TE0715 Test Board

Revision v.32

Exported on 2020-06-10

Online version of this document:

https://wiki.trenz-electronic.de/display/PD/TE0715+Test+Board

1 Table of Contents

2 Table of Figures

3 Table of Tables

4 Overview

Zynq Design PS with Linux and simple frequency counter to measure MGT Reference CLK with Vivado HW-Manager.

Refer to http://trenz.org/te0715-info for the current online version of this manual and other available documentation.

4.1 Key Features

- Vovadp 2018.3
- PetaLinux
- SD
- ETH
- MAC from EEPROM
- USB
- I2C
- RTC
- FMeter
- Modified FSBL (some additional outputs and SI5338 reconfiguration)
- Special FSBL for QSPI Programming

4.2 Revision History

Date	Vivado	Project Built	Authors	Description
2019-05-09	2018.3	TE0715-test_board-vivado_2018.3-build_05_20190509094447.zip TE0715-test_board_noprebuilt-vivado_2018.3-build_05_20190509094505.zip	John Hartfiel	• TE Script update • rework of the FSBLs • some additional Linux features • MAC from EEPROM
2018-10-01	2018.2	TE0715-test_board-vivado_2018.2-build_03_20181001131411.zip TE0715-test_board_noprebuilt-vivado_2018.2-build_03_20181001131421.zip	John Hartfiel	• Rework Board Part Files (PS) • small design changes • SI5338 reconfiguration default activated on FSBL • update linux startup app
2018-04-26	2017.4	TE0715-test_board-vivado_2017.4-build_07_20180426171530.zip TE0715-test_board_noprebuilt-vivado_2017.4-build_07_20180426171546.zip	John Hartfiel	• new assembly variant

Date	Vivado	Project Built	Authors	Description
2018-03-27	2017.4	te0715-test_board-vivado_2017.4-build_07_20180327223552.zip te0715-test_board_noprebuilt-vivado_2017.4-build_07_20180327223606.zip	John Hartfiel	• Board Part Bug fix with UART 1
2018-01-05	2017.4	te0715-test_board-vivado_2017.4-build_01_20180105195436.zip te0715-test_board_noprebuilt-vivado_2017.4-build_01_20180105195452.zip	John Hartfiel	• No Design changes • Add FSBL for Flash Programming
2017-11-10	2017.2	te0715-test_board-vivado_2017.2-build_05_20171110134232.zip te0715-test_board_noprebuilt-vivado_2017.2-build_05_20171110134247.zip	John Hartfiel	• New Web Link on Board Part Files • Add optional FSBL Code to reprogram SI5338
2017-10-19	2017.2	te0715-test_board-vivado_2017.2-build_04_20171019141808.zip te0715-test_board_noprebuilt-vivado_2017.2-build_04_20171019141825.zip	John Hartfiel	• changed Flash typ on TE0715_board_files.csv (older one is not supported on Vivado 2017.2)
2017-09-22	2017.2	te0715-test_board-vivado_2017.2-build_02_20170927143412.zip te0715-test_board_noprebuilt-vivado_2017.2-build_02_20170927143427.zip	John Hartfiel	• initial release

Table 1: Design Revision History

4.3 Release Notes and Know Issues

Issues	Description	Workaround	To be fixed version
Timing problems with Frequency counter	can be ignored	---	with 2018-10-01 update

Table 2: Known Issues

4.4 Requirements

4.4.1 Software

Software	Version	Note
Vivado	2018.3	needed
SDK	2018.3	needed
PetaLinux	2018.3	needed
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
TE0715-03-15-1C	03_15_1c_1gb	REV03|REV02|REV01	1GB	32MB	NA	NA	NA
TE0715-03-15-1I	03_15_1i_1gb	REV03|REV02|REV01	1GB	32MB	NA	NA	NA
TE0715-03-15-2I	03_15_2i_1gb	REV03|REV02|REV01	1GB	32MB	NA	NA	NA
TE0715-03-30-1C	03_30_1c_1gb	REV03|REV02|REV01	1GB	32MB	NA	NA	NA

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

Module Model	Board Part Short Name	PCB Revision Support	DDR	QSPI Flash	EMMC	Others	Notes
TE0715-03-30-1I	03_30_1i_1gb	REV03|REV02|REV01	1GB	32MB	NA	NA	NA
TE0715-03-30-3E	03_30_3e_1gb	REV03|REV02|REV01	1GB	32MB	NA	NA	NA
TE0715-04-15-1C	04_15_1c_1gb	REV04	1GB	32MB	NA	NA	Low Power DDR
TE0715-04-15-1I	04_15_1i_1gb	REV04	1GB	32MB	NA	NA	Low Power DDR
TE0715-04-15-1I3	04_15_1i_1gb	REV04	1GB	32MB	NA	NA	Low Power DDR
TE0715-04-15-2I	04_15_2i_1gb	REV04	1GB	32MB	NA	NA	Low Power DDR
TE0715-04-30-1C	04_30_1c_1gb	REV04	1GB	32MB	NA	NA	Low Power DDR
TE0715-04-30-1I	04_30_1i_1gb	REV04	1GB	32MB	NA	NA	Low Power DDR
TE0715-04-30-1I3	04_30_1i_1gb	REV04	1GB	32MB	NA	NA	Low Power DDR
TE0715-04-30-3E	04_30_3e_1gb	REV04	1GB	32MB	NA	NA	Low Power DDR
TE0715-04-12S-1C	04_12s_1c_1gb	REV04	1GB	32MB	NA	NA	Low Power DDR
TE0715-04-30-1IA	04_30_1i_1gb	REV04	1GB	32MB	NA	NA	Low Power DDR. Micron Flash

Table 4: Hardware Modules

Design supports following carriers:

Carrier Model	Notes
TE0701	
TE0703	used as reference carrier
TE0705	
TE0706	
TEBA0841-02	

Table 5: Hardware Carrier
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

Table 6: Additional Hardware

4.5 Content

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

4.5.1 Design Sources

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

Table 7: Design sources

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

4.5.2 Additional Sources

Type	Location	Notes
SI5338	<design name>/misc/Si5338	SI5338 Project with current PLL Configuration
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
DebugProbes-File	*.ltx	Definition File for Vivado/Vivado Labtools Debugging Interface
Diverse Reports	---	Report files in different formats
Hardware-Platform-Specification-Files	*.hdf	Exported Vivado Hardware Specification for SDK/HSI and PetaLinux
LabTools Project-File	*.lpr	Vivado Labtools Project File
OS-Image	*.ub	Image with Linux Kernel (On Petalinux optional with Devicetree and RAM-Disk)
Software-Application-File	*.elf	Software Application for Zynq or MicroBlaze Processor Systems

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

4.5.4 Download

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

Reference Design is available on:

- TE0715 "Test Board" Reference Design[3]

3 https://shop.trenz-electronic.de/Download/?path=Trenz_Electronic/Modules_and_Module_Carriers/4x5/TE0715/Reference_Design/2018.3/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.

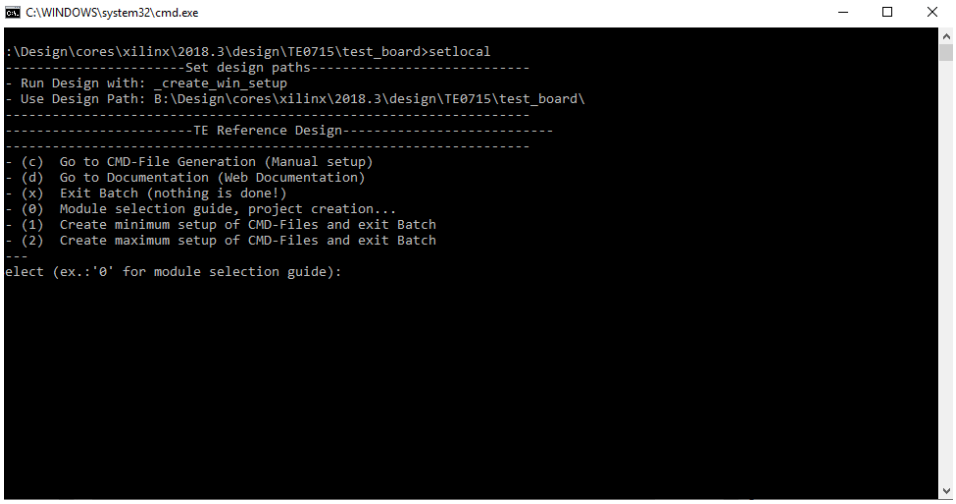
See also:

- 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. Run _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"
   b. Note: Select correct one, see TE Board Part Files[8]
5. Create HDF and export to prebuilt folder

4 https://wiki.trenz-electronic.de/display/PD/Xilinx+Development+Tools#XilinxDevelopmentTools-XilinxSoftware-BasicUserGuides
5 https://wiki.trenz-electronic.de/display/PD/Vivado+Projects+-+TE+Reference+Design
6 https://wiki.trenz-electronic.de/display/PD/Project+Delivery+-+Xilinx+devices
7 https://wiki.trenz-electronic.de/display/PD/Project+Delivery+-+Xilinx+devices#ProjectDelivery-Xilinxdevices-Currentlylimitationsoffunctionality
8 https://wiki.trenz-electronic.de/display/PD/TE+Board+Part+Files

   a. Run on Vivado TCL: TE::hw_build_design -export_prebuilt
      Note: Script generate design and export files into \prebuilt\hardware\<short dir>. Use GUI is the same, except file export to prebuilt folder
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[9]
      i. Use TE Template from /os/petalinux
         Note: run init_config.sh before you start petalinux config. This will set correct temporary path variable.
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 HSI/SDK
   a. Run on Vivado TCL: TE::sw_run_hsi
      Note: Scripts generate applications and bootable files, which are defined in "sw_lib\apps_list.csv"
   b. (alternative) Start SDK with Vivado GUI or start with TE Scripts on Vivado TCL: TE::sw_run_sdk
      Note: See SDK Projects[10]

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

6 Launch

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 lunch.
TE0715-0x-30-xx only: HP IO Banks max power supply voltage is 1.8V.

6.1 Programming

Xilinx documentation for programming and debugging: Vivado/SDK/SDSoC-Xilinx Software Programming and Debugging[11]

6.1.1 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 (zynq_fsbl_flash) on setup
   optional "TE::pr_program_flash_binfile -swapp hello_te0715" possible
4. Copy image.ub on SD-Card
5. Set Boot Mode to QSPI-Boot and insered SD.
   - Depends on Carrier, see carrier TRM.

6.1.2 SD

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

6.1.3 JTAG

Not used on this Example.

6.2 Usage

1. Prepare HW like described on section Programming(see page 15)
2. Connect UART USB (most cases same as JTAG)
3. Select SD Card as Boot Mode (or QSPI - depending on step 1)
   Note: See TRM of the Carrier, which is used.

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

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 1 Bus type: i2cdetect -y -r 1
   b. RTC check: dmesg | grep rtc
   c. ETH0 works with udhcpc
4. Option Features
   a. Webserver to get access to Zynq
      i. insert IP on web browser to start web interface
   b. init.sh scripts
      i. add init.sh script on SD, content will be load automatically on startup (template included in ./misc/SD)

6.2.2 Vivado HW Manager

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

- Monitoring:
  - Si5338 CLKS:
    - Set radix from VIO signals to unsigned integer. Note: Frequency Counter is inaccurate and displayed unit is Hz
    - MGT CLK is configured to 125MHz by default, FCLK is not configured by default (optional possible over FSBL → 50MHz on delivered configuration, see FSBL description).

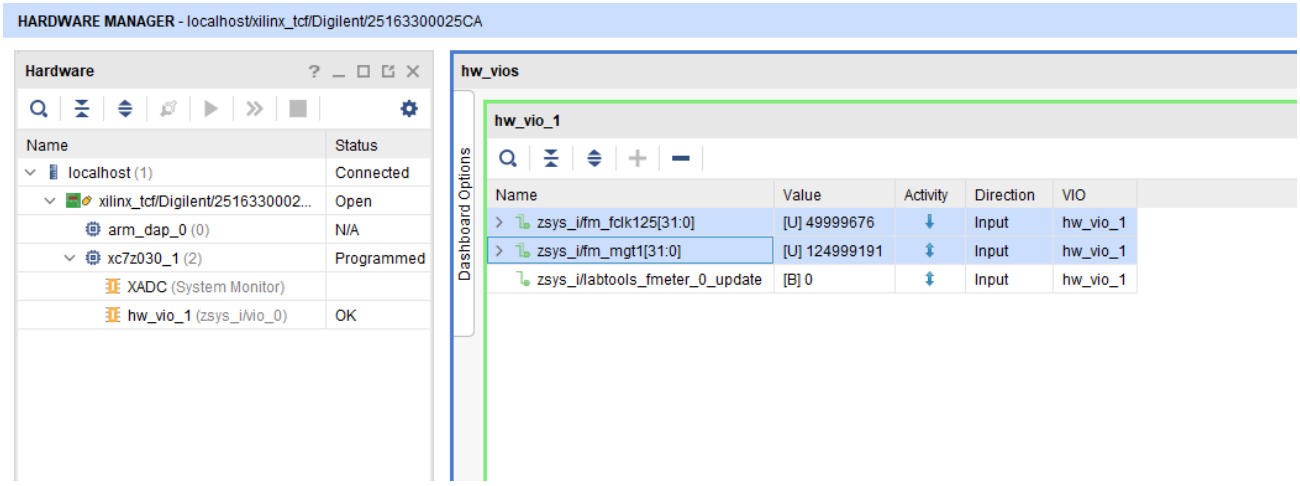


Figure 1: Vivado Hardware Manager

7 System Design - Vivado

7.1 Block Design

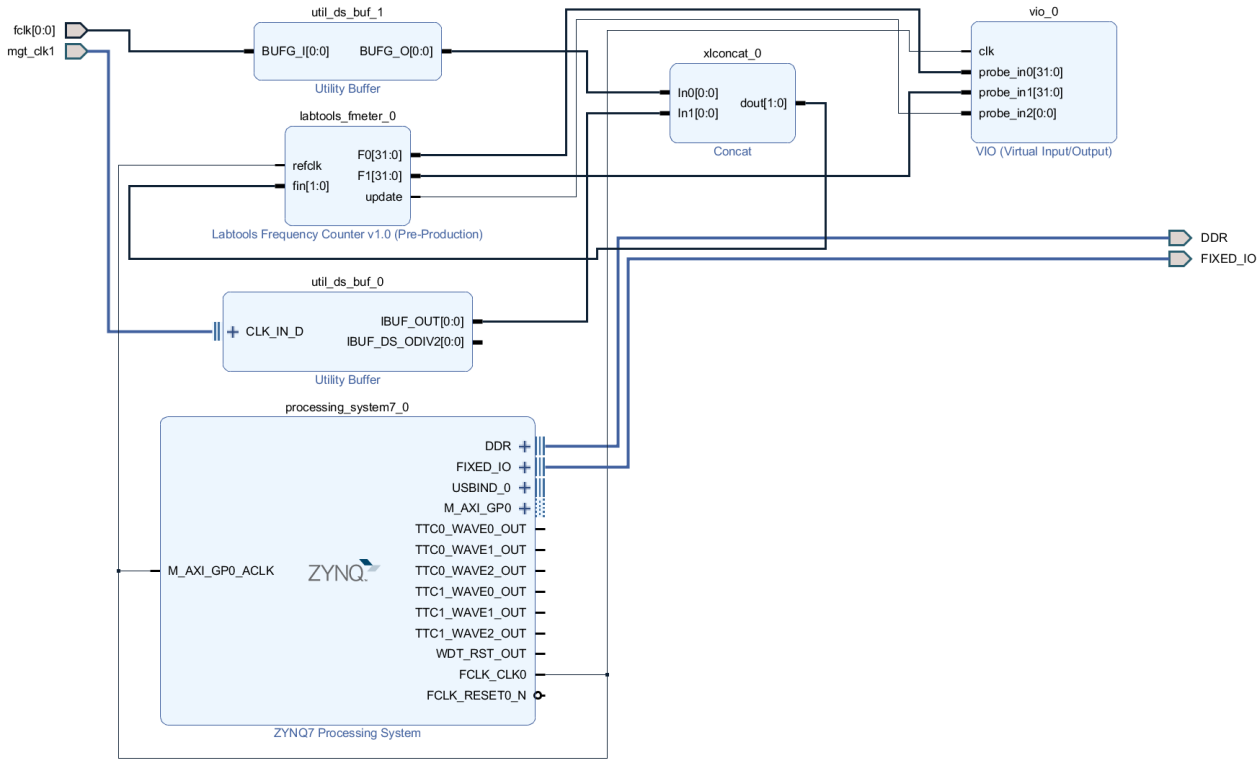


Figure 2: Block Design

7.1.1 PS Interfaces

Activated interfaces:

Type	Note
DDR	---
QSPI	MIO
I2C1	MIO
UART0	MIO
GPIO	MIO
ETH, USB Rst	MIO

Type	Note
SD0	MIO
USB0	MIO
ETH0	MIO
TTC0..1	EMIO
WDT	EMIO

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 CONFIG_VOLTAGE 3.3 [current_design]
set_property CFGBVS VCCO [current_design]

set_property BITSTREAM.CONFIG.USR_ACCESS TIMESTAMP [current_design]
```

_i_unused_io.xdc

```
set_property BITSTREAM.CONFIG.UNUSEDPIN PULLNONE [current_design]
```

7.2.2 Design specific constrain

_i_io.xdc

```
set_property PACKAGE_PIN K2 [get_ports {fclk[0]}]
set_property IOSTANDARD LVCMOS18 [get_ports {fclk[0]}]
set_property CLOCK_DEDICATED_ROUTE FALSE [get_nets fclk_IBUF[0]]
```

_i_timing.xdc

```
# for fmeter only
# set_false_path -from [get_clocks clk_fpga_0] -to [get_clocks {zsys_i/util_ds_buf_0/
U0/IBUF_OUT[0]}]
# set_false_path -from [get_clocks {zsys_i/util_ds_buf_0/U0/IBUF_OUT[0]}] -to
[get_clocks clk_fpga_0]
# set_false_path -from [get_clocks clk_fpga_0] -to [get_clocks {zsys_i/util_ds_buf_1/
U0/BUFG_O[0]}]
```

8 Software Design - SDK/HSI

For SDK project creation, follow instructions from:

- SDK Projects[12]

8.1 Application

Template location: ./sw_lib/sw_apps/

8.1.1 zynq_fsbl

TE modified 2018.3 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_*
    - SI5338 Configuration

8.1.2 zynq_fsbl_flash

TE modified 2018.3 FSBL

General:

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

8.1.3 hello_te0715

Hello TE0715 is a Xilinx Hello World example as endless loop instead of one console output.

8.1.4 u-boot

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

12 https://wiki.trenz-electronic.de/display/PD/SDK+Projects

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

Change platform-top.h:

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

```
#include <configs/platform-auto.h>
#define CONFIG_SYS_BOOTM_LEN 0xF000000
#define DFU_ALT_INFO_RAM \
                "dfu_ram_info=" \
        "setenv dfu_alt_info " \
        "image.ub ram $netstart 0x1e00000\0" \
        "dfu_ram=run dfu_ram_info && dfu 0 ram 0\0" \
        "thor_ram=run dfu_ram_info && thordown 0 ram 0\0"

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


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



/*Define CONFIG_ZYNQ_EEPROM here and its necessaries in u-boot menuconfig if you had
EEPROM memory. */
//#ifdef CONFIG_ZYNQ_EEPROM
//#define CONFIG_SYS_I2C_EEPROM_ADDR_LEN         1
//#define CONFIG_SYS_I2C_EEPROM_ADDR             0x50
//#define CONFIG_SYS_EEPROM_PAGE_WRITE_BITS      4
//#define CONFIG_SYS_EEPROM_PAGE_WRITE_DELAY_MS  5
//#define CONFIG_SYS_EEPROM_SIZE                 1024 /* Bytes */
//#define CONFIG_SYS_I2C_MUX_ADDR                0x74
//#define CONFIG_SYS_I2C_MUX_EEPROM_SEL          0x4
//#endif

#define CONFIG_ZYNQ_EEPROM
#ifdef CONFIG_ZYNQ_EEPROM
#define CONFIG_SYS_I2C_EEPROM_ADDR_LEN  1
#define CONFIG_CMD_EEPROM
#define CONFIG_ZYNQ_EEPROM_BUS          0
#define CONFIG_ZYNQ_GEM_EEPROM_ADDR     0x50
#define CONFIG_ZYNQ_GEM_I2C_MAC_OFFSET  0xFA
#endif
```

9.3 Device Tree

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

/* default */

/* 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>;
    };
};



/* ETH PHY */
&gem0 {

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


/* USB PHY */
/{
    usb_phy0: usb_phy@0 {
        compatible = "ulpi-phy";
        //compatible = "usb-nop-xceiv";
        #phy-cells = <0>;
        reg = <0xe0002000 0x1000>;
        view-port = <0x0170>;
        drv-vbus;
    };
};

&usb0 {
    dr_mode = "host";
    //dr_mode = "peripheral";
    usb-phy = <&usb_phy0>;
};
```

```
/* I2C */
// i2c PLL: 0x70, i2c eeprom: 0x50

&i2c1 {
    rtc@6F {        // Real Time Clock
       compatible = "isl12022";
       reg = <0x6F>;
   };

};
```

9.4 Kernel

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

Changes:

- CONFIG_RTC_DRV_ISL12022=y

9.5 Rootfs

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

Changes:

- CONFIG_i2c-tools=y
- CONFIG_busybox-httpd=y (for web server app)
- CONFIG_usbutils=y

9.6 Applications

9.6.1 startup

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

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

9.6.2 webfwu

Webserver application accemble for Zynq access. Need busybox-httpd

10 Additional Software

10.1 SI5338

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

General documentation how you work with these project will be available on Si5338[14]

14 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
2019-05-09	v.32(see page 6)	John Hartfiel[15]	• Release 2018.3 • FSBL Rework • Script rework • some optional features
2018-10-01	v.31	John Hartfiel	• Release 2018.2 • Redesign Board Part Files • New activate SI5338 example over FSBL • small Design changes • Update Documentation Style
2019-04-06	v.30	John Hartfiel	• New assembly variant
2018-03-27	v.29	John Hartfiel	• Bugfix Board Part Files
2018-02-13	v.28	John Hartfiel	• Release 2017.4
2017-11-10	v.22	John Hartfiel	• Design Update with new options • Add Si5338 section • Update FSBL section
2017-10-19	v.21	John Hartfiel	• Download Update
2017-10-19	v.20	John Hartfiel	• Document style update
2017-10-06	v.18	John Hartfiel	• Text correction • Update Launch section • Supported PCBs
2017-10-02	v.14	John Hartfiel	• Document update on Prebuilt section

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

Date	Document Revision	Authors	Description
2017-09-28	v.13	John Hartfiel	• Initial Release 2017.2
--	all	John Hartfiel[16]	--

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

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.

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

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[17]. 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[18] 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)[19].

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