Automatic Generation of PLC Projects Using Standardized Components and Data Models

Sylvia Huynh
PLC Developer and Support Engineer

ICALEPCS 2019
New York, 10.10.2019
European XFEL at a glance

linear accelerator
for electrons (10.5, 14.0, 17.5 GeV)

SASE 2
0.05 nm - 0.4 nm

SASE 1
0.05 nm - 0.4 nm

SASE 3
0.4 nm - 4.7 nm

electron tunnel
electron switch
electron bend
electron dump

photon tunnel
undulator

MID Materials Imaging and Dynamics
HED High Energy Density Science
Optional space for two undulators and four instruments
SPB Single Particles, Clusters, and Biomolecules and Serial Femtosecond Crystallography
SFX Femtosecond X-ray Experiments
FXE Small Quantum Systems
SCS Spectroscopy & Coherent Scattering

Image credit to: https://www.xfel.eu
Background

- Requirement to develop, deploy and commission several PLCs in a short period of time
- Manual creation of PLC projects are highly repetitive, error prone and time consuming
- Solution: Development of the PLC Management System (PLCMS)
The PLCMS

- Predominately built using Python 3
- Open source tools and Object-Relational Mapping (ORM)
- Interfaces into a relational database
- Utilizes the eXtended Automation Interface (XAE) provided by Beckhoff
- Provides functionality to expedite PLC Project generation

Divided into two main facets:
- PLC Development
- PLC Generation
PLCMS Development Process

- Manually developed within a PLC framework Project in Structured Text (ST)
- Adheres to a structural interface template

- Revision control of the source code
- Compiles the collection of softdevices into a library

- Extracts interface related information of the softdevice
- Data integrity and structural checks

- Stores interfacing aspects of the softdevice as a set of relationships
- Query the information for changelogs, comparisons across versions…
PLC Framework and Database Model

TwinCAT PLC Framework

Provides the set of available functionality for a collection of softdevices

Compiled as a .library file to be used by PLC projects

Database Model

Breaks down the framework into datasets that define the softdevice interface.

The PLC Framework data is captured for every version, and is available to be queried and used for an array of tasks.
The PLCMS Project Generation Process Overview

- PLCMS DB
  - Reads in data from all sources
  - Data integrity checks and generates the PLC Project File

- PLC Project File
  - XML File, with a selected framework version
  - Configuration and Initialization values

- Project Builder
  - Beckhoff XAE – C#
  - Creates a TwinCAT PLC Project

- Ready for deployment
The PLC allocation is designed around beamline infrastructure, broken up by sub-system. This wiring of all the hardware is defined within EPLAN, which is then exported into an XML file.

The EPLAN export contains:

- Hierarchy of the fieldbus terminals with EtherCAT Addressing
- Instantiated softdevice names and signals used
- Associated fieldbus terminal for the softdevice signal
Interlocks are defined within an Excel spreadsheet as a set of Boolean conditions which trigger a series of protective actions.

They are commonly used for equipment protection in addition to rules relating to preservation of photon beam properties and vacuum system integrity.

Project Data File contains initialization and configurational parameters, in addition to the desired framework library version as an XML file. Other information pertaining to the current PLC such as peer-to-peer communications which isn’t covered by the previously mentioned files are also included.
PLC Project Generation

The PLCMS parses all of the aforementioned files to build the I/O fieldbus tree. Provided with the PLC Framework library version obtained from the Project Data file, the PLCMS:

- Checks for completeness within the softdevice
- Ensures correct EtherCAT Fieldbus addressing – no double ups.
- Replaces unsupported softdevices with a set of simplified components (e.g. Digital Out)
- Raise warnings/errors
- Generates a PLC Project XML file
PLC Project Builder

- Interfaces with the Beckhoff eXtended Automation Engineering interface (XAE)
- Developed in C#
- Consumes the XML file (generated by the PLCMS) to:
  - Build TwinCAT PLC Project
  - Generate the list of hardware terminals
  - Linking of hardware terminals to softdevices
  - Compiles the PLC project – ready for deployment.
PLC Deploy

- Generated PLC Project is placed on SVN
- Sanity checks are applied
- Code is manually downloaded from the control network onto the production PLCs
Future and Ongoing Developments

- Database Independent
- Configuration and Initialization Settings Integration
- Web Interface
- SCADA (Karabo) Integration

Image credit to: https://www.xfel.eu
Sylvia Huynh
PLC Developer and Support Engineer
European XFEL
sylvia.huynh@xfel.eu

Questions?