TE0821 Test Board

Revision v.6

Exported on 2022-10-19

Online version of this document:

https://wiki.trenz-electronic.de/display/PD/TE0821+Test+Board

1 Table of Contents

2 Table of Figures

3 Table of Tables

4 Overview

ZynqMP PS Design with Linux Example and simple frequency counter to measure SI5338 Reference CLK with Vivado HW-Manager.
Wiki Resources page: http://trenz.org/te0821-info

4.1 Key Features

- Vitis/Vivado 2020.2
- PetaLinux
- SD
- ETH
- USB
- I2C
- RTC
- FMeter
- MAC from EEPROM
- User LED (PCB REV03 only)
- Modified FSBL for SI5338 programming
- Special FSBL for QSPI programming

4.2 Revision History

Date	Vivado	Project Built	Authors	Description
2021-10-21	2020.2	TE0821-test_board-vivado_2020.2-build_8_20211013085513.zip TE0821-test_board_noprebuilt-vivado_2020.2-build_8_20211013085523.zip	John Hartfiel	• Replace 19.2 FSBL with 20.2 FSBL version • bugfix template, to support different DDR size • bugfix 2GB linux image
2021-08-24	2020.2	TE0821-test_board_noprebuilt-vivado_2020.2-build_7_20210824103059.zip TE0821-test_board-vivado_2020.2-build_7_20210824103042.zip	Mohsen Chamanbaz	• startup application added • webfwu application added
2021-08-17	2020.2	TE0821-test_board_noprebuilt-vivado_2020.2-build_7_20210817112843.zip TE0821-test_board-vivado_2020.2-build_7_20210817112826.zip	Mohsen Chamanbaz	• 2020.2 release

Date	Vivado	Project Built	Authors	Description
2020-10-06	2019.2	TE0821-test_board_noprebuilt-vivado_2019.2-build_15_20201006104048.zip TE0821-test_board-vivado_2019.2-build_15_20201006103533.zip	John Hartfiel	• new assembly variants
2020-05-29	2019.2	TE0821-test_board_noprebuilt-vivado_2019.2-build_12_20200529054245.zip TE0821-test_board-vivado_2019.2-build_12_20200529054223.zip	John Hartfiel	• initial release

Table 1: Design Revision History

4.3 Release Notes and Know Issues

Issues	Description	Workaround	To be fixed version

Table 2: Known Issues

4.4 Requirements

4.4.1 Software

Software	Version	Note
Vitis	2020.2	needed Vivado is included into Vitis installation
PetaLinux	2020.2	needed
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
TE0821-01-2AE31KA *	2cg_1e_4gb	REV03	4GB	128MB	64GB	NA	NA
TE0821-01-3BI21FA	3eg_1i_2gb	REV03	2GB	128MB	8GB	NA	NA
TE0821-01-3BI21FL	3eg_1i_2gb	REV03	2GB	128MB	8GB	2.5 mm connectors	NA
TE0821-01-3BE21FA	3eg_1e_2gb	REV03	2GB	128MB	8GB	NA	NA
TE0821-01-3BE21FL*	3eg_1e_2gb	REV03	2GB	128MB	8GB	2.5 mm connectors	NA
TE0821-01-3BE21FC	3eg_1e_2gb	REV03	2GB	128MB	8GB	NA	without encryption/ NCNR
TE0821-01-3AE31KA	3cg_1e_4gb	REV03	4GB	128MB	64GB	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
TE0821-01-4DE31FL	4ev_1e_4gb	REV03	4GB	128MB	8GB	2.5 mm connectors	

Table 4: Hardware Modules
*used as reference

Design supports following carriers:

Carrier Model	Notes
TE0701	• Important: See restrictions on usage with 7 Serie Carriers: 4 x 5 SoM Carriers[2]
TE0703	• Important: See restrictions on usage with 7 Serie Carriers: 4 x 5 cm carriers [3] • Used as reference carrier.
TE0705	• Important: See restrictions on usage with 7 Serie Carriers: 4 x 5 SoM Carriers[4]
TE0706 *	• Important: See restrictions on usage with 7 Serie Carriers: 4 x 5 SoM Carriers[5]
TEBA0841	• Important: See restrictions on usage with 7 Serie Carriers: 4 x 5 SoM Carriers[6] • No SD Slot available, pins goes to Pin Header • For TEBA0841 REV01, please contact TE support

Table 5: Hardware Carrier
*used as reference

Additional HW Requirements:

Additional Hardware	Notes
USB Cable for JTAG/UART	Check Carrier Board and Programmer for correct typ
XMOD Programmer	Carrier Board dependent, only if carrier has no own FTDI

2 https://wiki.trenz-electronic.de/display/PD/4+x+5+SoM+Carriers
3 https://wiki.trenz-electronic.de/display/PD/4+x+5+SoM+Carriers
4 https://wiki.trenz-electronic.de/display/PD/4+x+5+SoM+Carriers
5 https://wiki.trenz-electronic.de/display/PD/4+x+5+SoM+Carriers
6 https://wiki.trenz-electronic.de/display/PD/4+x+5+SoM+Carriers

Additional Hardware	Notes
Cooler	It's recommended to use cooler on ZynqMP device

Table 6: Additional Hardware

4.5 Content

For general structure and of the reference design, see Project Delivery - Xilinx devices[7]

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
SI5338	<design name>/misc/Si5338	SI5338 Project with current PLL Configuration
init.sh	<design name>/sd/	Additional Initialization Script for Linux

Table 8: Additional design sources

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

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
Hardware-Platform-Specification-Files	*.xsa	Exported Vivado Hardware Specification forVitis 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:

- TE0821 "Test Board" Reference Design[8]

8 https://shop.trenz-electronic.de/Download/?path=Trenz_Electronic/Modules_and_Module_Carriers/4x5/TE0821/Reference_Design/2020.2/test_board

5 Design Flow

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

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

See also:

- Xilinx Development Tools[9]
- Vivado Projects - TE Reference Design[10]
- Project Delivery.[11]

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[12]

> **Caution!** Win OS has a 260 character limit for path lengths which can affect the Vivado tools. To avoid this issue, use Virtual Drive or the shortest possible names and directory locations for the reference design (for example "x:\<project folder>")

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

_create_win_setup.cmd/_create_linux_setup.sh

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

2. Press 0 and enter to start "Module Selection Guide"

9 https://wiki.trenz-electronic.de/display/PD/Xilinx+Development+Tools#XilinxDevelopmentTools-XilinxSoftware-BasicUserGuides
10 https://wiki.trenz-electronic.de/display/PD/Vivado+Projects+-+TE+Reference+Design
11 https://wiki.trenz-electronic.de/display/PD/Project+Delivery+-+Xilinx+devices
12 https://wiki.trenz-electronic.de/display/PD/Project+Delivery+-+Xilinx+devices#ProjectDeliveryXilinxdevices-Currentlylimitationsoffunctionality

3. (optional Win OS) Generate Virtual Drive or use short directory for the reference design (for example x:\<design name>)
4. 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[13]

5. 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 "\prebuilt\hardware\")

```
\prebuilt\hardware\")">
TE::hw_build_design -export_prebuilt
```

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

6. Create and configure your PetaLinux project with exported .xsa-file, see PetaLinux KICKstart[14]
   - 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
7. Configure the **boot.scr** file as needed, see Distro Boot with Boot.scr[15]
8. Copy PetaLinux build image files to prebuilt folder
   - copy **u-boot.elf**, **image.ub, bl31.elf** 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>"

9. 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[16]

13 https://wiki.trenz-electronic.de/display/PD/Vivado+Board+Part+Flow
14 https://wiki.trenz-electronic.de/display/PD/PetaLinux+KICKstart
15 https://wiki.trenz-electronic.de/display/PD/Distro+Boot+with+Boot.scr
16 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.
> 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/SDK/SDSoC-Xilinx Software Programming and Debugging[17]

Note: Depending on CPLD Firmware and Boot Mode settings, QSPI boot with Linux image on SD or complete SD Boot is possible.

6.1.1 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

6.1.2 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_te0821 (optional)
   ```

   > ⚠ To program with Vitis/Vivado GUI, use special FSBL (fsbl_flash) on setup

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(see page 15)
   - or use prebuilt file location, see "<project folder>\prebuilt\file_location.txt"
4. Set Boot Mode to **QSPI-Boot** and insert **SD** or **USB**.
   - Depends on Carrier, see carrier TRM.

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

6.1.3 SD-Boot mode

Use this description for CPLD Firmware with SD Boot selectable.

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(see page 15)
   - 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.

6.1.4 JTAG

Not used on this Example.

6.2 Usage

1. Prepare HW like described on section Programming(see page 15)
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[18]

4. Power On PCB
   boot process

   1. ZynqMP Boot ROM loads PMU Firmware and FSBL from SD/QSPI Flash into OCM

   2. FSBL init PS, programs PL using the bitstream and loads U-boot from SD into DDR,

   3. U-boot loads Linux (**image.ub**) from SD/QSPI/... into DDR

6.2.1 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:

18 https://wiki.trenz-electronic.de/display/PD/Distro+Boot+with+Boot.scr

```
petalinux login: root
Password: root
```

> Note: Wait until Linux boot finished

3. You can use Linux shell now.

```
i2cdetect -y -r 0   (check I2C 0 Bus)
dmesg | grep rtc    (RTC check)
udhcpc              (ETH0 check)
lsusb               (USB check)
```

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 ./misc/SD)

6.2.2 Vivado HW Manager

Monitoring:

- SI5338_CLK0 Counter:
  - Open Vivado HW-Manager and add VIO signal to dashboard (*.ltx located on prebuilt folder).
    - Set radix from VIO signals to unsigned integer.
      Note: Frequency Counter is inaccurate and displayed unit is Hz for CLK signals
- SI5338 CLK1 is configured to 200MHz by default amd SI5338 CLK3 is configured to 125MHz by default.

Control:

- Simple loopback vio_test_in test_out
- LED over X0/X1 , see TE0821 CPLD[19]

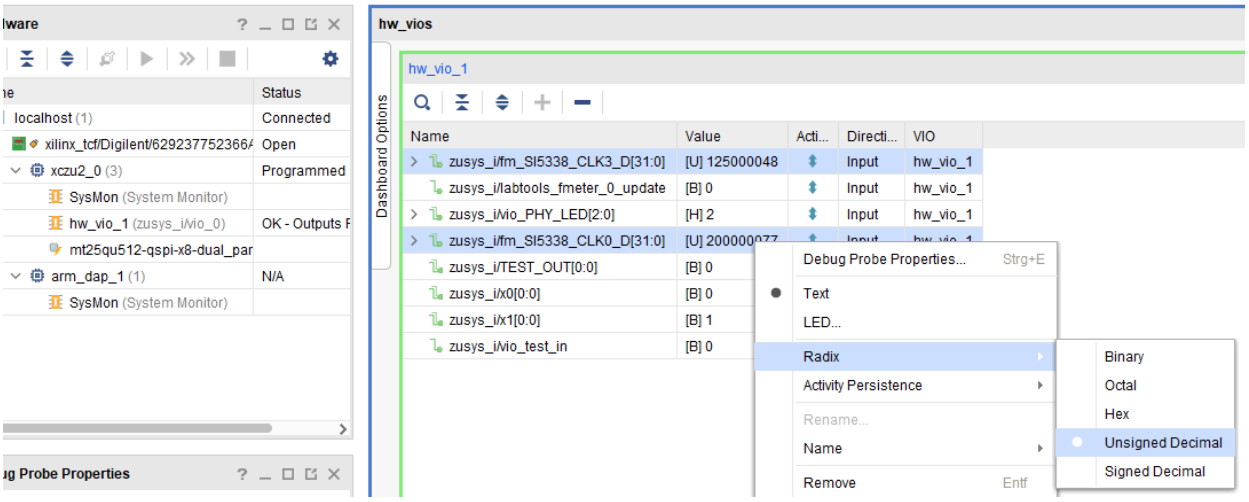

Figure 1: Vivado Hardware Manager

19 https://wiki.trenz-electronic.de/display/PD/TE0821+CPLD#TE0821CPLD-LED

7 System Design - Vivado

7.1 Block Design

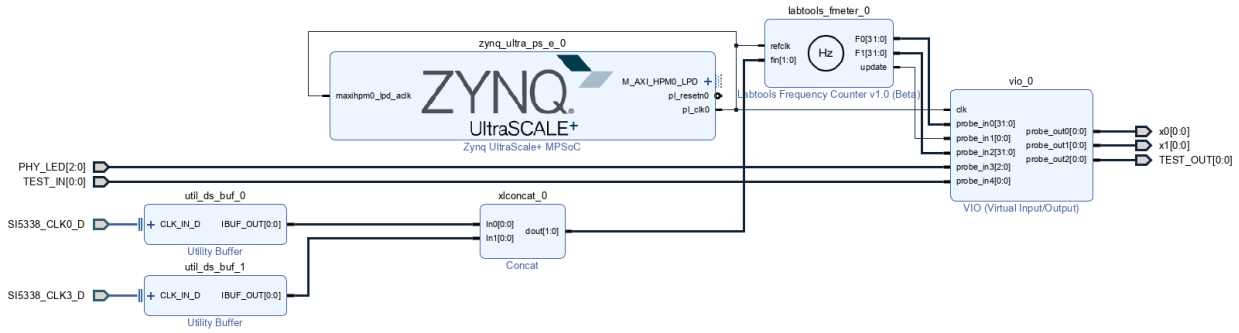

Figure 2: Block Design

7.1.1 PS Interfaces

Activated interfaces:

Type	Note
DDR	
QSPI	MIO
SD0	MIO
SD1	MIO
I2C0	MIO
UART0	MIO
GPIO0	MIO
SWDT0..1	
TTC0..3	

Type	Note
GEM3	MIO
USB0	MIO, USB2 only

Table 10: PS Interfaces

7.2 Constrains

7.2.1 Basic module constrains

_i_bitgen_common.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

```
set_property PACKAGE_PIN E5 [get_ports {SI5338_CLK0_D_clk_p[0]}]
set_property IOSTANDARD LVDS [get_ports {SI5338_CLK0_D_clk_p[0]}]
set_property PACKAGE_PIN C3 [get_ports {SI5338_CLK3_D_clk_p[0]}]
set_property IOSTANDARD LVDS [get_ports {SI5338_CLK3_D_clk_p[0]}]

set_property PACKAGE_PIN B1 [get_ports {x0[0]}]
set_property IOSTANDARD LVCMOS18 [get_ports {x0[0]}]
set_property PACKAGE_PIN C1 [get_ports {x1[0]}]
set_property IOSTANDARD LVCMOS18 [get_ports {x1[0]}]

set_property PACKAGE_PIN G8 [get_ports {PHY_LED[0]}]
set_property PACKAGE_PIN E9 [get_ports {PHY_LED[1]}]
set_property PACKAGE_PIN D9 [get_ports {PHY_LED[2]}]
set_property IOSTANDARD LVCMOS18 [get_ports {PHY_LED[*]}]

set_property PACKAGE_PIN A5 [get_ports {TEST_IN[0]}]
set_property PACKAGE_PIN B6 [get_ports {TEST_OUT[0]}]
set_property IOSTANDARD LVCMOS18 [get_ports {TEST_IN[0]}]
set_property IOSTANDARD LVCMOS18 [get_ports {TEST_OUT[0]}]
```

8 Software Design - Vitis

For SDK project creation, follow instructions from:

Vitis[20]

8.1 Application

Template location: ./sw_lib/sw_apps/

8.1.1 zynqmp_fsbl

TE modified 2020.2 FSBL

General:

- Modified Files: xfsbl_main.c, xfsbl_hooks.h/.c, xfsbl_board.h/.c(search for 'TE Mod' on source code)
- Add Files: te_xfsbl_hooks.h/.c (for hooks and board)\n\
- General Changes:
    - Display FSBL Banner and Device Name

Module Specific:

- Add Files: all TE Files start with te_*
    - Si5338 Configuration
    - ETH+OTG Reset over MIO

8.1.2 zynqmp_fsbl_flash

TE modified 2020.2 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_te0821

Hello TE0821 is a Xilinx Hello World example as endless loop instead of one console output.

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

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[21]

9.1 Config

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

Changes:

- 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
- CONFIG_I2C_EEPROM=y
- CONFIG_ZYNQ_GEM_I2C_MAC_OFFSET=0xFA
- CONFIG_SYS_I2C_EEPROM_ADDR=0x50
- 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;
  };
};


/* SDIO */

&sdhci1 {
```

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

```
   disable-wp;
   no-1-8-v;
};

/* ETH PHY */
&gem3 {

    status = "okay";
  ethernet_phy0: ethernet-phy@0 {
        compatible = "marvell,88e1510";
        device_type = "ethernet-phy";
            reg = <1>;
    };
};
/* USB 2.0 */

/* USB  */
&dwc3_0 {
    status = "okay";
    dr_mode = "host";
    maximum-speed = "high-speed";
    /delete-property/phy-names;
    /delete-property/phys;
    /delete-property/snps,usb3_lpm_capable;
     snps,dis_u2_susphy_quirk;
    snps,dis_u3_susphy_quirk;
};

&usb0 {
    status = "okay";
    /delete-property/ clocks;
    /delete-property/ clock-names;
    clocks = <0x3 0x20>;
    clock-names = "bus_clk";
};




/* QSPI PHY */
&qspi {
    #address-cells = <1>;
    #size-cells = <0>;
    status = "okay";
    flash0: flash@0 {
        compatible = "jedec,spi-nor";
        reg = <0x0>;
        #address-cells = <1>;
        #size-cells = <1>;
    };
};

&i2c0 {
  eeprom: eeprom@50 {
     compatible = "atmel,24c08";
     reg = <0x50>;
```

```
    };
};
```

9.4 FSBL patch

Must be add manually, see template

9.5 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_EDAC_CORTEX_ARM64=y

9.6 Rootfs

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

Changes:

- CONFIG_i2c-tools=y
- 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)

9.7 Applications

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

9.7.1 startup

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

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

9.7.2 webfwu

Webserver application accemble for Zynq access. Need busybox-httpd

10 Additional Software

10.1 SI5338

File location <design name>/misc/Si5338/Si5338-*.slabtimeproj

General documentation how you work with these project will be available on Si5338[22]

22 https://wiki.trenz-electronic.de/display/PD/Si5338

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
2022-06-02	v.6(see page 6)	@ John Hartfiel[23]	• Update Design flow section
2021-10-13	v.5	John Hartfiel	• Update Design files (bugfix)
2021-08-24	v.4	Mohsen Chamanbaz	• startup application added • webfwu application added
2021-08-17	v.3	Mohsen Chamanbaz	• 2020..2 release
2020-10-06	v.2	John Hartfiel	• new assembly variants
2020-05-29	v.1	John Hartfiel	• initial release
	All	@ Mohsen Chamanbaz[24], John Hartfiel[25]	

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

23 https://wiki.trenz-electronic.de/display/~j.hartfiel
24 https://wiki.trenz-electronic.de/display/~M.Chamanbaz
25 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[26]. 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

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

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[27] 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)[28].

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

27 https://echa.europa.eu/candidate-list-table
28 http://www.echa.europa.eu/