

ADS demystified

Torsten Bögershausen

www.europeanspallationsource.se

- Automation Device Specification
- “Network protocol”
- Runs on PLCs, Bus coupler EK9000
- Spec:
https://infosys.beckhoff.com/english.php?content=../content/1033/ams_nat/4275563275.html&id

- Identification/Addressing of nodes inside the network: AMSnetID
- Use AMSnetID (6 bytes) as identifier
 - Unique (inside your site)
 - Default from Beckhoff, like 5.79.68.216.1.1 or 5.5.65.73.226.1.1
 - Or change to be follow IP.1.1, like 192.168.88.57.1.1

- ADS Router is a daemon
- ADS Routers talk to each other via one (!) TCP/IP connection
- Mapping IP \leftrightarrow AMSnetId needed:
Routes must be configured
(Typical pitfall)

- What is a client:
 - 2 PLC projects on the same CPU
 - PLC project on different CPU
(similar to "CP" in an EPICS record)
 - Development IDE: TwinCAT.exe
 - The TwinCAT OPC UA server (!)
 - an EPICS IOC
 - pyads, test scripts, python
 - py-ads-monitor (like camonitor)

- ADS under Windows from the beginning
- A windows system with TwinCAT has an ADS router started as a system service (daemon)
- Windows DLL: VC++, Visual Basic, C#, Delphi, ADS.NET, etc

ADS via Linux, MacOS, BSD

- Linux, BSD, MacOS, since 2015:
- github.com/Beckhoff/ADS
- Written in C++
(Why ?ADS is not object-oriented)

ADS via Linux, MacOS, BSD - 2

- ESS: 2017..2020
<https://github.com/epics-modules/twincat-ads>
- SLAC uses a fork:
<https://github.com/pcdshub/twincat-ads>
- Some improvements (polling and others)
Merged into ESS at Nov 2023; v2.1.0
- Used at CEA (and more places)

ADS via Linux, MacOS, BSD - 3

- Cosylab: New driver in 2022: <https://indico.cern.ch/event/1173788/contributions/5008269/>
- Highly optimized for reading/writing many, many variables fast

ADS applications github/gitlab

- <https://github.com/Beckhoff/ADS>
- <https://github.com/epics-modules/twincat-ads>
-
- <https://github.com/EuropeanSpallationSource/m-epics-ethercatmc/blob/master/ethercatmcApp/src/ethercatmcADS.cpp>
-
- <https://github.com/stlehmann/pyads.git>
- <https://gitlab.esss.lu.se/mcag/sources/py-ads-monitor.git>
- <https://gitlab.esss.lu.se/mcag/sources/motion-tests.git>
-

ADS via network

- What can be done via ADS ?
- Everything.
 - Get state of system
 - Restart, download SW
- Peek & Poke. (=Read, Write)
 - Get symbol information ("ReadWrite")
 - Open a handle to a symbol
 - Subscribe to value changes (notifications)

ADS lookup Symbol

- > Transmission Control Protocol, Src Port: 58259, Dst Port: 48898, Seq: 39
- ▼ AMS
 - AMS Target Net Id: 5.81.143.110.1.1
 - AMS Target port: 852
 - AMS Sender Net Id: 192.168.88.77.1.1
 - AMS Sender port: 30000
 - CmdId: ADS Read Write (9)
 - > StateFlags: 0x0004
 - cbData: 50
 - ErrorCode: NO ERROR (0x00000000)
 - InvokeId: 0x80000295
 - ▼ ADS ReadWrite Request
 - IndexGroup: 0x0000f009
 - IndexOffset: 0x00000000
 - CBReadLength: 824
 - CBWriteLength: 34

Data	
0000	02 00 00 00 45 00 00 8c 00 00 40 00 40 06 1f e1 E @ . @ . . .
0010	ac 12 9c 18 ac 1e 26 42 e3 93 bf 02 d2 3a 93 c5 & B : . .
0020	77 4b 55 0d 80 18 08 0c 96 ca 00 00 01 01 08 0a w K U
0030	36 44 e6 1f 9d 01 2a 86 00 00 52 00 00 00 05 51 6 D * . . . R Q
0040	8f 6e 01 01 54 03 c0 a8 58 4d 01 01 30 75 09 00 . n . . T . . . X M . . 0 u . .
0050	04 00 32 00 00 00 00 00 00 00 95 02 00 80 09 f0 . . 2
0060	00 00 00 00 00 00 38 03 00 00 22 00 00 00 47 56 8 . . . " . . . G V
0070	4c 2e 61 73 74 41 78 65 73 5b 31 5d 2e 73 74 43 L . a s t A x e s [1] . s t C
0080	6f 6e 74 72 6f 6c 2e 66 56 65 6c 6f 63 69 74 79 o n t r o l . f V e l o c i t y

ADS lookup Symbol Resonse

```

CmdId: ADS Read Write (9)
> StateFlags: 0x0005
  cbData: 122
  ErrorCode: NO ERROR (0x00000000)
  InvokeId: 0x80000295
  ADS ReadWrite Response
    Result: NO ERROR (0x00000000)
    CbLength: 114
  
```

Data

0000	02 00 00 00 45 00 00 d4	00 00 40 00 3d 06 22 99E.... @.=."
0010	ac 1e 26 42 ac 12 9c 18	bf 02 e3 93 77 4b 55 0d	..&B.... ..wKU.
0020	d2 3a 94 1d 80 18 04 00	23 78 00 00 01 01 08 0a	.:..... #x.....
0030	9d 01 2a c0 36 44 e6 1f	00 00 9a 00 00 00 c0 a8	..*.6D.. ..
0040	58 4d 01 01 30 75 05 51	8f 6e 01 01 54 03 09 00	XM..0u.Q .n..T...
0050	05 00 7a 00 00 00 00 00	00 00 95 02 00 80 00 00	..z..... ..
0060	00 00 72 00 00 00 72 00	00 00 40 40 00 00 68 51	..r...r. ..@@..hQ
0070	09 00 08 00 00 00 05 00	00 00 08 00 00 00 22 00"
0080	05 00 1a 00 47 56 4c 2e	61 73 74 41 78 65 73 5bGVL. astAxes [
0090	31 5d 2e 73 74 43 6f 6e	74 72 6f 6c 2e 66 56 65	1].stCon trol.fVe
00a0	6c 6f 63 69 74 79 00 4c	52 45 41 4c 00 56 65 6c	locity.L REAL.VeL
00b0	6f 63 69 74 79 20 6d 75	73 74 20 62 65 20 73 70	ocity mu st be sp
00c0	65 63 69 66 69 65 64 00	95 19 07 18 00 00 00 00	ecified.
00d0	00 00 00 00 00 00 00 0e	

AMS

AMS Target Net Id: 5.81.143.110.1.1

AMS Target port: 852

AMS Sender Net Id: 192.168.88.77.1.1

AMS Sender port: 10000

CmdId: ADS Add Device Notification (6)

> StateFlags: 0x0004

cbData: 40

ErrorCode: NO ERROR (0x00000000)

InvokeId: 0x8000029b

ADS Add Device Notification Request

IndexGroup: 0x0000f005

IndexOffset: 0x0180006f

CbLength: 4

Trans Mode: SERVER ON CHANGE (4)

Max Delay: 5000000

Cycle Time: 10000

Get notifications



```
AMS
AMS Target Net Id: 192.168.88.77.1.1
AMS Target port: 10000
AMS Sender Net Id: 5.81.143.110.1.1
AMS Sender port: 852
CmdId: ADS Device Notification (8)
> StateFlags: 0x0004
  cbData: 656
  ErrorCode: NO ERROR (0x00000000)
  InvokeId: 0x00000000
  ADS Device Notification Request
    CbLength: 652
    Count of Stamps: 18
```

```
050 04 00 90 02 00 00 00 00 00 00 00 00 00 8c 02 .....
060 00 00 12 00 00 00 e0 01 ca e9 a1 b7 dc 01 02 00 ..
070 00 00 2b 00 00 00 04 00 00 00 2f 16 00 00 2d 00 ..+...../..-
080 00 00 04 00 00 00 2f 16 00 00 80 88 cb e9 a1 b7 ...../.....
090 dc 01 02 00 00 00 2b 00 00 00 04 00 00 00 2e 16 .....+.....
0a0 00 00 2d 00 00 00 04 00 00 00 2e 16 00 00 20 0f ..-.....+.
0b0 cd e9 a1 b7 dc 01 02 00 00 00 2b 00 00 00 04 00 .....+.....
0c0 00 00 2c 16 00 00 2d 00 00 00 04 00 00 00 2c 16 ..,....-.....,
0d0 00 00 c0 95 ce e9 a1 b7 dc 01 02 00 00 00 2b 00 .....+.....+
0e0 00 00 04 00 00 00 2b 16 00 00 2d 00 00 00 04 00 .....+.....-
0f0 00 00 2b 16 00 00 60 1c d0 e9 a1 b7 dc 01 02 00 ..+....\.....
100 00 00 2b 00 00 00 04 00 00 00 29 16 00 00 2d 00 ..+.....)....-
110 00 00 04 00 00 00 29 16 00 00 00 a3 d1 e9 a1 b7 .....).....
120 dc 01 02 00 00 00 2b 00 00 00 04 00 00 00 27 16 .....+.....!
130 00 00 2d 00 00 00 04 00 00 00 27 16 00 00 a0 29 ..-.....!.....)
140 d3 e9 a1 b7 dc 01 02 00 00 00 2b 00 00 00 04 00 .....+.....
150 00 00 26 16 00 00 2d 00 00 00 04 00 00 00 26 16 ..&....-.....&
160 00 00 40 b0 d4 e9 a1 b7 dc 01 02 00 00 00 2b 00 ..@.....+
170 00 00 04 00 00 00 24 16 00 00 2d 00 00 00 04 00 .....$.-----
```

- Don't like to peek and poke in PLC variables?
Want a clean interface ?
The PLC has value + status (alarms)
The IOC asks which devices exist in the PLC !
PLC Interface Layout Specification
- Control system agnostic: EPICS or Tango
- Fetch all in “one go” with timestamp
- <https://forge.frm2.tum.de/public/doc/plc/v2.0/singlehtml/>

PILS example

- **Example:**
 - "PTPOffset", addr=128; value + status -> longin
 - "PTPState", addr=160; value + status -> mbbi
 - "PTPSyncSeqNum", addr=168, value 2 byte → longin
 - "PTPErrorStatus", addr=172, bits + status → longin + stringin
Bit15: CableNotConnected, Bit14: RdDiagError, Bit 11: NotSynchronized
 - "Motor_M1", addr=264, setpoint, act pos, status, parameter -> motor

Summary

- Drivers using ADS:
- epics-modules/twincat-ads
 - generic
 - motion
 - Can use handles for input

Summary

- Drivers using ADS:
- Cosylab/AdsDriver:
 - generic
 - many updates/second
 - using handles

- PILS
 - generic, motion
 - timestamped motion
 - alarms from PLC into EPICS
 - variables in PLC are hidden
 - “handshake”: PLC may reject writes
 - flat memory, self-describing

TODO

- Questions ?
- Over to EtherCAT train