COMILLAS
UNIVERSIDAD PONTIFICIA
ICAI

MÁSTER UNIVERSITARIO EN INGENIERÍA INDUSTRIAL

TRABAJO FIN DE MÁSTER

AUTOMATION OF SCADA SYSTEM TESTS USING HARDWARE IN THE LOOP

Autor: Daniel Campo Pagán

Director: Estanislao Brotons Llor

Co-Director: Simisa Simiç

Madrid

Julio de 2025

Declaro, bajo mi responsabilidad, que el Proyecto presentado con el título

AUTOMATION OF SCADA SYSTEM TESTS USING HARDWARE IN THE LOOP

en la ETS de Ingeniería - ICAI de la Universidad Pontificia Comillas en el curso académico 2024/25 es de mi autoría, original e inédito y no ha sido presentado con anterioridad a otros efectos.

El Proyecto no es plagio de otro, ni total ni parcialmente y la información que ha sido tomada de otros documentos está debidamente referenciada.

Fdo.: Daniel Campo Pagán   Fecha: 23/08/2025

Autorizada la entrega del proyecto

EL DIRECTOR DEL PROYECTO

Estanislao Brotons Llor

Fdo.: Estanislao Brotons Llor   Fecha: 25/08/2025

AUTOMATION OF SCADA SYSTEM TESTS USING HARDWARE IN THE LOOP

Autor: Campo Pagán, Daniel.
Director: Brotons Llor, Estanislao / Simiç, Simisa
Entidad Colaboradora: GEDLux Sistemas de Control S.L. / Typhoon HIL

RESUMEN DEL PROYECTO

Palabras clave: SCADA, HIL, Automatización de Pruebas, Informe, FAT

1. Introducción

Este proyecto surge de la reevaluación de las pruebas de aceptación en fábrica (FAT) para sistemas SCADA y de la idea de que estos procedimientos pueden acelerarse y hacerse más eficientes, especialmente mediante la automatización. Las FAT han sido históricamente fundamentales para la implementación de sistemas de control, ya que cualquier fallo no detectado puede provocar tiempos de inactividad, riesgos de seguridad o interrupciones operativas desencadenando eventos de mayor calibre. En el contexto de los sistemas eléctricos, esto es especialmente crítico: cuando se producen fallos debido a fenómenos naturales, como rayos, o a inestabilidades del sistema en la tensión o la frecuencia, los interruptores deben funcionar sin excepción. Por lo tanto, las FAT garantizan el cumplimiento de las especificaciones del cliente y validan la correcta ejecución de las operaciones preprogramadas en escenarios definidos. Un diseño robusto de las FAT garantiza una puesta en marcha rápida y minimiza los errores durante la instalación.

No obstante, las prácticas actuales de FAT se caracterizan por rutinas repetitivas y mecánicas que pueden prolongarse durante varias semanas. Estos procesos tan largos aumentan el riesgo de error humano, ya que la fatiga de los ingenieros puede comprometer la fiabilidad de los resultados al producir falsos positivos o negativos. Al mismo tiempo, este carácter repetitivo también destaca la FAT como un fuerte candidato para la automatización mediante la ejecución de scripts o programas. Por lo tanto, se busca una solución más económica y adaptable, en la que los resultados se obtengan más rápidamente, con mayor fiabilidad y mejor escalabilidad.

Esta es la función que cumple la simulación Hardware-in-the-Loop (HIL). Al conectar los controladores físicos con entornos simulados en tiempo real, HIL permite simular correctamente el comportamiento del sistema eléctrico sin necesidad de prototipos físicos. Se proporciona un marco riguroso pero flexible para automatizar las pruebas FAT de

SCADA, combinando las ventajas de la eficiencia, la repetibilidad y la reducción del tiempo de puesta en marcha, al tiempo que garantiza la solidez de las pruebas en entornos de infraestructura crítica.

2. Metodología

La metodología del proyecto se estructuró en torno a la cadena de herramientas Typhoon HIL, que integra el modelado, la validación manual y las pruebas automatizadas. Tres componentes principales fueron clave para este flujo de trabajo: el Schematic Editor, utilizado para modelar los modelos eléctricos y de comunicación del IED; HIL SCADA, empleado para pruebas manuales y semiautomatizadas; y el IDE TyphoonTest, que permitió la creación y ejecución de scripts de prueba basados en Python (Figura 1).

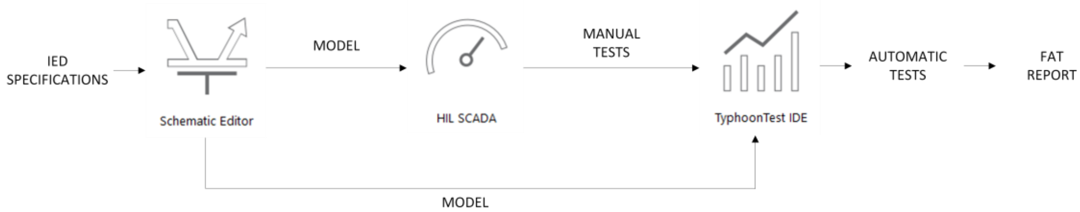


Figura 1. Esquema del Desarrollo del Proyecto

El proyecto base cuenta con ocho bahías del IED de ABB REX615. Estas están conectadas a través de GOOSE y Sampled Values (SV) según las normas IEC 61850 a un SSC600 utilizando un protocolo de redundancia paralela (PRP) (Figura 2).

La implementación se dividió en dos fases principales. La primera fase se centró en la construcción del modelo REX615 dentro del editor de esquemas Typhoon HIL. Se requería que el modelo fuera totalmente funcional tanto en el suministro de señales físicas como en la comunicación a través de GOOSE y SV. El proceso comenzó con la importación de la configuración PCM proporcionada por el cliente para definir la estructura de comunicación. Se desarrolló un primer modelo de la bahía 1 y se validó con el SSC600. Una vez confirmadas sus funcionalidades, se creó un Library Component que permitía la replicación en las bahías restantes. Se añadieron siete instancias adicionales, con pequeños ajustes para tener en cuenta las variaciones específicas de cada bahía en las relaciones de los transformadores de corriente y tensión. La validez del modelo se logró una vez que el SSC600 interpretó los dispositivos REX615 simulados como indistinguibles de sus homólogos físicos.

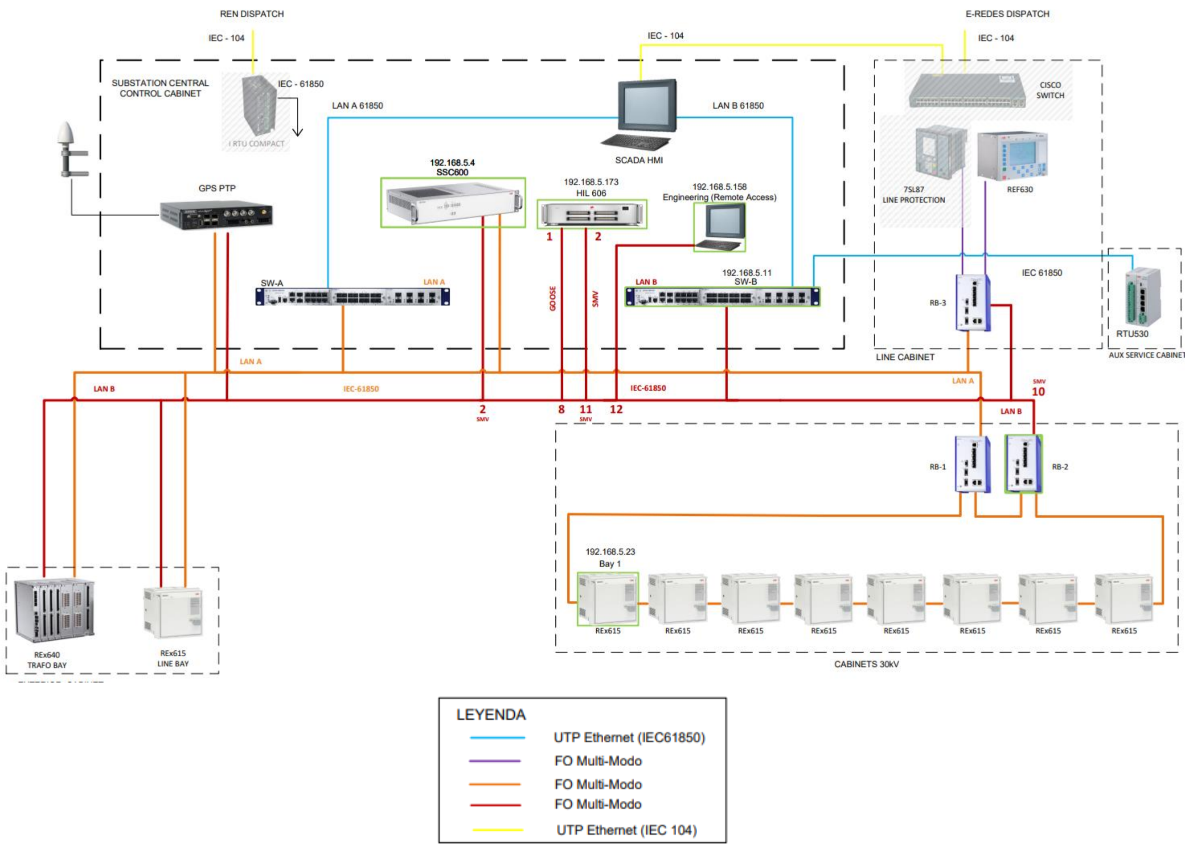


Figura 2. Esquema de Comunicaciones

La segunda fase se centró en las pruebas progresivas y la automatización. Inicialmente, la verificación FAT manual se replicó utilizando paneles SCADA personalizados dentro de HIL SCADA. Estos paneles permitieron a los operadores realizar comandos de posición, observar valores de medición y activar fallos, imitando de cerca la práctica FAT actual. Una vez validado, se desarrollaron macros semiautomatizadas para agilizar las secuencias de pruebas repetitivas, proporcionando un paso intermedio entre la operación manual y la automatización completa.

La etapa final introdujo la automatización completa, integrando Pytest con la API Typhoon HIL para ejecutar rutinas de prueba estandarizadas. Los scripts de Python controlaban el entorno de simulación, generaban condiciones de prueba y registraban los resultados automáticamente. La ejecución comenzó con la validación de una sola bahía y luego se amplió a las ocho bahías. Para mejorar la usabilidad, se debe diseñar un script maestro que coordine las pruebas individuales a nivel de bahía sin necesidad de reconfiguración manual. Este enfoque garantiza la reproducibilidad y minimiza el esfuerzo de ingeniería, proporcionando una solución escalable para campañas FAT automatizadas.

3. Resultados

El marco automatizado logró una reducción del 60 % en el tiempo de ejecución de FAT, lo que redujo el tiempo de pruebas desde aproximadamente 2,5 semanas a 1 semana. Los scripts automatizados validaron las lógicas de protección, las condiciones de disparo y los comportamientos de comunicación en todas las bahías modeladas.

Desde una perspectiva económica, la tasa interna de rendimiento (TIR) se calculó entre el 17 % y el 23 % en escenarios de rango medio, mientras que el valor actual neto (VAN) osciló entre 91.280 € y 130.880 € en un horizonte de 10 años al 5 %. Los resultados resaltan la viabilidad financiera de las FAT automatizadas, con proyecciones que sugieren que se pueden lograr mejoras de eficiencia de hasta el 80 % debido al proceso de aprendizaje y perfeccionamiento.

4. Conclusiones

El proyecto demuestra que la automatización SCADA FAT mediante HIL es técnicamente viable y económicamente atractiva. Este enfoque mejora las prácticas manuales al reducir la duración de las pruebas, garantizar la reproducibilidad y simplificar la depuración, mientras que la generación automática de informes refuerza la trazabilidad y la documentación. Al ofrecer una alternativa flexible y de bajo coste a las pruebas tradicionales basadas en hardware, el marco permite el desarrollo iterativo y la integración continua, lo que lo convierte en una solución atractiva para proyectos de automatización de subestaciones.

De cara al futuro, ampliar el modelado a una gama más amplia de IED e incorporar dispositivos específicos de los fabricantes será clave para permitir entornos de prueba realistas y estandarizados. La integración de controladores virtualizados, como el SSC600, pone de relieve el potencial de las líneas de pruebas totalmente digitales, lo que a su vez permite desacoplar las pruebas de los calendarios de fabricación. Estos avances están en consonancia con la adopción generalizada de gemelos digitales y la puesta en marcha virtual, que pueden acortar los plazos de los proyectos, reducir los riesgos y mejorar la agilidad en la ingeniería de subestaciones.

AUTOMATION OF SCADA SYSTEM TESTS USING HARDWARE IN THE LOOP

Author: Campo Pagán, Daniel.
Supervisor: Brotons Llor, Estanislao / Simiç, Simisa
Collaborating Entity: GEDLux Sistemas de Control S.L. / Typhoon HIL

ABSTRACT

Keywords: SCADA, HIL, Test Automation, Report, FAT

1. Introduction

This project emerges from the reassessment of Factory Acceptance Tests (FAT) for SCADA systems and the idea that these procedures can be accelerated and made more efficient, especially through automation. FAT has historically been central to the deployment of control systems, as any undetected malfunction can result in downtime, safety risks, or operational disruptions with cascading effects. In the context of power systems, this is particularly critical: when faults occur due to natural events such as lightning strikes or due to system instabilities in voltage or frequency, circuit breakers must operate without exception. FAT therefore guarantees compliance with customer specifications and validates the proper execution of pre-programmed operations under defined scenarios. A robust FAT design ensures swift commissioning and minimises errors during installation.

Nonetheless, current FAT practices are characterised by repetitive and mechanical routines that may extend over several weeks. Such lengthy processes increase the risk of human error, as engineer fatigue can compromise the reliability of results by producing false positives or negatives. At the same time, this repetitive character also highlights FAT as a strong candidate for automation through scripted or programmatic execution. A more economical and adaptable solution is thus sought, where results are delivered faster, with higher reliability and improved scalability.

This is the role fulfilled by Hardware-in-the-Loop (HIL) simulation. By interfacing physical controllers with real-time simulated environments, HIL enables the faithful emulation of power system behaviour without the need for physical prototypes. It provides a rigorous yet flexible framework for automating SCADA FAT, combining the advantages of efficiency, repeatability, and reduced commissioning time, while ensuring the robustness of testing in critical infrastructure environments.

2. Methodology

The project methodology was structured around the Typhoon HIL toolchain, which integrates modelling, manual validation, and automated testing. Three main components were key to this workflow: the Schematic Editor, used to model the electrical and communication layers of the IED; HIL SCADA, employed for manual and semi-automated testing; and the TyphoonTest IDE, which enabled Python-based test scripting and execution (Figure 1).

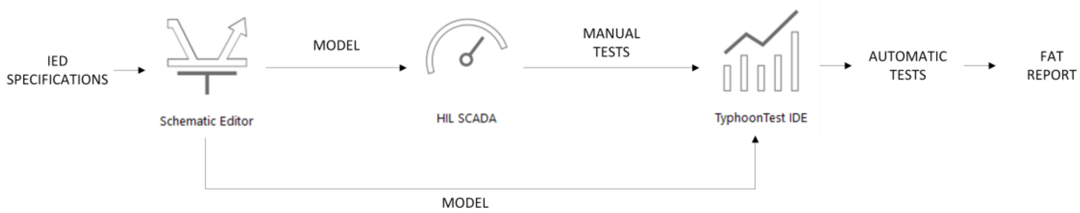


Figure 1. Workflow Schematic

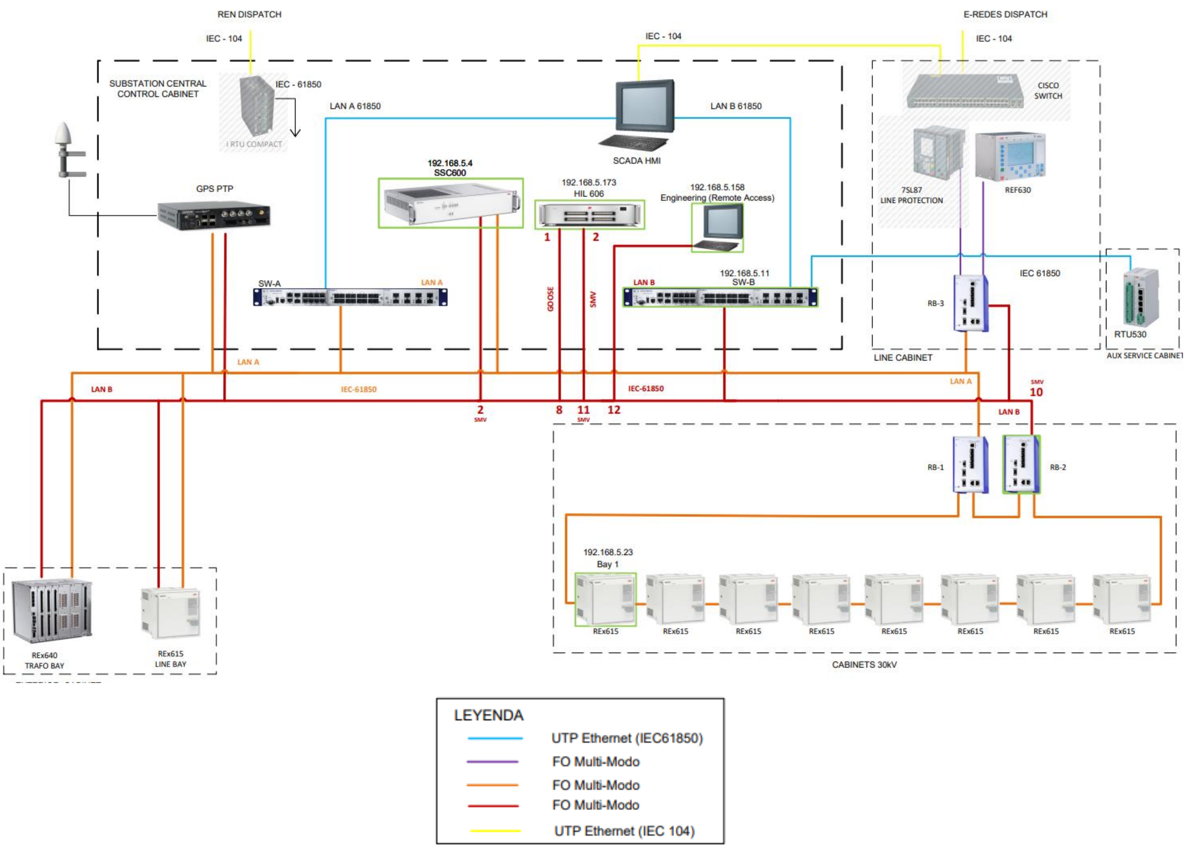


Figure 2. Communications Layout

The base project has eight bays of ABB REX615 IED. These are connected via GOOSE and Sampled Values under IEC 61850 standards to an SSC600 using a Parallel Redundancy Protocol (PRP) (Figure 2).

The implementation was divided into two main phases. The first phase focused on the construction of the REX615 model within the Typhoon HIL Schematic Editor. The model was required to be fully functional in both physical signal provision and communication through GOOSE and Sampled Values. The process began with importing the client-provided PCM configuration to define the communication structure. A first model of Bay 1 was developed and validated against the SSC600. Once its functionalities were confirmed, a library component was created, enabling replication across the remaining bays. Seven additional instances were added, with minor adjustments to account for bay-specific variations in current and voltage transformer ratios. Model validity was achieved once the SSC600 interpreted the simulated REX615 devices as indistinguishable from physical counterparts.

The second phase centred on progressive testing and automation. Initially, manual FAT verification was replicated using customised SCADA panels within HIL SCADA. These panels enabled operators to perform position commands, observe measurement values, and trigger faults, closely imitating current FAT practice. Once validated, semi-automated macros were developed to streamline repetitive test sequences, providing an intermediate step between manual operation and full automation.

The final stage introduced complete automation, integrating Pytest with the Typhoon HIL API to execute standardised test routines. Python scripts controlled the simulation environment, generated test conditions, and logged results automatically. Execution began with single-bay validation and was then expanded to all eight bays. To improve usability, a master script must be designed to coordinate individual bay-level tests without requiring manual reconfiguration. This approach ensures reproducibility and minimised engineering effort, providing a scalable solution for automated FAT campaigns.

3. Results

The automated framework achieved a 60% reduction in FAT execution time, reducing test campaigns from approximately 2.5 weeks to 1 week. Automated scripts validated protection logics, tripping conditions, and communication behaviours across the modelled bays.

From an economic perspective, the Internal Rate of Return (IRR) was calculated at 17%–23% under mid-range scenarios, while the Net Present Value (NPV) ranged between €91,280 and €130,880 over a 10-year horizon at 5%. The results highlight the financial viability of automated FAT, with projections suggesting that efficiency improvements of up to 80% are achievable with further refinement.

4. Conclusions

The project demonstrates that SCADA FAT automation using HIL is both technically viable and economically attractive. The approach improves upon manual practices by reducing testing duration, ensuring reproducibility, and simplifying debugging, while automated reporting strengthens traceability and documentation. By offering a low-cost and flexible alternative to traditional hardware-based testing, the framework supports iterative development and continuous integration, making it a robust solution for substation automation projects.

Looking ahead, extending the modelling to a broader range of IEDs and incorporating manufacturer-specific devices will be key to enabling realistic and standardised testing environments. The integration of virtualised controllers, such as the SSC600, highlights the potential for fully digital test pipelines which in turn enables decoupling testing from manufacturing schedules. These developments align with the wider adoption of digital twins and virtual commissioning, which can shorten project lead times, reduce risks, and enhance agility in substation engineering.

Index

List of Abbreviations .... ***VI***

Chapter 1. Introduction ***7***

Chapter 2. Technology Description ***8***

2.1 Typhoon HIL 8

2.1.1 HIL *8*

2.1.2 Schematic Editor *9*

2.1.3 HIL SCADA *9*

2.1.4 Test IDE *10*

2.1.5 HIL Connect *10*

2.1.6 Overview *11*

2.2 Historical Evolution of Control and Protection in Electrical Substation 11

2.2.1 Electromechanical Protections *11*

2.2.2 Electronic Protections with Communications *11*

2.2.3 IEC 61850 Substations with GOOSE Messages *12*

2.2.4 Full Digital Substations with Intelligent CTs/VTs *12*

2.2.5 Centralized Protection Systems in Substation Automation *13*

2.3 Current Electrical Substations 13

2.3.1 FAT *13*

2.3.2 ABB / HITACHI Products *14*

2.4 Other Software 16

2.4.1 Wireshark *16*

2.4.2 UA Server Client *16*

2.5 Other Hardware 17

2.5.1 Switch *17*

2.5.2 Injection Equipment *17*

Chapter 3. Literature Review ***20***

3.1 Early adoption of HIL Testing of IEC61850 20

3.2 IED Modelling and HIL approach 20

3.3 Digital substation testing 21

Chapter 4. Scope of Work ***23***

4.1 Justification 23

4.2 Objectives 24

4.3 Methodology 25

4.4 Planning and Economic Estimation 26

Chapter 5. Model Creation ***28***

5.1 Electrical Model 30

5.1.1 Electrical Overview – Root Level *31*

5.1.2 Individual REX615 Bay - Root Level *32*

5.1.3 Electrical Components – Bay Level *33*

5.2 Communication Model 35

5.2.1 Communication Overview – Root Level *35*

5.2.2 Individual REX615 Bay Communication Overview *36*

5.2.3 Communication Model *37*

5.2.4 OPC UA *48*

5.3 Special Cases / Modifications 50

5.3.1 Bay 3 .... *50*

5.4 General Remarks 51

5.4.1 Schematic Editor Functionality *51*

Chapter 6. Additions to Base Model ***54***

6.1 REF630 54

6.1.1 HIL Connect – Root Level *54*

6.2 Virtualised SSC600 55

Chapter 7. SCADA Panels ***56***

7.1 Individual Position Open/Close Commands 57

7.2 OPC UA 58

7.2.1 Receiving From SSC600 (Bits) *60*

7.2.2 Receiving from SSC600 (Analogue) *61*

7.2.3 Sending to SSC600 (Bits) *62*

7.3 Manual generation of signals 63

Chapter 8. Test Automation ***64***

8.1 Trip_Current.py 65

8.1.1 Script *65*

8.1.2 Additional files *68*
8.1.3 Test Passed *70*
8.1.4 Test Failed *71*
8.2 Bit_opc_ua.py 72
8.2.1 Script *72*
8.2.2 Additional Files *74*
8.3 Receive _Trip.py 77
8.3.1 Script *77*

Chapter 9. Project Analysis* *79
9.1 Assumptions 79
9.2 Formulation 81
9.2.1 Variables *81*
9.2.2 Calculated Variables *82*
9.3 Calculations 83
9.3.1 Input Values *83*
9.3.2 Output Variables *83*
9.4 Result Analysis 84

Chapter 10. Conclusions and Future Works* *86

Chapter 11. References* *88

ANNEX I – Sustainable Development Goals* *91

ANNEX II - HIL 606* *92

ANNEX III - REX615* *96

ANNEX IV – SSC600* *102

Figures Index

Figure 1. Workflow Schematic 11
Figure 2. Project Timeline 27
Figure 3. Communications Layout 28
Figure 4. Model Electrical Overview (Root) 31
Figure 5. REX615 Electrical Overview (Root) 32
Figure 6. REX615 Electrical & Communication Overview (Bay) 33
Figure 7. Communication Overview (Root) 35
Figure 8. REX615 Communication Overview (Bay) 36
Figure 9. REX615 Electrical & Communication Overview (Bay) 37
Figure 10. SV Communication Overview 39
Figure 11. Signal Transmission from SSC600 to HIL606 42
Figure 12. GOOSE Subscriber 43
Figure 13. GOOSE Subscriber Parameters Linkage 45
Figure 14. GOOSE Publisher 46
Figure 15. GOOSE Publisher Parameters Linkage 47
Figure 16. OPC UA Server Setup 48
Figure 17. Modifications (Green) made to GOOSE Subscriber for Bay 3 50
Figure 18. HIL Connect Setup (Root) 54
Figure 19. Open/Close Bay Position Panel 57
Figure 20. OPC UA Send/Receive Panel 58
Figure 21. OPC UA Example 1: Receive Digital from SSC600 (Blank) 60
Figure 22. OPC UA Example 2: Receive Digital from SSC600 (Received) 61
Figure 23. OPC UA Example 3: Receive Analogue from SSC600 61
Figure 24. OPC UA Example 4: Send to SSC600 62
Figure 25. Manual Analogue Signal Generation Panel 63
Figure 26. Protection configuration for 3x51 and 50N for Bay 2 68
Figure 27. Passed test for 3x51 and 50N for Bay 2 70
Figure 28. Failed test for 3x51 and 50N for Bay 2 71

Table Index

Table 1. Injection Values for 50N and 51N Protections 18
Table 2. Nominal Voltage, Power and Current for all Bays 30
Table 3. Flip Flop Switch Logic 37
Table 4. Signals to be tested by OPC UA 74
Table 5. Extra Revenue per Year 83
Table 6. IRR at 10 years 84
Table 7. NPV at 10 years with 5% 84

LIST OF ABBREVIATIONS

- CB – Circuit Breaker
- C-HIL – Controller Hardware-in-the-Loop
- ES – Earthing Switch
- FAT – Factory Acceptance Test
- GOOSE – Generic Object Oriented Substation Event
- HIL – Hardware-in-the-Loop
- HMI – Human-Machine Interface
- IED – Intelligent Electronic Device
- IEC – International Electrotechnical Commission
- IDE – Integrated Development Environment
- IRR – Internal Rate of Return
- KVM – Kernel-based Virtual Machine
- NPV – Net Present Value
- OPC – Open Platform Communications
- OPC UA – OPC Unified Architecture
- PCM600 – Protection and Control IED Manager
- PRP – Parallel Redundancy Protocol
- PTP – Precision Time Protocol
- ROI – Return on Investment
- RTDS – Real-Time Digital Simulator
- SCADA – Supervisory Control and Data Acquisition
- SDG – Sustainable Development Goals
- SIL – Software-in-the-Loop
- SSC600 – Smart Substation Control and Protection Device
- SV – Sampled Values
- VM – Virtual Machine
- WHMI – Web Human-Machine Interface

Chapter 1. INTRODUCTION

This project emerges from the revaluation of a standard practice, the Factory Acceptance Tests (FAT) for SCADA systems, and how it could be sped up, made more efficient or automate it to combine both.

Historically, FAT testing in control systems has been a key area for a project to be deployed effectively. A control system undermined by errors can lead to downtimes, safety risks and operational disruptions which can set off chain reactions and propagate through the system. This is especially critical when discussing power systems.

When a fault occurs, due to natural causes, such as lighting strikes, or due to fluctuations in the system's voltage or frequency, the circuit breaker apparatus must operate. No exceptions. Therefore, FAT testing ensures both the compliance with customer specifications as well as pre-programmed operations' functionality and how they execute under specific circumstances. All in all, a good FAT design will ensure a swift commissioning as well as reduce errors in the installation process, with all the benefits these entail.

Nevertheless, FAT testing can result in repetitive and mechanical verification processes which can take up to several weeks at a time. This can lead to individual engineer burnout which in turn can cause defective testing and give either positive tests when negative or vice versa. On another note, this repetitive process can prove useful when it comes to automating the FAT SCADA testing as it can be automated through scripts or programs.

Therefore, a more economical solution is the target where the company can produce faster, more reliable and more adaptable FAT testing results and reports.

This is where the Hardware-in-the-Loop solution comes into play. Hardware-in-the-Loop (HIL) simulation is a technique that enables the development and testing of real-time embedded control systems by interfacing actual control hardware with a simulated environment representing the physical system.

Chapter 2. TECHNOLOGY DESCRIPTION

2.1 TYPHOON HIL

2.1.1 HIL

Hardware-in-the-Loop simulation, sometimes also mentioned as Controller Hardware-in-the-Loop (C-HIL) [1] [2], is characterised by the direct interaction between a real, physical controller and a high-fidelity, real-time digital model of the power stage. The simulation is executed on dedicated HIL devices that operate with sub-microsecond time steps, ensuring accurate emulation of fast-switching dynamics typical of power electronics systems. This setup allows for the assessment of control strategies under a wide range of operating conditions, including edge cases that are impractical or unsafe to reproduce with physical prototypes.

HIL frameworks eliminate the need for hardware prototypes during early development stages, thereby reducing dependency on physical testbeds. This is particularly significant in power electronics, where conventional testing often involves cumbersome and hazardous setups. Through HIL, engineers can replicate fault conditions, conduct corner-case testing, and validate protection mechanisms with full control over the simulation parameters, all while the controller operates exactly as it would in a deployed system.

Furthermore, the digital nature of the simulated plant enables seamless integration into automated testing pipelines. HIL systems can be embedded into software development workflows that support version control, continuous integration, and scripted regression testing. This aligns control system development with agile methodologies and shortens the design-verification loop. Notably, HIL supports model-based design by enabling closed-loop testing of control algorithms against evolving models, facilitating concurrent hardware and software development.

The deterministic and low-latency communication between the controller and the HIL simulator is central to preserving the real-time characteristics of the emulated system. The high bandwidth and precision of the signal interface ensure that the control signals and feedback loops behave identically to those in a physical environment. This realism enables accurate tuning and validation of control parameters before deployment.

In contrast to traditional software-in-the-loop (SIL) or purely offline simulation environments, HIL provides a platform for exhaustive testing with physical control hardware, including programmable logic controllers, digital signal processors, and embedded systems. This not only enhances test coverage but also contributes to improved robustness, reliability, and time-to-market performance in power electronics applications.

2.1.2 SCHEMATIC EDITOR

The Schematic Editor within Typhoon HIL enables graphical model construction through the configuration of predefined component blocks. However, its practical use can have some limitations that directly impact the development process, particularly in terms of simulation performance, model flexibility, and debugging efficiency. Understanding these limitations is essential to recommend appropriate hardware specifications and design practices that ensure both resilient and flexible models.

2.1.3 HIL SCADA

HIL SCADA is the graphical interface tool within the Typhoon HIL ecosystem designed for real-time monitoring, control, and visualisation of signals in Hardware-in-the-Loop simulations. It enables users to create customised dashboards by placing widgets such as switches, indicators, and plots, which are directly linked to model variables and signals. These interfaces allow interaction with both digital and analogue inputs and outputs during simulation, supporting tasks such as signal observation, state toggling, and manual control of test scenarios. HIL SCADA operates synchronously with the real-time simulation, ensuring deterministic behaviour and accurate feedback, which is essential for validating control algorithms and protection logic in power system applications. It supports hierarchical organisation of screens and includes user access management features, facilitating structured and secure test environments.

2.1.4 TEST IDE

TyphoonTest IDE is the integrated development environment within the Typhoon HIL toolchain designed for creating, managing, and executing automated test routines. It enables users to write test scripts in Python using the Typhoon API, providing direct access to model parameters, signal states, and control functions. The environment supports test versioning, parameterisation, and structured test organisation through test suites and groups. Real-time execution and result logging facilitate systematic validation of control and protection schemes in Hardware-in-the-Loop simulations. The testing process will make full use of the Pytest [3] library for test execution and the allure [4] framework for structured and comprehensive report generation.

2.1.5 HIL CONNECT

HIL Connect is the signal interface module developed by Typhoon HIL that enables direct interaction between the real-time simulator and external devices or systems. It facilitates the exchange of low-voltage analogue and digital signals, making it suitable for integrating protection relays, controllers, or sensors into Hardware-in-the-Loop test setups. The module supports galvanic isolation and configurable signal conditioning to ensure compatibility with various input and output requirements. HIL Connect is essential for validating behaviour under realistic operating conditions, allowing physical devices to operate within a fully controlled and repeatable simulation environment.

2.1.6 OVERVIEW

Below is an overview about how the process would look like using all the tools in the Typhoon HIL Control Centre [5]:

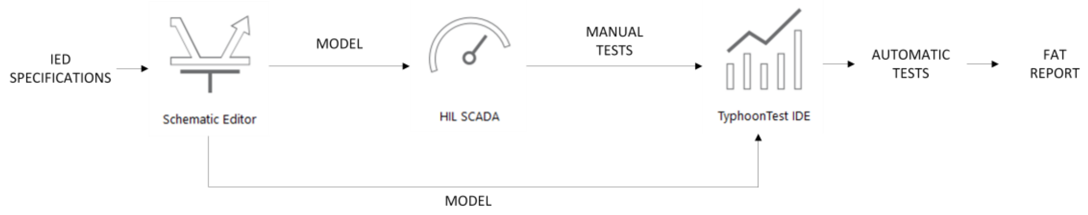


Figure 1. Workflow Schematic

2.2 HISTORICAL EVOLUTION OF CONTROL AND PROTECTION IN ELECTRICAL SUBSTATION

2.2.1 ELECTROMECHANICAL PROTECTIONS

The earliest electrical substations relied on electromechanical protection relays [6], which used physical components like coils, springs, and moving contacts to detect faults. These devices, such as overcurrent and differential relays, operated based on magnetic and induction principles. While robust and simple, they had limited accuracy and slow response times. Coordination between relays was achieved through time grading, requiring precise settings. Maintenance was manual, and fault analysis was post-event, making these systems less flexible. Despite their limitations, electromechanical protections laid the foundation for modern relaying and were widely used until the mid-20th century.

2.2.2 ELECTRONIC PROTECTIONS WITH COMMUNICATIONS

The introduction of solid-state electronics in the late 80s-90s century revolutionized substation protections. [7] [8] Static relays, using analogue circuits and later digital microprocessors, improved accuracy and speed. These relays incorporated communication capabilities, enabling remote monitoring and control via SCADA systems. Protocols like Modbus and DNP3 allowed data exchange between relays and control centres, enhancing situational awareness. While more reliable than electromechanical relays, these systems still relied on hardwired connections for tripping signals. The shift to electronic protections marked a significant step toward automation, though interoperability between devices remained a challenge.

2.2.3 IEC 61850 SUBSTATIONS WITH GOOSE MESSAGES

The IEC 61850 standard revolutionized substation communications by introducing Ethernet-based networks and Generic Object-Oriented Substation Events (GOOSE). GOOSE messages allowed high-speed, peer-to-peer communication between Intelligent Electronic Devices (IEDs), enabling real-time protection schemes like breaker failure and busbar blocking. This eliminated the need for extensive hardwiring, reducing complexity and improving reliability. Standardized data modelling (SCL files) enhanced interoperability, while advanced diagnostics simplified maintenance. However, traditional CTs and VTs were still used, meaning analogue-to-digital conversion was still required at the protection level.

2.2.4 FULL DIGITAL SUBSTATIONS WITH INTELLIGENT CTS/VTS

The latest advancement is the fully digital substation, where non-conventional sensors like optical or Rogowski coils digitize measurements directly at the source. These Intelligent Electronic Transformers (IETs) stream digital Sampled Measured Values (SMVs) via the Process Bus using IEC 61850-9-2 LE, synchronized via Precision Time Protocol (PTP). Protections process real-time data without analogue conversion, improving speed and accuracy. Merging Units (MUs) aggregate measurements, while GOOSE handles tripping signals. This architecture minimizes copper wiring, enhances cybersecurity, and supports advanced grid automation, paving the way for self-healing smart grids.

It is also worth noting the replacement of the 61850-9-2LE by the new IEC 61869-1:2023. Unlike 9-2LE, which was a limited interoperability profile, IEC 61869-9 defines a comprehensive and standardised format for SMVs, including support for multiple phases, multiple sensors, and neutral point information within a single stream. It also enables the transmission of up to 4 sets of three-phase voltages and currents simultaneously, addressing the scalability limitations of 9-2LE. Additionally, IEC 61869-9 provides explicit support for synchronisation, configuration, and timestamping, using PTP for high-precision clock alignment. These improvements ensure better interoperability, greater flexibility in data modelling, and more accurate time-critical protection operations in fully digital substations.

2.2.5 CENTRALIZED PROTECTION SYSTEMS IN SUBSTATION AUTOMATION

A significant evolution in substation protection was the introduction of centralized protection [9] systems, which integrated multiple protection and control functions into a single platform. These systems replaced standalone relays with a unified architecture, where a central processing unit handled various protection schemes, such as distance, differential, and overcurrent, for the entire substation.

By consolidating protection logic in a single device, these systems reduced wiring complexity, simplified maintenance, and improved coordination between different functions. Communication capabilities allowed seamless integration with SCADA and other automation systems, enhancing real-time monitoring and control.

While centralized protection offered advantages in standardization and reduced hardware footprint, it also introduced challenges in redundancy and reliability. A single failure could impact multiple protection functions, requiring robust fail-safe mechanisms. Despite this, such systems marked a key transition toward more integrated and software-driven substation automation.

Later advancements, such as distributed protection schemes using IEC 61850, built upon these concepts while addressing redundancy concerns, paving the way for modern digital substations.

2.3 CURRENT ELECTRICAL SUBSTATIONS

2.3.1 FAT

Factory Acceptance Tests (FAT) are standardised procedures conducted at the manufacturer's site to verify that equipment or systems meet contractual specifications and functional requirements before delivery. FATs involve a combination of visual inspections, functional tests, interlock verifications, and documentation reviews, ensuring conformity to design standards and operational expectations. These tests mitigate risk by detecting faults early, facilitating corrective actions prior to installation. FATs also serve as a baseline reference for commissioning and site acceptance tests (SAT). In engineering projects, especially within industrial automation and energy sectors, FATs are essential for validating system integration, performance reliability, and overall compliance with project specifications. It is worth noting that the evolutionary process of substation technology has shifted from installations where all equipment was connected through copper cabling to communication-based links. This transition fundamentally changes the traditional approach to testing, both at the FAT and SAT stages, thereby reinforcing the increasing importance of FAT as the primary validation step.

2.3.2 ABB / HITACHI PRODUCTS

The products listed in this section are either from ABB or Hitachi. They will be specified accordingly in the title next to the name.

2.3.2.1 Hardware

2.3.2.1.1 SSC600 (ABB)

The SSC600 is a product described in the manual as "a Smart Substation device designed for protection, control, measurement and supervision of utility substations and industrial switchgear and equipment." [10] It is also compliant with IEC61850 standard for communication and is readily available to be compatible with IEDs and supports the download of "special application packages and additional communication modules".

In essence, for the scope of this project, the importance lies in the fact it must connect to a supported IED, or in this case a model of it, using an IEC61850 communication protocol and all that it entails. This will enable the SSC600 to perform its protection functionalities, which have been preprogrammed, when receiving the pertinent data from the modelled IEDs. This is the key to verifying if the model is successful: it must correctly simulate and produce the signals for the SSC600 to trip or not in the correct circumstances.

2.3.2.1.2 REX615 (ABB)

The REX615 is a product described in its manual as "a freely configurable all-in-one protection and control relay for power generation and distribution applications." [11] It is also compliant with IEC61850 standard for communication, is readily available to be "fully modular" and allows for scalability in terms of both software and hardware.

As mentioned in the scope of this project, the aim is to model this IED and then replicate it up to 7 additional times to simulate the substation and its entire protection functionality.

The key features to be highlighted are:

- Ethernet Port: Used to communicate with the upstream SSC600
- HMI: Used by the substation operator, this will NOT be modelled
- WHMI: Used by the substation operator, this will NOT be modelled
- Basic Protection System: This must be replicated in the model

2.3.2.2 Software

2.3.2.2.1 PCM600 (ABB/Hitachi)

PCM600 [12] (Protection and Control IED Manager) is an engineering software tool developed for configuring, maintaining and operating protection and control IEDs within their Relion products. It enables centralised access to IED data throughout the entire device life cycle, supporting parameter setting, communication engineering, signal monitoring, and disturbance analysis. As stated by the manufacturer, "PCM600 provides all the functionality needed to manage your protection and control IEDs throughout their entire life cycle". The interface is structured around a substation-oriented topology that mirrors real-world configurations, facilitating accurate system modelling and IEC 61850 interoperability.

Among its key features are intuitive parameter filtering, differential comparison of IED configurations, and comprehensive I/O and GOOSE message mapping between bays and station-level devices. The tool enables "IEC 61850-compliant protection and control application configuration and communication engineering in a substation-oriented manner," which proves most important for this project's scope and the debugging process involved,

2.4 OTHER SOFTWARE

2.4.1 WIRESHARK

Wireshark [13] is an open-source network protocol analyser that enables the inspection and analysis of network traffic at a granular level. It captures packets in real time and decodes numerous communication protocols, allowing for detailed examination of network behaviours and message structures. Within the context of the project, it is particularly relevant for the IEC 61850 communication in substations to debug SV and GOOSE messages.

In SV communication, Wireshark facilitates verification of sampling rates, sequence continuity, and timestamp integrity. It allows the user to detect issues such as incorrect Application Identifiers (AppIDs), out-of-order sequence numbers, or malformed Ethernet frames, which may indicate misconfiguration in merging units or timing discrepancies caused by network delays or synchronisation faults.

For GOOSE messages, Wireshark enables the tracking of message state transitions through the stNum (state number) and sqNum (sequence number) fields. This aids in diagnosing issues such as repeated retransmissions, loss of messages, or delays in propagation that could compromise protection performance.

2.4.2 UA SERVER CLIENT

This program provides the interface for the OPC UA server block in the Typhoon HIL. It is marketed under a series which includes other software but in general it is used to implement “the serialization, security and transport of messages exchanged between different UA Applications.” [14]

2.5 OTHER HARDWARE

2.5.1 SWITCH

In this project, the GREYHOUND 1142 [15] switch functions as the central Ethernet infrastructure that interconnects all critical components, including the GPS clock, the SSC600 protection IED, the HIL606 real-time simulator, and the engineering workstation. It establishes the IEEE 802.3-based network over which IEC 61850 communication, such as Sampled Values and GOOSE messages, is reliably exchanged. The switch ensures that synchronisation signals from the GPS clock are distributed to maintain a common time base, that Sampled Values generated by the HIL606 reach the SSC600 for protection analysis, and that GOOSE commands are transmitted back to emulate tripping or control actions. During the engineering phase, the switch enables the workstation to configure the SSC600, load scenarios into the HIL606, and adjust network or device parameters. It is also fundamental for debugging, as it aggregates all Ethernet traffic, allowing detailed inspection with tools such as Wireshark to verify message integrity, time synchronisation, and compliance with IEC 61850 performance requirements.

2.5.2 INJECTION EQUIPMENT

2.5.2.1 Omicron

An OMICRON CMC356 is employed and connected to the REX615. This is a "six-phase test set for testing all types of protection relays" through the use of "high-current sources" and provide "solutions for system-based protection testing". [16]

An injection table is concluded from the specifications provided by the client for the current protection system. It is important to note, the exact curve breakpoints derived from the parameters are not to be tested (these are properties of the SSC600), it is only relevant for a trip to happen or not within a reasonable timeframe and value. The other testing will be carried on site with the real equipment as this can severely influence measurements.

An example for 50N and 51N is shown in Table 1:

Bay 1			
Protection		50N	51N
Time (ms)		20	20
Curve Parameters	A	28.2	28.2
	B	0.1217	0.1217
	C	2	2
	D	29.1	29.1
	E	1	1
	TYPE	IEC Def Time	IEC Norm Inv
Other Properties	In (A)	115.5	115.5
	Un (kV)	30	30
	xIn	1.44	0.44
	xUn	-	-
	Start Mult	1	1
	Time Mult	1	0.2
	Operate Delay (ms)	500	40
Trip	Margin	5%	5%
	IA (A)	0	0
	IB (A)	0	0
	IC (A)	0	0
	I0 (A)	175	53
	Min Time (ms)	520	203
	Max Time (ms)	572	223
No trip	Margin	5%	5%
	IA (A)	0	0
	IB (A)	0	0
	IC (A)	0	0
	I0 (A)	158	48
	Max Time (ms)	1000	1000

Table 1. Injection Values for 50N and 51N Protections

The operator then injects each point of a curve and checks visually if the desired outcome of trip or no trip has occurred. Then a table is filled out for each event, if there is a failure, the protection is reconfigured, and the process must be repeated again from the beginning.

(The OMICRON also supports automatic injection through the use of predetermined tables which verify the different points of the characteristic curve of each relay and this functionality will be replicated by the simulation test automation).

2.5.2.2 I/O microswitches

This is carried out with a standard protection relay testing panel. It allows for flip switching (in the case of ON/OFF), maintained pushbuttons (for lasting positions like the CB) and LEDs to verify position signals. Simple operations are cabled to the REX615 and verified one by one by the operator.

Chapter 3. LITERATURE REVIEW

3.1 EARLY ADOPTION OF HIL TESTING OF IEC61850

The work presented in 2009 by the name *Distributed Simulation of Power Systems using Real-Time Digital* Simulator [17] constitutes one of the earliest documented adoptions of Hardware-in-the-Loop (HIL) testing under the IEC 61850 standard. This integration marked a significant step forward in the digital testing of substation automation and communication protocols. The study utilised the RTDS (Real-Time Digital Simulator), leveraging its firmware to transmit sampled values in accordance with IEC 61850-9-2, allowing real-time interaction with IEDs and relays.

A key technical distinction of this early HIL approach lies in its computational resolution: the RTDS simulator achieved a minimum step time of 2 microseconds. While this could be considered state-of-the-art at the time, the minimum simulation time step of 2 microseconds employed by the RTDS system is significantly larger than the 25-nanosecond resolution (employed in the case of high speed DC/DC converters) supported by more recent hardware-in-the-loop platforms such as the Typhoon HIL 606, commercially available since 2021. [18] This substantial reduction in time step exemplifies the marked progress in real-time simulation fidelity, enabling accurate emulation of high-frequency switching events and complex converter dynamics.

Thus, although the RTDS-based platform from 2009 did not yet offer sub-microsecond resolution, it established an operational precedent for incorporating IEC 61850 HIL testing into wide-area distributed simulation frameworks.

3.2 IED MODELLING AND HIL APPROACH

The modelling of Intelligent Electronic Devices (IEDs) within digital substations has increasingly relied on HIL techniques to ensure realistic, reproducible, and safe testing environments. [19] HIL simulation frameworks enable the integration of physical or emulated IEDs with real-time simulated grid conditions, supporting comprehensive validation of protection logics, communication performance, and system-level coordination. This methodology allows testing under diverse operating scenarios, including fault transients and cybersecurity events, without endangering real infrastructure.

Existing research has highlighted the effectiveness of HIL in replicating the behaviour of cyber-physical systems while addressing cost, scalability, and interoperability constraints. Physical IEDs have been used for prototyping and functional testing, yet such setups are often limited by hardware availability and constrained communication architectures—particularly in the context of IEC 61850 protocols, where peer-to-peer messaging like GOOSE can restrict message dissemination in single-receiver configurations.

To overcome these limitations, emulated IEDs have been adopted as a scalable and flexible alternative. Their use enables the simulation of multiple logical nodes and switchgear elements using minimal hardware resources while maintaining communication compliance and functional equivalence. This approach supports the development of extensive testbeds that incorporate both electrical and communication layers, enabling performance benchmarking and the evaluation of complex automation schemes. Furthermore, these configurations facilitate detailed analysis of IED behaviour under cyber-physical attack scenarios, firmware updates, and configuration changes.

3.3 DIGITAL SUBSTATION TESTING

Recent advancements in testing methodologies for digital substations have centred around the adoption of HIL technology, which enables the integration of real protection and control devices within a simulated power system environment. [20] This approach facilitates comprehensive testing under dynamic and faulted conditions without the risks associated with field trials. The HIL setup allows for the emulation of complex transient behaviour, ensuring that protection schemes and control algorithms can be validated in real time with high fidelity. The use of real devices, such as protection relays and merging units, connected via IEC 61850 protocols ensures a realistic assessment of both functional performance and interoperability across digital substation components.

The application of HIL has also supported the testing of advanced substation architectures, including the validation of SV and GOOSE messaging. The platform permits the simulation of various grid scenarios while preserving deterministic communication patterns and latency constraints inherent to substation networks. Moreover, testing under such a framework enables iterative development and regression testing, which are essential for lifecycle maintenance and updates in substation automation systems. Through this process, both firmware changes and configuration updates can be safely validated, promoting more agile and secure deployment of digital substations.

Chapter 4. SCOPE OF WORK

4.1 JUSTIFICATION

The proposed project stems from a critical reassessment of Factory Acceptance Testing processes for SCADA systems and the opportunity to modernise them using Hardware-in-the-Loop technologies. Despite FAT's pivotal role in safeguarding operational integrity and system compliance in power automation deployments, its current implementation is characterised by laborious manual routines. These often entail prolonged durations, repetitive checks, and heightened risk of human error due to fatigue, all of which constrain scalability, introduce variability in testing quality, and inflate commissioning costs.

The review of existing literature confirms that while HIL-based testing has become an established method for validating protection logic and communication protocols in digital substations, particularly those utilising the IEC 61850 standard, its application has largely remained confined to functional or protection-related verification of IEDs (Ravikumar et al., 2009). Even when emulated IEDs and real-time fault scenarios are incorporated, the primary focus has been limited to device interoperability and cyber-physical reliability. However, a gap remains in integrating such testing capabilities into the pre-deployment verification of full SCADA architectures, especially with an emphasis on automation and iterative regression testing.

The present project addresses this gap by proposing an automated FAT solution built on a HIL simulation framework. This involves replicating the client's IED configuration virtually (e.g., modelling the REX615 for multiple bays), enabling real-time communication via GOOSE and Sampled Values, and systematically validating operational performance through standardised test cases. The novelty lies in the integration of Pytest and Typhoon HIL's API to facilitate test scripting, execution, and reporting in a fully automated fashion. Unlike current industrial practices that rely on engineer-driven manual validation, this approach enables machine-executable test sequences that minimise human interaction, improve repeatability, and support faster development cycles.

Moreover, by virtualising the SSC600 and embedding flexible testbench configurations, the system becomes capable of adapting to changing customer requirements or firmware updates without necessitating physical reconfiguration. This capability aligns closely with industry trends toward virtual commissioning, digital twins, and continuous integration pipelines in critical infrastructure environments.

Therefore, the justification for the project lies in the pressing industrial need for scalable, cost-effective, and reproducible SCADA FAT processes in power automation projects. By automating these procedures within a HIL ecosystem, the project directly enhances testing throughput, reliability, and adaptability, while mitigating labour intensity and reducing time-to-market. These benefits render the solution attractive to equipment vendors, system integrators, and utilities aiming to modernise their digital substation development workflows.

4.2 OBJECTIVES

The principal objective of this project is to determine the feasibility of automating FAT procedures for SCADA systems used in digital substations. This involves not only assessing whether the tests can be executed without manual intervention, but also verifying the reliability, repeatability, and scalability of such automated procedures. The project aims to demonstrate that test coverage, functional validation, and performance verification of substation Intelligent Electronic Devices (IEDs) and control schemes can be performed through a fully virtual and script-driven environment. By leveraging the capabilities of the Typhoon HIL platform and integrating it with Python-based testing libraries, the project seeks to construct a methodology that allows FAT routines to evolve from static, engineer-driven sequences into modular, automated workflows.

To support the realisation of this main goal, a set of complementary sub-objectives has been defined. These include the modelling of the REX615 protection relay for one representative bay, the development of a functional library block that can be replicated across multiple bays, and the implementation of communication protocols compliant with IEC 61850

(GOOSE and SV). Additionally, the virtualisation of the central control device SSC600 is required to complete the simulated system topology. Once individual components and their communication behaviours are validated, successive testing phases will be conducted—starting with manual execution through SCADA interfaces, followed by semi-automated tests using macros, and culminating in fully automated tests programmed via Pytest and the Typhoon HIL API. A further objective is the creation of a master testing script capable of launching all bay-specific test instances without requiring internal modifications, thereby enhancing usability and reducing setup time for engineering teams.

Nevertheless, as the project progressed, other details were added to the scope such as the creation of a virtualised SSC600 or the modelling of a REF630.

4.3 METHODOLOGY

The project has been broken down into two main parts: building the REX615 model in Typhoon HIL Schematic Editor (SE) and then testing the functionalities in Typhoon HIL SCADA (HIL-SCADA) and Test IDE.

The model must be fully functional both in terms of physical signals provided as well as communication through GOOSE and SV. For this, the starting block will be taken from the master PCM file, in this case its .scd.

The IED (REX615) to be modelled will be for Bay 1 and once the functionalities are verified and completed, a library component will be created as a basis block. Then, this solution will be replicated seven more times to complete the eight bays specified by the client in the project. Small adjustments will have to be made due to the nature of each bay varying slightly in terms of current or voltage transformers, which affect reading scaling, or connections, for example bay 3, is connected above all as the main outsource to the power grid.

The functionality will be definitive once the master protection, the SCC600, "sees a normal REX615" as if it were a physical device and not a model running in the HIL 606.

Once this is completed, phase two can begin where different SCADA panels will be built to test different protections, and it will be manually tested. Basically, the testbed which the current FAT approach uses must be replicated.

Then a semi-automated approach will be conducted where macros will be implemented. This solution will gradually grow in complexity and length.

Once this has been achieved, the fully automated testing can begin.

This will be programmed with the help of the pytest library as well as the Typhoon HIL API. The latter will provide the required connexion with the model and previously programmed HIL SCADA panels and the test IDE and is already fully compatible with Python language.

These tests will also start with one bay and then gradually more bays will be added. It is important to note that several versions of what seem to be the same script will need to be created. This is because the aim of these tests is to be able to execute them from a master file which names each individual test. This removes the need for the testing engineer to change small details inside the scripts every time for every different bay.

4.4 PLANNING AND ECONOMIC ESTIMATION

The project is estimated to cost €80,380.

This is made up four main costs which consist of:

1. Wages: Taking into account separate costs for intern and engineer hours.
2. Typhoon HIL hardware & software cost:
    a. Real-time system (HIL606, HIL Connect).
    b. Extensions such as software packages for IEC61850 support.
3. Typhoon HIL development and project management.
4. Typhoon HIL support.

The project is estimated to be completed in the span of four months, but it is subject to the client's facilitation of files, specifications and data.

The modelling is expected to take two months to two months and a half.

The SCADA and FAT automation is estimated at one month and a half to two months.

An estimated timeline is shown below in Figure 2:

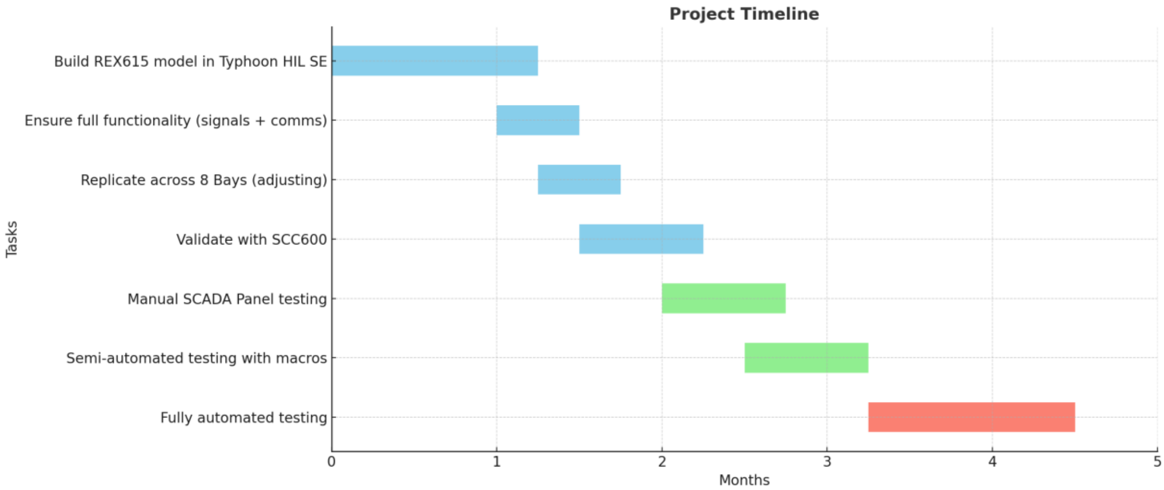


Figure 2. Project Timeline

These time constraints could be shortened should the person developing the project be employed full time or the collaboration with the client be on a day-to-day basis with both meetings and demonstrations.

The intern will dedicate around 12h-15h per week.

The total estimated engineer hours are 50h in total.

Chapter 5. MODEL CREATION

The first part is to setup the hardware and connect the equipment as per the layout. This layout is a version of the real one which will be at the substation with small modifications. This is done to keep model fidelity and accuracy.

The layout is as follows:

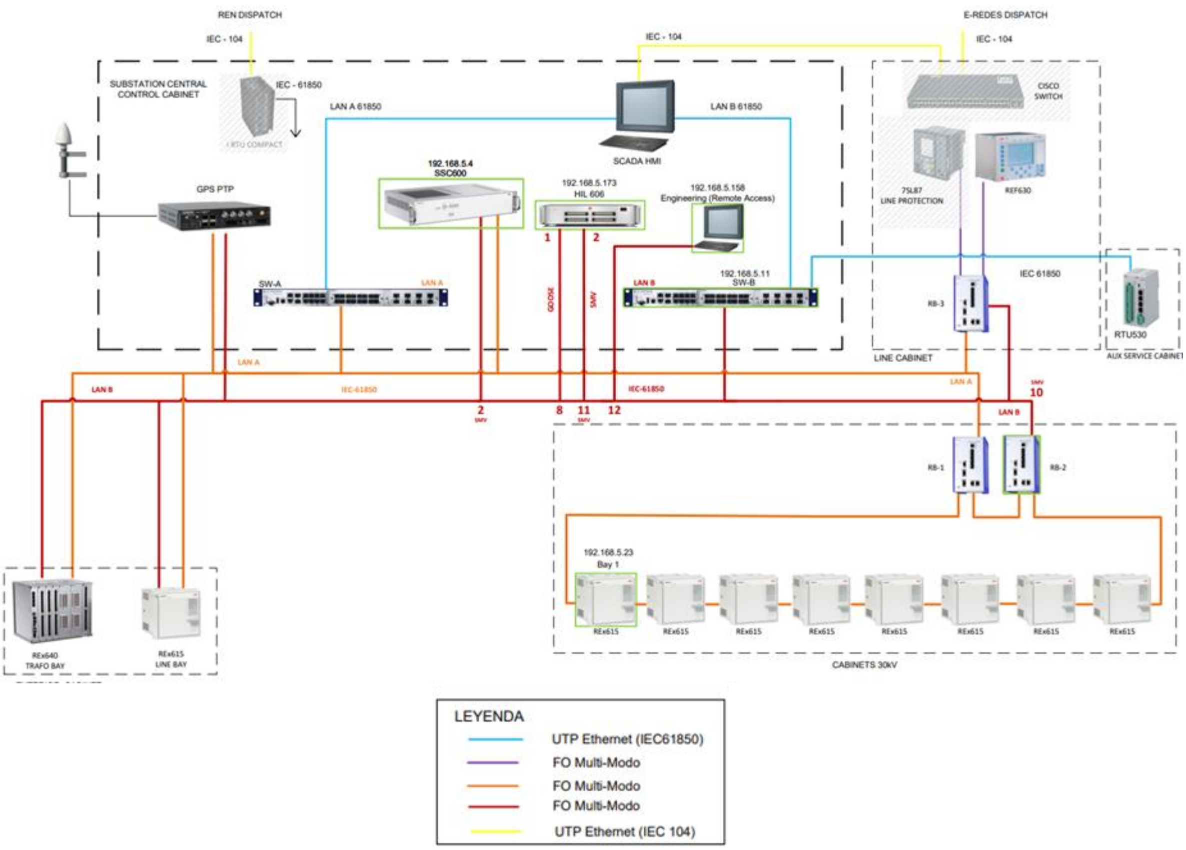


Figure 3. Communications Layout

Figure 3 shows the connection setup which will be employed at the substation while also including the development components (shown in green boxes). This setup allows for backup connectivity through the double upstream switches connected in Parallel Redundancy Protocol (PRP). For the scope of the project, only switch 2 will be analysed (LAN B).

The numbers shown in at the junction of the LAN with the different apparatus show the ethernet ports in which they are connected as well as showing where the GOOSE and SMV streams are located physically. This is all valuable information when debugging as it will help locate possible physical issues or discard them altogether and focus on the software side.

Once this is setup, the model creation is started.

The model has been created with the functionalities of the electrical part (a) separate from the communication one (b). A distinction will be made at the different levels in the model.

The model is based on a hierarchical system. The levels grow in complexity the deeper they are and go in the following order, being the first the uppermost layer:

1. Root
    a. Contains all bay components as well as the generic grid components, the power grid simulator and the constant impedances connected to each bay.
    b. Has the setup block that enables the SV and GOOSE messages
2. Bay
    a. Has connections to the grid fault simulator component
    b. Initializes SV and GOOSE through their library components
3. Library Component
    a. Contains the value that is changed to force different faults.
    b. Contains the values for the SV and GOOSE such as IP addresses.

5.1 ELECTRICAL MODEL

The electrical model has been created to be as simple as possible while keeping the resemblance to the actual system. (The more added complexity there is, the heavier the load on the model which results in longer computation times).

The grid is connected to the system through a single busbar, monitored by bay 3, that has 7 merging units connected to it. Each of these represent a connection to a different set of solar panels.

The grid has been defined as being 30kV, 50Hz and 100MVA short-circuit capacity.

The X/R ratio has been defined to be 5.

Each is connected to a constant impedance that has been defined as being of nominal 3-phase voltage 30kV and the constant power have been adapted to represent the transformer ranges and nominal currents. The following table shoes each of them:

Bay	Voltage (kV)	Power (kVA)	Current (A)
1	30	6000	115.5
2	30	6000	115.5
4	30	5200	100
5	30	520	10
6	30	6000	115.5
7	30	6000	115.5
8	30	6000	115.5

Table 2. Nominal Voltage, Power and Current for all Bays

5.1.1 ELECTRICAL OVERVIEW – ROOT LEVEL

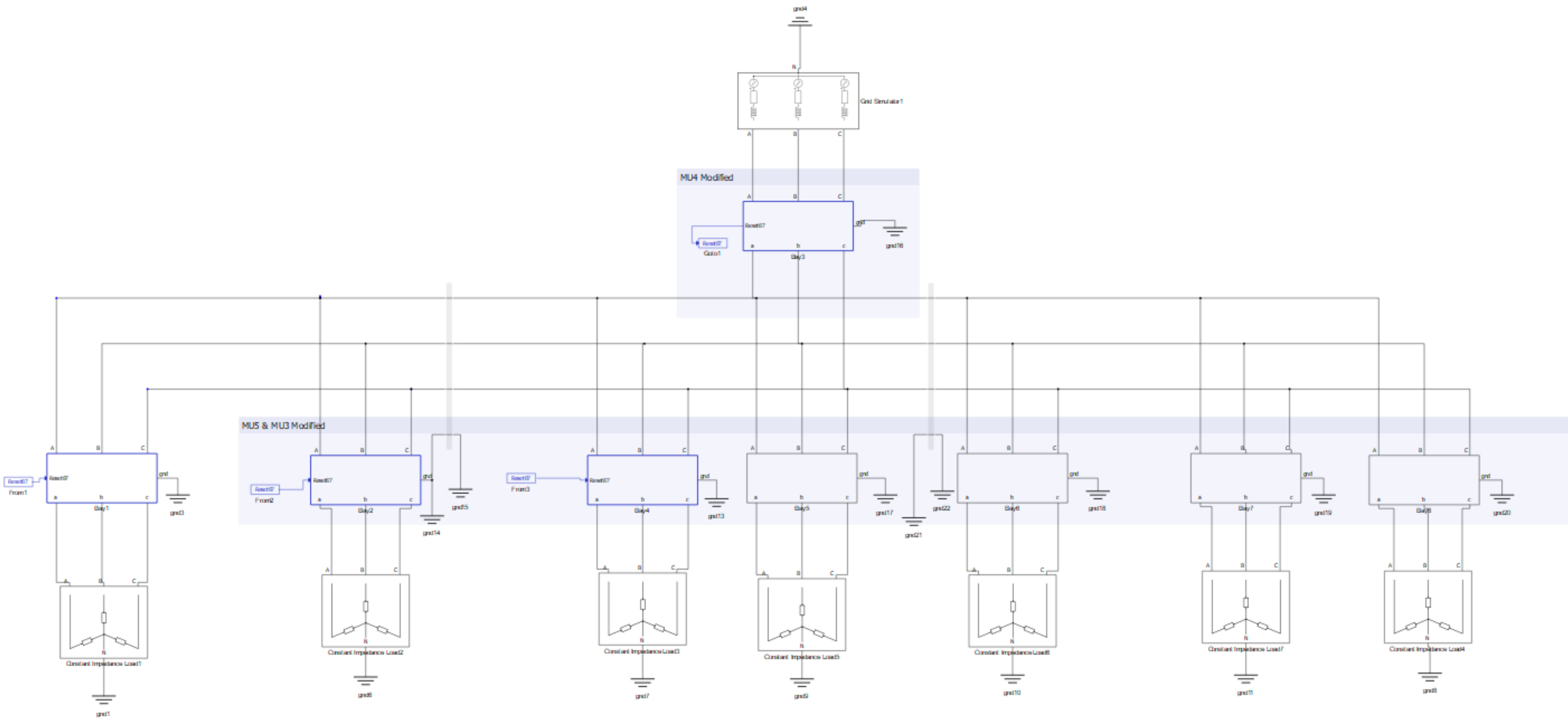


Figure 4. Model Electrical Overview (Root)

5.1.2 INDIVIDUAL REX615 BAY - ROOT LEVEL

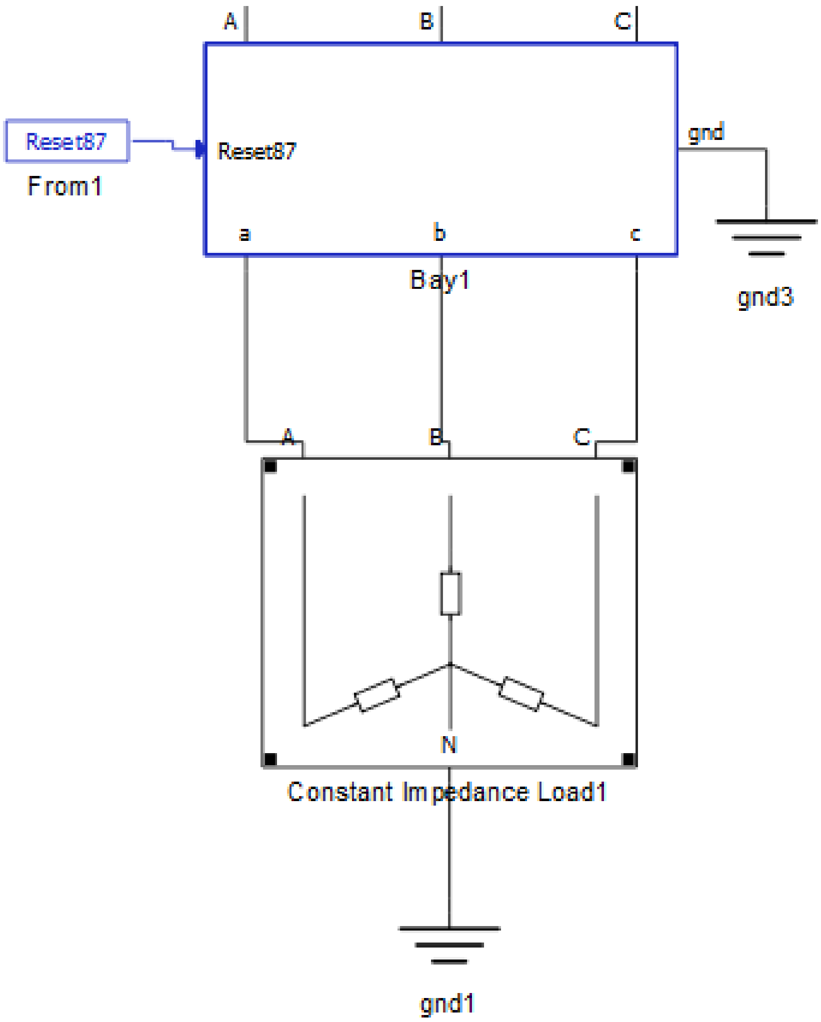


Figure 5. REX615 Electrical Overview (Root)

Features the Bay Library Component, the Constant Impedance, the connection to ground and the Reset87 block which will be explained in the Bay 3 chapter.

This Individual Bay is replicated six times and modified one time to build the general overview shown in Figure 4.

5.1.3 Electrical Components – Bay Level

5.1.3.1 Electrical (blue) link to communication (orange)

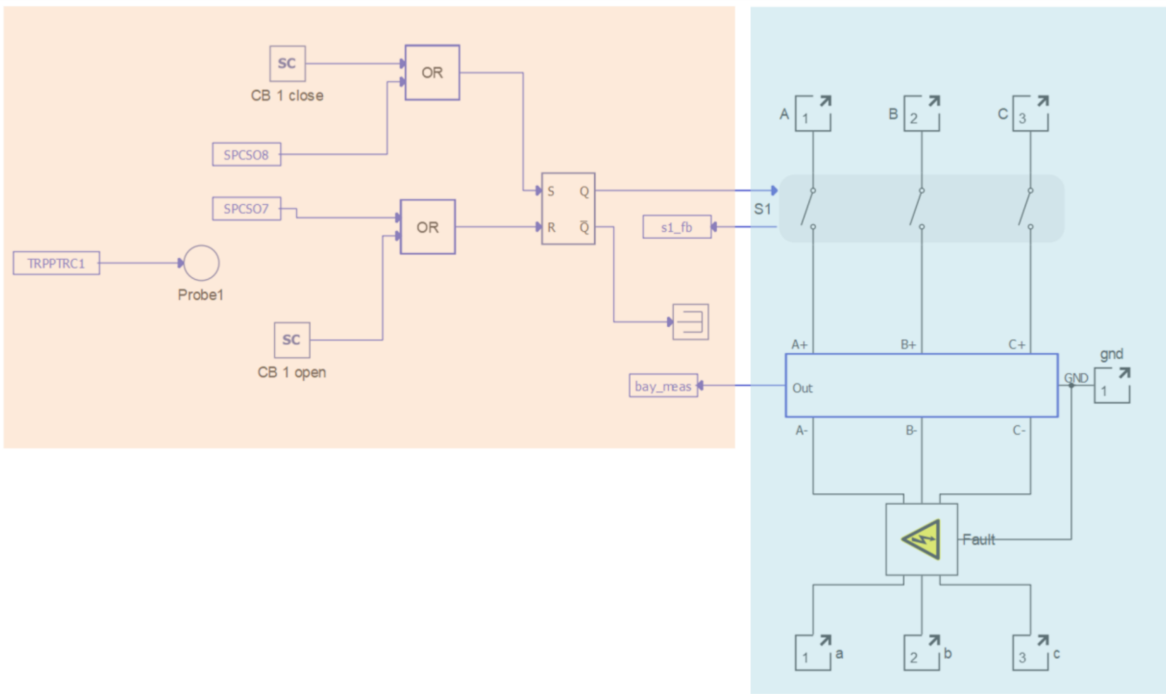


Figure 6. REX615 Electrical & Communication Overview (Bay)

This shows the connection with the previous block (it is inside the Bay Library Component). The letters match, e.g. A-A and a-a, to provide the electrical connection with the entire model at the superior layer.

It is worth noting the Fault block which enables up to 12 different faults to be activated depending on what is required, this ranges from A-N, A-B, A-B-C, A-B-C-N, and other combinations.

The values for the faults have been determined to be:

- R = 10Ω
- T_s = 100e-6

Which in turn calculates the values for L and C depending on the fault which is selected as:

$$L = T_s R = 1mH$$

$$C = \frac{T_s}{R} = 10\mu F$$

The values do not need to be anything concrete as for the scope of the project, the aim is to trip the line and see a change in the Voltage or Current measurements, not specific values.

5.2 COMMUNICATION MODEL

5.2.1 COMMUNICATION OVERVIEW – ROOT LEVEL

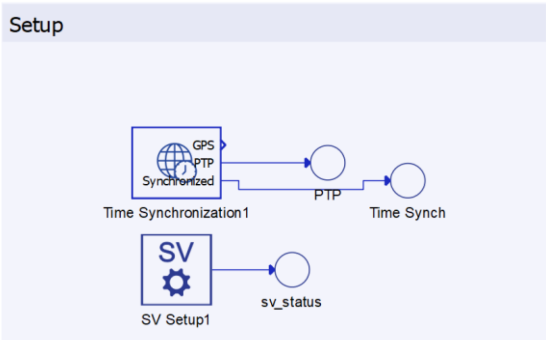


Figure 7. Communication Overview (Root)

This block determines the connection details to be able to communicate with the SSC600. If this setup is incorrect then the issues can be debugged using Wireshark and the PCM600 to verify correct IP addresses.

For the Time Synchronization Block the PTP option is checked and the IEC61850-9-3 option is selected.

For the SV Setup the ethernet port selected is number 2, the signal frequency is 50Hz and the samples per period are set to 80.

All of this is important for both model fidelity towards the real system as well as in the case of debugging and consequent analysis.

5.2.2 INDIVIDUAL REX615 BAY COMMUNICATION OVERVIEW

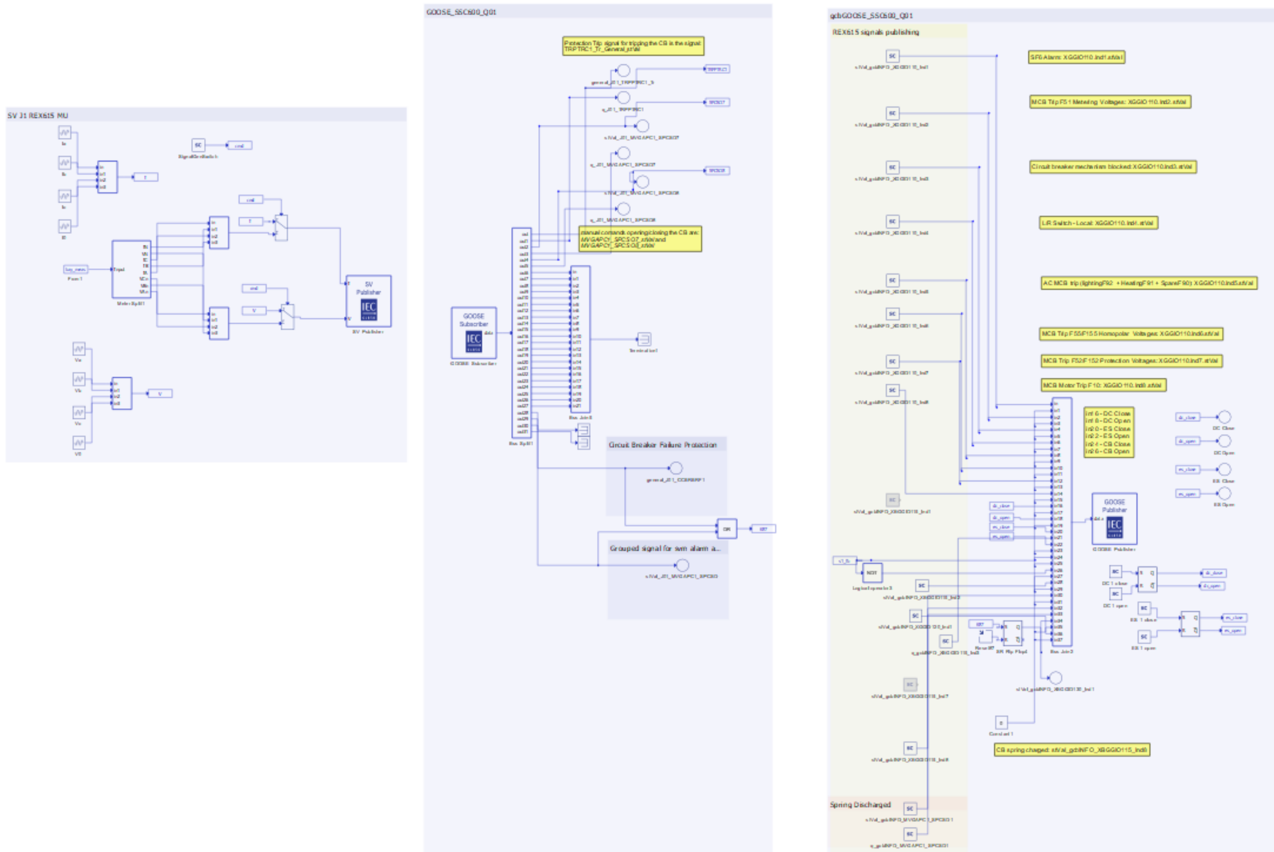


Figure 8. REX615 Communication Overview (Bay)

5.2.3 COMMUNICATION MODEL

5.2.3.1 Communication (orange) link to electrical (blue)

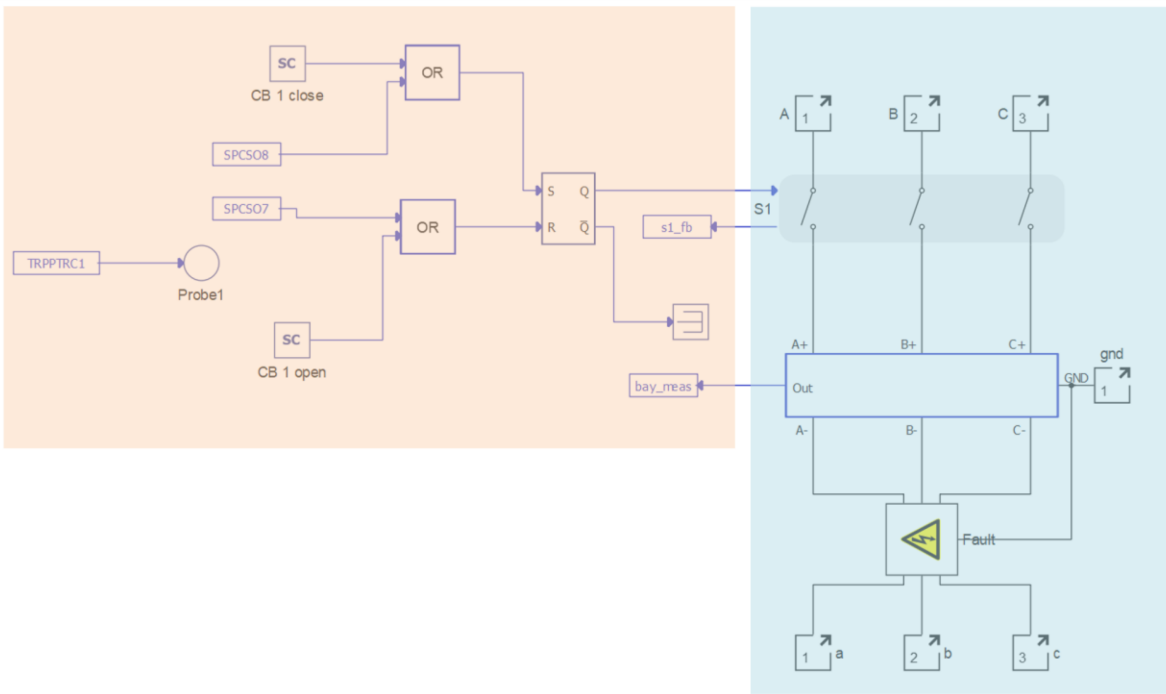


Figure 9. REX615 Electrical & Communication Overview (Bay)

5.2.3.1.1 S1 Switch

It can be seen the electrical S1 switch can be actioned through two positive inputs: CB 1 Close and SPCS08, or two negative inputs: CB 1 Open and SPCCS07. A flip-flop block is put in place to ensure the output Q is correct. The table below demonstrates the functionality:

S	R	Q	Q̄
0	0	No change	No change
1	0	1	0
0	1	0	1
1	1	-1	-1

Table 3. Flip Flop Switch Logic

This ensures the Output Q is stored and only changed should a new input be fed. Nevertheless, attention should be drawn to the fact the output can be -1 in case of forcefully testing the Open CB and Close CB at the same time.

5.2.3.1.2 Bay_meas block

A distinction must be made between the SCADA operated buttons (CB 1 close and CB 1 open) and the signals received from the SSC600 (which operate GOOSE signals for Open CB and Close CB, respectively named: SPCSO7 and SPCSO8) and the bay_meas which sends the value of the position's Voltage and Current (through the SV channel).

5.2.3.2 SV

5.2.3.2.1 Overview

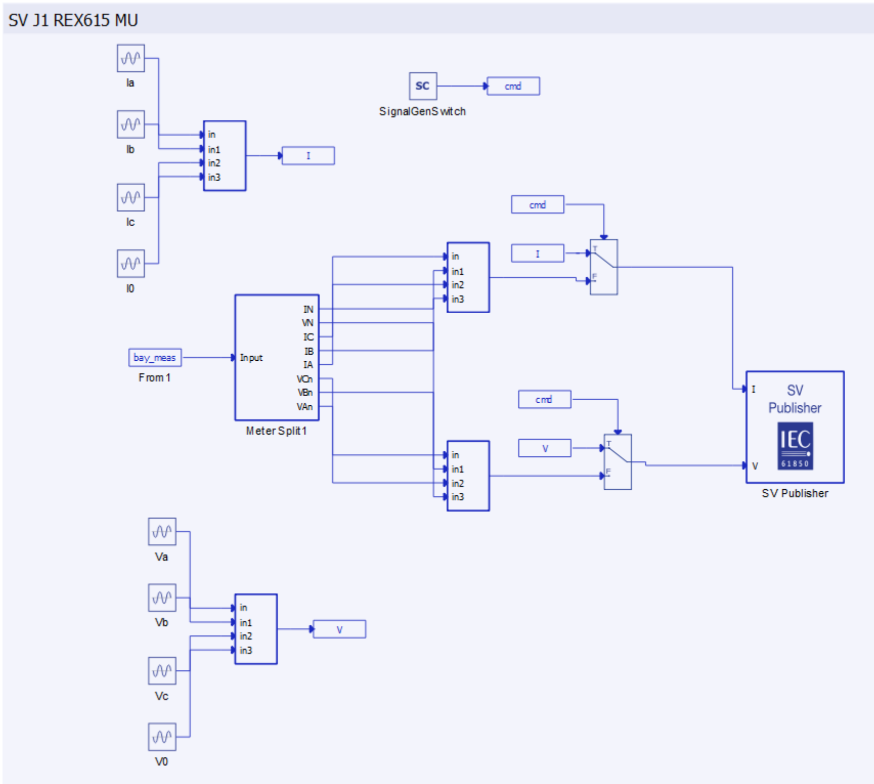


Figure 10. SV Communication Overview

The functionality of this is as follows:

1. There are two available connections for the SV Publisher. One for the current and the other for the voltage.
2. There are two available connections for each current and voltage: the "natural" electrical values from the real model and the "fake" injected current and voltage which can be set to any arbitrary value. This is done through the "SignalGenSwitch" command.

Regardless of the input, the SV Publisher block is responsible for the transmission of the values through the predetermined channel upstream to the SSC600.

5.2.3.3 GOOSE

The workflow for the GOOSE signals is bidirectional. This means the model must be able to send GOOSE signals through the GOOSE Publisher and receive through the GOOSE Subscriber. The workflows for both are very similar but the signals which go through each channel are completely different.

The following analysis is made for 2 different signals: 1 for the outward direction from the model and 1 for the inward direction.

5.2.3.3.1 Signal Mapping

The following illustrates the complete process for how the signals are mapped.

1. The signal is given by the specifications of the client, in this case by the solar panel client.
2. The signal is assigned a name in the excel sheet following an order and code. E.g. WA1\FOHC1_X615\LD0\XGGIO110\Ind1\stVal
    a. WA1 – Network Name
    b. FO – Foral Project
    c. HC1 – Bay name
    d. X615 – REX615
    e. LD0 – Logical Device Name
    f. XGGIO110 – Signal name
    g. Ind1 – Index for that signal
    h. stVal – Value of signal (1 or 0)
3. This is fed into the PCM600 project
4. The PCM600 produces an .scd file
5. This file is fed into the GOOSE Subscriber or Publisher
6. The required signal is selected
7. It is mapped into the signal Output or Input respectively

(It is important to note, all GOOSE messages treated in this project have a value (stVal) and a quality (q) that indicates if the signal is valid or not. For the purposes of the modelling all qualities are set to 0 meaning they are "good" quality and will be accepted by the system.)

5.2.3.3.2 Signal Flow Example

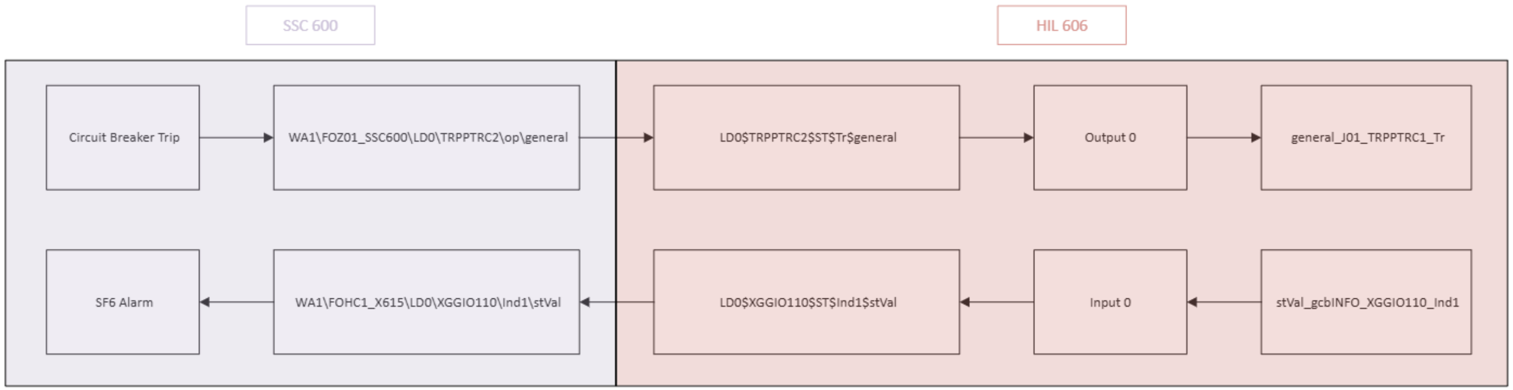


Figure 11. Signal Transmission from SSC600 to HIL606

The "Circuit Breaker Trip" signal is sent by the SSC600, it for example has tripped because of a Busbar Differential Protection Signal. The signal is passed onto the HIL 606 which in turn activates its "general_J01_TRRPTRC1_Tr" signal which can be used by the model to open the CB.

The second case is a separate example which demonstrates the workflow operating in the backwards direction. The input "stVal_gcbINFO_XGGIO110_Ind1" is set to 1 which continues through the channel until the signal is received by the SSC600 as an SF6 alarm. This can then be used by the SSC600 as a signal to operate or set off any pertinent alarms.

5.2.3.3.3 Overview - Subscriber

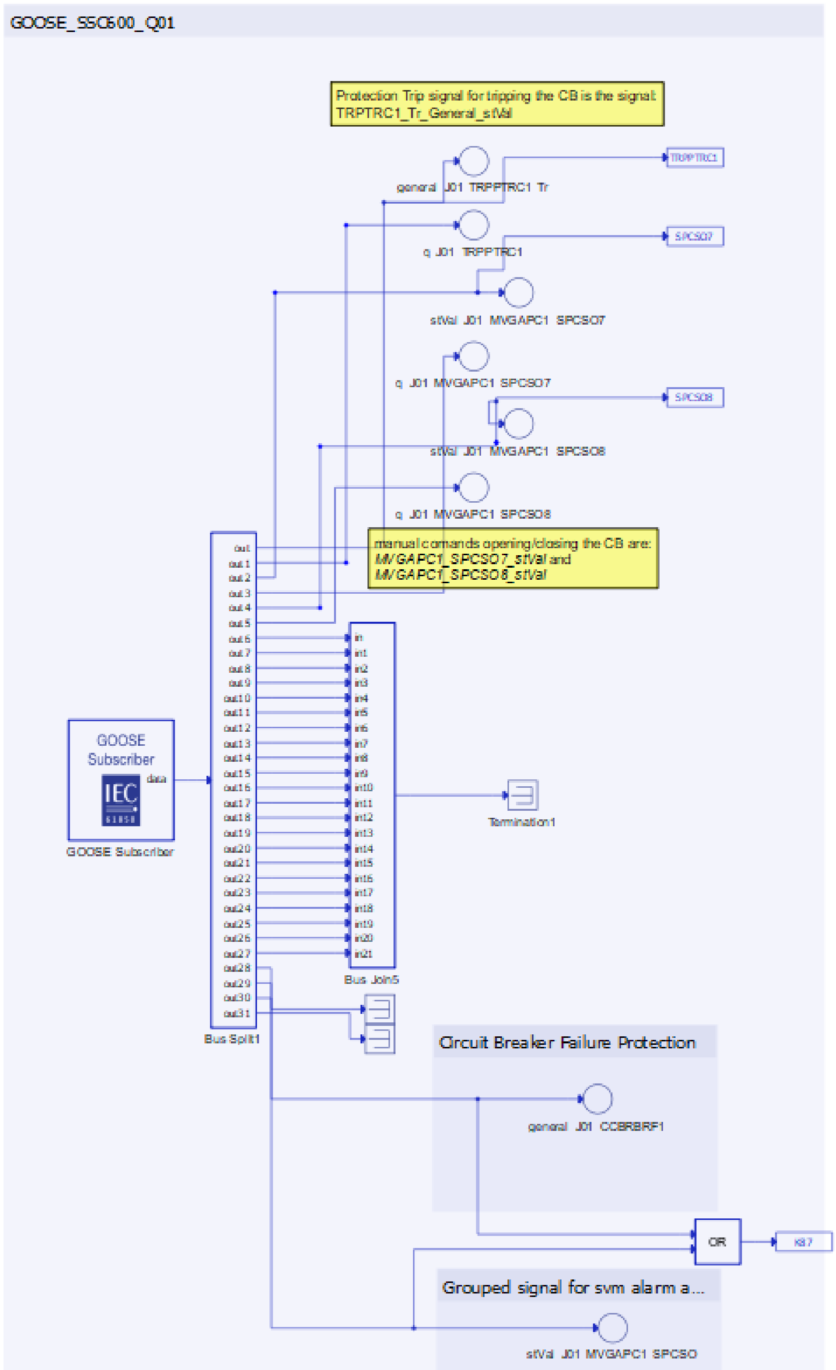


Figure 12. GOOSE Subscriber

The subscriber is responsible for receiving all signals sent by the SSC600 through the GOOSE channel to the modelled REX615.

There are two types of output which are needed:

- Signal outputs (represented by the circles): These enable them to be used by the HIL SCADA or the Test IDE.
- Tags (represented by the rectangles): These enable them to be used by the Schematic Editor in other parts of the model.

All inputs which are not needed at the present scope project are joined together and connected to a “termination” for completeness.

In addition, it is worth noting the last two signals and single tag. These are responsible for activating the K87 signal which represents the busbar differential trip signal. This must be propagated to the other bays as a failure in one bay must be notified and acted upon by the rest of the bays.

The signals that trip are:

1. The circuit breaker failure
2. SV alarm

5.2.3.3.4 Individual Subscriber

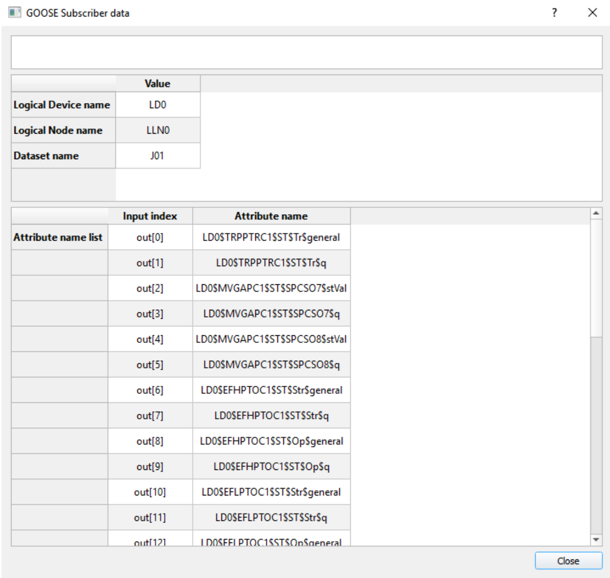


Figure 13. GOOSE Subscriber Parameters Linkage

This is the setup inside the GOOSE Subscriber block. It is responsible for linking the outputted .scd file values to the correct signals and blocks.

5.2.3.3.5 Overview – Publisher

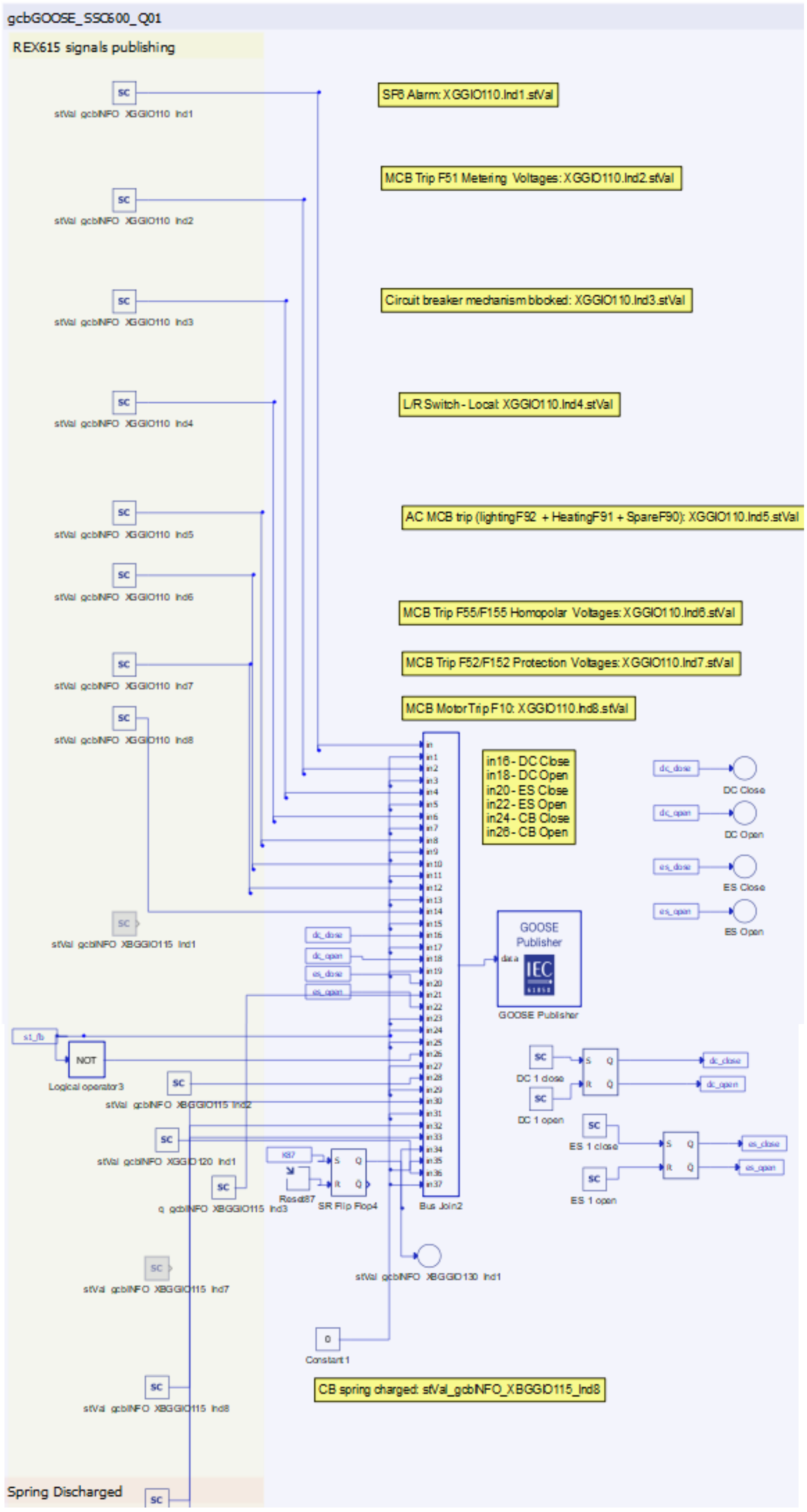


Figure 14. GOOSE Publisher

The publisher is responsible for sending all signals inputted through the SCADA panel or the Test IDE to the SSC600 through the GOOSE channel.

A note to be made is the fact that all the even inputs (or odd inX in the Bus Join2) are the quality signals corresponding to the odd inputs (or even inX in the BusJoin2). The quality signals are forced to quality 0 as per mentioned before.

In addition, it can be seen in the input in34 the functionality of the K87 being inputted into the Publisher to be sent to the other bays through the GOOSE protocol.

5.2.3.3.6 Individual Publisher

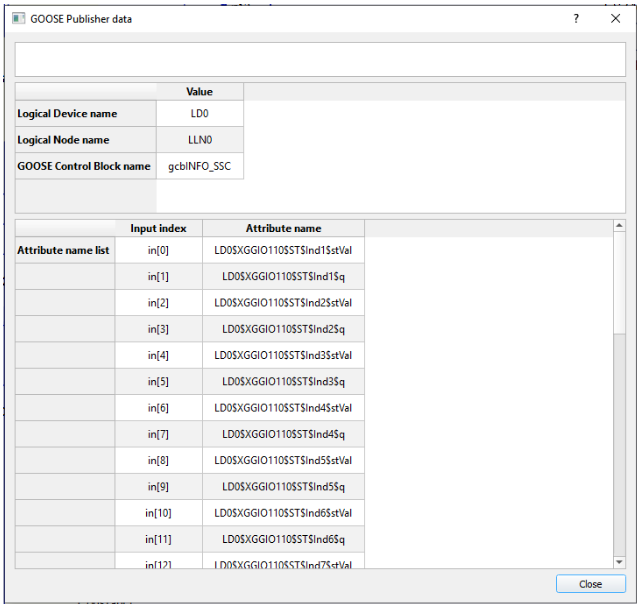


Figure 15. GOOSE Publisher Parameters Linkage

This is the setup inside the GOOSE Publisher block. It is responsible for linking the inputted SCADA inputs to the .scd file values.

5.2.4 OPC UA

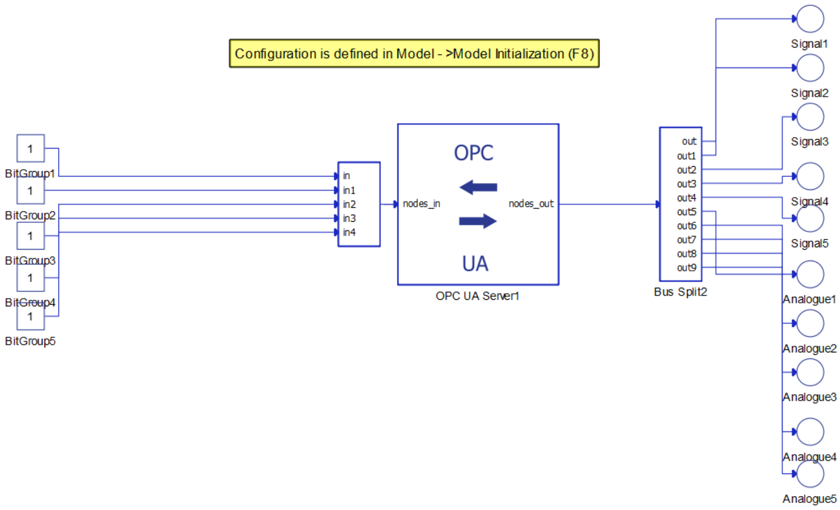


Figure 16. OPC UA Server Setup

The following code is employed to set-up the OPC-UA server:

```
config1 = {
            'ip_addr': '192.168.5.173',
            'port': 16664,
            'netmask': '255.255.255.0',
            'nodes_in' : {
                            'cmd_in1': (0, 'int'),
                            'cmd_in2': (1, 'int'),
                            'cmd_in3': (2, 'int'),
                            'cmd_in4': (3, 'int'),
                            'cmd_in5': (4, 'int'),
                        },
            'nodes_out' : {
                            'cmd_out1': (0, 'int'),
                            'cmd_out2': (1, 'int'),
                            'cmd_out3': (2, 'int'),
                            'cmd_out4': (3, 'int'),
                            'cmd_out5': (4, 'int'),
                            'analogue1': (5, 'real'),
                            'analogue2': (6, 'real'),
                            'analogue3': (7, 'real'),
                            'analogue4': (8, 'real'),
                            'analogue5': (9, 'real'),

                        },
          }
```

The most important things to take away from this are:

- The OPC-UA server is bidirectional on its own and does not need a separate Publisher and Subscriber.
- It must be initialized with code.
- The input consists of 5 integers.
- The output is made up of 5 integers and 5 real values.

This has been selected to be able to receive a coded binary number and send one back, through the integers, while sending up to 5 simultaneous analogue values back (voltage, current, power, frequency and an additional free).

5.3 *SPECIAL CASES / MODIFICATIONS*

5.3.1 BAY 3

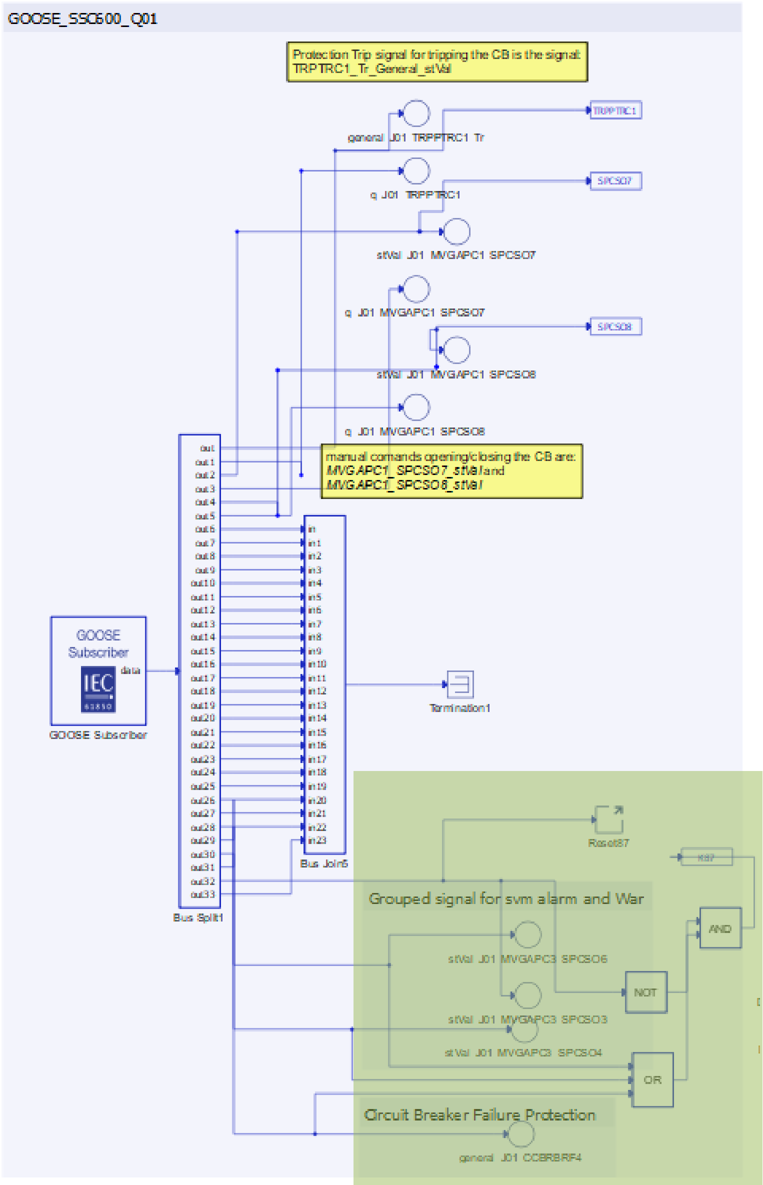


Figure 17. Modifications (Green) made to GOOSE Subscriber for Bay 3

A modification must be made as this bay is connected to all the others, which are downstream. Therefore, it is responsible for extra K87 activation cases as well as resetting the K87 once it receives the signal from the SSC600.

As it can be seen in the model, it can trip due to a combination of signals, two of them were already implemented in the other bays but three additional cases must be added to the final model solution:

1. Circuit Breaker Failure
2. SV Alarm
3. SPCSO3
4. SPCSO4
5. SPCSO5

The reset signal is sent outside the component to the Root level so that it can be distributed to all other bays and processed at each individual bay component level.

5.4 *GENERAL REMARKS*

5.4.1 SCHEMATIC EDITOR FUNCTIONALITY

The editor does not support incremental compilation, which means that any modification, regardless of scale, requires full recompilation of the model. This results in considerable compilation times, especially in large or complex systems. Therefore, careful planning and implementation of tuneable parameters is key.

5.4.1.1 Parameter Tuning and Simulation Flexibility

Most component blocks support a "tuneable" property, which permits real-time parameter adjustments without halting the simulation. This capability is essential for testing and debugging scenarios that require dynamic behaviour. For example, source blocks allow for predefined parameterisation (e.g. value and interconnections), and parameters such as voltage magnitude or frequency can be adjusted during execution if tunability is supported. This offers a flexible testing environment without the computational cost of repeated compilation.

However, not all parameters are tuneable. Phase angles in source blocks, for instance, must be initialised prior to simulation. Once the simulation begins, these parameters cannot be altered because the system loses its reference. Two main workflows exist to manage this:

1. Restart the simulation after each modification to the source configuration.
2. Pre-load all source configurations and trigger an initialization routine in the main test program

The first approach is advantageous during testing where an operator may wish to manually vary input angles for a single bay. The SCADA panel allows for fast simulation start-stop cycles, making this method useful for rapid, targeted debugging. However, the second approach proves more efficient during FAT scenarios. Here, parameter changes across multiple sources are more effectively handled through scripting or macros that automate the value assignment and initialisation processes. Although phase angle changes represent a small subset of use cases, designing around this limitation enhances model robustness.

5.4.1.2 Model Modification and Recompilation Strategy

Another significant constraint arises during iterative development. Any minor modification, such as adjusting signal scaling or correcting logic, requires full model recompilation. This was particularly problematic during debugging phases where iterative fine-tuning was necessary. To mitigate this inefficiency, a modular modelling approach was adopted. Components were grouped and isolated to form functional blocks that could be tested independently. In parallel, minimal SCADA interfaces were developed to test each module in isolation.

A notable example concerns the configuration of physical signals through the HIL Connect interface. Specific output signal ranges required dynamic scaling based on operational context. However, the HIL Connect Interface block lacks the tuneable property, and thus any modification to its configuration mandated direct model editing followed by full recompilation. This limitation prompted the implementation of signal scaling outside the block logic and encouraged further modularisation to isolate recompilation scope.

This modular approach, coupled with minimal SCADA interfaces, reduced the overall validation time during debugging and testing phases. It ensured that small, local changes did not cascade into full system recompilation, preserving developer productivity while maintaining model fidelity.

Chapter 6. ADDITIONS TO BASE MODEL

6.1 REF630

This was created after the initial model to support the modelling of another bay which had to model a REF630.

Nevertheless, the modelling of this was different as the REF630 requires physical inputs in terms of voltage and current to be able to operate. Thus, a HIL Connect solution was implemented.

6.1.1 HIL CONNECT – ROOT LEVEL

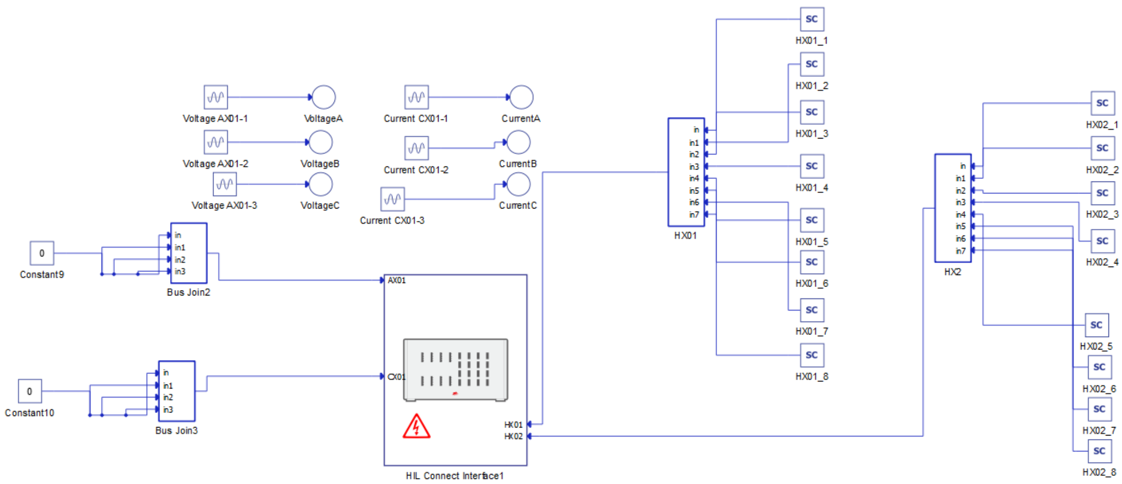


Figure 18. HIL Connect Setup (Root)

This model setup allows for the generation of 3 types of signals:

1. High-Voltage Output

   A three-phase signal in the range of ± 183.3V

2. Current Output

   A three-phase signal in the range of ± 2A

3. Digital Output
   A +5V signal is created in up to 16 different pins

6.2 VIRTUALISED SSC600

The virtualisation of the SSC600 [21] software represents a significant step forward. While not integrated into the final implementation, a virtualised version of SSC600 was successfully deployed and tested under Ubuntu Server [22] using KVM (Kernel-based Virtual Machine) and the real-time XanMod [23] kernel. This setup replicates the functional behaviour of the physical device entirely in software, enabling full-scale testing, commissioning procedures, and development workflows to be executed without physical hardware. Core functionalities, including time synchronisation using linuxptp and chrony, virtualised I/O bridging, and real-time performance tuning, were all confirmed to operate reliably in this environment. The software image and configuration files provided by ABB, once adapted to the host's specific network interfaces and tuning parameters, permit the SSC600 virtual machine to boot and operate identically to its hardware counterpart.

This virtual approach brings practical benefits for future project iterations, particularly in environments where access to physical devices is limited or where automated testing pipelines are desired. It supports continuous integration workflows by allowing engineers to simulate the substation protection and control logic within a reproducible virtual environment. The use of virsh for VM management, libvirt hooks, and startup scripts ensures consistent launch behaviour across sessions. Furthermore, it significantly reduces commissioning lead times by enabling full pre-deployment validation. This infrastructure, once integrated into the main project, opens the possibility for broader virtualised testing strategies, facilitating HIL simulations without the dependency on hardware setups.

Chapter 7. SCADA PANELS

Once the model has been constructed and has been compiled, SCADA[1] panels are created. The approach to this is incremental, thus meaning that smaller portions are created and tested before upscaling and creating for the entire model.

This chapter will display each individual SCADA panel with an image, will describe the different button, LEDs and other elements' functionalities and then give a brief summary of the operation. This will give a general idea of what was meant to be achieved both in the development phase as well as setting up the correct path for the manual, semi-automated and final automated testing.

[1] Important note: When referring to the SCADA panels these will always be the Typhoon HIL SCADA panels and not those which are used by an operator in a control room. Those are MicroSCADA panels and will not be discussed in the scope of this project.

7.1 INDIVIDUAL POSITION OPEN/CLOSE COMMANDS

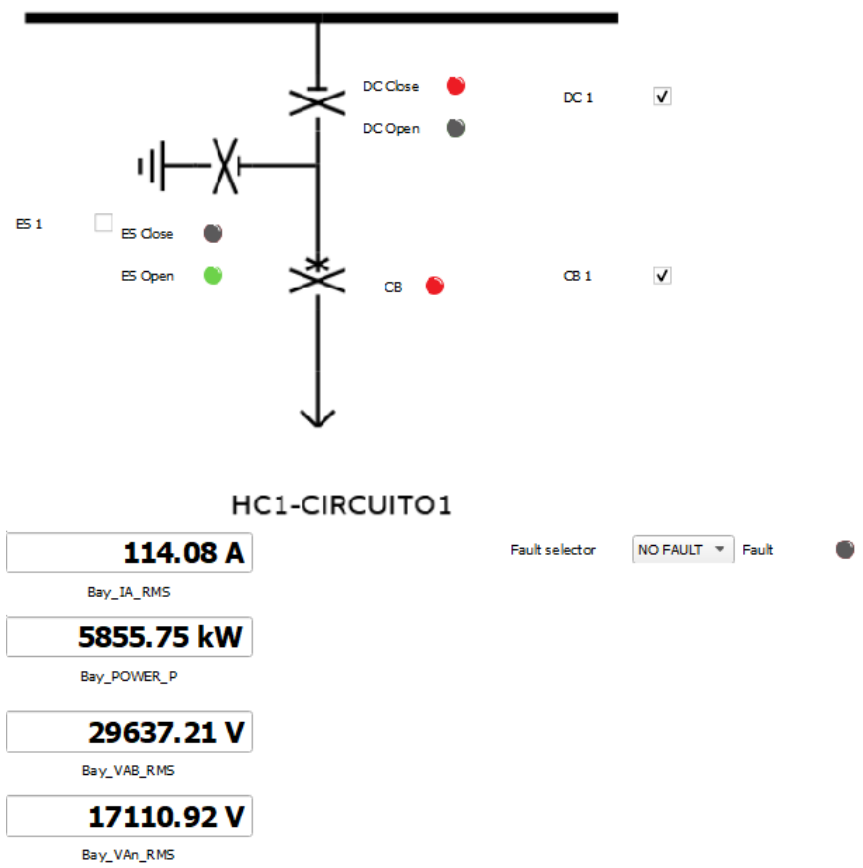


Figure 19. Open/Close Bay Position Panel

- Checkboxes for CB, DC and ES: Simulate the opening or closing of each position
- LEDs for CB, DC and ES: Show the model status
- Measurement values: Display the values the simulation is currently working with
- Fault Selector: Allows for the choice of type of fault
- Fault LED: Displays if there is a fault currently

7.2 OPC UA

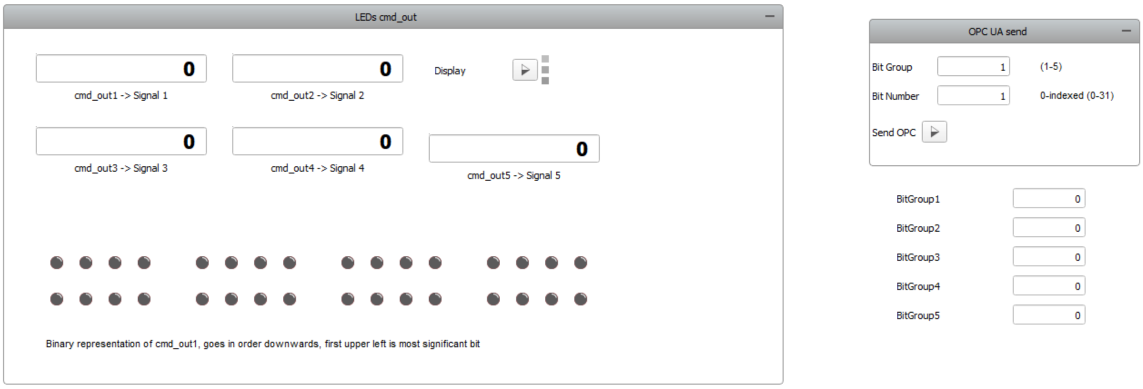


Figure 20. OPC UA Send/Receive Panel

This panel was developed primarily as a test environment to evaluate compatibility between the simulation model and the OPC UA server. A connection was established between the OPC UA server block embedded in the HIL 606 device and a UA client instance running on the MicroSCADA system.

The initial implementation aimed to verify the bidirectional exchange of signals. Particular attention was given to signal variable properties, such as type and length. The server was confirmed to support different data types, including REAL, INT, and BOOL, which could be freely configured for outgoing signals.

The process of sending signals was relatively straightforward, as they could be transmitted individually. To support this, five distinct output command variables (cmd_out1 through cmd_out5) were made available. Each one could be tested independently to ensure reliable communication.

Receiving signals presented more complexity due to the potential for multiple simultaneous inputs. The model's structure imposes constraints on input flexibility: input channels (designated as cmd_in) are statically defined at model creation and cannot be dynamically added or removed without modifying the model itself. As a result, a simplified structure is implemented to reduce the overall model complexity, avoiding the approach in which up to 200 individuals cmd_in channels would be present.

Instead, a compact encoding scheme was adopted. Five bit groups were defined, corresponding to the five logical cmd_in sources. Each group was represented as a single INT32 variable, allowing the transmission of up to 32 independent binary signals per group. A conversion algorithm was developed to decode and encode the bit positions into integers and vice versa on the receiving end, ensuring correct interpretation of signal states within the UA communication framework. An excerpt for LED visualization is shown below:

```
def create_bit_variables(int32_value):
    # Convert the integer to a 32-bit binary representation (handle both positive
and negative)
    # We use `& 0xFFFFFFFF` to mask negative numbers and ensure the binary
representation is 32 bits long
    binary_value = bin(int32_value & 0xFFFFFFFF)[2:].zfill(32)

    # Loop through the 32 bits (bit positions 0 to 31)
    for i in range(32):
        bit_value = int(binary_value[i])  # Convert string to integer (0 or 1)
        bit_dict[f"bit{i+1}"] = bit_value  # Store the integer bit value
        bool_dict[f"bit{i+1}"] = bool(bit_value)  # Convert the integer to a
boolean
        print(f"Bit number {i+1}: {bit_dict[f'bit{i+1}']} =>
{bool_dict[f'bit{i+1}']}")


def print_bit_dict():
    # Initialize output string
    output = ""

    # Split the `bit_dict` into groups of 4 bits
    for i in range(0, 32, 4):
        # Create a group of 4 bits
        group = " ".join(str(bit_dict[f"bit{j+1}"]) for j in range(i, i+4))
        output += group + " "  # Add 1 space between groups of 4

        # After every 4 bits (1 group of 4), add a new line after 4 groups (2
rows)
        if (i + 4) % 16 == 0:
            output += "\n"  # New line after every 16 bits (2 rows)

    # Print the formatted string with the output binary value
    print(f"Number in Binary is:\n{output.strip()}")

# Example usage:
```

```
int32 = hil.read_analog_signal(name='Signal1')  # Read a signal (as per your
SCADA environment)
int32 = int(int32)  # Ensure the signal is converted to an integer
create_bit_variables(int32)
print_bit_dict()
```

In addition, five analogue signals were capable of being sent from the SSC600 as type float (this was chosen to support Voltage, Current, Active Power, Reactive Power and Frequency)

7.2.1 RECEIVING FROM SSC600 (BITS)

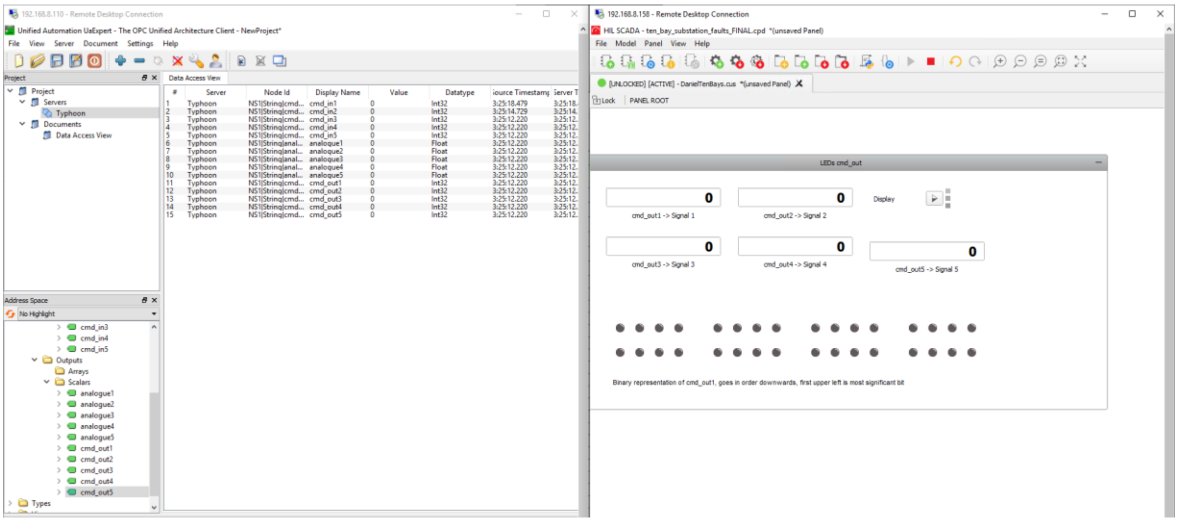


Figure 21. OPC UA Example 1: Receive Digital from SSC600 (Blank)

On the right is the SCADA panel, on the left is the UA Expert which is the gateway through which MicroSCADA accesses the OPC UA channel.

The SCADA panel features:

- 5 cmd_out which each represent a signal to be received
- 32 LEDs which light up to represent the signal cmd_out1 in binary
- Display button which makes the LEDs show up

An example is shown below in Figure 1Figure 22:

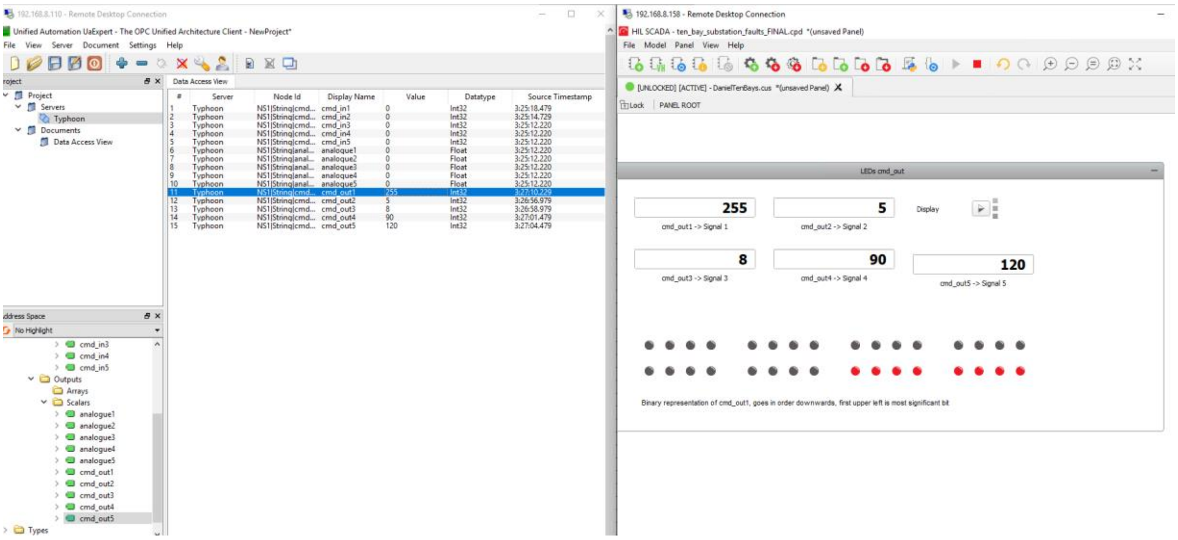


Figure 22. OPC UA Example 2: Receive Digital from SSC600 (Received)

The UA Expert is sending 5 different signals through the channel, and these are correctly received on the HIL 606 end. Timestamps show different times of receival indicating the time of input and therefore correct functioning. In addition, the number which is sent, 255, is displayed in bit form as 0000 0000 1111 1111 correctly.

7.2.2 RECEIVING FROM SSC600 (ANALOGUE)

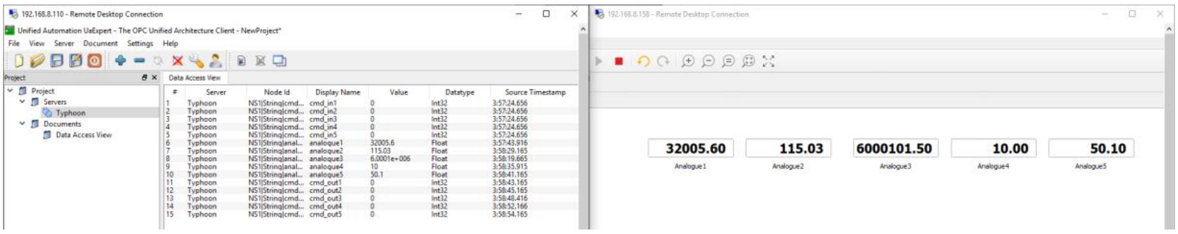


Figure 23. OPC UA Example 3: Receive Analogue from SSC600

This is a simpler implementation as only five number displays had to be created in order to correctly visualize the data. The most notable thing is the support of correct decimal visualization corresponding to the FLOAT type.

7.2.3 SENDING TO SSC600 (BITS)

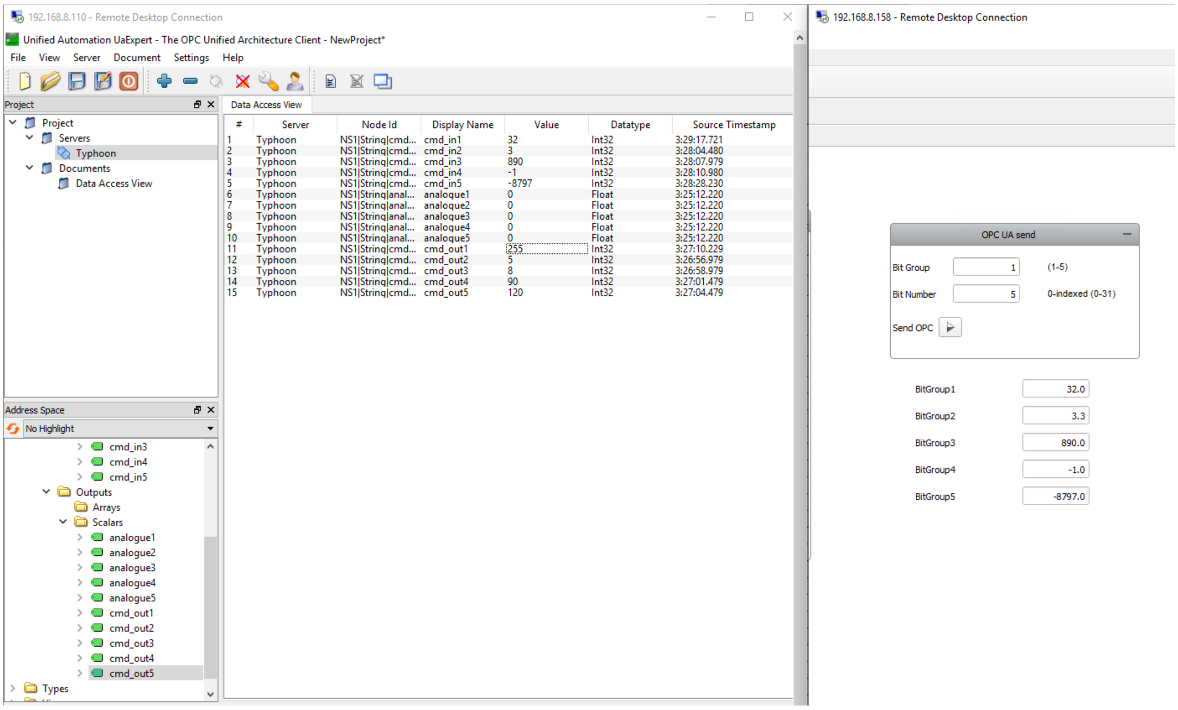


Figure 24. OPC UA Example 4: Send to SSC600

On the right side is the sending SCADA panel. It features two connected functionalities which demonstrate the incremental development aforementioned.

1. The textboxes on the bottom were the first iteration where it was tested whether each individual "BitGroup" could be sent correctly. This implied no decimals were sent and that negative numbers were supported (It can be seen in the UA Expert on the left that these functionalities work accordingly).
2. The Group Box on top is the final implementation where a "Bit Group" is selected and a Bit Number in that group is chosen. The algorithm activates when pressing the Send OPC button and directly transforms the request into binary (It again works correctly as $2^5 = 32 = 0001\ 0000$.

(A sending Analogue signals was not implemented as these are automatically sent to the SSC600 through the SV channel mentioned in Chapter 5.).

7.3 MANUAL GENERATION OF SIGNALS

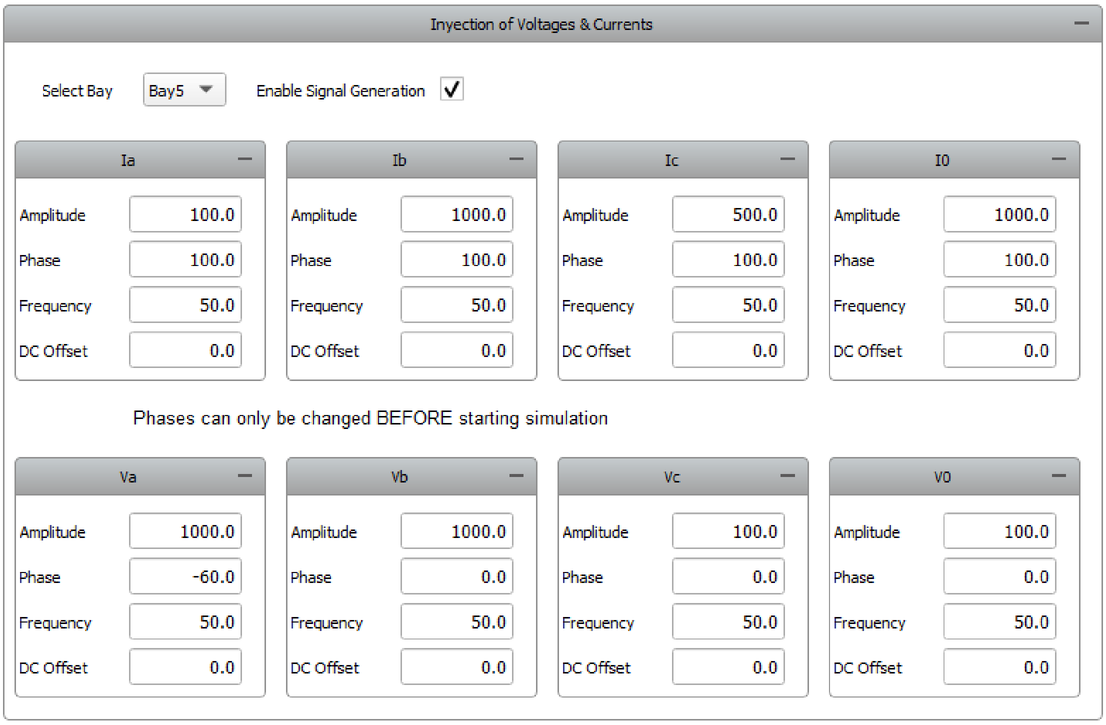


Figure 25. Manual Analogue Signal Generation Panel

- Bay Selector enables which bay to choose from.
- Enable Signal Generation can be activated/deactivated to switch between real and injected generation.
- Textboxes to fill out values.

Allows for the input of any combination of three-phase current or voltage in any of the bays. This allows for the selected bay to be temporarily "disconnected" from the electrically modelled system for fine tuning in terms of specific values.

Features variation of variables regarding Amplitude, Phase, Frequency and DC Offset for each phase and for each bay.

Chapter 8. TEST AUTOMATION

Once the SCADA Panels have been implemented and the small macros, such as that which converts the number into Bits, have been tested manually, the last step is to create the scripts.

This chapter will be similar to the last in the sense that the script will be presented and then its main functionality, defining characteristics and other important points will be made.

Nevertheless, all scripts have the same format:

1. Import the libraries to be used.
2. Set any constants that want to be employed.
3. Define the fixture, this will englobe all actions that want to be done before carrying out the test and will perform the "hil.start" as the last action to start the simulation.
4. Define the main testing where all signals are generated, observed and validated, as well as asserting if each individual test has passed or failed according to stated specifications.

Important note: Many functionalities depend on the ability to parse data accordingly from different source files. This is the reason some tests will have accompanying excel files, text files and the python scripts used to automate this process. When these are shown, shorter versions, with one or two elements being parsed, will be displayed for the sake of shortening lengthier explanations. In addition, for the most complex example, the test excerpts will be shown both for passing and failing events for completeness.

8.1 TRIP_CURRENT.PY

This script will try to operate certain protections. In the example shown, for Bay 2 (H02) protections 3x51 and 50N will have to operate in order to pass the test. The data is extracted from the excel file (trips.xlsx) which in turn is exported directly from the PCM600 file that contains the SSC600 tripping specifications.

It is important to note that the script is programmed to allow for the definition of Base Currents, Intervals at which the operations are carried out (in this case is the connection/disconnection of DC and CB), as well as maximum time permitted for the trip to occur. This allows for future personalization and adaptability.

8.1.1 SCRIPT

```
import re
import pandas as pd
import pytest
import typhoon.api.hil as hil
import time


# Define base currents
IN = 300
INR = 50


INTERVAL = 0.5
BAY_CURRENT = 2  # example value
BAY_COUNT = 2    # example value
TIME = 5         # max wait time in seconds
MARGIN = 0.1     #for the currents: 0.1 = 110%

# Read from trips.txt
with open('trips.txt', 'r') as f:
    lines = f.readlines()

records = []
for line in lines:
    match = re.match(

r"bay_name:\s*(\S+),\s*protection_type:\s*(\S+),\s*start_value:\s*([\d.]+),\s*cur
ve:\s*(\d+),\s*time_delay:\s*(\d+)",
        line.strip()
    )
    if match:
```

```
        bay_name, protection_type, start_value_str, curve, time_delay =
match.groups()
        start_value = float(start_value_str)
        curve = int(curve)
        time_delay = int(time_delay)

        base_current = INR if 'N' in protection_type else IN
        injection_value = base_current * start_value * (1 + MARGIN)

        # Define current settings based on protection type
        if '3' in protection_type:
            current_settings = {
                'Ia': {'amplitude': injection_value, 'phase': 0, 'frequency': 50,
'dc_offset': 0},
                'Ib': {'amplitude': injection_value, 'phase': 120, 'frequency':
50, 'dc_offset': 0},
                'Ic': {'amplitude': injection_value, 'phase': 240, 'frequency':
50, 'dc_offset': 0}
            }
        else:
            current_settings = {
                'I0': {'amplitude': injection_value, 'phase': 0, 'frequency': 50,
'dc_offset': 0}
            }

        records.append({
            'bay_name': bay_name,
            'protection_type': protection_type,
            'start_value': start_value,
            'curve': curve,
            'time_delay': time_delay,
            'injection_value': injection_value,
            'current_settings': current_settings
        })

def set_current(bay_number, current_settings):
    for phase, params in current_settings.items():
        for param_name, value in params.items():
            hil.model_write(f'Bay{bay_number}.{phase}.{param_name}', [value])
    hil.wait_sec(INTERVAL)

def inject_current(bay_number):
    hil.set_scada_input_value(f'Bay{BAY_CURRENT}.SignalGenSwitch', 1.0)

def stop_current(bay_number):
    hil.set_scada_input_value(f'Bay{BAY_CURRENT}.SignalGenSwitch', 0.0)

def close_cb(bay_number):
    hil.set_scada_input_value(scadaInputName=f'Bay{bay_number}.CB 1 close',
value=1)
    hil.set_scada_input_value(scadaInputName=f'Bay{bay_number}.CB 1 open',
value=0)
```

```
def open_cb(bay_number):
    hil.set_scada_input_value(scadaInputName=f'Bay{bay_number}.CB 1 open',
value=1)
    hil.set_scada_input_value(scadaInputName=f'Bay{bay_number}.CB 1 close',
value=0)

def close_dc(bay_number):
    hil.set_scada_input_value(scadaInputName=f'Bay{bay_number}.DC 1 close',
value=1)
    hil.set_scada_input_value(scadaInputName=f'Bay{bay_number}.DC 1 open',
value=0)

def open_dc(bay_number):
    hil.set_scada_input_value(scadaInputName=f'Bay{bay_number}.DC 1 open',
value=1)
    hil.set_scada_input_value(scadaInputName=f'Bay{bay_number}.DC 1 close',
value=0)

def close_dc_cb(bay_number):
    open_cb(bay_number)
    hil.wait_sec(INTERVAL)
    open_dc(bay_number)
    hil.wait_sec(INTERVAL)
    close_dc(bay_number)
    hil.wait_sec(INTERVAL)
    close_cb(bay_number)
    hil.wait_sec(INTERVAL)

@pytest.fixture(scope="session", autouse=True)
def setup_simulation():

hil.load_model(file='C:\\Users\\Administrador\\Desktop\\Final\\eight_bay_substati
on_faults Target files\\eight_bay_substation_faults.cpd')
    hil.start_simulation()
    yield
    # hil.stop_simulation()

@pytest.mark.parametrize("record", records)
def test_cb_sequence(record):
    bay_number = BAY_CURRENT
    set_current(bay_number, record['current_settings'])
    close_dc_cb(bay_number)
    hil.wait_msec(record['time_delay'])
    start_time = time.time()
    inject_current(bay_number)
    while time.time() - start_time < TIME:
        trip =
hil.read_analog_signal(name=f'Bay{bay_number}.general_J01_TRPPTRC1_Tr')
        if trip == 1:
            stop_current(bay_number)
            open_cb(bay_number)
            elapsed = time.time() - start_time
            print(f"Tripped after {elapsed:.2f} seconds")
```

```
            return
        time.sleep(0.001)
    pytest.fail(f"Did not trip within {TIME} seconds")
```

8.1.2 ADDITIONAL FILES

8.1.2.1 Excel File containing trips (trips.xlsx)

This file has been exported from the PCM600 from the SSC600 protection configuration. Only two protections have been selected and in particular for Bay 2.

	A	B
1	SSC600/Application Configuration/H02_REX615/Current protection/3x51_H02; PHLPTOC: 2/Settings/Setting Group 1	
2	Operating curve type	15
3	Type of reset curve	1
4	Time multiplier	1
5	Start value	1.2
6	Start value Mult	1
7	Operate delay time	900
8		
9		
10		
11		
12		
13	SSC600/Application Configuration/H02_REX615/Current protection/50N_H02; EFHPTOC: 2/Settings/Setting Group 1	
14	Start value	1.44
15	Start value Mult	1
16	Operating curve type	15
17	Type of reset curve	1
18	Time multiplier	1
19	Operate delay time	500
20		

Figure 26. Protection configuration for 3x51 and 50N for Bay 2

8.1.2.2 export_trips.py

This script parses the data from the excel sheet above, creating a text file with all the data, which is relevant, such as obtaining from the long string name data like Bay or Protection Type.

```
import pandas as pd
import re

# Load the Excel file
df = pd.read_excel('trips.xlsx', sheet_name='Sheet1', header=None)

# Flatten the DataFrame into a single list of strings
```

```
data = df.fillna('').astype(str).values.flatten()

# Join all data into a single text block
text = '\n'.join(data)

# Split based on the start of new protection blocks
blocks = re.split(r'SSC600/Application Configuration/', text)[1:]  # Remove empty
part before first match

results = []

for block in blocks:
    # Extract bay name and protection type
    header_match = re.match(r'([^/]+)/Current protection/([^;]+);', block)
    if header_match:
        bay_name = header_match.group(1)
        protection_type = header_match.group(2)
    else:
        bay_name = protection_type = 'N/A'

    # Extract data values
    start_value_match = re.search(r'Start value\s*([\d.]+)', block)
    curve_type_match = re.search(r'Operating curve type\s*(\d+)', block)
    delay_match = re.search(r'Operate delay time\s*(\d+)', block)

    start_value = start_value_match.group(1) if start_value_match else 'N/A'
    curve_type = curve_type_match.group(1) if curve_type_match else 'N/A'
    time_delay = delay_match.group(1) if delay_match else 'N/A'

    results.append(f"bay_name: {bay_name}, protection_type: {protection_type},
start_value: {start_value}, curve: {curve_type}, time_delay: {time_delay}")

# Write to text file
with open('trips.txt', 'w') as f:
    for line in results:
        f.write(line + '\n')
```

8.1.2.3 trips.txt

Resulting file from export_trips.py

```
bay_name: H02_REX615, protection_type: 3x51_H02, start_value: 1.2, curve: 15,
time_delay: 900
bay_name: H02_REX615, protection_type: 50N_H02, start_value: 1.44, curve: 15,
time_delay: 500
```

8.1.3 TEST PASSED

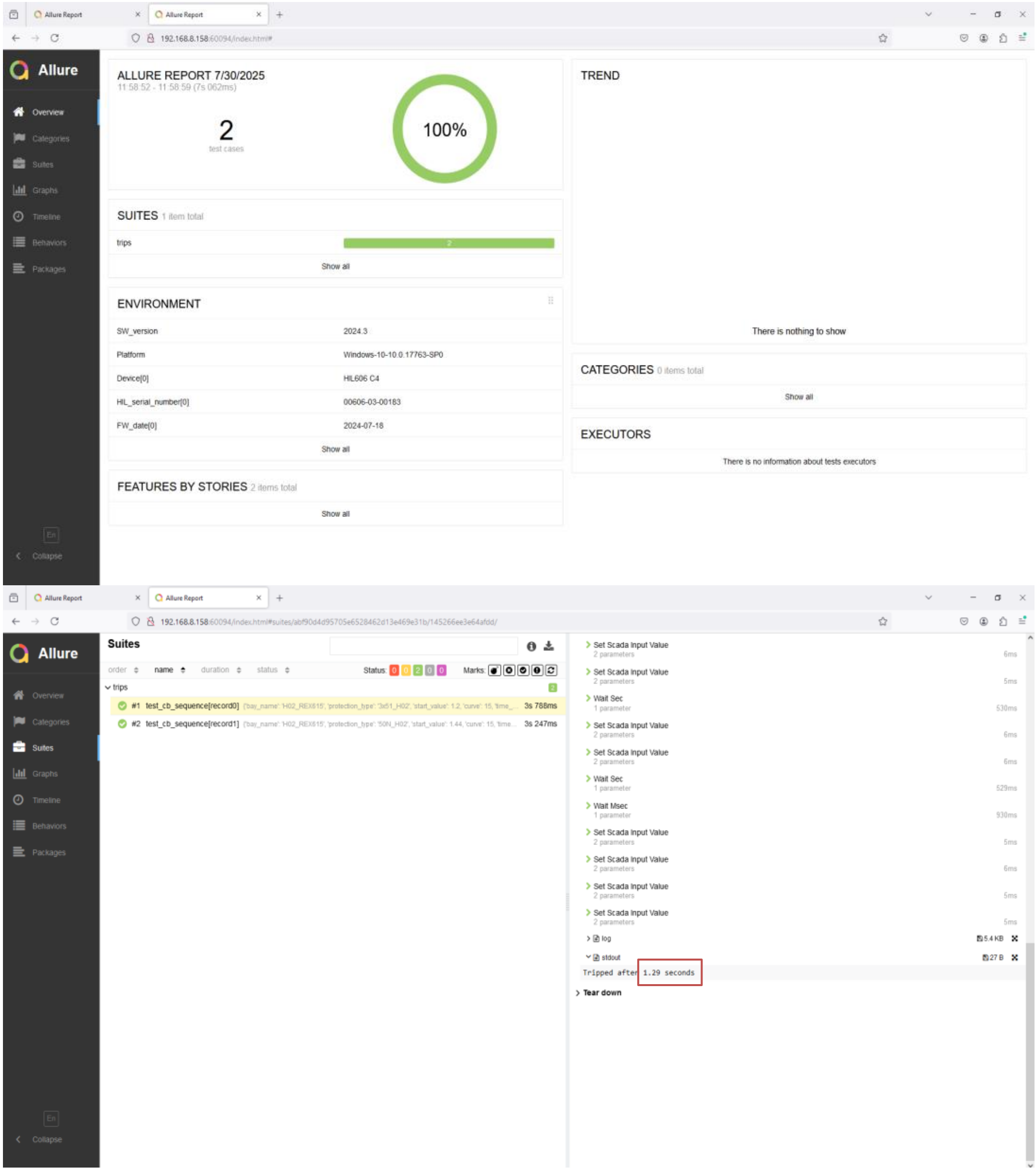


Figure 27. Passed test for 3x51 and 50N for Bay 2

Above: Both tests have passed and the test was completed in just over 7 seconds.

Below: shows the test has passed in 1.29 seconds showing the 3x51 has tripped. Further details are provided such as the input values or the steps the test has performed and at what time intervals.

8.1.4 TEST FAILED

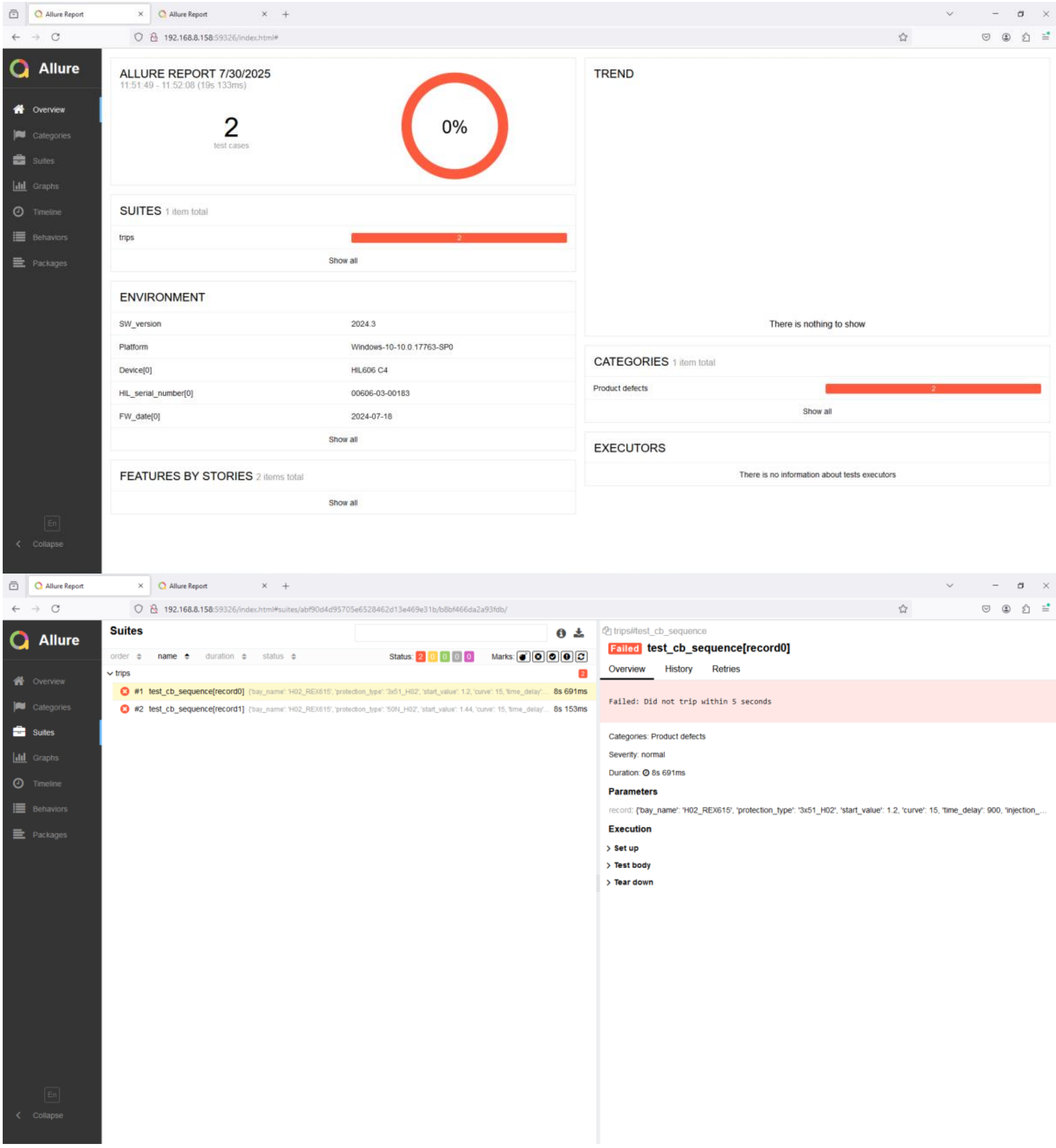


Figure 28. Failed test for 3x51 and 50N for Bay 2

Above: Both tests have failed as shown at the top.

Below: The 3x51 did not trip within the specified 5 seconds.

8.2 *BIT_OPC_UA.PY*

This script is the building block for the full circle communication. It takes signals from the signal_name.txt file, which in turn has been generated from an excel sheet that selects the signals which want to be generated and sent through the OPC UA channel. It then waits for a return signal in the specified bit group and bit to be received.

8.2.1 SCRIPT

```
import pytest
import typhoon.api.hil as hil

# Analysis duration (s)
duration = 5

# Test interval time (s)
interval = 1

# Signal list text file name
file_name = 'signal_name.txt'


@pytest.fixture(scope="module")
def setup():
    """
    Load the simulation model and start the simulation.
    """
    hil.load_model(file='single_bay_with_opc Target
files\\single_bay_with_opc.cpd',
                   vhil_device=False)
    hil.start_simulation()


def load_test_data(file_path):
    """
    Load test data from the provided file and parse the signal names, OPC, and
bit information.
    """
    with open(file_path, 'r') as file:
        test_data = []
        for line in file:
            if line.strip():  # Skip empty lines
                parts = line.split(':')
                signal_name = parts[0].strip().strip("'")
                rest = parts[1].strip().strip('(),')  # Remove enclosing
parentheses and commas
                values = rest.split(',')
                opc = values[2].strip().strip("'") if len(values) > 2 else ""
```

```
                bit = values[3].strip().strip("'") if len(values) > 3 else ""
                test_data.append((signal_name, opc, bit))
        return test_data


def create_bit_variables(int32_value, opc_number):
    """
    Create bit and boolean dictionaries for a given int32 value and OPC number.
    """
    # Convert the integer to a 32-bit binary representation
    binary_value = bin(int32_value & 0xFFFFFFFF)[2:].zfill(32)
    bit_dict = {}
    bool_dict = {}

    # Populate the bit and boolean dictionaries for the given OPC
    for i in range(32):
        bit_key = f"opc{opc_number}_bit{i+1}"
        bit_value = int(binary_value[i])
        bit_dict[bit_key] = bit_value
        bool_dict[bit_key] = bool(bit_value)

    return bit_dict, bool_dict


@pytest.mark.parametrize("signal_name, opc, bit",
load_test_data("signal_name.txt"))
def test_multiple(setup, signal_name, opc, bit):
    """
    Test function that evaluates the specific bit in the signal for the given OPC
and bit index.
    """
    hil.wait_sec(interval)

    # Set SCADA input value for the given signal name
    hil.set_scada_input_value(signal_name, 1)

    # Determine which signal to check based on the OPC value
    opc_number = int(opc)  # Convert OPC to integer
    target_signal_name = f"Signal{opc_number}"  # Map OPC to the corresponding
signal

    # Read the int32 signal value dynamically for the target signal
    int32_value = hil.read_analog_signal(name=target_signal_name)
    int32_value = int(int32_value)  # Ensure it's an integer

    # Create bit variables for the given OPC
    bit_dict, bool_dict = create_bit_variables(int32_value, opc_number)

    # Construct the bit key and retrieve the specific bit value
    bit_key = f"opc{opc_number}_bit{bit}"
    assert bit_key in bool_dict, f"Bit key {bit_key} not found in bool_dict"
    specific_bit_value = bool_dict[bit_key]
```

```
    # Log for debugging
    print(f"Testing {signal_name} for OPC={opc}, Bit={bit} on
{target_signal_name} with value={specific_bit_value}")

    # Assert the rule: Verify the specific bit in the boolean dictionary
    expected_bit_value = True  # The bit should be True
    assert specific_bit_value == expected_bit_value, (
        f"Bit {bit} in OPC {opc} for {target_signal_name} is
{specific_bit_value}, "
        f"expected {expected_bit_value}"
    )
```

8.2.2 ADDITIONAL FILES

8.2.2.1 Lookup.xlsx

	A	E	F	G
1	UN	IN	OPC	BIT
91	0			
92	0			
93	0			
94	61	stVal gcbINFO XGGIO110 Ind1	1	1
95	62	stVal gcbINFO XGGIO110 Ind2	1	2
96	63	stVal gcbINFO XGGIO110 Ind3	1	3
97	64	stVal gcbINFO XGGIO110 Ind4	1	4
98	65	stVal gcbINFO XGGIO110 Ind5	1	5
99	66	stVal gcbINFO XGGIO110 Ind6	1	6
100	67	stVal gcbINFO XGGIO110 Ind7	1	7
101	80	stVal gcbINFO XGGIO110 Ind8	1	8
102	81	stVal gcbINFO XBGGIO115 Ind1	1	9
103	82	stVal gcbINFO XBGGIO115 Ind2	2	10
104	83	stVal gcbINFO XBGGIO115 Ind3	2	11
105	85	stVal gcbINFO XBGGIO115 Ind4	2	12
106	0	stVal gcbINFO XBGGIO115 Ind5	2	13
107	0	stVal gcbINFO XBGGIO115 Ind6	2	14
108	192	stVal gcbINFO XBGGIO115 Ind7	2	15
109	192	stVal gcbINFO XBGGIO115 Ind8	2	16
110	192	stVal gcbINFO MVGAPC1 SPCSO1	2	17
111	192	stVal gcbINFO MVGAPC1 SPCSO2	2	18
112	192	stVal gcbINFO MVGAPC1 SPCSO3	2	19
113	192	stVal gcbINFO MVGAPC1 SPCSO4	2	20
114	192	stVal gcbINFO MVGAPC1 SPCSO5	1	21
115	192	stVal gcbINFO MVGAPC1 SPCSO6	1	22
116	192	stVal gcbINFO MVGAPC1 SPCSO7	2	23
117	192	stVal gcbINFO MVGAPC1 SPCSO8	1	24
118	192	stVal gcbINFO TCSSCBR1 CircAlm	2	25
119	192	stVal gcbINFO XBGGIO120 Ind3	1	26

Table 4. Signals to be tested by OPC UA

8.2.2.2 Extract_signal.py

This is an example of a not so flexible implementation as the row numbers are fixed; however, this is due to the size of the original excel file and the difficulty to parse perfectly.

```
import pandas as pd

# Step 1: Read the Excel file into a DataFrame
file_path = "Lookup.xlsx"
df = pd.read_excel(file_path)

# Step 2: Extract the values from columns "IN", "OPC", and "BIT" for rows 94 to
119
in_values = df.loc[93:120, 'IN'].dropna().tolist()
opc_values = df.loc[93:120, 'OPC'].dropna().tolist()
bit_values = df.loc[93:120, 'BIT'].dropna().tolist()

# Step 3: Ensure all lists have the same length
# If OPC and BIT columns have NaN values, this will align them with 'IN' values
max_length = max(len(in_values), len(opc_values), len(bit_values))
in_values += [''] * (max_length - len(in_values))  # Fill missing values with
empty strings
opc_values += [''] * (max_length - len(opc_values))
bit_values += [''] * (max_length - len(bit_values))

# Step 4: Replace backslashes "\\" with periods "." in the "IN" column values
in_values = [str(value).replace("\\", ".") for value in in_values]

# Step 5: Format the values, including OPC and BIT in the string
formatted_values = [
    f"'{in_value}': (0, 'real', '{opc_value}', '{bit_value}')"
    for in_value, opc_value, bit_value in zip(in_values, opc_values, bit_values)
]

# Step 6: Write the formatted values to the "signal_name.txt" file
with open("signal_name.txt", "w") as file:
    for i, item in enumerate(formatted_values):
        if i == len(formatted_values) - 1:  # If it's the last element
            file.write(item + '\n')  # Write without comma
        else:
            file.write(item + ',\n')  # Write with comma after each item

print("Data has been written to signal_name.txt.")
```

8.2.2.3 Signal_name.txt

```
'stVal_gcbINFO_XGGIO110_Ind2': (0, 'real', '1.0', '2.0'),
'stVal_gcbINFO_XGGIO110_Ind3': (0, 'real', '1.0', '3.0'),
'stVal_gcbINFO_XGGIO110_Ind4': (0, 'real', '1.0', '4.0'),
'stVal_gcbINFO_XGGIO110_Ind5': (0, 'real', '1.0', '5.0'),
'stVal_gcbINFO_XGGIO110_Ind6': (0, 'real', '1.0', '6.0'),
'stVal_gcbINFO_XGGIO110_Ind7': (0, 'real', '1.0', '7.0'),
'stVal_gcbINFO_XGGIO110_Ind8': (0, 'real', '1.0', '8.0'),
'stVal_gcbINFO_XBGGIO115_Ind1': (0, 'real', '1.0', '9.0'),
'stVal_gcbINFO_XBGGIO115_Ind2': (0, 'real', '2.0', '10.0'),
'stVal_gcbINFO_XBGGIO115_Ind3': (0, 'real', '2.0', '11.0'),
'stVal_gcbINFO_XBGGIO115_Ind4': (0, 'real', '2.0', '12.0'),
'stVal_gcbINFO_XBGGIO115_Ind5': (0, 'real', '2.0', '13.0'),
'stVal_gcbINFO_XBGGIO115_Ind6': (0, 'real', '2.0', '14.0'),
'stVal_gcbINFO_XBGGIO115_Ind7': (0, 'real', '2.0', '15.0'),
'stVal_gcbINFO_XBGGIO115_Ind8': (0, 'real', '2.0', '16.0'),
'stVal_gcbINFO_MVGAPC1_SPCSO1': (0, 'real', '2.0', '17.0'),
'stVal_gcbINFO_MVGAPC1_SPCSO2': (0, 'real', '2.0', '18.0'),
'stVal_gcbINFO_MVGAPC1_SPCSO3': (0, 'real', '2.0', '19.0'),
'stVal_gcbINFO_MVGAPC1_SPCSO4': (0, 'real', '2.0', '20.0'),
'stVal_gcbINFO_MVGAPC1_SPCSO5': (0, 'real', '1.0', '21.0'),
'stVal_gcbINFO_MVGAPC1_SPCSO6': (0, 'real', '1.0', '22.0'),
'stVal_gcbINFO_MVGAPC1_SPCSO7': (0, 'real', '2.0', '23.0'),
'stVal_gcbINFO_MVGAPC1_SPCSO8': (0, 'real', '1.0', '24.0'),
'stVal_gcbINFO_TCSSCBR1_CircAlm': (0, 'real', '2.0', '25.0'),
'stVal_gcbINFO_XBGGIO120_Ind3': (0, 'real', '1.0', '26.0'),
```

8.3 RECEIVE_TRIP.PY

This script is included to show another approach where the full circle does not need to be completed. This means no OPC UA is required to operate. Instead of the bit being scanned, the script checks for a change in the GOOSE signal for the specific position (q_J01_TRPPTRC1) in a time interval, a step.

8.3.1 SCRIPT

```
import pytest
import typhoon.api.hil as hil
import typhoon.test.signals as sig
import typhoon.test.capture as cap

#Capture rate
fs=100e3

#Analysis duration (s)
duration=0.1

#Analysis strictness (less than 1)
strictness = 0.9

#Voltage to trip
voltage = 20e3

@pytest.fixture(scope="module")
def setup():
    hil.load_model(file='single_bay_daniel Target files\\single_bay_daniel.cpd',
                    vhil_device=False,
                   )
    hil.start_simulation()

    hil.set_scada_input_value('DC 1 close', 1.0)
    hil.set_scada_input_value('DC 1 open', 0.0)

    hil.set_scada_input_value('ES 1 close', 0.0)
    hil.set_scada_input_value('ES 1 open', 1.0)

    hil.set_scada_input_value('CB 1 close', 0.0)
    hil.set_scada_input_value('CB 1 open', 1.0)

def test_trip(setup):

    #Close CB
    hil.set_scada_input_value('CB 1 close', 1.0)
    hil.set_scada_input_value('CB 1 open', 0.0)
```

```
    #Close DC
    hil.set_scada_input_value('DC 1 close', 1.0)
    hil.set_scada_input_value('DC 1 open', 0.0)

    #set new voltage
    hil.set_source_sine_waveform(name='Vs1',
                                  rms=voltage,
                                 frequency=50,
                                 phase=0,
                                  )

    cap.start_capture(duration=duration,
                      rate=fs,
                      signals=['q_J01_TRPPTRC1']
                      )

    sig_data = cap.get_capture_results(wait_capture=True)


    with pytest.assume:
        sig.assert_is_step(signal='q_J01_TRPPTRC1',
                           from_value=0,
                           to_value=1,
                           at_t=(0,duration-1/fs),
                           strictness=strictness
                           )
```

Chapter 9. PROJECT ANALYSIS

Having discussed all the modelling, SCADA testing and FAT automation, an analysis must be made to see if the project is viable in the long run and other aspects such as when will it break even. The most defining factor is how much time is saved per FAT, bay or project as a whole.

The conclusion, reported by the internal engineering team logging, was that the project FAT part was sped up 60%. It decreased from 2 ½ weeks to 1 week.

As a result, the extra revenue per year (or the cost which is not put into a project) will represent how much percent of the workload has been automated. This will be deemed the project's final number estimator for traditional economic analysis metrics such as the Internal Rate of Return (IRR) or the Net Present Value (NPV).

9.1 ASSUMPTIONS

Several assumptions have been made:

1. FAT expedited time remains constant and will not increase
   This is a pessimistic approach, but it is chosen as it is the most conservative metric which can be employed.

2. Exclusion of Transport-Related Costs:
   Costs associated with the transport of IEDs have been excluded. This includes both:
   a. Inbound logistics: movement of IEDs from the manufacturer to the programming facility.
   b. Outbound logistics: dispatch to substations or return transport for post-SAT reprogramming.

   These elements are excluded due to their variability, dependency on third-party logistics, and minimal influence on long-term break-even evaluations.

3. Wage Rates Assumed as Fixed:
   Hourly wage rates for interns and engineers have been set at €15/h and €50/h, respectively. It is assumed that these rates remain constant over the assessment period, without inflationary correction or regional adjustments.

4. Static Project Volume and Scope:
   The number of projects per year and average bays per project have been held constant. This does not account for expansion or contraction in project volume, market demand variability, or seasonal effects. The analysis therefore presumes a steady workload throughput, in line with the recent five-year average.

5. Revenue Derived Solely from Time Savings:
   The calculated extra revenue is derived exclusively from the differential in labour hours saved per tested bay multiplied by the engineer hourly wage and corrected for project variation and adaptability. No additional operational improvements (e.g., error rate reduction, documentation speed-up) have been monetised.

6. Time Horizon and Discount Rate:
   The analysis adopts a 10-year project horizon and a fixed discount rate of 5% for NPV calculation. Alternative time frames or sensitivity to discount rates have not been explored.

9.2 FORMULATION

9.2.1 VARIABLES

$$h_t = \text{Hours per Tested Bay (Traditional)} \quad [h/bay]$$

$$h_a = \text{Hours per Tested Bay (Automated)} \quad [h/bay]$$

$$b_t = \text{Total Tested Bays} \quad [\,bay]$$

$$t_i = \text{FAT Automation Development Hours (intern)} \quad [h]$$

$$t_e = \text{FAT Automation Development Hours (engineer)} \quad [h]$$

$$c_{\text{hw}} = \text{Hardware Costs} \quad [€]$$

$$c_{\text{eng}} = \text{Support Engineering Costs} \quad [€]$$

$$l_i = \text{Intern Wage} \quad [€/h]$$

$$l_e = \text{Engineer Wage} \quad [€/h]$$

$$m = \text{Maintenance Fee per Year} \quad [€/year]$$

$$p = \text{Average Projects per Year (last 5)} \quad [\,project/year]$$

$$b_{\text{avg}} = \text{Average Bays per Project} \quad [\,bay/project]$$

$$\eta_X = Adaptability\ in\ Scenario\ X$$

$$\delta_Y = Project\ Variation\ in\ Scenario\ Y$$

9.2.2 CALCULATED VARIABLES

$$L = \text{Development Wages } [€]$$

$$C = \text{Project Total Cost } [€]$$

$$B = \text{Bay per Year } [\text{bay/year}]$$

$$R = \text{Extra Revenue per year } [€/year]$$

$$L = t_i \cdot l_i + t_e \cdot l_e$$

$$C = \mathrm{L} + \mathrm{c_{hw}} + \mathrm{c_{eng}}$$

$$B = p \cdot b_{avg}$$

$$R = (h_t - \mathrm{h_a}) \cdot (1 + \delta_\mathrm{Y}) \cdot \eta_\mathrm{X} \cdot \mathrm{l_e} - m$$

9.3 CALCULATIONS

9.3.1 INPUT VALUES

Variable	Value	Unit
h_t	10	h
h_a	4	h
b_t	10	-
t_i	180	h
t_e	56	h
c_{hw}	61180	€
c_{eng}	14000	€
l_i	15	€/h
l_e	50	€/h
m	6090	€
p	8	-
b_{avg}	12	-

9.3.2 OUTPUT VARIABLES

9.3.2.1 Extra Revenue

Extra Revenue / year					
		Project Variation			
		10%	5%	-5%	-10%
Adaptability	95%	€ 24,006.00	€ 22,638.00	€ 19,902.00	€ 18,534.00
	90%	€ 22,422.00	€ 21,126.00	€ 18,534.00	€ 17,238.00
	85%	€ 20,838.00	€ 19,614.00	€ 17,166.00	€ 15,942.00
	80%	€ 19,254.00	€ 18,102.00	€ 15,798.00	€ 14,646.00

Table 5. Extra Revenue per Year

Calculating then the IRR and NPV at 5%, for 10 years:

9.3.2.2 IRR

IRR					
		Project Variation			
		10%	5%	-5%	-10%
Adaptability	95%	27%	25%	21%	19%
	90%	25%	23%	19%	17%
	85%	23%	21%	17%	15%
	80%	20%	18%	15%	13%

Table 6. IRR at 10 years

9.3.2.3 NPV (5%)

NPV					
5%		Project Variation			
		10%	5%	-5%	-10%
Adaptability	95%	€ 159,680.00	€ 146,000.00	€ 118,640.00	€ 104,960.00
	90%	€ 143,840.00	€ 130,880.00	€ 104,960.00	€ 92,000.00
	85%	€ 128,000.00	€ 115,760.00	€ 91,280.00	€ 79,040.00
	80%	€ 112,160.00	€ 100,640.00	€ 77,600.00	€ 66,080.00

Table 7. NPV at 10 years with 5%

9.4 RESULT ANALYSIS

The four highlighted central cells in each matrix represent the most probable scenarios, corresponding to mid-range assumptions of ±5% project variation and 90% or 85% adaptability. These combinations are the most realistic and therefore should be given greater analytical weight.

In the IRR matrix, the central cases yield values between 17% and 23%, which remain significantly above typical project rates, indicating robust investment viability. The NPV values for these same scenarios range from €91,280 to €130,880, supporting a favourable financial outlook even with modest project variations. Similarly, the extra annual revenue for these adaptability levels fluctuates between €17,166 and €21,126, suggesting that operational revenue benefits are maintained under plausible uncertainty margins.

Extreme cases (±10% variation and 95% or 80% adaptability) exhibit wider financial dispersion, yet the centre-weighted outcomes support the project's overall resilience to moderate deviations in performance and assumptions.

Chapter 10. CONCLUSIONS AND FUTURE WORKS

The present work has demonstrated the technical viability and functional relevance of modelling the REX615 IED and how virtualising them is proven to be beneficial. The current observed time reduction of approximately 60% is very promising and with further improvements in automation, test coverage, and configuration management it is expected to increase this acceleration up to 80%. In addition, the methodology employed ensured reproducibility, ease of debugging, and enhanced scalability for future deployments guaranteeing positive results for the projects and years to come.

From an economic perspective, the proposed approach offers a low-cost yet effective alternative to hardware-based testing setups, which are typically expensive, inflexible, and require significant maintenance. By leveraging open-source tools and modular architectures, the solution supports iterative development cycles and continuous integration, significantly reducing development and verification overheads. This translates into considerable savings in both time and resources for substation automation projects.

Looking ahead, the framework established in this thesis presents a compelling foundation for further research and practical extensions. A fully virtualised and modelled substation remains a highly promising objective. By expanding the current models to incorporate a wider array of IEDs, it would be possible to simulate entire substations with high fidelity. This could greatly enhance training capabilities, expedite functional testing during development phases, and support real-time decision-making processes in control centres.

Furthermore, the modelling of manufacturer-specific IEDs presents a valuable opportunity for collaboration. Partnerships with equipment manufacturers would enable the integration of proprietary communication protocols, configuration schemas, and failure modes into the simulation environment. Such collaboration could foster the development of standardised test suites and digital twins that align more closely with actual field conditions.

Nevertheless, the virtualization approach offers an opportunity. Validation and testing can occur before any physical manufacturing takes place. This brings the advantage of decoupling engineering from manufacturing which enables the development to proceed in parallel or even ahead of the factory schedule. Benefits entailed include significantly shorter project lead times, faster time-to-market, flexible and agile engineering cycles as well as reduced project risk and cost.

All in all, virtualization and automatic testing brings net positive results to all parties involved and it should be pursued and options to further improve must be explored.

Chapter 11. REFERENCES

[1] O. Caio and D. Santo, "What is Controller Hardware-In-The-Loop Simulation, or C-HIL?," Typhoon HIL, 19 September 2024. [Online]. Available: https://www.typhoon-hil.com/blog/what-is-controller-hardware-in-the-loop-c-hil-simulation/. [Accessed July 2025].

[2] H. Chang, "Power Electronics Control Design and Testing in the 21st Century," Typhoon HIL, 10 May 2016. [Online]. Available: https://www.typhoon-hil.com/blog/power-electronics-control-design-and-testing-in-the-21st-century/. [Accessed July 2025].

[3] "Pytest Docuumentation," [Online]. Available: https://docs.pytest.org/en/latest/. [Accessed July 2025].

[4] "Allure Pytest," [Online]. Available: https://allurereport.org/docs/pytest/. [Accessed July 2025].

[5] Typhoon HIL, "Quick Start: Typhoon HIL Control Center," Typhoon HIL, [Online]. Available: https://www.typhoon-hil.com/documentation/typhoon-hil-software-manual/concepts/realtime_quick_start_guide.html?hl=control. [Accessed July 2025].

[6] W. Schossig, "History Protection - Generations of Protection," *PAC World Magazine,* no. September, 2012.

[7] B. S. Tatera and H. L. Smith, "The evolution of monitoring and controlling in electric power substations," in *IEEE Power and Energy Society General Meeting - Conversion and Delivery of Electrical Energy in the 21st Century*, Pittsburgh, 2008.

[8] S.-A. L. Hans-Erik Olovsson, “Substation design in the 1900s and,” ABB Review, 2008.

[9] IEEE PES PSRC K15 Working Group, “Report of Working Group K15 of the Substation Protection Subcommittee,” *PAC World Magazine,* no. December, 2019.

[10] ABB, “SSC600 and SSC600 SW Technical Manual,” ABB, 13 December 2024. [Online]. Available: https://techdoc.relays.protection-control.abb/r/SSC600-and-SSC600-SW-Technical-Manual/1.5/en-US/Introduction. [Accessed July 2025].

[11] ABB, “REX615 Technical Manual,” ABB, 20 August 2024. [Online]. Available: https://techdoc.relays.protection-control.abb/r/REX615-Technical-Manual/PCL1/en-US/REX615-Technical-Manual. [Accessed July 2025].

[12] ABB, “PCM600 - Protection and control IED manager,” ABB, [Online]. Available: https://new.abb.com/medium-voltage/digital-substations/software-products/protection-and-control-ied-manager-pcm600. [Accessed July 2025].

[13] “Wireshark Documentation,” [Online]. Available: https://www.wireshark.org/docs/. [Accessed July 2025].

[14] Unified Automation, “ANSI C Based OPC UA Client & Server SDK,” [Online]. Available: https://www.unified-automation.com/products/server-sdk/ansi-c-ua-server-sdk.html. [Accessed July 2025].

[15] Hirschmann, “GRS1142,” [Online]. Available: https://www.doc.hirschmann.com/pdf/IG_GRS_1042-1142_05-0218_en_2018-02-13.pdf. [Accessed July 2025].

[16] OMICRON, “CMC 356,” OMICRON, [Online]. Available: https://www.omicronenergy.com/en/products/cmc-356/. [Accessed July 2025].

[17] Ravikumar, Krishnanjan; G. Schulz, Noel N. and Srivastava, Anurag K., "Distributed simulation of power systems using real-time digital simulator," in *IEEE/PES Power Systems Conference and Exposition*, Seattle, 2009.

[18] Typhoon HIL, "HIL 606," Typhoon HIL, [Online]. Available: https://www.typhoon-hil.com/products/hil-simulator/hil606/. [Accessed July 2025].

[19] M. Hemmati, H. Palahalli, G. Gruosso and S. Grillo, "Interoperability analysis of IEC61850 protocol using an emulated IED in a HIL microgrid testbed," in *IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, Aachen, 2021.

[20] S. Simic, "https://www.typhoon-hil.com/blog/testing-digital-substations-advancements-with-hardware-in-the-loop-technology/," Typhoon HIL, 4 September 2024. [Online]. Available: https://www.typhoon-hil.com/blog/testing-digital-substations-advancements-with-hardware-in-the-loop-technology/. [Accessed July 2025].

[21] ABB, "Smart substation control and protection SSC600 SW," ABB, [Online]. Available: https://new.abb.com/medium-voltage/digital-substations/protection-relays/multiapplication/smart-substation-control-and-protection-ssc600-sw. [Accessed July 2025].

[22] "Get Ubuntu Server," [Online]. Available: https://ubuntu.com/download/server. [Accessed July 2025].

[23] A. Frade, "XanMod Kernel," 2015. [Online]. Available: https://xanmod.org/. [Accessed July 2025].

ANNEX I – SUSTAINABLE DEVELOPMENT GOALS

This project can be associated with several of the United Nations Sustainable Development Goals (SDG), particularly those focused on energy, innovation, and efficiency in industrial practice.

Firstly, the project aligns with SDG 7: Affordable and Clean Energy. By introducing a digital testing framework for IEDs and SCADA platforms, it reduces reliance on physical hardware during FAT procedures. This minimises resource consumption, energy use, and the environmental footprint associated with hardware transport and setup. Moreover, by ensuring more reliable and efficient automation systems, the project contributes to the stability of electrical infrastructures, which is essential for secure access to clean energy.

Secondly, it supports SDG 9: Industry, Innovation and Infrastructure. The digitalisation of testing processes represents an innovative practice that modernises industrial procedures and is one of the first projects to try to fully automatise testing. HIL simulation facilitates flexible, scalable, and reproducible validation methods that improve resilience and efficiency of industrial infrastructures.

Thirdly and lastly, the project contributes to SDG 12: Responsible Consumption and Production. The reduction in physical prototypes, transportation, and repetitive manual testing activities directly decreases material waste and operational inefficiencies. This promotes a more responsible use of resources in the energy and automation sector, while also enabling circular practices through software-based replication of hardware systems.

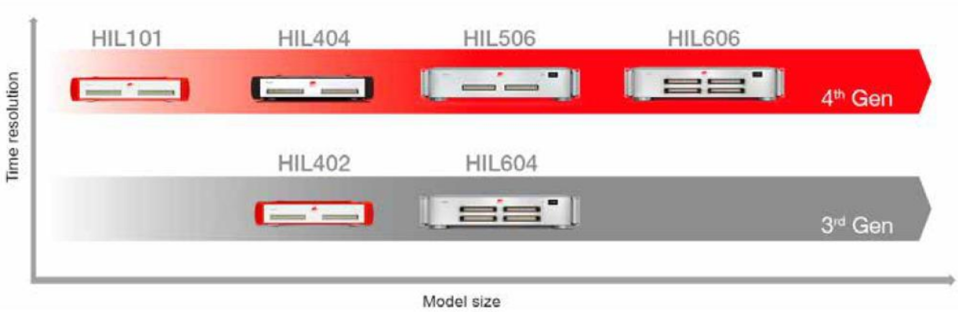
ANNEX II - HIL 606 [2] [3]

[2] https://www.typhoon-hil.com/wp-content/uploads/2025/07/Brochure_A4-HIL606-Digital.pdf

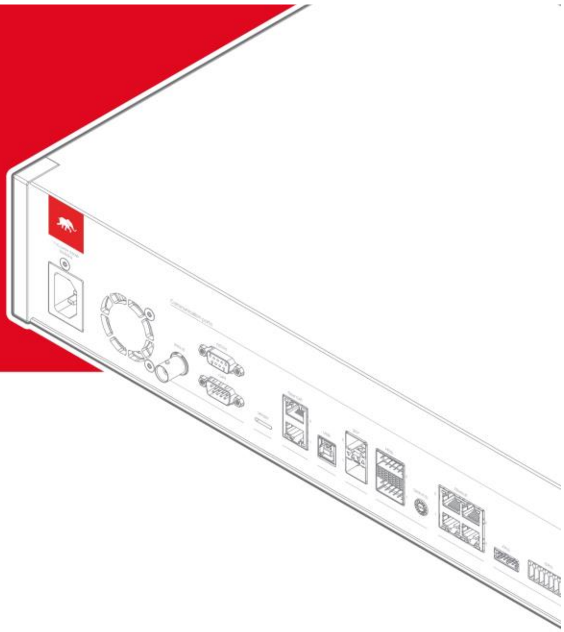
[3] For further information, the webpage https://www.typhoon-hil.com/solutions/grid-modernization/distribution-automation/substation-protection-testing/ provides a comprehensive insight into further IEC61850 and IEC61869 testing.

HIL device comparison.

	HIL101	HIL404	HIL506	HIL606
Model capacity				
Detailed (switching) converter models (1ph / 3ph)	4 / 2	8 / 4	12 / 6	16 / 8
Average converter models (3ph)	8	12	18	24
Machine Solvers upported (max.)	1	2+	4	4
Distribution network simulation	✓	✓	✓	✓
Time resolution				
Minimal simulation step	250 ns	200 ns	200 ns	200 ns
DI sampling resolution	4.5 ns	3.5 ns	3.5 ns	3.5 ns
IO				
Analog I/O per unit	16 / 16	16 / 16	16 / 32	32 / 64
Digital I/O per unit	32 / 32	32 / 32	32 / 32	64 / 64
Connectivity				
Ethernet, USB 3.0, CAN, RS232, GPIO, HSSL	✓	✓	✓	✓
JTAG	✓	✓	✓	✓
Time synchronization (PPS and IRIG-B)			✓	✓
EtherCAT (master and slave)			✓	✓
CAN FD			✓	✓
SFP, QSFP			✓	✓
Paralleling	up to 4 units (HIL101)	up to 4 units (HIL404)	up to 16 units (HIL506 and HIL606)	up to 16 units (HIL506 and HIL606)



HIL606 technical details.



For more technical details read
Typhoon HIL Hardware documentation.

Processor	Processor	ZU9EG Zynq UltraScale+ MPSoC
	Processor configurations	up to 8 processing cores
Analog inputs (AI)	Channels	32
	Resolution	16 bit ADC
	Input Voltage Range:	±10 V
	Sample Rate	up to 1 MSPS
	Linearity (DNL/INL)	1/2
	Gain error / Offset error	0.01% / 1 mV
	Input Resistance	25.5 kΩ
	Protection	±24 V tolerant, ESD protection
Analog Outputs (AO)	Channels	64
	Resolution	16 bit ADC
	Output Voltage Range	±10 V
	Sample Rate	up to 5 MSPS
	Linearity (DNL/INL)	1/1
	Gain error / Offset error	0.01 % / 1 mV
	Output Resistance	~0 Ω
	Protection	±24 V tolerant, ESD protection
Analog IO connector	Connector	DIN 41612, type C 96 pin male connector

13

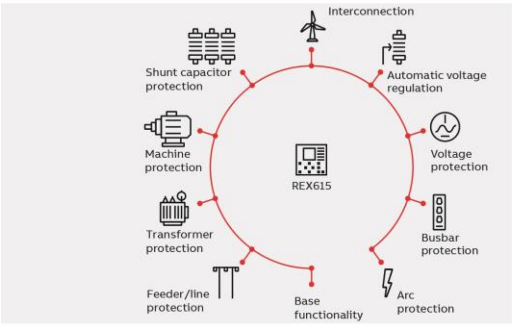
User Power Supply Stage (PSU)	±5 V analog	up to 2 A, resettable protection
	±12 V analog	up to 2 A, resettable protection
	+3.3 V digital	up to 2 A, resettable protection
	+5 V digital	up to 2 A, resettable protection
Digital inputs (DI)	Channels	64 channels
	Input voltage range Vi	-15 V < V_i < 15 V
	Threshold voltages (low, high)	(V_{IL}(max) = 0.8 V; V_{IH}(min) = 2 V)
	Input resistance	10 kΩ
	Protection	±24 V tolerant, ESD protection
	DI sampling resolution	3.5 ns
Digital outputs (DO)	Channels	64 channels
	Output voltage range Vo	5 V
	Output voltages (low, high)	(V_{OL}(max) = 0.2 V; V_{OH}(min) = 4.8 V)
	Output resistance	430 Ω
	Protection	±24 V tolerant, ESD protection
Digital IO connector	Connector type	DIN 41612, type C 96 pin male connector
Connectivity	Ethernet	4x RJ45 connectors; 10/100/1000 Mbps
	USB 3.0	1 x type B connector
	CAN	2 x DE9 male Connector
	FDCAN	2 x DE9 male Connector
	RS232	DE9 Female Connector
	High speed serial link	
	Quad SFP	
	JTAG	Molex 87833-1420
	GPIO	12+ multi-purpose IO pins, terminal blocks
	EtherCAT	2 x
	Time synchronization (IRIG-B)	1 x
Housing	Dimensions	19" rack mountable; 2U height
	Weight	up to 10 kg
Power supply	Input voltage	100 - 250 VAC
	Power consumption	up to 250 W

ANNEX III - REX615[4]

ABB

Protection and control REX615

All-in-one protection for power generation and distribution applications



REX615 is a freely configurable all-in-one protection relay for power generation and distribution applications, representing the next step for ABB's 615 and 620 series relays. Multiapplication coverage combined with a fully modular and scalable hardware and software ensures maximum flexibility and optimal cost-effectivity throughout the relay life cycle – and supports sustainability.

— 01 Supported applications

Addressing aging infrastructure by future-proofing the grid

- Ideal for one-to-one replacement of 615 and 620 series relays as the next step in terms of technological innovation, flexibility, cost-effectiveness and standardization
- Easy customization and adaptation to changing protection requirements with modular and scalable hardware and software
- Extensive functionality to support the increasing integration of distributed generation and evolving grid codes
- Condition monitoring and supervision for optimized component maintenance and return on investment
- Relay retrofit program for smooth and easy replacement of SPACOM with REX615 relays

Supporting the digitalization of substations

- Designed to support the increasing digitalization of substations – with IEC 61850 9-2-based process bus for future integration with centralized protection
- Standardized IEC 61850-compliant communication – a cost-effective and ecological choice
- Smooth interoperability between substation automation devices with the world's first truly IEC 61850-based protection relay
- Optimal cyber security to safeguard critical infrastructure

Optimizing asset management

- Brand-new relay but with an impressive history – familiar relay saves time and efforts
- Less to learn and less spare devices to store with one relay covering multiple application areas
- Extensive life cycle services – long-term global partner with longtime experience close to you in your language
- Adaptable to changing requirements for the lifetime of the relay through easy adding or replacing of modules or application packages
- Easy access to up-to-date information via the e-business platform ABB Relays-Online
- Latest addition to ABB's renown Relion® relay family – strengthening the REX portfolio

Complying with evolving regulations

- Interconnection protection package for a better way of connecting to renewables
- Compliance with the latest grid codes to ensure renewables remain connected during disturbances
- Extensive protection and fault location functionality to ensure compliance with the regulation of electricity utilities to minimize costly downtime
- Third-party verified Environmental Product Declaration compliant with ISO 14025

4 https://search.abb.com/library/Download.aspx?DocumentID=2NGA001876&LanguageCode=en&DocumentPartId=&Action=Launch

Function description	IEC 60617	ANSI	IEC 61850	Instances	Application package
Protection					
Line differential protection with inzone power transformer	3Id/I>	87L	LNPLDF	1	APP6
Binary signal transfer	BST	BST	BSTGAPC	1	APP6
Switch-onto-fault protection	CVPSOF	SOTF	CVPSOF	1	APP1
Three-phase non-directional overcurrent protection, low stage	3I>	51P-1	PHLPTOC	3	APP1
Three-phase non-directional overcurrent protection, high stage	3I>>	51P-2	PHHPTOC	3	APP1
Three-phase non-directional overcurrent protection, instantaneous or definite time stage	3I>>>	50P/51P	PHIPTOC	3	APP1
Three-phase non-directional long time overcurrent protection	3I>	51LT	PHLTPTOC	1	APP1
Three-phase non-directional overcurrent protection, instantaneous only stage	3I>>>>	50P	PHIPIOC	3	APP1
Three-independent-phase non-directional overcurrent protection, low stage	3I_3>	51P_3-1	PH3LPTOC	2	APP1
Three-independent-phase non-directional overcurrent protection, high stage	3I_3>>	51P_3-2	PH3HPTOC	2	APP1
Three-independent-phase non-directional overcurrent protection, instantaneous or definite time stage	3I_3>>>	50P_3/51P_3	PH3IPTOC	2	APP1
Three-phase directional overcurrent protection, low stage	3I> ->	67P/51P-1	DPHLPDOC	2	APP3, APP6, APP8, ADD1, APP10
Three-phase directional overcurrent protection, high stage	3I>> ->	67P/51P-2	DPHHPDOC	2	APP3, APP6, APP8, ADD1, APP10
Directional three-independent-phase directional overcurrent protection, low stage	3I_3> ->	67P_3/51P_3-1	DPH3LPDOC	2	APP3, APP6, APP8, ADD1, APP10
Directional three-independent-phase directional overcurrent protection, high stage	3I_3>> ->	67P_3/51P_3-2	DPH3HPDOC	2	APP3, APP6, APP8, ADD1, APP10
Non-directional earth-fault protection, low stage	Io>	51G/51N-1	EFLPTOC	3	APP1
Non-directional earth-fault protection, high stage	Io>>	51G/51N-2	EFHPTOC	3	APP1
Non-directional earth-fault protection, instantaneous stage	Io>>>	50G/N, 51G/N	EFIPTOC	3	APP1
Non-directional earth-fault protection, instantaneous only stage	Io>>>>	50G/50N	EFIPIOC	3	APP1
Directional earth-fault protection, low stage	Io> ->	67G/N-1 51G/N-1	DEFLPDEF	3	APP1
Directional earth-fault protection, high stage	Io>> ->	67G/N-1 51G/N-2	DEFHPDEF	2	APP1
Three-phase power directional element	I1 ->	67P-TC	DPSRDIR	2	APP3, APP8, ADD1, APP10, APP12
Neutral power directional element	I2 ->, Io ->	67N-TC	DNZSRDIR	2	APP2, APP3
Admittance-based earth-fault protection	Yo> ->	21NY	EFPADM	3	APP2
Multifrequency admittance-based earth-fault protection	Io> -> Y	67NYH	MFADPSDE	3	APP2
Wattmetric-based earth-fault protection	Po> ->	32N	WPWDE	3	APP2
Transient/intermittent earth-fault protection	Io> -> IEF	67NTEF/NIEF	INTRPTEF	1	APP2
Harmonics-based earth-fault protection	Io>HA	51NH	HAEFPTOC	1	APP2
Touch voltage based earth-fault current protection	IF>/UT>	46SNQ/59N	IFPTOC	3	APP2
Negative-sequence overcurrent protection	I2>	46	NSPTOC	3	APP1
Directional negative-sequence overcurrent protection	I2> ->	67Q	DNSPDOC	2	APP3, APP6, APP10
Phase discontinuity protection	I2/I1>	46PD	PDNSPTOC	1	APP1
Residual overvoltage protection	Uo>	59G/59N	ROVPTOV	4	APP1, APP5
Three-phase undervoltage protection	3U<	27	PHPTUV	4	APP5
Single-phase undervoltage protection	U_A<	27_A	PHAPTUV	1	APP5
Three-phase overvoltage variation protection	3Urms>	59.S1	PHVPTOV	2	APP5, APP8
Three-phase overvoltage protection	3U>	59	PHPTOV	4	APP5
Single-phase overvoltage protection	U_A>	59_A	PHAPTOV	1	APP5
Positive-sequence overvoltage protection	U1>	59PS	PSPTOV	2	APP5
Positive-sequence undervoltage protection	U1<	27PS	PSPTUV	2	APP5
Negative-sequence overvoltage protection	U2>	59NS	NSPTOV	4	APP5
Frequency protection	f>/f<, df/dt	81	FRPFRQ	10	APP5
Three-phase voltage-dependent overcurrent protection	3I(U)>	51V	PHPVOC	2	APP3, APP8, ADD1, APP10
Overexcitation protection	U/f>	24	OEPVPH	2	ADD1, APP10

Function description	IEC 60617	ANSI	IEC 61850	Instances	Application package
Three-phase thermal protection for feeders, cables and distribution transformers	3Ith>F	49F	T1PTTR	1	APP1
Three-phase thermal overload protection, two time constants	3Ith>T/G/C	49T/G/C	T2PTTR	1	APP1
Three-phase overload protection for shunt capacitor banks	3I> 3I<	51, 37, 86C	COLPTOC	1	APP7
Current unbalance protection for shunt capacitor banks	dI>C	60N	CUBPTOC	1	APP7
Three-phase current unbalance protection for shunt capacitor banks	3dI>C	60P	HCUBPTOC	1	APP7
Shunt capacitor bank switching resonance protection, current based	TD>	55ITHD	SRCPTOC	1	APP7
Compensated neutral unbalance voltage protection	CNU>	59NU	CNUPTOV	1	APP7
Low-voltage ride-through protection	U<RT	27RT	LVRTPTUV	3	APP8
Voltage vector shift protection	VS	78VS	VVSPPAM	2	APP8
Directional reactive power undervoltage protection	Q> ->, 3U<	32Q, 27	DQPTUV	2	APP8
Reverse power/directional overpower protection	P>/Q>	32R/32O	DOPPDPR	3	APP3, APP8, ADD1, ADD2, APP10
Underpower protection	P<	32U	DUPPDPR	2	APP3, ADD1,ADD2, APP10
Three-phase underimpedance protection	Z<G	21G	UZPDIS	3	ADD1, APP10
Directional negative sequence impedance protection	Z2 ->	Z2Q	DNZPDIS	2	APP3, APP6, APP10
Three-phase underexcitation protection	X<	40	UEXPDIS	2	ADD1
Third harmonic-based stator earth-fault protection	dUo>/Uo3H	64TN	H3EFPSEF	1	ADD1
Rotor earth-fault protection (injection method)	Io>R	64R	MREFPTOC	1	ADD1
Thermal overload protection for rotors	3Ith>R	49R	RPTTR	1	ADD1,ADD2
High-impedance or flux-balance based differential protection	3dIHi>M	87HIM	MHZPDIF	1	ADD1, ADD2
Out-of-step protection with double blinders	OOS	78PS	OOSRPSB	1	ADD1
Negative-sequence overcurrent protection for machines	I2>M	46M	MNSPTOC	2	APP9
Loss of phase (undercurrent)	3I<	37	PHPTUC	3	APP1
Loss of load supervision	3I<	37	LOFLPTUC	2	APP9
Motor load jam protection	Ist>	50TDJAM	JAMPTOC	2	APP9
Motor start-up supervision	Is2t n<	49, 66, 48, 50TDLR	STTPMSU	1	APP9
Motor start counter	n<	66	MSCPMRI	1	APP9
Phase reversal protection	I2>>	46R	PREVPTOC	1	APP9
Thermal overload protection for motors	3Ith>M	49M	MPTTR	1	APP9
Stabilized and instantaneous differential protection for machines	3dI>M/G	87M/87G	MPDIF	1	ADD1, ADD2
Underpower factor protection	PF<	55U	MPUPF	2	APP3, APP8, ADD1, APP10
Stabilized and instantaneous differential protection for two-winding transformers	3dI>T	87T	TR2PTDF	1	APP10
Numerical stabilized low-impedance restricted earth-fault protection	dIoLo>	87NLI	LREFPNDF	2	APP1
High-impedance based restricted earth-fault protection	dIoHi>	87NHI	HREFPDIF	2	APP1
High-impedance differential protection for phase A	dHi_A>	87_A	HIAPDIF	2	ADD1,ADD2, APP10, APP11
High-impedance differential protection for phase B	dHi_B>	87_B	HIBPDIF	2	ADD1, ADD2, APP10, APP11
High-impedance differential protection for phase C	dHi_C>	87_C	HICPDIF	2	ADD1, ADD2, APP10, APP11
Circuit breaker failure protection	3I>/Io>BF	50BF	CCBRBRF	3	APP1
Three-phase inrush detector	3I2f>	68HB	INRPHAR	2	APP1
Master trip	Master Trip	94/86	TRPPTRC	6	Base
Arc protection	ARC	AFD	ARCSARC	3	Base
High-impedance fault detection	HIF	HIZ	PHIZ	1	APP2
Fault locator	FLOC	FLOC	SCEFRFLO	1	APP4
Load-shedding and restoration	UFLS/R	81LSH	LSHDPFRQ	10	APP5
Multipurpose protection	MAP	MAP	MAPGAPC	20	Base
Accidental energization protection	U<, I>	27/50	GAEPVOC	1	ADD1
Load blinder	LB	21LB	LBRDOB	2	APP3, APP6
Cable Fault Detection	CFD	CFD	RCFD	1	APP2, APP3

Function description	IEC 60617	ANSI	IEC 61850	Instances	Application package
Control					
Circuit-breaker control	I <-> O CB	52	CBXCBR	3	Base
Disconnector control	I <-> O DCC	29DS	DCXSWI	4	Base
Earthing switch control	I <-> O ESC	29GS	ESXSWI	3	Base
Disconnector position indication	I <-> O DC	29DS	DCSXSWI	4	Base
Earthing switch position indication	I <-> O ES	29GS	ESSXSWI	3	Base
Emergency start-up	ESTART	EST, 62	ESMGAPC	1	APP9
Autoreclosing	O -> I	79	DARREC	2	APP1
Circuit breaker uncorresponding position start-up	CBUPS	52OU	UPCALH	3	Base
Synchronism and energizing check	SYNC	25	SECRSYN	2	APP5
Tap changer control with voltage regulator, legacy	COLTC	90V	OLATCC	1	APP12
Tap changer control with voltage regulator	COLTC	90V	OL5ATCC	1	APP12
Transformer data combiner	OLGAPC	OLGAPC	OLGAPC	5	APP12
Tap changer position indication	TPOSM	84T	TPOSYLTC	1	APP10, APP12
Condition Monitoring and Supervision					
Circuit-breaker condition monitoring	CBCM	52CM	SSCBR	3	Base
Motor controlled earthing switch and disconnector supervision	ESDCCM	29CM	ESDCSSWI	7	Base
Hot-spot and insulation ageing rate monitoring for transformers	3Ihp>T	26/49HS	HSARSPTR	1	APP10
Trip circuit supervision	TCS	TCM	TCSSCBR	3	Base
Current circuit supervision	MCS 3I	CCM	CCSPVC	2	Base
Current circuit supervision for transformers	MCS 3I, I2	CCM 3I, I2	CTSRCTF	1	APP10
Current circuit supervision for line differential	MCS_L 3I, I2	CCM_L 3I, I2	LNCTSRCTF	1	APP6
Current transformer supervision for high-impedance protection scheme for phase A	MCS I_A	CCM_A	HZCCASPVC	2	APP11
Current transformer supervision for high-impedance protection scheme for phase B	MCS I_B	CCM_B	HZCCBSPVC	2	APP11
Current transformer supervision for high-impedance protection scheme for phase C	MCS I_C	CCM_C	HZCCCSPVC	2	APP11
Fuse failure supervision	FUSEF	VCM, 60	SEQSPVC	2	Base
Protection communication supervision	PCS	PCS	PCSITPC	1	APP6
Runtime counter for machines and devices	OPTS	OPTM	MDSOPT	2	Base
Three-phase remanent undervoltage supervision	3U<R	27R	MSVPR	1	APP5
Voltage presence	PHSVPR	PHSVPR	PHSVPR	2	APP5
Measurement					
Three-phase current measurement	3I	IA, IB, IC	CMMXU	4	Base
Sequence current measurement	I1, I2, I0	I1, I2, I0	CSMSQI	4	Base
Residual current measurement	Io	IG	RESCMMXU	4	Base
Three-phase voltage measurement	3U	VA, VB, VC	VMMXU	4	Base
Single-phase voltage measurement	U_A	V_A	VAMMXU	2	Base
Phase voltage measurement	3UL	VL	VPHMMXU	2	Base
Residual voltage measurement	Uo	VG/VN	RESVMMXU	4	Base
Sequence voltage measurement	U1, U2, U0	V1, V2, V0	VSMSQI	4	Base
Three-phase power and energy measurement	P, E	P, E	PEMMXU	3	Base
Single-phase power and energy measurement	P_A, E_A	P_A, E_A	SPEMMXU	1	Base
Frequency measurement	f	f	FMMXU	3	Base

Function description	IEC 60617	ANSI	IEC 61850	Instances	Application package
Power Quality					
Current total demand, harmonic distortion, DC component (TDD, THD, DC) and individual harmonics	PQM3IH	PQM ITHD, IDC	CHMHAI	2	Base
Voltage total harmonic distortion, DC component (THD, DC) and individual harmonics	PQM3VH	PQM VTHD, VDC	VHMHAI	2	Base
Voltage variation	PQMU	PQMV SWE, SAG, INT	PHQVVR	2	Base
Voltage unbalance	PQUUB	PQMV UB	VSQVUB	2	Base
Programmable LED indication and push buttons					
Individual programmable LED control	LED	LED	LED	11	Base
Standard: Programmable buttons (4 buttons)	FKEY	FKEY	FKEY4GGIO	1	Base
Wide: Programmable buttons (16 buttons)	FKEY	FKEY	FKEYGGIO	1	Base
Logging functions					
Disturbance recorder (common functionality)	DR	DFR	RDRE	1	Base
Disturbance recorder, analog channels 1...12	A1RADR	A1RADR	A1RADR	1	Base
Disturbance recorder, binary channels 1...32	B1RBDR	B1RBDR	B1RBDR	1	Base
Disturbance recorder, binary channels 33...64	B2RBDR	B2RBDR	B2RBDR	1	Base
Fault recorder	FAULTREC	FR	FLTRFRC	1	Base
Sequence event recorder	SER	SER	SER	1	Base
Load profile recorder	LOADPROF	LOADPROF	LDPRLRC	1	Base
Communication					
IEC 61850-1 MMS	MMS	MMS	MMSLPRT	1	
IEC 61850-1 GOOSE	GSE	GSE	GSELPRT	1	
SMV stream receiver (IEC 61850-9-2LE)	SMVRCV	SMVRCV	SMVRCV	2	
SMV stream sender (IEC 61850-9-2LE)	SMVSENDER	SMVSENDER	SMVSENDER	1	
IEC 60870-5-103 protocol	I3C	I3C	I3CLPRT	2	
IEC 60870-5-104 protocol	I5C	I5C	I5CLPRT	5	
DNP3 protocol	DNP 3.0	DNP 3.0	DNPLPRT	5	
Modbus protocol	MBS	MBS	MBSLPRT	5	
IEC 61850 Edition 2.1 and Edition 1					
Redundancy protocols PRP/HSR					
Time synchronization for non-redundant and redundant protocols: SNTP, IRIG-B, PTP IEEE 1588 (redundant only)					
Ethernet communication with dual IP address					

Cyber security
Role-based account management including password hardening
Relay communication hardening with detection and shielding for Denial of Service attacks
Security event logging supporting Central Activity Logging (CAL) via Syslog
Secure communication with WebHMI and relay configuration tool
Central Account Management with Windows AD
Centralised security certificate management with public key infrastructure

Hardware	Amount
Standard	
Current inputs	0-7
Voltage inputs	0-5
Current sensors (according to IEC 61869)	0/3
Voltage sensors (according to IEC 61869)	0/3
Binary inputs/Binary outputs	0-18/6-13
RTD/mA inputs	0-6/0-2
Communication ports RJ45/LC/RS-485/ST	0-3/0-3/0-1/0-1
Wide	
Current inputs	0-8
Voltage inputs	0-6
Current sensors (according to IEC 61869)	0/3
Current inputs (according to IEC 61869)	0/3
Binary inputs/Binary outputs	0-32/6-21
RTD/mA inputs	0-14/0-5
Communication ports RJ45/LC/RS-485/ST	0-3/0-3/0-1/0-1

ABB Oy
Distribution Solutions
P.O. Box 699
65101 Vaasa, Finland
abb.com/mediumvoltage

ANNEX IV – SSC600[5]

1. Description

Smart Substation Control and Protection for electrical systems SSC600 is a smart substation device designed for protection, control, measurement and supervision of utility substations and industrial switchgear and equipment. The design of the product has been guided by the IEC 61850 standard for communication and interoperability of substation automation devices. It is fully integrable with Relion series IEDs and standard compliant merging units for creating a complete solution. Optional functionality is available at the time of order for both software and hardware, for example, special application packages and additional communication modules.

Figure 1: SSC600

Re-engineered from the ground up, the SSC series has been designed to unleash the full potential of the IEC 61850 standard for communication and interoperability between substation automation devices.

The device provides main protection for overhead lines and cable feeders in distribution networks. The device is also used in applications, where an independent and redundant protection system is required.

Depending on the chosen product options, the device is adapted for:

- Protection for overhead line and cable feeder in isolated neutral, resistance earthed, compensated and solidly earthed networks.
- Protection, control, measurement and supervision of asynchronous motors in manufacturing and process industry.
- Transformer protection and control for power transformers, unit and step-up transformers including power generator-transformer blocks in utility and industry power distribution systems.
- Busbar protection for medium voltage switchgears
- Shunt capacitor bank protection

Once the product has been given the application-specific settings, it can directly be put into service.

In addition to the turnkey SSC600, the centralized protection product is also available as a pure software distribution – SSC600 SW. SSC600 SW includes the same functionality as SSC600. SSC600 SW is distributed as a virtual machine which can be installed to KVM or VMware hypervisors.

SSC600 SW can be installed to freely chosen computing hardware. As long as the minimum requirements of the HW are met, SSC600 SW will perform according to the same technical data as SSC600.

2. Protection functions

The product offers directional and non-directional overcurrent and thermal overload protection as well as directional and nondirectional earth-fault protection. Some product options allow as an option admittance-based or wattmetric-based earth-fault protection to be used in addition to directional earth-fault protection. Further, the device features sensitive earth-fault protection, phase discontinuity protection, transient/intermittent earth-fault protection, distance protection, overvoltage and undervoltage protection, residual overvoltage protection, positive-sequence undervoltage and negative-sequence

[5] https://techdoc.relays.protection-control.abb/v/u/SSC600-and-SSC600-SW-Product-Guide/1.5/en-US

overvoltage protection. Frequency protection, including overfrequency, underfrequency and frequency rate-of-change protection, is offered in devices with some product options. The device also incorporates optional three-pole multishot autoreclosing functions for overhead line feeders. Starting from version 1.5, SSC600 also offers line differential protection for up to 4 lines.

The device features three-phase, multi-slope stabilized (biased) stage transformer differential protection and an instantaneous stage to provide fast and selective protection for phase-to-phase short circuit, winding interturn fault and bushing flashover protection. Besides second harmonic restraint, an advanced waveform-based blocking algorithm ensures stability at transformer energization and a fifth harmonic restraint function ensures good protection stability at moderate overexcitation of power transformers.

Sensitive restricted earth-fault protection completes the overall differential protection providing detection of even single phase-to-earth faults close to the neutral earthing point of the transformer. Numerical low-impedance scheme can be used for protection of the transformer windings. When low-impedance restricted earth-fault protection is used neither stabilizing resistors nor varistors are needed and as a further benefit the transforming ratio of the neutral earthing CTs can differ from those of the phase current transformers. Due to its unit protection character and absolute selectivity restricted earth fault does not need to be timegraded with other protection schemes, and therefore high-speed fault clearance can be achieved. The device also incorporates a thermal overload protection function, which supervises the thermal stress of the transformer windings to prevent premature aging of the insulation of the windings. Multiple stages of short circuit, phase overcurrent, negative-sequence and earth-fault backup protection are separately available for both sides of the power transformer. Earth-fault protection based on the measured or calculated residual voltage is also available. Further, the device also offers circuit breaker failure protection.

The device offers all the functionality needed to manage motor start-ups and normal operation, also including protection and fault clearance in abnormal situations. The main features of the device include thermal overload protection, motor start-up supervision, locked rotor protection and protection against too frequent motor start-ups. The device also incorporates nondirectional earth-fault protection, negative phase-sequence current unbalance protection and backup overcurrent protection. Furthermore, the device offers motor running stall protection, loss-of-load supervision and phase reversal protection. In certain motor drives of special importance there must be a possibility to override the motor thermal overload protection to perform an emergency start of a hot motor. To enable an emergency hot start, the SSC600 offers a forced start execution feature.

Some product options additionally offer multifrequency admittance-based earth-fault protection providing selective directional earth-fault protection for high-impedance earthed networks. The operation is based on multifrequency neutral admittance measurement utilizing fundamental frequency and harmonic components in Uo and Io. A special filtering algorithm enables dependable and secure fault direction also during intermittent/restriking earth faults. It provides a very good combination of reliability and sensitivity of protection with a single function for low ohmic and higher ohmic earth faults and for transient and intermittent or restriking earth faults.

SSC600 also features optional arc fault protection of the circuit breaker, busbar and cable compartment of metal-enclosed indoor switchgear. Light detection needs to be included in the bay level devices. Another option for busbar protection is low-impedance based busbar differential protection covering up to 30 three phase current measurement points and 4 protection zones.

3. Supported ABB Solutions

The SSC600 Smart Substation devices together with the ABB Ability™ Electrification Monitoring and Control ZEE600 constitute a genuine IEC 61850 solution for reliable power distribution in utility and industrial power systems. To facilitate the system engineering, ABB's devices are supplied with connectivity packages. The connectivity packages include a compilation of software and device-specific information, including single-line diagram templates and a full device data model. The data model includes event and parameter lists. With the connectivity packages, the devices can be readily configured using PCM600 and integrated with ZEE600.

The SSC600 Smart Substation devices offer native support for IEC 61850 Edition 2 also including binary and analog horizontal GOOSE messaging. In addition, with process bus the receiving and sending of sampled values is supported. Compared to traditional hard-wired, inter-device signaling, peer-to-peer communication over a switched Ethernet LAN offers an advanced and versatile platform for power system protection. Among the distinctive features of the protection system approach, enabled by the full implementation of the IEC 61850 substation automation standard, are fast communication capability, continuous supervision of the protection and communication system's integrity, and an inherent flexibility regarding reconfiguration and upgrades.

At substation level, ZEE600 uses the data content of the devices to enhance substation level functionality. ZEE600 features a Web browser-based HMI, which provides a customizable graphical display for visualizing single-line mimic diagrams for switchgear bay solutions. Substation devices and processes can also be remotely accessed through the Web HMI, which improves personnel safety.

ZEE600 can also function as a gateway and provide seamless connectivity between the substation devices and network-level control and management systems.

4. Control

SSC600 integrates functionality for the control of a circuit breaker via the Web HMI or by means of remote controls. In addition to the circuit-breaker control the device features control blocks which are intended for motor-operated control of disconnectors or circuit breaker truck and for their position indications. Further, the device offers control block that is intended for motor-operated control of earthing switch control and its position indication.

Two physical binary inputs and two physical binary outputs are needed in the bay level merging unit or in the IED (not in SSC) for each controllable primary device taken into use.

If the amount of available binary inputs or outputs of the chosen merging unit or IED is not sufficient, an external input or output module, for example, RIO600 can be integrated to the IED. The binary inputs and outputs of the external I/O module can be used for the less time critical binary signals of the application.

SSC600 includes WHMI and a single-line diagram (SLD) with position indication for the relevant primary devices. Interlocking schemes required by the application are configured using the signal matrix or the application configuration functionality of PCM600. Depending on the product options, the device also incorporates a synchrocheck function to ensure that the voltage, phase angle and frequency on either side of an open circuit breaker satisfy the conditions for safe interconnection of two networks.

5. Measurements

Based on received sampled value streams, SSC600 continuously measures

- The phase currents
- The symmetrical components of the currents
- The residual current and residual voltage, based on the received process bus measurements
- The residual voltage
- The phase voltages
- The voltage sequence components
- Frequency

The device also calculates the demand value of the current over a user-selectable, pre-set time frame, the thermal overload of the protected object, and the phase unbalance based on the ratio between the negative-sequence and positive-sequence current.

Furthermore, the device offers three-phase power and energy measurement including power factor.

The measured values can be accessed remotely via the communication interface of the device. The values can also be accessed locally or remotely using the Web HMI.

6. Power quality

In the EN standards, power quality is defined through the characteristics of the supply voltage. Transients, short-duration and long-duration voltage variations and unbalance and waveform distortions are the key characteristics describing power quality. The distortion monitoring functions are used for monitoring the current total demand distortion and the voltage total harmonic distortion.

Power quality monitoring is an essential service that utilities can provide for their industrial and key customers. A monitoring system can provide information about system disturbances and their possible causes. It can also detect problem conditions throughout the system before they cause customer complaints, equipment malfunctions and even equipment damage or failure. Power quality problems are not limited to the utility side of the system. In fact, the majority of power quality problems are localized within customer facilities. Thus, power quality monitoring is not only an effective customer service strategy but also a way to protect a utility's reputation for quality power and service.

The protection relay has the following power quality monitoring functions.

- Voltage variation
- Voltage unbalance
- Current harmonics
- Voltage harmonics

The voltage unbalance and voltage variation functions are used for measuring short-duration voltage variations and monitoring voltage unbalance conditions in power transmission and distribution networks.

The voltage and current harmonics functions provide a method for monitoring the power quality by means of the current waveform distortion and voltage waveform distortion. The functions provide a short-term three-second average and a long-term demand for total demand distortion TDD and total harmonic distortion THD.

Historical data of the selected power quality measurements can be viewed with SSC600 WHMI. This functionality can be used when analyzing long term trends or when identifying power quality related problems in the past.

7. Disturbance recorder

The device is provided with a disturbance recorder featuring up to 390 analog channels and 512 binary channels.

The binary signal channels can be set to start a recording either on the rising or the falling edge of the binary signal or on both.

Normally, the binary channels are set to record external or internal device signals, for example, the start or trip signals of the device stages, or external blocking or control signals. Binary device signals, such as protection start and trip signals, or an external device control signal via a binary input, can be set to trigger the recording. Recorded information is stored in a non-volatile memory and can be uploaded for subsequent fault analysis.

SSC600 also includes functionality to trigger disturbance recordings based on indeterministic anomalies in the electrical grid. These recordings can then be used in combination with artificial intelligence to achieve advanced analytics like for example fault prediction.

8. Event log

To collect sequence-of-events information, the device has a nonvolatile memory capable of storing 100000 events with the associated time stamps. The non-volatile memory retains its data even if the device temporarily loses its auxiliary power supply. The event log facilitates detailed pre- and post-fault analyses of feeder faults and disturbances. The considerable capacity to process and store data and events in the device facilitates meeting the growing information demand of future network configurations.

The sequence-of-events information can be accessed remotely via the communication interface of the device. The information can also be accessed locally or remotely using the Web HMI.

9. Recorded data

The device has the capacity to store the records of the 128 latest fault events. The records can be used to analyze the power system events. Each record includes, for example, current, voltage and angle values and a time stamp. The fault recording can be triggered by the start or the trip signal of a protection block, or by both. The available measurement modes include DFT, RMS and peak-to-peak. Fault records store device measurement values at the moment when any protection function starts. In addition, the maximum demand current with time stamp is separately recorded. The records are stored in the non-volatile memory.

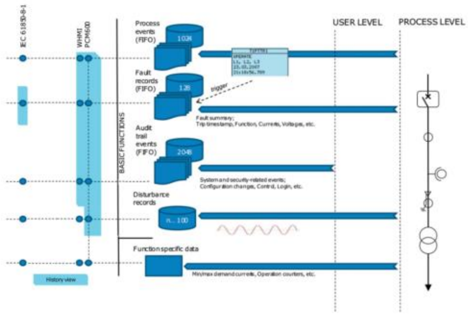


Figure 2: Recording and event capabilities overview

10. Condition monitoring

The condition monitoring functions of the device constantly monitor the performance and the condition of the circuit breaker. The monitoring comprises the spring charging time, gas pressure, the travel time and the inactivity time of the circuit breaker.

The monitoring functions provide operational circuit breaker history data, which can be used for scheduling preventive circuit breaker maintenance.

In addition, the device includes a runtime counter for monitoring of how many hours a protected device has been in operation thus enabling scheduling of time-based preventive maintenance of the device.

11. Self-supervision

The device's built-in self-supervision system continuously monitors the state of the device hardware and the operation of the device software. Any fault or malfunction detected is used for alerting the operator.

A permanent device fault blocks the protection functions to prevent incorrect operation.

12. Access control

To protect the device from unauthorized access and to maintain information integrity, the device is provided with a role based access control with programmable individual passwords for freely configurable users and roles. The access control applies to the Web HMI and PCM600.

13. Station communication

The device supports a range of communication protocols including IEC 61850 Edition 1, Edition 2, DNP3 and IEC 60870-5-104. Operational information and controls are available through these protocols.

The IEC 61850 protocol is a core part of the device as the protection and control application is fully based on standard modelling. The device supports Edition 2 and Edition 1 versions of the standard. With Edition 2 support, the device has the functionality modelling for substation applications and the best interoperability for modern substations. It incorporates also the full support of standard device mode functionality supporting different test applications including simulation as specified in IEC 61850. Control applications can utilize the new safe and advanced station control authority feature.

The IEC 61850 communication implementation supports monitoring and control functions. Additionally, parameter settings, disturbance recordings and fault records can be accessed using the IEC 61850 protocol. Disturbance recordings are available to any Ethernet-based application in the standard COMTRADE file format. The device supports simultaneous event reporting to five different clients on the station bus. The device can exchange data with other devices using the IEC 61850 protocol.

The device can send binary and analog signals to other devices using the IEC 61850-8-1 GOOSE (Generic Object Oriented Substation Event) profile. Binary GOOSE messaging can, for example, be employed for giving control commands to merging units. The device meets the GOOSE performance requirements for tripping applications in distribution substations, as defined by the IEC 61850 edition 2 standard (<3 ms point-to-point data exchange between the devices). The device also supports the sending and receiving of analog values using GOOSE messaging. Analog GOOSE messaging enables easy transfer of analog measurement values over the station bus, thus facilitating for example the sending of measurement values between the devices when controlling parallel running transformers.

The device also supports IEC 61850 process bus by receiving sampled values of voltages and currents. With this functionality the galvanic interpanel wiring can be replaced with Ethernet communication. The measured values are received as sampled values using IEC 61850-9-2 LE or IEC 61869-9 protocol. SSC600 devices with process bus based applications use IEEE 1588 for high accuracy time synchronization.

For redundant Ethernet communication, the device offers either two optical or two galvanic Ethernet network interfaces, depending on the product variant. Ethernet network redundancy can be achieved using the parallel redundancy protocol (PRP). In addition to the process communication, SSC600 also has dedicated Ethernet interfaces for local WHMI, engineering and remote access. Protection communication, mainly used in line differential protection, have a dedicated optical Ethernet interface.

The IEC 61850 standard specifies network redundancy which improves the system availability for the substation communication. The network redundancy is based on protocols defined in the IEC 62439-3 standard: PRP protocol. The protocol is able to overcome a failure of a link or switch with a zero switch-over time. In the protocol, each network node has two identical Ethernet ports dedicated for one network connection. The protocol relies on the duplication of all transmitted information and provide a zero switchover time if the links or switches fail, thus fulfilling all the stringent real-time requirements of substation automation.

In PRP, each network node is attached to two independent networks operated in parallel. The networks are completely separated to ensure failure independence and can have different topologies. The networks operate in parallel, thus providing zero-time recovery and continuous checking of redundancy to avoid failures.

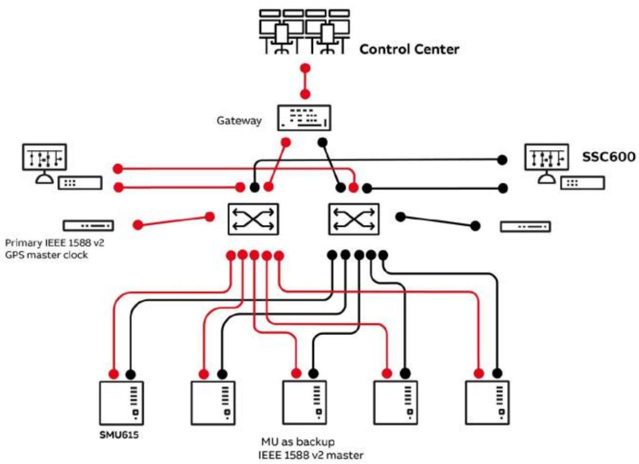


Figure 3: Parallel redundancy protocol (PRP) solution

The device can be connected to Ethernet-based communication systems via the RJ-45 connector (1000Base-TX) or the fiber optic LC connector (1000Base-SX), depending on the product variant.

The device supports the following high accuracy time synchronization method with a timestamping resolution of 4 µs required especially in process bus applications.

- PTP (IEEE 1588) v2 receive (slave) with Power Profile

Required accuracy of grandmaster clock is +/-1 µs.

14. Technical data

Table 1: Dimensions

Description	Value
Width	440 mm / 17.3"
Height	88 mm / 3.4"
Depth	220 mm / 8.6"
Weight	6.0kg
Mounting	2U Rack mount (19"), fits into standard 19" rack

Table 2: Power supply

Description	High Voltage variant	Low Voltage variant
Nominal auxiliary voltage U_n	100...240 VAC 50 and 60Hz	36...72 VDC
	100...240 VDC	

Table continues on the next page