IEC 61850 Test Modes in Substations', *PAC World Magazine*, Mar. 2024, Accessed: 2024-06-08. [Online]. Available: `https://www.pacw.org/optional-use-of-iec-61850-test-modes-in-substations`.

[64] OMICRON, *StationScout*, `https://www.omicronenergy.com/en/products/stationscout/`, Accessed: 2024-06-01.

[65] B. Tahincioglu, J. Kruse-Hansen and R. R. Syed, 'Automatic generation, execution, and assessment of interlocking tests', in *Proc. 2024 PAC World Conference*, Athens, Greece: OMICRON electronics GmBH Austria, NTNU and TU Delft, Jun. 2024.

[66] C. Ahmadi, J.-L. Chen and Y.-C. Lai, 'Enhancing Cybersecurity With P-Code Analysis and XGBoost: A Novel Approach for Malicious VBA Macro Detection in Office Documents', *IEEE Access*, vol. 12, pp. 71 746–71 760, 2024. DOI: `10.1109/ACCESS.2024.3402956`.

[67] National Cyber Security Centre, *Macro Security for Microsoft Office*, `https://www.ncsc.gov.uk/guidance/macro-security-for-microsoft-office`, Version 3.0, Accessed: 2024-06-09, Feb. 2019. [Online]. Available: `https://www.ncsc.gov.uk/guidance/macro-security-for-microsoft-office`.

[68] Python Software Foundation, *xml.etree.ElementTree — The ElementTree XML API*, Accessed: 2024-06-01, 2024.

[69] Python Software Foundation, *tkinter — Python interface to Tcl/Tk*, Python 3.11.6 documentation. Accessed: 2024-06-09, 2024. [Online]. Available: `https://docs.python.org/3/library/tkinter.html`.

[70] C. Brauner and E. Carvalheira, 'Functional testing of iec 61850 based substation automation systems', Feb. 2021. [Online]. Available: `https://electrical-engineering-portal.com/functional-testing-iec-61850-based-substation-automation-systems`.

[71] OMICRON electronics GmbH, *MBX2*, `https://www.omicronenergy.com/en/products/mbx/`, Accessed: 2024-06-26.

[72] OMICRON electronics GmbH, *VBX1*, `https://www.omicronenergy.com/en/products/vbx1/`, Accessed: 2024-06-26.

[73] VMware, *VMware Workstation Player*, `https://www.vmware.com/products/workstation-player/workstation-player-evaluation.html.html`, Accessed: 2024-06-01.

[74] OMICRON, *IEDScout*, `https://www.omicronenergy.com/en/products/iedscout/`, Accessed: 2024-06-01.

[75] OMICRON electronics, *CMC 356 - Universal relay test set and commissioning tool*, `https://www.omicronenergy.com/en/products/cmc-356/`, Accessed: 2024-05-28, 2024.

[76] ABB, *Protection and Control IED Manager PCM600*, `https://new.abb.com/medium-voltage/digital-substations/software-products/protection-and-control-ied-manager-pcm600`, Accessed: 2024-06-01.

[77] OpenSCD, *OpenSCD*, https://openscd.github.io/, First published by OMICRON in 2020. Accessed: 2024-06-01, 2020.

[78] ABB, *RED670 1.2 Pre-configured Product Guide*, https://library.e.abb.com/public/4916b9b1e8424727882a95df66a6e312/1MRK505228-BEN_E_en_Product_Guide__RED670_1.2_pre-configured.pdf, Accessed: 2024-06-01, Feb. 2015.

[79] The Wireshark Development Team, *Wireshark*, https://www.wireshark.org/, Accessed: 2024-06-01.

[80] B. T. Rizwan Rafique Syed, *E-mail to Jens Kruse-Hansen*, E-mail, See Appendix B for email contents., Jun. 2024.

Appendix

A. E-mail from Christoph Brunner

E-mail response from Christoph Brunner after he was asked about the implementation of interlocking logic in IEC 61850-90-30 and whether it is being looked into

Dear Jens,

No worries – you are not overstepping. . .

About logic integration into IEC 61850 (and logic in general; not just Interlocking)

There is the part IEC TR 61850-90-11 which is a first approach to integrate logic specification into IEC 61850. It basically addresses the two key aspects:
How to describe logic
How to link the variables from a logic described into an IEC 61850 data model

That part is published, but not yet used. . .

Then, we went a step further with IEC 61850-90-30 and IEC 61850-7-6, which both rely on the SCL extension namespace IEC 61850-6-100

IEC 61850-7-6 defines ways how to describe basic application profiles in SCL – relying on SCL extensions introduced in IEC 61850-90-30. That includes the possibility, to describe the behavior of an application. One option is, to describe that behavior in IEC 61131.

So while IEC 61850 is not defining syntax, to describe behavior (including logic), it adds the possibilities to refer to files that describe behavior in IEC 61131; e.g. structured text or function block diagrams.

Does this answer your question?

Kind regards
Christoph

B. E-mail from Rizwan Rafique Syed

E-mail responses from Rizwan Rafique Syed and Burak Tahincioglu after they were asked how the presentation of the conference paper and the conference in general went.

Hi Jens,
It was very smooth and interesting; it was one of the "automation and control" related papers in the conference. We received good feedback from main Utilities, TSOs such as RTE France, ESB Ireland, TenneT Germany about the work. They said they will send it to their colleagues to use it.
Thanks again for the great work done and let's continue staying connected,

Best Regards from Køge,
Burak

Hi Jens,

It went very well, and the utility personnel showed interest in the work. The papers presented were really interesting and the overall experience was great. We will talk more when meet you in person.

Best regards,
Rizwan

From: Jens Kruse-Hansen
Sent: Monday, June 24, 2024 2:14 PM
To: Burak Tahincioglu, Rizwan Rafique Syed
Subject: PACW Conference

Hi Guys,

I just wanted to ask if the presentation went well. Did you enjoy the conference?

Best regards,
Jens

C. Q01 Hardware Configuration

IED Type: RED670

Variant: 6U 1/2 19" rack casing, 1 TRM

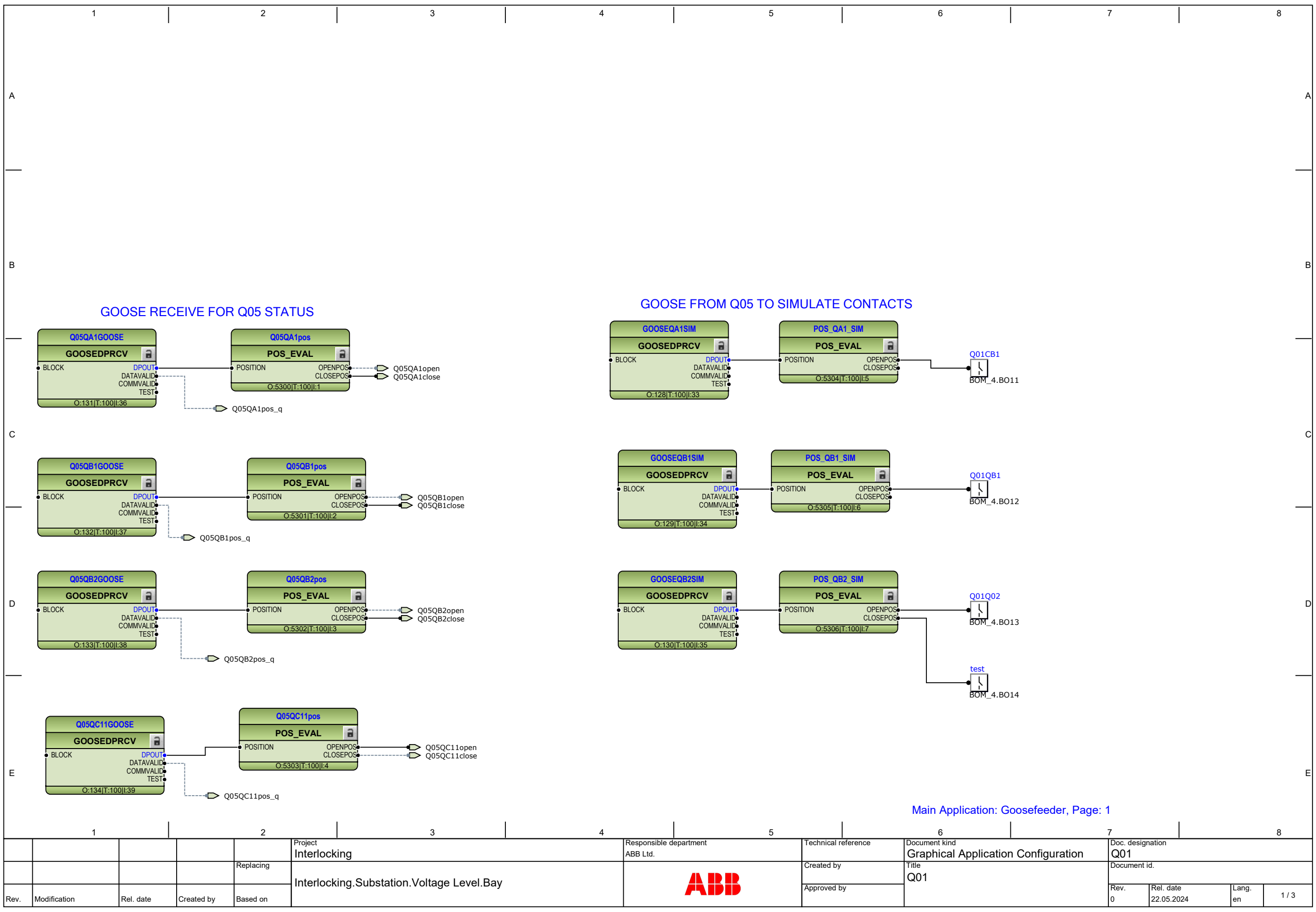
Display Type: IEC 6U 1/2 19"

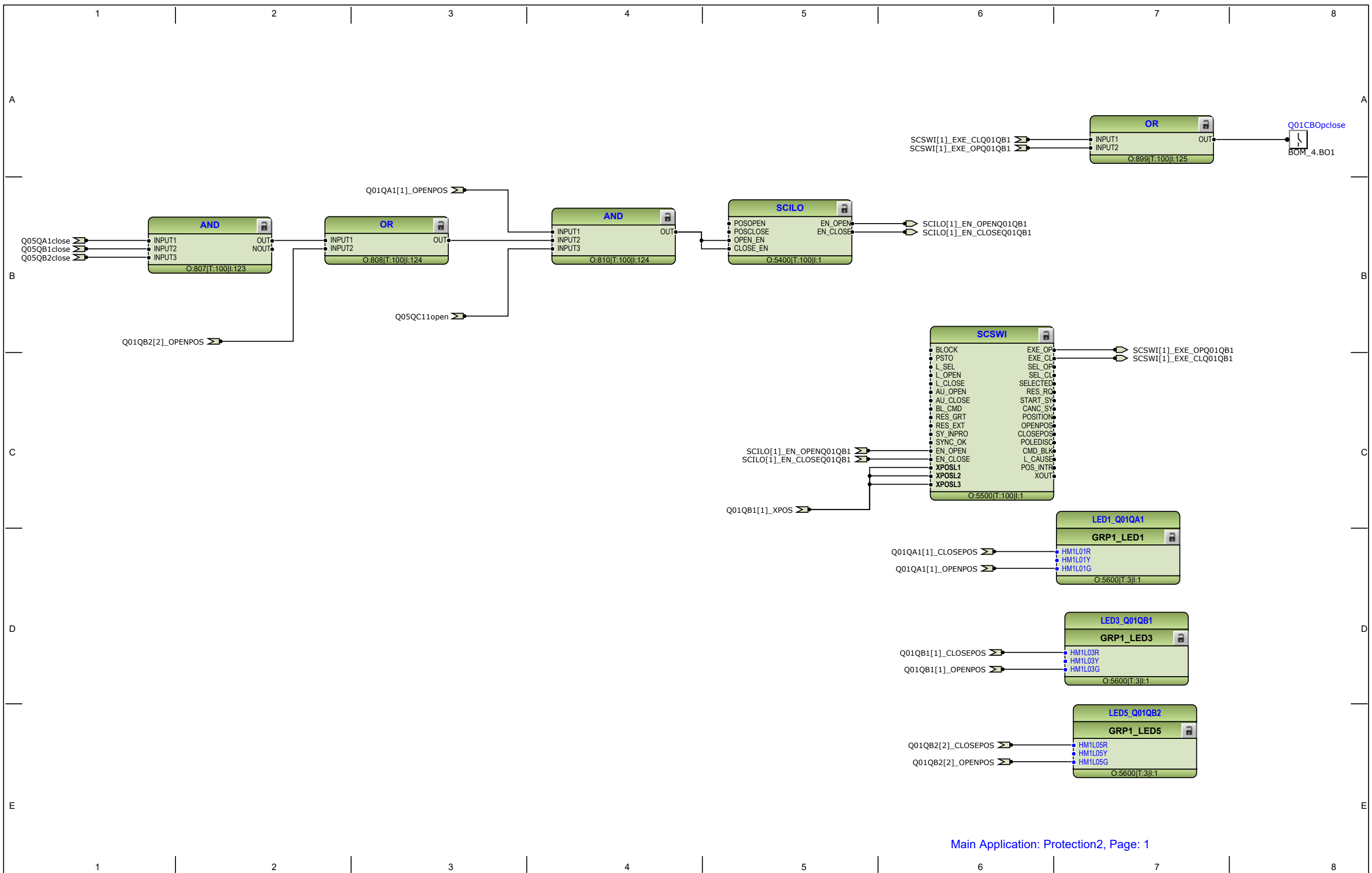
					Project Interlocking	Responsible department ABB Ltd.	Technical ref...	Document kind Hardware Conf.	Doc. designation Q01
				Repla...	Interlocking.Substation.Voltage Level.Bay	ABB	Created by	Title Q01	Document id.
Rev.	Modification	Rel. date	Created by	1 Based on			Approved by		Rev. 0 · Rel. date 22.05.2024 · Lan en · 1 / 2

Hardware Configuration

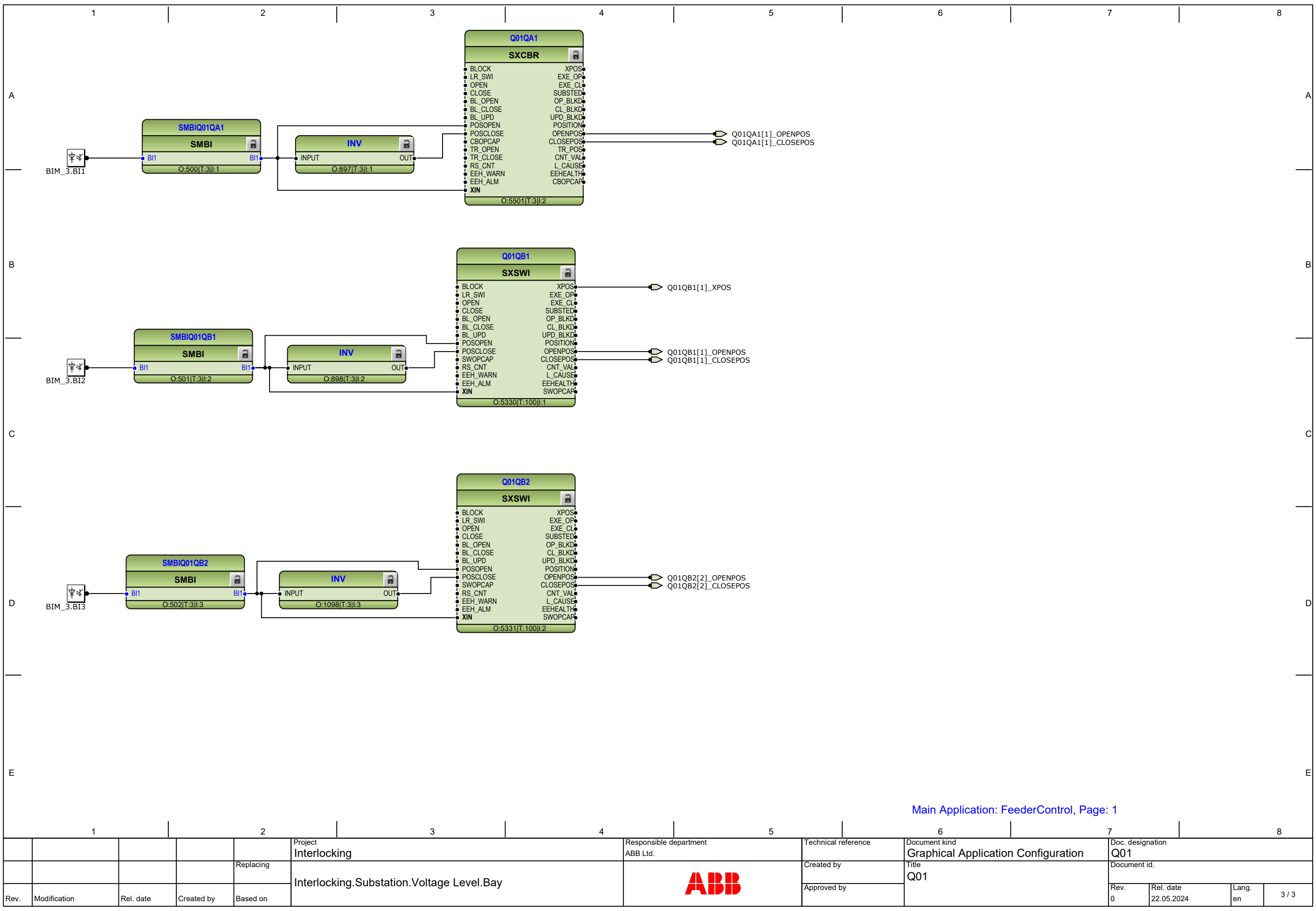
Card Space	Card Type	Card Identifier
p1	PSM	PSM_1
p3	BIM	BIM_3
p4	BOM	BOM_4
p5		
p30	NUM	NUM_30
p30:1		
p30:2	LDCM Analog 1	LDCM_302
p30:3		
p31	ADM	ADM_31
p31:1	OEM 2 port	OEM_311
p31:2	GTM	GTM_312
p31:3	IRIG	IRIG-B_313
p40	TRM 6I + 6U	TRM_40
s921	MU1 4I + 4U	MU1_921
s922		
s923		
s924		
s925		
s926		

D. Q01 Application Configuration

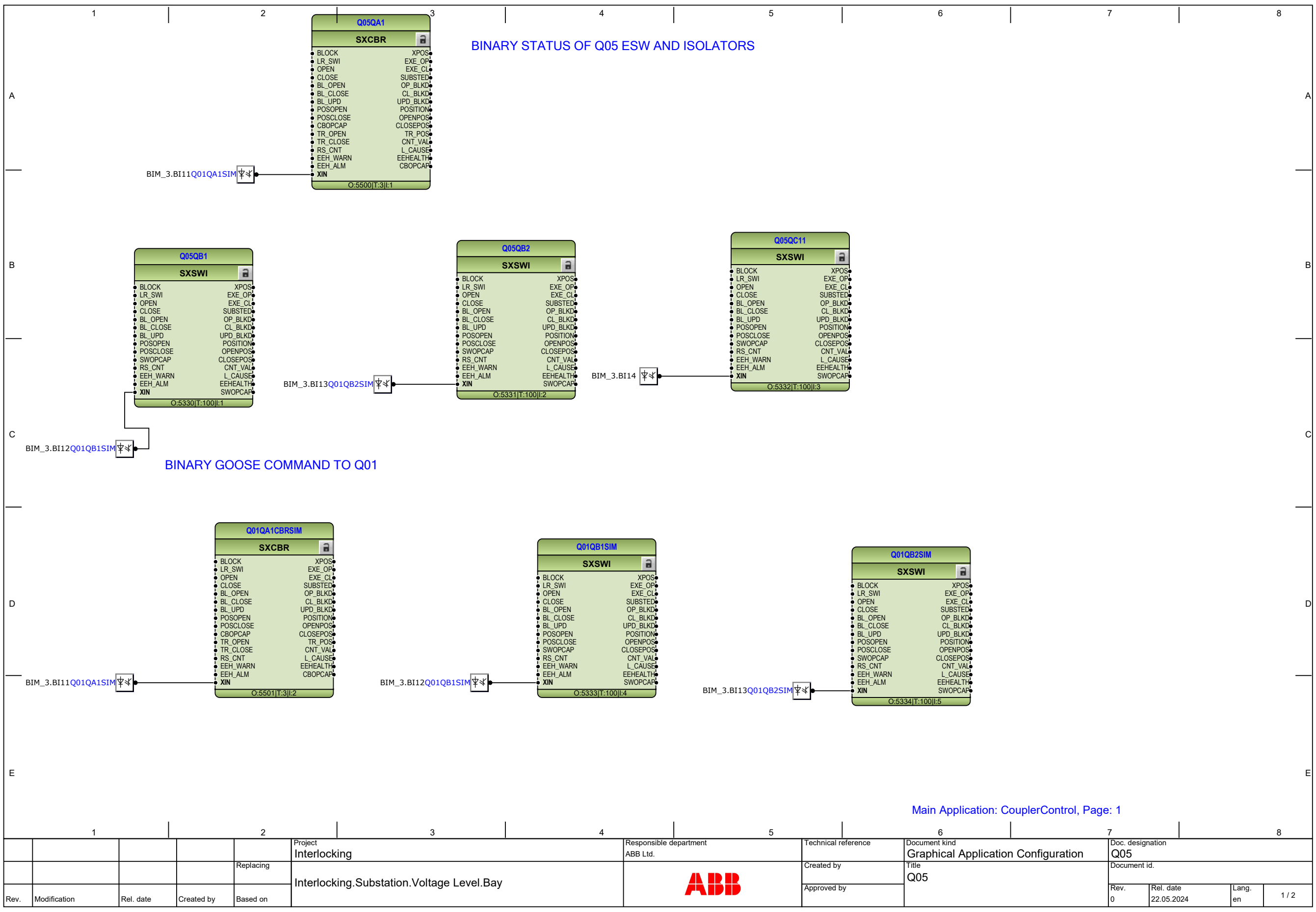




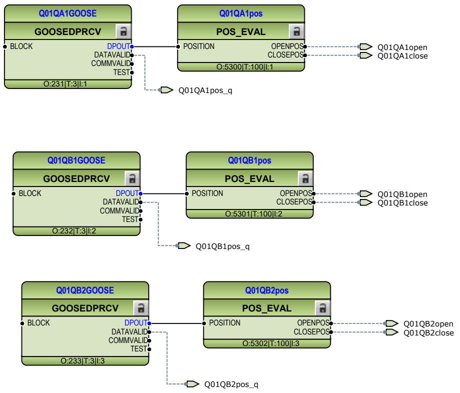
				Project: Interlocking	Responsible department: ABB Ltd.	Technical reference	Document kind: Graphical Application Configuration	Doc. designation: Q01
				Replacing	Interlocking.Substation.Voltage Level.Bay	ABB	Created by	Title: Q01	Document id.
Rev.	Modification	Rel. date	Created by	Based on			Approved by		Rev. 0 · Rel. date 22.05.2024 · Lang. en · 2 / 3


Q01QA1
SXCBR
SMBIQ01QA1
SMBI
INV
BIM_3.BI1
O:500|T:3|I:1
O:897|T:3|I:1
O:5501|T:3|I:2
Q01QA1[1]_OPENPOS
Q01QA1[1]_CLOSEPOS
Q01QB1
SXSWI
SMBIQ01QB1
BIM_3.BI2
O:501|T:3|I:2
O:898|T:3|I:2
O:5330|T:100|I:1
Q01QB1[1]_XPOS
Q01QB1[1]_OPENPOS
Q01QB1[1]_CLOSEPOS
Q01QB2
SMBIQ01QB2
BIM_3.BI3
O:502|T:3|I:3
O:1098|T:3|I:3
O:5331|T:100|I:2
Q01QB2[2]_OPENPOS
Q01QB2[2]_CLOSEPOS
Main Application: FeederControl, Page: 1
Project
Interlocking
Interlocking.Substation.Voltage Level.Bay
Responsible department
ABB Ltd.
Technical reference
Created by
Approved by
Document kind
Graphical Application Configuration
Title
Q01
Doc. designation
Q01
Document id.
Rev.
0
Rel. date
22.05.2024
Lang.
en
3 / 3
Rev.
Modification
Rel. date
Created by
Based on
Replacing

E. Q05 Application Configuration



GOOSE RECEIVED FROM Q01



Main Application: Goosecoupler, Page: 1

				Project: Interlocking	Responsible department: ABB Ltd.	Technical reference	Document kind: Graphical Application Configuration	Doc. designation: Q05
			Replacing	Interlocking.Substation.Voltage Level.Bay	ABB	Created by	Title: Q05	Document id.
Rev.	Modification	Rel. date	Created by	Based on		Approved by		Rev.: 0 / Rel. date: 22.05.2024 / Lang.: en / 2 / 2

F. Verification of IED Configuration and HW Connections

Using the CMC356 the positions of the configured switches are changed to match step four of Table 3.1. The earthing switch in the bus coupler bay is opened and the CBs and disconnectors of the bus coupler are closed. The QB2 disconnector of the feeder bay is also closed and the circuit breaker in the same bay is open. This is expected to result in a true assessment, meaning it is permissible for the DUT, QB1, to operate. This verification was done prior to using the automated test cases, and therefore, the statuses of the various switches were checked in PCM600 and IEDScout. In Figure 1 a view of both PCM600 and IEDScout during step four.

The PCM600 window is seen at the top of the figure and IEDScout is seen below. It is seen from the IEDScout view, that the switches are indeed aligned with step four of Table 3.1. This is easily confirmed by cross-referencing the addresses with those listed in Table 4.1. The PCM600 view reveals that the CILO LN matches the expected assessment by outputting a 1. This means that the DUT is allowed to operate during this specific configuration.

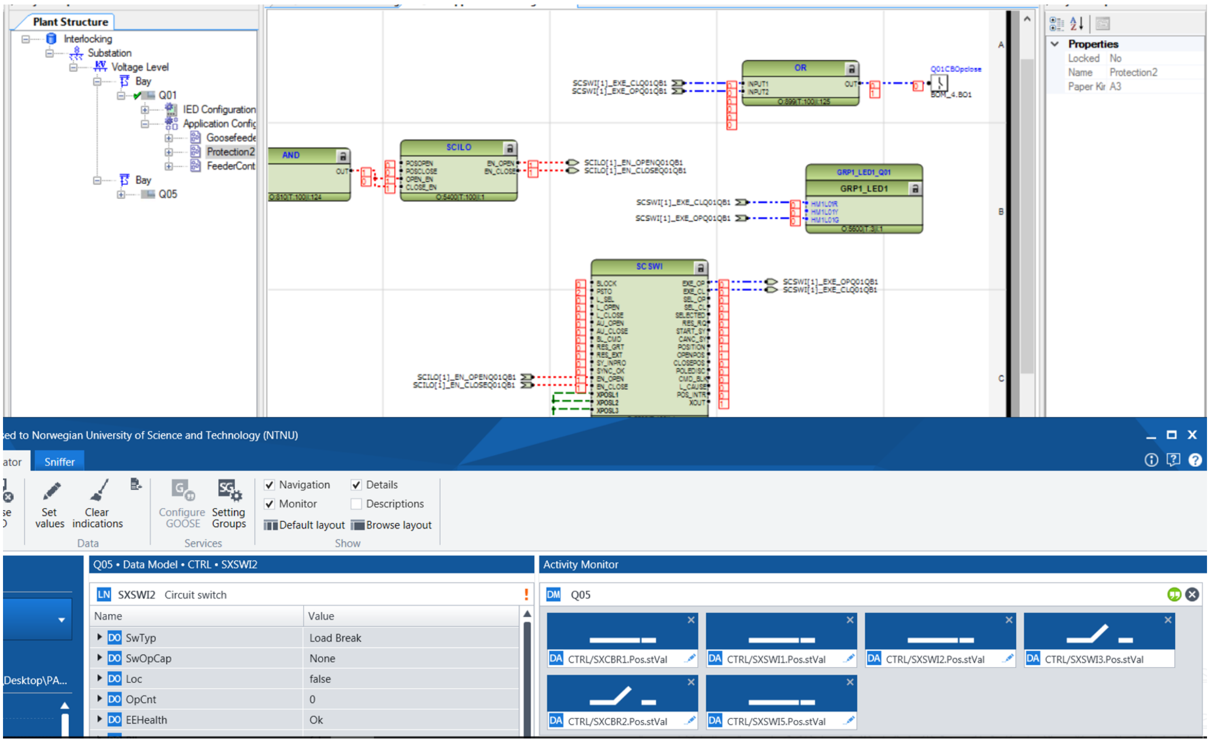


Figure 1.: Verification of the configuration of the IEDs and the HW connections in the laboratory using the CMC356. The PCM600 view is seen at the top and the IEDScout window at the bottom.

G. Generated JSON File Used for HW Test

Here, the JSON Test file generated and used for the HW test is seen. Note that Q05 is used to control the contacts of Q01 in addition to its own contacts. Q01QB1 is Q05CTRL/SXSWI4, Q01QA1 is Q05CTRL/SXCBR2, and Q01QB2 is Q05CTRL/SXSWI5.


```
  {
"version": "1.1",
"testCases": [
  {
    "name": "Lab_test_1",
    "autoSetControlValues": true,
    "autoAssess": true,
    "assessmentLockoutTime": 1.5,
    "autoAssessTimeout": 1.5,
    "switchOperationTime": 1.5,
    "parent": "Q01",
    "signalGroups": [
      {
        "groupType": "CONTROL",
        "signalRefs": [
          "Q05CTRL/SXCBR2.Pos.stVal",
          "Q05CTRL/SXSWI5.Pos.stVal",
          "Q05CTRL/SXCBR1.Pos.stVal",
          "Q05CTRL/SXSWI1.Pos.stVal",
          "Q05CTRL/SXSWI2.Pos.stVal",
          "Q05CTRL/SXSWI3.Pos.stVal"
        ]
      },
      {
        "groupType": "ASSESS",
        "signalRefs": [
          "Q01CTRL/SCILO1.EnaCls.stVal"
        ]
      },
      {
        "groupType": "COMMAND",
        "signalRefs": [
          "Q05CTRL/SXSWI4.Pos.stVal"
        ]
      }
    ],
    "testSteps": [
      {
        "description": "",
        "ordered": false,
        "expected": [
          {
            "signalRef": "Q05CTRL/SXCBR2.Pos.stVal",
            "value": "POS_OFF"
          },
          {
            "signalRef": "Q05CTRL/SXSWI5.Pos.stVal",
            "value": "POS_OFF"
          },
          {
            "signalRef": "Q05CTRL/SXCBR1.Pos.stVal",
            "value": "POS_OFF"
          },
          {
            "signalRef": "Q05CTRL/SXSWI1.Pos.stVal",
            "value": "POS_OFF"
```

```
    },
    {
      "signalRef": "Q05CTRL/SXSWI2.Pos.stVal",
      "value": "POS_OFF"
    },
    {
      "signalRef": "Q05CTRL/SXSWI3.Pos.stVal",
      "value": "POS_OFF"
    },
    {
      "signalRef": "Q01CTRL/SCILO1.EnaCls.stVal",
      "value": true
    },
    {
      "signalRef": "Q05CTRL/SXSWI4.Pos.stVal",
      "commandResult": "CAR_POSITION_CHANGED"
    }
  ]
},
{
  "description": "",
  "ordered": true,
  "expected": [
    {
      "signalRef": "Q05CTRL/SXSWI1.Pos.stVal",
      "value": "POS_OFF"
    },
    {
      "signalRef": "Q05CTRL/SXSWI2.Pos.stVal",
      "value": "POS_OFF"
    },
    {
      "signalRef": "Q05CTRL/SXSWI3.Pos.stVal",
      "value": "POS_OFF"
    },
    {
      "signalRef": "Q05CTRL/SXSWI5.Pos.stVal",
      "value": "POS_OFF"
    },
    {
      "signalRef": "Q05CTRL/SXCBR1.Pos.stVal",
      "value": "POS_OFF"
    },
    {
      "signalRef": "Q05CTRL/SXCBR2.Pos.stVal",
      "value": "POS_ON"
    },
    {
      "signalRef": "Q01CTRL/SCILO1.EnaCls.stVal",
      "value": false
    },
    {
      "signalRef": "Q05CTRL/SXSWI4.Pos.stVal",
      "commandResult": "CAR_BLOCKED_BY_INTERLOCKING"
    }
  ]
},
{
  "description": "",
  "ordered": true,
  "expected": [
    {
      "signalRef": "Q05CTRL/SXCBR1.Pos.stVal",
```

```
        "value": "POS_OFF"
      },
      {
        "signalRef": "Q05CTRL/SXCBR2.Pos.stVal",
        "value": "POS_OFF"
      },
      {
        "signalRef": "Q05CTRL/SXSWI1.Pos.stVal",
        "value": "POS_OFF"
      },
      {
        "signalRef": "Q05CTRL/SXSWI2.Pos.stVal",
        "value": "POS_OFF"
      },
      {
        "signalRef": "Q05CTRL/SXSWI3.Pos.stVal",
        "value": "POS_OFF"
      },
      {
        "signalRef": "Q05CTRL/SXSWI5.Pos.stVal",
        "value": "POS_OFF"
      },
      {
        "signalRef": "Q01CTRL/SCILO1.EnaCls.stVal",
        "value": true
      },
      {
        "signalRef": "Q05CTRL/SXSWI4.Pos.stVal",
        "commandResult": "CAR_NO_OPERATION"
      }
    ]
  },
  {
    "description": "",
    "ordered": true,
    "expected": [
      {
        "signalRef": "Q05CTRL/SXCBR2.Pos.stVal",
        "value": "POS_OFF"
      },
      {
        "signalRef": "Q05CTRL/SXSWI5.Pos.stVal",
        "value": "POS_ON"
      },
      {
        "signalRef": "Q05CTRL/SXCBR1.Pos.stVal",
        "value": "POS_OFF"
      },
      {
        "signalRef": "Q05CTRL/SXSWI1.Pos.stVal",
        "value": "POS_OFF"
      },
      {
        "signalRef": "Q05CTRL/SXSWI2.Pos.stVal",
        "value": "POS_OFF"
      },
      {
        "signalRef": "Q05CTRL/SXSWI3.Pos.stVal",
        "value": "POS_OFF"
      },
      {
        "signalRef": "Q01CTRL/SCILO1.EnaCls.stVal",
        "value": false
```

```
      },
      {
        "signalRef": "Q05CTRL/SXSWI4.Pos.stVal",
        "commandResult": "CAR_NO_OPERATION"
      }
    ]
  },
  {
    "description": "",
    "ordered": true,
    "expected": [
      {
        "signalRef": "Q05CTRL/SXCBR2.Pos.stVal",
        "value": "POS_OFF"
      },
      {
        "signalRef": "Q05CTRL/SXSWI5.Pos.stVal",
        "value": "POS_ON"
      },
      {
        "signalRef": "Q05CTRL/SXCBR1.Pos.stVal",
        "value": "POS_ON"
      },
      {
        "signalRef": "Q05CTRL/SXSWI1.Pos.stVal",
        "value": "POS_ON"
      },
      {
        "signalRef": "Q05CTRL/SXSWI2.Pos.stVal",
        "value": "POS_ON"
      },
      {
        "signalRef": "Q05CTRL/SXSWI3.Pos.stVal",
        "value": "POS_OFF"
      },
      {
        "signalRef": "Q01CTRL/SCILO1.EnaCls.stVal",
        "value": true
      },
      {
        "signalRef": "Q05CTRL/SXSWI4.Pos.stVal",
        "commandResult": "CAR_NO_OPERATION"
      }
    ]
  },
  {
    "description": "",
    "ordered": true,
    "expected": [
      {
        "signalRef": "Q05CTRL/SXCBR2.Pos.stVal",
        "value": "POS_OFF"
      },
      {
        "signalRef": "Q05CTRL/SXSWI5.Pos.stVal",
        "value": "POS_ON"
      },
      {
        "signalRef": "Q05CTRL/SXCBR1.Pos.stVal",
        "value": "POS_OFF"
      },
      {
        "signalRef": "Q05CTRL/SXSWI1.Pos.stVal",
```

```
        "value": "POS_ON"
      },
      {
        "signalRef": "Q05CTRL/SXSWI2.Pos.stVal",
        "value": "POS_ON"
      },
      {
        "signalRef": "Q05CTRL/SXSWI3.Pos.stVal",
        "value": "POS_OFF"
      },
      {
        "signalRef": "Q01CTRL/SCILO1.EnaCls.stVal",
        "value": false
      },
      {
        "signalRef": "Q05CTRL/SXSWI4.Pos.stVal",
        "commandResult": "CAR_NO_OPERATION"
      }
    ]
  },
  {
    "description": "",
    "ordered": true,
    "expected": [
      {
        "signalRef": "Q05CTRL/SXCBR2.Pos.stVal",
        "value": "POS_OFF"
      },
      {
        "signalRef": "Q05CTRL/SXSWI5.Pos.stVal",
        "value": "POS_OFF"
      },
      {
        "signalRef": "Q05CTRL/SXCBR1.Pos.stVal",
        "value": "POS_OFF"
      },
      {
        "signalRef": "Q05CTRL/SXSWI1.Pos.stVal",
        "value": "POS_OFF"
      },
      {
        "signalRef": "Q05CTRL/SXSWI2.Pos.stVal",
        "value": "POS_OFF"
      },
      {
        "signalRef": "Q05CTRL/SXSWI3.Pos.stVal",
        "value": "POS_ON"
      },
      {
        "signalRef": "Q01CTRL/SCILO1.EnaCls.stVal",
        "value": false
      },
      {
        "signalRef": "Q05CTRL/SXSWI4.Pos.stVal",
        "commandResult": "CAR_NO_OPERATION"
      }
    ]
  },
  {
    "description": "",
    "ordered": true,
    "expected": [
      {
```

```
                "signalRef": "Q05CTRL/SXCBR2.Pos.stVal",
                "value": "POS_OFF"
              },
              {
                "signalRef": "Q05CTRL/SXSWI5.Pos.stVal",
                "value": "POS_OFF"
              },
              {
                "signalRef": "Q05CTRL/SXCBR1.Pos.stVal",
                "value": "POS_OFF"
              },
              {
                "signalRef": "Q05CTRL/SXSWI1.Pos.stVal",
                "value": "POS_OFF"
              },
              {
                "signalRef": "Q05CTRL/SXSWI2.Pos.stVal",
                "value": "POS_OFF"
              },
              {
                "signalRef": "Q05CTRL/SXSWI3.Pos.stVal",
                "value": "POS_OFF"
              },
              {
                "signalRef": "Q01CTRL/SCILO1.EnaCls.stVal",
                "value": true
              },
              {
                "signalRef": "Q05CTRL/SXSWI4.Pos.stVal",
                "commandResult": "CAR_NO_OPERATION"
              }
            ]
          }
        ]
      }
    ]
  }
```

H. Expanded JSON Test Case

Here, the expanded json file is seen, where an additional earthing switch has been added in bay Q01 and an additional step has been added at the end of the test, where only this new earthing switch is closed.


```json
  {
"version": "1.1",
"testCases": [
  {
    "name": "Expanded_Test_Case",
    "autoSetControlValues": true,
    "autoAssess": true,
    "assessmentLockoutTime": 1.5,
    "autoAssessTimeout": 1.5,
    "switchOperationTime": 1.5,
    "parent": "AA1D1Q01Q1",
    "signalGroups": [
      {
        "groupType": "CONTROL",
        "signalRefs": [
          "AA1D1Q01Q1QA1/CSWI1.Pos",
          "AA1D1Q01Q1QB2/CSWI1.Pos",
          "AA1D1Q01Q1QC9/CSWI1.Pos",
          "AA1D1Q05Q1CBSW/XCBR1.Pos.stVal",
          "AA1D1Q05Q1CBSW/XSWI1.Pos.stVal",
          "AA1D1Q05Q1CBSW/XSWI2.Pos.stVal",
          "AA1D1Q05Q1CBSW/XSWI3.Pos.stVal"
        ]
      },
      {
        "groupType": "ASSESS",
        "signalRefs": [
          "AA1D1Q01Q1QB1/CILO1.EnaCls"
        ]
      },
      {
        "groupType": "COMMAND",
        "signalRefs": [
          "AA1D1Q01Q1QB1/CSWI1.Pos"
        ]
      }
    ],
    "testSteps": [
      {
        "description": "",
        "ordered": false,
        "expected": [
          {
            "signalRef": "AA1D1Q01Q1QA1/CSWI1.Pos",
            "value": "POS_OFF"
          },
          {
            "signalRef": "AA1D1Q01Q1QB2/CSWI1.Pos",
            "value": "POS_OFF"
          },
          {
            "signalRef": "AA1D1Q01Q1QC9/CSWI1.Pos",
            "value": "POS_OFF"
          },
          {
            "signalRef": "AA1D1Q05Q1CBSW/XCBR1.Pos.stVal",
```

```
        "value": "POS_OFF"
      },
      {
        "signalRef": "AA1D1Q05Q1CBSW/XSWI1.Pos.stVal",
        "value": "POS_OFF"
      },
      {
        "signalRef": "AA1D1Q05Q1CBSW/XSWI2.Pos.stVal",
        "value": "POS_OFF"
      },
      {
        "signalRef": "AA1D1Q05Q1CBSW/XSWI3.Pos.stVal",
        "value": "POS_OFF"
      },
      {
        "signalRef": "AA1D1Q01Q1QB1/CILO1.EnaCls",
        "value": true
      },
      {
        "signalRef": "AA1D1Q01Q1QB1/CSWI1.Pos",
        "commandResult": "CAR_POSITION_CHANGED"
      }
    ]
  },
  {
    "description": "",
    "ordered": true,
    "expected": [
      {
        "signalRef": "AA1D1Q01Q1QA1/CSWI1.Pos",
        "value": "POS_ON"
      },
      {
        "signalRef": "AA1D1Q01Q1QB2/CSWI1.Pos",
        "value": "POS_OFF"
      },
      {
        "signalRef": "AA1D1Q01Q1QC9/CSWI1.Pos",
        "value": "POS_OFF"
      },
      {
        "signalRef": "AA1D1Q05Q1CBSW/XCBR1.Pos.stVal",
        "value": "POS_OFF"
      },
      {
        "signalRef": "AA1D1Q05Q1CBSW/XSWI1.Pos.stVal",
        "value": "POS_OFF"
      },
      {
        "signalRef": "AA1D1Q05Q1CBSW/XSWI2.Pos.stVal",
        "value": "POS_OFF"
      },
      {
        "signalRef": "AA1D1Q05Q1CBSW/XSWI3.Pos.stVal",
        "value": "POS_OFF"
      },
      {
        "signalRef": "AA1D1Q01Q1QB1/CILO1.EnaCls",
        "value": false
      },
      {
        "signalRef": "AA1D1Q01Q1QB1/CSWI1.Pos",
        "commandResult": "CAR_BLOCKED_BY_INTERLOCKING"
```

```
    }
  ]
},
{
  "description": "",
  "ordered": true,
  "expected": [
    {
      "signalRef": "AA1D1Q01Q1QA1/CSWI1.Pos",
      "value": "POS_OFF"
    },
    {
      "signalRef": "AA1D1Q01Q1QB2/CSWI1.Pos",
      "value": "POS_OFF"
    },
    {
      "signalRef": "AA1D1Q01Q1QC9/CSWI1.Pos",
      "value": "POS_OFF"
    },
    {
      "signalRef": "AA1D1Q05Q1CBSW/XCBR1.Pos.stVal",
      "value": "POS_OFF"
    },
    {
      "signalRef": "AA1D1Q05Q1CBSW/XSWI1.Pos.stVal",
      "value": "POS_OFF"
    },
    {
      "signalRef": "AA1D1Q05Q1CBSW/XSWI2.Pos.stVal",
      "value": "POS_OFF"
    },
    {
      "signalRef": "AA1D1Q05Q1CBSW/XSWI3.Pos.stVal",
      "value": "POS_OFF"
    },
    {
      "signalRef": "AA1D1Q01Q1QB1/CILO1.EnaCls",
      "value": true
    },
    {
      "signalRef": "AA1D1Q01Q1QB1/CSWI1.Pos",
      "commandResult": "CAR_NO_OPERATION"
    }
  ]
},
{
  "description": "",
  "ordered": true,
  "expected": [
    {
      "signalRef": "AA1D1Q01Q1QA1/CSWI1.Pos",
      "value": "POS_OFF"
    },
    {
      "signalRef": "AA1D1Q01Q1QB2/CSWI1.Pos",
      "value": "POS_ON"
    },
    {
      "signalRef": "AA1D1Q01Q1QC9/CSWI1.Pos",
      "value": "POS_ON"
    },
    {
      "signalRef": "AA1D1Q05Q1CBSW/XCBR1.Pos.stVal",
```

```
        "value": "POS_OFF"
      },
      {
        "signalRef": "AA1D1Q05Q1CBSW/XSWI1.Pos.stVal",
        "value": "POS_OFF"
      },
      {
        "signalRef": "AA1D1Q05Q1CBSW/XSWI2.Pos.stVal",
        "value": "POS_OFF"
      },
      {
        "signalRef": "AA1D1Q05Q1CBSW/XSWI3.Pos.stVal",
        "value": "POS_OFF"
      },
      {
        "signalRef": "AA1D1Q01Q1QB1/CILO1.EnaCls",
        "value": false
      },
      {
        "signalRef": "AA1D1Q01Q1QB1/CSWI1.Pos",
        "commandResult": "CAR_NO_OPERATION"
      }
    ]
  },


  {
    "description": "",
    "ordered": true,
    "expected": [
      {
        "signalRef": "AA1D1Q05Q1CBSW/XSWI1.Pos.stVal",
        "value": "POS_ON"
      },
      {
        "signalRef": "AA1D1Q05Q1CBSW/XSWI2.Pos.stVal",
        "value": "POS_ON"
      },
      {
        "signalRef": "AA1D1Q05Q1CBSW/XSWI3.Pos.stVal",
        "value": "POS_OFF"
      },
      {
        "signalRef": "AA1D1Q01Q1QA1/CSWI1.Pos",
        "value": "POS_OFF"
      },
      {
        "signalRef": "AA1D1Q01Q1QB2/CSWI1.Pos",
        "value": "POS_ON"
      },
      {
        "signalRef": "AA1D1Q01Q1QC9/CSWI1.Pos",
        "value": "POS_OFF"
      },
      {
        "signalRef": "AA1D1Q05Q1CBSW/XCBR1.Pos.stVal",
        "value": "POS_ON"
      },
      {
        "signalRef": "AA1D1Q01Q1QB1/CILO1.EnaCls",
        "value": true
      },
      {
        "signalRef": "AA1D1Q01Q1QB1/CSWI1.Pos",
```

```
      "commandResult": "CAR_NO_OPERATION"
    }
  ]
},

{
  "description": "",
  "ordered": true,
  "expected": [
    {
      "signalRef": "AA1D1Q01Q1QA1/CSWI1.Pos",
      "value": "POS_OFF"
    },
    {
      "signalRef": "AA1D1Q01Q1QB2/CSWI1.Pos",
      "value": "POS_ON"
    },
    {
      "signalRef": "AA1D1Q01Q1QC9/CSWI1.Pos",
      "value": "POS_OFF"
    },
    {
      "signalRef": "AA1D1Q05Q1CBSW/XCBR1.Pos.stVal",
      "value": "POS_OFF"
    },
    {
      "signalRef": "AA1D1Q05Q1CBSW/XSWI1.Pos.stVal",
      "value": "POS_ON"
    },
    {
      "signalRef": "AA1D1Q05Q1CBSW/XSWI2.Pos.stVal",
      "value": "POS_ON"
    },
    {
      "signalRef": "AA1D1Q05Q1CBSW/XSWI3.Pos.stVal",
      "value": "POS_OFF"
    },
    {
      "signalRef": "AA1D1Q01Q1QB1/CILO1.EnaCls",
      "value": false
    },
    {
      "signalRef": "AA1D1Q01Q1QB1/CSWI1.Pos",
      "commandResult": "CAR_NO_OPERATION"
    }
  ]
},
{
  "description": "",
  "ordered": true,
  "expected": [
    {
      "signalRef": "AA1D1Q01Q1QA1/CSWI1.Pos",
      "value": "POS_OFF"
    },
    {
      "signalRef": "AA1D1Q01Q1QB2/CSWI1.Pos",
      "value": "POS_OFF"
    },
    {
      "signalRef": "AA1D1Q01Q1QC9/CSWI1.Pos",
      "value": "POS_OFF"
    },
```

```
    {
      "signalRef": "AA1D1Q05Q1CBSW/XCBR1.Pos.stVal",
      "value": "POS_OFF"
    },
    {
      "signalRef": "AA1D1Q05Q1CBSW/XSWI1.Pos.stVal",
      "value": "POS_OFF"
    },
    {
      "signalRef": "AA1D1Q05Q1CBSW/XSWI2.Pos.stVal",
      "value": "POS_OFF"
    },
    {
      "signalRef": "AA1D1Q05Q1CBSW/XSWI3.Pos.stVal",
      "value": "POS_ON"
    },
    {
      "signalRef": "AA1D1Q01Q1QB1/CILO1.EnaCls",
      "value": false
    },
    {
      "signalRef": "AA1D1Q01Q1QB1/CSWI1.Pos",
      "commandResult": "CAR_NO_OPERATION"
    }
  ]
},
```

```
{
  "description": "",
  "ordered": true,
  "expected": [
    {
      "signalRef": "AA1D1Q01Q1QA1/CSWI1.Pos",
      "value": "POS_OFF"
    },
    {
      "signalRef": "AA1D1Q01Q1QB2/CSWI1.Pos",
      "value": "POS_OFF"
    },
    {
      "signalRef": "AA1D1Q01Q1QC9/CSWI1.Pos",
      "value": "POS_OFF"
    },
    {
      "signalRef": "AA1D1Q05Q1CBSW/XCBR1.Pos.stVal",
      "value": "POS_OFF"
    },
    {
      "signalRef": "AA1D1Q05Q1CBSW/XSWI1.Pos.stVal",
      "value": "POS_OFF"
    },
    {
      "signalRef": "AA1D1Q05Q1CBSW/XSWI2.Pos.stVal",
      "value": "POS_OFF"
    },
    {
      "signalRef": "AA1D1Q05Q1CBSW/XSWI3.Pos.stVal",
      "value": "POS_OFF"
    },
    {
      "signalRef": "AA1D1Q01Q1QB1/CILO1.EnaCls",
```

```
            "value": true
          },
          {
            "signalRef": "AA1D1Q01Q1QB1/CSWI1.Pos",
            "commandResult": "CAR_NO_OPERATION"
          }
        ]
      },
```


Test step added from the case expansion:


```
      {
        "description": "",
        "ordered": true,
        "expected": [
          {
            "signalRef": "AA1D1Q01Q1QA1/CSWI1.Pos",
            "value": "POS_OFF"
          },
          {
            "signalRef": "AA1D1Q01Q1QB2/CSWI1.Pos",
            "value": "POS_OFF"
          },
          {
            "signalRef": "AA1D1Q01Q1QC9/CSWI1.Pos",
            "value": "POS_ON"
          },
                      {
            "signalRef": "AA1D1Q05Q1CBSW/XCBR1.Pos.stVal",
            "value": "POS_OFF"
          },
          {
            "signalRef": "AA1D1Q05Q1CBSW/XSWI1.Pos.stVal",
            "value": "POS_OFF"
          },
          {
            "signalRef": "AA1D1Q05Q1CBSW/XSWI2.Pos.stVal",
            "value": "POS_OFF"
          },
          {
            "signalRef": "AA1D1Q05Q1CBSW/XSWI3.Pos.stVal",
            "value": "POS_OFF"
          },
          {
            "signalRef": "AA1D1Q01Q1QB1/CILO1.EnaCls",
            "value": false
          },
          {
            "signalRef": "AA1D1Q01Q1QB1/CSWI1.Pos",
            "commandResult": "CAR_NO_OPERATION"
          }
        ]
      }
    ]
  }
 ]
}
```

I. Interlocking Test Case Results Report

Project: Project1
Creation Date/Time: 2024-05-26 17:20:38.934+02:00
Test System:
Device: Interlocking Test 21/05 OM8412C7
Software Version: 2.40.0071
Files: Interlocking15052024.scd

Test Results

Q01 - ABB 670 Series

Test Case: Lab_test_1 – Passed
Test Type: Logic
Automated Control: On
Automated Assessment: On
Step 1 – 2024-05-21 18:06:40.093+02:00 – Passed

Control

Signal	Timestamp	Value	Expected Value
Q05 QA1 – Q05CTRL/SXCBR1.Pos	2024-05-21 18:06:12.234+02:00	Open	Open
Command to Q01 QA1 – Q05CTRL/SXCBR2.Pos	2024-05-21 18:06:07.563+02:00	Open	Open
Q05 QB1 – Q05CTRL/SXSWI1.Pos	2024-05-21 18:06:13.795+02:00	Open	Open
Q05 QB2 – Q05CTRL/SXSWI2.Pos	2024-05-21 18:06:13.808+02:00	Open	Open
Q05 QC11 – Q05CTRL/SXSWI3.Pos	2024-05-21 18:06:15.375+02:00	Open	Open
Command to Q01 QB2 – Q05CTRL/SXSWI5.Pos	2024-05-21 18:06:13.791+02:00	Open	Open

Signal Assessment

Signal	Timestamp	Value	Expected Value
Close operation at open or interm. or bad pos. is enabled – Q01CTRL/SCILO1.EnaCls	2024-04-28 20:47:44.081+02:00	True	True

Command Assessment

Signal	Timestamp	Value	Expected Value
Command to Q01 QB1 – Q05CTRL/SXSWI4.Pos	2024-05-21 18:06:40.090+02:00	Position-changed	Position-changed

Step 2 – 2024-05-21 18:06:41.631+02:00 – Passed

Control

Signal	Timestamp	Value	Expected Value
Q05 QA1 – Q05CTRL/SXCBR1.Pos	2024-05-21 18:06:12.234+02:00	Open	Open
Command to Q01 QA1 – Q05CTRL/SXCBR2.Pos	2024-05-21 18:06:40.111+02:00	Closed	Closed
Q05 QB1 – Q05CTRL/SXSWI1.Pos	2024-05-21 18:06:13.795+02:00	Open	Open
Q05 QB2 – Q05CTRL/SXSWI2.Pos	2024-05-21 18:06:13.808+02:00	Open	Open
Q05 QC11 – Q05CTRL/SXSWI3.Pos	2024-05-21 18:06:15.375+02:00	Open	Open
Command to Q01 – QB2Q05CTRL/SXSWI5.Pos	2024-05-21 18:06:13.791+02:00	Open	Open

Signal Assessment

Signal	Timestamp	Value	Expected Value
Close operation at open or interm. or bad pos. is enabled – Q01CTRL/SCILO1.EnaCls	2024-04-28 20:48:08.781+02:00	False	False

Command Assessment

Signal	Timestamp	Value	Expected Value
Command to Q01 QB1Q05CTRL/SXSWI4.Pos	2024-05-21 18:06:41.630+02:00	Blocked-by-interlocking	Blocked-by-interlocking

Step 3 – 2024-05-21 18:06:43.174+02:00 – Passed

Control

Signal	Timestamp	Value	Expected Value
Q05 QA1 – Q05CTRL/SXCBR1.Pos	2024-05-21 18:06:12.234+02:00	Open	Open
Command to Q01 QA1 – Q05CTRL/SXCBR2.Pos	2024-05-21 18:06:41.650+02:00	Open	Open
Q05 QB1 – Q05CTRL/SXSWI1.Pos	2024-05-21 18:06:13.795+02:00	Open	Open
Q05 QB2 – Q05CTRL/SXSWI2.Pos	2024-05-21 18:06:13.808+02:00	Open	Open
Q05 QC11 – Q05CTRL/SXSWI3.Pos	2024-05-21 18:06:15.375+02:00	Open	Open
Command to Q01 QB2 – Q05CTRL/SXSWI5.Pos	2024-05-21 18:06:13.791+02:00	Open	Open

Signal Assessment

Signal	Timestamp	Value	Expected Value
Close operation at open or interm. or bad pos. is enabled – Q01CTRL/SCILO1.EnaCls	2024-04-28 20:48:10.381+02:00	True	True

Command Assessment

Signal	Timestamp	Value	Expected Value
Command to Q01 QB1 – Q05CTRL/SXSWI4.Pos	2024-05-21 18:06:43.169+02:00	No-operation	No-operation

Step 4 – 2024-05-21 18:06:44.721+02:00 – Passed

Control

Signal	Timestamp	Value	Expected Value
Q05 QA1 – Q05CTRL/SXCBR1.Pos	2024-05-21 18:06:12.234+02:00	Open	Open
Command to Q01 QA1 – Q05CTRL/SXCBR2.Pos	2024-05-21 18:06:41.650+02:00	Open	Open
Q05 QB1 – Q05CTRL/SXSWI1.Pos	2024-05-21 18:06:13.795+02:00	Open	Open
Q05 QB2 – Q05CTRL/SXSWI2.Pos	2024-05-21 18:06:13.808+02:00	Open	Open
Q05 QC11 – Q05CTRL/SXSWI3.Pos	2024-05-21 18:06:15.375+02:00	Open	Open
Command to Q01 QB2 – Q05CTRL/SXSWI5.Pos	2024-05-21 18:06:43.200+02:00	Closed	Closed

Signal Assessment

Signal	Timestamp	Value	Expected Value
Close operation at open or interm. or bad pos. is enabled – Q01CTRL/SCILO1.EnaCls	2024-04-28 20:48:11.981+02:00	False	False

Command Assessment

Signal	Timestamp	Value	Expected Value
Command to Q01 QB1 – Q05CTRL/SXSWI4.Pos	2024-05-21 18:06:44.718+02:00	No-operation	No-operation

Step 5 – 2024-05-21 18:06:46.300+02:00 – Passed

Control

Signal	Timestamp	Value	Expected Value
Q05 QA1 – Q05CTRL/SXCBR1.Pos	2024-05-21 18:06:44.751+02:00	Closed	Closed
Command to Q01 QA1 – Q05CTRL/SXCBR2.Pos	2024-05-21 18:06:41.650+02:00	Open	Open
Q05 QB1 – Q05CTRL/SXSWI1.Pos	2024-05-21 18:06:44.759+02:00	Closed	Closed
Q05 QB2 – Q05CTRL/SXSWI2.Pos	2024-05-21 18:06:44.767+02:00	Closed	Closed
Q05 QC11 – Q05CTRL/SXSWI3.Pos	2024-05-21 18:06:15.375+02:00	Open	Open
Command to Q01 QB2 – Q05CTRL/SXSWI5.Pos	2024-05-21 18:06:43.200+02:00	Closed	Closed

Signal Assessment

Signal	Timestamp	Value	Expected Value
Close operation at open or interm. or bad pos. is enabled – Q01CTRL/SCILO1.EnaCls	2024-04-28 20:48:13.481+02:00	True	True

Command Assessment

Signal	Timestamp	Value	Expected Value
Command to Q01 QB1 – Q05CTRL/SXSWI4.Pos	2024-05-21 18:06:46.298+02:00	No-operation	No-operation

Step 6 – 2024-05-21 18:06:47.841+02:00 – Passed

Control

Signal	Timestamp	Value	Expected Value
Q05 QA1 – Q05CTRL/SXCBR1.Pos	2024-05-21 18:06:46.322+02:00	Open	Open
Command to Q01 QA1 – Q05CTRL/SXCBR2.Pos	2024-05-21 18:06:41.650+02:00	Open	Open
Q05 QB1 – Q05CTRL/SXSWI1.Pos	2024-05-21 18:06:44.759+02:00	Closed	Closed
Q05 QB2 – Q05CTRL/SXSWI2.Pos	2024-05-21 18:06:44.767+02:00	Closed	Closed
Q05 QC11 – Q05CTRL/SXSWI3.Pos	2024-05-21 18:06:15.375+02:00	Open	Open
Command to Q01 QB2 – Q05CTRL/SXSWI5.Pos	2024-05-21 18:06:43.200+02:00	Closed	Closed

Signal Assessment

Signal	Timestamp	Value	Expected Value
Close operation at open or interm. or bad pos. is enabled – Q01CTRL/SCILO1.EnaCls	2024-04-28 20:48:14.981+02:00	False	False

Command Assessment

Signal	Timestamp	Value	Expected Value
Command to Q01 QB1 – Q05CTRL/SXSWI4.Pos	2024-05-21 18:06:47.839+02:00	No-operation	No-operation

Step 7 – 2024-05-21 18:06:49.433+02:00 – Passed

Control

Signal	Timestamp	Value	Expected Value
Q05 QA1 – Q05CTRL/SXCBR1.Pos	2024-05-21 18:06:46.322+02:00	Open	Open
Command to Q01 QA1 – Q05CTRL/SXCBR2.Pos	2024-05-21 18:06:41.650+02:00	Open	Open
Q05 QB1 – Q05CTRL/SXSWI1.Pos	2024-05-21 18:06:47.869+02:00	Open	Open
Q05 QB2 – Q05CTRL/SXSWI2.Pos	2024-05-21 18:06:47.873+02:00	Open	Open
Q05 QC11 – Q05CTRL/SXSWI3.Pos	2024-05-21 18:06:47.897+02:00	Closed	Closed
Command to Q01 QB2 – Q05CTRL/SXSWI5.Pos	2024-05-21 18:06:47.864+02:00	Open	Open

Signal Assessment

Signal	Timestamp	Value	Expected Value
Close operation at open or interm. or bad pos. is enabled – Q01CTRL/SCILO1.EnaCls	2024-04-28 20:48:14.981+02:00	False	False

Command Assessment

Signal	Timestamp	Value	Expected Value
Command to Q01 QB1 – Q05CTRL/SXSWI4.Pos	2024-05-21 18:06:49.429+02:00	No-operation	No-operation

Step 8 – 2024-05-21 18:06:50.986+02:00 – Passed

Control

Signal	Timestamp	Value	Expected Value
Q05 QA1 – Q05CTRL/SXCBR1.Pos	2024-05-21 18:06:46.322+02:00	Open	Open
Command to Q01 QA1 – Q05CTRL/SXCBR2.Pos	2024-05-21 18:06:41.650+02:00	Open	Open
Q05 QB1 – Q05CTRL/SXSWI1.Pos	2024-05-21 18:06:47.869+02:00	Open	Open
Q05 QB2 – Q05CTRL/SXSWI2.Pos	2024-05-21 18:06:47.873+02:00	Open	Open
Q05 QC11 – Q05CTRL/SXSWI3.Pos	2024-05-21 18:06:49.467+02:00	Open	Open
Command to Q01 QB2 – Q05CTRL/SXSWI5.Pos	2024-05-21 18:06:47.864+02:00	Open	Open

Signal Assessment

Signal	Timestamp	Value	Expected Value
Close operation at open or interm. or bad pos. is enabled – Q01CTRL/SCILO1.EnaCls	2024-04-28 20:48:18.181+02:00	True	True

Command Assessment

Signal	Timestamp	Value	Expected Value
Command to Q01 QB1 – Q05CTRL/SXSWI4.Pos	2024-05-21 18:06:50.983+02:00	No-operation	No-operation