# Table of Contents

Overview .................................................................................................................. 5  
Key Features ............................................................................................................. 5  
Revision History ....................................................................................................... 5  
Release Notes and Know Issues ............................................................................... 5  
Requirements .......................................................................................................... 5  
  Software ................................................................................................................ 5  
  Hardware ............................................................................................................... 6  
Content ..................................................................................................................... 6  
  Design Sources .................................................................................................... 7  
  Additional Sources ............................................................................................... 7  
  Prebuilt ............................................................................................................... 7  
  Download ........................................................................................................... 7  
Design Flow .............................................................................................................. 8  
Launch ...................................................................................................................... 10  
  Programming ....................................................................................................... 10  
    QSPI ............................................................................................................... 10  
    SD ............................................................................................................... 10  
    JTAG ........................................................................................................... 10  
  Usage .................................................................................................................. 10  
    Linux ............................................................................................................. 11  
    Vivado HW Manager: .................................................................................... 11  
System Design - Vivado .......................................................................................... 12  
  Block Design .................................................................................................... 12  
  Constrains ......................................................................................................... 13  
    Basic module constrains ............................................................................. 13  
    Design specific constrain ........................................................................... 13  
Software Design - SDK/HSI .................................................................................. 15  
  Application ......................................................................................................... 15  
    SCU ............................................................................................................. 15  
    SREC SPI BootLoader .................................................................................... 15  
    xilisf_v5_8 ................................................................................................. 15  
    U-Boot ........................................................................................................ 15  
Software Design - PetaLinux .................................................................................. 16  
  Config ............................................................................................................... 16  
  U-Boot ............................................................................................................. 16  
  Device Tree ...................................................................................................... 17  
  Kernel .............................................................................................................. 17  
  Rootfs ............................................................................................................. 17  
  Applications ................................................................................................. 17  
Additional Software ............................................................................................... 18  
  SI5338 ......................................................................................................... 18  
Appx. A: Change History and Legal Notices ............................................................. 19
| Document Change History                          | 19 |
| Legal Notices                                      | 19 |
| Data privacy                                       | 19 |
| Document Warranty                                  | 19 |
| Limitation of Liability                            | 20 |
| Copyright Notice                                   | 20 |
| Technology Licenses                                | 20 |
| Environmental Protection                           | 20 |
| REACH, RoHS and WEEE                               | 21 |
Overview

Key Features

- PetaLinux
- MicroBlaze
- SREC
- I2C
- Flash
- MIG
- FMeter
- SI5338 initialisation with MCS
- ETH

Revision History

<table>
<thead>
<tr>
<th>Date</th>
<th>Vivado</th>
<th>Project Built</th>
<th>Authors</th>
<th>Description</th>
</tr>
</thead>
</table>
| 2017-12-15 | 2017.2 | te0712-test_board-vivado_2017.2-build_07_20171215172447.zip  
te0712-test_board_noprebuild-vivado_2017.2-build_07_20171215172514.zip | John Hartfiel | • Add SI5338 initialisation with MCS  
• Add Ethernet IP |
| 2017-11-07 | 2017.2 | te0712-test_board-vivado_2017.2-build_05_20171107172917.zip  
te0712-test_board_noprebuild-vivado_2017.2-build_05_20171107172939.zip | John Hartfiel | • Add Wiki Link in Boart Part Files  
• Set Correct Short Link for te0712-02-200-2c |
| 2017-10-05 | 2017.2 | te0712-test_board-vivado_2017.2-build_03_20171005082148.zip  
te0712-test_board_noprebuild-vivado_2017.2-build_03_20171005082225.zip | John Hartfiel | • Initial release |

Release Notes and Know Issues

<table>
<thead>
<tr>
<th>Issues</th>
<th>Description</th>
<th>Workaround</th>
<th>To be fixed version</th>
</tr>
</thead>
<tbody>
<tr>
<td>No known issues</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
</tbody>
</table>

Requirements

Software

<table>
<thead>
<tr>
<th>Software</th>
<th>Version</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vivado</td>
<td>2017.2</td>
<td>needed</td>
</tr>
</tbody>
</table>
Hardware

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

Complete List is available on <design name>/board_files/*_board_files.csv

Design supports following modules:

<table>
<thead>
<tr>
<th>Module Model</th>
<th>Board Part Short Name</th>
<th>PCB Revision Support</th>
<th>DDR</th>
<th>QSPI Flash</th>
<th>Others</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>te0712-02-35-2i</td>
<td>35_2i</td>
<td>REV02</td>
<td>1GB</td>
<td>32MB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>te0712-02-100-1i</td>
<td>100_1i</td>
<td>REV01, REV02</td>
<td>1GB</td>
<td>32MB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>te0712-02-100-2c</td>
<td>100_2c</td>
<td>REV01, REV02</td>
<td>1GB</td>
<td>32MB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>te0712-02-100-2c3</td>
<td>100_2c</td>
<td>REV01, REV02</td>
<td>1GB</td>
<td>32MB</td>
<td>2,5 mm connector</td>
<td></td>
</tr>
<tr>
<td>te0712-02-200-1i</td>
<td>200_1i</td>
<td>REV01, REV02</td>
<td>1GB</td>
<td>32MB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>te0712-02-200-1i3</td>
<td>200_1i</td>
<td>REV01, REV02</td>
<td>1GB</td>
<td>32MB</td>
<td>2,5 mm connector</td>
<td></td>
</tr>
<tr>
<td>te0712-02-200-2i</td>
<td>200_2i</td>
<td>REV01, REV02</td>
<td>1GB</td>
<td>32MB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>te0712-02-200-2c</td>
<td>200_2c</td>
<td>REV01, REV02</td>
<td>1GB</td>
<td>32MB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>te0712-02-200-2c3</td>
<td>200_2c</td>
<td>REV01, REV02</td>
<td>1GB</td>
<td>32MB</td>
<td>2,5 mm connector</td>
<td></td>
</tr>
</tbody>
</table>

Design supports following carriers:

<table>
<thead>
<tr>
<th>Carrier Model</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>TE0701</td>
<td></td>
</tr>
<tr>
<td>TE0703</td>
<td>used as reference carrier</td>
</tr>
<tr>
<td>TE0705</td>
<td></td>
</tr>
<tr>
<td>TE0706</td>
<td></td>
</tr>
<tr>
<td>TEBA0841</td>
<td></td>
</tr>
</tbody>
</table>

Additional HW Requirements:

<table>
<thead>
<tr>
<th>Additional Hardware</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>USB Cable for JTAG/UART</td>
<td>Check Carrier Board and Programmer for correct typ</td>
</tr>
<tr>
<td>XMOD Programmer</td>
<td>Carrier Board dependent, only if carrier has no own FTDI</td>
</tr>
</tbody>
</table>

Content

For general structure and of the reference design, see Project Delivery
Design Sources

<table>
<thead>
<tr>
<th>Type</th>
<th>Location</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vivado</td>
<td>&lt;design name&gt;/block_design</td>
<td>Vivado Project will be generated by TE Scripts</td>
</tr>
<tr>
<td></td>
<td>&lt;design name&gt;/constraints</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&lt;design name&gt;/ip_lib</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&lt;design name&gt;/firmware</td>
<td></td>
</tr>
<tr>
<td>SDK/HSI</td>
<td>&lt;design name&gt;/sw_lib</td>
<td>Additional Software Template for SDK/HSI and</td>
</tr>
<tr>
<td></td>
<td></td>
<td>apps_list.csv with settings for HSI</td>
</tr>
<tr>
<td>PetaLinux</td>
<td>&lt;design name&gt;/os/petalinux</td>
<td>PetaLinux template with current configuration</td>
</tr>
</tbody>
</table>

Additional Sources

<table>
<thead>
<tr>
<th>Type</th>
<th>Location</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>SI5338 Project</td>
<td>\misc\SI5338</td>
<td></td>
</tr>
</tbody>
</table>

Prebuilt

<table>
<thead>
<tr>
<th>File</th>
<th>File-Extension</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BIT-File</td>
<td>*.bit</td>
<td>FPGA (PL Part) Configuration File</td>
</tr>
<tr>
<td>DebugProbes-File</td>
<td>*.ltx</td>
<td>Definition File for Vivado/Vivado Labtools Debugging Interface</td>
</tr>
<tr>
<td>Diverse Reports</td>
<td>---</td>
<td>Report files in different formats</td>
</tr>
<tr>
<td>Hardware-Platform-Specification-Files</td>
<td>*.hdf</td>
<td>Exported Vivado Hardware Specification for SDK/HSI and PetaLinux</td>
</tr>
<tr>
<td>LabTools Project-File</td>
<td>*.lpr</td>
<td>Vivado Labtools Project File</td>
</tr>
<tr>
<td>MCS-File</td>
<td>*.mcs</td>
<td>Flash Configuration File with Boot-Image (MicroBlaze or FPGA part only)</td>
</tr>
<tr>
<td>MMI-File</td>
<td>*.mmi</td>
<td>File with BRAM-Location to generate MCS or BIT-File with *.elf content (MicroBlaze only)</td>
</tr>
<tr>
<td>OS-Image</td>
<td>*.ub</td>
<td>Image with Linux Kernel (On Petalinux optional with Devicetree and RAM-Disk)</td>
</tr>
<tr>
<td>Software-Application-File</td>
<td>*.elf</td>
<td>Software Application for Zynq or MicroBlaze Processor Systems</td>
</tr>
<tr>
<td>SREC-File</td>
<td>*.srec</td>
<td>Converted Software Application for MicroBlaze Processor Systems</td>
</tr>
</tbody>
</table>

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:

- TE0712 Reference Design Download Area
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: Vivado/SDK/SDSoC#XilinxSoftware-BasicUserGuides

- Vivado/SDK/SDSoC#XilinxSoftware-BasicUserGuides
- Vivado Projects
- 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/SDK GUI. For currently Scripts limitations on Win and Linux OS see: Project Delivery Currently limitations of functionality

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

2. Press 0 and enter for minimum setup
3. (optional Win OS) Generate Virtual Drive or use short directory for the reference design (for example x:\design name)
4. Create Project
   a. 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
5. Create HDF and export to prebuilt folder
   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
6. Create Linux (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.
   b. Create Linux images on VM, see PetaLinux KICKstart
      i. Use TE Template from /os/petalinux
         Note: run init_config.sh before you start petalinux config. This will set correct temporary path variable.
         Important Note: Select correct Flash partition offset on petalinux-config: Subsystem Auto HW Settings Flash Settings, FPGA+Boot+bootenv=0x900000 (increase automatically generate Boot partition)
7. Add Linux files (uboot.elf and image.ub) to prebuilt folder
   a. "prebuilt\os\petalinux\default" or "prebuilt\os\petalinux\<short name>"
      Notes: Scripts select "prebuilt\os\petalinux\<short name>", if exist, otherwise "prebuilt\os\petalinux\default"
8. Generate UBoot SREC:
   a. Create SDK Project with TE Scripts on Vivado TCL: TE::sw_runSdk
   b. Create "uboot-dummy" application
      Note: Use Hello World Example
   c. Copy u.boot.elf into "workspace\sdk\uboot-dummy\Debug"
   d. Open "uboot-dummy" properties C/C++ Build Settings and go into Build Steps Tab.
   e. Add to Post-build steps: mb-objcopy -O srec u-boot.elf u-boot.srec
   f. Press Apply or regenerate project
      Note: src is generated on "workspace\sdk\uboot-dummy\Debug\u-boot.srec"
9. Generate MCS Firmware (optional):
   a. Create SDK Project with TE Scripts on Vivado TCL: TE::sw_runSdk
   b. Create "SCU" application
      Note: Select MCS Microblaze and SCU Application
   c. Select Release Built
   d. Regenerate App
10. 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
11. Copy "\prebuilt\software\<short name>\src_spi_bootloader.elf" into "\firmware\microblaze_0."
12. (optional) Copy "\workspace\sdk\scu\Release\scu.elf" into "\firmware\microblaze_mcs_0."
13. Regenerate Vivado Project or Update Bitfile only with "src_spi_bootloader.elf" and "scu.elf"
Launch

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

QSPI

1. Connect JTAG and power on PCB
2. (if not done) Select correct device and Xilinx install path on "design_basic_settings.cmd" and create Vivado project with "vivado_create_project_guimode.cmd" or open with "vivado_open_project_guimode.cmd", if generated.
3. Type on Vivado Console: TE::pr_program_flash_mcsfile -swapp u-boot
   Note: Alternative use SDK or setup Flash on Vivado manually
4. Reboot (if not done automatically)

SD

Not used on this Example.

JTAG

Not used on this Example.

Usage

1. Prepare HW like described on section Programming
2. Connect UART USB (most cases same as JTAG)
3. Power on PCB
   Note: FPGA Loads Bitfile from Flash, MCS Firmware configure SI5338 and starts Microblaze, SREC Bootloader from Bitfile Firmware loads U-Boot into DDR (This takes a while), U-boot loads Linux from QSPI Flash into DDR

Boot process takes a while, please wait.
1. Open Serial Console (e.g. putty)
   a. Speed: 9600
   b. COM Port: Win OS, see device manager, Linux OS see dmesg |grep tty (UART is *USB1)

2. Linux Console:
   - Note: Wait until Linux boot finished For Linux Login use:
     a. User Name: root
     b. Password: root
   - You can use Linux shell now.
     a. ETH0 works with udhcpc

**Vivado HW Manager:**

1. Open Vivado HW-Manager and add VIO signal to dashboard (*.ltx located on prebuilt folder).
   a. Set radix from VIO signals (MGT REF, MIG_OUT, CLK1B, CLK0) to unsigned integer.
      - Note: Frequency Counter is inaccurate and displayed unit is Hz
   b. MGT REF~125MHz, MIG_50MHZ~50MHz., CLK1B ~50MHz, CLK0~100MHz
   c. Additional Infos: System reset from MCS and GIO outputs
Constrains

Basic module constrains

_i_bitgen_common.xdc

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

_i_bitgen.xdc

set_property BITSTREAM.CONFIG.UNUSEDPIN PULLDOWN [current_design]

Design specific constrain

_i_reset.xdc

set_property PULLDOWN true [get_ports reset]

_i_io.xdc

#I2C
set_property PACKAGE_PIN W21 [get_ports pll_i2c_scl_io]
set_property IOSTANDARD LVCMOS33 [get_ports pll_i2c_scl_io]
set_property PACKAGE_PIN T20 [get_ports pll_i2c_sda_io]
set_property IOSTANDARD LVCMOS33 [get_ports pll_i2c_sda_io]

#Reset
set_property PACKAGE_PIN T3 [get_ports reset]
set_property IOSTANDARD LVCMOS15 [get_ports reset]

#CLKS
set_property PACKAGE_PIN R4 [get_ports {CLK1B[0]}]
set_property IOSTANDARD SSTL15 [get_ports {CLK1B[0]}]
set_property PACKAGE_PIN K4 [get_ports {CLK0_clk_p[0]}]
set_property IOSTANDARD DIFF_SSTL15 [get_ports {CLK0_clk_p[0]}]

#ETH PHY
set_property PACKAGE_PIN N17 [get_ports phy_rst_n]
set_property IOSTANDARD LVCMOS33 [get_ports phy_rst_n]
_i_timing.xdc

create_clock -period 8.000 -name mgt_clk0_clk_p -waveform {0.000 4.000} [get_ports mgt_clk0_clk_p]

create_clock -period 10.000 -name {CLK0_clk_p[0]} -waveform {0.000 5.000} [get_ports {CLK0_clk_p [0]}]
create_clock -period 20.000 -name msys_i/axi_quad_spi_0/NO_DUAL_QUAD_MODE.QSPI_NORMAL/QSPI_LEGACY_MD_GEN.QSPI_CORE_INTERFACE_I/LOGIC_FOR_MD_12_GEN.SCK_MISO_STARTUP_USED.QSPI_STARTUP_BLOCK_I/cfgmclk -waveform {0.000 7.576} [get_pins msys_i/axi_quad_spi_0/NO_DUAL_QUAD_MODE.QSPI_NORMAL/QSPI_LEGACY_MD_GEN.QSPI_CORE_INTERFACE_I/LOGIC_FOR_MD_12_GEN.SCK_MISO_STARTUP_USED.QSPI_STARTUP_BLOCK_I/cfgmclk]

set_false_path -from [get_clocks {CLK0_clk_p[0]}] -to [get_clocks clk_pll_i]
set_false_path -from [get_clocks mgt_clk0_clk_p] -to [get_clocks clk_pll_i]
set_false_path -from [get_clocks msys_i/axi_quad_spi_0/NO_DUAL_QUAD_MODE.QSPI_NORMAL/QSPI_LEGACY_MD_GEN.QSPI_CORE_INTERFACE_I/LOGIC_FOR_MD_12_GEN.SCK_MISO_STARTUP_USED.QSPI_STARTUP_BLOCK_I/cfgmclk] -to [get_clocks clk_pll_i]

set_false_path -from [get_clocks -of_objects [get_pins msys_i/mig_7series_0/u_msys_mig_7series_0_0_mig/u_ddr3_infrastructure/gen_ui_extra_clocks.mmmcm_i/CLKFBOUT]] -to [get_clocks mgt_clk0_clk_p]
set_false_path -from [get_pins msys_i/labtools_fmeter_0/U0/COUNTER_REFCLK_inst/bl.DSP48E_2/CLK] -to [get_pins {msys_i/vio_0/inst/PROBE_IN_INST/probe_in_reg_reg[*]/D}]
set_false_path -from [get_pins {msys_i/labtools_fmeter_0/U0/F_reg[*]/C}] -to [get_pins {msys_i/vio_0/inst/PROBE_IN_INST/probe_in_reg_reg[*]/D}]

http://www.trenz-electronic.de
Software Design - SDK/HSI

For SDK project creation, follow instructions from:

SDK Projects

Application

SCU
MCS Firmware to configure SI5338 and Reset System.
Template location: \sw_lib\sw_apps\scu

SREC SPI BootLoader
Add some Console outputs and changed Bootloader Read Address.
Template location: \sw_lib\sw_apps\srec_spi_bootloader

xilisf_v5_8
Changed default Flash Typ to 5.
Template location: \sw_lib\sw_services

U-Boot
U-Boot.elf is generated with PetaLinux. SDK/HSI is used to generate u-boot.srec. Vivado to generate *.mcs
Software Design - PetaLinux

- PetaLinux KICKstart

Description currently not available.

Config

- Set kernel flash Address to 0x900000 and Kernel size to 0xA00000:
  (--> Subsystem Auto Hardware Settings --> Flash Settings)
  - SUBSYSTEM_FLASH_AXI_QUAD_SPI_0_BANKLESS_PART0_SIZE = 0x400000
  - SUBSYSTEM_FLASH_AXI_QUAD_SPI_0_BANKLESS_PART1_SIZE = 0x4E0000
  - SUBSYSTEM_FLASH_AXI_QUAD_SPI_0_BANKLESS_PART2_SIZE = 0x20000
  - SUBSYSTEM_FLASH_AXI_QUAD_SPI_0_BANKLESS_PART3_SIZE = 0xA00000

U-Boot

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

#undef CONFIG_PHY_XILINX
#undef XILINX_EMACLITE_BASEADDR 0x40E00000
#undef CONFIG_MII
#undef CONFIG_PHY_GIGE
#undef CONFIG_PHY_MARVELL
#undef CONFIG_PHY_NATSEMI
#undef CONFIG_NET_MULTI
#undef CONFIG_BOOTP_MAY_FAIL
#undef CONFIG_NETCONSOLE 1
#undef CONFIG_SERVERIP 192.168.150.117
#undef CONFIG_IPADDR

/* PREBOOT */
#define CONFIG_PREBOOT "echo U-BOOT for petalinux;setenv preboot; echo; 
```
Device Tree

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

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

Kernel
No changes.

Rootfs
No changes.

Applications
No changes.
Additional Software

SI5338

Download ClockBuilder Desktop for SI5338

1. Install and start ClockBuilder
2. Select SI5338
3. Options  Open register map file
   Note: File location <design name>/misc/SI5338/RegisterMap.txt
4. Modify settings
5. Options  save C code header files
6. Replace Header files from FSBL template with generated file
Appx. A: Change History and Legal Notices

Document Change History

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

<table>
<thead>
<tr>
<th>Date</th>
<th>Document Revision</th>
<th>Authors</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2018-09-06</td>
<td>v.30</td>
<td>John Hartfiel</td>
<td>Add SCU sourc path</td>
</tr>
<tr>
<td>2017-12-15</td>
<td>v.15</td>
<td>John Hartfiel</td>
<td>Update Design and Description</td>
</tr>
<tr>
<td>2017-11-07</td>
<td>v.11</td>
<td>John Hartfiel</td>
<td>Update Design Files</td>
</tr>
<tr>
<td>2017-10-06</td>
<td>v.10</td>
<td>John Hartfiel</td>
<td>small Document Update</td>
</tr>
<tr>
<td>2017-10-05</td>
<td>v.8</td>
<td>John Hartfiel</td>
<td>Release 2017.2</td>
</tr>
<tr>
<td>2017-09-11</td>
<td>v.1</td>
<td>John Hartfiel</td>
<td>Initial release</td>
</tr>
<tr>
<td>All</td>
<td></td>
<td>John Hartfiel</td>
<td></td>
</tr>
</tbody>
</table>

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


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.

2018-09-18