TE0727 Zynqberry Demo1

Revision v.5

Exported on 2021-10-19

Online version of this document:

https://wiki.trenz-electronic.de/pages/viewpage.action?pageId=134775531

1 Table of Contents

2 Table of Figures

3 Table of Tables

$body

4 Overview

Zynq PS Design with Linux Example and Camera Demo.

Refer to http://trenz.org/te0727-info for the current online version of this manual and other available documentation.

4.1 Key Features

- Vitis/Vivado 2019.2
- RPI Camera 1.3 or 2.1
- HDMI
- PetaLinux
- SD
- USB
- I2C
- Special FSBL for QSPI programming

4.2 Revision History

Date	Vivado	Project Built	Authors	Description
2020-11-24	2019.2	TE0727-zbzerodemo1_noprebuilt-vivado_2019.2-build_15_20201124064113.zip TE0727-zbzerodemo1-vivado_2019.2-build_15_20201124064045.zip	Oleksandr Kiyenko/ John Hartfiel	• initial release

Table 1: Design Revision History

4.3 Release Notes and Know Issues

Issues	Description	Workaround	To be fixed version
init.sh	automatically camera selection failed	select camera manually on init.sg	---

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	Others	Notes
TE0727-01-010-1C	10_512MB	REV01	512 MB DDR 3L	16MB		small design modification needed (I2C for camera)
TE0727-02-41C34	10_512MB	REV01	512 MB DDR 3L	16MB		

Table 4: Hardware Modules

Design supports following carriers:

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

Carrier Model	Notes

Table 5: Hardware Carrier

Additional HW Requirements:

Additional Hardware	Notes
USB Power	Use USB2.0 or higher for power supply via USB
USB Cable	Connect to USB2 or better USB3 Hub for proper power supply over USB
Raspberry Pi Camera Rev 1.3 or Camera Rev 2.1	
Monitor	DELL Model Number: U2412M
HDMI Cable	--
HDMI to Mini HTMI adapter	

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	Vivado Project will be generated by TE Scripts

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

Type	Location	Notes
	<design name>/ip_lib	
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/init_script	Additional Initialization Script for Linux (used to enable camera)

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

File	File-Extension	Description
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/SDK/PetaLinux/SDx version. Do never use different Versions of Xilinx Software for the same Project.

Reference Design is available on:

- TE0727 "Zynqberry Demo1" Reference Design[3]

3 https://shop.trenz-electronic.de/Download/?path=Trenz_Electronic/Modules_and_Module_Carriers/3.05x6.5/TE0727/Reference_Design/2019.2/zbzerodemo1

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.

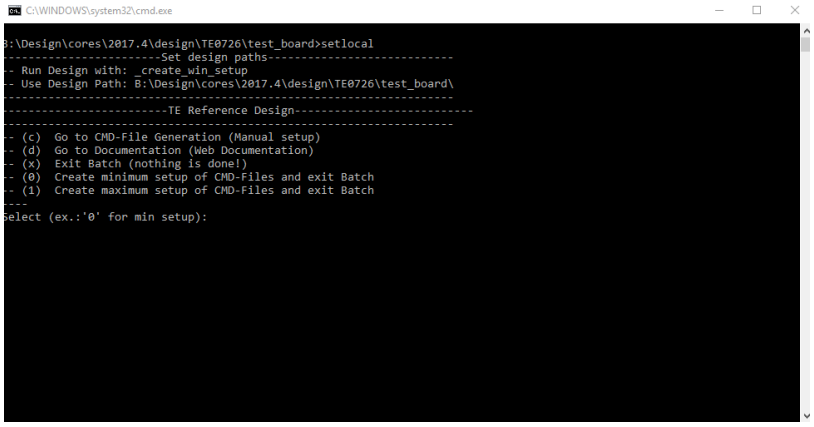
See also:

- 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 be 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 TE Board Part Files[8]
5. Create HDF and export to prebuilt folder
   a. Run on Vivado TCL: TE::hw_build_design -export_prebuilt
      Note: Script generate design and export files into

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

\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
        ii. For 128MB and 64MB only:Netboot Offset must be reduced manually, see Config(see page 21)
7. Add Linux files (uboot.elf and image.ub) to prebuilt folder
    a. "prebuilt\os\petalinux\<DDR size>" or "prebuilt\os\petalinux\<short name>"
    Notes: Scripts select "prebuilt\os\petalinux\<DDR size>", if exist, otherwise "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

1. Connect JTAG and power on the 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 -swapp u-boot
   Note: To program with SDK/Vivado GUI, use special FSBL (zynq_fsbl_flash) on setup
   optional "TE::pr_program_flash -swapp hello_te0726" possible
4. 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[12]
   - For correct prebuilt file location, see <design_name>/prebuilt/readme_file_location.txt
   - Important: Do not copy Boot.bin on SD(is not used see SD note), only other files.
5. Copy init.sh on SD-Card
   - location: <design_name>/misc/sd/
6. Insert SD-Card

11 https://wiki.trenz-electronic.de/display/PD/Xilinx+Development+Tools#XilinxDevelopmentTools-XilinxSoftwareProgrammingandDebugging
12 https://wiki.trenz-electronic.de/display/PD/TE0726+Test+Board#TE0726TestBoard-Getprebuiltbootbinaries

6.1.3 SD

Xilinx Zynq devices in CLG225 package do not support SD Card boot directly from ROM bootloader. Use QSPI for primary boot and SD for secondary boot (u-boot)

6.1.4 JTAG

Not used on this Example.

6.2 Usage

1. Prepare HW like described in section Programming(see page 14)
2. Connect UART USB (most cases same as JTAG)
3. Insert SD Card with image.ub
4. Power On PCB
   Note: 1. Zynq Boot ROM loads FSBL from QSPI into OCM, 2. FSBL loads U-boot from QSPI 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
      Note: Wait until Linux boot finished For Linux Login use:
      i. User Name: root
      ii. Password: root
3. You can use a Linux shell now.
   a. I2C 1 Bus type: i2cdetect -y -r 5
      Bus 0...5 possible
   b. USB: insert USB device
4. Camera stream will be enabled via init.sh script on SD
5. Take image from camera (must be enabled with init.sh scripts):
   a. write image to webserver: fbgrab -d /dev/fb1 /srv/www/camera.png
   b. Display image on host PC: http://<ZynqBerry IP>/camera.png

7 System Design - Vivado

7.1 Block Design

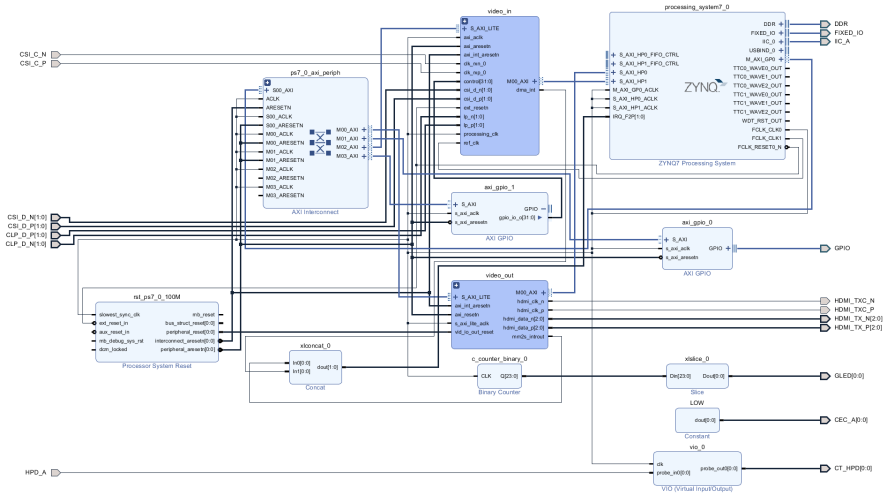


Figure 1: Block Design

7.1.1 PS Interfaces

Activated interfaces:

Type	Note
DDR	---
QSPI	MIO
USB0	MIO,
SD1	MIO
UART1	MIO
I2C0	EMIO
I2C1	MIO
GPIO	MIO / EMIO
USB RST	MIO
TTC0..1	MIO

Type	Note
WDT	MIO
AXI HP0..1	
DMA0..1	

Table 10: PS Interfaces

7.2 Constraints

7.2.1 Basic module constraints

_i_bitgen_common.xdc

```
#
# Common BITGEN related settings for TE0727 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 constraint

_i_common.xdc

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

_i_te0727.xdc

```
set_property PACKAGE_PIN G11 [get_ports {CEC_A[0]}]
set_property IOSTANDARD LVCMOS33 [get_ports {CEC_A[0]}]
set_property PACKAGE_PIN H13 [get_ports {HPD_A}]
set_property IOSTANDARD LVCMOS33 [get_ports {HPD_A}]
set_property PACKAGE_PIN G14 [get_ports {GLED[0]}]
set_property IOSTANDARD LVCMOS33 [get_ports {GLED[0]}]
set_property PACKAGE_PIN G12 [get_ports {IIC_A_scl_io}]
set_property PACKAGE_PIN H12 [get_ports {IIC_A_sda_io}]
```

```
set_property IOSTANDARD LVCMOS33 [get_ports {IIC_A_*}]
set_property PACKAGE_PIN K12 [get_ports {CT_HPD[0]}]
set_property IOSTANDARD LVCMOS33 [get_ports {CT_HPD[0]}]

set_property PACKAGE_PIN F12 [get_ports {HDMI_TXC_P}]
set_property PACKAGE_PIN E13 [get_ports {HDMI_TXC_N}]
set_property PACKAGE_PIN E11 [get_ports {HDMI_TX_P[0]}]
set_property PACKAGE_PIN E12 [get_ports {HDMI_TX_N[0]}]
set_property PACKAGE_PIN G15 [get_ports {HDMI_TX_P[1]}]
set_property PACKAGE_PIN F15 [get_ports {HDMI_TX_N[1]}]
set_property PACKAGE_PIN F14 [get_ports {HDMI_TX_N[2]}]
set_property PACKAGE_PIN F13 [get_ports {HDMI_TX_P[2]}]
set_property IOSTANDARD TMDS_33 [get_ports {HDMI_*}]

set_property PACKAGE_PIN J11 [get_ports {GPIO_tri_io[0]}]
set_property PACKAGE_PIN H11 [get_ports {GPIO_tri_io[1]}]
set_property PACKAGE_PIN J15 [get_ports {GPIO_tri_io[2]}]
set_property PACKAGE_PIN L15 [get_ports {GPIO_tri_io[3]}]
set_property PACKAGE_PIN N13 [get_ports {GPIO_tri_io[4]}]
set_property PACKAGE_PIN P8  [get_ports {GPIO_tri_io[5]}]
set_property PACKAGE_PIN M10 [get_ports {GPIO_tri_io[6]}]
set_property PACKAGE_PIN L12 [get_ports {GPIO_tri_io[7]}]
set_property PACKAGE_PIN M11 [get_ports {GPIO_tri_io[8]}]
set_property PACKAGE_PIN P10 [get_ports {GPIO_tri_io[9]}]
set_property PACKAGE_PIN P9  [get_ports {GPIO_tri_io[10]}
]
set_property PACKAGE_PIN K15 [get_ports {GPIO_tri_io[11]}
]
set_property PACKAGE_PIN M9  [get_ports {GPIO_tri_io[12]}
]
set_property PACKAGE_PIN L13 [get_ports {GPIO_tri_io[13]}
]
set_property PACKAGE_PIN L14 [get_ports {GPIO_tri_io[14]}
]
set_property PACKAGE_PIN M15 [get_ports {GPIO_tri_io[15]}
]
set_property PACKAGE_PIN J14 [get_ports {GPIO_tri_io[16]}
]
set_property PACKAGE_PIN N14 [get_ports {GPIO_tri_io[17]}
]
set_property PACKAGE_PIN K11 [get_ports {GPIO_tri_io[18]}
]
set_property PACKAGE_PIN N9  [get_ports {GPIO_tri_io[19]}
]
set_property PACKAGE_PIN J13 [get_ports {GPIO_tri_io[20]}
]
set_property PACKAGE_PIN H14 [get_ports {GPIO_tri_io[21]}
]
set_property PACKAGE_PIN R10 [get_ports {GPIO_tri_io[22]}
]
set_property PACKAGE_PIN M14 [get_ports {GPIO_tri_io[23]}
]
set_property PACKAGE_PIN P15 [get_ports {GPIO_tri_io[24]}
]
set_property PACKAGE_PIN M12 [get_ports {GPIO_tri_io[25]}
]
```

```
set_property PACKAGE_PIN K13 [get_ports {GPIO_tri_io[26]}
]
set_property PACKAGE_PIN R15 [get_ports {GPIO_tri_io[27]}
]
set_property IOSTANDARD LVCMOS33 [get_ports
{GPIO_tri_io*}]

set_property PACKAGE_PIN N12 [get_ports {CSI_C_N}]
set_property PACKAGE_PIN N11 [get_ports {CSI_C_P}]
set_property PACKAGE_PIN R8  [get_ports {CSI_D_N[0]}]
set_property PACKAGE_PIN R7  [get_ports {CSI_D_P[0]}]
set_property PACKAGE_PIN R13 [get_ports {CSI_D_N[1]}]
set_property PACKAGE_PIN R12 [get_ports {CSI_D_P[1]}]
set_property IOSTANDARD LVDS_25 [get_ports {CSI_*}]
set_property PACKAGE_PIN N8  [get_ports {CLP_D_N[0]}]
set_property PACKAGE_PIN N7  [get_ports {CLP_D_P[0]}]
set_property PACKAGE_PIN P14 [get_ports {CLP_D_N[1]}]
set_property PACKAGE_PIN P13 [get_ports {CLP_D_P[1]}]
#set_property PACKAGE_PIN R11 [get_ports {CLP_C_N}]
#set_property PACKAGE_PIN P11 [get_ports {CLP_C_P}]
set_property IOSTANDARD HSUL_12 [get_ports {CLP_*}]
set_property PULLDOWN true [get_ports {CLP_*}]
set_property INTERNAL_VREF 0.6 [get_iobanks 34]
create_clock -period 6.250 -name csi_clk -add [get_ports
CSI_C_P]
```

8 Software Design - Vitis

For SDK project creation, follow instructions from:

Vitis[13]

8.1 Application

SDK 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 VTC and VDMA cores for camera access

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_te0727

Hello TE0727 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. Vitis is used to generate Boot.bin.

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

9 Software Design - PetaLinux

For PetaLinux installation and project creation, follow instructions from:

- PetaLinux KICKstart[14]

9.1 Config

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

Changes:

- No changes

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

9.3 Device Tree

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


/ {
    #address-cells = <1>;
    #size-cells = <1>;

    reserved-memory {
        #address-cells = <1>;
        #size-cells = <1>;
        ranges;
        // HDMI Output frame buffer
        hdmi_fb_reserved_region@1FC00000 {
            compatible = "removed-dma-pool";
            no-map;
            // 512M (M modules)
            reg = <0x1FC00000 0x400000>;
            // 128M (R modules)
            //reg = <0x7C00000 0x400000>;
        };
/*  // Use second frame buffer if you want separate area
for camera image
        camera_fb_reserved_region@1FC00000 {
            compatible = "removed-dma-pool";
```

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

```
            no-map;
            // 512M (M modules)
            reg = <0x1FC00000 0x400000>;
            // 128M (R modules)
            //reg = <0x7800000 0x400000>;
        };
*/
    };

    hdmi_fb: framebuffer@0x1FC00000
{           // HDMI out
        compatible = "simple-framebuffer";
        // 512M (M modules)
        reg = <0x1FC00000 (1280 * 720 * 4)>;    // 720p
        // 128M (R modules)
        //reg = <0x7C00000 (1280 * 720 * 4)>;   // 720p
        width =
<1280>;                         // 720p
        height =
<720>;                         // 720p
        stride = <(1280 *
4)>;                  // 720p
        format = "a8b8g8r8";
        status = "okay";
    };
/*  // In "go through" mode only one framebuffer is used
    camera_fb: framebuffer@0x1FC00000 {         // CAMERA
in
        compatible = "simple-framebuffer";
        // 512M (M modules)
        reg = <0x1FC00000 (1280 * 720 * 4)>;    // 720p
        // 128M (R modules)
        //reg = <0x7800000 (1280 * 720 * 4)>;   // 720p
        width = <1280>;                         // 720p
        height = <720>;                         // 720p
        stride = <(1280 * 4)>;                  // 720p
        format = "a8b8g8r8";
    };
*/
    vcc_3V3: fixedregulator@0 {
        compatible = "regulator-fixed";
        regulator-name = "vccaux-supply";
        regulator-min-microvolt = <3300000>;
        regulator-max-microvolt = <3300000>;
        regulator-always-on;
    };
};

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

```
        #size-cells = <1>;
        spi-max-frequency = <50000000>;
        partition@0x00000000 {
            label = "boot";
            reg = <0x00000000 0x00500000>;
        };
        partition@0x00500000 {
            label = "bootenv";
            reg = <0x00500000 0x00020000>;
        };
        partition@0x00520000 {
            label = "kernel";
            reg = <0x00520000 0x00a80000>;
        };
        partition@0x00fa0000 {
            label = "spare";
            reg = <0x00fa0000 0x00000000>;
        };
    };
};

/*
* We need to disable Linux VDMA driver as VDMA
* already configured in FSBL
*/


&video_out_axi_vdma_0 {
    // Solution 1: Disable satandard VDMA driver (VDMA
configuration should be done in the FSBL)
    status = "disabled";
    // Solution 2: Configure VDMA using the custom driver
(VDMA configuration in FSBL should be disabled)
    //compatible = "trenz,vdmafb";
    //width = <1280>;
    //height = <720>;
    //stride = <(1280 * 4)>;
    //format = "a8b8g8r8";
};

&video_in_axi_vdma_0 {
    // Solution 1: Disable satandard VDMA driver (VDMA
configuration should be done in the FSBL)
    status = "disabled";
};

&gpio0 {
    interrupt-controller;
    #interrupt-cells = <2>;
};


/* I2C1 */
&i2c1 {
    #address-cells = <1>;
    #size-cells = <0>;
```

```
    i2cmux: i2cmux@70  {
        compatible = "nxp,pca9540";
        #address-cells = <1>;
        #size-cells = <0>;
        reg = <0x70>;

        ID_I2C@0 {
            #address-cells = <1>;
            #size-cells = <0>;
            reg = <0>;
        };
        CSI_I2C@1 {
            #address-cells = <1>;
            #size-cells = <0>;
            reg = <1>;
        };
    };

};
/* USB */



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

&usb0 {
    usb-phy = <&usb_phy0>;
} ;
```

9.4 Kernel

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

Changes:

- CONFIG_MII=y
- CONFIG_XILINX_GMII2RGMII=y
- CONFIG_USB_USBNET=y
- CONFIG_USB_NET_AX8817X=y
- CONFIG_USB_NET_AX88179_178A=y
- CONFIG_USB_NET_CDCETHER=y
- # CONFIG_USB_NET_CDC_EEM is not set
- CONFIG_USB_NET_CDC_NCM=y
- # CONFIG_USB_NET_HUAWEI_CDC_NCM is not set
- # CONFIG_USB_NET_CDC_MBIM is not set

- # CONFIG_USB_NET_DM9601 is not set
- # CONFIG_USB_NET_SR9700 is not set
- # CONFIG_USB_NET_SR9800 is not set
- # CONFIG_USB_NET_SMSC75XX is not set
- CONFIG_USB_NET_SMSC95XX=y
- # CONFIG_USB_NET_GL620A is not set
- CONFIG_USB_NET_NET1080=y
- # CONFIG_USB_NET_PLUSB is not set
- # CONFIG_USB_NET_MCS7830 is not set
- # CONFIG_USB_NET_RNDIS_HOST is not set
- CONFIG_USB_NET_CDC_SUBSET_ENABLE=y
- CONFIG_USB_NET_CDC_SUBSET=y
- # CONFIG_USB_ALI_M5632 is not set
- # CONFIG_USB_AN2720 is not set
- CONFIG_USB_BELKIN=y
- CONFIG_USB_ARMLINUX=y
- # CONFIG_USB_EPSON2888 is not set
- # CONFIG_USB_KC2190 is not set
- CONFIG_USB_NET_ZAURUS=y
- # CONFIG_USB_NET_CX82310_ETH is not set
- # CONFIG_USB_NET_KALMIA is not set
- # CONFIG_USB_NET_QMI_WWAN is not set
- # CONFIG_USB_NET_INT51X1 is not set
- # CONFIG_USB_SIERRA_NET is not set
- # CONFIG_USB_VL600 is not set
- # CONFIG_USB_NET_CH9200 is not set
- CONFIG_FB_SIMPLE=y
- # CONFIG_FRAMEBUFFER_CONSOLE is not set
- CONFIG_SND_SIMPLE_CARD_UTILS=y
- CONFIG_SND_SIMPLE_CARD=y
- CONFIG_USBIP_CORE=y
- # CONFIG_USBIP_VHCI_HCD is not set
- # CONFIG_USBIP_HOST is not set
- # CONFIG_USBIP_VUDC is not set
- # CONFIG_USBIP_DEBUG is not set

9.5 Rootfs

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

Changes:

- i2c-tools
- alsa-plugins
- alsa-lib-dev
- libasound
- alsa-conf-base
- alsa-conf
- alsa-utils
- alsa-utils-aplay
- busybox-httpd

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 rpicam

Application used to enable and configure Raspbery Pi camera module

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

9.6.3 fbgrab

Application used to take screenshot from camera

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

9.6.4 webfwu

Webserver application accemble for Zynq access. Need busybox-httpd

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

9.7 Kernel Modules

9.7.1 te-audio-codec

Simple module stab to use audio interface.

See: \os\petalinux\project-spec\meta-user\recipes-modules\te-audio-codec\files

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
2020-11-24	v.5[15]	John Hartfiel[16]	• 19.2 release
--	all	John Hartfiel[17] , Manuela Strücker[18]	--

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.

15 https://wiki.trenz-electronic.de/pages/viewpage.action?pageId=134775531
16 https://wiki.trenz-electronic.de/display/~j.hartfiel
17 https://wiki.trenz-electronic.de/display/~j.hartfiel
18 https://wiki.trenz-electronic.de/display/~m.struecker

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[19]. 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[20] 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

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

of 0,1 % weight by weight) will be classified as SVHC by the European Chemicals Agency (ECHA)[21].

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

21 http://www.echa.europa.eu/