TE0720 SDSoC EDDP FOC with TE0701

Table of contents

Overview

Key Features

- EDDP with FOC algorithm designed in SDSoC for TE0701 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	TE0701_zsys_SDSoC_EDDP_FOC-vivado_2017.1-build_05_20180827095945.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
TE0701-6	

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	https://shop.trenz-electronic.de/en/28169-Interchangeable-Plug-with-four-adapters-and-cable-12V/2.5A?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

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

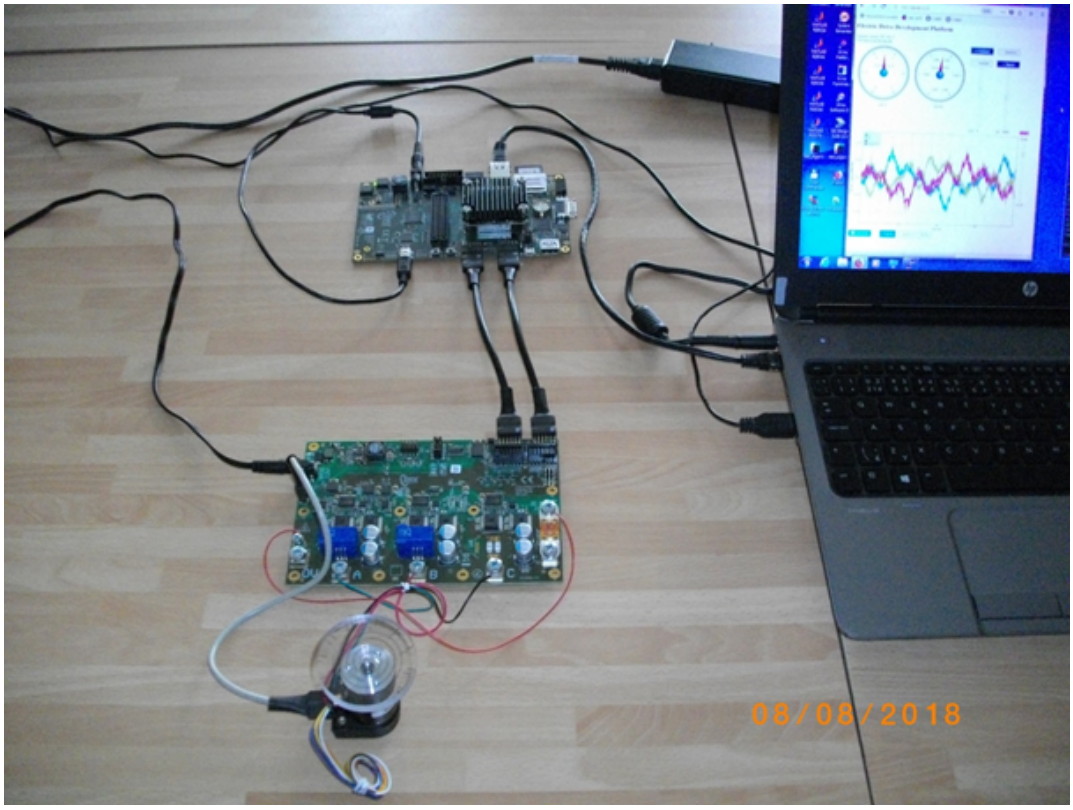
Reference Design is only usable with the specified Vivado/SDK/PetaLinux/SDx version. Do never use different Versions of Xilinx Software for the same Project.

Reference Design is available on:

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

Hardware Setup

TE0701_zsys_SDSoC_EDDP_FOC 2017.1 platform with TEC0053 Power Stage



3-phase brush-less DC motor control with field oriented control (FOC) algorithm implemented in SDSoC 2017.1 on TE0720 module and TE0701-06 carrier board. The TEC0053-04 - EDPS Power Stage controls the BLDC Motor with mounted Encoder.

The following steps are describing how to connect and setup hardware parts shown in the Figure above.

1. Set TE0701-06 carrier board FMC_VADJ = 3.3V by switch S4:

S4	Setup
S4_1	ON
S4_2	ON
S4_3	ON
S4_4	OFF

2. Set FMC_VADJ (set in step 1 to 3.3V) to drive both, the VIOTA and VIOTB by this arrangement of J16, J17 and J21.

Jumper	Configuration
J6	Short 1-2
J17	no connection

J21	Short 2-3

VIOTA will provide 3.3V to PMOD J5 and FMC_VADJ will provide 3.3V to PMOD J6

3. Set switch S3 of the TE0701-06 carrier board to:

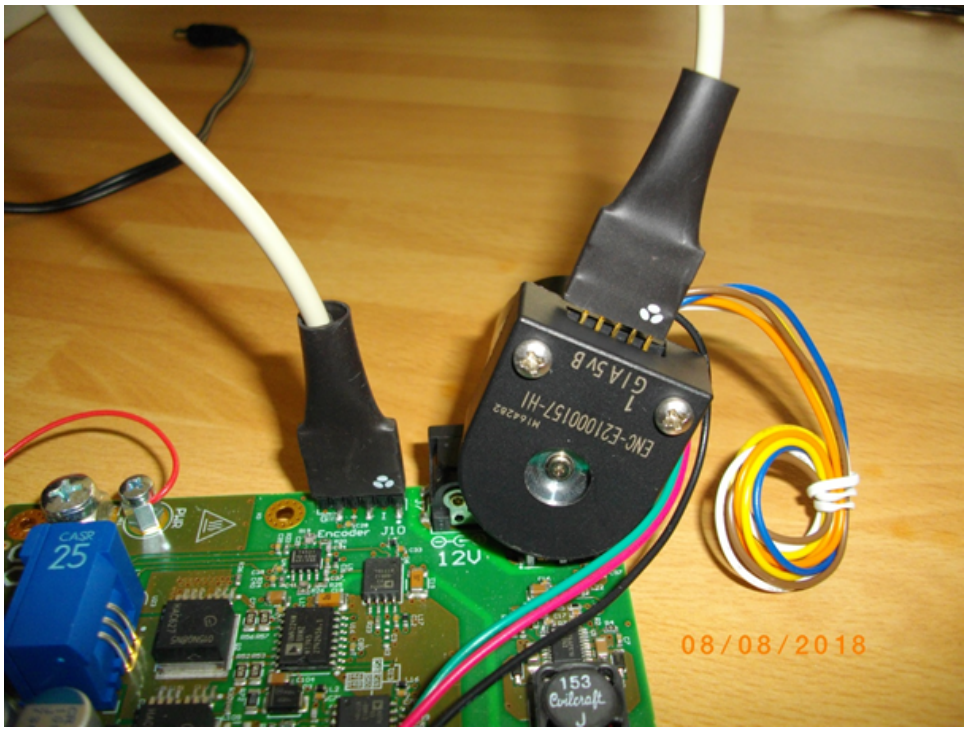
S3	Setup
S3_1	any
S3_2	any
S3_3	ON
S3_4	OFF

S3_1 and S3_2 serve as general purpose pins connected to the CPLD on the TE0701-06.

> **IMPORTANT**
>
> Before connecting to TEC0053-04 by PMOD 12pin cables, power on the TE0701-06 (12V) and measure presence of the 3.3V voltage on the TE0701-06 PMOD J5 pin 12 and pin 6 and on the TE0701-06 PMOD J6 pin 12 and pin 6.

4. Connect TEC0053 to two 3.3V PMOD connectors on TE0701 carrier with two Pmod 12-pin cables as shown in following image.

5. Connect of motor rotation encoder as shown in following figure



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 phases are connected to the TEC0053-04 - EDPS Power Stage points A, B and 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 labled “+DC” (see the first picture).

Description of connections of TE0701 with TEC0053-04

```
# First PMOD cable 12-pin:

# Connections of                                        # TE0701-06 J5                with TEC0053
J9
set_property PACKAGE_PIN W18 [get_ports {SDV}];           #TE0701-06 J5:7         with TEC0053 J9:7
set_property PACKAGE_PIN W17 [get_ports {ENC_A}];   #TE0701-06 J5:8         with TEC0053 J9:8
set_property PACKAGE_PIN Y19 [get_ports {ENC_B}];         #TE0701-06 J5:9         with TEC0053 J9:9
set_property PACKAGE_PIN AA19 [get_ports {ENC_I}];        #TE0701-06 J5:10         with TEC0053 J9:10
# GND                                                                                    #TE0701-
06 J5:11         with TEC0053 J9:11
# 3,3V                                                                                    #TE0701-
06 J5:12         with TEC0053 J9:12
set_property PACKAGE_PIN Y16 [get_ports {SCLK}];          #TE0701-06 J5:1         with TEC0053 J9:1
set_property PACKAGE_PIN W16 [get_ports {SDI1}];          #TE0701-06 J5:2         with TEC0053 J9:2
set_property PACKAGE_PIN Y18 [get_ports {SDI2}];          #TE0701-06 J5:3         with TEC0053 J9:3
set_property PACKAGE_PIN AA18 [get_ports {SDI3}];   #TE0701-06 J5:4         with TEC0053 J9:4
```

```
#GND
#TE0701-06 J5:5         with TEC0053 J9:5
#3,3V
#TE0701-06 J5:6         with TEC0053 J9:6
#All signals connected by the first PMOD cable cable belong to TE0720 Zynq Bank 33.


# Second PMOD cable 12-pin:
# Connections of                                                                    #TE0701-06
J6                 with TEC0053 J8
set_property PACKAGE_PIN Y8 [get_ports {GL[0]}];         #TE0701-06 J6:7         with TEC0053 J8:7
set_property PACKAGE_PIN Y9 [get_ports {GL[1]}];         #TE0701-06 J6:8         with TEC0053 J8:8
set_property PACKAGE_PIN V9 [get_ports {GL[2]}];         #TE0701-06 J6:9         with TEC0053 J8:9
#GND
#TE0701-06 J6:11         with TEC0053 J8:11
#3,3V
#TE0701-06 J6:12         with TEC0053 J8:12
set_property PACKAGE_PIN AA7 [get_ports {GH[0]}];         #TE0701-06 J6:1                with TEC0053
J8:1
set_property PACKAGE_PIN AA6 [get_ports {GH[1]}];         #TE0701-06 J6:2         with TEC0053 J8:2
set_property PACKAGE_PIN U11 [get_ports {GH[2]}];         #TE0701-06 J6:3         with TEC0053 J8:3
#GND
#TE0701-06 J6:5         with TEC0053 J8:5
#3,3V
#TE0701-06 J6:6         with TEC0053 J8:6


# Second PMOD Cable 12-pin contains these two wires unconnected to the SDSoC design:
#set_property PACKAGE_PIN V10 [get_ports {gpio_0_tri_io[0]}];         #TE0701-06 J6:10 - TEC0053 J8:10
#set_property PACKAGE_PIN U12 [get_ports {gpio_0_tri_io[1]}];         #TE0701-06 J6:4 - TEC0053 J8:4
#All signals connected by the second Pmod cable belong to TE0720 Zynq Bank 13.
```

Older Carrier Board Revisions

The **older TE0701-04 or TE0701-05** carrier boards can be used with the identical platform, but
the setup for generation of 3.3V on the PMOD J5 and PMOD J6 is different. There is no S4 switch and the FMC_VADJ = 3.3 V needs to be set by switch S3. Set switch S3 of the TE0701-04 or TE0701-05 carrier board to: S3_1 OFF; S3_2 ON; S3_3 ON; S3_4 OFF
Arrange jumpers J17 and J21:
J17: [1,2,3] connect 1-2
J21: [1,2,3] connect 2-3

Fixed 3.3V will go to PMOD J5. FMC_VADJ will provide 3.3V to PMOD J6Before connecting to TEC0053-04 by PMOD 12pin cables, power on (12V) the TE0701-04 or the TE0701-05 carrier board and measure presence of the 3.3V voltage on the TE0701-06 PMOD J5 pin 12 on pin 6 and presence of the 3.3V voltage coming from the adjustable FMC_VADJ on the TE0701-04 or TE0701-05 PMOD J6 pin 12 and pin 6.

Design Setup

Create SDSoC Platform from TE Reference Design

1. Unzip Reference Design

IMPORTANT

Do not change base folder name after extraction! The name must be:
<install_path>\TE0701_zsys_SDSoC_EDDP_FOC\zsys\

2. 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.

3. 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).

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

5. 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 16.04, with installed Petalinux 2017.1 SDK.

IMPORTANT

Before petalinux project can be built, the executable rights must be set for these 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 permanently modified in this step by copying constrain files locally to project.

> **NOTE**
>
> If needed, recreate original project with batch file (step 4) to restore original Vivado project with externally linked constrains.

e. Wait for project creation:
f. SDSoC Platform is created in X:\zsys\SDSoC_PFM\<TE::SHORTDIR>\zsys
    i. Copy
    X:\zsys_use_virtual_drive.cmd
    to
    X: \SDSoC_PFM\<TE::SHORTDIR>\zsys
    ii. 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
    iii. Copy directory with all files from:
    X:\zsys\misc\src\
    to
    X: \SDSoC_PFM\<TE::SHORTDIR>\zsys\src\
    iv. 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\
g. Close current Vivado project
h. Clear working project files by script
X:\zsys\design_clear_design_folders.cmd
i. 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
j. Compile support libraries serving for connection to 64bit AXI I/O.
k. Open the SDx Terminal 2017.1
    i. CD to: <install_path>\TE0701_zsys_SDSoC_EDDP_FOC\SDSoC_PFM\<TE::SHORTDIR>\zsys\src\
    ii. In the SDx Terminal 2017.1, run batch file:
    build_linux.bat
    iii. Library for the SDSoC Linux target is created:
    <install_path>\TE0701_zsys_SDSoC_EDDP_FOC\SDSoC_PFM\<TE::SHORTDIR>\zsys\src\libte0720_foc.a
    iv. Move the created library libte0720_foc.a to
    <install_path>\TE0701_zsys_SDSoC_EDDP_FOC\SDSoC_PFM\<TE::SHORTDIR>\zsys\sw\aarch32-linux\lib\libte0720_foc.a
    v. Delete the created _sds directory
    <install_path>\TE0701_zsys_SDSoC_EDDP_FOC\SDSoC_PFM\<TE::SHORTDIR>\zsys\src_sds
    vi. In the SDx Terminal 2017.1, run batch file:
    build_standalone.bat
    vii. Library for the SDSoC standalone target is created:
    <install_path>\TE0701_zsys_SDSoC_EDDP_FOC\SDSoC_PFM\<TE::SHORTDIR>\zsys\src\libte0720_foc.a
    viii. Move the created library libte0720_foc.a to
    <install_path>\TE0701_zsys_SDSoC_EDDP_FOC\SDSoC_PFM\<TE::SHORTDIR>\zsys\sw\aarch32-none\lib\ libte0720_foc.a
    ix. Delete the created _sds directory
    <install_path>\TE0701_zsys_SDSoC_EDDP_FOC\SDSoC_PFM\<TE::SHORTDIR>\zsys\src_sds
    x. The SDSoC platform for the target \<TE::SHORTDIR> is in
    <install_path>\TE0701_zsys_SDSoC_EDDP_FOC\SDSoC_PFM\<TE::SHORTDIR>
    xi. 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>\TE0701_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 and Build 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 un-selections 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 3 min instead of 20 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 20 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 TE0701 board.
4. Connect serial terminal via the USB cable.
5. Power ON TEC0053-04 - EDPS Power Stage (12V).
6. Power ON TE0701 board (12V).
7. On PC, open serial terminal.
8. Reset TE0701 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 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.proxy279.$Proxy4022#hasContentLevelPermission. Cannot resolve which method to invoke for [null, class java.lang.String, class	**Error rendering macro 'page-info'** Ambiguous method overloading for method jdk.proxy279.$Proxy4022#hasContentLevelPermission. Cannot resolve which method to invoke for [null, class java.lang.String, class com.atlassian.confluence.pages.Page] due to overlapping prototypes	**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,	• update download

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]	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'	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]	
23 Aug 2018	v.17	UTIA	• 2017.1 release
2018-08-15	v.1	**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.	• Initial release

		atlassian.confluence.core. ContentEntityObject] [interface com.atlassian. user.User, class java.lang. String, class com.atlassian. confluence.core. ContentEntityObject]	
	All	**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]	

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]