



Österbottens förbund
Pohjanmaan liitto

Regional Council
of Ostrobothnia



Vaasan yliopisto
UNIVERSITY OF VAASA

Light-Weight IEC 61850 IEDs

Reducing complexity

SESP

PROJECT REPORT

Mike Mekkanen

29.04.2019

Contents

1	THE NEW SUBSTATION STANDARD IEC 61850.....	1
1.1	IEC61850standard Overview and Basic Concepts	2
1.2	IEC 61850 Impact and Benefits.....	4
1.3	IEC 61850 Information Model	4
1.4	Virtualization of the Physical Devices and LN, LD Concept... ..	6
1.5	Communication and Logical Interfaces within SAS	8
1.6	IEC 61850 Communication Protocols	9
1.7	Abstract Communication Service Interface (ACSI)	10
1.8	GSSE, GOOSE and SV	12
1.9	Manufacturing Messaging Specification (MMS)	13
1.10	Substation Configuration description Language (SCL).....	14
1.11	IEC 61850 Technical Challenges Implementation Issues	15
1.11.1	System Configuration Challenging.....	16
2	LIGHT-WEIGHT IEC 61850	18
2.1	Light- Weight IEC 61850 reducing complexity	19
2.2	Light-Weight IEC 61850 implementation process	20
2.2.1	Setting up the device	20
2.3	libiec61850 library	23
2.3.1	Overview of libiec61850 library	24
2.3.2	libiec61850 Installation.....	25
3.	CONCLUSION	27
	REFERENCES	28
	APPENDICES	30

Figures

Figure 1.	Two standard merge.....	1
Figure 2.	Hierarchy of IEC 641850 data model.	5
Figure 3.	IEC 61850 Object Name Structure.	6
Figure 4.	Virtualization process.	7
Figure 5.	IEC 61850 logical grouping.....	8
Figure 6.	SAS levels and logical interfaces.....	9
Figure 7.	IEC 61850 application messages mapping to OSI layers.	10
Figure 8.	Abstract communication service interface concept.	11
Figure 9.	GOOSE service operation mechanism.....	12
Figure 10.	MMS concept.....	14
Figure 11.	DERs logical devices and logical nodes concepts.	18
Figure 12.	USB connection cable.....	21
Figure 14.	Prompt window	22

Figure 15. FTDI cable23
Figure 16. libiec61850 folders.26

Tables

Table 1. Scope of the first version of the IEC 61850 standard.3
Table 2. SCL file types and extinctions. 15

Abbreviations

ACSI	Abstract communication service interface
CSMA	Carrier sense multiple access
DER	Distributed Energy Resources
EPS	Electrical power system
GOOSE	Generic objective oriented substation events
GSSE	Generic service substation events
HM	Human machine interface
IEC	International Electrotechnical Commission
IED	Intelligent electronic devices
LD	Logical device
LN	Logical node
MMS	Manufacturing message interface
MU	Merging unit
PLC	Programmable logical controller
PMU	Phasor measurements unit
RTU	Remote terminal unit
SAS	Substation automation system
SCADA	Supervisory Control and Data Acquisitions
SCL	Substation configuration language
SCAM	Specific communication service mapping
SPS	Special protection scheme
SV	Sampled value
VMD	Virtual manufacturing device
XML	eXtensible Markup Language

1 THE NEW SUBSTATION STANDARD IEC 61850

Exchanging real-time information becomes a dominant task for any successive system operation. In substation, the real-time information needs to exchange speedily and accurately between substation as well as other substations devices that created demands to integrate and consolidate IEDs. This task requires a communication standard means of standardized the communication language between devices to facilitate interfaces, since the existing solution reaches their limits. In particular, the Electrical Power Research Institute EPRI and IEEE raise the concept of Utility Communication Architecture UCA in early 1990s. The idea behind the concept is to identify the requirements, structure and specific communication technologies that used to implement the standardization scheme in which that suitable for future extending. The first version was concentrated on the interfaces between the control centers as well as substation to the control center [1].

In 1994, the next phase of UCA is UCA 2.0 started to define the substation's communication bus. The UCA architecture comprised of data object on the top layer application layer. Service interface is the middle layer in which that based on its operation provide tasks such as defining, retrieving and logging of process data and in the bottom communication profile's layer. While, technical Committee 57 of the IEC starts its work on IEC61850 in 1996 for the same concept. The standard IEC61850 defined as the common international standard in 1997 when the two groups agreed to work together and merge their works. Resulting, the harmonization process is the current IEC61850 specifications that superset of UCA 2.0 besides offers additional features that published in as international standard in 2003. Figure 12 illustrates the merging process [2][3].

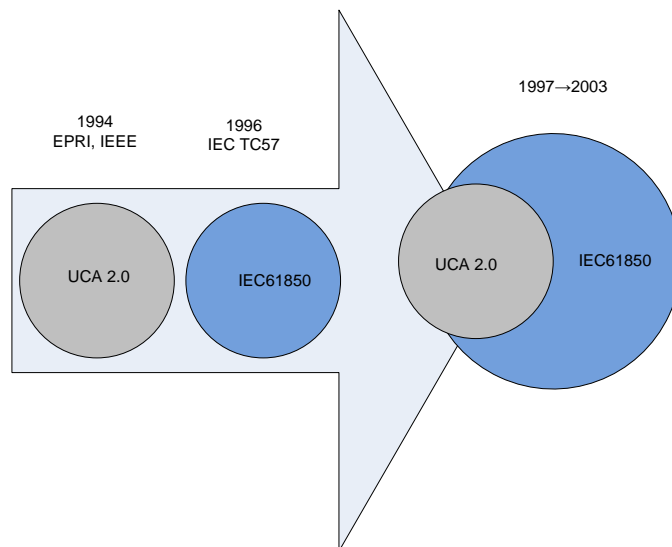


Figure 1. Two standard merge.

1.1 IEC61850standard Overview and Basic Concepts

Nowadays, IEC 61850 standard becomes one of the most promising and powerful solutions for the existing power industry limitations and expected to support energy systems evolutions. The key point it provides a uniform framework for all the related system levels. IEC61850 takes considerations for the all various aspects that are common at the substation site such as data models, communication solution, engineering and conformity over the channel. While organizing the data in terms of applications by means of syntax and semantics within the devices they did not specify it.

The main aspect that the IEC 61850 adopts with is the associated architectural construct “abstracting” data object definition and its services. These data objects and their associated services have been abstracted independently from any underlying protocol that supports a comprehensive set of substation functions and provides strong services in order to facilitate the energy system communication. The abstract definitions of the data object allows mapping it to any protocol that can meet the best data and service requirements as IEC 61850 standards did not specify any protocol. Therefore, IEC61850 specification can be capsule into three major focusing issues,

- 1 Standardizing the available information (data object model), substation functions (functional model) and IEDs name in which that provides to the IEDs a shared vocabulary that support the intended semantic meaning.
- 2 Standardizing different ways of the accessing scheme to the available data abstract communication services interface (*ACSI*). These ways are defined as services. Further, specification of the mapping scheme to the communication services and the data to a number of protocols.
- 3 Defined a language eXtensible Markup Language (XML) implemented to describe all the configuration information exchanged between IEDs, network and power system.

The scope of the first version of the IEC 61850 standard is composed of 10 major parts that together define the various aspects and the entire requirement that has to be fulfilled by SAS. The main goal is to achieve interoperability among the IEDs within the SAS as illustrated in Table 2 (IEC 61850 2003) [4].

Table 1. Scope of the first version of the IEC 61850 standard.

Part	Definitions
1	<i>Introduction and Overview</i> : it provides introduction to IEC 61850 and general overview to all parts.
2	<i>Glossary</i> : it gives definitions to the specific terms used in SAS.
3	<i>General Requirements</i> : it defines quality requirements based system operation.
4	<i>System and project management</i> : it specifies engineering service requirements.
5	<i>Communication Requirements for Functions and devices Models</i> : it defines the virtualizations aspect and its performance requirements.
6	<i>Configuration Description language for communication in Electrical Substation</i> : it specifies a file format for describing system configuration and relation between devices.
7	<i>Basic Communication Structure for Substation and feeder Equipment</i> :
7-1	<i>Principle and Models</i> : it defines the communication and information model principles also mapping scheme.
7-2	<i>Abstract Communication Service Interface (ACSI)</i> : it defines the cooperation of various devices.
7-3	<i>Common Data Classes (CDC)</i> : it defines the common attribute type and common data classes related to substation applications.
7-4	<i>Compatible logical nodes and data classes</i> : it specifies the data classes with regard to syntax and semantics.
8	<i>Specific Communication Service Mapping (SCSM)</i>
8-1	<i>Mapping to MMS (ISO/IEC 9506-1 & 2) and to (ISO/IEC 8802-3)</i> : it describes the communication mapping for the entire system.
9	<i>Specific Communication Services Mapping (SCSM)</i>
9-1	<i>Sampled Values over Serial Unidirectional Multi-drop Point-to-Point Link</i> : it describe the point-to-point unidirectional communication mapping services.
9-2	<i>Sampled values over ISO/IEC 8802-3</i> : it describes the SCSM for bus-type.
10	<i>Conformance testing</i> : it specifies the implementations conformance testing techniques and the declaring performance parameters measurements techniques.

1.2 IEC 61850 Impact and Benefits

The positive impact of the IEC 61850 standard in substation operation cost it is clearly known in terms of increasing the power quality and reducing the outage response. However, this goal requires paying attentions on how to implement the IEC 61850 standard in order to build, integrate and operate the SAS. While, the energy systems are moving into the digital environment where the huge amounts of real-time information data are available that allows a new kind of calculation for higher level substation functions based protection and automation. IEC 61850 standard is unique, which designed from the base up to operate over the state-of-the-art technologies. It provides a novel set of functionalities which are not exists within the legacy SAS operation. From IEC 61850 implementation point view that attempts to utilize the fullest functionality, numerous of benefits will be achieved such as [5] [6].

- 1 Open system for protection automation and data sharing by using the standard representation for the whole energy system objects and a common technology infrastructure to eliminate the procurement ambiguity.
- 2 Interoperability between devices from various manufacturers within the energy system and the ability to configure the system with the available configuration system tools independently from the on-site manufacturer support.
- 3 Secure and dependable overall system by means of several techniques that allows flexible information transfers
- 4 Reduce operation and maintenance overall system costs.
- 5 Flexible and expandable functions, easy adaptation by means of self-description in a standardized manner.

1.3 IEC 61850 Information Model

Information model in IEC 61850 is hierarchically structured where logical nodes are the essential elements of this model. Modeling is performed in the standardized way in order to provide an opportunity for interoperability among IEDs within the system. Where, IEC 61850 models virtualizes the real devices in energy systems to logical devices where the logical nodes are hosed by. LN represents a specific function within a device where the function can be split into multiple LNs that can be

located into various physical devices which in terms of this infrastructure are called distributed functions. Therefore, LN may consider as the small part of function that possesses the capability to exchange data. The hierarchy information modeling can be classified into five levels in order to perform the applications within SAS. The levels have an inheritable relationship. In substation one or more physical devices can be defined. The individual physical device may have zero or multiple servers, typically at least one server. Server object based on the hierarchical data model is located on the topmost as illustrates in Figure 11 which may have one or multiple access points. In compliance with data model the server class can be described as a collection of the objects below it in the data model. Further, it possesses methods that can create all the underlying objects. Based on server implementation perspective, in order to access the objects of the data model from the server, the list of object references and corresponding references to instances of the objects are stored in the server class which allows performing this task. Defining logical device (LD) and server is freely up to the manufacturers or administrators of the substations. Standardization is the way that the LNs are predefined with. This fact makes LNs the most crucial point of the information model. In terms of this restriction interoperability between different IEDs from various manufacturers can be achieved (IEC 61850-7-2 2003) (IEc 61850-7-4 2003) [7].

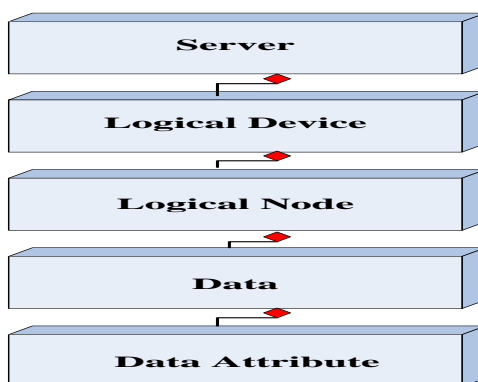


Figure 2. Hierarchy of IEC 61850 data model.

Consequently, the data are grouped into 13 different groups. According to these groups, all the substation operation data can be assigned to one of these groups, for instances the metering and measurements function data group is started with “M” and protection function data group is started with “P” etc. In addition the groups are further divided into LNs which are named based on its associated services that are logically related to functions in substation. LNs can be defined into 86 different types. For instance, “Switchgear” function group which started with “X” comprises two different LNs “XCBR” and “XSWI”. Each LN is constructed by seven categories of data classes such as status information, measured information etc. Each one

from the data classes consists from a number of data attributes, as we have 355 different data attributes that have a specific name, specific type and various purposes as illustrated in Figure 12. Browsing to individual objects may be easy upon this model from the fact that the data object is named by means of its place and path through the information tree model. For instance, as in Figure 12 from the left to the right the first name is the device name, the second part represents the LN. The third part is a functional constraints which is used to group the individual attribute that has predefined function based on its functionality, as in our example “ST” stand for status attributes. Last “Pos” attribute represents the position of the circuit breaker and “q” contains the quality to the value of the status that has been sent (IEC 61850-5 2003) (IEC 61850-7-1 2003).

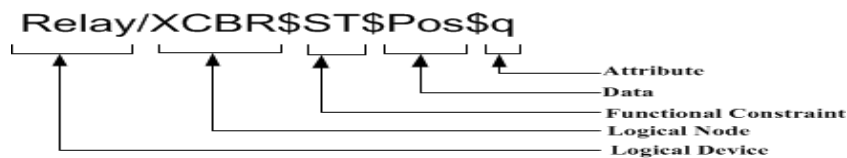


Figure 3. IEC 61850 Object Name Structure.

1.4 Virtualization of the Physical Devices and LN, LD Concept

The IEC 61850 standard provides the LN concept that can be considered as the major role of the standard which by means of virtualization the standard capable of virtualizes the substation physical device into the data model. This data model consists of a number of LNs. These LNs by a reasonable distributed allocation can build the logical device. The LD is often initiated in one physical device that cannot be distributed. Specific function in SAS is often performed by different physical devices which called distributed function and the devices are called distributed devices. In order to implement the distributed function correctly sharing information is required among these devices. The interfaces are performed based on the IEC 61850 standard communication services. These interfaces follow the predefined rules and the assigned performance requirements. Where, the rules allow interoperability between devices from various manufacturers. Figure 13 illustrates the modeling approach where the real physical substation devices in the right side are modeled into a virtual model. The virtual model contains the logical devices that hosted the logical nodes which encapsulating the real device and services. The data model and services with associated information are mapped to a network communication protocol such as manufacturing message specification (MMS), transmission control protocol TCP/IP, Ethernet etc. (IEC 61850-7-1 2003).

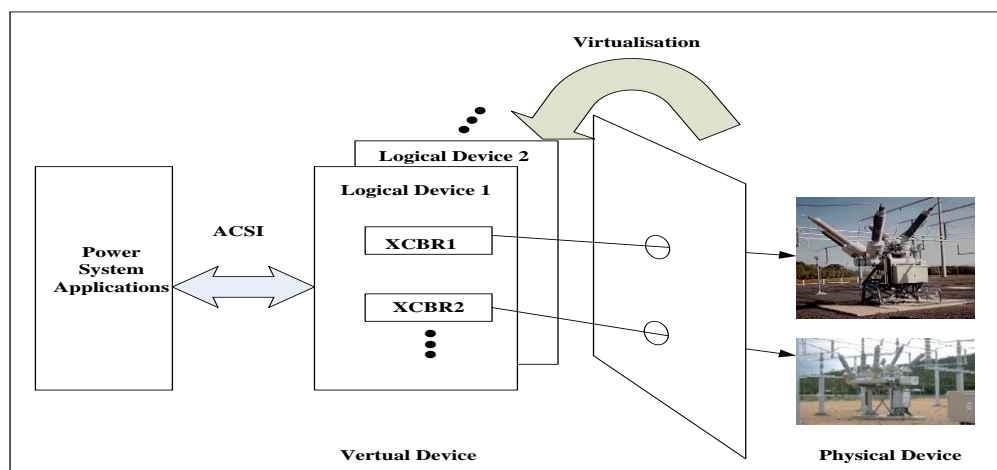


Figure 4. Virtualization process.

The LNs based on their functionality consist of the list of data associated with dedicated data attributes. These data have the predefined structure (syntax and semantic). The LNs with the associated data are crucial for the description and information sharing in the energy system protection and automation. Figure 14 illustrates more clearly the concepts of the LD, LNs, common data classes and data attribute that map to the real world. The virtualization task start with specifying the container the physical device which is containing one or more LDs, each LD may contain one or more LNs, were each LN may comprise a set of common data classes. The concept behind introducing the common data classes was to group and construct larger data object. The data classes may contain a set of data attributes. The terms logical device, logical nod and data object are all virtual which represents real data. This real data has been used by the energy protection and automation systems over a reliable high-speed communication system network that link between the defined physical devices. Further, the information modeling and the sharing data are defined independently from the programing software, operating system and storage device the fact that provides the ability to use the state-of-the-art technologies [8].

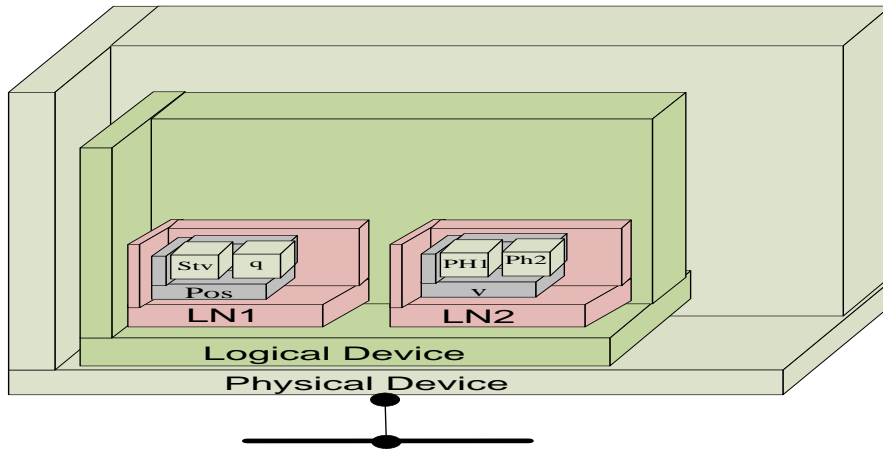


Figure 5. IEC 61850 logical grouping.

1.5 Communication and Logical Interfaces within SAS

Substation architecture can be divided into three various levels' station, bay and process level. These levels have been separated based on their functionalities. However, they are really well interconnected based on several logical interfaces. Figure 15 illustrates the three levels besides the associated logical interfaces numbers, from bottom to up. The process level has been linked to the bay level by interface number four and five. Those logical interfaces provide the ability to exchange control commands and information data. Usually, primary apparatus such as circuit breaker, transformers, switchgear etc. are located in process level that may also have IEDs such as intelligent sensors and actuators. The input and output messages of these apparatuses are basically consisted of information such as the transformer voltage and current values as an analogue signal format, and controls commands from the bay relays as a binary signal format. In order to convert the analogue signal into a digitalized standard packet form is done by so-called Merging Unit (MU). MU may locate in the yard besides or integrated with the instrument transformer and should contain logical nodes voltage and current transformer (TVTR, TCTR). This conversion has many advantages for instance, increasing the reliability of the protection and automation systems in terms of broadcasting the data and makes it available for the entire system. Further, reducing the overall SAS cost in terms of limiting the copper wires' utilization. This reduction can be achieved by replacing the electrical wires' connection by the logic interfaces. The output packet stream samples that the MU may transfer over the point-to-point type connection to any IED or broadcasted over the local area network (LAN) in the similar manner as GOOSE messages (IEC 61850-9-1 2003).

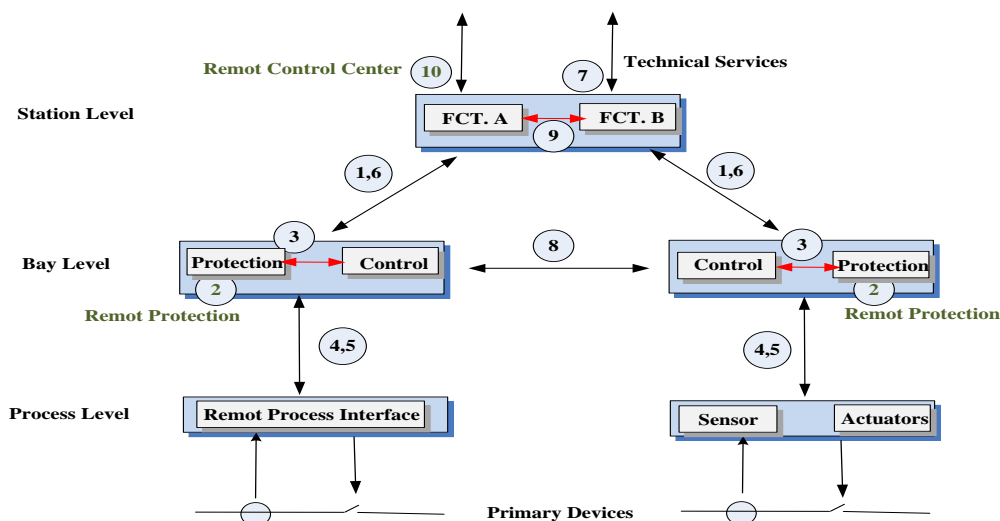


Figure 6. SAS levels and logical interfaces.

Within the bay level IEDs such as protection and control units provides protection functions by means of implementing the functions output signals initiated on one bay that may performs an action on the primary apparatus of the primary level one bay. The IEC 61850 standard offers the features that the SAS functions may be freely or logically allocated between IEDs. Nowadays, state-of-the-art IEDs may provide multiple functions such as monitoring, protection and control within an individual IED. On the other hand, different functions within individual bay unit have the ability to share data as illustrated by interfaces number three. While, various bays in terms of the horizontal communication GOOSE messages are able to communicate among each other by interface number eight. Interfaces four and five illustrate the communication between the bay level and process level and interfaces one and six illustrate the communication between bay level and station level. The station level equipment such as human machine interface (HMI), station workplace, alarm unit and features, remote control center, database etc. communicates with the bay level within the interfaces one to exchange protection-data and six to exchange control-data. Further, interface nine illustrates the sharing data within the station level. Interfaces seven and ten illustrate the sharing data outside the local station operator. These tasks have been carried out within the second version of the standard IEC 61850-7-420.

1.6 IEC 61850 Communication Protocols

The main concept of the IEC 61850 standard is that the data object model and services are separated from the communication ISO/OSI layers stack. This approach

gives the opportunity to implement the state-of-the-art of the communication technologies. The mainstream technologies are the communication schemes that been used over the ISO/OSI reference model. The ISO/SOI stack is upon the concept of layering communication functionality that consists of seven layers as illustrates in Figure 16. Layers one and two are the Ethernet physical and link layer, Layers three and four are the TCP/IP layer and layer six and seven are the MMS layer. The IEC 61850 object models are mapped over various different layers in terms of their services and requirements. The object model based on Client/Server services ACSI is mapped to the 5-7 MMS layers. Whereas the high speed time critical messages such as SV, the status indications blocking trip commands and GOOSE are mapped directly to the Ethernet link layer (IEC 61850-8-1 2003).

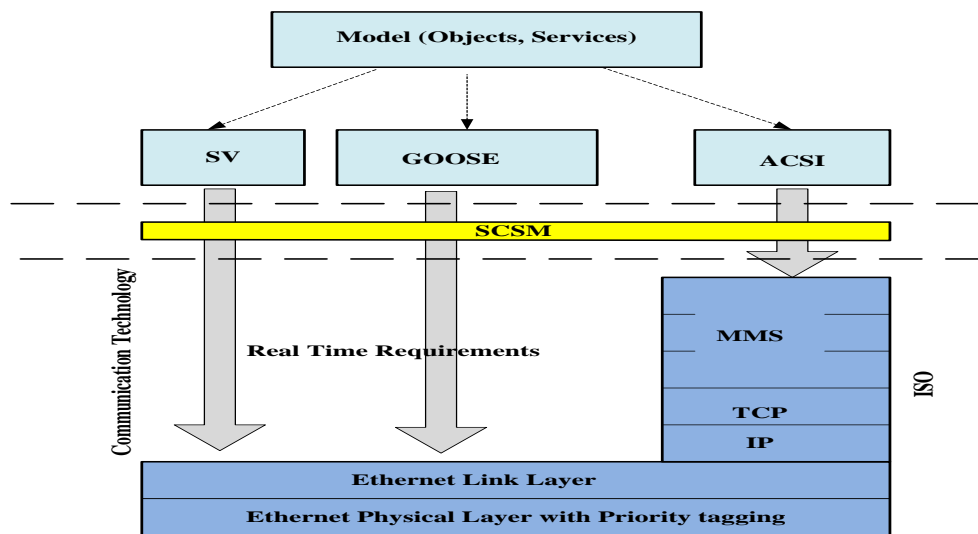


Figure 7. IEC 61850 application messages mapping to OSI layers.

1.7 Abstract Communication Service Interface (ACSI)

IEDs are described by the standardized method that allows all IEDs to share data by means of identical structure that related to their functions. From the network behavior perspective the ACSI provides the specification for the basic model that represents the definition for the substation-specific information models. Further, it specifies a set of information exchange service model and the response to those services. This specification allows various IEDs to exhibit identical behavior. The abstraction technique that has been adopted by IEC 61850 is one of the most significant features in which that separate the SAS application from the underlying

communication protocol and operating systems as illustrated in Figure 17. (Adamiak etc. 2009).

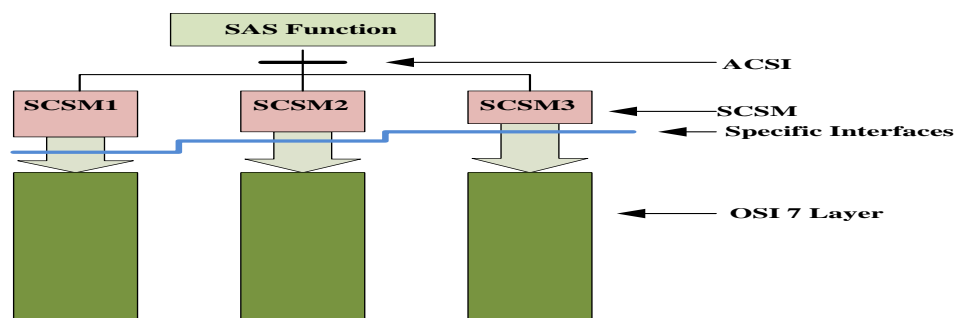


Figure 8. Abstract communication service interface concept.

An ACSI concept has two approaches. Firstly, based on the basic information model, that only aspects of real devices or real functions that are visible and accessible over the network are modeled, resulting to hieratical class models such as LOGICAL-DEVICE, LOGICAL-NODE, DATA and DataAttribute. Secondly, upon the exchange service model, the abstract can be defined from the aspect of the way that the devices able to share the information in terms of the definition focuses on aspects of the purpose of the services, instead of describing how the services are built (IEC 61850-7-2 2003). In a real implementation the basic information model and services' model are mapped into an existing communication stack. The mapping schemes are achieved through the specific communication service mapping (SCSM). In IEC61850 two mapping scheme are specified (IEC61850-9-1 2003) for the transmission of the SV and (IEC61850-8-1 2003) for the transmission of wide station events and all other communication services. Further, the ACSI provides abstract interfaces that describe communication between a client and server. This type of interfaces can be used for the real-time application such as for instance data access, data recovery, device control, publisher/subscriber applications, event reporting and transferring, self-description, self-heling, data typing and data reading. Further, it describes communication between applications in one device as a publisher to many applications in various devices as a subscriber for fast and reliable system-wide events distribution such as GOOSE, GSE and SV. The ACSI interfaces which are defined above allowed the client to observe the data model, to get and set data, manipulate data-sets, to log etc. by means of calling method such as GetDataValues and SetDataValues. These methods in programing language are reasonable traditional methods with assigned arguments led to return output values (Pedersen 2010). Consequently, the ACSI model defined the services set within the client application, while, within the server application defined the response to the requested services. It also defined the concept of application associations. This feature represents the controlling accesses' mechanism to the object within a device.

In order to restrict particular devices' visibility different access control schemes can be used.

1.8 GSSE, GOOSE and SV

The Generic Substation Events (GSE) service model is one of the main features of IEC 61850 that offers the fast and reliable real time application to deliver the SAS data values over the communication system network. The GSE service model is built over the concept of independency decentralization. It uses multicast/broadcast services model based on an efficient method. This multicast/broadcast services' model provides the simultaneous distribution for the SAS event values to whole SAS subscriber IEDs. The generic substation event distributions also support peer-to-peer and client/server communication models. IEC61850-7-2 defined two control classes and the structures of two messages such as,

- 1 GOOSE, supports wide variety of the SAS common data such as analog, binary and integer value data type grouped by the Data-Set.
- 2 GSSE, support only status change information events, fixed structure binary event, bit pairs.

Therefore, the type of sharing information is the major difference between the GOOSE and GSSE services. The flexible GOOSE model is used by all new systems which convey a wide range of messages, binary and analogue data. Whereas, GSSE is the first and older, that delivers just binary values within their messages. Figure 18 illustrates the GOOSE model. The message is based on publisher/subscriber exchanging mechanism method. From the implementation point view, at the publisher side the values are written in the local buffer, were at receiving side the subscribers read the values from the local buffer. The local buffers of the subscribers are updated by the communication system where the GSE control class had been used as a controller for the procedure from the publisher side (IEC 61850-7-2 2003).

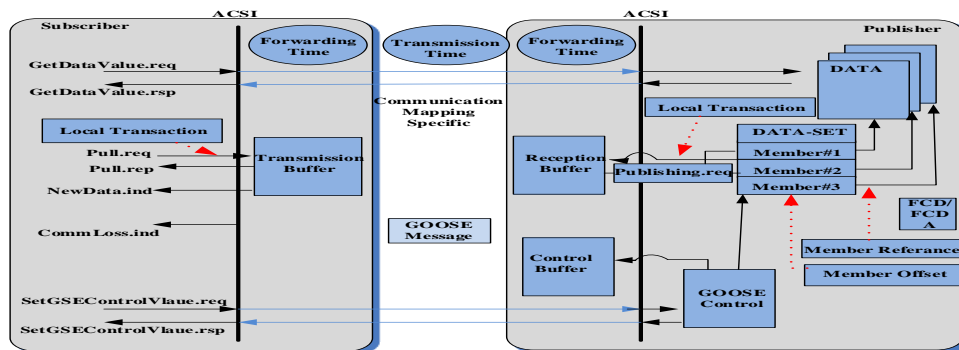


Figure 9. GOOSE service operation mechanism.

The substation IEDs know the changing status and when the last status changing occur upon receiving the GOOSE messages which contain all needed information. Further, the local timer of the subscriber can be set based on the related time of the latest status change event. At this point, GOOSE has been identified as one of the highest speed times critical messages within the SAS. Therefore, GOOSE messages are mapped directly to the Ethernet layer in order to support the real-time operation requirements. Typical protection events such as trip, interlock and status indication are recognized as high priority services in which that the processing time must be less than the quarter of a cycle. For instance, the message transmission time for the 50Hz cycling frequency system is specified as <4ms. Moreover, since the GOOSE messages are directly mapped over the Ethernet layer which does not guarantee the delivering services. The retransmission scheme should be used to ensure that the messages are received based on a multicast transmission. Lastly, since GOOSE message can serve various applications with different performance requirements and varies data types it may be considered as a flexible tool, [9].

Another high speed time critical messages within the SAS are the SV messages that are used to deliver measured values from switchyard to bay IEDs in a digitized form. Multicast is the transmission scheme that the measured values at one location can be delivered to any number of subscribers. Intelligent instrument or the MU is responsible for digitalization process. Particular target may require a distinctive sampling rate which can be freely selected upon their needs more details about SV in chapter three [10].

1.9 Manufacturing Messaging Specification (MMS)

The object model and service that defined in IEC61850-7-x are mapped over the application layer of MMS (layer seven) which is a part from ACSI that not times critical messages. The MMS is an international standardized messaging system (ISO 9506) that been used to exchange real-time data and services among network devices. The mean feature of the MMS it is independent of the application function being performed and the device, software manufacturer. Further, based on the high generic of the messages' services that provided by MMS, it is appropriate for different types of function, devices and industries, for instance based on ACSI implementation requirements the information modeling and services that provided by MMS exactly meet the needs (Stalling 2007). When looking at the benefit of implementing MMS messaging services, three major effects that can be evaluated which are contributed to reduce a cost. First, interoperability that allows network applications, IEDs to share their real-time information without the needs to create the information environments. Second, independence that provides interoperability

independently from the developer of the application, network connectivity and functions based on provide common communication services. Last, data access that facilitates application to provide useful functions by obtaining the information that required throw the network application (IEC 61850-7-2 2003). From implementation perspective of the MMS messages services, the object of ACSI server class is mapped based on IEC61850-8-1 by SCSM one-to-one over the virtual machine device (VMD) object. The VMD is the key feature of the MMS messages services which specify how the server behaves upon the client application point view. Moreover, it represents that part from an application task that enables the monitoring and control services by means of a set of resources and functionality associated with one or more devices. Generally, the VMD defined the objects than what the server consists of. Further, the services that enable a client to access and manipulate these objects by means of assigning one or multiple communication addresses that creates the service access point SAPs where the MMS services can be exchanged. Figure 19 illustrates the VMD concept.

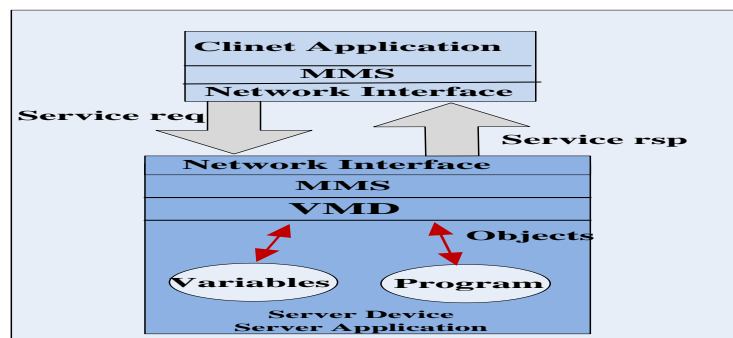


Figure 10. MMS concept.

1.10 Substation Configuration description Language (SCL)

In IEC61850-6-1 that specifies the substation configuration language SCL based on the XML. System integrator may need to configure the system devices both before utilizations and also in the case of adding new devices to the system. The SCL exhibit a hierarchy files configuration in which that allows unambiguous description within multiple levels of the system based on the standardized XML files. It is a uniform description for the system configuration and the relation between the system and the allocated functions. Therefore, the purpose for using the SCL file format, it makes possible to describe in a standardized manner, communication and function related device capability, device parameters, communication system configuration and allocation of the system devices. These descriptions' files for the

entire system and its devices can be shared between devices, engineering tools and various system engineering tools to build up the automation system (Brand 2004).

In order to achieve the system integration in the standardized way that compliance with IEC 61850 data model, SCL defined an object model that describes the IEDs connection, allocation and how this object model can be published within various SCL files. These files can be exported imported with various system engineering tools. Six different SCL files data type are defined based on the configuration step, content and their needs. They are constructed within identical format and method, however, they have various scope and extensions. From these files system configuration tool can obtain all the needed information such as substation network topology, IEDs description, communication system description and data type templates. Table 3 describes the six file types and its extinctions (IEC 61850-1 2003).

Table 2. SCL file types and extinctions.

Ex.	Name	Description
.icd	IED Capability Description	Describe the capabilities of an IED, IED allocation, IED communication description and system description
.ssd	System Specification Description	Describe the SAS and the required LNs
.scd	Substation Configuration Description	Specify all IEDs, communication configuration and substation description
.cid	Configured IED Description	Facilitate the communication between an IED configuration tool, IED allocation within the substation
.IID	Instantiated IED Description	Describe the configuration for an IED
.SED	System Exchange Description	Describe the exchanging configuration between substations

1.11 IEC 61850 Technical Challenges Implementation Issues

From the first step, as IEC 61850 standard had been published the major scope of the standard was to clarify the communication issues within the SAS that compose

various manufacturer devices. This task has been fulfilled by defining several communication protocols that facilitate the sharing information among those various manufacturer devices. However, several implementation issues have been left to the researchers designer engineers to deal with such as, reliability of the SAS and reliability of the IEC 61850 distributed functions, improve system reliability based on redundancy approach and type of redundancy, commissioning of various manufacturer devices' tasks, clarification of actual complete system design and its operation and maintenance issues. Moreover, the open nature of the IEC 61850 standard that gives a wide range of freedom for the manufacturers to operate with that may increase the complexity of the IEC 61850 implementation task. In addition, the emerging concept of the SG puts a high pressure on the existing Utilities infrastructure operation. Since, several of the available automation and protection functions their fundamental operations schemes are collapse as mentioned earlier. Therefore, more discussions and testing works need to be processed in order to meet the end users requirements and successful implementation of the IEC 61850 standard. In this section, the technical challenges and the IEC 61850 standard implementation issues have been highlighted where within the rest of this thesis these issues are analyzed and appropriate solutions were developed. These developing solutions have been carried out to facilitate and bring of a complete SAS that has the ability of utilizing the vision of IEC 61850 standard.

1.11.1 System Configuration Challenging

Practical system interoperability testing project requires various methods and tools. However the IEC 61850 standard defines the IEDs model, communication services and their various common files to describe those models and does not try to specify the IED or the system configuration tool. Those various manufacturers IEDs and system configuration tools present big challenges for protection and integrator engineers while configuring individual IED or the whole system. The challenges have been raised based on the IED protection and configuration settings parameters are different for each manufacturer in which that require proprietary software for each IED. Moreover, engineers need to be trained to implement different proprietary configuration tools for the same purpose. References [11], [12], [13] and [14] have been identified that the major challenges facing the implementation of the IEC 61850 standard within a multi-vendor IEDs and the most costly, efforts and consuming-time process was the configuration task based on the available SAS configuration tools. Most of the above mentioned works are pilot projects that were used the commercial IEDs in a designed laboratory platform and the commercial software configuration tools. Moreover, the commission task needs a full sport form the manufacturers products lines. Therefore, in order to reduce cost, efforts

and consuming-time a novel approach for a new SAS configuration tool has to be developed and to be independent of any commercial IEDs brand. In addition, it has the ability to import SCL files from various manufacturers, IEDs, systems and data bases, makes and raises the IEDs configuration to the system level. As a result, within this thesis the vendor natural system configuration tool approach has been invented and proposed to enhance the SAS configuration task. In addition, it may go beyond the SAS that supports, for example, large Utilities that compose of various manufacturer units, IEDs or the smart grid concept.

2 LIGHT-WEIGHT IEC 61850

The global incoming booming of the DERs that need to be integrate to the energy grid, and their impacts on the distribution power systems turn into raising challenges. These chal-lenges’ stimuli the Utilities and DER manufacturers to announce the concept of growing need to define and standardize the communication outside the individual SAS that may in-clude various DER IEDs. As a result, the standard IEC 61850-7-420 was published in 2009 as an extension of the IEC 61850 standard and to address these issues. The IEC 61850-7-420 specify various types of LNs and information modeling that applicable for the various DERs e.g. fuel cell systems, photo-voltaic (PV), combined heat and power (CHP), etc. That infor-mation mod-eling, LNs facilitates the communication and the integration of the DERs into the Utilities protection and automation systems. Utilities and DER manufacturers expected to achieve benefits from utilizing the IEC 61850-7-420 in terms of reducing the installation, maintenance costs. Further, offering standardization of all DERs data models that will im-prove the interoperability among distributed automation system (DAS) and DERs and in-crease the reliability of the energy grid as illus-trated in Figure 11.

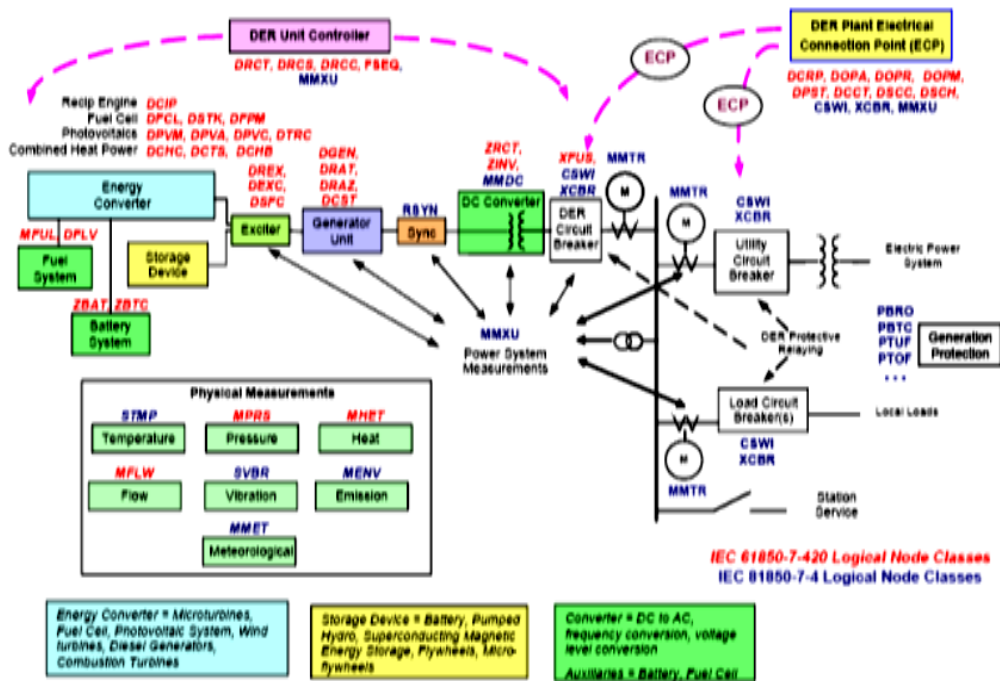


Figure 11. DERs logical devices and logical nodes concepts.

From Figure 11 the consideration has been raised that this IEC 91850-7-420 stand-ard address the information modelling for various DERs, whereas, other IEC 61850

implementation aspects such as the services modeling, assigned system configuration language (SCL) and the mapping schemes over the defined protocols had been covered within the previous IEC 61850 10 parts first version. These DERs information modeling defined by IEC 61850-7-420 standard involve not only for the local communication among the local DERs and the local management service systems, however, they may support the sharing information with the main grids operators or aggregators who manage the electrical grid operation. The defined DERs LNs based on IEC 61850-7-420 grouped into four groups upon their operation characteristic (node classes and common data classes (CDC)). These DERs LNs groups are logical nodes for DER management systems, logical nodes for DER generation systems, logical nodes for specific types of DER and logical nodes for auxiliary systems. These defined DERs LNs represents all the DERs operation aspects parameters such as for instance, connecting status, availability status, economic dispatch parameters, start/stop time, operating mode etc.

2.1 Light- Weight IEC 61850 reducing complexity

Although IEC 61850 standards offering many benefits, research and development tasks can be more costly and time consuming. However, the availability of the open source library up-on various solutions may reduce the complexity of the IEC 61850 standards [8], [9]. Those solutions are automatically generating the low-level machine code required for the IEC 61850 functions implementing within different operating systems. Embedded open-source operating systems running on micro-controller become more increasingly popular in both industrial and academe sides, since, in general offering a high stability, networking ability and extremely affordable price.

According to the implementation of the open-source libiec61850 library, communication among the embedded distributed systems (DS) done on top of the MMS protocol in standard C language. Moreover, it supports the inter-substation GOOSE and SV communication protocols and all useful services of the IEC 61850 standards by generating C code from the available SCL file. It provides a server and client library that includes varies files with a pre-defined functions such as “IedServer iedserver = IedServer_create()” to create a server, “IedConnection_connect()” to connect a client. Those functions take arguments based on the pre-assigned variables and parameters.

In this report embedded Linux micro-controllers and FPGA are explained and implemented as IED devices running the IEC 61850 using the libiec61850 library. This solution provides the data in accordance to the IEC 61850 standards data model. These data model accessed and manipulated through different IEC 61850 clients such as another embedded Linux micro-controller, COM600, IEDScout etc. over various communication system networks and technologies. One of the most dominated distributed communication network globally is the Internet that may link most of the distributed systems in the future. It has no centralized decision maker based on its implementation or policies for access and usage, were, data can be exchanged among distributed systems almost anywhere regardless to the communication medium.

2.2 Light-Weight IEC 61850 implementation process

This section from the report not intended to teach Linux neither C language. It is a brief illustration of the libiec61850, microcontroller and FPGA features and how to use them to connect between different designed IEDs over the communication system based on the IEC 61850 communication protocols. It is intended to help users to implement the light-weight IED to do the project work. In order to study Linux and C, please refer to free online books, announce in references.

2.2.1 *Setting up the device*

The first major step into the world of embedded microcontroller is to hook it up and get to a command prompt so you can start executing command and working with files. From this step, you'll be able to customize the system to suit your preferences and start creating your own projects.

Firstly, start connecting the embedded microcontroller (BeagleBone) via SSH over USB as in Figure 12 below,

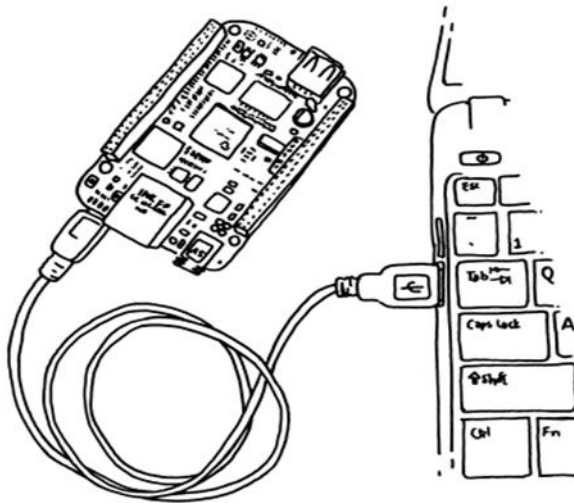


Figure 12. USB connection cable

In Windows PC this step needs to install PuTTY software, and in PuTTY window you need to select SSH radio button and enter the host IP address 192.168.7.2, since your PC generate virtual adapter with the IP address 192.168.7.1 when you connect the USB cable. Now you need to click open as illustrated in Figure ,

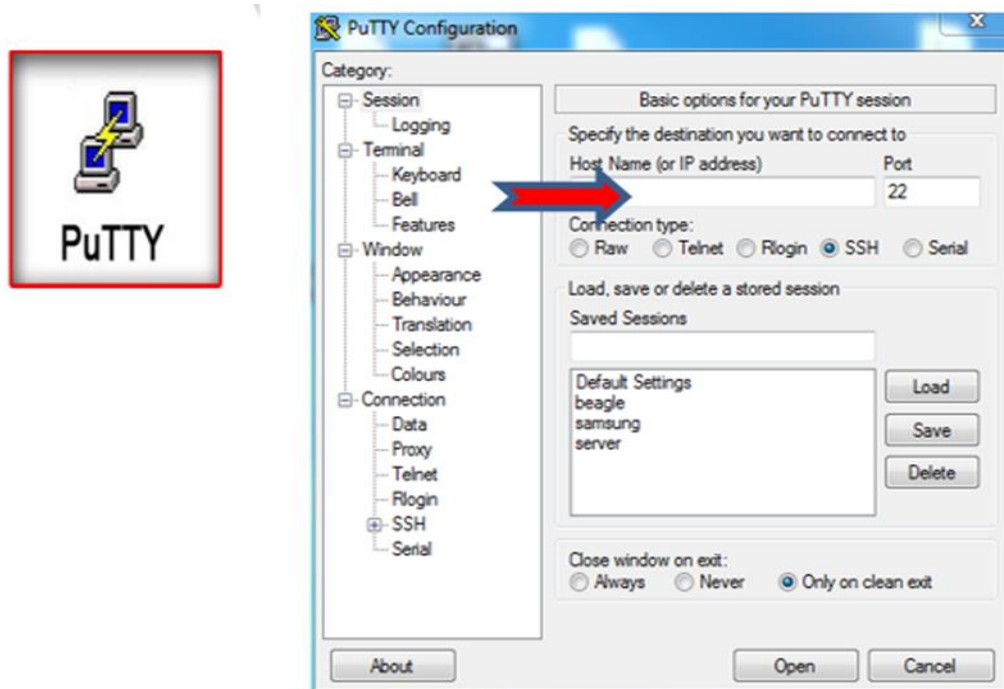


Figure 13. PuTTY window

When it shows you the prompt “login as:” type “root” and press enter as illustrated in Figure 14, by default there is no password, however if you assign password you need also to type it as

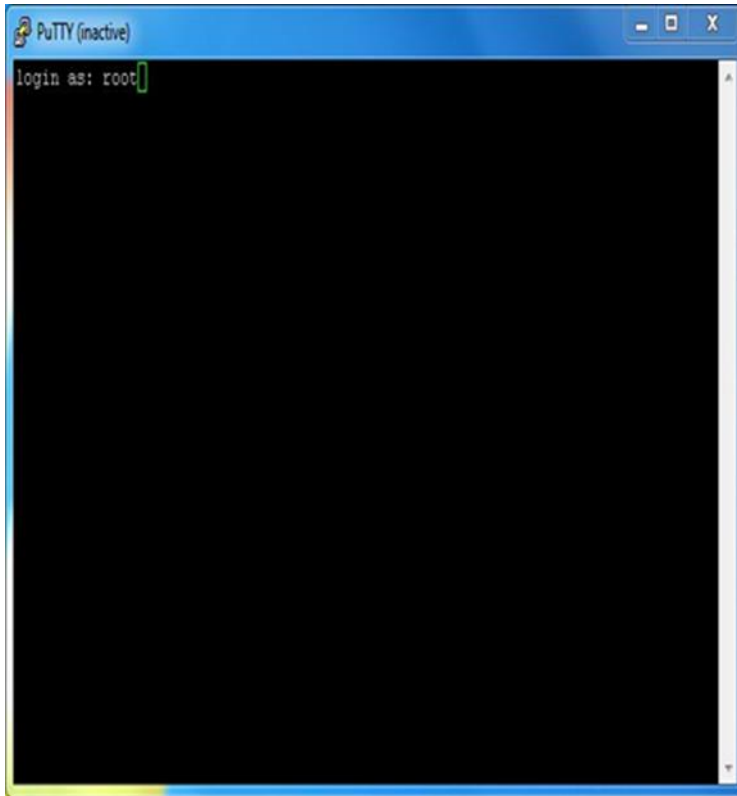


Figure 14. Prompt window

Secondly connecting the beaglebone via SSH over Ethernet, lunch PuTTY then enter “beagle-bone.local” as the host address, making sure that “SSH” is selected and press connect. When it shows you the prompt “login as:” type root and press enter. If the host name beaglebone.local does not work, you may need also to use the IP address of the board instead. Find it by logging into your router and looking for “beaglebone” on the DHCP clients list. The first time you connect, your SSH client may warn you that the host is unknown. It’s OK to accept the host key and dismiss this message.

Thirdly, connecting to the BeagleBone via Serial, with the FTDI cable as illustrated in Figure 15,



Figure 15. FTDI cable

Connect the USB side of a 3.3V FTDI cable to your computer. Connect the other side of the FTDI cable to the six male pins marked J1 on the BeagleBone Black. The FTDI cable's black wire should be closer to the "J1" label. Launch Putty and select the connection type choose Serial and type in the name of the serial port for your connection. You might need to look in Device Manager (Windows Key-R, then type devmgmt.msc, click OK and look under Ports) to find it then click OK. For the speed type 115200. The rest of the defaults should be fine then click OK. The prompt window will pop up and you will see a flashing cursor in the terminal window. Press enter to be taken to the login screen and Log in with the user name root.

2.3 libiec61850 library

There are several open source IEC 61850 libraries focused on system integration and application development for energy management and control systems, which use the IEC 61850 communication protocol. These libraries implemented in different languages for example an OpenIEC61850 library is Java implementation; libiec61850 library is C implementation etc.

In this report, we consider libiec61850 library based C. This library provides a server and client solution for the IEC 61850 communication protocols such as MMS, GOOSE and IEC Sampled Values. The library is designed according to edition 2 of the IEC 61850 standard series, but should be compatible to edition 1 in most cases.

2.3.1 *Overview of libiec61850 library*

This library provides an implementation of IEC61850 on top of the MMS (Manufacturing Message Specification) protocol in standard C. It also provides support for intra-substation communication via GOOSE. The goal of this project is to provide an implementation that is very portable and can run on embedded systems and micro-controllers. Also provided is a set of simple examples that can be used as a starting point for own applications.

This implementation runs on SoC, embedded systems, embedded Linux systems as well as on desktop computers running Linux, Windows or MacOS.

The following features are provided:

- Full ISO protocol stack on top of TCP/IP
- static implementation of the IED model by generation of C code from SCL file
- dynamic creation of the IED model by API calls or by configuration file
- Association service (with password authentication)
- Read and write services for simple and complex MMS variables
- Model browsing services (GetServerDirectory, GetDeviceDirectory ...)
- Data set services, including dynamic creation and deletion of data sets
- IEC 61850 client and server API
- Simple MMS client API (e.g. to be used with TASE.2)
- Buffered and unbuffered reporting service
- Log service
- Support for Setting Group Control services
- all IEC 61850 control models are supported by server and client
- GOOSE publisher and subscriber code that can also be used standalone
- IEC 61850-9-2 (Sampled Values) publisher and subscriber code that can be used standalone

- Hardware abstraction layer and implementations for POSIX(Linux), WIN32, BSD style systems (Mac OS X, FreeBSD)
- Conversion tool to translate SCL files to static IED models
- Conversion tool to translate SCL files to server configuration files
- All file services (directory browsing, getFile, setFile, deleteFile...
- Also suitable for use in resource constraint embedded systems
- C# client API for applications running on .NET and Mono
- ready for UCA certification of edition 1 and edition 2 devices

2.3.2 *libiec61850 Installation*

In In Linux system, create a directory use the command `mkdir` to create a new directory and name it. Change to the newly created directory use the command `cd` “name of the folder”. If your Linux system has a connection to the Internet, you can easily install or upgrade software and code libraries. The software comes in packages. So now, download the `libiec61850` library in Linux environment and save it in to the created directory. In addition, the `libiec61850` library can be downloaded in Windows system and move it to the created directory using WinSCP software.

Unzip the `libiec61850` downloaded file using the command `tar -xvzf` “name of the file”.tar.gz . In order to build the library `libiec61850` supports two different build systems. The traditional `make` based system that works good with a GNU based toolchain, and the new `cmake` based system. With the `cmake` based build system it is also possible to perform builds with Visual Studio or other toolchains especially for the Windows platform. Up to now the `make` based system is required to cross-compile for embedded Linux systems. Within the library, there are some example ready to start with, for example, GOOSE publisher and GOOSE subscriber as illustrated in Figure see appendix 1.

In addition, developing and building your own examples based on defined application are possible, however that need deep knowledge in Linux and C.

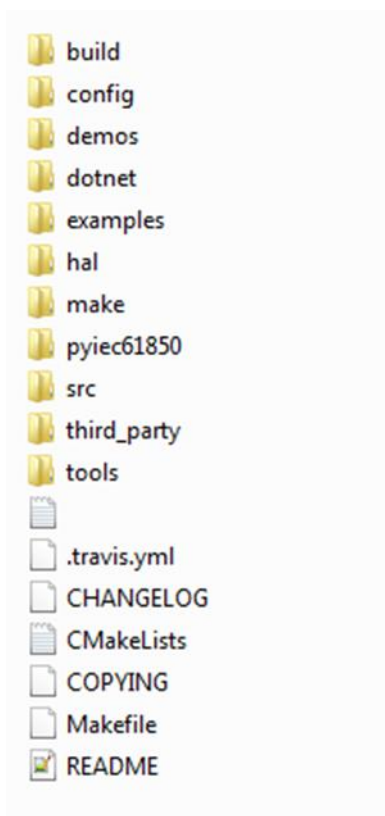


Figure 16. libiec61850 folders.

The same procedure that explained above can also be used to implement the libiec61850 within the SoC FPGA.

3. CONCLUSION

The aim of the report was to introduce, implement and test the libiec61850 library as a proof-of-concept of the Light-Weigh IEC 61850. The Light-Weigh IEC 61850 is implemented using BeagleBone and SoC FPGA that can publish and subscribe GOOSE messages. In addition the developed Light-Weight IEDs can publish and subscribe GOOSE messages to/from Opal rt real time simulator as CHIL. Light-weight IEC 61850 within inexpensive embedded microcontroller is a promising solution that reduces the time, complexity and overhead information associated with IEC 61850 implementation. Moreover, it allows the modeling of different IEC 61850-7-420 DER unit LNs for developing, testing and research purposes. Units of this kind do not exist yet within the commercial IEDs.

The main objective of adding the light-weight IEC 61850 microcontroller IEDs was to verify the implementation of the light-weight IEC 61850 GOOSE based CHIL monitoring and protection application.

The proposed Light-weight IEC 61850 solution can be extended to a promising new type of supervisory situation awareness system by integrating other different data objects from other LNs of DERs to be published in the GOOSE message data sets. These are, for instance, DCCT: economic dispatch parameters for DER operation, and DSCC: control of energy and ancillary services schedules. These DERs data objects can be used to achieve optimal grid operation and protection parameters. Furthermore, the monitored parameters can be used to classify different grid events in real-time.

More testing is still needed for the developed IEC 61850 protocols routing GOOSE and routing SMV and also e.g. wireless communication. Since wireless communication, technologies are being developed rapidly. However, wireless communication channels exhibit uncertainties and attenuation based on their implementations. This in turn brings more questions about the receiving messages latencies within these attenuated and uncertain communication channels. In addition, the effects of techniques to compensate the lost, out of order or delayed messages needs to be further studied. In practice, the use of parallel alternative communication channels may become necessary.

REFERENCES

- [1] D. Proudfoot, "UCA and 61850 For Dummies," Siemens Power Transmission and Distribution, 2002.
- [2] M. Adamiak, W. Premerlani, "The Role of Utility Communication in a Deregulated Environment," Co, System Sciences, Hawaii, 1999.
- [3] C. R. Ozansoy, A. Zayegh and A. Kalam, "Object Modeling of DataSet in the International Standard IEC61850," IEEE Power Delivery, Vol.24,No.3, 2009.
- [4] Overview of IEC61850 and benefits. In IEEE Power Engineering Society General Meeting
- [5] Janssen, M., & Apostolov, A. (2008). IEC61850 Impact on Substation Design. In Proceeding IEEE/PES Conference and Exposition on Transmission and Distribution. 1-7.
- [6] Holbach, J., Rodriguez, J., Wester, C., Baigent, D., Frisk, M., Kunsman, S., & Hossenlopp, L., (2007). STATUS ON THE FIRST IEC61850 BASED PROTECTION AND CONTROL MULTI-VENDOR PROJECT IN THE UNITED STATES. In Proceeding 60th Annual Conference for Protective Relay Engineering's. 283-306.
- [7] Zhang, C., Tan, J., Kirby, B., Thompson, S., & Bo, Z. (2009). Modeling and Implementation Device Based on IEC61850. In Proceeding IEEE Conference on Electrical engineering
- [8] Brand, K., P., Brunner, C., & Wimmer, W., (2004). DESIGN OF IEC61850 BASED SUBSTATION AUTOMATION SYSTEM ACCORDING TO CUSTOMER REQUIREMENTS. In CIGRE Session. Paris, 2004.
- [9] Zhang, L., X., & Kumal, N., (2008). Testing Protective Relays in IEC61850 Framework. In Proceeding Power Engineering Conference (AUPEC'08). 1-6.
- [10] Hou, D., & Dolezike, D., (2008). IEC61850-What it Can and Cannot offer to Traditional Protection Schemes. [Web document]. Schweitzer Engineering Laboratories. Available at:http://www.ucaiug.org/Meeting/CIGRE_2014/USB%20.
- [11] Ali, I., Thomas, M., S., & Gupta, S., (2012). Methodology & Tools for Performance Evaluation of IEC 61850 Goose based Protection Schemes. In IEEE 5th Power India Conference. 1-6.
- [12] Holbach, J., Rodriguez, J., Wester, C., Baigent, D., Frisk, M., Kunsman, S., & Hossenlopp, L., (2007). STATUS ON THE FIRST IEC61850 BASED PROTECTION AND CONTROL MULTI-VENDOR PROJECT IN THE UNITED

STATES. In *Proceeding 60th Annual Conference for Protective Relay Engineering's*. 283-306

[13] Chen, S., J., Hsiang, Y., Hung, C., Sheng, T., & Fang, R., (2012). Using Multi-Vendor IEDs for IEC6180 Interoperability and HMI-SCADA Applications. In International Symposium on Computer Consumer and Control (IS3C). 745-748.

[14] Mekkanen, M., Virrankoski, R., Elmusrati, M., & Antila, E., (2013). Performance Evaluation of IEC 61850 GOOSE based Interoperability. In WIT Transaction on Engineering Sciences. 209-217.

[15] libiec61850 <https://github.com/mz-automation/libiec61850>.

APPENDICES

Appendix 1.

```
/*  
  
* server_example_goose.c  
  
*  
* This example demonstrates how to use GOOSE publishing, Reporting and the  
* control model.  
*  
*/  
  
#include "iec61850_server.h"  
  
#include "hal_thread.h" /* for Thread_sleep() */  
  
#include <signal.h>  
  
#include <stdlib.h>  
  
#include <stdio.h>  
  
  
#include "static_model.h"  
  
/* import IEC 61850 device model created from SCL-File */  
  
extern IedModel iedModel;  
  
  
static int running = 0;  
  
static IedServer iedServer = NULL;
```

```
void sigint_handler(int signalId)
{
    running = 0;
}

void
controlHandlerForBinaryOutput(void* parameter, MmsValue* value)
{
    uint64_t timestamp = Hal_getTimeInMs();

    if (parameter == IEDMODEL_GenericIO_GGIO1_SPCSO1) {
        IedServer_updateUTCTimeAttributeValue(iedServer, IEDMODEL_GenericIO_GGIO1_SPCSO1_t, timestamp);

        IedServer_updateAttributeValue(iedServer, IEDMODEL_GenericIO_GGIO1_SPCSO1_stVal, value);
    }

    if (parameter == IEDMODEL_GenericIO_GGIO1_SPCSO2) {
        IedServer_updateUTCTimeAttributeValue(iedServer, IEDMODEL_GenericIO_GGIO1_SPCSO2_t, timestamp);

        IedServer_updateAttributeValue(iedServer, IEDMODEL_GenericIO_GGIO1_SPCSO2_stVal, value);
    }
}
```

```
if (parameter == IEDMODEL_GenericIO_GGIO1_SPCSO3) {

    IedServer_updateUTCTimeAttributeValue(iedServer, IEDMODEL_Generi-
cIO_GGIO1_SPCSO3_t, timestamp);

    IedServer_updateAttributeValue(iedServer,          IEDMODEL_Generi-
cIO_GGIO1_SPCSO3_stVal, value);

}

if (parameter == IEDMODEL_GenericIO_GGIO1_SPCSO4) {

    IedServer_updateUTCTimeAttributeValue(iedServer, IEDMODEL_Generi-
cIO_GGIO1_SPCSO4_t, timestamp);

    IedServer_updateAttributeValue(iedServer,          IEDMODEL_Generi-
cIO_GGIO1_SPCSO4_stVal, value);

}

}

int main(int argc, char** argv) {

    iedServer = IedServer_create(&iedModel);

    if (argc > 1) {

        char* ethernetIfcID = argv[1];

        printf("Using GOOSE interface: %s\n", ethernetIfcID);

        IedServer_setGooseInterfaceId(iedServer, ethernetIfcID);

    }

}
```

```
/* MMS server will be instructed to start listening to client connections.
*/

IedServer_start(iedServer, 102);

IedServer_setControlHandler(iedServer, IEDMODEL_GenericIO_GGIO1_SPCSO1, (ControlHandler) controlHandlerForBinaryOutput,
IEDMODEL_GenericIO_GGIO1_SPCSO1);

IedServer_setControlHandler(iedServer, IEDMODEL_GenericIO_GGIO1_SPCSO2, (ControlHandler) controlHandlerForBinaryOutput,
IEDMODEL_GenericIO_GGIO1_SPCSO2);

IedServer_setControlHandler(iedServer, IEDMODEL_GenericIO_GGIO1_SPCSO3, (ControlHandler) controlHandlerForBinaryOutput,
IEDMODEL_GenericIO_GGIO1_SPCSO3);

IedServer_setControlHandler(iedServer, IEDMODEL_GenericIO_GGIO1_SPCSO4, (ControlHandler) controlHandlerForBinaryOutput,
IEDMODEL_GenericIO_GGIO1_SPCSO4);

if (!IedServer_isRunning(iedServer)) {
    printf("Starting server failed! Exit.\n");
    IedServer_destroy(iedServer);
    exit(-1);
}
```

```
/* Start GOOSE publishing */  
  
IedServer_enableGoosePublishing(iedServer);  
  
running = 1;  
  
signal(SIGINT, sigint_handler);  
  
float anIn1 = 0.f;  
  
while (running) {  
  
    IedServer_lockDataModel(iedServer);  
  
    IedServer_updateUTCTimeAttributeValue(iedServer, IEDMODEL_Generi-  
cIO_GGIO1_AnIn1_t, Hal_getTimeInMs());  
  
    IedServer_updateFloatAttributeValue(iedServer, IEDMODEL_Ge-  
nericIO_GGIO1_AnIn1_mag_f, anIn1);  
  
    IedServer_unlockDataModel(iedServer);  
  
    anIn1 += 0.1;  
  
    Thread_sleep(1000);  
}
```



```
    }

    /* stop MMS server - close TCP server socket and all client sockets */
    IedServer_stop(iedServer);

    /* Cleanup - free all resources */
    IedServer_destroy(iedServer);
} /* main() */

.....

/*
 * goose_subscriber_example.c
 *
 * This is an example for a standalone GOOSE subscriber
 *
 * Has to be started as root in Linux.
 */

#include "goose_receiver.h"

#include "goose_subscriber.h"

#include "hal_thread.h"

#include <stdlib.h>

#include <stdio.h>
```

```
#include <signal.h>
```

```
static int running = 1;
```

```
void sigint_handler(int signalId)
```

```
{
```

```
    running = 0;
```

```
}
```

```
void
```

```
gooseListener(GooseSubscriber subscriber, void* parameter)
```

```
{
```

```
    printf("GOOSE event:\n");
```

```
    printf("  stNum: %u sqNum: %u\n", GooseSubscriber_getStNum(subscriber),
```

```
          GooseSubscriber_getSqNum(subscriber));
```

```
    printf("  timeToLive: %u\n", GooseSubscriber_getTimeAllowedToLive(sub-  
subscriber));
```

```
    uint64_t timestamp = GooseSubscriber_getTimestamp(subscriber);
```

```
    printf("  timestamp: %u.%u\n", (uint32_t) (timestamp / 1000), (uint32_t)  
(timestamp % 1000));
```

```
    MmsValue* values = GooseSubscriber_getDataSetValues(subscriber);
```

```
char buffer[1024];

MmsValue_printToBuffer(values, buffer, 1024);

printf("%s\n", buffer);
}

int
main(int argc, char** argv)
{
    GooseReceiver receiver = GooseReceiver_create();

    if (argc > 1) {
        printf("Set interface id: %s\n", argv[1]);
        GooseReceiver_setInterfaceId(receiver, argv[1]);
    }
    else {
        printf("Using interface eth0\n");
        GooseReceiver_setInterfaceId(receiver, "eth0");
    }

    GooseSubscriber subscriber = GooseSubscriber_create("simpleIOGeneri-
cIO/LLN0$GO$gcbAnalogValues", NULL);
```

```
GooseSubscriber_setAppId(subscriber, 1000);

GooseSubscriber_setListener(subscriber, gooseListener, NULL);

GooseReceiver_addSubscriber(receiver, subscriber);

GooseReceiver_start(receiver);

signal(SIGINT, sigint_handler);

while (running) {
    Thread_sleep(100);
}

GooseReceiver_stop(receiver);

GooseReceiver_destroy(receiver);
}
```