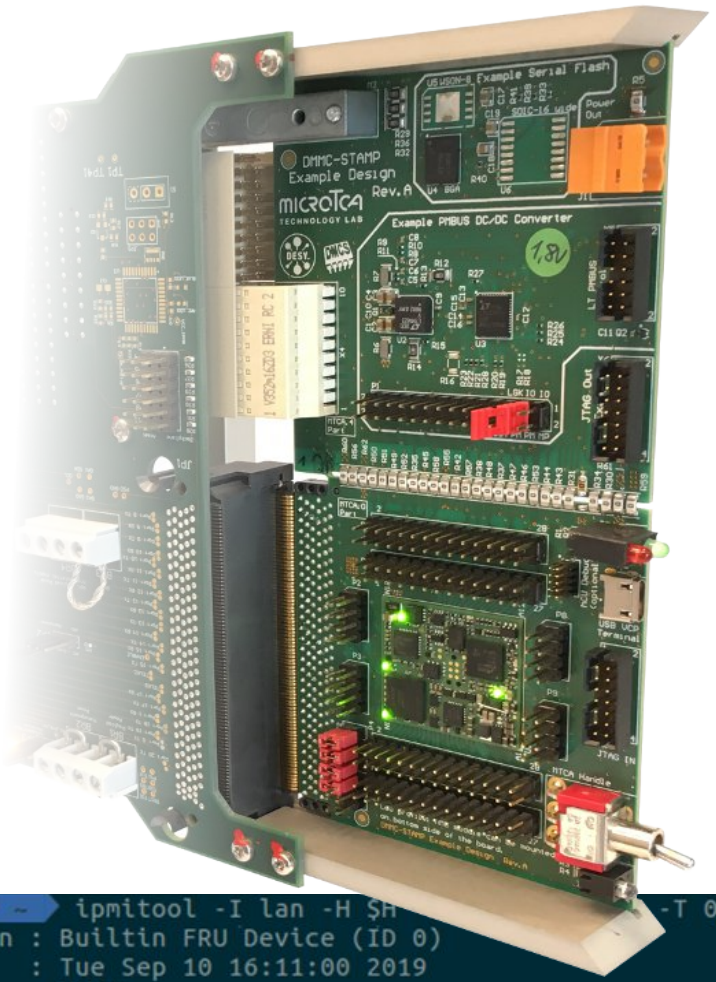



```
~ ipmitool -I lan -H $H                    -T 0x82 -t 0x7a fru
FRU Device Description : Builtin FRU Device (ID 0)
 Board Mfg Date        : Tue Sep 10 16:11:00 2019
 Board Mfg             : DESY
 Board Product         : DMMC-STAMP Rev.A
 Board Serial          : 0000
 Board Part Number     : 0000
 Product Manufacturer  : DESY
 Product Name          : DMMC-STAMP Rev.A
 Product Part Number   : 0000
 Product Version       : 0000
 Product Serial        : 0000
 Product Asset Tag     : none
```

DESY
Deutsches Elektronen-Synchrotron
Notkestr. 85 • 22607 Hamburg
mtca-techlab@desy.de
techlab.desy.de

MICROTCA TECHNOLOGY LAB
A HELMHOLTZ INNOVATION LAB

Revision history

Revision	Date	Modification
1.0	2019-09-20	Initial revision
1.1	2020-08-14	Updates and clarifications, factory programming section added
1.1.1	2020-12-10	Added trademarks (™ and ®)
2.0	2021-10-19	Renamed document to User Manual, changed CLI, added mmcterm description, added firmware description, minor fixes and improvements
2.1	2021-11-25	Added information regarding USB cable type, clarification of environmental conditions (incl. max. USB voltage) and stand-alone operation warnings. Moved PMBUS™ header pinout into separate chapter; Removed chapter “Restoring firmware defaults”

Authors: Jan Marjanovic, Michael Fenner, Sven Stubbe

Table of Contents

1 Operating Conditions

Operating Temperature	0 - 60°C
Minimum Airflow	None
Input Voltage (AMC payload power)	12 V nominal, 10 V -14 V range
Input Voltage (AMC management power)	3.3 V +/- 10%
USB Input Voltage (VBUS)	4.5 - 5.25 V
Output Voltage (External Connector)	1.8 V +/- 5%
Maximum Output Current (External Connector)	3 A
Maximum voltage on all Digital input Pins	VCC +0.3 V
Voltage on FPGAx supply pins	1.2V - 3.3 V
Maximum voltage on all analog input pins	1.0 V
Maximum output current on each LED pin	12 mA
Maximum output current on each MCU pin	3.2 mA
Maximum output current each CPLD pin	8 mA
Maximum output current on 3V3_MMC pin	100 mA

2 Introduction

DMMC-STAMP Breakout Board (DMMC-STAMP-BoB) is a cost-optimized development platform to demonstrate the capabilities and the typical usage of the DMMC-STAMP. It has and Advanced Mezzanine Card (AMC) formfactor and provides access to all LGA module pins via Standard 2.54 mm pin header for measurement and laboratory setups.

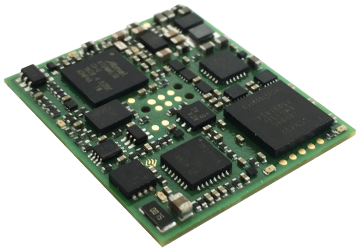


Figure 1: DMMC-STAMP module

The DMMC-STAMP is a small stamp-sized (26x30x2.5 mm) System on Module (SoM) which can be mounted on top or bottom side of any AMC. The DMMC-STAMP SoM provides a complete hardware and firmware solution to manage an AMC including power system, FPGAs/SoCs and Rear Transition Modules (RTMs). The firmware running on the DMMC-STAMP is well-tested and widely-deployed.

Using the DMMC-STAMP significantly reduces the effort of designing new AMCs for MicroTCA.4 platforms and eliminates the need for interoperability testing, e.g. with different MCH vendors.

Shown in Figure 2 are the main components of DMMC-STAMP-BoB:

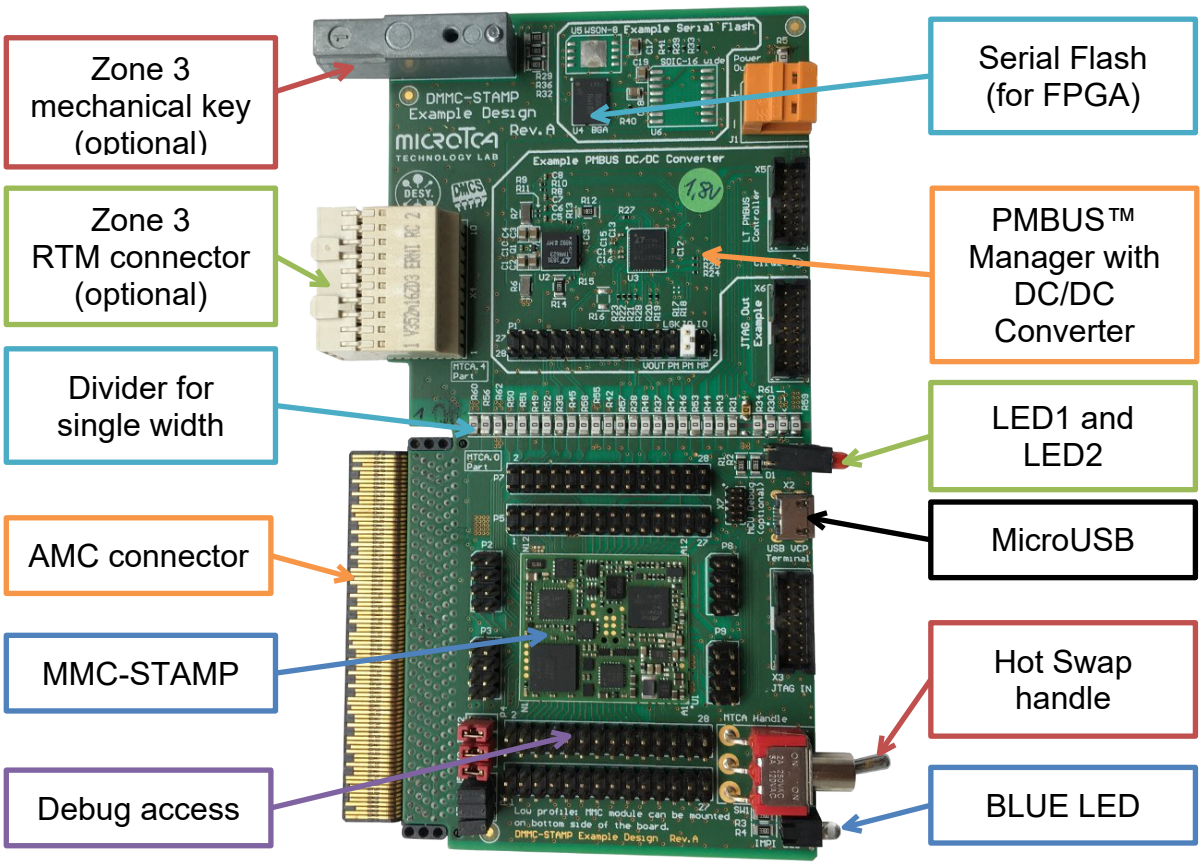


Figure 2: Main components of DMMC-STAMP Breakout Board (Rev. A)

DESY
Deutsches Elektronen-Synchrotron
Notkestr. 85 • 22607 Hamburg
mtca-techlab@desy.de
techlab.desy.de

The DMMC-STAMP-BoB provides all components that are mandatorily required by the MicroTCA standard such as IPMI LEDs and Hot-plug handle. Additional optional components that are used to demonstrate the capabilities of the DMMC-STAMP are listed here:

- A JTAG “input” is provided as an entry point to the board JTAG devices.
- A Micro-USB-B connector provides access to the MMC command line interface (virtual COM port). (See footnote 1)
- A 10-pin IDC connector with 1.27 mm pitch with standard ARM® pin assignment provides optional debug access to the DMMC-STAMP Microcontroller.
- An ERNI Zone3 connector allows to evaluate the MicroTCA.4 RTM management, power control and inrush limit functions.

Furthermore, some example peripherals have been added:

- A PMBUS™ controller do demonstrate the interaction of the DMMC-STAMP with a Power Manager.
- A DC/DC converter to demonstrate startup and measuring of an onboard voltage.
- A JTAG “output” to demonstrate JTAG access and arbitration from the JTAG “input” or from the backplane using a JTAG Switch Module (JSM).
- A SPI flash to demonstrate FPGA Flash programming via HPM

Figure 3 shows the described DMMC-STAMP-BoB components in a block diagram.

The DMMC-STAMP-BoB complies with PICMG AMC.0 Revision 2.0, PICMG MicroTCA.0 Revision 1.0 and PICMG MicroTCA.4 Revision 1.0.

Footnote 1: The DMMC-STAMP-BoB provides a standard-compliant USB interface on its Micro-USB-B connector. A conventional (standard-compliant) USB-to-Micro-USB cable, as it can be found on many industrial and consumer devices, is required for connecting to this port. Using special cables such as a proprietary passive Micro-USB to SUB-D9 adapter that is supplied with some MCHs will damage the board: These MCHs provide a classical +/- 12V RS232 serial interface using the USB connector, instead of using

the standard SUBD-D9 connector that is common for legacy RS232 interfaces.

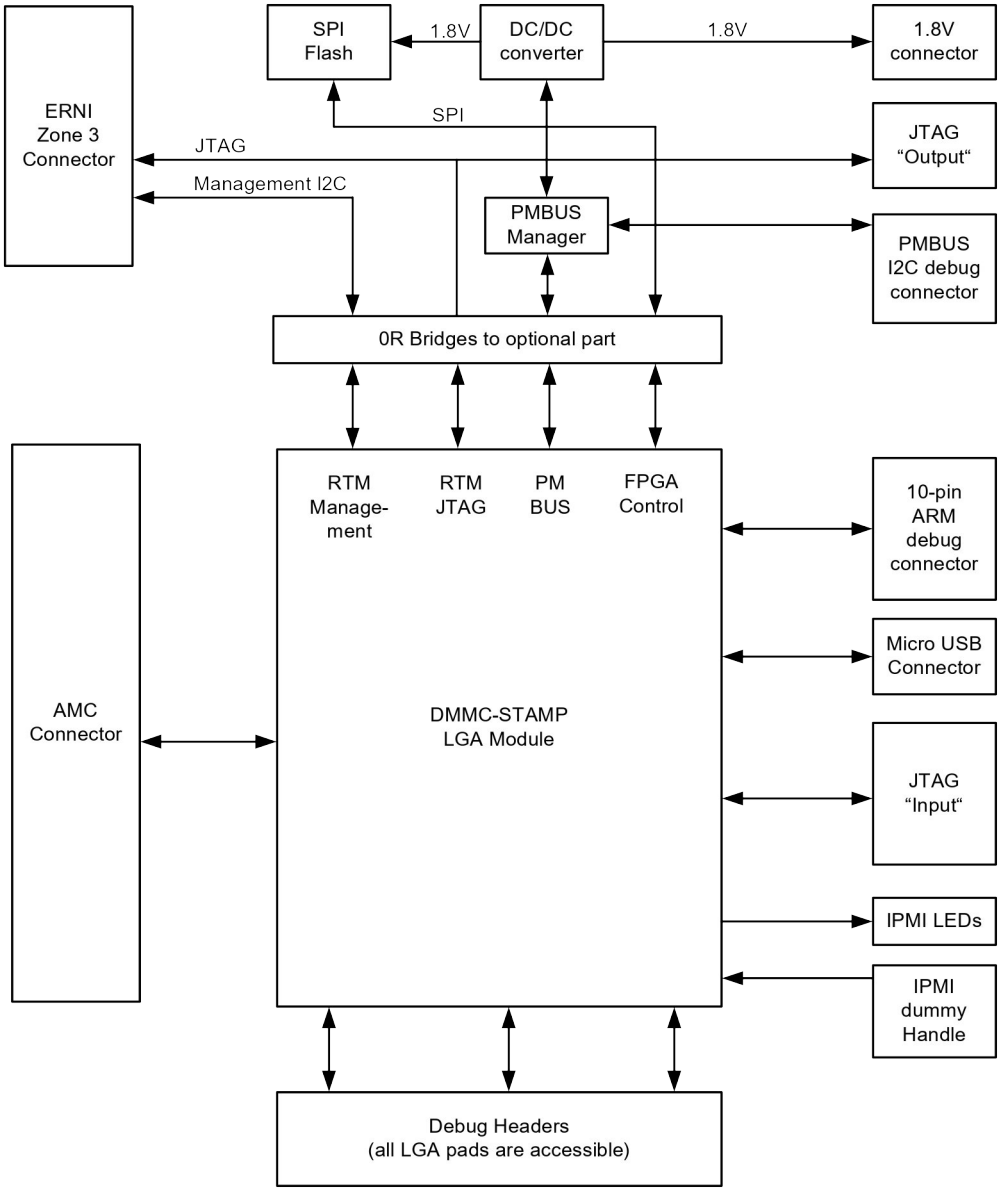


Figure 3: DMMC-STAMP Breakout Board Block Diagram

To provide a cost-optimized solution, the DMMC-STAMP-BoB is produced in a 1/3 Double Width form factor. It is assumed that DMMC-STAMP-BoB will be used for development, only. Therefore, it does not need to provide retention screws and a proper hot-plug handle.

Figure 4 shows the BoB mounted on an AMC extender kit, installed in a MicroTCA system.

Figure 4: DMMC-STAMP-BoB installed in a MicroTCA system

I case the DMMC-STAMP-BoB is not used with an AMC extender kit one need to take care when unplugging the board from the crate: Removing the BoB with force could result in a damaged ESD discharge pin.

3 Startup Procedure

This section describes how to connect to the DMMC-STAMP-BoB within a MicroTCA system and how to get a welcome message on the console.

The DMMC-STAMP-BoB can be operated in two different ways which can be select by jumper settings:

1. Operation within a MicoTCA system (default)
2. Operation in stand-alone on the desk

Figure 5 shows how the BoB is shipped and how the jumpers are set per default:

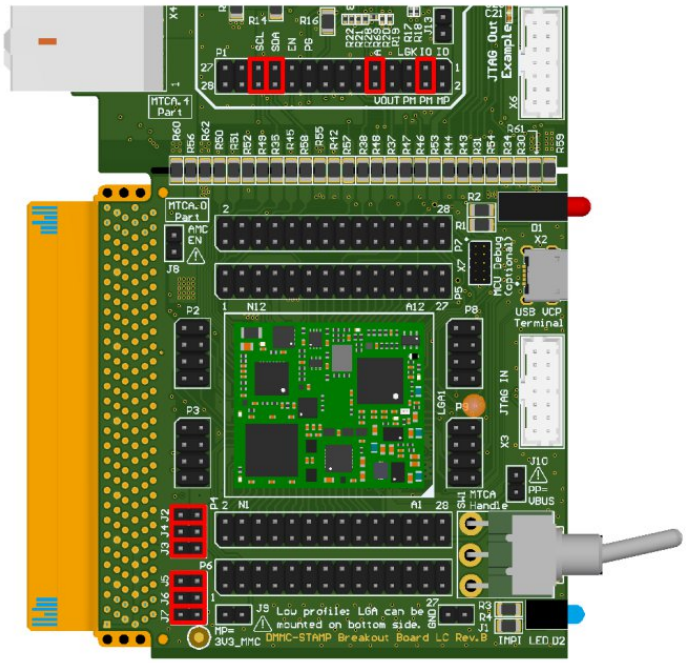


Figure 5: DMMC-STAMP-BoB default jumper settings (Rev. B)

To initially bring up the DMMC-STAMP-BoB plug the board into a MicroTCA chassis, activate the Hot-plug handle and turn the system on (see chapter 8 for operation in stand-alone mode). Connect a microUSB cable to the front panel and open a virtual COM port with settings 115200/8/N1 (see Appendix E or Appendix F). You should have access to the MMC console, now.

When issuing a reset (type “r” to the console) you should see the MMC welcome screen as shown below:

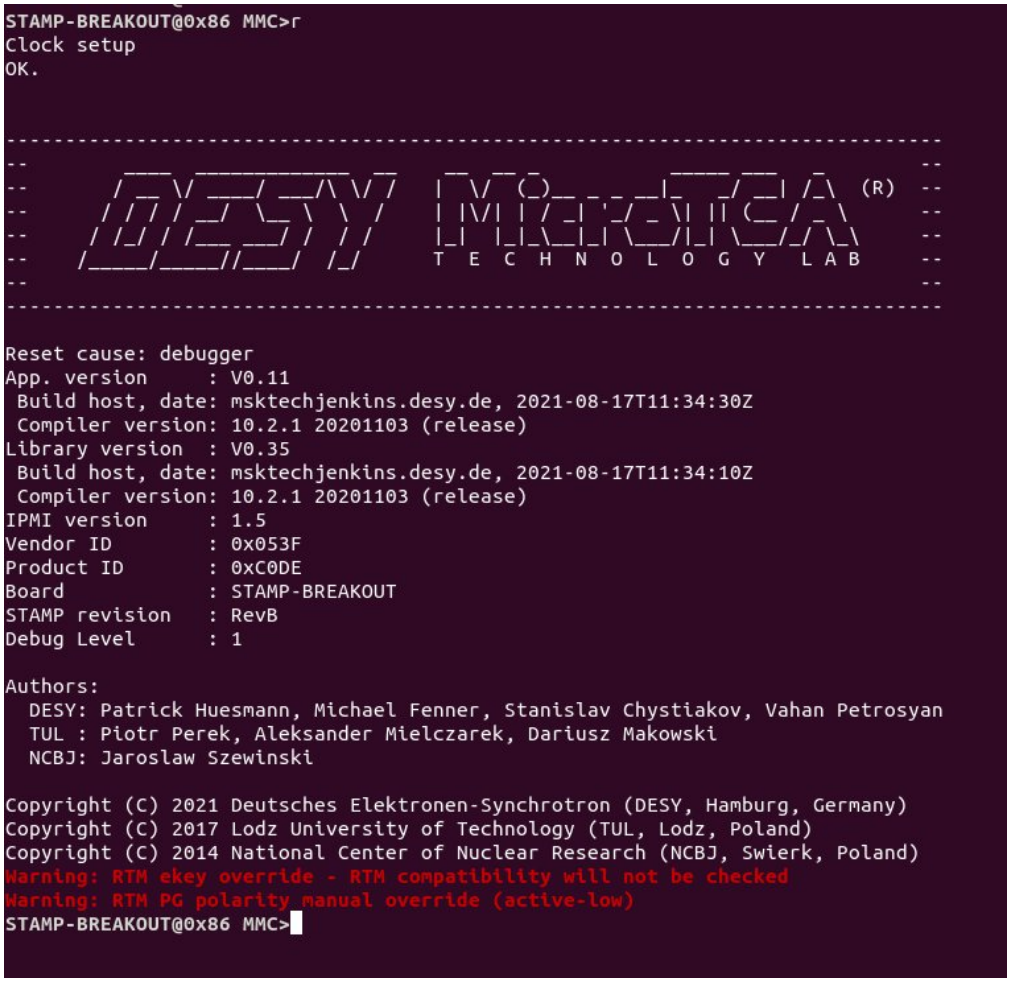


Figure 6: MMC console, welcome screen after reset (“r”)

4 MMC Console

The DMMC-STAMP on the DMMC-STAMP-BoB can be configured and monitored by using a Command Line Interface (CLI, MMC console). The CLI can be accessed either by opening a virtual COM port as described in the previous chapter or by using Serial over IPMI implemented in the open source tool **mmcterm** provided by the MicroTCA Technology Lab. Details on how to install and how to use mmcterm are given in Appendix B. Please refer to Appendix E and Appendix F for the explanation on how to set up serial connection on Windows® and GNU/Linux.

Once the MMC console is open users can type **?** to get a list with available commands (see below, the output of the command is trimmed to make it fit on the page). The MMC console comes with tab completion and integrated help functionality which gives hints during typing and simplifies operation.

```
STAMP-BREAKOUT@0x86 MMC>?
? / h / help: Show list of available commands
c ..................................... Clear screen
r ..................................... Reset MMC
v ..................................... Show firmware version
xm [0..n] ............................. Start XMODEM update
sb .................................... Start bootloader
vb [0..6] ............................. Get/set verbosity
tm [smart|dumb|auto] .................. Get/set terminal mode
s ..................................... Get status
lc [0..3] [on|off|blink] [on_ms] [off_ms] ... Set LED
ser [addr] [lun] ...................... Get/set event receiver
pu .................................... Payload power up
pd .................................... Payload power down
pc .................................... Toggle CPLD programming
sj [con|bp|raw] [fpga(1|2|12)|rtm|fmc(1|2)] . Get/set JTAG multiplexing
st [0..15] ............................ Get/set RTM temp sensor mask
rte [enable|override] ................. Get/set RTM e-keying policy
cfu ................................... CPLD force update
fru [0..n] ............................ Dump FRU information
rtp [auto|high|low] ................... Get/set RTM Power Good
i2cd [ipmb|sens|rtm|pmbus] ............ Detect I2C peripherals
pmc [verify|write] .................... Verify or write PMBUS PSM
sg [mmc|fpga1|fpga2|reset] [0..9] [i|o] [0|1] Set MMC GPIO direction
STAMP-BREAKOUT@0x86 MMC>
STAMP-BREAKOUT@0x86 MMC>? Show list of available commands
```

5 MMC Firmware

The DMMC-STAMP on the DMMC-STAMP-BoB will be shipped with DMMC-STAMP-BoB firmware preinstalled which provides complete as described in the sections 6 and 7. The DMMC-STAMP default firmware which is usually pre-installed when the DMMC-STAMP gets shipped was extended by the hardware functionality provided by the BoB. Basically, in a typical scenario, this DMMC-STAMP default firmware is suitable for the very basic user board operation without any modifications.

If users wish to adapt the pre-installed DMMC-STAMP default firmware for their own hardware, a dedicated Software Development Kit (SDK) is available. It allows implementation of custom functions with minimum need for non-board specific firmware coding. Useful features like custom I2C sensor readout, user-specific IPMI command implementation or PMBUSTM -based power control can be realized with the SDK. It provides a clear separation between the common codebase and board specific code. Firmware adaptions for newly developed boards are possible with minimal effort. The table below gives an overview about DMMC-STAMP functionalities which are available in the pre-installed default firmware.

Feature	**pre-installed firmware**	**SDK**
Full IPMI handling (MCH communication, LEDs, power, FRU)	yes	yes
Board failure handling (RTM over-current, over-temperature)	yes	yes
Re-configurable JTAG chain management	yes	yes
Serial-over-IPMI (remote access of CLI)	yes	yes
USB virtual COM port for MMC and FPGAs	yes	yes
RTM control	yes	yes
Basic FPGA control (PROG, INIT, DONE, RESET)	yes	yes
Basic sensors (temperature, voltages, FPGA done)	yes	yes
Control of up to two FMC modules	yes	yes
DMMC-Stamp (including CPLD) in-system-update via HPM.1	yes	yes
Custom FRU read/write (AMC/RTM)	yes	yes
Custom IPMI commands	no	yes[1]
Support of additional sensors on user-board	no	yes[1]
User FPGA in-system-update via HPM.1	no	yes[1]
DC/DC converter control	no	yes[1]
Complete AMC power management via PMBUS™	no	yes[1]
MAC address forwarding to user-FPGA	no	yes[1]
User-specific GPIO Pins control (MCU, CPLD)	no	yes[1]

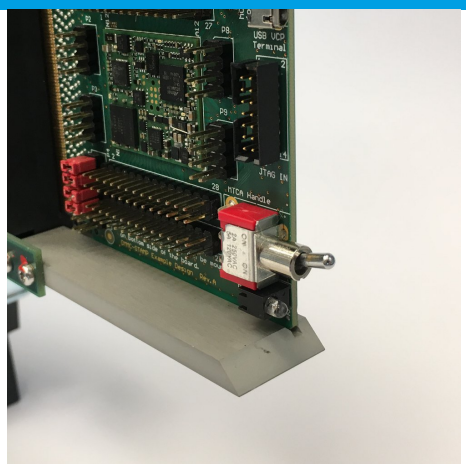
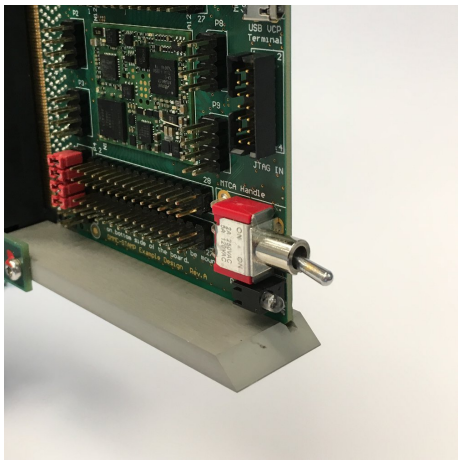
[1] Note: Requires board-specific instructions; SDK user is responsible for the implementation

6 Basic MicroTCA Management Features

This chapter introduces the basic MMC management features for AMCs that are required by the MicroTCA standard. These features are implemented in the default firmware where the DMMC-STAMP-BoB is already programmed with.

6.1 Hot-Swap Handle

For cost-optimization the three-position hot-swap handle (as described in Chapter “2.2.5 Module Handle” of PICMG AMC.0 standard) was replaced with a simple SPDT switch. This switch simulates hot-swap handle extracted when in UP position, and hot-swap handle inserted when in DOWN position (see table below):

	Proper hot-swap handle position	Switch on BoB equivalent
Extracted		
Inserted		

6.2 Field Replaceable Unit (FRU)

As already described the DMMC-STAMP comes pre-programmed with Field Replaceable Unit (FRU) Inventory populated. Stored in FRU Inventory are:

- board vendor and board name, board manufacturing date
- product vendor and product name, product manufacturing date
- PICMG module current requirements record

After inserting the board in MicroTCA crate these FRU information can be obtained from MCH e.g. by using the IPMI tool. Please refer to Appendix C for more information on how to use the `ipmitool` syntax:

```
$ ipmitool -I lan -H <hostname> -P "" -B 0 -b 7 -T 0x82 -t <fru-id> fru
FRU Device Description : Builtin FRU Device (ID 0)
 Board Mfg Date        : Tue Sep 10 14:11:00 2019
 Board Mfg             : DESY
 Board Product         : DMMC-STAMP Rev.A
 Board Serial          : 0000
 Board Part Number     : 0000
 Product Manufacturer  : DESY
 Product Name          : DMMC-STAMP Rev.A
 Product Part Number   : 0000
 Product Version       : 0000
 Product Serial        : 0000
 Product Asset Tag     : none
```

Another option to obtain FRU Inventory information is using the MCH console. The example below shows the output of `show_fru` command using an N.A.T. MCH. For more information about MCH commands please refer to vendor's documentation.

```
nat> show_fru

FRU Information:
---------------
 FRU  Device   State  Name
=========================================
  0   MCH      M4     NMCH-CM
  3   mcmc1    M4     NAT-MCH-MCMC
  5   AMC1     M4     CCT AM 902/411
  8   AMC4     M1     AFC 3.1
  9   AMC5     M4     DMMC-STAMP
 40   CU1      M4     Schroff uTCA CU
 50   PM1      M4     PM1 GW V20
 60   Clock1   M4     MCH-Clock
 61   HubMod1  M4     MCH-PCIe
=========================================
```

On N.A.T. MCHs the `show_fruinfo` command (with appropriate FRU ID) can be used to obtain more information about an individual board.

```
nat> show_fruinfo 9
---------------------------------------
FRU Info for device 9:
---------------------------------------
Common Header     : 0x01 0x00 0x00 0x01 0x07 0x0e 0x00 0xe9
---------------------------------------
Internal Use Area : -
---------------------------------------
Chassis Info Area : -
---------------------------------------
Board Info Area           : at offs=8, len=48
Manufacturer(04)          : DESY
Board Name(16)            : DMMC-STAMP Rev.A
Serial Number(04)         : 0000
Part Number(04)           : 0000
FRU file ID(04)           : none
---------------------------------------
Product Info Area         : at offs=56, len=56
Manufacturer(04)          : DESY
Product Name(16)          : DMMC-STAMP Rev.A
Product Number(04)        : 0000
Part Version(04)          : 0000
Product Serial Number(04): 0000
Asset Tag(04)             : none
FRU file ID(04)           : none
---------------------------------------
Multi Record Area  : at offs=112

Record(0): Type ID=0xc0, PICMG Record ID=0x16, offset=0x000, len=11
Module Current Requirements Record:
    Current Draw: 6.5 A
---------------------------------------
```

6.3 Sensors

Monitoring on-board temperature and voltage sensors is one of the most important features of the MMC in MicroTCA. The DMMC-STAMP SoM LGA module has built-in sensors for observing:

- Temperature
- 3.3 V Management Power
- 12 V Payload Power
- RTM currents and voltages (12 V / 3.3 V MP levels, currents and power goods)
- 8 User ADC inputs (0 V to 1 V)
- Unique ID

The Breakout Board has additional sensors for observing:

- PMBUS™ manager internal temperature
- PMBUS™ controller external temperature diode
- 1.8 V voltage level
- 1.8 V current

Shown in Figure 7 is the layout of sensors on the DMMC-STAMP-BoB. There are three separate I2C busses available on DMMC-STAMP for sensors (SENS_I2C), PMBUS™ and partially also RTM_I2C.

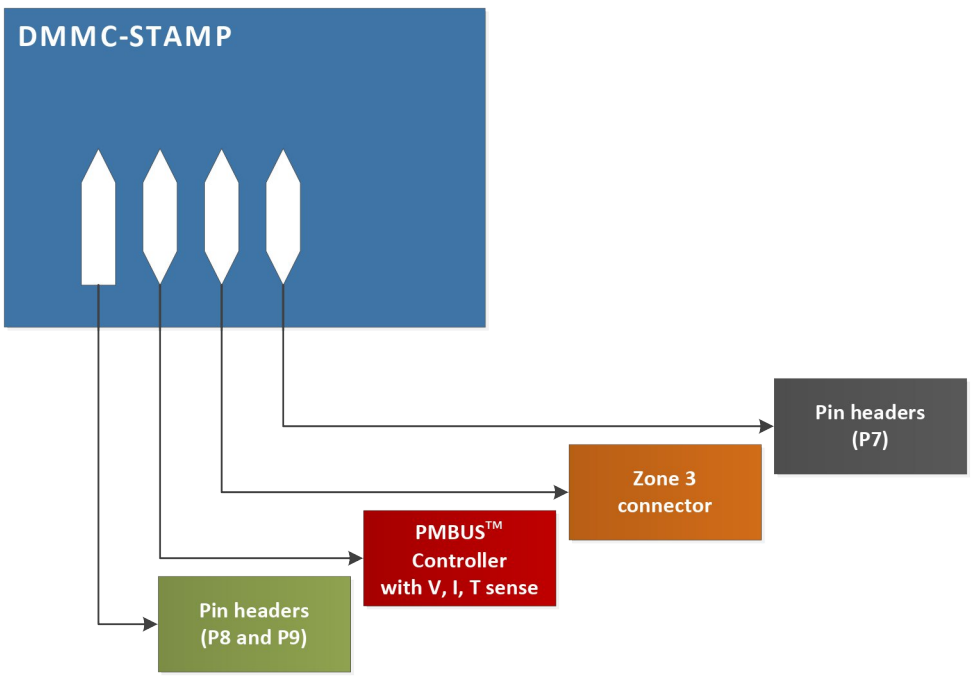


Figure 7: DMMC-STAMP sensor interfaces

Data read back from the sensors can be obtained with `ipmitool` together with the `sensor` command. Here the thresholds are shown as well (the output of the command is trimmed to make it fit on the page):

```
$ ipmitool -I lan -H <hostname> -P "" -B 0 -b 7 -T 0x82 -t <fru-id> sensor
Hot Swap         | 0x0       | discrete   | 0x0100| na        | na        | na        | na        |
STAMP TEMP       | 25.000    | degrees C  | ok    | 0.000     | 3.000     | 5.000     | 60.000    |
3.3 V            | 3.280     | Volts      | ok    | 2.496     | 2.992     | 3.088     | 3.488     |
12 V             | 12.224    | Volts      | ok    | 5.952     | 9.984     | 10.944    | 12.992    |
ADC0             | 1.000     | Volts      | ok    | 0.800     | 0.895     | 0.945     | 1.045     |
ADC1             | 1.000     | Volts      | ok    | 0.800     | 0.895     | 0.945     | 1.045     |
ADC2             | 1.000     | Volts      | ok    | 0.800     | 0.895     | 0.945     | 1.045     |
I RTM 3V3MP      | 0.007     | Amps       | ok    | 0.000     | 0.000     | 0.000     | 0.042     |
I RTM 12VPP      | 0.000     | Amps       | ok    | 0.000     | 0.000     | 0.000     | 2.196     |
3.3V PMBUS volt. | 1.792     | Volts      | ok    | 2.496     | 2.736     | 2.992     | 3.488     |
3.3V PMBUS curr. | 0.000     | Amps       | ok    | 0.000     | 0.000     | 0.000     | 1.664     |
3.3V PMBUS Temp. | 28.000    | degrees C  | ok    | na        | na        | na        | na        |
PMBUS Board Temp | 26.500    | degrees C  | ok    | na        | na        | na        | na        |
801F127C4431     | 0x0       | discrete   | 0x0000| na        | na        | na        | na        |
PMBUS PGood      | 0x0       | discrete   | 0x0000| na        | na        | na        | na        |
FPGA1 INIT       | 0x1       | discrete   | 0x0000| na        | na        | na        | na        |
FPGA1 DONE       | 0x1       | discrete   | 0x0000| na        | na        | na        | na        |
RTM 3V3 PGood    | 0x1       | discrete   | 0x0000| na        | na        | na        | na        |
RTM 12V PGood    | 0x0       | discrete   | 0x0000| na        | na        | na        | na        |
```

The sensor reading can also be obtained from MCH console, e.g. on N.A.T. MCHs by using the command `show_sensorinfo`, followed by FRU ID:

```
nat> show_sensorinfo 9
Sensor Information for FRU 9 / AMC5
=================================================================
  #    SDRType   Sensor Entity Inst  Value    State    Name
-----------------------------------------------------------------
  -    MDevLoc           0xc1  0x65                   DMMC-STAMP
  0    Full      0xf2    0xc1  0x65  0x01             Hot Swap
  1    Full      Temp    0xc1  0x65  31.0 C     ok    STAMP TEMP
  2    Full      Voltage 0xc1  0x65  3.360 V    ok    3.3 V
  3    Full      Voltage 0xc1  0x65  12.480 V   ok    12 V
  4    Full      Voltage 0xc1  0x65  1.000 V    ok    ADC0
  5    Full      Voltage 0xc1  0x65  1.000 V    ok    ADC1
  6    Full      Voltage 0xc1  0x65  1.000 V    ok    ADC2
  7    Full      Current 0xc1  0x65  0.0099 A   ok    I RTM 3V3MP
  8    Full      Current 0xc1  0x65  0.120 A    ok    I RTM 12VPP
  9    Full      Voltage 0xc1  0x65  1.792 V    ok    3.3V PMBUS volt.
 10    Full      Current 0xc1  0x65  0.000 A    ok    3.3V PMBUS curr.
 11    Full      Temp    0xc1  0x65  31.00 C    ok    3.3V PMBUS Temp.
 12    Full      Temp    0xc1  0x65  29.00 C    ok    PMBUS Board Temp
 13    Compact   0x0b    0xc1  0x65  0x00             0x00  801F127C3A3B
 14    Compact   0x14    0xc1  0x65  0x00             0x00  PMBUS PGood
 15    Compact   0x14    0xc1  0x65  0x01             0x00  FPGA1 INIT
 16    Compact   0x14    0xc1  0x65  0x01             0x00  FPGA1 DONE
 17    Compact   0x14    0xc1  0x65  0x01             0x00  RTM 3V3 PGood
 18    Compact   0x14    0xc1  0x65  0x01             0x00  RTM 12V PGood
 19    Compact   0xf0    0xc1  0x65  0x10             HS 009 AMC5
-----------------------------------------------------------------
```

6.4 Front Panel LEDs

There are 3 LEDs on the front panel as required by “PICMG Advanced Mezzanine Card AMC.0 Specification R2.0”, described in detail in chapter “2.2.4.1 Module LEDs”. The *BLUE LED* indicates the Hot Swap status of the module, *LED 1* (red) indicates Out of Service status and *LED 2* (green) indicates an operational board.

All LEDs are controlled directly from DMMC-STAMP and no additional components or logic is required on DMMC-STAMP-BoB.

Because the DMMC-STAMP-BoB can be used as a MicroTCA.0 board as well, the placement of *LED 1* and *LED 2* corresponds to a MicroTCA.0 card and does not comply with the AMC.0 requirements “REQ 2.149” and “REQ 2.151” – the LEDs **shall** be placed at the top part of the board. The *BLUE LED* is placed correctly.

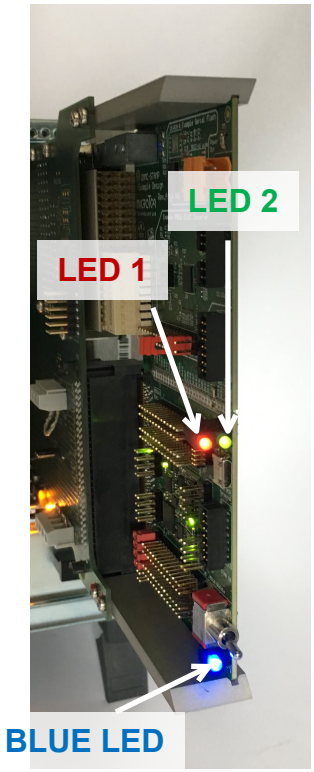


Figure 8: MicroTCA frontpanel LEDs

6.4.1 Manual control of the LEDs

The “PICMG AMC.0” standard defines several IPMI commands which are mandatory for MMC and can be used to remotely control module LEDs: “Get FRU LED Properties”, “Get LED Color Capabilities”, “Set FRU LED State”, and “Get FRU LED State”.

`ipmitool` can be used to obtain the LED information:

```
$ ipmitool -I lan -H <h> -P "" -B 0 -b 7 -T 0x82 -t <fru-id> picmg led get
0 0
LED states:                                3
  Local Control function:      0   [OFF]
  Local Control On-Duration:   0
  Local Control Color:         1   [BLUE]
$ ipmitool -I lan -H <h> -P "" -B 0 -b 7 -T 0x82 -t <fru-id> picmg led get
0 1
LED states:                                1   [LOCAL CONTROL]
  Local Control function:      0   [OFF]
  Local Control On-Duration:   0
  Local Control Color:         2   [RED]
$ ipmitool -I lan -H <h> -P "" -B 0 -b 7 -T 0x82 -t <fru-id> picmg led get
0 2
LED states:                                1   [LOCAL CONTROL]
  Local Control function:      0   [OFF]
  Local Control On-Duration:   0
  Local Control Color:         3   [GREEN]
```

The LEDs can be also controlled via IPMI (e.g. for board identification purposes). The following command makes the *LED 2* (green) blink with 1 Hz (500 ms on, 500 ms off):

```
$ ipmitool -I lan -H <hostname>  -P "" -B 0 -b 7 -T 0x82 -t <fru-id> \
picmg led set 0 2 50 50 0xF
```


DESY
Deutsches Elektronen-Synchrotron
Notkestr. 85 • 22607 Hamburg
mtca-techlab@desy.de
techlab.desy.de

7 Board Specific MicroTCA Management Features

7.1 Front Panel USB

Additionally, the DMMC-STAMP provides a USB connection both to the MMC itself and to further payload components, such as FPGAs/SoCs, CPUs and DSPs. The on-board USB-to-UART bridge (SiLabs CP2105) provides a virtual COM port to the microcontroller on the DMMC-STAMP and to the FPGA (not connected on the DMMC-STAMP-BoB).

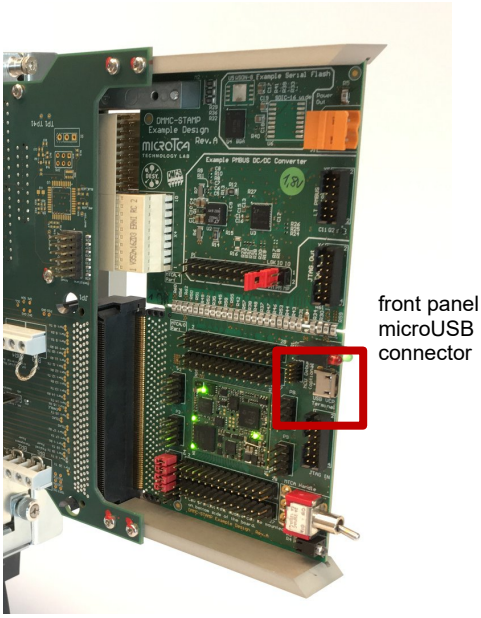


Figure 9: Front panel microUSB connector

To use the virtual COM ports, the BoB provides a microUSB connector on the front panel. To connect to the virtual COM ports, use the following parameters:

- Baud rate: 115200 bps
- Data Bits: 8
- Stop Bits 1 (8N1)
- Parity: None
- Flow Control: None

A Command Line Interface (CLI, MMC console) is provided for debugging and development purposes. Details on how to use the MMC console are given in section 4.

7.2 Rear Transition Module

The DMMC-STAMP-BoB contains one ERNI 30-pin connector for connecting Rear Transition Module (RTM) devices. Only power (management power, 12 V payload power), management (presence, I2C) and JTAG signals are connected to make this board compatible with both Class A1 and Class D1 of the "Zone 3 Connector Pin Assignment Recommendation“.

(https://techlab.desy.de/support/zone_3_recommendation/index_eng.html).

Mounted by default on the BoB is the mechanical key TE CONNECTIVITY 5223986-1 with 0 degree angle (N=1, LVDS as defined in Table 2-2 in “PICMG® MicroTCA.4 Enhancements for Rear I/O and Timing R1.0”). This makes the BoB compatible with Class D1. As the basic signals are connected, only, the mechanical key can be replaced by the user with the appropriate mechanical key for Class A1.

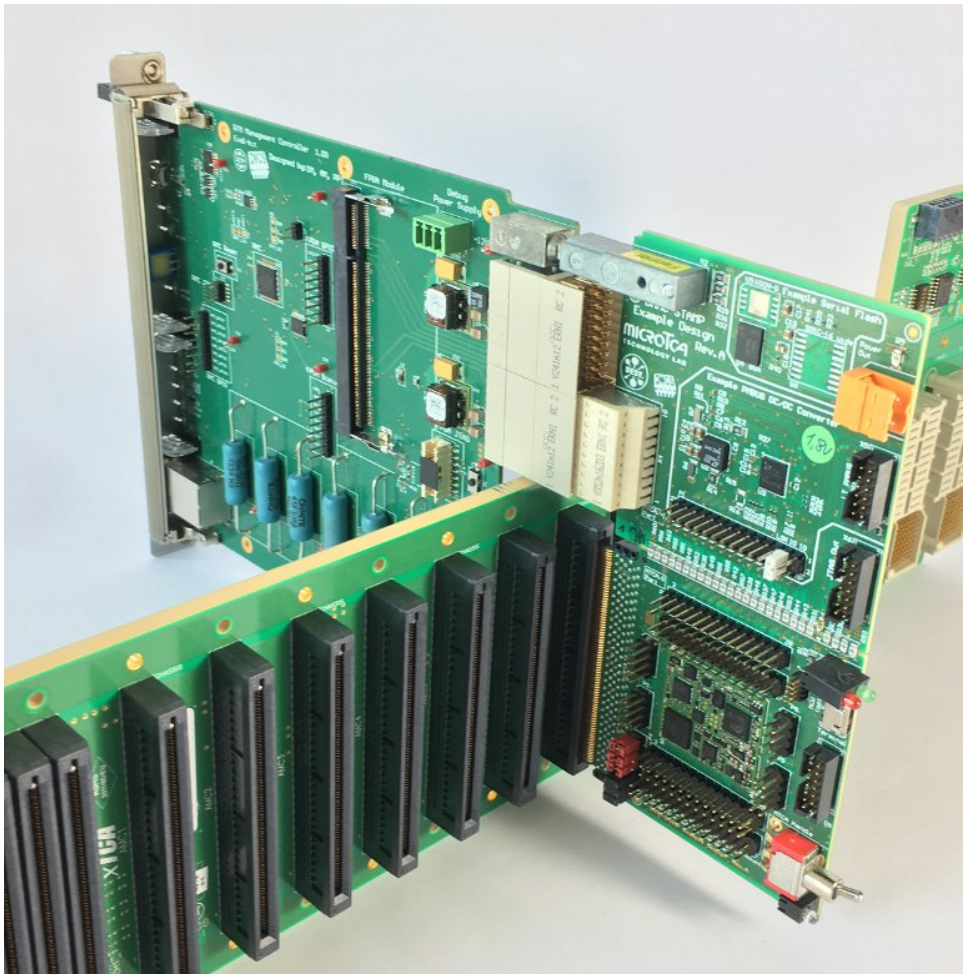


Figure 10: DMMC-STAMP-BoB with RTM conncted (backplane outside of the crate)

When both BoB and RTM are inserted, both boards are visible to the MCH, as shown on the listing below (the AMCs are available with FRU Device IDs from 5-39, while the RTMs are available with FRU Device IDs 90-124, as specified in chapter “3.3 FRU Device IDs” of “PICMG® MicroTCA.4 Enhancements for Rear I/O and Timing R1.0”):

```
nat> show_fru

FRU Information:
----------------
 FRU  Device    State  Name
=========================================
  0   MCH        M4    NMCH-CM
  3   mcmc1      M4    NAT-MCH-MCMC
  5   AMC1       M4    CCT AM 902/411
  9   AMC5       M4    DMMC-STAMP
 40   CU1        M4    Schroff uTCA CU
 50   PM1        M4    PM1 GW V20
 60   Clock1     M4    MCH-Clock
 61   HubMod1    M4    MCH-PCIe
 95   AMC6-RTM   M4    DMMC-STAMP RTM
=========================================
```

We can then obtain the FRU information about the RTM plugged into BoB:

```
nat> show_fruinfo 95
---------------------------------------
FRU Info for device 95:
---------------------------------------
Common Header     : 0x01 0x00 0x00 0x00 0x01 0x0a 0x00 0xf4
---------------------------------------
...
---------------------------------------
Product Info Area         : at offs=8, len=72
Manufacturer(09)          : DESY/DMCS
Product Name(10)          : DRTM-VM2LF
Product Number(03)        : 1.0
Part Version(03)          : 2.3
Product Serial Number(12): 001EC0FC5CC9
Asset Tag(02)             : --
FRU file ID(18)           : FRU_DM: 2015.09.29
---------------------------------------
Multi Record Area   : at offs=80

Record(0): Type ID=0xc0, PICMG Record ID=0x16, offset=0x000, len=11
Module Current Requirements Record:
    Current Draw: 2.5 A

Record(1): Type ID=0xc0, PICMG Record ID=0x30, offset=0x00b, len=10
Zone 3 Interface Compatibility record:
IRTM.0 REP number: 0x32200000
---------------------------------------
```

And similarly, sensors (such as temperature of the RTM) can also be monitored:

```
nat> show_sensorinfo 95
Sensor Information for FRU 95 / RTM6
=====================================================================
  #    SDRType   Sensor Entity Inst  Value    State     Name
---------------------------------------------------------------------
  -    FDevLoc           0xc0  0x66                    DMMC-STAMP RTM
 32    Full      0xf2    0xc0  0x66  0xa1              RTM Hot Swap
 33    Compact   0x0b    0xc0  0x66  0x00              0x00  801F127C3A3B
 34    Full      Temp    0xc0  0x66  22.5 C     ok     Temp 1
 35    Compact   0xf0    0xc0  0x66  0x10              HS 095 RTM6
---------------------------------------------------------------------
```

The here connected RTM load board was configured to draw 1 A from the RTM payload power rail, and the following sensor information from BoB were obtained (some lines are removed for clarity):

```
nat> show_sensorinfo 9
Sensor Information for FRU 9 / AMC5
=====================================================================
  #    SDRType   Sensor Entity Inst  Value    State     Name
---------------------------------------------------------------------
  -    MDevLoc           0xc1  0x65                    DMMC-STAMP
  0    Full      0xf2    0xc1  0x65  0x01              Hot Swap

...

  7    Full      Current 0xc1  0x65  0.0648 A   ok     I RTM 3V3MP
  8    Full      Current 0xc1  0x65  0.896 A    ok     I RTM 12VPP

...

 17    Compact   0x14    0xc1  0x65  0x01              0x00  RTM 3V3 PGood
 18    Compact   0x14    0xc1  0x65  0x01              0x00  RTM 12V PGood
 19    Compact   0xf0    0xc1  0x65  0x10              HS 009 AMC5
---------------------------------------------------------------------
```

It can be noted that the measurement of the current on RTM 12V payload power measures 0.896A. The current measurement is not very precise (i.e. there is a 10% error), but it nonetheless allows detection of a faulty RTM (with excessive current consumption) during the operation.

7.3 JTAG subsystem

Usually, an Advanced Mezzanine Card (AMC) has one or more components which can be programmed or controlled over a JTAG interface. Most commonly these devices are FPGAs/SoCs, DSPs or CPUs. Prescribed in "Zone 3 Connector Pin Assignment Recommendation" is also a JTAG interface towards the RTM.

An AMC usually contains a JTAG connector and can also accept JTAG signals from the backplane. Shown in Figure 11 are the main components of the JTAG subsystem on the DMMC-STAMP-BoB. There are two inputs (from AMC and from on-board connector JTAG IN), and two outputs (to RTM, and when the RTM is not present, the signals are routed to JTAG OUT connector).

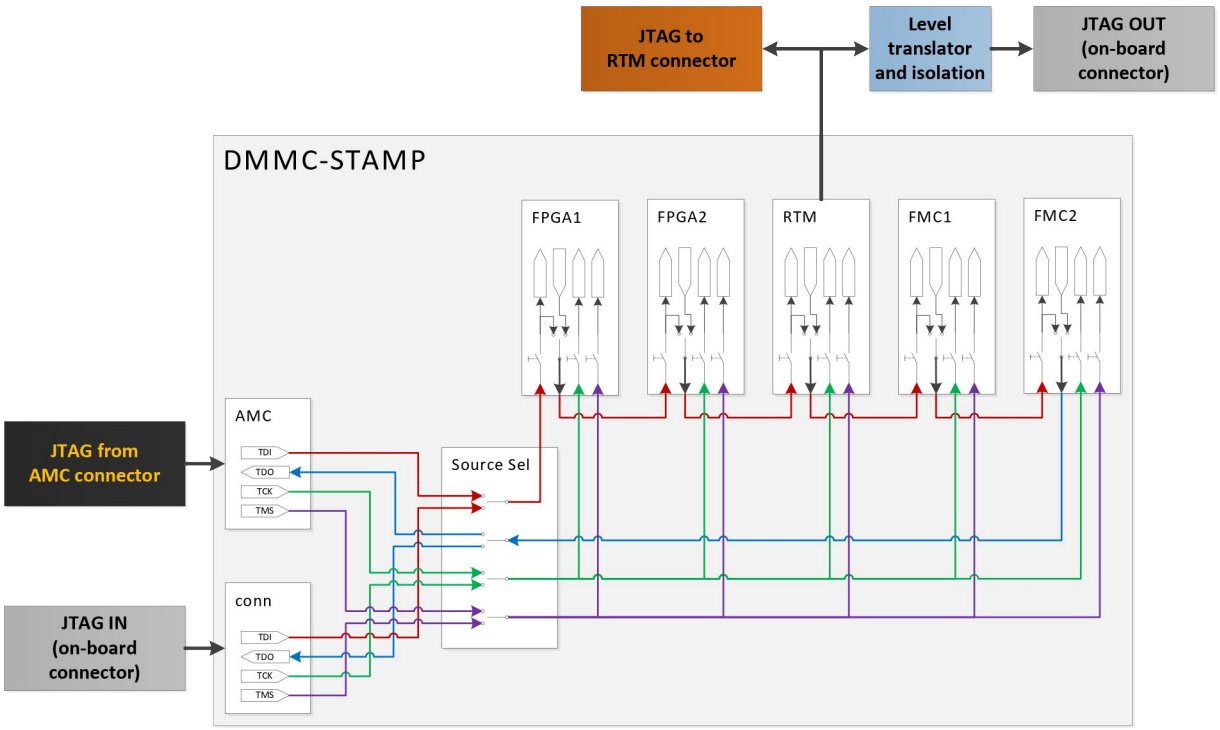


Figure 11: JTAG subsystem on the DMMC-STAMP-BoB

7.3.1 JTAG System Control

The JTAG mux on the DMMC-STAMP can be either controlled through IPMI messages (send to board over MCH with, for example, `ipmitool`) or through the MMC console. A detailed description of the registers can be found in the DMMC-STAMP User Manual.

7.3.1.1 Control through IPMI

This interface will be described in next release of this Quick Start Guide.

7.3.1.2 Control through MMC console

The **JTAG IN** connector by default connects to CPLD on the DMMC-STAMP, but can be configured to forward the JTAG interface to other components on the board. Use the `pc` command to put the CPLD into forward mode:

```
STAMP-BREAKOUT@0x86 MMC>
STAMP-BREAKOUT@0x86 MMC>vb 3
verbosity: 3 (INFO)
STAMP-BREAKOUT@0x86 MMC>pc
Setting CPLD to ISP (JTAG) mode (JTAGENB:=HIGH)
STAMP-BREAKOUT@0x86 MMC>pc
Setting CPLD to forward mode (JTAGENB:=LOW)
STAMP-BREAKOUT@0x86 MMC>
```

Attention: Please ensure that the verbosity log level is set to 3 (vb 3, at least) to get a proper feedback when running these commands.

Similarly, the JTAG subsystem can be configured to route the signals from sources:

- con – on-board JTAG connector
- bp – backplane
- raw – use for writing raw values

to one of the available components on the board:

- fpga1, fpga2 or both (fpga12)
- rtm
- fmc1 or fmc2

This configuration task is performed with `sj` command with the following syntax:

```
STAMP-BREAKOUT@0x86 MMC>
STAMP-BREAKOUT@0x86 MMC>sj con fpga1
EEPROM value = 0x1
source: con, dest: fpga1
STAMP-BREAKOUT@0x86 MMC>sj bp fpga1
EEPROM value = 0x21
source: bp, dest: fpga1
STAMP-BREAKOUT@0x86 MMC>
```

7.3.1.2.1 Example 1: On-board connector to JTAG OUT

To configure the JTAG subsystem to accept the signals from JTAG IN connector and forward the JTAG signals to **JTAG OUT** connector (when the RTM is not connected), use the following command:

```
STAMP-BREAKOUT@0x86 MMC>sj con rtm
EEPROM value = 0x4
source: con, dest: rtm
```

This hardware configuration is shown in Figure 12:

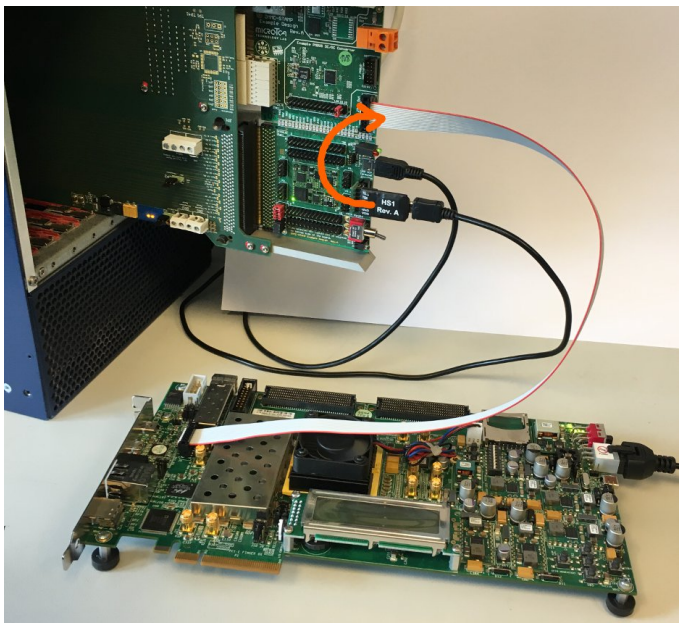


Figure 12: JTAG IN to JTAG OUT

7.3.1.2.2 Example 2: AMC connector to JTAG OUT

To configure the JTAG subsystem to accept the signals from AMC backplane and forward the JTAG signals to **JTAG OUT** connector (when the RTM is not connected), use the following command:

```
STAMP-BREAKOUT@0x86 MMC>sj bp rtm
EEPROM value = 0x24
source: bp, dest: rtm
```

This hardware configuration is shown in Figure 13:

Figure 13: AMC JTAG to JTAG OUT

7.3.2 JTAG subsystem demonstration

Presented in this chapter is a use-case where the JTAG signals are connected from a JTAG Switch Module (JSM) in a MicroTCA crate over the backplane through the DMMC-STAMP-BoB to a Xilinx® evaluation kit, VC707 with Virtex-7 FPGA in this case.

Shown in Figure 14 is the demo setup. The BoB is mounted on the extender board and inserted in MicroTCA crate with a MCH from N.A.T. Not visible on this photo is JSM, inserted in the crate on the rear side. The JTAG OUT from the Breakout Board is connected to the JTAG connector on the Xilinx® evaluation kit.

Steps regarding the MCH, its interfaces and configuration are described briefly, only. Please consult vendor documentation for more details.

Figure 14: Hardware setup for demonstrating the JTAG subsystem

First, we need to determine the slots where the board is placed. As shown on the listing below, the DMMC-STAMP is (in this example) inserted in slot AMC3.

```
nat> show_fru

FRU Information:
---------------
 FRU  Device    State  Name
=========================================
  0   MCH        M4    NAT-MCH-CM
  3   mcmc1      M4    NAT-MCH-MCMC
  5   AMC1       M4    CCT AM G64/471
  7   AMC3       M4    DMMC-STAMP
 11   AMC7       M4    SIS8160 AMC
 13   AMC9       M4    DAMC-TCK7
 40   CU1        M4    Schroff uTCA CU
 41   CU2        M4    Schroff uTCA CU
 51   PM2        M4    PM2 GW V20
 53   PM4        M4    PM4 GW V20
 60   Clock1     M4    MCH-Clock
 61   HubMod1    M4    MCH-PCIe
=========================================
```

Then we open Web interface of the MCH and open the JSM page. Shown in Figure 15 is the relevant page on the N.A.T. MCH with the TCP port number corresponding to the AMC slot annotated in red.

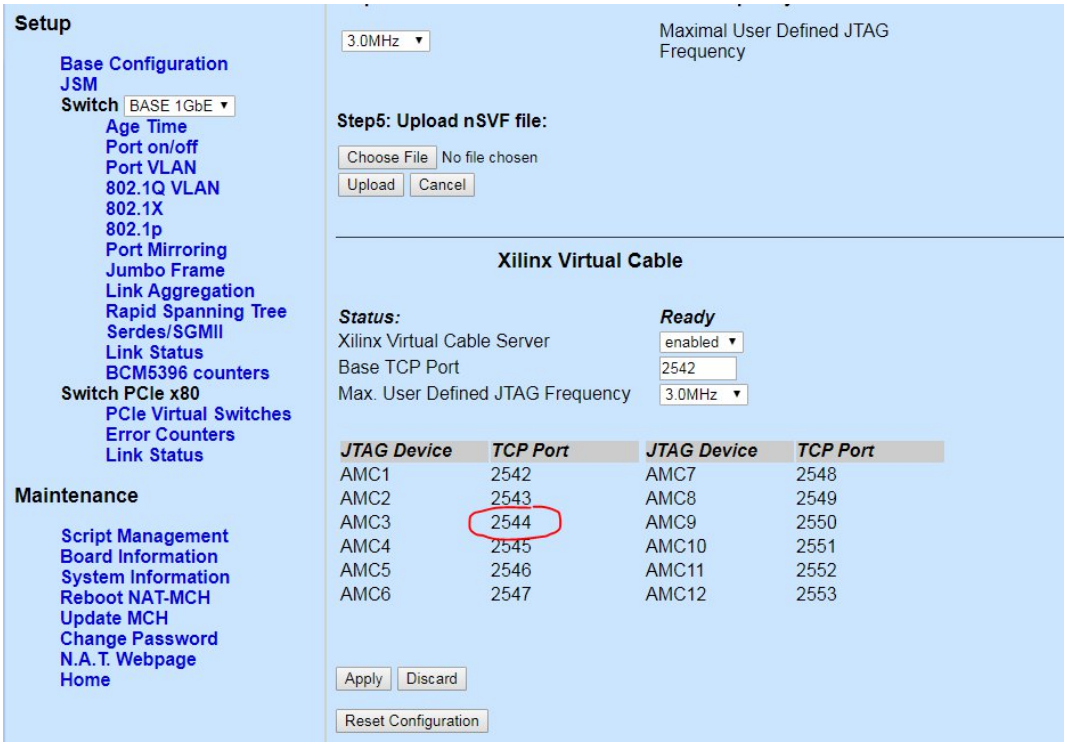


Figure 15: JSM page on N.A.T. MCH Web interface

We can observe that the TCP port corresponding to the slot AMC3 is 2544. We can then use this port when connecting to the MCH. Open Xilinx® Vivado™, click “Open Hardware Manager”, and run the following two commands in the Tcl:

```
connect_hw_server
open_hw_target -xvc_url <MCH-hostname>:<tcp-port>
```

where `<MCH-hostname>` is the hostname or IP address of the MCH and `<tcp-port>` is the port address obtained from the web interface.

The following listing shows an example of successful programming of the evaluation kit through MCH, JTAG Switch Module and DMMC-STAMP-BoB:

```
> connect_hw_server
INFO: [Labtools 27-2285] Connecting to hw_server url TCP:localhost:3121
INFO: [Labtools 27-2222] Launching hw_server...
INFO: [Labtools 27-2221] Launch Output:

****** Xilinx hw_server v2019.1
  **** Build date : May 24 2019 at 15:13:31
    ** Copyright 1986-2019 Xilinx, Inc. All Rights Reserved.

localhost:3121
> open_hw_target -xvc_url <HOSTNAME>:2544
INFO: [Labtools 27-2285] Connecting to hw_server url TCP:localhost:3121
INFO: [Labtoolstcl 44-466] Opening hw_target
localhost:3121/xilinx_tcf/Xilinx/<HOSTNAME>:2544
> set_property PROBES.FILE {} [get_hw_devices xc7vx485t_0]
> set_property FULL_PROBES.FILE {} [get_hw_devices xc7vx485t_0]
> set_property PROGRAM.FILE {E:/Projects/xilinx_pcie_2_1_ep_7x.bit}
[get_hw_devices xc7vx485t_0]
> program_hw_devices [get_hw_devices xc7vx485t_0]
INFO: [Labtools 27-3164] End of startup status: HIGH
program_hw_devices: Time (s): cpu = 00:03:12 ; elapsed = 00:03:10 . Memory
(MB): peak = 1882.066 ; gain = 1.477
```

As shown in Figure 16, Vivado™ reports the device as programmed.

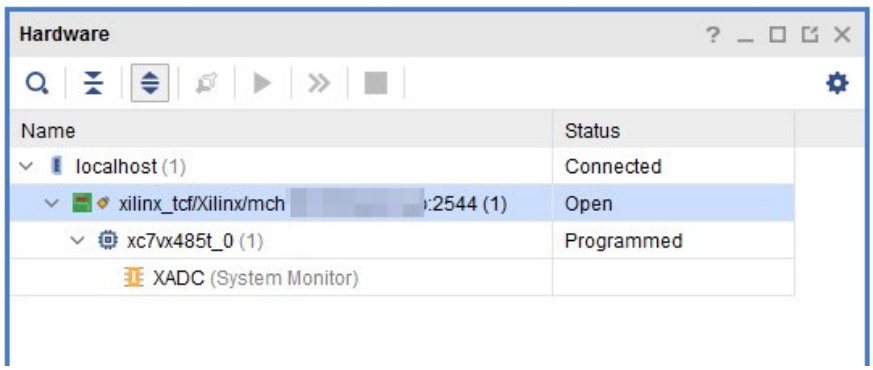


Figure 16: View of the JSM from Xilinx® Vivado™

7.4 HPM.1 Firmware Upgrade

Attention: This feature has not been implemented in DMMC-STAMP-BoB firmware, yet

The PICMG HPM.1 standard (https://www.picmg.org/openstandards/hardware-platform-management/) specifies a method for upgrading the firmware of the MMC and also the firmware images stored in on-board Flash memories. As such, this protocol is very useful, as it allows out-of-band update of FPGA images, even in the cases when the normal in-band communication protocol (e.g. PCI Express or Gigabit Ethernet) is not available.

To demonstrate the update capabilities over HPM.1, there is one Flash component (1Gb, part number MT25QU01GBBB8E12-0SIT) placed on DMMC-STAMP-BoB. This Flash is not connected to any other device; on a real AMC board the Flash would most likely be connected to an FPGA or a similar device.

`ipmitool` together with `hpm check` argument can be used to retrieve the information about the components which can be upgraded on the BoB:

```
$ ipmitool -I lan -H <hostname> -P "" -B 0 -b 7 -T 0x82 -t <fru-id> hpm check

PICMG HPM.1 Upgrade Agent 1.0.9:

-------Target Information-------
Device Id           : 0x0
Device Revision     : 0x80
Product Id          : 0xc0de
Manufacturer Id     : 0x053f (Unknown (0x53F))


--------------------------------------------------------------------------------
|ID  | Name         |                      Versions                           |
|    |              |     Active      |     Backup      |      Deferred      |
--------------------------------------------------------------------------------
|   0|STAMP_SAM4L   |   7.82 00000000 | ---.-- -------- | ---.-- -------- |
|   1|MT25QU01G     |   0.00 EE010000 | ---.-- -------- | ---.-- -------- |
--------------------------------------------------------------------------------
(*) Component requires Payload Cold Reset
```

7.4.1 Firmware Upgrade Procedure

To update the content of the Flash, we must first convert the file into HPM format. `bin2hpm` tool can be used for this conversion, usage of the tool is described in Appendix D. Once we have obtained a `.hpm` file (or `.rle.hpm` file) we can use `ipmitool` with `hpm upgrade` argument to download the binary to on-board Flash.

An example of an upgrade is shown on the listing below.

```
$ ipmitool -H <hostname> -P "" \
       -B 0 -b 7 -T 0x82 -t <fru-id> \
       hpm upgrade fpga_image.rle.hpm

PICMG HPM.1 Upgrade Agent 1.0.8:

Validating firmware image integrity...OK
Performing preparation stage...OK

Performing upgrade stage:

--------------------------------------------------------------------------------
|ID  | Name        |                    Versions                          | %  |
|    |             |      Active     |      Backup     |      File       |    |
|----|-------------|-----------------|-----------------|-----------------|----|
|   1|MT25QU01G    |   0.00 EE010000 | ---.-- -------- |   0.01 00000000 |100%|
|    |Upload Time: 12:56             | Image Size:  554126 bytes              |
--------------------------------------------------------------------------------
(*) Component requires Payload Cold Reset

Firmware upgrade procedure successful
```

During the upgrade, MMC will print some status information on the serial console on USB. Shown in Figure 17 is an example of such an output.

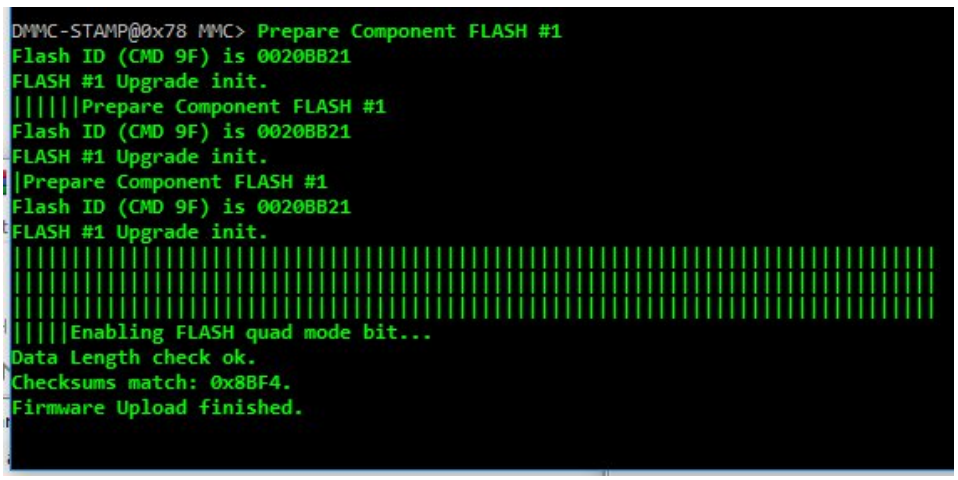
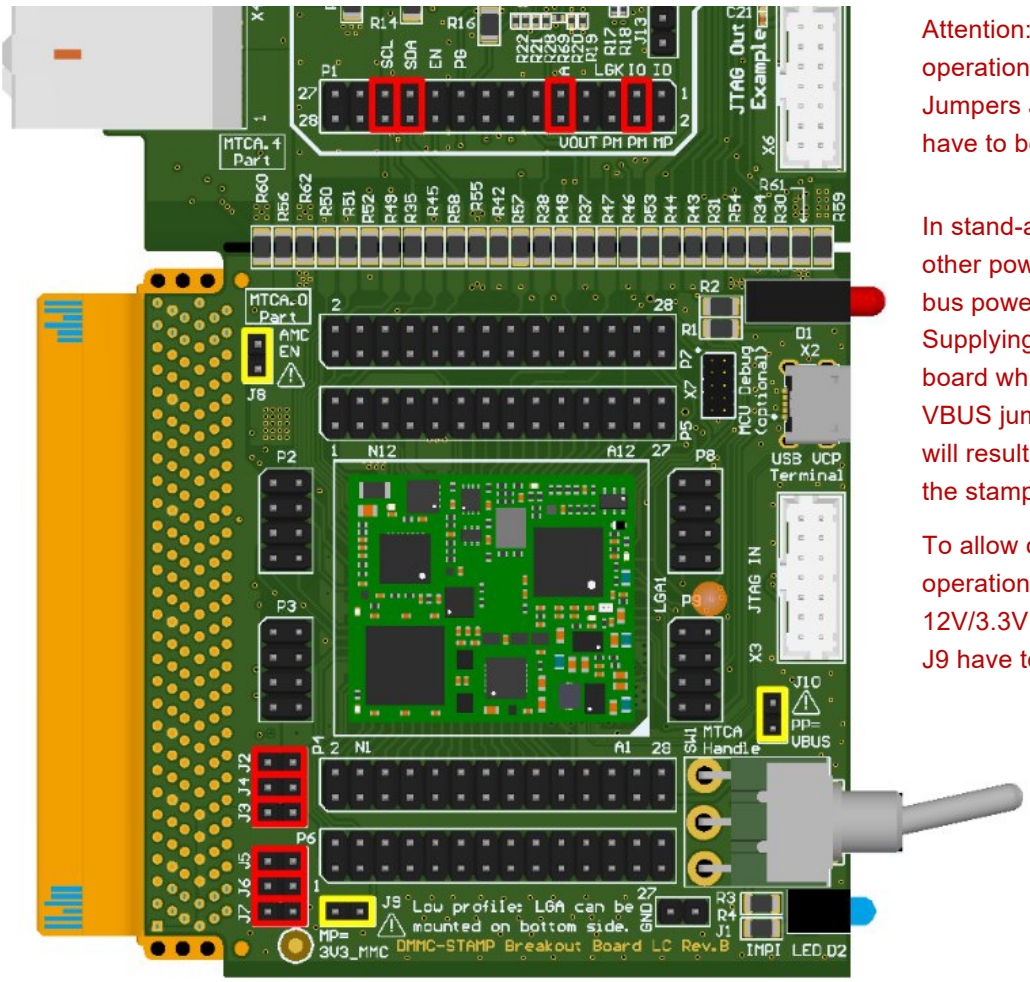


Figure 17: Output from MMC during HPM upgrade

8 Stand-alone operation

The DMMC-STAMP-BoB allows stand-alone (i.e. outside of MicroTCA crate) operation. This feature is useful for testing custom MMC firmware enhancements, as the Payload Power and Management power is derived from the USB cable only. This mode provides bench-top operation with easy access to all stamp pins on the board without the need for an external power supply.

To use the BoB in stand-alone mode, populate the jumpers **J8**, **J9** and **J10** as shown in Figure 18. This will connect the `IPMI_ENABLE_N` to ground (J8); it will connect the 5V VBUS from USB to the 12V AMC payload power rail (J10), so that the input of stamp DC/DC converter is supplied from USB only. In addition, the 3.3V power output from MMC-STAMP (3V3_MMC) will be connected to the Management Power rail (J9) to supply the MMC itself.



Attention: For in-crate operation the stand-alone Jumpers J8, J9 and J10 have to be removed.

In stand-alone mode no other power than USB bus power is allowed. Supplying 12V to the board while the PP-VBUS jumper is installed, will result in a damage of the stamp.

To allow on-bench operation with external 12V/3.3V supply, J10 and J9 have to be removed.

Red Jumpers are mandatory for normal operation
Yellow jumpers must be installed for stand-alone (on-desk) operation only.

Figure 18: DMMC-STAMP-BoB jumper settings for stand-alone operation

9 Current measurement

The MicroTCA standard is strict with current consumption of each individual Advanced Mezzanine Card and power supplies are desired (as discussed in Section “4. Power” of “PICMG® Specification MTCA.0 R1.0”) to monitor the current on both Management Power and Payload Power and take appropriate action when over-current condition is detected.

It is therefore useful to be able to measure the power consumption of the board, both of the management part and payload part. Available on the DMMC-STAMP-BoBs are three jumpers which enable the measurement of currents on individual voltage rails.

Shown on Figure 19 are the three jumpers which can be replaced with ammeter to measure the current.

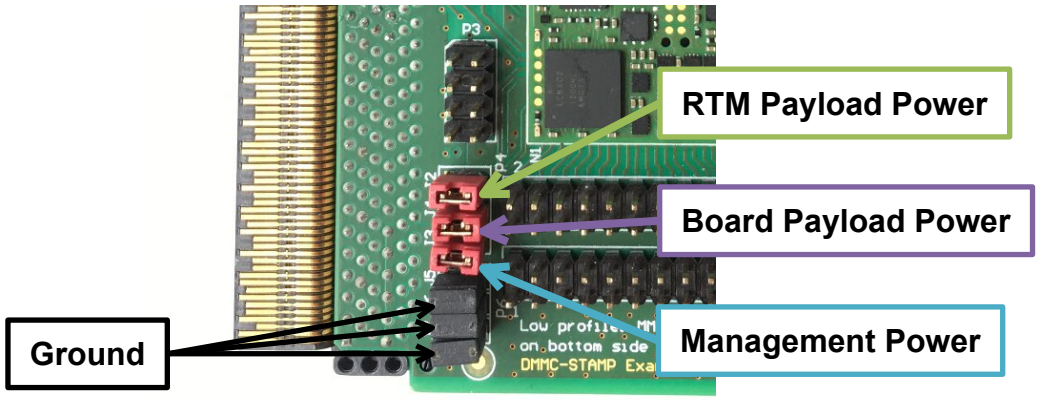


Figure 19: Current measurement jumpers

The power consumption of the board depends on several factors. Shown in the table below is the power consumption when the board is idle and when there is HPM.1 update is progress.

Jumper	Measured rail	Typical consumption - idle [mA]	Typical consumption - HPM.1 update [mA]
J2	RTM Payload Power	1.0	1.0
J3	Payload Power	7.3	7.4
J4	Management Power	14.8	21.0

The following consumption was measured with the PP load current on RTM load board set to 1A:

Jumper	Measured rail	Typical consumption - RTM load [mA]
J2	RTM Payload Power	1006
J3	Payload Power	6.0
J4	Management Power	14.8

10 PMBUS™ Header

The board is equipped with an Analog Devices LTC2972 Power Manager. The control pins can be accessed on the P1 Header. Some Jumpers are installed by default, to connect the power manager to the stamp.

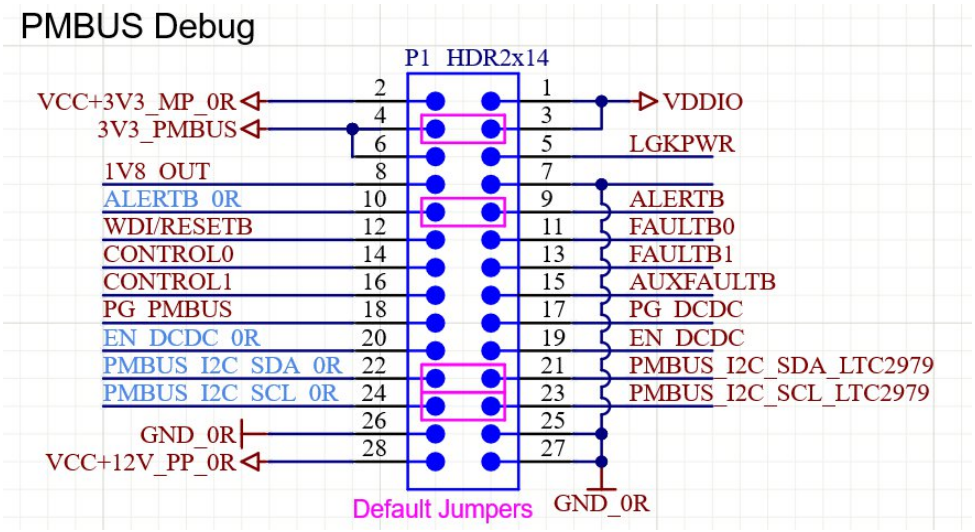


Figure 20: Detail view of PMBUS™ Jumper field.

Appendix A Glossary

AMC	Advanced Mezzanine Card
CPLD	Complex Programmable Logic Device
FMC	FPGA Mezzanine Card
FRU	Field Replaceable Unit
HPM	Hardware Platform Management
IPMB	Intelligent Platform Management Bus
IPMI	Intelligent Platform Management Interface
JTAG	Joint Test Action Group (standard for on-chip instrumentation, such as boundary scan and in-system debugging)
MCU	Micro-Controller
MMC	Module Management Controller
PICMG	PCI Industrial Computer Manufacturer Group
RTM	Rear Transition Module

Appendix B mmcterm usage

mmcterm is a utility that uses a custom “Serial over IPMB” protocol to open an MMC console in a terminal. The tool is available at https://github.com/MicroTCA-Tech-Lab/mmcterm. For details on the usage a part of the README.md file is given below.

```
$ mmcterm [-h] [-v] [-c CHANNEL] [-l] [-d] [-i] mch_addr mmc_addr


DESY MMC Serial over IPMB console

positional arguments:
  mch_addr                IP address or hostname of MCH
  mmc_addr                IPMB-L address of MMC

optional arguments:
  -h, --help              show this help message and exit
  -v, --version           show program's version number and exit
  -c CHANNEL, --channel   CHANNEL
                          console channel
  -l,  --list             list available channels
  -d, --debug             pyipmi debug mode
  -i,  --ipmitool         make pyipmi use ipmitool instead of native rmcp
```

A typical command to open mmcterm can look as in the following:

```
$ mmcterm 192.168.1.205 0x76
Press Ctrl-x to exit
DAMC-FMC1Z7IO@0x76 MMC>help
```

Appendix C ipmitool usage

IPMItool is an open-source project, available at https://github.com/ipmitool/ipmitool/. Description from the project's README page:

> ipmitool is a utility for managing and configuring devices that support the Intelligent Platform Management Interface. IPMI is an open standard for monitoring, logging, recovery, inventory, and control of hardware that is implemented independent of the main CPU, BIOS, and OS.

In MicroTCA environment, ipmitool is used to communicate over LAN to MCH, which is then responsible for routing the IPMI messages on the backplane over IPMB-L. IPMItool is available on most Linux distributions and with Cygwin or *Windows® Subsystem for Linux* on Windows®.

A typical command with IPMItool will usually have these arguments:

```
ipmitool -I lan -H <hostname> -P <password> \
-B 0 -b 7 -T 0x82 -t <fru-id> \
<command>
```

- `-I lan` specifies that the interface uses should be LAN (i.e. Ethernet).
- `-H <hostname>` (replace <hostname> with the hostname or IP address of the MCH) specifies the address to which to send the IPMI packets to.
- `-P` specifies password. If no password is set, use `""`
- `-B 0 -b 7 -T 0x82` specifies the routing of the IPMI packets inside of the MCH.
- `-t <fru-id>` specifies the location of the board in MicroTCA crate – see the paragraph below for a detailed explanation and an example.
- `<command>` can be any command supported by IPMItool. Several commands are already demonstrated in this document; use help to print all available commands.

As described in Chapter "3.2.1 Geographic Address [2..0] (GA[2..0])" of PICMG AMC.0 standard, the IPMB-L address of a Module can be calculated as (70h + Site Number x 2).

Example:

	Physical Address	
IPMB-L address	Site Number	Site Type
0x72	1	AMC (0x07)
0x74	2	AMC (0x07)
…	…	…

Appendix D bin2hpm usage

`bin2hpm` is a utility, available at https://github.com/MicroTCA-Tech-Lab/bin2hpm to convert binary files (e.g. FPGA configuration files) into HPM format. To reduce the download time, Run Length Encoding can be also enabled – this feature is specific to DESY MMC implementation.

The following is syntax for the utility:

```
Syntax: bin2hpm FileName [v=MM.NN / v=MMNN] [c=N] [m=ID] [p=ID] [d=ID] [a=ID]
[/bit]
    c: Component ID (dec). Defaults to 1
    d: Device ID (hex). Defaults to ID 0.
    m: ManufacturerID (hex). Defaults to 0x000000.
    p: Product ID (hex). Defaults to 0x0000.
    v: Version Number (dec.dec). Defaults to 00.00.
    a: Auxilary Version Information (hex). Defaults to 0x0000.
       Use v=1.01 or v=101 (Auto-Conversion)
    /bit: Interpret file as Xilinx bit file.
   /compress: Activate RLE compression (bit file only; requires DESY MMC).
Example: bin2hpm SIS8300L2.bin v=1.0 c=0 m=0x00053F p=0x0002
```

And the following is an example usage of bin2hpm to prepare an FPGA image for HPM upgrade (some lines were removed to fit the output on this page):

```
bin2hpm fpga_image.bit /bit /compress v=0.1 c=1 m=0x053f p=0xc0de
BIN-to-HPM file converter. Version: 2.0.2
Written by Michael Fenner (C) DESY 2014;
with Xilinx BIT file support and
with compression based on VPackBits (C) Michael Dipperstein (LGPL) and
with MD5 algorithm from Alexander Peslyak (public domain)

Reading file                    : fpga_image.bit
File Lengh is                   : 16007020 bytes
Bit File mode. Header check ok.
a-Section detected. Design info : mgt_top;UserID=0XFFFFFFFF;Version=2019.1
b-Section detected. Part name   : xcku040-ffva1156-1-c
c-Section detected. File date   : 2019/07/23
d-Section detected. File time   : 17:08:05
e-Section detected. Image size  : 0x00F43EFC (15631KB)
BIT header sucessfully parsed.
Generating HPM header...
Device ID                       : 0x00
Manufacturer ID                 : 0x00053F
Product ID                      : 0xC0DE

...

Verifying decompression         : Checksum OK / Length OK
Generating MD5 checksum         : 1F98E9A79F8F72F130D94CC3C82A5D22
Output data length is           : 554214 bytes
Overhead (header+MD5) size is   : 103 bytes
Writing file                    : fpga_image.rle.hpm
Bytes actually written          : 554214
All done.
```

Appendix E Serial port on Windows®

To identify the COM port for DMMC-STAMP, connect USB cable to DMMC-STAMP-BoB.

The board uses a CP2105 virtual COM port bridge. If drivers are needed, they can be found at the Silicon Labs web page in the section “CP210x USB to UART Bridge VCP Drivers”. At the time this manual was created, they were located on this address:

https://www.silabs.com/products/development-tools/software/usb-to-uart-bridge-vcp-drivers

The board is visible in the Windows® Device Manager. In the section “Ports (COM & LPT)” there should be two entries which start with “Silicon Labs Dual CP2105 USB to UART Bridge”. The COM port associated with the MMC itself (the other is connected to FPGA) is the first one. Shown in Figure 21 is an example, in this case the MMC is available on COM21.

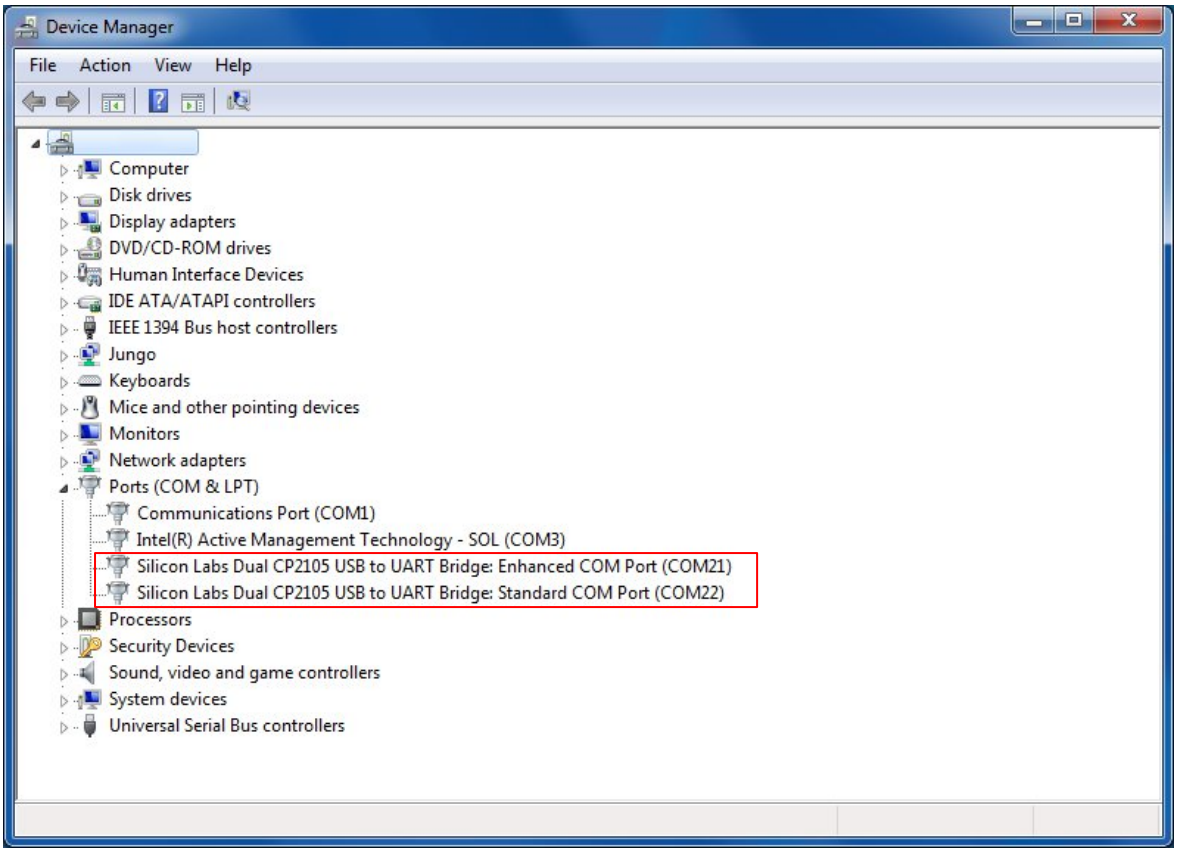


Figure 21: COM ports in Device Manager

Open PuTTY (https://www.putty.org/) or some other terminal client and open Serial connection. Settings to use: 115200 baud, 8 bits, 1 stop bit, no parity, XON/XOFF flow control. An example of serial port configuration is shown in Figure 22.

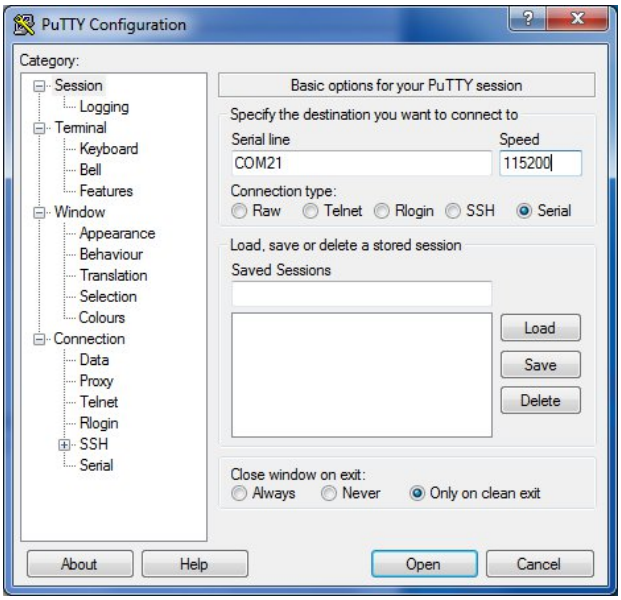


Figure 22: Serial port configuration in PuTTY

The terminal window should open. Press **ENTER** to display the prompt from the MMC Stamp.

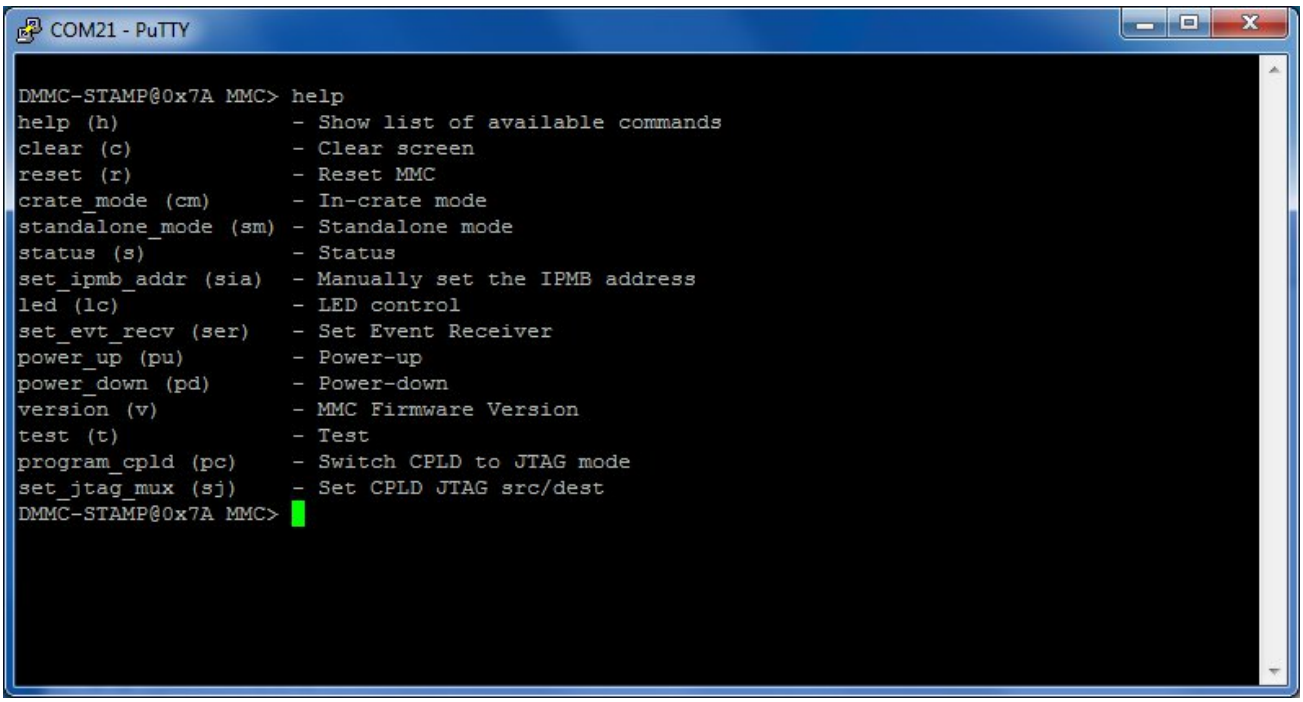


Figure 23: DMMC-STAMP USB communication on Windows®

Appendix F Serial port on GNU/Linux

The device which corresponds to the attached DMMC-STAMP can be found from the **dmesg** output (there are two serial connections; MMC is connected to the first one):

```
$ dmesg | tail
[   98.557843] usb 3-1: new full-speed USB device number 4 using xhci_hcd
[   98.707804] usb 3-1: New USB device found, idVendor=10c4, idProduct=ea70
[   98.707808] usb 3-1: New USB device strings: Mfr=1, Product=2, SerialNumber=5
[   98.707811] usb 3-1: Product: CP2105 Dual USB to UART Bridge Controller
[   98.707813] usb 3-1: Manufacturer: Silicon Labs
[   98.707815] usb 3-1: SerialNumber: 009F7CB3
[   98.708643] cp210x 3-1:1.0: cp210x converter detected
[   98.710526] usb 3-1: cp210x converter now attached to ttyUSB0
[   98.710872] cp210x 3-1:1.1: cp210x converter detected
[   98.712427] usb 3-1: cp210x converter now attached to ttyUSB1
```

Use **minicom -s** to start minicom, select “Serial port setup” and configure the settings as shown on the listing below:

```
    +-----------------------------------------------------------------------+
    | A -    Serial Device      : /dev/ttyUSB0                              |
    | B - Lockfile Location     : /var/lock                                 |
    | C -   Callin Program      :                                           |
    | D -  Callout Program      :                                           |
    | E -    Bps/Par/Bits       : 115200 8N1                                |
    | F - Hardware Flow Control : No                                        |
    | G - Software Flow Control : Yes                                       |
    |                                                                       |
    |    Change which setting?                                              |
    +-----------------------------------------------------------------------+
```

Exit the menu and the Welcome screen will show up. Press **ENTER** to display the prompt from the MMC Stamp:

```
Welcome to minicom 2.7

OPTIONS: I18n
Compiled on Nov 15 2018, 20:18:47.
Port /dev/ttyUSB0, 18:43:12

Press CTRL-A Z for help on special keys


DMMC-STAMP@0x7A MMC>
```

Appendix G Differential Ports

Differential ports on the backplane are connected to the AMC connector on the DMMC-STAMP-BoB, but are not routed to any device. An unconventional use of BoB can be to bring out the differential pairs from the backplane (e.g. PCI Express, Gigabit Ethernet, MLVDS lanes or Point-to-Point links) to an external board. An example is presented here:

NOTE: SOLDERING WIRES TO THE BOARD WILL VOID THE WARRANTY. THE OPERATION SHOULD BE PERFORMED BY A QUALIFIED TECHNICIAN.

Coaxial cables can be soldered directly to the connector – this minimizes the stubs and provides best signal integrity. Please check AMC.0 specification for the connector pinout. Figure 24 shows an example of port 4 (PCI Express, lane 0) connected to coaxial cables.

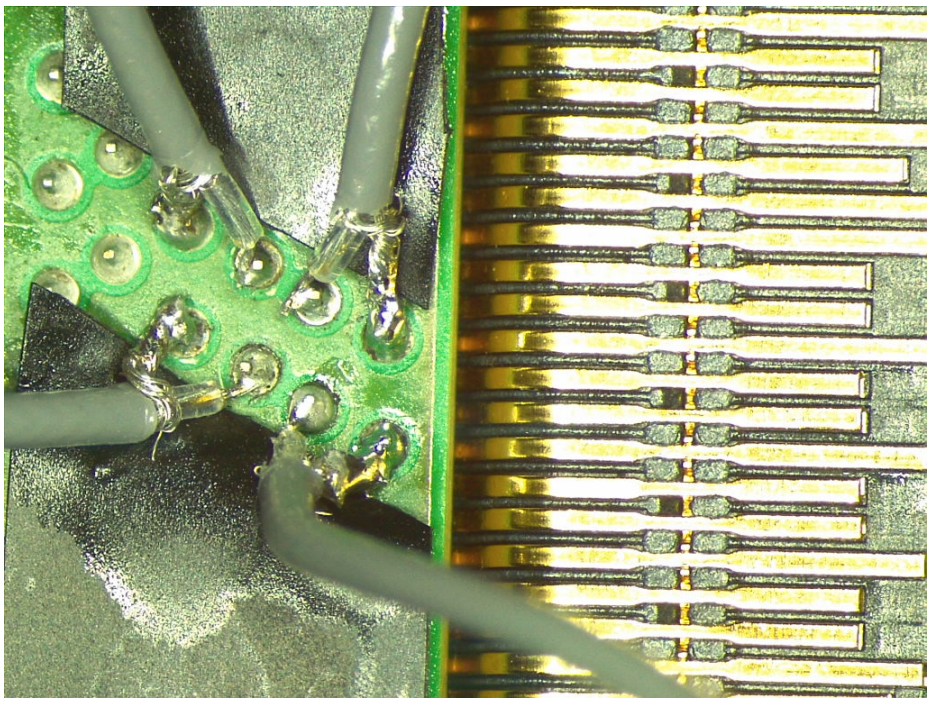


Figure 24: Coaxial cables on AMC port 4

The FRU needs to be changed accordingly – AdvancedMC Point-to-Point record needs to be added for the taped-out interface, see Chapter “4.2 FRU Generator” in DMMC-STAMP User’s Manual.

The cables can then be connected to another board, such as FPGA Evaluation Kit or, as presented in Figure 25, to an external test stand. The test stand connects all ports on AMC connector to SMA connectors, allowing easy access to all backplane connections. With the setup shown Figure 25 it is possible to operate the AMC board outside of the MicroTCA over PCI Express. A total of 6 coaxial cables was needed, 2 for TX, 2 for RX and 2 for FCLKA. The PCI Express connection between the connector on MMC-STAMP and the test stand was working reliably at 2.5 GT/s and 5.0 GT/s but had significant issues at 8.0 GT/s.

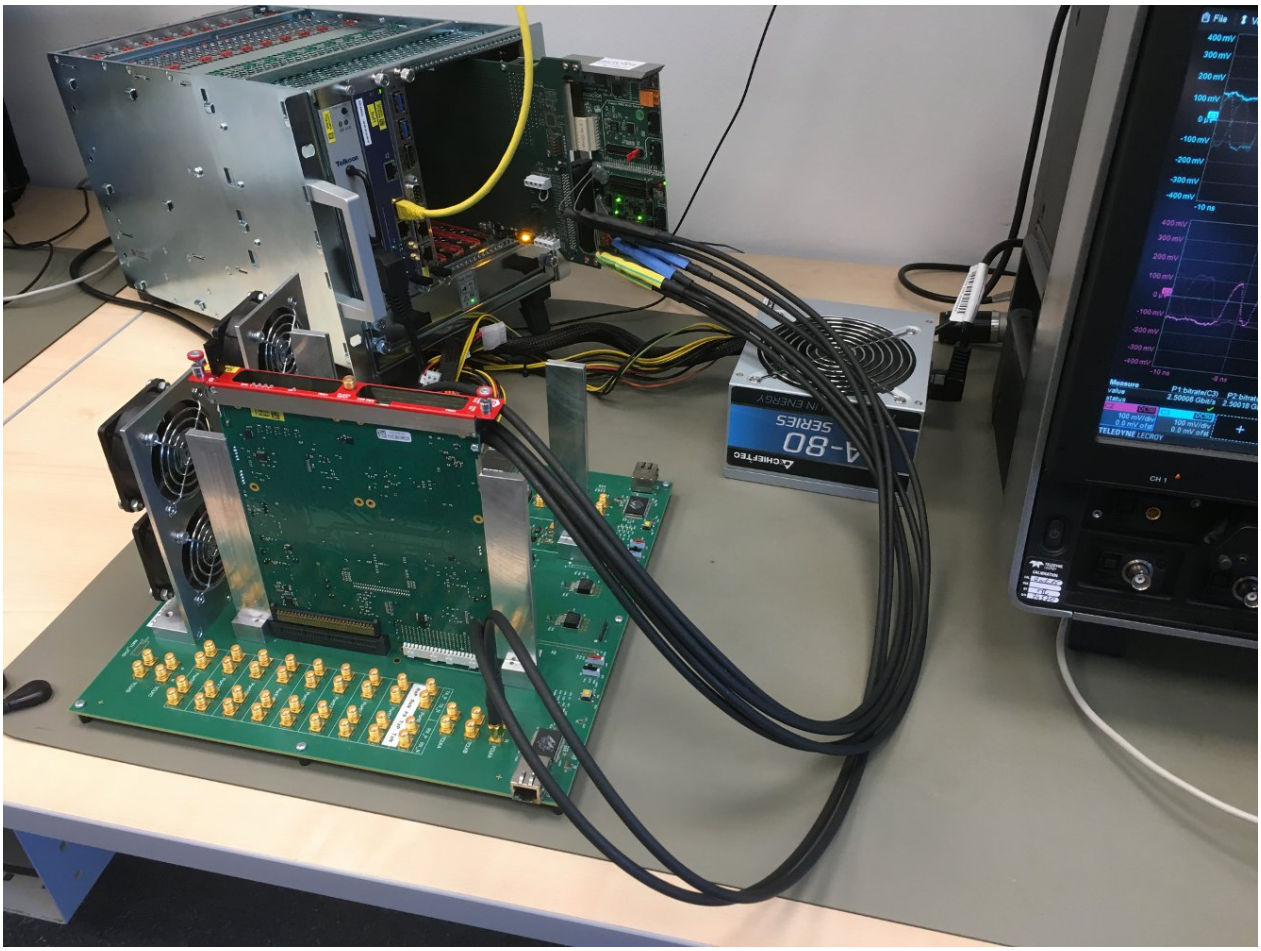


Figure 25: Experimental setup with PCIe tapped out from DMMC-STAMP-BoB

Appendix H Troubleshooting

When there is no output at all (and no blinking LED), the bootloader is not programmed or has been erased by during application programming.

A marching dot "….." on handle insertion (switch down) means that the board is waiting for the power supply to come up (power good signal). If the marching continues, the PMBUS™ manager is not programmed correctly

When CPLD write errors are reported, the CPLD is not programmed correctly