TE0808 StarterKit

Revision v.32

Exported on 2020-01-22

Online version of this document:

https://wiki.trenz-electronic.de/display/PD/TE0808+StarterKit

1 Table of Contents

2 Table of Figures

3 Table of Tables

4 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.

4.1 Key Features

- Vivado 2018.3
- TEBF0808
- Linux
- USB
- ETH
- MAC from EEPROM
- PCIe
- SATA
- SD
- I2C
- RGPIO
- DP
- user LED access
- Modified FSBL for Si5338 programming
- Special FSBL for QSPI Programming

4.2 Revision History

Date	Vivado	Project Built	Authors	Description
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

Date	Vivado	Project Built	Authors	Description
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

Date	Vivado	Project Built	Authors	Description
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

Table 1: Design Revision History

4.3 Release Notes and Know Issues

Issues	Description	Workaround/Solution	To be fixed version
Flash access on Linux	Device tree is not correct on Linux	add compatibility to "compatible “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 nessecary: 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

Table 2: Known Issues

4.4 Requirements

4.4.1 Software

Software	Version	Note
Vivado	2018.3	needed
SDK	2018.3	needed
PetaLinux	2018.3	needed

Software	Version	Note
SI ClockBuilder Pro	---	optional

Table 3: Software

4.4.2 Hardware

Basic description of TE Board Part Files is available on TE Board Part Files.[1]

Complete List is available on <design name>/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	Slower DDR Speed
TE0808-2ES2	2es2_2gb	REV04|REV03	2GB	64MB	NA	NA	Slower DDR Speed
TE0808-04-09EG-1EA	9eg_1e_2gb	REV04	2GB	64MB	NA	NA	
TE0808-04-09EG-1EB	9eg_1e_4gb	REV04	4GB	64MB	NA	NA	
TE0808-04-09EG-1ED	9eg_1e_4gb	REV04	4GB	64MB	NA	1 mm connectors	
TE0808-04-09EG-2IB	9eg_2i_4gb	REV04	4GB	64MB	NA	NA	
TE0808-04-15EG-1EB	15eg_1e_4gb	REV04	4GB	64MB	NA	NA	

[1] https://wiki.trenz-electronic.de/display/PD/TE+Board+Part+Files

Module Model	Board Part Short Name	PCB Revision Support	DDR	QSPI Flash	EMMC	Others	Notes
TE0808-04-09EG-1EE	9eg_1e_4gb	REV04	4GB	128MB	NA	NA	
TE0808-04-09EG-1EL	9eg_1e_4gb	REV04	4GB	128MB	NA	1 mm connectors	
TE0808-04-09EG-2IE	9eg_2i_4gb	REV04	4GB	128MB	NA	NA	
TE0808-04-15EG-1EE	15eg_1e_4gb	REV04	4GB	128MB	NA	NA	
TE0808-04-06EG-1EE	6eg_1e_4gb	REV04	4GB	128MB	NA	NA	
TE0808-04-06EG-1E3	6eg_1e_4gb	REV04	4GB	128MB	NA	1 mm connectors	
TE0808-04-6GI21-L	6eg_2i_4gb	REV04	4GB	128MB	NA	1 mm connectors	
TE0808-04-6GI21-A	6eg_2i_4gb	REV04	4GB	128MB	NA	NA	
TE0808-04-6BI21-A	6eg_1i_4gb	REV04	4GB	128MB	NA	NA	
TE0808-04-9GI21-A	9eg_2i_4gb	REV04	4GB	128MB	NA	NA	
TE0808-04-9BE21-A	9eg_1e_4gb	REV04	4GB	128MB	NA	NA	
TE0808-04-6BE21-L	6eg_1e_4gb	REV04	4GB	128MB	NA	1 mm connectors	

Module Model	Board Part Short Name	PCB Revision Support	DDR	QSPI Flash	EMMC	Others	Notes
TE0808-04-6BE21-A	6eg_1e_4gb	REV04	4GB	128MB	NA	NA	
TE0808-04-9BE21-L	9eg_1e_4gb	REV04	4GB	128MB	NA	1 mm connectors	
TE0808-04-BBE21-A	15eg_1e_4gb	REV04	4GB	128MB	NA	NA	

Table 4: Hardware Modules

Note: Design contains also Board Part Files for TE0808 only configuration, this boart 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

Table 5: Hardware Carrier

Additional HW Requirements:

Additional Hardware	Notes

Table 6: Additional Hardware

4.5 Content

For general structure and of the reference design, see Project Delivery - Xilinx devices[2]

[2] https://wiki.trenz-electronic.de/display/PD/Project+Delivery+-+Xilinx+devices

4.5.1 Design Sources

Type	Location	Notes
Vivado	<design name>/block_design <design name>/constraints <design name>/ip_lib	Vivado Project will be generated by TE Scripts
SDK/HSI	<design name>/sw_lib	Additional Software Template for SDK/HSI and apps_list.csv with settings for HSI
PetaLinux	<design name>/os/petalinux	PetaLinux template with current configuration

Table 7: Design sources

4.5.2 Additional Sources

Type	Location	Notes
SI5345	<design name>/misc/Si5345	SI5345 Project with current PLL Configuration
init.sh	<design name>/sd/	Additional Initialization Script for Linux

Table 8: Additional design sources

4.5.3 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
DebugProbes-File	*.ltx	Definition File for Vivado/Vivado Labtools Debugging Interface
Diverse Reports	---	Report files in different formats

File	File-Extension	Description
Hardware-Platform-Specification-Files	*.hdf	Exported Vivado Hardware Specification for SDK/HSI 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

Table 9: Prebuilt files (only on ZIP with prebult content)

4.5.4 Download

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

Reference Design is available on:

- TE0808 "StarterKit" Reference Design[3]

[3] https://shop.trenz-electronic.de/Download/?path=Trenz_Electronic/Modules_and_Module_Carriers/5.2x7.6/TE0808/Reference_Design/2018.3/StarterKit

5 Design Flow

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

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

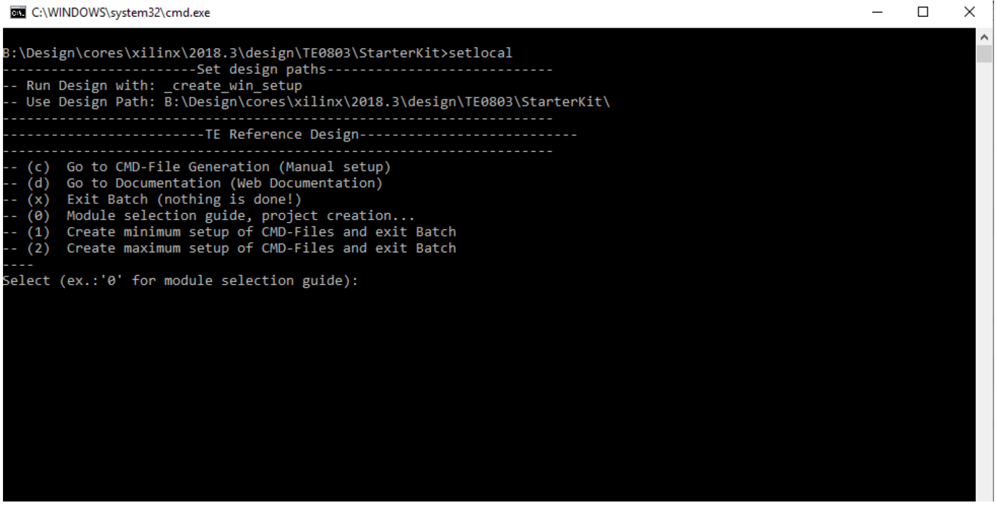
See also:

- Xilinx Development Tools[4]
- Vivado Projects - TE Reference Design[5]
- Project Delivery.[6]

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/SDK GUI. For currently Scripts limitations on Win and Linux OS see: Project Delivery Currently limitations of functionality[7]

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



```
B:\Design\cores\xilinx\2018.3\design\TE0803\StarterKit>setlocal
------------------------Set design paths----------------------------
-- Run Design with: _create_win_setup
-- Use Design Path: B:\Design\cores\xilinx\2018.3\design\TE0803\StarterKit\
--------------------------------------------------------------------
-------------------------TE Reference Design--------------------------
--------------------------------------------------------------------
-- (c)  Go to CMD-File Generation (Manual setup)
-- (d)  Go to Documentation (Web Documentation)
-- (x)  Exit Batch (nothing is done!)
-- (0)  Module selection guide, project creation...
-- (1)  Create minimum setup of CMD-Files and exit Batch
-- (2)  Create maximum setup of CMD-Files and exit Batch
----
Select (ex.:'0' for module selection guide):
```

2. Press 0 and enter to start "Module Selection Guide"
3. (optional Win OS) Generate Virtual Drive or use short directory for the reference design (for example x:\<design name>)
4. Create Project (follow instruction of the product selection guide), settings file will be configured automatically during this process
5. Create Project
   a. (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 TE Board Part Files[8]
      Important: Use Board Part Files, which ends with *_tebf0808
6. Create HDF and export to prebuilt folder

[4] https://wiki.trenz-electronic.de/display/PD/Xilinx+Development+Tools#XilinxDevelopmentTools-XilinxSoftware-BasicUserGuides

[5] https://wiki.trenz-electronic.de/display/PD/Vivado+Projects+-+TE+Reference+Design

[6] https://wiki.trenz-electronic.de/display/PD/Project+Delivery+-+Xilinx+devices

[7] https://wiki.trenz-electronic.de/display/PD/Project+Delivery+-+Xilinx+devices#ProjectDelivery-Xilinxdevices-Currentlylimitationsoffunctionality

[8] https://wiki.trenz-electronic.de/display/PD/TE+Board+Part+Files

   a. Run on Vivado TCL: TE::hw_build_design -export_prebuilt
      Note: Script generate design and export files into \prebuilt\hardware\<short dir>. Use GUI is the same, except file export to prebuilt folder

7. Create Linux (bl31.elf, uboot.elf and image.ub) with exported HDF
   a. HDF is exported to "prebuilt\hardware\<short name>"
      Note: HW Export from Vivado GUI create another path as default workspace.
      Create Linux images on VM, see PetaLinux KICKstart[9]
      i. Use TE Template from /os/petalinux
         Note: run init_config.sh before you start petalinux config. This will set correct temporary path variable.
8. Add Linux files (bl31.elf, uboot.elf and image.ub) to prebuilt folder
   a. prebuilt\os\petalinux\<ddr size>" or "prebuilt\os\petalinux\<short name>"
9. Generate Programming Files with HSI/SDK
   a. Run on Vivado TCL: TE::sw_run_hsi
      Note: Scripts generate applications and bootable files, which are defined in "sw_lib\apps_list.csv"
   b. (alternative) Start SDK with Vivado GUI or start with TE Scripts on Vivado TCL: TE::sw_run_sdk
      Note: See SDK Projects[10]

[9] https://wiki.trenz-electronic.de/display/PD/PetaLinux+KICKstart

[10] https://wiki.trenz-electronic.de/display/PD/SDK+Projects

6 Launch

For basic board setup, LEDs... see: TEBF0808 Getting Started[11]

6.1 Programming

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

Xilinx documentation for programming and debugging: Vivado/SDK/SDSoC-Xilinx Software Programming and Debugging[12]

6.1.1 QSPI

Optional for Boot.bin on QSPI Flash and image.ub on SD.

- Connect JTAG and power on carrier with module
- Open Vivado Project with "vivado_open_existing_project_guimode.cmd" or if not created, create with "vivado_create_project_guimode.cmd"
- Type on Vivado TCL Console: TE::pr_program_flash_binfile -swapp u-boot
  Note: To program with SDK/Vivado GUI, use special FSBL (zynqmp_fsbl_flash) on setup
  optional "TE::pr_program_flash_binfile -swapp hello_te0803" possible
- Copy image.ub on SD-Card
  - For correct prebuilt file location, see <design_name>/prebuilt/readme_file_location.txt
- Set Boot Mode to QSPI-Boot and insered SD.
  - Depends on Carrier, see carrier TRM.
  - TEBF0808 change automatically the Boot Mode to SD, if SD is insered, optional CPLD Firmware without Boot Mode changing for mircoSD Slot is available on the download area

6.1.2 SD

1. Copy image.ub and Boot.bin on SD-Card.
   - For correct prebuilt file location, see <design_name>/prebuilt/readme_file_location.txt
2. Set Boot Mode to SD-Boot.
   - Depends on Carrier, see carrier TRM.
3. Insert SD-Card in SD-Slot.

6.1.3 JTAG

Not used on this Example.

6.2 Usage

1. Prepare HW like described on section TE0808 StarterKit#Programming(see page 16)
2. Connect UART USB (JTAG XMOD)

[11] https://wiki.trenz-electronic.de/display/PD/TEBF0808+Getting+Started

[12] https://wiki.trenz-electronic.de/display/PD/Xilinx+Development+Tools#XilinxDevelopmentTools-XilinxSoftwareProgrammingandDebugging

3. Select SD Card as Boot Mode (or QSPI - depending on step 1)
   Note: See TRM of the Carrier, which is used.
4. (Optional) Insert PCIe Card (detection depends on Linux driver. Only some basic drivers are installed)
5. (Optional) Connect Sata Disc
6. (Optional) Connect DisplayPort Monitor (List of usable Monitors: https://www.xilinx.com/support/answers/68671.html)
7. (Optional) Connect Network Cable
8. Power On PCB
   Note: 1. ZynqMP Boot ROM loads PMU Firmware and FSBL from SD into OCM, 2. FSBL loads ATF(bl31.elf) and U-boot from SD/QSPI into DDR, 3. U-boot load Linux from SD into DDR.

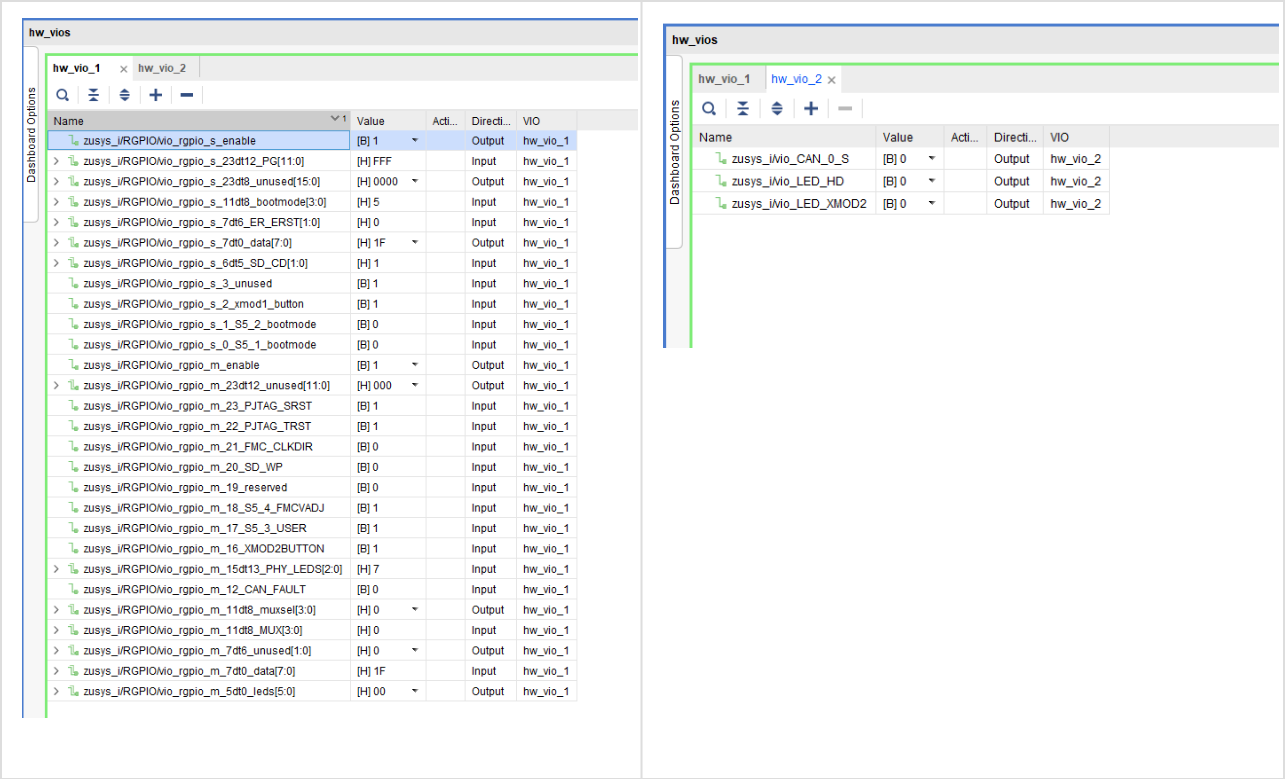
6.2.1 Linux

1. Open Serial Console (e.g. putty)
   a. Speed: 115200
   b. COM Port: Win OS, see device manager, Linux OS see dmesg |grep tty (UART is *USB1)
2. Linux Console:
   Note: Wait until Linux boot finished For Linux Login use:
   a. User Name: root
   b. Password: root
3. You can use Linux shell now.
   a. I2C 0 Bus type: i2cdetect -y -r 0
   b. ETH0 works with udhcpc
   c. USB type "lsusb" or connect USB device
   d. PCIe type "lspci"
4. Option Features
   a. Webserver to get access to Zynq
      i. insert IP on web browser to start web interface
   b. init.sh scripts
      i. add init.sh script on SD, content will be load automatically on startup (template included in ./misc/SD)

6.2.2 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



Name	Value	Acti...	Directi...	VIO
zusys_i/RGPIO/vio_rgpio_s_enable	[B] 1		Output	hw_vio_1
zusys_i/RGPIO/vio_rgpio_s_23dt12_PG[11:0]	[H] FFF		Input	hw_vio_1
zusys_i/RGPIO/vio_rgpio_s_23dt8_unused[15:0]	[H] 0000		Output	hw_vio_1
zusys_i/RGPIO/vio_rgpio_s_11dt8_bootmode[3:0]	[H] 5		Input	hw_vio_1
zusys_i/RGPIO/vio_rgpio_s_7dt6_ER_ERST[1:0]	[H] 0		Input	hw_vio_1
zusys_i/RGPIO/vio_rgpio_s_7dt0_data[7:0]	[H] 1F		Output	hw_vio_1
zusys_i/RGPIO/vio_rgpio_s_6dt5_SD_CD[1:0]	[H] 1		Input	hw_vio_1
zusys_i/RGPIO/vio_rgpio_s_3_unused	[B] 1		Input	hw_vio_1
zusys_i/RGPIO/vio_rgpio_s_2_xmod1_button	[B] 1		Input	hw_vio_1
zusys_i/RGPIO/vio_rgpio_s_1_S5_2_bootmode	[B] 0		Input	hw_vio_1
zusys_i/RGPIO/vio_rgpio_s_0_S5_1_bootmode	[B] 0		Input	hw_vio_1
zusys_i/RGPIO/vio_rgpio_m_enable	[B] 1		Output	hw_vio_1
zusys_i/RGPIO/vio_rgpio_m_23dt12_unused[11:0]	[H] 000		Output	hw_vio_1
zusys_i/RGPIO/vio_rgpio_m_23_PJTAG_SRST	[B] 1		Input	hw_vio_1
zusys_i/RGPIO/vio_rgpio_m_22_PJTAG_TRST	[B] 1		Input	hw_vio_1
zusys_i/RGPIO/vio_rgpio_m_21_FMC_CLKDIR	[B] 0		Input	hw_vio_1
zusys_i/RGPIO/vio_rgpio_m_20_SD_WP	[B] 0		Input	hw_vio_1
zusys_i/RGPIO/vio_rgpio_m_19_reserved	[B] 0		Input	hw_vio_1
zusys_i/RGPIO/vio_rgpio_m_18_S5_4_FMCVADJ	[B] 1		Input	hw_vio_1
zusys_i/RGPIO/vio_rgpio_m_17_S5_3_USER	[B] 1		Input	hw_vio_1
zusys_i/RGPIO/vio_rgpio_m_16_XMOD2BUTTON	[B] 1		Input	hw_vio_1
zusys_i/RGPIO/vio_rgpio_m_15dt13_PHY_LEDS[2:0]	[H] 7		Input	hw_vio_1
zusys_i/RGPIO/vio_rgpio_m_12_CAN_FAULT	[B] 0		Input	hw_vio_1
zusys_i/RGPIO/vio_rgpio_m_11dt8_muxsel[3:0]	[H] 0		Output	hw_vio_1
zusys_i/RGPIO/vio_rgpio_m_11dt8_MUX[3:0]	[H] 0		Input	hw_vio_1
zusys_i/RGPIO/vio_rgpio_m_7dt6_unused[1:0]	[H] 0		Output	hw_vio_1
zusys_i/RGPIO/vio_rgpio_m_7dt0_data[7:0]	[H] 1F		Input	hw_vio_1
zusys_i/RGPIO/vio_rgpio_m_5dt0_leds[5:0]	[H] 00		Output	hw_vio_1

Name	Value	Acti...	Directi...	VIO
zusys_i/vio_CAN_0_S	[B] 0		Output	hw_vio_2
zusys_i/vio_LED_HD	[B] 0		Output	hw_vio_2
zusys_i/vio_LED_XMOD2	[B] 0		Output	hw_vio_2

Table 10: Vivado Hardware Manager

7 System Design - Vivado

7.1 Block Design

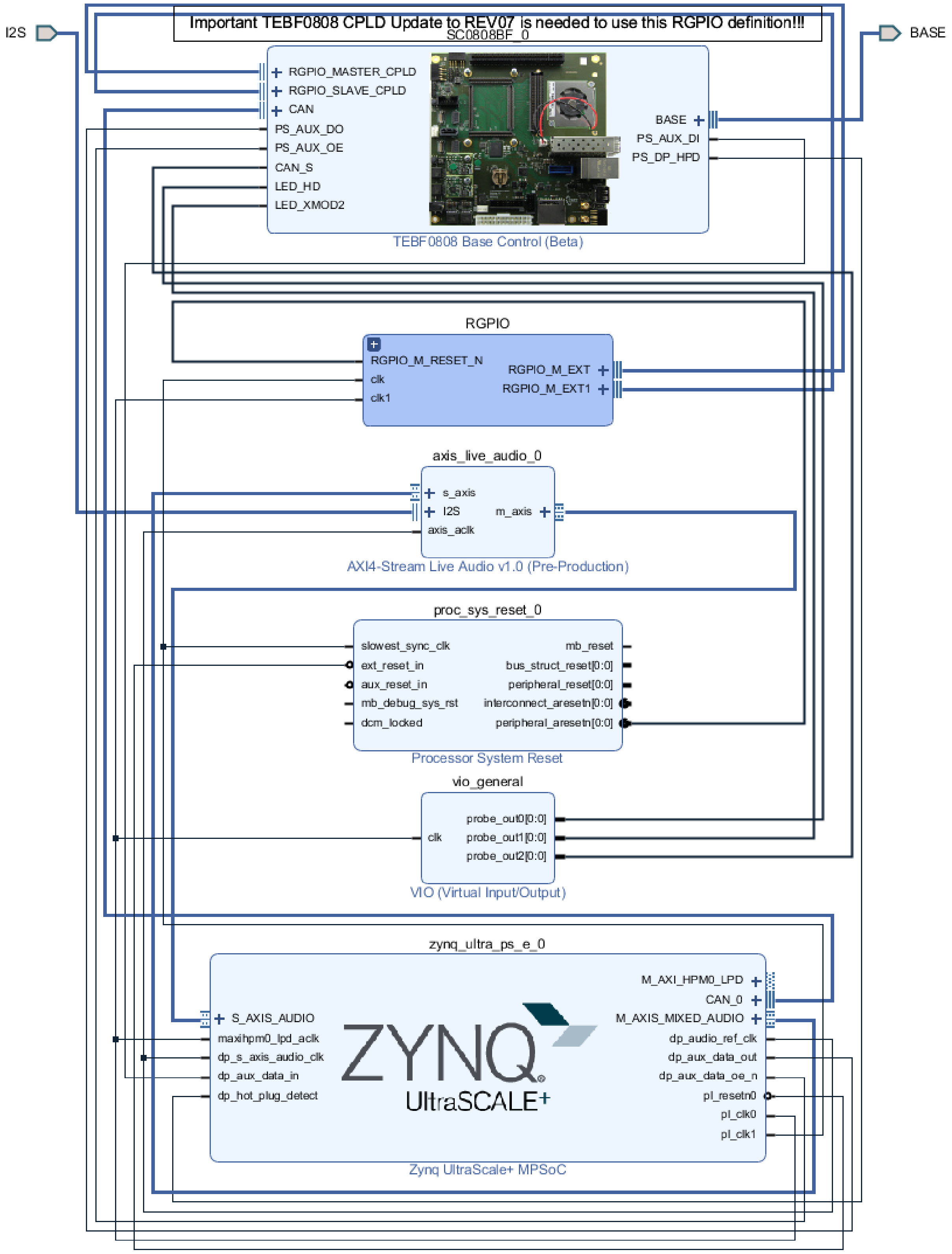


Figure 1: Block Design

7.1.1 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
DisplayPort	EMIO/GTP

Table 11: PS Interfaces

7.2 Constrains

7.2.1 Basic module constrains

_i_bitgen.xdc

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

7.2.2 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
set_property PACKAGE_PIN J14 [get_ports BASE_sc0]
set_property PACKAGE_PIN G13 [get_ports BASE_sc5]
set_property PACKAGE_PIN J15 [get_ports BASE_sc6]
set_property PACKAGE_PIN K15 [get_ports BASE_sc7]
set_property PACKAGE_PIN A15 [get_ports BASE_sc10_io]
set_property PACKAGE_PIN B15 [get_ports BASE_sc11]
set_property PACKAGE_PIN C13 [get_ports BASE_sc12]
set_property PACKAGE_PIN C14 [get_ports BASE_sc13]
set_property PACKAGE_PIN E13 [get_ports BASE_sc14]
set_property PACKAGE_PIN E14 [get_ports BASE_sc15]
set_property PACKAGE_PIN A13 [get_ports BASE_sc16]
set_property PACKAGE_PIN B13 [get_ports BASE_sc17]
set_property PACKAGE_PIN A14 [get_ports BASE_sc18]
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]

# PLL
#set_property PACKAGE_PIN AH6 [get_ports {si570_clk_p[0]}]
#set_property IOSTANDARD LVDS [get_ports {si570_clk_p[0]}]
#set_property IOSTANDARD LVDS [get_ports {si570_clk_n[0]}]
# Clocks
#set_property PACKAGE_PIN J8 [get_ports {B229_CLK1_clk_p[0]}]
#set_property PACKAGE_PIN F25 [get_ports {B128_CLK0_clk_p[0]}]
# SFP
#set_property PACKAGE_PIN G8 [get_ports {B230_CLK0_clk_p}]
# B230_RX3_P
#set_property PACKAGE_PIN A4 [get_ports {SFP1_rxp}]
# B230_TX3_P
#set_property PACKAGE_PIN A8 [get_ports {SFP1_txp}]
```

```
# B230_RX2_P
#set_property PACKAGE_PIN B2 [get_ports {SFP2_rxp}]
# B230_TX2_P
#set_property PACKAGE_PIN B6 [get_ports {SFP2_txp}]

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

8 Software Design - SDK/HSI

For SDK project creation, follow instructions from:

SDK Projects[13]

8.1 Application

SDK template in ./sw_lib/sw_apps/ available.

8.1.1 zynqmp_fsbl

TE modified 2018.3 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)\n\
- 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

8.1.2 zynqmp_fsbl_flash

TE modified 2018.3 FSBL

General:

- Modified Files: xfsbl_initialisation.c, xfsbl_hw.h, xfsbl_handoff.c, xfsbl_main.c
- General Changes:
    - Display FSBL Banner
    - Set FSBL Boot Mode to JTAG
    - Disable Memory initialisation

8.1.3 zynqmp_pmufw

Xilinx default PMU firmware.

8.1.4 hello_te0808

Hello TE0808 is a Xilinx Hello World example as endless loop instead of one console output.

[13] https://wiki.trenz-electronic.de/display/PD/SDK+Projects

8.1.5 u-boot

U-Boot.elf is generated with PetaLinux. SDK/HSI is used to generate Boot.bin.

9 Software Design - PetaLinux

For PetaLinux installation and project creation, follow instructions from:

- PetaLinux KICKstart[14]

9.1 Config

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

Activate:

- CONFIG_SUBSYSTEM_PRIMARY_SD_PSU_SD_1_SELECT=y
- CONFIG_SUBSYSTEM_ETHERNET_PSU_ETHERNET_3_MAC=""

9.2 U-Boot

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

- CONFIG_ENV_IS_NOWHERE=y
- CONFIG_ENV_IS_IN_SPI_FLASH is not set

Change platform-top.h:

[14] https://wiki.trenz-electronic.de/display/PD/PetaLinux+KICKstart

```
#include <configs/platform-auto.h>
#define CONFIG_SYS_BOOTM_LEN 0xF000000

#define DFU_ALT_INFO_RAM \
                "dfu_ram_info=" \
        "setenv dfu_alt_info " \
        "image.ub ram $netstart 0x1e00000\0" \
        "dfu_ram=run dfu_ram_info && dfu 0 ram 0\0" \
        "thor_ram=run dfu_ram_info && thordown 0 ram 0\0"

#define DFU_ALT_INFO_MMC \
        "dfu_mmc_info=" \
        "set dfu_alt_info " \
        "${kernel_image} fat 0 1\\\\;" \
        "dfu_mmc=run dfu_mmc_info && dfu 0 mmc 0\0" \
        "thor_mmc=run dfu_mmc_info && thordown 0 mmc 0\0"

/*Required for uartless designs */
#ifndef CONFIG_BAUDRATE
#define CONFIG_BAUDRATE 115200
#ifdef CONFIG_DEBUG_UART
#undef CONFIG_DEBUG_UART
#endif
#endif

/*Define CONFIG_ZYNQMP_EEPROM here and its necessaries in u-boot menuconfig if you
had EEPROM memory. */
#define CONFIG_ZYNQMP_EEPROM
#ifdef CONFIG_ZYNQMP_EEPROM
#define CONFIG_SYS_I2C_EEPROM_ADDR_LEN  1
#define CONFIG_CMD_EEPROM
#define CONFIG_ZYNQ_EEPROM_BUS          0
#define CONFIG_ZYNQ_GEM_EEPROM_ADDR     0x50
#define CONFIG_ZYNQ_GEM_I2C_MAC_OFFSET  0xFA
#endif
```

9.3 Device Tree

```
/include/ "system-conf.dtsi"
/ {
};
/* notes:
serdes: // PHY TYP see: dt-bindings/phy/phy.h
*/

/* default */

/* SD */

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

};

/*PCIE*/
&pcie {
     phys = <&lane0 2 0 2 100000000>;  //not recognized at the moment on linux
};

/* DP */
&zynqmp_dpsub {
    phys = <&lane3 5 0 3 27000000>; //Xilinx default is 5 (UFS), 6 (DP) does not work
};

/* SATA */

&sata {
    phys = <&lane2 1 0 1 150000000>;  //TE0808,TE0807
    //phys = <&lane2 1 0 0 150000000>; //TE0803
};


/* 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";
    phys = <&lane1 4 0 2 100000000>;
    maximum-speed = "super-speed";
};

/* ETH PHY */
```

```
&gem3 {
    phy-handle = <&phy0>;
    phy0: phy0@1 {
        device_type = "ethernet-phy";
        reg = <1>;
    };
};

/* 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>;
    };
};

/* I2C */

&i2c0 {
    i2cswitch@73 { // u
        compatible = "nxp,pca9548";
        #address-cells = <1>;
        #size-cells = <0>;
        reg = <0x73>;
        i2c-mux-idle-disconnect;

        i2c@2 { // PCIe
            #address-cells = <1>;
            #size-cells = <0>;
            reg = <2>;
        };
        i2c@3 { // i2c SFP
            #address-cells = <1>;
            #size-cells = <0>;
            reg = <3>;
        };
        i2c@4 { // i2c SFP
            #address-cells = <1>;
            #size-cells = <0>;
            reg = <4>;
        };
        i2c@5 { // i2c EEPROM
            #address-cells = <1>;
            #size-cells = <0>;
            reg = <5>;
        };
        i2c@6 { // i2c FMC
            #address-cells = <1>;
            #size-cells = <0>;
            reg = <6>;
            si570_2: clock-generator3@5d {
                #clock-cells = <0>;
                compatible = "silabs,si570";
                reg = <0x5d>;
                temperature-stability = <50>;
                factory-fout = <156250000>;
                clock-frequency = <78800000>;
            };
        };
        i2c@7 { // i2c USB HUB
            #address-cells = <1>;
            #size-cells = <0>;
            reg = <7>;
        };
    };
    i2cswitch@77 { // u
        compatible = "nxp,pca9548";
        #address-cells = <1>;
        #size-cells = <0>;
        reg = <0x77>;
        i2c-mux-idle-disconnect;
        i2c@0 { // i2c PMOD
            #address-cells = <1>;
            #size-cells = <0>;
            reg = <0>;
        };
        i2c@1 { // i2c Audio Codec
            #address-cells = <1>;
            #size-cells = <0>;
            reg = <1>;
            /*
            adau1761: adau1761@38 {
                compatible = "adi,adau1761";
                reg = <0x38>;
            };
            */
        };
        i2c@2 { // i2c FireFly A
            #address-cells = <1>;
            #size-cells = <0>;
            reg = <2>;
        };
        i2c@3 { // i2c FireFly B
            #address-cells = <1>;
            #size-cells = <0>;
            reg = <3>;
        };
        i2c@4 { // i2c PLL
            #address-cells = <1>;
            #size-cells = <0>;
            reg = <4>;
        };
        i2c@5 { // i2c SC
            #address-cells = <1>;
            #size-cells = <0>;
            reg = <5>;
        };
        i2c@6 { // i2c
            #address-cells = <1>;
            #size-cells = <0>;
            reg = <6>;
        };
        i2c@7 { // i2c
            #address-cells = <1>;
            #size-cells = <0>;
            reg = <7>;
        };
    };
};
```

9.4 Kernel

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

Changes:

- CONFIG_CPU_IDLE is not set (only needed to fix JTAG Debug issue)
- CONFIG_CPU_FREQ is not set (only needed to fix JTAG Debug issue)
- CONFIG_CPU_FREQ_DEFAULT_GOV_USERSPACE is not set (only needed to fix JTAG Debug issue)
- CONFIG_EDAC_CORTEX_ARM64=y

9.5 Rootfs

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

Changes:

- CONFIG_busybox-httpd=y (for web server app)
- CONFIG_packagegroup-petalinux-utils(util-linux,cpufrequtils,bridge-utils,mtd-utils,usbutils,pciutils,canutils,i2c-tools,smartmontools,e2fsprogs)

9.6 Applications

9.6.1 startup

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

See: \os\petalinux\project-spec\meta-user\recipes-apps\startup\files

9.6.2 webfwu

Webserver application accemble for Zynq access. Need busybox-httpd

10 Additional Software

10.1 SI5345

File location <design name>/misc/Si5345/Si5345-*.slabtimeproj

General documentation how you work with these project will be available on Si5345[15]

[15] https://wiki.trenz-electronic.de/display/PD/Si5345

11 Appx. A: Change History and Legal Notices

11.1 Document Change History

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

Date	Document Revision	Authors	Description
2019-08-09	v.32(see page 6)	John Hartfiel[16]	• 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

[16] https://wiki.trenz-electronic.de/display/~j.hartfiel

Date	Document Revision	Authors	Description
	All	John Hartfiel[17]	

Table 12: Document change history.

11.2 Legal Notices

11.3 Data Privacy

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

11.4 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.

11.5 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.

11.6 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.

11.7 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.

[17] https://wiki.trenz-electronic.de/display/~j.hartfiel

11.8 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.

11.9 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[18]. 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[19] 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)[20].

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.

2019-06-07

[18] http://guidance.echa.europa.eu/
[19] https://echa.europa.eu/candidate-list-table
[20] http://www.echa.europa.eu/