

TwinCAT supports wind power industry standards

Standardized communication between wind turbines

Regenerative energies are on the rise. More and more wind turbines are being installed worldwide, both onshore and offshore; most of these are organized into wind parks. The IEC 61400-25 communication standard is intended to make communication possible in such wind parks among wind turbines made by different manufacturers. This is based on the IEC 61850 basic standard and will be available from mid-2009 as a PLC library within one of the TwinCAT supplementary products.



The use of regenerative energies is, of course, highly topical on account of the world's ever dwindling fossil fuel resources. The wind turbine market is booming and competition among suppliers is increasingly fierce. This is why it is not unusual for wind turbines made by different manufacturers to be operated in the same farm by the wind farm operators. Up until now, proprietary protocols have often been used for the necessary communication between the wind turbines and the control station, which was the cause of considerable expense for the required adaptations on both sides. In order to minimize these costs, wind turbine manufacturers and operators sat down together at the table and, during harmonization work that lasted several years, wrote the IEC 61400-25 standard, which governs communication for the supervision and control of wind turbines. On the basis of this standard, turbines and control stations from different manufacturers can now communicate with one another without problem. Particular importance was placed on the simplicity of the communication. However, this does not mean that complex procedures cannot also be managed using this protocol. The object-oriented approach simplifies configuration, diagnostics and maintenance for all concerned.

IEC 61400-25 is based on IEC 61850, a general transmission protocol for protec-

tive and control equipment in medium and high voltage electrical switchgear (station automation). Physically, the communication is based on Ethernet and TCP/IP (Transmission Control Protocol/Internet Protocol); MMS (Manufacturing Message Specification) is used for client-server communication. Published in 2004, IEC 61850 is strictly object-oriented and its data models are hierarchically structured. The objects are self-descriptive, i.e. the structure of the objects is transmitted with the telegram. Besides MMS and TCP communication, the standard also offers fast access to measurement data, which is exchanged directly via Ethernet. An OPC layer can also be used in place of an MMS layer.

In the field of telecontrol engineering, Beckhoff has implemented a great many protocols in its TwinCAT automation suite. The protocols from IEC 60870-5, the predecessor to IEC 61850, are already used in many areas of telecontrol engineering. Alongside classic telecontrol in the field of energy supply, this also includes the automation of tunnels and roads, for example. IEC 60870-5 describes connection-oriented protocols for various areas of application: IEC 60870-5-101 (telecontrol tasks with serial communication), IEC 60870-5-102 (counter state evaluation) and IEC 60870-5-103 (serial communication of digital protective devices). Communication can take place via TCP/IP with the IEC 60870-5-104 pro-

tol (telecontrol tasks in IP networks). All of these protocols have already been in use for many years, but still offer room for manufacturer-specific implementations. This should change with the introduction of IEC 61850.

IEC 61400-25 communication standard

The TCP/IP basic transmission protocol is data-stream-oriented, i.e. no statements about the start, end or length of a message are contained in the telegrams. However, these statements are extremely important for the automation technology since, as in the upper layers of the IEC 61400-25 communication stack, the working method is message-oriented. One possibility of sending and receiving single messages via a TCP data stream is represented by the ‘Request for Comments 1006’ (RFC1006). This protocol extension with the title ‘ISO Transport Service on top of the TCP’, which is integrated in the TwinCAT PLC, is based on the data-stream-oriented TCP protocol and makes message-oriented communication available to applications. In addition to the TCP data, further information is transmitted between the devices here in order to provide certain services to the user (ISO services as an extension to TCP).

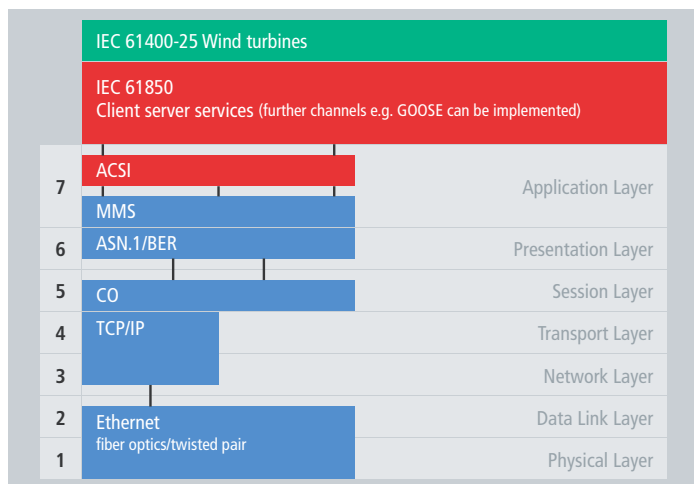


Figure 1: Communication stack according to IEC 61850/IEC 61400-25

The ‘Abstract Syntax Notation One’ (ASN.1), see figure 1, is a description language with which data and information can be described independently of the environment in which they specifically occur. It is used, for example, to describe the data transferred from the ‘session layer’ to the ‘presentation layer.’ The ‘Basic Encoding Rules for ASN.1’ (BER) provide the necessary clear encoding of the data. ASN.1 is the only abstract syntax to have been standardized by the ISO so far. Special TwinCAT PLC blocks assume the encoding and decoding of the BER rules, independent of the IEC 61400-25 protocol.

The ISO standard 9506 ‘Manufacturing Message Specification’ (MMS) defines the object-oriented exchange of data. The protocol concerned is an application protocol (application layer) that, among other things, forms the basis for the

IEC 61850 transmission protocol. MMS provides the subordinate protocols with a large number of services for manufacturer-independent communication. Furthermore, MMS is based on a strict client-server model, in which a client sends orders or requests to a server. The services required for this communication are usually ‘confirmed services.’ The principle of this classic type of communication can be seen in figure 2.

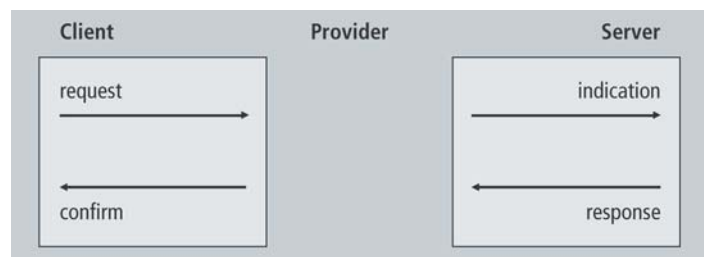


Figure 2: Principle of the classic server/client communication

The services defined in IEC 61850 for the exchange of data are mapped on MMS services so that access can be gained to the communication stack. In order for the standard to be independent of the MMS protocol, the mapping of the services takes place via an abstract service interface. This ACSI interface (Abstract Communication Service Interface) allows universal connection of the standard to the subordinate protocols. In this way, connection via Web services (OPC) could be possible in future without specially adapting the IEC 61850 standard for this purpose.

The IEC 61400-25 data model

IEC 61850 forms the basis for IEC 61400-25; for this reason, the data or object models are identical for both standards. The data contained in the basic standard is used to a large extent by IEC 61400-25 and is supplemented by several pieces of information specific to wind power. This also applies to further specializations of the basic standard. For example, standards for hydroelectric power plants and distributed energy resources are currently being defined.

Since the data exchange is standardized rather than the application functions in this standard, it is referred to as a data or object model. This is subdivided into five hierarchical levels:

- | Server
- | Logical Device
- | Logical Node
- | Data Object
- | Data Attribute

The server offers the connection point of a device to the communication system, thus forming the uppermost hierarchical level. Since this logical connection point can be implemented at several points, it permits a redundant structure.

The second hierarchical level subdivides a single physical device into several sep-



'Alpha Ventus' pilot project: offshore application at sea

The TwinCAT PLC libraries for IEC 61400-25 communication will be used for the first time in a German offshore wind farm on the North Sea. The test phase with the responsible control system manufacturers went very well and speaks in favor of the development strategy pursued by Beckhoff.

The Alpha Ventus offshore wind farm, around 45 kilometers (28 miles) north of the island of Borkum, is the first German wind farm at sea. The design, construction, operation and grid integra-

tion of the Alpha Ventus research project will provide invaluable insights for the future commercial utilization of offshore wind farms. For 2008 the project schedule includes construction of the southern half of the wind farm with six Multibrud M5000 turbines and an offshore substation. The M5000 is a 5 MW offshore turbine. Multibrud has overhauled the control concept in close cooperation with Beckhoff and has based it on a uniform platform.

- Wind special PC Control 02|2008 www.pc-control.net
- Multibrud www.multibrud.com
- Alpha Ventus project www.alpha-ventus.de

ate parts, so-called *logical devices*. The advantage of this subdivision is that functions or objects associated with one another stand together on account of their commonness and can be commonly switched to a different operating mode, for example.

The third hierarchical level is formed by the *logical nodes*. These represent the information from all conceivable sub-functions that could exist in the digital station control technology in wind turbines, such as all safety functions, automation functions and functions for the processing of measured values and counter readings, among others. Defined identifiers in the form of an abbreviation with four letters exist for all standardized *logical nodes* in IEC 61850 and IEC 61400-25. WTUR, for example, is the abbreviation for the logical node 'wind turbine general information.' A *logical device* may consist of several such nodes.

A *logical node* consists in turn of several *data objects*; these objects form the fourth hierarchical level. A special feature of this hierarchical level is that data objects can, in turn, be hierarchically nested, i.e. a data object may consist of several subordinated data objects before a data type can ultimately be accessed. This then forms the lowest hierarchical level. These so-called *data attributes* represent detailed information or the values of the data objects. Since the same number of data attributes is always defined for many data objects, the attributes of an object are summarized in *common data classes* (CDC). In most cases, the data attributes are elementary data types such as *Boolean* or *integer*, but structures and arrays of elementary data types are also possible.

In order to read or write a certain value from or to a server, a reference (path) to this variable is required. The different hierarchical levels down to the basic type are thereby appended to one another, for example as in the status query WEA/WTRF\$TrfOpTmRs\$manRs\$stVal, which a control station sends to a wind turbine. In this case, the object status of the node 'wind turbine transformer information' is queried or read.

Several such references can be summarized by means of so-called *data sets* and transmitted together by means of reporting functions, which are defined in the standard as 'saving' or 'non-saving.' In this way, the individual calling of the references for data objects that are mostly communicated in context to one another can be dispensed with.

Realization in TwinCAT PLC

The communication stack has been implemented in the TwinCAT PLC in the form of libraries. In doing so care was taken that, if possible, every layer of the 'Open

System Interconnection Reference Model' (OSI model) is implemented independently, i.e. in its own library. The decisive libraries for the user are TcIEC61850.lib for the basic standard and TcIEC61400_25.lib for specialization in the field of wind turbines. TcIEC61850.lib contains all of the basic data types defined in the standard. On top of that, the entire data model described in the standard can be found in this library; 91 predefined *logical nodes* and 29 *common data classes* are available to the user for the compilation of their *logical devices*. All of the services required for reading the created data structure are similarly contained in this library. All nodes and data classes specific to wind turbines are implemented in TcIEC61400_25.lib and can be used in conjunction with TcIEC61850.lib for communication in wind farms.

This division into individual libraries provides the advantage of ensuring the clearness and modularity of the products. After all, we are dealing with two different standards, each of which can be considered to be a separate implementation layer. Therefore, the consistent and independent development of libraries, as applied to the OSI model, will be continued.

The simple configuration of logical devices is clearly shown by the following example: a *logical device* with the name MyWEA (instance of ST_LD_WEA) is created as a structure in the TwinCAT PLC. The mandatory nodes LLNO (*logical node zero*) and LPHD (*logical node physical device*) must be integrated in each logical device, as is also described in the two standard parts described here.

```

0001 TYPE ST_LD_WEA :
0002 STRUCT
0003   info      : ST_ACSI_DATAINFO := ( eType := eACSI_TYPE_STRUCTURE );
0004
0005   LLNO      : ST_IN_LLNO;      (* Logical node zero *)
0006   LPHD      : ST_IN_LPHD;     (* Physical device information *)
0007
0008   WGEN      : ST_IN_WGEN;     (* Wind turbine generator information *)
0009   WROT      : ST_IN_WROT;     (* Wind turbine rotor information *)
0010   WTOW      : ST_IN_WTOW;     (* Wind turbine tower information *)
0011   WTRF      : ST_IN_WTRF;     (* Wind turbine transformer information *)
0012   WTUR      : ST_IN_WTUR;     (* Wind turbine general information *)
0013 END_STRUCT
0014 END_TYPE
    
```

Figure 3: Instances of several logical nodes in a logical device ST_LD_WEA

On top of that, all of the nodes specific to this device can be implemented. In our example, these are the *logical nodes* WGEN (wind turbine rotor information), WROT (wind turbine rotor information), WTOW (wind turbine tower information), WTRF (wind turbine transformer information) and WTUR (wind turbine general information). Since all of the logical nodes are already available in the respective libraries, they need only be instantiated by the user. The data objects of the nodes are located on the next lower hierarchical level. The standard thereby distin-

guishes whether it is mandatory, conditional or optional for a data object to be contained in a *logical node*. If a customer wishes to communicate the default data objects, it is sufficient to create the device MyWEA. If optional data objects are not to be communicated, it is possible to comment these out. This procedure is clearly shown in figure 4 using the *logical node* WROT as an example:

```

0001 TYPE ST_LN_WROT : (* LN: Wind turbine rotor information *)
0002 STRUCT
0003 info : ST_ACSI_DATAINFO := (eType := eACSI_TYPE_STRUCTURE);
0004
0005 (* UCLN Class Mandatory: *)
0006 NamePlt : ST_CDC_LPL; (* Name plate *)
0007
0008 (* Status information: *)
0009 RotSt : ST_CDC_STV; (* Status of rotor *)
0010 BlStB11 : ST_CDC_STV; (* Status of blade 1 (reference) *)
0011 BlStB12 : ST_CDC_STV; (* Status of blade 2 *)
0012 BlStB13 : ST_CDC_STV; (* Status of blade 3 *)
0013 PtCtlSt : ST_CDC_STV; (* Status of pitch control *)
0014
0015 (* Analogue information: *)
0016 RotSpd : ST_CDC_MV; (* Value of rotor speed at rotor side *)
0017 RotPos : ST_CDC_MV; (* Angular rotor position *)
0018 HubTap : ST_CDC_MV; (* Temperature in the rotor hub *)
0019
0020 (* PtHyPresB11 : ST_CDC_MV; *) (* Pressure of hydraulic pitch system for blade 1 *)
0021 (* PtHyPresB12 : ST_CDC_MV; *) (* Pressure of hydraulic pitch system for blade 2 *)
0022 (* PtHyPresB13 : ST_CDC_MV; *) (* Pressure of hydraulic pitch system for blade 3 *)
0023
0024 PtAngSpB11 : ST_CDC_MV; (* Pitch angle set points for blade 1 (reference) *)
0025 PtAngSpB12 : ST_CDC_MV; (* Pitch angle set points for blade 2 *)
0026 PtAngSpB13 : ST_CDC_MV; (* Pitch angle set points for blade 3 *)
0027 PtAngValB11 : ST_CDC_MV; (* Pitch angle for blade 1 (reference) *)
0028 PtAngValB12 : ST_CDC_MV; (* Pitch angle for blade 2 *)
0029 PtAngValB13 : ST_CDC_MV; (* Pitch angle for blade 3 *)
0030
0031 (* Control information: *)
0032 BlkRot : ST_CDC_CMD; (* Set rotor to blocked position *)
0033 PtEngChk : ST_CDC_CMD; (* Check emergency pitch system *)
0034 END_STRUCT
0035 END_TYPE
    
```

Figure 4: Logical Node WROT with 'mandatory' and 'optional' Data Objects

In this case, the optional data objects for the hydraulic pressure of the pitch system are not communicated. Further development work will be aimed at making "commenting out" easier for the user.

All in all, it becomes clear that the high development expenditure is reflected by a low application expenditure for the customer. Once the user has configured all of their devices, the user can begin communication with any client that supports the IEC 61850 and IEC 61400-25 standards. In the initialization step, the created

data structure of the server is loaded first. After that, it is possible to execute a password query during the establishment of a connection with the Beckhoff server. In this way, connection requests from unauthorized clients can be denied. To do this, only the password query (bAccessControl) needs to be switched on and the authorization mechanism set up, as can be seen in figure 5. Several passwords for various users can be specified using the accessCtrl array.

```

fbServer
├── mmsPara
│   ├── maxPDUsize = 16#0000FDE8
│   ├── localDetail = 16#0000FDE8
│   ├── maxServOutCalling = 16#000A
│   └── maxServOutCalled = 16#000A
├── services
├── parameters
│   ├── nestingLevel = 16#05
│   └── bAccessCtrl = TRUE
├── accessCtrl
├── accessCtrl[1]
│   ├── bAuthentication = TRUE
│   ├── authenticMechanism = '2.2.3.1'
│   └── authenticPassword = 'PASSWORD'
├── services
├── parameters
│   └── nestingLevel = 16#05
├── accessCtrl[2]
├── accessCtrl[3]
└── accessCtrl[4]
    
```

Figure 5: Authentication of the clients in the TwinCAT server

If the server has accepted the client's connection request, the latter can read out the data structure of the server. The exchange of the data, which is encoded with the BER Encoding Rules, takes place via TCP/IP. For this type of communication,

the TwinCAT-TCP/IP connection server is used, which enables one or more TCP/IP server clients to be implemented in the TwinCAT PLC. This tried and tested supplementary product from Beckhoff allows a server to be implemented on the basis of Windows XP and Windows CE systems. The IEC 61850 communication services responsible for data exchange are, as described, mapped on MMS services. The most important of these are the reading and writing services.

In order to use the reporting function mentioned in the 'Data model' section, the user must create *data sets*, which allow access to an arbitrary number of data objects using a single name. This name, or rather the reference to the data set itself, is merely a single reference to several specific references of the data objects. In order to create such a data set, the number of objects and their references must be transferred to the function block (see figure 6). The data set is given a unique name, which is transferred to a report block that, on account of occurring events, triggers the transmission of the data set. Events are taken to mean the evaluation of the so-called trigger options.

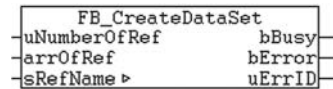


Figure 6: Function block for the creation of a data set

The report block can thereby be informed of the type of event to which it should react: if it reacts to a *data change*, it communicates the data as soon as a value within this set changes.

Practical experiences and outlook

During the initial server development phase, demo clients from various suppliers offered an initial useful test platform. The development team deliberately developed no test client of its own so as to avoid any vendor-specific solutions. In this way, it was possible from the first development step onwards to test and ensure the server's conformity to the standard.

The next development steps will focus on the further improvement of user-friendliness. Among other items, it is intended to develop tools to replace the commenting out of optional data objects in common data classes. Similarly, other protocols besides MMS, such as Web services or OPC, should be integrated in order to offer the customer more flexibility. A completely different aspect is the development of the GOOSE channel used by this standard. GOOSE stands for 'Generic Object Oriented Substation Events' and is responsible for the exchange of high-priority real-time data. Real-time communication can be used for protective triggers and blocks, and is based directly on Ethernet technology.