OpenIPMC

A portable FOSS software for IPMCs

André Cascadan, Bruno Casu, Luigi Calligaris (SPRACE)

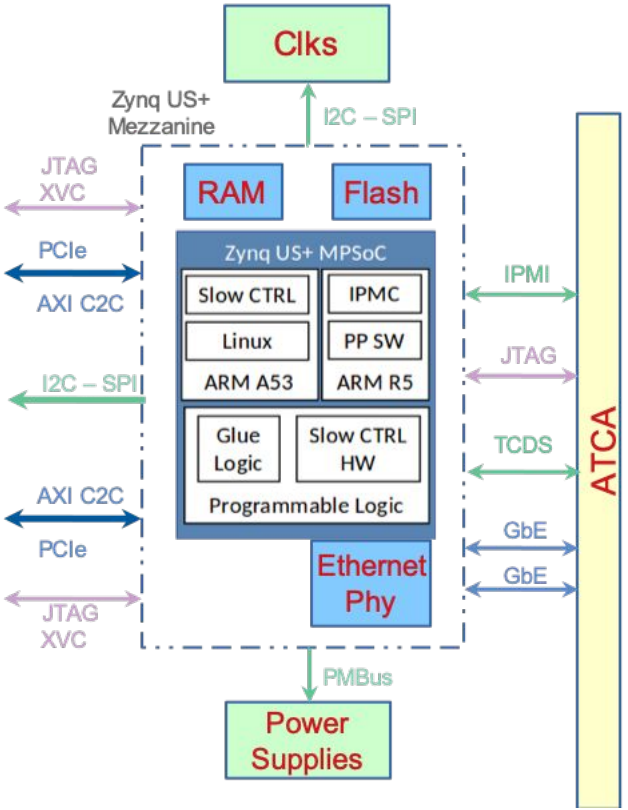
Lucas Arruda Ramalho (UNEMAT), Luis Ardila, Oliver Sander (KIT)

In this presentation

1. Motivation: ZynqMP as central control module
2. Pigeon Point IPMC and OpenIPMC
3. Reference development platform: Ultra96
4. Tests on Trenz + Serenity setup @ KIT
5. Tests on ESP32 setup

Motivation: ZynqMP as central control module

- **ATCA boards for LHC experiments need**
  - An IPMC for board management & monitoring
  - A Linux system for higher-level functions
  - The ZynqMP is a proposed integrated solution
- **Zynq Ultrascale+ MPSoC**
  - Two ARM processor domains (APU and real-time)
  - Xilinx FPGA programmable logic (good 4 sys integration)
  - Plethora of peripherals (PCIe, Eth, I2C, UART, USB, …)
- **Power domain partitioning**
  - LPD: IPMC (standalone/RTOS) on ARM-R5, TCM
  - FPD: Slow Control (Linux) on ARM-A53 quad cores
  - FPGA for services with HW support
- **Two firmware solutions proposed for IPMC**
  - Commercial closed-source by Pigeon Point
  - Free and open-source (OpenIPMC)



Zynq reference design for ZynqMP development

Trenz + Serenity setup @ KIT

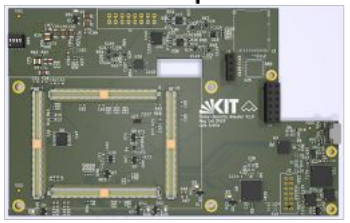
- Based on Serenity (Imperial College)

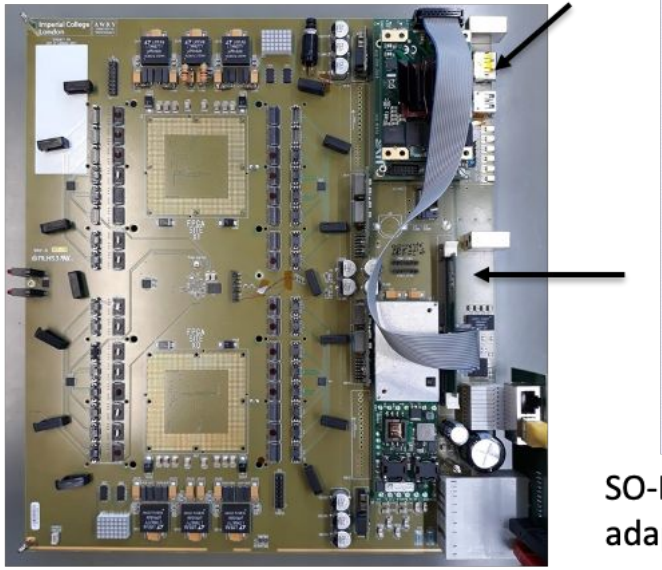
Hardware

- Trenz Elektronik TE0803 module
  - Zynq US+ ZU4EG SoC
- Trenz Adapter board
  - Interface TE0803 to COM Express slot
  - Additional IPMC features
  - (I2C buffers, Eth Phy, EEPROM, SDCard…)
  - Interface to DIMM adapter
- DDR3 Mini-DIMM Adapter
  - Fits into CERN IPMC-compatible slot
  - Access to IPMC backplane signals

Trenz 0803

Trenz Adapter





Serenity

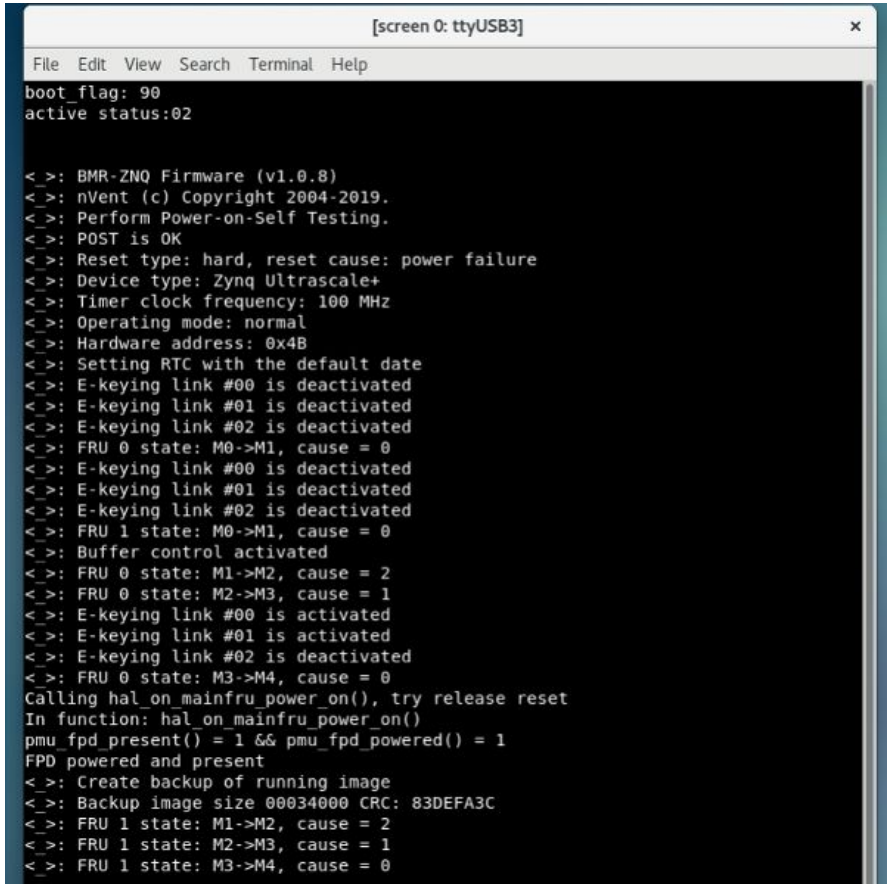
SO-DIMM adapter

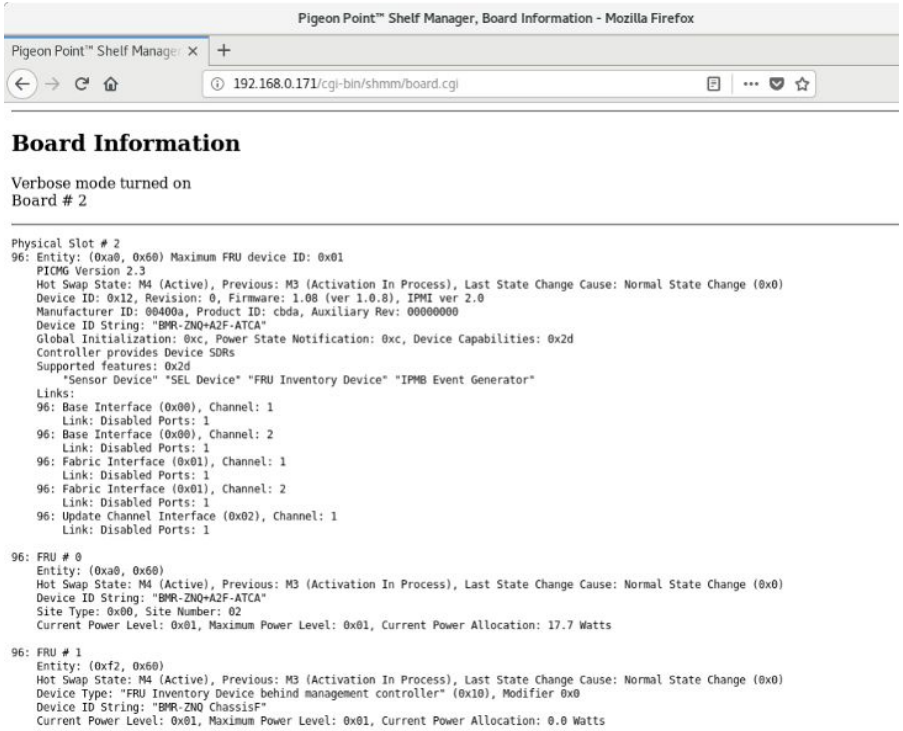
IPMC software solution: Pigeon Point IPMC

- Pigeon Point ATCA IPMC software
  - Prototype version based on VPX version for ZynqMP (BMR-ZNQ-VPX)
  - Extensions by Pigeon Point and KIT for ATCA compliance
- Status
  - Integration and functional tests at KIT
  - Functional tests in Tracker Integration Facility (TIF) at CERN
  - Compliance tests at CERN test stand successful
    (some failed because of disabled functionality)

Pass	Fail	Error	Abort	Skip	Inconclusive
79	34	0	0	24	0

Pigeon Point IPMC - Boot log & IPMI FRU info



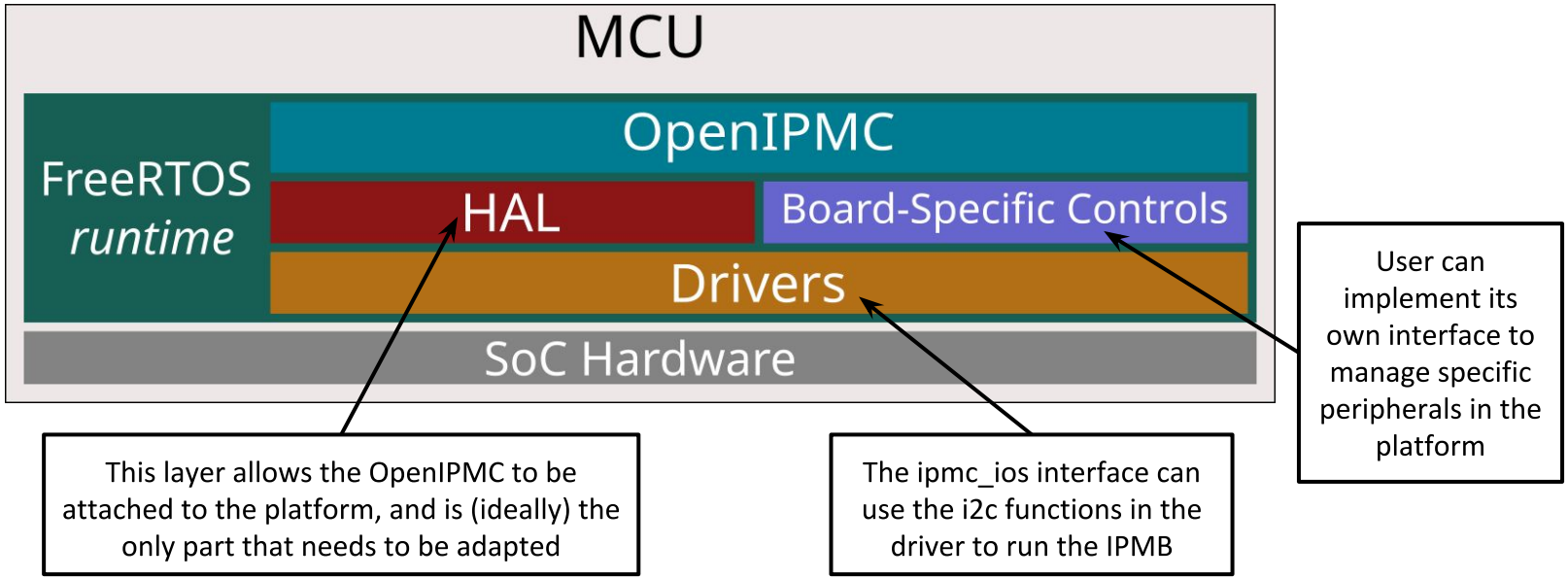


IPMC software solution: OpenIPMC

- Free open source PICMG-compliant IPMC software
  - Implements the IPMI functions required by PICMG ATCA
    - Instantiate board sensors, declare them to ShM, read-out and publish data
    - Power negotiation and hot-swap (M-states, handle, etc.)
    - Focus on simplicity: optional functions can be added to the project by the user
  - Developed at SPRACE to run on the ZynqMP Management Module (KIT)
    - Platform-independent choices allowed to quickly extend to other architectures
- FreeRTOS operating system
  - Can run independent processes in parallel (w/ prioritization)
  - Flexible software development, thanks to process decoupling
  - Supported by many SoC manufacturers (TI, NXP, Xilinx, Microsemi…)

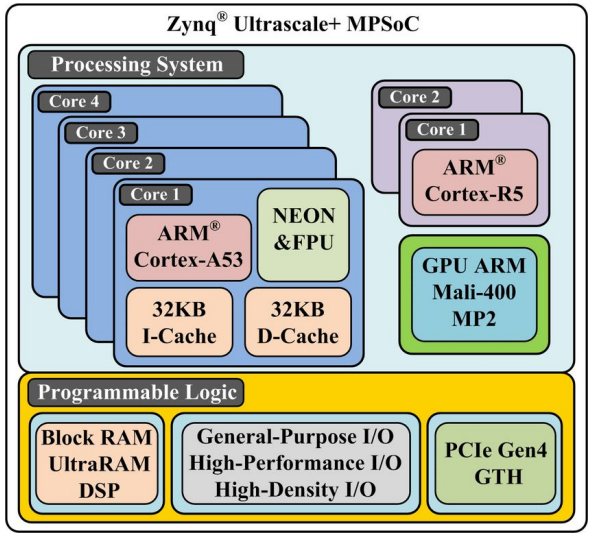
Hardware decoupling

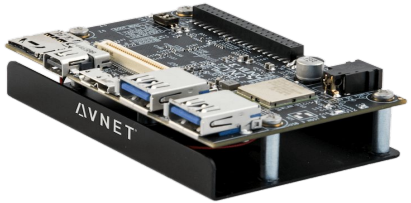
- For each uController architecture, OpenIPMC needs a Hardware Abstraction Layer (HAL)
- HAL functions are passed to OpenIPMC to interface it with the low-level drivers
  - This enables hardware management (I2C multi-master mode, GPIOs readings, etc)



Reference development platform: Ultra96 board

- We used this board at SPRACE for development
  - https://www.96boards.org/product/ultra96/
- Zynq Ultrascale+ ZU3EG
  - Same SoC family as ZynqMP Mgmt. Module
  - 4 Cortex A-53 + 2 Cortex R-5 + PL and peripherals
  - Plenty of tutorials & Vivado support
  - Excellent price (249$)
    - More boards for devs working on the project
- Why did we use HAL from the start?
  - IO config may vary considerably between boards
  - The HAL makes porting much easier



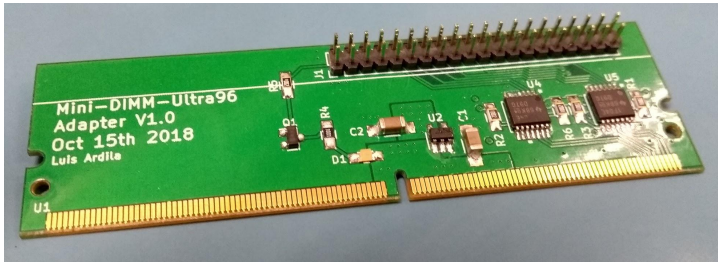


AVNET Ultra96 (Xilinx Zynq UltraScale+ ZU3EG)
A very popular 250$ Zynq dev board targeting makers

Reference development platform: full setup

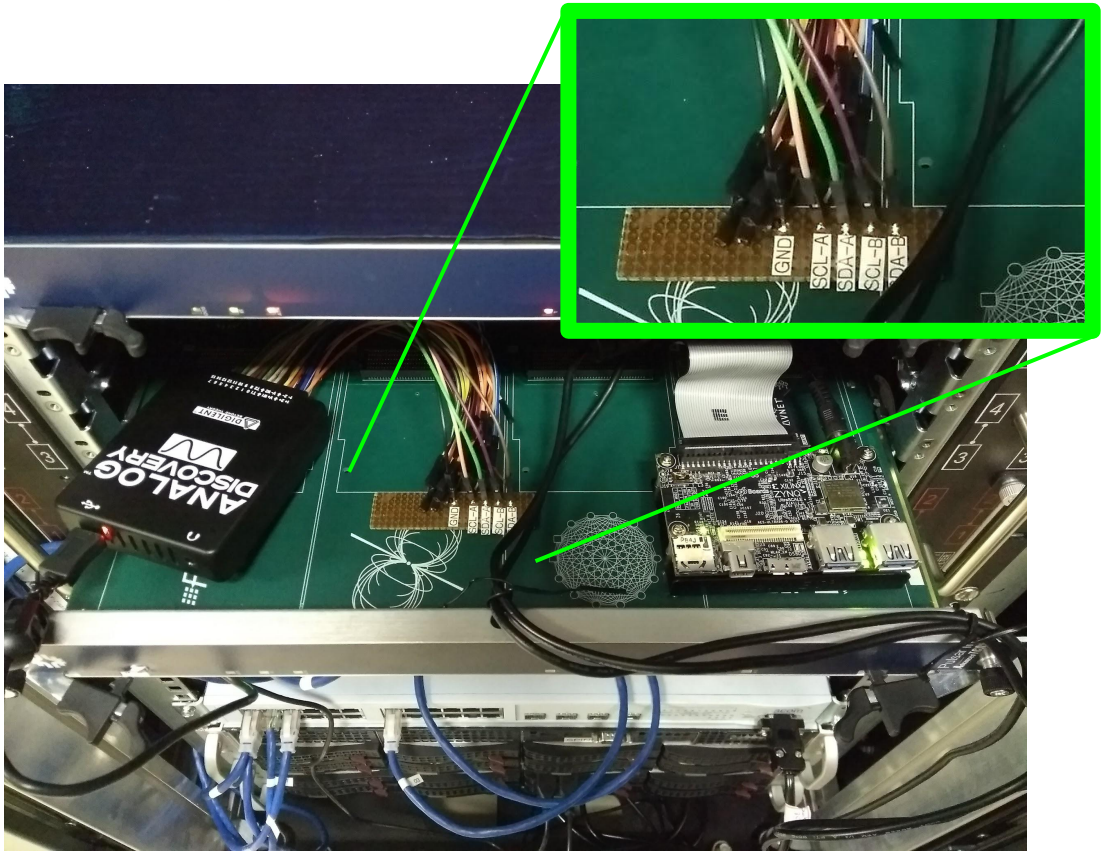
- Ultra96 Dev board
- Pulsar-2b ATCA board
  - Exposes the IPMB-A and -B buses to DIMM slot
  - We are currently not reading P2B sensors
- Mini-DIMM adapter
  - Connects Pulsar-2b DIMM slot to Ultra96
  - Translates 1.8 V (Ultra96)↔3.3V (ATCA)
  - Design and manufacture by Luis Ardila (KIT)
- Comtel CO6 ATCA chassis
  - Full-mesh, 6 slots horizontal
  - 2 PigeonPoint ShelfManagers (redundant)

Ultra96 mated to the Pulsar-II through Mini-DIMM adapter



Mini-DIMM adapter

Development Platform: Ultra96



Digilent Analog Discovery USB o'scope as I2C logic analyzer

Monitoring IPMB-A with TeK o'scope

OpenIPMC tests on Trenz + Serenity setup @ KIT

- OpenIPMC code was successfully ported and tested in **Trenz+Serenity** setup at KIT
- All changes relays into the `ipmc_ios.c` file: the physical layer of OpenIPMC
- Hot-Swap operation were successfully performed on Serenity board
- Since no real sensor are currently being read in this hardware

Trenz-Serenity setup at KIT

OpenIPMC tests on Trenz + Serenity setup @ KIT

Activation Status

```
# clia fru -v 96

Pigeon Point Shelf Manager Command Line Interpreter

96: FRU # 0
    Entity: (0xb0, 0x1)
    Hot Swap State: M4 (Active), Previous: M3 (Activation In Process), Last State Change Cause: Normal State Change (0x0)
    Device ID String: "Trenz-Serenity"
    Site Type: 0x00, Site Number: 02
    Current Power Level: 0x02, Maximum Power Level: 0x02, Current Power Allocation: 100.0 Watts
```

FRU Information (testing data from example code)

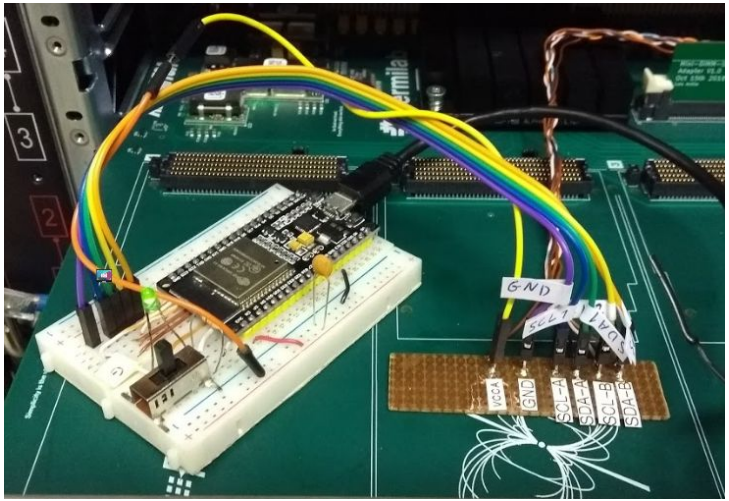
```
# clia fruinfo 96 0

Pigeon Point Shelf Manager Command Line Interpreter

96: FRU # 0, FRU Info
Common Header:    Format Version = 1

Board Info Area:
    Version     = 1
    Language Code           = 25
    Mfg Date/Time           = Oct  1 00:00:00 2019 (12490560 minutes since 1996)
    Board Manufacturer      = SPRACE - KIT
    Board Product Name      = OpenIPMC @ Trenz-Serenity
    Board Serial Number     = 189189981-18998
    Board Part Number       = AA00Y99
    FRU Programmer File ID  = 01
```

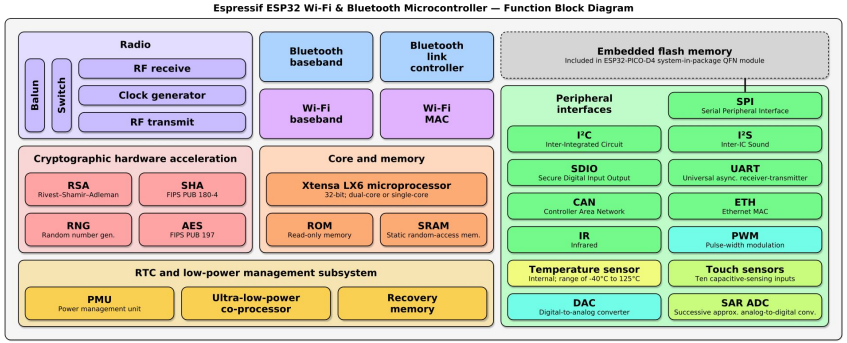
Sensor Reading (testing data from example code)

```
# clia sensordata 96 3

Pigeon Point Shelf Manager Command Line Interpreter

96: LUN: 0, Sensor # 3 ("FPGA TEMP")
    Type: Threshold (0x01), "Temperature" (0x01)
    Belongs to entity (0xa0, 0x60)
    Status: 0xc0
        All event messages enabled from this sensor
        Sensor scanning enabled
        Initial update completed
    Raw data: 50 (0x32)
    Processed data: 50.000000 degrees C
    Current State Mask: 0x00
```

OpenIPMC portability exercise: ESP32

- ESP32 microcontroller (see backup slides)
  - Very different from a Zynq US+
- Questions answered by this test
  - Architecture independency?
    - Trivial, thanks to C and FreeROTS
  - Ease of integration on a different SoC?
    - OpenIPMC needs I2C peripheral
    - Many SoCs have 2 or more
  - Effort needed to port OpenIPMC?
    - Mainly IO/HAL interface bindings
    - Fixes needed in ESP32 IDF (see backup)
    - Porting took just 3 person-weeks :-)
- Overall the exercise was a success
- Repo: gitlab.com/openipmc/ipmc-esp32

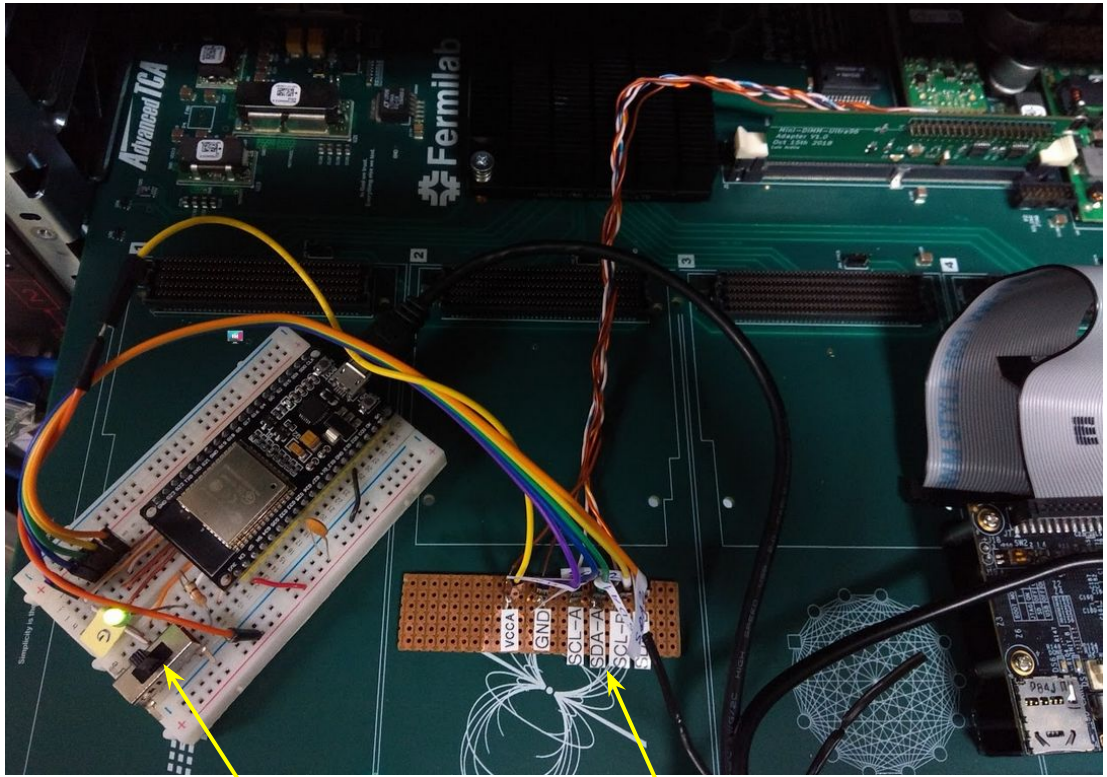




OpenIPMC tests on ESP32

- IPMBus communication works
- ShM happily accepts the FRU
- Activation and deactivation are triggered using an ‘improvised’ Handle Switch
- The activation time is significantly longer when compared with the operation using the Ultra96 board

```
86: LUN: 0, Sensor # 4 ("AIR TEMP")
    Type: Threshold (0x01), "Temperature" (0x01)
    Belongs to entity (0xb0, 0x60)
    Status: 0xc0
        All event messages enabled from this sensor
        Sensor scanning enabled
        Initial update completed
    Raw data: 32 (0x20)
    Processed data: 32.000000 degrees C
    Current State Mask: 0x00

86: LUN: 0, Sensor # 5 ("VCC1V0 VOUT")
    Type: Threshold (0x01), "Voltage" (0x02)
    Belongs to entity (0xb0, 0x60)
    Status: 0xc0
        All event messages enabled from this sensor
        Sensor scanning enabled
        Initial update completed
    Raw data: 22 (0x16)
    Processed data: 0.345400 Volts
    Current State Mask: 0x00

# clia fru 86

Pigeon Point Shelf Manager Command Line Interpreter

86: FRU # 0
    Entity: (0x0, 0x0)
    Hot Swap State: M1 (Inactive), Previous: M6 (Deactivation In Progress)
    Device ID String: " "
```



‘Handle Switch’ for tests

I2C lines connected to DIMM

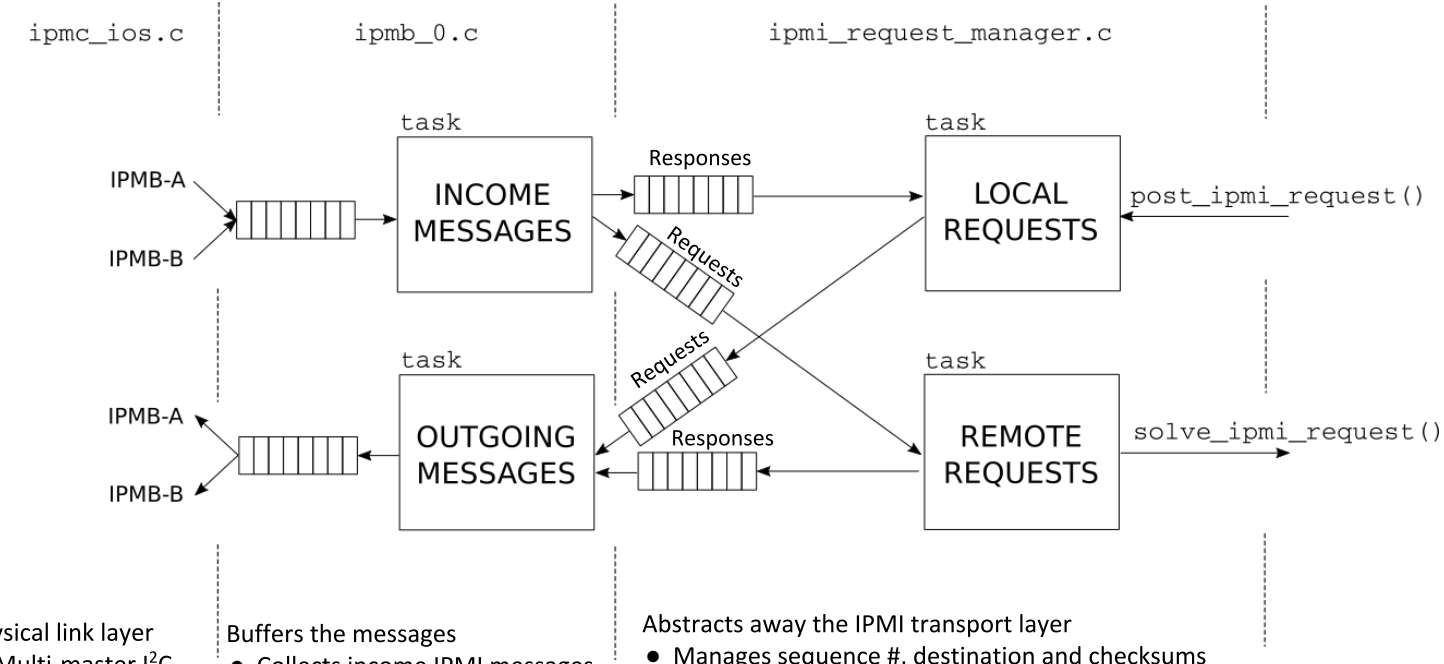
Summary

- ZynqMP Management Module requires customized IPMC software
  - Pigeon Point provides a commercial IPMC solution
  - OpenIPMC is a FOSS solution for the same application
- OpenIPMC project
  - Portable to any architecture with FreeRTOS support (w/ I2C periph.)
  - Fully customizable, simple, easy to debug
  - Templates for sensor declaration in the examples
  - GitLab repository here and Doxygen documentation here
- Successfully tested on 3 different hardwares
  - Ultra96 (Zynq UltraScale+, ARM Cortex-R5)
  - Serenity board with Trenz TE0803 mezzanine (Zynq UltraScale+, ARM Cortex-R5)
  - ESP32 (Tensilica Xtensa LX6)
  - Can be easily ported to other archs (SmartFusion? MSP430? STM32? LPC1700? ...)

Questions?

Backup Slides

OpenIPMC Internal Concept



"Application layer"

- Hot Swap operation
- Power Negotiation
- Sensor Records
- Sensor Readings
- Other IPMI functions
  - user-implemented

Physical link layer

- Multi-master I^2C
- Use 2 PS I^2C ctrlers

Buffers the messages

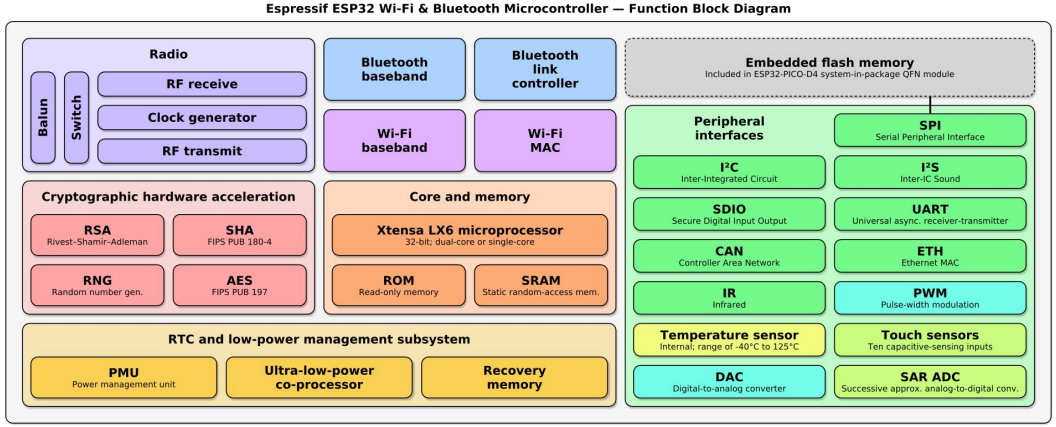
- Collects income IPMI messages (Requests and Responses)
- Manage the transmission over the IPMB channels (arbitration)

Abstracts away the IPMI transport layer

- Manages sequence #, destination and checksums
- Associates responses to requests
- Retries and timeouts
- Accepts multiple internal requests (from different tasks)
- Call the specific functions to solve external requests

OpenIPMC on ESP32 (Espressif Systems, CN)

- Quite powerful & flexible uC
  - 240 MHz Xtensa LX6 dual core
  - FPU, Big INTs & Crypto
  - WiFi, BT, SPI, I2C, UART…
  - **FreeRTOS support**
- Cheap Linux-supported boards
  - CP2102 USBtoUART converter
  - Boards sell for 5$
- 3.3 V device (same as IPMB)
- Development software
  - Arduino IDE, PlatformIO or **esp-idf**
- Very different arch w.r.t a Zynq US+
  - **Good exercise on portability/**

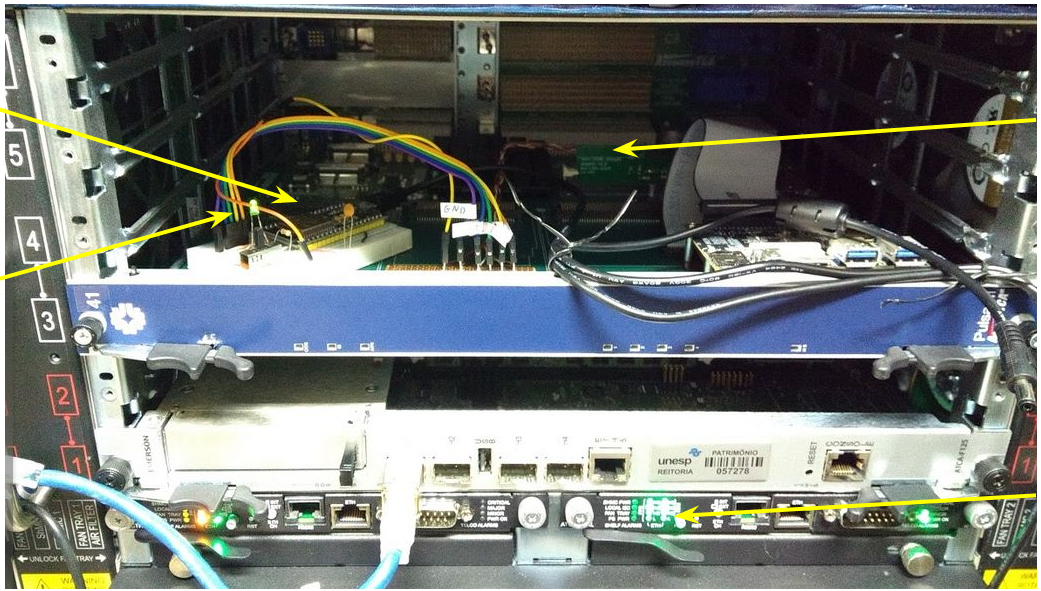


Fixes to ESP32 Integrated Development Framework

- I2C multi-master with variable size I2C msgs is required for IPMB bus communication
  - End of message is signalled by a stop bit
- The official esp-idf I2C driver was not supporting variable size msgs correctly
  - The driver expected a message size to be specified in advance
- We modified the driver* and now the slave read function correctly returns if receiving a stop bit

ESP32 w/ OpenIPMC

‘Blue LED’



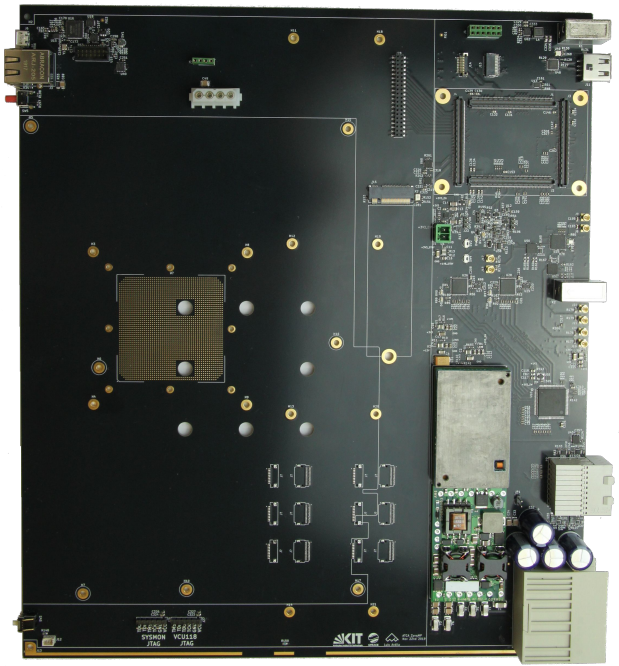
DIMM adapter giving access to the backplane

Shelf Manager

*The github esp-idf repository was forked. A merge request is still in progress

ZynqMP-IPMC ATCA Test Board

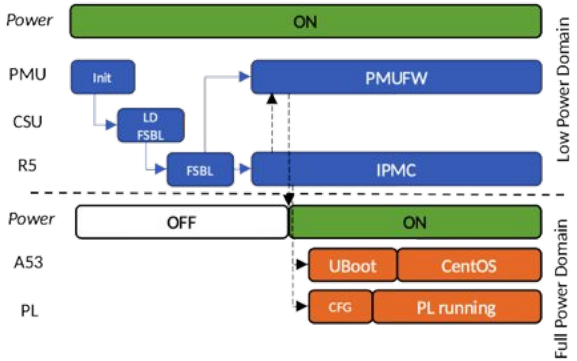
- modified stencil + glue allowed all TPS82130 to operate as expected
- one PCB fully assembled and operational used to bring up vivado project for future OpenIPMC & PP-IPMC software migration
- two more PCBs assembled, but one not working, needs rework on problematic regulators



https://gitlab.cern.ch/p2-xware/hardware/kicad-pcbs/atca/atca_zynqmp

Pigeon Point - Performed Tests

- IPMC
  - Power-up
  - Com. to shelf manager
  - State change to M4
  - Sensor readings
  - Activation/Deactivation
  - Handle Switch
  - Cold Reset
  - Trigger Linux boot



- IPMC compliance test
  - 137 tests executed (most automatic)
  - failing tests were expected to fail because of missing functionality

Pass	Fail	Error	Abort	Skip	Inconclusive
79	34	0	0	24	0

- Linux
  - Start U-boot
  - Network stack in U-boot
  - Load Linux kernel via tftp
  - Start Linux (login shell)
  - Rootfs
    - Initramfs
    - NFS
    - Sdcard
  - Network connection
  - SSH to ZynqMP
  - SSH from ZynqMP
  - CENTOS rootfs
  - Yum install