TE0741 Test Board

Overview

Table of contents

Refer to http://trenz.org/te0741-info for the current online version of this manual and other available documentation.

This example shows how to reconfigure SI5338 with the MicroBlaze_MCS and monitor the CLKs. An additional MicroBlaze is added for running the Hello_TE0741 standalone application.

Key Features

- Vitis/Vivado 2023.2
- MicroBlaze
- I2C
- UART
- Flash
- Fmeter
- JTAG to AXI Master
- SI5338 initialisation with MCS

Revision History

Date	Vivado	Project Built	Authors	Description
2023-12-18	2023.2	TE0741-test_board_noprebuilt-vivado_2023.2-build_3_20231218145348.zip TE0741-test_board-vivado_2023.2-build_3_20231218145348.zip	Waldemar Hanemann	• update 2023.2
2022-06-09	2021.2	TE0741-test_board_noprebuilt-vivado_2021.2-build_14_20220609134340.zip TE0741-test_board-vivado_2021.2-build_14_20220609134340.zip	Waldemar Hanemann	• pll_in4 i2c address pin pulled to GND in System controller ip • added new variants
2022-03-15	2021.2	TE0741-test_board_noprebuilt-vivado_2021.2-build_11_20220315112318.zip TE0741-test_board-vivado_2021.2-build_11_20220315112318.zip	Waldemar Hanemann	• update 2021.2
2018-04-16	2017.4	TE0741-test_board-vivado_2017.4-build_07_20180416142156.zip TE0741-test_board_noprebuilt-vivado_2017.4-build_07_20180416142217.zip	John Hartfiel	• initial release

Design Revision History

Release Notes and Know Issues

Issues	Description	Workaround	To be fixed version
No known issues	---	---	---

Known Issues

Requirements

Software

Software	Version	Note
Vitis	2023.2	needed, Vivado is included into Vitis installation
SI ClockBuilder Pro	---	optional

Software

Hardware

Basic description of TE Board Part Files is available on TE Board Part Files.

Complete List is available on "<project folder>\board_files*_board_files.csv"

Design supports following modules:

Module Model	Board Part Short Name	PCB Revision Support	DDR	QSPI Flash	Others	Notes
TE0741-03-070-2IF	070_2if	REV02, REV03	---	32MB	MGT LR: 6,6 Gb/s	
TE0741-03-160-2IF	160_2if	REV02, REV03	---	32MB	MGT LR: 6,6 Gb/s	
TE0741-03-325-2IF	325_2if	REV02, REV03	---	32MB	MGT LR: 6,6 Gb/s	
TE0741-03-410-2IF	410_2if	REV02, REV03	---	32MB	MGT LR: 6,6 Gb/s	
TE0741-03-070-2CF	070_2cf	REV02, REV03	---	32MB	MGT LR: 6,6 Gb/s	
TE0741-03-160-2CF	160_2cf	REV02, REV03	---	32MB	MGT LR: 6,6 Gb/s	
TE0741-03-325-2CF*	325_2cf	REV02, REV03	---	32MB	MGT LR: 6,6 Gb/s	
TE0741-03-410-2CF	410_2cf	REV02, REV03	---	32MB	MGT LR: 6,6 Gb/s	
TE0741-03-160-2C1	160_2c1	REV02, REV03	---	32MB	MGT LR: 10,3125 Gb/s	
TE0741-02-B3E-1-AF	160_3e1	REV02, REV03	---	32MB	MGT LR: 10,3125 Gb/s	
TE0741-04-A2I-1-A	070_2if	REV04	---	32MB	MGT LR: 6,6 Gb/s	
TE0741-04-G2C-1-A	410_2cf	REV04	---	32MB	MGT LR: 6,6 Gb/s	
TE0741-03-S002	410_2cf	REV03	---	32MB	MGT LR: 6,6 Gb/s	
TE0741-04-S001	410_2cf	REV04	---	32MB	MGT LR: 6,6 Gb/s	
TE0741-05-A2I-1-A	070_2if	REV05	---	32MB	MGT LR: 6,6 Gb/s	

TE0741-05-B2C-1-A	160_2cf	REV05	---	32MB	MGT LR: 6,6 Gb/s	
TE0741-05-B2I-1-A	160_2if	REV05	---	32MB	MGT LR: 6,6 Gb/s	
TE0741-05-D2C-1-A	325_2cf	REV05	---	32MB	MGT LR: 6,6 Gb/s	
TE0741-05-D2I-1-A	325_2if	REV05	---	32MB	MGT LR: 6,6 Gb/s	
TE0741-05-G2C-1-A	410_2cf	REV05	---	32MB	MGT LR: 6,6 Gb/s	

*used as reference

Hardware Modules

Design supports following carriers:

Carrier Model	Notes
TE0701	
TE0703*	
TE0705	
TE0706	

*used as reference

Hardware Carrier

Additional HW Requirements:

Additional Hardware	Notes
USB Cable for JTAG/UART	Check Carrier Board and Programmer for correct type
XMOD Programmer	Carrier Board dependent, only if carrier has no own FTDI

*used as reference

Additional Hardware

Content

For general structure and usage of the reference design, see Project Delivery - AMD devices

Design Sources

Type	Location	Notes

Vivado	<project folder>\block_design <project folder>\constraints <project folder>\ip_lib <project folder>\board_files	Vivado Project will be generated by TE Scripts
Vitis	<project folder>\sw_lib	Additional Software Template for Vitis and apps_list.csv with settings automatically for Vitis app generation

Design sources

Additional Sources

Type	Location	Notes
SI5338	<project folder>\misc\Si5338	SI5338 Project with current PLL Configuration

Additional design sources

Prebuilt

File	File-Extension	Description
BIT-File	*.bit	FPGA (PL Part) Configuration File
DebugProbes-File	*.ltx	Definition File for Vivado/Vivado Labtools Debugging Interface
Diverse Reports	---	Report files in different formats
Xilinx Support Archive	*.xsa	Exported Vivado Hardware Specification for Vitis and PetaLinux
LabTools Project-File	*.lpr	Vivado Labtools Project File
MCS-File	*.mcs	Flash Configuration File with Boot-Image (MicroBlaze or FPGA part only)
MMI-File	*.mmi	File with BRAM-Location to generate MCS or BIT-File with *.elf content (MicroBlaze only)
Software-Application-File	*.elf	Software Application for Zynq or MicroBlaze Processor Systems

Prebuilt files (only on ZIP with prebuilt content)

Download

Reference Design is only usable with the specified Vivado/Vitis/PetaLinux version. Do never use different Versions of AMD Software for the same Project.

Reference Design is available on:

- TE0741 "Test Board" Reference Design

Design Flow

⚠ Reference Design is available with and without prebuilt files. It's recommended to use TE prebuilt files for first launch.

Trenz Electronic provides a tcl based built environment based on AMD Design Flow.

See also:

- AMD Development Tools
- Vivado Projects - TE Reference Design
- Project Delivery - AMD Devices.

The Trenz Electronic FPGA Reference Designs are TCL-script based project. Command files for execution will be generated with "_create_win_setup.cmd" on Windows OS and "_create_linux_setup.sh" on Linux OS.

TE Scripts are only needed to generate the vivado project, all other additional steps are optional and can also executed by AMD Vivado/Vitis GUI. For currently Scripts limitations on Win and Linux OS see: Project Delivery Currently limitations of functionality

⚠ **Caution!** Win OS has a 260 character limit for path lengths which can affect the Vivado tools. To avoid this issue, use Virtual Drive or the shortest possible names and directory locations for the reference design (for example "x:\<project folder>")

1. Run _create_win_setup.cmd/_create_linux_setup.sh and follow instructions on shell:

_create_win_setup.cmd/_create_linux_setup.sh

```
------------------------Set design paths---------------------------
-- Run Design with: _create_win_setup
-- Use Design Path: <absolute project path>
-------------------------------------------------------------------
------------------------TE Reference
Design--------------------------
-------------------------------------------------------------------
-- (0)  Module selection guide, project creation...prebuilt export...
-- (1)  Create minimum setup of CMD-Files and exit Batch
-- (2)  Create maximum setup of CMD-Files and exit Batch
-- (3)  (internal only) Dev
-- (4)  (internal only) Prod
-- (c)  Go to CMD-File Generation (Manual setup)
-- (d)  Go to Documentation (Web Documentation)
-- (g)  Install Board Files from Xilinx Board Store (beta)
-- (a)  Start design with unsupported Vivado Version (beta)
-- (x)  Exit Batch (nothing is done!)
----
Select (ex.:'0' for module selection guide):
```

2. Press 0 and enter to start "Module Selection Guide"
3. Create project and follow instructions of the product selection guide, settings file will be configured automatically during this process.
    - optional for manual changes: Select correct device and Xilinx install path on "design_basic_settings.cmd" and create Vivado project with "vivado_create_project_guimode.cmd"

⚠ Note: Select correct one, see also Vivado Board Part Flow

4. Create hardware description file (.xsa file) and export to prebuilt folder

   run on Vivado TCL (Script generates design and export files into "<project folder>\prebuilt\hardware\<short name>")

   ```
   TE::hw_build_design -export_prebuilt
   ```

   Using Vivado GUI is the same, except file export to prebuilt folder.

5. Generate Programming Files with Vitis
   a. Run on Vivado TCL:

      Script generates applications and bootable files, which are defined in "sw_lib\apps_list.csv"

      ```
      TE::sw_run_vitis -all
      ```

   b. Copy "\prebuilt\software\<short name>\hello_te0741.elf" into "\firmware\microblaze_0\"
   c. (optional) Copy "\prebuilt\software\<short name>\scu.elf" into "\firmware\microblaze_mcs_0\"
   d. Regenerate Vivado Project or Update Bitfile only, with new "hello_te0741.elf" and "scu.elf"

      TCL scripts generate also platform project, this must be done manually in case GUI is used. See Vitis

Launch

Programming

Check Module and Carrier TRMs for proper HW configuration before you try any design.

Reference Design is also available with prebuilt files. It's recommended to use TE prebuilt files for first launch.

AMD documentation for programming and debugging: Vivado/Vitis/SDSoC-Xilinx Software Programming and Debugging

Get prebuilt boot binaries

1. Run _create_win_setup.cmd/_create_linux_setup.sh and follow instructions on shell
2. Press 0 and enter to start "Module Selection Guide"
   a. Select assembly version
   b. Validate selection
   c. Select create and open delivery binary folder

      Note: Folder "<project folder>_binaries_<Article Name>" with subfolder "boot_<app name>" for different applications will be generated

QSPI

1. Connect JTAG and power on carrier with module
2. Open Vivado Project with "vivado_open_existing_project_guimode.cmd" or if not created, create with "vivado_create_project_guimode.cmd"

run on Vivado TCL (Script programs BOOT.bin on QSPI flash)
`TE::pr_program_flash -swapp hello_te0741`

⚠ To program with Vitis/Vivado GUI, use special FSBL (fsbl_flash) on setup

SD-Boot mode

Not used on this Example.

JTAG

1. Connect JTAG and power on PCB
2. Open Vivado HW Manager
3. Program FPGA with Bitfile from "prebuilt\hardware\<short dir>\"

Usage

1. Prepare HW like described on section Programming
2. Connect UART USB (most cases same as JTAG)
3. Select QSPI as Boot Mode

ⓘ Note: See TRM of the Carrier, which is used.

4. Power On PCB

1. FPGA Loads Bitfile from Flash

2. MCS Firmware configure SI5338 and starts Microblaze

3. Hello TE0741 from Bitfile Example will be run on UART console.

info: Do not reboot, if Bitfile programming over JTAG is used as programming method.

UART

Open Serial Console (e.g. putty)

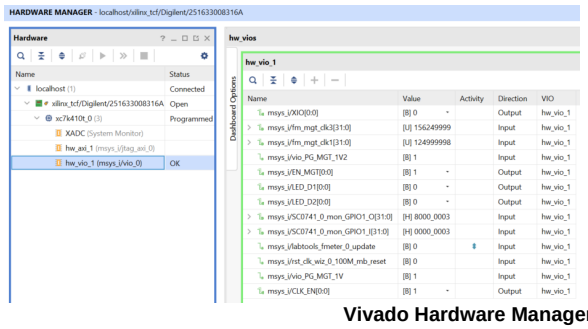
a. Speed: 9600
b. COM Port: Win OS, see device manager, Linux OS see dmesg |grep tty (UART is *USB1)

```
Hello Trenz Module TE0741 (Loop: 2)
Hello Trenz Module TE0741 (Loop: 3)
MGT_CLOCK_3:  156.250 MHz
MGT_CLOCK_1:  125.  0 MHz
Hello Trenz Module TE0741 (Loop: 4)
Hello Trenz Module TE0741 (Loop: 5)
Hello Trenz Module TE0741 (Loop: 6)
MGT_CLOCK_3:  156.250 MHz
MGT_CLOCK_1:  125.  0 MHz
```

Vivado HW Manager

Open Vivado HW-Manager and add VIO signal to dashboard (*.ltx located on prebuilt folder)

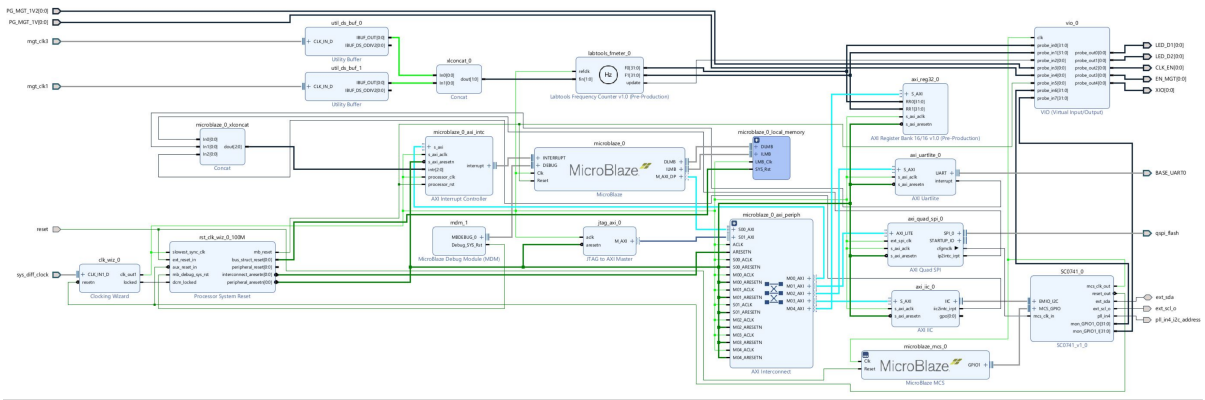
- Control:
  - LED_D1/D2 control. Control LED D4 with XIO.
  - SI5338 25MHz REF CLK Enable
  - MGT Enable
- Monitoring:
  - Set radix from VIO signals (MGT...) to unsigned integer.
  - Note: Frequency Counter is inaccurate and displayed unit is Hz
  - MGT REFCLK1~125MHz, GT_REFCLK3~156,25MHz (default off, configured with MCS Firmware)
  - MGT Power Monitoring



Vivado Hardware Manager

System Design - Vivado

Block Design



Block Design

Constraints

Basic module constraints

_i_bitgen_common.xdc

```
set_property BITSTREAM.GENERAL.COMPRESS TRUE [current_design]
set_property BITSTREAM.CONFIG.CONFIGRATE 66 [current_design]
set_property CONFIG_VOLTAGE 3.3 [current_design]
set_property CFGBVS VCCO [current_design]
set_property CONFIG_MODE SPIx4 [current_design]
set_property BITSTREAM.CONFIG.SPI_32BIT_ADDR YES [current_design]
set_property BITSTREAM.CONFIG.SPI_BUSWIDTH 4 [current_design]
set_property BITSTREAM.CONFIG.M1PIN PULLNONE [current_design]
set_property BITSTREAM.CONFIG.M2PIN PULLNONE [current_design]
set_property BITSTREAM.CONFIG.M0PIN PULLNONE [current_design]

set_property BITSTREAM.CONFIG.USR_ACCESS TIMESTAMP [current_design
```

Design specific constraints

_i_io.xdc

```
#LED
set_property PACKAGE_PIN D26 [get_ports {LED_D1[0]}]
set_property IOSTANDARD LVCMOS33 [get_ports {LED_D1[0]}]
set_property PACKAGE_PIN E26 [get_ports {LED_D2[0]}]
set_property IOSTANDARD LVCMOS33 [get_ports {LED_D2[0]}]
#MGT Power
set_property PACKAGE_PIN G25 [get_ports {PG_MGT_1V2[0]}]
set_property IOSTANDARD LVCMOS33 [get_ports {PG_MGT_1V2[0]}]
set_property PACKAGE_PIN K23 [get_ports {PG_MGT_1V[0]}]
set_property IOSTANDARD LVCMOS33 [get_ports {PG_MGT_1V[0]}]
set_property PACKAGE_PIN H22 [get_ports {EN_MGT[0]}]
set_property IOSTANDARD LVCMOS33 [get_ports {EN_MGT[0]}]
#SI5338 CLK
set_property PACKAGE_PIN C26 [get_ports {CLK_EN[0]}]
set_property IOSTANDARD LVCMOS33 [get_ports {CLK_EN[0]}]
#I2C PLL SI5338
set_property PACKAGE_PIN A20 [get_ports ext_scl_o]
set_property IOSTANDARD LVCMOS33 [get_ports ext_scl_o]
set_property PACKAGE_PIN B21 [get_ports ext_sda]
set_property IOSTANDARD LVCMOS33 [get_ports ext_sda]
set_property PACKAGE_PIN B20 [get_ports pll_in4_i2c_address]
set_property IOSTANDARD LVCMOS33 [get_ports pll_in4_i2c_address]

#OneWire
set_property IOSTANDARD LVCMOS33 [get_ports XIO]
set_property PACKAGE_PIN H26 [get_ports XIO]


set_property IOSTANDARD LVCMOS33 [get_ports reset]
set_property PACKAGE_PIN L23 [get_ports reset]
set_property PACKAGE_PIN C23 [get_ports qspi_flash_ss_io]
set_property IOSTANDARD LVCMOS33 [get_ports qspi_flash_ss_io]
set_property PACKAGE_PIN B24 [get_ports qspi_flash_io0_io]
set_property IOSTANDARD LVCMOS33 [get_ports qspi_flash_io0_io]
set_property PACKAGE_PIN A25 [get_ports qspi_flash_io1_io]
set_property IOSTANDARD LVCMOS33 [get_ports qspi_flash_io1_io]
set_property PACKAGE_PIN B22 [get_ports qspi_flash_io2_io]
set_property PACKAGE_PIN A22 [get_ports qspi_flash_io3_io]
set_property IOSTANDARD LVCMOS33 [get_ports qspi_flash_io2_io]
set_property IOSTANDARD LVCMOS33 [get_ports qspi_flash_io3_io]
#set_property C_CLK_INPUT_FREQ_HZ 300000000 [get_debug_cores dbg_hub]
#set_property C_ENABLE_CLK_DIVIDER false [get_debug_cores dbg_hub]
#set_property C_USER_SCAN_CHAIN 1 [get_debug_cores dbg_hub]
#connect_debug_port dbg_hub/clk [get_nets clk]
```

_i_timing.xdc

```
#Fmeter can be ignored, it's only simple measurement

set_false_path -from [get_pins {msys_i/labtools_fmeter_0/U0/FMETER_gen[*].
COUNTER_F_inst/bl.DSP48E_2/CLK}] -to [get_pins {msys_i/labtools_fmeter_0/U0
/F_reg[*]/D}]

set_false_path -from [get_pins msys_i/labtools_fmeter_0/U0/toggle_reg/C] -
to [get_pins {msys_i/labtools_fmeter_0/U0/FMETER_gen[*].COUNTER_F_inst/bl.
DSP48E_2/RSTC}]
set_false_path -from [get_pins msys_i/labtools_fmeter_0/U0/toggle_reg/C] -
to [get_pins {msys_i/labtools_fmeter_0/U0/FMETER_gen[*].COUNTER_F_inst/bl.
DSP48E_2/RSTA}]
set_false_path -from [get_pins msys_i/labtools_fmeter_0/U0/toggle_reg/C] -
to [get_pins {msys_i/labtools_fmeter_0/U0/FMETER_gen[*].COUNTER_F_inst/bl.
DSP48E_2/RSTB}]
set_false_path -from [get_pins msys_i/labtools_fmeter_0/U0/toggle_reg/C] -
to [get_pins {msys_i/labtools_fmeter_0/U0/FMETER_gen[*].COUNTER_F_inst/bl.
DSP48E_2/CEALUMODE}]
set_false_path -from [get_pins msys_i/labtools_fmeter_0/U0/toggle_reg/C] -
to [get_pins {msys_i/labtools_fmeter_0/U0/FMETER_gen[*].COUNTER_F_inst/bl.
DSP48E_2/RSTCTRL}]


#set_false_path -from [get_clocks -of_objects [get_pins msys_i/clk_wiz_0
/inst/mmcm_adv_inst/clk_out1]] -to [get_clocks {msys_i/util_ds_buf_0/U0
/IBUF_OUT[0]}]
#set_false_path -from [get_clocks -of_objects [get_pins msys_i/clk_wiz_0
/inst/mmcm_adv_inst/clk_out1]] -to [get_clocks {msys_i/util_ds_buf_1/U0
/IBUF_OUT[0]}]

set_false_path -from [get_clocks -of_objects [get_pins msys_i/clk_wiz_0
/inst/mmcm_adv_inst/CLKOUT0]] -to [get_clocks mgt_clk1_clk_p]
set_false_path -from [get_clocks -of_objects [get_pins msys_i/clk_wiz_0
/inst/mmcm_adv_inst/CLKOUT0]] -to [get_clocks mgt_clk3_clk_p
```

Software Design - Vitis

For Vitis project creation, follow instructions from:

Vitis

Application

Template location: "<project folder>\sw_lib\sw_apps\"

SCU

MCS Firmware to configure SI5338 and Reset System.

Template location: \sw_lib\sw_apps\scu

Hello TE0741

Xilinx Hello World example as endless loop with reading Clock registers from fmeter IP

Template location: \sw_lib\sw_apps\hello_te0741

Additional Software

SI5338

File location "<project folder>\misc\Si5338\Si5338-*.slabtimeproj"

General documentation how you work with this project will be available on Si5338

App. A: Change History and Legal Notices

Document Change History

To get content of older revision go to "Change History" of this page and select older document revision number.

Date	Document Revision	Authors	Description
Error rendering macro 'page-info' Ambiguous method overloading for method jdk.proxy279.$Proxy4022#hasContentLevelPermission. Cannot resolve which method to	**Error rendering macro 'page-info'** Ambiguous method overloading for method jdk.proxy279.$Proxy4022#hasContentLevelPermission. Cannot resolve which method to	**Error rendering macro 'page-info'** Ambiguous method overloading for method jdk.proxy279.$Proxy4022#hasContentLevelPermission. Cannot resolve which method to	• update 2023.2

invoke for [null, class java.lang. String, class com. atlassian. confluence. pages. Page] due to overlapping prototypes between: [interface com. atlassian. confluence. user. Confluence User, class java.lang. String, class com. atlassian. confluence. core. ContentEntit yObject] [interface com. atlassian. user.User, class java. lang.String, class com. atlassian. confluence. core. ContentEntit

invoke for [null, class java.lang. String, class com. atlassian. confluence. pages. Page] due to overlapping prototypes between: [interface com. atlassian. confluence. user. Confluence User, class java.lang. String, class com. atlassian. confluence. core. ContentEntit yObject] [interface com. atlassian. user.User, class java. lang.String, class com. atlassian. confluence. core. ContentEntit

invoke for [null, class java.lang. String, class com. atlassian. confluence. pages. Page] due to overlapping prototypes between: [interface com. atlassian. confluence. user. Confluence User, class java.lang. String, class com. atlassian. confluence. core. ContentEntit yObject] [interface com. atlassian. user.User, class java. lang.String, class com. atlassian. confluence. core. ContentEntit

yObject]	yObject]	yObject]	
2022-06-09	v.9	Waldemar Hanemann	• pll_in4 - i2c address pin pulled to zero in System Controller ip • added new variants
2022-03-15	v.8	Waldemar Hanemann	• update 2021.2
2018-05-15	v.7	John Hartfiel	• Release 2017.4 • small description update
2018-04-16	v.1	John Hartfiel	• initial release
--	all	**Error rendering macro 'page-info'** Ambiguous method overloading for method jdk.proxy279.$Proxy4022#hasContentLevelPermission. Cannot resolve which method to	--

invoke for [null, class java.lang.String, class com.atlassian.confluence.pages.Page] due to overlapping prototypes between: [interface com.atlassian.confluence.user.ConfluenceUser, class java.lang.String, class com.atlassian.confluence.core.ContentEntityObject] [interface com.atlassian.user.User, class java.lang.String, class com.atlassian.confluence.core.ContentEntit

		yObject]	

Document change history.

Legal Notices

Data Privacy

Please also note our data protection declaration at https://www.trenz-electronic.de/en/Data-protection-Privacy

Document Warranty

The material contained in this document is provided “as is” and is subject to being changed at any time without notice. Trenz Electronic does not warrant the accuracy and completeness of the materials in this document. Further, to the maximum extent permitted by applicable law, Trenz Electronic disclaims all warranties, either express or implied, with regard to this document and any information contained herein, including but not limited to the implied warranties of merchantability, fitness for a particular purpose or non infringement of intellectual property. Trenz Electronic shall not be liable for errors or for incidental or consequential damages in connection with the furnishing, use, or performance of this document or of any information contained herein.

Limitation of Liability

In no event will Trenz Electronic, its suppliers, or other third parties mentioned in this document be liable for any damages whatsoever (including, without limitation, those resulting from lost profits, lost data or business interruption) arising out of the use, inability to use, or the results of use of this document, any documents linked to this document, or the materials or information contained at any or all such documents. If your use of the materials or information from this document results in the need for servicing, repair or correction of equipment or data, you assume all costs thereof.

Copyright Notice

No part of this manual may be reproduced in any form or by any means (including electronic storage and retrieval or translation into a foreign language) without prior agreement and written consent from Trenz Electronic.

Technology Licenses

The hardware / firmware / software described in this document are furnished under a license and may be used /modified / copied only in accordance with the terms of such license.

Environmental Protection

To confront directly with the responsibility toward the environment, the global community and eventually also oneself. Such a resolution should be integral part not only of everybody's life. Also enterprises shall be conscious of their social responsibility and contribute to the preservation of our common living space. That is why Trenz Electronic invests in the protection of our Environment.

REACH, RoHS and WEEE

REACH

Trenz Electronic is a manufacturer and a distributor of electronic products. It is therefore a so called downstream user in the sense of REACH. The products we supply to you are solely non-chemical products (goods). Moreover and under normal and reasonably foreseeable circumstances of application, the goods supplied to you shall not release any substance. For that, Trenz Electronic is obliged to neither register nor to provide safety data sheet. According to present knowledge and to best of our knowledge, no SVHC (Substances of Very High Concern) on the Candidate List are contained in our products. Furthermore, we will immediately and unsolicited inform our customers in compliance with REACH - Article 33 if any substance present in our goods (above a concentration of 0,1 % weight by weight) will be classified as SVHC by the European Chemicals Agency (ECHA).

RoHS

Trenz Electronic GmbH herewith declares that all its products are developed, manufactured and distributed RoHS compliant.

WEEE

Information for users within the European Union in accordance with Directive 2002/96/EC of the European Parliament and of the Council of 27 January 2003 on waste electrical and electronic equipment (WEEE).

Users of electrical and electronic equipment in private households are required not to dispose of waste electrical and electronic equipment as unsorted municipal waste and to collect such waste electrical and electronic equipment separately. By the 13 August 2005, Member States shall have ensured that systems are set up allowing final holders and distributors to return waste electrical and electronic equipment at least free of charge. Member States shall ensure the availability and accessibility of the necessary collection facilities. Separate collection is the precondition to ensure specific treatment and recycling of waste electrical and electronic equipment and is necessary to achieve the chosen level of protection of human health and the environment in the European Union. Consumers have to actively contribute to the success of such collection and the return of waste electrical and electronic equipment. Presence of hazardous substances in electrical and electronic equipment results in potential effects on the environment and human health. The symbol consisting of the crossed-out wheeled bin indicates separate collection for waste electrical and electronic equipment.

Trenz Electronic is registered under WEEE-Reg.-Nr. DE97922676.

Error rendering macro 'page-info'

Ambiguous method overloading for method jdk.proxy279.$Proxy4022#hasContentLevelPermission. Cannot resolve which method to invoke for [null, class java.lang.String, class com.atlassian.confluence.pages.Page] due to overlapping prototypes between: [interface com.atlassian.confluence.user.ConfluenceUser, class java.lang.String, class com.atlassian.confluence.core.ContentEntityObject] [interface com.atlassian.user.User, class java.lang.String, class com.atlassian.confluence.core.ContentEntityObject]