TALLINN UNIVERSITY OF TECHNOLOGY
School of Information Technologies

Veiko Rütter 221946 IASM

DEVELOPMENT OF A CAMERA CONTROLLER PROTOTYPE FOR DRONES

Master's Thesis

Supervisor: Peeter Ellervee
PhD
Co-Supervisor: Mattias Luha

Tallinn 2023

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Veiko Rütter 221946 IASM

KAAMERA KONTROLLERI PROTOTÜÜBI ARENDUS DROONIDELE

Magistritöö

Juhendaja:	Peeter Ellervee
	PhD
Kaasjuhendaja:	Mattias Luha

Tallinn 2023

Author’s declaration of originality

I hereby certify that I am the sole author of this thesis. All the used materials, references to the literature and the work of others have been referred to. This thesis has not been presented for examination anywhere else.

Author: Veiko Rütter

08.05.2023

Abstract

Krattworks Ltd builds drones with on-board machine vision module. Their drones can detect humans, cars, location of the fire etc. automatically and share this information instantly with the Control Room or units on the ground over radio networks.

The central task of any drone is to provide aerial data for the users: all drones are looking for the best balance between maximum image quality (heavy cameras and lenses) and minimum weight of the payload (maximizing flight time). The challenge KrattWorks faces today is that their camera system does not have any optical zoom capability: at the same time they can not increase the total weight of the drone and the size of the camera system should stay more-or-less the same. Commercially available optical zoom lenses are heavy in weight and would increase the size of the camera system a lot. Finding anything small in size and weight and customizable would be very expensive.

The potential solution is to design a camera controller that enables to use 2 visual light cameras: one with wide, and the other with narrow field of view lens. This enables to combine digital and optical zoom to achieve 10X optical zoom with light weight, small size and without any moving parts.

The thesis focuses on development this type of controller. As a result of the work, a working prototype is completed. The design being created includes visible light camera and an infrared camera. The work includes hardware design, camera interfaces in HDL (*Hardware Description Language*) and software.

This thesis is written in English and is 41 pages long, including 9 chapters, 28 figures and 10 tables.

Annotatsioon

Kaamera kontrolleri prototüübi arendus droonidele

OÜ Krattworks ehitab droone mille pardal on masinnägemise võimekus. Need droonid suudavad tuvastada inimesi, autosid, tulekahjude asukohta jne automaatselt ning edastada saadud info juhtimise keskusesse üle raadiovõrgu.

Droonide põhiline ülesanne on saata kvaliteetset aeroinfot kasutajale. Selle ülesande efektiivseks saavutamiseks tuleb leida mõistlik piir maksimaalse pildikvaliteedi (raskemad kaamerad ja läätsed) ning minimaalse kaalu vahel (pikem lennuaeg). Krattworksi väljakutse käesoleval hetkel seisneb selles, et nende kaamerasüsteemil puudub optilise suurenduse (nn. suumi) võimekus. Kuid seda lisada on keeruline, kuna kaamerasüsteemi kaalu ei saa tõsta, sest kogu drooni kaal peab jääma ettenähtud piiridesse. Lisaks on kommertslahendused kas liiga kallid või neid ei saa vastvalt vajadusele konfigureerida või liidestada.

Üks võimalikke lahendusi on sellise kaamerasüsteemi loomine, mis sisaldab kahte nähtava valguse kaamerat: üks laia vaatenurgaga objektiiviga ning teine kitsa vaatenurgaga objektiiviga. Selline lahendus võimaldaks kombineerituna koos digitaalse suumiga saavutada 10X pildi suurendus ilma liikuvate osadeta, jäädes seejuures kergeks ja väikseks.

Antud lõputöö keskendub sellise kontrolleri loomisele. Töö tulemusena valmib kaamerasüsteemi riistvara prototüüp, mis sisaldab valitud nähtava valguse kaamerat ning infrapuna kamaerat. Töö sisaldab endas riistvara disaini, kaameraliideste disaini HDL'is (*Hardware Description Language*) ning vajalikku tarkvara.

Lõputöö on kirjutatud inglise keeles ning sisaldab teksti 41 leheküljel, 9 peatükki, 28 joonist, 10 tabelit.

List of abbreviations and terms

ARM	Advanced RISC Machines
AXI	Advanced eXtensible Interface
BGA	Ball Grid Array
CPU	Central Processing Unit
CSI	Camera Serial Interface
D-RAM	Dynamic Random-Access Memory
DMA	Direct Memory Access
FAT	File Allocation Table
FFC	Flat Flex Cable
FHD	Full High-Definition
FIT	Flattened Image Tree
FPGA	Field Programmable Gate Array
GPIO	General Purpose Input-Output
HDL	Hardware Description Language
HDMI	High-Definition Multimedia Interface
IO	Input-Output
IP	Intellectual Property
ITB	Image Tree Blob
ITS	Image Tree Source
JTAG	Joint Test Action Group
LDO	Low Dropout
LUT	Look-up Table
LVDS	Low-Voltage Differential Signaling
MIPI	Mobile Industry Processor Interface
NDA	Non-Disclosure Agreement
PCB	Printer Circuit Board
PID	Proportional-Integral-Derivative
PoC	Proof of Concept
RISC	Reduced Instruction Set Computer

SD	Secure Digital
SoC	System on a Chip
SOM	System on a Module
SOT	Small Outline Transistor
SPI	Serial Peripheral Interface
SPL	Secondary Program Loader
SSH	Secure Shell
UART	Universal Asynchronous Receiver-Transmitter
USB	Universal Serial Bus

Table of Contents

List of Figures

List of Tables

1 Introduction

This thesis focuses on creating a new camera system for Krattworks drones.

Krattworks Ltd builds drones with machine vision capabilities on board. Their drones are equipped with a controller that has a visible light camera and a thermal camera [1]. Their drones can detect humans, cars, location of the fire etc. automatically and share this information instantly with the Control Room or units on the ground over radio networks.

The problem Krattworks is facing today, is that the current design does not allow to use cameras with different hardware interfaces and therefore it is difficult to change cameras in this design. Only a maximum of two MIPI (*Mobile Industry Processor Interface*) cameras can be used and there is no parallel camera interface. Because of that, the thermal camera is connected via USB (*Universal Serial Bus*) and this connection not reliable.

To solve the situation, a new camera controller was developed as part of the thesis. The work includes searching for new components, building a hardware prototype, installing an operating system and writing software to set up camera streams.

As a result, a working prototype was completed and the choice of processing system and camera sensors was proven. The result is a PoC (*Proof of Concept*) for the new camera controller hardware.

In order to get a better overview of the system, the existing solution is described below.

1.1 Overview of the existing system

The drone has a total weight of 1.6kg and flight time of 40 minutes. The drone has a visual light camera and a thermal camera. For communication, the drone has a 5G radio and a 2.4GHz point-to-point radio. The drone can also be operated out from the drone nest without the need for an onsite pilot: the drone takes off and lands autonomously and charges inside the box [2]. The drone currently in use is shown in Figure 1.

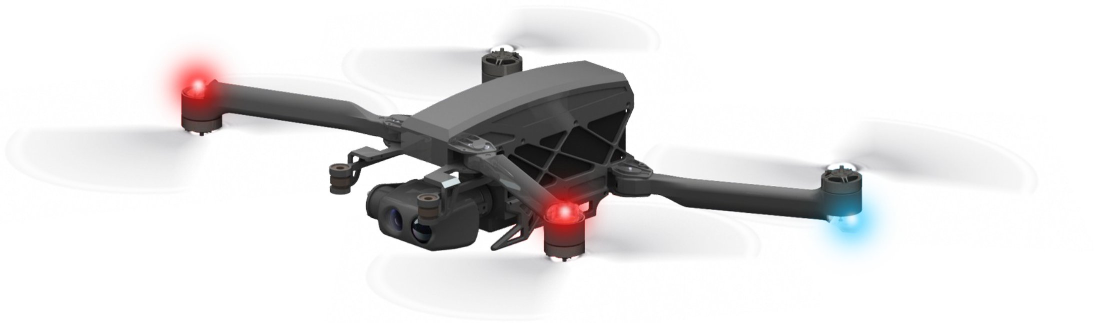


Figure 1. Drone with camera controller [2].

In order to transmit video from the cameras to the rest of the system, the camera controller has an ethernet interface. Commands are also given to the camera controller over the ethernet interface.

The system uses one visible light camera module that uses MIPI CSI-2 (*Camera Serial Interface*) and a thermal camera that uses USB interface. Oclea CV25 is used as the processing system, which contains Ambarella™ SoC (*System on a Chip*) [3]. Figure 2 gives a simplified overview of the existing system and also the design of the new system for comparison.

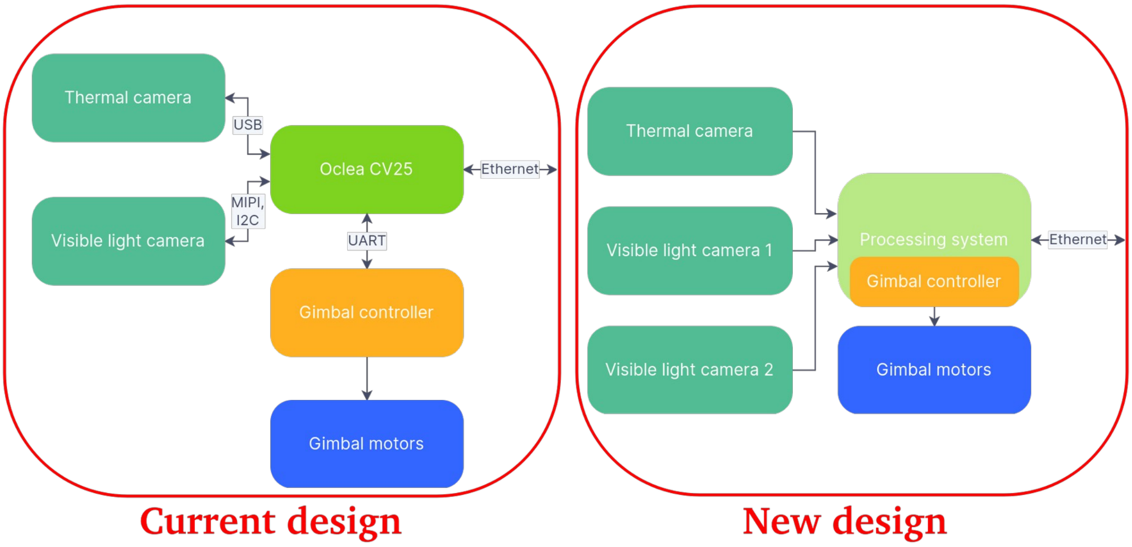


Figure 2. Overview of current and new architecture.

The disadvantages of the current processing system are that a maximum of two MIPI CSI-2 cameras can be connected to it and it has no parallel camera interface. Also, the USB connection of the thermal camera has a very high latency and is not stable enough

- the connection with the camera is often interrupted, and the camera needs to be restarted to establish the connection again.

1.2 Overview of the redesigned system

The aim is to replace the camera sensors and processing system in the existing system and ideally, integrate the controller of the gimbal motors with the processing system. The motor controller has to work in real time, which is why a separate microcontroller was used at the moment. If it were possible to use real-time resources, the PID (*Proportional-Integral-Derivative*) controller, which drives the gimbal motors, could be integrated with the processing system and the number of system components could be reduced. Also, the new design could have the option of adding more than 3 camera sensors, although this option remains unused for now. But the need may arise in the future.

Below is a list of the tasks that were done to complete the prototype:

- Selection and analysis of the components of the new system
- Electronic circuit design
- Hardware PCB (*Printed Circuit Board*) design
- Building the bootloader
- Building the Linux kernel
- Creating an operating system image
- FPGA (*Field Programmable Gate Array*) design
- Software development
- Testing, conclusions

The tasks were done in the same order, as each previous task is a prerequisite for the next task. The tasks are described in more detail below.

2 Analysis of requirements

The first step before designing the system is to select the core components. The following chapters concentrate on the requirements of different parts of the system. But first of all, research was done on existing camera controllers.

2.1 Overview of commercial camera controllers

There are existing camera controllers on the market. But they are not customizable enough. In the Krattworks solution, it is necessary to run the software as close as possible to the camera interfaces. But commercial controllers usually send out a video stream, and in turn, an additional controller would be needed to process this stream. For comparison, some studies have been done on existing camera controllers. Table 1 describes different camera controllers that would be suitable with concessions.

Table 1. List of commercial camera controllers with similar parameters.

Camera controller	Weight	Size	Thermal Resolution
GSG 201 [4]	3.5kg	323x200x200mm	1024x768
USG-400 [5]	3.1kg	180x180x250mm	640x512
HD25-LV [6]	360g	72x72x110mm	640x480
Foxtech Bi-Focus [7]	633g	140x146x160mm	320x240

2.2 Thermal camera selection

When choosing a thermal camera, the first requirement is the physical size of the camera. This is important because, in most cases, the thermal camera module is shipped with appropriate lens. The camera data interface must be low-level protocol to reduce latency, i.e. MIPI CSI-2 or parallel camera interface. And of course, in this case, documentation is also needed. In addition, it must be possible to buy the camera somewhere, so reliability of supply is important. And not much less important is the price of the camera. The resolution of the image must be at least 640x480 pixels. During the design of this system, two thermal camera modules were found that would fit this system. These cameras are also sold in small quantities and documentation can be obtained by signing an NDA (*Non-Disclosure Agreement*). The final decision depends on price and quality. The plan was to order both for testing purposes, and after the test results, choose which one to continue with. Suitable thermal cameras are listed in Table 2.

Table 2. List of suitable thermal cameras.

Camera module	Interface	Resolution	
Dione 640 [8]	Parallel Camera Interface	640x480	
InfiRay Xcore MicroIII [9]	Parallel Camera Interface	640x512	

2.3 Visible light camera sensor selection

When choosing a sensor, it was important to have an image with a higher resolution than the screen resolution, so that there would be the possibility of digitally enlarging the image. But at the same time, the sensor must have very good low-light vision capabilities. Most of the display screens used in the Krattworks application have a FHD (*Full High-Definition*) resolution: 1920x1080 pixels (2.1 MP), and even for screens with a higher resolution, the video quality is mostly limited to HD (1280x720 pixels) or FHD resolution, since the radio communication does not allow sending a higher resolution video.

The first requirement when choosing a sensor was therefore the resolution required for digital zoom, which had to be much higher than the 2.1MP screen resolution.

This is contradicted by the second most important requirement: to have a very good ability to see in low light, since the best low-light ability (for a sensor with similar characteristics) is ensured by a maximum pixel pitch on the sensor. The larger the pixel on the sensor, the more light falls on it and the more light-sensitive it is. Therefore, a sensor with a lower resolution is more sensitive to light than a camera with a higher resolution if the sensors have the same physical dimensions.

Since the camera system had to be small and light in the end, it was possible to use maximum M12 lenses there. However, these lenses do not allow the use of a sensor larger than 1/1.7": the light ring coming through the lens must be larger than the diagonal of the sensor. Since the lenses of the next largest "C-Mount" and "CS-Mount" are several times heavier and larger than the "M12 Mount" lenses, it was not possible to choose a larger sensor to ensure the ability to see in low light. Otherwise, it would have been possible to go for both a higher resolution and a larger sensor, which would have left the pixel size on the sensor the same. Due to the physical size and light transmission capacity of the lenses, it was possible to choose a suitable sensor from the list of available sensors with a maximum size of 1/1.7".

18 possible variants of sensors of this physical size were identified. Of these, 13 units had low-light capability as seen in Table 3. On-Semi AR0821 and AR0522 type sensors and Sony IMX Starvis type sensors turned out to be the best.

Table 3. Lits of suitable sensors

Sensor	Size (inches)	Resolution (MP)	Pixel pitch (µm)	Low light	Frame rate
IMX477	1/2.3" (6.2x4.6 mm)	12.5 (4072x3064)	1.55x1.55 µm	yes	60
OX08B40 RGB + NIR	1/1.7" (7.5x5.5 mm)	8.3 (3840x2160)	2.1x2.1 µm	yes	36
OX08A4Y	1/1.7" (7.5x5.5 mm)	8.3 (3840x2160)	2.1x2.1 µm	yes	36
AR0221	1/1.7" (7.5x5.5 mm)	2.2 (2048x1088)	4.2x4.2 µm	yes	60
IMX464 Starvis	1/1.8" (7.2x5.3 mm)	4.1 (2688x1520)	2.9x2.9 µm	yes	90
IMX347 Starvis	1/1.8" (7.2x5.3 mm)	4.1 (2688x1520)	2.9x2.9 µm	yes	90
IMX678 Starvis2	1/1.8" (7.2x5.3 mm)	8.3 (3840x2160)	2.0x2.0 µm	yes	60
OS08A10 RGB+NIR	1/1.8" (7.2x5.3 mm)	8.3 (3840x2160)	2.0x2.0 µm	yes	60
OX08B4C	1/1.7" (7.5x5.5 mm)	8.3 (3840x2160)	2.1x2.1 µm	yes	36
OS04A10 RGB + NIR	1/1.8" (7.2x5.3 mm)	4.1 (2688x1520)	2.9x2.9 µm	yes	30
OX02A10	1/2" (6.6x4.4 mm)	1.7 (1820x940)	4.2x4.2 µm	yes	60
OS12D40	1/2.5" (5.7x4.28 mm)	11.3 (4512x2512)	1.4x1.4 µm		60
e2v EV76C570 2M Saphire	1/1.8" (7.2x5.3 mm)	1.9 (1600x1200)	4.5x4.5 µm		50
OS08A20 RGB+NIR	1/1.8" (7.2x5.3 mm)	8.3 (3840x2160)	2.0x2.0 µm		60
AR0821CS	1/1.7" (7.5x5.5 mm)	8.3 (3840x2160)	2.1x2.1 µm		60
AR0820AT	1/2" (6.6x4.4 mm)	8.3 (3840x2160)	2.1x2.1 µm		40
AR0521	1/2.5" (5.7x4.28 mm)	5.1 (2592x1944)	2.2x2.2 µm	yes	60
AR0522 RGB+NIR	1/2.5" (5.7x4.28 mm)	5.1 (2592x1944)	2.2x2.2 µm	yes	60

Since the availability of Sony sensors is significantly more difficult than that of On-Semi sensors, it was decided to go with On-Semi sensors, and AR0522 was chosen as the first preference, which provides a lower digital zoom capability, but for which it was possible to find suitable lenses with similar physical dimensions [10]. It turned out that M12-sized lenses are mostly made for low-resolution (2.1MP) security cameras, and many of them are not suitable for an 8MP sensor.

2.4 Visible light camera lens selection

Since the lens mount was chosen as M12, a selection was made from the range of lenses with different FOV (*Field of View*) that can be attached to it. The calculations are based on the drone's flight height of 250m and the camera angle of 50 degrees, so the distance to the object is 389m. Calculations are made for the sizes of the most common objects which the drone observes during operation. Table 4 gives a list of such objects.

Table 4. List of the most common objects and their sizes.

Object	Horizontal size	Vertical size
Human	50cm	150cm
Car	410cm	140cm
Minivan	455cm	240cm
Truck	900cm	300cm
Armoured vehicle	715cm	320cm

Two options for the lenses were evaluated:

- Option A with 44x33° FOV wide lens and 28x21° FOV narrow lens.
- Option B with 44x33° FOV wide lens and 9x6.8° FOV narrow lens.

Option A would offer a seamless zoom experience for the user. The way it would work out is that the wide lens would have 44x33° FOV in full resolution and after digital zoom to HD reoslution it would have 21x11.8° FOV. Once zoomed in even further the video would switch to the second camera with a 18x13.9° FOV lens. After digizoom to HD resolution the second camera would have 8.8x5° FOV. This would mean a total of 5X zoom.

Option B would offer 10X zoom although there is a jump in zoom from camera 1 to camera 2. Camera 1 would have the same wide lens 44x33° FOV and after digital zoom to HD level it would have 21x11.8° FOV. When zoomed even further there would be a jump to another camera with 9x6.8° FOV. After digital zoom to HD level the final FOV would be 4.4x2.5°. This arrangement would offer 10X zoom without any moving parts or zoom optics in the gimbal.

Table 5 shows calculations for different lenses and object sizes.

Table 5. Calculated object sizes in pixels with different lens and zoom levels.

	Option A and B (wide lens)				**Option A (narrow lens)**				**Option B (narrow lens)**			
	Minimum digital zoom		Maximum digital zoom		Minimum digital zoom		Maximum digital zoom		Minimum digital zoom		Maximum digital zoom	
	Horizontal	Vertical	Horizontal	Vertical	Horizontal	Vertical	Horizontal	Vertical	Horizontal	Vertical	Horizontal	Vertical
Lens angle	44°	33°	21°	11.8°	18°	13.9°	8.8°	5°	9°	6.8°	4.4°	2.5°
Camera FOV at target	314m	230m	144m	80m	123m	95m	60m	34m	61m	46m	30m	17m
Image resolution at target	12.1cm/px	11.8cm/px	5.5cm/px	4.1cm/px	4.7cm/px	4.9cm/px	2.3cm/px	1.7cm/px	2.4cm/px	2.4cm/px	1.2cm/px	0.9cm/px
Object size: human	4px	13px	9px	36px	11px	31px	22px	86px	21px	63px	43px	172px
Object size: car	34px	12px	74px	34px	86px	29px	178px	80px	174px	59px	356px	161px
Object size: minivan	38px	20px	82px	58px	96px	49px	198px	138px	193px	101px	396px	276px
Object size: truck	74px	25px	162px	73px	190px	62px	391px	172px	382px	126px	783px	344px
Object size: armoured vehicle	50px	27px	129px	78px	151px	66px	310px	184px	303px	135px	622px	367px

2.5 Processing system selection

The most important thing about a processing system is that it must be able to run the Linux operating system. Because current software is written for Linux. Linux is a freeware operating system suitable for embedded systems. On the other hand, it is difficult to find a microprocessor with very different camera interfaces. One way to interface the cameras is to use the help of the FPGA. FPGA gives the system great flexibility and it is possible to easily change cameras later. It can also be used to process video streams without wasting microprocessor resources.

In order to speed up the development of the system, it is reasonable to use a ready-made microprocessor module. The same has been done with the existing camera controller. It also eliminates the number of possible design errors.

In addition, since today there are SoCs with an integrated CPU (*Central Processing Unit*) and FPGA, it was decided to choose from such chips. In most cases, an ARM (*Advanced RISC Machines*) processor is used for the CPU. The most common and available SoCs with integrated CPU and FPGA are listed in Table 6.

Table 6. List of SoCs with the required functionality

SoC	Manufacturer	CPU	Logic Cells
Zynq 7000S [11]	Xilinx / AMD	Single-core ARM Cortex-A9, 766MHz	65K
Zynq 7000 [11]	Xilinx / AMD	Dual-core ARM Cortex-A9, 1GHz	350K
Zynq UltraScale+ [12]	Xilinx / AMD	Dual-core ARM Cortex-A53, 1.3GHz	600K
Cyclone V SoC FPGA [13]	Altera / Intel	Dual-core ARM Cortex-A9, 925MHz	301K

The Zynq Ultrascale+ is undoubtedly the most capable and includes the hardware H.265 video encoding capabilities, but it is very expensive for this solution. However, taking into account the large number of demo projects available online for the Zynq 7000, the decision was made to use the Zynq 7000 SoC, specifically the Zynq 7020. The Zynq 7020 is the most capable of this series and is also available at a relatively affordable price. Also, a large number of development boards have been made with the Zynq 7020 chip.

2.6 Development board selection

To validate and test the compatibility of the components, a prototype is built using a development board. In addition to the development board, a PCB is designed to connect the components together. The most common development kit is the Zybo Z7 by Digilent, Figure 3 [14].

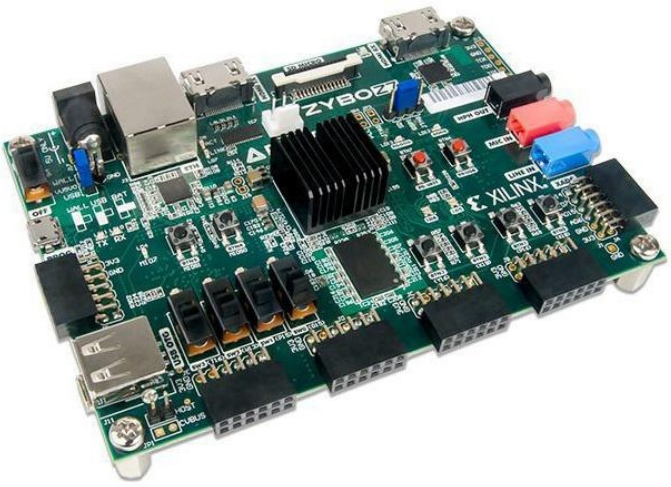


Figure 3. Digilent Zybo Z7 [14].

But research revealed that a kit with roughly the same capabilities is available for half the price. This is a Z-Turn board developed by MYIR, Figure 4 [15].

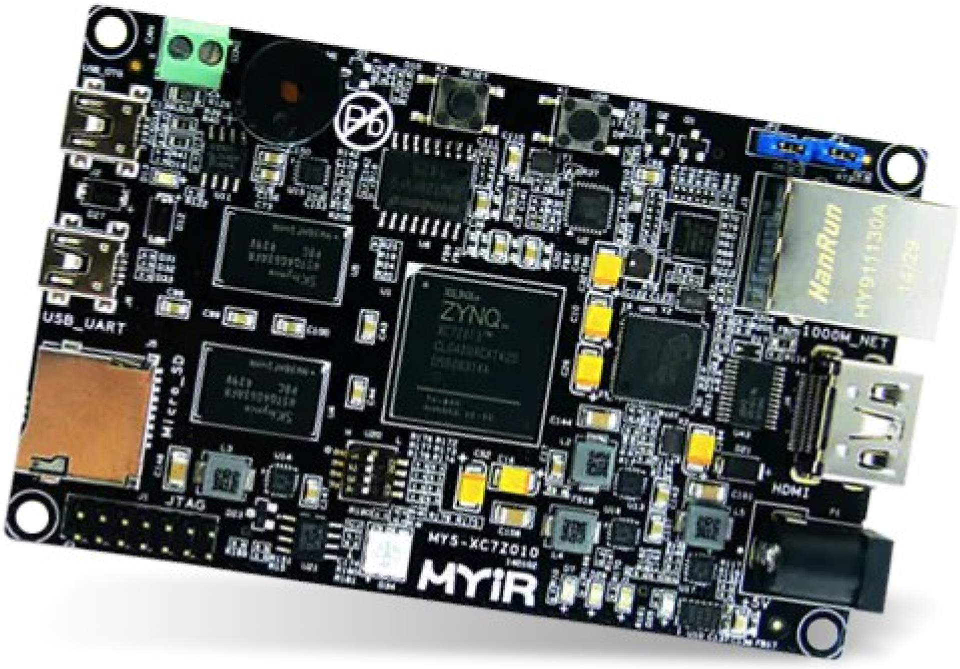


Figure 4. MYIR Z-Turn [15].

Because of the price, the Z-Turn Board was chosen for development. Both boards are also available for purchase at most common electronics stores. For example, like mouser [16]. The boards also have an HDMI (*High-Definition Multimedia Interface*) output, through which it is very convenient to monitor the image of the cameras.

In addition to the development board, the company ALINX produces a very small SOM (*System on a Module*) with the same SoC [17]. This is a potential candidate for a later finished product. The module is shown in Figure 5.

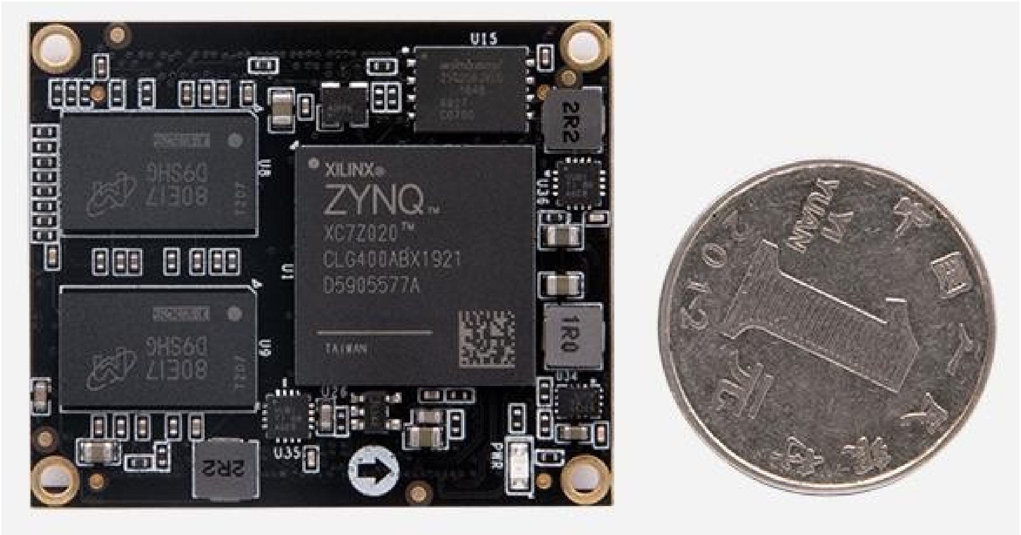


Figure 5. ALINX FPGA Core XC7Z020 SOM [17].

3 Electronic circuit design

When designing the electronic circuit, it was taken into account to use as many components as possible, which are planned to be included in the final product. Because of this, many components have an unnecessarily small package, considering that it is a prototype. The final product must be as small as possible and therefore it is necessary to use components of minimum size.

At the beginning, it was analyzed what voltages and interfaces the sensors and the development board need. Table 7 provides an overview of the components power requirements.

Table 7. Power consumption requirements of components.

Component	Voltage	Maximum Current	Description	IO Level(s)
Z-Turn Kit	5V	2A	Power Input	3.3V
AR0522	2.7V	112mA	Analog Voltage	1.8V, LVDS
	1.8V	0.12mA	I/O Digital Voltage	
	1.2V	171mA	Core Digital Voltage	
Xcore Micro III	5V	1.3A	Main Power	1.8V
	3.3V	250mA	VDD3.3	
	1.8V	150mA	VDD1.8	

3.1 Z-Turn IO banks configuration

The default configuration of the Z-Turn board is configured in such a way that all IO (*Input-Output*) banks use 3.3V voltage. But in our design we needed different voltage levels for different banks. More specifically, the system requires IO banks with voltages of 1.8V and 2.5V. 2.5V is needed for LVDS (*Low-Voltage Differential Signaling*). On the other hand, there are already chips on the Z-Turn board that are interfaced to some of the IOs. Looking at the schematic, it turned out that bank 35 is completely unused on the board, but the SiI9022A (HDMI transmitter) is connected to bank 34. This chip is needed to display HDMI video [18]. So it would be good if there was a way to keep this chip working. The datasheet revealed that the chip's IOs can work in both 3.3V and 1.8V modes. To do this, the IOVCC voltage must be adjusted and the IO_SEL pin must be set high. However, the IOVCC pins are already connected to the power rail of the bank 34, so it is only necessary to add a 0Ω resistor in place of R213 to set the IO_SEL high, as seen in Figure 6.

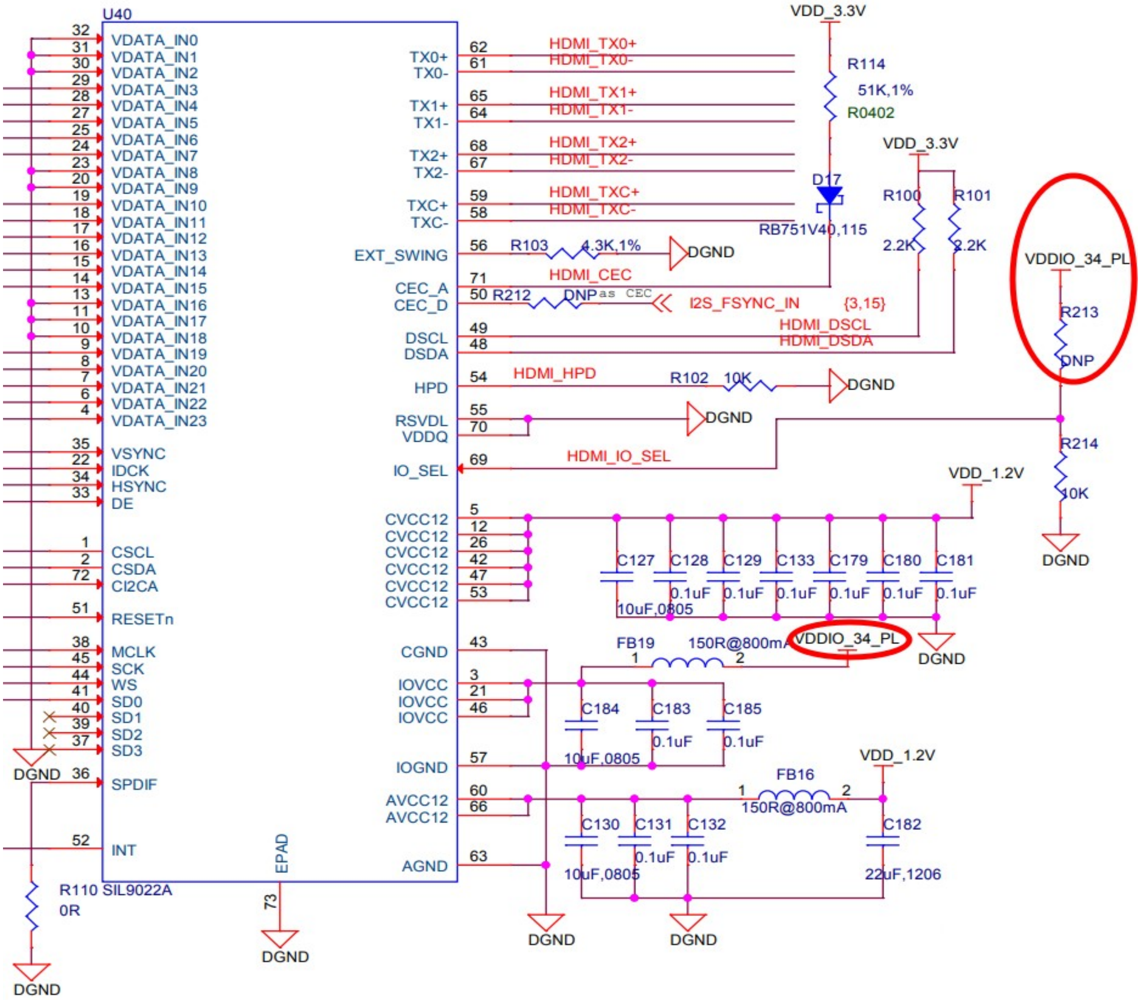


Figure 6. SiI9022A IO configuration.

Since it was possible to configure bank 34 to 1.8V level, bank 35 remained for LVDS and was configured to 2.5V level. However, in order to do this, it was first necessary to remove the inductors on the development board (FB5 and FB18), which provided 3.3V power to these banks [19]. In addition, it is possible to connect bank 34 to 1.8V power rail directly on the Z-Turn board using the unmounted place of FB25 as seen in Figure 7.

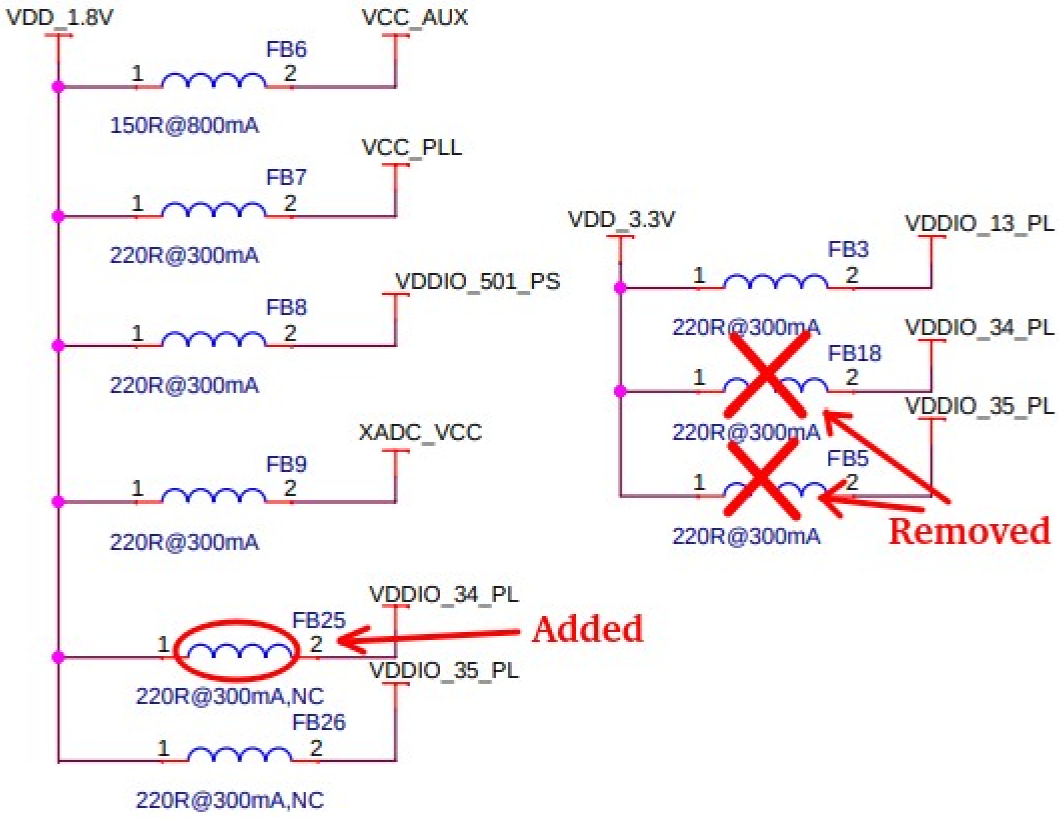


Figure 7. Z-Turn power configuration.

As a result, the three usable IO bank voltages of the Zynq 7000 are configured as follows:

- Bank 13 – 3.3V
- Bank 34 – 1.8V
- Bank 35 – 2.5V

3.2 Power supply design

Although the Z-Turn already has a 5V input and a power connector, it was decided to design a 12V input for the development board. This is because the infrared camera also needs 5V, but quite high currents. A separate power input allows powering the infrared camera as well. Voltage converters were selected from those available in the Mouser e-store. The prerequisites were the smallest possible packages and the necessary output currents. It was also important that the converters could be turned on and off with the control pin. The diagram of the power solution is shown in Figure 8.

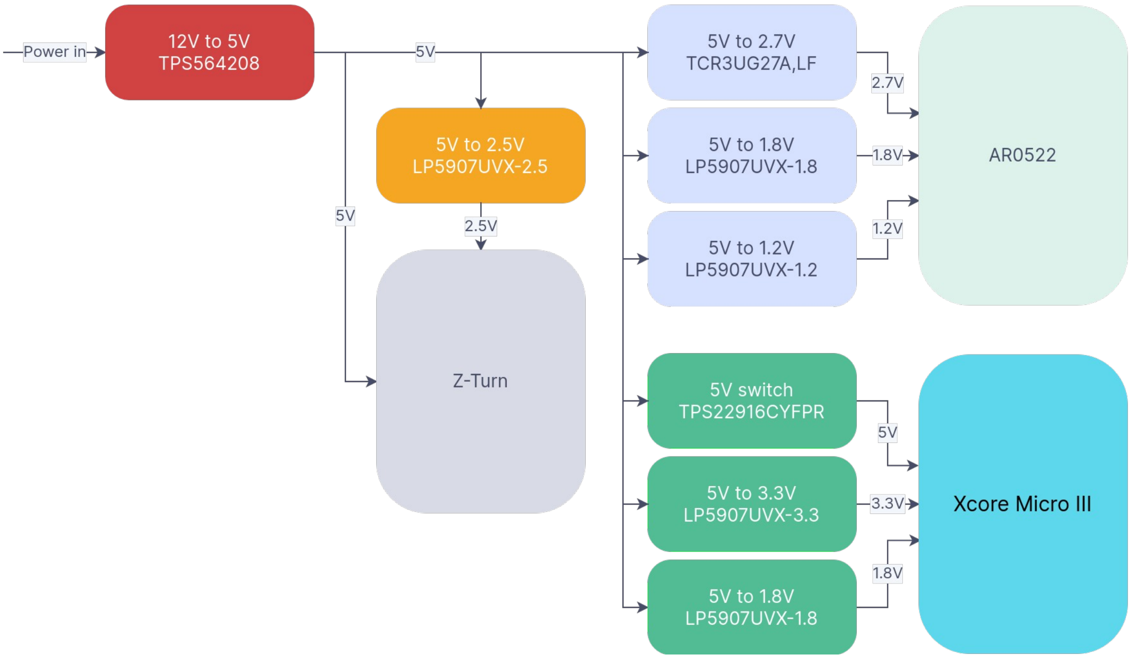


Figure 8. Power diagram.

The Texas Instruments chip TPS564208 was chosen as the 12V to 5V DC-DC converter because it was readily available in stores [20]. It has a 560kHz switching frequency and is capable of supplying 4A current. It's in a SOT-23 (*Small Outline Transistor*) package, which isn't the smallest, but that part of the schematic won't make it into the final solution. The final solution will run on 5V and the DC-DC converter is not needed.

The TPS22916 load switch from Texas Instruments is used for switching the 5 V power supply of the infrared camera. It is in BGA-4 (*Ball Grid Array*) package with only 0.8mm x 0.8mm in size. And it needs only two external capacitors to operate. It has an internal pulldown resistor on the ON pin, so it does not need to be added externally.

LDO (Low-Dropout) regulators were selected for the remaining power converters. They generate less noise than DC-DC converters and require fewer external parts. LDO regulators were selected in a BGA-4 package with an EN pin option, because it is necessary to turn them on and off with software. And if possible, LDO regulators with an internal pulldown resistor on the EN pin, are preferred.

Table 8 gives an overview of the voltage regulators and power switches used in the system. Since the voltage level required to activate the EN pin is mostly 1.2V on the regulators, any bank pin can be used to enable the regulator.

Table 8. Overview of used voltage regulators.

Regulator	**Output voltage**	**Output current**	**Consumer**	**Needed current**	**Package**	**EN voltage**	**EN Pulldown**
TPS564208	5V	4A	System	~4A	SOT-23	-	-
LP5907UVX-2.5/NOPB	2.5V	250mA	Bank 35	200mA	BGA	1.2V	YES
TCR3UG27A,LF	2.7V	300mA	AR0522	112mA	BGA	1.0V	YES
LP5907UVX-1.8/NOPB	1.8V	250mA	AR0522	0.12mA	BGA	1.2V	YES
LP5907UVX-1.2/NOPB	1.2V	250mA	AR0522	171mA	BGA	1.2V	YES
TPS22916CYFPR	5V	2A	IR Camera	1.3A	BGA	1.0V	YES
LP5907UVX-3.3/NOPB	3.3V	250mA	IR Camera	250mA	BGA	1.2V	YES
LP5907UVX-1.8/NOPB	1.8V	250mA	IR Camera	150mA	BGA	1.2V	YES

As the TPS56420 has adjustable output, it is necessary to use voltage divider from the output node to the VFB pin. Texas Instruments recommends to use 1% tolerance or better divider resistors. The output voltage can be calculated using the following formula:

$$V_{out} = 0.760 \times \left(1 + \frac{R1}{R2}\right)$$

However the datasheet already has a table with recommended resistor values. From there it turns out that for 5V output, it is necessary to use resistors 54.9kΩ and 10.0kΩ. The circuit can be seen in Figure 9.

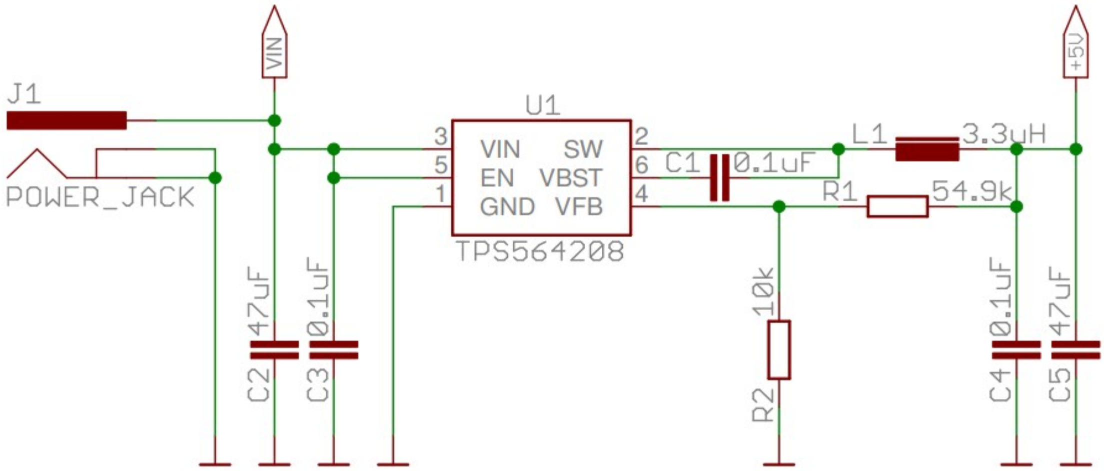


Figure 9. 5V power circuit

Since the PCB was hand soldered using a reflow oven, 0Ω resistors were added to the power inputs of the AR0522. At first, the resistors were left unsoldered, and after checking that the voltages were correct, the resistors were soldered. The reason was that the AR0522 is also soldered to the board and the sensor cannot be removed during testing. The Z-Turn board and IR camera are removable, so they didn't need such a solution. The need for such testing was due to the fact that it is very difficult to solder small BGA components by hand, and if a short circuit occurs under the component due to soldering, the wrong voltage can go to the sensor. The AR0522 power supply circuit and 0Ω resistors are illustrated in Figure 10.

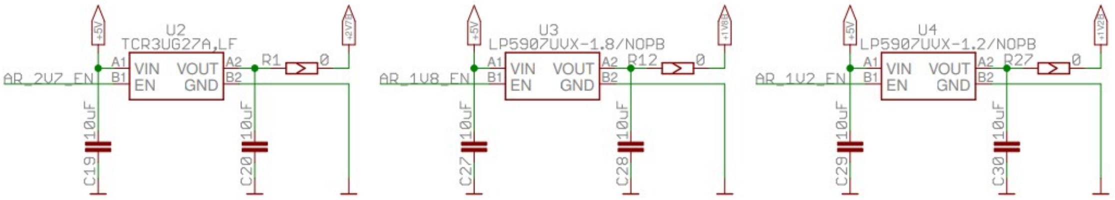


Figure 10. AR0522 power supply circuit

3.3 AR0522 interfacing

AR5022 sensor needs some IO pins with 1.8V level and MIPI CSI-2 D-PHY for interfacing. 1.8V IOs are available in bank 34. But D-PHY is not natively supported by Zynq 7020. It is possible to use external chips for D-PHY conversion, such as the MC20001, but they require more space on the PCB [24].

However, Xilinx application note points out, that D-PHY interface can be emulated using LVDS_25 and 1.8V IO using external resistors [25]. The minimum voltage of Zynq 7020 LVDS_25 interface can be 0.7V and the maximum voltage can be 1.675V. Therefore, this interface can receive D-PHY high speed data, which has an average voltage of 1.2V. Regarding this, the supply voltage of bank 35 must be set to 2.5V.

The specified level for low-power single-ended IO with D-PHY is also 1.2V. But the Zynq 7020 can receive this data with 1.8V IO if the IO is configured in HSUL_12 mode and the corresponding IO bank VREF is set to 0.6V. Figure 11 shows the Zynq 7020 compatible D-PHY receiver schematic.

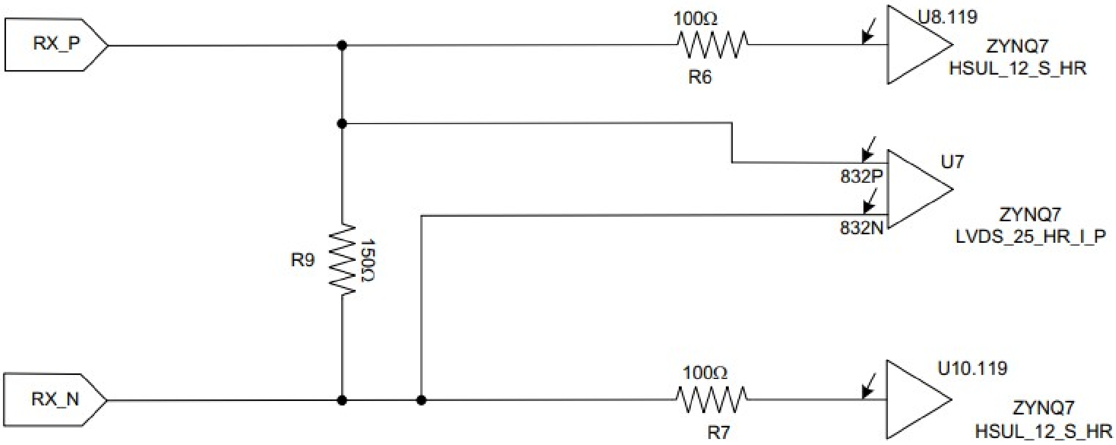


Figure 11. FPGA Compatible D-PHY Receiver.

Such a circuit had to be used for each D-PHY data pair. In addition, clock signal pair must be connected to the Zynq 7020 pins that have clocking capability [26].

The sensor must be fed with a 27MHz clock. This is easily done using FPGA. The sensor is configured via the standard I^{2}C protocol [27]. The recommended power−up sequence for the AR0522 is shown in Figure 12, where t1 = 1ms and each subsequent t is 1ms longer than the previous one.

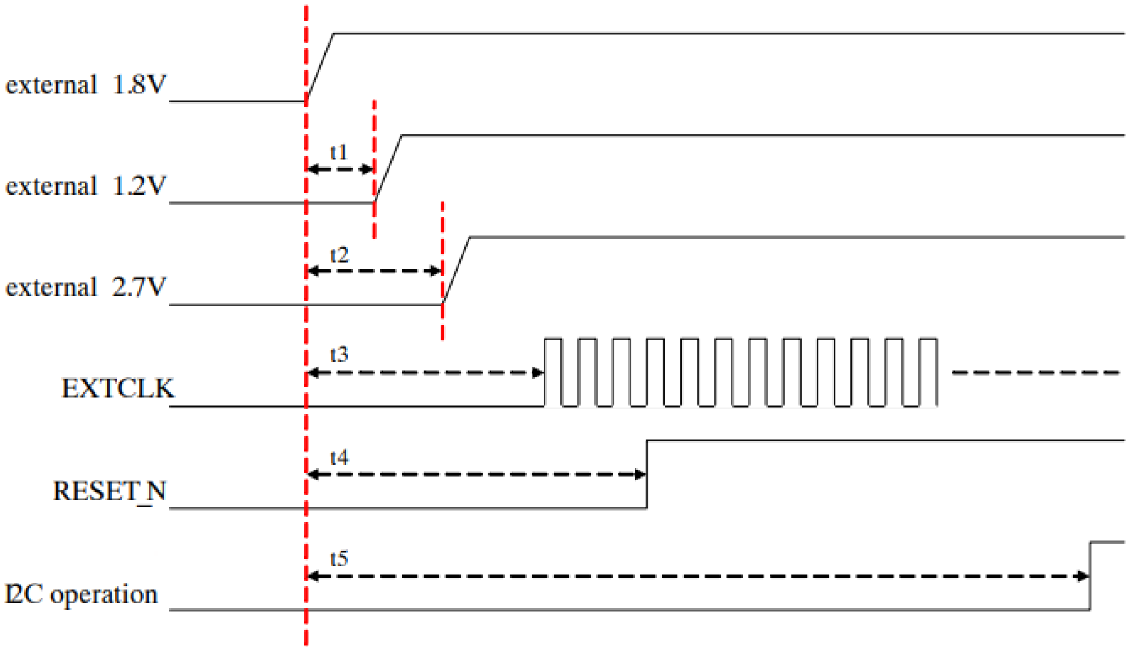


Figure 12. AR0522 power-on sequence.

3.4 Xcore Micro III interfacing

All voltage levels of the IO pins of the camera are 1.8V. Thus, the IO pins of the infrared camera are connected to bank 34. The camera uses a standard parallel camera interface, so no special connection circuit is needed [28]. The camera needs 18.75MHz clock input. A standard UART (*Universal Asynchronous Receiver-Transmitter*) interface is used to configure the camera.

The camera uses a pluggable connection to connect to the PCB. The Hirose 70PIN 0.4mm pitch connector named DF40C-70DP-0.4V(51) is used on the camera. The socket DF40HC(3.0)-70DS-0.4V(51) is compatible with this connector [29]. In addition, the camera can be supported with two (or four) M1.5 screws.

When turning on the camera, the correct sequence of the different voltages must be ensured. Power-on timing is shown in Figure 13.

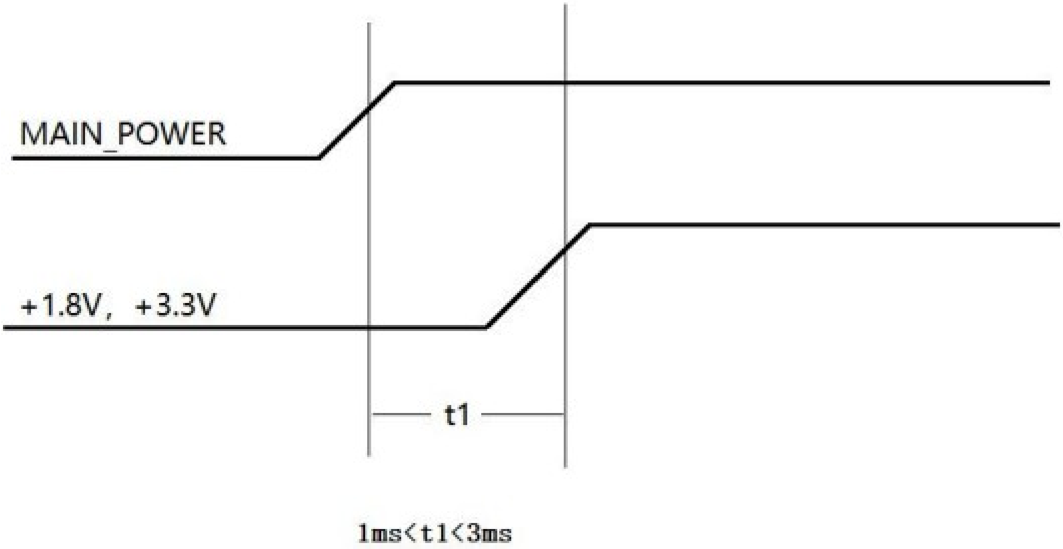


Figure 13. Xcore Micro III power-on sequence.

3.5 Additional interfaces

Since there are enough IO pins, two more connectors were added to the circuit, where different external cameras can be connected for testing purposes. The first one is a 2.54mm pitch double row female header, where can be connected OV7670 and similar camera modules. The IO voltage level of these modules is 3.3V, and therefore this connector is connected to bank 13. The protocol used is a parallel camera interface. A picture of such a camera module is shown in Figure 14 [30].

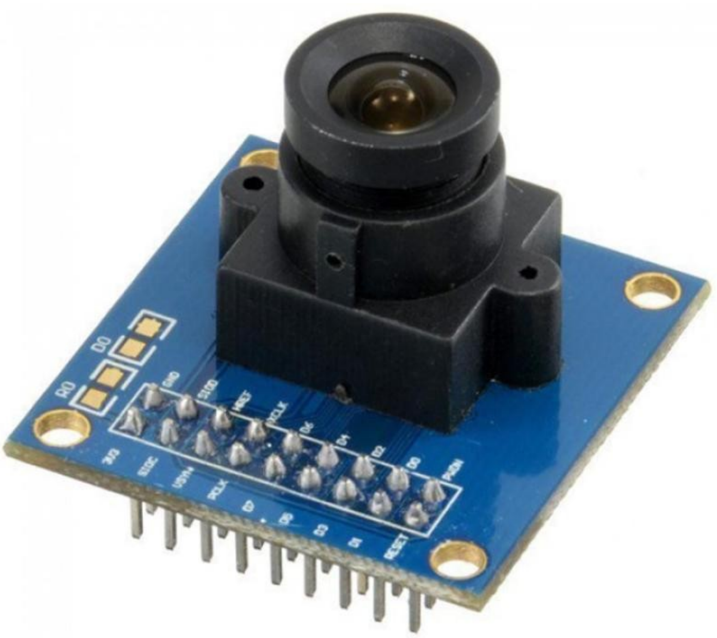


Figure 14. Arducam OV7670 Camera Module [30].

Another connector that was added to the circuit is a Raspberry PI style MIPI CSI-2 connector. It is an FFC (*Flat Flex Cable*) connector of the type SFW15R-2STE1LF by Amphenol [31]. It contains a D-PHY clock pair, two data pairs, I^{2}C and 3.3V power [32]. Again, for this MIPI CSI-2 differential pairs, D-PHY was made with resistors and both bank 34 and bank 35 IO pins were used. The camera connected to the Raspberry PI can be seen in Figure 15.

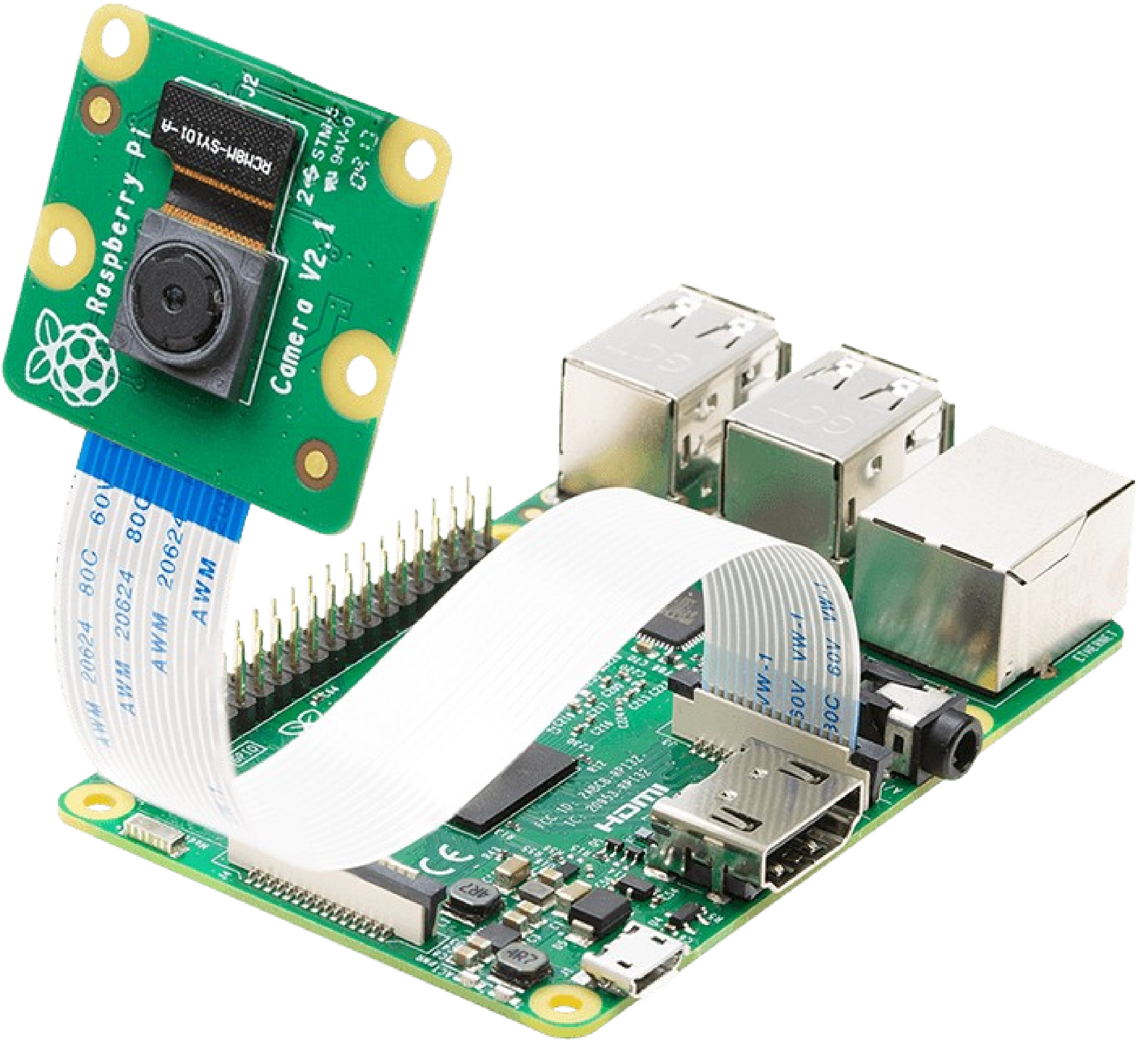


Figure 15. The camera connected to the Raspberry PI [32].

In addition to the two camera connectors, two slide switches with pullup resistors were added to the IO pins. The switches are of type JS202011JAQN by C&K Components [33]. The purpose of the switches are for choosing which camera stream to display via HDMI to make it more convenient to test the system.

A JTAG (*Joint Test Action Group*) connector is also provided for debugging purposes.

All other necessary interfaces are already exposed on the Z-Turn board.

4 PCB design

Since many devices on the board require quite a lot of power, there is a separate layer for 5V power and ground. The necessary voltages for different cameras are converted directly from 5V near the camera using LDOs. PCB with total of 4 layers can be routed easily because the FPGA IOs give a certain flexibility. And a 4-layer PCB is not excessively expensive. The company JLCPCB was used to produce the boards [34]. In addition to PCBs, a stencil was ordered for applying solder paste to use reflow soldering. As many components as possible were placed on the top side of PCB to simplify soldering.

4.1 AR0522 sensor mounting considerations

The most important thing about the AR0522 sensor is that it needs a decoupling capacitors near each power pin. However, it is not possible to place them near the sensor on the top side of PCB, because the lens holder is mounted on top of the sensor. Therefore, these capacitors are one of the few components placed on the bottom side. The lens mount is made of plastic, and all it needs are two screw holes and space around the sensor. The location of the lens mount is also marked on the silkscreen, as shown in Figure 16.

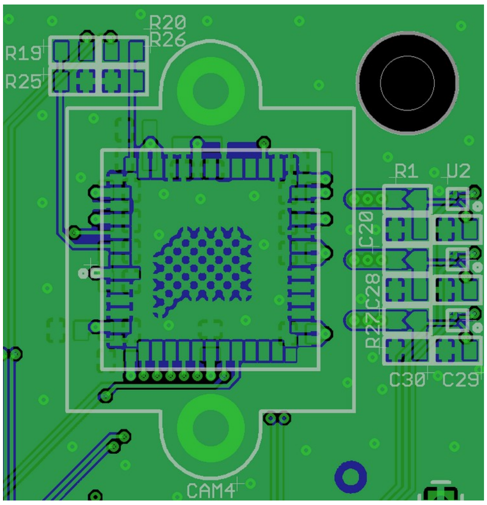


Figure 16. AR0522 sensor placement.

4.2 Xcore Micro III mounting considerations

Infrared camera decoupling capacitors are also placed on bottom side. The reason is the same as for the AR0522 sensor - the camera body with lenses takes up all the space on the top side. But the body of the IR camera is made of metal, and the places where it touches the PCB should not have tracks on the top side, otherwise there is a risk of short circuit. That is why vias are used and the necessary tracks run along the inner PCB layers. Also, the data sheet does not say whether the camera body is connected to ground or not, so the copper was completely removed from the contact point. Figure 17 shows the layout of the IR camera on the PCB.

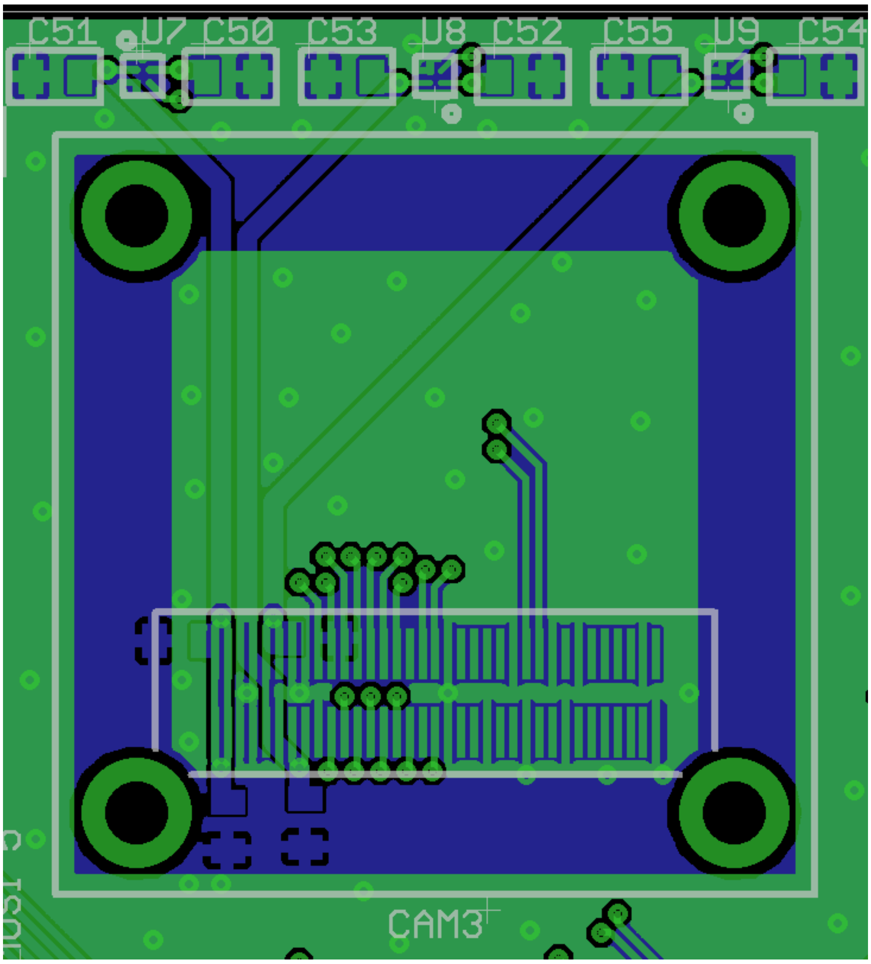


Figure 17. IR camera placement.

4.3 Placement of other components

The components were placed in such a way that the connections between them were as short as possible. The input DC-DC converter is placed on the other side of the Z-Turn board, away from the cameras, to avoid unwanted noise. All places where there are no tracks are filled with ground planes. Ground planes of different layers are connected by stitching vias. Since the power input uses a standard 5.5mm x 2.1mm barrel plug for 12V, but the Z-Turn uses the same plug for 5V, the Z-Turn socket is covered with tape to prevent accidental plugging of the 12V to the Z-Turn power input. Figure 18 provides an overview of the layout of the components.

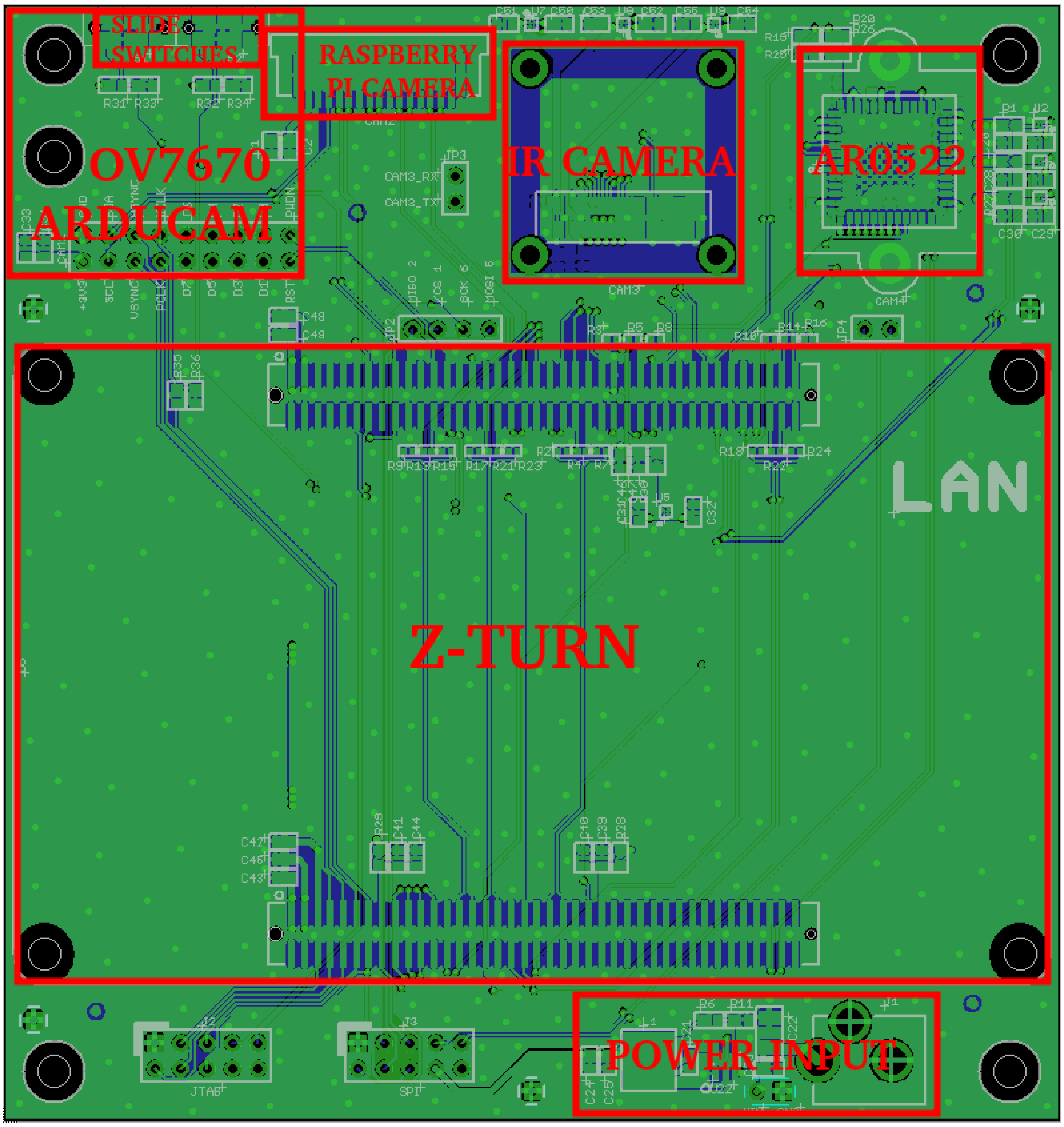


Figure 18. Component placement.

The assembled PCB with all possible cameras is shown in Figure 19.

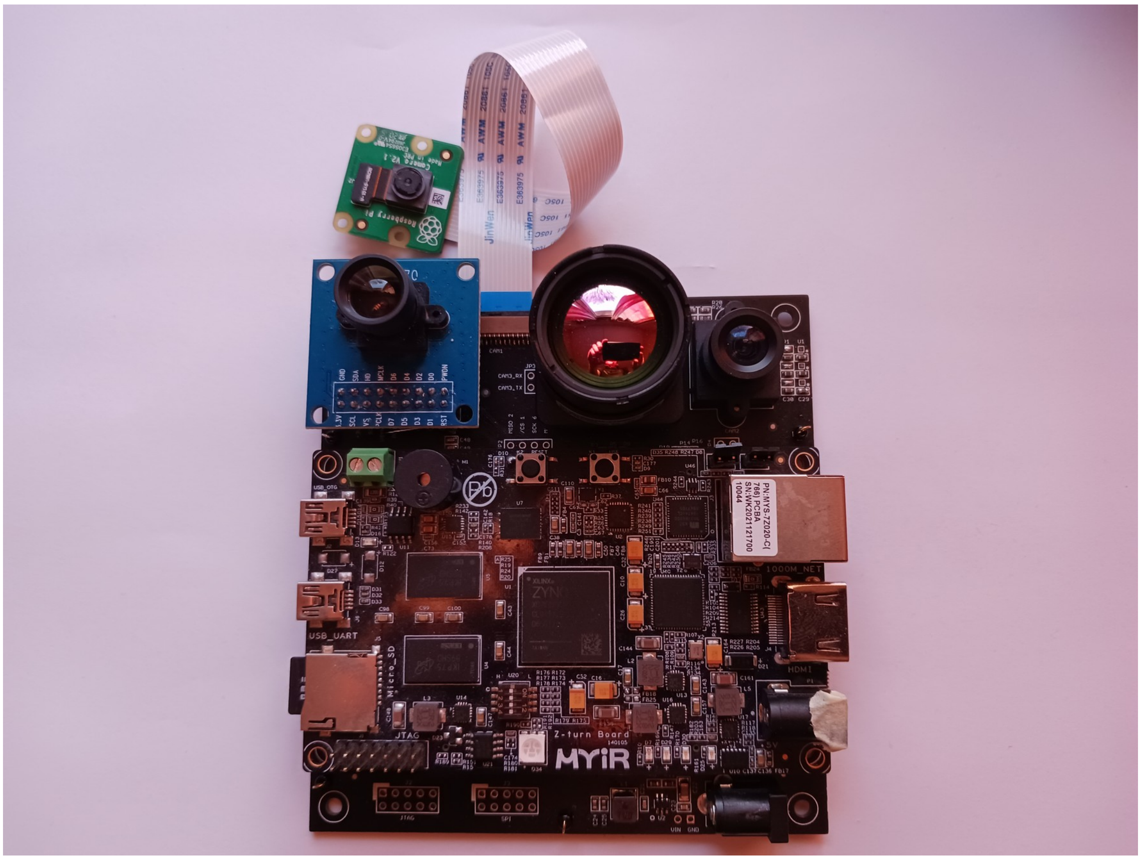


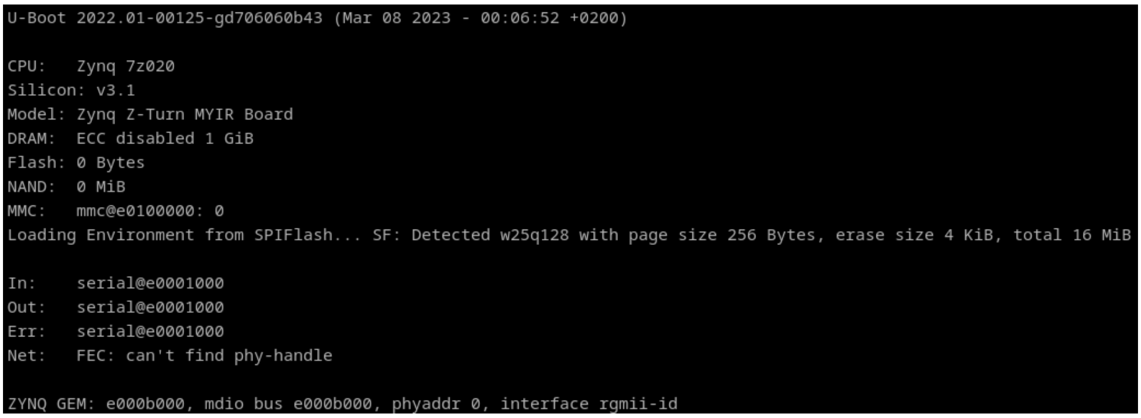
Figure 19. Assembled PCB.

5 Operating system

Due to the prerequisites, it had to be possible to install the Linux operating system on the processing system. The Z-Turn can boot from SPI (*Serial Peripheral Interface*) NOR flash on its PCB or from an external SD (*Secure Digital*) card. The booting source is set with 2.54mm pitch jumpers on the Z-Turn board. In order to load the Linux kernel, it is necessary to use a bootloader. A bootloader is a specific little piece of software built to support the corresponding hardware, which in turn can load a more generic Linux kernel.

5.1 U-Boot

The bootloader used in the system is U-Boot. U-Boot is an opensource bootloader for embedded systems [35]. Xilinx has updated the U-Boot source code by adding the necessary drivers for its SoCs, so the Xilinx U-Boot source code was used [36]. Before using an U-Boot, it was necessary to configure it for the desired system. Since it is possible to boot from both SPI flash and from SD card, these configurations are different. For development purposes, the U-Boot was configured to boot from SD card, because then it is easier to change operating system images. But SPI flash will be used in the final product, and U-Boot support for it was also tested during this project. Figure 20 shows U-Boot starting from SPI flash.



```
U-Boot 2022.01-00125-gd706060b43 (Mar 08 2023 - 00:06:52 +0200)

CPU:   Zynq 7z020
Silicon: v3.1
Model: Zynq Z-Turn MYIR Board
DRAM:  ECC disabled 1 GiB
Flash: 0 Bytes
NAND:  0 MiB
MMC:   mmc@e0100000: 0
Loading Environment from SPIFlash... SF: Detected w25q128 with page size 256 Bytes, erase size 4 KiB, total 16 MiB

In:    serial@e0001000
Out:   serial@e0001000
Err:   serial@e0001000
Net:   FEC: can't find phy-handle

ZYNQ GEM: e000b000, mdio bus e000b000, phyaddr 0, interface rgmii-id
```

Figure 20. SPI boot using U-Boot.

When using an SD card, the first partition of the SD card must be formatted as FAT (*File Allocation Table*), and the U-Boot binary must be copied to the root directory named “BOOT.bin”. U-Boot tries to load an operating system image named "image.ub" from the root directory.

In SPI flash, U-Boot is split into two binaries. The first so-called SPL (*Secondary Program Loader*) must be flashed to address 0x0. The second, the U-Boot main binary, must be flashed to address 0x100000.

U-Boot prepares the D-RAM (*Dynamic Random-Access Memory*), loads operating system kernel, devicetree, ramdisk and is also able to configure an FPGA with the bitstream. To build U-Boot for Z-Turn, the necessary commands are in Figure 21.

```
make ARCH=arm CROSS_COMPILE=arm-linux-gnueabi- distclean
make ARCH=arm CROSS_COMPILE=arm-linux-gnueabi- xilinx_zynq_virt_defconfig
make ARCH=arm CROSS_COMPILE=arm-linux-gnueabi- menuconfig
make ARCH=arm CROSS_COMPILE=arm-linux-gnueabi- DEVICE_TREE="zynq-zturn"
```

Figure 21. U-Boot building commands.

5.2 Operating system image

FIT (*Flattened Image Tree*) image format was used as the operating system image [37]. The advantages over this image format are that the entire operating system can be made to exist in one file. In this way, it is very convenient to quickly replace the entire operating system. This is also an advantage when installing the final product.

The Linux kernel, device tree, ramdisk and FPGA bitstream are packed into FIT image file. This was done with a utility called “mkimage”, which is included in the U-Boot sources. The utility uses a given formatting file called ITS (*Image Tree Source*), and generates an operating system image ITB (*Image Tree Blob*) [38].

Like with U-Boot, Xilinx has also developed the necessary Linux kernel drivers. The corresponding source code is available on the Xilinx repository [39]. The commands needed to build the Linux kernel are shown in Figure 22. When configuring the kernel, support for "FPGA Configuration Framework" must be enabled. This is for loading the FPGA bitstream in Linux userspace.

```
make ARCH=arm CROSS_COMPILE=arm-none-eabi- distclean
make ARCH=arm CROSS_COMPILE=arm-none-eabi- xilinx_zynq_defconfig
make ARCH=arm CROSS_COMPILE=arm-none-eabi- menuconfig
make ARCH=arm CROSS_COMPILE=arm-none-eabi-
```

Figure 22. Linux kernel building commands.

A ramdisk is also included in the FIT image. It is planned that the entire system will only use a ramdisk. The present system requires only a few executables running, and they can be embedded in a ramdisk without making it excessively large. During project development, a program named "init" was written in C language. It is a program that is started by the Linux kernel. And all other programs are started by the init program. For example, init also starts a shell from busybox that uses an UART for its input and output [40]. In addition, a program called dropbear is also used to create an SSH (*Secure Shell*) server that allows remotely logging into the Linux over the network [41]. The flow of booting the entire system can be seen in Figure 23.

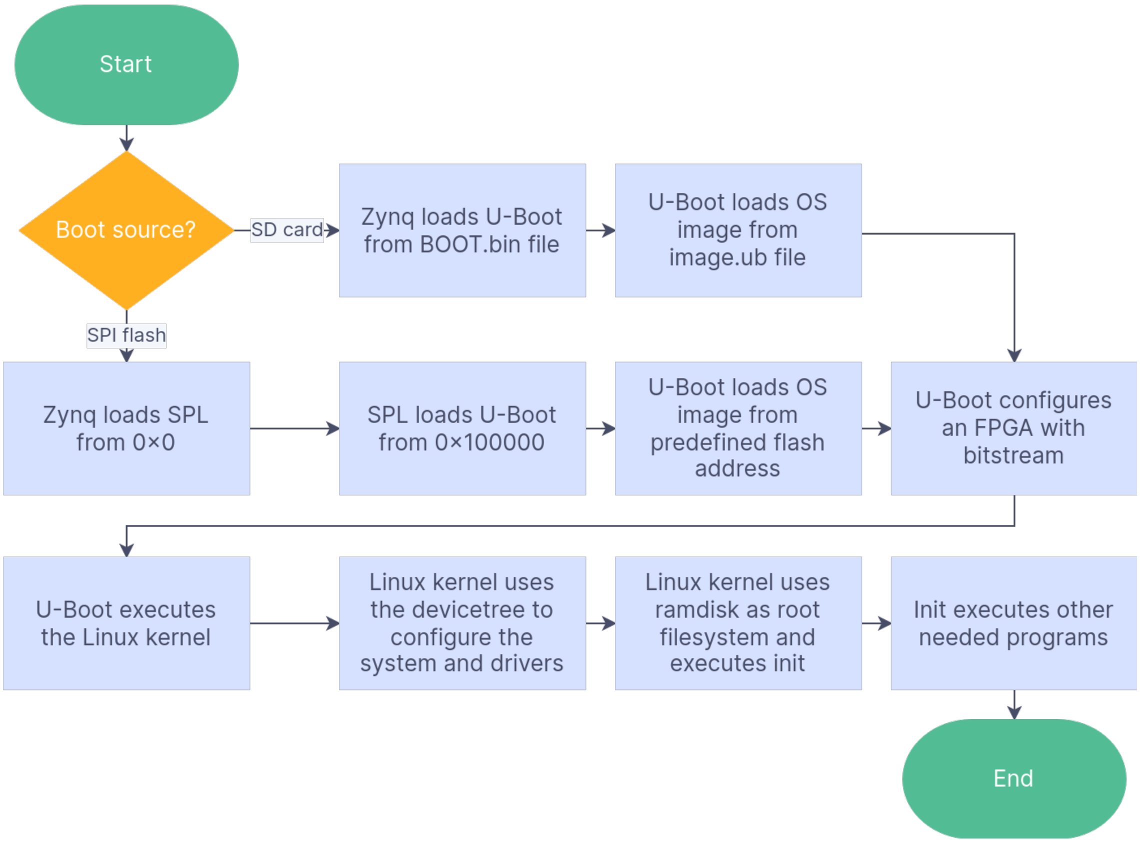

Start
Boot source?
SD card
SPI flash
Zynq loads U-Boot from BOOT.bin file
U-Boot loads OS image from image.ub file
Zynq loads SPL from 0×0
SPL loads U-Boot from 0×100000
U-Boot loads OS image from predefined flash address
U-Boot configures an FPGA with bitstream
U-Boot executes the Linux kernel
Linux kernel uses the devicetree to configure the system and drivers
Linux kernel uses ramdisk as root filesystem and executes init
Init executes other needed programs
End

Figure 23. Boot flow of Zynq SoC.

6 FPGA design

In order to interface the camera sensors with the processing system, it is necessary to implement a video processing pipeline in HDL for FPGA. Zynq ARM cores have only a few hardware peripheral interfaces, like UART, SPI and I^{2}C. The more complex design is intended to be done inside the FPGA. Xilinx Vivado software version 2022.2 was used in the design, as it was the most recent version when the project started [42]. The block design method was used, where existing IP (*Intellectual Property*) core blocks can be easily connected. In order to process video streams, the streams coming from the cameras are first converted to AXI (*Advanced eXtensible Interface*) format, more precisely - to the AXI4-Stream format. Figure 24 gives a simplified overview of video pipeline and HDL components, which was done in order to validate the operation of the cameras.

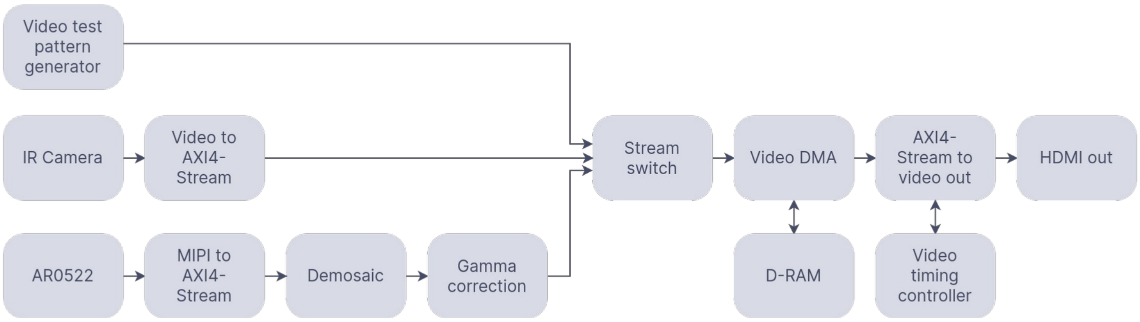


Figure 24. A simplified overview of the video pipeline.

The “Video test pattern generator” is not necessarily needed, but it gives an opportunity to validate the operation of different parts of the system [43].

6.1 AR0522 HDL interfacing

The AR0522 sensor uses the MIPI protocol, so the Xilinx IP core called “MIPI CSI-2 Receiver Subsystem” is suitable for its interfacing [44]. However, the sensor outputs a monochrome stream, seen through a Bayer filter inside the sensor, so it is necessary to process the frames to get a color image [45]. In addition, the resulting video also needs gamma correction [46]. Both of these operations can be done in HDL and Xilinx offers IP cores called “Sensor demosaic” and “Gamma LUT” (*Look-up Table*) [47][48]. The HDL interface required for interfacing the AR0522 sensor is illustrated in Figure 25.

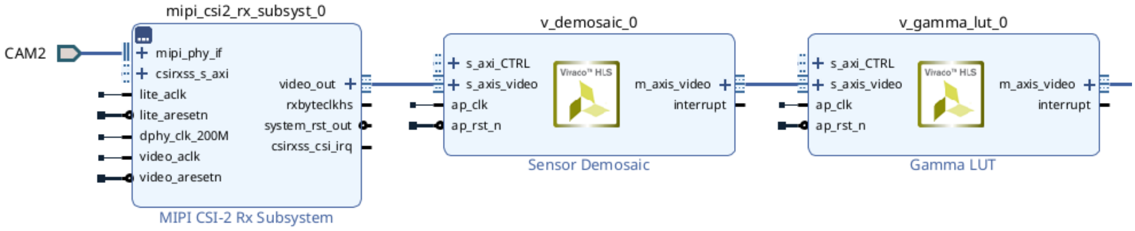


Figure 25. AR0522 sensor interface in HDL.

6.2 Xcore Micro III HDL interfacing

Xcore Micro III camera datasheet states that it uses a standard parallel camera interface, which uses pixel clock, horizontal sync and vertical sync in addition to data signals. Therefore, “Video-In to AXI4-Stream” IP core component from Xilinx should be suitable for converting the camera stream [49]. But in reality there are several variations of the protocol. For example, it is not defined whether sync signals are active high or active low. In addition, it is possible to use blanking signals instead of sync signals. Xilinx's corresponding IP core allows different configurations to be used. As a result of experimenting, it turned out that the IR camera uses inverted blanking signals. Therefore, the sync signals had to be connected to the corresponding blanking signal inputs of the IP core, using inverters. The final signal connections for the IR camera can be seen in Figure 26.

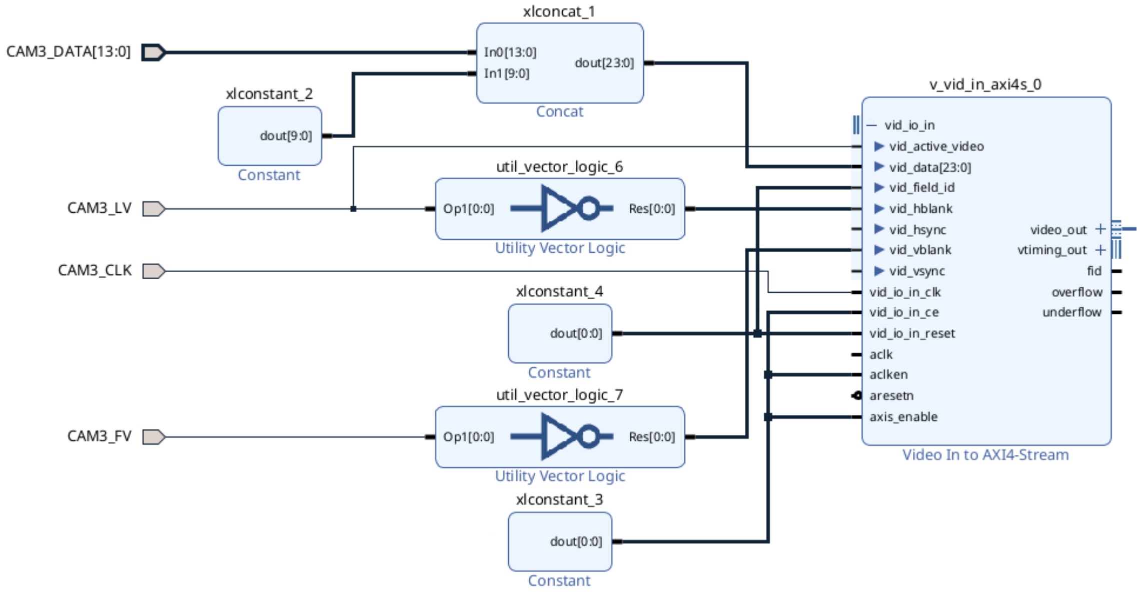


Figure 26. IR camera connections in HDL.

The IR camera transmits an image that does not need to be processed separately, so the resulting stream is already suitable for the further pipeline.

6.3 Other HDL components

Different video streams are sent through the “AXI4-Stream switch” IP core [50]. This IP core allows only one input stream to pass through it. The input can be selected either by software or by using GPIO (*General Purpose Input-Output*) signals. Although it would be possible to connect external switches to these GPIO signals, in this project it was decided in favor of a software solution. The software checks the positions of the switches and sends the corresponding command to the IP core.

Next, the previously selected stream goes to the "Video DMA" (*Direct Memory Access*) IP core [51]. This IP core has several tasks. First, it allows the software to receive the video stream, since this IP core can write the video frames to memory without the participation of the CPU. Second, it allows to change the video frame rate. To change the frame rate, the IP core uses the so-called triple-buffering method. It works so that the IP core is configured with a memory of three frames. If the incoming stream is written to a frame, the outgoing stream is read from a different frame. There is always one free frame where to write the next incoming frame [52]. It is necessary to change the frame rate when displaying the video on an external display. Display screens require a standard video format with a fixed frame rate, but cameras and sensors, on the other hand, usually send data as fast as possible.

For the external display, the video is generated with two Xilinx IP cores named "AXI4-Stream to video out" and "Video timing controller" [53][54]. The first of them converts the AXI4 stream to parallel video format, and the second helps to generate sync signals with correct timings [55]. To make the video signals suitable for HDMI, Z-Turn PCB has SiI9022A HDMI transmitter with HDMI connector.

Table 9 shows the use of resources in final HDL design.

Table 9. Resource utilization in HDL design.

Resource	Utilization	Available	Utilization %
LUT	23766	53200	44.67
LUTRAM	1420	17400	8.16
FF	32013	106400	30.09
BRAM	136.5	140	97.5
DSP	29	220	13.18
IO	76	125	60.8
BUFG	7	32	21.88
MMCM	2	4	50.0

7 Software design

The bitstream developed in HDL will not work without software. All video pipeline components need to be configured and started via software. It is also necessary to switch on the sensors' power and configure the sensors' parameters. All the necessary software for this project was developed in such a way that it works in the Linux user-space. For the final product, it is necessary to develop Linux kernel drivers for the cameras, but to validate the operation of the cameras, user-space software is sufficient. The software must be able to communicate with different components using different interfaces. Table 10 shows the components and interfaces of the system that must be configured with the software [56][57][58][59].

Table 10. Software configurable system components.

Component	Interface	Linux device	Address
Slide switches	GPIO	/dev/gpiochip0	-
Video test pattern generator	Memory mapped	/dev/mem	0x43C00000
Sensor power switches	GPIO	/dev/gpiochip0	-
AR0522 configuration	I^2C	/dev/i2c-0	0x36
MIPI CSI-2 RX	Memory mapped	/dev/mem	0x43C10000
Sensor demosaic	Memory mapped	/dev/mem	0x43C50000
Gamma LUT	Memory mapped	/dev/mem	0x43C60000
IR camera configuration	UART	/dev/ttyPS1	-
Video DMA	Memory mapped	/dev/mem	0x43000000
SiI9022A configuration	I^2C	/dev/i2c-0	0x3B

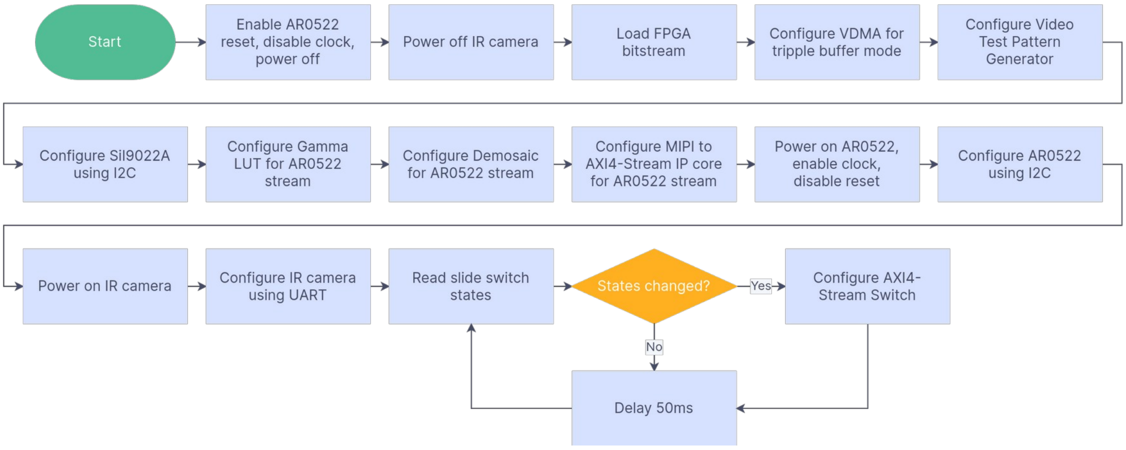
The configuration of most IP cores and devices is described in their datasheets. However the SiI9022A datasheet lacks register information. Therefore, open source code was used to configure the SiI9022A [60]. The necessary information is also available in the linux kernel code [61].

Although the initial bitstream is loaded into the FPGA by U-Boot, the Linux kernel driver was used during development to configure the FPGA [62]. Example commands for loading a bitstream are shown in Figure 27 [63].

```
echo 0 > /sys/class/fpga_manager/fpga0/flags
mkdir -p /lib/firmware
cp /media/design_1_wrapper.bit.bin /lib/firmware/
echo design_1_wrapper.bit.bin > /sys/class/fpga_manager/fpga0/firmware
```

Figure 27. Steps for programming the Full Bitstream [63].

To setup the IP cores and camera sensors, a program was written in C language, which configures all components of the video pipeline one by one. When the system components are configured, the program remains in a loop and checks the slide switch states and, according to their position, gives a command to the AXI4-Stream Switch, which in turn lets the stream of the selected camera pass through it. The program code is started by the "init" process when Linux is booted. The steps for initializing video pipeline components are illustrated in Figure 28.



Start
Enable AR0522 reset, disable clock, power off
Power off IR camera
Load FPGA bitstream
Configure VDMA for tripple buffer mode
Configure Video Test Pattern Generator
Configure Sil9022A using I2C
Configure Gamma LUT for AR0522 stream
Configure Demosaic for AR0522 stream
Configure MIPI to AXI4-Stream IP core for AR0522 stream
Power on AR0522, enable clock, disable reset
Configure AR0522 using I2C
Power on IR camera
Configure IR camera using UART
Read slide switch states
States changed?
Yes
Configure AXI4-Stream Switch
No
Delay 50ms

Figure 28. Software diagram for configuring video pipeline components.

8 Testing

Although the validation of the system took place gradually in parallel with the development, more comprehensive tests were also performed to verify the result. Below are some test scenarios that are important for this system and which the new design successfully passed.

- Power on/off test. The system must always boot. Even if the power was cut unexpectedly beforehand. The system was left running for a few days, and the power was switched on and off regularly with the help of the auxiliary controller. The system booted every time and the camera streams started working.

- Keeping the system running. Similar to the previous test, the system was kept running for a few days. The cameras didn't stop streaming (on the previous system, the USB connection of the IR camera regularly stopped working).

- Working in the cold. The system was tested at -10°C. Although the Z-Turn development kit is not designed to operate below 0°C, there were no problems due to slight warming of the Zynq 7000 during operation. This test was needed especially for cameras. The final product uses industrial grade components that allow a temperature range of -40°C to 85°C.

- Possible sensor alternatives. Since the Raspberry PI camera connector and Arduino connector were added to the hardware PCB, these cameras were also tested. Additional HDL logic and additional piece of software were developed for these sensors. Thanks to the capabilities of the FPGA, there were no difficulties in interfacing them.

9 Conclusion and future work

The thesis focused on the design of camera hardware and software interfaces. During the work, technical solutions were found, which were a prerequisite for creating the final system. The system proved to be stable, and this design has great potential for additional functionality or adding alternative camera sensors.

The author will continue with this work, and the final system will be designed as the next step.

The next step would be to add the necessary camera drivers to the Linux kernel. Once the camera drivers are added to the kernel, the current camera system software can be run on the microprocessor. If everything works as it should, then move on to the final hardware design, which includes two visible light cameras and which is as compact as possible. The ALINX module, which is very small, was already ordered for this purpose.

The controller of the gimbal motors can be designed for this system using, for example, FPGA logic, which controls the motors according to the position of the IMU sensor. Alternatively, a RiscV or ARM Cortex-M processor can be created inside the FPGA on which real-time logic can be run [64][65].

Depending on the FPGA resources, it may be possible to add IP core that encodes the video stream in H.264 or H.265 format [66].

It is also possible to use FPGA for video processing, for example picture-in-picture streaming from multiple cameras, etc.

References

[1] Krattworks | Autonomous surveillance UAV technology. Available at https://www.krattworks.com/ Accessed 12.03.2023

[2] Innovatsioonifond investeerib targa linna lahendustesse 100 000 eurot. Available at https://pealinn.ee/2022/05/19/innovatsioonifond-investeerib-targa-linna-lahendustesse-100-000-eurot/ Accessed 12.03.2023

[3] Oclea™ System on Modules (µSoM). Available at https://oclea.com/products/system-on-modules/ Accessed 22.04.2023

[4] GSG 201, Gyro-stabilized two-axis gimbal for day and night surveillance, with Laser Rangefinder. Available at https://www.uavos.com/products/gimbals/gsg-201-gimbal/ Accessed 12.03.2023

[5] USG-400 EO/IR/LRF gimbal. Available at https://ukrspecsystems.com/drone-gimbals Accessed 12.03.2023

[6] Trillium HD25. Available at https://www.trilliumeng.com/hd25 Accessed 12.03.2023

[7] Foxtech Bi-Focus 10X Optical Zoom 320x240 IR Thermal Camera with 3-axis Gimbal. Available at https://www.foxtechfpv.com/foxtech-bi-focus-10x-optical-zoom-thermal-camera-3-axis-gimbal.html Accessed 12.03.2023

[8] Dione 640 CAM Series, Available at https://www.xenics.com/long-wave-infrared-imagers/dione-640-camera-series/ Accessed 12.03.2023

[9] MicroIII High Resolution Thermal Camera Module, Available at https://www.infiray.com/products/microiii-384t-640t-high-resolution-thermal-camera-module.html Accessed 12.03.2023

[10] Image Sensors | AR0522. Available at https://www.onsemi.com/products/sensors/image-sensors/ar0522 Accessed 02.04.2023

[11] SoCs with Hardware and Software Programmability. Available at https://www.xilinx.com/products/silicon-devices/soc/zynq-7000.html Accessed 09.04.2023

[12] Zynq™ UltraScale+™ MPSoC. Available at https://www.xilinx.com/products/silicon-devices/soc/zynq-ultrascale-mpsoc.html Accessed 09.04.2023

[13] Cyclone® V FPGA and SoC FPGA. Available at https://www.intel.com/content/www/us/en/products/details/fpga/cyclone/v.html Accessed 09.04.2023

[14] Zybo Z7. Available at https://digilent.com/reference/programmable-logic/zybo-z7/start Accessed 09.04.2023

[15] Z-turn Board. Available at https://www.myirtech.com/list.asp?id=502 Accessed 09.04.2023

[16] Mouser Electronics. Available at https://www.mouser.ee/ Accessed 09.04.2023

[17] Xilinx ZYNQ7000 ARM SOM FPGA Core XC7Z020. Available at https://alinx.com/en/detail/296 Accessed 09.04.2023

[18] SiI9022A/SiI9024A HDMI Transmitter. Available at https://datasheet.lcsc.com/lcsc/1912111437_Lattice-SiI9024ACNU_C369574.pdf Accessed 09.04.2023

[19] ZTurn and Active Test Board. Available at https://atlaswiki.lbl.gov/strips/powerboard/masstester/zturnnotes Accessed 09.04.2023

[20] TPS564208 4.5 V to 17 V input, 4 A output, synchronous step-down converter in FCCM mode. Available at https://www.ti.com/product/TPS564208 Accessed 09.04.2023

[21] TPS22916CYFPR 5.5-V, 2-A, 60-mΩ, 10-nA leakage load switch with output discharge. Available at https://www.ti.com/product/TPS22916 Accessed 09.04.2023

[22] LP5907 250-mA, low-noise, high-PSRR, ultra-low-dropout voltage regulator with low IQ and enable. Available at https://www.ti.com/product/LP5907 Accessed 09.04.2023

[23] TCR3UG series Ultra low quiescent current, Fast Load Transient 300 mA CMOS Low Dropout Regulator in ultra small package. Available at https://www.mouser.ee/datasheet/2/408/TCR3UG27A_datasheet_en_20220111-1289317.pdf Accessed 09.04.2023

[24] MC20001. Available at https://www.meticom.com/Products.html Accessed 09.04.2023

[25] D-PHY Solutions. Avilable at https://docs.xilinx.com/v/u/en-US/xapp894-d-phy-solutions Accessed 09.04.2023

[26] Zynq-7000 SoC Packaging and Pinout. Available at https://docs.xilinx.com/v/u/en-US/ug865-Zynq-7000-Pkg-Pinout Accessed 09.04.2023

[27] I2C. Available at https://learn.sparkfun.com/tutorials/i2c/all Accessed 09.04.2023

[28] Camera interface. Available at https://en.wikipedia.org/wiki/Camera_interface Accessed 09.04.2023

[29] 0.4mm Pitch, 1.5 to 4.0mm Height, Board-to-Board and Board-to-FPC Connectors. Available at https://www.mouser.ee/datasheet/2/185/DF40_Catalog_D31649_en-2301840.pdf Accessed 09.04.2023

[30] VGA OV7670 Camera Module I2C 640X480. Available at https://core-electronics.com.au/vga-ov7670-camera-module-i2c-640x480.html Accessed 10.04.2023

[31] 1.00mm Flex Connector, SFW-R series, 15 Position, Top side Contact, Side Entry Surface Mount ZIF Connector, Lead Free. Available at https://www.amphenol-cs.com/1-00mm-pitch-flex-connectors-sfw15r2ste1lf.html Accessed 10.04.2023

[32] Raspberry Pi MIPI CSI Camera Pinout. Available at https://www.arducam.com/raspberry-pi-camera-pinout/ Accessed 10.04.2023

[33] JS202011JAQN JS0 SERIES SUB-MINIATURE SLIDE SWITCH. Available at https://www.ckswitches.com/products/switches/product-details/Slide/JS/JS202011JAQN/ Accessed 10.04.2023

[34] Your PCB Partner – JLCPCB. Available at https://jlcpcb.com/ Accessed 10.04.2023

[35] The U-Boot Documentation. Available at https://u-boot.readthedocs.io/en/latest/ Accessed 16.04.2023

[36] The official Xilinx u-boot repository. Available at https://github.com/Xilinx/u-boot-xlnx Accessed 16.04.2023

[37] Flattened uImage Tree (FIT) Images. Available at https://www.thegoodpenguin.co.uk/blog/u-boot-fit-image-overview/ Accessed 16.04.2023

[38] U-Boot new uImage source file format. Available at https://github.com/Xilinx/u-boot-xlnx/blob/master/doc/uImage.FIT/source_file_format.txt Accessed 16.04.2023

[39] The official Linux kernel from Xilinx. Available at https://github.com/Xilinx/linux-xlnx Accessed 16.04.2023

[40] Busybox. Available at https://busybox.net/ Accessed 16.04.2023

[41] Dropbear SSH. Available at https://matt.ucc.asn.au/dropbear/dropbear.html Accessed 16.04.2023

[42] Vivado 2022.2 - Installation and Licensing. Available at https://www.xilinx.com/support/documentation-navigation/design-hubs/dh0013-vivado-installation-and-licensing-hub.html Accessed 16.04.2023

[43] Video Test Pattern Generator. Available at https://docs.xilinx.com/r/en-US/ug1449-multimedia/Video-Test-Pattern-Generator Accessed 16.04.2023

[44] MIPI CSI-2 Receiver Subsystem Product Guide. Available at https://docs.xilinx.com/r/en-US/pg232-mipi-csi2-rx Accessed 16.04.2023

[45] Bayer filter. Available at https://en.wikipedia.org/wiki/Bayer_filter Accessed 16.04.2023

[46] Gamma correction. Available at https://en.wikipedia.org/wiki/Gamma_correction Accessed 16.04.2023

[47] Sensor Demosaic v1.1 LogiCORE IP Product Guide. Available at https://docs.xilinx.com/r/en-US/pg286-v-demosaic Accessed 16.04.2023

[48] Gamma LUT. Available at https://docs.xilinx.com/r/en-US/ug1449-multimedia/Gamma-LUT Accessed 16.04.2023

[49] Video In to AXI4-Stream v5.0 LogiCORE IP Product Guide. Available at https://docs.xilinx.com/r/en-US/pg043_v_vid_in_axi4s Accessed 16.04.2023

[50] AXI4-Stream Switch. Available at https://docs.xilinx.com/r/en-US/pg085-axi4stream-infrastructure/AXI4-Stream-Switch Accessed 16.04.2023

[51] AXI Video Direct Memory Access v6.3 LogiCORE IP Product Guide. Available at https://docs.xilinx.com/r/en-US/pg020_axi_vdma Accessed 16.04.2023

[52] Video Series 24: Using the AXI VDMA in Triple Buffer Mode. Available at https://support.xilinx.com/s/article/938327?language=en_US Accessed 16.04.2023

[53] AXI4-Stream to Video Out v4.0. Available at https://docs.xilinx.com/v/u/en-US/pg044_v_axis_vid_out Accessed 16.04.2023

[54] Video Timing Controller. Available at https://docs.xilinx.com/r/en-US/pg350-v-hdmi-txss1/Video-Timing-Controller Accessed 16.04.2023

[55] VGA Signal Timing. Available at http://tinyvga.com/vga-timing Accessed 16.04.2023

[56] Control GPIO using the new Linux user space GPIO API. Available at https://blog.lxsang.me/post/id/33 Accessed 22.04.2023

[57] Implementing I2C device drivers in userspace. Available at https://docs.kernel.org/i2c/dev-interface.html Accessed 22.04.2023

[58] Directly Access Your Physical Memory (dev/mem). Available at https://bakhi.github.io/devmem/ Accessed 22.04.2023

[59] Linux Serial Ports Using C/C++. Available at https://blog.mbedded.ninja/programming/operating-systems/linux/linux-serial-ports-using-c-cpp/ Accessed 22.04.2023

[60] Z-Turn board HDMI out. Available at https://github.com/hauerdie/z-turn-board-hdmi-out Accessed 22.04.2023

[61] The Linux Kernel Archives. Available at https://www.kernel.org/ Accessed 22.04.2023

[62] FPGA Subsystem. Available at https://static.lwn.net/kerneldoc/driver-api/fpga/intro.html Accessed 22.04.2023

[63] Solution ZynqMP PL Programming. Available at https://xilinx-wiki.atlassian.net/wiki/spaces/A/pages/18841847/Solution+ZynqMP+PL+Programming Accessed 22.04.2023

[64] Booting up my first RISC-V core on an FPGA. Available at https://embeddedinn.xyz/articles/tutorial/booting-my-first-RISC-V-core-on-an-FPGA/ Accessed 22.04.2023

[65] Cortex-M3 DesignStart FPGA-Xilinx edition package. Available at https://developer.arm.com/documentation/101483/0000/introduction/cortex-m3-designstart-fpga-xilinx-edition-package Accessed 22.04.2023

[66] H.264 CODEC Overview. Available at https://www.a2etechnologies.com/h264-CODEC.html Accessed 22.04.2023

Appendix 1 – Non-exclusive licence for reproduction and publication of a graduation thesis[1]

I Veiko Rütter

1 Grant Tallinn University of Technology free licence (non-exclusive licence) for my thesis “Development of A Camera controller prototype for drones”, supervised by Peeter Ellervee

1.1 to be reproduced for the purposes of preservation and electronic publication of the graduation thesis, incl. to be entered in the digital collection of the library of Tallinn University of Technology until expiry of the term of copyright;

1.2 to be published via the web of Tallinn University of Technology, incl. to be entered in the digital collection of the library of Tallinn University of Technology until expiry of the term of copyright.

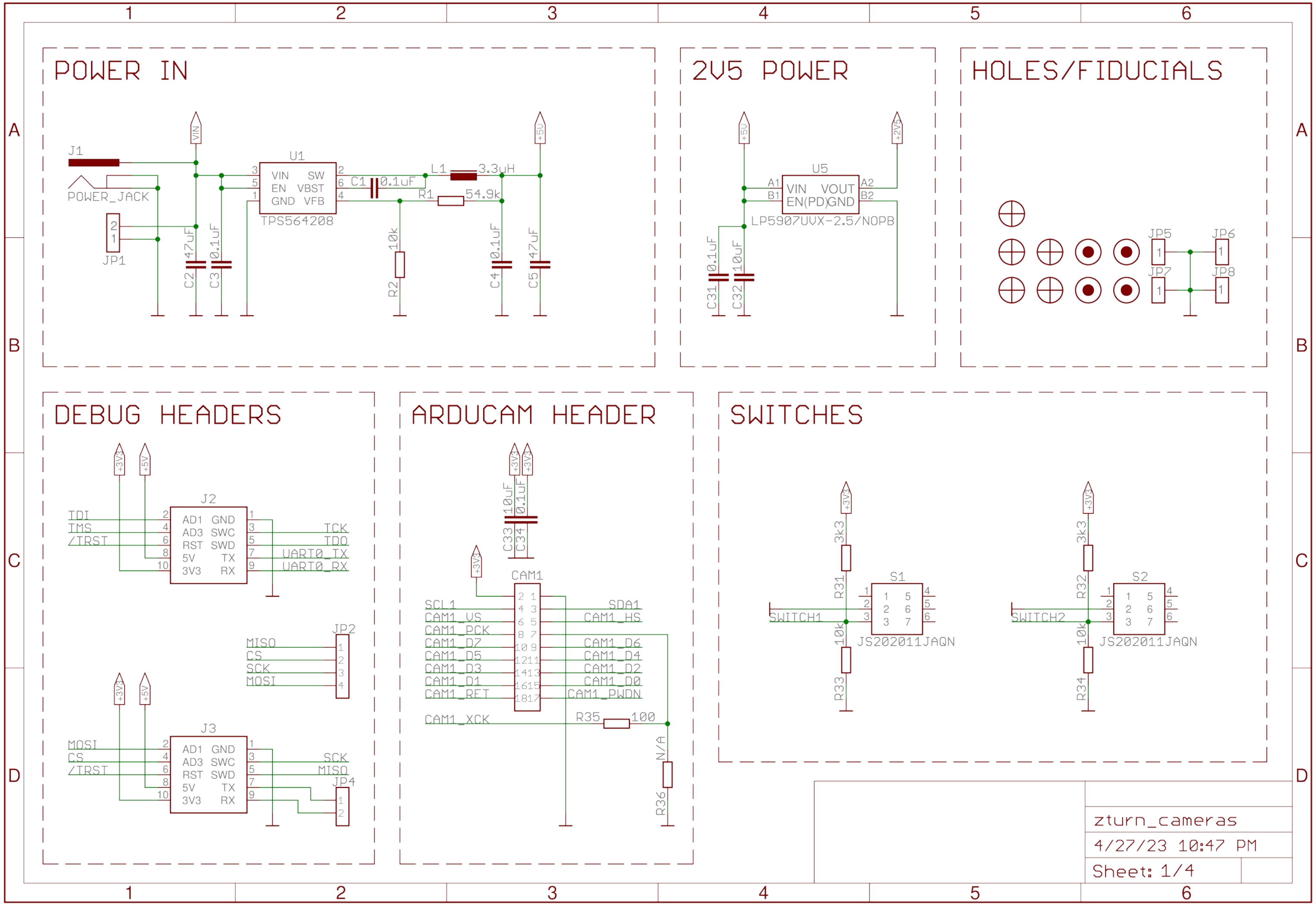
2 I am aware that the author also retains the rights specified in clause 1 of the non-exclusive licence.

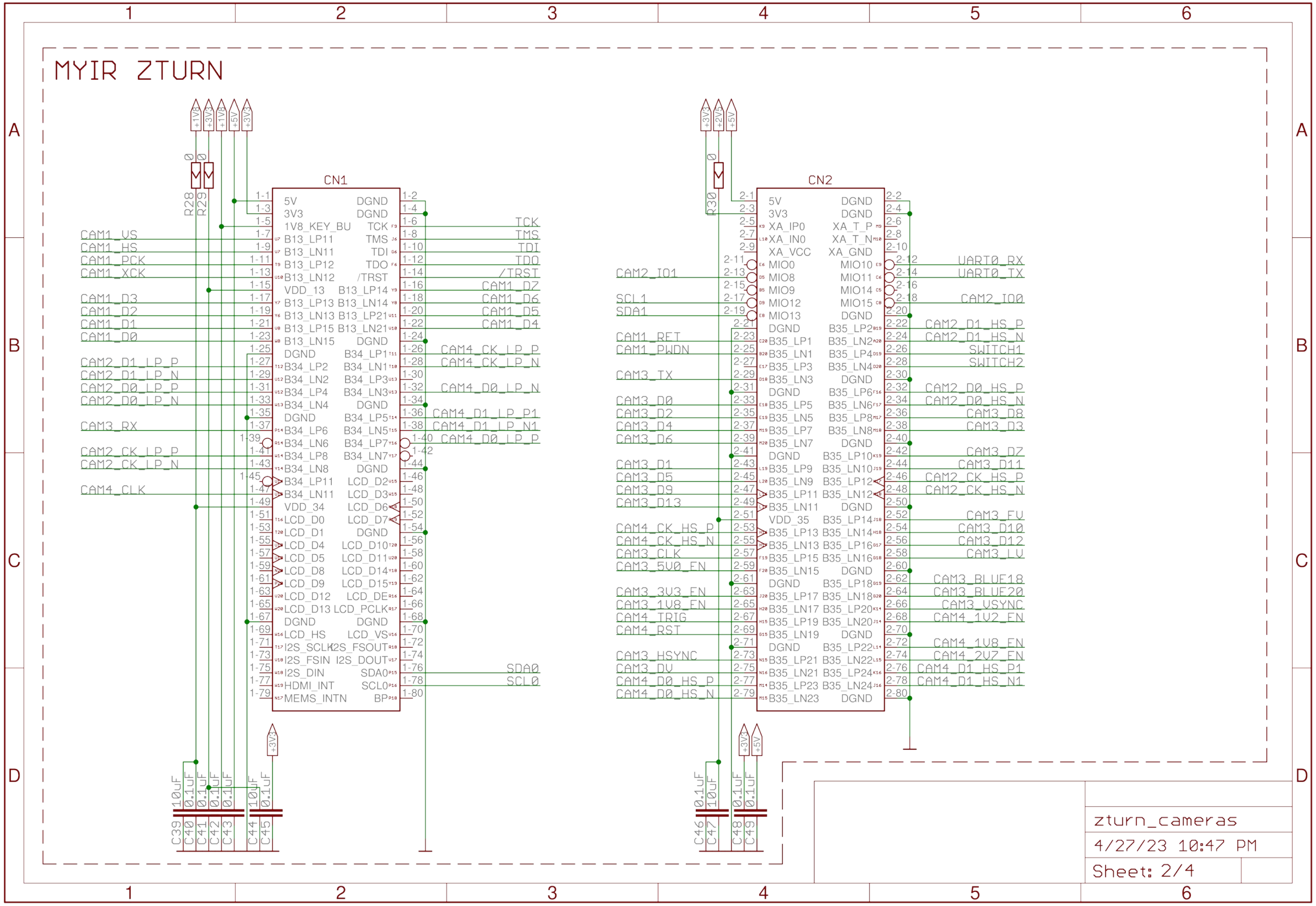
3 I confirm that granting the non-exclusive licence does not infringe other persons' intellectual property rights, the rights arising from the Personal Data Protection Act or rights arising from other legislation.

08.05.2023

[1] The non-exclusive licence is not valid during the validity of access restriction indicated in the student's application for restriction on access to the graduation thesis that has been signed by the school's dean, except in case of the university's right to reproduce the thesis for preservation purposes only. If a graduation thesis is based on the joint creative activity of two or more persons and the co-author(s) has/have not granted, by the set deadline, the student defending his/her graduation thesis consent to reproduce and publish the graduation thesis in compliance with clauses 1.1 and 1.2 of the non-exclusive licence, the non-exclusive license shall not be valid for the period.

Appendix 2 – Circuit diagram

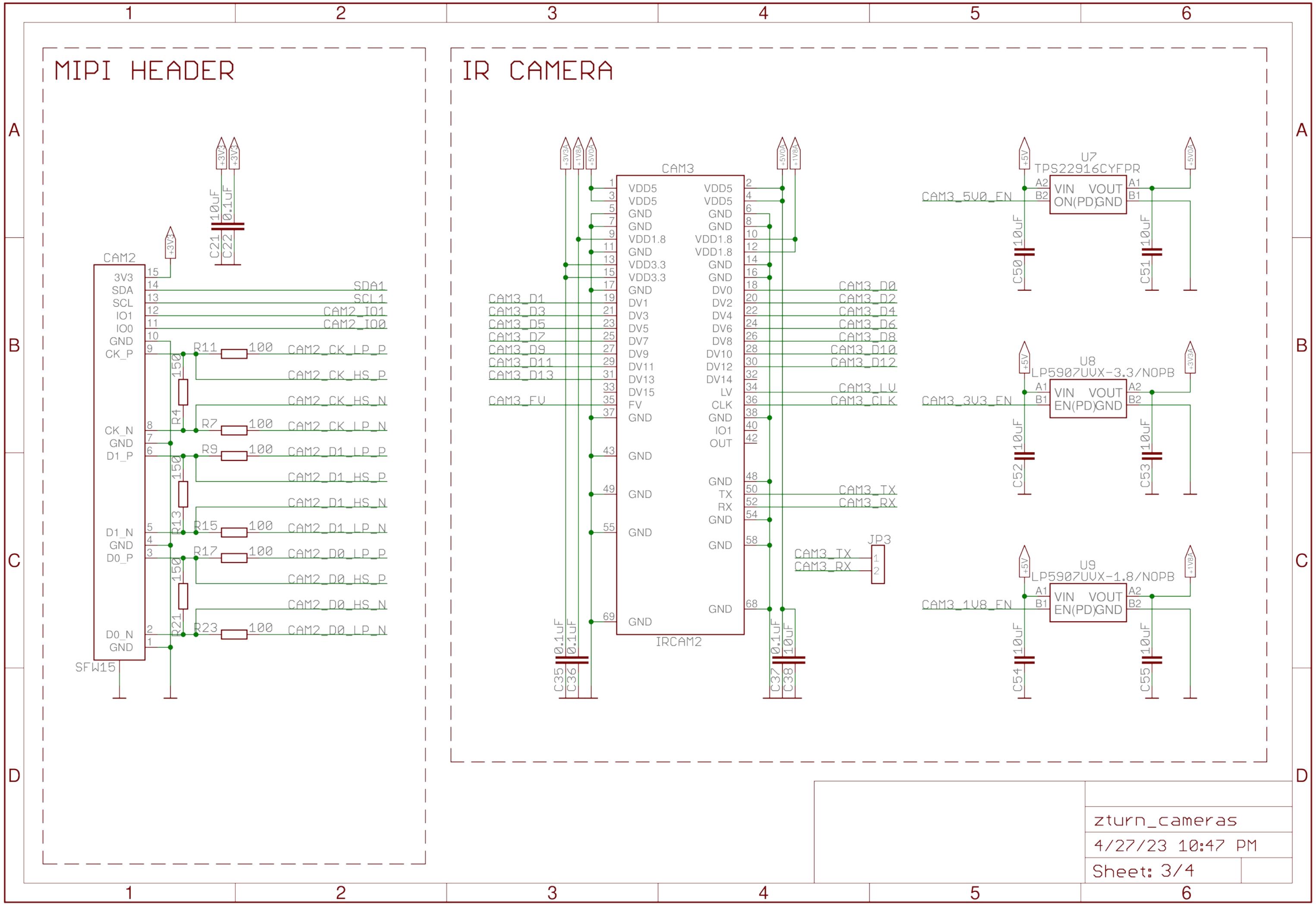



MYIR ZTURN
+1V8
+3V3
+1V8
+5V
+3V3
R28 0
R29 0
CN1
1-1 5V | DGND 1-2
1-3 3V3 | DGND 1-4
1-5 1V8_KEY_BU | TCK 1-6 TCK
CAM1_VS 1-7 B13_LP11 | TMS 1-8 TMS
CAM1_HS 1-9 B13_LN11 | TDI 1-10 TDI
CAM1_PCK 1-11 B13_LP12 | TDO 1-12 TDO
CAM1_XCK 1-13 B13_LN12 | /TRST 1-14 /TRST
1-15 VDD_13 | B13_LP14 1-16 CAM1_D7
CAM1_D3 1-17 B13_LP13 | B13_LN14 1-18 CAM1_D6
CAM1_D2 1-19 B13_LN13 | B13_LP21 1-20 CAM1_D5
CAM1_D1 1-21 B13_LP15 | B13_LN21 1-22 CAM1_D4
CAM1_D0 1-23 B13_LN15 | DGND 1-24
1-25 DGND | B34_LP1 1-26 CAM4_CK_LP_P
CAM2_D1_LP_P 1-27 B34_LP2 | B34_LN1 1-28 CAM4_CK_LP_N
CAM2_D1_LP_N 1-29 B34_LN2 | B34_LP3 1-30
CAM2_D0_LP_P 1-31 B34_LP4 | B34_LN3 1-32 CAM4_D0_LP_N
CAM2_D0_LP_N 1-33 B34_LN4 | DGND 1-34
1-35 DGND | B34_LP5 1-36 CAM4_D1_LP_P1
CAM3_RX 1-37 B34_LP6 | B34_LN5 1-38 CAM4_D1_LP_N1
1-39 B34_LN6 | B34_LP7 1-40 CAM4_D0_LP_P
CAM2_CK_LP_P 1-41 B34_LP8 | B34_LN7 1-42
CAM2_CK_LP_N 1-43 B34_LN8 | DGND 1-44
1-45 B34_LP11 | LCD_D2 1-46
CAM4_CLK 1-47 B34_LN11 | LCD_D3 1-48
1-49 VDD_34 | LCD_D6 1-50
1-51 LCD_D0 | LCD_D7 1-52
1-53 LCD_D1 | DGND 1-54
1-55 LCD_D4 | LCD_D10 1-56
1-57 LCD_D5 | LCD_D11 1-58
1-59 LCD_D8 | LCD_D14 1-60
1-61 LCD_D9 | LCD_D15 1-62
1-63 LCD_D12 | LCD_DE 1-64
1-65 LCD_D13 | LCD_PCLK 1-66
1-67 DGND | DGND 1-68
1-69 LCD_HS | LCD_VS 1-70
1-71 I2S_SCLK | I2S_FSOUT 1-72
1-73 I2S_FSIN | I2S_DOUT 1-74
1-75 I2S_DIN | SDA0 1-76 SDA0
1-77 HDMI_INT | SCL0 1-78 SCL0
1-79 MEMS_INTN | BP 1-80
+3V3
C39 10uF
C40 0.1uF
C41 0.1uF
C42 0.1uF
C43 0.1uF
C44 10uF
C45 0.1uF
+3V3
+2V5
+5V
R30 0
CN2
2-1 5V | DGND 2-2
2-3 3V3 | DGND 2-4
2-5 XA_IP0 | XA_T_P 2-6
2-7 XA_IN0 | XA_T_N 2-8
2-9 XA_VCC | XA_GND 2-10
2-11 MIO0 | MIO10 2-12 UART0_RX
CAM2_IO1 2-13 MIO8 | MIO11 2-14 UART0_TX
2-15 MIO9 | MIO14 2-16
SCL1 2-17 MIO12 | MIO15 2-18 CAM2_IO0
SDA1 2-19 MIO13 | DGND 2-20
2-21 DGND | B35_LP2 2-22 CAM2_D1_HS_P
CAM1_RET 2-23 B35_LP1 | B35_LN2 2-24 CAM2_D1_HS_N
CAM1_PWDN 2-25 B35_LN1 | B35_LP4 2-26 SWITCH1
2-27 B35_LP3 | B35_LN4 2-28 SWITCH2
CAM3_TX 2-29 B35_LN3 | DGND 2-30
2-31 DGND | B35_LP6 2-32 CAM2_D0_HS_P
CAM3_D0 2-33 B35_LP5 | B35_LN6 2-34 CAM2_D0_HS_N
CAM3_D2 2-35 B35_LN5 | B35_LP8 2-36 CAM3_D8
CAM3_D4 2-37 B35_LP7 | B35_LN8 2-38 CAM3_D3
CAM3_D6 2-39 B35_LN7 | DGND 2-40
2-41 DGND | B35_LP10 2-42 CAM3_D7
CAM3_D1 2-43 B35_LP9 | B35_LN10 2-44 CAM3_D11
CAM3_D5 2-45 B35_LN9 | B35_LP12 2-46 CAM2_CK_HS_P
CAM3_D9 2-47 B35_LP11 | B35_LN12 2-48 CAM2_CK_HS_N
CAM3_D13 2-49 B35_LN11 | DGND 2-50
2-51 VDD_35 | B35_LP14 2-52 CAM3_FV
CAM4_CK_HS_P 2-53 B35_LP13 | B35_LN14 2-54 CAM3_D10
CAM4_CK_HS_N 2-55 B35_LN13 | B35_LP16 2-56 CAM3_D12
CAM3_CLK 2-57 B35_LP15 | B35_LN16 2-58 CAM3_LV
CAM3_5V0_EN 2-59 B35_LN15 | DGND 2-60
2-61 DGND | B35_LP18 2-62 CAM3_BLUE18
CAM3_3V3_EN 2-63 B35_LP17 | B35_LN18 2-64 CAM3_BLUE20
CAM3_1V8_EN 2-65 B35_LN17 | B35_LP20 2-66 CAM3_VSYNC
CAM4_TRIG 2-67 B35_LP19 | B35_LN20 2-68 CAM4_1V2_EN
CAM4_RST 2-69 B35_LN19 | DGND 2-70
2-71 DGND | B35_LP22 2-72 CAM4_1V8_EN
CAM3_HSYNC 2-73 B35_LP21 | B35_LN22 2-74 CAM4_2V7_EN
CAM3_DV 2-75 B35_LN21 | B35_LP24 2-76 CAM4_D1_HS_P1
CAM4_D0_HS_P 2-77 B35_LP23 | B35_LN24 2-78 CAM4_D1_HS_N1
CAM4_D0_HS_N 2-79 B35_LN23 | DGND 2-80
+3V3
+5V
C46 0.1uF
C47 10uF
C48 0.1uF
C49 0.1uF
1 2 3 4 5 6
A B C D

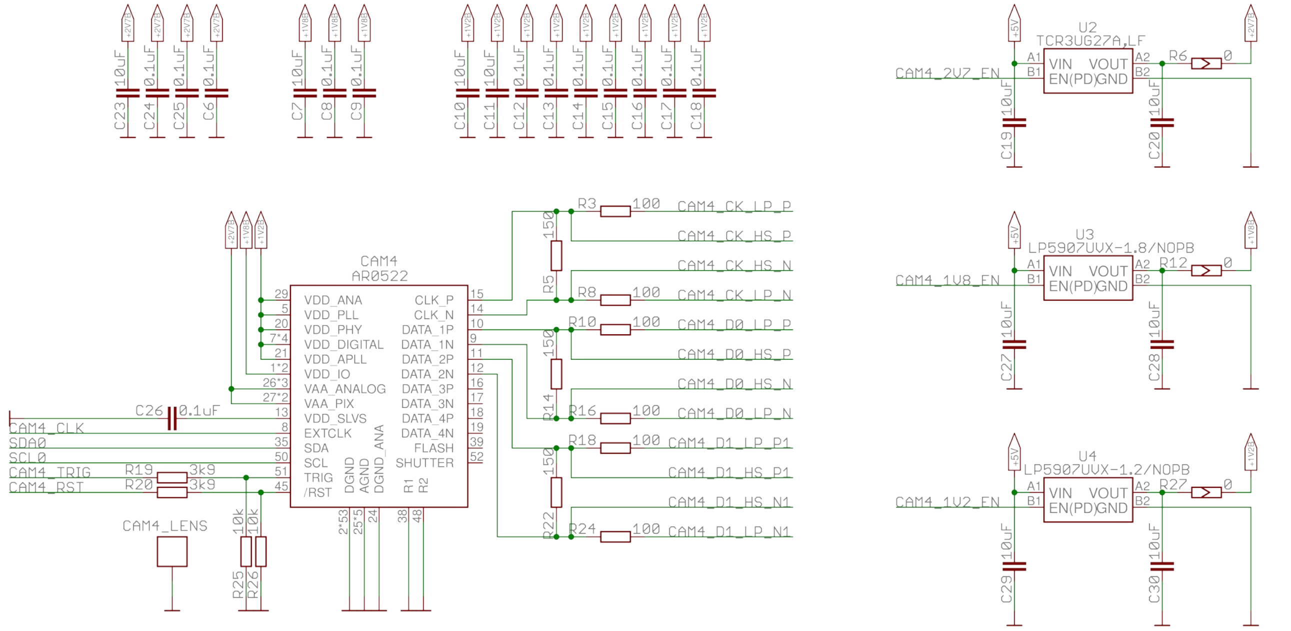
zturn_cameras

4/27/23 10:47 PM

Sheet: 2/4


MIPI HEADER
IR CAMERA
CAM2
SFW15
3V3
SDA
SCL
IO1
IO0
GND
CK_P
CK_N
GND
D1_P
D1_N
GND
D0_P
D0_N
GND
SDA1
SCL1
CAM2_IO1
CAM2_IO0
CAM2_CK_LP_P
CAM2_CK_HS_P
CAM2_CK_HS_N
CAM2_CK_LP_N
CAM2_D1_LP_P
CAM2_D1_HS_P
CAM2_D1_HS_N
CAM2_D1_LP_N
CAM2_D0_LP_P
CAM2_D0_HS_P
CAM2_D0_HS_N
CAM2_D0_LP_N
R11 100
R7 100
R9 100
R15 100
R17 100
R23 100
R4 150
R13 150
R21 150
C21 10uF
C22 0.1uF
+3V3
CAM3
IRCAM2
VDD5
GND
VDD1.8
VDD3.3
DV0
DV1
DV2
DV3
DV4
DV5
DV6
DV7
DV8
DV9
DV10
DV11
DV12
DV13
DV14
DV15
FV
LV
CLK
IO1
OUT
TX
RX
CAM3_D0
CAM3_D1
CAM3_D2
CAM3_D3
CAM3_D4
CAM3_D5
CAM3_D6
CAM3_D7
CAM3_D8
CAM3_D9
CAM3_D10
CAM3_D11
CAM3_D12
CAM3_D13
CAM3_FV
CAM3_LV
CAM3_CLK
CAM3_TX
CAM3_RX
JP3
+3V3
+1V8
+5V0
C35 0.1uF
C36 0.1uF
C37 0.1uF
C38 10uF
U7
TPS22916CYFPR
VIN
VOUT
ON(PD)
GND
CAM3_5V0_EN
C50 10uF
C51 10uF
+5V
+5V0
U8
LP5907UVX-3.3/NOPB
EN(PD)
CAM3_3V3_EN
C52 10uF
C53 10uF
U9
LP5907UVX-1.8/NOPB
CAM3_1V8_EN
C54 10uF
C55 10uF
1
2
3
4
5
6
A
B
C
D
zturn_cameras
4/27/23 10:47 PM
Sheet: 3/4

AR0522 CAMERA

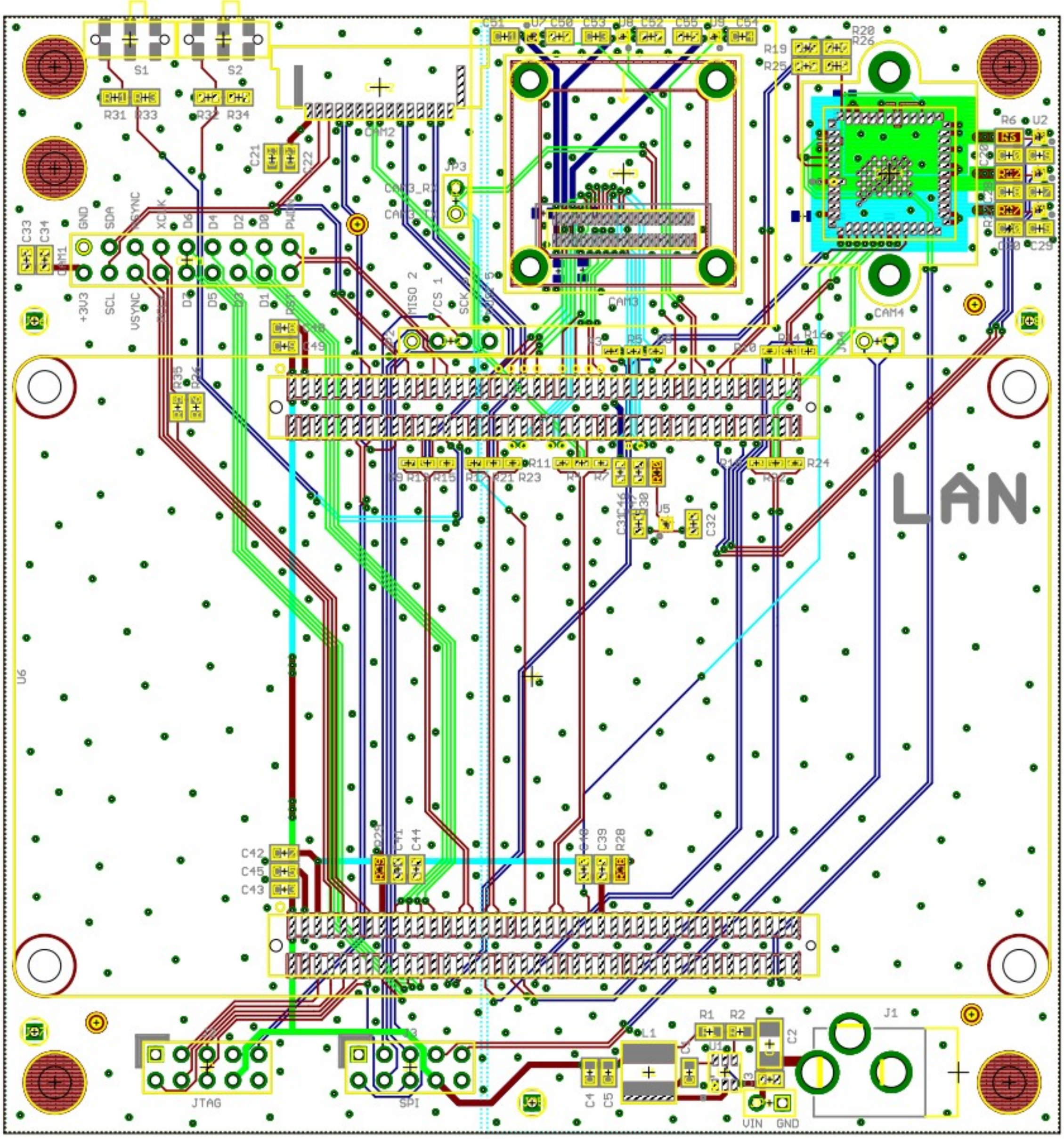


zturn_cameras

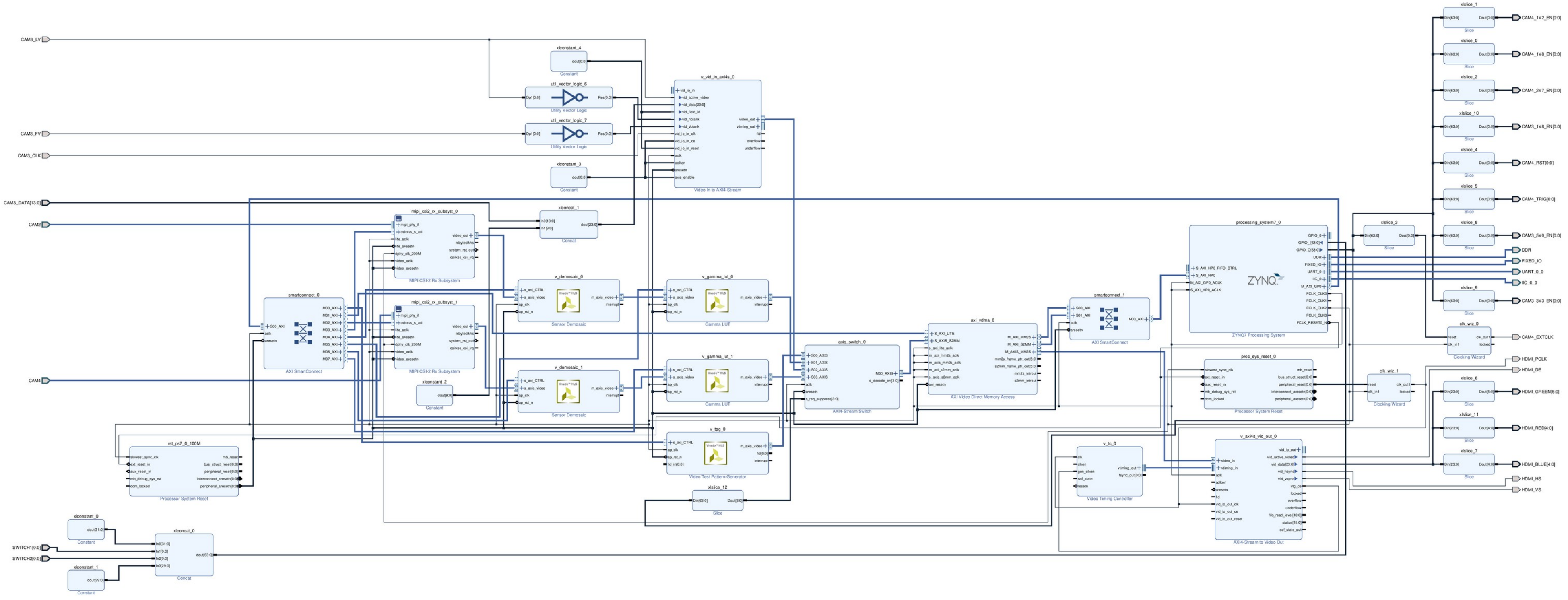
4/27/23 10:47 PM

Sheet: 4/4

Appendix 3 – PCB layout



Appendix 4 – HDL block design


CAM3_LV
CAM3_FV
CAM3_CLK
CAM3_DATA[13:0]
CAM2
CAM4
SWITCH1[0:0]
SWITCH2[0:0]
xlconstant_4
Constant
util_vector_logic_6
Utility Vector Logic
util_vector_logic_7
Utility Vector Logic
xlconstant_3
Constant
v_vid_in_axi4s_0
Video In to AXI4-Stream
xlconcat_1
Concat
mipi_csi2_rx_subsyst_0
MIPI CSI-2 Rx Subsystem
mipi_csi2_rx_subsyst_1
MIPI CSI-2 Rx Subsystem
xlconstant_2
Constant
smartconnect_0
AXI SmartConnect
v_demosaic_0
Sensor Demosaic
v_demosaic_1
Sensor Demosaic
v_gamma_lut_0
Gamma LUT
v_gamma_lut_1
Gamma LUT
v_tpg_0
Video Test Pattern Generator
xlslice_12
Slice
rst_ps7_0_100M
Processor System Reset
xlconstant_0
Constant
xlconstant_1
Constant
xlconcat_0
Concat
axis_switch_0
AXI4-Stream Switch
axi_vdma_0
AXI Video Direct Memory Access
smartconnect_1
AXI SmartConnect
processing_system7_0
ZYNQ
ZYNQ7 Processing System
proc_sys_reset_0
Processor System Reset
v_tc_0
Video Timing Controller
v_axi4s_vid_out_0
AXI4-Stream to Video Out
clk_wiz_0
Clocking Wizard
clk_wiz_1
Clocking Wizard
xlslice_1
xlslice_0
xlslice_2
xlslice_10
xlslice_4
xlslice_5
xlslice_3
xlslice_8
xlslice_9
xlslice_6
xlslice_11
xlslice_7
Slice
CAM4_1V2_EN[0:0]
CAM4_1V8_EN[0:0]
CAM4_2V7_EN[0:0]
CAM3_1V8_EN[0:0]
CAM4_RST[0:0]
CAM4_TRIG[0:0]
CAM3_5V0_EN[0:0]
DDR
FIXED_IO
UART_0_0
IIC_0_0
CAM3_3V3_EN[0:0]
CAM4_EXTCLK
HDMI_PCLK
HDMI_DE
HDMI_GREEN[5:0]
HDMI_RED[4:0]
HDMI_BLUE[4:0]
HDMI_HS
HDMI_VS