

Dear USE 61400-25 Members,

Our implementation guideline workshop on 18th and 19th of February 2014 in Hamburg will focus on modelling. In addition I would like to organize training for software developers who want to implement a client or server for IEC61400-25. The training could start at the earliest after April 2014.

I am in contact with Michael Zillgith, who would like to train you on his open source library for IEC 61850 (<http://libiec61850.com/libiec61850/about/>). Michael Zillgith studied automation engineering at Offenburg. Since 2006 he works as a scientific associate at the Fraunhofer Institut für Solare Energiesysteme ISE (Freiburg).

He developed the C-library in the context with smart grid and "Decentralized Energy Resources (DER)".

As the libIEC61850 provides a server and client library for the IEC 61850/MMS, it's easy to setup a communication with IEC 61400-25/MMS.

The training is planned as a three day crash course for advanced C++ programmers and should focus on developing a client application or a server application, depending on your needs.

Below you will find the first draft of a training plan by Michael Zillgith

Please let me know if you would like to attend the training and what your goal would be.

I am looking forward to hearing from you

Hennig Harden
USE61400-25 user group
Technical Team Chairman

<http://www.use61400-25.com>

Workshop: IEC 61850 implementation

Goals

Participants should become...

- familiar with the technical details of the IEC 61850 communication protocols
- familiar with the IEC 61850 client and server API of libIEC61850
- able to create a client, to connect to a wind power plant, and to perform typical tasks (reading data, configuration, plant control, ...)
- able to implement a simple IEC 61400-25 compliant server
- able to analyze communication problems

Prerequisites for participants

- Laptop with access to an administrator account
- Windows 7/8 or Linux (Ubuntu) OS
- preferred development tool chain (Visual Studio 2010/2012, GCC/Eclipse CDT) installed
- Wireshark installed

Outline

First day (morning):

Block 1: Theory

- Introduction
- The technical details of IEC 61850 communication protocols
 - MMS
 - technical basics (ASN.1, ISO protocol layers)
 - services
 - model discovery (browsing)
 - read and write MMS variables
 - data set services
 - event based data transmission (reports)
 - journals (logs)
 - Mapping of IEC 61400-25 data model elements to MMS variables
 - Example: Analysis of an MMS association with the Wireshark protocol analyzer

Block 2: Warming up with libIEC61850

- Setting up the development environment
 - Download the source code
 - Installing the required tools (cmake, eclipse, ...)
- Building the examples
- Starting an example server and client
- Exercise: Analyze client/server communication with wireshark:
 - Short introduction to Wireshark
 - Capture and analyze different communication problems

First day (afternoon)

Block 3: Introduction to libIEC61850

- Protocol stack architecture
- Overview client API
- Overview server API

Block 4: Introduction to the client API (consisting of explanations and practical exercises with the example code)

- Data handling, Memory management, Error handling
- Connect to a server
- Read variables
- Write variables
- Read a data set
- Create and delete data sets
- Configuration and processing of reports
- Using control services

Second day (morning):

Block 5: Client side exercises:

- Participants develop their own client code to communicate with a demo server (e.g. REpower virtual power plant, provided example servers)
- read variables from the server
- writing configuration variables
- Configuration and activation of reports for event based transmissions
- Controlling a device

Second day (afternoon):

Block 6: Introduction to the server stack and API

- Server implementation workflow
 - Creating the data model and choosing the required services (→ XML/ICD file)
 - Hardware requirements: choosing an adequate hardware platform
 - dealing with resource constraints
 - Convert XML/IED model to C source code
- Handling client requests (observing variables)
- Update the server data model with process values
- Server side control model support