TEI0022 Test Board

Table of contents

Overview

Refer to https://ohwr.org/tei0022-info for the current online version of this manual and other available documentation.

Key Features

- Quartus Prime Lite 21.1
- Intel SoC FPGA EDS Standard Edition 20.1
- Yocto
- SD
- UART
- ETH
- USB
- I2C
- QSPI
- HDMI
- MAC from EEPROM
- DDR3 memory
- User LED

Revision History

Date	Quartus	Project Built	Authors	Description
2022-06-15	21.1 Lite	TEI0022-test_board_noprebuilt-quartus_21.1.0-20220615163042.zip TEI0022-test_board-quartus_21.1.0-20220615163226.zip	Thomas Dück	• update to Quartus Prime Lite 21.1 • bugfixes
2022-04-26	20.1.1 Lite	TEI0022-test_board_noprebuilt-quartus_20.1.1-20220426153812.zip TEI0022-test_board-quartus_20.1.1-20220426153922.zip	Thomas Dück	• add lock_avalon_base_address command to qsys source files
2022-02-03	20.1.1 Lite	TEI0022-test_board_noprebuilt-quartus_20.1.1-20220203152427.zip TEI0022-test_board-quartus_20.1.1-20220203153430.zip	Thomas Dück	• 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
Quartus Prime Lite	21.1	needed
Intel SoC FPGA EDS Standard Edition	20.1	needed
Yocto	dunfell	optional (more information: Yocto KICKstart#Used source files)

Software

Hardware

Complete List is available on *<project folder>/board_files/*_board_files.csv*

Design supports following modules:

Module Model	Board Part Short Name	Yocto Machine Name	PCB Revision	DDR Support	QSPI Flash	EMMC	Others	Notes
TEI0022-03*	A5_C8_2GB	tei0022-a5-c8-2gb	REV03	2GB	32MB	--	--	--

*used as reference

Hardware Modules

Design supports following carriers:

Carrier Model	Notes
---	

*used as reference

Hardware Carrier

Additional HW Requirements:

Additional Hardware	Notes
USB cable for JTAG/UART	Check Carrier Board and Programmer for correct type
Monitor	tested with DELL U2412M
Keyboard	--
Mouse	--
HDMI cable	--

RJ45 ethernet cable	--

*used as reference

Additional Hardware

Content

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

Design Sources

Type	Location	Notes
Quartus	<project folder>/source_files /quartus <project folder>/source_files /<Board Part Short Name> /quartus	Quartus project will be generated by TE Scripts optional, source files for specific assembly variants
Yocto	<project folder>/source_files/os /yocto	Yocto BSP layer template for linux

Design sources

Prebuilt

File	File-Extension	Description
SOPC Information File	*.sopcinfo	File with description of the .qsys file to create software for the target hardware
SRAM Object File	*.sof	Ram configuration file
Raw binary file	*.rbf	FPGA configuration file
Diverse Reports	---	Report files in different formats
Device Tree	*.dtb	Device tree blob
SFP-File	*.sfp	Boot image with SPL (Secondary Program Loader)
BIN-File	*.bin	Image with linux kernel and ram disk
CONF-File	*.conf	Boot configuration file (extlinux. conf)

Prebuilt files (only on ZIP with prebult content)

Download

Reference Design is only usable with the specified Quartus version. Do never use different versions of Quartus software for the same project.

Reference Design is available on:

- TEI0022 "Test Board" Reference Design

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 Quartus Design Flow.

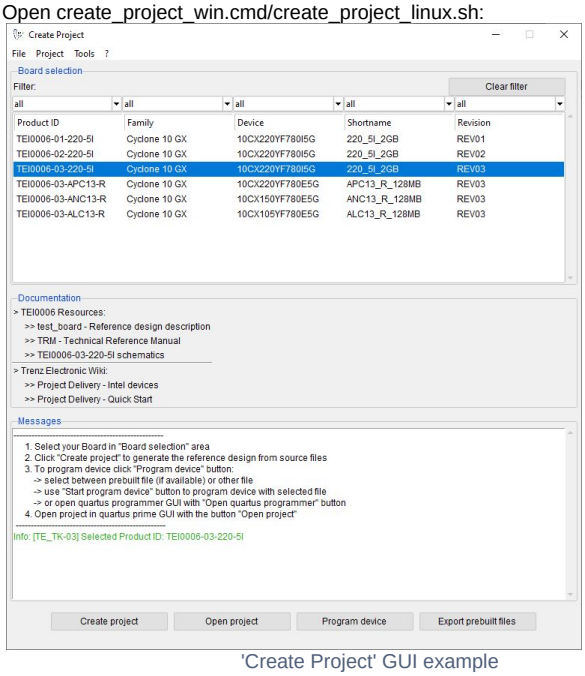
See also:

- Project Delivery - Intel devices

The Trenz Electronic FPGA Reference Designs are TCL-script based projects. To create a project, open a project or program a device execute "create_project_win.cmd" on Windows OS and "create_project_linux.sh" on Linux OS.

TE Scripts are only needed to generate the quartus project, all other additional steps are optional and can also executed by Intel Quartus/SDK GUI. For currently Scripts limitations on Win OS and Linux OS see: Project Delivery - Intel devices Currently limitations of functionality

1. Open create_project_win.cmd/create_project_linux.sh:



'Create Project' GUI example

2. Select Board in "Board selection"
3. Click on "Create project" button to create project
    a. (optional for manual changes) Select correct quartus installation path in "*<project folder>/settings/design_basic_settings.tcl*"
4. Create and configure your Yocto Linux project, see Yocto KICKstart
    a. Copy the generated meta-<module> folder from *<project name>/os/yocto/meta-<module>* to the *path/to/yocto/poky/* directory
    b. Follow the steps from Yocto KICKstart#Create a project for an Intel FPGA device without running the 'bitbake' command
    c. Add the generated bsp layer meta-<machine> to *path/to/yocto/poky/build/conf/bblayers.conf* with:

```
bitbake-layers add-layer ../meta-tei0022
```

> Note: The generated meta-tei0022 layer depends on the meta-altera layer (for more information see: Yocto KICKstart#Used source files), so you need to add both bsp layers to bblayers.conf

d. Redefine the variable MACHINE with 'tei0022-<Board-Part-Short-Name>' in *path/to/yocto/poky/build/conf/local.conf*. The correct MACHINE name can be found in the #Hardware table.
Also define the variables INITRAMFS_IMAGE_BUNDLE and INITRAMFS_IMAGE to create a ram disk image.

```
sed -i '/^MACHINE/s/MACHINE/#MACHINE/g' conf/local.conf
echo -e '\nMACHINE = "tei0022-<Board-Part-Short-Name>"' >>
conf/local.conf
echo -e '\nINITRAMFS_IMAGE_BUNDLE = "1"' >> conf/local.conf
echo -e 'INITRAMFS_IMAGE = "te-initramfs"' >> conf/local.conf
```

e. Build the image with following command (the image recipes are located in *meta-tei0022/recipes-core/images/*):

```
bitbake te-image-minimal
```

5. [optional] Create a debian or ubuntu rootfs with/without desktop environment for this board. For more information and instructions see: Create debian/ubuntu rootfs - Intel devices

Launch

Programming

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

Get prebuilt boot binaries

> Reference Design is also available with prebuilt files. It's recommended to use TE prebuilt files for first launch.

1. Run create_project_win.cmd/create_project_linux.sh
2. Select Module in 'Board selection'
3. Click on 'Export prebuilt files' button
    a. Folder <*project folder*>/_binaries_<*Article Name*> with subfolder *boot_linux* will be generated and opened

QSPI-Boot mode

Option for **u-boot-with-spl.sfp** on QSPI flash and **zimage-initramfs-<Yocto Machine Name>.bin, <Yocto Machine Name>.dtb, soc_system.rbf** and **extlinux/extlinux.conf** on SD card

Use files from "<project folder>_binaries_<Article Name>\boot_linux" from generated binary folder,see: # Get prebuilt boot binaries

1. Set JTAGSEL0 and JTAGSEL1 to Cyclone V HPS access
    - see TEI0022 TRM#Micro USB Connector (JTAG) for correct settings
2. Connect JTAG (USB connector J13) and power on carrier with module
3. Open *path/to/intelFPGA_lite/21.1/embedded/Embedded_Command_Shell.bat* (Win OS)/*path/to /intelFPGA_lite/21.1/embedded/embedded_command_shell.sh* (Linux OS) from Intel SoC FPGA EDS
4. Run following commands:

```
quartus_hps -c 1 -o pv -a 0x0 path/to/_binaries_<Article Name>
/boot_linux/u-boot-with-spl.sfp
```

5. Copy **zimage-initramfs-<Yocto Machine Name>.bin, <Yocto Machine Name>.dtb, soc_system.rbf** and the **extlinux** folder from *path/to/_binaries_<Article Name>/boot_linux/* to SD card
6. Set Boot Mode to QSPI-Boot and insert the SD card in the SD-Slot
    - see TEI0022 TRM#Configuration Signals for correct settings

SD-Boot mode

1. Prepare SD card as follows for SD-Boot
    a. Run following command to get the device name of the SD card (e.g. /dev/sdx):

```
lsblk
```

    b. Insert SD card in the SD card reader, unmount and erase it

```
sudo umount /dev/sdx
sudo sfdisk --delete /dev/sdx
```

    c. Create required partitions on the SD card (partition 1: 50MB, FAT32 / partition 2: 2MB, a2)

```
echo -e ',50M,c\n,2M,a2' | sudo sfdisk /dev/sdb --force
sudo mkfs.vfat -F 32 -n boot /dev/sdb1
```

    d. Copy the u-boot file to partition 2 of the SD card

```
sudo dd if=path/to/_binaries_<Article Name>/boot_linux/u-boot-
with-spl.sfp of=/dev/sdb2 bs=1M seek=0
sync
```

    e. Copy **zimage-initramfs-<Yocto Machine Name>.bin, <Yocto Machine Name>.dtb, soc_system.rbf** and the **extlinux** folder from *path/to/_binaries_<Article Name> /boot_linux/* via file manager to the partition 1 (named 'boot') on SD card.
2. Set Boot Mode to SD-Boot.
    - see TEI0022 TRM#Configuration Signals for correct settings
3. Insert SD-Card in the SD-Slot.

JTAG

Not used on this example.

Usage

1. Prepare HW like described on section #Programming
2. Connect UART USB (USB connector J5)
3. Connect your board to the network
4. Power on PCB

UART

1. Open Serial Console (e.g. PuTTY)
    a. select COM Port

    > Win OS: see device manager
    >
    > Linux OS: see dmesg | grep tty (UART is *USB1)

    b. Speed: 115200
2. Press reset button
3. Linux Console:
    a. Login data:

    > Note: Wait until Linux boot finished

    ```
    Username: root
    Password: root
    ```

    b. You can use Linux shell now.

    ```
    #check I2C 1 Bus
    i2cdetect -y -r 1
    #ETH0 check
    udhcpc
    #USB check
    lsusb
    #toggle leds (state= 0 or 1 / led_name= hps_led1, hps_led2,
    fpga_led1, fpga_led2)
    echo <state> > /sys/class/leds/<led_name>/brightness
    #check temperature (Unit: millidegree Celsius)
    cat /sys/class/hwmon/hwmon0/device/temp1_input
    ```

Monitor

1. Connect the Monitor to HDMI
2. Connect the Mouse+Keyboard to USB
3. Press reset button
4. The linux console is displayed:
    a. Login data:

    > Note: Wait until Linux boot finished

```
Username: root
Password: root
```

b. You can use Linux shell now.

```
#check I2C 1 Bus
i2cdetect -y -r 1
#ETH0 check
udhcpc
#USB check
lsusb
#toggle leds (state= 0 or 1 / led_name= hps_led1, hps_led2,
fpga_led1, fpga_led2)
echo <state> > /sys/class/leds/<led_name>/brightness
#check temperature (Unit: millidegree Celsius)
cat /sys/class/hwmon/hwmon0/device/temp1_input
```

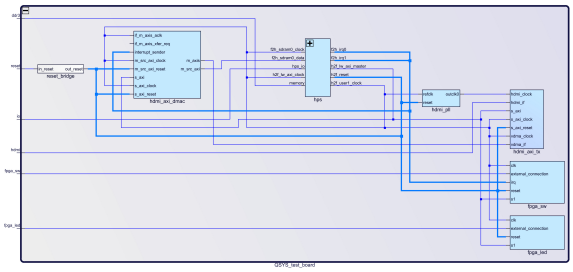
5. [optional] Ubuntu/Debian desktop will be started automatically (for more information see #Rootfs)

System Design - Quartus

Block Design

The block designs may differ depending on the assembly variant.

Block Design - Project



Block Design - Platform Desginer

HPS Interfaces

Activated interfaces:

Type	Note
DDR	--
EMAC1	--
QSPI	--
SDMMC	--
USB1	--
UART0	--
I2C0	--
I2C1	--
GPIO35	connected to ETH PHY_INT pin
GPIO42	connected to USB_RST pin
GPIO43	connected to ETH_RST pin
GPIO48	connected to CPU_GPIO_0 pin
GPIO53	connected to LED_HPS_1 pin
GPIO54	connected to LED_HPS_2 pin
GPIO55	connected to CPU_GPIO_3 pin
GPIO56	connected to CPU_GPIO_2 pin
GPIO57	connected to USER_BTN_HPS pin
GPIO58	connected to CPU_GPIO_1 pin
GPIO61	connected to CPU_GPIO_4 pin

Software Design - Yocto

For Yocto installation and project creation, follow instructions from:

- Yocto KICKstart
- Create a custom BSP layer for Intel SoC or FPGA
- Create debian/ubuntu rootfs - Intel devices

U-Boot

Start with Create a custom BSP layer for Intel SoC or FPGA#Configure u-boot

File location: *meta-tei0022/recipes-bsp/u-boot/*

Changes:

- select tei0022 board
    - # CONFIG_TARGET_SOCFPGA_CYCLONE5_SOCDK is not set
    - CONFIG_TARGET_TEI0022=y
- configure bootcommand (load soc_system.rbf file into the FPGA
    - CONFIG_BOOTCOMMAND="load mmc 0:1 $loadaddr soc_system.rbf; fpga load 0 $loadaddr $filesize; bridge enable; run distro_bootcmd"
- enable misc_init_r function (need to call TE_read_eeprom_mac function)
    - CONFIG_MISC_INIT_R=y
    - CONFIG_MISC=y
- MAC from eeprom together with uboot:
    - CONFIG_I2C_EEPROM=y
    - CONFIG_SYS_I2C_EEPROM_ADDR=0x50
    - CONFIG_SYS_I2C_EEPROM_BUS=1
    - 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
- configure eth
    - CONFIG_PHYLIB=y
    - CONFIG_NETDEVICES=y
    - CONFIG_RGMII=y
    - # CONFIG_MII is not set
- select device tree
    - CONFIG_DEFAULT_DEVICE_TREE="tei0022_<Board Part Short Name>"
    - CONFIG_DEFAULT_FDT_FILE="tei0022_<Board Part Short Name>.dtb"

Device Tree

U-boot Device Tree

Excerpts from test_board/os/yocto/meta-tei0022/recipes-bsp/u-boot/files

/tei0022_<Board_Part_Short_Name>/dts/tei0022_<Board_Part_Short_Name>.dts

```
#include "socfpga_cyclone5.
dtsi"

/ {
        model = "Trenz Electronic - TEI0022";
        compatible = "altr,socfpga-cyclone5", "altr,socfpga";

        chosen {
```

```
                bootargs = "earlyprintk";
                stdout-path = "serial0:115200n8";
        };

        memory {
                name = "memory";
                device_type = "memory";
                reg = <0x0 0x40000000>;
        };

        aliases {
                ethernet0 = &gmac1;
        };
};

&gpio0 {
        status = "okay";
};

&gpio1 {
        status = "okay";
};

&gpio2 {
        status = "okay";
};

&gmac1 {
        #address-cells = <1>;
        #size-cells = <0>;

        status = "okay";
        phy-mode = "rgmii";

        ethernet-phy@1 {
                reg = <1>;
                adi,rx-internal-delay-ps = <2000>;
                adi,tx-internal-delay-ps = <2000>;
        };
};

&i2c1 {
        status = "okay";
        clock-frequency = <100000>;
        eeprom: eeprom@50 {
                compatible = "microchip,24aa02e48","atmel,24c02";
                reg = <0x50>;
        };
};

&uart0 {
        clock-frequency = <100000000>;
};

&mmc0 {
        status = "okay";
};

&qspi {
        status = "okay";
```

```
        flash: mt25ql256a@0 {
                #address-cells = <1>;
                #size-cells = <1>;
                compatible = "jedec,spi-nor";
                reg = <0>;
                spi-max-frequency = <100000000>;
                m25p,fast-read;

                cdns,page-size = <256>;
                cdns,block-size = <16>;
                cdns,read-delay = <4>;
                cdns,tshsl-ns = <50>;
                cdns,tsd2d-ns = <50>;
                cdns,tchsh-ns = <4>;
                cdns,tslch-ns = <4>;

                partition@qspi-boot {
                        label = "Flash 0 Raw Data";
                        reg = <0x0 0x400000>;
                };
        };
};
```

Excerpts from test_board/os/yocto/meta-tei0022/recipes-bsp/u-boot/files/tei0022_<Board_Part_Short_Name>/dts/tei0022_<Board_Part_Short_Name>-u-boot.dtsi

```
#include "socfpga-common-u-boot.dtsi"

&watchdog0 {
        status = "disabled";
};

&mmc {
        u-boot,dm-pre-reloc;
};

&qspi {
        u-boot,dm-pre-reloc;
};

&flash {
        compatible = "jedec,spi-nor";
        u-boot,dm-pre-reloc;

        partition@qspi-boot {
                label = "Flash 0 Raw Data";
                reg = <0x0 0x400000>;
        };
};

&uart0 {
        clock-frequency = <100000000>;
        u-boot,dm-pre-reloc;
};

&porta {
        bank-name = "porta";
};

&portb {
        bank-name = "portb";
};

&portc {
        bank-name = "portc";
};
```

Kernel Device Tree

Excerpts from test_board/os/yocto/meta-tei0022/recipes-kernel/linux/files/dts/tei0022_<Board_Part_Short_Name>.dts

```
#include "socfpga_cyclone5.dtsi"

/ {
        model = "Trenz Electronic - TEI0022";
```

```
compatible = "altr,socfpga-cyclone5", "altr,socfpga";

chosen {
        bootargs = "earlyprintk";
        stdout-path = "serial0:115200n8";
};

memory {
        name = "memory";
        device_type = "memory";
        reg = <0x0 0x40000000>;
};

aliases {
        ethernet0 = &gmac1;
};

regulator_1_8v: 1-8-v-regulator {
        compatible = "regulator-fixed";
        regulator-name = "1.8V";
        regulator-min-microvolt = <1800000>;
        regulator-max-microvolt = <1800000>;
        regulator-always-on;
};

regulator_3_3v: 3-3-v-regulator {
        compatible = "regulator-fixed";
        regulator-name = "3.3V";
        regulator-min-microvolt = <3300000>;
        regulator-max-microvolt = <3300000>;
        regulator-always-on;
};

hdmi_pll: hdmi_pll {
        compatible = "altr,altera_iopll-18.1";
        #clock-cells = <1>;
                hdmi_pll_outclk0: hdmi_pll_outclk0 {
                compatible = "fixed-clock";
                #clock-cells = <0>;
                clock-frequency = <148500000>;
                clock-output-names = "hdmi_pll-outclk0";
        };
};

sys_hps_bridges: bridge@ff200000 {
        compatible = "simple-bus";
        reg = <0xff200000 0x00200000>;
        reg-names = "axi_h2f_lw";
        #address-cells = <2>;
        #size-cells = <1>;
        ranges = <0x00000001 0x00001000 0xff201000 0x00000010>,
                <0x00000001 0x00001010 0xff201010 0x00000010>,
                <0x00000001 0x00001020 0xff201020 0x00000008>,
                <0x00000001 0x00001030 0xff201030 0x00000008>,
                <0x00000001 0x00010000 0xff210000 0x00000800>,
                <0x00000001 0x00020000 0xff220000 0x00010000>;

        fpga_sw: fpga-sw@100001000 {
                compatible = "altr,pio-1.0";
                reg = <0x00000001 0x00001000 0x00000010>;
                interrupts = <0 41 1>;
```

```
        altr,gpio-bank-width = <2>;
        #gpio-cells = <2>;
        gpio-controller;
        interrupt-cells = <1>;
        interrupt-controller;
        altr,interrupt-type = <2>;
};

fpga_led: fpga-led@100001010 {
        compatible = "altr,pio-1.0";
        reg = <0x00000001 0x00001010 0x00000010>;
        altr,gpio-bank-width = <2>;
        #gpio-cells = <2>;
        gpio-controller;
};

leds {
        compatible = "gpio-leds";

        fpgaled1 {
                label = "fpga_led1";
                gpios = <&fpga_led 0 0>;
        };

        fpgaled2 {
                label = "fpga_led2";
                gpios = <&fpga_led 1 0>;
        };

        hpsled1 {
                label = "hps_led1";
                gpios = <&portb 24 0>;  /* GPIO 53 */
        };

        hpsled2 {
                label = "hps_led2";
                gpios = <&portb 25 0>;  /* GPIO 54 */
        };
};

hdmi_axi_dmac: axi-dmac@100010000 {
        compatible = "adi,axi-dmac-1.00.a";
        reg = <0x00000001 0x00010000 0x00000800>;
        #dma-cells = <1>;
        interrupt-parent = <&intc>;
        interrupts = <0 42 4>;
        clocks = <&h2f_usr1_clk>;
        status = "okay";

        adi,channels {
                #size-cells = <0>;
                #address-cells = <1>;

                dma-channel@0 {
                        reg = <0>;
                        adi,source-bus-width = <64>;
                        adi,source-bus-type = <0>;
                        adi,destination-bus-width = <64>;
                        adi,destination-bus-type = <1>;
                };
        };
```

```
                };


                hdmi_axi_tx: axi-hdmi-tx@100020000 {
                        compatible = "adi,axi-hdmi-tx-1.00.a";
                        reg = <0x00000001 0x00020000 0x10000>;
                        dmas = <&hdmi_axi_dmac 0>;
                        dma-names = "video";
                        clocks = <&hdmi_pll_outclk0 0>;
                        status = "okay";

                        port {
                                axi_hdmi_out: endpoint {
                                        remote-endpoint = <&adv7511_in>;
                                };
                        };
                };
        };
};

&mmc {
        status = "okay";
};

&uart0 {
        clock-frequency = <100000000>;
};

&usb1 {
        status = "okay";
        dr_mode = "host";
};

&i2c0 {
        status = "okay";
        speed-mode = <0>;
};

&i2c1 {
        status = "okay";
        speed-mode = <0>;

        adv7511: adv7511@39 {
                compatible = "adi,adv7511";
                reg = <0x39>, <0x3f>;
                reg-names = "primary", "edid";
                status = "okay";

                adi,input-depth = <8>;
                adi,input-colorspace = "yuv422";
                adi,input-clock = "1x";
                adi,input-style = <1>;
                adi,input-justification = "right";
                adi,clock-delay = <(0)>;

                avdd-supply = <&regulator_1_8v>;
                dvdd-supply = <&regulator_1_8v>;
                pvdd-supply = <&regulator_1_8v>;
                dvdd-3v-supply = <&regulator_3_3v>;
                bgvdd-supply = <&regulator_1_8v>;
```

```
                ports {
                        #address-cells = <1>;
                        #size-cells = <0>;

                        port@0 {
                                reg = <0>;
                                adv7511_in: endpoint {
                                        remote-endpoint = <&axi_hdmi_out>;
                                };
                        };

                        port@1 {
                                reg = <1>;
                        };

                };
        };

        adt7410: adt7410@4a {
                compatible = "adt7410";
                reg = <0x4a>;
        };

        eeprom: eeprom@50 {
                compatible = "microchip,24aa02e48","atmel,24c02";
                reg = <0x50>;
        };
};

&gmac1 {

        #address-cells = <1>;
        #size-cells = <0>;

        status = "okay";
        phy-mode = "rgmii-id";

        ethernet-phy@1 {
                reg = <1>;
                adi,rx-internal-delay-ps = <2000>;
                adi,tx-internal-delay-ps = <2000>;
        };
};

&gpio0 {
        status = "okay";
};

&gpio1 {
        status = "okay";
};

&gpio2 {
        status = "okay";
};
```

Kernel

Start with Create a custom BSP layer for Intel SoC or FPGA#Configure linux kernel

File location: *meta-tei0022/recipes-kernel/linux/*

Changes:

- for hdmi output
  - CONFIG_AXI_DMAC=y
  - CONFIG_CMA=y
  - CONFIG_DMA_CMA=y
  - CONFIG_CMA_SIZE_MBYTES=128
  - CONFIG_DRM=y
  - CONFIG_DRM_BRIDGE=y
  - CONFIG_DRM_I2C_ADV7511=y
  - CONFIG_DRM_ADI_AXI_HDMI=y
- set TE boot logo
  - CONFIG_LOGO=y
  - CONFIG_LOGO_TRENZELECTRONICS_CLUT224=y
  - # CONFIG_LOGO_LINUX_MONO is not set
  - # CONFIG_LOGO_LINUX_VGA16 is not set
  - # CONFIG_LOGO_LINUX_CLUT224 is not set
- config ethernet phy
  - CONFIG_PHYLIB=y
  - CONFIG_ADIN_PHY=y
- set adt7410 tempprature sensor driver
  - CONFIG_SENSORS_ADT7X10=y
  - CONFIG_SENSORS_ADT7410=y
- set debug settings
  - CONFIG_DEBUG_LL=y
  - CONFIG_DEBUG_SOCFPGA_UART0=y
  - CONFIG_EARLY_PRINTK=y

Images

Image recipe for minimal console image.

File location: *meta-tei0022/recipes-core/images/*

Image recipes:

- te-image-minimal.bb: create minimal linux image
- te-initramfs.bb: required for building an image with initial RAM Filesystem

Added packages/recipes:

- tei0022-rbf
- ethtool
- i2c-tools
- net-tools
- usbutils
- mtd-utils

Rootfs

Used filesystem: Initial RAM Filesystem (initramfs)

It's Optionally possible to create a debian or ubuntu rootfs with/without desktop environment for this board. For more information and instructions see: Create debian/ubuntu rootfs - Intel devices

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.proxy279.$Proxy4022#hasContentLevelPermission. Cannot resolve which method to invoke for [null, class java.	**Error rendering macro 'page-info'** Ambiguous method overloading for method jdk.proxy279.$Proxy4022#hasContentLevelPermission. Cannot resolve which method to invoke for [null, class java.	**Error rendering macro 'page-info'** Ambiguous method overloading for method jdk.proxy279.$Proxy4022#hasContentLevelPermission. Cannot resolve which method to invoke for [null, class java.	• update to Quartus Prime Lite 21.1

lang.
String,
class
com.
atlassian
.
confluen
ce.
pages.
Page]
due to
overlapp
ing
prototyp
es
between
:
[interfac
e com.
atlassian
.
confluen
ce.user.
Conflue
nceUser
, class
java.
lang.
String,
class
com.
atlassian
.
confluen
ce.core.
Content
EntityOb
ject]
[interfac

lang.
String,
class
com.
atlassian
.
confluen
ce.
pages.
Page]
due to
overlapp
ing
prototyp
es
between
:
[interfac
e com.
atlassian
.
confluen
ce.user.
Conflue
nceUser
, class
java.
lang.
String,
class
com.
atlassian
.
confluen
ce.core.
Content
EntityOb
ject]
[interfac

lang.
String,
class
com.
atlassian
.
confluen
ce.
pages.
Page]
due to
overlapp
ing
prototyp
es
between
:
[interfac
e com.
atlassian
.
confluen
ce.user.
Conflue
nceUser
, class
java.
lang.
String,
class
com.
atlassian
.
confluen
ce.core.
Content
EntityOb
ject]
[interfac

e com. atlassian .user. User, class java. lang. String, class com. atlassian . confluen ce.core. Content EntityOb ject]	e com. atlassian .user. User, class java. lang. String, class com. atlassian . confluen ce.core. Content EntityOb ject]	e com. atlassian .user. User, class java. lang. String, class com. atlassian . confluen ce.core. Content EntityOb ject]	
2022-04-28	v.7	Thomas Dück	• add missing commands to qsys source files
2022-02-08	v.6	Thomas Dück	• intial release 20.1
--	all	**Error renderi ng macro 'page-info'** Ambiguo us method overload ing for method jdk.	--

proxy279.$Proxy4022#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.

		Conflue nceUser , class java. lang. String, class com. atlassian . confluen ce.core. Content EntityOb ject] [interfac e com. atlassian .user. User, class java. lang. String, class com. atlassian . confluen ce.core. Content EntityOb ject]	

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

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.

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.proxy279.$Proxy4022#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]