TE0724 Test Board

Revision v.10

Exported on 2022-10-21

Online version of this document:

https://wiki.trenz-electronic.de/display/PD/TE0724+Test+Board

1 Table of Contents

2 Table of Figures

3 Table of Tables

4 Overview

Refer to http://trenz.org/te0724-info for the current online version of this manual and other available documentation.

4.1 Key Features

- Vitis/Vivado 2019.2
- PetaLinux
- SD
- ETH
- MAC from EEPROM
- I2C
- RTC
- FMeter
- FSBL to enable I2C Buffer for PMIC(RTC) and external I2C
- Special FSBL for QSPI programming

4.2 Revision History

Date	Vivado	Project Built	Authors	Description
2020-03-25	2019.2	TE0724-test_board-vivado_2019.2-build_8_20200325075929.zip 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 TE0724-test_board-vivado_2019.2-build_4_20200130130040.zip	John Hartfiel	• 2019.2 update • Vitis support • FSBL changes • petalinux device tree and u-boot update
2019-13-12	2018.2	TE0724-test_board_noprebuilt-vivado_2018.2-build_04_20191212064015.zip 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 TE0724-test_board_noprebuilt-	Oleksandr Kiyenko,	• add app to get access to EEPROM U10

Date	Vivado	Project Built	Authors	Description
		vivado_2018.2-build_04_20190613115049.zip	John Hartfiel	
2019-02-04	2018.2	TE0724-test_board-vivado_2018.2-build_04_20190204111543.zip TE0724-test_board_noprebuilt-vivado_2018.2-build_04_20190204111557.zip	John Hartfiel	• Important Board Part File Update • change DDR3 to DDR3 Low Power
2018-08-29	2018.2	TE0724-test_board_noprebuilt-vivado_2018.2-build_03_20180830170634.zip TE0724-test_board-vivado_2018.2-build_03_20180830170621.zip	John Hartfiel	• initial release

Table 1: Design Revision History

4.3 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	---

Table 2: Known Issues

4.4 Requirements

4.4.1 Software

Software	Version	Note
Vitis	2019.2	needed, Vivado is included into Vitis installation
PetaLinux	2019.2	needed

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
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
TE0724-03-10-1I	10_1i_1gb	REV03	1GB	32MB	NA	NA	NA
TE0724-03-20-1	20_1i_1gb	REV03	1GB	32MB	NA	NA	NA

Table 4: Hardware Modules

Design supports following carriers:

Carrier Model	Notes
TEB0724	

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]

1 https://wiki.trenz-electronic.de/display/PD/TE+Board+Part+Files
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
Vitis	<design name>/sw_lib	Additional Software Template for Vitis and apps_list.csv with settings automatically for Vitis app generation
PetaLinux	<design name>/os/petalinux	PetaLinux template with current configuration

Table 7: Design sources

4.5.2 Additional Sources

Type	Location	Notes
init.sh	<design name>/misc/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

File	File-Extension	Description
DebugProbes-File	*.ltx	Definition File for Vivado/Vivado Labtools Debugging Interface
Diverse Reports	---	Report files in different formats
Hardware-Platform-Specification-Files	*.xsa	Exported Vivado Hardware Specification 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

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

4.5.4 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[3]

3 https://shop.trenz-electronic.de/Download/?path=Trenz_Electronic/Modules_and_Module_Carriers/4x6/TE0724/Reference_Design/2019.2/test_board

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.

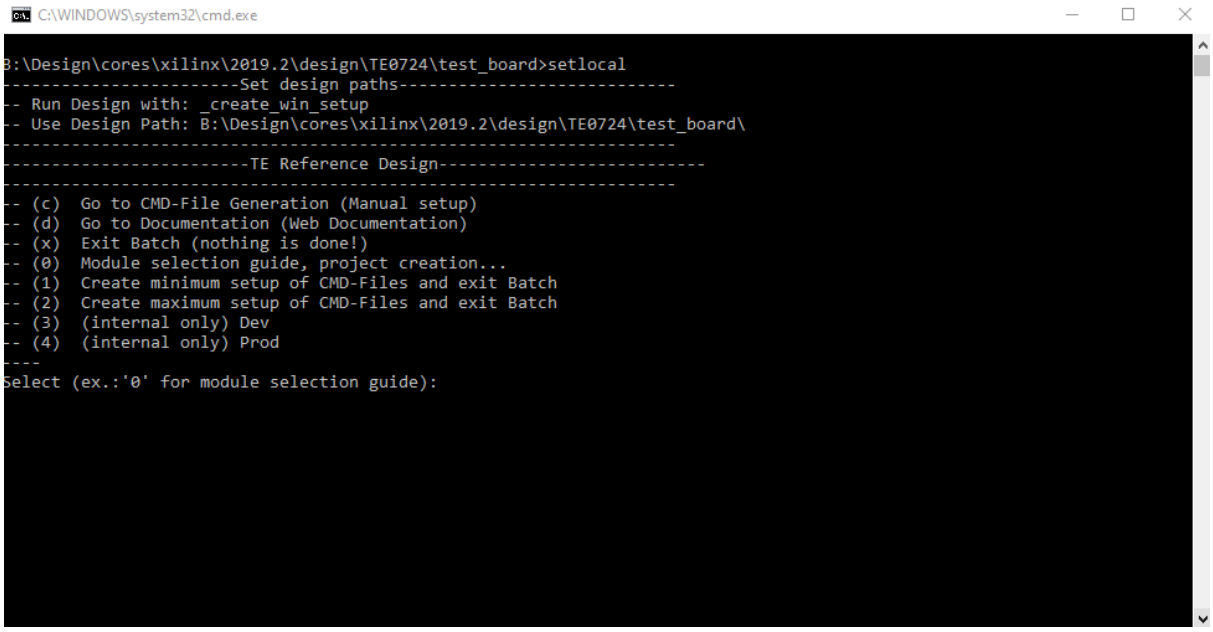
See also:

- 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:



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
   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 alsoTE Board Part Files[8]
5. Create XSA 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#ProjectDeliveryXilinxdevices-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
6. Create Linux (uboot.elf and image.ub) with exported XSA
   a. XSA 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
7. Add Linux files (uboot.elf and image.ub) to prebuilt folder
   a. "prebuilt\os\petalinux\<ddr size>" or "prebuilt\os\petalinux\<short name>"
8. Generate Programming Files with Vitis
   a. Run on Vivado TCL: TE::sw_run_vitis -all
      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_vitis
      Note: TCL scripts generate also platform project, this must be done manuelly in case GUI is used. See Vitis[10]

9 https://wiki.trenz-electronic.de/display/PD/PetaLinux+KICKstart
10 https://wiki.trenz-electronic.de/display/PD/Vitis

6 Launch

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[11]

6.1.1 Get prebuilt boot binaries

1. _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 foler>/_binaries_<Artikel Name>) with subfolder (boot_<app name>) for different applications will be generated

6.1.2 QSPI

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

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"
3. 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_te0724" possible
4. Set Boot Mode to QSPI.
   - Depends on Carrier, see carrier TRM.
5. Copy image.ub on SD-Card
   - use files from (<project foler>/_binaries_<Articel Name>)/boot_linux from generated binary folder,see: Get prebuilt boot binaries(see page 13)
   - or use prebuilt file location, see <design_name>/prebuilt/readme_file_location.txt
6. Insert SD-Card

6.1.3 SD

1. Copy image.ub and Boot.bin on SD-Card.
   - use files from (<project foler>/_binaries_<Articel Name>)/boot_linux from generated binary folder,see: Get prebuilt boot binaries(see page 13)
   - or use 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.

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

6.1.4 JTAG

Not used on this Example.

6.2 Usage

1. Prepare HW like described on section Programming(see page 13)
2. Connect UART USB (most cases same as JTAG)
3. Select SD Card as Boot Mode (or QSPI depending on programming option)
   Note: See TRM of the Carrier, which is used.
4. Power On PCB
   Note: 1. Zynq Boot ROM loads FSBL from SD into OCM, 2. FSBL loads U-boot from SD 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. RTC check: dmesg | grep rtc
   c. ETH0 works with udhcpc

6.2.2 Vivado HW Manager

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

- Control:
  - CAN Standby control
  - module LED control
  - TEB0724 LED control
- Monitoring:
  - TEB0724 Button
  - PMIC GPIO
  - PHY 125MHz

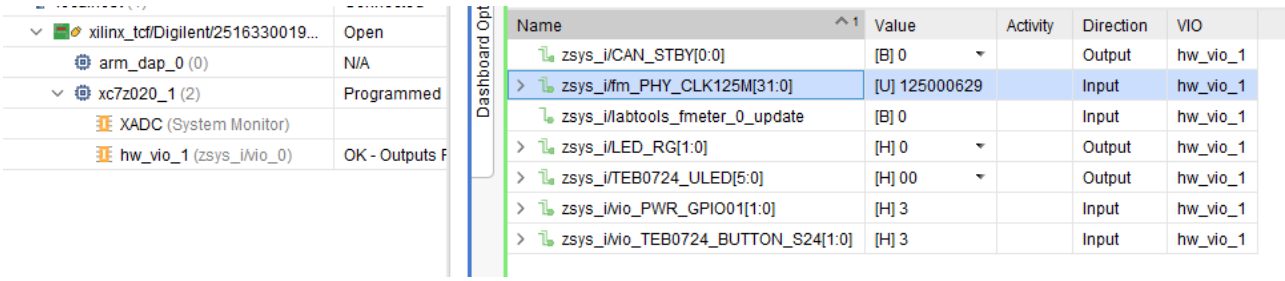


Figure 1: Vivado Hardware Manager

7 System Design - Vivado

7.1 Block Design

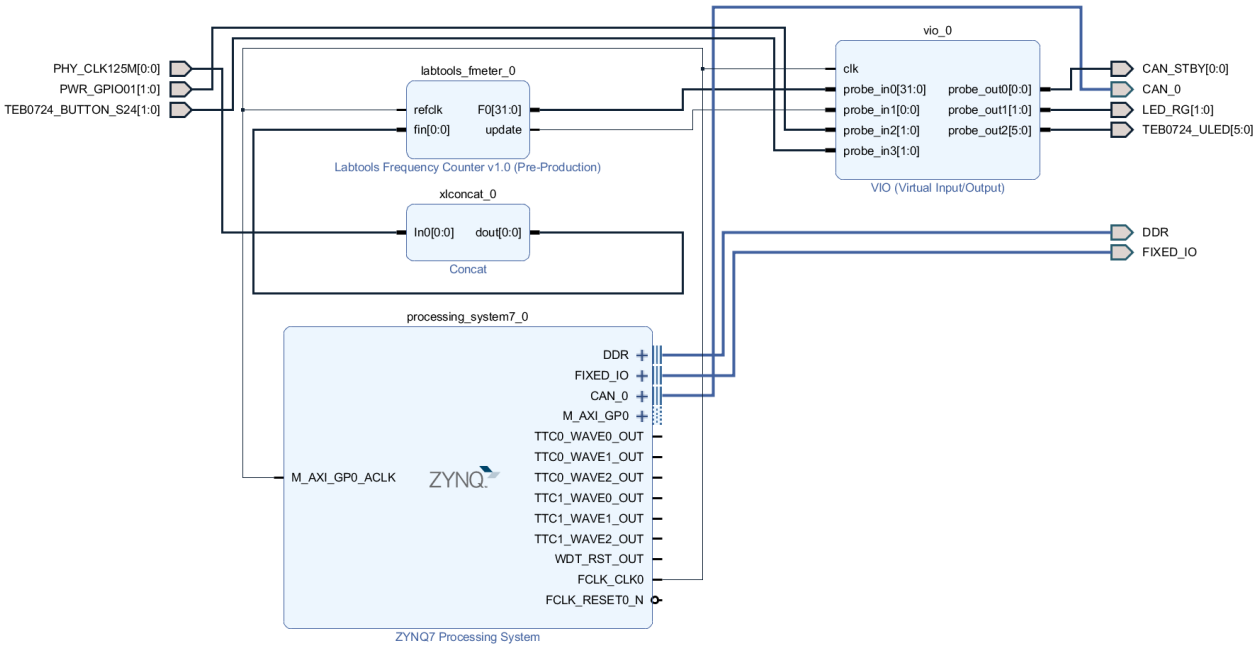


Figure 2: Block Design

7.1.1 PS Interfaces

Type	Note
DDR	
QSPI	MIO
ETH0	MIO
SD0	MIO
UART1	MIO
I2C1	MIO

Type	Note
CAN0	EMIO
GPIO	MIO
TTC0..1	EMIO
WDT	EMIO

Table 10: PS Interfaces

7.2 Constrains

7.2.1 Basic module constrains

_i_bitgen_common.xdc

```
#
# Common BITGEN related settings for TE0720 SoM
#
set_property BITSTREAM.GENERAL.COMPRESS TRUE [current_design]
set_property CONFIG_VOLTAGE 3.3 [current_design]
set_property CFGBVS VCCO [current_design]
```

7.2.2 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[*]}]
```

8 Software Design - Vitis

For SDK project creation, follow instructions from:

Vitis[12]

8.1 Application

Template location: ./sw_lib/sw_apps/

8.1.1 zynq_fsbl

TE modified 2019.2 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)\n\
- General Changes:
    - Display FSBL Banner and Device ID

Module Specific:

- Add Files: all TE Files start with te_*
    - enable I2C Buffer over MIO38, needed for RTC and external I2C

8.1.2 zynq_fsbl_flash

TE modified 2019.2 FSBL

General:

- Modified Files: main.c
- General Changes:
    - Display FSBL Banner
    - Set FSBL Boot Mode to JTAG
    - Disable Memory initialisation

8.1.3 hello_te0724

Hello World App in Endless loop.

8.1.4 u-boot

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

12 https://wiki.trenz-electronic.de/display/PD/Vitis

9 Software Design - PetaLinux

For PetaLinux installation and project creation, follow instructions from:

- PetaLinux KICKstart[13]

9.1 Config

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

Changes:

- CONFIG_SUBSYSTEM_ETHERNET_PS7_ETHERNET_0_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
- CONFIG_I2C_EEPROM=y
- CONFIG_ZYNQ_GEM_I2C_MAC_OFFSET=0xFA
- CONFIG_SYS_I2C_EEPROM_ADDR=0x53
- CONFIG_SYS_I2C_EEPROM_BUS=0
- CONFIG_SYS_EEPROM_SIZE=256
- CONFIG_SYS_EEPROM_PAGE_WRITE_BITS=0
- CONFIG_SYS_EEPROM_PAGE_WRITE_DELAY_MS=0
- CONFIG_SYS_I2C_EEPROM_ADDR_LEN=1
- CONFIG_SYS_I2C_EEPROM_ADDR_OVERFLOW=0

Change platform-top.h:

9.3 Device Tree

```
/include/ "system-conf.dtsi"
/ {
  chosen {
    xlnx,eeprom = &eeprom;
  };
};


/* default */

/* QSPI PHY */
&qspi {
    #address-cells = <1>;
```

13 https://wiki.trenz-electronic.de/display/PD/PetaLinux+KICKstart

```
    #size-cells = <0>;
    status = "okay";
    flash0: flash@0 {
        compatible = "jedec,spi-nor";
        reg = <0x0>;
        #address-cells = <1>;
        #size-cells = <1>;
    };
};


/* ETH PHY */
&gem0 {
    phy-handle = <&phy0>;
    mdio {
        #address-cells = <1>;
        #size-cells = <0>;
        phy0: phy@0 {
            compatible = "marvell,88e1510";
            device_type = "ethernet-phy";
            reg = <1>;
        };
    };
};


/* I2C */


//pmic
&i2c1 {
    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 = "atmel,24c08";
    reg = <0x53>;
  };
 //user EEPROM
  eeprom50: eeprom@50 {
    compatible = "atmel,24c128";
    reg = <0x50>;
  };
};
```

9.4 Kernel

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

Changes:

- CONFIG_REGMAP_IRQ=y
- # 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

9.5 Rootfs

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

Changes:

- i2c-tools
- 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

No additional software is needed.

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
2020-03-25	v.10 (see page 6)	John Hartfiel[14]	• script update
2020-01-30	v.9	John Hartfiel	• Release 2019.2 • document style update
2019-12-12	v.8	John Hartfiel	• Bugfix IO constrains
2019-06-13	v.7	John Hartfiel	• Update Design Files • Notes U10 access
2019-02-04	v.6	John Hartfiel	• Update Design Files
2018-08-30	v.5	John Hartfiel	• 2018.2 release
--	all	John Hartfiel[15]	--

Table 11: 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

14 https://wiki.trenz-electronic.de/display/~j.hartfiel
15 https://wiki.trenz-electronic.de/display/~j.hartfiel

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.

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[16]. 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

16 http://guidance.echa.europa.eu/

Candidate List[17] 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)[18].

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

17 https://echa.europa.eu/candidate-list-table
18 http://www.echa.europa.eu/