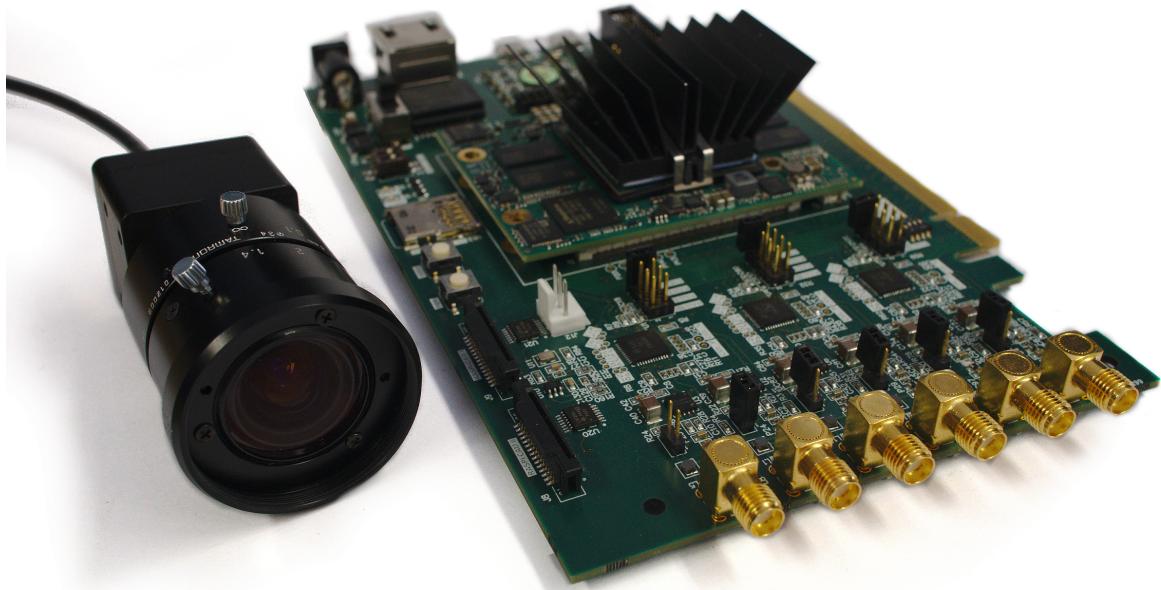


Development of a FPD-Link III - PCIe Baseboard



ZURICH UNIVERSITY OF APPLIED SCIENCES

INSTITUTE OF EMBEDDED SYSTEMS

Author Simone Schwizer

Version 1.0

Last changes October 15, 2019

Copyright Information

This document is the property of the Zurich University of Applied Sciences in Winterthur, Switzerland: All rights reserved. No part of this document may be used or copied in any way without the prior written permission of the Institute.

Contact Information

c/o Inst. of Embedded Systems (InES)
Zürcher Hochschule für Angewandte Wissenschaften
Technikumstrasse 22
CH-8401 Winterthur

Tel.: +41 (0)58 934 75 25

Fax.: +41 (0)58 935 75 25

E-Mail: sczr@zhaw.ch

Homepage: <http://www.ines.zhaw.ch>

Erklärung betreffend das selbständige Verfassen einer Vertiefungsarbeit/Masterarbeit im Departement School of Engineering

Mit der Abgabe dieser **Vertiefungsarbeit/Masterarbeit** versichert der/die Studierende, dass er/sie die Arbeit selbständig und ohne fremde Hilfe verfasst hat.

Der/die unterzeichnende Studierende erklärt, dass alle verwendeten Quellen (auch Internetseiten) im Text oder Anhang korrekt ausgewiesen sind, d.h. dass die **Vertiefungsarbeit/Masterarbeit** keine Plagiate enthält, also keine Teile, die teilweise oder vollständig aus einem fremden Text oder einer fremden Arbeit unter Vorgabe der eigenen Urheberschaft bzw. ohne Quellenangabe übernommen worden sind.

Bei Verfehlungen aller Art treten Paragraph 39 und Paragraph 40 der Rahmenprüfungsordnung für die Bachelor- und Masterstudiengänge an der Zürcher Hochschule für Angewandte Wissenschaften vom 29. Januar 2008 sowie die Bestimmungen der Disziplinarmassnahmen der Hochschulordnung in Kraft.

Ort, Datum:

Unterschrift:

Das Original dieses Formulars ist bei der ZHAW-Version aller abgegebenen **Vertiefungsarbeiten/Masterarbeiten** im Anhang mit Original-Unterschriften und -Datum (keine Kopie) einzufügen.

Abstract

A computer with a GPU combined with an FPGA is a powerful tool for high speed video processing. An FPGA can preprocess multiple video streams in realtime and then send the data to the GPU for further processing. A master thesis called *FPGA-GPU Codesign* has worked out two implementations which optimise the data exchange between an FPGA and a GPU over PCI Express without the CPU being a bottleneck.

FPD-Link III is a cost-effective solution for high speed video transmission. It has made a name for itself for its widespread use in the automotive industry. The transmission can be done over a simple coaxial cable but includes not just a video data stream, but also a bidirectional configuration channel and a power supply for the camera.

The purpose of this thesis is to design hardware which makes it possible to take full advantage of the developed FPAG-GPU codesign and to combine it with an FPD-Link III interface. It is a PCI Express baseboard with an FPGA that can have up to 6 FPD-Link III cameras connected. The FPGA is embedded in a system-on-chip and could potentially also be used stand-alone. As a further video source option, it also includes two connectors for MIPI CSI cameras. These are designed to be compatible with RaspberryPi cameras.

Contents

1. Introduction	2
1.1. Motivation	2
1.2. Goal	2
1.3. Background	3
1.3.1. FPGA	3
1.3.2. PCI Express	3
1.3.3. FPD-Link III	3
1.3.4. MIPI CSI-2	4
2. Concept	5
2.1. Video Pipeline Concept	5
2.2. Baseboard concept	5
2.3. User Panel Concept	6
2.4. Scope	6
3. Implementation	7
3.1. Component Selection	8
3.1.1. System-on-Chip FPGA	8
3.1.2. FPD-Link III Deserializer	9
3.1.3. Power Supplies	10
3.1.4. Ethernet	10
3.1.5. JTAG to USB	10
3.1.6. UART to USB	10
3.1.7. Clock	10
3.2. Mercury+ XU8 Module	11
3.2.1. MIPI CSI-2	11
3.2.2. Boot mode	12
3.2.3. Status LED's	12
3.2.4. Reset Buttons	13
3.2.5. User GPIOs	13
3.3. Deserializer DS90UB954-Q1	14
3.3.1. FPD-Link III	14
3.3.2. Power over Coax	14
3.3.3. I2C	15
3.3.4. GPIO	15
4. Verification & Conclusion	16
4.1. Initial Bring-Up	16
4.1.1. Mercury+ XU8	16
4.1.2. PCIe	18
4.1.3. RaspberryPi Camera	19
4.2. Conclusion	21
Directories	22
Bibliography	22
List of Figures	23
List of Tables	I
A. Appendix	II
A.1. Tools	II
A.1.1. Software	II

A.1.2. Hardware	II
A.2. Scripts	IV
A.2.1. Rapsberry Pi Camera	IV
A.3. Baseboard Schematics	V
A.4. User Panel Schematics	XXIV

1. Introduction

1.1. Motivation

In a standard computer, PCIe (PCI Express) offers the possibility for two devices to exchange data on up to 16 high speed data lanes. The CPU is master of the PCIe interface and therefore usually initiates data transfers. This causes overhead and limits the maximum transfer speed for certain applications. The master thesis *FPGA-GPU Codesign* implements solutions to transfer data directly from an FPGA over PCIe to a GPU without the CPU being a bottleneck for the data throughput. This approach can be especially useful for applications with high resolution video streams which need to be processed in real time. Live video streams from multiple cameras can be preprocessed in the FPGA and then be transmitted via PCIe to the GPU for further processing. This thesis is about developing hardware suitable to take full advantage of this idea.

The vision is to be able to connect multiple cameras to a computer via a PCIe baseboard with an FPGA. The camera interface chosen for this baseboard is FPD-Link III.

FPD-Link III is a cost-effective solution for high speed video transmission. It has gained relevance in automotive applications. Cameras in cars are becoming more common. Nowadays even low cost cars come with a rear camera to assist while parking. FPD-Link III can also be used in different industrial applications, especially for real time use-cases which require high-bandwidth transmissions.

1.2. Goal

The goal of this thesis is to develop a Baseboard that makes it possible to connect multiple FPD-Link III cameras to a standard computer with high data rates. The video data from the cameras should be preprocessed in the FPGA and then be forwarded to the computer via PCIe. Cameras also require to be configured. This configuration will be handled in the FPGA over FPD-Link III.

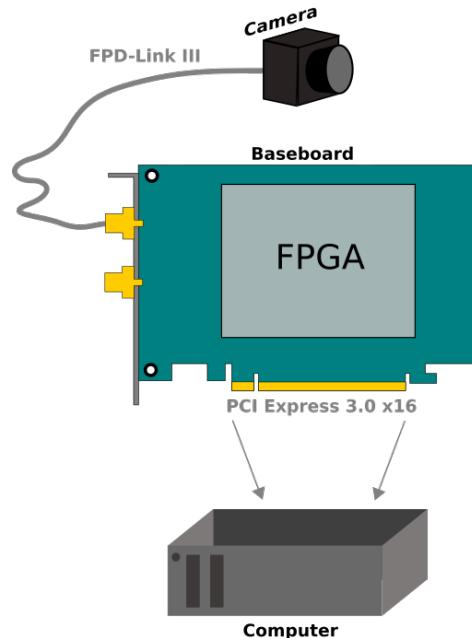


Figure 1.1.: Concept

1.3. Background

This section gives a description of the fundamental terms and concepts this thesis focuses on.

1.3.1. FPGA

FPGA stands for Field Programmable Gate Array. It is an integrated circuit (IC) made up of programmable logic blocks which can be programmed to perform complex combinatorial functions. This makes the execution of the functions much faster than if it were implemented in software. FPGA's have also been combined with embedded microprocessors. Together they form a *system on a programmable chip* (SoC).^[1]

1.3.2. PCI Express

Peripheral Component Interconnect Express (PCIe) is a high-speed serial expansion bus standard found in computers. The devices on the PCIe bus communicate via a link, which is a point-to-point communication channel. A link can consist of up to 16 data lanes. It is also capable of different transfer rates. One lane consists of two differential signaling pairs, one for receiving and one for transmitting. The PCIe interface also provides power supplies of 12 and 3.3 volts for a PCIe card.^[2]

1.3.3. FPD-Link III

Flat panel display link III (FPD-Link III) can be used to receive data from a camera or to send data to a display. The well known standards for high speed video transmission on the consumer market are HDMI, DisplayPort and USB. However these cables are expensive and better suited for short distances. FPD-Link III can be used with coaxial or a shielded twisted-pair (STP) cables. A 15m coaxial cable supports data rates up to 6 Gbps and a 10m long STQ (shielded twisted quad) cable supports up to about 5 Gbps. But FPD-Link III does not only transmit video data. In addition to the video channel, there is also a bidirectional control channel. This is needed for a processor to configure the camera sensor. In case of a display, the control channel can for example be used to send commands from the touchscreen to the processor. The video channel occupies the frequency range between 70MHz and 700MHz whereas the control channel lies between 1MHz and 5MHz.^[3] [4]

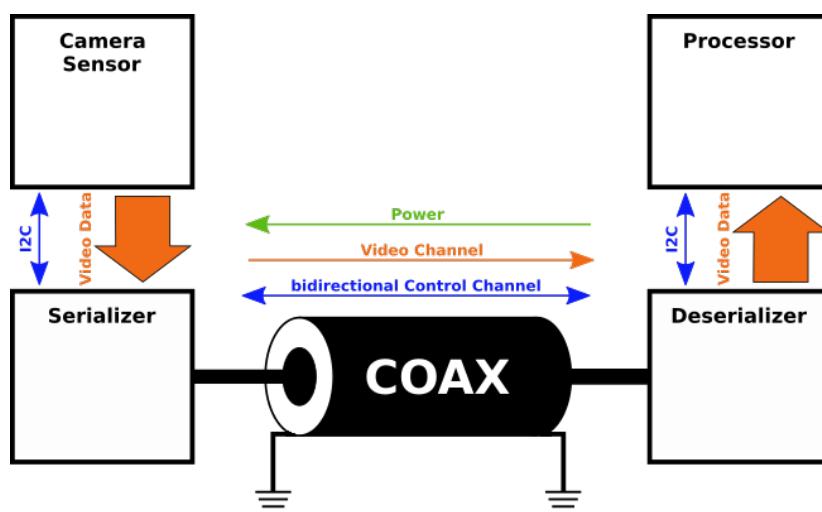


Figure 1.2.: Camera Sensor with FPD-Link III over a Coaxial Cable

An additional feature of FPD-Link III is *Power over coax* which offers the possibility to power the image sensor over the coax cable. This eliminates the need for a further cable for power supply.

1.3.4. MIPI CSI-2

The Mobile Industry Processor Interface Alliance (MIPI Alliance) has defined the transmission protocol MIPI CSI-2. CSI stands for Camera Serial Interface and is a high performance serial interface for camera sensors. It is the leading protocol in the mobile industry. In contrast to FPD-Link III, MIPI CSI-2 requires multiple signals to be transmitted over a cable and is designed for short distances. It can be used for high performance applications including 1080p, 4K, 8K resolutions and more. The number of lanes to be used for data is scalable. The latest release has two physical layers to choose from: MIPI D-PHY and MIPI C-PHY. MIPI D-PHY has a differential pair signal for the clock and differential signal pairs as data lanes. The number of data lanes is scalable. MIPI C-PHY uses three-wire data lanes which have the clock embedded into them. The figure shows the two different physical layers.^[5]

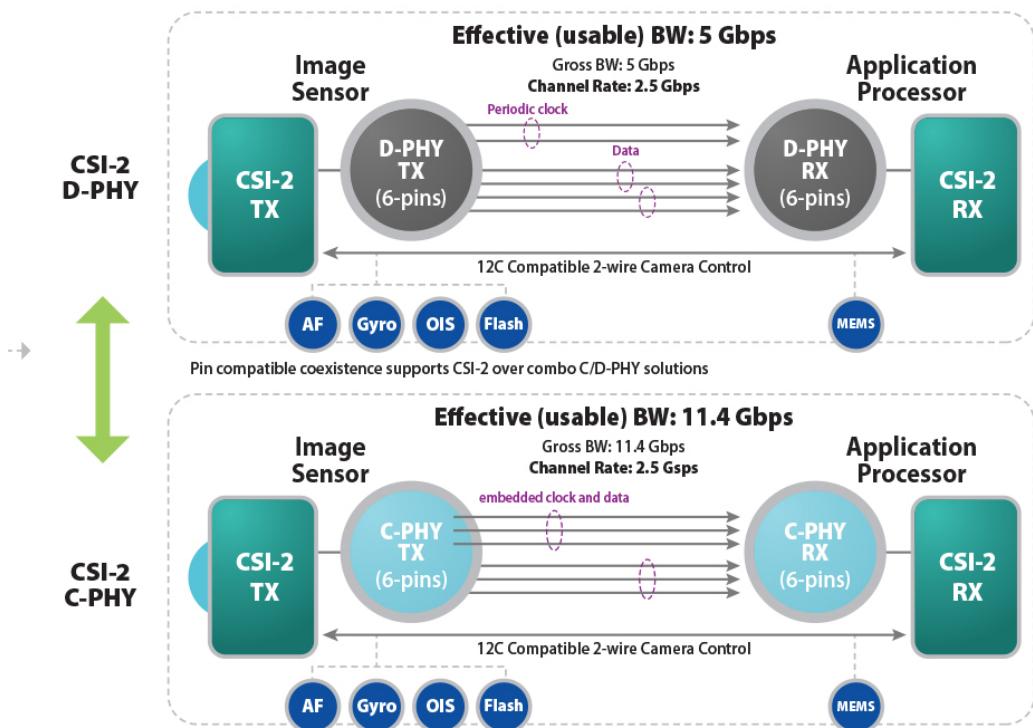


Figure 1.3.: MIPI CSI-2 Protocol & Physical Layer Visualisation^[5]

2. Concept

2.1. Video Pipeline Concept

As stated in the introduction, the main goal is to design hardware which makes it possible to connect multiple FPD-Link III cameras to a standard PC over PCIe. As shown in figure 2.1, the video is transmitted from a camera sensor to a serializer which sends the video over FPD-Link III in a coaxial cable to the deserializer. The deserializer transmit the data to the FPGA over MIPI CSI-2 D-PHY. In the FPGA, the data can be preprocessed and then be sent over PCIe to the memory on the computer. There will be multiple deserializers on the baseboard so multiple cameras can be connected.

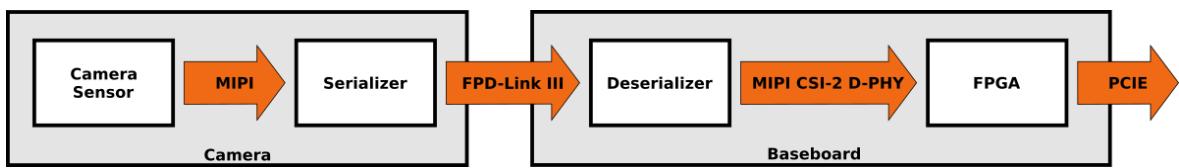


Figure 2.1.: Video Pipeline

2.2. Baseboard concept

The following figure shows a sketch of the concept for the baseboard. The path of the video data is colored in bright green. On the left side there are six coax connectors for the FPD-Link III interface (number of coax connectors is limited to six because of the defined maximum width of a PCIe card). The data goes through the deserializer to the FPGA. From the FPGA the data is transmitted to a computer over PCIe. In addition to the FPD-Link III connectors there are two MIPI CSI-2 D-PHY connectors which are compatable with the RaspberryPi-Camera. This gives the user an additional option for a video source.

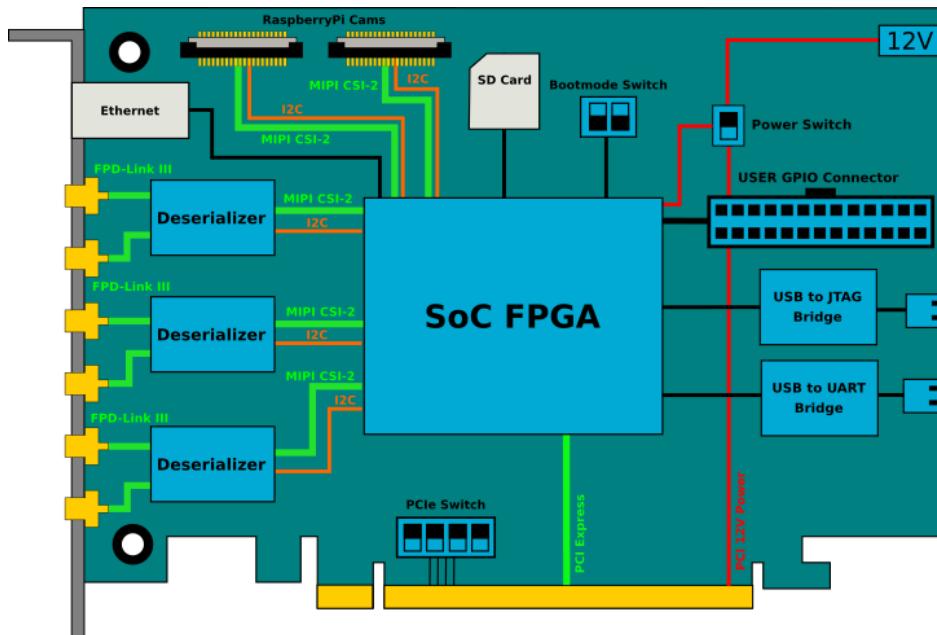


Figure 2.2.: Concept Sketch

The main focus of the project is the deserializer for FPD-Link III, the integration of a SoC FPGA and the PCIe interface. The company Enclustra offers a selection of SoC modules which entail a Xilinx MPSoC (Multiprocessor system-on-chip), SDRAM, flash memory and more. The module can be mounted to the baseboard via connectors. Figure 2.2 shows additional hardware and interfaces which are needed for an operating baseboard. The SoC needs a JTAG interface for programming and debugging. An SD-card slot is added and can be used as the boot device. The boot mode switch can be used to change the source device for the boot process. The SoC can be reset over a button and a status LED gives further information about the state of the SoC. The UART (Universal Asynchronous Receiver Transmitter) is needed for the console output of the processor. An ethernet connector makes it possible to get access to the processor with SSH (Secure Shell).

A power switch makes it possible to choose between an external power supply or the 12V supplied over the PCIe interface from the computer. The external supply is needed when the SoC should be programmed before the computer is booted. The connection to PCIe devices are established while booting in the bios, which means the FPGA should be programmed before the computer turns on. The external supply is also useful when the board consumes more power than the computer can supply.

PCIe can be used in four different lane configurations: 1-, 4-, 8- or 16-lanes. The PCIe switch is used to choose between these options.

2.3. User Panel Concept

When the baseboard is mounted on the PCIe connector inside a computer, it becomes quite difficult to interact with the baseboard. To make the development on the baseboard easier, a panel is planned, which will have push buttons, dipswitches, LED's and a reset button. This panel can then be mounted to the front of the computer and connected to the baseboard with a ribbon cable. Figure 2.3 shows how the front and back are planned to look like.

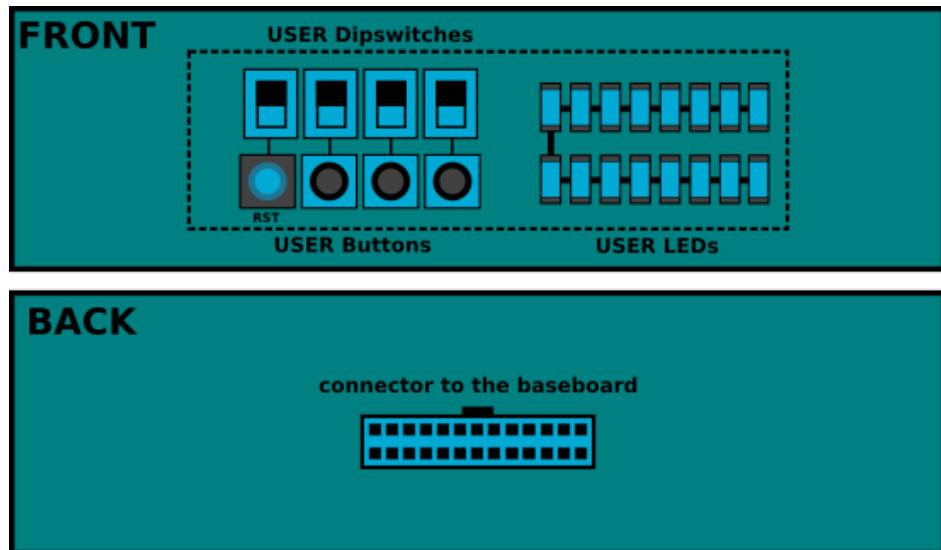


Figure 2.3.: User Panel

2.4. Scope

The scope of this thesis includes the component selection, designing the schematics of the baseboard and user panel. It also includes an initial bring-up of the baseboard.

3. Implementation

This chapter discusses the different components of the baseboard. This includes component selection, configurations and descriptions of the used parts. Figure 3.1 shows the manufactured and assembled baseboard. The three white connectors in the middle of the baseboard are for the SoC module from Enclustra.

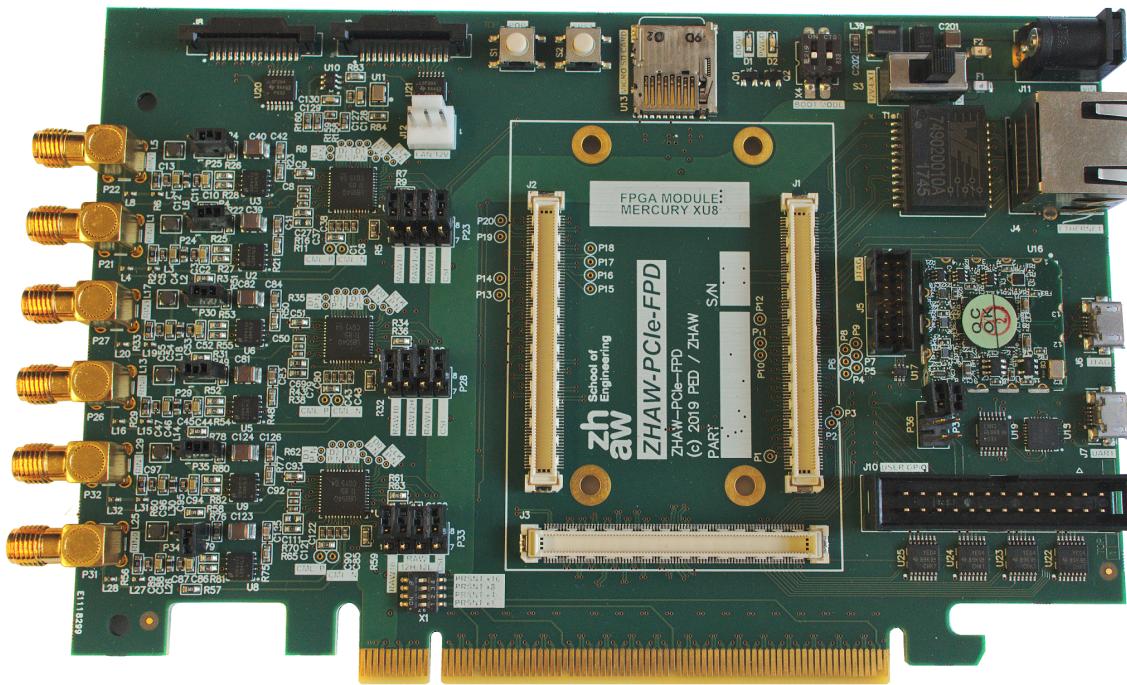


Figure 3.1.: FPD-Link III PCIe Baseboard

The details given should allow any person continuing the work with this hardware to understand the decision processes and quickly find the most important entry points into additional materials.

3.1. Component Selection

3.1.1. System-on-Chip FPGA

The main requirements for the SoC FPGA are the following:

- 16 PCIe Gen3 lanes

The total number of possible lanes is chosen to maximise the possible data throughput from the FPGA to the computer. PCI Express version 3 with 16 lanes has a maximum data rate of 15.75 GB/s (without protocol overhead)^[2].

- HP IO signals

The video data from the deserializer and from the RaspberryPi camera are sent over MIPI D-PHY CSI-2. The MIPI_DPHY_DCI I/O standard is only supported in HP I/O banks in the UltraScale+ FPGA architecture^[6]. One full HP I/O bank with 48 differential signals (24 differential pairs) are sufficient for three deserializers (3x5 differential pairs) and two RaspberryPi cameras (2x3 differential pairs).

- Speed grade 2

The deserializer from TI supports CSI-2 line rates of 400 Mbps/800 Mbps/1.5 Gbps/1.6 Gbps. Speed grade 1 allows for a maximum MIPI D-PHY receiver speed of 1.26 Gbps whereas speed grade 2 has a maximum speed of 1.5 Gbps^[7].

The Enclustra modules are available with different MPSoC options. Table 3.1 shows the comparison between the Ultrascale+ MPSoC's. The dark red fields highlight the exclusion criteria.

Ultrascale + MPSoCs	Module Name	PCIe Gen3x?	HD IO Banks	HP IO Banks	Speedgrade
ZU4CG	Mercury+ XU8	16	2	3	1
	Mercury+ XU9	16	2	3	1
ZU11EG	Andromeda XZU80 (in development)				
	Andromeda XZU80 (in development)				
ZU19EG					
ZU4EV	Mercury XU5	4/0			
	Mercury+ XU6	4			
ZU5EV	Mercury+ XU8	16	2	3	1
	Mercury+ XU9 (in development)				
ZU7EV	Mercury+ XU6	4			
	Mercury XU5	4/0			
ZU7EV	Mercury+ XU8	16	2	3	2
	Mercury+ XU9 (in development)				

Table 3.1.: Ultrascale+ Module Comparison

The Enclustra module **Mercury+ XU8** with the Xilinx Ultrascale+ MPSoC **ZU7EV** (XCZU7EV-2FBVB900I) meets all requirements and was chosen for this baseboard.

3.1.2. FPD-Link III Deserializer

A deserializer is needed to convert the FPD-Link III signal to MIPI CSI-2 which can be connected to the FPGA. Texas Instruments (TI) offers a variety of solutions for FPD-Link III serializers and deserializers. The requirements for choosing a deserializer are the following:

- Input: FPD-Link III LVDS
- Output: MIPI CSI-2
- Able to connect to 2+MP (mega pixel) cameras

TI provides two deserializers which meet the given requirements. These are DS90UB960-Q1 (960) and DS90UB954-Q1 (954). They can be found in Figure 3.2.

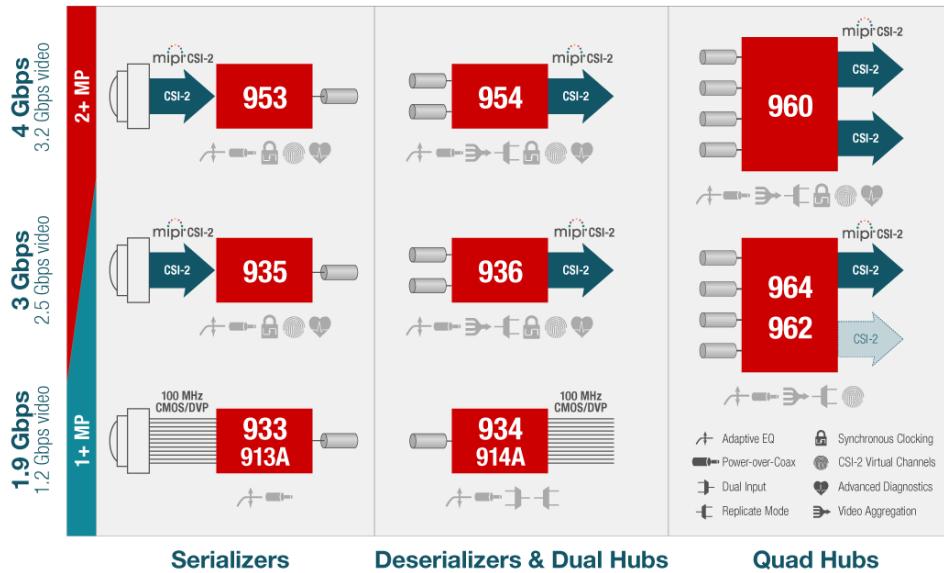


Figure 3.2.: TI Portfolio for FPD-Link III cameras^[8]

960 and 954 have the same maximum data rates for FPD-Link III and MIPI CSI-2. However 954 has more GPIO pins per camera. 954 has 7 GPIOs for 2 cameras and 960 has 8 GPIOs for 4 cameras. GPIO signals are useful to get diagnostics of the deserializer but they can also be used to connect directly to the camera sensor board. Some cameras need for example an enable signal which can be set by the processor over these GPIOs. For more information about GPIOs see section 3.3.4. The deserializer chosen for this baseboard is the **DS90UB954-Q1**.

3.1.3. Power Supplies

The main power supply of the baseboard is 12V (volts) and will be supplied from an external supply or by the PCIe connector. A power supply on the SoC module provides 1.8V, 2.5V and 3.3V which can be used to power components on the baseboard. The additional supplies which are needed on the board are 1.2V for an I/O bank of the module and 9V for the power over coax.

9V power supply for PoC

The power supply chosen is the LM2941LD Low Dropout Adjustable Regulator with a maximum current of 1 ampere. The input voltage range is between 6V and 26V. The output is adjustable between 5V and 20V. The supply is recommended from TI in their reference design for the deserializer^[9].

1.2V power supply for FPGA I/O Bank

The FPGA pins which will be used to connect to the MIPI CSI-2 D-PHY signals, must use the I/O standard MIPI_DPHY_DCI. This I/O standard requires the FPGA bank voltage of these pins to be 1.2 volts (see Table 1-77 in Xilinx document UG571^[10]). The power supply chosen is the TPS73601DBVR Cap-Free, NMOS, 400-mA Low-Dropout Regulator with Reverse Current Protection. The input voltage needs to be between 1.7V and 5.5V. It will be powered by the 3.3V supplied by the module. The regulator can have a fixed output voltage of 1.2V to 5V. It is used in noise-sensitive circuits and is able to maintain the constant voltage even during load changes.

3.1.4. Ethernet

The Ethernet signal of the SoC processor is routed to a LAN transformer before going out via an ethernet connector. The chosen transformer is 749020010A from Wurth Electronics.

3.1.5. JTAG to USB

The JTAG signal can either be accessed with the Xilinx JTAG connector over a Molex connector or by a micro USB. The signal to the Xilinx connector does not need to be converted. The JTAG USB connection needs an adapter. Digilent provides a Programming Module for Xilinx FPGAs. The mounted module is called JTAG-SMT2-NC and can be mounted onto a PCB.

3.1.6. UART to USB

Silicon Labs manufactures a variety of different USB to UART bridges. They offer USB to Quad/Dual UART bridges. It is planned, for the SoC to only use one UART interface. The chosen bridge is the CP2110-F02-GM1 HID USB to UART bridge. The standard USB to UART bridges CP2101, CP2102 & CP2103 are no longer recommended by Silicon Labs for new designs.

3.1.7. Clock

The clocking on the SoC module is already handled on the module itself. The only other IC which requires a clock is the FPD-Link III deserializer. The deserializer requires a reference clock between 23 MHz and 26 MHz. The chosen crystal SG-210STF 25.0000ML3 from Epson has a frequency of 25 MHz. The clock is recommended by TI in the reference design for the deserializer^[9]. Because there are three deserializers, a fan out buffer IC is used which produces three clock outputs with the same frequency. This isolates the clock source from the loads. The chosen fan out buffer is the PL133-37TC from Micrel.

3.2. Mercury+ XU8 Module

This section describes the connections to the Mercury+ XU8 module. The first part goes into the pin assignments for the MIPI signals. Then follows the boot mode selection, the signals used for the status LED's and the two reset buttons.

3.2.1. MIPI CSI-2

The deserializer and the RaspberryPi cameras send their data over MIPI CSI-2 DPHY. These signals need to be connected to FPGA pins with the I/O standard MIPI_DPHY_DCI. Only FPGA pins on an HP (high performance) I/O bank and with a bank voltage of 1.2V can be used with this standard. The baseboard has three deserializers. Each of them has one differential clock lane and 4 differential data lanes. The two RaspberryPi cameras each use one differential clock lane and 2 differential data lanes. In total this results in 5 differential clock lanes and 16 differential data lanes. On the Mercury+ XU8, the HP I/O bank 66 has 48 differential signal pairs. However the signals can not be connected arbitrarily. The following figure 3.3 from Xilinx shows how a bank is split into different bytes and nibbles. The yellow column shows the pins that can be used as clock pins. [6] [10] [11]

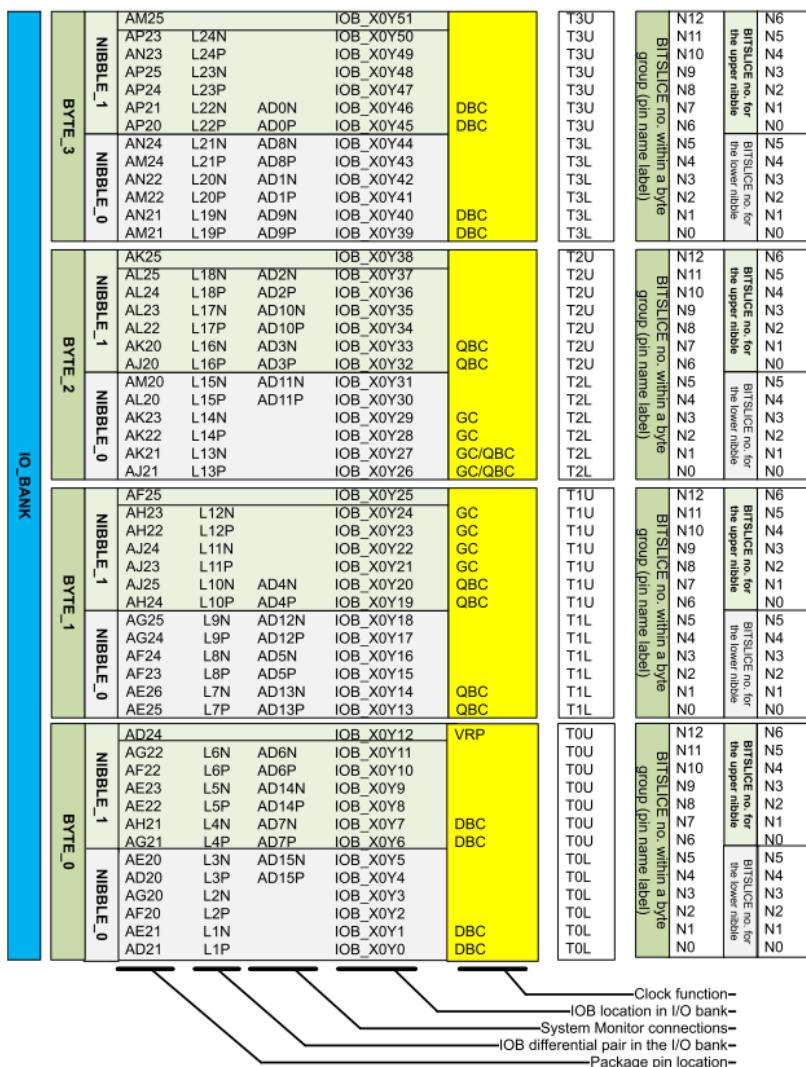


Figure 3.3.: Example I/O Bank (Bank 44 of XCKU040FFVA1156) [10]

DBC stands for dedicated byte clock and QBC for quad byte clocks. The differential data lanes for a DBC need to be inside the same byte of the bank. The QBC clock can be used with data lanes in all four bytes of the bank. It is recommended to use adjacent I/O pins for a clock and the respective data lanes. If they are not chosen continuously, additional I/O pins will automatically be used by a MIPI receiver block in the FPGA for clock/strobe propagation. The SoC on the Mercury+ XU8 is the XCZU7EV-FBVB900. The package text file from Xilinx shows which pins have DBC or QBC capabilities^[12].

Additional note: The MIPI_DPHY_DCI I/O standard requires a 240 ohm external pull down resistor on the VRP pin of the corresponding FPGA bank. This resistor is already present on the Mercury+ XU8 and was therefore not needed on the baseboard.

3.2.2. Boot mode

The Mercury+ XU8 module offers three different options to boot from:

- eMMC flash (1: ON, 2: ON)
- QSPI flash (1: ON, 2: OFF)
- SD card (1: OFF, 2: OFF)

The eMMC and QSPI flash are both located on the module. A boot mode switch was added to be able to choose between these options. Figure 3.4 shows the location of the switch. The configuration for each option can be seen in the brackets in the list above.

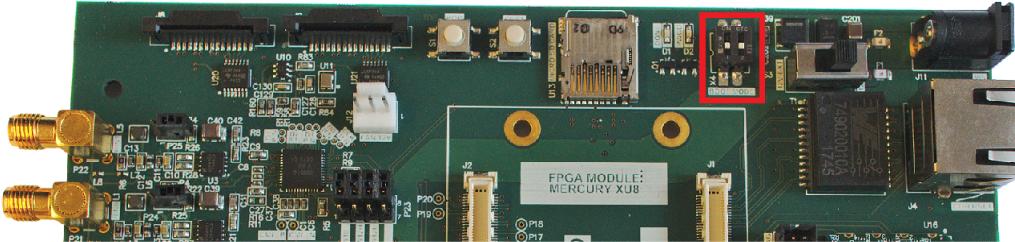


Figure 3.4.: Boot Mode Switch

3.2.3. Status LED's

There are two status LED's. The LED called PWGD (power good) turns on if all the power supplies on the module are within their range. DONE indicates if the FPGA configuration is done. Figure 3.5 shows the location of the status LED's.

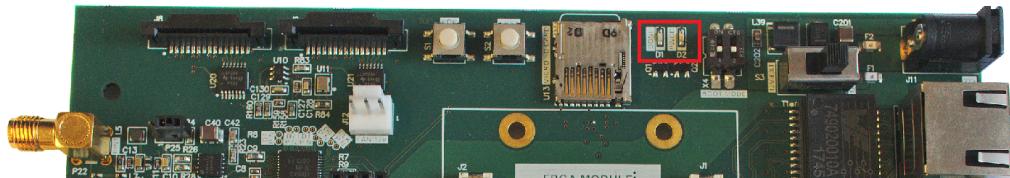


Figure 3.5.: Status LED's

3.2.4. Reset Buttons

There are two options to reset the Mercury+ XU8 module:

- SRST soft-reset: resets the MPSoC device
- POR power-on reset: resets the MPSoC device, USB PHYs, QSPI flash & eMMC flash

Figure 3.6 indicates the location of the two reset buttons.

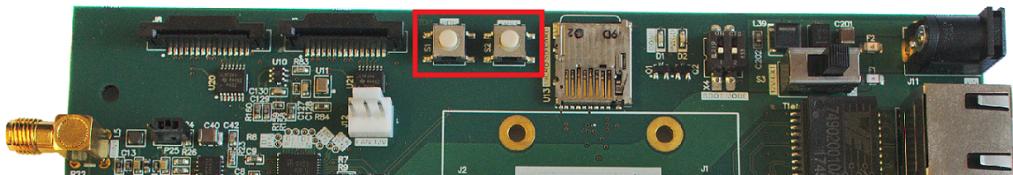


Figure 3.6.: Reset Buttons

The reset signals can also be used from the user panel. To get the POR on the user panel, connect pin 1 and 2 on connector P36 with a jumper. To get SRST on the user panel, connect pin 1 and 2 on P37. If the resets are not needed on the user panel, the jumpers can be plugged onto pin 2 and 3. This will turn the buttons on the user panel into user GPIOs which are connected to the FPGA. Figure 3.7 shows P36 and P37.



Figure 3.7.: Connect Reset Buttons to User Panel (P36, P37)

3.2.5. User GPIOs

The connector for the user panel (see figure 3.8) includes signals for 16 LEDs, 4 dip switches and 4 push buttons (2 of which can be used as resets).

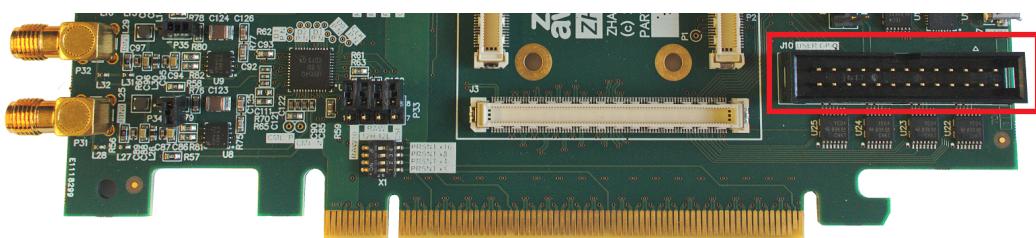


Figure 3.8.: User GPIOs Connector for User Panel

3.3. Deserializer DS90UB954-Q1

This chapter deals with the design of the deserializer. The sections about I2C and GPIO do not mainly concentrate on the design part. They give an inside on application specific implementation details and limitations. The datasheet^[13] and the reference design^[9] of the DS90UB954-Q1 were used as source for this chapter.

3.3.1. FPD-Link III

The DS90UB954-Q1 deserializer can be used with one or two camera sensors over FPD-Link III. It supports 2MP@60fps and 4MP@40fps cameras. The two input channels RIN0/RIN1 can be enabled and disabled through registers of the deserializer (Register: RX_PORT_CTL, 0x0C). The input channels can be used as single ended (coaxial channel) or as differential (STP). This baseboard uses single ended coaxial connections. This means the RN- port of a channel is connected to ground with a 15nF-capacitor and a 50 ohm resistor. The RN+ port is connected to the conductor inside the coax connector with a 33nF capacitor in series. This capacitor blocks the receiver ports of the deserializer from any DC voltage. This is especially important if power over coax is used.

3.3.2. Power over Coax

Power over coax (PoC) makes it possible to use the coaxial cable to supply power to the camera sensor. The output of a power supply is connected to the coaxial connector with a filter in series. This filter is needed to shield the power supply from the AC signal transmitted between deserializer and camera sensor over FPD-Link III.

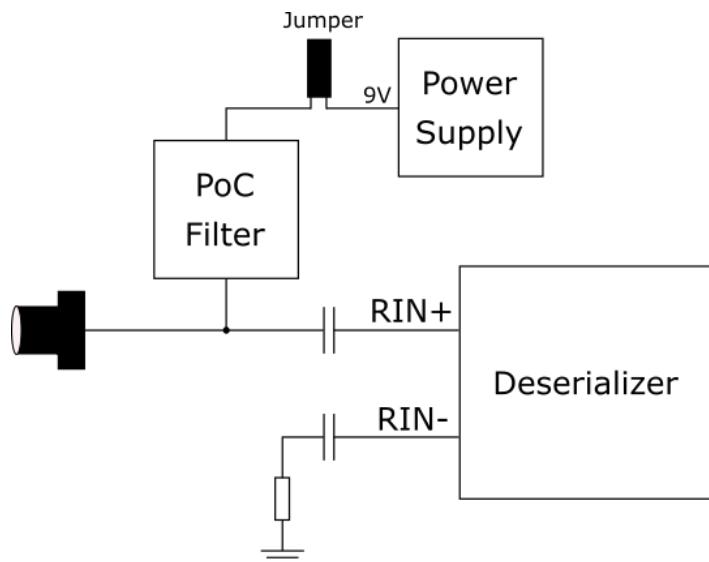


Figure 3.9.: Power over coax

Typically the PoC voltage is between 5V and 36V^[14]. A 9V PoC voltage was chosen for this baseboard. Before connecting a camera to the baseboard it must always be checked what PoC voltage is tolerated by the camera board. If needed, the PoC voltage can be cut off from the coax connector by removing the jumper in figure 3.9. The design for the PoC filter was taken from the reference design provided from TI^[9].

3.3.3. I2C

The deserializer has a I2C interface. This allows setting and reading configurations out of the registers. The I2C address of the deserializer can be set with a voltage divider on the IDX pin. The three deserializers on the baseboard have the 7-bit addresses 0x30, 0x36 and 0x3D.

Forwarding I2C over FPD-Link III

A deserializer also extends the I2C signal over the bidirectional control channel of the FPD-Link III to the camera. But not all I2C messages are forwarded. There is a problem one faces when connecting multiple identical cameras to the baseboard. These cameras will most likely have the same I2C addresses. The solution for this problem is aliases. A deserializer can have up to 8 pairs of aliases defined for each RX port. This pair consists of a SlaveAlias and a SlaveID. The SlaveID is selected equal to the actual I2C address of a remote slave and the SlaveAlias can be chosen freely. The host can then send an I2C command addressed to a SlaveAlias and the deserializer will forward the command over FPD-Link III to the SlaveID. One SlaveID can be assigned multiple times to different aliases. A SlaveAlias can also be used for multiple SlaveID's. This makes it possible to broadcast an I2C command to multiple devices at once.

3.3.4. GPIO

A deserializer has 7 general-purpose inputs/outputs (GPIOs). These can be used for monitoring, configuring or be controlled through I2C. A GPIO can be set to forward the FrameValid (FV) or LineValid (LV) signal from the serializer to the host. They can also be configured to send the LOCK or PASS signal (status of FPD link). But they can also be directly linked to the GPIOs of a serializer. Any GPIO port of a serializer can be connected to any GPIO port of the deserializer. These connections can be very crucial if a camera sensor requires for example an enable signal. If a GPIO pin is set as input (back channel GPIO) it can be mapped to multiple GPIOs of any RX port.

4. Verification & Conclusion

This chapter is about the verification of the hardware. The conclusion summarizes the achieved product and discusses what the next steps are for the baseboard.

4.1. Initial Bring-Up

This section describes the initial bring-up of the baseboard. To verify the design, the individual components are tested.

4.1.1. Mercury+ XU8

The following parts were tested to verify that the XU8 module is working properly:

- JTAG
- UART
- Flash and Boot from QSPI
- SD card boot

The SoC on the module is from Xilinx. The development environment for Xilinx MPSoC's are Vivado and Xilinx SDK.

JTAG

There is a JTAG USB and a Xilinx JTAG connector on the baseboard. The baseboard was successfully detected by both JTAG connectors and could also be programmed. Figure 4.1 shows the Vivado Hardware Manager recognising the JTAG over USB (Digilent chip). The right side shows the view after the MPSoC was successfully programmed.

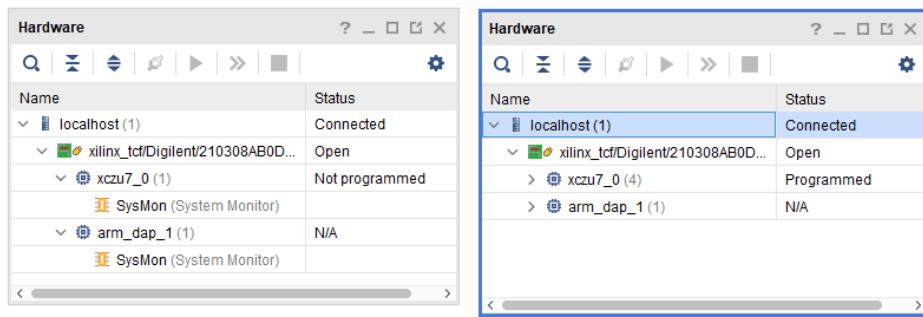


Figure 4.1.: Vivado - Detected MPSoC with JTAG over USB

UART

The HID USB to UART Bridge from Silicon Labs comes with example code and an executable HidUartExample. The executable can be used to connect to the UART bridge and transfer/receive data from the processor. Enclustra offers a reference design for the XU8 module for their baseboard called Mercury PE1. This design was used and modified to fit the baseboard of this thesis. The reference design includes a simple program called HelloWorld, which writes out "Hello World" multiple times over UART and makes the LED on the XU8 blink. The test was successful and the output of the application HelloWorld can be seen in figure 4.2.

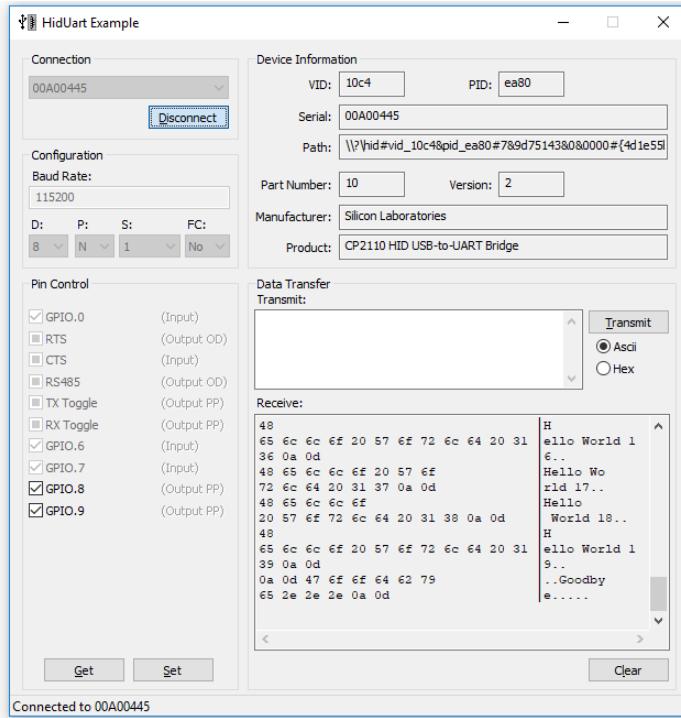


Figure 4.2.: HidUartExample shows output of HelloWorld

Flash and Boot from QSPI

On the XU8 module there is a QSPI flash which can be flashed with the Xilinx SDK. The *Mercury XU8 Reference Design for Mercury PE1 User Manual*^[15] shows a step-by-step guide for this task. After flashing the QSPI flash with the program HelloWorld, the boot mode was set to QSPI (see section 3.2.2 on how to set the boot mode switch). After a power-cycle the module successfully loaded the program from the flash and the LED started blinking.

SD card boot

The SD card boot was tested similarly as the QSPI boot with the guide in the reference design manual^[15]. The program HelloWorld was copied onto the SD card. The boot mode switch was set to SD card and after a power-cycle the LED started blinking as expected.

4.1.2. PCIe

For the PCIe testing, the baseboard was plugged into a *Dell Precision Tower 5810*. The PCIe slot used was the *SLOT4 PCIe3x16 75W +EXT225W*. The computer is running Ubuntu 18.04.2 LTS.

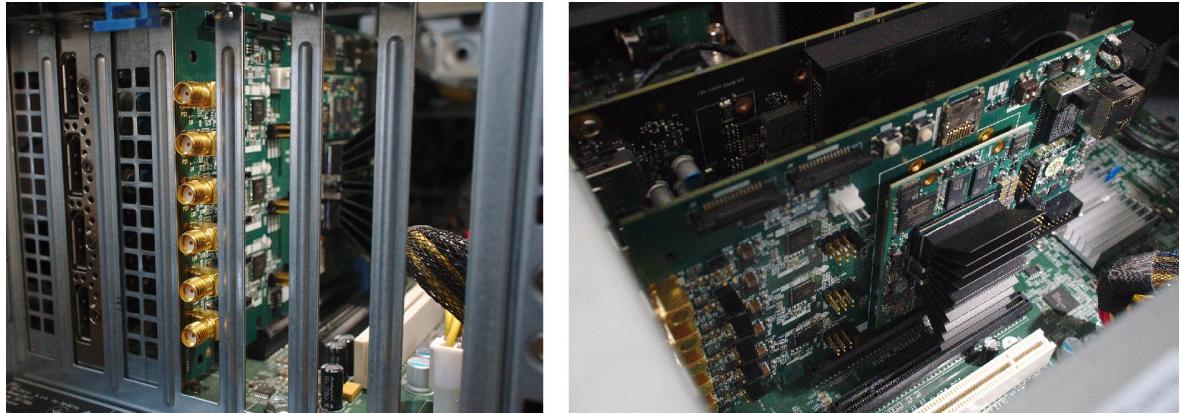


Figure 4.3.: PCIe Test Setup

Figure 4.4 shows the block design used for testing. It can establish a link with a host computer over PCIe. When a link is established the baseboard shows up as a Xilinx device in the console output of lspci. Lspci is a command line tool which can list detailed information about all devices connected to the PCIe bus.

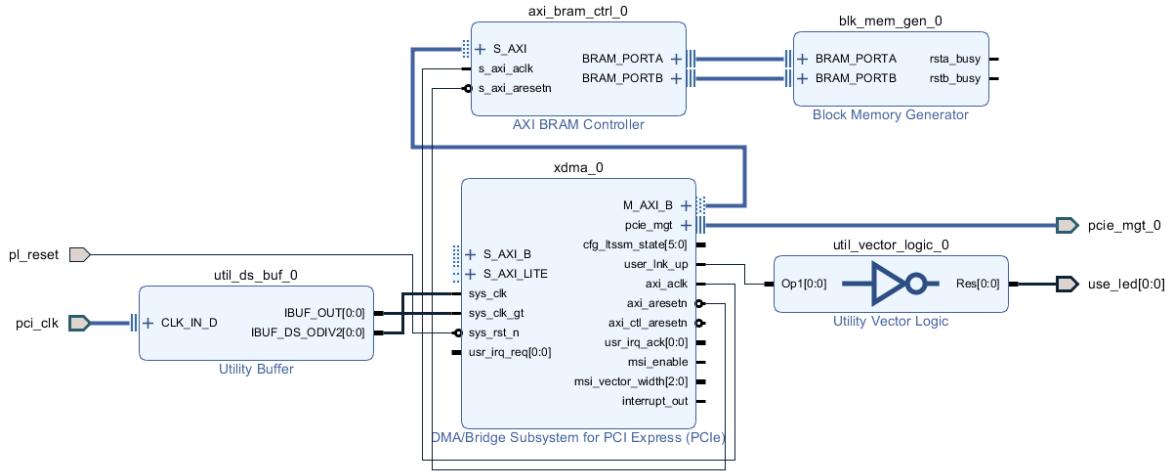


Figure 4.4.: Vivado Block Design with XDMA

The lane width and the maximum link speed can be selected in the settings of the *DMA/Bridge Subsystem for PCI Express* block.

These configurations were tested with the following results:

- line width: 1, maximum speed link: 8 GT/s, Test: successful, link established
- line width: 4, maximum speed link: 8 GT/s, Test: successful, link established
- line width: 8, maximum speed link: 2.5, 5.0 & 8 GT/s, Test: unsuccessful, no link detected
- line width: 16, maximum speed link: 2.5, 5.0 & 8 GT/s, Test: unsuccessful, no link detected

4.1.3. RaspberryPi Camera

The camera used for this test was a RaspberryPi Camera V2.1 with a Sony IMX219 image sensor. A ribbon cable connects the camera with the baseboard.

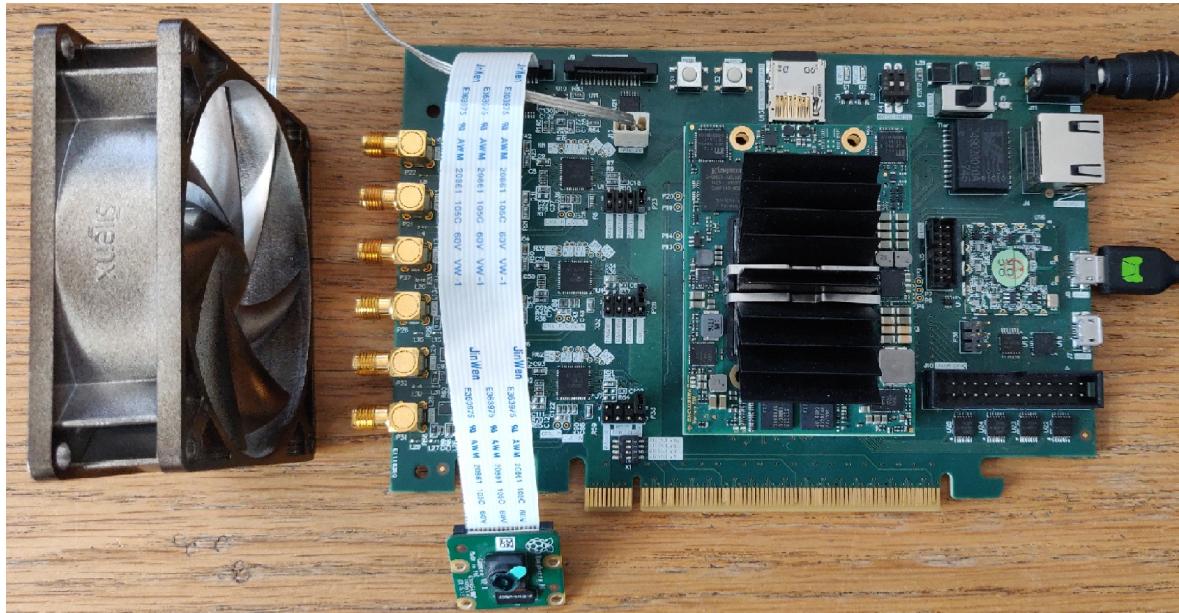


Figure 4.5.: Setup with RaspberryPi Camera

The signals which are routed over the ribbon cable are: MIPI CSI-2 clock lane & two data lanes, i2c and an enable signal. The block design which is loaded onto the FPGA includes the component *MIPI CSI-2 RX Subsystem* which is used to receive the MIPI signal. Figure 4.6 shows the component.

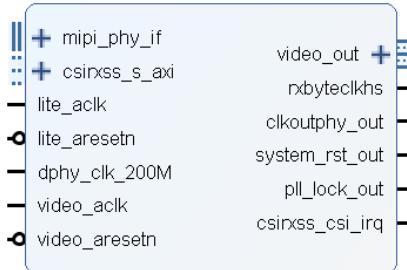


Figure 4.6.: MIPI CSI-2 RX Subsystem

The clock signal and one data signal were measured with the oscilloscope *Waverunner 8404M*. The bring-up of the camera was done in the following order:

- Reset camera: Set enable signal low
- Enable camera: Set enable signal high
- Configure camera over I2C (see script in appendix A.2)
- Test if MIPI data is received from FPGA

After configuration, the camera starts sending data over the MIPI interface. The measurement by the oscilloscope of the clock and a data line can be seen in figure 4.7 and 4.8.

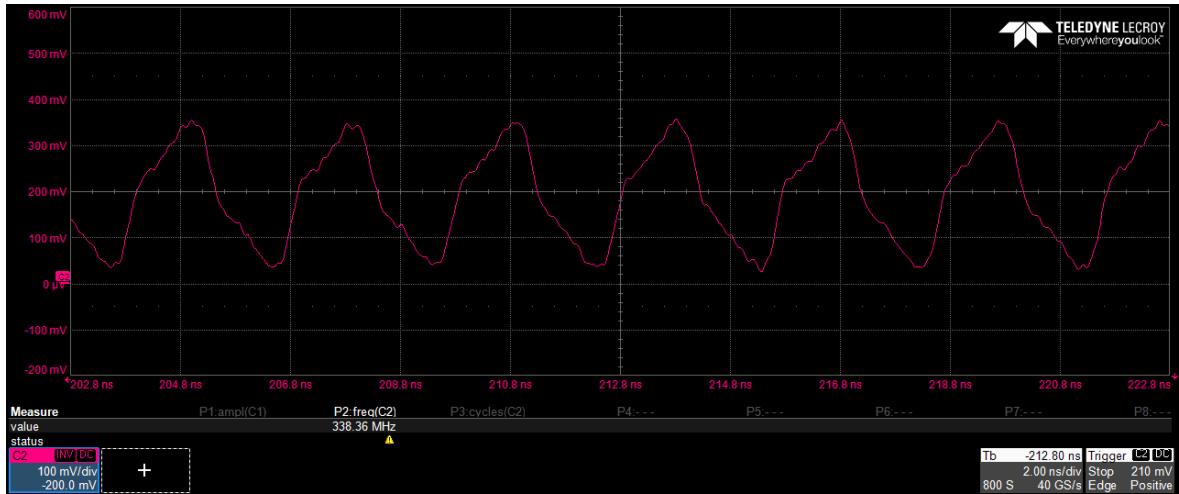


Figure 4.7.: Measured clock signal

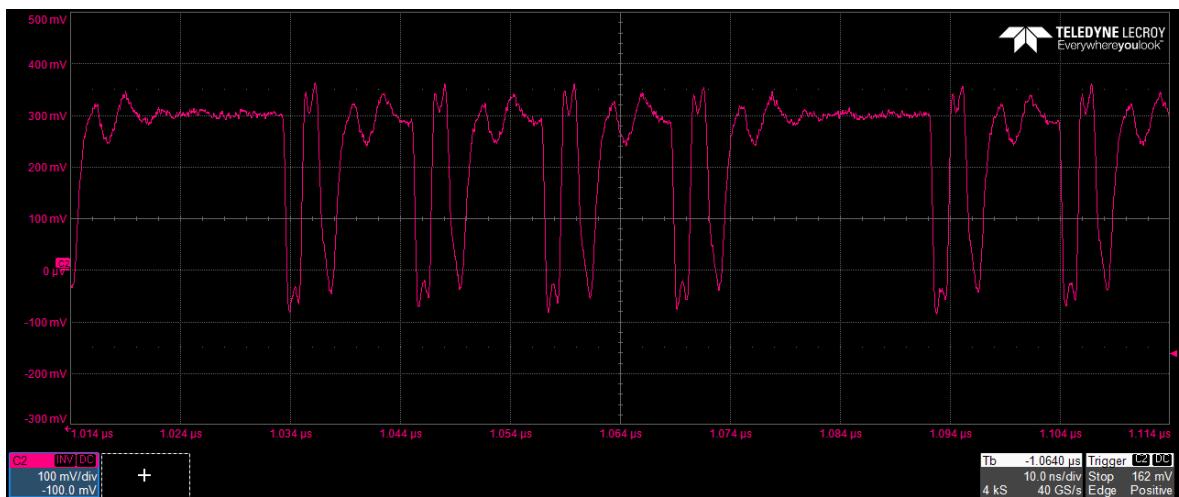


Figure 4.8.: Measured data lane 1

The data signal looks noisy. The MIPI RX subsystem recognizes this as data and produces a video output signal. However the video output signal transmits a constant output value which could mean that the camera is not correctly configured yet or the FPGA cannot read the MIPI data properly. The noisy data signal could be caused by a not perfectly tuned configuration of the MIPI RX subsystem and the RaspberryPi camera.

4.2. Conclusion

The FPD-Link III PCIe baseboard was successfully designed and produced. Parts of the baseboard are tested and verify that the baseboard as such is functional. The components for the MPSoC are working properly and the baseboard was successfully detected over PCIe. Some tests showed that there is still work to do in the bring-up of the baseboard.

A task that is still open is the debugging of the PCIe to get a working link with lane width of 8 and 16. The link to a PCIe device is established during the boot of the BIOS (basic input/output system) which is demanding to debug.

The RaspberryPi camera can be configured over I2C and it sends data which is recognized by the MIPI receive block in the FPGA. But the data shows that it is not yet sending frames correctly. The camera configurations and the FPGA design need to be revised.

The most important task is setting up the FPD-Link III cameras. This could not be realised in this thesis because of time limitations. This task includes setting the registers of the deserializer correctly and then configuring the serializer and the camera sensor properly over FPD-Link III.

Once all these individual parts are completed, they can be combined into one single system that collects video streams from multiple cameras and transfers the data to a computer via PCIe.

Bibliography

- [1] Wikipedia, The Free Encyclopedia. Field-programmable gate array. [Online]. https://en.wikipedia.org/wiki/Field-programmable_gate_array (Accessed July 24, 2019).
- [2] Wikipedia, The Free Encyclopedia. Pci express. [Online]. https://en.wikipedia.org/wiki/PCI_Express (Accessed July 24, 2019).
- [3] T. K. Chin. What you need to know about high-speed cables for FPD-Link III SerDes. Technical report, Texas Instruments, 2017.
- [4] Mark Sauerwald. FPD-Link III - doing more with less. Technical report, Texas Instruments, 2014.
- [5] mipi alliance. Mipi camera serial interface 2 (mipi csi-2). [Online]. <https://www.mipi.org/specifications/csi-2> (Accessed July 24, 2019).
- [6] Xilinx and Inc. MIPI D-PHY v4.1, LogiCORE IP Product Guide, Vivado Design Suite PG202 (v4.1). 2019. p. 30.
- [7] Xilinx and Inc. Kintex UltraScale+ FPGAs Data Sheet: DC and AC Switching Characteristics. Technical report, Xilinx, 2015. p. 20.
- [8] Texas Instuments. Kamera-SerDes Übersicht. [Online]. <http://www.ti.com/de-de/interface/fpd-link-serdes/camera-serdes/overview.html> (Accessed July 27, 2019).
- [9] DS90UB954-Q1EVM Quick Start. Technical Report August, Texas Instuments, 2017.
- [10] UltraScale Architecture SelectIO Resources, User Guide, UG571. Technical report, Xilinx, 2019. pp 141, 154.
- [11] MIPI CSI-2 Receiver Subsystem v3.0, PG232. Technical report, Xilinx, 2018. p. 32.
- [12] Xilinx. Ascii pinout file. [Online]. <https://www.xilinx.com/support/packagefiles/zupackages/xczu7evfbvb900pkg.txt> (Accessed August 3, 2019).
- [13] DS90UB954-Q1 Dual 4 . 16 Gbps FPD-Link III Deserializer Hub With MIPI CSI-2 Outputs for 2MP / 60fps Cameras and RADAR. Technical report, Texas Instruments, 2018.
- [14] Cole Macias and Mandeep Singh. Power-over-Coax Design Guidelines for DS90UB953-Q1. Technical report, Texas Instruments, 2018. p. 5.
- [15] DDUE. Mercury + XU8 SoC Module, Reference Design for Mercury+ PE1 Base Board User Manual. Technical report, enclustra, 2018. pp 23-29.

List of Figures

1.1. Concept	2
1.2. Camera Sensor with FPD-Link III over a Coaxial Cable	3
1.3. MIPI CSI-2 Protocol & Physical Layer Visualisation ^[5]	4
2.1. Video Pipeline	5
2.2. Concept Sketch	5
2.3. User Panel	6
3.1. FPD-Link III PCIe Baseboard	7
3.2. TI Portfolio for FPD-Link III cameras ^[8]	9
3.3. Example I/O Bank (Bank 44 of XCKU040FFVA1156) ^[10]	11
3.4. Boot Mode Switch	12
3.5. Status LED's	12
3.6. Reset Buttons	13
3.7. Connect Reset Buttons to User Panel (P36, P37)	13
3.8. User GPIOs Connector for User Panel	13
3.9. Power over coax	14
4.1. Vivado - Detected MPSOC with JTAG over USB	16
4.2. HidUartExample shows output of HelloWorld	17
4.3. PCIe Test Setup	18
4.4. Vivado Block Design with XDMA	18
4.5. Setup with RaspberryPi Camera	19
4.6. MIPI CSI-2 RX Subsystem	19
4.7. Measured clock signal	20
4.8. Measured data lane 1	20

List of Tables

3.1. Ultrascale+ Module Comparison	8
--	---

A. Appendix

A.1. Tools

A.1.1. Software

- Altium Designer 18
- Vivado 2018.2
- Xilinx SDK 2018.2
- CP2110 SDK with HidUartExample.exe

A.1.2. Hardware

- Enclustra Modul Mercury+ XU8
- Dell Precision Tower 5810
- Raspberry Pi Camera v2.1
- Oscilloscope Waverunner 8404M

A.2. Scripts

A.2.1. Raspberry Pi Camera

```

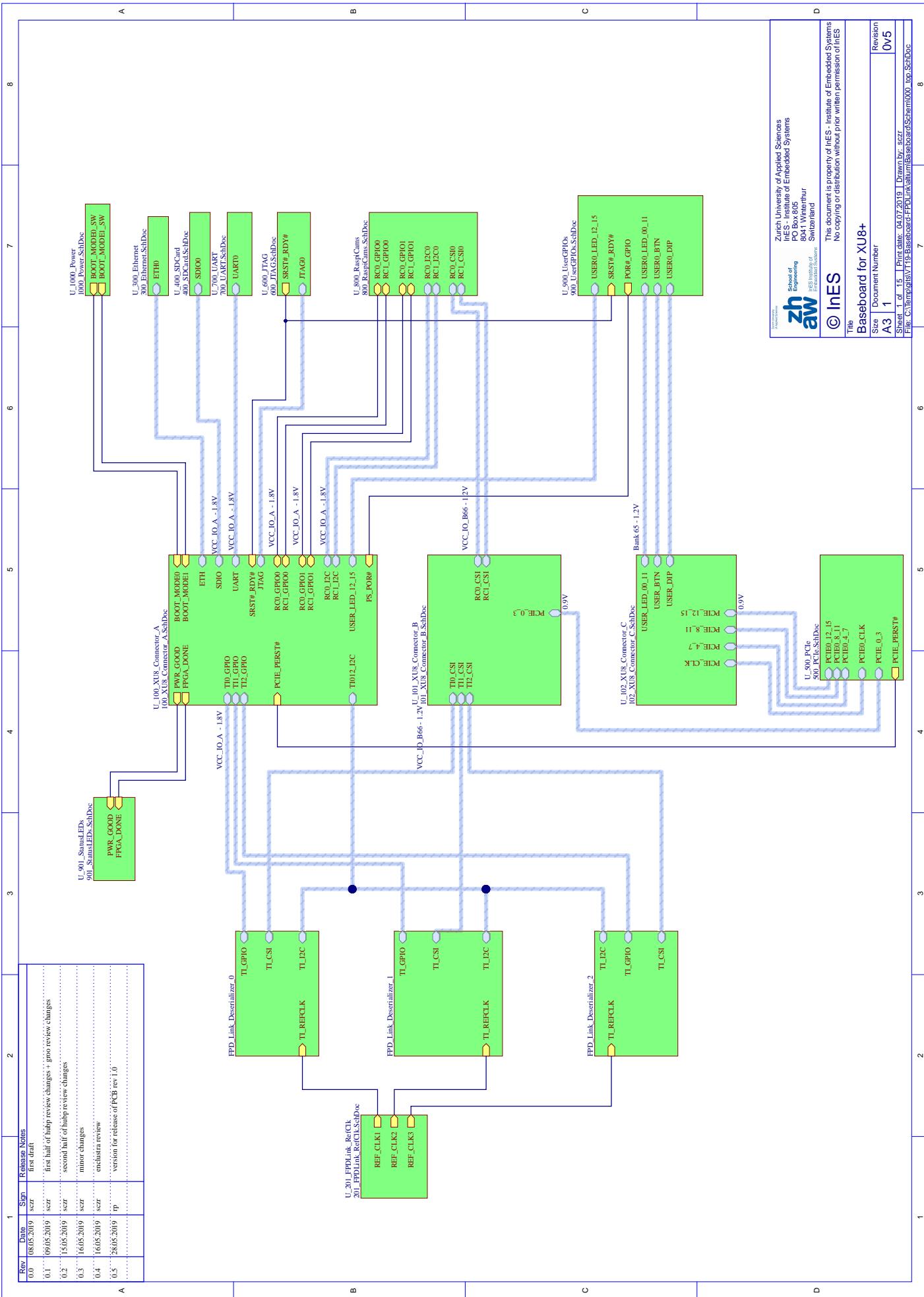
1 // Access Code needed to access registers over 0x3000
2 writeRspReg(0x0100, 0x00);      // 0=OFF, 1=Stream, 2=MAX
3 writeRspReg(0x30EB, 0x05);
4 writeRspReg(0x30EB, 0x0C);
5 writeRspReg(0x300A, 0xFF);
6 writeRspReg(0x300B, 0xFF);
7 writeRspReg(0x30EB, 0x05);
8 writeRspReg(0x30EB, 0x09);
9 writeRspReg(0x0114, 0x01);      // CSI MIPI Lanes [1:0] (0x01=2, 0x03
10 =4)
11 writeRspReg(0x0128, 0x00);      // DPHY_CNTRL
12 writeRspReg(0x012A, 0x18);      // EXCK_FREQ [15:8]
13 writeRspReg(0x012B, 0x00);      // EXCK_FREQ [7:0]
14 writeRspReg(0x0157, 0x00);      // Analog Gain
15 writeRspReg(0x0158, 0x00);      // Digital Gain [15:8]
16 writeRspReg(0x0159, 0x00);      // Digital Gain [7:0]
17 // {0x015A, 0x01}, // Shutter/Integration Time [15:8]
18 // {0x015B, 0x00}, // Shutter/Integration Time [7:0]
19 writeRspReg(0x0160, 0x09);      // Frame Length [15:8]
20 writeRspReg(0x0161, 0xC8);      // Frame Length [7:0]
21 writeRspReg(0x0162, 0x0D);      // Line Length [15:8]
22 writeRspReg(0x0163, 0x78);      // Line Length [7:0]
23 writeRspReg(0x0164, 0x00);
24 writeRspReg(0x0165, 0x00);
25 writeRspReg(0x0166, 0x0C);
26 writeRspReg(0x0167, 0xCF);
27 writeRspReg(0x0168, 0x00);
28 writeRspReg(0x0169, 0x00);
29 writeRspReg(0x016A, 0x09);
30 writeRspReg(0x016B, 0x9F);
31 writeRspReg(0x016C, 0x0C);
32 writeRspReg(0x016D, 0xD0);
33 writeRspReg(0x016E, 0x09);
34 writeRspReg(0x016F, 0xA0);
35 writeRspReg(0x0170, 0x01);      // X_ODD_INC [2:0]
36 writeRspReg(0x0171, 0x01);      // Y_ODD_INC [2:0]
37 writeRspReg(0x0172, 0x03);
38 writeRspReg(0x0174, 0x00);      // Binning Mode H_A
39 writeRspReg(0x0175, 0x00);      // Binning Mode V_A
40 writeRspReg(0x018C, 0x0A);      // CSI Data Format [15:8]
41 writeRspReg(0x018D, 0x0A);      // CSI Data Format [7:0]
42 writeRspReg(0x0301, 0x05);      // VTPXCK_DIV
43 writeRspReg(0x0303, 0x01);      // VTSYCK_DIV
44 writeRspReg(0x0304, 0x03);      // PREPLLCK_VT_DIV [3:0]
45 writeRspReg(0x0305, 0x03);      // PREPLLCK_OP_DIV [3:0]
46 writeRspReg(0x0306, 0x00);      // PLL_VT_MPY [10:8]
47 writeRspReg(0x0307, 0x2B);      // PLL_VT_MPY [7:0]
48 writeRspReg(0x0309, 0x0A);      // OPPXCK_DIV [4:0]
49 writeRspReg(0x030B, 0x01);      // OPSYCK_DIV
50 writeRspReg(0x030C, 0x00);      // PLL_OP_MPY [10:8]
51 writeRspReg(0x030D, 0x55);      // PLL_OP_MPY [7:0]
52 writeRspReg(0x455E, 0x00);      // CIS Tuning ?
53 writeRspReg(0x471E, 0x4B);      // CIS Tuning ?
54 writeRspReg(0x4767, 0x0F);      // CIS Tuning ?
55 writeRspReg(0x4750, 0x14);      // CIS Tuning ?

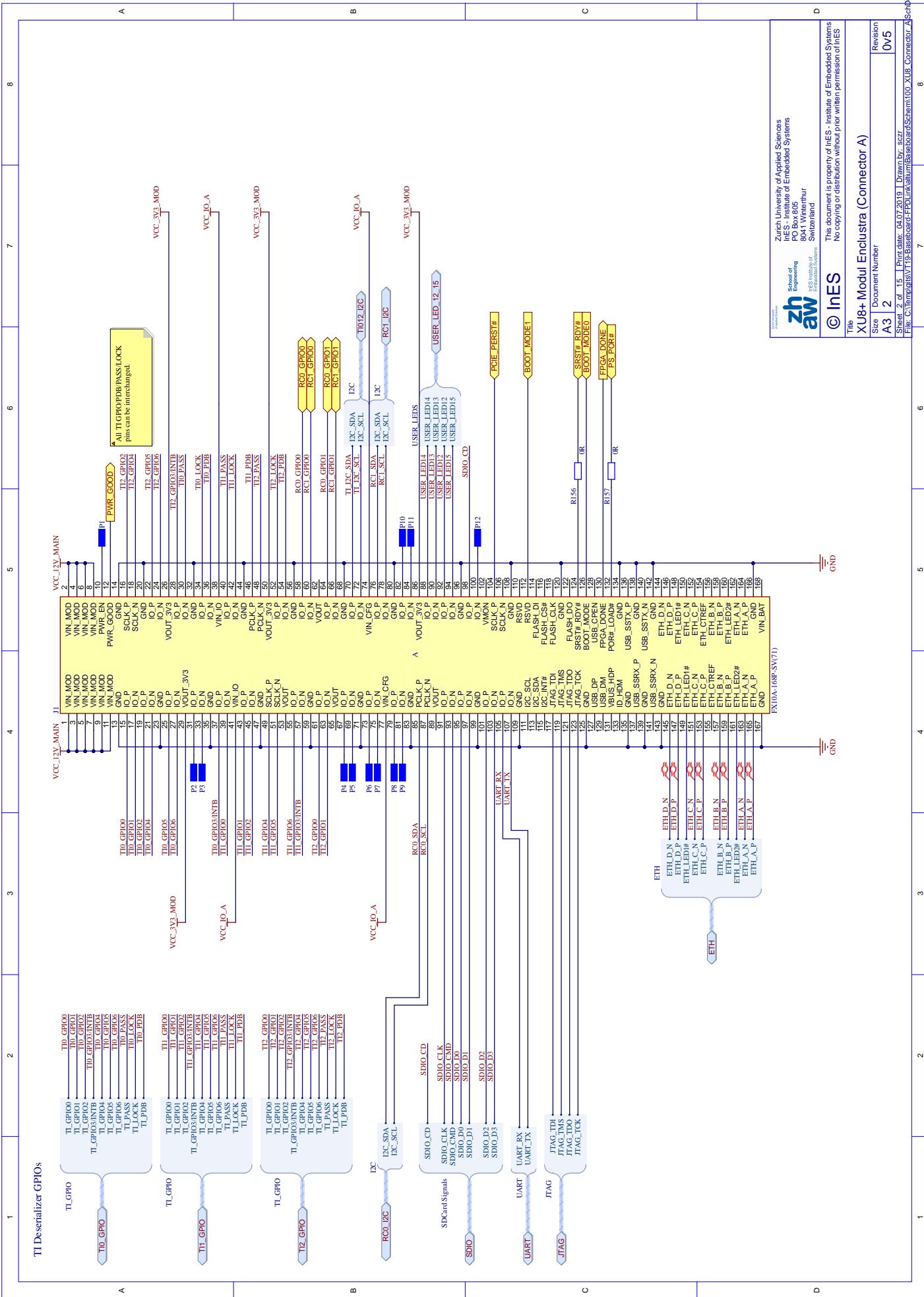
```

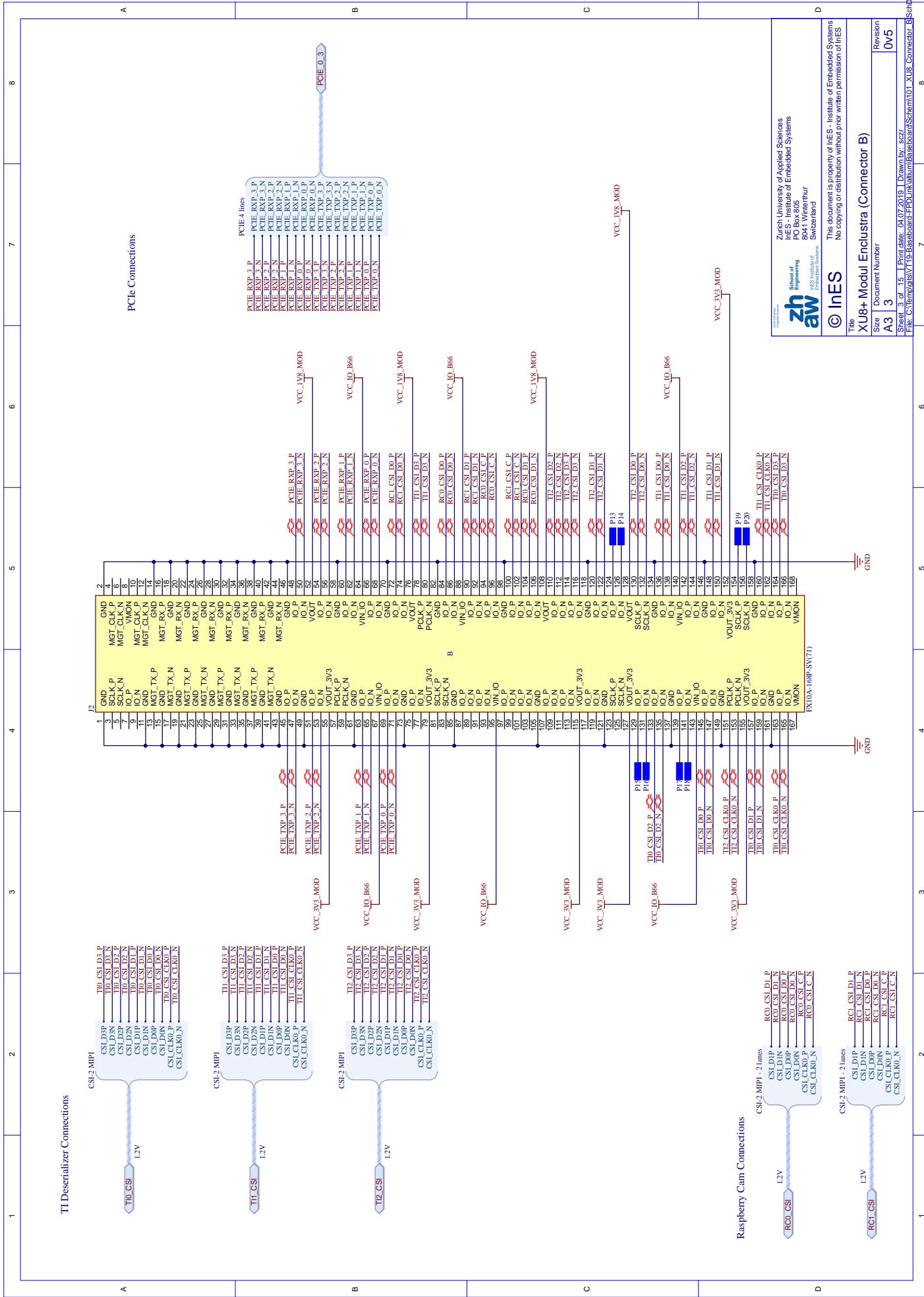
```
1   writeRspReg(0x4540, 0x00);      // CIS Tuning ?
2   writeRspReg(0x47B4, 0x14);      // CIS Tuning ?
3   writeRspReg(0x4713, 0x30);      // CIS Tuning ?
4   writeRspReg(0x478B, 0x10);      // CIS Tuning ?
5   writeRspReg(0x478F, 0x10);      // CIS Tuning ?
6   writeRspReg(0x4797, 0x0E);      // CIS Tuning ?
7   writeRspReg(0x479B, 0x0E);      // CIS Tuning
8
9
10  //start csi mipi data
11  writeRspReg(0x0100, 0x01);
12 }
```

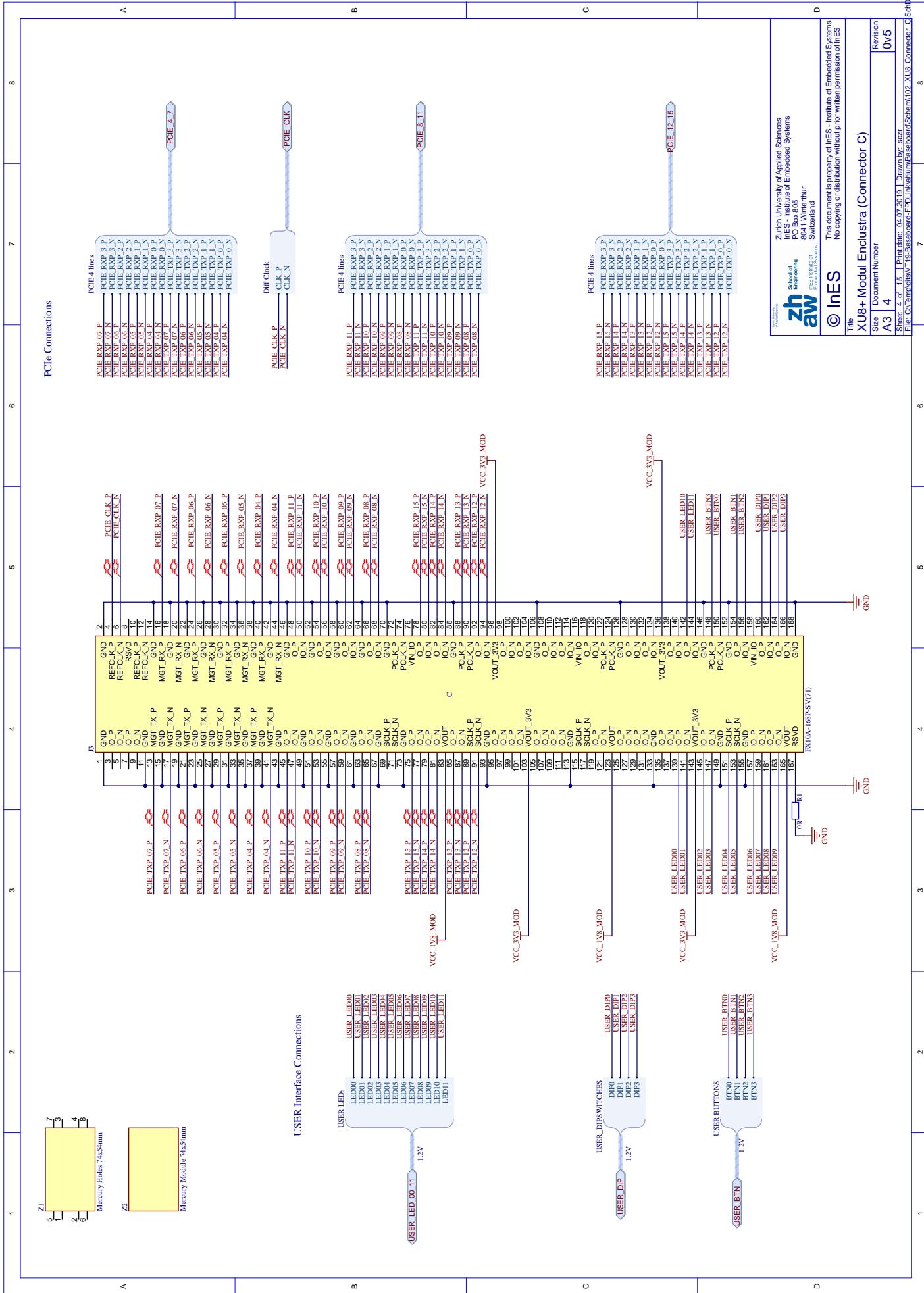
Listing A.1: RaspberryPi Camera Register Setup

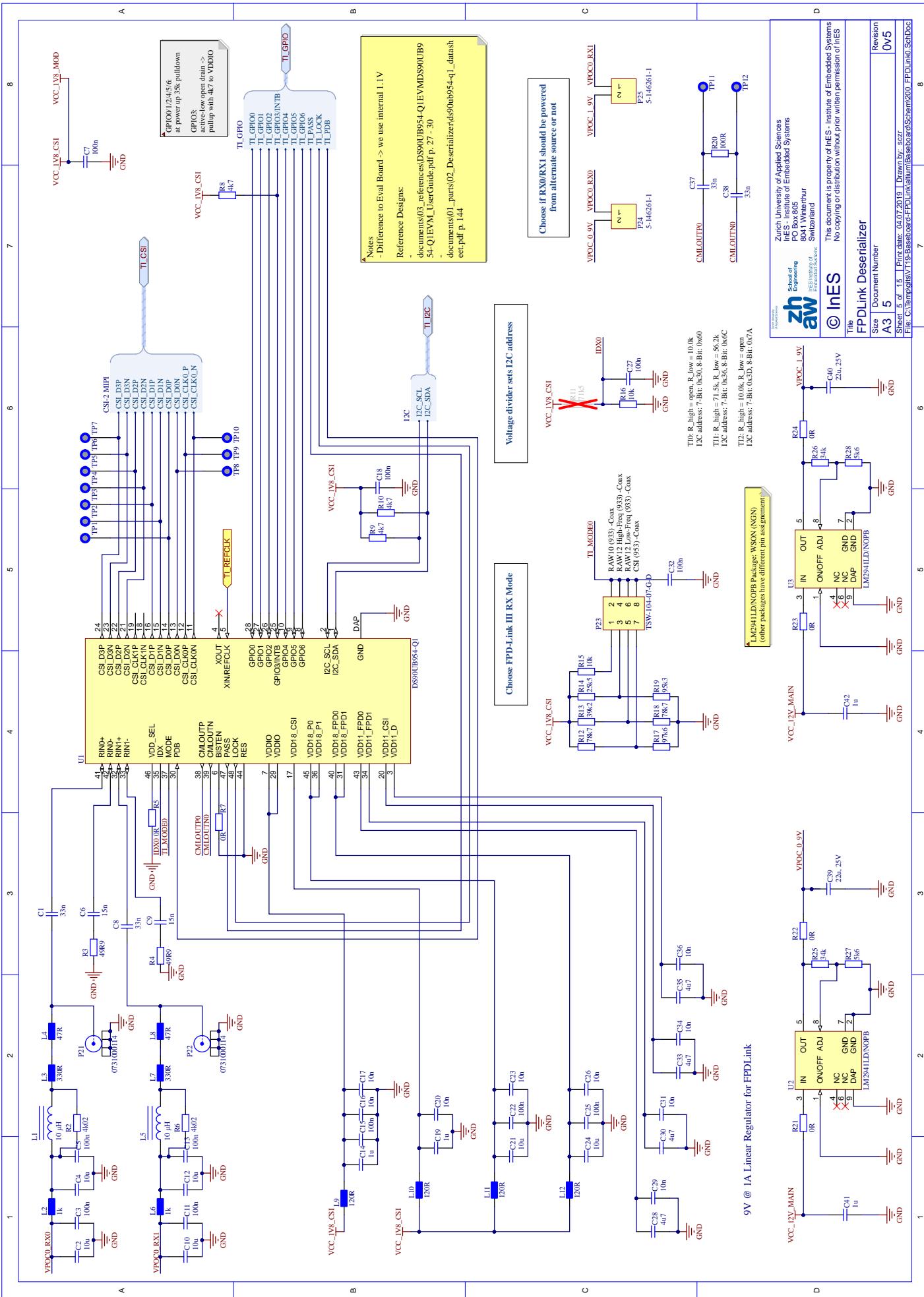
A.3. Baseboard Schematics

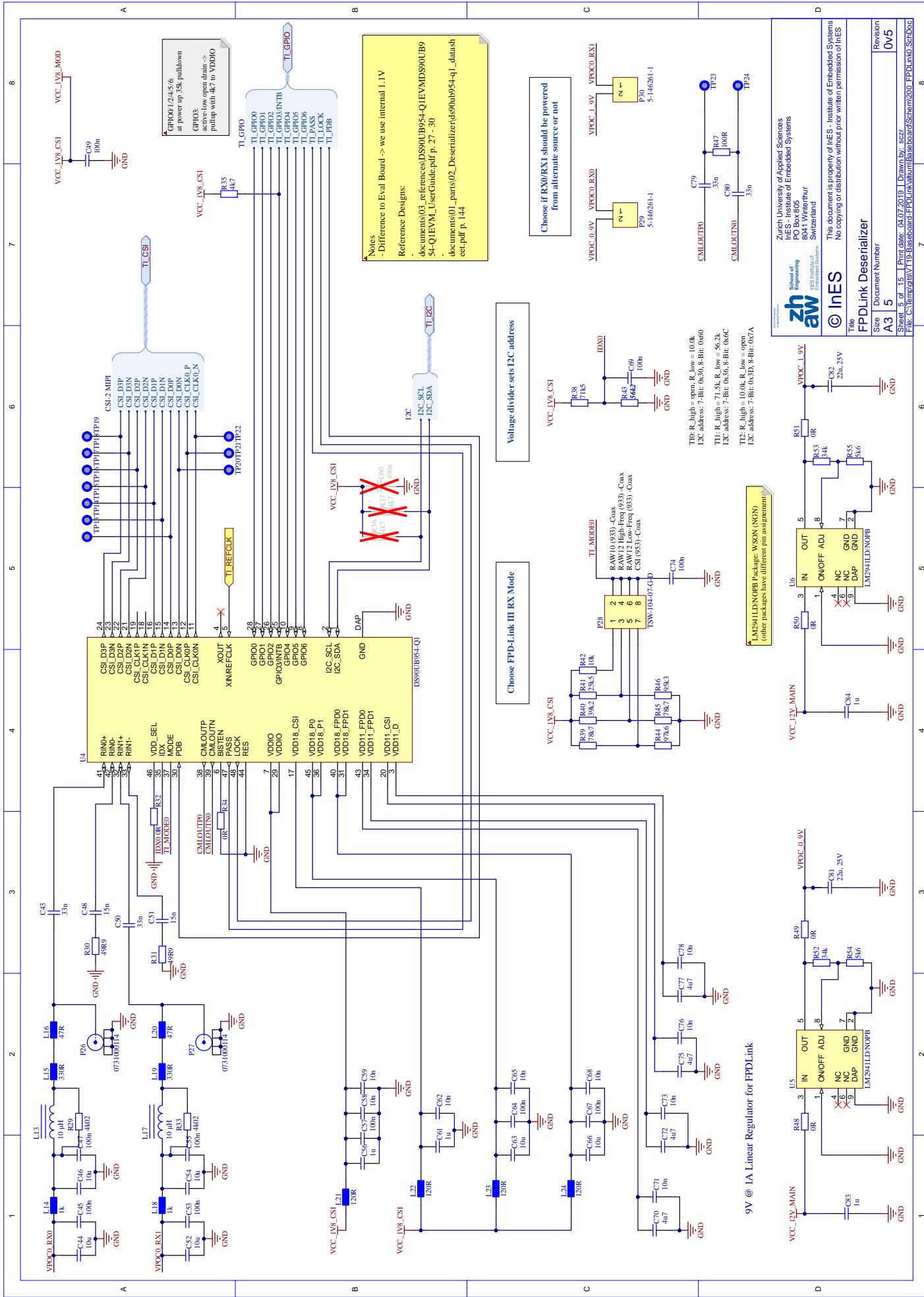


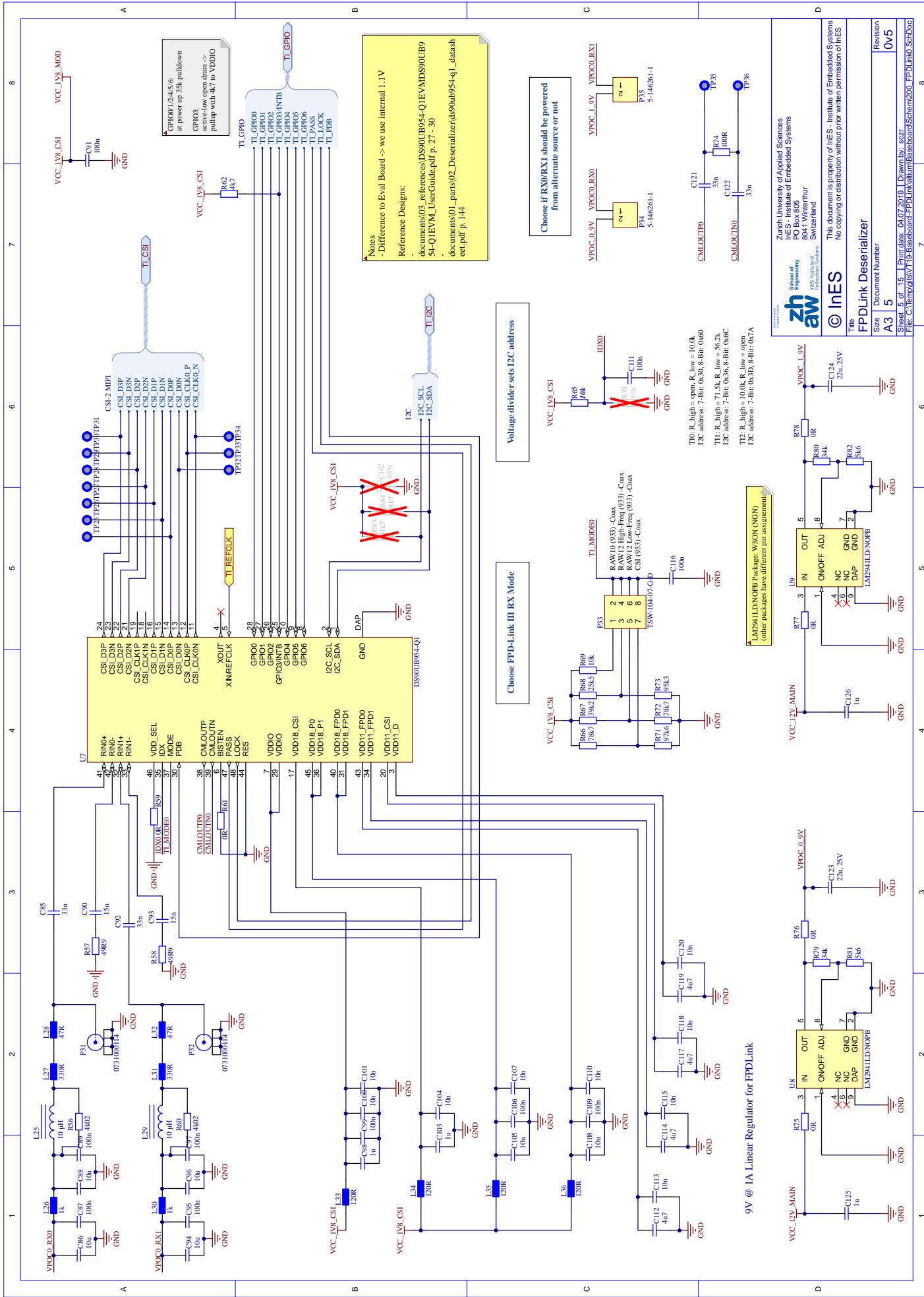


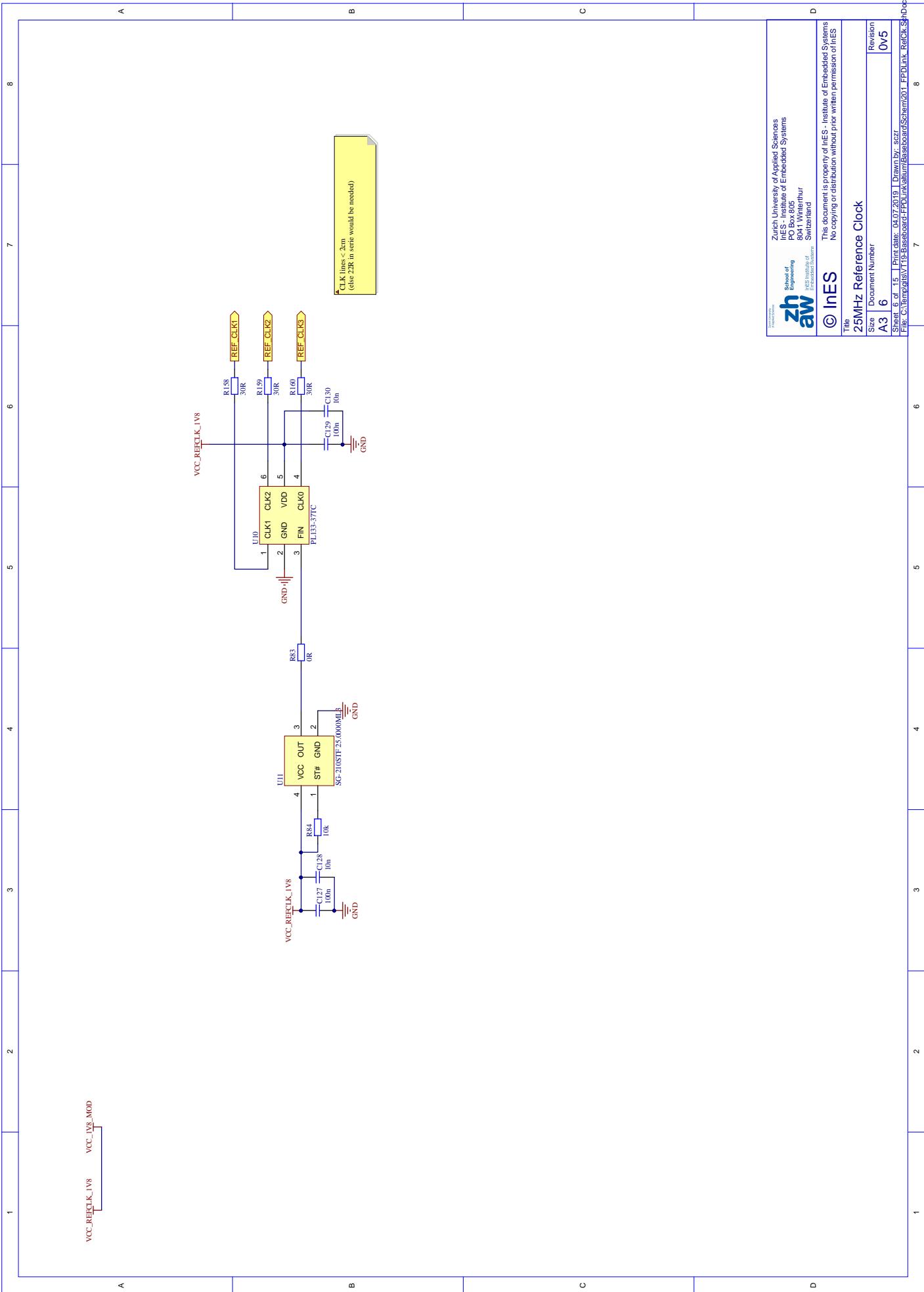


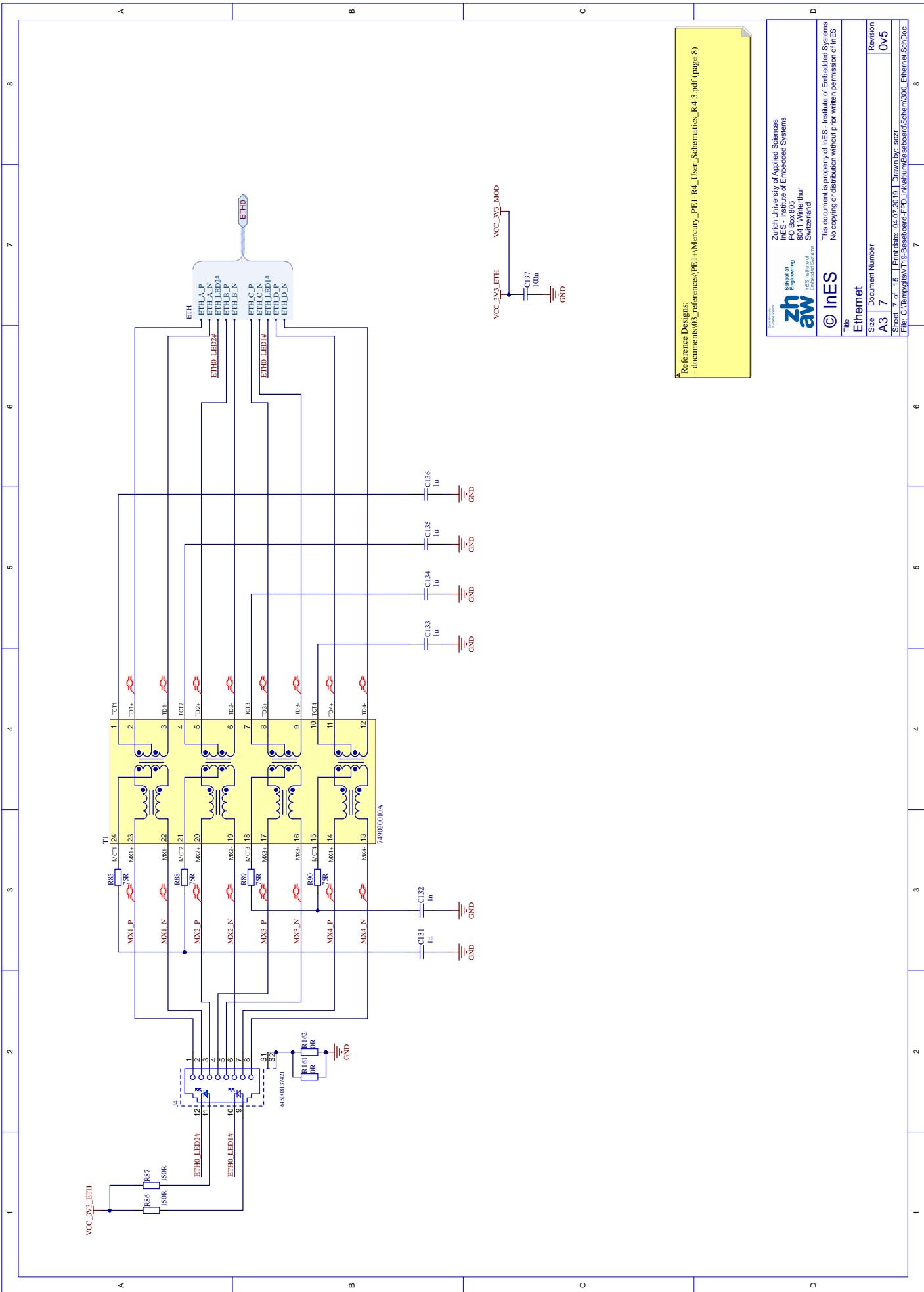


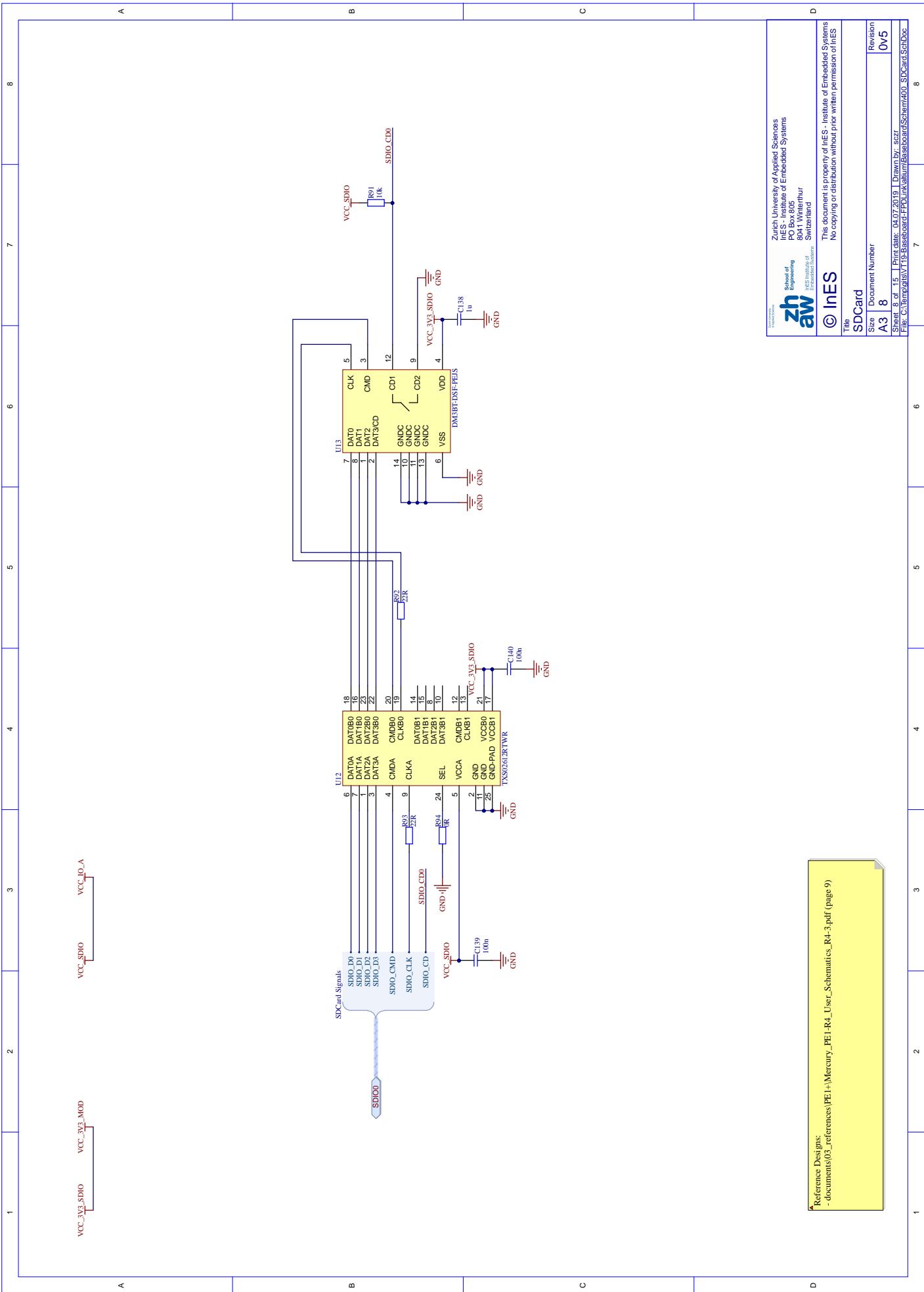


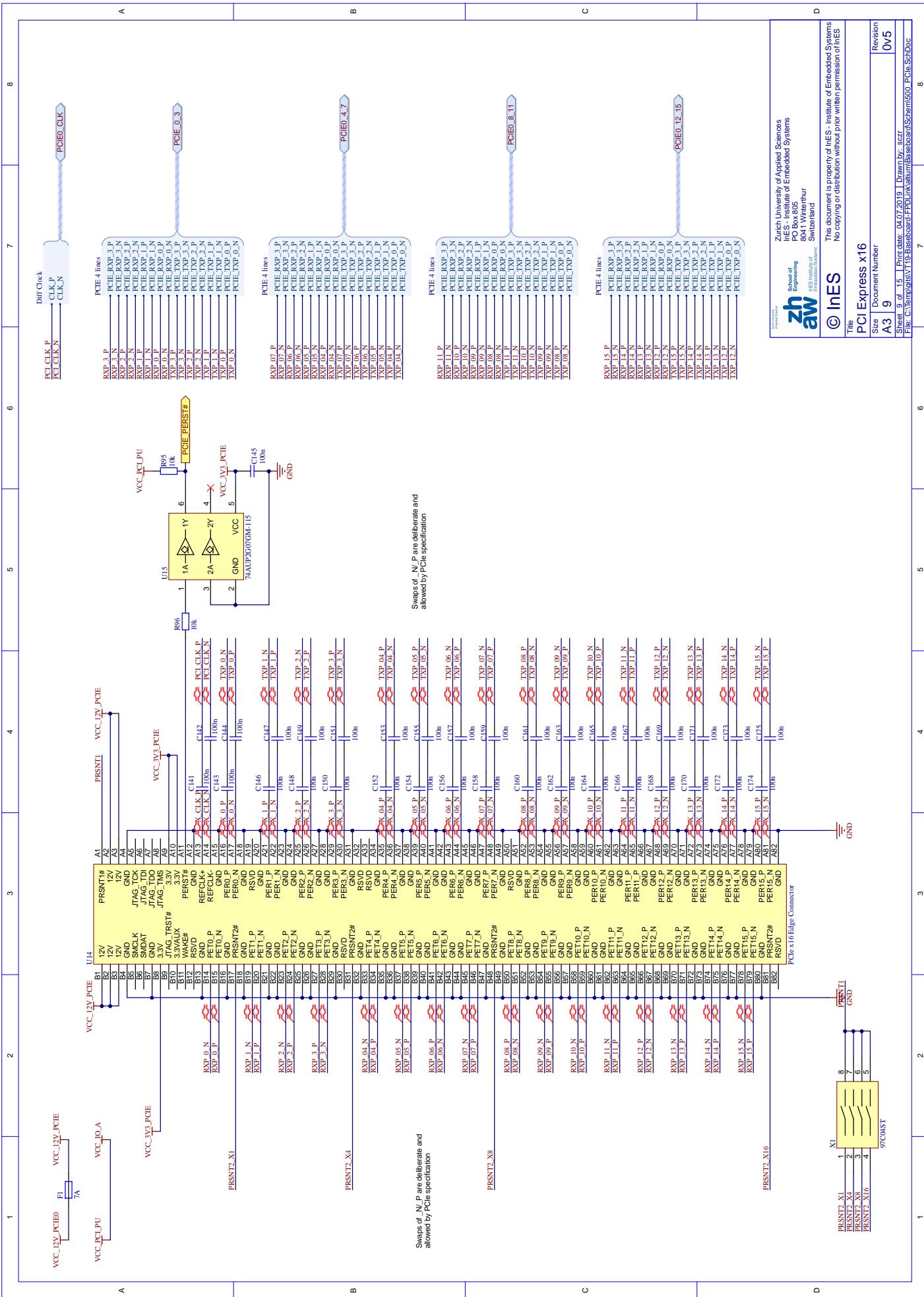


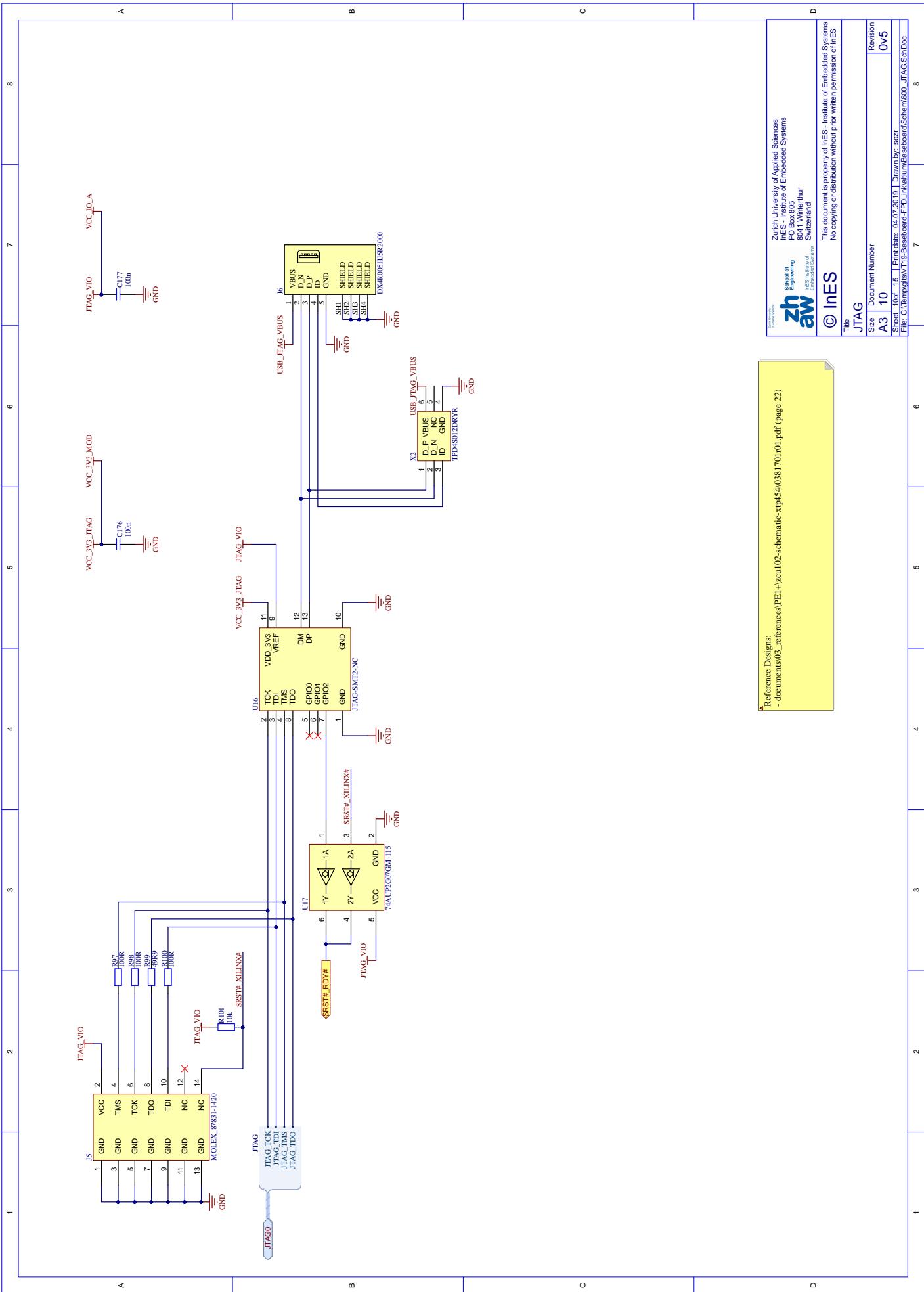


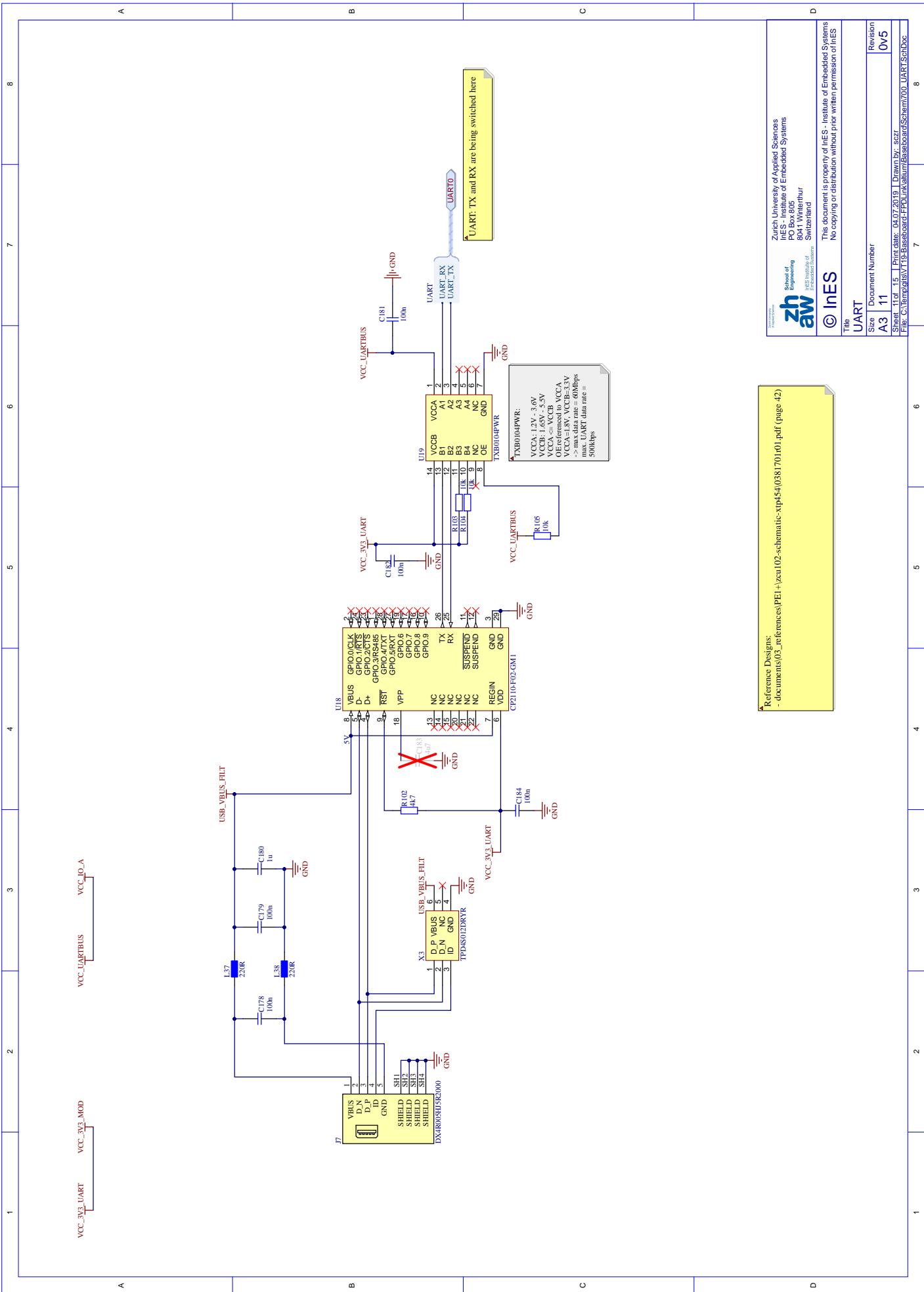


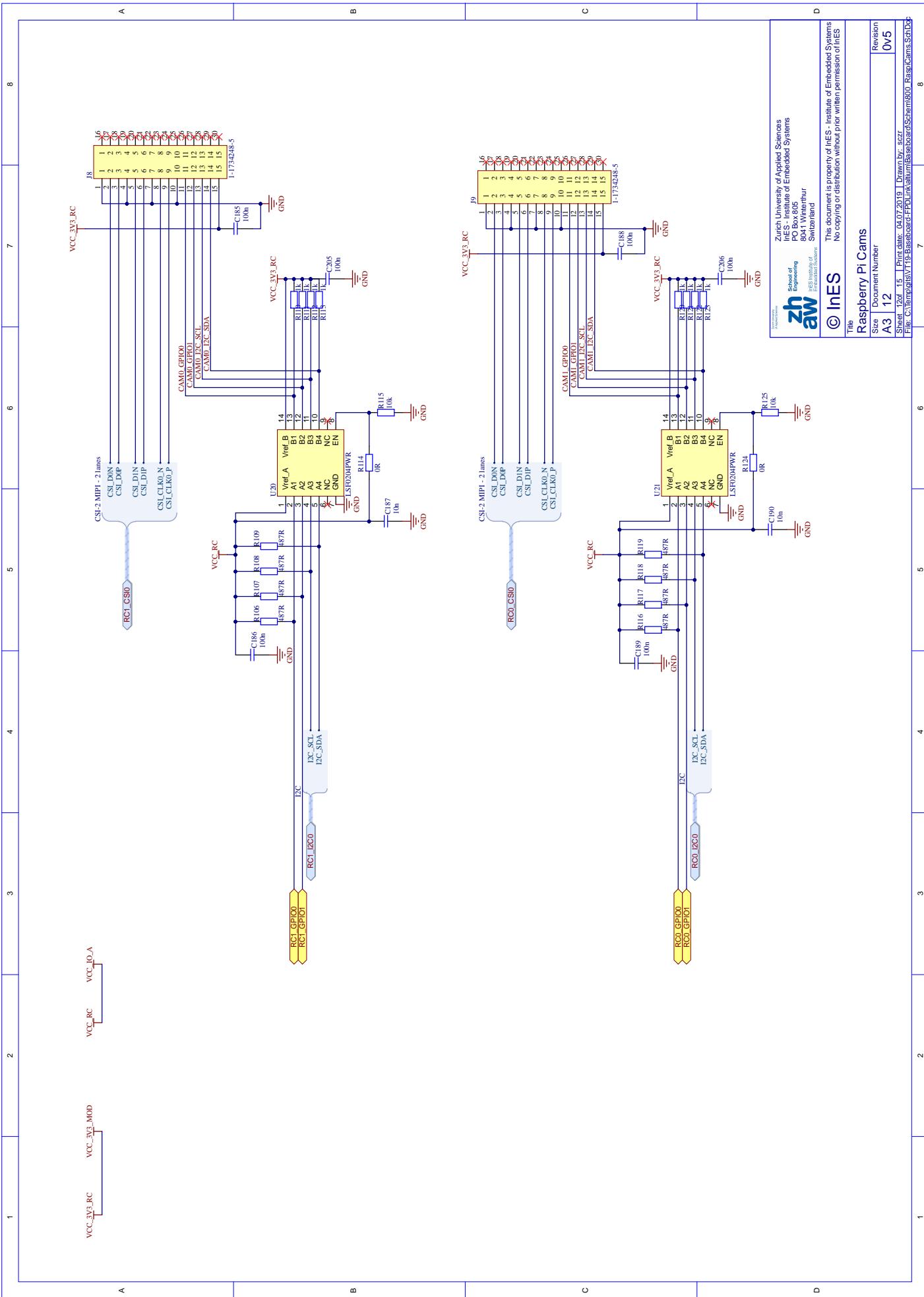


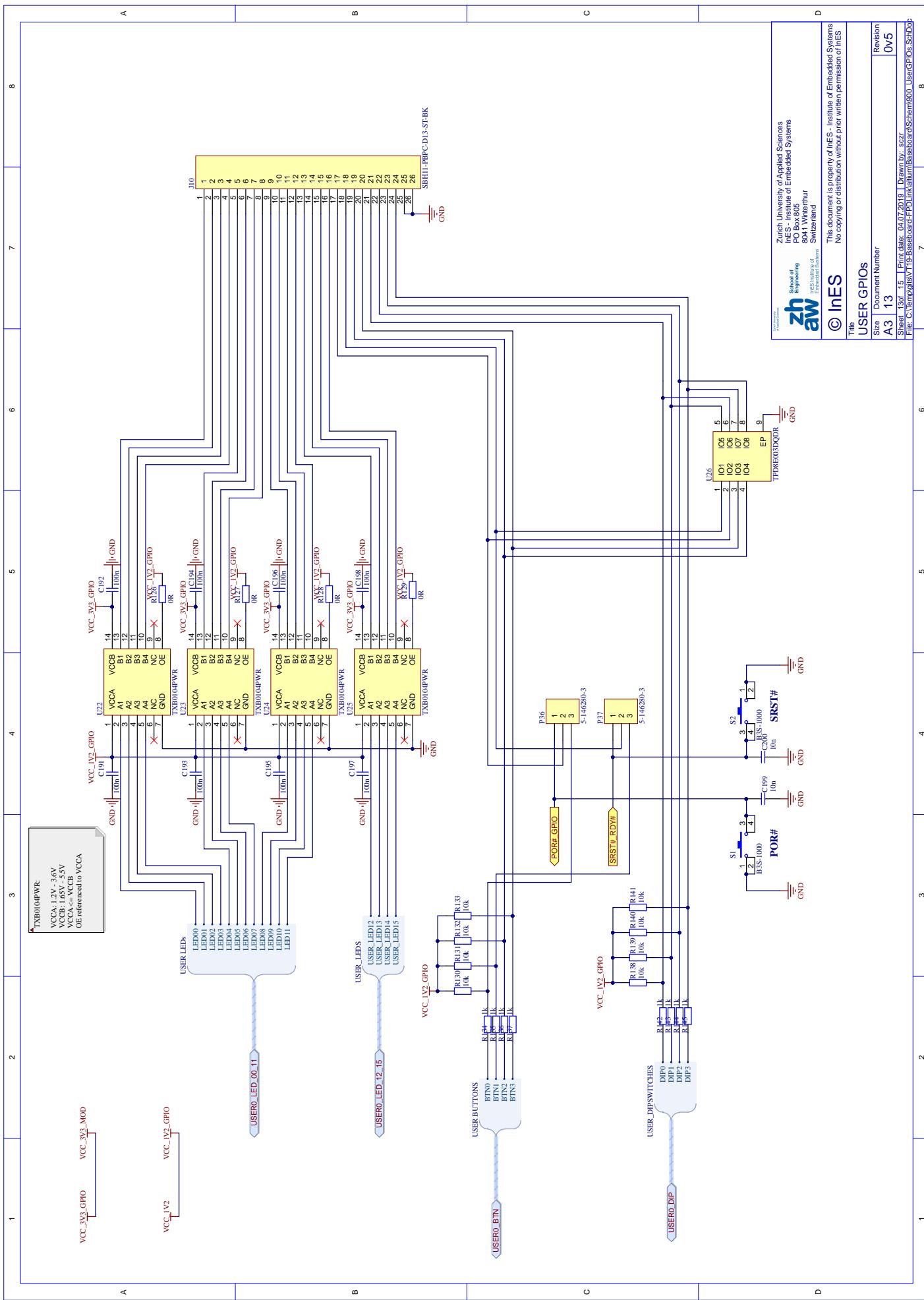


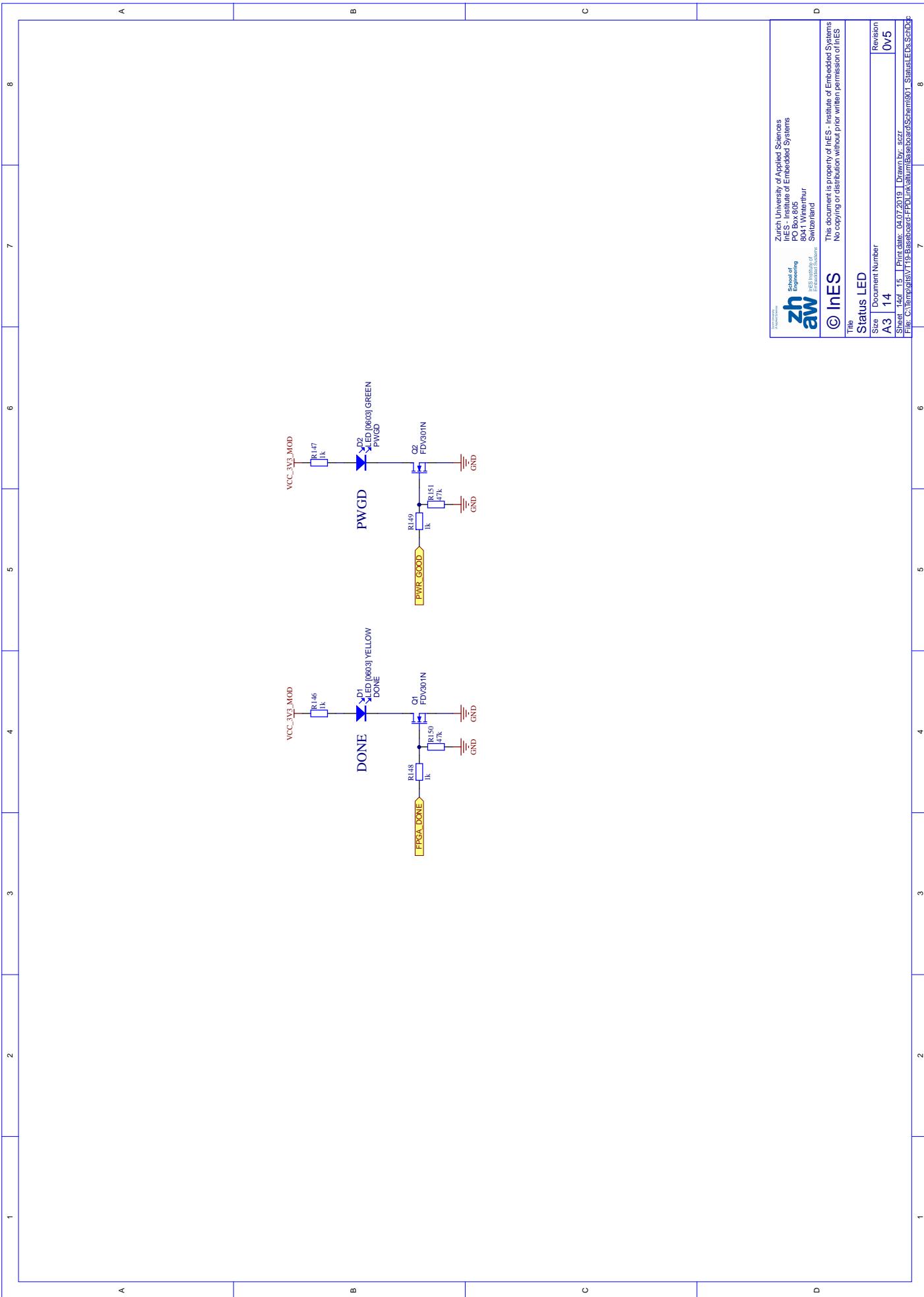


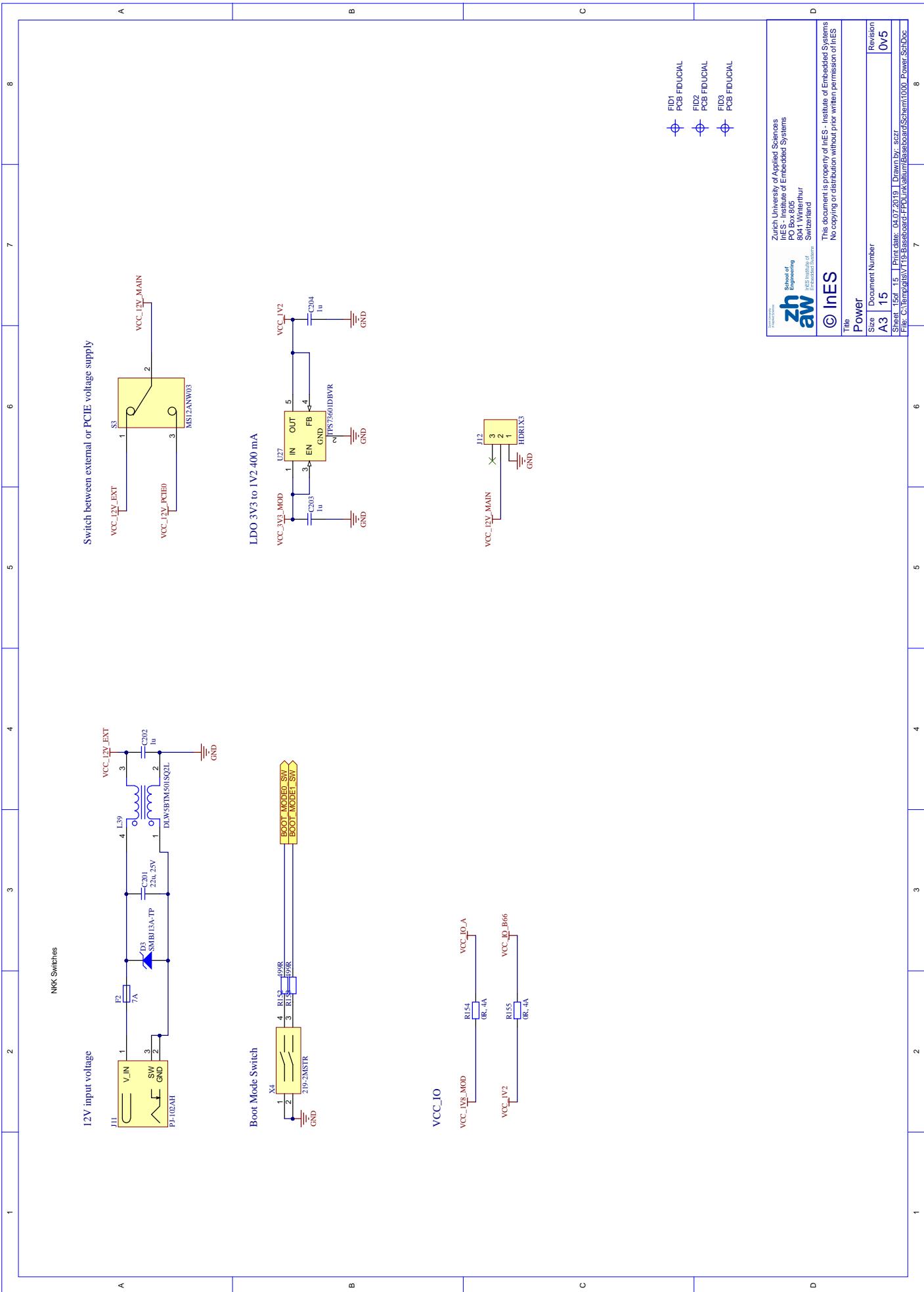












PENDER ELECTRONIC DESIGN
PED-00377-PCB
04.07.2019

Gerber Layer
M01 – Board Outline GM1



A

A

140

B

120

C

100

D

80

8

60

6

40

4

20

2

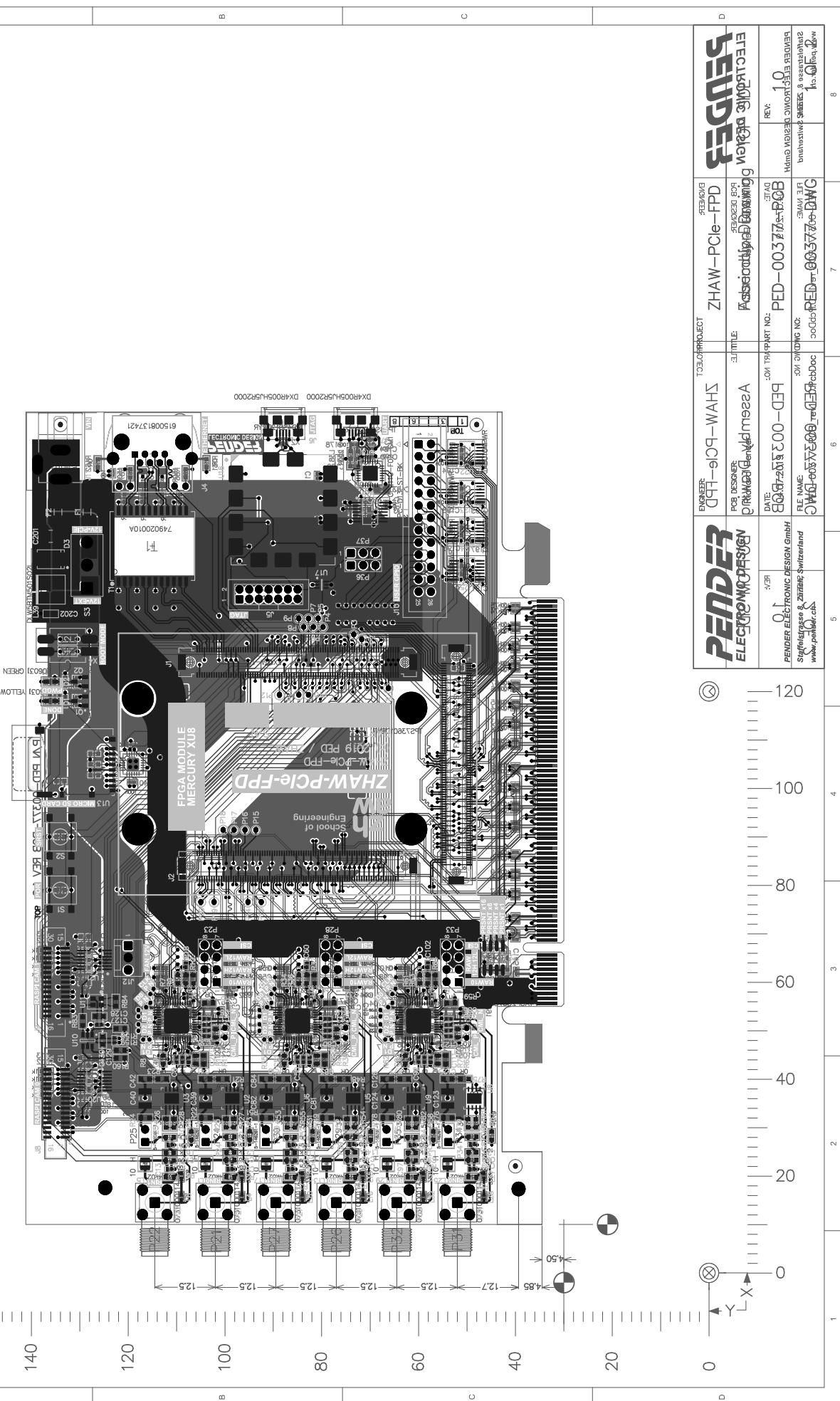
0

1

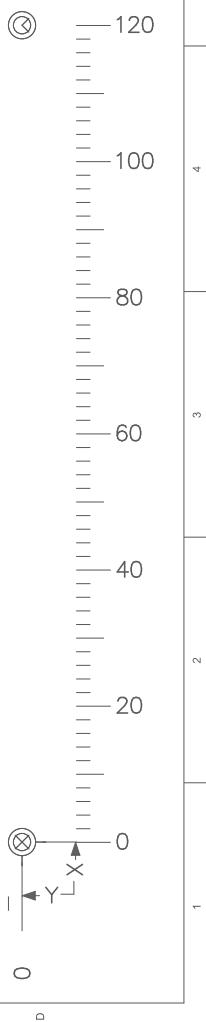
0

Assembly Variant: [No Variations]

Multi-layer Composite Print
Component Assembly Drawing : Reference Designators
Component Assembly Drawing : Part Values

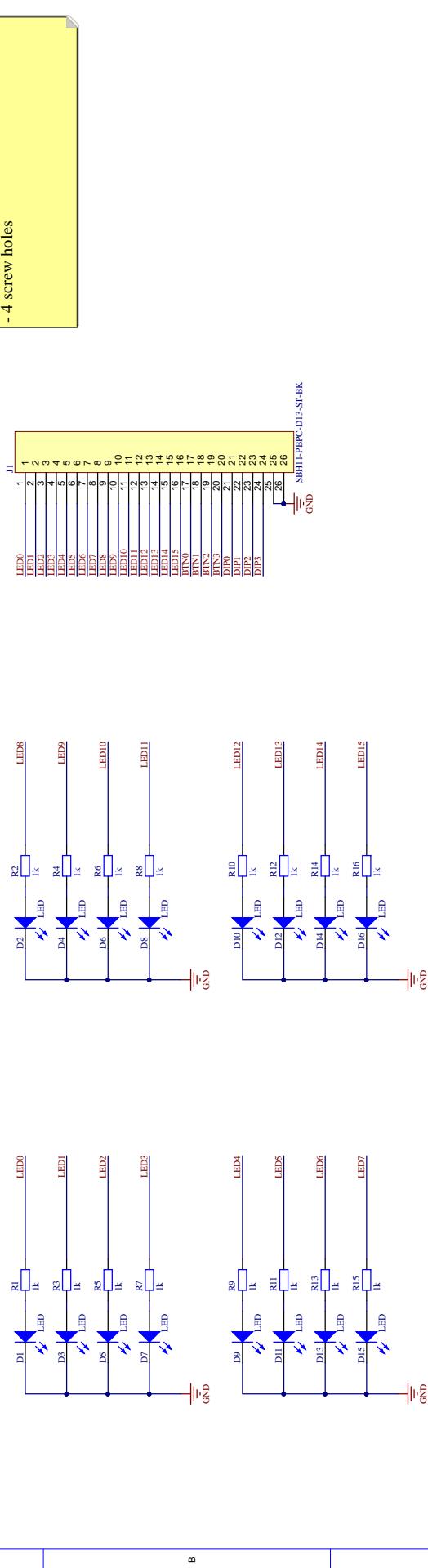


PED-00377-PCB		ZHAW-PCIe-FPD		DATE: 04.07.2019	
DESIGNER:	PROJECT:	FILE NAME:	FILE NO.:	REV.:	REVISION:
EDUARD HABERL	ZHAW-PCIe-FPD	EDUARD HABERL	PED-00377-PCB	1.0	1.0
PENDER ELECTRONIC DESIGN GmbH	Steffelgasse 2, 8401 Winterthur, Switzerland	www.pender.ch			



A.4. User Panel Schematics

Rev	Date	Sign	Release Notes
0.0	08/05/2019	sczr	First Draft
0.1	14/05/2019	sczr	Rename RST > BTN0, added CS, added Note, removed BTN resistors
1			
2			
3			
4			
5			
6			
7			
8			



- Measurement: 30mm x 100mm
- 4 screw holes

