OpenIPMC-HW

An open-source Intelligent Platform Management Controller module

André Cascadan, Luigi Calligaris (SPRACE - UNESP)

In this presentation

- ATCA and IPMC
- OpenIPMC software
- OpenIPMC-HW: The proposed hardware
- OpenIPMC-HW: Testing and Development

ATCA and IPMC

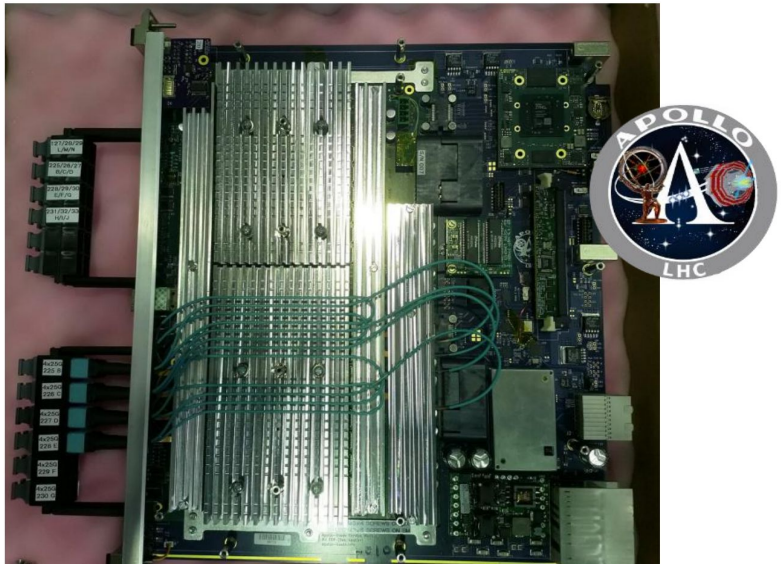
Advanced Telecommunication Computing Architecture (ATCA)

- Industrial computing standard
  - Reliability
    - System monitoring & management
    - Components redundancy
  - High power available to devices
    - Forced air cooling by the use of fans
    - More than 10 kW / cabinet possible
    - Big FPGAs and fast optical transceivers
  - Good inter-board backplane connections
    - Precise clock synchronization (10s of ps jitter)
    - High-speed communication (16-20 Gbit)
- ATCA crates used by CMS: up to 14 boards
  - 2 shelves are planned per cabinet
  - Cabinets for OT DTC, IT DTC and TF
  - Cabinets will be **bought from a commercial supplier**

ATCA boards in HEP

Serenity board (Imperial College London)

- Proposed for the role of OT DTC
- Total of 216 to be used
- ATCA standard

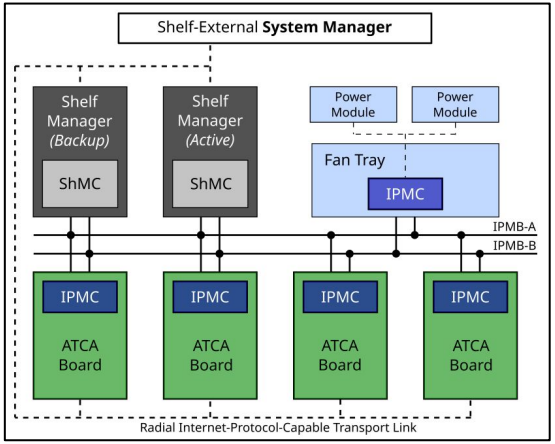


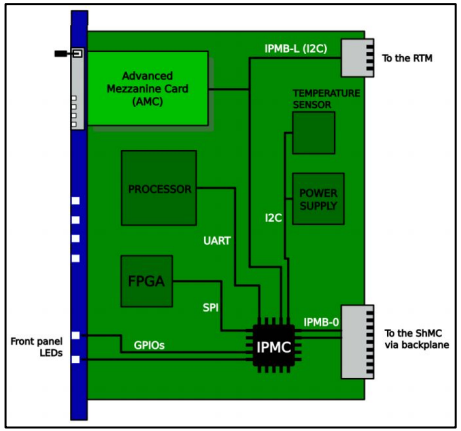
Apollo board (Boston University)

- Proposed for the roles of IT DTC and TF
- Total of 28 (DTC) + 162 (TF) to be used
- ATCA standard

ATCA Shelf Manager and IPMC

- ATCA shelf like a LEGO set
  - Mechanical frame + interconnection backplane
  - Replaceable components (FRUs)
    - Boards, power supplies, boards, shelf managers, ...
- Shelf Manager Controller (ShMC)
  - Orchestrates turning on/off the FRUs
  - Controls the power of the fans (cooling loop)
  - Polls to each FRU to monitor its conditions
  - Provides a shelf management interface via UDP/IP
- Intelligent Platform Management Controller (IPMC)
  - One for each FRU (including the boards)
  - Controls the power of the FRU
  - Reports the status of the board to the ShMC
  - Agrees with the ShMC changes in the power state



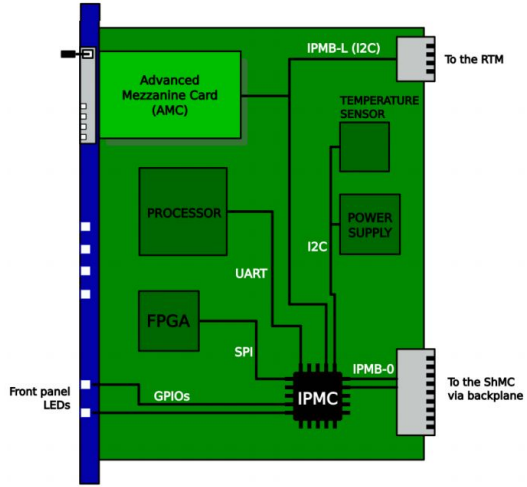


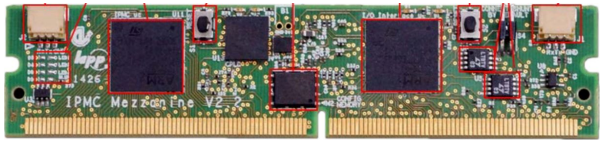
The ShMC is an orchestra director, doesn't know how to play an instrument

The IPMC follows the ShMC and knows to play its own specific instrument

IPMCs for the ATCA boards

- Now, consider ATCA boards designed by the user
  - e.g. the Serenity or the Apollo boards for the CMS tracker
- The IPMC is **specific** to that board design
  - Different boards have different components
    - CPUs, FPGAs, hard disks, radio transceivers, …
    - The IPMC needs to know how to turn them on/off
    - The IPMC needs to know how to read temp/voltage/...
- The board designer chooses an IPMC for the board
  - This can be an IPMC designed by him...
  - ...or an IPMC designed by **someone else**
  - What counts is the **configuration** of the IPMC for that board
- LHC expts adopted an IPMC DIMM standard by LAPP (Annecy, FR)
  - LAPP IPMC module was quite complicated → abandoned
  - FNAL IPMC designed uniquely for Pulsar2b (**fw by SPRACE**)
  - CERN IPMC is the adaptation of a commercial product by PPS
    - **BIG** problems with license, NDAs, support...





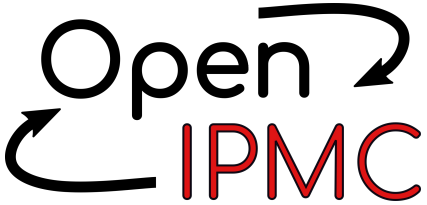
LAPP IPMC

FNAL IPMC

CERN IPMC

OpenIPMC software

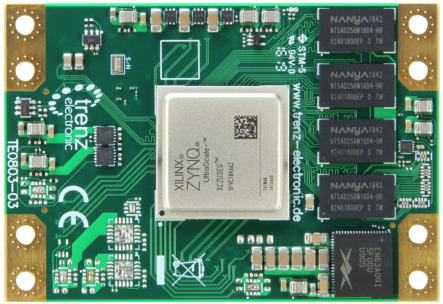
OpenIPMC software

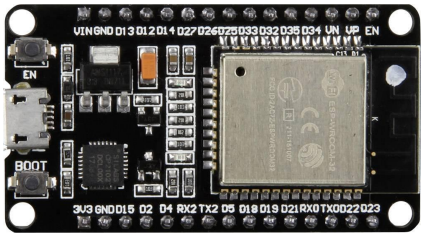


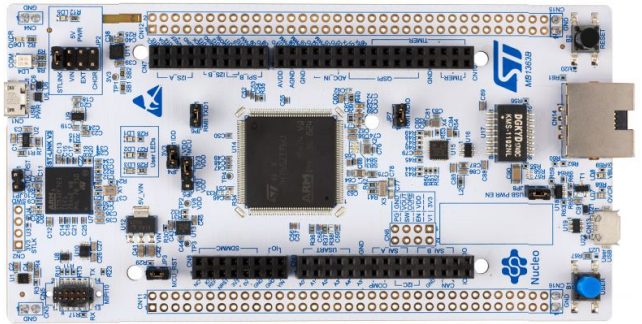
- IPMC software implementing PICMG-compliant IPMI functions
  - Power negotiation and hot-swap (M-states, handle, etc.)
  - Instantiate board sensor records, declare them to ShM, read-out and publish data
  - Focus on simplicity: optional functions can be added to the project by the user
- Platform-independent design, written in C
  - Can quickly port the project to different architectures (e.g. ZynqMP, ESP32, STM32)
- Based on FreeRTOS operating system
  - Can run independent “tasks” in parallel (w/ prioritization)
  - Flexible software development, thanks to task decoupling
  - Supported by many SoC manufacturers (TI, NXP, ST, Xilinx, Microsemi…)
- OpenIPMC is free and open source software (git repo)
  - Can be easily customized to fit a new board, and modified to be debugged
  - No need to sign NDAs for contributors, curious newcomers and students

Evolution of OpenIPMC support on different devices

- First platform: Cortex-R5 cores on Zynq US+
  - IPMC (R5) and Linux (A53) running in the same device
  - Targeting the ATCA-ZynqMP management module by KIT (proposed for Serenity-A2577)
- Portability exercise: ESP32 microcontroller
  - Not a “serious” device, but very different arch from Zynq, cheap and very flexible
- First mainstream MCU: STM32 microcontroller
  - Successful porting opened the way to design of the DIMM module

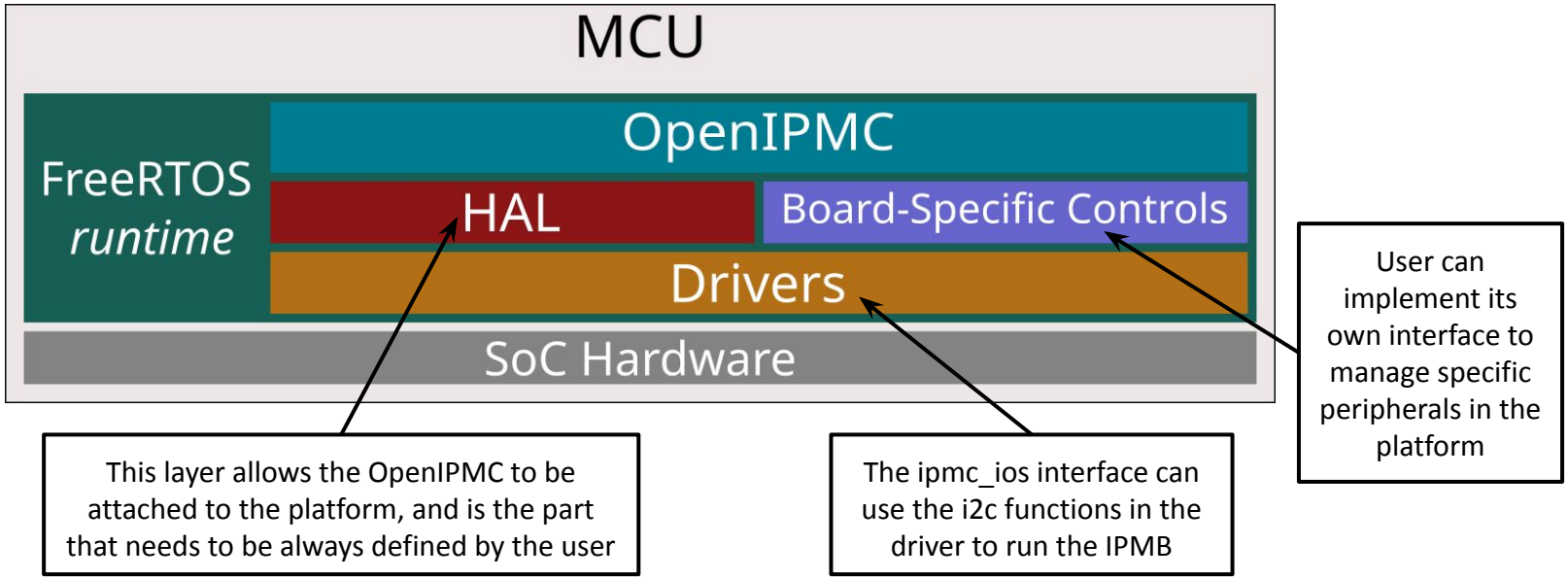






OpenIPMC runtime structure

- Two interfaces between OpenIPMC hardware-agnostic code and hardware drivers
  - **Hardware Abstraction Layer** → interface to hardware driver used for IPMI functions (IPMB, blue led..)
  - **Board-specific controls** → customize board-specific behavior (how to turn on power, read sensors..)
- Other tasks (not shown in picture) can run aside of the OpenIPMC stack



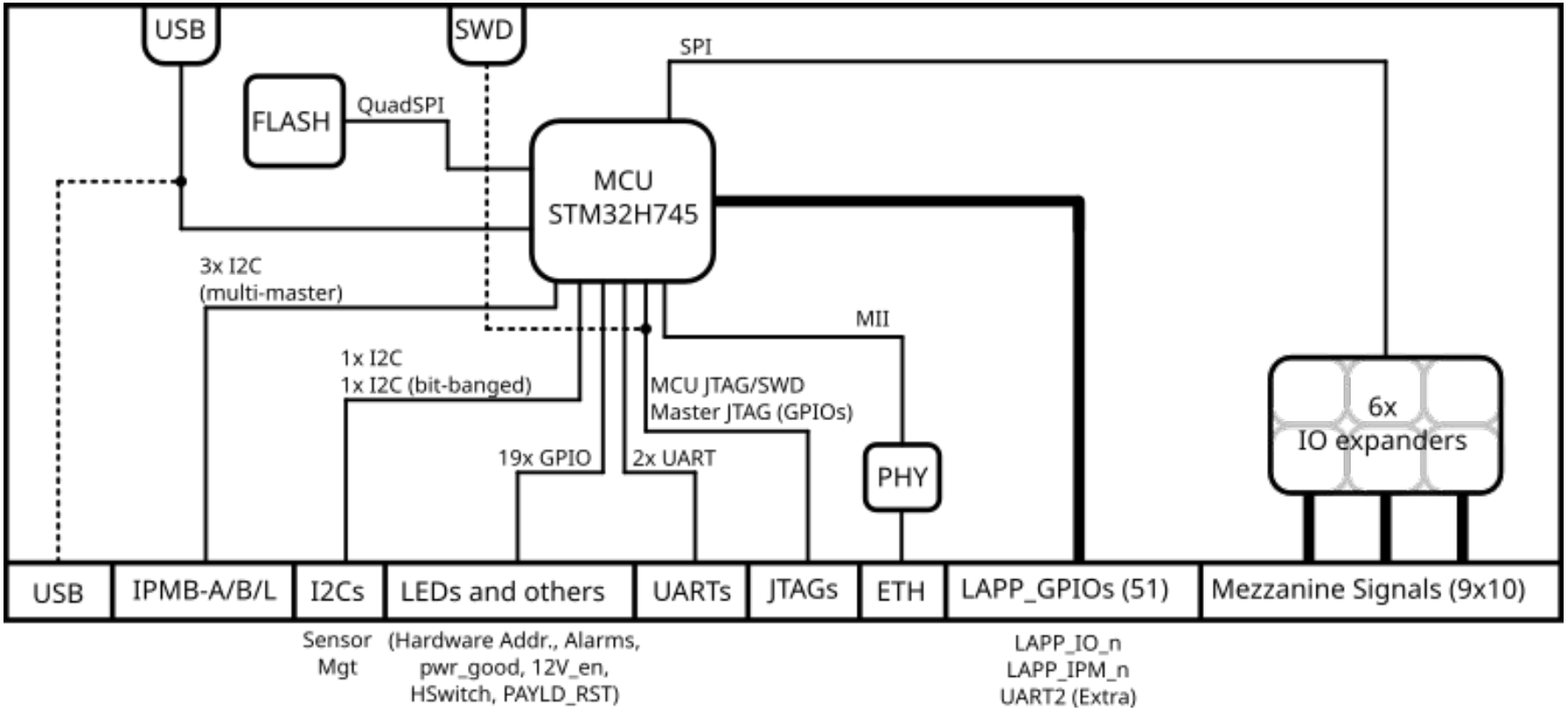
OpenIPMC - HW
The proposed hardware

The OpenIPMC-HW board

- An open hardware platform for IPMC software (like OpenIPMC)
  - Complements OpenIPMC to realize a fully open management platform
  - Designed using the CERN-sponsored KiCAD EDA suite
- Fully compatible with CERN-IPMC and LAPP IPMC specifications
  - **JEDEC MO-244 LP-DIMM** form factor: 82,0 x 18,3 mm
  - Compatible with the new UART scheme (SoC debug via SOL + IPMC debug + SoC configuration)
  - 9 sets of AMC control pins + all LAPP GPIOs routed to the
- Hardware components
  - We chose low-cost components, in current production with long lifecycle
  - ST Microelectronics **STM32H745** Microcontroller
    - Extensive documentation & free toolchain/IDE software (STM32CubeIDE)
- Relatively simple manufacturing
  - 8 layers FR4, no buried/blind vias, 0.8 mm pitch BGA, needs hard gold edge fingers

Link to GitLab repo.: https://gitlab.com/openipmc/openipmc-hw

OpenIPMC-HW layout: schematic

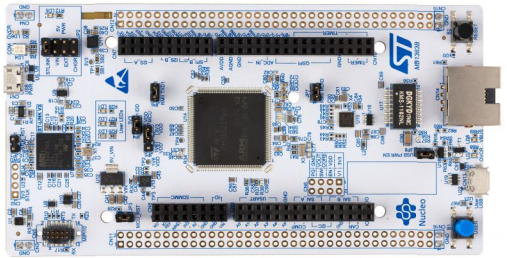


Choice of the microcontroller

- The OpenIPMC software runs on top of FreeRTOS
  - Software shown to be easily portable on new MCUs (~3 wks)
  - Plenty of MCU manufacturers to choose from

- We chose **STM32H745XIH6** by STMicroelectronics
  - Number of I2C/SPI hardware peripherals → 4 / 6
  - Number of GPIOs/UART/USART → up to 168 / 4 / 4
  - Availability of an free toolchain → STM32CubeIDE
  - Availability of an evaluation board → NUCLEO-H745ZI-Q (cost: 23 CHF)
  - Our experience with other STM32 MCUs → STM32F103C8T6 (e.g. “Blue pill” board)
  - Performance margin for future upgrades → 480 MHz Cortex-M7+240 MHz Cortex-M4
  - Large SRAM/Flash memories → 1024 kiB / 2048 kiB
  - Expected reliability of the manufacturer → STMicroelectronics is a leader in MCUs
  - Cost → 17.45 $ per piece

- What we get in addition
  - High speed USB device/host/OTG → USB programming & terminal
  - Efficient SMPS to power the core → better thermals
  - External memory support → store config/firmwares/etc
  - Lots of other features we will not use (e.g. HDMI driver)

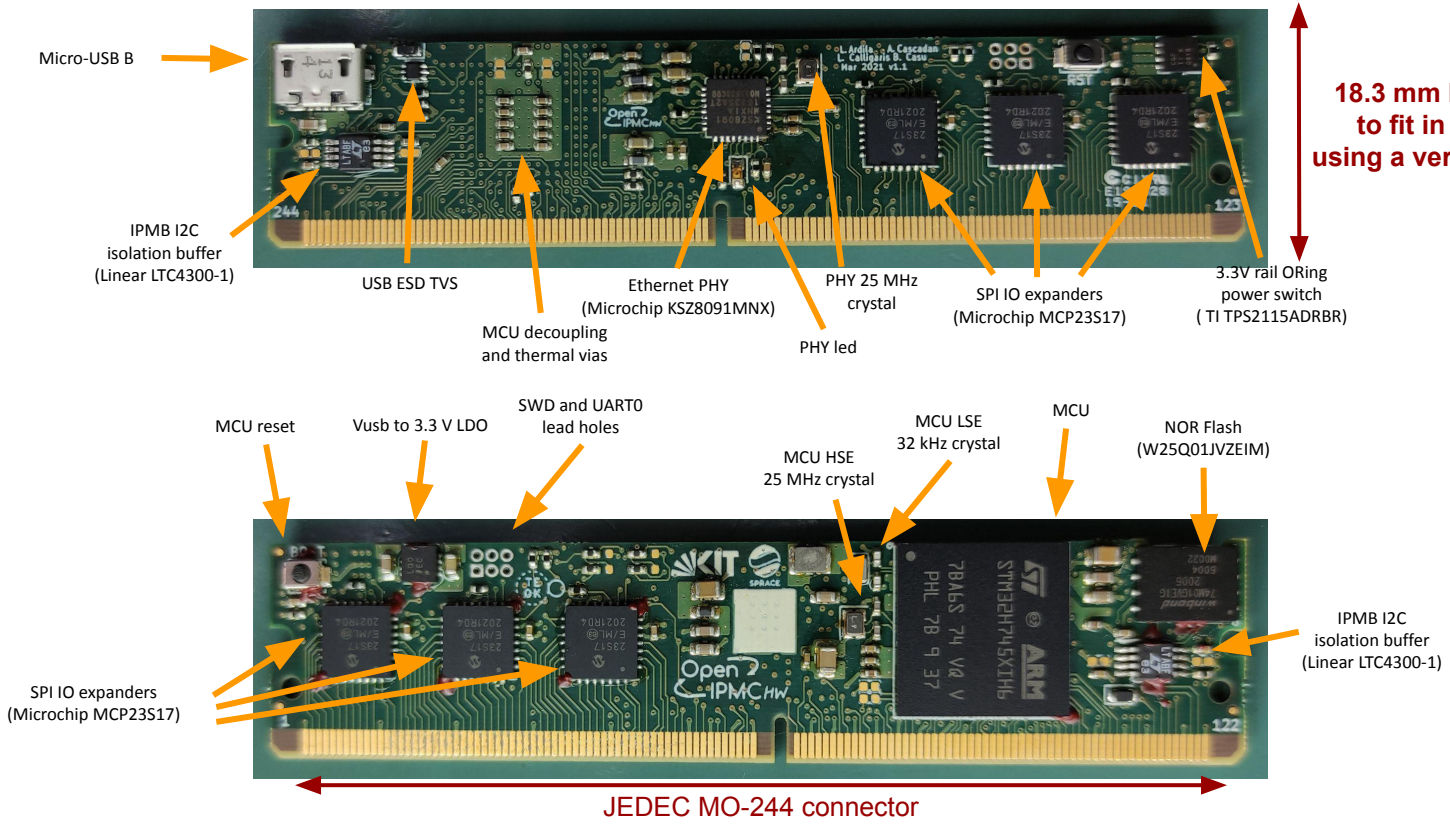


STM32 NUCLEO-H745ZI-Q

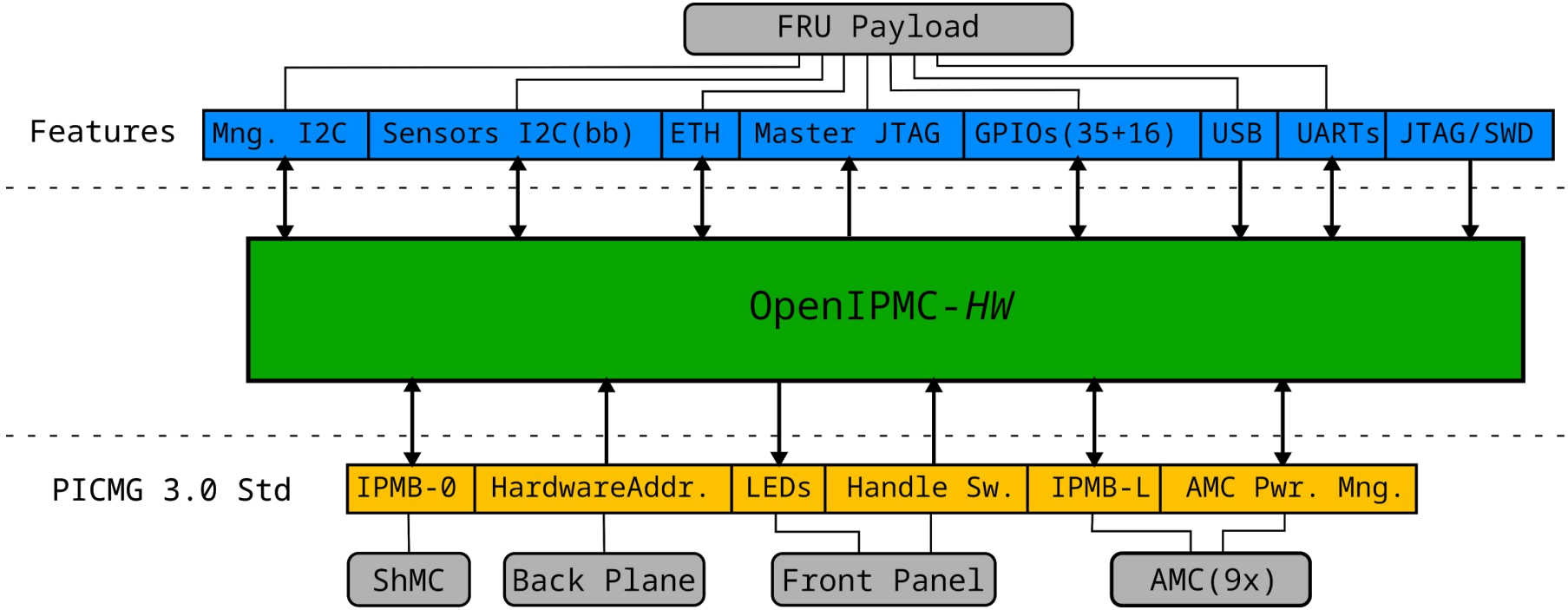
Full documentation on ST site

https://www.st.com/en/microcontrollers-microprocessors/stm32h745-755.html#documentation

OpenIPMC-HW layout: picture



OpenIPMC-HW DIMM connections



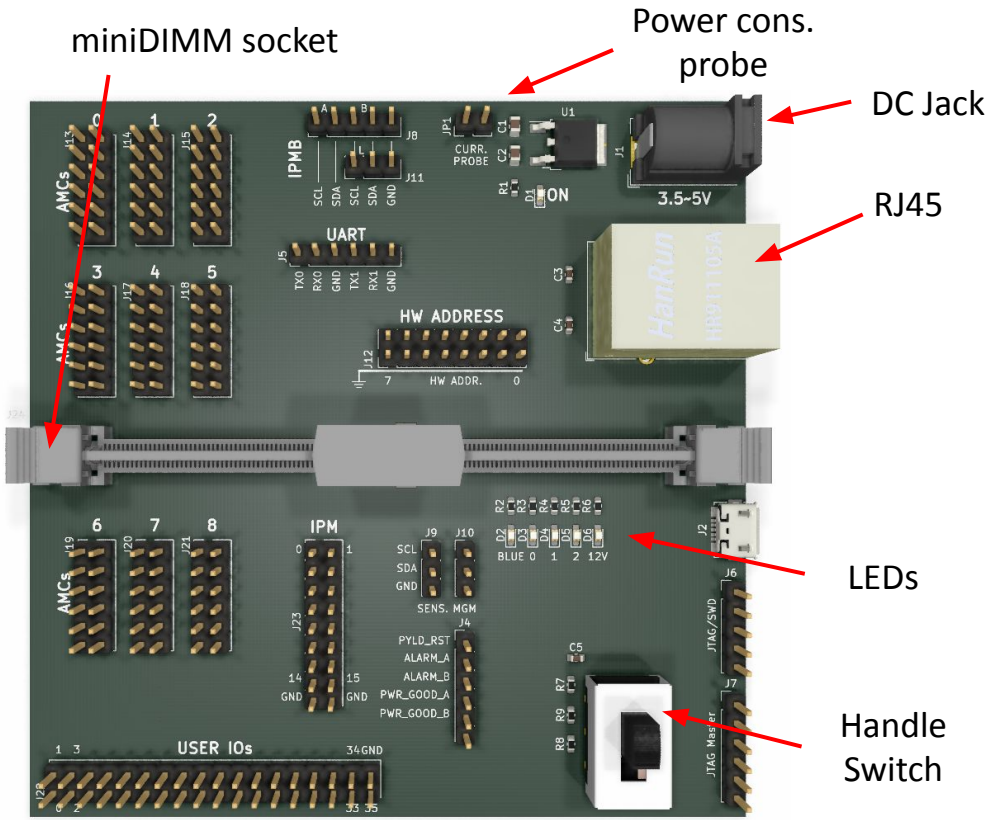
Note: among the GPIOs some pins can be configured as UARTs, following the SoC Interest group layout

OpenIPMC-HW
Testing and Development

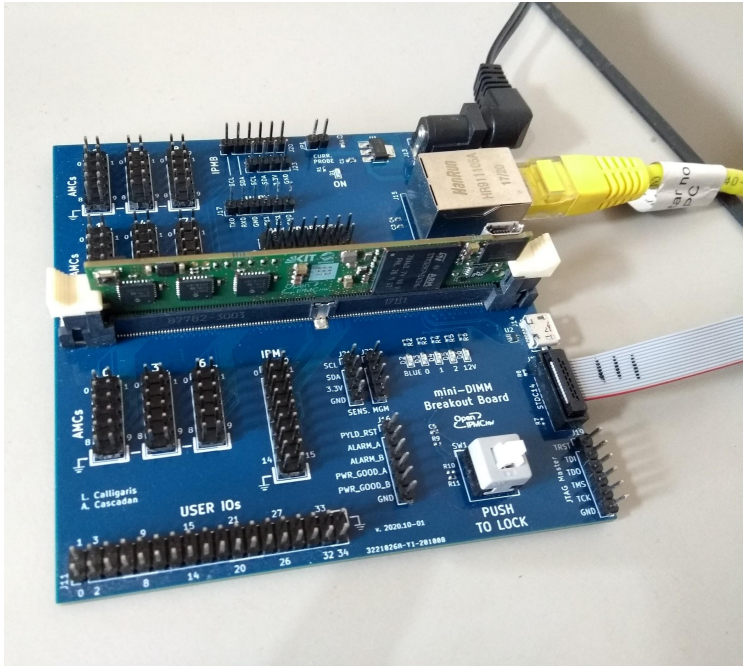
OpenIPMC-HW Breakout Board

- Designed to break out the DIMM pins to various 100-mil connectors
- Simple design allowing cheap manufacturing from any source
  - Just the necessary active components (e.g. a 3.3V LDO)
  - No BGA components, all hand-solderable if preferred (except miniDIMM conn.)
  - Four layers FR4, 10 cm x 10 cm
- Important tool to test and validate the OpenIPMC-HW design, and the porting of OpenIPMC to STM32

OpenIPMC-HW Breakout Board



3D model for the breakout board

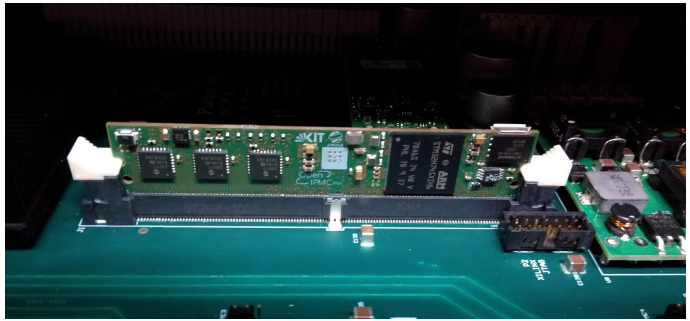


Manufactured PCBs for the breakout board

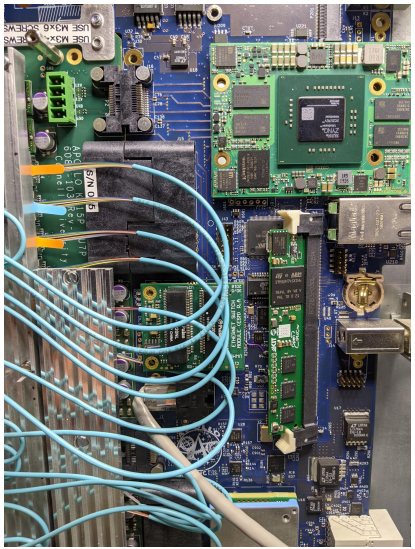
Test setups

- OpenIPMC-HW is currently being tested in 3 ATCA boards
  - Pulsar-IIb, at SPRACE (São Paulo)
  - Serenity, at KIT (Germany) and CERN
  - Apollo, at Boston University

For each target board, the OpenIPMC-FW code is forked and adapted to the specific hardware.
We took care to make this process easy for users.



Pulsar-IIb @ SPRACE



Apollo @ BU

Serenity @ KIT

Sensor readings

HotSwap Sensor

IPMB Sensor

Temperature form PIM400

```
#
# clia sensordata 8c

Pigeon Point Shelf Manager Command Line Interpreter

8c: LUN: 0, Sensor # 1 ("Hot Swap Carrier")
    Type: Discrete (0x6f), "Hot Swap" (0xf0)
    Belongs to entity (0xa0, 0x60): FRU # 0
    Status: 0xc0
        All event messages enabled from this sensor
        Sensor scanning enabled
        Initial update completed
    Sensor reading: 0x00
    Current State Mask 0x0010

8c: LUN: 0, Sensor # 2 ("IPMB-0 Sensor")
    Type: Discrete (0x6f), "IPMB Link" (0xf1)
    Belongs to entity (0xa0, 0x60): FRU # 0
    Status: 0xc0
        All event messages enabled from this sensor
        Sensor scanning enabled
        Initial update completed
    Sensor reading: 0x88
    Current State Mask 0x0008

8c: LUN: 0, Sensor # 3 ("TEMP PIM400")
    Type: Threshold (0x01), "Temperature" (0x01)
    Belongs to entity (0xa0, 0x60): FRU # 0
    Status: 0xc0
        All event messages enabled from this sensor
        Sensor scanning enabled
        Initial update completed
    Raw data: 42 (0x2a)
    Processed data: 32.320000 degrees C
    Current State Mask: 0x00
```

Current on PIM400

Voltage on -48 line (Channels A and B)

```
8c: LUN: 0, Sensor # 4 ("CURRENT PIM400")
    Type: Threshold (0x01), "Current" (0x03)
    Belongs to entity (0xa0, 0x60): FRU # 0
    Status: 0xc0
        All event messages enabled from this sensor
        Sensor scanning enabled
        Initial update completed
    Raw data: 3 (0x03)
    Processed data: 0.282000 Amps
    Current State Mask: 0x00

8c: LUN: 0, Sensor # 5 ("-48V_A PIM400")
    Type: Threshold (0x01), "Voltage" (0x02)
    Belongs to entity (0xa0, 0x60): FRU # 0
    Status: 0xc0
        All event messages enabled from this sensor
        Sensor scanning enabled
        Initial update completed
    Raw data: 162 (0xa2)
    Processed data: 52.650000 Volts
    Current State Mask: 0x00

8c: LUN: 0, Sensor # 6 ("-48V_B PIM400")
    Type: Threshold (0x01), "Voltage" (0x02)
    Belongs to entity (0xa0, 0x60): FRU # 0
    Status: 0xc0
        All event messages enabled from this sensor
        Sensor scanning enabled
        Initial update completed
    Raw data: 162 (0xa2)
    Processed data: 52.650000 Volts
    Current State Mask: 0x00

# 
```

Sensor reading test: Shelf Manager CLI is printing the sensor readings of Serenity @ KIT.

Summary

- The ATCA standard is very interesting for use in HEP experiments
  - High speed backplane, high power, high reliability
  - Management capability
- IPMCs are essential devices for ATCA board management
  - All ATCA boards must have it
  - Its configuration is highly dependent on the ATCA board characteristics
- SPRACE developed OpenIPMC-HW
  - Compatible with CERN-IPMC and LAPP IPMC specifications
  - Free and open-source hardware
  - Executes a port of OpenIPMC portable software for IPMI operations
- It is being tested in 3 different boards
  - Pulsar-IIb: at São Paulo
  - Serenity: at KIT(Germany) and CERN
  - Apollo: at Boston University

Thanks from the SPRACE team!

A pesquisa aqui apresentada recebe o suporte da FAPESP (processos 2018/18955-0 e 2019/18166-9)