TE0720 HDMI701

Table of contents

Overview

Zynq PS Design with Linux for TE0701 with HDMI support.

Refer to http://trenz.org/te0720-info for the current online version of this manual and other available documentation.

Key Features

- Vitis/Vivado 2019.2
- PetaLinux/Ubuntu/Debian
- SD
- ETH (use EEPROM MAC)
- USB
- I2C
- RTC
- VIO PHY LED
- HDMI
- FSBL for EEPROM MAC and CPLD access and HDMI DMA
- Special FSBL for QSPI Programming
- TE0701 (only supported)

Revision History

Date	Vivado	Project Built	Authors	Description
2020-03-25	2019.2	TE0720-HDMI701_noprebuilt-vivado_2019.2-build_8_20200325075641.zip TE0720-HDMI701-vivado_2019.2-build_8_20200325075631.zip	Mohsen Chamanbaz/John Hartfiel	• script update
2020-02-27	2019.2	TE0720-HDMI701_noprebuilt-vivado_2019.2-build_7_20200227113153.zip TE0720-HDMI701-vivado_2019.2-build_7_20200227113133.zip	Mohsen Chamanbaz	• update vivado 2019.2 • Ubuntu/Debian as root file system
2017-12-04	2017.2	te0720-HDMI701_noprebuilt-vivado_2017.2-build_05_20171204082246.zip te0720-HDMI701-vivado_2017.2-build_05_20171204081435.zip	Oleksandr Kiyenko	• initial release

Design Revision History

Release Notes and Know Issues

Issues	Description	Workaround	To be fixed version
No known issues	---	--	---

Known Issues

Requirements

Software

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

Software

Hardware

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

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

Design supports following modules:

Module Model	Board Part Short Name	PCB Revision Support	DDR	QSPI Flash	EMMC	Others	Notes
TE0720-03-2IF	2if_1gb	REV03|REV02	1GB	32MB	4GB	NA	NA
TE0720-03-2IFC3	2if_1gb	REV03|REV02	1GB	32MB	4GB	2.5 mm connectors	NA
TE0720-03-2IFC8	2if_1gb	REV03|REV02	1GB	32MB	32GB	NA	NA
TE0720-03-1QF	1qf_1gb	REV03|REV02	1GB	32MB	4GB	NA	NA
TE0720-03-1CF	1cf_1gb	REV03|REV02	1GB	32MB	4GB	NA	NA
TE0720-03-1CFA	1cf_1gb	REV03|REV02	1GB	32MB	8GB	NA	NA
TE0720-03-2EF	2ef_1gb	REV03|REV02	1GB	32MB	4GB	NA	NA
~~TE0720-03-1CR~~	1cr_256mb	REV03|REV02	256MB	32MB	NA	NA	not supported on this demo (changes into FSBL and device tree template are need)
~~TE0720-03-L1IF~~	l1if_512mb	REV03|REV02	512MB	32MB	4GB	NA	LP DDR3, not supported on this demo (changes into FSBL and device tree template are need)
TE0720-03-14S-1C	14s_1gb	REV03|REV02	1GB	32MB	4GB	NA	NA
TE0720-03-1QFA	1qf_1gb	REV03|REV02	1GB	32MB	4GB	NA	Micron Flash
TE0720-03-2IFA	2if_1gb	REV03|REV02	1GB	32MB	4GB	NA	Micron Flash
TE0720-03-1QFL	1qf_1gb	REV03|REV02	1GB	32MB	4GB	2.5 mm connectors	NA

Hardware Modules

Design supports following carriers:

Carrier Model	Notes
TE0701	• See restrictions on usage with 7 Series Carriers: 4 x 5 SoM Carriers

Hardware Carrier

Additional HW Requirements:

Additional Hardware	Notes
Monitor with HDMI	Tested with DELL U2412M
Micro USB to USB A Adapter	Adapter for USB Hub
USB HUB	To connnect Mouse and Keyboard simultaneously
Keyboard	need for Ubuntu/Debian GUI
Mouse	need for Ubuntu/Debian GUI

Additional Hardware

Content

For general structure and of the reference design, see Project Delivery - AMD devices

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

Design sources

Additional Sources

Type	Location	Notes
mkdebian_stretch.sh	<design name>/os/petalinux	create Debian image
mkubuntu_BionicBeaver.sh	<design name>/os/petalinux	create Ubuntu image

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
DebugProbes-File	*.ltx	Definition File for Vivado/Vivado Labtools Debugging Interface
Debian/Ubuntu SD-Image	*.img	Ubuntu/Debian Image for SD-Card (separate available on the download area)
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

Prebuilt files (only on ZIP with prebult content)

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:

- TE0720 "HDMI701" Reference Design Download Area

Design Flow

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

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

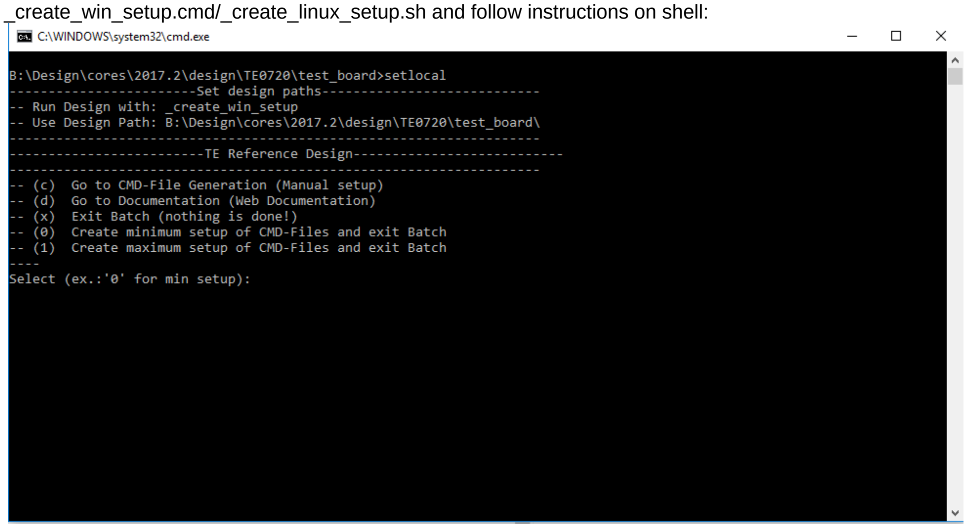
See also:

- 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/SDK GUI. For currently Scripts limitations on Win and Linux OS see: Project Delivery Currently limitations of functionality

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



2. Press 0 and enter for minimum setup
3. (optional Win OS) Generate Virtual Drive or use short directory for the reference design (for example x:\<design name>)
4. Create Project
    a. Select correct device and Xilinx install path on "design_basic_settings.cmd" and create Vivado project with "vivado_create_project_guimode.cmd"
       Note: Select correct one, see TE Board Part Files
5. Create XSA and export to prebuilt folder
    a. Run on Vivado TCL: TE::hw_build_design -export_prebuilt
       Note: Script generate design and export files into \prebuilt\hardware\<short dir>. Use GUI is the same, except file export to prebuilt folder
6. Create Linux (uboot.elf and image.ub) with exported 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
        i. Use TE Template from /os/petalinux
    c. Build the Debian image/Ubuntu image file with executing the "mkdebian_stretch.sh"/"mkubuntu_BionicBeaver.sh" file in Linux Terminal
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

Launch

Programming

Check Module and Carrier TRMs for proper HW configuration before you try any design.

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

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 folder>/_binaries_<Articel Name>) with subfolder (boot_linux)

QSPI

Not used on this Example.

SD

1. Format the SD Card with SD Card Formatter or other tool
2. Write the Debian image or Ubuntu image file on SD Card with Win32DiskImager
3. Copy Petalinux 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
    - or use prebuilt file location, see <design_name>/prebuilt/readme_file_location.txt
4. Set Boot Mode to SD-Boot.
    - Depends on Carrier, see carrier TRM.
5. Insert SD-Card in SD-Slot.

JTAG

Not used on this Example.

Usage

1. Prepare HW like described on section TE0720 HDMI701#Programming
2. Connect UART USB (most cases same as JTAG)
3. Select SD Card as Boot Mode
   Note: On TE0701 Default Firmware Boot Mode is selected via SD card (insered SD Card for SD Boot Mode)
4. Connect HDMI to Monitor
5. Connect USB Adapter with Hub and Mouse+Keyboard
6. 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

Linux

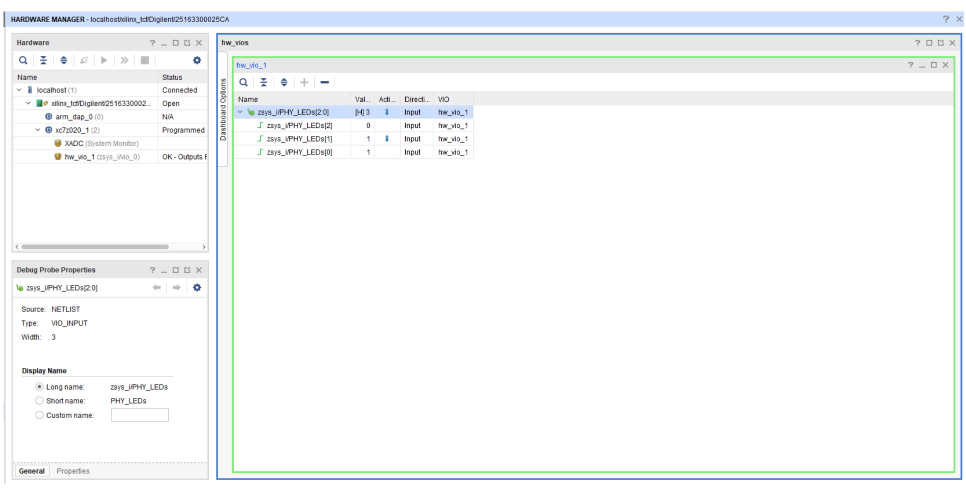
1. Open Serial Console (e.g. putty)
    a. Speed: 115200
    b. COM Port: Win OS, see device manager, Linux OS see dmesg |grep tty (UART is *USB1)
2. Linux Console:
   Note: Wait until Linux boot finished For Linux Login use:
    a. User Name: root
    b. Password: root
3. You can use Linux shell now.
    a. I2C 0 Bus type: i2cdetect -y -r 0
    b. I2C 1 Bus type: i2cdetect -y -r 1
    c. RTC check: dmesg | grep rtc
    d. ETH0 works with udhcpc
    e. USB: insert USB device
4. Debian Desktop
    a. Debian Desktop will be started automatically
    b. Use connected mouse + keyboard for interaction with GUI
    c. Web Browser Dillo open console and type dillo or use browser
    d. open console and start video or audio with "mplayer <video or audio file>"
5. Ubuntu Desktop
    a. Ubuntu Desktop will be started automatically
    b. Use connected mouse + keyboard for interaction with GUI
    c. Web Browser Mozilla firefox can be used.
    d. Audio or Vider file can also be performed directly in GUI.

HDMI Monitor

Second Linux GUI is displayed on HDMI monitor.

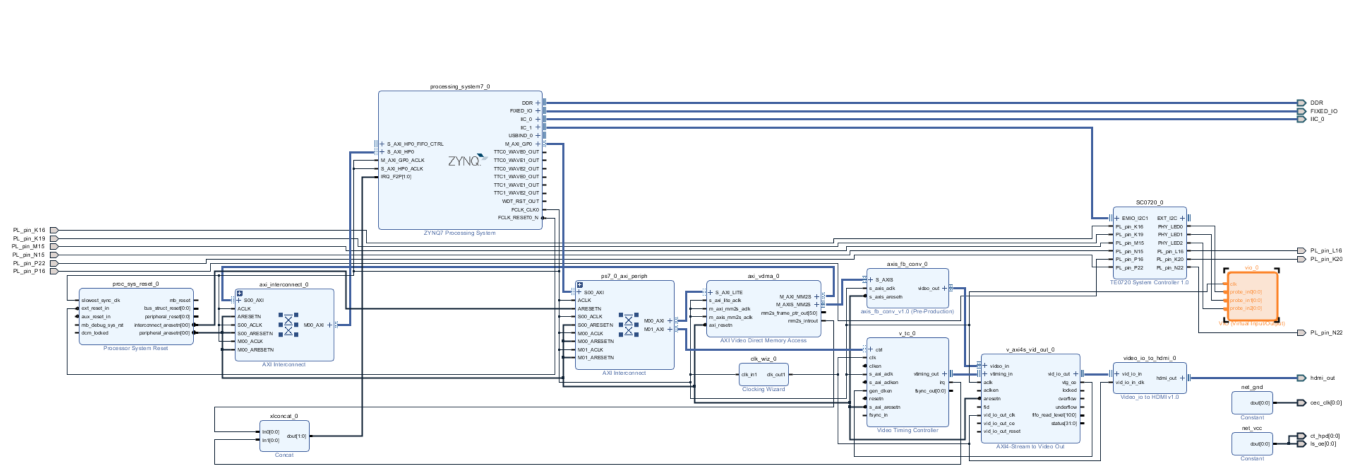
Vivado HW Manager

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



System Design - Vivado

Block Design



Block Design

PS Interfaces

Type	Note
DDR	---
QSPI	MIO
ETH0	MIO
USB0	MIO
SD0	MIO
SD1	MIO
UART0	MIO
UART1	MIO
I2C0	EMIO TE0701
I2C1	EMIO TE0720
GPIO	MIO
TTC0..1	EMIO
WDT	EMIO

PS Interfaces

Constrains

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
```

_i_common.xdc

```
#
set_property BITSTREAM.CONFIG.UNUSEDPIN PULLUP [current_design]
```

Design specific constrain

_i_TE0720-SC.xdc

```
#
# Constraints for System controller support logic
#
set_property PACKAGE_PIN K16 [get_ports PL_pin_K16]
set_property PACKAGE_PIN K19 [get_ports PL_pin_K19]
set_property PACKAGE_PIN K20 [get_ports PL_pin_K20]
set_property PACKAGE_PIN L16 [get_ports PL_pin_L16]
set_property PACKAGE_PIN M15 [get_ports PL_pin_M15]
set_property PACKAGE_PIN N15 [get_ports PL_pin_N15]
set_property PACKAGE_PIN N22 [get_ports PL_pin_N22]
set_property PACKAGE_PIN P16 [get_ports PL_pin_P16]
set_property PACKAGE_PIN P22 [get_ports PL_pin_P22]

#
# If Bank 34 is not 2.5V Powered need change the IOSTANDARD
#
set_property IOSTANDARD LVCMOS25 [get_ports PL_pin*]
```

_i_hdmi.xdc

```
#
# TE0701 I2C Bus
#
set_property PACKAGE_PIN W20 [get_ports IIC_0_scl_io]
set_property PACKAGE_PIN W21 [get_ports IIC_0_sda_io]
set_property IOSTANDARD LVCMOS25 [get_ports IIC_0_scl_io]
set_property IOSTANDARD LVCMOS25 [get_ports IIC_0_sda_io]

#
# ADV7511 Interface
#
set_property PACKAGE_PIN N20 [get_ports hdmi_out_clk]
set_property PACKAGE_PIN N19 [get_ports hdmi_out_de]
set_property PACKAGE_PIN R19 [get_ports hdmi_out_hsync]
set_property PACKAGE_PIN T19 [get_ports hdmi_out_vsync]
set_property PACKAGE_PIN T18 [get_ports {hdmi_out_data[0]}]
set_property PACKAGE_PIN R18 [get_ports {hdmi_out_data[1]}]
set_property PACKAGE_PIN R21 [get_ports {hdmi_out_data[2]}]
set_property PACKAGE_PIN R20 [get_ports {hdmi_out_data[3]}]
set_property PACKAGE_PIN M22 [get_ports {hdmi_out_data[4]}]
set_property PACKAGE_PIN K21 [get_ports {hdmi_out_data[5]}]
set_property PACKAGE_PIN M21 [get_ports {hdmi_out_data[6]}]
set_property PACKAGE_PIN J20 [get_ports {hdmi_out_data[7]}]
set_property PACKAGE_PIN T17 [get_ports {hdmi_out_data[8]}]
set_property PACKAGE_PIN J22 [get_ports {hdmi_out_data[9]}]
set_property PACKAGE_PIN T16 [get_ports {hdmi_out_data[10]}]
set_property PACKAGE_PIN J21 [get_ports {hdmi_out_data[11]}]
set_property IOSTANDARD LVCMOS25 [get_ports hdmi_*]

set_property PACKAGE_PIN AB16 [get_ports {cec_clk[0]}]
set_property PACKAGE_PIN AB17 [get_ports {ct_hpd[0]}]
set_property PACKAGE_PIN AA16 [get_ports {ls_oe[0]}]
set_property IOSTANDARD LVCMOS25 [get_ports {cec_clk[0]}]
set_property IOSTANDARD LVCMOS25 [get_ports {ct_hpd[0]}]
set_property IOSTANDARD LVCMOS25 [get_ports {ls_oe[0]}]
```

Software Design - Vitis

For SDK project creation, follow instructions from:

Vitis

Application

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_*
    - READ MAC from EEPROM and make Address accessible by UBOOT (need copy defines on uboot platform-top.h)
    - CPLD access
    - Read CPLD Firmware and SoC Type
    - Configure Marvell PHY
    - USB PHY Reset
    - Configure LED usage

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_*
    - READ MAC from EEPROM and make Address accessible by UBOOT (need copy defines on uboot platform-top.h)
    - CPLD access
    - Read CPLD Firmware and SoC Type
    - Configure Marvell PHY
    - Configure ADV7511
    - Configure Video Timing Controller core
    - Configure VDMA core and enable transfers

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

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**

Select Image Packaging Configuration ==> Root filesystem type ==> Select SD Card

Changes:

- # CONFIG_SUBSYSTEM_ROOTFS_INITRAMFS is not set
- # CONFIG_SUBSYSTEM_ROOTFS_INITRD is not set
- # CONFIG_SUBSYSTEM_ROOTFS_JFFS2 is not set
- # CONFIG_SUBSYSTEM_ROOTFS_NFS is not set
- CONFIG_SUBSYSTEM_ROOTFS_SD=y
- # CONFIG_SUBSYSTEM_ROOTFS_OTHER is not set

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:

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


#define UBOOT_ENV_MAGIC 0xCAFEBABE
#define UBOOT_ENV_MAGIC_ADDR 0xFFFFFC00
#define UBOOT_ENV_ADDR 0xFFFFFC04


#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

/*Dependencies for ENV to be stored in EEPROM. Ensure environment fits in eeprom size*/
#ifdef CONFIG_ENV_IS_IN_EEPROM
#define CONFIG_SYS_I2C_EEPROM_ADDR_LEN         1
#define CONFIG_SYS_I2C_EEPROM_ADDR             0x54
#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_PREBOOT    "echo U-BOOT for petalinux;echo importing env from FSBL shared area at 0xFFFFFC00; if
itest *0xFFFFFC00 == 0xCAFEBABE; then echo Found valid magic; env import -t 0xFFFFFC04; fi;setenv preboot;
echo; dhcp"
```

Device Tree

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

```
/ {
    memory {    // Reduce memory for framebuffers
        device_type = "memory";
        reg = <0x0 0x3FC00000>;  // Reduce memory for 1GB assembly variant
      //  reg = <0x0 0x1FC00000>;  // Reduce memory for 1GB assembly variant
      //  reg = <0x0 0x0FC00000>;  // Reduce memory for 1GB assembly variant
    };


    framebuffer0: framebuffer@0x3FC00000 {      // HDMI out
        compatible = "simple-framebuffer";
        reg = <0x3FC00000 (1280 * 720 * 4)>;    // 720p
        width = <1280>;                         // 720p
        height = <720>;                         // 720p
        stride = <(1280 * 4)>;                  // 720p
        format = "a8b8g8r8";
    };

/*
    framebuffer0: framebuffer@0x1FC00000 {      // HDMI out
        compatible = "simple-framebuffer";
        reg = <0x1FC00000 (1280 * 720 * 4)>;    // 720p
        width = <1280>;                         // 720p
        height = <720>;                         // 720p
        stride = <(1280 * 4)>;                  // 720p
        format = "a8b8g8r8";
    };
*/
/*
    framebuffer0: framebuffer@0x0FC00000 {      // HDMI out
        compatible = "simple-framebuffer";
        reg = <0x0FC00000 (1280 * 720 * 4)>;    // 720p
        width = <1280>;                         // 720p
        height = <720>;                         // 720p
        stride = <(1280 * 4)>;                  // 720p
        format = "a8b8g8r8";
    };
*/
};

&axi_vdma_0 {
   status = "disabled";
};

&v_tc_0 {
    //xilinx-vtc: probe of 43c20000.v_tc failed with error -2
    status = "disabled";
};


/* 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 {
    phy-handle = <&phy0>;
    mdio {
        #address-cells = <1>;
        #size-cells = <0>;
        phy0: 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 need I2C1 connected to te0720 system controller ip */
&i2c1 {

    iexp@20 {       // GPIO in CPLD
        #gpio-cells = <2>;
        compatible = "ti,pcf8574";
        reg = <0x20>;
        gpio-controller;
    };

    iexp@21 {       // GPIO in CPLD
        #gpio-cells = <2>;
        compatible = "ti,pcf8574";
        reg = <0x21>;
        gpio-controller;
    };

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

Kernel

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

Changes:

- RTC_DRV_ISL12022
- CONFIG_FB_SIMPLE
- CONFIG_LOGO
- CONFIG_LOGO_LINUX_MONO
- CONFIG_LOGO_LINUX_VGA16
- CONFIG_LOGO_LINUX_CLUT224

Rootfs

File system will be generated with Debian script or Ubuntu script (mkdebian_stretch.sh/mkubuntu_BionicBeaver.sh)

Applications

Applications will be generated with Debian script or Ubuntu script (mkdebian_stretch.sh/mkubuntu_BionicBeaver.sh)

Additional Software

No additional software is needed.

Appx. A: Change History and Legal Notices

Document Change History

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

Date	Document Revision	Authors	Description

Error rendering macro 'page-info' Ambiguous method overloading for method jdk.proxy244.$Proxy3589#hasContentLevelPermission. Cannot resolve which method to invoke for [null, class java.lang.String, class com.atlassian.confluence.pages.Page] due to overlapping prototypes between: [interface com.atlassian.confluence.user.ConfluenceUser, class java.lang.String, class com.atlassian.confluence.core.ContentEntityObject] [interface com.atlassian.user.User, class java.lang.String, class com.atlassian.confluence.core.ContentEntityObject]	**Error rendering macro 'page-info'** Ambiguous method overloading for method jdk.proxy244.$Proxy3589#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]	**Error rendering macro 'page-info'** Ambiguous method overloading for method jdk.proxy244.$Proxy3589#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]	• script update
2020-02-27	v.11	Mohsen Chamanbaz	• 2019.2 release • Ubuntu/Debian as root file system
2018-02-13	v.10	John Hartfiel	• 2017.2 release

	All	**Error rendering macro 'page-info'** Ambiguous method overloading for method jdk.proxy244.$Proxy3589#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

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.proxy244.$Proxy3589#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]