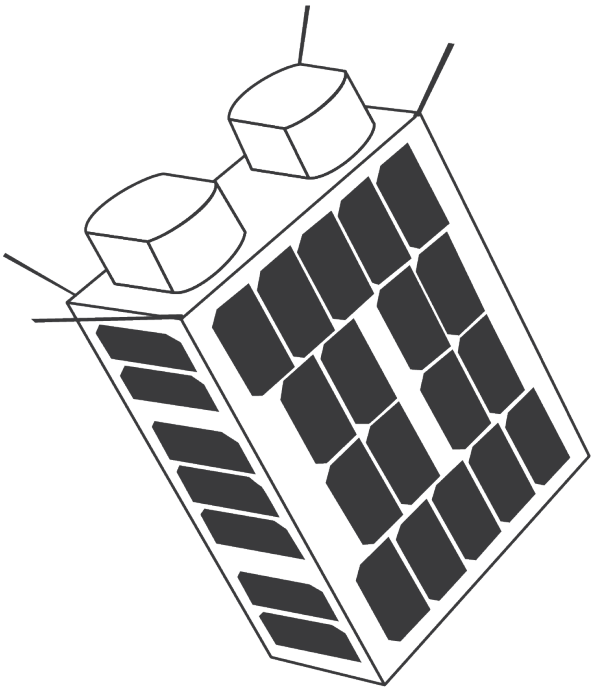
Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Engineering Cybernetics

Magnus Danielsen

System Integration and Testing of On-Board Processing System for a Hyperspectral Imaging Payload in a CubeSat

Master's thesis in Cybernetics and Robotics
Supervisor: Tor Arne Johansen. Co-Supervisor: Sivert Bakken
June 2020



NTNU
Kunnskap for en bedre verden

Magnus Danielsen

System Integration and Testing of On-Board Processing System for a Hyperspectral Imaging Payload in a CubeSat

Master's thesis in Cybernetics and Robotics
Supervisor: Tor Arne Johansen. Co-Supervisor: Sivert Bakken
June 2020

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Engineering Cybernetics

NTNU

Kunnskap for en bedre verden

Sammendrag

På NTNU jobber et tverrfaglig team som en del av NTNU SmallSatLab med å designe og utvikle nyttelast for en kubesatellitt. Nyttelasten skal integreres i en satellittbuss levert av NanoAvionics, med planlagt oppskytning i første kvartal 2021.

Formålet med satellitten er å avbilde havområder med et hyperspektralt kamera, et kamera som splitter lys opp i individuelle bølgelengder. En mikrokontroller i satellittens nyttelast skal utføre avansert bildebehandling i både prosessor og programmerbar logikk for å raskt kunne detektere fenomener i havområdene.

Målsetningen for masteroppgaven var å sørge for en fungerende minimal prosesseringskjede for hyperspektral bildetagning på kubesatellittens nyttelast. Prosesseringskjeden skal bestå av tolkning av kommandoer fra bakkestasjonen, kommunikasjon med det hyperspektrale kameraet og prossesering av hyperspektral data i både prosessor og programmerbar logikk. Arbeidet skulle ta utgangspunkt i funksjonalitet som er utviklet ved NTNU SmallSatLab, og integrere dette i eksisterende systemer for nyttelastens programvare.

Denne masteroppgaven beskriver hvordan enkeltstående bestanddeler har blitt integrert til en prosesseringskjede for bildebehandling av hyperspektrale bilder. I tillegg beskrives det hvordan bestanddelene har blitt testet, både som for seg selv og satt sammen. Prosesseringskjeden har under testing tilfredstilt prosjektets krav for en minimal prosesseringskjede. Det har i tillegg blitt integrert en modul for tidsfesting av enkeltbilder under bildetagningen. Arbeid med kodekvalitet har pågått gjennom hele semesteret. Det har også blitt utført arbeid på prosjektet som ikke direkte har vært knyttet opp mot målsetningen for oppgaven.

Summary

At NTNU, a multidisciplinary team is working as a part of NTNU SmallSatLab to design and develop a payload for a CubeSat. The payload is to be integrated into a satellite bus by NanoAvionics, with a planned launch in the first quarter of 2021.

The purpose of the satellite is to do oceanographic imaging using a hyperspectral camera. This is a camera that splits light into individual wavelengths. A microcontroller in the payload of the satellite does advanced image processing on the hyperspectral data in both processor and programmable logic. This is to quickly detect phenomena in the oceans.

The goal of this Master's thesis was to ensure a working minimal processing pipeline for hyperspectral imaging on the satellite's payload. The pipeline is to consist of interpretation of commands from a ground station, interfacing with the hyperspectral camera and do the processing of the hyperspectral data in both processor and programmable logic. The work was to be based on functionality developed at the NTNU SmallSatLab. This functionality was to be integrated into existing systems for the payload's software.

This Master's thesis describes how separate components have been integrated into a processing pipeline for processing of hyperspectral images. The testing of the pipeline is described for individual components as well as the entire pipeline. The processing pipeline has met the team's requirements for a minimal processing pipeline. In addition to this, a module for timestamping individual frames during capture has been integrated. Work on code quality has been done throughout the semester. In addition, there has been done work on the project that is not directly related to the goal of this thesis.

Preface

Permission to Use Personal Information

All persons involved in issues or pull requests included in the appendix of this thesis have given their consent to have their username and/or profile picture/avatar shown.

Access to Internal Documents and Private Repositories

This thesis references both internal documents of the SmallSatLab organization on NTNU, as well as private repositories belonging to the *NTNU SmallSat Lab* organization on GitHub. Access to these documents or repositories can be requested from my co-supervisor Sivert Bakken at sivert.bakken@ntnu.no.

Previous Work

Prior to working on this thesis, a specialization project [11] was written on the same topic. This thesis is a continuation of the work done on the project assignment. Some of the background material for this thesis is the same as for the project assignment. Some sections in this thesis are based on similar sections in the project report. Following is a complete list of the cases of similarities:

- Chapter 1 introduction is based on chapter 1 in the project report. Except for section 1.5 about the thesis' structure.
- Section 2.1 about HSI is the based on section 2.1 in the project report.
- Section 2.2 about network communication is based on section 2.2 in the project report.
- Section 2.4 about embedded Linux is based on section 2.3 in the project report.
- Section 2.5 about building embedded Linux is based on section 2.4 in the project report.
- Subsection 2.6.1 about Docker based on section 2.5 in the project report.
- Section 3.1 about the payload hardware is based on section 3.1 in the project report.
- Section 3.2 about the payload software is based on section 3.2 in the project report. Except for:
  - Subsection 3.2.3 on the repository *test-mcus*.
  - Subsection 3.2.8 on CCSDS123.
  - Subsection 3.2.9 about the imaging pipeline.
  - Subsection 3.2.10 about the timestamping module.

Acknowledgments

I want to thank all members of the HYPSO project for all support and help throughout the past year. No question was too stupid, and no debugging session too long. I want to give a special thanks to Milica Orlandic for all the counseling I got, even though she didn't have to, and to Christoffer Boothby for solving (almost all) my low-level and Linux kernel specific issues. And to Dennis Langer for being my partner in crime in developing software for the HSI camera.

I also want to thank all persons who have contributed to the thesis with illustrations and photographs. The illustration on the cover was made by Live Jacobsen.

COVID-19 Outbreak

The global outbreak of COVID-19 in the spring of 2020 affected the entire world in some way or another. When Norway shut down on the 12th of March, the campus was no exception. We who were present in the lab that day had to act quickly to make the equipment we had accessible remotely over the internet. Some of us also brought home equipment relevant to thesis work.

The rest of the semester, only having access to the lab or other team members remotely had an impact on the workflow. Not being able to sit down with another developer to troubleshoot a bug made the development process harder. As did having to share the single test setup in the lab with an HSI (Hyperspectral Imaging) camera when it was not always possible to tell if someone was using it. Integrating a service developed by someone else, and not being able to sit down with them and have them explain how it is supposed to be interfaced can lead to flawed implementations.

The impacts of the shutdown have affected all work done on this thesis since the shutdown. However, the team's willingness to attend online meetings, and endure debugging sessions on Slack, the chosen communication platform for the team, that lasts for hours has made this thesis possible regardless.

Table of Contents

List of Tables

List of Figures

Listings

Abbreviations and Terms

Abbreviations

ACK	Acknowledgement. 12
ADCS	Attitude Determination Control System. 4
AoI	Area of Interest. 37, 63, 67
AOSN	Autonomous Ocean Sampling Network. 1–3
API	Application Program Interface. 13, 20, 36, 38, 42, 79
ARM	Acorn RISC Machine. 28, 29, 35, 36, 56, 58
ASV	Autonomous Surface Vehicle. 1
AUV	Autonomous Underwater Vehicle. 1
BIP	Band Interleaved by Pixel. 39, 59, 62
BoB	Breakout Board. xvii, 6, 21, 22, 77, 79, 80
BOBIP	Baseline On-Board Imaging Pipeline. 42, 73
BSc	Bachelor of Science. 2, 43, 79, 84, 87
BSQ	Band Sequential. 39
CAN	Controller Area Network. 5–7, 11, 12, 17, 28, 44, 46, 72
CCSDS	The Consultative Committee for Space Data Systems. xvii
CLAW-1	Colored Littoral Zone and Algae Watcher 1. 6, 8, 13, 21, 25, 26
CLI	Command Line Interface. 28, 29, 56, 58, 62, 78
COTS	Commercial Off-The-Shelf. 2, 89
CPU	Central Processing Unit. 21
CRC	Cyclic Redundancy Check. 12
CSP	Cubesat Space Protocol. 7, 11, 12, 28, 42, 56, 58, 61, 74, 78, 87
DMA	Direct Memory Access. xvii, 39, 73
ECC	Error-Correcting Code. 71, 73
eMMC	Embedded Multimedia Card. 21
EOF	End-Of-Field. 12
EPS	Electrical Power System. 4, 6, 28, 44, 46, 52, 85
ESD	Electrostatic Discharge. 22, 24, 44, 52
FC	Flight Computer. 4, 34, 43, 44, 80
FPGA	Field-programmable Gate Array. 7, 21, 24, 39, 42, 43, 56, 59, 71, 73, 74, 84, 87, 88

FPS Frames Per Second. 37, 38
FS File System. 17, 58
FT File Transfer. 29, 42, 53, 71, 72

GB Gigabyte. 21
GPIO General Purpose IO. 38
GPS Global Positioning System. 4, 6, 34, 44, 80
GS Ground Station. 29
GSE Ground Support Equipment. 80, 83–85, 88
GUI Graphical User Interface. xvii, 17

HDMI High Definition Multimedia Interface. 24
HIL Hardware-in-the-Loop. 44
HSI Hyperspectral Imaging. iii, iv, xvii, 2, 6, 7, 9, 10, 13, 20, 22, 25, 26, 29, 35–40, 42–44, 46, 56–59, 61–63, 67, 71, 73, 75–79, 83–85, 87, 88
HW Hardware. 39, 54
HYPSO Hyperspectral SmallSat for Ocean Observation. iv, 2, 4, 8, 10, 12, 15, 17–19, 21, 22, 28, 36, 38, 39, 42, 44, 47, 49–51, 53, 59, 63, 80, 86–89, A-345

I^2C Inter-Integrated Circuit. 12
IDE Integrated Development Environment. 18, 34, 80
iDS Imaging Development Systems. xvii, 13, 20, 25, 26, 36, 38, 42, 79
IEEE Institute of Electrical and Electronics Engineers. 13
IO Input-output. xv, 17, 38
IP Internet Protocol. 11–13, 31, 44, 52, 54, 56, 60, 71, 72
IP Intellectual Property. 36, 54
IRQ Interrupt Request. 40, 41

JTAG Joint Test Action Group. 22

LV Launch Vehicle. 2

M6P Multi-Purpose Nano-Satellite Bus. 4–6, 12, 22, 34, 43, 44
MCU MicroController Unit. 2, 34, 44, 46, 77, 80, 85
MIO Multiplexed IO. 7, 17, 21, 54

MM2S Memory Map to Stream. 39, 40, 73
MOBIP Minimal On-Board Processing Pipeline. 42, 53, 56, 59, 62, 83, 84, 87
MSc Master of Science. 2, 35, 44, 73, 82, 88

NA NanoAvionics. i, ii, 2, 4, 44
NTNU Norwegian University for Science and Technology. i–iii, 2, 52, 54, 80

OPU On-board Processing Unit. 6, 7, 17, 28, 29, 31, 36, 39, 42, 44, 54, 56–58, 71–73, 78, 82, 86–88
OS Operating System. 17, 29, 31, 38, 39, 54, 73

PC Payload Controller. 4, 6, 12, 42, 44, 46, 52, 72
PCB Printed Circuit Board. 44
PhD Doctor of Philosophy. 2, 35
PL Programmable Logic. 7, 17, 21, 42, 71
PNG Portable Network Graphics. 38, 57
PPS Pulse-Per-Second. 6, 34, 43, 44, 46, 77–80, 83–85, 88
PR Pull Request. 15, 49, 50, 53, 57, 59, 61, 62, 71–76, 78–80, 82–86, 88
PS Processing System. 6, 7, 17, 21, 35, 36, 39, 42, 59, 73

QSPI Quad-SPI. 21

RAM Random Access Memory. 21, 36, 38
RGB Red Green Blue. 2, 6, 7, 22, 26, 29, 54, 56
RISC Reduced Instruction Set Computer. xiv, 28
RTC Real-Time Clock. 28

S2MM Stream to Memory Map. 39, 41, 73
SD Secure Digital. 22
SIMD Single Instruction, Multiple Data. 36, 56
SoC System-on-chip. xviii, 21, 24, 39, 42, 54, 55, 75
SOF Start-Of-Field. 12
SoM System-on-module. xviii, 6, 21
SPI Serial Peripheral Interface. xvi, 21
SSH Secure Shell. 52, 54
SW Software. xvii, 15, 18, 28, 49–51, 57, 59, 84, 85

TCL Tool Command Language. 17, 78
TCP Transmission Control Protocol. 11, 13

UART Universal Asynchronous Receiver-Transmitter. 6, 17, 22
UAV Unmanned Aerial Vehicle. 1
UDP User Datagram Protocol. 11, 13, 20, 25
UHF Ultra-high Frequency. 4
UNIS The University Centre in Svalbard. 25
UNIX Uniplexed Information and Computing System. 43
USB Universal Serial Bus. 6, 7, 21, 24, 26, 85

VCS Version Control System. 14
VPN Virtual Private Network. 52

Terms

iDS Camera Manager — Program by iDS to manage their cameras using a GUI. The camera manager can upload firmware, manage network settings and display camera output in real-time. 36

BoB V1R2 — Breakout Board Version 1, Revision 2. 22, 23
BoB V3 — Breakout Board Version 3. 22, 23

CCSDS123 — A lossless compression algorithm by the The Consultative Committee for Space Data Systems (CCSDS). iii, 7, 35, 42, 56, 57, 59, 73, 78, 88
Cube DMA — A DMA module developed for HSI cubes. xiii, 7, 35, 39, 42, 53–56, 59, 63, 67, 71, 73–78, 88, A-345

Docker — A service that creates *containers* for consistent Software development. iii, 8, 18, 29, 31, 57, 77, 78, 82

Petalinux — A program by Xilinx for configuring embedded Linux. 17, 18, 31, 35, 78, 82, 83, 85
PicoBoB — Collective term for PicoZed mounted on BoB. 13, 21, 24, 26, 44, 46, 52, 54, 74, 75, 77, 78

README — A file that is enclosed with software telling the user how to use it. Commonly displayed on the front page of a repository on GitHub. 31, 35

Vivado A program by Xilinx for configuring microprocessor SoC or SoMs. 17, 18, 31, 35, 39, 54, 55, 73, 78, 88

ZedBoard Zynq Evaluation and Development Board. 24, 31, 35, 36, 44, 54, 56, 61, 71–76

Chapter 1

Introduction

I remember on the trip home on Apollo 11 it suddenly struck me that that tiny pea, pretty and blue, was the Earth. I put up my thumb and shut one eye, and my thumb blotted out the planet Earth. I didn't feel like a giant. I felt very, very small

Neil Armstrong [42, p. 675]

The blue color Armstrong saw on the Earth came from the water on the surface. Approximately 70% of the surface is water, and plays a crucial role in the ecosystem of the entire earth. Monitoring the health of the oceans is important, as it may have an impact on different systems on the Earth. Such monitoring has traditionally been carried out from ships, but this has its limitations. Most significant are the limitations in time and space. A ship uses a lot of time to cover only a small part of the ocean surface, giving only bits of information about a large and complex system. The oceans can be a hostile environment for humans, who have been necessary to conduct experiments and operate ships.

With advances in technology, new ways of data collection are possible. A distributed network of autonomous agents is envisioned in the AOSN (Autonomous Ocean Sampling Network) program. Autonomous vehicles such as UAVs (Unmanned Aerial Vehicles), AUVs (Autonomous Underwater Vehicles), and ASVs (Autonomous Surface Vehicles) together with satellites are to continuously collect data on a larger scale than ever before. In this way better, more accurate, and recent information will be available [10].

1.1 CubeSats

Since Armstrong and the crew on Apollo 11 landed on the moon in 1969, mankind has taken a new *giant leap* into space. In the past, space was reserved for powerful nations with large agencies and big budgets. But with the recent advances in technology, smaller institutions and companies can go to space [37] for a reasonable price. Spacecrafts have, over the last decades, grown smaller, more advanced, more reliable, and have become cheaper. It is now possible to build a small satellite with COTS (Commercial Off-The-Shelf) components. This has made space more available.

A CubeSat (cube satellite), is much smaller in size than regular satellites. The size is standardized into *units*. A 1U picosatellite is a cube with sides of 10 cm, which gives it a 1-liter volume. CubeSats shall not have pyrotechnics, and therefore not change their orbits [37]. CubeSats ride on a LV (Launch Vehicle)[1] from the surface of the Earth, and are ejected into orbit from a dispenser. Once the satellite is in orbit, the orbit cannot be changed as it is not permitted to carry pyrotechnics for propulsion [37].

1.2 The HYPSO Mission

HYPSO (Hyperspectral SmallSat for Ocean Observation) is a 6U (unit) CubeSat with a hyperspectral payload being developed at NTNU in Trondheim, Norway. The HYPSO team is multi-disciplinary with PhD candidates, MSc and BSc students, postdocs, professors and engineers from cybernetics, electronic systems, interaction design, and product development among other disciplines [23]. With a planned launch in the first quarter of 2021, the HYPSO satellite will provide HSI from space to the AOSN seen in figure 1.1. The satellite will have a hyperspectral payload consisting of a MCU (MicroController Unit), a HSI camera and a RGB (Red Green Blue) camera integrated to a satellite bus from NA (NanoAvionics).

A hyperspectral camera is capable of detecting colors in a way the human eye cannot. The light is *diffracted* into separate wavelengths, and may even be able to detect wavelengths in the near-infrared spectra. Algae blooms give off light in this spectra [26], and can be detected by the HYPSO satellite. Early detection of algal blooms from space might give the AOSN time to investigate further, before a bloom is a threat to fish farms. We have seen that algae blooms can be harmful to fish in fish farms, and in the spring of 2019 an algae bloom killed salmon in fish farms off the coast of northern Norway [32]. Early detection of such blooms from space can give owners of fish farms time to take action to ensure the safety of their fish.

[1] Rocket

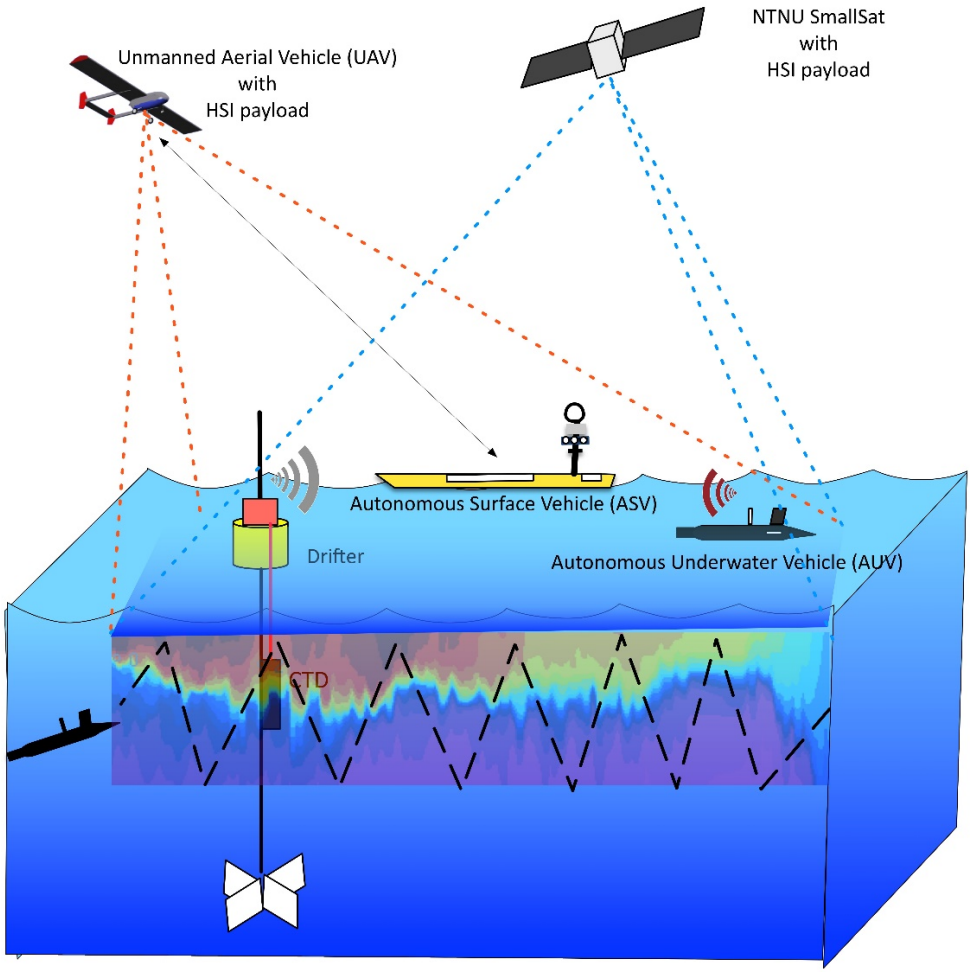


Figure 1.1: The main autonomous vehicles of the AOSN. From internal document: [18]

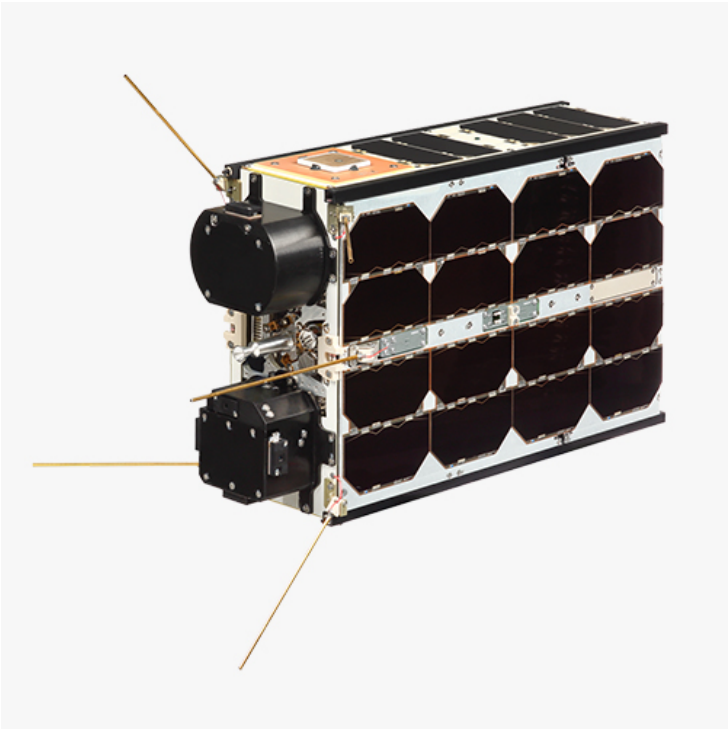


Figure 1.2: A 6u CubeSat. From NanoAvionics' website [31].

1.3 Satellite Modules

As mentioned in the previous section, the HYPSO satellite consists of the hyperspectral payload and the satellite bus provided by NA (NanoAvionics). NA is a company that specializes in CubeSats. They make everything needed for a satellite mission, from satellite subsystems to the entire satellite bus. HYPSOs payload is going to be integrated into a M6P (Multi-Purpose Nano-Satellite Bus) satellite bus, which is shown in figure 1.2. The outer dimensions of the M6P frame are $10cm \times 20cm \times 30cm$. The submodules in the M6P bus are:

- The satellite frame with solar panels.
- EPS (Electrical Power System).
- PC (Payload Controller)[2].
- FC (Flight Computer) with sub-modules:
  - ADCS (Attitude Determination Control System).
  - GPS (Global Positioning System).
- S-band and UHF (Ultra-high Frequency) radios[3].

These modules with communication channels between them are shown in figure 1.3.

[2]Note that the payload controller shares abbreviation with *Personal Computer*. In this thesis *PC* will only be used to refer to the payload controller.

[3]The S-band radio features greater bandwidth than the UHF, but requires the satellite to be poining at a ground station.

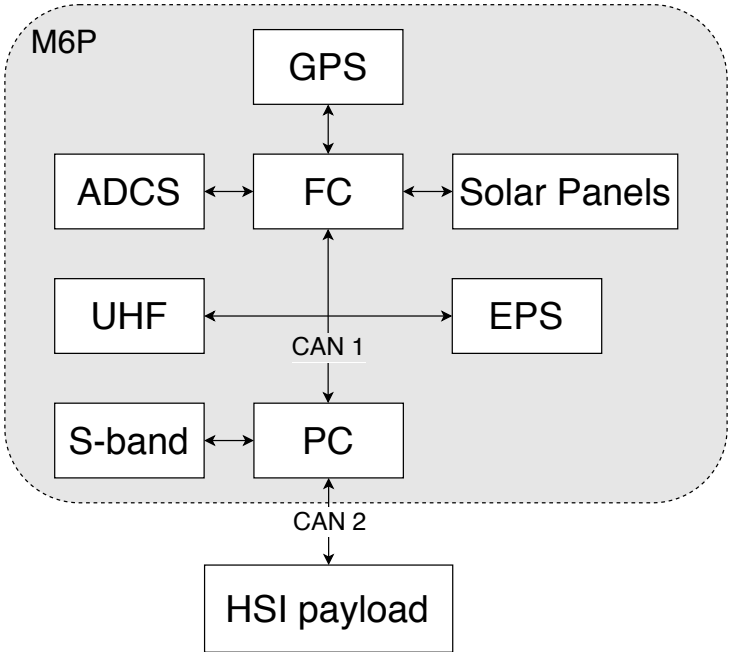


Figure 1.3: The components of the M6P bus and their communication channels. Channels are shown with arrows, and the CAN buses are labeled.

1.4 HYPSO's Payload

The hyperspectral payload, also called CLAW-1 (Colored Littoral Zone and Algae Watcher 1) is made up by the following hardware, illustrated in figure 1.4:

- A PicoZed SoM (System-on-module) with a Zynq 7030 PS (Processing System). This is the unit that performs the processing on the payload.
- A BoB (Breakout Board) for interfacing between the PicoZed and the rest of the payload, as well as the M6P bus. The BoB (Breakout Board) is mainly used for routing signals and power, and only minimal processing[4] happens here.
- The HSI camera. This is the main instrument of the payload.
- An RGB camera for georeferencing. This is a secondary instrument.

A term that is much used in the project, and also this thesis is OPU (On-board Processing Unit). The OPU is a logical subsystem of the payload that performs processing on the payload.

Figure 1.4 shows a schematic overview of the payload with the interfaces between the modules in the payload, as well as the interfaces between the payload and the M6P bus. The RGB camera uses a standard USB interface for data and power. The HSI transfers data and commands over a gigabit Ethernet, and power and a flash signal over a *HIROSE* cable. The interfaces with the rest of the M6P are a CAN connection to the PC, power lines to the EPS and a PPS signal from the GPS via the PC. More on the communication interfaces can be found in section 2.2, and on hardware in section 3.1.

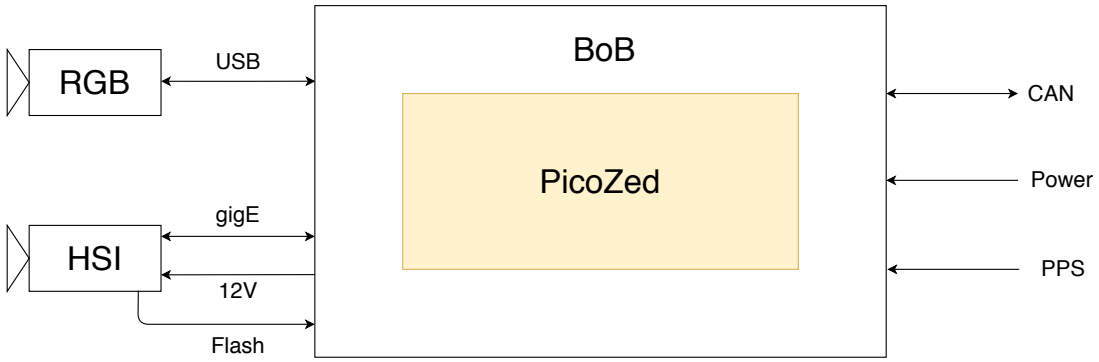


Figure 1.4: Schematic overview of the CLAW-1 payload with internal and external interfaces.

[4]BoB is equipped with a CAN transceiver and UART controller.

The OPU runs embedded Linux and has all the necessary mission software. An overview of the internals of the PicoZed and their interfaces is shown in figure 1.5. On the PS side the program *opu-services* interprets CSP messages and takes appropriate actions. The camera drivers interfaces with the HSI and RGB cameras. All interfacing with hardware on the PL side is done through the kernel. All peripherals such as CAN, Ethernet and USB are connected through the MIO. On the PL side, the Cube DMA streams HSI data to and from the memory. This data is compressed using CCSDS123. A timer module in the FPGA communicates with the kernel using AXI lite to provide accurate timestamping of each frame captured by the HSI camera.

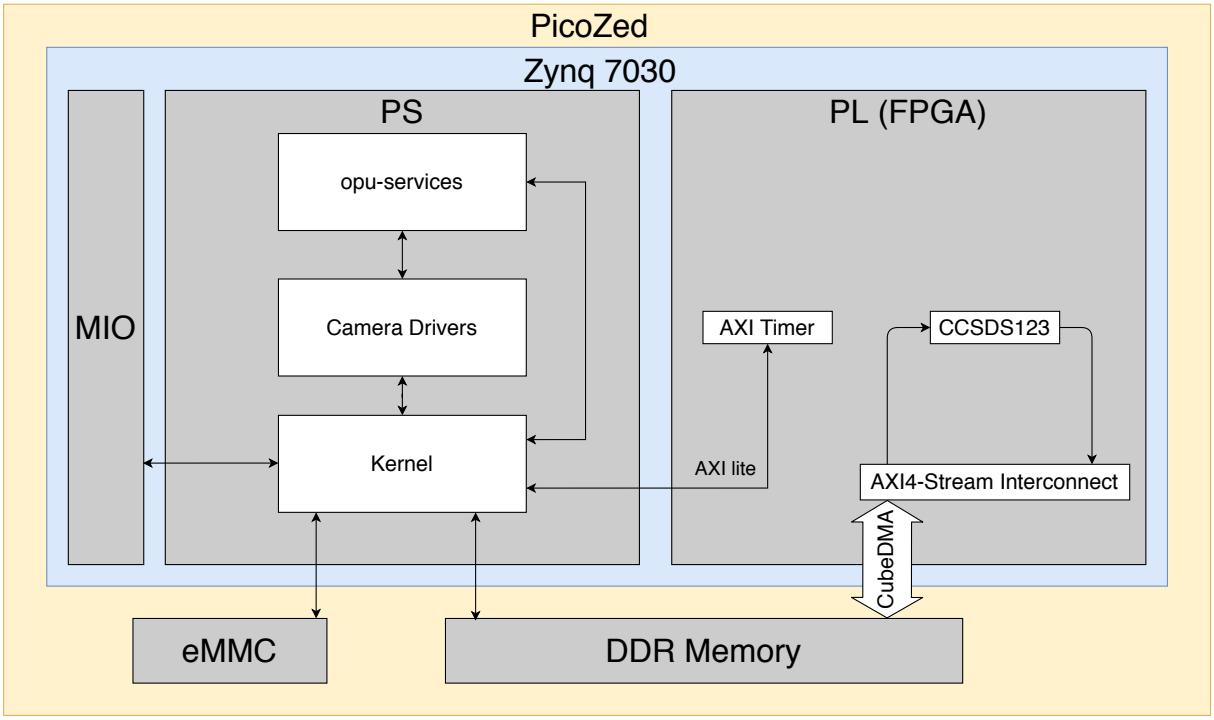


Figure 1.5: Overview of the internals of the PicoZed

1.5 Structure of the Thesis

This thesis consists of six chapters and an appendix, where the chapters are as follows:

- Chapter 1 gives an introduction to the HYPSO satellite and mission.
- Chapter 2 explains concepts the reader should know about when reading this thesis:
  - Hyperspectral imaging is covered in section 2.1.
  - An introduction to network communication can be found in section 2.2.
  - The version control system git and its uses is explained in section 2.3.
  - What embedded Linux is, and how it is built in the HYPSO project is covered in sections 2.4 and 2.5.
  - subsection 2.6 is about toolchains such as Docker.
  - Software development is not just writing code, and section 2.7 explains why.
- Chapter 3 describes the hardware (in section 3.1) and software (in section 3.2) of the CLAW-1 payload, and describes the test setups used for this thesis (in section 3.3).
- Chapter 4 is about the different methods and tools that were used when working on this thesis.
- Chapter 5 summarizes the work done and discusses the results. This is done in portions of two weeks called *sprints*, and discusses the relevant *issues* and *pull requests* from the projects' GitHub repositories.
- Chapter 6 wraps up what was achieved working on this thesis, and the impact this has had on the project.
- The appendix contains everything that did not fit in the main text of this thesis. The two first appendices feature test reports written as part of this thesis, and the last appendix contains longer bits of code in *listings*. In between these are the *issues* and *pull requests* from the project repositories on GitHub.

Chapter 2

Background

2.1 Hyperspectral Imaging

When light waves of different wavelengths mix, the eye will perceive this as a new color, a mix of the two original waves. When light of several wavelengths mix, the original sources are hard to distinguish, even with a good camera. Such light can be dispersed into its base wavelengths by for example a prism. By looking at the dispersed light, it is possible to gather information about the light source and the different media it has traveled through or been reflected by. The goal with HSI (Hyperspectral Imaging) is to quantify this information in a way regular imaging cannot do. While traditional imaging is done by focusing light with spatial information onto an optical sensor, HSI disperses a thin line of light into its base components and focuses these onto a sensor. Figure 2.1 shows the *spectrum* of an argon light source captured with an HSI camera.

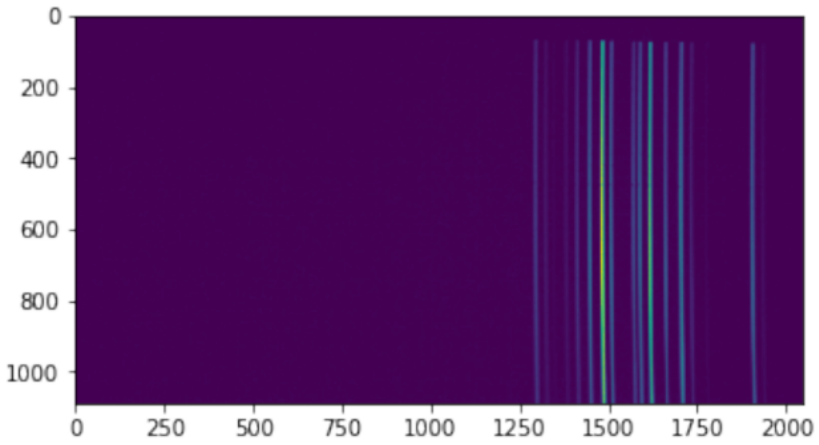


Figure 2.1: Spectrum captured by an HSI camera looking at an argon light source. The numbers on both axes show pixel count, but the x-axis represents wave length and the y-axis represents spatial distance. Each pixel indicates the intensity of light for a wavelength for a spatial point. With permission from Elizabeth Prentice.

This following paragraph about the working principle of a HSI camera and push broom imaging is based on the article *Kelp forest mapping by use of airborne hyperspectral imager* by Volent, Johnsen, and Sigernes [41].

As the HSI only captures a thin slit spatial information with each HSI frame captured, either the camera or the target needs to move to gather information with spatial extent. Figure 2.2 shows how a HSI data cube is constructed when a hyperspectral imager is flown over a segment of ground. To the right in the figure, a hyperspectral imager is shown with its internals to the left. Light from the ground is focused by a front lens (L_1) onto a slit S. This slit allows only a thin line of light through. This thin line of light represents a thin line in the Y-direction on the ground. Lens L_2 focuses this line of light onto a *dispersing element*[1]. The dispersing element will disperse the light into its base wavelengths. Lens L_3 focuses the dispersed light onto a photosensor. The sensor will then capture spatial information (along the Y-axis on the ground) along one of its axes, and spectral information (dispersed light waves) along the other. The camera will capture several frames while it moves. Each frame will be appended to previously captured frames to construct a HSI cube. This is shown to the left in the figure. This way of obtaining a HSI cube is called *push broom* and is the way the HYPSO satellite will perform imaging.

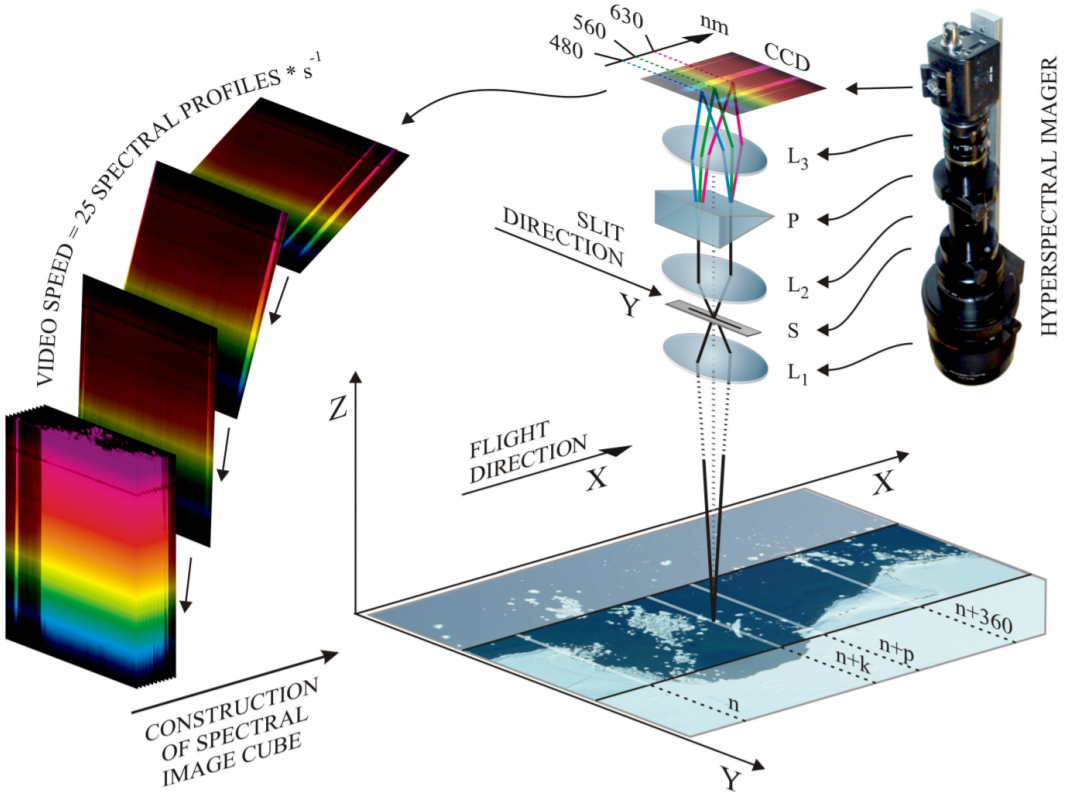


Figure 2.2: Working principle of a HSI camera and push broom scanning [41, p. 17, Fig.17], with permission from Zsolt Volent, SINTEF Ocean.

[1]Can be a prism, grating or a combination (*grism*)

2.2 Network Communication

A model for network communication consists of five layers [35], and is illustrated in table 2.1. This model is used when explaining different communication protocols in this chapter.

#	Layer
5	Application
4	Transport
3	Network
2	Link
1	Physical

Table 2.1: The 5-layer model of networks. Based on [35, p. 48, Figure 1-23].

A computer network such as the internet is a very complex construct that is virtually impossible to understand as *one* thing. A good model that divides and abstracts the complexity into different layers is a useful tool to help with understanding such large systems. A higher level interfaces with the layer beneath it, which has encapsulated its implementation. This means that a developer in the network layer does not need to be concerned with the implementation of the link layer as long as the interfaces are clear and concise. The physical layer does not need to be a concern at all. As long as the developer makes reasonable interfaces, he does not need to be concerned with what a developer in the transport layer is going to implement or how it works.

In the lowest layer, the physical layer, the transfer media is specified. Examples of transfer media are electric or light signals through wires or radio waves propagating through space. The second layer is the link layer. The link layer is concerned with sending finitely long messages such as CAN (Controller Area Network) packets. Next comes the network layer. Here, a path between two nodes is found as long as they are in the same network or network of networks. Finding the path between two IP (Internet Protocol) addresses is an example of this. They might be two computers connected physically to the same router in an office building, or two mobile phones connected through 4G. The fourth layer is the transport layer. This layer contains the implementation of CSP (Cubesat Space Protocol), UDP (User Datagram Protocol) and TCP (Transmission Control Protocol). These protocols provide increased reliability and delivery abstractions, meaning that it is more likely that data gets delivered. Last comes the application level at the top of the abstraction. This is where applications or programs that use network communication are run. When using the network from the application layer, a protocol and receiver ID is chosen, and the underlying layers do the rest to get the message through.

2.2.1 CAN

The HYPSO satellite uses CAN for most of its internal communication. There are two CAN buses in the satellite. One between the payload and the PC and one that connects the modules of the M6P bus.

Specified by ISO-11898:2003 [36], extended CAN is a link layer protocol that can send up to 8 bytes of data in one frame. The frame is shown in figure 2.3. The protocol also features an identifier field of 29 bits[2], CRC (Cyclic Redundancy Check) and ACK (Acknowledgement). The CRC field contains a checksum to verify the integrity of a received message, and ACK is used when a reply message is sent to confirm the message was correctly received. The remaining fields SOF (Start-Of-Field) and EOF (End-Of-Field) indicate the start and end of a packet respectively.

The CAN protocol requires a two-wire bus of preferably twisted wires[3] for the differential signal. The bus should be terminated by 120Ω resistors between the wires in both ends to absorb otherwise reflected signals. High-speed CAN can achieve a signaling rate of up to 1 Mbps with a bus length of 40m with no more than 30 nodes according to the ISO-11898 standard [36].

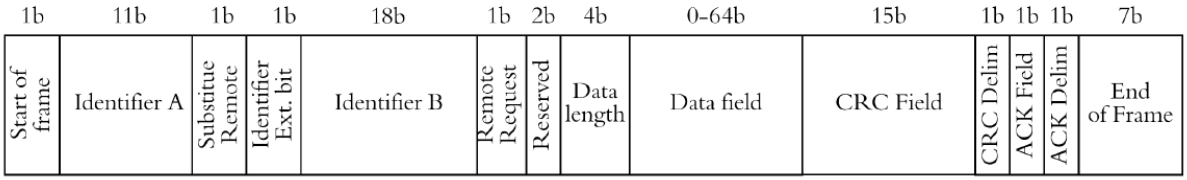


Figure 2.3: The extended CAN frame. Source: [19, p.18, figure 2.6], with permission from Magne Hov

2.2.2 CSP

CSP is a transport layer protocol (layer 4 in table 2.1) developed by GomSpace for CubeSats. The protocol is described in **CubeSat Space Protocol** [17]. CSP features a router-core (layer 3 in table 2.1), with several interfaces to the link and physical layers. This makes it possible for a CSP network to have nodes connected by CAN, radio link, serial communication or I^2C (Inter-Integrated Circuit). Every node in a CSP network has an ID (also called address) unique for the network, analogous to an IP address on the internet. By using these addresses, a node in a CSP network can send packets shown in figure 2.4 to another node in the same network. A CSP network can have up to 16 nodes.

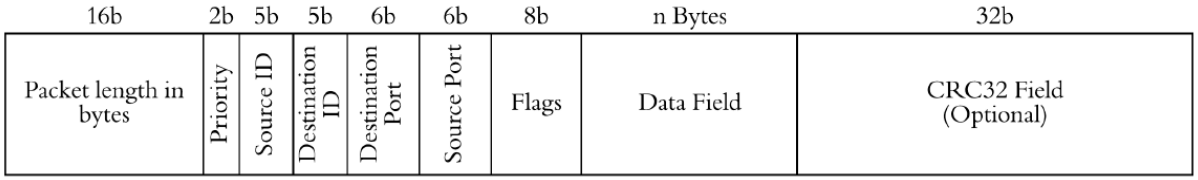


Figure 2.4: The CSP frame [19]. Source: [19, p.20, figure 2.7], with permission from Magne Hov

[2]Or 11 in the case of the standard CAN protocol.

[3]to reduce noise

2.2.3 Ethernet

The link layer (layer 2 in table 2.1) protocol Ethernet is specified in standard *IEEE 802.3* [25] by IEEE (Institute of Electrical and Electronics Engineers). Ethernet specifies the packets sent on the physical layer. Ethernet packets are commonly sent on twisted pair, fiber optic or coaxial cables. A sub-standard of Ethernet is Gigabit Ethernet. This standard allows for transfer speeds up to 1Gbps, and is used in the CLAW-1 payload between the PicoBoB and HSI camera on twisted pair cables.

2.2.4 UDP

The UDP protocol is a transport layer protocol that provides minimal protocol mechanisms. The protocol does not implement any delivery guarantee or handshaking. If such functionality is wanted, it has to be implemented separately or use the more extensive TCP protocol. UDP together with UDP uses IP as the underlying network protocol.

UDP is the protocol chosen by iDS (Imaging Development Systems) to be used for camera interfacing with the HSI camera using their API (Application Program Interface) [24].

2.3 The Version Control Software Git

The following section about the free and oper source VCS (Version Control System) git and how it works is based on the book *Pro Git* [9].

A VCS is a system that keeps track of the history of file versions in a file system. Git performs version control through snapshots called *commits* of a file system called a *repository*. This repository can either be *local* on a given computer or *remote* on a server [9].

Files in a repository must be *checked out* to a local directory to be changed. Here, changes can be applied at will. These changes can be added to the *staging area* at any time. The staging area is the stage before the snapshot is committed. When one decides to commit the changes to make a snapshot in the repository, all files in the staging will be stored in the specified repository. This process can be seen in figure 2.5. Each commit will get its unique commit *hash*, a way to identify the snapshot.

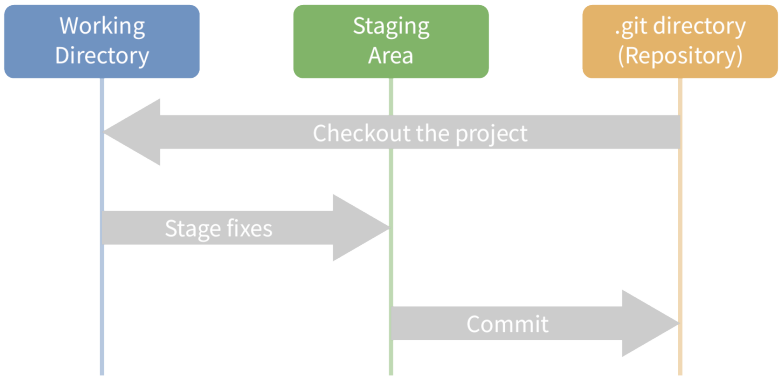


Figure 2.5: This figure shows the process of creating a commit in Git. Source: [9]

Another tool Git provides is called *branching*. Branching is usually done to keep the implementation of new features separate from the core functionality until the new feature is sufficiently developed and tested, and ready to be a part of the core functionality. All repositories have one branch per default called *master* or *master branch*. When branching out from an existing branch, a new branch will be a copy of the original one at the base. New commits can be added to either branch independently, making the two branches diverge. Two branches can be joined through a process called a *merge*. When one branch is merged into another, the branch that has been merged into will contain the full commit history of the other branch. In the case where a file has content with commits from both branches, there will be a *merge conflict*. Git will tell you what the status of the file is from both branches so you can sort it out and continue the merging.

A Git repository can be included as a directory in another as a *submodule*. This is a way to include code from a different repository while keeping them separated. All public repositories can be included as submodules, and new commits from the submodule can be fetched with built-in Git commands. Including code as submodules is therefore preferred to copying code, as new changes can be automatically integrated when they are available.

2.3.1 GitHub

Several services provide online hosting of Git repositories. The single largest host is GitHub [9]. GitHub is owned by Microsoft [30], and not directly associated with Git [9]. GitHub provides several features that complement the Git basis. The ability to have *organizations* and *teams* makes it suitable for collaboration.

Another useful feature is *issue tracking*. This is a system where information about software tasks or *issues* is stored. An issue is a textual description of something related to the code. Examples of issues are a description of a bug someone has experienced, a request for a new feature, or a question related to the software. All members of the team that has access to a repository can submit their issues and comment on the issues of others. Ramírez-Mora, Oktaba, and Gómez-Adorno have shown that giving textual description increases the success rate of the issue [33]. A *kanban board* can be used to manage the state of issues as a tool for project management. It organizes issues and PRs (Pull Requests) in *swim lanes* based on the status of the issue to get an overview of the status of a project. The kanban board used by the SW team is showed in figure 2.6. Other features and settings also exist to customize the use of Git.

When merging two branches, this can be done by a PR. The PR is a request to merge the two branches. A PR usually includes a *peer review* of the code, where readability and coding standards are addressed. Testing is also usually done in a PR. This is to check that the feature in the branch does not introduce any errors or bugs to the target branch.

HYPSOs SW team uses repositories on GitHub for version control of the various codebases the team develops, as well as the mentioned features to support the development.

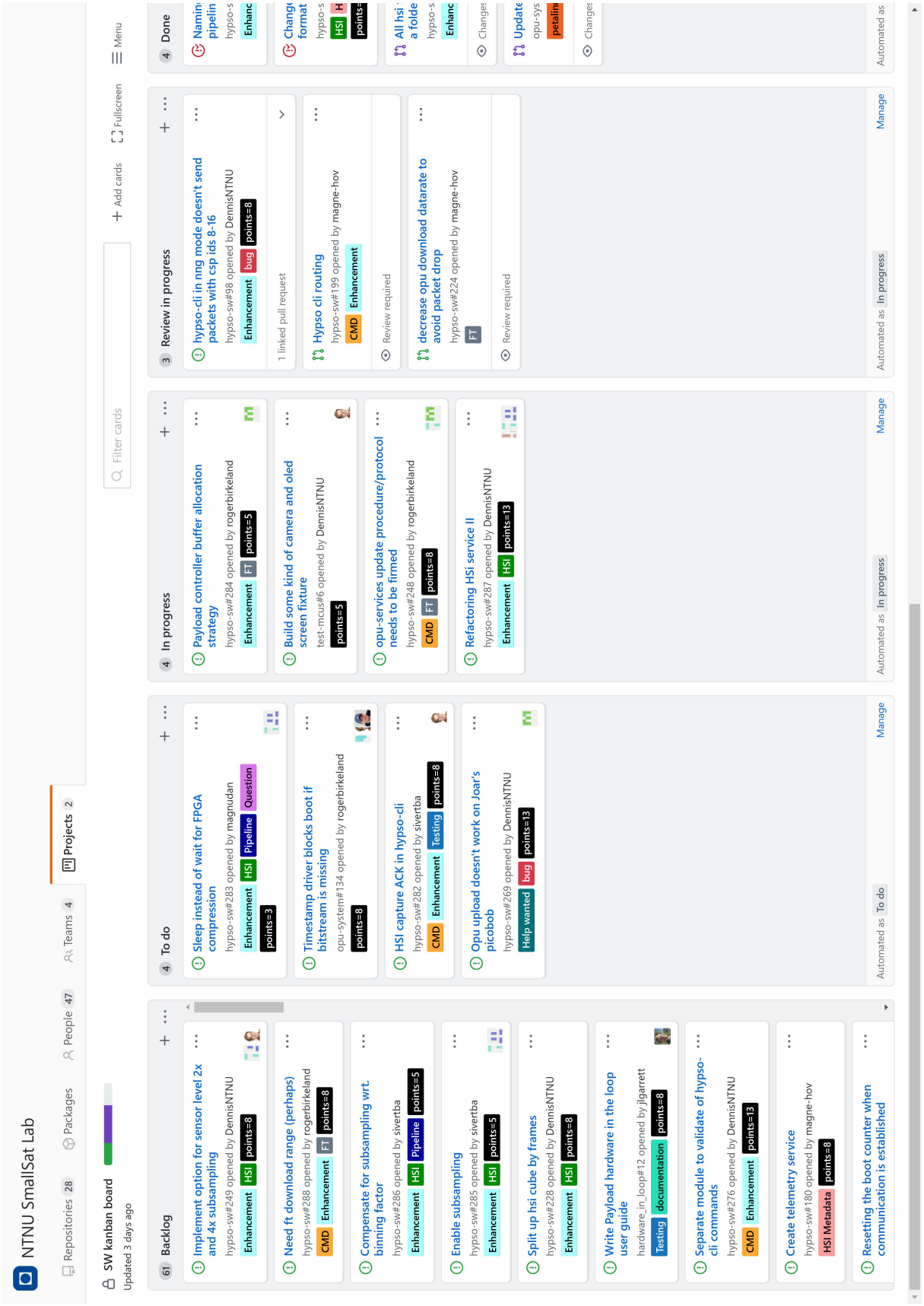


Figure 2.6: HYPSO software teams Kanban board on the 19th of June 2020.

2.4 Embedded Linux

This section is based on the book *Learning Embedded Linux Using the Yocto Project* [38].

At the core of an operating system lies the *kernel*. The kernel manages all interactions between the (user-)programs that run on a computer and the hardware. The kernel also manages resources, peripheral devices and is responsible for booting. Linux is a kernel that is free and open-source. Being free and open-source, Linux can be tailored to specific uses. Embedded Linux is a light-weight version of the Linux kernel that is meant to be used on embedded devices, such as the OPU on HYPSO.

2.5 Build Process for Embedded Linux

Building a OS (Operating System) for the OPU is a process with multiple possible configurations. First, in Vivado the hardware of the system is configured. The PS (Processing System), PL (Programmable Logic) and their interfaces such as memory, MIO (Multiplexed IO), and clocks are configured using Vivado. MIO pins can be configured to be used for standard protocols such as UART (Universal Asynchronous Receiver-Transmitter), CAN or Ethernet. Both a GUI and TCL scripting can be used to configure projects in Vivado to output the hardware definition files (`.hdf`) and bitstream files (`.bit`) needed by Petalinux.

Then, using the hardware files from Vivado, Petalinux is used to customize and build embedded Linux [43]. Among the configurations that can be done in Petalinux are the configuration of the booting sequence, memory regions, FS (File System), and the Linux kernel. Custom kernel modules can also be added to the root file system using Petalinux.

2.6 Toolchains

A *Toolchain* is a term for the tools used in the process of refining source (code)files to a program or new files needed by a system. When creating a program from code, the process might consist of some or all of the following steps [1]: Static code analyzer to assert correctness and code quality. This might be integrated in an IDE (Integrated Development Environment). A compiler is used to translate the code to optimized machine code and link required libraries together. However, a toolchain can offer more functionality than this.

When building bootfiles for embedded Linux, the entire process mentioned in section 2.5 is automated in two steps with two toolchains. The first step automatically builds hardware files using the tool Vivado. The second toolchain is more intricate. It moves files, compiles executables and kernel modules, and uses Petalinux. All with one call in the command line.

The C code developed by the HYPSO-SW (Software) team also needs a toolchain. This toolchain consists of *makefiles* that link standard and 3rd party libraries as well as files created by the team to compile the source code for the target hardware.

2.6.1 Docker

"Developing apps today requires so much more than writing code. Multiple languages, frameworks, architectures, and discontinuous interfaces between tools for each lifecycle stage creates enormous complexity." - Excerpt from the Docker website [12].

Docker is a service that addresses the mentioned complexity by running SW in *containers* on a host computer. A container is a virtual environment where software can be installed. The software installed in the container and the computer the container runs on does not affect each other. The container is installed using a file that describes all software to be installed inside the container. This means that a container can be tailored for a specific purpose and will have the same contents every time it is started. This makes Dockers containers convenient when development requires specific toolchains and build environments. Multiple developers can then be sure to use the same environment across computers or operating systems, and end up with the same result.

The HYPSO team uses Docker to build executables from the hypso-sw repository on GitHub, as a specific toolchain is required for compilation. The toolchain with all its components and libraries is installed using docker, and the toolchain is run inside the container. Docker is also used to install correct versions of the programs Vivado and Petalinux, as especially Petalinux needs to run on Linux with certain packages installed. The use of Docker for these purposes makes the functionality easily accessible to all developers on the team, and software is built consistently and reliably.

2.7 Software Development

McConnel [29] writes that developing software has more aspects than just *coding* and *debugging* (i.e. writing code and making it work as intended). He calls the process of developing software *construction* or *programming*. Coding and debugging are the most important aspects of programming, but a project is seldom successful without thorough *planning* and *design* based on *requirements*. Another important aspect of programming is *testing*. Testing should be carried out on the modules or units the program is made up of (*unit testing*), as part of integration within the program or with other programs (*integration testing*) and on the program or system of programs as a whole (*system testing*).

A programmer's productivity is not measured solely on the amount of code written. Design, integration, and testing are processes that take time. The state of the project can go from not functioning to functioning by adding or removing a line of code, or changing parameters.

2.7.1 Agile Development

Agile is an approach to structure software development. The article *Software Development Life Cycle AGILE vs Traditional Approaches* [27] discusses the pros and cons of agile development versus traditional approaches to software development such as *waterfall* and the *V-model*. Where the traditional approaches tend to do development as a top-down process in defining the system, then construct the system before testing and validating is performed in a bottom-up manner. This is a step-wise process where the team proceeds from one step to the next. The *agile* approach is to divide the project into different more or less stand-alone functionalities, and focus on one of these at a time in an iterative process. The goal is to develop one functionality over a short time-span. After this time-span, new functionality is worked on. If needed, the functionalities are revisited.

The advantages of agile versus traditional development approaches are [27]:

- Better suited for smaller projects.
- More flexible with regards to changes in requirements.
- Testing is done for each functionality implementation.
- More social interaction within a team.

The HYPSO team has chosen to do agile development using the *scrum* method. This is detailed later in section 4.2.

2.7.2 Code Quality

When programming something that is a part of a bigger project, the programmer should be concerned with making the code understandable and usable by others. There is a possibility that the program will interface with other programs, be modified by others or integrated as part of a bigger service. McConnel [29] describes many aspects of how code should be structured and what it should look like. The practices the author describes are not only nice to have for others, but the developer will also benefit from these. Some of the author's views are summarized in this section.

Good code should be abstracted and encapsulated. When interfacing with a program, this should be in a logical layer of abstraction and the implementation should be encapsulated. Take the HSI camera as an example. iDS has created an API with verbose function calls. A developer using this API does not need to know anything about the underlying communication using UDP over Ethernet or the machine code on the camera. This API requires several different function calls to set up the camera, and more to start capturing HSI frames. It would make sense to encapsulate the implementation using this API and leave an interface that provides functionality to start and stop the camera, and a function that sets up camera parameters.

Naming schemes makes the code more understandable. Names for variables and functions should state in words what they represent or what they do. On the other side, it is equally important that they do or represent what the name indicates that they do, and nothing more. If this is done correctly, a person skimming through the code does not need to read every function in detail to figure out what it does.

The visual layout helps to make the code readable. Just as with regular text, code can be formatted in a way that makes it easier to read than if it had just been a wall of text. Good use of indentation and whitespace is important to ensure readability.

Comments can be necessary to document the code. Even though all the above items are covered, the code might not be entirely self-documenting. Commenting can be necessary in cases where the intent of functions or origin of constants is not given. In other words, what a function does should not be commented as it should be clear from the code itself, but a comment on what problem the function solves might be useful information. It might be ambiguous what a constant represents and where it gets its value from. Does `#define X_POS 5;` represent centimeters or meters? Why is `#define FPS_MAX 32;` defined as it is? A comment referencing a requirement or restriction makes it less of a *magic number* appearing out of nowhere.

Chapter 3

System

3.1 Payload Hardware

3.1.1 On-Board Processing Unit

The processing unit on HYPSOs payload CLAW-1 will feature a PicoZed SoM (System-on-module) by Avnet [3] shown in figure 3.1 mounted on the BoB. The PicoZed has a Xilinx XC7Z030-1SBG485 industrial grade SoC (System-on-chip) PS with two CPUs (Central Processing Units) cores on ARM processors and a PL (Programmable Logic) FPGA (Field-programmable Gate Array). Other modules on the bard are 8GB eMMC, 1GB RAM (Random Access Memory), 128 Mbit QSPI (Quad-SPI) flash, an an Ethernet interface, a USB 2.0 controller, and 52 MIO pins.

The electrical interfaces on the PicoZed are located on the 100-pin BergStak connectors. This requires the PicoZed to be mounted on a separate device to be interfaced with. The CLAW-1 payload will feature a BoB designed specifically for use in the HYPSO satellite. When the PicoZed is mounted to the BoB, the new unit is called PicoBoB.

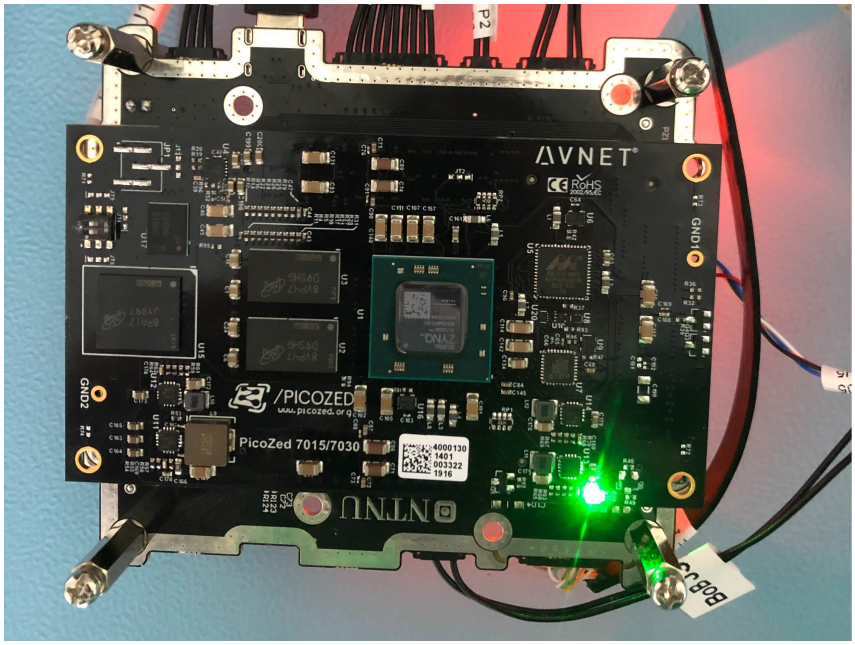


Figure 3.1: The PicoZed mounted on the breakout board. Courtesy of Martine Hjertenæs.

3.1.2 The Breakout Board

The breakout board BoB is designed by the HYPSO team to be the mechanical and electrical interface between the PicoZed, the RGB and HSI cameras and the M6P satellite bus. The current version in use in the SmallSatLab is BoB V3 (see figure 3.3b). The main difference between BoB V3 and BoB V1R2 (see figure 3.3a) which was in use during the project report is that the new version features PicoLock connectors[1], two slots for SD (Secure Digital) cards and the power regulators were changed to reduce heat generation. An issue with UART connection was also fixed. A revision of BoBV3 has been received. It has only minor fixes on hardware, and lead solder instead of lead-free.

BoB itself does no processing, except transceivers and controllers for communication. The interfaces on BoB and the modules being interfaced with are shown in figure 3.2. Note that some of the interfaces are not to be used in the satellite, but exist for test purposes. These are the JTAG (Joint Test Action Group), UART, and EXT and P3 connectors. The thermocouple will not be used either.

To reduce the risk of damaging the BoB or PicoZed, this hardware shall only be operated by personnel protected against ESD (Electrostatic Discharge). The SmallSatLab has an area dedicated for ESD sensitive equipment, and all handling of these must be done here or in similarly protected areas.

[1] Which are more suitable for the space environment than the standard connectors

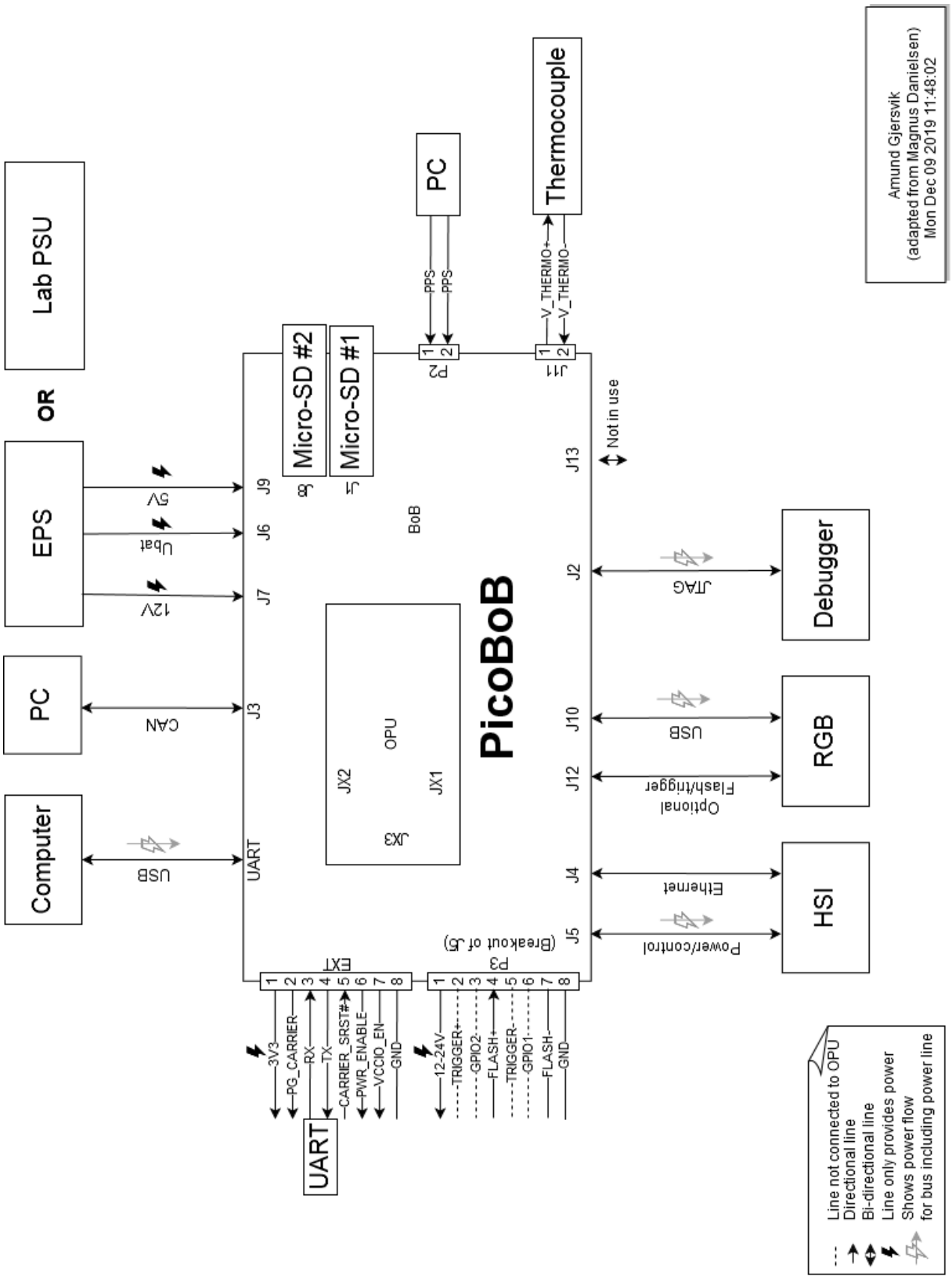
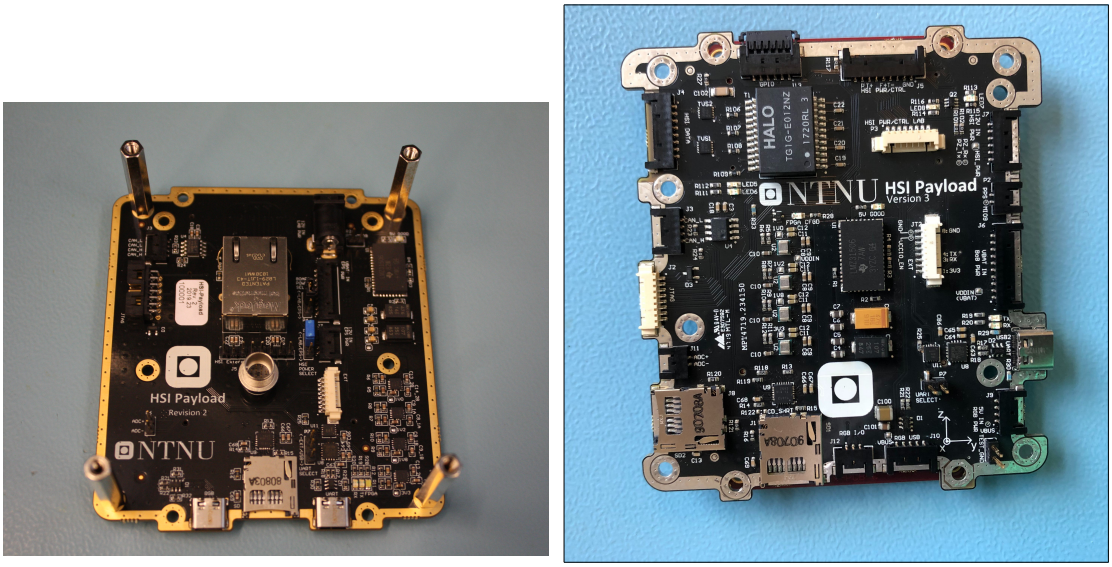


Figure 3.2: Connection diagram for BoB V3. Based on similar diagram for BoB V1R2 made during summer internship. Adapted for v3 by Amund Gjersvik. With permission.



(a) BoB version 1 revision 2

(b) BoB version 3. Courtesy of Martine Hjertenæs.

Figure 3.3: Two versions of the breakout board

3.1.3 Zynq Evaluation and Development Board

The Zynq Evaluation and Development Board (ZedBoard for short) features a XC7Z020-1CLG484C SoC. This SoC is slightly different than the one on the PicoZed. The *Zynq-7000 SoC Product Selection Guide* [4] gives that the processors are of the same type, but the ones on the ZedBoard are slower than on the PicoZed. The FPGA on the PicoZed is also substantially larger than on the ZedBoard. These differences aside, they are quite similar. The code compiled for the PicoZed will run on the ZedBoard without any problems. The ZedBoard is shown i figure 3.4.

As mentioned in the previous sections, there are limitations to working with PicoBoB. PicoBoB requires personnel to be protected against ESD, and non-standard connections make it nontrivial to work with. For faster prototyping and standard interfacing, the ZedBoard is more convenient to work with. The on-board peripherals include an RJ-45 connector for Ethernet, USB (Universal Serial Bus) ports, an HDMI (High Definition Multimedia Interface) port, and standard 100-mil pin headers. While the ZedBoard does work for prototyping, integration must be done on the PicoBoB, as the differences between the two are not negligible. Especially regarding features on the FPGA, since they are slightly different.

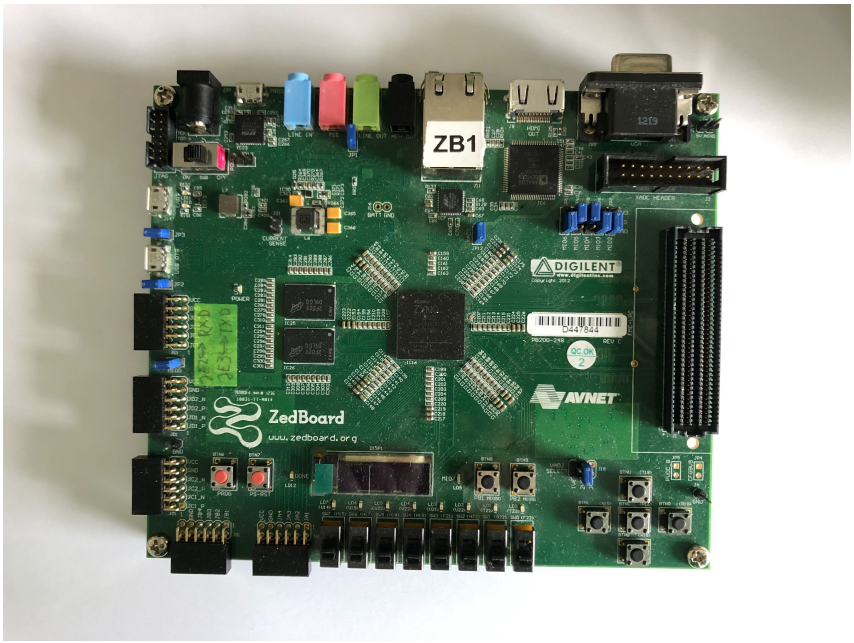


Figure 3.4: The ZedBoard. Courtesy of Martine Hjertenæs.

3.1.4 Hyperspectral Camera

The primary instrument on the CLAW-1 payload is an HSI camera. The imager is shown in the center of figure 3.6. The setup has been assembled according to the internal document *HYPSO-RP-006: HSIv6 Assembly, Fred's Design* [15] based on a design by Fred Sigernes from UNIS (The University Centre in Svalbard). The camera features a monochrome image sensor UI-5260CP-M-GL R2[2] by iDS and optics and a blazed transmission grating by Edmund Optics. The grating has 300 grooves with a blaze angle of 17.5deg. The uEye camera uses UDP over Ethernet for data and commands and a HIROSE connector for power and trigger and flash signals [24]. The assembly in the report does not reflect the current situation, where the camera, slit, and optics are fitted in a machined and anodized piece of aluminum.

A frame captured by the HSI camera can be seen in figures 2.1 and 3.5. The light sources are argon and mercury respectively, and were used to calibrate the setup. The information along the x-axis represents wave length[3] and spatial distance along the y-axis[4]. The numbers along the axes show pixel count in either direction.

[2]Changed from UI-3060CP-M-GL-R2 in the report
[3]*Columns* in iDS' documentation
[4]*Columns* in iDS' documentation

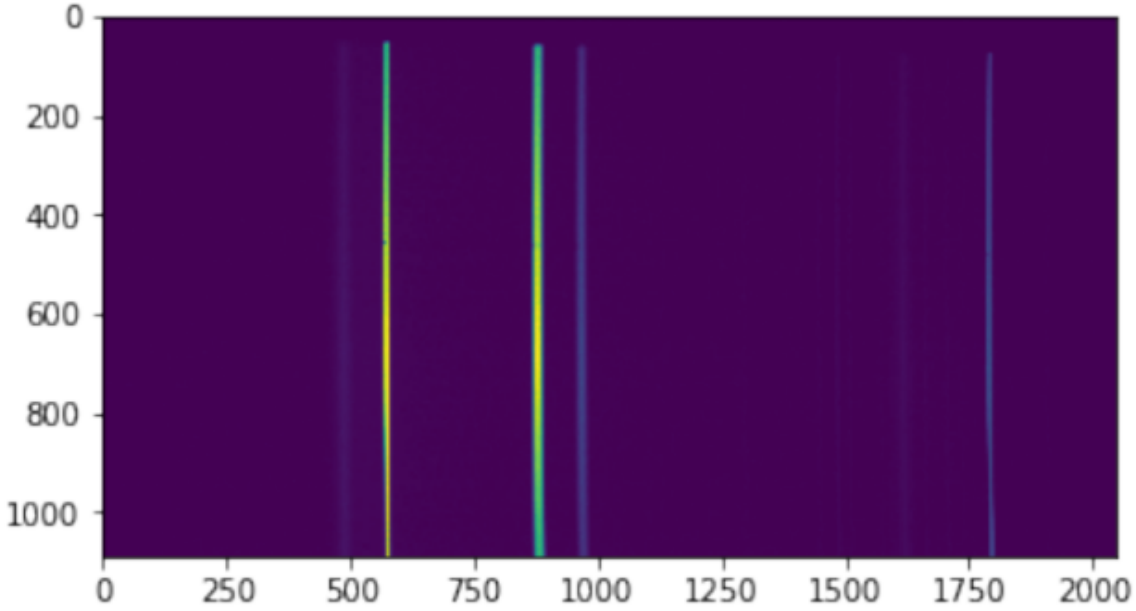


Figure 3.5: Frame captured by the HSI camera looking at a mercury light source. The numbers on both axes show pixel count, but the x-axis represents wave length and the y-axis represents spatial distance. Each pixel indicates the intensity of light for a wavelength for a spatial point. With permission from Elizabeth Prentice, also used in project report [11, p. 16, fig. 3.4].

3.1.5 RGB Camera

CLAW-1 also features an RGB camera, shown to the left in figure 3.6. This camera is to be used for *georeferencing*. When a HSI capture is halfway, the RGB camera is to capture a single snapshot of the Earth's surface. The RGB camera is also constructed from a camera by iDS (UI-1250SE-C-HQ) and optics from Edmund Optics. It uses USB for both data and power transmission from PicoBoB.

Figure 3.6: The hyperspectral imager (center) and RGB camera (left) mounted on platform for satellite integration. Courtesy of Martine Hjertenæs.

3.2 Payload Software

The code developed for the various segments and systems of the HYPSO project is under version control on GitHub in the NTNU-SmallSatLab organization. In this organization there are multiple repositories with code made for different uses. Four of these are directly related to work on this thesis. These are:

- *hypso-sw* [20]
- *opu-system* [21]
- *test-mcus* [22]
- *MasterThesis* [39]

The three first repositories are active repositories under development by the HYPSO team. While these are related to work on this thesis, multiple members of the team contribute to the repositories with code, issues and reviews. The final repository *MasterThesis* is inactive, as it was only used by a previous Master's student for a thesis.

3.2.1 hypso-sw

This section is entirely based on the repository itself [20], the Master's thesis *Design and Implementation of Hardware and Software Interfaces for a Hyperspectral Payload in a Small Satellite* [19], and the experience gained by using it for this thesis.

This repository contains the source code for the custom SW that runs on the OPU, a CLI (Command Line Interface) to communicate with the HYPSO satellite using CSP. The main use for this repository is to compile the *cli/services* pair:

- `hypso-cli` is an executable for regular computers that parses user input and generates CSP packets that are distributed on the CAN network in the HYPSO satellite.
- `opu-services` is an executable that interprets CSP packets, and performs one or several actions accordingly. The executable can be compiled for both ARM (Acorn RISC Machine) (the processor on the OPU) or regular computers (x86).

Two other executables are compiled by this repository:

- `eps-rtc-sync` is an executable for the OPU to fetch the real time from the RTC (Real-Time Clock) module on the EPS and sync the time on the OPU to the real time.
- `packet-dropper` is an executable that will deliberatly drop packets on the CAN network for testing purposes.

All source code is written in C. The toolchain used to compile is encapsulated inside a Docker container. In the container, the code is compiled simply by typing `make <option>` in the terminal. The available options are:

- None: Compiles the CLI and `opu-services` for x86.
- `ARCH=arm`: Compiles `opu-services` for ARM.
- `test`: Runs unit tests.
- `clean`: Removes all compiled executables and other generated files from compilation.

The executable `opu-services` will start several threads upon starting. A thread provides functionality tied to a *service*. The services are different functionalities that need to be present on the OPU. At the time of writing this thesis the different services are:

- A FT (File Transfer) service for moving files between the OPU and the GS (Ground Station).
- A shell service for using the shell of the OS on the OPU.
- An HSI service for capturing and processing HSI cubes.
- An RGB service for capturing and processing RGB images.

The general structure from the root directory with significant files for this thesis is given in figure 3.7.

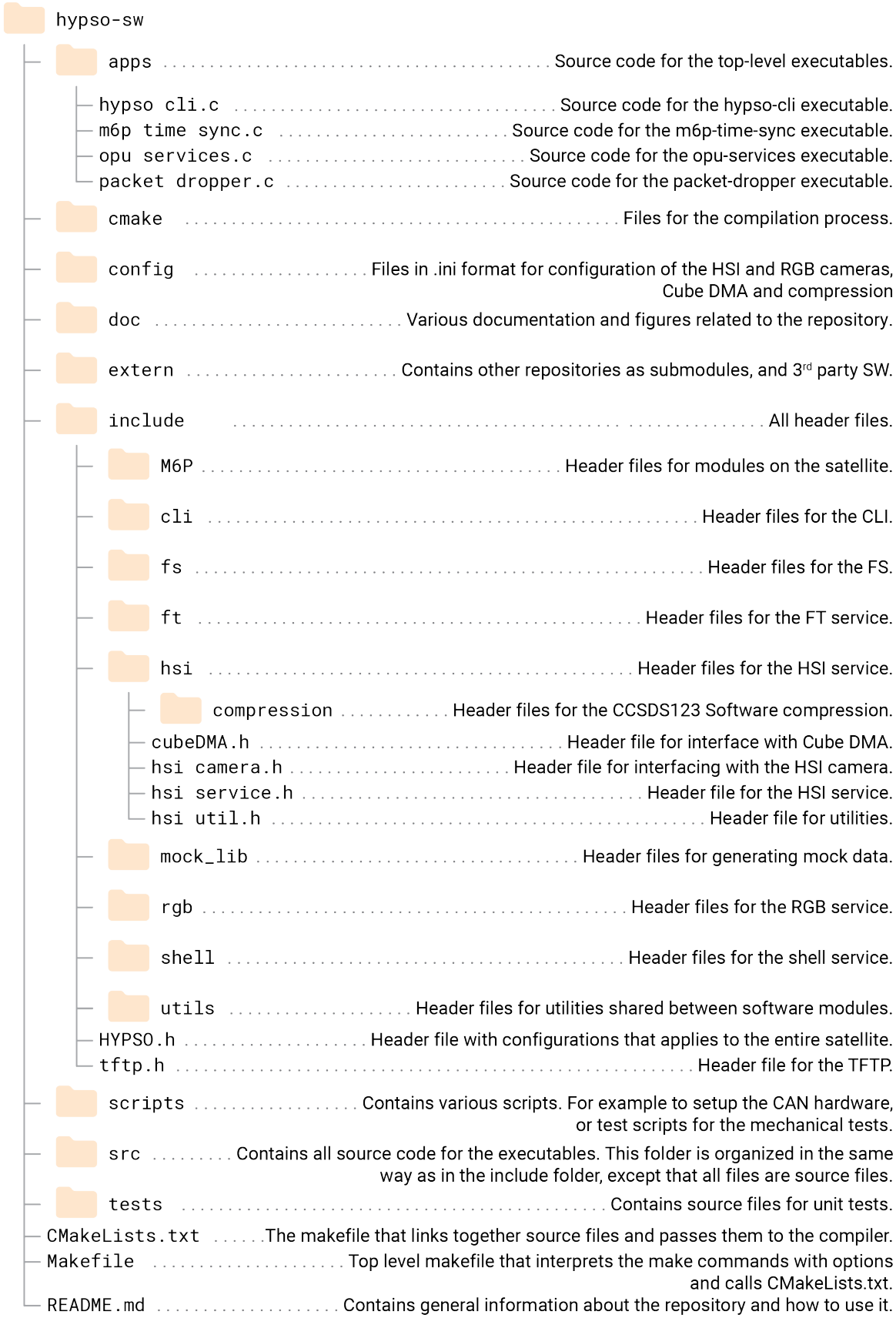


Figure 3.7: Folder structure of the hypso-sw repository.

3.2.2 opu-system

This section is entirely based on the repository itself [21], and the experience gained by using it for this thesis.

In this repository, there are instructions and scripts for installing and using the tools for building embedded Linux for the OPU or the ZedBoard. The main tools are Vivado 2019.1, Petalinux 2019.1, and the hypso-sw Docker container[5]. Once these are installed it is possible to generate bootfiles from scratch in a consistent way:

1. Generate hardware files.
   (a) Open Vivado
   (b) Under *tools*, choose *run .tcl script*
   (c) Navigate to, and select the script for either ZedBoard or PicoZed.
2. Generate bootfiles[6].
   (a) Navigate to the root folder of the repository.
   (b) Run the `build_all` script with the desired ID and flag:
   `./build_all [id] <flag>`.
   - `[id]` can be an integer between 1 and 20 according to the device list in the README. The ID will determine the static IP of the OS after boot.
   - `<flag>` can be either nothing indicating a standard build for the OPU, `-zed` for ZedBoard or `-dev` for a OPU build used for development[7].

The general structure from the root directory with significant files for this thesis is given in figures 3.8 and 3.9. Two figures must be used to fit in a physical format.

[5] Either from a script in the hypso-sw repository itself (see section 3.2.1) or a script in the docker directory.

[6] Not dependent on step 1, as the repository will contain hardware files that will work.

[7] At the point of writing this thesis, the only difference between no flag and `-dev` is that `-dev` does not start `opu_services` on boot.

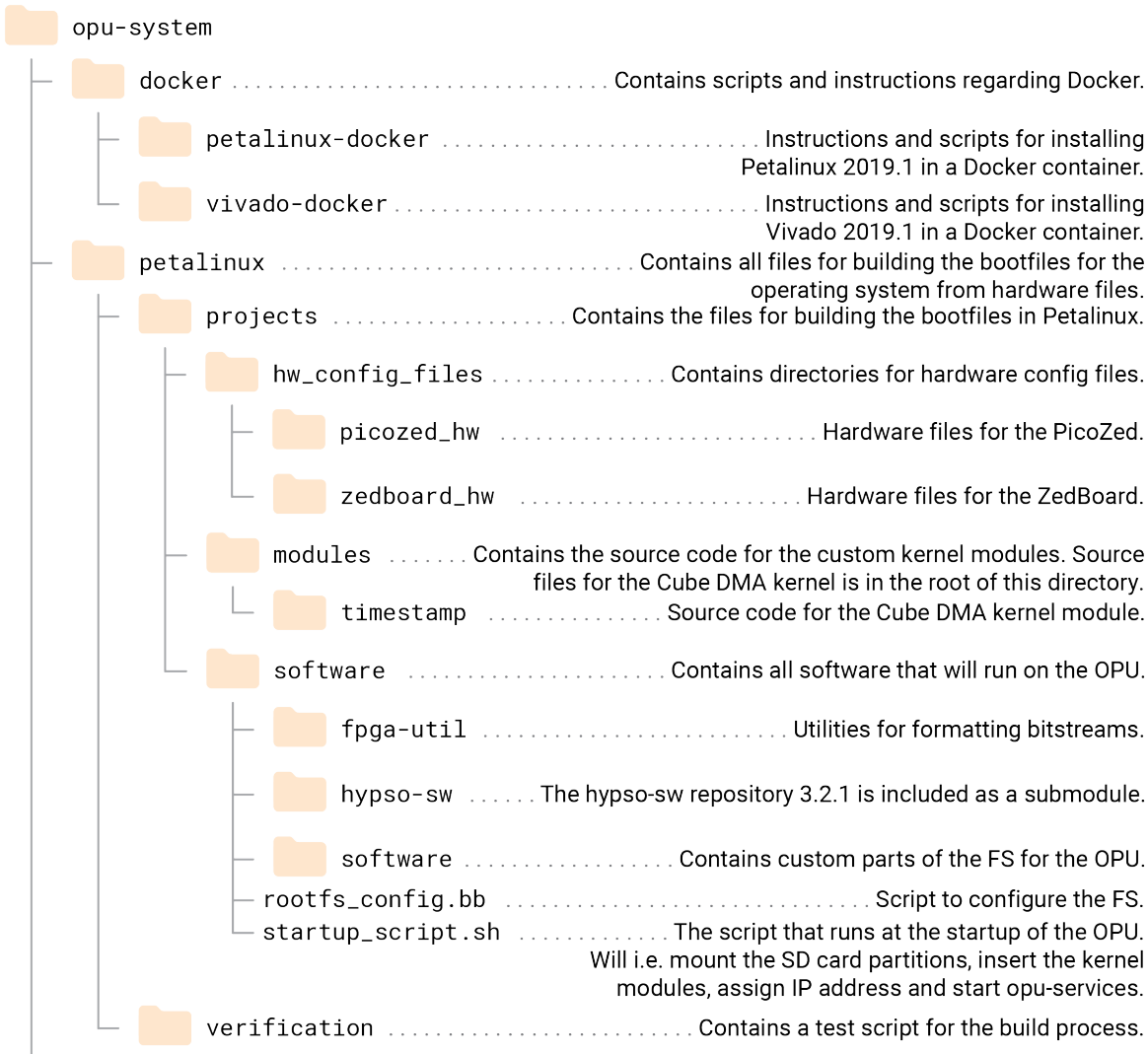


Figure 3.8: Folder structure of the opu-system repository. Pt. 1/2

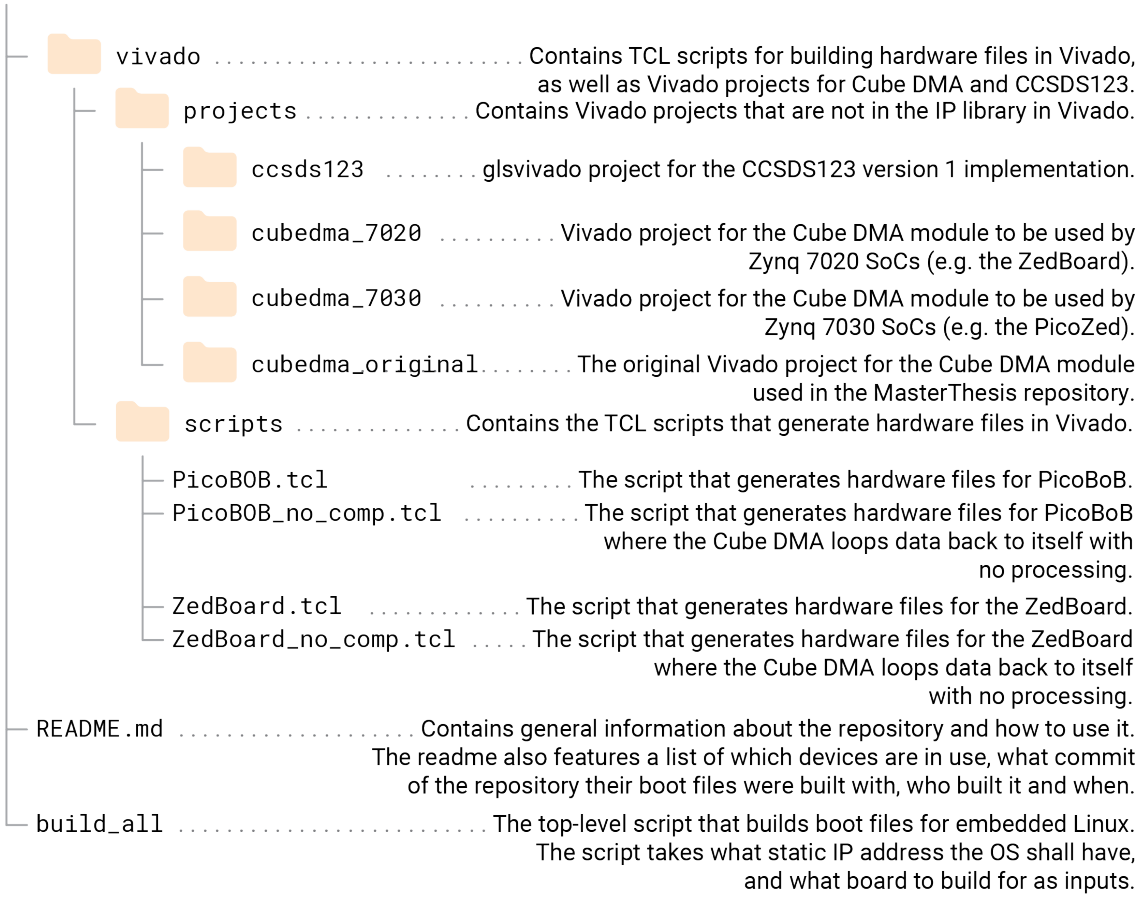


Figure 3.9: Folder structure of the opu-system repository. Pt. 2/2

3.2.3 test-mcus

This section is entirely based on the repository itself [22], and the experience gained by using it for this thesis.

This repository contains information and code for setting up MCUs that are not part of the M6P bus used for test purposes in the SmallSatLab. When writing this thesis, the repository contains two such MCUs. One MCU provides a small oled screen that shows a timer with milliseconds since the MCU powered on. The second MCU generates a PPS (Pulse-Per-Second) signal to emulate the PPS from the GPS on the FC. More on the use of the MCUs in the LidSat steup can be found in section 3.3.1 about the LidSat. The Arduino IDE is used to compile and load code to both MCUs. Arduino provides both MCUs to easily interface with hardware and an IDE to easily write code for and upload to these MCUs [2].

Both MCUs have separate folders in the repository: This gives the general structure shown in figure 3.10.

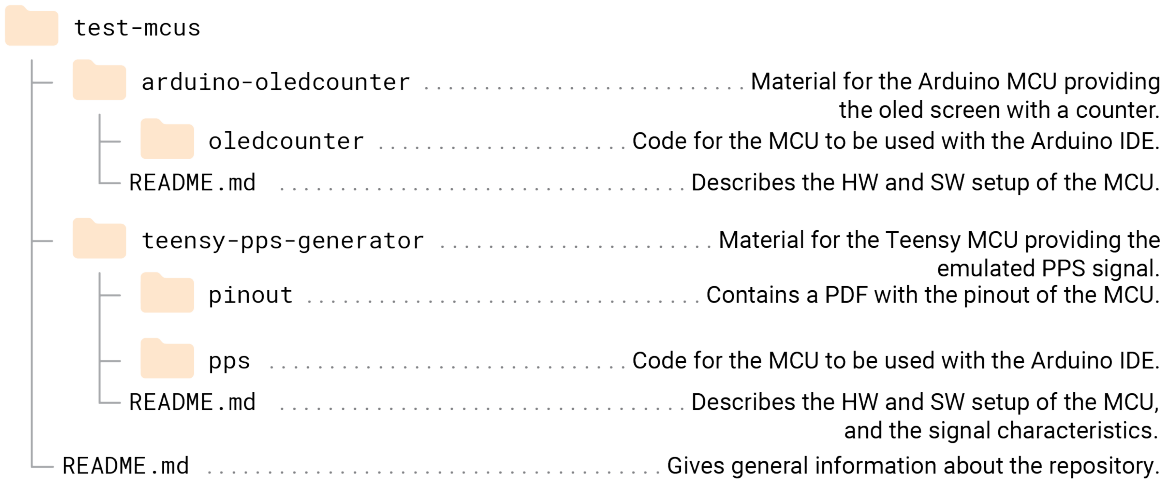


Figure 3.10: Folder structure of the test-mcus repository.

3.2.4 MasterThesis

This section is entirely based on the repository itself [39], and experience gained by using it for this thesis.

MasterThesis is a software repository on GitHub that contains instructions on how to build a basic HSI imaging pipeline for a ZedBoard, and was used for the Master's Thesis *Assembly and Testing of Baseline Processing Chain* [40]. The setup involves several steps, which are explained in the README in the thesis' appendix. The steps are, in short:

1. Build hardware definition files in Vivado 2018.3 featuring the Zynq-7000 PS, Cube DMA and CCSDS123
2. Using these files to make a bootloader and file system for the ZedBoard using Petalinux 2018.3
3. Setup of the camera
4. Extract the kernel from the ZedBoard
5. Instructions on downloading and using an ARM toolchain to cross-compile source code for the pipeline in C++ to an executable.
6. Use the kernel to make a kernel module for Cube DMA
7. Running the program on the ZedBoard

The process was not well documented. Some steps were missing, while other steps could be explained in greater detail. Besides, the entire process was to be done manually. This made the processes hard to follow, and therefore prone to errors. This is discussed more in the project assignment leading up to this thesis [11]. Several persons have been and are still working on improving the process. In the project work leading up to this thesis, the build process for Vivado was scripted [11]. Another member is working on the bootloader for his Master's thesis and has automated and added to this process. The building of the kernel module for Cube DMA has been integrated into this process by yet another Master's student, who re-wrote parts of the module.

The C++ source code had options on how to operate the camera, and how to process and format the captured HSI cube [39]. The codebase was developed for testing the performance of the processing chain in different scenarios to be used in a Master's thesis last year. Although not intended to be used by others than the MSc student who wrote it[8], it has served as a reference for camera interfacing, binning and interfacing with Cube DMA for work on his thesis.

[8]From a conversation with a PhD candidate on the team

3.2.5 Binning

This section is based on the way *binning* was implemented as a part of the Master's thesis *Assembly and Testing of Baseline Processing Chain* [40].

The name *binning* comes from the analogy of putting data in a bin. A binning algorithm works by evaluating a set of data and representing this set by a new value. The mean or median value of the binned data are examples of representative values. Depending of the amount of data in the bin, a dataset can be greatly reduced in size.

The HSI camera displays spectral information along the x-axis of a frame. Pixels close to each other in the y-direction represent similar data. This allows data in the y-direction to be binned without losing significant amount of data, as long as the bins are not too big. In the Master's thesis [40], the size of bins was set to 12, and the bins were represented by the mean value of the bin.

The ARM PS on the PicoZed and ZedBoard allows for SIMD (Single Instruction, Multiple Data) operations. These operations performs the same instructions on multiple data simultaneously [40], having the potential to speed up the binning algorithm.

3.2.6 uEYE Camera Interface

To interface with the uEYE cameras, an API made by iDS is used. This API contains several functions that can be used to handle the various cameras iDS produces. The HSI camera on HYPSO is connected to the OPU with a gigabit Ethernet connection. Communication over Ethernet is abstracted away and encapsulated in the API, giving the user a set of verbose functions, structs, parameters, and error codes to work with. The implementation of these is the IP (Intellectual Property) of iDS, and not accessible to a user.

Both the OPU[9] must be on the same subnet as the camera, with the same subnet mask. The firmware on the camera must be of the same version as the API. Firmware can be uploaded using the iDS Camera Manager.

Table 3.1 gives an overview of the functions used in the HSI source code, and what they do. When relevant, the configuration used on the OPU is stated. If parameters can be chosen this is also stated. The functions are grouped thematically, and not necessarily in a sequence that can be used for operating the HSI camera. Full documentation with inputs, outputs, and error codes, as well as all available functions in the API can be found in the uEYE manual [24].

Function	Functionality
`is_SetErrorReport()`	Enables or disables error report. Error report is enabled by the OPU.
`is_InitCamera()`	Initializes the camera.
`is_ExitCamera()`	Disables the camera and frees data structures and memory regions not already freed manually.
`is_PixelClock()`	Gets pixel clock range or sets the pixel clock on the camera.
`is_SetDisplayMode()`	Sets display mode. The satellite will not use display, so the data is configured to be sent to the RAM

[9] Or another host computer

`is_ImageFormat()`	Set or get image format i.e. resolution. We set image resolution mode 36 (full resolution) according to the table of supported image profiles for image sensor UI-2561 in the manual.
`is_AOI()`	Sets the size and position of AoI (Area of Interest) within the sensor. The AoI is determined by rows and columns. Rows can be chosen as an integer between 2 and 1216 with increments of 2, and columns between 96 and 1936 with increments of 8. The AoI is a configurable camera parameter by the HSI service.
`is_Exposure()`	Sets or gets the exposure time. The exposure time is a configurable camera parameter by the HSI service.
`is_SetColorMode()`	Sets color mode. We use 12-bit monochrome mode.
`is_SetGainBoost()`	Enables an additional analog hardware gain boost feature on the sensor.
`is_SetHardwareGain()`	Sets sensor gain[10]. The gain is a configurable camera parameter by the HSI service.
`is_AllocImageMem()`	Allocates memory for one image on the host computer.
`is_AddToSequence()`	Allocates memory for image allocated by `is_AllocImageMem()`.
`is_FreeImageMem()`	Removes a single image memory from the sequence that were added using `is_AddToSequence()`.
`is_ClearSequence()`	Removes all image memories from the sequence that were added using `is_AddToSequence()`.
`is_SetFrameRate()`	Sets the FPS (Frames Per Second) in freerun mode. The desired frame rate is sent in as input, and the nearest possible value is returned through a pointer[11]. The exposure time set will be corrected to $\frac{1}{\text{FPS}}$ if it is larger than $\frac{1}{\text{FPS}}$. The FPS is a configurable camera parameter by the HSI service.
`is_SetExternalTrigger()`	Configures the use of an external trigger signal for frame captures. Freerun mode is enabled by setting the external trigger off.
`is_CaptureVideo()`	Starts frame capture in freerun mode.

[10] Note that signal gain will result in noise gain

[11] The FPS available depends on various parameters, i.e. AoI, color mode and pixel clock.

`is_StopLiveVideo()`	Stops frame capture started by `is_CaptureVideo()`.
`is_CaptureStatus()`	Returns error codes and count of present errors during capture.
`is_GetActSeqBuf()`	Gets the address to the most recent frame in the frame buffer.
`is_GetFramesPerSecond()`	Returns the measured FPS during capture.
`is_GetVsyncCount()`	Reads the VSYNC counter, which increments with 1 each time the camera starts a frame capture.
`is_EnableEvent()`	Enables the API to signal the OS when a chosen event has happened. We enable frame events, making the API signal the OS when a new frame is ready in RAM.
`is_WaitEvent()`	Makes the OS wait for an event signal indicating that a new frame is available.
`is_DisableEvent()`	Deactivates event signaling.
`is_UnlockSeqBuf()`	Unlocks a previously locked image memory in order to make it available again for storing images. The image memory is re-inserted at its previous position in the sequence list.
`is_IO()`	Configures the flash and GPIO (General Purpose IO) pins on the camera. We enable the flash return from the camera. It is configured to have a rising flank at the start of exposure and last for 10ms[12].
`is_ImageFile()`	Stores a given frame in a given image format. We can enable the HSI service to store single frames in PNG (Portable Network Graphics) format.
`is_DeviceInfo()`	Returns a struct with information about the camera and sensor[13]. For the HYPSO project, camera temperature and Ethernet link speed are interesting.

Table 3.1: Functions from the iDS API used in the HSI source code

[12] In freerun mode, it is not possible to configure the flash signal to have falling flank at the end of the exposure, hence an arbitrary value has been chosen.

[13] The camera does not need to be initialized before using this function.

3.2.7 Cube DMA

This section is based on the Master's thesis *Testing of Communication between Various Peripherals on ZEDBOARD* [13] (unless stated otherwise) of a former student on the HYPSO team, as the design and implementation was a part of said thesis.

In general, a DMA (Direct Memory Access) is used to move data between different HW (Hardware) modules. This includes moving data between the PS and the FPGA on the SoC featured in HYPSOs OPU. Three specifications were given for a DMA to be used for HSI data on HYPSO:

1. Capability of streaming a HSI cube (stored in BIP (Band Interleaved by Pixel) format) in BIP and BSQ (Band Sequential) order.

2. Capability of streaming a HSI cube block-wise.

3. Support for components of sizes that are not byte multiples, e.g. 10 or 12 bits.

None of the DMA cores provided by Xilinx satisfy all three specifications. This lead to the creation of the Cube DMA, which in addition to implement the requested functionality increased performance. The throughput of block-wise data was increased by 128% compared to the DMAs by Xilinx.

During imaging the Cube DMA will be configured using register interfaces. In Vivado, the base address of these registers are set (for example to `0x43C00000`, as specified in [40]). Tables 3.2 and 3.3 describe register interfaces for MM2S (Memory Map to Stream) and S2MM (Stream to Memory Map) respectively. The registers are offset by the given values from the base. By writing data to given bits or bit ranges, the Cube DMA will be configured accordingly. This is done from the HSI service. See listing 1 in appendix BP.1 for an example of how to configure these registers. The referenced listing is not from [13], but a result of work done on this thesis. Note that the registers are not accessed directly with the given hex value, but with integers from the first `enum` in the listing.

The Cube DMA reads data from the memory address specified by the MM2S *base addres* register. How much data it reads is specified by the MM2S *cube dimension* register See table 3.2. Data returned from the FPGA is written to the memory address specified by the S2MM *base address* register, and the number of bytes received is written to the S2MM register *received length*. See table 3.3.

To be able to write to and read from these specific memory ranges, the Linux kernel must be configured to allow this. This is done by writing a custom *kernel module* which is loaded into the OS. How these kernel modules work or are configured is not a part of this thesis.

MM2S Register (offset)		
Field	Description	Bit nr.
Control Register (0x00)		
Start	Core starts transfer when this bit transitions from 0 to 1	0
Block-wise mode	Cube is read in blocks of specified size	2
Plane-wise mode	Cube is read plane-wise, with a given number of planes in parallel	3
Error IRQ enable	Trigger IRQ when error condition arises	4
Completion IRQ enable	Trigger IRQ when transfer is complete	5
Number of plane transfers	How many plane transfers to perform	8 to 15
Start offset	Plane offset to start transferring from	16 to 23
Status Register (0x04)		
Transfer done	Core starts transfer when this bit transitions from 0 to 1	0
Error mask	Indicates which errors occurred	1 to 3
Error IRQ flag	Set when IRQ was triggered due to an error. Cleared when 1 is written to this bit	4
Completion IRQ flag	Set when IRQ was triggered due to completion. Cleared when 1 is written to this bit	5
Base Address Register (0x08)		
Base address	The address of the first component in the first pixel of the HSI cube	0 to 31
Cube Dimension Register (0x0C)		
Width	The width of the HSI cube	0 to 11
Height	The height of the HSI cube	12 to 23
Depth	The depth of the HSI cube	24 to 31
Block Dimension Register (0x10)		
Block width	$\log_2$ of the width of each block	0 to 11
Bloch height	$\log_2$ of the height of each block	12 to 23
Last block row size	Number of components in each row of the last block in a row	12 to 31
Row Size Register (0x14)		
Row size	Number of components in one row of the cube	0 to 19

Table 3.2: MM2S channel register interface. Based on table 2.5 (page 29) from [13]

S2MM Register (offset)		
Field	Description	Bit nr.
Control Register (0x20)		
Start	Core starts transfer when this bit transitions from 0 to 1	0
Error IRQ enable	Triggers IRQ when error condition arises	4
Completion IRQ enable	Triggers IRQ when transfer is completed	5
Status Register (0x24)		
Transfer done	Indicates whether the transfer is completed	0
Error mask	Indicates which error(s) occurred	1 to 3
Error IRQ flag	Set when IRQ was triggered due to error. Cleared when 1 is written to this bit.	4
Completion IRQ flag	Set when IRQ was triggered due to completion. Cleared when 1 is written to this bit.	5
Base Address Register (0x28)		
Base address	The address of where to store the incoming stream data	0 to 31
Received Length Register (0x2C)		
Received length	The number of bytes received from start of transfer until TLAST was asserted	0 to 31

Table 3.3: S2MM channel register interface. Based on table 2.6 (page 30) from [13]

3.2.8 CCSDS 123 Compression

This section is based on the article *An Efficient Real-Time FPGA Implementation of the CCSDS-123 Compression Standard for Hyperspectral Images* [14] about the compression algorithm used in the HYPSO satellite.

To be able to downlink an HSI cube from the satellite to a ground station in a reasonable time, it must be compressed to reduce data size. Processing power in the PS on the OPU is limited, meaning compression in PS will be slow[14]. This has lead to the implementation of the compression algorithm CCSDS123 in the SoC's FPGA. This implementation achieved a throughput of 2.350 Gb/s, which was higher than any of the comparable solutions presented in the paper.

3.2.9 Minimal On-Board Processing Pipeline

MOBIP (Minimal On-Board Processing Pipeline) is a term that includes components that make up the minimal viable HSI imaging pipeline on HYPSO. The components of that make up the MOBIP are given in figure 3.11.

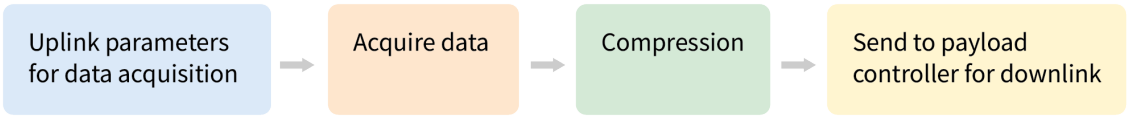


Figure 3.11: Components of the minimal on-board processing pipeline.

The first step is to uplink parameters for data acquisition or *camera parameters*. This can be done either by uploading a file with the configurations using the FT service or a CSP command for HSI capture containing desired parameters. Next comes the step where data is acquired. In this step, the iDS API is used to interface with the HSI camera to capture HSI frames. Binning is also performed in this step. The next step is compression. The HSI data is compressed using the CCSDS123 compression algorithm. The Cube DMA is used to move HSI data to and from the FPGA. Last comes the downlink step. The FT service is used to downlink the HSI cube via the PC.

The MOBIP will eventually be extended to BOBIP (Baseline On-Board Imaging Pipeline), a pipeline with more advanced image processing. New processing stages are for example correction of atmospheric aberrations and dimensionality reduction [7].

[14]The team has experienced that compression of cubes in PL takes about half an hour, finishes in under a second one the FPGA.

3.2.10 Timestamping

To be able to relate a captured HSI cube to an area on the Earth's surface, the time of capture needs to be known. The time of capture is related to satellite telemetry such as position in orbit and satellite angle to find the area covered by the camera at the time. Two BSc students worked this semester to implement a module for timestamping of HSI frames for their thesis *Time Synchronization of Hyperspectral Image Capture on board a Nanosatellite* [28].

The module developed by the BSc students consists of a microsecond timer in the FPGA, a kernel module for handling interrupts and communication with the timer, and an interface to communicate with the kernel module. The module uses the PPS signal from the FC in the M6P bus and a flash signal from the HSI camera.

The timer counts microseconds and is re-started at the rising flank of the PPS signal, which happens once every second with high precision. A second counter is incremented with every PPS signal. This prevents the microsecond counter from overflowing or drifting. When a rising flank of the flash signal from the HSI camera is detected, the total number of seconds and microseconds is written to memory. The HSI camera is configured to pull the flash signal high at the start of each exposure in the imaging sequence. In this way, the timestamp is recorded for each frame. The timestamps are relative to when the timer is started and must be related UNIX (Uniplexed Information and Computing System) time. UNIX time is the number of seconds since midnight on the 1st of January 1970, and a universal way of referencing computer time. The timestamps are stored to file. An example of a file with timestamps is shown in listing 3.1.

Listing 3.1: Example file with 6 timestamps.

```
=== System time in UNIX ===
1590507192 896689332

=== Timestamps in us since system time ===
  3355030
  3480295
  3605559
  3730825
  3856090
  3981354
```

3.3 Setups for Testing

This section describes the different hardware setups used when working on this thesis.

3.3.1 LidSat

The *LidSat* is a setup in the SmallSatLab where different components of the HYPSO satellite are mounted on the lid of an ESD box [8]. Figure 3.12 gives a schematic overview, while figure 3.13 shows the physical setup. The setup contains a PC and EPS from the M6P bus. The other modules of the M6P are located in NAs facilities in Vilnius, Lithuania. These are connected to the LidSat through a virtual CAN bridge over internet.

The HSI camera in this setup has standard optics without a dispersing element. This means that the camera captures regular images.

There are two PicoBoBs connected to the LidSat. This is to have more OPUs in the setup for testing purposes. During the Corona lock-down of campus there needed to be multiple OPUs to access in the lab, as only one person could use one OPU at a time. The two OPUs were assigned CAN IDs and static IP addresses to be able to communicate with them separately.

Figure 3.12 does not include the two MCUs described in section 3.2.3, as they will not be a part of the satellite. One of the MCUs has an OLED screen with a timer. The HSI focuses on this screen so that it is possible to tell separate frames from each other. The second MCU provides an emulated PPS signal to one of the OPUs to be used for timestamping. This is because the PPS signal that will be used in the satellite originates from the GPS on the FC. This module is in Vilnius. To have access to a signal that resembles the PPS, it had to be generated by something in the lab.

3.3.2 PHiL

The *PHiL* (Payload-Hardware-in-the-Loop) setup was intended for HIL (Hardware-in-the-Loop) tests on the OPU. It consisted of a ZedBoard for initial testing and prototyping of the setup and a PicoBoB to be gradually phased in. Both were connected to a computer via CAN. The setup also had a HSI camera. Due to Corona lock-down the PicoBoB was removed from this setup for a MSc student to have for testing. This resulted in the setup only having the ZedBoard available most of this semester.

3.3.3 Home Office

During most of the Corona lock-down, a ZedBoard and a CAN adapter with a PCB (Printed Circuit Board) CAN transceiver was brought to the home office. Although the setup was a bit limited, it was useful for some testing and initial prototyping. Integration still had to be done on the hardware in the lab remotely.

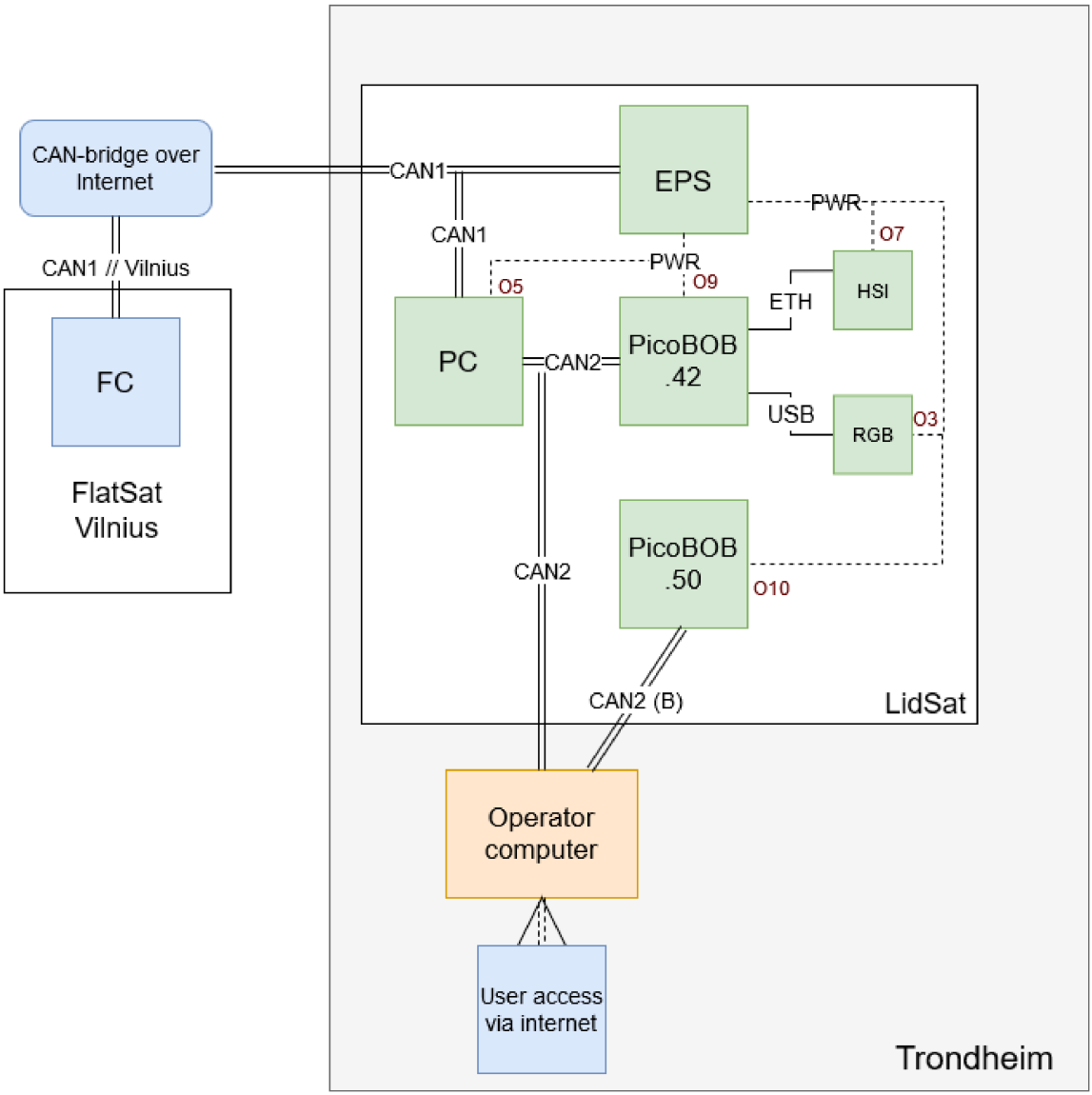


Figure 3.12: Schematic overview of the LidSat from [8]. With permission from Roger Birkeland.

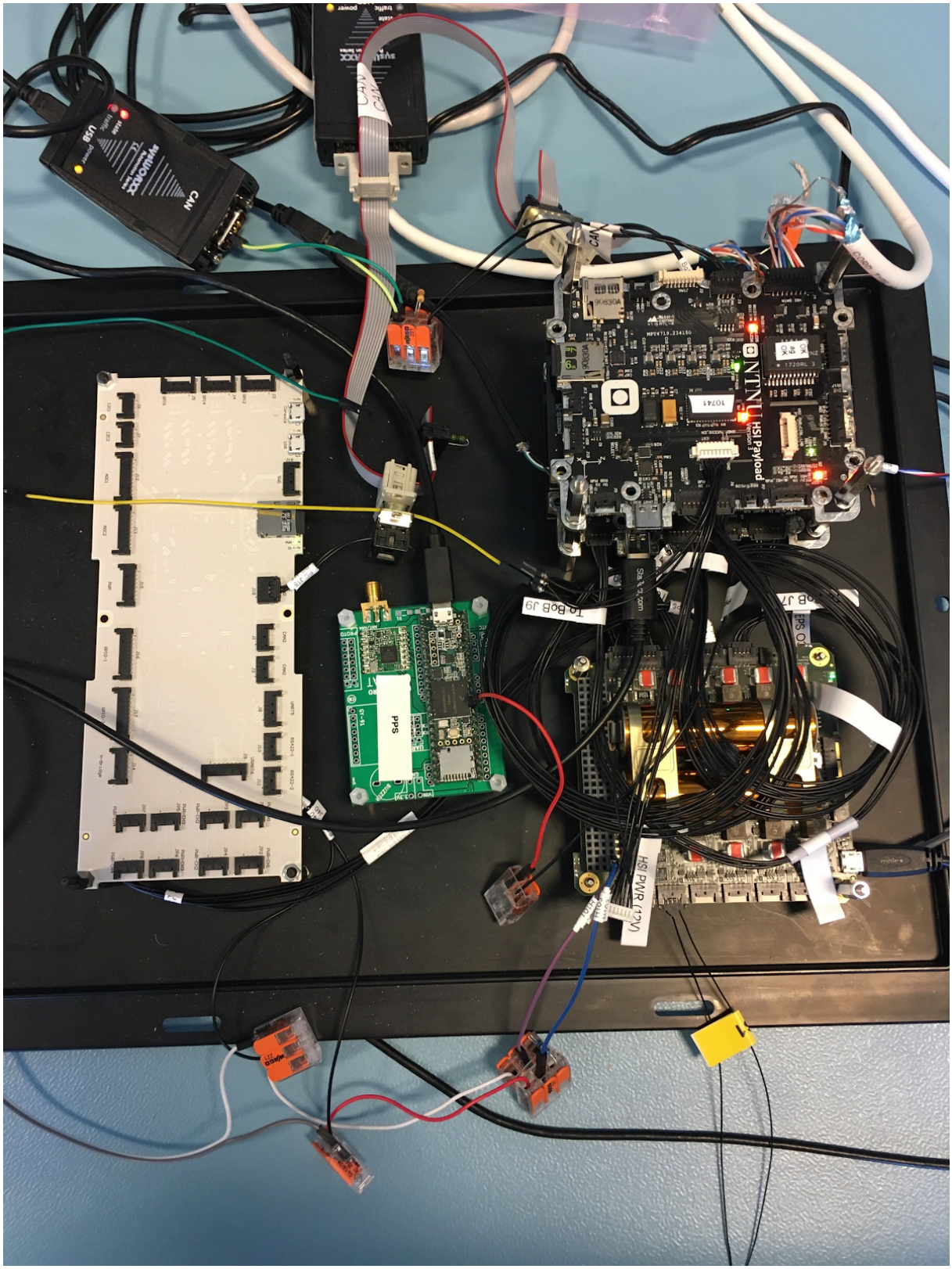


Figure 3.13: LidSat setup. To the left is the PC, in the upper right corner on the lid is PicoBoB and in the lower right corner is the EPS. The figure also shows a MCU that emulates a PPS signal in the middle. Two CAN adapters are shown in the top of the figure. The HSI camera is not shown in the figure. Courtesy of Sivert Bakken.

Chapter 4

Methods and Tools

4.1 GitHub Workflow

This section is based on the internal HYPSO document *Workflow for Software Development* [5] and on experience working in the project. The workflow was introduced at the beginning of the semester, but it took some time for the team to adapt to the workflow.

The document *Workflow for Software Development* describes how GitHub and its tools shall be used by the HYPSO software team to develop software for systems related to the project. One of the things the document specifies is that the team shall use the *GitHub workflow* when developing software. There exists different strategies on how to use *branching* to support the development of a project. Other workflows might operate with additional fixed branches such as *development*, *test*, *deployment* and *production* with rules on how to merge code between them. The *GitHub workflow* aims to have a working master branch with as little overhead as possible. An illustration of the workflow can be seen in figure 4.1. The workflow and additional tools used will be explained in the following subsections.

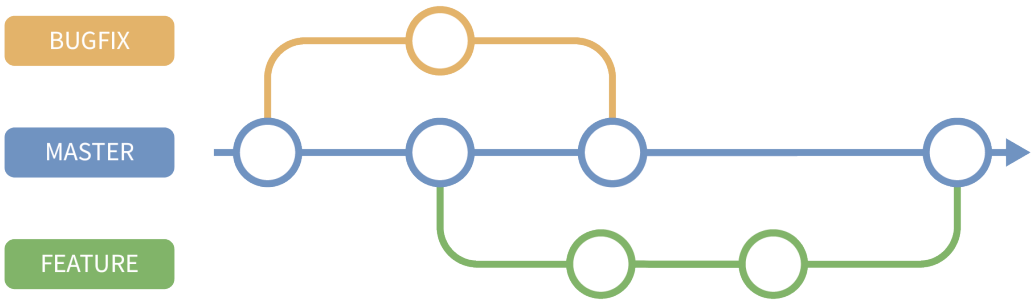


Figure 4.1: GitHub workflow.

4.1.1 Branching

The only persistent branch is the *master* branch, which must be as free of bugs and errors as possible. When new features are to be added, or existing features improved upon, a branch must be created. Development is to be done on this *feature branch* and then merged to master. Branching out to fix bugs is treated in the same way as feature branches. An illustration of branching in the context of the GitHub workflow can be seen in figure 4.2.

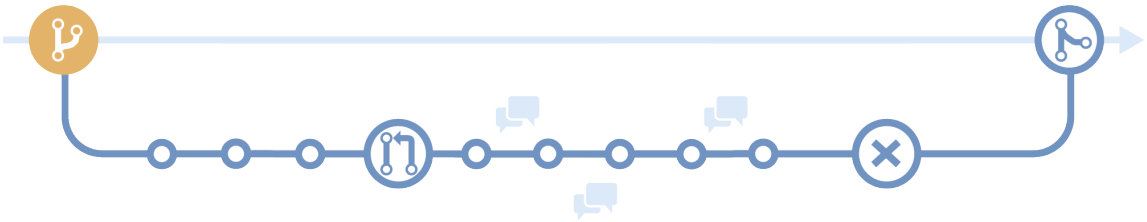


Figure 4.2: Branching in the GitHub Workflow.

Branches shall have descriptive names so that it is clear what feature they are developing/improving or what bug/issue they are fixing.

4.1.2 Commits

Committing and pushing directly to the master branch is disabled for the main repositories hypso-sw and opu-system. This is to help to keep the master branch as free of bugs and errors as possible. Commits shall only be done to feature branches. Frequent committing is encouraged, as version history is important to have if something needs to be rolled back to a known working commit. Committing in the context of the GitHub Workflow can be seen in figure 4.3.

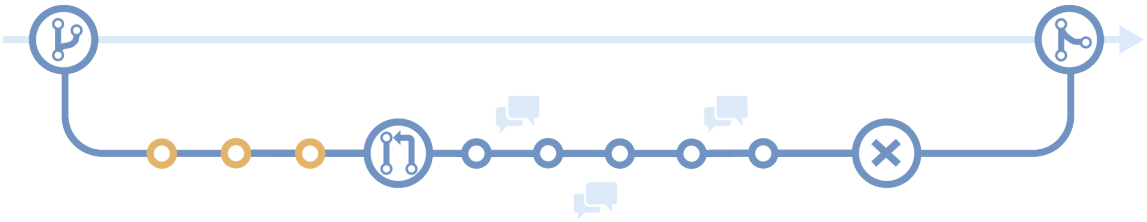


Figure 4.3: Committing in the GitHub workflow.

Commits shall have clear descriptive messages, clearly stating what has been changed with the commit.

4.1.3 Pull Requests

When the developer feels the code is ready to be merged back into the master branch, he will open up a PR. After the PR has been opened, other members of the HYPSO SW team are requested to review the changed code. The person that opened the PR should also write a summary of what has been implemented, which *issues* (see subsection on issues: 4.1.6) it addresses or solves, and what to be aware of when testing. Figure 4.4 shows when a PR is opened in the context of GitHub workflow.

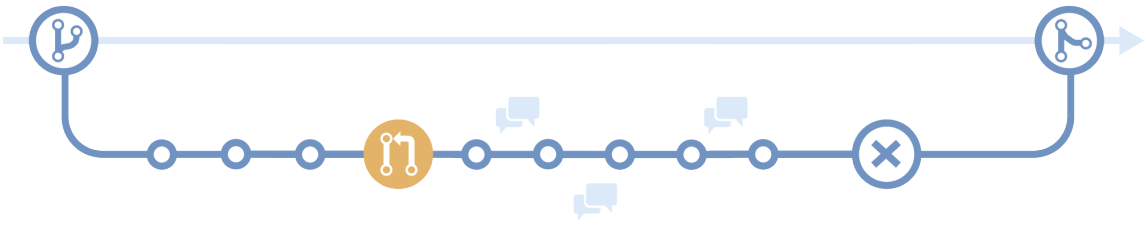


Figure 4.4: Opening a PR in the GitHub workflow.

4.1.4 Peer Review

After opening a PR there is a *peer review*. This section is highlighted in figure 4.5. In this section, other members of the HYPSO SW read and review the code to ensure that the quality conforms with the standards of the rest of the codebase. It is also in this section the process where the contents of the branch are tested. Reviewers should check that the functionality of the branch works when integrated into the rest of the system, and the system still works as it should. Comments from reviewers are added to the PR log. The author has the opportunity to push new commits to the branch during this stage to address the comments from reviewers. When a reviewer is satisfied with the work that has been done, the reviewer can *approve* the request. At a point during the semester, the settings for the hypso-sw repository were changed so that a PR must have at least one approval to be merged with master. This helps to keep the master branch free from errors and bugs as well as unnecessary content.

Examples of PRs are numerous in the appendices. An example of a PR that displays a good review is hypso-sw PR #191 in appendix AB. The PR contains comments from several reviewers. Some had comments regarding the code, others tested the functionality. It was shown that the feature did not work for the PicoZed. This was addressed and solved.

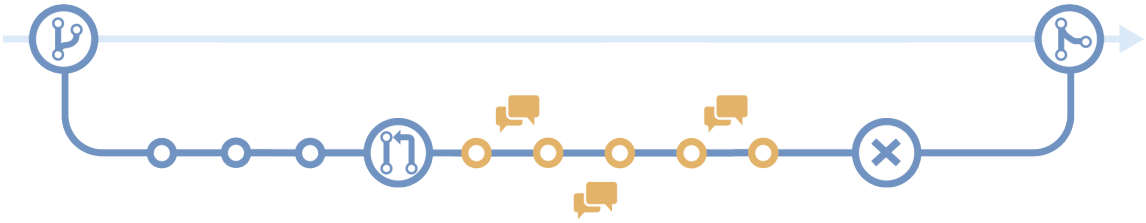


Figure 4.5: Peer review in the GitHub workflow.

4.1.5 Merging

Once a PR has been approved by a person other than the author, the branch can safely be merged with master. This is the last step in the process and is shown in figure 4.6.

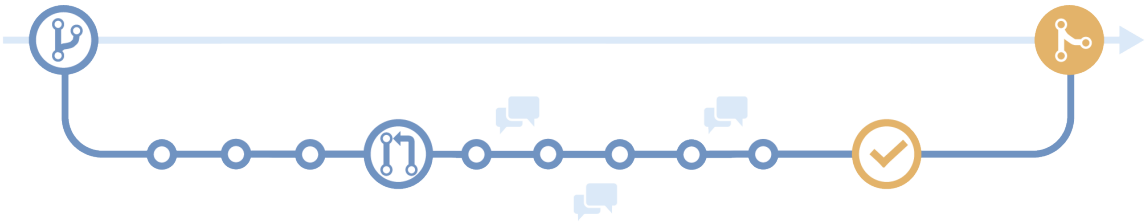


Figure 4.6: Merging in the GitHub workflow.

4.1.6 Issues

As stated in the Background chapter about GitHub (2.3.1), there is a feature called *issues*. This feature can be used in many ways. The most used by the HYPSO SW team are tracking bugs and requesting new features, but issues can also be used to ask questions, state concerns, or draft new ideas. An issue can be commented on by all members of the team to facilitate cooperation, discussions, and troubleshooting.

Examples of issues are numerous in the appendices. An example of a good issue is opu-system issue #108 in appendix BF. The issue explains a bug, how to fix this, and a possible solution. The solution is discussed in the comments, and a person other than the author of the issue found and implemented a better solution.

4.2 Scrum

The HYPSO team has this semester started using the agile development process *scrum*. Scrum is a process for small teams working towards a common goal in a relatively short time-span called *sprints*. Each sprint is preceded by a planning meeting and ended with a review. During the sprints, frequent and short meetings are held to update on the process and discuss problems that arise [34].

All tasks to be done are specified as issues in the appropriate repository. The issues will be assigned *points* estimating the extent of the issue. A 1-point issue has the scope of changing a few, known, lines of code, while a 20-point issue could be a major part of a Master's thesis.

4.2.1 Sprint Planning

The sprint planning meeting starts by assigning points to each issue that has not any points from before. A participant in the meeting will submit his points hidden, and reveal them when all have submitted. If the estimates are not equal, then the meeting must discuss and agree upon how many points to assign the issue. When all issues have been assigned points, the meeting will agree on what issues to focus on in the sprint. The focus of a sprint can for example be to prepare for a specific test or have a feature implemented.

4.2.2 Sprints

The sprints were usually two weeks long. During the sprints, short *stand-up* meetings were held on Mondays, Thursdays, and Fridays. Longer SW-meetings were held on Tuesdays, and all HYPSO team members[1] worked together in the lab on Wednesdays[2].

4.2.3 Sprint Review

When a sprint is over, it is reviewed in a *sprint review*. Tasks that are done will be demonstrated or documentation is shown. The demonstration is usually done for a customer or product owner. Since the HYPSO project is owned by itself, we demonstrate to ourselves. Tasks that are not done will be talked about. Once the review is done, the meeting transitions to a sprint planning for the next Sprint.

[1] Even those not working on software

[2] Until the Corona lock-down of campus

4.3 Working Remotely

As was mentioned in section 3.1.2, the PicoBoB must be protected against ESD. This also applies to the EPS and PC. This hardware must be operated in ESD safe environments. To reduce the need to be in physical contact with this hardware it has been set up to be used remotely. The tools used for this are

- Static IP for all devices connected to the internet.
- SSH (Secure Shell) servers installed on all devices connected to the internet.

This way, the hardware could be accessed from outside the ESD secure area in the lab. The SSH connection gives access to a terminal shell on the host computer, so basic knowledge of shell commands on Linux is a requirement. With the use of VPN (Virtual Private Network), one could also access hardware in the lab from outside the campus network. The VPN establishes an extra secure connection to the campus network, and is required by NTNU for remote connections. These were essential tools to have to work with hardware during the corona lock-down of campus.

Chapter 5

Work and Results

This semester has consisted of 8 sprints, the first started on February 11th and the last ended on June 11th. Work on this thesis has been done both before and after these eight sprints. The duration of these periods are shown on either side of the sprints in table 5.1. This table also shows the start and end date for the individual sprints. The table has a column called *focus*, giving the main focus of the sprint with regards to this thesis. Other participants in the sprint might have had other focuses than what is written here.

Sprint nr.	Start date	End date	Focus
-	January 15th	February 19th	No sprint
1	February 19th	March 2nd	MOBIP
2	March 5th	March 19th	Debugcapture and Modularization
3	March 19th	April 2nd	FT testing and Cube DMA
4	April 2nd	April 16th	Cube DMA
5	April 16th	April 30th	Cube DMA
6	April 30th	May 14th	Cube DMA and Timestamping
7	May 14th	May 28th	Timestamping and Writing thesis
8	May 28th	June 11th	Writing thesis
-	June 11th	June 20th	No sprint

Table 5.1: Overview of sprints

This chapter will feature a section on each sprint, and some sections on work done outside sprints or not directly related to sprints. When a section features a sprint, it will contain subsections with related issues and pull requests. These tables represent issues and PRs related to work done this semester. Some address functionality directly linked to the topic of this thesis, while some are related to work done on the HYPSO project. Issues and PRs that only relate to others' work are not included in these tables. Therefore, these tables do not represent the total amount of work done on the HYPSO project.

5.1 Pre-Sprint Work

5.1.1 Updating Vivado Project for Cube DMA to Work for the OPU

As seen in table 5.1, the sprints did not start at the same time as the semester. Due to this, there were not written any issues on GitHub for this portion of work. The procedure is therefore not as detailed and results were not verified in the same way as for the following sections. Solid proof of functioning Cube DMA was not obtained before later in the semester. See section 5.4.4.

When the interface for Cube DMA developed for the Master's thesis *Assembly and testing of baseline processing chain* [40] this was done for the ZedBoard. The HW dependency originates from the first step in Vivado, where the SoC chip is specified. To be able to interface with the Cube DMA for the PicoZed, hardware files with Cube DMA configured for the correct SoC must be made.

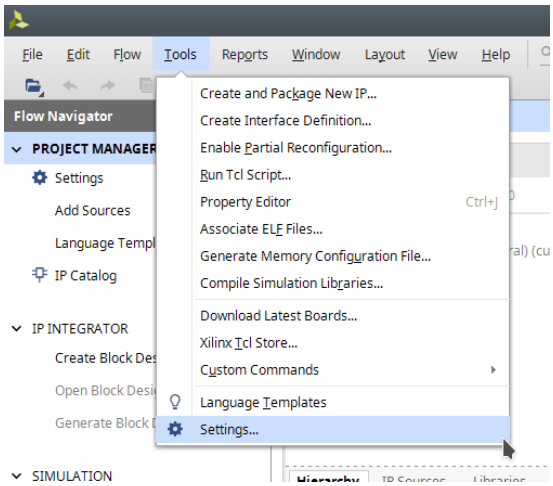
When opening the Cube DMA project for Zynq 7020, the Zynq SoC can be changed under settings as seen in figure 5.1a. The SoC can be changed in the menu shown in figure 5.1b. The correct SoC (see section 3.1.1 for the part number) can be found by the filtering shown in figure 5.1c. After confirming these choices, select the option to re-package IP to update the project to be compatible with the OPU.

5.1.2 Static IP Addresses

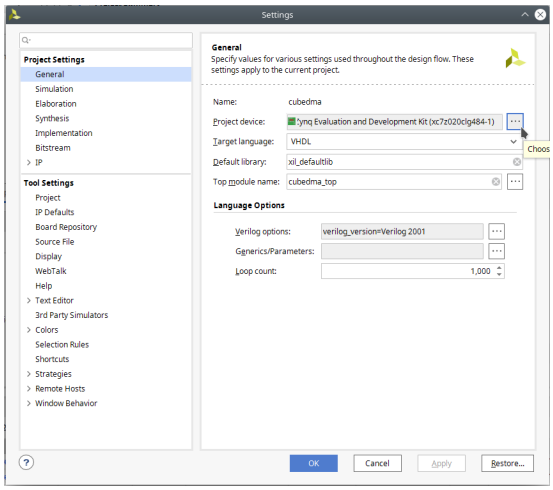
To be able to work remotely with SSH, the target computers IP address must be known. Computes can get a new IP address on reboot. To prevent this, they can be assgned static IP addresses. The network on NTNU is maintained by an organization known as *Orakel*, and a request for static addresses must be granted from them. A request for 37 IP addresses reserved for the SmallSatLab was granted on January 30th.

5.1.3 Troubleshooting USB Interface for RGB Camera

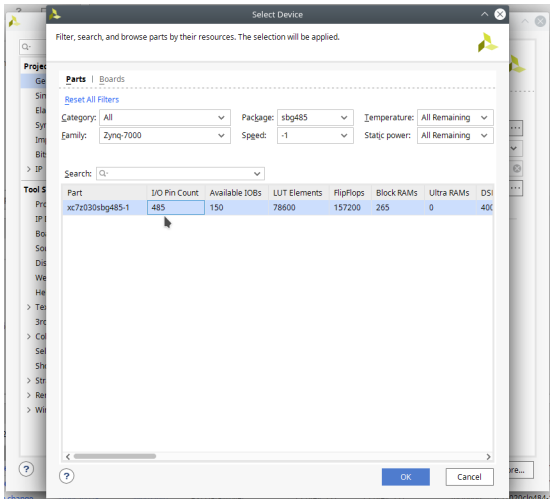
Issue #39 from the opu-system repository (appendix AR) describes an error where the RGB camera could not be detected by the PicoBoB in the PHiL setup in the lab. The student who opened this issue was working on the setup op the OS of PicoBoB, and connection to the RGB was not working. With the insight gained from constructing the `.tcl` scripts last semester [11], the setup of MIO speeds was looked into. This seemed to have been a partial solution to the problem, as the student lists it as one of the things that were done to solve the issue.



(a)



(b)



(c)

Figure 5.1: Procedure of changing the SoC in the Vivado project for the Cube DMA.

5.2 Sprint 1: HSI Integration

The goal of the first sprint was a functional test of the MOBIP from an operational standpoint. This test was carried out on March 3rd and 4th, the days between sprints 1 and 2 (see table 5.1). Camera parameters were to be uploaded to capture an HSI cube which was to be compressed and downlinked. All these steps were to happen based on CSP commands to the OPU.

The functionality for interfacing with the HSI camera and the CCSDS123 compression core in the FPGA on the OPU was located in a different repository than the source code for sending and receiving CSP commands and file transfer (see sections 3.2.1 and 3.2.4). As was discussed in the *Future Work* section in the project report [11] leading up to the Master's thesis, this functionality needed to be migrated to the hypso-sw repository to be integrated with the services on the OPU. This work is described in subsection 5.2.1.

In order to have a backup solution for compression on the FPGA, a software version of CCSDS123 was to be implemented and integrated. Another student on the team was responsible for the implementation. The integration is written about in subsection 5.2.2.

All work in this sprint was done on the ZedBoard in the PHiL setup (section 3.3.2).

Issues in the Sprint

The issues that were included in the first sprint are found in table 5.2. All these issues, except #32, are related to the migration of the HSI source code. No issues were written for the software implementation of CCSDS123. In the sprint planning meeting, issue #54 was given infinite points because it was deemed a too big topic for one single issue. Instead, the issues #79 through #83 were created. These issues split up the original into smaller and more manageable issues. Issues #88 and #89 were both opened and closed during the sprint, and therefore not given any points.

Repository	Issue #	Title	Points	Appendix
hypso-sw	54	Integrate services for the HSI to the CLI	∞	C
hypso-sw	76	Create CSP command for uploading configuration parameters for HSI camera	3	D
hypso-sw	79	Create function to initialize HSI camera	1	E
hypso-sw	80	Create function to configure HSI camera	3	F
hypso-sw	81	Create function for frame captures	3	G
hypso-sw	82	Configure interface with Cube DMA	8	H
hypso-sw	83	Create function to exit/de-initialize the HSI camera	1	I
hypso-sw	84	Create CSP command to capture HSI cube w/ RGB capture for georeferencing	13	J
hypso-sw	85	Create CSP command to capture n frames w/o georeferencing	5	K
hypso-sw	88	Consider threading of binning HSI frames	-	L
hypso-sw	89	Addition of SIMD neon operations for ARM makes ARCH=x86 not compile	-	M
opu-system	32	Option to enable static IP	5	AQ

Table 5.2: Issues in sprint 1

Pull Requests in the Sprint

Table 5.3 shows the two PRs in the first sprint. PR #111 merges the SW version of CCSDS123 from its feature branch into the branch with HSI integration.

In this sprint, efforts to make the HSI source code work were made to the last minute before testing started. As a result of this, the source code was first pushed straight to master without a PR. To comply with the agreed-upon workflow, PR #119 was made after the sprint to have a review. A rule forbidding merging to master in the hypso-sw repository without a PR was made to the repository at this point. The opu-system repository did not get this rule until later in the semester.

Repository	PR #	Title	Related issues	Appendix
hypso-sw	111	HSI service SWcompression	-	N
hypso-sw	119	HSI service pull request	#79 #80 #81 #82 #83 #85 #88 #89	P

Table 5.3: Pull request in sprint 1

Issue #76 in hypso-sw was later closed because it was no longer relevant, issue #84 in hypso-sw is not yet implemented, and #32 in opu-system was closed without a PR.

The reason for the column *Related issues* being named as it is, is that these issues were not closed with the PR. This is because there was no validation of the camera interfacing at this point. No valid proof of correct camera interfacing was given until later, when the pipeline could produce images in PNG format.

5.2.1 Migrating the HSI source code

Migrating the source code to the hypso-sw repository would make it more accessible. All services would then be a part of the `opu-services` executable, whereas the HSI functionality had its own executable before. Another benefit of moving it is that it will be compiled using Docker along with the rest of hypso-sw instead of being dependent on its own compilation toolchain. This would also lead to increased accessibility, as one had to follow a step-by-step guide to finding, installing, and using the correct toolchain in the original repository.

The original source code for the HSI was written in the programming language C++, whereas all other code for the OPU is written in C. C++ is a language meant for object orienting, and is built on C. This makes the two languages quite similar, although not compatible. All the original source code had to be translated to regular C to be integrated into the correct repository.

To be able to make the HSI functionality a new service on the OPU, the following items had to be done:

- All new source files were added to the new directory `src/hsi/`.
  - The top-level HSI source code is located in `src/hsi/hsi_service.c`.
- All hew header files were added to the new directory `include/hsi/`.
  - A file `src/hsi/hsi.h` including the port for CSP commands was also required.
- A new source file for CLI commands was added: `src/cli/hsi_cli.c`.
- The corresponding header file was added: `include/cli/hsi_cli.h`
- The service was added to `opu-services` by including the lines:
  `{ .thread_name = "CLAW-1 Payload Service",`
  `.start_routine = hsi_service_task},`
  to the `struct service_context service_contexts[]` in `apps/opu_services.c`.
- Add the HSI commands to the CLI by adding
  `cli_hsi_init_cmds(&root);` towards the end of the function
  `static void cli_cmds_init(void)` in `apps/hypso_cli.c`
- Link all `.c` source files to executables in `CMakeLists.txt`.
- And `#include` all header files to relevant source files.

Issues #79 through #83 dictated how the base modules of the service should be divided. The camera interfacing was to be divided into five separate functions, each with its specific purpose. The function to configure the camera is shown in appendix BP.2, listing 2. To be usable by the CLI/services pair, these lower-level functions must be called by a higher-level function through the CLI. Two such functions were envisioned in issues #84 and #85. Only #85 was implemented in this sprint, but not as a function. In the top-level `.c` file, the CSP commands are interpreted in a `switch` statement. The functionality in #85 was a case in this statement. The contents of the case is shown in listing 3 in appendix BP.3. A user could send a CLI command to start a HSI capture with a specified number of frames: `hsi capture <frames>` to trigger the case. The content of these functions and cases has changed quite a bit during the semester, but the core functionality is still the same.

The HSI camera was envisioned to be configured using a separate file in the FS on the OPU. Issue #76 was written to address this, but it was later closed since the procedure of setting camera parameters has not yet been formalized. The functionality described in issue #84 is still to be implemented.

While working on integration, a problem with multithreading emerged in the binning method. The `pragma omp parrallell` command was used to start multithreading of the binning. Issue #88 was opened to address this. With help from other members of the team, a solution was found and implemented. The solution is documented in the issue. A new error was discovered when trying to compile the CLI executable for x86 architecture. The toolchain tried to compile ARM specific instructions for x86, and returned an error. The problem is described in issue #89, where a solution was found and documented with the help of team members.

Two issues came up during migration and integration that were not solved during the sprint. The first was hypso-sw issue #117 about Cube DMA not working as it should. This error was addressed in the first MOBIP test and was the main focus for sprints 3 through 6. The second issue was #120 about modularization. This issue was the main focus for sprint 2. Both of these issues are mentioned in comments in the HSI integration PR #119.

5.2.2 Integrating a SW-Version of CCSDS 123

Should the HSI cube not be compressed successfully in the FPGA for some reason, a backup version in SW should take over. This is a backup solution, as the FPGA will compress the cube much faster than the PS.The code was developed by another student in the HYPSO SW team, and was integrated to the HSI service while the service itself (section 5.2.1) was being migrated.

This backup solution for compression was also to be ready for the initial MOBIP test, and therefore not tested for correctness in the PR. Some concerns were addressed by reviewers. Firstly, the author had used a separate makefile in the toolchain. Secondly, there was suggested some changes to optimize the code. When these concerns were solved, the branch was merged into the HSI service feature branch. This branch was then merged with the master branch as discussed in the previous section.

5.2.3 Initial Test of the Minimal On-Board Imaging Pipeline

The MOBIP consists of elements to configure and perform HSI data acquisition, then compress and downlink the data. This test was done from an operational perspective to assess the pipeline as a whole. The MOBIP test was conducted on the 3rd and 4th of March, at after the first sprint. The test was partially successful, as the testers found that compression in FPGA increased the size of the cube instead of decreasing it [16].

The test failed as Cube DMA did not work, and therefore compression did not work as intended. Although not all print from opu-services is included in [16] this was a known fault and addressed in hypso-sw issue #117 the same date as the test.

The capture command used in this test would store the HSI cube at three points in the pipeline:

- The full cube: A cube with all un-processed frames in BIP format.
- The binned cube: A cube with binned frames in BIP format.
- The compressed cube. This is the final output of the pipeline. The CCSDS123 algorithm has compressed the binned cube.

To be able to verify that camera interfacing is correct, the testers requested that individual frames should be stored during capture. Preferably in both BIP format and a regular image format that a computer can open without further processing. This feature is requested in hypso-sw issue #126.

The MOBIP test was re-done at the end of May, with a much more positive result. See section 5.8.2 for further details.

5.2.4 Static IPs

The availability of IP addresses reserved for use in the SmallSatLab made it possible to build boot images with static IPs. Issue #32 in opu-system was opened to request this functionality. Another student included this as an input to the automatic building script and closed the issue.

5.3 Sprint 2: Modularization

In the second sprint, the main goal was to achieve a better modularization and encapsulation of the HSI source code. The MasterThesis repository (section 3.2.4) repository was not intended to be used by others, as it was implemented to do testing for a Master's thesis [40]. This made the code difficult to work with. Functions and data structures were not named logically, and not in reasonable modules. As a result, the program flow became intricate and hard to follow.

A side goal of the sprint was to construct a CSP command that captures an HSI cube while storing more information during the processing than a regular capture. This was to be used for testing and debugging purposes.

Work in this sprint was done on the ZedBoard in the PHiL setup.

Issues in the Sprint

Table 5.4 shows the issues to be done in the second sprint. Issue #120 addressed the modularization and issue #126 the construction of the new CSP command. A new issue was opened and closed in the sprint, and not given any points. This was issue #133, which stated that functions in the HSI service did not return consistent error codes.

Repository	Issue #	Title	Points	Appendix
hypso-sw	120	Modularize functionality in HSI source code	13	Q
hypso-sw	126	Make a CSP command for capturing frames for debug/testing purposes	3	R
hypso-sw	133	Consistency in returning error codes in hsi service	-	T

Table 5.4: Issues in sprint 2

Pull Requests in the Sprint

In this sprint, all related issues were closed by PRs as seen in table 5.5. PR #128 merged the *debugcapture* feature to the master branch of hypso-sw. A comment in the PR stated that some data allocation could be more efficient, but it was ready to be merged.

PR #143 was approved and merged without significant review. Because of changes that had happened to dependencies on the master branch since this feature branch was forked out, the master branch did not compile after merging. The following PRs amends to that by first reverting the master branch (#146), fixing the bugs on the feature branch (#147), and finally realizing that the master branch should not have been reverted in the first place (#148). These PRs show that bugs lead to errors when not tested properly. The master branch should have been merged into the feature branch and then the feature branch merged into the master branch. If done this way, the bugs would have been found and removed on the feature branch and not the master branch. The PRs also shows that the team responded quickly and effectively to mitigate the errors.

Repository	PR #	Title	Closing issues	Appendix
hypso-sw	128	Debug capture	#126	S
hypso-sw	143	Modularized the HSI service and added documentation	#120 #133	U
hypso-sw	146	Revert ”Modularized the HSI service and added documentation”	-	V
hypso-sw	147	Trying to merge again!	#120 #133	W
hypso-sw	148	Revert ”Revert ”Modularized the HSI service and added documentation””	-	X

Table 5.5: Pull request in sprint 2

5.3.1 HSI Debug capture

The feature requested in issue #126 mentioned in the section 5.2.3 about MOBIP testing was implemented as the CLI-command `hsi debugcapture`. This command would do a capture of no more than 10 frames. Each frame was to be stored in BIP[1] format as well as a format easily readable for a human[2]. The HSI cube was also to be stored after different stages in the pipeline: Un-processed[3], binned cube and compressed cube.

The CLI-command was implemented to call the same high-level function `hsi_capture()` mentioned in 5.2.1 with a flag that in turn would trigger a conditional to save individual frames during capture.

The feature was merged to master early in the sprint, leaving the rest of the sprint for issue #120

5.3.2 Modularization of HSI Source Code

The original code in the repository MasterThesis had two modules: *HSICamera* and *CubeDMADriver*. There existed a *main* module as well, but its only purpose was to trigger functions in *HSICamera*. A class diagram of the modules is shown in figure 5.2. The source code contained a lot of code that was used for various testing and was not meaningful to migrate to the hypso-sw repository. Figure 5.3 shows a class diagram of the same source codebase, but with superfluous functions removed. Identifying useful code was a first step to making better modules, as it is clearer what needs to be done when only the essential parts are visible. Doing this reduced the lines of code in HSICamera from nearly 1100 to just under 500 lines of code.

[1] Called *raw* in the issue.

[2] This means a format that a regular computer can open without any further processing

[3] called *entire cube* in the issue

The program flow of the remaining source code is shown in figure 5.4. There are several items to address:

- The name HSICamera implicitly indicates that the module contains code for the HSI camera and nothing more. Yet, it communicates directly with the kernel, performs binning and stores data in addition to camera interfacing.
- The initialization function is not encapsulated, as it initializes the camera, starts frame captures and acquires memory regions from the kernel. This is too much to do for a function that only should initialize the camera.
- The `runCubeCapture()` function's only purpose is to call `freeRunCapture()`.
- Inside the loop, extraction of AoI and binning should be separated out into their own functions instead of being a part of a larger function.
- The interfacing with Cube DMA has fairly good modularization, but should not be called from the camera module, as the camera itself should not be concerned with what happens with the data it produces.

By implementing hypso-sw issues #79 through #83 in the first sprint (section 5.2.1), the modularization of the HSI source code had started. By addressing these issues and the items above, as well as hypso-sw issue #120 the code quality of the service was improved. Issue #120 is also concerned with the interfaces between the modules. The issue defines a set of data structures that the service should use, and which modules that need access to what data. This defines the interfaces between the modules in the service and makes the flow of information more visible. Although thorough, the issue is not comprehensive. It outlined work to be done in the sprint. Both modules and interfaces have been subject to change over the course of the semester, but the foundation for these changes was laid during this sprint.

During modularization, it was discovered that there was not defined any consistent way to return error codes from functions. Issue #133 in hypso-sw (see appendix T was written to address this. First and foremost to document that a choice had been made for the HSI service, but project management wanted this to be consistent for the whole project. Thus, it is now a part of the introduction of the internal HYPSO document *Software Development*, which states: *Functions returning error codes shall return 0 indicating "success" and errors as integers* [5].

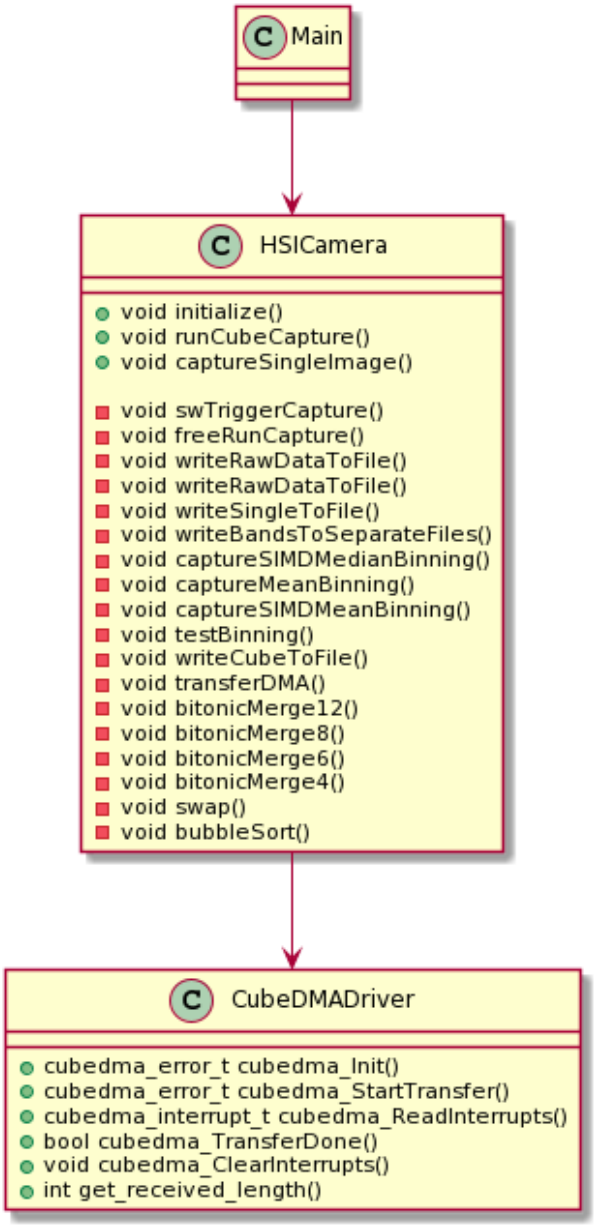


Figure 5.2: Class diagram showing the modules and functions in the original HSI source code

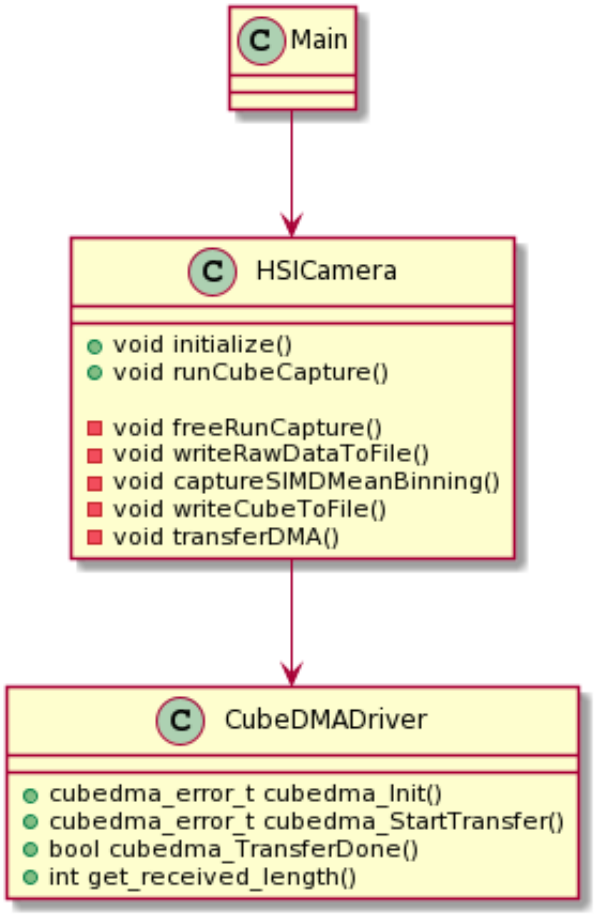


Figure 5.3: Class diagram showing the modules and functions in the original HSI source code after removing unnecessary code

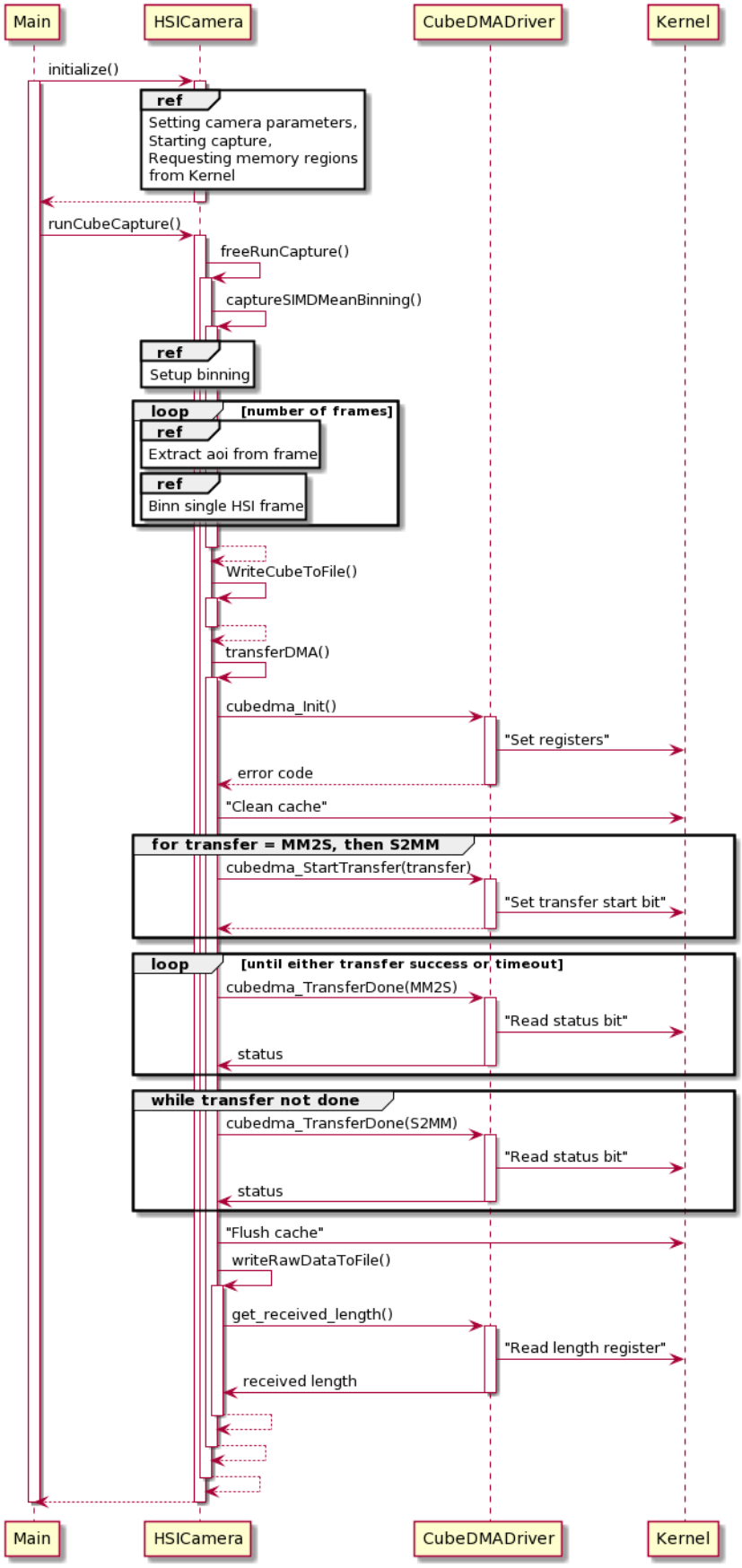


Figure 5.4: Sequence diagram showing program flow in the original HSI source code

After refactoring the code according to what was specified in issue #120 and comments from the list at the beginning of this section the code was divided into four main modules instead of two. The modules and their sub-modules[4] are shown in class diagram 5.5. The top-level module hsi_service has the over-all control logic and is responsible for calling other modules. The module hsi_camera is only concerned with interfacing with the HSI camera and has functionality divided into reasonable functions. These functions do only what their name implies. In the Cube DMA module, not much has changed as it had adequate encapsulation from before. In the last module, hsi_util contains the rest of the functions, which did not fit in other modules. This is where binning, storing, and extraction of AoI were put. The modularization and encapsulation at this point were not perfect, but usable. The hsi_util module is not well defined, and the top-level module should not have to communicate directly with the Cube DMA module. However, the sprint was ending, and a lot of progress was made on the topic of modularization.

The new program flow is shown in two figures, as it had to be split up to fit in a physical format. Figure 5.6 shows the first part of the service. This part contains the initialize and capture process. The next part is shown in figure 5.7, and contains the interfacing with Cube DMA.

By comparing sequence diagram 5.4 for the original codebase with the modularized codebase in figures 5.6 and 5.7[5], it can be seen that dividing up the modules has resluted in a better program flow.

[4] functions

[5] The diagram had to be split in two to fit

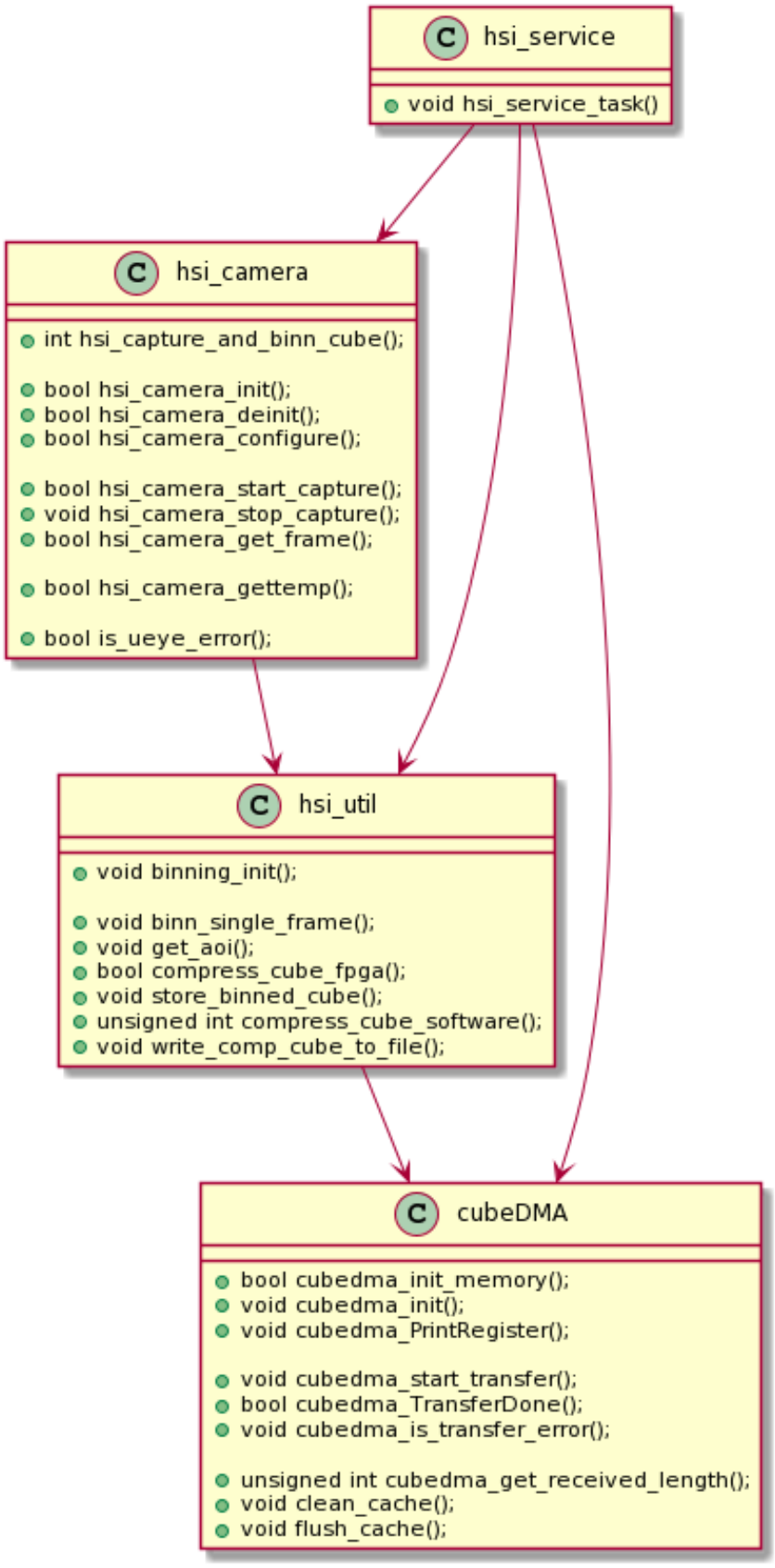


Figure 5.5: Class diagram of the modularized source code

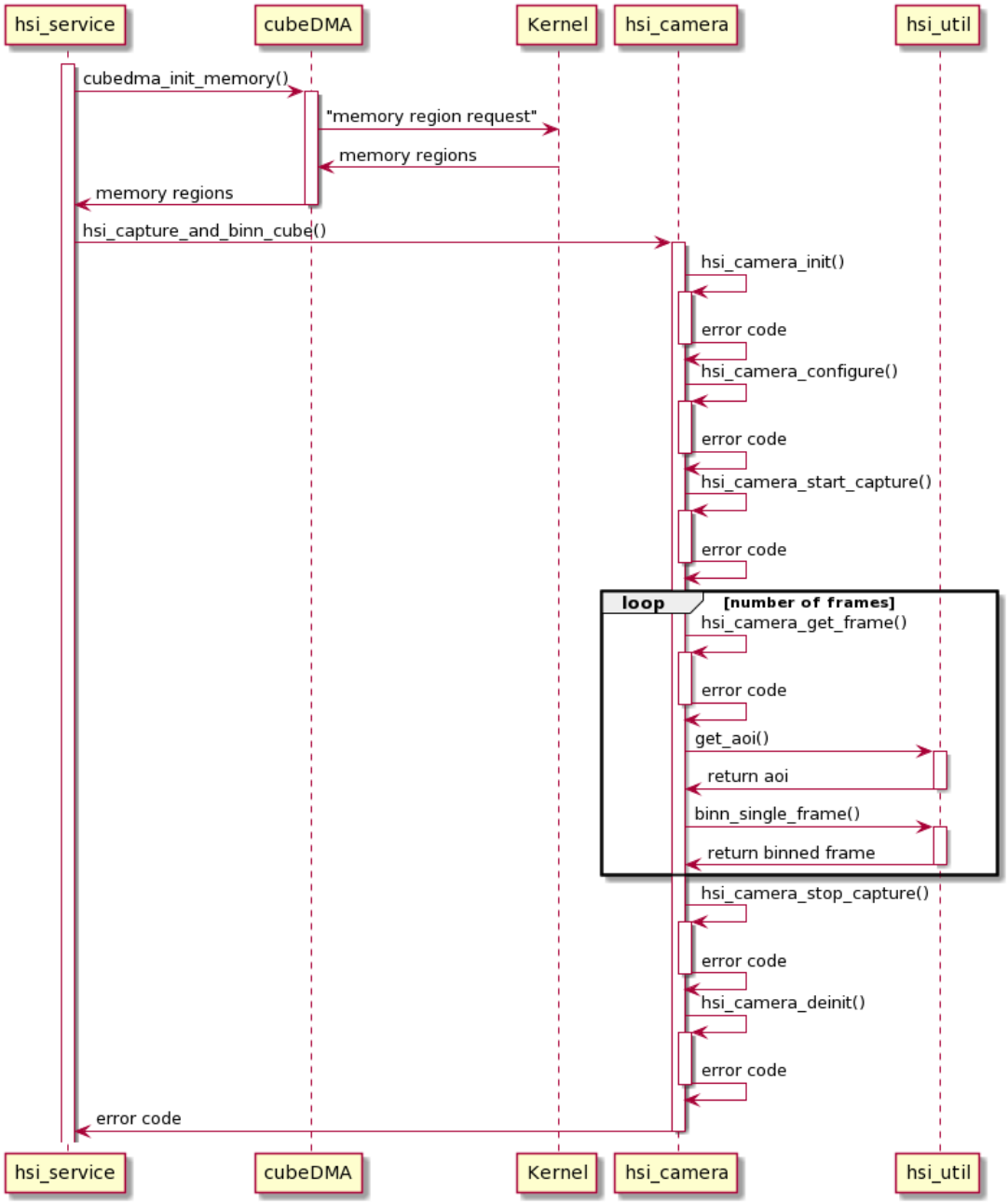


Figure 5.6: Sequence diagram of program flow of camera interfacing in the modularized source code (pt. 1 of 2)

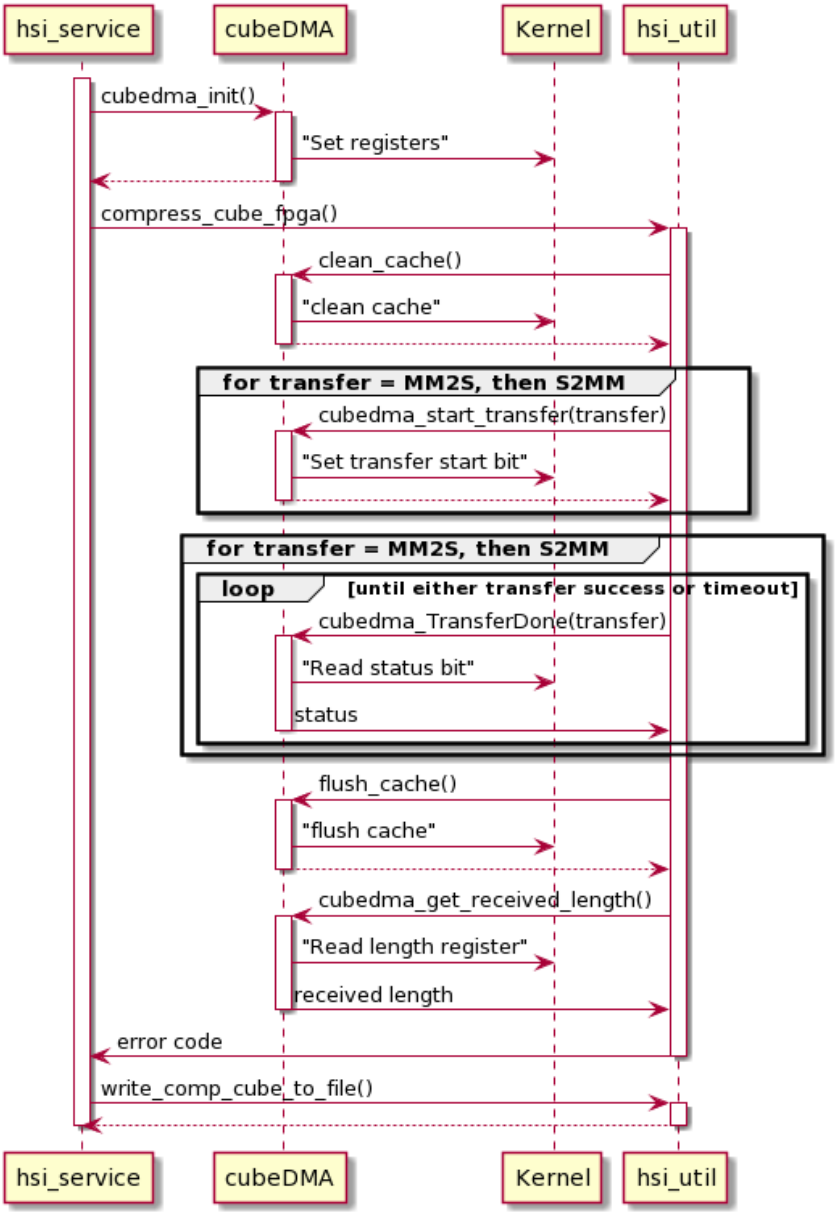


Figure 5.7: Sequence diagram of program flow of Cube DMA interfacing in the modularized source code (pt. 2 of 2)

5.4 Sprint 3: File Transfer Testing and Cube DMA

The third sprint of the semester aimed to have a working interface with the Cube DMA. That is a consistent way to move HSI data to and from the FPGA on the OPU.

This was the first sprint to be done entirely in Corona lock-down. This meant having good functionality and routines to move files between setups and modules in setups was needed. The first part of the sprint was therefore used to help with testing this functionality.

During this sprint, both the ZedBoard in the PHiL setup and the home office setup (see section 3.3.3) were used.

Issues in the Sprint

The main issue for this sprint was hypso-sw issue #117 describing the state of the Cube DMA. Issue #169 from hypso-sw describes a bug where the debug capture saves a binned cube file with no data. The rest of the issues were from opu-system, and opened and closed during the sprint. Issues #60, #62, and #66 describe bugs experienced on the repository, while #65 is related to work on the Cube DMA. All issues are included in table 5.6.

Repository	Issue #	Title	Points	Appendix
hypso-sw	117	Cube DMA interfacing is not working correctly	20	O
hypso-sw	169	Debug-capture saves the binned cube with 0 bytes	5	Z
opu-system	60	Static IP not working	-	AS
opu-system	62	My permissions are not set right?	-	AT
opu-system	65	Cube DMA has ”ECC mode”?	-	AV
opu-system	66	ZedBoard is unable to boot without swapfile	-	AW

Table 5.6: Issues in sprint 3

Pull Requests in the Sprint

The PRs in this sprint are shown in table 5.7. PR #150 from hypso-sw was opened by a team member working the FT service. It is not related to any of the issues mentioned in 5.6, but it is included here as peer review and testing was done. PR #64 is related to the work on Cube DMA, but did not close any of the issues in table 5.6.

Repository	PR #	Title	Closing issues	Appendix
hypso-sw	150	Various Fixes	-	Y
opu-system	64	Fixmemory	-	AU

Table 5.7: Pull request in sprint 3

None of the PL listed in 5.7 closed any issues listed in 5.6. The issues from hypso-sw (#117 and #169) were not closed during the sprint and were carried through to the next sprint. Issues from opu-system were closed in the sprint but without PRs.

5.4.1 Issue With Static IP

When building boot files and trying to enable static IP address for ZedBoard, a bug was encountered. The ZedBoard did not get the desired IP. This bug is described in opu-system issue #60. The comments in the issue show the debugging process with the discovery of the bug and closing of the issue.

5.4.2 Testing the File Transfer Service

The FT service got a major update in this sprint with functionality merged in hypso-sw PR #150. This PR was opened by another team member, and all work was implemented by him. One of the issues solved by the PR was hypso-sw issue #69 (not included in this thesis) which increased the speed of file transfers. The PR also included a new FT command: `opu upload` which automated a cumbersome process. The new functionality was tested, code reviewed and the PR was approved.

With the increased upload speed, it was possible to test a full system update of the OPU. The procedure was to send the files to the computer in the PHiL setup, and transfer the boot files over CAN to the ZedBoard using the FT service. This procedure emulated the procedure of updating the payload system on the satellite in orbit. In orbit, the files would have to be uplinked and stored on the PC and transferred over CAN to the OPU. The procedure was practiced on the home office setup before it was tested on the PHiL setup. During preparations, the opu-system repository was cloned to a new computer in the home office setup. While doing configuration of the repository, there were problems with permissions in Linux. These concerns were raised in, discussed, and solved in issue #62.

The test was partially successful. One of the files was not transferred correctly over the CAN bus, and was transferred over Ethernet instead. The bug was addressed in a comment in PR #150. Due to a missing *swapfile*, the ZedBoard in the PHiL setup did not reboot correctly. This was addressed an thoroughly discussed in opu-system issue #66. It was later replaced by a new issue, #73, which is not included in this thesis. The full test report *HYPSO-TRP-EL-014: Remote Uploading of New Boot Files to Zedboard* is found in appendix A.

5.4.3 Binned Cube Stored Without Data

Issue #169 from hypso-sw describes a bug where the binned cube file is created after capture, but the file is empty. At the time, work was being done on the transfer test described in the previous section. A comment left in opu-system issue #66 shows that the ZedBoard in the home office setup had kernel panics. This in addition to that another member of the team was looking into the binning resulted in only a comment being written in hypso-sw issue #169 about what the solution might be. The PR for this issue was not merged until sprint 5. See table 5.11 in section 5.6 about the PRs in sprint 5. The PR mentions that the solution was given in this issue.

5.4.4 Configuration of Cube DMA Interface

Issue #117 from hypso-sw originated from the first sprint. This issue stated that the Cube DMA interface was not working. This module is meant to move HSI data from the PS to the FPGA for fast processing. When the DMA fails to move data, the binned cube will be the only HSI data available. This data can potentially be very large due to the lack of compression. The lack of a working DMA module will be fatal when more processing modules are introduced in the FPGA (BOBIP).

The Cube DMA is an intricate module that is configured in many places. This made it complicated to debug, as it was not evident what the fault was. According to hypso-sw issue #117, the errors we had to base this work on was that the transfers MM2S and S2MM timed out. This error message is triggered when a for-loop iterates a specified amount of times without the Cube DMA *done*-registers being set. See appendix BP.5 for listing 5 for the function that prints the errors and listing 6 for the function that checks the Cube DMA registers. Note that these functions are not from the commit the functionality was tested at in section 5.2.3, but are representative for how the functionality was at the time.

The interfacing had been proven to work on *bare metal* by earlier students. Bare metal means Without any OS or kernel. By working with bare metal, one has access to toolchains and libraries not available in hypso-sw. The module had been used in the pipeline assembled in the MasterThesis repository but no proof of a consistently working interface was given [40].

Debugging of the Cube DMA was done in multiple steps. Some also by other members of the team. PR #64 from opu-system implemented automatic building of the Cube DMA kernel module in the opu-system toolchain and loading of the module on boot. The kernel module had been updated to use generic data types and the memory ranges for reading and writing data had been moved. This was the first issue helping in on Cube DMA by a MSc student whose thesis concerned a new version of CCSDS123. This student helped out with the kernel module more throughout the following sprints, but issues and PRs are not included in this thesis.

At the time, an effort was laid down to enable ECC (Error-Correcting Code) for the entire memory on the OPU. This meant that extra bits were added to data to perform a check to see if the data is valid. A comment from the student looking into implementing ECC was written in issue #117. The student was concerned that the use of ECC could append bits to Cube DMA registers, making them contain different data than intended. the By looking at the implementation of Cube DMA in Vivado, an option to enable ECC_MODE was found. Issue #65 in opu-system was opened, hoping someone could look into this. A member of the team responded that the option did not change anything and the issue was closed.

A different approach to debugging was to look at data in the registers. Using a ZedBoard, different variations of data was written to the registers using `opu-services`. The data was then verified by checking the addresses by using `devmem` command in the ZedBoard shell. This showed that the desired registers could be set to the desired values using the services compiled from hypso-sw.

5.5 Sprint 4: Cube DMA

Getting the Cube DMA to work was the main focus of this sprint as well. Although tables 5.8 and 5.9 show no new issues or PRs regarding Cube DMA, progress was still being made and documented in issue #117.

Issues in the Sprint

Issues #117 and #169 had not been solved in the last sprint, and therefore not closed and carried over to sprint 4. Table 5.8 shows that a new issue in opu-system was included in this sprint. Issue #76 from opu-system details a bug where the toolchain in opu-system sometimes would crash. The bug had no impact on work on the thesis before sprint 6, but it was open and investigated by others from this sprint.

Repository	Issue #	Title	Points	Appendix
hypso-sw	117	Cube DMA interfacing is not working correctly	20	O
hypso-sw	169	Debug-capture saves the binned cube with 0 bytes	5	Z
opu-system	76	Errors causing building boot image (build_all script) to fail?	5	AX

Table 5.8: Issues in sprint 4

Pull Requests in the Sprint

Table 5.9 shows that no PRs were opened in this sprint that would close any active issues. The issues in table 5.8 were carried over to sprint 5.

Repository	PR #	Title	Closing issues	Appendix

Table 5.9: Pull request in sprint 4

5.5.1 Progress on the Cube DMA

During the semester's fourth sprint, a breakthrough was achieved with the Cube DMA interfacing. By trial and error, the correct sequence of setting the Cube DMA registers was found. The sequence is documented in a comment in hypso-sw issue #117. A new CSP command was created to do thorough testing of the interface. The command `hsi dmatest` would generate data, write the data to the memory region the Cube DMA reads from and then start the transfers. Then the output from the FPGA would be compared to the input. At this point, the FPGA was empty, so the in- and output should be equal. The test report *HYPSO-TRP-EL-017: Integration and Testing of Cube DMA* (see appendix B) was written to document the implementation and testing of the Cube DMA with the correct sequence of function calls to the Cube DMA module. The tests were performed on two ZedBoards: The one in the home office setup an the one in the PHiL setup. All tests were successful in both setups. The PR for issue #117 was opened early in sprint 5. Testing on the PicoBoB was not done in this sprint.

5.6 Sprint 5: Cube DMA

Sprint 4 made progress on the status of the Cube DMA, but the functionality had yet to be tested by others. As such the goal for the fifth sprint remained to make the Cube DMA work. During this sprint, the Cube DMA was tested on the PicoBoB in the LidSat (see section 3.3.1) setup. Testing on the PicoBoB opened new issues because it featured a different SoC than the ZedBoard, and the build process for the two boars are not equal.

In this sprint, the binning algorithm was worked on by a member of the team. Some issues in table 5.10 were opened requesting enhancements in the area.

Issues in the Sprint

From hypso-sw, issues #117 and #169 were carried over from the last sprint together with #76 from opu-system. The bug explained in this issue was experienced in this sprint, but not solved. Three new issues from hypso-sw were included in the sprint. Issue #194 requested that two functions for HSI capture were merged, #195 addressed a known bug from the Cube DMA, and #197 requested better naming of variables in the HSI service. All issues relevant to the sprint are shown in table 5.10.

Repository	Issue #	Title	Points	Appendix
hypso-sw	117	Cube DMA interfacing is not working correctly	20	O
hypso-sw	169	Debug-capture saves the binned cube with 0 bytes	5	Z
hypso-sw	194	Merge hsi_capture_and_binn_cube() and hsi_debug_capture_and_binn_cube()	5	AD
hypso-sw	195	HSI capture hangs after capture	3	AE
hypso-sw	197	Clarify variable names in HSI service	2	AF
opu-system	76	Errors causing building boot image (build_all script) to fail?	5	AX

Table 5.10: Issues in sprint 5

Pull Requests in the Sprint

PR #191 for hypso-sw issue #117 was opened early in the sprint requesting a review on the work done in sprint 4. The person working on binning opened PR #202, addressing multiple issues in hypso-sw. These issues are found in table 5.11.

Repository	PR #	Title	Closing issues	Appendix
hypso-sw	191	Cube DMA now works	#117 #195	AB
hypso-sw	202	Binning and other fixes	#169 #194 #197	AG

Table 5.11: Pull request in sprint 5

5.6.1 The Cube DMA Pull Request

When hypso-sw PR #191 was opened, the team was requested to review and test. All testing had been done on ZedBoards up to this point, as stated in the introduction to this sprint. When the PR was tested on a PicoZed in the LidSat setup it froze, but not during transfers, as on the ZedBoard. Some progress was made on the PR was made during the sprint by updating the system on the PicoZed in the LidSat setup, but no solution was found. This meant that the code could not be merged to master during this sprint, and had to be worked on in sprint 6 as well.

5.6.2 Involvement in Work on Binning

When working on the testing of the Cube DMA, some possible enhancements were discovered in the HSI source code. One was that there now were separate functions for regular HSI capture and the debug capture. The two functions had some duplicate code. This meant that if a bug was discovered in one of the functions, it had to be fixed for both. A better way to structure this is to merge the two functions and separate the debug part out of the function with a conditional. This is proposed in hypso-sw issue #194. Another enhancement was proposed in issue #197. The main concern of the issue was that a variable had *byte* (8 bits) in the name, while the data it represented was in *shorts* (16 bits). The issue also addressed that some variable names persisted from the migration from the MasterThesis repository (see section 5.2.1, and should be updated.

These two issues detailed above and hypso-sw issue #169 which was discussed in section 5.4.3 from sprint 3 were addressed in hypso-sw PR #202. The issues were solved and the PR was opened by the team member mentioned in section 5.4.3. A review was given on the PR, and the improvements in variable naming were highlighted as it answered issue #197 in a good way. This PR was merged to master without testing from the team, as some of the functionality fixed bugs on the main branch, and testers needed this to test properly.

5.7 Sprint 6: Cube DMA and Timestamping

This sprints original focus was only to fix the Cube DMA interface. However, a solution was found on the first day. This changed the focus of the sprint to integrate a module for the timestamping of HSI frames. All testing in this sprint was done on the LidSat setup.

Some new hardware was introduced in this sprint. A new cable between the HSI camera and the BoB in the LitSat setup was installed. This was to be able to read the *flash* signal from the camera. The timestamp module also required a PPS signal. An MCU was used to emulate this signal and installed in the LidSat setup, routing the signal to the BoB. A new repository *test-mcus* was created to have version control over this and other MCUs used for testing.

Issues in the Sprint

Issues #117 and #169 from hypso-sw and #76 from opu-system were carried over from sprint 5. The new issues in sprint 6 are found in table 5.12. From hypso-sw, issue #188 requests the timestamp module to be integrated, and #220 seeks to implement flash return from the HSI camera. The flash signal is part of the solution to #188.

Two issues were opened and closed during the sprint. These were given points on the sprint review for sprint 6. This was new for this sprint. Management wanted to address the workload of issues that were opened and closed during sprints. The first of the issues from opu-system was #89 that addressed the cause for Cube DMA freezing on PicoZed. The second issue, #103, requested that the Docker container with the hypso-sw toolchain should be run with different inputs to be less dependent on local configurations.

Repository	Issue #	Title	Points	Appendix
hypso-sw	117	Cube DMA interfacing is not working correctly	20	O
hypso-sw	169	Debug-capture saves the binned cube with 0 bytes	5	Z
hypso-sw	188	Integrate and test implementation of timestamping	13	AA
hypso-sw	220	Configure flash return from HSI camera	5	AH
opu-system	76	Errors causing building boot image (build_all script) to fail?	5	AX
opu-system	89	Update memory range for Cube DMA on PicoBoB	1	AY
opu-system	103	Use user and group IDs when running hypso-sw docker	1	BC

Table 5.12: Issues in sprint 6

Pull Requests in the Sprint

Along with some of the issues, hypso-sw PR #191 was carried over from last sprint. PR #236 from hypso-sw implements the flash return requested in issue #220. In opu-system, PR #91 was created to merge the fix of issue #89 and PR #104 addressed issue #103. PR #107 implemented the same solution for the Petalinux Docker as issue #103 requested for the hypos-sw Docker. The new repository test-mcus had its first PR in this sprint. PR #1 merged the source code for the emulated PPS signal from its feature branch to the master branch. All PRs for sprint 6 are found in table 5.13.

Repository	PR #	Title	Closing issues	Appendix
hypso-sw	191	Cube DMA now works	#117 #195	AB
hypso-sw	236	Flash is enabled	#220	AJ
opu-system	91	Updated memory regions and hard-coded addresses	#89	BA
opu-system	104	Update update-hypso-sw script	#103	BC
opu-system	107	No sudo	-	BE
test-mcus	1	PPS	-	BL

Table 5.13: Pull request in sprint 6

5.7.1 Merging the Solution for Cube DMA

The root of the Cube DMA problems was found early in this sprint. A configuration in the TCL (Tool Command Language) script for the PicoBoB configured an incorrect memory size for the Cube DMA. This resulted in erroneous hardware files being outputted from Vivado. This was addressed in opu-system issue #89, and quickly merged in PR #91. Boot files for PicoBoB build with these hardware files would be able to use Cube DMA correctly. This information was added to hypso-sw PR #191 to inform the testers that it was ready to be tested again.

The testers found the functionality to work, but addressed that the success messages were only printed in the terminal with the OPU shell. The terminal with the CLI printed a timeout message. This was because no CSP packet was returned to the CLI from the OPU. This return packet was configured to contain a string with the test results so the same information would be available on both the OPU and in the CLI.

The review then continued with a request to test camera functionality to be sure the HSI camera was still working as it should. The testers found no errors in the camera functionality but raised a question about the compressed cube size. The compressed cube contained the same amount of data regardless of how many frames were in the capture. This happened because the Cube DMA was programmed to transfer a fixed cube size regardless of how many frames were captured. The cube size was related to what CCSDS123 was programmed to receive in Vivado. The fact that CCSDS123 was not known when work on the Cube DMA started. To address this, issue #95 (see appendix BB) was opened. The issue requests the CCSDS123 compression to work for configurable cube sizes.

Once this was clarified, the working version of Cube DMA was finally merged with the master branch. This merge also solved issue #169.

5.7.2 Integration of Timestamp Functionality

By this sprint, the two BSc students had finished the implementation of the timestamp module and it was ready for integration into the HSI service. This was addressed in hypso-sw issue #188. The timestamp module requires a PPS and flash signal to work (see section 3.2.10). These signals did not exist in the LidSat setup at the start of the sprint and had to be introduced. Issue #188 outlines two possibilities for introducing the signals.

5.7.3 Enabling the Flash Signal

To have a setup that was as realistic as possible, the flash signal was to be acquired from the HSI camera in the LidSat. This required a new cable for the signal and using the iDS API to configure it. Issue #220 was opened to address the configuration of the flash signal and documentation of design choices according to iDS' documentation [24]. Listing 7 in appendix BP.6 shows the code that was used to configure flash return. This code was included in the `hsi_camera_configure()` function in `src/hsi/hsi_camera.c`.

Access was granted to the lab to install the cable and test the signal. Figure 5.8 shows the flash signal using an oscilloscope. The signal was probed on the connector on the BoB. The figure is the same as in issue #220. PR #236 was opened to merge the feature to the master branch. The feature was confirmed to not break the HSI capture, the code was reviewed, and merged to master.

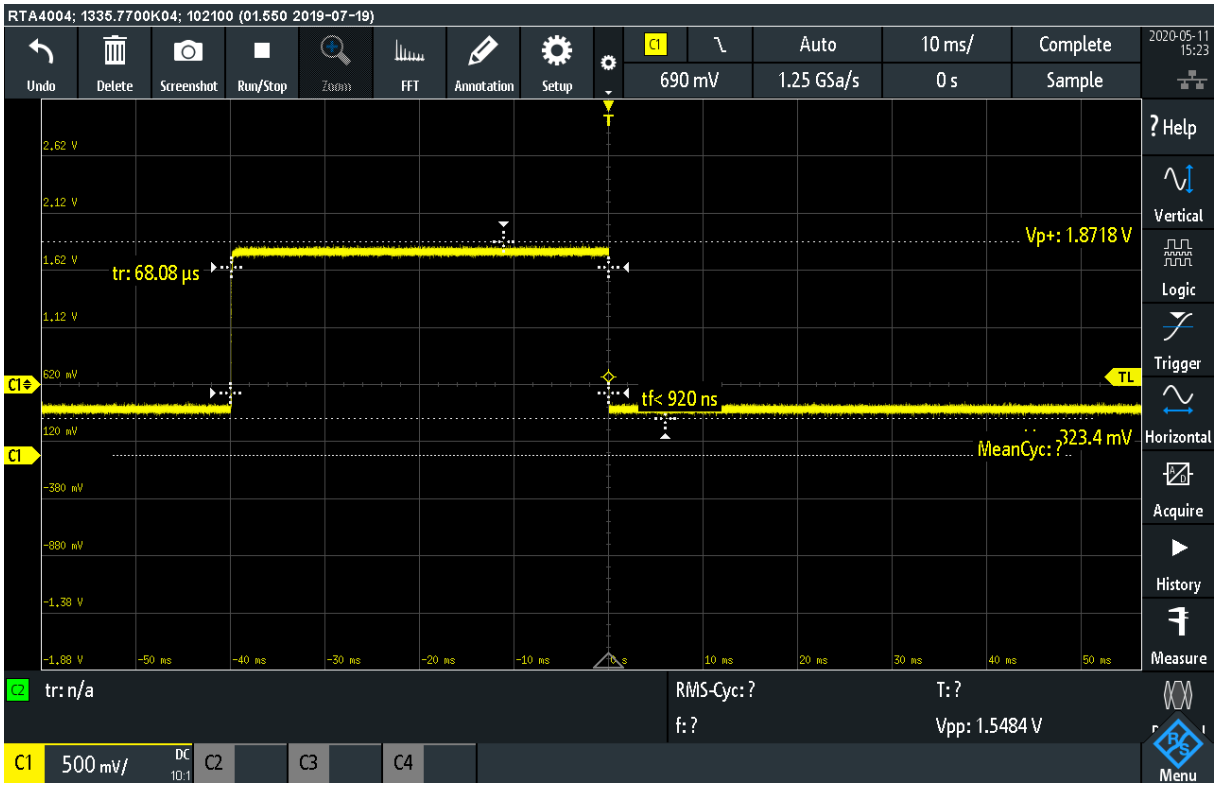


Figure 5.8: Oscilloscope reading of flash signal from HSI camera.

5.7.4 Emulating the Pulse-Per-Second Signal

The PPS signal had to be emulated, as it originates from a GPS module on the FC which is not in any setup in the SmallSatLab on NTNU. A Teensy MCU was chosen to emulate the signal, as someone in the HYPSO team had one to spare. The MCU could be used with the Arduino IDE with slight modifications. This made the MCU easy to use. A new repository was created to have version control of the software on the MCU and a place to document the setup. The new repository was called *test-mcus*. Another MCU already in the LidSat setup was also added to this repository.

PR #1 from test-mcus merged the source code and documentation for the Teensy, or GSE-PPS generator as it is also known. The prefix GSE (Ground Support Equipment) shows that the equipment is not used in space. The code for the GSE-PPS signal is shown in its entirety in listing 8 in appendix BP.7.

Access was granted to the lab to install the MCU in the LidSat setup. Figure 3.13 in section 3.3.1 shows the GSE-PPS generator in the middle. It is green and labeled PPS. The signal was probed on the input port on the BoB using an oscilloscope. Figure 5.9 shows one pulse of the signal. Listing 8 shows that the signal was configured to be 50ns, while the oscilloscope measured the duration to be 46ns. The whole period of the signal is shown in figure 5.10. While it can be hard to see the pulses in the figure, the oscilloscope gives the period to be 1.00197s. This is about 2ms longer than what was configured. These concerns were raised in test-mcus issue #5 (appendix BO). The issue was not worked on during any of the sprints.

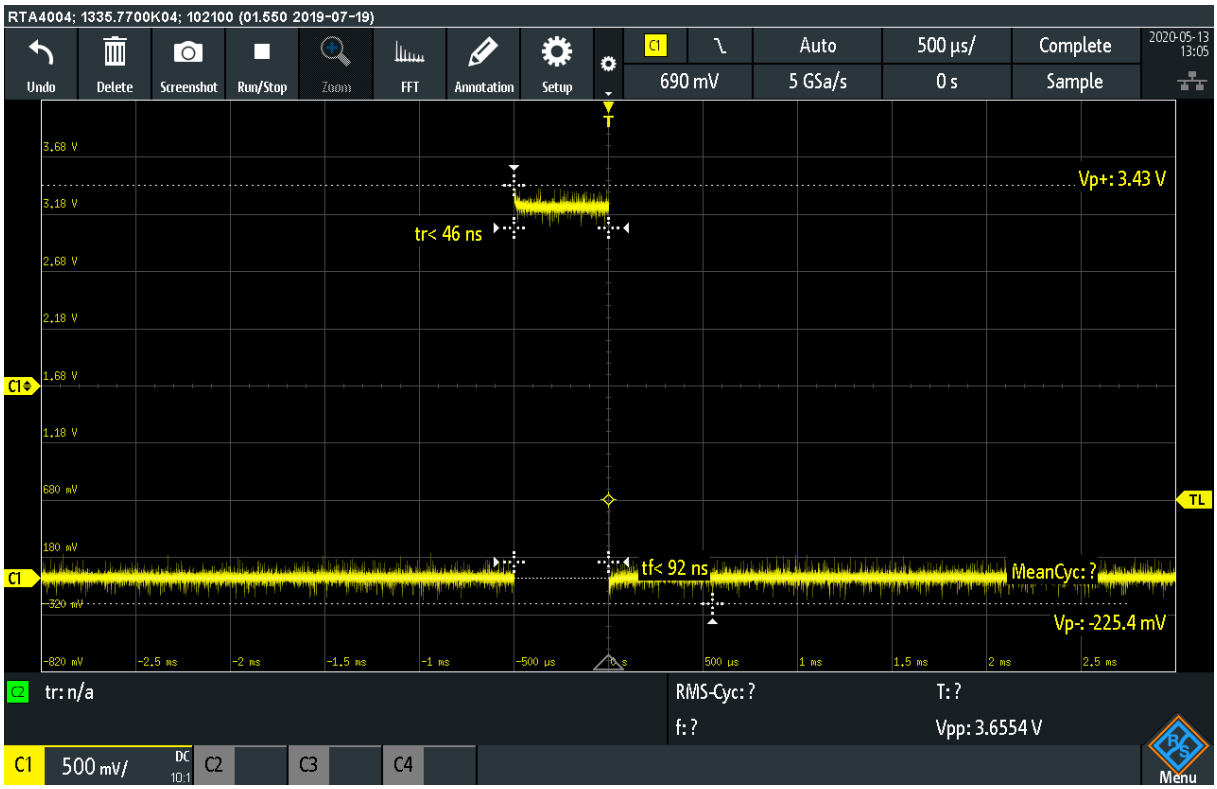


Figure 5.9: Oscilloscope reading of the emulated PPS signal. One pulse is shown.

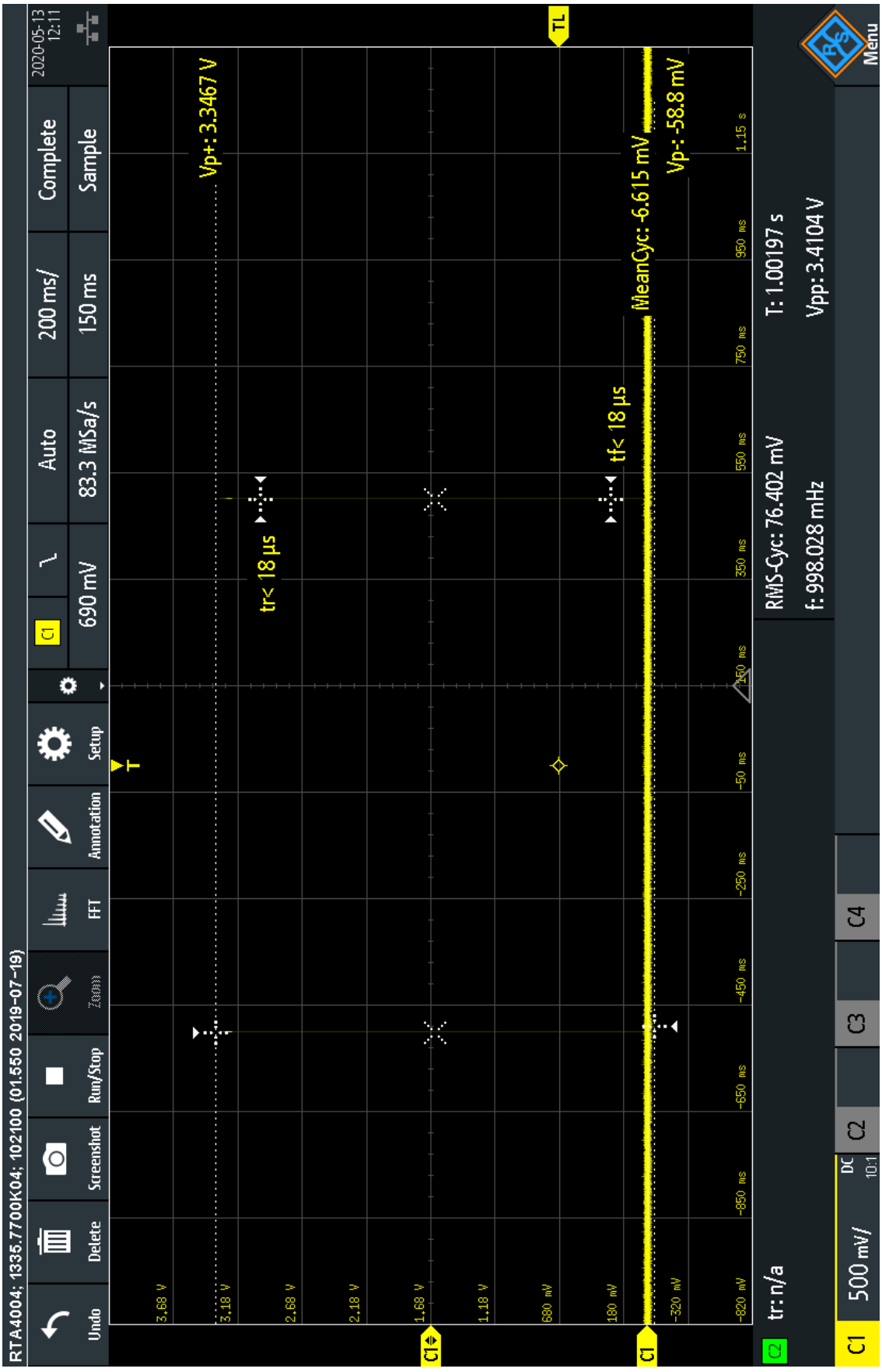


Figure 5.10: Oscilloscope reading of the emulated PPS signal. One period is shown.

5.7.5 Solving a Bug in the opu-system Toolchain

The bug described in opu-system issue #76 had been present for some sprints, but the cause had not been found. It disappeared and reappeared with no apparent pattern. While debugging this issue on Slack, a MSc student who worked on the boot procedure for the OPU discovered a thread from a forum with useful information. It was reported a step in the Petalinux toolchain would fail if it was executed in a window that was too small. This was documented in the issue, and the issue was closed. A new issue in opu-system, #108, was opened to address how to solve this. The issue was a part of both sprints 7 (table 5.14) and 8 (table 5.16).

5.7.6 Updating the Docker Commands in opu-system

A problem that had persisted since sprint 3 (opu-system issue #62, table 5.6) was concerning permissions of files to be used with Docker in the toolchain for building boot files in opu-system. If a file has wrong permissions, Docker cannot access them. If Docker is given *sudo* (also known as *root* or *superuser*) permissions files would not be accessible without root privileges. In this sprint, opu-system issue #103 was opened to suggest running the commands that start the Docker container with the hypso-sw submodule with arguments that ensure no problems with permission would arise in the submodule. PR #104 was opened soon after to request these changes to be merged with master. Some concerns about the impact on the toolchain were raised in the PR. These concerns were addressed, and the fix was merged. A similar solution for the Petalinux Docker implemented by another member of the was merged to master with opu-sysem PR #107. It was confirmed in the review that the toolchain could be used without root permissions. All files were usable by non-root users. This made the opu-system toolchain easier to use for all.

5.8 Sprint 7: Timestamping and Thesis

The focus in sprint 7 was to finish the work on the integration of the timestamp module. When this was done, it was time to start to focus on writing this thesis. Apart from opening issues and doing some reviews, little work was done on the software in this sprint after the timestamping was proven to work.

In this sprint, the MOBIP test was done again. The test succeeded, confirming that the modules of the MOBIP are integrated.

Issues in the Sprint

In table 5.14, hypso-sw issue #188 was carried over from sprint 6. Issue #235 addresses a bug regarding the HSI camera initialization that came up when testing the flash in sprint 6. An issue about the timestamp kernel module was opened in #251 and closed quickly after. From opu-system, issue #108 was opened in the previous sprint addressing the terminal window size. In test-mcus, issue #4 requested a new cable between the GSE-PPS.

Repository	Issue #	Title	Points	Appendix
hypso-sw	188	Integrate and test implementation of timestamping	13	AA
hypso-sw	235	HSI (debug)capture fails to init camera 50% of the times it's run	8	AI
hypso-sw	251	Want error handling for "Failed opening timestamp device."	3	AM
opu-system	108	Fix issue with Petalinux config crash if terminal window is too small	3	BF
test-mcus	4	Make a better cable setup for the PPS signal	-	BN

Table 5.14: Issues in sprint 7

Pull Requests in the Sprint

Table 5.15 shows the PRs in the sprint. PR #240 from hypso-sw integrates timestamping in the HSI source code. Issue #235 was addressed in hypso-sw PR #253. In opu-system, the source code for the timestamp modules kernel module and bitstream was merged to master.

Repository	PR #	Title	Closing issues	Appendix
hypso-sw	240	Timestamping of HSI frames is live!	#188	AK
hypso-sw	253	Multiple cam init attempts	#235	AN
opu-system	122	Integrate timestamp module	-	BH

Table 5.15: Pull request in sprint 7

5.8.1 Integration of Timestamping

With the work on the flash and GSE-PPS signals done in sprint 6 (section 5.7.2), the source code developed by the BSc students was ready to be tested. In hypso-sw, PR #240 was opened to merge the timestamp functionality to master. This PR required opu-system to be built with the files in opu-system PR #122. This was because PR #122 featured the FPGA timer and kernel module required for timestamping. The initial comment in PR #240 addressed this, as well as other prerequisites for testing.

In the review of PR #240, the structure of the outputted file with timestamps was discussed and agreed upon. The correctness of timestamping was also addressed in the review since no plan for validation had been made. Issue #242 (appendix AL) in hypso-sw was opened for continuing this discussion so that the PR could be merged. The outputted file from this PR had two columns for timestamps. One for the rising and one for the falling flank of the flash signal from the HSI camera. Since the flash signal only could be set to a static value, it made sense to only have one column per timestamp. This was changed at a later point in the semester to look like the example in listing 3.1 in section 3.2.10.

5.8.2 Minimal On-Board Processing Pipeline - Revisited

The first operational test of the MOBIP done after the first sprint was *completed to the minimal success criteria* [16]. The modules required for the MOBIP were present, but not performing in the way they should. This test was done again on the 27th of May, the day before the sprint ended. The test report *HYPSO-TRP-EL-024: MOBIP COMPLETE LIDSAT* concluded that *MOBIP works when using the FPGA for compression* and summarizes [6]:

- The number of frames were controllable from *hypso-cli*.
- The specified number of image frames for the cube were acquired.
- The image cube was compressed via the FPGA, reducing its size for the image cube size that the FPGA compression was configured for.
- The images were transferred to the operator station.
- The compressed image cube was decompressed.
- The original image cube was identical with the decompressed image cube, when compressed via FPGA. Compared byte for byte by the linux util cmp.

While the SW-compression did not work on the test, the modules all were present and testable. This indicates that the modules interface correctly with each other. A faulty module can be fixed within the module and keeping the interface intact.

5.9 Sprint 8: Thesis

No work on the SW was done in the final sprint of the semester. Some follow up was done by creating issues, but the main focus during this sprint was to write this thesis.

Issues in the Sprint

The issues for this sprint in are included in table 5.16. Issue #260 in hypso-sw addresses that the file endings for HSI cubes were not concise. In opu-services, issue #108 was carried over from the last sprint. A new cable to power the Teensy in the LidSat setup was requested in test-mcus issue #3. The MCU was originally powered over USB from a computer, but to have the same ground reference as the rest of the modules in the LidSat it should receive power from the EPS instead.

Repository	Issue #	Title	Points	Appendix
hypso-sw	260	Change .raw file format to .bip file format	2	AO
opu-system	108	Fix issue with Petalinux config crash if terminal window is too small	3	BF
test-mcus	3	Make the GSE-EPS power the Teensy GSE-PPS generator instead of power over USB	5	BM

Table 5.16: Issues in sprint 8

Pull Requests in the Sprint

In hypso-sw, PR #279 addressed the file ending change requested in issue #260 among other things. PR #120 in opu-system found a way to solve that the toolchain crashed, and implemented this.

Repository	PR #	Title	Closing issues	Appendix
hypso-sw	279	All HSI file names are static and stored in a folder	#260	AP
opu-system	120	Added command to scale terminal inside window & added –silentconfig	#108	BG

Table 5.17: Pull request in sprint 8

Chapter 6

Conclusion

6.1 Reflections

6.1.1 Workflow

A semester of eight sprints has come to an end. The software team has worked together on designing, developing, testing, and reviewing software that is going to run on the OPU for the payload on the HYPSO satellite. A new workflow was introduced at the beginning of the semester. The GitHub workflow (section 4.1) described how GitHub was to be used when developing the software. Scrum provided a methodology for cooperation and structuring our work. It took some time to get used to, but after a few sprints, the workflow was adopted by all team members. We understood what powerful tools issues and PRs were, and started writing more and better issues and PRs. This has made it easier to talk about software, document software, ask questions, or request new features. By looking at what issues are in the kanban board, an overview of the current status of the project can be obtained. These tools have been essential to do system integration as they provided ways to communicate and document effectively. Especially when the team has mostly worked from home this semester.

6.1.2 Testing

Testing became a natural part of the work when we were forced to review each other's work before a feature or fix could be merged to master. With a bigger merge came more extensive testing. Some reviews were approved only based on code review, while other reviews took days and the participation of several testers to be approved. This was a norm the team developed over sprints, not something that was specified by a workflow. The GitHub workflow ensured that bugs were found and addressed along the way instead of staying dormant. This made the team aware of the status of the software. Both on the master branch and feature branches.

6.1.3 Minimal On-Board Imaging Pipeline

In addition to the continuous testing by the team, two larger, operational tests were performed this semester. These tested the modules of the minimal processing pipeline on the OPU. The MOBIP (Minimal On-Board Processing Pipeline) contains several modules that were integrated as a part of this thesis. The CSP commands that interface between the operator and the payload, the communication between the OPU and the HSI camera, and the interface between the processor and the FPGA on the OPU have all been worked on during this semester. The second operational test of the MOBIP (section 5.8.2) concluded that the pipeline works when using compression in the FPGA. The compression in software did not work on the second test, but the fact that we know it did not work means that the interface exists and is working.

The fact that the MOBIP is working means that the payload on the HYPSO satellite works. This is an important milestone for the project, as the software now is ready for launch. The implementation is still minimal, and a considerable amount of testing remains.

6.1.4 Timestamping

A timestamping module has been integrated and proven to work on the test setup in the lab. This module was developed by two BSc students and integrated and tested as part of this thesis. The addition of timestamping to the HSI service provides data that makes it possible to tell what has been imaged on Earth's surface.

6.1.5 Code Quality

Code quality and ease of use have been focused on during the entire semester. An entire two-week sprint was dedicated to improving the HSI source code. The codebase was made more modular by dividing and grouping similar functionality. The interface between the modules was formalized and given an appropriate level of abstraction. Several issues have been opened in the different repositories on GitHub requesting improvement in code quality or ease of use. The toolchain in opu-system has been re-worked so that permission for files and directories is no longer an issue.

6.2 Future work

6.2.1 Baseline On-Board Imaging Pipeline

The minimal processing pipeline has been proven to work. After thorough testing, the HYPSO team will begin to implement the baseline processing pipeline. This pipeline will feature more modules in the FPGA such as target detection and correction of atmospheric aberrations. The examples mentioned have been worked on by MSc students this semester, but need to be tested on the OPU together with the existing modules.

6.2.2 Issues for Future Sprints

Table 6.1 shows issues that have been opened during this semester that have not been closed. Most of them seek to further enhance the usability of the toolchain in opu-system, others are related to testing of the payload. Issue #95 raises a concern that must be addressed before launch. The CCSDS123 compression in FPGA is programmed in Vivado to compress cubes with a fixed size. This was addressed in the Cube DMA PR (hypso-sw #191, section 5.7.1). The compression must either be able to be reconfigured by the OPU or a strategy to only compress cubes of fixed sizes must be made.

Repository	Issue #	Title	Points	Appendix
hypso-sw	193	Log processing rate of HSI frames	5	AC
hypso-sw	242	How do we test correctness of timestamping?	8	AL
opu-system	90	Enhance .tcl scripts	5	AZ
opu-system	95	CCSDS123 version 1 Compresses Fixed Cube Sizes	21	BB
opu-system	123	Build modules based on board	3	BI
opu-system	124	Sort out Cube DMA address dependencies	3	BJ
opu-system	125	Move kernel module files for Cube DMA to its own directory	1	BK
test-mcus	5	Improve accuracy of GSE-PPS mock signal	5	BO

Table 6.1: Issues for the future

6.2.3 Code Quality

Code quality is a subject that must not be neglected. With the high turnover that HYPSO has (some stay on-board for only a semester), it is important that students do not get lost in the source code. A considerable amount of time has been used on making sense of a codebase with poor code quality. Both in the specialization project [11] and during work on this thesis. The code written during work on this thesis is by no means perfect, but will hopefully be understandable and not scare new students away.

6.2.4 Launch

It will be exciting to follow the launch in the first quarter of 2021. Hopefully, HYPSO will find its orbit with everything intact. Then, everyone's hard work that contributed to the project will pay off when we perform advanced oceanographic observation with COTS components!

Bibliography

[1] A. W. Appel. Verified software toolchain. In G. Barthe, editor, *Programming Languages and Systems*, pages 1–17, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[2] Arduino. Ardiono website. https://www.arduino.cc, Accessed 07.06.20.

[3] Avnet. *PicoZed™ 7Z015/7Z030 SOM Hardware User Guide*. http://zedboard.org/sites/default/files/documentations/5279-UG-PicoZed-7015-7030-V2_0.pdf, 2 2018. Accessed 31.10.19.

[4] Avnet. *Zynq-7000 SoC Product Selection Guide*. https://www.xilinx.com/support/documentation/selection-guides/zynq-7000-product-selection-guide.pdf, Accessed 07.06.20, 2019.

[5] S. Bakken, J. Garrett, M. Orlandic, and M. Hov. *HYPSO-UM-002: Workflow for Software Development*. Internal document. Revision 1, Non-published.

[6] S. Bakken and D. Langer. *HYPSO-TRP-EL-024: MOBIP COMPLETE LIDSAT*. Internal document. Non-published., 2020.

[7] R. Birkeland and D. Langer. *HYPSO-TPL-005: MOBIP Test Plan*. Internal document. Non-published.

[8] R. Birkeland and D. Langer. *HYPSO-UM-004: Manual for FlatSat and LidSat*. Internal document. Revision 1, Non-published.

[9] S. Chacon and B. Straub. *Pro Git*. Apress, 2 edition, 11 2014.

[10] T. Curtin, J. Bellingham, J. Catipovic, and D. Webb. Autonomous oceanographic sampling network, 01 1993.

[11] M. Danielsen. *System Integration and Testing of On-Board Processing System for a Hyperspectral Imaging Payload in a CubeSat*, 2019. Non-published.

[12] Docker. Docker website. https://www.docker.com/, Accessed 06.06.20.

[13] J. Fjeldtvedt. *Efficient Streaming of Hyperspectral Images*. Master's thesis, NTNU, Norway, 2018.

[14] J. Fjeldtvedt, M. Orlandić, and T. A. Johansen. An efficient real-time fpga implementation of the ccsds-123 compression standard for hyperspectral images. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 11(10):3841–3852, 2018.

[15] H. Galtung, T. H. Kaasa, T. Tran, and E. Prentice. *HYPSO-RP-006: HSIv6 Assembly, Fred's Design*. Internal document. Revision 1, Non-published., 2019.

[16] J. Garret, R. Birkeland, and D. Langer. *HYPSO-TRP-EL-003: MOBIP Test Report*. Internal document. Non-published.

[17] GomSpace. *CubeSat Space Protocol (CSP)*. https://bytebucket.org/bbruner0/albertasat-on-board-computer/wiki/1.%20Resources/1.1.%20DataSheets/CSP/GS-CSP-1.1.pdf?rev=316ebd49bed49fdbb1d74efdeab74430e7cc726a, 6 2011. Accessed 06.06.20.

[18] M. E. Grøtte, J. Fortuna, R. Birkeland, J. Veisdal, M. Orlandic, J. Martens, J. T. Gravdal, F. Sigernes, T. A. Johansen, J. O. Reberg, J. G., and K. Rajan. *Hyper-spectral Imaging Small Satellite in Multi-agent Marine Observation System*. Internal document. Vol. 1, no. Non-published, 2017.

[19] M. Hov. *Design and Implementation of Hardware and Software Interfaces for a Hyperspectral Payload in a Small Satellite*. Master's thesis, NTNU, Norway, 2019.

[20] HYPSO-SW team. *hypso-sw*. Github repository: https://github.com/NTNU-SmallSat-Lab/hypso-sw.

[21] HYPSO-SW team. *opu-system*. Github repository: https://github.com/NTNU-SmallSat-Lab/opu-system.

[22] HYPSO-SW team. *test-mcus*. Github repository: https://github.com/NTNU-SmallSat-Lab/test-mcus.

[23] HYPSO team. Hypso webpage. https://hypso.space, 2019. Accessed 30.10.19.

[24] iDS. *uEYE Camera Manual*. IDS Imaging Development Systems GmbH, 2019. Version 4.92.2.

[25] IEEE Computer Society. *IEEE Standard for Ethernet*. https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8457469, 6 2018. Accessed 05.12.19.

[26] IOCCG. *Phytoplankton Functional Types from Space*, volume No. 15 of *Reports of the International Ocean Colour Coordinating Group*. International Ocean-Colour Coordinating Group, Dartmouth, Canada, 2014.

[27] L. Khong, L. Yu Beng, T. Yip, and T. Fun. Software development life cycle agile vs traditional approaches. 02 2012.

[28] J. A. Kornberg and S. Netteland. *Time Synchronization of Hyperspectral Image Capture on board a Nanosatellite*, 2020. Bachelor's Thesis. NTNU, Norway.

[29] S. McConnel. *Code Complete*. Microsoft Press, 2 edition, 2004.

[30] Microsoft. *Microsoft completes GitHub acquisition*. https://blogs.microsoft.com/blog/2018/10/26/microsoft-completes-github-acquisition/, Accessed 24.05.20, 10 2018.

[31] NanoAvionics. *6U nanosatellite bus M6P*. https://nanoavionics.com/nanosatellite-buses/6u-nanosatellite-bus-m6p/, Accessed 19.06.20.

[32] L.-J. Naustvoll. *What we know about the so-called "killer alga" in northern Norway*. https://www.imr.no/en/hi/news/2019/may/what-we-know-about-the-so-called-killer-alga-in-northern-norway, 5 2019. Accessed 30.10.19.

[33] S. L. Ramírez-Mora, H. Oktaba, and H. Gómez-Adorno. Descriptions of issues and comments for predicting issue success in software projects. *Journal of Systems and Software*, 168:110663, 2020.

[34] L. Rising and N. S. Janoff. The scrum software development process for small teams. *Software, IEEE*, 17:26 – 32, 08 2000.

[35] A. S. Tanenbaum and D. J. Wetherall. *Computer Networks*. Pearson, Permissions Department, 501 Boylston Street, Suite 900, Boston, Massachusetts 02116, 5 edition, 9 2010.

[36] Texas Instruments. *Introduction to the Controller Area Network (CAN)*. http://www.ti.com/lit/an/sloa101b/sloa101b.pdf, 5 2016. Accessed 04.12.19.

[37] The CubeSat Program. *6U CubeSat Design Specification*. https://static1.squarespace.com/static/5418c831e4b0fa4ecac1bacd/t/5b75dfcd70a6adbee5908fd9/1534451664215/6U_CDS_2018-06-07_rev_1.0.pdf, 7 2018. Revision 1, Accessed 04.06.20.

[38] A. Vaduva. *Learning Embedded Linux Using the Yocto Project*. Community Experience Distilled. Packt Publishing, 2015.

[39] A. Varntresk. *MasterThesis*. Github repository: https://github.com/NTNU-SmallSat-Lab/MasterThesis.

[40] A. Varntresk. *Assembly and Testing of Baseline Processing Chain*. Master's thesis, NTNU, Norway, 2019.

[41] Z. Volent, G. Johnsen, and F. Sigernes. Kelp forest mapping by use of airborne hyperspectral imager. *Journal of Applied Remote Sensing*, 1(1):1 – 21, 2007.

[42] D. Wallechinsky and I. Wallace. *The People's almanac*. Garden City, N.Y. : Doubleday, 1975.

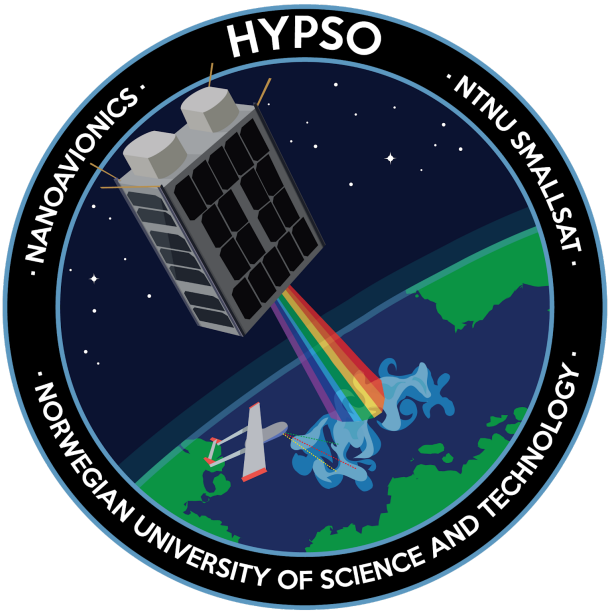
[43] Xilinx. *PetaLinux Tools Documentation, Reference Guide*. Xilinx Inc, 5 2019. UG1144 (v2019.1).

Appendices

A Test Report TRP-EL-014

Remote Uploading of New Boot Files to ZedBoard

HYPSO-TRP-EL-014



Prepared by: HYPSO Project Team
Reference: HYPSO-TRP-EL-014
Revision: 1
Date of issue: Date
Status: Preliminary
Document Type: TBD

Table Of Contents

Table 1: Table of Changes

Rev.	Summary of Changes	Author(s)	Effective Date
1	*First issue*	*Magnus Danielsen*	

1 Overview

The HYPSO Mission will primarily be a science-oriented technology demonstrator. It will enable low-cost & high-performance hyperspectral imaging and autonomous onboard processing that fulfill science requirements in ocean color remote sensing and oceanography. NTNU SmallSat is prospected to be the first SmallSat developed at NTNU with launch planned for Q4 2020 followed by a second mission later. Furthermore, vision of a constellation of remote-sensing focused SmallSat will constitute a space-asset platform added to the multi-agent architecture of UAVs, USVs, AUVs and buoys that have similar ocean characterization objectives.

1.1 Purpose

The purpose of this document is to keep a log of the procedure of remotely updating the boot files for embedded Linux on the ZedBoard in the SmallSat Lab. The boot files on the ZedBoard in the lab are of an unknown version, and we need it to be of the newest possible. This is to ensure that we work on systems of the same version.

It is relevant to do this remotely because:

1. During operation, we will need to upload new boot files to the OPU on the satellite.
2. This document is written during the 2020 Covid-19 lockdown of campus. It is not easy to get access to the lab to do this manually, as have been the procedure until now.
3. It will be a thorough test of the file transfer service, and the files generated by opu-services.

This procedure has two criteria for full success, but number one is sufficient:
The ZedBoard is able to boot with the new boot files
The ZedBoard boots with a valid, persistent IP address

1.2 Scope

This document describes the preparation, upload and testing of boot files for a ZedBoad. The results are discussed and the successfulness of the procedure is presented.

1.3 Summary

The boot files were generated automatically from scripts in a github repository. They were tested on a ZedBoard before being transferred to a computer connected to another ZedBoard with a CAN-bus. The boot files were transferred over the CAN-bus using software developed by the HYPSO-team using the CSP protocol. Once the files were transferred, the ZedBoard was

rebooted to load embedded Linux from these files. Due to a missing file, the ZedBoard did not boot correctly.

1.4 Applicable Documents

The following table lists the applicable documents for this document and work.

Table 2: Applicable Documents

ID	Author	Title
[AD01]		
[AD02]		
[AD03]		
[AD04]		
[AD05]		

1.5 Referenced Documents

The documents listed in have been used as reference in creation of this document.

Table 3: Referenced Documents

ID	Author	Title
[RD01]	HYPSO SW Team	Opu-system (GitHub repository)
[RD02]	Roger Birkeland	HYPSO-TRP-EL-006
[RD03]		
[RD04]		
[RD05]		

2 Tools and Equipment

The tools needed to perform this operation will be divided into two groups: Home office and SmallSatLab.

2.1 Home Office:

The set up in the home office consists of:

- Computer running Ubuntu 18.04 LTS with relevant software and connected hardware:
  - Cisco VPN (SW)
  - Git (SW)
  - Docker (SW)
  - SSH and CSP (SW)
  - Picocom (SW)
  - USB-A to Micro-USB cable to ZedBoard
  - Ethernet connection to internet
- ZedBoard connected via USB to computer for UART communication
- 8 Gb SD-card with two partitions:
  - 1.5 Gb, FAT32 partition called BOOT
  - 6.5 Gb, EXT4 partition called SD

2.2 SmallSatLab:

- Computer running Ubuntu 18.04 LTS with relevant software and connected hardware:
  - SSH server (SW)
  - Picocom (SW)
  - SysWorxx CAN drivers (SW)
  - Hypso-cli (SW)
  - USB to CAN adapter
  - USB-A to Micro-USB cable to ZedBoard for UART
  - Ethernet connection to internet
- ZedBoard with
  - CAN transceiver (or controller?)
  - UART connection to computer over USB
  - Ethernet connection to internet
  - Functioning boot image with
    - opu-services
    - Known IP address
    - Mounted BOOT partition of SD card

3 Generating Boot Files

Following the readme of [RD01], the GitHub repository is cloned using git. Then the readme is followed to install necessary software, and system setup. When this is done, from the top directory in the repository, the top-level makefile `build_all` is executed with device id `11` and build flag `-zed`: `./build_all 11 -zed`. The script will then run all necessary commands to create the bootable files for embedded Linux to replace those on the ZedBoard in the SmallSatLab. The readme of the repository is then updated with build date, commit ID, and who built it.

This was done when the master branch was on commit 9d73cb3, on March 22nd 2020.

3.1 Testing of Boot Files

The process described in section 2 generates four files: `bitstream.bit`, `BOOT.BIN`, `image.ub` and `swapfile`. These four files were copied to the BOOT partition of the SD-card, which was inserted into the ZedBoard. After turning the ZedBoard on, picocom was used to get UART communication with the board.

Due to some noise, the automatic booting was interrupted, and a boot command had to be given manually:
`Zynq > boot`
The system then booted as expected.
When using the `ip addr` command, it showed that the board had booted with the correct IP address:

```
3: eth0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc pfifo_fast state
DOWN group default qlen 1000
    link/ether 00:0a:35:00:1e:53 brd ff:ff:ff:ff:ff:ff
    inet 129.241.2.31/23 scope global eth0
       valid_lft forever preferred_lft forever
```

The boot files work locally, so they will be used to update the ZedBoard in the lab.

4 Uploading Boot Files

The boot files needs to be uploaded in two steps. One from the home office to the HIL computer in the lab using SCP, and one step from the HIL computer to the ZedBoard using CSP.

The `swapfile` is not to be uploaded, as there is no room for it on the boot-partition on the ZedBoard. The file is not needed by the system to run.

4.1 Step 1: Home to Lab

In this step, the files were moved using SCP (Secure Copy) over the internet. It is important that the target computer has some kind of ssh server activated for this. The procedure is a s follows:
`scp path/to/local/file user@target-ip:path/to/target/file`

This was done three times, one for each of the files. The commands were executed from the zed-directory in opu-system:
`scp BOOT.BIN hypso@smallsatlab-hil.ies.ntnu.no:Magnus`
`scp image.ub hypso@smallsatlab-hil.ies.ntnu.no:Magnus`
`scp bitstream.bit hypso@smallsatlab-hil.ies.ntnu.no:Magnus`

The transfer was verified by logging into the target computer with ssh and listing the files in that directory:
`home: ssh hypso @smallsatlab-hil.ies.ntnu.no`
`hil: cd Magnus`
`hil: ls`
Output: `bitstream.bit BOOT.BIN hypso-cli-M image.ub opu-services-M`

4.1.1 Setting Up the ZedBoard

The first step now is to make sure the boot partition of the SD card in the ZedBoard is mounted. Using SSH to log in to the Zedboard: `ssh root@smallsatlab-zedboard01.ies.ntnu.no`. Navigating to `/mnt`, where it is usual to mount drives. No directories showed up when using `ls`, so a directory `boot` was made. Using the output from `fdisk -l` to find the correct partition:

```
Device       Boot StartCHS    EndCHS        StartLBA     EndLBA    Sectors  Size
 Id Type
/dev/mmcblk0p1    0,32,33     12,93,17          2048     198655     196608 96.0M
  c Win95 FAT32 (LBA)
/dev/mmcblk0p2    12,125,50   1023,254,63     200704   30924799   30724096 14.6G
 83 Linux
```

Figure 1: Listing partitions on the ZedBoard

As the partition `/dev/mmcblk0p1` is formatted as FAT32, this is the partition we want to mount. This is done with `mount /dev/mmcblk0p1 /mnt/boot`, which is verified by listing the contents of `/mnt/boot`: `BOOT.BIN  image.ub`.

The files `BOOT.BIN` and `image.ub_OLD` were copied to `BOOT.BIN_OLD` and `image.ub_OLD`, and copied over to the home office for redundancy.

By starting opu-services, the ZedBoard will be ready to be updated:
`home/root/opu-services-M 12 can0`.

4.2 Step 2: From the HIL Computer to the ZedBoard

4.2.1 Transfer Procedure

After the ZedBoard is setup, the cli can be started on the HIL computer. First, log on to the computer: `ssh hypso@smallsatlab-hil.ies.ntnu.no`. Then start the cli: `sudo /home/hypso/Magnus/hypso-cli-M 16 -c can0`. Using the cli, there are several steps to upload a file [RD02]:

List the formatted files on the ZedBoard: `(hypso) ft list 12`
If the list is not empty, entries can be removed with `(hypso) ft deregister 12 <ID>`
Register a target file in the `/mnt/boot` directory:`(hypso) ft register 12 /mnt/boot/<target>.fmt <new ID>`
Verify that it was registered with `(hypso) ft list 12`
Prepare a local file for upload by formatting it: `(hypso) ft prepare local <path to file> <path to formatted file>.fmt <same ID as in step 2> <entry size>`
For the entry size, use 240, as that is the number of entries that fits in a CSP packet.
Pay attention to the number of total entries printed by executing this command, as it is needed by the next step.
Make the target file on the ZedBoard match the formatted file: `(hypso) ft format 12 <same ID as in step 2> <entry size> <total number of entries>`
Using the list commend, the formatting can be verified.
Upload the file: `(hypso) ft upload file 12 <path to formatted file>.fmt 1`
Any number of packets may fail to be transmitted. Re-run the command until all are transferred.

a. Using the command: `(hypso) ft check ALL 12 <the same ID as in step 2> <first entry of ID range> <last entry of ID range>`, the transfer status of a range of packets is printed.

Extract the formatted file: `(hypso) ft extract 12 <the same ID as in step 2> /mnt/boot/target`
Deregister the file: `(hypso) ft deregister 12 <the same ID as in step 2>`

When all the steps above succeed for all three files, the ZedBoard must be rebooted to apply the changes.

4.2.2 Test Procedure

The file to be uploaded was a previously captured image from the HSI during a test. It will be called `test_original.png`. After formatting it will be called `test_original_formatted.png.fmt`. The formatted file uploaded to the ZedBoard will be called `test_uploaded_formatted.png.fmt`, and the extracted file will be called `test_uploaded.png`. There will exist a file `test_uploaded.png` with arbitrary data in the ZedBoard before the procedure is started to see if it is overwritten, as the boot files must be overwritten for this to work. Both `test_original.png` and `test_uploaded.png` will be transferred to the home office for comparison. If they are the same, the procedure works, and can be attempted on the boot files.

Performing the steps in the previous section, with outputs:

```
(hypso) ft list 12
N/A
(hypso) ft register 12 /home/root/test_uploaded_formatted.png.fmt 1
N/A
```

b. `(hypso) ft list 12`

File name	ID	Last Entry	Total Entries	Cell Size	Cells Used	Max Cells	Sector Quantity	Sector Size	File Type
rmated.png.fmt	1	0	0	0	0	0	0	0	NOFILE

Figure 2: List of files registered in file transfer service

```
(hypso) ft prepare local test_original.png test_original_formatted.png.fmt 1
240
file_name:      test_original_formatted.png.fmt
      file_type:      STATIC
      file_id:        1
      entry_sz:       240
      max_entries:    4783
      first_entry_id: 1
      total_entries:  4783
ft format 12 1 240 4783
File ID: 1, status: 0
```

c. `(hypso) ft list 12`

File name	ID	Last Entry	Total Entries	Cell Size	Cells Used	Max Cells	Sector Quantity	Sector Size	File Type
rmatted.png.fmt	1	4783	4783	246	4783	4783	1	1176-650	STATIC

Figure 3: The file has been formatted correctly

```
(hypso) ft upload file 12 test_original_formatted.png.fmt 1
#Very many of these:
Uploading 1 missing ranges:
      [    4407-4783    ]: 100% [==========================]
      Uploading 1 missing ranges:
      [    4461-4783    ]: 100% [==========================]
      Uploading 8 missing ranges:
      [    4515-4517    ]: 100% [==========================]
      [    4519-4527    ]: 100% [==========================]
      [    4529-4539    ]: 100% [==========================]
      [    4541-4553    ]: 100% [==========================]
      [    4555-4570    ]: 100% [==========================]
      [    4572-4602    ]: 100% [==========================]
      [    4604-4605    ]: 100% [==========================]
      [    4607-4783    ]: 100% [==========================]
      #Before, finally:
      File is complete.
      Upload finished.
(hypso) ft extract 12 1 /home/root/test_uploaded.png
N/A
(hypso) ft deregister 12 1
```

Both `test_original.png` and `test_uploaded.png` were transferred to the home office, and compared.

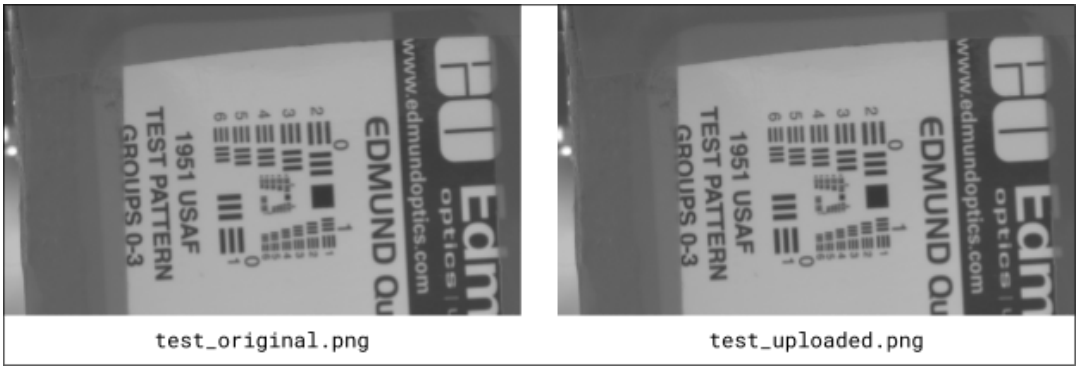


Figure 4: Comparison between original and uploaded image

In addition to visual comparison, the command `git diff test_original.png test_uploaded.png` was used. This command produced no output, which concludes that the images are identical. The procedure is therefore valid, and can be used to upload boot files to the ZedBoard over CAN.

4.2.3 Actual Procedures

The files were uploaded in the following order: `bitstream.bit`, `BOOT.BIN` and `image.ub`. The bitstream file does not overwrite anything, the image file is the largest. For all files, the procedure is the same. The commands for each step except from the list commands are documented. Only significant output is documented.

4.2.3.1 bitstream.bit

1. `(hypso) ft register 12 /mnt/boot/bitstream.bit.fmt 1`

2. `(hypso) ft prepare local bitstream.bit bitstream.bit.fmt 1 240`
   ```
   file_name:      bitstream.bit.fmt
   file_type:      STATIC
   file_id:        1
   entry_sz:       240
   max_entries:    16857
   first_entry_id: 1
   total_entries:  16857
   ```
3. `ft format 12 1 240 16857`
   ```
   File ID: 1, status: 0
   ```
4. `(hypso) ft upload file 12 bitstream.bit.fmt 1`
   Re-started transfer 2 times, making the upload divided into 3:
   1. Completely uploaded ranges 1 - 3232
   2. Completely uploaded ranges 3232 - 4977

3. Finished upload

5. `(hypso) ft extract 12 1 /mnt/boot/bitstream.bit`
   This step was not done before after the upload of due to an oversight. To fix this, the file `/mnt/boot/bitstream.bit.fmt` was registered again by executing step 1. Then, this step was executed. Due to a known error in the implementation, the extracted file was named instead of . This was corrected using SSH.
6. `(hypso) ft deregister 12 1`

4.2.3.2 BOOT.BIN

1. `(hypso) ft register 12 /mnt/boot/BOOT.BIN.fmt 1`

2. `(hypso) ft prepare local BOOT.BIN BOOT.BIN.fmt 1 240`
   ```
   file_name:      BOOT.BIN.fmt
   file_type:      STATIC
   file_id:        1
   entry_sz:       240
   max_entries:    2858
   first_entry_id: 1
   total_entries:  2858
   ```
3. `(hypso) ft format 12 1 240 2858`
   `File ID: 1, status: 0`
4. `(hypso) ft upload file 12 BOOT.BIN.fmt 1`
   Connection timed out after completing ranges 1 - 2689
   Second try successfully uploaded the rest
5. `(hypso) ft extract 12 1 /mnt/boot/BOOT.BIN`

6. `(hypso) ft deregister 12 1`
7.

4.2.3.3 image.ub

1. `(hypso) ft register 12 /mnt/boot/image.ub.fmt 1`

2. `(hypso) ft prepare local image.ub image.ub.fmt 1 240`
   ```
   file_name:      image.ub.fmt
   file_type:      STATIC
   file_id:        1
   entry_sz:       240
   max_entries:    338031
   first_entry_id: 1
   total_entries:  338031
   ```
3. `ft format 12 1 240 338031`
   ```
   /home/hypso/src/ft/ft_client.c:332:ft_client_format: Failed to get
   format response.
   `ft format` exited with return value: -110
   ```

After attempting the procedure a few times, the results were the same. The file was then uploaded using SCP instead.

5 Verification

After all three files were placed in `/mnt/boot/`, they were copied to the home computer for a final verification against the original files.

- `Bitstream.bit`: No changes
- `bitstream.bit` and `bitstream.bit`: No changes to the file itself, but execution was added to permissions.

Due to the permission change, the files were again used to boot the local ZedBoard, which booted successfully with correct IP.

This is enough to try to reboot the zedBoard with the new boot files. With `opu-sevices` running on the ZedBoard, the command `csp reboot 12` was used to reboot.

Being logged into the HIL workstation, the boot process was monitored over uart using the commando `sudo picocom /dev/ttyACM0 -b 115200`. The process was running as normal until the startup script. The script did not find a `swapfile` and froze:

```
################################# HYPSO STARTUP SCRIPT BELOW
################################
mount: mounting /dev/mmcblk1p2 on /media/emmc failed: No such file or
directory
mkswap: can't open '/media/sd/swapfile': No such file or directory
```

No effort to communicate with the ZedBoard were successful. UART, CSP/CAN and Ethernet were tried without any response.

6 Discussion

6.1 Generation and Testing of Boot Files

The boot files created for this procedure were able to do what they were supposed to - when all files were included in the boot-partition on the SD-card. While being aware that the card in the ZedBoard in the lab was not big enough to include the `swapfile`, this was not tested with the actual files created. If this had been tested, the files from this commit would not have been used, since the system froze because of an issue in the startup script. The ZedBoard would have been able to boot without the `swapfile` in normal circumstances.

6.2 Uploading of Boot Files

This part simulates the operational situation to some extent. In operation the files would first have been uplinked over a radiolink (SCP in this procedure), then buffered on the payload controller (HIL computer in this procedure) and then transferred over CAN/CSP (as in this procedure). The transfer media and commands for uplink differ, but the steps resemble each other.

The transfer procedure from HIL computer to the ZedBoard consists of a good number of commands, that possibly could be run together as one command. We also see that the `ft format` command fails for the `image.ub` file. This needs to be investigated. It could be because of the file size. These issues have been addressed on GitHub so that those with insight of the implementation can address them.

6.3 Verification

The verification process would have discovered any discrepancies between the uploaded files and the files generated by petalinux. The `git diff` command is a powerful tool that would have caught any difference between the files.

A step that should have been carried out was to boot the available ZedBoard with the files uploaded as a final verification step. This would have caught the fault, and a solution could have been investigated before trying again.

7 Conclusion

Although the ZedBoard did not boot into a useable state, there were valuable giveaways from trying. Firstly, the upload procedure was tested thoroughly, and some possible improvements have been addressed. Secondly, the importance of testing *everything* showed itself from this procedure. Even though something has been previously tested and should work in theory, it might not work this time because some other related component has changed.

11 List of Abbreviations

Table 3: List of Abbreviations

Abbrv.	Description
ABD	Aided Blind Deconvolution
AC	Atmospheric Correction
AIT	Assembly, Integration and Test
ADC	Analog to Digital Converter
ADCS	Attitude Determination and Control System
AOCS	Attitude and Orbit Control System
AoI	Area of Interest
API	Application Programming Interface
AxV	Autonomous Vehicles
BB	Breadboard
BER	Bit Error Rate
CAD	Computer Aided Design
CAN	Controlled Area Network
CCSDS	Consultative Committee for Space Data Systems
CDR	Critical Design Review
CoG/COG	Centre of Gravity
COM	Communication
CoM	Center of Mass
COTS	Commercial Off-The-Shelf
CPU	Central Processing Unit
CSP	Cubesat Space Protocol
CTE	Coefficient of Thermal Expansion
DAC	Digital to Analog Converter
DN	Digital Number

DSP	Digital Signal Processor
ECEF	Earth Centered Earth Fixed
ECI	Earth Centered Inertial
EEE	Electrical, Electronic and Electro-mechanical
EM	Engineering Model
EPS	Electric Power System
ESA	European Space Agency
FC	Flight Computer
FEM	Finite Element Method
FFT	Fast Fourier Transform
FM	Flight Model
FOV	Field of View
FPGA	Field Programmable Gate Array
FPS	Frames Per Second
FRR	Flight Readiness Review
FWHM	Full-Width Half-Maximum
GMSK	Gaussian Minimum Shift Keying
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
GSE	Ground Support Equipment
HIL	Hardware In-the-Loop
HSI	HyperSpectral Imager
HW	Hardware
HYPSO	HYPer-spectral Smallsat for Ocean observation
ICD	Interface Control Document
IMU	Inertial Measurement Unit
IOCCG	International Ocean-Colour Coordinating Group
IOD	In Orbit Demonstration

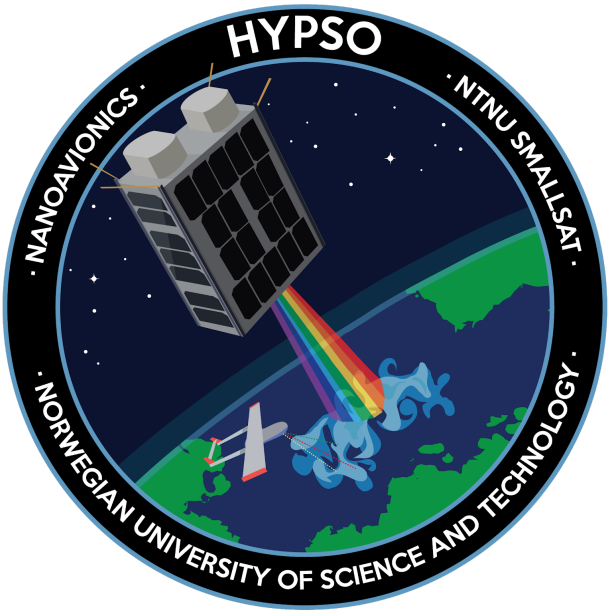
IOP	Inherent Optical Properties
IR	InfraRed
I2C	Inter-Integrated Circuit
LEO	Low-Earth Orbit
LEOP	Launch and Early Orbit Phase
LNA	Low Noise Amplifier
LQR	Linear-Quadratic Regulator
Lw	Water Leaving Radiance
MM	Mass Model
MoI/MOI	Moment of Inertia
MPC	Model Predictive Control
MTF	Modular Transfer Function
NASA	National Aeronautics and Space Administration
NTNU	Norwegian University of Science and Technology
OBPG	Ocean Biology Processing Group
OTFP	On-The-Fly-Processing
PA	Power Amplifier
PCB	Printed Circuit Board
PDR	Preliminary Design Review
PID	Proportional-Derivative-Integral
PSD	Power Spectral Density
PSF	Point Spread Function
QAR	Qualification and Acceptance Review
RAM	Random Access Memory
RF	Radio Frequency
RGB	Red-Green-Blue
RMS	Root-Mean-Square
RW	Reaction Wheel

RX	Receive
SCP	Secure Copy
SD	Secure Digital
SDR	Software Defined Radio
SNR	System to Noise Ratio
SOC	System-on-Chip
SOM	System-on-Module
SSH	Secure Shell
SST	NX Space Systems Thermal
STM	Structural Thermal Models
SW	Software
SWIR	Short-Wave Infrared
TBC	To Be Confirmed
TBD	To Be Determined
TM/TC	Telemetry/Telecommand
TRL	Technology Readiness Level
TRB	Test Review Board
TRR	Test Readiness Review
TX	Transmit
UART	Universal Asynchronous Receiver-Transmitter
UHF	Ultra High Frequency
UxV	Unmanned Vehicles
VPN	Virtual Private Network
WCS	World Coordinate System

B Test Report TRP-EL-017

Integration and Testing of Cube DMA

HYPSO-TRP-EL-017



Prepared by: HYPSO Project Team
Reference: HYPSO-TRP-EL-017
Revision: 1
Date of issue: Date
Status: Preliminary
Document Type: TBD

Table Of Contents

Table 1: Table of Changes

Rev.	Summary of Changes	Author(s)	Effective Date
1	*First issue*	*Magnus Danielsen*	

1 Overview

The HYPSO Mission will primarily be a science-oriented technology demonstrator. It will enable low-cost & high-performance hyperspectral imaging and autonomous onboard processing that fulfill science requirements in ocean color remote sensing and oceanography. NTNU SmallSat is prospected to be the first SmallSat developed at NTNU with launch planned for Q4 2020 followed by a second mission later. Furthermore, vision of a constellation of remote-sensing focused SmallSat will constitute a space-asset platform added to the multi-agent architecture of UAVs, USVs, AUVs and buoys that have similar ocean characterization objectives.

1.1 Purpose

The purpose of this document is to document how to interface with the Cube DMA module in the imaging pipeline, and provide documentation of testing.

The the module is to be tested in the context it will be used, also called an integration test. That is as a part of opu-services, with the functionality that exists in the code base and not as a standalone implementation. It is also desirable to have this included in the GitHub repository to be trackable.

1.2 Scope

The Cube DMA is a module that moves data between the processing system and programmable logic on the OPU. Hyperspectral frames are captured by an HSI camera, binned by the OPU and sent to the Cube DMA for streaming to the FPGA, where one or more modules will process the data, before it is returned through the Cube DMA back to the processing system.

This document will mainly verify that the module works by looping data back to itself, with no processing performed on the data.

1.3 Summary

1.4 Applicable Documents

The following table lists the applicable documents for this document and work.

Table 2: Applicable Documents

ID	Author	Title
[AD01]		
[AD02]		
[AD03]		
[AD04]		
[AD05]		

1.5 Referenced Documents

The documents listed in have been used as reference in creation of this document.

Table 3: Referenced Documents

ID	Author	Title
[RD01]	Johan Fjeldtvedt	Testing of Communication Between Various Peripherals on ZedBoard
[RD02]	HYPSO Software Team	Hypso-sw GitHub Repository
[RD03]	HYPSO Software Team	Opu-system GitHub Repository
[RD04]		
[RD05]		

2 Tools and Equipment

The tools used in this test was a ZedBoard development kit by AVNET, a USB to CAN module by SysWorxx, boot images built by [RD03] at commit 339905d and software built by [RD02] at commit 21b0d1aa.

The boot images were built without the CCSDS 123 (compression) module in Vivado.

3 Software

The software used in this test is from commit 21b0d1aa on [RD02]. A CSP command has been made to execute the DMA test on the OPU.

The top level of the test is implemented in hsi-service.c, and is implemented as follows:

1. Declare required data structures

```
uint16_t* in_data = NULL;
uint8_t* out_data = NULL;

unsigned int bytes_in = 0;
unsigned int bytes_out = 0;
```

2. Initialize the Cube DMA

```
cubedma_init(&in_data, &out_data);
```

3. Generate data

```
for(int i = 0; i < CUBE_SIZE; i++)
{
    in_data[i] = i % 0xFFFF;
    bytes_in += 2;
}
```

4. Send the data to the Cube DMA

```
int err = compress_cube_fpga(&bytes_out);

if(err)
{
    return 1;
}
```

5. Check the output
    a. If the number of bytes in equals the number of bytes out, the cube DMA should be connected to itself. If this is the case, the in- and output data is equal

```
if(bytes_out == bytes_in)
{
    printf("The number of in- and out-bytes are the same\n");
    unsigned int equals = 0;
    uint16_t out_data_compound = 0;
    for(unsigned int i = 0; i < bytes_out/2; i++)
    {
```

```
        // To compare uint8_t to uint16_t,
        // two uint8_t's must be joined
        out_data_compound = (out_data[2 * i + 1] << 8)
                            + out_data[2 * i];
        if(in_data[i] == out_data_compound)
        {
            equals++;
        }
        else
        {
            printf("->The data is not equal at %i (0x%06x):\n",
                   i, i*4);
            printf("    In-data:  0x%04x\n", in_data[i]);
            printf("    Out-data: 0x%04x\n", out_data_compound);
            break;
        }
    }
    if(equals == bytes_out/2)
    {
        printf("->The data is equal!\n");
    }
}
```

b. If the number of bytes out is smaller than bytes in, then it is reason to believe compression is included in the FPGA

```
else if(bytes_out < bytes_in)
{
   printf("The data has been compressed from %i to %i bytes!\n",
          bytes_in, bytes_out);
}
```

c. If neither a or b is the case, this is unspecified behaviour, and a general error message is printed

```
else
{
    printf("The output data is larger that the input!\n");
}
```

This test uses two functions from other modules: cubedma_init() from cubeDMA.c and compress_cube_fpga() from hsi_util.c. Usually, the Cube DMA streams data to a ccsds 123 compression core in the FPGA, hence the function name. In this case, the Cube DMA loops data back to itself.

The files cubeDMA.c and .h are included in appendices A and B, the function compress_cube_fpga() in appendix C, and the source code for the kernel module required for Cube DMA is in appendix D.

3 Method

A computer and a ZedBoard are connected via a CAN-bus with a sysworxx USB to CAN adapter. The computer is running hypso-cli with id 16 in one terminal and the ZedBoard is running opu-services with id 12 over UART in another. The computer sends the command hsi dmatest, and the output is shown in the second terminal.

This has been tested on two separate ZedBoards: ZB01 and ZB06.

Results

Output from cli-terminal:

```
Logging stdout to '2020-04-17T11:02:25Z_hypso-cli.log'.
Logging stderr to '2020-04-17T11:02:25Z_hypso-cli.log'.
Init can interface can0
Hostname:  smallsatlab
Model:     hypso-cli
Revision:  Apr 14 2020
Type list to see commands.
Type help <command> for specific help.
(hypso) hsi dmatest
--> Sending Request to test DMA
   ACK Timeout
`hsi dmatest` failed: 1 (Operation not permitted)
(hypso)
```

Output from opu-terminal:

```
/home/hypso/src/utils/logging.c:339:log_init_std_streams: Failed to
openxilinx_can e0008000.can can0: bitrate error 0.
0%
log_stream 2020-04-07T08:04:22Z_opu-services.log (Invalid argument)
Init can interface can0
CSP initialisation complete
Hostname: opu
Model:    opu-services
Revision: Apr 14 2020
[Started] CSP Services              Thread ID: 1630
[Started] File Transfer Service     Thread ID: 1631
[Started] TFTP Service              Thread ID: 1632
[Started] CLI Service               Thread ID: 1633
[Started] RGB Camera Service        Thread ID: 1634
[Started] CLAW-1 Payload Service    Thread ID: 1637
   RGB WARNING: No ueye cameras are connected!
Received DMA Test request
   Generating data
       Starting transfer
   MM2S completed after 354437 "times"
   S2MM completed after 0 "times"
   Transfer success
The number of in- and out-bytes are the same
->The data is equal!
End of test!
```

5 Conclusion

This test shows that the both the Cube DMA module and interface works.

6 List of Abbreviations

Table 3: List of Abbreviations

Abbrv.	Description
ABD	Aided Blind Deconvolution
AC	Atmospheric Correction
AIT	Assembly, Integration and Test
ADC	Analog to Digital Converter
ADCS	Attitude Determination and Control System
AOCS	Attitude and Orbit Control System
AoI	Area of Interest
API	Application Programming Interface
AxV	Autonomous Vehicles
BB	Breadboard
BER	Bit Error Rate
CAD	Computer Aided Design
CAN	Controlled Area Network
CCSDS	Consultative Committee for Space Data Systems
CDR	Critical Design Review
CoG/COG	Centre of Gravity
COM	Communication
CoM	Center of Mass
COTS	Commercial Off-The-Shelf
CPU	Central Processing Unit
CSP	Cubesat Space Protocol
CTE	Coefficient of Thermal Expansion
DAC	Digital to Analog Converter
DMA	Direct Memory Acess

DN	Digital Number
DSP	Digital Signal Processor
ECEF	Earth Centered Earth Fixed
ECI	Earth Centered Inertial
EEE	Electrical, Electronic and Electro-mechanical
EM	Engineering Model
EPS	Electric Power System
ESA	European Space Agency
FC	Flight Computer
FEM	Finite Element Method
FFT	Fast Fourier Transform
FM	Flight Model
FOV	Field of View
FPGA	Field Programmable Gate Array
FPS	Frames Per Second
FRR	Flight Readiness Review
FWHM	Full-Width Half-Maximum
GMSK	Gaussian Minimum Shift Keying
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
GSE	Ground Support Equipment
HIL	Hardware In-the-Loop
HSI	HyperSpectral Imager
HW	Hardware
HYPSO	HYPer-spectral Smallsat for Ocean observation
ICD	Interface Control Document
IMU	Inertial Measurement Unit
IOCCG	International Ocean-Colour Coordinating Group

IOD	In Orbit Demonstration
IOP	Inherent Optical Properties
IR	InfraRed
I2C	Inter-Integrated Circuit
LEO	Low-Earth Orbit
LEOP	Launch and Early Orbit Phase
LNA	Low Noise Amplifier
LQR	Linear-Quadratic Regulator
Lw	Water Leaving Radiance
MM	Mass Model
MoI/MOI	Moment of Inertia
MPC	Model Predictive Control
MTF	Modular Transfer Function
NASA	National Aeronautics and Space Administration
NTNU	Norwegian University of Science and Technology
OBPG	Ocean Biology Processing Group
OTFP	On-The-Fly-Processing
PA	Power Amplifier
PCB	Printed Circuit Board
PDR	Preliminary Design Review
PID	Proportional-Derivative-Integral
PSD	Power Spectral Density
PSF	Point Spread Function
QAR	Qualification and Acceptance Review
RAM	Random Access Memory
RF	Radio Frequency
RGB	Red-Green-Blue
RMS	Root-Mean-Square

RW	Reaction Wheel
RX	Receive
SCP	Secure Copy
SD	Secure Digital
SDR	Software Defined Radio
SNR	System to Noise Ratio
SOC	System-on-Chip
SOM	System-on-Module
SSH	Secure Shell
SST	NX Space Systems Thermal
STM	Structural Thermal Models
SW	Software
SWIR	Short-Wave Infrared
TBC	To Be Confirmed
TBD	To Be Determined
TM/TC	Telemetry/Telecommand
TRL	Technology Readiness Level
TRB	Test Review Board
TRR	Test Readiness Review
TX	Transmit
UART	Universal Asynchronous Receiver-Transmitter
UHF	Ultra High Frequency
UxV	Unmanned Vehicles
VPN	Virtual Private Network
WCS	World Coordinate System

Appendix A - cubeDMA.h

```
/*
Shared data between user space and kernel space
*/
#ifndef CUBEDMA_H_
#define CUBEDMA_H_

#include <stdint.h>

#define CUBE_DEPTH  107 //Spectral columns
#define CUBE_HEIGHT 500 //Number of frames
#define CUBE_WIDTH  720 //Spatial rows
#define CUBE_SIZE (CUBE_DEPTH * CUBE_HEIGHT * CUBE_WIDTH)
#define SEND_PHYS_ADDR 0x30000000
#define RECEIVE_PHYS_ADDR 0x38000000
#define CUBEDMA_BASE 0x43C00000

typedef enum
{
      MM2S_CTRL_REG = 0,      //0x00
      MM2S_STAT_REG,          //0x04
      MM2S_ADDR_REG,          //0x08
      MM2S_CUBE_DIM_REG,      //0x0C
      MM2S_BLOCK_DIM_REG,     //0x10
      MM2S_ROW_DIM_REG,       //0x14

      S2MM_CTRL_REG = 8,//0x20
      S2MM_STAT_REG,          //0x24
      S2MM_ADDR_REG,          //0x28
      S2MM_LEN_REG            //0x2C
} reg_t;

typedef enum
{
      MM2S,
      S2MM
} transfer_t;

/*
      Makes the pointers to the binned and compressed cube point to physical
memory addresses
      reserved for the Cube DMA in the kernel

      hsi_config: struct with configurations for the hsi camera
      binned_cube: pointer to memory location of the binned cube
      compressed_cube: pointer to memory location of the compressed cube
*/
bool cubedma_init(uint16_t** send_channel, uint8_t** receive_channel);
```

```
/*
      Configures the registers of the CubeDMA
*/
void cubedma_configure(void);

/*
      Prints the contents of a specified register

      reg: a register Cube DMA uses
            See reg_t enum for full list
*/
void cubedma_PrintRegister(reg_t reg);

/*
      Signals to the Cube DMA to start transfers
*/
void cubedma_start_transfer(void);

/*
      Checks of a transfer is done

      transfer: either MM2S or S2MM

      returns:
            - true if done
            - false if not done
*/
bool cubedma_transfer_done(transfer_t transfer);

/*
      Prints transfer errors

      transfer: either MM2S or S2MM
*/
void cubedma_transfer_error(transfer_t transfer);

/*
      Returns the length of received bitstream in bytes
*/
unsigned int cubedma_get_received_length(void);

/*
      Cleans the cache. Must be executed after data has been placed
      in the physical memory Cube DMA reads from
*/
void clean_cache(void);

/*
      Flushes the cache. Must be executed after Cube DMA has finished
*/
```

```
void flush_cache(void);

#endif // CUBEDMA_H_
```

Appendix B - cubeDMA.c

```
#include <stdio.h>
#include <stdint.h>
#include <stdlib.h>
#include <unistd.h>
#include <fcntl.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <sys/ioctl.h>
#include <sys/mman.h>

#include "HYPSO.h"
#include "hsi/cubeDMA.h"

#define SR_DONE_MSK 0x1

#define cubedma_RegWrite(reg, value) deviceMem[reg] = value;
#define cubedma_RegRead(reg) deviceMem[reg]

#define CTRL_REG_OFFSET(transfer) (transfer==MM2S)? \
    (MM2S_CTRL_REG) : (S2MM_CTRL_REG)

#define STAT_REG_OFFSET(transfer) (transfer==MM2S)? \
    (MM2S_STAT_REG) : (S2MM_STAT_REG)


static int fd_send;
static int fd_receive;

uint16_t *send_channel_test;
uint16_t *receive_channel_test;
uint32_t *deviceMem;

bool cubedma_init(uint16_t** send_channel, uint8_t** receive_channel)
{
    // Using kernel module functionality to get memory addresses
    // for send and receive channels, and CubeDMA registers

    int fd = open("/dev/mem", O_RDWR|O_SYNC);

    if (fd < 0)
    {
        perror("/dev/mem");
        return 1;
    }

    fd_send = open("/dev/cubedmasend", O_RDWR);
    if(fd_send < 1)
```

```
    {
        printf("        Unable to open CubeDMA send channel");
        return 1;
    }

    fd_receive = open("/dev/cubedmarecieve", O_RDWR);
    if (fd_receive < 1)
    {
        printf("        Unable to open receive channel");
        return 1;
    }

    *send_channel = mmap(0, CUBE_SIZE*sizeof(uint16_t),
                    PROT_READ | PROT_WRITE, MAP_SHARED, fd_send,
                    SEND_PHYS_ADDR);

    *receive_channel = mmap(0, CUBE_SIZE*sizeof(uint16_t),
                       PROT_READ | PROT_WRITE, MAP_SHARED, fd_receive,
                       RECEIVE_PHYS_ADDR);

    deviceMem = (uint32_t *) mmap(NULL, getpagesize(), PROT_READ|PROT_WRITE,
                             MAP_SHARED, fd, CUBEDMA_BASE);

    if((*send_channel == NULL) || (*receive_channel == NULL))
    {
        return 1;
    }

    // Initializing the control registers to 0
    cubedma_RegWrite(MM2S_CTRL_REG, 0x0);
    cubedma_RegWrite(S2MM_CTRL_REG, 0x0);

    return 0;
}

void cubedma_configure(void)
{
    /*
        Setting the dimentions of the cube in the
        correct bits according to the dodumentation
        Johans Master's Thesis, page 29
    */
    uint32_t dims =
        (CUBE_WIDTH  & 0xFFF) << 0  |
        (CUBE_HEIGHT & 0xFFF) << 12 |
        (CUBE_DEPTH  & 0xFF)  << 24;

    cubedma_RegWrite(MM2S_ADDR_REG, (uint32_t)(SEND_PHYS_ADDR));
    cubedma_RegWrite(MM2S_CUBE_DIM_REG, dims);
    cubedma_RegWrite(MM2S_BLOCK_DIM_REG, 0x0); // We don't use blocks
    cubedma_RegWrite(MM2S_ROW_DIM_REG, CUBE_WIDTH * CUBE_DEPTH);
```

```
    cubedma_RegWrite(S2MM_ADDR_REG, (uint32_t)(RECEIVE_PHYS_ADDR));
}

void cubedma_PrintRegister(reg_t reg)
{
    char* reg_string;
    switch (reg)
    {
        case MM2S_CTRL_REG:
            reg_string = "MM2S_CTRL_REG      :";
        break;

        case MM2S_STAT_REG:
            reg_string = "MM2S_STAT_REG      :";
        break;

        case MM2S_ADDR_REG:
            reg_string = "MM2S_ADDR_REG      :";
        break;

        case MM2S_CUBE_DIM_REG:
            reg_string = "MM2S_CUBE_DIM_REG  :";
        break;

        case MM2S_BLOCK_DIM_REG:
            reg_string = "MM2S_BLOCK_DIM_REG :";
        break;

        case MM2S_ROW_DIM_REG:
            reg_string = "MM2S_ROW_DIM_REG   :";
        break;

        case S2MM_CTRL_REG:
            reg_string = "S2MM_CTRL_REG      :";
        break;

        case S2MM_STAT_REG:
            reg_string = "S2MM_STAT_REG      :";
        break;

        case S2MM_ADDR_REG:
            reg_string = "S2MM_ADDR_REG      :";
        break;

        case S2MM_LEN_REG:
            reg_string = "S2MM_LEN_REG       :";
        break;

        default:
            return;
    }
```

```
    printf("        Contents of %s0x%08x\n", reg_string,
cubedma_RegRead(reg));
}

void cubedma_start_transfer(void)
{
    clean_cache();

    cubedma_configure();

    printf("        Starting transfer\n");
    cubedma_RegWrite(MM2S_CTRL_REG, 0x1);
    cubedma_RegWrite(S2MM_CTRL_REG, 0x1);
}

bool cubedma_transfer_done(transfer_t transfer)
{
    if (cubedma_RegRead(STAT_REG_OFFSET(transfer)) & SR_DONE_MSK)
    {
        if (transfer == S2MM)
        {
            cubedma_RegWrite(S2MM_CTRL_REG, 0x0);
            flush_cache(); // Assuming that S2MM is the last transfer to
finish
        }
        else
        {
            cubedma_RegWrite(MM2S_CTRL_REG, 0x0);
        }
        return true;
    }
    return false;
}

void cubedma_transfer_error(transfer_t transfer)
{
    char* type;

    if (transfer == MM2S)
    {
        type = "MM2S";
    } else
    {
        type = "S2MM";
    }

    uint32_t err = cubedma_RegRead(STAT_REG_OFFSET(transfer)) << 4;
    err &= 1;

    printf("        Error in transfer %s: %x\n", type, err);
}
```

```
unsigned int cubedma_get_received_length(void)
{
    return (unsigned int)deviceMem[S2MM_LEN_REG];
}

void clean_cache(void)
{
    ioctl(fd_send, 0);
}

void flush_cache(void)
{
    ioctl(fd_receive, 1);
}
```

Appendix C - compress_cube_fpga() from hsi_util.c

```
bool compress_cube_fpga(unsigned int* compressed_bytes)
{
    cubedma_start_transfer();

    bool MM2S_success = 0;
    bool S2MM_success = 0;
    volatile uint32_t time;

    for (time = 0; time < TIMEOUT; time++)
    {
        if (cubedma_transfer_done(MM2S))
        {
            printf("    MM2S completed after %i \"times\"\n", time);
            MM2S_success = 1;
            break;
        }
    }

    if (time == TIMEOUT)
    {
        printf("    ERROR: MM2S timeout\n");
    }

    for (time = 0; time < TIMEOUT; time++)
    {
        if (cubedma_transfer_done(S2MM))
        {
            printf("    S2MM completed after %i \"times\"\n", time);
            S2MM_success = 1;
            break;
        }
    }
    if (time == TIMEOUT)
    {
        printf("    ERROR: S2MM timeout\n");
    }

    if (MM2S_success && S2MM_success)
    {
        printf("    Transfer success\n");
        *compressed_bytes = cubedma_get_received_length();
        return 0;
    }
    else
    {
        printf("    CubeDMA transfer not successfull.\n");
        return 1;
```

```
    }
}
```

Appendix D - CubeDMA kernel module

The source code was implemented by another member of the HYPSO team based on the implementation of [RD01]

```
include <asm/cacheflush.h>
#include <asm/outercache.h>
#include <linux/ioctl.h>
#include <linux/module.h>         // Core header for loading LKMs into the
kernel
#include <linux/device.h>         // Header to support the kernel Driver
Model

#include "dma_parameters.h"



struct device_data{
      int    major_number;               ///< Stores the device number --
determined automatically
      struct class*  p_device_class;     ///< The device-driver class
struct pointer
      struct device* p_device;                        ///< The device-driver
device struct pointer
      struct cdev cdev;
      uint16_t * p_dma_data
};

static struct device_data dma_channel[2]; //0=send 1=recieve

static int     dev_open(struct inode *, struct file *);
static int     dev_release(struct inode *, struct file *);
static ssize_t dev_read(struct file *, char *, size_t, loff_t *);
static ssize_t dev_write(struct file *, const char *, size_t, loff_t *);



static const struct vm_operations_struct uio_physical_vm_ops = {
#ifdef CONFIG_HAVE_IOREMAP_PROT
      .access = generic_access_phys,
#endif
};

// Maps physical send or recieve adresses to user space
static int dev_mmap(struct file *file_p, struct vm_area_struct *vma){
      struct inode *inode = (struct inode *)file_p->private_data;

      vma->vm_ops = &uio_physical_vm_ops;
      int ret = 0;
```

```
	if(imajor(inode)==dma_channel[0].major_number){//send
		ret = remap_pfn_range(vma, vma->vm_start, SEND_PHYS_ADDR >>
PAGE_SHIFT, vma->vm_end - vma->vm_start, vma->vm_page_prot);
	}
	else{//recieve
		ret = remap_pfn_range(vma, vma->vm_start, RECIEVE_PHYS_ADDR >>
PAGE_SHIFT, vma->vm_end - vma->vm_start, vma->vm_page_prot);
	}
	return ret;
}

// Clean send and recieve caches, or invalidate recieve buffer
static long dev_ioctl(struct file* file, unsigned int cmd, unsigned long
arg){
	if(cmd==0){//Clean whole thing
		__cpuc_flush_dcache_area(dma_channel[0].p_dma_data,
(CUBE_SIZE)*sizeof(__UINT16_TYPE__));
		__cpuc_flush_dcache_area(dma_channel[1].p_dma_data,
(CUBE_SIZE)*sizeof(__UINT16_TYPE__));
		outer_clean_range(SEND_PHYS_ADDR, 0x40000000);
	}
	else{//Invalidate recieve
		outer_inv_range(RECIEVE_PHYS_ADDR, 0x40000000);
		__cpuc_flush_dcache_area(dma_channel[1].p_dma_data,
(CUBE_SIZE)*sizeof(__UINT16_TYPE__));
	}
	return 0;
}

static struct file_operations fops = {
	.open = dev_open,
	.read = dev_read,
	.write = dev_write,
	.release = dev_release,
	.mmap = dev_mmap,
	.unlocked_ioctl = dev_ioctl
};

// Initialize both recieve and send devices
static int __init init_channel(struct device_data *dma_chan, char *name){
	int error_message;

	char dev_name[32] = DEVICE_NAME;
	strcat(dev_name, name);

	char class_name[32] = DEVICE_NAME;
	strcat(class_name, name);

	if(strcmp(name, "send")==0){
		if(request_mem_region(SEND_PHYS_ADDR,
(CUBE_SIZE)*sizeof(__UINT16_TYPE__), "cubedma send") == NULL) {
```

```
                  printk(KERN_ALERT "Failure to obtain memory region \n");
                  return -EBUSY;
            }
      } else {
            if(request_mem_region(RECIEVE_PHYS_ADDR,
(CUBE_SIZE)*sizeof(__UINT16_TYPE__), "cubedma receive") == NULL) {
                  printk(KERN_ALERT "Failure to obtain memory region \n");
                  return -EBUSY;
            }
      }

      cdev_init(&dma_chan->cdev, &fops);
      dma_chan->cdev.owner = THIS_MODULE;

      error_message = cdev_add(&dma_chan->cdev,
MKDEV(dma_chan->major_number, 0), 1);
      if (error_message) {
            dev_err(dma_chan->p_device, "unable to add char device\n");
      }

      dma_chan->major_number = register_chrdev(0, dev_name, &fops);
      if (dma_chan->major_number<0){
            printk(KERN_ALERT "Device failed to register a major number\n");
            return dma_chan->major_number;
      }

      dma_chan->p_device_class = class_create(THIS_MODULE, class_name);
      if (IS_ERR(dma_chan->p_device_class)){                // Check for
error and clean up if there is
            unregister_chrdev(dma_chan->major_number, dev_name);
            printk(KERN_ALERT "Failed to register device class\n");
            return PTR_ERR(dma_chan->p_device_class);          // Correct
way to return an error on a pointer
      }

      dma_chan->p_device = device_create(dma_chan->p_device_class, NULL,
MKDEV(dma_chan->major_number, 0), NULL, dev_name);
      if (IS_ERR(dma_chan->p_device)){               // Clean up if there is
an error
            class_destroy(dma_chan->p_device_class);           // Repeated
code but the alternative is goto statements
            unregister_chrdev(dma_chan->major_number, dev_name);
            printk(KERN_ALERT "Failed to create the device\n");
            return PTR_ERR(dma_chan->p_device);
      }
      printk(KERN_INFO "Device registered correctly with major number %d\n",
dma_chan->major_number);


      // Map memory to kernel space
      if(strcmp(name, "send")==0){
```

```
            dma_chan->p_dma_data = memremap(SEND_PHYS_ADDR,
(CUBE_SIZE)*sizeof(__UINT16_TYPE__), MEMREMAP_WB);
      }
      else{
            dma_chan->p_dma_data = memremap(RECIEVE_PHYS_ADDR,
(CUBE_SIZE)*sizeof(__UINT16_TYPE__), MEMREMAP_WB);
      }

      if(dma_chan->p_dma_data==NULL){
            printk(KERN_INFO "Failed to map memory to kernel module\n");
      }
      else{
            printk(KERN_INFO "Device allocated memory at 0x%08X\n",
(unsigned int)dma_chan->p_dma_data);
      }
      return 0;
}

static int __init ebbchar_init(void){

      int ret;
      ret = init_channel(&dma_channel[0], "send");

      if (ret) {

            return ret;
      }

      ret = init_channel(&dma_channel[1], "recieve");
      if (ret) {
            return ret;
      }

      return 0;
}

// Cleanup
static void __exit ebbchar_exit(void){
      device_destroy(dma_channel[0].p_device_class,
MKDEV(dma_channel[0].major_number, 0));     // remove the device
      class_unregister(dma_channel[0].p_device_class);
// unregister the device class
      class_destroy(dma_channel[0].p_device_class);
// remove the device class
      unregister_chrdev(dma_channel[0].major_number, "Send channel");
// unregister the major number
      cdev_del(&dma_channel[0].cdev);

      device_destroy(dma_channel[1].p_device_class,
MKDEV(dma_channel[1].major_number, 0));     // remove the device
      class_unregister(dma_channel[1].p_device_class);
```

```
// unregister the device class
      class_destroy(dma_channel[1].p_device_class);
// remove the device class
      unregister_chrdev(dma_channel[1].major_number, "Recieve channel");
// unregister the major number
      cdev_del(&dma_channel[1].cdev);

      release_mem_region(SEND_PHYS_ADDR,
(CUBE_SIZE)*sizeof(__UINT16_TYPE__));
      release_mem_region(RECIEVE_PHYS_ADDR,
(CUBE_SIZE)*sizeof(__UINT16_TYPE__));
      printk(KERN_INFO "EBBChar: Goodbye from the LKM!\n");
}

static int dev_open(struct inode *inodep, struct file *filep){
      filep->private_data = inodep;
      return 0;
}


static ssize_t dev_read(struct file *filep, char *buffer, size_t len, loff_t
*offset){
      return 0;
}

static ssize_t dev_write(struct file *filep, const char *buffer, size_t len,
loff_t *offset){
      return 0;
}

static int dev_release(struct inode *inodep, struct file *filep){
  return 0;
}

module_init(ebbchar_init);
module_exit(ebbchar_exit);
MODULE_LICENSE("GPL");
```

C hypso-sw Issue 54

Edit | New issue

Jump to bottom

Integrate services for the HSI to the CLI #54

Closed **magnudan** opened this issue on Jan 28 · 1 comment

Assignees	
Labels	blocked points=inf
Projects	SW kanban board

magnudan commented on Jan 28

The functionality in src/hsi on the branch hsi_integration is to be divided and separated into services on the branch hsi-service. The following services are to be implemented for HSI:

- Initialize: Initialize the camera
- Configure: Set parameters such as fps, exposure time, gain ++ (see source code in hsi_integration branch) for the capture of an HSI cube
- Capture: Start the capture of an HSI cube with the number of frames to capture and name/path of the stored cube as parameters
- Deinit: De-initialize the camera

magnudan self-assigned this on Jan 28

magne-hov commented on Jan 31

How is this work coming along?

evelynlimore added this to **To do** in **SW kanban board** on Feb 18

evelynlimore added points=8 blocked points=inf and removed points=8 labels on Feb 19

JoarGjersund moved this from **To do** to **Backlog** in **SW kanban board** on Mar 4

evelynlimore moved this from **Backlog** to **To do** in **SW kanban board** on Mar 5

evelynlimore closed this on Mar 19

SW kanban board automation moved this from **To do** to **Done** on Mar 19

Assignees

magnudan

Labels

blocked

points=inf

Projects

SW kanban board

Done

Milestone

No milestone

Linked pull requests

Successfully merging a pull request may close this issue.

None yet

3 participants

Pin issue

D hypso-sw Issue 76

Edit | New issue

Jump to bottom

Create CSP command for uploading configuration parameters for HSI camera #76

Closed **magnudan** opened this issue on Feb 19 · 2 comments

Assignees

Labels points=3

Projects **SW kanban board**

magnudan commented on Feb 19 • edited by evelynlimore

The configuration shall be stored to file.

From sprint plan: this includes planning

magnudan self-assigned this on Feb 19

magnudan added this to **To do** in **SW kanban board** via automation on Feb 19

evelynlimore added the points=8 label on Feb 19

JoarGjersund moved this from **To do** to **Backlog** in **SW kanban board** on Mar 4

evelynlimore commented on Mar 5

Can use the existing commands to achieve this (tftp or ft upload)

evelynlimore moved this from **Backlog** to **To do** in **SW kanban board** on Mar 5

evelynlimore added points=3 and removed points=8 labels on Mar 5

evelynlimore commented on Mar 19

Not relevant, we need a software and operations' meeting figuring out what we need of functions.

evelynlimore closed this on Mar 19

SW kanban board automation moved this from **To do** to **Done** on Mar 19

Assignees

magnudan

Labels

points=3

Projects

SW kanban board

Done

Milestone

No milestone

Linked pull requests

Successfully merging a pull request may close this issue.

None yet

2 participants

Pin issue

E hypso-sw Issue 79

Edit | New issue

Jump to bottom

Create function to initialize HSI camera #79

Closed **magnudan** opened this issue on Feb 19 · 0 comments

Assignees	
Labels	points=1
Projects	SW kanban board
Milestone	MOBIP

magnudan commented on Feb 19

A generic initialization function that can be used by several CSP commands

evelynlimore added this to **To do** in **SW kanban board** on Feb 19

evelynlimore added the points=1 label on Feb 19

evelynlimore added this to the **MOBIP** milestone on Feb 19

evelynlimore moved this from **To do** to **MOBIP Sprint** in **SW kanban board** on Feb 19

evelynlimore assigned **magnudan** on Feb 19

evelynlimore moved this from **MOBIP Sprint** to **In progress** in **SW kanban board** on Feb 19

evelynlimore moved this from **In progress** to **Review in progress** in **SW kanban board** on Mar 2

JoarGjersund moved this from **Review in progress** to **Done** in **SW kanban board** on Mar 4

magnudan closed this on Apr 17

Assignees

magnudan

Labels

points=1

Projects

SW kanban board

Done

Milestone

MOBIP

Linked pull requests

Successfully merging a pull request may close this issue.

None yet

2 participants

Pin issue

F hypso-sw Issue 80

Edit | New issue

Jump to bottom

Create function to configure HSI camera #80

Closed **magnudan** opened this issue on Feb 19 · 0 comments

Assignees	
Labels	points=3
Projects	SW kanban board
Milestone	MOBIP

magnudan commented on Feb 19

The function should be able to either configure the camera with parameters from file or input directly from Operations

evelynlimore added this to **To do** in **SW kanban board** on Feb 19

evelynlimore added the points=3 label on Feb 19

evelynlimore added this to the **MOBIP** milestone on Feb 19

evelynlimore moved this from **To do** to **MOBIP Sprint** in **SW kanban board** on Feb 19

evelynlimore assigned **magnudan** on Feb 19

evelynlimore moved this from **MOBIP Sprint** to **In progress** in **SW kanban board** on Feb 19

evelynlimore moved this from **In progress** to **Review in progress** in **SW kanban board** on Mar 2

JoarGjersund moved this from **Review in progress** to **In progress** in **SW kanban board** on Mar 4

JoarGjersund moved this from **In progress** to **To do** in **SW kanban board** on Mar 4

JoarGjersund moved this from **To do** to **Backlog** in **SW kanban board** on Mar 4

evelynlimore moved this from **Backlog** to **To do** in **SW kanban board** on Mar 5

evelynlimore moved this from **To do** to **In progress** in **SW kanban board** on Mar 5

evelynlimore moved this from **In progress** to **Done** in **SW kanban board** on Mar 19

magnudan closed this on Apr 17

Assignees

magnudan

Labels

points=3

Projects

SW kanban board

Done

Milestone

MOBIP

Linked pull requests

Successfully merging a pull request may close this issue.

None yet

2 participants

Pin issue

G hypso-sw Issue 81

Edit | New issue

Jump to bottom

Create function for frame captures #81

Closed **magnudan** opened this issue on Feb 19 · 0 comments

Assignees	
Labels	points=3
Projects	SW kanban board
Milestone	MOBIP

magnudan commented on Feb 19

The function shall capture a given number of frames with the existing camera configuration. Binning shall be optional.

evelynlimore added this to **To do** in **SW kanban board** on Feb 19

evelynlimore added the points=3 label on Feb 19

evelynlimore added this to the **MOBIP** milestone on Feb 19

evelynlimore moved this from **To do** to **MOBIP Sprint** in **SW kanban board** on Feb 19

evelynlimore assigned **magnudan** on Feb 19

evelynlimore moved this from **MOBIP Sprint** to **In progress** in **SW kanban board** on Feb 19

evelynlimore moved this from **In progress** to **Review in progress** in **SW kanban board** on Mar 2

evelynlimore moved this from **Review in progress** to **Done** in **SW kanban board** on Apr 2

 **magnudan** closed this on May 1

Assignees

 **magnudan**

Labels

points=3

Projects

SW kanban board

Done

Milestone

MOBIP

Linked pull requests

Successfully merging a pull request may close this issue.

None yet

2 participants

Pin issue

H hypso-sw Issue 82

Edit | New issue

Jump to bottom

Configure interface with CubeDMA #82

Closed **magnudan** opened this issue on Feb 19 · 0 comments

Assignees	
Labels	points=8
Projects	SW kanban board
Milestone	MOBIP

magnudan commented on Feb 19

CubeDMA shall load its configurations from a file. The HSI cube shall be transferred after capture

evelynlimore added this to **To do** in **SW kanban board** on Feb 19

evelynlimore added the points=8 label on Feb 19

evelynlimore added this to the **MOBIP** milestone on Feb 19

evelynlimore moved this from **To do** to **MOBIP Sprint** in **SW kanban board** on Feb 19

evelynlimore assigned **magnudan** on Feb 19

sivertba moved this from **MOBIP Sprint** to **In progress** in **SW kanban board** on Feb 25

JoarGjersund moved this from **In progress** to **To do** in **SW kanban board** on Mar 4

JoarGjersund moved this from **To do** to **Backlog** in **SW kanban board** on Mar 4

evelynlimore moved this from **Backlog** to **To do** in **SW kanban board** on Mar 5

evelynlimore moved this from **To do** to **Done** in **SW kanban board** on Mar 27

magnudan closed this on Apr 17

Assignees

magnudan

Labels

points=8

Projects

SW kanban board

Done

Milestone

MOBIP

Linked pull requests

Successfully merging a pull request may close this issue.

None yet

2 participants

Pin issue

I hypso-sw Issue 83

Edit | New issue

Jump to bottom

Create function to exit/de-initialize the HSI camera #83

Closed **magnudan** opened this issue on Feb 19 · 0 comments

Assignees	
Labels	points=1
Projects	SW kanban board
Milestone	MOBIP

magnudan commented on Feb 19

A generic de-initialization function that can be used by several CSP commands

evelynlimore added this to **To do** in **SW kanban board** on Feb 19

evelynlimore added the points=1 label on Feb 19

evelynlimore added this to the **MOBIP** milestone on Feb 19

evelynlimore moved this from **To do** to **MOBIP Sprint** in **SW kanban board** on Feb 19

evelynlimore assigned **magnudan** on Feb 19

evelynlimore moved this from **MOBIP Sprint** to **In progress** in **SW kanban board** on Feb 19

evelynlimore moved this from **In progress** to **Review in progress** in **SW kanban board** on Mar 2

JoarGjersund moved this from **Review in progress** to **Done** in **SW kanban board** on Mar 4

magnudan closed this on Apr 17

Assignees

magnudan

Labels

points=1

Projects

SW kanban board

Done

Milestone

MOBIP

Linked pull requests

Successfully merging a pull request may close this issue.

None yet

2 participants

Pin issue

J hypso-sw Issue 84

Edit | New issue

Jump to bottom

Create CSP command to capture HSI cube w/ RGB capture for georeferencing #84

Open **magnudan** opened this issue on Feb 19 · 0 comments

Labels CMD HSI RGB points=13

Projects SW kanban board

magnudan commented on Feb 19

The command shall:

- Initialize and configure the HSI
- Initialize and configure the RGB
- Capture first 50% of given HSI frames
  - Binn + timestamp frames
- Capture a single RGB image
- Capture last 50% of given HSI frames
  - Binn + timestamp frames
- De-initialize the cameras
- Initialize CubeDMA
- Compress HSI Cube in BIP
- Store HSI Cube in BIP
- Store RGB image in TBD format
- TBD: Handle timestamps
- Send some sort of ack to operations

Notes:

- Decide how to handle timestamps
- Decide on RGB format
- Use lower level HSI functions (init, conf, capture, de-init)

evelynlimore added this to **To do** in **SW kanban board** on Feb 19

evelynlimore changed the title ~~**Create CSP command to capture HSI cube w/ RGB georeferencing**~~ **Create CSP command to capture HSI cube w/ RGB capture for georeferencing**

on Feb 19

evelynlimore added the points=13 label on Feb 19

JoarGjersund moved this from **To do** to **Backlog** in **SW kanban board** on Mar 4

sivertba mentioned this issue 19 days ago

RGB capture from HSI capture: Georeferencing image #247 Open

DennisNTNU added HSI RGB CMD labels 17 days ago

Assignees

No one—assign yourself

Labels

CMD
HSI
RGB
points=13

Projects

SW kanban board

Backlog

Milestone

No milestone

Linked pull requests

Successfully merging a pull request may close this issue.

None yet

3 participants

Pin issue

K hypso-sw Issue 85

Edit | New issue

Jump to bottom

Create CSP command to capture n frames w/o georeferencing #85

Closed **magnudan** opened this issue on Feb 19 · 6 comments

Assignees	
Labels	points=5
Projects	SW kanban board
Milestone	MOBIP

magnudan commented on Feb 19

- Just capture n frames
- Binning is optional
- No compression

evelynlimore added this to **To do** in **SW kanban board** on Feb 19

evelynlimore added the points=5 label on Feb 19

evelynlimore added this to the **MOBIP** milestone on Feb 19

evelynlimore moved this from **To do** to **MOBIP Sprint** in **SW kanban board** on Feb 19

evelynlimore assigned **magnudan** on Feb 19

evelynlimore moved this from **MOBIP Sprint** to **In progress** in **SW kanban board** on Feb 20

evelynlimore moved this from **In progress** to **Review in progress** in **SW kanban board** on Mar 2

evelynlimore assigned **rogerbirkeland** on Apr 2

evelynlimore moved this from **Review in progress** to **To do** in **SW kanban board** on Apr 16

evelynlimore moved this from **To do** to **Backlog** in **SW kanban board** on Apr 16

magnudan commented on May 1

This issue might be outdated.

- We have a CSP command for capturing n frames
- I don't remember the reasoning for optional binning
- I don't remember the reasoning for not having compression
- The CSP command we have now does both binning (and soon also HW) compression

rogerbirkeland commented on May 1

Somewhat agree to what you are saying. However, its probably good to have a
mode that does not interfere much with the raw data.

What is the state of the config file we talked about? Is that still on? If
so, I guess this issue will be included in that.

fre. 1. mai 2020, 16:37 skrev Magnus Danielsen <notifications@github.com>:
...

magnudan commented on May 4 • edited

The command `hsi debugcapture` stores the frames in all stages of processing, so that you get raw frame (also in .png), binned cube and compressed cube. If that's what you want.

I haven't worked with the config file, but **@DennisNTNU** might know something.

magnudan closed this on May 4

SW kanban board automation moved this from **Backlog** to **Done** on May 4

magnudan reopened this on May 4

SW kanban board automation moved this from **Done** to **In progress** on May 4

DennisNTNU commented on May 4

I have just commit changes that makes hsi capture work like this:

```
hsi capture -d -f 15.0 -e 30.0 -n 521
```

`-d` to enable debug mode
`-f 15.0` to set framerate to 15 fps
`-e 30.0` to set exposure time to 30ms
`-n 512` to capture 512 frames

Options to not do binning or compression should be straightforward to add as well. I am planning to add the option `-c <path>` to load a config file.

❤️ 2

rogerbirkeland commented on May 4

Cool! Which branch is this in?

DennisNTNU commented on May 4

The branch is `hsi-capture-cmd-extension`. I have done no testing on the picobob though.

DennisNTNU mentioned this issue on May 12

Hsi capture cmd extension #229 Merged

sivertba moved this from **In progress** to **Review in progress** in **SW kanban board** on May 14

evelynlimore moved this from **Review in progress** to **Done** in **SW kanban board** on May 15

DennisNTNU closed this in #229 on May 15

Assignees

rogerbirkeland

magnudan

Labels

points=5

Projects

SW kanban board

Done

Milestone

MOBIP

Linked pull requests

Successfully merging a pull request may close this issue.

Hsi capture cmd extension

4 participants

Pin issue

L hypso-sw Issue 88

Edit | New issue

Jump to bottom

Consider threading of binning HSI frames #88

Closed **magnudan** opened this issue on Feb 21 · 3 comments

Labels	Help wanted **Question** **completed**
Projects	**SW kanban board**

magnudan commented on Feb 21

In Andreas' source code for binning, two threads were spawned for binning using `#pragma omp parallel for num_threads(2)`. When porting from C++ to C, this syntax is no longer valid. The necessity for parallelism should be looked into.

magnudan added Help wanted **Question** labels on Feb 21

magnudan added this to **To do** in **SW kanban board** via automation on Feb 21

magne-hov commented on Feb 22

Considering the performance impact of this, it might be worth considering mixing C and C++ for our code. This would include compiling the main translation units (in `./apps`) as C++, so some changes would be necessary to our build defintions (and probably some declarations on our module interfaces).

drowzie commented on Feb 24

openMP is supported by GCC, but this might be a linker issue.
The compiler flag `-fopenmp` must be used to enable the `#pragma omp directive`

e.g this is the makefile i use for CCSDS 123 software implementation.

```
CC=gcc

csrc = $(wildcard src/*.c) \
       $(wildcard src/predictor/*.c) \
           $(wildcard src/cliparser/*.c) \
```

```
            $(wildcard src/encoder/*.c) \
            $(wildcard src/utils/*.c)
obj = $(csrc:.c=.o)

#NOTE: Main folder for every include is set to src
CFLAGS= -I./src/
LDFLAGS = -O3 -g -Wall -lm -fopenmp

main.out: $(obj)
        $(CC) -o $@ $^ $(LDFLAGS)
.PHONY: clean
clean:
        rm -f $(obj) main.out
```

https://gcc.gnu.org/onlinedocs/gcc-7.5.0/libgomp/Enabling-OpenMP.html

magnudan commented on Feb 25

Commit 73b207b on the branch hsi-service fixes this issue. The solution was:

- `#include <omp.h>` in `src/hsi_util.c`
- Add `gomp` to `COMMON_LIBS` in `CMakeLists.txt`
- Add compiler flag `-fopenmp` to `target_compile_options()` for `hypso-cli` and `opu-services` in `CMakeLists.txt`

Thanks to **@drowzie** for troubleshooting :)

NOTE: `-fopenmp` was not added to `target_compile_options()` for `packet-dropper`

magnudan closed this on Feb 25

SW kanban board automation moved this from **To do** to **Done** on Feb 25

magnudan added the **completed** label on Feb 25

Assignees

No one—assign yourself

Labels

Help wanted

Question

completed

Projects

SW kanban board

Done

Milestone

No milestone

Linked pull requests

Successfully merging a pull request may close this issue.

None yet

3 participants

Pin issue

M hypso-sw Issue 89

Edit | New issue

Jump to bottom

Addition of SIMD neon operations for ARM makes ARCH=x86 not compile #89

Closed **magnudan** opened this issue on Feb 21 · 3 comments

Labels Help wanted bug completed

magnudan commented on Feb 21

Binning operations for HSI frames requires instructions from header file `<arm_neon.h>`. This header is included into ARM toolchain by setting compiler flag `-mfpu=neon`. This is not (to my knowledge) possible for the x86 toolchain because the operations cannot be done on x86 architecture. A workaround for this needs to be in place for `hypso-cli` to compile at head of the hsi-service branch. This issue was written when the head of the branch was at commit fdc65a0.

magnudan added bug Help wanted labels on Feb 21

DennisNTNU commented on Feb 21

The top level makefile includes `cmake -DARCH:STRING=x86` and `-DARCH:STRING=arm`, which look like preprocessor defines that will be passed onto the compiler, defining the constant `ARCH` to be either "arm" or "x86" (having a compiler flag `-DTEST=1` is the same as writing `#define TEST 1` in code).

This could then be used in code using some `#if` directive to switch between the desired SIMD code if ARCH is "arm" and some dummy x86 friendly code or an equivalent x86 implementation if ARCH is "x86".

But I don't know how to use compiler directives to do string comparisons, and through a few minutes of googling I didn't find any applicable solution.

magne-hov commented on Feb 21

Compilers will make some predefined defines based on what architecture they're compiling for (see f.ex https://sourceforge.net/p/predef/wiki/Architectures/).
I would use

```
#ifdef __arm__
   <youre arm specific code>
#endif /* __arm__ */
```

 **magnudan** commented on Feb 22

Defining out ARM-specific code and include partially solved the problem. When setting the flag `-mfpu=neon` in `target_compile_options()` in `CMakeLists.txt`, `hypso-sw` still does not compile. Error log from running `$ make`:

```
[ 28%] Building C object CMakeFiles/opu-services.dir/apps/opu_services.c.o
cc: error: unrecognized command line option '-mfpu=neon'
CMakeFiles/opu-services.dir/build.make:62: recipe for target 'CMakeFiles/opu-services.dir/
make[3]: *** [CMakeFiles/opu-services.dir/apps/opu_services.c.o] Error 1
make[3]: Leaving directory '/home/hypso/build/x86'
CMakeFiles/Makefile2:205: recipe for target 'CMakeFiles/opu-services.dir/all' failed
make[2]: *** [CMakeFiles/opu-services.dir/all] Error 2
make[2]: Leaving directory '/home/hypso/build/x86'
Makefile:94: recipe for target 'all' failed
make[1]: *** [all] Error 2
make[1]: Leaving directory '/home/hypso/build/x86'
Makefile:8: recipe for target 'all' failed
make: *** [all] Error 2
```

This was solved by moving the flag to `set(CMAKE_C_COMPILER arm-linux-gnueabihf-gcc -mfpu=neon)` in `cmake/toolchain-arm.cmake`.

This issue seems to be solved at commit 59b2e6f on branch hsi-service, and will be closed.

 **magnudan** closed this on Feb 22

  **magnudan** added the **completed** label on Feb 22

Assignees

No one—assign yourself

Labels

Help wanted

bug

completed

Projects

None yet

Milestone

No milestone

Linked pull requests

Successfully merging a pull request may close this issue.

None yet

3 participants

Pin issue

N hypso-sw Pull Request 111

NTNU-SmallSat-Lab / **hypso-sw** Private

Hsi service s wcompression #111

Edit | Open with

Merged **magnudan** merged 9 commits into `hsi-service` from `hsi-service-SWcompression` on Mar 2

Conversation 9 | Commits 9 | Checks 0 | Files changed 16 | +691 −31

drowzie commented on Mar 1

Integrates the software version of the compression algorithm.

drowzie added 2 commits on Feb 27

- `Added SW compression to hypso-sw, it compiles but it is untested. Fun…` `4a1e058`
  ...
- `Cleaning of functions for integration of software compression in HSI` `6e7cc97`

drowzie requested a review from **magnudan** on Mar 1

- `Restructured code and integrated into cubeDma.c` `ba92c8d`

magnudan commented on Mar 1

Why is the makefile different? Is there a reason?

magne-hov requested changes on Mar 1 — View changes

magne-hov left a comment

Nice work on this! I left some questions in comments.

It looks like you've overwritten our top-level `Makefile` with a CMake-generated makefile. We usually keep the CMake-generated makefile in `build/x86` or `build/arm` and don't include it in the repository. It doesn't make sense to keep both the x86 and arm makefile tracked in the repo when they are (1) automatically generated and (2) only slightly different for arm and x86. It would be nice if you could revert the Makefile changes.

`include/hsi/compression/predictor/predictor.h` — Show resolved

`src/hsi/compression/compressor.c` Outdated — Show resolved

`src/hsi/compression/compressor.c` Outdated — Show resolved

drowzie and others added 6 commits on Mar 1

- `Delete Makefile` Verified `1f0b5e7`
- `Minor changes according to PR` `f383247`
- `Merge branch 'hsi-service-SWcompression' of github.com:NTNU-SmallSat-…` `06d6278`
  ...
- `Merge branch 'hsi-service' into hsi-service-SWcompression` Verified `789d618`
- `error code` `5f5ec99`

Reviewers
magne-hov
magnudan

Assignees
No one—assign yourself

Labels
None yet

Projects
None yet

Milestone
No milestone

Linked issues
Successfully merging this pull request may close these issues.
None yet

3 participants

`Merge branch 'hsi-service-SWcompression' of github.com:NTNU-SmallSat-…` `3a9e0a8`

…

magnudan approved these changes on Mar 2 — View changes

magnudan left a comment

Looks good. Will re-add the make file after merge if needed

magnudan merged commit **435ea5d** into `hsi-service` on Mar 2

drowzie deleted the `hsi-service-SWcompression` branch on Mar 2 — Restore branch

O hypso-sw Issue 117

Edit | New issue

Jump to bottom

CubeDMA interfacing is not working correctly #117

Closed **magnudan** opened this issue on Mar 3 · 7 comments

Assignees	
Labels	HIGH PRIORITY bug points=20
Projects	SW kanban board

magnudan commented on Mar 3

The interfaces with CubeDMA are:

- Various registers for control, status, addresses etc.
- Dedicated memory for binned cube data, which the DMA reads from
- Dedicated memory for compressed cube data, which the DMA writes to

Current symptoms:

- The MM2S (memory map to stream) and S2MM (stream to memory map) transfers start, but do not always finish.
- The received length of the compressed cube is suspiciously large
- Some error bits in status register 0x04 are set

The first approaches to the investigation:

- Check address mapping for CubeDMA in Vivado.
- Test CubeDMA by running it on mock data.

Further down the road (if we do not get good results from the above approach):

- Build embedded Linux without CCSDS123 compression and test if `input == output`

magnudan added bug HIGH PRIORITY labels on Mar 3

magnudan self-assigned this on Mar 3

magnudan added this to **To do** in **SW kanban board** via automation on Mar 3

JoarGjersund commented on Mar 3

I know we have talked about this already, but just want to remind you that you should consider debugging it with ecc disabled in opu-system tcl to start with. Not sure, but I am afraid PS ECC will mess things up. There might be support for runtime disabling and enabling of PS ECC, but I have'nt looked into that yet.

👍 1

JoarGjersund moved this from **To do** to **Backlog** in **SW kanban board** on Mar 4

evelynlimore added the points=20 label on Mar 5

evelynlimore moved this from **Backlog** to **To do** in **SW kanban board** on Mar 5

drowzie commented on Mar 16

Got CubeDMA to work standalone where it moved some data from A to B. Problem seems like it was the address editor not having the defined addresses for it to properly work(From what i read the address editor defines what the AXI interconnect module routes(i think...)).

Second point is accessing the memory region for receive address causes a kernel panic. Can't properly test CCSDS compression when this occur, but it seems like CCSDS only accept the synthesized parameters of a cube to properly compress. Size: 107x500x720. A cube size of 1x1x1 is illegal and i assume CCSDS will stall and wait for new data to properly compute.

Third point, accessing the driver memory mapped data(mmap) of the cube locations needs to be flushed or cleared when accessing these.

drowzie closed this on Mar 16

SW kanban board automation moved this from **To do** to **Done** on Mar 16

JoarGjersund commented on Mar 16

should we have a pull-request open on opu-services before we close this?

JoarGjersund reopened this on Mar 16

SW kanban board automation moved this from **Done** to **In progress** on Mar 16

drowzie commented on Apr 2

@**JoarGjersund** Close this as b0fc231e793458d3a0db938eb310ee5bd55cc79f fixes this?

drowzie commented on Apr 2

Closing this as cubeDMA interface is now working thanks to opusystem memoryfix.
Hardware/Software compression will be verified by issue.

drowzie closed this on Apr 2

SW kanban board automation moved this from **In progress** to **Done** on Apr 2

evelynlimore moved this from **Done** to **In progress** in **SW kanban board** on Apr 6

magnudan commented on Apr 8

The Cube DMA runs consistently on my ZedBoard using the csp command `hsi dmatest` on hypso-sw commit 8e53bef. This commit contains all quick-fixes and changes to make it work in the opu-services framework. This is by no means finished work.

Before closing this issue, I want to tidy up the code and test it on the board in the lab.

I have discovered that Cube DMA needs things to be done in a very specific way to work correctly, and it was not very straight forward to integrate it. This is a summary of the procedure:

1. Initialize the memory regions and pointers
2. Set control registers to 0
3. Put data in the memory region cube DMA reads from
4. Clean cache (`ioctl(fd_send, 0)`)
5. Set address and dimension registers for cube DMA

6. Start both MM2S and S2MM transfers
    - Wait for both transfers to finish
7. Stop both transfers (set control registers to 0)
8. Flush cache (`ioctl(fd_receive, 1)`)

If for example step 5 was done before 3, the register containing the number of bytes received would not be correct. Other bugs I encountered by not following the exact procedure were that the transfers would never finish, or that data in and out did not match.

 **magnudan** reopened this on Apr 8

  **evelynlimore** moved this from **In progress** to **Review in progress** in **SW kanban board** on Apr 17

  **magnudan** mentioned this issue on Apr 17

Cube DMA now works #191 **Merged**

 **magnudan** commented on Apr 17

The interfacing with Cube DMA should now work at the head of the branch DMAtest at commit 21b0d1aa.

I've also made a test report documenting the implementation.

 **magnudan** closed this on May 5

 **SW kanban board** automation moved this from **Review in progress** to **Done** on May 5

Assignees 

 **magnudan**

Labels 

HIGH PRIORITY

bug

points=20

Projects

SW kanban board

Done

Milestone

No milestone

Linked pull requests

Successfully merging a pull request may close this issue.

None yet

4 participants

Pin issue

P hypso-sw Pull Request 119

NTNU-SmallSat-Lab / **hypso-sw** Private

HSI service pull request #119

Edit | Open with

Closed **DennisNTNU** wants to merge 75 commits into `pre-hsi-master` from `post-hsi-master`

Conversation 1 | Commits 75 | Checks 0 | Files changed 45 | +3,592 −124

DennisNTNU commented on Mar 4 • edited

Creating the pull-request on behalf of **@magnudan**.

During reviewing, you may skip the .ini, .uml files, as those do not contain code, as well as config.c, as that file has been approved previously.

DennisNTNU and others added 30 commits on Jan 7

Skeleton HSI service and hsi cli command	0fa3821
hsi service csp ping pong	71f721d
Simple get temperature command implemented	1233e41
Fix: using device ID instead of camera ID & dont need to init the cam…	2e93294
testing how to pull submodules for Jenkins compatibility	c254bcd
Merge branch 'hsi-service' of github.com:NTNU-SmallSat-Lab/hypso-sw i…	63720d0
updated all submodule urls	7282720
Submodules back to ssh URLs	94d032a
return non 0 on ACK timeout	57fd309
trying to write my first service	ce610aa
added missing semicolon and fixed syntax	cabbde9
Merging from master	b1c33d3
Skeletons for HSI services has been added	784edca
Updated to version 4.93	f22a65e
Functionality for de-/initialing implemented	62d7820
Service for configure seems to work. Segfault in capture. Debugging t…	256e6a3
hsi capture works. No segfaults	1821570
PlantUML script for HSI operation	fd8f6a2
Added timer entity and flash signals	5b6b92c
Split one diagram to 4. Added CubeDMA + CCSDS123	592c37d
Cleanup	a6c54ab
Modularized hsi service	fea17fb
In the process of adding binning. Encountered issue with arm_neon dep…	9cf706f
Still not compiling due to SIMD instructions. Pulling changes from ma…	1fd9483
Merge branch 'master' into hsi-service	bf1eaed
Added flag to compile neon SIMD commands	576a327

Reviewers
magnudan

Assignees
magnudan
DennisNTNU

Labels
None yet

Projects
SW kanban board
Done

Milestone
No milestone

Linked issues
Successfully merging this pull request may close these issues.
None yet

4 participants

Removed optimization flag for ARM de8ef94

Source code compiles for ARM without errors or warnings. Correctness … fdc65a0

Setting compile flag -mfpu=neon correctly 5ef6b98

defining out block of ARM code for x86 8594b59

23 hidden items
Load more…

DennisNTNU and others added 22 commits on Mar 1

Add utils/config.c also to opu-services 161a35b

Implemented pullrequest review sugestions 8fe1ca2

auto formatting e30b5c7

Merge pull request **#102** from NTNU-SmallSat-Lab/hsi-rgb-config Verified f555018

Delete Makefile Verified 1f0b5e7

add forgotten break; to remove warning 444a11c

Minor changes according to PR f383247

Merge branch 'hsi-service-SWcompression' of github.com:NTNU-SmallSat-… 06d6278

Merge branch 'hsi-service' into hsi-service-SWcompression Verified 789d618

Ready for merge from SW compression branch 7d603c8

error code 5f5ec99

Merge branch 'hsi-service-SWcompression' of github.com:NTNU-SmallSat-… 3a9e0a8

Merge pull request **#111** from NTNU-SmallSat-Lab/hsi-service-SWcompression Verified 435ea5d

Merge branch 'hsi-service' of github.com:NTNU-SmallSat-Lab/hypso-sw i… 3f323ed

Added makefile that was removed in merge d207849

Integrated SW compression aeb57a9

Merge branch 'hsi-service' edf1571

Update hsi_service.c Verified 11ba271

Update hsi_service.c Verified db49501

Merge pull request **#114** from NTNU-SmallSat-Lab/drowzie-patch-1 Verified 5b71e88

Update hsi_util.c Verified e2f49fc

Merge pull request **#116** from NTNU-SmallSat-Lab/drowzie-patch-1 Verified d1eca0a

DennisNTNU assigned **DennisNTNU** and **magnudan** on Mar 4

magnudan reviewed on Mar 4 View changes

magnudan left a comment

There are several things I would like to improve on in this code. The two main things are mentioned in issues #117 and #120. In addition, there are inconsistencies in naming schemes.

evelynlimore added this to **Backlog** in **SW kanban board** on Mar 5

evelynlimore moved this from **Backlog** to **To do** in **SW kanban board** on Mar 5

DennisNTNU closed this on Mar 6

SW kanban board automation moved this from **To do** to **Done** on Mar 6

DennisNTNU deleted the `post-hsi-master` branch on Mar 6 Restore branch

Q hypso-sw Issue 120

Edit | New issue

Jump to bottom

Modularize functionality in hsi source code #120

Closed **magnudan** opened this issue on Mar 4 · 0 comments

Assignees	
Labels	Enhancement HIGH PRIORITY points=13
Projects	SW kanban board

 **magnudan** commented on Mar 4

To be able to run meaningful tests in the HIL setup, the functionality in `hsi-service` shall be divided into meaningful modules with consistent interfaces. The current implementation of `hsi-service` was put together to have something for the initial MOBIP tests. This implementation has started the work on modularization, but work on this still remains. The interfaces at this point are sub-par.

The current suggestion for significant data structures are:

- A struct with HSI camera configuration: `hsi_config_s hsi_config`
- A struct with CubeDMA configuration: `cdma_config_s cdma_config`
- A struct with Compression parameters: `compr_config_s compr_config`
- A char pointer to the raw frame: `char* raw_frame`
- A uint16_t pointer to the area of interest (AoI) within the raw frame: `uint16_t* aoi`
- A uint16_t pointer to the binned cube (the address where CubeDMA reads from): `uint16_t binned_cube`
- A uint8_t pointer to the compressed cube (the address were CubeDMA writes to): `uint8_t compressed_cube`
- A char pointer to the frame queue array used by iDS' API: `char* frameQueue[]`
- An int array with frame ID's used by iDS' API: `int frameID[]`

The current suggestion for significant modules are:

- `hsi_camera_init(void)`
- `hsi_camera_configure(hsi_config, frameQueue, frameID)`
- `configure_binning(hsi config)`
- `hsi_capture(hsi_config, raw_frame)`
- `get_aoi(hsi_config, raw_frame, aoi)`
- `binn_aoi(hsi_config, aoi, binned_cube)`
- `hsi_camera_deinit(frameQueue, frameID)`
- `cubedma_start_transfers(void)`

magnudan added **Enhancement** **HIGH PRIORITY** labels on Mar 4

magnudan self-assigned this on Mar 4

magnudan added this to **To do** in **SW kanban board** via automation on Mar 4

JoarGjersund moved this from **To do** to **Backlog** in **SW kanban board** on Mar 4

evelynlimore added the points=13 label on Mar 5

evelynlimore moved this from **Backlog** to **To do** in **SW kanban board** on Mar 5

This was referenced on Mar 17

Modularized the HSI service and added documentation #143 Merged

Trying to merge again! #147 Merged

evelynlimore moved this from **To do** to **Done** in **SW kanban board** on Mar 19

magnudan closed this on Apr 27

Assignees

magnudan

Labels

Enhancement

HIGH PRIORITY

points=13

Projects

SW kanban board

Done

Milestone

No milestone

Linked pull requests

Successfully merging a pull request may close this issue.

None yet

2 participants

Pin issue

R hypso-sw Issue 126

Edit | New issue

Jump to bottom

Make a CSP command for capturing frames for debug/testing purposes #126

Closed **magnudan** opened this issue on Mar 5 · 2 comments

Assignees

Labels Enhancement In progress Testing points=3

Projects SW kanban board

magnudan commented on Mar 5

The command shall:

- Capture no more than 10 frames
- Store each frame as both raw data and in a human-readable format
- Store the entire cube in raw format
- Store the binned cube in raw format
- Store the compressed cube in raw format

magnudan added Enhancement In progress Testing labels on Mar 5

magnudan self-assigned this on Mar 5

magnudan added this to **To do** in **SW kanban board** via automation on Mar 5

evelynlimore moved this from **To do** to **Backlog** in **SW kanban board** on Mar 5

evelynlimore added the points=3 label on Mar 5

evelynlimore moved this from **Backlog** to **To do** in **SW kanban board** on Mar 5

evelynlimore moved this from **To do** to **In progress** in **SW kanban board** on Mar 5

magnudan commented on Mar 5

The functionality has been implemented in 3f1541d7, and is ready to be tested

magnudan mentioned this issue on Mar 5

Debug capture #128 Merged

magnudan commented on Mar 6

Active pull request at #128

evelynlimore moved this from **In progress** to **Done** in **SW kanban board** on Mar 19

magnudan closed this on May 1

Assignees

magnudan

Labels

Enhancement
In progress
Testing
points=3

Projects

SW kanban board

Done

Milestone

No milestone

Linked pull requests

Successfully merging a pull request may close this issue.

None yet

2 participants

Pin issue

S hypso-sw Pull Request 128

NTNU-SmallSat-Lab / **hypso-sw** Private

Debug capture #128

Edit | Open with

Merged **magnudan** merged 6 commits into `master` from `debug-capture` on Mar 10

Conversation 9 | Commits 6 | Checks 0 | Files changed 9 | +470 −255

magnudan commented on Mar 5

Implements functionality in issue #126

`debugging changes that saves images individually using IDS's library …` `db27d1a`

…

New changes since you last viewed View changes

`Adds CSP command for hsi debug capture` `3f1541d`

magnudan requested review from **magne-hov** and **DennisNTNU** on Mar 5

magnudan commented on Mar 5 Author

Disclaimer: The hsi command on x86 side will timeout before the command is executed on OPU

`Added binning flag for debug capture` Verified `39c4c0d`

magne-hov requested changes on Mar 6 View changes

`apps/opu_services.c` Show resolved

`src/cli/cli_hsi.c` Outdated Show resolved

`src/cli/cli_hsi.c` Outdated

```
140 +         numFrames = 10;
141 +     }
142 +
143 +     memcpy(&(packet->data[1]), &numFrames, 4);
```

magne-hov on Mar 6

Here you copy 4 bytes into packet->data[1], but you requested `csp_buffer_get(3)`, which theoretically wouldn't be big enough to hold 4 bytes. (practically, it works, becaues we know that our CSP buffers are large enough, but if we're being strict with error checking, then csp_buffer_get should be called with a size parameter of the size you expect to copy in to the buffer, and then check that we successfully got a buffer: `if (!packet) {/*error path*/}`)

magne-hov on Mar 6

It might be better to declare `char numFrames;` and `memcpy(&(packet->data[1]), &numFrames, sizeof(numFrames));`

Reply…

Resolve conversation

Reviewers
- sivertba ✓
- magne-hov ✓
- DennisNTNU ●

Assignees
No one—assign yourself

Labels
None yet

Projects
None yet

Milestone
No milestone

Linked issues
Successfully merging this pull request may close these issues.
None yet

4 participants

magnudan mentioned this pull request on Mar 6

Make a CSP command for capturing frames for debug/testing purposes Closed

#126

magne-hov approved these changes on Mar 6 View changes

magne-hov left a comment

Feel free to merge without addressing the comments :)

DennisNTNU added 3 commits on Mar 6

`Various changes` ··· `8e8f4b6`

`fix indentation in hsi_util.c` `ac48dd6`

`Reworked hsi ueye function call sequence for frame capture` ··· `0bdd0d6`

sivertba approved these changes on Mar 10 View changes

sivertba left a comment

What is missing for this to get pulled?

magnudan merged commit `255dbfe` into `master` on Mar 10 Revert

DennisNTNU deleted the `debug-capture` branch on Mar 10 Restore branch

T hypso-sw Issue 133

Edit | New issue

Jump to bottom

Consistency in returning error codes in hsi service #133

Closed **magnudan** opened this issue on Mar 12 · 2 comments

Projects SW kanban board

magnudan commented on Mar 12

Some functions return 1 indicating "success", while others return 1 indicating "error is present". Functions returning error codes shall return 0 indicating "success" and errors as integers.

evelynlimore commented on Mar 19

@**sivertba** please add to the proposed workflow...Functions returning error codes shall return 0 indicating "success" and errors as integers.
@**jlgarrett** please follow up on all the busy bees

The functions shall be documented properly, so we can figure it out and make sure that we're consistent in some way.

evelynlimore added this to **In progress** in **SW kanban board** on Mar 19

sivertba commented on Mar 19

The document describing the desired workflow is now updated to contain information about functions and proper use of error codes

sivertba closed this on Mar 24

SW kanban board automation moved this from **In progress** to **Done** on Mar 24

Assignees

No one—assign yourself

Labels

None yet

Projects

SW kanban board

Done

Milestone

No milestone

Linked pull requests

Successfully merging a pull request may close this issue.

None yet

3 participants

Pin issue

U hypso-sw Pull Request 143

NTNU-SmallSat-Lab / **hypso-sw** Private

Modularized the HSI service and added documentation #143

Edit | Open with

Merged **magnudan** merged 18 commits into `master` from `modularize` on Mar 18

Conversation 0 | Commits 18 | Checks 0 | Files changed 9 | +1,290 −879

magnudan commented on Mar 17

Summary:

- The source code has been un-spaghettied
- The source code has been divided into modules
  - Camera interfacing is moved to separate .c/.h files
- The modules have standardized interfaces
- The header files contain documentation for exposed functionality
- Interfacing with Cube DMA is not fixed

This PR is linked to issue #120

Once merged to master, a test similar to the MOBIP test should be run to verify correctnes

magnudan and others added 16 commits on Mar 5

Started changing interfaces	e3a4c63
Still refactoring. Work in progress	5fe3ddc
Still refactoring. Compiles, but not tested	bee7bc2
Moved camera functionality to separate .c and .h files	45e013f
Added function for hsi capture in camera module	2270f23
Done? refactoring. Ready to get changes from master	65cdf86
Merge branch 'master' into modularize	b55706d
Fix segfaults & valgrind warnings; using calloc instead of malloc	adf1345
Re-added debug capture, saving raw and binned cube, removed unnecessa… ...	1fb7329
Add camera AOI to debug capture, with corresponding size for frame bu… ...	de380c5
Using store_binned_cube() function, possibly fixed cdma init memory b… ...	b8321c8
Fixed return values	3589138
Found and fixed more errors in return values	a2b963b
Includes storing of individual frames in .png	45137fc
Added function to store single frames in .raw format	785846d
Documentation and cleanup	f2d696a

magnudan requested review from **DennisNTNU**, **drowzie**, **magne-hov** and **sivertba** on Mar 17

Merge branch 'master' into modularize Verified b305acb

sivertba approved these changes on Mar 18 View changes

Reviewers
- sivertba ✓
- drowzie
- magne-hov
- DennisNTNU

Assignees
No one—assign yourself

Labels
None yet

Projects
None yet

Milestone
No milestone

Linked issues
Successfully merging this pull request may close these issues.
None yet

3 participants

Merge branch 'master' into modularize Verified 6632e0b

magnudan merged commit **4615ba8** into `master` on Mar 18 Revert

This was referenced on Mar 18

Revert "Modularized the HSI service and added documentation" #146 Merged

Trying to merge again! #147 Merged

V hypso-sw Pull Request 146

NTNU-SmallSat-Lab / **hypso-sw** Private

Revert "Modularized the HSI service and added documentation" #146

Edit | Open with

Merged **DennisNTNU** merged 1 commit into `master` from `revert-143-modularize` on Mar 18

Conversation 0 | Commits 1 | Checks 0 | Files changed 9 | +879 −1,290

magnudan commented on Mar 18

Forgot to test after merging master into modularize...
Reverts #143

`Revert "Modularized the HSI service and added documentation"` Verified `185db61`

DennisNTNU approved these changes on Mar 18 — View changes

DennisNTNU merged commit `fc2a97b` into `master` on Mar 18 — Revert

DennisNTNU mentioned this pull request on Mar 18

Revert "Revert "Modularized the HSI service and added documentation"" #148 — Merged

DennisNTNU deleted the `revert-143-modularize` branch on Mar 18 — Restore branch

Reviewers
DennisNTNU ✓

Assignees
No one—assign yourself

Labels
None yet

Projects
None yet

Milestone
No milestone

Linked issues
Successfully merging this pull request may close these issues.
None yet

2 participants

W hypso-sw Pull Request 147

NTNU-SmallSat-Lab / **hypso-sw** Private

Trying to merge again! #147

Edit | Open with

Merged **DennisNTNU** merged 1 commit into `master` from `modularize` on Mar 18

Conversation 7 | Commits 1 | Checks 0 | Files changed 5 | +8 −8

magnudan commented on Mar 18

This PR fixes the issues with PR #143 that was merged too early in to master.

Text from original PR:

Summary:

- The source code has been un-spaghettied
- The source code has been divided into modules
  - Camera interfacing is moved to separate .c/.h files
- The modules have standardized interfaces
- The header files contain documentation for exposed functionality
- Interfacing with Cube DMA is not fixed

This PR is linked to issue #120

Once merged to master, a test similar to the MOBIP test should be run to verify correctnes

`took into account compression config changes from config.h in hsi ser…` `e8c5734`
...

magnudan requested review from **sivertba**, **magne-hov** and **DennisNTNU** on Mar 18

DennisNTNU commented on Mar 18

Why are there conflicts now.. and why don't all the other commits on the modularized branch show up? git isn't aware of the reversion?

magnudan commented on Mar 18 Author

I was wondering about the same. The conflicts are different from the ones we sorted out last week, though

DennisNTNU commented on Mar 18

It seems to me that at this point it would be better to revert also the revert pull request, in order to apply the missing commits to the master branch again, then we should be able to merge this pull request with the fix without conflicts.

magnudan commented on Mar 18 Author

Yes, that might be easier. It's still a mess, though

magnudan commented on Mar 18 Author

Do you want to take the lead?

Reviewers
DennisNTNU ✓
sivertba ●
magne-hov ●

Assignees
No one—assign yourself

Labels
None yet

Projects
None yet

Milestone
No milestone

Linked issues
Successfully merging this pull request may close these issues.
None yet

2 participants

magnudan closed this on Mar 18

DennisNTNU commented on Mar 18

sure

DennisNTNU reopened this on Mar 18

DennisNTNU commented on Mar 18

yay no conflicts

❤️ 1

DennisNTNU approved these changes on Mar 18 View changes

DennisNTNU merged commit `3270bdd` into `master` on Mar 18 Revert

DennisNTNU deleted the `modularize` branch on Mar 18 Restore branch

X hypso-sw Pull Request 148

NTNU-SmallSat-Lab / **hypso-sw** Private

Revert "Revert "Modularized the HSI service and added documentation"" #148

Edit | Open with

Merged **DennisNTNU** merged 1 commit into `master` from `revert-146-revert-143-modularize` on Mar 18

Conversation 0 | Commits 1 | Checks 0 | Files changed 9 | +1,290 −879

DennisNTNU commented on Mar 18

Reverts #146

`Revert "Revert "Modularized the HSI service and added documentation""` Verified `72d976e`

DennisNTNU merged commit `0ee08ff` into `master` on Mar 18 — Revert

magnudan approved these changes on Mar 18 — View changes

DennisNTNU deleted the `revert-146-revert-143-modularize` branch on Mar 18 — Restore branch

Reviewers
magnudan ✓

Assignees
No one—assign yourself

Labels
None yet

Projects
None yet

Milestone
No milestone

Linked issues
Successfully merging this pull request may close these issues.
None yet

2 participants

Y hypso-sw Pull Request 150

NTNU-SmallSat-Lab / **hypso-sw** Private

Various Fixes #150

Edit | Open with

Merged **JoarGjersund** merged 28 commits into `master` from `ft-fixes` on Mar 24

Conversation 13 | Commits 28 | Checks 0 | Files changed 28 | +438 −162

magne-hov commented on Mar 22 • edited

- updated the head of `libcsp` to use to the one with updated socketcan fix. Fixes #69.
- Made the download command check whether the file being downloaded has valid data. Failure to do so have lead to crashes.
- Removed some confusing constants

Update:

- I've added a commit with what I think is a fix for #103. Without the commit it's failing every 30% of tries, with the commit it hasn't failed me yet after a lot of tries.

Update 2:

- I've added a `opu download <remote path> <local path>` command, which eliminates the need to mess around with formatting, preparing or extracting files, it does all of that for you. With this you give a remote path and a local path and it does its best to transfer the file.

magne-hov added 8 commits on Mar 19

`minor helptext change`	`030e820`
`updated head of libcsp`	`354275f`
`pass correct size of entry ID to ft check fn`	`5b4577a`
`rename HYPSO_MTU to HYPSO_CSP_MTU`	`f916e22`
`define constant for max entries in check request`	`ce82f51`
`rework client_check to take conn and return result`	`0b558a8`
`add ft check function for a whole file`	`a372596`
`remove unused constants for buffer sizes`	`a66808a`

magne-hov requested review from **DennisNTNU** and **rogerbirkeland** on Mar 22

magne-hov force-pushed the `ft-fixes` branch from **`94f63f8`** to **`0982558`** on Mar 22

magne-hov added 5 commits on Mar 22

`make ft check bitmap function return count`	`b723035`
`check a remote file for corruptness before downloading`	`4635bc3`
`change order on ft download checks`	`075ae79`
`rename shell remote timeout set command`	`c2ff594`
`add diagnostics and fix overflow in shell service`	`8fb14f6`

magne-hov force-pushed the `ft-fixes` branch from **`96a048b`** to **`8fb14f6`** on Mar 22

magne-hov changed the title ~~**Various FT Fixes**~~ **Various Fixes** on Mar 22

Reviewers
- magnudan ✓
- rogerbirkeland ●
- DennisNTNU ●

Assignees
No one—assign yourself

Labels
None yet

Projects
SW kanban board
Done

Milestone
No milestone

Linked issues
Successfully merging this pull request may close these issues.
1Mbit CAN only almost doubles tra…

4 participants

magne-hov added 12 commits on Mar 22

rename OPU CSP address define	07ae896
add cli cmd function pointer typedef	d0e9a80
define file ID 0 as the one to use automatically ···	8a7ea58
fix incorrect ft download file prototype	3d4fef3
add macro for calling cli command functions	da439bf
add formatted file suffixes	4754bc0
minor formatting change	fa17232
add opu download command which combines ft commands	ed051ae
make ft check local do whole file	b326c50
make fs check return large int	190c4ac
pull local file checking logic into function ···	5dbc9e3
check file when downloading	107a87c

magne-hov requested a review from **magnudan** on Mar 22

rogerbirkeland commented on Mar 22

Really like your "do everything" - download! Looking forward to test it

magne-hov commented on Mar 22 Author

For testing: build.zip

add more info to opu download command e650e9e

rogerbirkeland commented on Mar 22

Tested:
Uploaded the arm-binary that I got from Magne, and ran it on LidSat. The transfer-rate is much quicker now! I also tested with the current opu-services that is on the SD-card, then the transfer took much longer (09:45 for the same file, a 4.9 time improvement!)

```
(hypso) opu download /media/sd/bitstream.bit
Usage: opu download <remote path> <local path>
(hypso) opu download /media/sd/bitstream.bit bitstream.bit.dl
Registering remote path /media/sd/bitstream.bit as file ID 0.
Using a default maximum download duration of 1800.
===  Source File Info  ===
file_id:       0
status:        SUCCESS (0)
file_status:   SUCCESS (0)
last_entry_id: 0
total_entries: 0
cell_sz:       0
used_cells:    0
max_cells:     0
sector_qty:    1
sector_sz:     5980026
file_type:     UNFORMATTED
file_name:     d/bitstream.bit
[       1-24917   ]: 100% [========================]
===   Download Summary   ===
Entries transferred:    24917
Entries requested:      24917
```

```
Entries retransmitted:  0
Packet loss:            0.00 %
Duration:               00:01:59
Entry size:             240B
Bytes transferred:      5980080 B
Effective datarate:     50.1322 [KBps]
                        401.0574 [Kbps]
Start entry ID: 1
Last entry ID:  24917
Total entries:  24917
Extracting downloded file from bitstream.bit.dl.download-format to bitstream.bit.dl.
Removing formatted file bitstream.bit.dl.download-format.
(hypso) ls
total 13900
-rw-r--r-- 1 root  root  5980026 mars  22 18:54 bitstream.bit.dl
```

opu download /media/sd/bitstream.bit bitstream.bit.dl

rogerbirkeland mentioned this pull request on Mar 22

1Mbit CAN only almost doubles transfer speed compared to 125Kbit #69 Closed

`add info to buffer helptext` `3c5043d`

magne-hov commented on Mar 22 Author

Good results, **@rogerbirkeland** ! I will wait until either **@DennisNTNU** or **@magnudan** has had a chance to look at the PR before merging.

`keep formatted file when calling opu download` `12a950e`

magnudan approved these changes on Mar 23 View changes

magnudan left a comment

Great work!

I skimmed through the changes, and they look good. Nothing to add.

I tested this on the ZedBoard in the HIL setup. The enclosed file did not run there:

```
root@avnet-digilent-zedboard-2019_1:~# ./opu-services-test
-sh: ./opu-services-test: cannot execute binary file: Exec format error
```

so I compiled a new executable, which worked (after updating libcsp).

I used `hsi debugcapture 1` to capture a `.png` image to test the download command with.

From the cli side:

```
(hypso) hsi debugcapture 1
--> Sending HSI Debug Capture Request
<-- Received capture request of 1 frames
(hypso) opu download /home/root/frame_0.png ./frame_0.png
Registering remote path /home/root/frame_0.png as file ID 0.
Using a default maximum download duration of 1800.
===  Source File Info  ===
file_id:       0
status:        SUCCESS (0)
file_status:   SUCCESS (0)
last_entry_id: 0
total_entries: 0
cell_sz:       0
used_cells:    0
max_cells:     0
sector_qty:    1
sector_sz:     1147873
file_type:     UNFORMATTED
file_name:     oot/frame_0.png
```

```
[       1-4783    ]: 100% [=========================]
===   Download Summary   ===
Entries transferred:    4783
Entries requested:      4783
Entries retransmitted:  0
Packet loss:            0.00 %
Duration:               00:00:21
Entry size:             240B
Bytes transferred:      1147920 B
Effective datarate:     53.9083 [KBps]
                        431.2663 [Kbps]
Start entry ID: 1
Last entry ID:  4783
Total entries:  4783
Extracting downloded file from ./frame_0.png.download-format to ./frame_0.png.
Removing formatted file ./frame_0.png.download-format.
(hypso)
hypso@hypso-HP-Compaq-8100-Elite-SFF-PC:~/Magnus$ ls
frame_0.png  hypso-cli-M  opu-services-blabla  opu-services-M
```

I then copied the frame directly from the ZedBoard and called it `frame_0.png`, and the frame from the HIL workstation and called it `csp.png`.

`frame_.png`:

`csp.png`:

I'd say this works :)

rogerbirkeland commented on Mar 23

Cool!
I will work more today too. But looks very promising.
The file included was for x86, that's why it did not work. (Had the same

problem.)

man. 23. mar. 2020 kl. 10:41 skrev Magnus Danielsen <
notifications@github.com>:
...

rogerbirkeland commented on Mar 23 • edited

I fucked up; I tested with WRONG opu-services running... Edits below is with correct software versions

`ft upload` successfully uploads the a formatted file very quick!

`ft check` still has strange exit codes:

```
File is complete.
Upload finished.
(hypso) ft check
cli_ft_check: Invalid input.
Usage: ft check <ALL|PRESENCE|INTEGRITY> <csp address> <file id> <first entry id>
<last entry id>
(hypso) ft check ALL 12 3 0 1900
Entries 0-1900:
X.....................................................................................

`ft check` failed: 1 (Operation not permitted)
(hypso) ft check 12 3 1900 1991
cli_ft_check: Invalid input.
Usage: ft check <ALL|PRESENCE|INTEGRITY> <csp address> <file id> <first entry id>
<last entry id>
(hypso) ft check ALL 12 3 1900 1991
Entries 1900-1991:
.....................................................................................

(hypso)
```

ft extract names the file wrongly:
From `ls /media/sd`: `opu2-armell_header`

Command executed: `ft extract 12 3 /media/sd/opu2-arm`

and then did the same 20 times:

`ft extract 12 3 /media/sd/opu2-arm<n>` where is [1,20]

```
opu2-arm10service_task
opu2-arm11s_cube_software
opu2-arm12erTMCloneTable
opu2-arm13C_2.4
opu2-arm14ibc_start_main@@GLIBC_2.4
opu2-arm15t_service_format
opu2-arm16try
opu2-arm17er_clone
opu2-arm18_array
opu2-arm19
opu2-arm1dma_init
opu2-arm20
opu2-arm2dex@@GLIBC_2.4
opu2-arm3
opu2-arm4_post@@GLIBC_2.4
opu2-arm5d
opu2-arm62.4
opu2-arm7ubedma_PrintRegister
opu2-arm8ce_send_range
opu2-arm9ureStatus
opu2-armell_header
```

Look into the extract thing, then we can approve.

sivertba added this to **In progress** in **SW kanban board** on Mar 24

JoarGjersund commented on Mar 24

here is my result. I am still not able to do a firmware upgrade by replacing the image.ub on the SD-Card. The uploading time is also the same, ca. 1 hour for 80MB

```
Init can interface can0
Hostname:  smallsatlab
Model:     hypso-cli
Revision:  Mar 24 2020
Type list to see commands.
Type help <command> for specific help.
(hypso) ft list 12
(hypso) ft register 12 image.ub image.ub.fmt 1 240
Invalid input.
Usage: ft register <csp address> <pathname> <file id>
(hypso) ft register 12 image.ub image.ub.fmt 1
Invalid input.
Usage: ft register <csp address> <pathname> <file id>
(hypso) ft register 12 image.ub 1
(hypso) ft prepare local image image.ub.fmt 1 240
Failed to open image
`ft prepare local` failed: 5 (Input/output error)
(hypso) ft prepare local image.ub image.ub.fmt 1 240
file_name:      image.ub.fmt
file_type:      STATIC
file_id:        1
entry_sz:       240
max_entries:    338755
first_entry_id: 1
total_entries:  338755
(hypso) ft format 12 1 240 338753
File ID: 1, status: 0
(hypso) ft format 12 1 240 338755
File ID: 1, status: 0
(hypso) ft upload file 12 image.ub.fmt 1
Uploading 1 missing ranges:
[       1-338755  ]: 100% [========================]
File is complete.
Upload finished.
(hypso) ft extract 12 1 /media/sd/image.ub
/home/hypso/src/ft/ft_client.c:722:ft_client_extract: response length: 0, expected 3.
cli_ft_extract failed with ret: -71
`ft extract` failed: 71 (Protocol error)
```

magnudan commented on Mar 24

I'm having similiar problems as Joar, I think.

I tried to upload the file `image.ub` from the HIL workstation to the ZedBoard using CSP. These are the commands I used:

```
(hypso) ft register 12 /mnt/boot/image.ub.fmt 1

(hypso) ft list 12
+-------------------------------------------------------------------------------
------------------+
| File     | ID | Last     | Total    | Cell     | Cells    | Max      | Sector   |
Sector   | File     |
| name     |    | Entry    | Entries  | Size     | Used     | Cells    | Quantity |
Size     | Type     |
*-------------------------------------------------------------------------------
------------------*
| image.u- | 1  | 0        | 0        | 0        | 0        | 0        | 0        | 0
| NOFILE   |
| b.fmt    |    |          |          |          |          |          |          |
|          |
+-------------------------------------------------------------------------------
------------------+

(hypso) ft prepare local image.ub image.ub.fmt 1 240
file_name:      image.ub.fmt
file_type:      STATIC
file_id:        1
entry_sz:       240
```

```
max_entries:    338031
first_entry_id: 1
total_entries:  338031

(hypso) ft format 12 1 240 338031
/home/hypso/src/ft/ft_client.c:332:ft_client_format: Failed to get format response.
`ft format` exited with return value: -110

(hypso) list 12
+-------------------------------------------------------------------------------
------------------+
| File     | ID | Last     | Total    | Cell     | Cells    | Max      | Sector   |
Sector   | File     |
| name     |    | Entry    | Entries  | Size     | Used     | Cells    | Quantity |
Size     | Type     |
*-------------------------------------------------------------------------------
------------------*
| ot/imag- | 1  | 338031   | 338031   | 246      | 338031   | 338031   | 1        |
82669568 | STATIC   |
| e.ub.fmt |    |          |          |          |          |          |          |
|          |
+-------------------------------------------------------------------------------
------------------+
```

Output on the ZedBoard:

```
/home/hypso/src/ft/ft_service.c:498:ft_service_format: Received format request for
338031 entries of size 240 for file ID 1.
```

I then tried to download the file `image.ub` from the ZedBoard using `opu download` after uploading it over SCP:

```
(hypso) opu download image.ub image_downloaded.ub
Registering remote path image.ub as file ID 0.
Using a default maximum download duration of 1800.
===  Source File Info  ===
file_id:       0
status:        SUCCESS (0)
file_status:   SUCCESS (0)
last_entry_id: 0
total_entries: 0
cell_sz:       0
used_cells:    0
max_cells:     0
sector_qty:    1
sector_sz:     81127348
file_type:     UNFORMATTED
file_name:     image.ub
[       1-338031  ]: 100% [=========================]
===   Download Summary   ===
Entries transferred:    338031
Entries requested:      338031
Entries retransmitted:  0
Packet loss:            0.00 %
Duration:               00:25:05
Entry size:             240B
Bytes transferred:      81127440 B
Effective datarate:     53.8723 [KBps]
                        430.9781 [Kbps]
Start entry ID: 1
Last entry ID:  338031
Total entries:  338031
Extracting downloded file from image_downloaded.ub.download-format to
image_downloaded.ub.
Removing formatted file image_downloaded.ub.download-format.
```

magnudan commented on Mar 24

And I'm not experiencing faster upload speed. Uploading a bitstream file of approx. 4MB took well over an hour with the following commands:

```
(hypso) ft register 12 /mnt/boot/bitstream.bit.fmt 1

(hypso) ft prepare local bitstream.bit bitstream.bit.fmt 1 240

(hypso) ft format 12 1 240 16857
```

```
(hypso) ft upload file 12 bitstream.bit.fmt 1
```

JoarGjersund commented on Mar 24 • edited

@magnudan Are you using the LidSat? I think Roger is working on it now as well, might be the source of some of your trouble.. As I wrote, uploading 80MB took a little bit less than 1 hour to complete, so uploading 4 MB really shouln't take more than a few minutes.

JoarGjersund commented on Mar 24 • edited

After running ft extract I can see two files of image.ub. the one called image.ub-e appears to have the correct filesize, but I am not able to copy it over to the sd card. (probably because of some permission issues.

```
(OPU) ls -l
drwxr-xr-x    2 root     root          1420 Mar 24 10:08 bin
drwxr-xr-x    2 root     root            40 Mar 24 09:50 boot
drwxr-xr-x    2 root     root            40 Mar 24 10:09 config
drwxr-xr-x    9 root     root          2880 Mar 24 10:09 dev
drwxr-xr-x   24 1000     ueye          1260 Mar 24 10:09 etc
-rw-r--r--    1 root     root          4500 Mar 24 10:09 file_store.dat
drwxr-xr-x    3 1000     ueye            60 Mar 13 13:02 home
-rw-r--r--    1 root     root      83333762 Mar 24 11:04 image.ub
-rw-r--r--    1 root     root      81301108 Mar 24 11:05 image.ub-e
lrwxrwxrwx    1 root     root            10 Mar 24 10:09 init -> /sbin/init
drwxr-xr-x    6 1000     ueye          1260 Mar 24 10:09 lib
drwxr-xr-x    4 root     root            80 Mar 24 10:09 media
drwxr-xr-x    2 root     root            40 Mar 24 09:50 mnt
-rw-r--r--    1 root     root        251936 Mar 24 10:09 opu_log
dr-xr-xr-x   76 root     root             0 Jan  1  1970 proc
drwxr-xr-x    2 root     root            40 Mar 24 10:09 rgb-images
drwx------    2 root     root            40 Mar 24 09:54 root
drwxr-xr-x    6 root     root           260 Mar 24 10:09 run
drwxr-xr-x    2 root     root          1380 Mar 24 10:08 sbin
dr-xr-xr-x   12 root     root             0 Jan  1  1970 sys
lrwxrwxrwx    1 root     root             8 Mar 24 10:08 tmp -> /var/tmp
drwxr-xr-x   10 1000     ueye           200 Mar 13 13:02 usr
drwxr-xr-x    8 root     root           240 May 22  2019 var
(OPU) cp -fr image.ub-e /media/sd/image.ub
cp: can't stat 'image.ub-e': No such file or directory
```

EDIT1 : wasn't a permission issue after all. It was just an invisible ASCII character appended to the filename..

```
root@092a743-primary:/# ls image.ub-e
ls: image.ub-e: No such file or directory
root@092a743-primary:/# ls image.ub-e\360
image.ub-e
root@092a743-primary:/# ls image.ub-e\360  | hexdump -C
00000000  69 6d 61 67 65 2e 75 62  2d 65 0a                 |image.ub-e.|
0000000b
```

EDIT2: The file upload seems to work, but the bugs mentioned above should be fixed. This is how I was able to perform a firmware upgrade:

```
(hypso) ft list 12
(hypso) ft register 12 image.ub 1
(hypso) ft prepare local image.ub image.ub.fmt 1 240
file_name:      image.ub.fmt
file_type:      STATIC
file_id:        1
entry_sz:       240
max_entries:    338755
first_entry_id: 1
total_entries:  338755
(hypso) ft format 12 1 240 338755
File ID: 1, status: 0
(hypso) ft upload file 12 image.ub.fmt 1
Uploading 1 missing ranges:
[       1-338755  ]: 100% [========================]
```

```
File is complete.
Upload finished.
(hypso) ft extract 12 1 image.ub
/home/hypso/src/ft/ft_client.c:722:ft_client_extract: response length: 0, expected 3.
cli_ft_extract failed with ret: -71
`ft extract` failed: 71 (Protocol error)
(hypso) shell remote 12 10000000
Enter "exit", "quit" or "q" to exit remote shell.
(OPU) ls image.ub-e
```

(I then had to press tab for autocomplete) getting the following result

```
(OPU) ls image.ub-e\360
```

I then had remove ls prepend cp -fr and append /media/sd/image.ub, like this:

```
(OPU) cp -fr image.ub-e\360 /media/sd/image.ub
```

then I did a reboot

```
(OPU) reboot
```

The new image (version 092a743) was confirmed by printing the hostname

```
(hypso) shell remote 12 10000
Enter "exit", "quit" or "q" to exit remote shell.
(OPU) hostname
092a743-primary
```

🎉 2 🚀 1

magnudan commented on Mar 24

I was working on the ZedBoard in the HIL setup, there should not have been any traffic there

JoarGjersund merged commit `a0a876a` into `master` on Mar 24 Revert

SW kanban board automation moved this from **In progress** to **Done** on Mar 24

This was referenced on Mar 25

ft extract can still cause segfault #164 Closed

Error uploading, and extracting uploaded formatted file #99 Closed

Z hypso-sw Issue 169

Edit | New issue

Jump to bottom

Debug-capture saves the binned cube with 0 bytes #169

Closed **rogerbirkeland** opened this issue on Mar 26 · 2 comments

Labels	bug points=5
Projects	SW kanban board

rogerbirkeland commented on Mar 26

`hsi debugcapture` creates a binned_cube.raw-file, but it is empty.

evelynlimore added this to **To do** in **SW kanban board** on Mar 27

evelynlimore moved this from **To do** to **In progress** in **SW kanban board** on Mar 27

magnudan commented on Mar 27 • edited

Does `hsi capture` create a non-empty `binned_cube.raw` ?

I don't want to push what I tested yet, as I don't know exactly why it crashed my ZedBoard, but I have discovered some things that I hope will help to solve this issue.

I see that the function `hsi_debug_capture_and_binn_cube()` in `src/hsi/hsi_camera.c` is missing the initialization of binning:

```
int hsi_debug_capture_and_binn_cube(struct hsi_config_s* hsi_config,
                                    uint16_t* binned_cube,
                                    unsigned int* binned_bytes)
{
    int hsi_err = 0;
    hsi_err = hsi_camera_init();
    if (hsi_err)
    {
        printf("    Unable to initialize camera\n");
        return 1;
    }

    hsi_err = hsi_camera_configure(hsi_config);
    if (hsi_err)
    {
```

```
        printf("    Unable to configure camera\n");
        return 2;
    }

    char** frame_buffers = calloc(FRAME_BUFFER_SIZE, sizeof(*frame_buffers));
    int* frame_buffer_IDs = calloc(FRAME_BUFFER_SIZE, sizeof(*frame_buffer_IDs));

    binning_init(hsi_config); // This line is missing!!!

    hsi_err = hsi_camera_start_capture(hsi_config, frame_buffers, frame_buffer_IDs);
    if (hsi_err)
    {
        printf("    Unable to start capture\n");
        free(frame_buffers);
        free(frame_buffer_IDs);
        return 3;
    }
```

I have also noticed that in `binn_single_frame()` in `src/hsi/hsi_utils.c`, when incrementing the value of `binned_bytes`, this might increment the pointer instead of the value. To test this, change

```
pthread_mutex_lock(&lock);
{
     binned_bytes += 2;
}
pthread_mutex_unlock(&lock);
```

to

```
pthread_mutex_lock(&lock);
{
     *binned_bytes = *binned_bytes + 2;
}
pthread_mutex_unlock(&lock);
```

and

```
pthread_mutex_lock(&lock);
{
     binned_bytes ++;
}
pthread_mutex_unlock(&lock);
```

to

```
pthread_mutex_lock(&lock);
{
     *binned_bytes = *binned_bytes + 1;
}
pthread_mutex_unlock(&lock);
```

magnudan commented on Apr 14

Another thing: The binned data is 16-bit, so the `binned_bytes` shall be incremented by 4 and 2, not 2 and 1, as a byte is 8-bit

evelynlimore added bug points=5 labels on Apr 16

DennisNTNU mentioned this issue on Apr 23

Binning and other fixes #202 Merged

sivertba moved this from **In progress** to **Review in progress** in **SW kanban board** on Apr 30

DennisNTNU closed this on May 1

SW kanban board automation moved this from **Review in progress** to **Done** on May 1

Assignees

No one—assign yourself

Labels

bug

points=5

Projects

SW kanban board

Done

Milestone

No milestone

Linked pull requests

Successfully merging a pull request may close this issue.

None yet

4 participants

Pin issue

AA hypso-sw Issue 188

Edit | New issue

Jump to bottom

Integrate and test implementation of timestamping #188

Closed **magnudan** opened this issue on Apr 15 · 2 comments

Assignees	
Labels	points=13
Projects	SW kanban board

magnudan commented on Apr 15 • edited

@jonakor and **@simennett** have made a module for timestamping HSI frames using PPS and flash signals in `src/utils/timestamp.c` on the timestamp branch. This needs to be integrated with the existing HSI service.

There are (as I see this) two ways to test this:

1. Using a signal generator to emulate BOTH PPS and flash signals
2. Using a signal generator to emulate the PPS, and get the flash signal from the HSI. This requires a new cable for the HSI to be made.

2

evelynlimore added this to **Backlog** in **SW kanban board** on Apr 16

evelynlimore added the points=13 label on Apr 16

evelynlimore moved this from **Backlog** to **To do** in **SW kanban board** on Apr 16

magnudan self-assigned this on Apr 17

sivertba commented on Apr 30

Split integrate and test?
Need cable from Amund

magnudan commented on May 1

The title might not need the word "test", as tests must be performed and documented for the functionality to be merged to master?

This was referenced on May 8

Configure flash return from HSI camera #220 Closed

Flash is enabled #236 Merged

sivertba moved this from **To do** to **In progress** in **SW kanban board** on May 14

magnudan mentioned this issue on May 15

Timestamping of HSI frames is live! #240 Merged

evelynlimore moved this from **In progress** to **Review in progress** in **SW kanban board** on May 15

magnudan closed this in #240 on May 15

SW kanban board automation moved this from **Review in progress** to **Done** on May 15

Assignees

magnudan

Labels

points=13

Projects

SW kanban board

Done

Milestone

No milestone

Linked pull requests

Successfully merging a pull request may close this issue.

Timestamping of HSI frames is live!

3 participants

Pin issue

AB hypso-sw Pull Request 191

NTNU-SmallSat-Lab / **hypso-sw** Private

Cube DMA now works #191

Edit | Open with

Merged **magnudan** merged 13 commits into `master` from `DMAtest` on May 5

Conversation 30 | Commits 13 | Checks 0 | Files changed 10 | +227 −309

magnudan commented on Apr 17

The cli now has a command `hsi dmatest` that runs a test of the Cube DMA on target hardware. The target hardware (either ZedBoard or PicoBOB) must have a bitstream with only the CubeDMA looped back to itself, without ccsds 123 to be verified. If ccsds 123 is in the bitstream, there will be a print describing how much the data has been compressed, but there is no verification of compression here.

This PR solves issue #117

@DennisNTNU: I have some changes here w.r.t. binning, as well. Feel free to discard them when you merge your fix :)

I'm attaching the bitstream file without ccsds 123 for ZedBoard. This might also work for PicoBOB, but I'm not sure... If it doesn't work, I can make a bitstream for PicoBOB if anyone needs it.

magnudan added 6 commits on Apr 8

Works on my ZedBoard without CCSDS 123	8e53bef
Refactored cube DMA code. Works on local ZedBoard	d7296e4
Registers as enum instead of defines	077bc44
Typo	78ca3f1
Removed unused includes	c16b76b
Formatting and comment	21b0d1a

magnudan requested review from **drowzie**, **JoarGjersund**, **magne-hov** and **DennisNTNU** on Apr 17

magnudan commented on Apr 17 — Author

The reason I have changed code in binning and mock_lib is that the first test I wrote generated mock frames that I binned and sent to Cube DMA.

drowzie approved these changes on Apr 17 — View changes

drowzie left a comment

Ship it.

magne-hov commented on Apr 17

This is totes cool, will review this weekend 💯💯💯

magnudan mentioned this pull request on Apr 17

Verify compression script #192 Merged

Reviewers
- magne-hov
- drowzie ✓
- sivertba ✓
- rogerbirkeland ✓
- JoarGjersund ✓
- DennisNTNU

Assignees
- DennisNTNU

Labels
None yet

Projects
SW kanban board — Done

Milestone
No milestone

Linked issues
Successfully merging this pull request may close these issues.
None yet

7 participants

magnudan commented on Apr 17 • edited ▾ Author

As mentioned in PR #192 (cross-referencing, yay!), I get segfaults when using `store_compressed_cube()` and `store_binned_cube()` in the test code. I suspect this has something to do with inconsistency in counting bits, bytes, chars, shorts etc. Will make an issue on this. (Edit: Issue #197)

 **magnudan** mentioned this pull request on Apr 17

hsi capture hangs after capture #195 Closed

rogerbirkeland commented on Apr 21

Is this ready or not to be merged so we can get working hsi-capture?

sivertba requested changes on Apr 22 View changes

sivertba left a comment • edited ▾

So here is the results I got when I tried to test this branch on the LidSat:
Top terminal being the OPU, and bottom as HYPSO-CLI

```
                                                  hypso@hypso-HP-Compaq-8100-E
File  Edit  View  Search  Terminal  Tabs  Help
                  sivertba@workpc: ~/hypso-sw/scripts/mech_scripts
Model:    opu-services
Revision: Apr 22 2020
[Started] CSP Services              Thread ID: 2265
[Started] File Transfer Service     Thread ID: 2266
[Started] TFTP Service              Thread ID: 2267
[Started] CLI Service               Thread ID: 2268
[Started] RGB Camera Service        Thread ID: 2269
[Started] CLAW-1 Payload Service    Thread ID: 2272
Received hsi camera temperature poll request
    Getting Device Info of camera with device ID: 1001 ...
    Firmware Version: 0
    Temperature: 51.3 | 51.3
    Link speed [Mb]: 1000
    Comport offset?: 65535
    Device ID: 1001
Received DMA Test request
```

```
                                                  hypso@hypso-HP-Compaq-8100-E
File  Edit  View  Search  Terminal  Help
Description: CLAW-1 specific commands.

Sub-commands:
hsi capture                 - Captures given number of HSI frames.
hsi geocapture              - Captures some HSI frames and georef.
hsi gettemp                 - Poll the HSI camera temperature.
hsi debugcapture            - Captures and saves up to 10 frames
hsi dmatest                 - Tests the CubeDMA module.

(hypso) hsi gettemp
--> Sending HSI Camera Temperature Request
<-- Camera Temperature is 51.3
(hypso) hsi dmatest
--> Sending Request to test DMA
    ACK Timeout
`hsi dmatest` failed: 1 (Operation not permitted)
(hypso)
```

~~There is a high risk that I've done something weird, but please have a closer look at it :)~~

I am not sure where to put the bitstream

 **DennisNTNU** mentioned this pull request on Apr 23

Binning and other fixes #202 **Merged**

magne-hov reviewed on Apr 25 — View changes

`src/hsi/hsi_util.c` Outdated — Show resolved

magne-hov reviewed on Apr 25 — View changes

`src/mock_lib/mock_lib.c` — Show resolved

magne-hov commented on Apr 25

The code looks alright to me, although I can't really make out how it works. Could I help in testing? Would need some instructions in that case

🎉 1 ❤️ 1

magnudan commented on Apr 29 • edited — Author

I have started testing on the Picozed on BoBv2 in the LidSat setup. After updating the boot files to the newest version (commit 4457085 on the opu-system repository), I am able to interface with the Cube DMA, but the transfers time out. This is not an issue I have on ZedBoard, where the transfers complete consistently every time.

This is the printout from `opu-services`:

```
Received DMA Test request
    Generating data
    Data generated
        Starting transfer
        Contents of MM2S_CTRL_REG      :0x00000001
        Contents of MM2S_STAT_REG      :0x00000000
        Contents of MM2S_ADDR_REG      :0x30000000
        Contents of MM2S_CUBE_DIM_REG  :0x6b1f42d0
        Contents of MM2S_BLOCK_DIM_REG :0x00000000
        Contents of MM2S_ROW_DIM_REG   :0x00012cf0
        Contents of S2MM_CTRL_REG      :0x00000001
        Contents of S2MM_STAT_REG      :0x00000000
        Contents of S2MM_ADDR_REG      :0x38000000
        Contents of S2MM_LEN_REG       :0x00000000
    ERROR: MM2S timeout
    ERROR: S2MM timeout
    CubeDMA transfer not successfull.
        Contents of MM2S_CTRL_REG      :0x00000001
        Contents of MM2S_STAT_REG      :0x00000014
        Contents of MM2S_ADDR_REG      :0x30000000
        Contents of MM2S_CUBE_DIM_REG  :0x6b1f42d0
        Contents of MM2S_BLOCK_DIM_REG :0x00000000
        Contents of MM2S_ROW_DIM_REG   :0x00012cf0
        Contents of S2MM_CTRL_REG      :0x00000001
        Contents of S2MM_STAT_REG      :0x00000000
        Contents of S2MM_ADDR_REG      :0x38000000
        Contents of S2MM_LEN_REG       :0x00000000
```

This is the bitstream I'm using for zynq7030.

The bitstream has been made using this script

sivertba added this to **Review in progress** in **SW kanban board** on Apr 30

magnudan commented on Apr 30 • edited — Author

The solution for Cube DMA on PicoBOB will be merged to master on the repository opu-system with PR #91. Turns out Vivado automatically assigned a too small memory range for the Cube DMA. This will now be set manually in the .tcl scripts.

Both PicoZeds in the LidSat setup in the lab now have correct boot files to test this implementation. In the `/media/sd` directory there are three bitstream files:

- `bitstream.bit` : The bitstream that is loaded to the FPGA on boot
- `bitstream_with_compression.bit` : A bitstream that compresses data using ccsds 123 in FPGA
- `bitstream_without_compression.bit` : A bitstream that loops data directly back to itself

To change bitstream, copy the desired bitstream to `bitstream.bit` :

```
# cp <desired bitstream> bitstream.bit
# reboot
```

On the PicoZed with IP .50 there is only one `opu-services` , while on the one with IP .42 there are several. Use the one named `opu-services-DMAtest` .

If you don't have the opportunity to compile `hypso-cli` from this branch yourself, there is one in `/home/magnus/cli` on the LidSat computer.

Finally, I have all versions of the bitstreams for the different architectures here.

Knock yourselves out 👊🤘

❤️ 2

Merge branch 'master' into DMAtest Verified 656932e

JoarGjersund approved these changes on Apr 30 View changes

JoarGjersund left a comment

Great. Tested and can confirm the cubedma test is working on my picobob setup as well!

❤️ 1

JoarGjersund commented on Apr 30 • edited

okey.. almost working.. its not working on the hypso-cli side due to a timeout issue. Think this might be related to #182.

```
Received DMA Test request                    hsi geocapture              - Captures some HSI frames and georef.
    Generating data                          hsi gettemp                 - Poll the HSI camera temperature.
        Starting transfer                    hsi debugcapture            - Captures and saves up to 10 frames
    MM2S completed after 360084 "times"      hsi dmatest                 - Tests the CubeDMA module.
    S2MM completed after 0 "times"
    Transfer success                         (hypso) hsi dmatest
The number of in- and out-bytes are the same --> Sending Request to test DMA
->The data is equal!                             ACK Timeout
End of test!                                 `hsi dmatest` failed: 1 (Operation not permitted)
```

magnudan commented on Apr 30 Author

@**JoarGjersund**: That timeout message is misleading... There has not been implemented any return from opu to cli...

Fixed other issues due to merge from master 9025f8c

magnudan commented on Apr 30 Author

I discovered some issues with the merge that resulted in the build not succeeding. I have pushed a fix, so that the build succeeds. Will test that they work tomorrow!

SW kanban board automation moved this from **Review in progress** to **Reviewer approved** on May 1

sivertba approved these changes on May 1 — View changes

sivertba left a comment

I have tested the system as you proposed (although I found the hypso-cli in cli folder and not the opu folder) and got the same results as **@JoarGjersund**

If you can provide a screenshot of the hypso-CLI **not** timing out I would say that this PR is ready to be merged :)

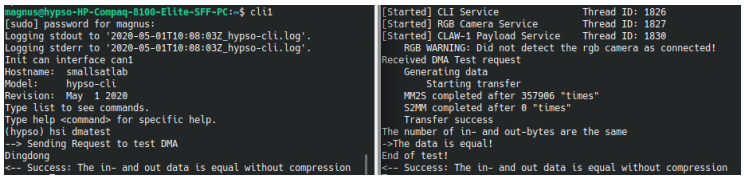
magnudan commented on May 1 — Author

@DennisNTNU: Can you read through the code, and see that the merge conflicts have been resolved correctly, and can you test that camera interfacing on this branch works as they should?

 `Configured return packet from opu to cli with verbose messages` 5969cf2

magnudan commented on May 1 • edited — Author

Commit 5969cf2 implements return messages from opu to cli. The cli will now print the results from the test unless, of course, it times out.



magnudan added 3 commits on May 1

 `Removed 'Dingding' from cli print` bacc65f

 `Removed hardcoded error that skips HW compression` 8cf73f2

 `Removed print explaining the hardcoded removed in last commit` 27445a0

magnudan commented on May 1 — Author

With commit 27445a0, HW compression is enabled for HSI cubes obtained through the capture functions. Logs for both `hsi capture 10` and `hsi debugcapture 10` are included at the bottom of this comment. These have been tested on the PicoBOB in the LidSat setup with IP .50. This PicoBOB is connected to a HSI camera with the following ID configuration in `src/hsi/hsi_camera.c` , and must changed accordingly if used with this setup:

```
static HIDS hCam = 1001; // default device ID of ehternet cameras
static HIDS camera = 1;  // Camera ID set in ueyesetid or idscameramanager
```

Even though this seems to work now, I would still like to get a review from **@DennisNTNU**

`hsi capture 10` :

```
Received capture request of 10 frames
Max res cube byte size: 47083520
AOI cube byte size: 18432000
Individual frame byte size: 1843200
Binned cube byte size: 1540800
  Camera initialized!
  Setting camera config parameters ...
    Pixel clock 118
    Image format: 36
    Area of interest x,y: 328,248 width,height: 1280,720
```

```
    Frame rate 19.000000
    Exposure time 40.000000
    Color mode to 12 bit
    Display mode to memory
    Turn on Gain Boost
    Setting master gain 0
    Enable Freerun mode
    Enable new frame events
  Camera parameters set!
  Frame buffers:
  Index; ID; Address
    0;   1;  0x7255e008
    1;   2;  0x7239b008
    2;   3;  0x721d8008
    3;   4;  0x72015008
    4;   5;  0x71e52008
    5;   6;  0x71c8f008
    6;   7;  0x71acc008
    7;   8;  0x71909008
    8;   9;  0x71746008
    9;   10;  0x71583008
Starting freerun capture
Waiting for frame: 1/10
    Next frame will be at ID 2, 0x7239b008
    Newest frame is at buffer address: 0x7255e008
  Given buffer 0x7255e008 has index 0, and ID 1
  Measured framerate: 1.500618
  Binning frame and storing in binned cube memory
Waiting for frame: 2/10
    Next frame will be at ID 3, 0x721d8008
    Newest frame is at buffer address: 0x7239b008
  Given buffer 0x7239b008 has index 1, and ID 2
  Measured framerate: 14.963877
  Binning frame and storing in binned cube memory
Waiting for frame: 3/10
    Next frame will be at ID 4, 0x72015008
    Newest frame is at buffer address: 0x721d8008
  Given buffer 0x721d8008 has index 2, and ID 3
  Measured framerate: 14.963855
  Binning frame and storing in binned cube memory
Waiting for frame: 4/10
    Next frame will be at ID 5, 0x71e52008
    Newest frame is at buffer address: 0x72015008
  Given buffer 0x72015008 has index 3, and ID 4
  Measured framerate: 14.963855
  Binning frame and storing in binned cube memory
Waiting for frame: 5/10
    Next frame will be at ID 6, 0x71c8f008
    Newest frame is at buffer address: 0x71e52008
  Given buffer 0x71e52008 has index 4, and ID 5
  Measured framerate: 14.963855
  Binning frame and storing in binned cube memory
Waiting for frame: 6/10
    Next frame will be at ID 7, 0x71acc008
    Newest frame is at buffer address: 0x71c8f008
  Given buffer 0x71c8f008 has index 5, and ID 6
  Measured framerate: 14.963877
  Binning frame and storing in binned cube memory
Waiting for frame: 7/10
    Next frame will be at ID 8, 0x71909008
    Newest frame is at buffer address: 0x71acc008
  Given buffer 0x71acc008 has index 6, and ID 7
  Measured framerate: 14.963855
  Binning frame and storing in binned cube memory
Waiting for frame: 8/10
    Next frame will be at ID 9, 0x71746008
    Newest frame is at buffer address: 0x71909008
  Given buffer 0x71909008 has index 7, and ID 8
  Measured framerate: 14.963855
  Binning frame and storing in binned cube memory
Waiting for frame: 9/10
    Next frame will be at ID 10, 0x71583008
    Newest frame is at buffer address: 0x71746008
  Given buffer 0x71746008 has index 8, and ID 9
  Measured framerate: 14.963855
  Binning frame and storing in binned cube memory
Waiting for frame: 10/10
    Next frame will be at ID 1, 0x7255e008
    Newest frame is at buffer address: 0x71583008
  Given buffer 0x71583008 has index 9, and ID 10
  Measured framerate: 14.963855
  Binning frame and storing in binned cube memory
HSI cube captured and binned
Camera de-initialized!
Beginning Compression
        Doing hardware compression
```

```
          Cube dma initialized
          Starting transfer
      MM2S completed after 358961 "times"
      S2MM completed after 0 "times"
      Transfer success
  Compression done
 Saving configuration
 Capture cmd finished, 0


hsi debugcapture 10：


  Received capture request of 10 frames
  Max res cube byte size: 47083520
  AOI cube byte size: 18432000
  Individual frame byte size: 1843200
  Binned cube byte size: 1540800
    Camera initialized!
    Setting camera config parameters ...
      Pixel clock 118
      Image format: 36
      Area of interest x,y: 328,248 width,height: 1280,720
      Frame rate 1.000000
      Exposure time 40.000000
      Color mode to 12 bit
      Display mode to memory
      Turn on Gain Boost
      Setting master gain 0
      Enable Freerun mode
      Enable new frame events
    Camera parameters set!
    Frame buffers:
    Index; ID; Address
      0;   1;  0xa0c3e008
      1;   2;  0xa0a7b008
      2;   3;  0xa08b8008
      3;   4;  0xa06f5008
      4;   5;  0xa0532008
      5;   6;  0xa036f008
      6;   7;  0xa01ac008
      7;   8;  0x9ffe9008
      8;   9;  0x9fe26008
      9;   10;  0x9fc63008
  Starting freerun capture
  Waiting for frame: 1/10
      Next frame will be at ID 2, 0xa0a7b008
      Newest frame is at buffer address: 0xa0c3e008
    Given buffer 0xa0c3e008 has index 0, and ID 1
    Saving frame: ID: 1, 0xa0c3e008
      as raw BIP
    Append frame buffer to full cube file
      written 921600 shorts
    Measured framerate: 1.847671
    Binning frame and storing in binned cube memory
  Waiting for frame: 2/10
      Next frame will be at ID 3, 0xa08b8008
      Newest frame is at buffer address: 0xa0a7b008
    Given buffer 0xa0a7b008 has index 1, and ID 2
    Saving frame: ID: 2, 0xa0a7b008
      as raw BIP
    Append frame buffer to full cube file
      written 921600 shorts
    Measured framerate: 0.999733
    Binning frame and storing in binned cube memory
  Waiting for frame: 3/10
      Next frame will be at ID 4, 0xa06f5008
      Newest frame is at buffer address: 0xa08b8008
    Given buffer 0xa08b8008 has index 2, and ID 3
    Saving frame: ID: 3, 0xa08b8008
      as raw BIP
    Append frame buffer to full cube file
      written 921600 shorts
    Measured framerate: 1.000282
    Binning frame and storing in binned cube memory
  Waiting for frame: 4/10
      Next frame will be at ID 5, 0xa0532008
      Newest frame is at buffer address: 0xa06f5008
    Given buffer 0xa06f5008 has index 3, and ID 4
    Saving frame: ID: 4, 0xa06f5008
      as raw BIP
    Append frame buffer to full cube file
      written 921600 shorts
    Measured framerate: 0.999733
    Binning frame and storing in binned cube memory
  Waiting for frame: 5/10
```

```
    Next frame will be at ID 6, 0xa036f008
    Newest frame is at buffer address: 0xa0532008
  Given buffer 0xa0532008 has index 4, and ID 5
  Saving frame: ID: 5, 0xa0532008
    as raw BIP
  Append frame buffer to full cube file
    written 921600 shorts
  Measured framerate: 1.000282
  Binning frame and storing in binned cube memory
Waiting for frame: 6/10
    Next frame will be at ID 7, 0xa01ac008
    Newest frame is at buffer address: 0xa036f008
  Given buffer 0xa036f008 has index 5, and ID 6
  Saving frame: ID: 6, 0xa036f008
    as raw BIP
  Append frame buffer to full cube file
    written 921600 shorts
  Measured framerate: 0.999733
  Binning frame and storing in binned cube memory
Waiting for frame: 7/10
    Next frame will be at ID 8, 0x9ffe9008
    Newest frame is at buffer address: 0xa01ac008
  Given buffer 0xa01ac008 has index 6, and ID 7
  Saving frame: ID: 7, 0xa01ac008
    as raw BIP
  Append frame buffer to full cube file
    written 921600 shorts
  Measured framerate: 0.999733
  Binning frame and storing in binned cube memory
Waiting for frame: 8/10
    Next frame will be at ID 9, 0x9fe26008
    Newest frame is at buffer address: 0x9ffe9008
  Given buffer 0x9ffe9008 has index 7, and ID 8
  Saving frame: ID: 8, 0x9ffe9008
    as raw BIP
  Append frame buffer to full cube file
    written 921600 shorts
  Measured framerate: 1.000282
  Binning frame and storing in binned cube memory
Waiting for frame: 9/10
    Next frame will be at ID 2, 0xa0a7b008
    Newest frame is at buffer address: 0xa0c3e008
  Given buffer 0xa0c3e008 has index 0, and ID 1
  Saving frame: ID: 1, 0xa0c3e008
    as raw BIP
  Append frame buffer to full cube file
    written 921600 shorts
  Measured framerate: 1.000282
  Binning frame and storing in binned cube memory
Waiting for frame: 10/10
    Next frame will be at ID 3, 0xa08b8008
    Newest frame is at buffer address: 0xa0a7b008
  Given buffer 0xa0a7b008 has index 1, and ID 2
  Saving frame: ID: 2, 0xa0a7b008
    as raw BIP
  Append frame buffer to full cube file
    written 921600 shorts
  Measured framerate: 0.999733
  Binning frame and storing in binned cube memory
HSI cube captured and binned
Camera de-initialized!
Saving binned cube
Beginning Compression
        Doing hardware compression
        Cube dma initialized
        Starting transfer
    MM2S completed after 358878 "times"
    S2MM completed after 0 "times"
    Transfer success
 Compression done
Saving configuration
Debug capture cmd finished, 0
```

magnudan assigned **DennisNTNU** on May 1

magnudan requested a review from **rogerbirkeland** on May 1

DennisNTNU commented on May 4 • edited

I am getting no error, but the size of the compressed cube seems to be always the hardcoded 500 * 720 * 107 = 77040000 Bytes, or 73.5MB,

```
-rw-r--r--    1 root     root      77040000 Apr 30 12:04 883860308Data.raw
```

as defined in line 11 in cubeDMA.h

```
#define CUBE_SIZE (500 * 720 * 107) // frames * height * width
```

I tested with 15 and with 50 frames, and the size is the same. Decompression takes a long time, but does not give me errors. The decompressed file is way too big, and I am not able to extract images with my usual tool. So the issue might just be that a way too large file is stored, and that confuses compression.

Also, a question that I had for a while, this hardcoded size, is that the size that is also allocated on boot for the binned cube, and would that mean that we cant take cubes containing more than 500 frames?

This was referenced on May 4

Logging process #183 Merged

opu status flags and TM message #208 Closed

rogerbirkeland approved these changes on May 4 View changes

rogerbirkeland left a comment

I see the same case with file size as **@DennisNTNU** . Apart from that, and with my very limited insight into code and functionality, `hsi dmatest` , `hsi debugcapture` and `hsi capture` produce what looks like sensible outputs.

To get on with proper testing of both this and #183 , I think we should merge.

rogerbirkeland mentioned this pull request on May 4

Not possible to continue downloads with opu download? #213 Closed

magnudan commented on May 4 Author

TL;DR

- Cube DMA works
- CCSDS 123 Version 1 (compression) works as Johan implemented it, but it is not what we want

The long story

I have realized that the issue addressed in opu-system issue nr. 95 might not be known to all. While the interfacing with Cube DMA now works as it should, the FPGA compression expects cubes of one, predefined, size. The size it expects now is 500 frames of 720 * 107 pixels (h*w), or 38520000 shorts (`uint16_t` s).

Cube DMA will still move the amount of shorts defined in `CUBE_SIZE` . The reason this is hardcoded as a `#define` is that it is what CCSDS 123 expects. I see that it is not the best practice to implement it this way, as it will not be a constant value in the future.

When executing one of the `hsi capture` commands with a bitstream containing compression is loaded, one of three things will happen. This is based on x, the number of shorts in the binned cube:

- x < 38520000: Cube DMA will move the x shorts and the following (38520000 - x) shorts to CCSDS 123, resulting in a compressed cube with trailing rubbish data. The following shorts will be whatever is in the physical memory locations Cube DMA reads from.
- x = 38520000: Cube DMA moves the entire binned cube and compresses only the data you want. YAY!
- x > 38520000: Only the 38520000 first shorts of the binned cube is compressed.

I see that the difference between Cube DMA and CCSDS 123 might not be as clear to all, and I should have been more specific when writing this PR. In this PR, I have focused on Cube DMA, the module that moves data to/from the FPGA, and not compression (CCSDS 123) in FPGA.

rogerbirkeland commented on May 4

Some more questions:

```
* x < 38520000: Cube DMA will move the x shorts and the following (38520000 - x)
shorts to CCSDS 123, resulting in a compressed cube with trailing rubbish data.
The following shorts will be whatever is in the physical memory locations Cube
DMA reads from.
```

If the rubbish data is trailing, can we make a chopper that removes this data so we dont have to download everything?

```
* x = 38520000: Cube DMA moves the entire binned cube and compresses only the
data you want. YAY!

* x > 38520000: Only the 38520000 first shorts of the binned cube is compressed.
```

OK, so that limits the size of frames/physical area of observation and so on. Can we (if we are not able to fix?) use SW compression in these cases?

> I see that the difference between Cube DMA and CCSDS 123 might not be as clear to all, and I should have been more specific when writing this PR. In this PR, I have focused on Cube DMA, the module that moves data to/from the FPGA, and not compression (CCSDS 123) in FPGA.

Good clarification, thanks!

magnudan commented on May 4 Author

@rogerbirkeland and **@DennisNTNU**: Can you confirm that camera interfacing works like it should, not considering compression?

rogerbirkeland commented on May 4

> **@rogerbirkeland** and **@DennisNTNU**: Can you confirm that camera interfacing works like it should, not considering compression?

Did not have any issues, so it seems OK. As I wrote in #208 , things are slow/hangs a bit when the FPGA is working (I thing), so `hsi gettemp` does not return. But it seems like the requests are buffered, and handeled later. This also means that its possible to overfill the buffer.

DennisNTNU commented on May 4 • edited

The reason why `hsi gettemp` doesn't answer, is because the hsi-service is busy. opu-services should still respond to csp pings, since the csp service is in a different thread, whereas `hsi capture` and `hsi gettemp` get handled in the same thread. Its also csp (different thread), that handles enqueueing of incoming csp packets, which is why the `hsi gettemp` command does get handled one `hsi capture` is done.

In `hsi_setup_and_capture()` , hsi_service.c, line 187, the function `cubedma_init()` is called and later in `hsi_setup_and_capture()` , line 223, the function `hsi_compress_cube()` , also calls `cubedma_init()` on line 261. Can the second call be removed?

magnudan commented on May 4 • edited Author

@rogerbirkeland: I will address these questions in tomorrows software meeting

@DennisNTNU: Yes, the `hsi_init()` needs only to be called once, but it must be called before capture. This is because the function will make `*binned_cube` point to the physical memory Cube DMA reads from. From what I can see, it would make sense to change the lines in the function `hsi_compress_cube()`, line 255 to 270 from

```
    mmap_error = 0;

    int fpga_comp_error = 0;
    if (!mmap_error)
    {
        printf("        Doing hardware compression\n");
        cubedma_init(&binned_cube, &compressed_cube);
        printf("        Cube dma initialized\n");
        fpga_comp_error = compress_cube_fpga(&compressed_cube_size_bytes);
    }

    if (mmap_error || fpga_comp_error)
    {
        debug_printf("Doing software compression, mmap err: %i; fpga err: %i", mmap_erro
        sw_comp_error = compress_cube_software(hsi_config, binned_cube, compressed_cube,
    }
```

to

```
int fpga_comp_error = 0;
if (!mmap_error)
{
    printf("        Doing hardware compression\n");
    fpga_comp_error = compress_cube_fpga(&compressed_cube_size_bytes);
}
if (mmap_error || fpga_comp_error)
{
    debug_printf("Doing software compression, mmap err: %i; fpga err: %i", mmap_error, f
    sw_comp_error = compress_cube_software(hsi_config, binned_cube, compressed_cube, &co
}
```

Removed hardcoding of mmap_error and cubedma_init() from hsi_compress… 0e11568

…

magnudan commented on May 5 Author

The above changes have been implemented, and everything still seems to work with `hsi debugcapture 10`. Log attatched below. As this was the last suggestion for improvements, I will merge the branch to master.

```
Received capture request of 10 frames
Max res cube byte size: 47083520
AOI cube byte size: 18432000
Individual frame byte size: 1843200
Binned cube byte size: 1540800
  Camera initialized!
  Setting camera config parameters ...
    Pixel clock 118
    Image format: 36
    Area of interest x,y: 328,248 width,height: 1280,720
    Frame rate 1.000000
    Exposure time 40.000000
    Color mode to 12 bit
    Display mode to memory
    Turn on Gain Boost
    Setting master gain 0
    Enable Freerun mode
    Enable new frame events
  Camera parameters set!
  Frame buffers:
  Index; ID; Address
    0;   1;  0xa0c3e008
    1;   2;  0xa0a7b008
    2;   3;  0xa08b8008
    3;   4;  0xa06f5008
    4;   5;  0xa0532008
    5;   6;  0xa036f008
    6;   7;  0xa01ac008
    7;   8;  0x9ffe9008
```

```
    8;   9;  0x9fe26008
    9;   10;  0x9fc63008
Starting freerun capture
Waiting for frame: 1/10
    Next frame will be at ID 2, 0xa0a7b008
    Newest frame is at buffer address: 0xa0c3e008
  Given buffer 0xa0c3e008 has index 0, and ID 1
  Saving frame: ID: 1, 0xa0c3e008
    as raw BIP
  Append frame buffer to full cube file
    written 921600 shorts
  Measured framerate: 1.845958
  Binning frame and storing in binned cube memory
Waiting for frame: 2/10
    Next frame will be at ID 3, 0xa08b8008
    Newest frame is at buffer address: 0xa0a7b008
  Given buffer 0xa0a7b008 has index 1, and ID 2
  Saving frame: ID: 2, 0xa0a7b008
    as raw BIP
  Append frame buffer to full cube file
    written 921600 shorts
  Measured framerate: 1.000007
  Binning frame and storing in binned cube memory
Waiting for frame: 3/10
    Next frame will be at ID 4, 0xa06f5008
    Newest frame is at buffer address: 0xa08b8008
  Given buffer 0xa08b8008 has index 2, and ID 3
  Saving frame: ID: 3, 0xa08b8008
    as raw BIP
  Append frame buffer to full cube file
    written 921600 shorts
  Measured framerate: 1.000007
  Binning frame and storing in binned cube memory
Waiting for frame: 4/10
    Next frame will be at ID 5, 0xa0532008
    Newest frame is at buffer address: 0xa06f5008
  Given buffer 0xa06f5008 has index 3, and ID 4
  Saving frame: ID: 4, 0xa06f5008
    as raw BIP
  Append frame buffer to full cube file
    written 921600 shorts
  Measured framerate: 1.000007
  Binning frame and storing in binned cube memory
Waiting for frame: 5/10
    Next frame will be at ID 6, 0xa036f008
    Newest frame is at buffer address: 0xa0532008
  Given buffer 0xa0532008 has index 4, and ID 5
  Saving frame: ID: 5, 0xa0532008
    as raw BIP
  Append frame buffer to full cube file
    written 921600 shorts
  Measured framerate: 1.000007
  Binning frame and storing in binned cube memory
Waiting for frame: 6/10
    Next frame will be at ID 7, 0xa01ac008
    Newest frame is at buffer address: 0xa036f008
  Given buffer 0xa036f008 has index 5, and ID 6
  Saving frame: ID: 6, 0xa036f008
    as raw BIP
  Append frame buffer to full cube file
    written 921600 shorts
  Measured framerate: 1.000007
  Binning frame and storing in binned cube memory
Waiting for frame: 7/10
    Next frame will be at ID 8, 0x9ffe9008
    Newest frame is at buffer address: 0xa01ac008
  Given buffer 0xa01ac008 has index 6, and ID 7
  Saving frame: ID: 7, 0xa01ac008
    as raw BIP
  Append frame buffer to full cube file
    written 921600 shorts
  Measured framerate: 1.000007
  Binning frame and storing in binned cube memory
Waiting for frame: 8/10
    Next frame will be at ID 9, 0x9fe26008
    Newest frame is at buffer address: 0x9ffe9008
  Given buffer 0x9ffe9008 has index 7, and ID 8
  Saving frame: ID: 8, 0x9ffe9008
    as raw BIP
  Append frame buffer to full cube file
    written 921600 shorts
  Measured framerate: 1.000007
  Binning frame and storing in binned cube memory
Waiting for frame: 9/10
    Next frame will be at ID 1, 0xa0c3e008
    Newest frame is at buffer address: 0x9fc63008
```

```
  Given buffer 0x9fc63008 has index 9, and ID 10
  Saving frame: ID: 10, 0x9fc63008
    as raw BIP
  Append frame buffer to full cube file
    written 921600 shorts
  Measured framerate: 1.000007
  Binning frame and storing in binned cube memory
Waiting for frame: 10/10
    Next frame will be at ID 2, 0xa0a7b008
    Newest frame is at buffer address: 0xa0c3e008
  Given buffer 0xa0c3e008 has index 0, and ID 1
  Saving frame: ID: 1, 0xa0c3e008
    as raw BIP
  Append frame buffer to full cube file
    written 921600 shorts
  Measured framerate: 1.000007
  Binning frame and storing in binned cube memory
HSI cube captured and binned
Camera de-initialized!
Saving binned cube
Beginning Compression
        Doing hardware compression
        Cube dma initialized
        Starting transfer
    MM2S completed after 352869 "times"
    S2MM completed after 0 "times"
    Transfer success
 Compression done
Saving configuration
Debug capture cmd finished, 0
```

magnudan merged commit **635b4ce** into `master` on May 5 **Revert**

SW kanban board automation moved this from **Reviewer approved** to **Done** on May 5

magnudan deleted the `DMAtest` branch on May 5 **Restore branch**

AC hypso-sw Issue 193

Edit | New issue

Jump to bottom

Log processing rate of HSI frames #193

Open **magnudan** opened this issue on Apr 17 · 2 comments

Labels	HSI points=5
Projects	SW kanban board

magnudan commented on Apr 17

It is crucial that the OPU is able to process frames at a higher rate than the camera captures frames to prevent congestion.

It's especially useful for testing, and also maybe for telemetry/opu health?

DennisNTNU commented on Apr 24

Something like this could be useful, but It is not 100% clear to me how this should be implemented. Did you have implementation details in mind when making this issue? Maybe the OPU could print timestamps after each frame, or print the time elapsed between different stages of the frame processing sequence? Maybe also generating a separate file/log containing specifically frame processing info.

sivertba commented on Apr 30

clock cycles from the binning?

sivertba added the points=5 label on Apr 30

sivertba added this to **Backlog** in **SW kanban board** on Apr 30

DennisNTNU added the HSI label 17 days ago

Assignees

No one—assign yourself

Labels

HSI

points=5

Projects

SW kanban board

Backlog

Milestone

No milestone

Linked pull requests

Successfully merging a pull request may close this issue.

None yet

3 participants

Pin issue

AD hypso-sw Issue 194

Edit | New issue

Jump to bottom

Merge hsi_capture_and_binn_cube() and hsi_debug_capture_and_binn_cube() #194

Closed **magnudan** opened this issue on Apr 17 · 0 comments

Labels points=5

Projects SW kanban board

magnudan commented on Apr 17

The two functions in `src/hsi/hsi_camera.c` should have the same core functionality, with some extra prints etc. for debug. The reason we should merge the two is to ensure that the two capture frames, does binning etc. in exactly the same way. Then, when `hsi debugcapture` is used to fix some issue, changing the source code should also fix the issue for `hsi capture`. This is easier to do when the commands call on the same function. Also less code ❤️

I suggest that the merged function takes in either a boolean for debug mode or a struct with options (for future expansion?), and do debug-stuff in `if`s

DennisNTNU mentioned this issue on Apr 23

Binning and other fixes #202 Merged

sivertba added this to **Review in progress** in **SW kanban board** on Apr 30

DennisNTNU closed this on Apr 30

SW kanban board automation moved this from **Review in progress** to **Done** on Apr 30

sivertba added the points=5 label on May 14

Assignees

No one—assign yourself

Labels

points=5

Projects

SW kanban board

Done

Milestone

No milestone

Linked pull requests

Successfully merging a pull request may close this issue.

None yet

3 participants

Pin issue

AE hypso-sw Issue 195

Edit | New issue

Jump to bottom

hsi capture hangs after capture #195

Closed **rogerbirkeland** opened this issue on Apr 17 · 5 comments

Labels	points=3
Projects	SW kanban board

rogerbirkeland commented on Apr 17

opu-services and hypso-cli is latest from master 17. april.

hypso-cli: hsi cature 10

After cature is done, everything hangs.

Output from opu-services:

```
root@339905d-primary:/media/sd/rbi# ./opu-services 12 can0
src/utils/logging.c:366:log_init_std_streams: Failed to open log_streaxilinx_can
e0008000.can can0: bitrate error 0.0%
m 2020-04-17T07:15:02Z_opu-services.log (Invalid argument)
Init can interface can0
CSP initialisation complete
Hostname: opu
Model:    opu-services
Revision: Apr 17 2020
[Started] CSP Services              Thread ID: 1639
[Started] File Transfer Service     Thread ID: 1640
[Started] TFTP Service              Thread ID: 1641
[Started] CLI Service               Thread ID: 1642
[Started] RGB Camera Service        Thread ID: 1643
[Started] CLAW-1 Payload Service    Thread ID: 1644
    RGB WARNING: Did not detect the rgb camera as connected!
Received hsi camera temperature poll request
    Getting Device Info of camera with device ID: 1001 ...
    Firmware Version: 0
    Temperature: 51.4 | 51.4
    Link speed [Mb]: 1000
    Comport offset?: 65535
    Device ID: 1001
Received capture request of 10 frames
    Camera initialized!
    Camera parameters set!
    1: 0xa9a7b008
    2: 0xa98b8008
    3: 0xa96f5008
    4: 0xa9532008
    5: 0xa936f008
    6: 0xa91ac008
    7: 0xa8fe9008
```

```
8: 0xa8e26008
9: 0xa8c63008
10: 0xa8aa0008
    Frame: 1/10
    Frame: 2/10
    Frame: 3/10
    Frame: 4/10
    Frame: 5/10
    Frame: 6/10
    Frame: 7/10
    Frame: 8/10
    Frame: 9/10
    Frame: 10/10
HSI cube captured!
Camera de-initialized!
Beginning Compression
```

magnudan commented on Apr 17

I think this is because the master branch has a faulty interface with Cube DMA. This *should* be fixed with PR #191

rogerbirkeland commented on Apr 21

Had more problems with debug-capture today. See last comment in #183. This time, also debugcapture died.

DennisNTNU mentioned this issue on Apr 23

Binning and other fixes #202 Merged

sivertba added the points=3 label on Apr 30

sivertba added this to **To do** in **SW kanban board** on Apr 30

sivertba moved this from **To do** to **Backlog** in **SW kanban board** on Apr 30

sivertba moved this from **Backlog** to **To do** in **SW kanban board** on Apr 30

sivertba moved this from **To do** to **Review in progress** in **SW kanban board** on May 5

magnudan commented on May 6

What's the status on this? Did the Cube DMA fix solve this?

DennisNTNU commented on May 8

I think the CubeDMA did solve this, I haven't had any freezes/crashes since. The issue can be closed unless someone still experienced hangs like this.

sivertba commented on May 14

#191 closes #195

sivertba closed this on May 14

SW kanban board automation moved this from **Review in progress** to **Done** on May 14

Assignees

No one—assign yourself

Labels

points=3

Projects

SW kanban board

Done

Milestone

No milestone

Linked pull requests

Successfully merging a pull request may close this issue.

None yet

4 participants

Pin issue

AF hypso-sw Issue 197

Edit | New issue

Jump to bottom

Clarify variable names in HSI service #197

Closed **magnudan** opened this issue on Apr 17 · 0 comments

Labels points=2

Projects SW kanban board

magnudan commented on Apr 17

I am specifically referring to the instances of `binned_bytes` and functions that use it. It is misleading because it should represent the number of `uint16_t` s (shorts) written to the Cube DMA, but because the variable is called *bytes*, it logically must be twice as large. This has, and will cause confusion if not clarified.

In general, all variables that count things should end with `_count` , and be very specific to what they count.

There are also a lot of variables in `src/hsi/hsi_util.c` that can be a lot better.

magnudan mentioned this issue on Apr 17

Cube DMA now works #191 Merged

DennisNTNU mentioned this issue on Apr 23

Binning and other fixes #202 Merged

sivertba added the points=2 label on Apr 30

sivertba added this to **Backlog** in **SW kanban board** on Apr 30

sivertba moved this from **Backlog** to **Review in progress** in **SW kanban board** on Apr 30

DennisNTNU closed this on Apr 30

SW kanban board automation moved this from **Review in progress** to **Done** on Apr 30

Assignees

No one—assign yourself

Labels

points=2

Projects

SW kanban board

Done

Milestone

No milestone

Linked pull requests

Successfully merging a pull request may close this issue.

None yet

3 participants

Pin issue

AG hypso-sw Pull Request 202

NTNU-SmallSat-Lab / **hypso-sw** Private

Binning and other fixes #202

Edit | Open with

Merged **DennisNTNU** merged 14 commits into `master` from `binning-and-other-fixes` on Apr 30

Conversation 8 | Commits 14 | Checks 0 | Files changed 13 | +975 −520

DennisNTNU commented on Apr 23 • edited

Highlights:

- *Should* now no longer segfault during capture no matter the fps or frame count setting, but instead be very slow and print lots of errors, when fps and/or frame count is too high.
- The default fps was reduced from 20 to 10, and the picobob is capable of capturing with 10 fps at 1280x720 resolution.
- After capture, the hsi-service writes a timestamped capture configuration to file containing parameters like fps, frame count, resolution etc.

The picozed completely freezes and no longer responds to csp pings or network pings when compression starts. this is Issue #195. Maybe because the wrong bitstream file is used or because Magnus' cdma fix isn't on this branch. Software compression doesn't crash when I force it by setting a flag in the code. I have however not yet verified capture by decopressing the compressed cube and looking at the actual images.

Changes

- Includes Magnus' binning fix suggestions.
- Implemented binning for x86.
- Merged the capture and debugcapture code. Now debugcapture does the same as normal capture except some additional printing and file saving.
- Adjusted the ueye function call sequence during capture a bit.
- Improved the ueye api error code check function.
- Saving config to file after capture

I don't want to merge this before PR #191 is merged, because there will be conflicts that I'd rather like to resolve in this PR.

I changed the default camera ID to be pointing to the lidsat camera (camera ID from 1 to 2), and no longer to the other camera near the HIL computer. During testing, the lidsat camera was capable of capturing with higher framerates than the HIL camera for to me unknown reasons.

Solves #124, #169, #194. Partly addresses #175, #197. Hopefully also solves #196.

🎉 3

Reviewers
- rogerbirkeland ✓
- magnudan ✓
- magne-hov ●
- sivertba ●

Assignees
No one—assign yourself

Labels
None yet

Projects
SW kanban board
Done

Milestone
No milestone

Linked issues
Successfully merging this pull request may close these issues.
None yet

5 participants

DennisNTNU added 8 commits on Apr 16

magnus' binning fixes	618a096
Binning for x86	24011d4
Add write config to path functions	53c8f7a
writing config used in capture to file	c81208c
Added bigger ueye error function and other changes	33e3784
ueye serror function adjustemnt, lower fps, code reordering, +	82a9077
Add some common capture functions	4ecf55b
Using the new capture functions; forcing SW compression	03b6153

rogerbirkeland commented on Apr 23

Very good work and analysis. I wondered if the error I see was due to the fpga-stuff failing. Is the bitstream.bit just supposed to be in the current working dir?

It seems like we need to have a bit more control over the version configs, if the bitstream can kill everything?

👍 1

DennisNTNU added 4 commits on Apr 27

- `skip compression when capture function returns error` `595bf90`
- `Adjust ueye api function call sequence for capture, improve info prin…` `0d3404f`
  ...
- `verbose camera configuration` `acec6d3`
- `Save also binned cube in debug capture, better variable naming in hsi…` `52a6fb5`
  ...

DennisNTNU requested review from **sivertba**, **magne-hov** and **rogerbirkeland** and removed request for **sivertba** on Apr 29

- `fix overflow error for high frame count in printf of cube sizes` `2800845`

sivertba added this to **Review in progress** in **SW kanban board** on Apr 30

rogerbirkeland approved these changes on Apr 30 View changes

rogerbirkeland left a comment

Looks like it works as it should. I think this should be merged as soon as we can, as some of these requests are lagging behind other features "needed" to do proper testing.

SW kanban board automation moved this from **Review in progress** to **Reviewer approved** on Apr 30

rogerbirkeland commented on Apr 30

@DennisNTNU , **@magnudan** One of you decide if it should be merged now. I think it looks ready.

sivertba commented on Apr 30

@magnudan maybe look at this?

DennisNTNU commented on Apr 30 Author

I will merge this in the evening.

magnudan reviewed on Apr 30 View changes

src/hsi/hsi_util.c

```
31  -  static int g_samples_last_bin_count;
32  -  static int g_full_binns_per_row_count;
33  -  static int g_bands_binned_per_row_count;
34  -  static int g_binned_pixels_per_frame_count;
35  -  static int g_cube_rows_count;
```

```
36      - static int g_cube_clumns_count;
37      - static int last_pixel_in_row_offset;
38      - static int frame_pos_offset_first_row;
    33  + static int sample_count_in_last_bin;
    34  + static int full_bins_count;
    35  + static int total_bins_count;
    36  + static int binned_frame_pixel_size;
39  37
40      - static int frame_x_offset_sensor;
41      - static int frame_y_offset_sensor;
42      -
43      - static uint16_t newArr1[BINNINGFACTOR];
44      - static uint16_t newArr2[BINNINGFACTOR];
45      - static uint16_t newArr3[BINNINGFACTOR];
    38  + #ifdef __arm__
    39  + static uint16_t bin_array1[BINNINGFACTOR];
    40  + static uint16_t bin_array2[BINNINGFACTOR];
    41  + static uint16_t bin_array3[BINNINGFACTOR];
    42  + #endif // __arm__
```

Comment on lines -31 to +42

magnudan on Apr 30

YES!

😄 1 👀 1

Reply…

Resolve conversation

magnudan approved these changes on Apr 30 — **View changes**

magnudan left a comment

Very good job! Have not tested, but the code looks great

`update comments, fix comments` `9f69335`

DennisNTNU merged commit `ba9f45f` into `master` on Apr 30 — **Revert**

SW kanban board automation moved this from **Reviewer approved** to **Done** on Apr 30

DennisNTNU deleted the `binning-and-other-fixes` branch on Apr 30 — **Restore branch**

magne-hov commented on Apr 30

Cool stuff! Code looks good :)

❤️ 1

AH hypso-sw Issue 220

Edit | New issue

Jump to bottom

Configure flash return from HSI camera #220

Closed **magnudan** opened this issue on May 8 · 2 comments

Assignees	
Labels	HIGH PRIORITY points=5
Projects	SW kanban board

 **magnudan** commented on May 8

Is your feature request related to a problem? Please describe.
In order for time-stamping (issue #188) to work, the system needs a signal to interrupt at the start and stop of camera exposure. The iDS camera used for HSI imaging can be configured to pull a signal high or low during exposure. This is called the flash signal. There are several ways to configure this signal. When to start, stop, pull high or low etc.

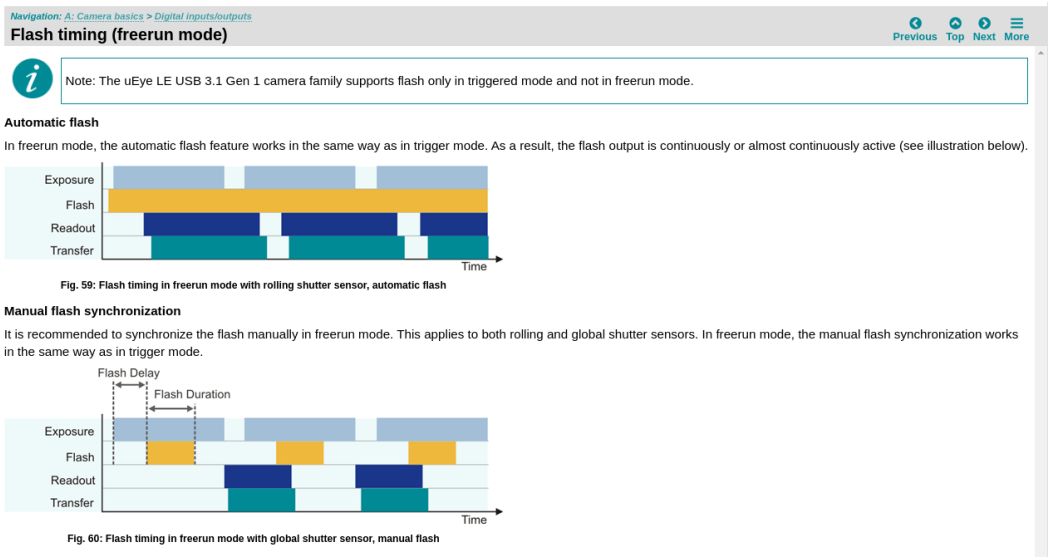
Describe the solution you'd like
We want the signal to be pulled low from the start of exposure of a frame to the end of the exposure of a frame.

Describe alternatives you've considered
If this is not possible, the most important information is the start of the frame.

Additional context
I find iDS' documentation a bit ambiguous. One place it's written that

> Flash duration (in µs). If 0 is passed, the flash output will be active until the end of the exposure time. For sensors with Global Start Shutter this is the time until the end of exposure of the first sensor row.

I understand this as the flash is active only during exposure, but this screenshot from the page on flash timing in freerun mode (requires login) suggests that the flash will be active during the entire imaging sequence.



If this is right, we must choose a constant flash duration, and only get a timestamp at the start of a frame.

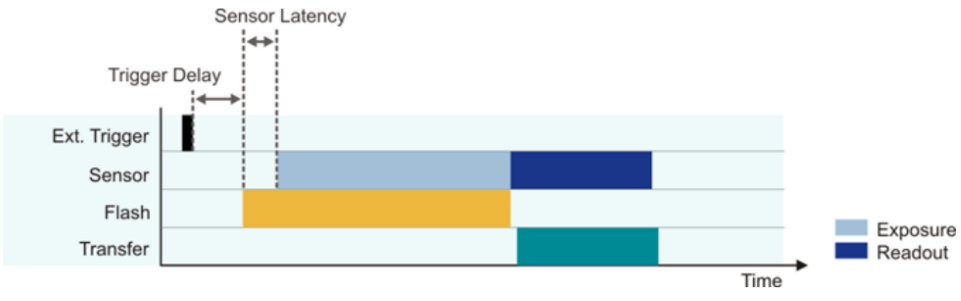
magnudan self-assigned this on May 8

magnudan added the **HIGH PRIORITY** label on May 8

sivertba added this to **In progress** in **SW kanban board** on May 11

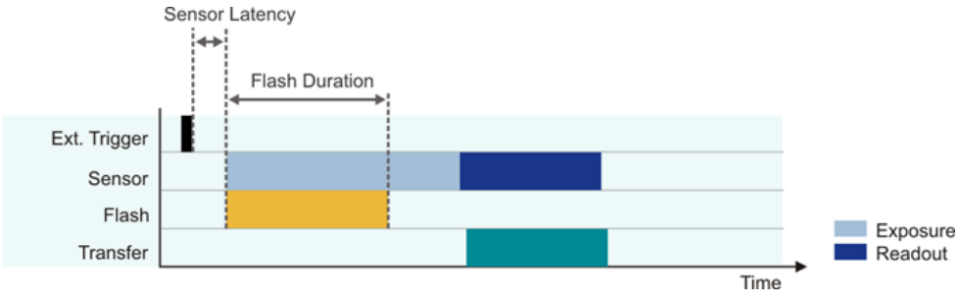
magnudan commented on May 11

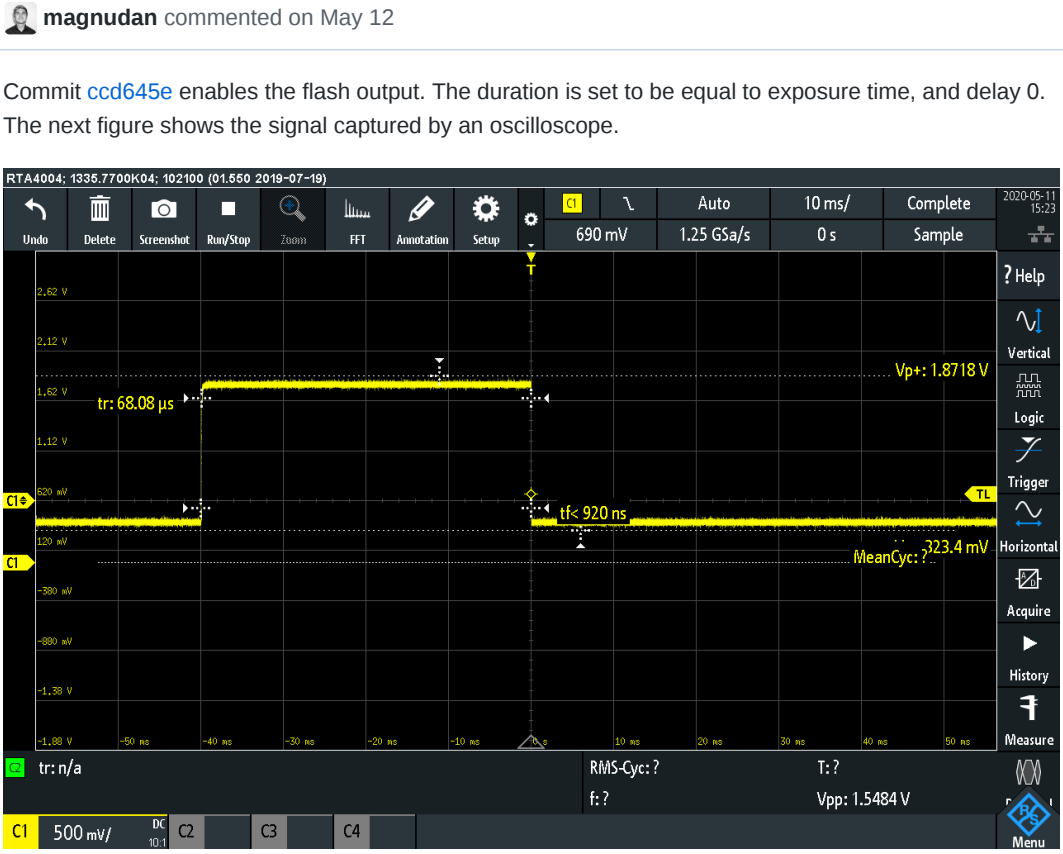
We want to use manual flash synchronization, as automatic is inaccurate:

> The disadvantage is that the flash signal is active slightly longer than the exposure time.
>
> 
>

While manual is based on the exact start of exposure

You can thus achieve a higher accuracy with the manual flash synchronization than with the automatic flash feature.





1

magnudan mentioned this issue on May 13

Flash is enabled #236 Merged

magnudan closed this in #236 on May 14

SW kanban board automation moved this from **In progress** to **Done** on May 14

sivertba added the points=5 label on May 14

Assignees

magnudan

Labels

HIGH PRIORITY

points=5

Projects

SW kanban board

Done

Milestone

No milestone

Linked pull requests

Successfully merging a pull request may close this issue.

Flash is enabled

2 participants

Pin issue

AI hypso-sw Issue 235

Edit | New issue

Jump to bottom

HSI (debug)capture fails to init camera 50% of the times it's run #235

Closed **magnudan** opened this issue on May 13 · 3 comments

Labels bug points=8

Projects SW kanban board

magnudan commented on May 13 • edited

Describe the bug
Every other time the command `hsi (debug)capute` fails to init the camera 50% of the times it's run.

To Reproduce
Steps to reproduce the behavior:

1. `hsi (debug)capute`

Runs normally

2. `hsi (debug)capute`

```
Unable to initialize camera
hsi_camera.c:239 ueye_api error is_InitCamera code 3: An attempt to initialize or
select the camera failed (no camera connected or initialization error).
src/hsi/hsi_service.c:325:hsi_setup_and_capture: Capture error, code 1
```

3. `hsi (debug)capute`

Runs normally

4. `hsi (debug)capute`

```
Unable to initialize camera
hsi_camera.c:239 ueye_api error is_InitCamera code 3: An attempt to initialize or
select the camera failed (no camera connected or initialization error).
src/hsi/hsi_service.c:325:hsi_setup_and_capture: Capture error, code 1
```

Expected behavior
A normal execution every time

Additional context
@**sivertba** has experienced this using the `opu-services` that boots on the LidSat IP .42
@**rogerbirkeland** has also experienced this and documented it well in PR #229
I'm also experiencing this issue on the flash-branch at commit `ccd645e`

magnudan added the `bug` label on May 13

This was referenced on May 13

Flash is enabled #236 — Merged

Hsi capture cmd extension #229 — Merged

sivertba commented on May 14

A restart of opu-services for each image might help?
something with the init procedure needs to be looked at

sivertba added the `points=8` label on May 14

sivertba added this to **Backlog** in **SW kanban board** on May 14

sivertba moved this from **Backlog** to **To do** in **SW kanban board** on May 14

DennisNTNU commented 28 days ago • edited

When I was testing the ueye api function `is_GetVsyncCount` yesterday, I was able to perform multiple captures with differing fps from the same process of opu-services without the error code 3, attempt to initialize camera failed. The branch was `driver-frame-count`, but I was using a 'dirty' master branch because I was testing before committing. But it still works today on both the master branch `5db7da1` and on the `driver-frame-count` branch `82802d3`.

In the hypso-cli log, you can see 8 consecuting `hsi capture` commands with differing options.
When testing now however, one of the 8 commands still resulted in an error, a different one though, can be seen at line 1285 in the opu-services log

```
hsi_camera.c:257 ueye_api error is_InitCamera code 197: The device is already paired.
```

The I did some double checking with `ueyesetid`, quickly starting and exiting it to monitor the status of the camera and it went through 4 different phases. Right after capture was done, the camera was still in use:

```
2 devices found
+------+----------+---------+--------+----------------------+--------------+
|Device|Device id |Camera id| in use |    Camera Model      |  Serial No.  |
+------+----------+---------+--------+----------------------+--------------+
|   1  |   1001   |    1    |   no   | UI526xSE-M           | 4103494261   |
|   2  |   1002   |    2    |   yes  | UI526xCP-M           | 4103433958   |
+------+----------+---------+--------+----------------------+--------------+
```

Then the camera was gone:

```
1 device found
+------+----------+---------+--------+----------------------+--------------+
|Device|Device id |Camera id| in use |    Camera Model      |  Serial No.  |
+------+----------+---------+--------+----------------------+--------------+
|   1  |   1001   |    1    |   no   | UI526xSE-M           | 4103494261   |
+------+----------+---------+--------+----------------------+--------------+
```

Then it showed up again but still in use:

```
2 devices found
+------+----------+---------+--------+----------------------+--------------+
|Device|Device id |Camera id| in use |    Camera Model      |  Serial No.  |
+------+----------+---------+--------+----------------------+--------------+
|   1  |   1001   |    1    |   no   | UI526xSE-M           | 4103494261   |
|   2  |   1002   |    2    |   yes  | UI526xCP-M           | 4103433958   |
+------+----------+---------+--------+----------------------+--------------+
```

Then the in use flag disappeared:

```
2 devices found
+------+----------+---------+--------+----------------------+--------------+
|Device|Device id |Camera id| in use |    Camera Model      |  Serial No.  |
+------+----------+---------+--------+----------------------+--------------+
|   1  |   1001   |    1    |   no   | UI526xSE-M           | 4103494261   |
|   2  |   1002   |    2    |   no   | UI526xCP-M           | 4103433958   |
+------+----------+---------+--------+----------------------+--------------+
```

This all took about 3 seconds. So there is some reset procedure the camera goes through after it was closed in software.

This leads me to assume that last time, there was some issue with this reset process causing it to be fulfilled only after the camera was attempted to be opened again.

logs from the test run on `driver-frame-count , 82802d3` :
hypso-cli:
200519T120921_hypso-cli.log
opu-services:
200519T120910_opu-services.log

DennisNTNU mentioned this issue 25 days ago

Multiple cam init attempts #253 Merged

sivertba commented 25 days ago

This issue does not persist as off commit `7805541`
Thus, this issue is closed.

1

sivertba closed this 25 days ago

SW kanban board automation moved this from **To do** to **Done** 25 days ago

Assignees

No one—assign yourself

Labels

bug
points=8

Projects

SW kanban board

Done

Milestone

No milestone

Linked pull requests

Successfully merging a pull request may close this issue.

None yet

3 participants

Pin issue

AJ hypso-sw Pull Request 236

NTNU-SmallSat-Lab / **hypso-sw** Private

Flash is enabled #236

Edit | Open with

Merged **magnudan** merged 4 commits into `master` from `flash` on May 14

Conversation 1 | Commits 4 | Checks 0 | Files changed 1 | +34 −1

magnudan commented on May 13

This PR resolves #220

Note: The flash return is wired to the PicoBOB with IP .42, and is only usable by that board. Other PicoBOBs will still be able to run this code, but only the one with IP .42 is able to register the signal.

There is not yet a way to verify that the signal works from opu-services or hypso-cli. This will be implemented with issue #188. For now, please verify that the additional code introduced in this PR is sound and that it does not break captures. Proof that the flash signal is correct can bee seen in issue #220.

The flash return is set to have rising flank at the beginning of exposure (delay = 0) and duration equal to exposure time. The delay shall not be changed, but the duration should be arbitrary, as long as it's less than or equal to exposure time.

The bug described in #235 that causes the camera not to be initialized 50% of the times is also observed on this branch, but as it does not originate from this branch, I think it's safe to merge.

magnudan added 4 commits on May 7

`Enabling flash signal`	`b5be6d8`
`Manual configuration of flash delay and duration`	`b76a282`
`Scailng exposure time. Comments`	`182bf07`
`Setting mode made flash work!`	`ccd645e`

magnudan requested review from **rogerbirkeland**, **magne-hov** and **DennisNTNU** on May 13

DennisNTNU approved these changes on May 14 — View changes

DennisNTNU left a comment • edited

Verified capture is still working. Code is good.

magnudan merged commit `f0e5412` into `master` on May 14 — Revert

magnudan deleted the `flash` branch on May 14 — Restore branch

Reviewers
- DennisNTNU ✓
- rogerbirkeland ●
- magne-hov ●

Assignees
No one—assign yourself

Labels
None yet

Projects
None yet

Milestone
No milestone

Linked issues
Successfully merging this pull request may close these issues.
Configure flash return from HSI ca…

2 participants

AK hypso-sw Pull Request 240

NTNU-SmallSat-Lab / **hypso-sw** Private

Timestamping of HSI frames is live! #240

Edit | Open with

Merged **magnudan** merged 32 commits into `master` from `timestamp` on May 15

Conversation 13 | Commits 32 | Checks 0 | Files changed 5 | +194 −2

magnudan commented on May 15

This PR resolves #188.

This implementation is dependent on:

- The flash return signal from the HSI camera
  - Flash (-) is connected to GND on connector J5 pin 2 on BOB v3
  - Flash (+) is connected to MIO47 on PicoZed (JX3-40 on BOB) and 1V8 through a pull-up resistor on connector J5 pin 5 on BOB v3
- A PPS signal connected to MIO9 (JX2-8 on BOB) on connector P2 pin 1 and/or 2 on BOB v3
  - An external MCU is used to supply this signal in the LidSat setup. See the test-mcus repository
- Operating system from PR NTNU-SmallSat-Lab/opu-system#122

Notes on implementation
In the source code for timestamps (`src/utils/timestamp.c` and `include/utils/timestamp.h`), there are three functions:

- `timestamps_enable()` : Initiates interrupts and saves the current system time to a variable. **Must be called before the following functions**
- `timestamps_to_file(char *path, unsigned int n_frames)` : Saves timestamps for `n_frames` to the file specified in `*path` . The first line of the file will be the system time from the enable function. The following lines will be the rising and falling flank of the flash signal in us since the system time given. **Must be called after HSI capture has finished**
- `timestamps_disable` : Disables interrupts.

Notes on testing

- The only hardware setup to be able to test this as of now is the PicoBOB in the LidSat setup with IP .42
- The timestamps are saved to a file `timestamps.txt` :
  - The first timestamp of a rising flank is not what is expected. I suspect this is because the flash signal is pulled high when flash is initialized on the camera instead of at the beginning of first exposure...
  - Right now, I'm saving the timestamps for one more frame than is captured in case the first timestamp is garbage data, and we actually need to read `n_frames + 1` timestamps to get all. This needs to be looked into!
- To test this thoroughly, we need to be in the lab, scope the signal, compare the signal to the oled counter in the frames, and analyze the results.

Discussion points

- Formatting of the timestamp file
  - Naming scheme (static/dynamic)
  - Formatting of system time: How do we want to store this information?
  - The actual timestamps are in us relative to when the interrupts are enabled. How to display this info?
- When in the HSI capture sequence should we enable, save, and disable timestamps/interrupts? I just put it somewhere...

Example of timestamp file with comments. Capture of 5 frames, exposure of 40ms

```
942764548                // System time at start of capture
  1004216        4292251 // Start of first frame is not correct. Difference: 3288035
us. Start time should be around 4252295 (4292251 - 39956)
  6220084        6260038 // Differnece: 39954 us
  7218093        7258049 // Differnece: 39956 us
  8216106        8256060 // Differnece: 39954 us
  9214117        9254071 // Differnece: 39954 us
```

Reviewers
DennisNTNU ✓
rogerbirkeland ●
magne-hov ●

Assignees
No one—assign yourself

Labels
None yet

Projects
None yet

Milestone
No milestone

Linked issues
Successfully merging this pull request may close these issues.
Integrate and test implementation …

5 participants

```
  9497066        10369726 // Saved one more frame than captured. Seems like rubbish
data, since neither rising or falling edge fit into the pattern. Difference: 872660
us
```

Can someone relay this information to Mariusz?

simennett and others added 28 commits on Feb 27

Added timestamp.c file containing timestamp code	90fad21
initial commit with timestamp header file	f1a0fa7
Merge branch 'timestamp' of https://github.com/NTNU-SmallSat-Lab/hyps… ...	333120d
updatet due to functions update	653da03
no change just clean up	ac1321f
corrected funtion name and removed unnecessary definitions and includes	5c0a133
Fixed bug with ppsCount resetting	a3fd449
Renamed the old timestamp codes, added new ones for implementing into… ...	a18b0ed
made folders for source files	36bc3b3
Merge branch 'master' into timestamp	d27c83e
Added functions to write timestamps to fil	d7241de
Removed old files	11fcf5c
Fixed a typo. added return unread bytes. calloc -> malloc	1929c7c
Added a test application. Fixed for loop bug	ab102cd
Moved some printing	635f4a0
Merge branch 'master' into timestamp	1f4cafc
Fixed some printing and error handing	269c34c
Changed names	8812084
Merge branch 'master' into timestamp	bb67ce3
Changed source name in CMakeLists.txt	1842a22
Two new functions: enable/disable timestamp interrupt	22b429e
Changed som var names	3bb0212
Clang-format	ea14912
Merge branch 'master' into timestamp	0ae5a18
Merge branch 'master' into timestamp	41e1257
Integrated timestamps source code to hsi service	3f76bc0
Merge branch 'master' into timestamp	735ea29
Print rising edge of last frame?	51460f7

magnudan requested review from **rogerbirkeland** and **magne-hov** on May 15

magnudan requested a review from **DennisNTNU** on May 15

DennisNTNU commented on May 15

I just debugcaptured and got a `timestamp.txt` file that looks like this:

```
671535964
  1000000         5085264
  7016274         7056231
  8014287         8054241
  9012297         9052252
 10010309        10050264
 11008319        11048275
 12006331        12046287
 13004340        13044298
 14002353        14042309
 15000362        15040318
        0             233
```

Seems as expected, the configured fps is 1Hz, though its strange that the camera skipped the second frame but i guess that can happen with the ueye cameras. What I am mostly wondering about is what the first value represents: `671535964` . This doesn't seem to be a unix timestamp, how can we convert this number to date and time?

magnudan commented on May 15 • edited Author

> What I am mostly wondering about is what the first value represents: 671535964. This doesn't seem to be a unix timestamp, how can we convert this number to date and time?

This is how it's generated. It's copy-pasted from how the compressed cubes are named. If you have another preferred way to do this, please update the procedure. This can be considered a placeholder:

```
// Global
static struct timespec system_time;

// In timestamps_enable()
clock_gettime(CLOCK_REALTIME, &system_time);

// In timestamps_to_file(), after opening the file
char system_time_string[32];
sprintf(system_time_string, "%lli", (long long)system_time.tv_nsec);
fprintf(fdFile, "%s\n", system_time_string);
```

DennisNTNU commented on May 15 • edited

OH so thats how the compressed cube names are generated and why they seem to be so random. This number is only the nanosecond part of the `timespec` structure. I am going to commit a change that also prints the seconds part, with which the numbers can be converted to a date.

❤️ 1

magnudan commented on May 15 Author

Ideally, there should be a global struct with a time that is defining for the whole capture

magnudan commented on May 15 Author

And based on your timestamp file, I guess there is no need to print timestamps for one more frame than we capture? Or is that because of the missing 2nd frame? Does the number of timestamps correspond with how many frames you wanted to capture?

I see that this file can help a lot in detecting missing frames etc. Nice additional feature 🕐🕑🕒🕓🕔

DennisNTNU commented on May 15 • edited

> Does the number of timestamps correspond with how many frames you wanted to capture?

Yes the number of timestamps correspond with how many frames I wanted to capture.

> I guess there is no need to print timestamps for one more frame than we capture?

No I don't think so, because the last line, `0 233` doesn't seem like it contains useful or correct information so unless it actually does it could be removed.

I made a small change that causes the `timestamps.txt` to look as follows

```
1589538159 599367424
  1000000         4786066
  6717097         6757048
  7715109         7755059
  8713119         8753070
  9711130         9751080
 10709141        10749093
 11707154        11747105
 12705165        12745115
        0              37
```

Where the first number in the first line is the usual Unix time as in seconds from 01.01.1970.

`Print also seconds in first line of timestamps.txt` `89fbabf`

DennisNTNU approved these changes on May 15 — View changes

DennisNTNU left a comment

And with that I think its good to be merged

`Print only timestamps for captured frames, not +1` `f74618e`

magnudan commented on May 15 — Author

I pushed a last commit that removes the last line. I don't think you need to re-test this

👍 1

magnudan commented on May 15 — Author

Also, I'm considering adding some sort of helptext:

```
=== System time in UNIX ===
1589538159 599367424

=== Timestamps in us since system time ===
  1000000         4786066
  6717097         6757048
  7715109         7755059
  8713119         8753070
  9711130         9751080
 10709141        10749093
 11707154        11747105
 12705165        12745115
```

👍 1

magnudan added 2 commits on May 15

`Added helptext to timestamp file` `ab9c363`

`Fixed error in print statement` `97e41a9`

magnudan commented on May 15 — Author

This is the file generated from `hsi debugcapture 10` at commit `97e41a9`:

```
=== System time in UNIX ===
1589552701 370307804

=== Timestamps in us since system time ===
  1000000         4361811
  6292945         6332898
  7290956         7330908
  8288966         8328920
  9286977         9326931
 10284990        10324944
 11282999        11322951
 12281009        12320963
 13279021        13318974
 14277033        14316985
```

Unless someone objects, I'll merge this soon™

rogerbirkeland commented on May 15

Haven't been able to test this, but followed the discussion. Looks good. How is it possible to validate and compare with "true" time?

magnudan commented on May 15 • edited Author

> How is it possible to validate and compare with "true" time?

This is a big one! I don't know the accuracy of the time displayed on the OLED screen the HSI points at.

When we get the time from the EPS, I think we must trust that the accuracy is as good as we can get it.

I don't really know if we should be concerned with testing the accuracy of clocks in this PR. I think it will be better to raise these concerns in a separate issue. This way we get the code to master quicker. If there are any bugs, they will be found quicker this way, as it is accessible to more testers.

rogerbirkeland commented on May 15

Yes, totally agree. Did not intend to raise an issue postponing the merge.

fre. 15. mai 2020, 17:17 skrev Magnus Danielsen <notifications@github.com>:
...

👍 1

magnudan merged commit `016c072` into `master` on May 15 Revert

magnudan deleted the `timestamp` branch on May 15 Restore branch

magnudan mentioned this pull request on May 15

How do we test correctness of timestamping? #242 Open

AL hypso-sw Issue 242

Edit | New issue

Jump to bottom

How do we test correctness of timestamping? #242

Open **magnudan** opened this issue on May 15 · 3 comments

Labels	HSI Metadata, Help wanted, Question, Testing, points=8
Projects	SW kanban board

magnudan commented on May 15

As discussed in PR #240, we don't really have a way to verify the correctness of system time and timestamps. Bullet points from the top of my head:

- The setup has an OLED screen with a counter
- The OPU will get system time from the EPS, and we can't really do anything about the accuracy NA has. We do, however, need to account for transfer delay
- We have an oscilloscope that can be used to verify the signals frequency and duty cycle if its accuracy is good enough

Any other thoughts or ideas?

magnudan added Help wanted, Question, Testing labels on May 15

magnudan commented on May 15

As mentioned in NTNU-SmallSat-Lab/test-mcus#5, the accuracy of the PPS might be off by 2 ms. This will lead to the counter resetting 2 ms too early every second.

DennisNTNU commented 26 days ago

Maybe a logic analyzer on the flash pin could be useful.

sivertba commented 19 days ago

should look at clock drift in MCU used NTNU-SmallSat-Lab/test-mcus#5 and maybe combine with a GPS module.

Validation of relative time should not be an issue, absolute timestamping needs a GPS module.
Could consider syncing with justervesenet or something

sivertba added the points=8 label 19 days ago

sivertba added this to **Backlog** in **SW kanban board** 19 days ago

DennisNTNU added the **HSI Metadata** label 17 days ago

magnudan mentioned this issue 5 days ago

Make a test for timestamping images #121 Closed

Assignees

No one—assign yourself

Labels

HSI Metadata
Help wanted
Question
Testing
points=8

Projects

SW kanban board

Backlog

Milestone

No milestone

Linked pull requests

Successfully merging a pull request may close this issue.

None yet

3 participants

Pin issue

AM hypso-sw Issue 251

Edit | New issue

Jump to bottom

Want error handling for "Failed opening timestamp device." #251

Closed **sivertba** opened this issue 27 days ago · 2 comments

Assignees 

Labels Enhancement bug

sivertba commented 27 days ago • edited

Describe the bug
When the timestamping is not found on the OPU, we get the following prompt:
"Failed opening timestamp device."
and a segmentation fault

To Reproduce
This was found on the following commit on hypso-sw: `5db7da1`
and NTNU-SmallSat-Lab/opu-system@ `f9d3acf`

```
Automatic run of hypso-sw # 67
Commit: 5db7da11f9
Job cause : Started by timer
Result: SUCCESS
Duration: 4 min 24 sec and counting
```

Might be a problem with the wrong bitstream or a mismatch between the hypso-sw and the opu-system build

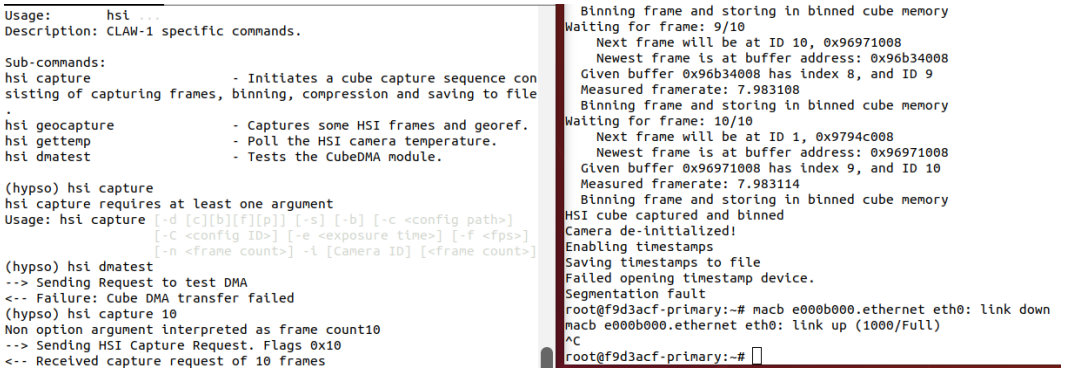
Run opu-services and request 10 frames in hypso-cli by typing:

```
hsi capture 10
```

Expected behavior

- Having the time stamping feature as part of the bitstram and not failing
- If failing to open the timestamp device, do the following
  - Prompt in/to hypso-cli "no timestamp device found
  - Save the images without timestamp on OPU
- Keep calm and no segmentation faulting

Screenshots



sivertba added bug **Enhancement** labels 27 days ago

sivertba assigned **magnudan** 27 days ago

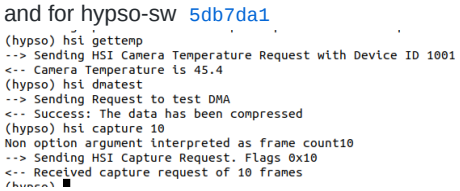
magnudan commented 27 days ago

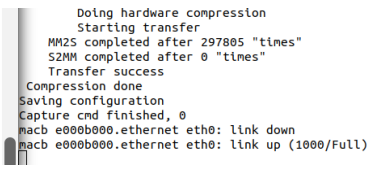
After talking with **@sivertba** on slack, we found out that opu-system was built on commit f9d3acf (sorry, no link), which is before timestamps were introduced to the master branch. The segfault likely happened because the program tried to interact with a non-existent kernel module.

@sivertba will try to build new boot files from master, and hopefully this will solve the issue

sivertba commented 27 days ago

Looks like using a matching commit for opu-system and hypso-sw did the trick!
With:
NTNU-SmallSat-Lab/opu-system@ `f9d3acf`
and for hypso-sw `5db7da1`






2

 **sivertba** closed this 27 days ago

Assignees

 **magnudan**

Labels

Enhancement

bug

Projects

None yet

Milestone

No milestone

Linked pull requests

Successfully merging a pull request may close this issue.

None yet

2 participants

Pin issue

AN hypso-sw Pull Request 253

NTNU-SmallSat-Lab / **hypso-sw** Private

Multiple cam init attempts #253

Edit | Open with

Merged **DennisNTNU** merged 3 commits into `master` from `multiple-cam-init-attempts` 19 days ago

Conversation 3 | Commits 3 | Checks 0 | Files changed 1 | +19 −9

DennisNTNU commented 25 days ago • edited

- Attempts to open the camera 3 times before returning an error.
- Waits 1.5 seconds between attempt.

I didn't get the problem described in #235 so I couldn't tests if multiple init attempts would fix it.

DennisNTNU added 3 commits 27 days ago

`Attempting to open the camera 3 times before returning error` b0868c5

`Merge branch 'master' into multiple-cam-init-attempts` a93b6fe

`use for instead for while; add comments` 8b468b8

evelynlimore added this to **Review in progress** in **SW kanban board** 25 days ago

DennisNTNU requested review from **magnudan** and **rogerbirkeland** 23 days ago

rogerbirkeland commented 20 days ago

Is this still needed, or did this issue magically get fixed?

SW kanban board automation moved this from **Review in progress** to **Reviewer approved** 20 days ago

rogerbirkeland approved these changes 20 days ago View changes

rogerbirkeland left a comment

Not tested; code looks resonable.

DennisNTNU commented 20 days ago Author

The code gods scared the issue away and to our relief and someones detriment, it migrated to a different random repository on github. Still a nice-to-have, thanks for approving!

👀 1

DennisNTNU merged commit `4585f42` into `master` 19 days ago Revert

SW kanban board automation moved this from **Reviewer approved** to **Done** 19 days ago

DennisNTNU deleted the `multiple-cam-init-attempts` branch 19 days ago Restore branch

Reviewers
rogerbirkeland ✓
magnudan ●

Assignees
No one—assign yourself

Labels
None yet

Projects
SW kanban board
Done

Milestone
No milestone

Linked issues
Successfully merging this pull request may close these issues.
None yet

2 participants

NTNU-SmallSat-Lab / **hypso-sw** Private

Delete apps/tstest.c #255

Edit | Open with

Merged **jonakor** merged 1 commit into `master` from `remove_tstest` 25 days ago

Conversation 2 | Commits 1 | Checks 0 | Files changed 1 | +0 −37

jonakor commented 25 days ago

The tstest.c was used during testing of timestamping without opu-services. can be removed

`Delete apps/tstest.c` ··· Verified `8641b67`

jonakor requested a review from **magnudan** 25 days ago

magnudan commented 25 days ago

I agree that the test code can be deleted. I think I removed the make instructions from `CMakeLists.txt` , but is there more that needs to be removed?

I have compiled both for x86 and ARM without any issues. I have not tested the executables, but I cannot see that this will have affected them in any way.

Feel free to merge when you have addressed my first question

magnudan approved these changes 25 days ago — View changes

jonakor commented 25 days ago • edited — Author

yes, I made changes only in CMakeLists.txt and you **@magnudan** have removed them as I can see

jonakor merged commit `95edaf2` into `master` 25 days ago — Revert

jonakor deleted the `remove_tstest` branch 25 days ago — Restore branch

Reviewers
magnudan ✓

Assignees
No one—assign yourself

Labels
None yet

Projects
None yet

Milestone
No milestone

Linked issues
Successfully merging this pull request may close these issues.
None yet

2 participants

AO hypso-sw Issue 260

Edit | New issue

Jump to bottom

Change .raw file format to .bip file format #260

Closed **magnudan** opened this issue 21 days ago · 5 comments

Assignees	
Labels	HSI, HSI Metadata, Pipeline, Question, points=2
Projects	SW kanban board

magnudan commented 21 days ago

As .raw is ambiguous, and both cubes and frames are stored in BIP (Band Interleaved by Pixel) format. I think it will be better to call the format .bip, as that is what it is.

magnudan added the Question label 21 days ago

DennisNTNU commented 21 days ago

I completely agree

drowzie commented 21 days ago • edited

I also suggest adding some extra information for more clarification of the image as we do not use metadata for uncompressed images, also might be helpful for compressed.

- signed/unsigned
- Pixelsize: 16
- Image sizes
- Endianess
  For eaxmple, images i am currently testing and working on is called:
  NAME-XXXXXXXXX-u16be-ZZZxYYYxXXX.BIP
  Where u16 refers to unsigned integers of bitsize16.
  BE refers to big endian, but for us this would be low endian.

For compressed images it might be more beneficial to use some other naming convention. The compressed images is not BIP, and it should perhaps use .bin or create a new one.

sivertba commented 19 days ago

Should create new issue that adresses having a header file for the BIP file following the ENVI standard

sivertba added the points=2 label 19 days ago

sivertba commented 19 days ago

Look if CCSDS have a standard for compressed image cubes

sivertba added this to **Backlog** in **SW kanban board** 19 days ago

sivertba mentioned this issue 19 days ago

Command to the hsi-service to compress a previously captured binned cube #265 Open

sivertba moved this from **Backlog** to **To do** in **SW kanban board** 19 days ago

sivertba assigned **sivertba** and **DennisNTNU** 19 days ago

sivertba mentioned this issue 19 days ago

Naming strategies for processing pipeline outputs #230 Closed

DennisNTNU added **HSI Metadata** Pipeline HSI labels 17 days ago

drowzie commented 17 days ago • edited

CCSDS does not have a naming standard as image details are in the image header.
Could still add extra ENVI metadata as an extra file for extra information. Even so, the compressed files should still be using different naming convention.

DennisNTNU mentioned this issue 7 days ago

All hsi file names are static and stored in a folder #279 Merged

sivertba moved this from **To do** to **Review in progress** in **SW kanban board** 7 days ago

DennisNTNU closed this 3 days ago

SW kanban board automation moved this from **Review in progress** to **Done** 3 days ago

Assignees

sivertba

DennisNTNU

Labels

HSI

HSI Metadata

Pipeline

Question

points=2

Projects

SW kanban board

Done

Milestone

No milestone

Linked pull requests

Successfully merging a pull request may close this issue.

None yet

4 participants

Pin issue

AP hypso-sw Pull Request 279

NTNU-SmallSat-Lab / **hypso-sw** Private

All hsi file names are static and stored in a folder #279

Edit | Open with

Merged **DennisNTNU** merged 14 commits into `master` from `hsi-naming` 3 days ago

Conversation 2 | Commits 14 | Checks 0 | Files changed 10 | +424 −294

DennisNTNU commented 7 days ago

Uncompressed data (full cube, individual frames, binned cube) now has the file ending .bip, according to #260. Compressed cube now has the static name `compressed_cube.bip.cmpr` . All these files including the config file and the timestamps are now saved in a folder called hsi.

Solves #260 and depending on whether we want more dynamic numbering the `hsi` folder e.g. `hsi0`, `hsi1`, `...` or not also solves possibly #230.

DennisNTNU added 10 commits 7 days ago

Save binned cube if Hw compression fails and if it isnt already saved	54a7d08
remove old commented code	3d1b662
Save hsi files in a folder	ad4105a
change .raw to .bip	69390d8
some clang formatting	35b84ed
add struct for less complex variable passing in hsi capture code	5611b9d
Merge branch 'save-binned-b4-SW-compr' into hsi-naming	bbcdca2
Finalized hsi capture state struct implementation	1118694
Define hsi capture folder in hsi capture state struct instead of as c… ...	c496549
Merge branch 'master' into hsi-naming	26095ee

DennisNTNU requested review from **magnudan**, **magne-hov**, **rogerbirkeland** and **sivertba** and removed request for **magnudan** 7 days ago

DennisNTNU added **Enhancement** HSI **HSI Metadata** Pipeline labels 7 days ago

Merge branch 'master' into hsi-naming	947c51d

rogerbirkeland approved these changes 6 days ago — View changes

rogerbirkeland left a comment

For now, I think this is enough and should be merged (will help with starting with scripting and automation of buffering and so on).

However, I do think that we should have multiple folders (`hsi0` , `hsi1` , `hsi2` ...)

DennisNTNU added 3 commits 5 days ago

add hsi capture files folder indexing and creating a new indexed fold… ...	6e51d2b
clang formatting hsi_util.c	7172d22

Reviewers
- rogerbirkeland ✓
- magne-hov ●
- magnudan ●
- sivertba ●

Assignees
No one—assign yourself

Labels
Enhancement
HSI
HSI Metadata
Pipeline

Projects
SW kanban board
Done

Milestone
No milestone

Linked issues
Successfully merging this pull request may close these issues.
None yet

2 participants

`Creating new hsi folder only if camera has been initialized; Add comm…` `3dec9d8`
…

DennisNTNU commented 5 days ago • edited Author

> However, I do think that we should have multiple folders (hsi0, hsi1, hsi2...)

That is now implemented as well. The hsi service now looks for how many folders there are already in the current working directory that start with `hsi` and their appended index `x`, and chooses the least index `x_min` that doesn't exist already and creates a folder with that index `hsix_min`.

evelynlimore added this to **Review in progress** in **SW kanban board** 5 days ago

DennisNTNU merged commit **`3850d1a`** into `master` 3 days ago Revert

SW kanban board automation moved this from **Review in progress** to **Done** 3 days ago

DennisNTNU deleted the `hsi-naming` branch 3 days ago Restore branch

DennisNTNU mentioned this pull request 3 days ago

Naming strategies for processing pipeline outputs #230 Closed

AQ opu-system Issue 32

Edit | New issue

Jump to bottom

Option to enable static IP #32

Closed **magnudan** opened this issue on Feb 4 · 4 comments

Labels: HIGH PRIORITY enhancement points=5

Projects: SW kanban board

magnudan commented on Feb 4

PicoZeds and ZedBoards that connect to the internet should boot with a static IP address. Orakel has supplied us with a set of addresses. A static IP-address is specified in `/etc/network/interfaces`, and should look like this:

```
# The loopback interface
auto lo
iface lo inet loopback

# The ETH interface
auto eth0
iface eth0 inet static
    address 129.241.2.XXX
    netmask 255.255.254.0
    broadcast 129.241.2.255
    gateway 129.241.2.1
```

Is it possible to specify the `XXX` part when configuring embedded Linux?

magnudan added the **enhancement** label on Feb 4

magnudan commented on Feb 4 • edited

A workaround is to unzip `/opu-system/petalinux/projects/software/software.tar.xz`, and add the file `/opu-system/petalinux/projects/software/<unzipped>/etc/network/interfaces` with the correct configuration, and then re-zip it and replace `/opu-system/petalinux/projects/software/software.tar.xz`. Then the petalinux project can be made in the usual way.

evelynlimore added this to **To do** in **SW kanban board** on Feb 18

evelynlimore added the points=5 label on Feb 19

rogerbirkeland added the **HIGH PRIORITY** label on Feb 28

rogerbirkeland commented on Feb 28

This (seems to be) needed for MOBIP. But #48 should be fixed first to see if static IP is needed.

JoarGjersund commented on Mar 2

@magnudan the workaround mentioned is not valid as the network driver will read the configuration file before it is beeing extracted by the startup script. Should however be possible to put it directly on the rootfs by adding it in rootfs_config.bb in /opu-system/petalinux/projects/software/. Will look into it..

JoarGjersund commented on Mar 2

should be solved in `9aca3ea`

JoarGjersund closed this on Mar 2

SW kanban board automation moved this from **To do** to **Done** on Mar 2

Assignees

No one—assign yourself

Labels

HIGH PRIORITY
enhancement
points=5

Projects

SW kanban board

Done

Milestone

No milestone

Linked pull requests

Successfully merging a pull request may close this issue.

None yet

4 participants

Pin issue

AR opu-system Issue 39

Edit | New issue

Jump to bottom

current head of master does not set up USB correctly #39

Closed **JoarGjersund** opened this issue on Feb 7 · 12 comments

Assignees

Labels bug

JoarGjersund commented on Feb 7 • edited

We have no contact with rgb camera on the current master head. The system running on the Hardware-in-the-loop is able to connect with the rgb-camera over usb, but is running with a different setup. We currently do not know this setup and what is configuration is wrong on the master head.

I cannot stress enough how important it is that we use the makeproject script in the petalinux folder when we generate bootfiles for testing.

JoarGjersund added the bug label on Feb 7

JoarGjersund assigned **JoarGjersund**, **magnudan**, **DennisNTNU** and **jlgarrett** on Feb 7

magnudan commented on Feb 8

I have created the branch rgb-testing and started looking into the problem in Vivado configuration.

The current commit c2c945b
changes speeds for MIO on PicoBOB based on what they are on the ZedBoard. This change has not fixed the problem of connecting to the RGB.

The ZedBoard is mostly configured using a standard-setting in Vivado, while many settings for the PicoBOB must be configured manually. As the ZedBoard successfully interfaces with the RGB camera, I will find instances where the two configure USB interfacing differently, and change the PicoBOB configuration to match the ZedBoard to try to solve the issue.

JoarGjersund commented on Feb 9 • edited

Zedboard does NOT currently successfully interface with the RGB camera. The firmware/software running on the HIL is an unknown version not made with the petalinux build script in opu-system master branch. No need to change anything in the tcl script yet. The issue might be with the current petalinux build setup. Let me first go through the documentation marion made last semester and check that nothing is overlooked.

JoarGjersund commented on Feb 9

solved in `f8ccd49` for zedboard.

JoarGjersund closed this on Feb 9

magne-hov commented on Feb 9

Is this solved only for Zedboard or also Picozed?

JoarGjersund commented on Feb 9 • edited

Zedboard will currently report whenever usb is connected or not, however rgb init over CAN does currently not seem to work. Looking into this now. The PicoBoB v3 does currently not respond to the USB beeing connected. Trying to figure out if it is a software or hardware bug.

JoarGjersund reopened this on Feb 9

magne-hov commented on Feb 9

> rgb init over CAN does currently not seem to work

Not sure if it makes a difference, but have you tried the latest `rgb-service` branch?

JoarGjersund commented on Feb 9

Nope, not yet. when building it is only the opu-services binary that must be updated or are there any other files?

magne-hov commented on Feb 9

opu-services should be the only program required for the payload

magne-hov commented on Feb 9

@**magnudan** have you been able to contact usb camera with AVNET presets for FMC board?

magnudan commented on Feb 10

No, I have not been able to do that from the FMC

JoarGjersund commented on Feb 10 • edited

This bug is finnaly solved in `703a92d`
What I did was

1. change MIO pins for usb to pullup disabled, and speed fast.
2. add nodes to device tree which enables usb0 and set it as a host. (device tree was found here: https://github.com/Xilinx/meta-xilinx/blob/master/meta-xilinx-bsp/recipes-bsp/device-tree/files/picozed-zynq7.dts) Dont ask me why AVNET does not include this in their bsp.

👍 2 😄 1 🎉 2 ❤️ 1 🚀 1 👀 1

JoarGjersund commented on Feb 10 • edited

Everything works now (#40). However, the startup script is currently running before the drivers are set up so you need to press ctrl+C after the booting has finnished and opu-services has started to continue setting up the rest of the drivers. Afteer everything is set up you need to login with username/password root/root and then run ./opu-services 12 can0. After this you probably also need to turn on and off the 5v power supply to the vbus.
I will now mark this issue as closed, since the usb now is correctly setup, although the above issues should also be resolved.

JoarGjersund closed this on Feb 10

Assignees

JoarGjersund
magnudan
DennisNTNU
jlgarrett

Labels

bug

Projects

None yet

Milestone

No milestone

Linked pull requests

Successfully merging a pull request may close this issue.

None yet

5 participants

Pin issue

AS opu-system Issue 60

Edit | New issue

Jump to bottom

Static IP not working #60

Closed **magnudan** opened this issue on Mar 19 · 11 comments

Labels bug

Projects SW kanban board

magnudan commented on Mar 19

Using the boot files created by `$ sudo ./build_all 16 -zed`, the board did not get the ip-address 129.241.2.36.

Output from `# ip addr` on the ZedBoard:

```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen
1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: can0: <NOARP,ECHO> mtu 16 qdisc noop state DOWN group default qlen 10
    link/can
3: eth0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc pfifo_fast state DOWN group
default qlen 1000
    link/ether 00:0a:35:00:1e:53 brd ff:ff:ff:ff:ff:ff
4: sit0@NONE: <NOARP> mtu 1480 qdisc noop state DOWN group default qlen 1000
    link/sit 0.0.0.0 brd 0.0.0.0
```

Contents of `/etc/network/interfaces`:

```
# /etc/network/interfaces -- configuration file for ifup(8), ifdown(8)

# The loopback interface
auto lo
iface lo inet loopback

# Wireless interfaces
iface wlan0 inet dhcp
        wireless_mode managed
        wireless_essid any
        wpa-driver wext
        wpa-conf /etc/wpa_supplicant.conf

iface atml0 inet dhcp

# Wired or wireless interfaces
auto eth0
```

```
iface eth0 inet dhcp
iface eth1 inet dhcp

# Ethernet/RNDIS gadget (g_ether)
# ... or on host side, usbnet and random hwaddr
iface usb0 inet static
        address 192.168.7.2
        netmask 255.255.255.0
        network 192.168.7.0
        gateway 192.168.7.1

# Bluetooth networking
iface bnep0 inet dhcp
```

Serial output from startup script:

```
################################# HYPSO STARTUP SCRIPT BELOW
################################
Error: either "local" is duplicate, or "eth0" is a garbage.
mount: mounting /dev/mmcblk1p2 on /media/emmc failed: No such file or directory
Setting up swapspace version 1, size = 1073737728 bytes
mkswap: can't open '/media/emmc/swapfile': No such file or directory
random: fast init done
Adding 1048572k swap on /media/sd/swapfile.  Priority:-2 extents:1 across:1048572k SS
/usr/bin/ueyesetup: line 80: file: command not found
/usr/bin/ueyesetup: line 124: getent: command not found
Info: Group 'ueye' created.
/usr/bin/ueyesetup: line 140: getent: command not found
Info: User 'ueyed' created.
Info: Config /etc/ids/ueye/ueyeusbd.conf created.
Info: Config /etc/ids/ueye/ueyeapimachine.conf created.
Info: Run directory '/run/ueyed' created.
Info: Autostart for USB driver enabled.
/usr/bin/ueyesetup: line 80: file: command not found
/usr/bin/ueyesetup: line 124: getent: command not found
Info: Group 'ueye' created.
/usr/bin/ueyesetup: line 140: getent: command not found
Info: User 'ueyed' created.
Info: Config /etc/ids/ueye/ueyeethd.conf created.
Info: Run directory '/run/ueyed' created.
Info: Autostart for ETH driver enabled.
Starting ueyeethd... IDS ueyeethd  4.93.1192 32bit build Jan 16 2020 07:10:15 ( cmdl
cap hll scq )
ps: invalid option -- 'p'
BusyBox v1.29.2 (2020-03-19 10:27:28 UTC) multi-call binary.

Usage: ps

Show list of processes

        w       Wide output
        l       Long output
ueyeethd is not running but pidfile exists.
Starting ueyeusbd... IDS ueyeusbd  4.93.1192 32bit build Jan 16 2020 07:10:15 ( cmdl
cap scq )
ps: invalid option -- 'p'
BusyBox v1.29.2 (2020-03-19 10:27:28 UTC) multi-call binary.

Usage: ps
```

```
Show list of processes

        w       Wide output
        l       Long output
ueyeusbd is not running but pidfile exists.
flash_eraseall has been replaced by `flash_erase <mtddev> 0 0`; please use it
Erasing 64 Kibyte @ 0 --  0 % complete flash_erase:  Cleanmarker Updated.
Erasing 64 Kibyte @ 10000 -- 50 % complete flash_erase:  Cleanmarker Updated.
Erasing 64 Kibyte @ 10000 -- 100 % complete
char_device: loading out-of-tree module taints kernel.
Device registered correctly with major number 244
Device allocated memory at 0xC9000000
Device registered correctly with major number 243
Device allocated memory at 0xCFFFF000
Starting syslogd/klogd: done
Starting tcf-agent: OK

PetaLinux 2019.1 728699e-primary /dev/ttyPS0
```

magnudan added the `bug` label on Mar 19

DennisNTNU commented on Mar 19

Two things:

1. The `/etc/network/interfaces` configures the usb0 interface to have a static ip, but it should be the `eth0` interface.
2. The first printout from the start up script is: `Error: either "local" is duplicate, or "eth0" is a garbage.` indicates some issue with an `ip` command.

The ip commands in the startup script on the master branch are

```
ip addr flush dev eth0
ip addr add 129.241.2.42/23 eth0
```

These commands should work, because they do on the lidsat picobob (it always gets the ip 129.241.2.42). Maybe this is an issue with the zedboard? Given the error message, eth0 isn't found so maybe the commands are being executed before eth0 is initialized?

Also, using both the startup script and the `/etc/network/interfaces` file is probably redundant and one may may overwrite the other. Can you try removing the ip commands from the startup script and modifying the file to set a static ip to the `eth0` interface?

Since the output from `ip addr` shows `eth0` to be down, can you try entering `ip link set eth0 up`, and seeing whether it gets an ip address?

evelynlimore added this to **In progress** in **SW kanban board** on Mar 19

magnudan commented on Mar 19

`ip link set eth0 up` does not change anything when printing `ip addr`

magnudan commented on Mar 19

I don't think the file `/etc/network/interfaces` is used in the build process. I thought it might be useful to know the contents, so I added the prints of `cat /etc/network/interfaces` .

JoarGjersund commented on Mar 19 • edited

you are right. the static ip is set in the startup script. How does you startup script look?

check it out by typing cat /etc/init.d/myapp-init

1

DennisNTNU commented on Mar 19

Can you try performing the startup script commands manually? i.e.
`ip addr flush dev eth0` followed by `ip addr add 129.241.2.42/23 eth0` and then `ip link set eth0 up` again for good measure?

magnudan commented on Mar 19

`cat /etc/init.d/myapp-init` :

```
root@10df715-primary:~# cat /etc/init.d/myapp-init
#!/bin/sh
### BEGIN INIT INFO
# Provides:        myapp-init
# Required-Start: $ALL
```

```
# Should-Start:
# Required-Stop:
# Should-Stop:
# Default-Start:  2 3 5
# Default-Stop:
# Description:    Linux Startup Script
### END INIT INFO
start ()
{
        echo "################################# HYPSO STARTUP SCRIPT BELOW
################################"

        #Set static ip address.
        ip addr flush dev eth0
        ip addr add 129.241.2.36/23 eth0

        #mount sd-card (read and write)
        mkdir -m 755 /media/sd
        mount /dev/mmcblk0p1 /media/sd || mount /dev/mmcblk0 /media/sd



        #mount eMMC
        mkdir -m 755 /media/emmc
        mount /dev/mmcblk1p2 /media/emmc

        #enable memory swap.
        mkswap /media/sd/swapfile || test -f /media/emmc/swapfile || dd if=/dev/zero
of=/media/emmc/swapfile bs=1024 count=1048576 && mkswap /media/emmc/swapfile
        swapon /media/sd/swapfile || swapon /media/emmc/swapfile

        #increase available size of rootfs to include available swap memory.
        mount -o remount,size=1G /

        #mount gpio for sd select.
        echo 960 > /sys/class/gpio/export


        #extract software to rootfs and delete compressed folder.
        tar -xf /software.tar.xz -C /
        rm /software.tar.xz

        # run ueye setup script
        /usr/bin/ueyesetup -i usb
        /usr/bin/ueyesetup -i eth



        #start eueye ethernet & usb
        /etc/init.d/ueyeethdrc start
        /etc/init.d/ueyeusbdrc start

        # reset bootcounter (should probably be executed after we have established
communication.)
        flash_eraseall -j /dev/mtd1

        #start char_devices kernel module
        insmod /home/root/char_device.ko

        exit 1
        #start opu-services from image.
```

```
        /home/root/opu-services 12 can0

        #if above is exited. start opu-services from sd card.
        /media/sd/opu-services 12 can0
}

stop ()
{


echo "Bye, bye hypso."
}
restart()
{
stop
start
}
case "$1" in
start)
start; ;;
stop)
stop; ;;
restart)
restart; ;;
*)
echo "Usage: $0 {start|stop|restart}"
exit 1
esac
exit $?



root@10df715-primary:~# ip addr flush dev eth0
root@10df715-primary:~# ip addr add 129.241.2.36/23 eth0
Error: either "local" is duplicate, or "eth0" is a garbage.
root@10df715-primary:~# ip link set eth0 up
root@10df715-primary:~# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen
1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: can0: <NOARP,ECHO> mtu 16 qdisc noop state DOWN group default qlen 10
    link/can
3: eth0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc pfifo_fast state DOWN group
default qlen 1000
    link/ether 00:0a:35:00:1e:53 brd ff:ff:ff:ff:ff:ff
4: sit0@NONE: <NOARP> mtu 1480 qdisc noop state DOWN group default qlen 1000
    link/sit 0.0.0.0 brd 0.0.0.0
root@10df715-primary:~#
```

🚀 1 ☺

JoarGjersund commented on Mar 19

are you sure the ethernet is connected?

 **magnudan** commented on Mar 19

I managed to set the IP manually. The only thing missing from Dennis' suggestion was a `dev` . The commands I used were

```
# ip addr flush dev eth0
# ip addr 129.241.2.36/23 dev eth0
# ip link set eth0 up
```

I will test updating the startup script once I have set the right permissions as per issue #62

 **magnudan** commented on Mar 20

The ZedBoard boots with a static IP address when using `dev` as described above.

 1 

 **magnudan** closed this on Mar 20

 **SW kanban board** automation moved this from **In progress** to **Done** on Mar 20

 **JoarGjersund** reopened this on Mar 20

 **SW kanban board** automation moved this from **Done** to **In progress** on Mar 20

 **JoarGjersund** commented on Mar 20

the commands you manually entered are the same as the ones in the startup script. It is critical that you dont need to enter those commands manually. Is the static IP address set corretly after a reboot of the system? If not we need to investigate this further.

JoarGjersund commented on Mar 20

oh. I see the mistake in the startup script. Will make a commit to master solving this.

👍 1

JoarGjersund added a commit that referenced this issue on Mar 20

`Update startup_script.sh` ··· Verified `6c128a1`

JoarGjersund closed this on Mar 20

SW kanban board automation moved this from **In progress** to **Done** on Mar 20

Assignees

No one—assign yourself

Labels

bug

Projects

SW kanban board

Done

Milestone

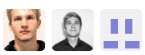
No milestone

Linked pull requests

Successfully merging a pull request may close this issue.

None yet

3 participants




Pin issue

AT opu-system Issue 62

Edit | New issue

Jump to bottom

My permissions are not set right? #62

Closed **magnudan** opened this issue on Mar 19 · 4 comments

Labels question

magnudan commented on Mar 19

Tried to do a clean clone and setup of this repository:

- `git clone git@github.com:NTNU-SmallSat-Lab/opu-system.git --recursive`
- `sudo ./setup-petalinux-docker`
- `sudo ./setup-hypso-sw-docker`

Then I executed `./build_all 16 -zed`. I suspect that I should have changed permissions for the repository, as a lot of commands in various scripts fail because they don't have permission. Is there a quick fix for this? The full log is added below.

```
magnus@hypso-desktop:~/Repos/opu-system$ ./build_all 16 -zed
Building with IP-Adress: 129.241.2.36/23
rm: cannot remove 'build': No such file or directory
mkdir: cannot create directory 'build': Permission denied
Makefile:18: recipe for target 'build/arm/Makefile' failed
make: *** [build/arm/Makefile] Error 1
cp: cannot stat 'hypso-sw/build/arm/opu-services': No such file or directory
etc/
etc/udev/
etc/udev/rules.d/
etc/udev/rules.d/99-ueyeusbd.rules
etc/init.d/
etc/init.d/ueyeusbdrc
etc/init.d/ueyeethdrc
home/
home/root/
home/root/opu-services
home/root/char_device.ko
lib/
lib/libpng16.so.16.37.0
lib/libusb-1.0.so.0.2.0
lib/libgomp.so.1
lib/libusb-1.0.so.0
lib/libgcc_s.so.1
lib/libudev.so.1.5.0
lib/libudev.so.1
lib/libstdc++.so.6
lib/libpng16.so.16
lib/libjpeg.so.62.2.0
lib/libjpeg.so.62
lib/firmware/
```

```
lib/firmware/ids/
lib/firmware/ids/ueye/
lib/firmware/ids/ueye/usb3_addon/
lib/firmware/ids/ueye/usb3_addon/usb3_cp_ddr_1v8_parallel.fwc
lib/firmware/ids/ueye/usb3_addon/usb3_ml_le_1v8_parallel.fwc
lib/firmware/ids/ueye/usb3_addon/usb3_le2_1v8_lvds_aptina_2phy.fwc
lib/firmware/ids/ueye/usb3_addon/usb3_cp_1v8_parallel.fwc
lib/firmware/ids/ueye/usb3_addon/usb3_le2_1v8_lvds_sony_i2c.fwc
lib/firmware/ids/ueye/usb3_addon/usb3_cp_1v8_lvds_sony.fwc
lib/firmware/ids/ueye/usb3_addon/usb3_cp_no_ddr_1v8_lvds_aptina_2phy.fwc
lib/firmware/ids/ueye/usb3_addon/usb2_se_3v3_parallel.fwc
lib/firmware/ids/ueye/usb3_addon/usb3_cp_3v3_parallel.fwc
lib/firmware/ids/ueye/usb3_addon/usb3_cp_no_ddr_3v3_lvds_onsemi.fwc
lib/firmware/ids/ueye/usb3_addon/usb3_cp_1v8_lvds_aptina.fwc
lib/firmware/ids/ueye/usb3_addon/usb3_le2_3v3_lvds_onsemi.fwc
lib/firmware/ids/ueye/usb3_addon/usb3_le2_1v8_lvds_sony_spi.fwc
lib/firmware/ids/ueye/usb3_addon/usb3_cp_ddr_1v8_lvds_aptina_2phy.fwc
lib/firmware/ids/ueye/usb3_addon/usb3_ml_le_3v3_parallel.fwc
lib/firmware/ids/ueye/usb3_addon/usb3_cp_ddr_3v3_parallel.fwc
lib/firmware/ids/ueye/usb3_addon/usb3_le2_1v8_lvds_sony.fwc
lib/firmware/ids/ueye/usb3_addon/usb3_cp_no_ddr_3v3_lvds_cmosis.fwc
lib/firmware/ids/ueye/usb3_addon/usb3_cp_no_ddr_1v8_lvds_aptina.fwc
lib/firmware/ids/ueye/usb3_addon/usb3_cp_ddr_3v3_lvds_cmosis.fwc
lib/firmware/ids/ueye/usb3_addon/usb3_cp_ddr_1v8_lvds_aptina.fwc
lib/firmware/ids/ueye/usb3_addon/usb3_cp_no_ddr_1v8_lvds_sony.fwc
lib/firmware/ids/ueye/usb3_addon/usb3_cp_ddr_1v8_lvds_sony.fwc
lib/firmware/ids/ueye/usb3_addon/usb3_cp_no_ddr_1v8_parallel.fwc
lib/firmware/ids/ueye/usb3_addon/usb3_cp_no_ddr_3v3_parallel.fwc
lib/firmware/ids/ueye/usb3_addon/usb3_cp_3v3_lvds_cmosis.fwc
lib/firmware/ids/ueye/usb3_addon/usb3_cp_ddr_3v3_lvds_onsemi.fwc
lib/firmware/ids/ueye/eth/
lib/firmware/ids/ueye/eth/te_A7_est.fwc
lib/firmware/ids/ueye/eth/leet18_est.fwc
lib/firmware/ids/ueye/eth/gige_cp_lvds_sony_5260.fwc
lib/firmware/ids/ueye/eth/te_A4_est.fwc
lib/firmware/ids/ueye/eth/sep_est.fwc
lib/firmware/ids/ueye/eth/le18_est.fwc
lib/firmware/ids/ueye/eth/le30_est.fwc
lib/firmware/ids/ueye/eth/gige_cp2_c4_ddr_lvds_cmosis_3v3.fwc
lib/firmware/ids/ueye/eth/gige_cp2_c4_ddr_lvds_onsemi_3v3.fwc
lib/firmware/ids/ueye/eth/he_eru.fwc
lib/firmware/ids/ueye/eth/gige_cp2_c4_ddr_parallel_1v8.fwc
lib/firmware/ids/ueye/eth/gige_cp2_c4_ddr_lvds_sony_1v8.fwc
lib/firmware/ids/ueye/eth/cp18_est.fwc
lib/firmware/ids/ueye/eth/cp30_est.fwc
lib/firmware/ids/ueye/eth/gige_cp_lvds_cmosis_5370.fwc
lib/firmware/ids/ueye/eth/se_est.fwc
lib/firmware/ids/ueye/eth/gige_cp2_c4_ddr_parallel_3v0.fwc
lib/firmware/ids/ueye/usb3/
lib/firmware/ids/ueye/usb3/cx3_firmware.fwc
lib/firmware/ids/ueye/usb3/fx3_ddr_firmware.fwc
lib/firmware/ids/ueye/usb3/fx3_firmware.fwc
lib/firmware/ids/ueye/usb3/fx2g2_firmware.fwc
usr/
usr/bin/
usr/bin/idscameramanager
usr/bin/ueyeusbd
usr/bin/ueyeethd
usr/bin/ueyemakeconfig
usr/bin/ueyereport
usr/bin/ueyesetip
```

```
usr/bin/ueyesetup
usr/bin/ueyesetid
usr/lib/
usr/lib/libueye_api.so.4.93
usr/lib/libueye_api.so.1
usr/lib/libueye_api.so
usr/include/
usr/include/ueye.h
usr/include/ueye_deprecated.h
PetaLinux environment set to '/opt/Xilinx/petalinux'
WARNING: /bin/sh is not bash!
bash is PetaLinux recommended shell. Please set your default shell to bash.
INFO: Checking free disk space
INFO: Checking installed tools
INFO: Checking installed development libraries
INFO: Checking network and other services
WARNING: No tftp server found - please refer to "PetaLinux SDK Installation Guide" for
its impact and solution
INFO: Create project: zed-primary
mkdir: cannot create directory ‘/home/hypso/petalinux/projects/zed-primary’: Permission
denied
rsync: mkdir "/home/hypso/petalinux/projects/zed-primary" failed: Permission denied
(13)
rsync error: error in file IO (code 11) at main.c(675) [Receiver=3.1.2]
rsync: mkdir "/home/hypso/petalinux/projects/zed-primary" failed: Permission denied
(13)
rsync error: error in file IO (code 11) at main.c(675) [Receiver=3.1.2]
mkdir: cannot create directory ‘/home/hypso/petalinux/projects/zed-primary/project-
spec/meta-user/recipes-apps/’: No such file or directory
rsync: mkdir "/home/hypso/petalinux/projects/zed-primary/project-spec/meta-
user/recipes-apps" failed: No such file or directory (2)
rsync error: error in file IO (code 11) at main.c(675) [Receiver=3.1.2]
find: ‘/home/hypso/petalinux/projects/zed-primary’: No such file or directory
environment: line 656: pushd: /home/hypso/petalinux/projects/zed-primary: No such file
or directory
mkdir: cannot create directory ‘.petalinux’: Permission denied
environment: line 1269: .gitignore: Permission denied
environment: line 661: popd: directory stack empty
INFO: New project successfully created in /home/hypso/petalinux/projects/zed-primary
environment: line 304:
/home/hypso/petalinux/projects/.petalinux/usage_statistics_token: No such file or
directory
cat: /home/hypso/petalinux/projects/.petalinux/usage_statistics_token: No such file or
directory
environment: line 304:
/home/hypso/petalinux/projects/.petalinux/usage_statistics_token: No such file or
directory
cat: /home/hypso/petalinux/projects/.petalinux/usage_statistics_token: No such file or
directory
Building for ZedBoard
./makeproject: line 119: cd: zed-primary: No such file or directory
sed: can't read project-spec/configs/config: No such file or directory
./makeproject: line 123: project-spec/configs/config: No such file or directory
ERROR: Failed to locate the Vivado export to SDK directory, please make sure the
directory directory exists!
Configures the project or the specified component with menuconfig.

Usage:
  petalinux-config [options] {--component <COMPONENT> |--get-hw-description[=SRC]}

Options:
```

```
  -h, --help                      show function usage
  -p, --project <PROJECT>         path to PetaLinux SDK project.
                                  default is the working project
  --oldconfig,--silentconfig      takes the default configuration and skips the GUI.
  -c, --component <COMPONENT>     Specify the component
                                  If no component is specified, it will do
                                  top level project configuration .
                                  If you specify a component,it will
                                  configure it with menuconfig and saves
                                  user's config fragments in meta-user.
                                  E.g. -c rootfs, -c busybox
  --get-hw-description [SRC]      get hardware description.
                                  if [SRC] is specified, look in that
                                  location for an Vivado export to SDK directory.
                                  Otherwise, this MUST be run from
                                  WITHIN the vivado export to SDK directory.
  --defconfig [DEFCONFIG_TARGET]  defconfig the specified component.
                                  It applies to kernel and u-boot.
  -v, --verbose                   verbose mode

Note: There is no validation for configurable components. User can provide
      any component, bitbake will throw error for invalid components.

cp: cannot stat '../system-user_zedboard.dtsi': No such file or directory
zedboard_hw loaded
./makeproject: line 140: project-spec/meta-user/recipes-bsp/u-boot/files/platform-
top.h: No such file or directory
sed: can't read project-spec/meta-user/recipes-bsp/u-boot/files/platform-top.h: No such
file or directory
./makeproject: line 157: project-spec/meta-user/recipes-bsp/u-boot/files/platform-
top.h: No such file or directory
./makeproject: line 158: project-spec/meta-user/recipes-bsp/u-boot/files/platform-
top.h: No such file or directory
./makeproject: line 159: project-spec/meta-user/recipes-bsp/u-boot/files/platform-
top.h: No such file or directory
./makeproject: line 160: project-spec/meta-user/recipes-bsp/u-boot/files/platform-
top.h: No such file or directory
./makeproject: line 161: project-spec/meta-user/recipes-bsp/u-boot/files/platform-
top.h: No such file or directory
./makeproject: line 167: project-spec/meta-user/recipes-bsp/u-boot/files/platform-
top.h: No such file or directory
./makeproject: line 168: project-spec/meta-user/recipes-bsp/u-boot/files/platform-
top.h: No such file or directory
./makeproject: line 169: project-spec/meta-user/recipes-bsp/u-boot/files/platform-
top.h: No such file or directory
ERROR: You are not inside a PetaLinux project. Please specify a PetaLinux project!
This command creates a new PetaLinux Project or component

Usage:
  petalinux-create [options] <-t|--type <TYPE> <-n|--name <COMPONENT_NAME>

Required:
  -t, --type <TYPE>                    Available type:
                                         * project : PetaLinux project
                                         * apps    : Linux user application
                                         * modules : Linux user module
  -n, --name <COMPONENT_NAME>          specify a name for the component or
                                       project. It is OPTIONAL to create a
                                       PROJECT. If you specify source BSP when
                                       you create a project, you are not
                                       required to specify the name.
```

```
Options:
  -p, --project <PROJECT>              specify full path to a PetaLinux project
                                       this option is NOT USED for PROJECT CREATION.
                                       default is the working project.
  --force                              force overwriting an existing component
                                       directory.
  -h, --help                           show function usage
  --enable                             this option applies to all types except
                                       project.
                                       enable the created component

Options for apps:
  --template <TEMPLATE>                <c|c++|autoconf|install>
                                       c   : c user application(default)
                                       c++ : c++ user application
                                       autoconf: autoconf user application
                                       install: install data only
  -s, --source <SOURCE>                valid source name format:
                                         *.tar.gz, *.tgz, *.tar.bz2, *.tar,
                                         *.zip, app source directory


Create an app and enable it:
  $ petalinux-create -t apps -n myapp --enable
The application "myapp" will be created with c template in:
  <PROJECT>/project-spec/meta-user/recipes-apps/myapp

cp: cannot stat '../software/startup_script.sh': No such file or directory
cp: cannot stat '../software/rootfs_config.bb': No such file or directory
cp: cannot stat '../software/software.tar.xz': No such file or directory
sed: can't read project-spec/meta-user/recipes-apps/myapp-init/files/myapp-init: No
such file or directory
sed: can't read project-spec/meta-user/recipes-bsp/u-boot/files/platform-top.h: No such
file or directory
sed: can't read project-spec/meta-user/recipes-apps/myapp-init/files/myapp-init: No
such file or directory
sed: can't read project-spec/configs/config: No such file or directory
sed: can't read project-spec/configs/config: No such file or directory
sed: can't read project-spec/configs/config: No such file or directory
sed: can't read project-spec/configs/config: No such file or directory
WARNING: --oldconfig will be deprecated in the feature releases,Please use --
silentconfig.
ERROR: You are not inside a PetaLinux project. Please specify a PetaLinux project!
Configures the project or the specified component with menuconfig.

Usage:
  petalinux-config [options] {--component <COMPONENT> |--get-hw-description[=SRC]}

Options:
  -h, --help                      show function usage
  -p, --project <PROJECT>         path to PetaLinux SDK project.
                                  default is the working project
  --oldconfig,--silentconfig      takes the default configuration and skips the GUI.
  -c, --component <COMPONENT>     Specify the component
                                  If no component is specified, it will do
                                  top level project configuration .
                                  If you specify a component,it will
                                  configure it with menuconfig and saves
                                  user's config fragments in meta-user.
                                  E.g. -c rootfs, -c busybox
  --get-hw-description [SRC]      get hardware description.
```

```
                                if [SRC] is specified, look in that
                                location for an Vivado export to SDK directory.
                                Otherwise, this MUST be run from
                                WITHIN the vivado export to SDK directory.
  --defconfig [DEFCONFIG_TARGET]  defconfig the specified component.
                                It applies to kernel and u-boot.
  -v, --verbose                 verbose mode

Note: There is no validation for configurable components. User can provide
      any component, bitbake will throw error for invalid components.

sed: can't read project-spec/configs/rootfs_config: No such file or directory
sed: can't read project-spec/configs/rootfs_config: No such file or directory
sed: can't read project-spec/configs/rootfs_config: No such file or directory
sed: can't read project-spec/configs/config: No such file or directory
sed: can't read project-spec/configs/config: No such file or directory
sed: can't read project-spec/configs/config: No such file or directory
./makeproject: line 93: project-spec/meta-plnx-generated/recipes-
kernel/linux/configs/plnx_kernel.config: No such file or directory
./makeproject: line 94: project-spec/meta-plnx-generated/recipes-
kernel/linux/configs/plnx_kernel.config: No such file or directory
./makeproject: line 98: project-spec/meta-plnx-generated/recipes-
kernel/linux/configs/plnx_kernel.config: No such file or directory
./makeproject: line 99: project-spec/meta-plnx-generated/recipes-
kernel/linux/configs/plnx_kernel.config: No such file or directory
./makeproject: line 100: project-spec/meta-plnx-generated/recipes-
kernel/linux/configs/plnx_kernel.config: No such file or directory
sed: can't read project-spec/configs/config: No such file or directory
ERROR: You are not inside a PetaLinux project. Please specify a PetaLinux project!
Builds the project or the specified components.

Usage:
  petalinux-build [options]

Required:

Options:
  -h, --help                        show function usage
  -p, --project <PROJECT>           path to PetaLinux SDK project.
                                    Default is working project.
  -c, --component <COMPONENT>       Specify the component
                                    it will build the specified component and its
dependencies
                                    E.g. -c rootfs
                                    E.g. -c myapp
  -x, --execute <tasks of bitbake>  Specify a bitbake task of the component
                                    To know the list tasks for a component:
                                    E.g. -x do_listtasks
  -f, --force                       Force run a specific task ignoring the stamps
                                    Force run has to be for a component or its tasks
                                    E.g. -c myapp -f
                                    E.g. -c myapp -x compile -f
  -v, --verbose                     Show compile messages verbose mode
  -s, --sdk                         Build SDK ==> do_populate_sdk
  -b, --buildfile  <.bb recipe>     Execute tasks from a specific .bb recipe directly.
                                    WARNING: Does not handle any dependencies from
other recipes.

ERROR: You are not inside a PetaLinux project. Please specify a PetaLinux project!
ERROR: You are not inside a PetaLinux project. Please specify a PetaLinux project!
Commmand "petalinux-package --boot" packages boot file
```

```
Usage:
  petalinux-package --boot [options]

Options for boot image package:
  -h|--help                   Display help messages
  --force                     Force overwrite the boot binary image
  --format <FORMAT>           Avaiable formats:
                               For Zynq:
                                * BIN (default): generate BIN file to be put to
                                  Flash or SD for Zynq to boot from it.
                               For MicroBlaze:
                                * MCS: generate MCS file
                                * DOWNLOAD.BIT: Merges the fs-boot into the FPGA
                                                bitstream by mapping the ELF data
                                                onto the memory map information (MMI)
                                                for the block RAMs in the design
  --fsbl <FSBL_ELF>           Path to FSBL ELF image location.
                               For Zynq:
                                Default is images//zynq_fsbl.elf
                               For ZynqMP:
                                Default is images//zynqmp_fsbl.elf
                               For MicroBlaze:
                                Default is images//fs-boot.elf
                               To skip packing fsbl use --fsbl no or --fsbl none
  --fpga <BITSTREAM>          Path to FPGA bitstream image location
                                Default is: images//*.bit (The one copied from the HDF)
  --atf [<ATF_IMG>]           ZynqMP and Versal only. Path to ATF
                              To skip packing atf use --atf no or --atf none
  --dtb [<DTB_IMG>]           Path to DTB image location
  --pmufw [<PMUFW_ELF>]       Path to the PMUFW ELF location
                              Optional and applicable only for ZynqMP.
                              Default is: <PROJECT>/images/linux/pmufw.elf
                              To skip packing pmufw use --pmufw no
  --plm [<PLM_ELF>]           Path to the PLM ELF location
                              Optional and applicable only for Versal.
                              Default is: <PROJECT>/images/linux/plm.elf
                              To skip packing plm use --plm no
  --psmfw [<PMUFW_ELF>]       Path to the PSMFW ELF location
                              Optional and applicable only for Versal.
                              Default is: <PROJECT>/images/linux/psmfw.elf
                              To skip packing psmfw use --psmfw no
  --u-boot [<U_BOOT_IMG>]     Path to the u-boot image location.
                              Not valid for DOWNLOAD.BIT
                               For Zynq:
                                * path to the u-boot ELF image
                                  default: <PROJECT>/images//u-boot.elf
                               For MicroBlaze:
                                * path to the u-boot-s.bin image
                                  default: <PROJECT>/images//u-boot-s.bin
  --kernel [<KERNEL_ING>]     Path to the kernel image location(fitImage)
                              Not valid for DOWNLOAD.BIT
                              Default: <PROJECT>/images//image.ub
  --add <DATA_FILE>           Path to the data to add
  --add-cdo <CDO_FILE>        Path to the cdo bin file for Versal only.
  --offset <OFFSET>           Partition offset of previously specified data file, u-
boot
                              , or kernel image.

  --load <LOADADDR>           Load address for specified data file. The Ram address
where
                              to load the specified data file.
```

```
                                Ex:[ partition_type=raw, load=0x01000 ] <image>

  --mmi <MMI_FILE>            Bitstream MMI file. Valid for MicroBlaze only.
                              It will be used to generate the download.bit with
                              bootloader in the bram. Default will be the MMI
                              file in the same directory as the FPGA bitstream.
  --flash-size <FLASH_SIZE>   Flash size in MBytes of the PROM device is targeted.
                              It must be power of 2.
                              Only valid for MicroBlaze MCS format.
                              If this value is not specified. It will auto detect
                              the system flash configured from system config.
                              If it is parallel flash, it will auto detect flash
                              size.
                              If it is SPI flash, the default is 16 Mbytes.
  --flash-intf <INTERFACE>    Flash interface. Available options:
                              Only valid for MicroBlaze MCS
                               * SERIALx1
                               * SPIx1
                               * SPIx2
                               * SPIx4
                               * BPIx8
                               * BPIx16
                               * SMAPx8
                               * SMAPx16
                               * SMAPx32
                              If not specified, it will auto detect the system
                              flash configured from system config.
                              If it is parallel flash, it will auto detect flash
                              width. If it is SPI flash, the default will be SPIx1.
  -o, --output <PKGNAME>      Generated boot image name
  -p, --project <PROJECT>     PetaLinux SDK project location.
                              Default is the working project.
  --cpu <DESTINATE CPU>               ZynqMP only. destination CPU of the data file
  --file-attribute <DATA File ATTR>  Zynq/ZynqMP only. data file file-attribute
  --bif-attribute <ATTRIBUTE>         Zynq/ZynqMP only. name of BIF attribute
  --bif-attribute-value <VALUE>       Zynq/ZynqMP only. value of the attribute specified
by
                                      --attribute argument
  --fsblconfig <BIF FSBL CONFIG>      ZynqMP only. BIF fsbl config value
  --bif <BIF>                         Zynq/ZynqMP/Versal only. BIF file. It overrides
all
                                      other settings:
                                      --fsbl, --fpga, --u-boot, --add, --fsblconfig,
                                      --file-attribute, --bif-attribute and
                                      --bif-attribute-value.
  --boot-device <BOOT_DEV>            valid for BIN for Zynq and ZynqMP only.
                                      sd | flash
                                      default will be the one selected from system
select menu
                                      of boot image settings
  --bootgen-extra-args <ARGS>         Zynq/ZynqMP/Versal only. Extra arguments to be
passed while invoking
                                      bootgen command

EXAMPLES:
Package BOOT.BIN for Versal:
$ petalinux-package --boot --u-boot --dtb
  It will add all the dependencies into BOOT.BIN to boot u-boot.

$ petalinux-package --boot --plm <PLM_ELF> --psmfw <PSMFW_ELF> --u-boot --dtb
  It will generate BOOT.BIN,BOOT_bh.bin,pmc_cdo.bin and qemu_boot.bin in specified
```

```
directory.
  The default dtb load address will be 0x1000. To change the dtb load address Use below
command

$ petalinux-package --boot --plm <PLM_ELF> --psmfw <PSMFW_ELF> --u-boot --dtb --load
<load_address>
  It will generate a BOOT.BIN with specifed load address for dtb.

Package BOOT.BIN for ZynqMP:
$ petalinux-package --boot --u-boot
  It will add all the dependencies into BOOT.BIN to boot u-boot.

$ petalinux-package --boot --fsbl <FSBL_ELF> --fpga <BITSTREAM> --u-boot --pmufw
<PMUFW_ELF>
  It will generate a BOOT.BIN in your working directory with:
    * specified <BITSTREAM>
    * specified <FSBL_ELF>
    * specified < PMUFW_ELF > *
    * newly built u-boot image which is <PROJECT>/images/linux/u-boot.elf
Generate bitstream merged with fsbl
$ petalinux-package --boot --fsbl <FSBL_ELF> --fpga <BITSTREAM> --format DOWNLOAD.BIT
It will generate a download.bit in <PROJECT>/images/linux, with specified <BITSTREAM>
and <FSBL_ELF>.

ERROR: Please run the command inside a PetaLinux project, or use -p|--project option to
specify a PetaLinux project.
mkdir: cannot create directory ‘bootfiles’: Permission denied
./makeproject: line 189: cd: bootfiles: No such file or directory
mkdir: cannot create directory ‘zed’: Permission denied
cp: cannot stat 'zed-primary/images/linux/image.ub': No such file or directory
cp: cannot stat 'hw_config_files/zedboard_hw/bitstream.bit': No such file or directory
cp: cannot stat 'zed-primary/images/linux/BOOT.BIN': No such file or directory
dd: failed to open 'bootfiles/zed/swapfile': No such file or directory
Success!
1) copy files in petalinux/projects/bootfiles/pico/ to FAT-formatted SD-Card.
2) set bootmode to SD-Card and start up PicoBoB. wait 5 min
3) switch bootmode to QSPI.
```

magnudan added the **question** label on Mar 19

magnudan mentioned this issue on Mar 19

Static IP not working #60 Closed

JoarGjersund commented on Mar 19 • edited

this is weird and didn't happen for me. Are you sure you are not elevated when running build all? did you run sudo su earlier? Either way, looks like a permission issue. Does running

```
chmod --recursive 777 petalinux/projects/software/hypso-sw
```

solve the problem?

☺

DennisNTNU commented on Mar 19

You shouldn't have to deal with the permissions, when I clone a fresh opu-system repository to somewhere within my laptops home directory and run `./build-all 16 -zed`, I don't get any permission errors and hypso-sw is being build. Also, when I am running `./build-all 16 -zed`, inbetween

```
Fast-forwarded HEAD to 3270bddbfd060d5212632bcb4c1c564ccc6e7422.
Submodule path '../petalinux/projects/software/hypso-sw': rebased into
'3270bddbfd060d5212632bcb4c1c564ccc6e7422'
```

and

```
rm: cannot remove 'build': No such file or directory
```

I am getting asked to enter my sudo password, this does not happen for you?

👍 1 ☺

magnudan commented on Mar 20

@**DennisNTNU**: Yes, the first time I run the command, I get prompted for sudo password there.

@**JoarGjersund**: Setting 777 permissions for the hypso-sw directory solved the permission issues for hypso-sw. Setting 777 permissions for the entire `petalinux/projects` directory solved issues for the petalinux build as well.

Setting 777 works as a quick fix, but I feel that giving everyone r/w/x rights is not good.

Should permissions be set in one of the scripts, or is it better to write it in the Readme?

Or should it be solved some other way? I am not familiar with how to use groups, but could it be a solution?

☺

JoarGjersund commented on Mar 20

I also agree that setting 777 permissions is not a general solution, since it depends on who has access to your computer, etc. Since you are the only one so far experience this issue, I think we should just keep it as it is for now.

👍 1

JoarGjersund closed this on Mar 20

JoarGjersund mentioned this issue on Mar 20

Device table in origin is not updated #59 Closed

Assignees

No one—assign yourself

Labels

question

Projects

None yet

Milestone

No milestone

Linked pull requests

Successfully merging a pull request may close this issue.

None yet

3 participants

Pin issue

AU opu-system Pull Request 64

NTNU-SmallSat-Lab / **opu-system** Private

Fixmemory #64

Edit | Open with

Merged **JoarGjersund** merged 7 commits into `master` from `fixmemory` on Mar 20

Conversation 3 | Commits 7 | Checks 0 | Files changed 7 | +266 −13

drowzie commented on Mar 20

Fixes the problem with memory accesses to the reserved memory. This will reduce the amount of memory that OS can allocate from 0x40000000 to 0x3000000.

Updated the petalinux building to include the kernel driver during the process. Easier and less hassle with building it standalone. Allows to patch master branch with new updates to the kernel driver. These drivers will be placed in /lib/modules/(xilinx linux version)/extra/ and the startupscript has been updated to insmod it at boot.

@magnudan
dma_parameters.h is also cleaned where HYPSO-SW needs to edit.
SEND_PHYS address is updated to 0x30000000
RECEIVE_PHYS address is updated to 0x38000000

The driver will no longer allocates a struct but will allocate a uint16_t array.

```
    uint16_t * send_channel = mmap(0, CUBE_SIZE*sizeof(uint16_t),
                  PROT_READ | PROT_WRITE, MAP_SHARED, fd_send, SEND_PHYS_ADDR);
    uint16_t * receive_channel = mmap(0, CUBE_SIZE*sizeof(uint16_t),
                  PROT_READ | PROT_WRITE, MAP_SHARED, fd_receive,
RECEIVE_PHYS_ADDR);
```

drowzie added 4 commits on Mar 18

Fix for illegal memory access		1f588a6
Fixes the memory access issue. Seems like Linux does not respect the … ...		70ee621
Update makeproject	Verified	fb23961
Update makeproject	Verified	ab4312d

drowzie requested review from **JoarGjersund** and **magnudan** on Mar 20

Update kernel driver for uint16 pointers 8e2909f

JoarGjersund reviewed on Mar 20 — View changes

JoarGjersund left a comment • edited

Good work! I only have a few cocerns.

1. reducing the total memory accessible to the os should probably not be set this way since the default size will depend on the available memory and if ecc is enabled or not, so replacing that variable will not work if memory size is different (when ecc is enabled the default bankless size will be 0x20000000 ..
2. in the startup script the commit `6c128a1` which adds dev before eth0 when setting static ip is not included.
3. The device tree for picozed shold probably also be updated.

Reviewers
JoarGjersund
magnudan

Assignees
No one—assign yourself

Labels
None yet

Projects
None yet

Milestone
No milestone

Linked issues
Successfully merging this pull request may close these issues.
None yet

3 participants

magnudan approved these changes on Mar 20 View changes

JoarGjersund added 2 commits on Mar 20

`Update startup_script.sh` ··· Verified `542ff5e`

`Update makeproject` Verified `7ea2c11`

JoarGjersund commented on Mar 20

Is this tested and confirmed working for zedboard? If so I will merge it.

drowzie commented on Mar 20 • edited Author

> Is this tested and confirmed working for zedboard? If so I will merge it.

Yep, just did a test to access the memory region and it is working. No kernel panics.
Also now the driver will be seen in /proc/iomem of the used memory

```
root@8e2909f-primary:~# cat /proc/iomem
00000000-2fffffff : System RAM
  00008000-008fffff : Kernel code
  00a00000-00a544cf : Kernel data
30000000-3497897f : cubedma send
38000000-3c97897f : cubedma receive
e0001000-e0001fff : xuartps
e0008000-e0008fff : can@e0008000
e000a000-e000afff : gpio@e000a000
e000b000-e000bfff : ethernet@e000b000
e000d000-e000dfff : spi@e000d000
e0100000-e0100fff : mmc@e0100000
f8003000-f8003fff : dmac@f8003000
  f8003000-f8003fff : dmac@f8003000
f8005000-f8005fff : watchdog@f8005000
f8007000-f80070ff : devcfg@f8007000
f8007100-f800711f : adc@f8007100
f800c000-f800cfff : ocmc@f800c000
fffc0000-ffffffff : f800c000.ocmc
```

1

JoarGjersund merged commit `b0fc231` into `master` on Mar 20 Revert

JoarGjersund deleted the `fixmemory` branch on Mar 20 Restore branch

AV opu-system Issue 65

Edit | New issue

Jump to bottom

Cube DMA has "ecc mode"? #65

Closed **magnudan** opened this issue on Mar 20 · 2 comments

Labels question

magnudan commented on Mar 20

From the component `unpacker.vhd` in the Cube DMA project in the vivado directory: `ECC_MODE => "no_ecc",` (line 116). Does this mean that Cube DMA easily works with ecc enabled?

Someone™ should look into this

👍 1

magnudan added the question label on Mar 20

drowzie commented on Mar 22 • edited

Sadly this is only a MACRO made by xilinx when creating a FIFO.
macro for sync

magnudan commented on Mar 23

Damn. I got my hopes up

sivertba closed this on Mar 24

Assignees

No one—assign yourself

Labels

question

Projects

None yet

Milestone

No milestone

Linked pull requests

Successfully merging a pull request may close this issue.

None yet

3 participants

Pin issue

AW opu-system Issue 66

Edit | New issue

Jump to bottom

ZedBoard is unable to boot without swapfile #66

Closed **magnudan** opened this issue on Mar 24 · 17 comments

Labels bug

magnudan commented on Mar 24

Issue regarding master branch on commit 9d73cb3.

I tried to boot a ZedBoard with all files except `swapfile` after running `./build_all 11 -zed`. The boot was monitored over UART. Everything was normal until the prints froze at a point in the startup script:

```
################################ HYPSO STARTUP SCRIPT BELOW
################################
mount: mounting /dev/mmcblk1p2 on /media/emmc failed: No such file or directory
mkswap: can't open '/media/sd/swapfile': No such file or directory
```

This happened for both the ZedBoard I have at home and the ZedBoard in the HIL setup in the lab. Neither responded to UART, CAN or Ethernet.

magnudan added the bug label on Mar 24

JoarGjersund commented on Mar 24

I am not able to reproduce this since is do not have a zedboard available. I am also able to boot without a swapfile for picozed, so I suspect the issue might be related to smething else. @**drowzie** can you look at this?

JoarGjersund commented on Mar 24 • edited

I am curious if the issue still there in commit `9b23502` ? Also, could you provide me with the full log @**magnudan** ? starting from when you turn on power.

JoarGjersund commented on Mar 24

To avoid having to deal with this problem again I have added golden image fallback for zedboard per #67

❤️ 1

magnudan commented on Mar 24 • edited

This comment is still based on files made from commit 9d73cb3.

Full log:
boot_without_swapfile_9d73cb3.log

In my initial post, nothing was printed after what I included from the boot log. I haven't seen these messages before just now.

This is the boot log when the swapfile is included:
boot_with_swapfile_9d73cb3.log

magnudan commented on Mar 24 • edited

On commit 9b23502, the ZedBoard booted successfully without the swapfile:
boot_without_swapfile_9b23502.log

With swapfile:
boot_with_swapfile_9b23502.log

JoarGjersund added a commit that referenced this issue on Mar 24

`added check if emmc is mounted before creating swapfile. should solve #…` `e20f476`
…

JoarGjersund commented on Mar 24

I think the reason why it didnt work was because it is trying to create a file on a media that has not been mounted. (the emmc). Also explains why this haven't been a problem on picozed. @**magnudan** can you check if it works now?

magnudan commented on Mar 25

For this commit (e90cb38), the ZedBoard froze in both cases (with and without swapfile). I have included both logs as files to reduce scrolling time. Let me know if this is better or not :)

boot_with_swap_e90cb38.log
boot_without_swap_e90cb38.log

JoarGjersund commented on Mar 25

OK. then I suspect this has to do with the commit made by **@drowzie** `b0fc231`

can someone try commit `9b23502` and see if that works?

jonakor commented on Mar 25

I built commit `1496ea4` just now and it worked. Tht commit included **@drowzie** changes

JoarGjersund commented on Mar 25 • edited

So the issue seems to be traced down to `f41e70f` when the programming of fpga during booting was introduced. I suspect that the issue might be related to the memory configurations made by **@drowzie** and that those bugs are only showing when the fpga is programmed. One solution is to revert back to `1496ea4` , and use fpga manager to program the fpga after boot. another solution is to figure out exactly why the kernel hangs. I think reverting back to the working commit is easier. I suspect we might end up with some mysterous bugs whenever we program the fpga via fpga manager. for now just use this commit, and if you find out its working well revert the master (or simply undo the changes made in commit `f41e70f`) BUT, be aware that if something strange happens and the system freezes, the reason might be wrong memory configurations of the kernel.

jonakor commented on Mar 25

I built the commit `f41e70f` now and it runs! this one include the program FPGA during boot part #68 .

JoarGjersund commented on Mar 25

okey. revert back to commit `f41e70f` then

magnudan commented on Mar 26

This commit works for me as well. I will use this version on the SD card that will be used with the ZedBoard in the lab

jonakor commented on Mar 26

@**JoarGjersund** is the later commits enhancements or bug fixing. Could master be reverted to `f41e70f` ?

JoarGjersund commented on Mar 27

Before we revert,can someone try the latest commit after they have successfully booted the system once first? I suspect the issue might be with the uboot environment not getting erased when upgrading the bootloader. Also. if someone allready have it installed and fail to boot, does it help pressing enter during uboot and write `run eraseenv` in the u-boot terminal?

JoarGjersund added a commit that referenced this issue on Mar 27

`Update makeproject` ··· Verified `dccfc01`

magnudan commented on Mar 27

The new commit results in kernel panics in both cases. Logs attached

boot_with_eraseenv_dccfc01.log
regular_boot_dccfc01.log

JoarGjersund commented on Mar 30

moved issue to #73

JoarGjersund closed this on Mar 30

Assignees

No one—assign yourself

Labels

bug

Projects

None yet

Milestone

No milestone

Linked pull requests

Successfully merging a pull request may close this issue.

None yet

3 participants

Pin issue

AX opu-system Issue 76

Edit | New issue

Jump to bottom

Errors causing building boot image (build_all script) to fail? #76

Closed **DennisNTNU** opened this issue on Apr 1 · 15 comments

Assignees	
Labels	petalinux points=5
Projects	SW kanban board

DennisNTNU commented on Apr 1

When I try building a boot image my self on my laptop, the `build_all` script prints a few errors and warnings, the script returns after less than a minute instead of the usual longer amount, and the boot image is not created.

This is the console output of running the script as follows `./build_all 2 -dev` :

```
Building with IP-Adress: 129.241.2.42/23
cp: cannot stat 'vivado/projects/PicoBOB/PicoBOB.sdk/*': No such file or directory
cp: cannot stat 'vivado/projects/ZedBoard/ZedBoard.sdk/*': No such file or directory
PetaLinux environment set to '/opt/Xilinx/petalinux'
WARNING: /bin/sh is not bash!
bash is PetaLinux recommended shell. Please set your default shell to bash.
INFO: Checking free disk space
INFO: Checking installed tools
INFO: Checking installed development libraries
INFO: Checking network and other services
WARNING: No tftp server found - please refer to "PetaLinux SDK Installation Guide" for
its impact and solution
INFO: Create project: pico-primary
INFO: New project successfully created in /home/hypso/petalinux/projects/pico-primary
Building For PicoZed
INFO: Getting hardware description...
[INFO] generating Kconfig for project
[INFO] menuconfig project
ERROR: Failed to menu config project component
ERROR: Failed to config project.
ERROR: Get hw description Failed!.
picozed_hw loaded
sed: can't read project-spec/meta-plnx-generated/recipes-bsp/u-boot/configs/platform-
auto.h: No such file or directory
INFO: Create apps: myapp-init
INFO: New apps successfully created in /home/hypso/petalinux/projects/pico-
primary/project-spec/meta-user/recipes-apps/myapp-init
INFO: Enabling created component...
INFO: sourcing bitbake
INFO: oldconfig rootfs
INFO: myapp-init has been enabled
```

```
cp: cannot stat '../software/software.tar.xz': No such file or directory
INFO: Create apps: cubedma-test
INFO: New apps successfully created in /home/hypso/petalinux/projects/pico-
primary/project-spec/meta-user/recipes-apps/cubedma-test
INFO: Enabling created component...
INFO: sourcing bitbake
ERROR: Failed to source yocto environment
ERROR: However, failed to enable created component.
INFO: Create modules: cubedma
INFO: New modules successfully created in /home/hypso/petalinux/projects/pico-
primary/project-spec/meta-user/recipes-modules/cubedma
INFO: Enabling created component...
INFO: sourcing bitbake
ERROR: Failed to source yocto environment
ERROR: However, failed to enable created component.
./load_software: line 64: project-spec/meta-plnx-generated/recipes-
kernel/linux/configs/plnx_kernel.config: No such file or directory
./load_software: line 65: project-spec/meta-plnx-generated/recipes-
kernel/linux/configs/plnx_kernel.config: No such file or directory
./load_software: line 69: project-spec/meta-plnx-generated/recipes-
kernel/linux/configs/plnx_kernel.config: No such file or directory
./load_software: line 70: project-spec/meta-plnx-generated/recipes-
kernel/linux/configs/plnx_kernel.config: No such file or directory
./load_software: line 71: project-spec/meta-plnx-generated/recipes-
kernel/linux/configs/plnx_kernel.config: No such file or directory
[INFO] building project
[INFO] sourcing bitbake
ERROR: Failed to source bitbake
ERROR: Failed to build project
ERROR: Failed to package boot, FSBL file: images/linux/zynq_fsbl.elf doesn't exist.
mkdir: cannot create directory ‘bootfiles’: File exists
mkdir: cannot create directory ‘pico’: File exists
cp: cannot stat 'pico-primary/images/linux/image.ub': No such file or directory
cp: cannot stat 'pico-primary/images/linux/BOOT.BIN': No such file or directory
Success!
1) copy files in petalinux/projects/bootfiles/pico/ to FAT-formatted SD-Card.
2) set bootmode to SD-Card and start up PicoBoB. wait 5 min
3) switch bootmode to QSPI.
```

Another thing is that may indicate the underlying issue is that during step 9 of building of the petalinux docker container, is that I get three times the warning

```
dpkg: warning: failed to open configuration file '/root/.dpkg.cfg' for reading:
Permission denied
```

JoarGjersund commented on Apr 1

What commit? what files are in your hw_config_files/picozed_hw folder ? Did you run as sudo when running the set up petalinux docker script?

DennisNTNU commented on Apr 1

Commit should be the newest `57553d4` .
In the `hw_config_files/picozed_hw` are three files `bitstream.bit` , `pz7030_fmc2_2019_1.bsp` and `system.hdf`
I did use sudo when running setup_petalinux_docker: `sudo ./setup-petalinux-docker`

evelynlimore added this to **Backlog** in **SW kanban board** on Apr 2

evelynlimore assigned **drowzie** on Apr 2

evelynlimore added the `points=5` label on Apr 2

drowzie commented on Apr 2 • edited

I recommend cloning the repo again. I have had an issue once where "something" happened that caused an early crash in the build script.
`petalinux-config --get-hw-description=`
Seemed to not be successfull and recloning the repository seemed to remove any issue.

```
[INFO] menuconfig project
ERROR: Failed to menu config project component
ERROR: Failed to config project.
ERROR: Get hw description Failed!.
```

DennisNTNU commented on Apr 3

Recloning the repo and making sure I made no error solved the issue indeed. The error cause was that I ran the `build_all` script once with the wrong arguments (forgot to add `2 -dev`) and then ctrl+c'ed it. I cancelled it before it came to the part where I get the first `ERROR:` so I thought the cancelling could not be the cause.

DennisNTNU closed this on Apr 3

SW kanban board automation moved this from **Backlog** to **Done** on Apr 3

magnudan commented on Apr 17

I experienced the same issue, but re-cloning did not work for me. However, rebooting my computer did solve the problem

JoarGjersund commented on Apr 28

reopening this because it is a bug that keeps popping up.

JoarGjersund reopened this on Apr 28

JoarGjersund commented on Apr 28

Can also be solved by deleting and reinstalling the docker petalinux image. Also, I will add a check to make sure that everyone pass the minimum required number of arguments to the build_all script.

JoarGjersund added a commit that referenced this issue on Apr 28

`Update build_all` ··· Verified `8d44e79`

JoarGjersund closed this on Apr 28

DennisNTNU commented on May 6 • edited

I am still having this issue, and rebuilding the petalinux image didn't help. What I did in detail:

- `./build_all 2 -dev` on my laptop, gives the error
- `./build_all 2 -dev` on the new workstation PC 2, gives the error as well
- rebuilding petalinux image and then `./build_all 2 -dev` on the new workstation PC 2 same thing

All this has been done using fresh clones of the repository. I copied the .bsp file into the petalinux/projects/hw_config_files/picozed_hw folder after every clone.

magnudan reopened this on May 6

magnudan commented on May 6

What errors do you get?

I often have issues with permissions with files and folders that are used both inside and outside the petalinux docker image. Adding 777 permissions usually solve this, but it's not an elegant solution

magnudan added the **petalinux** label on May 6

DennisNTNU commented on May 6

The same as in the original description. The first error is the

```
ERROR: Failed to menu config project component
```

And other errors follow

magnudan commented on May 6

> I copied the .bsp file into the petalinux/projects/hw_config_files/picozed_hw folder after every clone.

Since commit 339905d, the .bsp file is not needed anymore. It's not a solution to this issue, but not having to copy for every fresh clone is nice

JoarGjersund commented on May 6

It is working for me, so it must be something with your setup. Perhaps something is wrong with the petalinux install file. You could try to reinstall the docker image with a new copy of the install file perhaps?

DennisNTNU commented on May 6

I did that on WS2, but not on my laptop, and its on my laptop that it works again. My problem was apparently that I cloned with the `--depth=1` option, which I thought just limits how many commits back in time you want to include in the cloning. I did that to make cloning a bit faster, to get only the most recent state of the repository, but apparently that breaks the building. I did not get it to work on WS2 due to other errors, permission errors.

JoarGjersund commented on May 6

Great! Then we can probably go ahead and close this issue again.

DennisNTNU closed this on May 6

magnudan commented on May 8

Hopefully, this is the last post to this issue...

@**JoarGjersund** found this post on a Xilinx forum suggesting that the SIZE OF THE TERMINAL might (does) cause kconfig to fail!

If you get the following error messages, try to make the terminal window larger.

```
PetaLinux environment set to '/opt/Xilinx/petalinux'
WARNING: /bin/sh is not bash!
bash is PetaLinux recommended shell. Please set your default shell to bash.
INFO: Checking free disk space
INFO: Checking installed tools
INFO: Checking installed development libraries
INFO: Checking network and other services
WARNING: No tftp server found - please refer to "PetaLinux SDK Installation Guide" for
its impact and solution
INFO: Create project: pico-primary
INFO: New project successfully created in /home/hypso/petalinux/projects/pico-primary
Building For PicoZed
INFO: Getting hardware description...
[INFO] generating Kconfig for project
```

```
[INFO] menuconfig project
ERROR: Failed to menu config project component
ERROR: Failed to config project.
ERROR: Get hw description Failed!.
```

Honestly, Xilinx... 😞

magnudan mentioned this issue on May 8

Fix issue with petalinux config crash if terminal window is too small #108

Closed

Assignees

drowzie

Labels

petalinux

points=5

Projects

SW kanban board

Done

Milestone

No milestone

Linked pull requests

Successfully merging a pull request may close this issue.

None yet

5 participants

Pin issue

Edit | New issue

Jump to bottom

Master branch is broken #85

Closed **magnudan** opened this issue on Apr 27 · 2 comments

Labels HIGH PRIORITY bug

magnudan commented on Apr 27

When trying to build boot images on master branch on commit e6fc2c1, there are errors that have to do with the Cube DMA kernel module.

```
NOTE: Executing RunQueue Tasks
ERROR: cubedma-1.0-r0 do_compile: oe_runmake failed
ERROR: cubedma-1.0-r0 do_compile: Function failed: do_compile (log file is located at /home/hyp
so/petalinux/projects/pico-primary/build/tmp/work/plnx_zynq7-xilinx-linux-gnueabi/cubedma/1.0-r
0/temp/log.do_compile.27844)
ERROR: Logfile of failure stored in: /home/hypso/petalinux/projects/pico-primary/build/tmp/work
/plnx_zynq7-xilinx-linux-gnueabi/cubedma/1.0-r0/temp/log.do_compile.27844
Log data follows:
| DEBUG: Executing shell function do_compile
| NOTE: make -j 4 KERNEL_SRC=/home/hypso/petalinux/projects/pico-primary/build/tmp/work-shared/
plnx-zynq7/kernel-source KERNEL_PATH=/home/hypso/petalinux/projects/pico-primary/build/tmp/work
-shared/plnx-zynq7/kernel-source KERNEL_VERSION=4.19.0-xilinx-v2019.1 CC=arm-xilinx-linux-gnuea
bi-gcc  -mno-thumb-interwork -marm -fuse-ld=bfd -fdebug-prefix-map=/home/hypso/petalinux/projec
ts/pico-primary/build/tmp/work/plnx_zynq7-xilinx-linux-gnueabi/cubedma/1.0-r0=/usr/src/debug/cu
bedma/1.0-r0 -fdebug-prefix-map=/home/hypso/petalinux/projects/pico-primary/build/tmp/work/plnx
_zynq7-xilinx-linux-gnueabi/cubedma/1.0-r0/recipe-sysroot= -fdebug-prefix-map=/home/hypso/petal
inux/projects/pico-primary/build/tmp/work/plnx_zynq7-xilinx-linux-gnueabi/cubedma/1.0-r0/recipe
-sysroot-native=  -fdebug-prefix-map=/home/hypso/petalinux/projects/pico-primary/build/tmp/work
-shared/plnx-zynq7/kernel-source=/usr/src/kernel LD=arm-xilinx-linux-gnueabi-ld.bfd   AR=arm-xi
linx-linux-gnueabi-ar  O=/home/hypso/petalinux/projects/pico-primary/build/tmp/work-shared/plnx
-zynq7/kernel-build-artifacts KBUILD_EXTRA_SYMBOLS=
| make -C /home/hypso/petalinux/projects/pico-primary/build/tmp/work-shared/plnx-zynq7/kernel-s
ource M=/home/hypso/petalinux/projects/pico-primary/build/tmp/work/plnx_zynq7-xilinx-linux-gnue
abi/cubedma/1.0-r0
| make[1]: Entering directory '/home/hypso/petalinux/projects/pico-primary/build/tmp/work-share
d/plnx-zynq7/kernel-source'
| make[2]: Entering directory '/home/hypso/petalinux/projects/pico-primary/build/tmp/work-share
d/plnx-zynq7/kernel-build-artifacts'
|   CC [M]  /home/hypso/petalinux/projects/pico-primary/build/tmp/work/plnx_zynq7-xilinx-linux-
gnueabi/cubedma/1.0-r0/cubedma.o
| /home/hypso/petalinux/projects/pico-primary/build/tmp/work/plnx_zynq7-xilinx-linux-gnueabi/cu
bedma/1.0-r0/cubedma.c:19:1: warning: no semicolon at end of struct or union
|  };
|  ^
| /home/hypso/petalinux/projects/pico-primary/build/tmp/work/plnx_zynq7-xilinx-linux-gnueabi/cu
bedma/1.0-r0/cubedma.c: In function 'dev_mmap':
| /home/hypso/petalinux/projects/pico-primary/build/tmp/work/plnx_zynq7-xilinx-linux-gnueabi/cu
bedma/1.0-r0/cubedma.c:41:2: warning: ISO C90 forbids mixed declarations and code [-Wdeclaratio
n-after-statement]
|   int ret = 0;
|   ^~~
| /home/hypso/petalinux/projects/pico-primary/build/tmp/work/plnx_zynq7-xilinx-linux-gnueabi/cu
bedma/1.0-r0/cubedma.c: At top level:
| /home/hypso/petalinux/projects/pico-primary/build/tmp/work/plnx_zynq7-xilinx-linux-gnueabi/cu
bedma/1.0-r0/cubedma.c:57:20: error: 'dev_ioctl' undeclared here (not in a function); did you m
ean 'blkdev_ioctl'?
|   .unlocked_ioctl = dev_ioctl
|                     ^~~~~~~~~
|                     blkdev_ioctl
| /home/hypso/petalinux/projects/pico-primary/build/tmp/work/plnx_zynq7-xilinx-linux-gnueabi/cu
bedma/1.0-r0/cubedma.c: In function 'init_channel':
| /home/hypso/petalinux/projects/pico-primary/build/tmp/work/plnx_zynq7-xilinx-linux-gnueabi/cu
bedma/1.0-r0/cubedma.c:67:2: warning: ISO C90 forbids mixed declarations and code [-Wdeclaratio
n-after-statement]
|   char class_name[32] = DEVICE_NAME;
|   ^~~~
| make[3]: *** [/home/hypso/petalinux/projects/pico-primary/build/tmp/work-shared/plnx-zynq7/ke
rnel-source/scripts/Makefile.build:312: /home/hypso/petalinux/projects/pico-primary/build/tmp/w
ork/plnx_zynq7-xilinx-linux-gnueabi/cubedma/1.0-r0/cubedma.o] Error 1
| make[2]: *** [/home/hypso/petalinux/projects/pico-primary/build/tmp/work-shared/plnx-zynq7/ke
rnel-source/Makefile:1517: _module_/home/hypso/petalinux/projects/pico-primary/build/tmp/work/p
lnx_zynq7-xilinx-linux-gnueabi/cubedma/1.0-r0] Error 2
| make[2]: Leaving directory '/home/hypso/petalinux/projects/pico-primary/build/tmp/work-shared
/plnx-zynq7/kernel-build-artifacts'
| make[1]: *** [Makefile:146: sub-make] Error 2
| make[1]: Leaving directory '/home/hypso/petalinux/projects/pico-primary/build/tmp/work-shared
/plnx-zynq7/kernel-source'
| make: *** [Makefile:6: all] Error 2
| ERROR: oe_runmake failed
| WARNING: exit code 1 from a shell command.
| ERROR: Function failed: do_compile (log file is located at /home/hypso/petalinux/projects/pic
o-primary/build/tmp/work/plnx_zynq7-xilinx-linux-gnueabi/cubedma/1.0-r0/temp/log.do_compile.278
```

```
44)
ERROR: Task (/home/hypso/petalinux/projects/pico-primary/project-spec/meta-user/recipes-modules
/cubedma/cubedma.bb:do_compile) failed with exit code '1'
```

Full log: full.log

magnudan added bug **HIGH PRIORITY** labels on Apr 27

drowzie commented on Apr 27

i forgot a semicolon...

```
struct device_data{
        int    major_number;            ///< Stores the device number -- determined
automatically
        struct class*  p_device_class;  ///< The device-driver class struct pointer
        struct device* p_device;                              ///< The device-driver
device struct pointer
        struct cdev cdev;
        uint16_t * p_dma_data
};
```

magnudan commented on Apr 28

Master branch works now on 9f67eb6. Have both built and booted successfully

magnudan closed this on Apr 28

Assignees

No one—assign yourself

Labels

HIGH PRIORITY

bug

Projects

None yet

Milestone

No milestone

Linked pull requests

Successfully merging a pull request may close this issue.

None yet

2 participants

Pin issue

AY opu-system Issue 89

Edit | New issue

Jump to bottom

Update memory range for Cube DMA on PicoBOB #89

Closed **magnudan** opened this issue on Apr 30 · 3 comments

Labels Vivado points=1

Projects SW kanban board

magnudan commented on Apr 30

The Cube DMA is set to a range of 512M, while it needs 1G for Cube DMA to work correctly:

```
set_property range 1G [get_bd_addr_segs
{cubedma_top_0/m_axi_mem/SEG_processing_system7_0_HP0_DDR_LOWOCM}]
```

sivertba added the points=1 label on Apr 30

sivertba added this to **Backlog** in **SW kanban board** on Apr 30

sivertba moved this from **Backlog** to **To do** in **SW kanban board** on Apr 30

magnudan mentioned this issue on Apr 30

Updated memory regions and hard-coded addresses #91 Merged

JoarGjersund commented on Apr 30 • edited

Wouldn't this cause problems with memory corruption and kernel panics in linux if we give cube dma access to memory outside the reserved region?

magnudan added the Vivado label on Apr 30

magnudan commented on Apr 30

I don't know if this can cause new problems, but when set to 512M instead of 1G it will fail every time. But when it is 1G it will succeed every time

JoarGjersund commented on Apr 30

I have fate. Merged in #91 .

JoarGjersund closed this on Apr 30

SW kanban board automation moved this from **To do** to **Done** on Apr 30

JoarGjersund mentioned this issue on May 12

Status of CCSDS on FPGA: Borked #116 Closed

Assignees

No one—assign yourself

Labels

Vivado

points=1

Projects

SW kanban board

Done

Milestone

No milestone

Linked pull requests

Successfully merging a pull request may close this issue.

None yet

3 participants

Pin issue

AZ opu-system Issue 90

Edit | New issue

Jump to bottom

Enhance .tcl scripts #90

Open **magnudan** opened this issue on Apr 30 · 2 comments

Labels	enhancement hypso-2 in progress points=5
Projects	SW kanban board

magnudan commented on Apr 30 • edited

The `.tcl` scripts for PicoBOB and ZedBoard in `vivado/scripts` contain a lot of the same steps. It would be nice to have common changes apply to both builds. I see two ways this can be done:

- Have one script where user input defines what to build
- Have one script for ZedBoard and one for PicoBOB and common settings in another script. The "main" scripts call on the common.

Other things that would be nice to have is the possibility to change path with user input, so that we don't have to change the scripts.

magnudan added enhancement hypso-2 labels on Apr 30

sivertba commented on Apr 30 • edited

I would cast my vote for the one script to rule them all.

Edit:
Let the user input be prompted in the terminal rather than parameters sent into the script call.

❤️ 1

sivertba added the points=5 label on May 14

sivertba added this to **Backlog** in **SW kanban board** on May 14

JoarGjersund commented on May 16

Adding to this. We should also define static and reconfigurable partitions, I am currently looking into how to do this. I am thinking of making only one reconfigurable partition called image_processing which has the same interfaces as the ccsds123, with ccsds123 beeing a reconfigurable module that can be loaded onto this partition. This will save a lot of time for people who develop and prototype RTL logic as they do not need to re-synthesis and implement the static region everytime they want to test something on the board. Partial bitstreams are also usually a lot smaller in size and will therefore be faster to upload, and to load onto the board during image processing.

JoarGjersund added the **in progress** label on May 16

Assignees

No one—assign yourself

Labels

enhancement
hypso-2
in progress
points=5

Projects

SW kanban board

Backlog

Milestone

No milestone

Linked pull requests

Successfully merging a pull request may close this issue.

None yet

3 participants

Pin issue

BA opu-system Pull Request 91

NTNU-SmallSat-Lab / **opu-system** Private

Updated memory regions and hard-coded addresses #91

Edit | Open with

Merged **JoarGjersund** merged 2 commits into `master` from `update_tcl_scripts` on Apr 30

Conversation 2 | Commits 2 | Checks 0 | Files changed 10 | +1,277 −20

magnudan commented on Apr 30

The memory region for Cube DMA on PicoZed has been set to 1G according to issue #89.

I have also hard-coded the configuration registers for Cube DMA to "0x43C00000" for both PicoZed and ZedBoard. The mapping was done automatically by Vivado before, and I don't really trust Vivado...

I have also added two new .tcl scripts. They make hardware files with Cube DMA looped back to itself. Might be useful to have, but are not needed to test interfacing.

`Updated memory regions and hard-coded addresses` `fe83418`

magnudan requested review from **jonakor**, **JoarGjersund** and **drowzie** on Apr 30

magnudan commented on Apr 30 Author

@JoarGjersund @drowzie @jonakor
It would be great if you could just read through the code. You don't need to build and test, as we don't really have the time...

`Update device list` `c0f7372`

magnudan mentioned this pull request on Apr 30

Cube DMA now works NTNU-SmallSat-Lab/hypso-sw#191 Merged

JoarGjersund merged commit `6012d41` into `master` on Apr 30 Revert

jonakor reviewed on Apr 30 View changes

jonakor left a comment

Looking good to me

JoarGjersund mentioned this pull request on Apr 30

Update memory range for Cube DMA on PicoBOB #89 Closed

magnudan deleted the `update_tcl_scripts` branch on May 1 Restore branch

Reviewers
jonakor
drowzie
JoarGjersund

Assignees
No one—assign yourself

Labels
None yet

Projects
None yet

Milestone
No milestone

Linked issues
Successfully merging this pull request may close these issues.
None yet

3 participants

BB opu-system Issue 95

Edit | New issue

Jump to bottom

CCSDS 123 version 1 Compresses Fixed Cube Sizes #95

Open **magnudan** opened this issue on May 4 · 6 comments

Labels	Vivado points=21
Projects	SW kanban board

magnudan commented on May 4

Description

As CCSDS 123 version 1 is implemented right now, it must be configured in Vivado with the cube size it is going to compress. This means that once the bitstream is generated, it is going to compress input data as a `CONFIG.NX * CONFIG.NY * CONFIG.NZ` cube no matter what it is (see code block below). The size can be changed by changing the parameters in the following code block in the `.tcl` script corresponding to the desired hardware. Then the script must be run in Vivado to generate a bitstream for compression of this new cube size.

```
#*****************************************************************************************
# Including and configuring the ccsds123 module
#*****************************************************************************************
create_bd_cell -type module -reference ccsds123_top ccsds123_top_0

set_property -dict [ list \
  CONFIG.BUS_WIDTH     {64} \
  CONFIG.COL_ORIENTED  {false} \
  CONFIG.COUNTER_SIZE  {6} \
  CONFIG.D             {16} \
  CONFIG.INITIAL_COUNT {1} \
  CONFIG.ISUNSIGNED    {true} \
  CONFIG.KZ_PRIME      {5} \
  CONFIG.LITTLE_ENDIAN {true} \
  CONFIG.NX            {720} \
  CONFIG.NY            {500} \
  CONFIG.NZ            {107} \
  CONFIG.OMEGA         {13} \
  CONFIG.ON_THE_FLY    {false} \
  CONFIG.P             {3} \
  CONFIG.PIPELINES     {4} \
  CONFIG.R             {32} \
  CONFIG.REDUCED       {false} \
  CONFIG.TINC_LOG      {6} \
  CONFIG.UMAX          {16} \
  CONFIG.V_MAX         {3} \
  CONFIG.V_MIN         {-1} \
] [get_bd_cells ccsds123_top_0]
```

@**drowzie** is working on refactoring the source code for CCSDS 123 version 1 to be re-configurable in-flight, but this is what we have to work with right now.

magnudan added the Vivado label on May 4

rogerbirkeland commented on May 4

What is in the rest of the file, if the result should be (much) less than the configured size?

drowzie commented on May 4 • edited

> What is in the rest of the file, if the result should be (much) less than the configured size?

Only Garbage or data from previous images.

magnudan commented on May 4

Or whatever data is in the physical memory locations that Cube DMA reads from

drowzie commented on May 4 • edited

As there is short amount of time until the thesis is due. It is difficult for me to fix this problem within the time frame. I am working on implementing variable cube sizes into my own design in CCSDS123V2.

But if this is to be implemented into CCSDS123V1 then it would take ~1-2 months to refactor Johans design.

rogerbirkeland commented on May 4

> Or whatever data is in the physical memory locations that Cube DMA reads from

Is this stored sequentially, or interleaved? (I have no idea of how the data format is made). Meaning; can we chop the "fixed" file and then get a smaller file of only the desired data?

magnudan mentioned this issue on May 4

Cube DMA now works NTNU-SmallSat-Lab/hypso-sw#191 Merged

magnudan commented on May 4

It is stored sequentially. So if we have a way of knowing when the useful data ends, we can discard the rest

1

sivertba added the points=21 label on May 14

sivertba added this to **Backlog** in **SW kanban board** on May 14

Assignees

No one—assign yourself

Labels

Vivado

points=21

Projects

SW kanban board

Backlog

Milestone

No milestone

Linked pull requests

Successfully merging a pull request may close this issue.

None yet

4 participants

Pin issue

Edit | New issue

Jump to bottom

Use script in hypso-sw when building opu-system #99

Closed **magnudan** opened this issue on May 6 · 2 comments

Labels enhancement

magnudan commented on May 6 • edited

Describe the solution you'd like
The script `update-hypso-sw` in `docker/` could execute the script `start_docker` in `etalinux/projects/software/hypso-sw/scripts/` instead of the `docker run <...>` command.

Additional context
The reason I'm requesting this is to remove redundant code. The `docker run <...>` command in the script in hypso-sw is updated, while the command in the script in opu-services is not

magnudan changed the title ~~Update hypso~~ **Use script in hypso-sw when building opu-system** on May 6

magnudan added the enhancement label on May 6

JoarGjersund commented on May 6

not really. The script in update-hypso-sw does more than just starting docker, it also runs the commands `rm build; make; exit;`
One alternative is to add an input parameter in the start_docker script for running commands inside the docker container.

magnudan commented on May 7

I see. That be more hassle than it's worth. I guess we're not going to update the `docker run` commands very often

I'll create a new issue about updating the script, then. As this solution might not be the best.

 **magnudan** closed this on May 7

Assignees

No one—assign yourself

Labels

enhancement

Projects

None yet

Milestone

No milestone

Linked pull requests

Successfully merging a pull request may close this issue.

None yet

2 participants

Pin issue

Edit | New issue

Jump to bottom

Old version of hypso-sw #102

Closed **magnudan** opened this issue on May 6 · 6 comments

Labels: bug, concern, points=5

Projects: SW kanban board

magnudan commented on May 6

Describe the bug
The hypso-sw submodule of opu-system defaults to commit 783e458.

To Reproduce
A fresh clone of opu-services by either

```
$ git clone git@github.com:NTNU-SmallSat-Lab/opu-system.git --recursive
```

or

```
$ git clone git@github.com:NTNU-SmallSat-Lab/opu-system.git
$ cd opu-system
opu-system$ git submodule update --init
```

will result in hypso-sw being at commit 783e458:

```
opu-system/petalinux/projects/software/hypso-sw$ git status
HEAD detached at 783e458
nothing to commit, working tree clean
```

Manually checking out the master branch in the submodule works

```
opu-system/petalinux/projects/software/hypso-sw$ git checkout master
Previous HEAD position was 783e458 Merge pull request #172 from NTNU-SmallSat-Lab/opu-
upload
Switched to branch 'master'
Your branch is up to date with 'origin/master'.
```

Running the update script keeps the submodule at HEAD of master, but git is not satisfied:

```
opu-system$ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
```

```
    (use "git checkout -- <file>..." to discard changes in working directory)

          modified:   petalinux/projects/software/hypso-sw (new commits)

  no changes added to commit (use "git add" and/or "git commit -a")
```

Updating manually reverts the submodule back to commit 783e458:

```
opu-system$ git submodule update
Submodule path 'petalinux/projects/software/hypso-sw': checked out
'783e4586064b1c02fc527283e2ef627042ff0bd5'
```

Expected behavior
I don't understand why the submodule is not always at HEAD of master on hypso-sw

 **magnudan** added `bug` `concern` labels on May 6

 **JoarGjersund** commented on May 6 • edited

when running build_all it automatically pulls the latest commit on hypso-sw master branch. It also does a rebase:

opu-system/docker/update-hypso-sw
Line 8 in `f9d3acf`

```
8     git submodule update --rebase --remote
```

I dont remember why I added the rebase flag, but perhaps that answers your question?

 **JoarGjersund** commented on May 6

would probably make more sense to just use git pull --recurse, and not let the build_all script do any git magic in the backround. Would be more transparent to the user I guess..

 **magnudan** commented on May 7 • edited

From a fresh clone:

```
$ git clone git@github.com:NTNU-SmallSat-Lab/opu-system.git --recursive
.
.
.
```

Checking commit of hypso-sw:

```
opu-system/petalinux/projects/software/hypso-sw$ git status
HEAD detached at 783e458
nothing to commit, working tree clean
```

Trying the `./build_all` script:

```
opu-system$ ./build_all 16 -zed
Building with IP-Adress: 129.241.2.36/23
cp: cannot stat 'vivado/projects/PicoBOB/PicoBOB.sdk/*': No such file or directory
cp: cannot stat 'vivado/projects/ZedBoard/ZedBoard.sdk/*': No such file or directory
remote: Enumerating objects: 2, done.
remote: Counting objects: 100% (2/2), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 2 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (2/2), done.
From github.com:NTNU-SmallSat-Lab/hypso-sw
   26964b6..3d823db  master     -> origin/master
First, rewinding head to replay your work on top of it...
warning: unable to rmdir 'extern/linenoise': Directory not empty
Fast-forwarded HEAD to 3d823db1f6ecb8bc8c835c074e3a36981dfa0ac8.
Submodule path '../petalinux/projects/software/hypso-sw': rebased into
'3d823db1f6ecb8bc8c835c074e3a36981dfa0ac8'
.
.
.
```

While it says that the submodule is updated, it's still in a detached state:

```
opu-system/petalinux/projects/software/hypso-sw$ git status
HEAD detached from 783e458
Untracked files:
  (use "git add <file>..." to include in what will be committed)

        extern/linenoise/

nothing added to commit but untracked files present (use "git add" to track)
```

 JoarGjersund commented on May 7

```
warning: unable to rmdir
```

Do you have any of the files or folders in hypso-sw open while running the build_all script?

 **sivertba** commented on May 14

Consider using git fetch and checkout master or recursively.
Needs to be looked at ...

  **sivertba** added the `points=5` label on May 14

  **sivertba** added this to **Backlog** in **SW kanban board** on May 14

  **sivertba** moved this from **Backlog** to **To do** in **SW kanban board** on May 14

 **DennisNTNU** commented on May 14 • edited

> While it says that the submodule is updated, it's still in a detached state:
>
> ```
> opu-system/petalinux/projects/software/hypso-sw$ git status
> HEAD detached from 783e458
> Untracked files:
>   (use "git add <file>..." to include in what will be committed)
>
>         extern/linenoise/
>
> nothing added to commit but untracked files present (use "git add" to track)
> ```

I checked the actual source file contents and the repository is really updated, despite the detached head and commit hash from 5 weeks ago. I don't fully understand what rebasing does exactly but it seems a property of it is to apply changes without changing the current commit hash of the repo.

I would still like to suggest replacing the

```
git submodule update --rebase --remote
```

in the `docker/update-hypso-sw` script with

```
git submodule foreach "git checkout master; git pull; git submodule update --init"
```

or something similar, which enters each submodule, checks out master, pulls and checks out the submodules of the submodules in the recorded commit. Doing `git pull --recurse-submodules` doesn't work because submodules aren't on any branches and they need to be on a branch to pull. Doing the `checkout master` and `pull` recursively for `hypso-sw` 's submodules is I think not desirable, because those are added at a specific commit that we know works and I fear always updating them might cause bugs. For example the `extern/libtable` submodule in hypso-sw is added as submodule on the branch `hypso` and not `master` .

> > `warning: unable to rmdir`

> Do you have any of the files or folders in hypso-sw open while running the build_all script?

This error comes from git and is nothing to worry about. With the latest, rebased changes this submodule is removed and git tries to remove the folder, but doesn't if it isn't empty.

DennisNTNU mentioned this issue on May 14

Checkout master and pulling in submodules #121 Merged

evelynlimore moved this from **To do** to **Review in progress** in **SW kanban board** on May 15

DennisNTNU closed this in #121 27 days ago

SW kanban board automation moved this from **Review in progress** to **Done** 27 days ago

Assignees

No one—assign yourself

Labels

bug

concern

points=5

Projects

SW kanban board

Done

Milestone

No milestone

Linked pull requests

Successfully merging a pull request may close this issue.

Checkout master and pulling in submodules

4 participants

Pin issue

BC opu-system Issue 103

Edit | New issue

Jump to bottom

Use user and group IDs when running hypso-sw docker #103

Closed **magnudan** opened this issue on May 7 · 0 comments

Labels **enhancement** points=1

Projects **SW kanban board**

magnudan commented on May 7

Is your feature request related to a problem? Please describe.
To minimize potential problems with file permissions, user and group IDs can be given as parameters to the `docker run command` . This will result in the user in the docker container having the same IDs as the user that started the container, giving all these users the same permissions to the same files. This also removes the need for the script to be run as super user (`sudo`).

Describe the solution you'd like
This has been implemented in the script for starting docker in the hypso-sw repository:

```
if [ $# -eq 2 ]
  then
    USER_ID=$1
    GROUP_ID=$2
  else
    USER_ID=$(id -u)
    GROUP_ID=$(id -g)
fi

echo "using uid $USER_ID and gid $GROUP_ID"
docker run -it -v $SCRIPTPATH/..:/home/hypso/ --user $USER_ID:$GROUP_ID hypso-sw bash
```

magnudan mentioned this issue on May 7

Update update-hypso-sw script #104 **Merged**

magnudan added the **enhancement** label on May 7

sivertba added the points=1 label on May 14

 **sivertba** closed this on May 14

  **sivertba** added this to **Done** in **SW kanban board** on May 14

Assignees

No one—assign yourself

Labels

enhancement

points=1

Projects

SW kanban board

Done

Milestone

No milestone

Linked pull requests

Successfully merging a pull request may close this issue.

None yet

2 participants

Pin issue

BD opu-system Pull Request 104

NTNU-SmallSat-Lab / **opu-system** Private

Update update-hypso-sw script #104

Edit | Open with

Merged **magnudan** merged 1 commit into `master` from `update_update-hypso-sw_script` on May 11

Conversation 5 | Commits 1 | Checks 0 | Files changed 1 | +9 −2

magnudan commented on May 7

Implements feature in #103

`added user and group IDs to docker run command, and removed sudo` `b99ee8c`

magnudan requested review from **JoarGjersund**, **DennisNTNU** and **magne-hov** on May 7

JoarGjersund commented on May 7

Not sure what you are trying to acomplish with this, but you should probably update the build_all script as well, to include the arguments. I am also not sure what will happen if not [$# -eq 2].

The way it is now I am afraid it will break the build_all script. Is this tested? Can you provide a quick testreport?

DennisNTNU approved these changes on May 7 View changes

DennisNTNU left a comment

I managed to build an image so the build_all script does still work. I can confirm that building the hypso-sw works now without sudo, but I am asked for a password at a later point in the script, right after the part where all the ueye driver files are being listed. I'd assume this other sudo is for starting the petalinux docker image? Do you know how to implement the same fix for that?

magnudan commented on May 7 Author

This is just to run the docker image for hypso-sw in the same way for the *submodule* in this repo as in the hypso-sw *main* repo.

I don't know how to do this for the petalinux image, as there is a lot going on there. And yes, I believe this is where you're prompted for a password

DennisNTNU commented on May 8 • edited

> [...] but you should probably update the build_all script as well, to include the arguments. I am also not sure what will happen if not [$# -eq 2].

Its safe to run this script without arguments. If the script has exactly to arguments, it assumes a custom user and group id is given. If there are no arguments, the IDs are found from the user running the script.

magnudan commented on May 8 • edited Author

Is it OK to merge, then?

Reviewers
- DennisNTNU ✓
- JoarGjersund ●
- magne-hov ●

Assignees
No one—assign yourself

Labels
None yet

Projects
None yet

Milestone
No milestone

Linked issues
Successfully merging this pull request may close these issues.
None yet

3 participants

DennisNTNU mentioned this pull request on May 8

No sudo #107 Merged

magnudan merged commit `76dc62f` into `master` on May 11 Revert

magnudan deleted the `update_update-hypso-sw_script` branch on May 11 Restore branch

BE opu-system Pull Request 107

NTNU-SmallSat-Lab / **opu-system** Private

No sudo #107

Edit | Open with

Merged **magnudan** merged 2 commits into `master` from `no-sudo` on May 11

Conversation 5 | Commits 2 | Checks 0 | Files changed 2 | +11 −6

DennisNTNU commented on May 8 • edited

These changes together with #104 removes the need to enter sudo password form the `build_all` script. Another prerequisite may be that the user running `build_all` needs to be part of the docker group: `sudo adduser <user> docker` .
This only works properly if the changes from #104 are also present. If *not*, you will still be required to enter pw for building the newest opu-services and you *may* get tons of

```
rm: descend into write-protected directory '<path>'?
```

queries, that fail when answered with `y` or `yes` . Despite this, building the image afterwards worked for me when I tried it.

Details:

The docker image mounts the repository's `petalinux` folder into the container folder `/home/hypso/petalinux` , which may has different ID's from the host user. The image building process writes to this folder and also in the hypso's home foder `home/hypso` , which have the default user's user and groupd ID `1000:1000` . Due to this, the same fix that worked for `hypso-sw` , doesn't work here. This solution is to change the IDs of the petalinux folder to the container user's IDs by `sudo chown -R 1000:100 petalinux` , perfrom the building, and afterward change the IDs to the host user's IDs, so that the built files are still properly accessible from without the container.

Additional note

For a reason unknown to me, the mounted folder `/home/hypso/petalinux` doesn't actually have the host user's IDs, but the ID's of the root user (`0:0`). This is also a reason for why the same solution that worked for `hypso-sw` doesn't work here.

DennisNTNU added 2 commits on May 8

`docker user and group ID changes in bootfiles script` 4d613da

`remove sudo from build_all` a1ac169

DennisNTNU requested review from **JoarGjersund** and **magnudan** on May 8

magnudan commented on May 8

First, I did a fresh clone of the repository, then copied over the `update-hypso-sw` script from #104.
Now is when I *usually* would have to give 777 permissions to the entire petalinux directory to not encounter any permission errors. But in good faith, I started the `build_all` script to build files for the ZedBoard I have at home.

I did not get prompted for my password once, I did not see anything unusual during the build, the build completed successfully, and the boot files worked as expected on my ZedBoard.

I am happy 😄

😄 1

magnudan approved these changes on May 8 View changes

Reviewers
magnudan ✓
JoarGjersund ●

Assignees
No one—assign yourself

Labels
None yet

Projects
None yet

Milestone
No milestone

Linked issues
Successfully merging this pull request may close these issues.
None yet

3 participants

magnudan merged commit `727cf29` into `master` on May 11 — Revert

magnudan deleted the `no-sudo` branch on May 11 — Restore branch

JoarGjersund commented on May 12

I am now getting the following error when running build_all.

```
docker: Got permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Post http://%2Fvar%2Frun%2Fdocker.sock/v1.40/containers/create: dial unix /var/run/docker.sock: connect: permission denied.
```

anyone else experiencing this?

DennisNTNU commented on May 12 • edited — Author

Can you try `sudo adduser <your username> docker` and relogging?

JoarGjersund commented on May 12

running these commands solved it. Will add it to the readme file.

```
sudo groupadd docker
sudo gpasswd -a $USER docker
```

DennisNTNU commented on May 12 • edited — Author

Did you need to log out and in again after running `sudo gpasswd -a $USER docker` ?

Can you omit adding `sudo groupadd docker` , because anyone having installed docker should have the docker group already making adding it unnecessary.

`sudo gpasswd -a $USER docker` does probably the same as `sudo adduser <your username> docker` or `sudo adduser $USER docker` .

BF opu-system Issue 108

Edit | New issue

Jump to bottom

Fix issue with petalinux config crash if terminal window is too small #108

Closed **magnudan** opened this issue on May 8 · 5 comments

Labels: bug help wanted **petalinux** points=3 **question**

Projects: **SW kanban board**

magnudan commented on May 8

Describe the bug
The bug is described in detail in issue #76. In short: The petalinux config commands will fail if the terminal is too small.

To Reproduce
Make the terminal small and run the `build_all` script in the root folder of the repository.

Quick fix
The bash command `printf '\e[8;50;100t'` (source) will resize the terminal to 50 lines and 100 columns. It will however not change the size of the window. This means that the window might not see all that is happening in the terminal unless resized manually.

If this command is added to the script `load_firmware` in `petalinux/projects` before the petalinux config commands they *should* not fail...

Disclaimers

- I've only tested this once
- It is not elegant, and there is probably a better solution

If you have any other suggestions, please discuss

magnudan added bug help wanted **petalinux** **question** labels on May 8

JoarGjersund commented on May 8

Sound like a good solution!

sivertba added the points=3 label on May 14

sivertba added this to **Backlog** in **SW kanban board** on May 14

sivertba commented on May 14

Need to use ubuntu standard terminal?

sivertba moved this from **Backlog** to **To do** in **SW kanban board** on May 14

magnudan mentioned this issue on May 14

Added command to scale terminal inside window & added --silentconfig Merged

#120

evelynlimore moved this from **To do** to **Review in progress** in **SW kanban board** on May 15

DennisNTNU commented 18 days ago • edited

Some good news and bad news. I found the fix to why the boot image building stuff in jenkins couldn't find the petalinux-create command. The problem was basically that I wasn't doing exactly the same that the build-bootfiles script does after all.

But now I am getting the same looking error right after the menuconfig step as with running the the script from a too small terminal window.

There must be a way to build the linux OS without the menuconfig step.

DennisNTNU commented 18 days ago

I couldn't keep myself from searching around and I found something promising. Upon googleing "petalinux config without menuconfig" one of the links was this

Which said that the menuconfig window can be suppressed by adding `--oldconfig` to the `petalinux-config` command. Doing this actually gets past where jenkins previously error'd and its doing the bitbake now. It also prints out that `--oldconfig` is deprecated and will be removed in the future, and an alternative is `--silentconfig`. So Its probably this flag that we can add to the `petalinux-create` calls in the `petalinux/projects/load_firmware` script.

 **magnudan** commented 18 days ago

This is promising! I think it's better to solve this with petalinux rather than hacking some solution by scaling the terminal

  **evelynlimore** moved this from **Review in progress** to **Done** in **SW kanban board** 11 days ago

  **evelynlimore** moved this from **Done** to **Review in progress** in **SW kanban board** 11 days ago

 **DennisNTNU** closed this in #120 8 days ago

 **SW kanban board** automation moved this from **Review in progress** to **Done** 8 days ago

Assignees 

No one—assign yourself

Labels 

bug
help wanted
petalinux
points=3
question

Projects 

 **SW kanban board**

Done

Milestone

No milestone

Linked pull requests

Successfully merging a pull request may close this issue.

Added command to scale terminal inside window & added --silentconfig

4 participants

Pin issue

BG opu-system Pull Request 120

NTNU-SmallSat-Lab / **opu-system** Private

Added command to scale terminal inside window & added --silentconfig #120

Edit | Open with

Merged **DennisNTNU** merged 5 commits into `master` from `test-terminal-scaling` 8 days ago

Conversation 4 | Commits 5 | Checks 0 | Files changed 7 | +24 −24

magnudan commented on May 14 • edited

If approved, this PR will close #108

I have tested this on a small window and a big window without kconfig errors in petalinux. I use the Konslole terminal in KDE on my "local" docker repository, and ssh into the repository I have on WS3 in the lab.

Please test on different setups such as tiling desktop managers and Jenkins!

NOTE
As the window itself does not scale, prints in terminal might not be visible during builds. In my experience, the terminal resets to window size once the build completes.

`Added command to scale terminal inside window` 68bd306

magnudan requested review from **drowzie**, **JoarGjersund** and **DennisNTNU** on May 14

DennisNTNU commented on May 14

This really doesn't work with my window manager. I would have expected that the `printf "\e[8;50;100t"` makes text go off the screen due to some virtual size being larger than the window size, but text is still wrapped properly. I wish I could test it on jenkins, but there is another issue that prevents the script from reaching the menuconfig part.

evelynlimore added this to **Review in progress** in **SW kanban board** 25 days ago

sivertba commented 19 days ago

The window manager Dennis is using is i3, tiled window manager. Do not know how it works with Jenkins either

`Add --silentconfig to petalinux-config command` 3d804f6

DennisNTNU commented 17 days ago

I am hijacking this branch and PR for adding `--silentconfig` . I tested it and building works on my end, even with a tiny ca. 50x5 character terminal window.

1

DennisNTNU changed the title ~~**Added command to scale terminal inside window**~~ **Added command to scale terminal inside window & added --silentconfig** 13 days ago

DennisNTNU added 3 commits 12 days ago

Reviewers
DennisNTNU ✓
drowzie ●
JoarGjersund ●

Assignees
No one—assign yourself

Labels
None yet

Projects
SW kanban board
Done

Milestone
No milestone

Linked issues
Successfully merging this pull request may close these issues.
Fix issue with petalinux config cra…

3 participants

`Remove unnecessary whitespace` `32709bd`

`Add 'apt update' to fix 404 error during apt install ssh rsync` `a5d2dec`

`Remove terminal resizing, because --silentconfig made it unnecessary` `f469782`

DennisNTNU approved these changes 11 days ago — View changes

DennisNTNU left a comment

I am just going to approve this PR, even though I kind of took it over and by now made most of the changes myself. I tested building on both my laptop and the WS2 and am confident the changes work as intended.

SW kanban board automation moved this from **Review in progress** to **Reviewer approved** 11 days ago

DennisNTNU merged commit `7c1de9d` into `master` 8 days ago — Revert

SW kanban board automation moved this from **Reviewer approved** to **Done** 8 days ago

DennisNTNU deleted the `test-terminal-scaling` branch 8 days ago — Restore branch

NTNU-SmallSat-Lab / **opu-system** Private

Checkout master and pulling in submodules #121

Edit | Open with

Merged **DennisNTNU** merged 1 commit into `master` from `submodule-update` 27 days ago

Conversation 0 | Commits 1 | Checks 0 | Files changed 1 | +1 −1

DennisNTNU commented on May 14

resolves #102

Updates the submodules (i.e. just `hypso-sw`) by checking out master and pulling, instead of rebasing. I'd like to do the updating this way, because when the upcomming git versioning changes (NTNU-SmallSat-Lab/hypso-sw#238) reads the current commit hash and branch of the HEAD of the submodule, I suspect the wrong hash and branch name will be compiled when rebasing as documented in issue #102.

`checkout master and pulling in submodules` `04aeee2`

DennisNTNU requested review from **JoarGjersund** and **magnudan** on May 14

sivertba approved these changes 29 days ago — View changes

DennisNTNU merged commit `79237e4` into `master` 27 days ago — Revert

DennisNTNU deleted the `submodule-update` branch 27 days ago

Reviewers
- sivertba ✓
- JoarGjersund ●
- magnudan ●

Assignees
No one—assign yourself

Labels
None yet

Projects
None yet

Milestone
No milestone

Linked issues
Successfully merging this pull request may close these issues.
Old version of hypso-sw

2 participants

BH opu-system Pull Request 122

NTNU-SmallSat-Lab / **opu-system** Private

Integrate timestamp module #122

Edit | Open with

Merged **magnudan** merged 51 commits into `master` from `timestamp_module` on May 15

Conversation 4 | Commits 51 | Checks 0 | Files changed 8 | +381 −3

magnudan commented on May 15

Integrates timestamp module by **@simennett** and **@jonakor**.

The `build_all` script will now also build a kernel module for registering interrupts on MIO9 (PPS) and MIO47 (Flash) for the timestamping of HSI frames. The module will be included in the file system and `insmod` 'ed on boot.

The MIO pins mentioned above might only work on PicoBOB. Pins for ZedBoard are commented out in `petalinux/projects/modules/timestamp/timestamp_main.c`

simennett and others added 30 commits on Feb 27

Added timer block w/ connections and memory address	912a077
Changed the AXT timer part to full auto connection	5575614
Merge branch 'timer_implementation' of https://github.com/NTNU-SmallS… ...	a592dbc
Changing manual connection to automatic	a256f26
Trying to make a functional TCL script with timer	6d914d3
Fixed .tcl script with timer block.	32c3dfe
Added AXI Timer module. Mapped its' registers and connected it with t… ...	e97c210
added spesific address mapping for axi reg	f0ea22f
Added spesific address range and offset	5a8bec0
Merge branch 'master' into timer_implementation	f23ed71
Freshly built hw files	7ae52b7
Added spesific address for timer & changed project path	d495634
Merge with origin	07e3e36
Adding new hardware files for ZedBoard	fb26312
merge with older master commit f41e70f	f618d5c
Testing. Possible revert	677c237
Trying to fix .ko issue. not working yet	d4c22b3
Timestamp module added to build	d6ae425
Added print levels to kernel print	0488621
Update branch with master	35b3ebd
Update from master	b801a0e
Fresh hw-files	e60f916
Update from master	69a07fc
Update hw files	88d6d5e
Adding timestamp module. + some nicening	6021055

Reviewers
- jonakor
- DennisNTNU ✓
- rogerbirkeland ●
- simennett ●

Assignees
No one—assign yourself

Labels
None yet

Projects
None yet

Milestone
No milestone

Linked issues
Successfully merging this pull request may close these issues.
None yet

4 participants

Commented out som print 5d92f25

Changed the MIO pin numbers b14e358

Added echo ··· Verified ee094d4

Removed some printing 5dcafa6

Merge branch 'timestamp_module' into timestamp_test 861bbf3

4 hidden items
Load more…

jonakor and others added 17 commits on Apr 15

Added pin numbers for PicoZed, commented out d5a297c

Commented out some debug print 65eb86f

readded opu-services 078753c

update README with latest build commit f0fda28

Added ioctl for interrupt enable/disable 4db20d8

Moved init of timestamp to ioctl 3afb76f

Update README with build-commit d4a62d7

Changed timerPtr to volatile. Removed while loop in timer source 2a57063

Added timer-reset in ioctl 50c0c2d

clang-format c2e8396

Masking done correct 0f7a1e9

Reorg. declaration 4e7e3ef

Update from master 7fcafaf

Merge branch 'master' into timestamp_module b16e42e

IF no longer needed after enabling/disabling interrupt was implemented d9a6df2

Changed config to PicoBOB fea9ca9

Merge from master 758f528

magnudan requested review from **rogerbirkeland**, **DennisNTNU**, **jonakor** and **simennett** on May 15

magnudan commented on May 15 Author

Added **@simennett** and **@jonakor** because I've merged from master, and want to make sure I've not broken anything by doing so

This was referenced on May 15

Timestamping of HSI frames is live! NTNU-SmallSat-Lab/hypso-sw#240 Merged

Move kernel module files for Cube DMA to its own directory #125 Open

DennisNTNU approved these changes on May 15 View changes

DennisNTNU left a comment • edited

I don't know much about kernel module programming, but I build the boot image from this branch and captured some images and timestamping is working as expected.

🚀 1

magnudan commented on May 15 Author

I have also been able to timestamp a few times, so I'll merge this

 **magnudan** merged commit `b49794c` into `master` on May 15 **Revert**

  **magnudan** deleted the `timestamp_module` branch on May 15 **Restore branch**

 **jonakor** reviewed on May 15 **View changes**

jonakor left a comment

Don't think you've broken it :)

BI opu-system Issue 123

Edit | New issue

Jump to bottom

Build modules based on board #123

Open **magnudan** opened this issue on May 15 · 1 comment

Labels	PicoZed enhancement help wanted points=3 question zedboard
Projects	SW kanban board

magnudan commented on May 15

Is your feature request related to a problem? Please describe.
Right now, there must be done manual work to configure Cube DMA and timestamp modules for either PicoZed or ZedBoard. Either in header files or source files by commenting in/out lines or including different files.

Describe the solution you'd like
I would like to have the correct files to be built automatically based on flags given to the build_all script at top level.

Describe alternatives you've considered
I think this can be done by setting `add_custom_target` to `PicoZed` or `ZedBoard` in the make file and pass it to the compiler. See issue NTNU-SmallSat-Lab/hypso-sw#216 and its PR NTNU-SmallSat-Lab/hypso-sw#238. Then in the source files, conditionals can be made like this:

```
#ifdef __PicoZed__
    // PicoZed, Zynq7030 or BOB specific code
#else // __ZedBoard__
   // Zedboard or Zynq7020 specific code
#endif // End of conditional
```

magnudan added enhancement help wanted question zedboard PicoZed labels on May 15

magnudan mentioned this issue on May 15

Sort out Cube DMA address dependencies #124 Open

sivertba commented 19 days ago • edited

The memory that CubeDMA is reading from is different for picozed and zedboard.
This is now solved by different header files for the two targets.
Want a compile flag to specify which board you're compiling to.

Could fix in build_all potentially easy

 **sivertba** added the points=3 label 19 days ago

 **sivertba** added this to **Backlog** in **SW kanban board** 19 days ago

Assignees

No one—assign yourself

Labels

PicoZed
enhancement
help wanted
points=3
question
zedboard

Projects

SW kanban board

Backlog

Milestone

No milestone

Linked pull requests

Successfully merging a pull request may close this issue.

None yet

2 participants

Pin issue

BJ opu-system Issue 124

Edit | New issue

Jump to bottom

Sort out Cube DMA address dependencies #124

Open **magnudan** opened this issue on May 15 · 3 comments

Labels	PicoZed enhancement help wanted points=3 question zedboard
Projects	SW kanban board

magnudan commented on May 15

Is your feature request related to a problem? Please describe.
The addresses the Cube DMA reads data to and from are dependent on is dependent on the chip/board in use. This must be configured both in the header file in the submodule `petalinux/projects/software/hypso-sw/includes/hsi/cubeDMA.h` and by choosing either `petalinux/projects/modules/dma_parameters.h` or `petalinux/projects/modules/dma_parameters_zed.h`.

Describe the solution you'd like
I think this should be configured by executing the top-level `build_all` script and flags.

Describe alternatives you've considered
This could be solved in the same way as #123.

Additional context
It might also be possible to define the address ranges only for the kernel module, and have `opu-services` read them from there. If this could be done, then the code in the submodule/repo hypso-sw can be made independent on the setting of these addresses. Maybe **@jonakor**, **@simennett** or **@drowzie** can confirm this?

magnudan added enhancement help wanted PicoZed question zedboard labels on May 15

JoarGjersund commented on May 15

Should also note that the address that should be reserved to the dma must also be configured in:

opu-system/petalinux/projects/load_firmware
Lines 85 to 86 in b3dd9a9

```
85      # DDR MEMORY SIZE (MINUS ALLOCATED MEMORY)
86      sed -i 's/CONFIG_SUBSYSTEM_MEMORY_PS7_DDR_0_BANKLESS_SIZE=0x40000000/CONFIG_SUBS
```

and

opu-system/petalinux/projects/system-user.dtsi
Lines 23 to 24 in b3dd9a9

```
23      reserved: buffer@0x30000000 {
24              reg = <0x30000000 0x10000000>;
```

and

opu-system/petalinux/projects/system-user_zedboard.dtsi
Lines 12 to 13 in b3dd9a9

```
12      reserved: buffer@0x10000000 {
13              reg = <0x10000000 0x10000000>;
```

👍 1

jonakor commented on May 15 • edited

@**JoarGjersund** should anything be added to the device three concerning the timestamp kernel module?
@**magnudan** For now, the timer's base address is hardcoded in the hyptimer.h

sivertba added the points=3 label 19 days ago

sivertba added this to **Backlog** in **SW kanban board** 19 days ago

JoarGjersund commented 7 days ago • edited

> @**JoarGjersund** should anything be added to the device three concerning the timestamp kernel module?

I have not looked to deep into how the timestamp kernel is working, but I dont think you need to add anything, unless you are writing to memory regions mapped by the Linux Kernel. If I remember correctly you can see what memory regions are managed by the Linux kernel by typing `cat /proc/iomem`

magnudan changed the title ~~**Sort out Cube DMA address dependenies**~~ **Sort out Cube DMA address dependencies** 8 hours ago

Assignees

No one—assign yourself

Labels

PicoZed
enhancement
help wanted
points=3
question
zedboard

Projects

SW kanban board

Backlog

Milestone

No milestone

Linked pull requests

Successfully merging a pull request may close this issue.

None yet

4 participants

Pin issue

BK opu-system Issue 125

Edit | New issue

Jump to bottom

Move kernel module files for Cube DMA to its own directory #125

Open **magnudan** opened this issue on May 15 · 1 comment

Labels: documentation, **enhancement**, help wanted, points=1

Projects: **SW kanban board**

magnudan commented on May 15

Is your feature request related to a problem? Please describe.
In `petalinux/projects/modules/` all kernel modules should be in their own sub-directory. When timestamping is merged with PR #122, the timestamp will be in its own sub directory in `modules/`, while the Cube DMA files will be in the root of `modules/`

Describe the solution you'd like
All modules in their own sub-directories in `petalinux/projects/modules/`

Describe alternatives you've considered
None. I really think this is the optimal way to organize modules

Note
I guess a lot of scripts must be updated with this issue

magnudan added documentation, **enhancement**, help wanted labels on May 15

JoarGjersund commented on May 15

Can easly be done. But remember to update the path in:

opu-system/petalinux/projects/load_software
Lines 24 to 43 in `b3dd9a9`

```
24    add_modules() {
25    #       petalinux-create -t modules -n testapp --enable
26    #       cp -fr ../modules/testapp.c project-spec/meta-user/recipes-modules/test
27    #       cp -fr ../modules/testapp.bb project-spec/meta-user/recipes-modules/tes
28
29
30            petalinux-create -t modules -n cubedma --enable
```

```
31          if echo $* | grep -e "pico" -q
32          then
33                  cp -fr ../modules/cubedma.c project-spec/meta-user/recipes-modu
34                  cp -fr ../modules/dma_parameters.h project-spec/meta-user/recip
```

👍 1

sivertba added the points=1 label 19 days ago

sivertba added this to **Backlog** in **SW kanban board** 19 days ago

Assignees

No one—assign yourself

Labels

documentation
enhancement
help wanted
points=1

Projects

SW kanban board

Backlog

Milestone

No milestone

Linked pull requests

Successfully merging a pull request may close this issue.

None yet

3 participants

Pin issue

BL test-mcus Pull Request 1

NTNU-SmallSat-Lab / **test-mcus** Private

PPS #1

Edit | Open with

Merged **magnudan** merged 5 commits into `master` from `pps` on May 13

Conversation 1 | Commits 5 | Checks 0 | Files changed 4 | +51 −2

magnudan commented on May 12

Basic PPS code, readme and pinout

magnudan added 5 commits on May 12

Initial commit for PPS mcu		cd35ce5
Changed folder name to follow Dennis' naming scheme		a58e0f1
Update README.md	Verified	80020cb
Moved .ino file, added pdf with pinout		3cc48e4
Merge branch 'pps' of github.com:NTNU-SmallSat-Lab/test-mcus into pps		d95755d

magnudan requested a review from **DennisNTNU** on May 12

DennisNTNU approved these changes on May 12 — View changes

DennisNTNU left a comment

Code is simple and seems sufficient.

magnudan merged commit **844fb08** into `master` on May 13 — Revert

magnudan deleted the `pps` branch on May 15 — Restore branch

Reviewers
DennisNTNU ✓

Assignees
No one—assign yourself

Labels
None yet

Projects
None yet

Milestone
No milestone

Linked issues
Successfully merging this pull request may close these issues.
None yet

2 participants

BM test-mcus Issue 3

Edit | New issue

Jump to bottom

Make the GSE-EPS power the Teensy GSE-PPS generator instead of power over USB #3

Closed **magnudan** opened this issue on May 13 · 7 comments

Assignees

Labels PPS enhancement points=5

Projects SW kanban board

magnudan commented on May 13

The Teensy 3.5 in the LidSat setup that provides the mock PPS signal is currently powered by USB from the Operations station. It's better to power it from the EPS because we can power cycle it, and it will have the same ground potential as everything else.

magnudan added enhancement PPS labels on May 13

magnudan assigned **rogerbirkeland** and **amundgj** and unassigned **rogerbirkeland** on May 13

sivertba commented on May 14

Need cable and connector

sivertba added the points=5 label on May 14

sivertba added this to **Backlog** in **SW kanban board** on May 14

sivertba moved this from **Backlog** to **To do** in **SW kanban board** on May 14

rogerbirkeland commented 22 days ago

@**amundgj** Can you make a cable that is EPS-picolock-output terminated in one end, and either USB or just single leads header (female) in the other end?

magnudan changed the title ~~**Make the EPS power the Teensy PPS generator instead of power over USB**~~ **Make the GSE-EPS power the Teensy GSE-PPS generator instead of power over USB** 17 days ago

amundgj commented 17 days ago

I will terminate a GSE-PPS power cable with a USB micro-A male connector in one end, and a 4-pin PicoLock in the other. Where to connect on the EPS? Which output?

magnudan commented 17 days ago

> Where to connect on the EPS? Which output?

It will have to be one of the 5V outputs, but I don't know which are suitable or available. @**Roger** or @**DennisNTNU** might know

rogerbirkeland commented 17 days ago

NA has upgraded the EPS, and it seems like its a bug in the print-out of
the output channel voltages. But consult the datasheet and take any of the
5 V ones that are free. Doesnt matter which one. If memory serves, its
O1-O3?.
...

amundgj commented 6 days ago

I have cannibalized a USB cable and left the USB micro-B in on end and terminated the other end in 4-pin PicoLock. Only VCC (PicoLock pin1) and GND (PicoLock pin4) are terminated. The cable has been put into use. It is connected to EPS connector O1, and is now supplying the Teensy. Issue is closed.

🎉 2

amundgj closed this 6 days ago

SW kanban board automation moved this from **To do** to **Done** 6 days ago

rogerbirkeland commented 6 days ago

I will fix EPS startup config so it's auto on.
...

Assignees

rogerbirkeland

amundgj

Labels

PPS

enhancement

points=5

Projects

SW kanban board

Done

Milestone

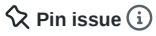
No milestone

Linked pull requests

Successfully merging a pull request may close this issue.

None yet

4 participants


Pin issue

BN test-mcus Issue 4

Edit | New issue

Jump to bottom

Make a better cable setup for the PPS signal #4

Closed **magnudan** opened this issue on May 13 · 2 comments

Assignees	
Labels	PPS enhancement points=3
Projects	SW kanban board

magnudan commented on May 13

On the Teensy side, the connector a standard 100 mil (unsure if this is the right term) male pin. On the BOB side, the connector is a 2-pin picolock.

Now, they're connected with a standard 100 mil cable to a 2-way wago clamp to a picolock cable without the connector. This is just fastened on one of the pins in the picolock housing on the BOB.

Figure will be added

magnudan added **enhancement** PPS labels on May 13

magnudan assigned **rogerbirkeland** and **amundgj** on May 13

sivertba added the points=3 label on May 14

sivertba added this to **Backlog** in **SW kanban board** on May 14

evelynlimore moved this from **Backlog** to **To do** in **SW kanban board** on May 14

magnudan commented on May 15

@**amundgj**: Do you think the solution you made is sufficient to close this issue?

amundgj commented 29 days ago

> **@amundgj**: Do you think the solution you made is sufficient to close this issue?

Yes, I think the setup now is sufficient. Room for improvement as always, but good enough.

❤️ 1

amundgj closed this 29 days ago

SW kanban board automation moved this from **To do** to **Done** 29 days ago

Assignees

rogerbirkeland

amundgj

Labels

PPS

enhancement

points=3

Projects

SW kanban board

Done

Milestone

No milestone

Linked pull requests

Successfully merging a pull request may close this issue.

None yet

4 participants

Pin issue

BO test-mcus Issue 5

Edit | New issue

Jump to bottom

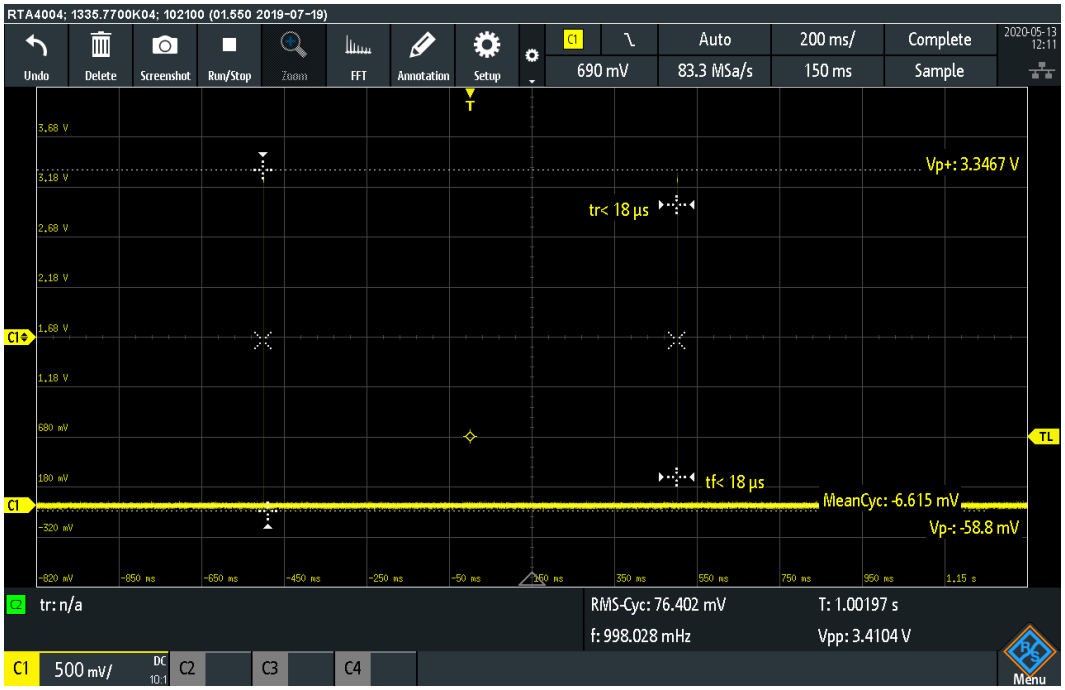
Improve accuracy of GSE-PPS mock signal #5

Open **magnudan** opened this issue on May 15 · 2 comments

Labels	PPS **enhancement** points=5
Projects	**SW kanban board**

magnudan commented on May 15

NA has an accuracy of <1ns on the period of the PPS signal from the ADCS. The following screenshot from the oscilloscope in the lab shows that the signal generated by the Teensy has a period of 1.00197 s, so 2 ms off.



I suspect that we won't reach NA's accuracy, and I don't think we should aim for that, but we can improve what we have quite a bit.

The current implementation uses a sleep function to time the signal. This is not optimal. I think we should use an interrupt based timer instead.

magnudan added **enhancement** PPS labels on May 15

magnudan commented on May 15

Check the accuracy of the oscilloscope

magnudan mentioned this issue on May 15

How do we test correctness of timestamping? NTNU-SmallSat-Lab/hypso-sw#242 Open

sivertba changed the title ~~**Improve accuracy of PPS signal**~~ **Improve accuracy of GSE-PPS mock signal** 19 days ago

sivertba added the points=5 label 19 days ago

sivertba added this to **Backlog** in **SW kanban board** 19 days ago

sivertba commented 19 days ago • edited

A cheap GPS module can give us a more real PPS signal.
Make into a future issue

Assignees

No one—assign yourself

Labels

PPS
enhancement
points=5

Projects

SW kanban board

Backlog

Milestone

No milestone

Linked pull requests

Successfully merging a pull request may close this issue.

None yet

2 participants

Pin issue

BP Listings

The listings in this section do not necessarily represent code in a repository, or code that will work (or compile) as a standalone feature. Formatting might not adhere to HYPSO standards, as it has been fitted to a physical format. The code should be understood in the context of the section it is referred from.

BP.1 Set Cube DMA Registers

Listing 1: Example of how the configuration registers of Cube DMA can be set

```
#define CUBE_DEPTH  107 //Spectral columns
#define CUBE_HEIGHT 500 //Number of frames
#define CUBE_WIDTH  720 //Spatial rows
#define CUBE_SIZE (CUBE_DEPTH * CUBE_HEIGHT * CUBE_WIDTH)

#define SEND_PHYS_ADDR      0x30000000
#define RECEIVE_PHYS_ADDR   0x38000000
#define CUBEDMA_BASE_ADDR   0x43C00000

//deviceMem[0] points to CUBEDMA_BASE_ADDR
#define cubedma_RegWrite(reg, value) deviceMem[reg] = value;

typedef enum
{
    MM2S_CTRL_REG = 0,  //0x00
    MM2S_STAT_REG,      //0x04
    MM2S_ADDR_REG,      //0x08
    MM2S_CUBE_DIM_REG,  //0x0C
    MM2S_BLOCK_DIM_REG, //0x10
    MM2S_ROW_DIM_REG,   //0x14

    S2MM_CTRL_REG = 8,  //0x20
    S2MM_STAT_REG,      //0x24
    S2MM_ADDR_REG,      //0x28
    S2MM_LEN_REG        //0x2C
} reg_t;

typedef enum
{
    MM2S,
    S2MM
} transfer_t;

void cubedma_configure(void)
{
    /*
        Setting the dimensions of the cube in the
        correct bits according to the documentation
    */
    uint32_t dims =
        (CUBE_WIDTH  & 0xFFF) << 0  |
        (CUBE_HEIGHT & 0xFFF) << 12 |
        (CUBE_DEPTH  & 0xFF)  << 24;

    cubedma_RegWrite(MM2S_ADDR_REG, (uint32_t)(SEND_PHYS_ADDR));
    cubedma_RegWrite(MM2S_CUBE_DIM_REG, dims);
    // We don't use blocks:
    cubedma_RegWrite(MM2S_BLOCK_DIM_REG, 0x0);
    cubedma_RegWrite(MM2S_ROW_DIM_REG, CUBE_WIDTH * CUBE_DEPTH);
    cubedma_RegWrite(S2MM_ADDR_REG,
        (uint32_t)(RECEIVE_PHYS_ADDR));
}
```

BP.2 HSI Camera Configure

Listing 2: Function to configure the HSI camera

```
bool hsi_camera_configure(void)
{
    UINT pixelClockRange[3];

    // Temporary assignments:
    double exposure = 10;
    double fps = 10;
    aoiRows = 720;
    aoiCols = 1280;
    INT hardwareGain = 50;

    INT nRet;

    nRet = is_PixelClock(camera, IS_PIXELCLOCK_CMD_GET_RANGE,
                         (void*)pixelClockRange, sizeof(pixelClockRange));
    if(!is_ueye_error(nRet, "PixelClock"))
    {
        nRet = is_PixelClock(camera, IS_PIXELCLOCK_CMD_SET,
                             (void*)&pixelClockRange[1], sizeof(pixelClockRange[1]));
        is_ueye_error(nRet, "PixelClock");
    }

    nRet = is_SetDisplayMode(camera, IS_SET_DM_DIB);
    is_ueye_error(nRet, "SetDisplayMode");

    UINT resolutionMode = 36; // Full resolution
    UINT nSizeOfParam = 4;
    nRet = is_ImageFormat(camera, IMGFRMT_CMD_SET_FORMAT,
                          &resolutionMode, nSizeOfParam);
    is_ueye_error(nRet, "ImageFormat");

    IS_RECT aoiParameters;
    int image_x_offset_sensor = (SENSOR_WIDTH - aoiCols)/2;
    int image_y_offset_sensor = (SENSOR_HEIGHT - aoiRows)/2;
    aoiParameters.s32X     = image_x_offset_sensor | IS_AOI_IMAGE_POS_ABSOLUTE;
    aoiParameters.s32Y     = image_y_offset_sensor | IS_AOI_IMAGE_POS_ABSOLUTE;
    aoiParameters.s32Width = aoiCols;
    aoiParameters.s32Height = aoiRows;

    nRet = is_AOI(camera, IS_AOI_IMAGE_SET_AOI,
                  (void*)&aoiParameters, sizeof(aoiParameters));
    is_ueye_error(nRet, "AOI");

    nRet = is_Exposure(camera, IS_EXPOSURE_CMD_SET_EXPOSURE,
                       (void*)&exposure, sizeof(exposure));
    is_ueye_error(nRet, "Exposure");

    nRet = is_SetColorMode(camera, IS_CM_MONO12);
    is_ueye_error(nRet, "SetColorMode");

    nRet = is_SetGainBoost(camera, IS_SET_GAINBOOST_ON);
    is_ueye_error(nRet, "SetGainBoost");

    nRet = is_SetHardwareGain(camera, hardwareGain, IS_IGNORE_PARAMETER,
                              IS_IGNORE_PARAMETER, IS_IGNORE_PARAMETER);
    is_ueye_error(nRet, "SetHardwareGain");

    double newFPS;
    nRet = is_SetFrameRate(camera, fps, &newFPS);
    if(!is_ueye_error(nRet, "SetFrameRate"))
    {
        double diff = fabs(fps - newFPS);
        if(diff > 0.5)
        {
            fprintf(stdout, "    FPS changed from %f to %f\n", fps, newFPS);
        }
    }
    fprintf(stdout, "    Camera parameters set!\n");
    return 1;
}
```

BP.3 HSI capture

Listing 3: Case in switch statement to capture HSI frames

```
case HSI_CMD_CAPTURE:
    printf("Received capture request\n");
    int numFrames_higher = packet->data[1];
    int numFrames_lower = packet->data[2];
    int numFrames = numFrames_lower + (numFrames_higher << 8);

    if(hsi_camera_init())
    {
        if(hsi_camera_configure())
        {
            hsi_capture(numFrames, 1, 0, 1);
            if hsi_store_comp_cube(numFrames, 1);
            }
                hsi_camera_deinit();
            }
        }
    }
break;
```

BP.4 Modularized HSI camera functions

Listing 4: Header file for camera interfacing after modularization pull request

```
#ifndef _HSI_SERVICE_H_
#define _HSI_SERVICE_H_

#include <ueye.h>

#include "csp/csp.h"

/*
    This module contains all functions that interface
    with the HSI camera, or are directly related to
    it. The top level functions are:
        - hsi_capture_and_binn_cube()
        - hsi_debug_capture_and_binn_cube()
        - hsi_camera_gettemp()
    These can be run independently.
    Functions that interface directly with the HSI camera:
        - hsi_camera_init()
        - hsi_camera_deinit()
        - his_camera_configure()
        - hsi_camera_start_capture()
        - hsi_camera_stop_capture()
        - hsi_camera_get_frame()
        - hsi_camera_store_frame_as_png()
    These function must be run in a specific order
    to work properly. Refer to the top level
    functions for examples.
*/


/*
    Camera sensor parameters
*/
#define SENSOR_WIDTH    1936
#define SENSOR_HEIGHT   1216

/*
    Set the number of full (1936 by 1216 pixels) frames
    the HSI camera is able to store in memory before
    being processed by the HSI service.
*/
#define FRAME_BUFFER_SIZE 10

/*
    These two functions will use the HSI camera to capture
    a given number of frames. Each frame is binned in the
    spectral dimension after capture.The debug version
    will in addition store each frame as both .png and
    .raw, and the full cube as .raw.
    hsi_config: struct with configurations for the hsi camera
    binned_cube: pointer to memory location of the binned cube
    binned_bytes: number of bytes in the binned cube
    Returns:
        - 0 if successful
        - 1 if unable to initialize
        - 2 if unable to configure
        - 3 if unable to start capture
*/
int hsi_capture_and_binn_cube(struct hsi_config_s* hsi_config,
        uint16_t* binned_cube, unsigned int* binned_bytes);
int hsi_debug_capture_and_binn_cube(
        struct hsi_config_s* hsi_config,
        uint16_t* binned_cube, unsigned int* binned_bytes);

/*
    Gets the temperature of the HSI camera
    Prints other info for debug purposes
    tempur: points to temperature variable
*/
bool hsi_camera_gettemp(float* temptur);
```

```
/*
    Initializes the HSI camera.
    Must be run before any of the following functions.
*/
bool hsi_camera_init(void);

/*
    Exits the HSI camera.
    Must be run when camera interfacing is over.
*/
bool hsi_camera_deinit(void);

/*
    Configures camera with given parameters.
    Must be run before capture
*/
bool hsi_camera_configure(struct hsi_config_s* hsi_config);

/*
    These two functions starts and stops the capture with a
    initialized and configured HSI camera
    hsi_config: struct with configurations for the hsi camera
    frame_buffers: buffer where the iDS API stores raw
        frame data for up to FRAME_BUFFER_SIZE frames
    frame_buffer_IDs: array where the iDS API stores the ID of
        frames in frame_buffers
*/
bool hsi_camera_start_capture(struct hsi_config_s* hsi_config,
        char** frame_buffers,
        int* frame_buffer_IDs);
void hsi_camera_stop_capture(char** frame_buffers,
        int* frame_buffer_IDs);

/*
    Gets the oldest frame in the frame buffer.
    Must be run after hsi_camera_start_capture().
    Must be followed by is_UnlockSeqBuf() to make frame
    available for writing again.
    raw_frame: pointer to memory location
        of the current raw frame
    frame_buffer_IDs: array where the iDS API stores the
        ID of frames in frame_buffers
*/
bool hsi_camera_get_frame(char** raw_frame,
        int* frame_buffer_IDs);

/*
    Stores the current frame in .png format.
    frame_number: the number of the current frame in the
        capture sequence Used for file name
    raw_frame: pointer to memory location of the
        current raw frame
    frame_ID: the ID of the current frame
*/
bool hsi_camera_store_frame_as_png(int frame_number,
        char** raw_frame, int* frame_ID);
#endif
```

BP.5 Cube DMA Timeout Code

Listing 5: Function that checks transfer status of the Cube DMA

```
//Arbitrary large number
#define TIMEOUT 0xFFFFFF

bool compress_cube_fpga(unsigned int* compressed_bytes)
{
    cubedma_start_transfer();

    bool MM2S_success = 0;
    bool S2MM_success = 0;
    volatile uint32_t time;

    for (time = 0; time < TIMEOUT; time++)
    {
        if (cubedma_transfer_done(MM2S))
        {
            printf("MM2S completed after %i \"times\"\n", time);
            MM2S_success = 1;
            break;
        }
    }

    if (time == TIMEOUT)
    {
        printf("ERROR: MM2S timeout\n");
    }

    for (time = 0; time < TIMEOUT; time++)
    {
        if (cubedma_transfer_done(S2MM))
        {
            printf("S2MM completed after %i \"times\"\n", time);
            S2MM_success = 1;
            break;
        }
    }
    if (time == TIMEOUT)
    {
        printf("ERROR: S2MM timeout\n");
    }

    if (MM2S_success && S2MM_success)
    {
        printf("Transfer success\n");
        *compressed_bytes = cubedma_get_received_length();
        return 0;
    }
    else
    {
        printf("CubeDMA transfer not successfull.\n");
        return 1;
    }
}
```

Listing 6: Function that checks cube DMA transfer registers

```
#define SR_DONE_MSK 0x1

#define STAT_REG_OFFSET(transfer) (transfer==MM2S)? \
    (MM2S_STAT_REG) : (S2MM_STAT_REG)

bool cubedma_transfer_done(transfer_t transfer)
{
    if (cubedma_RegRead(STAT_REG_OFFSET(transfer)) & SR_DONE_MSK)
    {
        if (transfer == S2MM)
        {
            cubedma_RegWrite(S2MM_CTRL_REG, 0x0);
            // Assuming that S2MM is the last transfer to finish
            flush_cache();
        }
        else
        {
            cubedma_RegWrite(MM2S_CTRL_REG, 0x0);
        }
        return true;
    }
    return false;
}
```

BP.6 Configuring Flash Return

Listing 7: Configuration of flash return for HSI camera

```
printf("    Enabling flash signal\n");
UINT flash_mode = IO_FLASH_MODE_FREERUN_LO_ACTIVE;
nRet = is_IO(camera, IS_IO_CMD_FLASH_SET_MODE,
      (void*)&flash_mode, sizeof(flash_mode));
if (!is_ueye_error(nRet, "is_IO; Setting flash mode"))
{
IO_FLASH_PARAMS flash_params;

    //Start flash signal at start of exposure
    flash_params.s32Delay = 0;
    //Set duration of flash signal equal to exposure time
    //Exposure is in ms, while flas duration is in us
    flash_params.u32Duration = hsi_config->exp * 1000;

    printf("    Setting parameters\n");
    printf("    Delay: %d us\n", flash_params.s32Delay);
    printf("    Duration: %d us\n", flash_params.u32Duration);

    nRet = is_IO(camera, IS_IO_CMD_FLASH_SET_PARAMS,
            (void*)&flash_params, sizeof(flash_params));
    is_ueye_error(nRet, "is_IO; Setting flash parameters");

    nRet = is_IO(camera, IS_IO_CMD_FLASH_GET_PARAMS,
            (void*)&flash_params, sizeof(flash_params));
    if (!is_ueye_error(nRet, "is_IO; Getting flash parameters"))
    {
        printf("    Reading flash parameters back from camera\n");
        printf("    Delay: %d us\n", flash_params.s32Delay);
        printf("    Duration: %d us\n", flash_params.u32Duration);
    }
}
```

BP.7 Source Code for GSE-PPS Generator

Listing 8: Arduino source code for GSE-PPS signal

```
/*
  PPS
  Sets the PPS signal high for 500 microseconds,
  then low for 999500 microseconds.
  This gives the signal a period of 1 second

  The signal is set to use the same pin as
  the on-chip LED to give visual feedback
 */

int PPS = 13;

// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(PPS, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(PPS, HIGH);
  delayMicroseconds(500);
  digitalWrite(PPS, LOW);
  delayMicroseconds(999500);
}
```

Magnus Danielsen

Integration and Testing of a HSI Processing Pipeline

NTNU
Kunnskap for en bedre verden