PROCEEDINGS OF SCIENCE

CERN-IPMC Solution for AdvancedTCA Blades

Julian Mendez[1]
CERN
Geneva CH-1211, Switzerland
E-mail: `julian.mendez@cern.ch`

Vincent Bobillier[a], **Stefan Ludwig Haas**[b], **Markus Joos**[c], **Sylvain Mico**[d], **Francois Vasey**[e]
CERN
Geneva CH-1211, Switzerland
E-mail: [a]`vincent.bobillier@cern.ch`, [b]`stefan.haas@cern.ch`, [c]`markus.joos@cern.ch`, [d]`sylvain.mico@cern.ch`, [e]`francois.vasey@cern.ch`

The AdvancedTCA standard has been selected as one of the hardware platforms for the upgrades of the back-end electronics of the CMS and ATLAS experiments of the Large Hadron Collider. In this context, the CERN EP-ESE group has designed and produced an IPMC mezzanine card for the management of AdvancedTCA blades. This paper presents the CERN-IPMC hardware and the software environment to be used for its customization and describes a test pad that can also be used as a development kit.

Topical Workshop on Electronics for Particle Physics
11 - 14 September 2017
Santa Cruz, California

[1] Speaker

1. Introduction

Originally developed for the telecommunication industry, the AdvancedTCA standard has been selected as one of the platforms for the upgrades of the ATLAS and CMS back-end electronics at CERN. In this context, the CERN EP-ESE group launched in 2011 the xTCA evaluation project whose aim is to perform technical evaluation of equipment, provide support for the selected components as well as design and support standardized system components.

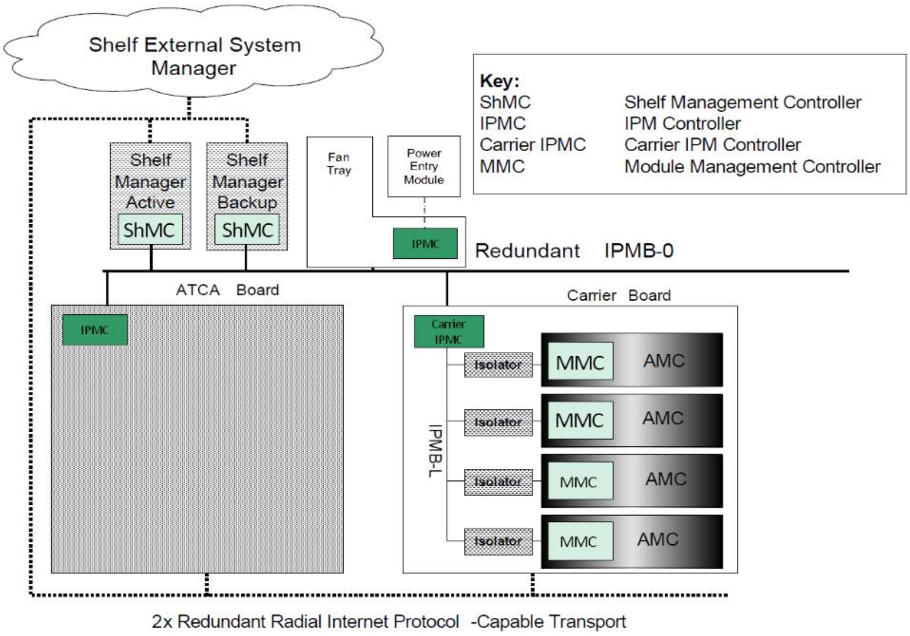

Figure 1: Hardware Platform Management architecture as defined by the AdvancedTCA standard

The AdvancedTCA standard [1], defined by the PCI Industrial Computer Manufacturer Group (PICMG), outlines a modular architecture by defining physical, electrical and functional specifications. Figure 1 shows the hardware management architecture offered by the standard to monitor (temperatures, voltages, current, etc.) and control (fan speed, power management, etc.) the system as well as ensure its proper operation (interoperability, current requirement, e-keying, etc.). These actions are performed by specific controller modules which are interconnected via an Intelligent Platform Management Interface (IPMI) bus: Module Management Controller (MMC) for AMCs, Intelligent Platform Management Controller (IPMC) for ATCA blades and Carrier IPMC for ATCA carriers as well as Shelf Manager for ATCA shelves and MicroTCA Carrier Hubs. In the framework of the CERN xTCA evaluation project, a commercial Intelligent Platform Management Controller (IPMC) solution from Pigeon Point [2] was evaluated in 2015. Following this evaluation, an IPMC mezzanine card was designed to be used on existing and future AdvancedTCA cards.

1.1 General overview

The commercial IPMC solution from Pigeon Point System was adapted to answer the needs defined by the CERN-IPMC specification. To be compliant with already designed AdvancedTCA blades, the DIMM-DDR3 VLP form factor and connector pinout previously defined for the solution developed at LAPP [3] were used. Figure 2 shows the production version of the mezzanine card.

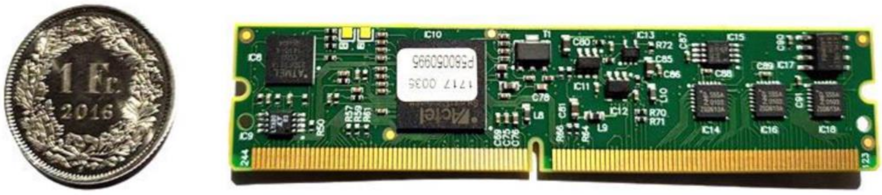

Figure 2: CERN-IPMC mezzanine card

The Firmware has been developed to support the functions defined by the AdvancedTCA rev. 3.0 standard (hotswap, sensor monitoring, etc.), up to 8 AMCs and 1 intelligent-RTM, an Ethernet interface with RMCP/RCMP+, TCP/IP subset and UDP protocols, a serial interface for debug or Serial Over LAN (SOL), up to 51 user I/Os an a JTAG master interface that can be accessed via a Xilinx Virtual Cable server. Finally, the original Pigeon Point firmware was adapted to fit the specified needs and all of the supported features were successfully tested.

2. Architecture

The CERN-IPMC architecture is mainly defined by the use of a SmartFusion FPGA. This Microsemi device provides all of the interfaces required and runs the firmware application on its embedded ARM Cortex M3 processor.

2.1 Hardware

To fit into the selected DIMM-DDR3 VLP form factor, the smallest FPGA with embedded SoC from Microsemi was selected. Figure 3 shows the complete architecture of the CERN-IPMC mezzanine card.

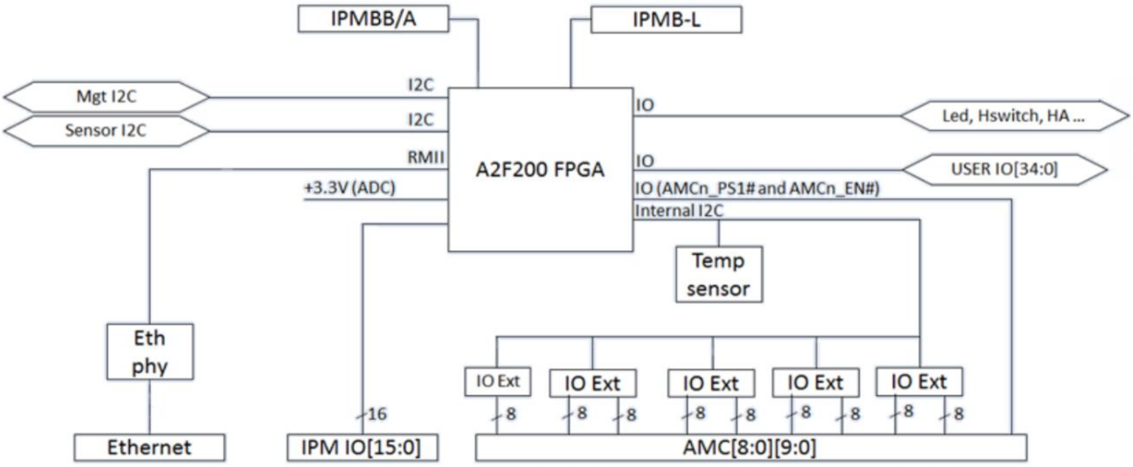

Figure 3: Hardware architecture

Because of resource restriction due to the small component size, only the GPIO cores and the I2C interfaces used for the IPMB buses are implemented in the FPGA Firmware. All of the other I2C lines, for the management, sensor and internal functions are emulated in Software using GPIOs. In addition, the limited number of pins available on this package imposes to control all of the AMC ports, which do not require high speed control, through IO extenders. Finally, the Ethernet port is routed to the RMII interface of the embedded controller via an external PHY.

2.2 Software

The CERN-IPMC software architecture derives from the Pigeon Point one. Almost all of the application source code is re-used and only a few modules were added. Figure 4 shows the architecture available on a git repository.

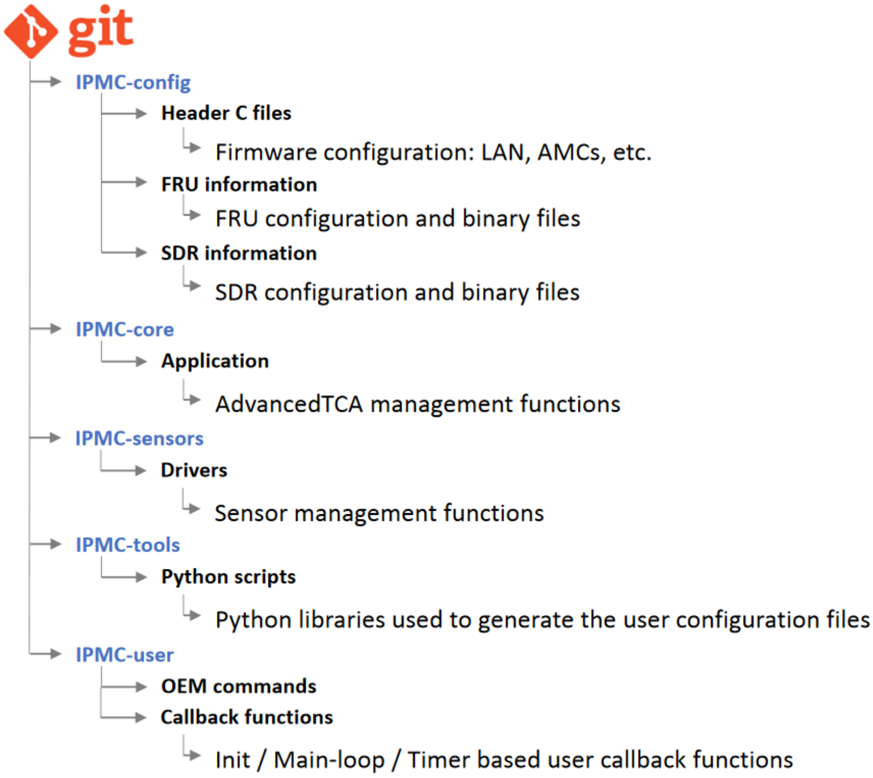
git
IPMC-config
Header C files
Firmware configuration: LAN, AMCs, etc.
FRU information
FRU configuration and binary files
SDR information
SDR configuration and binary files
IPMC-core
Application
AdvancedTCA management functions
IPMC-sensors
Drivers
Sensor management functions
IPMC-tools
Python scripts
Python libraries used to generate the user configuration files
IPMC-user
OEM commands
Callback functions
Init / Main-loop / Timer based user callback functions

Figure 4: Software architecture

The repository is divided in 5 parts. 4 are used for the source code and one for additional tools. The *IPMC-core* contains the functions required to run the features supported by the module (e.g.: Hotswap management, IPMI commands, Ethernet protocols, etc.). In particular, a part of the application consists in reading the sensors and detecting thresholds. However, to perform this task, the core calls device specific and user defined drivers located in the *IPMC-sensor* directory. The configuration files required to customize the application, located in the *IPMC-config* directory, are mainly header files. They contain the description of the different features like the IP address, number of AMCs, Hardware interfaces, Sensors, etc. The folder also contains the FRU binary that describes the AdvancedTCA blade with generic information (manufacturer name, board name, version, etc.) and technical data (backplane ports, maximum current, etc.). To simplify the creation of these different files, a configuration tool, written in Python, was developed. Based on a user defined custom XML file, the system generates the header and binary files automatically. The *IPMC-tools* directory contains the libraries required by the main script. Optionally, custom C functions can be written to implement OEM IPMI commands or to execute specific routines using callbacks. These callbacks, located in the *IPMC-user* directory, can be called at initialization time, everytime the main-loop is executed or can be timer based and executed every x ms. Finally, libraries and examples are provided to show how the user C code can be written and additional features can be added to the main application of the controller.

3. Automatic tester

An automatic tester system was designed to verify the IPMC functionality after production. To get a complete environment, which can be used either for automatic test or manual debugging, all of the interfaces of the mezzanine cards are emulated. Figure 5 shows the hardware architecture of the tester.

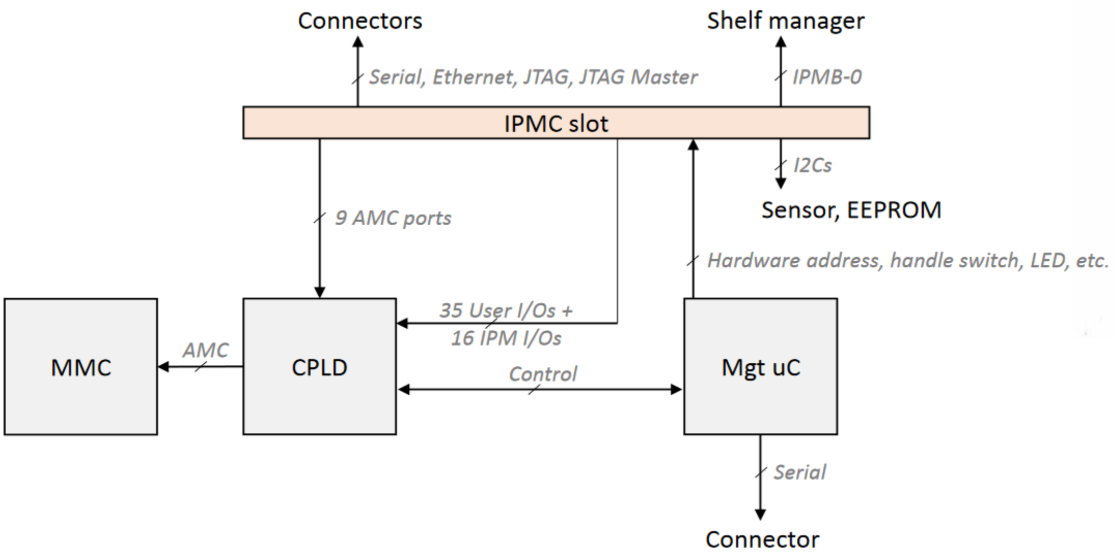

Figure 5: IPMC tester architecture

The system is supervised using an 8bit microcontroller managed by a PC via a USB to serial interface. A LABView library was designed to execute commands for the monitoring and control of the IPMC interfaces. The library also provides functions to configure the CPLD used for the dynamic routing of the GPIOs and AMC ports. All of the AMC ports are tested using an emulated mezzanine card with an on-board MMC module. Finally, the I2C interfaces are connected to sensors and EEPROM to check their functionality. Serial, Ethernet and JTAG ports are routed to connectors for functional check using external devices

4. Conclusion

A first batch of 50 CERN-IPMC mezzanine cards has been produced and successfully tested. They are currently being beta-tested by 2 users in ATLAS (MUCTPI) and CMS (Demonstrator blade) and the software continues to evolve based on their feedback. A complete IPMC kit is available for distribution made of the IPMC mezzanine card, Pigeon Point licence fee, access to the source code and support.

References

[1] PICMG, *AdvancedTCA Short Form Specification*, https://www.picmg.org/openstandards/advancedtca/

[2] Pigeon Point, *Pigeon Point's PICMG products*, http://www.pigeonpoint.com/products_picmg.html.

[3] LAPP, *IPMC mezzanine card*, http://lappwiki.in2p3.fr/twiki/bin/view/AtlasLapp/ATCA.