OpenIPMC

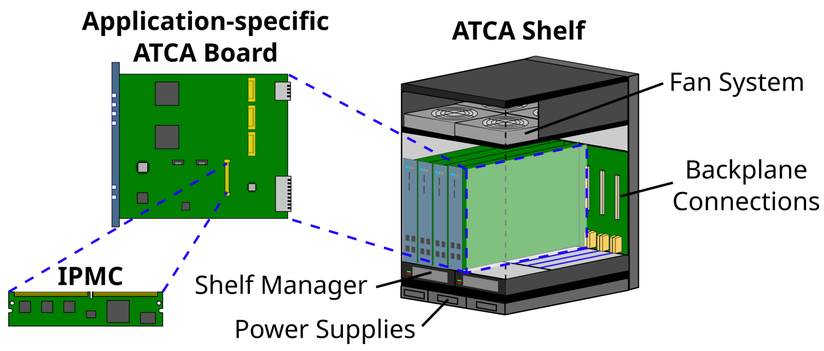
An open-source Intelligent Platform Management Controller

Luigi Calligaris, André Cascadan, Bruno Augusto Casu Pereira de Sousa (UNESP, São Paulo); Lucas Arruda Ramalho (UNEMAT, Barra do Bugres); Alison da França Costa, Ailton Akira Shinoda (UNESP, Ilha Solteria); Luis Ardila-Perez, Oliver Sander (KIT, Karlsruhe)

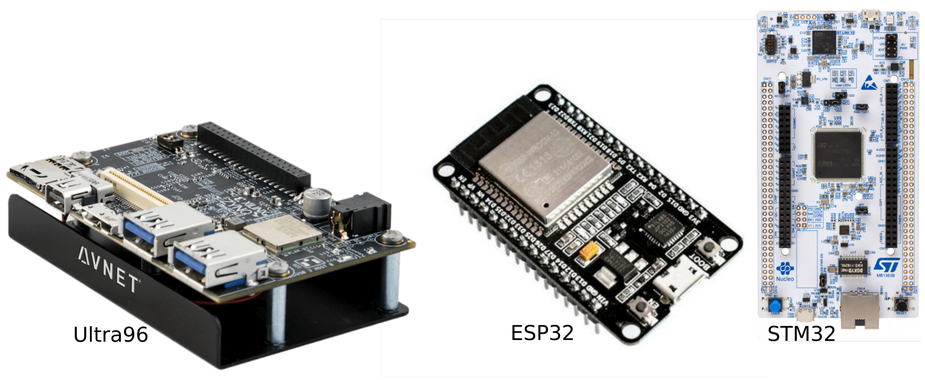
Luigi.Calligaris@cern.ch

Introduction

Electronic systems compliant to the PICMG ATCA [1] standard are attractive for the back-end electronics of future HEP experiments, including those planned at the High-Luminosity LHC (HL-LHC)[2][3]. In an ATCA shelf, the Shelf Manager Controller (ShMC) monitors all the Field Replaceable Units (FRUs), such as electronic boards and fans, responding to planned and unplanned events (extraction of a board or failure in a cooling fan, for example). In addition, each FRU is equipped with an Intelligent Platform Management Controller (IPMC), tasked to provide the ShMC sensor readings and FRU information, and manage FRU powering.



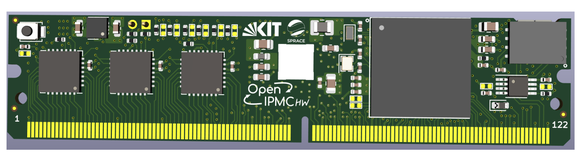
Tested architectures (so far)



We developed OpenIPMC on the AVNET Ultra96 board, powered by an ARM-based Zynq UltraScale+ ZU3EG SoC. Then, as an exercise to prove the portability of our software, we ported it to the Espressif ESP32: a very popular microcontroller based on a Tensilica Xtensa LX6 core. We then ported OpenIPMC to the STM32H745 platform, a powerful dual ARM M7/M4 designed for industrial control. In all cases porting took just 3 person-weeks, showing that portability is rather easy thanks to the abstractions presented to user code by FreeRTOS.

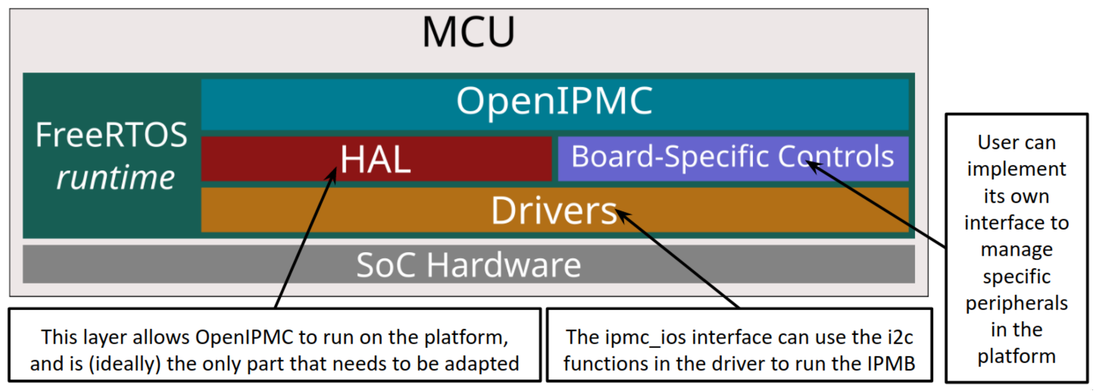
Future Outlook

An open IPMC hardware solution running OpenIPMC is currently under development. It is based on a powerful STM32H745 microcontroller.



OpenIPMC

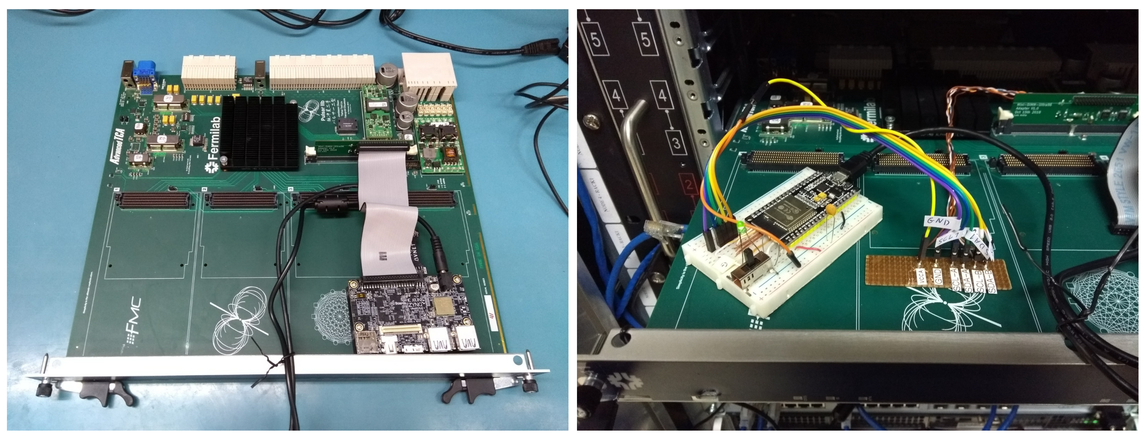
OpenIPMC is a free and open-source IPMC software written in C, which can be built for any embedded platform supporting FreeRTOS. It runs as a set of tasks interacting with each other via mutexes, queues, and other native resources provided by the aforementioned operating system. A Hardware Abstraction Layer (HAL) is used to interface OpenIPMC to the hardware-specific IO drivers, making the software very easy to port.



For a given hardware implementation, the user registers callbacks to driver functions using OpenIPMC's API, allowing the latter to use these drivers to operate the device. This gives wide flexibility in adapting the software to the user's needs.

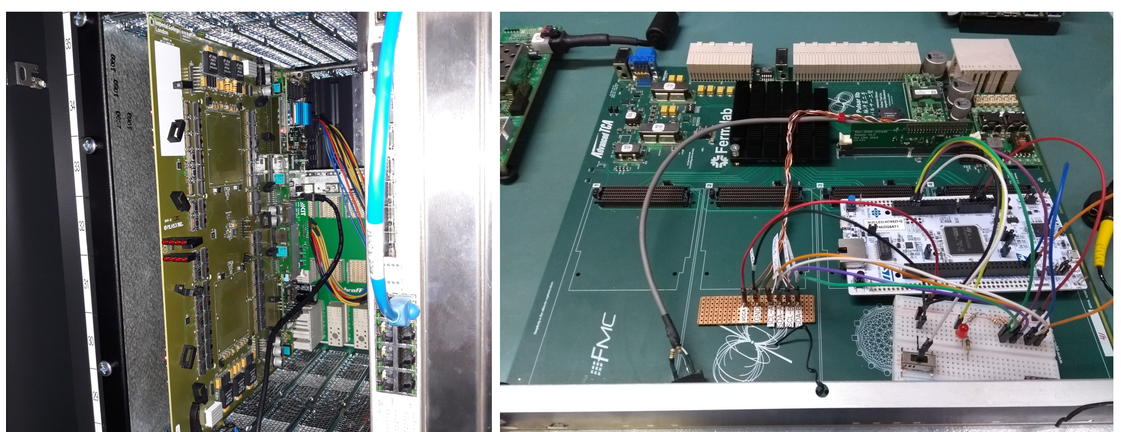
Development setups

To interface the ATCA shelf with the Ultra96, ESP32 and STM32 development boards running OpenIPMC at SPRACE, we used a Pulsar 2b board[4], equipped with a suitable DIMM adapter sitting in the IPMC slot, providing level shifting and break outs for the IPMB-A and -B buses.



Running OpenIPMC on Ultra96 (left) and ESP32 (right) at SPRACE.

We are also currently testing OpenIPMC on a test setup designed and installed at KIT, based on a Serenity board[5] equipped with a custom management module based on a Zynq Ultrascale+ SoC. In this system, OpenIPMC resides in one of the embedded memories (TCM) and runs on the real-time Cortex-R5 cores.



Running OpenIPMC on a ZYNQ module at KIT (left) and on STM32 at SPRACE (right).

The four different platforms presented here run the same OpenIPMC code, with modifications only in the HAL layer and board-specific controls.

Links, references and acknowledgments

OpenIPMC source published at: https://gitlab.com/openipmc
Doxygen documentation at: https://openipmc.gitlab.io/openipmc

[1] PICMG, "3.0 AdvancedTCA Base Specification", https://www.picmg.org/openstandards/advancedtca/
[2] CMS Coll., "The Phase-2 Upgrade of the CMS Tracker", CERN-LHCC-2017-009
[3] ATLAS Coll., "Technical Design Report for the ATLAS Inner Tracker Pixel Detector", CERN-LHCC-2017-021
[4] J. Olsen and T. Liu and Y. Okumura, "A full mesh ATCA-based general purpose data processing board", JINST 9, no.01, C01041 (2014)
[5] A. Rose et al.,"Serenity: An ATCA prototyping platform for CMS Phase-2", PoS TWEPP2018, 115 (2019)

This material is based upon work supported by the São Paulo Research Foundation (FAPESP) under Grants No. 2013/01907-0 and 2018/18955-0. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of FAPESP.