

MTCA4U — The DESY MicroTCA.4 User Tool Kit.

Generic software and drivers for MicroTCA.4 based controls

Martin Killenberg



M. Heuer, L. Petrosyan, C. Schmidt, G. Varghese, *DESY, Hamburg, Germany*

S. Marsching, *aquenos GmbH, Baden-Baden, Germany*

J. Krašna, M. Mehle, T. Sušnik, K. Žagar, *Cosylab d.d., Ljubljana, Slovenia*

A. Piotrowski, *FastLogic Sp. z o.o., Łódź, Poland*

T. Kozak, P. Prędko, J. Wychowaniak, *Łódź University of Technology, Łódź, Poland*

Goal

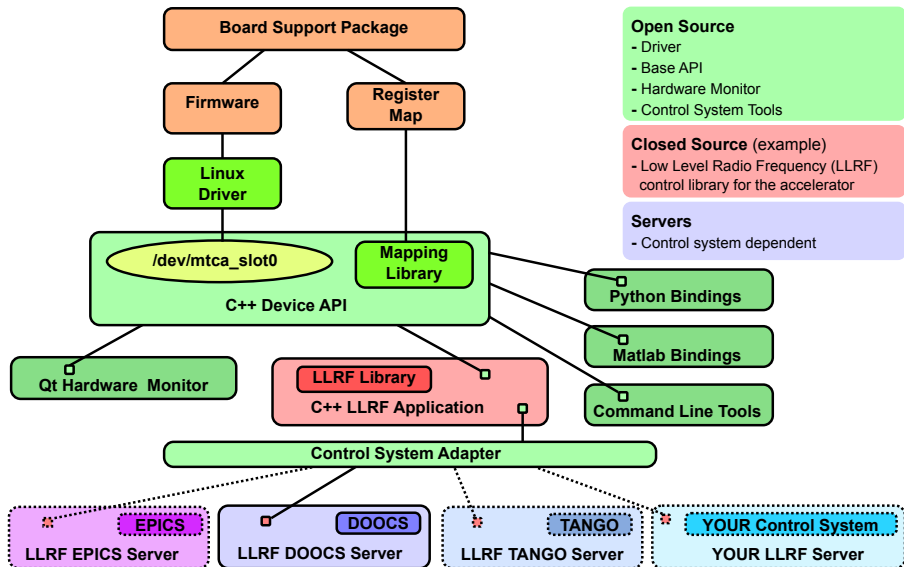
Provide a tool kit to facilitate the development for MicroTCA.4 based control applications.

MTCA4U comprises

- Linux drivers for PCIeexpress
- Intuitive C++ API
- Tools for easy integration into control systems
- Board-specific classes for implementations used at DESY

Requirements

- Independent from the control system
- Universal and extensible
- Base version open source (compile on many distributions)
- Board-specific classes can be closed source (protection of intellectual property)



Firmware

- Standard Register Set at DESY
⇒ All boards can use the same driver
- PICMG is working on Standard Hardware API
(see talk by Till Straumann, Wednesday 11:15 h)
- Firmware board support package automatically generates a register map

Goal

Make all firmware compliant to the PICMG Standard Hardware API.

PCIexpress driver

- Common driver for all boards
(see talk by Ludwig Petrosyan, Thursday 9:30 h)
- The drivers are part of MTCA4U

Basic C++ API

- Classes for convenient read/write via PCIeexpress
- Interface for Direct Memory Access (no need to bother with driver implementation details)
- Register name mapping

Goal

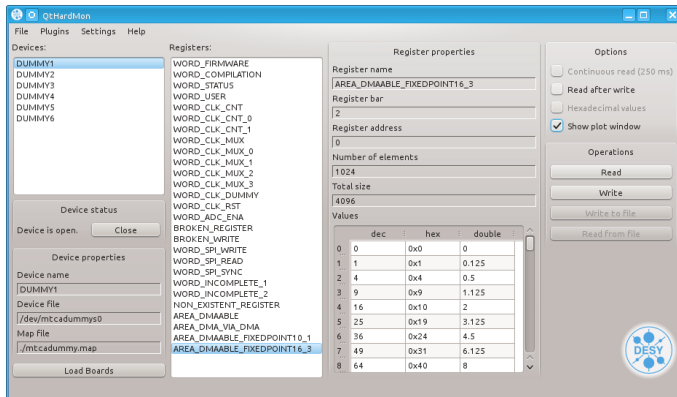
Make MTCA4U a reference implementation of the PICMG Standard Device Model (see talk by Till Straumann, Wednesday 11:15 h).

Register Name Mapping

- Map is automatically generated by the firmware board support package
 - Register name
 - Address
 - Size
 - Data type

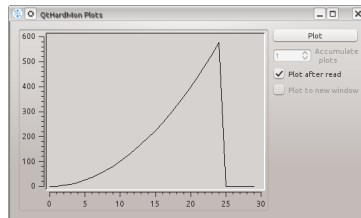
Advantages:

- Use descriptive names instead of hex-addresses
- Better code readability
- User code becomes independent from firmware version
- Automated type conversion



	dec	hex	double
0	0	0x0	0
1	1	0x1	0.125
2	4	0x4	0.5
3	9	0x9	1.125
4	16	0x10	2
5	25	0x19	3.125
6	36	0x24	4.5
7	49	0x31	6.125
8	64	0x40	8

- Display devices and registers by name
- Show and modify register content
- Basic plotting functionality



Task

Complex control algorithms should be used with different control systems.

Contradicting requirements

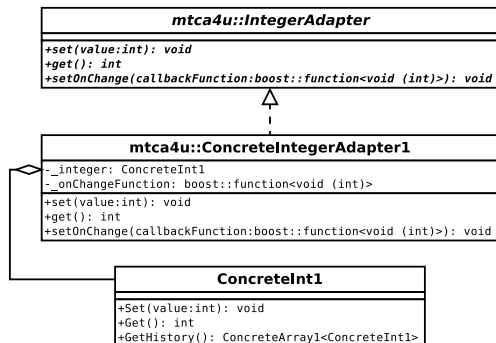
- Keep application code control system independent
 - The algorithm must interact with the control system
 - Do not re-implement functionality provided by the control system
- ⇒ Keep the layer as thin as possible

Additional requirements:

- Thread-safety
- Real-time capability
- Must not copy large data objects (arrays)

Control System Adapter

- Process variables to transfer data to/from the control system
- Callback mechanism to perform actions



- Wrapper classes for
 - simple data types
 - arrays of simple data types
- Contains instance of control system variable
- Only basic interface
 - Get function
 - Set function
 - Callback function on change
 - **No control system functionality!**

Adapters

- DOOCS (in preparation)
- EPICS (in preparation)
- OPC-UA (planned)

Command line tools

- Query devices (list registers)
- Read/write incl. register mapping
- First version is released

Matlab bindings

- Directly use MicroTCA.4 devices inside of Matlab
- Uses the C++ library when running on the front end CPU
- Can tunnel to a remote host via ssh, using the command line tools

Python bindings

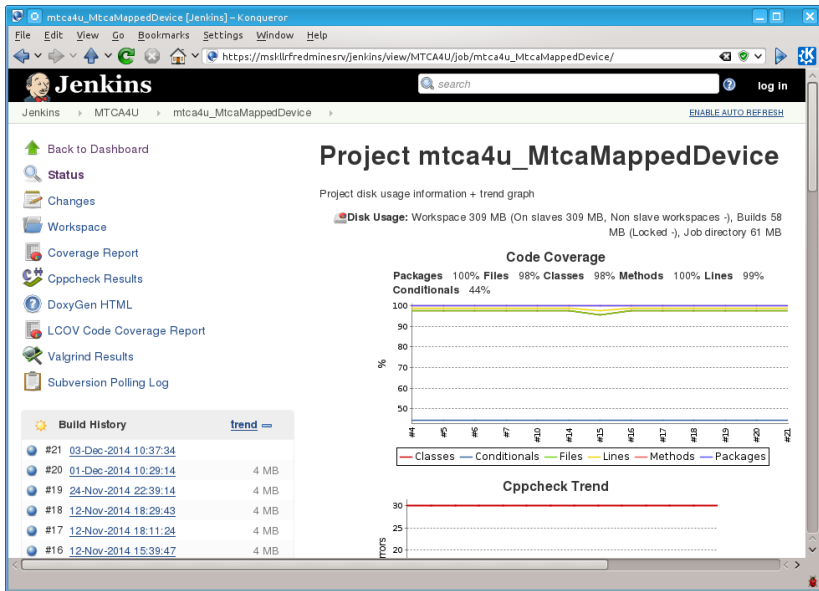
- Use the C++ library from python
- Work has just started

Test suite

- Unit tests with very high code coverage (99 %)
- Dummy driver to test the I/O classes
 - Simulates PCIe registers in the Linux kernel memory
- Dummy device for writing mock classes
 - Loads the mapping file
 - Simulates all registers in user space memory
 - Register callback functions to inject functionality
- Planned: Reference firmware to unit-test the driver

Continuous integration tests

- Check out every subversion commit
- Compile, install and run tests
- Send email in case of errors



Engineering versions for

- Board support package and firmware
- Universal PCIexpress driver
- C++ I/O class
- Register name mapping and **type conversion**
- Hardware monitor GUI
- **Command line tools and Matlab bindings**
- **Quality control (unit tests and continuous integration)**

Under development

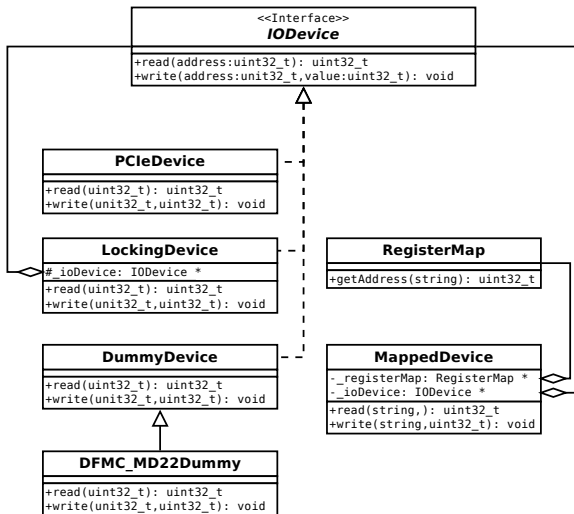
- Control system adapter
- Python bindings

MTCA4U is published under the GNU General Public License.

Subversion repository on the DESY svn server:

<https://svnsrv.desy.de/public/mtca4u/>

Backup



Modern, object oriented design

- Easy to use interfaces
- Multiple abstraction layers, adapted to the different use cases
 - Normal operation
 - Calibration/setup
 - Expert

Unit testing framework

- Well tested code
- Facilitates refactoring
- Dummy devices for software development without hardware access
- Code coverage

Doxygen documentation

- Complete, browsable API documentation

Three types of actions in the control system adapter

- 1 Synchronous actions on set/get
 - Only one process variable is updated
 - Callback function registered with each process variable
 - 2 Update functions triggered by the control system
 - Periodic updates
 - Control system triggers
 - All process variables are updated
 - Callback function registered with the control system adapter
 - 3 Synchronisation triggered by the business logic
 - Control loop running in its own thread
 - Business logic determines when to synchronise
 - All process variables are updated
 - Callback function registered with the control system adapter
- The control system adapter assures thread safety for all process variables inside the callback functions.
 - All callback functions have a control system independent signature.

DFMC-MD22 Motor Controller: mtca4u::MotorDriverCard Class Reference – Konqueror

File Edit View Go Bookmarks Settings Window Help

/space/killenb/MotorDriverCard_trunk/doc/html/classmtca4u_1_1_motor_driver_card.html

mtca4u > MotorDriverCard >

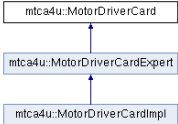
Public Member Functions

mtca4u::MotorDriverCard Class Reference

A class to access the DFMC-MD22 motor driver card, which provides two MotorControllers. More...

```
#include <MotorDriverCard.h>
```

Inheritance diagram for mtca4u::MotorDriverCard:



```

classDiagram
    class mtca4u_MotorDriverCardImpl["mtca4u::MotorDriverCardImpl"]
    class mtca4u_MotorDriverCardExpert["mtca4u::MotorDriverCardExpert"]
    class mtca4u_MotorDriverCard["mtca4u::MotorDriverCard"]
    mtca4u_MotorDriverCardImpl --|> mtca4u_MotorDriverCardExpert
    mtca4u_MotorDriverCardExpert --|> mtca4u_MotorDriverCard
  
```

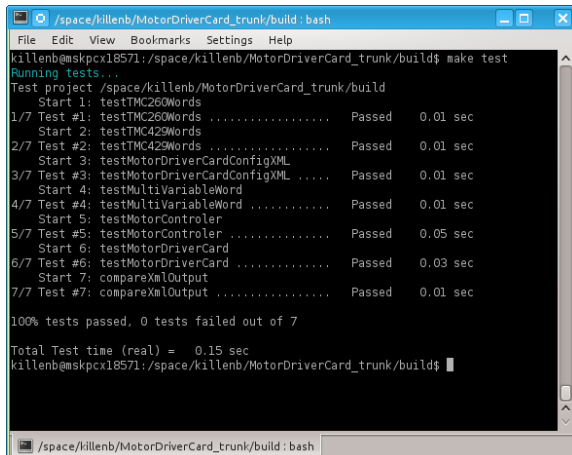
List of all members.

Public Member Functions

virtual unsigned int	getControlerChipVersion ()=0
virtual MotorControler &	getMotorControler (unsigned int motorControlerID)=0 Get acces to one of the two motor controlers on this board.
virtual PowerMonitor &	getPowerMonitor ()=0 Get a reference to the power monitor.
virtual ReferenceSwitchData	getReferenceSwitchData ()=0

Detailed Description

- Tests written using the boost::test library
- Fully integrated into the CMake build system
 - Automatically run when packaging, e.g.
- Used to create code coverage report
 - Goal: Test every single line of code



```
killenb@mskpcx18571:/space/killenb/MotorDriverCard_trunk/build$ make test
Running tests...
Test project /space/killenb/MotorDriverCard_trunk/build
  Start 1: testTMC260Words
1/7 Test #1: testTMC260Words ..... Passed    0.01 sec
  Start 2: testTMC429Words
2/7 Test #2: testTMC429Words ..... Passed    0.01 sec
  Start 3: testMotorDriverCardConfigXML
3/7 Test #3: testMotorDriverCardConfigXML ..... Passed    0.01 sec
  Start 4: testMultiVariableWord
4/7 Test #4: testMultiVariableWord ..... Passed    0.01 sec
  Start 5: testMotorController
5/7 Test #5: testMotorController ..... Passed    0.05 sec
  Start 6: testMotorDriverCard
6/7 Test #6: testMotorDriverCard ..... Passed    0.03 sec
  Start 7: compareXmlOutput
7/7 Test #7: compareXmlOutput ..... Passed    0.01 sec

100% tests passed, 0 tests failed out of 7

Total Test time (real) =  0.15 sec
killenb@mskpcx18571:/space/killenb/MotorDriverCard_trunk/build$
```

