Drivers and Software for MicroTCA.4

M. Killenberg, M. Heuer, M. Hierholzer, L. Petrosyan, C. Schmidt, N. Shehzad, G. Varghese, M. Viti (DESY, Germany), S. Marsching (aquenos GmbH, Germany), M. Mehle, T. Sušnik, K. Žagar (Cosylab, Slovenia), A. Piotrowski (FastLogic, Poland), T. Kozak, P. Prędki, J. Wychowaniak (Łódź University of Technology, Poland)

MicroTCA.4 Technology

Based on Advanced Telecommunications Computing Architecture (ATCA)
Widespread use in telecommunications since 2005
> High speed serial bus topology
> High modularity due to Advanced Mezzanine Cards (AMCs)
> High availability due to redundancy
> Reduced down-time due to hot-swap capability

MicroTCA.4 Enhancements for Rear I/O and Precision Timing
> Definition of Micro Rear Transition Modules (μRTMs)
> Definition of AMC-μRTM connection
> Radial clock lines for precision timing
> Low latency point to point serial I/O
> Advanced shelf management
> High signal integrity by separation of analog and digital processing

Drivers and Software for MicroTCA.4

The DESY MicroTCA.4 User Tool Kit (MTCA4U)

- Board Support Package
- Firmware
- Register Map
- Linux Driver
- C++ Device API
- Mapping Library
- Python Bindings
- Matlab Bindings

Use Case

Low Level Radio Frequency Control at the European XFEL and FLASH
- Superconducting accelerators provide multi-GeV electron beams for Free Electron Lasers (FELs)
- Digital low level radio frequency (LLRF) control based on MicroTCA.4
- Pulsed operation (10 Hz)

The C++ Device API

Device
- DeviceBackend
- RegisterInfoMap
- DeviceLib
- Device
- Register

Device Backend
- Abstract interface
- PCI Express
- Register-based over TCP (Rebot)

Back-End Factory
- Automatically determine the type
- Add new back-ends at run time

Dummy Backend
- Implement firmware mock-ups

Register Name Mapping
- Access registers by name
- Improved code readability
- Robust against firmware changes
- Automatic data type conversion
- Fixed point to floating point

Control System Adapter

Device Library
- Adapter Variable Pair "VOLTAGE"
- Use "VOLTAGE" update "TEMPERATURE"

Control System
- Control System Variable "VOLTAGE"
- Control System Variable "TEMPERATURE"
- Update

Task
- Keep the application code (device library) independent from the control system
- Minimise device-dependent code on the control system side

Requirements
- Thread safety
- Real-time capability
- Do not copy large data objects

This work is supported by the Helmholtz Validation Fund HVF-0016 "MTCA.4 for Industry"