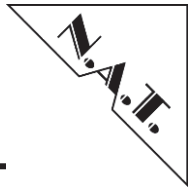


N.A.T. MCH
E1-T1-CLK Transceiver
Technical Reference Manual V1.3
HW Revision 1.0

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

The N.A.T. MCH has been designed by:

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

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

TABLE OF CONTENTS **4**
LIST OF TABLES **6**
LIST OF FIGURES **6**
CONVENTIONS **7**
1 INTRODUCTION **9**
2 OVERVIEW **10**
2.1 MAJOR FEATURES 10
2.2 BLOCK DIAGRAM 11
2.3 LOCATION DIAGRAM 12
3 BOARD FEATURES **13**
3.1 I²C-TO-SPI-INTERFACE 13
3.2 E1-T1 LINE INTERFACE 13
4 HARDWARE **14**
4.1 FRONT PANEL AND LEDS 14
4.2 CONNECTORS AND SWITCHES 15
4.2.1 MCH Connector 16
4.2.2 RJ45 Connector 16
5 PROGRAMMING NOTES **17**
5.1 SPI INTERFACE 17
5.2 I²C INTERFACE 17
5.3 REGISTER 17
5.3.1 BOARD_ID – 0x00 18
5.3.2 PCB_VERS – 0x01 19
5.3.3 FW_VERS – 0x02 19
5.3.4 DS26504_OUT – 0x03 20
5.3.5 DS26504_IN – 0x04 20
5.3.6 PCB_REL – 0x0A…0x0D 21
5.3.7 DS26504 Register Map - 0x10…0xFF 21
5.4 E1-T1-CLK-TRANSCEIVER CONFIGURATION 25
5.4.1 Write E1-T1-CLK-Transceiver Byte Register 26
5.4.2 OR Value to E1-T1-CLK-Transceiver Byte Register 26
5.4.3 AND Value to E1-T1-CLK-Transceiver Register 27
5.4.4 Get reference clock status of E1-T1-CLK-Transceiver 27
5.5 E1-T1-CLK-TRANSCEIVER DIAGNOSTIC 29
5.5.1 CLI Diagnostic Menu 29
5.5.2 CLI Configuration Command Interface 30
5.5.3 CLI Updating Firmware of E1-T1-CLK module 32
INSTALLATION **34**
5.6 SAFETY NOTE 34
5.7 INSTALLATION PREREQUISITES AND REQUIREMENTS 34
5.7.1 Requirements 34
5.7.2 Power supply 34

5.7.3 Automatic Power Up....34
5.8 STATEMENT ON ENVIRONMENTAL PROTECTION....35
5.8.1 Compliance to RoHS Directive....35
5.8.2 Compliance to WEEE Directive....35
5.8.3 Compliance to CE Directive....36
5.8.4 Product Safety....36
5.8.5 Compliance to REACH....36
6 KNOWN BUGS / RESTRICTIONS....37
APPENDIX A: MODULE CONFIGURATION FOR T1 FUNCTIONALITY....38
APPENDIX B: MODULE CONFIGURATION FOR E1 FUNCTIONALITY....45
APPENDIX C: REFERENCE DOCUMENTATION....52
DOCUMENT'S HISTORY....53

List of Tables

Table 1: List of used abbreviations 7
Table 2: N.A.T. MCH – Local I²C-Devices 13
Table 3: N.A.T. MCH E1-T1-CLK - LED Status 14
Table 4: J1: MCH Connector – Pin Assignment 16
Table 5: N.A.T.-MCH E1-T1-CLK-Transceiver – Register overview 18

List of Figures

Figure 1: E1-T1-CLK-Transceiver – Block Diagram 11
Figure 2: N.A.T. MCH E1-T1-CLK – Location Diagram 12
Figure 3: N.A.T. MCH E1-T1-CLK – Front Panel Mid-Size 14
Figure 4: N.A.T. MCH E1-T1-CLK– Connector Location – Overview 15

Conventions

If not otherwise specified, addresses and memory maps are written in hexadecimal notation, identified by *0x*. Table 1 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
CRC	Cyclic Redundancy Check
DDR3 SDRAM	Double Data Rate Synchronous Dynamic RAM
DIP SW	Dual In-Line Switch
E1	2.048 Mbit G.703 Interface
ECI	Ethernet Control Interface
EEPROM	Electrically Erasable PROM
ESSI	Extended Serial SONET/SDH Interface
FPGA	Field Programmable Gate Array
GbE	Gigabit Ethernet
H.110	Timeslot Interchange Bus
HDLC	High-Level Data Link Control
HS	Hot Swap
I²C	Inter-Integrated Circuit
I/O	Input/Output
IP	Internet Protocol
IPMB	Intelligent Platform Management Bus
IPMI	Intelligent Platform Management Interface
IRQ	Interrupt Request
iTDM	Internal TDM
JTAG	Joint Test Action Group
LIF	Line Interface
µC	Microcontroller
µTCA	Micro Telecommunications Computing Architecture
MAC	Media Access Control
MUX	Multiplexer
OC	Optical Carrier
PCB	Printed Circuit Board
PCI(e)	Peripheral Component Interconnect (Express)
QDR 2+ SRAM	Quad Data Rate Static RAM
Rx	Receiver
R/W	Read/Write
RAM	Random Access Memory
(P)ROM	(Programmable) Read Only Memory
PLL	Phase Locked Loop
SBI	Scalable Bandwidth Interface
SDH	Synchronous Digital Hierarchy
SFP	Small Form-Factor Pluggable
SONET	Synchronous Optical Networking
STM	Synchronous Transfer Mode
T1	1,544 Mbit G.703 Interface (USA)

Abbreviation	Description
TCKL	Telecom Clock
TDM	Time Division Multiplex
TSI	Time Slot Interchanger
Tx	Transmitter
XAUI	10 GbE (via 4x 3.125 GB/s)

1 Introduction

The E1-T1-CLK Transceiver is an enhancement for the N.A.T. MCH Base-Module v3.4 to realize an external BITS/SETS timing interface for telecom industry standards.

2 Overview

The N.A.T. MCH has usually two external synchronization interfaces using SMA coaxial connectors which require a non-gapped frequency input.

The E1-T1-CLK-Transceiver is an enhancement to use this interface as a synchronization interface for telecom industry standards. Instead of the SMA coaxial connectors one RJ45 connector is reused to be used as E1-T1-CLK Interface.

The telecom industry has its own synchronization standards which in North America is called BITS (Building Integrated Timing Supply). Outside of North America the timing standard is SETS (SDH Equipment Timing Source). The BITS interface is a T1 formatted (bipolar) data signal operating at 1.544Mb/s and is transmitted over twisted pair cable, balanced. The typical connector is an RJ-48 (physically identical to the RJ-45, but wired for T1). The SETS timing source physical interface has three different standards. The SETS implementation of this enhancement will use the E1 formatted balanced interface (RJ48).

2.1 Major Features

The E1-T1-CLK-Transceiver is mounted on the N.A.T. MCH Base-Module v3.4. Together with the RJ45-Clock assembly option of the N.A.T. MCH Base-Module it can connect to a synchronized E1 or T1 network; allowing the N.A.T.-MCH-Clock Module to extract or insert a clock signal. In case the clock is not, or no longer available, a holdover functionality can be provided by the NAT-MCH Clock-Module.
The E1/T1 Line interface Unit is realized with a DS26504 device from Maxim Dallas. This device can be configured by the on board Microcontroller via a SPI slave interface.

The DS26504 has the following main features:

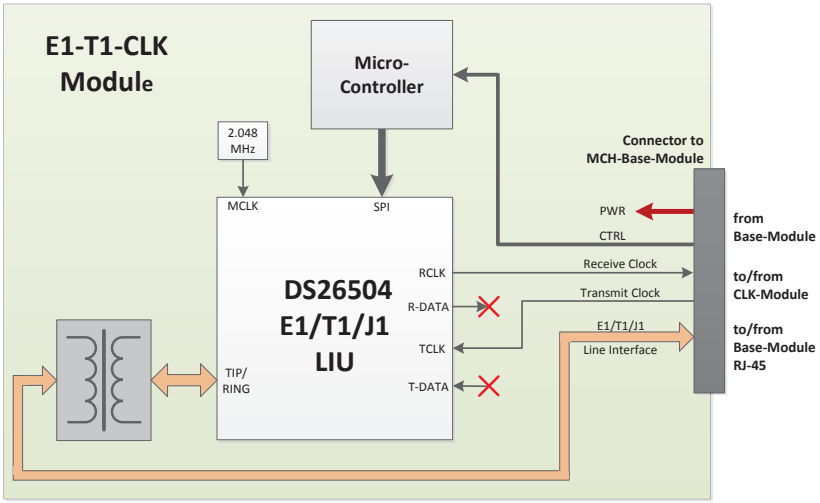
- Supports both Long-Haul and Short-Haul Trunks
- Internal Software-Selectable Receive-Side Termination for 75Ω/100Ω/120Ω
- 32-Bit or 128-Bit Crystal-Less Jitter Attenuator only requires a 2.048MHz Master Clock for both E1 and T1 with the option to use a 1.544MHz clock for T1
- Generates the appropriate Line Build-Outs, with and without return loss, for E1 and DSX-1 and CSU Line Build-Outs for T1

The data connections of the DS26504 towards the system side are not connected.
Only the RCLK (receiver clock) and TCLK (transmit clock) are connected to the standard N.A.T.-MCH Clock-Module.

2.2 Block Diagram

Figure 1 shows a block diagram of the **E1-T1-CLK-Transceiver**.

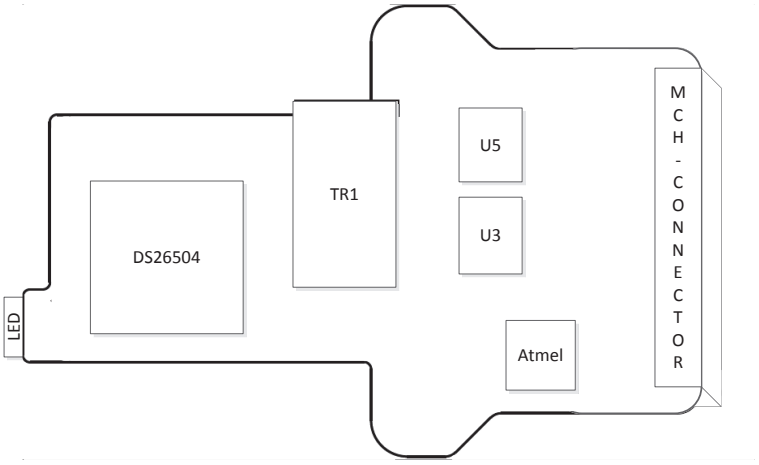
Figure 1: **E1-T1-CLK-Transceiver – Block Diagram**



2.3 Location Diagram

The position of important components is shown in the following location overview.

Figure 2: **N.A.T. MCH E1-T1-CLK – Location Diagram**



3 Board Features

The **E1-T1-CLK-Transceiver** can be divided into two functional blocks, which are described in the following paragraphs.

3.1 I²C-To-SPI-Interface

An ATxmega8E5 8-Bit micro controller from Atmel is functioning as a gateway between the MCH Base-Module and the E1/T1 Line interface device DS26504. The micro controller has an I²C connection towards the MCH Base-Module and a SPI connection towards the DS26504. Therefore the complete SPI functional option of the DS26504 is accessible from the MCH Base-Module. This allows facilitate use of the various configuration options of the LIU.
The E1-T1-CLK-Transceiver is accessible via the following I²C address:

Table 2: N.A.T. MCH – Local I²C-Devices

Device	Function	I²C-Address
E1-T1-CLK		0x08

3.2 E1-T1 Line Interface

The E1-T1 Line Interface is realized with the device DS26504 from Dallas Maxim. For a detailed description of the features please refer to the datasheet of this device [5].

4 Hardware

4.1 Front Panel and LEDs

The figure below shows the front panel of the **N.A.T. MCH E1-T1-CLK-Transceiver** when the E1-T1-CLK is mounted on the N.A.T.-MCH Base-Module v3.4 together with a N.A.T. HUB-PCIe-x48 module. The right-most RJ45 connector normally supporting a second Ethernet interface is reused to support the E1/T1 interface.
The E1-T1-CLK-Transceiver is equipped with a LED reflecting the status of the module.
For details on the other LEDs refer to the N.A.T.-MCH – User's Manual [1] or the N.A.T.-MCH HUB-Module PCIe – Technical Reference Manual [2].

Figure 3: **N.A.T. MCH E1-T1-CLK – Front Panel Mid-Size**

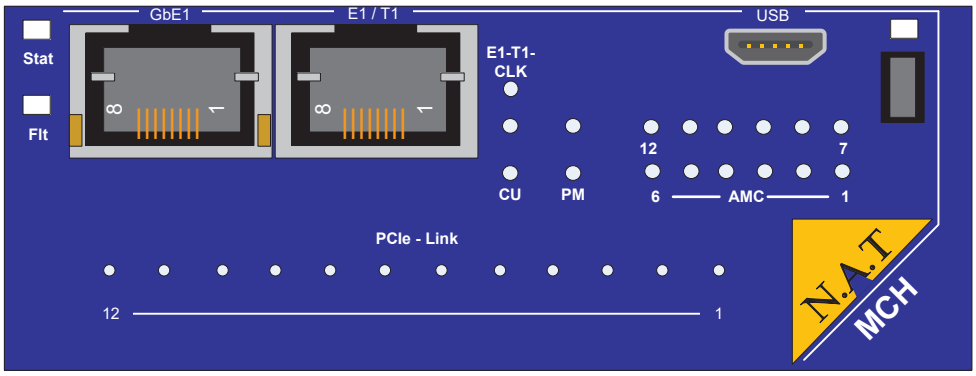


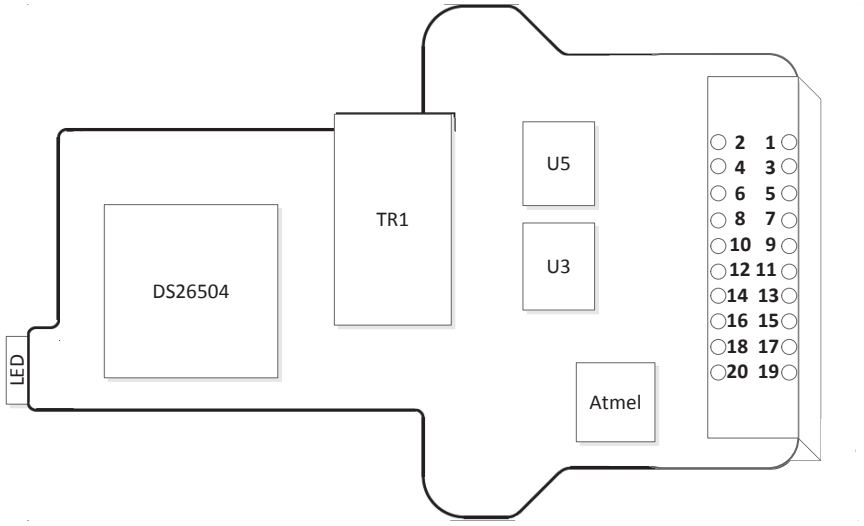
Table 3: N.A.T. MCH E1-T1-CLK - LED Status

LED State	Status
OFF	RLOS is detected
ON	RLOS is not detected

The green LED indicates the status of the E1-T1-CLK-Transceiver. This LED is on if the E1 or T1 signal is received by the DS26504 regardless which status RLOF is indicating. In case that a RLOS is detected the LED is turned off.

4.2 Connectors and Switches

Figure 4: **N.A.T. MCH E1-T1-CLK– Connector Location – Overview**



Please refer to the following tables to look up the connector pin assignment of the **N.A.T. MCH E1-T1-CLK**.

4.2.1 MCH Connector

The connection from the E1-T1-CLK-Transceiver towards the MCH has the following pin assignment:

Table 4: J1: MCH Connector – Pin Assignment

Pin #	Signal	Signal	Pin #
2	GND	+3.3V	1
4	RX_P / RJ45_Pin1	SDA / EXTREF_CONF1	3
6	RX_N / RJ45_Pin2	SCL / EXTREF_CONF2	5
8	TX_P / RJ45_Pin4	RCLK / EXTREF1_P	7
10	TX_N / RJ45_Pin5	/ EXTREF1_N	9
12	nc	/ EXTREF2_P	11
14	nc	TCLK / EXTREF2_N	13
16	nc	PDI_DAT / EXTREF_CONF3	15
18	nc	nc	17
20	PDI_CLK/RST / EXTREF_CONF4	GND	19

4.2.2 RJ45 Connector

The RJ45 connector implementation is as follows:

Pin Number	Pin Name	Description
1	RX tip	Receive tip
2	RX ring	Receive ring
3	-	Not connected
4	TX tip	Transmit tip
5	TX ring	Transmit ring
6	-	Not connected
7	-	Not connected
8	-	Not connected

5 Programming Notes

5.1 SPI Interface

The SPI interface on the **N.A.T.-MCH E1-T1-CLK-Transceiver** is used for accessing the complete SPI functional option of the DS26504 from the MCH Base-Module.

5.2 I²C Interface

The I²C interface is the main communication interface between the microcontroller of the **E1-T1-CLK-Transceiver** and the CPU of the **N.A.T.-MCH BASE-Module**.

5.3 Register

A register interface is implemented in the Atmel microcontroller. With the help of this interface different functions can be controlled and various identification values can be read. The complete function of the DS26504 can be configured via this register interface from the N.A.T.-MCH Base Module on start-up via a text based configuration file. This configuration file contains the initializing of the DS26504 to select the right functionality and is stored in a specific flash area on the N.A.T.-MCH Base Module.
Refer to Appendix A: Module configuration for T1 functionality or Appendix B: Module configuration for E1 functionality to see examples of the configuration file.

Table 5: N.A.T.-MCH E1-T1-CLK-Transceiver – Register overview

	7	6	5	4	3	2	1	0
0x00	BOARD_ID							
0x01	PCB_VERS							
0x02	FW_VERS							
0x03	DS26504_OUT							
0x04	DS26504_IN							
0x05.. 0x09	reserved							
0x0A.. 0x0D	PCB_REL							
0x0E.. 0x0F	reserved							
DS26504 Register Map addressable via an offset of 0x10 (besides the register addresses 0xE0-0xEFof the DS26504 which aren't accessible)								
0x10	TSTRREG							
0x11	IOCR1							
0x12	IOCR2							
0x13	T1RCR1							
0x14	T1RCR2							
…	…							
0xFD	Test Register 14 TEST14							
0xFE	Test Register 15 TEST15							
0xFF	Test Register 16 TEST16							

5.3.1 BOARD_ID – 0x00

Bit	Name	Description	Default	Access
7..0	BOARD_ID	The Board Identifier Register contains the Board ID that identifies the board as **N.A.T.-MCH E1-T1-CLK-Transceiver.**	0xF1	Read Only

5.3.2 PCB_VERS – 0x01

Bit	Name	Description	Default	Access
7..4	PCB_VERS_MAJ	The PCB Version Register contains the version code of the **N.A.T.-MCH E1-T1-CLK-Transceiver.** Bit 7 to 4 contain the major version and bit 3 to 0 contain the minor version. That means if the PCB version is e.g. v3.1 the PCB Version Register contains the value 0x31.	0xXX	Read Only
3..0	PCB_VERS_MIN		0xXX	Read Only

5.3.3 FW_VERS – 0x02

Bit	Name	Description	Default	Access
7..4	FW_VERS_MAJ	The Atmel Version Register contains the version of the Atmel firmware. Bit 7 to 4 contain the major version and bit 3 to 0 contain the minor version. That means if the firmware running on the Atmel is v1.3 the Firmware Version Register contains the value 0x13.	0xXX	Read Only
3..0	FW_VERS_MIN		0xXX	Read Only

5.3.4 DS26504_OUT – 0x03

Bit	Name	Description	Default	Access
7	---	A few relevant DS26504 output signals are connected to pins of the Atmel and are accessible via this Register bit7: --- bit6: TS_8K_4 - TSYNC, 8kHz Sync, 400Hz Sync. bit5: RS_8K - Receive Sync/8kHz Clock bit4: RSER - Receive Serial Data bit3: --- bit2: /DS_INT - /INT/JACKS0 - Active-Low Interrupt/Jitter Attenuator Clock Select 0 bit1: RLOF - RLOF_CCE - Receive Loss of Frame_Composite Clock Error bit0: RLOS - Receive Loss of Signal	0	Read Only
6	TS_8K_4		0	Read Only
5	RS_8K		0	Read Only
4	RSER		0	Read Only
3	---		0	Read Only
2	/INT JACKS0		1	Read Only
1	RLOF		0	Read Only
0	RLOS		0	Read Only

5.3.5 DS26504_IN – 0x04

Bit	Name	Description	Default	Access
7	---	A few relevant DS26504 input signals are connected to pins of the Atmel and are accessible via this Register bit7: --- bit6: TS_8K_4 - TSYNC, 8kHz Sync, 400Hz Sync. bit5: --- bit4: TSER - Transmit Serial Data bit3: --- bit2: /DS_INT - /INT/JACKS0 - Active-Low Interrupt/Jitter Attenuator Clock Select 0 bit1: TSTRST: Three-State Control and Device Reset. bit0: THZE: Transmit High-Impedance Enable.	0	Read Only
6	TS_8K_4		0	Read/ Write
5	---		0	Read Only
4	TSER		1	Read/ Write
3	---		0	Read Only
2	/INT JACKS0		1	Read Only
1	TSTRST		0	Read/ Write
0	THZE		0	Read/ Write

5.3.6 PCB_REL – 0x0A…0x0D

Byte	Name	Description	Default	Access
0x0A	PCB_REL_MIN	The PCB Release Registers contain the release code of the **N.A.T.-MCH E1-T1-CLK-Transceiver.** Byte 0x0A contains the minor release, byte 0x0B contains the middle release and byte 0x0C contains the major release. That means if the PCB release is e.g. 150310 the PCB Release Register contain the values 0x10, 0x03 and 0x15.	0xXX	Read Only
0x0B	PCB_REL_MID		0xXX	Read Only
0x0C	PCB_REL_MAJ		0xXX	Read Only
0x0D	---		0x00	Read Only

5.3.7 DS26504 Register Map - 0x10…0xFF

The complete DS26504 Register Map (DS26504 register address range 0x00…0xFF) is accessible via the register interface with an offset of 0x10 except the DS26504 register address range 0xE0…0xEF. For a detailed description of these registers refer to the datasheet of the DS26504. The following register map is extracted from this datasheet to give a summarized overview of these registers.

Byte	Name	Description	DS26504 Reg. Addr.	Access
0x10	TSTRREG	Test Reset Register	0x00	R/W
0x11	IOCR1	I/O Configuration Register 1	0x01	R/W
0x12	IOCR2	I/O Configuration Register 2	0x02	R/W
0x13	T1RCR1	T1 Receive Control Register 1	0x03	R/W
0x14	T1RCR2	T1 Receive Control Register 2	0x04	R/W
0x15	T1TCR1	T1 Transmit Control Register 1	0x05	R/W
0x16	T1TCR2	T1 Transmit Control Register 2	0x06	R/W
0x17	T1CCR	T1 Common Control Register	0x07	R/W
0x18	MCREG	Mode Configuration Register	0x08	R/W
0x19	TPCR1	Transmit PLL Control Register 1	0x09	R/W
0x1A	—	Transmit PLL Control Register 2	0x0A	R/W
0x1B	(Note 1)	Reserved	0x0B	—
0x1C	(Note 1)	Reserved	0x0C	—
0x1D	(Note 1)	Reserved	0x0D	—
0x1E	(Note 1)	Reserved	0x0E	—

Byte	Name	Description	DS26504 Reg. Addr.	Access
0x1F	(Note 1)	Reserved	0x0F	—
0x20	IDR	Device Identification Register	0x10	R
0x21	INFO1	Information Register 1	0x11	R
0x22	INFO2	Information Register 2	0x12	R
0x23	IIR	Interrupt Information Register	0x13	R
0x24	SR1	Status Register 1	0x14	R
0x25	IMR1	Interrupt Mask Register 1	0x15	R/W
0x26	SR2	Status Register 2	0x16	R
0x27	IMR2	Interrupt Mask Register 2	0x17	R/W
0x28	SR3	Status Register 3	0x18	R
0x29	IMR3	Interrupt Mask Register 3	0x19	R/W
0x2A	SR4	Status Register 4	0x1A	R
0x2B	IMR4	Interrupt Mask Register 4	0x1B	R/W
0x2C	INFO3	Information Register 3	0x1C	R
0x2D	E1RCR	E1 Receive Control Register	0x1D	R/W
0x2E	E1TCR	E1 Transmit Control Register	0x1E	R/W
0x2F	BOCC	BOC Control Register	0x1F	R/W
0x30	LBCR	Loopback Control Register	0x20	R/W
0x31	SR5	Status Register 5	0x21	R
0x32	IMR5	Internal Mask Register 5	0x22	R/W
0x33	(Note 1)	Reserved	0x23-0x2F	—
...	...	...	...	...
0x3F	(Note 1)	Reserved	0x23-0x2F	—
0x40	LIC1	Line Interface Control 1	0x30	R/W
0x41	LIC2	Line Interface Control 2	0x31	R/W
0x42	LIC3	Line Interface Control 3	0x32	R/W
0x43	LIC4	Line Interface Control 4	0x33	R/W
0x44	TLBC	Transmit Line Build-Out Control	0x34	R/W
0x45	(Note 1)	Reserved	0x35-0x3F	—
0x4E	...	...	...	...
0x4F	(Note 1)	Reserved	0x35-0x3F	—

Byte	Name	Description	DS26504 Reg. Addr.	Access
0x50	TAF	Transmit Align Frame Register	0x40	R/W
0x51	TNAF	Transmit Non-Align Frame Register	0x41	R/W
0x52	TsiAF	Transmit Si Align Frame	0x42	R/W
0x53	TSiNAF	Transmit Si Non-Align Frame	0x43	R/W
0x54	TRA	Transmit Remote Alarm Bits	0x44	R/W
0x55	Tsa4	Transmit Sa4 Bits	0x45	R/W
0x56	Tsa5	Transmit Sa5 Bits	0x46	R/W
0x57	Tsa6	Transmit Sa6 Bits	0x47	R/W
0x58	Tsa7	Transmit Sa7 Bits	0x48	R/W
0x59	Tsa8	Transmit Sa8 Bits	0x49	R/W
0x5A	TSACR	Transmit Sa Bit Control Register	0x4A	R/W
0x5B	(Note 1)	Reserved	0x4B-0x4F	—
0x5E	…	…	…	…
0x5F	(Note 1)	Reserved	0x4B-0x4F	—
0x60	RFDL	Receive FDL Register	0x50	R
0x61	TFDL	Transmit FDL Register	0x51	R/W
0x62	RFDLM1	Receive Facility Data Link Match Register 1	0x52	R/W
0x63	RFDLM2	Receive Facility Data Link Match Register 2	0x53	R/W
0x64	(Note 1)	Reserved	0x54-0x55	—
0x65	(Note 1)	Reserved	0x54-0x55	—
0x66	RAF	Receive Align Frame Register	0x56	R
0x67	RNAF	Receive Non-Align Frame Register	0x57	R
0x68	RsiAF	Receive Si Align Frame	0x58	R
0x69	RSiNAF	Receive Si Non-Align Frame	0x59	R
0x6A	RRA	Receive Remote Alarm Bits	0x5A	R
0x6B	Rsa4	Receive Sa4 Bits	0x5B	R
0x6C	Rsa5	Receive Sa5 Bits	0x5C	R
0x6D	Rsa6	Receive Sa6 Bits	0x5D	R
0x6E	Rsa7	Receive Sa7 Bits	0x5E	R
0x6F	Rsa8	Receive Sa8 Bits	0x5F	R
0x70	(Note 1)	Reserved	0x60-	—

Byte	Name	Description	DS26504 Reg. Addr.	Access
			0xDF	
…	…	…	…	…
0xEF	(Note 1)	Reserved	0x60-0xDF	—
Note: The 16 Test Registers of the DS26504 don't have an address offset against the register map of the DS26504 [5] and are accessible due to this arrangement!				
0xF0	(Note 2)	Test Register 1 TEST1	0xF0	R/W
0xF1	(Note 2)	Test Register 2 TEST2	0xF1	R/W
0xF2	(Note 2)	Test Register 3 TEST3	0xF2	R/W
0xF3	(Note 2)	Test Register 4 TEST4	0xF3	R/W
0xF4	(Note 2)	Test Register 5 TEST5	0xF4	R/W
0xF5	(Note 2)	Test Register 6 TEST6	0xF5	R/W
0xF6	(Note 2)	Test Register 7 TEST7	0xF6	R/W
0xF7	(Note 2)	Test Register 8 TEST8	0xF7	R/W
0xF8	(Note 2)	Test Register 9 TEST9	0xF8	R/W
0xF9	(Note 2)	Test Register 10 TEST10	0xF9	R/W
0xFA	(Note 2)	Test Register 11 TEST11	0xFA	R/W
0xFB	(Note 2)	Test Register 12 TEST12	0xFB	R/W
0xFC	(Note 2)	Test Register 13 TEST13	0xFC	R/W
0xFD	(Note 2)	Test Register 14 TEST14	0xFD	R/W
0xFE	(Note 2)	Test Register 15 TEST15	0xFE	R/W
0xFF	(Note 2)	Test Register 16 TEST16	0xFF	R/W

5.4 E1-T1-CLK-Transceiver Configuration

The E1-T1-CLK-Transceiver mounted on the N.A.T.-MCH can be configured via a text based script file, similar to the configuration file used for the Ethernet switch or Clock modules on the N.A.T.-MCH.
The different functions of the E1-T1-CLK-Transceiver can be set by using so called "configuration items". Each configuration item consists of an identifier and one or more parameters.
Comments within the configuration begin with a hash character ('#'), like:

```
#
# Item << e1t1clk_wreg>>: write e1t1clk module register
#
```

The current E1-T1-CLK-Transceiver configuration can be downloaded from the N.A.T.-MCH e.g. by using the "Backup Settings" function in the web based configuration interface (for details refer to chapter 9 in the N.A.T.-MCH-User Manual [1]). The generated text file can be changed with a standard text editor and can be used as a starting point for a user defined configuration. The changed configuration can be uploaded again, e.g. via the web based configuration interface.
The following chapter describes the configuration items which can be used to configure the E1-T1-CLK-Transceiver. Keep in mind that when configuring the register of DS26504 an offset of 0x10 has to be added to the DS26504 register address.

A so called device locator identifier is specified at the beginning of the part of the script configuration file which is intended for the E1-T1-CLK-Transceiver module configuration:

```
#
# ===  device location identifier  ===
#

mch_id = 0
mez_id = 0
inst_id = 1
```

These three parameters are also needed when configuring the E1-T1-CLK-Transceiver via the CLI command < cfg_ctrl_cmd> (for details refer to 5.5 E1-T1-CLK-Transceiver Diagnostic).

5.4.1 Write E1-T1-CLK-Transceiver Byte Register

Description:
Write a value to an E1-T1-CLK-Transceiver byte register.

Syntax:

```
e1t1clk_wreg = offs, value
```

Parameter Description:

```
offs: register offset

value:      value to write
```

Example:

```
e1t1clk_wreg = 0x10, 0xff
```

Write value 0xff to register at offset 0x10.

5.4.2 OR Value to E1-T1-CLK-Transceiver Byte Register

Description:
ORs a value to an E1-T1-CLK-Transceiver byte register. This can be used to set specific bits of a register.

Syntax:

```
e1t1clk_orreg = offs, value
```

Parameter Description:

```
offs: register offset

value:      value to OR with current register value
```

Example:

```
e1t1clk_orreg = 0x10, 0x01
```

Set the bit 0 (0x01) in E1-T1-CLK-Transceiver byte register at offset 0x10.

5.4.3 AND Value to E1-T1-CLK-Transceiver Register

Description:
ANDs a value to an E1-T1-CLK-Transceiver byte register. This can be used to clear specific bits of a register.

Syntax:

```
e1t1clk_andreg = offs, value
```

Parameter Description:

```
offs: register offset

value:      value to AND with current register value
```

Example:

```
e1t1clk_andreg = 0x10, 0x7f
```

Clear the bit 7 (0x80) in E1-T1-CLK-Transceiver register at offset 0x10.

5.4.4 Get reference clock status of E1-T1-CLK-Transceiver

Description:
This script configuration file command is actually desired only to request the status of the reference clock by using the CLI command <cfg_ctrl_cmd> in a similar way as for the other script configuration file commands. It checks the status of the configured clock module PLL1 and the bits RLOS and RLOF of the register DS26504_OUT – 0x03. The response has the following syntax:

Reference clock status: <Clock module PLL1 status>, <E1-T1 Status>

<Clock module PLL1 status>:	Free-run Holdover Locked Prelocked 2 Prelocked Loss-of-lock
<E1-T1 Status>:	a) “empty” := “everything is fine” b) RLOS c) RLOF

Syntax:

```
e1t1clk_getstatus
```

Parameter Description:

```
none
```

Example:

```
nat> cfg_ctrl_cmd
Enter CLI command line
[CLI CMD LINE]:>0, 0, 1, e1t1clk_getstatus
Reference clock status: Locked, RLOS
nat>
```

Shows the status of the PLL1 and either RLOS or RLOF in case these signals are set by the E1-T1-CLK-Transceiver.

5.5 E1-T1-CLK-Transceiver Diagnostic

The E1-T1-CLK-Transceiver is accessible via the N.A.T.-MCH CLI for diagnostic and configuration purposes.

5.5.1 CLI Diagnostic Menu

Once when a CLI interface is set up to the N.A.T.-MCH the diagnostic menu of the E1-T1-CLK-Transceiver can be reached by entering diag and 14 as follow:

```
nat>
nat> diag
  [ 0] : no action (unsupported)
  [ 1] : (submenu) INFO menu
  [ 2] : (submenu) UPDATE menu
  [ 3] : (submenu) I2C menu
  [ 4] : (submenu) DSPI menu
  [ 5] : (submenu) ETH menu
  [ 6] : (submenu) PCIe PCB menu
  [ 7] : (submenu) SRIO PCB menu
  [ 8] : (submenu) XAUI PCB menu
  [ 9] : (submenu) CLOCK PCB menu
  [10] : (submenu) RTM PCB menu
  [11] : (submenu) AVR programming menu
  [12] : (submenu) ITDM
  [13] : (submenu) NVRAM
  [14] : (submenu) FrontIF
  [15] : switch debug
  [ ?] : ?: help
  [ h] : h: help
  [ q] : q: quit
DIAG (RET=0/0x0): 14
  [ 0] : no action (unsupported)
  [ 1] : (submenu) AVR PDI PROG menu
  [ 2] : initialize FRONTIF on MCH
  [ 3] : terminate FRONTIF on MCH
  [ 4] : read Atmel register
  [ 5] : write Atmel register
  [ 6] : read ref clock status
  [ ?] : ?: help
  [ h] : h: help
  [ q] : q: quit submenu
```

5.5.2 CLI Configuration Command Interface

From MCH firmware release version "MCH CM/ShM Firmware V2.17d Engineering (08:27:25 May 12 2015)" on the E1-T1-CLK-Transceiver and Clock-Module (PCB V4.1) commands of the script configuration file can also be executed via the CLI interface command <cfg_ctrl_cmd>. The following example shows how this command can be used:

```
nat>
nat> cfg_ctrl_cmd
Enter CLI command line
[CLI CMD LINE]:>0, 0, 1, e1t1clk_getstatus
Reference clock status: Free-run, RLOS
nat>
```

A prefix has to be added in front of the script configuration file command (e.g. 0, 0, 1,). This prefix depends on the device locator identifier of the module that shall be configured by this command.

Module	Prefix <device locator identifier>
E1-T1-CLK-Transceiver	0, 0, 1,
Clock-Module	0, 1, 0,

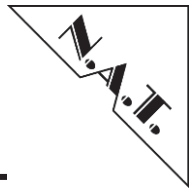
Table 5-1: CLI command <cfg_ctrl_cmd> command line prefix

The command <cfg_ctrl_cmd> can be used to reconfigure the clock module during run time. In case that a clock script configuration file for E1 configuration is permanently stored in the Flash of the MCH the clock module can be reconfigured towards a T1 configuration by executing the following CLI command sequence:

```
nat>
nat> cfg_ctrl_cmd
Enter CLI command line
[CLI CMD LINE]:>0, 1, 0, clk_pll_ref  =  1,   1544000
nat> cfg_ctrl_cmd
Enter CLI command line
[CLI CMD LINE]:>0, 1, 0, clk_oc_cfg  =  1,  0
nat> cfg_ctrl_cmd
Enter CLI command line
[CLI CMD LINE]:>0, 1, 0, clk_oc_cfg  =  4,  1544000
nat> cfg_ctrl_cmd
Enter CLI command line
[CLI CMD LINE]:>0, 1, 0, clk_out  = 42,  48
nat>
```

This command sequence is similar to the clock script configuration file with the following command lines for the clock module (except that clk_pll_mode = 1, 2, 1 isn't mentioned since it was already set by script configuration file in the example above):

```
# BITS
clk_pll_ref  =  1,   1544000
clk_pll_mode  = 1, 2, 1
clk_oc_cfg  =  4,  1544000
```



In case the clock shall be set to E1 configuration the following CLI command sequence has to be executed:

```
nat>
nat> cfg_ctrl_cmd
Enter CLI command line
[CLI CMD LINE]:>0, 1, 0, clk_pll_ref  =  1,    2048000
nat> cfg_ctrl_cmd
Enter CLI command line
[CLI CMD LINE]:>0, 1, 0, clk_oc_cfg  =  4,  0
nat> cfg_ctrl_cmd
Enter CLI command line
[CLI CMD LINE]:>0, 1, 0, clk_oc_cfg  =  1,  2048000
nat> cfg_ctrl_cmd
Enter CLI command line
[CLI CMD LINE]:>0, 1, 0, clk_out  = 42,  45
nat>
```

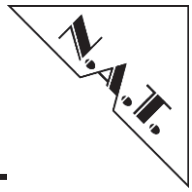
This command sequence is similar to the clock script configuration file with the following command lines (except that clk_pll_mode = 1, 2, 1 isn't mentioned since it was already set by script configuration file):

```
# SETS
clk_pll_ref  =  1,    2048000
clk_pll_mode  = 1, 2, 1
clk_oc_cfg  =  4,  2048000
```

5.5.3 CLI Updating Firmware of E1-T1-CLK module

From MCH firmware release version "MCH CM/ShM Firmware V2.17f Engineering (12:04:46 Aug 28 2015) the firmware on the E1-T1-CLK-Transceiver can be updated via the following CLI sequence:

```
nat>
nat> diag
FPGA already initialized
  [ 0] : no action (unsupported)
  [ 1] : (submenu) INFO menu
  [ 2] : (submenu) UPDATE menu
  [ 3] : (submenu) I2C menu
  [ 4] : (submenu) DSPI menu
  [ 5] : (submenu) ETH menu
  [ 6] : (submenu) PCIe PCB menu
  [ 7] : (submenu) SRIO PCB menu
  [ 8] : (submenu) XAUI PCB menu
  [ 9] : (submenu) CLOCK PCB menu
  [10] : (submenu) RTM PCB menu
  [11] : (submenu) AVR programming menu
  [12] : (submenu) ITDM
  [13] : (submenu) NVRAM
  [14] : (submenu) FrontIF
  [15] : switch debug
  [ ?] : ?: help
  [ h] : h: help
  [ q] : q: quit
DIAG (RET=0/0x0): 14
  [ 0] : no action (unsupported)
  [ 1] : (submenu) AVR PDI PROG menu
  [ 2] : initialize FRONTIF on MCH
  [ 3] : terminate FRONTIF on MCH
  [ 4] : read Atmel register
  [ 5] : write Atmel register
  [ 6] : read ref clock status
  [ 7] : initialize PDI IF on MCH
  [ 8] : terminate PDI IF on MCH
  [ ?] : ?: help
  [ h] : h: help
  [ q] : q: quit submenu
FRONTIF (RET=0/0x0): 1
  [ 0] : no action (unsupported)
  [ 1] : (submenu) AVR PROG menu
  [ 2] : program firmware to AVR flash
  [ 3] : verify firmware in AVR flash
  [ 4] : program AVR eeprom
  [ 5] : verify AVR eeprom
  [ 6] : AVR read device id
  [ ?] : ?: help
  [ h] : h: help
  [ q] : q: quit submenu
AVR PDI (RET=0/0x0): 2
AVR PDI device numbers:
```



NMCH_E1_T1_CLOCK_MOD : 6
NMCH_AVR_PDI_LED_MOD : 7
Choose AVR PDI device (RET=0/0x0): **6**
Enter host and file name of AVR firmware hex-image
[IP:FILENAME]:> *<enter path to firmware on TFTP server>*

The command line requests a <path to firmware on TFTP server> for entering the path to the hex file of the new firmware (e.g. 192.168.137.70:/home/download/released/mch_e1t1clk_avr_pcb1_0_v1_0.hex). The firmware is updated when the new firmware was entered and confirmed by pressing the button <ENTER>. This may takes a while (~ 1 minute) and no update proceed information is shown during this time. Once the new firmware is programmed a confirmation is shown in the CLI like the following example shows:

Example:
Nmch_AvrPDIProgFile: sizeof(fname) 256
TFTP: getting BIN file:
192.168.137.70:/home/download/released/mch_e1t1clk_avr_pcb1_0_v1_0.hex
...
TFTP: getting file done (18431 bytes)
Nmch_AvrPDIProgFile: bsize 6544
Wait until programming session is completed
.....

Programming completed!

Finished programm and verify firmware image
AVR PDI (RET=2/0x2): q ← quite submenue [1] : (submenu) AVR PROG menu with <q>
FRONTIF (RET=0/0x0):

Installation

5.6 Safety Note

To ensure proper functioning of the **N.A.T.-MCH E1-T1-CLK-Transceiver** during its usual life-time take refer to the safety note section of the **N.A.T.-MCH BASE-Module** Technical Reference Manual before handling the board.

5.7 Installation Prerequisites and Requirements

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

5.7.1 Requirements

The **N.A.T.-MCH E1-T1-CLK-Transceiver** is always fixed mounted on a **N.A.T.-MCH BASE-Module**. Therefore please refer to the requirements section of the **N.A.T.-MCH BASE-Module** Technical Reference Manual.

5.7.2 Power supply

The power supply for the **N.A.T.-MCH E1-T1-CLK-Transceiver** must meet the following specifications:

- required for the Module:
  - + 3,3V / *0.15A* max.

5.7.3 Automatic Power Up

Power ramping/monitoring and power up reset generation is done by the **N.A.T.-MCH BASE-Module**
In the following situations the **N.A.T.-MCH BASE-Module** will automatically be reset and proceed with a normal power up.

- The voltage sensor generates a reset, when +12 V voltage level drops below 8V.

5.8 Statement on Environmental Protection

5.8.1 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 poly-brominated 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.

5.8.2 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.

5.8.3 Compliance to CE Directive

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

5.8.4 Product Safety

The board complies with EN60950 and UL1950.

5.8.5 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: Module configuration for T1 functionality

This appendix shows an example of how to configure the N.A.T. Clock module and the E1-T1-CLK-Transceiver via the configuration file for T1 functionality. The configuration belongs to the following system configuration:

```
**********************************************************************
*** MCH CM/ShM Firmware V2.17d Engineering (08:27:25 May 12 2015) ***
**********************************************************************

NAT-MCH HW: PCB V3.4 Rev 120712  FPGA V1.8  AVR 1.2 - sn: 110334-7986 - Rel:120712
      AOPT: 0x3e - SRAM, E1-T1-CLK, HS Ctrl, 2nd FRT ETH, LED MOD
CLK MOD: PCB V4.1  MC V1.0  FPGA V1.2 (assembly option:  FPGA  PLL  HCSL buffer  PCIe Clock Gen) - sn: 6279 - Rel:130829
HUB MOD: PCB PCIe-x48 V2.4  MC V1.10  FPGA V1.5 (assembly option -X48 LOSC) - sn: 1357 - Rel:140411 - ChipRev: ca
```

Clock and E1-T1-CLK configuration:

```
################################################################################
###         Clock Module configuration for NAT-MCH                           ###
################################################################################

#
# === device location identifier ===
#

mch_id = 0
mez_id = 1
inst_id = 0


#
# Item <<clk_pcb_maj>>: Clock PCB major version
# Item <<clk_pcb_min>>: Clock PCB minor version
#
# Syntax: clk_pcb_maj = maj_ver
# Syntax: clk_pcb_min = min_ver
#
# Params: maj_ver: major PCB version
# Params: min_ver: minor PCB version
#

clk_pcb_maj  =  4
clk_pcb_min  =  1


#
# Item <<clk_pll_ref>>: PLL reference input configuration
#
# Syntax: clk_pll_ref = ref, freq
#
# Params: ref:  PLL reference input
#             1 -  PLL input IC1
#             2 -  PLL input IC2
#             3 -  PLL input IC3
#             4 -  PLL input IC4
#             5 -  PLL input IC5
#             6 -  PLL input IC6
#             7 -  PLL input IC7
#             8 -  PLL input IC8
#
#       freq: input clock frequency in Hz
#             Please refer to the NAT-MCH Users Manual for a list
#             of supported frequencies!
#

# PLL ref. input accepts 1.544MHz
# configure PLL input IC1 with input clock frequency 1.544MHz of RCLK
# PLL 1 with IC1 has to be used for the input clock frequency!
# Otherwise the diagnostic functionality regarding reference clock
# status (i.e. locked, unlocked, holdover) doesn't work!
clk_pll_ref  =  1,    1544000
```

```
clk_pll_ref  =  2,        0
clk_pll_ref  =  3,        0
clk_pll_ref  =  4,        0
clk_pll_ref  =  5,        0
clk_pll_ref  =  6,        0
clk_pll_ref  =  7,        0
clk_pll_ref  =  8,        0


#
# Item <<clk_pll_mode>>: PLL mode configuration
#
# Syntax: clk_pll_mode = pll#, mode, ref
#
# Params: #:    number of PLL to configure
#           1 -  PLL #1
#           2 -  PLL #2
#
#       mode: PLL operation mode
#           1 -  automatic selection between IC1 - IC8
#           2 -  forced mode (reference according to parameter ref)
#           3 -  free running mode
#
#       ref:  PLL reference input (only for forced mode)
#           1 -  PLL input IC1
#           2 -  PLL input IC2
#           3 -  PLL input IC3
#           4 -  PLL input IC4
#           5 -  PLL input IC5
#           6 -  PLL input IC6
#           7 -  PLL input IC7
#           8 -  PLL input IC8
#

# E1-T1-CLK configure RCLK as input signal on PLL #1: PLL #1, forced mode, PLL input IC1
# Note: PLL #1 (IC1) has to be used due to diagnostic functionality regarding reference
#       clock status!
clk_pll_mode  = 1, 2, 1
clk_pll_mode  = 2, 1, 0


#
# Item <<clk_oc_cfg>>: clock output configuration
#
# Syntax: clk_oc_cfg = oc, freq
#
# Params: oc: output clock # of pll
#            1 -  PLL OC1
#            2 -  PLL OC2
#            3 -  PLL OC3
#            4 -  PLL OC4
#            5 -  PLL OC5
#            6 -  PLL OC6
#            7 -  PLL OC7
#
# Params: freq: output clock frequency (in Hz)
#

clk_oc_cfg  =  1,  0
clk_oc_cfg  =  2,  0
clk_oc_cfg  =  3,  0
# for 1.544MHz we have to use OC4...OC7 (OC1...OC3 can't be configured with this frequency)
clk_oc_cfg  =  4,  1544000
clk_oc_cfg  =  5,  0
clk_oc_cfg  =  6,  0
clk_oc_cfg  =  7,  0



#
# Item <<clk_out>>: clock output configuration
#
# Syntax: clk_out = dst, src
#
# HCSL buffer detected
# - any source for CLK 3 output other than 0 will enable output of HCSL buffer
```

```
#
# Params: dst: destination clock identifier
#            1 -  CLK1 AMC  1
#            2 -  CLK1 AMC  2
#            3 -  CLK1 AMC  3
#            4 -  CLK1 AMC  4
#            5 -  CLK1 AMC  5
#            6 -  CLK1 AMC  6
#            7 -  CLK1 AMC  7
#            8 -  CLK1 AMC  8
#            9 -  CLK1 AMC  9
#           10 -  CLK1 AMC 10
#           11 -  CLK1 AMC 11
#           12 -  CLK1 AMC 12
#           13 -  CLK2 AMC  1
#           14 -  CLK2 AMC  2
#           15 -  CLK2 AMC  3
#           16 -  CLK2 AMC  4
#           17 -  CLK2 AMC  5
#           18 -  CLK2 AMC  6
#           19 -  CLK2 AMC  7
#           20 -  CLK2 AMC  8
#           21 -  CLK2 AMC  9
#           22 -  CLK2 AMC 10
#           23 -  CLK2 AMC 11
#           24 -  CLK2 AMC 12
#           25 -  CLK3 AMC  1
#           26 -  CLK3 AMC  2
#           27 -  CLK3 AMC  3
#           28 -  CLK3 AMC  4
#           29 -  CLK3 AMC  5
#           30 -  CLK3 AMC  6
#           31 -  CLK3 AMC  7
#           32 -  CLK3 AMC  8
#           33 -  CLK3 AMC  9
#           34 -  CLK3 AMC 10
#           35 -  CLK3 AMC 11
#           36 -  CLK3 AMC 12
#           37 -  CLK1 Update
#           38 -  CLK3 Update
#           39 -  EXT single ended 1 (INPUT  SMA 1)
#           40 -  EXT single ended 2 (OUTPUT SMA 1)
#           41 -  EXT single ended 3 (INPUT  SMA 2)
#           42 -  EXT single ended 4 (OUTPUT SMA 2)
#           43 -  EXT differential 1 (RJ45)
#           44 -  EXT differential 2 (RJ45)
#           45 -  PLL IC1
#           46 -  PLL IC2
#           47 -  PLL IC3
#           48 -  PLL IC4
#           49 -  PLL IC5
#           50 -  PLL IC6
#           51 -  PLL IC7
#           52 -  PLL IC8
#           53 -  PLL SYNC1
#           54 -  PLL SYNC2
#           55 -  PLL SYNC3
#           56 -  SYNC CLK input
#
# Params: src: source clock identifier
#            0 -  disabled
#            1 -  CLK1 AMC  1
#            2 -  CLK1 AMC  2
#            3 -  CLK1 AMC  3
#            4 -  CLK1 AMC  4
#            5 -  CLK1 AMC  5
#            6 -  CLK1 AMC  6
#            7 -  CLK1 AMC  7
#            8 -  CLK1 AMC  8
#            9 -  CLK1 AMC  9
#           10 -  CLK1 AMC 10
#           11 -  CLK1 AMC 11
#           12 -  CLK1 AMC 12
#           13 -  CLK2 AMC  1
#           14 -  CLK2 AMC  2
#           15 -  CLK2 AMC  3
```

```
#          16 -  CLK2 AMC  4
#          17 -  CLK2 AMC  5
#          18 -  CLK2 AMC  6
#          19 -  CLK2 AMC  7
#          20 -  CLK2 AMC  8
#          21 -  CLK2 AMC  9
#          22 -  CLK2 AMC 10
#          23 -  CLK2 AMC 11
#          24 -  CLK2 AMC 12
#          25 -  CLK3 AMC  1
#          26 -  CLK3 AMC  2
#          27 -  CLK3 AMC  3
#          28 -  CLK3 AMC  4
#          29 -  CLK3 AMC  5
#          30 -  CLK3 AMC  6
#          31 -  CLK3 AMC  7
#          32 -  CLK3 AMC  8
#          33 -  CLK3 AMC  9
#          34 -  CLK3 AMC 10
#          35 -  CLK3 AMC 11
#          36 -  CLK3 AMC 12
#          37 -  CLK1 Update
#          38 -  CLK3 Update
#          39 -  EXT single ended 1 (INPUT  SMA 1)
#          40 -  EXT single ended 2 (OUTPUT SMA 1)
#          41 -  EXT single ended 3 (INPUT  SMA 2)
#          42 -  EXT single ended 4 (OUTPUT SMA 2)
#          43 -  EXT differential 1 (RJ45)
#          44 -  EXT differential 2 (RJ45)
#          45 -  PLL OC1
#          46 -  PLL OC2
#          47 -  PLL OC3
#          48 -  PLL OC4
#          49 -  PLL OC5
#          50 -  PLL OC6
#          51 -  PLL OC7
#          52 -  PLL FSYNC 8kHz
#          53 -  PLL MFSYNC 2kHz
#          54 -  12.8MHz OSC
#          55 -  25MHz OSC (only with HCSL option)
#          56 -  LEVEL 0
#          57 -  LEVEL 1
#          58 -  SYNC_CLK
#

clk_out  =  1,  0
clk_out  =  2,  0
clk_out  =  3,  0
clk_out  =  4,  0
clk_out  =  5,  0
clk_out  =  6,  0
clk_out  =  7,  0
clk_out  =  8,  0
clk_out  =  9,  0
clk_out  = 10,  0
clk_out  = 11,  0
clk_out  = 12,  0
clk_out  = 13,  0
clk_out  = 14,  0
clk_out  = 15,  0
clk_out  = 16,  0
clk_out  = 17,  0
clk_out  = 18,  0
clk_out  = 19,  0
clk_out  = 20,  0
clk_out  = 21,  0
clk_out  = 22,  0
clk_out  = 23,  0
clk_out  = 24,  0
clk_out  = 25,  0
clk_out  = 26,  0
clk_out  = 27,  0
clk_out  = 28,  0
clk_out  = 29,  0
clk_out  = 30,  0
clk_out  = 31,  0
```

```
clk_out  = 32,  0
clk_out  = 33,  0
clk_out  = 34,  0
clk_out  = 35,  0
clk_out  = 36,  0
clk_out  = 37,  0
clk_out  = 38,  0
clk_out  = 39,  0
# MCH firmware drives Extref2 (reg 0x31 = value 40 here) and Extref3 (reg 0x32 = value 41 here)
# low due to over talking of TCLK. Therefore these configuration items have to be removed here
# from script (commented out via # below)!!!
#clk_out  = 40,  0
#clk_out  = 41,  0
# TCLK: destination 42 (EXT single ended 4 (OUTPUT SMA 2)) - with source clock of 48 -  PLL OC4
# Note: We have to use PLL OC4 instead of PLL OC1 for T1 due to 1.544MHz!
# (for details refer to comments above)
clk_out  = 42,  48
clk_out  = 43,  0
clk_out  = 44,  0
# destination clock 45 -  PLL IC1 with source clock of 39 -  EXT single ended 1 (INPUT  SMA 1)
# configured (EXT single ended 1 (INPUT  SMA 1) = RCLK)
clk_out  = 45,  39
clk_out  = 46,  0
clk_out  = 47,  0
clk_out  = 48,  0
clk_out  = 49,  0
clk_out  = 50,  0
clk_out  = 51,  0
clk_out  = 52,  0
clk_out  = 53,  0
clk_out  = 54,  0
clk_out  = 55,  0
clk_out  = 56,  0


#
# Item <<clk_wreg>>: write clock module register
#
# Syntax: clk_wreg = offs, value
#
# Params: offs: register offset
#         value: value to write
#


#
# Item <<clk_orreg>>: OR value to clock module register
#
# Syntax: clk_orreg = offs, value
#
# Params: offs: register offset
#         value: value to OR with current register value
#


#
# Item <<clk_andreg>>: AND value with clock module register
#
# Syntax: clk_andreg = offs, value
#
# Params: offs: register offset
#         value: value to AND with current register value
#



#
# Item <<clk_wblk>>: write block in clock module
#
# Syntax: clk_wblk = inst, len, data[0], data[1], ..., data[len - 1]
#
# Params: inst: addressed instance on clock module
#         len: number of bytes to write
#         data[n]: n-th data byte to write
#
```

```
#
# Item <<clk_wpllreg>>: write clock module PLL register
#
# Syntax: clk_wpllreg = offs, value
#
# Params: offs: PLL register offset
#         value: value to write
#



################################################################################
###          E1T1CLK Module configuration for NAT-MCH                ###
################################################################################

#
# ===  device location identifier  ===
#
# E1-T1-CLK module:
#           mch_id  = 0 = MCH1; 1 = MCH2
#           mez_id  = 0 (Base module)
#           inst_id = 1 (E1-T1-CLK)

mch_id = 0
mez_id = 0
inst_id = 1


#
# Item <<e1t1clk_wreg>>: write e1t1clk module register
#
# Syntax: e1t1clk_wreg = offs, value
#
# Params: offs: register offset
#         value: value to write
#
# Note: An offset of 0x10 has to be added on the register address of the DS26504
#       for configuring a register in DS26504!

# set DS26504 register MCREG (addr 0x08 in DS26504, addr 0x08 + 0x10 Atmel register --> 0x18)
# to T1 ESF Framing Mode for receive and transmit path --> 0x11
e1t1clk_wreg = 0x18, 0x11

# T1RCR1, T1RCR2, T1TCR1, T1TCR2, T1CCR
# Register Name: T1RCR1
# Register Description: T1 Receive Control Register 1
# Register Address: 03h
e1t1clk_wreg = 0x13, 0x00

# Register Name: T1TCR2
# Register Description: T1 Transmit Control Register 2
# Register Address: 06h
# set bit Bit 6: Transmit Fs-Bit Insertion Enable (TFSE). to zero (on default set to 1)
e1t1clk_wreg = 0x16, 0x00

# LIC4 Register Description:  Line Interface Control 4 (addr 0x33 + 0x10 offset = 0x43)
# RT2 bit = 0, RT1 = RT0 bit = 1 (bit EX133 in LIC3.5 = 0)
# E1: bit[5:3] = TT2...TT0 = 011bin = 120 ohm internal transmit termination
# E1: 0x1b / T1: 0x12
e1t1clk_wreg = 0x43, 0x12

# Register Name: LIC2
# Register Description: Line Interface Control 2
# Register Address: 31h
# E1: 0x00 / T1: 0x08
#e1t1clk_wreg = 0x41, 0x08
#Register Name:  LIC2 ; set TAIS / bit 4 to 1 --> transmit data normally, no AIS
e1t1clk_wreg = 0x41, 0x18

#Register Name:  LIC1 ; shall be the last one to be configured
#Register Description:  Line Interface Control 1
#Register Address:  30h plus 0x10 offset --> 0x40
# TPD (bit 0) = 1 (= normal transmitter operation
# Bit 4: Receive Equalizer Gain Limit (EGL). = 1 = -12dB (short haul)
e1t1clk_wreg = 0x40, 0x11
```

```
#
# Item <<e1t1clk_orreg>>: OR value to e1t1clk module register
#
# Syntax: e1t1clk_orreg = offs, value
#
# Params: offs: register offset
#         value: value to OR with current register value
#
# Note: An offset of 0x10 has to be added on the register address of the DS26504
#       for configuring a register in DS26504!
#


#
# Item <<e1t1clk_andreg>>: AND value with e1t1clk module register
#
# Syntax: e1t1clk_andreg = offs, value
#
# Params: offs: register offset
#         value: value to AND with current register value
#
# Note: An offset of 0x10 has to be added on the register address of the DS26504
#       for configuring a register in DS26504!
#
```

Appendix B: Module configuration for E1 functionality

This appendix shows an example of how to configure the N.A.T. Clock module and the E1-T1-CLK-Transceiver via the configuration file for E1 functionality. The configuration belongs to the following system configuration:

```
**********************************************************************
*** MCH CM/ShM Firmware V2.17d Engineering (08:27:25 May 12 2015) ***
**********************************************************************

NAT-MCH HW: PCB V3.4 Rev 120712  FPGA V1.8  AVR 1.2 - sn: 110334-7986 - Rel:120712
      AOPT: 0x3e - SRAM, E1-T1-CLK, HS Ctrl, 2nd FRT ETH, LED MOD
CLK MOD: PCB V4.1  MC V1.0  FPGA V1.2 (assembly option:  FPGA  PLL  HCSL buffer  PCIe Clock Gen) - sn: 6279 - Rel:130829
HUB MOD: PCB PCIe-x48 V2.4  MC V1.10  FPGA V1.5 (assembly option -X48 LOSC) - sn: 1357 - Rel:140411 - ChipRev: ca
```

Clock and E1-T1-CLK configuration:

```
################################################################################
###          Clock Module configuration for NAT-MCH                  ###
################################################################################

#
# === device location identifier ===
#

mch_id = 0
mez_id = 1
inst_id = 0


#
# Item <<clk_pcb_maj>>: Clock PCB major version
# Item <<clk_pcb_min>>: Clock PCB minor version
#
# Syntax: clk_pcb_maj = maj_ver
# Syntax: clk_pcb_min = min_ver
#
# Params: maj_ver: major PCB version
# Params: min_ver: minor PCB version
#

clk_pcb_maj  =  4
clk_pcb_min  =  1


#
# Item <<clk_pll_ref>>: PLL reference input configuration
#
# Syntax: clk_pll_ref = ref, freq
#
# Params: ref:  PLL reference input
#             1 -  PLL input IC1
#             2 -  PLL input IC2
#             3 -  PLL input IC3
#             4 -  PLL input IC4
#             5 -  PLL input IC5
#             6 -  PLL input IC6
#             7 -  PLL input IC7
#             8 -  PLL input IC8
#
#       freq: input clock frequency in Hz
#             Please refer to the NAT-MCH Users Manual for a list
#             of supported frequencies!
#

# PLL ref. input accepts 2.048MHz
# configure PLL input IC1 with input clock frequency 2.048MHz of RCLK
# PLL 1 with IC1 has to be used for the input clock frequency!
# Otherwise the diagnostic functionality regarding reference clock
# status (i.e. locked, unlocked, holdover) doesn't work!
clk_pll_ref  =  1,    2048000
clk_pll_ref  =  2,          0
```

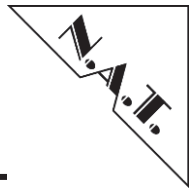
```
clk_pll_ref  =  3,        0
clk_pll_ref  =  4,        0
clk_pll_ref  =  5,        0
clk_pll_ref  =  6,        0
clk_pll_ref  =  7,        0
clk_pll_ref  =  8,        0


#
# Item <<clk_pll_mode>>: PLL mode configuration
#
# Syntax: clk_pll_mode = pll#, mode, ref
#
# Params: #:    number of PLL to configure
#           1 -  PLL #1
#           2 -  PLL #2
#
#       mode: PLL operation mode
#           1 -  automatic selection between IC1 - IC8
#           2 -  forced mode (reference according to parameter ref)
#           3 -  free running mode
#
#       ref:  PLL reference input (only for forced mode)
#           1 -  PLL input IC1
#           2 -  PLL input IC2
#           3 -  PLL input IC3
#           4 -  PLL input IC4
#           5 -  PLL input IC5
#           6 -  PLL input IC6
#           7 -  PLL input IC7
#           8 -  PLL input IC8
#

# E1-T1-CLK configure RCLK as input signal on PLL #1: PLL #1, forced mode, PLL input IC1
# Note: PLL #1 (IC1) has to be used due to diagnostic functionality regarding reference
#       clock status!
clk_pll_mode  = 1, 2, 1
clk_pll_mode  = 2, 1, 0


#
# Item <<clk_oc_cfg>>: clock output configuration
#
# Syntax: clk_oc_cfg = oc, freq
#
# Params: oc: output clock # of pll
#            1 -  PLL OC1
#            2 -  PLL OC2
#            3 -  PLL OC3
#            4 -  PLL OC4
#            5 -  PLL OC5
#            6 -  PLL OC6
#            7 -  PLL OC7
#
# Params: freq: output clock frequency (in Hz)
#

# PLL output OC1 generates 2.048MHz
clk_oc_cfg  =  1,  2048000
clk_oc_cfg  =  2,  0
clk_oc_cfg  =  3,  0
clk_oc_cfg  =  4,  0
clk_oc_cfg  =  5,  0
clk_oc_cfg  =  6,  0
clk_oc_cfg  =  7,  0



#
# Item <<clk_out>>: clock output configuration
#
# Syntax: clk_out = dst, src
#
# HCSL buffer detected
# - any source for CLK 3 output other than 0 will enable output of HCSL buffer
#
```

```
# Params: dst: destination clock identifier
#            1 -  CLK1 AMC  1
#            2 -  CLK1 AMC  2
#            3 -  CLK1 AMC  3
#            4 -  CLK1 AMC  4
#            5 -  CLK1 AMC  5
#            6 -  CLK1 AMC  6
#            7 -  CLK1 AMC  7
#            8 -  CLK1 AMC  8
#            9 -  CLK1 AMC  9
#           10 -  CLK1 AMC 10
#           11 -  CLK1 AMC 11
#           12 -  CLK1 AMC 12
#           13 -  CLK2 AMC  1
#           14 -  CLK2 AMC  2
#           15 -  CLK2 AMC  3
#           16 -  CLK2 AMC  4
#           17 -  CLK2 AMC  5
#           18 -  CLK2 AMC  6
#           19 -  CLK2 AMC  7
#           20 -  CLK2 AMC  8
#           21 -  CLK2 AMC  9
#           22 -  CLK2 AMC 10
#           23 -  CLK2 AMC 11
#           24 -  CLK2 AMC 12
#           25 -  CLK3 AMC  1
#           26 -  CLK3 AMC  2
#           27 -  CLK3 AMC  3
#           28 -  CLK3 AMC  4
#           29 -  CLK3 AMC  5
#           30 -  CLK3 AMC  6
#           31 -  CLK3 AMC  7
#           32 -  CLK3 AMC  8
#           33 -  CLK3 AMC  9
#           34 -  CLK3 AMC 10
#           35 -  CLK3 AMC 11
#           36 -  CLK3 AMC 12
#           37 -  CLK1 Update
#           38 -  CLK3 Update
#           39 -  EXT single ended 1 (INPUT  SMA 1)
#           40 -  EXT single ended 2 (OUTPUT SMA 1)
#           41 -  EXT single ended 3 (INPUT  SMA 2)
#           42 -  EXT single ended 4 (OUTPUT SMA 2)
#           43 -  EXT differential 1 (RJ45)
#           44 -  EXT differential 2 (RJ45)
#           45 -  PLL IC1
#           46 -  PLL IC2
#           47 -  PLL IC3
#           48 -  PLL IC4
#           49 -  PLL IC5
#           50 -  PLL IC6
#           51 -  PLL IC7
#           52 -  PLL IC8
#           53 -  PLL SYNC1
#           54 -  PLL SYNC2
#           55 -  PLL SYNC3
#           56 -  SYNC CLK input
#
# Params: src: source clock identifier
#            0 -  disabled
#            1 -  CLK1 AMC  1
#            2 -  CLK1 AMC  2
#            3 -  CLK1 AMC  3
#            4 -  CLK1 AMC  4
#            5 -  CLK1 AMC  5
#            6 -  CLK1 AMC  6
#            7 -  CLK1 AMC  7
#            8 -  CLK1 AMC  8
#            9 -  CLK1 AMC  9
#           10 -  CLK1 AMC 10
#           11 -  CLK1 AMC 11
#           12 -  CLK1 AMC 12
#           13 -  CLK2 AMC  1
#           14 -  CLK2 AMC  2
#           15 -  CLK2 AMC  3
#           16 -  CLK2 AMC  4
```

```
#          17 -  CLK2 AMC  5
#          18 -  CLK2 AMC  6
#          19 -  CLK2 AMC  7
#          20 -  CLK2 AMC  8
#          21 -  CLK2 AMC  9
#          22 -  CLK2 AMC 10
#          23 -  CLK2 AMC 11
#          24 -  CLK2 AMC 12
#          25 -  CLK3 AMC  1
#          26 -  CLK3 AMC  2
#          27 -  CLK3 AMC  3
#          28 -  CLK3 AMC  4
#          29 -  CLK3 AMC  5
#          30 -  CLK3 AMC  6
#          31 -  CLK3 AMC  7
#          32 -  CLK3 AMC  8
#          33 -  CLK3 AMC  9
#          34 -  CLK3 AMC 10
#          35 -  CLK3 AMC 11
#          36 -  CLK3 AMC 12
#          37 -  CLK1 Update
#          38 -  CLK3 Update
#          39 -  EXT single ended 1 (INPUT  SMA 1)
#          40 -  EXT single ended 2 (OUTPUT SMA 1)
#          41 -  EXT single ended 3 (INPUT  SMA 2)
#          42 -  EXT single ended 4 (OUTPUT SMA 2)
#          43 -  EXT differential 1 (RJ45)
#          44 -  EXT differential 2 (RJ45)
#          45 -  PLL OC1
#          46 -  PLL OC2
#          47 -  PLL OC3
#          48 -  PLL OC4
#          49 -  PLL OC5
#          50 -  PLL OC6
#          51 -  PLL OC7
#          52 -  PLL FSYNC 8kHz
#          53 -  PLL MFSYNC 2kHz
#          54 -  12.8MHz OSC
#          55 -  25MHz OSC (only with HCSL option)
#          56 -  LEVEL 0
#          57 -  LEVEL 1
#          58 -  SYNC_CLK
#

clk_out  =  1,  0
clk_out  =  2,  0
clk_out  =  3,  0
clk_out  =  4,  0
clk_out  =  5,  0
clk_out  =  6,  0
clk_out  =  7,  0
clk_out  =  8,  0
clk_out  =  9,  0
clk_out  = 10,  0
clk_out  = 11,  0
clk_out  = 12,  0
clk_out  = 13,  0
clk_out  = 14,  0
clk_out  = 15,  0
clk_out  = 16,  0
clk_out  = 17,  0
clk_out  = 18,  0
clk_out  = 19,  0
clk_out  = 20,  0
clk_out  = 21,  0
clk_out  = 22,  0
clk_out  = 23,  0
clk_out  = 24,  0
clk_out  = 25,  0
clk_out  = 26,  0
clk_out  = 27,  0
clk_out  = 28,  0
clk_out  = 29,  0
clk_out  = 30,  0
clk_out  = 31,  0
clk_out  = 32,  0
```

```
clk_out  = 33,  0
clk_out  = 34,  0
clk_out  = 35,  0
clk_out  = 36,  0
clk_out  = 37,  0
clk_out  = 38,  0
clk_out  = 39,  0
# MCH firmware drives Extref2 (reg 0x31 = value 40 here) and Extref3 (reg 0x32 = value 41 here)
# low due to over talking of TCLK. Therefore these configuration items have to be removed here
# from script (commented out via # below)!!!
#clk_out  = 40,  0
#clk_out  = 41,  0
# TCLK: destination 42 (EXT single ended 4 (OUTPUT SMA 2)) - with source clock of 45 -  PLL OC1
clk_out  = 42,  45
clk_out  = 43,  0
clk_out  = 44,  0
# destination clock 45 -  PLL IC1 with source clock of 39 -  EXT single ended 1 (INPUT  SMA 1)
# configured (EXT single ended 1 (INPUT  SMA 1) = RCLK)
clk_out  = 45,  39
clk_out  = 46,  0
clk_out  = 47,  0
clk_out  = 48,  0
clk_out  = 49,  0
clk_out  = 50,  0
clk_out  = 51,  0
clk_out  = 52,  0
clk_out  = 53,  0
clk_out  = 54,  0
clk_out  = 55,  0
clk_out  = 56,  0


#
# Item <<clk_wreg>>: write clock module register
#
# Syntax: clk_wreg = offs, value
#
# Params: offs: register offset
#         value: value to write
#


#
# Item <<clk_orreg>>: OR value to clock module register
#
# Syntax: clk_orreg = offs, value
#
# Params: offs: register offset
#         value: value to OR with current register value
#


#
# Item <<clk_andreg>>: AND value with clock module register
#
# Syntax: clk_andreg = offs, value
#
# Params: offs: register offset
#         value: value to AND with current register value
#



#
# Item <<clk_wblk>>: write block in clock module
#
# Syntax: clk_wblk = inst, len, data[0], data[1], ..., data[len - 1]
#
# Params: inst: addressed instance on clock module
#         len: number of bytes to write
#         data[n]: n-th data byte to write
#


#
# Item <<clk_wpllreg>>: write clock module PLL register
```

```
#
# Syntax: clk_wpllreg = offs, value
#
# Params: offs: PLL register offset
#         value: value to write
#


################################################################################
###          E1T1CLK Module configuration for NAT-MCH                ###
################################################################################

#
# ===  device location identifier  ===
#
# E1-T1-CLK module:
#           mch_id  = 0 = MCH1; 1 = MCH2
#           mez_id  = 0 (Base module)
#           inst_id = 1 (E1-T1-CLK)

mch_id = 0
mez_id = 0
inst_id = 1


#
# Item <<e1t1clk_wreg>>: write e1t1clk module register
#
# Syntax: e1t1clk_wreg = offs, value
#
# Params: offs: register offset
#         value: value to write
#
# Note: An offset of 0x10 has to be added on the register address of the DS26504
#       for configuring a register in DS26504!

# set DS26504 register MCREG (addr 0x08 in DS26504, addr 0x08 + 0x10 Atmel register --> 0x18)
# to E1 CRC4 FRaming Mode for receive and transmit path --> 0x66
e1t1clk_wreg = 0x18, 0x66

# TPCR1 set to 0x00 for master mode (The TCLK pin is the source of transmit clock.)
# clock module configuration will provide TCLK clock from RCLK
e1t1clk_wreg = 0x19, 0x00

# LIC4 Register Description:  Line Interface Control 4 (addr 0x33 + 0x10 offset = 0x43)
# RT2 bit = 0, RT1 = RT0 bit = 1 (bit EX133 in LIC3.5 = 0)
# bit[5:3] = TT2...TT0 = 011bin = 120 ohm internal transmit termination
e1t1clk_wreg = 0x43, 0x1b

#Register Name:  LIC2 ; set TAIS / bit 4 to 1 --> transmit data normally, no AIS
#Register Description:  Line Interface Control 2
#Register Address:  31h plus 0x10 offset --> 0x41
e1t1clk_wreg = 0x41, 0x10

#Register Name:  LIC1 ; shall be the last one to be configured
#Register Description:  Line Interface Control 1
#Register Address:  30h plus 0x10 offset --> 0x40
# TPD (bit 0) = 1 (= normal transmitter operation
# L2 (bit 7) to L0 (bit 5) = 001 = 120 Ohm
# Bit 4: Receive Equalizer Gain Limit (EGL). = 1 = -12dB (short haul)
e1t1clk_wreg = 0x40, 0x31

#
# Item <<e1t1clk_orreg>>: OR value to e1t1clk module register
#
# Syntax: e1t1clk_orreg = offs, value
#
# Params: offs: register offset
#         value: value to OR with current register value
#
# Note: An offset of 0x10 has to be added on the register address of the DS26504
#       for configuring a register in DS26504!
#
```



```
#
# Item <<e1t1clk_andreg>>: AND value with e1t1clk module register
#
# Syntax: e1t1clk_andreg = offs, value
#
# Params: offs: register offset
#         value: value to AND with current register value
#
# Note: An offset of 0x10 has to be added on the register address of the DS26504
#       for configuring a register in DS26504!
#
```

Appendix C: Reference Documentation

[1] N.A.T.-MCH – User's Manual
[2] N.A.T.-MCH HUB-Module PCIe – Technical Reference Manual
[3] N.A.T.-MCH CLK-Module – Technical Reference Manual
[4] Atmel, Atmel-8153J–AVR-ATxmega8E5-ATxmega16E5-ATxmega32E5_Datasheet
[5] DS26504, DS26504 T1/E1/J1/64KCC BITS Element (DALLAS SEMICONDUCTOR MAXIM)

Document's History

Revision	Date	Description	Author
1.0	23.03.2015 27.03.2015	initial release added chapter E1-T1-CLK-Transceiver Diagnostic	hn hn
1.1	12.05.2015	- added new chapters 5.4.4 Get reference clock status of E1-T1-CLK-Transceiver and 5.5.2 CLI Configuration Command Interface - introduced new CLI interface command <cfg_ctrl_cmd> (chapter 5.5.2 CLI Configuration Command Interface) - adapted chapters Appendix A: Module configuration for T1 functionality and Appendix B: Module configuration for E1 functionality with E1-T1-CLK device locator identifier and generic clock module commands	hn
1.2	12.08.2015 21.08.2015	- updated chapters Appendix A: Module configuration for T1 functionality and Appendix B: Module configuration for E1 functionality with setting of bit TAIS / bit 4 in register LIC2 to 1 --> transmit data normally, no AIS - added syntax of command e1t1clk_getstatus in 5.4.4 Get reference clock status of E1-T1-CLK-Transceiver - fixed layout of Table 4: J1: MCH Connector – Pin Assignment - updated Table 3: N.A.T. MCH E1-T1-CLK - LED Status	hn
1.3	21.08.2015	Added chapter 5.5.3 CLI Updating Firmware of E1-T1-CLK module	hn