

 **OPC UA**™ **Device Support**

Connecting Controllers

Ralph Lange, ITER Organization

Disclaimer: The views and opinions expressed herein do not necessarily reflect those of the ITER Organization

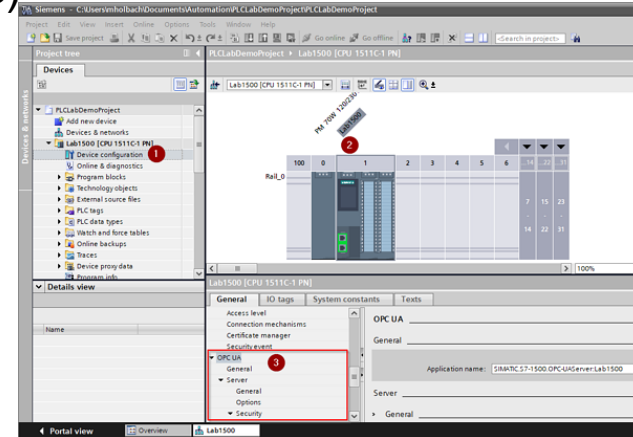
Requirements / Wish List

- Symbolic addressing
 - names, not numbers
- No PLC programming
 - keep contractors happy
- Industrial standard
 - one to connect them all
- Portability
 - at least Linux, Windows
- User-defined structures
 - for reusable PLC objects
- Subscription mechanism
 - update on change
- Server-side queues
 - handle bursts well
- Browsing support
 - find variables easily
- Security (X.509)
 - encrypt, sign, authenticate
- Remote procedure calls
 - with parameters and results



OPC UA

- Industrial standard (2006) to interface SCADA to PLCs
 - Covers data, alarms, events, historical data, remote methods
- Based on OPC Classic (Microsoft; 1996), plus
 - Portability → UA does *not* require DCOM/Windows like Classic
 - Safety/security (authentication, encryption)
 - Information modeling (user defined structures)
 - Remote method execution
- Gaining momentum in industrial context as universal integration standard
- Siemens S7-1500 series PLCs include an embedded OPC UA server



Limitations

- Mostly found on the server end
 - S7-1500: depends on the model (S/M/L) and data organization
 - How many, how large, how often?
 - Client-side fine tuning available
 - e.g., “registering” items to cache server-side name resolution*
 - Servers separate from PLCs introduce additional latency
- Found one limit in the client
 - LabView serving 600 arrays of 7500 doubles each (~36MB data)
 - > SDK client has a limit of ~16MB for the serializer
 - workaround: 3 subscriptions of 200 arrays each

EPICS Device Support

- Based on commercial C++ Client SDK by Unified Automation
 - 4k€ for source code and 1 year support (extend support: 20% per year)
one developer/many products or many developers/one product
 - Binaries can be deployed/distributed royalty-free
 - Platforms: Windows and Linux
 - Evaluation bundles available
- Evaluation of client library options and Device Support prototype
Bernhard Kuner (HZB/BESSY)
- ITER use cases tested by F4E (Spain) and TCS (India)
 - Against S7-1516/1518 embedded OPC UA server
 - Against WinCC-OA embedded OPC UA server

Status

- Requirements Specification v1.1: <https://bit.ly/opcua-srs-11>
- Design done (still no formal doc)
- Implementation nearly complete
 - All basic data types and arrays thereof (*read/write/subscribe*)
 - Supporting all applicable record types (bidirectional outputs)
 - User-defined structures (*read/write/subscribe*), timestamps from data
 - Server-side queues, configurable connection behavior
 - OPC UA Security (encrypt, sign, authenticate)
Lots of testing help by Roland Fleischhauer (HZB/BESSY)
 - Integrated end-to-end test against a software server
Work by Ross Elliot and Karl Vestin (ESS)

Users (as of 2022)

Facility	OPC UA Server	Status
ASIPP	LabVIEW	production
	PLC Siemens S7-1500	production
Australian Synchrotron	PLC Siemens S7-1500F	near production
BESSY II @HZB	PLC Siemens S7-1500	production
	Phoenix Contact	production
CHIMERA @CCFE	Softing uaGate	production
	PLC Siemens S7-1500	development
ESS	LabVIEW	development
	PLC Siemens S7-1500F	production
Fermilab	ABB Power SCADA	near production
	Siemens DESIGO	development
	Kepware KEPServerEX	testing
IPR	PLC Siemens S7-400	development
	PLC Siemens S7-1500	testing
ITER	PLC Siemens S7-1500	production
	Siemens WinCC OA	production
	PCVue	production
KATRIN @KIT	LabVIEW	prototyping
PSI	PLC Siemens S7-1500	development
Varian ProBeam	PLC Siemens S7	production
	PLC Beckhoff	production

Roadmap

- Currently working on:
 - Integration of (free) open62541 client library (will be available in next release)
Work by Dirk Zimoch (PSI) and Carsten Winkler (HZB/BESSY)
 - User Manual (based on ITER's user manual)
- Soon:
 - Support for OPC UA methods (remote execution of PLC code)
- Under EPICS license
- Upstream repository and statically linked binaries containing the UA client (shared libraries to link against your IOCs):
<https://github.com/epics-modules/opcu>