TE0720 SDSoC EDDP FOC with TE0706

Table of contents

- 1 Table of contents
- 2 Overview
  - 2.1 Key Features
  - 2.2 Revision History
  - 2.3 Release Notes and Know Issues
  - 2.4 Requirements
    - 2.4.1 Software
    - 2.4.2 Hardware
  - 2.5 Content
    - 2.5.1 Design Sources
    - 2.5.2 Additional Sources
    - 2.5.3 Download
- 3 Hardware Setup
- 4 Design Setup
  - 4.1 Create SDSoC Platform from TE Reference Design
  - 4.2 Set TE SDSoC Platform as local SDSoC Platform
  - 4.3 Create SDSoC Project
- 5 Launch
- 6 References
- 7 Appx. A: Change History and Legal Notices
  - 7.1 Document Change History
  - 7.2 Legal Notices
  - 7.3 Data Privacy
  - 7.4 Document Warranty
  - 7.5 Limitation of Liability
  - 7.6 Copyright Notice
  - 7.7 Technology Licenses
  - 7.8 Environmental Protection
  - 7.9 REACH, RoHS and WEEE

Overview

Key Features

- EDDP with FOC algorithm designed in SDSoC for TE0706 carrier board.
- Automated generation of SDSoC platforms for family of TE0720 modules.

Revision History

Date	Vivado	Project Built	Authors	Description
2018-08-27	2017.1	TE0706_zsys_SDSoC_EDDP_FOC-vivado_2017.1-build_05_20180827095117.zip	UTIA	initial release

Release Notes and Know Issues

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

Requirements

Software

Software	Version	Note
PetaLinux	2017.1	needed
SDx	2017.1	needed

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
TE0720-03-2IF	TE0720_2IF	REV03	1 GB	32		
TE0720-03-l1if	TE0720_L1IF	REV03	512MB (L)	32		
TE0720-03-1CF	TE0720_1CF	REV03	1 GB	32		
TE0720-03-2EF	TE0720_2EF	REV03	1 GB	32		
TE0720-03-07S	TE0720_07S	REV03	1 GB (L)	32		

Design supports following carriers:

Carrier Model	Notes
TE0706-2	

Additional HW Requirements:

Additional Hardware	Notes
TEC0053-04 - EDPS Power Stage	https://shop.trenz-electronic.de/en/TEC0053-04-EDPS-Power-Stage?c=474
BLDC Motor with mounted Encoder (1000SI)	https://shop.trenz-electronic.de/en/28170-BLDC-Motor-with-mounted-Encoder-1000SI?c=474
Interchangeable Plug with four adapters and cable, 12V/2.5A (12V for TEC0053-04)	https://shop.trenz-electronic.de/en/28169-Interchangeable-Plug-with-four-adapters-and-cable-12V/2.5A?c=35
Universal plug-in power supply unit with four adapters and cable, 15W 5V/3A (5V for TE0706)	https://shop.trenz-electronic.de/en/28485-Universal-plug-in-power-supply-unit-with-four-adapters-and-cable-15W-5V/3A?c=35
2x Pmod Cable Kit: 12-pin	https://shop.trenz-electronic.de/en/26742-Pmod-Cable-Kit-12-pin?c=37
Pmod Cable Kit: 6 pin cable connector kit, 30 cm (12") in length	https://shop.trenz-electronic.de/en/25250-Pmod-Cable-Kit-6-pin-cable-connector-kit-30-cm-12-in-length?c=37
XMOD FTDI JTAG Adapter - Xilinx compatible	https://shop.trenz-electronic.de/en/TE0790-02-XMOD-FTDI-JTAG-Adapter-Xilinx-compatible?path=Trenz_Electronic/JTAG_Programmer/TE0790/REV02/Documents

Content

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

Design Sources

Type	Location	Notes
Vivado	<design name>/block_design <design name>/constraints <design name>/ip_lib	Vivado Project will be generated by TE Scripts
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
SDSoC	<design name>/../SDSoC_PFM	SDSoC Platform will be generated by TE Scripts

Additional Sources

Type	Location	Notes

Download

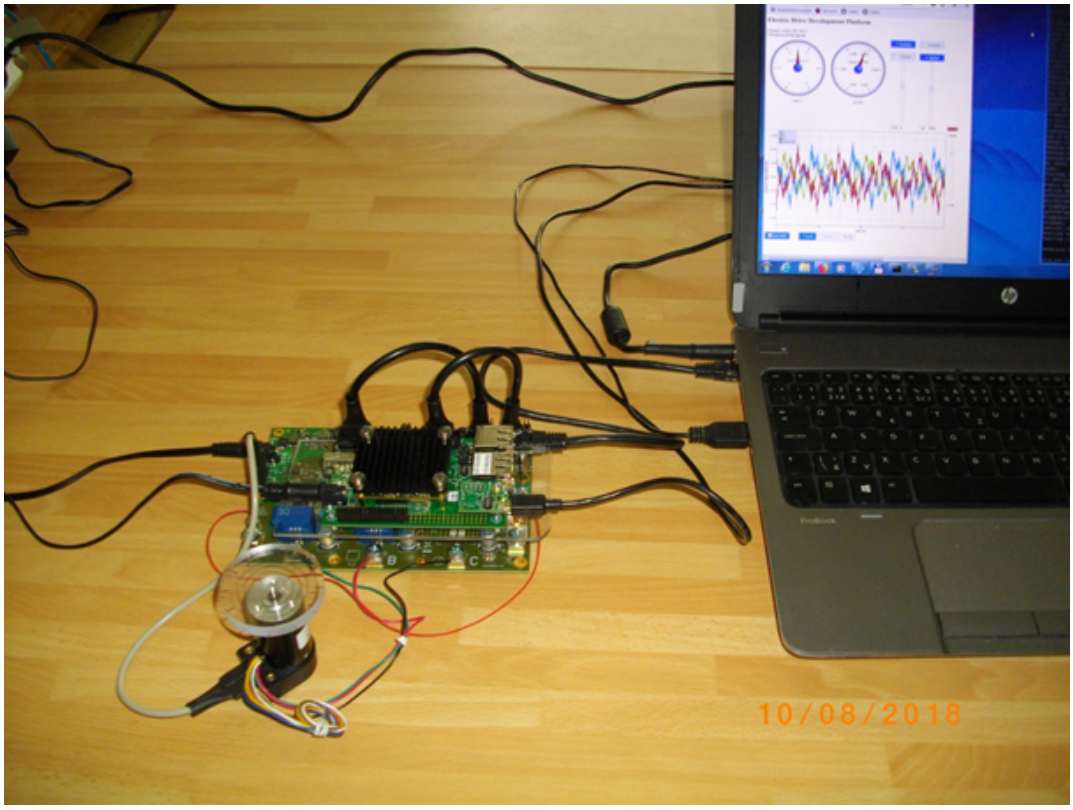
Reference Design is only usable with the specified Vivado/SDK/PetaLinux/SDx version. Do never use different Versions of Xilinx Software for the same Project.

Reference Design is available on:

- TE0720 "TE0720 SDSoC EDDP FOC with TE0701" Reference Design

Hardware Setup

TE0706_zsys_SDSoC_EDDP_FOC 2017.1 Platform with TEC0053-04 - EDPS Power Stage



3-phase brush-less DC motor control with field oriented control (FOC) algorithm implemented in SDSoC 2017.1 on TE0720 module and TE0706-02 carrier board. The TEC0053-04 - EDPS Power Stage controls the BLDC motor with mounted encoder.

How to setup hardware shown in the figure above is described in following steps:

1. On TE0706-02 carrier board, use jumpers J10, J11 and J12 to select 3.3V:

Jumper	Settings
J10	Short 2-3
J11	Short 2-3
J12	Short 2-3

2. On TE0706-02 carrier board, set switch S1 to:

Switch	Settings
S1_1	OFF
S1_2	OFF
S1_3	OFF

S1_4	ON

3. On TE0790-02 XMOD FTDI JTAG Adapter of the TE0706-02 board, set switch S2 to:

Switch	Settings
S2_1	ON
S2_2	OFF
S2_3	ON
S2_4	OFF

> **IMPORTANT**
>
> Before connecting to TEC0053-04 by PMOD 12pin cables, power on the TE0706-02 (NOTE: 5V!) and measure presence of 3.3V voltage on the TE0706-02 connector J5 pins: J5:5, J5:6, J5:45, J5:46.

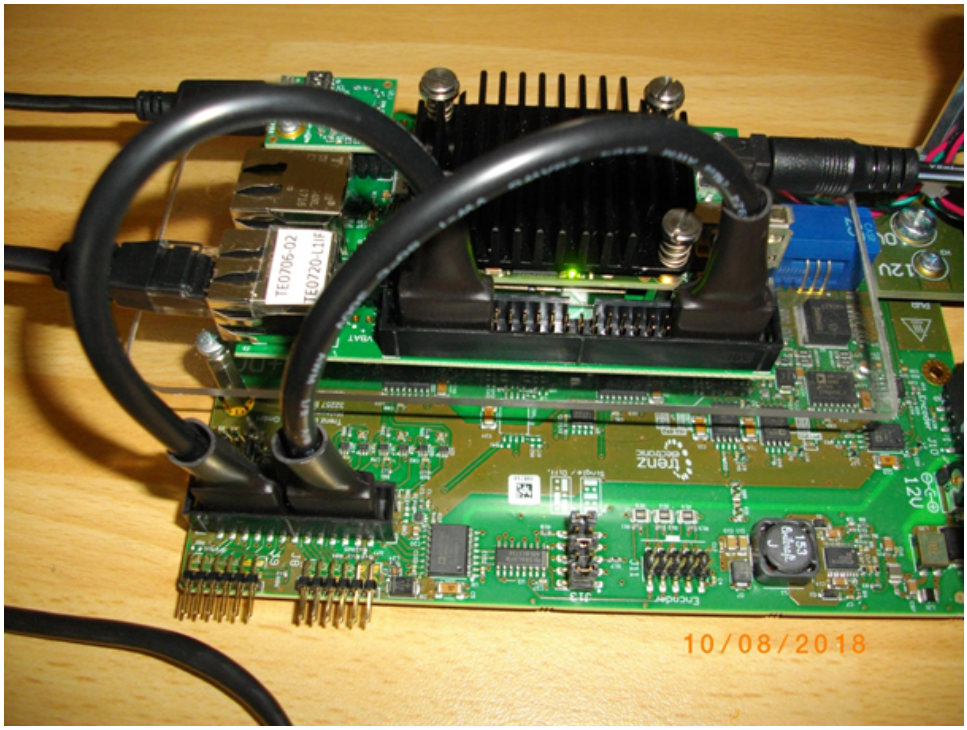
4. Connection of motor rotation encoder

Motor rotation encoder is connected to the TEC0053-04 - EDPS Power Stage Pmod 6 pin cable connector kit. See the orientation and position of the 5 wire connection. There are 6 pins on the TEC0053-04 board. There are only 5 pins on the motor encoder. Pin 1 connects to pin 1. 6-th wire is unconnected on the motor encoder side.

3-phase of the motor phase wires are connected to the TEC0053-04 Power Stage points A, B, C: A: green wire; B: red wire; C: black wire. All other motor wires are unused.

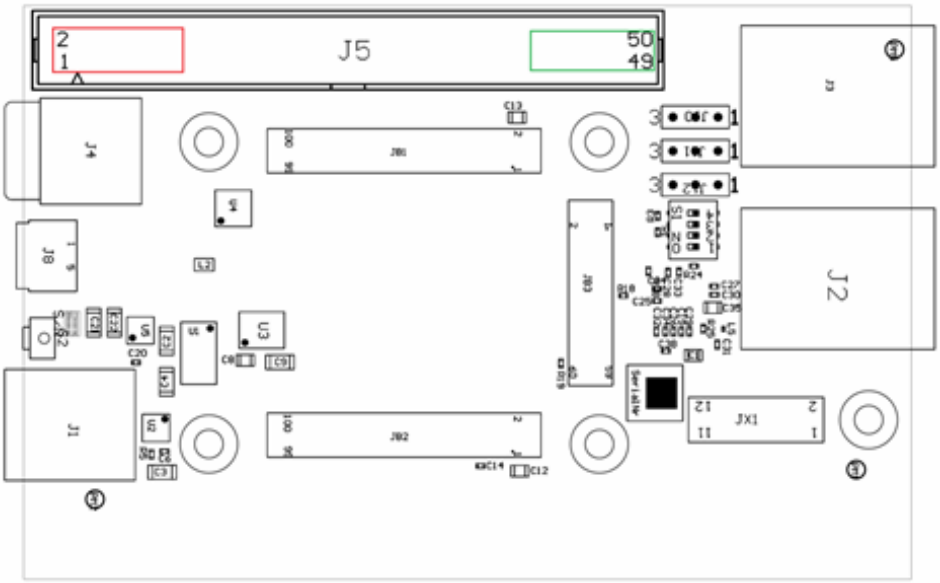
The TEC0053-04 - EDPS Power Stage can be powered by 12 V from the power supply by separate wire connecting of the point labled “PWR” (see above) with the fused point labeled “+DC” (see the first picture).

5. Connect TE0706 with TEC0053-04 by two PMOD 12-pin cables.

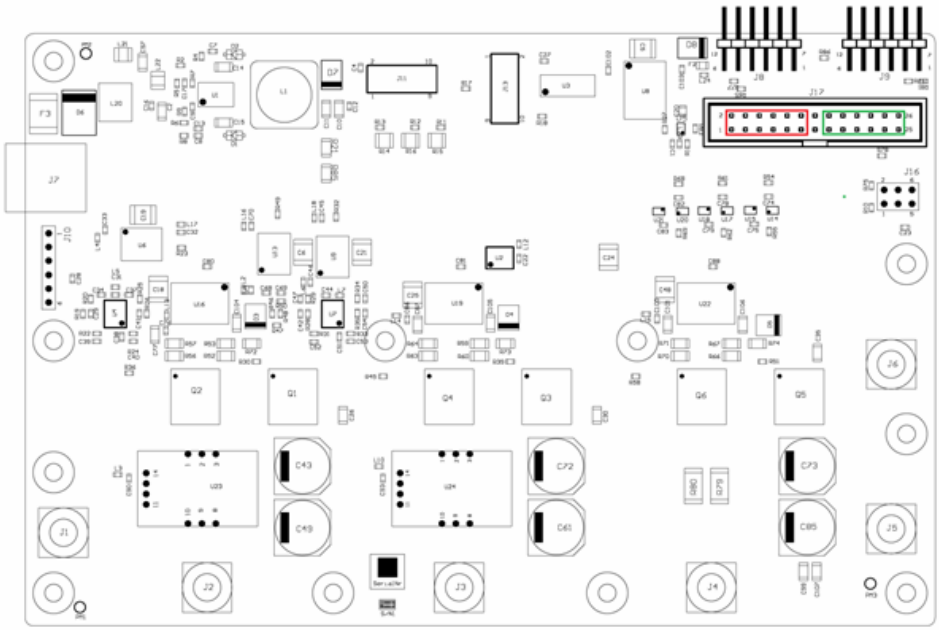


Description of connections of TE0706 with TEC0053-04

```
# First 12 pin PMOD cable
# Connections of TEC0053 J17 with TE0706-02 J5                    # TEC0053         # TE0706-02
# GND
# J17:25        # J5:50
set_property PACKAGE_PIN G20 [get_ports {SCLK}];                 # J17:23         # J5:48
# 3,3V
# J17:21         # J5:46
set_property PACKAGE_PIN E21 [get_ports {SDI1}];                 # J17:19         # J5:44
set_property PACKAGE_PIN B19 [get_ports {SDI2}];                 # J17:17         # J5:42
set_property PACKAGE_PIN D20 [get_ports {SDI3}];                 # J17:15         # J5:40
# GND
# J17:26        # J5:49
set_property PACKAGE_PIN G21 [get_ports {SDV}];                  # J17:24         # J5:47
# 3,3V
# J17:22         # J5:45
set_property PACKAGE_PIN D21 [get_ports {ENC_A}];                # J17:20         # J5:43
set_property PACKAGE_PIN B20 [get_ports {ENC_B}];                # J17:18         # J5:41
set_property PACKAGE_PIN C20 [get_ports {ENC_I}];                # J17:16         # J5:39
# All signals connected by the first PMOD cable cable belong to TE0720 Zynq Bank 35.
# Second 12 pin PMOD cable
# Connections of TEC0053 J17 with TE0706-02 J5                    # TEC0053         # TE0706-02
# GND
# J17:1               # J5:2
# Not used in the design                                                            # J17:
3                 # J5:4
# 3,3V
# J17:5                # J5:6
set_property PACKAGE_PIN W17 [get_ports {GH[2]}];                # J17:7           # J5:8
set_property PACKAGE_PIN W20 [get_ports {GH[1]}];                # J17:9           # J5:10
set_property PACKAGE_PIN AA16 [get_ports {GH[0]}];               # J17:11          # J5:12
# GND
# J17:2               # J5:1
# Not used in the design                                                            # J17:
4                 # J5:3
# 3,3V
# J17:6                # J5:5
set_property PACKAGE_PIN W18 [get_ports {GL[2]}];                # J17:8           # J5:7
set_property PACKAGE_PIN W21 [get_ports {GL[1]}];                # J17:10          # J5:9
set_property PACKAGE_PIN AB16 [get_ports {GL[0]}];               # J17:12          # J5:11
#All signals connected by the second PMOD cable belong to TE0720 Zynq Bank 33.
# Second PMOD Cable 12-pin contains these two wires unconnected to the SDSoC design:
# Unused connections of TEC0053 J17 with TE0706-02 J5                    # TEC0053         # TE0706-02
#set_property PACKAGE_PIN W16 [get_ports {gpio_0_tri_io[1]}];         # J17:3         # J5:4
#set_property PACKAGE_PIN Y16 [get_ports {gpio_0_tri_io[0]}];   # J17:4          #J5:3
```



TE0706-02 carrier board has highlighted position of two 12 pin PMOD cables. Please take care, when connecting both cables to respect pin connections as listed above.



TEC0053-04 power stage board has highlighted position of two 12 pin PMOD cables.

IMPORTANT

Wrong placement of the PMOD 12pin connects might result in a damaged electronic.

Design Setup

Create SDSoC Platform from TE Reference Design

1. Unzip Reference Design
    a. Do not change base folder name after extraction!
    <install_path>\TE0706_zsys_SDSoC_EDDP_FOC\zsys\
    CD to the directory and run from win terminal:
    _create_win_setup.cmd

    run from win terminal:
    _use_virtual_drive.cmd
    reply to select an virtual drive name (example X): X
    reply: 0

    cd X:\zsys

    This is shortest possible path and directory name for building of the platform
    in windows (to respect the 260 character limitations.)

> **NOTE**
>
> Do not change the name of the directory /zsys
> It has to be identical to the shortest possible platform name “zsys”
> for the Zynq 7000 targets.

2. Enable SDSOC, set install path of Xilinx tools, set your hardware assembly option in: "design_basic_settings.cmd" Select one of these supported modules (1,4,5,6,7):

ID	TE Module
1	te0720-03-2if
4	te0720-03-l1if
5	te0720-03-1cf
6	te0720-03-2ef
7	te0720-03-07s

> **NOTE**
>
> Selection 7 supports the TE0720-03-14S-1C module (xc7z014sclg484-1c device).

3. Create Reference Design: run "vivado_create_project_guimode.cmd"
4. VIVADO:

    a. TCL-Console type: TE::hw_build_design -export_prebuilt
    b. Find hardware handoff file .hdf under prebuilt folder abd copy it to Ubuntu 16.4, with Petalinux 2017.1.

> **IMPORTANT**
>
> The executable flag under Linux OS must be set for files:
>
> ./init_config.sh
> ./project-spec/meta-user/recipes-apps/libuv/files/checksparse.sh
> ./project-spec/meta-user/recipes-apps/libuv/files/gyp_uv.py
> ./project-spec/meta-user/recipes-apps/libuv/files/autogen.sh
> ./project-spec/meta-user/recipes-apps/libuv/files/android-configure
> ./project-spec/meta-user/recipes-apps/libuv/update-src.sh

In Ubuntu 16.04, build Petalinux image image.ub and uboot u-boot.elf using Petalinux BSP provided under os folder and place new images to correct subfolder in prebuilt/os

    c. TCL-Console type: TE::sw_run_hsi
    d. TCL-Console type:TE::ADV::beta_util_sdsoc_project
        i. Vivado project will be modified by copying constrain files locally to project.

> **NOTE**
>
> If needed, recreate project with batch file to restore original Vivado project with externally linked constrains.

    e. Wait for project creation:

        i. SDSoC Platform is created in
        X:\zsys\SDSoC_PFM\<TE::SHORTDIR>\zsys
        ii. Copy
        X:\zsys_use_virtual_drive.cmd
        to
        X: \SDSoC_PFM\<TE::SHORTDIR>\zsys
        iii. Copy these two files from:
        X:\zsys\init.sh
        X:\zsys\focserver.conf
        to
        X:\SDSoC_PFM\<TE::SHORTDIR>\zsys\sw\linux\image\init.sh
        X:\SDSoC_PFM\<TE::SHORTDIR>\zsys\sw\linux\image\focserver.conf
        iv. Copy directory with all files from:
        X:\zsys\misc\src\
        to
        X: \SDSoC_PFM\<TE::SHORTDIR>\zsys\src\

v. v. Copy directory with all files from:
X:\zsys\misc\sw\aarch32-linux\
X:\zsys\misc\sw\aarch32-none\
to
X: \SDSoC_PFM\<TE::SHORTDIR>\zsys\sw\aarch32-linux\
X: \SDSoC_PFM\<TE::SHORTDIR>\zsys\sw\ aarch32-none\

f. Close current Vivado project
g. Clear working project files by script
X:\zsys\design_clear_design_folders.cmd
h. From win terminal, execute:
_use_virtual_drive.cmd
reply to select an virtual drive name (example X): X
reply: 1
This will disconnect the virtual X: drive
i. Compile support libraries serving for connection to 64bit AXI I/O.
Open the SDx Terminal 2017.1
CD to: <install_path>\TE0706_zsys_SDSoC_EDDP_FOC\SDSoC_PFM\<TE::SHORTDIR>\zsys\src\
In the SDx Terminal 2017.1, run batch file:
build_linux.bat
Library for the SDSoC Linux target is created:
<install_path>\TE0706_zsys_SDSoC_EDDP_FOC\SDSoC_PFM\<TE::SHORTDIR>\zsys\src\libte0720_foc.a
Move the created library libte0720_foc.a to
c:\TV71u\TE0706_zsys_SDSoC_EDDP_FOC\SDSoC_PFM\te0720_2if\zsys\sw\aarch32-linux\lib\libte0720_foc.a
Delete the created _sds directory
<install_path>\TE0706_zsys_SDSoC_EDDP_FOC\SDSoC_PFM\<TE::SHORTDIR>\zsys\src_sds

In the SDx Terminal 2017.1, run batch file:
build_standalone.bat
Library for the SDSoC standalone target is created:
<install_path>\TE0706_zsys_SDSoC_EDDP_FOC\SDSoC_PFM\<TE::SHORTDIR>\zsys\src\libte0720_foc.a
Move the created library libte0720_foc.a to
<install_path>\TE0706_zsys_SDSoC_EDDP_FOC\SDSoC_PFM\<TE::SHORTDIR>\\zsys\sw\aarch32-none\lib\ libte0720_foc.a
Delete the created _sds directory
<install_path>\TE0706_zsys_SDSoC_EDDP_FOC\SDSoC_PFM\<TE::SHORTDIR>\zsys\src_sds
The SDSoC platform for the target \<TE::SHORTDIR> is in
<install_path>\TE0706_zsys_SDSoC_EDDP_FOC\SDSoC_PFM\<TE::SHORTDIR>

Close the SDx Terminal 2017.1

Set TE SDSoC Platform as local SDSoC Platform

1. Use the created SDSoC Platform for <TE::SHORTDIR> module present in the directory:
Open new windows terminal and CD to:

<install_path>\TE0706_zsys_SDSoC_EDDP_FOC\SDSoC_PFM\<TE::SHORTDIR>
   a. From win terminal, run
   _use_virtual_drive.cmd
   reply to select an virtual drive name (example X): X
   reply: 0
   CD to:
   X:\<TE::SHORTDIR>
   This is shortest possible path and directory name for the SDSoC project working with the created SDSoC 2017.1 platform in the directory:
   X:\<TE::SHORTDIR>\zsys

Create SDSoC Project

1. Start SDSoC 2017.1 in the directory
2. Select Workspace

   X:\ <TE::SHORTDIR>
3. Click "Create SDSoC Project"
   a. Set Project Name (example: foc01)
   b. Set Platform:

        i. Others. Path to Project is:
           X:\ <TE::SHORTDIR>\zsys
    c. Select OS: Linux
    d. Click "Next"
    e. Select Template Application "focserver" "Field Oriented Control with Web UI"
    f. Click "Finished"
    g. Right click on the project -> C/C++ Build Settings
       In the top level Configuration menu select [All configurations]
    h. Add libraries 'te0720_foc' and 'dl' to the linker flags! -> SDS++ Linker -> Libraries
    i. Add path to directory with Linux version of the 'libte0720_foc.a' library! -> SDS++ Linker -> Libraries
       Example for <TE::SHORTDIR> = te0720_2if:
       "X:/te0720_2if/zsys/sw/aarch32-linux/lib"
    j. Add path to directory with te0720_foc.h! -> SDSCC Compiler -> Directories
       Example for <TE::SHORTDIR> = te0720_2if:
       "X:/te0720_2if/zsys/sw/aarch32-linux/include"
    k. Add path to directory with te0720_foc.h! -> SDS++ Compiler -> Directories
       Example for <TE::SHORTDIR> = te0720_2if:
       "X:/te0720_2if/zsys/sw/aarch32-linux/include"
    l. In main SDx Project Settings:
       unselect box [] Generate bitstream
       unselect box [] Generate SD card image

       these two unselections will accelerate the initial compilation of the platform, needed for creation of the final platform hdf file needed for generation of the final image.ub in the Petalinux 2017.1 under the Ubuntu. (cca 4 min instead of 30 min with these options selected).

       The .hdf description of the foc01 HW design and related drivers is created in file (Example for <TE::SHORTDIR> = te0720_2if):

       X:\te0720_2if\foc01\Debug_sds\p0\ipi\zsys.sdk\zsys.hdf
    m. Copy created file zsys.hdf to Petalinux 2017.1 in Ubuntu 16.04 and recompile the configuration of Petalinux with this .hdf file.

       Result of this compilation is updated image.ub which includes device tree with the AXI-lite driver, created by the SDSoC initial compilation step. This driver is used by the focserver to set parameters of the HW accelerated SDSoC implementation of the FOC algorithm.
    n. Replace the initial image.ub of the SDSoC platform with the created final image.ub by copy to (Example for <TE::SHORTDIR> = te0720_2if):
       X:\te0720_2if\zsys\sw\linux\image\image.ub
    o. In SDSoC, clear the foc01 project.
    p. In main SDx Project Settings:
       select box [x] Generate bitstream
       select box [x] Generate SD card image
    q. Select Build project foc01
       The SDSoC project is recompiled (cca 30 min) with foc01 integrated in HW.
       SDCard image is created

Launch

1. Copy created files to the SD card.
2. ON PC, set the Ethernet address to 192.168.42.100
3. Connect PC with Ethernet cable to the TE0706 board.
4. Connect serial terminal via the USB cable.
5. Power ON TEC0053-04 - EDPS Power Stage (12V).
6. Power ON TE0706 board (12V).
7. On PC, open serial terminal.
8. Reset TE0705 board (by S2 button).
9. Boot of Linux starts up to login stage. Login as 'root' with password 'root'.
   To see top running processes, type
   top
   you can see running process
   focserver
10. On PC, open www browser and connect to
    http://192.168.42.123
    to connect to the focserver running on the TE0720 module.
11. Use the WWW GUI to start and control the BLDC motor and to visualize data.

References

- SDSoC Environment - User Guide (UG1027)
- SDSoC Environment User Guide - An Instruction to SDSoC Environment (UG1028)
- SDSoC Environment User Guide - Platforms and Libraries (UG1146)
- EDDP Resources - Sources and documentation of the original EDDP Development kit

Appx. A: Change History and Legal Notices

Document Change History

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

Date	Document Revision	Authors	Description
Error rendering macro 'page-info' Ambiguous method overloading for method jdk.proxy244.$Proxy3589#hasContentLevelPermission. Cannot resolve which method to invoke for [null, class java.lang.String, class com.atlassian.confluence.pages.Page] due to overlapping prototypes between: [interface com.atlassian.confluence.user.ConfluenceUser, class java.lang.String, class com.atlassian.confluence.core.ContentEntityObject] [interface com.atlassian.user.User, class java.lang.String, class com.atlassian.confluence.core.ContentEntityObject]	**Error rendering macro 'page-info'** Ambiguous method overloading for method jdk.proxy244.$Proxy3589#hasContentLevelPermission. Cannot resolve which method to invoke for [null, class java.lang.String, class com.atlassian.confluence.pages.Page] due to overlapping prototypes between: [interface com.atlassian.confluence.user.ConfluenceUser, class java.lang.String, class com.atlassian.confluence.core.ContentEntityObject] [interface com.atlassian.user.User, class java.lang.String, class com.atlassian.confluence.core.ContentEntityObject] Unknown macro: 'metadata'	**Error rendering macro 'page-info'** Ambiguous method overloading for method jdk.proxy244.$Proxy3589#hasContentLevelPermission. Cannot resolve which method to invoke for [null, class java.lang.String, class com.atlassian.confluence.pages.Page] due to overlapping prototypes between: [interface com.atlassian.confluence.user.ConfluenceUser, class java.lang.String, class com.atlassian.confluence.core.ContentEntityObject] [interface com.atlassian.user.User, class java.lang.String, class com.atlassian.confluence.core.ContentEntityObject]	• update download

23 Aug 2018	v.7	UTIA	• 2017.1 release
2018-08-15	v.1	**Error rendering macro 'page-info'** Ambiguous method overloading for method jdk.proxy244.$Proxy3589#hasContentLevelPermission. Cannot resolve which method to invoke for [null, class java.lang.String, class com.atlassian.confluence.pages.Page] due to overlapping prototypes between: [interface com.atlassian.confluence.user.ConfluenceUser, class java.lang.String, class com.atlassian.confluence.core.ContentEntityObject] [interface com.atlassian.user.User, class java.lang.String, class com.atlassian.confluence.core.ContentEntityObject]	• Initial release
	All	**Error rendering macro 'page-info'**	

		Ambiguous method overloading for method jdk.proxy244.$Proxy3589#hasContentLevelPermission. Cannot resolve which method to invoke for [null, class java.lang.String, class com.atlassian.confluence.pages.Page] due to overlapping prototypes between: [interface com.atlassian.confluence.user.ConfluenceUser, class java.lang.String, class com.atlassian.confluence.core.ContentEntityObject] [interface com.atlassian.user.User, class java.lang.String, class com.atlassian.confluence.core.ContentEntityObject]	

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.proxy244.$Proxy3589#hasContentLevelPermission. Cannot resolve which method to invoke for [null, class java.lang.String, class com.atlassian.confluence.pages.Page] due to overlapping prototypes between: [interface com.atlassian.confluence.user.ConfluenceUser, class java.lang.String, class com.atlassian.confluence.core.ContentEntityObject] [interface com.atlassian.user.User, class java.lang.String, class com.atlassian.confluence.core.ContentEntityObject]