

ChimeraTK Software Framework



Jump-start control application development with ChimeraTK

Dietrich Rothe, Martin Killenberg, DESY – MSK

Dec 7 2023

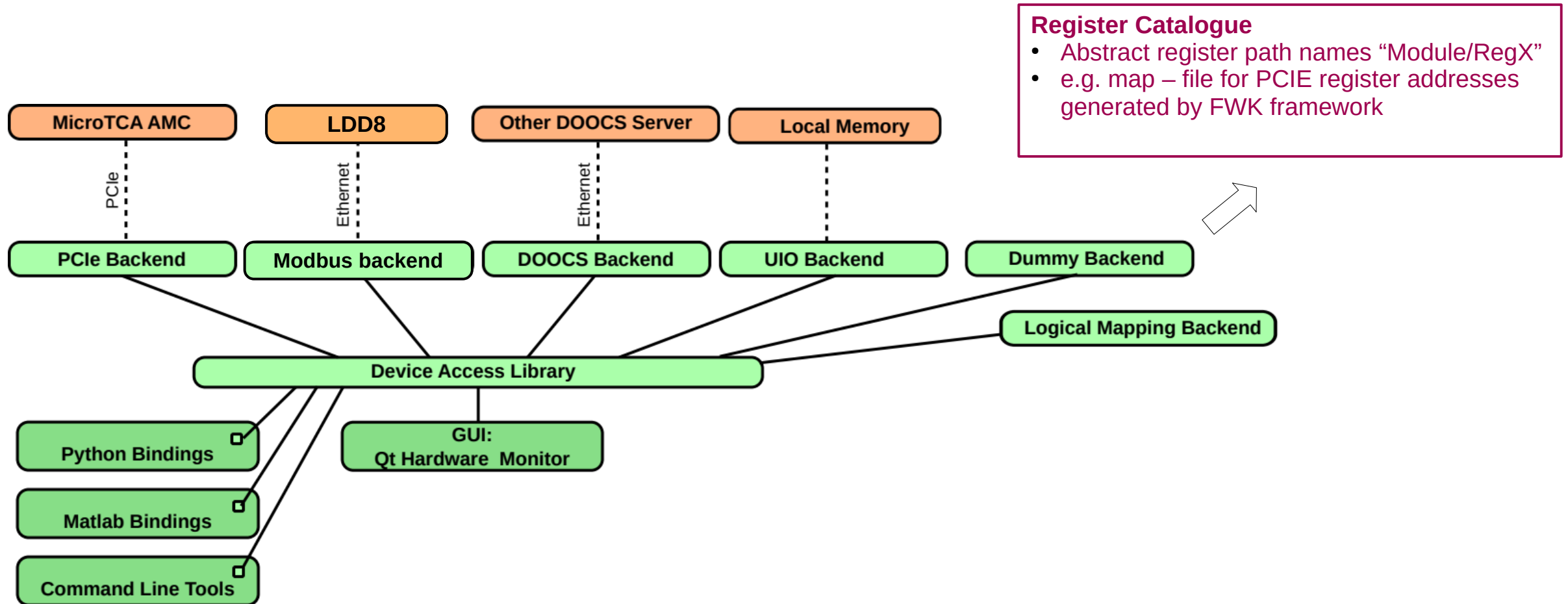
12th MicroTCA Workshop for Industry and Research

DESY, Hamburg



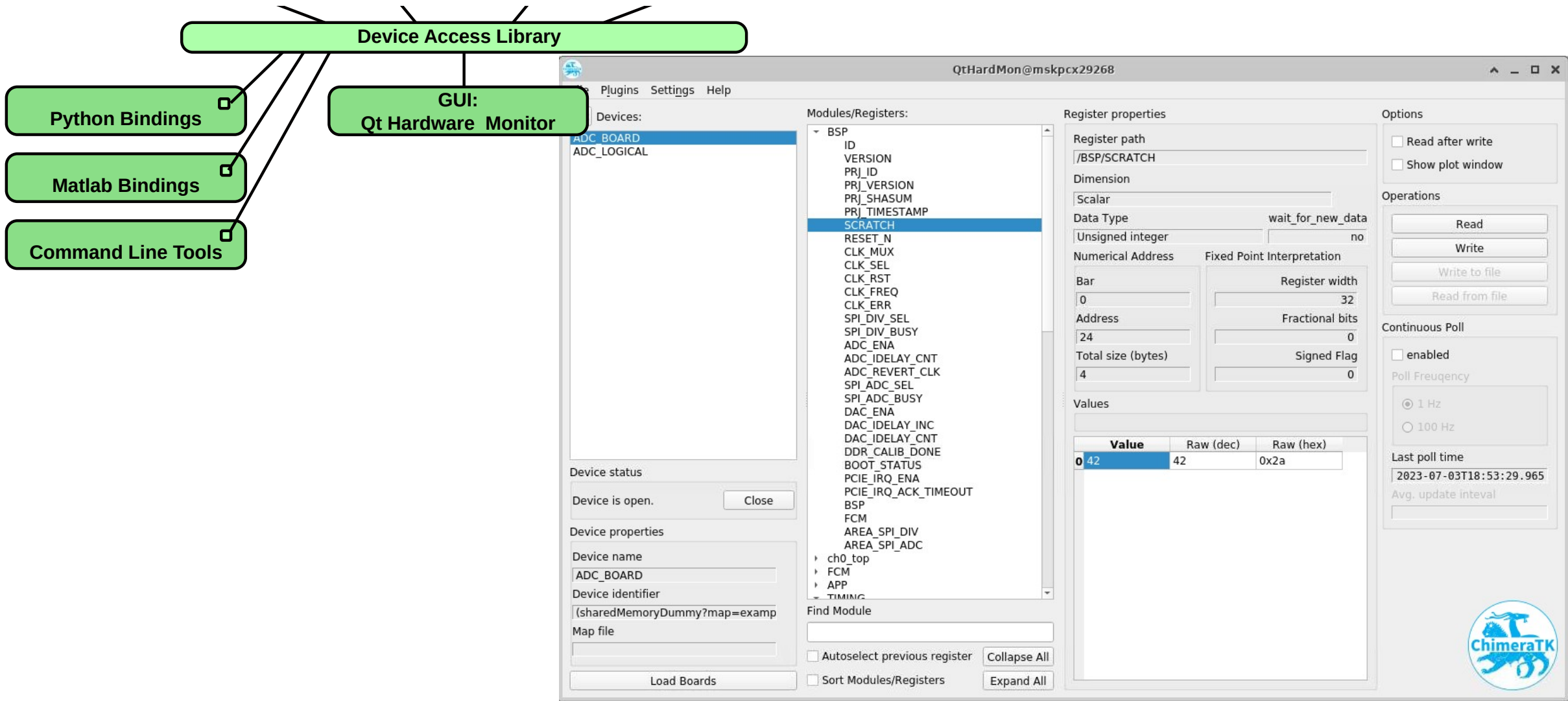
ChimeraTK Framework Overview

DeviceAccess



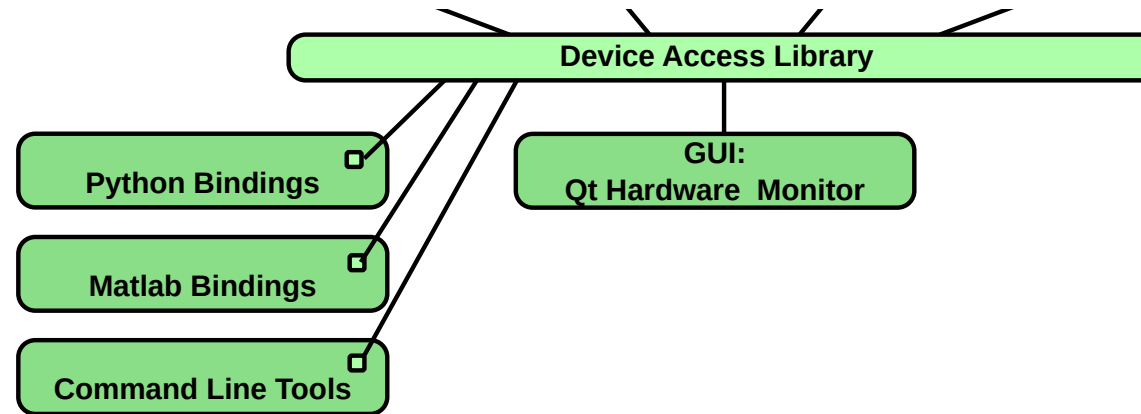
ChimeraTK Framework Overview

DeviceAccess



ChimeraTK Framework Overview

DeviceAccess



```
# initscript.py
import deviceaccess as da
import numpy as np

da.setDMapFilePath('devices.dmap')
d = da.Device('ADC_BOARD')
d.open()

d.write('TIMING', 'DIVIDER_VALUE', 1249999)

...
```

Logical Device

Logical name map - abstract technical firmware details

- Select & rename registers
- (Re-)define read/write access
server initialises all writeable automatically!
- Adjust data types
- Define new variables
- Apply simple formulas

Use with same DeviceAccess API as raw device

⇒ Use same tools for testing!

⇒ Enables better abstraction in server logic!

```
<logicalNameMap>
...
  <module name="Attenuator">
    <redirectedRegister name="setpoint">
      <targetDevice>ADC_BOARD</targetDevice>
      <targetRegister>RTM/ATT_VAL</targetRegister>
      <plugin name="math">
        <!-- converts dB to ADC counts -->
        <parameter name="formula"> x*2.7 </parameter>
      </plugin>
    </redirectedRegister>
  </module>

  <module name="SetupMonitor">
    <redirectedRegister name="clockDivider">
      <targetDevice>ADC_BOARD</targetDevice>
      <targetRegister>/TIMING/DIVIDER_VALUE</targetRegister>
      <plugin name="forceReadOnly"/>
    </redirectedRegister>
  </module>
</logicalNameMap>
```

Configure GenericDeviceServer

Control System Integration

- **Add raw devices**
 - List devices and access protocols $\Rightarrow$ devices.dmap
 - Register set: *.map
- **Initialization script!**
Use DeviceAccess Python bindings on raw devices
- **Add Logical Device: *.xmap**
- **Configure readout trigger**
 - Simple software PeriodicTimer
 - **Interrupt**
 - Another backend

```
# devices.dmap
ADC_BOARD (xdma:xdma/slot3?map=example_dwc8vm1.mapp)
ADC_LOGICAL (logicalNameMap?map=example_dwc8vm1.xlmap)
```

generic_chimeratk_server_configuration.xml

```
<configuration>

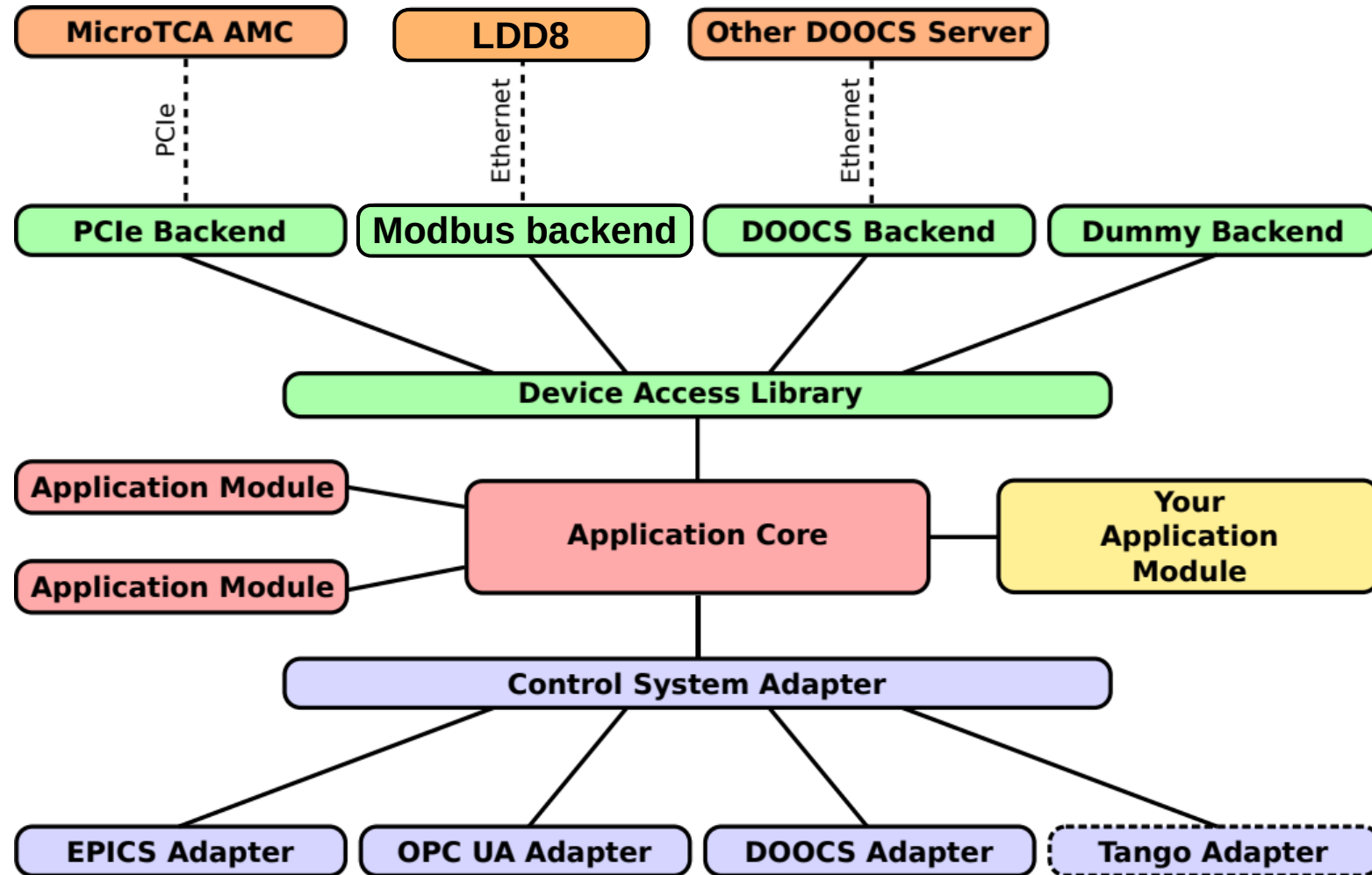
  <variable name="devices" type="string">
    <value i="0" v="ADC_LOGICAL"/>
  </variable>

  <module name="ADC_LOGICAL">
    <variable name="triggerPath" type="string"
      value="/ADC_LOGICAL/interrupts/dataReady"/>
    <variable name="pathInDevice" type="string" value=""/>
    <variable name="initScript" type="string" value="./initscript.py"/>
  </module>

</configuration>
```

ChimeraTK Framework Overview

Ecosystem



Control System Adapter for DOOCS

- Fully working device server just by configuration
- GenericDeviceServer** in use at **FLASH** master oscillator!

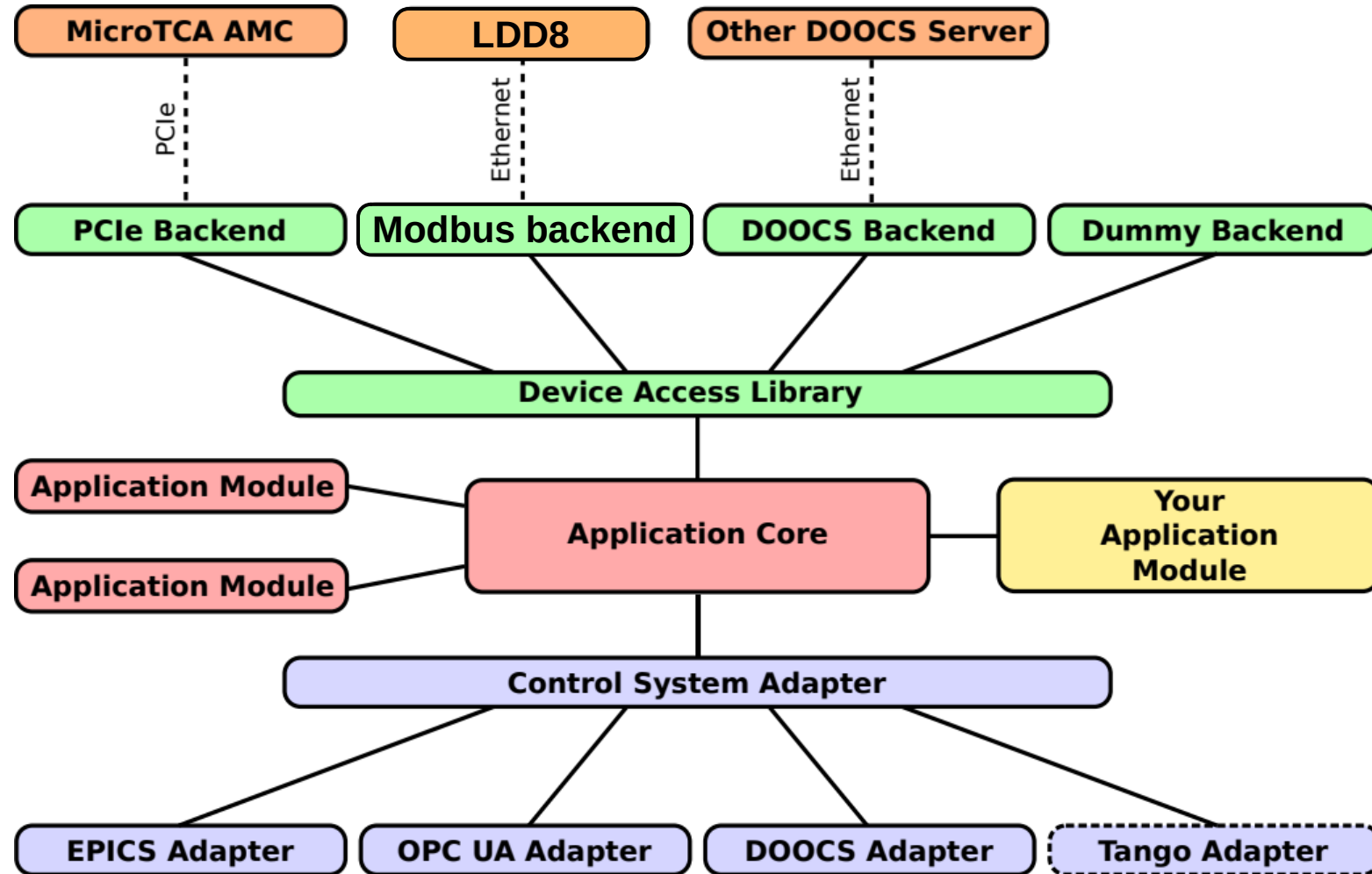
The screenshot displays the 'rpc_test.xml' interface for the 'FLASH.RF/LLRF.GEN1300/CH1.MO.TMCB/' system. It features four filter sections: Facility Filter (FLASH.RF), Device Filter (LLRF.GEN1300), Location Filter (CH1.MO.TMCB), and Properties Filter (CLEAR_PLL100.IENA). The main table lists 115 values, including various CLEAR and DEVICE parameters with their current values and timestamps. The bottom status bar shows 'DOOCS hosts= flashutil1 lib = 610499993 (server_mask = 4 auth_mask = 0 options = 0 status = 0)'.

FLASH.RF/LLRF.GEN1300/CH1.MO.TMCB/*	115 Values	type
CLEAR_PLL100.IENA	0	423
CLEAR_PLL100.IENA.HIST history	"04. Dec. 2023 12:35:15.153" 1.7016897...	423
CLEAR_PLL100.LockAC	0	423
CLEAR_PLL100.LockAC.HIST history	"04. Dec. 2023 12:35:15.153" 1.7016897...	423
CLEAR_PLL100.LockDC	0	423
CLEAR_PLL100.LockDC.HIST history	"04. Dec. 2023 12:35:15.153" 1.7016897...	423
CLEAR_PLL100.OutPwrOK	0	423
CLEAR_PLL100.OutPwrOK.HIST history	"04. Dec. 2023 12:35:15.153" 1.7016897...	423
CLEAR_PLL100.PWRGD	0	423
CLEAR_PLL100.PWRGD.HIST history	"04. Dec. 2023 12:35:15.153" 1.7016897...	423
CLEAR_PLL100.RefPwrOK	0	423
CLEAR_PLL100.RefPwrOK.HIST history	"04. Dec. 2023 12:35:15.153" 1.7016897...	423
CLEAR_PLL100.TAL1	0	423
CLEAR_PLL100.TAL1.HIST history	"04. Dec. 2023 12:35:15.153" 1.7016897...	423
CLEAR_PLL100.TAL2	0	423
CLEAR_PLL100.TAL2.HIST history	"04. Dec. 2023 12:35:15.153" 1.7016897...	423
CLEAR_PLL1300.LockAC	0	423
CLEAR_PLL1300.LockAC.HIST history	"04. Dec. 2023 12:35:15.153" 1.7016897...	423
CLEAR_PLL1300.LockDC	0	423
CLEAR_PLL1300.LockDC.HIST history	"04. Dec. 2023 12:35:15.153" 1.7016897...	423
CLEAR_PLL1300.N150K	0	423
CLEAR_PLL1300.N150K.HIST history	"04. Dec. 2023 12:35:15.153" 1.7016897...	423
CLEAR_PLL1300.OvenLock	0	423
CLEAR_PLL1300.OvenLock.HIST history	TS int array length = 3	423
CLEAR_PLL1300.P150K	0	423
CLEAR_PLL1300.P150K.HIST history	"04. Dec. 2023 12:35:15.153" 1.7016897...	423
CLEAR_PLL1300.RefPwrOK	0	423
CLEAR_PLL1300.RefPwrOK.HIST history	"04. Dec. 2023 12:35:15.153" 1.7016897...	423
DEVICE.INFO edit info about the device	0 0.0 0.0 0 None	423
DEVICE.METADATA device metadata in XML		423
DOOCS1Temp	21.40621	423

DOOCS hosts= flashutil1 lib = 610499993 (server_mask = 4 auth_mask = 0 options = 0 status = 0)

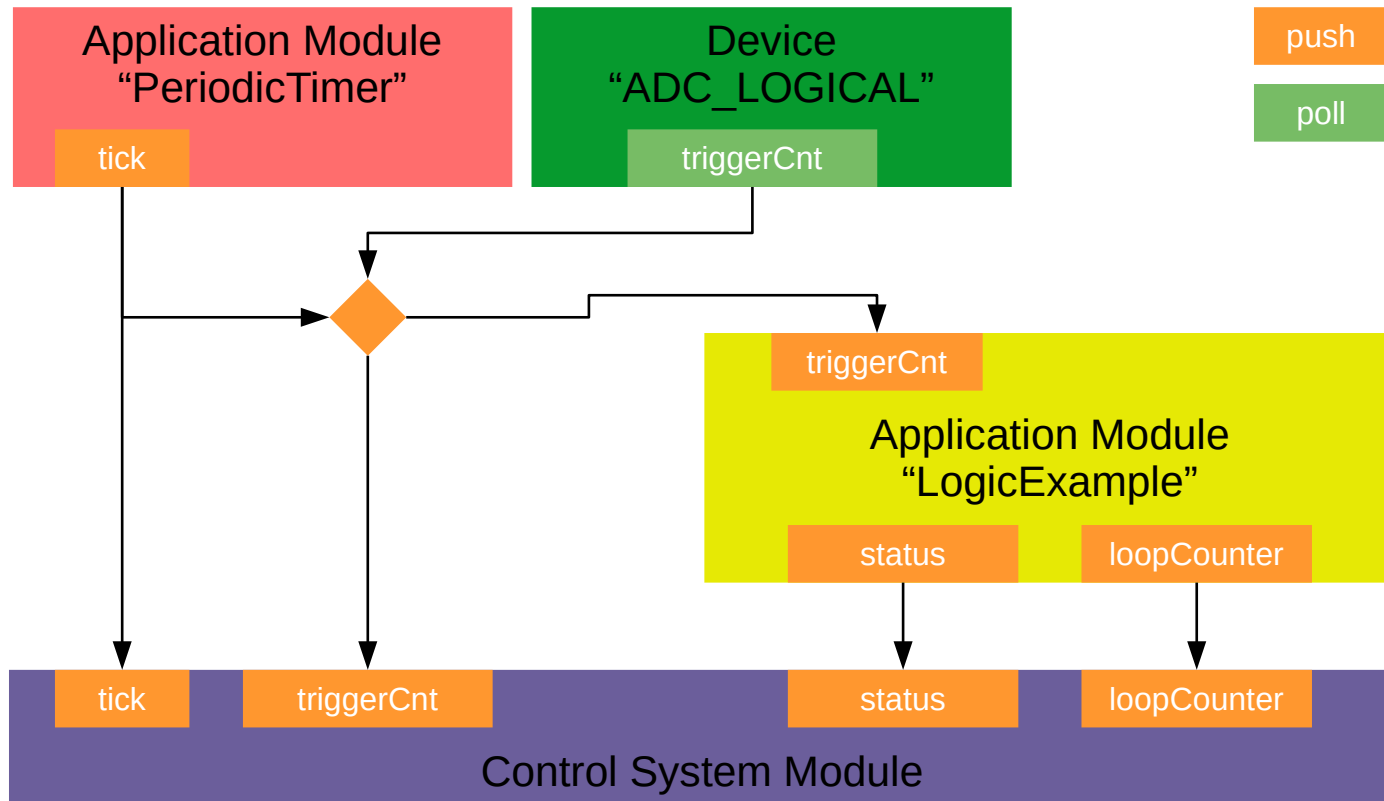
Extend server by business logic

ChimeraTK overview reminder



Extend server by business logic

ApplicationModule in C++



Register = Process Variable = CS Property

- New business logic as ApplicationModule
- Self-contained
- Runs in its own thread
- Modern inter-thread communication with lock-free queues
- Connection code is automatically generated
- Variables of same path/name are automatically connected
- Push-inputs are automatically connected to the trigger and the polled register

C++ code: only new ApplicationModule. All other functionality comes from the framework!

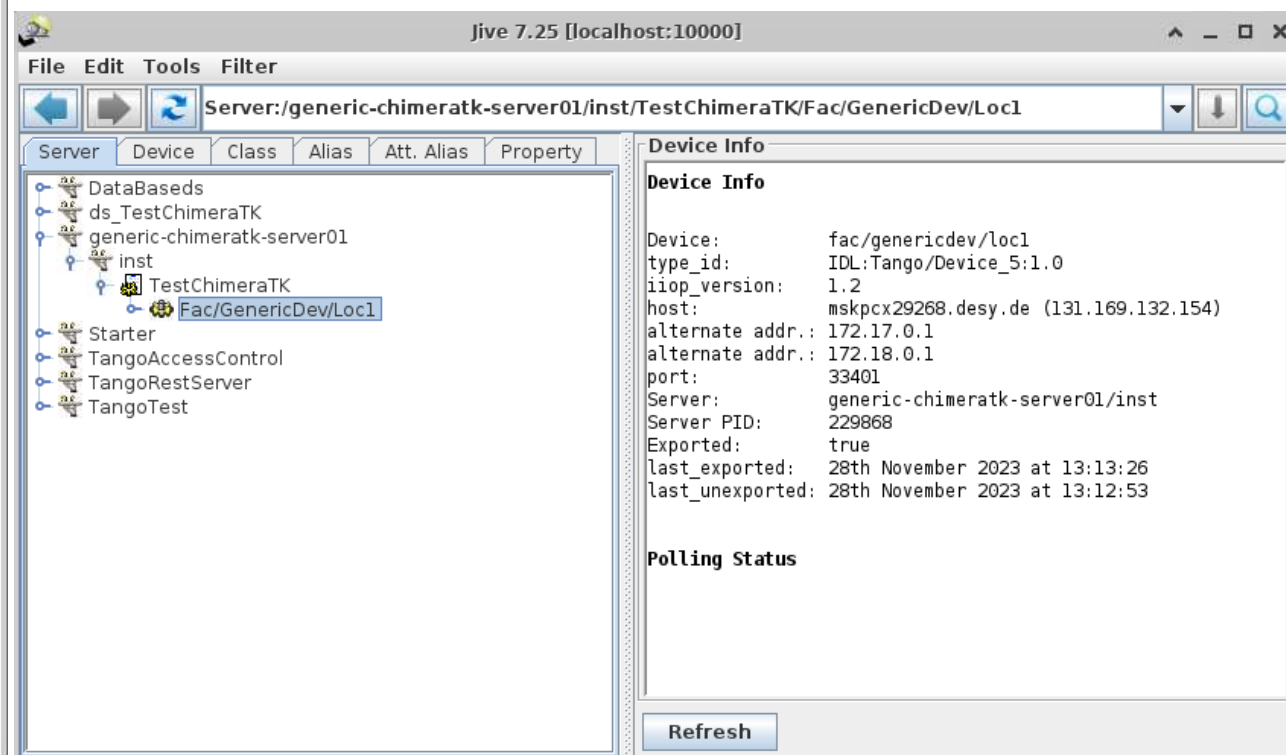
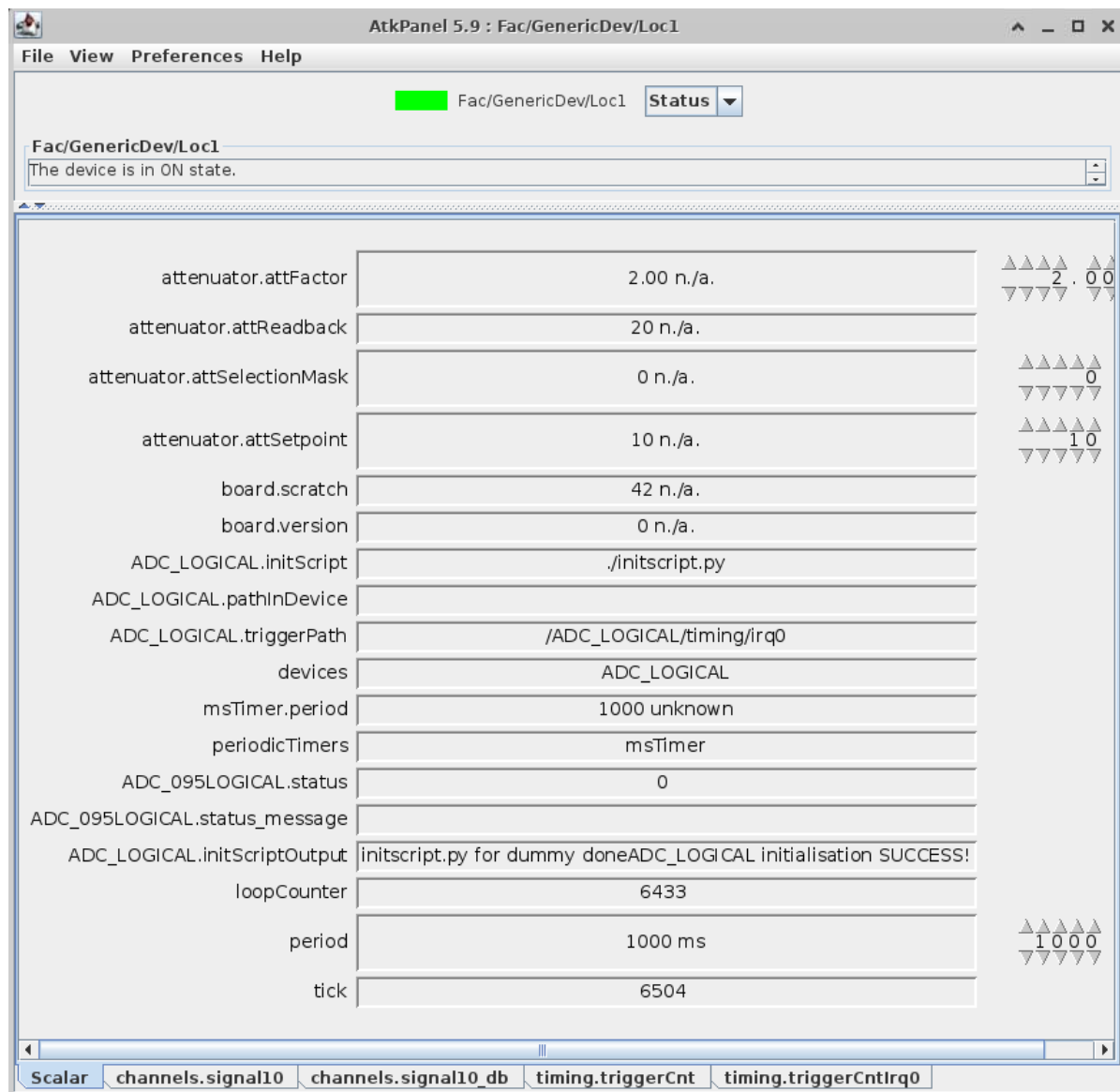
New: Control System Adapter for Tango



Currently under development by

EXPERIMENTAL!

```
$ cmake -DADAPTER="TANGO" -S ../GenericDeviceServer/ . ; make
$ ./tango-generic-chimeratk-server01
```



New: Support for embedded SoC

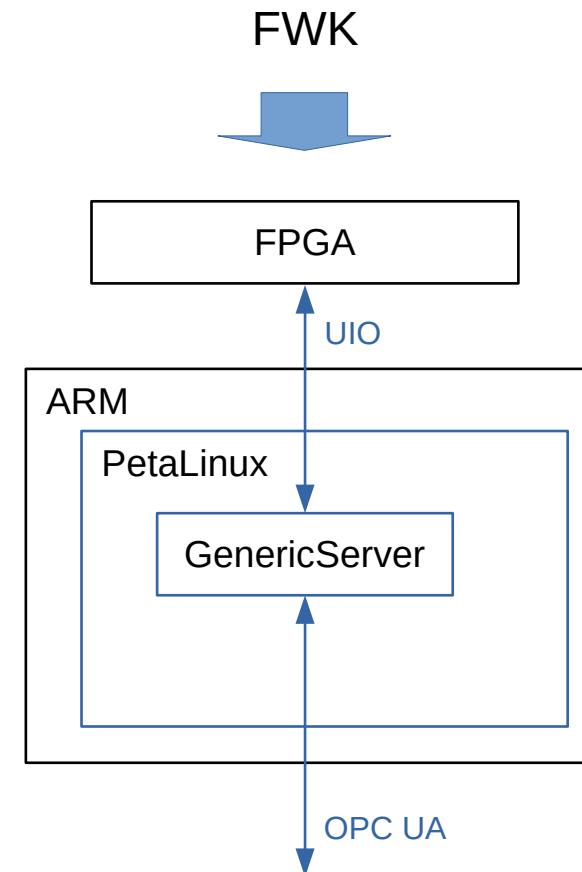
System on chip environment

- Linux Userspace Input/Output (UIO) backend for communication to FPGA
- Yocto Linux
ChimeraTK OpenEmbedded layer provides libs, tools and GenericServer

Yocto + OPC UA GenericServer on SoC used at Synchrotron SOLEIL !

What else is new

- ApplicationCore 3: Fully automatic connection of variables
- PCIe user interrupts with xDMA
- Double buffering & bit range logical name map plugins



Summary

DeviceAccess: Flexible, configurable hardware access

- Native C++ with Python, Matlab and command line interface
- Extensible backend mechanism
- Graphical user interface QtHardMon
- Logical name mapping with powerful plugins

ApplicationCore

- Modular, multi-threaded, C++ control applications
- Automatic device error handling and data validity propagation

Control System Adapters

- Native integration into DOOCS, EPICS, OPC UA or TANGO
- Configuration to adapt applications into a facility

Thank you

References

Source repositories

- ChimeraTK <https://github.com/ChimeraTK>
- GenericDeviceServer <https://github.com/ChimeraTK/GenericDeviceServer.git>

Ubuntu 20.04 packages [DESY DOOCS repository](#)

ChimeraTK Yocto Layer <https://github.com/ChimeraTK/meta-chimeratk>

API documentation <https://chimeratk.github.io/>

Backup: Extend server by business logic

ApplicationCore

```
struct LogicExample : public ChimeraTK::ApplicationModule {
    using ctk::ApplicationModule::ApplicationModule;
    ChimeraTK::ArrayPushInput<uint32_t> timingCnt{this, "/ADC_LOGICAL/timing/triggerCnt", "", 4, "hw trigger counters"};
    ChimeraTK::ScalarOutput<std::string> status{this, "status", "(no unit)", "fpga app status"};
    ChimeraTK::ScalarOutput<uint32_t> loopCounter{this, "loopCounter", "", ""};

    void mainLoop() override {
        // device is already initialized and current input values loaded
        uint32_t lastCnt = 0;
        while(true) {
            status = (timingCnt[1] != lastCnt) ? "fpga app running" : "fpga app not running";
            lastCnt = timingCnt[1];
            loopCounter++;
            writeAll();
            readAll();
        }
    }
};

struct ExtendedGenericApp : public ctk::Application {
    ...
    LogicExample sc{this, "logicExample", ""};
};
```

GenericDeviceServer with OPC UA Adapter

Select OPC UA Adapter

```
$ cmake -DADAPTER="OPCUA" -S ../GenericDeviceServer/ . ; make  
$ ./opcua-generic-chimeratk-server01
```

The screenshot displays the Unified Automation UaExpert interface. The left pane shows the 'Address Space' tree with the 'ChimeraTK-GenericServer' expanded, revealing various objects like 'ADC_LOGICAL', 'attenuator', 'board', 'channels', 'timing', 'Config', 'Devices', 'NewFolder', 'triggerCounter', 'logicExample', 'loopCounter', 'status', 'msTimer', 'Server', 'Types', and 'Views'. The 'signal10_doubleBuff' object is highlighted. The right pane shows the 'Data Access View' table with 12 rows of data.

#	Server	Node Id	Display Name	Value	Datatype	Source Timestamp	S
1	OPCUA demo techlab	NS1 String ChimeraTK-GenericServer/ADC_LOGICAL/board/scratch	scratch	42	UInt32	9:32:12.452 AM	9:32:12.452 AM
2	OPCUA demo techlab	NS1 String ChimeraTK-GenericServer/ADC_LOGICAL/board/version	version	33751296	UInt32	9:32:12.452 AM	9:32:12.452 AM
3	OPCUA demo techlab	NS1 String ChimeraTK-GenericServer/ADC_LOGICAL/attenuator/attFactor	attFactor	2	Float	9:32:08.869 AM	9:32:08.869 AM
4	OPCUA demo techlab	NS1 String ChimeraTK-GenericServer/ADC_LOGICAL/attenuator/attReadback	attReadback	20	Int32	9:32:12.452 AM	9:32:12.452 AM
5	OPCUA demo techlab	NS1 String ChimeraTK-GenericServer/ADC_LOGICAL/attenuator/attSetpoint	attSetpoint	10	Int32	9:32:08.869 AM	9:32:08.869 AM
6	OPCUA demo techlab	NS1 String ChimeraTK-GenericServer/ADC_LOGICAL/timing/triggerCnt	triggerCnt	{2014737201,3829273958,2014737203,2014737...	UInt32	10:07:18.953 AM	10:07:18.953 AM
7	OPCUA demo techlab	NS1 String ChimeraTK-GenericServer/msTimer/tick	tick	2109	UInt64	10:07:18.055 AM	10:07:18.055 AM
8	OPCUA demo techlab	NS1 String ChimeraTK-GenericServer/msTimer/period	period	1000	UInt32	9:32:08.869 AM	9:32:08.869 AM
9	OPCUA demo techlab	NS1 String ChimeraTK-GenericServer/logicExample/loopCounter	loopCounter	39181	UInt32	10:07:18.953 AM	10:07:18.953 AM
10	OPCUA demo techlab	NS1 String ChimeraTK-GenericServer/logicExample/status	status	fpga app running	String	10:02:43.499 AM	10:02:43.499 AM
11	OPCUA demo techlab	NS1 String ChimeraTK-GenericServer/ADC_LOGICAL/channels/signal10	signal10	{-23617,-23517,-23417,-23317,-23217,-23117,-2...	Int32	10:07:19.203 AM	10:07:19.203 AM
12	OPCUA demo techlab	NS1 String ChimeraTK-GenericServer/ADC_LOGICAL/channels/signal10_doubleBuff	signal10_doubleBuff	{-20321,-20221,-20121,-20021,-19921,-19821,-1...	Int32	10:07:00.953 AM	10:07:00.953 AM

Control System Adapter for DOOCS

Configuration

doocs-generic-chimeratk-server01.conf
= general configuration as required by DOOCS

- Access control
- Port numbers
- Global history & autosave parameters

```
eq_conf:

oper_uid: -1
oper_gid: 422
xpert_uid: 0
xpert_gid: 0

eq_fct_name: "NODENAME._SVR"
eq_fct_type: 1
{
  SVR.STORE.RATE: 10
  SVR.STORE.AUTO: 4
  SVR.RPC_NUMBER: 610498009
  SVR.FACILITY: "TEST.DOOCS"
}
```

generic_chimeratk_server-DoocsVariableConfig.xml
= DoocsAdapter specific configuration

- Select, rename, reorganize registers into DOOCS properties
- History configuration (per property)
- ...

```
<?xml version="1.0" encoding="UTF-8"?>
<device_server
xmlns="https://github.com/ChimeraTK/ControlSystemAdapter-
DoocsAdapter">
  <location name="ADC_LOGICAL">
    <import>/ADC_LOGICAL</import>
  </location>
  <location name="EverythingElse">
    <import>/</import>
  </location>
</device_server>
```

Extend server by business logic

What functionality do we need for a full Control System Application?

Business Logic

New

Connection Code

ApplicationCore $\Rightarrow$ Automatic

Error handling

Device Init

DeviceAccess & Python scripts

Setpoint persistency

Control system adapters $\Rightarrow$ Automatic / config

History