TEB0912 Test Board

Table of contents

Overview

Refer to http://trenz.org/teb0912-info for the current online version of this manual and other available documentation.

Key Features

- Vitis/Vivado 2022.2
- PetaLinux
- SD
- 2x ETH
- CAN
- USB
- I2C
- RTC
- FMeter
- PCIe
- LED
- Modified FSBL for SI5395 programming

Revision History

Date	Vivado	Project Built	Authors	Description
2023-08-04	2022.2	TEB0912-test_board_noprebuilt-vivado_2022.2-build_3_20230804122209.zip TEB0912-test_board-vivado_2022.2-build_3_20230804122209.zip	Manuela Strücker	• 2022.2 release
2021-12-21	2020.2	TEB0912-test_board_noprebuilt-vivado_2020.2-build_9_20211220123954.zip TEB0912-test_board-vivado_2020.2-build_9_20211220123937.zip	Mohsen Chamanbaz	• Bugfix (disable SD card write protection)
2021-07-08	2020.2	TEB0912-test_board_noprebuilt-vivado_2020.2-build_5_20210708093945.zip TEB0912-test_board-vivado_2020.2-build_5_20210708093928.zip	Mohsen Chamanbaz	• FSBL files update

2021-06-28	2020.2	TEB0912-test_board_noprebuilt-vivado_2020.2-build_5_20210628141347.zip TEB0912-test_board-vivado_2020.2-build_5_20210628141329.zip	Mohsen Chamanbaz	• 2020.2 release
2020-06-10	2019.2	TEB0912-test_board_noprebuilt-vivado_2019.2-build_12_20200610085718.zip TEB0912-test_board-vivado_2019.2-build_12_20200610085620.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	2022.2	needed, Vivado is included into Vitis installation
PetaLinux	2022.2	needed
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	EMMC	Others	Notes
TEB0912-02-ABI21-A	02_11eg_1e_4gb	REV02	4GB	128MB	NA	4GB PL DDR	

TEB0912-03-ABI21-A*	11eg_1e_4gb	REV03	4GB	128MB	NA	4GB PL DDR	

*used as reference

Hardware Modules

Design supports following carriers:

Carrier Model	Notes
---	

*used as reference

Hardware Carrier

Additional HW Requirements:

Additional Hardware	Notes

*used as reference

Additional Hardware

Content

For general structure and usage of the reference design, see Project Delivery - Xilinx 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
SI5395	<project folder>\misc\PLL\Si5395	SI5395 Project with current PLL Configuration

init.sh	<project folder>\misc\sd\	Additional Initialization Script for Linux to read temperature of six temperature sensors on the board. For more information refer to TEB0912 CPLD.

Additional design sources

Prebuilt

File	File-Extension	Description
BIF-File	*.bif	File with description to generate Bin-File
BIN-File	*.bin	Flash Configuration File with Boot-Image (Zynq-FPGAs)
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
Device Tree	*.dts	Device tree (2 possible, one for u-boot and one for linux)
Hardware-Platform-Description-File	*.xsa	Exported Vivado hardware description file 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 prebult content)

Download

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

Reference Design is available on:

- TEB0912 "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 Xilinx 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 Xilinx 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) 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.
   - 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. Generate Programming Files with Vitis (recommended)
   a. Copy PetaLinux build image files to prebuilt folder
      - copy **u-boot.elf**, **system.dtb, bl31.elf, image.ub** and **boot.scr** from "<plnx-proj-root>/images/linux" to prebuilt folder

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

   b. 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

8. Generate Programming Files with Petalinux (alternative), see PetaLinux KICKstart

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.

Xilinx 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 **Boot.bin** on QSPI Flash and **image.ub** and **boot.scr** on **SD** or **USB**.

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 u-boot
TE::pr_program_flash -swapp hello_teb0912 (optional)
```

3. Copy **image.ub** and **boot.scr** on **SD** or **USB**
    - use files from "<project folder>_binaries_<Article Name>\boot_linux" from generated binary folder,see: Get prebuilt boot binaries
    - or use prebuilt file location, see "<project folder>\prebuilt\file_location.txt"
4. Set Boot Mode to **QSPI-Boot** and insert **SD** or **USB**.
    - Depends on Carrier, see carrier TRM.

SD-Boot mode

1. Copy **image.ub**, **boot.src** and **Boot.bin** on **SD**
    - use files from "<project folder>_binaries_<Article Name>\boot_linux" from generated binary folder, see: Get prebuilt boot binaries
    - or use prebuilt file location, see "<project folder>\prebuilt\file_location.txt"
2. Set Boot Mode to SD-Boot.
    - Depends on Carrier, see carrier TRM.
3. Insert SD-Card in SD-Slot.

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. (Optional) Insert PCIe Card (detection depends on Linux driver. Only some basic drivers are installed)
5. (Optional) Connect Network Cable
6. Power On PCB

   1. ZynqMP Boot ROM loads FSBL from SD/QSPI into OCM,

   2. FSBL init the PS, programs the PL using the bitstream and loads PMU, ATF and U-boot from SD/QSPI into DDR,

   3. U-boot loads Linux (**image.ub**) from SD/QSPI/... into DDR

Linux

1. Open Serial Console (e.g. putty)
   - Speed: 115200
   - select COM Port

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

2. Linux Console:

```
# password disabled
petalinux login: root
Password: root
```

   Note: Wait until Linux boot finished

3. You can use Linux shell now.

```
i2cdetect -y -r 0        (check I2C 1 Bus)
dmesg | grep rtc        (RTC check)
udhcpc                              (ETH0/1 check)
lspci                              (PCIe check)
```

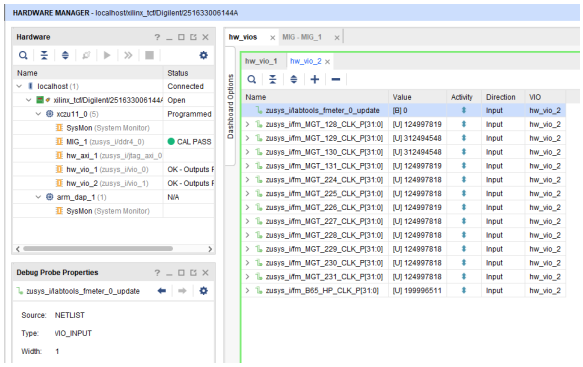
4. Option Features

   - Webserver to get access to ZynqMP
     - insert IP on web browser to start web interface
   - init.sh scripts
     - add init.sh script on SD, content will be load automatically on startup (template included in "<project folder>\misc\SD")
     - This Script file is responsible to read temperature of six temperature sensors on the board. For more information refer to TEB0912 CPLD.

Vivado HW Manager

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

- Control:
  - User LED Control (D16, D15)
- Monitoring:
  - MGT CLK Measurement:
    - Open Vivado HW-Manager and add VIO signal to dashboard (*.ltx located on prebuilt folder). Set radix from VIO signals to unsigned integer.Note: Frequency Counter is inaccurate and displayed unit is Hz
    - Default
      - B128_CLK_P, B131_CLK_P: 125MHz
      - B129_CL_P, B130_CLK_P: 312,5MHz
      - B224_CLK1 - B231_CLK1: 125MHz
      - B65_HP_CLK_P: 200MHz



Vivado Hardware Manager

System Design - Vivado

Block Design

PS Interfaces

Activated interfaces:

Type	Note
DDR	
QSPI	MIO
SD1	MIO
I2C0	MIO
I2C1	MIO
UART0	MIO
GPIO0..2	MIO
SWDT0..1	

TTC0..3	
GEM2	MIO
GEM3	MIO
USB0	MIO
PCIe	MIO/GTP
CAN0	MIO

PS Interfaces

Constrains

Basic module constrains

_i_bitgen_common.xdc

```
set_property BITSTREAM.GENERAL.COMPRESS TRUE [current_design]
set_property BITSTREAM.CONFIG.UNUSEDPIN PULLUP [current_design]
```

Design specific constrain

_i_io.xdc

```
#   AB34
MGT_128_CLK_P
#   AB35
MGT_128_CLK_N
#    W32
MGT_129_CLK_P
#    W33
MGT_129_CLK_N
#    R32
MGT_130_CLK_P
#    R33
MGT_130_CLK_N
#    L32
MGT_131_CLK_P
#    L33
MGT_131_CLK_N
set_property PACKAGE_PIN AB34 [get_ports {CLK_IN_D_128_131_clk_p[0]}]
set_property PACKAGE_PIN W32 [get_ports {CLK_IN_D_128_131_clk_p[1]}]
set_property PACKAGE_PIN R32 [get_ports {CLK_IN_D_128_131_clk_p[2]}]
set_property PACKAGE_PIN L32 [get_ports {CLK_IN_D_128_131_clk_p[3]}]

#   AB11
MGT_228_CLK_N
#   AB12
MGT_228_CLK_P
#    Y11
MGT_229_CLK_N
#    Y12
MGT_229_CLK_P
```

```
#    V11
MGT_230_CLK_N
#    V12
MGT_230_CLK_P
#    T11
MGT_231_CLK_N
#    T12
MGT_231_CLK_P
set_property PACKAGE_PIN AB12 [get_ports {CLK_IN_D_228_231_clk_p[0]}]
set_property PACKAGE_PIN Y12 [get_ports {CLK_IN_D_228_231_clk_p[1]}]
set_property PACKAGE_PIN V12 [get_ports {CLK_IN_D_228_231_clk_p[2]}]
set_property PACKAGE_PIN T12 [get_ports {CLK_IN_D_228_231_clk_p[3]}]

#   AK11
MGT_224_CLK_N
#   AK12
MGT_224_CLK_P
#   AH11
MGT_225_CLK_N
#   AH12
MGT_225_CLK_P
#   AF11
MGT_226_CLK_N
#   AF12
MGT_226_CLK_P
#   AD11
MGT_227_CLK_N
#   AD12
MGT_227_CLK_P
set_property PACKAGE_PIN AK12 [get_ports {CLK_IN_D_224_227_clk_p[0]}]
set_property PACKAGE_PIN AH12 [get_ports {CLK_IN_D_224_227_clk_p[1]}]
set_property PACKAGE_PIN AF12 [get_ports {CLK_IN_D_224_227_clk_p[2]}]
set_property PACKAGE_PIN AD12 [get_ports {CLK_IN_D_224_227_clk_p[3]}]

#   B65 CLK
set_property PACKAGE_PIN AR24 [get_ports {CLK_IN_D_B65_clk_p[0]}]
set_property IOSTANDARD LVDS [get_ports {CLK_IN_D_B65_clk_p[0]}]


#get_property -dict { IOSTANDARD LVCMOS33 PACKAGE_PIN D3
} [get_ports {+3.3V_ETH_PHY_EN}]   #removed on REV02 --> use unused pullup
for rev01
#set_property -dict { IOSTANDARD LVCMOS33 PACKAGE_PIN C1
} [get_ports {+3.3V_M2_KeyE_EN}]   #removed on REV02 --> use unused pullup
for rev01
#set_property -dict { IOSTANDARD LVCMOS33 PACKAGE_PIN G10
} [get_ports {ssd1_perstn[0]}]
set_property -dict { IOSTANDARD LVCMOS33 PACKAGE_PIN B6                  }
[get_ports {LED[0]}]
set_property -dict { IOSTANDARD LVCMOS33 PACKAGE_PIN B5                  }
[get_ports {LED[1]}]
set_property -dict { IOSTANDARD LVCMOS33 PACKAGE_PIN A5                  }
[get_ports {LED[2]}]
set_property -dict { IOSTANDARD LVCMOS33 PACKAGE_PIN A4                  }
[get_ports {LED[3]}]
#set_property -dict { IOSTANDARD LVCMOS33 PACKAGE_PIN G13
} [get_ports {M2M_SLEEP[0]}]
#set_property -dict { IOSTANDARD LVCMOS33 PACKAGE_PIN F13 PULLUP TRUE
} [get_ports {ssd1_wake[0]}]  #removed on REV02 --> use unused pullup for
rev01
#
```

```
#    B10
FF10_MPRS
#    C10
FF01_MPRS
#    C11
FF00_MPRS
#    D11
FF31_MPRS
#    D12
FF30_MPRS
#    E10
FF20_MPRS
#    E11
FF11_MPRS
#    E12
FF21_MPRS
#    J12
FFA_SDA
#    J14
FFA_SCL
#    K10
FFD_MPRS
#    K11
FFD_MSEL
#    K12
FFC_MPRS
#    K14
FFA_INTL
#    L10
FFD_INTL
#    L12
FFC_MSEL
#    L13
FFA_MPRS
#    L14
FFA_MSEL
#    M10
FFD_SDA
#    M11
FFB_SDA
#    M13
FFB_SCL
#    N10
FFD_SCL
#    N11
FF_AB_RSTL
#    N12
FF_CD_RSTL
#    N13
FFB_MSEL
#    N14
FFB_INTL
#    P12
FFC_INTL
#    P13
FFC_SCL
#    P14
FFB_MPRS
#    R14
FFC_SDA
#
```

```
#     E3         PEX_FATAL_ERRORn      REV02
only
#     E4         PEX_GPIO3             REV02
only
#     E5         PEX_LANE_GOOD2n       REV02
only
#     F4         PEX_LANE_GOOD1n       REV02
only
#     F5         PEX_LANE_GOOD0n       REV02
only
#
#     F6
DSPLL1_RST_N
#     F7
DSPLL0_RST_N
#
#    G11         W_DISABLE1n         REV01 other
name
#    G12         W_DISABLE2n         REV01 other
name
#    G13   M2M_SLEEP           REV02 only
#
#    F10         SSD1_CLKRQ          REV01
only
#    F13         SSD1_WAKE           REV01
only
#    G10         SSD1_PERSTn         REV01
only
#    G13         M2M_SLEEP           REV01 other
nameSSD1_SLEEP


set_property -dict { IOSTANDARD LVCMOS18 PACKAGE_PIN AK23               }
[get_ports {BUTTON[0]}]
set_property -dict { IOSTANDARD LVCMOS18 PACKAGE_PIN AL23               }
[get_ports {BUTTON[1]}]
set_property -dict { IOSTANDARD LVCMOS18 PACKAGE_PIN AJ24               }
[get_ports {BUTTON[2]}]
set_property -dict { IOSTANDARD LVCMOS18 PACKAGE_PIN AK24               }
[get_ports {BUTTON[3]}]

set_property -dict { IOSTANDARD DIFF_SSTL12_DCI  PACKAGE_PIN G26
} [get_ports {diff_clock_rtl_clk_p}]
set_property -dict { IOSTANDARD SSTL12_DCI PACKAGE_PIN N24
} [get_ports {ddr4_act_n}]
set_property -dict { IOSTANDARD SSTL12_DCI PACKAGE_PIN J27
} [get_ports {ddr4_adr[0]}]
set_property -dict { IOSTANDARD SSTL12_DCI PACKAGE_PIN J24
} [get_ports {ddr4_adr[1]}]
set_property -dict { IOSTANDARD SSTL12_DCI PACKAGE_PIN F27
} [get_ports {ddr4_adr[2]}]
set_property -dict { IOSTANDARD SSTL12_DCI PACKAGE_PIN E26
} [get_ports {ddr4_adr[3]}]
set_property -dict { IOSTANDARD SSTL12_DCI PACKAGE_PIN M25
} [get_ports {ddr4_adr[4]}]
set_property -dict { IOSTANDARD SSTL12_DCI PACKAGE_PIN D26
} [get_ports {ddr4_adr[5]}]
set_property -dict { IOSTANDARD SSTL12_DCI PACKAGE_PIN K27
} [get_ports {ddr4_adr[6]}]
set_property -dict { IOSTANDARD SSTL12_DCI PACKAGE_PIN E27
} [get_ports {ddr4_adr[7]}]
```

```
set_property -dict { IOSTANDARD SSTL12_DCI PACKAGE_PIN K26
} [get_ports {ddr4_adr[8]}]
set_property -dict { IOSTANDARD SSTL12_DCI PACKAGE_PIN H26
} [get_ports {ddr4_adr[9]}]
set_property -dict { IOSTANDARD SSTL12_DCI PACKAGE_PIN L24
} [get_ports {ddr4_adr[10]}]
set_property -dict { IOSTANDARD SSTL12_DCI PACKAGE_PIN F28
} [get_ports {ddr4_adr[11]}]
set_property -dict { IOSTANDARD SSTL12_DCI PACKAGE_PIN J23
} [get_ports {ddr4_adr[12]}]
set_property -dict { IOSTANDARD SSTL12_DCI PACKAGE_PIN J26
} [get_ports {ddr4_adr[13]}]
set_property -dict { IOSTANDARD SSTL12_DCI PACKAGE_PIN L23
} [get_ports {ddr4_adr[14]}]
set_property -dict { IOSTANDARD SSTL12_DCI PACKAGE_PIN K24
} [get_ports {ddr4_adr[15]}]
set_property -dict { IOSTANDARD SSTL12_DCI PACKAGE_PIN H23
} [get_ports {ddr4_adr[16]}]

## /* dummy for ddr4-ram */
set_property -dict { IOSTANDARD SSTL12_DCI PACKAGE_PIN G25              }
[get_ports {ddr4_adr17[0]}]


set_property -dict { IOSTANDARD SSTL12_DCI PACKAGE_PIN N26              }
[get_ports {ddr4_ba[0]}]
set_property -dict { IOSTANDARD SSTL12_DCI PACKAGE_PIN G23              }
[get_ports {ddr4_ba[1]}]
set_property -dict { IOSTANDARD SSTL12_DCI PACKAGE_PIN M23              }
[get_ports {ddr4_bg[0]}]
#set_property -dict { IOSTANDARD SSTL12_DCI PACKAGE_PIN P23             }
[get_ports {ddr4_bg[1]}]

## /* dummy for ddr4-ram */
set_property -dict { IOSTANDARD SSTL12_DCI PACKAGE_PIN P23              }
[get_ports {ddr4_bg1[0]}]

set_property -dict { IOSTANDARD DIFF_SSTL12_DCI PACKAGE_PIN F25         }
[get_ports {ddr4_ck_t[0]}]
set_property -dict { IOSTANDARD DIFF_SSTL12_DCI PACKAGE_PIN E25         }
[get_ports {ddr4_ck_c[0]}]

set_property -dict { IOSTANDARD SSTL12_DCI PACKAGE_PIN L25              }
[get_ports {ddr4_cke[0]}]
set_property -dict { IOSTANDARD SSTL12_DCI PACKAGE_PIN N23              }
[get_ports {ddr4_cs_n[0]}]
set_property -dict { IOSTANDARD POD12_DCI PACKAGE_PIN P18               }
[get_ports {ddr4_dm_n[0]}]
set_property -dict { IOSTANDARD POD12_DCI PACKAGE_PIN K19               }
[get_ports {ddr4_dm_n[1]}]
set_property -dict { IOSTANDARD POD12_DCI PACKAGE_PIN D19               }
[get_ports {ddr4_dm_n[2]}]
set_property -dict { IOSTANDARD POD12_DCI PACKAGE_PIN G22               }
[get_ports {ddr4_dm_n[3]}]
set_property -dict { IOSTANDARD POD12_DCI PACKAGE_PIN K29               }
[get_ports {ddr4_dm_n[4]}]
set_property -dict { IOSTANDARD POD12_DCI PACKAGE_PIN E29               }
[get_ports {ddr4_dm_n[5]}]
set_property -dict { IOSTANDARD POD12_DCI PACKAGE_PIN C36               }
[get_ports {ddr4_dm_n[6]}]
set_property -dict { IOSTANDARD POD12_DCI PACKAGE_PIN E32               }
```

```
[get_ports {ddr4_dm_n[7]}]

set_property -dict { IOSTANDARD DIFF_POD12_DCI PACKAGE_PIN N19          }
[get_ports {ddr4_dqs_c[0]}]
set_property -dict { IOSTANDARD DIFF_POD12_DCI PACKAGE_PIN J22          }
[get_ports {ddr4_dqs_c[1]}]
set_property -dict { IOSTANDARD DIFF_POD12_DCI PACKAGE_PIN B21          }
[get_ports {ddr4_dqs_c[2]}]
set_property -dict { IOSTANDARD DIFF_POD12_DCI PACKAGE_PIN E20          }
[get_ports {ddr4_dqs_c[3]}]
set_property -dict { IOSTANDARD DIFF_POD12_DCI PACKAGE_PIN F30          }
[get_ports {ddr4_dqs_c[4]}]
set_property -dict { IOSTANDARD DIFF_POD12_DCI PACKAGE_PIN A32          }
[get_ports {ddr4_dqs_c[5]}]
set_property -dict { IOSTANDARD DIFF_POD12_DCI PACKAGE_PIN A40          }
[get_ports {ddr4_dqs_c[6]}]
set_property -dict { IOSTANDARD DIFF_POD12_DCI PACKAGE_PIN C34          }
[get_ports {ddr4_dqs_c[7]}]
set_property -dict { IOSTANDARD DIFF_POD12_DCI PACKAGE_PIN N20          }
[get_ports {ddr4_dqs_t[0]}]
set_property -dict { IOSTANDARD DIFF_POD12_DCI PACKAGE_PIN K22          }
[get_ports {ddr4_dqs_t[1]}]
set_property -dict { IOSTANDARD DIFF_POD12_DCI PACKAGE_PIN C21          }
[get_ports {ddr4_dqs_t[2]}]
set_property -dict { IOSTANDARD DIFF_POD12_DCI PACKAGE_PIN F20          }
[get_ports {ddr4_dqs_t[3]}]
set_property -dict { IOSTANDARD DIFF_POD12_DCI PACKAGE_PIN G30          }
[get_ports {ddr4_dqs_t[4]}]
set_property -dict { IOSTANDARD DIFF_POD12_DCI PACKAGE_PIN B31          }
[get_ports {ddr4_dqs_t[5]}]
set_property -dict { IOSTANDARD DIFF_POD12_DCI PACKAGE_PIN A39          }
[get_ports {ddr4_dqs_t[6]}]
set_property -dict { IOSTANDARD DIFF_POD12_DCI PACKAGE_PIN D34          }
[get_ports {ddr4_dqs_t[7]}]

set_property -dict { IOSTANDARD SSTL12_DCI PACKAGE_PIN H24              }
[get_ports {ddr4_odt[0]}]
set_property -dict { IOSTANDARD POD12_DCI PACKAGE_PIN N25               }
[get_ports {ddr4_reset_n}]

set_property -dict { IOSTANDARD POD12_DCI PACKAGE_PIN N21               }
[get_ports {ddr4_dq[0]}]
set_property -dict { IOSTANDARD POD12_DCI PACKAGE_PIN M18               }
[get_ports {ddr4_dq[1]}]
set_property -dict { IOSTANDARD POD12_DCI PACKAGE_PIN M21               }
[get_ports {ddr4_dq[2]}]
set_property -dict { IOSTANDARD POD12_DCI PACKAGE_PIN M20               }
[get_ports {ddr4_dq[3]}]
set_property -dict { IOSTANDARD POD12_DCI PACKAGE_PIN P21               }
[get_ports {ddr4_dq[4]}]
set_property -dict { IOSTANDARD POD12_DCI PACKAGE_PIN L18               }
[get_ports {ddr4_dq[5]}]
set_property -dict { IOSTANDARD POD12_DCI PACKAGE_PIN M22               }
[get_ports {ddr4_dq[6]}]
set_property -dict { IOSTANDARD POD12_DCI PACKAGE_PIN L19               }
[get_ports {ddr4_dq[7]}]
set_property -dict { IOSTANDARD POD12_DCI PACKAGE_PIN H19               }
[get_ports {ddr4_dq[8]}]
set_property -dict { IOSTANDARD POD12_DCI PACKAGE_PIN L20               }
[get_ports {ddr4_dq[9]}]
```

```
set_property -dict { IOSTANDARD POD12_DCI PACKAGE_PIN H20               }
[get_ports {ddr4_dq[10]}]
set_property -dict { IOSTANDARD POD12_DCI PACKAGE_PIN K21               }
[get_ports {ddr4_dq[11]}]
set_property -dict { IOSTANDARD POD12_DCI PACKAGE_PIN G20               }
[get_ports {ddr4_dq[12]}]
set_property -dict { IOSTANDARD POD12_DCI PACKAGE_PIN K20               }
[get_ports {ddr4_dq[13]}]
set_property -dict { IOSTANDARD POD12_DCI PACKAGE_PIN H21               }
[get_ports {ddr4_dq[14]}]
set_property -dict { IOSTANDARD POD12_DCI PACKAGE_PIN J21               }
[get_ports {ddr4_dq[15]}]
set_property -dict { IOSTANDARD POD12_DCI PACKAGE_PIN B22               }
[get_ports {ddr4_dq[16]}]
set_property -dict { IOSTANDARD POD12_DCI PACKAGE_PIN C20               }
[get_ports {ddr4_dq[17]}]
set_property -dict { IOSTANDARD POD12_DCI PACKAGE_PIN A22               }
[get_ports {ddr4_dq[18]}]
set_property -dict { IOSTANDARD POD12_DCI PACKAGE_PIN A19               }
[get_ports {ddr4_dq[19]}]

set_property -dict { IOSTANDARD POD12_DCI PACKAGE_PIN B23               }
[get_ports {ddr4_dq[20]}]
set_property -dict { IOSTANDARD POD12_DCI PACKAGE_PIN B20               }
[get_ports {ddr4_dq[21]}]
set_property -dict { IOSTANDARD POD12_DCI PACKAGE_PIN A23               }
[get_ports {ddr4_dq[22]}]
set_property -dict { IOSTANDARD POD12_DCI PACKAGE_PIN A20               }
[get_ports {ddr4_dq[23]}]
set_property -dict { IOSTANDARD POD12_DCI PACKAGE_PIN F22               }
[get_ports {ddr4_dq[24]}]
set_property -dict { IOSTANDARD POD12_DCI PACKAGE_PIN E19               }
[get_ports {ddr4_dq[25]}]
set_property -dict { IOSTANDARD POD12_DCI PACKAGE_PIN E22               }
[get_ports {ddr4_dq[26]}]
set_property -dict { IOSTANDARD POD12_DCI PACKAGE_PIN D21               }
[get_ports {ddr4_dq[27]}]
set_property -dict { IOSTANDARD POD12_DCI PACKAGE_PIN F23               }
[get_ports {ddr4_dq[28]}]
set_property -dict { IOSTANDARD POD12_DCI PACKAGE_PIN F19               }
[get_ports {ddr4_dq[29]}]

set_property -dict { IOSTANDARD POD12_DCI PACKAGE_PIN D22               }
[get_ports {ddr4_dq[30]}]
set_property -dict { IOSTANDARD POD12_DCI PACKAGE_PIN E21               }
[get_ports {ddr4_dq[31]}]
set_property -dict { IOSTANDARD POD12_DCI PACKAGE_PIN F31               }
[get_ports {ddr4_dq[32]}]
set_property -dict { IOSTANDARD POD12_DCI PACKAGE_PIN J28               }
[get_ports {ddr4_dq[33]}]
set_property -dict { IOSTANDARD POD12_DCI PACKAGE_PIN J30               }
[get_ports {ddr4_dq[34]}]
set_property -dict { IOSTANDARD POD12_DCI PACKAGE_PIN H28               }
[get_ports {ddr4_dq[35]}]
set_property -dict { IOSTANDARD POD12_DCI PACKAGE_PIN F32               }
[get_ports {ddr4_dq[36]}]
set_property -dict { IOSTANDARD POD12_DCI PACKAGE_PIN G28               }
[get_ports {ddr4_dq[37]}]
set_property -dict { IOSTANDARD POD12_DCI PACKAGE_PIN H30               }
[get_ports {ddr4_dq[38]}]
set_property -dict { IOSTANDARD POD12_DCI PACKAGE_PIN F29               }
```

```
[get_ports {ddr4_dq[39]}]

set_property -dict { IOSTANDARD POD12_DCI PACKAGE_PIN E31               }
[get_ports {ddr4_dq[40]}]
set_property -dict { IOSTANDARD POD12_DCI PACKAGE_PIN C30               }
[get_ports {ddr4_dq[41]}]
set_property -dict { IOSTANDARD POD12_DCI PACKAGE_PIN C31               }
[get_ports {ddr4_dq[42]}]
set_property -dict { IOSTANDARD POD12_DCI PACKAGE_PIN B30               }
[get_ports {ddr4_dq[43]}]
set_property -dict { IOSTANDARD POD12_DCI PACKAGE_PIN D31               }
[get_ports {ddr4_dq[44]}]
set_property -dict { IOSTANDARD POD12_DCI PACKAGE_PIN C29               }
[get_ports {ddr4_dq[45]}]
set_property -dict { IOSTANDARD POD12_DCI PACKAGE_PIN A30               }
[get_ports {ddr4_dq[46]}]
set_property -dict { IOSTANDARD POD12_DCI PACKAGE_PIN A29               }
[get_ports {ddr4_dq[47]}]
set_property -dict { IOSTANDARD POD12_DCI PACKAGE_PIN C42               }
[get_ports {ddr4_dq[48]}]
set_property -dict { IOSTANDARD POD12_DCI PACKAGE_PIN B36               }
[get_ports {ddr4_dq[49]}]

set_property -dict { IOSTANDARD POD12_DCI PACKAGE_PIN B40               }
[get_ports {ddr4_dq[50]}]
set_property -dict { IOSTANDARD POD12_DCI PACKAGE_PIN B37               }
[get_ports {ddr4_dq[51]}]
set_property -dict { IOSTANDARD POD12_DCI PACKAGE_PIN B42               }
[get_ports {ddr4_dq[52]}]
set_property -dict { IOSTANDARD POD12_DCI PACKAGE_PIN A37               }
[get_ports {ddr4_dq[53]}]
set_property -dict { IOSTANDARD POD12_DCI PACKAGE_PIN B41               }
[get_ports {ddr4_dq[54]}]
set_property -dict { IOSTANDARD POD12_DCI PACKAGE_PIN A38               }
[get_ports {ddr4_dq[55]}]
set_property -dict { IOSTANDARD POD12_DCI PACKAGE_PIN B35               }
[get_ports {ddr4_dq[56]}]
set_property -dict { IOSTANDARD POD12_DCI PACKAGE_PIN B32               }
[get_ports {ddr4_dq[57]}]
set_property -dict { IOSTANDARD POD12_DCI PACKAGE_PIN A33               }
[get_ports {ddr4_dq[58]}]
set_property -dict { IOSTANDARD POD12_DCI PACKAGE_PIN D33               }
[get_ports {ddr4_dq[59]}]

set_property -dict { IOSTANDARD POD12_DCI PACKAGE_PIN A35               }
[get_ports {ddr4_dq[60]}]
set_property -dict { IOSTANDARD POD12_DCI PACKAGE_PIN A34               }
[get_ports {ddr4_dq[61]}]
set_property -dict { IOSTANDARD POD12_DCI PACKAGE_PIN C33               }
[get_ports {ddr4_dq[62]}]
set_property -dict { IOSTANDARD POD12_DCI PACKAGE_PIN B33               }
[get_ports {ddr4_dq[63]}]

set_property -dict { IOSTANDARD LVCMOS33 PACKAGE_PIN J9                 }
[get_ports {IIC_0_scl_io}]
set_property -dict { IOSTANDARD LVCMOS33 PACKAGE_PIN H9                 }
[get_ports {IIC_0_sda_io}]


#create_clock -name c0_sys_clk -period 4.998 [get_ports
diff_clock_rtl_clk_p]
```

Software Design - Vitis

For Vitis project creation, follow instructions from:

Vitis

Application

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

zynqmp_fsbl

TE modified 2022.2 FSBL

General:

- Modified Files: xfsbl_main.c, xfsbl_hooks.h/.c, xfsbl_board.h/.c (search for 'TE Mod' on source code)
- Add Files: te_xfsbl_hooks.h/.c (for hooks and board)
- General Changes:
    - Display FSBL Banner and Device Name

Module Specific:

- Add Files: all TE Files start with te_*
    - Si5395 DSPLL0 (U64) configuration
    - Si5395 DSPLL1 (U65) configuration
    - PCIe and eth reset

zynqmp_pmufw

Xilinx default PMU firmware.

General Example:

hello_teb0912

Hello TEB0912 is a Xilinx Hello World example as endless loop instead of one console output.

u-boot

U-Boot.elf is generated with PetaLinux. Vitis is used to generate Boot.bin.

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:

- add new flash partition for bootscr and sizing
  - CONFIG_SUBSYSTEM_FLASH_PSU_QSPI_0_BANKLESS_PART0_SIZE=0x3000000
  - CONFIG_SUBSYSTEM_FLASH_PSU_QSPI_0_BANKLESS_PART1_SIZE=0x2000000
  - CONFIG_SUBSYSTEM_FLASH_PSU_QSPI_0_BANKLESS_PART2_SIZE=0x40000
  - CONFIG_SUBSYSTEM_FLASH_PSU_QSPI_0_BANKLESS_PART3_NAME="bootscr"
  - CONFIG_SUBSYSTEM_FLASH_PSU_QSPI_0_BANKLESS_PART3_SIZE=0x80000
- Identification
  - CONFIG_SUBSYSTEM_HOSTNAME="Trenz"
  - CONFIG_SUBSYSTEM_PRODUCT="TEB0912"

U-Boot

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

Changes:

- MAC from eeprom together with uboot and device tree settings:
  - CONFIG_ENV_OVERWRITE=y
  - CONFIG_ZYNQ_MAC_IN_EEPROM is not set
  - CONFIG_NET_RANDOM_ETHADDR is not set
- Boot Modes:
  - CONFIG_QSPI_BOOT=y
  - CONFIG_SD_BOOT=y
  - CONFIG_ENV_IS_IN_FAT is not set
  - CONFIG_ENV_IS_IN_NAND is not set
  - CONFIG_ENV_IS_IN_SPI_FLASH is not set
  - CONFIG_SYS_REDUNDAND_ENVIRONMENT is not set
  - CONFIG_BOOT_SCRIPT_OFFSET=0x5040000
- Identification
  - CONFIG_IDENT_STRING=" TEB0912"

Change platform-top.h:

Device Tree

```
/include/ "system-conf.dtsi"

/*------------------------- QSPI -------------------------*/
&qspi {
    #address-cells = <1>;
    #size-cells = <0>;
    status = "okay";
    flash0: flash@0 {

        compatible = "jedec,spi-nor";
        reg = <0x0>;
        #address-cells = <1>;
        #size-cells = <1>;

    };
};


/*------------------ SD1 with level shifter ---------------*/
&sdhci1 {
```

```
    pinctrl-names = "default";
    pinctrl-0 = <&pinctrl_sdhci1_default>;
    no-1-8-v;
    disable-wp;
};

&pinctrl0 {
    status = "okay";
    pinctrl_sdhci1_default: sdhci1-default {
        mux {
            groups = "sdio1_0_grp";
            function = "sdio1";
        };

        conf {
            groups = "sdio1_0_grp";
            slew-rate = <1>;
            io-standard = <1>;
            bias-disable;
        };
/*
        mux-cd {
            groups = "sdio1_cd_0_grp";
            function = "sdio1_cd";
        };

        conf-cd {
            groups = "sdio1_cd_0_grp";
            bias-high-impedance;
            bias-pull-up;
            slew-rate = <1>;
            io-standard = <1>;
        };

        mux-wp {
            groups = "sdio1_wp_0_grp";
            function = "sdio1_wp";
        };

        conf-wp {
            groups = "sdio1_wp_0_grp";
            bias-high-impedance;
            bias-pull-up;
            slew-rate = <1>;
            io-standard = <1>;
        };
*/

    };
};


/*------------------------ ETH PHY ----------------------*/


&gem2 {
    status = "okay";
    phy-handle = <&ethernet_phy2>;

    nvmem-cells = <&eth2_addr>;
    nvmem-cell-names = "mac-address";
```

```
    mdio {

      ethernet_phy1: ethernet-phy@0 {

          //compatible = "marvell,88e1510";
          reg = <0>;
      };

      ethernet_phy2: ethernet-phy@1 {

          //compatible = "marvell,88e1510";
          reg = <1>;
      };

    };
};


&gem3 {
    status = "okay";
    phy-handle = <&ethernet_phy1>;

    nvmem-cells = <&eth1_addr>;
    nvmem-cell-names = "mac-address";

    phy-mode = "rgmii-id";
};


/*-------------------- I2C Bus on PL ----------------------*/
/* for CPLD I2C controller */
&axi_iic_0 {
        iexp@20 {// GPIO in CPLD
            #gpio-cells = <2>;
            //compatible = "ti,pcf8574";
            reg = <0x20>;
            gpio-controller;

        };
};


/*----------------- I2C Bus on PS MIO 34,35 ---------------*/
/* on MIO */
&i2c0 {
    eeprom51: eeprom@51 {
        compatible = "microchip,24aa025", "atmel,24c02";
        reg = <0x51>;

        #address-cells = <1>;
        #size-cells = <1>;
        eth2_addr: eth-mac-addr@FA {
          reg = <0xFA 0x06>;
        };
    };

    eeprom52: eeprom@52 {
        compatible = "microchip,24aa025", "atmel,24c02";
        reg = <0x52>;
```

```
        #address-cells = <1>;
        #size-cells = <1>;
        eth1_addr: eth-mac-addr@FA {
          reg = <0xFA 0x06>;

        };
    };

    eeprom53: eeprom@53 {
        compatible = "microchip,24aa025", "atmel,24c02";
        reg = <0x53>;
    };

    eeprom56: eeprom@56 {
        compatible = "microchip,24lc128", "atmel,24c128";
        reg = <0x56>;
    };

    rtc@6F {        // Real Time Clock
        compatible = "isl12022";
        reg = <0x6F>;
    };

};

&rtc {
    status = "disabled";
};


/*----------------- I2C Bus on PS MIO 28,29 ---------------*/
&i2c1 {
    i2cswitch@75 { // u35
        compatible = "nxp,pca9544";
        #address-cells = <1>;
        #size-cells = <0>;
        reg = <0x70>;
        i2c-mux-idle-disconnect;
        i2c@0 { // DSPLL0
            reg = <0>;
        };
        i2c@1 { // DSPLL1
            reg = <1>;
        };
        i2c@2 { // J34
            reg = <2>;
        };
        i2c@3 { // J34
            reg = <3>;
        };
    };
};
```

Kernel

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

Changes:

- Only needed to fix JTAG Debug issue:
  - # CONFIG_CPU_FREQ is not set
- Support PCIe memory card
  - CONFIG_NVME_CORE=y
  - CONFIG_BLK_DEV_NVME=y
  - # CONFIG_NVME_MULTIPATH is not set
  - # CONFIG_NVME_HWMON is not set
  - CONFIG_NVME_TARGET=y
  - # CONFIG_NVME_TARGET_PASSTHRU is not set
  - # CONFIG_NVME_TARGET_LOOP is not set
  - # CONFIG_NVME_TARGET_FC is not set
  - # CONFIG_NVME_TARGET_TCP is not set

Rootfs

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

Changes:

- For web server app:
  - CONFIG_busybox-httpd=y
- For additional test tools only:
  - CONFIG_i2c-tools=y
  - CONFIG_packagegroup-petalinux-utils=y (util-linux,cpufrequtils,bridge-utils,mtd-utils, usbutils,pciutils,canutils,i2c-tools,smartmontools,e2fsprogs)
- For auto login:
  - CONFIG_auto-login=y
  - CONFIG_ADD_EXTRA_USERS="root:root;petalinux:;"

FSBL patch (alternative for vitis fsbl trenz patch)

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

Applications

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

startup

Script App to load init.sh from SD Card if available.

webfwu

Webserver application suitable for ZynqMP access. Need busybox-httpd

Additional Software

SI5395 of carrier board DSPLL0 (U64)

File location "<project folder>\misc\PLL\Si5395-*_DSPLL0_*.slabtimeproj"

General documentation how you work with this project will be available on Si5395

SI5395 of carrier board DSPLL1 (U65)

File location "<project folder>\misc\PLL\Si5395-*_DSPLL1_*.slabtimeproj"

General documentation how you work with this project will be available on Si5395

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	• 2022.2 release

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.

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.

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.

Content EntityObject] [interface com.atlassian.user.User, class java.lang.String, class com.atlassian.confluence.core.ContentEntityObject]	Content EntityObject] [interface com.atlassian.user.User, class java.lang.String, class com.atlassian.confluence.core.ContentEntityObject]	Content EntityObject] [interface com.atlassian.user.User, class java.lang.String, class com.atlassian.confluence.core.ContentEntityObject]	
2021-12-21	v.10	Mohsen Chamanbaz	• Bugfix (disable SD card write protection)
2021-07-08	v.8	Mohsen Chamanbaz	• FSBL files update
2021-06-28	v.7	Mohsen Chamanbaz	• 2020.2 release
2020-06-10	v.3	John Hartfiel	• 2019.2 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

:
[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
.
confluen
ce.core.
Content
EntityOb

		ject]	

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]