UiT
THE ARCTIC UNIVERSITY OF NORWAY

Department of Electrical Engineering

Communication in microgrids and virtual power plants

Oleksandr Starynets

Master thesis in Electrical Engineering, June 2016

Title: Communication in microgrids and virtual power plants

Date: 20.06.2016

Classification: Open

Author: Oleksandr Starynets

Pages: 55

Attachments: 3

Departement: Department of Electrical Engineering

Studieretning: Electrical Engineering

Supervisor: Bjarte Hoff, Pawan Sharma

Principal: UiT – The Arctic University of Norway (Campus Narvik)

Principal contact: Bjarte Hoff

Keywords: Microgrid, Virtual power plant, Communication technologies, Modbus, IEC 61850

Abstract: One of the cornerstones of the steady operation of microgrids and virtual power plants as building blocks for smart grid is the communication system, which is the main objective for evaluation and research in this thesis.

The given project investigates the most widespread communication protocols along with IEC 61850 standard for substations automation applied in smart grids. Based on the presented analysis for communication technologies and protocols the appropriate communication solution for the laboratory microgrid at UiT – The Arctic University of Norway (Campus Narvik) is suggested and implemented.

Preface and Acknowledgements

The given thesis is submitted in fulfillment of the requirements for achieving the degree of Master of Science in Electrical Engineering at UiT – The Arctic University of Norway (Campus Narvik).

The accomplishing of this project would not be possible without specific and accurate aid of my supervisor, PhD Candidate Bjarte Hoff, during the whole period of the project development. Thank you for providing this interest topic for the master project, which fit with my background and let me expand my knowledge in this field and gain some valuable practical experience.

Also I would like to thank my second supervisor, Associate Professor Dr. Pawan Sharma, for his guidelines and support.

Special thank you goes to my classmates for these great two years of studies, sharing their knowledge and participating in different projects and courses.

Finally, I would like to thank my parents and my girlfriend for their belief, encouragement and constant support during my work on this master thesis.

Oleksandr Starynets Narvik 20.06.2016

Contents

Preface and Acknowledgements iii
List of Figures vi
List of Tables vii
List of Appendices viii
Abbreviations ix
1. Introduction 1
2. Microgrids 2
2.1. Overview of microgrid technology 2
2.2. Examples of microgrid's implementation 3
3. Virtual power plants 6
3.1. Overview of VPP technology 6
3.2. Examples of VPP's implementation 7
4. Communication technologies in microgrids and VPPs 10
4.1. Objectives and prerequisites 10
4.2. Communication infrastructures 10
4.3. Communication network architectures 15
4.4. Communication protocols 16
4.5. Preferred communication protocol 19
5. Laboratory microgrid test based on Modbus communication 21
5.1. Communication system setup for laboratory microgrid 21
5.2. Modbus protocol description 22
5.3. Hardware and software resources for Modbus device model development and verification 24
5.4. MATLAB/Simulink Modbus test model 25
5.5. Modbus model simulation tests over RS-232 communication 27
5.6. Modbus communication over RS-485 serial connection 31
5.6.1. Equipment connection and configuration 31
5.6.2. RS-485 network configuration 31
5.6.3. Communication tests over RS-485 network 32
6. Communication structure for substation and feeder equipment in IEC 61850 standard 34
6.1. Abstract Communication Service Interface (ACSI) 34
6.2. Substation Configuration Language (SCL) 36
6.3. Manufacturing Message Specification (MMS) 37
7. PowerLogic G3200 configuration 38
7.1. Creating and transferring the CID file to G3200 38
7.2. Configuring the CID file for dSPACE DS1104 connection panel 39
7.2.1. G3200 global structure 39

7.2.2. Creating the dSPACE DS1104 logical device 40
7.2.3. Configuring the dSPACE DS1104 logical device 42
8. Laboratory microgrid test based on IEC 61850 communication 48
8.1. IEC 61850 client software overview 48
8.2. Laboratory microgrid test and results 48
9. Conclusions and recommendations for further work 52
References 53
Appendices

List of Figures

Figure 2.1 - Organization of microgrid [1]. 2
Figure 2.2 – Typical structure of microgrid [6]. 3
Figure 2.3 – Hachinohe system [9]. 4
Figure 2.4 – BC Hydro Boston Bar system [6]. 5
Figure 2.5 – AEP CERTS microgrid [6]. 5
Figure 3.1 – Integration of DERs [13]. 6
Figure 3.2 – Configuration of VPP [13]. 7
Figure 3.3 – Virtual power plant: DERs aggregation (Fenix project) [14]. 8
Figure 3.4 – Architecture of the EV test bed [15]. 8
Figure 3.5 – EV test bed operation in a VPP framework [15]. 9
Figure 4.1 – Composition of CEDER microgrid communication and power system [20]. 11
Figure 4.2 – Multi-layer hierarchical structure of communication networks for SG [22]. 15
Figure 4.3 – The structure of Modbus protocol layers [29]. 17
Figure 4.4 – DNP3 network topologies [27]. 18
Figure 5.1 – Modbus over RS-485 communication equipment configuration. 21
Figure 5.2 – Modbus Master-Slave request-response cycle [33]. 22
Figure 5.3 – Modbus function codes categories [29]. 24
Figure 5.4 – Modbus simulation test hardware. 24
Figure 5.5 – The block diagram of Simulink model for Modbus protocol test. 25
Figure 5.6 – Modbus response function flow chart diagram. 26
Figure 5.7 – Connection setup of Modbus Poll software. 27
Figure 5.8 – DS1104SER block setup. 27
Figure 5.9 – DS1104SER setup for data receiving and transmitting. 28
Figure 5.10 – Modbus Poll communication session setup for 03 function code (10 registers). 28
Figure 5.11 – Communication traffic for Modbus RTU test of 03 function code (10 registers). 29
Figure 5.12 – Received data for 10 holding registers for Modbus RTU test of 03 function code. 29
Figure 5.13 – Modbus Poll communication session setup for 03 function code (6 registers). 29
Figure 5.14 – Communication traffic for Modbus RTU test of 03 function code (6 registers). 30
Figure 5.15 – Received data for 6 holding registers for Modbus RTU test of 03 function code. 30
Figure 5.16 – G3200 external wiring diagram. 31
Figure 5.17 – G3200 serial port configuration window. 32
Figure 5.18 – DS1104SER block setup for Modbus RTU communication over RS-485 network. 32
Figure 5.19 – G3200 communication session for Modbus 03 function code (10 registers). 33
Figure 5.20 - G3200 communication session for Modbus 03 function code (6 registers). 33
Figure 6.1 – Basic conceptual class model of the ACSI [37]. 34
Figure 6.2 – The concept of DAType class [37]. 35
Figure 6.3 – UML diagram overview of SCL schema [39]. 37
Figure 6.4 – Structure of an IEC 61850-8-1 Object Name [40]. 37
Figure 7.1 – G3200 Summary window in device’s web server. 39
Figure 7.2 – G3200 logical devices [36]. 39
Figure 7.3 – The CID file sections. 40
Figure 7.4 – The tree view of G3200 CID file. 41
Figure 7.5 – G3200 Summary web page with connected dSPACE DS1104. 47
Figure 8.1 – The application diagram of Sulkowski laboratory microgrid. 49
Figure 8.2 – The IED tree view (MMS) for the connected G3200 in IEDExplorer. 49
Figure 8.3 – Modbus holding registers red over IEC 61850 protocol. 50

List of Tables

Table 5.1 – Modbus RTU protocol specification....23
Table 5.2 – Modbus RTU message frame [33]. 23
Table 5.3 – Modbus data model [29]. 23
Table 7.1– LLNO (Logical node zero) description [36]. 40
Table 7.2 – LPHD (Physical device information) description [36]. 40
Table 7.3 – List of the IEC 61850 object types for application in G3200 [36]. 44
Table 7.4 – G3200 supported processing codes [36]. 45
Table 8.1 – Encoding and decoding of Modbus registers. 50

List of Appendices

Appendix A MATLAB/Simulink Modbus device model

Appendix B Modbus response function code

Appendix C The CID file for G3200 Modbus-to-IEC 61850 server ‘G3200- DS1104’

Abbreviations

ACSI	Abstract communication service interface
CID	Configured IED description
CRC	Cyclical redundancy check
DA	Data attribute
DER	Distributed energy resource
DG	Distributed generation
DO	Data object
FC	Functional constraint
FTP	File transfer protocol
ICD	IED capability description
IEC	International Electrotechnical Commision
IED	Intelligent electronic device
LAN	Local area network
LD	Logical device
LN	Logical node
MMS	Manufacturing message specification
PLC	Power line communication, programmable logic controller
RTU	Remote terminal unit
SCADA	Supervisory control and data acquisition
SCL	Substation configuration language
SG	Smart grid
VPP	Virtual power plant
XML	Extensible markup language

1. Introduction

A smart grid (SG), which can be named as smart electrical/power grid, intelligent grid, futuregrid or intragrid, is regarded as a next step in the development of the traditional power grid. The conventional power grids are functioning as vertically oriented systems providing energy from heavy-powered generators to consumer utilities through transmission and distribution networks. Opposing to that, the SG operates with bidirectional electricity and information flows to form automated and allocated advanced power supply system.

Organization and functioning of SG could not be possible without distributed generation (DG), which is a part of distributed energy resources (DERs) such as solar panels, small wind turbines, biogas installations, energy storages and loads with a rate of 3 kW to 10 MW. The formation of DG directs us to a new energy production approach called virtual power plant (VPP), a big amount of small generators, as well as other types of DERs, spread on a large area with common control and supervision. The overall capacity of these resources is equivalent to a traditional power plant and can be used for improving stability and reliability of the local power network by covering peak load demand electricity on short notice [1]. Additionally, VPP can be used as network frequency control sources.

DG enabled an establishment of a new grid organization, called microgrid, which is a building block of the future smart grids [1]. According to [2], microgrid is a complex energy system comprising several DERs and consumers operating in two modes: parallel mode or 'islanded' mode from the local utility grid. Along with that, microgrids use distributed storage (DS) when the generation and consumption are not coincide and there is a necessity in fulfillment of power and energy requirements of the grid [3].

The purpose of this project is to develop a new communication solution for microgrid in the Energy laboratory (Sulkowski laboratory) at UiT – The Arctic University of Norway (Campus Narvik). The laboratory microgrid comprises power electronics converter linked to the separate connection panel through measurement and communication interfaces. The laboratory also equipped with the personal computer (PC), which has installed software for configuration and communication and interacts with the power electronics converter through the PCI (Peripheral Component Interconnect) card, connected to the communication panel. The main objectives for the given project are:

- Literature research of microdrid and VPP technologies with the main accent in communication systems applied;
- Development of the appropriate communication solution for the Sulkowski laboratory microgrid either with already existing communication equipment or with some new configuration along with evaluating and utilizing the most suitable communication protocols including IEC 61850 standard;
- The selected protocols should be adopted and implemented for communication with the connection panel interfaces of power electronics converter.

The resulted communication system is intended to be capable of reading the data over chosen communication protocols without write or control functions. The expected outcome is the microgrid monitoring system deployed on the separate PC functioning through the local network communication switch.

2. Microgrids

2.1. Overview of microgrid technology

Exploiting the traditional energy sources for the power production is no longer reliable due to their considerable limitation in reserves. The extra high impact on the recent climate change caused serious environmental concerns. These prerequisites make DG, using renewable sources, especially important nowadays. Furthermore, a lot of countries implement local energy projects on development of renewables as a national energy policy. Such circumstances have given a necessary impulse for appearing and evolvement of microgrids [4].

The most prominent feature of microgrid is its self-sufficiency, which means that it can maintain self-control, safety and management. It acts as a complete separate power system for achieving power balance control, system operation optimization, fault detection and protection, power quality control function, etc. [5]. At the normal operation mode microgrid is connected to a conventional power grid (macrogrid) and the participants of microgrid can produce electrical energy using DERs. It can be disconnected from the point of common coupling with the macrogrid resulting in autonomous functioning or islanded mode, when DERs proceeding to supply the users without receiving power from the local utility grid. In such a way, a microgrid isolates its components from disturbances from larger network and provides highly reliable energy supply [1]. Fig. 2.1 shows an organization of the microgrid.

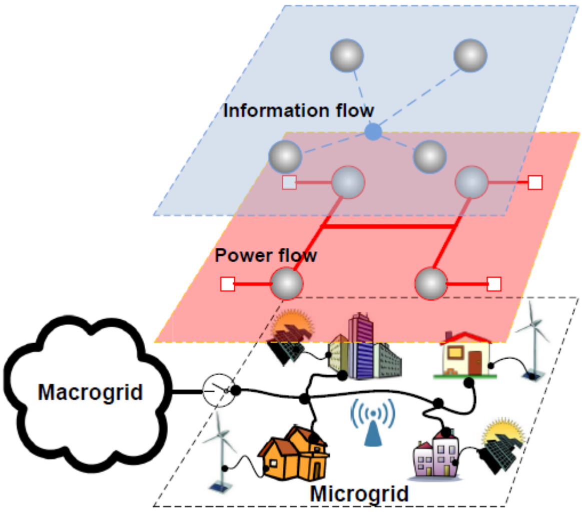


Figure 2.1 - Organization of microgrid [1].

Dependent on the structure and particular technical requirements the typical microgrid consists of the following elements:

- **Connection interface** – used for interaction with the utility grid. Can be implemented in several ways: electro-mechanical circuit breaker, solid-state switches or back-to-back (AC-DC-AC) power electronic converters;
- **Energy storage** – requires for balancing of power loads on the transition to the islanded mode as a fast-acting energy absorption or injection application. In the more technologically advanced microgrid control structure, energy storage could also be necessary for control net power flows in grid connected mode on both directions to the utility. In this way microgrid is capable to assist the stable network operation by providing improved power quality and voltage control;

- **Loads and micro-generation** – designate the requirements for energy storage and power quality for microgrids. Power electronics converters (PE) are added to load/generation units for extra controllability and speed of response. PE can also be used to provide continuous variable power consumption by noncritical power loads, for instance, ventilation. Disadvantages related to PE are extra harmonics injection to the grid and sensitiveness to system disturbances;
- **Central control** – needed for coordination of all microgrid elements to make them operate as a system. Such parameters as common microgrid state variables, instantaneous phase currents and voltages at the point of common coupling have to be regulated [6].

Typical structure of microgrid with key elements is shown in Fig. 2.2.

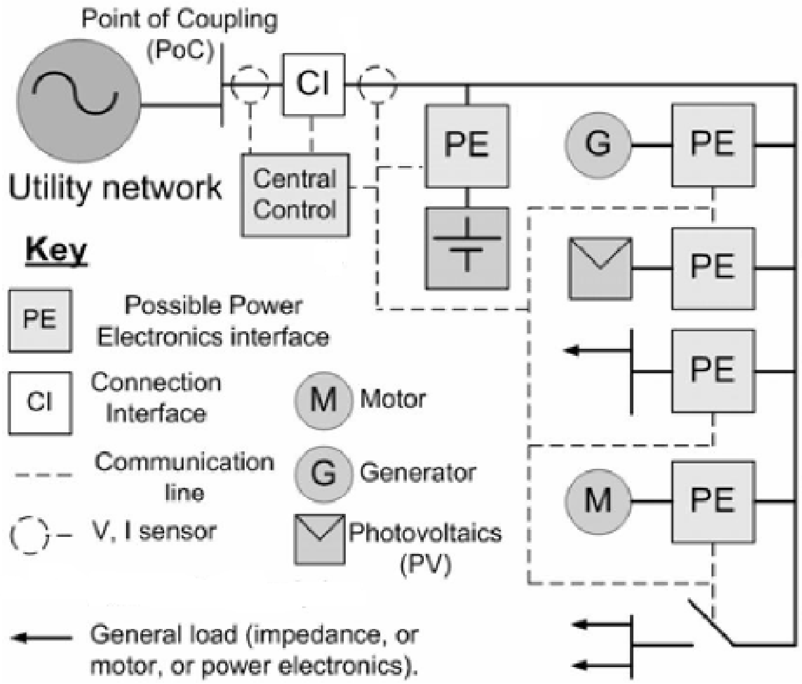


Figure 2.2 – Typical structure of microgrid [6].

Another important issue in the functioning of microgrids is the system protection. It has to respond both to utility and microgrid faults. If the fault occurs on the utility side, the desired response should be in the disconnecting of microgrid as fast as possible for protection of micgrogrid loads. The speed of isolation is dependent on the characteristics of customer’s loads. Sometimes, the voltage drop compensation can be used without disconnection from utility grid for the sake of protection of critical loads. When the fault occurs inside the microgrid, the protection system isolates the smallest possible section of the radial bus to exclude the fault.

The most convenient distribution protection uses short-circuit current sensing [7]. Another commonly used protection systems in microgrids nowadays are based on microprocessor technologies and communication links. Such solutions include differential protection, comparative voltage protection and the detection of high impedance faults and can be used both in utility-connected and islanded modes [8].

2.2. Examples of microgrid’s implementation

Hachinohe project

The New Energy and Industrial Technology Organization (NEDO) in Japan has started three demonstrations as a part of the the ‘Regional Power Renewable Energy Resources Project’. These projects are important for the national program because of significant share of renewable energy in their microgrids. One of them situated in Hachinohe, the city at the northern part of Honshu Island.

This project is the collaboration between Hachinohe city, Mitsubishi Research Institute and Mitsubishi Electrics. The grid consists of a 6 kV feeder which connects four schools, the local city hall and a sewage treatment facility. A number of loads are connected to PV, wind turbines, energy storage and three large gas engines running on biogas from sewage. Extracted heat from the gas engines is utilized in the sewage fermentation process [9].

The control system is represented by 'virtual 'prime mover'' scheme, where a central controller samples microgrid state variables and transmits signals to all DERs (or at least to those who have the biggest capacity) by using fast telecommunications. This aggregation forms one virtual power supply unit which dictates and controls microgrid's behavior [6]. Fig. 2.3 shows the Hachinohe project's microgrid (key notations are used from the Fig. 2.2).

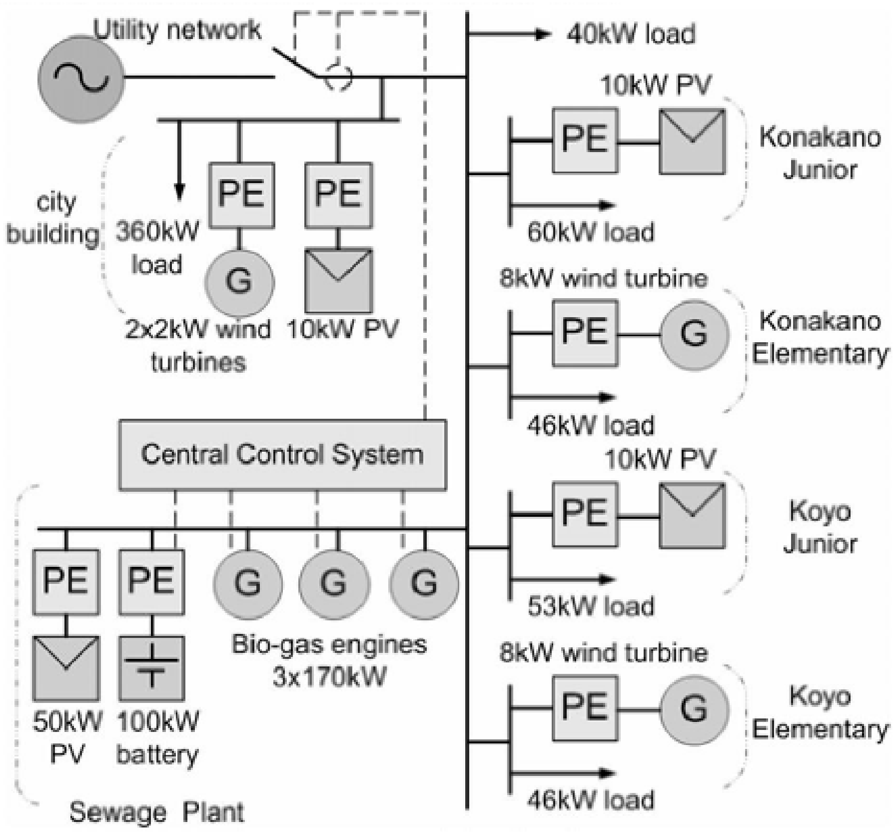


Figure 2.3 – Hachinohe system [9].

BC Hydro Boston Bar

The British Columbia Hydro Boston Bar substation permits necessary islanding of 3 MW of peak load and 8.6 MVA hydroelectric generation [10]. The power blackouts take place several times per year between 12 and 22 hours. The system effectively uses a single large power station to control the grid sub-system behavior. This control approach is called 'physical 'prime mover'' where a big central hardware unit (generator or storage) is responsible for handling transient power flows and setting the voltage magnitude and frequency to balance steady-state active and reactive power flows in islanded mode [6]. Fig. 2.4 depicts the organization of BC Hydro Boston Bar.

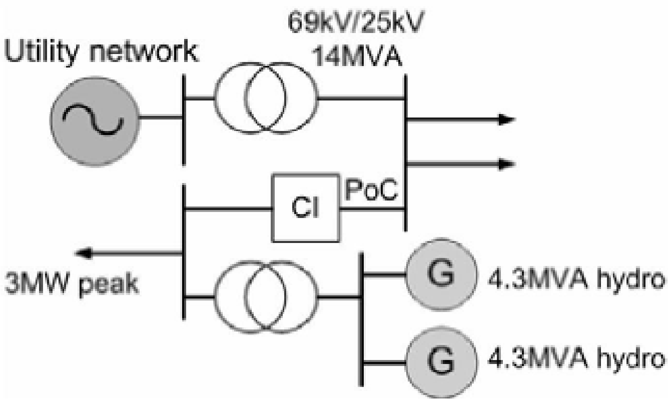


Figure 2.4 – BC Hydro Boston Bar system [6].

AEP CERTS

The Consortium for Electric Reliability Technology Solution (CERTS) microgrid is a collaboration between AEP (American Electric Power Inc.), Tecogen, Northern Power Systems, S&C Electric Co, Sandia National Laboratories and the University of Wisconsin. The system is located at AEP Walnut Test facility and organized by the radial feeder with the lengths near 175 yards connected through the 1.5 MVA 0.4/13.2 kV transformer to the utility grid. The Tecogen combined heat and power sources, functioning on gas engines and interfacing through controlled inverters constitute the micro-generation. Four load banks are used: three are impedances up to 95 kW with step variation and the fourth one additionally includes directly connected induction motor, which allows islanded motor starting test. Energy storage is connected through the bidirectional DC-DC converter to the DC terminals of power electronics converter of micro-generation. The critical loads of the systems are located to the right side from the solid-state (anti-parallel thyristor) connection interface and there are always sufficient generation/storage resources to meet the critical load power demand.

The given microgrid uses distributed control scheme, where each unit responds to variation of local state variables – voltage magnitude and frequency. A slow central controller sends signals to vary nominal set-points, but for redundancy purposes the local controllers determine transient and default behavior. The speed of response of DERs is achieved by power electronics interfaces. An ‘intelligent’ connection interface is required for reconnection of microgrid to utility when their voltages (which have different frequency) pass close to the alignment [6]. Fig. 2.5 shows the organization of the AEP CERTS microgrid.

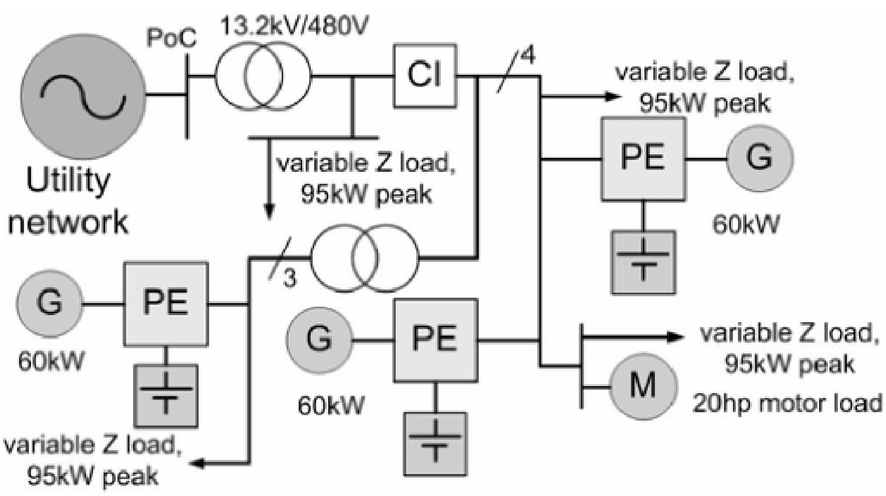


Figure 2.5 – AEP CERTS microgrid [6].

3. Virtual power plants

3.1. Overview of VPP technology

As it was stated in the introduction, the history of electricity markets development was represented by top-down approach, when the big generation facilities supplied energy to the consumers. At the present time, the traditional way of power system operation faces gradual transition to usage of DG, which is connected to distribution network or even on the consumer side. This shift is based on the market decentralization reforms and renewable energy policy. At the same time, the increased amount of DG creates challenges to the power system because of its small size, interrupted and incompletely predictable power output. To overcome these problems, DG should be integrated with DERs, comprising not only DG and energy storages, but also adjustable electrical loads, such as ventilation, heating and electrical vehicles. Such kind of integration (Fig. 3.1) leads us to a concept of virtual power plant, which gathers these various components, and participates as a separate entity in power system and energy market similar to transmission-connected generator [11], [12].

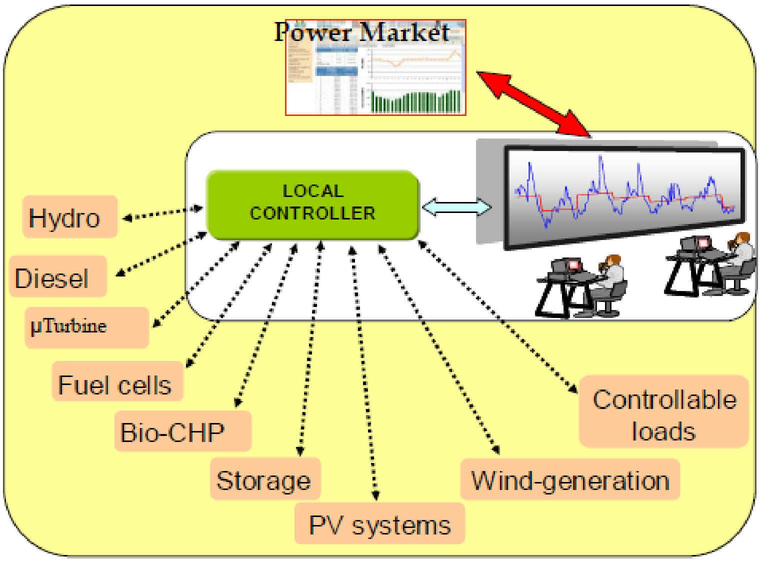


Figure 3.1 – Integration of DERs [13].

The VPP consists of some number of DERs with their individual parameters and capabilities, and aggregates them into the one profile in order to effectively contribute to the power grid. The high efficiency obtains by forming VPP in the distribution network, in such a way avoiding transmission of electricity over the long distances and consequently higher losses [13]. VPPs and microgrids are often regarded as alternative solutions for integration of DERs in the network. But in comparison to microgrids, VPPs are always grid-tied and cannot operate in islanded mode.

Virtual power plant can be characterized by three essential building blocks that form its structure:

- **Portfolio of DERs**
  It includes DG, storage units and flexible loads. As a result of integration, VPPs have a set of parameters conventionally associated with large generators, such as scheduled output, voltage regulation capability, ramp level and reserve capacity [12]. VPPs can be divided by the type of DERs presented. First type is supply-side VPPs comprising DG units. The second type is demand-response VPPs, which consist of combination of flexible loads and storages. The third type is the mixed asset VPPs having both the first and the second type features in it.

- **Control system**
  The control mechanism is crucial for effective participation in energy markets. VPP control systems are distinguished by the type of control of each DER presented in particular VPP. In a centralized system, there is one common coordination center controlling all the processes by means of logic algorithm while receiving signals from each individual unit. In decentralized VPP, all units are controlled individually by its own local controller. An integrated system is organized by connecting these controllers in separate communication network.
- **Participation in power system**
  The role of the VPP in power system is important because of its possibilities in providing services in the energy market. VPPs can act as an energy service or grid service provider. According to these functions, it can be distinguished commercial VPP (CVPP) and technical VPP (TVPP) [12]. The role of CVPP is to participate and trade in energy market in the same manner as large generators do by managing its DERs portfolio and participating in the wholesale energy market. TVPP focuses on delivering the system services to the network by assisting to transmission system operator (TSO) in frequency and voltage control, and to distribution system operator (DSO) in adjusting electricity demand or injections.

The recent years experience has given a new classification – static and dynamic VPPs. In comparison to the conventional static type, dynamic VPPs (or clusters) form separate entity on the temporary basis regarding to the current situation in the market and power forecasts from participating DERs. The dynamic VPP decomposes every time after providing of energy supply [11].

Fig. 3.2 depicts the typical structure of VPP connected to the distribution power network.

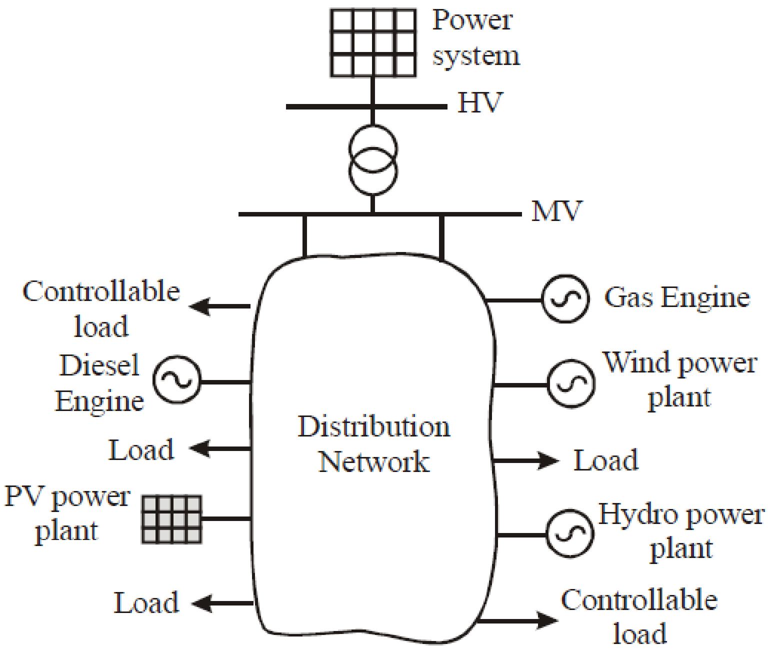


Figure 3.2 – Configuration of VPP [13].

3.2. Examples of VPP's implementation

VPP for voltage regulation

The FENIX project in the Southern Scenario demonstrator (Alava province, Spain) integrated DERs response into transmission and distribution network operation. Fig. 3.3 illustrates a portfolio of DERs connected to the distribution network, which could be seen as a virtual power plant at the transmission network node.

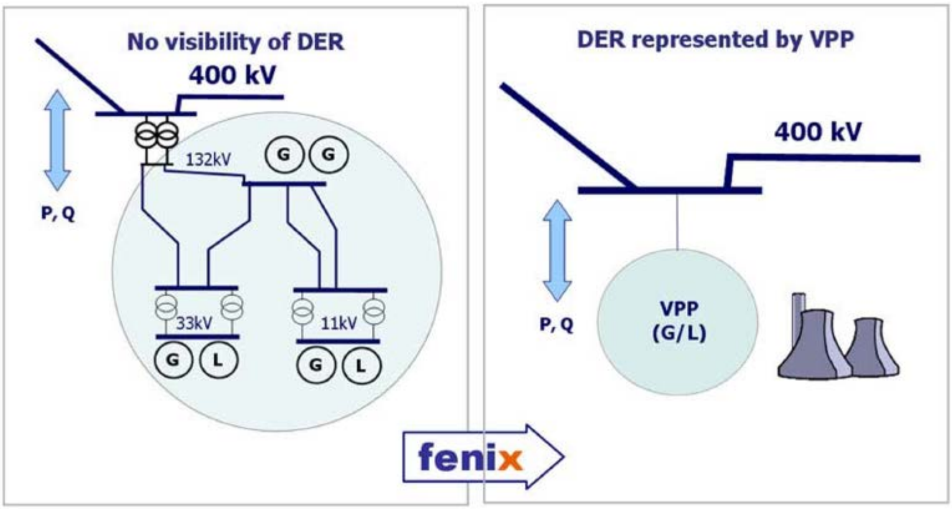


Figure 3.3 – Virtual power plant: DERs aggregation (Fenix project) [14].

The distribution management system (DMS) was presented with a constrained optimal power flow module considering control variables transformer taps, condenser switching and DERs reactive power injections for maintaining bus voltages within the stationary limits and network loadings. This voltage reactive power control module is tested on the real time DMS and received results were presented to the network operator for later execution. This voltage VAr control represents a technical VPP that, considering location specific issues, may be applied to achieve the effective integration of DER units into the power system operation [14].

Electric Vehicle Test Bed controlled by a VPP (Denmark)

This project evaluates the design of an electrical vehicle (EV) test bed, which by means of real EV components and communication interfaces, is capable to respond in real-time to smart grid control signals. The EV test bed consists of a Lithium-ion battery pack, a Battery Management System (BMS), a charger and a Vehicle-to-Grid (V2G) unit for delivering power back to the grid. The designed solution acts as a multifunctional grid-interactive EV, which a VPP or a generic EV coordinator can use for testing different control strategies, such as EV participation in regulating power services. Fig. 3.4 shows architecture of the EV test bed.

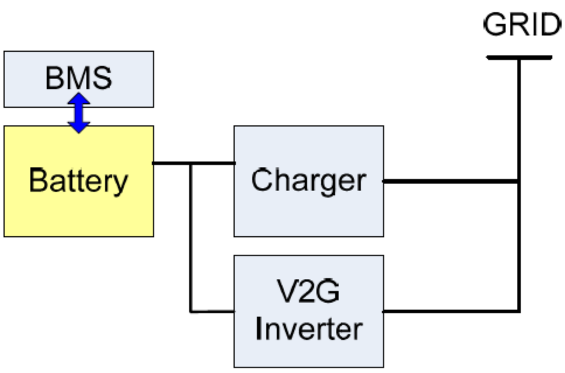


Figure 3.4 – Architecture of the EV test bed [15].

The EV test bed interface with VPP was established with the aid of Danish Edison project, where the contributors proposed a centralized control scheme for an efficient integration of EV in the power system. A real regulating power query from the Danish TSO is processed by the VPP and sent as a charging/discharging power schedule to the EV test bed. For the real functioning model different interfaces were established for interaction of the Edison VPP and other participants in the framework. The final project architecture is depicted in Fig. 3.5.

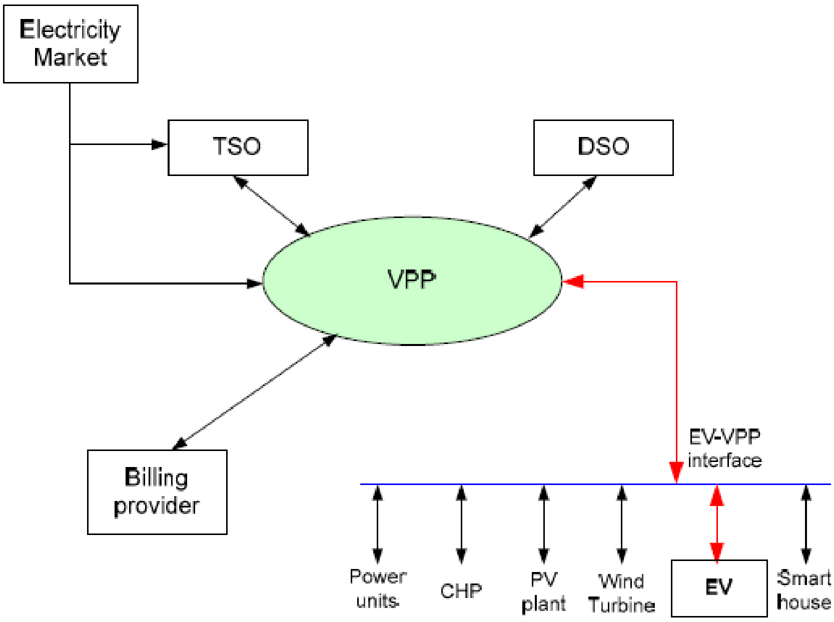


Figure 3.5 – EV test bed operation in a VPP framework [15].

Because of the relatively small capacity of the separate EV in comparison to network needs, it was assumed that VPP covers TSO requirements by utilizing the number of simulated EVs. For the real-time flexible interaction with VPP and utility grid, real EV components and interfaces were used. As a reference was taken a regulating power request for the load frequency control in 3-hours time interval sent by the Danish TSO on the 1st of January 2009. Test results have shown that EVs are potentially capable to fulfill different real-time charging/discharging requests according to various coordination plans [15].

4. Communication technologies in microgrids and VPPs

4.1. Objectives and prerequisites

The implementation of microgrid and VPP technologies involves participation of the big number of DERs and implies their simultaneous functioning in real-time. Stable, reliable and responsive operation of grid members in providing services to the utility network, such as near-instantaneous feedback on grid's injection/consumption demand, is impossible without bidirectional, fast and compound interaction. In case of microgrid systems, the member's coordination has extended requirements due to the additional islanded mode of operation and periods of transition to and from this mode.

These objectives make a communication system one of the cornerstones in the developing of particular microgrid or VPP. Several aspects influence the design of the communication system, such as:

- Type and configuration of equipment in the system, where particular devices require different time of response, by that demanding separate arrangement of communication equipment and interfaces, along with each communication channel's bandwidth;
- Number of participants in the system and their physical allocation determine the type of connection needed, data traffic volume and maintenance costs [16];
- Control system, which can be centralized or decentralized, designates directly system's communication architecture. The centralized communication system gives advantage of simplification and unifying of system components, because all the processes executes by one common software without significant problems regarded to operation conflicts and schedules. On the other hand, the decentralized communication system allows independent functioning of various devices and sections, and can provide high rate of redundancy to the system [17].

4.2. Communication infrastructures

Although microgrids and VPPs are relatively recent appeared technologies, they utilize existing and well developed communication infrastructures for automation in power systems. Basic classification divides communication networks by the method of signal transmitting on wired and wireless. Depends on location and available financial resources microgrids and VPPs are implemented in both ways. The following section gives an overview of existing and widespread communication infrastructures with references to already functioning examples.

Power Line Communication (PLC)

PLC is the technology that uses electrical power lines as signal carriers. It appeared in early 1900s as a low data rate service for remote control of power network components. From the introduction of PLC and until present times several frequency ranges and various signal modulation techniques were used to achieve the data rates at first from a few bps to a few kbps with audio/low frequency bands and up to 200 Mbps with a high frequency range (3-20 MHz) [18].

Despite its quite narrow application in the early years, the technology becomes more important nowadays because of the potential in high speed connection through medium and low voltage power lines, 15/50 kV and 110/220 V respectively. The traditional application of PLC technology is a field level in communication systems or 'last mile' communication – connection of end devices to the common network or backbone. PLC gives benefits by its low costs for implementation and total coverage, because the power lines are installed everywhere in the electrical networks.

However, there are several disadvantages connected with the PLC technology:

- Lack of protection to noises from electrical motors, radio signals interference and power supplies. Additional constraints arise due to the fact, that power lines are not twisted and shielded, so there is a considerably high possibility of interception of critical data transmission by unauthorized parties;

- Connection interrupts when there is an open circuit on the power line section with switches, reclosers and insulators;

- Physical topology of the grid, impedance fluctuations and wave reflection at the terminal points cause high signal attenuation and distortion, which lead to loss of communication [19].

Fig. 4.1 shows the microgrid deployed in the Center for the Development of Renewable Energy Sources in Spain (CEDER). It consists of DG, DS, loads and prosumers (producers + consumers).

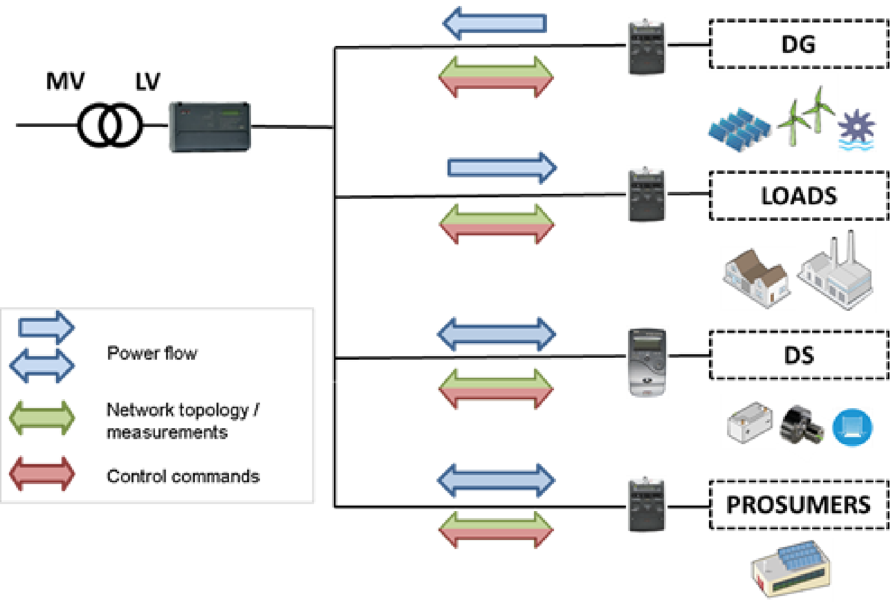


Figure 4.1 – Composition of CEDER microgrid communication and power system [20].

The microgrid comprises advanced metering system (AMS) consisting of 53 one-phase and three-phase smart meters and 9 concentrators for monitoring and management of generation, storage and load. The connection is established via PowerLine Intelligent Metering Evolution (PRIME) – open standard with the data model based on DLMS (Device Language Message Specification)/COSEM (Companion Specification for Energy Management) standard developed for electricity metering data exchange and later adopted by the International Electrotechnical Commission (IEC) in 62056 set of standards. The PRIME standard uses narrow band PLC technology based on Orthogonal Frequency Division Multiplexing (OFDM) modulation giving the advantage of robustness to interferences from the power grid [20].

Twisted Pair

The communication technologies based on twisted pair copper conductors have been used for many years evolving from the telephone lines signal carrier to the basic communication medium for local area networks (LAN) nowadays. The structure of the cable contains from one to several pairs of conductors with plastic insulation for each wire receiving and transmitting signals in the form of electrical current. In the single pair of wires, one is used for sending the signal to the receiver, and the second one is a ground reference. The difference between these two signals is used by the receiver on the other side.

Depending on the cable protection, twisted pair telecommunication cables are divided into an unshielded twisted pair (UTP), shielded and/or foiled twist pair (STP, FTP and S-FTP). The shield of the cable is implemented by braided mesh and metal foil, which covers each pair of conductors or the whole set of conductors. Shielded cables are used when there is a need for laying the communications in the electro-magnetic interference environment and by thus preventing the communication channel from the noise penetrations and cross talk.

Twisted pair cable is a cheap communication technology with limited distance capabilities and maximum up to 1.54 MHz channel capacity. Among the disadvantages are cable breakage and water ingress, difficulties in failure pinpointing, rise of ground potential due to power faults and lighting [21].

Although the technology is mostly applied in all types of LANs in commercial, industrial and residential areas, it is also widely used for automation in the energy sector as well as in other industries. Various automated systems in the power industry such as automated meter reading (AMR) systems and supervisory control and data acquisition (SCADA) systems by utilizing serial RS-232, RS-422 and RS-485 interfaces use twisted pair cables for communication on the field level.

Optic fiber communication

The communication systems based on optic fiber technology were introduced in 1960s and became a good alternative to conventional cooper-wired cables [19]. Among the general configurations of the optical fiber communication systems are:

- Passive Optical Network (PON) – provides point-to-multipoint network structure by utilizing optical splitters in order to establish the communication service for multiple customers by single wire;

- Wavelength Division Multiplexing (WDM) – uses cable bandwidth capacity to transmit several data streams simultaneously through the same fiber;

- Synchronous Optical Networking (SONET)/ Synchronous Digital Hierarchy (SDH) – technology for carrying the high capacity traffic by the time division multiplexing architecture [22].

Optic fiber cables give considerable advantages for automation in electrical systems by extremely high data flow rates (5, 10, 20 or 40 Gbps), invulnerability to Electro Magnetic Interference (EMI) and Radio Frequency Interference (RFI), long distance of data transmission with significantly smaller necessary amount of repeaters (every 100-1000 km). Listed features make the technology ideally suitable for backbone communication to the different SG applications and communication infrastructure for high voltage medium in electrical substations [19], [22]. Despite the obvious benefits for optic fiber, it has one main constraint in its very high cost for implementation. However, because of the huge bandwidth capacity, the technology is becoming viable by recovering the expenditure for its installation with big amount of consumers using one communication channel as a backbone. Therefore, the optic fiber infrastructure can serve as a highly reliable and fast communication [19].

One of the most remarkable existing examples of the technology deployment is the Fiber-optic Smart Grid in Chattanooga, Tennessee, USA. It is one of the first fully community-owned fiber-optic network installed by the city's Electric Power Board (EPB), which is used for Smart Grid applications together with triple-play media services (telephone, video and high speed internet).

The grid provides a number of various advantages such as improved energy efficiency, support of simultaneous uploading and downloading up to 1 Gbit of data, 40% in outage reductions due to implemented energy management for distribution system and intelligent switch technology, support capability of various data transferring devices (smart meters and appliances, demand response devices and distribution equipment) [23].

Satellite communication

The satellite communication provides transmission of the signals between two nodes via satellite. The process of the data transmission comprises the signal modulation and sending from the one point, amplifying by the satellite and sending back to the receiver on the earth. The technology supports the coverage between any communication points and data rate up to 1 Mbps.

The satellite communication is used for remote monitoring and control of electric substations and especially for the time synchronization based on global positioning system (GPS) technology [22]. The advantages of technology are global coverage, which permits establishing of connection between the nodes in a wide geographical area, and fast installation in comparison to wired technologies by only deploying the necessary equipment. Among the constrains are considerably long round-trip delay dependent on the satellite orbit altitude above the earth's surface, satellite channel characteristics that can be heavily influenced by weather conditions, the effect of fading and considerably high initial investments for satellite transceivers and equipment [19].

The Rio Grande Electric Monitors Remote Energy Assets Over Satellite project is one example of the technology implementation in Texas, USA. The Rio Grande Electric Cooperative (RGEC) in Brackettville, Texas, provides electric services across Texas and New Mexico states and owns 18 remotely located substations. The prerequisite for this project was the time of getting the maintenance crew to the substation for monitor and service purposes, which in some cases reached seven hours in one way. Another benefit for the satellite technology was the cost of the deployment in comparison to connecting remote substations via dial-up telephone link. The resulted communication system comprises satellite modem and Very Small Aperture Terminal (VSAT) satellite dish installed in each location and connected to the revenue electricity meter. The connection between the meter and satellite modem provided through the high-speed Ethernet (IEEE 802.3 standard) link. As a result, RGEC has well-established energy management system with installed monitoring software in the headquarters accessing utilities in real-time 24 hours per day with a high speed of response [24].

Wireless Communication

At the present time there is a big amount of different wireless technologies adopted for micrgorgid's and VPP's purposes. Basic classification divides the technology on short-range and long-range (cellular) wireless communication, which is described separately.

ZigBee technology provides short-range (up to 100 meters and up to 1.600 meters with ZigBee Pro) low-rate wireless communication for personal area networks and based on IEEE 802.15.4 standard. It uses unlicensed industrial, scientific and medical (ISM) bands and performs with the following data rates depending on the frequency: 20 kbps at 868 MHz, 40 kbps at 915 MHz and 250 kbps at 2.4 GHz. The technology supports different network topologies and applies for residential, commercial and industrial buildings automation, energy monitoring and AMR systems. The implementation costs and power consumption are quite low, which along with the high level of security make it a reliable communication solution. However, slow data rate, limited area coverage and the interference by parallel wireless networks (such as Wi-Fi) restrict deployment of ZigBee technology to in-home applications.

Wireless Local Area Network (WLAN) is a high-speed technology for wireless Internet and network communication based on IEEE 802.11 series of standards and commonly known as Wi-Fi. It provides data rates from 2 to 600 Mbps and operates on ISM frequency bands, such as 2.4 GHZ, 3.6 GHz and 5 GHz. Wi-Fi gives advantages in fast, secure and reliable connection, but short operation range (up to 100 meters), high cost for deployment and high power consumption constrain the implementation of WLAN technology to primarily residential and commercial local networks.

Wireless Mesh is an economically viable, reliable and scalable wireless network consisting of a large number of nodes including routers and mesh clients. This type of network uses dynamic routing, where every single communication point has a possibility to work as a repeater, transmitting in such a way data packets from one node to another. In the case of outage of one node, all others are capable to communicate through the rest of the points because of interconnected structure of the network. Wireless mesh networks exploit various standards such as IEEE 802.11, 802.15 and 802.16 and are able to cover large territories by using multiple radio-wave reflection technology for routing. Because of the benefits in high coverage, robustness and self-healing, wireless mesh has found its application in home automation and AMR systems. Along with that, the technology has limitations in low-speed data rates and vulnerability to the interference from other wireless networks.

Z-wave is a low-cost, short range and low power consumption wireless network technology. It uses 900 MHz ISM frequency and can provide 40 kbps data rate up to 30 meters. The technology applies in home automation for residential and commercial light utilities. Because of its low power requirements it can be used in smart grid application [22].

Cellular communication

The cellular technology is based on the radio network with a big amount of transmitters, which create cells and reuses frequencies form both coverage and data transmission. Cellular networks work in 850, 900, 1800 and 1900 MHz frequencies. The common classification separates cellular communication by generations of technology development: 1G, 2G (Groupe Special Mobile originally or Global System for Mobile communications - GSM), 2.5G (General Packet Radio Service – GPRS and Enhanced Data rates for GSM Evolution - EDGE), 3G (Universal Mobile Telecommunications System - UMTS), 3.5G (High Speed Packet Access - HSPA) and 4G (WiMAX and Long-Term Evolution LTE).

Cellular networks are widely employed in AMR systems worldwide by gathering the data from remote metering points through GSM/GPRS-modems in off-line or real-time modes. Additionally, 3G technology is regarded as one possible solution for SG applications because of its fast data rate, wide coverage, low latency and security. One of the examples is the cooperation between Siemens and RWE Deutschland AG resulting in implemented energy management system (EMS) based on GPRS modems for RWE VPP in Germany [25].

Constrains in cellular communication can arise due to sharing the network with mobile users, leading to congestions and lower network performance [22].

WiMAX (Worldwide interoperability for Microwave Access) is a 4G generation of cellular network communication based on IEEE 802.16 series of standards. WiMAX works in 2.3, 2.5, 3.3 and 3.5 GHz frequency bands together with unlicensed 5.8 GHz frequency giving up to 75 Mbps data rate on the 50 km operating range. Due to its small latency (10-50 ms) and the capability of providing communication to multiple users by exploiting just one base station the technology is applicable to real-time SG and substations automation applications [22]. The WiMAX is especially suits for

SCADA systems on electric substations in big industrial manufacturing facilities, where there is no possibility to provide wired communication. Despite its fast communication and wide coverage, the technology is costly in terms of initial investments, demands high power supply and can be influenced by severe weather conditions.

4.3. Communication network architectures

The organization of communication networks for microgrids and VPPs as building blocks for smart grids has a hierarchical structure and consists of three main layers according to grid's electrical generation, transmission and distribution systems. Fig. 4.2 shows the structure of the communication networks from bulk generation to customer premises.

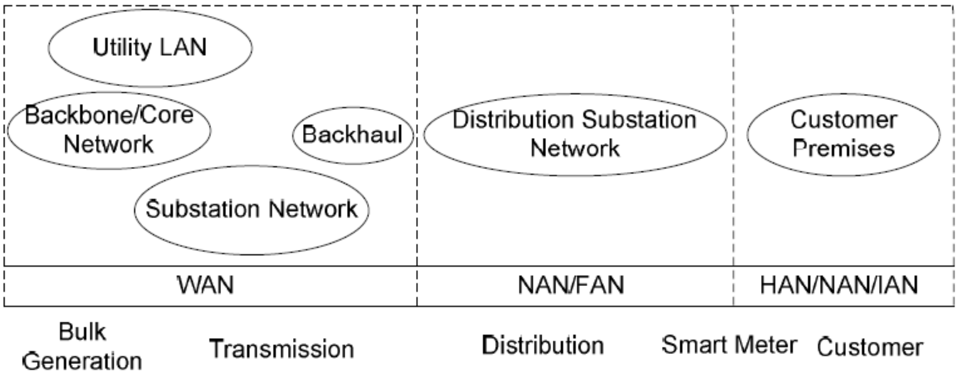


Figure 4.2 – Multi-layer hierarchical structure of communication networks for SG [22].

WAN

Wide Area Network (WAN) serves as a backbone for smart grid and provides communication for the control center with transmission and distribution substations. WAN comprises a big number of communication nodes, including smart meters, remote terminal units (RTU), phasor measurement units (PMU) and other sensors for remote automation purposes, EMS and SCADA systems. To ensure fast, broadband and real-time communication without considerable latency required for WAN, the optic fiber communication is mostly applied. Another option can be in WiMAX technology due to its relatively long distance coverage and high data flow rate.

NAN/FAN

Neighborhood Area Network/Field Area Network (NAN/FAN) is responsible for the distribution substations level as well as for control of information flows between WAN and customer premises networks. This type of network provides a large number of SG services, such as smart metering from customer site to control center, load management, distribution automation and others. Typical data flow rate can varies from 100 kbps to 10 Mbps covering the distance between 100 m and 10 km. Various communication infrastructures can be used dependent on the application of each particular network including ZigBee, WLAN, PLC, twisted pair and other wired and wireless technologies dependent on funding.

Customer Premises Area Network

There are different types of customer premises networks such as home are network (HAN), neighborhood area network (NAN), industrial area network (IAN) or building area network (BAN). These kinds of networks gather various types of devices dependent on application into microgrids and interact with FAN and WAN on purpose. HAN mostly includes home automation appliances and smart meters, but can also be composed from small energy sources and storages. IAN and

BAN are more complicated networks with large number of control devices and sensors for building and industrial EMS and SCADA systems. Customer premises networks require low data rate and power consumption, scalability and security of connection, which gives possibilities for implementation of various communication technologies [22].

According to the number and type of participants, microgrids and VPPs can be gathered in different types of networks including customer premises area networks as well as NANs or FANs.

4.4. Communication protocols

The most significant feature that allows reliable functioning of SG components is information exchange. For the operation of microgrids and VPPs the necessary data is needed regarding power system current state, its historic information and its applications. All this information is provided by communication system, which makes communication standards and protocols especially important. The communication system arranges the information exchange between different grid members, such as substations equipment, DERs and control centers through the common frame of regulations for data format and transmission. Different grid applications have different constrains in terms of necessary communication, which is resulted in various communication protocols existing for data exchange in power industry. For instance, protection devices require much faster response and real-time communication than monitoring applications. On the other hand, communication channels for customer metering infrastructure have to be more protected than substation monitoring devices [26].

The given subsection highlights the most commonly used and adopted communication protocols in microgrids and VPPs. The protocols can be distinguished between specifically developed for SG applications, such as Modbus, DNP3 and IEC 61850 series of standards, and those, who have a big importance but initially were designed for other purposes, such as Internet Protocol Suite.

Internet Protocol Suite

The Internet Protocol Suite (IPS) is a set of protocols for providing of the Internet services. The following are mostly used in SG applications: Network Timing Protocol (NTP) for time synchronization, Internet Protocol (IP), Transmission Control Protocol (TCP), User Datagram Protocol (UDP), File Transfer Protocol (FTP) and Simple Mail Transfer Protocol (SMTP). The protocols that supply transport and network services, such as TCP, UDP and IP, are of the highest importance.

The IP protocol sends individual packets of data (datagrams) from one node to another carrying the addresses of sender and receiver in the datagram header as 32-bit fields. It fragments data into smaller packages for transmitting purposes and reassembles them at the receiver end. However, the data delivering is a concern for higher layers and IP does not guarantee it by only checking the header sum. Besides that, IP does not establish or monitor the connection on the channel and just sends each datagram independently.

TCP utilizes functionality of IP with establishing of communication links between each end, maintains error-free transmission and makes possible the bidirectional communication. It fragments the data into sections and conveys them to IP, which further cuts those to smaller datagrams.

UDP also uses IP services additionally providing information about sender and receiver ports, and a checksum. Unlike TCP, UDP discards data packets with detected error, does not supervise reliable data delivering and it is not connection oriented. By utilizing IP broadcast addressing mode, UDP allows the transmission of data to several destinations. It applies for real-time systems where the smallest delay of connection is of the highest importance.

IPS is used for carrying such specific automation protocols as Modbus, DNP3 and IEC 61850 over a network environment by conveniently utilizing Ethernet data link. These protocols are encapsulated in the TCP/IP stack and act over its layers. This practice gives advantages in better utilization of the dedicated protocols for SG applications by modifying them specially for using with IPS as a standard for transmission over the network [27]. Another application of IPS in microgrids and VPPs is the data communication between terminal servers, data logging servers and control center [16].

Modbus

The Modbus protocol is originally designed for data exchange between PLCs. It is openly published and from 2004 controlled and maintained by the community of vendors and users of the automation equipment Modbus-IDA. The protocol works at application level with a foundation on a client/server architecture, where the client requests server's operation. By using master multiplexing in gateway based network for gathering diverse communication interfaces, Modbus is capable for routing of different system configurations with not only one master controlling all the slave devices [28]. Fig. 4.3 shows the layered structure of Modbus.

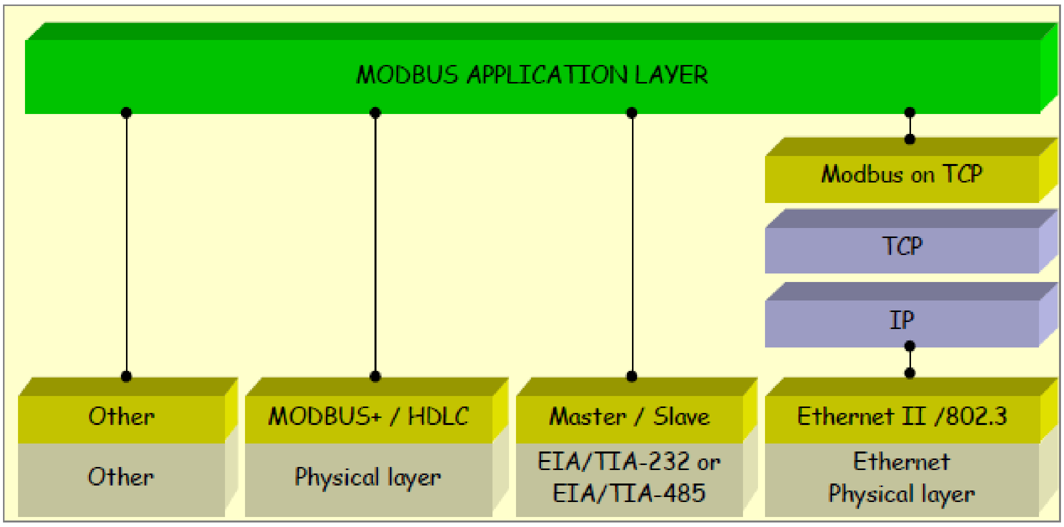


Figure 4.3 – The structure of Modbus protocol layers [29].

Modbus can be deployed over several communication interfaces, such as:

- TCP/IP over Ethernet – the data is framed into binary format in TCP using Ethernet protocol. Addressing and channel access control mechanisms are provided by Carrier Sense Multiple Access with Collision Detection (CSMA-CD);
- Serial Transmission – the data with serial communication can be transmitted in 8-bit binary format (Modbus RTU) or in 7-bit ASCII format (Modbus ASCII), which requires twice higher data traffic. This mode can be implemented over various communication mediums, such as wired, optic fiber or radio. The most commonly used standards in wired communication for serial transmission are RS-232, RS-422 and RS-485;
- Modbus Plus – the advanced technology for high data transfer with extra addressing, routing, data consistency and diagnostic features [28].

DNP3

The Distributed Network Protocol version 3.3 (DNP3) is an open standard for telecommunication designed for interaction between master stations, RTUs and other intelligent electronic devices (IEDs) in electrical utilities and industrial environments, such as oil & gas, water/waste and security. It was designed for SCADA systems to transmit considerably small data packets in an interruptible manner with a predefined sequence [27].

Fig. 4.4 illustrates four operation topologies that can be achieved utilizing DNP3 protocol with two types of devices: masters – central stations with power processing and data storage in electrical utilities, and slaves – remote or out- stations with a main function to collect the data from sensors in the grid and send it to the central station [30].

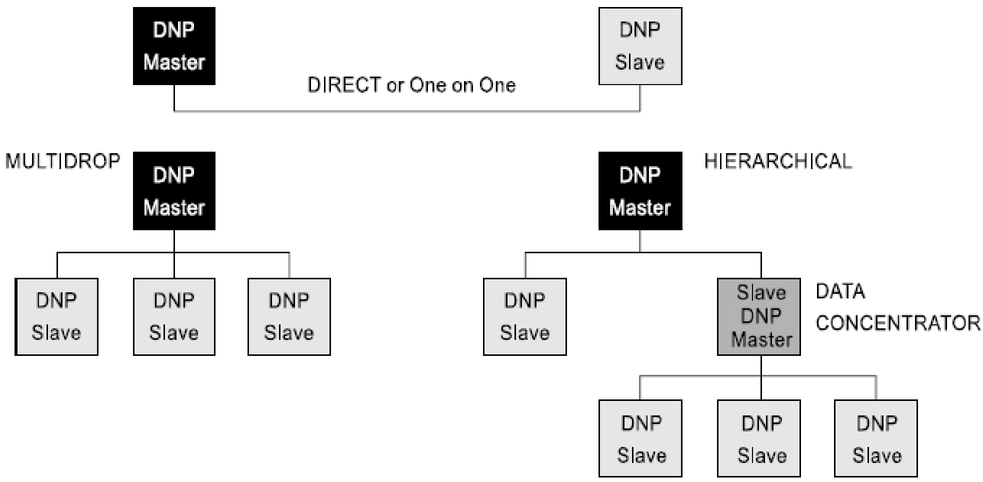


Figure 4.4 – DNP3 network topologies [27].

Direct or One on One is a straightforward point-to-point model where one central station interacts with one remote station.

In Multidrop topology, one master device communicates with several slaves by polling the data. Each remote station receives the request messages but sends the data to the master only when the destination address corresponds to its address.

The Hierarchical system comprises one master and several slave devices, which also can be a master device to other outstations in a lower layer. The last possible topology with utilization of DNP3 protocol is Data Concentrator, where several remote stations can have different protocols and by using gateways the central station can collect information from them [30].

DNP3 supports two modes of operation – poll and quiescent, or reporting by exception, meaning that master device does not send any requests to check if the communication with the remote station is stable. When the state of the channel is changed, the slave devise sends ‘unsolicited response’, and without any troubles the system remains in silent state. This mode of operation contributes to full bandwidth capacity utilization of the communication channel.

The above described modes of operation imply the initiation of communication performed by slave devices, which is referred as balance communication. However, only master devices are able to request data or send commands to other devices in the DNP3 supported networks [27].

As in the case of Modbus protocol, DNP3 is typically deployed on serial communication (RS-232 and RS-485) over various physical links, such as twisted pair, optic fiber, radio and satellite communication, and in some cases is encapsulated in TCP/IP protocol on Ethernet networks.

IEC 61850

IEC 61850 is a set of standards of IEC Technical Committee 57 (TC57) for electrical substations automation systems. The main objectives for developing a new standard in the electrical power industry were recent communication technologies improvements and the goal of common, open and multivendor communication protocol which would be able to significantly expand automation systems capabilities.

The IEC 61850 defines various aspects of substations communication system along with a number of several related standards for various DG applications, communication between substations and the industrial automation. The standard is organized in 10 main sections with few subsections performed as separate IEC 61850 documents.

The main communication architecture concept in IEC 61850 is the creating of data objects and services independent of any particular protocol, or 'abstracting' them. This allows further mapping of the data objects and services to any other protocol meeting the data/service requirements. For the purpose of building of large and abstract data objects the concept of 'Common Data Classes' was defined. The standard also determines the ways of mapping the abstract data and services into the Manufacturing Messaging Specification (MMS) standard and sampled measured values into the Ethernet data frame with point-to-point and multipoint, unidirectional and bidirectional communication respectively. For the explicit representation of the relations between automation system and substation itself the XML (Extensible Markup Language) based Substation Configuration Language (SCL) was developed, so that each device has to provide its own SCL file with the configuration description.

Unlike legacy communication protocols, which specify the format of bytes transmitting over the wire, IEC 61850 additionally gives the extensive model for the organization of data consistent with all vendors and devices in the power system. This feature reduces significantly the efforts needed for the engineers for devices configuring. It can be done simply by importing the SCL configuration file into the device, and then IEC 61850 client application can retrieve the object characteristics from the device over the network.

IEC 61850 provides a variety of advantages over the proprietary protocols including interoperability of devices from different suppliers, lower installation, configuration and maintenance costs, enhanced scalability and possibility for further improvements of systems automation processes [31]. It was originally developed for electrical substations LANs, so it mostly employs TCP/IP protocol and Ethernet link as a communication medium.

4.5. Preferred communication protocol

From the early years of substation automation the legacy communication protocols were designed to provide all necessary functionalities of power system devices along with carrying the minimum possible data rates for its operation. The latter demand for the protocols was dictated by the restricted communication channels bandwidth capabilities inside the substations [31]. Since the first half of 1990's, the technological advances in communication systems began to surpass the power system technologies, which caused the shift from the proprietary communication protocols to open and more accessible standards for achieving better automation systems solutions and interoperability with various equipment manufacturers [32]. With appearing of the Ethernet technology and TCP/IP network protocol, the serial protocols were adopted for operation over this communication medium [31]. Although such legacy protocols as Modbus and DNP3 in most of the part fulfilled new technological demands, the necessity for more advanced object oriented modeling structure appropriate for multi-vendor environment has led to development of a new communication standard for automation in power systems – IEC 61850 [32].

In case of legacy protocols, DNP3 gives more alternatives for topology operation modes in comparison to Modbus, which works in traditional client-server mode and has no possibility for response by exception mode (only with Modbus TCP). DNP3 protocol additionally provides multipoint and hierarchical topologies along with the capability to work as a data concentrator. Both these protocols can be chosen for operation with a small data rates, such as 1200-9600 bps. If the communication medium is selected to be deployed over Ethernet data frame, IEC 61850 is an obvious choice because of its original design specifically developed for this communication technology.

The main features, which make IEC 61850 standard more advanced communication protocol for SG applications over Modbus and DNP3, are:

- Provision of nearly 100 logical node classes with more than 2000 data objects and attributes;

- It can be extended easier than legacy protocols;

- Unlike indexed addressing in legacy protocols, IEC 61850 employs hierarchical names;

- The data (logical nodes, objects and attributes) have much detailed description;

- Better flexibility in control of parameter setting, choosing the data for reporting, designating the communication control objects and modifying reporting/logging behavior compared to legacy protocols;

- Transmission of substation events through Generic Object Oriented Substation Event (GOOSE) controlled model mechanism and sampled values;

- Access to the entire information hierarchy through the obtaining of directory, which is not available with DNP3 and Modbus;

- Full description of devise configuration in XML format;

- Support of manufacturer independent engineering development tools [32];

Despite its relatively complicated implementation routine compared to legacy protocols, the IEC 61850 standard is considered to be the protocol of future power industry communication systems and increasingly more and more vendors produce their new equipment IEC 61850 compatible.

The following sections of this project provide detailed description for the Modbus and IEC 61850 protocols and their application for communication system in the laboratory microgrid.

5. Laboratory microgrid test based on Modbus communication

5.1. Communication system setup for laboratory microgrid

The given section describes hardware and software tools applied for the creation of communication system in Sulkowski laboratory microgrid. The system consists of the following equipment available for communication in the laboratory: back-to-back AC-DC-AC power electronics converter with the control panel mounted into the rack cabinet; dSPACE Connector/LED Combi Panel for DS1104 R&D Controller Board connected to back-to-back converter; PowerLogic G3200 Modbus-to-IEC 61850 server manufactured by Schneider Electric; Ethernet network communication switch and rack mount test PC. Both the PC and dSPACE panel are mounted on the same rack with the converter. The connection diagram is shown in Fig. 5.1.

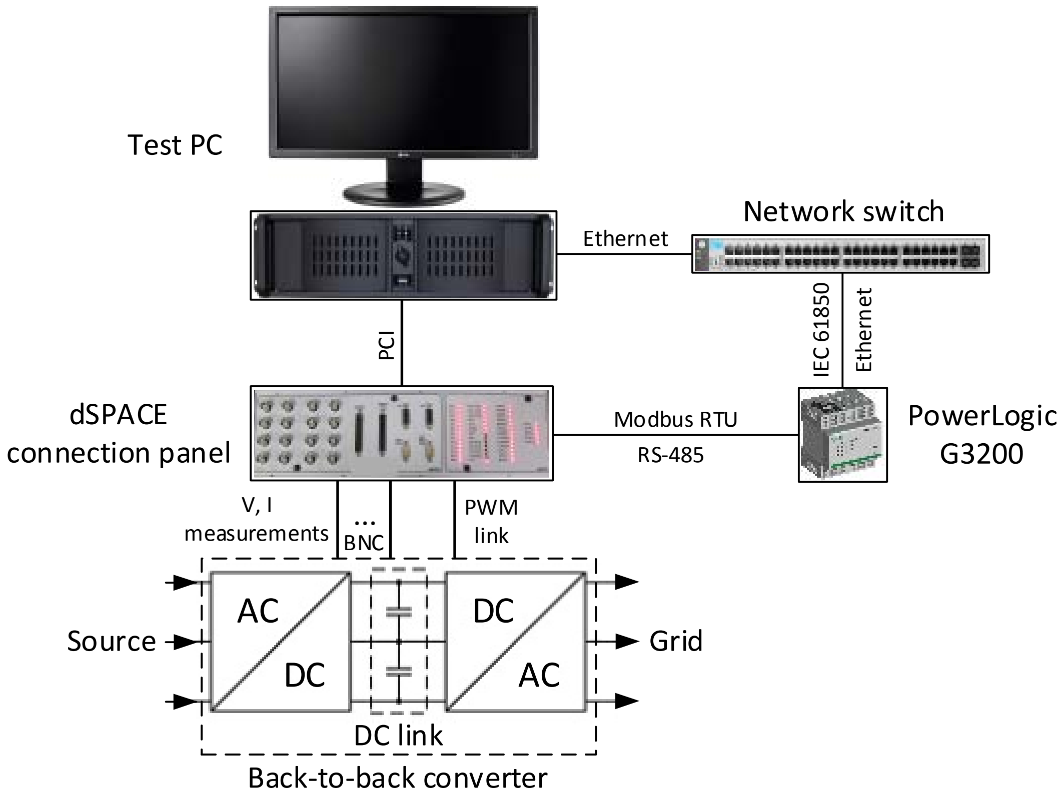


Figure 5.1 – Modbus over RS-485 communication equipment configuration.

The dSPACE panel is connected to converter control panel's voltage and current measurement outputs through ADC (analog to digital converter) inputs by BNC (Bayonet Neill-Concelman) connectors along with PWM (pulse width modulation) link. At the present connection configuration, it is possible to measure converter's voltages V_a, V_b and V_c together with currents I_a and I_b of Inverter 1 side and voltage V_{dc} of DC-side. In order to communicate with Test PC, dSPACE panel has a PCI connection, RS-232 and RS-422/485 serial transmission channels.

The PowerLogic G3200 is a simultaneously interface and communication protocol converter, which can connect Modbus devices through RS-485 interface from the field level to IEC 61850 clients over Ethernet on the substation monitoring and control level. The converter gives benefits in possibility of connection the third party Modbus devices, in this project it is back-to-back power electronics converter linked with dSPACE Connector/LED Combi Panel for DS1104 R&D Controller Board. The G3200 has to be configured for communication with XML-based CID (Configured IED Description) file based on SCL, which is defined in IEC 61850-6 standard. The process of creating and uploading of the CID file to G3200 for communication through IEC 61850 will be described in the next sections.

5.2. Modbus protocol description

As it was mentioned in the previous section, the Modbus protocol was established for communication between controllers of the company Modicon in 1979. The protocol determines a message format which is understandable by controllers whatever communication medium is used. It specifies the processes of requesting, responding and error checking for PLCs communication. During the communication session in Modbus network, the protocol stipulates in which way the controller is assigned to its particular address, how it will distinguish and understand the message sent to it and decide the necessary action to be performed or data to be written or red from it. In case if the response is required, the PLC will compose the message and send it over Modbus. The protocol describes the common frame for the data mapping and content of message fields.

In the Modbus network with serial communication, PLCs interact with each other through the master-slave model, where only one controller, the master, can launch communication process by sending request messages to other devices (slaves) that are usually called ‘polls’ or ‘queries’. The slaves respond with requested data from the query message or execute necessary action given in the request. The master device has the possibility to poll a single slave or to send broadcast request to all devices in the network. Slaves, in turn, provide the response to master in case of individual query and do not respond with any messages back to broadcast poll if the slave address in the request does not coincide.

Fig. 5.2 shows the request-response cycle of Modbus protocol communication.

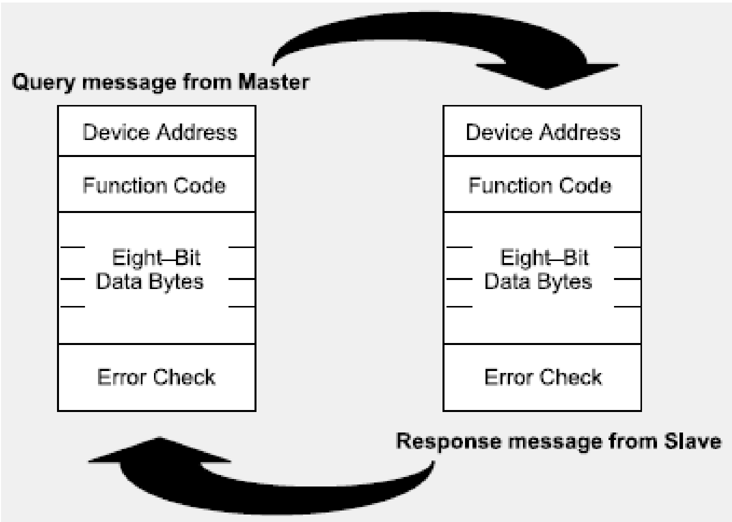


Figure 5.2 – Modbus Master-Slave request-response cycle [33].

The query contains slave device address, specific Modbus function code indicating the type of action is needed to perform by slave device, data bytes with additional information to already defined function code and error check for the message integrity validation. The response message has the same structure as a request and contains an echo of function code, necessary data specified to send back to master device and error check. In case of the error in query, the response will have an error function code and specific data in the message to inform the master about this error (also called exception response). The error check of exception response gives confirmation to master, that the message is legal.

The Modbus protocol is implemented over serial network transmission by two modes: RTU and ASCII. Both of them have the same principle but a slightly different implementation in terms of data packing into message fields and decoding [33]. The Modbus ASCII is used quite seldom nowadays in the industry, so the RTU mode is selected for implementation in this project.

In the RTU mode Modbus message every byte consists of two 4-bits hexadecimal characters, which gives larger characters density and two time larger data traffic with the same baud rate in comparison to ASCII mode. Every message is sent in a continuous stream. Table 5.1 lists main specifications of Modbus RTU protocol [33].

Table 5.1 – Modbus RTU protocol specification.

Coding system	8-bit binary, hexadecimal 0-9, A-F Two hexadecimal characters in each 8-bit field
Bits per Byte	1 start bit 8 data bits, least significant bit sent first 1 bit for even/odd parity; no bit for no parity 1 stop bit if parity is used; 2 bits if no parity
Error check field	Cyclical Redundancy Check (CRC)

Each request/response message is located by the transmitter into specific frame with determined beginning and end of this frame, giving the receiver the layout for correct understanding and decoding the message. In the RTU mode, transmission begins and ends with the interval of silence with at least 3.5 characters long (usually performed as a multiple of character times baud rate used for transmission). In case of the silence interval more than 1.5 characters in the middle of the message transfer or less than 3.5 characters before the beginning of the next message the transmission error will occur. The first byte of the message contains the device address in the network, second byte – specific Modbus function to execute, next bytes involves additional information regarding the defined function and last two bytes contain CRC check. Table 5.2 depicts Modbus RTU message frame.

Table 5.2 - Modbus RTU message frame [33].

Start	**Address**	**Function Code**	**Data**	**CRC Check**	**End**
≥ 3.5 char	1 Byte	1 Byte	N×1 Byte	2 Bytes	≥ 3.5 char

For correct and appropriate performance of the Modbus protocol, the controllers should apply particular data model with a sequence of tables with certain characteristics defined for the protocol functionality. Table 5.3 gives the characteristics of the four main data tables used in Modbus protocol data model.

Table 5.3 – Modbus data model [29].

Primary tables	Object type	Type of	Comments
Discretes Input	Single bit	Read-Only	This type of data can be provided by an I/O system.
Coils	Single bit	Read-Write	This type of data can be alterable by an application program.
Input Registers	16-bit word	Read-Only	This type of data can be provided by an I/O system
Holding Registers	16-bit word	Read-Write	This type of data can be alterable by an application program.

The protocol permits access up to 65536 individual data items for any of primary tables. In order to read or write data, or perform some action, Modbus provides 127 function codes divided into three groups: public, user-defined and reserved (used by some companies in their particular products and are not accessible for common use). Fig. 5.3 shows Modbus function codes categories.

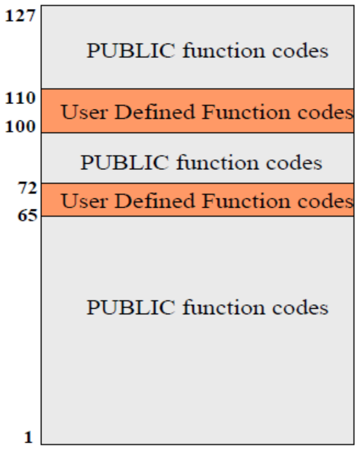


Figure 5.3 – Modbus function codes categories [29].

The most frequently used functions for basic Modbus operation are the public function codes, such as 01 – Read Coils, 02 – Read Discrete Inputs, 03 – Read Holding Registers, 04 – Read Input Registers, 05 – Write Single Coil, 06 – Write Single Register, 07 – Read Exception Status, 08 – Diagnostic and so on.

5.3. Hardware and software resources for Modbus device model development and verification

The hardware used for the development of Modbus device model and simulation test comprises the following equipment: back-to-back power electronics converter with the control panel; dSPACE DS1104 connection panel and Test PC. In this configuration RS-232 UART (universal asynchronous receiver-transmitter) port of DS1104 panel is used, the connection diagram is shown in Fig. 5.4.

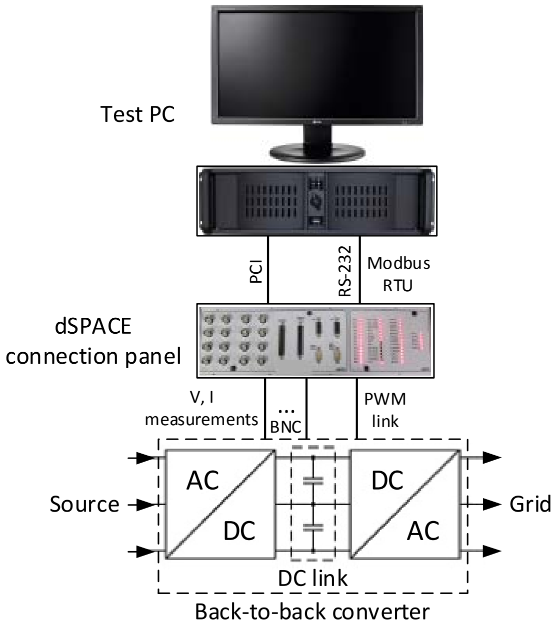


Figure 5.4 – Modbus simulation test hardware.

Test PC has installed MATLAB/Simulink software synchronized with dSPACE connection panel elements, which allows performing of simulation tests with real hardware components. Additionally, the PC has dedicated dSPACE panel software dSPACE Control Desk 5.0 used in this test for verification and monitoring of transferred signals to the panel's serial port receiver. The last component used in this simulation test is Modbus master simulator software Modbus Poll of Danish company Witte Software. The main purpose of this program is to create a physical Modbus network with master device (Modbus Poll) and slave device modelled with the aid of Simulink library tools comprising dSPACE RS-232 port for communication.

5.4. MATLAB/Simulink Modbus test model

The block diagram of developed Simulink Modbus device test model with main blocks and functions is shown in Fig. 5.5.

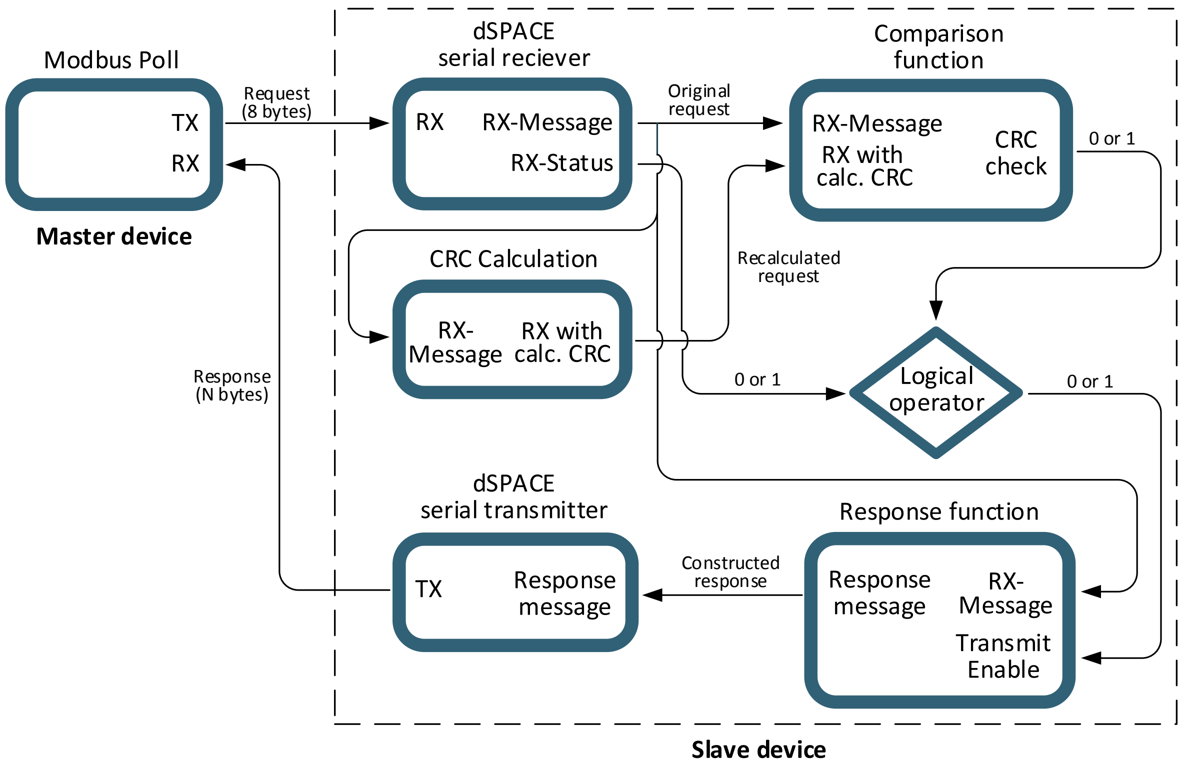


Figure 5.5 – The block diagram of Simulink model for Modbus protocol test.

The Simulink model consists of two main logical parts – Modbus master and Modbus slave devices (the model is shown in Appendix A). The Modbus master query message is implemented by simple 8-byte length block starting the transmission through Modbus Poll software, which forms the user request and it always has the same length of 8 bytes.

The slave device is constructed from the following blocks and functions: dSPACE serial receiver block DS1104SER_RX; CRC calculation function; CRC check comparison function; logical AND operator; Modbus response function and dSPACE serial transmitter block DS1104SER_TX.

DS1104SER_RX block receives the query message from Modbus Poll and outputs received message RX-Message to CRC check comparison function, CRC calculation function and Response function along with transmission status RX-Status to logical AND operator.

The CRC calculation function receives the request message from DS1104SER_RX block, takes first 6 bytes of the message, recalculates CRC [34] and adds it to the original message. The output of this function then goes to CRC check comparison function to verify the identity of received and appended with recalculated CRC check messages. In such a way the function of error check of the device communication through Modbus RTU protocol is implemented.

The output of CRC check comparison function in form of 0 or 1 transmits to the logical AND operator, which is together with RX-Status output (again 0 or 1) decides whether to send the response message to master device or not. The RX-Status output is required for the transmission block to respond only when the request is received.

In case of received 1 from logical AND operator, the Modbus response function, implemented in Simulink as Triggered Subsystem, constructs the response to the master request based on received message from DS1104SER_RX block. The flow chart of the response function is shown in Fig. 5.6.

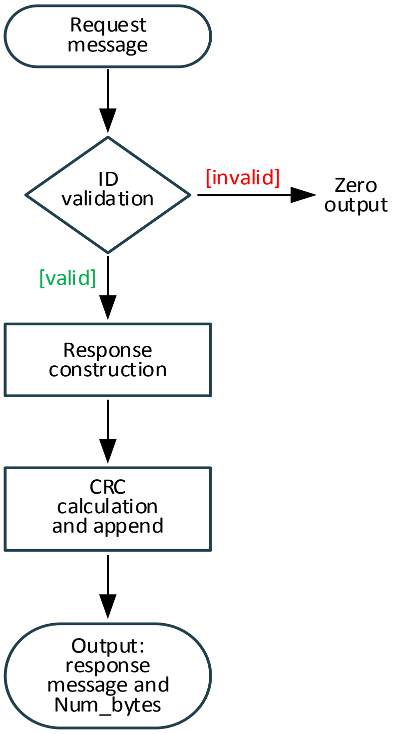


Figure 5.6 – Modbus response function flow chart diagram.

The Modus communication network model is designed to have a master and one slave device with its ID address 01. Therefore, the response function validates the correct ID of slave device sent in the request message and forms response in case of verified address. The function is developed to provide response for the 03 Modbus function code – Read Holding Registers. The data used for providing the output for this particular command consists of array of 20 random numbers in order to respond for up to 10 registers request:

```
[1 12 33 45 58 61 72 84 93 10 14 28 36 44 57 62 73 81 93 15]
```

The function takes the starting register and amount of registers necessary to be read, which is given in the query, and then constructs the response, calculates CRC check and appends it to the output. The second output of the function is the number of bytes Num_bytes (is taken from the third byte of the response message with addition of 5 corresponding to the number of bytes should be transmitted: 1 – ID, 2 – Function code, 3 – Number of bytes, N – Defined amount of bytes in the byte number 3, N+1 and N+2 – CRC check) required for the input to DS1104SER_TX block. The Modbus response function is presented in Appendix B. Finally, the DS1104SER_TX block transmits the response message to Modbus Poll receiver and the received information is visualized in the master device software interface.

5.5. Modbus model simulation tests over RS-232 communication

To begin the simulation test of the above described model, required serial ports transmission adjustments was performed both for master and slave devices. Fig. 5.7 shows the connection setup for master device over serial port.

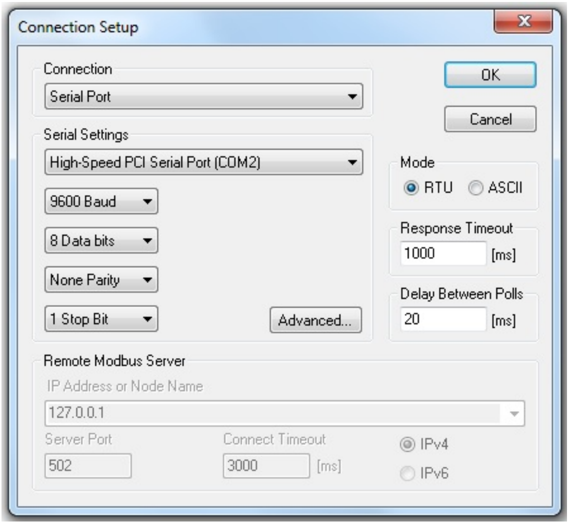


Figure 5.7 – Connection setup of Modbus Poll software.

The main settings are Baud rate (9600), Beats per byte (8 data bits), Parity (None), Number of stop bits (1). Same settings should be selected for the slave device for operation of Modbus network, which is shown in Fig. 5.8.

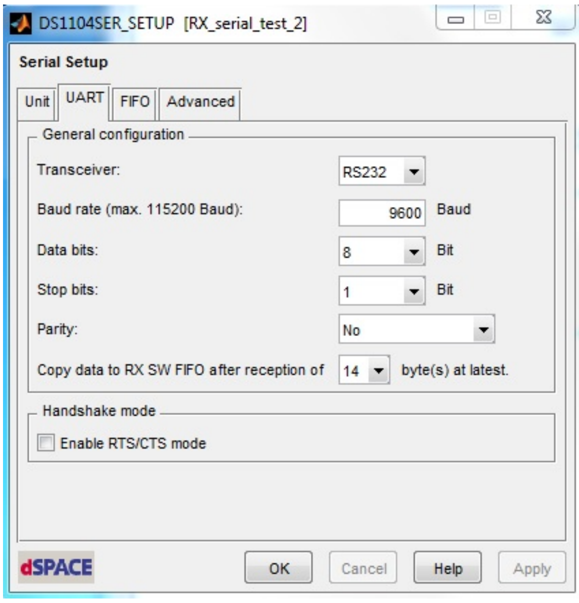


Figure 5.8 – DS1104SER block setup.

Additionally, the maximum number of received and transmitted bytes for DS1104SER is specified, which is shown in Fig. 5.9. The number of bytes to be received is always 8 because of fixed length of Modbus RTU query, and the maximum bytes to be sent is 25 by the longest response for the 03 function code to read 10 holding registers.

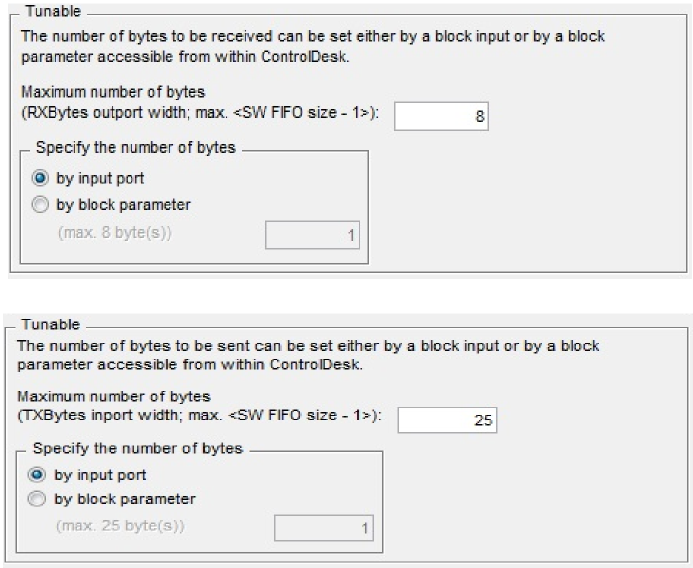


Figure 5.9 – DS1104SER setup for data receiving and transmitting.

The process of communication in the modelled Modbus network starts with pre-configuring of master device request setup. This includes slave ID address, necessary function code, starting register and the amount of registers to read, and finally the interval between consequent polls transmitting from Modbus master device. The adjustments of Modbus Poll communication session for function code 03, starting from register 1 to read 10 registers, are shown in Fig. 5.10.

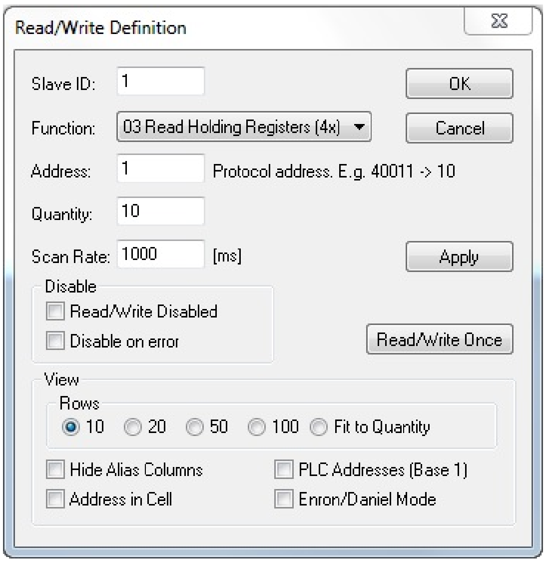


Figure 5.10 – Modbus Poll communication session setup for 03 function code (10 registers).

The response message to this request consists of 25 bytes in form of hexadecimal numbers beginning with slave ID – 01, function code to execute – 03, byte count – 14 (when converted to decimal is 20 – each register contains 2 hexadecimal numbers, and overall amount is 10) and two last bytes are CRC check. Therefore, taking into account the array of random numbers given in the previous subsection, first two registers will be represented in the received message as 01 0C and 21 2D, converting these two values into decimal numbers gives 268 and 8493. These two numbers are constructed from the first four members of array, which are 1, 12, 33 and 45. The request and received messages for this test in form of byte sequences in hexadecimal numbers are shown in communication traffic of Modbus Poll software in Fig. 5.11.

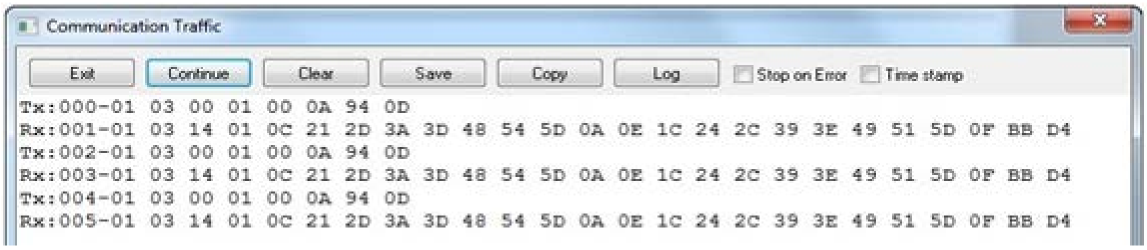


Figure 5.11 – Communication traffic for Modbus RTU test of 03 function code (10 registers).

Received data from the slave device for 10 holding registers is shown in separate test window of Modbus Poll software in Fig. 5.12.

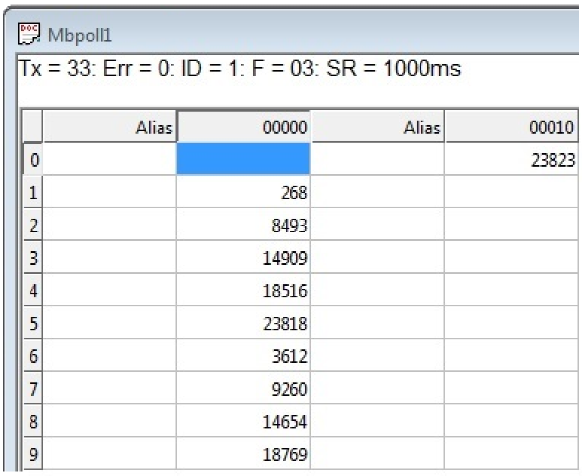


Figure 5.12 – Received data for 10 holding registers for Modbus RTU test of 03 function code.

Additional test was performed for the same function code but this time starting from the third register and read of six registers. The Modbus Poll communication session setup, communication traffic and received data from the slave device model in Simulink are shown in Fig. 5.13-5.15 respectively.

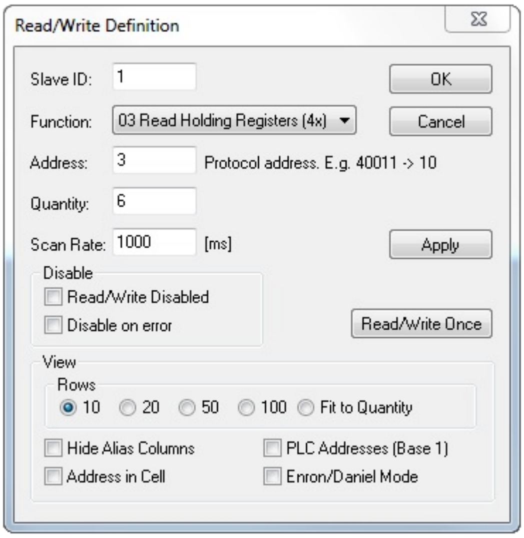


Figure 5.13 – Modbus Poll communication session setup for 03 function code (6 registers).

In this test reading of 6 registers means that response message will consist of three first bytes including the byte count, 12 bytes for data transfer and two final bytes for CRC check – overall 17 bytes.

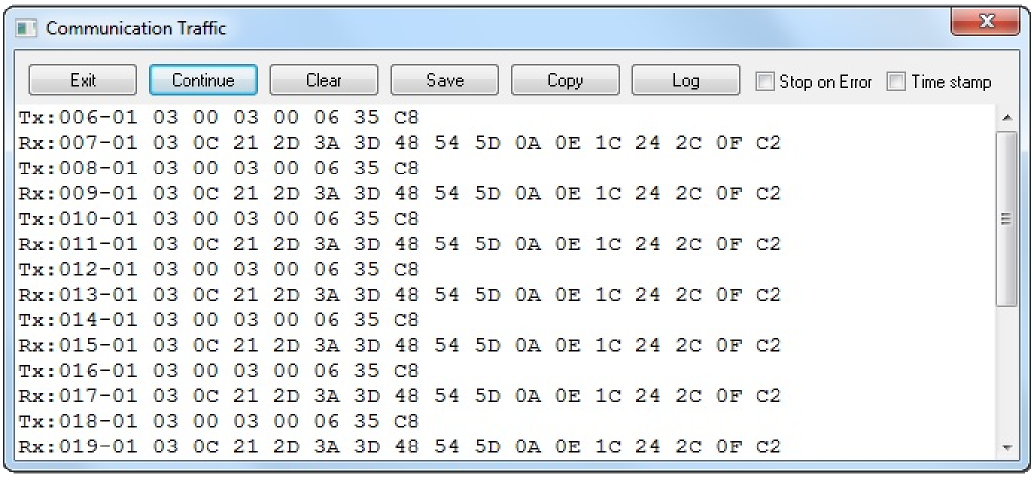


Figure 5.14 – Communication traffic for Modbus RTU test of 03 function code (6 registers).

Taking into account that the first register need to be red is 3, register 1 and 2 in the received data table are empty. Additionally, it can be seen that the value for register 3 corresponds to that one from register 2 in the previous test (Fig. 5.15). This shift is explained by the code for construction the buffer for response message (Appendix B). The response function is programmed to parse the message after the byte count (which is always the byte number 3) starting from the first value given in the array of random numbers. This is implemented in the following lines in the code:

```
for j=1:addr_range
        buffer(3+j) = data(addr+j-1);
    end
```

Here the variable 'addr' is a value received from the request message byte number 4. In case of starting address number 1 the additional -1 operation should be added to begin the construction of response message from the first value in the array. In case of the following registers, the parsing of the message for address number 2 and so on starts from the corresponding second number of the array, and the final values for the register will be different depending on the starting address requested. In case of the first register to be red is number 3, the first number for register 3 is 33 and the second is 45, giving the value 8493 after converting from hexadecimal numbers back to decimal.

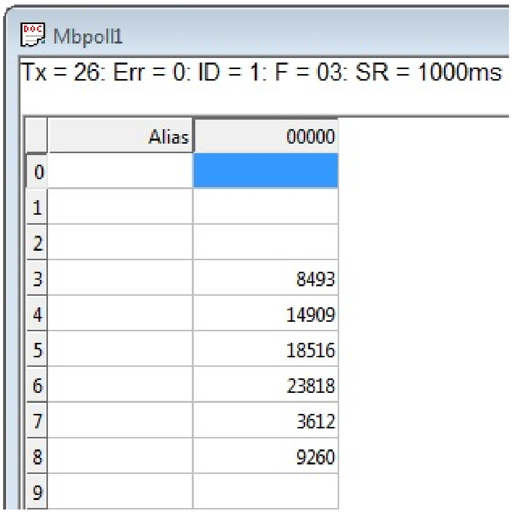


Figure 5.15 – Received data for 6 holding registers for Modbus RTU test of 03 function code.

5.6. Modbus communication over RS-485 serial connection

After the Modbus device model is created and verified the laboratory microgrid communication system presented in Fig. 5.1 can be tested. The network is designed for communication through serial RS-485 connection over Modbus RTU protocol and Ethernet link over IEC 61850 standard. The necessary configuration and background for the latter are presented in the sections 6, 7 and 8 of this project.

5.6.1. Equipment connection and configuration

For the RS-485 Modbus RTU communication test the G3200 is connected to RS-422/RS-485 port of dSPACE connection panel through DB-9 connector and to the network switch through the Ethernet interface on the output side. Additionally, it requires external 24 V 0.1 A of DC power supply provided by Instek GW SPS-606 switching DC power supply block. The external wiring diagram for G3200 is shown in Fig. 5.16.

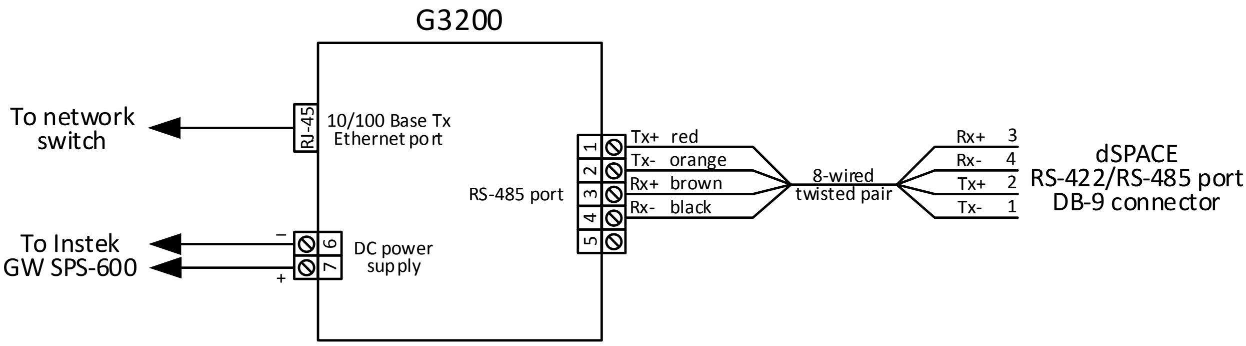


Figure 5.16 – G3200 external wiring diagram.

As it was mentioned above, the dSPACE connection panel for DS1104 R&D Controller Board has one UART with DB-9 connector for both RS-422 and RS-485 interfaces. According to the dSPACE connection panel hardware installation and configuration guide [35], it is possible to configure the port for 4-wired RS-422 or 2-wired RS-485 connection, where in the latter pins 4 and 3 together with 2 and 1 in DB-9 connector should be crosslinked. For the sake of simplicity and possibility to have simultaneous transmitting/receiving for the both master and slave devices in the network, it was decided to use 4-wire RS-422 configuration of the dSPACE connection panel UART port. Taking into account that the main difference between RS-422 and RS-485 interfaces is only in the number of drivers/receivers in the network and all physical and logical characteristics are the same, the RS-485 parameter settings selector switches at G3200 was adjusted for 4-wire network in accordance to device's user guide for communication with dSPACE connection panel RS-422 port [36].

5.6.2. RS-485 network configuration

In the RS-485 Modbus RTU communication test the master device is PowerLogic G3200 converter. It can be accessed for configuration setup and diagnostics through the Ethernet LAN with the aid of Internet Explorer 6.0 or lower. The process of accessing G3200 over Ethernet network described in [36]. In the device's serial port configuration window the settings are adjusted for RS-485 4-wire physical interface, 9600 baud rate, even parity and 1s response timeout as shown in Fig. 5.17.

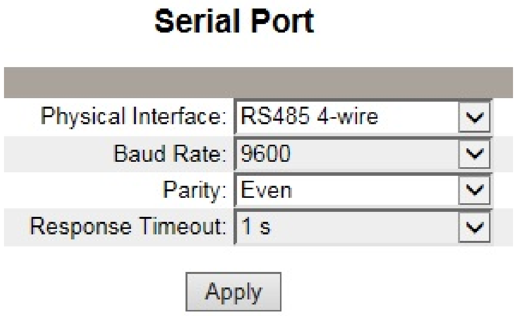


Figure 5.17 – G3200 serial port configuration window.

As for the slave device, the same MATLAB/Simulink Modbus test model described in subsection 5.3 is applied. In order to configure RS-485 communication network the following settings, shown in Fig. 5.18 for DS1104SER block, were selected regarding RS-422 physical interface.

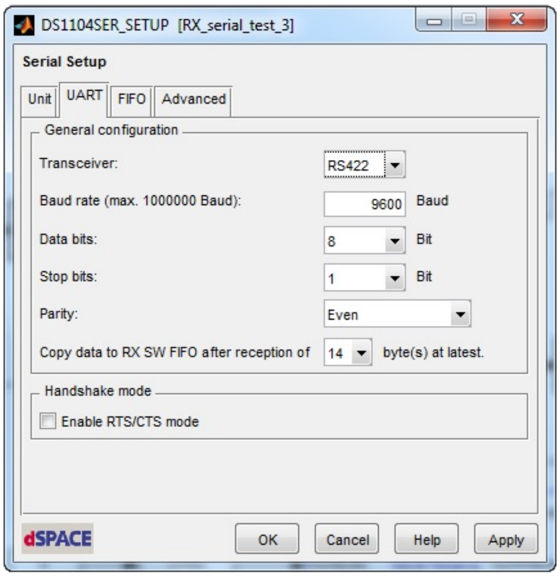


Figure 5.18 – DS1104SER block setup for Modbus RTU communication over RS-485 network.

All other settings for DS1104SER_RX and DS1104SER_TX blocks in the model are identical to those shown in the subsection 5.5 (Fig. 5.8).

5.6.3. Communication tests over RS-485 network

The G3200 embedded web server includes Read Device Registers page for the purpose of Modbus communication tests of G3200 and Modbus devices. This page gives possibility to execute master/slave communication with two Modbus function codes: 03 – Read Holding Registers, 04 – Read Input Registers. The received data from the slave devices can be represented in register columns in decimal, hexadecimal, binary or ASCII formats. The page also includes the following settings: Device ID, Starting register and Number of Registers to read.

The Modbus RTU communication tests over RS-485 network comprise to request/response sessions with 03 function code identical to those described in subsection 5.5 for RS-232 interface. The first request message from G3200 to Simulink model is to read 10 holding registers starting from register number 1 (Fig. 5.19).

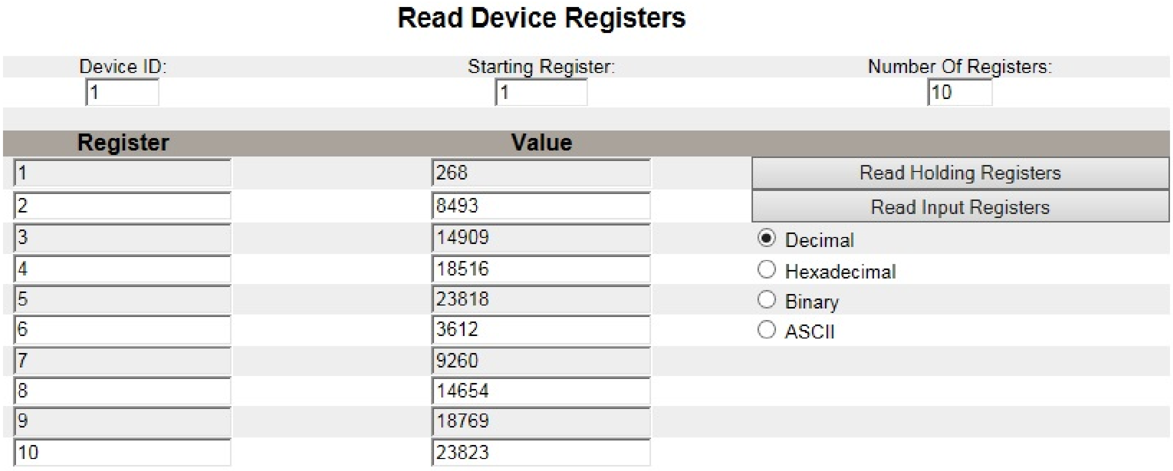


Figure 5.19 – G3200 communication session for Modbus 03 function code (10 registers).

The request message sends by pushing Read Holding Registers button which forms the Modbus RTU query message to execute 03 function code. The response with constructed 25 byte message in this case appears in the register Value column in the form of decimal numbers identical to those described in subsection 5.5.

The second communication session is executed to read 6 holding registers starting from number 3 (same as the second test of Simulink test model for Modbus communication through RS-232 interface in subsection 5.5). The request query and received data are shown in G3200 Read Device Registers web page (Fig. 5.20).

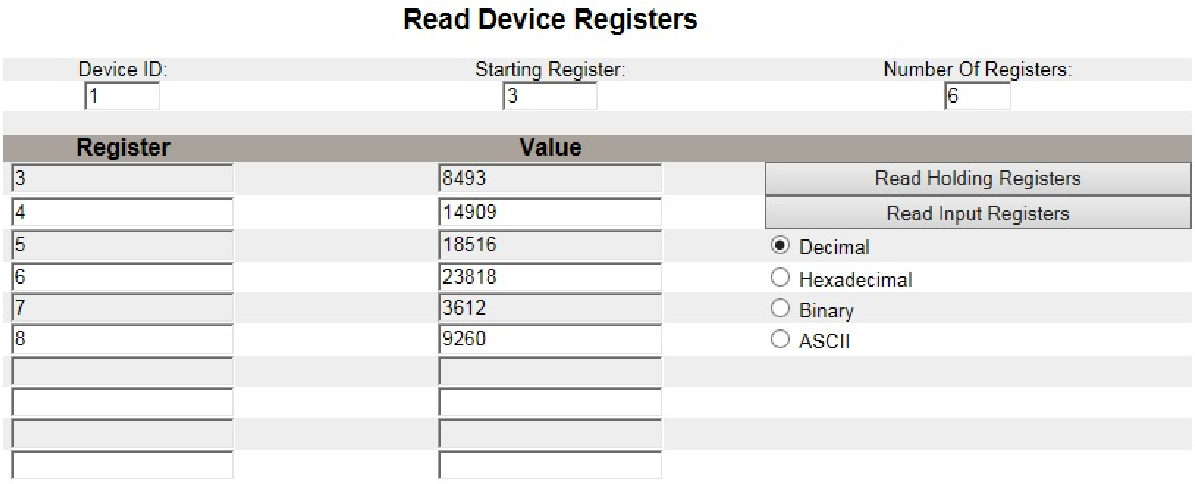


Figure 5.20 - G3200 communication session for Modbus 03 function code (6 registers).

In this case the Register column starts from the register number 3 and includes 6 registers up to number 8. The data appears again in the form of decimal numbers in Value columns (identical to those received in the RS-232 communication test).

6. Communication structure for substation and feeder equipment in IEC 61850 standard

The short overview, basic principles and advantages of IEC 61850 in comparison to proprietary communication protocols were given in the subsections 3.4 and 3.5. The given section provides a brief consideration of the parts of the standard used in this project for establishing the laboratory microgrid's communication, such as Abstract Communication Service Interface (IEC 61850-7-2), Substation Configuration Language (IEC 61850-6) and mappings of the standard's objects to Manufacturing Message Specification (IEC 61850-8-1).

6.1. Abstract Communication Service Interface (ACSI)

ACSI is described in the IEC 61850-7-2 [37] part of the standard providing specifications for multilayer substation communication architecture. The architecture is designed on abstract representations of classes and services, which are not dependent on particular protocol stacks, implementations and operating systems. The definition of ASCI for communication between IEDs and network control system is presented in this part of the standard in terms of:

- Hierarchical model of classes for the data that can be achieved through the communication network;
- Services provided for these classes;
- Parameters related to the services.

The basic conceptual class model of the ACSI used for communication network on IEC 61850 standard is shown in Fig. 6.1.

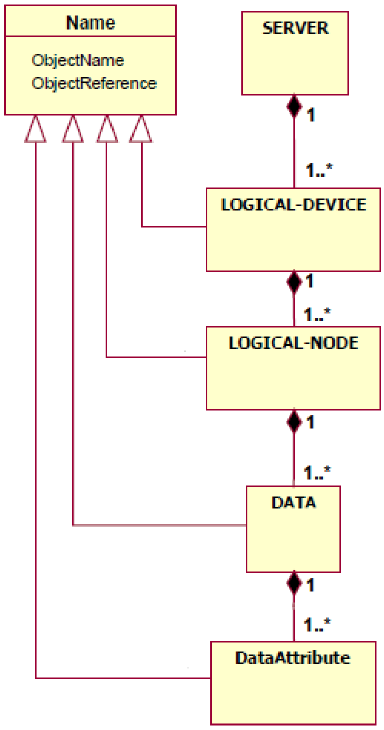


Figure 6.1 – Basic conceptual class model of the ACSI [37].

These essential classes are necessary for construction of domain-specific information models:

- **Server** – defines the overall visible behaviour of a device and comprises in itself all other classes;

- **Logical-Device (LD)** – consists of information generated and applied by domain-specific functions – Logical Nodes;
- **Logical-Nodes (LN)** – domain-specific functions defined for specific applications, for example, measurements, equipment control, protection and so on;
- **Data** – defines approaches for typed information performance, for example, measurement of phase voltage with time stamp and quality information.

The **Name** class is inherited by all classes listed above except **Server**. Additional ACSI classes that should be mentioned are **DataAttribute**, **Data-Set** for grouping of data with common data attributes, presented in the data attribute type (**DAType**) class, which defines specific data types for information in the standard.

The **DAType** class has indexes Name, Presence (mandatory or optional) and BasicTypes. The concept of the class is represented in Fig. 6.2.

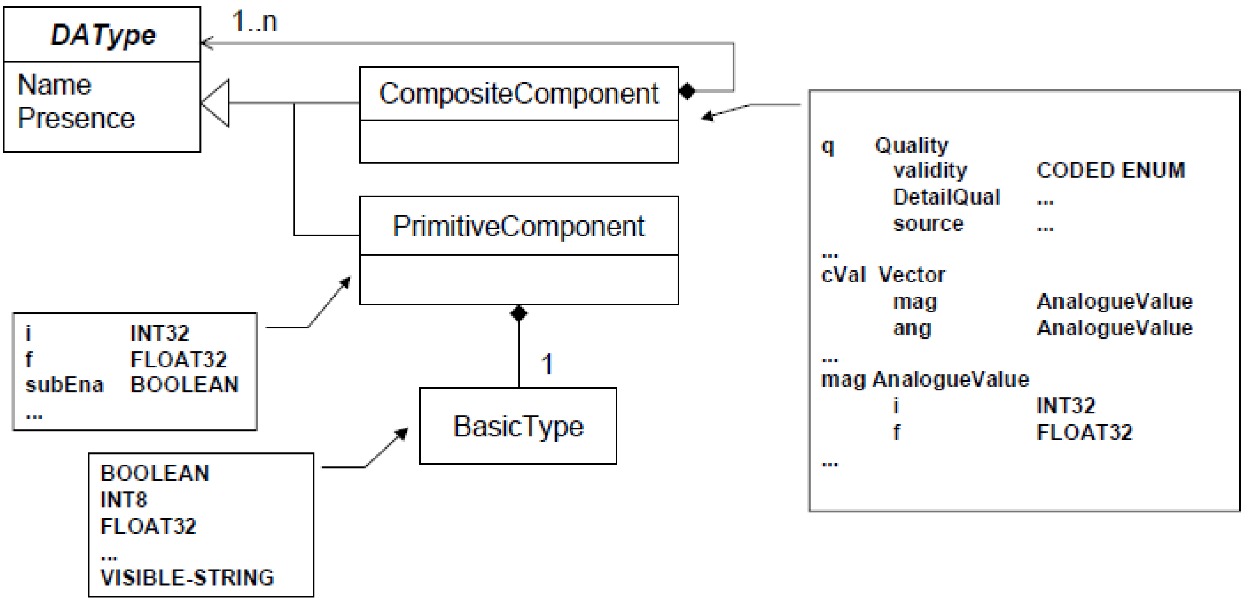


Figure 6.2 – The concept of **DAType** class [37].

The BasicTypes characterizes the data by its type (for instance, VISIBLE STRING, INT8, INT16 and so on) and used to construct PrimitiveComponents and CompositeComponents. PrimitiveComponents comprise Name, Presence and BasicType (for instance, Name = i, Presence = Mandatory, BasicType = INT32). The CompositeComponents include one or more PrimitiveComponents (for instance, Name = mag, which stands for 'magnitude', of type 'AnalogueValue' consists of two PrimitiveComponents – i (INT32) and f (FLOAT32)).

The **DataAttribute** class model includes the above mentioned **DAType** class and its components along with common data attribute (CDA) types and functional constraints (FC). In the example of CompositeComponent 'mag' it has CDA type 'AnalogueValue', which is a part of common data class (CDC) 'MV' – Measurement Value. The IEC 61850 has 30 CDCs that can be grouped by the application for status information, measurand information, controllable status, status settings, analog settings and descriptive information. As a part of every CDC, the standard has 12 CDAs similarly defined by their application: Quality; Analogue value; Configuration of analogue value; Range configuration; Step position with transient indication; Pulse configuration; Originator; Unit definition; Vector definition; Point definition; CtlsModel definition and SboClasses definition.

In order to receive the information from the IEDs with predefined sequence, which has particular rate of relevancy and time criticality, the FCs are included in **DataAttribute** class. FCs are dedicated to each data attribute of the data object (DO) in accordance to its function in the system, for instance, ST – status value, MX – measurand, CF – configuration, DC – description and so on [37].

6.2. Substation Configuration Language (SCL)

SCL is a description language designed for configuration of IEDs and communication system on electrical substations. It provides explicit definitions for relations between the substation automation system and switching, distribution and protection equipment in the substation.

SCL allows interchange of IED datasets for configuration description and communication data between automation system and IED configuration tool in a consistent way regardless of particular manufacturer. It is implemented with a specific file format based on XML version 1.0 for IED configuration and parameters related to communication process with automation and communication systems of substation.

The scope of the language lies in describing the model for:

- The power system structure with it functions and equipment connection;
- The communication system in terms of how IEDs are connected into networks, subnetworks and access points;
- The communication application level of the IEDs – how the data is grouped in devices and in which way IEDs organize communication sessions between them;
- Each IED in the system with its LDs, LNs, and DOs;
- Instantiated LNs definitions and relations between their hosting IEDs and substation equipment.

Besides the standard specified ACSI classes SCL allows creating the user-defined LNs and DOs in accordance to the rules provided in IEC 61850-4.

There are four main types of SCL files defined in the standard for transfer of configuration data between different tools of various manufacturers:

- **ICD.** files (IED Capability Description) – describe the capabilities of particular type of IED and contains necessary data type templates along with LN type definitions and optional substation type definition. The IED and substation section name are defined as ‘TEMPLATE’ and they are changed to specific ones in the process of configuration;
- **SSD.** files (System Specification Description) – intended to provide definition of substation’s single line diagram and associated LNs and data type templates. Required for data exchange between system specification and configuration tools;
- **SCD.** files (Substation Configuration Description) – contain all IEDs in the system, communication configuration and substation description sections. Required for data transfer from the system configuration tool to IED configuration tool;
- **CID.** files (Configured IED Description) – describe instantiated IED in the project with it address information, LDs, LNs and DOs. Used for data exchange from the IED configuration tool to IED.

The Edition 2 of the IEC 61850 standard provides two additional extensions for SCL files, IID (Instantiated IED Description) and SED (System Exchange Description) files. In this project only two types of SCL files are used for the IED configuration – ICD and CID files, which is described in the next section.

The relations between SCL elements are shown in UML (Unified Modelling Language) diagram in Fig. 6.3. The schema type elements starts with small letter t.

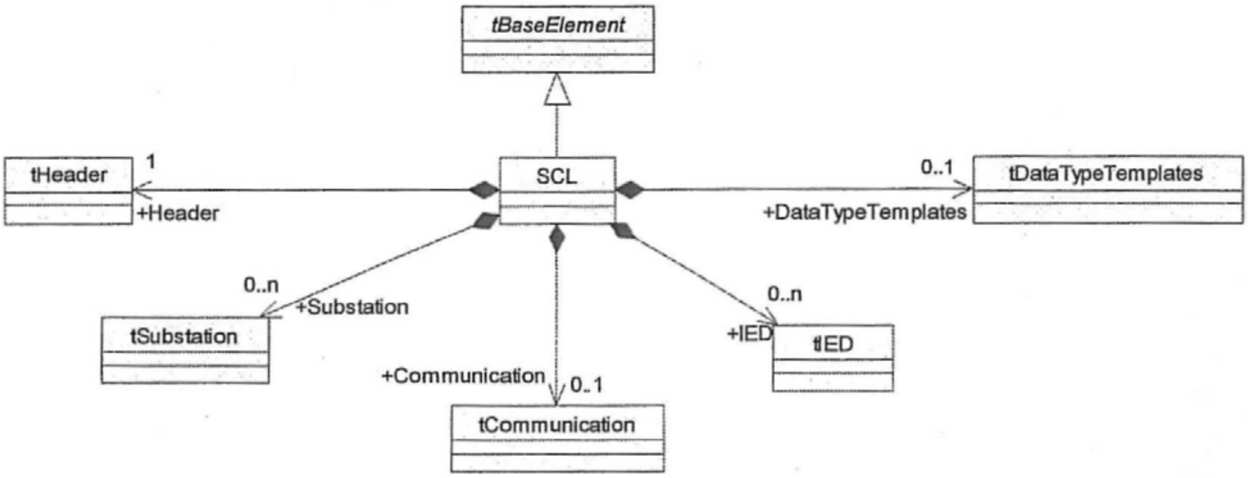


Figure 6.3 – UML diagram overview of SCL schema [39].

The SCL element in the schema is derived from tBasicElement, which gives possibility to contain, for instance, Private and Text definitions. It includes one tHeader element and can comprise several elements of tSubstation and tIEC types. The tCommunication and tDataTypeTemplates types have only one element of each type in the SCL schema [39].

6.3. Manufacturing Message Specification (MMS)

ACSI models of IEC 61850 determine the common behaviour of IEDs from the network perspective in terms of provided services and responses to those services. These abstract models have to be implemented over real widespread protocols and computing environments in the industry. The IEC 61850-8-1 part of the standard specifies mapping of core ACSI objects and services into MMS protocols of ISO (International Organization for Standardisation) 9506.

The given standard was chosen because of its open access, proven track record of implementation and necessary instrumentation for hierarchical naming and service models of IEC 61850. Unlike simple protocols providing only read/write/report services for variables, which can be accessed by register or index numbers, MMS supports complex named objects and wide set of flexible services for direct and trouble-free mapping of IEC 61850.

In order to understand better definitions of MMS translation of IEC 61850 standard the following example can be considered. For instance, it is necessary to have a logical device with name 'Relay1' comprising only one logical node for circuit breaker XCBR1. The task is to determine whether the circuit breaker is in remote or local mode of operation. The IEC 61850-8-1 describes the method of mapping the model information to a named MMS variable object resulting in specific and exclusive representation of every data element for the model [40]. The way the given object will be seen is depicted in Fig. 6.4.

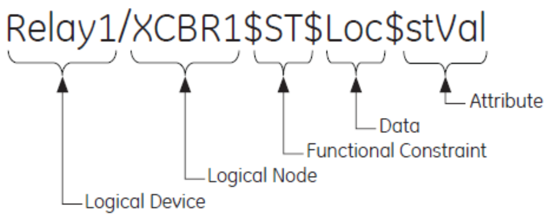


Figure 6.4 – Structure of an IEC 61850-8-1 Object Name [40].

7. PowerLogic G3200 configuration

The main purpose of PowerLogic G3200 Modbus-to-IEC61850 server by Schneider Electric is to include microprocessor devices working on Modbus protocol to the network, based on more advanced and comprehensive communication standard IEC 61850. It ensures inter-operability of Modbus devices with IEC 61850 station bus by having specific configuration file for mapping and integrating of all Modbus driven devices into IEC 61850 communication network. The G3200 provides compatibility with the following IEC 61850 standard sections:

- IEC 61850-6 (SCL Substation Configuration Language);
- IEC 61850-7-1 (modelling concepts);
- IEC 61850-7-2 (ACSI Abstract Communication Service Interface);
- IEC 61850-7-3 (common data classes);
- IEC 61850-7-4 (Logical Nodes data and attributes);
- IEC 61850-8-1 (mapping on Ethernet-based communication networks).

The G3200 includes the rules for reflecting the most common Modbus data types, based on register approach, to IEC 61850 standard, which are specified in the CID file [36]. In this project, G3200 is connected to Test PC through Ethernet LAN for IEC 61850 communication and to dSPACE connection panel UART port through RS-485 Modbus RTU network.

7.1. Creating and transferring the CID file to G3200

For the proper operation of G3200 a specific CID file is needed. This file defines the Modbus-to-IEC61850 translation rules as an arrangement of identifiers specifying the register addresses, register content's format and logic process codes for logic and mathematical operations to be executed on register content in order to map the data to IEC 61850 protocol. The data conversion is performed by requesting to and decoding in the Modbus translation engine inside the G3200 [36].

The creation of the CID file for G3200 is executed with the aid of specific software toll CET850 version 2.1.0 by Schneider Electric. This software contains a library of ICD files written in the SCL language for creating CID files for Schneider Electric devices including G3200. The ICD file defines functions and communication data available in Schneider Electric devices and used as a device model in process of configuration [41].

The procedure of creation a CID file for G3200 from the ICD file involves the following steps:

- Launching CET850 and creation of New SCL file from the File bar;
- From the Add bar the function of add IED file is selected. In the appeared window 'IED Add' the user is asked to select the ICD file from the library for particular IED (in this case – ICD file for G3200 'G3200/SE_ECI_G3200-F01_E1V01.icd'). Following that, the name of IED, description and subnet name should be specified. To finish the creation of CID file in CET850 the IP Address, Subnet mask and Gateway address have to be chosen (here – default G3200 IP Address is 169.254.0.10, default Subnet mask is 255.255.0.0 and Gateway address is 0.0.0.0). Additionally, the SubNetwork name should be provided, which in this case is 'lab_test';
- Finally, from the File bar the button 'Build a CID file' is chosen and by specifying the necessary destination folder and the name (in this case 'G3200_test') the CID file for G3200 is created and saved.

In order to transfer the created CID file to G3200 the Windows browser is used for accessing device's FTP server. The default FTP address is ftp//:169.254.0.10. The CID file is then copied to G3200 FTP server from the previously saved location of the file. The correct execution of this procedure can then be tracked in the G3200 Summary window of device's web server under the graph 'IEC 61850 Configuration files', which is shown in Fig. 7.1.

IEC 61850 Configuration files				
File	Name	Edit time	Version	Revision
Current	G3200_test.cid	2016-04-14 16:16:35	2	0
Backup				

Figure 7.1 – G3200 Summary window in device's web server.

7.2. Configuring the CID file for dSPACE DS1104 connection panel

At the present time there is a great number of free open source software available on the web for creating and editing of SCL files. The presented project implies connection and configuring of non-Schneider Electric Modbus device to G3200 converter. Considering this it was reasonable to configure the specific CID file for dSPACE DS1104 connection panel based on created one for G3200 (described in previous subsection) with one of the free downloaded SCL editors. However, after numerous attempts to accomplish that, the problem arose in Data Type Templates section of the file, where the particular type of Schneider Electric device should be given for proper SCL file validation. Another issue lies in G3200 converter itself, which can be correctly configured for communication only with the files created (or edited) with its dedicated CET850 software. Taking into account the aspects stated above, the goal of configuring of specific CID file for DS1104 was achieved with the aid of CET850 by editing created CID file for G3200.

7.2.1. G3200 global structure

The IEC 61850 server inside the G3200 has the following structure of logical devices shown in Fig. 7.2:

- Logical device corresponded to G3200 itself comprising only two mandatory LNs for each IEC 61850 LD: LLNO and LPHD;
- Logical device for every Modbus device connected to G3200 server with two mandatory LNs as in case of G3200 LD and other optional LNs defined in the specific ICD file for each device.

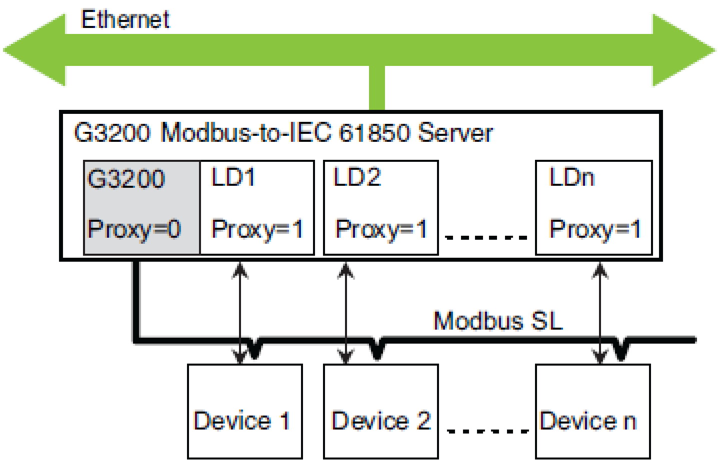


Figure 7.2 – G3200 logical devices [36].

The mandatory LNs belong to the system logical nodes group (L) defined in the IEC 61850-7-4 part of the standard. The DOs and their attributes of the mandatory LNs are given in the Table 7.1 and 7.2.

Table 7.1– LLNO (Logical node zero) description [36].

Attribute name	Attribute type	Explanation/Value	T	M/O/C/E	G3200	Connected Modbus device
LNName	Object Name	LLN0		M	■	■
Common logical node information						
Mod	INC	Mode		M	■	■
Beh	INS	Behavior		M	■	■
Health	INS	Health		M	■	■
NamPlt	LPL	Name plate		M	■	■

All logical nodes in the IEC 61850 network except for LPHD inherit the mandatory data objects listed in the table above from the Common LN [42].

Table 7.2 – LPHD (Physical device information) description [36].

Attribute name	Attribute type	Explanation/Value	T	M/O/C/E	G3200	Connected Modbus device
LNName	Object Name	LPHD1		M	■	■
PhyName	DPL	Physical device name plate		M	■	■
PhyHealth	INS	Physical device health		M	■	■
Proxy	SPS	Indicates if this LN is a proxy		M	■	■

Each LD in the network contains LPHD logical node and its mandatory DOs, which are independent of application domain.

7.2.2. Creating the dSPACE DS1104 logical device

As it was mentioned before, the correct functioning of G3200 can be achieved only with the CID file created or edited with CET850. This software is a powerful tool for processing of SCL files giving possibilities in XML code editing, tree view representation and SCL schema validation at file saving.

In order to create an applicable LD for DS1104 it was decided to use an ICD file from the standard Schneider Electric library provided with CET850 for Easergy T200, the device for remote control of switching equipment in medium/low voltage substations. The process of LD addition to the G3200 sub network consists of the following steps:

- Selecting function Add – Logical Device from the context menu of G3200 IED in the file tree view;
- Specifying LD name – ‘DS1104’, LD description – ‘dSPACE_DS1104’ and Modbus address of the device – ‘1’ ;
- Selecting ICD file from the software standard library for Easergy T200.

From the network perspective, all the LDs connected to the G3200 are seen inside the device IED with common access point including LD0 for the G3200 itself. The CID file sections view for G3200 is shown in Fig. 7.3.

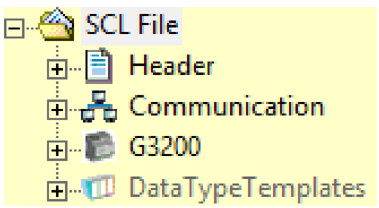


Figure 7.3 – The CID file sections.

The main sections of the file are the following:

- **Header** – contains SCL syntax specification; MD2 key for file validation, which is generated automatically by the software each time the file has been modified; the date of the file creation and communication parameters for the G3200 server;
- **Communication** – defines all sub networks included in the IEC 61850 network with the list of all IEDs. In this case there is only one sub network – 'lab_test' and one IED – G3200;
- **G3200 (IED section)** – contains the definition of each IED connected to IEC 61850 network with all its content: LDs, LNs, DS (Datasets) and RC (Report Control Blocks). For the G3200 CID file there is a common access point with one communication server inside and connected two logical devices – G3200 itself and DS1104;
- **Data Type Templates** – this section provides a detailed description of the data types for the IED elements used in the configuration. For the standard configuration procedure it is hidden by default but in order to adopt the logical device taken from the Schneider Electric library for the custom device this section should be modified, which is described further in this subsection.

The LD assigned as DS1104 contains several groups of LNs representing functionality of Easergy T200 such as logical nodes for supervisory control, switchgear, generic function references, metering and measuring, monitoring and two mandatory LNs. All of them are removed from the file except the one for metering and measuring – MMXU, which is suitable for transferring the data through the IEC 61850 network from the holding registers created in the Modbus Simulink model described in Section 5. Along with that, all the application specific Datasets for Easergy T200 in the LLNO were also excluded from the file. The eventual topology for the G3200 file including the DS1104 logical device is represented in Fig. 7.4.

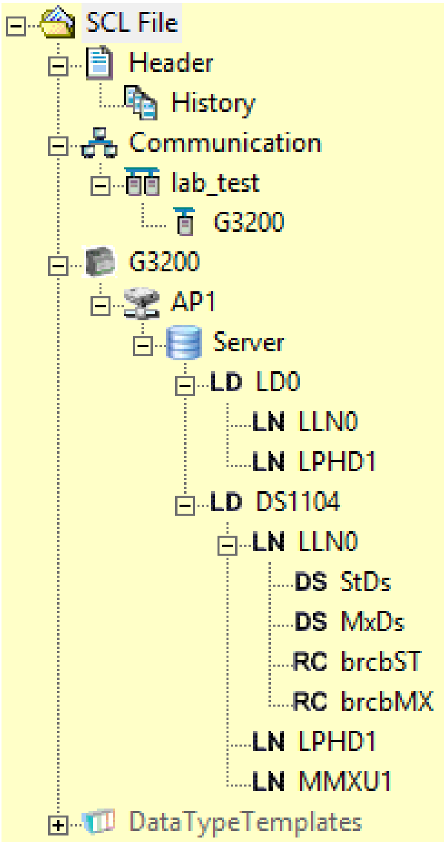


Figure 7.4 – The tree view of G3200 CID file.

7.2.3. Configuring the dSPACE DS1104 logical device

The process of configuring the dSPACE DS1104 LD connected to G3200 server involves editing and adding XML tags in the CID file. Schneider Electric provides specific rules and syntaxes for mapping the Modbus devices into the G3200 IEC 61850 server defined in [36]. The following part of subsection describes the process of setting up the Modbus register objects along with DS1104 LD specification into corresponding IEC 61850 data objects.

The mapping rules are encoded within XML private tags organized in the following manner:

<Private type="SchneiderElectric…">Tag value</Private>, where

<Private…>…</Private> - name of the XML tag, in this case it is 'Private' and slash sign shows the end of the tag;

type – name of the XML attribute;

"SchneiderElectric…" – the value of the XML attribute, which is typically enclosed between inverted commas (");

Tag value – indicates the value of the XML tag.

The code of the 'G3200_DS1104' CID file is unchanged since it was automatically generated by the CET850 until the IED section of DS1104 LD starting after the line

```
<LDevice desc="dSPACE_DS1104" inst="DS1104">
```

The next line is the device model private tag containing information required for checking the correct Modbus device model with specified address. It has the following view:

```
<Private type="SchneiderElectric-IED-
DevModel">series:configuration</Private>
```

The dSPACE DS1104 connection panel is not a Schneider Electric device, so for this case the series part of the tag value should be given as 'UNK' and the configuration part can contain up to 8 string characters for user defined specific device type, here – 'dSPACE'. After setting of the device type to 'UNK' the G3200 will only perform the link connectivity checking at short intervals by sending the Modbus table read requests with the specified addresses for Modbus device and registers in the CID file.

Following the device model designation the following private tags for the IED communication type and Modbus device address in the network are given:

```
<Private type="SchneiderElectric-IED-CommType">JBUS</Private>
<Private type="SchneiderElectric-IED-MdbAddr">1</Private>
```

Here the communication type is set to 'JBUS', which stands for direct register address reading without decrementing the address value, performed in the 'MODBUS' communication type tag value for several Schneider Electric devices. The Modbus address for the DS1104 LD is 1, as it was stated before.

After specifying the communication type and the network address the range of consecutive Modbus register addresses has to be given. This is done inside the Modbus table private tag:

```
<Private type="SchneiderElectric-IED-MdbTbl">
StartAddress:EndAddress:TableType:Proirity</Private>
```

The start and the end addresses for Modbus registers should be presented in the hexadecimal numbers. In case of Modbus device model developed and described in Section 5 the starting address for holding registers is 1 and the end address is A (corresponds to 10 in the decimal form). The table type for registers is assigned to ‘M’ corresponding to measured values and the priority is an optional field, which is without specifying is set to normal (default access to the table). Besides the private tag for measured values registers the initial IED section for LD instantiated as Easergy T200 contained additional Modbus table private tags for status values (S) and control values (C), that are not relevant for the DS1104 LD and, therefore, they were deleted from the file.

At this point the description of the logical device for dSPACE DS1104 inside the CID file is finished. The next step in the configuring of the CID file is creating of application specific data objects inside the LNs of the device.

The LNN0 of Easergy T200, for which the ICD file from the standard library was used, contains two datasets for defining the data objects inside the device: default status reporting dataset and default measurands reporting dataset. Both of them are used for configuring the DS1104 LD with some modifications.

In the default status reporting dataset all of the application specific data objects for Easergy T200, such as supervisory control, switchgear and generic function references objects were removed. For proper interaction of G3200 with created DS1004 LD only data objects with functional constraint for status values of mandatory LNs remained unchanged and the part of the code for default dataset has the following view:

```
<DataSet desc="Default status reporting dataset" name="StDs">
<FCDA doName="Loc" fc="ST" ldInst="DS1104" lnClass="LLN0"/>
<FCDA doName="PhyHealth" fc="ST" ldInst="DS1104" lnClass="LPHD"
lnInst="1" prefix=""/>
<FCDA doName="PwrDn" fc="ST" ldInst="DS1104" lnClass="LPHD"
lnInst="1" prefix=""/>
</DataSet>
```

The second defined dataset for default measurands contained in the initial configuration just a single data object for Easergy T200 application in MMXU LN, required for measured values in IEC 61850 standard, which was also deleted. In order to define the data objects for 10 Modbus holding registers programmed in Simulink model the following code in the CID file was added:

```
<DataSet desc="Default measurands reporting dataset" name="MxDs">
<FCDA doName="MdbHreg1" fc="MX" ldInst="DS1104" lnClass="MMXU" lnInst="1"
prefix=""/>
...
</DataSet>
```

Here the DO is shown for the first holding register with the name ‘MdbHreg1’ and remained nine registers are defined in the same manner with the corresponding number in the end of the name. The functional constraint for these DOs is ‘MX’ matching the measurands in IEC 61850. The ‘FCDA’ notation stands for functional constrained data attributes.

The rest code of the LNNO and LPHD logical nodes for DS1104 LD is left unchanged except for the value of Physical name DO, which can be chosen freely, assigned to ‘DS1104’ for the device representation later in the IEC 61850 client software:

```
<DOI name="PhyNam">
 <DAI name="model" valKind="Set">
  <Val>DS1104</Val>
 </DAI>
```

After the designation of specific dataset with DOs for all 10 holding registers it is necessary to define them in their host LN – in this case it is MMXU. The mapping of the Modbus objects to corresponding IEC 61850 data objects is performed with IEC-PntRef private tag. The tag can occur at the DOI (DO Instantiated), SDI (Sub-Data Instantiated – part of structured DO) or DAI (Data Attribute Instantiated) level. The application of IED-PntRef in the CID file code is done through the mapping string as a combination of number of fields separated by semicolon needed to provide epy G3200 necessary instructions about the type, range, size, scale and particular logic to be applied to Modbus register(s) with specific address [36]. It is given as following:

```
<Private type="SchneiderElectric-IED-PntRef">
Type;RegisterDescription1;...
RegisterDescriptionN;ProcessingCode</Private>
```

In the most common case of single attribute (usually the primary attribute) of an IEC 61850, the DO has a corresponding Modbus register in the device and mapping is executed at DOI level. This is also called the 'object level mapping', which applies for the case of DS1104 LD data mapping.

The 'Type' string on the object level is notated by uppercase letter 'T' followed by a colon and two uppercase letters for description of IEC 61850 CDC. The object types currently defined for G3200 application of IEC 61850 are listed in the Table 7.3

Table 7.3 – List of the IEC 61850 object types for application in G3200 [36].

Type	IEC 61850 Common Data Class
SS	SPS - Single Point Status
DS	DPS - Double Point Status
IS	INS - Integer Status (Integer)
ES	INS - Integer Status (Enumerated) from edition 1
IC	INC - Integer Status Controllable (control & status)
IO	INC - Integer Status Controllable (status only)
IN	INC - Integer Status Controllable (control only)
SC	SPC - Single Point Controllable (control & status)
SO	SPC - Single Point Controllable (status only)
SN	SPC - Single Point Controllable (control only)
DC	DPC - Double Point Controllable (control & status)
DO	DPC - Double Point Controllable (status only)
DN	DPC - Double Point Controllable (control only)
MV	MV - Measured Value
CM	CMV - Complex Measured Value
BC	BCR - Binary Counter Reading
AT	ACT - ACtivation
AD	ACD - ACtivation Directional
ST	String
UT	UTC1 time

For created Modbus holding registers in Simulink model the most appropriate choice for the IEC 61850 object type is MV – Measured Value.

After defining the type of DO the register description field is followed by 'Register Type' and 'Address' strings. There are four lowercase or uppercase letters to define the type of register:

- m/M – for Modbus holding register;
- c/C – for Modbus control coils;
- s/S – for Modbus status coils;
- i/I – for Modbus input registers.

The lower or uppercase letter define the format of the address is given – in decimal or hexadecimal numbers respectively. The 'Register Type' and 'Address' strings are followed by the size field after colon (here it is assigned to u1 representing one 16-bit unsigned register and can be extend to ten 16-bit registers in general case) and scaling field, which is not relevant for the purpose of direct mapping of holding registers values into IEC 61850 objects.

The last string included in the IED-PntRef tag is the 'Processing Code' defining the rules and formulas to be applied to the data red from the Modbus coils and registers. The notation for the processing code is 'L:P:N', where N stands for a number corresponding to the specific logic code to be applied to interpret or decode the data in a register or a set of registers in order to receive the value of the attribute of the corresponding IEC 61850 object. The codes are divided into groups depending on the type of data to which these rules are applied. The groups with designated code numbers are represented in Table 7.4.

Table 7.4 – G3200 supported processing codes [36].

Group name	Logical code range
Date/time	1-8
Modulo 10K	10-13
Values with bitmasks	20-29 and 226-229
Scaled registers	30-37
Strings	39
Maths operations on scale registers	40-46
Power factor	50-54
Unique situations	300-312
Write	101-103

The DO for MMXU LN of Easergy T200 logical device initially contained the processing code L:P:32 in its IED-PntRef tag for reading one or two sequential registers in FLOAT32 data format with scaling to one decimal. The same processing code is applied to Modbus holding registers in DS1104 Simulink model without field for scaling to one decimal. The resulted DO for the first holding register with IED-PntRef private tag for Modbus data mapping to IEC 61850 object is written as:

```
<DOI name="MdbHreg1">
<Private type="SchneiderElectric-IED-PntRef">T:MV;m:1:u1;L:P:32</Private>
</DOI>
```

The identical notation is applied to the rest of the registers except the number in the DO name and Modbus address corresponding to the respective value for each register provided in decimal format. With all required DOs defined inside the MMXU LN the process of configuring the CID file in the IED section is accomplished.

The final stage for DS1104 LD definition in the CID file for G3200 comprises the designation of Modbus registers DOs in Data Type Templates section of the file. The given section contains data types for LNs, DOs, DAs and Enumeration used for correct functioning of G3200 converter and DS1104 connection panel through IEC 61850.

The Data Type Templates section is generated automatically in the stage of CID file creation for G3200 and later for Easergy T200 logical devices described earlier in this subsection. As a rule, while creating LDs for Schneider Electric manufactured devices there is no need in editing this section. However, for the purpose of reflecting the custom DOs created manually for Modbus holding registers, these data objects have to be included instead of initially generated by the CET850 under the LN data type subsection for MMXU. This part of the XML code in the CID file has the following view:

```
<LNodeType id="SE_MMXU_Easergy_V001" iedType="" lnClass="MMXU">
 <DO name="Mod" transient="false" type="SE_Mod_Easergy_V001"/>
 <DO name="Beh" transient="false" type="SE_Beh_Easergy_V001"/>
 <DO name="Health" transient="false" type="SE_Health_Easergy_V001"/>
 <DO name="NamPlt" transient="false" type="SE_Lpl_Easergy_V001"/>
 <DO name="MdbHreg1" transient="false" type="SE_MvEx_Easergy_V001"/>
  ...
 <DO name="MdbHreg10" transient="false" type="SE_MvEx_Easergy_V001"/>
```

The identification for MMXU LN here is ‘SE_MMXU_Easergy_V001’ which stands for standard IEC 61850 logical node of this type in Schneider Electric Easergy device version 1. The first four DOs listed in this definition are inherited from LLN0 of the device. The last ten DOs reflect the data to be red from DS1104 Simulink Modbus model with the type ‘SE_MvEx_Easergy_V001’. This data type is defined further in the file for MV CDC in such a way:

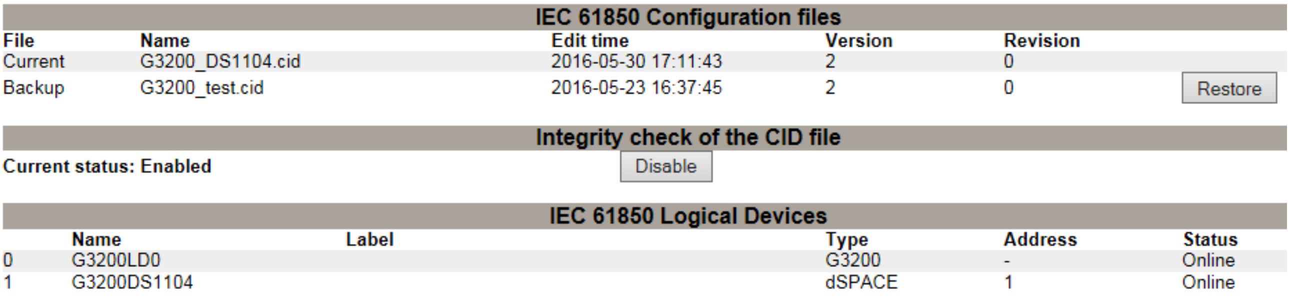
```
<DOType cdc="MV" id="SE_MvEx_Easergy_V001" iedType="">
 <DA bType="Struct" count="0" dchg="false" dupd="false" fc="MX"
name="mag" qchg="false" type="analogVal" valKind="Set"/>
 <DA bType="Quality" count="0" dchg="false" dupd="false" fc="MX" name="q"
qchg="true" valKind="Set"/>
 <DA bType="Timestamp" count="0" dchg="false" dupd="false" fc="MX"
name="t" qchg="false" valKind="Set"/>
 <DA bType="INT32U" count="0" dchg="false" dupd="false" fc="CF" name="db"
qchg="false" valKind="Set"/>
 <DA bType="VisString255" count="0" dchg="false" dupd="false" fc="EX"
name="dataNs" qchg="false" valKind="Set">
   <Val>SE_dataNs_V0001</Val>
 </DA>
</DOType>
```

The name of the data attribute for MV object type defined inside IED-PntRef private tag in IED section of the file is ‘mag’ (stands for magnitude) of type ‘analogVal’. This is the only one data attribute in the file involved in transferring of the specific data other than generated by default with CET850 software inside the CID file. The definition of the given DA is the following:

```
<DAType id="analogVal" iedType="">
 <BDA bType="FLOAT32" count="0" name="f" valKind="Set"/>
</DAType>
```

The basic data type for this data attribute is FLOAT32 corresponding to maximum integer value can be transferred up to 2,147,483,650. As it was mentioned above, this data format without specifying decimal scaling is completely suitable for data mapping of Modbus holding registers from DS1104 Simulink model.

After completion of the CID file configuring and transferring it through G3200 FTP server to the device, the G3200 Summary page in device’s web browser will contain the information about CID files created and logical devices with their Modbus addresses connected as shown in Fig. 7.5.



IEC 61850 Configuration files

File	Name	Edit time	Version	Revision	
Current	G3200_DS1104.cid	2016-05-30 17:11:43	2	0	
Backup	G3200_test.cid	2016-05-23 16:37:45	2	0	Restore

Integrity check of the CID file

Current status: Enabled	Disable

IEC 61850 Logical Devices

	Name	Label	Type	Address	Status
0	G3200LD0		G3200	-	Online
1	G3200DS1104		dSPACE	1	Online

Figure 7.5 – G3200 Summary web page with connected dSPACE DS1104.

The code of ‘G3200_DS1104’ CID file is given in Appendix C.

8. Laboratory microgrid test based on IEC 61850 communication

8.1. IEC 61850 client software overview

During the past decades IEC 61850 standard became a widespread tool for communication in automation systems for electrical power industry. The standard is deployed and supported by various SCADA and OPC (Open Platform Communications) software from different manufacturers all over the world because of its interoperability and broad range of provided communication services. Therefore, the process of selecting the suitable software for testing laboratory microgrid communication system via IEC 61850 involved reviewing of several options available in free access on the web or in form of demo versions.

One of the considered options was KEPServerEX, an OPC server software of Kepware Technologies company from USA. This is a powerful communications platform for industrial automation data supporting various protocols including IEC 61850. The configuration process for establishing the network monitoring with this software involved generating of communication channel and running OPC client session for the device through IEC 61850 MMS. The tag base for MMS service is created by specifying a CID file for IED, which also includes necessary local network parameters required for accessing the device.

Considering a relatively tedious procedure for creating a communication session with KEPServerEX and limited amount of running time for the free demo version, the choice was made for using of much simpler software – IEDExplorer 0.78 Experimental. It is an IEC 61850 client tool developed for testing and educational purposes and uploaded for free access by Pavel Charvat from Czech Republic [43].

IEDExplorer is a quite straightforward program providing the connection with IED through IEC 61850 MMS service. The only parameter required for communication with the device is the IP-address in the network. The interface of the software has available two types of the IED tree view tab:

- MMS View – the representation of the IED is organised according to functional constraints for every data object inside the LNs;
- IEC 61850 View – the tree view based on IED structure defined in ACSI in IEC 61850 standard.

The corresponding IED data view tab reflects names, types, values and communication addresses for the data selected in one of the tree views. Additionally, IEDExplorer has a LogView tab for initializing the actions performed for establishing the communication session and possible errors occurred.

8.2. Laboratory microgrid test and results

The MATLAB/Simulink model described in Section 5 emulates the functionality of Modbus slave device. It comprises CRC check calculation and verification together with response function code including predefined 10 holding registers. The model is capable of receiving Modbus queries for read up to 10 holding registers and responding according to requests from G3200 through the RS-422/485 port of the dSPACE DS1104 connection panel. The Modbus master device in the microgrid is the G3200 converter, which participates in the system as an IEC 61850 server, Modbus-to-IEC 61850 protocol gateway and RS-485/Ethernet interface converter simultaneously. The IEDExplorer software deployed on Test PC acts as an IEC 61850 client monitoring system by communicating with G3200 through IEC 61850 MMS service. The application diagram representing communication protocols, interfaces and software used in the laboratory microgrid is shown in Fig. 8.1.

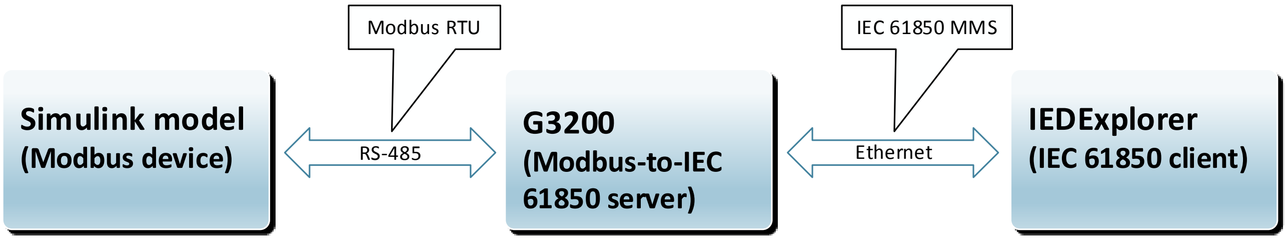


Figure 8.1 – The application diagram of Sulkowski laboratory microgrid.

The process of communication via Modbus RTU protocol in the microgrid was described in Section 5 with two tests presented – read ten holding registers starting from number 1 and six holding registers starting from number 3. The final test performed involves accessing the Modbus device and read all ten registers programmed in its memory by IEC 61850 Client software described above. The CID file required for configuration of G3200 and mapping the Modbus data to IEC 61850 protocol is created in such a way, that each holding register stored in Modbus device will be red and visualized in IEC 61850 monitoring software.

In order to launch the communication session from IEDExplorer the proper IP-address of G3200 should be entered in corresponding field 'IP Address of Server' – here it is 169.254.0.10. By clicking the button 'Run' the software starts connection with the device and all the actions performed to establish the communication is shown in the LogView tab. After connecting to the G3200 the following information appears in IED tree view tab of IEDExplorer shown in Fig. 8.2.

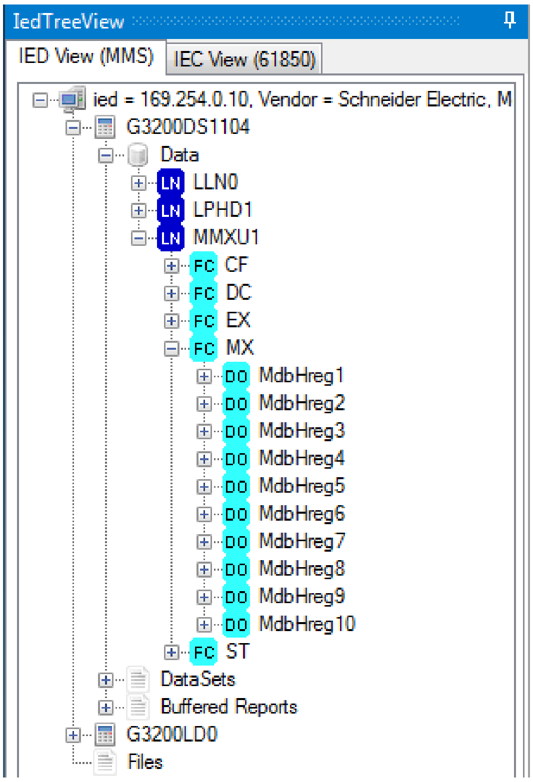


Figure 8.2 – The IED tree view (MMS) for the connected G3200 in IEDExplorer.

As it can be seen in the Fig. 8.2, the IED tree view of G3200 is organised to represent LDs contained in the device (G3200DS1104 for Modbus device and G3200LD0 for G3200 itself), LNs under each logical device as a part of the data defined in the CID file, DataSets and Buffered

Reports. In the MMS IED tree view the data organised according to its functional constrains and the Modbus holding registers are presented with FC Measured Variables – 'MX'. By selecting the given FC from the tree view the corresponding values for DOs appear in the IED data view tab as it shown in Fig. 8.3.

Name	Type	Value	Communication Address
G3200DS1104/MMXU1	IEDExplorer.NodeFC		Dom = G3200DS1104 Var = MMXU1$MX
------------ CHILD NODES ------------	------------	------------	------------
G3200DS1104/MMXU1.MdbHreg1.mag.f	floating_point	268	Dom = G3200DS1104 Var = MMXU1MXMdbHreg1magf
G3200DS1104/MMXU1.MdbHreg1.q	bit_string	0000000000000 [NONE]	Dom = G3200DS1104 Var = MMXU1MXMdbHreg1$q
G3200DS1104/MMXU1.MdbHreg1.t	utc_time	01.01.1970 13:00:01.934 [LOC] TimQualTimeBaseErr	Dom = G3200DS1104 Var = MMXU1MXMdbHreg1$t
G3200DS1104/MMXU1.MdbHreg2.mag.f	floating_point	8493	Dom = G3200DS1104 Var = MMXU1MXMdbHreg2magf
G3200DS1104/MMXU1.MdbHreg2.q	bit_string	0000000000000 [NONE]	Dom = G3200DS1104 Var = MMXU1MXMdbHreg2$q
G3200DS1104/MMXU1.MdbHreg2.t	utc_time	01.01.1970 13:00:01.934 [LOC] TimQualTimeBaseErr	Dom = G3200DS1104 Var = MMXU1MXMdbHreg2$t
G3200DS1104/MMXU1.MdbHreg3.mag.f	floating_point	14909	Dom = G3200DS1104 Var = MMXU1MXMdbHreg3magf
G3200DS1104/MMXU1.MdbHreg3.q	bit_string	0000000000000 [NONE]	Dom = G3200DS1104 Var = MMXU1MXMdbHreg3$q
G3200DS1104/MMXU1.MdbHreg3.t	utc_time	01.01.1970 13:00:01.934 [LOC] TimQualTimeBaseErr	Dom = G3200DS1104 Var = MMXU1MXMdbHreg3$t
G3200DS1104/MMXU1.MdbHreg4.mag.f	floating_point	18516	Dom = G3200DS1104 Var = MMXU1MXMdbHreg4magf
G3200DS1104/MMXU1.MdbHreg4.q	bit_string	0000000000000 [NONE]	Dom = G3200DS1104 Var = MMXU1MXMdbHreg4$q
G3200DS1104/MMXU1.MdbHreg4.t	utc_time	01.01.1970 13:00:01.934 [LOC] TimQualTimeBaseErr	Dom = G3200DS1104 Var = MMXU1MXMdbHreg4$t
G3200DS1104/MMXU1.MdbHreg5.mag.f	floating_point	23818	Dom = G3200DS1104 Var = MMXU1MXMdbHreg5magf
G3200DS1104/MMXU1.MdbHreg5.q	bit_string	0000000000000 [NONE]	Dom = G3200DS1104 Var = MMXU1MXMdbHreg5$q
G3200DS1104/MMXU1.MdbHreg5.t	utc_time	01.01.1970 13:00:01.934 [LOC] TimQualTimeBaseErr	Dom = G3200DS1104 Var = MMXU1MXMdbHreg5$t
G3200DS1104/MMXU1.MdbHreg6.mag.f	floating_point	3612	Dom = G3200DS1104 Var = MMXU1MXMdbHreg6magf
G3200DS1104/MMXU1.MdbHreg6.q	bit_string	0000000000000 [NONE]	Dom = G3200DS1104 Var = MMXU1MXMdbHreg6$q
G3200DS1104/MMXU1.MdbHreg6.t	utc_time	01.01.1970 13:00:01.934 [LOC] TimQualTimeBaseErr	Dom = G3200DS1104 Var = MMXU1MXMdbHreg6$t
G3200DS1104/MMXU1.MdbHreg7.mag.f	floating_point	9260	Dom = G3200DS1104 Var = MMXU1MXMdbHreg7magf
G3200DS1104/MMXU1.MdbHreg7.q	bit_string	0000000000000 [NONE]	Dom = G3200DS1104 Var = MMXU1MXMdbHreg7$q
G3200DS1104/MMXU1.MdbHreg7.t	utc_time	01.01.1970 13:00:01.934 [LOC] TimQualTimeBaseErr	Dom = G3200DS1104 Var = MMXU1MXMdbHreg7$t
G3200DS1104/MMXU1.MdbHreg8.mag.f	floating_point	14654	Dom = G3200DS1104 Var = MMXU1MXMdbHreg8magf
G3200DS1104/MMXU1.MdbHreg8.q	bit_string	0000000000000 [NONE]	Dom = G3200DS1104 Var = MMXU1MXMdbHreg8$q
G3200DS1104/MMXU1.MdbHreg8.t	utc_time	01.01.1970 13:00:01.934 [LOC] TimQualTimeBaseErr	Dom = G3200DS1104 Var = MMXU1MXMdbHreg8$t
G3200DS1104/MMXU1.MdbHreg9.mag.f	floating_point	18769	Dom = G3200DS1104 Var = MMXU1MXMdbHreg9magf
G3200DS1104/MMXU1.MdbHreg9.q	bit_string	0000000000000 [NONE]	Dom = G3200DS1104 Var = MMXU1MXMdbHreg9$q
G3200DS1104/MMXU1.MdbHreg9.t	utc_time	01.01.1970 13:00:01.934 [LOC] TimQualTimeBaseErr	Dom = G3200DS1104 Var = MMXU1MXMdbHreg9$t
G3200DS1104/MMXU1.MdbHreg10.mag.f	floating_point	23823	Dom = G3200DS1104 Var = MMXU1MXMdbHreg10magf
G3200DS1104/MMXU1.MdbHreg10.q	bit_string	0000000000000 [NONE]	Dom = G3200DS1104 Var = MMXU1MXMdbHreg10$q
G3200DS1104/MMXU1.MdbHreg10.t	utc_time	01.01.1970 13:00:01.934 [LOC] TimQualTimeBaseErr	Dom = G3200DS1104 Var = MMXU1MXMdbHreg10$t

Figure 8.3 – Modbus holding registers red over IEC 61850 protocol.

The values for registers are defined in Section 5 and constructed from the array of 20 random numbers. The register value is constructed from two bytes, the first (high byte) corresponds to the first decimal number and the second (low byte) is defined by the second number of the array. The numbers are sent in the hexadecimal form over Modbus and received bytes are then translated back to decimal form as it shown in Table 8.1 for the first two registers programmed in the Modbus device model.

Table 8.1 – Encoding and decoding of Modbus registers.

Number format	**Register 1**		**Register 2**	
	High byte	Low byte	High byte	Low byte
Decimal	1	12	33	45
Hexadecimal	01	0C	21	2D
Decimal (final value)	268		8493	

The type of the register data is floating point or FLOAT32 (according to IEC 61850) as it was defined in the CID file by the reason of the data type for Easergy T200 device initially used as a template for Modbus logical device. The DO for Modbus holding register is of structured data type designated with the name 'mag' as it is represented in Section 7 for configuring the CID file. The object 'mag' has an 'analogVal' data attribute type, which is separately defined in the CID file with the name 'f' and FLOAT32 basic data type.

Additionally, each register DO has a quality and time attributes denoted as ‘q’ and ‘t’ respectively. There is no information about the quality of DO, so the value received in bit string format is zeros. The time stamp for each register red corresponds to the default value with the data 01.01.1970 and time 13.00.01.934. It is a local time of the G3200 without time synchronization, which is not possible to perform without separate SNTP (Simple Network Time Protocol) server in the network.

The overall system performs communication in real-time mode with minor time delay for accessing the data, reading and visualizing. The IEDExplorer gives possibility for auto update the data red from the device with the smallest possible time delay of 100 ms. The data for Modbus holding registers can be changed manually inside the Simulink model and additional incremental build process should be applied for the model in order to change the data values inside the MATLAB. Considering that, the test for reading the updated values of the registers in the microgrid will take several seconds time delay.

9. Conclusions and recommendations for further work

The presented project investigated the smart grid as a key concept for sustainable operation of modern and future power system. Two main building blocks for this concept are microgrids and virtual power plants, which were reviewed in terms of technological structure and implementation examples. The main accent was done in consideration of communication systems in microgrids and VPPs. The detailed evaluation of legacy communication protocols together with IEC 61850 standard for substations automation was performed. The Modbus RTU and IEC 61850 were selected for practical realisation for the communication system in Sulkowski laboratory microgrid at UiT – The Arctic University of Norway (Campus Narvik).

The laboratory microgrid consists of back-to-back AC-DC-AC power electronics converter linked to the dSPACE DS1104 connection panel and Test PC. The first test performed is communication via Modbus RTU protocol over RS-232 serial connection. The Modbus master device was emulated on Test PC with the aid of Modbus Poll software. The Modbus slave device was created in MATLAB/Simulink model comprising transmitter and receiver blocks for DS1104 synchronized with the Simulink library and available for communication from the model. The Modbus device model was designed to have 10 holding registers with values constructed from random numbers and capability to transfer the data from created registers with Modbus RTU protocol communication.

The further development of the communication system for microgrid involved Modbus RTU test over RS-485 serial connection with included additional equipment – G3200 Modbus-to-IEC 61850 server by Schneider Electric and Ethernet network switch. In this configuration G3200 acted as Modbus master device initiating the communication sessions with the Simulink model. In the given test RS-422/485 serial port of DS1104 connection panel was used and corresponding adjustments for RS-485 serial connection were performed.

The final configuration of the laboratory microgrid communication system is based on IEC 61850 as a main communication protocol. The G3200 is an IEC 61850 server, which translates Modbus data objects to corresponding ones in IEC 61850 with the aid of specific CID configuration file written on SCL language. The monitoring system deployed on Test PC in form of IEDExplorer IEC 61850 client software accesses G3200 through Ethernet link and retrieves all the data from the Modbus slave device’s programmed holding registers in the real-time mode.

The created communication system operates simultaneously with two protocols and gives possibility of reading predefined data values from the Modbus device emulated in MATLAB/Simulink. The future development of the presented system involves improvement of existing Simulink model by including the DS1104 measurement blocks. In such a way, the real measurement values for currents and voltages of the back-to-back converter can be encapsulated inside Modbus message frame and transmitted over the communication system. Another suggestion can be done for the monitoring software, which can be implemented as a SCADA application with appropriate HMI (Human Machine Interface) for visualization of electrical diagram and corresponding measured parameters of the converter and microgrid. Additionally, the functionality of microgrid’s communication system can be extended for providing the possibility of writing the data to registers and remote control of the device(s).

References

[1] Xi Fang, Satyajayant Misra, Guoliang Xue and Dejun Yang. Smart Grid – The New and Improved Power Grid: A Survey. IEEE communications surveys & tutorials, vol. 14, no. 4, fourth quarter 2012.

[2] Peter Asmus. Microgrids, Virtual Power Plants and Our Distributed Energy Future. The Electricity Journal 12/2010.

[3] J. Duncan Glover, Mulukutla S. Sarma, Thomas J. Overbye. Power system analysis and design. Fifth edition.

[4] Dae-Geun Jin, Jong-Chan Choi, Dong-Jun Won, Hak-Ju Lee, Woo-Kyu Chae and Jung-Sung Park. A Practical Protection Coordination Strategy Applied to Secondary and Facility Microgrids. Energies 2012, 5, 3248-3265

[5] Jinyu Wang, Guoliang Chen , Enmei Wu, Libing Sun, Liying Zhang, Kai Sun. Research of Smart Microgrid Energy Monitoring System based on EMS. International Conference on Intelligent Systems Research and Mechatronics Engineering (ISRME 2015)

[6] Barnes, M. Kondoh, J. Asano, H. Oyarzabal, J. Ventakaramanan, Lasseter, Robert Hatziargyriou, N. Green. Real-World MicroGrids - An Overview. System of Systems Engineering, 2007. SoSE '07.

[7] R. H. Lasseter. MicroGrids. Power Engineering Society Winter Meeting, 2002. IEEE

[8] Gopalan, S.; Sreeram, V.; Iu, H.; Mishra, Y. An improved protection strategy for microgrids. Innovative Smart Grid Technologies Europe (ISGT EUROPE), 2013 4th IEEE/PES

[9] S. Morozumi, "Overview of Microgrid Research and Development Activities in Japan", International Symposium on Microgrids, Montreal, June 2006.

[10] R.H. Lasseter and P. Piagi, "Control and Design of Microgrid Components", Final project report, PSERC publication 06-03, http://certs.aeptechlab.com/

[11] Plancke, G.; De Vos, K.; Belmans, R.; Delnooz, A. Virtual power plants: Definition, applications and barriers to the implementation in the distribution system. European Energy Market (EEM), 2015 12th International Conference.

[12] D. Pudjianto, C. Ramsay, and G. Strbac, “Virtual power plant and system integration of distributed energy resources,” Renewable Power Generation, IET, vol.1, no. 1, pp. 10–16, 2007.

[13] Toma, L.; Otomega, B.; Bulac, C.; Tristiu, I. Coordination of distributed generators through the virtual power plant concept. Innovative Smart Grid Technologies (ISGT Europe), 2012 3rd IEEE PES International Conference and Exhibition

[14] Oyarzabal, J.; Marti, J.; Ilo, A.; Sebastian, M.; Alvira, D.; Johansen, K. Integration of DER into power system operation through Virtual Power Plant concept applied for voltage regulation. Integration of Wide-Scale Renewable Resources into the Power Delivery System, 2009 CIGRE/IEEE PES Joint Symposium.

[15] Marra, F.; Sacchetti, D.; Pedersen, A.B.; Andersen, P.B.; Traholt, C.; Larsen, E. Implementation of an Electric Vehicle test bed controlled by a Virtual Power Plant for contributing to regulating power reserves. Power and Energy Society General Meeting, 2012 IEEE

[16] Bani-Ahmed, A.; Weber, L.; Nasiri, A.; Hosseini, H. Microgrid communications: State of the art and future trends. Renewable Energy Research and Application (ICRERA), 2014 International Conference

[17] Moghimi, Mojtaba; Bennett, Christopher; Leskarac, Domagoj; Stegen, Sascha;Lu, Junwei. Communication architecture and data acquisition for experimental MicroGrid installations. Power and Energy Engineering Conference (APPEEC), 2015 IEEE PES Asia-Pacific

[18] Galli, S.; Scaglione, A.; Zhifang Wang. For the Grid and Through the Grid: The Role of Power Line Communications in the Smart Grid. Proceedings of the IEEE. Year: 2011, Volume: 99, Issue: 6. Pages: 998 - 1027

[19] V.C. Gungor, F.C. Lambert. A survey on communication networks for electric system automation. Computer Networks 50 (2006) 877–897

[20] Uribe-Perez, N.; Hernandez, L.; Gomez, R.; Soria, S.; de la Vega, D.; Angulo, I.; Arzuaga, T.; Gutierrez, L. Smart management of a distributed generation microgrid through PLC PRIME technology. Smart Electric Distribution Systems and Technologies (EDST), 2015 International Symposium

[21] Mini S. Thomas, John D. McDonald. Power system SCADA and Smart Grids. 2015.

[22] Kuzlu, M.; Pipattanasomporn, M. Assessment of communication technologies and network requirements for different smart grid applications. Innovative Smart Grid Technologies (ISGT), 2013 IEEE PES

[23] L. Baker. EPB Deploys America's Fastest Fiber-optic Smart Grid. Electric Energy T & D magazine. January/February 2011. Available: http://www.electricenergyonline.com/show_article.php?mag=&article=550

[24] A. Tisot, Rio Grande electric monitors remote energy assets via satellite, Utility Automation & Engineering T&D Magazine (July) (2004).

[25] The Virtual Power Plant as a Tool for the Energy Transition. RWE Deutschland AG. April 2013.

[26] S. Borlase. Smart Grids: Infrastructure, Technologies and Solutions. 2013

[27] G. Clarke, D. Reynders, E. Wright. Practical Modern SCADA Protocols: DNP3, 60870.5 and Related Systems. 2004.

[28] Guarese, G.B.M.; Sieben, F.G.; Webber, T.; Dillenburg, M.R.; Marcon, C. Exploiting Modbus Protocol in Wired and Wireless Multilevel Communication Architecture Brazilian Symposium on Computing System Engineering (SBESC), 2012

[29] Modbus-IDA. Modbus application protocol specification v1.1b. 51 p. Dec. 2006

[30] Ortega, A.; Schweitzer, C.M.; Akira Shinoda, A. Performance analysis of smart grid communication protocol DNP3 over TCP/IP in a heterogeneous traffic environment IEEE Colombian Conference on Communications and Computing (COLCOM), 2013

[31] Mackiewicz, R.E. Overview of IEC 61850 and benefits. Power Engineering Society General Meeting, 2006. IEEE

[32] Mohagheghi, S.; Stoupis, J.; Wang, Z. Communication protocols and networks for power systems - current status and future trends. IEEE/PES Power Systems Conference and Exposition, 2009. PSCE '09.

[33] MODICON, Inc., Industrial Automation Systems. Modicon Modbus protocol reference guide. June 1996.

[34] http://www.mathworks.com/matlabcentral/fileexchange/36022-rs485-modbus-communication-with-jld416pva-power-meter/content/append_crc.m

[35] DS1104 R&D Controller Board. Hardware Installation and Configuration. Release 2013-A – May 2013. dSpace.

[36] PowerLogic G3200 Modbus-to-IEC61850 Server. User's Manual. Schneider Electric. 12.2010.

[37] IEC TEC57 "Communication networks and systems in substations – Part 7-2: Basic communication structure for substation and feeder equipment – Abstract communication service interface (ACSI)" IEC Standard IEC61850-7-2:2003(E).

[38] Hubert Kirrman. Introduction to IEC 61850 substation communication standard. ABB Switzerland Ltd, Corporate Research, ABBCH-RD Project Red61850, CRID 30254, C1-069-03/00. 2004.

[39] IEC TEC57 "Communication networks and systems in substations – Part 6: Configuration description language for communication in electrical substations related to IEDs" IEC Standard IEC61850-6:2003(E).
[40] M. Adamiak, D. Baigent, R. Makiewicz. IEC 61850 Communication Networks and Systems in Substations: An Overview for Users.8th Edition. July 2009.
[41] CET850 IEC 61850 configuration software. User's manual. Schneider Electric. 01. 2013.
[42] IEC TEC57 "Communication networks and systems in substations – Part 7-4: Basic communication structure for substation and feeder equipment – Compatible logical node classes and data classes" IEC Standard IEC61850-7-4:2003(E).
[43] https://sourceforge.net/projects/iedexplorer/

Appendix A: MATLAB/Simulink Modbus device model

The Fig. 1 depicts the overall Modbus model while the Fig. 2 shows the Trigger Subsystem comprising the response message function and DS1104 RS-422/485 transmitter block.

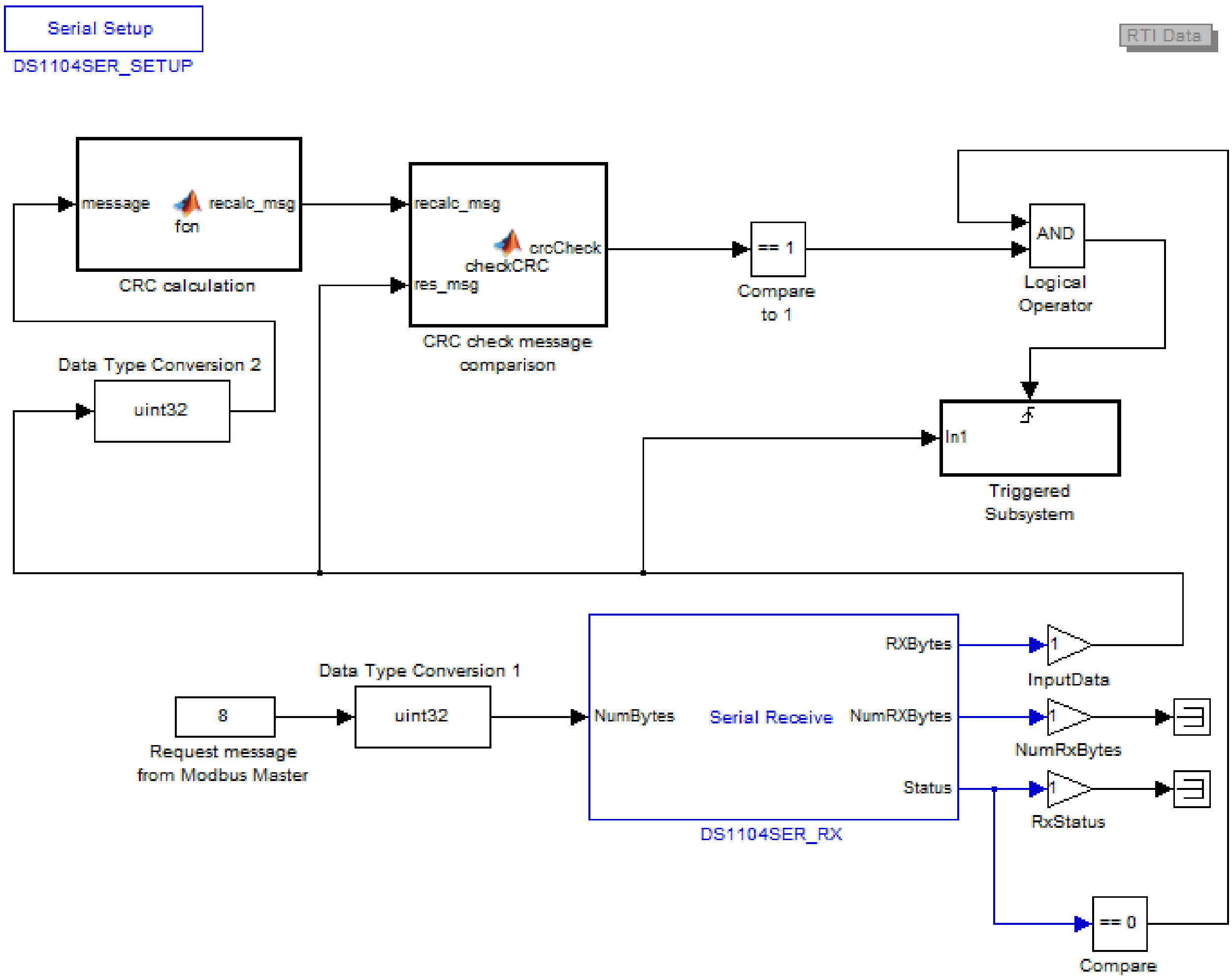


Figure 1 – Simulink Modbus device model.

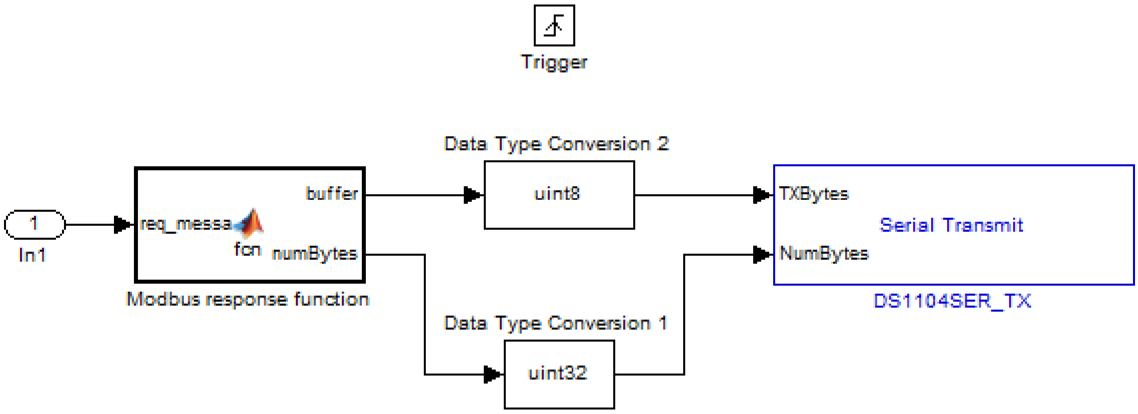


Figure 2 – Triggered subsystem for Modbus response function.

Appendix B: Modbus response function code

The part of the code for calculation of CRC is identical to that one used in CRC calculation function in Modbus device model except the counter N equals to 6 – the number of received bytes in request message without two bytes for CRC check.

```
function [buffer,numBytes] = fcn(req_message)

%Parameters
slave_ID = 1;

%Data registers
data = zeros(20,1);
data(1:20) = [1 12 33 45 58 61 72 84 93 10 14 28 36 44 57 62 73 81 93
15];

%Define buffer for response message and set to zero
buffer = uint32(zeros(25,1));

%Decode request
ID = req_message(1);
fcn = req_message(2);
addr = req_message(4);
addr_range = req_message(6)*2;

%Generate message
if ID == slave_ID
    buffer(1) = ID;
    buffer(2) = fcn;
    buffer(3) = addr_range;

    for j=1:addr_range
       buffer(3+j) = data(addr+j-1);
    end

    %Add CRC check
    N = addr_range+3;
    crc = uint32(hex2dec('ffff'));
    polynomial = hex2dec('a001');

    for i = 1:N
        crc = bitxor(crc,buffer(i));
        for j = 1:8
            if bitand(crc,1)
                crc = bitshift(crc,-1);
                crc = bitxor(crc,polynomial);
            else
                crc = bitshift(crc,-1);
            end
        end
    end

    lowByte = bitand(crc,hex2dec('ff'));
    highByte = bitshift(bitand(crc,hex2dec('ff00')),-8);

    %Add CRC to buffer

    buffer(addr_range+4) = lowByte;
```

```
    buffer(addr_range+5) = highByte;

    numBytes = double(addr_range)+5;

else
    numBytes = 0;
end
```

Appendix C: The CID file for G3200 Modbus-to-IEC 61850 server 'G3200-DS1104'


```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<SCL xmlns="http://www.iec.ch/61850/2003/SCL"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance">
  <Private type="SchneiderElectric-SFT-Key">3AA0C90DF02517AC37CA974A5DB771DE</Private>
  <Private type="SchneiderElectric-SFT-EditTime">2016-06-06 19:55:00</Private>
  <Private type="SchneiderElectric-SFT-Version">2.1.0</Private>
<Header id="My Project Id" nameStructure="IEDName" revision="0" toolID="CET850 v2.0"
version="2">
    <History>
  <Hitem revision="0" version="V1" what="Draft 1" when="2016-05-19" who="CET850Config"/>
    </History>
  </Header>
  <Communication>
    <SubNetwork name="lab_test">
      <ConnectedAP apName="AP1" iedName="G3200">
        <Address>
          <P type="IP" xsi:type="tP_IP">169.254.0.10</P>
          <P type="IP-SUBNET" xsi:type="tP_IP-SUBNET">255.255.0.0</P>
          <P type="IP-GATEWAY" xsi:type="tP_IP-GATEWAY">0.0.0.0</P>
          <P type="OSI-PSEL" xsi:type="tP_OSI-PSEL">00000001</P>
          <P type="OSI-SSEL" xsi:type="tP_OSI-SSEL">0001</P>
          <P type="OSI-TSEL" xsi:type="tP_OSI-TSEL">0001</P>
        </Address>
      </ConnectedAP>
    </SubNetwork>
  </Communication>
  <IED configVersion="1.0" desc="IEC61850 server for modbus devices"
manufacturer="Schneider Electric" name="G3200" type="G3200 Generic server">
    <Private type="SchneiderElectric-IED-Type">G3200</Private>
    <Services>
      <DynAssociation/>
      <SettingGroups/>
      <GetDirectory/>
      <GetDataObjectDefinition/>
      <DataObjectDirectory/>
      <GetDataSetValue/>
      <DataSetDirectory/>
      <ConfDataSet max="32" modify="true"/>
      <ReadWrite/>
      <ConfReportControl max="48"/>
      <GetCBValues/>
      <ReportSettings bufTime="Dyn" cbName="Conf" datSet="Conf" intgPd="Dyn"
optFields="Dyn" rptID="Dyn" trgOps="Dyn"/>
      <FileHandling/>
      <ConfLNs fixLnInst="true" fixPrefix="true"/>
    </Services>
    <AccessPoint clock="false" name="AP1" router="false">
      <Server timeout="30">
        <Private type="SchneiderElectric-SFT-Timeout">1000</Private>
        <Private type="SchneiderElectric-SFT-Retries">3</Private>
        <Authentication certificate="false" none="true" password="false" strong="false"
weak="false"/>
        <LDevice desc="G3200 Log Dev" inst="LD0">
          <Private type="SchneiderElectric-SFT-IcdFileName">SE_ECI_G3200-
F01_E1V01.icd</Private>
          <Private type="SchneiderElectric-SFT-IedVersion">0</Private>
          <Private type="SchneiderElectric-SFT-IedName">Gtw</Private>
          <Private type="SchneiderElectric-SFT-IedFamily">G3200</Private>
          <Private type="SchneiderElectric-SFT-IedAppli">G3200</Private>
          <Private type="SchneiderElectric-IED-DevModel">00:GTW</Private>
          <Private type="SchneiderElectric-IED-MdbAddr">255</Private>
          <LN0 desc="General" inst="" lnClass="LLN0" lnType="SE_LLN0_G3200_V001">
            <DOI name="Mod">
              <Private type="SchneiderElectric-IED-PntRef">T:IO;L:P:305</Private>
            </DOI>
          </LN0>
```

```
          <LN desc="Device" inst="1" lnClass="LPHD" lnType="SE_LPHD_G3200_V001"
prefix="">
            <DOI name="Proxy">
              <Private type="SchneiderElectric-IED-PntRef">T:SS;P:304</Private>
            </DOI>
          </LN>
        </LDevice>
        <LDevice desc="dSPACE_DS1104" inst="DS1104">
          <Private type="SchneiderElectric-IED-DevModel">UNK:dSPACE</Private>
          <Private type="SchneiderElectric-IED-CommType">JBUS</Private>
          <Private type="SchneiderElectric-IED-MdbAddr">1</Private>
          <Private type="SchneiderElectric-IED-MdbTbl">1:A:M</Private>
          <LN0 desc="General" inst="" lnClass="LLN0" lnType="SE_LLN0_Easergy_V001">
 <DataSet desc="Default status reporting dataset" name="StDs">
   <FCDA doName="Loc" fc="ST" ldInst="DS1104" lnClass="LLN0"/>
   <FCDA doName="PhyHealth" fc="ST" ldInst="DS1104" lnClass="LPHD" lnInst="1" prefix=""/>
   <FCDA doName="PwrDn" fc="ST" ldInst="DS1104" lnClass="LPHD" lnInst="1" prefix=""/>
 </DataSet>
 <DataSet desc="Default measurands reporting dataset" name="MxDs">
   <FCDA doName="MdbHreg1" fc="MX" ldInst="DS1104" lnClass="MMXU" lnInst="1" prefix=""/>
   <FCDA doName="MdbHreg2" fc="MX" ldInst="DS1104" lnClass="MMXU" lnInst="1" prefix=""/>
   <FCDA doName="MdbHreg3" fc="MX" ldInst="DS1104" lnClass="MMXU" lnInst="1" prefix=""/>
   <FCDA doName="MdbHreg4" fc="MX" ldInst="DS1104" lnClass="MMXU" lnInst="1" prefix=""/>
   <FCDA doName="MdbHreg5" fc="MX" ldInst="DS1104" lnClass="MMXU" lnInst="1" prefix=""/>
   <FCDA doName="MdbHreg6" fc="MX" ldInst="DS1104" lnClass="MMXU" lnInst="1" prefix=""/>
   <FCDA doName="MdbHreg7" fc="MX" ldInst="DS1104" lnClass="MMXU" lnInst="1" prefix=""/>
   <FCDA doName="MdbHreg8" fc="MX" ldInst="DS1104" lnClass="MMXU" lnInst="1" prefix=""/>
   <FCDA doName="MdbHreg9" fc="MX" ldInst="DS1104" lnClass="MMXU" lnInst="1" prefix=""/>
   <FCDA doName="MdbHreg10" fc="MX" ldInst="DS1104" lnClass="MMXU" lnInst="1" prefix=""/>
  </DataSet>
  <ReportControl bufTime="100" buffered="true" confRev="1" datSet="StDs" desc="Default
Status Report" intgPd="0" name="brcbST" rptID="StRpt">
   <TrgOps dchg="true" dupd="false" period="true" qchg="true"/>
   <OptFields bufOvfl="false" configRef="true" dataRef="true" dataSet="true"
entryID="true" reasonCode="true" segmentation="false" seqNum="true" timeStamp="true"/>
   <RptEnabled max="2"/>
  </ReportControl>
  <ReportControl bufTime="500" buffered="true" confRev="1" datSet="MxDs" desc="Default
Status Report" intgPd="0" name="brcbMX" rptID="MxRpt">
   <TrgOps dchg="true" dupd="false" period="true" qchg="true"/>
   <OptFields bufOvfl="false" configRef="true" dataRef="true" dataSet="true"
entryID="true" reasonCode="true" segmentation="false" seqNum="true" timeStamp="true"/>
   <RptEnabled max="2"/>
  </ReportControl>
            <DOI name="Loc">
              <Private type="SchneiderElectric-IED-PntRef">T:SS;s:918</Private>
            </DOI>
            <DOI name="Mod">
              <Private type="SchneiderElectric-IED-PntRef">T:IO;L:P:305</Private>
            </DOI>
          </LN0>
          <LN desc="Device" inst="1" lnClass="LPHD" lnType="SE_LPHD_Easergy_V001"
prefix="">
            <DOI name="PhyNam">
              <DAI name="model" valKind="Set">
                <Val>T200</Val>
              </DAI>
              <DAI desc="can be freely used" name="location" valKind="Set">
                <Val>location</Val>
              </DAI>
            </DOI>
            <DOI name="PhyHealth">
              <Private type="SchneiderElectric-IED-PntRef">T:ES;s:923;L:P:303</Private>
            </DOI>
            <DOI name="Proxy">
              <Private type="SchneiderElectric-IED-PntRef">T:SS;L:P:304</Private>
            </DOI>
            <DOI name="PwrDn">
              <Private type="SchneiderElectric-IED-PntRef">T:SS;s:919</Private>
            </DOI>
            <DOI name="ZPwrDnDel">
```

```
              <Private type="SchneiderElectric-IED-PntRef">T:SS;s:924</Private>
            </DOI>
          </LN>
          <LN desc="Measurements" inst="1" lnClass="MMXU" lnType="SE_MMXU_Easergy_V001"
prefix="">
            <DOI name="MdbHreg1">
              <Private type="SchneiderElectric-IED-PntRef">T:MV;m:1:u1;L:P:32</Private>
            </DOI>
            <DOI name="MdbHreg2">
              <Private type="SchneiderElectric-IED-PntRef">T:MV;m:2:u1;L:P:32</Private>
            </DOI>
            <DOI name="MdbHreg3">
              <Private type="SchneiderElectric-IED-PntRef">T:MV;m:3:u1;L:P:32</Private>
            </DOI>
            <DOI name="MdbHreg4">
              <Private type="SchneiderElectric-IED-PntRef">T:MV;m:4:u1;L:P:32</Private>
            </DOI>
            <DOI name="MdbHreg5">
              <Private type="SchneiderElectric-IED-PntRef">T:MV;m:5:u1;L:P:32</Private>
            </DOI>
            <DOI name="MdbHreg6">
              <Private type="SchneiderElectric-IED-PntRef">T:MV;m:6:u1;L:P:32</Private>
            </DOI>
            <DOI name="MdbHreg7">
              <Private type="SchneiderElectric-IED-PntRef">T:MV;m:7:u1;L:P:32</Private>
            </DOI>
            <DOI name="MdbHreg8">
              <Private type="SchneiderElectric-IED-PntRef">T:MV;m:8:u1;L:P:32</Private>
            </DOI>
            <DOI name="MdbHreg9">
              <Private type="SchneiderElectric-IED-PntRef">T:MV;m:9:u1;L:P:32</Private>
            </DOI>
            <DOI name="MdbHreg10">
              <Private type="SchneiderElectric-IED-PntRef">T:MV;m:10:u1;L:P:32</Private>
            </DOI>
          </LN>
        </LDevice>
      </Server>
    </AccessPoint>
  </IED>
  <DataTypeTemplates>
    <LNodeType id="SE_LLN0_G3200_V001" iedType="" lnClass="LLN0">
      <DO name="Mod" transient="false" type="SE_Mod_G3200_V001"/>
      <DO name="Beh" transient="false" type="SE_Beh_G3200_V001"/>
      <DO name="Health" transient="false" type="SE_Health_G3200_V001"/>
      <DO name="NamPlt" transient="false" type="SE_LplLLN0_G3200_V001"/>
    </LNodeType>
    <LNodeType id="SE_LPHD_G3200_V001" iedType="" lnClass="LPHD">
      <DO name="PhyNam" transient="false" type="SE_Dpl_G3200_V001"/>
      <DO name="PhyHealth" transient="false" type="SE_Health_G3200_V001"/>
      <DO name="Proxy" transient="false" type="SE_Sps_G3200_V001"/>
    </LNodeType>
    <LNodeType id="SE_LLN0_Easergy_V001" iedType="" lnClass="LLN0">
      <DO name="Mod" transient="false" type="SE_Mod_Easergy_V001"/>
      <DO name="Beh" transient="false" type="SE_Beh_Easergy_V001"/>
      <DO name="Health" transient="false" type="SE_Health_Easergy_V001"/>
      <DO name="NamPlt" transient="false" type="SE_LplLLN0_Easergy_V001"/>
      <DO name="Loc" transient="false" type="SE_Sps_Easergy_V001"/>
    </LNodeType>
    <LNodeType id="SE_LPHD_Easergy_V001" iedType="" lnClass="LPHD">
      <DO name="PhyNam" transient="false" type="SE_Dpl_Easergy_V001"/>
      <DO name="PhyHealth" transient="false" type="SE_Ins_Easergy_V001"/>
      <DO name="Proxy" transient="false" type="SE_Sps_Easergy_V001"/>
      <DO name="PwrDn" transient="false" type="SE_Sps_Easergy_V001"/>
      <DO name="ZPwrDnDel" transient="false" type="SE_SpsEx_Easergy_V001"/>
    </LNodeType>
    <LNodeType id="SE_MMXU_Easergy_V001" iedType="" lnClass="MMXU">
      <DO name="Mod" transient="false" type="SE_Mod_Easergy_V001"/>
      <DO name="Beh" transient="false" type="SE_Beh_Easergy_V001"/>
      <DO name="Health" transient="false" type="SE_Health_Easergy_V001"/>
      <DO name="NamPlt" transient="false" type="SE_Lpl_Easergy_V001"/>
```

```
      <DO name="MdbHreg1" transient="false" type="SE_MvEx_Easergy_V001"/>
      <DO name="MdbHreg2" transient="false" type="SE_MvEx_Easergy_V001"/>
      <DO name="MdbHreg3" transient="false" type="SE_MvEx_Easergy_V001"/>
      <DO name="MdbHreg4" transient="false" type="SE_MvEx_Easergy_V001"/>
      <DO name="MdbHreg5" transient="false" type="SE_MvEx_Easergy_V001"/>
      <DO name="MdbHreg6" transient="false" type="SE_MvEx_Easergy_V001"/>
      <DO name="MdbHreg7" transient="false" type="SE_MvEx_Easergy_V001"/>
      <DO name="MdbHreg8" transient="false" type="SE_MvEx_Easergy_V001"/>
      <DO name="MdbHreg9" transient="false" type="SE_MvEx_Easergy_V001"/>
      <DO name="MdbHreg10" transient="false" type="SE_MvEx_Easergy_V001"/>
    </LNodeType>
    <DOType cdc="LPL" id="SE_LplLLN0_G3200_V001" iedType="">
      <DA bType="VisString255" count="0" dchg="false" dupd="false" fc="DC" name="vendor"
qchg="false" valKind="RO">
        <Val>Schneider Electric</Val>
      </DA>
      <DA bType="VisString255" count="0" dchg="false" dupd="false" fc="DC" name="swRev"
qchg="false" valKind="RO"/>
      <DA bType="VisString255" count="0" dchg="false" dupd="false" fc="DC" name="d"
qchg="false" valKind="RO"/>
      <DA bType="VisString255" count="0" dchg="false" dupd="false" fc="DC"
name="configRev" qchg="false" valKind="RO"/>
      <DA bType="VisString255" count="0" dchg="false" dupd="false" fc="EX" name="ldNs"
qchg="false" valKind="Spec">
        <Val>IEC 61850-7-4:2003</Val>
      </DA>
    </DOType>
    <DOType cdc="SPS" id="SE_Sps_G3200_V001" iedType="">
      <DA bType="BOOLEAN" count="0" dchg="true" dupd="false" fc="ST" name="stVal"
qchg="false" valKind="Spec"/>
      <DA bType="Quality" count="0" dchg="false" dupd="false" fc="ST" name="q"
qchg="true" valKind="Spec"/>
      <DA bType="Timestamp" count="0" dchg="false" dupd="false" fc="ST" name="t"
qchg="false" valKind="Spec"/>
    </DOType>
    <DOType cdc="INC" id="SE_Mod_G3200_V001" iedType="">
      <DA bType="Enum" count="0" dchg="true" dupd="false" fc="ST" name="stVal"
qchg="false" type="Beh" valKind="Spec"/>
      <DA bType="Quality" count="0" dchg="false" dupd="false" fc="ST" name="q"
qchg="true" valKind="Spec"/>
      <DA bType="Timestamp" count="0" dchg="false" dupd="false" fc="ST" name="t"
qchg="false" valKind="Spec"/>
      <DA bType="Enum" count="0" dchg="false" dupd="false" fc="CF" name="ctlModel"
qchg="false" type="CtlModel" valKind="RO">
        <Val>status-only</Val>
      </DA>
    </DOType>
    <DOType cdc="DPL" id="SE_Dpl_G3200_V001" iedType="">
      <DA bType="VisString255" count="0" dchg="false" dupd="false" fc="DC" name="vendor"
qchg="false" valKind="RO">
        <Val>Schneider Electric</Val>
      </DA>
      <DA bType="VisString255" count="0" dchg="false" dupd="false" fc="DC" name="model"
qchg="false" valKind="RO"/>
      <DA bType="VisString255" count="0" dchg="false" dupd="false" fc="DC"
name="location" qchg="false" valKind="RO"/>
    </DOType>
    <DOType cdc="INS" id="SE_Health_G3200_V001" iedType="">
      <DA bType="Enum" count="0" dchg="true" dupd="false" fc="ST" name="stVal"
qchg="false" type="Health" valKind="Spec">
        <Val>Ok</Val>
      </DA>
      <DA bType="Quality" count="0" dchg="false" dupd="false" fc="ST" name="q"
qchg="true" valKind="Spec"/>
      <DA bType="Timestamp" count="0" dchg="false" dupd="false" fc="ST" name="t"
qchg="false" valKind="Spec"/>
    </DOType>
    <DOType cdc="INS" id="SE_Beh_G3200_V001" iedType="">
      <DA bType="Enum" count="0" dchg="true" dupd="false" fc="ST" name="stVal"
qchg="false" type="Beh" valKind="Spec"/>
```

```
      <DA bType="Quality" count="0" dchg="false" dupd="false" fc="ST" name="q"
qchg="true" valKind="Spec"/>
      <DA bType="Timestamp" count="0" dchg="false" dupd="false" fc="ST" name="t"
qchg="false" valKind="Spec"/>
    </DOType>
    <DOType cdc="INS" id="Beh" iedType="">
      <DA bType="Quality" count="0" dchg="false" dupd="false" fc="ST" name="q"
qchg="false" valKind="Spec"/>
      <DA bType="Enum" count="0" dchg="false" dupd="false" fc="ST" name="stVal"
qchg="false" type=" stValEnum" valKind="Spec">
        <Val>on</Val>
      </DA>
      <DA bType="Timestamp" count="0" dchg="false" dupd="false" fc="ST" name="t"
qchg="false" valKind="Spec"/>
    </DOType>
    <DOType cdc="INS" id="Health" iedType="">
      <DA bType="Quality" count="0" dchg="false" dupd="false" fc="ST" name="q"
qchg="false" valKind="Spec"/>
      <DA bType="Enum" count="0" dchg="false" dupd="false" fc="ST" name="stVal"
qchg="false" type=" stValEnum" valKind="Spec">
        <Val>on</Val>
      </DA>
      <DA bType="Timestamp" count="0" dchg="false" dupd="false" fc="ST" name="t"
qchg="false" valKind="Spec"/>
    </DOType>
    <DOType cdc="INC" id="SE_Mod_Easergy_V001" iedType="">
      <DA bType="Enum" count="0" dchg="true" dupd="false" fc="ST" name="stVal"
qchg="false" type="Beh" valKind="Set"/>
      <DA bType="Quality" count="0" dchg="false" dupd="false" fc="ST" name="q"
qchg="true" valKind="Set"/>
      <DA bType="Timestamp" count="0" dchg="false" dupd="false" fc="ST" name="t"
qchg="false" valKind="Set"/>
      <DA bType="Enum" count="0" dchg="false" dupd="false" fc="CF" name="ctlModel"
qchg="false" type="CtlModel" valKind="RO">
        <Val>status-only</Val>
      </DA>
    </DOType>
    <DOType cdc="INS" id="SE_Beh_Easergy_V001" iedType="">
      <DA bType="Enum" count="0" dchg="true" dupd="false" fc="ST" name="stVal"
qchg="false" type="Beh" valKind="Set"/>
      <DA bType="Quality" count="0" dchg="false" dupd="false" fc="ST" name="q"
qchg="true" valKind="Set"/>
      <DA bType="Timestamp" count="0" dchg="false" dupd="false" fc="ST" name="t"
qchg="false" valKind="Set"/>
    </DOType>
    <DOType cdc="INS" id="SE_Health_Easergy_V001" iedType="">
      <DA bType="Enum" dchg="true" fc="ST" name="stVal" type="Health">
        <Val>Ok</Val>
      </DA>
      <DA bType="Quality" fc="ST" name="q" qchg="true"/>
      <DA bType="Timestamp" fc="ST" name="t"/>
    </DOType>
    <DOType cdc="LPL" id="SE_Lpl_Easergy_V001" iedType="">
      <DA bType="VisString255" count="0" dchg="false" dupd="false" fc="DC" name="vendor"
qchg="false" valKind="RO">
        <Val>Schneider Electric</Val>
      </DA>
      <DA bType="VisString255" count="0" dchg="false" dupd="false" fc="DC" name="swRev"
qchg="false" valKind="RO"/>
      <DA bType="VisString255" count="0" dchg="false" dupd="false" fc="DC" name="d"
qchg="false" valKind="RO"/>
    </DOType>
    <DOType cdc="LPL" id="SE_LplLLN0_Easergy_V001" iedType="">
      <DA bType="VisString255" count="0" dchg="false" dupd="false" fc="DC" name="vendor"
qchg="false" valKind="RO">
        <Val>Schneider Electric</Val>
      </DA>
      <DA bType="VisString255" count="0" dchg="false" dupd="false" fc="DC" name="swRev"
qchg="false" valKind="RO"/>
      <DA bType="VisString255" count="0" dchg="false" dupd="false" fc="DC" name="d"
qchg="false" valKind="RO"/>
```

```
      <DA bType="VisString255" count="0" dchg="false" dupd="false" fc="DC"
name="configRev" qchg="false" valKind="RO"/>
      <DA bType="VisString255" count="0" dchg="false" dupd="false" fc="EX" name="ldNs"
qchg="false" valKind="Set">
        <Val>IEC 61850-7-4:2003</Val>
      </DA>
    </DOType>
    <DOType cdc="DPL" id="SE_Dpl_Easergy_V001" iedType="">
      <DA bType="VisString255" count="0" dchg="false" dupd="false" fc="DC" name="vendor"
qchg="false" valKind="RO">
        <Val>Schneider Electric</Val>
      </DA>
      <DA bType="VisString255" count="0" dchg="false" dupd="false" fc="DC" name="model"
qchg="false" valKind="RO"/>
      <DA bType="VisString255" count="0" dchg="false" dupd="false" fc="DC"
name="location" qchg="false" valKind="RO"/>
    </DOType>
    <DOType cdc="INS" id="SE_Ins_Easergy_V001" iedType="">
      <DA bType="INT32" count="0" dchg="true" dupd="false" fc="ST" name="stVal"
qchg="false" valKind="Set"/>
      <DA bType="Quality" count="0" dchg="false" dupd="false" fc="ST" name="q"
qchg="true" valKind="Set"/>
      <DA bType="Timestamp" count="0" dchg="false" dupd="false" fc="ST" name="t"
qchg="false" valKind="Set"/>
    </DOType>
    <DOType cdc="MV" id="SE_MvEx_Easergy_V001" iedType="">
      <DA bType="Struct" count="0" dchg="false" dupd="false" fc="MX" name="mag"
qchg="false" type="analogVal" valKind="Set"/>
      <DA bType="Quality" count="0" dchg="false" dupd="false" fc="MX" name="q"
qchg="true" valKind="Set"/>
      <DA bType="Timestamp" count="0" dchg="false" dupd="false" fc="MX" name="t"
qchg="false" valKind="Set"/>
      <DA bType="INT32U" count="0" dchg="false" dupd="false" fc="CF" name="db"
qchg="false" valKind="Set"/>
      <DA bType="VisString255" count="0" dchg="false" dupd="false" fc="EX" name="dataNs"
qchg="false" valKind="Set">
        <Val>SE_dataNs_V0001</Val>
      </DA>
    </DOType>
    <DOType cdc="SPS" id="SE_Sps_Easergy_V001" iedType="">
      <DA bType="BOOLEAN" count="0" dchg="true" dupd="false" fc="ST" name="stVal"
qchg="false" valKind="Set"/>
      <DA bType="Quality" count="0" dchg="false" dupd="false" fc="ST" name="q"
qchg="true" valKind="Set"/>
      <DA bType="Timestamp" count="0" dchg="false" dupd="false" fc="ST" name="t"
qchg="false" valKind="Set"/>
    </DOType>
    <DOType cdc="SPS" id="SE_SpsEx_Easergy_V001" iedType="">
      <DA bType="BOOLEAN" count="0" dchg="true" dupd="false" fc="ST" name="stVal"
qchg="false" valKind="Set"/>
      <DA bType="Quality" count="0" dchg="false" dupd="false" fc="ST" name="q"
qchg="true" valKind="Set"/>
      <DA bType="Timestamp" count="0" dchg="false" dupd="false" fc="ST" name="t"
qchg="false" valKind="Set"/>
      <DA bType="VisString255" count="0" dchg="false" dupd="false" fc="EX" name="dataNs"
qchg="false" valKind="Set">
        <Val>SE_dataNs_V0001</Val>
      </DA>
    </DOType>
    <DAType id="analogVal" iedType="">
      <BDA bType="FLOAT32" count="0" name="f" valKind="Set"/>
    </DAType>
    <EnumType id="Beh">
      <EnumVal ord="1">on</EnumVal>
      <EnumVal ord="2">blocked</EnumVal>
      <EnumVal ord="3">test</EnumVal>
      <EnumVal ord="4">test/blocked</EnumVal>
      <EnumVal ord="5">off</EnumVal>
    </EnumType>
    <EnumType id="Health">
      <EnumVal ord="1">Ok</EnumVal>
```

```
      <EnumVal ord="2">Warning</EnumVal>
      <EnumVal ord="3">Alarm</EnumVal>
    </EnumType>
    <EnumType id="CtlModel">
      <EnumVal ord="0">status-only</EnumVal>
      <EnumVal ord="1">direct-with-normal-security</EnumVal>
      <EnumVal ord="2">sbo-with-normal-security</EnumVal>
      <EnumVal ord="3">direct-with-enhanced-security</EnumVal>
      <EnumVal ord="4">sbo-with-enhanced-security</EnumVal>
    </EnumType>
  </DataTypeTemplates>
</SCL>
```