IPMI Slow Control for ONSEN

Björn Spruck for the Mainz Belle II Group

KPH, University Mainz

Trieste 10.9.2015

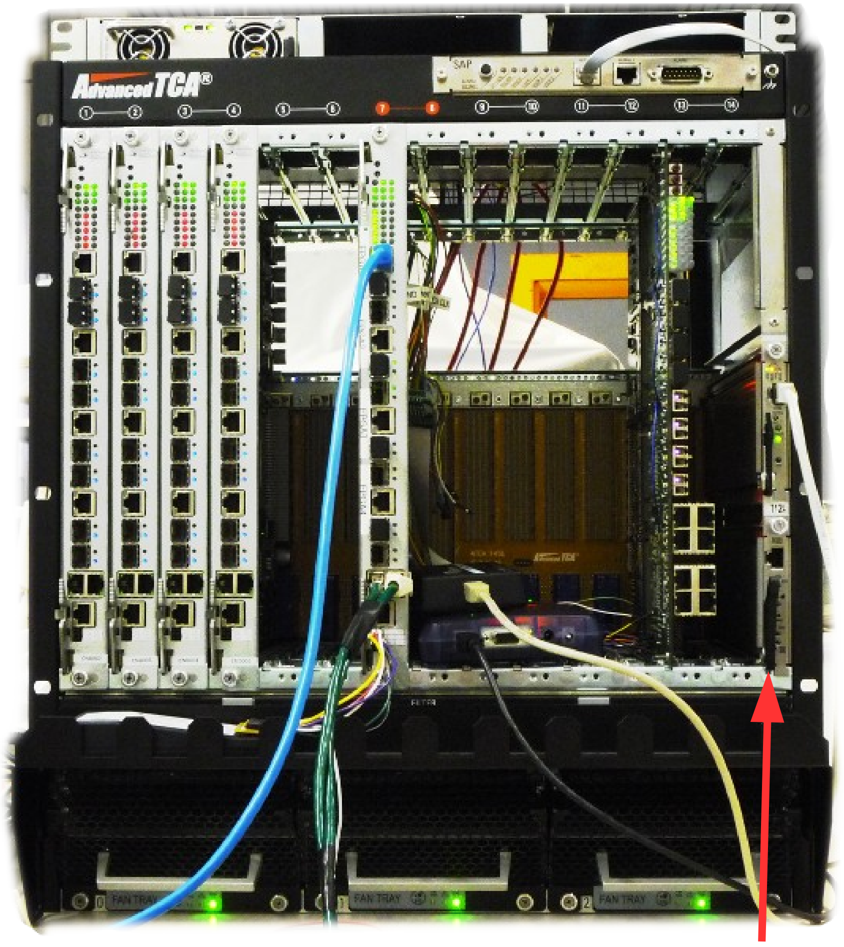
IPMI – ONSEN Slow Control

- IPMI – Requirements for ATCA
- Carrier IPMC – Hardware
- Firmware Tests
- EPICS Integration
- AMC MMC
- Summary

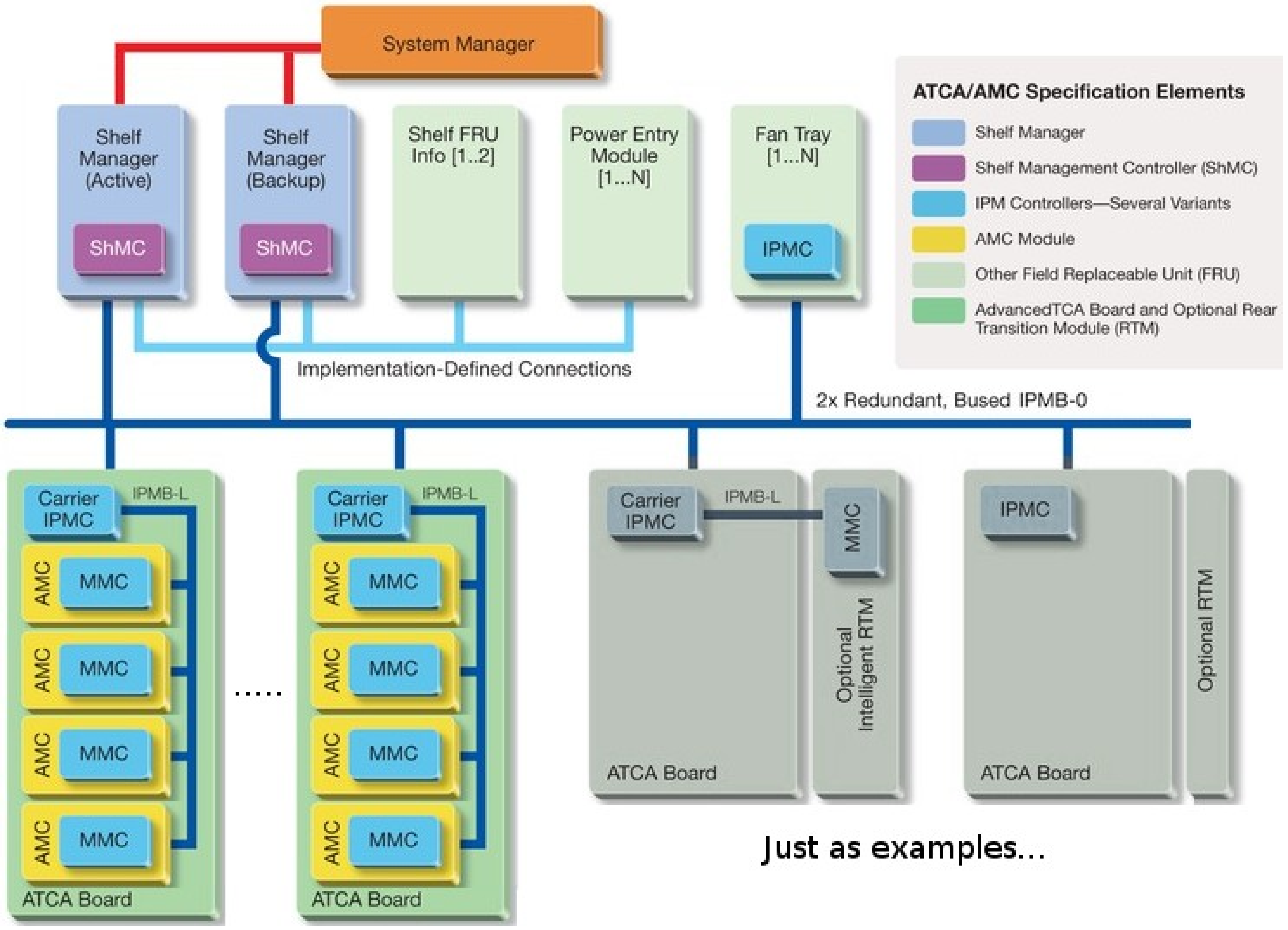
Introduction – IPMI

- Common tool for monitoring hardware
- Used in data centers and telco infrastructure
- ATCA and MTCA make heavy use of it (not only for monitoring)
- By IPMI the shelf controller talk with all FRUs (field replaceable units) in the shelf, power supply, fan trays, … and the boards.
- Hot swap, power negotiation, monitoring, …
- Requirements are strictly defined by PICMG standards, which is PICMG 3.0 R 2.0 for the ATCA Carrier and PICMG AMC.0 R 2.0 for the AMC card. Plus the IPMI v1.5 standard itself.
- (→ that does not mean we want to implement the whole specs...)

ATCA Shelf
(Advanced Telecommunications Computing Architecture)



IPMI Inside the ATCA Shelf



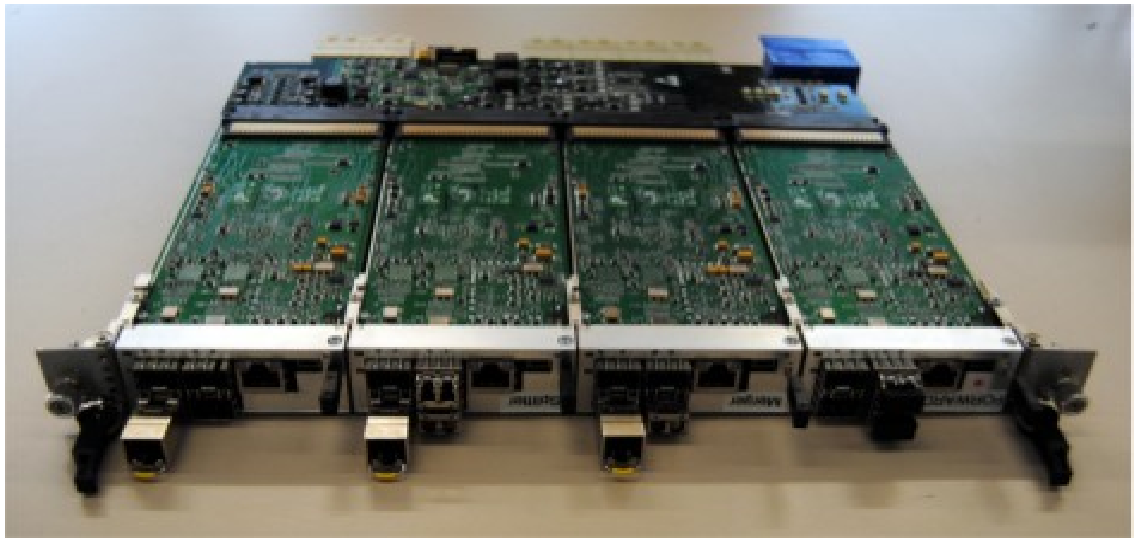
System Manager
Shelf Manager (Active)
ShMC
Shelf Manager (Backup)
ShMC
Shelf FRU Info [1..2]
Power Entry Module [1...N]
Fan Tray [1...N]
IPMC
Implementation-Defined Connections
2x Redundant, Bused IPMB-0
Carrier IPMC
IPMB-L
AMC
MMC
ATCA Board
.....
MMC
Optional Intelligent RTM
IPMC
Optional RTM
ATCA/AMC Specification Elements
Shelf Manager
Shelf Management Controller (ShMC)
IPM Controllers—Several Variants
AMC Module
Other Field Replaceable Unit (FRU)
AdvancedTCA Board and Optional Rear Transition Module (RTM)

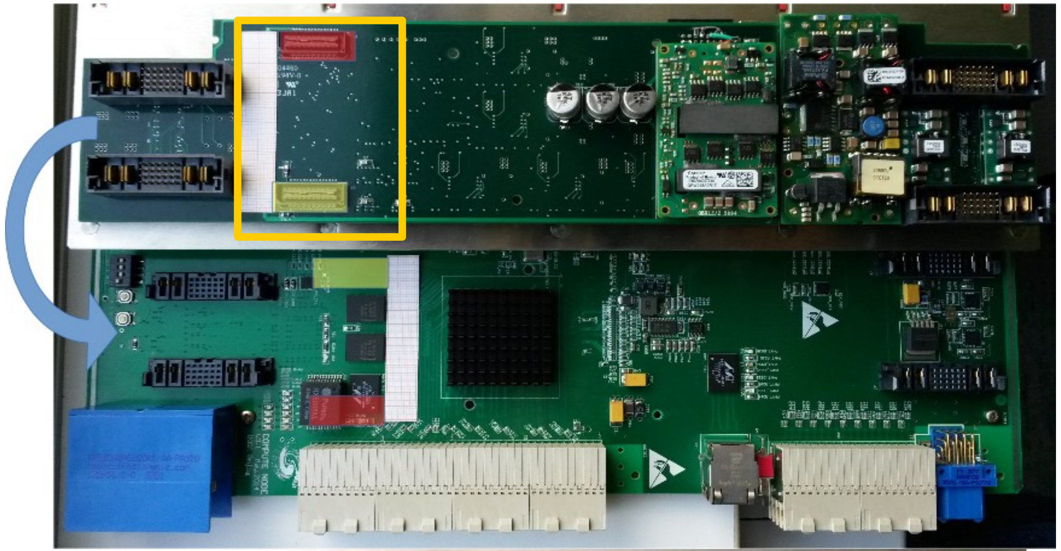
Just as examples...

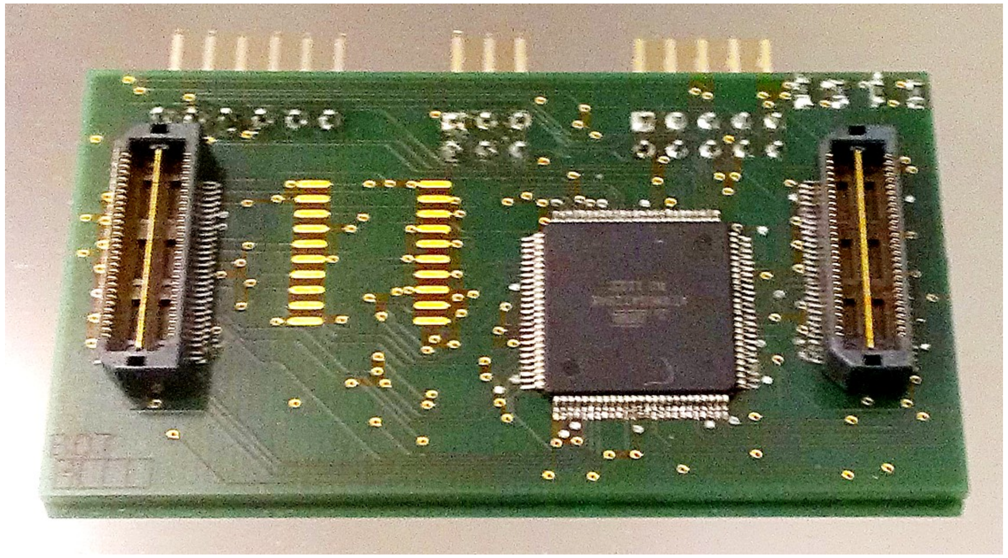
ONSEN: 9 ATCA Boards

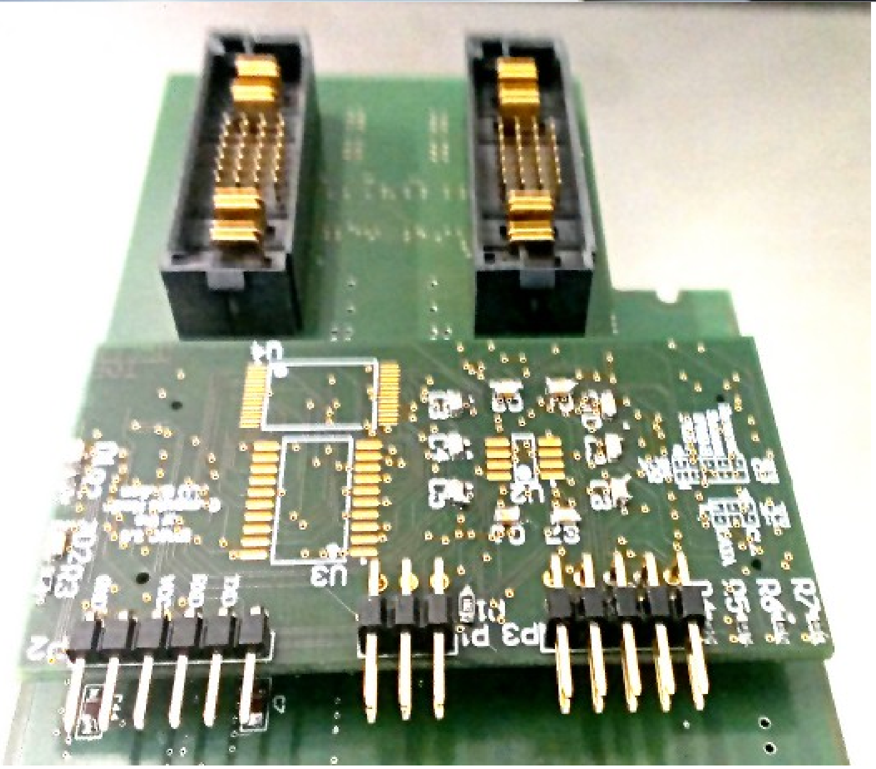
Carrier Power Supply Add-on and IPMC Add-on Board

IPMC plugs between Carrier and PSU board.





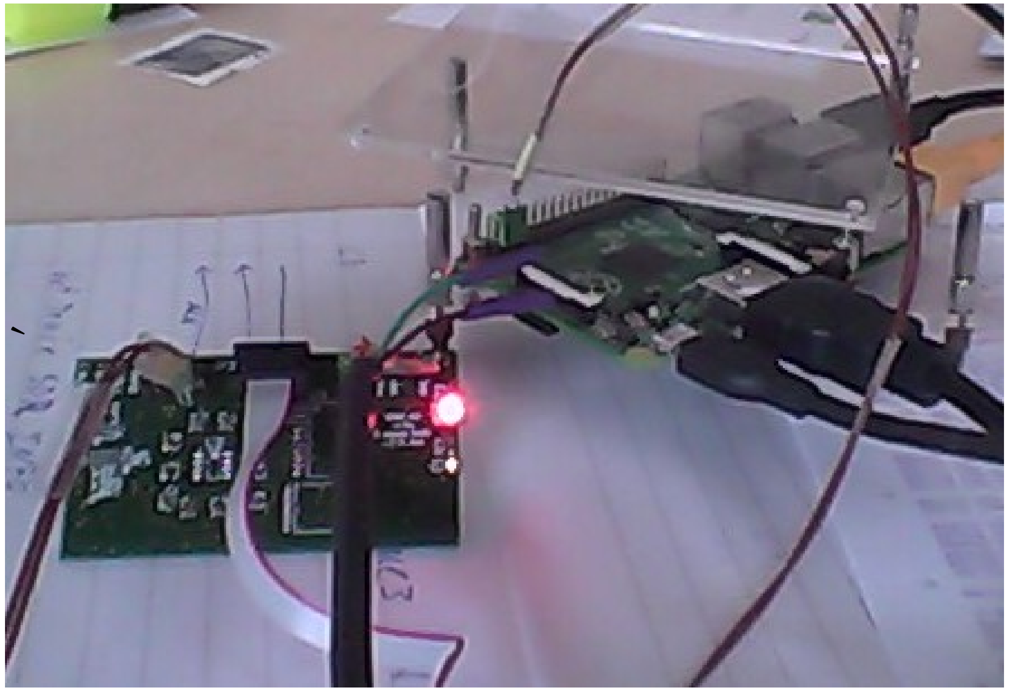




Limited in size (height!)
Hard to access for debugging
If inside shelf, you do not even see the LEDs

IPMC Boards Tests

- Prototype IPMC controller for Carrier Board (PCB Design: Th. Gessler)
  - XMEGA128A1U
- Basic Electric and Software testing done within two days, everything was working as expected
  - Power, Serial, LEDs, I2C
- No shorts found on all accessible pins
  - GND, VCC, neighbors, pullups
- →Test Firmware, test in a ATCA shelf

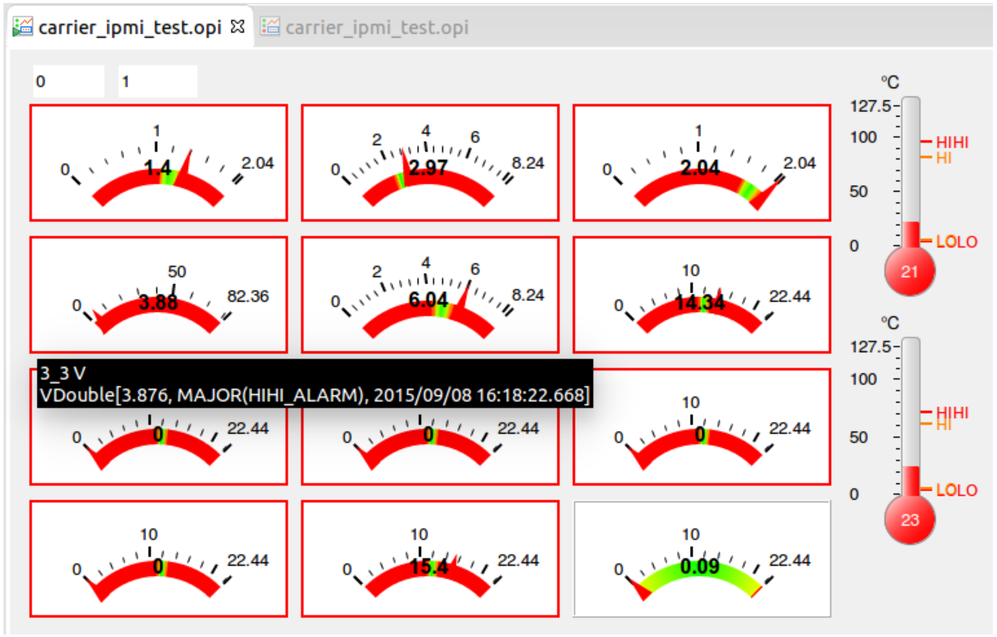


IPMC Firmware

- Based on mTCA.4/MMC example from DESY (NDA!)
- For same CPU, but quite a lot of things are different on our layout (pins, sensors, …), but the basic firmware can be kept (protocols). That was the hope.
- After changing pins assignments, removing not existing sensors, the software was working immediatley.
    - Boot up, show CLI on serial interface
    - IPMI msg send by busses A, B, L are correctly accepted and (on A and B) answered; the peripheral (P) bus sends read commands peripherals
- But: lot of stuff missing/not implemented
- Main issue: MMC vs IPMC (mTCA vs ATCA standard)
- Hope to reuse RTM module code for our AMCs – but only passive RTMs are supported here. We need active ones…
    - Lot of work ahead to get the Carrier ↔ AMC communication
    - PICMG Specs: ATCA (Carrier) ~200 pages + 90 for AMC (IPMI parts only)

Firmware: Monitoring

- All available sensors have been implemented
  - Voltage, Current
  - Temperature
  - HotSwap
  - Limits still need to be refined.



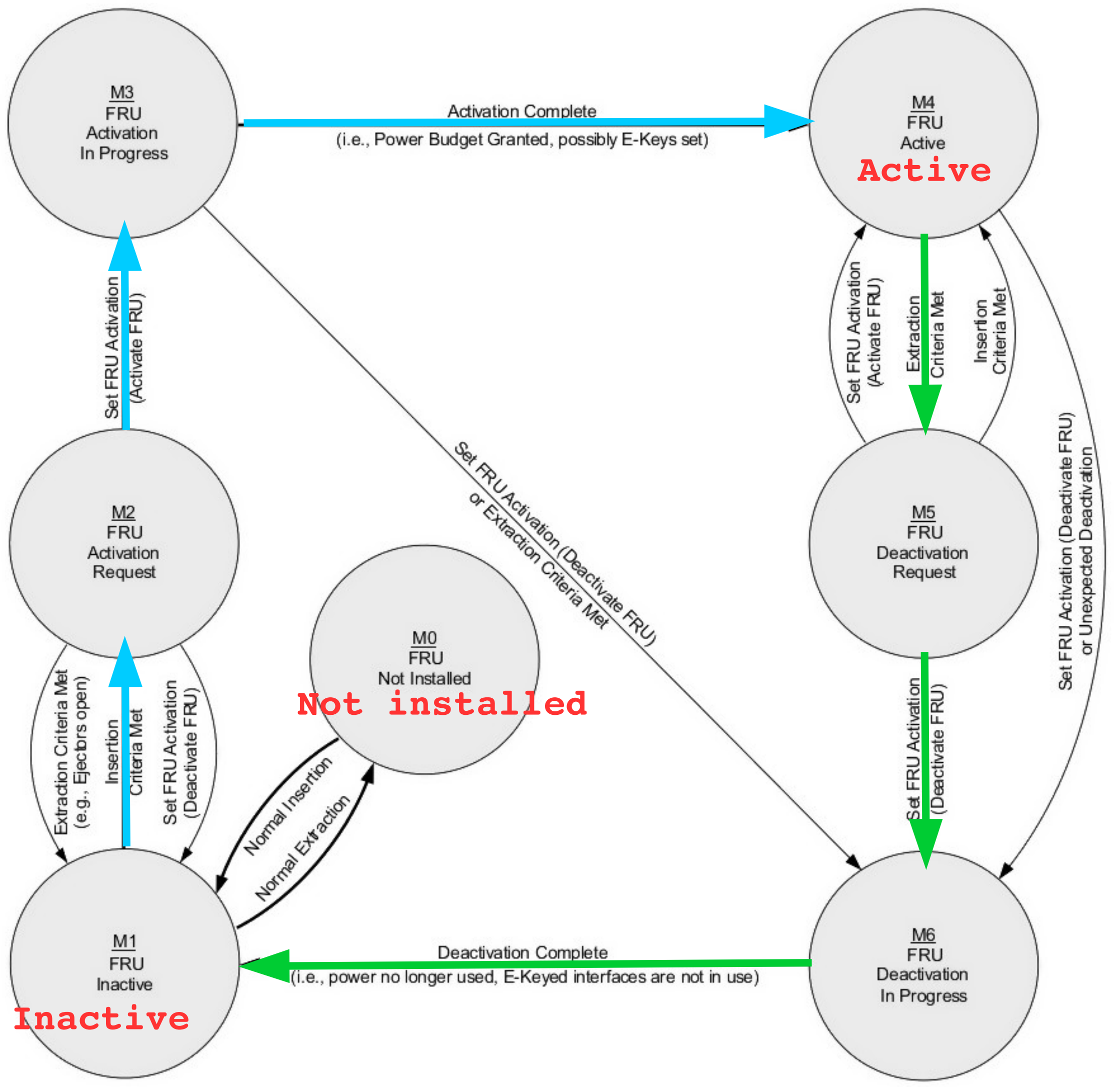
```
3_3V
VDouble[3.876, MAJOR(HIHI_ALARM), 2015/09/08 16:18:22.668]
```

- Sensor info is published to the shelf manager
- Can be read by impitool
- Can be imported to EPICS with ipmitool-IOC (by M. Ritzert)
- Limits are taken from sensor data and alarms are shown in GUI

Hot Plug – FRU States (Activation, Deactivation)

- Following PICMG Specs
  - Several steps for (de)activation
- Activation:
  - Insertion and handle closed → board talks with ShelfManager, negotiate power and finally switches power on
- Deactivation:
  - Handle opened → board asks for deactivation, turns power off, waits for extraction
- HotSwap LED shows current status
- Implemented and tested!

(simplified) FRU State diagram

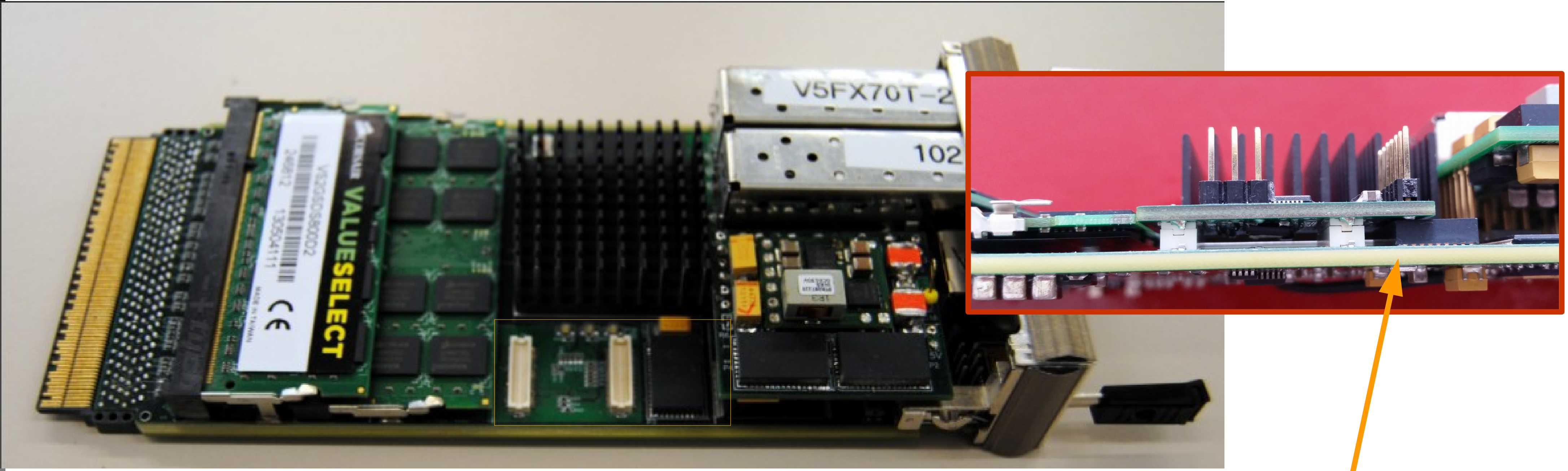

M3 FRU Activation In Progress
Activation Complete
(i.e., Power Budget Granted, possibly E-Keys set)
M4 FRU Active
Active
Set FRU Activation (Activate FRU)
Set FRU Activation (Activate FRU)
Extraction Criteria Met
Insertion Criteria Met
Set FRU Activation (Deactivate FRU) or Extraction Criteria Met
Set FRU Activation (Deactivate FRU) or Unexpected Deactivation
M2 FRU Activation Request
M5 FRU Deactivation Request
M0 FRU Not Installed
Not installed
Extraction Criteria Met (e.g., Ejectors open)
Insertion Criteria Met
Set FRU Activation (Deactivate FRU)
Normal Insertion
Normal Extraction
Set FRU Activation (Deactivate FRU)
M1 FRU Inactive
Inactive
Deactivation Complete
(i.e., power no longer used, E-Keyed interfaces are not in use)
M6 FRU Deactivation In Progress

In Shelf Test Results

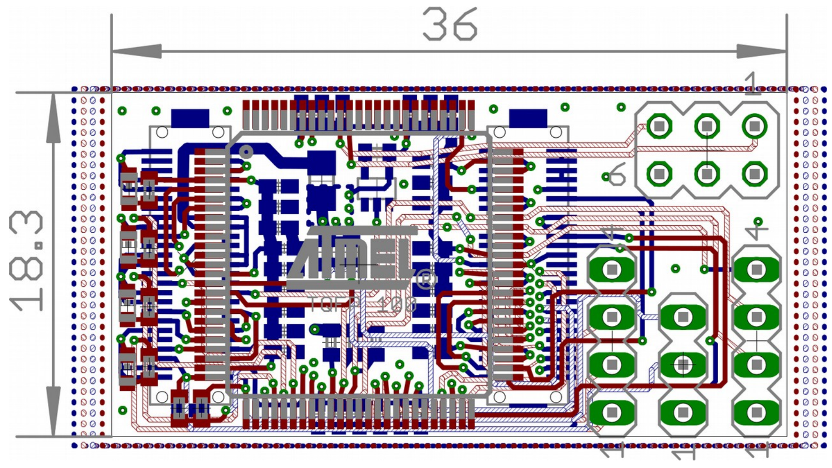
- Nearly all connections to PSU and Carrier could be tested
  - (a test for PSU and Carrier, too)
  - I2C, HotSwap handle and LED, AMC and RTM presence, management and payload power switch pins, I2C switches
  - CPLD and FPGA pins
- A few pins could not be verified:
  - MMC reset for AMC and RTM
  - FPGA init and done.
  - Some shelf address lines
- A few things look fishy, but can be related to non-final PSU and carrier board (which were used for this test)
  - Indication: depend on carrier board version used.
  - I2C bus hangs under certain circumstances (layout change in PSU v1.3)
  - Sensors cannot be read until payload power is on (…)

MMC: AMC Add-on

MMC (IPMI) add-on board plugs here



- Size is even more constrained. Height for pin headers not an issue.
- No components at bottom side on the right due to DC-DC converter
    - Connectors have too low profile
- Decided to use same XMEGA CPU
    - → avoid extra firmware work.
- PCB design finished and ordered (15 pcs)



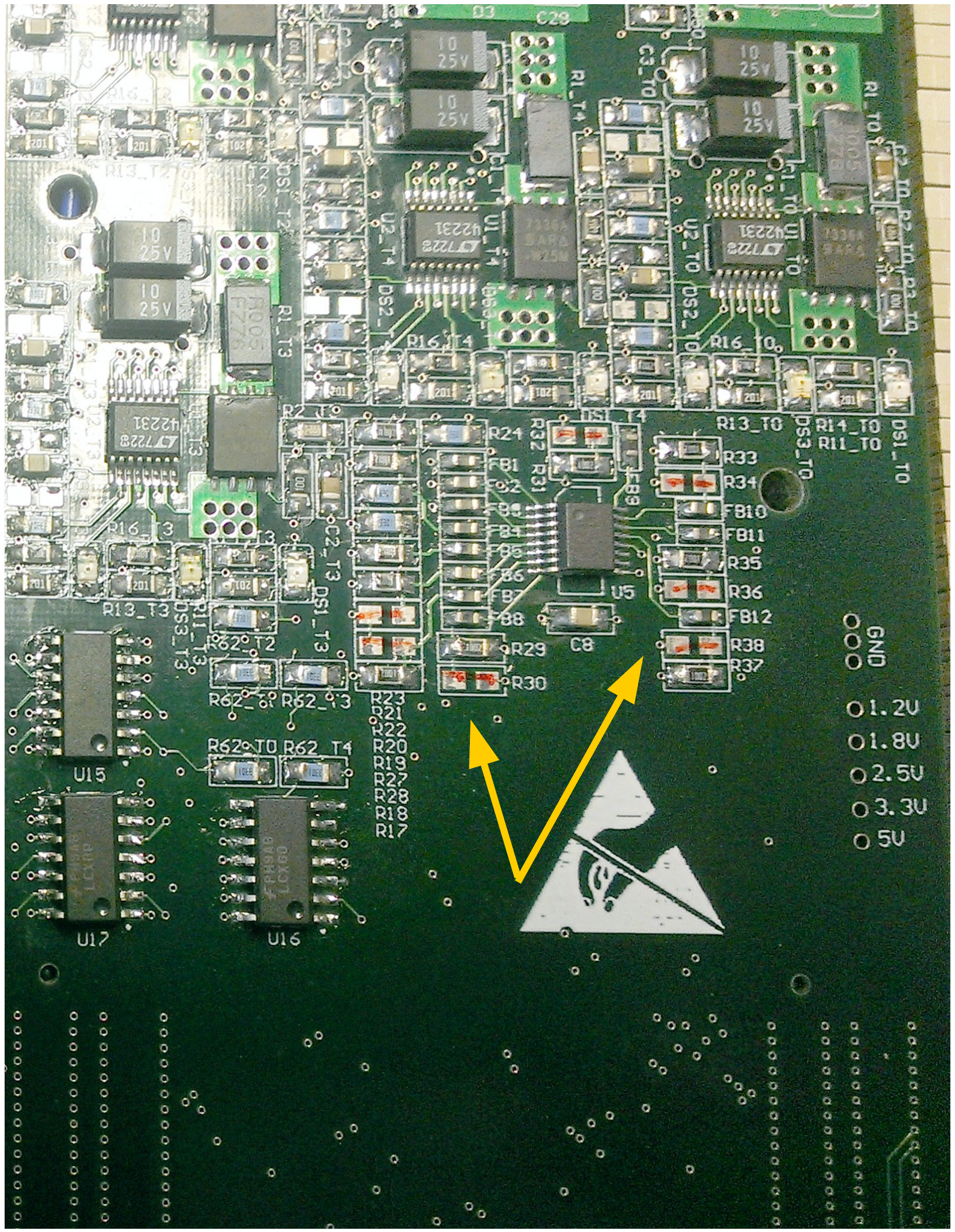
Summing up

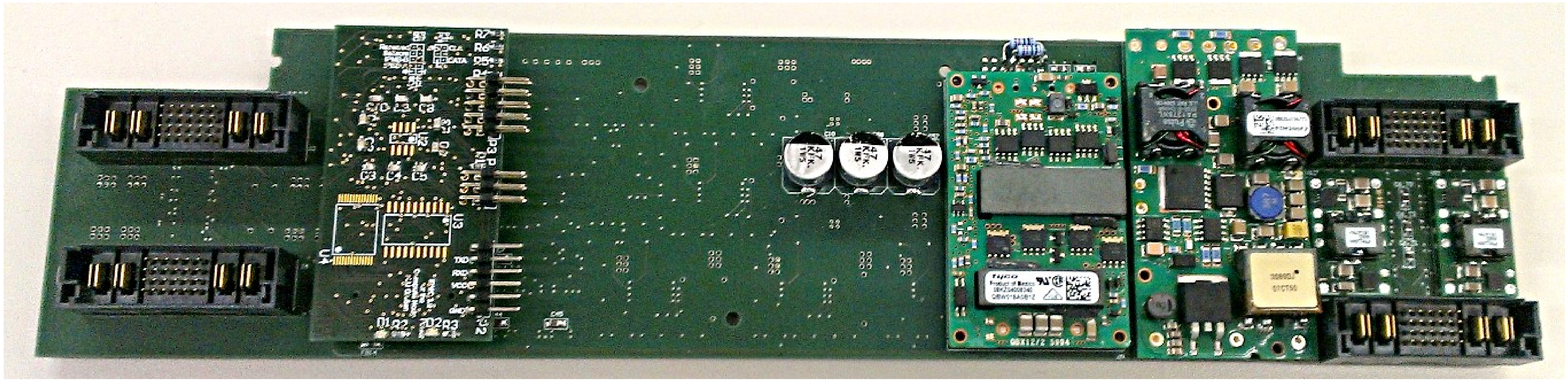
- IPMC hardware is working
- IPMC firmware is working but not to finished soon
    - Already sufficient for carrier only operation and monitoring
- Done:
    - Serial CLI, shelf manager (IPMB-A and B), reading peripherals (IPMB-P)
    - FRU and sensor information, Power for payload, AMC and RTM switchable
    - FRU states, HotSwap handle and LED, power negotiation
    - EPICS ipmitool-IOC can read and export PVs → SC & GUI
- Todo:
    - mass production when we are sure that all the pins to PSU and carrier are correctly working
    - MMC hard- and firmware
    - implement missing functionality for an carrier controller (needed for plugin AMC boards) – when MMC is available, test communication IMPC ↔ MMC

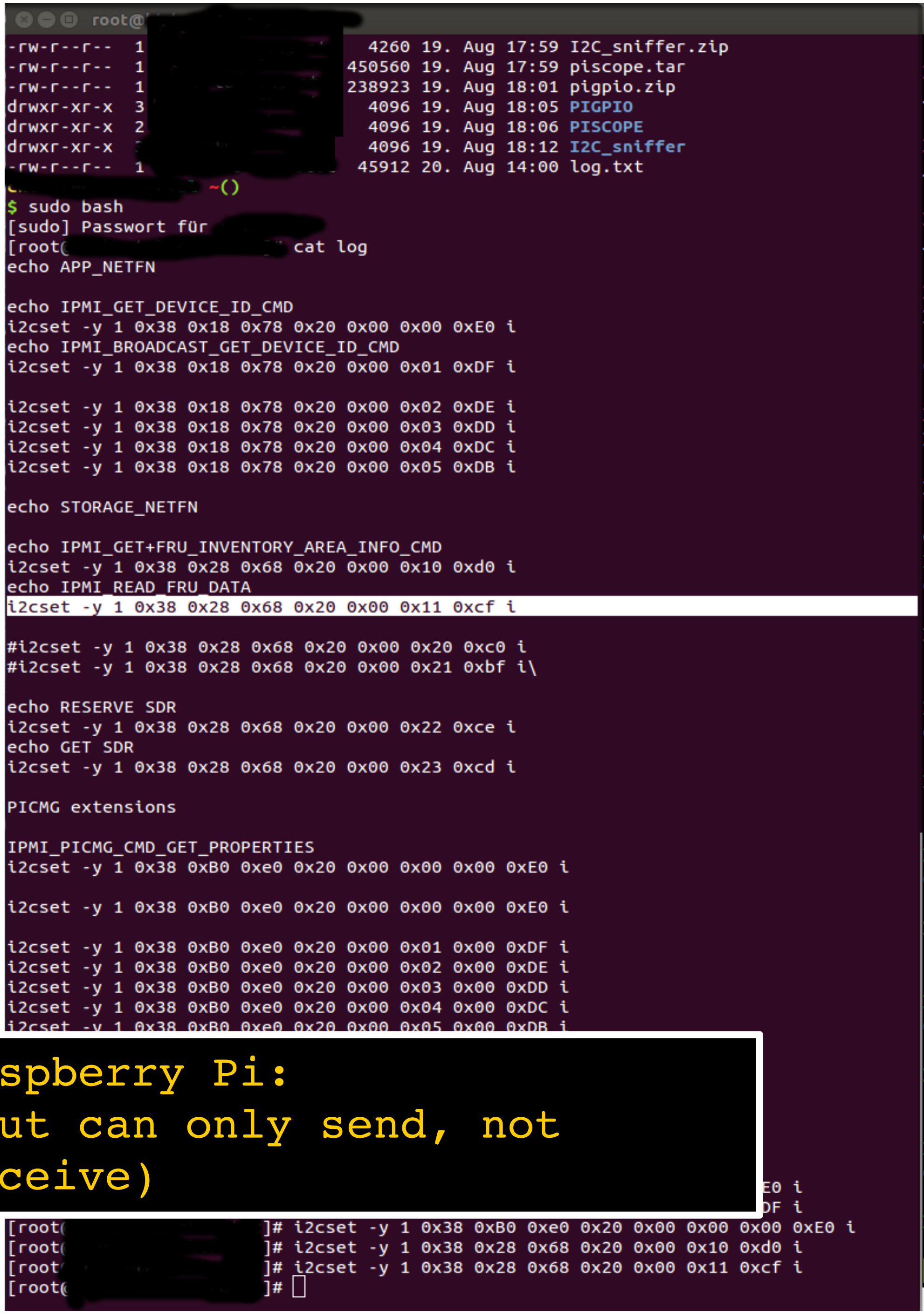
Backup

Reading I2C sensors, ADC and Temperature

- Voltages and current can be read
- Temperatures can be read
- Observations:
  - Voltages measure too high – ADC might have been damaged (voltage dividers were not present on the PSU board, thus 12V was applied to 2V inputs.)
  - (Voltage at chip pins is o.k.)
- I2C cannot be read if payload power is not switched on.
- AMC boards and sensors share the same bus
  - Fixed on V1.3 PSU

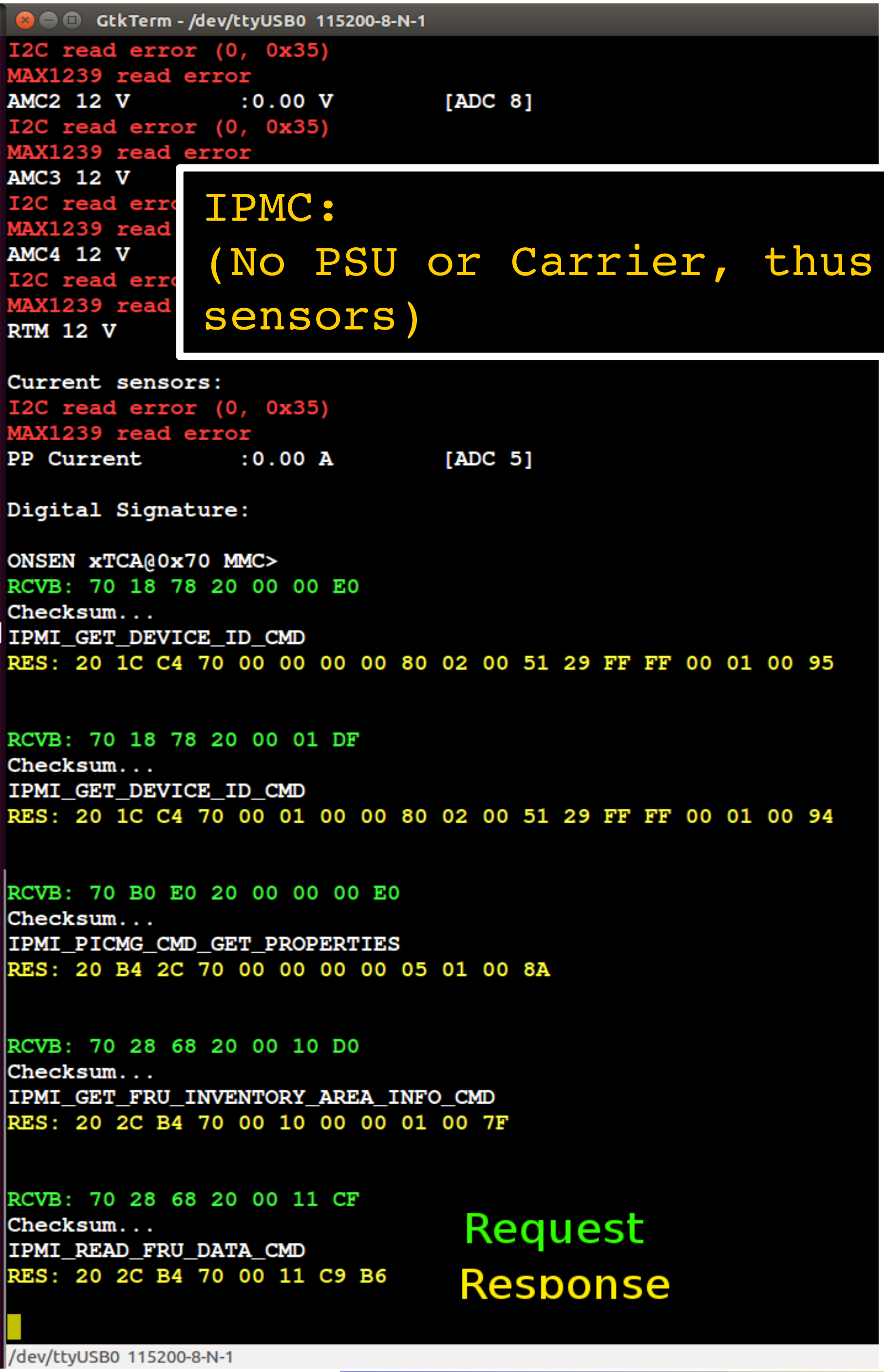






```
root@[redacted]
-rw-r--r--  1 [redacted]   4260 19. Aug 17:59 I2C_sniffer.zip
-rw-r--r--  1            450560 19. Aug 17:59 piscope.tar
-rw-r--r--  1            238923 19. Aug 18:01 pigpio.zip
drwxr-xr-x  3              4096 19. Aug 18:05 PIGPIO
drwxr-xr-x  2              4096 19. Aug 18:06 PISCOPE
drwxr-xr-x  3              4096 19. Aug 18:12 I2C_sniffer
-rw-r--r--  1             45912 20. Aug 14:00 log.txt
[redacted] ~()
$ sudo bash
[sudo] Passwort für [redacted]
[root[redacted] cat log
echo APP_NETFN

echo IPMI_GET_DEVICE_ID_CMD
i2cset -y 1 0x38 0x18 0x78 0x20 0x00 0x00 0xE0 i
echo IPMI_BROADCAST_GET_DEVICE_ID_CMD
i2cset -y 1 0x38 0x18 0x78 0x20 0x00 0x01 0xDF i

i2cset -y 1 0x38 0x18 0x78 0x20 0x00 0x02 0xDE i
i2cset -y 1 0x38 0x18 0x78 0x20 0x00 0x03 0xDD i
i2cset -y 1 0x38 0x18 0x78 0x20 0x00 0x04 0xDC i
i2cset -y 1 0x38 0x18 0x78 0x20 0x00 0x05 0xDB i

echo STORAGE_NETFN

echo IPMI_GET+FRU_INVENTORY_AREA_INFO_CMD
i2cset -y 1 0x38 0x28 0x68 0x20 0x00 0x10 0xd0 i
echo IPMI_READ_FRU_DATA
i2cset -y 1 0x38 0x28 0x68 0x20 0x00 0x11 0xcf i

#i2cset -y 1 0x38 0x28 0x68 0x20 0x00 0x20 0xc0 i
#i2cset -y 1 0x38 0x28 0x68 0x20 0x00 0x21 0xbf i\

echo RESERVE SDR
i2cset -y 1 0x38 0x28 0x68 0x20 0x00 0x22 0xce i
echo GET SDR
i2cset -y 1 0x38 0x28 0x68 0x20 0x00 0x23 0xcd i

PICMG extensions

IPMI_PICMG_CMD_GET_PROPERTIES
i2cset -y 1 0x38 0xB0 0xe0 0x20 0x00 0x00 0x00 0xE0 i

i2cset -y 1 0x38 0xB0 0xe0 0x20 0x00 0x00 0x00 0xE0 i

i2cset -y 1 0x38 0xB0 0xe0 0x20 0x00 0x01 0x00 0xDF i
i2cset -y 1 0x38 0xB0 0xe0 0x20 0x00 0x02 0x00 0xDE i
i2cset -y 1 0x38 0xB0 0xe0 0x20 0x00 0x03 0x00 0xDD i
i2cset -y 1 0x38 0xB0 0xe0 0x20 0x00 0x04 0x00 0xDC i
i2cset -y 1 0x38 0xB0 0xe0 0x20 0x00 0x05 0x00 0xDB i
                                                      E0 i
                                                      DF i
[root[redacted]]# i2cset -y 1 0x38 0xB0 0xe0 0x20 0x00 0x00 0x00 0xE0 i
[root[redacted]]# i2cset -y 1 0x38 0x28 0x68 0x20 0x00 0x10 0xd0 i
[root[redacted]]# i2cset -y 1 0x38 0x28 0x68 0x20 0x00 0x11 0xcf i
[root[redacted]]# 
```

Raspberry Pi:
(But can only send, not receive)

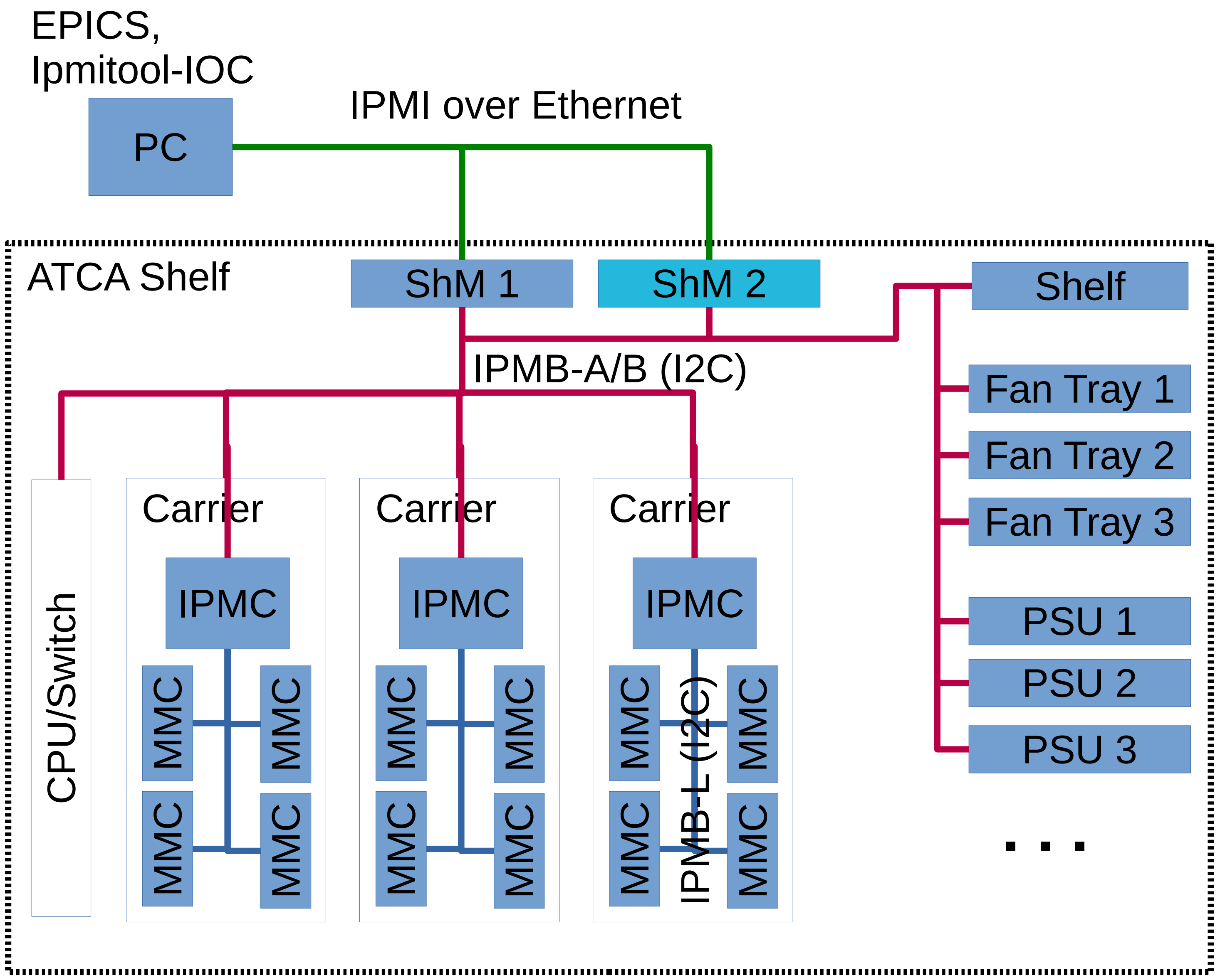


```
GtkTerm - /dev/ttyUSB0  115200-8-N-1
I2C read error (0, 0x35)
MAX1239 read error
AMC2 12 V          :0.00 V        [ADC 8]
I2C read error (0, 0x35)
MAX1239 read error
AMC3 12 V
I2C read err
MAX1239 read
AMC4 12 V
I2C read err
MAX1239 read
RTM 12 V

Current sensors:
I2C read error (0, 0x35)
MAX1239 read error
PP Current         :0.00 A        [ADC 5]

Digital Signature:

ONSEN xTCA@0x70 MMC>
RCVB: 70 18 78 20 00 00 E0
Checksum...
IPMI_GET_DEVICE_ID_CMD
RES: 20 1C C4 70 00 00 00 00 80 02 00 51 29 FF FF 00 01 00 95


RCVB: 70 18 78 20 00 01 DF
Checksum...
IPMI_GET_DEVICE_ID_CMD
RES: 20 1C C4 70 00 01 00 00 80 02 00 51 29 FF FF 00 01 00 94


RCVB: 70 B0 E0 20 00 00 00 E0
Checksum...
IPMI_PICMG_CMD_GET_PROPERTIES
RES: 20 B4 2C 70 00 00 00 00 05 01 00 8A


RCVB: 70 28 68 20 00 10 D0
Checksum...
IPMI_GET_FRU_INVENTORY_AREA_INFO_CMD
RES: 20 2C B4 70 00 10 00 00 01 00 7F


RCVB: 70 28 68 20 00 11 CF
Checksum...
IPMI_READ_FRU_DATA_CMD
RES: 20 2C B4 70 00 11 C9 B6

/dev/ttyUSB0 115200-8-N-1
```

IPMC:
(No PSU or Carrier, thus no sensors)

Request
Response

IPMI Schematic



~10 IPMCs
~40 MMCs

Run Control

- Test with RunControl with 8 (AMC) boards and 34 simulated boards (25 AMCs, 9 Carrier).
- Using bitstream and EPICS flash content from 2014 (KEK test)
- Set-up and programmed.
- Status:
    - EPICS (software IOCs, RC) and GUI ar on a different netwrok than the AMC cards. Tunneling of PVs nor working in both directions → Test postponed
    - Fix network topology, gateway