

# TE0783 Test Board

## Table of contents

Error rendering macro 'toc'

java.lang.RuntimeException: com.ctc.wstx.exc.WstxParsingException: String '--' not allowed in comment (missing '>?') at [row,col {unknown-source}]: [39,-1115]

## Overview

Zynq PS Design with Linux Example. Add simple frequency counter to measure SI5338 Reference CLK and RGPIO IP to get access to CPLD IOs with Vivado HW-Manager.

## Key Features

- PetaLinux
- ETH
- USB
- I2C
- RTC
- FMeter
- RGPIO (Beta)
- PL MIG
- Modified FSBL for SI5338 programming
- Special FSBL for QSPI programming

## Revision History

| Date       | Vivado | Project Built   | Authors       | Description     |
|------------|--------|---|---------------|-----------------|
| 2018-06-01 | 2017.4 | TE0783-test_board_noprebuilt-vivado_2017.4-build_10_20180611114036.zip<br>TE0783-test_board-vivado_2017.4-build_10_20180611114017.zip | John Hartfiel | initial release |

## Release Notes and Know Issues

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

## Requirements

### Software

| Software | Version | Note |
|----------|---------|------|
|----------|---------|------|

|                      |        |          |
|----------------------|--------|----------|
| Vivado               | 2017.4 | needed   |
| SDK                  | 2017.4 | needed   |
| PetaLinux            | 2017.4 | needed   |
| SI5338 Clock Builder | ---    | optional |

## 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 | Others | Notes |
|-----------------|-----------------------|----------------------|----------------|------------|--------|-------|
| TE0783-01-45-2I | 45_2i                 | REV01                | 1GB PS, 2GB PL | 32MB       |        |       |

Design supports following carriers:

| Carrier Model | Notes            |
|---------------|------------------|
| TEBT0782      | SD not available |

Additional HW Requirements:

| Additional Hardware     | Notes  |
|-------------------------|--|
| USB Cable for JTAG/UART | Check Carrier Board and Programmer for correct type      |
| XMOD Programmer         | Carrier Board dependent, only if carrier has no own FTDI |

## Content

For general structure and of the reference design, see [Project Delivery - AMD devices](#)

## Design Sources

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

## Additional Sources

| Type   | Location                  | Notes   |
|--------|---------------------------|---|
| SI5338 | <design name>/misc/SI5338 | SI5345 Project with current PLL Configuration |

## 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 SD-Image                       | *.img          | Debian Image for SD-Card   |
| Diverse Reports                       | ---            | Report files in different formats  |
| Hardware-Platform-Specification-Files | *.hdf          | Exported Vivado Hardware Specification for SDK/HSI and PetaLinux             |
| LabTools Project-File                 | *.lpr          | Vivado Labtools Project File   |
| OS-Image                              | *.ub           | Image with Linux Kernel (On Petalinux optional with Devicetree and RAM-Disk) |
| Software-Application-File             | *.elf          | Software Application for Zynq or MicroBlaze Processor Systems                |

## 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:

- [TE0783 "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 lunch.

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

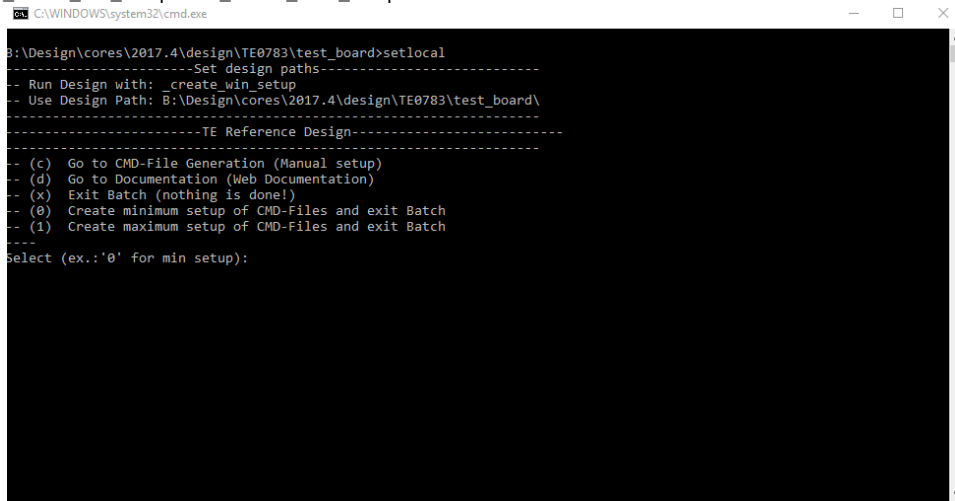
See also:

- [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\_gui mode.cmd"  
Note: Select correct one, see [TE Board Part Files](#)
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 (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.  
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 (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\_gui mode.cmd" or if not created, create with "vivado\_create\_project\_gui mode.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

## SD

Not used on this Example.

## JTAG

Not used on this Example.

## Usage

1. Prepare HW like described on section [Programming](#)
2. Connect UART USB (most cases same as JTAG)
3. Select QSPI as Boot Mode  
Note: See TRM of the Carrier, which is used.
4. Power On PCB  
Note: 1. Zynq Boot ROM loads FSBL from QSPI into OCM, 2. FSBL initialised SI5338 and loads U-boot from QSPI into DDR, 3. U-boot load Linux from QSPI 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. RTC check: `dmesg | grep rtc`
  - c. ETH0 works with udhcpc
  - d. USB type "lsusb" or connect USB2.0 device

## Vivado HW Manager

SI5338 MGT Reference CLKs:

- Open Vivado HW-Manager and add VIO signal to dashboard (\*.ltx located on prebuilt folder).
- Set radix from VIO signals to unsigned integer.  
Note: Frequency Counter is inaccurate and displayed unit is Hz
- SI5338 CLKs are configured to 125MHz with example FSBL initialisation.

PL MIG Status Status signal:

- Status signals connected to VIO

Hardware

localhost (1)

xc7z045\_1 (6)

arm\_dap\_0 (0)

XADC (System Monitor)

hw\_axi\_1 (AXI)

hw\_vio\_1 (zsys\_iVio\_RGP...

hw\_vio\_2 (zsys\_iVio\_cpId)

hw\_vio\_3 (zsys\_iVio\_fmet...

s25fl256s-3.3v-qsapi-x4-si...

Connected

Open

N/A

Programmed

Outputs out-of

OK - Outputs F

OK - Outputs F

hw\_vios

hw\_vio\_1

hw\_vio\_2

hw\_vio\_3

zsys\_i/fm\_mgt\_clk0\_110[31:0]

zsys\_i/fm\_mgt\_clk0\_112[31:0]

zsys\_i/fm\_mgt\_clk1\_109[31:0]

zsys\_i/fm\_mgt\_clk1\_111[31:0]

zsys\_i/fm\_mig\_ui\_clk[31:0]

zsys\_i/labtools\_fmeter\_0\_update

zsys\_i/mig\_7series\_0\_init\_calib\_complete

zsys\_i/mig\_7series\_0\_mmcm\_locked

[U] 124999230

[U] 124999230

[U] 124999230

[U] 124999230

[U] 99997163

[B] 0

[B] 1

[B] 1

↓

↓

↓

↓

↓

↓

Input

Input

Input

Input

Input

Input

Input

Input

hw\_vio\_3

hw\_vio\_3

hw\_vio\_3

hw\_vio\_3

hw\_vio\_3

hw\_vio\_3

hw\_vio\_3

hw\_vio\_3

Custom LED

Red LED D1 can be controlled via VIO.

hw\_vios

hw\_vio\_1

hw\_vio\_2

hw\_vio\_3

zsys\_i/LED[0:0]

[B] 1

Output

hw\_vio\_2

GPIO

GPIO Pins can be controlled via VIO

hw\_vios

hw\_vio\_1 × hw\_vio\_2 hw\_vio\_3

Dashboard Options

Q

≡

⌵

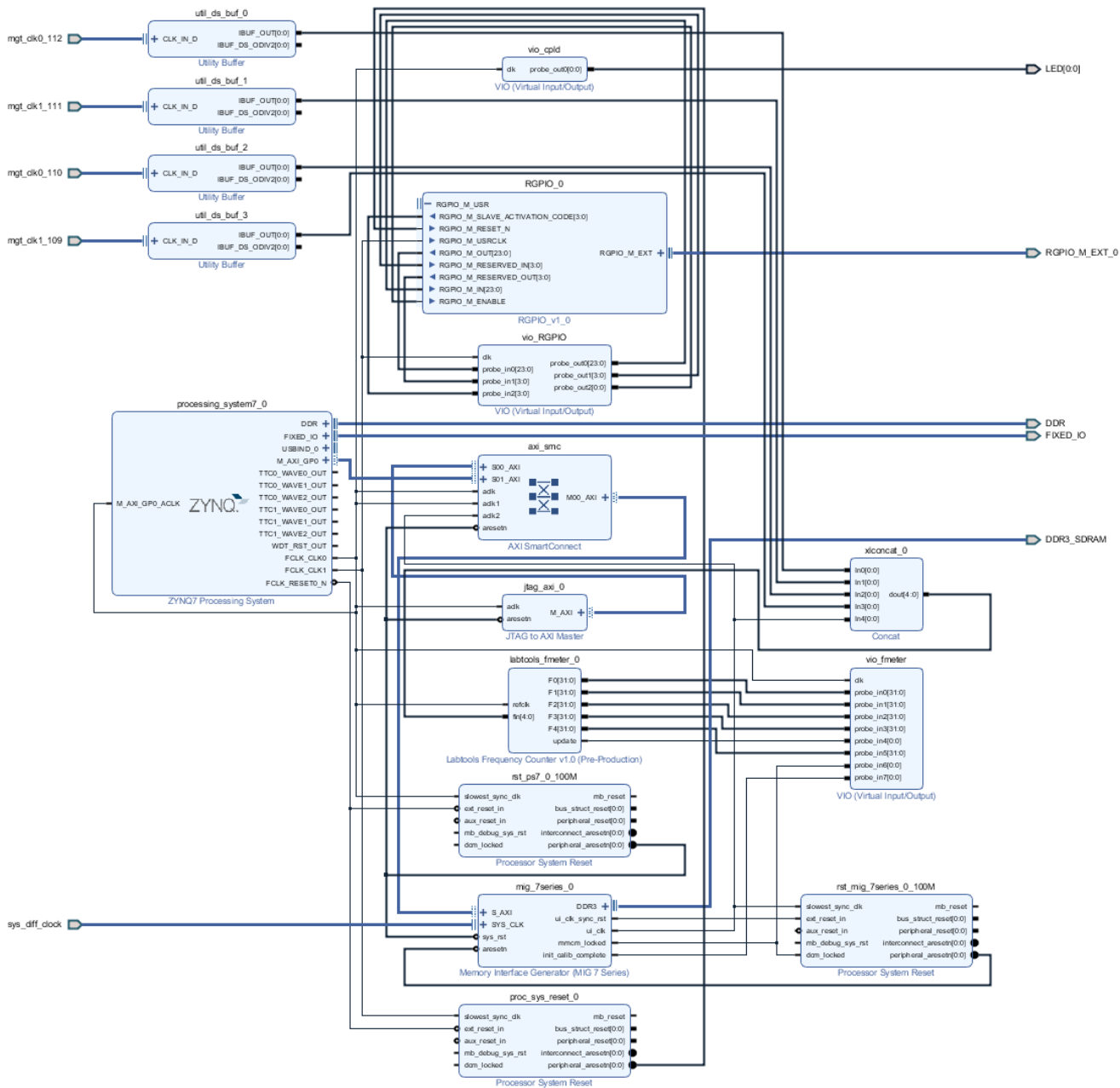
+

−

| Name  | Value       | Activity | Direction | VIO      |
|---|-------------|----------|-----------|----------|
| > zsys_i/RGPIO_0_RGPIO_M_OUT[23:0]                  | [H] AD_EEDF | ↑        | Input     | hw_vio_1 |
| > zsys_i/RGPIO_0_RGPIO_M_RESERVED_OUT[3:0]          | [H] 0       |          | Input     | hw_vio_1 |
| > zsys_i/RGPIO_0_RGPIO_M_SLAVE_ACTIVATION_CODE[3:0] | [H] A       |          | Input     | hw_vio_1 |
| zsys_i/RGPIO_M_ENABLE                               | [B] 1       |          | Output    | hw_vio_1 |
| > zsys_i/RGPIO_M_IN[23:0]                           | [H] AD_EEDF |          | Output    | hw_vio_1 |
| > zsys_i/RGPIO_M_RESERVED_IN[3:0]                   | [H] 0       |          | Output    | hw_vio_1 |

## System Design - Vivado

### Block Design



## PS Interfaces

| Typ  | Note |
|------|------|
| DDR3 |      |
| QSPI | MIO  |
| ETH0 | MIO  |
|      |      |



|          |     |
|----------|-----|
| USB0     | MIO |
| SD0      | MIO |
| SD1      | MIO |
| I2C0     | MIO |
| SWDT0..1 |     |
| TTC0..3  |     |

## Constrains

### Basic module constrains

#### **i\_bitgen\_common.xdc**

```
set_property BITSTREAM.GENERAL.COMPRESS TRUE [current_design]
set_property CONFIG_VOLTAGE 3.3 [current_design]
set_property CFGBVS VCC0 [current_design]
```

### Design specific constrain

#### **i\_io.xdc**

```
#set_property PACKAGE_PIN AA8 [get_ports {SI_MGT_CLK0_110_clk_p[0]}]
#set_property PACKAGE_PIN N8 [get_ports {SI_MGT_CLK0_112_clk_p[0]}]
#set_property PACKAGE_PIN AF10 [get_ports {SI_MGT_CLK1_109_clk_p[0]}]
#set_property PACKAGE_PIN W8 [get_ports {SI_MGT_CLK1_111_clk_p[0]}]
#set_property IOSTANDARD DIFF_SSTL15 [get_ports {MIG_SYS_CLK_clk_p[0]}]
#set_property PACKAGE_PIN H9 [get_ports {MIG_SYS_CLK_clk_p[0]}]
# -----
#LED
set_property PACKAGE_PIN AE20 [get_ports {LED[0]}]
set_property IOSTANDARD LVCMOS33 [get_ports {LED[0]}]
# -----
#RGPIO
set_property PACKAGE_PIN AB19 [get_ports RGPIO_M_EXT_0_clk]
set_property PACKAGE_PIN AB20 [get_ports RGPIO_M_EXT_0_rx]
set_property PACKAGE_PIN AD20 [get_ports RGPIO_M_EXT_0_tx]
set_property IOSTANDARD LVCMOS33 [get_ports RGPIO_M_EXT_0_clk]
set_property IOSTANDARD LVCMOS33 [get_ports RGPIO_M_EXT_0_rx]
set_property IOSTANDARD LVCMOS33 [get_ports RGPIO_M_EXT_0_tx]
```

#### **i\_fm.xdc**

```
set_false_path -from [get_clocks clk_fpga_0] -to [get_clocks {zsys_i/util_ds_buf_0/U0/IBUF_OUT[0]}]
set_false_path -from [get_clocks clk_fpga_0] -to [get_clocks {zsys_i/util_ds_buf_1/U0/IBUF_OUT[0]}]
set_false_path -from [get_clocks clk_fpga_0] -to [get_clocks {zsys_i/util_ds_buf_2/U0/IBUF_OUT[0]}]
set_false_path -from [get_clocks clk_fpga_0] -to [get_clocks {zsys_i/util_ds_buf_3/U0/IBUF_OUT[0]}]
set_false_path -from [get_clocks {zsys_i/util_ds_buf_0/U0/IBUF_OUT[0]}] -to [get_clocks clk_fpga_0]
```

```
set_false_path -from [get_clocks {zsys_i/util_ds_buf_1/U0/IBUF_OUT[0]}] -to [get_clocks clk_fpga_0]
set_false_path -from [get_clocks {zsys_i/util_ds_buf_2/U0/IBUF_OUT[0]}] -to [get_clocks clk_fpga_0]
set_false_path -from [get_clocks {zsys_i/util_ds_buf_3/U0/IBUF_OUT[0]}] -to [get_clocks clk_fpga_0]
```

## Software Design - SDK/HSI

For SDK project creation, follow instructions from:

[SDK Projects](#)

### Application

Source location: \sw\_lib\sw\_apps

#### zynq\_fsbl

TE modified 207.4 FSBL

Changes:

- Si5338 Configuration
  - see main.c, fsbl\_hooks.c
  - Add register\_map.h, si5338.c, si5338.h

#### zynq\_fsbl\_flash

TE modified 2017.4 FSBL

Changes:

- Set FSBL Boot Mode to JTAG
- Disable Memory initialisation

### 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

Deactivate:

- Primary SD/SDIO manual
  - only for usage with TEBT0782

### U-Boot

No changes.

## Device Tree

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

/* default */

/* QSPI */
&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>;
    };
};

/* ETH PHY ETH0 */
&gem0{
    status = "okay";
    phy-handle = <&phy0>;
    xlnx,has-mdio = <0x1>;
    mdio {
        #address-cells = <1>;
        #size-cells = <0>;
        phy0: phy@1 {
            compatible = "marvell,88e1510";
            device_type = "ethernet-phy";
            reg = <1>;
            marvell,reg-init = <0x3 0x10 0x0000 0x0501 0x3 0x11 0x0000 0x4415>;
        };
    };
};

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

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

/* RTC over I2C0 */
```

```
&i2c0 {
    rtc@6F {          // Real Time Clock
        compatible = "isl12022";
        reg = <0x6F>;
    };
};
```

## Kernel

Activate:

- RTC\_DRV\_ISL12022

## Rootfs

Activate:

- i2c-tools

## Applications

## Additional Software

No additional software is needed.

## SI5338

Download [ClockBuilder Desktop for SI5338](#)


1. Install and start ClockBuilder
2. Select SI5338
3. Options Open register map file  
Note: File location <design name>/misc/SI5338/RegisterMap.txt
4. Modify settings
5. Options save C code header files
6. 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.

| Date                              | Document Revision                 | Authors                           | Description   |
|-----------------------------------|-----------------------------------|-----------------------------------|---|
| Error rendering macro 'page-info' | Error rendering macro 'page-info' | Error rendering macro 'page-info' | <ul style="list-style-type: none"><li>• typo correction</li></ul> |

|  |   |  |   |
|--|---|--|---|
| <p>Ambiguous method overloading for method jdk.proxy241.\$Proxy3496#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]</p> | <p>Ambiguous method overloading for method jdk.proxy241.\$Proxy3496#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]</p> <div>  Unknown macro: 'metadata' </div> | <p>Ambiguous method overloading for method jdk.proxy241.\$Proxy3496#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]</p> |   |
| 11 Jun 2018  | v.4   | John Hartfiel  | Release 2017.4  |
| 2018-05-30   | v.1   | <div> <p><b>Error rendering macro 'page-info'</b></p> <p>Ambiguous method overloading for method jdk.proxy241.\$Proxy3496#hasContentLevelPermission. Cannot resolve which method to invoke for [null, class java.lang.String, class com.atlassian.</p> </div>  | <ul style="list-style-type: none"> <li>Initial release</li> </ul> |

|  |     |   |  |
|--|-----|---|--|
|  |     | <div>confluence.pages.Page]<br/>due to overlapping<br/>prototypes between:<br/>[interface com.atlassian.<br/>confluence.user.<br/>ConfluenceUser, class java.<br/>lang.String, class com.<br/>atlassian.confluence.core.<br/>ContentEntityObject]<br/>[interface com.atlassian.<br/>user.User, class java.lang.<br/>String, class com.atlassian.<br/>confluence.core.<br/>ContentEntityObject]</div>  |  |
|  | All | <div><b>Error rendering macro<br/>'page-info'</b><br/><br/>Ambiguous method<br/>overloading for method jdk.<br/>proxy241.\$Proxy3496#has<br/>ContentLevelPermission.<br/>Cannot resolve which<br/>method to invoke for [null,<br/>class java.lang.String,<br/>class com.atlassian.<br/>confluence.pages.Page]<br/>due to overlapping<br/>prototypes between:<br/>[interface com.atlassian.<br/>confluence.user.<br/>ConfluenceUser, class java.<br/>lang.String, class com.<br/>atlassian.confluence.core.<br/>ContentEntityObject]<br/>[interface com.atlassian.</div> |  |

|  |   |  |
|--|---|--|
|  | <pre>user.User, class java.lang.<br/>String, class com.atlassian.<br/>confluence.core.<br/>ContentEntityObject]</pre> |  |
|--|---|--|

## 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**

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.proxy241.$Proxy3496#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]`