CERN-IPMC: Introduction

ATLAS Upgrade - IPMC Workshop

CERN EP-ESE-BE

Julian Mendez

OUTLINE

- CERN IPMC – Introduction
  - xTCA evaluation project – general overview
  - CERN-IPMC: specifications
  - CERN-IPMC: hardware
  - CERN-IPMC: software configuration
  - Demo: introduction of the CERN-IPMC environment

- CERN-IPMC – Demonstration
  - Introduction
  - Adding a simple temperature sensor (LM75): driver and instantiation
  - Case of a multi-purpose sensor (LTC2990)
  - How does a sensor driver work?

OUTLINE

- CERN IPMC – Introduction
  - xTCA evaluation project – general overview
  - CERN-IPMC: specifications
  - CERN-IPMC: hardware
  - CERN-IPMC: software configuration
  - Demo: introduction of the CERN-IPMC environment
- CERN-IPMC – Demonstration
  - Introduction
  - Adding a simple temperature sensor (LM75): driver and instantiation
  - Case of a multi-purpose sensor (LTC2990)
  - How does a sensor driver work?

OUTLINE

- CERN IPMC – Introduction
  - **xTCA evaluation project – general overview**
  - CERN-IPMC: specifications
  - CERN-IPMC: hardware
  - CERN-IPMC: software configuration
  - Demo: introduction of the CERN-IPMC environment
- CERN-IPMC – Demonstration
  - Introduction
  - Adding a simple temperature sensor (LM75): driver and instantiation
  - Case of a multi-purpose sensor (LTC2990)
  - How does a sensor driver work?

xTCA EVALUATION PROJECT – GENERAL OVERVIEW

- **xTCA evaluation project launched in 2011 in EP-ESE**
- **Main objectives:**
  - Technical evaluation of components for **MTCA**, **MTCA.4** and **ATCA** systems
  - Technical evaluation of **AC/DC converters**
  - Development of tools (H/W and S/W) for the testing of commercial components
  - Conduct market surveys
  - Report and share results
- **Longer term goal**
  - Standardize MTCA and ATCA shelves and power supplies
    - Many options (backplanes, cooling, RTMs, power supply, …)
  - Propose acceptance test procedures
  - Propose a selected set of equipment to the experiments
  - **Provide centralized support for these items**

xTCA EVALUATION PROJECT – GENERAL OVERVIEW

- Important number of different equipment evaluated, both in MTCA and ATCA
- Modules were selected and purchase contract were established
- **An MMC module was designed for the AMC cards (mTCA)**

MTCA Crates

Vadatech VT892

ELMA 043-012 MTCA.4

Schroff MTCA.4 + AC/DC CM100

Schroff MTCA.4

PMs

NAT DC780

Vadatech UTC010

Wiener AC/DC

MCHs

Vadatech UTC001

NAT MCH

Kontron AM4904

AMCs

ELMA Load Board

Kontron AM5030

ESD ADIO24

CCT AM31

ATCA Crates

Schroff 14-slot

ASIS 14-slot

ATCA boards

Kontron AMC Carrier

Comtel Load blade

AC/DC Power Supply

Emerson NetSure501

PowerOne Aspiro

PowerOne Guardian

Lineage Power CPS

xTCA EVALUATION PROJECT – GENERAL OVERVIEW

- **An IPMC project was launched in 2015 – Based on the commercial solution from Pigeon Point System**

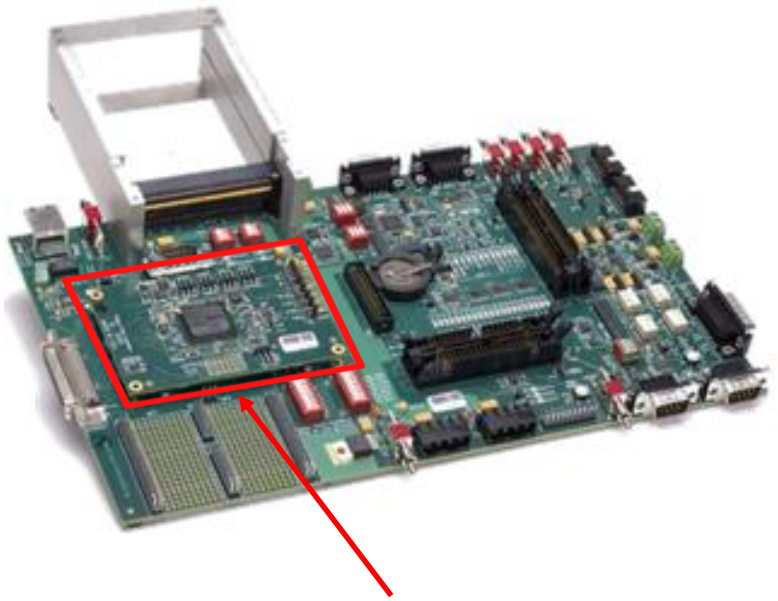
- Pigeon Point:
  - Part of the Pentair group
  - Provide solutions for the Hardware Platform Management modules:
    - Module Management Controller (MMC)
    - Intelligent Platform Management Controller (IPMC)
    - Shelf Manager (ShMM)
  - Pigeon Point shelf manager is used by almost all of the shelf manufacturers

- A license for the IPMC solution was acquired by CERN EP-ESE-BE. It includes:
  - Documentation
  - Development tools
  - Hardware, firmware and software reference designs

xTCA EVALUATION PROJECT – GENERAL OVERVIEW

- Pigeon solution was evaluated in 2015 using:
  - The starter kit IPMC mezzanine card
  - An Adapter card (CERN) to use the mezzanine with existing ATCA blade

- Evaluation outcomes:
  - ✓ Standard compliance (Polaris tester)
  - ✓ Management of the AdvancedTCA blade
  - ✓ Management of up to 4 AMCs
  - ✓ E-Keying and Clock configuration
  - ✓ OEM commands
  - ✓ Non-intelligent RTM
  - ✓ Ethernet interface
  - ✓ Remote upgrade (HPM.1)



Pigeon Point IPMC mezzanine

xTCA EVALUATION PROJECT – GENERAL OVERVIEW

- Considering:
  - The positive outcome of the PP IPMC evaluation (ensures the compliance with the standard)
  - The reasonable price of the solution

Decision was taken to develop an IPMC mezzanine based on the Pigeon Point commercial solution.

- Development project was started in 2016

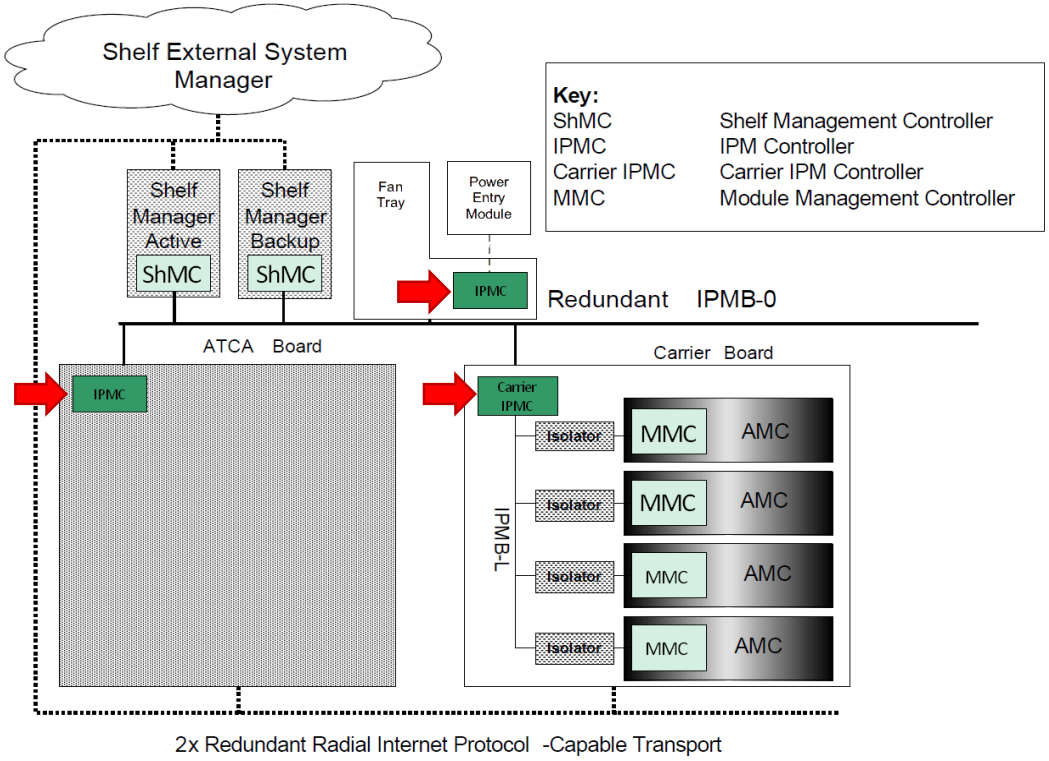
- Key considerations:
  - Compatibility with already designed ATCA blades:
    - Use the same form factor as the IPMC already developed by LAPP
    - Stay pin compatible with the existing IPMC
  - Adapt the Pigeon Point solution to be General purpose and easily customizable

OUTLINE

- CERN IPMC – Introduction
  - xTCA evaluation project – general overview
  - **CERN-IPMC: specifications**
  - CERN-IPMC: hardware
  - CERN-IPMC: software configuration
  - Demo: introduction of the CERN-IPMC environment
- CERN-IPMC – Demonstration
  - Introduction
  - Adding a simple temperature sensor (LM75): driver and instantiation
  - Case of a multi-purpose sensor (LTC2990)
  - How does a sensor driver work?

CERN-IPMC: SPECIFICATIONS

- Role of the Intelligent Controller for AdvancedTCA blades:
  - Monitoring sensors (Voltages, temperatures …)
  - Controlling the system (Power management, port/clock activation …)
  - Ensuring proper operations (Compatibility between the boards, hot swap, redundancy …)



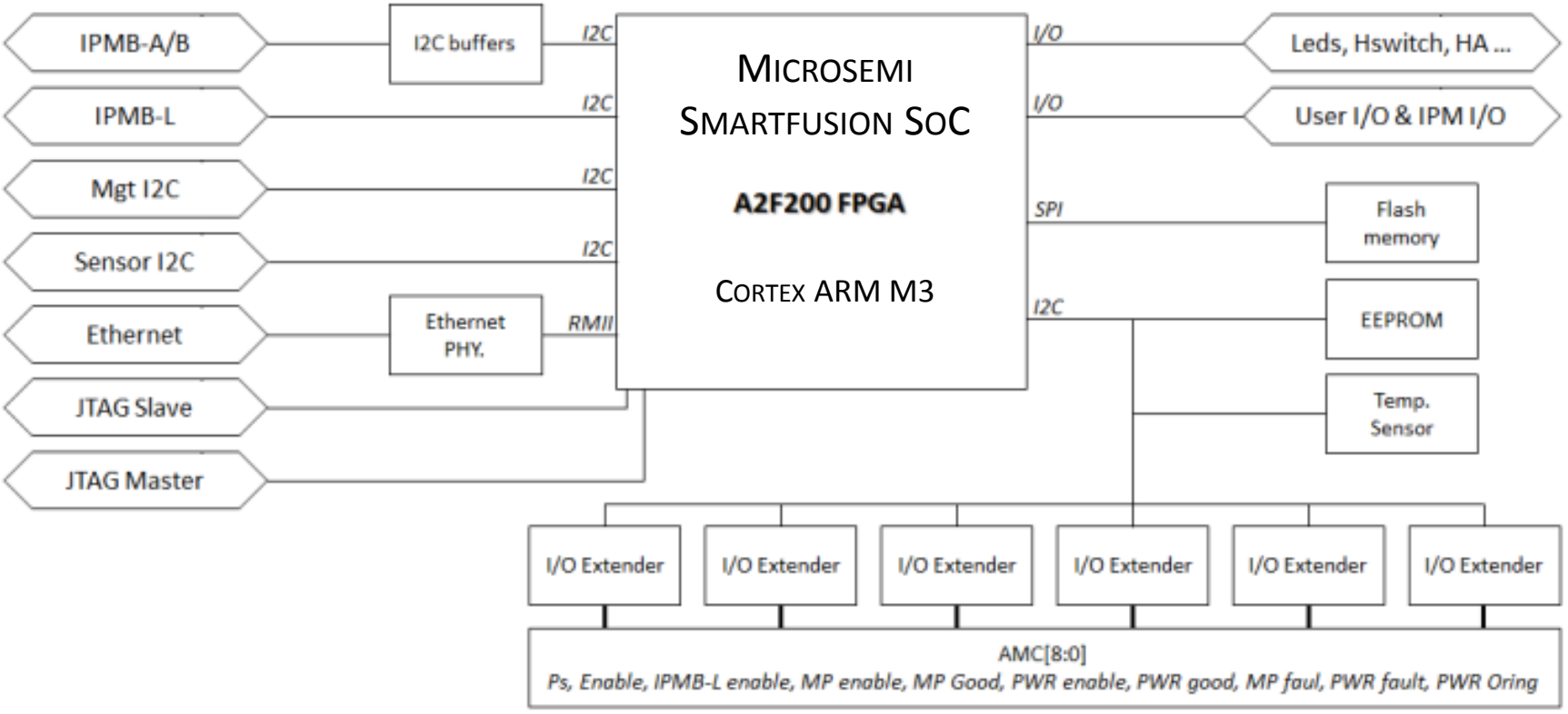
CERN-IPMC: SPECIFICATIONS

FULLY TESTED

- Supported features:
  - AdvancedTCA Rev.3.0 standard
    - Hot swap (FRU info., handle switch, LEDs, Hardware address, etc.)
    - Sensor monitoring (SDR, measurement, events, etc.)
    - Rear Transition Module (intelligent and non-intelligent RTM)
  - AMC standard (up to 9 AMCs – including iRTM)
  - Ethernet interface (RMCP/RMCP+, UDP, simplified TCP/IP)
  - Serial interface (SoL or debug interface)
  - I2C buses for sensor monitoring and power management
  - User I/Os (35 User I/Os + 16 IPM I/Os)
  - JTAG Master (Xilinx Virtual Cable daemon)

CERN-IPMC: SPECIFICATIONS

- Functional block diagram:

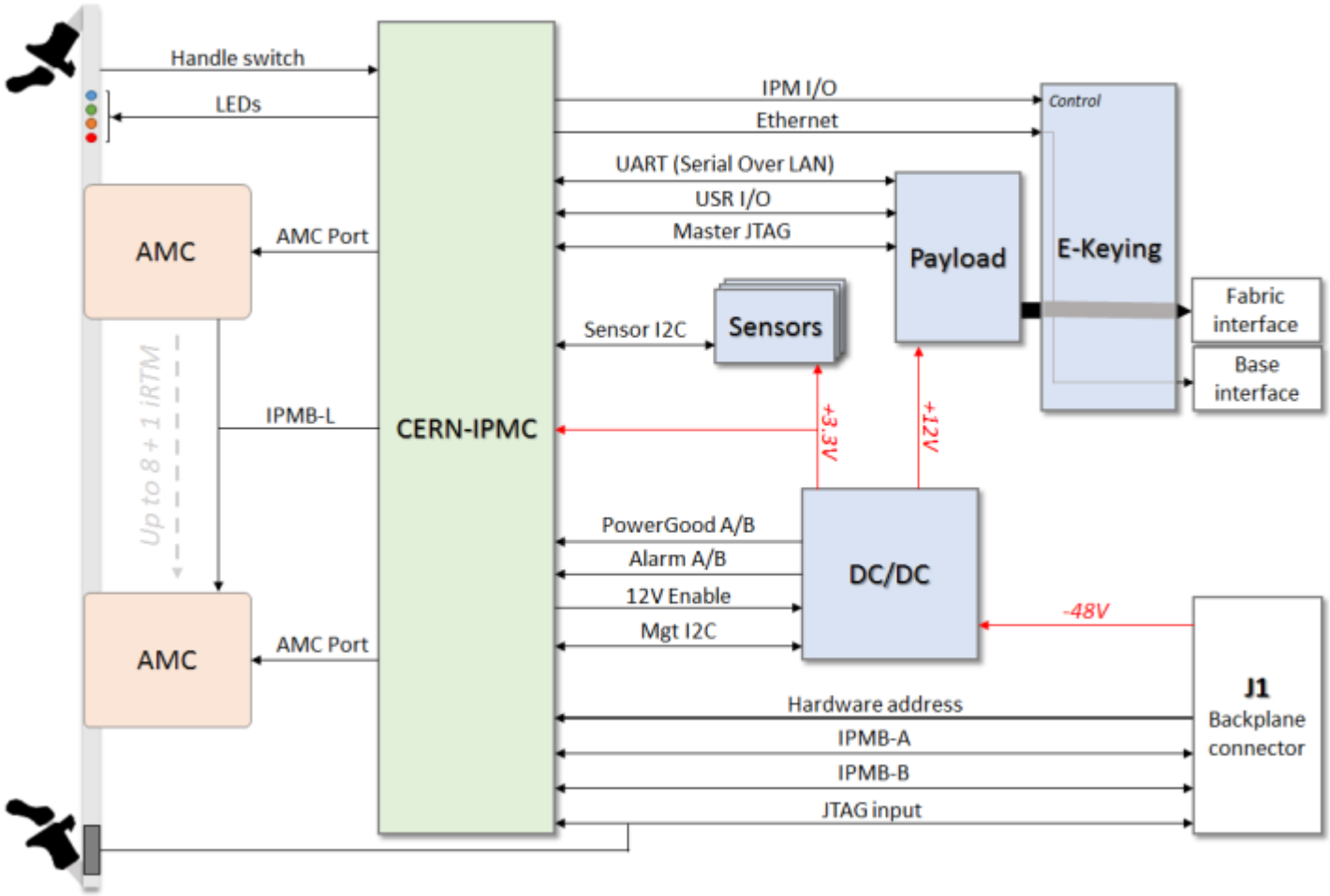


OUTLINE

- CERN IPMC – Introduction
  - xTCA evaluation project – general overview
  - CERN-IPMC: specifications
  - **CERN-IPMC: hardware**
  - CERN-IPMC: software configuration
  - Demo: introduction of the CERN-IPMC environment
- CERN-IPMC – Demonstration
  - Introduction
  - Adding a simple temperature sensor (LM75): driver and instantiation
  - Case of a multi-purpose sensor (LTC2990)
  - How does a sensor driver work?

CERN-IPMC: HARDWARE

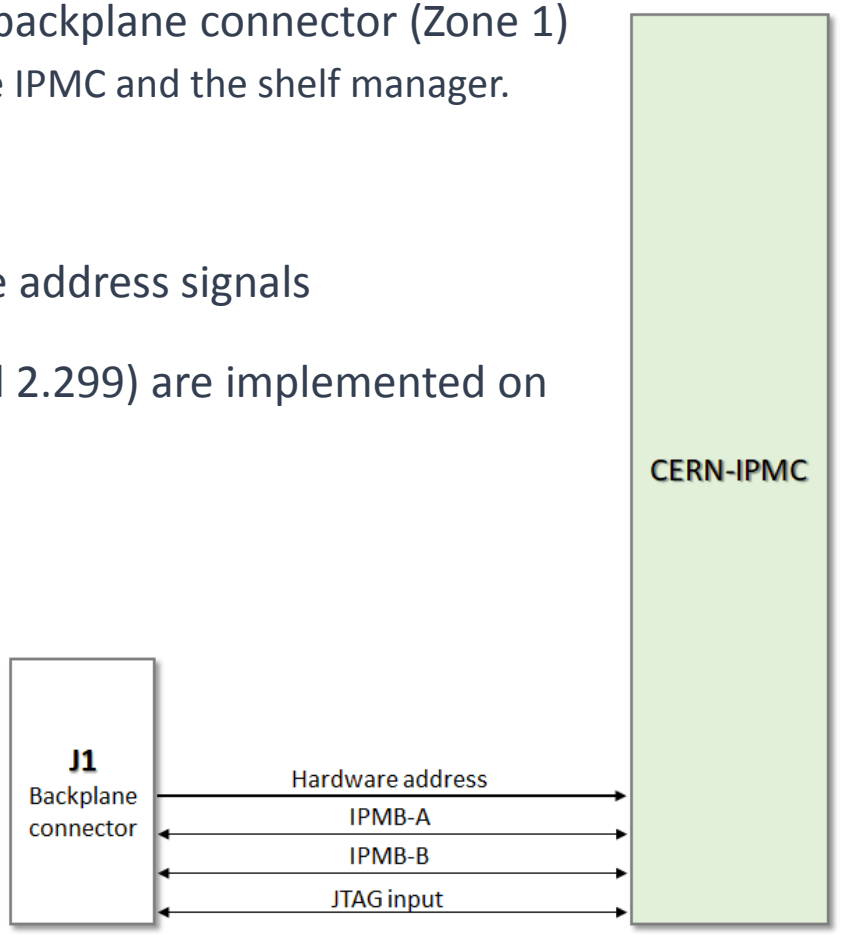
- Typical use of the CERN-IPMC



CERN-IPMC: HARDWARE

- Management interface comes from the ATCA backplane connector (Zone 1)
  - Provides all connections required between the IPMC and the shelf manager.
- Direct routing from IPMC to zone 1 connector.
- Capacitors and resistors required for hardware address signals
- I2C buffers for IPMB-A/B buses (REQ2.298 and 2.299) are implemented on the mezzanine.

IPMC	Backplane	Designation	Description	Explanation
115	5	HA0	HA0 Hardware Address Bit 0	Hardware Address Bits read by the IPMC
237	6	HA1	HA1 Hardware Address Bit 1	
116	7	HA2	HA2 Hardware Address Bit 2	
238	8	HA3	HA3 Hardware Address Bit 3	
117	9	HA4	HA4 Hardware Address Bit 4	
239	10	HA5	HA5 Hardware Address Bit 5	
118	11	HA6	HA6 Hardware Address Bit 6	
240	12	HA7/P	HA7/P Hardware Address Bit 7 (Odd Parity Bit)	
120	13	SCL_A	IPMB Clock, Port A	IPMB Connection Port A
121	14	SDA_A	IPMB Data, Port A	
242	15	SCL_B	IPMB Clock, Port B	IPMB Connection Port B
243	16	SDA_B	IPMB Data, Port B	



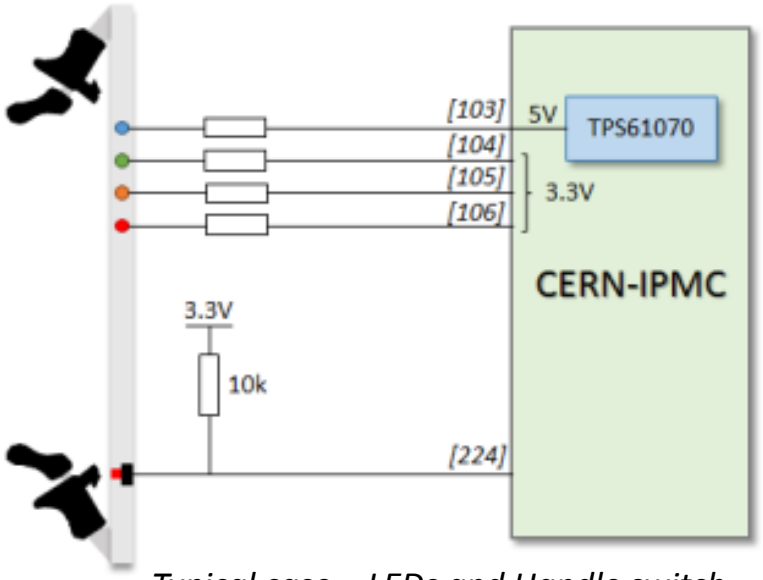
Typical case – Zone 1 connection

CERN-IPMC: HARDWARE

- Handle switch is active low by default.
  - Power up sequence starts when signal is at GND (10k pull-up required).
  - Active state can be easily modified in the software.

- Blue LED is driven through a TPS61070 booster (5V)

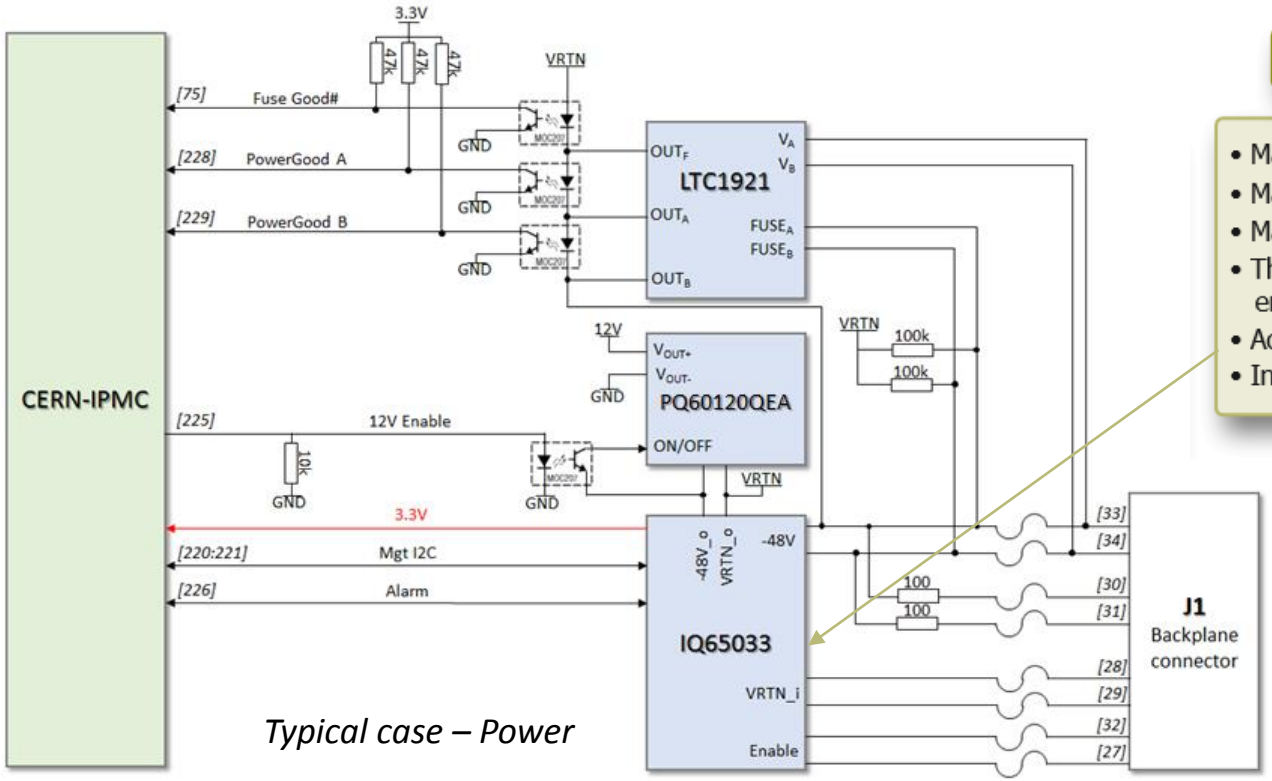
- Additional LEDs are driven by the controller:
  - IO at 3.3V
  - Maximum current: 30mA



Typical case – LEDs and Handle switch

CERN-IPMC: HARDWARE

- ATCA blade power management ensured by specific pins and Mgt I2C bus.
- Additional power management devices can be added to protect against over-v./cur.
  - Already protected by FUSES and IQ65033 module
  - PowerGood signals monitoring can be disabled in the software



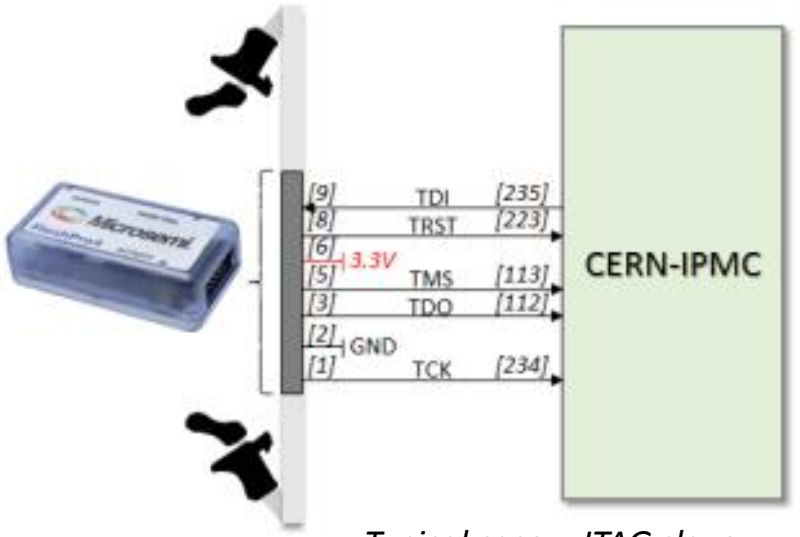
Typical case – Power

Protection Features

- Management power over-voltage protection
- Management power over-current protection
- Main output over-current protection
- Thermal shutdown protects the unit from abnormal environmental conditions
- Active back bias limit
- Input fuse/feed loss alarm

CERN-IPMC: HARDWARE

- JTAG to IPMC:
  - JTAG slave interface can be used to configure IPMC FPGA using a FlashPro4 JTAG programmer.
  - E.g.: connection to a front panel connector



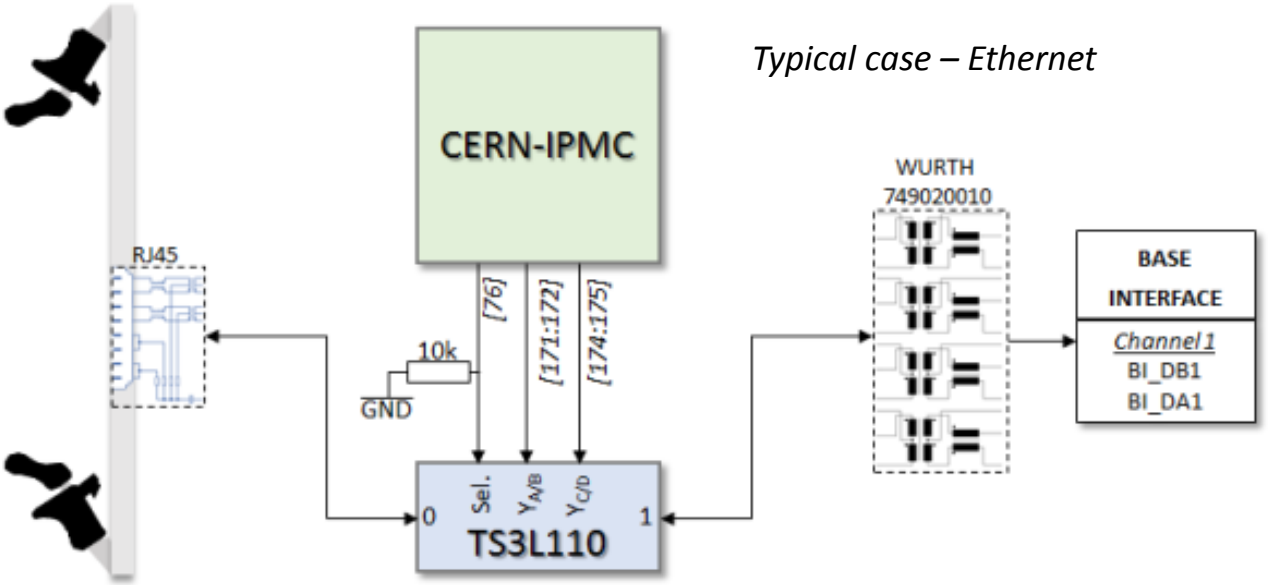
Typical case – JTAG slave

- JTAG Master:
  - JTAG master interface is made of GPIO running JTAG commands at 5MHz.
  - Connectivity between the ATCA on-board devices and the IPMC is user's responsibility.
  - Directly routed from the CERN-IPMC controller without any pull-up/down nor capacitors.

CERN-IPMC: HARDWARE

- The Ethernet lines available on the IPMC are ready to be connected to:
  - RJ45 connector
  - To a switch (e.g. TS3L110) for dual connectivity (E.g.: front panel or backplane).

- Required pull-ups and PHY transceiver are located on the IPMC mezzanine

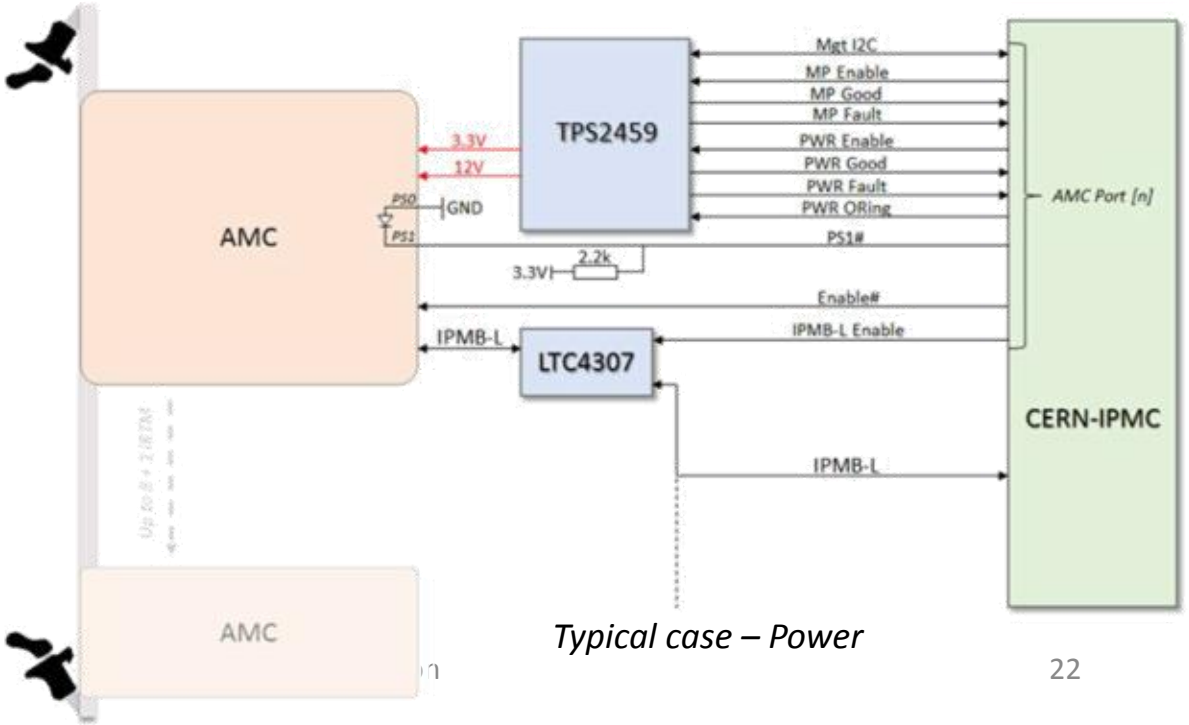


CERN-IPMC: HARDWARE

- UART port:
  - Connection to the payload is recommended:
    - Sending alerts (e.g.: sensor events)
    - Sending configuration (e.g.: using PI intf or Serial Over Lan)
  - Using the UART for the Debug interface (optional)

CERN-IPMC: HARDWARE

- Support of up to 8 AMCs + 1 iRTM
- I2C buffers must be added for each slot on the IPMB-L bus
  - Prevent communication issues upon insertion/extraction
- Supported power management devices:
  - TPS2358/2458
  - TPS2459
  - LTC4222



Typical case – Power

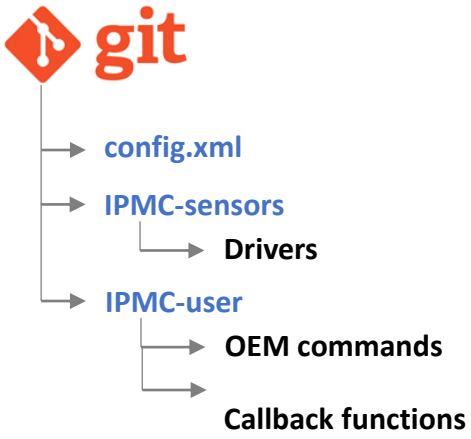
OUTLINE

- CERN IPMC – Introduction
  - xTCA evaluation project – general overview
  - CERN-IPMC: specifications
  - CERN-IPMC: hardware
  - **CERN-IPMC: software configuration**
  - Demo: introduction of the CERN-IPMC environment
- CERN-IPMC – Demonstration
  - Introduction
  - Adding a simple temperature sensor (LM75): driver and instantiation
  - Case of a multi-purpose sensor (LTC2990)
  - How does a sensor driver work?

CERN-IPMC: SOFTWARE

- **No NDA needed.**
  - Project can be customized and compiled remotely: core source code is kept private.

- Project architecture:



+

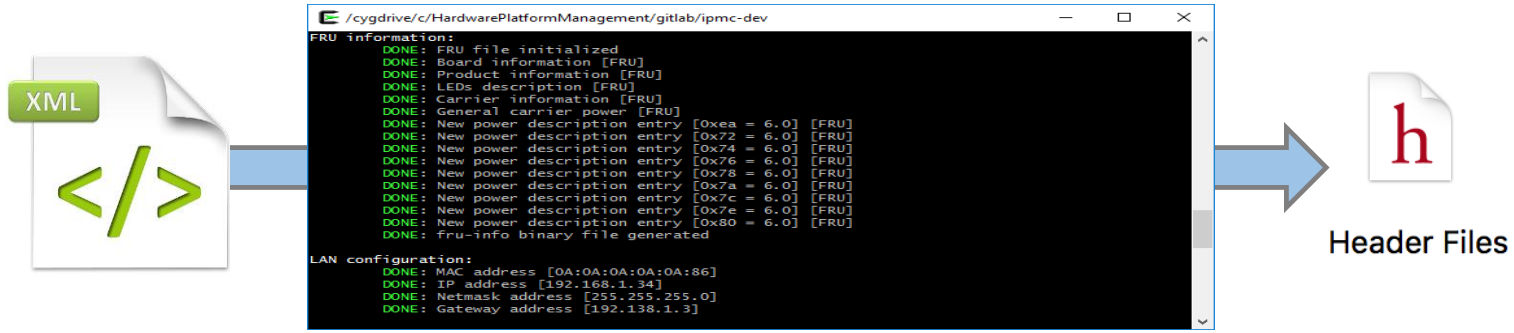
compile.py script

- Gitlab URL: https://gitlab.cern.ch/ep-ese-be-xtca/ipmc-project

CERN-IPMC: SOFTWARE

- **Why does the software have to be modified?**
  - User customizable features:
    - FRU Information (Device ID, Manufacturer info., Product info.)
    - LAN (MAC address, Default IP, slot specific IP, DHCP, Gateway, Netmask)
    - Modules (AMCs, iRTM/Non-intelligent RTM)
    - Sensors
    - E-Keying
    - Power sequencing

- Almost all of the modifications can be done using the config.xml file
  - Documented on git: https://gitlab.cern.ch/ep-ese-be-xtca/ipmc-project



CERN-IPMC: SOFTWARE

- General configuration:
  - Device details
  - Board details
  - Power information
  - Handle switch states
  - Serial interface configuration

```
<GeneralConfig>

    <DeviceID>0x12</DeviceID>
    <DeviceRevision>0x00</DeviceRevision>
    <ManufacturerID>0x000060</ManufacturerID>
    <ProductID>0x1236</ProductID>

    <ManufacturingDate>06/01/2017</ManufacturingDate>

    <BoardManuf>CERN</BoardManuf>
    <BoardName>CERN-IPMC</BoardName>
    <BoardSN>17170018</BoardSN>
    <BoardPN>P580050995</BoardPN>

    <ProductManuf>CERN</ProductManuf>
    <ProductName>IPMC-TestPAD</ProductName>
    <ProductPN>PP2018-33</ProductPN>
    <ProductSN>0010</ProductSN>
    <ProductVersion type="major">1</ProductVersion>
    <ProductVersion type="minor">20</ProductVersion>

    <MaxCurrent>10.0</MaxCurrent>
    <MaxInternalCurrent>1.0</MaxInternalCurrent>

    <!-- Hardware -->
    <HandleSwitch active="LOW" inactive="HIGH" />

    <ResetOnWrongHAEn />

    <SerialIntf>SDI_INTF</SerialIntf>
    <RedirectSDItoSOL/>

</GeneralConfig>
```

CERN-IPMC: SOFTWARE

- Power ON/OFF sequences:
  - List of step to be executed

```
<PowerManagement>

    <PowerONSeq>
        <step>PSQ_ENABLE_SIGNAL(CFG_PAYLOAD_DCDC_EN_SIGNAL)</step>
        <step>PSQ_END</step>
    </PowerONSeq>

    <PowerOFFSeq>
        <step>PSQ_DISABLE_SIGNAL(CFG_PAYLOAD_DCDC_EN_SIGNAL)</step>
        <step>PSQ_END</step>
    </PowerOFFSeq>

</PowerManagement>
```

- The steps can be:
  - PSQ_ENABLE_SIGNAL(signal_name)
  - PSQ_DISABLE_SIGNAL(signal_name)
  - PSQ_TEST_SIGNAL_JUMP_IF_SET(signal_name, offset):
  - PSQ_TEST_SIGNAL_JUMP_IFNOT_SET(signal_name, offset)
  - PSQ_SET_TIMER(timer, timeout)
  - PSQ_JUMP(offset)
  - PSQ_JUMP_IF_TIMEOUT(timer, offset)
  - PSQ_JUMP_IFNOT_TIMEOUT(timer, offset)
  - PSQ_SET_RECOVERY_POINT(address)
  - PSQ_END
  - PSQ_FAIL

- Signal name listed on the pinout documentation (here)

CERN-IPMC: SOFTWARE

- LAN interface configuration:
  - MAC Address
  - NetMask
  - Gateway

- IP can be set to be configured via DHCP or per slot (exclusive: IPAddrList or EnableDHCP)

```
<LANConfig>

    <MACAddr>0A:0A:0A:0A:0A:86</MACAddr>
    <NetMask>255.255.255.0</NetMask>
    <GatewayIP>192.138.1.3</GatewayIP>

    <!-- <EnableDHCP /> -->

    <IPAddrList> <!-- Default IP Addresses (used if DHCP is not active) -->
        <IPAddr slot_addr="default">192.168.1.34</IPAddr>
        <IPAddr slot_addr="0x41">192.168.1.20</IPAddr>
        <IPAddr slot_addr="0x42">192.168.1.21</IPAddr>
        <IPAddr slot_addr="0x43">192.168.1.22</IPAddr>
        <IPAddr slot_addr="0x44">192.168.1.23</IPAddr>
        <IPAddr slot_addr="0x45">192.168.1.24</IPAddr>
    </IPAddrList>

</LANConfig>
```

CERN-IPMC: SOFTWARE

- AMC/iRTM ports can be enabled and connected to specific IPMC ports

- E.g.: AMC slot 1 (0x72) is connected to IPMC port 1 with a max. current of 6 Amps.

```
<AMCSlots>

    <AMC site="1">
        <PhysicalPort>1</PhysicalPort>
        <MaxCurrent>6.0</MaxCurrent>
        <PowerGoodTimeout>300</PowerGoodTimeout>
        <DCDCEfficiency>85</DCDCEfficiency>
    </AMC>

</AMCSlots>
```

- Almost the same for iRTM except that the site number is replaced by the I2C address.

```
<iRTMSlot>
    <PhysicalPort>0</PhysicalPort>
    <MaxCurrent>6.0</MaxCurrent>
    <Address>0xea</Address>
    <PowerGoodTimeout>300</PowerGoodTimeout>
    <DCDCEfficiency>85</DCDCEfficiency>
</iRTMSlot>
```

CERN-IPMC: SOFTWARE

- Sensors can be instantiated using the config.xml file
  - Additional details during the “CERN-IPMC – Demonstration” part of the workshop

```
<SensorList>

    <Sensors type="MCP9801">

        <Sensor>
            <Name>Internal temp.</Name>

            <Type>Temperature</Type>
            <Units>degrees C</Units>

            <NominalReading>25</NominalReading>
            <NormalMaximum>60</NormalMaximum>
            <NormalMinimum>-10</NormalMinimum>

            <Point id="0" x="0" y="0" />
            <Point id="1" x="5" y="5" />

            <i2c_addr>0x090</i2c_addr>

            <Thresholds>
                <UpperNonRecovery>80</UpperNonRecovery>
                <UpperCritical>60</UpperCritical>
                <UpperNonCritical>40</UpperNonCritical>
                <LowerNonRecovery>-20</LowerNonRecovery>
                <LowerCritical>-10</LowerCritical>
```

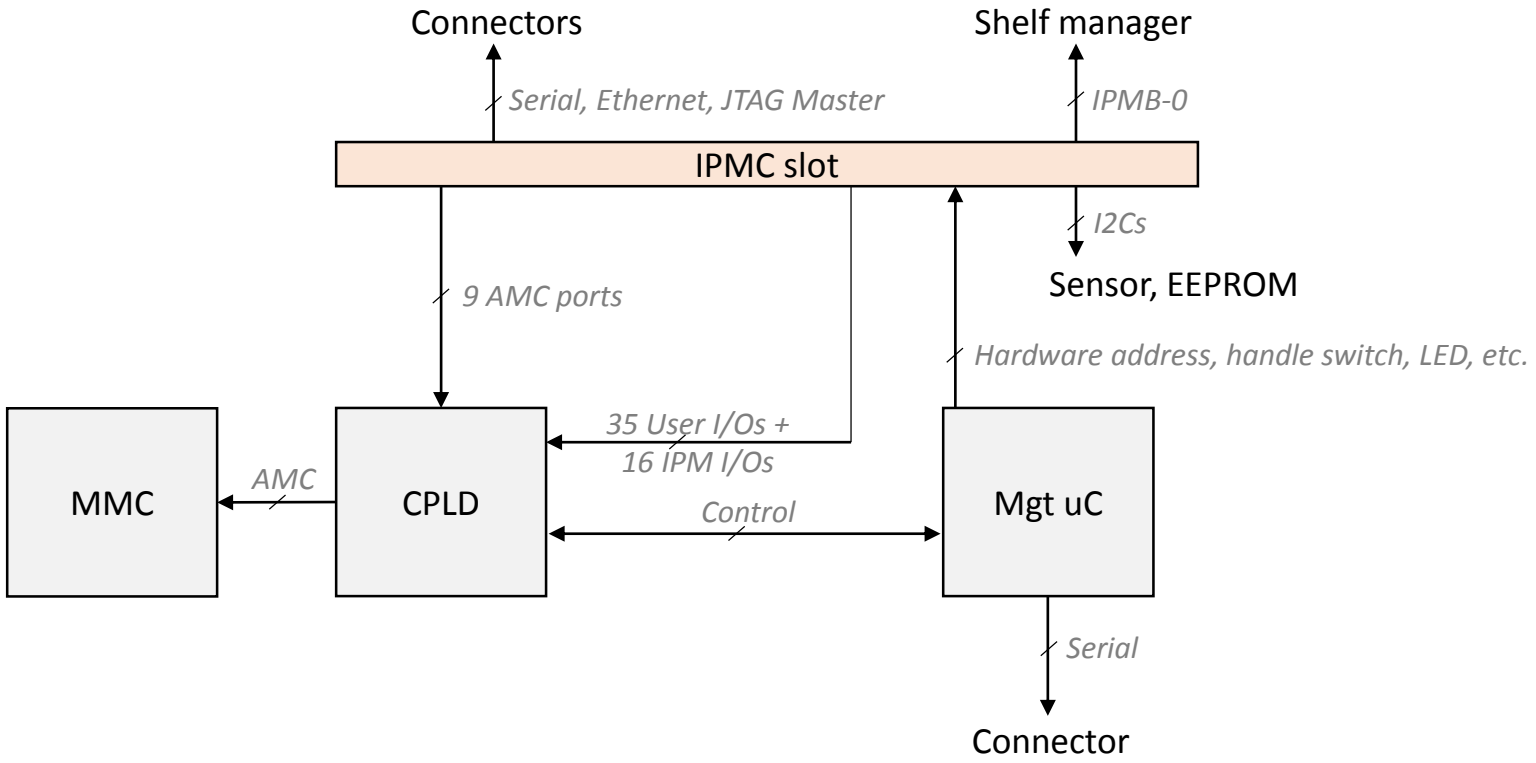
OUTLINE

- CERN IPMC – Introduction
  - xTCA evaluation project – general overview
  - CERN-IPMC: specifications
  - CERN-IPMC: hardware
  - CERN-IPMC: software configuration
  - **Demo: introduction of the CERN-IPMC environment**

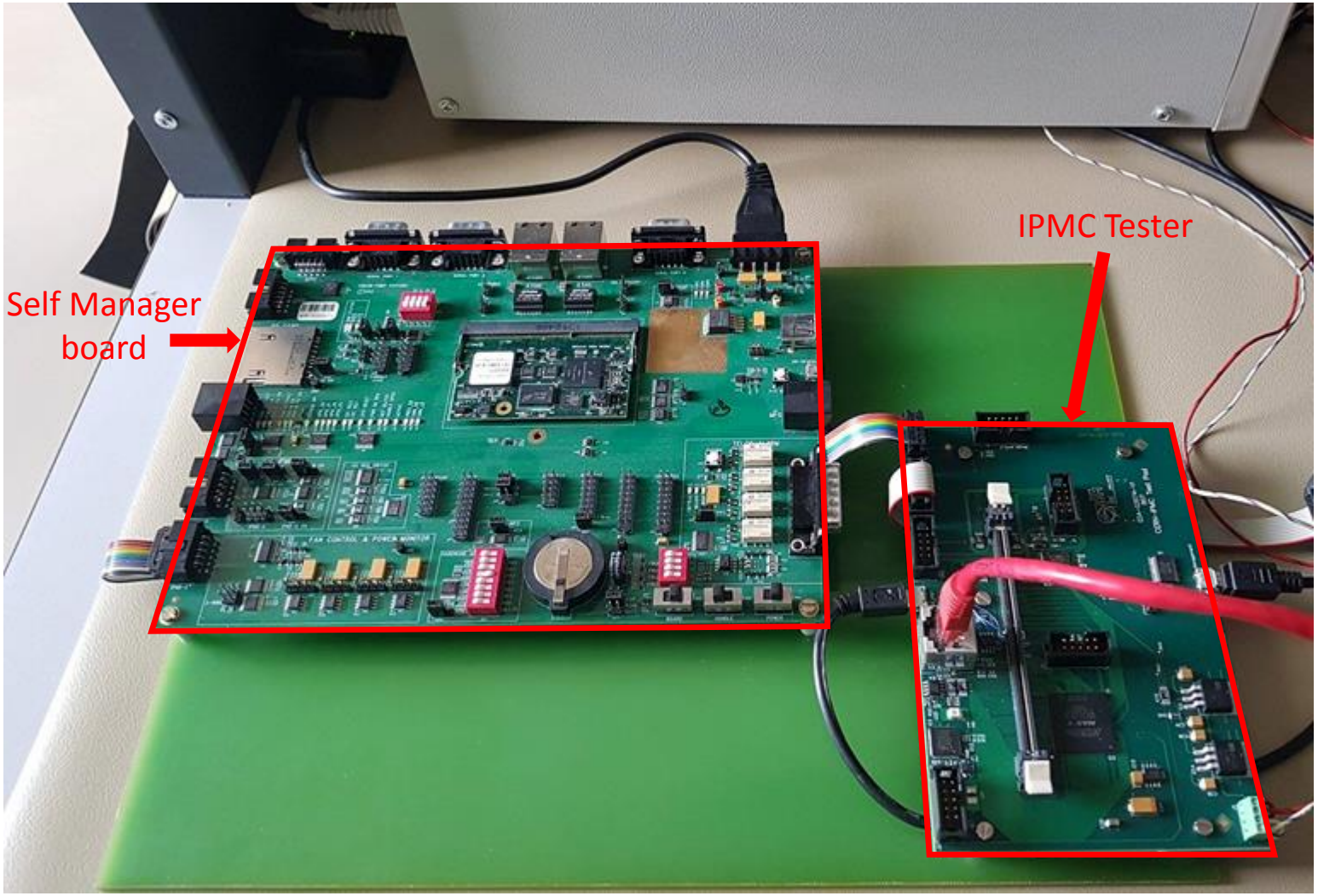
- CERN-IPMC – Demonstration
  - Introduction
  - Adding a simple temperature sensor (LM75): driver and instantiation
  - Case of a multi-purpose sensor (LTC2990)
  - How does a sensor driver work?

DEMO: INTRODUCTION OF THE CERN-IPMC ENVIRONMENT

- Based on the CERN-IPMC DevKit:



DEMO: INTRODUCTION OF THE CERN-IPMC ENVIRONMENT



DEMO: INTRODUCTION OF THE CERN-IPMC ENVIRONMENT

- Project creation
- Customize the CERN-IPMC software to run on the CERN-IPMC DevKit
- Compile the project
- Load the binary using HPM.1 (Ethernet)
- Run the CERN-IPMC test (using the CERN-IPMC DevKit)
- Configure an Artix 7 FPGA using the JTAG Master interface

- Documentation: https://gitlab.cern.ch/ep-ese-be-xtca/ipmc-project

Demo

SUMMARY

- NDA signature is not needed.
- CERN-IPMC is ready to be used and the package ready to be purchased.
- The CERN-IPMC kit includes:
  - The CERN-IPMC mezzanine card (200.- CHF)
  - Access to the ipmc-project repository (https://gitlab.cern.ch/ipmc-project)
  - Access to the CERN-IPMC WebApp (https://cern-ipmc.web.cern.ch)
  - Access to the CERN-IPMC documentation (https://cern-ipmc.web.cern.ch/doc)
    - Datasheet
    - Hardware
    - Software
    - Sensor
- Forum is available: https://cern-ipmc.web.cern.ch/forum