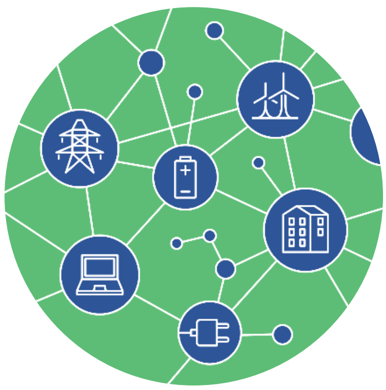
OSMOSE

OPTIMAL SYSTEM-MIX OF FLEXIBILITY SOLUTIONS FOR EUROPEAN ELECTRICITY

Execution of Demonstrator for Interoperability Framework

Deliverable 7.2



Contact: www.osmose-h2020.eu

The project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 773406

Document properties

Project Information

Programme	**O**ptimal **S**ystem-**M**ix **O**f Flexibility **S**olutions For **E**uropean Electricity
Project acronym	OSMOSE
Grant agreement number	773406
Number of the Deliverable	D7.2
WP/Task related	Task 7.1.2

Document information

Document Name	Execution of Demonstrator for Interoperability Framework
Date of delivery	23/03/2022
Status and Version	V3
Number of pages	135

Responsible

Document Responsible	Christoph Brunner (it4power)
Author(s)	Christoph Brunner (it4power)
Reviewer(s)	Nathalie Grisey (RTE) / Gorazd Azman (ELES)
Approver	Nathalie Grisey (RTE)

Dissemination Level

Type (distribution level)	☒ PU, Public ☐ CO – full consortium, Confidential, only for members of the consortium (including the Commission Services) ☐ CO – some partners, Confidential, only for some partners (list of partners to be defined)

Review History

Version	Date	Reviewer	Comment
V1.0	3.1.22	Christoph Brunner	Initial version distributed to participants
V1.1	4.2.22	Christoph Brunner	Integrated comments from Thomas
V2	4.3.22	Christoph Brunner	Added content for test of battery simulation
V3	23.3.22	Nathalie Grisey	Based on feedback received, added some content to the executive summary

Table of content

List of figures

List of tables

1 Executive summary

The Subtask 7.1.2 of OSMOSE aims at demonstrating the whole engineering process of the IEC 61850 ENTSO-E profile, through different specification tools and for implementation on products from different vendors.

This document describes the demonstrator setup as well as the activities that have been executed in the context of the demonstrator.

It describes the following.

In clause 3, the application scenario that has been used for the demonstrator is described. The application scenario is control and protection for a transmission line based on a real use case from ELIA. It includes communication between the two distance protections at both ends of the power line to implement a teleprotection scheme. For the tests, only a subset has been implemented.

Clause 4 describes the detailed setup of the demonstrator and the participating companies with their roles (IED manufacturer or tool supplier).

Clause 6 describes the functional testing that was done. Functional testing was done to verify that the engineering process was successful to configure the system correctly. Clause 7 describes the interoperability testing. Purpose of interoperability testing was, to prove that the enhanced concepts proposed by OSMOSE task 7.1 work and improve both interoperability as well as efficiency.

Clause 5 describes the concept of functional simulation. With the enhancements proposed by OSMOSE to the IEC 61850 engineering process, it is possible to do a functional simulation of the specification, in order to verify that the specification works as expected. That was verified in the OSMOSE project.

Finally, clause 8 describes a use case of a battery in a transmission substation. That was included in the demonstrator to verify the IEC 61850 models for DERs and storage. As the related standard (IEC 61850-7-420) was in development during the OSMOSE project, no IEDs implementing it were yet available. Therefore, we decided to implement that use case using simulation tools.

2 List of acronyms and abbreviations

In the table is listed the acronyms and abbreviations used in this document.

Acronym	Meaning
GOOSE	Generic object oriented system event
ICD	IED capability description
ICT	IED configuration tool
IEC	International electrotechnical commission
IED	Intelligent electronic device
ISD	IED specification description
LN	Logical Node
MMS	Manufacturing message specification – the layer 7 protocol used by IEC 61850 for client / server communication
SCD	System configuration description
SCT	System configuration tool
SED	System exchange description
SSD	System specification description
SST	System specification tool

3 Purpose of the demonstrator

In deliverable D 7.1, an enhanced engineering process with the aim to improve engineering efficiency and quality was introduced.

D 7.1 defined the following next steps:

1. Define the necessary SCL extensions required to execute the engineering process
2. Define the ENTSO-E profile enhancements and extension regarding gaps and energy storage
3. Implement this process in the participating engineering tools
4. Demonstrate it through a demonstrator setup
5. Provide input for standardization in IEC61850 WG10, in particular IEC61850-6-100 and IEC61850-7-4.

The Subtask 7.1.2 of OSMOSE aims to address items 3 and 4 of those next steps by implementing the necessary functionality in tools and products from different vendors and by building a demonstrator to verify the process and its implementation.

The purpose of the demonstrator is, to create a test environment based on a typical application scenario from a substation to allow the verification of the various steps defined in the engineering process. As the deployment in the IEDs that realize the application is a critical element, the verification of the functionality at the end is done as well. Therefore, the demonstrator realizes an implementation with real IEDs and a real time simulation of the switchgear and power flow.

The demonstrator related activities include the following steps:

- Use the application scenario to validate the specification and design of a project, including interoperability between the various tools.
    - The efficiency of the process is validated using the interoperability test specifications that have been developed as part of the OSMOSE based on the detailed engineering process described in [D7.3].
    - The result of the specification as well as of the SCD file from design can be validated using simulation tools.
- Implement the design using the IEDs and the real time simulator
    - The efficiency of the process is validated using the interoperability test specifications
    - The result is validated by functional tests that have been developed as part of the OSMOSE project, using the real time simulator to simulate fault conditions and other process behavior.

The application scenario behind the demonstrator does not only include a typical substation protection automation and control application – it includes as well a use case of a battery storage system. This has been added as part of the project, as a goal was as well to verify the IEC 61850 models for distributed energy resources.

4 The application scenario

4.1 Overview

The scenario consists of the usage of energy storage to optimize the load on a line. For that, on each side of the line, storage equipment is available. An overview of the functions is shown in Figure 1. Details for the line bay are discussed further in this chapter. The Application of the storage system is described in the deliverable 7.3 [D7.3] of the OSMOSE project.

Distance protection will be applied with a teleprotection scheme. The teleprotection scheme shall be realized conform to IEC 61850 either with 61850 compliant teleprotection equipment or with tunnelling of GOOSE messages.

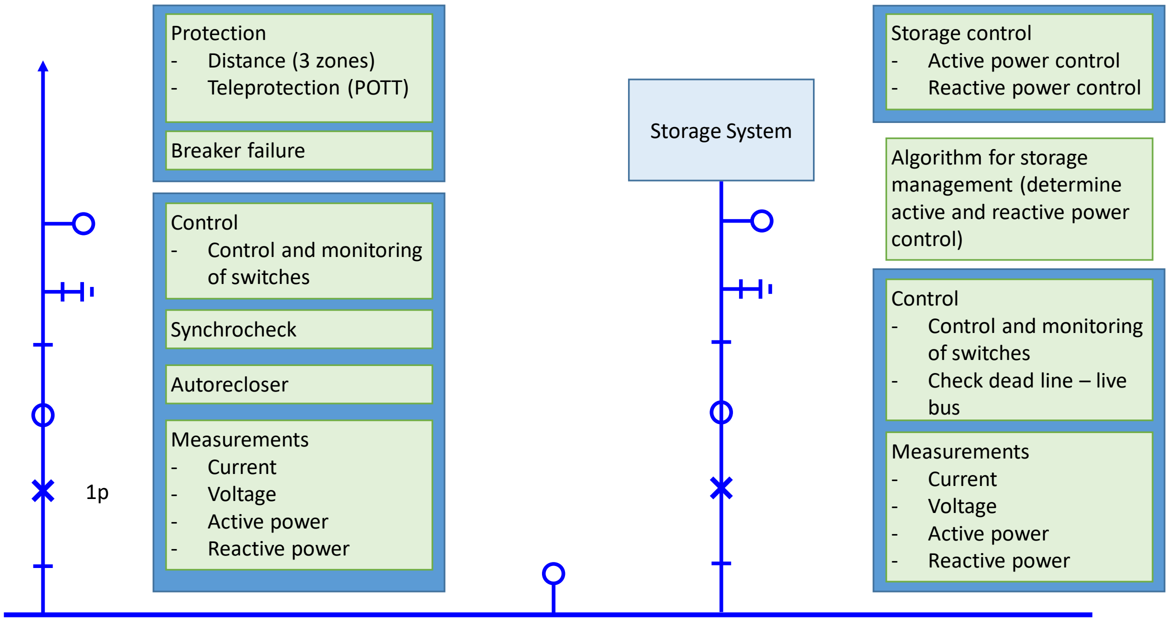


Figure 1: Overview of the functions in one substation

The overall scenario of the demonstrator is shown in Figure 2.

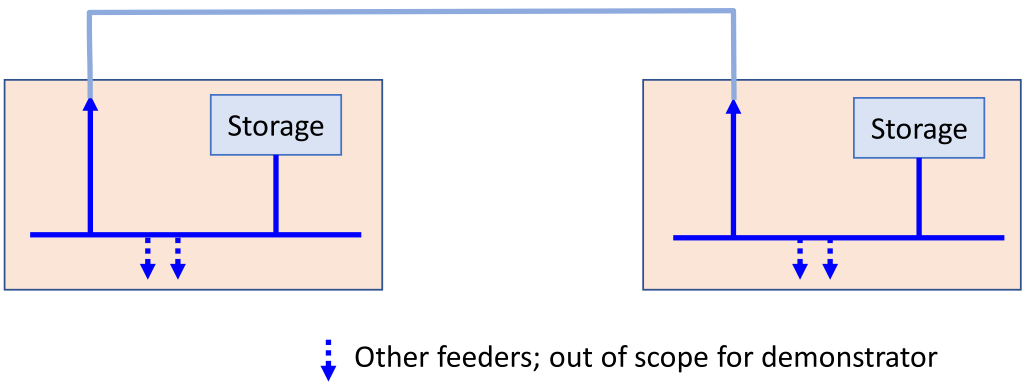


Figure 2: Overall scenario

For the demonstration and interoperability testing of the engineering process, a substation scenario was used rather than the storage scenario, as there were no devices yet available that would support the storage scenario. This, because Edition 2 of IEC 61850-7-420 which applies to DER and storage was in development during the OSMOSE project and has only been released in 2021.

4.2 Functional specification of the line bay

The functions and the function allocation of the line bay is shown in Figure 1. The line bay will have two IEDs:

- A protection device
    - Distance protection with three zones and single phase tripping capability
    - Teleprotection scheme overreach with transfer trip (POTT)
    - Switch on to fault protection
    - Breaker failure protection.
- A bay controller
    - Control and monitoring of the breaker and the switches
    - Synchrocheck
    - Autoreclosing (single phase and three phase autoreclose function)
    - Measurements

NOTE: We are aware that the function allocation may not be in line with all protection philosophies, but for the purpose of the project, which is to demonstrate interoperability, we limited the number of devices.

With the above function allocation, at least the following interactions shall be implemented using GOOSE messaging (note that this is not necessarily a complete list):

- Autorecloser initiate from protection device to bay controller
- External trip from breaker failure function to bay controller of storage bay
- Information about close control from bay controller to initiate switch on to fault protection

Teleprotection will be implemented using the tunnelling approach as described in IEC 61850-90-1. The GOOSE messages are directly exchanged with the protection at the other side of the line.

A detailed bloc diagram is provided in Annex B.

4.3 Subsets used for the demonstrator activities

As the focus of the activity was on optimizing the engineering interoperability and efficiency, it was decided to only implement a subset of the functionality for the testing.

The functionality of the first subset is shown in Figure 3 with red colour. It included distance protection function with two zones, breaker control with switch on to fault protection, synchrocheck and auto reclosing. That subset was limited to the communication within one substation only, so the SCL file exchange between projects (SED file according to IEC 61850-6) was not yet used for that subset. Subset 2 consisted of subset 1 and the parts in blue colour. This added the teleprotection scheme and as such communication between two substations.

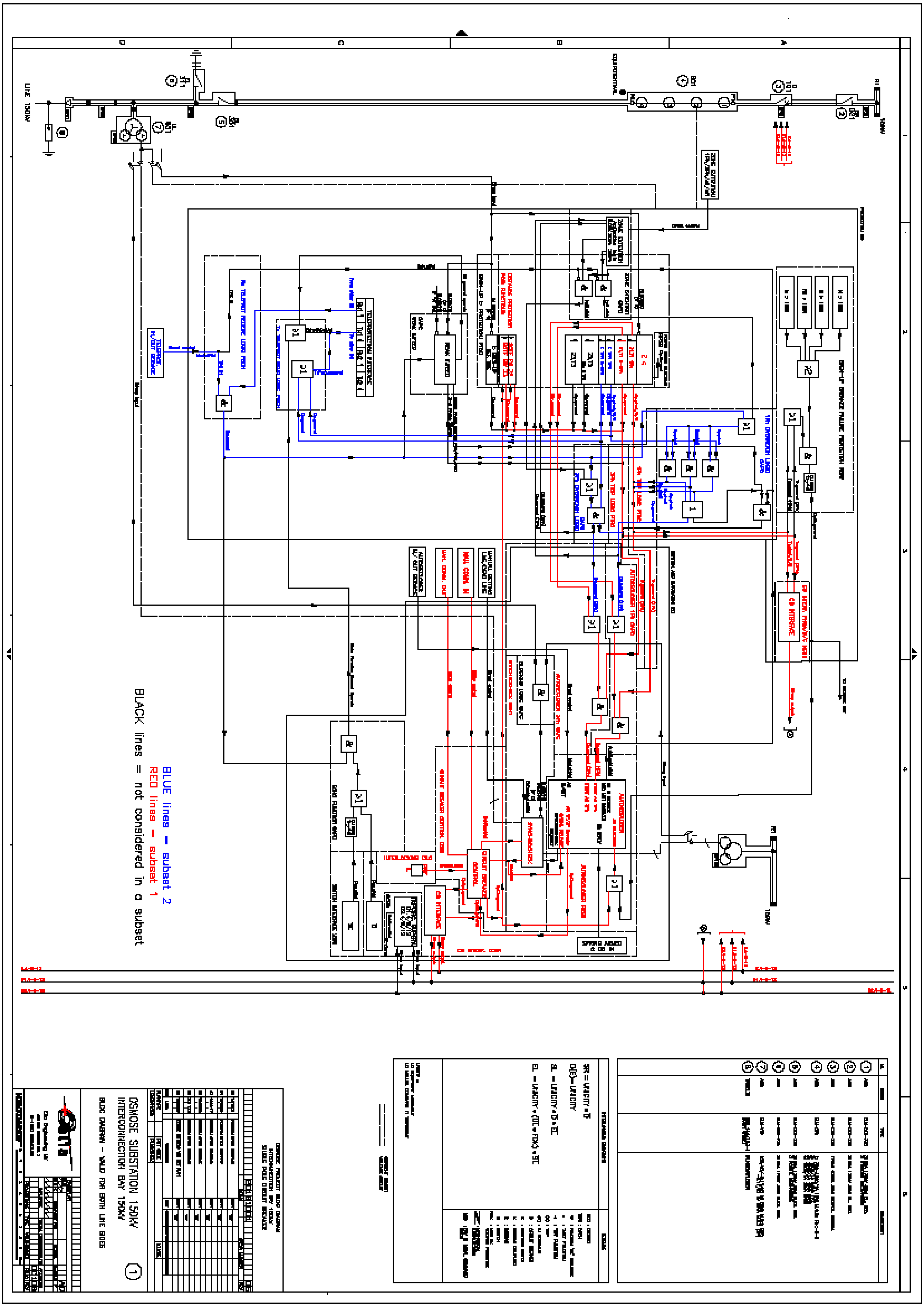

BLUE lines = subset 2
RED lines = subset 1
BLACK lines = not considered in a subset
OSMOSE SUBSTATION 150kV
INTERCONNECTION BAY 150kV

Figure 3: Functional Subset 1 and Subset 2

These two subsets were mainly used to verify and improve the engineering process and interoperability test specification and to improve the tools from the participants.

For a final verification of the interoperability between the tools based on the final version of the interoperability test specification, a further reduced subset called “Teleprotection Exercise” was used as shown in Figure 4. This subset contains all the required elements to perform a full interoperability test as described further down.

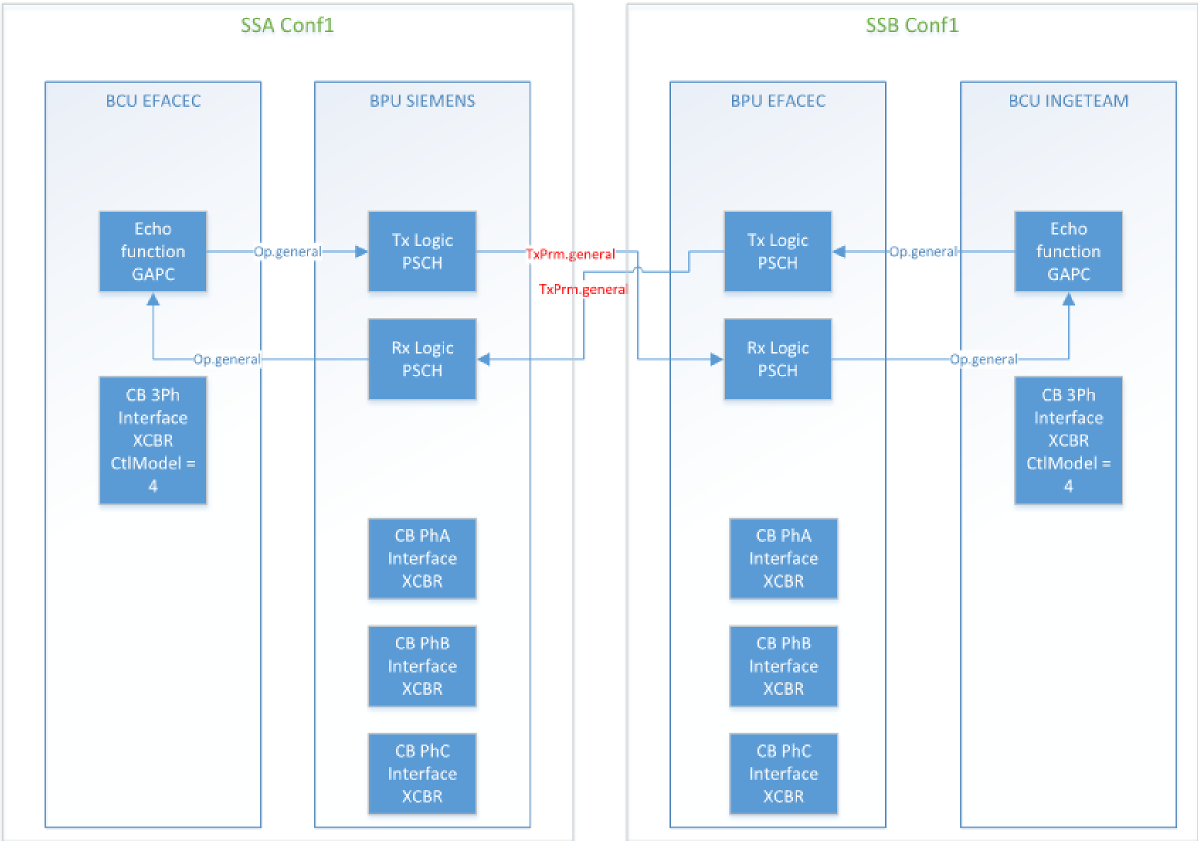


Figure 4: Subset Teleprotection Exercise

5 Demonstrator

5.1 Participants

The following table provides an overview on the products used in the demonstrator:

Component	**Efacec**	**Ingeteam**	**Siemens**	**Helinks**	**Schneider**
Protection device	1	1	1		
Bay controller	2	2			
System specification tool				x	x
System configuration tool	x	x	x	x	

Table 1: Participants and products

5.2 Infrastructure requirements

For the demonstrator, we need besides the IEDs the following infrastructure

- Communication network in each of the substations
- Emulation of the wide area communication between the substations
- Test tool for visualization in each substation
- Test equipment to simulate the power line
- Test equipment to simulate the storage system
- Test equipment to simulate the process (breakers, switches, power flow from other feeders)

5.3 Components of the demonstrator

To accommodate all participating vendors, the demonstrator will be realised twice with different vendors in the two setups. The two configurations of the setup of the demonstrator are shown in Figure 5 and Figure 6.

The two configurations will be designed with different approaches:

- Configuration 1: Specification with virtual IEDs as part of the specification including creation of isd files.
- Configuration 2: Specification without virtual IEDs.

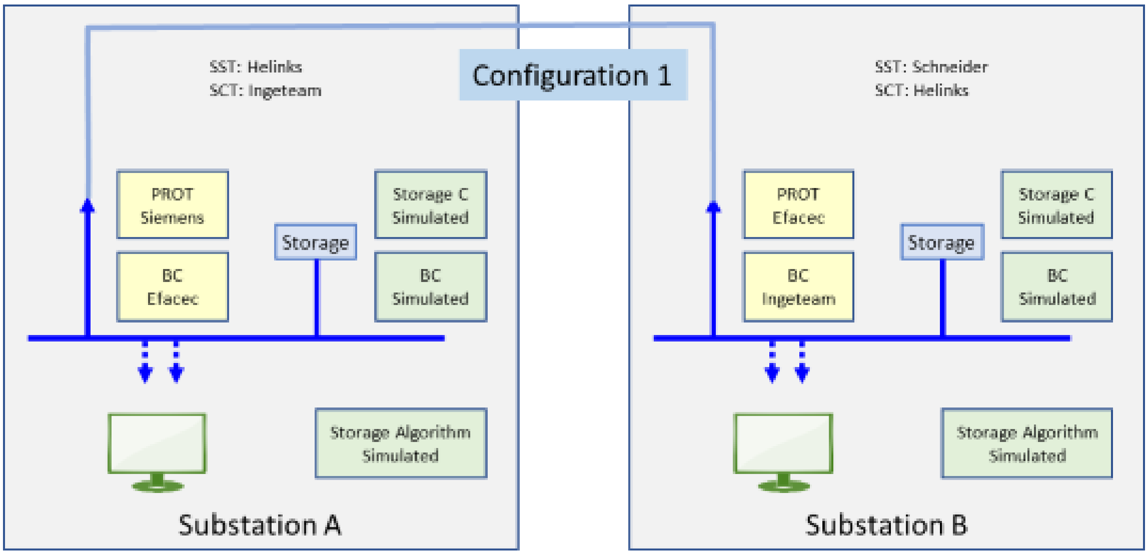


Figure 5: Demonstrator setup configuration 1

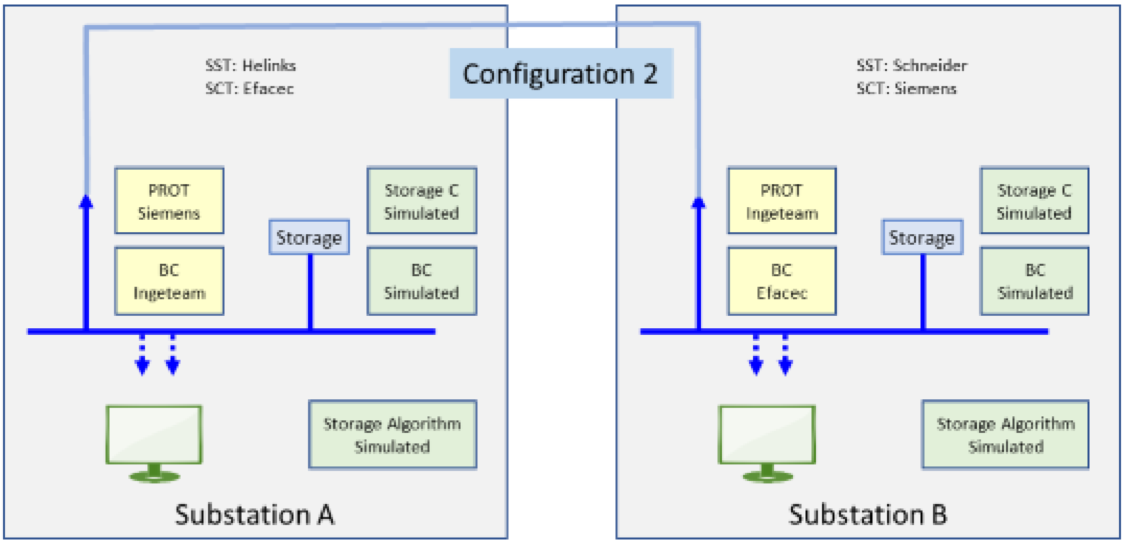


Figure 6: Demonstrator setup configuration 2

Besides the products shown above, the following infrastructure was used:

- The simulation of devices not present (Storage Controller, Bay controller will be done using DTM from Triangle Microworks)
- The local HMI will be done with Test Suite Pro from Triangle Microworks
- Process simulation of Switchgear for the line feeder will be done using a test set capable to simulate analogue and binary signals (e.g. Omicron, Doble, RTDS)
- Process simulation of the storage system and the lines not present are not requiring physical signals and can be done by DTM from Triangle Microworks
- Local communication network and wide area network between the substations

5.4 Demonstrator specification and design

The following table provides an overview on the tools used for specification and design.

	Substation A	**Substation B**
Configuration 1		
Specification	Helinks	Schneider
Design	Ingeteam	Helinks
Configuration 2		
Specification	Helinks	Schneider
Design	Efacec	Siemens

Table 2: Tools used for specification and design

5.5 Demonstrator realization in the lab of R&D Nester

The testing platform in the lab of R&D Nester comprised the following elements:

- Physical IEDs: BC (bay controller) and PROT (protection)
- RTPSS (real time power system simulator): simulates the power system (transmission line and busbar) and test automation
- OMICRON CMC 850: open loop tests (voltage injection)
- Communication switches
- DTM (software): simulates the storage bay (devices and algorithm)
- Test Suite Pro (local HMI)

Since the tests will not be done simultaneously to both platforms, the RTPSS will simulate only one instance of each substation. IED of both configurations will be in parallel. The testing platform is depicted, in a simplified way, in Figure 7.

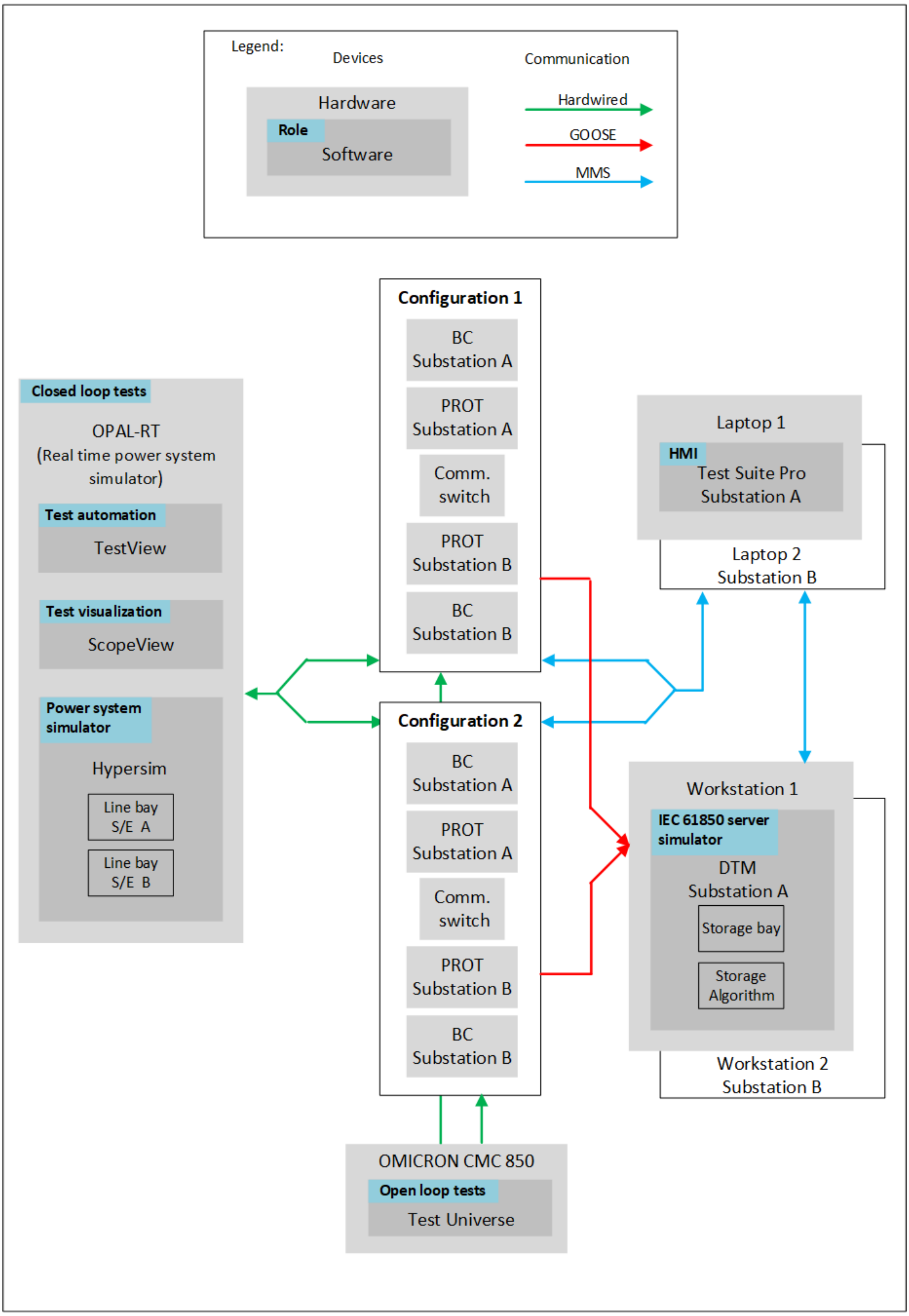


Figure 7: Testing platform architecture

5.5.1 Physical installation

The IEDs are placed in mounting racks. Annex C, sheet 'racks' depicts the position of the IEDs in the racks

5.5.2 Wiring

In Annex C, sheet 'binary', the wiring regarding the binary inputs and outputs and also the power supply are depicted. The following principles apply:

- Each horizontal line is a connection.
- Horizontal lines connected by a thick border means the points are shunted (plus and minus distribution for I/Os in the IEDs.
- The columns 'block' and 'connector' refer to the existing block connectors existing in the mounting racks.

Sheet 'analogue' of the same annex defines the connections regarding analogue inputs.

5.5.3 Communication network

The communication network that is implemented is depicted in Figure 8, indicating the existing VLANs.

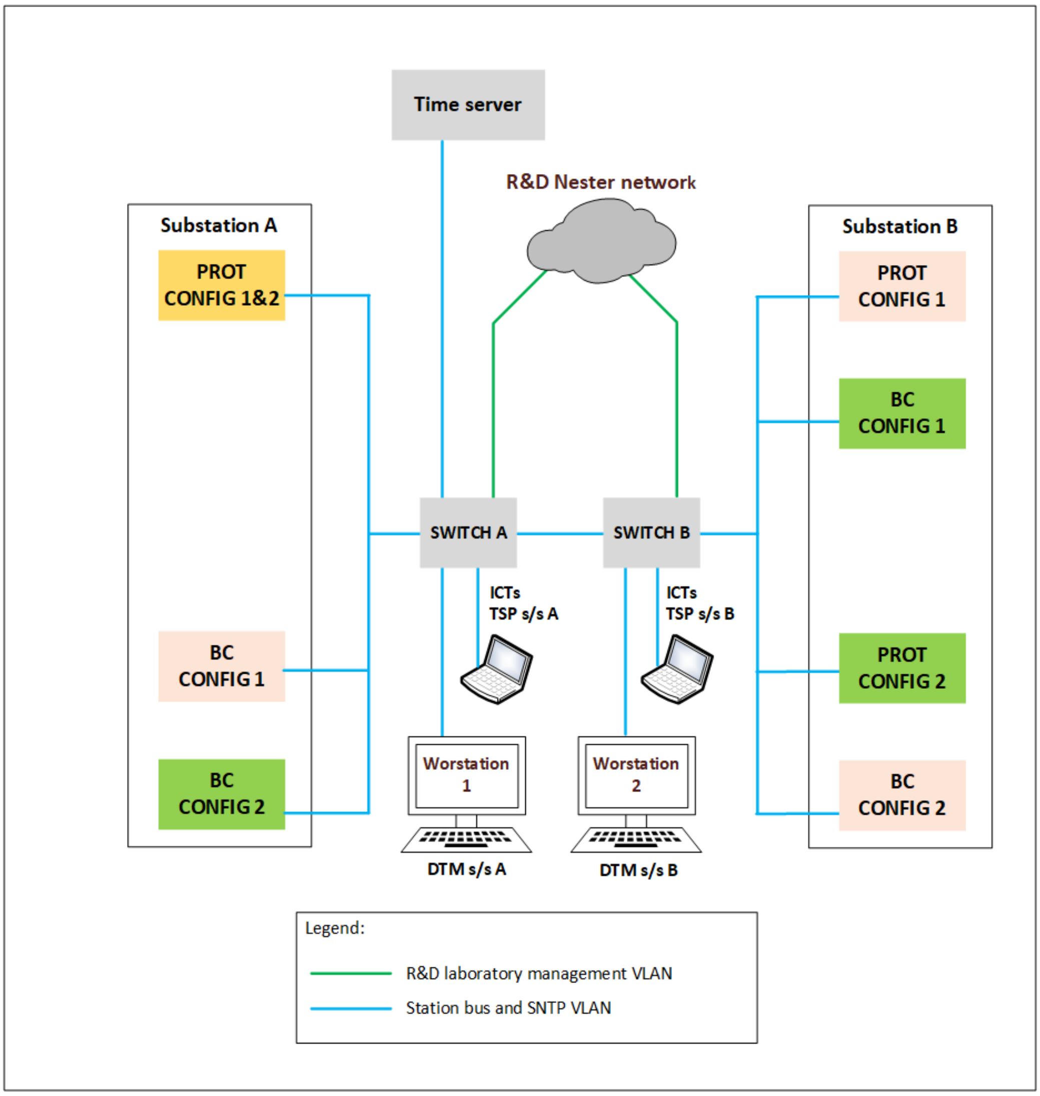


Figure 8: Communication network

5.5.4 GOOSE messages

The table in annex C, sheet ‘GOOSE’ indicates the GOOSE messages between the IEDs. The VLAN for each GOOSE message is indicated.

5.5.5 MMS communication

The table in annex C, sheet ‘MMS’, indicates the necessary MMS communication.

5.5.6 Line and protection data and parameters

The line and protection data and settings are provided in Annex D.

6 Simulation

6.1 Simulation in the context of the engineering process

Simulating a design in an early stage can save time later in the process. The engineering process as foreseen in OSMOSE (see [D7.3], chapter 3) proposes to perform a test without physical devices both at the Specification and at the Configuration stage. Such a test is done by using a simulation tool to simulate the design.

In the configuration stage, when a full SCD file is available, the simulation tool shall be able to simulate IEDs functionality and IED communication. Additionally, the simulation tool shall either simulate the process or it shall be able to interface to a real time simulator which itself is simulating the process.

In the specification stage, where no IEDs and no IEC 61850 communication is yet configured, the simulation tool shall be able to simulate the specified functions and exchange the information between the functions as specified in the SSD file using the SourceRef.

Simulation can as well be applied during commissioning, e.g. as partial simulation of some devices that are not available. In the OSMOSE project, the storage bay will not be equipped with real devices – so that will remain simulated.

6.2 Concepts for functional simulation of IEC 61850 applications

Based on the principles of IEC 61850, where the functionality is decomposed in Logical Nodes, it is reasonable, to use logical nodes as building blocks for the simulation. An IEC 61850 application can be considered as a function block diagram, where the logical nodes are the function blocks which interact with other logical nodes by exchanging signals. The application may be defined as one logic comprising multiple LNs and interactions. Figure 9: Interfaces to a Logical NodeFigure 9 shows the various interfaces to a LN.

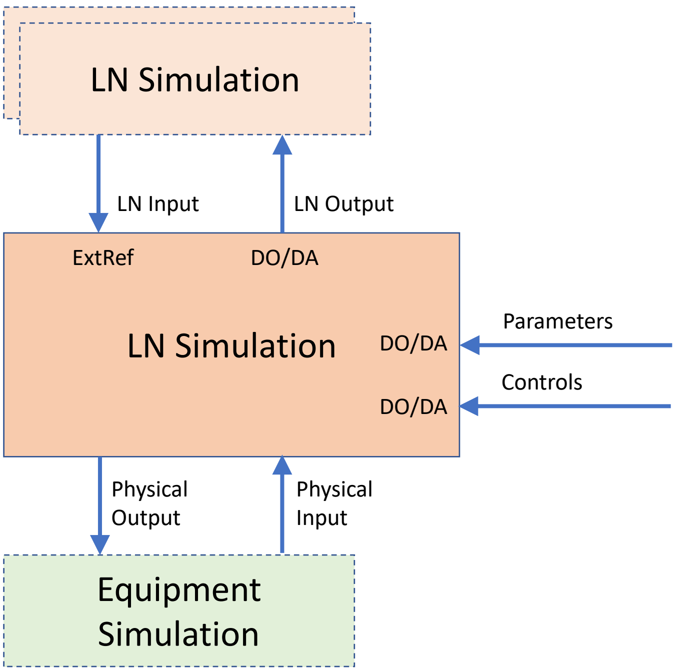


Figure 9: Interfaces to a Logical Node

For the interaction with the other LNs, a LN has inputs and outputs. The outputs are the data objects / data attributes that are standardized. Inputs are not standardized, but they are described in the SCL file with ExtRef and SourceRef. Additionally, there are Parameters that can influence the behavior and controls that may be executed. Those are as well standardized.

For logical nodes that are interfacing with the process / equipment, we have as well physical outputs and inputs.

The simulation tool used in the demonstrator (DTM from Triangle Microworks) includes a library of LN simulation. The connections between the LNs are created based on the SourceRef if available. Otherwise, they may be created from the context; in some cases, they need to be created by the user of the tool.

In some cases it may be required to customize the functional behavior of the LN from the library (e.g. specific interlocking rules). That can be done by creating a custom LN simulation programmed in structured text.

DTM can also simulate equipment behavior like a circuit breaker. With user interaction it is possible to generate specific test conditions like to force a breaker to fail to open.

As the focus of the simulation is to verify the design, which is mainly the connections between the different LNs to implement the application schemes, there is no need to do a full analog simulation of the protection algorithms. It is sufficient, to be able to stimulate e.g. a fault of a specific protection element and verify the correct behavior including tripping an possibly reclosing.

6.3 OSMOSE demonstrator simulation

The functional simulation for the subset 2 of the OSMOSE demonstrator is shown in Figure 10. Each block is a subfunction that represents a logical node. The yellow part are the functions implemented in the bay controller; the red part are the subset 1 of the functions in the protection device; the blue part the subset 2.

The simulation of the LNs GAPC is done as custom logic written in Java Script. Potentially that could as well be in another language like IEC 61131. Also the recloser (LN RREC) is using a custom functionality based on the specification from the OSMOSE application. That functionality is shown in Figure 11.

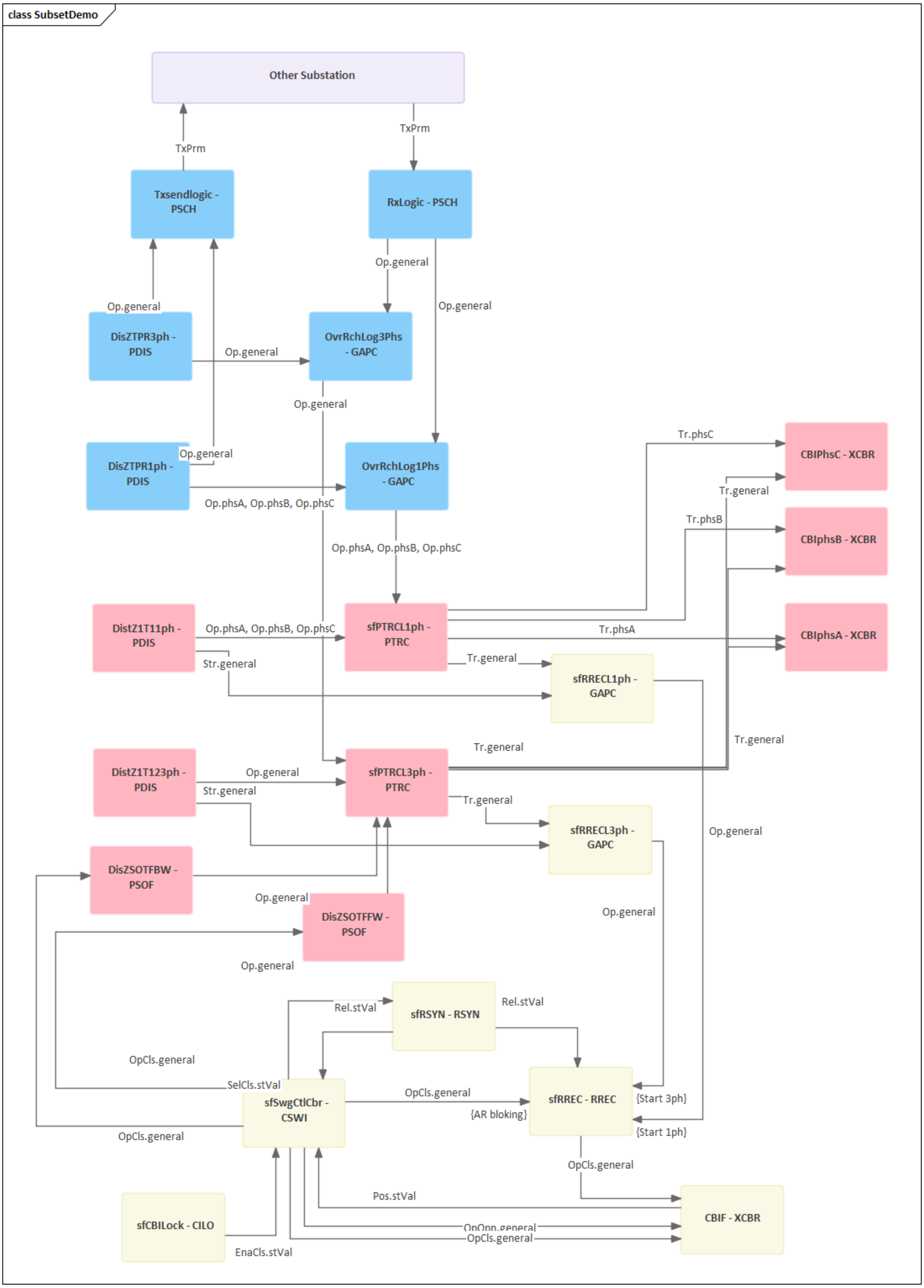


Figure 10: Subset 2 functionality simulated in DTM

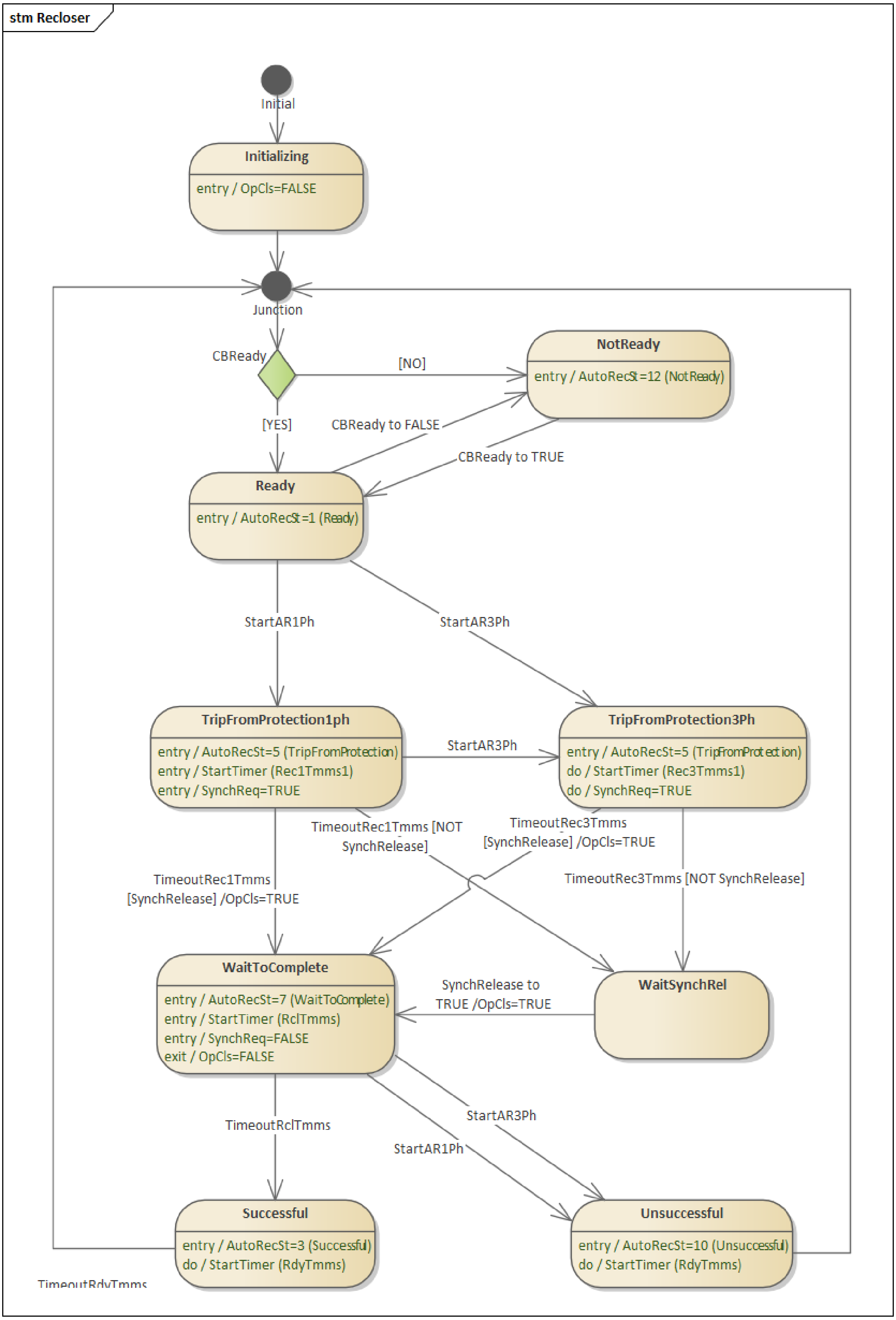


Figure 11: OSMOSE Recloser functionality

To test the teleprotection scheme, the display shown in Figure 12 has been created in DTM.

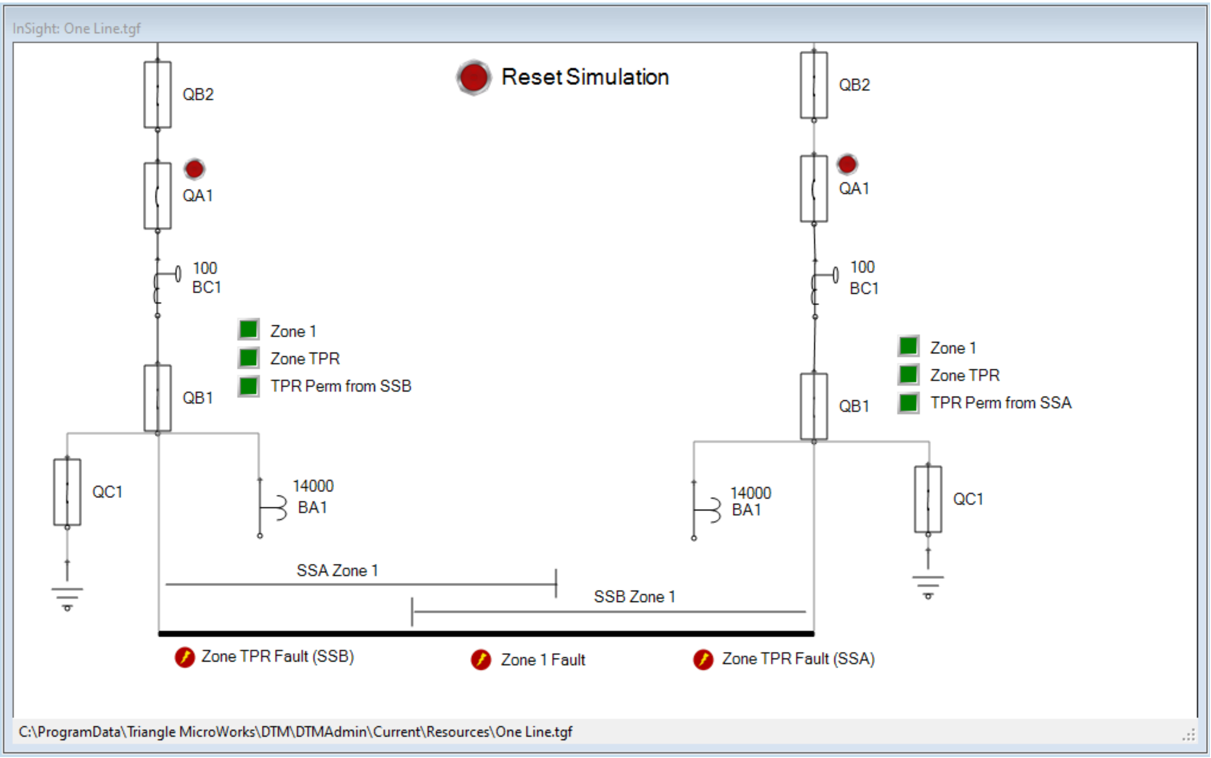


Figure 12: User interface to test the teleprotection scheme

The left part shows substation A, the right part substation B. The red circles with the flash at the bottom of the diagram are used to start a test assuming a fault in one of the zones as indicated. The green squares labeled “Zone 1” and “Zone TPR” show the status of the data object PDIS.Op of the two logical nodes for the two protection elements for zone 1 and for the teleprotection zone. The third green square shows the permissive signal received from the other substation.

With that user interface, a test can be initiated, and the result can be observed. Figure 13 shows the result of a test, where a fault in the teleprotection zone of substation B has been simulated. As it can be seen, the substation B on the right side did see the fault only in the teleprotection zone (Zone 1 did not operate, so the indication is still green). However, it received the permissive signal, so the breaker QA1 did open, which is indicated in the symbol for the breaker.

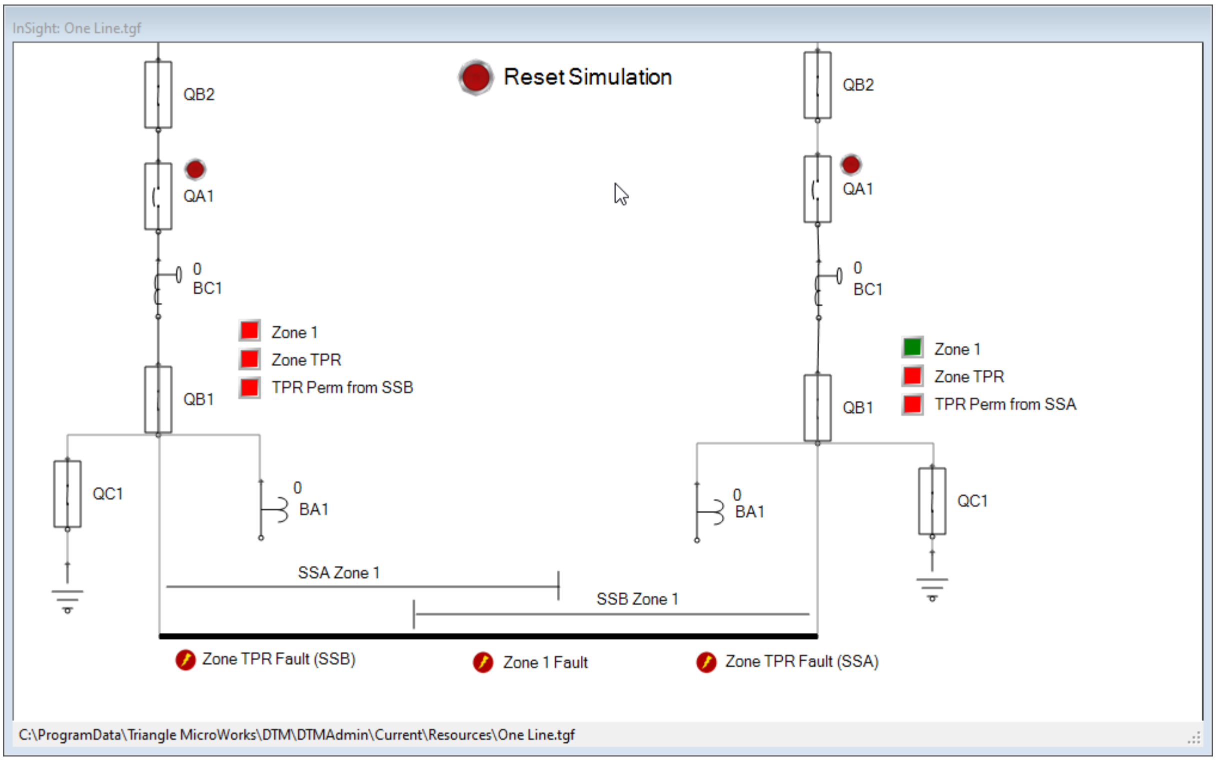


Figure 13: Result of a test simulating a fault in the TPR zone of substation B

7 Functional testing

To verify that the design works as expected, functional tests are performed. For that purpose, a functional test specification has been created (Annex E) and based on that, the detailed test procedures for subset 2 (Annex F).

The tests were performed in the lab on the configuration 1. The test results for substation A are in Annex G, the ones for substation 2 in Annex H.

8 Interoperability testing

8.1 Introduction

The goal of the interoperability testing is, to verify that the tools can create the files according the OSMOSE engineering process and that the files can be exchanged between the tools.

As one goal of the OSMOSE engineering process was as well, to enhance the overall efficiency of the IEC 61850 engineering process, the interoperability tests shall as well assess the efficiency of the process. This includes the verification that a tool makes use of information already provided in an SCL file it receives as input and the user does not need to enter the same information multiple times.

8.2 Workflow steps

According to the OSMOSE process, we have the following steps in the realization of a project:

- Specification
- Procurement
- Configuration
- Commissioning

Independent of a project, a utility may have the typical functions specified with subfunctions and their logical nodes and the required signals. Those functions can be described in SCL.

In the following chapters the process as it has been used for the interoperability testing is summarized. The detailed description of the process can be found in [D7.3].

8.2.1 Specification

In this step, the specification for one substation will be created using a system specification tool (SST). If functions are defined in SCL, those will serve as a starting point. Also, if functions are allocated to IEDs (Configuration 1 of the OSMOSE demonstrator), the IED specification shall be created for the various IEDs.

The system specification shall produce the SSD file defined in IEC 61850-6. The SSD file shall include the process section. It shall be enhanced to include virtual signal flow and possibly virtual IEDs. These enhancements are described in Deliverable 7.3 [D7.3].

For the teleprotection as well as for the storage application, information exchange with the other substation is required. To specify also the virtual signal flow with the other substation, an exchange between the specification tools for the two substations is required. Details are described in [D7.3].

The IED specification shall be an ISD file as proposed in the draft IEC 61850-6-100.

The result of this phase will be an SSD file and optionally multiple ISD files.

This step is done for each substation by one of the vendors of a specification tool participating in OSMOSE.

8.2.2 Procurement

In this step, IEDs are selected that can fulfil the requirements specified with the SSD or ISD file. The IED supplier creates an ICD file. The ICD file shall be enhanced by including a substation section with the mapping of the IED data model to the specified data model as proposed by the OSMOSE engineering process (see [D7.3]).

This step is done for each IED in the demonstrator, based on the SSD or ISD files created during specification.

8.2.3 Configuration

In this step, the design will be made using a system configuration tool (SCT). Input to the step are the SSD file created during specification and the ICD files create during procurement. The result will be an SCD file for the project.

For the teleprotection as well as for the storage application, information exchange with the other substation is required. This configuration of that information exchange shall be handled using the SED file between the tools.

This step is done for each substation by the vendors of a system configuration tool participating in the project.

8.2.4 Commissioning

In this step, the detailed configuration of the IEDs needs to be done using the IED configuration tools (ICT) of the devices. The ICT will import the SCD file.

Iterations between SCT and ICT may be required but shall be kept to a minimum. The process shall follow the standard top down process.

- The SCT defines the signal flow. Preferably, the IEDs support later binding and declare the input signals in the ICD file. If later binding is not supported, the SCT is free to choose where to connect inputs and the IED has to accept them.
- The ICT may add some configurations related to the signal flow as defined in [2] (e.g. modifications on ExtRef or configuration of LGOS). For that, it may be required to load an IID file back into the SCT.

The result of this step are IED configurations.

This step is done for each IED in each substation.

8.3 Interoperability test specification

To verify interoperability and process efficiency, an interoperability test specification has been prepared. That test specification is provided in Annex A.

The following tests have been defined for the different workflow steps:

Step	Test	Remarks
Specification	10	
	12	Extension to 10 for communication between substations
Procurement	151	Create ICD file from ISD file (when using virtual IEDs)
	152	Create ICD file from SSD file

Step	Test	Remarks
Configuration	211	With virtual IEDs
	212	Without virtual IEDs
	22	Extensions for communication between substations
Commissioning	30	

Table 3: Summary of interoperability tests

8.4 Test reports

8.4.1 Introduction

Tests have been done starting March 2020. As travel was not possible until the end of 2021, the focus was on the interoperability testing. This was done using web conferences with screen sharing.

The tests have been done based on the subsets of the functionality described in chapter 4.3.

8.4.2 Test on subset 1

In 2020, initial tests have been done on subset 1. During this stage, the tests were mainly used to refine the OSMOSE process as well as the interoperability test specification.

In May 2020, the following interoperability tests have been done on subset 1:

Test	Remarks
10.1 / 10.3	Specification SS A - Helinks
10.2 / 10.3	Specification SS B - Schneider
211.1	Configuration Config 1 / SS A - Ingeteam
211.2	Configuration Config 1 / SS B - Helinks
212.1	Configuration Config 2 / SS A - efacec
212.2	Configuration Config 2 / SS B - Siemens

Table 4: Interoperability tests done on subset 1

The test reports are provided in Annex K.

8.4.3 Test on subset 2

The focus on subset 2 was on implementing the configuration 1 in the demonstrator with the physical IEDs and perform functional tests. The SCL files were created by the partners offline, without following the interoperability test procedure.

8.4.4 Test on subset “teleprotection exercise”

For the final verification of the process and tools, the limited subset called “teleprotection exercise” was used (see Figure 4). The tests for configuration 2 were done in a partly physical meeting in Brussel in July 2021.

The overview on the tests are provided in the table below. As Siemens could not participate at the meeting, they were replaced by efacec for the IED and by Ingeteam for the system tool. During those tests, the detailed steps for the SED file exchange were defined; therefore, no test results were captured. The test reports are provided in Annex L.

Test	Remarks
152.1	Procurement BPU SS A – efacec (replacing Siemens)
152.2	Procurement BCU SS A – Ingeteam
152.3	Procurement BPU SS B – Ingeteam
152.4	Procurement BCU SS B – efacec
212.1	Configuration SS A - efacec
212.2	Configuration SS B – Ingeteam (replacing Siemens)

Table 5: Interoperability test done on subset "teleprotection exercise", configuration 2

The tests for configuration 1 were done in a physical meeting in Lisbon in October 2021. The overview on the tests performed are provided in the table below. The detailed test results are provided in Annex M.

Test	Remarks
10.1	Specification SS A - Helinks
10.2	Specification SS B - Schneider
12.1a	Specification SS A subscribing from SS B
12.1b	Specification SS B subscribing from SS A
151.1	Procurement BPU SS A – Siemens
151.2	Procurement BCU SS A – efacec
151.3	Procurement BPU SS B – efacec
151.4	Procurement BCU SS B – Ingeteam
211.1	Configuration SS A - Ingeteam
212.2	Configuration SS B – Helinks
22.1a	Configuration SS A subscribing from SS B
22.1b	Configuration SS B subscribing from SS A

Table 6: Interoperability test done on subset "teleprotection exercise", configuration 1

9 Battery use case

9.1 Use case for battery

The description of the use case implemented with the simulated battery system as well as the description of the IEC 61850 data model for the use case can be found in [D7.3].

9.2 Simulation

9.2.1 Architecture of the simulated system

For the simulation, we use a reduced subset of the demonstrator setup as shown in Figure 14: System architecture of battery system simulationWe have in each substation a battery controller in the bay Q2 and a reduced bay control unit (BCU), that is limited to provide the measurements of the bay. Additionally, we have the RTPSS (Real Time Power System Simulator) which provides the measurements for the bays AQ3, AQ1, BQ1 and BQ3.

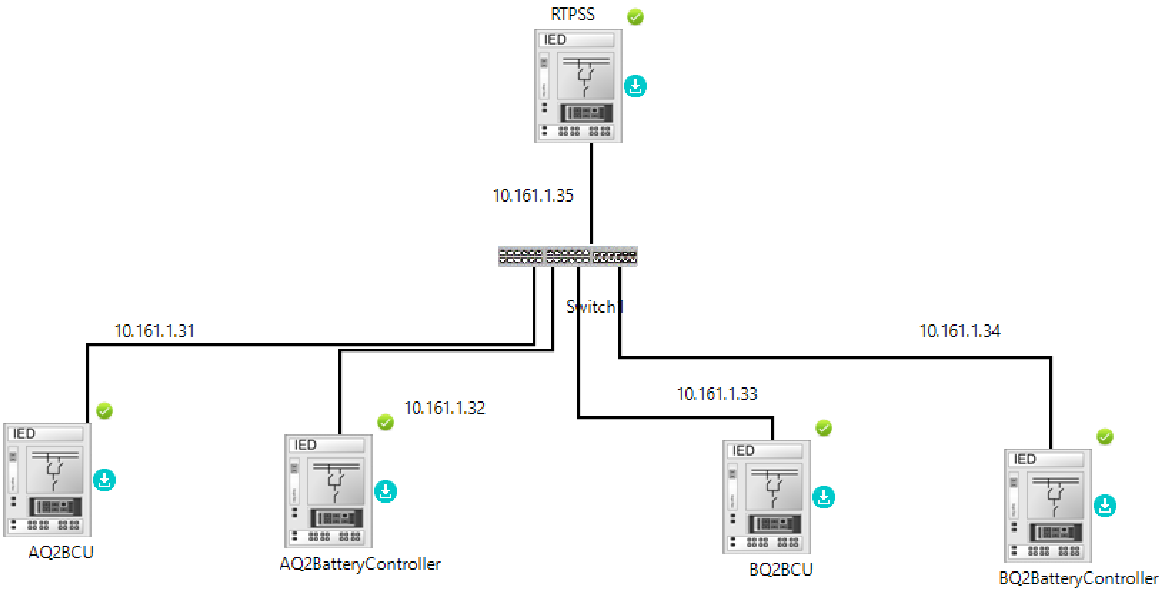


Figure 14: System architecture of battery system simulation

Q1 is the bay that connects to the line that is protected from overload; bay Q2 is the bay with the battery. The Bay Q3 can be considered as a virtual bay that provides as a measurement the sum of all the other bays connected to the busbar.

From a SW perspective, the simulation consists of three parts:

- The simulation of the battery controller IED which implements the use case and controls the battery inverter
- The simulation of the battery itself
- The power flow simulation

The battery controller IED and the battery are simulated with DTM (Distributed Test Manager) from Triangle Microworks. DTM also simulates the two BCUs in bay AQ2 and BQ2 which provide the measurement of the power flow to or from the batteries.

The power flow is simulated with the real time simulator at the R&D Nester Lab which as well simulates the GOOSE message to provide the measurements from bays AQ1, AQ3, BQ1 and BQ3.

9.2.2 Power flow simulation

The application is using total active power measurements (IEC 61850 data object MMXU.TotW). The following ratings are assumed:

- Line: 150 kV / 390 MW
- Battery: 30 MW / 15 MWh

Convention is such that power flow to the busbar is positive, power flow from the busbar is negative.

The power flow simulation needs to do the following:

- The user or test program shall be able to set the virtual power flow in one substation (M1) to observe different test scenarios
- The power flow simulation receives the power flow to both batteries from the battery simulation (M2 and M5)
- The power flow calculation calculates all the other relevant power measurements (M3, M4 and M6)

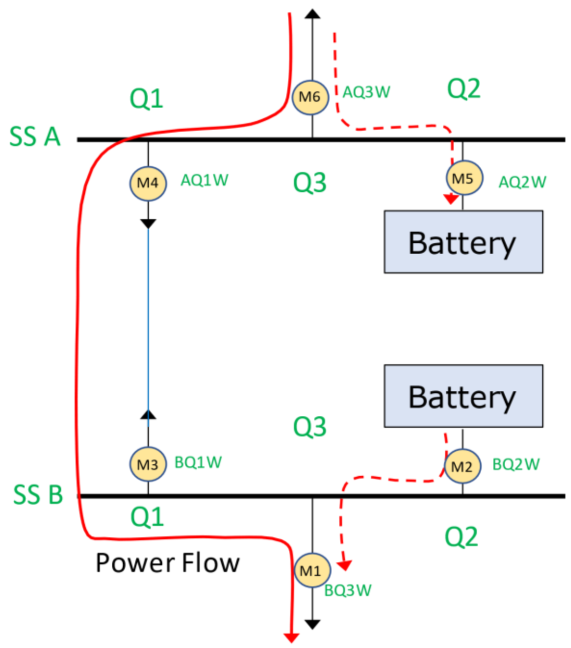


Figure 15: relevant measurements for power flow simulation

In this demonstrator, the R&D Nester's Real Time Power System simulator (RTPSS) from OPAL-RT was used to:

- simulate a small section of an electrical network (2 nodes, 1 line)
- simulate a variable load on SS B

- simulate the generation or consumption of power of the batteries, according to the measurements sent by the BCUs of bays AQ2 and BQ2 BCUs (bay controllers of batteries)
    - these set-points are received through GOOSE messages
- send the power measurements of each line end (AQ1 and BQ1), of generator (AQ3), and of the load (BQ3) to the Battery Controller of Bay AQ2 and BQ2
    - these measurements are sent through GOOSE message

The network model is shown in the figure below.

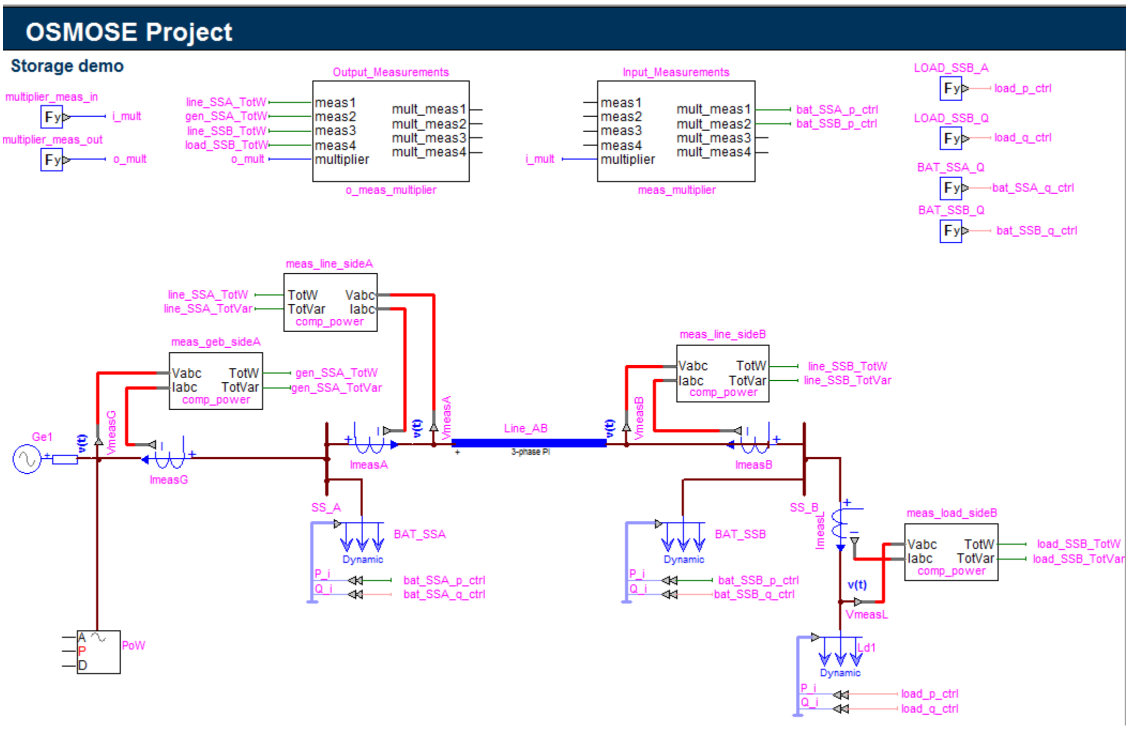


Figure 16: Network model in the RTPSS

The network model contains the following elements:

- generator “Ge1” connected to SS A
- dynamic load “Ld1” connected to SS B (controllable load)
- 150 kV line “Line_AB” connecting SS A to SS B
- dynamic loads “BAT_SSA” and “BAT_SSB” to model the power flows of the batteries located at SS A and SS B

The Dynamic load model can be configured with positive or negative values of active power P) and reactive power (Q). By controlling the power at “Ld1”, the direction and magnitude of the power flow in the line can be controlled, according to the desired scenario. Additionally, this element is also used to model the contribution of the batteries for the power flow, as its P and Q can be controlled through an external signal, which in this case is the power measurement of the batteries sent by AQ2 and BQ2.

Other than the network elements, the simulation model contains:

- Sensor elements for 3-phase voltage and current values (instant values)

- active and reactive power computation blocks, from instant 3-phase voltage and current values
- interface blocks to subscribe/receive GOOSE
  - these include conversion of units for sent/received measurements and deadband for sent measurements

9.2.3 Battery simulation

As the focus of that application was, to verify the DER models in IEC 61850-7-420, the battery simulation implemented has been kept to a minimum. It only considers active power. The simulation receives requests to charge / discharge and calculates the resulting state of charge over the time.

The simulation in DTM is done with two 61131 Applications: One for the battery and the battery controller of substation A, another one for Substation B.

9.2.4 Simulation of Battery Controller IED

The simulation of the battery controller IED is programmed in IEC 61131. DTM currently supports function block diagrams and structured text.

The simulation is following the IEC 61850 model. The application is decomposed into the IEC 61850 logical nodes and is implemented as a function block diagram as shown in Figure 17: Function block diagram for the battery controller.The individual logical node behavior is programmed in structured text.

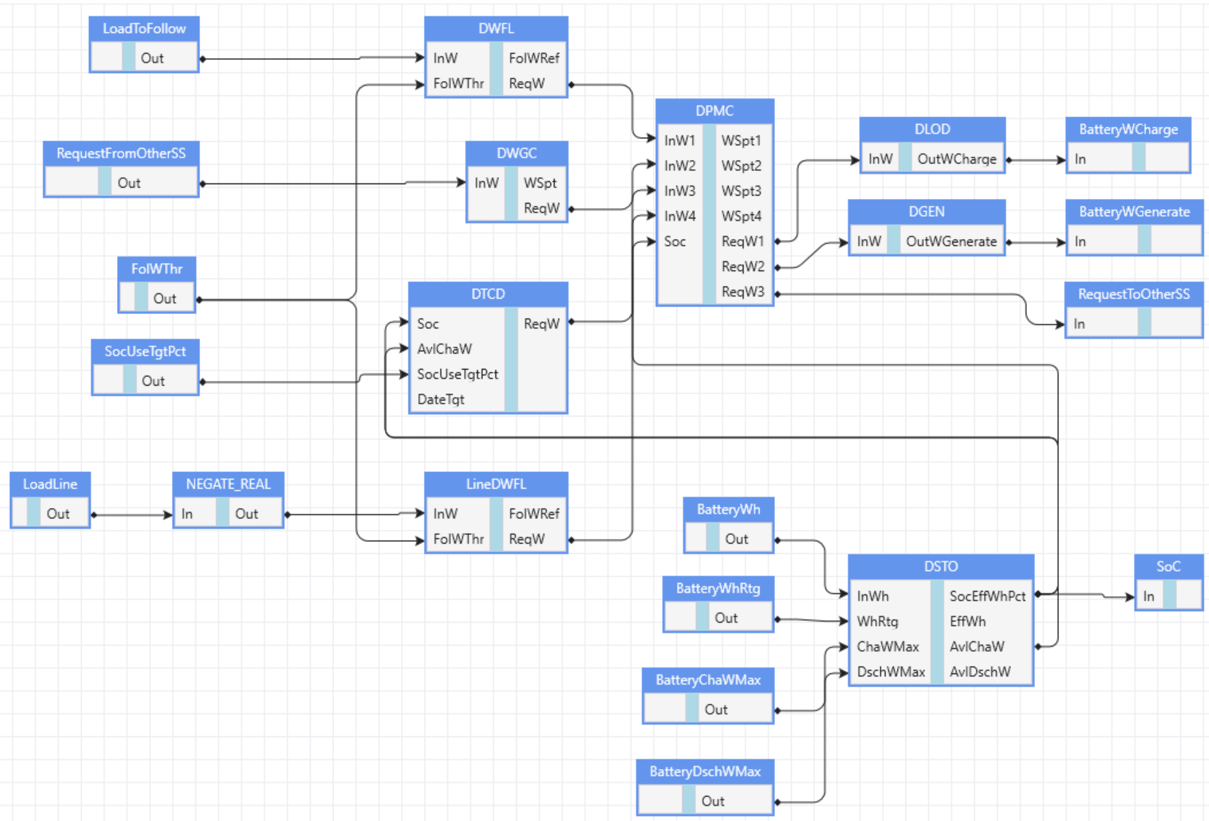


Figure 17: Function block diagram for the battery controller

Using the concepts introduced in OSMOSE, the simulation can be done on an SSD file only without IEDs or on an SCD file with IEDs.

DTM uses the LNode from the process / substation section of an SCL file for the functional simulation. When the simulation is done on the SSD only, data from the simulation are only available within DTM and are not available to an outside world in a standardized way. To make data available using IEC 61850 communication services, the functional LNs used in DTM (the LNode from the SCL) needs to be mapped on an LN implemented in an IED. This requires the SCD file to be present. In that case, DTM will simulate IEC 61850 communication like GOOSE messages and Reports as configured in the SCD file.

Based on the assumption from the power flow simulation, the parameters from the IEC 61850 model are configured as follows:

- For the maximum rating of the line (in the two LNs DWFL)
    - DWFL.FolWThr = -390 MW
- For the battery characteristics (in DSTO)
    - DSTO.WhRtg = 15 MWh
    - DSTO.ChaWMax = 30 MW
    - DSTO.DschWMax = 30 MW
- The target state of charge of the battery when reloading
    - DTCD.SocUseTgtPct = 49.5 %

Note: DTCD.SocUseTgtPct has been set to a value less than 50% to avoid an unstable situation where while bringing the batteries back to 50 %, the battery that is being discharged overshoots and starts recharging again.

9.2.5 Data exchange between DTM and the RTPSS

The RTPSS simulates the power flow in the network. The following data needs to be exchanged with the simulation run in DTM:

- From DTM to the real time simulator: The power flow from the batteries to the busbar (M2 / BQ2W and M5 / AQ2W)
- From the real time simulator to DTM: the other power flows (M1/BQ3W, M3/BQ1W, M4/AQ1W and M6/AQ3W)

The exchange will be done using GOOSE messaging. The signals exchanged are shown in Figure 18.

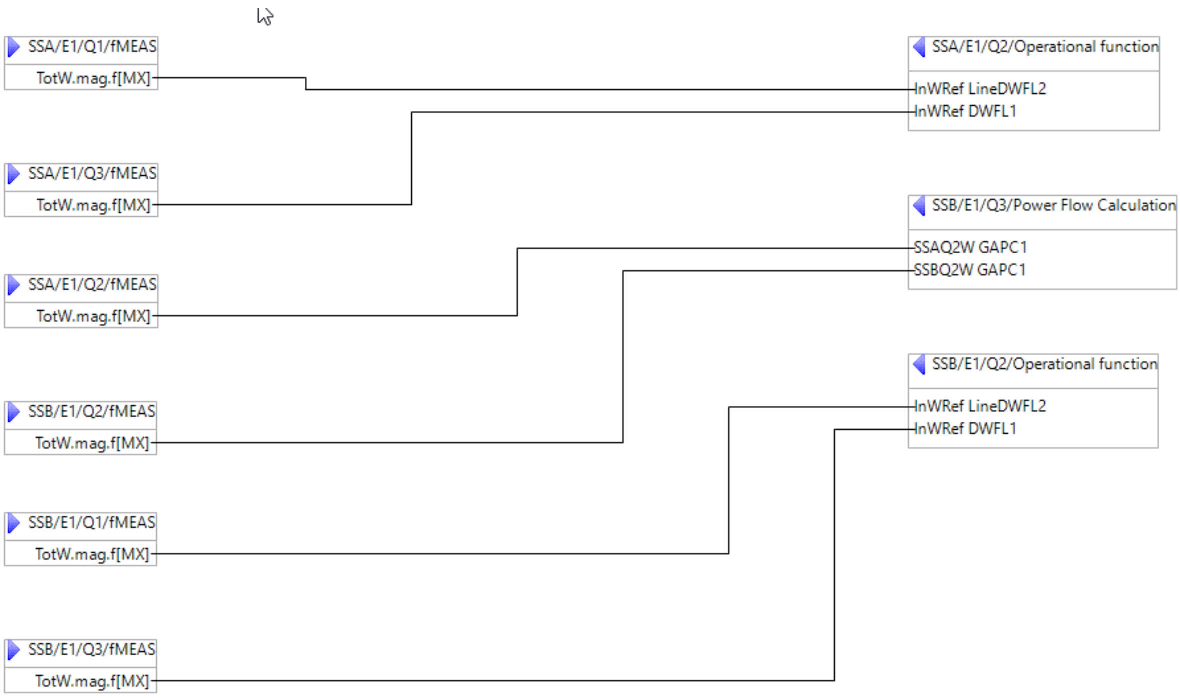


Figure 18: Signals exchanged over GOOSE

9.3 Observation of test results

9.3.1 Observation of results in DTM

DTM includes the possibility to create displays, where information from the system can be visualized. This can be information produced as part of the simulation in DTM, but it can as well be information received from other devices through IEC 61850 communication.

Figure 19 shows the display used in DTM to visualize the key values. It shows a situation where the load in Substation B has exceeded the maximum rating of the power line (390 MW). To compensate, the battery in Substation A is consuming10 MW while the battery in Substation B produces the 10 MW. As a result, the State of Charge of Battery A increases while the SoC of battery B decreases.

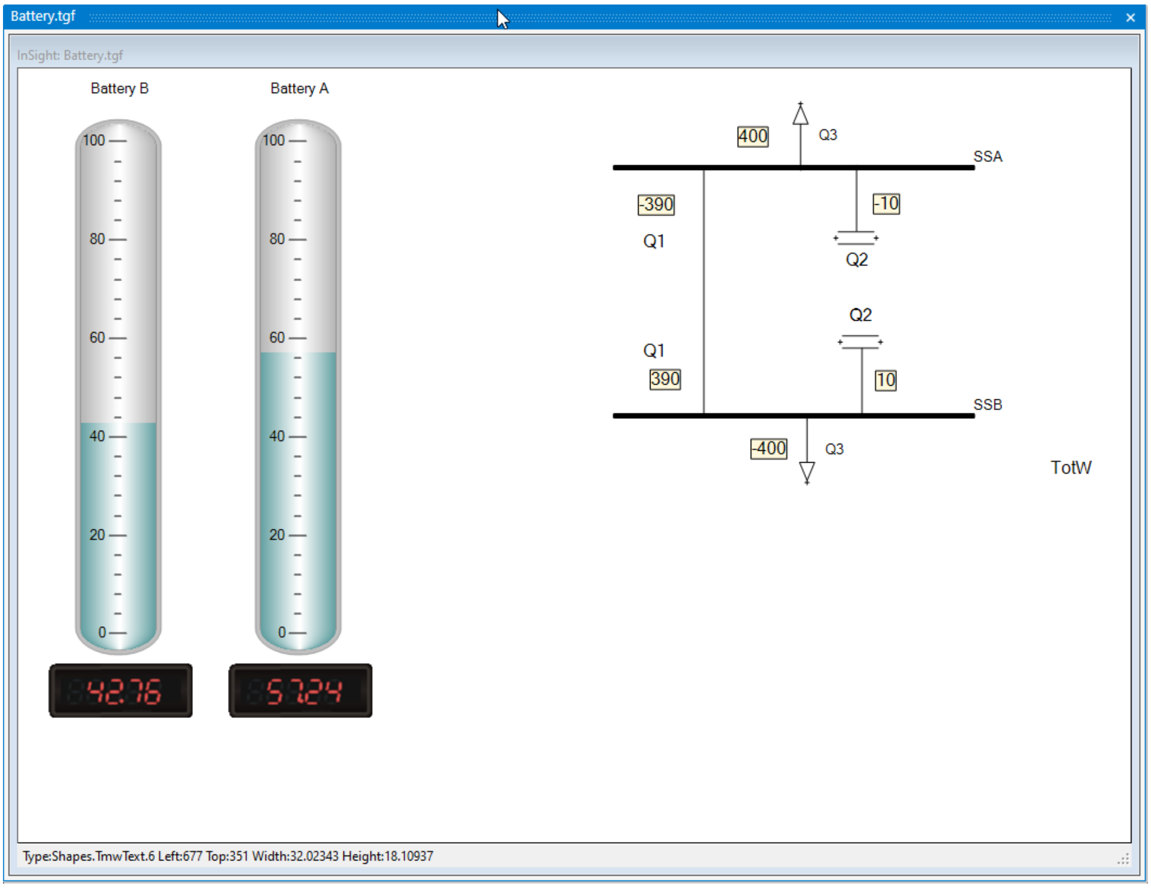


Figure 19: Visualization of the key values in DTM

9.3.2 Observation of the results with Test Suite Pro

Test Suite Pro (TSP) from Triangle Microworks is a tool that has been used in the demonstrator to act as a client and to visualize information. It acts as an IEC 61850 client as well as a GOOSE subscriber to collect information from the system.

With TSP it is possible to display values acquired either with reporting, with polling or with GOOSE. TSP as well observes the network for the GOOSE messages on the wire.

Figure 20 shows the visualization of the measured values. Figure 21 shows the GOOSE messages exchanged between the devices AQ2BCU and BQ2BCU, which are simulated in DTM and the RTPSS as well as the GOOSE message from the RTPSS to DTM.

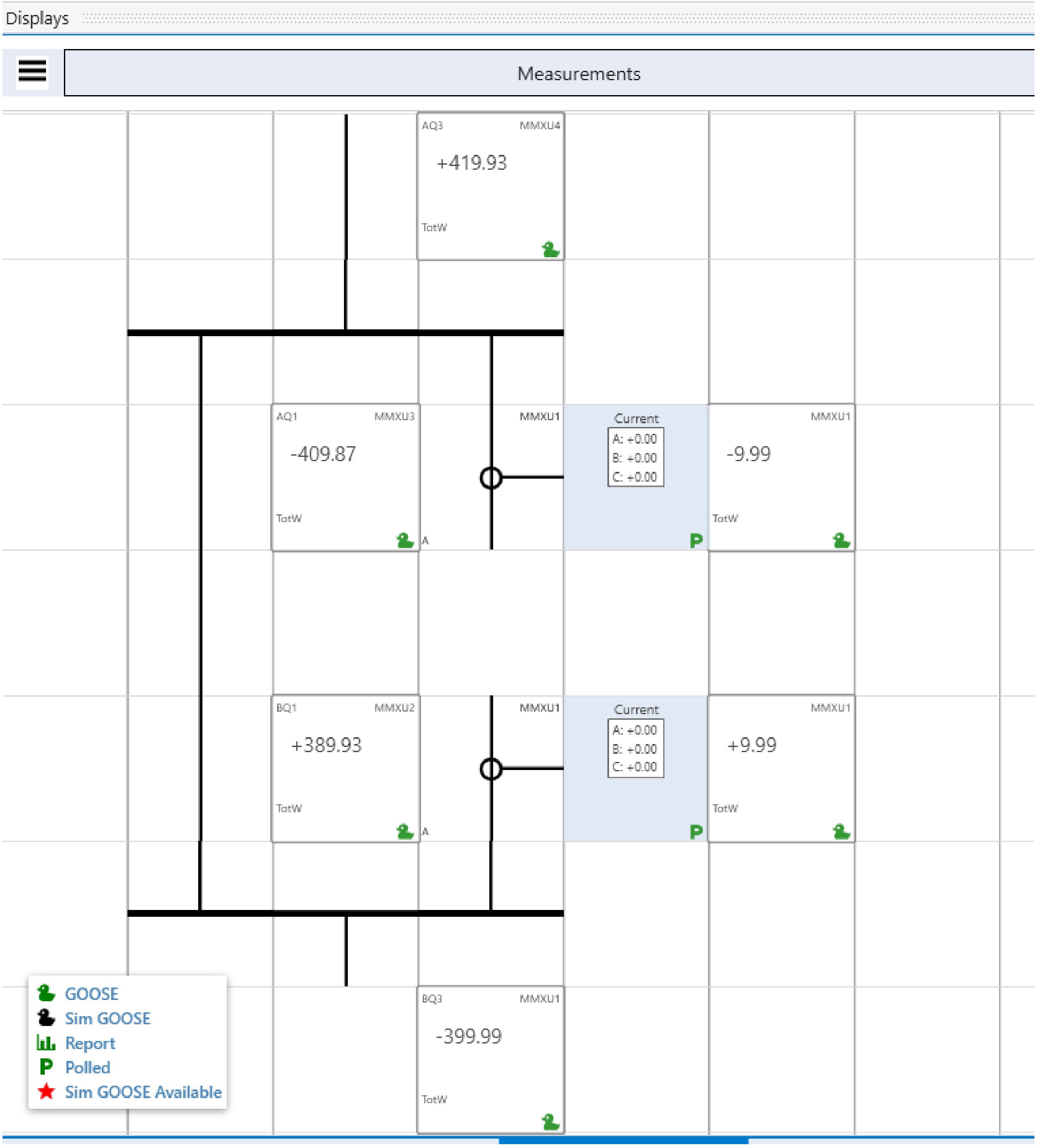
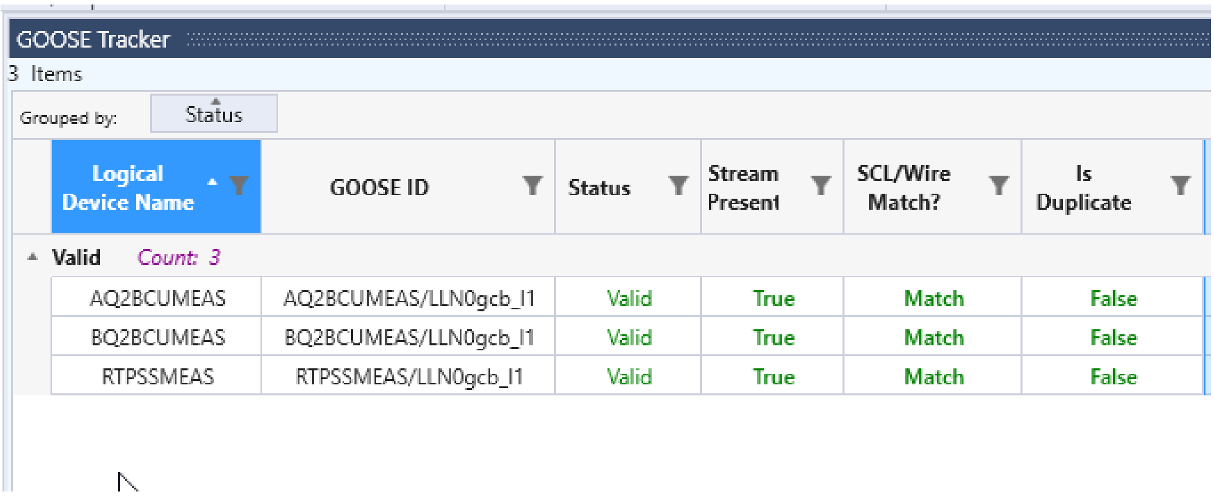


Figure 20: Visualization of the measurements in Test Suite Pro


GOOSE Tracker

3 Items

Grouped by: Status

Logical Device Name	GOOSE ID	Status	Stream Present	SCL/Wire Match?	Is Duplicate
Valid Count: 3					
AQ2BCUMEAS	AQ2BCUMEAS/LLN0gcb_I1	Valid	True	Match	False
BQ2BCUMEAS	BQ2BCUMEAS/LLN0gcb_I1	Valid	True	Match	False
RTPSSMEAS	RTPSSMEAS/LLN0gcb_I1	Valid	True	Match	False

Figure 21: GOOSE messages on the wire

9.3.3 Recording of the measurements in the RTPSS

The RTPSS allows the recording of the values for documentation of the results.

9.4 Test cases

For testing, we can differentiate between the following situations in each substation.

If the substation is a power source for the line (Q3MMXU.TotW > 0), the other substation is driving the action.

If the substation is a load for the line, we can differentiate between the following test situations:

(0) The load is beyond the rating of the line (Q3MMXU.TotW > -390 MW) – this is the normal situation

(1) The load is above the rating of the power line (Q3MMXU.TotW < -390 MW). The local battery generates the power for the additional load (up to the maximum power rating of the battery which is 30 MW in our example) and requests the battery in the other substation to consume the same amount of power.

(2) The battery was used to generate power and the load is back more than the maximum power rating of the battery beyond the rating of the power line (Q3MMXU.TotW > -360 MW). The local battery will be recharged with the maximum power of 30 MW and request the battery in the other substation to generate that amount of power.

(3) The battery was used to generate power and the load is back less than the maximum power rating of the battery beyond the rating of the power line (-360 MW > Q3MMXU.TotW > -390 MW). The local battery will be recharged with limited power in order to not overload the line and request the battery in the other substation to generate the same amount of power.

To test the behavior, the following test sequence was applied:

Step	No	Test description	Expected Behavior	Result
S1		Normal operation	No usage of battery	
S2	(B1)	Load in SS B higher than rating of line	Battery B generating, Battery A consuming	
S3	(B2)	Load in SS B more than Max Power of battery less than rating of Line	Battery B recharging / Battery A discharging with Max Power of Battery until desired SoC;	
S4		Wait until batteries are back in 50%	Charging / Generating shall stop	
S5	(1)	Repeat test (1) to start using the batteries again		
S6	(B3)	Load in SS B less than Max Power of Battery less than rating of line	Battery B recharging / Battery A discharging with less than Max Power of Battery until desired SoC	
S7		Wait until batteries are back in 50%	Charging / Generating shall stop	

Table 7: Test sequence for testing the battery application

The same tests are then repeated with SS A being the load and driving. This has been tested by assuming now positive values in SS B.

The test results are provided in Annex O.

10 References

[D7.3] OSMOSE Deliverable D7.3: Recommendations for IEC61850 WG10 and the industry

11 Annexes – external documents

A: Interoperability test specification (ID7.2.A)

B: Block Diagram of the OSMOSE Line Bay (ID7.2.B)

C: Testing platform description (ID7.2.C)

D: Line and protection data and settings (ID7.2.D)

E: Functional test specification (ID7.2.E)

F: Test procedures subset 2 (ID7.2.F)

G: Test results subset 2, substation A (ID7.2.G)

H: Test results subset 2, substation B (ID7.2.H)

I: SCL files of teleprotection exercise

12 Annex K: Interoperability test results subset 1

12.1 Test 10.1 – Specification SS A (Helinks)

	Test Step	**Verification**	**ok**	**Remarks / Observations**
A	Function specification			
A1	Export function specification from IST			
B	System specification			
B1	SST imports function specification from IST	SST is able to import the function specification	☐ok	Imported in library
B2	SST completes incoherent data type templates	SST is able to modify / complete enumtypes and struct types. This requires user interaction to select - What values out of the enum to support? - What child elements to support	☐ok	In communication Editor this can be done, but today manually
B3	Design single line diagram in SST		☐ok	Using a library with a single line template

B4	Instantiate functions	SST uses the IST export with the logical nodes to create and populate the function structure and uses the data objects from the IST export to create the LN templates Instantiation includes - Selecting the element where the function shall be instantiated - User interaction to add power system resource references where needed	☐ok	Done in function editor; can be done to function and subfunctions Currently it cannot be linked to sub equipment Allocating power system resource to the subfunction does not yet produce the link to the power system resource at the subfunction level
B5	Define signal flow	SST supports the creation of the signal flow based on the inputs defined in the IST export - Connections are only required on the SourceRef from the IST export file that are not "wired" or "control" - Connections are typically made by user interaction; if LNs are allocated to a power system resource this can be used by the tool to automatically create the connection	☐ok	Done manually in function editor

B6	Define pre-configuration values	SST allows the user to configure default values for e.g. settings	☐ok	Parameters are not supported yet by IST export
B7	Define virtual IEDs (only configuration 1)	SST supports the allocation of LNs to virtual IEDs. Logical devices are created: - For each function and non-leaf subfunction - Function / Subfunction hierarchy is reflected in LD hierarchy using GrRef	☐ok	Allocation can only be done at function level, not at subfunction level Work around for CBIF: two functions are created - one for the general XCBR which is assigned to BCU - one for the three phase specific XCBRs which are assigned to BPU
B8	Configure signal flow between virtual IEDs	SST creates: - ExtRef based on SourceRef - Optionally GOOSE messages; required if user has specific requirements	☐ok	
B9	Export SSD	SST is able to produce SSD file	☐ok	
B10	Export ISD (only configuration 1)	SST is able to produce ISD files	☐ok	
C	SSD file inspection			
C1	Verify step B3	In the SSD file, verify the substation structure with the voltage levels and bays as defined in the project specification	☐ok	

C2	Verify step B4	In the SSD file, verify that the function structure has been created using the elements from the IST export - Verify that the specification naming element is present - Verify that power system reference is present	☐ok	
C3	Verify step B5	In the SSD file, verify that the inputs have been connected	☐ok	
C4	Verify step B7 (only configuration 1)	In the SSD file, verify that the virtual IEDs are defined including verification of the signal flow (ExtRef and optionally GOOSE configuration)	☐ok	
C5	Check SSD file	Run SCD file through various SCL checkers and validators; report results for documentation	☐	
D	ISD file inspection (only configuration 1)			
D1	Verify step B6	Verify in the LD / LN Structure that all LNs are present and compare with the function / subfunction structure		

D2	Verify step B8	Verify that SourceRef have been translated in ExtRef with service type specified		

12.2 Test 10.2 – Specification SS B (Schneider)

	Test Step	Verification	ok	Remarks / Observations
A	Function specification			
A1	Export function specification from IST			
B	System specification			
B1	SST imports function specification from IST	SST is able to import the function specification	☐ok	
B2	SST completes incoherent data type templates	SST is able to modify / complete enumtypes and struct types. This requires user interaction to select - What values out of the enum to support? - What child elements to support	☐ok	In Schneider Tool, enumtype can currently not be modified
B3	Design single line diagram in SST			

B4	Instantiate functions	SST uses the IST export with the logical nodes to create and populate the function structure and uses the data objects from the IST export to create the LN templates Instantiation includes - Selecting the element where the function shall be instantiated - User interaction to add power system resource references where needed	☐ok	In Schneider tool, power system resource reference cannot be added currently
B5	Define signal flow	SST supports the creation of the signal flow based on the inputs defined in the IST export - Connections are only required on the SourceRef from the IST export file that are not "wired" or "control" - Connections are typically made by user interaction; if LNs are allocated to a power system resource this can be used by the tool to automatically create the connection	☐ok	User of Schneider tool needs to make the connection manually

B6	Define pre-configuration values	SST allows the user to configure default values for e.g. settings		Not supported yet in Schneider tool
B7	Define virtual IEDs (only configuration 1)	SST supports the allocation of LNs to virtual IEDs. Logical devices are created: - For each function and non-leaf subfunction - Function / Subfunction hierarchy is reflected in LD hierarchy using GrRef	☐ok	
B8	Configure signal flow between virtual IEDs	SST creates: - ExtRef based on SourceRef - Optionally GOOSE messages; required if user has specific requirements		
B9	Export SSD	SST is able to produce SSD file	☐ok	
B10	Export ISD (only configuration 1)	SST is able to produce ISD files	☐ok	
C	SSD file inspection			
C1	Verify step B3	In the SSD file, verify the substation structure with the voltage levels and bays as defined in the project specification	☐ok	

C2	Verify step B4	In the SSD file, verify that the function structure has been created using the elements from the IST export - Verify that the specification naming element is present - Verify that power system reference is present	☐ok	
C3	Verify step B5	In the SSD file, verify that the inputs have been connected	☐ok	
C4	Verify step B7 (only configuration 1)	In the SSD file, verify that the virtual IEDs are defined including verification of the signal flow (ExtRef and optionally GOOSE configuration)	☐	Service type was not configured for ExtRef
C5	Check SSD file	Run SCD file through various SCL checkers and validators; report results for documentation	☐	
D	ISD file inspection (only configuration 1)			
D1	Verify step B6	Verify in the LD / LN Structure that all LNs are present and compare with the function / subfunction structure		

D2	Verify step B8	Verify that SourceRef have been translated in ExtRef with service type specified		

12.3 Test 211.1 – Configuration config 1, SS A (Ingeteam)

Specification: Helinks
BCU: Ingeteam, PROT: Efacec

	Test Step	Verification	Ok ☐☑☒	Remarks / Observations
A	File import			
A1	SCT imports SSD file	SCT is able to import SSD file	☑	
A2	SCT imports ISD file (if needed; see note)	SCT is able to import ISD file	☐	
A3	SCT imports ICD file	SCT is able to import ICD file	☑	
B	ICD / ISD compare			
B1	Check that all specified LNs are implemented	SCT is able to show a relation between the specified LNs and the implemented LNs	☐	Not done
B2	Check that all DO/DA are implemented	SCT identifies missing DO/DA	☐	Not done
B3	Check service capabilities	SCT identifies limiting capabilities of IED	☐	Not done

C	Design			
C1	Replace virtual IED with physical IED	SCT replaces the virtual IED completely with the physical IED - The service section - The complete data model	☑	For the data model, this is done step by step as the mapping is updated Service section is compared and differences indicated
C2	Update the mapping of the LNodes	SCT updates the mapping of the LNodes to the LN in the physical IED using the mapping defined in the ICD file	☑	User has to select, which LNode mapping to update
C3 (a)	If communication section is already configured in SSD file, update Access point references	SCT updates access point references in the communication section to match the physical IED	☐	
C3 (b)	Create communication section, if none is specified in SSD	SCT creates the communication section	☑	
C4 (a)	If GOOSE configuration is present in SSD file, update data flow for protection and control schemes	SCT uses the existing configuration of GOOSE (Control block, dataset and GSE element in communication section) and LN Inputs but updates all the references to match the physical IED	☐	n/a

C4 (b)	Configure GOOSE and LN Inputs (ExtRef) based on the source Ref	SCT implements the ExtRefs for signals exchanged between the IEDs based on SourceRefs and configures GOOSE messages (Control block, dataset and GSE element in communication section)	☑	Needs to be done manually; not based on SourceRef
C5	Configure data flow to client devices	SCT configures reports based on specification	☐	Not done
C6	Export SCD file	SCT is able to export SCD file	☑	

12.4 Test 211.2 – Configuration config 1, SS B (Helinks)

Specification: Schneider
BCU: efacec, PROT: Siemens

	Test Step	**Verification**	**Ok** ☐☑☒	**Remarks / Observations**
A	File import			
A1	SCT imports SSD file	SCT is able to import SSD file	☑	Possibility to use a predefined single line diagram to arrange the graphics Import of power system resource is missing
A2	SCT imports ISD file (if needed; see note)	SCT is able to import ISD file	☐	

A3	SCT imports ICD file	SCT is able to import ICD file	☑	Overloads the predefined model from the SSD
B	ICD / ISD compare			
B1	Check that all specified LNs are implemented	SCT is able to show a relation between the specified LNs and the implemented LNs	☐	Not done
B2	Check that all DO/DA are implemented	SCT identifies missing DO/DA	☐	Not done
B3	Check service capabilities	SCT identifies limiting capabilities of IED	☐	Not done
C	Design			
C1	Replace virtual IED with physical IED	SCT replaces the virtual IED completely with the physical IED - The service section - The complete data model	☑	Done with the import of the icd;
C2	Update the mapping of the LNodes	SCT updates the mapping of the LNodes to the LN in the physical IED using the mapping defined in the ICD file	☐	Association is done on the function level;

C3 (a)	If communication section is already configured in SSD file, update Access point references	SCT updates access point references in the communication section to match the physical IED	☐	With SSD import, Communication section was imported. For one IED, access point was updated; but with that as well the IP address is changed – IP address should remain as specified. For second IED, access point was not updated
C3 (b)	Create communication section, if none is specified in SSD	SCT creates the communication section	☐	n/a
C4 (a)	If GOOSE configuration is present in SSD file, update data flow for protection and control schemes respecting the constraints declared in the service section of the IED	SCT uses the existing configuration of GOOSE (Control block, dataset and GSE element in communication section) and LN Inputs but updates all the references to match the physical IED	☐	
C4 (b)	Configure GOOSE and LN Inputs (ExtRef) based on the source Ref	SCT implements the ExtRefs for signals exchanged between the IEDs based on SourceRefs and configures GOOSE messages (Control block, dataset and GSE element in communication section)	☐	
C5	Configure data flow to client devices	SCT configures reports based on specification	☐	
C6	Export SCD file	SCT is able to export SCD file	☐	

C	SSD file inspection			

12.5 Test 212.1 – Configuration config 1, SS A (efacec)

	Test Step	Verification	ok	Remarks / Observations
A	File import			
A1	SCT imports SSD file	SCT is able to import SSD file	☐	
A2	SCT imports ICD file	SCT is able to import ICD file	☐	
B	ICD / SSD compare			
B1	Check that all specified LNs are implemented	SCT is able to show a relation between the specified LNs and the implemented LNs	☐	Not done
B2	Check that all DO/DA are implemented	SCT identifies missing DO/DA	☐	Not done
B3	Check service capabilities	SCT identifies limiting capabilities of IED	☐	Not done
C	Design			
C1	Create instances of the IED	SCT creates instances of the IED based on the icd files	☐	ok
C2	Map the LNodes	SCT maps the LNodes to the LN in the physical IED using the mapping defined in the ICD file	☐	Mapping does currently not use the mapping from the icd file

C3	Create communication section	SCT creates the communication section	☐	Is created together with the IED instance when the ICD file is imported
C4	Configure GOOSE and LN Inputs (ExtRef) based on the source Ref	SCT implements the ExtRefs for signals exchanged between the IEDs based on SourceRefs and configures GOOSE messages (Control block, dataset and GSE element in communication section)	☐	Configuration is done automatically in principle (during the test, not all signal connections where created however)
C5	Configure data flow to client devices	SCT configures reports based on specification	☐	
C6	Export SCD file	SCT is able to export SCD file	☐	

12.6 Test 212.2 – Configuration config 1, SS B (Siemens)

Specification: Schneider
IEDs: BC: Efacec, PROT: Ingeteam

	Test Step	Verification	Ok ☐☑☒	Remarks / Observations
A	File import			
A1	SCT imports SSD file	SCT is able to import SSD file	☑	
A2	SCT imports ICD file	SCT is able to import ICD file	☑	

B	ICD / SSD compare			
B1	Check that all specified LNs are implemented	SCT is able to show a relation between the specified LNs and the implemented LNs	☐	
B2	Check that all DO/DA are implemented	SCT identifies missing DO/DA	☐	
B3	Check service capabilities	SCT identifies limiting capabilities of IED	☐	
C	Design			
C1	Create instances of the IED	SCT creates instances of the IED based on the icd files	☑	Done with import
C2	Map the LNodes	SCT maps the LNodes to the LN in the physical IED using the mapping defined in the ICD file	☑	Is done automatically with the import based on the mapping in the icd file; however, that only works as there is just one bay and it only works if the names of the substation section match
C3	Create communication section	SCT creates the communication section	☑	
C4	Configure GOOSE and LN Inputs (ExtRef) based on the source Ref	SCT implements the ExtRefs for signals exchanged between the IEDs based on SourceRefs and configures GOOSE messages (Control block, dataset and GSE element in communication section)	☑	Done manually without considering the source ref
C5	Configure data flow to client devices	SCT configures reports based on specification	☐	

C6	Export SCD file	SCT is able to export SCD file	☑	

13 Annex L: Interoperability test results subset “Teleprotection exercise”, Config 2

13.1 Test 152.1 – Procurement SS A BPU (efacec)

Date of Test	13.7.21	Conf / Substation	Conf2 / SSA
SST	Helinks		
ICT	efacec	Device	BPU

	Test Step	Verification	Ok ☐☑☒	Remarks / Observations
A	File import			
A1	Import SSD file		☑	
B	IED design			
B1	Select functions from IED to match subset of functions specified by SSD	ICT is able to display functions from SSD and relate them to functions available in IED	☑	
B2	Map LNodes from specification to LNs from IED		☑	
B3	Copy LNode types from SSD to ICD		☒	
B4	Map DAs if needed	ICT is able to map DAs if LN Type does not fully support LNode Type	☑	
B5	Map inputs in case of later binding	In case of later binding, ICD fills in the ExtRefAddr	☑	

B6	Remove unneeded elements from substation section (LNodes not mapped and any dependencies)		☐	Needs to be done manually
B7	Export ICD file	ICT is able to produce ICD file	☑	
C	ICD file inspection			
C0	File validation against namespace	61850-6:2007B4 and 61850-6-100:2019A5	☒	Reference to 61850-6-100 version is missing Validation fails even when reference to 61850-6-100 is added (reason: see note on C6 below)
C1		Verify that IED Name and optionally substation name is “TEMPLATE”	☑	
C2	Verify step B2	Verify that required LNodes from specification are mapped	☑	
C3		Verify that LNodeSpecNaming is kept	☑	
C4		Verify that values from specification are configured as values in IED	☐☑☒	NA
C5	Verify step B3	Verify that lnTypes from specification are copied into ICD file and that lnType attribute of LNode is as from the ssd file	☒	
C6	Verify step B4	Verify the mapping of DAs that are implemented different	☒	The mapping is done but not in the 6-100 scl namespace

C7	Verify step B5	Verify the mapping of ExtRefAddr	☑	
C8	Verify step B6	Verify that all not mapped LNodes are removed from file, but for the mapped LNodes, the partial function structure is still present	☑	
C9		Verify, that LNode inputs do not contain srcRef attributes that point to something that is not in the file	☑	

13.2 Test 152.2 – Procurement SS A BCU (Ingeteam)

Date of Test	13.7.21	Conf / Substation	Conf 2 / SS-A
SST	Helinks		
ICT	Ingeteam	Device	BCU

	Test Step	**Verification**	**Ok** ☐☑☒	**Remarks / Observations**
A	File import			
A1	Import SSD file		☑	
B	IED design			
B1	Select functions from IED to match subset of functions specified by SSD	ICT is able to display functions from SSD and relate them to functions available in IED	☑	

B2	Map LNodes from specification to LNs from IED		☑	
B3	Copy LNode types from SSD to ICD		☒	
B4	Map DAs if needed	ICT is able to map DAs if LN Type does not fully support LNode Type	☐☑☒	NA
B5	Map inputs in case of later binding	In case of later binding, ICD fills in the ExtRefAddr	☐☑☒	NA – no later binding support
B6	Remove unneeded elements from substation section (LNodes not mapped and any dependencies)		☑	SourceRef element pointing to something that is not in file needs to be removed manually
B7	Export ICD file	ICT is able to produce ICD file	☑	
C	ICD file inspection			
C0	File validation against namespace	61850-6:2007B4 and 61850-6-100:2019A5	☐	File refers to 61850-6:2007B and 61850-6-100:2019A2 – after changing version reference, file does validate against 61850-6 required version, but does not validate against 61850-6-100 with the required versions, because the attribute “source” of SourceRef has been cleaned but not removed. 6-100 schema does not allow empty string on attribute “source”.
C1		Verify that IED Name and optionally substation name is “TEMPLATE”	☑	Substation name is SS-A

C2	Verify step B2	Verify that required LNodes from specification are mapped	☑	
C3		Verify that LNodeSpecNaming is kept	☑	
C4		Verify that values from specification are configured as values in IED	☐☑☒	NA
C5	Verify step B3	Verify that lnTypes from specification are copied into ICD file and that lnType attribute of LNode is as from the ssd file	☒	
C6	Verify step B4	Verify the mapping of DAs that are implemented different	☐☑☒	NA
C7	Verify step B5	Verify the mapping of ExtRefAddr	☐☑☒	NA
C8	Verify step B6	Verify that all not mapped LNodes are removed from file, but for the mapped LNodes, the partial function structure is still present	☑	
C9		Verify, that LNode inputs do not contain SourceRef attributes that point to something that is not in the file	☐	Was not removed correctly (see note on C0)

13.3 Test 152.3 – Procurement SS B BPU (Ingeteam)

Date of Test	13.7.2021	Conf / Substation	Conf2 / SS-B
SST	Schneider		
ICT	Ingeteam	Device	BPU

	Test Step	Verification	Ok ☐☑☒	Remarks / Observations
A	File import			
A1	Import SSD file		☑	
B	IED design			
B1	Select functions from IED to match subset of functions specified by SSD	ICT is able to display functions from SSD and relate them to functions available in IED	☑	
B2	Map LNodes from specification to LNs from IED		☑	
B3	Copy LNode types from SSD to ICD		☒	
B4	Map DAs if needed	ICT is able to map DAs if LN Type does not fully support LNode Type	☐	NA
B5	Map inputs in case of later binding	In case of later binding, ICD fills in the ExtRefAddr	☐	NA because no later bindding

B6	Remove unneeded elements from substation section (LNodes not mapped and any dependencies)		☑	
B7	Export ICD file	ICT is able to produce ICD file	☑	
C	ICD file inspection			
C0	File validation against namespace	61850-6:2007B4 and 61850-6-100:2019A5	☐	File refers to 61850-6:2007B and 61850-6-100:2019A2 – after changing version reference, file validates with the required versions
C1		Verify that IED Name and optionally substation name is “TEMPLATE”	☑	Substation name is SSB
C2	Verify step B2	Verify that required LNodes from specification are mapped	☑	
C3		Verify that LNodeSpecNaming is kept	☑	
C4		Verify that values from specification are configured as values in IED	☐☑☒	NA
C5	Verify step B3	Verify that lnTypes from specification are copied into ICD file and that lnType attribute of LNode is as from the ssd file	☒	
C6	Verify step B4	Verify the mapping of DAs that are implemented different	☐☑☒	NA

C7	Verify step B5	Verify the mapping of ExtRefAddr	☐☑☒	NA
C8	Verify step B6	Verify that all not mapped LNodes are removed from file, but for the mapped LNodes, the partial function structure is still present	☑	
C9		Verify, that LNode inputs do not contain SourceRef attributes that point to something that is not in the file	☑	

13.4 Test 152.4 – Procurement SS B BCU (efacec)

Date of Test	13.7.21	Conf / Substation	Conf2 / SSB
SST	Schneider		
ICT	efacec	Device	BCU

	Test Step	Verification	Ok ☐☑☒	Remarks / Observations
A	File import			
A1	Import SSD file		☑	
B	IED design			
B1	Select functions from IED to match subset of functions specified by SSD	ICT is able to display functions from SSD and relate them to functions available in IED	☑	ICT shows the functions structure from the specification to do the mapping of LNodes to LNs

B2	Map LNodes from specification to LNs from IED		☑	LNodes can be mapped on LN
B3	Copy LNode types from SSD to ICD		☒	
B4	Map DAs if needed	ICT is able to map DAs if LN Type does not fully support LNode Type	☐	NA
B5	Map inputs in case of later binding	In case of later binding, ICD fills in the ExtRefAddr	☑	ExtRefs are available in different LNs (GGIO), so the mapping of the inputs is on a different LN
B6	Remove unneeded elements from substation section (LNodes not mapped and any dependencies)		☐	Needs to be done manually
B7	Export ICD file	ICT is able to produce ICD file	☑	
C	ICD file inspection			
C0	File validation against namespace	61850-6:2007B4 and 61850-6-100:2019A5	☐	Version reference to 61850-6-100 is missing but it validates against that version
C1		Verify that IED Name and optionally substation name is “TEMPLATE”	☑	Substation kept SSB
C2	Verify step B2	Verify that required LNodes from specification are mapped	☑	
C3		Verify that LNodeSpecNaming is kept	☑	
C4		Verify that values from specification are configured as values in IED	☐	NA

C5	Verify step B3	Verify that InTypes from specification are copied into ICD file and that InType attribute of LNode is as from the ssd file	☒	
C6	Verify step B4	Verify the mapping of DAs that are implemented different		NA
C7	Verify step B5	Verify the mapping of ExtRefAddr	☑	
C8	Verify step B6	Verify that all not mapped LNodes are removed from file, but for the mapped LNodes, the partial function structure is still present	☐	Needs to done manually
C9		Verify, that LNode inputs do not contain srcRef attributes that point to something that is not in the file	☑	Was done manually

13.5 Test 212.1 – Configuration SS A (efacec)

Date of Test	14.7.21	Substation:	Conf 2 / SS-A
SST	Helinks	SCD	Efacec
ICT - BPU	Efacec	ICT - BCU	Ingeteam

	Test Step	Verification	☐☑☒	Remarks / Observations
A	File import			

A1	SCT imports SSD file	SCT is able to import SSD file	☑	
A2	SCT imports ICD file	SCT is able to import ICD file	☑	
B	ICD / SSD compare			
B1	Check that all specified LNs are implemented	SCT is able to show a relation between the specified LNs and the implemented LNs	☐☑☒	Not done
B2	Check that all DO/DA are implemented	SCT identifies missing DO/DA	☐☑☒	Not done
B3	Check service capabilities	SCT identifies limiting capabilities of IED	☐☑☒	Not done
C	Design			
C1	Create instances of the IED	SCT creates instances of the IED based on the icd files	☑	
C2	Map the LNodes	SCT maps the LNodes to the LN in the physical IED using the mapping defined in the ICD file	☑	
C3	Create communication section	SCT creates the communication section	☑	
C4	Configure GOOSE and LN Inputs (ExtRef) based on the source Ref	SCT implements the ExtRefs for signals exchanged between the IEDs based on SourceRefs and configures GOOSE messages (Control block, dataset and GSE element in communication section)	☑	

C5	Configure data flow to client devices	SCT configures reports based on specification	☐☑☒	NA
C6	Export SCD file	SCT is able to export SCD file	☑	
D	SCD file inspection			
D0	File validation against namespace	61850-6:2007B4 and 61850-6-100:2019A5	☒	Identification of version for 61850-6-100 is missing; the name used for the namespace is wrong. Once the expected namespace reference is added, 6-100 does not validate: in LNodeSpecNaming the attribute sLnInst shall not be an empty string.
D1	Validate step C1	Verify that all IEDs are present	☑	
D2	Validate step C2	Verify that LNodes from specification have a reference to an IED with the LN class from the IED and the original LNodeSpecNaming is kept	☑	
D3		Verify that mapping for DOS/DAS from icd are kept	☒	
D4		Verify that SourceRefs are filled in as provided by SSD file (note that they may have been suppressed by the icd file)	☑	

D5	Validate step C3	Verify that subnet has been created and all IEDs are connected with the access points and valid IP address is configured	☑	
D6	Validate step C4/C5	Verify that all SourceRefs pointing to another IED have an ExtRef which is complete and in line with the SourceRef and possible DOS/DAS mapping.	☒	ExtRef was only created for the for the IED from efacec (BPU) For BCU a connection was made to an existing ExtRef in a GGIO with matching pDO, pDA. However, as IED supports ICTBinding and no later binding was indicated in the icd file, the SCT shall create an ExtRef in the LN where it is expected by SourceRef.
D7		Verify that ExtRefAddr that where already filled in by the icd file are kept and has been completed	☒	
D8		Verify that all SourceRefs where no ExtRefAddr was filled in now have an ExtRefAddr that correspond to an ExtRef which is complete.	☒	
D9		Verify for al ExtRef, that the srcCBName refers to a control block of the type defined in serviceType of the ExtRef	☑	

D10		Verify that for all ExtRef the DO/DA referred to by ExtRef is in the dataset referred by the control block referred by the ExtRef	☑	
D11		Verify that the control block referred by the ExtRef has a reference to the subscriber IED	☑	
D12		Verify the presence of the GSE or SMV element under the access point of the publishing IED for all GOOSE and SV control blocks	☑	

13.6 Test 212.1 – Configuration SS B (Ingeteam)

Date of Test	14.7.21	Substation:	Conf 2 / SS-B
SST	Schneider	SCD	Ingeteam
ICT - BPU	Ingeteam	ICT - BCU	efacec

	Test Step	**Verification**	☐☑☒	**Remarks / Observations**
A	File import			
A1	SCT imports SSD file	SCT is able to import SSD file	☑	
A2	SCT imports ICD file	SCT is able to import ICD file	☑	
B	ICD / SSD compare			

B1	Check that all specified LNs are implemented	SCT is able to show a relation between the specified LNs and the implemented LNs	☐☑☒	Not done
B2	Check that all DO/DA are implemented	SCT identifies missing DO/DA	☐☑☒	Not done
B3	Check service capabilities	SCT identifies limiting capabilities of IED	☐☑☒	Not done
C	Design			
C1	Create instances of the IED	SCT creates instances of the IED based on the icd files	☑	
C2	Map the LNodes	SCT maps the LNodes to the LN in the physical IED using the mapping defined in the ICD file	☑	
C3	Create communication section	SCT creates the communication section	☑	
C4	Configure GOOSE and LN Inputs (ExtRef) based on the source Ref	SCT implements the ExtRefs for signals exchanged between the IEDs based on SourceRefs and configures GOOSE messages (Control block, dataset and GSE element in communication section)	☑	
C5	Configure data flow to client devices	SCT configures reports based on specification	☐☑☒	NA
C6	Export SCD file	SCT is able to export SCD file	☑	

D	SCD file inspection			
D0	File validation against namespace	61850-6:2007B4 and 61850-6-100:2019A5	☒	Identification of version for 61850-6-100 is missing; the name used for the namespace is wrong. Once added, validation is ok.
D1	Validate step C1	Verify that all IEDs are present	☑	
D2	Validate step C2	Verify that LNodes from specification have a reference to an IED with the LN class from the IED and the original LNodeSpecNaming is kept	☒	LNodes related to functions from BPU are missing
D3		Verify that mapping for DOS/DAS from icd are kept	☐☑☒	NA
D4		Verify that SourceRefs are filled in as provided by SSD file (note that they may have been suppressed by the icd file)	☒	SourceRefs that have been eliminated by icd files have not been restored
D5	Validate step C3	Verify that subnet has been created and all IEDs are connected with the access points and valid IP address is configured	☑	

D6	Validate step C4/C5	Verify that all SourceRefs pointing to another IED have an ExtRef which is complete and in line with the SourceRef and possible DOS/DAS mapping.	☒	No ExtRefs filled in. May have been missed as focus was for exchange between substations
D7		Verify that ExtRefAddr that where already filled in by the icd file are kept and has been completed	☒	
D8		Verify that all SourceRefs where no ExtRefAddr was filled in now have an ExtRefAddr that correspond to an ExtRef which is complete.	☒	
D9		Verify for all ExtRef, that the srcCBName refers to a control block of the type defined in serviceType of the ExtRef	☐☑☒	NA as no ExtRef. But GOOSE are correctly engineered
D10		Verify that for all ExtRef the DO/DA referred to by ExtRef is in the dataset referred by the control block referred by the ExtRef	☐☑☒	NA
D11		Verify that the control block referred by the ExtRef has a reference to the subscriber IED	☐☑☒	NA

D12		Verify the presence of the GSE or SMV element under the access point of the publishing IED for all GOOSE and SV control blocks	☑	

14 Annex M – Interoperability test results subset “Teleprotection exercise”, Config 1

14.1 Test 10.1 – Specification SS A (Helinks)

Date of Test	11.10.21/13.10.21	Conf / Substation	Conf 1 -SSA
SST	Helinks		
Expected Namespace		61850-6	61850-6-100
		2007B4	2019A6

	Test Step	Verification	Ok ☐☑☒	Remarks / Observations
A	Function specification			
A1	Export function specification from (FTD files)			
B	System specification			
B1	SST imports function specification (FTD files)	SST is able to import the function specification	☑	
B2	SST completes incoherent data type templates	SST is able to ask for the data model namespace to be used and modify / complete enumtypes and struct types. This requires user interaction to select - What values out of the enum to support? - What child elements to support	☑	

B3	Design single line diagram in SST		☑	
B4	Instantiate functions	SST uses the FTD file with the logical nodes to create and populate the function structure and uses the data objects from the FTD file to create the LN templates Instantiation includes - Selecting the element where the function shall be instantiated and optionally update function name - User interaction to add power system resource references where needed	☑	
B5	Define signal flow	SST supports the creation of the signal flow based on the inputs defined in the FTD file. - Connections are typically made by user interaction; if LNs are allocated to a power system resource this can be used by the tool to automatically create the connection	☑	
B6	Define pre-configuration values	SST allows the user to configure default values for e.g. settings	☒	The FTD did not have the parameter to be set, so it was added by the tool, but values could not be configured

B7	Define virtual IEDs (only configuration 1)	SST supports the allocation of LNs to virtual IEDs. Logical devices are created for each function	☑	Helinks STS cannot allocate subfunctions from the same function to multiple IEDs – as a work around, the function structure provided from the FTD has to be modified
B8	Specify IED requirements in service section	SST supports the specification of requirements on the IED in the service section	☑	Service section can be loaded through a template or can be edited
B9	Optionally configure signal flow between virtual IEDs	SST creates: - ExtRef based on SourceRef - Optionally GOOSE messages; required if user has specific requirements	☑	
B10	Export SSD	SST is able to produce SSD file	☑	
B11	Export ISD (only configuration 1)	SST is able to produce ISD files	☑	
C	SSD file inspection			Was re-done based on a new version of the file
C0	File validation against namespace		☒	Reference to version and revision of 6-100 namespace is missing LDName for an LD0 was empty, but should be removed if no value needed
C1	Verify step B3	In the SSD file, verify the substation structure with the voltage levels and bays as defined in the project specification	☑	

C2	Verify step B4	In the SSD file, verify that the function structure has been created using the elements from the FTD file - Verify that the LNodeSpecNaming element is present - Verify that power system reference is present	☑	
C3	Verify step B5	In the SSD file, verify that the inputs have been connected (SourceRef)	☑	
C4	Verify step B6	In the SSD file, verify that DOS/DAS elements are filled in	☐	N/A as they were not added by the tool (see Step B6 above)
C5	Verify step B7 (only configuration 1)	In the SSD file, verify that the virtual IEDs are defined	☑	
C6	Verify step B8	Check the service section of the virtual IED	☑	
C7	Verify step B9 (optional)	In the SSD file, verify that the virtual IEDs include signal flow configuration (ExtRef and optionally GOOSE configuration)	☑	
C8	Check SSD file	Run SSD file through various SCL checkers and validators; report results for documentation	☑	

D	ISD file inspection (only configuration 1)			
D0	File validation against namespace		☒	Reference to version and revision of 6-100 namespace is missing LDName for an LD0 was empty, but should be removed if no value needed
D1	Verify step B4	Verify, that the LNodes that are mapped to LNs in the virtual IED are present in the substation structure	☑	
D2	Verify step B6	Verify that DOS/DAS have required values and those values are available in the DOI/DAI	☐	N/A as they were not added by the tool (see Step B6 above)
D3	Verify step B7	Verify in the LD / LN Structure that all LNs are present and compare with the function / subfunction structure	☑	
D4	Verify step B9 (optional)	Verify that SourceRef have been translated in ExtRef with service type specified	☒	ExtRef is not empty – it still shows a link to another IED

14.2 Test 10.2 – Specification SS B (Schneider)

Date of Test	11.10.2021	Conf / Substation	Conf 1 – SS-B
SST	Schneider		

Expected Namespace	61850-6:	61850-6-100
	2007B4	2019A6

	Test Step	Verification	Ok ☐☑☒	Remarks / Observations
A	Function specification			
A1	Export function specification from (FTD files)			
B	System specification			
B1	SST imports function specification (FTD files)	SST is able to import the function specification	☑	
B2	SST completes incoherent data type templates	SST is able to ask for the namespace to be used and modify / complete enumtypes and struct types. This requires user interaction to select - What values out of the enum to support? - What child elements to support	☒	Struct Type already complete in IST export Enum Types are not in IST export and have not been added by tool
B3	Design single line diagram in SST		☑	

B4	Instantiate functions	SST uses the FTD file with the logical nodes to create and populate the function structure and uses the data objects from the FTD file to create the LN templates Instantiation includes - Selecting the element where the function shall be instantiated and optionally update function name - User interaction to add power system resource references where needed	☑	Power system resource can not be added
B5	Define signal flow	SST supports the creation of the signal flow based on the inputs defined in the FTD file. - Connections are typically made by user interaction; if LNs are allocated to a power system resource this can be used by the tool to automatically create the connection	☑	Is done by user interaction
B6	Define pre-configuration values	SST allows the user to configure default values for e.g. settings	☒	Not yet supported by tool
B7	Define virtual IEDs (only configuration 1)	SST supports the allocation of LNs to virtual IEDs. Logical devices are created for each function	☑	In order to create the LD names, manual configuration has to be done.

B8	Specify IED requirements in service section	SST supports the specification of requirements on the IED in the service section	☒	SST uses a predefined set
B9	Optionally configure signal flow between virtual IEDs	SST creates: - ExtRef based on SourceRef - Optionally GOOSE messages; required if user has specific requirements	☑	ExtRefs are created automatically
B10	Export SSD	SST is able to produce SSD file	☑	
B11	Export ISD (only configuration 1)	SST is able to produce ISD files	☑	
C	SSD file inspection			
C0	File validation against namespace		☑	
C1	Verify step B3	In the SSD file, verify the substation structure with the voltage levels and bays as defined in the project specification	☑	

C2	Verify step B4	In the SSD file, verify that the function structure has been created using the elements from the FTD file - Verify that the LNodeSpecNaming element is present - Verify that power system reference is present	☒	Spec naming and power system reference are currently not supported
C3	Verify step B5	In the SSD file, verify that the inputs have been connected	☑	
C4	Verify step B6	In the SSD file, verify that DOS/DAS elements are filled in	☐	N/A as not supported by tool (see step B6)
C5	Verify step B7 (only configuration 1)	In the SSD file, verify that the virtual IEDs are defined	☑	
C6	Verify step B8	Check the service section of the virtual IED	☐	Not done
C7	Verify step B9 (optional)	In the SSD file, verify that the virtual IEDs include signal flow configuration (ExtRef and optionally GOOSE configuration)	☑	
C8	Check SSD file	Run SSD file through various SCL checkers and validators; report results for documentation	☐☑☒	RiseClips: Enumeration type is not present (step B2)

D	ISD file inspection (only configuration 1)			
D0	File validation against namespace		☑	
D1	Verify step B4	Verify, that the LNodes that are mapped to LNs in the virtual IED are present in the substation structure	☑	
D2	Verify step B6	Verify that DOS/DAS have required values and those values are available in the DOI/DAI	☐	N/A as not supported by tool (see step B6)
D3	Verify step B7	Verify in the LD / LN Structure that all LNs are present and compare with the function / subfunction structure	☑	
D4	Verify step B9 (optional)	Verify that SourceRef have been translated in ExtRef with service type specified	☒	

14.3 Test 12.1a – Specification SS A subscribing from SS B

Date of Test	13.10.21	Conf / Substation	Conf 1, SS-A
SST-p	Schneider	SST-s	Helinks
Expected Namespace		61850-6:	61850-6-100
		2007B4	2019A6

	Test Step	Verification	Ok ☐☑☒	Remarks / Observations
A	File export (SSD-p)			
A1	Export (subset) of SSD			a subset based on mutual understanding of what is needed in the other substation is exported
B	SSD file inspection (SSD-p)			
B0	File validation against namespace		☑	
B1	Verify that sending function is present in exported file	Verify in the SSD-p file that the function with the signal to subscribe is present	☑	
C	Specify subscription in SST-s			
C1	SST-s imports subset of SSD-p	SST is able to import the SSD file	☑	
C2	SST-s adds the sending function from other project to own substation	SST is able to add the second substation with the relevant parts for the publishing function	☑	
C3	SST-s connects the signal from the other SS to the local input		☑	
C4	Export SSD-s	SST-s exports SSD file with the SourceRef receiving from the other substation connected	☑	
D	SSD file inspection (SSD-s)			
D0	File validation against namespace		☑	6-100 version/revision/release are not present LDName element is not empty

D1	Verify step C2	Verify that the other substation with the sending function is included in SSD-s	☑	
D2	Verify step C3	Verify that the connection is made from the SourceRef to the publishing function / signal	☑	
E	Configure publishing in SST-p			Not done in OSMOSE
E1	Import SSD-s with connected SourceRef	SST-p is able to import SSD-s	☐☑☒	
E2	SST-p adds the subscribing function from other project to own substation	SST is able to add the second substation with the relevant parts from the subscribing function	☐☑☒	
E3	SST-p adds the subscribing IED from other project	SST is able to identify and add the subscribing IED from the function	☐☑☒	
E4	SST-p creates the GOOSE message for the other SS		☐☑☒	
E5	Export SSD-p	SST-p is able to export the SSD-p with GOOSE configuration	☐☑☒	
F	SSD file inspection (SSD-p)			Not done in OSMOSE
F0	File validation against namespace		☐☑☒	
F1	Verify step E2	Verify that the other substation with the subscribing function is included in SSD-p	☐☑☒	

F2	Verify step E3	Verify that the subscribing IED is present	☐☑☒	
F3	Verify step E4	Verify that the GOOSE message is configured	☐☑☒	
F4	Verify step E4	Verify that the IED from other SS has been added as subscribing IED	☐☑☒	
G	Configure Subscription			Not done in OSMOSE
G1	Import SSD-p with configured GOOSE	SST-s is able to import SSD-p	☐☑☒	
G2	SST-s adds the publishing IED from other project	SST is able to identify and add the publishing IED from the function	☐☑☒	
G3	SST-s creates the GOOSE subscription	SST-s is updating the ExtRef with the GOOSE configuration	☐☑☒	
G4	Export SSD-s	SST-s is able to export the SSD-s with GOOSE subscription configured	☐☑☒	
H	SSD file inspection (SSD-s)			Not done in OSMOSE
H0	File validation against namespace		☐☑☒	
H1	Verify step G2	Verify that the publishing IED is present	☐☑☒	
H2	Verify step G3	Verify that the ExtRef has been updated with the configuration of the publishing GOOSE	☐☑☒	

14.4 Test 12.1b – Specification SS B subscribing from SS A

Date of Test	12.10.2021	Conf / Substation	Conf 1, SS-B
SST-p	Helinks	SST-s	Schneider
Expected Namespace		61850-6	61850-6-100
		2007B4	2019A6

	Test Step	Verification	Ok ☐☑☒	Remarks / Observations
A	File export (SSD-p)			
A1	Export (subset) of SSD			a subset based on mutual understanding of what is needed in the other substation is exported
B	SSD file inspection (SSD-p)			
B0	File validation against namespace		☑	
B1	Verify that sending function is present in exported file	Verify in the SSD-p file that the function with the signal to subscribe is present	☑	
C	Specify subscription in SST-s			
C1	SST-s imports subset of SSD-p	SST is able to import the SSD file	☑	Before importing, 6-100 release needs to be changed to 2019A2, as Schneider SST does not yet support 2019A6 (which is not used in OSMOSE context) Decided not to import virtual IEDs

C2	SST-s adds the sending function from other project to own substation	SST is able to add the second substation with the relevant parts for the publishing function	☑	
C3	SST-s connects the signal from the other SS to the local input		☑	
C4	Export SSD-s	SST-s exports SSD file with the SourceRef receiving from the other substation connected	☑	
D	SSD file inspection (SSD-s)			
D0	File validation against namespace		☑	Schneider SST is supporting 6-100:2019A2
D1	Verify step C2	Verify that the other substation with the sending function is included in SSD-s	☑	
D2	Verify step C3	Verify that the connection is made from the SourceRef to the publishing function / signal	☑	
E	Configure publishing in SST-p			Not done in OSMOSE
E1	Import SSD-s with connected SourceRef	SST-p is able to import SSD-s	☐☑☒	
E2	SST-p adds the subscribing function from other project to own substation	SST is able to add the second substation with the relevant parts from the subscribing function	☐☑☒	
E3	SST-p adds the subscribing IED from other project	SST is able to identify and add the subscribing IED from the function	☐☑☒	

E4	SST-p creates the GOOSE message for the other SS		☐☑☒	
E5	Export SSD-p	SST-p is able to export the SSD-p with GOOSE configuration	☐☑☒	
F	SSD file inspection (SSD-p)			Not done in OSMOSE
F0	File validation against namespace		☐☑☒	
F1	Verify step E2	Verify that the other substation with the subscribing function is included in SSD-p	☐☑☒	
F2	Verify step E3	Verify that the subscribing IED is present	☐☑☒	
F3	Verify step E4	Verify that the GOOSE message is configured	☐☑☒	
F4	Verify step E4	Verify that the IED from other SS has been added as subscribing IED	☐☑☒	
G	Configure Subscription			Not done in OSMOSE
G1	Import SSD-p with configured GOOSE	SST-s is able to import SSD-p	☐☑☒	
G2	SST-s adds the publishing IED from other project	SST is able to identify and add the publishing IED from the function	☐☑☒	
G3	SST-s creates the GOOSE subscription	SST-s is updating the ExtRef with the GOOSE configuration	☐☑☒	

G4	Export SSD-s	SST-s is able to export the SSD-s with GOOSE subscription configured	☐☑☒	
H	SSD file inspection (SSD-s)			Not done in OSMOSE
H0	File validation against namespace		☐☑☒	
H1	Verify step G2	Verify that the publishing IED is present	☐☑☒	
H2	Verify step G3	Verify that the ExtRef has been updated with the configuration of the publishing GOOSE	☐☑☒	

14.5 Test 151.1 – Procurement SS A BPU (Siemens)

Date of Test	13.10.21	Conf / Substation	Conf 1, SS-A
SST	Helinks		
ICT	Siemens	Device	
Expected Namespace		61850-6	61850-6-100
		2007B4	2019A6

	Test Step	Verification	Ok ☐☑☒	Remarks / Observations
A	File import			
A1	Import ISD file		☑	File type needs to be renamed to ssd
B	IED design			

B1	Select functions from IED to match functions specified by ISD	ICT is able to display functions from ISD and relate them to functions available in IED	☑	
B2	Map LNodes from specification to LNs from IED		☑	
B3	Copy LNode types from ISD to ICD		☐☑☒	
B4	Map DAs if needed	ICT is able to map DAs if LN Type does not fully support LNode Type	☐	
B5	Map inputs in case of later binding	In case of later binding, ICD fills in the ExtRefAddr in the SourceRef	☒	Not supported by the tool – needs to be done manually
B6	Remove unneeded elements from substation section (LNodes not mapped and any dependencies)		☐☑☒	
B7	Export ICD file	ICT is able to produce ICD file	☑	Exports as iid file
C	ICD file inspection			
C0	File validation against namespace		☑	Version/Revision/Release of 6-100 is missing, as well as schema location
C1		Verify that IED Name and optionally substation name is “TEMPLATE”	☐	NA as iid file was exported
C2	Verify step B2	Verify that all LNodes from specification are mapped	☑	

C3		Verify that LNodeSpecNaming is kept	☑	
C4		Verify that values from specification are configured as values in IED	☐	NA
C5	Verify step B3	Verify that lnTypes from specification are available in ICD file as equivalent types and that lnType attribute of LNode points to the corresponding lnType (optimisation of types is allowed to be done by the ICT)	☒	Tool has replaced LNode type with the type from the implementation
C6	Verify step B4	Verify the mapping of DAs that are implemented different	☐	NA
C7	Verify step B5	Verify in SourceRef the mapping of ExtRefAddr	☐	Not supported by tool (see step B5)
C8	Verify step B6	Verify that all not mapped LNodes are removed from file, but for the mapped LNodes, the partial function structure is still present	☑	
C9		Verify, that LNode inputs do not contain SourceRef attributes that point to something that is not in the file	☒	SourceRef is pointing to function which has been removed

14.6 Test 151.2 – Procurement SS A BCU (efacec)

Date of Test	13.10.21	Conf / Substation	Conf 1 / SS-A
SST	Helinks		
ICT	efacec	Device	
Expected Namespace		61850-6	61850-6-100
		2007B4	2019A6

	Test Step	Verification	Ok ☐☑☒	Remarks / Observations
A	File import			
A1	Import ISD file		☑	Changed extensions to ssd
B	IED design			
B1	Select functions from IED to match functions specified by ISD	ICT is able to display functions from ISD and relate them to functions available in IED	☐	Directly mapping the LNodes
B2	Map LNodes from specification to LNs from IED		☑	
B3	Copy LNode types from ISD to ICD		☐	Done with export
B4	Map DAs if needed	ICT is able to map DAs if LN Type does not fully support LNode Type	☐	NA as LNs match
B5	Map inputs in case of later binding	In case of later binding, ICD fills in the ExtRefAddr in the SourceRef	☒	Done manually

B6	Remove unneeded elements from substation section (LNodes not mapped and any dependencies)		☐	Done with export
B7	Export ICD file	ICT is able to produce ICD file	☑	
C	ICD file inspection			
C0	File validation against namespace		☑	Version/Revision/Release for 6-100 is not in the file
C1		Verify that IED Name and optionally substation name is "TEMPLATE"	☑	SS name is kept
C2	Verify step B2	Verify that all LNodes from specification are mapped	☑	
C3		Verify that LNodeSpecNaming is kept	☑	
C4		Verify that values from specification are configured as values in IED	☐	NA as they were not in the isd file
C5	Verify step B3	Verify that lnTypes from specification are available in ICD file as equivalent types and that lnType attribute of LNode points to the corresponding lnType (optimisation of types is allowed to be done by the ICT)	☑	

C6	Verify step B4	Verify the mapping of DAs that are implemented different	☐	NA as LNs matched
C7	Verify step B5	Verify in SourceRef the mapping of ExtRefAddr	☑	
C8	Verify step B6	Verify that all not mapped LNodes are removed from file, but for the mapped LNodes, the partial function structure is still present	☑	
C9		Verify, that LNode inputs do not contain SourceRef attributes that point to something that is not in the file	☑	

14.7 Test 151.3 – Procurement SS B BPU (efacec)

Date of Test	12.10.21	Conf / Substation	Conf 1, SS-B
SST	Schneider		
ICT	efacec	Device	
Expected Namespace		61850-6	61850-6-100
		2007B4	2019A6

	Test Step	Verification	Ok ☐☑☒	Remarks / Observations
A	File import			

A1	Import ISD file		☐☑☒	File type needs to be renamed to ssd
B	IED design			
B1	Select functions from IED to match functions specified by ISD	ICT is able to display functions from ISD and relate them to functions available in IED	☑	
B2	Map LNodes from specification to LNs from IED		☑	The mapping of RxPSCH is on a GGIO
B3	Copy LNode types from ISD to ICD		☑	Merged when output is produced
B4	Map DAs if needed	ICT is able to map DAs if LN Type does not fully support LNode Type	☑	PSCH.Op.general is mapped on GGIO.Ind02.stVal
B5	Map inputs in case of later binding	In case of later binding, ICD fills in the ExtRefAddr in the SourceRef	☑	Input to TxPSCH has been mapped to an input to a GGIO
B6	Remove unneeded elements from substation section (LNodes not mapped and any dependencies)		☑	Done when file is generated
B7	Export ICD file	ICT is able to produce ICD file	☑	
C	ICD file inspection			
C0	File validation against namespace		☒	Version, revision and release of the 6-100 is missing xmlns for 6-100 points to a wrong schema DAS element, attribute “MappedDAName” had wrong pattern

C1		Verify that IED Name and optionally substation name is “TEMPLATE”	☑	Substation name has been kept
C2	Verify step B2	Verify that all LNodes from specification are mapped	☑	
C3		Verify that LNodeSpecNaming is kept	☑	
C4		Verify that values from specification are configured as values in IED	☐	NA – value is only configured for BCU
C5	Verify step B3	Verify that lnTypes from specification are available in ICD file as equivalent types and that lnType attribute of LNode points to the corresponding lnType (optimisation of types is allowed to be done by the ICT)	☑	Some lnTypes where optimized
C6	Verify step B4	Verify the mapping of DAs that are implemented different	☒	The element DOS/DAS is on the wrong place – should be in the private
C7	Verify step B5	Verify in SourceRef the mapping of ExtRefAddr	☑	
C8	Verify step B6	Verify that all not mapped LNodes are removed from file, but for the mapped LNodes, the partial function structure is still present	☑	

C9		Verify, that LNode inputs do not contain SourceRef attributes that point to something that is not in the file	☑	

14.8 Test 151.4 – Procurement SS B BCU (Ingeteam)

Date of Test		Conf / Substation	Config 1, SS-B
SST	Schneider		
ICT	Ingeteam	Device	
Expected Namespace		61850-6	61850-6-100
		2007B4	2019A6

	Test Step	Verification	Ok ☐☑☒	Remarks / Observations
A	File import			
A1	Import ISD file		☑	
B	IED design			
B1	Select functions from IED to match functions specified by ISD	ICT is able to display functions from ISD and relate them to functions available in IED	☐	Is directly done on the LNodes
B2	Map LNodes from specification to LNs from IED		☑	
B3	Copy LNode types from ISD to ICD		☑	

B4	Map DAs if needed	ICT is able to map DAs if LN Type does not fully support LNode Type	☐	NA
B5	Map inputs in case of later binding	In case of later binding, ICD fills in the ExtRefAddr in the SourceRef	☒	Not supported
B6	Remove unneeded elements from substation section (LNodes not mapped and any dependencies)		☑	Done with export
B7	Export ICD file	ICT is able to produce ICD file	☑	
C	ICD file inspection			
C0	File validation against namespace		☑	Tool supports only 6-100:2019A2 File type is .cid
C1		Verify that IED Name and optionally substation name is “TEMPLATE”	☑	It has to be done manually
C2	Verify step B2	Verify that all LNodes from specification are mapped	☑	
C3		Verify that LNodeSpecNaming is kept	☑	
C4		Verify that values from specification are configured as values in IED	☐	Not done; shall be done by SCT

C5	Verify step B3	Verify that lnTypes from specification are available in ICD file as equivalent types and that lnType attribute of LNode points to the corresponding lnType (optimisation of types is allowed to be done by the ICT)	☒	Reference has been kept – but the type is not kept in the data type template section
C6	Verify step B4	Verify the mapping of DAs that are implemented different	☐	NA
C7	Verify step B5	Verify in SourceRef the mapping of ExtRefAddr	☐	NA, Not supported by tool (see step B5)
C8	Verify step B6	Verify that all not mapped LNodes are removed from file, but for the mapped LNodes, the partial function structure is still present	☑	
C9		Verify, that LNode inputs do not contain SourceRef attributes that point to something that is not in the file	☑	

14.9 Test 211.1 – Configuration SS A (Ingeteam)

Date of Test	13.10.21	Substation:	Conf 1, SS-A
SST	Helinks	SCD	Ingeteam
ICT - BPU	Siemens	ICT - BCU	Efacec

Expected Namespace	61850-6	61850-6-100
	2007B4	2019A6

	Test Step	Verification	Ok ☐☑☒	Remarks / Observations
A	File import			
A1	SCT imports SSD file	SCT is able to import SSD file	☑	
A2	SCT imports ISD file for ISD/ICD compare	SCT is able to import ISD file	☐☑☒	Not done
A3	SCT imports ICD file	SCT is able to import ICD file	☑	
B	ICD / ISD compare			Not done
B1	Check that all specified LNs are implemented	SCT is able to show a relation between the specified LNs and the implemented LNs	☐☑☒	
B2	Check that all DO/DA are implemented	SCT identifies missing DO/DA	☐☑☒	
B3	Check service capabilities	SCT identifies limiting capabilities of IED	☐☑☒	
C	Design			
C1	Implement virtual IED with physical IED	SCT replaces the virtual IED completely with the physical IED - The service section - The complete data model	☑	

C2	Update the mapping of the LNodes	SCT updates the mapping of the LNodes to the LN in the physical IED using the mapping defined in the ICD file	☑	
C3 (a)	If communication section is already configured in SSD file, update Access point references	SCT updates access point references in the communication section to match the physical IED	☑	
C3 (b)	Create communication section, if none is specified in SSD	SCT creates the communication section	☐	
C4 (a)	If GOOSE configuration is present in SSD file and IED supports requested capabilities, update data flow for protection and control schemes	SCT uses the existing configuration of GOOSE (Control block, dataset and GSE element in communication section) and ExtRef but updates all the references to match the physical IED	☒	ExtRefAddr at SourceRef is not updated ExtRef is not updated
C4 (b)	Configure GOOSE and LN Inputs (ExtRef) based on the source Ref	SCT implements the ExtRefs for signals exchanged between the IEDs based on SourceRefs and configures GOOSE messages (Control block, dataset and GSE element in communication section)	☐	
C5	Configure data flow to client devices	SCT configures reports based on specification	☐	N/A
C6	Export SCD file	SCT is able to export SCD file	☐☑☒	

D	SCD file inspection			
D0	File validation against namespace		☑	There are two different prefixes for the 6-100 namespaces in the file. This is not allowed, but as one of them refers to a wrong namespace file, it is not a problem
D1	Validate step C1	Verify that all IEDs are present	☑	
D2	Validate step C2	Verify that LNodes from specification have a reference to an IED with the LN class from the IED and the original LNodeSpecNaming is kept	☑	
D3		Verify that mapping for DOS/DAS from icd are kept	☐	NA
D4		Verify that SourceRefs are filled in as provided by SSD file (note that they may have been suppressed by the icd file)	☒	SourceRefs have been removed
D5	Validate step C3	Verify that subnet has been created and all IEDs are connected with the access points and valid IP address is configured	☒	There is one additional subnetwork from SSD as mapping was not done correctly in the SCT The subnetwork created by the tool is correct

D6	Validate step C4/C5	Verify that all SourceRefs pointing to another IED have an ExtRef which is complete and in line with the SourceRef and possible DOS/DAS mapping.	☐	NA as tool did not create (see step B5)
D7		Verify that ExtRefAddr that where already filled in by the icd file are kept and has been completed	☒	ExtRefAddr have not been imported
D8		Verify that all SourceRefs where no ExtRefAddr was filled in now have an ExtRefAddr that correspond to an ExtRef which is complete.	☐	NA as tool did not create (see step B5)
D9		Verify for al ExtRef, that the srcCBName refers to a control block of the type defined in serviceType of the ExtRef	☐	NA as tool did not create (see step B5)
D10		Verify that for all ExtRef the DO/DA referred to by ExtRef is in the dataset referred by the control block referred by the ExtRef	☐	NA as tool did not create (see step B5)
D11		Verify that the control block referred by the ExtRef has a reference to the subscriber IED	☑	

D12		Verify the presence of the GSE or SMV element under the access point of the publishing IED for all GOOSE and SV control blocks	☒	The one for PROT is missing

14.10Test 211.2 – Configuration SS B (Helinks)

Date of Test	13.10.2021	Substation:	Conf 1, SS-B
SST	Schneider	SCD	Helinks
ICT - BPU	Efacec	ICT - BCU	Ingeteam
Expected Namespace		61850-6	61850-6-100
		2007B4	2019A6

	Test Step	**Verification**	**Ok** ☐☑☒	**Remarks / Observations**
A	File import			
A1	SCT imports SSD file	SCT is able to import SSD file	☑	
A2	SCT imports ISD file for ISD/ICD compare	SCT is able to import ISD file	☐☑☒	Not done
A3	SCT imports ICD file	SCT is able to import ICD file	☑	
B	ICD / ISD compare			Not done
B1	Check that all specified LNs are implemented	SCT is able to show a relation between the specified LNs and the implemented LNs	☐☑☒	

B2	Check that all DO/DA are implemented	SCT identifies missing DO/DA	☐☑☒	
B3	Check service capabilities	SCT identifies limiting capabilities of IED	☐☑☒	
C	Design			
C1	Implement virtual IED with physical IED	SCT replaces the virtual IED completely with the physical IED - The service section - The complete data model	☑	
C2	Update the mapping of the LNodes	SCT updates the mapping of the LNodes to the LN in the physical IED using the mapping defined in the ICD file	☒	Is done manually As tool does not support do individually map subfunctions, the specified functions had to be split
C3 (a)	If communication section is already configured in SSD file, update Access point references	SCT updates access point references in the communication section to match the physical IED	☑	
C3 (b)	Create communication section, if none is specified in SSD	SCT creates the communication section	☐☑☒	NA
C4 (a)	If GOOSE configuration is present in SSD file and IED supports requested capabilities, update data flow for protection and control schemes	SCT uses the existing configuration of GOOSE (Control block, dataset and GSE element in communication section) and ExtRef but updates all the references to match the physical IED	☑	Update of ExtRef needs to be done manually Dataset needs to be manually updated for DO that was mapped differently Other elements can be added by copying from SSD

C4 (b)	Configure GOOSE and LN Inputs (ExtRef) based on the source Ref	SCT implements the ExtRefs for signals exchanged between the IEDs based on SourceRefs and configures GOOSE messages (Control block, dataset and GSE element in communication section)	☐☑☒	NA
C5	Configure data flow to client devices	SCT configures reports based on specification	☐☑☒	NA
C6	Export SCD file	SCT is able to export SCD file	☑	
D	SSD file inspection			
D0	File validation against namespace		☑	Version/Revision/Release of 6-100 is not present
D1	Validate step C1	Verify that all IEDs are present	☑	
D2	Validate step C2	Verify that LNodes from specification have a reference to an IED with the LN class from the IED and the original LNodeSpecNaming is kept	☑	LNodeSpecNaming has been updated, because specification has been updated (see step C2 above). Better approach would have been, not to update the Spec, but add new functions with new LNodes with no LNodeSpecNaming and not map the LNodes from the original function
D3		Verify that mapping for DOS/DAS from icd are kept	☑	
D4		Verify that SourceRefs are filled in as provided by SSD file (note that they may have been suppressed by the icd file)	☒	Tool did not import the sourceRef when they do not belong to an application

D5	Validate step C3	Verify that subnet has been created and all IEDs are connected with the access points and valid IP address is configured	☑	
D6	Validate step C4/C5	Verify that all SourceRefs pointing to another IED have an ExtRef which is complete and in line with the SourceRef and possible DOS/DAS mapping.	☑	ExtRef are there (but no SourceRef, see D4)
D7		Verify that ExtRefAddr that where already filled in by the icd file are kept and has been completed	☐	NA as there are no SourceRef in the file
D8		Verify that all SourceRefs where no ExtRefAddr was filled in now have an ExtRefAddr that correspond to an ExtRef which is complete.	☐	NA as there are no SourceRef in the file
D9		Verify for al ExtRef, that the srcCBName refers to a control block of the type defined in serviceType of the ExtRef	☑	
D10		Verify that for all ExtRef the DO/DA referred to by ExtRef is in the dataset referred by the control block referred by the ExtRef	☑	

D11		Verify that the control block referred by the ExtRef has a reference to the subscriber IED	☑	
D12		Verify the presence of the GSE or SMV element under the access point of the publishing IED for all GOOSE and SV control blocks	☑	
D13	Check SCD file	Run SCD file through various SCL checkers and validators; report results for documentation	☑	

14.11 Test 22.1a – Configuration SS A subscribing from SS B

Date of Test	14.10.2021	Substation	SS-A
SCD-s	Ingeteam	SCD-p	Helinks
Expected Namespace		61850-6	61850-6-100
		2007B4	2019A6

	Test Step	Verification	Ok ☐☑☒	Remarks / Observations
A	File export / import			
A1	SCT-s exports the SED file (SED-s)	SCT is able to import SED file	☑	
B	SED file inspection (SED-s)			
B0	File validation against namespace		☒	Errors with FCDA in datasets, LNInst element with empty values

B1		Verify in the file the presence of the subscribing IED (IED-s)	☑	
B2		Verify in the file the presence of the subscribing function with filled in SourceRef and the publishing function	☑	Publishing function not included
B3	If later binding is supported by the IED-s	Verify that ExtRefAddr is available on the SourceRef	☐☑☒	NA
B4		Verify that engineering rights for the subscribing IED are set to “data flow” and that the “owner” attribute is set	☑	
C	Configure publication of GOOSE			
C1	SCT-p imports the SED file (SED-s)	SCT is able to import SED file	☐☑☒	
C2	SCT-p configures a GOOSE message based on the SourceRef found including subscribing IED		☐☑☒	SourceRef can not be loaded; configuration was done manually
C3	SCT-p optionally configures the ExtRef in IED-s		☐☑☒	NA
C4	SCT-p adds IED-s to project (IED element and Access point attached to own subnet without IP addresses and only with GSE subscribed by an IED in that project)		☑	

C5	SCT-p exports the SED file (SED-p) including the IED-p with engineering rights “fixed” and the updated ExtRef in IED-s (keep engineering rights “data flow” for IED-s)		☑	
C6	SCT-p exports the SCD file (SCD-p)		☑	
C7	SCT-p sets engineering rights for IED-s to fixed in its project		☑	
D	SED file inspection (SED-p)			
D0	File validation against namespace		☑	Version/Revision/Release of 6-100 namespace not in file
D1	Verify step C5	Verify the presence of IED-p and IED-s with “owner” attributes and engineering rights	☑	
D2	Verify step C2	Verify the configuration of the GOOSE control block and dataset in IED-p	☑	
D3		Verify the configuration of the communication parameters for the GOOSE message, including subscribing IED	☑	

D4	Verify step C3	Verify the ExtRef in IED-s and if IED-s supports later binding the ExtRef is in line with the ExtRefAddr of the SourceRef	☐☑☒	NA
D5	Verify step C4	Verify the presence of the access point related to IED-p with the GSE element for the published GOOSE	☑	
D6	Verify step C5	Verify that engineering rights for the publishing IED are set to “fixed”	☑	
E	SCD file inspection (SCD-p)			
E0	File validation against namespace		☑	Version/Revision/Release of 6-100 namespace not in file
E1	Verify step C2	Verify the configuration of the GOOSE control block and dataset in IED-p	☑	
E2		Verify the configuration of the communication parameters for the GOOSE message, including subscribing IED	☑	
E3	Verify step C4	Verify that IED-s is present in the project	☑	

E4		Verify the presence of the access point related to IED-s on the own subnet with no communication parameters	☑	
E5	Verify step C3	Verify the ExtRef in IED-s	☐☑☒	NA
E6	Verify step C6	Verify that engineering rights for the subscribing IED are set to “fix”	☑	
F	Finalize configuration subscriber			
F1	SCT-s imports SED-p	SCT is able to import SED file	☑	
F2	Add IED-p to project (IED element and Access point attached to own subnet with GSE element for subscribed message; optionally description of GSE element set to “received by tunnel”)		☑	
F3	If not yet done by SCT-p, configure the ExtRef; otherwise update it from SED-p		☒	Tool does not support creation of ExtRef without later binding
F4	Optionally configure LGOS		☐☑☒	NA
F5	Export SCD file (SCD-s)		☑	
G	SCD file inspection (SCD-s)			
G0	File validation against namespace		☒	Reference to schema file and namespace incorrect LNInst element with empty values

G1	Verify step F2	Verify that IED-p is present in the project	☑	
G2		Verify that the access point for IED-p is present in the project on the own subnet with only the GSE element of the subscribed GOOSE message, no IP address information and optionally the description set to “received by tunnel”	☑	
G3	Verify step F3	Verify the ExtRef is filled in correctly	☐	NA – tool was not able to configure
G4	Verify step F4	Check configuration of GOOSE message	☑	

14.12Test 22.1b – Configuration SS B subscribing from SS A

Date of Test	14.10.2021	Substation	SS-B
SCD-s	Helinks	SCD-p	Ingeteam
Expected Namespace		61850-6	61850-6-100
		2007B4	2019A6

	Test Step	Verification	Ok ☐☑☒	Remarks / Observations
A	File export / import			

A1	SCT-s exports the SED file (SED-s)	SCT is able to export SED file	☑	
B	SED file inspection (SED-s)			
B0	File validation against namespace		☑	Version/Revision/Release of 6-100 namespace not present
B1		Verify in the file the presence of the subscribing IED (IED-s)	☑	
B2		Verify in the file the presence of the subscribing function with filled in SourceRef and the publishing function	☑	
B3	If later binding is supported by the IED-s	Verify that ExtRefAddr is available on the SourceRef	☐	NA as for SS-SS exchange we are not using later binding
B4		Verify that engineering rights for the subscribing IED are set to “data flow” and that the “owner” attribute is set	☑	
C	Configure publication of GOOSE			
C1	SCT-p imports the SED file (SED-s)	SCT is able to import SED file	☑	
C2	SCT-p configures a GOOSE message based on the SourceRef found including subscribing IED		☑	
C3	SCT-p optionally configures the ExtRef in IED-s		☐	NA as we are not doing later binding

C4	SCT-p adds IED-s to project (IED element and Access point attached to own subnet without IP addresses and only with GSE subscribed by an IED in that project)		☑	
C5	SCT-p exports the SED file (SED-p) including the IED-p with engineering rights “fixed” and the updated ExtRef in IED-s (keep engineering rights “data flow” for IED-s)		☑	
C6	SCT-p exports the SCD file (SCD-p)		☑	
C7	SCT-p sets engineering rights for IED-s to fixed in its project		☒	was set to data flow
D	SED file inspection (SED-p)			
D0	File validation against namespace		☒	FCDA contained Errors Attribute engRight was misspelled
D1	Verify step C5	Verify the presence of IED-p and IED-s with “owner” attributes and engineering rights	☑	Engineering rights were inverted
D2	Verify step C2	Verify the configuration of the GOOSE control block and dataset in IED-p	☑	

D3		Verify the configuration of the communication parameters for the GOOSE message, including subscribing IED	☑	
D4	Verify step C3	Verify the ExtRef in IED-s and if IED-s supports later binding the ExtRef is in line with the ExtRefAddr of the SourceRef	☐☑☒	NA
D5	Verify step C4	Verify the presence of the access point related to IED-p with the GSE element for the published GOOSE	☑	
D6	Verify step C5	Verify that engineering rights for the publishing IED are set to "fixed"	☒	
E	SCD file inspection (SCD-p)			
E0	File validation against namespace		☒	Reference to schema file and namespace incorrect Same errors as with SED
E1	Verify step C2	Verify the configuration of the GOOSE control block and dataset in IED-p	☑	
E2		Verify the configuration of the communication parameters for the GOOSE message, including subscribing IED	☑	

E3	Verify step C4	Verify that IED-s is present in the project	☑	
E4		Verify the presence of the access point related to IED-s on the own subnet with no communication parameters	☒	Was on an own subnet
E5	Verify step C3	Verify the ExtRef in IED-s	☐☑☒	NA
E6	Verify step C6	Verify that engineering rights for the subscribing IED are set to “fix”	☒	Was set to data flow
F	Finalize configuration subscriber			
F1	SCT-s imports SED-p	SCT is able to import SED file	☑	
F2	Add IED-p to project (IED element and Access point attached to own subnet with GSE element for subscribed message; optionally description of GSE element set to “received by tunnel”)		☑	
F3	If not yet done by SCT-p, configure the ExtRef; otherwise update it from SED-p		☒	Needs to be done manually
F4	Optionally configure LGOS		☐☑☒	NA
F5	Export SCD file (SCD-s)		☑	
G	SCD file inspection (SCD-s)			
G0	File validation against namespace		☑	Version/Revision/Release of 6-100 not present

G1	Verify step F2	Verify that IED-p is present in the project	☑	
G2		Verify that the access point for IED-p is present in the project on the own subnet with only the GSE element of the subscribed GOOSE message, no IP address information and optionally the description set to “received by tunnel”	☑	
G3	Verify step F3	Verify the ExtRef is filled in correctly	☑	
G4	Verify step F4	Check configuration of LGOS	☐☑☒	NA

15 Annex N – Structured text program for battery controller LNs

15.1 LN DWFL

```
FUNCTION_BLOCK DWFL
// This implementation is doing load following (not generation following)
// That means, the threshold is a negative value (load), also the input value will be negative
when the power flow is towards the line

VAR_INPUT
// Measurement to follow
InW : REAL;
// Settings
FolWThr : REAL;
END_VAR

VAR_OUTPUT
FolWRef : REAL;       //Reflects the input value of the load to follow
ReqW : REAL;       //Active power requested by the function
END_VAR

FolWRef := InW;
IF InW < FolWThr THEN
   ReqW := -(InW - FolWThr);
ELSE
   ReqW := 0;
END_IF;

END_FUNCTION_BLOCK
```

15.2 LN DWGC

```
FUNCTION_BLOCK DWGC

VAR_INPUT
// Requested active power
InW : REAL;
END_VAR

VAR_OUTPUT
WSpt : REAL;    // mxVal reflecting the active power setpoint applied
ReqW : REAL;    // active power requested by the function
END_VAR
```

```
WSpt := InW;
ReqW := WSpt;

END_FUNCTION_BLOCK
```

15.3 LN DTCD

```
FUNCTION_BLOCK DTCD

VAR_INPUT
// Input
Soc : REAL;
AvlChaW : REAL;
// Settings
SocUseTgtPct : REAL;
DateTgt : TIME;
END_VAR

VAR_OUTPUT
ReqW : REAL;
END_VAR

IF Soc < SocUseTgtPct
THEN
   ReqW := AvlChaW;
ELSE
   ReqW:= 0;
END_IF;
END_FUNCTION_BLOCK
```

15.4 LN DPMC

```
FUNCTION_BLOCK DPMC

VAR_INPUT
// Requested active power
InW1 : REAL;
InW2 : REAL;
InW3 : REAL;
InW4 : REAL;
// SOC of battery
Soc : REAL;
END_VAR
```

```
VAR_OUTPUT
WSpt1 : REAL;     // active power requested on InW1 (from DWFL)
WSpt2 : REAL;    // active power requested on InW2 (from DWGC)
WSpt3 : REAL;    // active power requested on InW3 (from DTCD)
WSpt4 : REAL;    // active power requested on InW4 (from LineDWFL)
ReqW1 : REAL;    // active power requested to be consumed (DLOD)
ReqW2 : REAL;     // active power requested to be gemerated (DGEN)
ReqW3 : REAL;    // active power requested to be consumed by battery in other SS
END_VAR

VAR
NotFirstRun : BOOL;
Leading : BOOL;          // This side sends request to other side
TargetBattery : REAL;    // target for the battery - positive means generate
SetBattery: REAL;        // setpoint for battery - considering ramping
END_VAR

IF NotFirstRun = FALSE THEN
   SetBattery := 0;
   NotFirstRun := TRUE;
END_IF;

WSpt1 := InW1;
WSpt2 := InW2;
WSpt3 := InW3;
WSpt4 := InW4;

// Determinate the request to the battery (positive = generating)
IF InW1 > 0 THEN
// Load following requires generation; active power control will be ignored
   Leading := TRUE;
   TargetBattery := InW1;
ELSIF InW2 < 0 THEN
// Active power control from other side; negative is request to consume - balancing load
   Leading := FALSE;
   TargetBattery := InW2;
ELSIF InW2 > 0 THEN
// Active power control from other side; positive is request to generate - balancing back to
50%
   Leading := FALSE;
   TargetBattery := InW2;
ELSIF InW3 > 0 THEN
// Request to recharge the battery
   Leading := TRUE;
   IF InW4 > 0 THEN
      // recharge created overload situation
```

```
        TargetBattery := SetBattery + InW4;
    ELSE
    //    TargetBattery := - (InW3 - InW4);
        TargetBattery := -InW3;
    END_IF;
ELSE
// no request
    Leading := FALSE;
    TargetBattery := 0;
END_IF;

// Creating the Battery setpoint with ramping
IF Leading THEN
    IF TargetBattery - SetBattery > 1 THEN
        SetBattery := SetBattery+0.5;
    ELSIF TargetBattery - SetBattery < -1 THEN
        SetBattery := SetBattery - 0.5;
    ELSE
        SetBattery := TargetBattery;
    END_IF;
ELSE
    SetBattery := TargetBattery;
END_IF;

// setting the outputs and watch SoC
IF SetBattery > 0 THEN
    IF Soc > 5 THEN
        ReqW2 := SetBattery;
    ELSE
        ReqW2 := 0;
    END_IF;
    ReqW1 := 0;
ELSE
    IF Soc < 95 THEN
        ReqW1 := - SetBattery;
    ELSE
        ReqW1 := 0;
    END_IF;
    ReqW2 := 0;
END_IF;
IF Leading THEN
    ReqW3 := - SetBattery;
ELSE
    ReqW3 := 0;
END_IF;
```

```
END_FUNCTION_BLOCK
```

16 Annex O – Test results Battery application

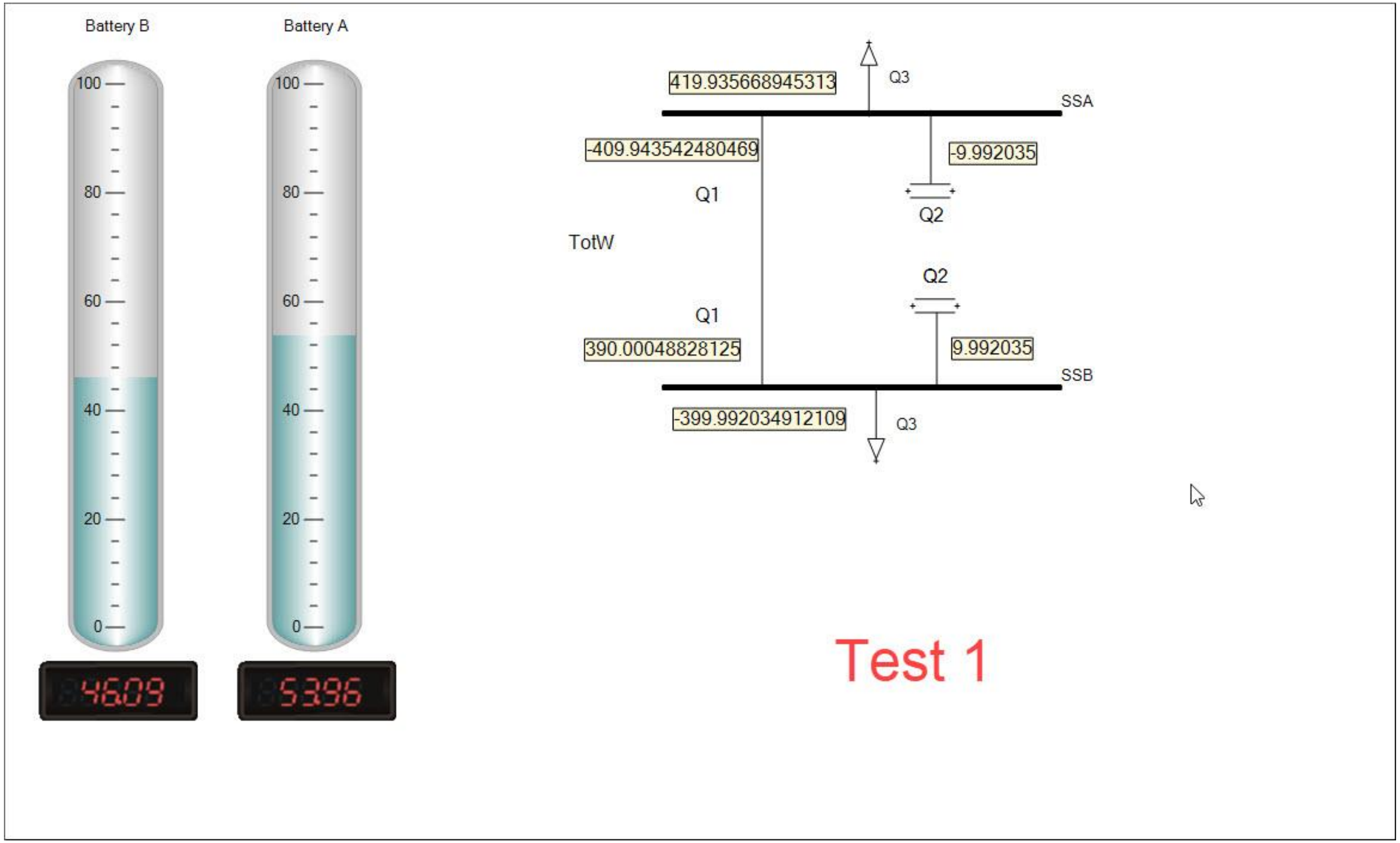
16.1 Introduction

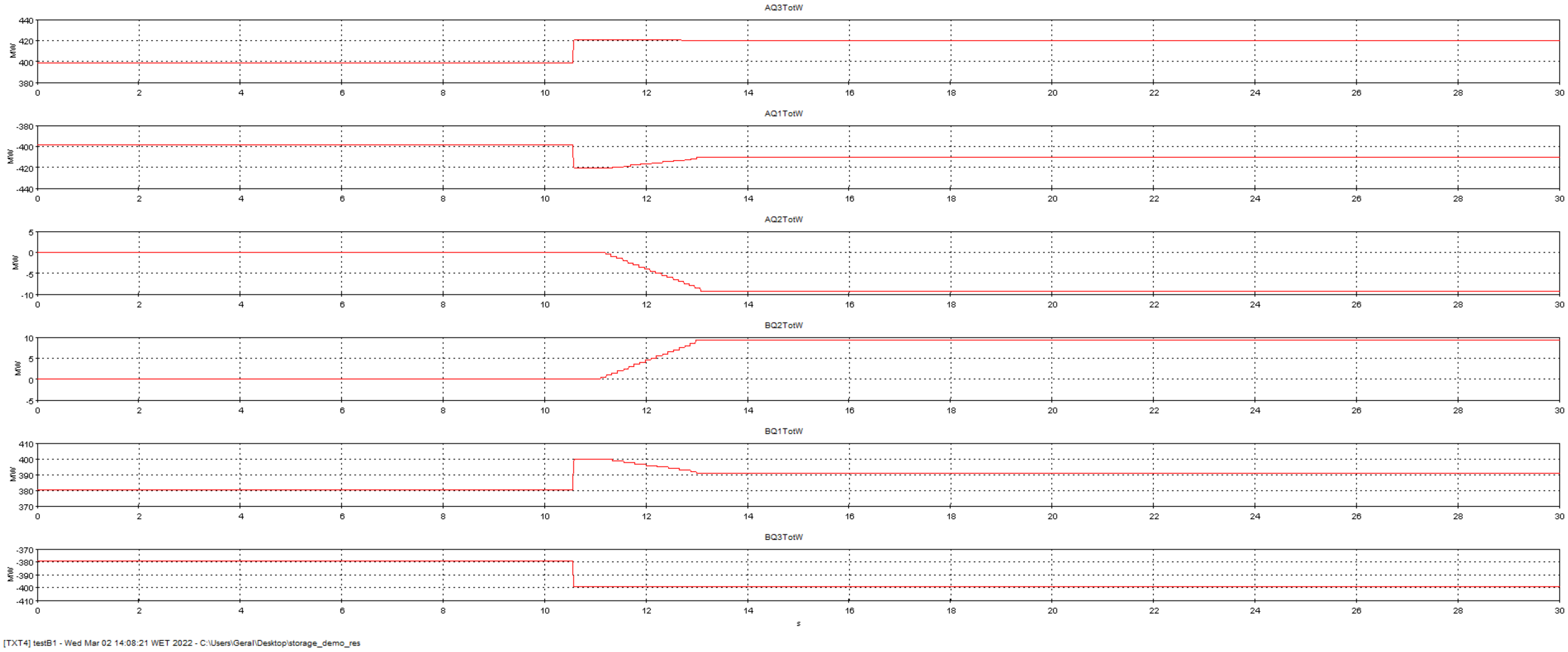
Below are the results for the tests according to Table 7. For each of the test we show

- A screenshot of the values captured in DTM once the situation was stable
- Curves showing the values changing. In the curves, the impact from the ramping of the battery output can be observed

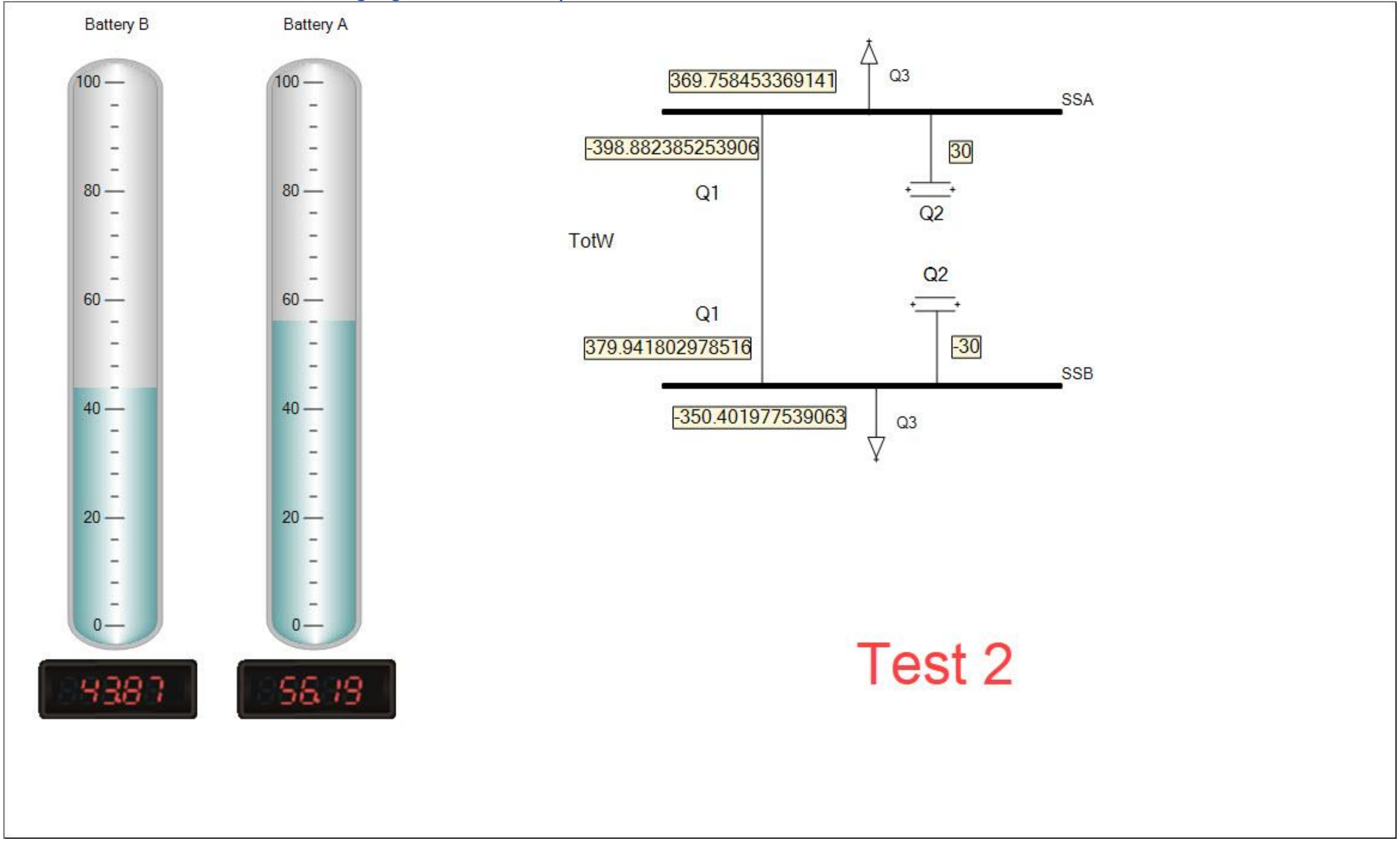
16.2 Tests with the load side in substation B

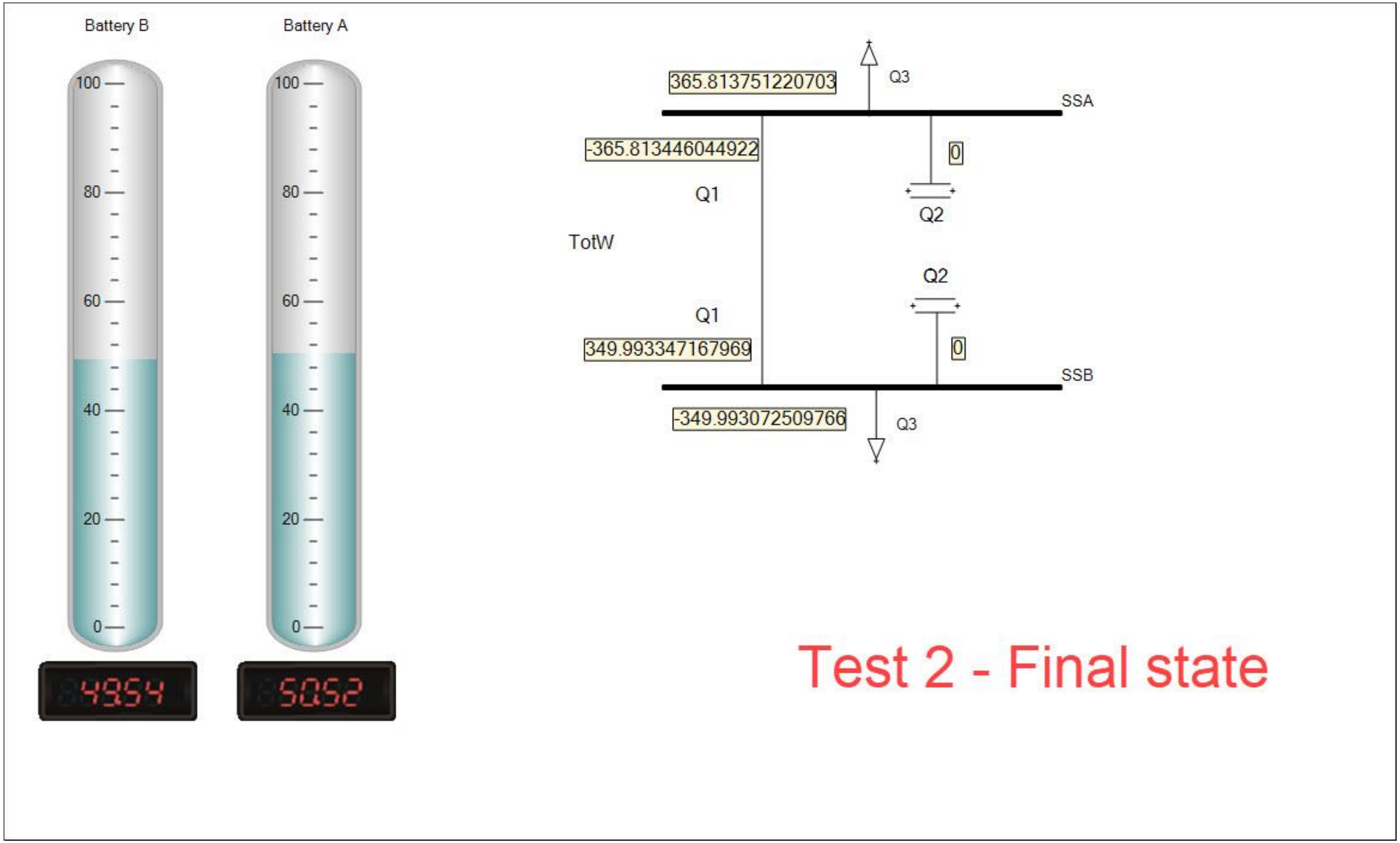
16.2.1 Situation 1: overload

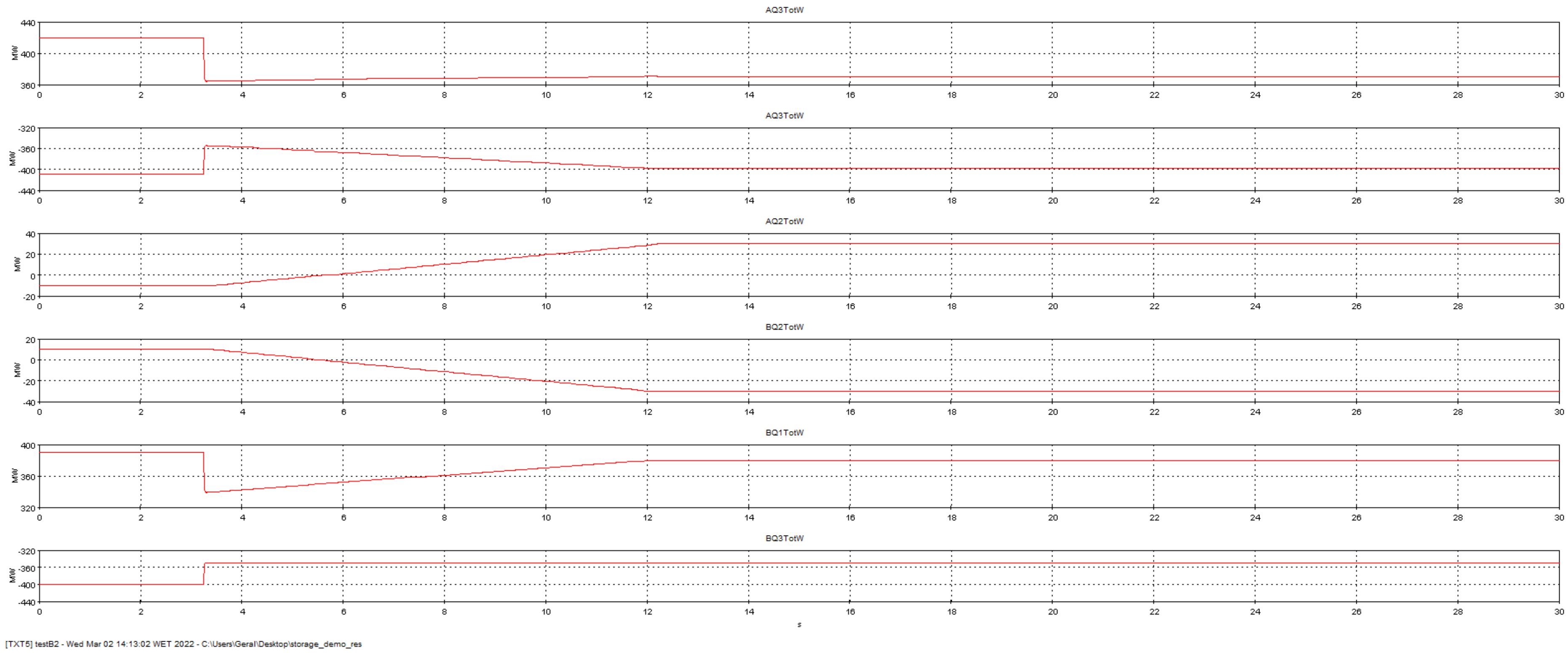



AQ3TotW
AQ1TotW
AQ2TotW
BQ2TotW
BQ1TotW
BQ3TotW
MW
s
[TXT4] testB1 - Wed Mar 02 14:08:21 WET 2022 - C:\Users\Geral\Desktop\storage_demo_res

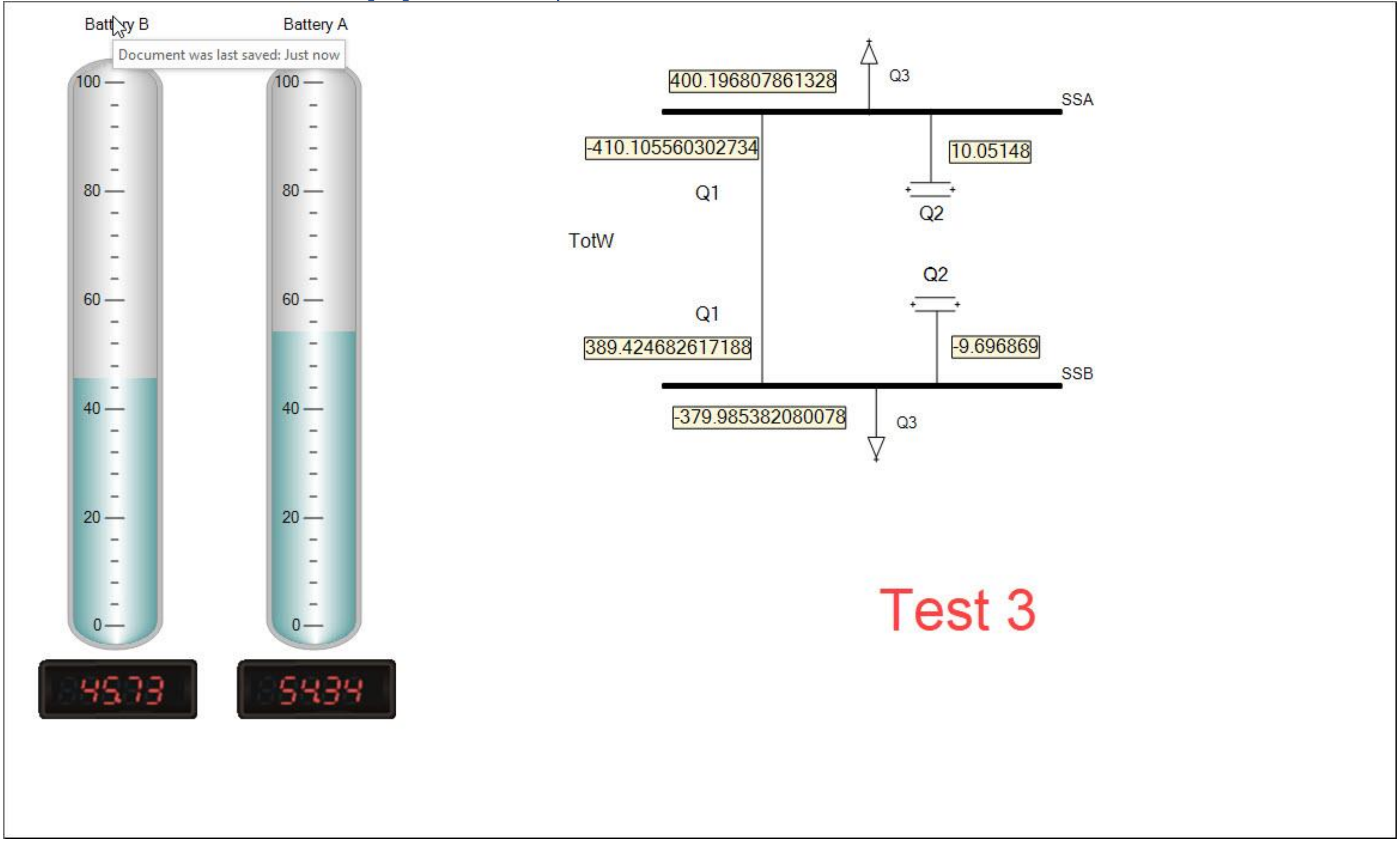
16.2.2 Situation 2: recharging at maximum power

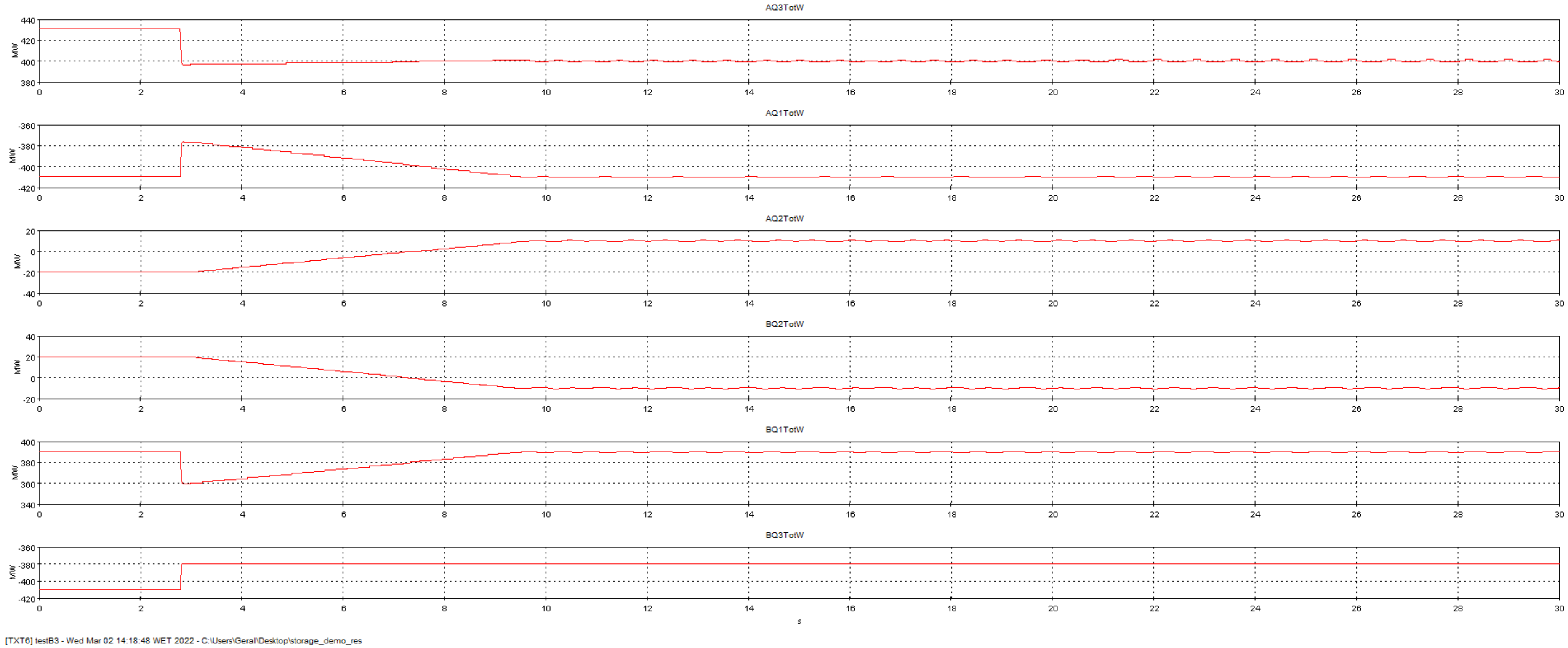



Battery B
Battery A
100
80
60
40
20
0
4954
5052
365.813751220703
Q3
SSA
-365.813446044922
0
Q1
Q2
TotW
Q2
Q1
349.993347167969
0
SSB
-349.993072509766
Q3
Test 2 - Final state


AQ3TotW
MW
AQ3TotW
MW
AQ2TotW
MW
BQ2TotW
MW
BQ1TotW
MW
BQ3TotW
MW
s
[TXT5] testB2 - Wed Mar 02 14:13:02 WET 2022 - C:\Users\Geral\Desktop\storage_demo_res

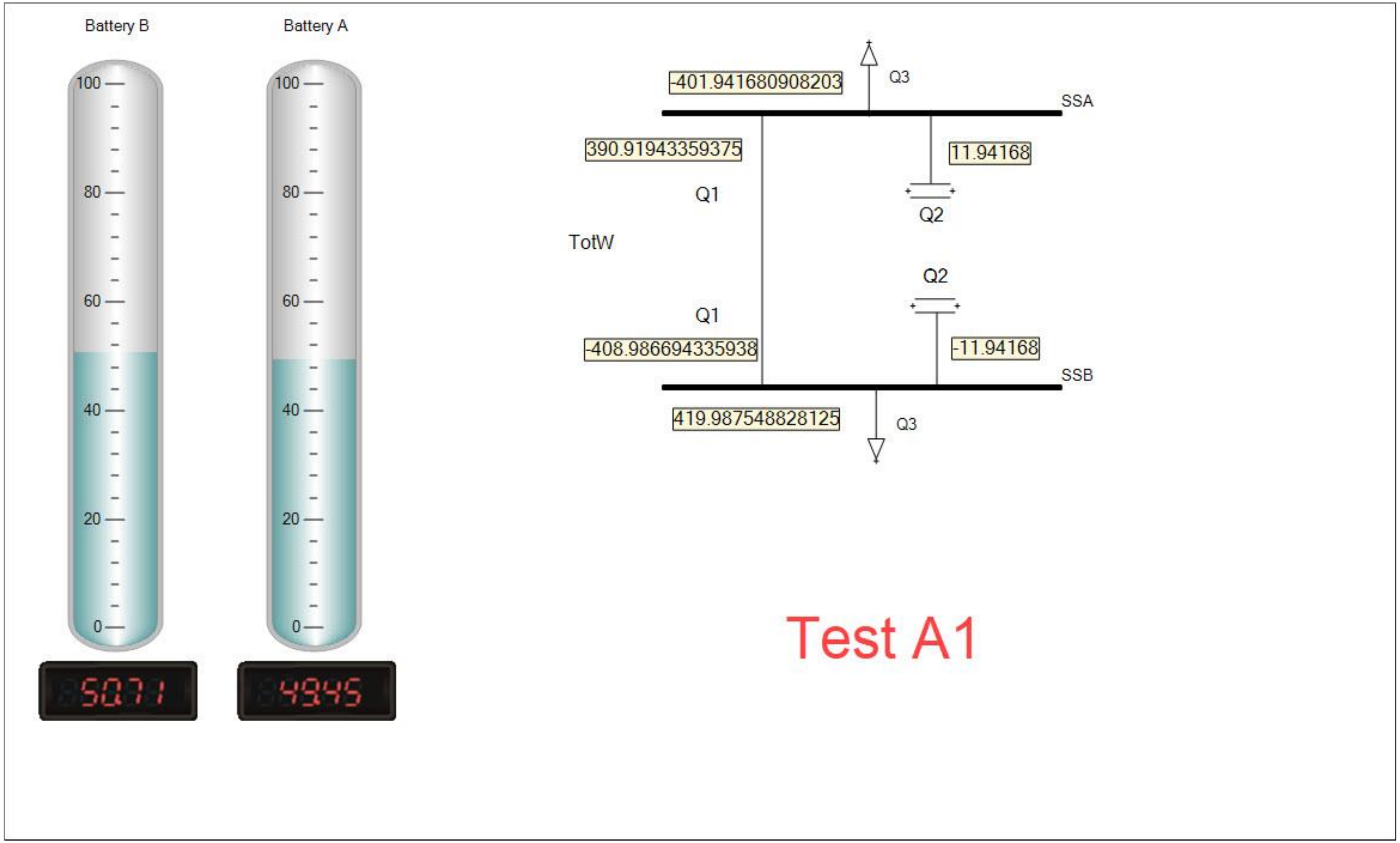
16.2.3 Situation 3: Recharging with limited power

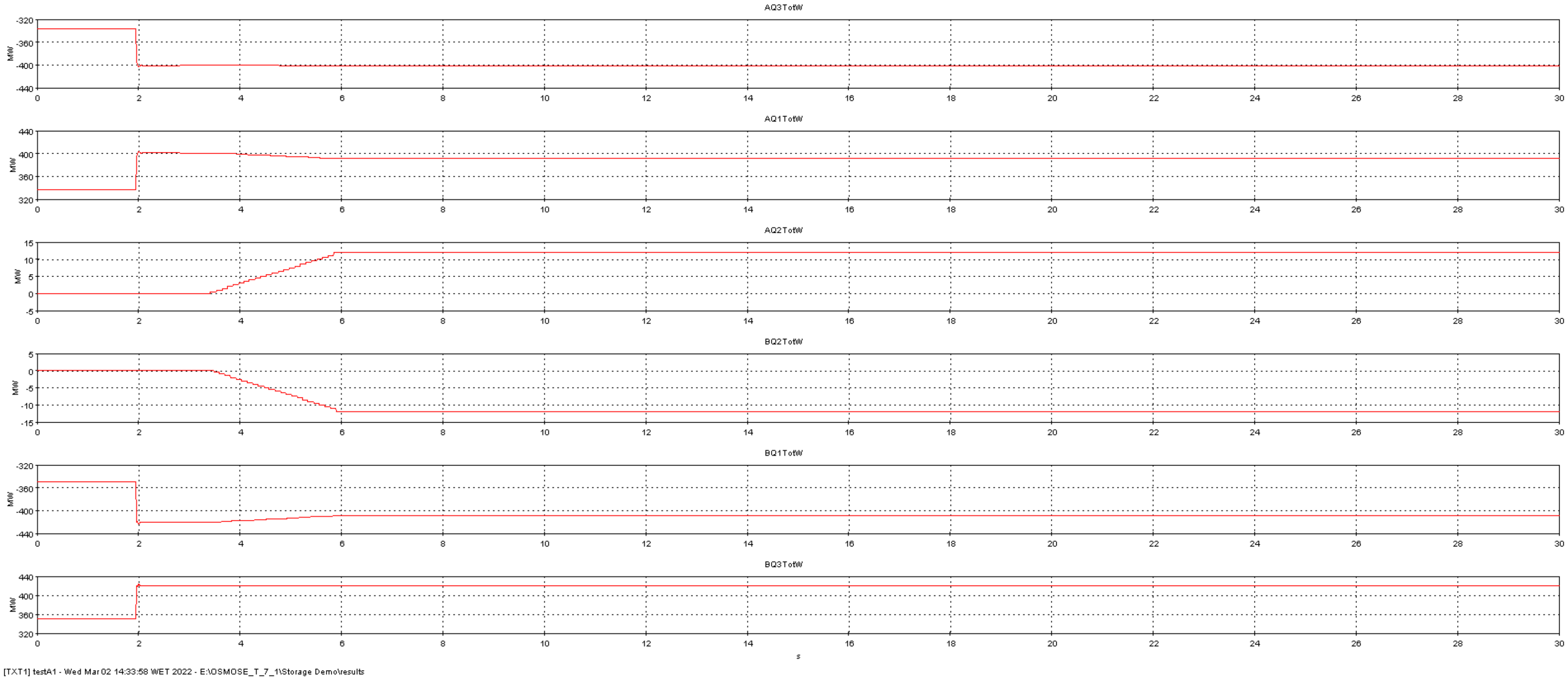



AQ3TotW
MW
AQ1TotW
MW
AQ2TotW
MW
BQ2TotW
MW
BQ1TotW
MW
BQ3TotW
MW
s
[TXT6] testB3 - Wed Mar 02 14:18:48 WET 2022 - C:\Users\Geral\Desktop\storage_demo_res

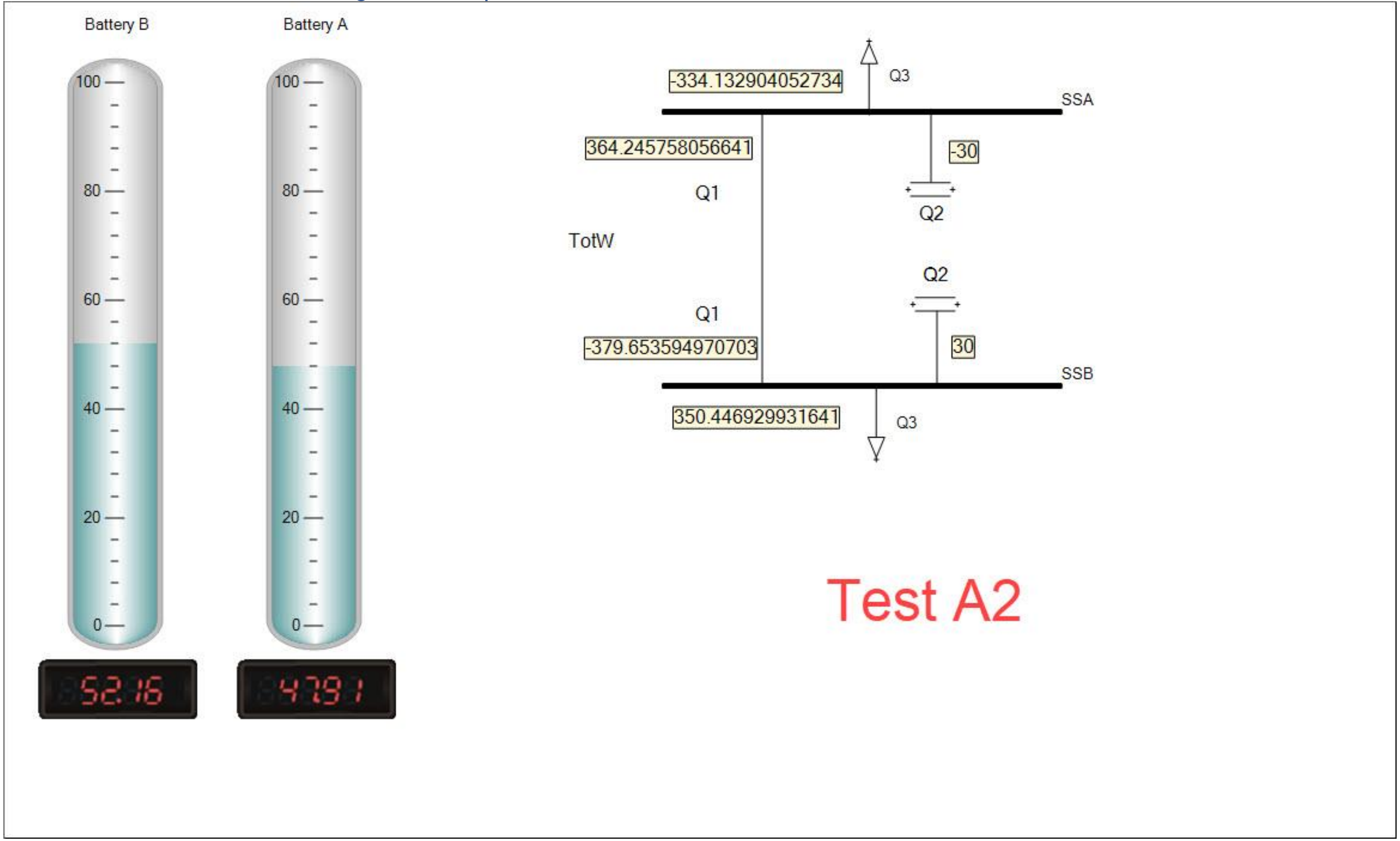
16.3 Tests with load side in Substation A

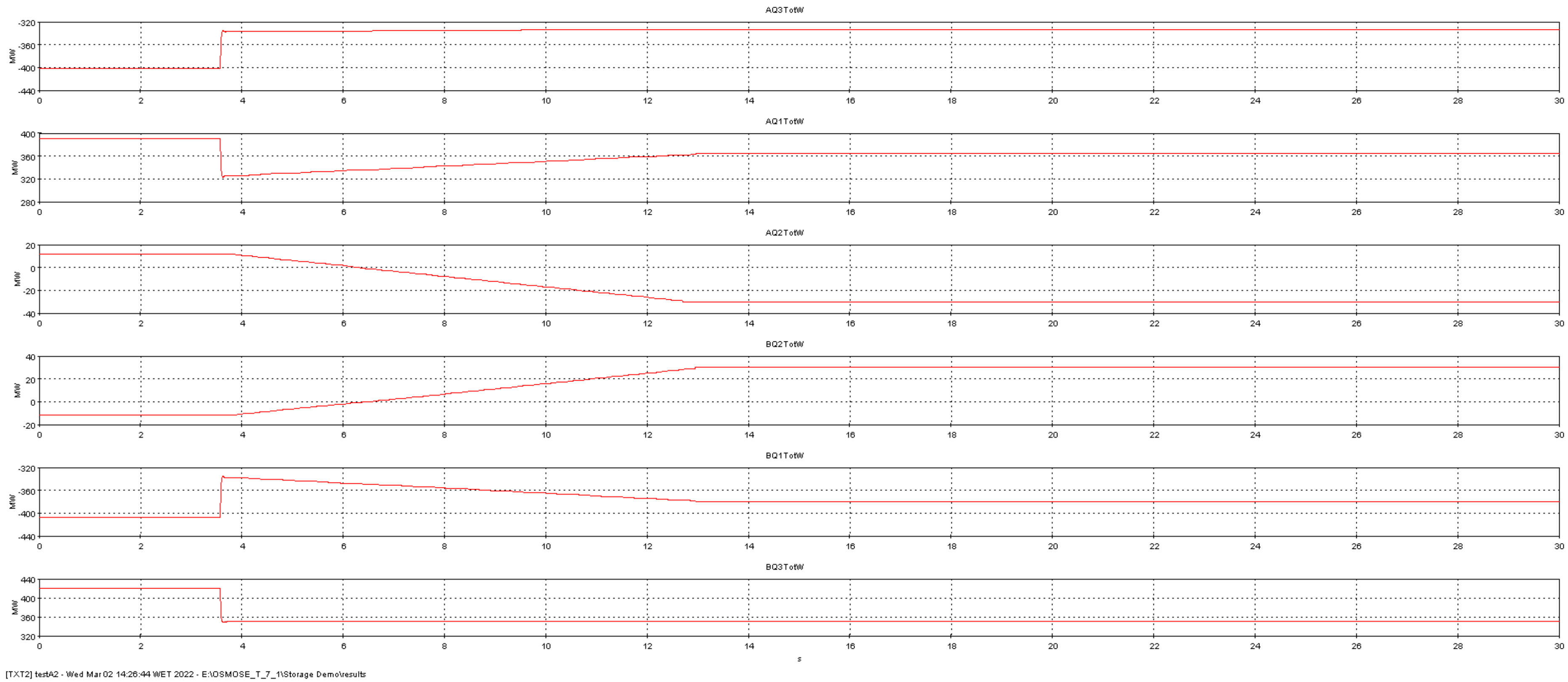
16.3.1 Situation 1: overload



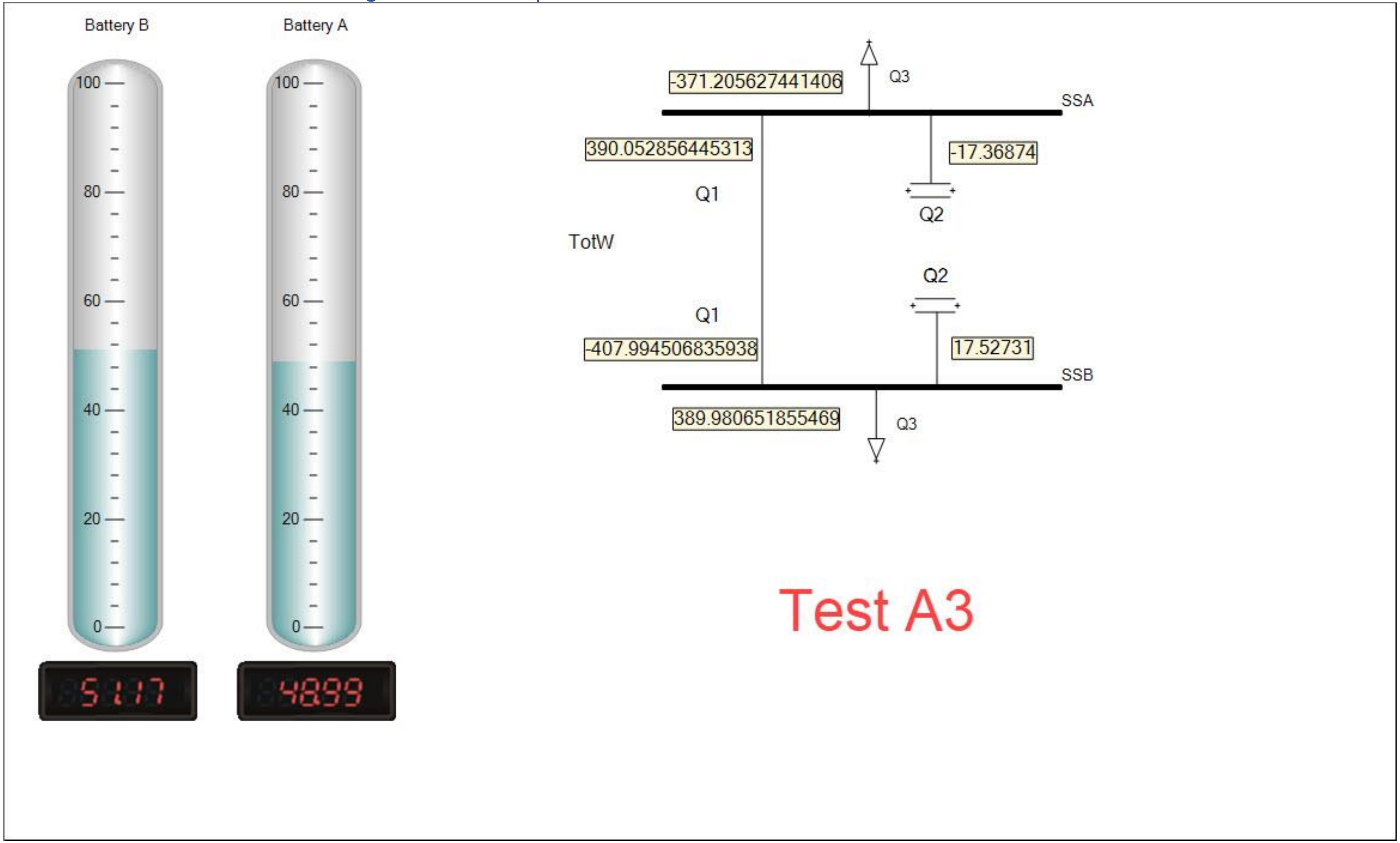

AQ3TotW
AQ1TotW
AQ2TotW
BQ2TotW
BQ1TotW
BQ3TotW
MW
s
[TXT1] testA1 - Wed Mar 02 14:33:58 WET 2022 - E:\OSMOSE_T_7_1\Storage Demo\results

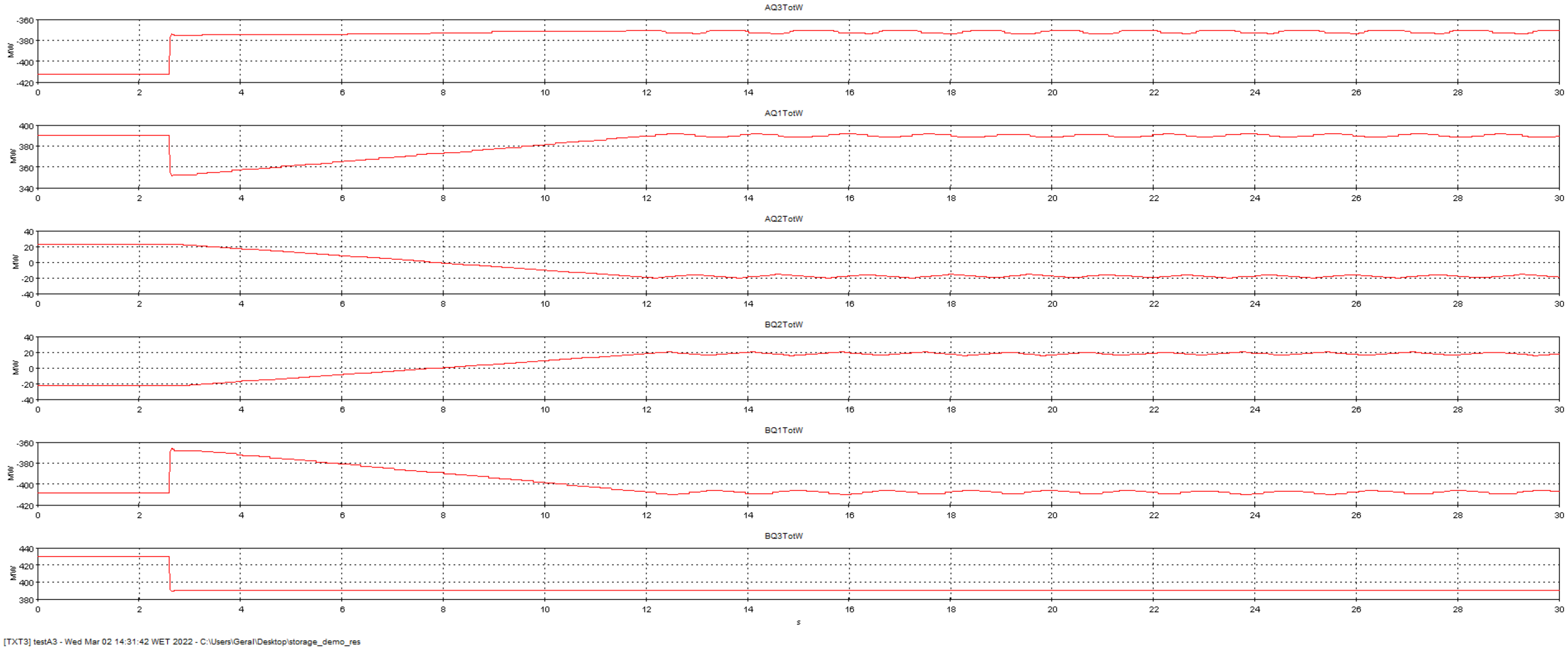
16.3.2 Situation 2: recharge with full power




AQ3TotW
AQ1TotW
AQ2TotW
BQ2TotW
BQ1TotW
BQ3TotW
MW
s
[TXT2] testA2 - Wed Mar 02 14:26:44 WET 2022 - E:\OSMOSE_T_7_1\Storage Demo\results

16.3.3 Situation 3: Recharge with limited power




AQ3TotW
AQ1TotW
AQ2TotW
BQ2TotW
BQ1TotW
BQ3TotW
MW
s
[TXT3] testA3 - Wed Mar 02 14:31:42 WET 2022 - C:\Users\Geral\Desktop\storage_demo_res