TE0808 StarterKit

Table of contents

Overview

Linux with basic periphery of TE0808 StarterKit (TEBF0808 Carrier).

Refer to http://trenz.org/te0808-info for the current online version of this manual and other available documentation.

Key Features

- Vitis/Vivado 2023.2
- TEBF0808
- Linux
- USB
- ETH
- MAC from EEPROM
- PCIe
- SATA
- SD
- I2C
- RGPIO
- Display Port (DP)
- user LED access

Revision History

Date	Vivado	Project Built	Authors	Description
2024-03-13	2023.2	TE0808-StarterKit-vivado_2023.2-build_4_20240313131239.zip TE0808-StarterKit_noprebuilt-vivado_2023.2-build_4_20240313131239.zip	Manuela Strücker	• 2023.2 release • new assembly variants
2023-06-01	2022.2	TE0808-StarterKit-vivado_2022.2-build_1_20230601094128.zip TE0808-StarterKit_noprebuilt-vivado_2022.2-build_1_20230601094128.zip	Manuela Strücker	• 2022.2 release • new assembly variants
2023-04-13	2021.2.1	TE0808-StarterKit-vivado_2021.2-build_20_20230413092755.zip TE0808-StarterKit_noprebuilt-vivado_2021.2-build_20_20230413092755.zip	Manuela Strücker	• new assembly variants

2022-09-29	2021.2.1	TE0808-StarterKit-vivado_2021.2-build_17_20220929082218.zip TE0808-StarterKit_noprebuilt-vivado_2021.2-build_17_20220929082218.zip	Manuela Strücker	• script update • new assembly variants
2022-09-12	2021.2.1	TE0808-StarterKit-vivado_2021.2-build_15_20220912090625.zip TE0808-StarterKit_noprebuilt-vivado_2021.2-build_15_20220912090625.zip	Manuela Strücker	• update board part files compatible to Vivado 2021.2.1
2022-03-16	2021.2	TE0808-StarterKit-vivado_2021.2-build_11_20220316082848.zip TE0808-StarterKit_noprebuilt-vivado_2021.2-build_11_20220316082848.zip	Manuela Strücker	• 2021.2 release • update board files
2021-05-12	2020.2	TE0808-StarterKit-vivado_2020.2-build_5_20210512133800.zip TE0808-StarterKit_noprebuilt-vivado_2020.2-build_5_20210512133822.zip	John Hartfiel	• update board files • boot.scr update to version1 image.ub on sd, eMMC, USB possible
2021-02-05	2020.2	TE0808-StarterKit-vivado_2020.2-build_1_20210205120058.zip TE0808-StarterKit_noprebuilt-vivado_2020.2-build_1_20210205120122.zip	John Hartfiel	• bugfix init.sh script usage
2021-02-05	2020.2	TE0808-StarterKit_noprebuilt-vivado_2020.2-build_1_20210204142828.zip TE0808-StarterKit-vivado_2020.2-build_1_20210204142713.zip	John Hartfiel	• 2020.2 update • add boot.scr file • device tree has change • petalinux fsbl patch (betaversion)
2020-09-29	2019.2	TE0808-StarterKit_noprebuilt-vivado_2019.2-build_15_20200928195324.zip TE0808-StarterKit-vivado_2019.2-build_15_20200928195304.zip	John Hartfiel	• bugfix 8GB board part files

2020-09-22	2019.2	TE0808-StarterKit_noprebuilt-vivado_2019.2-build_14_20200922071643.zip TE0808-StarterKit-vivado_2019.2-build_14_20200922071704.zip	John Hartfiel	• new assembly variants
2020-03-25	2019.2	TE0808-StarterKit_noprebuilt-vivado_2019.2-build_8_20200325083508.zip TE0808-StarterKit-vivado_2019.2-build_8_20200325083436.zip	John Hartfiel	• script update
2020-01-22	2019.2	TE0808-StarterKit_noprebuilt-vivado_2019.2-build_3_20200122142340.zip TE0808-StarterKit-vivado_2019.2-build_3_20200122142318.zip	John Hartfiel	• 2019.2 update • Vitis support • FSBL SI programming procedure update • petalinux device tree and u-boot update
2019-08-09	2018.3	TE0808-StarterKit_noprebuilt-vivado_2018.3-build_07_20190809131638.zip TE0808-StarterKit-vivado_2018.3-build_07_20190809131620.zip	John Hartfiel	• new assembly variants • small fsbl update (supports all GTR disabled now)
2019-05-07	2018.3	TE0808-StarterKit_noprebuilt-vivado_2018.3-build_05_20190507124429.zip TE0808-StarterKit-vivado_2018.3-build_05_20190507124418.zip	John Hartfiel	• new assembly variant • TE Script update • rework of the FSBLs • some additional Linux features • MAC from EEPROM • new assembly variants • remove special compiler flags, which was needed in 2018.2

2018-07-11	2018.2	TE0808-StarterKit_noprebuilt-vivado_2018.2-build_02_20180711091558.zip TE0808-StarterKit-vivado_2018.2-build_02_20180711091049.zip	John Hartfiel	• small petalinux changes • IO renaming • PL Design changes • additional notes for FSBL generated with Win SDK • changed *.bif
2018-05-24	2017.4	TE0808-StarterKit_noprebuilt-vivado_2017.4-build_10_20180524091231.zip TE0808-StarterKit-vivado_2017.4-build_10_20180524091208.zip	John Hartfiel	• solved Linux flash issue
2018-03-29	2017.4	TE0808-StarterKit_noprebuilt-vivado_2017.4-build_07_20180329145308.zip TE0808-StarterKit-vivado_2017.4-build_07_20180329145246.zip	John Hartfiel	• new assembly variant
2018-02-06	2017.4	TE0808-StarterKit_noprebuilt-vivado_2017.4-build_05_20180206082740.zip TE0808-StarterKit-vivado_2017.4-build_05_20180206082722.zip	John Hartfiel	• same clk for both VIO
2018-02-05	2017.4	TE0808-StarterKit_noprebuilt-vivado_2017.4-build_05_20180205083231.zip TE0808-StarterKit-vivado_2017.4-build_05_20180205083208.zip	John Hartfiel	• solved JTAG /Linux problem
2018-01-17	2017.4	TE0808-StarterKit-vivado_2017.4-build_05_20180117094213.zip TE0808-StarterKit_noprebuilt-vivado_2017.4-build_05_20180117094231.zip	John Hartfiel	• solved USB problem • small board part update
2018-01-15	2017.4	TE0808-StarterKit-vivado_2017.4-build_03_20180115092306.zip TE0808-StarterKit_noprebuilt-vivado_2017.4-build_03_20180115092511.zip	John Hartfiel	• rework board part files • rework design

2017-12-18	2017.2	TE0808-StarterKit_noprebuilt-vivado_2017.2-build_07_20171219151749.zip TE0808-StarterKit-vivado_2017.2-build_07_20171219151728.zip	John Hartfiel	• initial release

Design Revision History

Release Notes and Know Issues

Issues	Description	Workaround/Solution	To be fixed version
Xilinx Software	Incompatibility of board files for ZynqMP with eMMC activated between 2021.2 and 2021.2.1 patch, see Xilinx Forum Request	use corresponding board files for the Vivado versions	--
MAC from EEPROM	The MAC address stored in the EEPROM is not read out and initialised correctly during start-up. This is caused by two I2C expanders each switched to the same EEPROM with the same address i2cswitch@73 --> i2c@5 --> reg = <0x50> and i2cswitch@77 --> i2c@4 --> reg = <0x50>	Switching the second I2C expander (i2cswitch@77) to another channel in the fsbl solves the error during the start-up procedure.	**Solved** with 20220316 update
QSPI Flash	Flash programming is not supported with boot mode QSPI or SD.	If flash programming fails, configure device for JTAG boot mode and try again or use oder Vivado Versions for programming. (Vivado 2020.2 or 2019.2)	--
Flash access on Linux	Device tree is not correct on Linux	add compatibility to "comp atible “jedec,spi-nor""	**Solved** with 20180524 update
USB UART Terminal is blocked/ SDK Debugging is blocked	This happens only with 2017.4 Linux, when JTAG connection is established on Vivado HW Manager.	Do not use HW Manager connection, or if debugging is necessary: 1. Boot linux with usb terminal 2. From the terminal: root root mount ifconfig eth0 3. Open two new SSH terminals via ethernet: root root , run user application ... 4. Exit and close the usb terminal	**Solved** with 20180205 update

Known Issues

Requirements

Software

Software	Version	Note
Vitis	2023.2	needed, Vivado is included into Vitis installation
PetaLinux	2023.2	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
~~TE0808-ES1~~	es1_2gb	REV03|REV02	2GB	64MB	NA	NA	Not longer supported by vivado
~~TE0808-ES2~~	es2_2gb	REV04|REV03	2GB	64MB	NA	NA	Not longer supported by vivado
~~TE0808-2ES2~~	2es2_2gb	REV04|REV03	2GB	64MB	NA	NA	Not longer supported by vivado
TE0808-04-06EG-1E3	6eg_1e_4gb	REV04	4GB	128MB	NA	1 mm connectors	NA
TE0808-04-06EG-1EE	6eg_1e_4gb	REV04	4GB	128MB	NA	NA	NA
TE0808-04-09EG-1EA	9eg_1e_2gb	REV04	2GB	64MB	NA	NA	NA
TE0808-04-09EG-1EB	9eg_1e_4gb	REV04	4GB	64MB	NA	NA	NA
TE0808-04-09EG-1ED	9eg_1e_4gb	REV04	4GB	64MB	NA	1 mm connectors	NA
TE0808-04-09EG-1EE	9eg_1e_4gb	REV04	4GB	128MB	NA	NA	NA
TE0808-04-09EG-1EL	9eg_1e_4gb	REV04	4GB	128MB	NA	1 mm connectors	NA
TE0808-04-09EG-2IB	9eg_2i_4gb	REV04	4GB	64MB	NA	NA	NA
TE0808-04-09EG-2IE	9eg_2i_4gb	REV04	4GB	128MB	NA	NA	NA
TE0808-04-6BE21-A	6eg_1e_4gb	REV04	4GB	128MB	NA	NA	NA
TE0808-04-6BE21-L	6eg_1e_4gb	REV04	4GB	128MB	NA	1 mm connectors	NA
TE0808-04-6BI21-A	6eg_1i_4gb	REV04	4GB	128MB	NA	NA	NA
TE0808-04-6BI21-X	6eg_1i_4gb	REV04	4GB	128MB	NA	NA	U41 replaced with schottky diodes

TE0808-04-6GI21-L	6eg_2i_4gb	REV04	4GB	128MB	NA	1 mm connectors	NA
TE0808-04-9BE21-A	9eg_1e_4gb	REV04	4GB	128MB	NA	NA	NA
TE0808-04-9BE21-L	9eg_1e_4gb	REV04	4GB	128MB	NA	1 mm connectors	NA
TE0808-04-9GI21-A	9eg_2i_4gb	REV04	4GB	128MB	NA	NA	NA
TE0808-04-15EG-1EB	15eg_1e_4gb	REV04	4GB	64MB	NA	NA	NA
TE0808-04-15EG-1EE	15eg_1e_4gb	REV04	4GB	128MB	NA	NA	NA
TE0808-04-BBE21-A	15eg_1e_4gb	REV04	4GB	128MB	NA	NA	NA
TE0808-05-6BE21-A	6eg_1e_4gb	REV05	4GB	128MB	NA	NA	NA
TE0808-05-6BE21-F	6eg_1e_4gb	REV05	4GB	128MB	NA	NA	NA
TE0808-05-6BE21-AK	6eg_1e_4gb	REV05	4GB	128MB	NA	NA	NA
TE0808-05-6BE21-L	6eg_1e_4gb	REV05	4GB	128MB	NA	1 mm connectors	NA
TE0808-05-6BI21-D	6eg_1i_4gb	REV05	4GB	128MB	NA	1 mm connectors	SoC without encryption
TE0808-05-6BI21-X	6eg_1i_4gb	REV05	4GB	128MB	NA	NA	U41 replaced with schottky diodes
TE0808-05-6BI41-X	6eg_1i_8gb	REV05	8GB	128MB	NA	NA	Single Die DDR; U41 replaced with schottky diodes
TE0808-05-9BE21-A	9eg_1e_4gb	REV05	4GB	128MB	NA	NA	NA
TE0808-05-9BE21-AK	9eg_1e_4gb	REV05	4GB	128MB	NA	1 mm connectors	NA
TE0808-05-9BE21-AZ	9eg_1e_4gb	REV05	4GB	128MB	NA	1 mm connectors	NA
TE0808-05-9BE21-KZ	9eg_1e_4gb	REV05	4GB	128MB	NA	1 mm connectors	NA
TE0808-05-9BE21-L	9eg_1e_4gb	REV05	4GB	128MB	NA	1 mm connectors	NA
TE0808-05-9BE21-LK	9eg_1e_4gb	REV05	4GB	128MB	NA	1 mm connectors	NA
TE0808-05-9BE21-LZ	9eg_1e_4gb	REV05	4GB	128MB	NA	1 mm connectors	NA
TE0808-05-9BE81-A	9eg_1e_4gb	REV05	4GB	128MB	NA	NA	NA
TE0808-05-9BI41-X	9eg_1i_8gb	REV05	8GB	128MB	NA	NA	Single Die DDR; U41 replaced with schottky diodes
TE0808-05-9GI21-A	9eg_2i_4gb	REV05	4GB	128MB	NA	NA	NA
TE0808-05-9GI21-AK	9eg_2i_4gb	REV05	4GB	128MB	NA	NA	NA
TE0808-05-9GI21-AZ	9eg_2i_4gb	REV05	4GB	128MB	NA	NA	NA
TE0808-05-9GI21-C	9eg_2i_4gb	REV05	4GB	128MB	NA	NA	SoC without encryption
TE0808-05-9GI21-KZ	9eg_2i_4gb	REV05	4GB	128MB	NA	CAO	CAO

TE0808-05-BBE21-A	15eg_1e_4gb	REV05	4GB	128MB	NA	NA	NA
TE0808-05-BBE21-AK	15eg_1e_4gb	REV05	4GB	128MB	NA	NA	NA
TE0808-05-BBE21-AZ	15eg_1e_4gb	REV05	4GB	128MB	NA	NA	NA
TE0808-05-BBE21-L	15eg_1e_4gb	REV05	4GB	128MB	NA	1 mm connectors	NA
TE0808-05-BBE81-A	15eg_1e_4gb	REV05	4GB	128MB	NA	NA	NA
TE0808-05-BBE81-E	15eg_1e_4gb	REV05	4GB	128MB	NA	NA	NA
TE0808-05-BBE81-EK	15eg_1e_4gb	REV05	4GB	128MB	NA	NA	NA
TE0808-05-S001	9eg_1e_8gb_D	REV05	8GB	128MB	NA	CAO	CAO;Single Die DDR
TE0808-05-S002	15eg_1e_4gb	REV05	4GB	128MB	NA	CAO	CAO
TE0808-05-S003	15eg_1e_4gb	REV05	4GB	128MB	NA	CAO	CAO
TE0808-05-S004	9eg_2i_4gb	REV05	4GB	128MB	NA	CAO	CAO
TE0808-05-S005	9eg_2i_4gb	REV05	4GB	128MB	NA	CAO	CAO
TE0808-05-S006	9eg_2i_4gb	REV05	4GB	128MB	NA	CAO	CAO
TE0808-05-S007	9eg_1e_4gb	REV05	4GB	128MB	NA	NA	CAO
TE0808-05-S014	9eg_1e_4gb	REV05	4GB	128MB	NA	NA	CAO
TE0808-05-S016	9eg_1e_4gb	REV05	4GB	128MB	NA	CAO	CAO
TE0808-05-S018	9eg_2e_4gb	REV05	4GB	128MB	NA	CAO	CAO
TE0808-05-S019	9eg_2e_4gb	REV05	4GB	128MB	NA	CAO	CAO
TE0808-05-S020	9eg_2i_4gb	REV05	4GB	128MB	NA	CAO	CAO
TE0808-05-S021	9eg_2i_4gb	REV05	4GB	128MB	NA	CAO	CAO
TE0808-05-S022	6cg_1e_4gb	REV05	4GB	128MB	NA	CAO	CAO
TE0808-05-S025	6eg_1e_4gb_D	REV05	4GB	128MB	NA	CAO	CAO
TE0808-05-S026	9eg_2i_4gb	REV05	4GB	128MB	NA	CAO:Si5345 not assembled	CAO: without PLL
TE0808-05-S027	9eg_2i_4gb	REV05	4GB	128MB	NA	CAO	CAO
TE0808-05-S029	9eg_2i_4gb	REV05	4GB	128MB	NA	CAO	CAO
TE0808-05-S033	15eg_1e_4gb	REV05	4GB	128MB	NA	NA	CAO
TE0808-05-S035	15eg_1e_4gb	REV05	4GB	128MB	NA	CAO	CAO
TE0808-05-S036	15eg_1e_4gb	REV05	4GB	128MB	NA	CAO	CAO
TE0808-05-S038	9eg_1e_4gb	REV05	4GB	128MB	NA	CAO	CAO
TE0808-05-S039	6eg_1e_4gb	REV05	4GB	128MB	NA	CAO	CAO: without PLL

TE0808-05-S041	6eg_1e_4gb_D	REV05	4GB	128MB	NA	CAO	CAO

*used as reference

Hardware Modules

Note: Design contains also Board Part Files for TE0808 only configuration, this board part files are not used for this reference design.

Design supports following carriers:

Carrier Model	Notes
TEBF0808*	Used as reference carrier. **Important**: CPLD Firmware REV07 or newer is recommended

*used as reference

Hardware Carrier

Additional HW Requirements:

Additional Hardware	Notes
Display Port Monitor	Optional HW Not all monitors are supported, also Adapter to other Standard can make trouble. Design was tested with **DELL U2412M**
USB Keyboard	Optional HW Can be used to get access to console which is show on Display Port
USB Stick	Optional HW USB was tested with USB memory stick
SATA Disk	Optional HW
PCIe Card	Optional HW
ETH cable	Optional HW Ethernet works with DHCP, but can be setup also manually
SD card	with fat32 partition

*used as reference

Additional Hardware

Content

For general structure and 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
SI5345	<project folder>/misc/PLL/	SI5345 Project with current PLL Configuration
init.sh	<project folder>/misc/sd/	Additional initialization script for Linux

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:

- TE0808 "StarterKit" 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:

- AMD 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
    >
    > Important: Use Board Part Files, which ends with *_tebf0808

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
    b. Generate Programming Files

    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

For basic board setup, LEDs... see: TEBF0808 Getting Started

Programming

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

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.

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_te0808`

3. Set Boot Mode to **QSPI-Boot**
    - Depends on Carrier, see carrier TRM.
    - TEBF0808 change automatically the Boot Mode to SD, if SD is inserted, optional CPLD Firmware without Boot Mode changing for microSD Slot is available on the download area

SD-Boot mode

1. Copy **image.ub**, **boot.scr** 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 with TEBF0808) Insert PCIe Card (detection depends on Linux driver. Only some basic drivers are installed)
5. (Optional with TEBF0808) Connect SATA Disc
6. (Optional with TEBF0808) Connect Display Port Monitor (List of usable Monitors: https://www.xilinx.com/support/answers/68671.html)
7. (Optional with TEBF0808) Connect Network Cable
8. 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 0 Bus, replace 0 with other bus
number is also possible)
dmesg | grep rtc        (RTC check)
udhcpc                                (ETH0 check)
lsusb                                (USB check)
lspci              (PCIe check)
```

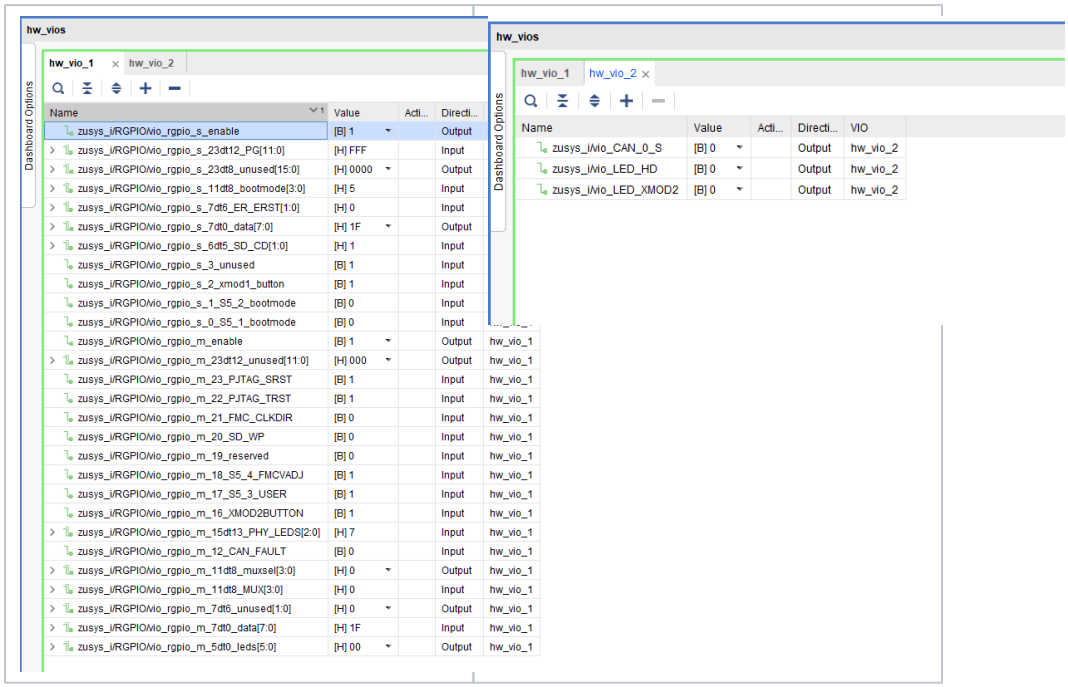
4. Option Features

- 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")

Vivado Hardware Manager

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

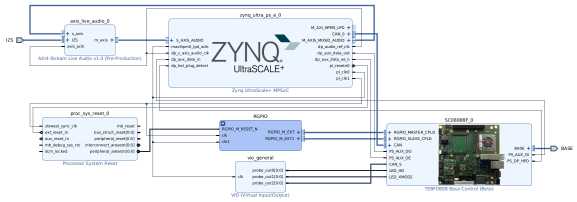
- RGPIO Interface (**Important**: CPLD Firmware REV07 or newer is needed) for Control and Monitoring:
    - Set Enable to send Write date over RGPIO interface.
        - **Important use CPLD Firmware REV07 or newer:** https://wiki.trenz-electronic.de/display/PD/TEBF0808+CPLD
            - Buttons, LEDs, Status...
- Control:
    - LEDs: XMOD 2 (without green dot) and HD LED are accessible.
    - CAN_S



Vivado Hardware Manager

System Design - Vivado

Block Design



Block Design

PS Interfaces

Activated interfaces:

Type	Note
DDR	
QSPI	MIO
SD0	MIO
SD1	MIO
CAN0	EMIO
I2C0	MIO
PJTAG0	MIO
UART0	MIO
GPIO0	MIO
SWDT0..1	
TTC0..3	
GEM3	MIO
USB0	MIO/GTP
PCIe	MIO/GTP
SATA	GTP
Display Port	EMIO/GTP

PS Interfaces

Constrains

Basic module constrains

_i_bitgen.xdc

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

Design specific constrain

_i_io.xdc

```
#System Controller IP
  #LED_HD SC0 J3:31
  #LED_XMOD SC17 J3:48
  #CAN RX SC19 J3:52 B47_L2_P in
  #CAN TX SC18 J3:50 B47_L2_N out
  #CAN S  SC16 J3:46 B47_L3_N out
#HDIO_SC1    K14
#HDIO_SC2    H13
#HDIO_SC3    H14
#HDIO_SC4    F13



#HDIO_SC0    J14
set_property PACKAGE_PIN J14 [get_ports BASE_sc0]
#HDIO_SC5    G13
set_property PACKAGE_PIN G13 [get_ports BASE_sc5]
#HDIO_SC6    J15
set_property PACKAGE_PIN J15 [get_ports BASE_sc6]
#HDIO_SC7    K15
set_property PACKAGE_PIN K15 [get_ports BASE_sc7]
#HDIO_SC10   A15
set_property PACKAGE_PIN A15 [get_ports BASE_sc10_io]
#HDIO_SC11   B15
set_property PACKAGE_PIN B15 [get_ports BASE_sc11]
#HDIO_SC12   C13
set_property PACKAGE_PIN C13 [get_ports BASE_sc12]
#HDIO_SC13   C14
set_property PACKAGE_PIN C14 [get_ports BASE_sc13]
#HDIO_SC14   E13
set_property PACKAGE_PIN E13 [get_ports BASE_sc14]
#HDIO_SC15   E14
set_property PACKAGE_PIN E14 [get_ports BASE_sc15]
#HDIO_SC16   A13
set_property PACKAGE_PIN A13 [get_ports BASE_sc16]
#HDIO_SC17   B13
set_property PACKAGE_PIN B13 [get_ports BASE_sc17]
#HDIO_SC18   A14
set_property PACKAGE_PIN A14 [get_ports BASE_sc18]
#HDIO_SC19   B14
set_property PACKAGE_PIN B14 [get_ports BASE_sc19]
set_property IOSTANDARD LVCMOS18 [get_ports BASE_sc0]
set_property IOSTANDARD LVCMOS18 [get_ports BASE_sc5]
set_property IOSTANDARD LVCMOS18 [get_ports BASE_sc6]
set_property IOSTANDARD LVCMOS18 [get_ports BASE_sc7]
set_property IOSTANDARD LVCMOS18 [get_ports BASE_sc10_io]
set_property IOSTANDARD LVCMOS18 [get_ports BASE_sc11]
```

```
set_property IOSTANDARD LVCMOS18 [get_ports BASE_sc12]
set_property IOSTANDARD LVCMOS18 [get_ports BASE_sc13]
set_property IOSTANDARD LVCMOS18 [get_ports BASE_sc14]
set_property IOSTANDARD LVCMOS18 [get_ports BASE_sc15]
set_property IOSTANDARD LVCMOS18 [get_ports BASE_sc16]
set_property IOSTANDARD LVCMOS18 [get_ports BASE_sc17]
set_property IOSTANDARD LVCMOS18 [get_ports BASE_sc18]
set_property IOSTANDARD LVCMOS18 [get_ports BASE_sc19]


# Audio Codec
#LRCLK                  J3:49 B47_L9_N
#BCLK                  J3:51 B47_L9_P
#DAC_SDATA          J3:53 B47_L7_N
#ADC_SDATA          J3:55 B47_L7_P

#LRCLK G14
set_property PACKAGE_PIN G14 [get_ports I2S_lrclk ]
#BCLK G15
set_property PACKAGE_PIN G15 [get_ports I2S_bclk ]
#DAC_SDATA E15
set_property PACKAGE_PIN E15 [get_ports I2S_sdin ]
#ADC_SDATA F15
set_property PACKAGE_PIN F15 [get_ports I2S_sdout ]
set_property IOSTANDARD LVCMOS18 [get_ports I2S_lrclk ]
set_property IOSTANDARD LVCMOS18 [get_ports I2S_bclk ]
set_property IOSTANDARD LVCMOS18 [get_ports I2S_sdin ]
set_property IOSTANDARD LVCMOS18 [get_ports I2S_sdout ]
```

Software Design - Vitis

For Vitis project creation, follow instructions from:

Vitis

Application

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

zynqmp_fsbl

TE modified 2023.2 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_*
    - Si5345 Configuration
    - OTG+PCIe Reset over MIO

- I2C MUX for EEPROM MAC

zynqmp_pmufw

Xilinx default PMU firmware.

hello_te0808

Hello TE0808 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:

- select SD default instead of eMMC:
  - CONFIG_SUBSYSTEM_PRIMARY_SD_PSU_SD_1_SELECT=y
- add new flash partition for bootscr and sizing
  - CONFIG_SUBSYSTEM_FLASH_PSU_QSPI_0_BANKLESS_PART0_SIZE=0xA00000
  - 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="TE0808_TEBF0808"

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_NVMEM=y
  - CONFIG_DM_RTC=y (needed for nvmem driver because of bug in uboot)
- 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_BOOT_SCRIPT_OFFSET=0x2A40000
- Identification
  - CONFIG_IDENT_STRING=" TE0808_TEBF0808"

Change platform-top.h:

```
#no changes
```

Device Tree

project-spec\meta-user\recipes-bsp\device-tree\files\system-user.dtsi

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


/*------------------ gtr --------------------*/

//https://xilinx-wiki.atlassian.net/wiki/spaces/A/pages/18841716
/Zynq+Ultrascale+MPSOC+Linux+SIOU+driver

/ {
  refclk3:psgtr_dp_clock {
          compatible = "fixed-clock";
          #clock-cells = <0x00>;
          clock-frequency = <27000000>;
  };

  refclk2:psgtr_pcie_usb_clock {
          compatible = "fixed-clock";
          #clock-cells = <0x00>;
          clock-frequency = <100000000>;
  };

  refclk1:psgtr_sata_clock {
          compatible = "fixed-clock";
          #clock-cells = <0x00>;
          clock-frequency = <150000000>;
  };

  //refclk0:psgtr_unused_clock {
  //        compatible = "fixed-clock";
  //        #clock-cells = <0x00>;
  //        clock-frequency = <100000000>;
  //};
};

&psgtr {
  clocks = <&refclk1 &refclk2 &refclk3>;
  /* ref clk instances used per lane */
  clock-names = "ref1\0ref2\0ref3";
};



/*------------------ SD --------------------*/
&sdhci0 {
    // disable-wp;
    no-1-8-v;
};
```

```
&sdhci1 {
    // disable-wp;
    no-1-8-v;
};


/*------------------- USB --------------------*/
&dwc3_0 {
    status = "okay";
    dr_mode = "host";
    snps,usb3_lpm_capable;
    snps,dis_u3_susphy_quirk;
    snps,dis_u2_susphy_quirk;
    phy-names = "usb2-phy","usb3-phy";
    maximum-speed = "super-speed";
};


/*------------------ ETH PHY --------------------*/
&gem3 {
    /delete-property/ local-mac-address;
    phy-handle = <&phy0>;

    nvmem-cells = <&eth0_addr>;
    nvmem-cell-names = "mac-address";

    phy0: phy0@1 {
        device_type = "ethernet-phy";
        reg = <1>;
    };
};


/*----------------- SATA PHY --------------------*/
&sata {

     ceva,p0-burst-params = <0x13084a06>;
     ceva,p0-cominit-params = <0x18401828>;
     ceva,p0-comwake-params = <0x614080e>;
     ceva,p0-retry-params = <0x96a43ffc>;
     ceva,p1-burst-params = <0x13084a06>;
     ceva,p1-cominit-params = <0x18401828>;
     ceva,p1-comwake-params = <0x614080e>;
     ceva,p1-retry-params = <0x96a43ffc>;

};


/*-------------------- 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>;

        spi-rx-bus-width = <4>;
        spi-tx-bus-width = <4>;
```

```
        spi-max-frequency = <90000000>;
    };
};


/*------------------ I2C --------------------*/
&i2c0 {
    i2cswitch@73 { // u
        compatible = "nxp,pca9548";
        #address-cells = <1>;
        #size-cells = <0>;
        reg = <0x73>;
        i2c-mux-idle-disconnect;
        i2c@0 { // MCLK TEBF0808 SI5338A, 570FBB000290DG_unassembled
            reg = <0>;
        };
        i2c@1 { // SFP TEBF0808 PCF8574DWR
            reg = <1>;
        };
        i2c@2 { // PCIe
            reg = <2>;
        };
        i2c@3 { // SFP1 TEBF0808
            reg = <3>;
        };
        i2c@4 {// SFP2 TEBF0808
            reg = <4>;
        };
        i2c@5 { // TEBF0808 EEPROM
            reg = <5>;
            eeprom: eeprom@50 {
                compatible = "microchip,24aa025", "atmel,24c02";
                reg = <0x50>;

                #address-cells = <1>;
                #size-cells = <1>;
                eth0_addr: eth-mac-addr@FA {
                  reg = <0xFA 0x06>;
                };
            };
        };
        i2c@6 { // TEBF0808 FMC
            reg = <6>;
        };
        i2c@7 { // TEBF0808 USB HUB
            reg = <7>;
        };
    };
    i2cswitch@77 { // u
        compatible = "nxp,pca9548";
        reg = <0x77>;
        i2c-mux-idle-disconnect;
        i2c@0 { // TEBF0808 PMOD P1
            reg = <0>;
        };
        i2c@1 { // i2c Audio Codec
            reg = <1>;
            /*
            adau1761: adau1761@38 {
                compatible = "adi,adau1761";
                reg = <0x38>;
```

```
            };
            */
        };
        i2c@2 { // TEBF0808 Firefly A
            reg = <2>;
        };
        i2c@3 { // TEBF0808 Firefly B
            reg = <3>;
        };
        i2c@4 { //Module PLL Si5338 or SI5345
            reg = <4>;
        };
        i2c@5 { //TEBF0808 CPLD
            reg = <5>;
        };
        i2c@6 { //TEBF0808 Firefly PCF8574DWR
            reg = <6>;
        };
        i2c@7 { // TEBF0808 PMOD P3
            reg = <7>;
        };
    };
};
```

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_VERBOSE_ERRORS is not set
  - # CONFIG_NVME_HWMON is not set
  - # CONFIG_NVME_AUTH 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
  - # CONFIG_NVME_TARGET_AUTH is not set
  - CONFIG_SATA_AHCI=y
  - CONFIG_SATA_MOBILE_LPM_POLICY=0

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_imagefeature-serial-autologin-root=y

FSBL patch (alternative for vitis fsbl trenz patch)

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

> ⚠ te_* files are identical to files in "<project folder>\sw_lib\sw_apps\zynqmp_fsbl\src" except for the PLL files (SI5345) which depend on PLL revision. The PLL files may have to be copied again manually into the appropriate petalinux folder "<project folder>\os\petalinux\project-spec\meta-user\recipes-bsp\embeddedsw\fsbl-firmware\git\lib\sw_apps\zynqmp_fsbl\src"
> Petalinux Troubleshoot#Petalinux2023.2

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

SI5345

File location "<project folder>/misc/PLL/Si5345_*/Si5345-*.slabtimeproj"

General documentation how you work with these project will be available on Si5345

Appx. 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
			• 2023.2 release • new assembly variants

Error rendering macro 'page-info' Ambiguous method overloading for method jdk.proxy279.$Proxy4022#hasContentLevelPermission. Cannot resolve which method to invoke for [null, class java.lang.String, class com.atlassian.confluence.pages.Page]	**Error rendering macro 'page-info'** Ambiguous method overloading for method jdk.proxy279.$Proxy4022#hasContentLevelPermission. Cannot resolve which method to invoke for [null, class java.lang.String, class com.atlassian.confluence.pages.Page]	**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.

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]	
2023-06-13	v.58	Manuela Strücker	• added chapter FSBL patch • added alternative generation of BOOT.bin in Petalinux (chapter Design flow)
2023-06-01	v.56	Manuela Strücker	• 2022.2 release • new assembly variants
2023-04-13	v.55	Manuela Strücker	• script update • new assembly variants
2022-09-29	v.53	Manuela Strücker	• new assembly variants
2022-09-29	v.51	Manuela Strücker	• update board part files compatible to Vivado 2021.2.1
2022-09-06	v.50	Manuela Strücker	• typo
2022-03.16	v.48	Manuela Strücker	• 2021.2 release • update board files
2022-02-03	v.47	John Hartfiel	• Typo correction on key features section
2021-07-15	v.46	Manuela Strücker	• Document Style update
2021-05-12	v.44	John Hartfiel	• update board files • update design

2021-02-05	v.43	John Hartfiel	• 2020.2 release • document style update
2020-11-06	v.41	John Hartfiel	• typo bugfix for programming part
2020-09-29	v.40	John Hartfiel	• new assembly variants
2020-03-25	v.37	John Hartfiel	• script update
2020-02-25	v.35	John Hartfiel	• Update requirement section
2020-01-23	v.34	John Hartfiel	• new assembly variants • Release 2019.2
2019-08-09	v.32	John Hartfiel	• new assembly variants • small FSBL update • minor document style update
2019-05-07	v.29	John Hartfiel	• Release 2018.3
2018-08-09	v.27	John Hartfiel	• Release 2018.2
2018-05-25	v.21	John Hartfiel	• Solved known issues
2018-04-30	v.19	John Hartfiel	• Update known issues
2018-03-29	v.18	John Hartfiel	• New assembly variant
2018-02-08	v.16	John Hartfiel	• Solved known issues
2018-01-29	v.10	John Hartfiel	• Update known issues
2018-01-18	v.8	John Hartfiel	• Update documentation only
2018-01-17	v.7	John Hartfiel	• Update design

2018-01-15	v.4	John Hartfiel	• Release 2017.4
2017-12-20	v.2	John Hartfiel	• Release 2017.2
	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 . 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

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]