# Table of Contents

Overview

5

Key Features

5

Revision History

5

Release Notes and Know Issues

5

Requirements

6

Software

6

Hardware

6

Content

7

Design Sources

7

Additional Sources

7

Prebuilt

7

Download

7

Design Flow

8

Launch

10

Programming

10

QSPI

10

SD

10

JTAG

10

Usage

10

Linux

11

Vivado Hardware Manager

11

System Design - Vivado

12

Block Design

12

PS Interfaces

12

Constrains

13

Basic module constrains

13

Design specific constrain

13

Software Design - SDK/HSI

15

Application

15

FSBL

15

zynqmp_fsbl_flash

15

PMU

15

Hello TE0807

15

U-Boot

15

Software Design - PetaLinux

16

Config

16

U-Boot

16

Device Tree

16

Kernel

20

Rootfs

20

Applications

20

startup

20
adau1761init ................................................................................................. 20
Additional Software ......................................................................................... 21
SI5345 .............................................................................................................. 21
Appx. A: Change History and Legal Notices ....................................................... 22
Document Change History .................................................................................. 22
Legal Notices ...................................................................................................... 22
Data privacy ......................................................................................................... 22
Document Warranty ............................................................................................ 22
Limitation of Liability .......................................................................................... 23
Copyright Notice .................................................................................................. 23
Technology Licenses ............................................................................................. 23
Environmental Protection ..................................................................................... 23
REACH, RoHS and WEEE ................................................................................... 24
Online version of this manual and other related documents can be found at https://wiki.trenz-electronic.de/display/PD/Trenz+Electronic+Documentation
Overview

Linux with basic periphery of TE0807 Starterkit (TEBF0808 Carrier).

Key Features

- TEBF0808
- Linux
- USB
- ETH
- PCIe
- SATA
- SD
- I2C
- DP
- RGPIO
- user LED access
- Modified FSBL for Si5345 programming
- Special FSBL for QSPI Programming

Revision History

<table>
<thead>
<tr>
<th>Date</th>
<th>Vivado</th>
<th>Project Built</th>
<th>Authors</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2018-05-24</td>
<td>2017.4</td>
<td>TE0807-StarterKit_noprebuild-vivado_2017.4-build_10_20180524150124.zip</td>
<td>John Hartfiel</td>
<td>solved Linux Flash issue</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TE0807-StarterKit-vivado_2017.4-build_10_20180524150106.zip</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2018-02-06</td>
<td>2017.4</td>
<td>TE0807-StarterKit_noprebuild-vivado_2017.4-build_05_20180206082637.zip</td>
<td>John Hartfiel</td>
<td>same CLK for VIO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TE0807-StarterKit-vivado_2017.4-build_05_20180206082621.zip</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2018-02-05</td>
<td>2017.4</td>
<td>TE0807-StarterKit-vivado_2017.4-build_05_20180205101306.zip</td>
<td>John Hartfiel</td>
<td>solved JTAG/Linux issue</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TE0807-StarterKit_noprebuild-vivado_2017.4-build_05_20180205101252.zip</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2018-01-18</td>
<td>2017.4</td>
<td>TE0807-StarterKit-vivado_2017.4-build_05_20180118152922.zip</td>
<td>John Hartfiel</td>
<td>initial release</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TE0807-StarterKit_noprebuild-vivado_2017.4-build_05_20180118152938.zip</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Release Notes and Know Issues

<table>
<thead>
<tr>
<th>Issues</th>
<th>Description</th>
<th>Workaround/Solution</th>
<th>To be fixed version</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flash access on Linux</td>
<td>Device tree is not correct on Linux</td>
<td>add compatibility to &quot;compatible &quot;jedec, spi-nor&quot;**</td>
<td><strong>Solved with 20180524 update</strong></td>
</tr>
</tbody>
</table>
### Issues

<table>
<thead>
<tr>
<th>Description</th>
<th>Workaround/Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>USB UART Terminal is blocked / SDK Debugging is blocked</td>
<td>Do not use HW Manager connection, or if debugging is necessary:</td>
</tr>
<tr>
<td>This happens only with 2017.4 Linux, when JTAG connection is established on Vivado HW Manager.</td>
<td>1. Boot linux with usb terminal</td>
</tr>
<tr>
<td></td>
<td>2. From the terminal: root root mount ifconfig eth0</td>
</tr>
<tr>
<td></td>
<td>3. Open two new SSH terminals via ethernet: root root, run user application ...</td>
</tr>
<tr>
<td></td>
<td>4. Exit and close the usb terminal</td>
</tr>
</tbody>
</table>

**To be fixed**

Solved with 20180205 update

### Requirements

#### Software

<table>
<thead>
<tr>
<th>Software</th>
<th>Version</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vivado</td>
<td>2017.4</td>
<td>needed</td>
</tr>
<tr>
<td>SDK</td>
<td>2017.4</td>
<td>needed</td>
</tr>
<tr>
<td>PetaLinux</td>
<td>2017.4</td>
<td>needed</td>
</tr>
</tbody>
</table>

#### 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:

<table>
<thead>
<tr>
<th>Module Model</th>
<th>Board Part Short Name</th>
<th>PCB Revision Support</th>
<th>DDR</th>
<th>QSPI Flash</th>
<th>Others</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>TE0807-01-ES2</td>
<td>es2_sk</td>
<td>REV01</td>
<td>2GB</td>
<td>64MB</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note: Design contains also Board Part Files for TE0807 only configuration, this board part files are not used for this reference design.

Design supports following carriers:

<table>
<thead>
<tr>
<th>Carrier Model</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>TEBF0808</td>
<td>Used as reference carrier.</td>
</tr>
</tbody>
</table>

Additional HW Requirements:
Content

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

Design Sources

<table>
<thead>
<tr>
<th>Type</th>
<th>Location</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vivado</td>
<td>&lt;design name&gt;/block_design</td>
<td>Vivado Project will be generated by TE Scripts</td>
</tr>
<tr>
<td></td>
<td>&lt;design name&gt;/constraints</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&lt;design name&gt;/ip_lib</td>
<td></td>
</tr>
<tr>
<td>SDK/HSI</td>
<td>&lt;design name&gt;/sw_lib</td>
<td>Additional Software Template for SDK/HSI and apps_list.csv with settings for HSI</td>
</tr>
<tr>
<td>PetaLinux</td>
<td>&lt;design name&gt;/os/petalinux</td>
<td>PetaLinux template with current configuration</td>
</tr>
</tbody>
</table>

Additional Sources

<table>
<thead>
<tr>
<th>Type</th>
<th>Location</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>SI5345</td>
<td>&lt;design name&gt;/misc/Si5345</td>
<td>SI5345 Project with current PLL Configuration</td>
</tr>
</tbody>
</table>

Prebuilt

<table>
<thead>
<tr>
<th>File</th>
<th>File-Extension</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BIF-File</td>
<td>*.bif</td>
<td>File with description to generate Bin-File</td>
</tr>
<tr>
<td>BIN-File</td>
<td>*.bin</td>
<td>Flash Configuration File with Boot-Image (Zynq-FPGAs)</td>
</tr>
<tr>
<td>BIT-File</td>
<td>*.bit</td>
<td>FPGA (PL Part) Configuration File</td>
</tr>
<tr>
<td>DebugProbes-File</td>
<td>*.ltx</td>
<td>Definition File for Vivado/Vivado Labtools Debugging Interface</td>
</tr>
<tr>
<td>Diverse Reports</td>
<td>---</td>
<td>Report files in different formats</td>
</tr>
<tr>
<td>Hardware-Platform-Specification-Files</td>
<td>*.hdf</td>
<td>Exported Vivado Hardware Specification for SDK/HSI and PetaLinux</td>
</tr>
<tr>
<td>LabTools Project-File</td>
<td>*.lpr</td>
<td>Vivado Labtools Project File</td>
</tr>
<tr>
<td>OS-Image</td>
<td>*.ub</td>
<td>Image with Linux Kernel (On Petalinux optional with Devicetree and RAM-Disk)</td>
</tr>
<tr>
<td>Software-Application-File</td>
<td>*.elf</td>
<td>Software Application for Zynq or MicroBlaze Processor Systems</td>
</tr>
</tbody>
</table>

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:

- TE0807 “StarterKit” Reference Design
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:

- Vivado/SDK/SDSoC/XilinxSoftware-BasicUserGuides
- Vivado Projects
- 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
   Important: Use Board Part Files, which ends with *__tebf0808
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 `\prebuilt\hardware\<short dir>`. Use GUI is
      the same, except file export to prebuilt folder
6. Create Linux (bl31.elf, 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
      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 (bl31.elf, uboot.elf and image.ub) to prebuilt folder
   a. "\prebuilt\os\petalinux\default" or "\prebuilt\os\petalinux\<short name>"
      Notes: Scripts select "\prebuilt\os\petalinux\<short name>", if exist, otherwise
      "\prebuilt\os\petalinux\default"
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
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

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 (zynqmp_fsbl_flash) on setup
4. Copy image.ub on SD-Card
5. Insert SD-Card

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.
3. Insert SD-Card in SD-Slot.

JTAG

Not used on this Example.

Usage

1. Prepare HW like described on section Programming
2. Connect UART USB (JTAG XMOD)
3. Select SD Card as Boot Mode (or QSPI - depending on step 1)
   Note: See TRM of the Carrier, which is used.
4. (Optional) Insert PCIe Card (detection depends on Linux driver. Only some basic drivers are installed)
5. (Optional) Connect Sata Disc
7. (Optional) Connect Network Cable
8. Power On PCB
   
   Note: 1. ZynqMP Boot ROM loads PMU Firmware and FSBL from SD into OCM, 2. FSBL loads ATF (bl31.elf) and U-boot from SD/QSPI 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. ETH0 works with udhcpc
   c. USB type "lsusb" or connect USB device
   d. PCIe type "lspci"

### Vivado Hardware Manager

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

- **RGPIO Interface:**
  - Set Bit 31-28 to "1010" to activat RGPIO Interface of Master or Slave CPLD.
    - Description: TEBF0808 Master CPLD#RGPIO, TEBF0808 Slave CPLD#RGPIO
- **LED Control:**
  - XMOD 2(without green dot) and HD LED are accessible.
System Design - Vivado

Block Design

PS Interfaces

Activated interfaces:

<table>
<thead>
<tr>
<th>Type</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>DDR</td>
<td>MIO</td>
</tr>
<tr>
<td>QSPI</td>
<td>MIO</td>
</tr>
<tr>
<td>SD0</td>
<td>MIO</td>
</tr>
<tr>
<td>SD1</td>
<td>MIO</td>
</tr>
<tr>
<td>CAN0</td>
<td>EMIO</td>
</tr>
<tr>
<td>I2C0</td>
<td>MIO</td>
</tr>
<tr>
<td>PJTAG0</td>
<td>MIO</td>
</tr>
<tr>
<td>UART0</td>
<td>MIO</td>
</tr>
<tr>
<td>GPIO0</td>
<td>MIO</td>
</tr>
</tbody>
</table>
### Type

<table>
<thead>
<tr>
<th>Type</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>SWDT0..1</td>
<td></td>
</tr>
<tr>
<td>TTC0..3</td>
<td></td>
</tr>
<tr>
<td>GEM3</td>
<td>MIO</td>
</tr>
<tr>
<td>USB0</td>
<td>MIO/GTP</td>
</tr>
<tr>
<td>PCIe</td>
<td>MIO/GTP</td>
</tr>
<tr>
<td>SATA</td>
<td>GTP</td>
</tr>
</tbody>
</table>

### Constrains

#### Basic module constrains

```
_i_bitgen.xdc

set_property BITSTREAM GENERAL COMPRESS TRUE [current_design]
set_property BITSTREAM CONFIG UNUSEDPIN PULLNONE [current_design]
```

#### Design specific constrain

```
_i_io.xdc

#LED_HD SC0 J3:31
set_property PACKAGE_PIN K11 [get_ports {LED_HD[0]}]
set_property IOSTANDARD LVCMOS18 [get_ports {LED_HD[0]}]

#LED_XMOD SC17 J3:48
set_property PACKAGE_PIN B12 [get_ports {LED_XMOD2[0]}]
set_property IOSTANDARD LVCMOS18 [get_ports {LED_XMOD2[0]}]

#System Controller IP
#J3:32
set_property PACKAGE_PIN J12 [get_ports BASE_sc10_io]
#J3:34
set_property PACKAGE_PIN K13 [get_ports BASE_sc11]
#J3:36
set_property PACKAGE_PIN A13 [get_ports BASE_sc12]
#J3:38
set_property PACKAGE_PIN A14 [get_ports BASE_sc13]
#J3:40
set_property PACKAGE_PIN E12 [get_ports BASE_sc14]
#J3:42
set_property PACKAGE_PIN F12 [get_ports BASE_sc15]
#J3:41
set_property PACKAGE_PIN E14 [get_ports BASE_sc5]
#J3:45
set_property PACKAGE_PIN C12 [get_ports BASE_sc6]
#J3:47
set_property PACKAGE_PIN D12 [get_ports BASE_sc7]
set_property IOSTANDARD LVCMOS18 [get_ports BASE_sc5]
set_property IOSTANDARD LVCMOS18 [get_ports BASE_sc6]
```
set_property IOSTANDARD LVCMOS18 [get_ports BASE_sc7]
set_property IOSTANDARD LVCMOS18 [get_ports BASE_sc10_io]
set_property IOSTANDARD LVCMOS18 [get_ports BASE_sc11]
set_property IOSTANDARD LVCMOS18 [get_ports BASE_sc12]
set_property IOSTANDARD LVCMOS18 [get_ports BASE_sc13]
set_property IOSTANDARD LVCMOS18 [get_ports BASE_sc14]
set_property IOSTANDARD LVCMOS18 [get_ports BASE_sc15]

# PLL
#J4:74
#set_property PACKAGE_PIN AF15 [get_ports {si570_clk_p[0]}]
#set_property IOSTANDARD LVDS [get_ports {si570_clk_p[0]}]
#set_property IOSTANDARD LVDS [get_ports {si570_clk_n[0]}]

# Audio Codec
#LRCLK        J3:49 B47_L9_N
set_property PACKAGE_PIN G14 [get_ports LRCLK ]
#BCLK        J3:51 B47_L9_P
set_property PACKAGE_PIN H14 [get_ports BCLK ]
#DAC_SDATA   J3:53 B47_L7_N
set_property PACKAGE_PIN C13 [get_ports DAC_SDATA ]
#ADC_SDATA   J3:55 B47_L7_P
set_property PACKAGE_PIN D14 [get_ports ADC_SDATA ]
set_property IOSTANDARD LVCMOS18 [get_ports LRCLK ]
set_property IOSTANDARD LVCMOS18 [get_ports BCLK ]
set_property IOSTANDARD LVCMOS18 [get_ports DAC_SDATA ]
set_property IOSTANDARD LVCMOS18 [get_ports ADC_SDATA ]

# CAN
#CAN RX SC19 J3:52 B47_L10_P
#CAN TX SC18 J3:50 B47_L10_N
#CAN S SC16 J3:46 B47_L12_N
set_property PACKAGE_PIN A12 [get_ports CAN_0_S ]
set_property IOSTANDARD LVCMOS18 [get_ports CAN_0_S ]
set_property PACKAGE_PIN C14 [get_ports CAN_0_rx ]
set_property IOSTANDARD LVCMOS18 [get_ports CAN_0_rx ]
set_property PACKAGE_PIN B14 [get_ports CAN_0_tx ]
set_property IOSTANDARD LVCMOS18 [get_ports CAN_0_tx ]
Software Design - SDK/HSI

For SDK project creation, follow instructions from:

SDK Projects

Application

FSBL

TE modified 2017.4 FSBL

Changes:

- Si5345Configuration, PCIe Reset over GPIO see xfsbl_board.c and xfsbl_board.h
- Add Si5345-Registers.h, si5345.c, si5345.h

zynqmp_fsbl_flash

TE modified 2017.4 FSBL

Changes:

- Set FSBL Boot Mode to JTAG
- Disable Memory initialisation

PMU

Xilinx default PMU firmware.

Hello TE0807

Hello TE0807 is a Xilinx Hello World example as endless loop instead of one console output.

U-Boot

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

For PetaLinux installation and project creation, follow instructions from:

- PetaLinux KICKstart

Config

No changes.

U-Boot

- Change platform-top.h

```c
#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 
   DFU_ALT_INFO_RAM

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

/*select sd instead of mmc for autoboot */

#define CONFIG_BOOTCOMMAND   "run uenvboot; mmcinfo && fatload mmc 1 ${netstart} ${kernel_img};bootm ${netstart}"
```

Device Tree

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

/*/ default */
/* SD */
```c
#define bootloader_name "TE0807 StarterKit"
#define bootloader_revision "v.15"

#include "fsl_defines.h"

/* USB */
#define usb_doctor "okay"
#define usb_host_mode "host"

/* ETH PHY */
#define eth_phy_type "ethernet-phy"
#define eth_phy_reg <1>

/* QSPI */
#define qspi_addr_cells <1>
#define qspi_size_cells <0>
#define flash_addr_cells <1>
#define flash_size_cells <1>

/* I2C */
#define i2c_addr_cells <1>
#define i2c_size_cells <0>
#define i2c_addr_reg <0x73>
#define i2c_mux_idle_disconnect
```

```
#define i2c_reg <2>
#define i2c0 [ // PCI
    #address-cells = <1>
    #size-cells = <0>
    reg = <2>
]
#define i2c0 [ // i2c SFP
    #address-cells = <1>
    #size-cells = <0>
    reg = <3>
]
```
i2c@4 { // i2c SFP
    #address-cells = <1>;
    #size-cells = <0>;
    reg = <4>;
};
i2c@5 { // i2c EEPROM
    #address-cells = <1>;
    #size-cells = <0>;
    reg = <5>;
};
i2c@6 { // i2c FMC
    #address-cells = <1>;
    #size-cells = <0>;
    reg = <6>;
    si570_2: clock-generator3@5d {
        #clock-cells = <0>;
        compatible = "silabs,si570";
        reg = <0x5d>;
        temperature-stability = <50>;
        factory-fout = <156250000>;
        clock-frequency = <78800000>;
    };
};
i2c@7 { // i2c USB HUB
    #address-cells = <1>;
    #size-cells = <0>;
    reg = <7>;
};

i2cswitch@77 { // u
    compatible = "nxp,pca9548";
    #address-cells = <1>;
    #size-cells = <0>;
    reg = <0x77>;
    i2c-mux-idle-disconnect;
}
i2c@0 { // i2c PMOD
    #address-cells = <1>;
    #size-cells = <0>;
    reg = <0>;
};
i2c@1 { // i2c Audio Codec
    #address-cells = <1>;
    #size-cells = <0>;
    reg = <1>;
    /*
    adau1761: adau1761@38 {
        compatible = "adi,adau1761";
        reg = <0x38>;
    };
    */
};
i2c@2 { // i2c FireFly A
    #address-cells = <1>;
    #size-cells = <0>;
    reg = <2>;
};
i2c@3 { // i2c FireFly B
#address-cells = <1>
#size-cells = <0>
reg = <3>
"

i2c@4 { // i2c PLL
    #address-cells = <1>
    #size-cells = <0>
    reg = <4>
}

i2c@5 { // i2c SC
    #address-cells = <1>
    #size-cells = <0>
    reg = <5>
}

i2c@6 { // i2c
    #address-cells = <1>
    #size-cells = <0>
    reg = <6>
}

i2c@7 { // i2c
    #address-cells = <1>
    #size-cells = <0>
    reg = <7>
}

/* UNUSED DMA disable */

&lpd_dma_chan1 {
    status = "disabled";
};

&lpd_dma_chan2 {
    status = "disabled";
};

&lpd_dma_chan3 {
    status = "disabled";
};

&lpd_dma_chan4 {
    status = "disabled";
};

&lpd_dma_chan5 {
    status = "disabled";
};

&lpd_dma_chan6 {
    status = "disabled";
};

&lpd_dma_chan7 {
    status = "disabled";
};

&lpd_dma_chan8 {
    status = "disabled";
};
Kernel

Deactivate:

- CONFIG_CPU_IDLE (only needed to fix JTAG Debug issue)
- CONFIG_CPU_FREQ (only needed to fix JTAG Debug issue)

Rootfs

Activate:

- i2c-tools

Applications

startup

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

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

adau1761init

Audio initialisation.
Additional Software

SI5345

Download ClockBuilder Pro for SI5345

1. Install and start ClockBuilder
2. Open "/misc/SI5345/Si5345-RevB-0807-02A-Project.slabtimeproj"
3. Modify settings
4. Export Register File select C code header save to file
5. Replace Header files from FSBL template with generated file
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.

<table>
<thead>
<tr>
<th>Date</th>
<th>Document Revision</th>
<th>Authors</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2019-02-07</td>
<td>v.15</td>
<td>John Hartfiel</td>
<td>Design update</td>
</tr>
<tr>
<td>30.04.2018</td>
<td>v.10</td>
<td>John Hartfiel</td>
<td>Update known issues</td>
</tr>
<tr>
<td>08.02.2018</td>
<td>v.9</td>
<td>John Hartfiel</td>
<td>Design update</td>
</tr>
<tr>
<td>2018-01-29</td>
<td>v.4</td>
<td>John Hartfiel</td>
<td>Update known issues</td>
</tr>
<tr>
<td>2018-01-18</td>
<td>v.3</td>
<td>John Hartfiel</td>
<td>Release 2017.4</td>
</tr>
<tr>
<td>All</td>
<td></td>
<td>John Hartfiel</td>
<td></td>
</tr>
</tbody>
</table>

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

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.

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


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.

2018-09-18