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

Motivation

Existing EPICS interfaces to Siemens S7 PLCs:

- s7plc: TCP Data Block Transfer
  - needs programming on the PLC
  - complete data blocks only
- s7nodave: Snap7 based (S7 protocol)
  - no need for PLC side programming
  - access to the whole PLC memory
- Both work fine, but have disadvantages
  - low level configuration (byte offsets into data blocks)
  - address mismatch may have drastic consequences!
  - pretty specific/proprietary solutions

What do we want?

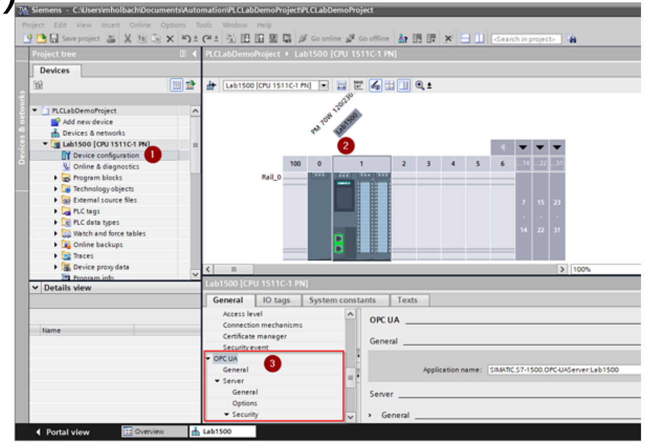
- Symbolic addressing
  - names, not numbers
- No PLC programming
  - keep contractors happy
- Industrial standard
  - one to connect them all
- Portability
  - at least Linux, Windows
- User-defined structures
  - for reusable PLC objects

- Subscription mechanism
  - update on change
- Server-side queues
  - handle bursts well
- Browsing support
  - find variables easily
- Security (TLS, X.509)
  - encrypt, sign, authenticate
- Remote procedure calls
  - with parameters and results

OPC UA™

OPC UA

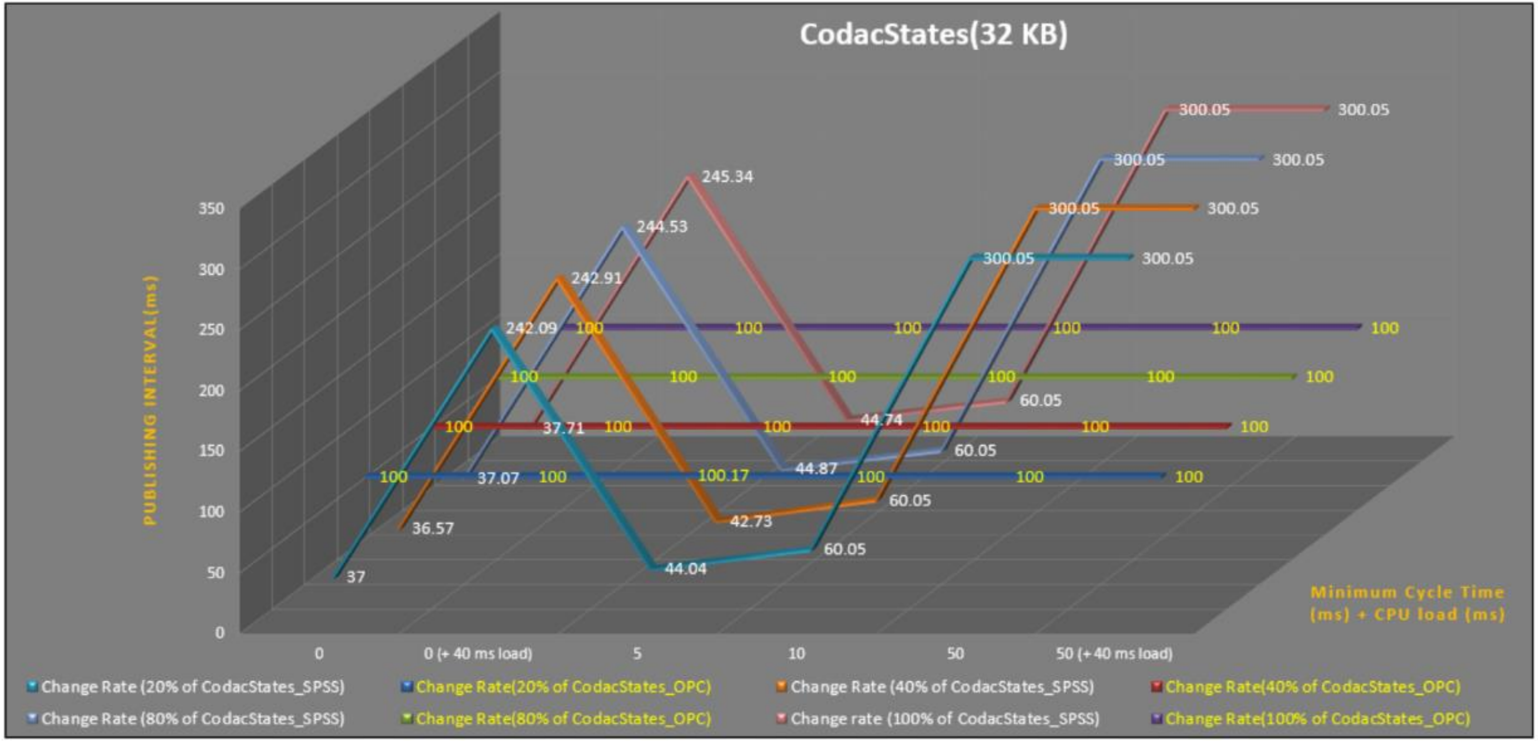
- Industrial standard (2006) to interface SCADA to PLCs
  - Covers data, alarms, events, historical data, remote methods
- Based on OPC Classic (Microsoft; 1996), plus
  - Portability → UA does *not* require DCOM/Windows like Classic
  - Safety/security (authentication, encryption)
  - Information modeling (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



Application name: SIMATIC.S7-1500.OPC-UAServer:Lab1500

Performance

- It all depends – mostly on the server
  early tests by TCS: S7-1518, 32kB data (integers), “empty” / 40ms cycle artificial load



Performance

- Recent test for ITER building automation integration
  - Trying to verify realistic conditions
  - 3x the estimated size (500kB data, mixed types)
  - On a mid-size S7-1516
  - 250ms sampling/publishing period
  - No performance fine-tuning

- Results:
  - Fit for this purpose
  - Best performance when using few large structures

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 can “register” items to do server-side name resolution only once
  - Servers separate from PLCs introduce additional latency

- Recently found a 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, “usable pre-release” state
  - 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)
- Used in production at ITER, BESSY II, ASIPP
- Under evaluation / used in developments at ESS, Fermilab, PSI, AS, LNLS, Varian Medical, UKAEA

Roadmap

- Soon:
  - Support for OPC UA methods (remote execution of PLC code)
  - Integration of (free) open62541 client library
    Dirk Zimoch (PSI) and Carsten Winkler (HZB/BESSY)
  - Integrated end-to-end test against a software server
    Ross Elliot and Karl Vestin (ESS)

- Under EPICS license
- Upstream repository and statically linked binaries on GitHub:
  https://github.com/ralphlange/opcua