IEC TR 61850-7-510

Edition 2.0 2021-12

TECHNICAL REPORT

colour inside

Communication networks and systems for power utility automation – Part 7-510: Basic communication structure – Hydroelectric power plants, steam and gas turbines – Modelling concepts and guidelines

IEC TR 61850-7-510:2021-12(en)

About the IEC
The International Electrotechnical Commission (IEC) is the leading global organization that prepares and publishes International Standards for all electrical, electronic and related technologies.

About IEC publications
The technical content of IEC publications is kept under constant review by the IEC. Please make sure that you have the latest edition, a corrigendum or an amendment might have been published.

IEC publications search - webstore.iec.ch/advsearchform
The advanced search enables to find IEC publications by a variety of criteria (reference number, text, technical committee, …). It also gives information on projects, replaced and withdrawn publications.

IEC Just Published - webstore.iec.ch/justpublished
Stay up to date on all new IEC publications. Just Published details all new publications released. Available online and once a month by email.

IEC Customer Service Centre - webstore.iec.ch/csc
If you wish to give us your feedback on this publication or need further assistance, please contact the Customer Service Centre: sales@iec.ch.

IEC online collection - oc.iec.ch
Discover our powerful search engine and read freely all the publications previews. With a subscription you will always have access to up to date content tailored to your needs.

Electropedia - www.electropedia.org
The world's leading online dictionary on electrotechnology, containing more than 22 000 terminological entries in English and French, with equivalent terms in 18 additional languages. Also known as the International Electrotechnical Vocabulary (IEV) online.

IEC TR 61850-7-510

Edition 2.0 2021-12

TECHNICAL REPORT

colour inside

Communication networks and systems for power utility automation – Part 7-510: Basic communication structure – Hydroelectric power plants, steam and gas turbines – Modelling concepts and guidelines

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

ICS 33.200

ISBN 978-2-8322-1062-2

Warning! Make sure that you obtained this publication from an authorized distributor.

CONTENTS

INTERNATIONAL ELECTROTECHNICAL COMMISSION

____

COMMUNICATION NETWORKS AND SYSTEMS FOR POWER UTILITY AUTOMATION –

Part 7-510: Basic communication structure – Hydroelectric power plants, steam and gas turbines – Modelling concepts and guidelines

FOREWORD

1) The International Electrotechnical Commission (IEC) is a worldwide organization for standardization comprising all national electrotechnical committees (IEC National Committees). The object of IEC is to promote international co-operation on all questions concerning standardization in the electrical and electronic fields. To this end and in addition to other activities, IEC publishes International Standards, Technical Specifications, Technical Reports, Publicly Available Specifications (PAS) and Guides (hereafter referred to as "IEC Publication(s)"). Their preparation is entrusted to technical committees; any IEC National Committee interested in the subject dealt with may participate in this preparatory work. International, governmental and non-governmental organizations liaising with the IEC also participate in this preparation. IEC collaborates closely with the International Organization for Standardization (ISO) in accordance with conditions determined by agreement between the two organizations.
2) The formal decisions or agreements of IEC on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested IEC National Committees.
3) IEC Publications have the form of recommendations for international use and are accepted by IEC National Committees in that sense. While all reasonable efforts are made to ensure that the technical content of IEC Publications is accurate, IEC cannot be held responsible for the way in which they are used or for any misinterpretation by any end user.
4) In order to promote international uniformity, IEC National Committees undertake to apply IEC Publications transparently to the maximum extent possible in their national and regional publications. Any divergence between any IEC Publication and the corresponding national or regional publication shall be clearly indicated in the latter.
5) IEC itself does not provide any attestation of conformity. Independent certification bodies provide conformity assessment services and, in some areas, access to IEC marks of conformity. IEC is not responsible for any services carried out by independent certification bodies.
6) All users should ensure that they have the latest edition of this publication.
7) No liability shall attach to IEC or its directors, employees, servants or agents including individual experts and members of its technical committees and IEC National Committees for any personal injury, property damage or other damage of any nature whatsoever, whether direct or indirect, or for costs (including legal fees) and expenses arising out of the publication, use of, or reliance upon, this IEC Publication or any other IEC Publications.
8) Attention is drawn to the Normative references cited in this publication. Use of the referenced publications is indispensable for the correct application of this publication.
9) Attention is drawn to the possibility that some of the elements of this IEC Publication may be the subject of patent rights. IEC shall not be held responsible for identifying any or all such patent rights.

IEC TR 61850-7-510 has been prepared by IEC technical committee 57: Power systems management and associated information exchange. It is a Technical Report.

This second edition cancels and replaces the first edition published in 2012. This edition constitutes a technical revision.

This edition includes the following significant technical changes with respect to the previous edition:

a) Process modelling according to IEC 61850-6:2009, including IEC 61850-6:2009/AMD1:2018.
b) Examples of application of Reference Designation System together with the process modelling, in particular application of IEC/ISO 81346.
c) Description of modelling related to Steam- and Gas turbines.
d) Annexes with examples of application of SCL according to the examples in the Technical Report.

e) The dynamic exchange of values by using polling, GOOSE, Reporting or Sampled Values is no longer included in the Technical Report.

f) Updated examples of application of SCL:Process and IED modelling applying the Logical Nodes defined in IEC 61850-7-410:2012, including IEC 61850-7-410:2012/AMD1:2015.

The text of this Technical Report is based on the following documents:

DTR	Report on voting
57/2391/DTR	57/2432/RVDTR

Full information on the voting for its approval can be found in the report on voting indicated in the above table.

The language used for the development of this Technical Report is English.

This document was drafted in accordance with ISO/IEC Directives, Part 2, and developed in accordance with ISO/IEC Directives, Part 1 and ISO/IEC Directives, IEC Supplement, available at www.iec.ch/members_experts/refdocs. The main document types developed by IEC are described in greater detail at www.iec.ch/standardsdev/publications.

A list of all parts of the IEC 61850 series, under the general title: *Communication networks and systems for power utility automation*, can be found on the IEC website.

The committee has decided that the contents of this document will remain unchanged until the stability date indicated on the IEC website under webstore.iec.ch in the data related to the specific document. At this date, the document will be

- reconfirmed,
- withdrawn,
- replaced by a revised edition, or
- amended.

IMPORTANT – The "colour inside" logo on the cover page of this document indicates that it contains colours which are considered to be useful for the correct understanding of its contents. Users should therefore print this document using a colour printer.

INTRODUCTION

This Technical Report is connected with IEC 61850-7-410, as well as IEC 61850-7-4:2010, explaining how the control system and other functions in a hydropower, steam or gas turbine plant can use logical nodes and information exchange services within the complete IEC 61850 package to specify the information needed and generated by, and exchanged between functions.

The dynamic exchange of values by using polling, GOOSE, Reporting or Sampled Values is beyond the scope of this document.

This document applies the SCL Process element structure for modelling of the processes.

Examples of application of SCL Code according to the modelling examples in this document are presented in Annex B and Annex C.

COMMUNICATION NETWORKS AND SYSTEMS FOR POWER UTILITY AUTOMATION –

Part 7-510: Basic communication structure – Hydroelectric power plants, steam and gas turbines – Modelling concepts and guidelines

1 Scope

This part of IEC 61850, which is a technical report, is intended to provide explanations on how to use the Logical Nodes defined in IEC 61850-7-410 as well as other documents in the IEC 61850 series to model complex control functions in power plants, including variable speed pumped storage power plants.

IEC 61850-7-410 introduced the general modelling concepts of IEC 61850 for power plants. It is however not obvious from the standard how the modelling concepts can be implemented in actual power plants.

This document explains how the data model and the concepts defined in the IEC 61850 standard can be applied in Hydro; both directly at the process control level, but also for data structuring and data exchange at a higher level. Application of the data model for Thermal is limited to power evacuation (in principle the extraction of the generated electrical power) and the prime mover shaft and bearing system. The interfaces of the fuel and steam valves are modelled for the purpose of process control.

Communication services, and description of the use of mappings of the IEC 61850 data model to different communication protocols, are outside the scope of this document.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

IEC 61362:2012, *Guide to specification of hydraulic turbine governing systems*

IEC 61850-6:2009, *Communication networks and systems for power utility automation – Part 6: Configuration description language for communication in electrical substations related to IEDs*

IEC 61850-7-3:2010, *Communication networks and systems for power utility automation – Part 7-3: Basic communication structure – Common data classes*
IEC 61850-7-3:2010/AMD1:2020

IEC 61850-7-4:2010, *Communication networks and systems for power utility automation – Part 7-4: Basic communication structure – Compatible logical node classes and data object classes*
IEC 61850-7-4:2010/AMD1:2020

IEC 61850-7-410:2012, *Communication networks and systems for power utility automation – Part 7-410: Basic communication structure – Hydroelectric power plants – Communication for monitoring and control*
IEC 61850-7-410:2012/AMD1:2015

ISO 81346-10:—[1], *Industrial systems, installations and equipment and industrial products – Structuring principles and reference designations – Part 10: Power Supply systems*

3 Terms and definitions

No terms and definitions are listed in this document.

ISO and IEC maintain terminological databases for use in standardization at the following addresses:

- IEC Electropedia: available at http://www.electropedia.org/
- ISO Online browsing platform: available at http://www.iso.org/obp

4 Overview

4.1 General

This clause describes the target group of the document and introduces the modelled power plant domain.

4.2 Target group

This document targets engineers and system integrators working with control and modelling of Hydro Power and Thermal Power plant processes.

The document gives an overview of the process control in the different contexts and provides examples on how to structure and name the systems in a model, and how to use the DataObjects in control and supervision of the power plant processes. The document provides guidance on how to apply the IEC 61850 data model defined in IEC 61850-7-410.

4.3 Hydro power domain

4.3.1 General

In hydro power, the power is derived from the potential energy difference of water transferred from a higher to a lower level through a rotating turbine. The turbine transfers the power from the flowing water to a rotating shaft, and a generator transforms the mechanical power into electrical power. To handle the water level and the flow of water several types of gates are used.

4.3.2 Hydropower plant specific information

Different devices handle active and reactive power control. The turbine governor provides the active power control by regulating the water flow through the turbine and thus the pole angle between the rotating magnetic flux and the rotor. The excitation system provides the reactive power control by regulating the voltage of the generator. The magnetic flux corresponds to the shaft torque to keep the generator synchronised to the grid.

Figure 1 shows an example of an arrangement including a joint control function. The set-points will be issued from a dispatch centre and could be one of three optional values. Therefore, the type of set-point that will be used depends on the water control mode that is used for the plant.

1 Under preparation. Stage at the time of publication: ISO/DIS 81346-10:2021.

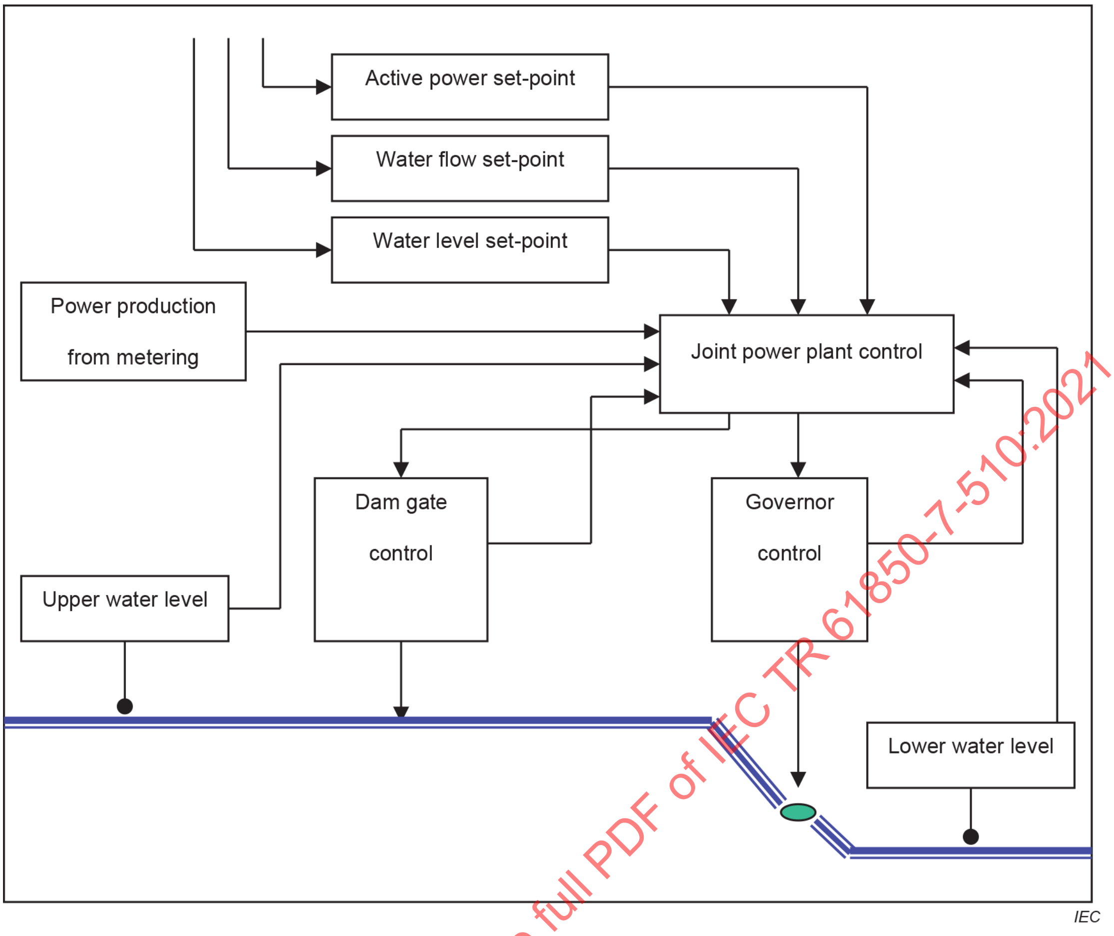


Figure 1 – Principles for the joint control function

In case of a reservoir without any power production, the water control function will get the water control set-points from a dispatch centre; in case of a power plant, it will normally be the joint control function that sets the values. The set-point will be either water level or water flow set-points.

The total water flow is the sum of flow through turbines and gates. The turbine control system can, due to this, be provided with different set-points for the control.

- Water flow set-point. The control system will base the regulation on the given water flow level and try to optimise the production.
- Active power set-point. The control system will try to meet the active power, the water flow will be reported back to the overall water control system.
- Active power control with speed droop. This is the mode when the unit is contributing to the network frequency control. The active power set-point is balanced over the speed droop setting to obtain the desired power/frequency amplification.
- Frequency set-point. In case of an islanded system or a power plant in peak load duty, the active power will be controlled to exactly meet the demand. This control mode is also used during start-up of the unit, up to the point when the generator is synchronised. Water flow will be reported.

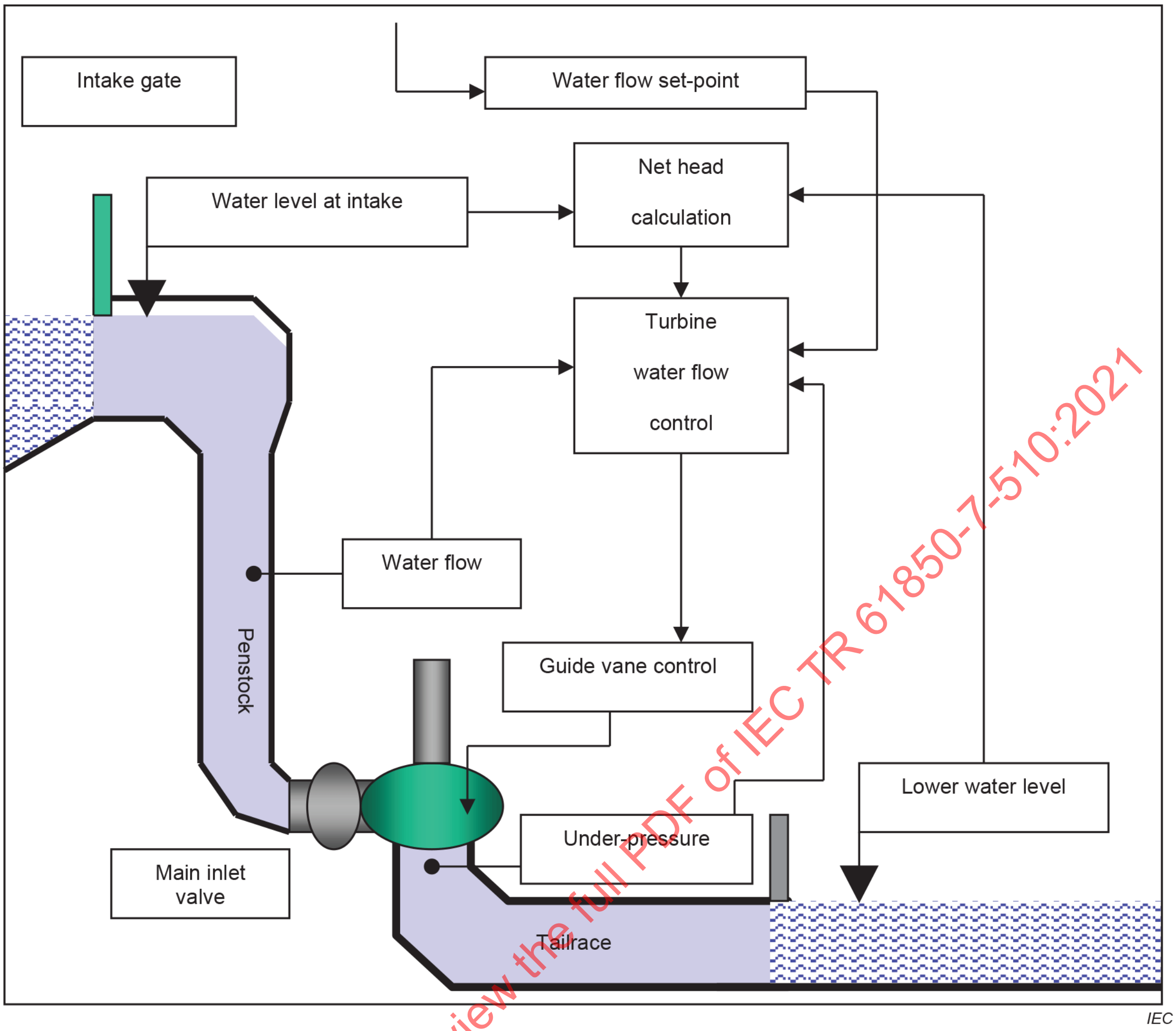


Figure 2 – Water flow control of a turbine

Figure 2 shows an example of water flow control for a turbine. Direct measurement of the water flow, as indicated in the figure, is less common. The flow is normally calculated, using the net head, the opening angle of the guide vanes and a correlation curve.

Main inlet valves to shut off the turbine chamber are used for pumped storage plants and power plants with high penstocks.

It is important to differentiate between the water levels of the dam and at the intake. Due to the intake design or if the turbine is running close to rated power, the water level at the intake might be considerably lower than the average for the dam.

The measurement of under-pressure below the turbine chamber is a safety measure, to ensure that the operation of the guide vanes does not cause any dangerous conditions in the tailrace part.

4.4 Thermal power domain

4.4.1 General

In thermal power the power is derived from the change in enthalpy of a medium flowing through a system. Commonly the combustion of a fuel is used to boil water into heated and pressurized steam which is directed through one or several turbines. Other ways of heating water are also in use in thermal power (thermal solar, geothermal, nuclear). The turbine transfers the power from the flowing steam to a rotating shaft, and a generator transforms the mechanical power into electrical power. To handle the steam pressure and temperatures several valves are used.

4.4.2 Steam turbine power plant specific information

As in hydropower plants, different devices handle active and reactive power control.

A steam turbine gets its power from the steam produced e.g. in a boiler or a steam generator. Usually a steam turbine consists of several stages: The steam is passed through a high-pressure turbine, then to the medium pressure turbine and finally through (several) low pressure turbine(s). They are on the same shaft together with the generator, see Figure 3 and Figure 4.

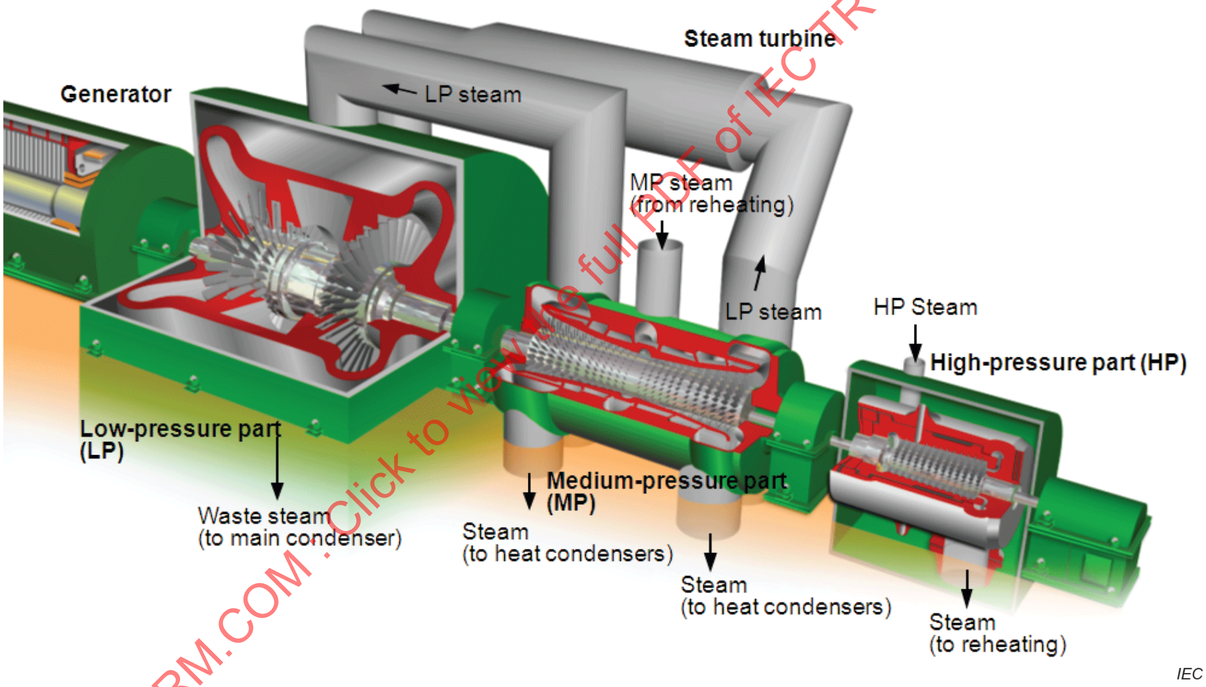


Figure 3 – Example of a large steam turbine

The turbine produces power from the steam provided by the boiler (or steam generator), according to its efficiency. The control of the turbine is done by the steam valve of the first stage of the steam turbine (high pressure turbine).

Figure 4 shows a possible configuration of a steam turbine integrated in a conventional water-steam-cycle. The burners/combustor may use hard coal, oil, gas, biomass or whatever is available for burning. There is usually an exhaust gas treatment after the boiler to fulfil environmental constraints. The greyed area shows the part of the process that are covered by the information model in IEC 61850-7-410.

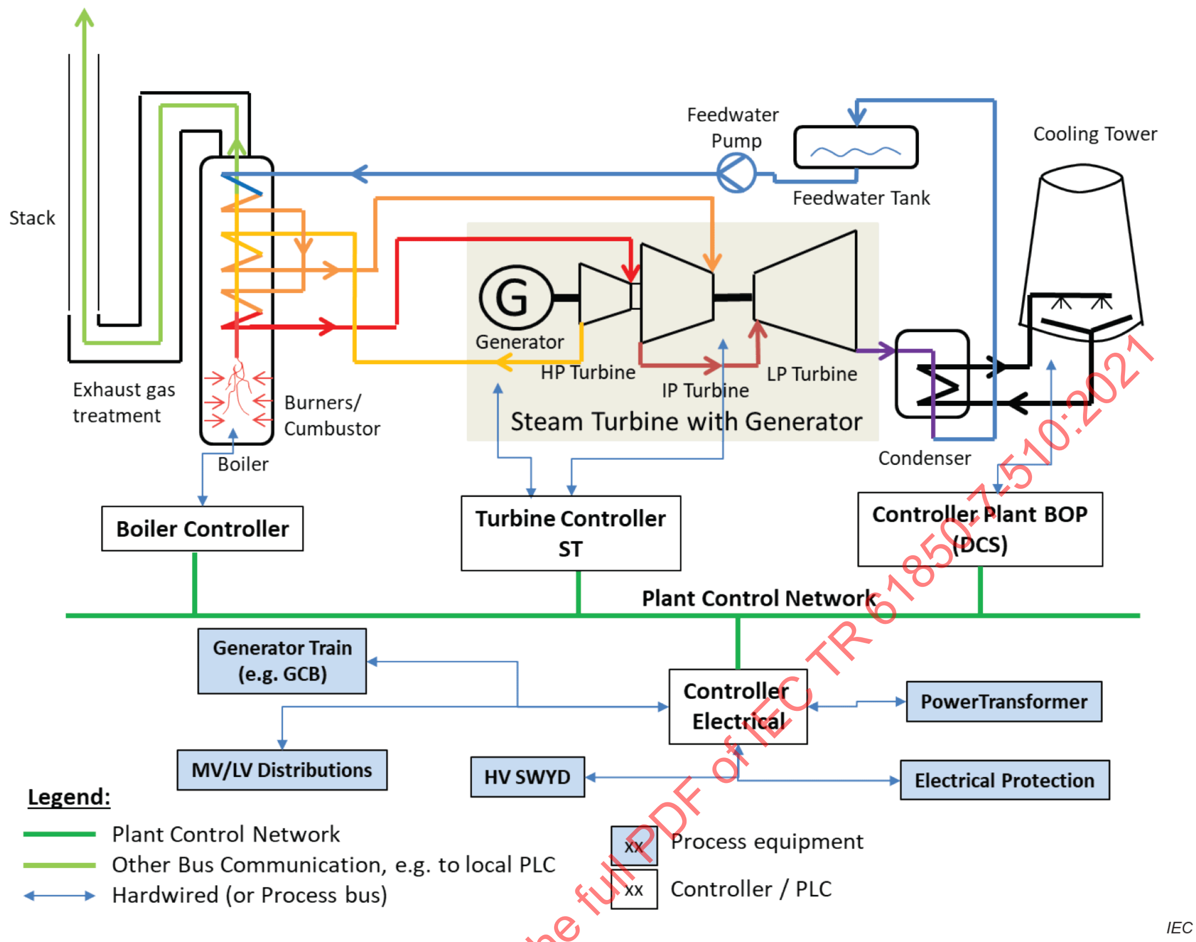


Figure 4 – Simplified example of a large steam turbine power plant with typical control system

4.4.3 Gas turbine specific information

In contrast to a steam turbine, where the working media is steam, the working media of a gas turbine is gas, i.e. air. First, the air is filtered at the air intake. Then it is compressed in the compressor. At the combustor afterwards the fuel (natural gas or oil) is injected and burned. The hot gas is expanded through the turbine which produces the mechanical torque. The mechanical torque is used on one hand side to drive the compressor and on the other hand to drive the generator, that finally produces the electrical power. Figure 5 shows an example of a gas turbine.

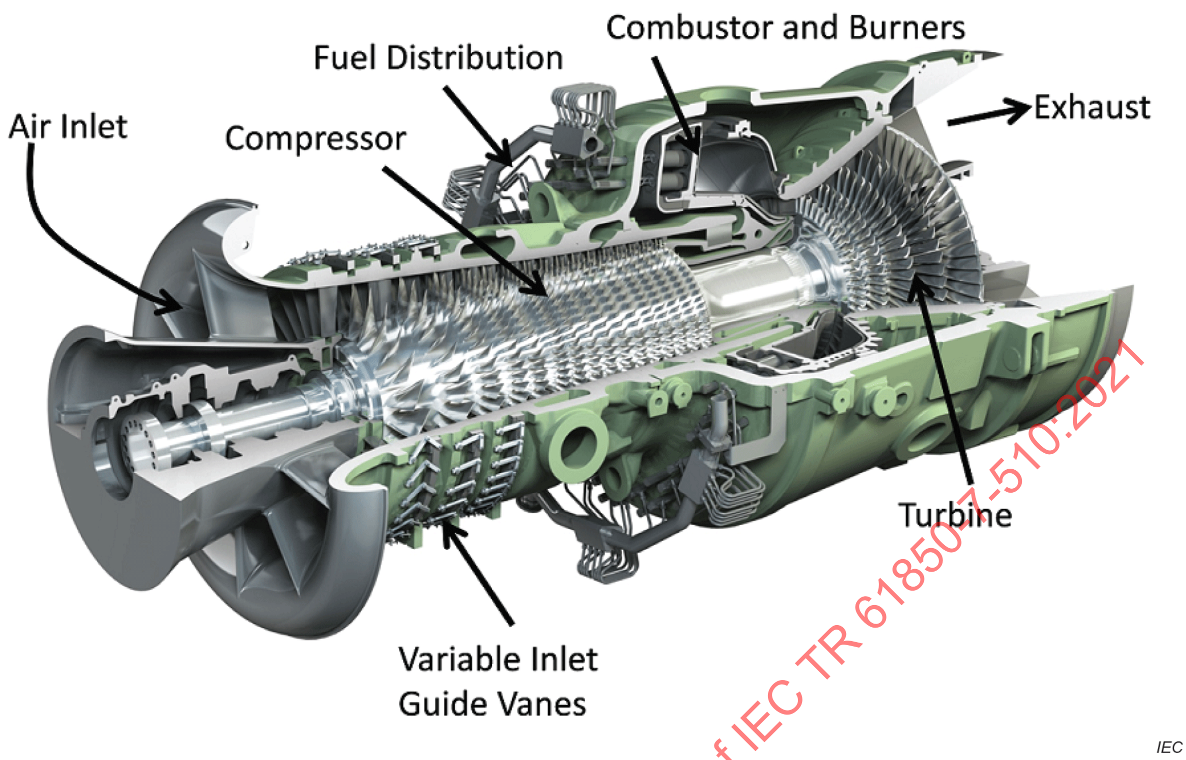


Figure 5 – Example of a gas turbine

4.4.4 Combined cycle power plants

Figure 6 shows a typical block of a combined cycle power plant consisting of a gas turbine (GT), a heat recovery steam generator (HRSG) that produces steam from the hot exhaust of the GT, and a steam turbine (ST). Usually, the GT produces about 2/3 and the ST about 1/3 of the total block power. There are as well configurations possible with two GTs and one ST or to have both turbines on the same shaft with one generator. The last is shown in Figure 7. The greyed areas in Figure 6 and Figure 7 show the parts of the process that are covered by the information model in IEC 61850-7-410.

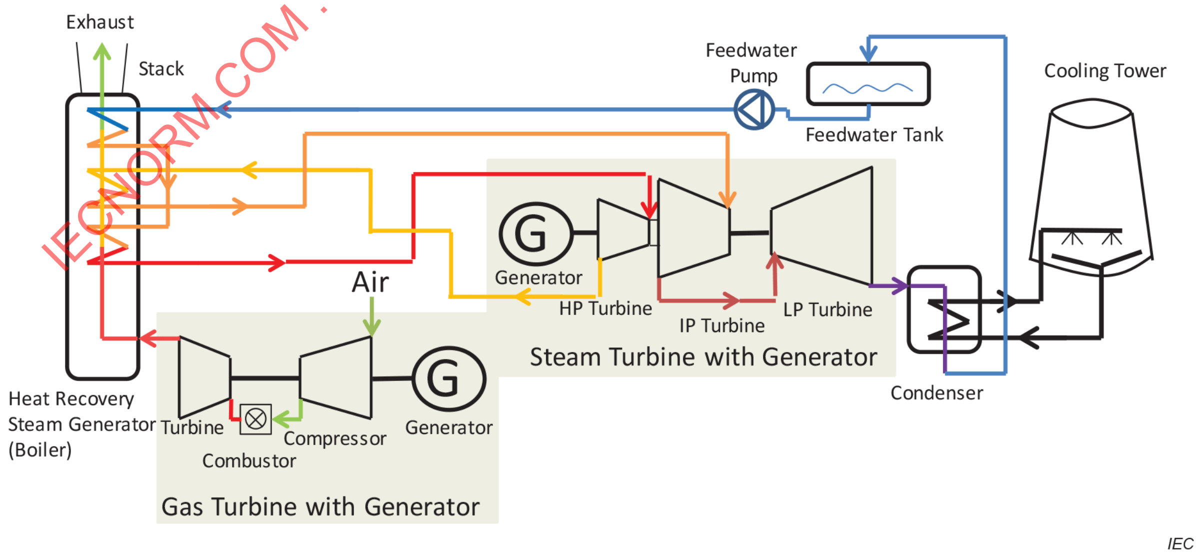


Figure 6 – Example of a combined cycle power plant with one GT and one ST in a multi-shaft configuration

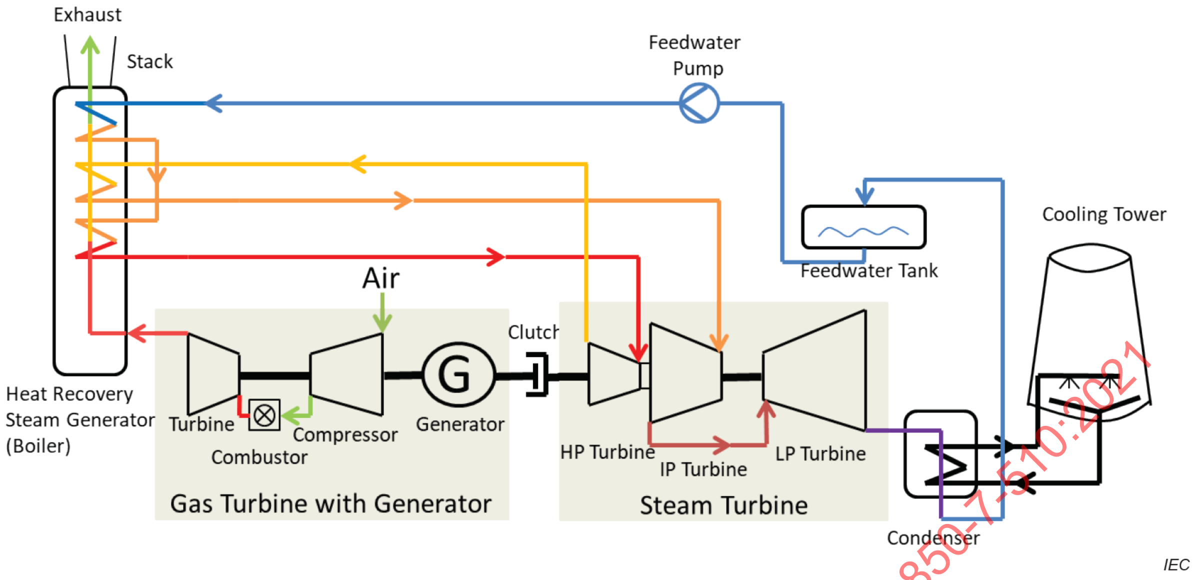


Figure 7 – Example of a combined cycle power plant with one GT and one ST in a single shaft configuration

In a combined cycle power plant, the steam turbine is usually operated with the control valve always fully open (sliding pressure mode), i.e. all steam produced by the heat recovery steam generator with the exhaust of the gas turbine is used by the steam turbine. The power of the combined cycle block is regulated by the GT only.

4.4.5 Coal-fired power plant specific information

Coal-fired power plants utilize coal as fuel to produce electricity. The basic production process is that coal is burned to boil water in order to generate steam. The steam is led to the turbine where heat energy is converted into mechanical energy. The turbine then drives the generator, where the mechanical energy is converted into electrical energy.

The processes for handling and processing of the raw coal, primary air and flue gas is outside the scope of this document.

In a typical coal fired power plant secondary reheating units reheat the exhaust steam of a high-pressure turbine. The reheated steam is then led to a medium pressure turbine, and after that the low-pressure turbine. The exhaust steam of the low-pressure turbine is cooled in the condenser. The condensate water is pumped back to the heating process again to complete a cycle.

The control system of a coal-fired power plant is used to control parameters of the various thermal processes, including temperature, pressure, flow rate and liquid level, to achieve safe and economic operation of the power plant. The control system generally consists of sensors or transmitter units, connection units, regulation units (governors) and execution system.

Figure 8 shows a typical block of a coal fired power plant. The parts of the coal fired power plant process covered by the Scope of IEC 61850-7-410:2012+AMD1:2015 is marked in the figure. The greyed area shows the part of the process that are covered by the information model in IEC 61850-7-410.

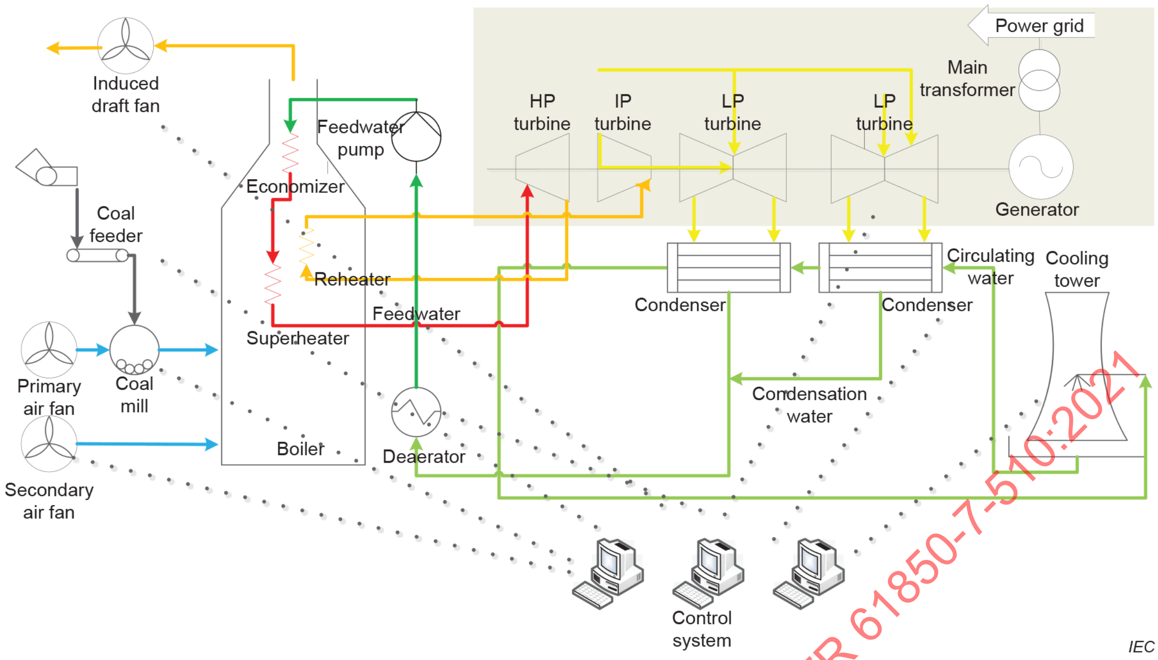


Figure 8 – Example of heat flow diagram of a coal-fired power plant

5 Process modelling

5.1 Reference designation system

5.1.1 General

To select an unambiguous reference designation, it is recommended to apply an established reference designation system (RDS). The selection of an RDS is a local issue. For the examples in this document the RDS defined in IEC/ISO 81346 is used.

5.1.2 Structuring principles and reference designation system

IEC/ISO 81346 is an international naming convention for systems and their elements, and as a practical technique to establish a common understanding among different technical domains.

The output of the system application is a system breakdown with accompanying reference designations. These act as labels/tags to the system and system elements enabling instant and unambiguous recognition across different kinds of models or documentation. It is, therefore, a key for bridging and binding technical domains, data sources and systems information sharing, throughout the system lifecycle.

The potential application of the RDS is limitless when considering technical domains or applications schemes. For this document, it will be presented as a potential object referencing system which can be used as a logical device structure as intended within IEC 61850.

The use of the RDS as logical device structures will enable coupling of signals directly to a structure also used by other domains than process control.

5.1.3 Object ownership principle

The principle in RDS gives the ownership of an object to the receiving system. Figure 9 illustrates the ownership principle.

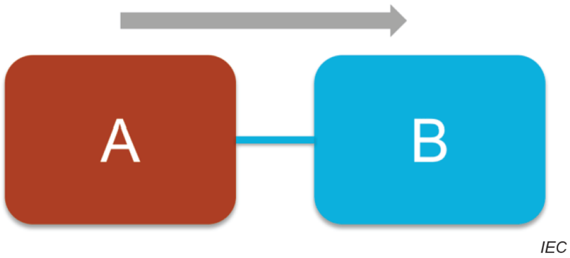


Figure 9 – IEC/ISO 81346 ownership principle

Example: The inlet system to a tunnel "B" (part of the transport system), from a reservoir "A" (part of the supply/storage system) is the separating agent between the two main systems. The inlet system belongs to the transport system "B", as "B" is the energy receiving system.

5.1.4 The concept of aspects

Aspects determine the view of the system that is used when forming RDS-structures and define the kind of system elements that are considered and how the compositional relations are defined. This means that the aspect will determine what kind of elements make up the RDS-structure.

For example, a functional aspect will consider elements and relations relating to functionality. Structures based on a functional aspect could, therefore, contain elements that represent functions or functional design solutions.

Four basic aspects (with one used in two different ways) are treated in the IEC/ISO 81346 series. They are represented by a prefix character used in the RDS syntax. Table 1 describes the IEC/ISO 81346 aspects.

Table 1 – IEC/ISO 81346 aspects

Prefix	Name	Description
=	Functional aspect	Based on the intended functionality of the system i.e. what functionality the system elements fulfil, disregarding how it is constructed.
-	Product aspect	Based on how the system is constructed, i.e. how system elements are arranged in order to construct the system and disregarding their functionality. It is often referred to as the "assembly" aspect.
+	Location aspect (Site of location)	The site of location aspect utilises an existing structure (functional aspect structure for the sake of this document) to point to the system that is physically hosting the object of interest.
++	Location aspect (Point of location)	The point of location is used to create references for objects physical locations and special arrangement.
%	Type aspect	The type aspect offers a view of a system that structures the system based on the commonality of its system elements.

Note that in this document the focus will mostly be on the functional aspect. It is the preferred aspect for the creation of an object reference and the natural choice for signal modelling purposes.

5.1.5 The RDS-structure and classification

An RDS-structure is a hierarchical tree-structure that describes the composition of systems.

Because the system concept is recursive, this means that systems are composed of the system elements, which can in turn themselves be systems (see Figure 10).

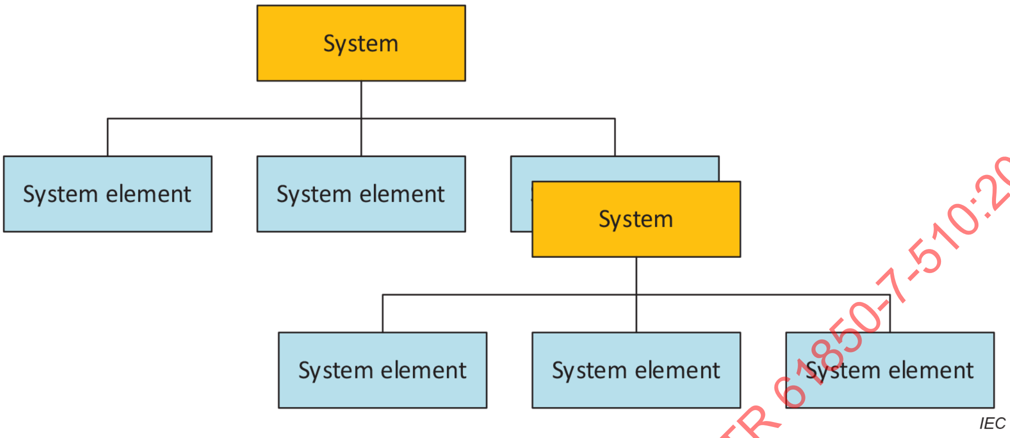


Figure 10 – A system breakdown structure showing the recursive phenomenon of system elements also being systems

The classification defines three levels of abstraction when defining the systems and system elements. They allow for the classification of systems at varying levels of abstraction and scope. The levels are Functional, Technical and Component system class levels. The Functional System level will often be referred to as "Main" systems to avoid confusion (see Figure 11).

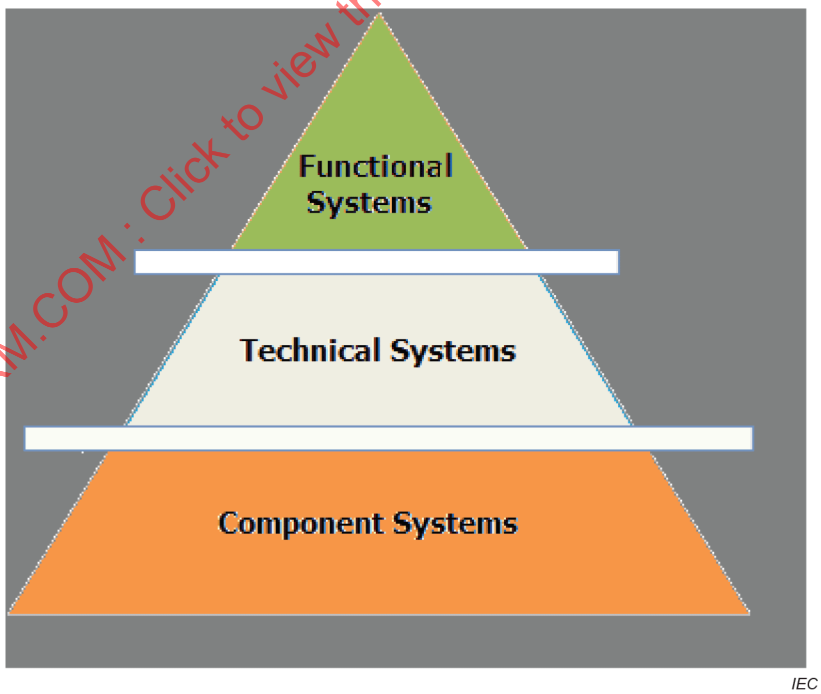


Figure 11 – Three levels of classes within RDS

Letter codes are classified by the number of letters:

- Class codes for functional ("main") systems are recognized by **ONE** letter code
- Class codes for technical systems are recognized by **TWO** letter codes
- Class codes for component systems are recognized by **THREE** letter codes

Each of these levels follows a dedicated classification table which (for the technical and component system) has a structure within the classification scheme itself.

5.1.6 Example: Unit 2 main inlet valve with a bypass system

Technical systems classes with J as the first letter are all transport systems for gas (JA), electricity (JF) and so on. In the case in Figure 12, the bypass system is a transport system for water JB.

As shown in Figure 12 using the resulting breakdown structure the reference designations can actively be used as identifiers within the system associated with the object of interest position within the process hierarchy.

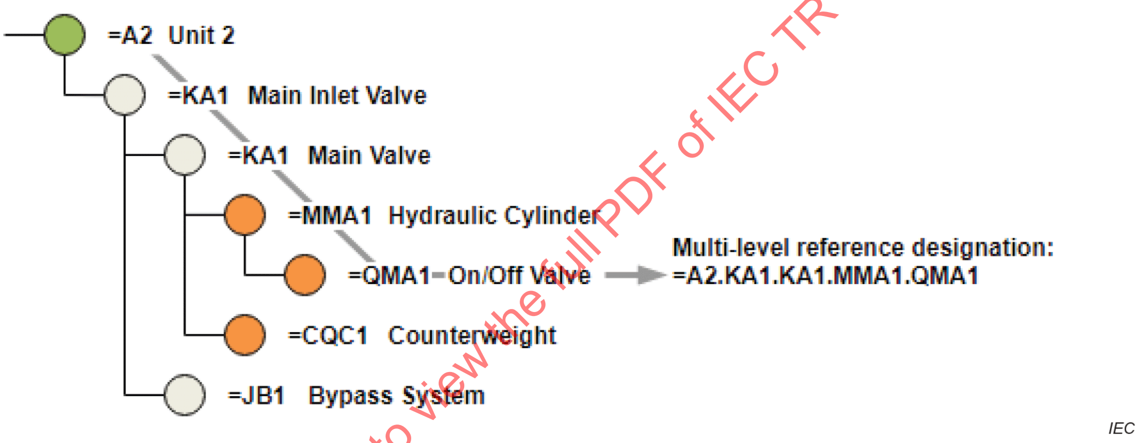


Figure 12 – A system breakdown structure for a system of interest

Reference designations are created in compliance with the rules of IEC 81346-1 and come in two forms:

Single-level reference designations are the reference designations that are formed by simply classifying the object and omitting any relationship to parent objects within the breakdown structure. In Figure 12 the code =QMA1 is a single-level reference designation of the On/Off Valve.

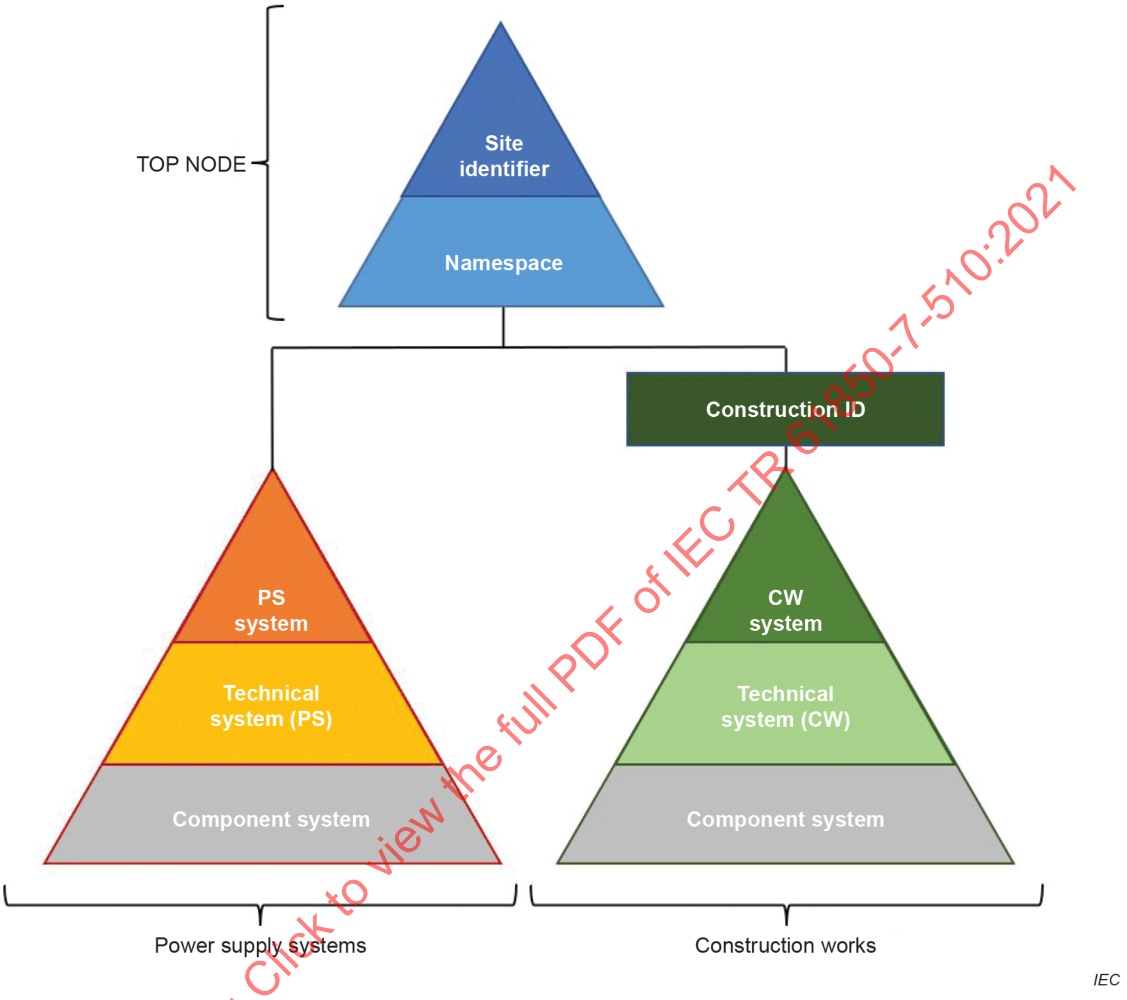
Multi-level reference designations are the reference designations that are formed by concatenating single-level reference designations according to the breakdown structure. They represent the full reference designation of the system element. In Figure 12 the code =A2.KA1.KA1.MMA1.QMA1 is the full multi-level reference designation of the highlighted On/Off valve.

5.1.7 The top node

The top node is what gives any reference designation its uniqueness. The information contained within the top node is not regulated by the IEC/ISO 81346 standard series. However, two pieces of information should be found in it: A unique identifier, and namespace information. Besides this, it can be advisable to refer to modelling guideline used to create the breakdown structure (referred to as "profile").

As per IEC/ISO 81346-1 (2009) angle brackets (<...>) are used to identify the top node identifier.

Figure 13 is a suggestion to a top node structure.



RDS Top node reference for Alpha Valley River System

Power Supply system:<{SiteIdentifier}.{Namespace}>.Systems

Construction Works:<{SiteIdentifier}.{Namespace}>Structure.Systems

Figure 13 – Example of an RDS top node implementation

1. Site identifier

Indicates a geographic name or code (such Alpha1 or 2109), often the powerplant name itself or a code-id associated to it. The identifier needs to end with a number but does not have to contain letters.

Note that, in the case of a name spelt or abbreviated, followed by a number, the number could be used as a site classification. This should be tailored to the need of the company, and well described in a company profile.

2. Namespace (RDS)

Points to the reference designation classes used. Within ISO 81346, parts 10[2] and 12 define class-tables for Power Supply systems and Construction Works. The abbreviations PS and CW could be used along with a numbering indicating the edition of the standard parts.

Ex: PS1 or CW1

3. Namespace attribute profile

Refers to the application profile, like the one this document represents. This information will aid in the understanding of the structure. The profile is a specialisation of the Normative namespace.

4. Structure

When the structural hierarchy branches to the construction work namespace, a structure element is added to designate a physical civil structure within a site that contains RDS systems.

5.2 SCL modelling of the functional structure of a hydropower plant

The functional structure of a hydropower plant can be modelled according to the SCL:process hierarchical structure described in IEC 61850-6.

A typical hydropower plant will include a "substation" part that will be identical to what is described in the IEC 61850 series. The hydropower works with the hydropower plant and the generating units with their related equipment are represented as an SCL: processes in the functional structure.

To create a functional structure, it is necessary to identify how the functional blocks (SCL:processes) are related to each other. A generating unit consists of a turbine-generator set with dedicated auxiliary equipment and supporting functions. Other auxiliary equipment may be related to the hydropower plant itself rather than as a subfunction of a generating unit. Generator transformers can be referenced in the same way as substation transformers; there is not always any one-to-one connection between generating units and transformers.

The functional structure also applies to dam structures. It is common to associate at least one dam with each hydropower plant. There are also examples of reservoirs that are not related to any specific power plant, equally there are power plants from which more than one dam are being controlled. There can also be dams with more than one hydropower plant. The actual relationships between dams and hydropower plants is a local issue. The SCL: process structure is used to represent the hierarchical parent-child relationships.

In the abstract functional structure of the hydropower plant, a dam is modelled as an SCL:function of the SCL:process Reservoir. Figure 14 and Figure 15 show examples of how the relationships of a hydropower plant can be represented by SCL objects.

The identification of processes in a hydropower plant is a local issue. The processes should be chosen such that they can be designated according to the RDS system.

Once the processes have been identified according to RDS they can be mapped to SCL Process elements. This forms a two-step procedure.

---

[2] IEC/ISO 81346 class definitions are subject to changes before the publication of ISO 81346-10:—.

The RDS levels can be mapped to a corresponding hierarchy of SCL elements.

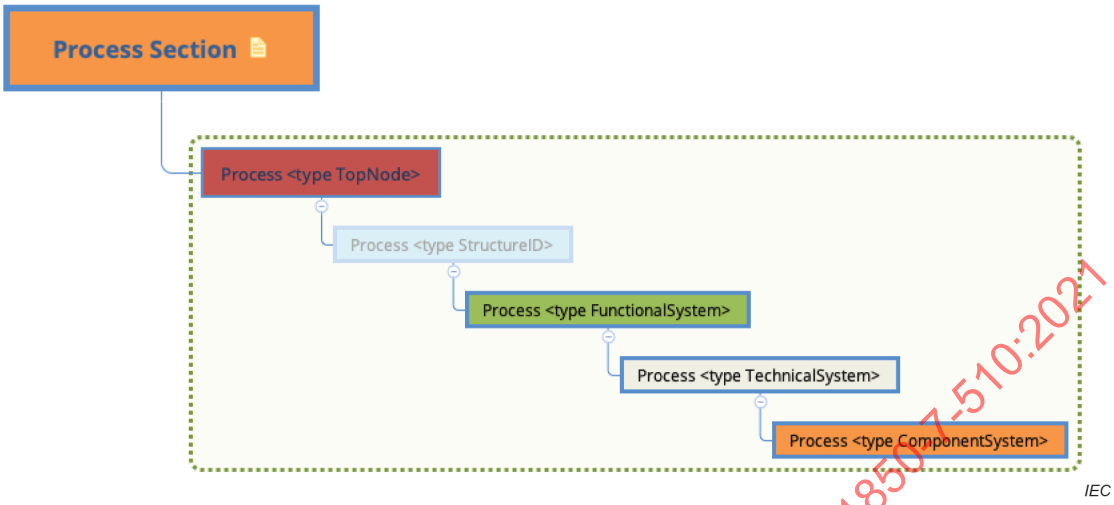


Figure 14 – SCL Process elements are structured according to the RDS power supply system designations

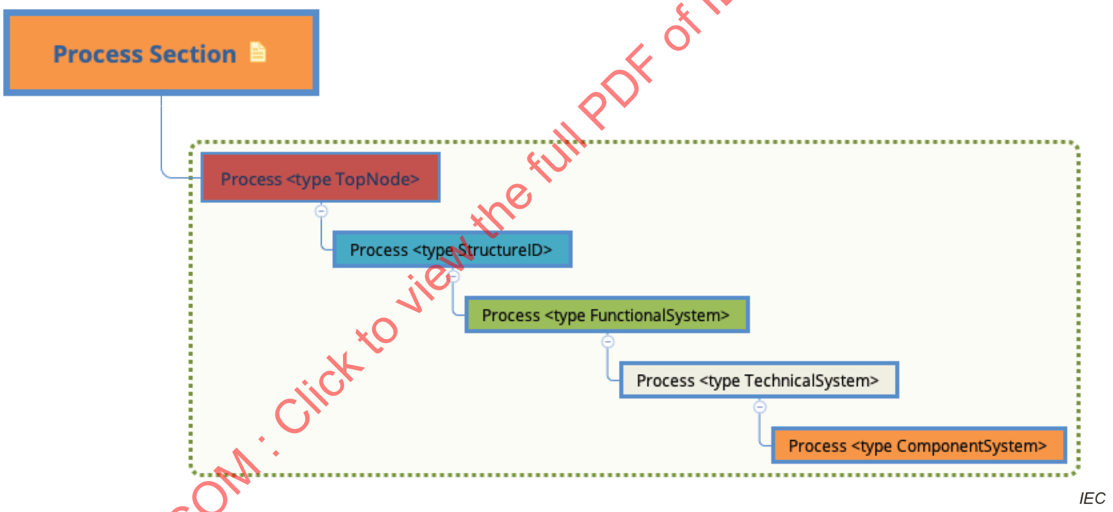


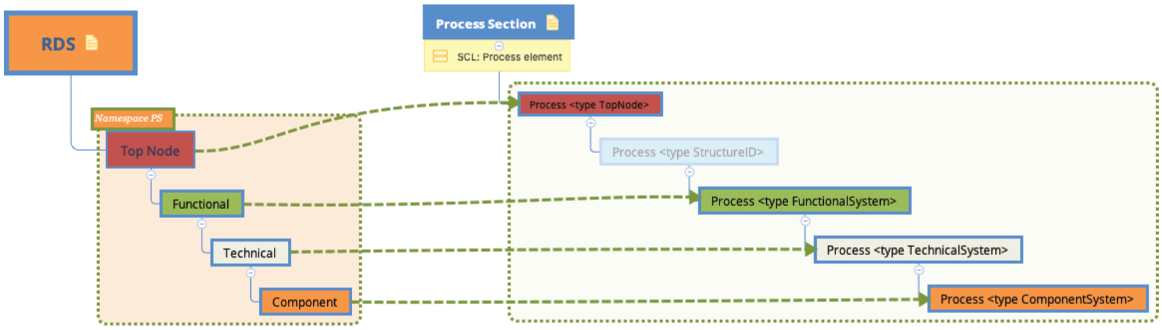
Figure 15 – SCL Process elements are structured according to the RDS construction works designations

5.3 Mapping the SCL process structure to the reference designation system RDS

5.3.1 General

The RDS levels each correspond to one layer of SCL Process elements in the structure. In Figure 16 and Figure 17, there is only one SCL Process element at each level in the hierarchy. Depending on the process structure there might be several parallel SCL Process elements on each level in the hierarchy.

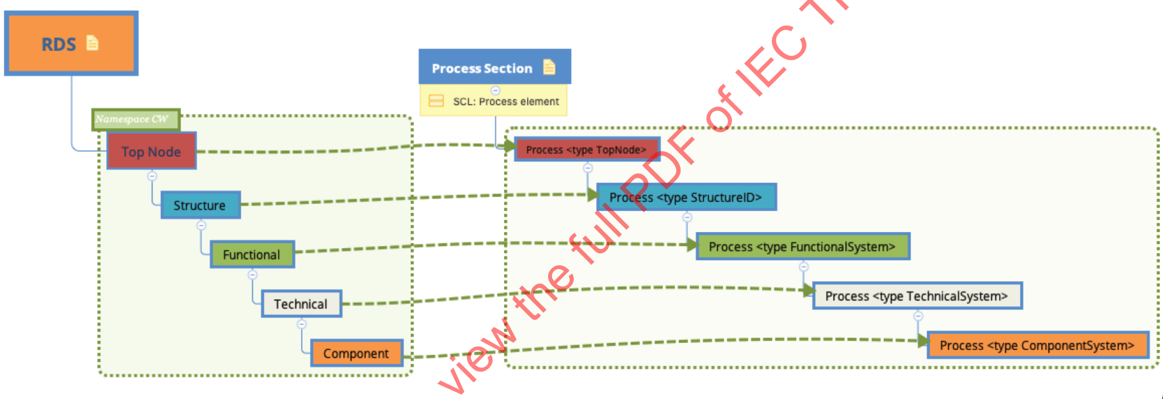
In order to connect Logical Nodes (LN) to the process hierarchy, SCL Function elements are used. Here, the SCL Function elements are used as containers for the LNs only and are not given any semantical meaning. In Clause 9 it is shown how the SCL function name can be given a locally defined meaning.



Key:

On the left-hand side the RDS-PS structure is mapped onto the corresponding process section in the SCL file. Since in the PS namespace the "StructureID" element is not relevant, the FunctionalSystem will be directly allocated under the TopNode element, thus skipping that element.

Figure 16 – IED model (LNs) linked to the SCL Process structure with the power supply system profile



Key:

On the left-hand side the RDS-CW structure is mapped onto the corresponding process section in the SCL file. In this case all RDS levels are mapped to a corresponding SCL process element, relevant types.

Figure 17 – IED model (LNs) linked to the SCL Process structure with the construction works profile

5.3.2 Hierarchical mapping of information

Table 2 provides the mapping between SCL and RDS-PS. This hierarchical layered mapping provides the relation between SCL Process elements and the RDS system layers. The SCL Function acts as the ultimate link to the IEC 61850 data structure, starting with the logical node instance.

In SCL, the RDS profile is omitted in the hierarchy.

The SCL uses the element Process in this document along the following way:

- an element that hierarchically performs a series of mechanical, electrical or electronic operations to change or preserve a state. For example, a power plant, reservoir, lubrication pump, water treatment plant, spillway gate, a sensor would be considered to be a process-element.

The SCL uses the element Function in this document along the following way:

- an element that performs a series of logical operations to change or preserve a state;
- the element function is a child of a process-element and parent to a subfunction;
- for example, speed regulation, speed matching, voltage regulation, synchronisation, the start-stop sequence would be considered functions.

Table 2 – Mapping SCL to RDS-PS

A mapping between SCL and RDS		
SCL Element **Add attribute for RDS**	**RDS -PS**	**RDS -CW**
SCL/Process/Process/@name	Tope node Site Identifier	Tope node Site Identifier
SCL/Process/Process/Function	Transparent link to a Logical node	Transparent link to a Logical node
SCL/Process/Process/Process/@name	Topnode Namespace	Topnode Namespace
SCL/Process/Process/Process/Function	Transparent link to a Logical node	Transparent link to a Logical node
SCL/Process/Process/Process/Process/ @name	Functional System	Structure
SCL/Process/Process/Process/Process/ Function	Transparent link to a Logical node	Transparent link to a Logical node
SCL/Process/Process/Process/Process/ Process/@name	Technical System	Functional System
SCL/Process/Process/Process/Process/ Process/Function	Transparent link to a Logical node	Transparent link to a Logical node
SCL/Process/Process/Process/Process/ Process/Process/@name	Technical System (subsystem)	Technical System
SCL/Process/Process/Process/Process/ Process/Process/Function	Transparent link to a Logical node	Transparent link to a Logical node
SCL/Process/Process/Process/Process/ /Process/Process/Process/@name	Component System	Technical System (subsystem)
SCL/Process/Process/Process/Process/ Process/Process/Process/Function	Transparent link to a Logical node	Transparent link to a Logical node
SCL/Process/Process/Process/Process/ /Process/Process/Process /Process/@name		Component System
SCL/Process/Process/Process/Process/ Process/Process /Process/Process/Function		Transparent link to a Logical node

Table legend:

<SCL parent element>
<SCL child element>

Note that there is however no standardised way of arranging overall control functions, the structure will depend on whether the plant is manned or remote-controlled, as well as traditions within the utility that owns the plant. In order to cover most arrangements, some of the Logical Nodes defined in the standard document are more or less overlapping. This will allow the user to arrange Logical Devices by selecting the most appropriate Logical Nodes that suit the actual design and methods of operation of the plant. Other Logical Nodes are very small, to provide simple building blocks that will allow as much freedom as possible in arranging the control system.

Note that RDS-CW includes one more level compared with RDS-PS, causing the bottom cell to be blank for RDS-PS in Table 2.

5.3.3 Process object reference design considerations

In this document, the full stop sign "." is used as separator between process (and function) object designations to build up an object reference. The transition between RDS designation and IEC 61850 Logical Node structure is made with the slash sign "/". The transition is denoted in the same way as the transition between IEC 61850 Logical Device and Logical Node structure. In the following sections the designations are written with these separator signs in order to make them easier to read.

Ultimately the use of separator signs will depend on the local implementation and the tools used. Tools may for example not interpret the slash sign "/" as a designation separator sign.

In practice, the RDS designation may be written without any separator tags; even though it will make the designations harder to read for a human.

5.3.4 Choice of logical node classes

Use of measured values typically involves either the raw data, a calculated representation or both. Logical Nodes of group T can be used to represent transducers and sensors sending raw data in the form of measured values to an external function. Logical Nodes of group S is typically used to treat the raw data from the transducer into a calculated representation. For measurement systems, it is proposed that Logical nodes for instrument transformers and sensors (LN Group: T) use an RDS structure referring to the transducer/sensor/transformer in question. These will typically be raw data signals. For logical nodes for Supervision and monitoring (LN Group: S) the RDS structure is expected to end on a higher level in the system hierarchy, aiming only at unambiguity for the signal interpretation.

5.4 The Alpha Valley River System examples

5.4.1 Introduction

The Alpha Valley River System examples cover installations for hydropower production including functions for monitoring and water management.

Figure 18 illustrates the examples for the Alpha Valley hydroelectric power plants.

In the example, different monitoring stations, generating facilities, and dam structures are identified as sites. The designated sites are used to form a process structure for the Hydropower system's SCL representation.

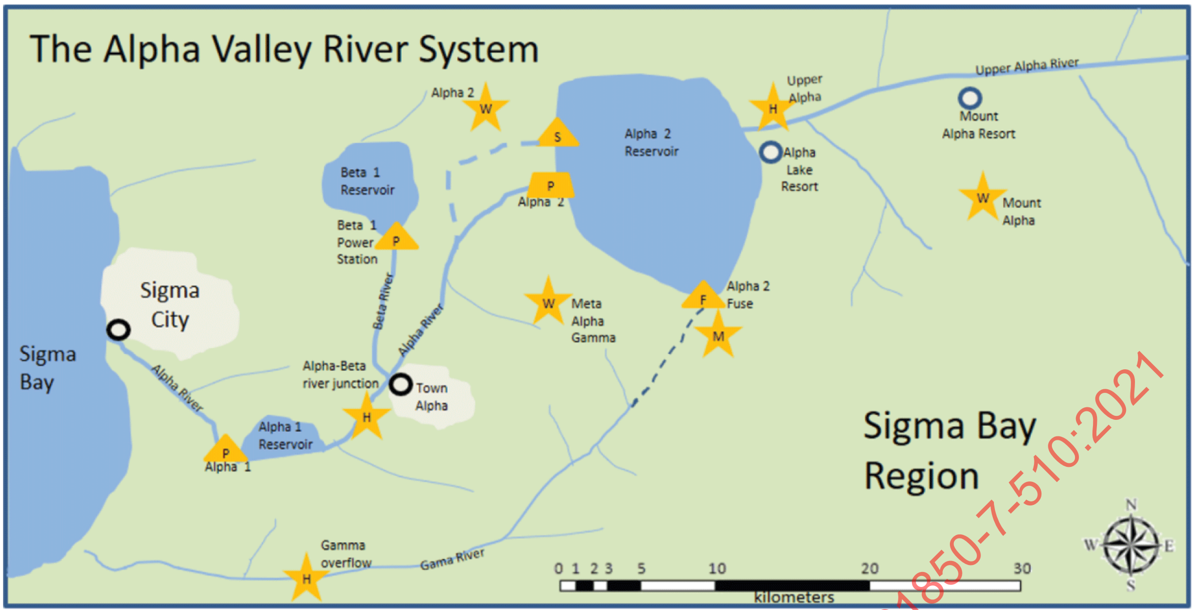


Figure 18 – The Alpha Valley River System example

The examples modelling principles are not intended to be the unique possible or imposed modelling decisions.

The functional aspects of the IEC/ISO 81346 series of Standards are used for the reference designation structure, the reference designations are mapped to the SCL process structure. The IEC/ISO 81346 class levels can be combined to virtually cover all types of system-, and sub-system including system components, for:

- Constructions Works (namespace code: CW)
- Power Supply systems (namespace code: PS)

CW Construction Work and building services (ISO 81346-12:2018). Hydroelectric power construction works are mainly hydro-dams, waterways, tunnels, buildings including related ancillary services (light & small power and HVAC (Heating, Ventilation and Air Condition)).

PS Power Supply system (ISO 81346-10:—). For hydroelectric power installations, the Power Supply systems includes equipment and systems, classically defined as E/M (Electrical/Mechanical) equipment and systems, such as gates, turbines, generators, power transformers including related auxiliary systems.

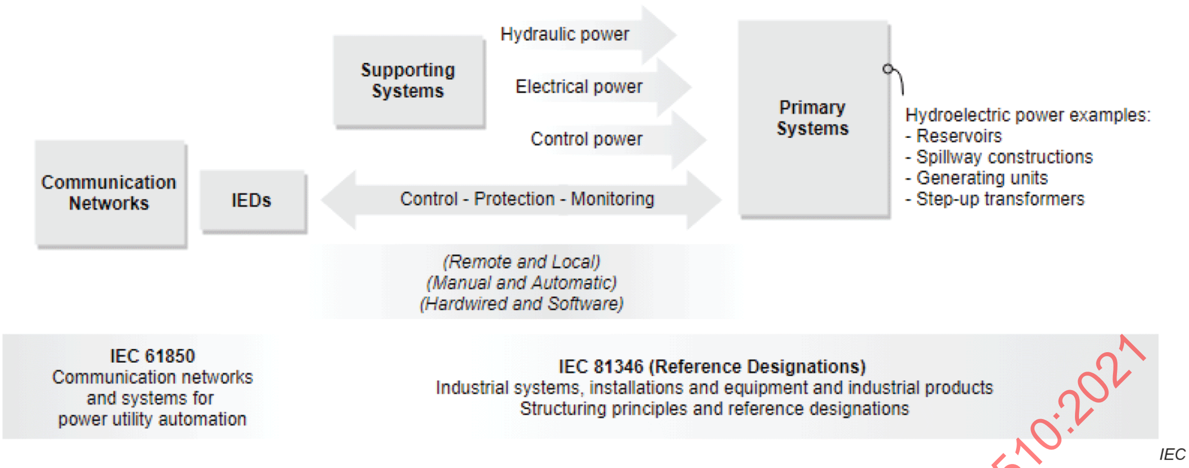


Figure 19 – Primary and supporting system to SCL overview

The IEC/ISO 81346 series of standards can be used to create reference designations for all systems, primary and supporting, including individual components such as for systems in Figure 19.

The IEC/ISO 81346 series of standards are more comprehensive and versatile than the IEC 61850 series of standards and includes classes that are not included in the latter series. For example, a class for system elements providing perceptible information; site glasses and staff gauges (a measuring device with a graduated scale anchored in the water and dam structures, read by observing the level of the water surface). See Figure 20.

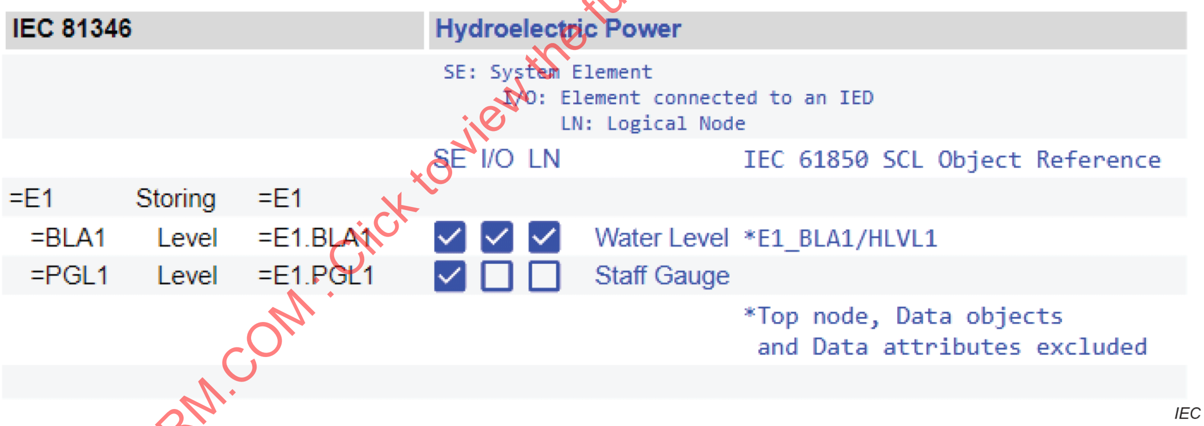
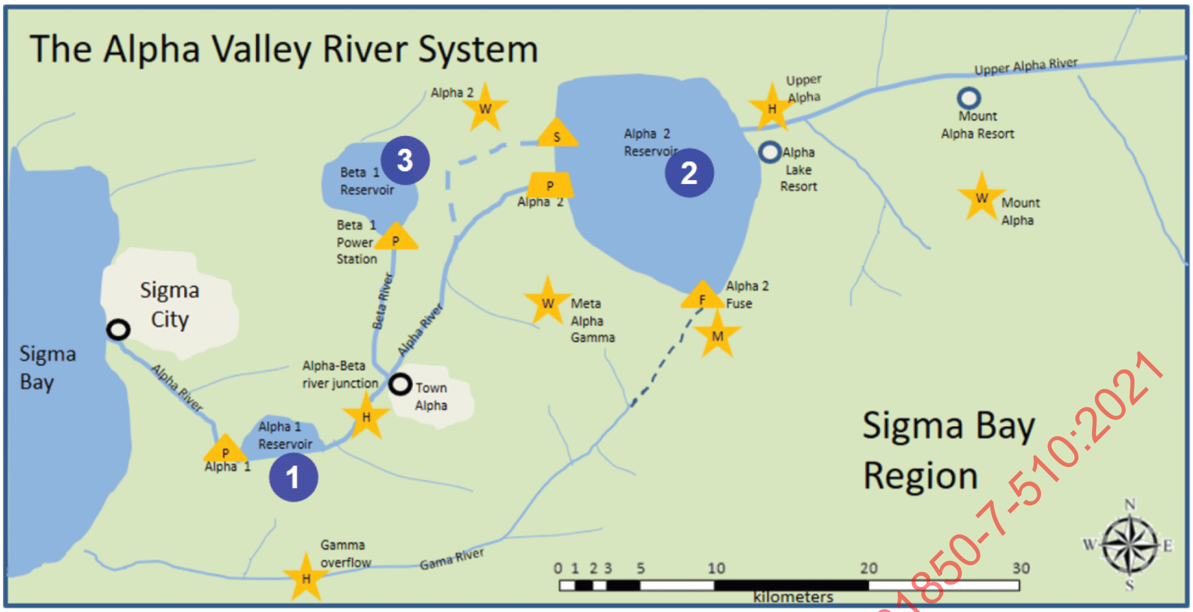


Figure 20 – Mapping between IEC/ISO 81346 (RDS) and IEC 61850 (SCL)

The examples for the Alpha Valley River System project will only include systems elements having a mapping in SCL, one or more LNs (Logical Nodes).

5.4.2 The Reservoirs

Figure 21 locates the three reservoirs.



IEC

Figure 21 – Reservoir locations

Table 3 describes the reservoirs.

Table 3 – Reservoir descriptions

1	Alpha 1 Reservoir.	A weekly reservoir (the Alpha 1 power plant is run-of-river)
2	Alpha 2 Reservoir.	A multipurpose reservoir for recreation and power production (with limited water level variations)
3	Beta 1 Reservoir.	A yearly reservoir (with significant water level variations)

For the examples which follow, the full geographical names for areas and locations are used:

- Alpha1
- Alpha2
- Beta1

The four Alpha 2 locations are in the examples named Alpha2:

- Alpha 2 – W – Weather stations
- Alpha 2 – S – Spillway
- Alpha 2 – P – Powerhouse
- Alpha 2 – F – Fuse Plug

The top nodes used for the examples are based on the geographical names and the namespace code, CW (Construction Works) and PS (Power Supply systems).

The combination of the geographic name and the namespace code creates the top nodes.

<Alpha1.CW1>, <Alpha2.CW1> and <Beta1.CW1> for Construction Works.

<Alpha1.PS1>, <Alpha2.PS1> and < Beta1.PS1> for Power Supply systems.

Each part, name of locations, namespace codes, and component codes ends with a number, where a number is part of the name, such as Alpha 1, an addition of a number is not required.

The three Alpha Valley River System reservoirs primary use is for hydroelectric power production, and therefore the namespace used is PS (Power Supply system). For reservoirs not used for hydroelectric power production, the namespace would typically be CW (Construction Works).

During the various stages of development of the IEC 61850 series of standards, different logical nodes have been developed, which all includes data for level measurements, where a logical node represents more than a single sensor measurement, For project consistency, it is essential to specify which logical nodes and data objects to use and their purpose.

5.4.3 Hydrometric

5.4.3.1 General

Reservoir water surface elevation, i.e. water level typically unit is MAMSL (Meters Above Mean Sea Level), or MAML (Meters Above Mean Level) is the primary measurement for monitoring of reservoirs. If not otherwise stated, the following examples do only include examples of water levels and rate of discharge, to exemplify mapping between ISO 81346-10 and IEC 61850.

Notice: The analogue value representation type; ".i" for INT32 and ".f" FLOAT32 are excluded from the IEC 61850 object references, example:

- "MHYD1.Lev.mag.i" is included as "MHYD1.Lev.mag"
- "MHYD1.Lev.mag.f" is included as "MHYD1.Lev.mag"

5.4.3.2 Alpha 2 water level measurements

Table 4 describes the examples representing different solutions for Alpha 2 head- and tailwater level measurements.

Table 4 – Examples of water level measurements

Examples	Logical node	Title	Logical node group
Figure 22	TLVL	Media level sensor	T: Instrument transforms and sensors
Figure 23	HLVL	Water level indicator	H: Hydropower
Figure 24	MHYD	Hydrological information	M: Metering and measurement

#	| Project: Alpha Valley River System | Alpha 2 Water Levels |			
	| Plant: Alpha 2 | Top node: <Alpha2.PS1> |			
	IEC 81346-10 (reference designation)			IEC 61850 (object reference)
	E Storing System H Disposing BLA Level, with scalar output			TLVL Media level sensor .LevPctSv.instMag Sampled media level measurement
	=E1	<Alpha2.PS1>=E1	Headwater	
1	=BLA1	<Alpha2.PS1>=E1.BLA1	Measurement 1	Alpha2PS1_E1_BLA1/TLVL1.LevPctSv.instMag
2	=BLA2	<Alpha2.PS1>=E1.BLA2	Measurement 2	Alpha2PS1_E1_BLA2/TLVL1.LevPctSv.instMag
3	=BLA3	<Alpha2.PS1>=E1.BLA3	Measurement 3	Alpha2PS1_E1_BLA3/TLVL1.LevPctSv.instMag
	=H2	<Alpha2.PS1>=H2	Tailwater	
4	=BLA1	<Alpha2.PS1>=H2.BLA1	Measurement 1	Alpha2PS1_H1_BLA1/TLVL1.LevPctSv.instMag

IEC

Figure 22 – Mapping of water levels with logical node TLVL

#	| Project: Alpha Valley River System | Alpha 2 Water Levels |			
	| Plant: Alpha 2 | Top node: <Alpha2.PS1> |			
	IEC 81346-10 (reference designation)			IEC 61850 (object reference)
	E Storing System H Disposing BLA Level, with scalar output			HLVL Water level indicator .LevM.mag Water level
	=E1	<Alpha2.PS1>=E1	Headwater	
1	=BLA1	<Alpha2.PS1>=E1.BLA1	Measurement 1	Alpha2PS1_E1_BLA1/HLVL1.LevM.mag
2	=BLA2	<Alpha2.PS1>=E1.BLA2	Measurement 2	Alpha2PS1_E1_BLA2/HLVL1.LevM.mag
3	=BLA3	<Alpha2.PS1>=E1.BLA3	Measurement 3	Alpha2PS1_E1_BLA3/HLVL1.LevM.mag
	=H2	<Alpha2.PS1>=H2	Tailwater	
4	=BLA1	<Alpha2.PS1>=H2.BLA1	Measurement 1	Alpha2PS1_H1_BLA1/HLVL1.LevM.mag

IEC

Figure 23 – Mapping of water levels with logical HLVL

#	| Project: Alpha Valley River System | Alpha 2 Water Levels |			
	| Plant: Alpha 2 | Top node: <Alpha2.PS1> |			
	IEC 81346-10 (reference designation)			IEC 61850 (object reference)
	E Storing System H Disposing BLA Level, with scalar output			MHYD Hydrological information .Lev.mag Water level
	=E1	<Alpha2.PS1>=E1	Headwater	
1	=BLA1	<Alpha2.PS1>=E1.BLA1	Measurement 1	Alpha2PS1_E1_BLA1/MHYD1.Lev.mag
2	=BLA2	<Alpha2.PS1>=E1.BLA2	Measurement 2	Alpha2PS1_E1_BLA2/MHYD1.Lev.mag
3	=BLA3	<Alpha2.PS1>=E1.BLA3	Measurement 3	Alpha2PS1_E1_BLA3/MHYD1.Lev.mag
	=H2	<Alpha2.PS1>=H2	Tailwater	
4	=BLA1	<Alpha2.PS1>=H2.BLA1	Measurement 1	Alpha2PS1_H1_BLA1/MHYD1.Lev.mag

IEC

Figure 24 – Mapping of water levels with logical MHYD

5.4.3.3 Alpha 2 reservoir discharge measurements

For hydroelectric power plants, where the rate of flow is used for control, the rate of discharge, i.e. rate of flow is typically calculated from predetermined rating tables. When using tables, the logical node FCSD (Curve shape description) can be used for the relation between water level, the opening of the object (such as a valve, gate or a turbine guide position), and rate of flow. Rate of flow in penstocks can accurately be measured; such measuring systems are typically not installed in hydroelectric power plants.

From an operation and control point of view, there is no difference if the rate of flow is calculated from predetermined rating tables or measured. Both physically measured (from sensors, transmitters, transducers etc.) and calculated measurements are from the ISO 81346-10 *functional* aspect treated the same, i.e. =BFA (flow, with scalar output), is used for both physically measured and calculated measurements.

If not otherwise stated, the following examples do only include examples of the rate of discharge, to exemplify mapping between ISO 81346-10 and IEC 61850.

Table 5 describes the examples representing different solutions for Alpha 2 rate of discharge measurements.

Table 5 – Examples of the rate of discharge measurements

Examples	Logical node	Title	Logical node group
Figure 25	TFLW	Liquid flow sensor	T: Instrument transforms and sensors
Figure 26	HWCL	Water control	H: Hydropower
Figure 27	MHYD	Hydrological information	M: Metering and measurement

#	| Project: Alpha Valley River System | Alpha 2 Rate of Discharge |			
	| Plant: Alpha 2 | Top node: <Alpha2.PS1> |			
	IEC 81346-10 (reference designation)			IEC 61850 (object reference)
	A Transforming (forms of energy) H Disposing KA Control of flow BFA Flow, with scalar output			TFLW Liquid flow sensor .FlwSv.instMag Sampled liquid flow rate measurement
	=A1	<Alpha2.PS1>=A1	Unit 1	
1	=BFA1	<Alpha2.PS1>=A1.BFA1	Discharge	Alpha2PS1_A1_BFA1/TFLW1.FlwSv.instMag
	=A2	<Alpha2.PS1>=A2	Unit 2	
2	=BFA1	<Alpha2.PS1>=A2.BFA1	Discharge	Alpha2PS1_A1_BFA1/TFLW1.FlwSv.instMag
	=H1	<Alpha2.PS1>=H1	Spillway 1	
	=KA1	<Alpha2.PS1>=H1.KA1	Gate 1	
3	=BFA1	<Alpha2.PS1>=H1.KA1.BFA1	Discharge	Alpha2PS1_H1_KA1_BFA1/TFLW1.FlwSv.instMag
	=KA2	<Alpha2.PS1>=H1.KA2	Gate 2	
4	=BFA1	<Alpha2.PS1>=H1.KA2.BFA1	Discharge	Alpha2PS1_H1_KA2_BFA1/TFLW1.FlwSv.instMag

IEC

Figure 25 – Mapping of the rate of discharge with logical node TFLW

Figure 26 is an example of automated gate control, the flow setpoints and measurements are included.

#	| Project: Alpha Valley River System | Alpha 2 Rate of Discharge |			
	| Plant: Alpha 2 | Top node: <Alpha2.PS1> |			
	IEC 81346-10 (reference designation)			IEC 61850 (object reference)
	H Disposing KA Control of flow			HWCL Water control .FlwSpt.mxVal Water flow setpoint .Flw.mag Calculated water flow
	=H1	<Alpha2.PS1>=H1	Spillway 1	
	=KA1	<Alpha2.PS1>=H1.KA1	Gate 1	Alpha2PS1_H1_KA1/HWCL1
			Flow setpoint	Alpha2PS1_H1_KA1/HWCL1.FlwSpt.mxVal
1			Discharge	Alpha2PS1_H1_KA1/HWCL1.Flw.mag
	=KA2	<Alpha2.PS1>=H1.KA2	Gate 2	Alpha2PS1_H1_KA2/HWCL1
			Flow setpoint	Alpha2PS1_H1_KA2/HWCL1.FlwSpt.mxVal
2			Discharge	Alpha2PS1_H1_KA2/HWCL1.Flw.mag

IEC

Figure 26 – Mapping of the rate of discharge with logical node HWCL

#	| Project: Alpha Valley River System | Alpha 2 Rate of Discharge |			
	| Plant: Alpha 2 | Top node: <Alpha2.PS1> |			
	IEC 81346-10 (reference designation)			IEC 61850 (object reference)
	A Transforming (forms of energy) H Disposing KA Control of flow BFA Flow, with scalar output			MHYD Hydrological information .Flw.mag Flow rate of water
	=A1	<Alpha2.PS1>=A1	Unit 1	
1	=BFA1	<Alpha2.PS1>=A1.BFA1	Discharge	Alpha2PS1_A1_BFA1/MHYD1.Flw.mag
	=A2	<Alpha2.PS1>=A2	Unit 2	
2	=BFA1	<Alpha2.PS1>=A2.BFA1	Discharge	Alpha2PS1_A1_BFA1/MHYD1.Flw.mag
	=H1	<Alpha2.PS1>=H1	Spillway 1	
	=KA1	<Alpha2.PS1>=H1.KA1	Gate 1	
3	=BFA1	<Alpha2.PS1>=H1.KA1.BFA1	Discharge	Alpha2PS1_H1_KA1_BFA1/MHYD1.Flw.mag
	=KA2	<Alpha2.PS1>=H1.KA2	Gate 2	
4	=BFA1	<Alpha2.PS1>=H1.KA2.BFA1	Discharge	Alpha2PS1_H1_KA2_BFA1/MHYD1.Flw.mag

IEC

Figure 27 – Mapping of the rate of discharge with logical node MHYD

6 SCL:DataType template modelling

6.1 General

The DataType template is the basis for defining the nomenclature profile for logical node, data objects, data attribute types.

The use of the profile might be project specific, defined by a Power Company and used in several of the company's implementations, or be defined as a national standard, etc.

In the SCL file, the Datatype template element is composed of the following children elements:

1) LNodeType (logical node type definition)
2) DOtype (data object type definition)
3) DAtype (data attribute type definition)
4) EnumType (enumeration type definition)

It is a good practice to form type names by parts according to the abbreviations listed in IEC 61850-7-4 and IEC 61850-7-410.

NOTE "SIGMA" represents the fictitious "Sigma power corporation". For further details please refer to the Annex B for the SCL example.

When versioning and revising types, versions should use year and revisions should use a letter A to Z, as per IEC 61850-7-7.

6.2 LNodeType definition

It is a good practice that once the modelling on the process section is completed, LN instances must be allocated to the different process elements, under relevant function container. Each LN instance allocated in the process elements must be linked to a LNodeType in the relevant DataType Template section of the SCL document. Each LNodeType is a specialisation of a LNClass, and it is uniquely defined by its name.

Inside each LNodeType definition, different DOs are defined. Since the type definition for each and every DO is still to be defined, a proper selection inside each CDC must be carried out to select which optional DOs are relevant.

The following table shows an exemplary way to concatenate a LNodeType names:

LNodeType	<Author>	<LNClass>	<Variant name>	<Version>	<Revision>
	SIGMA_	MMXU_	TotW_TotVAr_	2020	A
	SIGMA_	MMXU_	W_Var_A_PPV_	2020	A

The following is an example of LNodeType in the SCL form:

```
<LNodeType id="SIGMA_MMXU_ TotW_TotVAr_2020A" lnClass="MMXU">
        <DO name="Mod" type="SIGMA_ENC_Mod_2020A"/>

        <DO name="Beh" type="SIGMA_ENS_Beh_2020A"/>

        <DO name="Health" type="SIGMA_ENS_Health_2020A"/>

        <DO name="NamPlt" type="SIGMA_LPL_2020A"/>

        <DO name="TotW" type="SIGMA_MV_MW_2020A"/>

        <DO name="TotVAr" type="SIGMA_MV_MVar_2020A"/>

</LNodeType>
```

6.3 DOType definition

Inside each DOType definition, different DAs are defined. Since the type definition for each and every DA is still to be defined, a proper selection inside each CDC must be carried out, to select which optional DAs are relevant for the project. Enumeration DAs are also to be defined, one for each enumeration type which is used inside the project. Enumeration types are defined on the following DAType section.

The following table shows an exemplary way to concatenate a DOType names:

DOType	<Author>	<DOClass>	<Variant name>	<Version>	<Revision>
	SIGMA_	MV_	MW_	2020	A
	SIGMA_	WYE_	A_	2020	A
	SIGMA_	DEL_	PPV_	2020	A

The following is an example of DOType in the SCL form:

```
<DOType id="SIGMA_MV_ MW_2020A" cdc="MV">

<DA name="mag" bType="Struct" type="SIGMA_AnalogueValue_f_2020A" fc="MX"/>

<DA name="q" bType="Quality" fc="MX"/>

<DA name="t" bType="Timestamp" fc="MX"/>

<DA name="units" bType="Struct" type="SIGMA_Units_MW_2020A" fc="CF"/>

<DA name="d" bType="VisString255" fc="DC">

        <Val>MW</Val>
```

```
</DA>

<DA name="dU" bType="Unicode255" fc="DC">

<Val>Active power measurement</Val>

</DA>

</DOType>
```

6.4 DAType and EnumType definition

Inside each DAType definition, predefined values Enumeration DOs are also to be defined, one for each enumeration type which is used inside the project.

The following table shows an exemplary way to concatenate a DAType names:

DAType	<Author>	<DAClass>	<Variant name>	<Version>	<Revision>
	SIGMA_	Units_	MW_	2020.	A
	SIGMA_	Units_	kA_	2020	A
	SIGMA_	Units_	kV_	2020	A

The following is an example of DAType in the SCL form:

```
<DAType id="SIGMA_Units_MW_2020A">

        <BDA name="SIUnit" bType="Enum" type="SIGMA_SIUnitKind_2020A">

          <Val>62</Val>

        </BDA>

        <BDA name="multiplier" bType="Enum" type="SIGMA_MultiplierKind_2020A">

        <Val>6</Val>

        </BDA>

</DAType>

<DAType id="HQ_AnalogueValue_f_2020A">

        <BDA name="f" bType="FLOAT32"/>

</DAType>
```

The following table shows an exemplary way to concatenate a EnumType names:

EnumType	<Author>	<Kind>	<Version>	<Revision>
	SIGMA_	SIUnitKind_	2020	A
	SIGMA_	MultiplierKind _	2020	A

6.5 Example using SLVL

Figure 28 provides a multilayer view of how the type definition are linked together. In this example, the figure shows how the LN class SIGMA_SLVL_LevPct_2020A's DO LevPct is tied to DOType SIGMA_MV_LevPct_2020A and further down how its DAs are tied to SIGMA_AnalogueValue_f_2020A.

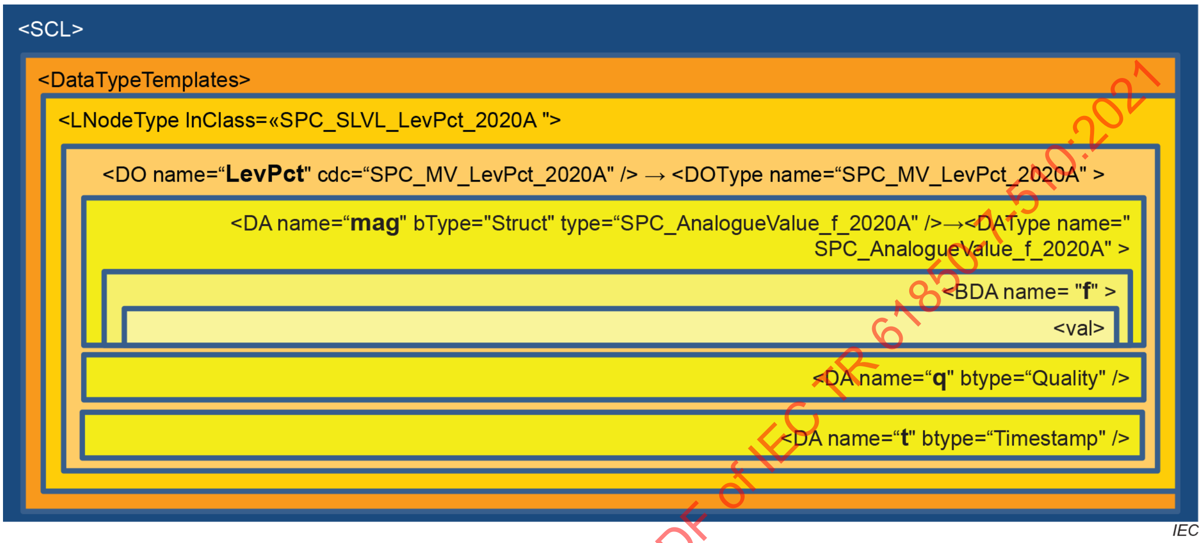


Figure 28 – The structure of LN SLVL

7 SCL:IED modelling

7.1 General

The IED model uses the IED section in the SCL to describe the information. This model is closely tied to the electronic asset responsible doing the acquisition and control of a process. Thus its structure could greatly differ from the process model. The IED consists of four layers IED, access point (AP), server, logical device (LD) before reaching a logical node (LN).

The IED model is normally defined by the IED supplier. Certain IED suppliers may provide a means to manage the IED model on a configuration tool and allow a third party to modify the model.

The IED hosts the Logical node instances linked to a logical node type defined in the SCL:DataTypeTemplates.

7.2 Linking the SCL:IED model to the SCL:process model

This information view differs from the process modelling view thus a mapping between the two views is required. The process view uses a layered approach of processes to finally reach a transparent function that acts as a reference to the IED view as shown in Figure 29.

7.3 Referencing the Logical Device

The RDS structure along with the functional naming may be concatenated to a Logical Device instance. For information regarding Function-related naming in SCL, refer to 8.5.3 of IEC 61850-6:2009+AMD1:2018.

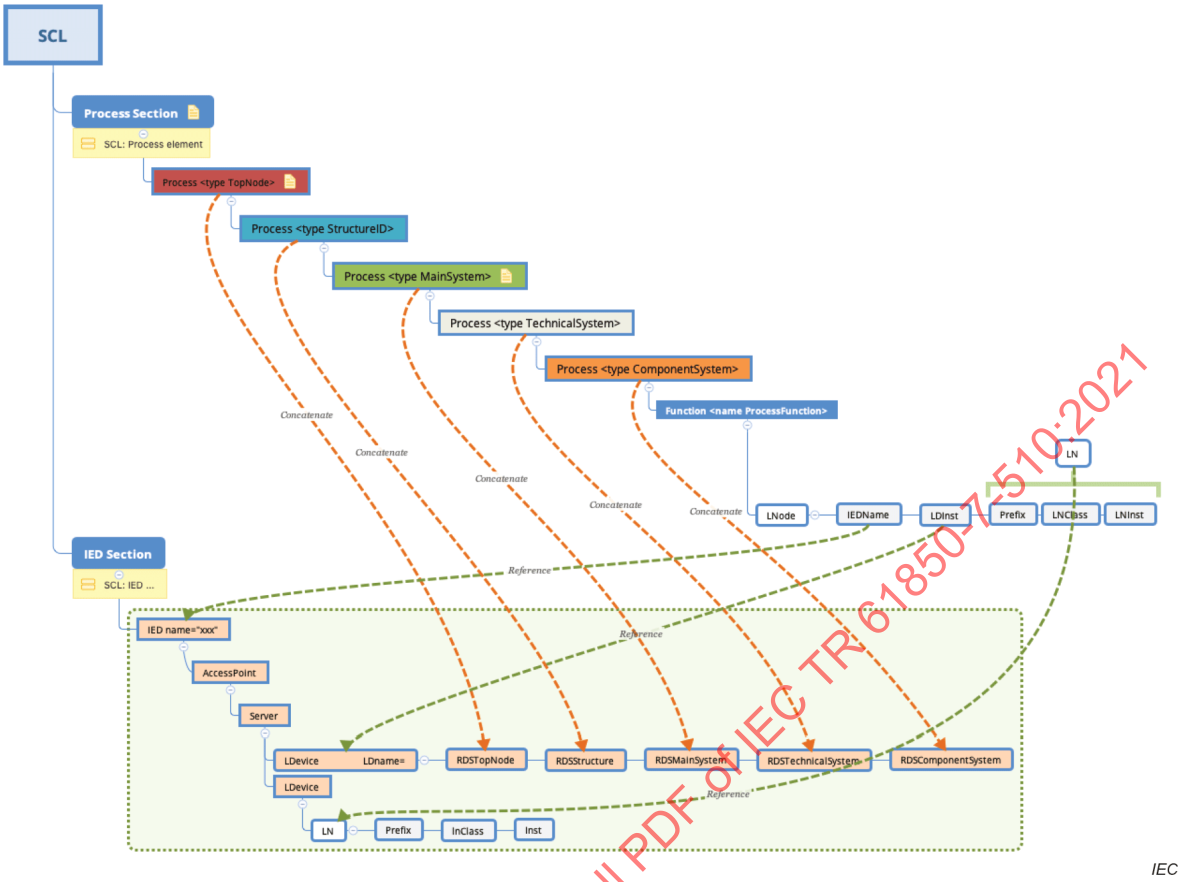
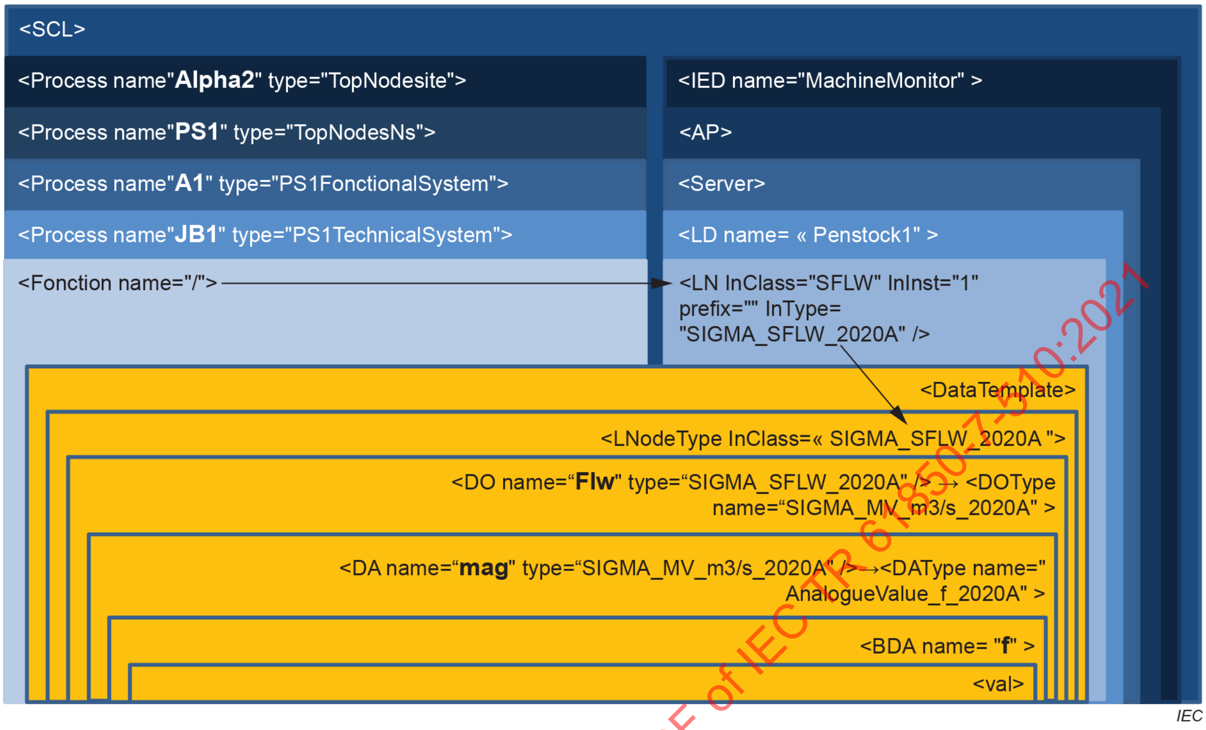


Figure 29 – Schematic mapping of the process element to IED

The IED elements are then linked to the DataType templates as shown in Figure 30. In the example below, the flow measurement for the Alpha river penstock is implemented in a Logical Device "Penstock1" located in a physical IED named "MachineMonitor".



Process tag: **Alpha2PS1A1JB1/SFLW1.Flw.mag.f**

IED tag: **MachineMonitorPenstock1/SFLW1.Flw.mag.f**

Figure 30 – Mapping the process element to IED and DataTemplate

Figure 30 shows an example of how to concatenate the process designation elements to form a logical device name. This example shows that both process modelling and physical implementation are pointing to the same information (LNinst.DO.DA).

7.4 SCL:Function element

The SCL:Function is only used as a container for the IEC 61850 Logical Node. However, the SCL:Function name itself can be used to hold information. The meaning of the SCL:Function name is always a local issue and is not standardized. For example, the SCL:Function name may be used to hold information about the type of regulating algorithm used.

8 Communication Modelling

8.1 General

Power generation communication networks may use different physical (OSI layer 1) network topologies such as radial and ring topologies. By using a bus topology with station and process bus according to IEC 61850, it is possible to abstracts the physical network links. This provides a mean to simplify the generic representation of the communication model. Figure 31 provides the communication model a power generation installation. In many implementations, the routing function is replaced by an RTU (remote terminal unit).

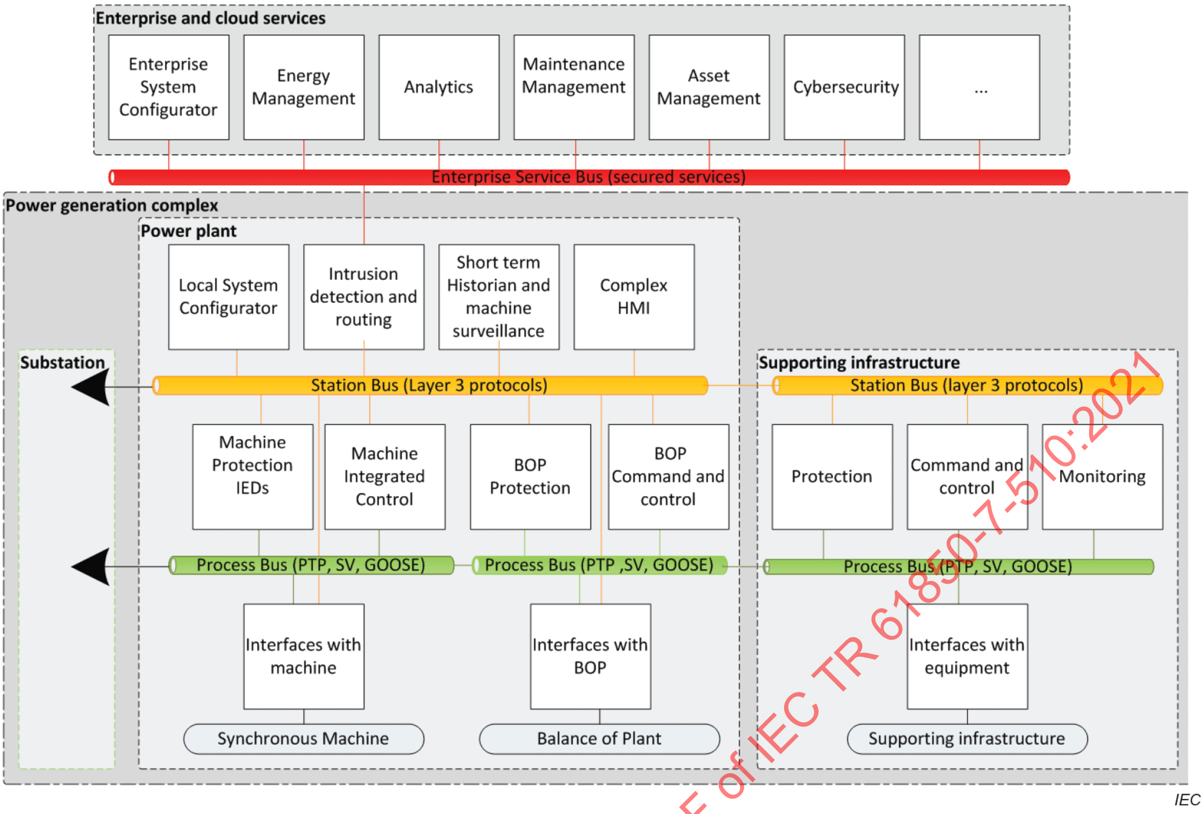


Figure 31 – Bus and services example

Figure 31 illustrates an example of communication busses and services.

In this example an enterprise service bus (ESB) implements an interactive communication framework to provide secure channels between software applications using a service-oriented architecture. This service-oriented layer may use a client-server or publish-subscribe model, such as MMS and GOOSE IEC 61850-8-1 service models.

The Station Bus in this example interconnects all bays with each other and the station supervisory level and carries present function status/values as well as control in sequence such as start-stop, alarms, and operation set points.

The Process Bus in this example interconnects the IEDs within a bay that carries real-time measurements for security such as:

- Sampled Values (SV),
- GOOSE
- Precision time protocol (PTP)

Process Bus goes a step further than Station Bus as it provides the digital link to the primary equipment like switchgear and instrument transformers, sensors on the synchronous machine or sensors on other processes.

The boxed services in the example provide an aggregation of operating services that sustain the power grid. By using the words operating services, we mean the basic services require to operate an electrical grid such voltage and power regulation. Figure 31 provides an abstract example for aggregating operating services.

8.2 Communication structure in hydro power plants

8.2.1 General

The hydro communication model has an ability to extend the substation buses to provide required network services. Figure 32 provides an example of how the communications model can be applied in a hydro power plant.

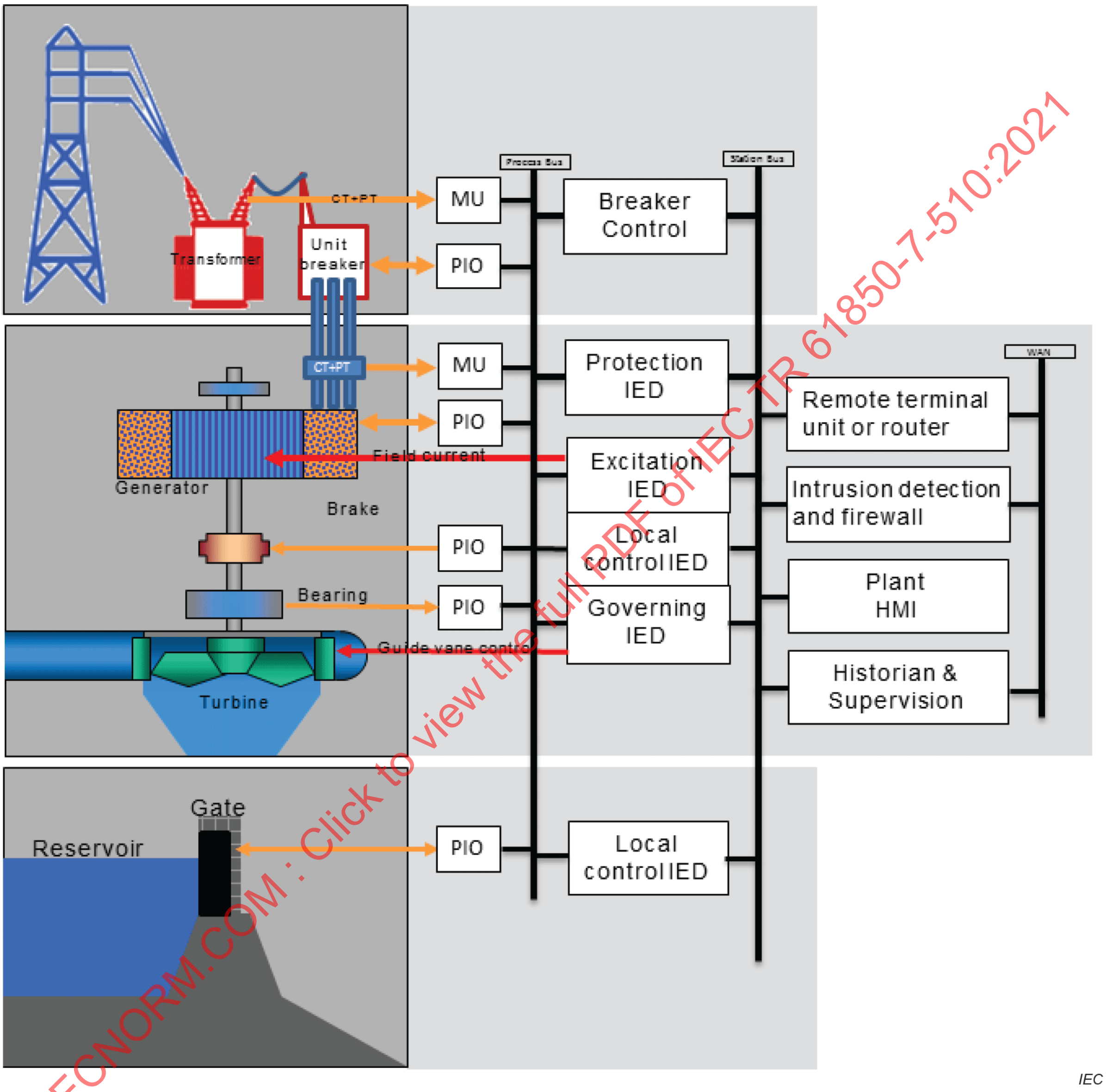


Figure 32 – Hydro bus and services

8.2.2 Process bus level

At process bus level, merging units (MU) and process input outputs (PIO), when interfaced to the process, are able to provide almost all the required data to the operating services. The merging units acquire instantaneous current and voltage reading and convert them into digital reading that are published using the SV or GOOSE protocol.

The control and protection subscribe to information provided by the MUs and PIOs. In some exceptional cases, the IEDs interface directly to the process by bypassing the MUs and PIOs because of performance considerations.

8.2.3 Station Bus

The station bus provides a means to gather the communications from the substation, the turbine generating units and the dam. The power station centralised services gather information either for operational or for historical purposes. The Intrusion detection system and firewall provides an access point to push or access information on the enterprise service bus.

8.2.4 Enterprise Bus

The enterprise bus may be used to connect to an enterprise (cloud) service.

8.3 Communication structure in thermal power plants

The different turbines always have their own controllers. As well other systems like the boiler/HRSG or gas compressor have their own controller/PLC. The electrical system is controlled via its own control IEC 61850 electrical control network with dedicated IED's. An interface controller is connecting the electrical control network to the plant control network.

All those systems are connected to the plant controller, where central functions like plant sequencing, Automatic Generation Control (AGC, dispatching between different turbines of a plant) and interfacing to SCADA are covered.

Figure 33 shows a typical communication structure of a combined cycle power plant, using the same platform for all process controllers. In this example, the process controllers do not have an IEC 61850 interface. The controllers are connected with the Plant Control Network (green), which is usually using a proprietary protocol of the control system supplier. To control the electrical systems, a dedicated IEC 61850 network is used.

In the example, the ring architecture of the network is typical only and shall be optimized according to the specific arrangement of the equipment. An interface controller is used to share information between the plant control network (resp. its controllers) and the IEC 61850 network, where the different IED's are connected to. This interface ensures both horizontal communication from the turbine controllers (e.g. to start a motor) and vertical communication to the HMI, to control the electrical system, e.g. to energize the distribution boards or to initiate a bus transfer. The communication of critical signals in the process requires extra care in the system design. Signals like trips from electrical protection could for this reason be directly hardwired to the turbine protection.

The excitation system and synchronizing are usually controlled from the corresponding turbine controllers. Nevertheless, to use those Logical Nodes in the IEC 61850 network, they may be provided with an additional interface to connect to the IEC 61850 electrical control network (dashed red lines in Figure 33). Any Logical Node of the turbines used (e.g. EGTU, ESTU) must be provided by the IEC 61850 Interface Controller. Any interlocking, that would be realized hardwired in a conventional control setup, can be realized by IEC 61850 protocol mappings (e.g. GOOSE), what results in a considerable reduction in cabling and increases flexibility.

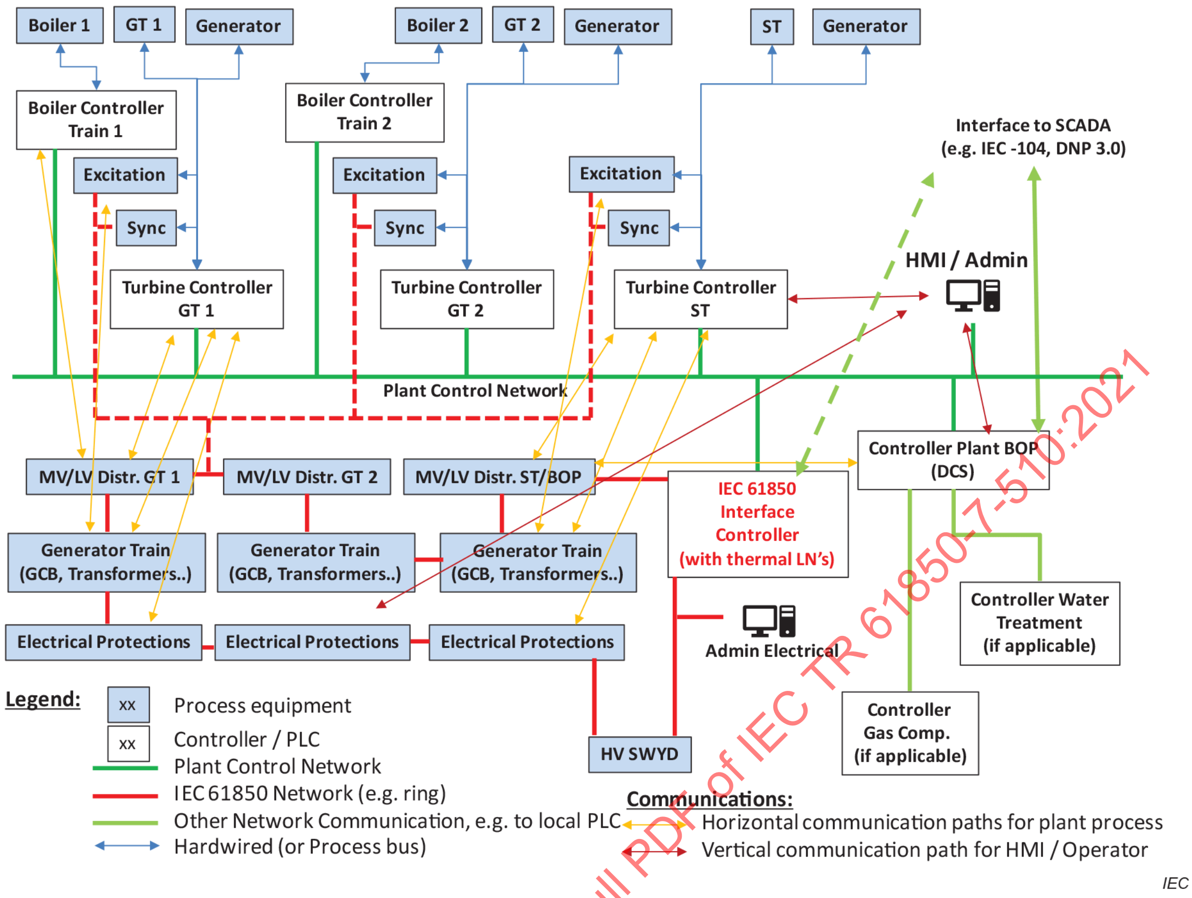


Figure 33 – Typical communication structure with two GTs and one ST, with the use of IEC 61850 interface controller

If the process controllers are equipped with an IEC 61850 interface, they can be connected directly to the IEC 61850 network and communicate to the IED's. In that case the separate IEC 61850 interface controller is not required. Corresponding Logical Nodes (e.g. EGTU, ESTU) are in that case implemented in the corresponding controllers. A typical communication structure is reflected in Figure 34.

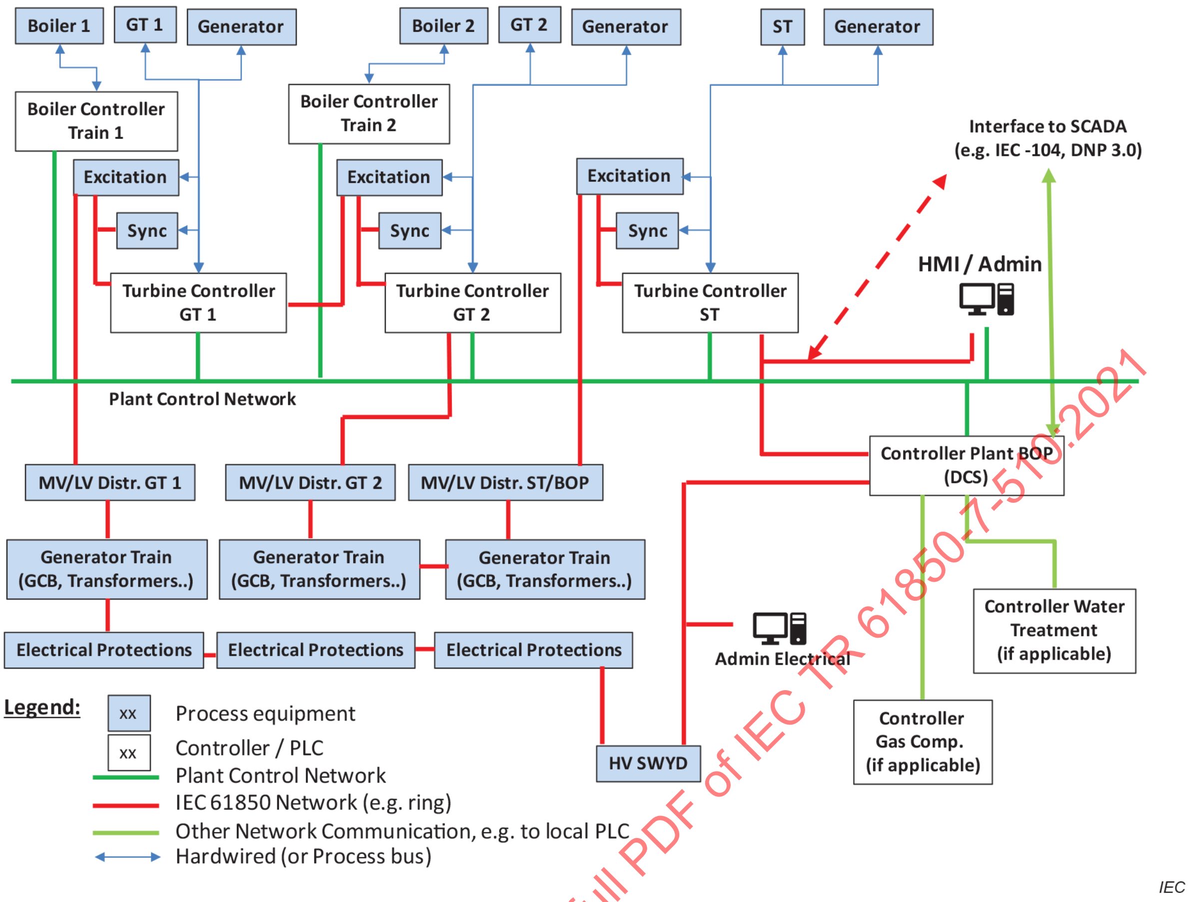


Figure 34 – Typical communication structure with two GTs and one ST, with IEC 61850 interface of process controllers

Sometimes, different process controller types are used for the different turbines and the power plant. This is usually the case in "equipment only"-projects, where the GT(s), ST and boilers are supplied from different manufacturers. In this configuration, the process controller may not be integrated in a common plant network but connected to the plant controller via dedicated buses (e.g. by OPC UA). The signal exchange between the different turbine controllers, or at least a part of it, could be realized by the corresponding Logical Nodes of the turbines implemented in the turbine controllers.

A typical communication structure is shown in Figure 35. In this figure, a common HMI is shown, but often each turbine and the power plant have their dedicated HMIs, and only a limited number of signals is exchanged between the turbine controllers and the power plant controller.

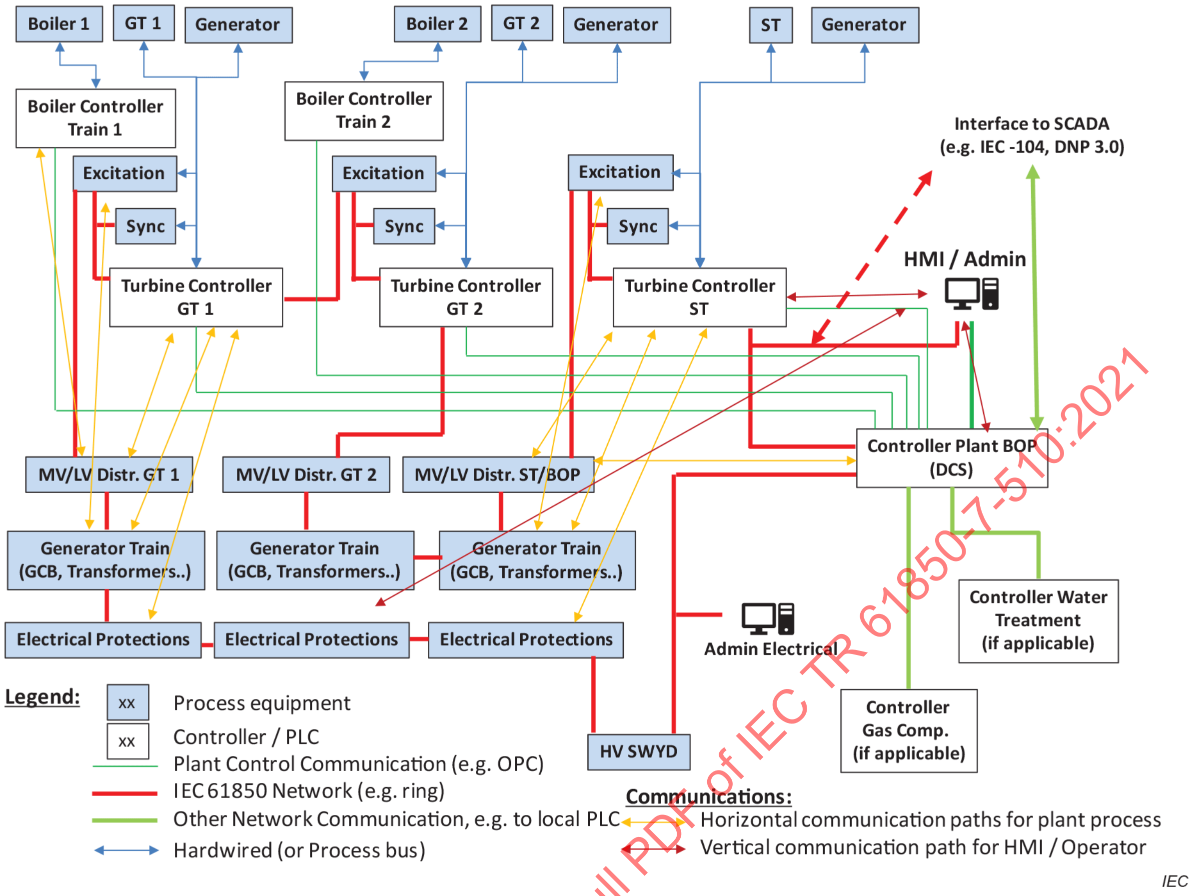


Figure 35 – Typical communication structure with two GTs and one ST, with IEC 61850 interface of process controllers from different manufacturers

A typical example of the communication structure of any type of conventional thermal power plant, e.g. coal or oil fired, with one steam turbine is shown in Figure 36.

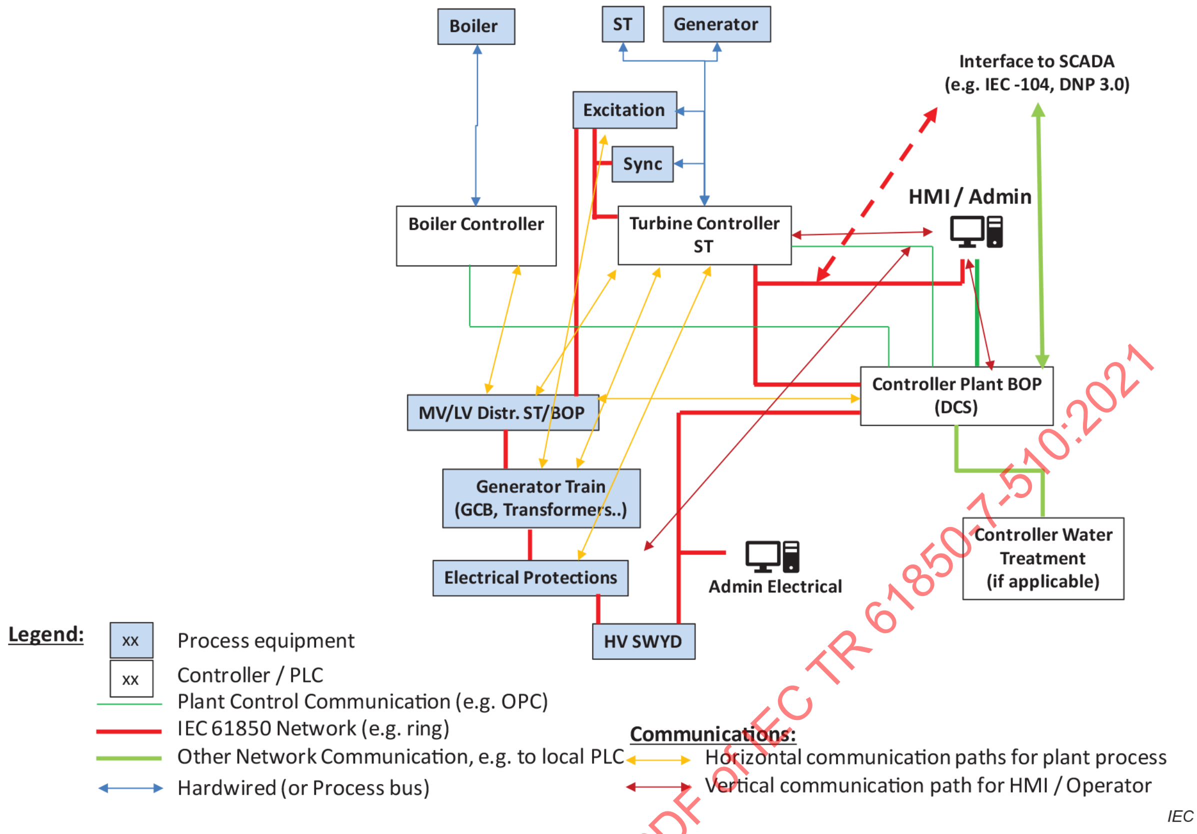


Figure 36 – Typical communication structure with one ST, with IEC 61850 interface of process controllers

9 Modelling of controls

9.1 General

This clause provides descriptions and examples of how to model controls using a combination of RDS and IEC 61850 nomenclature. It primarily provides a way to the map a data model to user defined algorithms.

9.2 Operational modes for hydropower plants

A power plant can be operated in different modes: active power production mode or condenser mode. The generator can be used as a pure synchronous condenser, without any active power production and with the runner spinning in air.

In a pumped storage plant, there is a motor mode for the generator. A generator in a pumped storage plant can also be used for voltage control in a synchronous condenser mode, in this case normally with an empty turbine chamber.

The following steady states are defined for the unit:

- Stopped – Unit is at standstill
- Speed no load, not excited – No field current is applied, no voltage is generated; the generator is running at rated speed but not connected to any external load.
- Speed no load, excited – Field current is applied, and a voltage is generated, the generator is however not connected to any external load, there is no significant stator current.
- Synchronised – The generator is synchronised to an external network. This is the normal status of an operating generator.

- Synchronised in condenser mode – The generator is synchronised. However, it does not primarily produce active power. In condenser mode, it will produce or consume reactive power.
- Island operation mode – The external network has been separated and the power plant shall control the frequency.
- Local supply mode – In case of a larger disturbance of the external network, one or more generators in a power plant can be set at a minimum production to provide power for local supply only. This type of operation is common in thermal power plants to shorten the start-up time once the network is restored but can also be used in hydropower plants for practical reasons.

9.3 Operational modes for thermal power plants

A thermal power plant can be operated in different modes:

- Idle operation – excited, not connected – Field current is applied, and the voltage is generated, the generator is however not connected to the external network. The turbine is in speed control mode to keep the speed at the rated grid frequency.
- Load operation – synchronised – The turbine / generator is synchronised to an external network. This is the normal operation. It can be divided as follows:
  - Load control operation mode – The turbine / generator produces power according to its set-point.
  - Frequency response operation mode – The generator / turbine adapts the load around the load set-point according to the deviation of the frequency to the rated grid frequency
  - Automatic generation control (AGC) – The load set-point for the generator / turbine is given by a superimposed load controller of an entire block, e.g. 2 GT's and 1 ST in a combined cycle power plant.
- House load operation – local supply mode – The power plant has been disconnected from the grid and the generator / turbine produces power for its own auxiliaries. This is to shorten the start-up time once the network is restored. The turbine is in speed control mode.
- Island operation mode – the external network has been separated and the power plant shall control the frequency.

9.4 Fundamental control strategies for hydropower plants

The control of hydropower plants can follow different strategies, depending on the external requirements put on the operation of the system.

Speed control in isolated mode:

The purpose of the speed control basically is to maintain constant frequency. For more detailed description, see IEC 61362.

Active power control:

The active power output control with a separate power controller is applied with the unit connected to the grid. For more detailed description, see IEC 61362.

Reactive power control:

Reactive power control includes voltage and power factor control. This can include synchronous condenser mode without active power output, but also added to active power production.

Water flow control:

In this type of control, the power production is roughly adapted to the water flow that is available at the moment. The rate of flow is controlled while the water level is allowed to vary between high and low alarm levels in the dams. The dams are classified after the time over which the inflow and outflow shall add up (daily, weekly, etc.).

Water level control:

In some locations, there are strict limits imposed on the allowed variation of the water level of the dam. This might be due to maritime shipping or by other environmental requirements. In this case, the upper water level of the dam is the overriding concern; power production is adjusted by the water level control function to provide correct flow to maintain the water level.

Cascade control:

In rivers with more than one power plant, the overall water flow in the river is coordinated between plants to ensure an optimal use of the water. Each individual plant can be operated according to the water level model or the water flow model as best suited, depending on the capacity of the local dam and allowed variation in water levels. The coordination is normally done at dispatch centre level, but power plants often have feed-forward functions that automatically will notify the next plant downstream if there is a sudden change of water flow.

Power plants with more than one generating unit and/or more than one dam gate can be provided with a joint control function that controls the total water flow through the plant as well as the water level control.

9.5 Joint control modelling examples

9.5.1 General

Joint control is an algorithm that enables the automatic regulation control of two or more generation or water regulation units. The goal of joint control is to control a power plant or reservoir in an optimal way. Joint control requires contributing units to be in automatic mode.

Joint control can be used for controlling voltage, reactive power, active power, power factor, water level, water flow. The geometry of contributing units can be varied by using instances of logical node AJCL and HJCL. AJCL is to be used for power control related strategies and HJCL for water related control strategies.

Table 6 shows a summary of functions used for joint control modelling.

Table 6 – Functional breakdown of an RDS component with functions for joint control

SCL:Process structure	SCL:Function name	Functionality
Alpha2.PS1	Fct2	Joint control of reactive power
Alpha2.PS1	Fct32	Joint control of active power
Alpha2.PS1	Fct34	Joint control of water

9.5.2 Joint control of active power

Active Power (watt) joint control logical device uses a specialisation of the AJCL and FSPT logical nodes to interface with joint control algorithm and the external logical devices (Gr..nGov., Gr..nMeas, etc.).

Figure 37 shows an example of an SCL:Function with Logical Nodes used for Joint Control of active power. IHMI represents the operator interface status. The operator interface is subscribed to values from the joint controller algorithm (e.g.: Alpha2.PS1.Fct32/W_AJCL1.UntSpt[1..n] DO). The LN instance numbers have been omitted to simplify the diagram.

How to select which units that will contribute to the joint control effort is a local implementation issue and is not shown in the example.

Which units are contributing to the joint control effort is presented to the operator interface as Alpha2.PS1.Fct32/W_AJCL1.Unt[1..n].stVal=true.

The operator interface can attribute a setpoint value for the joint control total active power production (Alpha2.PS1.Fct32/W_FSPT1.SptVal DO).

In this example information from the bearing vibration monitoring system is used in the control algorithm, but that is not a mandatory solution. The joint control algorithm can be designed in many different ways, using different input data.

The joint control algorithm in this example calculates the redistribution of total watts (Alpha2.PS1.Fct32/W_FSPT1.SptVal) to each of the contributing generating units based on various operating conditions of each unit: opening position (OpnPct/FCSD), bearing vibration (BrgVbr/FCSD), active power (MMXU). The distributed power setpoints to each unit are written to each unit's FSPT1.SptVal.

Each unit setpoint as calculated by the joint control algorithm is presented in Alpha2.PS1.Fct32/W_AJCL1.UntSpt[1..n].mxVal.f. In this case, since the algorithm is local, the published data attribute is used as status-only; Alpha2.PS1.Fct32/W_AJCL1.UntSpt[1..n].ctlModel=0 (status-only mode).

Table 7 describes the setpoint data flow in the example.

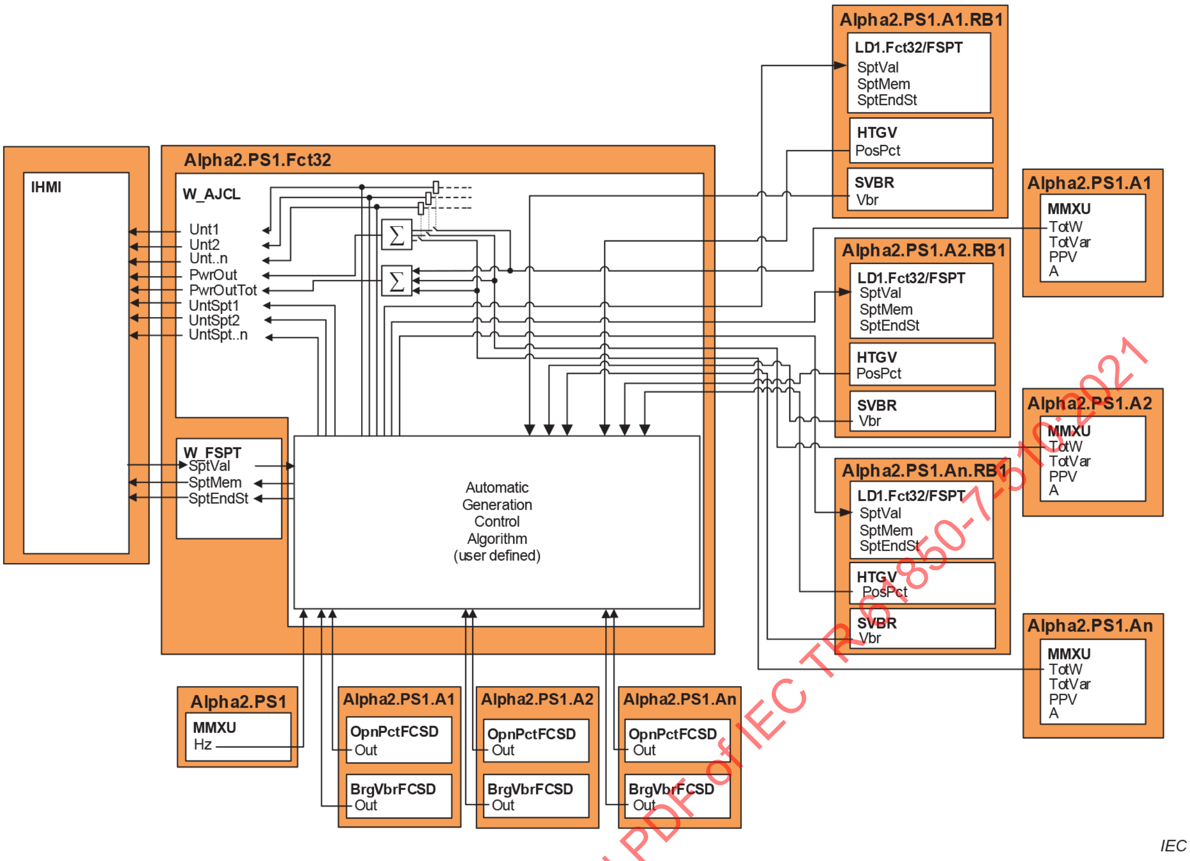


Figure 37 – Joint Control of active power

Table 7 – Joint Control active power setpoints data flow

Alpha2.PS1.Fct32... (Joint control of active power)			Alpha2.PS1...
Plant Setpoint	**Local Joint Control Algorithm**	**Calculated Setpoint to Units**	**Setpoint in Unit Controllers**
/W_FSPT1.SptVal	Automatic Generation Control (including load sharing between units selected for joint control)	/W_AJCL1.UntSpt1	A1.RB1.LD1.Fct32/FSPT.SptVal
		/W_AJCL1.UntSpt2	A2.RB1.LD1.Fct32/FSPT.SptVal
		/W_AJCL1.UntSptn	An.RB1.LD1.Fct32/FSPT.SptVal

9.5.3 Joint Control of Reactive Power

Var joint control logical device, as shown in Figure 38, uses a specialisation of the AJCL and FSPT logical nodes to interface with joint control algorithm and the external processes (Alpha2.PS1.A[1..n].RA1.LD1.Fct2.FSPT1, Alpha2.PS1.A[1..n].RA1.LD1.Fct2.MMXU1, etc.). The LN instance numbers have been omitted to simplify the diagram.

IHMI represents the operator interface status. The operator interface is subscribed to values from the joint controller algorithm (e.g.: Alpha2.PS1.Fct2/VAR_AJCL1.UntSpt[1..n] DO).

How to select which units that will contribute to the joint control effort is a local implementation issue and is not shown in the example.

Which units are contributing to the joint control effort is presented to the operator interface as Alpha2.PS1.Fct32/W_AJCL1.Unt[1..n].stVal=true.

The operator interface can attribute a setpoint value for the joint control total reactive power production (Alpha2.PS1.Fct2/VAr_FSPT1.SptVal DO).

The operator interface can attribute a joint control setpoint value for either:

- Reactive Power (Alpha2.PS1.Fct2/VAr_FSPT1.setVal DO)
- Power Factor (Alpha2.PS1.Fct2/PF_FSPT1.setVal DO)
- Voltage (Alpha2.PS1.Fct2/Volt_FSPT1.setVal DO).

The joint control algorithm in this example calculates the redistribution of total VARs (Alpha2.PS1.Fct2/VAr_FSPT1.SptVal) to the contributing generating units. The distributed power setpoints to each unit are written to each unit's FSPT.SptVal.

Each unit setpoint as calculated by the joint control algorithm is presented in Alpha2.PS1.Fct2/VAr_AJCL1.UntSpt[1..n].mxVal.f. In this case, since the algorithm is local, the published data attribute is used as status-only; Alpha2.PS1.Fct2/VAr_AJCL1.UntSpt[1..n].ctlModel=0 (status-only mode).

The corresponding semantics apply for joint control of Power Factor (PF_) and Voltage (Volt_).

Table 8 describes the setpoint data flow in the example.

In this example three instances of LN FSPT are used in parallel for the Plant setpoint of reactive power, power factor and voltage respectively. How to select which control mode (setpoint) that is active is a local issue.

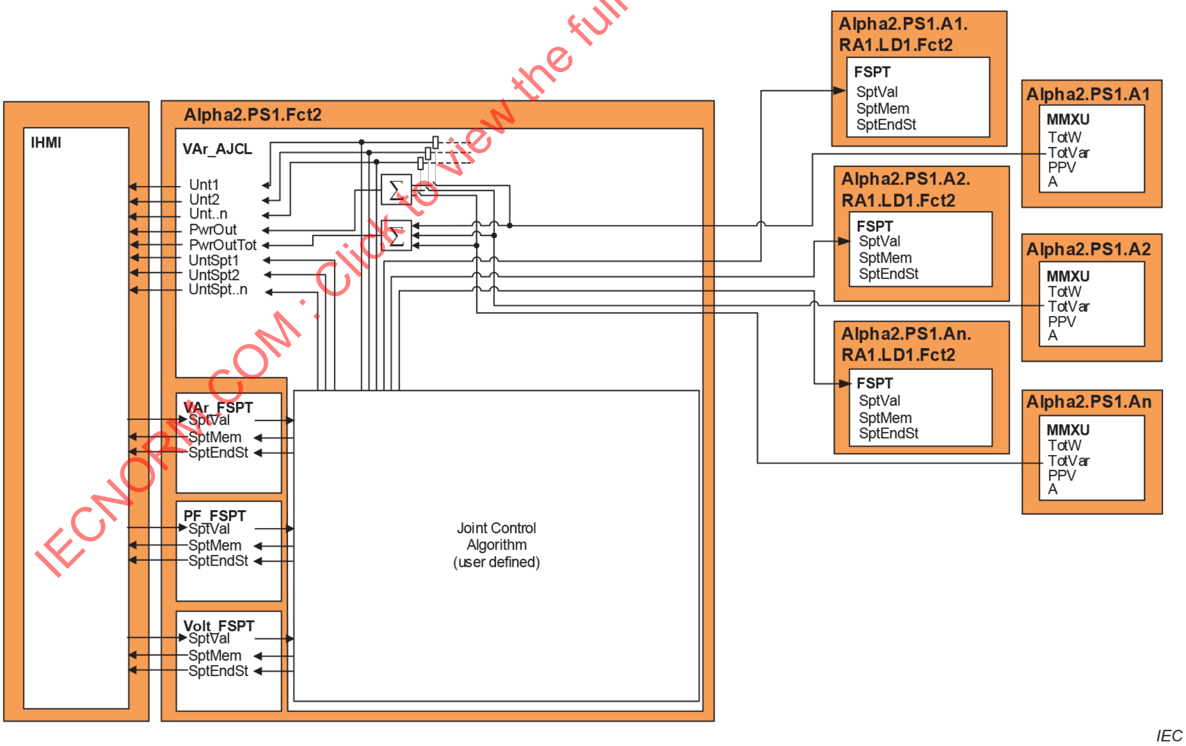


Figure 38 – Joint control of reactive power (SCL:Function:Fct2)

Table 8 – Joint Control reactive power setpoints data flow

Alpha2.PS1.Fct2... (Joint control of reactive power)			Alpha2.PS1...
Plant Setpoint	**Local Joint Control Algorithm**	**Calculated Setpoints to Units**	**Setpoint in Unit Controllers**
/VAr_FSPT1.SptVal	Automatic Generation Control (including load sharing between units selected for joint control)	/VAr_AJCL1.UntSpt1	A1.RA1.LD1.Fct2/FSPT.SptVal
		/VAr_AJCL1.UntSpt2	A2.RA1.LD1.Fct2/FSPT.SptVal
		/VAr_AJCL1.UntSptn	An.RA1.LD1.Fct2/FSPT.SptVal

9.5.4 Joint Control of Water

Water joint control logical device uses a specialisation of the HJCL and FSPT logical nodes to interface with joint control algorithm and the external processes (from Alpha2.PS1.A[1..n].RB1.LD1).

Figure 39 provides an example of how water joint control can be implemented. The joint controller can set the power setpoint for a governor and flow setpoint for a governor or gate.

IHMI represents the operator interface status. The operator interface is subscribed to values from the joint controller algorithm (e.g.: Alpha2.PS1.Fct34/HJCL1.Gte..n DO and calculated flow). The LN instance numbers have been omitted to simplify the diagram.

How to select which gates and units that will contribute to the joint control effort is a local implementation issue and is not shown in the example.

Which spillway gates are contributing to the joint control effort is presented to the operator interface as Alpha2.PS1.Fct34/HJCL1.Gte[1..n].stVal=true. How to present which units that are contributing to the joint control effort is a local implementation issue, but one solution is to present the information as dedicated HJCL1.Gte[1…n] DO instances.

The operator interface can attribute a setpoint value for total flow Alpha2.PS1.Fct34/Flw_FSPT1.setVal DO, or reservoir level Alpha2.PS1.Fct34/Lvl_FSPT1.setVal DO.

The joint control algorithm in this example calculates a redistribution of total flow expressed as an active power setpoint to each of the contributing units and an opening setpoint to each of the contributing gates. The distributed setpoint to each unit or gate is written to their respective FSPT.SptVal.

Each unit or gate setpoint as calculated by the joint control algorithm is presented in Alpha2.PS1.Fct34/HJCL1.ClcPwrSpt[1..n].mxVal.f and Alpha2.PS1.Fct34/HJCL1.ClcGteSpt[1…n].mxVal.f respectively.

In this case, since the algorithm is local, the published data attribute is used as status-only; Alpha2.PS1.Fct34/HJCL1.ClcPwrSpt[1..n].ctlModel=0 and Alpha2.PS1.Fct34/HJCL1.ClcGteSpt[1..n].ctlModel=0 (status-only mode).

Table 9 describes the setpoint data flow in the example.

In this example two instances of LN FSPT are used in parallel for the Plant setpoint of water level and water flow respectively. How to select which control mode (setpoint) that is active is a local issue.

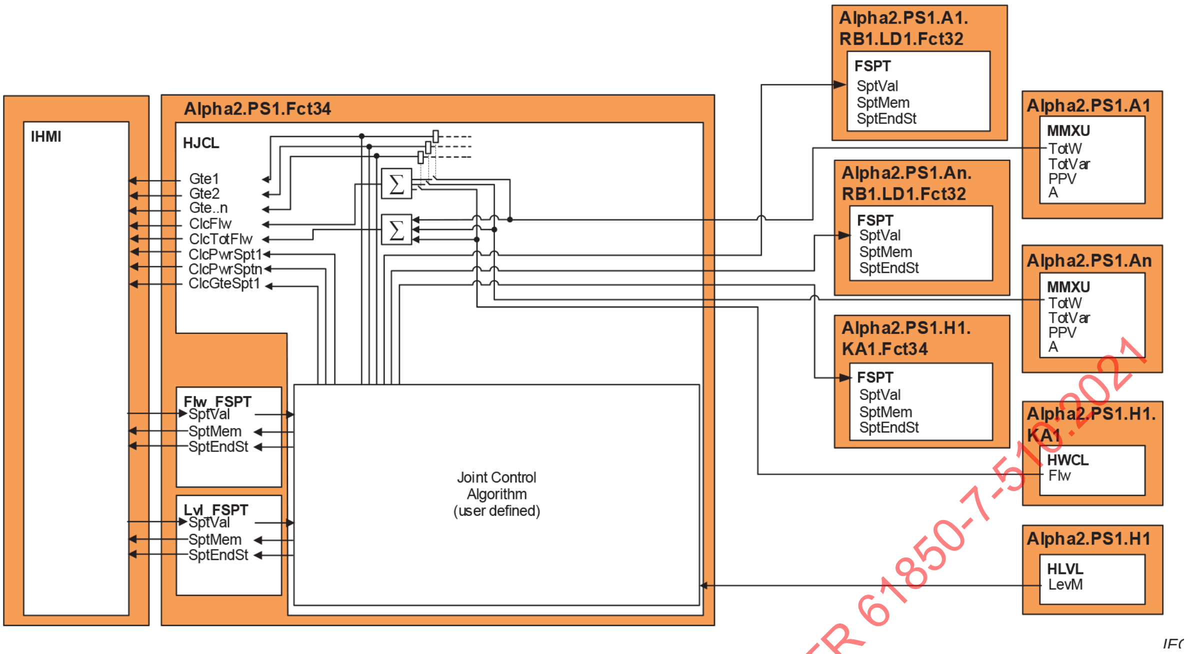


Figure 39 – Example of joint control of water

Table 9 – Joint Control flow setpoints data flow

Alpha2.PS1.Fct34... (Joint control of flow or level)			**Alpha2.PS1...**
Plant Setpoint	**Local Joint Control Algorithm**	**Calculated Setpoints to Units and Gate**	**Setpoint in Unit Controllers and Spillway Gate respectively**
/Flw_FSPT1.SptVal	Automatic Flow Control (including sharing of flow between units and gate(s) selected for joint control)	/HJCL1.ClcPwrSpt1	A1.RB1.LD1.Fct32/FSPT.SptVal
		/HJCL1.ClcPwrSptn	An.RB1.LD1.Fct32/FSPT.SptVal
/Lvl_FSPT1.SptVal	Automatic Level Control (including sharing of flow between units and gate(s) selected for joint control)	/HJCL1.ClcGteSpt1	H1.KA1.Fct34/FSPT.SptVal

9.6 Scheduling Example

IEC 61850 provides a means to map a schedule. This mean can be summarized into two logical nodes: FSCC and FSCH. The FSCC logical node is the schedule controller and FSCH is the schedule.

The FSCC logical node provides a list of schedules that can be controlled and an indication of which schedule is active.

The FSCH consists of an indication of the duration of the schedule interval and a series of entries, representing the "absolute" expected output of the schedule at each consecutive schedule interval time, the first value being attached to the first time interval. It also provides a start time for the schedule which can be based on a weekday occurrence.

In the following example, the Alpha2.PS1.A1.Fct40 function provides an output for the value in the active power joint controller function Alpha2.PS1.A1.Fct32 data object FSPT.SptMem. The output value is derived from the active schedule's active period. A time selector algorithm is required to detect which schedule is active based on the weekday occurrence inscribed in the start time data object.

Figure 40 has one 24h Schedule for active power production per Week Day, and a Schedule Controller to handle the Schedules. The number of Schedules and the duration of each Schedule is a local implementation issue. In the referred figure, FSCH and FSCC are according to the definitions in IEC 61850-7-4:2010+AMD1:2020.

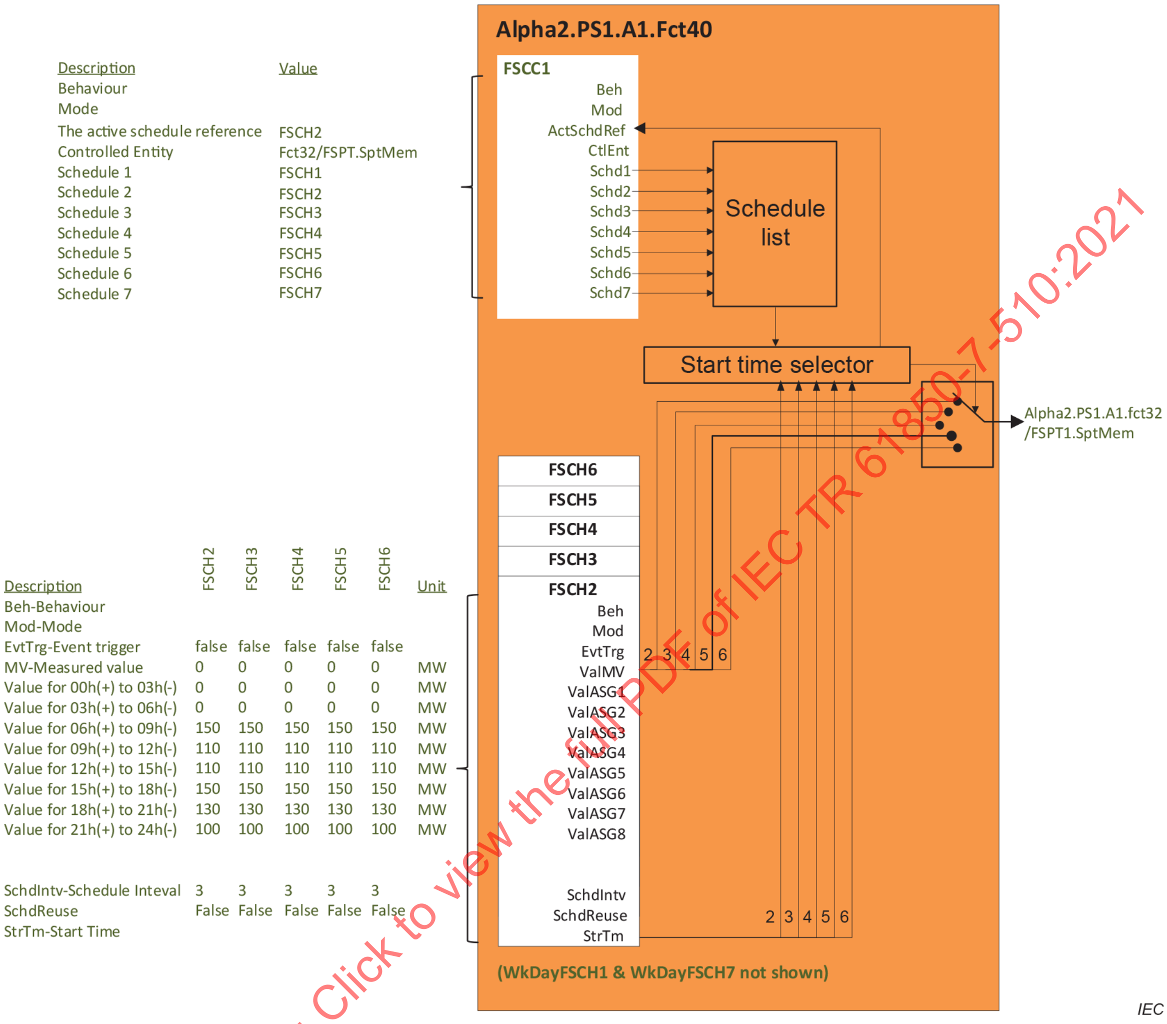


Figure 40 – An example of scheduling of active power output

9.7 Example of application for an excitation system

9.7.1 General

Figure 41 shows an example of functional blocks of an excitation system based on an IED representation, with typical logical nodes that can be used within each functional element.

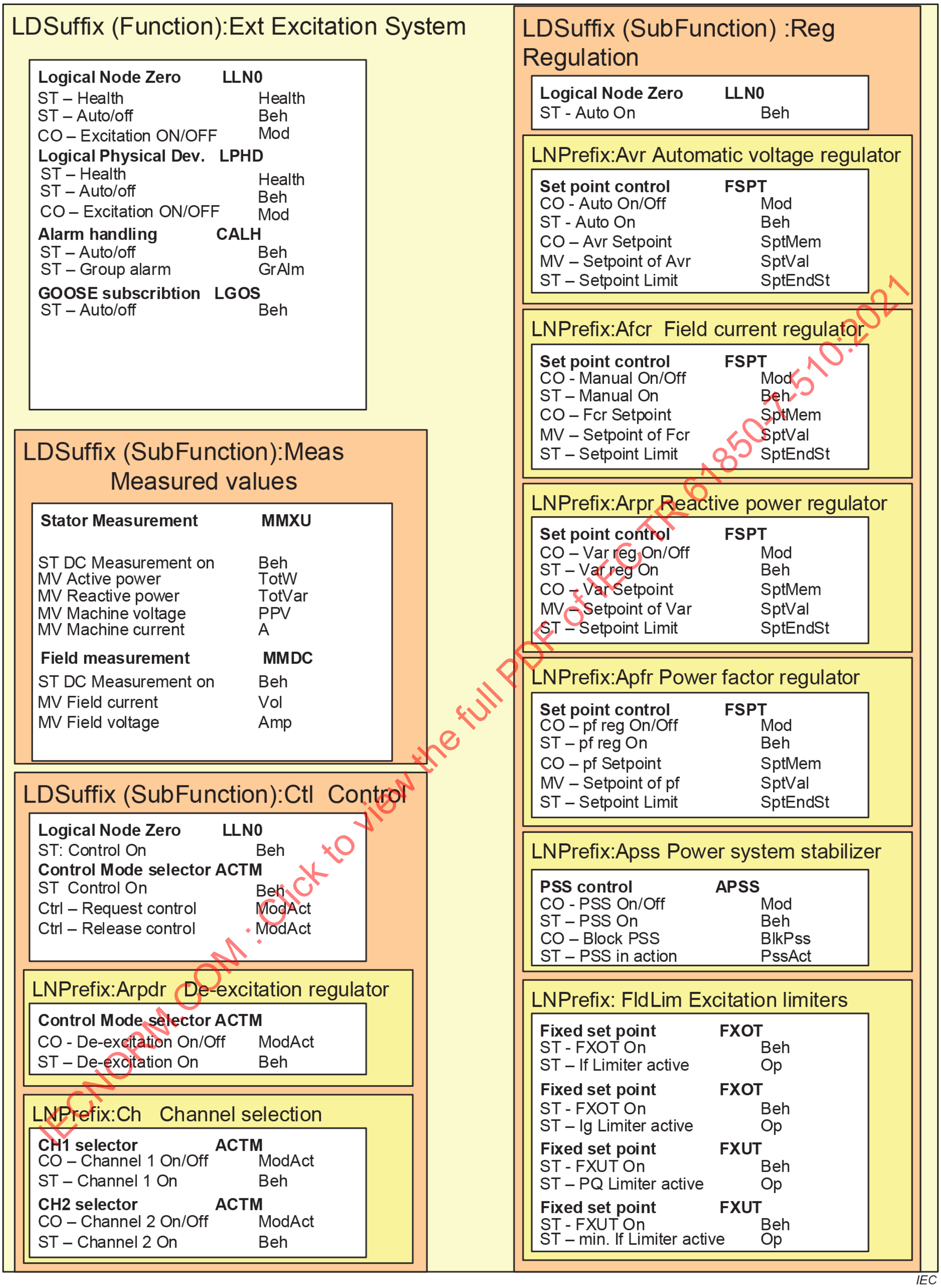


Figure 41 – Examples of logical nodes used in an excitation system

For practical purposes, the excitation system will be divided in a number of Process and Functions that can be addressed and handled separately using an RDS-PS designation system.

The division into process blocks (Control, Field breaker, Converter and Protection), as well as sub functions (KV, Field, PF, VAR regulators) as represented in Figure 41 is only informative and may be interpreted in different ways.

Figure 42 gives a non-exhaustive example of a functional break down for a larger excitation system. The SCL:Function name is based on the concatenation of RDS component system with an IEEE C37.2 function number (example: LA90: regulator).

In Table 10 an example of the SCL element structure of an excitation system. The SCL:Function name is used to show how it may be used to show the type of algorithm used in the implementation, based on an assumed list of definitions for algorithms pf type "Fct 1 ... n". The meaning of the SCL:Function names is a local implementation issue.

Within the Hydro Power industry, the IEEE C37.2 numbering system is often used to denote types of functions. A power plant operator may therefore decide to create SCL:Function names based on that numbering system. If the SCL:Function name is used to hold information the meaning of the SCL:Function name should be defined locally.

Table 10 – Functional breakdown of a Process child RDS component with functions

SCL:Process structure	SCL:Function name	Functionality
Alpha2.PS1.A1.LD1		Excitation system
Alpha2.PS1.A1.LD1		Regulators
Alpha2.PS1.A1.LD1	Fct1	Automatic voltage regulator
Alpha2.PS1.A1.LD1	Fct2	Reactive power regulator
Alpha2.PS1.A1.LD1	Fct3	Power factor regulator
Alpha2.PS1.A1.LD1	Fct4	Field current regulator
Alpha2.PS1.A1.LD1	Fct5	PSS
Alpha2.PS1.A1.LD1	Fct6	Automatic control mode
Alpha2.PS1.A1.LD1	Fct7	Manual control mode
Alpha2.PS1.A1.LD1	Fct8	Boosting function
Alpha2.PS1.A1.LD1	Fct9	Excitation limiter
Alpha2.PS1.A1.LD1	Fct10	Flux limiter (volt /hertz limiter)
Alpha2.PS1.A1.LD1	Fct11	Stator current limiter
Alpha2.PS1.A1.LD1	Fct12	De-excitation regulator
Alpha2.PS1.A1.LD1	Fct13	Start excitation (field flash control)
Alpha2.PS1.A1.LD1	Fct14	Field breaker control
Alpha2.PS1.A1.LD1	Fct15	Thyristor control
Alpha2.PS1.A1.LD1	Fct16	Converter protections
Alpha2.PS1.A1.LD1	Fct17	Channel selector
Alpha2.PS1.A1.LD1	Fct18	Excitation Control mode

Figure 42 shows an example of the relationship between an Excitation (with Logical Nodes typical for the Excitation system), and the operator HMI, the internal regulator functions and the external thyristor control respectively.

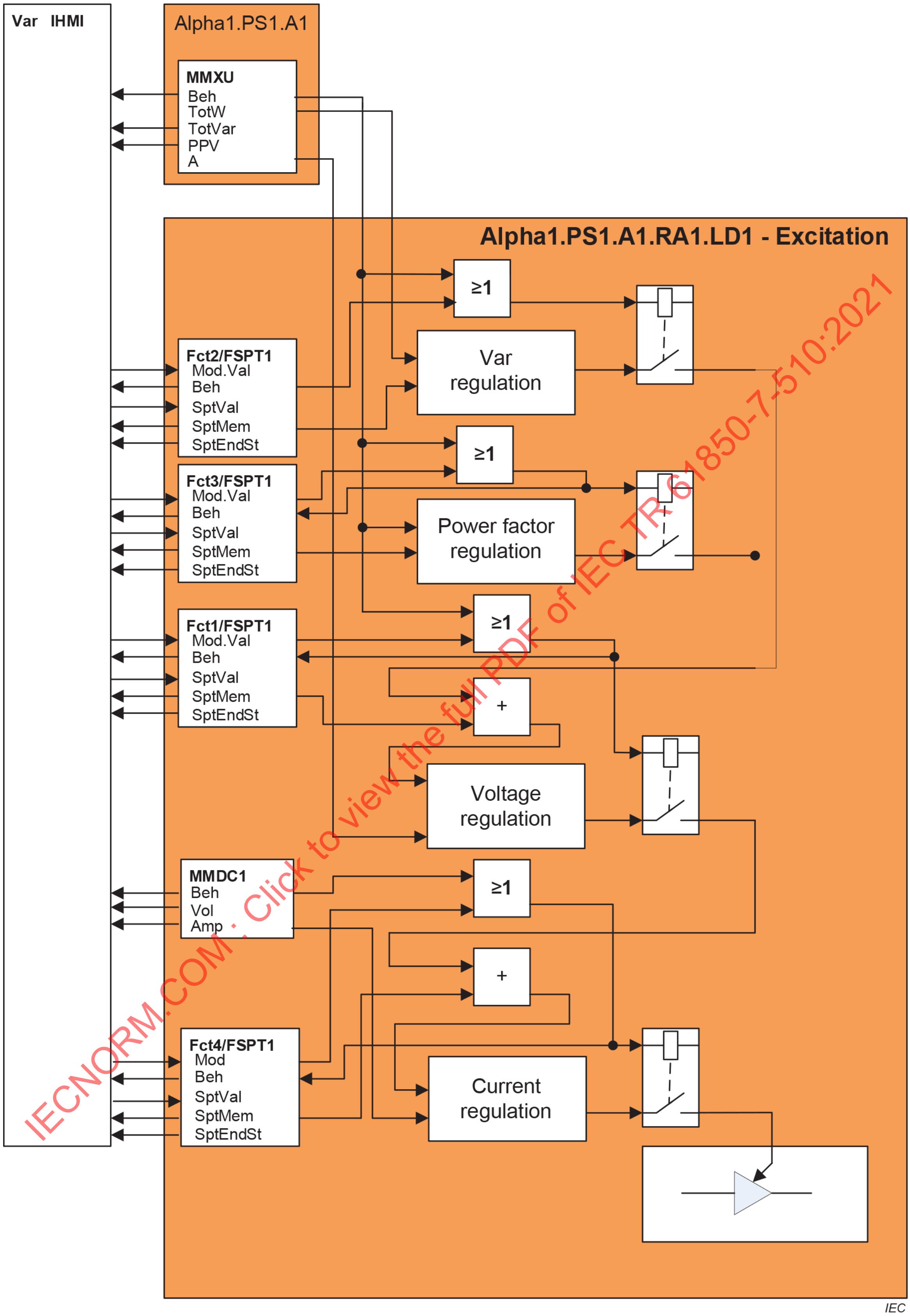


Figure 42 – Example of an excitation a functional breakdown

Figure 43 is an example for the functional structure of the regulation part of an excitation system. The coloured parts represent logical devices.

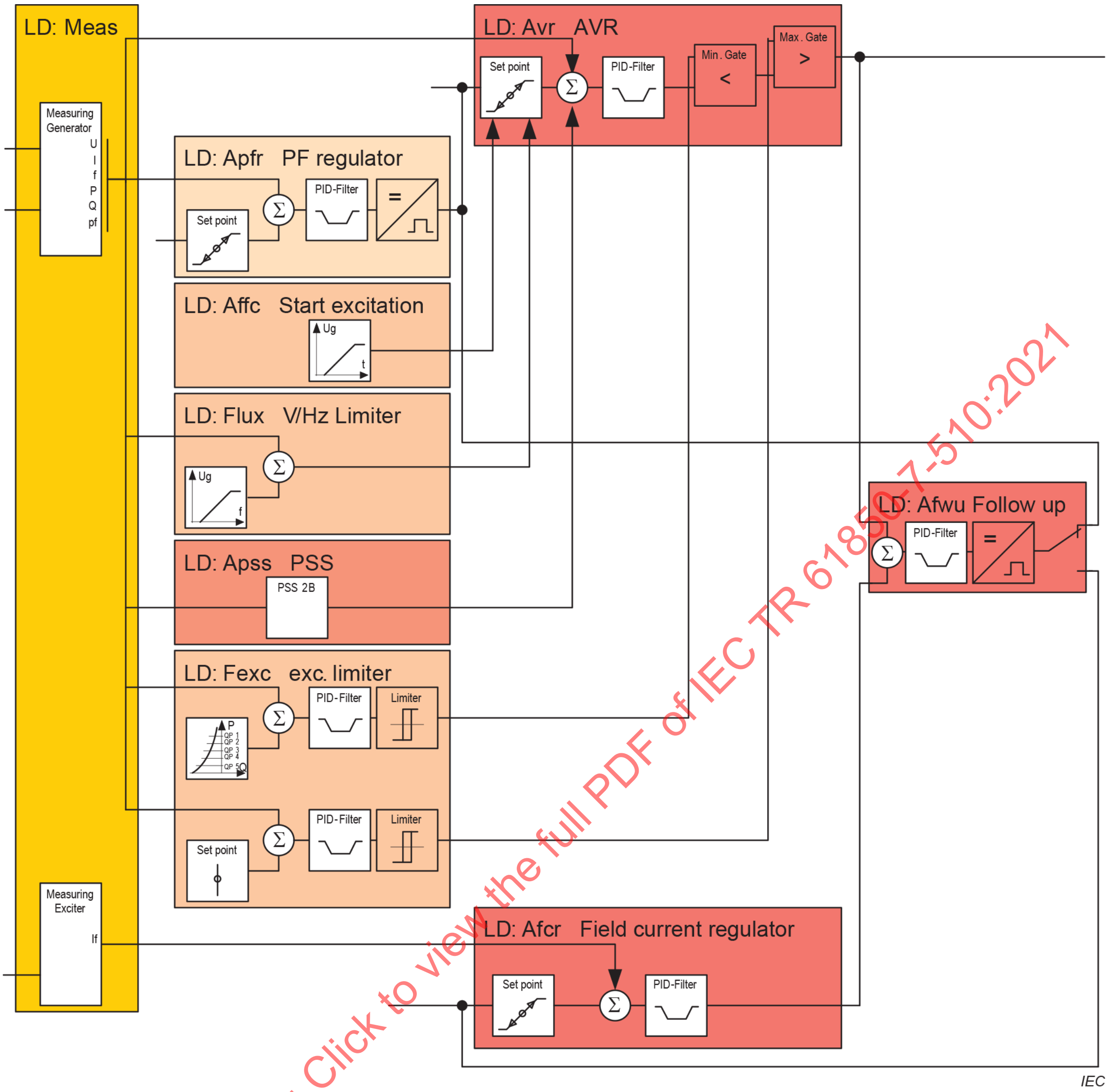


Figure 43 – Example of logical devices of the regulation part of an excitation system

NOTE IEC 61850 generally allows reading and writing parameter settings using the IEC 61850 interface. From a practical point of view, writing parameters using the IEC 61850 interface will not be allowed. Therefore, the following examples show settings as read only signals.

9.7.2 Voltage regulation example

Figure 44 shows an automatic voltage regulator.

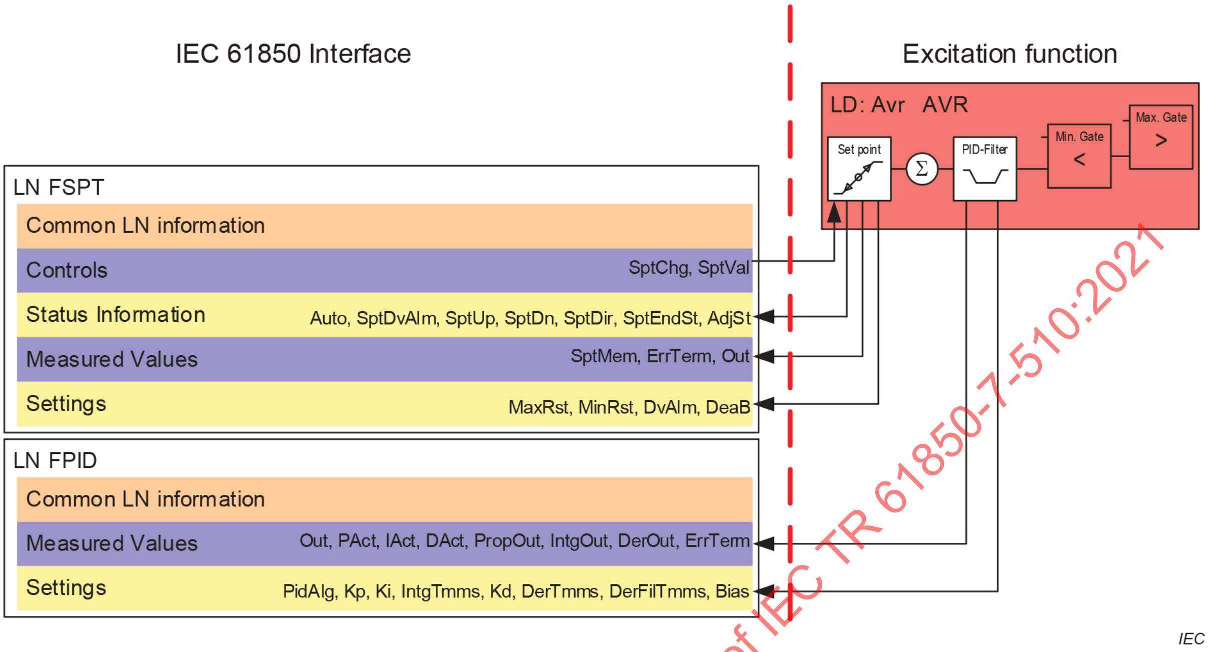


Figure 44 – AVR basic regulator

Figure 45 shows a power factor regulator.

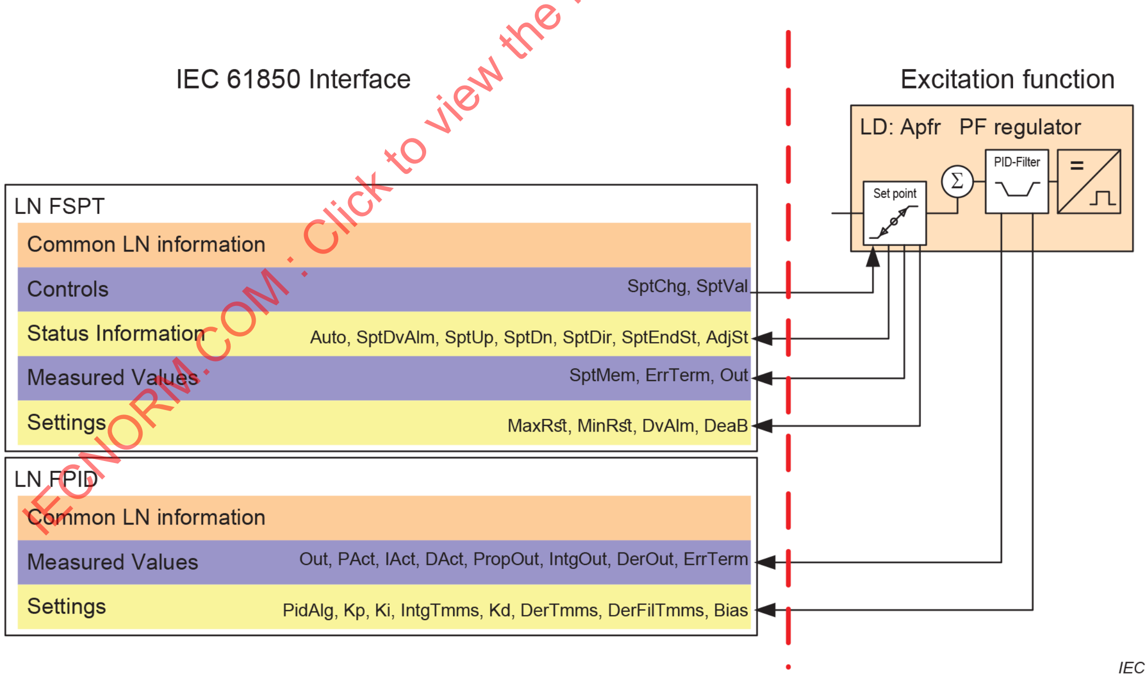


Figure 45 – Superimposed regulators, power factor regulator

Figure 46 shows an over-excitation limiter.

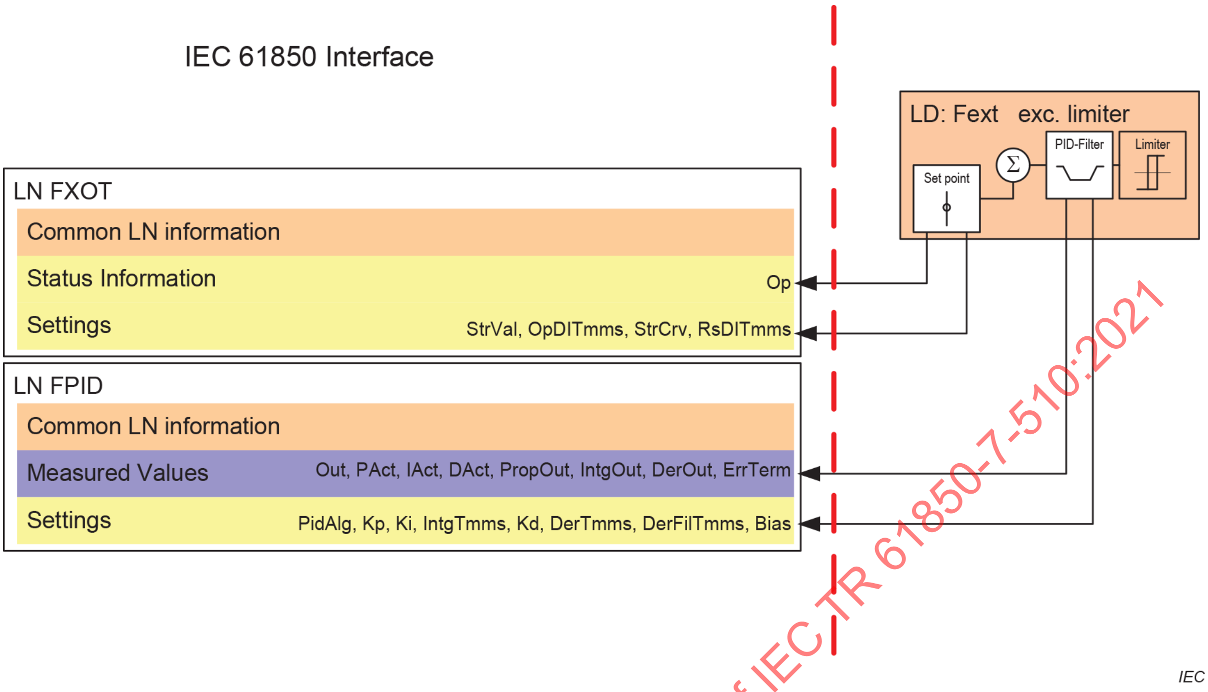


Figure 46 – Superimposed regulators, over-excitation limiter

Figure 47 shows an under-excitation limiter.

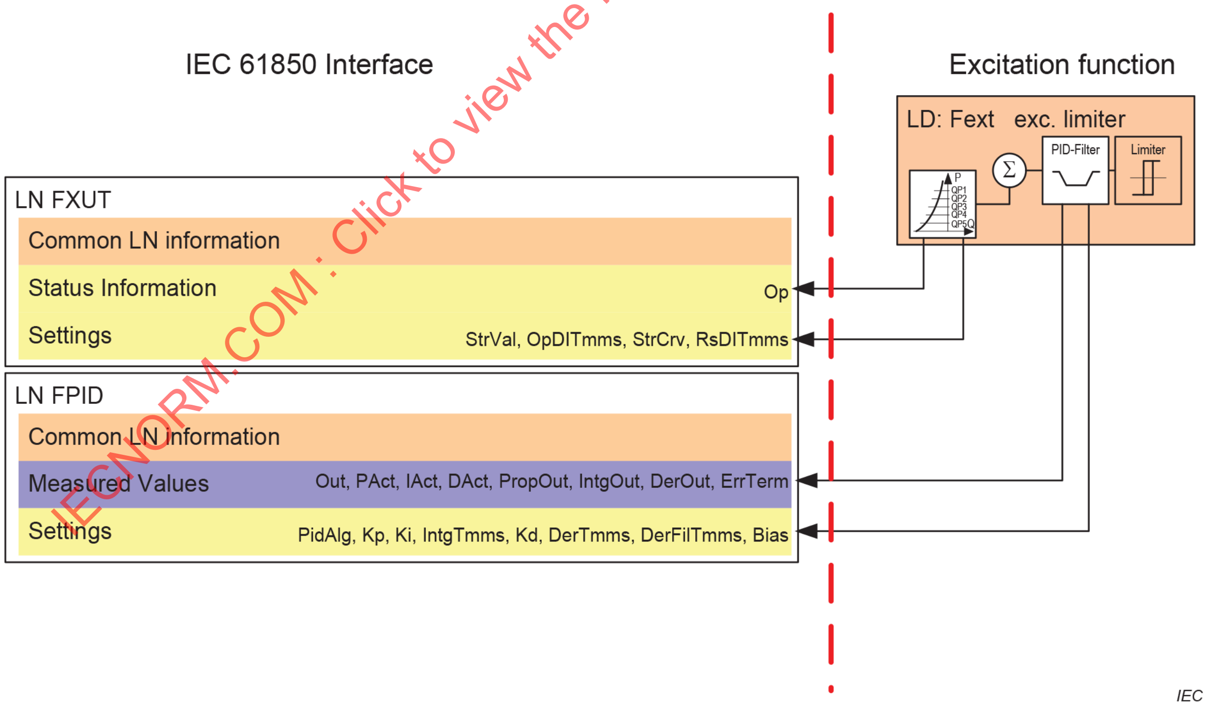


Figure 47 – Superimposed regulators, under-excitation limiter

Figure 48 shows a follow up.

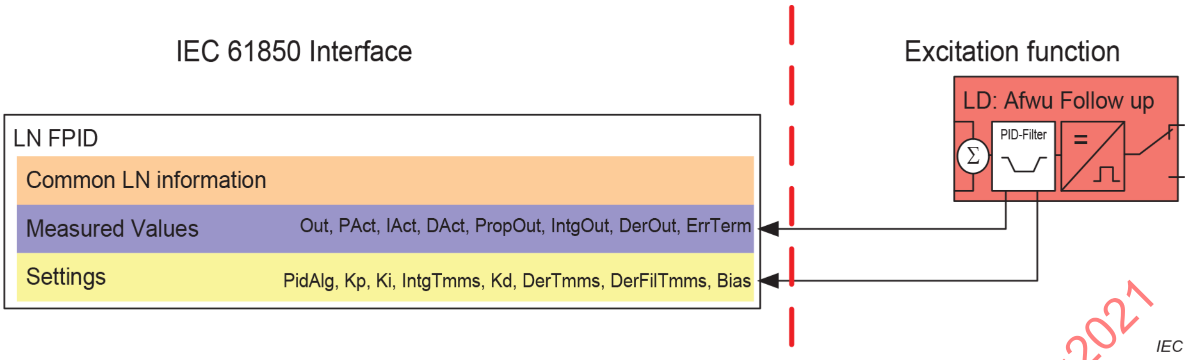


Figure 48 – Superimposed regulators, follow up

9.7.3 PSS example

Figure 49 shows a power system stabilizer function.

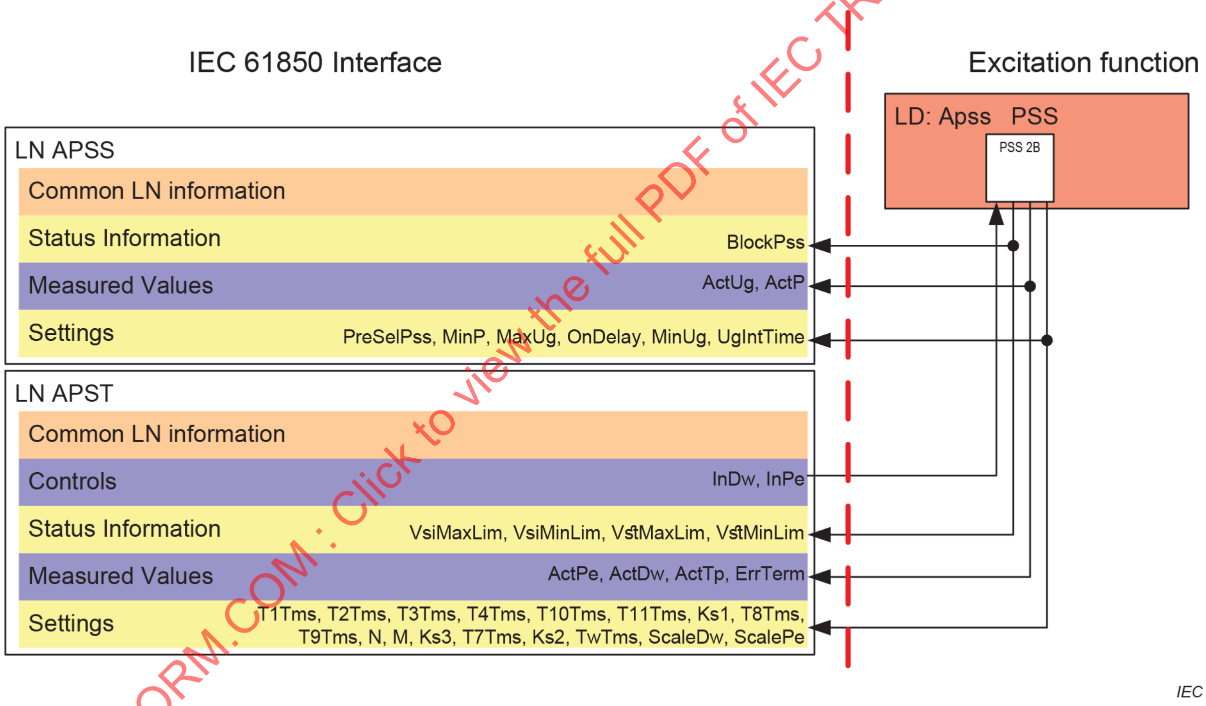


Figure 49 – Power system stabilizer function

The standard model of IEC 61850 part -7-410 includes three logical nodes that can be combined to represent a PSS function: APSS that provides general information, APST that represents an IEEE 2A/B filter and APSF that represents a IEEE 4A filter. In Figure 49, only the 2A/B filter is shown.

9.8 Example of application for a turbine governor system

9.8.1 General

The example is based on a minimum signal list on which everyone can agree, consisting of data points that are absolutely necessary for operating a turbine governor. In addition, consideration has been made to enable extension to other projects:

- no fixed definition of used controllers,
- clear separation of the control algorithm from the operation conditions and the data acquisition.

The following assumptions have been made.

- Francis turbine with one single actuator (only one position indication). In case of individual wicket gate control, individual position can be added, and threshold associated to the actuators are being managed by internal data of the turbine governor;
- single turbine governor (no redundant signal or system);
- communication with an HMI (Human-Machine Interface) is not included.

9.8.2 Signal hierarchy

Figure 50 shows a signal hierarchy.

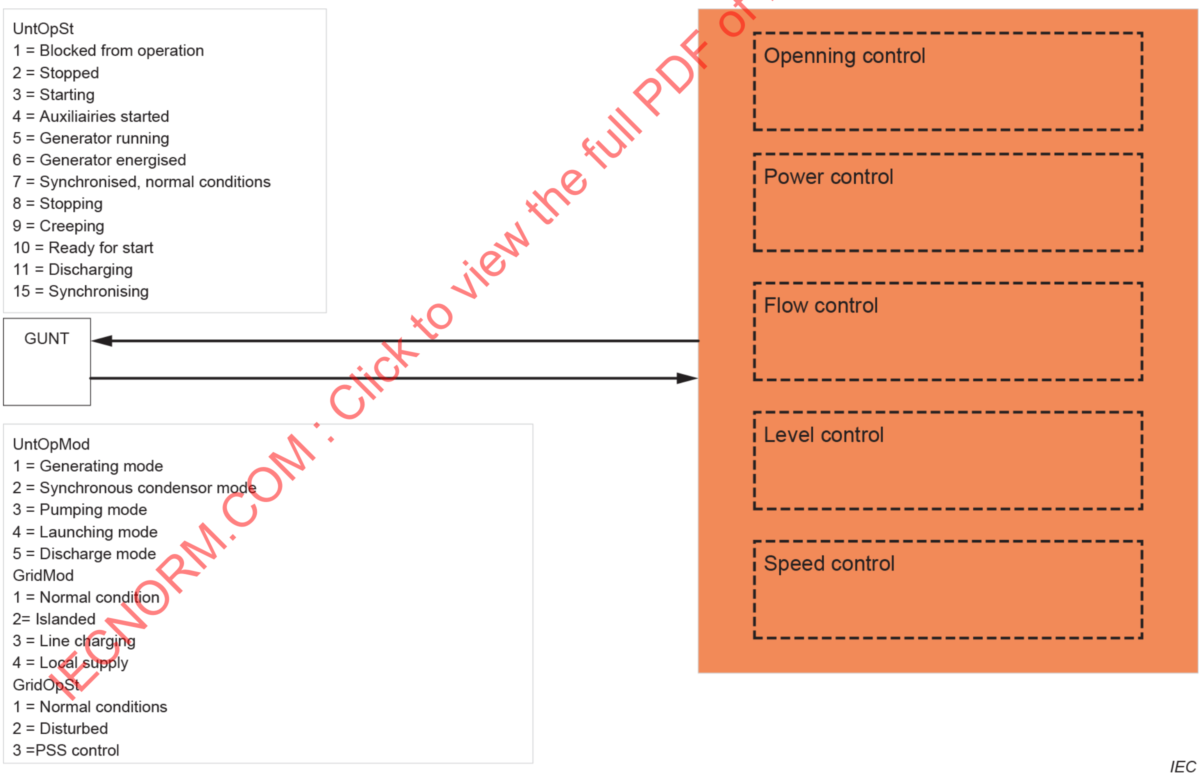


Figure 50 – Signal hierarchy

The GUNT LN communicates the actual state of the grid and the desired operation mode to the turbine control. In dependency on these states the actual governor (power, opening, flow, level or speed) and the correspondent parameter set are selected.

9.8.3 Basic overview

Figure 51 shows the typical functional blocks of a turbine governing system, from the communication point of view, and by considering the example of a Francis turbine.

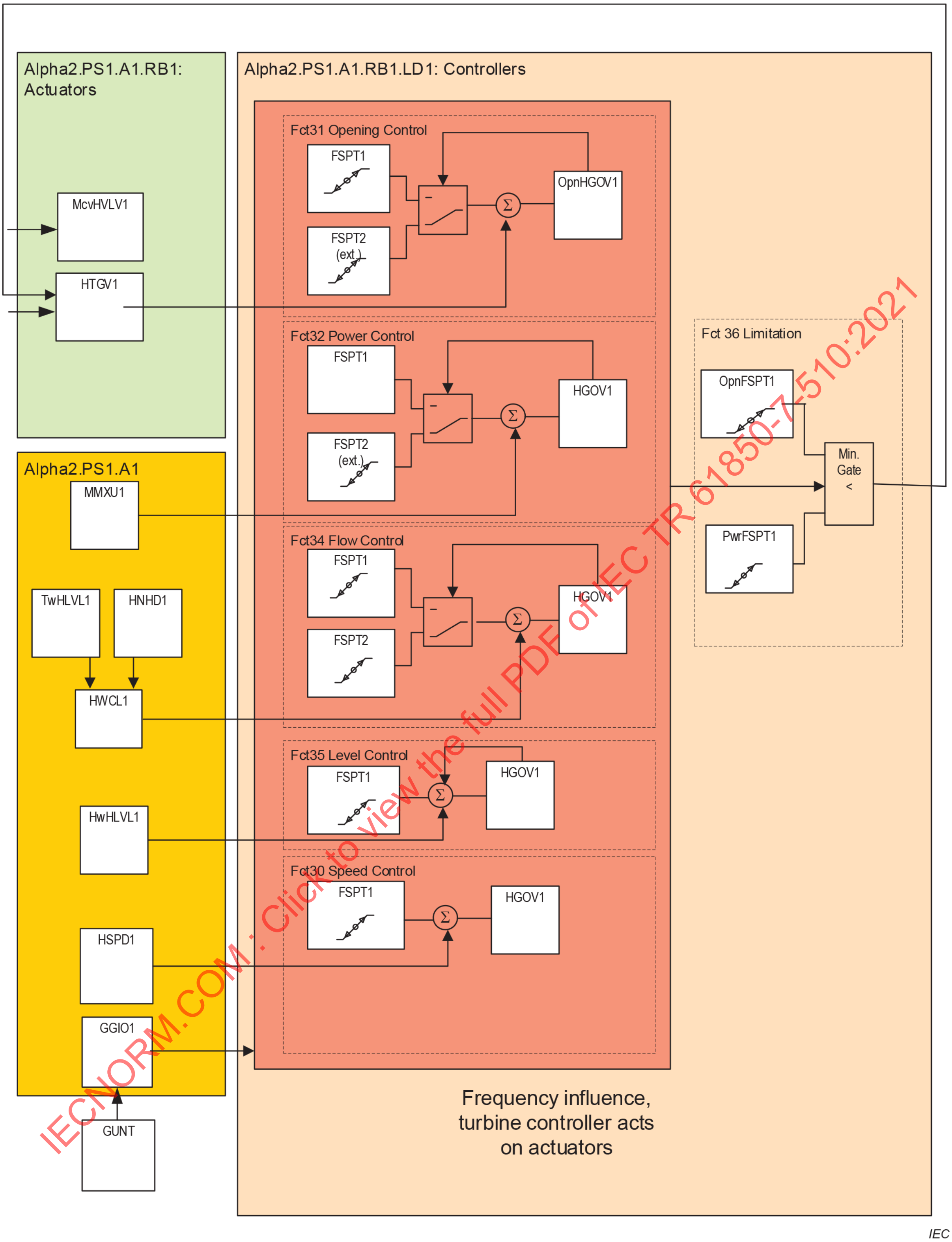


Figure 51 – Use of Logical Node HGOV with RDS-PS

As mentioned in 9.8.2, the general structure is based on the allocation of the different signals to be exchanged, among 3 main logical devices, which are interacting:

1) Logical Device "Actuators": it mainly concerns the position of the wicket gate, and the corresponding failures of the positioning circuit.

2) Logical Device "Turbine information": it mainly concerns the different operating modes of the turbine (e.g. start / stop, synchronous condenser mode …) and the different hydraulic variables (e.g. water levels and flow, penstock pressure …).
3) Logical Device "Controllers": it mainly includes a large block made up of a combination of different single controllers (speed controller, power controller), which are interacting; the output signal of this combination is then limited by the block "Limitation", and is finally acting as a command signal for the actuators.

Table 11 gives a non-exhaustive example of how SCL:Process and SCL:Function names can be defined for a turbine control system.

The logical nodes are represented by white boxes in Figure 51.

Table 11 – Functional breakdown of a Process child RDS component with functions

SCL:Process parent	SCL:Function name	Functionality
Alpha2.PS1.A1.RB1		Actuators
Alpha2.PS1.A1.RB1	Fcn30	Speed regulator
Alpha2.PS1.A1.RB1	Fcn31	Opening
Alpha2.PS1.A1.RB1	Fcn32	MW
Alpha2.PS1.A1.RB1	Fcn33	Speed
Alpha2.PS1.A1.RB1	Fcn34	Water Flow
Alpha2.PS1.A1.RB1	Fcn36	Governor Limiters

As in the previous example above, in this example it is also assumed that there is a local definition of the used SCL:Function names to denote for example the type of algorithms used. Naming of SCL:Function names is a local implementation issue.

Concerning the controllers block in Figure 51 (in red colour), and according to IEC 61362, some controllers can be made inoperative: for example, the level controller will be made inoperative when the water level control is not required, or operated by the joint power plant control. Furthermore, the global control structure using a combination of such controllers can be either a series or a parallel structure: for example, the power controller and the speed controller, linked by the frequency power droop.

Finally, and generally speaking, the logical node HGOV, as shown in Figure 51, activates all these controllers.

Figure 51 shows the use of Logical Node HGOV.

9.8.4 Detailed description of used IED structure

The following is a detailed block description and ideas behind the used LNs.

Use of quality information (xxx.q) for indication of fault in the device, control loop errors are handled separately.

Some controllable settings are only set internally.

In the three Logical Devices, the Logical Nodes model the data exchange via IEC 61850 to any communication partner. Only the information, which is relevant for external devices, is available at the Logical Nodes and can be used to monitor and control the governor.

Any kind of information that describes the state (quality), like "good", "invalid", "questionable", etc., is mapped to the attribute of the information it belongs to. For example, the information "failure signal" is mapped to the DataObjectName "q" of the type "Quality" in the used CommonDataClass. This is done only for the CDC's MV and APC. All analogue information in these CDC's are based on floating point values and not on integer values. Please see the detailed information on the quality information in IEC 61850-7-3.

The Logical Nodes are identified and selected by their main functionality. To describe the advised function of a LN, the prefix is used. The use of general LNs like GGIO should be limited to cases where other defined LN classes are not suitable . The general LN classes represent information that has not been assigned a dedicated LN class.

At the Logical Device "Turbine Information" the prefix "Unt" or "Pwr" is used to point out that the LN is unit related. To describe a plant related LN, the prefix "Hw" or "Tw" is used. Governor wide information or control is done by the LN GUNT located in this LD. For example, this includes the local/remote control location. For serving the water flow and penstock pressure information, the LN HWCL is used. All required electrical information (active power, frequency) are inside the integrated LN MMXU. The used GGIO only displays the summary information of secondary signal failures.

Figure 52 shows a governor control.

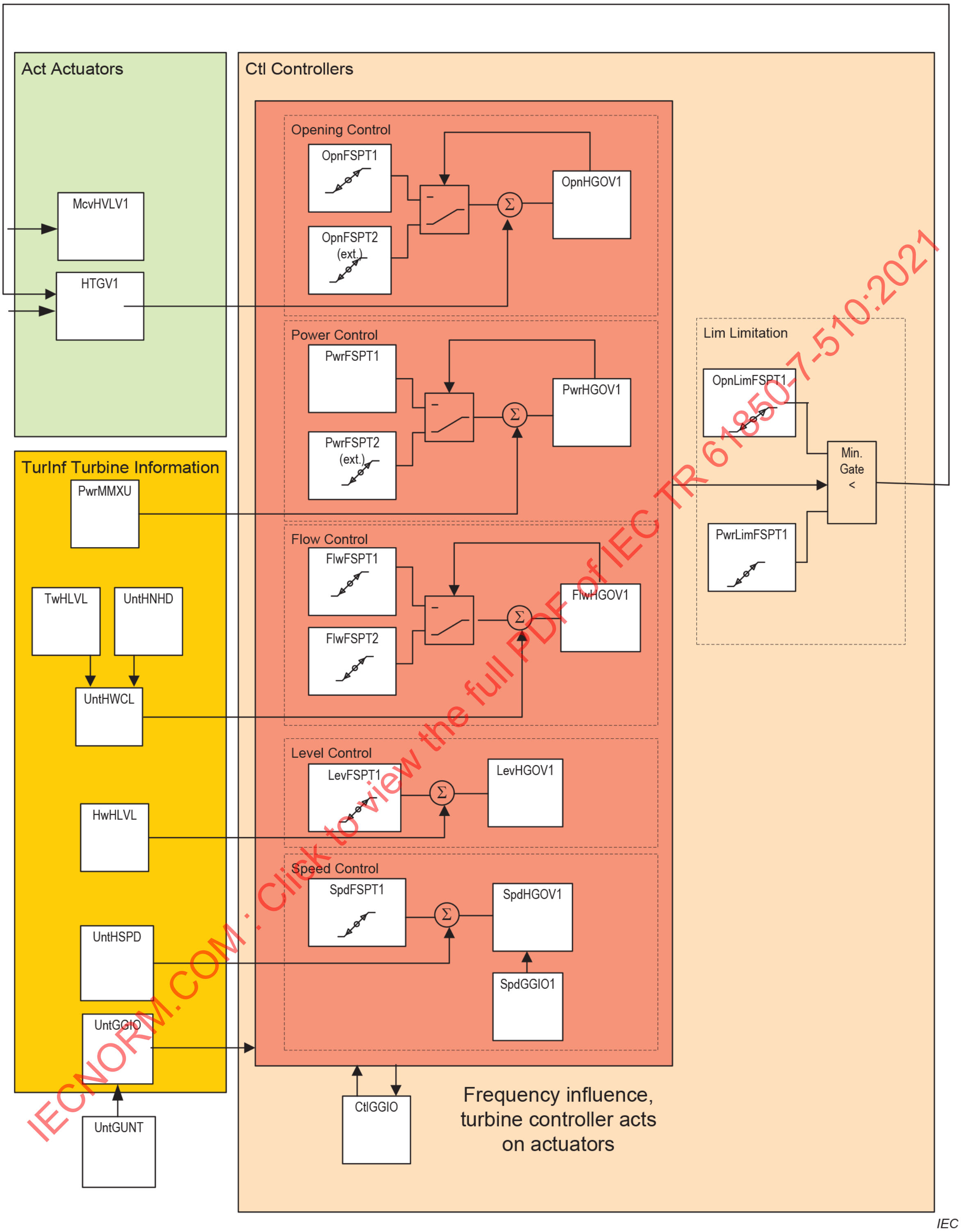


Figure 52 – Governor control

The behaviour of the Governor is controlled by the LNs inside the LD "Controllers". Except the "activation of the frequency influence" and the "information of acting on actuators", the LD contains all control modes and the output limitation. All process related information used in the LD "Controllers" is provided by the LD "Turbine Information" and the feedback information of the guide vane by the LD "Actuators". The control modes are structured all in the same way, to ensure that each control mode can act independent of all others. Generally, each control mode consists of up to two set points. The selection of the actual used set point is done via the LN HGOV. The actual used set point at this LN FSPT is forwarded to the DataName "Out" and "SptMem". The LN HGOV is also used for specifying the Droop, activating a control mode and serving the unlimited output of each control mode.

Figure 53 shows a flow control, Figure 54 shows a level control, Figure 55 shows a speed control.

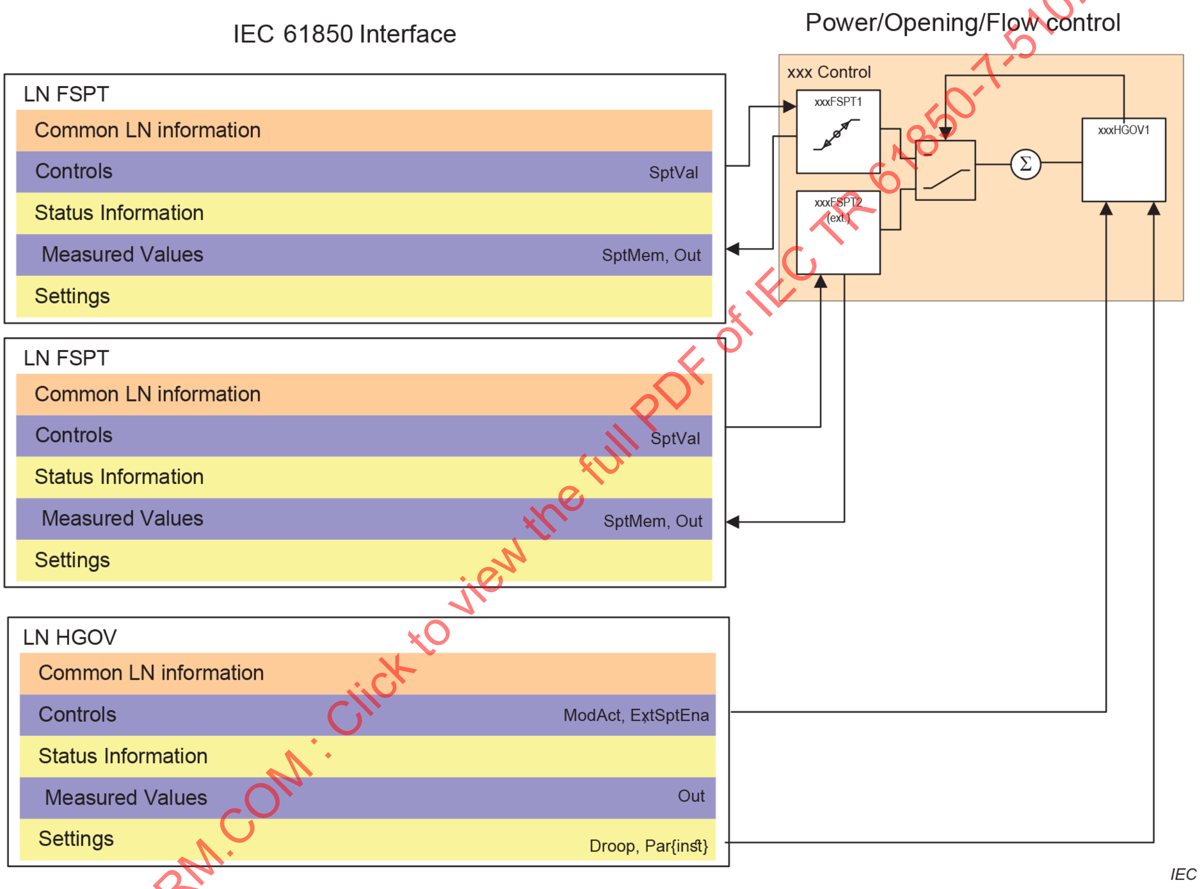


Figure 53 – Flow control

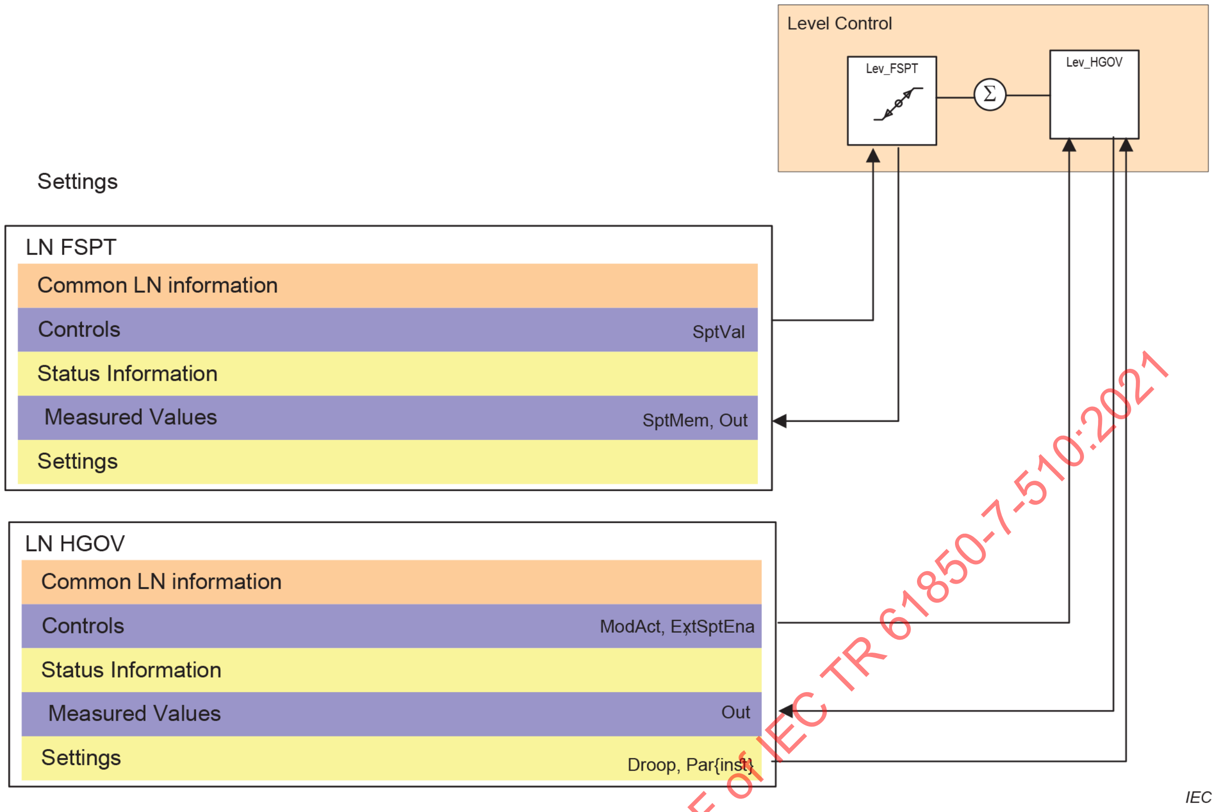


Figure 54 – Level control

It is recommended to use the established sea level as the reference level for water levels. Additionally, a local offset to the level can be used. The level is then expressed as meters above sea level (masl), including a possible offset. Refer to LN HLVL.

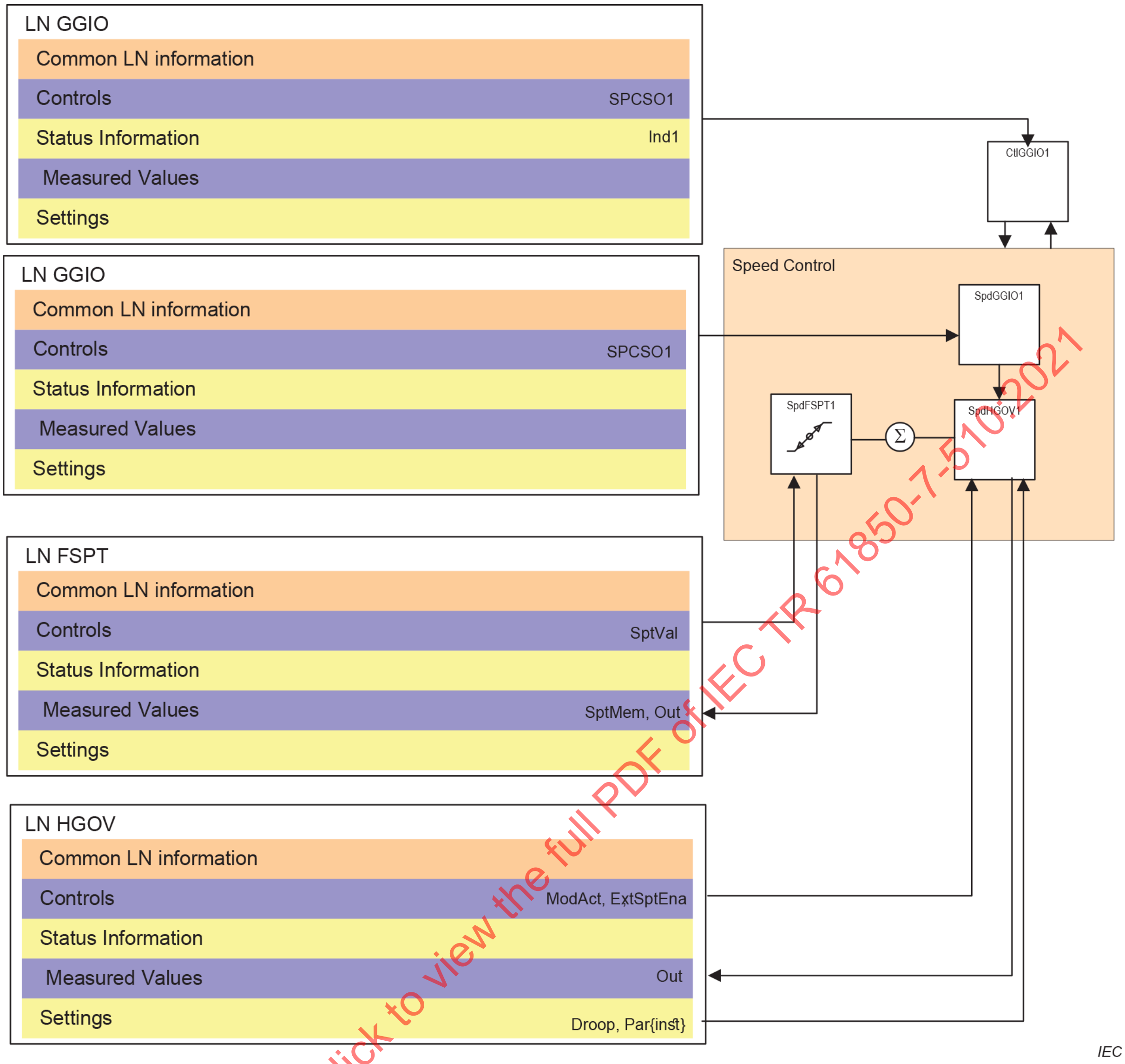


Figure 55 – Speed control

At the speed controller, the control mode can be extended by the option to activate the insensitivity mode via the LN "Spd_GGIO". The LN "Ctl_GGIO" takes care about the "activation of the frequency influence" and the "information of acting on actuators" for all control modes.

The output of any individual controller can be used as an input for another control loop or as final output signal of the selected control mode. This resulting output is forwarded to an output limitation to ensure neither the opening or power limit is exceeded. Both limitations set points can either be set via the LN FSPT at the DataName "SptVal" or with an internal calculated value. At the DataName "SptMem", the internal calculated set point is displayed.

The DataName "Out" indicates the actual used set point. Figure 56 shows limitations.

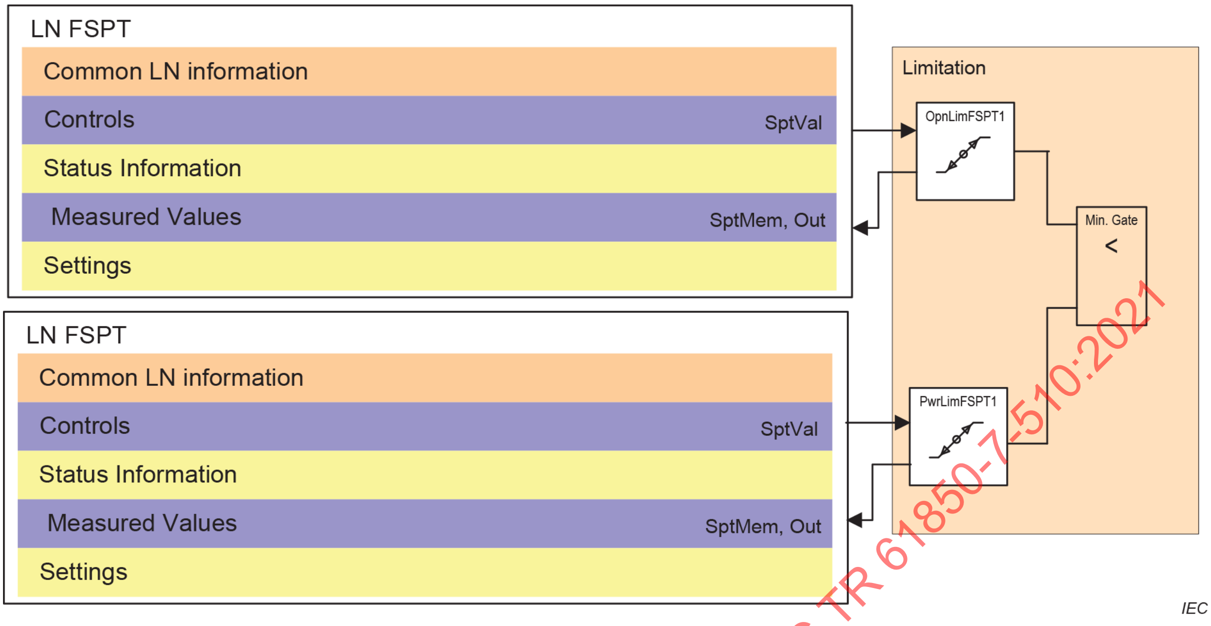


Figure 56 – Limitations

The set point "PosSpt" for the guide vane control LN HTGV in the LD "Actuators" is provided by the LD "Controllers". From the main control valve only status information is available at the LN HVLV. Figure 57 shows an actuator control.

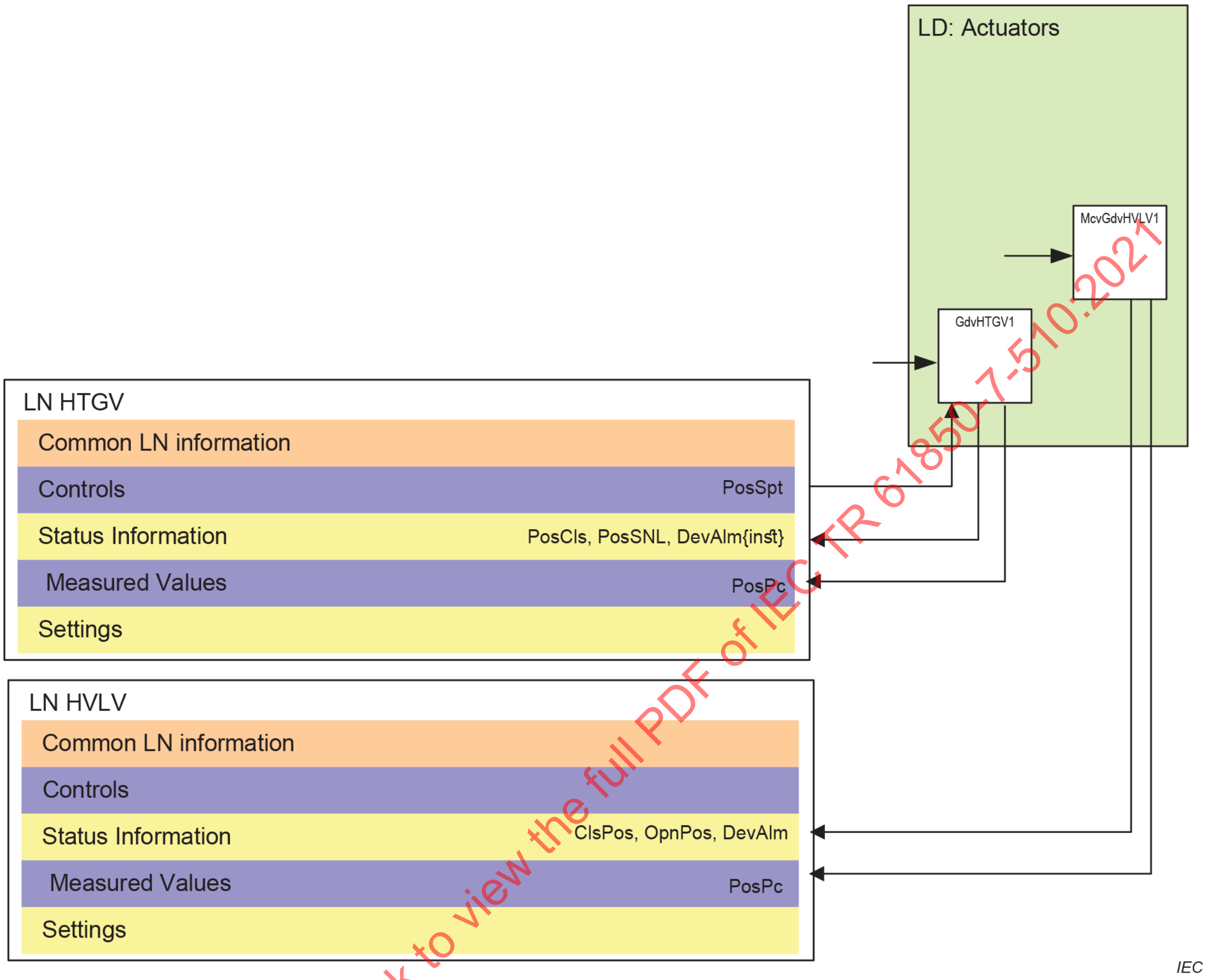


Figure 57 – Actuator control

9.9 Example of a braking system

9.9.1 General

A brake is a mechanical device that inhibits motion by absorbing energy from a moving system. It is used in hydro for slowing or stopping a rotating generator shaft or to prevent its motion, most often accomplished by means of friction applied by a brake pad.

The brake pad is applied by using pressured oil or air via a two-way valve. The brake is normally equipped with a limit switch to safely indicate its position. Two examples are provided in this document.

9.9.2 Brake control with mandatory data objects in LN: HMBR

Figure 58: example uses solely the HMBR logical node with its mandatory data objects.

The circuit works in the following way. The pressure is applied by sending an Operate order to the two-way valve. The way valve latches in the position which provides pressure to the brake pad. The Operate.stVal data attribute confirms that the two-way valve acted on order.

The pressure on the brake pad is unapplied by sending the OpRs order to the two-way valve. The way valve latches in the position to vent the pressure to atmospheric pressure. The OpRs.stVal data attribute confirms that the two-way valve acted on the order.

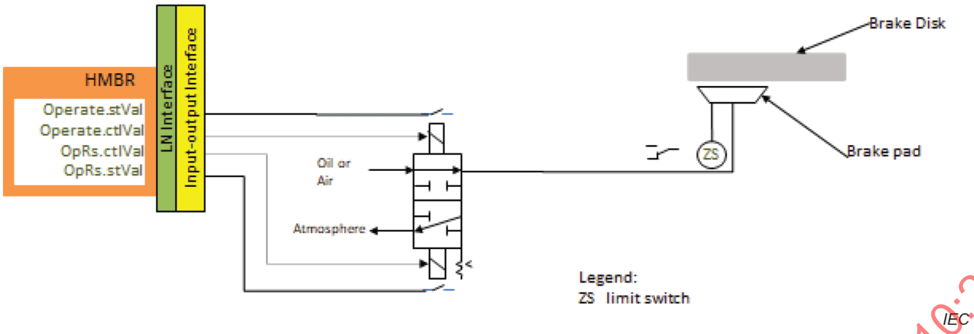


Figure 58 – Brake control with mandatory data objects

9.9.3 Brake control with process indications

Figure 59: example uses mandatory and optional data objects in the HMBR logical node to achieve pertinent process indications needed to supervise the brake system.

The pressure is applied by sending an Operate order to the two-way valve. The two-way valve latches in the position which provides pressure to the brake pad. The Operate.stVal data attribute confirms that the two-way valve acted on the order. The ZS limit switch confirms that the Brake is applied onto the braking surface.

The pressure on the brake pad is unapplied by sending the OpRs order to the two-way valve. The way valve latches in the position to vent the pressure to atmospheric pressure. The OpRs.stVal data attribute confirms that the two-way valve acted on the order. The ZS limit switch confirms that the Brake is unapplied onto the braking surface.

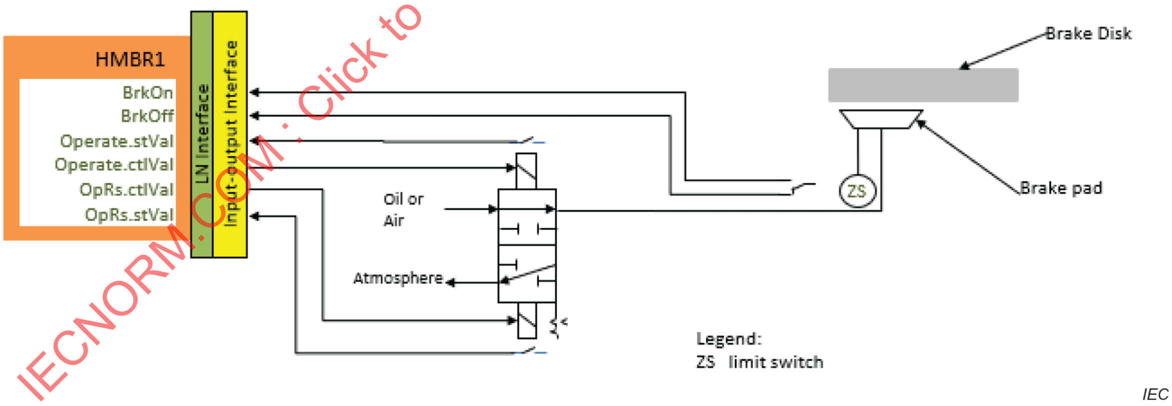


Figure 59 – Brake control with indications

9.10 Example of a heater system

9.10.1 General

Heaters are devices which increase the temperature of an environment to a desired setting. Heaters are composed of heating resistive elements. It is used in hydro for maintaining the temperature of hydraulic fluids for bearing lubrication and servo motors. Heaters are also used for anticondensation/anti-frost purposes in different systems (generators, control panels, oil sumps, etc.).

The heater can have multi-stage heating elements to maintain the temperature within a temperature band. Each stage can be individually controlled by a threshold setting.

9.10.2 Example of a LN: KHTR usage

Figure 60: KHTR can be used to model those different types of heaters. In this example it is used to model an 8-stage heater controlled by a step controller interface for maintaining the oil temperature in a tank.

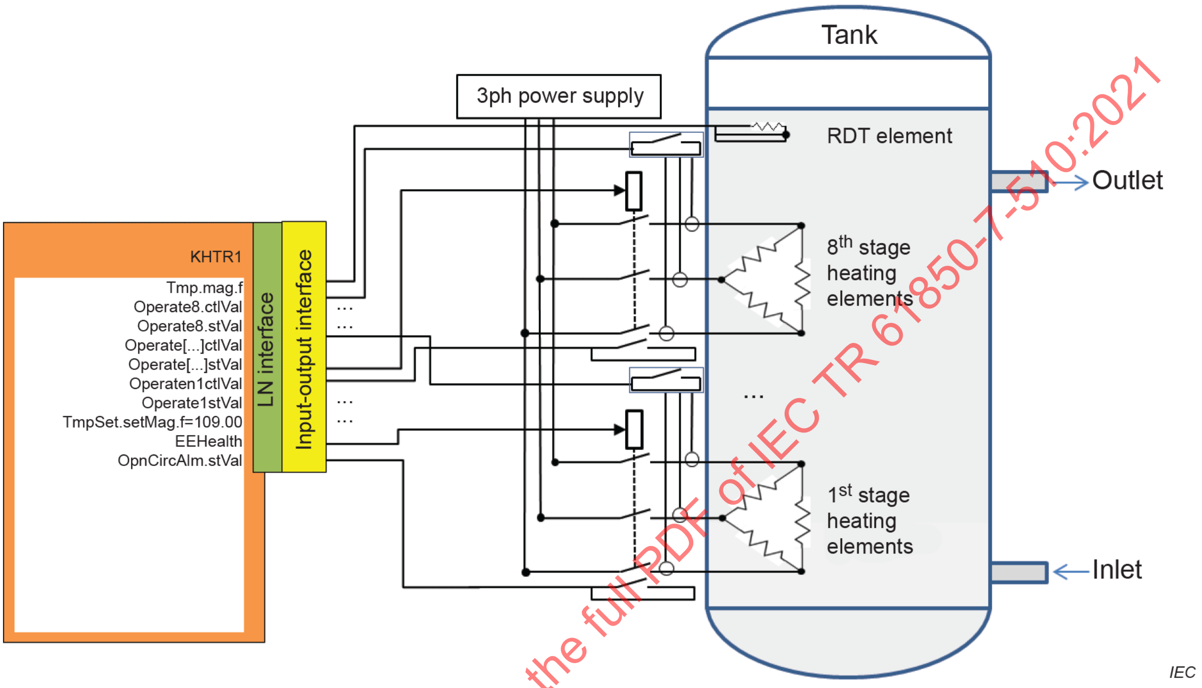


Figure 60 – Oil tank heater using a step controller

Temperature setpoint is set via the KHTR1.TmpSet.setMag.f attribute. A resistive temperature detector (RTD element) captures the temperature in the tank and can be read via the KHTR1.Tmp.mag.f attribute. The heating elements stages are activated to control the temperature of the medium in the tank. The control logic is not included in this example. A current detector is used to detect a broken element by generating the open circuit alarm KHTR1.OpnCircAlm.stVal. The EEHealth data object contains information regarding the overall health of the external equipment (the heater itself).

9.11 Examples of how to reference a start / stop sequencer of a hydropower unit

9.11.1 General

The examples are for a simple turbine generator unit with an intake gate and no inlet valve. The guide vanes are equipped with servomotor locks and the unit is supplied with a lubrification unit and brakes. The generator has a cooling system.

There are several different tripping strategies widely used as common practice today depending on a combination of different tripping criteria, different servomotor shutdown initiating devices and the corresponding sequence of tripping actions.

The example below is widely used in the hydro community (see Table C.2 of IEC 61362:2012); a second widely used strategy is described in Table C.1 of IEC 61362:2012.

9.11.2 Unit sequences definition with IEC 61850

Each unit sequence is defined by a "HSEQ[3]" LN and it is included in a dedicated LD. All of them are grouped together in a group reference LD called "Seq" (unit start-stop sequencer). Only the "Seq" group reference logical device will have the LLN0 and LPHD logical nodes. Figure 61 shows a sequencer overview.

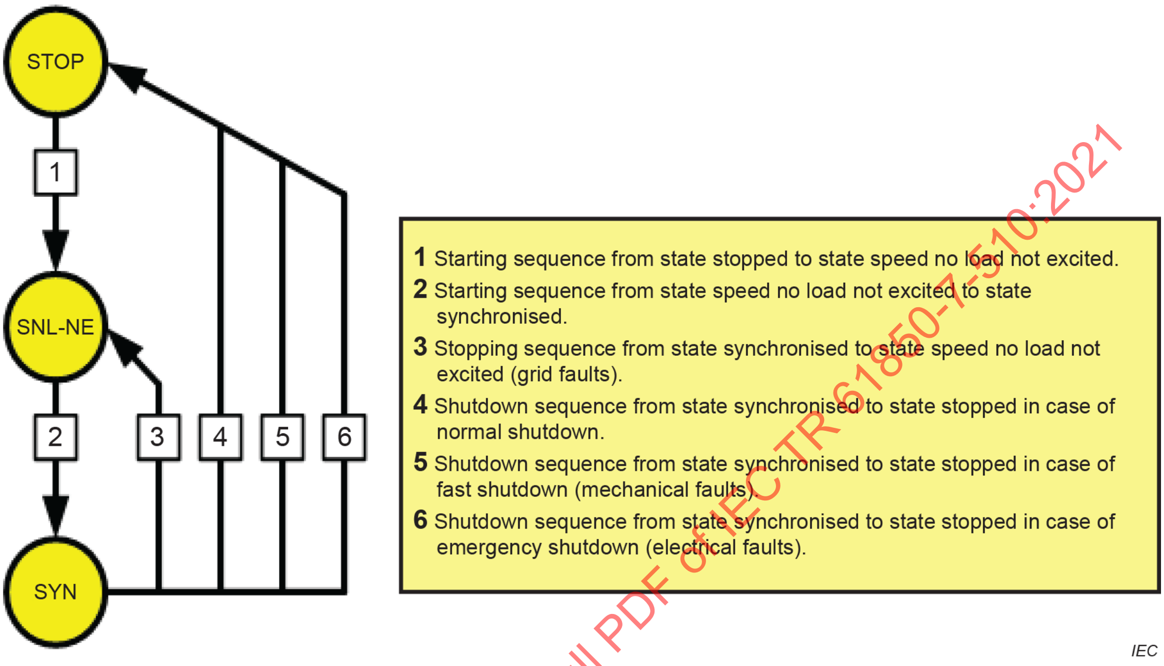


Figure 61 – Sequencer overview

Table 12 summarizes the most common specified names for the Processes used in sequencing. Note that in this example all sequencies are assumed to be modelled at unit level (A1). As in the previous examples it is assumed that there is a local definition of the SCL:Function names to identify for example the type of sequence used. The SCL:Function name is a local implementation issue.

Table 12 – Alpha2 Typical sequences

Sequence in Figure 22	SCL:Process structure	SCL: Function name	Description
1	Alpha2.PS1.A1	Seq1	Starting sequence from stop to speed no load not excited
2	Alpha2.PS1.A1	Seq2	Sequence from speed no load not excited to state synchronised
3	Alpha2.PS1.A1	Seq3	Grid fault trip
4	Alpha2.PS1.A1	Seq4	Normal Shutdown sequence
5	Alpha2.PS1.A1	Seq5	Mechanical fault
6	Alpha2.PS1.A1	Seq6	Electrical fault shutdown

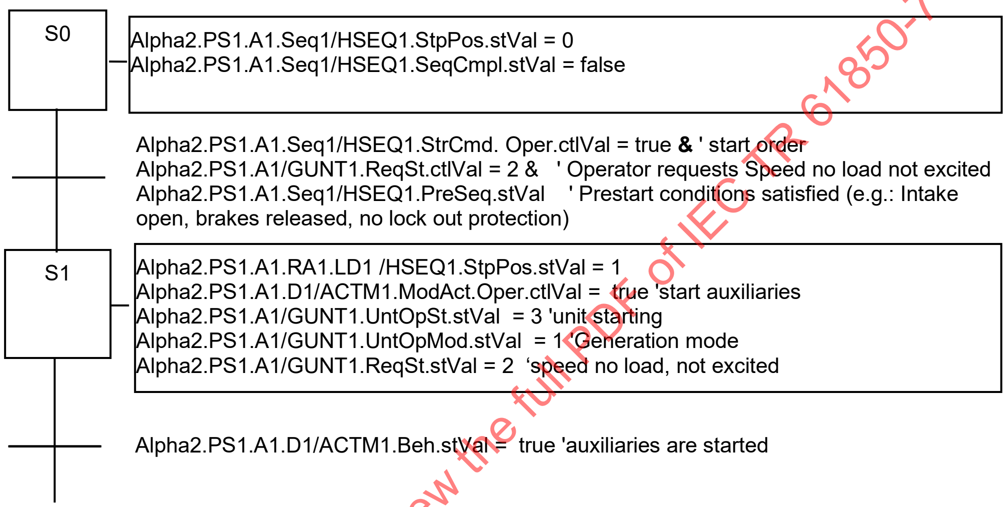
3 Logical Nodes HSEQ and ASEQ defined in IEC 61850-7-410:2012+Amd.1:2015 are actually the same Logical Node (HSEQ) because of an editorial error in the Amendment 1.

9.11.3 Start sequence from a state "stopped" to a state "speed no load not excited" (Sequence 1)

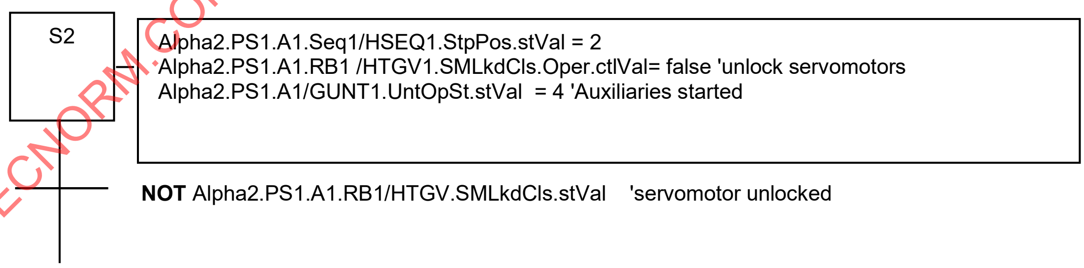
The starting sequence from a state "stopped" to a state "speed no load not excited" is activated from an operator order only if the unit is in the state stopped and if the starting initial conditions are present. At the end of each step and if the step by step mode is selected, an operator validation is necessary to activate the step N+1.

The sequence (start sequence up to "speed no load not excited" state) may be broken down into the following steps:

- Step 1: Starting of the unit auxiliaries (Lubrification, oil pressure ...). Step 1 is valid while the feedback of the unit auxiliaries started is not present. A timer is necessary to control the length of step 1. In case of step 1 taking too long, the fast shutdown sequence is automatically activated.

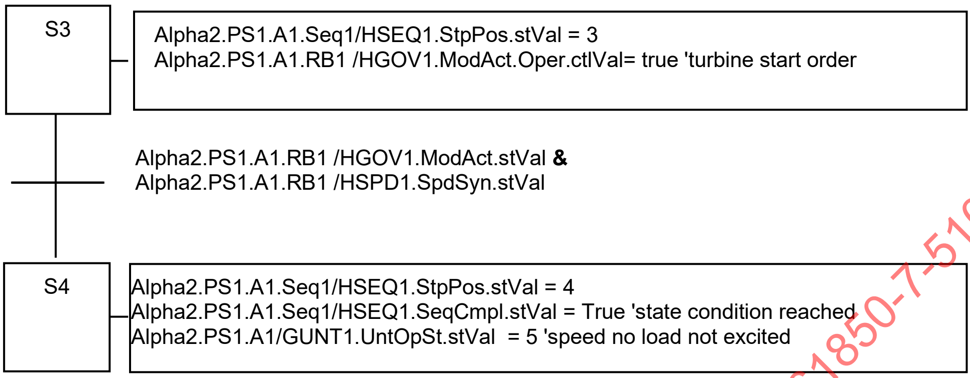


- Step 2: Releasing of the unit locks (locks on the wicket gates). Step 2 is valid while the feedback of the unit locks released is not present. A timer is necessary to control the length of step 2. In case of step 2 too long length, the quick shutdown sequence is automatically activated.



- Step 3: Running of the unit (managed by the speed regulator). Step 3 is valid while the feedback of the start excitation unit speed is not present. A timer is necessary to control the length of step 3. In case of step 3 taking too long, the fast shutdown sequence (sequence 5) is automatically activated.

  At the end of step 3, the state "speed no load not excited" is reached.



9.11.4 Start sequence from state "speed no load not excited" to state "synchronised" (Sequence 2)

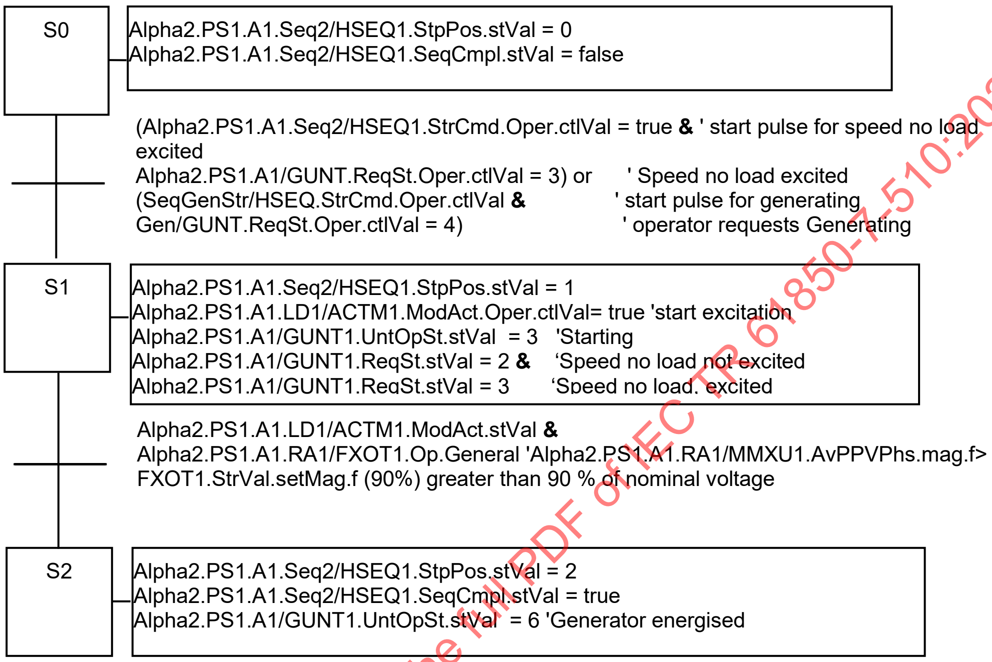
The start sequence from state "speed no load not excited" to state “synchronised" is automatically activated if an operator requested the unit in state “speed no load not excited” to change to the state “synchronised”.

After a grid fault and an automatic return from the state “synchronised” to the state "speed no load not excited", the start sequence can also be activated by an operator to bring the unit back to state “synchronised” (when the grid fault has been acknowledged and is not present any more). At the end of each step and if the step by step mode is selected, an operator validation is necessary to activate the step N+1.

The sequence (starting sequence up to "synchronised" state) may be broken down into the following steps:

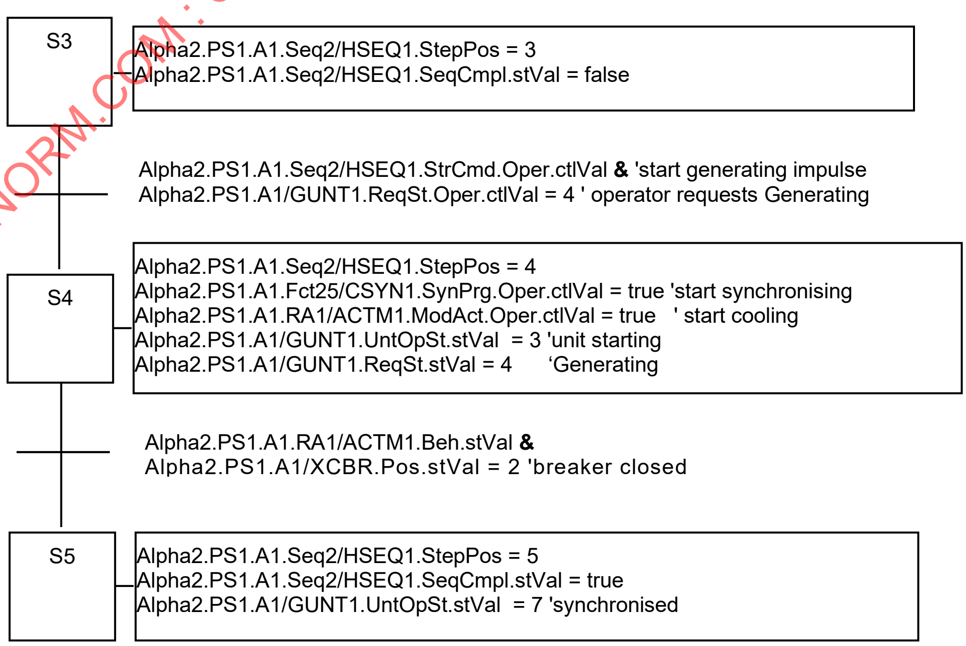
- Step 1: Starting of the excitation system. Step 1 is valid while the feedback of the unit voltage equal to 90 % of the nominal voltage is not present. A timer is necessary to control the length of step 1. In case of step 1 taking too long, the fast shutdown sequence (sequence 5) is automatically activated.

  At the end of step 1, the state "speed no load excited" is reached.



- Step 2: Starting of the generator cooling and unit synchronization to the grid. Step 2 is valid while the feedback of the generator cooling started, and the feedback of the unit circuit breaker closed are not present. A timer is necessary to control the length of step 2. In case of step 2 taking too long, the fast shutdown sequence is automatically activated.

  At the end of the sequence, the state synchronised is reached.

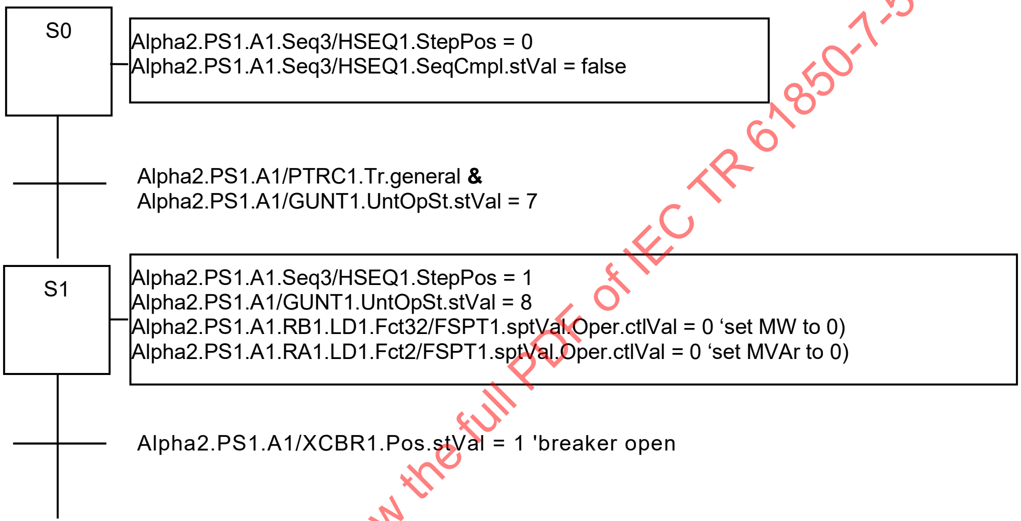


9.11.5 Stop sequence from state "synchronised" to state "speed no load not excited" (sequence 3)

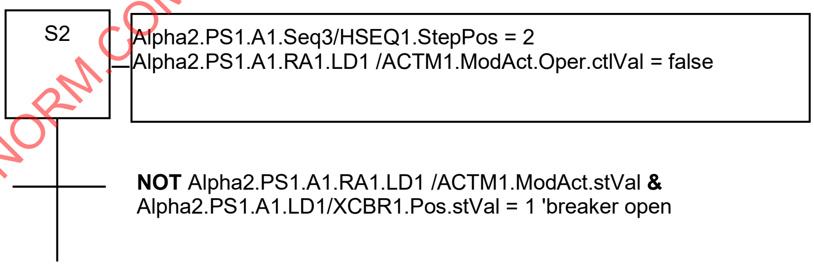
The grid fault stopping sequence from state " synchronised " to state "speed no load not excited" is automatically activated if a grid fault occurs and the state of the unit is synchronised. If the step-by-step mode was selected, it is automatically changed to the automatic mode when the sequence is activated.

The sequence (stopping sequence up to "speed no load not excited" state) may be broken down into the following steps:

- Step 1: Opening of the unit circuit breaker and updating of the active power set point with the value 0. Step 1 is valid while the feedback of the unit circuit breaker open is not present. A timer is necessary to control the length of step 1. In case of step 1 taking too long, the fast shutdown sequence is automatically activated.

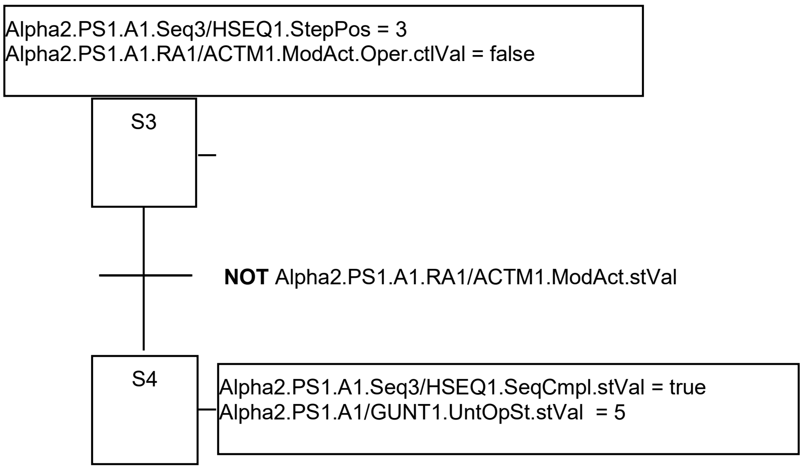


- Step 2: Stopping of the excitation system. Step 2 is valid while the feedback of the unit excitation stopped is not present. A timer is necessary to control the length of step 2. In case of step 2 taking too long, the fast shutdown sequence is automatically activated.



- Step 3: Stopping of the generator cooling. Step 3 is valid while the feedback of the generator cooling stopped is not present. A timer is necessary to control the length of step 3. In case of step 3 taking too long, the fast shutdown sequence is automatically activated.

  At the end of the sequence, the state "speed no load not excited" is reached.

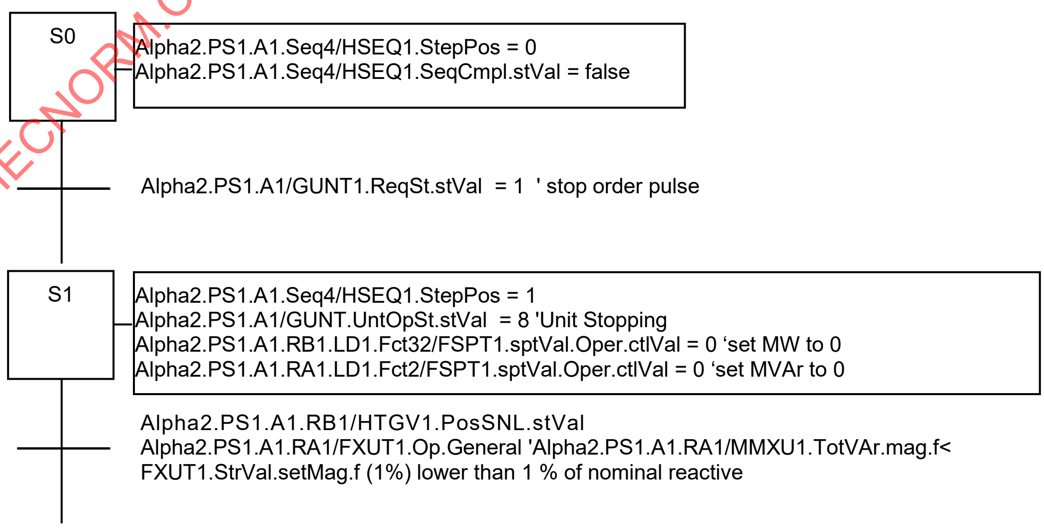


9.11.6 Shutdown sequence from state " synchronised " to state "stopped" (Sequence 4)

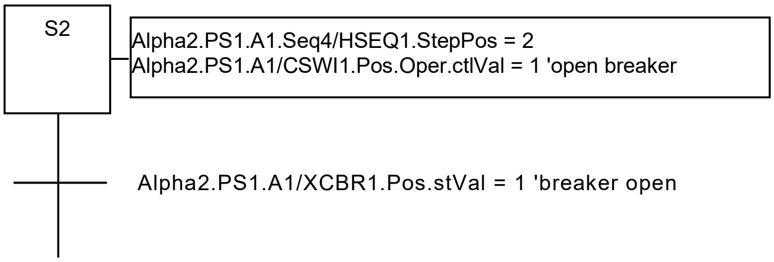
The normal shutdown sequence is activated from an operator order if the state of the unit is different from the state stopped or blocked and if the higher priority shutdown sequences are not already activated (fast shutdown sequence). At the end of each step and if the step by step mode is selected, an operator validation is necessary to activate the step N+1.

The sequence (shutdown sequence up to stopped state) may be broken down into the following steps:

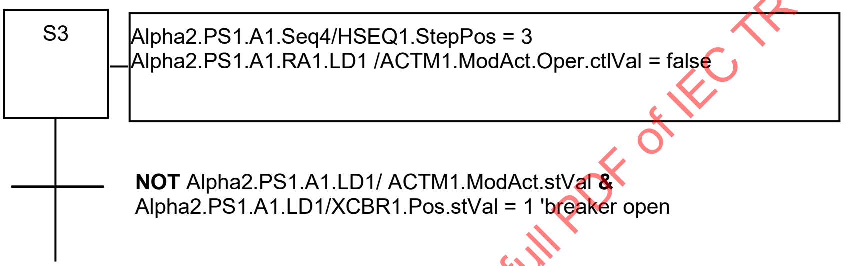
- Step 1: Updating of the active power set point with the value 0. Step 1 is valid while the feedback of the unit speed no load position on the wicket gates is not present. A timer is necessary to control the length of step 1. In case of step 1 taking too long, the fast shutdown sequence is automatically activated.



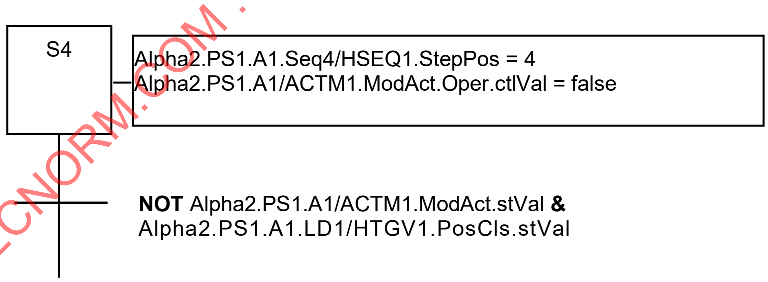
- Step 2: Opening of the unit circuit breaker. Step 2 is valid while the feedback of the unit circuit breaker open is not present. A timer is necessary to control the length of step 2. In case of step 2 taking too long, the fast shutdown sequence is automatically activated.



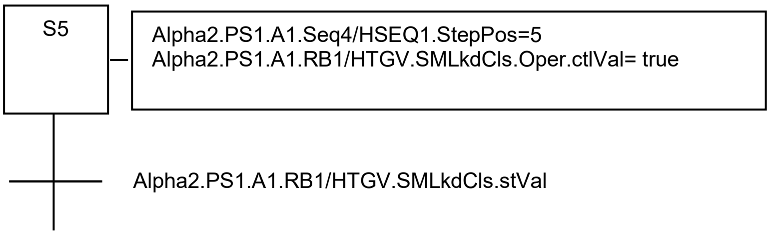
- Step 3: Stopping of the excitation system. Step 3 is valid while the feedback of the unit excitation stopped is not present. A timer is necessary to control the length of step 3. In case of step 3 taking too long, the fast shutdown sequence is automatically activated.



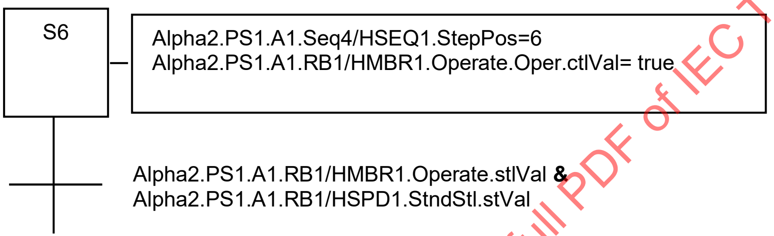
- Step 4: Stopping of the generator cooling and complete closing of the wicket gates. Step 4 is valid while the feedback of the generator cooling stopped, and the feedback of the wicket gates closed are not present. A timer is necessary to control the length of step 4. In case of step 4 taking too long, the fast shutdown sequence is automatically activated.



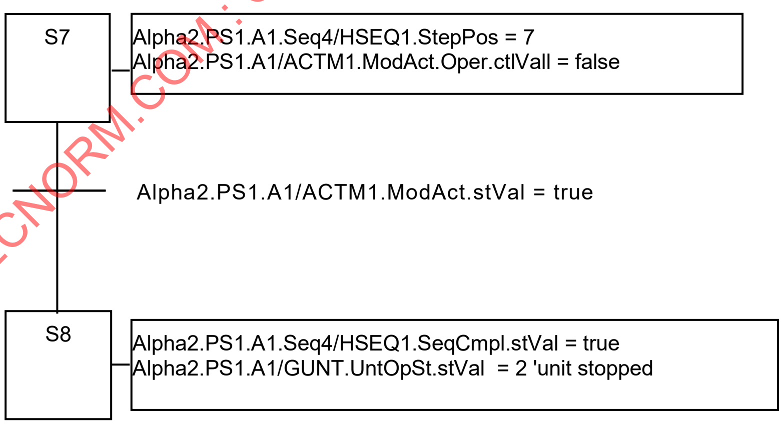
- Step 5: Applying of the unit locks (locks on the wicket gates). Step 5 is valid while the feedback of the unit locks applied, and the feedback of the unit braking speed are not present. A timer is necessary to control the length of step 5. In case of step 5 taking too long l, the fast shutdown sequence is automatically activated.



- Step 6: Applying of the unit brakes. Step 6 is valid while the feedback of the unit speed equal to 0 is not present. A timer is necessary to control the length of step 6. In case of step 6 taking too long, the fast shutdown sequence (sequence 5) is automatically activated.



- Step 7: Stopping of the unit auxiliaries (lubrification, oil pressure ...). Step 7 is valid while the feedback of the unit auxiliaries stopped is not present. A timer is necessary to control the length of step 7. In case of step 7 taking too long, the fast shutdown sequence is automatically activated. At the end of the sequence 4, the state stopped is reached.

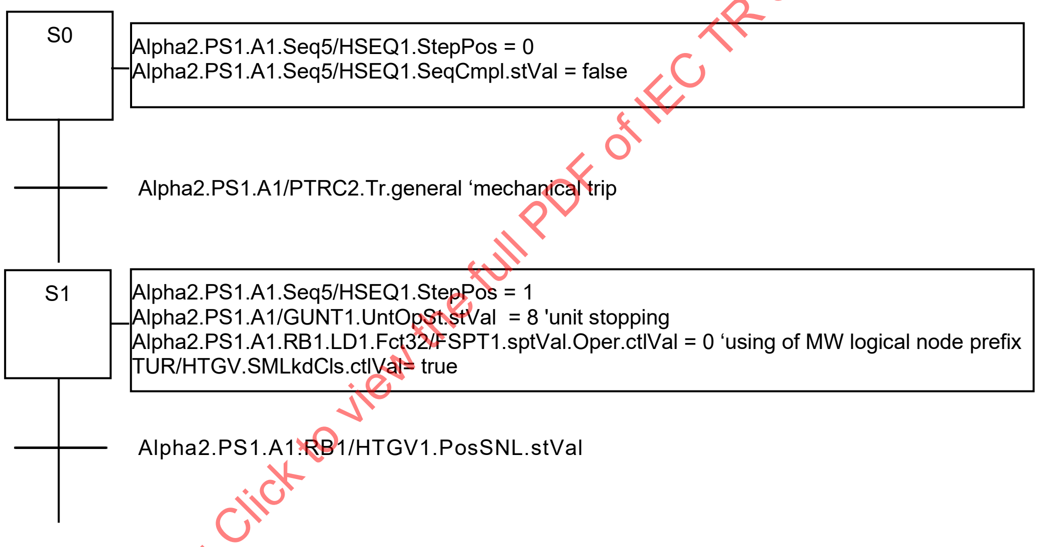


9.11.7 Fast shutdown sequence from state " synchronised " to state "stopped" (Sequence 5)

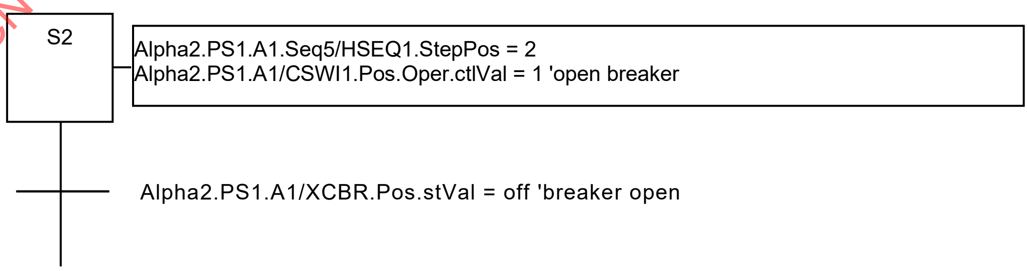
The fast shutdown sequence from state " synchronised " to state "stopped" is automatically activated if a mechanical fault occurs, a loss of the permanent running conditions occurs when the unit is in the mode synchronised, a sequence step taking too long occurs, if the state of the unit is different from the state stopped or blocked and if the highest priority shutdown sequence is not already activated (emergency shutdown sequence). If the step by step mode was selected, it is automatically changed to the automatic mode when the sequence is activated. For historical reasons the name of this sequence may be different (e.g. quick shut down, or emergency shut down) between implementations.

The sequence (shutdown sequence up to stopped state in case of mechanical fault) may be broken down into the following steps:

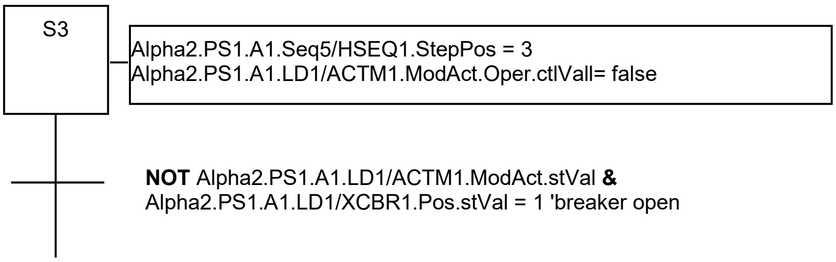
- Step 1: Complete closing of the wicket gates and updating of the active power set point with the value 0. Step 1 is valid while the feedback of the unit speed no load position on the wicket gates is not present. A timer is necessary to control the length of step 1. In case of step 1 taking too long, the emergency shutdown sequence is automatically activated.



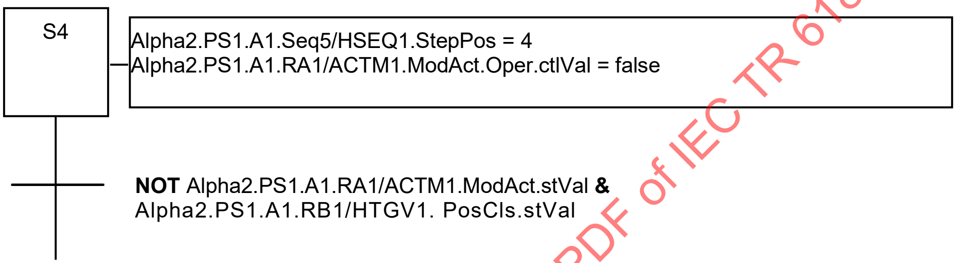
- Step 2: Opening of the unit circuit breaker. Step 2 is valid while the feedback of the unit circuit breaker open is not present. A timer is necessary to control the length of step 2. In case of step 2 taking too long, the emergency shutdown sequence is automatically activated.



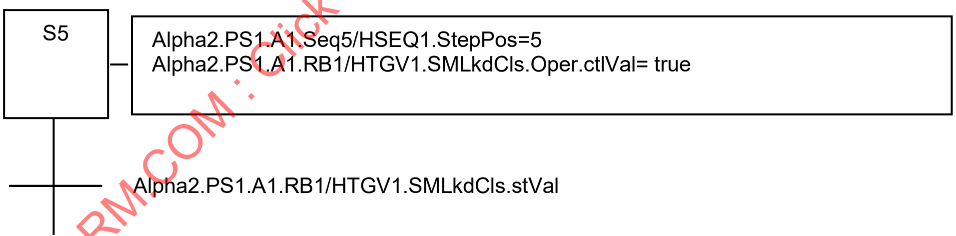
- Step 3: Stopping of the excitation system. Step 3 is valid while the feedback of the unit excitation stopped is not present. A timer is necessary to control the length of step 3. In case of step 3 taking too long, the emergency shutdown sequence is automatically activated.



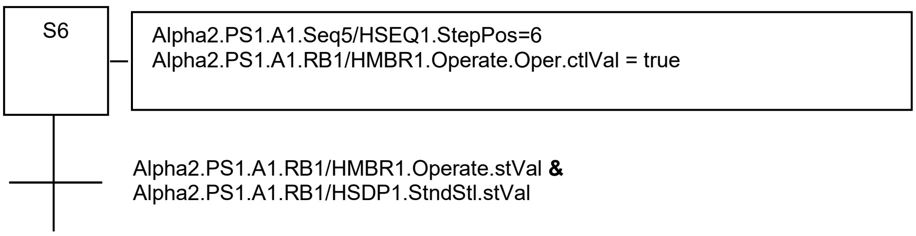
- Step 4: Stopping of the generator cooling. Step 4 is valid while the feedback of the generator cooling stopped, and the feedback of the wicket gates closed are not present. A timer is necessary to control the length of step 4. In case of step 4 taking too long, the emergency shutdown sequence is automatically activated.



- Step 5: Applying of the unit locks (locks on the wicket gates). Step 5 is valid while the feedback of the unit locks applied, and the feedback of the unit braking speed are not present. A timer is necessary to control the length of step 5. In case of step 5 taking too long, the emergency shutdown sequence is automatically activated.

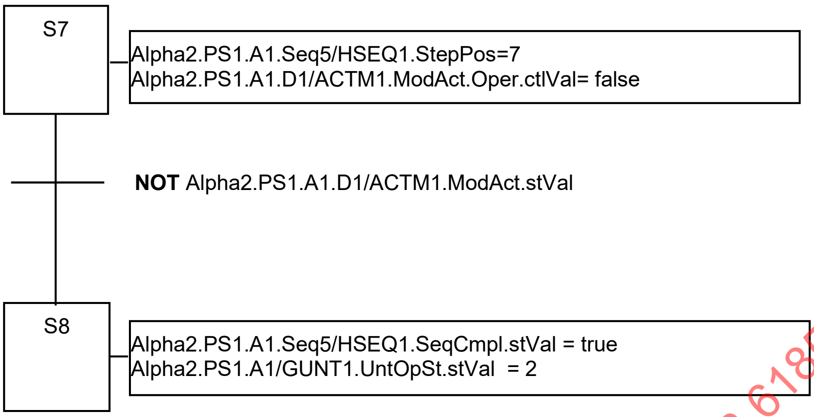


- Step 6: Applying of the unit brakes. Step 6 is valid while the feedback of the unit speed equal to 0 is not present. A timer is necessary to control the length of step 6. In case of step 6 taking too long, the emergency shutdown sequence is automatically activated.



- Step 7: Stopping of the unit auxiliaries (lubrification, oil pressure …). Step 7 is valid while the feedback of the unit auxiliaries stopped is not present. A timer is necessary to control the length of step 7. In case of step 7 taking too long, the emergency shutdown sequence is automatically activated.

  At the end of the sequence, the state stopped is reached.

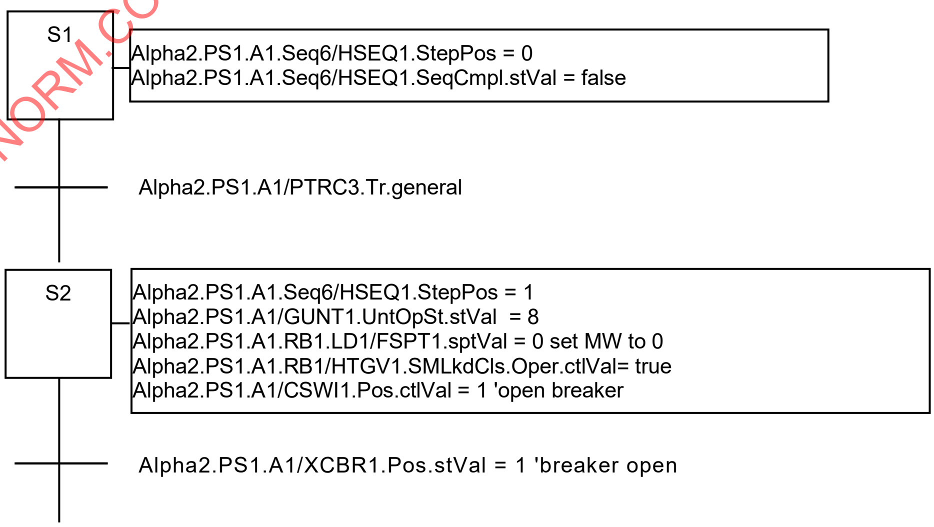


9.11.8 Emergency shutdown sequence from state " synchronised " to state "stopped" (sequence 6)

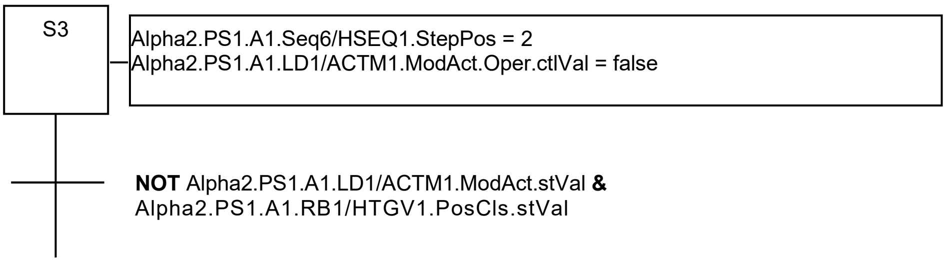
The emergency shutdown sequence from state "synchronised" to state "stopped" is automatically activated if either a fault occurs, or an operator order is given. If the step by step mode was selected, it is automatically changed to the automatic mode when the sequence is activated.

The sequence may be broken down into the following steps:

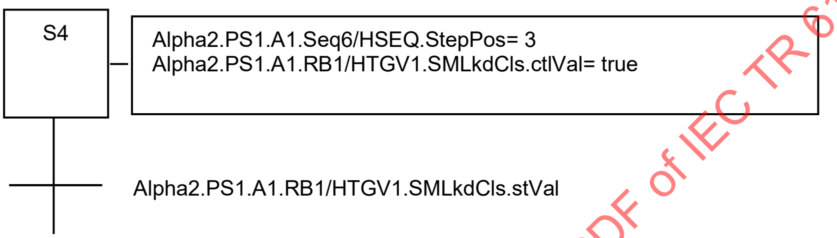
- Step 1: Complete closing of the wicket gates, opening of the unit circuit breaker and updating of the active power set point with the value 0. Step 1 is valid while the feedback of the unit circuit breaker open is not present. A timer is necessary to control the length of the step 1. In case of step 1 taking too long, the unit state blocked is automatically activated.



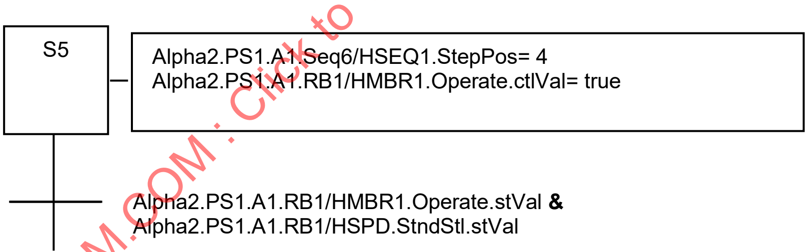
- Step 2: Stopping of the excitation system. Step 2 is valid while the feedback of the unit excitation stopped is not present. A timer is necessary to control the length of step 2. In case of step 2 taking too long, the unit state blocked is automatically activated.



- Step 3: Applying of the unit locks (locks on the wicket gates). Step 3 is valid while the feedback of the unit locks applied, and the feedback of the unit braking speed are not present. A timer is necessary to control the length of step 3. In case of step 3 taking too long, the unit state blocked is automatically activated.

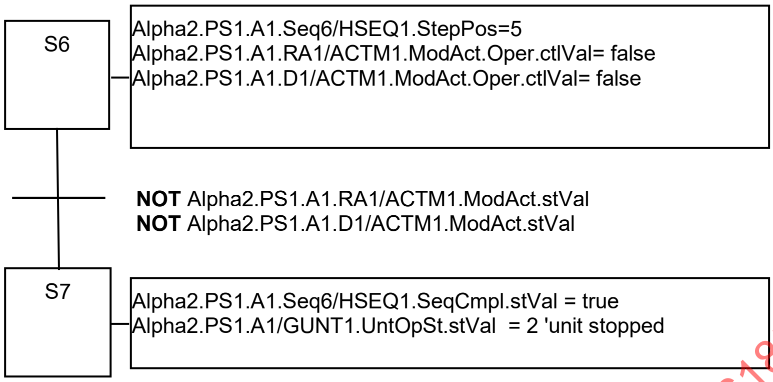


- Step 4: Applying of the unit brakes. Step 4 is valid while the feedback of the unit speed equal to 0 is not present. A timer is necessary to control the length of step 4. In case of step 4 taking too long, the unit state blocked is automatically activated.



- Step 5: Stopping of the unit auxiliaries (Lubrification, cooling, oil pressure …). Step 5 is valid while the feedback of the unit auxiliaries stopped is not present. A timer is necessary to control the length of step 5. In case of step 5 taking too long, the unit state blocked is automatically activated.

  At the end of the sequence, the state stopped is reached.



9.12 Example of a capability chart representation

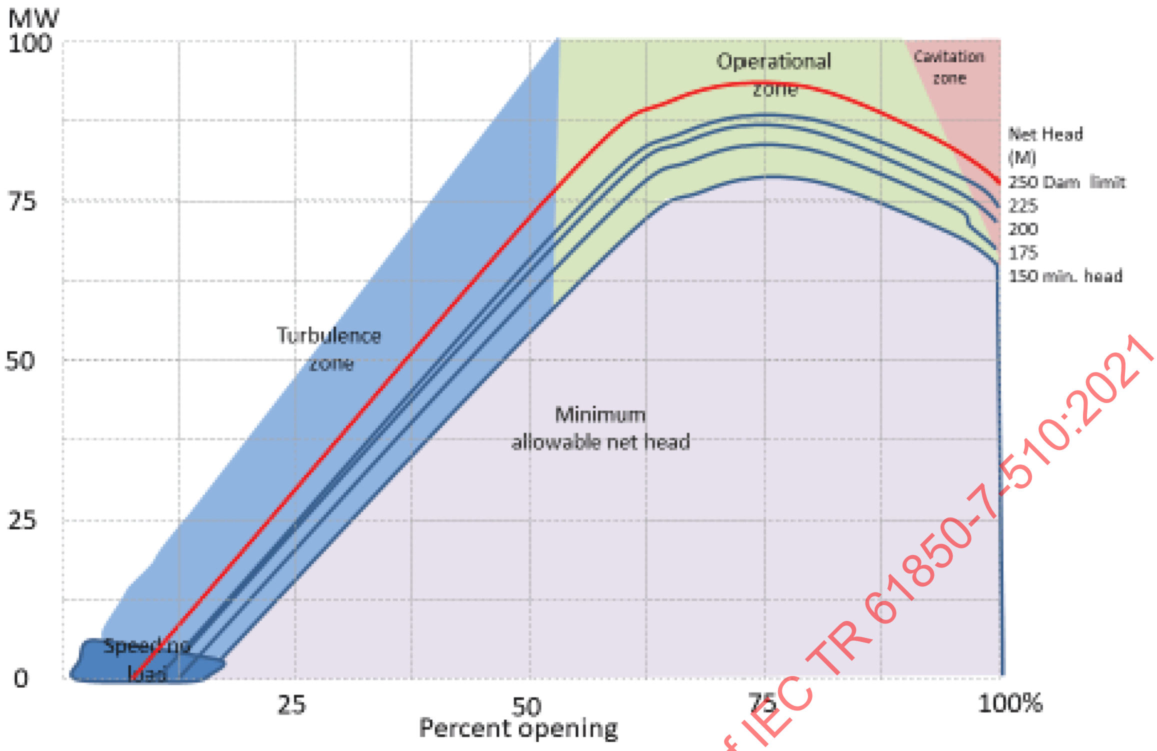
9.12.1 General

The capability chart (Hill chart) is commonly used to define operational zones and to select suitable operational setpoints to e.g. optimize process efficiency while minimising the risk of cavitation.

In IEC 61850 the capability chart can be represented as curves.

9.12.2 Example of a capability curve

Figure 62 shows as example of a capability curve of a hydropower turbine:



IEC

Figure 62 – An example of a capability curve

The information in the capability curve can be mapped to IEC 61850 according to Table 13.

Table 13 – Capability table

Capability curve	Description	X:Opening Y:MW Z:Net Head
CapFCSD1	curve 1 in the capabilty curve	Array(Point(X,Y,Z))
TrbIncFXOT1	Turbulence zone	Array(Point(X,Y))
OperFXOT2	Operational zone	Array(Point(X,Y))
CavFXOT3	Cavitation zone	Array(Point(X,Y))
SNLFXOT4	Speed no load zone	Array(Point(X,Y))

9.12.3 Example of a Hill chart

Figure 63 shows as an example of a Hill chart of a hydropower turbine:

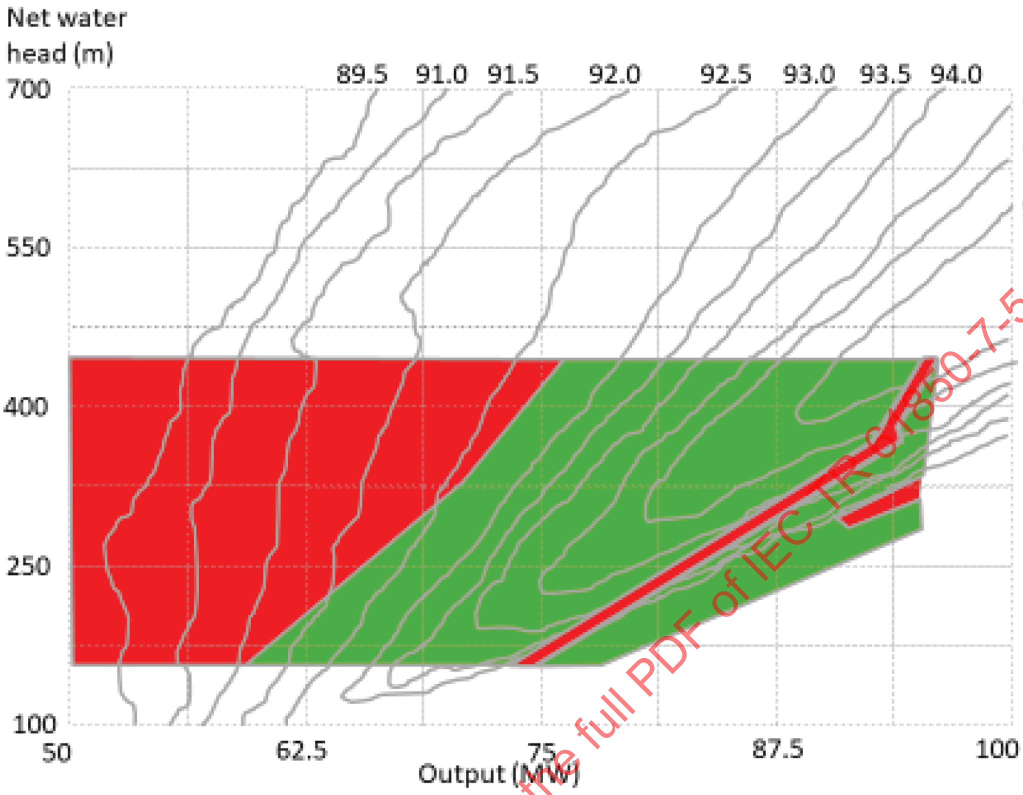


Figure 63 – An example of a Hill chart (five variables)

The information in the Hill chart can be mapped to IEC 61850. Table 14 shows the mapping for the five variable Hill charts.

Table 14 – Mapping of Hill charts

Capability curve	Description	X:Net Head Y:MW Z: Efficiency
CapFCSD1	curve 1 in the capabilty curve	Array(Point(X,Y,Z))
RedFXOT1	Turbulence zone	Array(Point(X,Y))
RedFXOT2	Operational zone	Array(Point(X,Y))
GreenFXOT1	Cavitation zone	Array(Point(X,Y))
GreenFXOT2	Speed no load zone	Array(Point(X,Y))

9.12.4 Example of a multi-layer capability chart

9.12.4.1 General

Figure 64 shows as an example of a multi-layer capability chart of a hydropower turbine:

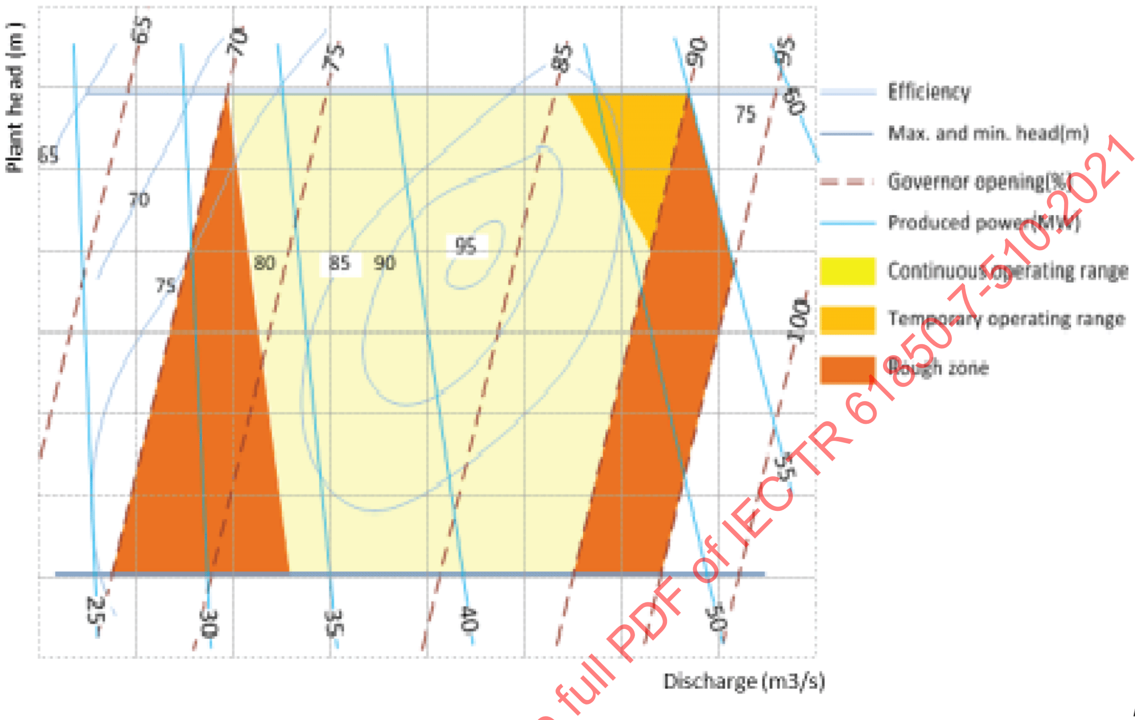


Figure 64 – An example of a multi layered capability chart (five dimensions)

The rough zone is an area of points representing combinations of operational variables (setpoints) that shall not be used during operation, e.g. because of too high cavitation.

The temporary operation range represents combinations of variables (setpoints) that may be used for a limited time, e.g. during starting or stopping of the unit.

The multi layered capability Hill chart can be mapped to IEC 61850. Table 15 shows the mapping for the five-dimensional capability chart.

Table 15 – Five-dimensional capability chart

Capability curve	**Description**	**X: Opening** **Y:MW** **Z: Net Head**
CapFCSD[1..n].Crv[1..n]* **CapFCSD[1..n].Crv**	Extended model (5D): curve [1..n] in the capability curve [1..n] Standard model (4D):	Array(Point(X,Y,Z)) ___
YellowFXOT1	Temporary zone	Array(Point(X,Y))
RoughFXOT2	Cavitation zone 1	Array(Point(X,Y))
RoughFXOT3	Cavitation zone 2	Array(Point(X,Y))
SNLFXOT4	Speed no load zone	Array(Point(X,Y))
*) Note that in this example the Logical Node FCSD Curve Shape Description has been extended with instantiated curve shapes: The DataObject Crv Curve shape has been extended with PrescenceCondition = Mmulti. Without extension of LN FCSD (Crv PrescendeCondition = M) the same principle can be used to map a four-dimensional capability chart to IEC 61850.		

9.12.4.2 Suggested use of dimensions for modelling of capability curves

The example below shows the suggested use of the different dimensions when modelling a hydropower turbine capability chart in IEC 61850.

Dimension 1

CapFCSD[*w*=1..n]: Capability curve(s), for different properties (e.g. efficiency)

Variants of Dimension 1 is represented as Logical Node *instances*:

CapFCSD1: Efficiency (*w*=1) as function of head and flow

CapFCSD2: Power (*w*=2) as function of head and flow

CapFCSD3: Opening (*w*=3) as function of head and flow

CapFCSD[w].crv: Capability curve as function of selected variables according to above

crv.crvPts[*p* = 1...n]. Array of curve points

Dimension 2

crv.crvPts[*p*].xVal: x-value of the curve point *p*: Head

Dimension 3

crv.crvPts[*p*].yVal: y-value of the curve point *p*: Flow

Dimension 4

crv.crvPts[*p*].zVal: z-value of the curve point *p*: Efficiency, Power, or Opening)

Dimension 5

Extended LN FCSD: CapFCSD[*w*=1..n].crv[*b*=1..m]: Capability Curve at a given value *b* corresponding to e.g. kaplan runner opening (%) or cavitation (α).

9.13 Pump start priorities of a high-pressure oil system

9.13.1 General

This example will demonstrate how to use IEC 61850 logical nodes to program a start-stop sequence for pumps using the FXPS logical node to manage the pump start priorities of a high-pressure oil system.

The high pressure oil system is identified using the SCL:Process identifier based on RDS-PS as Alpha2.PS1.A1.RB1.HE1, since it is associated to the turbine technical system. The high-pressure unit has two sequences as identified in Table 16.

Table 16 – Alpha2 Typical pump sequences

SCL:Process structure	SCL: Function name	Description
Alpha2.PS1.A1.RB1.HE1		High pressure oil supply system
Alpha2.PS1.A1.RB1.HE1	Seq10	Pump start priority
Alpha2.PS1.A1.RB1.HE1	Seq11	Pump start logic

The high-pressure oil system is composed of two pumps and a tank. Two oil level markers show the threshold points for the low-level indicators. Figure 65 shows a graphical representation of the high-pressure oil pumping unit.

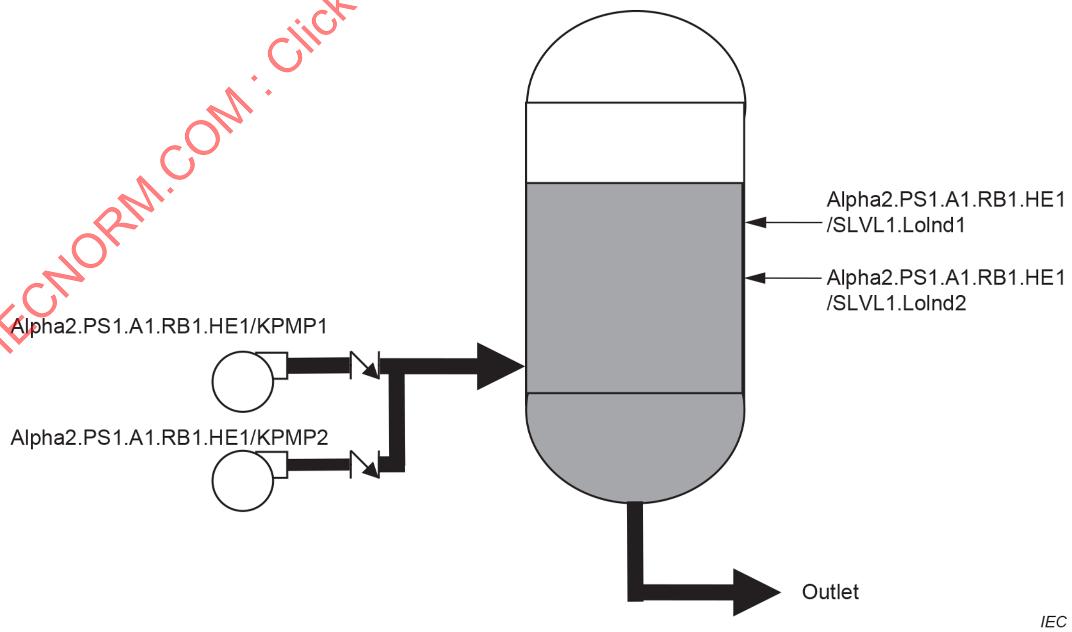


Figure 65 – Graphical representation of the high-pressure oil pumping unit

9.13.2 Sequence to manage a pump start priorities

The goal of this sequence is to manage the pump start priorities. This sequence can be used to start other types of equipment such as fans or heaters. Note that when a Boolean Oper.ctlVal is set true or false, its return value and readable value is stVal for the same data object instance.

The sequence may be broken down into the following steps:

- Step 0: This is the initial step. In this case, the step is active only when the program is initiated (after a soft reset, a reboot, or a forced re-initiation of the sequencer). In this step, the following data attributes of logical node " Alpha2.PS1.A1.RB1.HE1/FXPS1" are set:
  - CtlMod.stVal, set at 6 (Alternate + First In First Out),
  - StndStrQu1.stVal which is the queue order in which pump 1 is located,
  - StndStrQu2.stVal which is the queue order in which pump 2 is located.
- Step 1: This step stops all pumps, synchronises the pump start priority with respect to pumps location in the Queue, and waits for a low oil level.
- Step 2: This step awaits a selection confirmation for the pump that has the priority to start.
- Step 3 or 4: If " Alpha2.PS1.A1.RB1.HE1/FXPS1.StrPrt1.stVal = 1" then step 3 will become active and Pump 1 will receive an order to start. The Queue order for the priority start will then be inverted. If " Alpha2.PS1.A1.RB1.HE1/FXPS1.StrPrt2.stVal = 1" then step 4 will become active and Pump 2 will receive an order to start. The Queue order for the priority start will then be inverted.
- Step 5: This step selects branch according two conditions. If the low oil level condition disappears, the next active step will be step 1 and thus the pumps will receive a stop order. If a lower oil level condition appears, the sequence will branch to the next active step depending on the start priority of each pump.
- Step 6 or 8: These steps will send a start order to Pump 1 when in step 6 or a start order to Pump 2 when in step 8. These steps wait for the lower level condition to disappear before being inactivated.
- Step 7 or 9: These steps will send a stop order to Pump 1 when in step 6 or a stop order to Pump 2 when in step 8 and the sequence will go to back to step 5.

A logic sequence of these steps is shown in Figure 66.

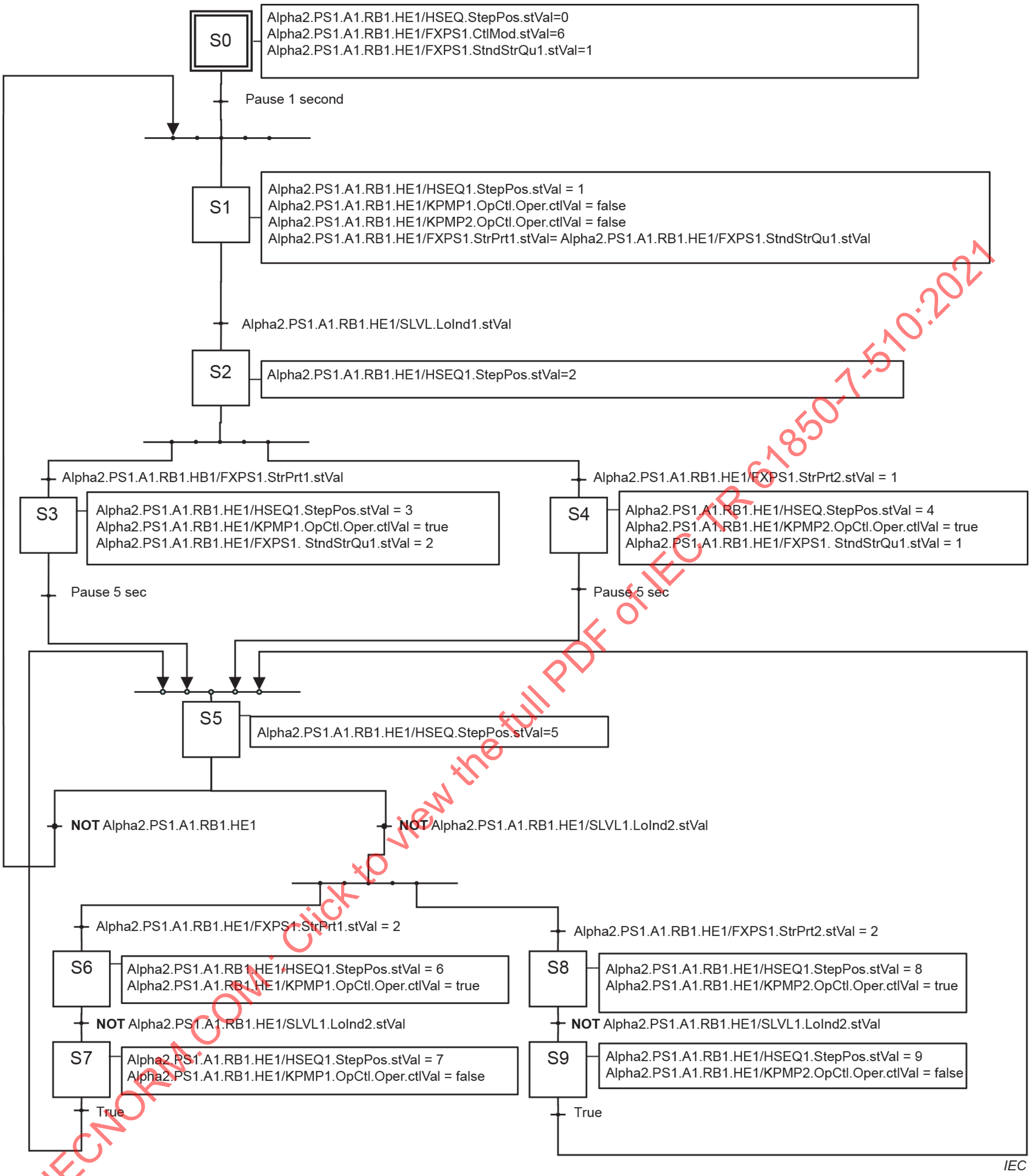


Figure 66 – Example of pump priority start logic sequence

9.13.3 Sequence to manage a pump

The goal of this sequence is to manage the pump. This sequence can be used to start other types of equipment such as fans or heaters.

The sequence may be broken down into the following steps:

- Step 0: This is the initial step. In this case, the step is active only when the program is initiated (after a soft reset, a reboot, or a forced re-initiation of the sequencer). The value of Alpha2.PS1.A1.RB1.HE1/KPMP1.Oper.stVal is set equal to the position of the pump starter coming from the discrete input Alpha2.PS1.A1.RB1.HE1/GGIO1.Ind01.stVal.
- Step 1: When the system is not in local and the sequence receives a stop order, the discrete output Alpha2.PS1.A1.RB1.HE1/GGIO1.SPCSO1.ctlVal is set to true for 0.5 s to stop the pump.
- Step 2: When the system is not in local and the sequence receives a start order, the discrete output Alpha2.PS1.A1.RB1.HE1/GGIO2.SPCSO2.ctlVal is set to true for 0.5 s to start the pump.

Figure 67 shows an example of pump start logic sequence.

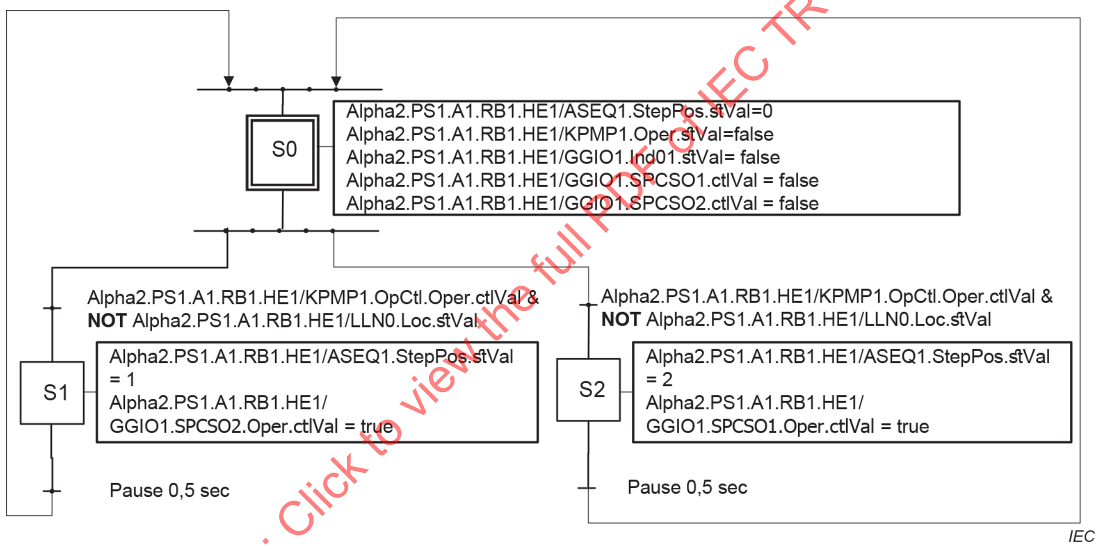


Figure 67 – Example of pump start logic sequence

9.14 Examples of how to use various types of curves and curve shape descriptions

This subclause shows various ways of using curve shape descriptions to describe a relationship of values within at a turbine generating unit level.

In Figure 68, we can see an example of a 2-dimensional curve used for shaping a flow value based on the gate position. The values entered in the table are based on statistical data obtained following a series of homologous model tests at a certain head level.

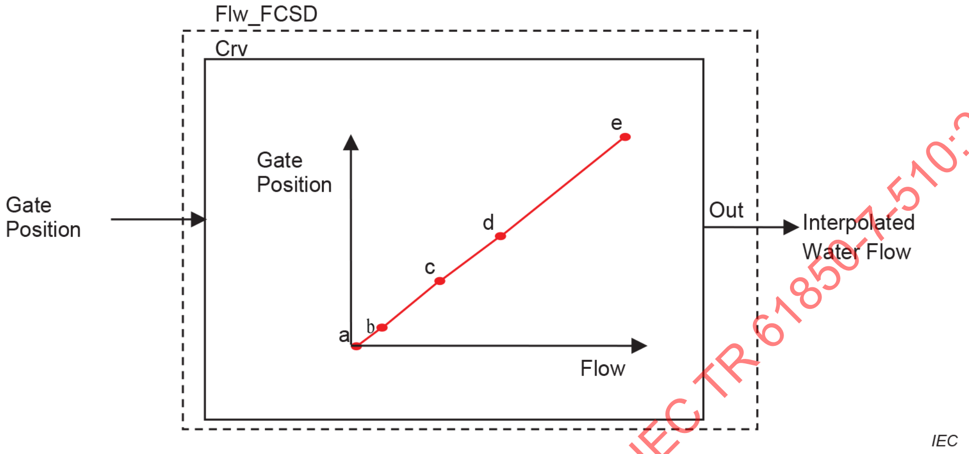


Figure 68 – Gate flow correlation

In Figure 69, we can see an example of a 3-dimensional curve used for shaping a runner blade position based on two variables, the net head and the gate position. To achieve such a function, the HCOM logical node is used. In the example 15 XYZ coordinate points is used to define the curve. The coordinates of such points are based on statistical data obtained following a series of index tests.

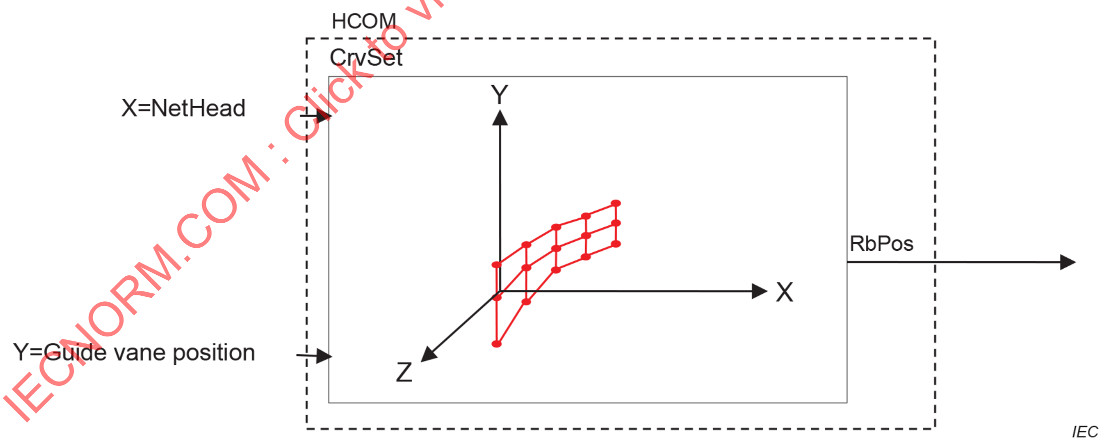


Figure 69 – Turbine correlation curve

9.15 Examples of voltage matching function

Traditionally, voltage matching pulses are sent from the synchroniser to the AVR. The length of the pulse is proportional to the actual voltage difference ΔU. Figure 70 shows an example of traditional voltage adjusting pulses.

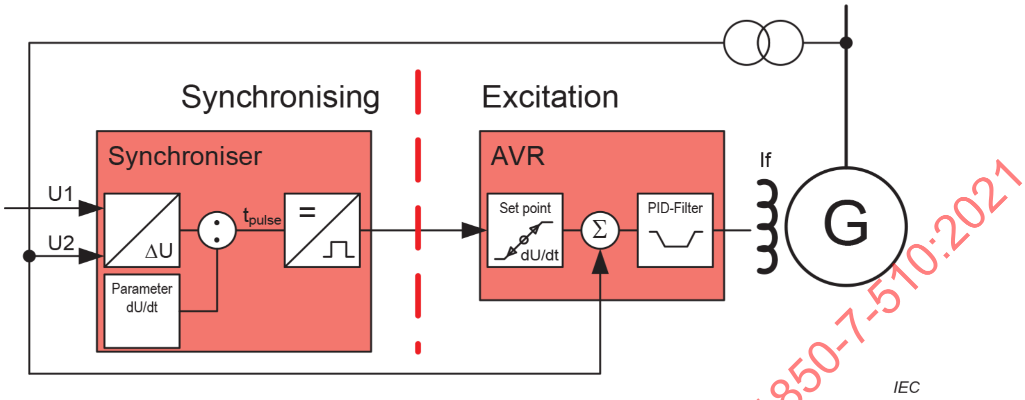


Figure 70 – Example of traditional voltage adjusting pulses

The voltage adjusting pulses may be mapped as follows (see Figure 71):

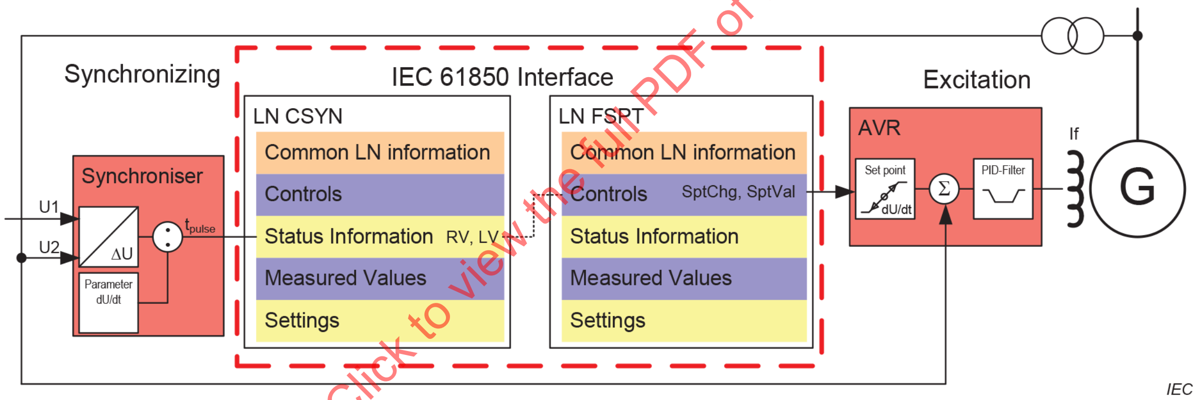


Figure 71 – Example of mapping of the pulse time in IEC 61850

Alternatively, the adjusting command may be used directly, e.g., without using the pulse time calculation (see Figure 72):

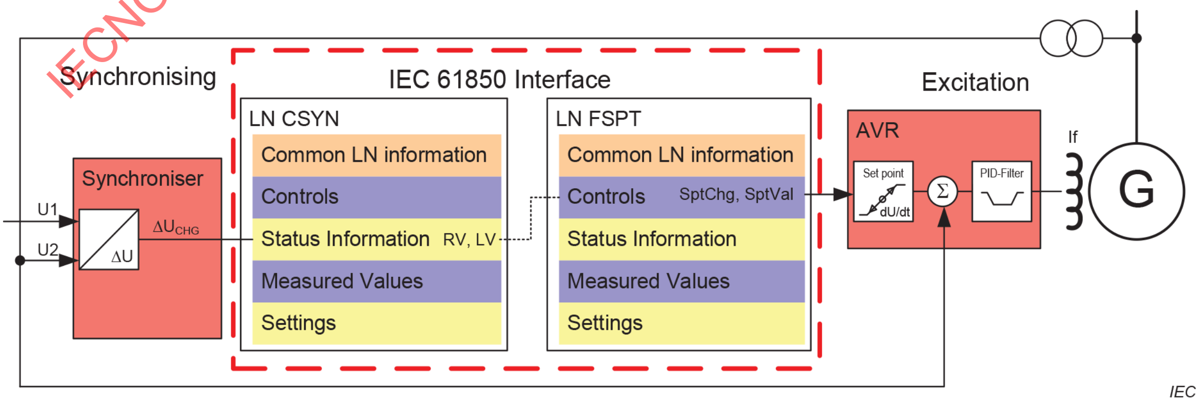


Figure 72 – Example of an IEC 61850 voltage adjusting command

Annex A
(informative)

Electrical single line diagrams of thermal power plants

There are different configurations possible that are reflected in the corresponding single line diagrams. For the auxiliary supply there is usually a transformer connected between the generator and the step-up transformer. This transformer provides usually as well the start-up power, but there are as well configurations possible, where the start-up power is coming from a different medium voltage grid.

Figure A.1 shows a typical single line diagram for a steam turbine power plant.

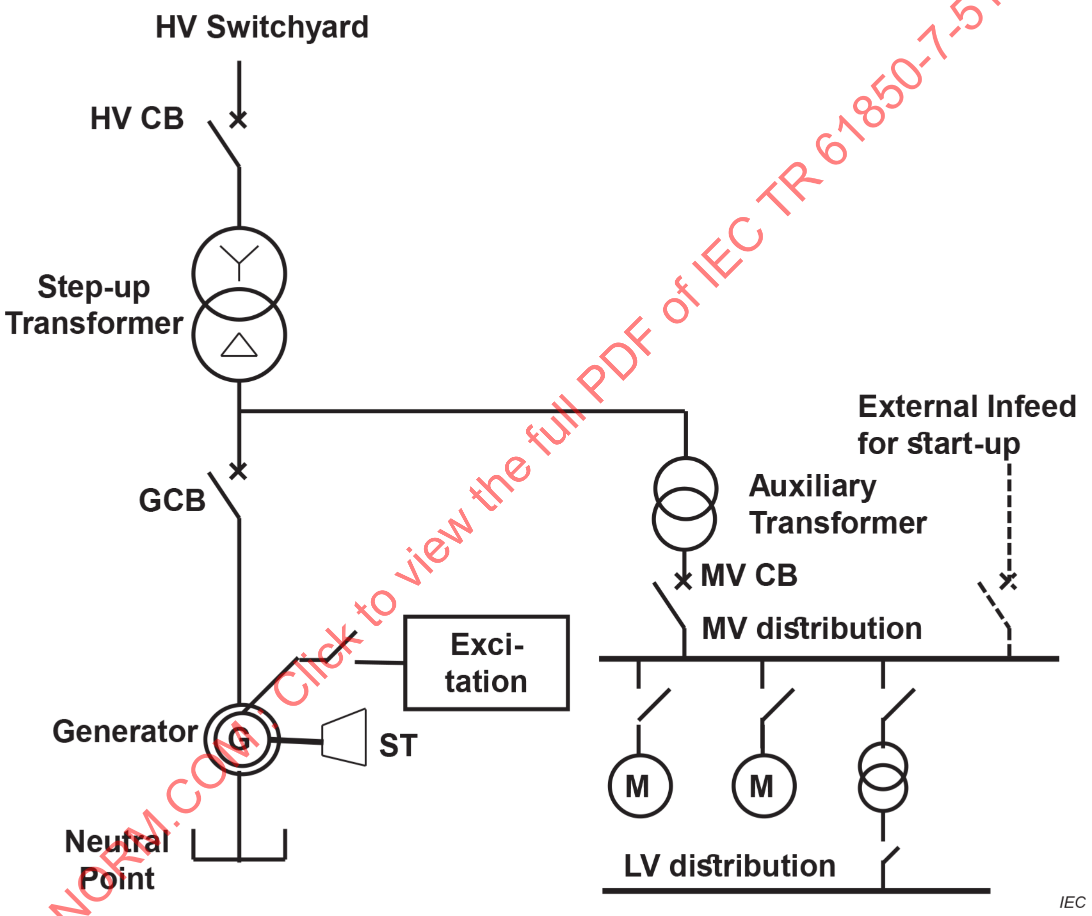


Figure A.1 – Typical Single Line Diagram of a steam turbine power plant

Figure A.2 shows a typical single line diagram for a gas turbine power plant in simple cycle or a combined cycle power plant in single shaft configuration. For start-up, the generator is used as a motor, supplied by a static frequency converter (static starting device). It may be supplied from the medium voltage board, or as shown in Figure A.2 by a dedicated winding of a 3-winding auxiliary transformer. In combined cycle single-shaft configuration the steam turbine is connected to the shaft after synchronizing the gas turbine by a self-synchronizing clutch.

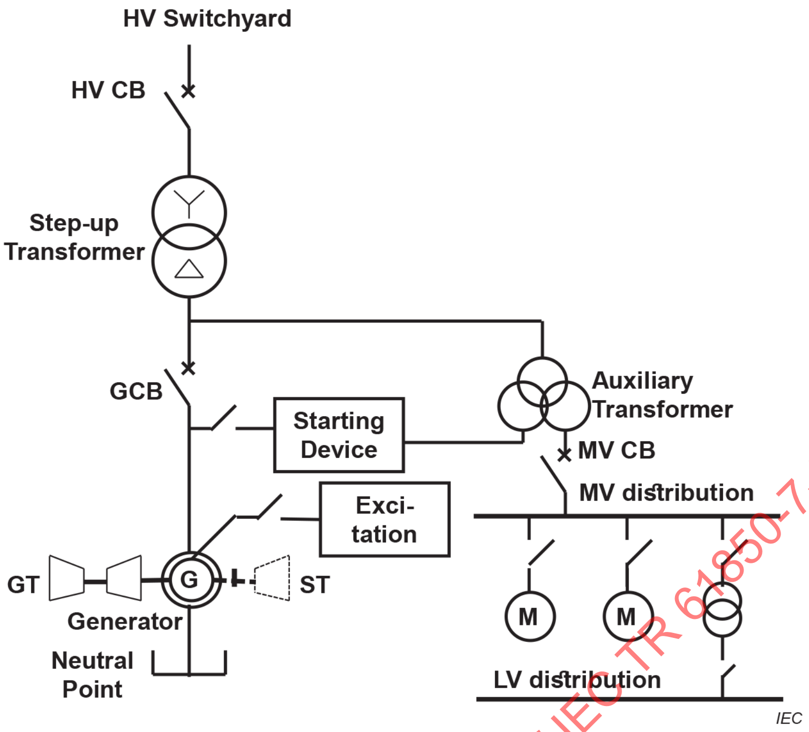


Figure A.2 – Typical Single Line Diagram of a gas turbine power plant or a combined cycle power plant in single shaft configuration

Figure A.3 and Figure A.**4** show typical single line diagrams for multi-shaft combined cycle power plants.

In Figure A.3 the gas turbine and the steam turbine have their own step-up transformers. The auxiliaries of both turbines are supplied by an auxiliary transformer connected to the gas turbine power train, and the steam turbine has no dedicated auxiliary transformer. The steam turbine is synchronized by the high voltage circuit breaker, and no generator breaker is required. In a configuration with two gas turbines and one steam turbine, each gas turbine has usually its own auxiliary transformer and some switching over on the auxiliary distribution is possible, since the steam turbine auxiliaries are distributed to the distribution boards of both gas turbines.

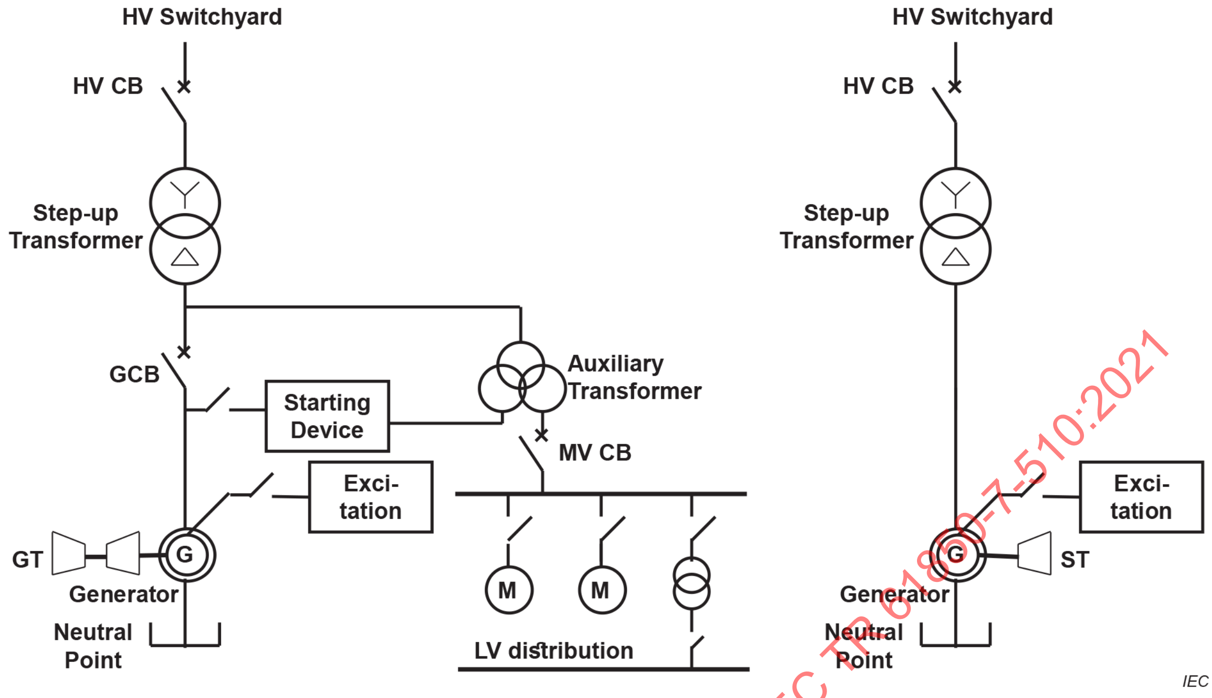


Figure A.3 – Typical Single Line Diagram of a combined cycle power plant in multi-shaft configuration with separate step-up transformers

In Figure A.**4** both turbines have a common 3-winding step-up transformer. This configuration is only possible for one gas turbine with one steam turbine. The advantage is that only one bay at the high voltage switchyard is required for the multi-shaft power plant, but a generator circuit breaker at the steam turbine is mandatory.

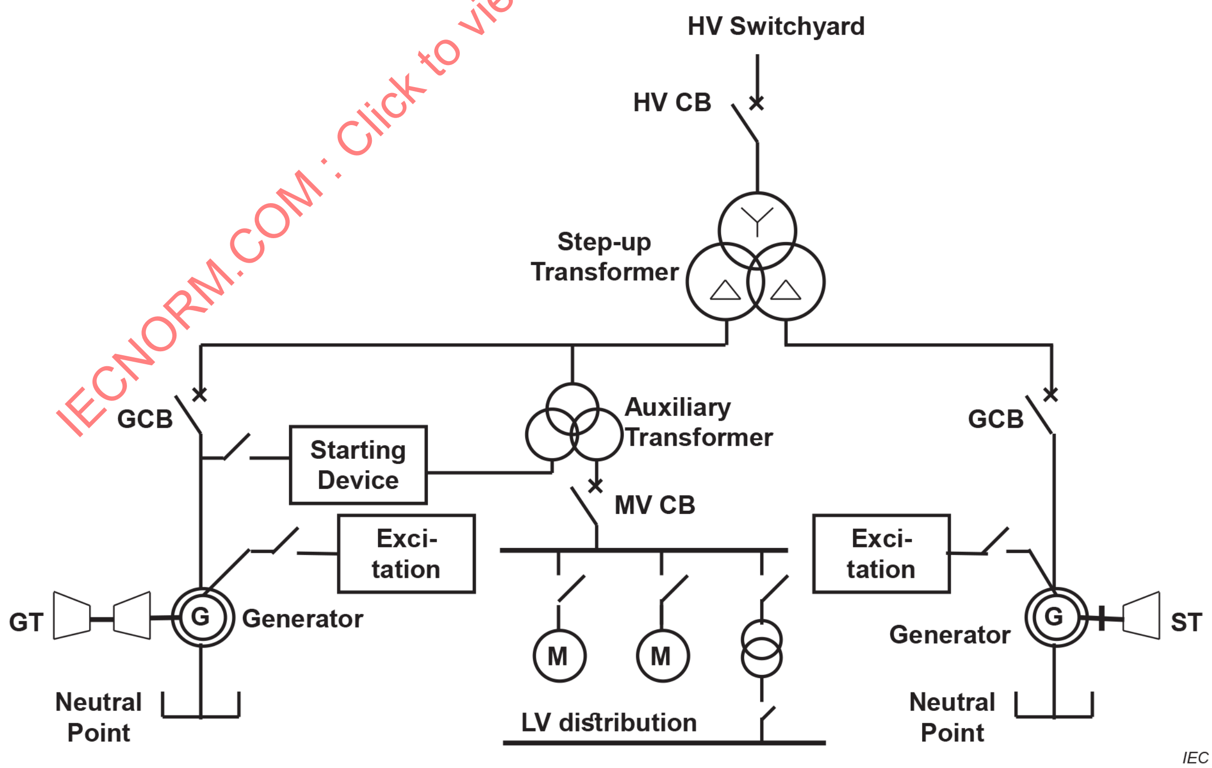


Figure A.4 – Typical Single Line Diagram of a combined cycle power plant in multi-shaft configuration with 3-winding step-up transformers

Annex B
(informative)

System Specification Description for the Alpha 2 power plant

This annex provides the system specification description, as a code component, using the SCL from IEC 61850-6 for the cited Alpha 2 power plant examples in this document.

The process element breakdown is used to describe the assets using the RDS designation system. The data template section applies to the nomenclature found in IEC 61850-7-3, 7-4 and 7-410. The function element is used to link the RDS structure to the IEC 61850 data templates.

<CODE BEGINS>

```
<?xml version="1.0" encoding="UTF-8"?>
<SCL xmlns="http://www.iec.ch/61850/2003/SCL" revision="B" version="2007" release="4">
 <Header id="IEC-7-510B" nameStructure="IEDName" revision="R000" version="V0.0">
  <History/>
 </Header>
 <Communication/>
 <DataTypeTemplates>
  <LNodeType id="SIGMA_AJCL_MVar_2020A" lnClass="AJCL">
   <DO name="NamPlt" type="SIGMA_LPL_HYDNs_2020A" transient="false"/>
   <DO name="Mod" type="SIGMA_ENC_Mod_2020A" transient="false"/>
   <DO name="Beh" type="SIGMA_ENS_Beh_2020A" transient="false"/>
   <DO name="Unt1" type="SIGMA_SPS_2020A" transient="false"/>
   <DO name="Unt2" type="SIGMA_SPS_2020A" transient="false"/>
   <DO name="PwrOut" type="SIGMA_MV_MVar_2020A" transient="false"/>
   <DO name="PwrOutTot" type="SIGMA_MV_MVar_2020A" transient="false"/>
   <DO name="JCtlTag" type="SIGMA_TAG_2020A" transient="false"/>
   <DO name="UntTag1" type="SIGMA_TAG_2020A" transient="false"/>
   <DO name="UntTag2" type="SIGMA_TAG_2020A" transient="false"/>
   <DO name="OpCnt" type="SIGMA_INS_2020A" transient="false"/>
   <DO name="Loc" type="SIGMA_SPS_2020A" transient="false"/>
   <DO name="LocKey" type="SIGMA_SPS_2020A" transient="false"/>
   <DO name="LocSta" type="SIGMA_SPC_2020A" transient="false"/>
   <DO name="ModAct" type="SIGMA_SPC_2020A" transient="false"/>
  </LNodeType>
  <LNodeType id="SIGMA_FSPT_MVAR_2020A" lnClass="FSPT">
   <DO name="Mod" type="SIGMA_ENC_Mod_2020A" transient="false"/>
   <DO name="Beh" type="SIGMA_ENS_Beh_2020A" transient="false"/>
   <DO name="Health" type="SIGMA_ENS_Health_2020A" transient="false"/>
```

```
  <DO desc="Setpoint" name="SptVal" type="SIGMA_APC_2020A" transient="false"/>
  <DO name="SptEndSt" type="SIGMA_ENS_SptEndSt_2020A" transient="false"/>
  <DO name="AdjSt" type="SIGMA_ENS_AdjSt_2020A" transient="false"/>
  <DO name="SptMem" type="SIGMA_MV_2020A" transient="false"/>
  <DO name="ErrTerm" type="SIGMA_MV_2020A" transient="false"/>
  <DO name="Out" type="SIGMA_MV_2020A" transient="false"/>
  <DO name="MaxRst" type="SIGMA_ASG_2020A" transient="false"/>
  <DO name="MinRst" type="SIGMA_ASG_2020A" transient="false"/>
  <DO name="Auto" type="SIGMA_SPC_2020A" transient="false"/>
  <DO name="SptDvAlm" type="SIGMA_SPS_2020A" transient="false"/>
  <DO name="DvAlm" type="SIGMA_ASG_2020A" transient="false"/>
  <DO name="SptUp" type="SIGMA_SPS_2020A" transient="false"/>
  <DO name="SptDn" type="SIGMA_SPS_2020A" transient="false"/>
  <DO name="SptDir" type="SIGMA_SPS_2020A" transient="false"/>
  <DO name="Blk" type="SIGMA_SPS_2020A" transient="false"/>
</LNodeType>
<LNodeType id="SIGMA_FSPT_KV_2020A" lnClass="FSPT">
  <DO name="Mod" type="SIGMA_ENC_Mod_2020A" transient="false"/>
  <DO name="Beh" type="SIGMA_ENS_Beh_2020A" transient="false"/>
  <DO name="Health" type="SIGMA_ENS_Health_2020A" transient="false"/>
  <DO name="SptVal" type="SIGMA_APC_KV_2020A" transient="false"/>
  <DO name="SptEndSt" type="SIGMA_ENS_SptEndSt_2020A" transient="false"/>
  <DO name="AdjSt" type="SIGMA_ENS_AdjSt_2020A" transient="false"/>
  <DO name="SptMem" type="SIGMA_MV_KV_2020A" transient="false"/>
  <DO name="ErrTerm" type="SIGMA_MV_KV_2020A" transient="false"/>
  <DO name="Out" type="SIGMA_MV_KV_2020A" transient="false"/>
  <DO name="MaxRst" type="SIGMA_ASG_KV_2020A" transient="false"/>
  <DO name="MinRst" type="SIGMA_ASG_KV_2020A" transient="false"/>
  <DO name="Auto" type="SIGMA_SPC_2020A" transient="false"/>
  <DO name="SptDvAlm" type="SIGMA_SPS_2020A" transient="false"/>
  <DO name="DvAlm" type="SIGMA_ASG_KV_2020A" transient="false"/>
  <DO name="SptUp" type="SIGMA_SPS_2020A" transient="false"/>
  <DO name="SptDn" type="SIGMA_SPS_2020A" transient="false"/>
  <DO name="SptDir" type="SIGMA_SPS_2020A" transient="false"/>
  <DO name="Blk" type="SIGMA_SPS_2020A" transient="false"/>
</LNodeType>
<LNodeType id="SIGMA_FSPT_PF_2020A" lnClass="FSPT">
```

```xml
  <DO name="Mod" type="SIGMA_ENC_Mod_2020A" transient="false"/>
  <DO name="Beh" type="SIGMA_ENS_Beh_2020A" transient="false"/>
  <DO name="Health" type="SIGMA_ENS_Health_2020A" transient="false"/>
  <DO name="SptVal" type="SIGMA_APC_2020A" transient="false"/>
  <DO name="SptEndSt" type="SIGMA_ENS_SptEndSt_2020A" transient="false"/>
  <DO name="AdjSt" type="SIGMA_ENS_AdjSt_2020A" transient="false"/>
  <DO name="SptMem" type="SIGMA_MV_2020A" transient="false"/>
  <DO name="ErrTerm" type="SIGMA_MV_2020A" transient="false"/>
  <DO name="Out" type="SIGMA_MV_2020A" transient="false"/>
  <DO name="MaxRst" type="SIGMA_ASG_2020A" transient="false"/>
  <DO name="MinRst" type="SIGMA_ASG_2020A" transient="false"/>
  <DO name="Auto" type="SIGMA_SPC_2020A" transient="false"/>
  <DO name="SptDvAlm" type="SIGMA_SPS_2020A" transient="false"/>
  <DO name="DvAlm" type="SIGMA_ASG_2020A" transient="false"/>
  <DO name="SptUp" type="SIGMA_SPS_2020A" transient="false"/>
  <DO name="SptDn" type="SIGMA_SPS_2020A" transient="false"/>
  <DO name="SptDir" type="SIGMA_SPS_2020A" transient="false"/>
  <DO name="Blk" type="SIGMA_SPS_2020A" transient="false"/>
</LNodeType>
<LNodeType id="SIGMA_AJCL_MW_2020A" lnClass="AJCL">
  <DO name="NamPlt" type="SIGMA_LPL_HYDNs_2020A" transient="false"/>
  <DO name="Mod" type="SIGMA_ENC_Mod_2020A" transient="false"/>
  <DO name="Beh" type="SIGMA_ENS_Beh_2020A" transient="false"/>
  <DO name="Unt1" type="SIGMA_SPS_2020A" transient="false"/>
  <DO name="Unt2" type="SIGMA_SPS_2020A" transient="false"/>
  <DO name="PwrOut" type="SIGMA_MV_MW_2020A" transient="false"/>
  <DO name="PwrOutTot" type="SIGMA_MV_MW_2020A" transient="false"/>
  <DO name="JCtlTag" type="SIGMA_TAG_2020A" transient="false"/>
  <DO name="UntTag1" type="SIGMA_TAG_2020A" transient="false"/>
  <DO name="UntTag2" type="SIGMA_TAG_2020A" transient="false"/>
  <DO name="OpCnt" type="SIGMA_INS_2020A" transient="false"/>
  <DO name="Loc" type="SIGMA_SPS_2020A" transient="false"/>
  <DO name="LocKey" type="SIGMA_SPS_2020A" transient="false"/>
  <DO name="LocSta" type="SIGMA_SPC_2020A" transient="false"/>
  <DO name="ModAct" type="SIGMA_SPC_2020A" transient="false"/>
</LNodeType>
<LNodeType id="SIGMA_FSPT_MW_2020A" lnClass="FSPT">
```

```xml
  <DO name="Mod" type="SIGMA_ENC_Mod_2020A" transient="false"/>
  <DO name="Beh" type="SIGMA_ENS_Beh_2020A" transient="false"/>
  <DO name="Health" type="SIGMA_ENS_Health_2020A" transient="false"/>
  <DO name="SptVal" type="SIGMA_APC_MW_2020A" transient="false"/>
  <DO name="SptEndSt" type="SIGMA_ENS_SptEndSt_2020A" transient="false"/>
  <DO name="AdjSt" type="SIGMA_ENS_AdjSt_2020A" transient="false"/>
  <DO name="SptMem" type="SIGMA_MV_MW_2020A" transient="false"/>
  <DO name="ErrTerm" type="SIGMA_MV_MW_2020A" transient="false"/>
  <DO name="Out" type="SIGMA_MV_MW_2020A" transient="false"/>
  <DO name="MaxRst" type="SIGMA_ASG_MW_2020A" transient="false"/>
  <DO name="MinRst" type="SIGMA_ASG_MW_2020A" transient="false"/>
  <DO name="Auto" type="SIGMA_SPC_2020A" transient="false"/>
  <DO name="SptDvAlm" type="SIGMA_SPS_2020A" transient="false"/>
  <DO name="DvAlm" type="SIGMA_ASG_MW_2020A" transient="false"/>
  <DO name="SptUp" type="SIGMA_SPS_2020A" transient="false"/>
  <DO name="SptDn" type="SIGMA_SPS_2020A" transient="false"/>
  <DO name="SptDir" type="SIGMA_SPS_2020A" transient="false"/>
  <DO name="Blk" type="SIGMA_SPS_2020A" transient="false"/>
</LNodeType>
<LNodeType id="SIGMA_MMXU_MV_2020A" lnClass="MMXU">
  <DO name="PPV" type="SIGMA_DEL_2020A" transient="false"/>
  <DO name="Mod" type="SIGMA_ENC_Mod_2020A" transient="false"/>
  <DO name="Beh" type="SIGMA_ENS_Beh_2020A" transient="false"/>
  <DO name="Health" type="SIGMA_ENS_Health_2020A" transient="false"/>
  <DO name="NamPlt" type="SIGMA_LPL_2020A" transient="false"/>
  <DO name="TotW" type="SIGMA_MV_MW_2020A" transient="false"/>
  <DO name="TotVAr" type="SIGMA_MV_MVar_2020A" transient="false"/>
  <DO name="TotPF" type="SIGMA_MV_2020A" transient="false"/>
  <DO name="Hz" type="SIGMA_MV_Hz_2020A" transient="false"/>
  <DO name="AvPPVPhs" type="SIGMA_MV_KV_2020A" transient="false"/>
  <DO name="AvAPhs" type="SIGMA_MV_2020A" transient="false"/>
</LNodeType>
<LNodeType id="SIGMA_FCSD_OpeningWattHead_2020A" lnClass="FCSD">
  <DO name="Beh" type="SIGMA_ENS_Beh_2020A" transient="false"/>
  <DO name="Crv" type="SIGMA_CSG_OpeningWattHead_2020A" transient="false"/>
  <DO name="Out" type="SIGMA_MV_MW_2020A" transient="false"/>
</LNodeType>
```

```xml
<LNodeType id="SIGMA_HJCL_Flw_2020A" lnClass="HJCL">
  <DO name="NamPlt" type="SIGMA_LPL_HYDNs_2020A" transient="false"/>
  <DO name="Mod" type="SIGMA_ENC_Mod_2020A" transient="false"/>
  <DO name="Beh" type="SIGMA_ENS_Beh_2020A" transient="false"/>
  <DO name="Unt1" type="SIGMA_SPS_2020A" transient="false"/>
  <DO name="Unt2" type="SIGMA_SPS_2020A" transient="false"/>
  <DO name="ClcFlw" type="SIGMA_MV_Flw_2020A" transient="false"/>
  <DO name="ClcTotFlw" type="SIGMA_MV_Flw_2020A" transient="false"/>
  <DO name="JCtlTag" type="SIGMA_TAG_2020A" transient="false"/>
  <DO name="ClcFlwSpt1" type="SIGMA_MV_Flw_2020A" transient="false"/>
  <DO name="ClcFlwSpt2" type="SIGMA_MV_Flw_2020A" transient="false"/>
  <DO name="ClcGteSpt1" type="SIGMA_MV_metre_2020A" transient="false"/>
  <DO name="ClcPwrSpt1" type="SIGMA_MV_MW_2020A" transient="false"/>
  <DO name="ClcPwrSpt2" type="SIGMA_MV_MW_2020A" transient="false"/>
  <DO name="ClcGteSpt2" type="SIGMA_MV_metre_2020A" transient="false"/>
  <DO name="UntTag1" type="SIGMA_TAG_2020A" transient="false"/>
  <DO name="UntTag2" type="SIGMA_TAG_2020A" transient="false"/>
  <DO name="OpCnt" type="SIGMA_INS_2020A" transient="false"/>
  <DO name="Loc" type="SIGMA_SPS_2020A" transient="false"/>
  <DO name="LocKey" type="SIGMA_SPS_2020A" transient="false"/>
  <DO name="LocSta" type="SIGMA_SPC_2020A" transient="false"/>
  <DO name="ModAct" type="SIGMA_SPC_2020A" transient="false"/>
</LNodeType>
<LNodeType id="SIGMA_FSPT_Flw_2020A" lnClass="FSPT">
  <DO name="Mod" type="SIGMA_ENC_Mod_2020A" transient="false"/>
  <DO name="Beh" type="SIGMA_ENS_Beh_2020A" transient="false"/>
  <DO name="Health" type="SIGMA_ENS_Health_2020A" transient="false"/>
  <DO name="SptVal" type="SIGMA_APC_Amp_2020A" transient="false"/>
  <DO name="SptEndSt" type="SIGMA_ENS_SptEndSt_2020A" transient="false"/>
  <DO name="AdjSt" type="SIGMA_ENS_AdjSt_2020A" transient="false"/>
  <DO name="SptMem" type="SIGMA_MV_Flw_2020A" transient="false"/>
  <DO name="ErrTerm" type="SIGMA_MV_Flw_2020A" transient="false"/>
  <DO name="Out" type="SIGMA_MV_Flw_2020A" transient="false"/>
  <DO name="MaxRst" type="SIGMA_ASG_Flw_2020A" transient="false"/>
  <DO name="MinRst" type="SIGMA_ASG_Flw_2020A" transient="false"/>
  <DO name="Auto" type="SIGMA_SPC_2020A" transient="false"/>
  <DO name="SptDvAlm" type="SIGMA_SPS_2020A" transient="false"/>
```

```
    <DO name="DvAlm" type="SIGMA_ASG_Flw_2020A" transient="false"/>
    <DO name="SptUp" type="SIGMA_SPS_2020A" transient="false"/>
    <DO name="SptDn" type="SIGMA_SPS_2020A" transient="false"/>
    <DO name="SptDir" type="SIGMA_SPS_2020A" transient="false"/>
    <DO name="Blk" type="SIGMA_SPS_2020A" transient="false"/>
  </LNodeType>
  <LNodeType id="SIGMA_FSPT_Lev_2020A" lnClass="FSPT">
    <DO name="Mod" type="SIGMA_ENC_Mod_2020A" transient="false"/>
    <DO name="Beh" type="SIGMA_ENS_Beh_2020A" transient="false"/>
    <DO name="Health" type="SIGMA_ENS_Health_2020A" transient="false"/>
    <DO name="SptVal" type="SIGMA_APC_metre_2020A" transient="false"/>
    <DO name="SptEndSt" type="SIGMA_ENS_SptEndSt_2020A" transient="false"/>
    <DO name="AdjSt" type="SIGMA_ENS_AdjSt_2020A" transient="false"/>
    <DO name="SptMem" type="SIGMA_MV_metre_2020A" transient="false"/>
    <DO name="ErrTerm" type="SIGMA_MV_metre_2020A" transient="false"/>
    <DO name="Out" type="SIGMA_MV_metre_2020A" transient="false"/>
    <DO name="MaxRst" type="SIGMA_ASG_metre_2020A" transient="false"/>
    <DO name="MinRst" type="SIGMA_ASG_metre_2020A" transient="false"/>
    <DO name="Auto" type="SIGMA_SPC_2020A" transient="false"/>
    <DO name="SptDvAlm" type="SIGMA_SPS_2020A" transient="false"/>
    <DO name="DvAlm" type="SIGMA_ASG_metre_2020A" transient="false"/>
    <DO name="SptUp" type="SIGMA_SPS_2020A" transient="false"/>
    <DO name="SptDn" type="SIGMA_SPS_2020A" transient="false"/>
    <DO name="SptDir" type="SIGMA_SPS_2020A" transient="false"/>
    <DO name="Blk" type="SIGMA_SPS_2020A" transient="false"/>
  </LNodeType>
  <LNodeType id="SIGMA_HLVL_2020A" lnClass="HLVL">
    <DO name="NamPlt" type="SIGMA_LPL_HYDNs_2020A" transient="false"/>
    <DO name="Mod" type="SIGMA_ENC_Mod_2020A" transient="false"/>
    <DO name="Beh" type="SIGMA_ENS_Beh_2020A" transient="false"/>
    <DO name="Health" type="SIGMA_ENS_Health_2020A" transient="false"/>
    <DO name="EEHealth" type="SIGMA_ENS_Health_2020A" transient="false"/>
    <DO name="EEName" type="SIGMA_DPL_2020A" transient="false"/>
    <DO name="OpCnt" type="SIGMA_INS_2020A" transient="false"/>
    <DO name="Loc" type="SIGMA_SPS_2020A" transient="false"/>
    <DO name="Stuck" type="SIGMA_SPS_2020A" transient="false"/>
    <DO name="LevOfs" type="SIGMA_ASG_2020A" transient="false"/>
```

```
  <DO name="LevM" type="SIGMA_MV_2020A" transient="false"/>
</LNodeType>
<LNodeType id="SIGMA_MHYD_2020A" lnClass="MHYD">
  <DO name="Beh" type="SIGMA_ENS_Beh_2020A" transient="false"/>
  <DO name="Health" type="SIGMA_ENS_Health_2020A" transient="false"/>
  <DO name="Mod" type="SIGMA_ENC_Mod_2020A" transient="false"/>
  <DO name="Lev" type="SIGMA_MV_metre_2020A" transient="false"/>
  <DO name="Flw" type="SIGMA_MV_Flw_2020A" transient="false"/>
  <DO name="Tmp" type="SIGMA_MV_Tmp_2020A" transient="false"/>
  <DO name="FishCnt" type="SIGMA_BCR_2020A" transient="false"/>
</LNodeType>
<LNodeType id="SIGMA_TLVL_2020A" lnClass="TLVL">
  <DO name="Beh" type="SIGMA_ENS_Beh_2020A" transient="false"/>
  <DO name="Health" type="SIGMA_ENS_Health_2020A" transient="false"/>
  <DO name="Mod" type="SIGMA_ENC_Mod_2020A" transient="false"/>
  <DO name="SmpRte" type="SIGMA_ING_2020A" transient="false"/>
  <DO name="HoldTmms" type="SIGMA_ING_2020A" transient="false"/>
  <DO name="LevPctSv" type="SIGMA_SAV_2020A" transient="false"/>
</LNodeType>
<LNodeType id="SIGMA_HWCL_2020A" lnClass="HWCL">
  <DO name="NamPlt" type="SIGMA_LPL_HYDNs_2020A" transient="false"/>
  <DO name="LocKey" type="SIGMA_SPS_2020A" transient="false"/>
  <DO name="Loc" type="SIGMA_SPS_2020A" transient="false"/>
  <DO name="FlwMax" type="SIGMA_SPS_2020A" transient="false"/>
  <DO name="FlwMin" type="SIGMA_SPS_2020A" transient="false"/>
  <DO name="HiLevUp" type="SIGMA_SPS_2020A" transient="false"/>
  <DO name="LoLevUp" type="SIGMA_SPS_2020A" transient="false"/>
  <DO name="HiLevDn" type="SIGMA_SPS_2020A" transient="false"/>
  <DO name="LoLevDn" type="SIGMA_SPS_2020A" transient="false"/>
  <DO name="FlwLevAlm" type="SIGMA_SPS_2020A" transient="false"/>
  <DO name="FlwMaxLim" type="SIGMA_ASG_2020A" transient="false"/>
  <DO name="FlwMinLim" type="SIGMA_ASG_2020A" transient="false"/>
  <DO name="LevHiSpt" type="SIGMA_ASG_metre_2020A" transient="false"/>
  <DO name="LevLoSpt" type="SIGMA_ASG_metre_2020A" transient="false"/>
  <DO name="LevDnHiSpt" type="SIGMA_ASG_metre_2020A" transient="false"/>
  <DO name="LevDnLoSpt" type="SIGMA_ASG_metre_2020A" transient="false"/>
  <DO name="Flw" type="SIGMA_MV_2020A" transient="false"/>
```

```xml
    <DO name="FlwPct" type="SIGMA_MV_2020A" transient="false"/>
    <DO name="PskPres" type="SIGMA_MV_MPa_2020A" transient="false"/>
    <DO name="LocSta" type="SIGMA_SPC_2020A" transient="false"/>
    <DO name="Auto" type="SIGMA_SPC_2020A" transient="false"/>
    <DO name="HdrCtlMod" type="SIGMA_ENC_HdrCtlMod_2020A" transient="false"/>
    <DO name="FlwSpt" type="SIGMA_APC_2020A" transient="false"/>
    <DO name="LevSpt" type="SIGMA_APC_metre_2020A" transient="false"/>
    <DO name="PosChg" type="SIGMA_BSC_PosChg_2020A" transient="false"/>
    <DO name="PosChgIncr" type="SIGMA_INC_2020A" transient="false"/>
    <DO name="ActPwrR" type="SIGMA_SPC_2020A" transient="false"/>
    <DO name="ActPwrL" type="SIGMA_SPC_2020A" transient="false"/>
    <DO name="Beh" type="SIGMA_ENS_Beh_2020A" transient="false"/>
    <DO name="Mod" type="SIGMA_ENC_Mod_2020A" transient="false"/>
  </LNodeType>
  <LNodeType id="SIGMA_TFLW_2020A" lnClass="TFLW">
    <DO name="Beh" type="SIGMA_ENS_Beh_2020A" transient="false"/>
    <DO name="Health" type="SIGMA_ENS_Health_2020A" transient="false"/>
    <DO name="Mod" type="SIGMA_ENC_Mod_2020A" transient="false"/>
    <DO name="SmpRte" type="SIGMA_ING_2020A" transient="false"/>
    <DO name="HoldTmms" type="SIGMA_ING_2020A" transient="false"/>
    <DO name="FlwSv" type="SIGMA_SAV_2020A" transient="false"/>
  </LNodeType>
  <LNodeType id="SIGMA_HNHD_2020A" lnClass="HNHD">
    <DO name="NamPlt" type="SIGMA_LPL_HYDNs_2020A" transient="false"/>
    <DO name="Stuck" type="SIGMA_SPS_2020A" transient="false"/>
    <DO name="LevOfs" type="SIGMA_ASG_2020A" transient="false"/>
    <DO name="NHd" type="SIGMA_MV_2020A" transient="false"/>
    <DO name="DifPres" type="SIGMA_MV_2020A" transient="false"/>
    <DO name="Beh" type="SIGMA_ENS_Beh_2020A" transient="false"/>
  </LNodeType>
  <LNodeType id="SIGMA_GUNT_2020A" lnClass="GUNT">
    <DO name="NamPlt" type="SIGMA_LPL_HYDNs_2020A" transient="false"/>
    <DO name="Mod" type="SIGMA_ENC_Mod_2020A" transient="false"/>
    <DO name="Beh" type="SIGMA_ENS_Beh_2020A" transient="false"/>
    <DO name="Health" type="SIGMA_ENS_Health_2020A" transient="false"/>
    <DO name="Loc" type="SIGMA_SPS_2020A" transient="false"/>
    <DO name="LocKey" type="SIGMA_SPS_2020A" transient="false"/>
```

```xml
  <DO name="Iner" type="SIGMA_INS_2020A" transient="false"/>
  <DO name="LocSta" type="SIGMA_SPC_2020A" transient="false"/>
  <DO name="OpCntRs" type="SIGMA_INC_2020A" transient="false"/>
  <DO name="GridMod" type="SIGMA_ENS_GridMod_2020A" transient="false"/>
  <DO name="UntOpMod" type="SIGMA_ENS_UntOpMod_2020A" transient="false"/>
  <DO name="GridOpSt" type="SIGMA_ENS_GridOpSt_2020A" transient="false"/>
  <DO name="PaOpnMod" type="SIGMA_SPS_2020A" transient="false"/>
  <DO name="LimAct" type="SIGMA_SPS_2020A" transient="false"/>
  <DO name="StopVlv" type="SIGMA_SPS_2020A" transient="false"/>
  <DO name="UntOpSt" type="SIGMA_ENS_UntOpSt_2020A" transient="false"/>
  <DO name="ExSptEna" type="SIGMA_SPC_2020A" transient="false"/>
  <DO name="PwrRtgLim" type="SIGMA_RST_2020A" transient="false"/>
  <DO name="VRtgLim" type="SIGMA_RST_2020A" transient="false"/>
  <DO name="ReqSt" type="SIGMA_ENC_ReqSt_2020A" transient="false"/>
  <DO name="StepOp" type="SIGMA_SPC_2020A" transient="false"/>
  <DO name="StrNxt" type="SIGMA_SPC_2020A" transient="false"/>
  <DO name="Tag" type="SIGMA_TAG_2020A" transient="false"/>
</LNodeType>
<LNodeType id="SIGMA_CSWI_2020A" lnClass="CSWI">
  <DO desc="Operation Open Switch" name="OpOpn" type="SIGMA_ACT_2020A" transient="false"/>
  <DO desc="Switch, general" name="Pos" type="SIGMA_DPC_2020A" transient="false"/>
  <DO desc="Switch L1" name="PosA" type="SIGMA_DPC_2020A" transient="false"/>
  <DO desc="Switch L2" name="PosB" type="SIGMA_DPC_2020A" transient="false"/>
  <DO desc="Switch L3" name="PosC" type="SIGMA_DPC_2020A" transient="false"/>
  <DO name="Mod" type="SIGMA_ENC_Mod_2020A" transient="false"/>
  <DO desc="Resetable operation counter" name="OpCntRs" type="SIGMA_INC_2020A" transient="false"/>
  <DO name="Beh" type="SIGMA_ENS_Beh_2020A" transient="false"/>
  <DO name="Health" type="SIGMA_ENS_Health_2020A" transient="false"/>
  <DO name="NamPlt" type="SIGMA_LPL_2020A" transient="false"/>
  <DO desc="Local operation" name="Loc" type="SIGMA_SPS_2020A" transient="false"/>
</LNodeType>
<LNodeType id="SIGMA_XCBR_2020A" lnClass="XCBR">
  <DO name="Beh" type="SIGMA_ENS_Beh_2020A" transient="false"/>
  <DO name="Health" type="SIGMA_ENS_Health_2020A" transient="false"/>
  <DO name="Mod" type="SIGMA_ENC_Mod_2020A" transient="false"/>
  <DO name="OpTmh" type="SIGMA_INS_2020A" transient="false"/>
  <DO name="Loc" type="SIGMA_SPS_2020A" transient="false"/>
```

```
  <DO name="LocKey" type="SIGMA_SPS_2020A" transient="false"/>
  <DO name="OpCnt" type="SIGMA_INS_2020A" transient="false"/>
  <DO name="Dsc" type="SIGMA_SPS_2020A" transient="false"/>
  <DO name="Pos" type="SIGMA_DPC_2020A" transient="false"/>
  <DO name="DscDlTmms" type="SIGMA_ING_2020A" transient="false"/>
  <DO name="CBTmms" type="SIGMA_ING_2020A" transient="false"/>
</LNodeType>
<LNodeType id="SIGMA_PTRC_Tr_2020A" lnClass="PTRC">
  <DO name="Beh" type="SIGMA_ENS_Beh_2020A" transient="false"/>
  <DO name="Mod" type="SIGMA_ENC_Mod_2020A" transient="false"/>
  <DO name="Op" type="SIGMA_ACT_2020A" transient="true"/>
  <DO name="Tr" type="SIGMA_ACT_2020A" transient="false"/>
</LNodeType>
<LNodeType id="SIGMA_HSEQ_Unt_2020A" lnClass="HSEQ">
  <DO name="NamPlt" type="SIGMA_LPL_HYDNs_2020A" transient="false"/>
  <DO name="Mod" type="SIGMA_ENC_Mod_2020A" transient="false"/>
  <DO name="Beh" type="SIGMA_ENS_Beh_2020A" transient="false"/>
  <DO name="Health" type="SIGMA_ENS_Health_2020A" transient="false"/>
  <DO name="OpCntRs" type="SIGMA_INC_2020A" transient="false"/>
  <DO name="Loc" type="SIGMA_SPS_2020A" transient="false"/>
  <DO name="PrecSeq" type="SIGMA_SPS_2020A" transient="false"/>
  <DO name="SeqAct" type="SIGMA_SPS_2020A" transient="false"/>
  <DO name="PrecStep" type="SIGMA_SPS_2020A" transient="false"/>
  <DO name="StepPos" type="SIGMA_INS_2020A" transient="false"/>
  <DO name="StepTmOut" type="SIGMA_SPS_2020A" transient="false"/>
  <DO name="SeqTmOut" type="SIGMA_SPS_2020A" transient="false"/>
  <DO name="SeqCmpl" type="SIGMA_SPS_2020A" transient="false"/>
  <DO name="StrCmd" type="SIGMA_SPC_2020A" transient="false"/>
  <DO name="StrNxt" type="SIGMA_SPC_2020A" transient="false"/>
  <DO name="StepLimTms" type="SIGMA_ING_2020A" transient="false"/>
  <DO name="SeqLimTms" type="SIGMA_ING_2020A" transient="false"/>
</LNodeType>
<LNodeType id="SIGMA_FXOT_KV_2020A" lnClass="FXOT">
  <DO name="Beh" type="SIGMA_ENS_Beh_2020A" transient="false"/>
  <DO name="Health" type="SIGMA_ENS_Health_2020A" transient="false"/>
  <DO name="StrVal" type="SIGMA_ASG_KV_2020A" transient="false"/>
  <DO name="OpDlTmms" type="SIGMA_ING_2020A" transient="false"/>
```

```
  <DO name="RsDlTmms" type="SIGMA_ING_2020A" transient="false"/>
  <DO name="Blk" type="SIGMA_SPS_2020A" transient="false"/>
  <DO name="Op" type="SIGMA_ACT_2020A" transient="false"/>
</LNodeType>
<LNodeType id="SIGMA_ACTM_Coo_2020A" lnClass="ACTM">
  <DO name="Mod" type="SIGMA_ENC_Mod_2020A" transient="false"/>
  <DO name="Beh" type="SIGMA_ENS_Beh_2020A" transient="false"/>
  <DO name="Health" type="SIGMA_ENS_Health_2020A" transient="false"/>
  <DO name="NamPlt" type="SIGMA_LPL_HYDNs_2020A" transient="false"/>
  <DO name="LocSta" type="SIGMA_SPC_2020A" transient="false"/>
  <DO name="ModAct" type="SIGMA_SPC_2020A" transient="false"/>
  <DO name="LocKey" type="SIGMA_SPS_2020A" transient="false"/>
  <DO name="Loc" type="SIGMA_SPS_2020A" transient="false"/>
  <DO name="Flt1" type="SIGMA_SPS_2020A" transient="false"/>
</LNodeType>
<LNodeType id="SIGMA_FXUT_MVar_2020A" lnClass="FXUT">
  <DO name="Mod" type="SIGMA_ENC_Mod_2020A" transient="false"/>
  <DO name="Beh" type="SIGMA_ENS_Beh_2020A" transient="false"/>
  <DO name="Health" type="SIGMA_ENS_Health_2020A" transient="false"/>
  <DO name="StrVal" type="SIGMA_ASG_MVar_2020A" transient="false"/>
  <DO name="OpDlTmms" type="SIGMA_ING_2020A" transient="false"/>
  <DO name="RsDlTmms" type="SIGMA_ING_2020A" transient="false"/>
  <DO name="Blk" type="SIGMA_SPS_2020A" transient="false"/>
  <DO name="Op" type="SIGMA_ACT_2020A" transient="false"/>
</LNodeType>
<LNodeType id="SIGMA_CSYN_Ctl_2020A" lnClass="CSYN">
  <DO name="Beh" type="SIGMA_ENS_Beh_2020A" transient="false"/>
  <DO name="Loc" type="SIGMA_SPS_2020A" transient="false"/>
  <DO name="LocKey" type="SIGMA_SPS_2020A" transient="false"/>
  <DO name="Cmd" type="SIGMA_SPS_2020A" transient="false"/>
  <DO name="Rel" type="SIGMA_SPS_2020A" transient="false"/>
  <DO name="RV" type="SIGMA_SPS_2020A" transient="false"/>
  <DO name="LV" type="SIGMA_SPS_2020A" transient="false"/>
  <DO name="RHz" type="SIGMA_SPS_2020A" transient="false"/>
  <DO name="LHz" type="SIGMA_SPS_2020A" transient="false"/>
  <DO name="VInd" type="SIGMA_SPS_2020A" transient="false"/>
  <DO name="AngInd" type="SIGMA_SPS_2020A" transient="false"/>
```

```
<DO name="HzInd" type="SIGMA_SPS_2020A" transient="false"/>
<DO name="RotDir" type="SIGMA_ENS_RotDir_2020A" transient="false"/>
<DO name="DifVClc" type="SIGMA_MV_2020A" transient="false"/>
<DO name="DifHzClc" type="SIGMA_MV_2020A" transient="false"/>
<DO name="DifAngClc" type="SIGMA_MV_2020A" transient="false"/>
<DO name="V1Clc" type="SIGMA_MV_2020A" transient="false"/>
<DO name="V2Clc" type="SIGMA_MV_2020A" transient="false"/>
<DO name="Hz1Clc" type="SIGMA_MV_2020A" transient="false"/>
<DO name="Hz2Clc" type="SIGMA_MV_2020A" transient="false"/>
<DO name="AccClc" type="SIGMA_MV_2020A" transient="false"/>
<DO name="AccClcDev" type="SIGMA_MV_2020A" transient="false"/>
<DO name="OpCntRs" type="SIGMA_INC_2020A" transient="false"/>
<DO name="LocSta" type="SIGMA_SPC_2020A" transient="false"/>
<DO name="SynPrg" type="SIGMA_SPC_2020A" transient="false"/>
<DO name="RelDeaBus" type="SIGMA_SPC_2020A" transient="false"/>
<DO name="OpModSyn" type="SIGMA_ENC_OpModSyn_2020A" transient="false"/>
<DO name="VNom" type="SIGMA_ASG_2020A" transient="false"/>
<DO name="HzNom" type="SIGMA_ASG_2020A" transient="false"/>
<DO name="VAdpFact" type="SIGMA_ASG_2020A" transient="false"/>
<DO name="AdpAngDeg" type="SIGMA_ASG_2020A" transient="false"/>
<DO name="DlTmms" type="SIGMA_ING_2020A" transient="false"/>
<DO name="MltCmd" type="SIGMA_SPG_2020A" transient="false"/>
<DO name="DifVNg" type="SIGMA_ASG_2020A" transient="false"/>
<DO name="DifVPs" type="SIGMA_ASG_2020A" transient="false"/>
<DO name="DifHzNg" type="SIGMA_ASG_2020A" transient="false"/>
<DO name="DifHzPs" type="SIGMA_ASG_2020A" transient="false"/>
<DO name="DifAngNg" type="SIGMA_ASG_2020A" transient="false"/>
<DO name="DifAngPs" type="SIGMA_ASG_2020A" transient="false"/>
<DO name="MinVSyn" type="SIGMA_ASG_2020A" transient="false"/>
<DO name="MaxVSyn" type="SIGMA_ASG_2020A" transient="false"/>
<DO name="DetSyn" type="SIGMA_ASG_2020A" transient="false"/>
<DO name="LivDeaMod" type="SIGMA_ENG_LivDeaMod_2020A" transient="false"/>
<DO name="DeaLinVal" type="SIGMA_ASG_2020A" transient="false"/>
<DO name="LivLinVal" type="SIGMA_ASG_2020A" transient="false"/>
<DO name="DeaBusVal" type="SIGMA_ASG_2020A" transient="false"/>
<DO name="LivBusVal" type="SIGMA_ASG_2020A" transient="false"/>
<DO name="VAdj" type="SIGMA_SPG_2020A" transient="false"/>
```

```xml
  <DO name="VChr" type="SIGMA_ASG_2020A" transient="false"/>
  <DO name="VIntvTmms" type="SIGMA_ING_2020A" transient="false"/>
  <DO name="MinVTmms" type="SIGMA_ING_2020A" transient="false"/>
  <DO name="MaxVTmms" type="SIGMA_ING_2020A" transient="false"/>
  <DO name="HzAdj" type="SIGMA_SPG_2020A" transient="false"/>
  <DO name="HzChr" type="SIGMA_ASG_2020A" transient="false"/>
  <DO name="HzIntvTmms" type="SIGMA_ING_2020A" transient="false"/>
  <DO name="MinHzTmms" type="SIGMA_ING_2020A" transient="false"/>
  <DO name="MaxHzTmms" type="SIGMA_ING_2020A" transient="false"/>
  <DO name="HzTgtVal" type="SIGMA_ASG_2020A" transient="false"/>
  <DO name="KckPls" type="SIGMA_SPG_2020A" transient="false"/>
  <DO name="DlSynTmms" type="SIGMA_ING_2020A" transient="false"/>
  <DO name="TotTmms" type="SIGMA_ING_2020A" transient="false"/>
</LNodeType>
<LNodeType id="SIGMA_HTGV_2020A" lnClass="HTGV">
  <DO name="NamPlt" type="SIGMA_LPL_HYDNs_2020A" transient="false"/>
  <DO name="EEHealth" type="SIGMA_ENS_Health_2020A" transient="false"/>
  <DO name="EEName" type="SIGMA_DPL_2020A" transient="false"/>
  <DO name="OpCnt" type="SIGMA_INS_2020A" transient="false"/>
  <DO name="Loc" type="SIGMA_SPS_2020A" transient="false"/>
  <DO name="LocKey" type="SIGMA_SPS_2020A" transient="false"/>
  <DO name="PosCls" type="SIGMA_SPS_2020A" transient="false"/>
  <DO name="PosOpn" type="SIGMA_SPS_2020A" transient="false"/>
  <DO name="PosSNL" type="SIGMA_SPS_2020A" transient="false"/>
  <DO name="SMLkdCls" type="SIGMA_SPS_2020A" transient="false"/>
  <DO name="SMLkdMnt" type="SIGMA_SPS_2020A" transient="false"/>
  <DO name="PinAlm" type="SIGMA_SPS_2020A" transient="false"/>
  <DO name="RodAlm" type="SIGMA_SPS_2020A" transient="false"/>
  <DO name="DvWrn" type="SIGMA_SPS_2020A" transient="false"/>
  <DO name="DvAlm" type="SIGMA_ASG_2020A" transient="false"/>
  <DO name="OpCntRs" type="SIGMA_INC_2020A" transient="false"/>
  <DO name="PosSpt" type="SIGMA_APC_2020A" transient="false"/>
  <DO name="DithAct" type="SIGMA_SPC_2020A" transient="false"/>
  <DO name="PosSNLSet" type="SIGMA_ASG_2020A" transient="false"/>
  <DO name="DvWrnSpt" type="SIGMA_ASG_2020A" transient="false"/>
  <DO name="DvAlmSpt" type="SIGMA_ASG_2020A" transient="false"/>
  <DO name="CbrRng" type="SIGMA_ASG_2020A" transient="false"/>
```

```xml
  <DO name="ClsLimRst" type="SIGMA_RST_2020A" transient="false"/>
  <DO name="ClsLimHys" type="SIGMA_ASG_2020A" transient="false"/>
  <DO name="ClsTmsSet" type="SIGMA_ING_2020A" transient="false"/>
  <DO name="OpnTmsSet" type="SIGMA_ING_2020A" transient="false"/>
  <DO name="DithOfs" type="SIGMA_ASG_2020A" transient="false"/>
  <DO name="PosPct" type="SIGMA_MV_2020A" transient="false"/>
  <DO name="PosDeg" type="SIGMA_MV_2020A" transient="false"/>
  <DO name="SMPres" type="SIGMA_MV_2020A" transient="false"/>
  <DO name="Beh" type="SIGMA_ENS_Beh_2020A" transient="false"/>
 </LNodeType>
 <LNodeType id="SIGMA_HSPD_2020A" lnClass="HSPD">
  <DO name="NamPlt" type="SIGMA_LPL_HYDNs_2020A" transient="false"/>
  <DO name="Mod" type="SIGMA_ENC_Mod_2020A" transient="false"/>
  <DO name="Beh" type="SIGMA_ENS_Beh_2020A" transient="false"/>
  <DO name="Health" type="SIGMA_ENS_Health_2020A" transient="false"/>
  <DO name="EEHealth" type="SIGMA_ENS_Health_2020A" transient="false"/>
  <DO name="EEName" type="SIGMA_DPL_2020A" transient="false"/>
  <DO name="Loc" type="SIGMA_SPS_2020A" transient="false"/>
  <DO name="StndStl" type="SIGMA_SPS_2020A" transient="false"/>
  <DO name="SpdCrp" type="SIGMA_SPS_2020A" transient="false"/>
  <DO name="SpdBrk" type="SIGMA_SPS_2020A" transient="false"/>
  <DO name="SpdLub" type="SIGMA_SPS_2020A" transient="false"/>
  <DO name="SpdLft" type="SIGMA_SPS_2020A" transient="false"/>
  <DO name="SpdRB" type="SIGMA_SPS_2020A" transient="false"/>
  <DO name="SpdExt" type="SIGMA_SPS_2020A" transient="false"/>
  <DO name="SpdSyn" type="SIGMA_SPS_2020A" transient="false"/>
  <DO name="SpdOvr" type="SIGMA_SPS_2020A" transient="false"/>
  <DO name="DirRot" type="SIGMA_SPS_2020A" transient="false"/>
  <DO name="SetSpdCrp" type="SIGMA_ASG_2020A" transient="false"/>
  <DO name="SetSpdBrk" type="SIGMA_ASG_2020A" transient="false"/>
  <DO name="SetSpdLub" type="SIGMA_ASG_2020A" transient="false"/>
  <DO name="SetSpdLft" type="SIGMA_ASG_2020A" transient="false"/>
  <DO name="SetSpdRb" type="SIGMA_ASG_2020A" transient="false"/>
  <DO name="SetSpdExt" type="SIGMA_ASG_2020A" transient="false"/>
  <DO name="SetSpdSyn" type="SIGMA_ASG_2020A" transient="false"/>
  <DO name="SetSpdOv" type="SIGMA_ASG_2020A" transient="false"/>
  <DO name="SpdPct" type="SIGMA_MV_2020A" transient="false"/>
```

```xml
  <DO name="Blk" type="SIGMA_SPS_2020A" transient="false"/>
</LNodeType>
<LNodeType id="SIGMA_SVBR_2020A" lnClass="SVBR">
  <DO name="Beh" type="SIGMA_ENS_Beh_2020A" transient="false"/>
  <DO name="Health" type="SIGMA_ENS_Health_2020A" transient="false"/>
  <DO name="NamPlt" type="SIGMA_LPL_HYDNs_2020A" transient="false"/>
  <DO name="Mod" type="SIGMA_ENC_Mod_2020A" transient="false"/>
  <DO name="Trip" type="SIGMA_SPS_2020A" transient="false"/>
  <DO name="AlmSpt" type="SIGMA_SPS_2020A" transient="false"/>
  <DO name="TripSpt" type="SIGMA_ASG_2020A" transient="false"/>
  <DO name="Vbr" type="SIGMA_MV_Smax_2020A" transient="false"/>
  <DO name="Alm" type="SIGMA_SPS_2020A" transient="false"/>
</LNodeType>
<LNodeType id="SIGMA_HGOV_2020A" lnClass="HGOV">
  <DO name="NamPlt" type="SIGMA_LPL_HYDNs_2020A" transient="false"/>
  <DO name="Mod" type="SIGMA_ENC_Mod_2020A" transient="false"/>
  <DO name="Beh" type="SIGMA_ENS_Beh_2020A" transient="false"/>
  <DO name="Health" type="SIGMA_ENS_Health_2020A" transient="false"/>
  <DO name="Loc" type="SIGMA_SPS_2020A" transient="false"/>
  <DO name="LocKey" type="SIGMA_SPS_2020A" transient="false"/>
  <DO name="Flt" type="SIGMA_SPS_2020A" transient="false"/>
  <DO name="Out" type="SIGMA_MV_2020A" transient="false"/>
  <DO name="LocSta" type="SIGMA_SPC_2020A" transient="false"/>
  <DO name="OpCntRs" type="SIGMA_INC_2020A" transient="false"/>
  <DO name="ModAct" type="SIGMA_SPC_2020A" transient="false"/>
  <DO name="ExSptEna" type="SIGMA_SPC_2020A" transient="false"/>
  <DO name="Droop" type="SIGMA_ASG_2020A" transient="false"/>
</LNodeType>
<LNodeType id="SIGMA_HVLV_2020A" lnClass="HVLV">
  <DO name="NamPlt" type="SIGMA_LPL_HYDNs_2020A" transient="false"/>
  <DO name="Mod" type="SIGMA_ENC_Mod_2020A" transient="false"/>
  <DO name="Beh" type="SIGMA_ENS_Beh_2020A" transient="false"/>
  <DO name="Health" type="SIGMA_ENS_Health_2020A" transient="false"/>
  <DO name="EEHealth" type="SIGMA_ENS_Health_2020A" transient="false"/>
  <DO name="EEName" type="SIGMA_DPL_2020A" transient="false"/>
  <DO name="LocKey" type="SIGMA_SPS_2020A" transient="false"/>
  <DO name="Loc" type="SIGMA_SPS_2020A" transient="false"/>
```

```
  <DO name="PosCls" type="SIGMA_SPS_2020A" transient="false"/>
  <DO name="PosOpn" type="SIGMA_SPS_2020A" transient="false"/>
  <DO name="Mvm" type="SIGMA_SPS_2020A" transient="false"/>
  <DO name="Stuck" type="SIGMA_SPS_2020A" transient="false"/>
  <DO name="DvWrn" type="SIGMA_SPS_2020A" transient="false"/>
  <DO name="DvAlm" type="SIGMA_SPS_2020A" transient="false"/>
  <DO name="OpnLimRst" type="SIGMA_RST_2020A" transient="false"/>
  <DO name="ClsLimRst" type="SIGMA_RST_2020A" transient="false"/>
  <DO name="Incr" type="SIGMA_ASG_2020A" transient="false"/>
  <DO name="SldStrPs" type="SIGMA_ASG_2020A" transient="false"/>
  <DO name="SldStrNg" type="SIGMA_ASG_2020A" transient="false"/>
  <DO name="SldOfsPs" type="SIGMA_ASG_2020A" transient="false"/>
  <DO name="SldOfsNg" type="SIGMA_ASG_2020A" transient="false"/>
  <DO name="CbrRng" type="SIGMA_ASG_2020A" transient="false"/>
  <DO name="DvWrnSpt" type="SIGMA_ASG_2020A" transient="false"/>
  <DO name="DvAlmSpt" type="SIGMA_ASG_2020A" transient="false"/>
  <DO name="PosPct" type="SIGMA_MV_2020A" transient="false"/>
  <DO name="PosDeg" type="SIGMA_MV_2020A" transient="false"/>
  <DO name="Flw" type="SIGMA_MV_2020A" transient="false"/>
  <DO name="LocSta" type="SIGMA_SPC_2020A" transient="false"/>
  <DO name="OpCntRs" type="SIGMA_INC_2020A" transient="false"/>
  <DO name="PosSpt" type="SIGMA_APC_2020A" transient="false"/>
  <DO name="Opn" type="SIGMA_SPC_2020A" transient="false"/>
  <DO name="PosChg" type="SIGMA_BSC_PosChg_2020A" transient="false"/>
  <DO name="BlkOpn" type="SIGMA_SPC_2020A" transient="false"/>
  <DO name="BlkCls" type="SIGMA_SPC_2020A" transient="false"/>
</LNodeType>
<LNodeType id="SIGMA_HMBR_2020A" lnClass="HMBR">
  <DO name="Beh" type="SIGMA_ENS_Beh_2020A" transient="false"/>
  <DO name="Mod" type="SIGMA_ENC_Mod_2020A" transient="false"/>
  <DO name="BlkOn" type="SIGMA_SPC_2020A" transient="false"/>
  <DO name="BrkOn" type="SIGMA_SPS_2020A" transient="false"/>
  <DO name="Operate" type="SIGMA_SPC_2020A" transient="false"/>
  <DO name="BrkOff" type="SIGMA_SPS_2020A" transient="false"/>
  <DO name="OpRs" type="SIGMA_SPC_2020A" transient="false"/>
</LNodeType>
<LNodeType id="SIGMA_FSPT_2020A" lnClass="FSPT">
```

```xml
  <DO name="Mod" type="SIGMA_ENC_Mod_2020A" transient="false"/>
  <DO name="Beh" type="SIGMA_ENS_Beh_2020A" transient="false"/>
  <DO name="Health" type="SIGMA_ENS_Health_2020A" transient="false"/>
  <DO name="SptVal" type="SIGMA_APC_2020A" transient="false"/>
  <DO name="SptEndSt" type="SIGMA_ENS_SptEndSt_2020A" transient="false"/>
  <DO name="AdjSt" type="SIGMA_ENS_AdjSt_2020A" transient="false"/>
  <DO name="SptMem" type="SIGMA_MV_2020A" transient="false"/>
  <DO name="ErrTerm" type="SIGMA_MV_2020A" transient="false"/>
  <DO name="Out" type="SIGMA_MV_2020A" transient="false"/>
  <DO name="MaxRst" type="SIGMA_ASG_2020A" transient="false"/>
  <DO name="MinRst" type="SIGMA_ASG_2020A" transient="false"/>
  <DO name="Auto" type="SIGMA_SPC_2020A" transient="false"/>
  <DO name="SptDvAlm" type="SIGMA_SPS_2020A" transient="false"/>
  <DO name="DvAlm" type="SIGMA_ASG_2020A" transient="false"/>
  <DO name="SptUp" type="SIGMA_SPS_2020A" transient="false"/>
  <DO name="SptDn" type="SIGMA_SPS_2020A" transient="false"/>
  <DO name="SptDir" type="SIGMA_SPS_2020A" transient="false"/>
  <DO name="Blk" type="SIGMA_SPS_2020A" transient="false"/>
</LNodeType>
<LNodeType id="SIGMA_MMDC_2020A" lnClass="MMDC">
  <DO name="Mod" type="SIGMA_ENC_Mod_2020A" transient="false"/>
  <DO name="Beh" type="SIGMA_ENS_Beh_2020A" transient="false"/>
  <DO name="Health" type="SIGMA_ENS_Health_2020A" transient="false"/>
  <DO name="Watt" type="SIGMA_MV_2020A" transient="false"/>
  <DO name="Amp" type="SIGMA_MV_2020A" transient="false"/>
  <DO name="Vol" type="SIGMA_MV_2020A" transient="false"/>
  <DO name="VolPsGnd" type="SIGMA_MV_2020A" transient="false"/>
  <DO name="VolNgGnd" type="SIGMA_MV_2020A" transient="false"/>
  <DO name="Ris" type="SIGMA_MV_Ris_2020A" transient="false"/>
  <DO name="RisPsGnd" type="SIGMA_MV_Ris_2020A" transient="false"/>
  <DO name="RisNgGnd" type="SIGMA_MV_Ris_2020A" transient="false"/>
</LNodeType>
<LNodeType id="SIGMA_ACTM_Exc_2020A" lnClass="ACTM">
  <DO name="Mod" type="SIGMA_ENC_Mod_2020A" transient="false"/>
  <DO name="Beh" type="SIGMA_ENS_Beh_2020A" transient="false"/>
  <DO name="Health" type="SIGMA_ENS_Health_2020A" transient="false"/>
  <DO name="NamPlt" type="SIGMA_LPL_HYDNs_2020A" transient="false"/>
```

```
  <DO name="LocSta" type="SIGMA_SPC_2020A" transient="false"/>
  <DO name="ModAct" type="SIGMA_SPC_2020A" transient="false"/>
  <DO name="LocKey" type="SIGMA_SPS_2020A" transient="false"/>
  <DO name="Loc" type="SIGMA_SPS_2020A" transient="false"/>
  <DO name="Flt1" type="SIGMA_SPS_2020A" transient="false"/>
</LNodeType>
<LNodeType id="SIGMA_FSPT_Amp_2020A" lnClass="FSPT">
  <DO name="Mod" type="SIGMA_ENC_Mod_2020A" transient="false"/>
  <DO name="Beh" type="SIGMA_ENS_Beh_2020A" transient="false"/>
  <DO name="Health" type="SIGMA_ENS_Health_2020A" transient="false"/>
  <DO name="SptVal" type="SIGMA_APC_Amp_2020A" transient="false"/>
  <DO name="SptEndSt" type="SIGMA_ENS_SptEndSt_2020A" transient="false"/>
  <DO name="AdjSt" type="SIGMA_ENS_AdjSt_2020A" transient="false"/>
  <DO name="SptMem" type="SIGMA_MV_Amp_2020A" transient="false"/>
  <DO name="ErrTerm" type="SIGMA_MV_Amp_2020A" transient="false"/>
  <DO name="Out" type="SIGMA_MV_Amp_2020A" transient="false"/>
  <DO name="MaxRst" type="SIGMA_ASG_Amps_2020A" transient="false"/>
  <DO name="MinRst" type="SIGMA_ASG_Amps_2020A" transient="false"/>
  <DO name="Auto" type="SIGMA_SPC_2020A" transient="false"/>
  <DO name="SptDvAlm" type="SIGMA_SPS_2020A" transient="false"/>
  <DO name="DvAlm" type="SIGMA_ASG_Amps_2020A" transient="false"/>
  <DO name="SptUp" type="SIGMA_SPS_2020A" transient="false"/>
  <DO name="SptDn" type="SIGMA_SPS_2020A" transient="false"/>
  <DO name="SptDir" type="SIGMA_SPS_2020A" transient="false"/>
  <DO name="Blk" type="SIGMA_SPS_2020A" transient="false"/>
</LNodeType>
<LNodeType id="SIGMA_FSCC_Mw_2020A" lnClass="FSCC">
  <DO name="Mod" type="SIGMA_ENC_Mod_2020A" transient="false"/>
  <DO name="Beh" type="SIGMA_ENS_Beh_2020A" transient="false"/>
  <DO name="ActSchdRef" type="SIGMA_ORG_FSCC_2020A" transient="false"/>
  <DO name="CtlEnt" type="SIGMA_ORG_FSCC_2020A" transient="false"/>
  <DO name="Schd1" type="SIGMA_ORG_FSCC_2020A" transient="false"/>
  <DO name="Schd2" type="SIGMA_ORG_FSCC_2020A" transient="false"/>
  <DO name="Schd3" type="SIGMA_ORG_FSCC_2020A" transient="false"/>
  <DO name="Schd4" type="SIGMA_ORG_FSCC_2020A" transient="false"/>
  <DO name="Schd5" type="SIGMA_ORG_FSCC_2020A" transient="false"/>
  <DO name="Schd6" type="SIGMA_ORG_FSCC_2020A" transient="false"/>
```

```xml
  <DO name="Schd7" type="SIGMA_ORG_FSCC_2020A" transient="false"/>
</LNodeType>
<LNodeType id="SIGMA_FSCH_Mw_2020A" lnClass="FSCH">
  <DO name="NamPlt" type="SIGMA_LPL_HYDNs_2020A" transient="false"/>
  <DO name="Mod" type="SIGMA_ENC_Mod_2020A" transient="false"/>
  <DO name="Beh" type="SIGMA_ENS_Beh_2020A" transient="false"/>
  <DO name="EvTrg" type="SIGMA_SPG_2020A" transient="false"/>
  <DO name="ValMV" type="SIGMA_MV_MW_2020A" transient="false"/>
  <DO name="ValASG1" type="SIGMA_ASG_MW_2020A" transient="false"/>
  <DO name="ValASG2" type="SIGMA_ASG_MW_2020A" transient="false"/>
  <DO name="ValASG3" type="SIGMA_ASG_MW_2020A" transient="false"/>
  <DO name="ValASG4" type="SIGMA_ASG_MW_2020A" transient="false"/>
  <DO name="ValASG5" type="SIGMA_ASG_MW_2020A" transient="false"/>
  <DO name="ValASG6" type="SIGMA_ASG_MW_2020A" transient="false"/>
  <DO name="ValASG7" type="SIGMA_ASG_MW_2020A" transient="false"/>
  <DO name="ValASG8" type="SIGMA_ASG_MW_2020A" transient="false"/>
  <DO name="IntvPer" type="SIGMA_ASG_MW_2020A" transient="false"/>
  <DO name="SchdIntv" type="SIGMA_ING_2020A" transient="false"/>
  <DO name="SchdReuse" type="SIGMA_SPG_2020A" transient="false"/>
  <DO name="StrTm" type="SIGMA_TSG_2020A" transient="false"/>
</LNodeType>
<LNodeType id="SIGMA_ACTM_Aux_2020A" lnClass="ACTM">
  <DO name="Mod" type="SIGMA_ENC_Mod_2020A" transient="false"/>
  <DO name="Beh" type="SIGMA_ENS_Beh_2020A" transient="false"/>
  <DO name="Health" type="SIGMA_ENS_Health_2020A" transient="false"/>
  <DO name="NamPlt" type="SIGMA_LPL_HYDNs_2020A" transient="false"/>
  <DO name="LocSta" type="SIGMA_SPC_2020A" transient="false"/>
  <DO name="ModAct" type="SIGMA_SPC_2020A" transient="false"/>
  <DO name="LocKey" type="SIGMA_SPS_2020A" transient="false"/>
  <DO name="Loc" type="SIGMA_SPS_2020A" transient="false"/>
  <DO name="Flt1" type="SIGMA_SPS_2020A" transient="false"/>
</LNodeType>
<DOType id="SIGMA_LPL_HYDNs_2020A" cdc="LPL">
  <DA bType="VisString255" name="vendor" dchg="false" dupd="false" fc="DC" qchg="false"/>
  <DA bType="VisString255" name="swRev" dchg="false" dupd="false" fc="DC" qchg="false"/>
  <DA bType="VisString255" name="d" dchg="false" dupd="false" fc="DC" qchg="false"/>
  <DA bType="Unicode255" name="dU" dchg="false" dupd="false" fc="DC" qchg="false"/>
```

```xml
    <DA bType="VisString255" name="configRev" dchg="false" dupd="false" fc="DC" qchg="false"/>
    <DA bType="INT32" name="paramRev" dchg="false" dupd="false" fc="ST" qchg="false"/>
    <DA bType="INT32" name="valRev" dchg="false" dupd="false" fc="ST" qchg="false"/>
    <DA bType="VisString255" name="ldNs" dchg="false" dupd="false" fc="EX" qchg="false"/>
  </DOType>
  <DOType id="SIGMA_SPS_2020A" cdc="SPS">
    <DA bType="BOOLEAN" name="stVal" dchg="false" dupd="false" fc="ST" qchg="false"/>
    <DA bType="Quality" name="q" dchg="false" dupd="false" fc="ST" qchg="false"/>
    <DA bType="Timestamp" name="t" dchg="false" dupd="false" fc="ST" qchg="false"/>
    <DA bType="VisString255" name="d" dchg="false" dupd="false" fc="DC" qchg="false"/>
    <DA bType="Unicode255" name="dU" dchg="false" dupd="false" fc="DC" qchg="false"/>
  </DOType>
  <DOType id="SIGMA_ENC_Mod_2020A" cdc="ENC">
    <DA bType="Struct" name="origin" type="SIGMA_Originator_2020A" dchg="false" dupd="false" fc="ST" qchg="false"/>
    <DA bType="INT8U" name="ctlNum" dchg="false" dupd="false" fc="ST" qchg="false"/>
    <DA bType="Enum" name="stVal" type="SIGMA_BehaviourModeKind_2020A" dchg="false" dupd="false" fc="ST" qchg="false"/>
    <DA bType="Quality" name="q" dchg="false" dupd="false" fc="ST" qchg="false"/>
    <DA bType="Timestamp" name="t" dchg="false" dupd="false" fc="ST" qchg="false"/>
    <DA bType="BOOLEAN" name="stSeld" dchg="false" dupd="false" fc="ST" qchg="false"/>
    <DA bType="BOOLEAN" name="opOk" dchg="false" dupd="false" fc="OR" qchg="false"/>
    <DA bType="Timestamp" name="tOpOk" dchg="false" dupd="false" fc="OR" qchg="false"/>
    <DA bType="Enum" name="ctlModel" type="SIGMA_CtlModelKind_2020A" dchg="false" dupd="false" fc="CF" qchg="false"/>
    <DA bType="INT32U" name="sboTimeout" dchg="false" dupd="false" fc="CF" qchg="false"/>
    <DA bType="Enum" name="sboClass" type="SIGMA_SboClassKind_2020A" dchg="false" dupd="false" fc="CF" qchg="false"/>
    <DA bType="INT32U" name="operTimeout" dchg="false" dupd="false" fc="CF" qchg="false"/>
    <DA bType="VisString255" name="d" dchg="false" dupd="false" fc="DC" qchg="false"/>
    <DA bType="Unicode255" name="dU" dchg="false" dupd="false" fc="DC" qchg="false"/>
    <DA bType="Struct" name="Oper" type="SIGMA_Oper_Mod_2020A" dchg="false" dupd="false" fc="CO" qchg="false"/>
  </DOType>
  <DOType id="SIGMA_MV_MVar_2020A" cdc="MV">
    <DA bType="Struct" name="mag" type="SIGMA_AnalogueValue_f_2020A" dchg="false" dupd="false" fc="MX" qchg="false"/>
    <DA bType="Enum" name="range" type="SIGMA_RangeKind_2020A" dchg="false" dupd="false" fc="MX" qchg="false"/>
    <DA bType="Quality" name="q" dchg="false" dupd="false" fc="MX" qchg="false"/>
    <DA bType="Timestamp" name="t" dchg="false" dupd="false" fc="MX" qchg="false"/>
```

```
    <DA bType="Struct" name="units" type="SIGMA_Units_MVar_2020A" dchg="false" dupd="false" fc="CF"
qchg="false"/>
    <DA bType="INT32U" name="db" dchg="false" dupd="false" fc="CF" qchg="false"/>
    <DA bType="INT32U" name="zeroDb" dchg="false" dupd="false" fc="CF" qchg="false"/>
    <DA bType="Struct" name="sVC" type="SIGMA_ScaledValueConfig_2020A" dchg="false" dupd="false" fc="CF"
qchg="false"/>
    <DA bType="Struct" name="rangeC" type="SIGMA_RangeConfig_2020A" dchg="false" dupd="false" fc="CF"
qchg="false"/>
    <DA bType="INT32U" name="smpRate" dchg="false" dupd="false" fc="CF" qchg="false"/>
    <DA bType="VisString255" name="d" dchg="false" dupd="false" fc="DC" qchg="false"/>
    <DA bType="Unicode255" name="dU" dchg="false" dupd="false" fc="DC" qchg="false"/>
   </DOType>
   <DOType id="SIGMA_TAG_2020A" cdc="TAG">
    <DA bType="Struct" name="origin" type="SIGMA_Originator_2020A" dchg="false" dupd="false" fc="ST"
qchg="false"/>
    <DA bType="INT8U" name="ctlNum" dchg="false" dupd="false" fc="ST" qchg="false"/>
    <DA bType="BOOLEAN" name="stVal" dchg="false" dupd="false" fc="ST" qchg="false"/>
    <DA bType="Quality" name="q" dchg="false" dupd="false" fc="ST" qchg="false"/>
    <DA bType="Timestamp" name="t" dchg="false" dupd="false" fc="ST" qchg="false"/>
    <DA bType="BOOLEAN" name="stSeld" dchg="false" dupd="false" fc="ST" qchg="false"/>
    <DA bType="VisString255" name="id" dchg="false" dupd="false" fc="CF" qchg="false"/>
    <DA bType="VisString255" name="startTime" dchg="false" dupd="false" fc="CF" qchg="false"/>
    <DA bType="VisString255" name="stopTime" dchg="false" dupd="false" fc="CF" qchg="false"/>
    <DA bType="Enum" name="ctlModel" type="SIGMA_CtlModelKind_2020A" dchg="false" dupd="false" fc="CF"
qchg="false"/>
    <DA bType="INT32U" name="sboTimeout" dchg="false" dupd="false" fc="CF" qchg="false"/>
    <DA bType="VisString255" name="d" dchg="false" dupd="false" fc="DC" qchg="false"/>
    <DA bType="Unicode255" name="dU" dchg="false" dupd="false" fc="DC" qchg="false"/>
    <DA bType="VisString255" name="cdcNs" dchg="false" dupd="false" fc="EX" qchg="false"/>
    <DA bType="VisString255" name="cdcName" dchg="false" dupd="false" fc="EX" qchg="false"/>
    <DA bType="VisString255" name="dataNs" dchg="false" dupd="false" fc="EX" qchg="false"/>
   </DOType>
   <DOType id="SIGMA_ENS_Beh_2020A" cdc="ENS">
    <DA bType="Enum" name="stVal" type="SIGMA_BehaviourModeKind_2020A" dchg="false" dupd="false" fc="ST"
qchg="false"/>
    <DA bType="Quality" name="q" dchg="false" dupd="false" fc="ST" qchg="false"/>
    <DA bType="Timestamp" name="t" dchg="false" dupd="false" fc="ST" qchg="false"/>
    <DA bType="VisString255" name="d" dchg="false" dupd="false" fc="DC" qchg="false"/>
    <DA bType="Unicode255" name="dU" dchg="false" dupd="false" fc="DC" qchg="false"/>
   </DOType>
```

```
<DOType id="SIGMA_SPC_2020A" cdc="SPC">
  <DA bType="Struct" name="origin" type="SIGMA_Originator_2020A" dchg="false" dupd="false" fc="ST"
qchg="false"/>
  <DA bType="Struct" name="Oper" type="SIGMA_Oper_SPC_2020A" dchg="false" dupd="false" fc="CO"
qchg="false"/>
  <DA bType="INT8U" name="ctlNum" dchg="false" dupd="false" fc="ST" qchg="false"/>
  <DA bType="BOOLEAN" name="stVal" dchg="false" dupd="false" fc="ST" qchg="false"/>
  <DA bType="Quality" name="q" dchg="false" dupd="false" fc="ST" qchg="false"/>
  <DA bType="Timestamp" name="t" dchg="false" dupd="false" fc="ST" qchg="false"/>
  <DA bType="BOOLEAN" name="stSeld" dchg="false" dupd="false" fc="ST" qchg="false"/>
  <DA bType="BOOLEAN" name="opRcvd" dchg="false" dupd="false" fc="OR" qchg="false"/>
  <DA bType="BOOLEAN" name="opOk" dchg="false" dupd="false" fc="OR" qchg="false"/>
  <DA bType="Timestamp" name="tOpOk" dchg="false" dupd="false" fc="OR" qchg="false"/>
  <DA bType="Struct" name="pulseConfig" type="SIGMA_PulseConfig_2020A" dchg="false" dupd="false" fc="CF"
qchg="false"/>
  <DA bType="Enum" name="ctlModel" type="SIGMA_CtlModelKind_2020A" dchg="false" dupd="false" fc="CF"
qchg="false"/>
  <DA bType="INT32U" name="sboTimeout" dchg="false" dupd="false" fc="CF" qchg="false"/>
  <DA bType="Enum" name="sboClass" type="SIGMA_SboClassKind_2020A" dchg="false" dupd="false" fc="CF"
qchg="false"/>
  <DA bType="INT32U" name="operTimeout" dchg="false" dupd="false" fc="CF" qchg="false"/>
  <DA bType="VisString255" name="d" dchg="false" dupd="false" fc="DC" qchg="false"/>
  <DA bType="Unicode255" name="dU" dchg="false" dupd="false" fc="DC" qchg="false"/>
</DOType>
<DOType id="SIGMA_INS_2020A" cdc="INS">
  <DA bType="INT32" name="stVal" dchg="false" dupd="false" fc="ST" qchg="false"/>
  <DA bType="Quality" name="q" dchg="false" dupd="false" fc="ST" qchg="false"/>
  <DA bType="Timestamp" name="t" dchg="false" dupd="false" fc="ST" qchg="false"/>
  <DA bType="Struct" name="units" type="SIGMA_Units_2020A" dchg="false" dupd="false" fc="CF" qchg="false"/>
  <DA bType="VisString255" name="d" dchg="false" dupd="false" fc="DC" qchg="false"/>
  <DA bType="Unicode255" name="dU" dchg="false" dupd="false" fc="DC" qchg="false"/>
</DOType>
<DOType id="SIGMA_MV_2020A" cdc="MV">
  <DA bType="Struct" name="instMag" type="SIGMA_AnalogueValue_f_2020A" dchg="false" dupd="false" fc="MX"
qchg="false"/>
  <DA bType="Struct" name="mag" type="SIGMA_AnalogueValue_f_2020A" dchg="false" dupd="false" fc="MX"
qchg="false"/>
  <DA bType="Enum" name="range" type="SIGMA_RangeKind_2020A" dchg="false" dupd="false" fc="MX"
qchg="false"/>
  <DA bType="Quality" name="q" dchg="false" dupd="false" fc="MX" qchg="false"/>
  <DA bType="Timestamp" name="t" dchg="false" dupd="false" fc="MX" qchg="false"/>
```

```xml
<DA bType="Struct" name="units" type="SIGMA_Units_2020A" dchg="false" dupd="false" fc="CF" qchg="false"/>
<DA bType="INT32U" name="db" dchg="false" dupd="false" fc="CF" qchg="false"/>
<DA bType="INT32U" name="zeroDb" dchg="false" dupd="false" fc="CF" qchg="false"/>
<DA bType="Struct" name="sVC" type="SIGMA_ScaledValueConfig_2020A" dchg="false" dupd="false" fc="CF" qchg="false"/>
<DA bType="Struct" name="rangeC" type="SIGMA_RangeConfig_2020A" dchg="false" dupd="false" fc="CF" qchg="false"/>
<DA bType="INT32U" name="smpRate" dchg="false" dupd="false" fc="CF" qchg="false"/>
<DA bType="VisString255" name="d" dchg="false" dupd="false" fc="DC" qchg="false"/>
<DA bType="Unicode255" name="dU" dchg="false" dupd="false" fc="DC" qchg="false"/>
</DOType>
<DOType id="SIGMA_ENS_AdjSt_2020A" cdc="ENS">
<DA bType="Enum" name="stVal" type="SIGMA_AdjustmentKind_2020A" dchg="false" dupd="false" fc="ST" qchg="false"/>
<DA bType="Quality" name="q" dchg="false" dupd="false" fc="ST" qchg="false"/>
<DA bType="Timestamp" name="t" dchg="false" dupd="false" fc="ST" qchg="false"/>
<DA bType="VisString255" name="d" dchg="false" dupd="false" fc="DC" qchg="false"/>
<DA bType="Unicode255" name="dU" dchg="false" dupd="false" fc="DC" qchg="false"/>
</DOType>
<DOType id="SIGMA_APC_2020A" cdc="APC">
<DA bType="Struct" name="origin" type="SIGMA_Originator_2020A" dchg="false" dupd="false" fc="MX" qchg="false"/>
<DA bType="INT8U" name="ctlNum" dchg="false" dupd="false" fc="MX" qchg="false"/>
<DA bType="Struct" name="Oper" type="SIGMA_Oper_APC_2020A" dchg="false" dupd="false" fc="CO" qchg="false"/>
<DA bType="Struct" name="mxVal" type="SIGMA_AnalogueValue_f_2020A" dchg="false" dupd="false" fc="MX" qchg="false"/>
<DA bType="Quality" name="q" dchg="false" dupd="false" fc="MX" qchg="false"/>
<DA bType="Timestamp" name="t" dchg="false" dupd="false" fc="MX" qchg="false"/>
<DA bType="BOOLEAN" name="stSeld" dchg="false" dupd="false" fc="MX" qchg="false"/>
<DA bType="BOOLEAN" name="opRcvd" dchg="false" dupd="false" fc="OR" qchg="false"/>
<DA bType="BOOLEAN" name="opOk" dchg="false" dupd="false" fc="OR" qchg="false"/>
<DA bType="Timestamp" name="tOpOk" dchg="false" dupd="false" fc="OR" qchg="false"/>
<DA bType="Enum" name="ctlModel" type="SIGMA_CtlModelKind_2020A" dchg="false" dupd="false" fc="CF" qchg="false"/>
<DA bType="INT32U" name="sboTimeout" dchg="false" dupd="false" fc="CF" qchg="false"/>
<DA bType="Enum" name="sboClass" type="SIGMA_SboClassKind_2020A" dchg="false" dupd="false" fc="CF" qchg="false"/>
<DA bType="INT32U" name="db" dchg="false" dupd="false" fc="CF" qchg="false"/>
<DA bType="Struct" name="units" type="SIGMA_Units_2020A" dchg="false" dupd="false" fc="CF" qchg="false"/>
<DA bType="Struct" name="sVC" type="SIGMA_ScaledValueConfig_2020A" dchg="false" dupd="false" fc="CF" qchg="false"/>
```

```
<DA bType="Struct" name="minVal" type="SIGMA_AnalogueValue_f_2020A" dchg="false" dupd="false" fc="CF" qchg="false"/>

<DA bType="Struct" name="maxVal" type="SIGMA_AnalogueValue_f_2020A" dchg="false" dupd="false" fc="CF" qchg="false"/>

<DA bType="Struct" name="stepSize" type="SIGMA_AnalogueValue_f_2020A" dchg="false" dupd="false" fc="CF" qchg="false"/>

<DA bType="INT32U" name="operTimeout" dchg="false" dupd="false" fc="CF" qchg="false"/>

<DA bType="VisString255" name="d" dchg="false" dupd="false" fc="DC" qchg="false"/>

<DA bType="Unicode255" name="dU" dchg="false" dupd="false" fc="DC" qchg="false"/>

<DA bType="VisString255" name="dataNs" dchg="false" dupd="false" fc="EX" qchg="false"/>

</DOType>

<DOType id="SIGMA_ENS_SptEndSt_2020A" cdc="ENS">

<DA bType="Enum" name="stVal" type="SIGMA_SetpointEndKind_2020A" dchg="false" dupd="false" fc="ST" qchg="false"/>

<DA bType="Quality" name="q" dchg="false" dupd="false" fc="ST" qchg="false"/>

<DA bType="Timestamp" name="t" dchg="false" dupd="false" fc="ST" qchg="false"/>

<DA bType="VisString255" name="d" dchg="false" dupd="false" fc="DC" qchg="false"/>

<DA bType="Unicode255" name="dU" dchg="false" dupd="false" fc="DC" qchg="false"/>

</DOType>

<DOType id="SIGMA_ENS_Health_2020A" cdc="ENS">

<DA bType="Enum" name="stVal" type="SIGMA_HealthKind_2020A" dchg="false" dupd="false" fc="ST" qchg="false"/>

<DA bType="Quality" name="q" dchg="false" dupd="false" fc="ST" qchg="false"/>

<DA bType="Timestamp" name="t" dchg="false" dupd="false" fc="ST" qchg="false"/>

<DA bType="VisString255" name="d" dchg="false" dupd="false" fc="DC" qchg="false"/>

<DA bType="Unicode255" name="dU" dchg="false" dupd="false" fc="DC" qchg="false"/>

</DOType>

<DOType id="SIGMA_ASG_2020A" cdc="ASG">

<DA bType="Struct" name="setMag" type="SIGMA_AnalogueValue_f_2020A" dchg="false" dupd="false" fc="SP" qchg="false"/>

<DA bType="Struct" name="units" type="SIGMA_Units_2020A" dchg="false" dupd="false" fc="CF" qchg="false"/>

<DA bType="Struct" name="sVC" type="SIGMA_ScaledValueConfig_2020A" dchg="false" dupd="false" fc="CF" qchg="false"/>

<DA bType="Struct" name="minVal" type="SIGMA_AnalogueValue_f_2020A" dchg="false" dupd="false" fc="CF" qchg="false"/>

<DA bType="Struct" name="maxVal" type="SIGMA_AnalogueValue_f_2020A" dchg="false" dupd="false" fc="CF" qchg="false"/>

<DA bType="Struct" name="stepSize" type="SIGMA_AnalogueValue_f_2020A" dchg="false" dupd="false" fc="CF" qchg="false"/>

<DA bType="VisString255" name="d" dchg="false" dupd="false" fc="DC" qchg="false"/>

<DA bType="Unicode255" name="dU" dchg="false" dupd="false" fc="DC" qchg="false"/>

</DOType>
```

```
<DOType id="SIGMA_ASG_KV_2020A" cdc="ASG">
  <DA bType="Struct" name="setMag" type="SIGMA_AnalogueValue_f_2020A" dchg="false" dupd="false" fc="SP" qchg="false"/>
  <DA bType="Struct" name="units" type="SIGMA_Units_KV_2020A" dchg="false" dupd="false" fc="CF" qchg="false"/>
  <DA bType="Struct" name="sVC" type="SIGMA_ScaledValueConfig_2020A" dchg="false" dupd="false" fc="CF" qchg="false"/>
  <DA bType="Struct" name="minVal" type="SIGMA_AnalogueValue_f_2020A" dchg="false" dupd="false" fc="CF" qchg="false"/>
  <DA bType="Struct" name="maxVal" type="SIGMA_AnalogueValue_f_2020A" dchg="false" dupd="false" fc="CF" qchg="false"/>
  <DA bType="Struct" name="stepSize" type="SIGMA_AnalogueValue_f_2020A" dchg="false" dupd="false" fc="CF" qchg="false"/>
  <DA bType="VisString255" name="d" dchg="false" dupd="false" fc="DC" qchg="false"/>
  <DA bType="Unicode255" name="dU" dchg="false" dupd="false" fc="DC" qchg="false"/>
</DOType>
<DOType id="SIGMA_MV_KV_2020A" cdc="MV">
  <DA bType="Struct" name="mag" type="SIGMA_AnalogueValue_f_2020A" dchg="false" dupd="false" fc="MX" qchg="false"/>
  <DA bType="Enum" name="range" type="SIGMA_RangeKind_2020A" dchg="false" dupd="false" fc="MX" qchg="false"/>
  <DA bType="Quality" name="q" dchg="false" dupd="false" fc="MX" qchg="false"/>
  <DA bType="Timestamp" name="t" dchg="false" dupd="false" fc="MX" qchg="false"/>
  <DA bType="Struct" name="units" type="SIGMA_Units_MVar_2020A" dchg="false" dupd="false" fc="CF" qchg="false"/>
  <DA bType="INT32U" name="db" dchg="false" dupd="false" fc="CF" qchg="false"/>
  <DA bType="INT32U" name="zeroDb" dchg="false" dupd="false" fc="CF" qchg="false"/>
  <DA bType="Struct" name="sVC" type="SIGMA_ScaledValueConfig_2020A" dchg="false" dupd="false" fc="CF" qchg="false"/>
  <DA bType="Struct" name="rangeC" type="SIGMA_RangeConfig_2020A" dchg="false" dupd="false" fc="CF" qchg="false"/>
  <DA bType="INT32U" name="smpRate" dchg="false" dupd="false" fc="CF" qchg="false"/>
  <DA bType="VisString255" name="d" dchg="false" dupd="false" fc="DC" qchg="false"/>
  <DA bType="Unicode255" name="dU" dchg="false" dupd="false" fc="DC" qchg="false"/>
</DOType>
<DOType id="SIGMA_APC_KV_2020A" cdc="APC">
  <DA bType="Struct" name="origin" type="SIGMA_Originator_2020A" dchg="false" dupd="false" fc="MX" qchg="false"/>
  <DA bType="INT8U" name="ctlNum" dchg="false" dupd="false" fc="MX" qchg="false"/>
  <DA bType="Struct" name="Oper" type="SIGMA_Oper_APC_2020A" dchg="false" dupd="false" fc="CO" qchg="false"/>
  <DA bType="Struct" name="mxVal" type="SIGMA_AnalogueValue_f_2020A" dchg="false" dupd="false" fc="MX" qchg="false"/>
  <DA bType="Quality" name="q" dchg="false" dupd="false" fc="MX" qchg="false"/>
  <DA bType="Timestamp" name="t" dchg="false" dupd="false" fc="MX" qchg="false"/>
```

```xml
    <DA bType="BOOLEAN" name="stSeld" dchg="false" dupd="false" fc="MX" qchg="false"/>
    <DA bType="BOOLEAN" name="opRcvd" dchg="false" dupd="false" fc="OR" qchg="false"/>
    <DA bType="BOOLEAN" name="opOk" dchg="false" dupd="false" fc="OR" qchg="false"/>
    <DA bType="Timestamp" name="tOpOk" dchg="false" dupd="false" fc="OR" qchg="false"/>
    <DA bType="Enum" name="ctlModel" type="SIGMA_CtlModelKind_2020A" dchg="false" dupd="false" fc="CF" qchg="false"/>
    <DA bType="INT32U" name="sboTimeout" dchg="false" dupd="false" fc="CF" qchg="false"/>
    <DA bType="Enum" name="sboClass" type="SIGMA_SboClassKind_2020A" dchg="false" dupd="false" fc="CF" qchg="false"/>
    <DA bType="INT32U" name="db" dchg="false" dupd="false" fc="CF" qchg="false"/>
    <DA bType="Struct" name="units" type="SIGMA_Units_KV_2020A" dchg="false" dupd="false" fc="CF" qchg="false"/>
    <DA bType="Struct" name="sVC" type="SIGMA_ScaledValueConfig_2020A" dchg="false" dupd="false" fc="CF" qchg="false"/>
    <DA bType="Struct" name="minVal" type="SIGMA_AnalogueValue_f_2020A" dchg="false" dupd="false" fc="CF" qchg="false"/>
    <DA bType="Struct" name="maxVal" type="SIGMA_AnalogueValue_f_2020A" dchg="false" dupd="false" fc="CF" qchg="false"/>
    <DA bType="Struct" name="stepSize" type="SIGMA_AnalogueValue_f_2020A" dchg="false" dupd="false" fc="CF" qchg="false"/>
    <DA bType="INT32U" name="operTimeout" dchg="false" dupd="false" fc="CF" qchg="false"/>
    <DA bType="VisString255" name="d" dchg="false" dupd="false" fc="DC" qchg="false"/>
    <DA bType="Unicode255" name="dU" dchg="false" dupd="false" fc="DC" qchg="false"/>
    <DA bType="VisString255" name="dataNs" dchg="false" dupd="false" fc="EX" qchg="false"/>
  </DOType>
  <DOType id="SIGMA_MV_MW_2020A" cdc="MV">
    <DA bType="Struct" name="mag" type="SIGMA_AnalogueValue_f_2020A" dchg="false" dupd="false" fc="MX" qchg="false"/>
    <DA bType="Enum" name="range" type="SIGMA_RangeKind_2020A" dchg="false" dupd="false" fc="MX" qchg="false"/>
    <DA bType="Quality" name="q" dchg="false" dupd="false" fc="MX" qchg="false"/>
    <DA bType="Timestamp" name="t" dchg="false" dupd="false" fc="MX" qchg="false"/>
    <DA bType="Struct" name="units" type="SIGMA_Units_MW_2020A" dchg="false" dupd="false" fc="CF" qchg="false"/>
    <DA bType="INT32U" name="db" dchg="false" dupd="false" fc="CF" qchg="false"/>
    <DA bType="INT32U" name="zeroDb" dchg="false" dupd="false" fc="CF" qchg="false"/>
    <DA bType="Struct" name="sVC" type="SIGMA_ScaledValueConfig_2020A" dchg="false" dupd="false" fc="CF" qchg="false"/>
    <DA bType="Struct" name="rangeC" type="SIGMA_RangeConfig_2020A" dchg="false" dupd="false" fc="CF" qchg="false"/>
    <DA bType="INT32U" name="smpRate" dchg="false" dupd="false" fc="CF" qchg="false"/>
    <DA bType="VisString255" name="d" dchg="false" dupd="false" fc="DC" qchg="false"/>
    <DA bType="Unicode255" name="dU" dchg="false" dupd="false" fc="DC" qchg="false"/>
```

```xml
  </DOType>
  <DOType id="SIGMA_APC_MW_2020A" cdc="APC">
    <DA bType="Struct" name="origin" type="SIGMA_Originator_2020A" dchg="false" dupd="false" fc="MX"
qchg="false"/>
    <DA bType="INT8U" name="ctlNum" dchg="false" dupd="false" fc="MX" qchg="false"/>
    <DA bType="Struct" name="Oper" type="SIGMA_Oper_APC_2020A" dchg="false" dupd="false" fc="CO"
qchg="false"/>
    <DA bType="Struct" name="mxVal" type="SIGMA_AnalogueValue_f_2020A" dchg="false" dupd="false" fc="MX"
qchg="false"/>
    <DA bType="Quality" name="q" dchg="false" dupd="false" fc="MX" qchg="false"/>
    <DA bType="Timestamp" name="t" dchg="false" dupd="false" fc="MX" qchg="false"/>
    <DA bType="BOOLEAN" name="stSeld" dchg="false" dupd="false" fc="MX" qchg="false"/>
    <DA bType="BOOLEAN" name="opRcvd" dchg="false" dupd="false" fc="OR" qchg="false"/>
    <DA bType="BOOLEAN" name="opOk" dchg="false" dupd="false" fc="OR" qchg="false"/>
    <DA bType="Timestamp" name="tOpOk" dchg="false" dupd="false" fc="OR" qchg="false"/>
    <DA bType="Enum" name="ctlModel" type="SIGMA_CtlModelKind_2020A" dchg="false" dupd="false" fc="CF"
qchg="false"/>
    <DA bType="INT32U" name="sboTimeout" dchg="false" dupd="false" fc="CF" qchg="false"/>
    <DA bType="Enum" name="sboClass" type="SIGMA_SboClassKind_2020A" dchg="false" dupd="false" fc="CF"
qchg="false"/>
    <DA bType="INT32U" name="db" dchg="false" dupd="false" fc="CF" qchg="false"/>
    <DA bType="Struct" name="units" type="SIGMA_Units_MW_2020A" dchg="false" dupd="false" fc="CF"
qchg="false"/>
    <DA bType="Struct" name="sVC" type="SIGMA_ScaledValueConfig_2020A" dchg="false" dupd="false" fc="CF"
qchg="false"/>
    <DA bType="Struct" name="minVal" type="SIGMA_AnalogueValue_f_2020A" dchg="false" dupd="false" fc="CF"
qchg="false"/>
    <DA bType="Struct" name="maxVal" type="SIGMA_AnalogueValue_f_2020A" dchg="false" dupd="false" fc="CF"
qchg="false"/>
    <DA bType="Struct" name="stepSize" type="SIGMA_AnalogueValue_f_2020A" dchg="false" dupd="false" fc="CF"
qchg="false"/>
    <DA bType="INT32U" name="operTimeout" dchg="false" dupd="false" fc="CF" qchg="false"/>
    <DA bType="VisString255" name="d" dchg="false" dupd="false" fc="DC" qchg="false"/>
    <DA bType="Unicode255" name="dU" dchg="false" dupd="false" fc="DC" qchg="false"/>
    <DA bType="VisString255" name="dataNs" dchg="false" dupd="false" fc="EX" qchg="false"/>
  </DOType>
  <DOType id="SIGMA_ASG_MW_2020A" cdc="ASG">
    <DA bType="Struct" name="setMag" type="SIGMA_AnalogueValue_f_2020A" dchg="false" dupd="false" fc="SP"
qchg="false"/>
    <DA bType="Struct" name="units" type="SIGMA_Units_MW_2020A" dchg="false" dupd="false" fc="CF"
qchg="false"/>
    <DA bType="Struct" name="sVC" type="SIGMA_ScaledValueConfig_2020A" dchg="false" dupd="false" fc="CF"
qchg="false"/>
    <DA bType="Struct" name="minVal" type="SIGMA_AnalogueValue_f_2020A" dchg="false" dupd="false" fc="CF"
qchg="false"/>
```

```xml
<DA bType="Struct" name="maxVal" type="SIGMA_AnalogueValue_f_2020A" dchg="false" dupd="false" fc="CF" qchg="false"/>

<DA bType="Struct" name="stepSize" type="SIGMA_AnalogueValue_f_2020A" dchg="false" dupd="false" fc="CF" qchg="false"/>

<DA bType="VisString255" name="d" dchg="false" dupd="false" fc="DC" qchg="false"/>

<DA bType="Unicode255" name="dU" dchg="false" dupd="false" fc="DC" qchg="false"/>

</DOType>

<DOType id="SIGMA_LPL_2020A" cdc="LPL">

<DA bType="VisString255" name="vendor" dchg="false" dupd="false" fc="DC" qchg="false"/>

<DA bType="VisString255" name="swRev" dchg="false" dupd="false" fc="DC" qchg="false"/>

<DA bType="VisString255" name="d" dchg="false" dupd="false" fc="DC" qchg="false"/>

<DA bType="Unicode255" name="dU" dchg="false" dupd="false" fc="DC" qchg="false"/>

<DA bType="VisString255" name="configRev" dchg="false" dupd="false" fc="DC" qchg="false"/>

<DA bType="INT32" name="paramRev" dchg="false" dupd="false" fc="ST" qchg="false"/>

<DA bType="INT32" name="valRev" dchg="false" dupd="false" fc="ST" qchg="false"/>

</DOType>

<DOType id="SIGMA_CMV_2020A" cdc="CMV">

<DA bType="Struct" name="instCVal" type="SIGMA_Vector_mag_ang_2020A" dchg="false" dupd="false" fc="MX" qchg="false"/>

<DA bType="Struct" name="cVal" type="SIGMA_Vector_mag_ang_2020A" dchg="false" dupd="false" fc="MX" qchg="false"/>

<DA bType="Enum" name="range" type="SIGMA_RangeKind_2020A" dchg="false" dupd="false" fc="MX" qchg="false"/>

<DA bType="Quality" name="q" dchg="false" dupd="false" fc="MX" qchg="false"/>

<DA bType="Timestamp" name="t" dchg="false" dupd="false" fc="MX" qchg="false"/>

<DA bType="Struct" name="units" type="SIGMA_Units_2020A" dchg="false" dupd="false" fc="CF" qchg="false"/>

<DA bType="INT32U" name="db" dchg="false" dupd="false" fc="CF" qchg="false"/>

<DA bType="INT32U" name="dbAng" dchg="false" dupd="false" fc="CF" qchg="false"/>

<DA bType="INT32U" name="zeroDb" dchg="false" dupd="false" fc="CF" qchg="false"/>

<DA bType="Struct" name="rangeC" type="SIGMA_RangeConfig_2020A" dchg="false" dupd="false" fc="CF" qchg="false"/>

<DA bType="Struct" name="magSVC" type="SIGMA_ScaledValueConfig_2020A" dchg="false" dupd="false" fc="CF" qchg="false"/>

<DA bType="Struct" name="angSVC" type="SIGMA_ScaledValueConfig_2020A" dchg="false" dupd="false" fc="CF" qchg="false"/>

<DA bType="Enum" name="angRef" type="SIGMA_PhaseAngleReferenceKind_2020A" dchg="false" dupd="false" fc="CF" qchg="false"/>

<DA bType="INT32U" name="smpRate" dchg="false" dupd="false" fc="CF" qchg="false"/>

<DA bType="VisString255" name="d" dchg="false" dupd="false" fc="DC" qchg="false"/>

<DA bType="Unicode255" name="dU" dchg="false" dupd="false" fc="DC" qchg="false"/>

</DOType>

<DOType id="SIGMA_MV_Hz_2020A" cdc="MV">
```

```
<DA bType="Struct" name="mag" type="SIGMA_AnalogueValue_f_2020A" dchg="false" dupd="false" fc="MX" qchg="false"/>
<DA bType="Enum" name="range" type="SIGMA_RangeKind_2020A" dchg="false" dupd="false" fc="MX" qchg="false"/>
<DA bType="Quality" name="q" dchg="false" dupd="false" fc="MX" qchg="false"/>
<DA bType="Timestamp" name="t" dchg="false" dupd="false" fc="MX" qchg="false"/>
<DA bType="Struct" name="units" type="SIGMA_Units_Hz_2020A" dchg="false" dupd="false" fc="CF" qchg="false"/>
<DA bType="INT32U" name="db" dchg="false" dupd="false" fc="CF" qchg="false"/>
<DA bType="INT32U" name="zeroDb" dchg="false" dupd="false" fc="CF" qchg="false"/>
<DA bType="Struct" name="sVC" type="SIGMA_ScaledValueConfig_2020A" dchg="false" dupd="false" fc="CF" qchg="false"/>
<DA bType="Struct" name="rangeC" type="SIGMA_RangeConfig_2020A" dchg="false" dupd="false" fc="CF" qchg="false"/>
<DA bType="INT32U" name="smpRate" dchg="false" dupd="false" fc="CF" qchg="false"/>
<DA bType="VisString255" name="d" dchg="false" dupd="false" fc="DC" qchg="false"/>
<DA bType="Unicode255" name="dU" dchg="false" dupd="false" fc="DC" qchg="false"/>
</DOType>
<DOType id="SIGMA_DEL_2020A" cdc="DEL">
<SDO type="SIGMA_CMV_2020A" name="phsAB"/>
<SDO type="SIGMA_CMV_2020A" name="phsBC"/>
<SDO type="SIGMA_CMV_2020A" name="phsCA"/>
<DA bType="Enum" name="angRef" type="SIGMA_PhaseAngleReferenceKind_2020A" dchg="false" dupd="false" fc="CF" qchg="false"/>
<DA bType="VisString255" name="d" dchg="false" dupd="false" fc="DC" qchg="false"/>
<DA bType="Unicode255" name="dU" dchg="false" dupd="false" fc="DC" qchg="false"/>
</DOType>
<DOType id="SIGMA_CSG_OpeningWattHead_2020A" cdc="CSD">
<DA bType="Struct" name="xUnits" type="SIGMA_Units_2020A" dchg="false" dupd="false" fc="DC" qchg="false"/>
<DA bType="VisString255" name="xD" dchg="false" dupd="false" fc="DC" qchg="false"/>
<DA bType="Unicode255" name="xDU" dchg="false" dupd="false" fc="DC" qchg="false"/>
<DA bType="Struct" name="yUnits" type="SIGMA_Units_MW_2020A" dchg="false" dupd="false" fc="DC" qchg="false"/>
<DA bType="VisString255" name="yD" dchg="false" dupd="false" fc="DC" qchg="false"/>
<DA bType="Unicode255" name="yDU" dchg="false" dupd="false" fc="DC" qchg="false"/>
<DA bType="Struct" name="zUnits" type="SIGMA_Units_Metre_2020A" dchg="false" dupd="false" fc="DC" qchg="false"/>
<DA bType="VisString255" name="zD" dchg="false" dupd="false" fc="DC" qchg="false"/>
<DA bType="Unicode255" name="zDU" dchg="false" dupd="false" fc="DC" qchg="false"/>
<DA bType="INT16U" name="maxPts" dchg="false" dupd="false" fc="DC" qchg="false"/>
<DA bType="Struct" name="crvPts" type="SIGMA_Point_2020A" dchg="false" dupd="false" fc="DC" qchg="false"/>
```

```xml
<DA bType="VisString255" name="d" dchg="false" dupd="false" fc="DC" qchg="false"/>
<DA bType="Unicode255" name="dU" dchg="false" dupd="false" fc="DC" qchg="false"/>
</DOType>
<DOType id="SIGMA_MV_Flw_2020A" cdc="MV">
<DA bType="Struct" name="mag" type="SIGMA_AnalogueValue_f_2020A" dchg="false" dupd="false" fc="MX" qchg="false"/>
<DA bType="Enum" name="range" type="SIGMA_RangeKind_2020A" dchg="false" dupd="false" fc="MX" qchg="false"/>
<DA bType="Quality" name="q" dchg="false" dupd="false" fc="MX" qchg="false"/>
<DA bType="Timestamp" name="t" dchg="false" dupd="false" fc="MX" qchg="false"/>
<DA bType="Struct" name="units" type="SIGMA_Units_2020A" dchg="false" dupd="false" fc="CF" qchg="false"/>
<DA bType="INT32U" name="db" dchg="false" dupd="false" fc="CF" qchg="false"/>
<DA bType="INT32U" name="zeroDb" dchg="false" dupd="false" fc="CF" qchg="false"/>
<DA bType="Struct" name="sVC" type="SIGMA_ScaledValueConfig_2020A" dchg="false" dupd="false" fc="CF" qchg="false"/>
<DA bType="Struct" name="rangeC" type="SIGMA_RangeConfig_2020A" dchg="false" dupd="false" fc="CF" qchg="false"/>
<DA bType="INT32U" name="smpRate" dchg="false" dupd="false" fc="CF" qchg="false"/>
<DA bType="VisString255" name="d" dchg="false" dupd="false" fc="DC" qchg="false"/>
<DA bType="Unicode255" name="dU" dchg="false" dupd="false" fc="DC" qchg="false"/>
</DOType>
<DOType id="SIGMA_MV_metre_2020A" cdc="MV">
<DA bType="Struct" name="mag" type="SIGMA_AnalogueValue_f_2020A" dchg="false" dupd="false" fc="MX" qchg="false"/>
<DA bType="Enum" name="range" type="SIGMA_RangeKind_2020A" dchg="false" dupd="false" fc="MX" qchg="false"/>
<DA bType="Quality" name="q" dchg="false" dupd="false" fc="MX" qchg="false"/>
<DA bType="Timestamp" name="t" dchg="false" dupd="false" fc="MX" qchg="false"/>
<DA bType="Struct" name="units" type="SIGMA_Units_Metre_2020A" dchg="false" dupd="false" fc="CF" qchg="false"/>
<DA bType="INT32U" name="db" dchg="false" dupd="false" fc="CF" qchg="false"/>
<DA bType="INT32U" name="zeroDb" dchg="false" dupd="false" fc="CF" qchg="false"/>
<DA bType="Struct" name="sVC" type="SIGMA_ScaledValueConfig_2020A" dchg="false" dupd="false" fc="CF" qchg="false"/>
<DA bType="Struct" name="rangeC" type="SIGMA_RangeConfig_2020A" dchg="false" dupd="false" fc="CF" qchg="false"/>
<DA bType="INT32U" name="smpRate" dchg="false" dupd="false" fc="CF" qchg="false"/>
<DA bType="VisString255" name="d" dchg="false" dupd="false" fc="DC" qchg="false"/>
<DA bType="Unicode255" name="dU" dchg="false" dupd="false" fc="DC" qchg="false"/>
</DOType>
<DOType id="SIGMA_ASG_Flw_2020A" cdc="ASG">
```

```
<DA bType="Struct" name="setMag" type="SIGMA_AnalogueValue_f_2020A" dchg="false" dupd="false" fc="SP" qchg="false"/>

<DA bType="Struct" name="units" type="SIGMA_Units_flw_2020A" dchg="false" dupd="false" fc="CF" qchg="false"/>

<DA bType="Struct" name="sVC" type="SIGMA_ScaledValueConfig_2020A" dchg="false" dupd="false" fc="CF" qchg="false"/>

<DA bType="Struct" name="minVal" type="SIGMA_AnalogueValue_f_2020A" dchg="false" dupd="false" fc="CF" qchg="false"/>

<DA bType="Struct" name="maxVal" type="SIGMA_AnalogueValue_f_2020A" dchg="false" dupd="false" fc="CF" qchg="false"/>

<DA bType="Struct" name="stepSize" type="SIGMA_AnalogueValue_f_2020A" dchg="false" dupd="false" fc="CF" qchg="false"/>

<DA bType="VisString255" name="d" dchg="false" dupd="false" fc="DC" qchg="false"/>

<DA bType="Unicode255" name="dU" dchg="false" dupd="false" fc="DC" qchg="false"/>

</DOType>

<DOType id="SIGMA_APC_Amp_2020A" cdc="APC">

<DA bType="Struct" name="origin" type="SIGMA_Originator_2020A" dchg="false" dupd="false" fc="MX" qchg="false"/>

<DA bType="INT8U" name="ctlNum" dchg="false" dupd="false" fc="MX" qchg="false"/>

<DA bType="Struct" name="Oper" type="SIGMA_Oper_APC_2020A" dchg="false" dupd="false" fc="CO" qchg="false"/>

<DA bType="Struct" name="mxVal" type="SIGMA_AnalogueValue_f_2020A" dchg="false" dupd="false" fc="MX" qchg="false"/>

<DA bType="Quality" name="q" dchg="false" dupd="false" fc="MX" qchg="false"/>

<DA bType="Timestamp" name="t" dchg="false" dupd="false" fc="MX" qchg="false"/>

<DA bType="BOOLEAN" name="stSeld" dchg="false" dupd="false" fc="MX" qchg="false"/>

<DA bType="BOOLEAN" name="opRcvd" dchg="false" dupd="false" fc="OR" qchg="false"/>

<DA bType="BOOLEAN" name="opOk" dchg="false" dupd="false" fc="OR" qchg="false"/>

<DA bType="Timestamp" name="tOpOk" dchg="false" dupd="false" fc="OR" qchg="false"/>

<DA bType="Enum" name="ctlModel" type="SIGMA_CtlModelKind_2020A" dchg="false" dupd="false" fc="CF" qchg="false"/>

<DA bType="INT32U" name="sboTimeout" dchg="false" dupd="false" fc="CF" qchg="false"/>

<DA bType="Enum" name="sboClass" type="SIGMA_SboClassKind_2020A" dchg="false" dupd="false" fc="CF" qchg="false"/>

<DA bType="INT32U" name="db" dchg="false" dupd="false" fc="CF" qchg="false"/>

<DA bType="Struct" name="units" type="SIGMA_Units_Amp_2020A" dchg="false" dupd="false" fc="CF" qchg="false"/>

<DA bType="Struct" name="sVC" type="SIGMA_ScaledValueConfig_2020A" dchg="false" dupd="false" fc="CF" qchg="false"/>

<DA bType="Struct" name="minVal" type="SIGMA_AnalogueValue_f_2020A" dchg="false" dupd="false" fc="CF" qchg="false"/>

<DA bType="Struct" name="maxVal" type="SIGMA_AnalogueValue_f_2020A" dchg="false" dupd="false" fc="CF" qchg="false"/>

<DA bType="Struct" name="stepSize" type="SIGMA_AnalogueValue_f_2020A" dchg="false" dupd="false" fc="CF" qchg="false"/>

<DA bType="INT32U" name="operTimeout" dchg="false" dupd="false" fc="CF" qchg="false"/>
```

```xml
    <DA bType="VisString255" name="d" dchg="false" dupd="false" fc="DC" qchg="false"/>
    <DA bType="Unicode255" name="dU" dchg="false" dupd="false" fc="DC" qchg="false"/>
    <DA bType="VisString255" name="dataNs" dchg="false" dupd="false" fc="EX" qchg="false"/>
  </DOType>
  <DOType id="SIGMA_ASG_metre_2020A" cdc="ASG">
    <DA bType="Struct" name="setMag" type="SIGMA_AnalogueValue_f_2020A" dchg="false" dupd="false" fc="SP" qchg="false"/>
    <DA bType="Struct" name="units" type="SIGMA_Units_Metre_2020A" dchg="false" dupd="false" fc="CF" qchg="false"/>
    <DA bType="Struct" name="sVC" type="SIGMA_ScaledValueConfig_2020A" dchg="false" dupd="false" fc="CF" qchg="false"/>
    <DA bType="Struct" name="minVal" type="SIGMA_AnalogueValue_f_2020A" dchg="false" dupd="false" fc="CF" qchg="false"/>
    <DA bType="Struct" name="maxVal" type="SIGMA_AnalogueValue_f_2020A" dchg="false" dupd="false" fc="CF" qchg="false"/>
    <DA bType="Struct" name="stepSize" type="SIGMA_AnalogueValue_f_2020A" dchg="false" dupd="false" fc="CF" qchg="false"/>
    <DA bType="VisString255" name="d" dchg="false" dupd="false" fc="DC" qchg="false"/>
    <DA bType="Unicode255" name="dU" dchg="false" dupd="false" fc="DC" qchg="false"/>
  </DOType>
  <DOType id="SIGMA_APC_metre_2020A" cdc="APC">
    <DA bType="Struct" name="origin" type="SIGMA_Originator_2020A" dchg="false" dupd="false" fc="MX" qchg="false"/>
    <DA bType="INT8U" name="ctlNum" dchg="false" dupd="false" fc="MX" qchg="false"/>
    <DA bType="Struct" name="Oper" type="SIGMA_Oper_APC_2020A" dchg="false" dupd="false" fc="CO" qchg="false"/>
    <DA bType="Struct" name="mxVal" type="SIGMA_AnalogueValue_f_2020A" dchg="false" dupd="false" fc="MX" qchg="false"/>
    <DA bType="Quality" name="q" dchg="false" dupd="false" fc="MX" qchg="false"/>
    <DA bType="Timestamp" name="t" dchg="false" dupd="false" fc="MX" qchg="false"/>
    <DA bType="BOOLEAN" name="stSeld" dchg="false" dupd="false" fc="MX" qchg="false"/>
    <DA bType="BOOLEAN" name="opRcvd" dchg="false" dupd="false" fc="OR" qchg="false"/>
    <DA bType="BOOLEAN" name="opOk" dchg="false" dupd="false" fc="OR" qchg="false"/>
    <DA bType="Timestamp" name="tOpOk" dchg="false" dupd="false" fc="OR" qchg="false"/>
    <DA bType="Enum" name="ctlModel" type="SIGMA_CtlModelKind_2020A" dchg="false" dupd="false" fc="CF" qchg="false"/>
    <DA bType="INT32U" name="sboTimeout" dchg="false" dupd="false" fc="CF" qchg="false"/>
    <DA bType="Enum" name="sboClass" type="SIGMA_SboClassKind_2020A" dchg="false" dupd="false" fc="CF" qchg="false"/>
    <DA bType="INT32U" name="db" dchg="false" dupd="false" fc="CF" qchg="false"/>
    <DA bType="Struct" name="units" type="SIGMA_Units_Metre_2020A" dchg="false" dupd="false" fc="CF" qchg="false"/>
    <DA bType="Struct" name="sVC" type="SIGMA_ScaledValueConfig_2020A" dchg="false" dupd="false" fc="CF" qchg="false"/>
```

```
<DA bType="Struct" name="minVal" type="SIGMA_AnalogueValue_f_2020A" dchg="false" dupd="false" fc="CF" qchg="false"/>
<DA bType="Struct" name="maxVal" type="SIGMA_AnalogueValue_f_2020A" dchg="false" dupd="false" fc="CF" qchg="false"/>
<DA bType="Struct" name="stepSize" type="SIGMA_AnalogueValue_f_2020A" dchg="false" dupd="false" fc="CF" qchg="false"/>
<DA bType="INT32U" name="operTimeout" dchg="false" dupd="false" fc="CF" qchg="false"/>
<DA bType="VisString255" name="d" dchg="false" dupd="false" fc="DC" qchg="false"/>
<DA bType="Unicode255" name="dU" dchg="false" dupd="false" fc="DC" qchg="false"/>
<DA bType="VisString255" name="dataNs" dchg="false" dupd="false" fc="EX" qchg="false"/>
</DOType>
<DOType id="SIGMA_DPL_2020A" cdc="DPL">
<DA bType="VisString255" name="vendor" dchg="false" dupd="false" fc="DC" qchg="false"/>
<DA bType="VisString255" name="hwRev" dchg="false" dupd="false" fc="DC" qchg="false"/>
<DA bType="VisString255" name="swRev" dchg="false" dupd="false" fc="DC" qchg="false"/>
<DA bType="VisString255" name="serNum" dchg="false" dupd="false" fc="DC" qchg="false"/>
<DA bType="VisString255" name="model" dchg="false" dupd="false" fc="DC" qchg="false"/>
<DA bType="VisString255" name="location" dchg="false" dupd="false" fc="DC" qchg="false"/>
<DA bType="VisString64" name="name" dchg="false" dupd="false" fc="DC" qchg="false"/>
<DA bType="VisString255" name="owner" dchg="false" dupd="false" fc="DC" qchg="false"/>
<DA bType="VisString255" name="ePSName" dchg="false" dupd="false" fc="DC" qchg="false"/>
<DA bType="VisString255" name="primeOper" dchg="false" dupd="false" fc="DC" qchg="false"/>
<DA bType="VisString255" name="secondOper" dchg="false" dupd="false" fc="DC" qchg="false"/>
<DA bType="FLOAT32" name="latitude" dchg="false" dupd="false" fc="DC" qchg="false"/>
<DA bType="FLOAT32" name="longitude" dchg="false" dupd="false" fc="DC" qchg="false"/>
<DA bType="FLOAT32" name="altitude" dchg="false" dupd="false" fc="DC" qchg="false"/>
<DA bType="VisString255" name="mRID" dchg="false" dupd="false" fc="DC" qchg="false"/>
<DA bType="VisString255" name="d" dchg="false" dupd="false" fc="DC" qchg="false"/>
<DA bType="Unicode255" name="dU" dchg="false" dupd="false" fc="DC" qchg="false"/>
</DOType>
<DOType id="SIGMA_MV_Tmp_2020A" cdc="MV">
<DA bType="Struct" name="mag" type="SIGMA_AnalogueValue_f_2020A" dchg="false" dupd="false" fc="MX" qchg="false"/>
<DA bType="Enum" name="range" type="SIGMA_RangeKind_2020A" dchg="false" dupd="false" fc="MX" qchg="false"/>
<DA bType="Quality" name="q" dchg="false" dupd="false" fc="MX" qchg="false"/>
<DA bType="Timestamp" name="t" dchg="false" dupd="false" fc="MX" qchg="false"/>
<DA bType="Struct" name="units" type="SIGMA_Units_Tmp_2020A" dchg="false" dupd="false" fc="CF" qchg="false"/>
<DA bType="INT32U" name="db" dchg="false" dupd="false" fc="CF" qchg="false"/>
```

```
<DA bType="INT32U" name="zeroDb" dchg="false" dupd="false" fc="CF" qchg="false"/>
<DA bType="Struct" name="sVC" type="SIGMA_ScaledValueConfig_2020A" dchg="false" dupd="false" fc="CF" qchg="false"/>
<DA bType="Struct" name="rangeC" type="SIGMA_RangeConfig_2020A" dchg="false" dupd="false" fc="CF" qchg="false"/>
<DA bType="VisString255" name="d" dchg="false" dupd="false" fc="DC" qchg="false"/>
<DA bType="Unicode255" name="dU" dchg="false" dupd="false" fc="DC" qchg="false"/>
</DOType>
<DOType id="SIGMA_BCR_2020A" cdc="BCR">
<DA bType="INT64" name="actVal" dchg="false" dupd="false" fc="ST" qchg="false"/>
<DA bType="INT64" name="frVal" dchg="false" dupd="false" fc="ST" qchg="false"/>
<DA bType="Timestamp" name="frTm" dchg="false" dupd="false" fc="ST" qchg="false"/>
<DA bType="Quality" name="q" dchg="false" dupd="false" fc="ST" qchg="false"/>
<DA bType="Timestamp" name="t" dchg="false" dupd="false" fc="ST" qchg="false"/>
<DA bType="Struct" name="units" type="SIGMA_Units_2020A" dchg="false" dupd="false" fc="CF" qchg="false"/>
<DA bType="FLOAT32" name="pulsQty" dchg="false" dupd="false" fc="CF" qchg="false"/>
<DA bType="BOOLEAN" name="frEna" dchg="false" dupd="false" fc="CF" qchg="false"/>
<DA bType="Timestamp" name="strTm" dchg="false" dupd="false" fc="CF" qchg="false"/>
<DA bType="INT32" name="frPd" dchg="false" dupd="false" fc="CF" qchg="false"/>
<DA bType="BOOLEAN" name="frRs" dchg="false" dupd="false" fc="CF" qchg="false"/>
<DA bType="VisString255" name="d" dchg="false" dupd="false" fc="DC" qchg="false"/>
<DA bType="Unicode255" name="dU" dchg="false" dupd="false" fc="DC" qchg="false"/>
</DOType>
<DOType id="SIGMA_SAV_2020A" cdc="SAV">
<DA bType="Struct" name="instMag" type="SIGMA_AnalogueValue_f_2020A" dchg="false" dupd="false" fc="MX" qchg="false"/>
<DA bType="Quality" name="q" dchg="false" dupd="false" fc="MX" qchg="false"/>
<DA bType="Timestamp" name="t" dchg="false" dupd="false" fc="MX" qchg="false"/>
<DA bType="Struct" name="units" type="SIGMA_Units_2020A" dchg="false" dupd="false" fc="CF" qchg="false"/>
<DA bType="Struct" name="sVC" type="SIGMA_ScaledValueConfig_2020A" dchg="false" dupd="false" fc="CF" qchg="false"/>
<DA bType="Struct" name="min" type="SIGMA_AnalogueValue_f_2020A" dchg="false" dupd="false" fc="CF" qchg="false"/>
<DA bType="Struct" name="max" type="SIGMA_AnalogueValue_f_2020A" dchg="false" dupd="false" fc="CF" qchg="false"/>
<DA bType="VisString255" name="d" dchg="false" dupd="false" fc="DC" qchg="false"/>
<DA bType="Unicode255" name="dU" dchg="false" dupd="false" fc="DC" qchg="false"/>
</DOType>
<DOType id="SIGMA_ING_2020A" cdc="ING">
<DA bType="INT32" name="setVal" dchg="false" dupd="false" fc="SP" qchg="false"/>
```

```
      <DA bType="INT32" name="minVal" dchg="false" dupd="false" fc="CF" qchg="false"/>
      <DA bType="INT32" name="maxVal" dchg="false" dupd="false" fc="CF" qchg="false"/>
      <DA bType="INT32U" name="stepSize" dchg="false" dupd="false" fc="CF" qchg="false"/>
      <DA bType="Struct" name="units" type="SIGMA_Units_2020A" dchg="false" dupd="false" fc="CF" qchg="false"/>
      <DA bType="VisString255" name="d" dchg="false" dupd="false" fc="DC" qchg="false"/>
      <DA bType="Unicode255" name="dU" dchg="false" dupd="false" fc="DC" qchg="false"/>
    </DOType>
    <DOType id="SIGMA_INC_2020A" cdc="INC">
      <DA bType="Struct" name="origin" type="SIGMA_Originator_2020A" dchg="false" dupd="false" fc="ST" qchg="false"/>
      <DA bType="INT8U" name="ctlNum" dchg="false" dupd="false" fc="ST" qchg="false"/>
      <DA bType="Struct" name="Oper" type="SIGMA_Oper_INC_2020A" dchg="false" dupd="false" fc="CO" qchg="false"/>
      <DA bType="INT32" name="stVal" dchg="false" dupd="false" fc="ST" qchg="false"/>
      <DA bType="Quality" name="q" dchg="false" dupd="false" fc="ST" qchg="false"/>
      <DA bType="Timestamp" name="t" dchg="false" dupd="false" fc="ST" qchg="false"/>
      <DA bType="BOOLEAN" name="stSeld" dchg="false" dupd="false" fc="ST" qchg="false"/>
      <DA bType="BOOLEAN" name="opRcvd" dchg="false" dupd="false" fc="OR" qchg="false"/>
      <DA bType="BOOLEAN" name="opOk" dchg="false" dupd="false" fc="OR" qchg="false"/>
      <DA bType="Timestamp" name="tOpOk" dchg="false" dupd="false" fc="OR" qchg="false"/>
      <DA bType="Enum" name="ctlModel" type="SIGMA_CtlModelKind_2020A" dchg="false" dupd="false" fc="CF" qchg="false"/>
      <DA bType="INT32U" name="sboTimeout" dchg="false" dupd="false" fc="CF" qchg="false"/>
      <DA bType="Enum" name="sboClass" type="SIGMA_SboClassKind_2020A" dchg="false" dupd="false" fc="CF" qchg="false"/>
      <DA bType="INT32" name="minVal" dchg="false" dupd="false" fc="CF" qchg="false"/>
      <DA bType="INT32" name="maxVal" dchg="false" dupd="false" fc="CF" qchg="false"/>
      <DA bType="INT32U" name="stepSize" dchg="false" dupd="false" fc="CF" qchg="false"/>
      <DA bType="INT32U" name="operTimeout" dchg="false" dupd="false" fc="CF" qchg="false"/>
      <DA bType="Struct" name="units" type="SIGMA_Units_2020A" dchg="false" dupd="false" fc="CF" qchg="false"/>
      <DA bType="VisString255" name="d" dchg="false" dupd="false" fc="DC" qchg="false"/>
      <DA bType="Unicode255" name="dU" dchg="false" dupd="false" fc="DC" qchg="false"/>
      <DA bType="VisString255" name="dataNs" dchg="false" dupd="false" fc="EX" qchg="false"/>
    </DOType>
    <DOType id="SIGMA_MV_MPa_2020A" cdc="MV">
      <DA bType="Struct" name="instMag" type="SIGMA_AnalogueValue_f_2020A" dchg="false" dupd="false" fc="MX" qchg="false"/>
      <DA bType="Struct" name="mag" type="SIGMA_AnalogueValue_f_2020A" dchg="false" dupd="false" fc="MX" qchg="false"/>
      <DA bType="Enum" name="range" type="SIGMA_RangeKind_2020A" dchg="false" dupd="false" fc="MX" qchg="false"/>
```

```
<DA bType="Quality" name="q" dchg="false" dupd="false" fc="MX" qchg="false"/>
<DA bType="Timestamp" name="t" dchg="false" dupd="false" fc="MX" qchg="false"/>
<DA bType="Struct" name="units" type="SIGMA_Units_MPa_2020A" dchg="false" dupd="false" fc="CF" qchg="false"/>
<DA bType="INT32U" name="db" dchg="false" dupd="false" fc="CF" qchg="false"/>
<DA bType="INT32U" name="zeroDb" dchg="false" dupd="false" fc="CF" qchg="false"/>
<DA bType="Struct" name="sVC" type="SIGMA_ScaledValueConfig_2020A" dchg="false" dupd="false" fc="CF" qchg="false"/>
<DA bType="Struct" name="rangeC" type="SIGMA_RangeConfig_2020A" dchg="false" dupd="false" fc="CF" qchg="false"/>
<DA bType="INT32U" name="smpRate" dchg="false" dupd="false" fc="CF" qchg="false"/>
<DA bType="VisString255" name="d" dchg="false" dupd="false" fc="DC" qchg="false"/>
<DA bType="Unicode255" name="dU" dchg="false" dupd="false" fc="DC" qchg="false"/>
</DOType>
<DOType id="SIGMA_ENC_HdrCtlMod_2020A" cdc="ENC">
<DA bType="Struct" name="origin" type="SIGMA_Originator_2020A" dchg="false" dupd="false" fc="ST" qchg="false"/>
<DA bType="INT8U" name="ctlNum" dchg="false" dupd="false" fc="ST" qchg="false"/>
<DA bType="Enum" name="stVal" type="SIGMA_HdrControlModeKind_2020A" dchg="false" dupd="false" fc="ST" qchg="false"/>
<DA bType="Quality" name="q" dchg="false" dupd="false" fc="ST" qchg="false"/>
<DA bType="Timestamp" name="t" dchg="false" dupd="false" fc="ST" qchg="false"/>
<DA bType="BOOLEAN" name="stSeld" dchg="false" dupd="false" fc="ST" qchg="false"/>
<DA bType="BOOLEAN" name="opOk" dchg="false" dupd="false" fc="OR" qchg="false"/>
<DA bType="Timestamp" name="tOpOk" dchg="false" dupd="false" fc="OR" qchg="false"/>
<DA bType="Enum" name="ctlModel" type="SIGMA_CtlModelKind_2020A" dchg="false" dupd="false" fc="CF" qchg="false"/>
<DA bType="INT32U" name="sboTimeout" dchg="false" dupd="false" fc="CF" qchg="false"/>
<DA bType="Enum" name="sboClass" type="SIGMA_SboClassKind_2020A" dchg="false" dupd="false" fc="CF" qchg="false"/>
<DA bType="INT32U" name="operTimeout" dchg="false" dupd="false" fc="CF" qchg="false"/>
<DA bType="VisString255" name="d" dchg="false" dupd="false" fc="DC" qchg="false"/>
<DA bType="Unicode255" name="dU" dchg="false" dupd="false" fc="DC" qchg="false"/>
</DOType>
<DOType id="SIGMA_BSC_PosChg_2020A" cdc="BSC">
<DA bType="Struct" name="origin" type="SIGMA_Originator_2020A" dchg="false" dupd="false" fc="ST" qchg="false"/>
<DA bType="Struct" name="Oper" type="SIGMA_Oper_PosChg_BSC_2020A" dchg="false" dupd="false" fc="ST" qchg="false"/>
<DA bType="INT8U" name="ctlNum" dchg="false" dupd="false" fc="ST" qchg="false"/>
<DA bType="Struct" name="valWTr" type="SIGMA_ValWithTrans_2020A" dchg="false" dupd="false" fc="ST" qchg="false"/>
```

```
<DA bType="Quality" name="q" dchg="false" dupd="false" fc="ST" qchg="false"/>
<DA bType="Timestamp" name="t" dchg="false" dupd="false" fc="ST" qchg="false"/>
<DA bType="BOOLEAN" name="stSeld" dchg="false" dupd="false" fc="ST" qchg="false"/>
<DA bType="BOOLEAN" name="opRcvd" dchg="false" dupd="false" fc="OR" qchg="false"/>
<DA bType="BOOLEAN" name="opOk" dchg="false" dupd="false" fc="OR" qchg="false"/>
<DA bType="Timestamp" name="tOpOk" dchg="false" dupd="false" fc="OR" qchg="false"/>
<DA bType="BOOLEAN" name="persistent" dchg="false" dupd="false" fc="CF" qchg="false"/>
<DA bType="Enum" name="ctlModel" type="SIGMA_CtlModelKind_2020A" dchg="false" dupd="false" fc="CF" qchg="false"/>
<DA bType="INT32U" name="sboTimeout" dchg="false" dupd="false" fc="CF" qchg="false"/>
<DA bType="Enum" name="sboClass" type="SIGMA_SboClassKind_2020A" dchg="false" dupd="false" fc="CF" qchg="false"/>
<DA bType="INT8" name="minVal" dchg="false" dupd="false" fc="CF" qchg="false"/>
<DA bType="INT8" name="maxVal" dchg="false" dupd="false" fc="CF" qchg="false"/>
<DA bType="INT32U" name="operTimeout" dchg="false" dupd="false" fc="CF" qchg="false"/>
<DA bType="VisString255" name="d" dchg="false" dupd="false" fc="DC" qchg="false"/>
<DA bType="Unicode255" name="dU" dchg="false" dupd="false" fc="DC" qchg="false"/>
<DA bType="VisString255" name="dataNs" dchg="false" dupd="false" fc="EX" qchg="false"/>
</DOType>
<DOType id="SIGMA_ENS_UntOpMod_2020A" cdc="ENS">
<DA bType="Enum" name="stVal" type="SIGMA_UnitOperationModeKind_2020A" dchg="false" dupd="false" fc="ST" qchg="false"/>
<DA bType="Quality" name="q" dchg="false" dupd="false" fc="ST" qchg="false"/>
<DA bType="Timestamp" name="t" dchg="false" dupd="false" fc="ST" qchg="false"/>
<DA bType="VisString255" name="d" dchg="false" dupd="false" fc="DC" qchg="false"/>
<DA bType="Unicode255" name="dU" dchg="false" dupd="false" fc="DC" qchg="false"/>
<DA bType="VisString255" name="dataNs" dchg="false" dupd="false" fc="EX" qchg="false"/>
</DOType>
<DOType id="SIGMA_ENS_GridOpSt_2020A" cdc="ENS">
<DA bType="Enum" name="stVal" type="SIGMA_GridOperationStatusKind_2020A" dchg="false" dupd="false" fc="ST" qchg="false"/>
<DA bType="Quality" name="q" dchg="false" dupd="false" fc="ST" qchg="false"/>
<DA bType="Timestamp" name="t" dchg="false" dupd="false" fc="ST" qchg="false"/>
<DA bType="VisString255" name="d" dchg="false" dupd="false" fc="DC" qchg="false"/>
<DA bType="Unicode255" name="dU" dchg="false" dupd="false" fc="DC" qchg="false"/>
<DA bType="VisString255" name="dataNs" dchg="false" dupd="false" fc="EX" qchg="false"/>
</DOType>
<DOType id="SIGMA_ENC_ReqSt_2020A" cdc="ENC">
<DA bType="Struct" name="origin" type="SIGMA_Originator_2020A" dchg="false" dupd="false" fc="ST" qchg="false"/>
```

<DA bType="INT8U" name="ctlNum" dchg="false" dupd="false" fc="ST" qchg="false"/>

<DA bType="Enum" name="stVal" type="SIGMA_RequestedStateKind_2020A" dchg="false" dupd="false" fc="ST" qchg="false"/>

<DA bType="Quality" name="q" dchg="false" dupd="false" fc="ST" qchg="false"/>

<DA bType="Timestamp" name="t" dchg="false" dupd="false" fc="ST" qchg="false"/>

<DA bType="BOOLEAN" name="stSeld" dchg="false" dupd="false" fc="ST" qchg="false"/>

<DA bType="BOOLEAN" name="opOk" dchg="false" dupd="false" fc="OR" qchg="false"/>

<DA bType="Timestamp" name="tOpOk" dchg="false" dupd="false" fc="OR" qchg="false"/>

<DA bType="Enum" name="ctlModel" type="SIGMA_CtlModelKind_2020A" dchg="false" dupd="false" fc="CF" qchg="false"/>

<DA bType="INT32U" name="sboTimeout" dchg="false" dupd="false" fc="CF" qchg="false"/>

<DA bType="Enum" name="sboClass" type="SIGMA_SboClassKind_2020A" dchg="false" dupd="false" fc="CF" qchg="false"/>

<DA bType="INT32U" name="operTimeout" dchg="false" dupd="false" fc="CF" qchg="false"/>

<DA bType="VisString255" name="d" dchg="false" dupd="false" fc="DC" qchg="false"/>

<DA bType="Unicode255" name="dU" dchg="false" dupd="false" fc="DC" qchg="false"/>

<DA bType="VisString255" name="dataNs" dchg="false" dupd="false" fc="EX" qchg="false"/>

<DA bType="Struct" name="Oper" type="SIGMA_Oper_ReqSt_2020A" dchg="false" dupd="false" fc="CO" qchg="false"/>

</DOType>

<DOType id="SIGMA_RST_2020A" cdc="RST">

<DA bType="Struct" name="origin" type="SIGMA_Originator_2020A" dchg="false" dupd="false" fc="MX" qchg="false"/>

<DA bType="INT8U" name="ctlNum" dchg="false" dupd="false" fc="MX" qchg="false"/>

<DA bType="Struct" name="mxVal" type="SIGMA_AnalogueValue_f_2020A" dchg="false" dupd="false" fc="MX" qchg="false"/>

<DA bType="Quality" name="q" dchg="false" dupd="false" fc="MX" qchg="false"/>

<DA bType="Timestamp" name="t" dchg="false" dupd="false" fc="MX" qchg="false"/>

<DA bType="VisString255" name="id" dchg="false" dupd="false" fc="DC" qchg="false"/>

<DA bType="Struct" name="sVC" type="SIGMA_ScaledValueConfig_2020A" dchg="false" dupd="false" fc="CF" qchg="false"/>

<DA bType="Struct" name="units" type="SIGMA_Units_2020A" dchg="false" dupd="false" fc="CF" qchg="false"/>

<DA bType="Struct" name="minVal" type="SIGMA_AnalogueValue_f_2020A" dchg="false" dupd="false" fc="CF" qchg="false"/>

<DA bType="Struct" name="maxVal" type="SIGMA_AnalogueValue_f_2020A" dchg="false" dupd="false" fc="CF" qchg="false"/>

<DA bType="Struct" name="stepSize" type="SIGMA_AnalogueValue_f_2020A" dchg="false" dupd="false" fc="CF" qchg="false"/>

<DA bType="VisString255" name="d" dchg="false" dupd="false" fc="DC" qchg="false"/>

<DA bType="Unicode255" name="dU" dchg="false" dupd="false" fc="DC" qchg="false"/>

<DA bType="VisString255" name="cdcNs" dchg="false" dupd="false" fc="EX" qchg="false"/>

<DA bType="VisString255" name="cdcName" dchg="false" dupd="false" fc="EX" qchg="false"/>