DESY MMC STAMP.

Overview and recent improvements

Patrick Huesmann
Software Engineer, DESY MSK

2025-09-15

HELMHOLTZ

DMMC-STAMP overview

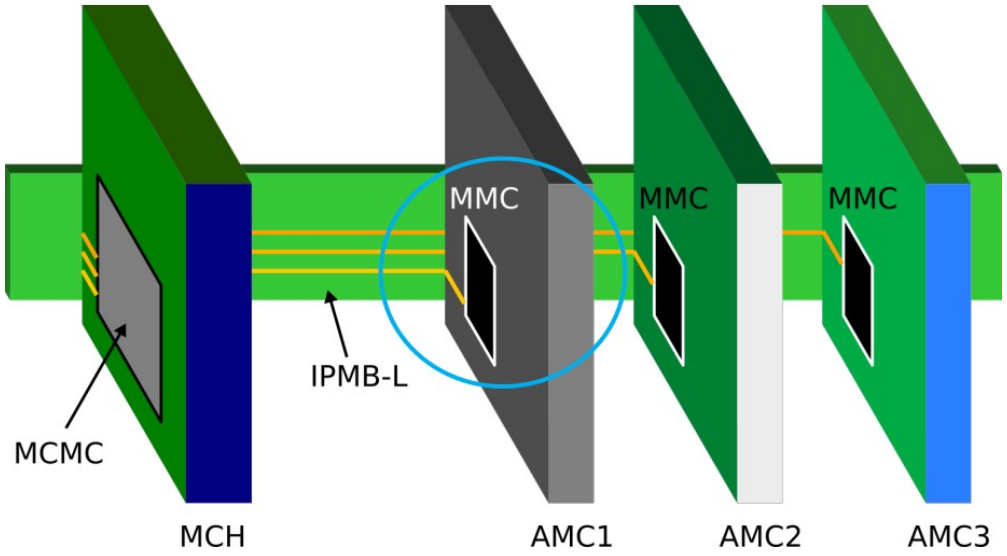
Overview MicroTCA MMC

Management layer in MicroTCA

- On MCH side:
  MicroTCA Carrier Management Controller (MCMC)
- On AMC side:
  Module Management Controller (MMC)

Responsibilities of the MMC

- IPMI protocol handling
- AMC payload (FPGAs/SoCs) management
- FRU information (AMC, RTM, FMCs) processing
- RTM control (hot-plug)
- Monitoring (temperatures, voltages, currents)
- Event handling (thresholds, alerts)



Overview DMMC-STAMP

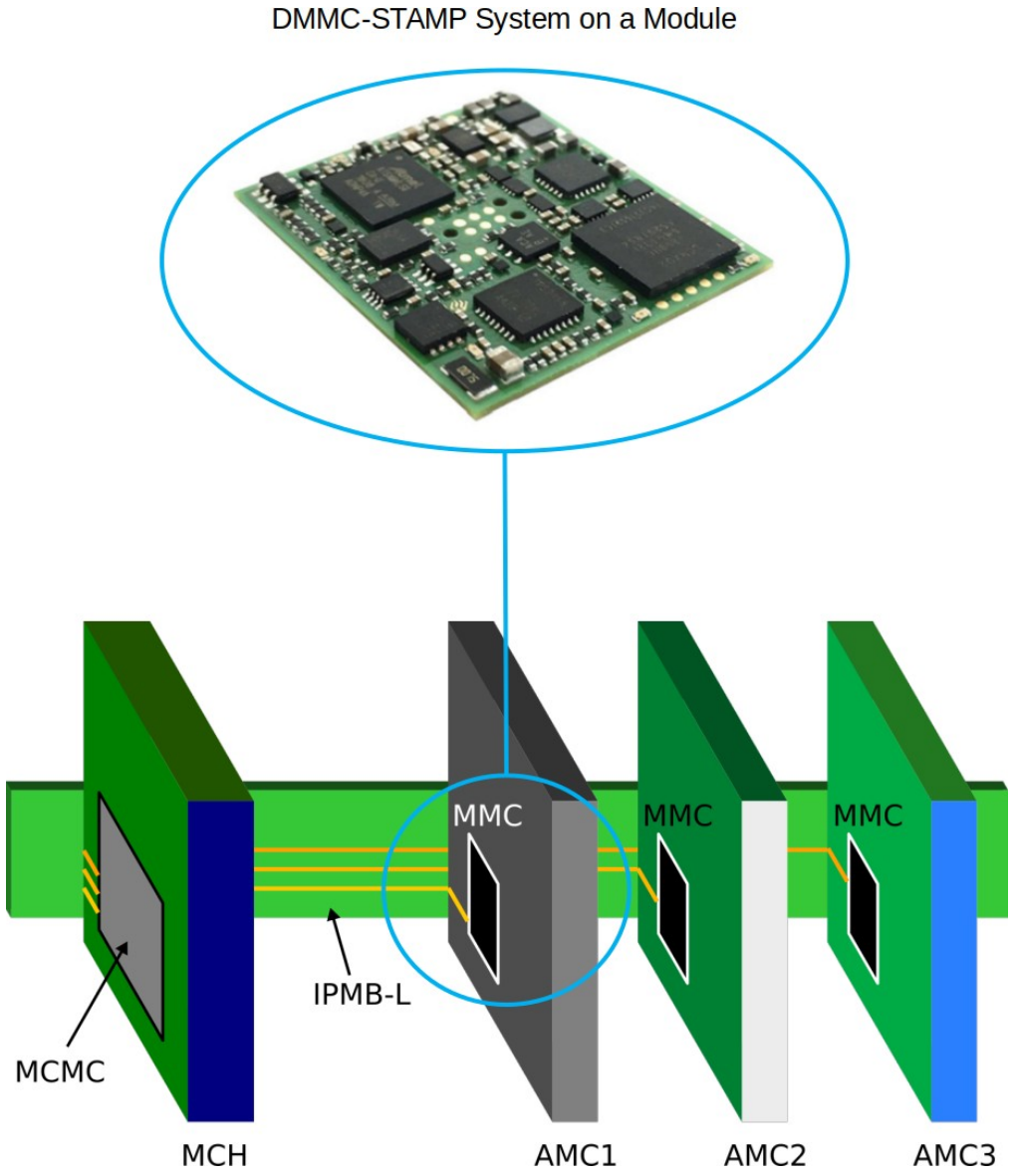
DESY's drop-in solution for AMC MMC

Used at DESY for internally developed AMCs

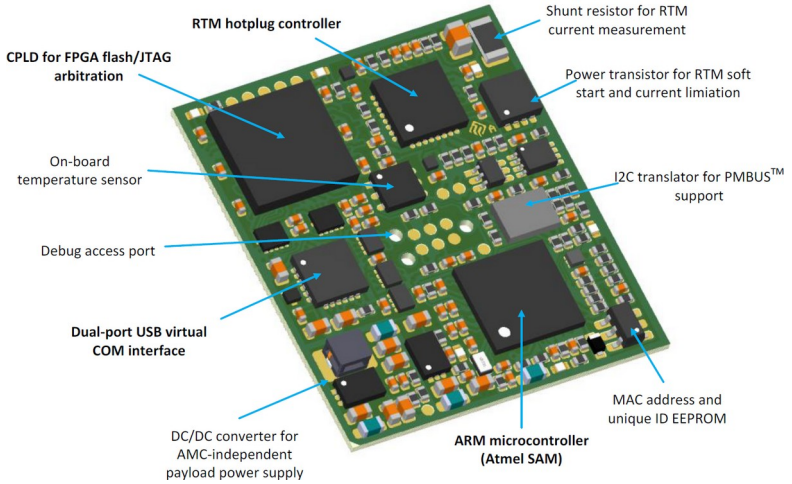
- DAMC-FMC2ZUP
- DAMC-FMC1Z7IO
- DAMC-DS812ZUP
- DAMC-MOTCTRL
- DAMC-UNIZUP
- DAMC-X3TIMER
- DAMC-DS5014DR

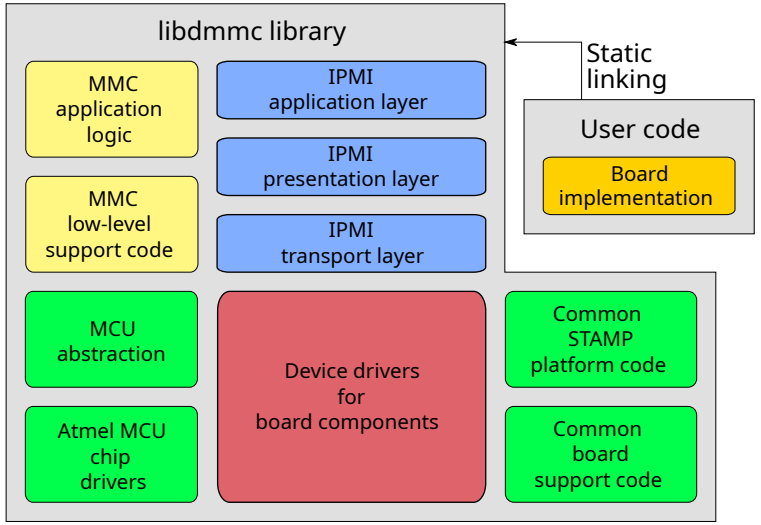
Used by several customers in research and industry

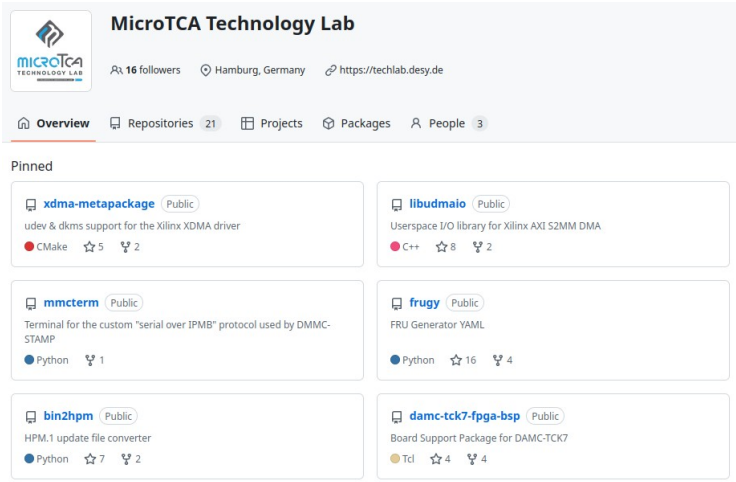
- Interoperability tested with different MCH vendors
- Customer feedback leads to continuous firmware improvement



DMMC-STAMP ecosystem







System on Module (SoM)

- 25.5 x 29.5 x 2.3 mm
- Pre-programmed firmware
- Evaluation board available DMMC-STAMP-BREAKOUT

Software Development Kit (SDK)

- Enables customization of DMMC-STAMP firmware for new boards
- Used by DMMC-STAMP customers and by DESY internally
- **Upstream bugfixes & features propagate to board implementations**

Open Source Tools and Templates

- **AMC & RTM PCB Templates**
- mmcterm: serial over IPMB
- bin2hpm: IPMI upgrade file builder
- frugy: read and write FRUs
- cpld-img-tools: bitstream conversion for Lattice CPLDs

Recent developments & special features

FMC sensor integration

Some FMCs dissipate lots of heat, requiring extra cooling.

To monitor their temperature and control cooling on demand, local temperature sensors are required

→ FMC-local temperature sensors need to be integrated into IPMI monitoring

Solution: "I2C Device Definition FRU record" from ANSI/VITA 57.1

- The FMC FRU record encodes a "device name" and a I2C address
- MMC implements a lookup table from device name to sensor driver
- MMC parses device name and address, then polls the sensor and exposes its reading to IPMI

→ MMC only needs to know the sensor, not the particular FMC

Field	Byte Offset	Bit Location	Length	Description
Subtype	0	7:4	4 bits	1 for I2C device definition subtype
Version	0	3:0	4 bits	0 for current version
Device String	1..N/8		N bits	Device address / name strings, see below

Table 9. Subtype 1: I2C Device Definition (variable length and optional)

The device string portion of this MultiRecord subtype consists of 6-bit ASCII text as defined in the ISD. The string is divided into one or more I2C device records. Each device record consists of one or more address characters followed by one or more bytes of device name.

From FMC standard (ANSI/VITA 57.1)

Example: DAC FMC (Atom Computing)

```
DAMC-FMC2ZUP@0x7E MMC>fru   Show parsed FRUs on MMC CLI
FRU #0:
Product info: DESY/CAEN ELS DAMC-FMC2ZUP-11EG
              S/N 22Y23W0632 P/N DAMCFMC2ZUP1
              Version revB
  Board info: DESY/CAEN ELS DAMC-FMC2ZUP-11EG
              S/N 22Y23W0632 P/N DAMCFMC2ZUP1
              Mfg.Date 2022-09-27 10:17:00
Module current requirements: 6.5A
Zone3 interface compat: Class D1.1

FRU #1:
Not present

FRU #2:  This is FMC1 on the FMC carrier
Product info: N/A
  Board info: Atom Computing, Inc. Opus DAC
              S/N 2 P/N Opus-DAC-revA
              Mfg.Date 2022-05-18 18:02:00
DC Load P1_12P0V: 12V (min 11.4, max 12.6) ~0mV, min 0mA / max 1000mA
DC Load P1_3P3V: 3.3V (min 3.12, max 3.46) ~0mV, min 0mA / max 100mA
DC Load P1_VADJ: 2.5V (min 1.8, max 3.3) ~0mV, min 0mA / max 100mA
DC Output P1_VREF_B_M2C: 0 -0+0V ~0mV, min 0mA / max 0mA
DC Output P1_VREF_A_M2C: 0 -0+0V ~0mV, min 0mA / max 0mA
DC Output P1_VIO_B_M2C: 0 -0+0V ~0mV, min 0mA / max 0mA
FMC size: single, clock dir: m2c, TCK max clock: 64
      P1: lpc, num signals: A 68, B 0, num Gbt trcv: 0
      P2: lpc, num signals: A 0, B 0, num Gbt trcv: 0
FMC I2C device definition:
  name: AT30TS75A, addrs: 9 (0x48)   FMC1 has an AT30TS75A sensor at
                                     I²C address 0x48

FRU #3:
Product info: N/A
  Board info: CAENELS FMC-4SFP+
              S/N 17006 P/N F
              Mfg.Date 2015-06-10 00:00:00
```

```
nat> show_sensorinfo 11   Show AMC sensors on MCH console
Sensor Information for FRU 11 / AMC7
=====================================================================
  #   SDRType  Sensor Entity Inst  Value    State    Name
---------------------------------------------------------------------
  -   MDevLoc          0xc1  0x67                   DAMC-FMC2ZUP-11E
  0   Full     0xf2    0xc1  0x67  0x01             AMC Hot Swap
  1   Compact  0x0b    0xc1  0x67  0x00             0x00  682719B5F5E3
  2   Full     Temp    0xc1  0x67  30.0 C     ok    STAMP Temp
  3   Full     Voltage 0xc1  0x67  3.360 V    ok    AMC MP 3V3
  4   Full     Voltage 0xc1  0x67  12.32 V    ok    AMC PP 12V
  5   Full     Current 0xc1  0x67  0.000 A    ok    I_RTM MP 3V3
  6   Full     Current 0xc1  0x67  0.00 A     ok    I_RTM PP 12V
  7   Compact  0x14    0xc1  0x67  0x01             0x00  CPLD Done
  8   Compact  0x14    0xc1  0x67  0x00             0x00  RTM MP 3V3 PG
  9   Compact  0x14    0xc1  0x67  0x00             0x00  RTM PP 12V PG
 10   Compact  0x14    0xc1  0x67  0x00             0x00  RTM Fault
 11   Compact  0x14    0xc1  0x67  0x01             0x00  PGood_A
 12   Compact  0x14    0xc1  0x67  0x01             0x00  PGood_B
 13   Compact  0x14    0xc1  0x67  0x01             0x00  FPGA1 Init
 14   Compact  0x14    0xc1  0x67  0x01             0x00  FPGA1 Done
 15   Compact  0x14    0xc1  0x67  0x01             0x00  FPGA2 Init
 16   Compact  0x14    0xc1  0x67  0x01             0x00  FPGA2 Done
 17   Full     Temp    0xc1  0x67  37.5 C     ok    Inlet Temp
 18   Full     Temp    0xc1  0x67  37.5 C     ok    Outlet Temp
 19   Full     Temp    0xc1  0x67  37.0 C     ok    LTM4630 Temp
 20   Full     Temp    0xc1  0x67  39.0 C     ok    LTM4650 Temp
 21   Full     Temp    0xc1  0x67  41.0 C     ok    LTM4633_F Temp

 30   Full     Voltage 0xc1  0x67  1.7856 V   ok    VCC_Vadj
 31   Full     Voltage 0xc1  0x67  1.1904 V   ok    VCC_1V2
 32   Compact  0x14    0xc1  0x67  0x01             0x00  Opus DAC PG_M2C
 33   Full     Temp    0xc1  0x67  35.0 C     ok    Opus DAC AT30TS7
 34   Compact  0x14    0xc1
 35   Compact  0xf0    0xc1
-----------------------------
```

FMC temp. integrated into MTCA monitoring

PMBUS multi-chip configuration

On complex boards with many power rails, a lot of power management logic is required.

Example DAMC-MOTCTRL:

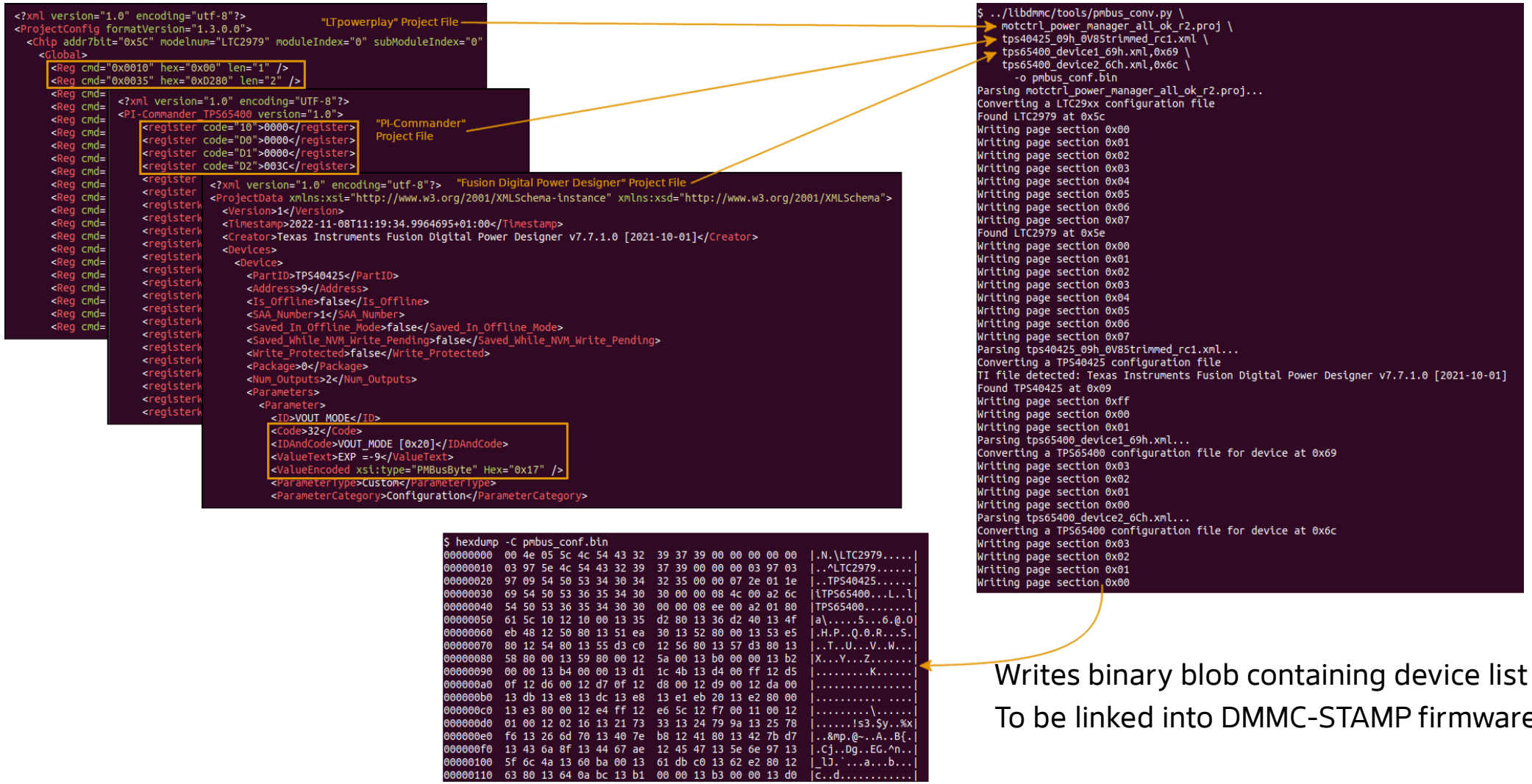
- 2x LTC2979 (PMBUS manager for ext. DC/DC converters)
- 1x TPS40425 (PMBUS controller with ext. MOSFETs)
- 2x TPS65400 (PMBUS converter with int. MOSFETs)

3 different Windows applications and 3 different programming adapters required, just to configure the PMBUS chips:

- Analog Devices LTpowerPlay & DC1613A
- Texas Instruments Fusion Digital Power Designer & USB2ANY
- Texas Instruments PI-Commander & USB-TO-GPIO2

Programming adapters required for power managers on DAMC-MOTCTRL

Conversion script for PMBUS project files



Writes binary blob containing device list & register contents

To be linked into DMMC-STAMP firmware

PMBUS configuration via DMMC-STAMP

Now the MMC configures the whole network of PMBUS chips with one CLI command.

- No more fiddling with 3 different HW / SW tools
- Saves time & effort during production / development
- Less error prone

- Build system can directly read project files of PMBUS config tools
- PMBUS project files become part of the versioned MMC source tree
- PMBUS config is in a known state across all boards
- PMBUS config can be updated **in application** (i.e. without programming adapter) by updating the MMC version

```
DAMC-MOTCTRL@0x7C MMC>pmc write
Waiting for PSM 0x5c shutdown...
Waiting for PSM 0x5e shutdown...
Write chip at 0x5c
LTC29xx device detected at 0x5c: 0x8061
Write globals
Write page 0
Write page 1
Write page 2
Write page 3
Write page 4
Write page 5
Write page 6
Write page 7
Write for #0 (LTC2979) successful
Store chip at 0x5c
LTC29xx device detected at 0x5c: 0x8061
Store for #0 (LTC2979) successful
Write chip at 0x5e
LTC29xx device detected at 0x5e: 0x8071
Write globals
Write page 0
Write page 1
Write page 2
Write page 3
Write page 4
Write page 5
Write page 6
Write page 7
Write for #1 (LTC2979) successful
Store chip at 0x5e
LTC29xx device detected at 0x5e: 0x8071
Store for #1 (LTC2979) successful
Write chip at 0x09
TPS40425 device detected at 0x09: 0x00c3
Write page 255
Write page 0
Write page 1
Write for #2 (TPS40425) successful
Store chip at 0x09
TPS40425 device detected at 0x09: 0x00c3
Store for #2 (TPS40425) successful
Write chip at 0x69
TPS65400 device detected at 0x69: 0x00f1
...
```

Remote console with Serial-over-IMPB

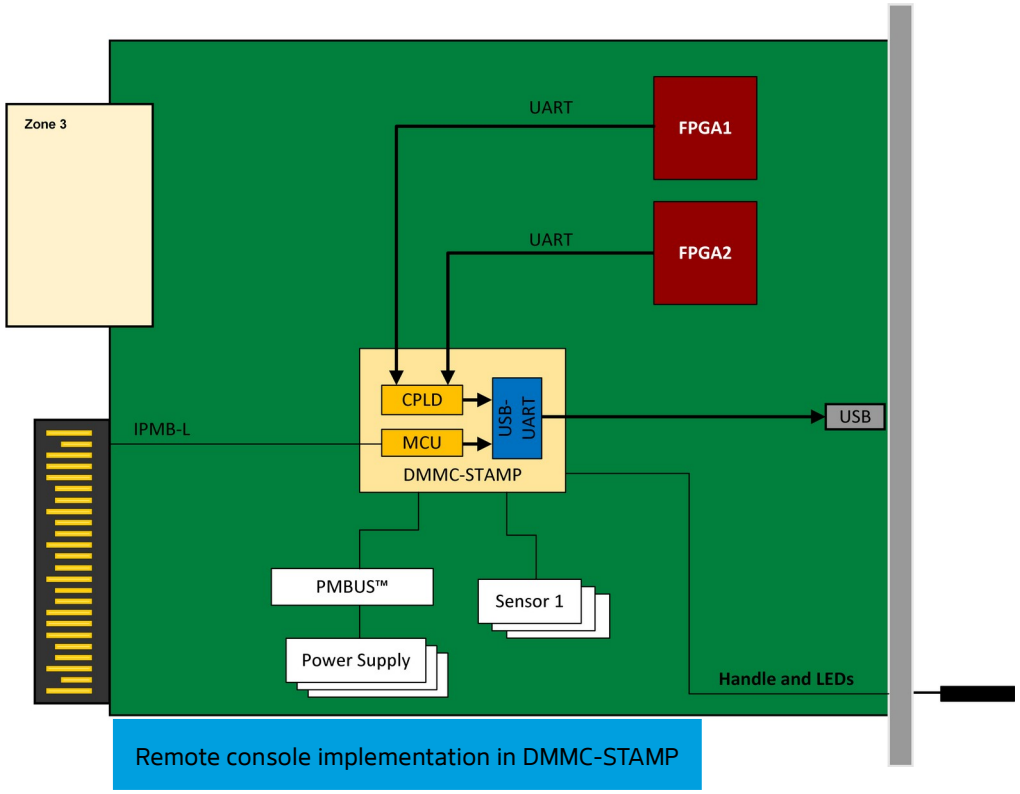
For serial console access of MMC and Payload SoC we have a USB-UART on the front panel.

But how to access the serial console remotely?

- Serial-over-IMPI would require MMC connected to Ethernet directly
- But in MicroTCA, Ethernet is part of payload, not management
  → DMMC-STAMP has no Ethernet connection, only IPMB(I2C)

Solution:

- Virtual UART implemented on DMMC-STAMP CPLD
- Custom IPMI protocol for serial port forwarding (Serial-over-IPMB)
- Open source tool: mmcterm
- Support for 3 channels (MMC, FPGA1, FPGA2)



Remote console implementation in DMMC-STAMP

Remote console with Serial-over-IMPB

```
      MCH URL IPMB addr
$ mmcterm mskmchscav2 0x7c -l List channels
channel 0: MMC Console
channel 1: MPSoC Console
channel 2: Kintex-7 Console

$ mmcterm mskmchscav2 0x7c -c 0 Open channel 0 (MMC)
Press Ctrl-x to exit
DAMC-MOTCTRL@0x7C MMC>v
App. version     : V2.01
 Build host, date: msktechjenkins.desy.de, 2023-03-29T11:15:41Z
 Compiler version: 10.2.1 20201103 (release)
Library version  : V2.08
 Build host, date: msktechjenkins.desy.de, 2023-03-29T11:15:41Z
 Compiler version: 10.2.1 20201103 (release)
IPMI version     : 1.5
Vendor ID        : 0x053F
Product ID       : 0x0710
Board            : DAMC-MOTCTRL
STAMP revision   : Rev. A
STAMP UID        : 801F1234DD71
Copyright (C) 2022 Deutsches Elektronen-Synchrotron (DESY)
DAMC-MOTCTRL@0x7C MMC>%

$ mmcterm mskmchscav2 0x7c -c 1 Open channel 1 (MPSoC)
Press Ctrl-x to exit

root@mskdamcmotctrl:~# uname -a
Linux mskdamcmotctrl 5.4.0-xilinx-v2020.2 #1 SMP Thu Sep 21 13:28:03 UTC 2023 aarch64 aa
root@mskdamcmotctrl:~# %
```

mmcterm usage example

mmcterm on GitHub

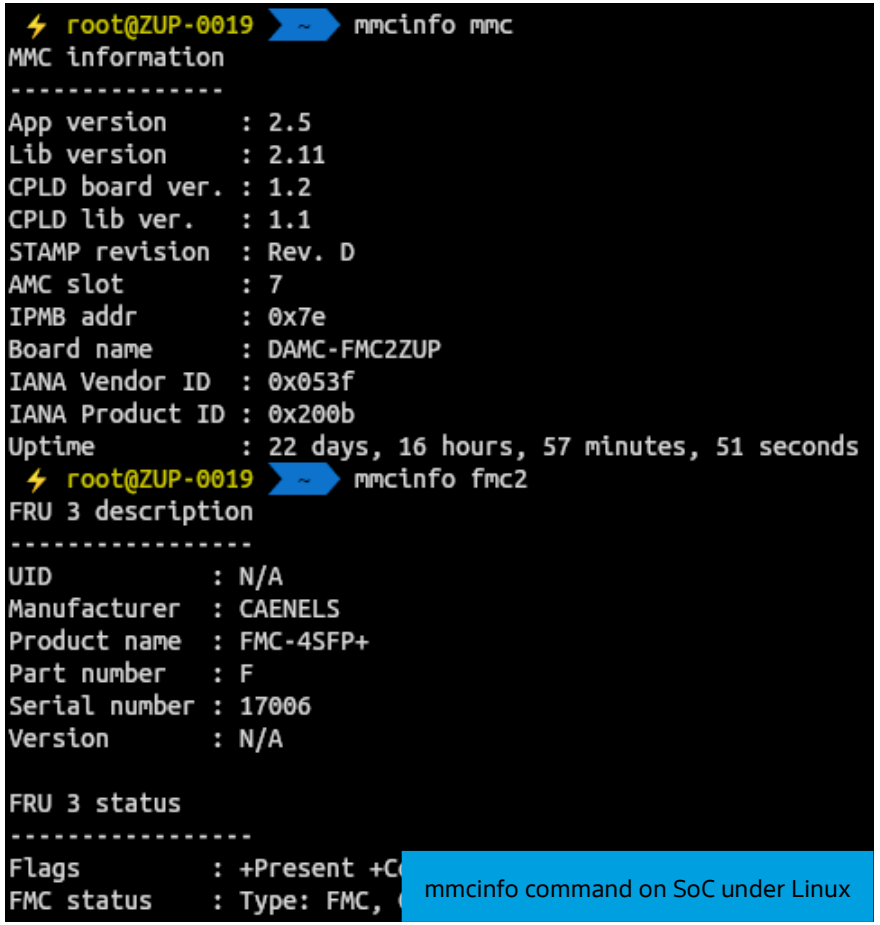
mmc-mailbox

Data interface between management (DMMC-STAMP) and payload (primary FPGA)

- Implemented as virtual I2C “EEPROM” for ease of access (U-Boot, etc.)

Use cases:

- Ethernet MAC Address from DMMC-STAMP UID
  (No dedicated EEPROM necessary)
- Orderly shutdown of Payload OS, when handle is pulled (M4 → M1)
- Propagation of management data
  (RTM/FMC FRU, sensor values, slot number, …) to payload side
- AMC Ethernet discovery
  MMC retrieves ethernet address from payload and exposes it over IPMI
- Application specific data transfer from/to MMC



mmcinfo command on SoC under Linux

Summary

- DMMC-STAMP is a drop-in solution for management of MicroTCA AMC boards
- FMC sensors are supported in compliance with ANSI/VITA 57.1
  → FMC temperatures are fully integrated into standard IPMI monitoring
  (i.e. can be monitored and trigger increase of cooling or AMC shutdown)
- In-system configuration of PMBUS components facilitates development & production
  Configuration of all PMBUS chips can be bundled with MMC code
- Remote console over Serial-over-IPMB provides useful remote debugging facilities for MMC itself and for payload FPGAs
- MMC mailbox allows data sharing between management (MMC) and payload (FPGA)

Thank you!

Acknowledgements: Michael Fenner, Carsten Dülsen

Contact

Patrick Huesmann
<patrick.huesmann@desy.de>

Deutsches Elektronen-Synchrotron DESY

MSK Department
Notkestraße 85
22607 Hamburg
www.desy.de