# TE0724 Test Board

## Table of contents



## Overview

Zynq PS Design with Linux Example and Virtual Input/Output (VIO) for Control and Monitoring with Vivado HW-Manager.
Wiki Resources page: https://wiki.trenz.org/te0724-info

## Key Features

- Vitis/Vivado 2022.2
- PetaLinux
- SD
- ETH
- MAC from EEPROM
- I2C
- RTC
- FMeter
- FSBL to enable I2C Buffer for PMIC (RTC) and external I2C

## Revision History

| Date | Vivado | Project Built | Authors | Description |
|---|---|---|---|---|
| 2023-08-11 | 2022.2 | TE0724-test_board-vivado_2022.2-build_4_20230811130817.zip<br>TE0724-test_board_noprebuilt-vivado_2022.2-build_4_20230811130817.zip | Manuela Strücker | • Release Vivado 2022.2<br>• new assembly variants |
| 2022-11-16 | 2021.2.1 | TE0724-test_board-vivado_2021.2-build_20_20221119073924.zip<br>TE0724-test_board_noprebuilt-vivado_2021.2-build_20_20221119073924.zip | Manuela Strücker | • Release Vivado 2021.2.1<br>• script update<br>• new assembly variants |
| 2020-03-25 | 2019.2 | TE0724-test_board-vivado_2019.2-build_8_20200325075929.zip<br>TE0724-test_board_noprebuilt-vivado_2019.2-build_8_20200325075950.zip | John Hartfiel | • script update |
| 2020-01-30 | 2019.2 | TE0724-test_board_noprebuilt-vivado_2019.2-build_4_20200130130053.zip<br>TE0724-test_board-vivado_2019.2-build_4_20200130130040.zip | John Hartfiel | • 2019.2 update<br>• Vitis support<br>• FSBL changes<br>• petalinux device tree and u-boot update |

| | | | | |
|---|---|---|---|---|
| 2019-13-12 | 2018.2 | TE0724-test_board_noprebuilt-vivado_2018.2-build_04_20191212064015.zip<br>TE0724-test_board-vivado_2018.2-build_04_20191212064001.zip | John Hartfiel | • bugfix IO constrains |
| 2019-06-13 | 2018.2 | TE0724-test_board-vivado_2018.2-build_04_20190613114927.zip<br>TE0724-test_board_noprebuilt-vivado_2018.2-build_04_20190613115049.zip | Oleksandr Kiyenko, John Hartfiel | • add app to get access to EEPROM U10 |
| 2019-02-04 | 2018.2 | TE0724-test_board-vivado_2018.2-build_04_20190204111543.zip<br>TE0724-test_board_noprebuilt-vivado_2018.2-build_04_20190204111557.zip | John Hartfiel | • Important Board Part File Update<br>  ◦ change DDR3 to DDR3 Low Power |
| 2018-08-29 | 2018.2 | TE0724-test_board_noprebuilt-vivado_2018.2-build_03_20180830170634.zip<br>TE0724-test_board-vivado_2018.2-build_03_20180830170621.zip | John Hartfiel | • initial release |

**Design Revision History**

# Release Notes and Know Issues

| Issues | Description | Workaround | To be fixed version |
|---|---|---|---|
| EEPROM U10 is not writeable | WP is fix on on PCB Revisions, which shipped before 2019-06-13 | PCB can be patched, send request to Trenz Electronic support | --- |

**Known Issues**

# Requirements

## Software

| Software | Version | Note |
|---|---|---|
| Vitis | 2022.2 | needed<br>Vivado is included into Vitis installation |
| PetaLinux | 2022.2 | needed |

**Software**

## Hardware

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

Complete List is available on "<project folder>\board_files*_board_files.csv"

Design supports following modules:

| Module Model | Board Part Short Name | PCB Revision Support | DDR | QSPI Flash | EMMC | Others | Notes |
|---|---|---|---|---|---|---|---|
| TE0724-02-10-1I | 10_1i_1gb | REV02 | 1GB | 32MB | NA | NA | NA |
| TE0724-02-20-1 | 20_1i_1gb | REV02 | 1GB | 32MB | NA | NA | NA |
| TE0724-02-20-1IC1 | 20_1i_1gb | REV02 | 1GB | 32MB | NA | NA | NA |
| TE0724-03-10-1I | 10_1i_1gb | REV03 | 1GB | 32MB | NA | NA | NA |
| TE0724-03-20-1I | 20_1i_1gb | REV03 | 1GB | 32MB | NA | NA | NA |
| TE0724-03-20-1IC1 | 20_1i_1gb | REV03 | 1GB | 32MB | NA | NA | NA |
| TE0724-04-41I32-A | 10_1i_1gb | REV04 | 1GB | 32MB | NA | NA | NA |
| TE0724-04-41I33-A* | 10_1i_1gb | REV04 | 1GB | 64MB | NA | NA | NA |
| TE0724-04-41I33-AZ | 10_1i_1gb | REV04 | 1GB | 64MB | NA | NA | NA |
| TE0724-04-61I32-A | 20_1i_1gb | REV04 | 1GB | 32MB | NA | NA | NA |
| TE0724-04-61I32-AZ | 20_1i_1gb | REV04 | 1GB | 32MB | NA | NA | NA |
| TE0724-04-61I33-AC | 20_1i_1gb | REV04 | 1GB | 64MB | NA | NA | NA |
| TE0724-04-S001 | 10_1i_1gb | REV04 | 1GB | 64MB | NA | NA | CAO |
| TE0724-04-S003 | 10_1i_1gb | REV04 | 1GB | 64MB | NA | NA | CAO |
| TE0724-04-S004 | 10_1i_1gb | REV04 | 1GB | 64MB | NA | NA | CAO |
| TE0724-04-S005 | 10_1i_1gb | REV04 | 1GB | 64MB | NA | NA | CAO |
| TE0724-04-S008 | 10_1i_1gb | REV04 | 1GB | 64MB | NA | NA | CAO |
| TE0724-04-S011C1 | 20_1i_1gb | REV04 | 1GB | 64MB | NA | NA | CAO |
| TE0724-04-61I33-A | 20_1i_1gb | REV04 | 1GB | 64MB | NA | NA | NA |

*used as reference

**Hardware Modules**

Design supports following carriers:

| Carrier Model | Notes |
|---|---|
| TEB0724 | |

*used as reference

**Hardware Carrier**

Additional HW Requirements:

| Additional Hardware | Notes |
|---|---|
| | |

*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<br><project folder>\constraints<br><project folder>\ip_lib<br><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 |
|---|---|---|
| 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:

- TE0724 "Test Board" Reference Design

# Design Flow



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

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

See also:

- 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>")

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

**_create_win_setup.cmd/_create_linux_setup.sh**

```
------------------------Set design paths---------------------------
-- Run Design with: _create_win_setup
-- Use Design Path: <absolute project path>
-------------------------------------------------------------------
------------------------TE Reference Design-------------------------
-------------------------------------------------------------------
-- (0)  Module selection guide, project creation...prebuilt export...
-- (1)  Create minimum setup of CMD-Files and exit Batch
-- (2)  Create maximum setup of CMD-Files and exit Batch
-- (3)  (internal only) Dev
-- (4)  (internal only) Prod
-- (c)  Go to CMD-File Generation (Manual setup)
-- (d)  Go to Documentation (Web Documentation)
-- (g)  Install Board Files from Xilinx Board Store (beta)
-- (a)  Start design with unsupported Vivado Version (beta)
-- (x)  Exit Batch (nothing is done!)
----
Select (ex.:'0' for module selection guide):
```

2. Press 0 and enter to start "Module Selection Guide"
3. Create project and follow instructions of the product selection guide, settings file will be configured automatically during this process.
    - optional for manual changes: Select correct device and Xilinx install path on "design_basic_settings.cmd" and create Vivado project with "vivado_create_project_guimode.cmd"

Note: Select correct one, see also Vivado Board Part Flow

4. Create hardware description file (.xsa file) for PetaLinux project and export to prebuilt folder

**run on Vivado TCL (Script generates design and export files into "<project folder>\prebuilt\hardware\<short name>")**

```
TE::hw_build_design -export_prebuilt
```

Using Vivado GUI is the same, except file export to prebuilt folder.

5. Create and configure your PetaLinux project with exported .xsa-file, see PetaLinux KICKstart
    - use TE Template from "<project folder>\os\petalinux"
    - use exported .xsa file from "<project folder>\prebuilt\hardware\<short name>" . **Note:** HW Export from Vivado GUI creates another path as default workspace.

- The build images are located in the "<plnx-proj-root>/images/linux" directory

6. Configure the **boot.scr** file as needed, see Distro Boot with Boot.scr
7. Generate Programming Files with Vitis (recommended)
    a. Copy PetaLinux build image files to prebuilt folder
        - copy **u-boot.elf**, **system.dtb, image.ub** and **boot.scr** from "<plnx-proj-root>/images/linux" to prebuilt folder

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

    b. Generate Programming Files with Vitis

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

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

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

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

# Launch

## Programming

> Check Module and Carrier TRMs for proper HW configuration before you try any design.
>
> Reference Design is also available with prebuilt files. It's recommended to use TE prebuilt files for first launch.

Xilinx documentation for programming and debugging: Vivado/Vitis/SDSoC-Xilinx Software Programming and Debugging

### Get prebuilt boot binaries

1. Run _create_win_setup.cmd/_create_linux_setup.sh and follow instructions on shell
2. Press 0 and enter to start "Module Selection Guide"
    a. Select assembly version
    b. Validate selection
    c. Select Create and open delivery binary folder

    > Note: Folder "<project folder>_binaries_<Article Name>" with subfolder "boot_<app name>" for different applications will be generated

### QSPI-Boot mode

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

1. Connect **JTAG** and power on carrier with module
2. Open Vivado Project with "vivado_open_existing_project_guimode.cmd" or if not created, create with "vivado_create_project_guimode.cmd"

**run on Vivado TCL (Script programs BOOT.bin on QSPI flash)**

```
TE::pr_program_flash -swapp u-boot
TE::pr_program_flash -swapp hello_te0724 (optional)
```

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

## SD-Boot mode

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

## JTAG

Not used on this Example.

# Usage

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

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

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

4. Power On PCB
   1. Zynq Boot ROM loads FSBL from SD/QSPI into OCM,
   2. FSBL init the PS, programs the PL using the bitstream and loads 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)

a. Speed: 115200
b. 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.

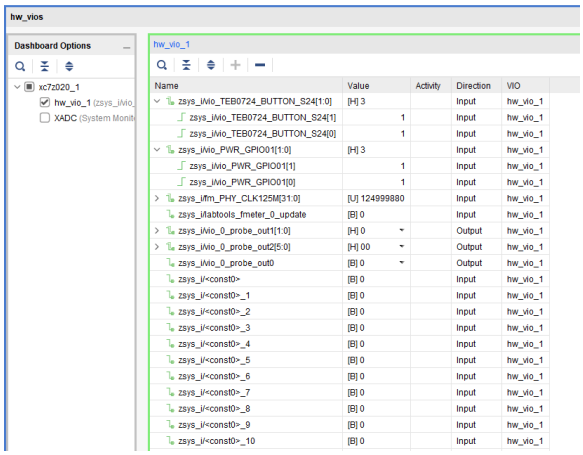
```
I2C
        i2cdetect -l        (Shows a list of the available I2C buses)
        i2cdetect -y -r 0        (check I2C 0 Bus)
RTC
        dmesg | grep rtc        (RTC check)
ETH0
        udhcpc                                  (ETH0 check)
GPIO
        gpiodetect                          (list all gpiochips present on the
system)
        gpioget `gpiofind "MIO51_J9-6"`               (read value of
specified GPIO)
         gpioset `gpiofind "MIO9_D8"`=1               (set value of specified
GPIO)
```

4. Option Features
    - Webserver to get access to Zynq
        - 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 HW Manager

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

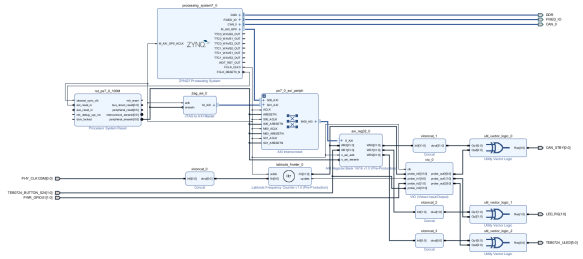
- Control:
    - CAN_STBY[0:0] CAN Standby control
    - LED_RG[1:0] module LED control
    - TEB0724_ULED[5:0] TEB0724 LED control
- Monitoring:
    - vio_TEB0724_BUTTON_S24[1:0] TEB0724 Button S2 and S4
    - vio_PWR__GPIO01[1:0] PMIC GPIO
    - fm_PHY_CLK125M[31:0] PHY Clock 125MHz
    - labtools_fmeter_0_update FMeter Update




**Vivado Hardware Manager**

# System Design - Vivado

## Block Design



**Block Design**

## PS Interfaces

| Type | Note |
|---|---|
| DDR | |
| QSPI | MIO |
| ETH0 | MIO |
| SD0 | MIO |
| UART1 | MIO |
| I2C1 | MIO |
| CAN0 | EMIO |
| GPIO | MIO |

| | |
|---|---|
| TTC0..1 | EMIO |
| WDT | EMIO |

**PS Interfaces**

# Constrains

## Basic module constrains

**_i_bitgen_common.xdc**

```
#
# Common BITGEN related settings for TE0724 SoM
#
set_property BITSTREAM.GENERAL.COMPRESS TRUE [current_design]
set_property CONFIG_VOLTAGE 3.3 [current_design]
set_property CFGBVS VCCO [current_design]
```

## Design specific constrain

**_i_io.xdc**

```
# can
set_property PACKAGE_PIN T11 [get_ports CAN_0_tx]
set_property IOSTANDARD LVCMOS33 [get_ports CAN_0_tx]
set_property PACKAGE_PIN T10 [get_ports CAN_0_rx]
set_property IOSTANDARD LVCMOS33 [get_ports CAN_0_rx]
set_property PACKAGE_PIN U13 [get_ports {CAN_STBY[0]}]
set_property IOSTANDARD LVCMOS33 [get_ports {CAN_STBY[0]}]
# led
set_property PACKAGE_PIN U12 [get_ports {LED_RG[0]}]
set_property PACKAGE_PIN W13 [get_ports {LED_RG[1]}]
set_property IOSTANDARD LVCMOS33 [get_ports {LED_RG[*]}]
# CLK
set_property PACKAGE_PIN U14 [get_ports {PHY_CLK125M[0]}]
set_property IOSTANDARD LVCMOS33 [get_ports {PHY_CLK125M[0]}]
# PWR GPIO
set_property PACKAGE_PIN T12 [get_ports {PWR_GPIO01[0]}]
set_property PACKAGE_PIN U15 [get_ports {PWR_GPIO01[1]}]
set_property IOSTANDARD LVCMOS33 [get_ports {PWR_GPIO01[*]}]
# TEB0724 Button
set_property PACKAGE_PIN Y19 [get_ports {TEB0724_BUTTON_S24[0]}]
set_property PACKAGE_PIN Y18 [get_ports {TEB0724_BUTTON_S24[1]}]
set_property IOSTANDARD LVCMOS33 [get_ports {TEB0724_BUTTON_S24[*]}]
# TEB0724 LED
set_property PACKAGE_PIN P18 [get_ports {TEB0724_ULED[0]}]
set_property PACKAGE_PIN N17 [get_ports {TEB0724_ULED[1]}]
set_property PACKAGE_PIN R17 [get_ports {TEB0724_ULED[2]}]
set_property PACKAGE_PIN R16 [get_ports {TEB0724_ULED[3]}]
set_property PACKAGE_PIN Y14 [get_ports {TEB0724_ULED[4]}]
set_property PACKAGE_PIN W14 [get_ports {TEB0724_ULED[5]}]
set_property IOSTANDARD LVCMOS33 [get_ports {TEB0724_ULED[*]}]
```

# Software Design - Vitis

For Vitis project creation, follow instructions from:

Vitis

## Application

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

### zynq_fsbl

TE modified 2022.2 FSBL

General:

- Modified Files: main.c, fsbl_hooks.h/.c (search for 'TE Mod' on source code)
- Add Files: te_fsbl_hooks.h/.c (for hooks and board)
- General Changes:
    - Display FSBL Banner and Device ID

Module Specific:

- Add Files: all TE Files start with te_*
    - enable I2C voltage-level translator over MIO38. Needed for PMIC-RTC and external I2C.

### hello_te0724

Hello TE0724 is a Xilinx Hello World example as endless loop instead of one console output.

### u-boot

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

# Software Design - PetaLinux

For PetaLinux installation and project creation, follow instructions from:

- PetaLinux KICKstart

## Config

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

Changes:

- add new flash partition for bootscr and sizing
    - CONFIG_SUBSYSTEM_FLASH_PSU_QSPI_0_BANKLESS_PART0_SIZE=0xA00000
    - CONFIG_SUBSYSTEM_FLASH_PSU_QSPI_0_BANKLESS_PART1_SIZE=0x1400000
    - CONFIG_SUBSYSTEM_FLASH_PSU_QSPI_0_BANKLESS_PART2_SIZE=0x20000
    - CONFIG_SUBSYSTEM_FLASH_PSU_QSPI_0_BANKLESS_PART3_NAME="bootscr"
    - CONFIG_SUBSYSTEM_FLASH_PSU_QSPI_0_BANKLESS_PART3_SIZE=0x40000
- Identification
    - CONFIG_SUBSYSTEM_HOSTNAME="Trenz"
    - CONFIG_SUBSYSTEM_PRODUCT="TE0724"
- MAC from eeprom together with uboot and device tree settings:
    - CONFIG_SUBSYSTEM_ETHERNET_PS7_ETHERNET_0_MAC_AUTO is not set

- CONFIG_SUBSYSTEM_ETHERNET_PS7_ETHERNET_0_MAC=""

# 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_NET_RANDOM_ETHADDR is not set
- Boot Modes:
  - CONFIG_QSPI_BOOT=y
  - CONFIG_SD_BOOT=y
  - CONFIG_ENV_IS_IN_FAT is not set
  - CONFIG_ENV_IS_IN_NAND is not set
  - CONFIG_ENV_IS_IN_SPI_FLASH is not set
  - CONFIG_SYS_REDUNDAND_ENVIRONMENT is not set
  - CONFIG_BOOT_SCRIPT_OFFSET=0x1E20000
- Identification
  - CONFIG_IDENT_STRING=" TE0724"

Change platform-top.h:

```
#include <configs/zynq-common.h>
#no changes
```

# Device Tree

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

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

 /*-------------------------- default ---------------------*/

 /*--------------------------- GPIO ----------------------*/
 &gpio0 {
    gpio-line-names =
        "MIO0_PWR"    , ""            , ""            , ""              ,
""              ,
        ""            , ""            , "MIO7"        , ""              ,
"MIO9_D8"     ,
        "MIO10_J7-5"  , "MIO11_J7-6"  , "MIO12_J7-7"  , "MIO13_J7-8"    , "MIO14_J7-
9"  ,
        "MIO15_J7-10" , ""            , ""            , ""              ,
""              ,
        ""            , ""            , ""            , ""              ,
""              ,
        ""            , ""            , ""            , ""              ,
""              ,
        "MIO30_nC"    , "MIO31_nC"    , "MIO32_nC"    , "MIO33_nC"      ,
"MIO34_nC"    ,
        "MIO35_nC"    , "MIO36_nC"    , "MIO37_nC"    , "MIO38_TCA-OE"  ,
""              ,
        ""            , ""            , ""            , ""              ,
""              ,
```

```
        ""            , "MIO46_J9-4"  , ""            , ""              ,
""
        ,
        "MIO50_J9-5"  , "MIO51_J9-6"  , ""            , ""              ;
 };

 /*--------------------------- QSPI ----------------------*/
 &qspi {
     is-dual = <0>;
     num-cs = <1>;
     #address-cells = <1>;
     #size-cells = <0>;
     status = "okay";
     spi-rx-bus-width = <4>;
     flash0: flash@0 {
         compatible = "jedec,spi-nor";
         reg = <0x0>;
         #address-cells = <1>;
         #size-cells = <1>;
         spi-rx-bus-width = <4>;
     };
 };

 /*-------------------------- ETH PHY --------------------*/
 &gem0 {
     /delete-property/ local-mac-address;
         phy-handle = <&phy0>;

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

         phy0: phy@0 {
             device_type = "ethernet-phy";
             reg = <1>;
         };
 };

 /*--------------------------- I2C ----------------------*/
 &i2c1 {

   //pmic
   pmic0: da9062@58 {
     compatible = "dlg,da9062";
     reg = <0x58>;
     interrupt-parent = <&gpio0>;
     interrupts = <0 8>;
     interrupt-controller;
     rtc {
         compatible = "dlg,da9062-rtc";
     };
   };

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

     #address-cells = <1>;
     #size-cells = <1>;
     eth0_addr: eth-mac-addr@FA {
       reg = <0xFA 0x06>;
     };
```

```
  };

 //user EEPROM
  eeprom50: eeprom@50 {
    compatible = "microchip,24aa128", "atmel,24c128";
    reg = <0x50>;
  };

};
```

## Kernel

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

Changes:

- CONFIG_REGMAP_IRQ=y
- # CONFIG_PINCTRL_DA9062 is not set
- # CONFIG_DA9062_THERMAL is not set
- # CONFIG_DA9062_WATCHDOG is not set
- CONFIG_MFD_DA9062=y
- # CONFIG_REGULATOR_DA9062 is not set
- CONFIG_RTC_DRV_DA9063=y

## 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 use of libgpiod-tools (gpiodetect, gpioset, gpioget, ...) together with device tree settings:
  - CONFIG_libgpiod-tools=y
- For auto login:
  - CONFIG_auto-login=y
  - CONFIG_ADD_EXTRA_USERS="root:root;petalinux:;"

Add in <project folder>\os\petalinux\project-spec\meta-user\conf\user-rootfsconfig

```
CONFIG_libgpiod-tools
```

## FSBL patch (alternative for vitis fsbl trenz patch)

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

## Applications

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

### startup

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

### webfwu

Webserver application suitable for Zynq access. Need busybox-httpd

## Additional Software

No additional software is needed.

## 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 |
|---|---|---|---|
| **Error rendering macro 'page-info'**<br>Ambiguous method overloading for method jdk.proxy241.$Proxy3496#hasContentLevelPermission. Cannot resolve which method to | **Error rendering macro 'page-info'**<br>Ambiguous method overloading for method jdk.proxy241.$Proxy3496#hasContentLevelPermission. Cannot resolve which method to | **Error rendering macro 'page-info'**<br>Ambiguous method overloading for method jdk.proxy241.$Proxy3496#hasContentLevelPermission. Cannot resolve which method to | • Release Vivado 2022.2<br>• new assembly variants |

invoke for [null, class java.lang. String, class com. atlassian. confluence. pages. Page] due to overlapping prototypes between: [interface com. atlassian. confluence. user. Confluence User, class java.lang. String, class com. atlassian. confluence. core. ContentEntit yObject] [interface com. atlassian. user.User, class java. lang.String, class com. atlassian. confluence. core. ContentEntit

invoke for [null, class java.lang. String, class com. atlassian. confluence. pages. Page] due to overlapping prototypes between: [interface com. atlassian. confluence. user. Confluence User, class java.lang. String, class com. atlassian. confluence. core. ContentEntit yObject] [interface com. atlassian. user.User, class java. lang.String, class com. atlassian. confluence. core. ContentEntit

invoke for [null, class java.lang. String, class com. atlassian. confluence. pages. Page] due to overlapping prototypes between: [interface com. atlassian. confluence. user. Confluence User, class java.lang. String, class com. atlassian. confluence. core. ContentEntit yObject] [interface com. atlassian. user.User, class java. lang.String, class com. atlassian. confluence. core. ContentEntit

| | | | |
|---|---|---|---|
| yObject] | yObject] | yObject] | |
| 2022-11-21 | v.12 | Manuela Strücker | • Release Vivado 2021.2.1<br>• script update<br>• new assembly variants |
| 2020-03-25 | v.10 | John Hartfiel | • script update |
| 2020-01-30 | v.9 | John Hartfiel | • Release 2019.2<br>• document style update |
| 2019-12-12 | v.8 | John Hartfiel | • Bugfix IO constrains |
| 2019-06-13 | v.7 | John Hartfiel | • Update Design Files<br>• Notes U10 access |
| 2019-02-04 | v.6 | John Hartfiel | • Update Design Files |
| 2018-08-30 | v.5 | John Hartfiel | • 2018.2 release |
| -- | all | **Error rendering macro 'page-info'**<br>Ambiguous method overloading for method jdk.proxy241.$Proxy3496#hasContentLevelPermission. Cannot resolve which method to invoke for | -- |

[null, class java.lang.String, class com.atlassian.confluence.pages.Page] due to overlapping prototypes between: [interface com.atlassian.confluence.user.ConfluenceUser, class java.lang.String, class com.atlassian.confluence.core.ContentEntityObject] [interface com.atlassian.user.User, class java.lang.String, class com.atlassian.confluence.core.ContentEntityObject]

| | | | |
|---|---|---|---|
| | | | |

**Document change history.**

# Legal Notices

# Data Privacy

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

# Document Warranty

The material contained in this document is provided “as is” and is subject to being changed at any time without notice. Trenz Electronic does not warrant the accuracy and completeness of the materials in this document. Further, to the maximum extent permitted by applicable law, Trenz Electronic disclaims all warranties, either express or implied, with regard to this document and any information contained herein, including but not limited to the implied warranties of merchantability, fitness for a particular purpose or non infringement of intellectual property. Trenz Electronic shall not be liable for errors or for incidental or consequential damages in connection with the furnishing, use, or performance of this document or of any information contained herein.

# Limitation of Liability

In no event will Trenz Electronic, its suppliers, or other third parties mentioned in this document be liable for any damages whatsoever (including, without limitation, those resulting from lost profits, lost data or business interruption) arising out of the use, inability to use, or the results of use of this document, any documents linked to this document, or the materials or information contained at any or all such documents. If your use of the materials or information from this document results in the need for servicing, repair or correction of equipment or data, you assume all costs thereof.

# Copyright Notice



# Technology Licenses

The hardware / firmware / software described in this document are furnished under a license and may be used /modified / copied only in accordance with the terms of such license.

# Environmental Protection

To confront directly with the responsibility toward the environment, the global community and eventually also oneself. Such a resolution should be integral part not only of everybody's life. Also enterprises shall be conscious of their social responsibility and contribute to the preservation of our common living space. That is why Trenz Electronic invests in the protection of our Environment.

# REACH, RoHS and WEEE

**REACH**

Trenz Electronic is a manufacturer and a distributor of electronic products. It is therefore a so called downstream user in the sense of REACH. The products we supply to you are solely non-chemical products (goods). Moreover and under normal and reasonably foreseeable circumstances of application, the goods supplied to you shall not release any substance. For that, Trenz Electronic is obliged to neither register nor to provide safety data sheet. According to present knowledge and to best of our knowledge, no SVHC (Substances of Very High Concern) on the Candidate List are contained in our products. Furthermore, we will immediately and unsolicited inform our customers in compliance with REACH - Article 33 if any substance present in our goods (above a concentration of 0,1 % weight by weight) will be classified as SVHC by the European Chemicals Agency (ECHA).

**RoHS**

Trenz Electronic GmbH herewith declares that all its products are developed, manufactured and distributed RoHS compliant.

**WEEE**

Information for users within the European Union in accordance with Directive 2002/96/EC of the European Parliament and of the Council of 27 January 2003 on waste electrical and electronic equipment (WEEE).

Users of electrical and electronic equipment in private households are required not to dispose of waste electrical and electronic equipment as unsorted municipal waste and to collect such waste electrical and electronic equipment separately. By the 13 August 2005, Member States shall have ensured that systems are set up allowing final holders and distributors to return waste electrical and electronic equipment at least free of charge. Member States shall ensure the availability and accessibility of the necessary collection facilities. Separate collection is the precondition to ensure specific treatment and recycling of waste electrical and electronic equipment and is necessary to achieve the chosen level of protection of human health and the environment in the European Union. Consumers have to actively contribute to the success of such collection and the return of waste electrical and electronic equipment. Presence of hazardous substances in electrical and electronic equipment results in potential effects on the environment and human health. The symbol consisting of the crossed-out wheeled bin indicates separate collection for waste electrical and electronic equipment.

Trenz Electronic is registered under WEEE-Reg.-Nr. DE97922676.

**Error rendering macro 'page-info'**

Ambiguous method overloading for method jdk.proxy241.$Proxy3496#hasContentLevelPermission. Cannot resolve which method to invoke for [null, class java.lang.String, class com.atlassian.confluence.pages.Page] due to overlapping prototypes between: [interface com.atlassian.confluence.user.ConfluenceUser, class java.lang.String, class com.atlassian.confluence.core.ContentEntityObject] [interface com.atlassian.user.User, class java.lang.String, class com.atlassian.confluence.core.ContentEntityObject]