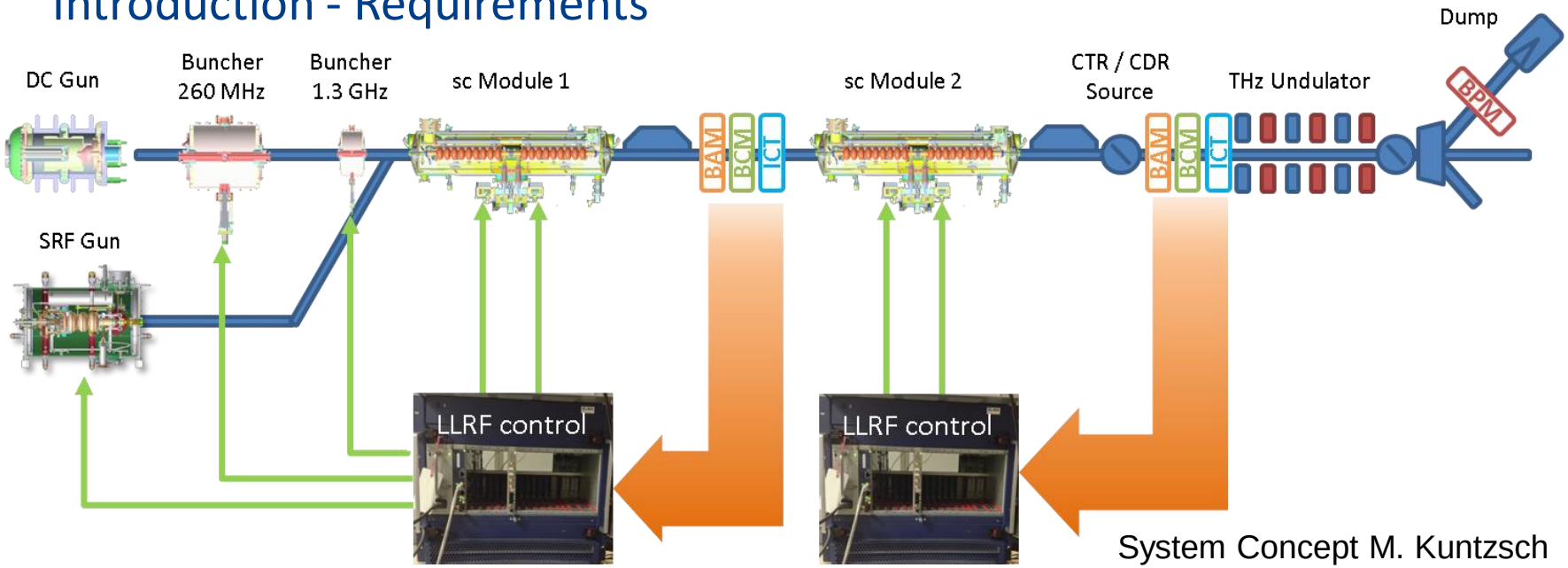
2nd Annual MT Meeting - Control Systems in MT

MTCA LLRF - ELBE Control System Integration

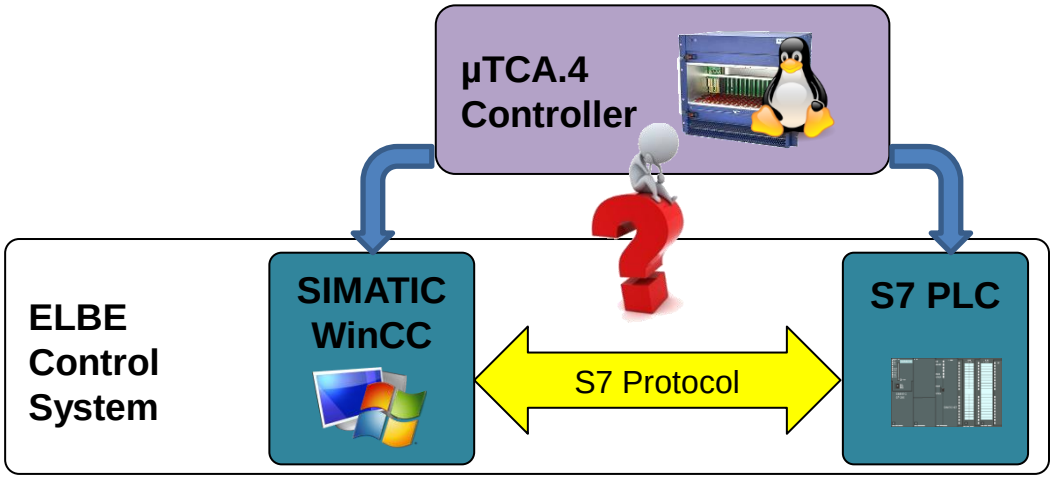
Status OPC UA Adapter

MTCA LLRF - ELBE Control System Integration

Introduction - Requirements



System Concept M. Kuntzsch



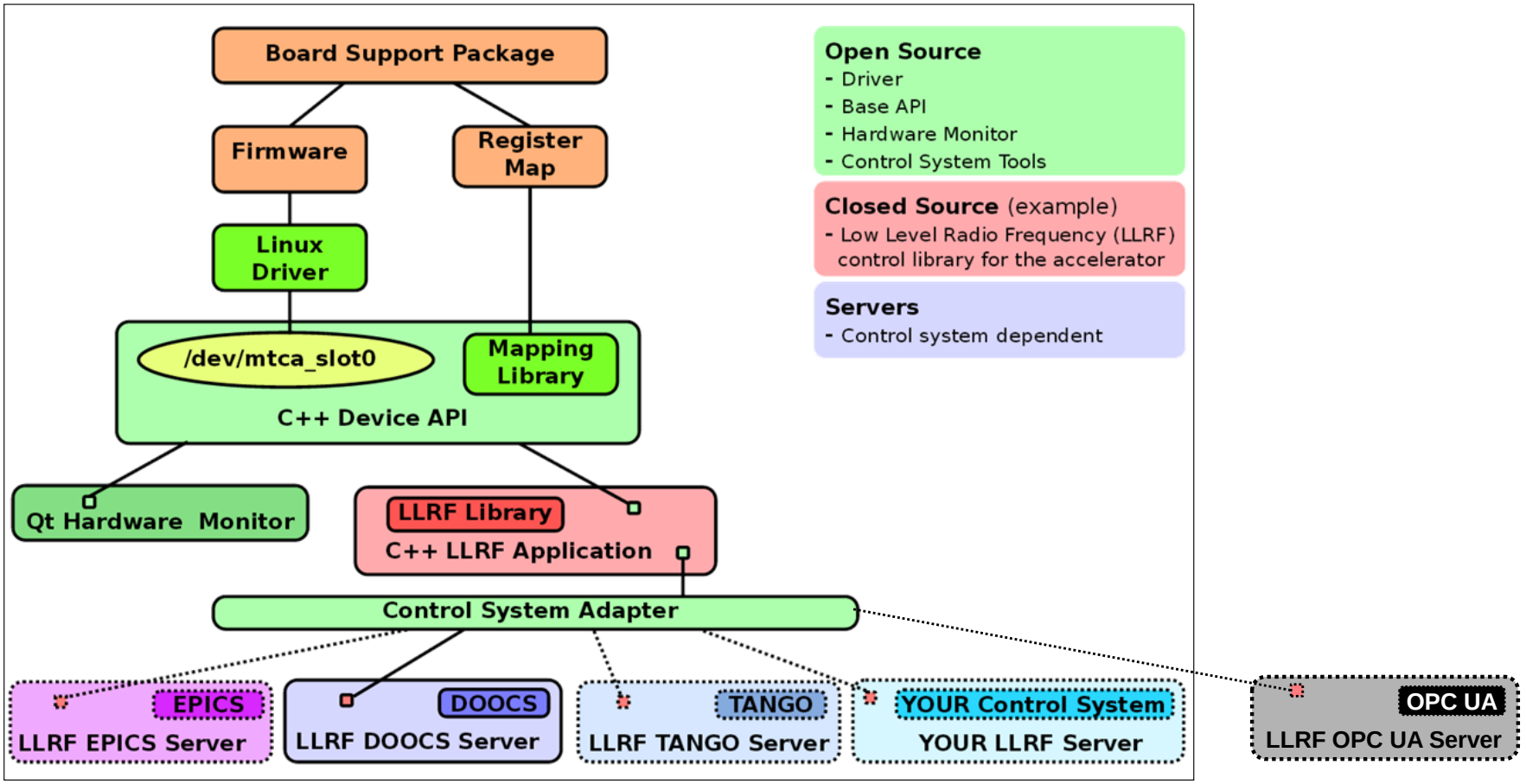
MTCA LLRF - ELBE Control System Integration

Introduction - Requirements

Requirements		OPC	OPC UA	Modbus TCP	Fieldbusses
Platform independence					
- Hardware	MTCA, PC, PXI, PLC	(-)	(+)	(+)	(+)
- Operating Systems	Linux, Windows	(-)	(+)	(+)	(+)
- Programming/Engineering Tools	C++, LabVIEW, WinCC	(-)	(+)	(+)	(+)
Process Data					
- Number of tags per Cavity	~100	(+)	(+)	(+)	(0)
- Data Types	elementary	(+)	(+)	(+)	(+)
	arrays of elementary (16384 elements)	(0)	(+)	(+)	(-)
- Data rate	on demand … 10 Hz	(+)	(+)	(+)	(+)
Bus Technology	Ethernet based	(+)	(+)	(+)	(+)
Licensing	Open source (compatible to MTCA4U licence scheme)	(-)	(+)	(+)	(-)
Reusability	Reusable for other MTCA4U applications	(-)	(+)	(0)	(0)
Security	Encryption (-), Authentication (0), Data Integrity (+)	(0)	(+)	(-)	(0)
Maintainability	Easy to Maintain, Extend, Engineering Tools	(+)	(+)	(-)	(0)
Simplicity	Easy to learn/implement	(-)	(-)	(+)	(-)

MTCA LLRF - ELBE Control System Integration

Introduction – Software Scheme



MTCA LLRF - ELBE Control System Integration

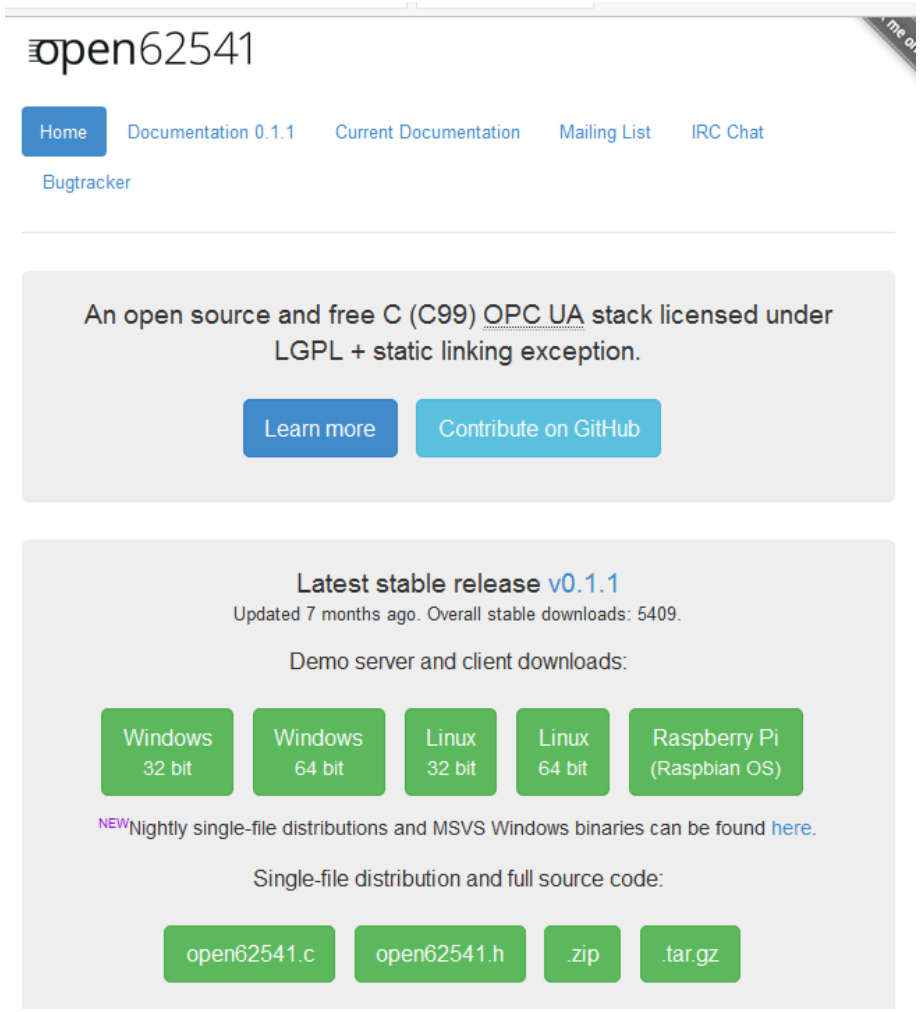
„OPC UA Adapter History“

Date	Event	Location	Topics
2014-12-10	3rd MicroTCA Workshop for Industry and Research	DESY, Hamburg	MTCA.4 interface for ELBE?
2015-02-23	POF Kick-OFF - MTCA Expert Session	DESY, Hamburg	discussion on MTCA4U adapter and possible OPC UA integration
2015-07-16	3rd ARD ST3 workshop	KIT, Karlsruhe	LLRF integration concept for ELBE with OPC UA presented on poster
2015-11-25	ARD-ST3-MiniWorkshop - Expert Session on OPC-UA based LLRF	HZDR, Dresden	Introduction of open62541 by TUDD, 1st draft of cooperation
2015-12-09	4th MicroTCA Workshop for Industry and Research	DESY, Hamburg	OPC UA is announced as being part of MTCA4U
2016-01-07	Project Discussion (TelCo) OPCUA4MTCA – DESY, TUDD, ELBE, AQUENOS	HZDR, Dresden	agreement to implement an OPC UA adapter based on open62541 on management level
2016-02-15	Project Discussion OPCUA4MTCA – TUDD, ELBE	TUDD, Dresden	planning next steps of cooperation …

MTCA LLRF - ELBE Control System Integration

open62541 - Open Source OPC UA Stack

http://open62541.org/



Open

- stack design based solely on IEC 62541
- licensed under open source (LGPL & static linking exception)
- royalty free, available on GitHub

Scalable

- single or multi-threaded architecture
- one thread per connection/session

Maintainable

- 85% of code generated from XML specification files

Portable

- written in C99 with POSIX support
- compiled server is smaller than 100kb
- runs on Windows (x86, x64), Linux (x86, x64, ARM e.g. Raspberry Pi, SPARCstation), QNX and Android

Extensible

- dynamically loadable and reconfigurable user models

More info?

Dig into the documentation, check the bugtracker or read the whitepaper (coming soon)

MTCA LLRF - ELBE Control System Integration

Cooperation Partners

open62541 Initiators:

ACPLT Chair of Process Control Engineering | RWTH AACHEN UNIVERSITY

Fraunhofer IOSB

TECHNISCHE UNIVERSITÄT DRESDEN — Chair of PROCESS CONTROL SYSTEMS ENGINEERING

TU Dresden Expert Team:

Prof. Dr.-Ing. habil. Leon Urbas
Dipl.-Ing. Chris Iatrou
Dipl.-Ing. Stephan Hensel

Cooperation for Implemetation of OPC UA adapter for MTCA4U

Partners:

DESY	(MTCA4U)
Aquenos	(MTCA4U adapter layer)
TUDD	(open62541 OPC UA stack)
HZDR	(CW LLRF integration)

DRESDEN concept HZDR

OPC-UA Adapter for MTCA4U

Cooperation Plans

- cooperation starts with a development assignment (HZDR -> TUDD)

Project Month	Working Packages	Main Responsibility
1	**analysis** of current adapter implementation of MTCA4U framework	**TUDD**
	development of **mapping schemes** to transfer process variables and meta information between adapter layer and OPC UA interface	**TUDD**
2 - 3	definition of **mapping scheme**	**DESY, Aquenos, TUDD, HZDR**
	development of **OPC UA information model**	**TUDD, HZDR**
4 – 4,5	definition of **adapter feature set**	**DESY, Aquenos, TUDD, HZDR**
4,5 – 5,5	implementation of **OPC UA adapter** features	**DESY, Aquenos, TUDD**
6	Project completion: Tested and documented **OPC UA adapter** for MTCA LLRF and software dummy device at ELBE	**TUDD, HZDR**

- cooperation will continue with PhD student(s)

OPC-UA Adapter for MTCA4U

open62541 Demo Implementation - Preconditions

Chris Iatrou from TU Dresden implemented a Demo OPCUA Server as proof of principle based on:

- open62541 Stack,
- MTC4U “mapped-device”.

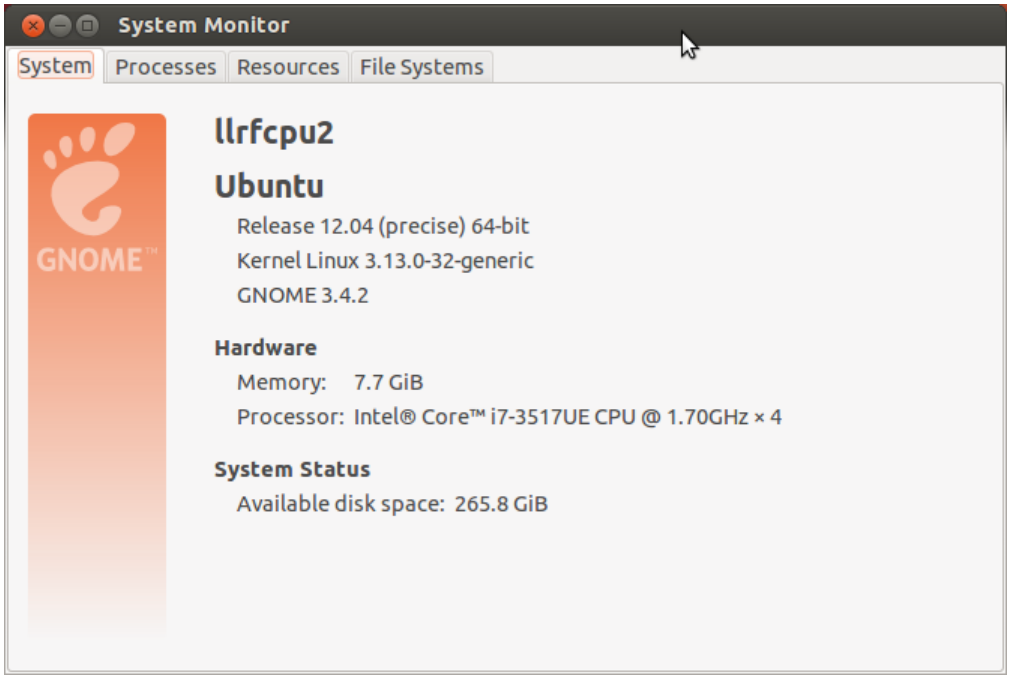
Test Setup at ELBE

MTCA hardware:

- 7 Struck ADC AMCs with LLRF RTMs
- X2Timer AMC (NAMC-psTimer)
- NAT MCH
- NAT CPU

Software

- currently local DOOCS servers necessary for test and operation
- X2Timer forced downgrade to Ubuntu 12 (no DOOCS server available for Ubuntu 14)



OPC-UA Adapter for MTCA4U

open62541 Demo Implementation - OPC-UA Adress Space

OPC UA adress space is automatically generated based on *.map and *.dmap files

DMAP

-> device map

Lister - [c:\Users\rs6082\Dropbox\Literatur\DIENSTREISEN\...

File Edit Options Encoding Help 100 %

```
DUMMY1   /dev/mtcadummys0 ./mtcadummy_withoutModules.map
DUMMY2   /dev/mtcadummys1 ./mtcadummy.map
DUMMY3   /dev/mtcadummys2 ./mtcadummy_bad.map
DUMMY4   /dev/mtcadummys3 ./mtcadummy_bad.map
DUMMY5   /dev/llrfdummys4 ./mtcadummy.map
DUMMY6   /dev/doesNotExist ./mtcadummy.map
```

MAP

-> register map

Lister - [c:\Users\rs6082\Dropbox\Literatur\DIENSTREISEN\2016-03-08 2nd Annual MT Meeting\OPC-UA-Anbindung\Software_unpack\mtcaOpen62541\map...

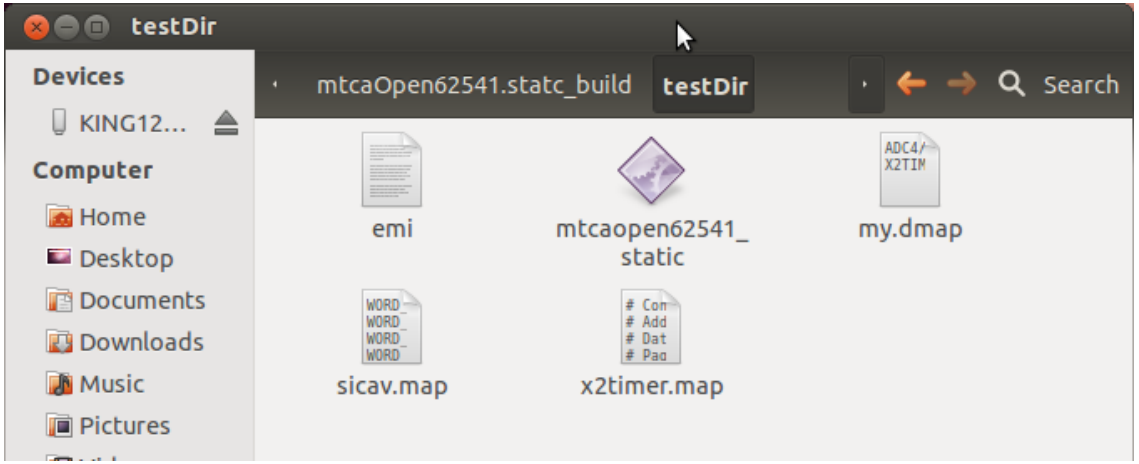
File Edit Options Encoding Help 77 %

```
# name                    nr of elements      address          size          bar     width   fracbits    signed
BOARD.WORD_FIRMWARE          0x00000001    0x00000000    0x00000004    0x00000000       32         0         0
BOARD.WORD_COMPILATION       0x00000001    0x00000004    0x00000004    0x00000000       32         0         0
BOARD.WORD_STATUS            0x00000001    0x00000008    0x00000004    0x00000000       32         0         0
BOARD.WORD_USER              0x00000001    0x0000000C    0x00000004    0x00000000       12         3         1
ADC.WORD_CLK_CNT             0x00000002    0x00000010    0x00000008    0x00000000       32         0         0
ADC.WORD_CLK_CNT_0           0x00000001    0x00000010    0x00000004    0x00000000       32         0         0
ADC.WORD_CLK_CNT_1           0x00000001    0x00000014    0x00000004    0x00000000       32         0         0
ADC.WORD_CLK_MUX             0x00000004    0x00000020    0x00000010    0x00000000       32         0         0
ADC.WORD_CLK_MUX_0           0x00000001    0x00000020    0x00000004    0x00000000       32         0         0
ADC.WORD_CLK_MUX_1           0x00000001    0x00000024    0x00000004    0x00000000       32         0         0
ADC.WORD_CLK_MUX_2           0x00000001    0x00000028    0x00000004    0x00000000       32         0         0
ADC.WORD_CLK_MUX_3           0x00000001    0x0000002C    0x00000004    0x00000000       32         0         0
ADC.WORD_CLK_DUMMY           0x00000001    0x0000003C    0x00000004    0x00000000       32         0         0
ADC.WORD_CLK_RST             0x00000001    0x00000040    0x00000004    0x00000000       32         0         0
ADC.WORD_ADC_ENA             0x00000001    0x00000044    0x00000004    0x00000000       32         0         0

#The broken register cannot be read or written, it always causes an I/O error for testing
TESTING.BROKEN_REGISTER      0x00000001    0x00000048    0x00000004    0x00000000       32         0         0
#Reading is possible, but writing causes an I/O error. Content is the offset.
TESTING.BROKEN_WRITE         0x00000001    0x0000004C    0x00000004    0x00000000       32         0         0

#Simulate spi via pcie with handshake:
# - write 0xff to the SPI SYNC register
```

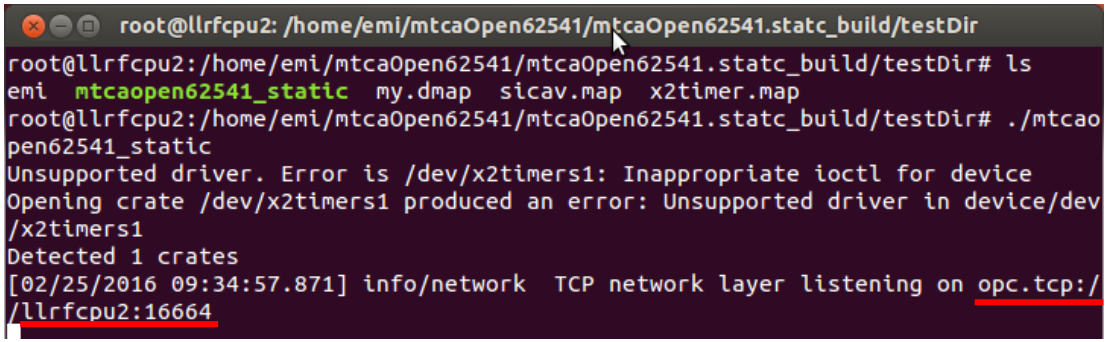
OPC-UA Adapter for MTCA4U

open62541 Demo Implementation – Start Up

Application with map files:



Application start up:



server endpoint: opc.tcp://hostname:port

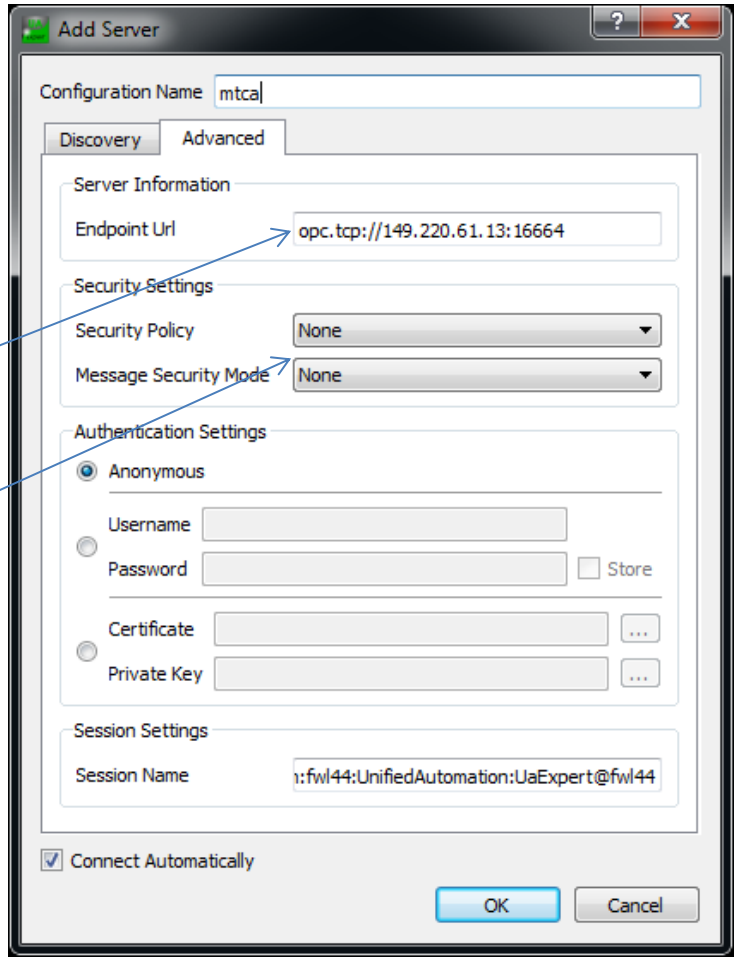
OPC-UA Adapter for MTCA4U

open62541 Demo Implementation – connect OPC UA Client (uaexpert)

Freely available OPC UA Client uaexpert from Unified Automation was used for tests.

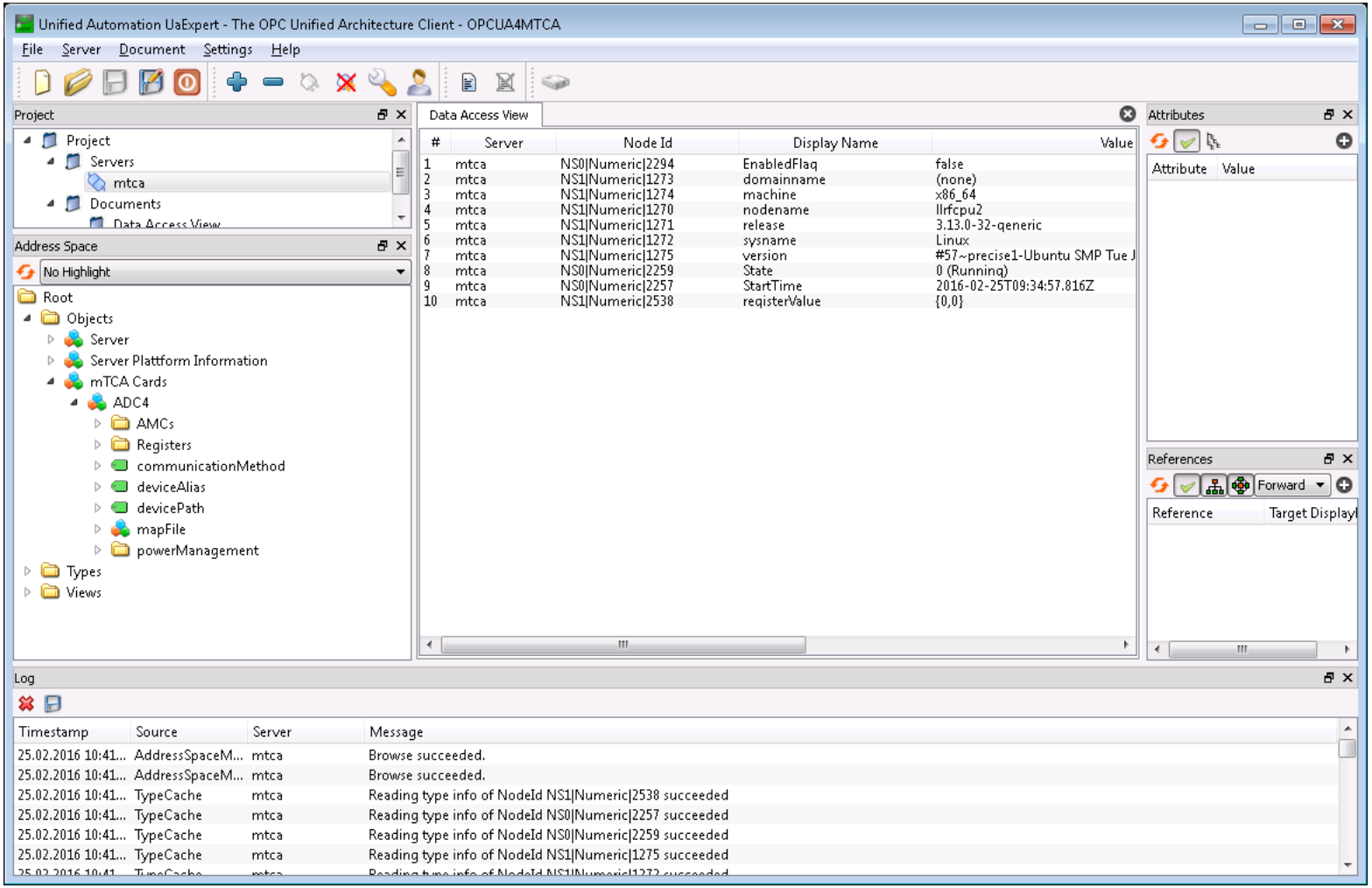
insert server endpoint

security level: none



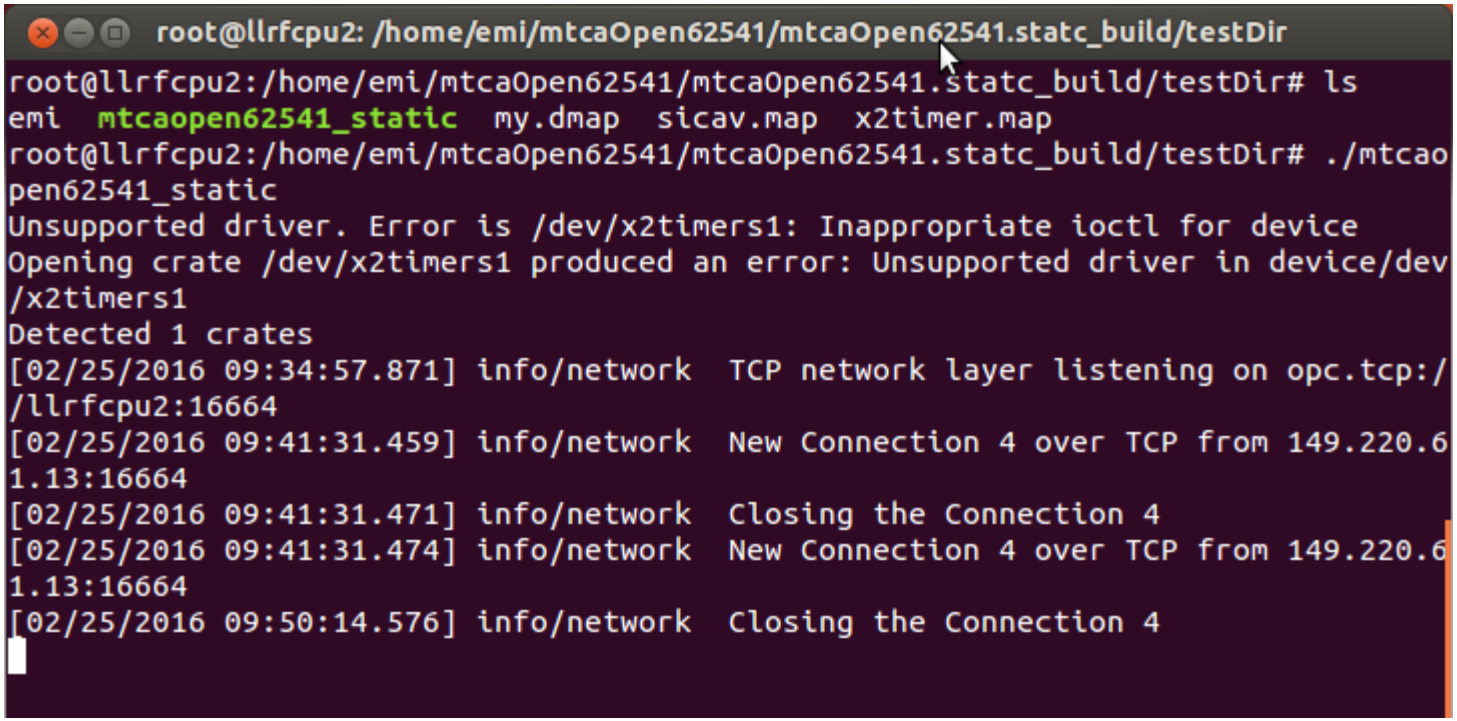
OPC-UA Adapter for MTCA4U

open62541 Demo Implementation – browse adress space



OPC-UA Adapter for MTCA4U

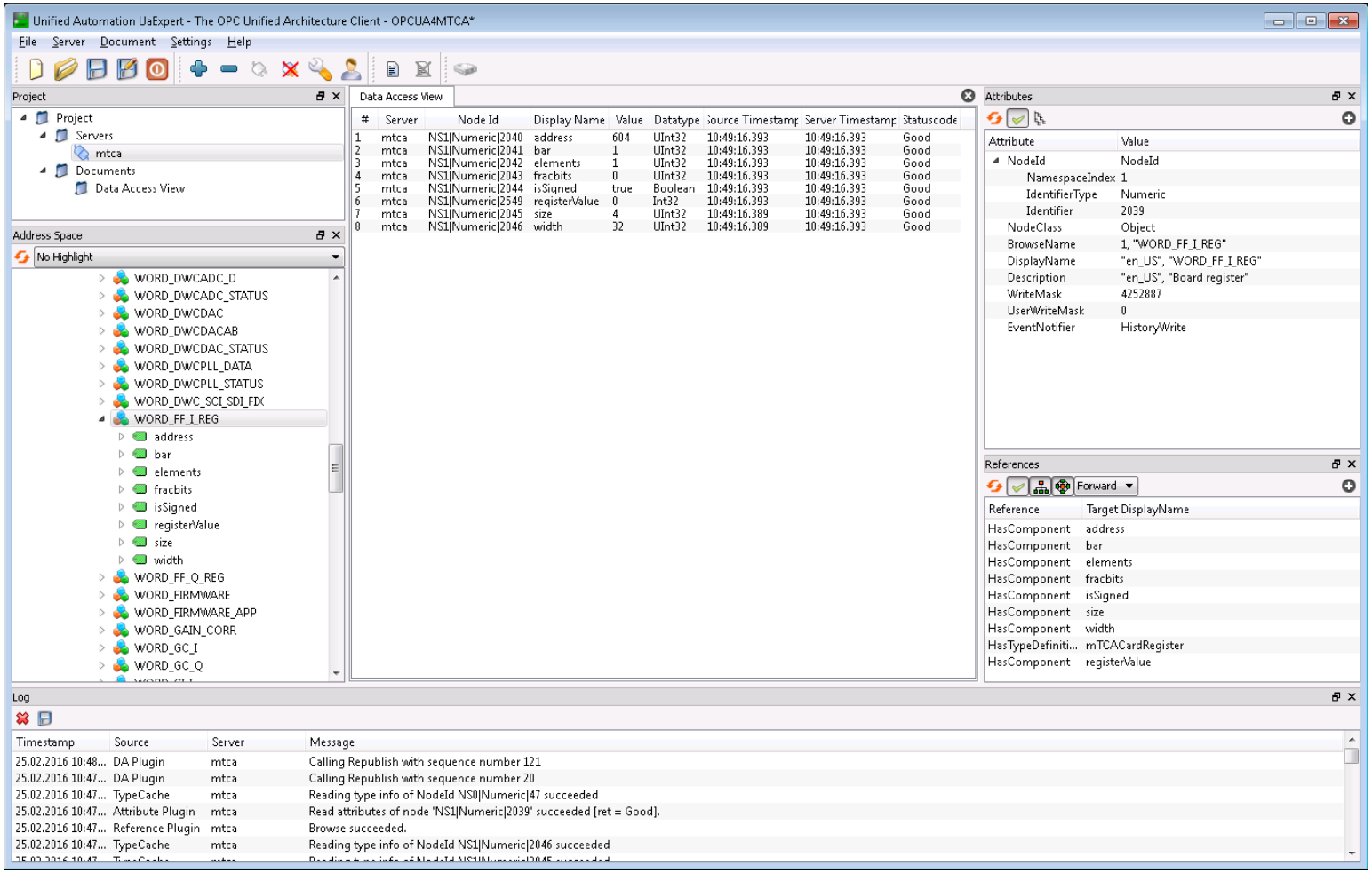
open62541 Demo Implementation – server shows client connection



Remark: Demo Server is bound to therminal where it was started. Closing the terminal closes the server.

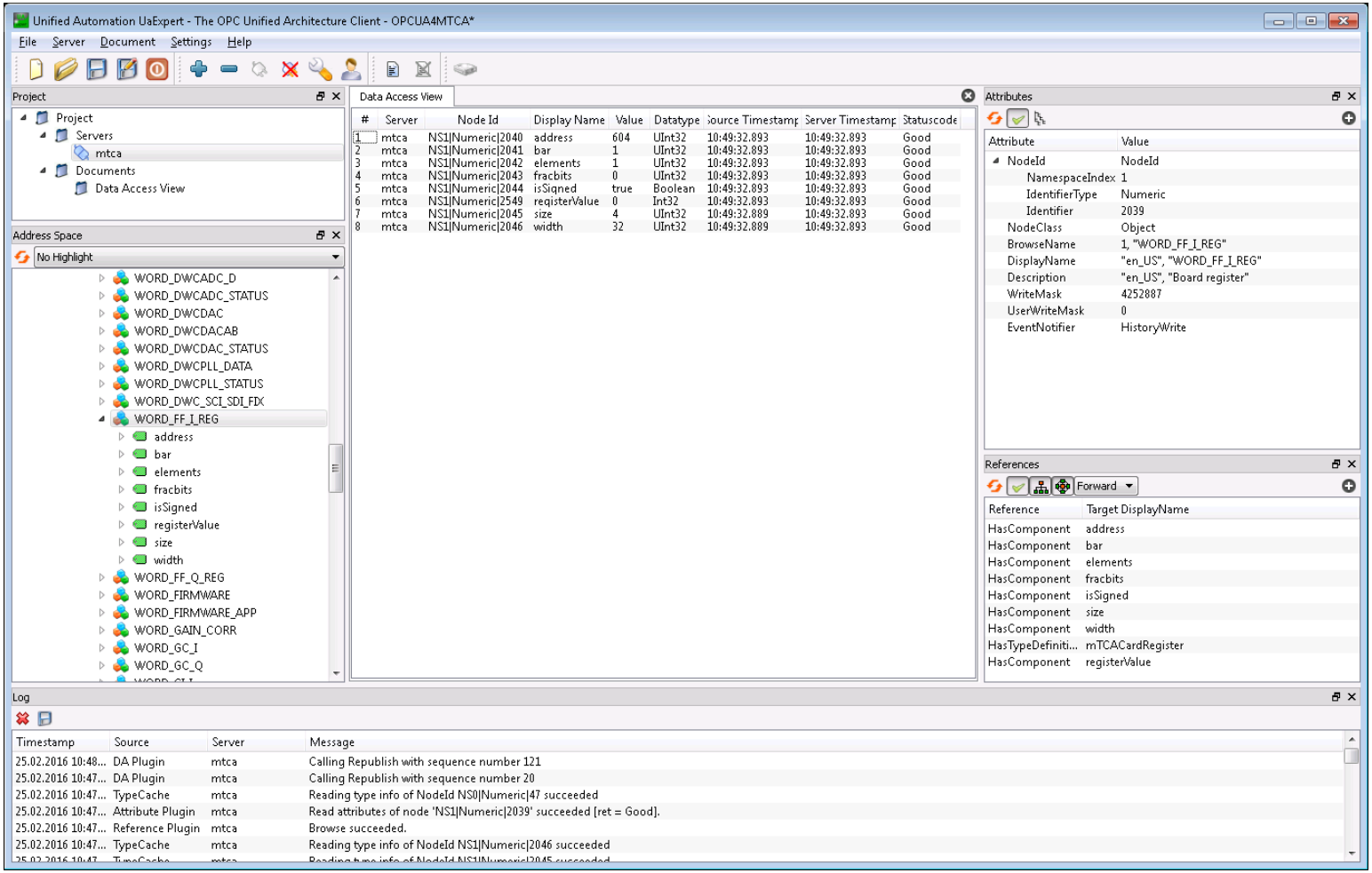
OPC-UA Adapter for MTCA4U

open62541 Demo Implementation – reading data



OPC-UA Adapter for MTCA4U

open62541 Demo Implementation – reading data



OPC-UA Adapter for MTCA4U

open62541 Demo Implementation – reading data