Rapid Hardware Integration with the ChimeraTK Software Framework

Martin Hierholzer

MicroTCA Workshop 2025, Hamburg,

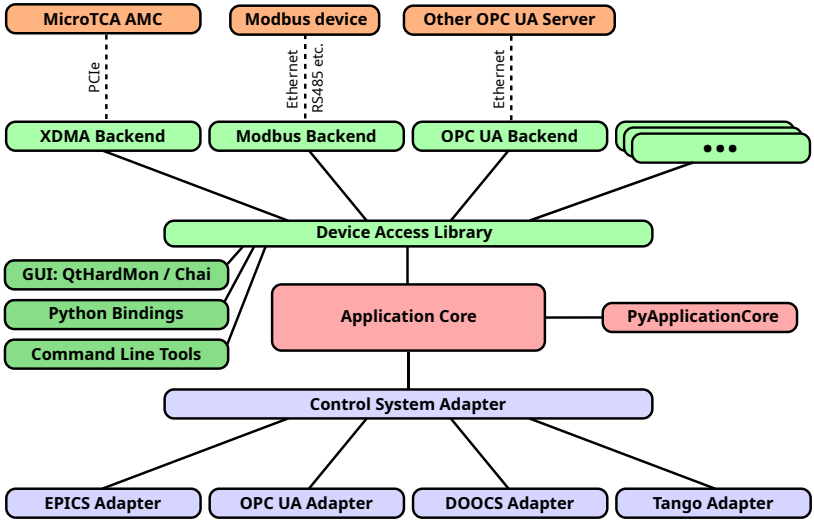
2025-12-02

HELMHOLTZ

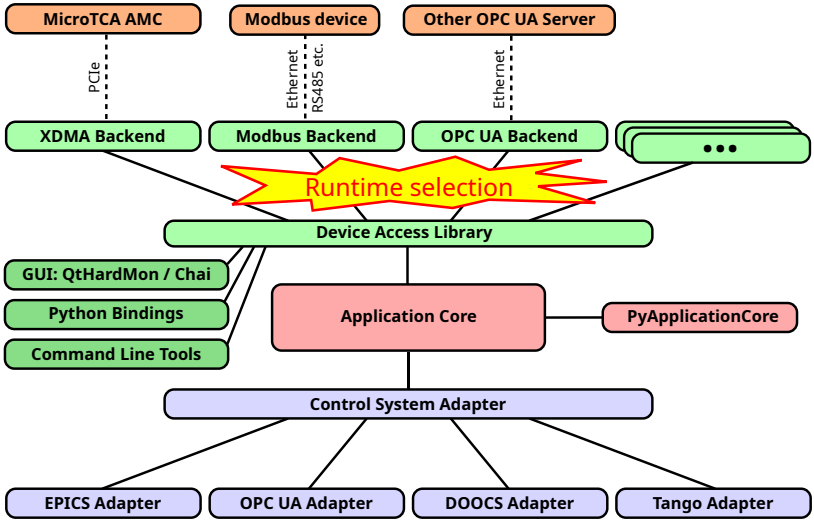
What is ChimeraTK?

Control system and Hardware Interface with Mapped and Extensible Register-based device Abstraction Tool Kit

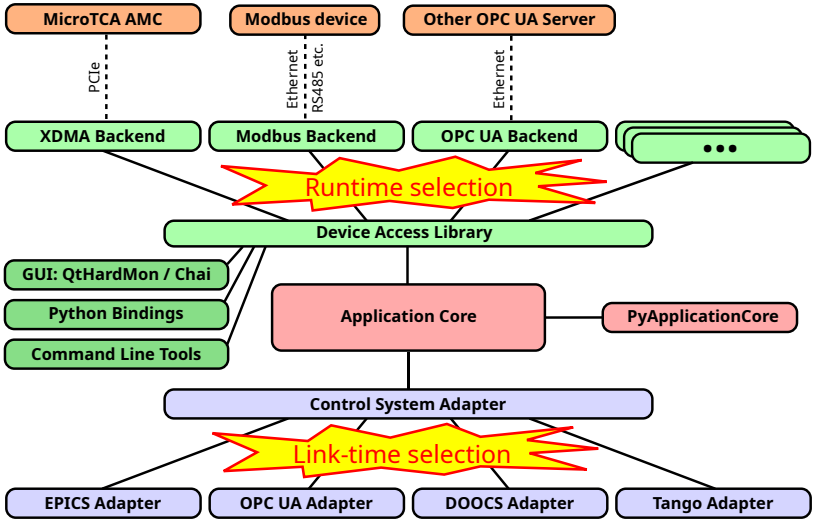
ChimeraTK overview



ChimeraTK overview



ChimeraTK overview



ChimeraTK overview

- ChimeraTK is a set of C++ libraries (with Python bindings) and stand-alone tools
- Can interface with hardware through various protocols via “DeviceBackends”
- Can interface with the rest of your control system through one of the “ControlSystemAdapters”

First contact

Let's talk to our hardware!

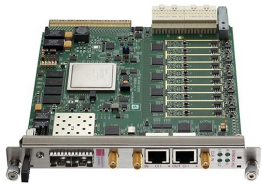
Assumptions

- MicroTCA crate with MCH

Assumptions

- MicroTCA crate with MCH

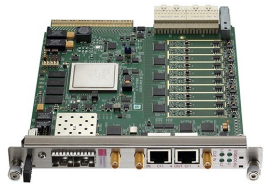
- Some AMC board to talk to



Assumptions

> MicroTCA crate with MCH

> Some AMC board to talk to



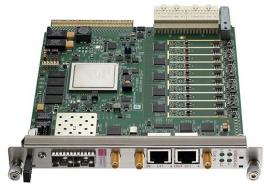
> FPGA firmware

Assumptions

> MicroTCA crate with MCH

> AMC CPU

> Some AMC board to talk to



> FPGA firmware

Assumptions

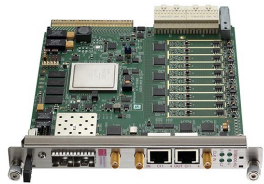
> MicroTCA crate with MCH

> AMC CPU

> Ubuntu 24.04

Ubuntu

> Some AMC board to talk to



> FPGA firmware

FWK
FPGA firmware framework

Assumptions

> AMC CPU

> MicroTCA crate with MCH

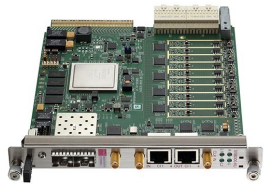
> Ubuntu 24.04

Ubuntu

Only example setup!

> None of these are mandatory requirements!

> Some AMC board to talk to



> FPGA firmware

FWK
FPGA firmware framework

Setup Ubuntu Package Repository

- For Ubuntu 24.04: Use DESY DOOCS package repository

```
wget -O - https://doocs-web.desy.de/pub/doocs/DOOCS-key.gpg.asc \
  | sudo gpg --dearmor -o /etc/apt/trusted.gpg.d/doocs-keyring.gpg
sudo sh -c 'echo "deb https://doocs-web.desy.de/pub/doocs \
  $(lsb_release -cs) main" > /etc/apt/sources.list.d/doocs.list'
sudo apt update
```

- See “DOOCS installation manual” on https://doocs.desy.de
- Install packages:

```
apt install qthardmon chimeratk-chai opcua-generic-chimeratk-server02
```

- May also choose epics-generic-chimeratk-server02, doocs-generic-chimeratk-server02 or tango-generic-chimeratk-server02

What you get from FWK firmware: the map file

Taken from SIS8300KU example design

```
@MAPFILE_REVISION 2.6.0-0-g1d930b9b
@![0,1] {"INTC" : {"path": "DAQ", "version":1} }
@![0,2] {"INTC" : {"path": "RTM", "version":1} }
@![0] {"INTC" : {"path": "APP", "version":1} }
@![1] {"INTC" : {"path": "TIMING", "version":1} }
APP.ISR.TRIGGER                                  0  0x00000000            0    0    0    0    0  INTERRUPT0:0
APP.ISR.DAQ                                      0  0x00000000            0    0    0    0    0  INTERRUPT0:1
DAQ.ISR.START                                    0  0x00000000            0    0    0    0    0  INTERRUPT0:1:0
DAQ.ISR.DONE                                     0  0x00000000            0    0    0    0    0  INTERRUPT0:1:1
=
BSP.ID                                           1  0x00000000            4    0   32    0    0  RO
ch0_top.BSP                                  19201  0x00000000        76804    0   32    0    0  RW
BSP.VERSION                                      1  0x00000004            4    0   32    0    0  RO
BSP.PRJ_ID                                       1  0x00000008            4    0   32    0    0  RO
BSP.PRJ_VERSION                                  1  0x0000000C            4    0   32    0    0  RO
BSP.PRJ_SHASUM                                   1  0x00000010            4    0   32    0    0  RO
BSP.PRJ_TIMESTAMP                                1  0x00000014            4    0   32    0    0  RO
BSP.SCRATCH                                      1  0x00000018            4    0   32    0    0  RW
BSP.RESET_N                                      1  0x0000001C            4    0    1    0    0  RW
BSP.CLK_MUX                                      6  0x00000020           24    0    2    0    0  RW
BSP.CLK_SEL                                      1  0x00000038            4    0    1    0    0  RW
BSP.CLK_RST                                      1  0x0000003C            4    0    1    0    0  RW
BSP.CLK_FREQ                                     4  0x00000040           16    0   32    0    0  RO
BSP.CLK_ERR                                      1  0x00000050            4    0    2    0    0  RO
```

What you get from FWK firmware: the map file

Taken from SIS8300KU example design

```
@MAPFILE_REVISION 2.6.0-0-g1d930b9b
@![0,1] {"INTC" : {"path": "DAQ", "version":1} }
@![0,2] {"INTC" : {"path": "RTM", "version":1} }
@![0] {"INTC" : {"path": "APP", "version":1} }
@![1] {"INTC" : {"path": "TIMING", "version":1} }
```

metadata

```
APP.ISR.TRIGGER                 0  0x00000000          0  0   0  0  0  INTERRUPT0:0
APP.ISR.DAQ                     0  0x00000000          0  0   0  0  0  INTERRUPT0:1
DAQ.ISR.START                   0  0x00000000          0  0   0  0  0  INTERRUPT0:1:0
DAQ.ISR.DONE                    0  0x00000000          0  0   0  0  0  INTERRUPT0:1:1
//
BSP.ID                          1  0x00000000          4  0  32  0  0  RO
ch0_top.BSP                 19201  0x00000000      76804  0  32  0  0  RW
BSP.VERSION                     1  0x00000004          4  0  32  0  0  RO
BSP.PRJ_ID                      1  0x00000008          4  0  32  0  0  RO
BSP.PRJ_VERSION                 1  0x0000000C          4  0  32  0  0  RO
BSP.PRJ_SHASUM                  1  0x00000010          4  0  32  0  0  RO
BSP.PRJ_TIMESTAMP               1  0x00000014          4  0  32  0  0  RO
BSP.SCRATCH                     1  0x00000018          4  0  32  0  0  RW
BSP.RESET_N                     1  0x0000001C          4  0   1  0  0  RW
BSP.CLK_MUX                     6  0x00000020         24  0   2  0  0  RW
BSP.CLK_SEL                     1  0x00000038          4  0   1  0  0  RW
BSP.CLK_RST                     1  0x0000003C          4  0   1  0  0  RW
BSP.CLK_FREQ                    4  0x00000040         16  0  32  0  0  RO
BSP.CLK_ERR                     1  0x00000050          4  0   2  0  0  RO
```

What you get from FWK firmware: the map file

Taken from SIS8300KU example design

```
@MAPFILE REVISION 2.6.0-0-g1d930b9b                                    metadata
@![0,1] {"INTC" : {"path": "DAQ", "version":1} }                                                    interrupt definition
@![0,2] {"INTC" : {"path": "RTM", "version":1} }
@![0] {"INTC" : {"path": "APP", "version":1} }
@![1] {"INTC" : {"path": "TIMING", "version":1} }
APP.ISR.TRIGGER                    0  0x00000000          0  0   0  0  0  INTERRUPT0:0
APP.ISR.DAQ                        0  0x00000000          0  0   0  0  0  INTERRUPT0:1
DAQ.ISR.START                      0  0x00000000          0  0   0  0  0  INTERRUPT0:1:0
DAQ.ISR.DONE                       0  0x00000000          0  0   0  0  0  INTERRUPT0:1:1
//
BSP.ID                             1  0x00000000          4  0  32  0  0  RO
ch0_top.BSP                    19201  0x00000000      76804  0  32  0  0  RW
BSP.VERSION                        1  0x00000004          4  0  32  0  0  RO
BSP.PRJ_ID                         1  0x00000008          4  0  32  0  0  RO
BSP.PRJ_VERSION                    1  0x0000000C          4  0  32  0  0  RO
BSP.PRJ_SHASUM                     1  0x00000010          4  0  32  0  0  RO
BSP.PRJ_TIMESTAMP                  1  0x00000014          4  0  32  0  0  RO
BSP.SCRATCH                        1  0x00000018          4  0  32  0  0  RW
BSP.RESET_N                        1  0x0000001C          4  0   1  0  0  RW
BSP.CLK_MUX                        6  0x00000020         24  0   2  0  0  RW
BSP.CLK_SEL                        1  0x00000038          4  0   1  0  0  RW
BSP.CLK_RST                        1  0x0000003C          4  0   1  0  0  RW
BSP.CLK_FREQ                       4  0x00000040         16  0  32  0  0  RO
BSP.CLK_ERR                        1  0x00000050          4  0   2  0  0  RO
```

What you get from FWK firmware: the map file

Taken from SIS8300KU example design

```
@MAPFILE REVISION 2.6.0-0-g1d930b9b                                    metadata
@![0,1] {"INTC" : {"path": "DAQ", "version":1} }                                                    interrupt definition
@![0,2] {"INTC" : {"path": "RTM", "version":1} }
@![0] {"INTC" : {"path": "APP", "version":1} }
@![1] {"INTC" : {"path": "TIMING", "version":1} }
APP.ISR.TRIGGER                    0  0x00000000          0  0  0  0  0  INTERRUPT0:0
APP.ISR.DAQ                        0  0x00000000          0  0  0  0  0  INTERRUPT0:1
DAQ.ISR.START                      0  0x00000000          0  0  0  0  0  INTERRUPT0:1:0
DAQ.ISR.DONE                       0  0x00000000          0  0  0  0  0  INTERRUPT0:1:1
//
BSP.ID                             1  0x00000000          4  0 32  0  0  RO
ch0_top.BSP                    19201  0x00000000      76804  0 32  0  0  RW
BSP.VERSION                        1  0x00000004          4  0 32  0  0  RO
BSP.PRJ_ID                         1  0x00000008          4  0 32  0  0  RO
BSP.PRJ_VERSION                    1  0x0000000C          4  0 32  0  0  RO
BSP.PRJ_SHASUM                     1  0x00000010          4  0 32  0  0  RO
BSP.PRJ_TIMESTAMP                  1  0x00000014          4  0 32  0  0  RO
BSP.SCRATCH                        1  0x00000018          4  0 32  0  0  RW
BSP.RESET_N                        1  0x0000001C          4  0  1  0  0  RW
BSP.CLK_MUX                        6  0x00000020         24  0  2  0  0  RW
BSP.CLK_SEL                        1  0x00000038          4  0  1  0  0  RW
BSP.CLK_RST                        1  0x0000003C          4  0  1  0  0  RW
BSP.CLK_FREQ                       4  0x00000040         16  0 32  0  0  RO
BSP.CLK_ERR                        1  0x00000050          4  0  2  0  0  RO
                                                                register name
```

What you get from FWK firmware: the map file

Taken from SIS8300KU example design

```
@MAPFILE REVISION 2.6.0-0-g1d930b9b
@![0,1] {"INTC" : {"path": "DAQ", "version":1} }
@![0,2] {"INTC" : {"path": "RTM", "version":1} }
@![0] {"INTC" : {"path": "APP", "version":1} }
@![1] {"INTC" : {"path": "TIMING", "version":1} }
```

metadata

interrupt definition

APP.ISR.TRIGGER	0	0x00000000	0	0	0	0	0	INTERRUPT0:0
APP.ISR.DAQ	0	0x00000000	0	0	0	0	0	INTERRUPT0:1
DAQ.ISR.START	0	0x00000000	0	0	0	0	0	INTERRUPT0:1:0
DAQ.ISR.DONE	0	0x00000000	0	0	0	0	0	INTERRUPT0:1:1

register name	nb. of elements		nb. of bytes					
BSP.ID	1	0x00000000	4	0	32	0	0	RO
ch0_top.BSP	19201	0x00000000	76804	0	32	0	0	RW
BSP.VERSION	1	0x00000004	4	0	32	0	0	RO
BSP.PRJ_ID	1	0x00000008	4	0	32	0	0	RO
BSP.PRJ_VERSION	1	0x0000000C	4	0	32	0	0	RO
BSP.PRJ_SHASUM	1	0x00000010	4	0	32	0	0	RO
BSP.PRJ_TIMESTAMP	1	0x00000014	4	0	32	0	0	RO
BSP.SCRATCH	1	0x00000018	4	0	32	0	0	RW
BSP.RESET_N	1	0x0000001C	4	0	1	0	0	RW
BSP.CLK_MUX	6	0x00000020	24	0	2	0	0	RW
BSP.CLK_SEL	1	0x00000038	4	0	1	0	0	RW
BSP.CLK_RST	1	0x0000003C	4	0	1	0	0	RW
BSP.CLK_FREQ	4	0x00000040	16	0	32	0	0	RO
BSP.CLK_ERR	1	0x00000050	4	0	2	0	0	RO

What you get from FWK firmware: the map file

Taken from SIS8300KU example design

```
@MAPFILE REVISION 2.6.0-0-g1d930b9b                                    metadata
@![0,1] {"INTC" : {"path": "DAQ", "version":1} }                       interrupt definition
@![0,2] {"INTC" : {"path": "RTM", "version":1} }
@![0] {"INTC" : {"path": "APP", "version":1} }
@![1] {"INTC" : {"path": "TIMING", "version":1} }
APP.ISR.TRIGGER                    0 0x00000000          0   0   0   0   0 INTERRUPT0:0
APP.ISR.DAQ                        0 0x00000000          0   0   0   0   0 INTERRUPT0:1
DAQ.ISR.START                      0 0x00000000          0   0   0   0   0 INTERRUPT0:1:0
DAQ.ISR.DONE                       0 0x00000000          0   0   0   0   0 INTERRUPT0:1:1
//
BSP.ID                             1 0x00000000          4   0  32   0   0 RO
ch0_top.BSP                    19201 0x00000000      76804   0  32   0   0 RW
BSP.VERSION                        1 0x00000004          4   0  32   0   0 RO
BSP.PRJ_ID                         1 0x00000008          4   0  32   0   0 RO
BSP.PRJ_VERSION                    1 0x0000000C          4   0  32   0   0 RO
BSP.PRJ_SHASUM                     1 0x00000010          4   0  32   0   0 RO
BSP.PRJ_TIMESTAMP                  1 0x00000014          4   0  32   0   0 RO
BSP.SCRATCH                        1 0x00000018          4   0  32   0   0 RW
BSP.RESET_N                        1 0x0000001C          4   0   1   0   0 RW
BSP.CLK_MUX                        6 0x00000020         24   0   2   0   0 RW
BSP.CLK_SEL                        1 0x00000038          4   0   1   0   0 RW
BSP.CLK_RST                        1 0x0000003C          4   0   1   0   0 RW
BSP.CLK_FREQ                       4 0x00000040         16   0  32   0   0 RO
BSP.CLK_ERR                        1 0x00000050          4   0   2   0   0 RO
                  register name  nb. of elements  address  nb. of bytes  BAR
```

What you get from FWK firmware: the map file

Taken from SIS8300KU example design

```
@MAPFILE REVISION 2.6.0-0-g1d930b9b                                                                                    metadata
@![0,1] {"INTC" : {"path": "DAQ", "version":1} }                                                                       interrupt definition
@![0,2] {"INTC" : {"path": "RTM", "version":1} }
@![0] {"INTC" : {"path": "APP", "version":1} }
@![1] {"INTC" : {"path": "TIMING", "version":1} }
APP.ISR.TRIGGER                                0 0x00000000          0   0   0   0   0 INTERRUPT0:0
APP.ISR.DAQ                                    0 0x00000000          0   0   0   0   0 INTERRUPT0:1
DAQ.ISR.START                                  0 0x00000000          0   0   0   0   0 INTERRUPT0:1:0
DAQ.ISR.DONE                                   0 0x00000000          0   0   0   0   0 INTERRUPT0:1:1
BSP.ID                                         1 0x00000000          4   0  32   0   0 RO
ch0_top.BSP                                19201 0x00000000      76804   0  32   0   0 RW
BSP.VERSION                                    1 0x00000004          4   0  32   0   0 RO
BSP.PRJ_ID                                     1 0x00000008          4   0  32   0   0 RO
BSP.PRJ_VERSION                                1 0x0000000C          4   0  32   0   0 RO
BSP.PRJ_SHASUM                                 1 0x00000010          4   0  32   0   0 RO
BSP.PRJ_TIMESTAMP                              1 0x00000014          4   0  32   0   0 RO
BSP.SCRATCH                                    1 0x00000018          4   0  32   0   0 RW
BSP.RESET_N                                    1 0x0000001C          4   0   1   0   0 RW
BSP.CLK_MUX                                    6 0x00000020         24   0   2   0   0 RW
BSP.CLK_SEL                                    1 0x00000038          4   0   1   0   0 RW
BSP.CLK_RST                                    1 0x0000003C          4   0   1   0   0 RW
BSP.CLK_FREQ                                   4 0x00000040         16   0  32   0   0 RO
BSP.CLK_ERR                                    1 0x00000050          4   0   2   0   0 RO
```

register name | nb. of elements | address | nb. of bytes | BAR | fixed point info

What you get from FWK firmware: the map file

Taken from SIS8300KU example design

```
@MAPFILE REVISION 2.6.0-0-g1d930b9b
@![0,1] {"INTC" : {"path": "DAQ", "version":1} }
@![0,2] {"INTC" : {"path": "RTM", "version":1} }
@![0] {"INTC" : {"path": "APP", "version":1} }
@![1] {"INTC" : {"path": "TIMING", "version":1} }
```

metadata

interrupt definition

APP.ISR.TRIGGER	0	0x00000000	0	0	0	0	0	INTERRUPT0:0
APP.ISR.DAQ	0	0x00000000	0	0	0	0	0	INTERRUPT0:1
DAQ.ISR.START	0	0x00000000	0	0	0	0	0	INTERRUPT0:1:0
DAQ.ISR.DONE	0	0x00000000	0	0	0	0	0	INTERRUPT0:1:1

register name	nb. of elements	address	nb. of bytes	BAR	fixed point info			access
BSP.ID	1	0x00000000	4	0	32	0	0	RO
ch0_top.BSP	19201	0x00000000	76804	0	32	0	0	RW
BSP.VERSION	1	0x00000004	4	0	32	0	0	RO
BSP.PRJ_ID	1	0x00000008	4	0	32	0	0	RO
BSP.PRJ_VERSION	1	0x0000000C	4	0	32	0	0	RO
BSP.PRJ_SHASUM	1	0x00000010	4	0	32	0	0	RO
BSP.PRJ_TIMESTAMP	1	0x00000014	4	0	32	0	0	RO
BSP.SCRATCH	1	0x00000018	4	0	32	0	0	RW
BSP.RESET_N	1	0x0000001C	4	0	1	0	0	RW
BSP.CLK_MUX	6	0x00000020	24	0	2	0	0	RW
BSP.CLK_SEL	1	0x00000038	4	0	1	0	0	RW
BSP.CLK_RST	1	0x0000003C	4	0	1	0	0	RW
BSP.CLK_FREQ	4	0x00000040	16	0	32	0	0	RO
BSP.CLK_ERR	1	0x00000050	4	0	2	0	0	RO

Define your devices

... by writing a DMAP file

```
MyAMC    (xdma:xdma/slot6?map=sis8300ku_example_design_dwc10_2.6.0-0-g1d930b9b.mapp)
Slot8    (xdma:xdma/slot8?map=sis8300ku_example_design_dwc10_2.6.0-0-g1d930b9b.mapp)
Slot7    (xdma:xdma/slot7?map=some_other_map_file.mapp)
OPCUADev (opcua:127.0.0.1?port=16664)
```

Define your devices

... by writing a DMAP file

MyAMC	(xdma:xdma/slot6?map=sis8300ku_example_design_dwc10_2.6.0-0-g1d930b9b.mapp)
Slot8	(xdma:xdma/slot8?map=sis8300ku_example_design_dwc10_2.6.0-0-g1d930b9b.mapp)
Slot7	(xdma:xdma/slot7?map=some_other_map_file.mapp)
OPCUADev	(opcua:127.0.0.1?port=16664)

alias name

Define your devices

... by writing a DMAP file

alias name	ChimeraTK Device Descriptor
MyAMC	(xdma:xdma/slot6?map=sis8300ku_example_design_dwc10_2.6.0-0-g1d930b9b.mapp)
Slot8	(xdma:xdma/slot8?map=sis8300ku_example_design_dwc10_2.6.0-0-g1d930b9b.mapp)
Slot7	(xdma:xdma/slot7?map=some_other_map_file.mapp)
OPCUADev	(opcua:127.0.0.1?port=16664)

- ChimeraTK Device Descriptor (CDD):
  - `(backend_type:address?key1=value1&key2=value2)`
  - CDD can be part of another CDD as a parameter value

Define your devices

... by writing a DMAP file

MyAMC	(xdma:xdma/slot6?map=sis8300ku_example_design_dwc10_2.6.0-0-g1d930b9b.mapp)
Slot8	(xdma:xdma/slot8?map=sis8300ku_example_design_dwc10_2.6.0-0-g1d930b9b.mapp)
Slot7	(xdma:xdma/slot7?map=some_other_map_file.mapp)
OPCUADev	(opcua:127.0.0.1?port=16664)

map file names

alias name

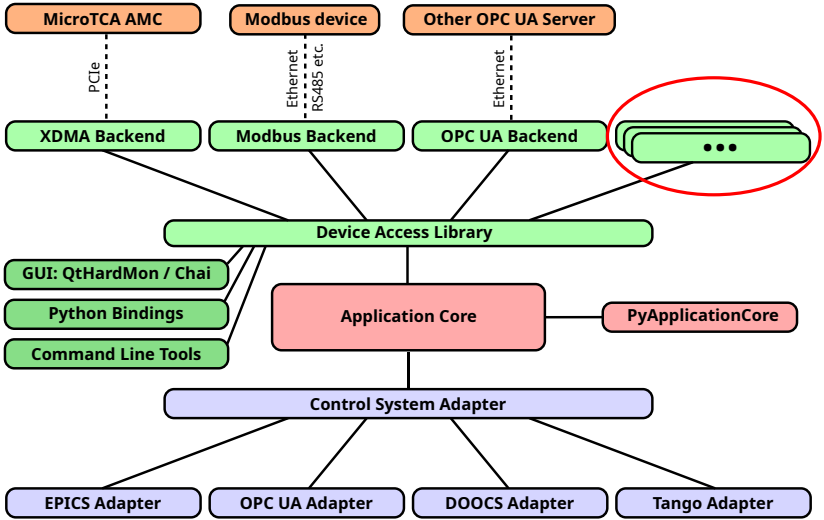
ChimeraTK Device Descriptor

- ChimeraTK Device Descriptor (CDD):
  - (backend_type:address?key1=value1&key2=value2)
  - CDD can be part of another CDD as a parameter value

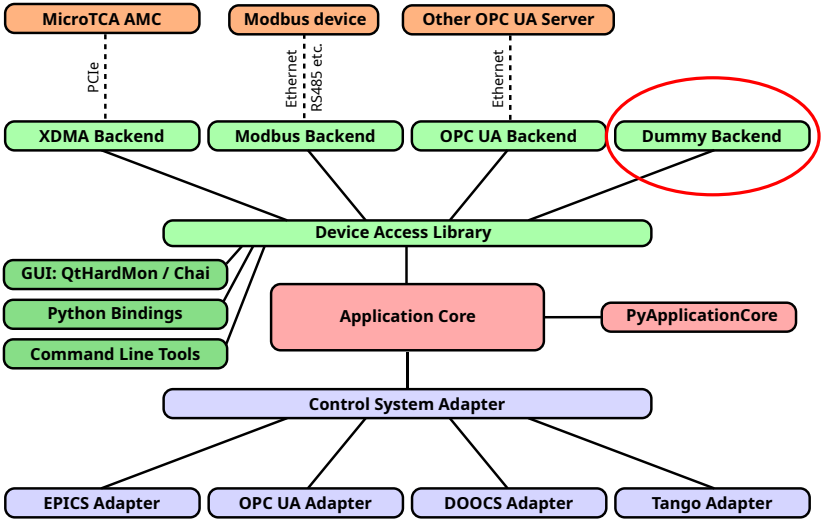
Live demo

- Show map file used throughout the demo
- QtHardMon and Chai
- Show dmap file

Dummy backend



Dummy backend

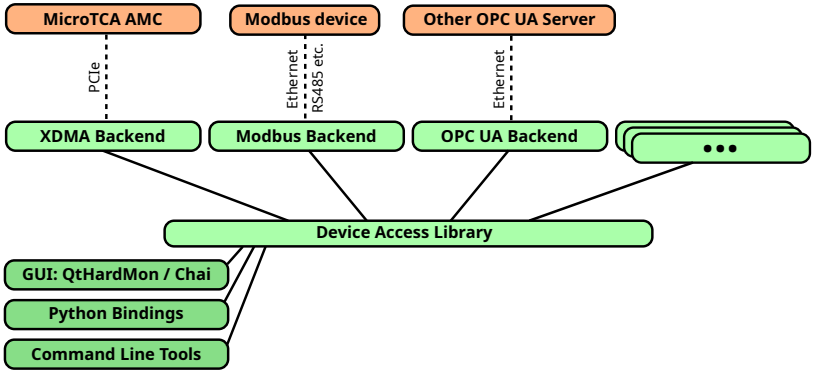


Live demo

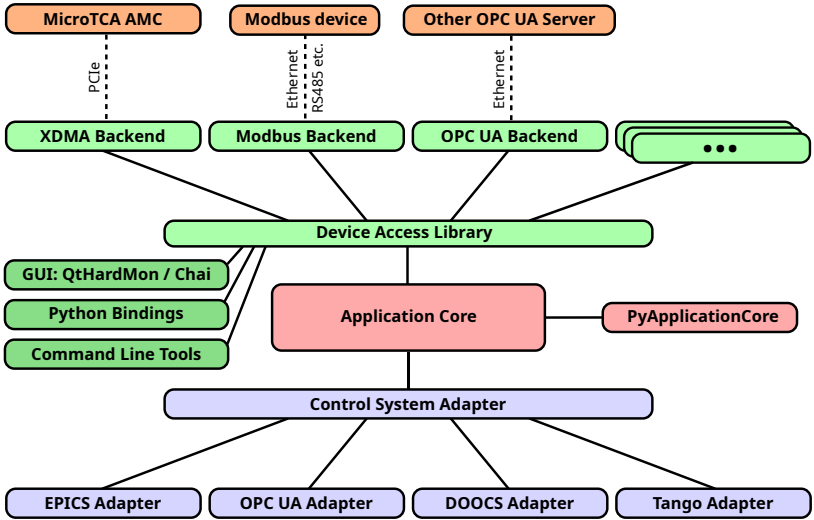
> Simulate hardware/firmware with second Chai

Integrate with your control system
using the GenericDeviceServer

ApplicationCore and the GenericDeviceServer



ApplicationCore and the GenericDeviceServer



ApplicationCore and the GenericDeviceServer

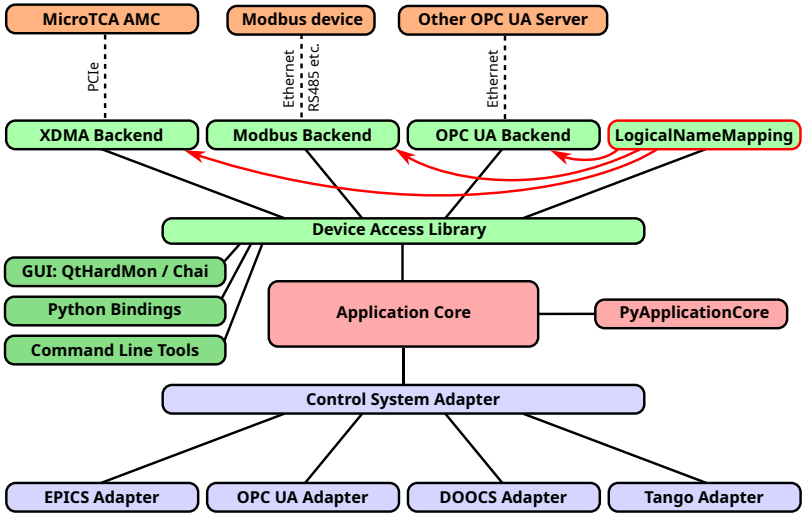
- GenericDeviceServer is ready-to-use ApplicationCore executable
- Linked against any of the adapter (EPICS, OPC UA, DOOCS, Tango)
- Will expose registers from devices to control system
- Configuration: which devices to expose and how
- No C++ or Python code needs to be written!

Live demo

- GenericDeviceServer configuration
- “Replace” first Chai with GenericDeviceServer
- Introduce logical name mapping

Logical Name Mapping backend

A meta-backend targeting any backend type

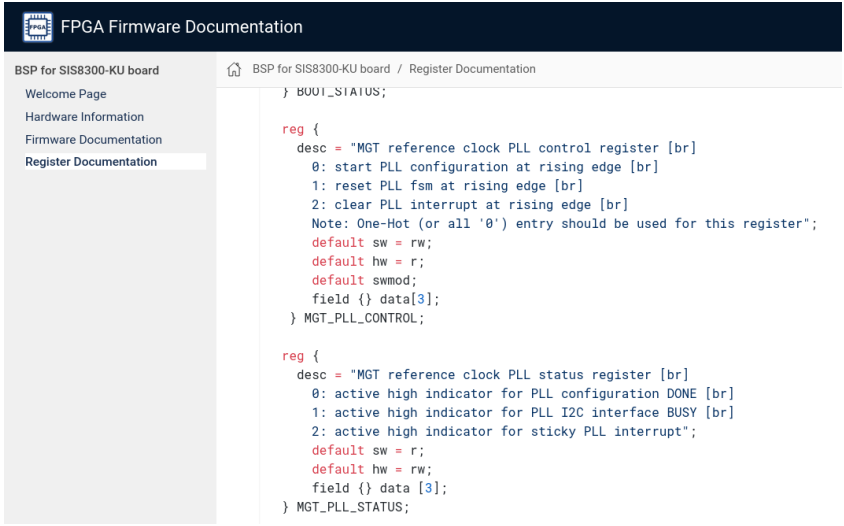


Logical name mapping

- Filter out registers not to be exposed
- Rename registers
- Extract array elements
- Extract individual bits and bit ranges
- Apply (simple) math transformations
- ... and many more things!

Documentation:
https://chimeratk.github.io/ChimeraTK-DeviceAccess/head/html/lmap.html.

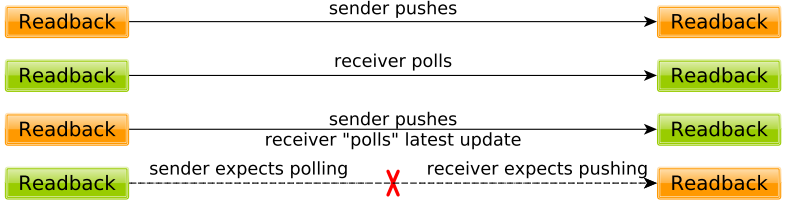
Common task: expose bits of bit field



Live demo

- Logical name mapping:
  - Single bit access
  - Math plugin

Triggering scheme and interrupts



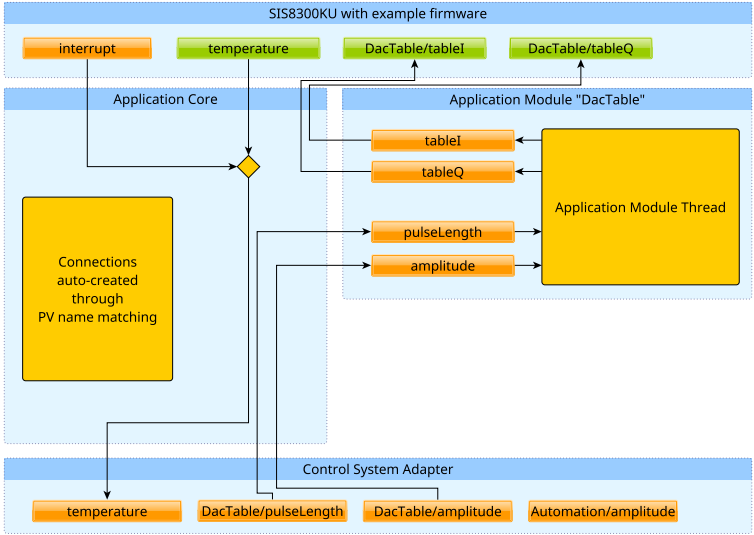
- Normal registers are passive:
  - read: poll-type
  - write: push-type
- Hardware interrupt: push-type (but no data: void-type)
- Map file can specify registers to poll on interrupt
- Interrupt can be used as readout trigger

Live demo

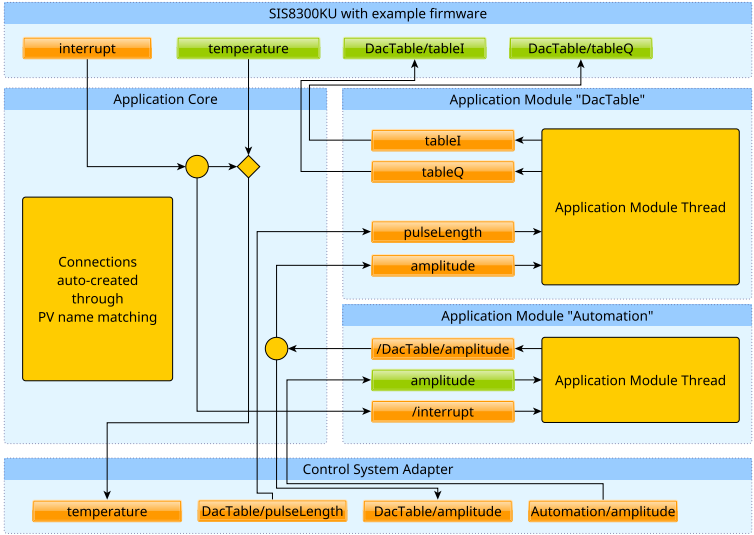
- Interrupts as push-type variables
- Interrupts as trigger for polling

Adding complex application logic
with PyApplicationCore

ApplicationCore modularity



ApplicationCore modularity

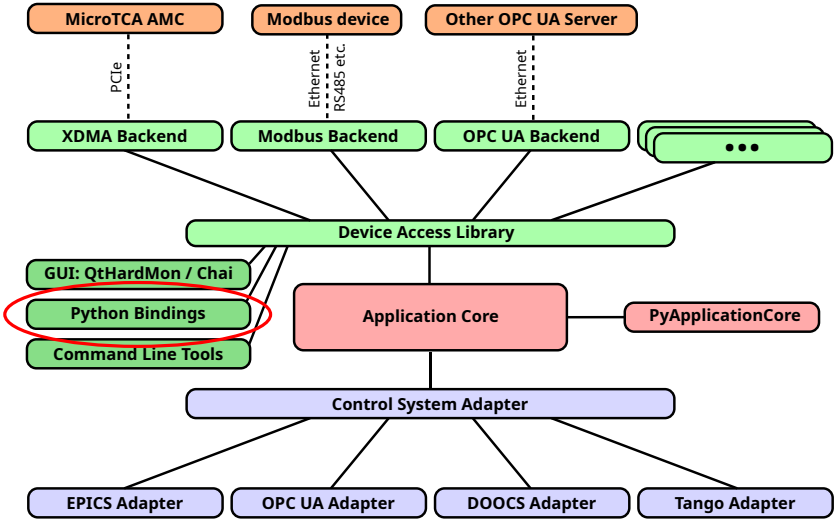


Live demo

- Source code of Python module
- Inject errors:
  - Python exception (use undefined variable)
  - Data type mismatch
  - PV name mismatch
- Add second Python module (modules shall be small, so use many!)

Device initialisation

DeviceAccess Python bindings



Live demo

- Python script with *DeviceAccess* Python bindings
- Add script to GenericDeviceServer configuration

Summary and Close-out

Things we did NOT talk about

- Control system specific configuration:
  - Each ControlSystemAdapter (OPC UA, EPICS, TANGO, DOOCS) has specific config file format
  - Allows to rename variables (e.g. to re-use application in different context)
  - Control specific features like history, special data types etc.
- Target non-MicroTCA hardware with other backends:
  - available for many industry standards like OPC UA, Modbus, SCPI
- Integrate SPI/I2C/... attached hardware with SubDeviceBackend
  - will have its own map file
  - several protocols supported (register layout in primary map file from FWK)
- Write ApplicationModules in C++
  - very similar to PyApplicationCore
  - recommended for maximum stability

Summary

- ChimeraTK framework for hardware access and control system interface
- GenericDeviceServer publishes device to control system
- Dummy backend for easy development on desktop PC
- Logical name mapping to select what to publish and how
- PyApplicationCore: implement application logic in Python
  - Modular approach: use many small ApplicationModules
  - Connect to registers and control system variables by (fully-qualified) name
- DeviceAccess Python bindings: device initialisation

Thank you!

- Live demo files: `https://github.com/mhier/ApplicationCoreLiveDemo`
- ChimeraTK documentation: `https://chimeratk.github.io`
- FWK documentation: `https://fpgafw.pages.desy.de/docs-pub/fwk/main`

Contact

Deutsches Elektronen-
Synchrotron DESY

www.desy.de

Martin Hierholzer
0000-0003-2285-1319
MSK Maschine Strahlkontrollen
martin.hierholzer@desy.de
+49-40-8998-1897