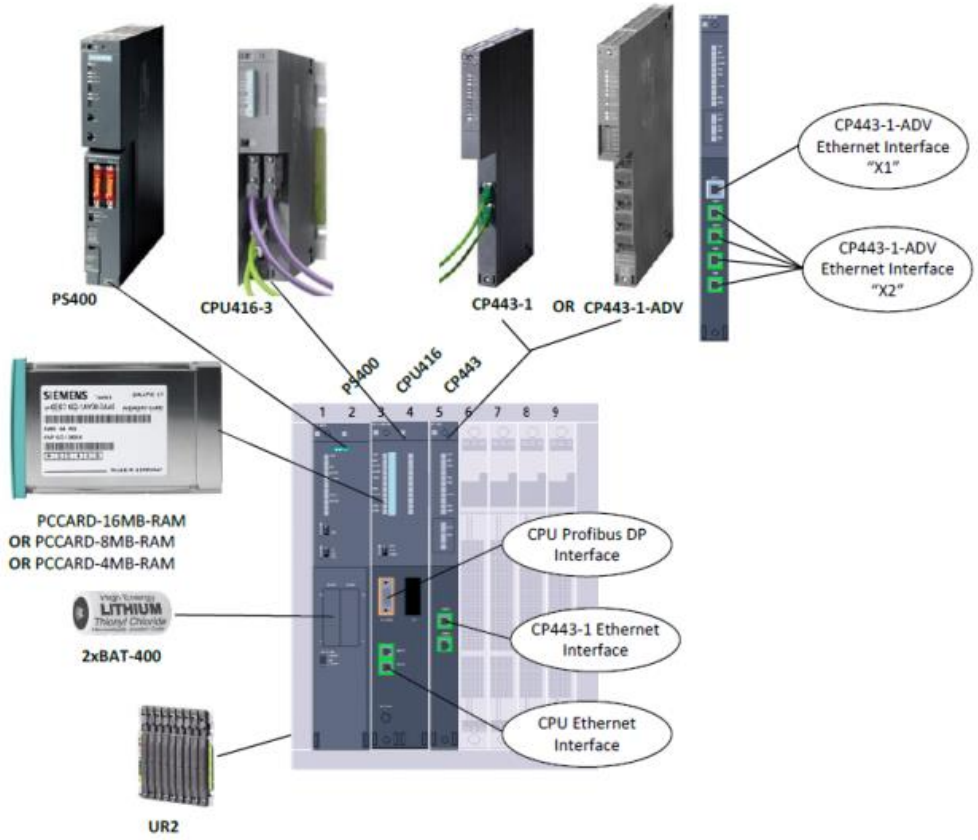
New ASYN Driver for S7 PLCs

Ralph Lange
ITER Organization
Control System Division

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

ITER's Slow Controller: S7

- Hardware catalogue defines S7 for Slow Controller
- CODAC Core System uses the PSI S7PLC support
- SDD creates configuration for PLC code and EPICS DB
- Connecting 1000s of PLCs



ITER Extensions to S7PLC Driver

- Redundant PLCs: driver talks to two PLCs
- Timestamps from PLC
- CODAC Frame:
  timestamp, redundancy status, magic numbers, application version

→ PSI code was heavily patched

Requirement: New Protocols

- Native S7 Protocol (using nodave or snap7)
  - Allows writing or reading single variables
  - Full access to PLC data
  - Does not require changing PLC side code
  - Relatively slow
- ISO-on-TCP
- …?

Requirement: Events

- Capture all changes in PLC retaining order (with PLC cycle resolution)
- Packed single bits (BOOLEAN) and a timestamp
- Both PLC and IOC buffer updates
- PLC sends on change

ASYN Allows Modular Design

- Raw TCP data block protocol: port driver
- S7 protocol: port driver
- Redundancy support: port driver
- ISO-on-TCP: interpose layer
- CODAC frame: interpose layer

- Above: generic ASYN device support
- Beneath: drvAsynIPPortDriver

Status

- Original PSI driver functionality is working
- Next: Redundancy, S7 protocol
- Then: Events, ISO-on-TCP

- Will be made available as pure EPICS module (no CODAC dependency) in Q1/2015

ASYN Driver for Files

- Initial use case:
  Read numeric data (Linux daemon stats) from file into EPICS
- Idea:
  Create port driver that reads from file,
  StreamDevice does the formatted read
- More important use case:
  Run unit tests of ASYN drivers without any hardware,
  by replacing the device input stream with a file port

Status

- Initial use case functionality (reading numbers from file) working
- Will be made available as pure ASYN driver (no CODAC dependency) in Q1/2015