TE0710 Test Board

Table of contents

Overview

Microblaze Design with linux example.

Refer to http://trenz.org/te0710-info for the current online version of this manual and other available documentation.

For directly getting started with the prebuilt files jump to the section Launch.

Key Features

- Vitis/Vivado 2023.2
- PetaLinux
- MicroBlaze
- SPI ELF Bootloader
- Flash
- MIG
- ETH(ETH1 and ETH2)
- LED
- EEPROM MAC
- I2C Interface to CPLD
- JTAG to AXI Master

Revision History

Date	Vivado	Project Built	Authors	Description
2024-02-19	2023.2	TE0710-test_board_noprebuilt-vivado_2023.2-build_4_20240219100859.zip TE0710-test_board-vivado_2023.2-build_4_20240219100859.zip	Waldemar Hanemann	• 2023.2 update
2022-02-16	2021.2	TE0710-test_board_noprebuilt-vivado_2021.2-build_11_20220216112910.zip TE0710-test_board-vivado_2021.2-build_11_20220216112910.zip	Waldemar Hanemann	• new spi bootloader by Henrik Brix Andersen • adjusted offsets

2022-02-04	2021.2	TE0710-test_board-vivado_2021.2-build_11_20220208153036.zip TE0710-test_board_noprebuilt-vivado_2021.2-build_11_20220208153036.zip	Waldemar Hanemann	• 2021.2 update • document style update • added boot script • added eeprom interface for MAC address read-out • added simple sd card interface • added 2nd Ethernet Interface
2020-04-21	2019.2	TE0710-test_board-vivado_2019.2-build_10_20200421063949.zip TE0710-test_board_noprebuilt-vivado_2019.2-build_10_20200421064005.zip	John Hartfiel	• 2019.2 update
2018-03-29	2017.4	te0710-test_board-vivado_2017.4-build_07_20180329130739.zip te0710-test_board_noprebuilt-vivado_2017.4-build_07_20180329130757.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
PetaLinux	2023.2	needed

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	EMMC	Others	Notes
TE0710-02-35-2CF	35_2cf_512mb	REV02	512MB	32MB	NA	NA	NA
TE0710-02-35-2IF	35_2if_512mb	REV02	512MB	32MB	NA	NA	NA
TE0710-02-100-2CF	100_2cf_512mb	REV02	512MB	32MB	NA	NA	NA
TE0710-02-100-2IF	100_2if_512mb	REV02	512MB	32MB	NA	NA	NA
TE0710-02-72I21-A	100_2if_512mb	REV02	512MB	32MB	NA	NA	NA
TE0710-02-S001	100_2cf_512mb	REV02	512MB	32MB	NA	NA	no ETH-PHY
TE0710-02-42I21-A	35_2if_512mb	REV02	512MB	32MB	NA	NA	NA
TE0710-02-S003	35_2if_512mb	REV02	512MB	32MB	NA	NA	NA
TE0710-03-42C21-A	35_2cf_512mb	REV03	512MB	32MB	NA	NA	NA
TE0710-03-42I21-A	35_2if_512mb	REV03	512MB	32MB	NA	NA	NA
TE0710-03-72C21-A*	100_2cf_512mb	REV03	512MB	32MB	NA	NA	NA
TE0710-03-72I21-A	100_2if_512mb	REV03	512MB	32MB	NA	NA	NA

*used as reference

Hardware Modules

Design supports following carriers:

Carrier Model	Notes
TE0701	
TE0703*	used as reference carrier
TE0705	
TE0706	
TEBA0841	

*used as reference

Hardware Carrier

Additional HW Requirements:

Additional Hardware	Notes

USB Cable for JTAG/UART	Check Carrier Board and Programmer for correct typ
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
PetaLinux	<project folder>\os\petalinux	PetaLinux template with current configuration

Design sources

Additional Sources

Type	Location	Notes

Additional design sources

Prebuilt

File	File-Extension	Description
BIT-File	*.bit	FPGA (PL Part) Configuration File
Boot Script-File	*.scr	Distro Boot Script file
DebugProbes-File	*.ltx	Definition File for Vivado/Vivado Labtools Debugging Interface
Diverse Reports	---	Report files in different formats
Hardware-Platform-Specification-Files	*.xsa	Exported Vivado Hardware Specification for Vitis and PetaLinux
LabTools Project-File	*.lpr	Vivado Labtools Project File

OS-Image	*.ub	Image with Linux Kernel (On Petalinux optional with Devicetree and RAM-Disk)
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:

- TE0710 "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:

- Xilinx Development Tools#XilinxSoftware-BasicUserGuides
- Vivado Projects - TE Reference Design
- Project Delivery.

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 AMD 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) for PetaLinux project 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. Create and configure your PetaLinux project with exported .xsa-file, see PetaLinux KICKstart
    - use TE Template from "<project folder>\os\petalinux"
    - use exported .xsa file from "<project folder>\prebuilt\hardware\<short name>" . **Note:** HW Export from Vivado GUI creates another path as default workspace.
    - **Important Note:** Select correct Flash partition offset on petalinux-config: Subsystem Auto HW Settings Flash Settings, FPGA+Boot+bootenv=0xA00000 (increase automatically generate Boot partition), increase image size to A:, see Config
    - The build images are located in the "<plnx-proj-root>/images/linux" directory
6. Configure the **boot.scr** file as needed, see Distro Boot with Boot.scr
7. Copy PetaLinux build image files to prebuilt folder
    - copy **u-boot.elf** and **image.ub** from "<plnx-proj-root>/images/linux" to prebuilt folder

"<project folder>\prebuilt\os\petalinux\<ddr size>" or "<project folder>\prebuilt\os\petalinux\<short name>"

8. Generate Programming Files with Vitis

run on Vivado TCL (Script generates applications and bootable files, which are defined in "test_board\sw_lib\apps_list.csv")

```
TE::sw_run_vitis -all
TE::sw_run_vitis (optional; Start Vitis from Vivado GUI or start with TE Scripts on Vivado TCL)
```

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

9. (Optional) BlockRam Firmware Update
    a. Copy "<project folder>\prebuilt\software\<short name>\spi_bootloader.elf" into "<project folder>\firmware\microblaze_0\"
    b. Regenerate Vivado Project or Update Bitfile only with "spi_bootloader.elf"

```
TE::hw_build_design -export_prebuilt
TE::sw_run_vitis -all
```

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-Boot mode

Option for **u-boot.mcs** on QSPI Flash.
(u-boot.mcs contains all files necessary to boot up linux)

1. Connect the **USB cable**(JTAG) and **power supply** 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 u-boot`

3. Reboot (if not done automatically)

SD-Boot mode

Not used on this Example.

JTAG

Not used on this example.

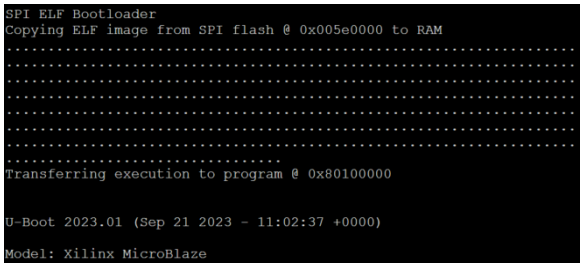
Usage

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

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

> Starting with Petalinux version 2020.1, the industry standard "Distro-Boot" boot flow for U-Boot was introduced, which significantly expands the possibilities of the boot process and has the primary goal of making booting much more standardised and predictable.
> The boot options described above describe the common boot processes for this hardware; other boot options are possible.
> For more information see Distro Boot with Boot.scr

4. Power On PCB
    1. FPGA Loads Bitfile from Flash,
    2. SPI Bootloader from Bitfile Firmware loads U-Boot into DDR (This takes a while)



3. U-boot loads Linux from QSPI Flash into DDR

Linux

1. Open Serial Console (e.g. putty)
    - Speed: 9600
    - select COM Port

    > Win OS, see device manager, Linux OS see dmesg |grep tty (UART is *USB1)

2. Linux Console:

    ```
    root@petalinux:
    ```

    > Note: Wait until Linux boot finished, autologin is activated.

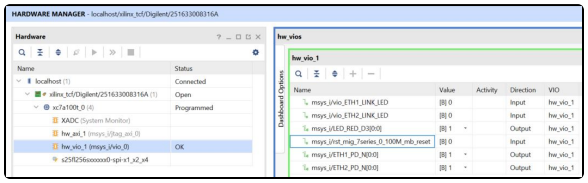
3. You can use Linux shell now.

    ```
    udhcpc                                  (ETH0 check)
    ```

Vivado HW Manager

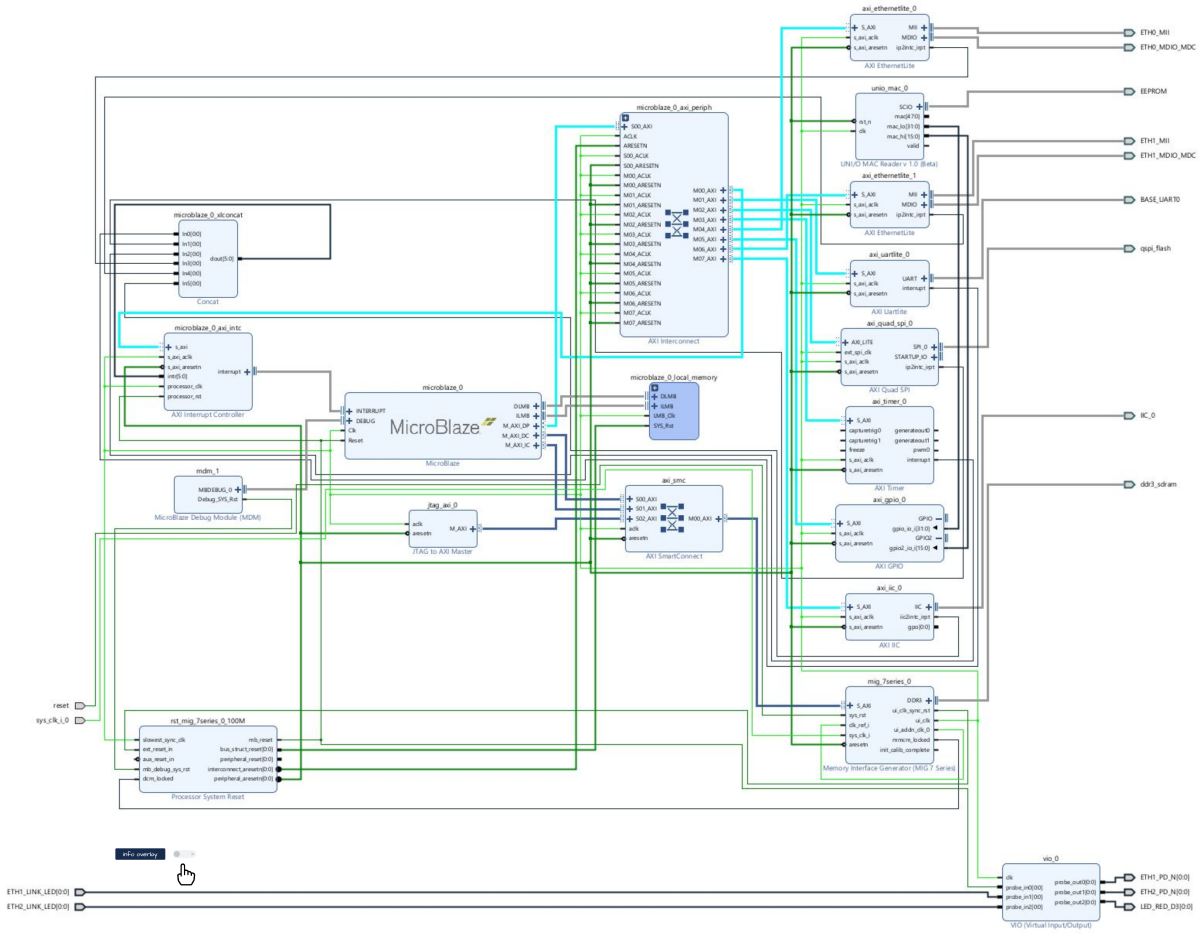
Open Vivado HW-Manager and add VIO signal to dashboard (*.ltx located on prebuilt folder)

- Control:
    - User LED Control
    - ETH Power Down
- Monitoring:
    - ETH Link Status
    - MicroBlaze Reset Status



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 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]
```

_i_bitgen.xdc

```
set_property BITSTREAM.CONFIG.UNUSEDPIN PULLDOWN [current_design]
```

Design specific constraints

_i_io.xdc

```
## set_property PACKAGE_PIN G3 [get_ports {LED_RED_XA_SC[0]}]
## set_property IOSTANDARD LVCMOS15 [get_ports {LED_RED_XA_SC[0]}]

set_property PACKAGE_PIN T10 [get_ports {ETH2_LINK_LED[0]}]
set_property IOSTANDARD LVCMOS33 [get_ports {ETH2_LINK_LED[0]}]
set_property PACKAGE_PIN V15 [get_ports {ETH1_LINK_LED[0]}]
set_property IOSTANDARD LVCMOS33 [get_ports {ETH1_LINK_LED[0]}]
set_property PACKAGE_PIN T18 [get_ports {ETH1_PD_N[0]}]
set_property PACKAGE_PIN D10 [get_ports {ETH2_PD_N[0]}]
set_property IOSTANDARD LVCMOS33 [get_ports {ETH2_PD_N[0]}]
set_property IOSTANDARD LVCMOS33 [get_ports {ETH1_PD_N[0]}]

set_property PACKAGE_PIN L15 [get_ports {LED_RED_D3[0]}]
set_property IOSTANDARD LVCMOS33 [get_ports {LED_RED_D3[0]}]

#EEPROM onewire (MAC ADDRESS)
set_property IOSTANDARD LVCMOS33 [get_ports EEPROM_tri_io]
set_property PACKAGE_PIN D9 [get_ports EEPROM_tri_io]

## IIC Interface
set_property PACKAGE_PIN G3 [get_ports IIC_0_sda_io]
set_property PACKAGE_PIN J5 [get_ports IIC_0_scl_io]
set_property IOSTANDARD LVCMOS15 [get_ports IIC_0_scl_io]
set_property IOSTANDARD LVCMOS15 [get_ports IIC_0_sda_io]
```

Software Design - Vitis

For Vitis project creation, follow instructions from:

Vitis

Application

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

spi_bootloader

TE modified SPI Bootloader from Henrik Brix Andersen.

Bootloader to load app or second bootloader from flash into DDR.

Here it loads the u-boot.elf from QSPI-Flash to RAM.

Descriptions:

- Modified Files: bootloader.c
- Changes:
    - Change the SPI defines in the header
    - Add some reiteration in the frist spi read call

hello_te0710

Hello TE0710 is a Hello World example as endless loop instead of one console output.

u-boot

U-Boot.elf is generated with PetaLinux. Vivado is used to generate *.mcs

Software Design - PetaLinux

For PetaLinux installation and project creation, follow instructions from:

- PetaLinux KICKstart

Config

Start with **petalinux-config** or **petalinux-config --get-hw-description**

Changes:

- SUBSYSTEM_FLASH_AXI_QUAD_SPI_0_BANKLESS_PART0_SIZE = **0x5E0000** (fpga)
- SUBSYSTEM_FLASH_AXI_QUAD_SPI_0_BANKLESS_PART1_SIZE = **0x400000** (boot)
- SUBSYSTEM_FLASH_AXI_QUAD_SPI_0_BANKLESS_PART2_SIZE = **0x20000** (bootenv)
- SUBSYSTEM_FLASH_AXI_QUAD_SPI_0_BANKLESS_PART3_SIZE = **0xD00000** (kernel)
    - (Set kernel flash Address to 0xA00000 (fpga+boot+bootenv) and Kernel size to 0xD00000)

U-Boot

Start with **petalinux-config -c u-boot**

Changes:

- CONFIG_ENV_IS_NOWHERE=y
- # CONFIG_ENV_IS_IN_SPI_FLASH is not set
- # CONFIG_PHY_ATHEROS is not set
- # CONFIG_PHY_BROADCOM is not set
- # CONFIG_PHY_DAVICOM is not set
- # CONFIG_PHY_LXT is not set
- # CONFIG_PHY_MICREL_KSZ90X1 is not set
- # CONFIG_PHY_MICREL is not set
- # CONFIG_PHY_NATSEMI is not set
- # CONFIG_PHY_REALTEK is not set
- CONFIG_RGMII=y

Busybox

Start with **petalinux-config -c busybox**

- Miscellaneous Utilities activate i2cget, i2cset, i2cdetect

Content of **platform-top.h** located in <plnx-proj-root>\project-spec\meta-user\recipes-bsp\u-boot\files:

```
#include <configs/microblaze-generic.h>
#include <configs/platform-auto.h>

#define CONFIG_SYS_BOOTM_LEN 0xF000000
```

Device Tree

Content of **system-user.dtsi** located in <petalinux project directory>\project-spec\meta-user\recipes-bsp\device-tree\files:

```
/include/ "system-conf.dtsi"
/ {
};

/* QSPI PHY */

&axi_quad_spi_0 {
    #address-cells = <1>;
    #size-cells = <0>;
    flash0: flash@0 {
        compatible = "jedec,spi-nor";
        spi-tx-bus-width=<1>;
        spi-rx-bus-width=<4>;
        reg = <0x0>;
        #address-cells = <1>;
        #size-cells = <1>;
        spi-max-frequency = <25000000>;
    };
};


/* ETH PHY */
&axi_ethernetlite_0 {
    phy-handle = <&phy0>;
    mdio {
        #address-cells = <1>;
        #size-cells = <0>;
        phy0: phy@0 {
            device_type = "ethernet-phy";
            reg = <1>;
        };
    };
};

/* ETH PHY 2nd */
&axi_ethernetlite_1 {

    phy-handle = <&phy1>;
    mdio {
        #address-cells = <1>;
        #size-cells = <0>;
        phy1: phy@1 {
            device_type = "ethernet-phy";
            reg = <1>;
        };
    };
};

/* i2c */
&axi_iic_0 {
                clock-frequency = <100000>;
                status = "okay";
        };
```

Kernel

Start with **petalinux-config -c kernel**

Changes:

- No changes.

Rootfs

Start with **petalinux-config -c rootfs**

Changes:

- # CONFIG_dropbear is not set
- # CONFIG_dropbear-dev is not set
- # CONFIG_dropbear-dbg is not set
- # CONFIG_packagegroup-core-ssh-dropbear is not set
- # CONFIG_packagegroup-core-ssh-dropbear-dev is not set
- # CONFIG_packagegroup-core-ssh-dropbear-dbg is not set
- # CONFIG_imagefeature-ssh-server-dropbear is not set
- CONFIG_imagefeature-serial-autologin-root = y

Applications

See "<project folder>\os\petalinux\project-spec\meta-user\recipes-apps\"

eeprom

eeprom is a simple bash script implemented in petalinux as an application that executes on startup. It reads the unique 48-bit MAC from the onboard eeprom and uses it to set the system MAC address.

Additional Software

No additional software is needed.

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
			• Release 2023.2

Error rendering macro 'page-info' Ambiguous method overloading for method jdk.proxy241.$Proxy3496#hasContentLevelPermission. Cannot resolve which method to invoke for [null, class java.lang.String, class com.atlassian.confluence.pages.Page]	**Error rendering macro 'page-info'** Ambiguous method overloading for method jdk.proxy241.$Proxy3496#hasContentLevelPermission. Cannot resolve which method to invoke for [null, class java.lang.String, class com.atlassian.confluence.pages.Page]	**Error rendering macro 'page-info'** Ambiguous method overloading for method jdk.proxy241.$Proxy3496#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.

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.

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]	atlassian . confluence.core. ContentEntityObject]	atlassian . confluence.core. ContentEntityObject]	
2022-02-16	v.9	John Hartfiel	• bugfix documenten style
2022-02-16	v.8	Waldemar Hanemann	• new spi bootloader by Henrik Brix Andersen • adjusted offsets
2022-02-14	v.7	Waldemar Hanemann	• 2021.2 update • document style update • added boot script • added eeprom interface for MAC address read-out • added simple sd card interface • added 2nd Ethernet Interface
2020-04-21	v.5	John Hartfiel	• Release 2019.2 • Docu update
2019-03-29	v.4	John Hartfiel	• Release 2017.4
2019-03-29	v.1	**Error rendering macro 'page-info'** Ambiguous method	• Initial release

overloading for method jdk.proxy241.$Proxy3496#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]	

---	All	**Error rendering macro 'page-info'** Ambiguous method overloading for method jdk.proxy241.$Proxy3496#hasContentLevelPermission. Cannot resolve which method to invoke for [null, class java.lang.String, class com.atlassian.confluence.	---

pages.
Page]
due to
overlapp
ing
prototyp
es
between
:
[interfac
e com.
atlassian
.
confluen
ce.user.
Conflue
nceUser
, class
java.
lang.
String,
class
com.
atlassian
.
confluen
ce.core.
Content
EntityOb
ject]
[interfac
e com.
atlassian
.user.
User,
class
java.
lang.
String,

		class com. atlassian . confluence.core. ContentEntityObject]	

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

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.proxy241.$Proxy3496#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]