

# LINUX general purpose PCIe driver for MTCA

*Thursday 10 December 2015 17:00 (15 minutes)*

Micro Telecommunications Computing Architecture (MTCA) is the new generation system, which should allow more stable and reliable control of the accelerator facilities such as the FLASH, the European XFEL, the PITZ and etc. Software development for MTCA devices is among the important tasks. It is undeniable that user space software development is preferable to kernel space software development. Therefore it is worthwhile to engage couple of years in the development of kernel space general purpose driver based on MTCA standards. Ultimately only user space software will be required to adopt new MTCA devices. The design and the development of the general driver started in 2013 with the objective of finally creating a driver that is able to handle as many MTCA devices as possible. The following rule is always kept: if a generalization of any functionality leads to penalty in performance or increase in memory usage or too complicated code is abandoned. In the case a device is not possible to be handled by this driver due to device very specific functionality, driver stacking can be used. The driver can be parent driver for specific device driver. This means for specific device driver only specific functionalities should be implemented with less coding. Some consideration will be presented about which kind of standard registers adding to MTCA standard could be helpful to make this kind of driver more complete. The driver already has use cases: a) PITZ timing devices, b) SIS8300 ADC board from Struck. After presenting the activity on this on 2014 MTCA workshop, collaboration was started with colleagues from Hamburg, who are busy on PCIE universal driver development for MTCA. A lot of development and investigations together have been done. The functionalities implemented so far, a comparison of different DMA schemas, and the performance analysis for some generalized functionalities also will be presented.

## Summary

I'll try to present results and convince that generalisation of almost all functionalities is doable without losing in performance or memory usage

**Primary author:** Dr KALANTARYAN, Davit (DESY)

**Co-author:** Mr PETROSYAN, Ludwig (DESY)

**Presenter:** Dr KALANTARYAN, Davit (DESY)

**Session Classification:** Session 8