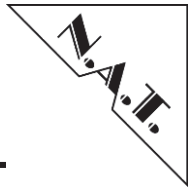
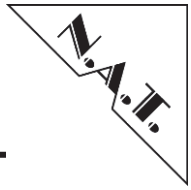


NAT-JSM
JTAG Switch Module
Technical Reference Manual V1.10
HW Revision 1.1

Ehlbeek 15a
30938 Burgwedel
fon 05139-9980-0
fax 05139-9980-49
www.powerbridge.de
info@powerbridge.de



The NAT-JSM has been designed by:

N.A.T. GmbH
Konrad-Zuse-Platz 9
53227 Bonn-Oberkassel

Phone: +49 / 228 / 965 864 – 0
Fax: +49 / 228 / 965 864 – 10

Internet: http://www.nateurope.com

Disclaimer

The following documentation, compiled by N.A.T. GmbH (henceforth called N.A.T.), represents the current status of the product´s development. The documentation is updated on a regular basis. Any changes which might ensue, including those necessitated by updated specifications, are considered in the latest version of this documentation. N.A.T. is under no obligation to notify any person, organization, or institution of such changes or to make these changes public in any other way.

We must caution you, that this publication could include technical inaccuracies or typographical errors.

N.A.T. offers no warranty, either expressed or implied, for the contents of this documentation or for the product described therein, including but not limited to the warranties of merchantability or the fitness of the product for any specific purpose.

In no event will N.A.T. be liable for any loss of data or for errors in data utilization or processing resulting from the use of this product or the documentation. In particular, N.A.T. will not be responsible for any direct or indirect damages (including lost profits, lost savings, delays or interruptions in the flow of business activities, including but not limited to, special, incidental, consequential, or other similar damages) arising out of the use of or inability to use this product or the associated documentation, even if N.A.T. or any authorized N.A.T. representative has been advised of the possibility of such damages.

The use of registered names, trademarks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations (patent laws, trade mark laws, etc.) and therefore free for general use. In no case does N.A.T. guarantee that the information given in this documentation is free of such third-party rights.

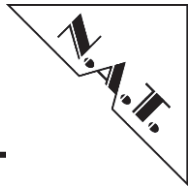
Neither this documentation nor any part thereof may be copied, translated, or reduced to any electronic medium or machine form without the prior written consent from N.A.T. GmbH.

This product (and the associated documentation) is governed by the N.A.T. General Conditions and Terms of Delivery and Payment.

Note:

The release of the Hardware Manual is related to a certain HW board revision given in the document title. For HW revisions earlier than the one given in the document title please contact N.A.T. for the corresponding older Hardware Manual release.

Table of Contents



List of Tables

List of Figures

Conventions

If not otherwise specified, addresses and memory maps are written in hexadecimal notation, identified by *0x*.
The following table gives a list of the abbreviations used in this document.

Table 1: List of used abbreviations

Abbreviation	Description
AMC	Advanced Mezzanine Card
ATCA	Advanced Telecommunications Computing Architecture
FPGA	Field Programmable Gate Array
I/O	Input/Output
JSM	JTAG Switch Module
JTAG	Joint Test Action Group
LED	Light Emitting Diode
µTCA/MTCA	Micro Telecommunications Computing Architecture
MCH	µTCA/MTCA Carrier Hub
SMP	Shared Management Power
TAP	Test Access Port
USB	Universal Serial Bus
XVC	Xilinx Virtual Cable

1 Introduction

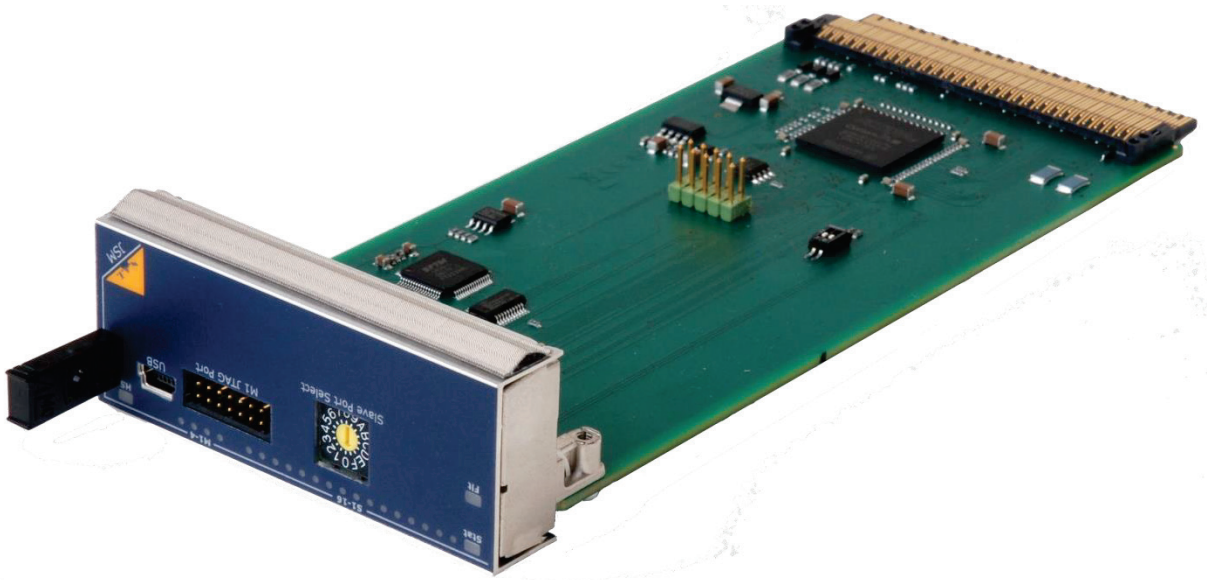
The **NAT-JSM** is an MTCA compliant JTAG switch module with a flexible design that makes it compatible with most of todays MTCA chassis providing a JSM slot.
The module detects whether it is inserted into an AMC, MCH or JSM slot; only if a JSM slot is detected the output drivers are turned on. The only prerequisite for misinsertion protection is that the pinout of the JSM-Connector follows the basic rules of the AMC-Connector.

The **NAT-JSM** can be adapted to nearly any existing JSM system connector by configuration of the onboard FPGA as long as it is based on the AMC-Connector layout.

By default the **NAT-JSM** automatically arbitrates the JTAG master port, the slave port is selected by the TAP controller. The automatic configuration can be overruled at any time by manual configuration through front panel elements.

The following figure shows a photo of the **NAT-JSM**.

Figure 1: NAT-JSM



2 Overview

2.1 Major Features

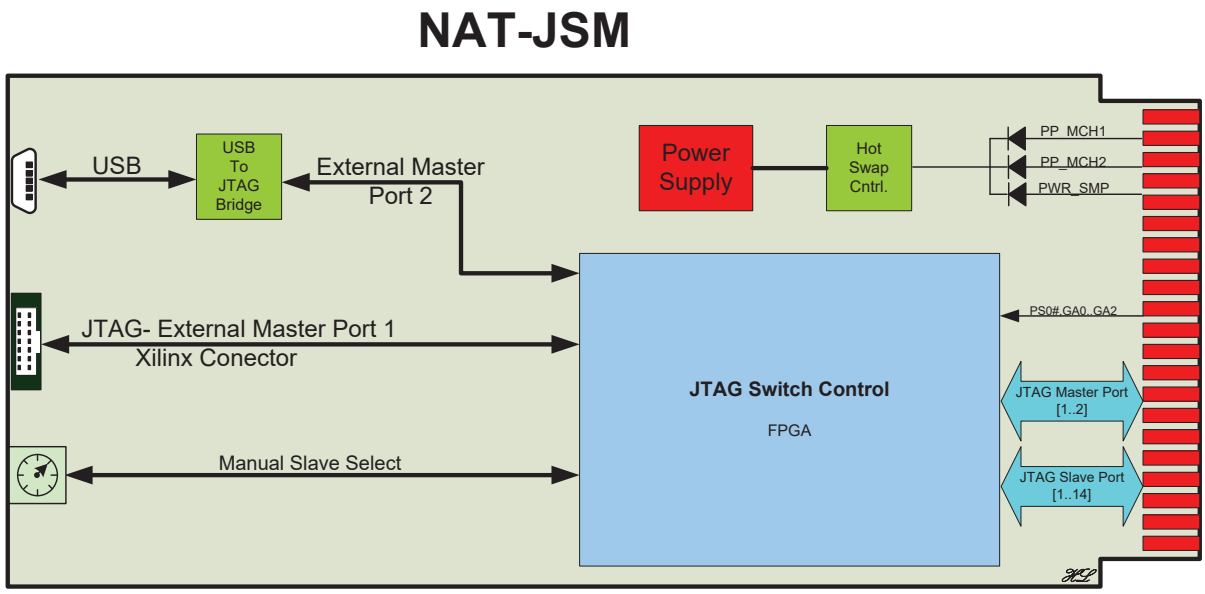
- JTAG download from MCH via WEB Interface
- JTAG programming connector at front panel
- Automatic arbitration between JTAG Masters
- Target selection through JTAG information
- Overrule of automatic operation and dedicated selection of JTAG target by front panel elements
- Multiple JSM Pinout configurations via FPGA
- Power supply through MCH power channels or power module SMP power
- Minimal power consumption

For detailed description, see the following chapter.

2.2 Block Diagram

The following figure shows a block diagram of the **NAT-JSM** board.

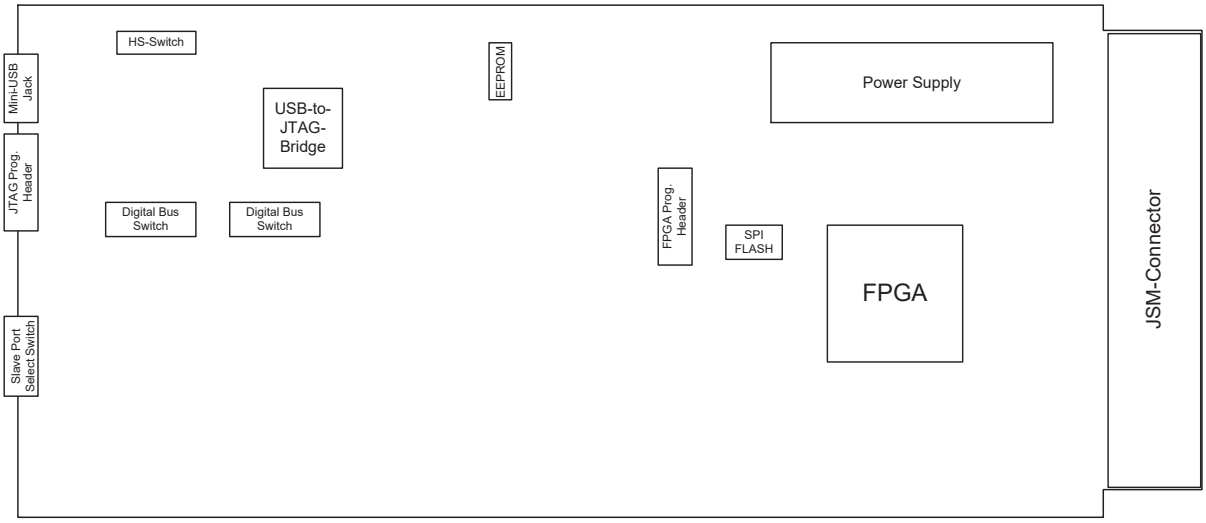
Figure 2: NAT-JSM – Block Diagram – Overview



2.3 Location Diagram

The position of important components is shown in the following location overview. Depending on the board type, it might be that the board does not include all components named in the location diagram.

Figure 3: NAT-JSM – Location Diagram – Overview



3 Programming Interfaces

The **NAT-JSM** per default performs an automatic arbitration for the various JTAG master ports. This means the module switches its master ports input to the last active master port.

3.1 Pin-Header Programming Interface

The **NAT-JSM** provides a XILINX compatible 14 pin programming header at the faceplate.
With a standard XILINX programming adapter the resources in a MTCA system can be programmed or debugged.

The interface has been tested successfully with

- Xilinx – Platform Cable USB II

The programming target is selected by the rotary switch.

3.2 USB Programming Interface

The standard USB connector at the faceplate serves as programming connector for common programming adapters from various vendors. To make usage of this programming interface the software driver of the respective tool should support programming interfaces based on the FTDI FT2232 USB-to-JTAG chip.

The interface has been tested successfully with

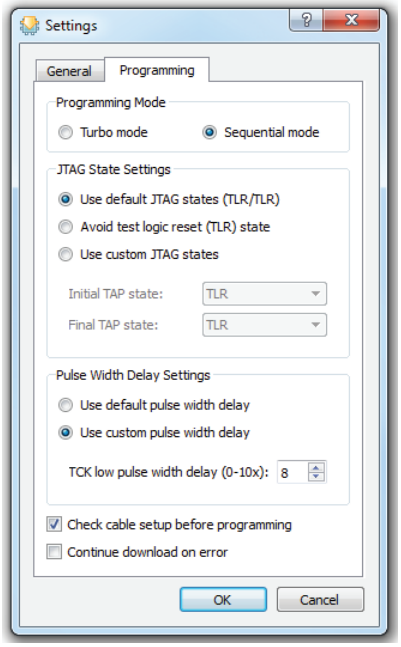
- Lattice – Diamond Programmer 3.0

The programming target is selected by the rotary switch or it can be selected via the web-interface of the **NAT-MCH**.

In the Lattice Diamond Programmer software, the "Detect Cable" function can be used to get access to the USB-to-JTAG bridge device.

Depending on the trace lengths and the used MTCA chassis, it might be necessary to reduce the standard JTAG clock frequency from about 15MHz to values below 8MHz via "Edit" → "Settings" → "Programming". After activating the radio-button "Use custom pulse width delay", a reduction of the TCK frequency can be adjusted.

Figure 4: Lattice Diamond Settings Dialog



3.3 Programming via MCH

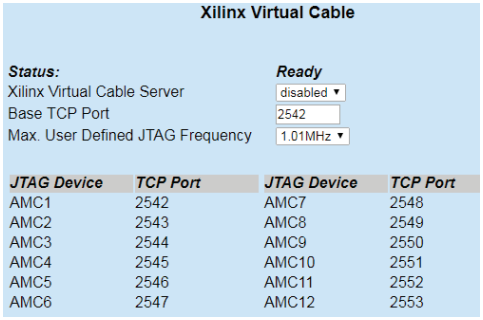
Since the **NAT-MCH** is connected to the **NAT-JSM**, it provides TCP/IP-based communication interfaces for the **NAT-JSM** and allows JTAG access via IP network to the programming targets (described in the *3.4 Target Selection*).

NOTE: Whenever the **NAT-JSM** is using the **NAT-MCH** programming interface, the programming header on the front panel must be disconnected!

3.3.1 Programming via Xilinx Virtual Cable Protocol

Xilinx Virtual Cable (XVC) is a TCP/IP-based protocol, which acts like a JTAG cable and provides a means to access and debug the FPGA or SoC design without using a physical cable[1].

Figure 5: Web interface of Xilinx Virtual Cable Protocol



JTAG Device	TCP Port	JTAG Device	TCP Port
AMC1	2542	AMC7	2548
AMC2	2543	AMC8	2549
AMC3	2544	AMC9	2550
AMC4	2545	AMC10	2551
AMC5	2546	AMC11	2552
AMC6	2547	AMC12	2553

[1] www.xilinx.com/products/intellectual-property/xvc.html

This server side application running on the **NAT-MCH** supports the Xilinx Virtual Cable (XVC) Protocol. The **NAT-MCH** offers 12 TCP ports for 12 programming targets.

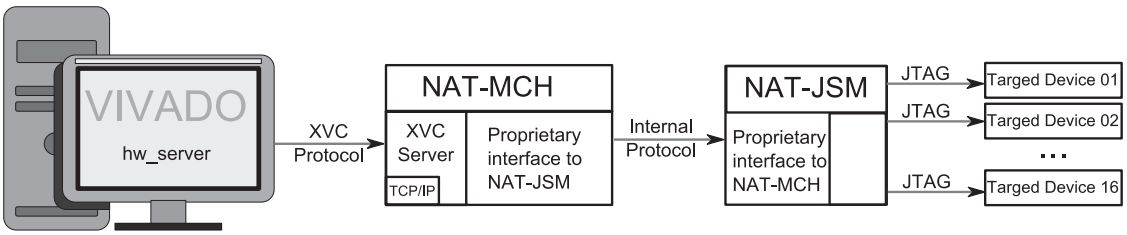
The first TCP port named *Base Port* is tuneable and can be set/saved by user. The other eleven ports are automatically selected by incrementing the Base Port number. Therefore, each programming target becomes dedicated TCP port to be programmed by client software. In the JSM-menu of the **NAT-MCH's** web interface, the mapping between TCP port and programming target is visible.

The *"Max. User Defined JTAG Frequency"* defines the frequency's upper limit for JTAG access to all programming targets.

3.3.1.1 Using Xilinx Vivado Software

The Vivado® design tools include the support of XVC protocol, which allows communicating JTAG commands over IP to an embedded system, so that a target Xilinx FPGA can be programmed and/or debugged. Therefore, the **NAT-MCH** parses the IP packets with a TCP/IP connection and converts the packets into JTAG commands. After the packets are processed, the **NAT-MCH** communicates with the **NAT-JSM** over internal protocol. The **NAT-JSM** switches the connection to the target device and provides logical connection between XVC server and target FPGA. Figure 6: illustrates a high-level block diagram of a typical XVC topology.

Figure 6: Block Diagram of XVC Topology for Programming via MCH



To initiate a connection, open Vivado Hardware Manager. Then start a Hardware Server session with following command in the Tcl Console:

```
>> connect_hw_server
```

After that, open hardware target with following command in the Tcl Console:

```
>> open_hw_target -xvc_url <IP_Address>:<TCP_Port>
```

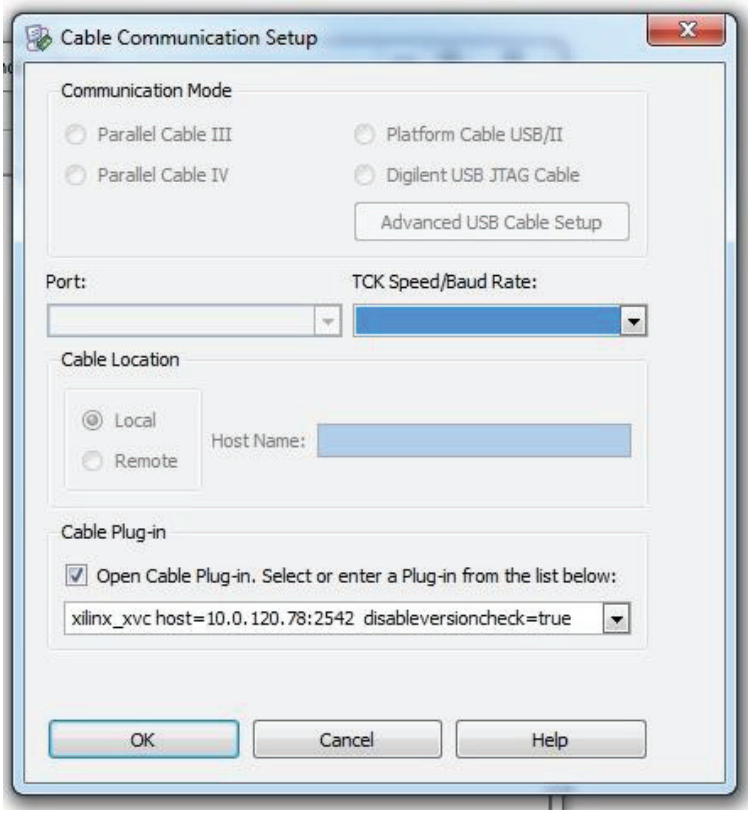
<IP_Address> contains the IP-address of the **NAT-MCH** and *<TCP Port>* describes the TCP port assigned to particular target device [3].

3.3.1.2 Using Xilinx iMPACT Software

As an alternative to the Xilinx Vivado® design tools (these support only 7-series and newer devices), the older Xilinx iMPACT Software is also capable of using the XVC protocol. Therefore, select "Output" → "Cable Setup" → "Cable Communication Setup" and check "Cable Plug-in". Enter the following text in the field below:

```
Xilinx_xvc host=<IP_Address>:<TCP_Port> disableversioncheck=true
```

Figure 7: iMPACT Cable Communication Setup Dialog



3.3.2 Programming SVF file via Web Interface

The Serial Vector Format (SVF) offers another way to access the programming targets via MCH JTAG interface.

SVF is a file format that contains boundary scan vectors to be sent to an electronic circuit using a JTAG interface. The SVF file is defined as an ASCII file that consists of a set of SVF statements. SVF is designed to encourage reuse of serial vectors throughout the life cycle of the product, from its inception in the design phase to its employment in the field service phase, and all phases in between. Life-cycle portability places restrictions on the design and capabilities of SVF.

The Serial Vector Format can be used to program an AMC FPGA via the **NAT-MCH**. However the LIB(X)SVF[2] library is available for the **NAT-MCH**, the SVF file has not to be upload directly, so the SVF Prepare Tool designed by N.A.T. GmbH has to be used first.

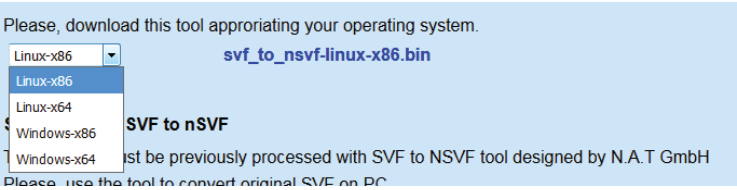
The access to programming targets demands five steps:

- Download the "SVF to NSVF"-Tool
- Generate nSVF-File
- Select AMC
- Select "*Maximal User Defined JTAG Frequency*"
- Upload nSVF-File via **NAT-MCH**

3.3.2.1 Step 1: Download "SVF to NSVF"-Tool

Use the dropdown menu to get the "SVF to NSVF"-Tool, which is appropriate to the used operation system.

Figure 8: Download "SVF to NSVF"-Tool



3.3.2.2 Step 2: Generate nSVF-File

To optimize file parsing on **NAT-MCH**, the original SVF file should be processed by SVF Prepare Tool designed by N.A.T. GmbH.

SVF to NSVF generates a new file with the same filename as source SVF, but with *.nsvf-extension. The new nSVF-file contains proprietary data format designed especially for the **NAT-MCH**.

---

[2] Lib(X)SVF - A library for implementing SVF and XSVF JTAG players
Copyright (C) 2009 RIEGL Research ForschungsGmbH
Copyright (C) 2009 Clifford Wolf <clifford@clifford.at>
Lib(X)SVF is free software licensed under the ISC license

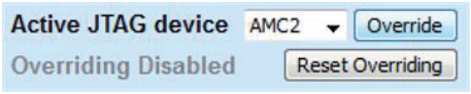
NOTE: THIS STEP IS MANDATORY!

Usage:
./svf_to_nsvf <svf source file>.

3.3.2.3 Step 3: Select AMC

Use the rotary switch on the **NAT-JSM's** faceplate to select the AMC to be programmed. If the system is operated remotely, the rotary switch state can be overridden via the web-interface of the **NAT-MCH**.

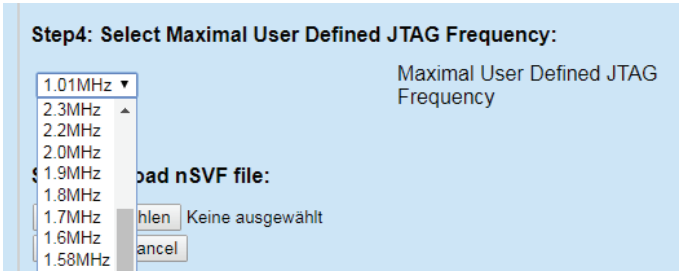
Figure 9: Select AMC



3.3.2.4 Step 4: Select "Maximal User Defined JTAG Frequency"

By choosing the *"Max. User Defined JTAG Frequency"* in the JSM-Menu, the frequency's upper limit, which can be requested by the SVF-File, is defined.

Figure 10: Select "Maximal User Defined JTAG Frequency"

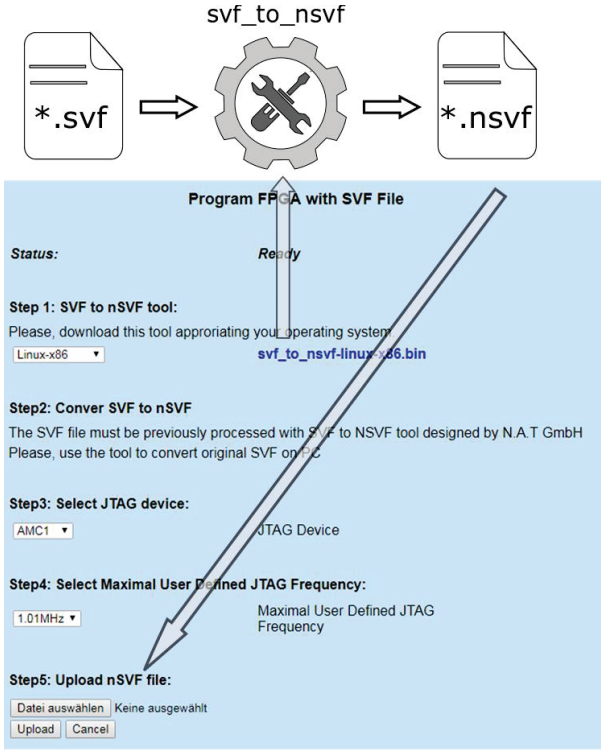


3.3.2.5 Step 5: Upload nSVF-File via NAT-MCH

The new nSVF-File is ready to be uploaded via the **NAT-MCH's** web-interface; the upload interface is available via the JSM-Menu.

NOTE: The web form allows to upload files with *.nsvf extension only.

Figure 11: Upload nSVF-File via NAT-MCH



3.3.3 Programming SVF file via URL

The SVF programming interface is designed to program a particular JTAG device via sole URL. It should contain action name "`/goform/ctrl_svf_proc`", programming target parameter "`XsvfReqTarget`" and Maximal User Defined JTAG Frequency parameter "`XsvfReqFreq`" as well as username and password, which are required for HTTP authentication. Of course, the URL interface serves only nSVF-Files generated by "`SVF to NSVF`"-Tool (see chapters 3.3.2.1and 3.3.2.2).

Table 2: Parameter "`XsvfReqTarget`" (Programming Target)

JTAG Device	XsvfReqTarget Value
AMC_SLOT_1	0
AMC_SLOT_2	1
AMC_SLOT_3	2
AMC_SLOT_4	3
AMC_SLOT_5	4
AMC_SLOT_6	5
AMC_SLOT_7	6
AMC_SLOT_8	7
AMC_SLOT_9	8
AMC_SLOT_10	9
AMC_SLOT_11	10
AMC_SLOT_12	11

Table 3: Parameter "`XsvfReqFreq`" (Max User Defined JTAG Frequency)

Frequency	XsvfReqFreq Value
FREQ_16_6_MHz	0
FREQ_11_1_MHz	1
FREQ_8_3_MHz	2
FREQ_6_6_MHz	3
FREQ_5_5_MHz	4
FREQ_4_7_MHz	5
FREQ_4_1_MHz	6
FREQ_3_7_MHz	7
FREQ_3_3_MHz	8
FREQ_3_0_MHz	9
FREQ_2_7_MHz	10
FREQ_2_5_MHz	11
FREQ_2_3_MHz	12
FREQ_2_2_MHz	13
FREQ_2_0_MHz	14
FREQ_1_9_MHz	15
FREQ_1_8_MHz	16
FREQ_1_7_MHz	17
FREQ_1_6M_MHz	18
FREQ_1_58_MHz	19
FREQ_1_51_MHz	20
FREQ_1_44_MHz	21

FREQ_1_38_MHz	22
FREQ_1_33_MHz	23
FREQ_1_28_MHz	24
FREQ_1_23_MHz	25
FREQ_1_19_MHz	26
FREQ_1_14_MHz	27
FREQ_1_11_MHz	28
FREQ_1_07_MHz	29
FREQ_1_04_MHz	30
FREQ_1_01_MHz	31

The URL interface allows automatic JSM-Programming via any command line tool. The following example demonstrates how the ***curl*** command-line tool programs a JTAG device on AMC_SLOT_5 with "`my_fpga_image.nsvf`" file. It uses username "`root`" and password "`nat`" for HTTP authentication.

```
curl -f -s\
  -H "Content-Type: multipart/form-data" \
  -F "XsvfReqTarget=4" \
  -F "XsvfReqFreq=6" \
  -F "filename=@/home/myuser/curl_script/my_fpga_image.nsvf" \
  -X POST  -u "root":"nat" \
  "http://192.168.1.193/goform/ctrl_svf_proc" >/dev/null\
        && echo "Programming has successfully completed." \
        || echo "Programming has failed."
```

3.4 Target Selection

When programming via one of the front interfaces, the programming target (JTAG Slave Port) can be selected by the rotary switch at the faceplate:

Table 4: Rotary Switch / Slave-LED – Channel-Assignment

Rotary Switch Value	Selected Target (Slave Port)	Slave LED
0	AMC 1	1
1	AMC 2	2
2	AMC 3	3
3	AMC 4	4
4	AMC 5	5
5	AMC 6	6
6	AMC 7	7
7	AMC 8	8
8	AMC 9	9
9	AMC 10	10
A	AMC 11	11
B	AMC 12	12
C	reserved	13
D	Reserved	14
E	Reserved	15
F	reserved	16

Alternatively, the target can be selected via the web-interface, which overrules the rotary setting.

Important:

The JTAG Slave Port selection is designed according to a standard JTAG routing. Per MTCA standard, information about custom JTAG routing of the backplane is not provided by the backplane EEPROM, and thus not apparent to the MCH via E-Keying.
So if the backplane the **NAT-JSM** is plugged into supports a different JTAG routing, the JTAG Slave Port selection has to be adjusted manually.

3.5 Master Selection

The JTAG Master port is automatically selected upon clock activity on one of the four master ports:

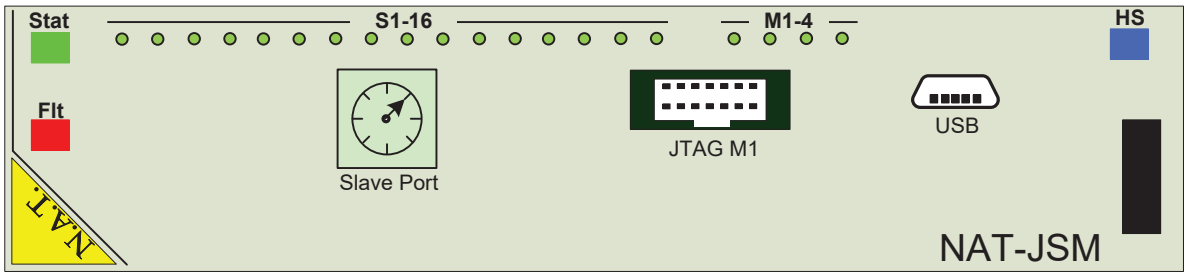
Table 5: Master-LED – Channel-Assignment

Master LED	Active Master
1	Face Plate Header M1
2	Onboard FTDI USB-JTAG Chip
3	MCH 1
4	MCH 2

3.6 Front Panel and LED

The **NAT-JSM** module is equipped with 16 green LEDs (S1-S16) indicating the chosen JTAG slave port; another four green LEDs (M1-M4) show the selected JTAG master port.

Figure 12: NAT-JSM – Front Panel View



Additionally, the module contains the standard AMC-LEDs consisting of a red Fault Indication LED, a General Purpose status LED and the Hot-Swap handle with the corresponding blue LED.

The Fault Indication LED turns to "On" if the temperature sensor registers a temperature value falling below or exceeding a threshold level. If the temperature returns to normal value, the LED is switched to "Off" again.

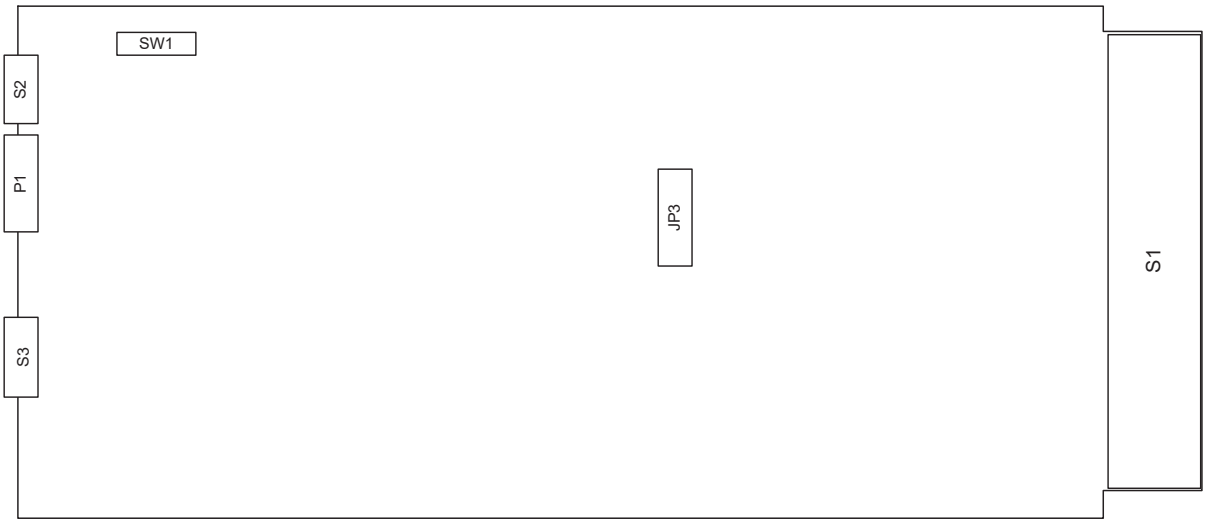
Although optically appearing as one LED, the General Purpose LED physically consists of two LEDs (green and orange) sharing the same hole in the Front Plate.

The LEDs have the following functionality:

Green LED on	: idle/ready state
Green blinking	: transfer in progress
Yellow LED on	: JTAG failure

3.7 Connectors and Switches

Figure 13: NAT-JSM – Connector and Switch Location – Overview



Please refer to the following tables to look up the connector and switch pin assignment of the **NAT-JSM**.

3.7.1 S1: JSM Connector

Table 6: S1: JSM Connector – Pin-Assignment

Pin #	JSM-Signal	JSM-Signal	Pin #
1	GND	GND	170
2	PP_MCH1	TMREQ1	169
3	#PS1	TMREQ2	168
4	+3.3V	FPGA_GP	167
5	GA0	FPGA_GP	166
6	IO_2	FPGA_GP	165
7	GND	GND	164
8	IO_3	PMTRST2	163
9	PP_MCH1	PMTRST1	162
10	GND	GND	161
11	TCK1	STRST12	160
12	TMS1	STRST11	159
13	GND	GND	158
14	TDI1	STRST10	157
15	TDO1	STRST9	156
16	GND	GND	155
17	GA1	STRST8	154
18	NC	STRST7	153
19	GND	GND	152
20	STCK1	STRSt6	151
21	STMS1	STRST5	150
22	GND	GND	149
23	STDI1	STRST4	148
24	STDO1	STRST3	147
25	GND	GND	146
26	GA2	STRST2	145
27	PWR_SMP	STRST1	144
28	GND	GND	143
29	STCK2	TRST2	142
30	STMS2	TRST1	141
31	GND	GND	140
32	STDI2	PMTDI2	139
33	STDO2	PMTDO2	138
34	GND	GND	137
35	STCK3	PMTMS2	136
36	STMS3	PMTCK2	135
37	GND	GND	134
38	STDI3	PMTCK4	133
39	STDO3	PMTMS4	132
40	GND	GND	131
41	IO_4	PMTDI4	130
42	NC	PMTDO4	129
43	GND	GND	128
44	STCK4	PMTCK3	127

Pin #	JSM-Signal	JSM-Signal	Pin #
45	STMS4	PMTMS3	126
46	GND	GND	125
47	STDI4	PMTDO3	124
48	STDO4	PMTDI3	123
49	GND	GND	122
50	STCK5	TCK2	121
51	STMS5	TMS2	120
52	GND	GND	119
53	STDI5	TDI2	118
54	STDO5	TDO2	117
55	GND	GND	116
56	IO_5	STCK12	115
57	NC	STMS12	114
58	GND	GND	113
59	STCK6	STDI12	112
60	STMS6	STDO12	111
61	GND	GND	110
62	STDI6	STCK11	109
63	STDO6	STMS11	108
64	GND	GND	107
65	STCK7	STDI11	106
66	STMS7	STDO11	105
67	GND	GND	104
68	STDI7	STCK10	103
69	STDO7	STMS10	102
70	GND	GND	101
71	IO_6	STDI10	100
72	PP_MCH2	STDO10	99
73	GND	GND	98
74	PMTCK1	STCK9	97
75	PMTMS1	STMS9	96
76	GND	GND	95
77	PMTDI1	STDI9	94
78	PMTDO1	STDO9	93
79	GND	GND	92
80	PMTRST3	STCK8	91
81	PMTRST4	STMS8	90
82	GND	GND	89
83	PS#0	STDI8	88
84	NC	STDO8	87
85	GND	GND	86

Note: Pin assignment of the JSM connector is determined by the FPGA. The table above shows the assignment as given by the standard FPGA image.

3.7.2 S2: Mini-USB Connector

The Mini-USB-Connector S2 provides a direct connection to the on-board USB-to-JTAG-Bridge.

Table 7: S2: Mini-USB Connector – Pin-Assignment

Pin#	Signal	Signal	Pin No.
1	VCC	D-	2
3	D+	nc	4
5	GND	SGND	6
7	SGND	SGND	8

3.7.3 JP3: FPGA Programming Header

Pin Header JP3 offers a FPGA programming interface via Altera Active-Serial-Header.

Table 8: JP3: FPGA Programming Header – Pin-Assignment

Pin#	Signal	Signal	Pin No.
1	DCLK_EP	GND	2
3	CONF_DONE	+3.3V	4
5	nCONFIG	nCE	6
7	DATA0_EP	nCS0	8
9	ASDI	GND	10

3.7.4 P1: JTAG Programming Header

The JTAG Programming Header P1 is located at the faceplate. It is the commonly used JTAG programming interface using a 7x2 pin header with 2mm pitch.

Table 9: P1: JTAG Programming Header – Pin-Assignment

Pin#	Signal	Signal	Pin No.
1	GND	+3.3V	2
3	GND	FR_TMS	4
5	GND	FR_TCK	6
7	GND	FR_TDO	8
9	GND	FR_TDI	10
11	GND	nc	12
13	GND	FR_SRST	14

3.7.5 SW1: Hot Swap Switch

Switch SW1 is used to support hot swapping of the module.

4 Board Specification

Table 10: NAT-JSM Features – Overview

FPGA	Altera Cyclone3
JSM-Module	Single Full Size
Front-I/O	Mini-USB for USB-to-JTAG-Bridge JTAG-Programming Header
Power Supply	12V / 0.2A
Environmental Conditions	Operating: 0-60°C Storage: -40°C-85°C Humidity: 10%-90% non-condensing

5 Installation

5.1 Safety Note

To ensure proper functioning of the **NAT-JSM** during its usual lifetime take the following precautions before handling the board:

CAUTION
Electrostatic discharge and incorrect board installation and uninstallation can damage circuits or shorten their lifetime!

- Before installing or uninstalling the **NAT-JSM** read this installation section
- Before installing or uninstalling the **NAT-JSM**, read the Installation Guide and the User's Manual of the carrier board used, or of the µTCA-System the board will be plugged into.
- Before installing or uninstalling the **NAT-JSM** on a carrier board or both in a rack:
  - Check all installed boards and modules for steps that you have to take before turning on or off the power
  - Take those steps
  - Finally turn on or off the power if necessary
  - Make sure the part to be installed / removed is hot swap capable, if you do not switch off the power.
- Before touching integrated circuits, ensure to take all required precautions for handling electrostatic devices.
- Ensure that the **NAT-JSM** is connected to the carrier board or to the µTCA backplane with the connector completely inserted.
- When operating the board in areas of strong electromagnetic radiation ensure that the module
  - is bolted the front panel or rack
  - and shielded by closed housing

5.2 Installation Prerequisites and Requirements

IMPORTANT
Before powering up check this section for installation prerequisites and requirements!

5.2.1 Requirements

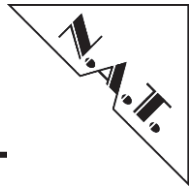
The installation requires only:

- a µTCA backplane with matching JSM pinout for connecting the **NAT-JSM**
- power supply
- cooling devices

5.2.2 Power supply

The power supply for the **NAT-JSM** must meet the following specifications:

- required for the module: +12V / 0.2A max.



5.3 Statement on Environmental Protection

5.3.1 Compliance to RoHS Directive

Directive 2011/65/EU of the European Parliament and of the Council of 8 June 2011 on the "Restriction of the use of certain Hazardous Substances in Electrical and Electronic Equipment" (RoHS) predicts that all electrical and electronic equipment being put on the European market after June 30th, 2006 must contain lead, mercury, hexavalent chromium, polybrominated biphenyls (PBB) and polybrominated diphenyl ethers (PBDE) and cadmium in maximum concentration values of 0.1% respective 0.01% by weight in homogenous materials only.

As these hazardous substances are currently used with semiconductors, plastics (i.e. semiconductor packages, connectors) and soldering tin any hardware product is affected by the RoHS directive if it does not belong to one of the groups of products exempted from the RoHS directive.

Although many of hardware products of N.A.T. are exempted from the RoHS directive it is a declared policy of N.A.T. to provide all products fully compliant to the RoHS directive as soon as possible. For this purpose since January 31st, 2005 N.A.T. is requesting RoHS compliant deliveries from its suppliers. Special attention and care has been paid to the production cycle, so that wherever and whenever possible RoHS components are used with N.A.T. hardware products already.

5.3.2 Compliance to WEEE Directive

Directive 2002/95/EC of the European Commission on "Waste Electrical and Electronic Equipment" (WEEE) predicts that every manufacturer of electrical and electronical equipment which is put on the European market has to contribute to the reuse, recycling and other forms of recovery of such waste so as to reduce disposal. Moreover this directive refers to the Directive 2002/95/EC of the European Commission on the "Restriction of the use of certain Hazardous Substances in Electrical and Electronic Equipment" (RoHS).

Having its main focus on private persons and households using such electrical and electronic equipment the directive also affects business-to-business relationships. The directive is quite restrictive on how such waste of private persons and households has to be handled by the supplier/manufacturer; however, it allows a greater flexibility in business-to-business relationships. This pays tribute to the fact with industrial use electrical and electronical products are commonly integrated into larger and more complex environments or systems that cannot easily be split up again when it comes to their disposal at the end of their life cycles.

As N.A.T. products are solely sold to industrial customers, by special arrangement at time of purchase the customer agreed to take the responsibility for a WEEE compliant disposal of the used N.A.T. product. Moreover, all N.A.T. products are marked according to the directive with a crossed out bin to indicate that these products within the European Community must not be disposed with regular waste.

If you have any questions on the policy of N.A.T. regarding the Directive 2011/65/EU of the European Parliament and of the Council of 8 June 2011 on the "Restriction of the use of certain Hazardous Substances in Electrical and Electronic Equipment" (RoHS) or the Directive 2002/95/EC of the European Commission on "Waste Electrical and Electronic Equipment" (WEEE) please contact N.A.T. by phone or e-mail.

5.3.3 Compliance to CE Directive

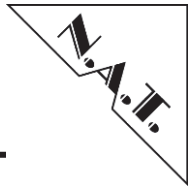
Compliance to the CE directive is declared. A 'CE' sign can be found on the PCB.

5.3.4 Product Safety

The board complies with EN60950 and UL1950.

5.3.5 Compliance to REACH

The REACH EU regulation (Regulation (EC) No 1907/2006) is known to N.A.T. GmbH. N.A.T. did not receive information from their European suppliers of substances of very high concern of the ECHA candidate list. Article 7(2) of REACH is notable as no substances are intentionally being released by NAT products and as no hazardous substances are contained. Information remains in effect or will be otherwise stated immediately to our customers.



6 Known Bugs / Restrictions

none

Appendix A: Reference Documentation

[1] Altera CycloneIII FPGA, Device Handbook Volume 1, V2.4, 07/2012
[2] FTDI USB-to-JTAG-Bridge, Datasheet FT_000061 V2.21, 06/2012
[3] XAPP1251 v1.0 04/ 2015
[4] XVC protocol specification and example designs available
[url:https://raw.githubusercontent.com/Xilinx/XilinxVirtualCable/master/README_XVC_v1_0.txt]

Appendix B: Document's History

Revision	Date	Description	Author
1.0	10.11.2014	Initial release	se
1.0	20.4.2015	Added content	te
1.1	15.5.2015	Updated Block diagram	hl
1.2	21.5.2015 7.10.2015	Updated Chapter "Programming via MCH" Removed Chapter 3.6.3 Pinout of rotary Switch Added function of green/yellow Led Added environmental conditions Some minor corrections	al hl
1.3	08.04.2016	Added description for using Xilinx Impact software Added description for changing Lattice Diamond TCK frequency	te
1.4	17.5.2016	Wording clarified for slave port selection	hl
1.4	18.05.2016	Added description for slave/master LED assignments	te
1.5	20.01.2018	Added description for programming via SVF file Removed MCHs and Power Module from JTAG target table (Chapter 3.4 Target Selection)	al
1.6	15.02.2018	Update Chapter 3.3.3 "Programming via SVF file" for web interface	al
1.7	11.04.2018	Reworked chapter 3.3 Programming via MCH: Added 3.3.3 Programming SVF file via URL	al
1.8	13.09.2018	Minor Changes (layout, typos, wording)	se
1.9	21.07.2020	Corrected typo in Table 2, page 19	te
	13.01.2021	Updated Table 6 – Pin Assignment JSM Connector	se
1.10	9.03.2021	Updated chapter 3.4 Target Selection	se