



PAST

Design Concepts and Overview

CubeSat Camera System (C2S)

Abbott F. — Perth Aerospace Student Team

ADCS Department — CubeSat 1 Mission

This document presents all the considerations and processes towards developing the CubeSat Camera System, an undergraduate student-based project as part of the Perth Aerospace Student Team. The project served as a strong platform to develop sophisticated understandings of various camera subsystems and succeed in refining skills of the Curtin Graduate Capabilities Matrix.

Acknowledgements

This document follows the IEEE referencing guide according to [17] [1] [8].

Summary

Contents

1	Introduction	1
1.1	Background	1
1.2	Project Scope	1
1.3	Project Milestones	2
1.4	Project Responsibilities	2
1.5	Timeline	2
2	Project Planning and Kick-off	4
2.1	System Requirements	4
2.2	Component Selection	4
2.2.1	Local Power Supply	4
2.2.2	Imaging Sensor	4
2.2.3	Lens	5
2.2.4	Image Processor	6
2.2.5	SD Card Reader	8
2.2.6	External Clock	8
3	C2Sv0 (Shelved)	9
3.1	Remarks	9
3.2	Schematic Design & Component Selection	10
3.2.1	Overview	10
3.2.2	Processor	10
3.2.3	Image Sensor	10
3.2.4	Clock	11
3.2.5	uSD Card	11
3.2.6	LPS	11
3.2.7	SWD Connector	12
3.2.8	USB-C Connector	12
3.2.9	Power Sequencer	13
3.3	PCB Layout and Review	14
3.3.1	Selecting Manufacturers	14
3.3.2	Current Carrying Capacity	14
3.3.3	BGA Fanout	14
3.3.4	CSI-2 Routing	14
3.3.5	Debug LEDs	15
4	C2Sv0.1	16
4.1	Remarks	16
4.2	Schematic Design & Component Selection	17
4.2.1	Overview	17
4.2.2	Image Sensor	18
4.2.3	MCU	19
4.2.4	Memory	20
4.2.5	USB-C	21

4.2.6	uSD	21
4.2.7	LPS	22
4.2.8	TAG	22
4.3	PCB Layout and Review	22
4.3.1	Overview	22
4.3.2	Component Placement	22
4.3.3	Routing	23
4.3.4	Power Distribution	23
4.3.5	Vias	23
4.3.6	Ground Fill	24
4.3.7	Power Pad	24
4.3.8	Coupling	24
4.3.9	Silkscreen	25
4.3.10	Mounting	25
4.3.11	Fiducial Markers	28
5	Manufacturing v0.1	29
5.1	Miscellaneous	29
5.2	Component Placement	29
5.2.1	Sensor Board	30
5.2.2	Processor Board	30
5.3	AT&MWA Placement	31
6	Programming v0.1	32
6.1	Initial Testing	32
6.1.1	LEDs	32
6.1.2	Audio Buzzer	32
6.1.3	Rotary Encoder	32
6.1.4	Current Sense Amplifier	33
6.1.5	S-R Latch	34
6.1.6	Buttons	35
6.1.7	uSD	35
References		36
Appendices		38
A	Supporting Calculations	38
A.1	C-L-C Values	38
A.2	Transconductance of Oscillator	38
A.3	Power Draw Estimation	38
A.4	Stitching Via Calculations	39
B	C2Sv0 Breakout Board	40
B.1	PCB Layout	40

C C2Sv0.1 Breakout Board	41
C.1 STM32U5G Power Management	41
C.1.1 Power Supplies	41
C.1.2 Decoupling Capacitors	42
C.2 Supplementary Circuits	44
C.3 Pin Justification Tables	44
C.4 PCB Layout	48
D Datasheet Extracts and Ratings	49
D.1 Processors	49
D.1.1 STM32N6	49
D.1.2 STM32F7	50
D.1.3 STM32U5	51
D.2 Image Sensors	51
D.2.1 AR0141 Sensor	51
D.3 Memory	52
D.3.1 IS42S16160J Memory	52
D.4 Power Components	53
D.4.1 TCR2EF LDO	53
List of Figures	54
List of Tables	55

1 Introduction

1.1 Background

The **CubeSat Camera System (C2S)** is the proposed mission payload for the Perth Aerospace Student Team's (PAST) first CubeSat. This inaugural mission represents a critical milestone in the team's broader goal of developing low-cost, modular spacecraft capable of performing meaningful scientific and engineering demonstrations in Low Earth Orbit (LEO). C2S is designed to operate as an Earth observation payload, taking photos of certain land- and oceanscapes. Beyond its imaging objectives, the system will serve as a testbed for evaluating the viability of student-developed space hardware in a real orbital environment. The payload aims to balance performance, reliability, and resource constraints typical of CubeSat-class missions.

This document outlines the key design drivers, system architecture, component selection strategies, and trade-offs that inform the development of the CubeSat Camera System prototype. It also provides an overview of the integration challenges and planned validation strategies for the final model.

Where appropriate, this document often refers to the preliminary literature review which can be found [here](#). If there is assumed knowledge from the research, it will be mentioned e.g., "see literature review for breakdown".

1.2 Project Scope

The primary objective of this project is to design, develop, and demonstrate a standalone CubeSat-compatible camera system prototype. The prototype aims to meet key performance and integration requirements for future low Earth orbit missions, with a focus on modularity, efficiency, and feasibility within typical CubeSat constraints. The system must operate independently to capture, process, and store image data using commercially available components.

The specific scope includes:

- Acquisition of RAW image data from a CMOS image sensor with pixel binning capabilities to enhance low-light performance and reduce onboard processing demands.
- Onboard conversion of RAW data to compressed JPEG format in the Y'UV colour space. Chroma subsampling will be employed to reduce image file size while preserving perceptual image quality, which is critical for bandwidth and storage-constrained space environments.
- Integration of an uSD card breakout board to store processed images, enabling straightforward data access during ground testing and allowing for future adaptation to in-flight non-volatile memory solutions.
- Implementation of a fixed focal length lens system to simplify the optical design, reduce mechanical complexity, and optimise image consistency. Lens and sensor pairing will be chosen based on trade-offs between resolution, field of view, and compatibility with CubeSat mechanical form factors.
- System design intended for eventual microcontroller or FPGA control, with all power, data, and interface components selected to support future embedded control integration.

The following features are intentionally excluded from the current development phase to maintain focus and ensure feasibility within available time, budget, and resource constraints:

- Wireless or wired image transmission beyond the camera module, such as radio downlink or USB tethering.
- Integration with full CubeSat systems, including EPS, onboard computers, or attitude control systems.
- Design or implementation of redundancy, fault tolerance, or radiation hardening beyond basic best practices for PCB layout and component selection.
- In-orbit deployment or qualification testing (e.g., thermal vacuum, vibration).

This scope defines a foundational platform from which future iterations may extend toward full CubeSat integration and space qualification.

1.3

Project Milestones

The project has been divided into five key-phases that each act as sub-projects and each lead into the next. This helps compartmentalise the project for better management and tracking.

ID	Milestone
1	Design and build a fully custom CubeSat-compatible prototype camera module, including PCB layout, component selection, and sensor integration.
2	Develop firmware to configure and control the image sensor (e.g., via I ² C/SPI), supporting raw image capture and tuning of parameters like exposure, gain, and white balance.
3	Implement an image pipeline to convert raw sensor data into usable image formats, including demosaicing, gamma correction, and compression.
4	Design a mechanical housing that meets CubeSat volume, mass, and thermal constraints, and could interface with the rest of the CubeSat.
5	Test and validate the camera module under simulated space conditions (thermal cycling, vibration, and low light).

Table 1: Milestone objectives for the duration of the project

1.4

Project Responsibilities

The prolonged timeline and importance of the project has shown that it is critical to delegate parts of the project to other PAST members. For the software and mechanical aspects of the project, the milestone has been delegated to other members. The following table highlights the milestone each member is responsible for.

Name	Department	Milestone Responsibility
Felix Abbott	Electrical	1, 2, 3, 5
TBD	Software	2, 3
TBD	Mechanical	4

Table 2: Breakdown of project responsibilities

1.5

Timeline

The timeline provides a clear schedule to track milestone completion, outlining which project phase should be active each month and the expected duration of each phase to ensure steady progress.

Date	Phase	Key Tasks	Relevant Objective ID
Aug '25	Project Planning & Kick-off	Define system requirements; Select image sensor; Preliminary architecture planning; Initial parts research	1
Sep–Nov '25	Schematic Design & Component Selection	Design camera circuit schematic (breakout board); Select power regulation, oscillators, EEPROM, etc.; Create block diagrams	1
Dec '25 – Jan '26	PCB Layout & Review	Begin PCB layout; Design for manufacturability (DFM); Include test points, headers, debugging pads. Begin investigating camera lens	1
Feb '26	PCB Fabrication & Assembly	Send board to fabricate and order components; Begin manufacturing	1
Mar '26	Firmware: Sensor Bring-Up	I ² C/SPI communication tests; Basic sensor initialisation; Test register writes/reads	2
Apr–May '26	Image Processing Pipeline	Capture raw Bayer data; Implement demosaicing, gain, white balance; Export images via USB/SD/Wi-Fi	2, 3
Jun–Jul '26	Mechanical and PCB Prototype	Design simple 3D-printed housing; Consider heatsinking or fan-based thermal dissipation; Design lens mount if needed. Alongside this, developing the PCB layout for the prototype board (not a breakout).	4
Aug–Oct '26	Functional Testing	Send board to fabricate and source any further components; Manufacture board once arrived. Test camera stability, power usage, image quality under varied lighting; Conduct thermal monitoring; Log long-duration operation	5
Nov '26	Optimisation	Tune firmware; Improve image quality; Reduce power and memory usage; Prepare documentation	2, 3
Dec '26	Final Integration & Reporting	Compile results; Document all hardware/software; Prepare final presentation/demo/report	

Table 3: Project schedule and key tasks

2 Project Planning and Kick-off

This phase focuses on establishing the foundation for the project by defining system requirements, selecting the image sensor, planning the preliminary architecture, and conducting initial parts research.

Please note. Although certain components are investigated, I only focus on key-components like the sensor and processor. Resistors, capacitors, debug LEDs and other components mentioned in datasheets (like recommended circuits) are not discussed. For more information, please see subsection 3.2

This phase was reviewed by Dr. Robert Howie on August 12th, 2025.

2.1 System Requirements

After the literature review, the following requirements for C2S are established:

Requirement	Description
1	A pixel resolution of 500-750 metres was chosen to capture major urban areas, such as Perth, in a single frame. This resolution balances ground coverage and system simplicity by avoiding deployable or extendable lenses.
2	A fixed focal length lens simplifies the optical system and ensures consistent sharpness at the set focal distance. This reduces complexity, mass, and power needs, making it suitable for compact CubeSat prototypes.
3	The system will use YUV format with chroma subsampling and JPEG compression to reduce file size while maintaining image quality, optimising bandwidth for CubeSat transmission.
4	A CMOS image sensor was chosen for its lower power use, faster readout, and easier digital integration compared to CCDs. This suits compact, power-sensitive platforms like CubeSats.
5	If the imaging sensor exceeds 1MP, pixel binning will allow switching between high-resolution and lower-resolution images with improved quality through increased light sensitivity and reduced noise.
6	Although lenses won't be investigated in this milestone, the breakout board must allocate space for M12 lenses to mount for later milestones.

Table 4: C2S system requirements as of August '25, Phase 1.

2.2 Component Selection

Please note, the following components are not heavily optimised for the system and can be improved. They were selected because they fit the requirements and scope of the breakout board. Further investigation into components will be conducted for the prototype.

2.2.1 Local Power Supply

Opposed to a switching converter, an LDO was chosen because of the cost and fewer external components required. Since the required power lines will most likely come from other projects like EPS, LDOs were a great choice for cheap and compact local power supplies on the breakout board.

The [TCR3DF](#) series Linear Drop-Off Voltage Regulators (LDOs) were selected as the local power supplies for the breakout board because I could source all of the required power lines from the same series. An extensive search for LDOs was not conducted since the first results met the criteria and I did not see a need to further optimise the component selection.

2.2.2 Imaging Sensor

The main image sensor picked out for the breakout board was the [MIRA220](#) (datasheet found [here](#)). It is a widely used image sensor, particularly in applications requiring high-speed, low-light, and near-infrared (NIR) imaging. Its key features include:

- 2.2 MP NIR-enhanced global shutter CMOS sensor, ideal for capturing motion without distortion and low-light near-infrared applications (see literature review for breakdown).
- It offers a flexible bit depth (8, 10, or 12 bits) and supports up to 90 fps at full resolution, allowing for high-quality imaging and fast frame capture.
- The sensor is available in monochrome, RGB, and RGBIR variants, enabling both colour and NIR-sensitive imaging options.
- Compact package and MIPI CSI-2 interface simplify board integration and compatibility with modern processors and FPGAs.
- Moderate power consumption (350 mW active) balances performance with CubeSat power constraints.
- Pixel size of 2.79 μm provides a good balance between resolution and light sensitivity for CubeSat imaging needs.
- Its global shutter with correlated double sampling reduces noise and improves image quality under variable lighting conditions.

While other sensors may offer advantages such as lower power consumption or smaller pixel sizes, the MIRA220 meets the key requirements for the breakout board, particularly its NIR sensitivity and global shutter capabilities which are valuable for enhanced surface imaging. Using a more specialised or lower-power sensor at this stage would significantly increase costs and design complexity, making the MIRA220 a practical choice for prototype development with room for upgrades in subsequent revisions.

Although the MIRA220 uses a different interface than LVDS, it maintains low power and complexity, which supports efficient prototyping without compromising image quality or system integration. It has native CSI-2, which means it can easily interface with the selected image processor [subsubsection 2.2.4](#) without any external components like LVDS drivers and receivers that consume significant amounts of power. Moreover, AMS Osram have open sourced their camera board designs ([found here](#)), which means the reference schematic and PCB will make the development process much smoother and less prone to errors.

2.2.3 Lens

This section contains information relevant to the literature review.

There's a clear trade-off between performance and zoom. Consider a single pixel of size d with a focal point located at a distance f from the pixel. The angle formed between the pixel's centre and its top edge changes depending on the pixel's height. Now, imagine extending lines from the pixel's edges through the focal point out for several hundred kilometres. As the pixel size grows, the area it covers on the Earth's surface also expands. Therefore, to maintain a constant spatial resolution when increasing the pixel size, the focal length must be increased accordingly.

The obvious limitations to our image are the file size and CubeSat size. I cannot be transmitting 2.2MP images down to Earth, so I must bin down or crop the image. If I bin down by n times, then the pixel size and focal length increases by n^2 . If I place a hard limit of 1MP on the pixel count, then I only need a focal length of <10mm. If the sensor is binned once (2.224MP \rightarrow 0.5MP, $2.79\mu\text{m} \rightarrow 11.16\mu\text{m}$) to improve performance and reduce file size, the required focal length for the camera is: $f = hd/s$ where d is the pixel size, h is the orbital height and s is the spatial resolution. I find that the expected focal length for a M12 lens would be 8.9mm for a spatial resolution of 500m at an altitude of 400km. This would capture an image 400km x 350km across. This fits within our (assumed) limits, however, I may adjust the scope towards the 750m spectral resolution range to comfortably sit within size constraints (depending on the results of testing).

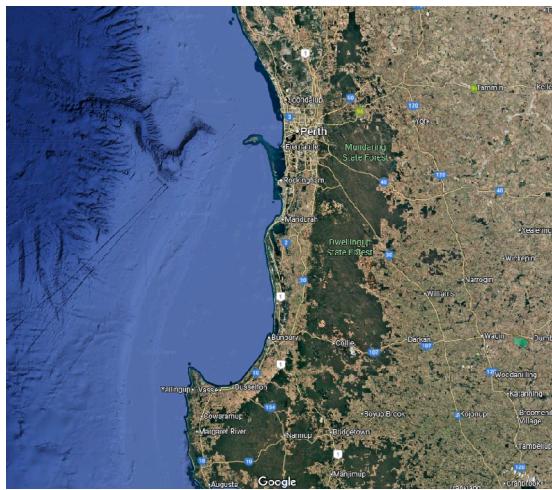


Figure 1: Expected image coverage and resolution for the MIRA220 sensor, binned once with a focal length of 8.9mm.

2.2.4

Image Processor

A lot of consideration was placed into the selection of the image processor. Initially, I investigated ASICs suited for image processing. I found that most active and supported imaging ASICs are manufactured for larger scopes i.e., video and AI processing, which exceeded the current power and thermal requirements of the project. Most ASICs that fit the current scope are more dated and less-supported. For example, the ADSP-BF70x line were prime candidates at the start of the project since their attributes fit the current scope and aren't "overkill". They processed low-resolution images at a high-energy efficiency, could utilise external RAM and interfaced with CMOS sensors with PPI. Despite this, the line (as of mid-2025) is more than a decade old and has limited support. Other processors from NXP, Microchip and Texas Instruments either showed the same limited support or lacked imaging processing capabilities (most designed for power systems). After consulting with online forums like [r/ECE](#) and [r/embedded](#), as well as with Dr. Robert Howie regarding his students' previous star tracker project, it was decided that a general processor like the STM32 was best suited for the current application.

STM32s have been widely used in CubeSat applications over the years and have become a widely supported platform for amateur aerospace applications. Unlike FPGAs, which can compute parallel processing, STM32s use high-level languages like C/C++, have faster development cycles and have a larger community. They are typically more power efficient and can have built-in image processing and encoders. Amongst the STM32 line, I selected the STM32N6 as the image processor over others like the H7 for the following reasons:

- Supports DCMI as well as MIPI CSI-2. H7s do not support the latter.
- Uses an image signal processor (ISP), digital signal processor (DSP) and neural processing unit (NPU) for processing unlike a traditional software-powered CPU.
- Has hardware-accelerated subsampling techniques, whereas others use software.
- Has a higher power efficiency compared to the H7 (and can be underclocked to save more power).
- More recent and will be supported for longer.

The following table explores some of the key differences between the STM32 processors:

Feature	STM32N6 Series	STM32H7 Series	STM32F4 Series
Core	Arm Cortex-M55 + Neural ART NPU	Arm Cortex-M7 (single or dual-core M7/M4)	Arm Cortex-M4
Performance	Up to 480 MHz (core), includes vector extensions	Up to 550 MHz (M7), strong DSP/float perf	Up to 180 MHz, moderate DSP capability
Memory (RAM/Flash)	Up to 4 MB SRAM, 4 MB Flash	Up to 1 MB SRAM, 2 MB Flash	Up to 256 KB SRAM, 2 MB Flash
Imaging Interfaces	DCMI, SPI, QuadSPI, JPEG Codec, Chrom-ART, DMA2D	DCMI, JPEG Codec, Chrom-ART, DMA2D	DCMI (on select models), basic DMA
Subsampling / Encoding	Hardware JPEG with native Y'UV support	Hardware JPEG + DMA2D for YUV	Software-based only
Power Efficiency	Moderate to High (based on workload + NPU efficiency)	Moderate (high-performance, higher dynamic power)	Good (lower performance = lower power)
Target Applications	AI vision, edge inference, intelligent sensors, imaging pipelines	High-performance imaging, audio, motor control, GUI apps	General-purpose MCU, audio, lower-end imaging
Availability	Early access or newly launched (as of 2025)	Mature and widely supported	Legacy but stable

Table 5: Comparison of STM32N6, STM32H7, and STM32F4 Microcontroller Families

I selected the [STM32N657I0H3Q](#) as the image processor for the breakout board. I opted for this processor amongst its other counterparts (like the [STM32N657X0H3Q](#)) because of a similar feature-list at a lower price and significantly lower pin count. Although this component is designed for more complex- and intense-tasks, I will be able to evaluate which features will be utilised and which won't be, which will aid in selecting the optimal processor for the prototype. It serves as a solid middle ground between the more extensive and cheaper counterparts.

Segment	Code	Meaning / Explanation
Manufacturer Family	STM32	STMicroelectronics 32-bit microcontroller family
Series	N6	STM32N6 series - based on Arm Cortex-M55 core with Neural ART NPU
Sub-Family / Feature Set	57	High-end variant in the STM32N6 family (e.g., more memory, interfaces, and image processing features)
Memory / Package Variant	I0	Midrange flash/RAM size and medium-pin-count package
Temperature Grade	H	High-temperature / industrial range (typically -40°C to +85°C or higher)
Variant Code	3	Specific feature or silicon variant (usually minor revisions or option sets)
Qualification Grade	Q	Automotive or industrial qualification

Table 6: Breakdown of STM32N657I0H3Q Part Number

Despite the current choice, it is highly recommended that if the scope of the project were to expand in the future, further research into ASICs should be completed. Compared to modern SoCs or FPGAs with image processing capabilities, some ASICs provide a lower-cost entry point with greater efficiency in real-time image processing.

Since the selected processor has sufficient internal memory to buffer a full-resolution image, external memory is not required to support real-time image acquisition and processing. If future revisions of the project (i.e., the prototype) require more memory to buffer the image, the selected processor is compatible with external uSDRAM like the [IS45S16320D-7TLA2](#).

2.2.5**SD Card Reader**

Since I need a short and effective method of checking the files in a remote environment without tethered access (USB connection), I opted for saving the files on an uSD card. Although this does not provide real-time imagery, the slightly smaller scope allows us to better focus on the functionality of the camera rather than how the fully processed image is moved around.

Instead of PCB-mounting an uSD card directly, I chose the [254 Adafruit](#) breakout board because it can be easily connected via standard PCB headers. Although it appears this step has been short-cut, the prototype will not use an uSD card reader so it suffices as an effective solution without having to deal with the technical aspects. As the prototype is expected to offload images to another device, this breakout-based approach works well for viewing the file in its intermediate stage (leaving the processor and being sent to FFS).

2.2.6**External Clock**

The [ECS-TXO-3225MV-240-TR](#) was selected primarily because it provides a highly stable and precise 24 MHz clock signal, which perfectly suits our system's timing requirements. This temperature-compensated crystal oscillator (TCXO) ensures minimal frequency drift across varying thermal conditions, a critical factor for reliable operation in aerospace environments. Additionally, the Avionics Department Representative recommended a TXCO, reinforcing its suitability for our application. Since the design only requires a single clock output rather than a differential pair, this oscillator meets our needs efficiently without unnecessary complexity. Additionally, the MIRA220 image sensor also requires a 24MHz reference clock, so only a single external clock is required for the breakout. Although the clock is connected to two components, its load capacitance of 15pF should prevent any overloading symptoms.

3 C2Sv0 (Shelved)

3.1 Remarks

Throughout the development phase of the breakout board, the project faced several scope revisions. *These remarks cover comments for both subsection 3.2 and subsection 3.3.*

The first revision focused on the structure of the breakout board. Initially, the breakout board was fully encapsulated, meaning that all the components (processor, sensor, uSD, etc.) were all housed on a single PCB. I found that it made more sense to split the board into the sensor and processor, given the prototype will use the same format. Not only would this prompt design considerations for the prototype, but if the breakout boards were successful, they could be used to debug the prototype i.e., the functioning-processor breakout could be used to program the prototype sensor, and vice-versa. This revision did not introduce any major roadblocks or inconveniences to the development since all draft PCBs at the time either had the sensor placed in an inconvenient location, or the sensor was essentially "disconnected" from the rest of the board (see Figure 2)

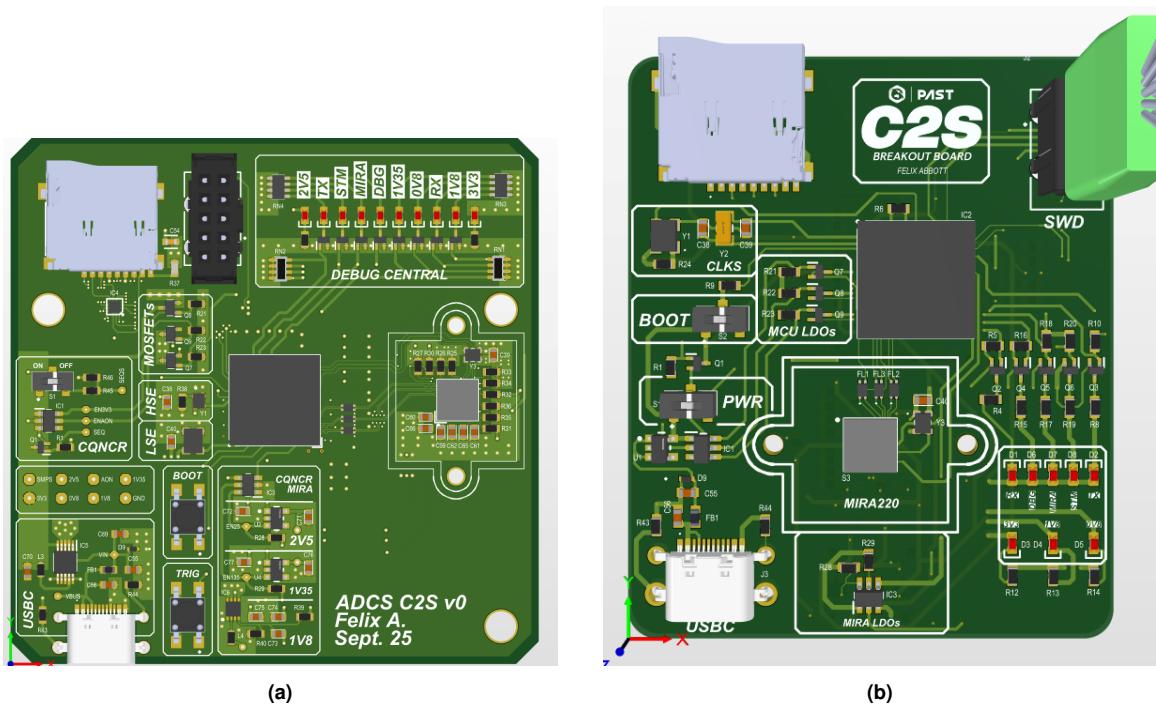


Figure 2: Draft PCBs of the C2S Breakout Board.

The second revision was centered around the selection of the image sensor. In section 2, I selected the MIRA220 as the image sensor for both the breakout and prototype PCB, however, throughout development, I found that the component's characteristics were too fine to reasonably justify (within the project budget). The BGA ball diameter of the MIRA220 was 0.2mm ($200\mu\text{m}$), much smaller than JLCPCB's manufacturing capabilities. Moreover, the trace widths required to route the BGA was much too thin. These tight tolerances drastically increased the manufacturing cost of the board from what I expected to be \$50AUD to \$300AUD (with some going beyond \$500AUD). The price to manufacture the boards with this sensor was far too expensive, so, I pivoted to the next-best sensor with larger tolerances. Although the tolerances of the new sensor was tight, the boards were now manufacturable by JLCPCB at a reasonable cost.

The third revision aimed at reducing the failure rate of the breakout board. In addition to splitting the PCB into the sensor and processor boards, I decided to add the STM32N6 Nucleo Board to the project scope as well. This would allow us to interface with the sensor directly even if the processor board didn't work. Now that the minimum success criteria of the breakout board no longer required the entire processor board to work, the risk of not meeting the minimum criteria drastically reduced.

The fourth revision involved reworking the naming scheme. As mentioned in [section 2](#), the first PCB was called the 'breakout' or 'breakout board'. However, as the number of PCBs in the project grew, I required a new method of referencing each board. In addition to the 'breakout' term, I had begun to call this version/iteration C2Sv0 to highlight on the project's initial development as a breakout board, whereas C2Sv1 would refer to the first functioning version of the project, the prototype. The terms v0 and 'breakout' have since been used interchangeably. Moreover, each PCB has been named according to its purpose:

- C2Sv0-P: (P)rocessor.
- C2Sv0-S: (S)ensor.
- C2Sv0-A: (A)daptor.

The fifth revision was dedicated to saving costs on the manufacturing process. After factoring in all of the component and manufacturing costs, I were facing approximately \$813AUD and \$1042AUD for one and two PCB sets (with one set of components for redundancy) respectively. This was far too expensive for just the breakout board, so modifications to v0 were required. To cut down on the cost, I decided to re-combine the processor and sensor board onto one PCB (with break points) to avoid the double-fee for tight tolerances and only be charged for tight tolerances on one PCB instead. Since JLCPCB does check the boards for this type of 'manipulation', I added some fake components and silk-screen onto the tabs (section to be snapped) to make it seem like it was one coherent PCB instead of two that were clearly going to be split after production.

On September 23rd, 2025, **this version was shelved** because of its high price point. After discussing the project with Trsitan Davies and Robert Howie, I concluded that the cost of the PCB (\$750AUD) was far too great to justify for a single revision with one person working on it, given multiple revisions may be required. The PCB was complete however, and the gerber files along with the components can be found on the C2S SharePoint folder in-case the version is picked up again in the future.

3.2 Schematic Design & Component Selection

3.2.1 Overview

This phase involves designing the camera circuit schematic using Altium Designer, laying the groundwork for subsequent PCB development and component selection.

The schematic is divided into eight sub-sheets:

1. Processor
2. Image Sensor
3. Clocks
4. uSD Card
5. LPS
6. SWD Connector
7. USB-C Connector
8. Power Sequencer

3.2.2 Processor

3.2.3 Image Sensor

Like the STM32N6, I followed the [AR0147 datasheet](#) on how to properly create the image sensor schematic. Although it wasn't specified in the datasheet, I included common mode chokes on the differential lines like the MIRA220 schematic.

For the connector to the processor, [this guide](#) was followed on how to properly set the pinout. I set every third connection to ground to ensure a reliable return path for all signals.

The most important design consideration with this schematic was the requirement of digital and analog ground (DGND/AGND). To understand how to work with different signal planes, the following resources from TI were used: [1](#) and [2](#). To connect both grounds, I used a simple star connection

with a NetTie. I also set DGND as the equivalent ground on the processor PCB (GND=DGND on the connector).

I selected the [ECS-TXO-3225MV-240-TR](#) as the external PLL oscillator for the image sensor. This component can be operated on 3V3, however, I chose 2V8 so the oscillator will only start when the sensor's specific power domains are provided and doesn't 'burn' power when the sensor is not in use. Page 22 of the sensor datasheet specifies that the tEXTCLK signal must have a rise/fall of between 0.2ns and 12.5ns (12.5ns extrapolated from 0.3*tEXTCLK). These requirements are met by the selected oscillator since the oscillator datasheet says it has a rise/fall time of 5 ± 0.5 ns. Since the generated signal is a clipped sine wave, I can estimate the duty cycle is 50% since it's symmetric.

3.2.4

Clock

The C2S Breakout Board required three individual clocks for operation:

- High-speed 24MHz (HSE) oscillator for the STM32.
- Low-speed 32.768KHz (LSE) oscillator for the STM32.
- High speed 24MHz external oscillator for the AR0147 image sensor (discussed in [subsubsection 3.2.3](#), not here).

For the HSE configuration, I referred to the "[Oscillator design guide](#)" for STM32 clocks and obtained the following system requirements: For the N6 series (Table 6 of oscillator guide), I require a frequency range of 16-48MHz which is further verified in the datasheet with a typical frequency of 40MHz (Table 49 of datasheet). Additionally, they must operate at 3V3 or 1V8 supply voltage (since those are the only two supplies available pre-boot).

I selected the [ECS-TXO-2016-33-400-TR](#) TCXO oscillator to provide the HSE clock signal. I opted for a 40MHz oscillator opposed to the 24MHz selected in Phase 1 since the guides recommended to use 40MHz. I were not able to calculate g_{mcrit} since the ESR is not specified on the datasheet. However, since other ECS oscillators are recommended like the [ECS-327-6-16-TR3](#), I have assumed it is compatible. For the LSE oscillator, I selected the [ECS-327TXO-33-TR](#). I opted for this component over others since it was the only available ECS TCXO oscillator in-stock at that frequency.

Since all three oscillators are TCXO, they already include internal circuitry (like the capacitor configuration) and simply need to be connected to the input and output pins of the STM32.

Each oscillator should continuously produce a clock signal from boot that doesn't interrupt start-up or shut-down sequences.

3.2.5

uSD Card

I used the getting started guide to construct the uSD Card schematic. Like the guide, I used the [NXS0506GUX](#) since the recommended component uses a BGA package that is too fine.

For faster readout speeds, I have connected the component in 4-bit mode, opposed to the simpler, but slower, 1-bit mode (connecting only DAT0).

Additionally, I added a decoupling capacitor to VCC using same value as the decoupling capacitors used on the image sensor.

3.2.6

LPS

The Local Power Supply (LPS) schematic was mostly copied from the MIRAA220 reference schematic. The LDO components were updated to support active components, and the 1V8 power line was modified to be generated when then 3V3 line was present, since the MCU requires 1V8 to boot. Moreover, Test points were added to the outputs of the LDOs as well as on the enable lines. Since each LDO uses the same package (SOT-25-5 / SMV), the schematic uses the 2V8 LDO as a placeholder. The selected LDOs were:

- [TCR2EF28,LM\(CT\)](#)

- [TCR2EF135,LM\(CT\)](#)

I were able to meet the following requirements from the datasheet:

- 3V3 input on V_{IN} with a $0.1\mu F$ decoupling capacitor (pin 1)
- $1\mu F$ decoupling capacitor on the output (pin 5).

As mentioned in [subsubsection 3.2.8](#), the system will require 2500mW in the worst case scenario. This is mostly attributed to V_{DDCORE} , the most power-intensive domain (2000mW). The internal SMPS is supplied by the 1V8 power domain, which corresponds to a current requirement of 1.11A for just the STM32's SMPS (rounded to 1.25A to account for other components). Fortunately, the other power domains operate below the chosen LDO's current limit of 200mA. However, I still need to accomodate for V_{DDCORE} . As a result, I selected the [TPS62061DSGR](#) as the buck converter to supply the 1V8 power domain. It can operate up to 1.6A output current at the required 1V8 domain.

Please see [subsubsection 3.2.2](#) for more detail about the resistor selection of the LEDs.

3.2.7 SWD Connector

The SWD connector followed the [Phil's Lab guide](#) on STM32 PCB design, as well as referencing the ADCS Integration Board for the component.

3.2.8 USB-C Connector

The USB-C connector was implemented as a source of power for the breakout board. Instead of needing to rely on FPC connectors or headers to jumper +3V3 power lines, I can simply use the power pins on the USBC connector. This enabled us to make use of other power supplies like powerbanks, rather than relying on stray wires or peripheral PCBs.

The schematic was referenced from the ADCS Integration Test Board. Unused pins were either left as NCs or pulled to ground. Test points were added on the main power levels (GND, VBUS, VIN, 3V3) to confirm potential readings during testing.

The ferrite bead and capacitors form a pi-filter (C-L-C) network. The input capacitor shunts high-frequency noise from the connector before it can reach the PCB. The ferrite bead acts as a suppressor to high-frequency noise (low-pass filter) by absorbing the signal and dissipating it as heat. At DC and low frequencies, it acts as a short-circuit, allowing typical signals to pass. The output capacitor provides local decoupling for the PCB and helps smooth out any remaining noise. Using two different valued capacitors help increase its performance over a wider range of frequencies. With an inductor value of $L = 190\text{nH}$, I chose capacitor values of $C_1 = 100\text{nF}$ and $C_2 = 10\mu F$ ([subsection A.1](#)) to create a cut-off frequency $C_{eq} = 1.2\text{MHz}$. I chose these values over others since they were seperated by two magnitudes and these values were already being ordered, which saves on unique component quantity, and the maximum value I can select for a capacitor on a USB line is $10\mu F$.

To introduce surge protection for electrostatic discharge, the schottky diode helps clamp voltage spikes to ground to protect other components in the PCB from being damaged. Although voltage spikes may not immediately break components, over time, they can wear down.

Lastly, I needed an effective and reliable method to maintain a +3V3 potential. If V_{DDCORE} requires 1500mW at its peak, I can estimate that the internal SMPS will require 2000mW to create the power domain (at 75% efficiency - Figure 18 of the STM datasheet illustrates the efficiency is 85%, however, I have selected 75% for padded margins). From the [N6 power supply guide](#), our maximum power draw is approximately 2500mW (worst-case scenario). At a 3V3 voltage supply, this corresponds to 750mA of current (no losses). Unfortunately, the LDOs used in [subsubsection 3.2.6](#) can only output a maximum of 200mA of current. Although I could chain multiple in parallel to reach the required current, I found it best to continue with a more efficient buck converter. I selected the [TPS62056DGS](#) as the buck converter to supply the main power domain to the PCB. It can output 800mA at 3V3 with an input of 2.7-10V at a very low operating supply current (12uA).

3.2.9

Power Sequencer

Unlike other STM32 MCUs like the H7, the N6 processors require a specific startup pattern to properly boot. As found on [pg. 27](#) of the datasheet and [pg. 9-10](#) of the getting started handbook, the startup sequence is:

1. V_{DD} and $V_{DDA18ON}$
2. V_{DDSMPS} via PWR_ON

From the power sequencer schematic, the component sequentially enables the FLAG3, FLAG2, FLAG1 pins. These enable pins are connected to the base of MOSFETs (see MCU schematic), which allow the sequencer to turn the power lines on and off in the required pattern. FLAG1 has been left as an NC since the final enable trigger is instead from PWR_ON. The PWR_ON pin connects to the base of the MOSFET on the VDDSMPS line to drive high SMPS when the MCU requires it.

To prevent the STM32 from prematurely shutting down from a hard-reset (switching the power switch, S1, off), a soft-reset function has been added. This means that the power sequencer, IC1, is controlled by two components, the switch and a MOSFET. Once the PCB is first turned on by closing the switch, the power regulator circuit properly boots the STM32. Once the STM is operational, it will drive the base of the MOSFET high via MCU_SEQ. This means that even if the power switch is opened, the sequencer remains enabled because the STM is holding the MOSFET circuit. To ensure that the STM actually shuts down, the MCU_SEQ_S (S meaning STATE) carries the state of the switch to the STM. This effectively tells the STM "it's time to shut down", allowing it to finalise everything before disabling the MCU_SEQ pin and shutting down.

Since there did not appear to be any voltage drop across the switch stated in the datasheet, a voltage divider was used to read the state to protect the GPIO from any surges. A large value R_2 was selected since the net is connected to the main input power and little current is required to read the voltage.

3.3 PCB Layout and Review

3.3.1 Selecting Manufacturers

I selected JLCPCB as the PCB manufacturer since they are capable of producing the PCBs within their capabilities, and I (PAST) have always used JLC to produce our PCBs.

Because of the BGA package size, I concluded that it was not feasible to solder those two components by hand. To place the BGAs, I consulted with Peter from AT&MWA to discuss their capabilities and whether they can place the two components or not. I found that:

- They are capable of placing BGAs of up to 0.2mm diameter and 0.5mm pitch.
- I should aim for a solder mask dam of 4mil (0.102mm) so the board can be fabricated with solder mask around each pad. I can use 2mil if required.
- I should produce a few spare boards in case I want to stress one or two out during testing.
- The two ICs should be placed in around 30-40 minutes a board. This would costs around \$35-\$47AUD+GST per board.
- They will produce the stencil themselves, and they will recommend the best way to panelise the boards to suit their machines.
- I am able to place the rest of the components first with the BGAs being last.

3.3.2 Current Carrying Capacity

Using the [Saturn PCB Toolkit](#), I can estimate the current carrying capacity of the BGA traces with the following parameters:

Parameter	Value	Unit
Conductor Width	4	mil
Conductor Length	50	mm
PCB Thickness	62	mil
Plane Present	Yes	-
Distance to Plane	10	mil
Conductor Current	1.13	A

Table 7: Current carrying capacity estimation of BGA traces in C2S Breakout Board.

As per [Table 7](#), the maximum current on a 4mil trace is 1.1A, which is less than the typical current requirement. The current carrying capacity is not a concern for the STM32N6 since its power traces are 0.2mm, which can carry 1.8A, equal to the maximum 1.8A drawn by the MCU.

3.3.3 BGA Fanout

I encountered clearance issues with the AR0147 BGA fanout. Unlike most BGAs which have a ball radius of 0.5-1.0mm with large pitches, the sensor uses a small package profile. This package made it difficult to properly route the component. [fanout methods](#) like Via-in-Pad were not possible as such a small scale, simply because the PCB capabilities didn't permit small enough vias. For most PCB manufacturers, the minimum via diameter size is 0.2mm, with a 0.15mm annular ring (0.35mm total diameter). Since the via was larger than the BGA pad, via-in-pad was not a feasible option. I then opted for the dogbone fanout method which admit to the manufacturer's trace width and clearance of 3.5mil (see [Figure 15a](#)). Unlike the sensor, the STM32N6 has a more coarse BGA package which allowed us to fanout without issues. In this case, I opted for the dog bone fanout since it allowed us to place our ground vias closer to the power vias. It also provided more clearance routing diagonally, allowing us to use thicker traces for a better current carrying capacity.

3.3.4 CSI-2 Routing

I used [TI's high-speed guidelines](#) for routing the differential signals. From Table A-7 of the guide (pg. 19), I found that the differential pair impedance is $100 \pm 15\Omega$, with lane skew of 40ps, with 2 total vias per pair. Using Altium's impedance calculator and [guide](#), I found the required separation

distance between the 3.5mil differential traces to be 3.542mil, or approximately the trace width. According to the STM32N6 getting started guide, I require a length match of $\pm 5\text{mil}$, and the MIRA220 datasheet (although not used) specifies a delay match of 40ps (which is greater than 5mil on the microstrip). I length matched the data pairs to achieve this low delay ($\pm 2\text{mil}$). Additionally, the getting started guide specifies a length match of $\pm 100\text{textrm{tt}mil}$ between the data and clock lines, which was achieved.

It was not possible to keep all the differential pairs on the top layer because of how the MIRA and STM pins aligned, one pair always ended up using vias. To ensure that each signal was grounded, I made sure to place stitching vias around the layer changes.

3.3.5

Debug LEDs

The TX and RX pins are **not representative of UART**. Since there was not enough space to add "READ" and "WRITE", I opted for the shorthand notation used in UART. In this case, RX means the MCU is receiving / reading from the image sensor, and TX means the MCU is transmitting / writing to the uSD card.

The power LEDs (3V3, 2V8, 1V8, 1V2, 0V8) show whether each line is active or not.

The "AR0" and "STM" LEDs show whether each component is active or not. They are each connected to a GPIO pin on the STM32. They will turn on when the STM32 is operating and can utilise its GPIO pins, and whether it can talk to the image sensor.

4 C2Sv0.1

4.1 Remarks

As mentioned in [section 3](#), the version was shelved because of the high price point. Unlike version 0 which used MIPI CSI-2, v0.1 focuses on the DCMI (DSI) interface (for more information about DCMI, [read here](#)). DCMI is an older and slower interface for select STM32 chips, so, it's cheaper than a fast and modern interface like CSI-2. Although it is slower, DCMI can still provide 720p at 30fps, which greatly exceeds the project requirements of buffering a single frame aperiodically.

Unlike CSI-2, processors that have DCMI typically have non-BGA packages, which means I can avoid using the ENIG surface finish across the entire processor PCB, and use much larger via sizes. Using LQFP or QFP packages for the processor also removes the AT&MWA placement fee entirely for the processor, meaning we can test and verify the processor without additional costs. Although the image sensor still requires a BGA package, sensors that use DCMI often tend to have larger packages because they are older, which means I can use larger vias to save more money.

Moreover, I had selected components with the space requirements in mind (vacuum compatible, large operating temperature range, etc.) so that circuit known to work can be recycled for the prototype. Ditching from BGA packages was the first step, removing the electrolytic capacitors was the second. Ceramic capacitors were chosen for vacuum resilience since they avoid any risk of a severely lowered life expectancy and potential explosion in altitude conditions [7].

Aside from the above, the board's features extend to:

- Two miscellaneous buttons for a custom purpose, like a timed photo or timelapse. This allows me to test even more functionality without needing to flash new code to the processor, reducing the programming timeframe (since I can work on one larger code set instead of several) and potentially shortening the testing timeline since I don't need to re-upload code.
- More LEDs to make the debugging and programming process easier, as well as some multi-colour LEDs to increase the density of information within each LED (e.g., a single +3V3 LED can be off, flash red for low battery or be green for stable health - relays more information than a monochrome LED). This will make the debugging process far easier since the processor now has far more avenues of communicating errors than in the previous version.
- A buzzer for audio indication (e.g., start-up, taking a photo, critical errors, etc.). This builds on the core reasoning behind having more LEDs, so that the processor has more signals for debugging.
- Utilising QSPI for external memory since all eligible processors do not meet the memory requirement. Although all octal memory options within requirements come in BGA packages, QSPI still allows us to quickly access external memory at a very low pin count - this makes it far easier to route since there are much less traces to length match.
- New data lines for testing transmission to an external computer: UART, I2C, SPI and HSPI. From this, we can easily test and confirm which protocol will be the best for the prototype (transferring the image to the flight computer). This will make it easier to justify which transmission protocol we select for the prototype since we will have tests to support the choice, and it will reduce the software timeline for the prototype since the necessary libraries and functions would have already been constructed.
- A 6 layer PCB to retain signal integrity and minimise compact routing issues. As of early October 2025, JLCPCB has a special offer for their 6-layer PCBs that costs roughly \$15AUD more than the 4-layer option. Having access to two more signal/data planes on the PCB to minimise issues risen from compact routing issues which will improve signal integrity. Although it is slightly more expensive than the 4-layer option, the benefits of the extra layers outweigh those of the 4-layer PCB.

Overall, this version has an arbitrary hard-limit of \$250AUD (inclusive of all costs: components, manufacturing, shipping, etc) for a single PCB set (one sensor and processor PCB fully assembled with all components). Exceeding this limit would make the budget on future revisions much tighter, especially if this version doesn't fully work and needs to be revised. The breakdown of the budget is as follows:

Item	Cost (\$AUD)
PCBs (shipping inc.)	88.27
AT&MWA BGA Fee	50
2x AR0141	?
2x STM32U5	?
Other Componenets	?
Total	\$240AUD

Table 8: Cost breakdown of C2Sv0.1.

From v0, the most notable lessons learnt were:

- Keeping the image sensor and processor on separate PCBs. Although this version can no longer use a nucleo board like the N6 for debugging, working v0.1 PCBs can be used to debug prototype (v1) PCBs.
- Spending more time planning out where to place the components before routing. With v0, I only landed on the desired PCB layout after three previous attempts at making a PCB because I was very eager to finish the PCB and very quickly got stuck in a tangled mess of an unplanned layout and wasted many hours on un-used PCBs. I found it best just to copy the PCB layout into Adobe Photoshop or MS Paint and draw out the lines to see if they'd actually work before doing so (Photoshop is unconventional, but was effective). Although I did end up changing my original layout for v0.1, I only required minor modifications, so the planning was an excellent step. This also helps prioritise the signal integrity of more important traces first, like camera and memory lines.
- Finding and using reference schematics was a time-saver. Although it may sometimes feel like cheating when following another design, it can save a lot of mistakes. If I hadn't found the MIRAS20 reference schematic before starting v0, I would have struggled a lot with developing the schematic, and even more when trying to debug my errors. Especially for a new topic like camera design, having reference schematics is a must - it prevents pin mapping errors. Despite this, I still try and develop the schematic myself, and use the reference as a checklist/guide to know whether I've made any mistakes or not.
- Utilising free resources and documents. There are plenty of complete-documents published by ST, TI, etc. that go in-depth with some critical pieces of knowledge. Before starting v0, I had never known about DGND and AGND, but I read through the TI document and learnt more about what they are, how they differ, and what to do with them [12]. ST has released lots of resources as well, virtually every topic has been covered by them (especially in camera design).

The schematic and PCB were **reviewed and conditionally approved by Tristan Davies on 19/11/2025** - adjust mounting hole pattern to fit vibration table.

4.2

Schematic Design & Component Selection

4.2.1

Overview

This phase involves designing the camera circuit schematic using Altium Designer, laying the groundwork for subsequent PCB development and component selection.

The schematic is divided into eight sub-sheets.

1. MCU

This contains all relevant circuits that interface with the processor and aren't integrated circuits (debug LEDs, buttons, etc.). Additionally, it includes the two external clocks for the U5G (although they are integrated circuits, they only have a single output going directly to the processor so it was easier to include them on the sub-sheet).

2. uSD Card

This hosts the uSD connector and the relevant dependencies.

3. LPS

The Local Power Supplies (LPS) sub-sheet holds all the voltage supplies like LDOs and buck converters. This doesn't need to interface with any other sheet since it only requires global power coming in and global power coming out.

4. TAG Connector

This is simply for the TAG connector. Keeping it on its own subsheet was easier since it cleaned up the processor subsheet.

5. USB-C Connector

The USB-C sub-sheet contains everything related to the main power supply (+3V3). It hosts the USB-C connector itself, as well as protection circuits, power switch and the 3V3 LDO.

6. Memory

This is for the external memory. Originally, SDRAM was used for the external memory which required 20+ lines to interface with, so it was awarded its own subsheet. Although the memory is no longer SDRAM, having its own sub-sheet cleans up the processor sheet.

7. Image Sensor

Although the image sensor schematic is stored within a separate project so it can be on its own PCB, it can still be considered a sub-sheet for simplicity. This contains all components that will be housed on the sensor PCB (sensor, clock, capacitors, connector, etc.).

To meet cost requirements, the PCB must admit to low-precision standards (minimum trace/clearance of 4mil and 0.3mm/0.4mm via hole/diameter). To admit this, I have selected non-BGA components where available, and for strictly BGA packages (like the image sensor), I have considered sensors with a pitch large enough to fit a 4mil trace between the pads.

All external components (those not found within the PAST component server or Manufacturer Search) are sourced from the local schematic and footprint libraries which can be found within the Altium project file.

4.2.2

Image Sensor

The image sensor is a very difficult category to save money on. Most will cost upwards of \$50AUD for decent/quality sensors, dipping below that line whilst seeking for the same quality has its drawbacks. The most notable drawback is the availability of the sensors, most that fit the requirements are either obsolete, or in its last-stock but requires bulk purchases. Some prime examples of this are the AR0544 and AR0830. The second drawback is the package. Image sensors are designed for compact systems, so finding sensors that have a BGA package within manufacturing capabilities is a challenge of its own. Sensors like the MIRA016 and AR0144 were both within our price range, but their BGA ball diameter and pitch no longer made it feasible to consider them. This left us with the [AR0141CS2M00SUEA0-DPBR](#), the M meaning mono or monochrome. The AR0141 was only option found within the price range and package requirements with an available datasheet. Although there were cheaper, colour image sensors like the [MT9M114EBLSTCZ-CR](#), they all had those two drawbacks. For the MT9, its package was too fine. Despite the monochrome filter, I concluded that this would not be an issue since the only difference between is the number of channels. Both colour and monochrome images undergo the exact same process (except for the Bayer decoding), so for a breakout board to test functionality, it was not an issue.

Since v0 and v0.1 share similar sensors, it was easy to develop the schematic. The only notable differences between them was the parallel interface. Since both sensors are from the same family, they both required a similar number of power, ground and data pins (just with different voltages). I selected a FPC connector (the same 36-pin connector used on FISHv1) because the connector type is compatible with CSI-2, so it should also be compatible with DCMI (in terms of FPC impedance and signal integrity). Furthermore, to minimise the complexity of the board, I only kept the bare essentials (and debug LEDs) on the schematic to minimise the size of the board, which lowers production costs (price scales with ENIG finish).

For the debug LEDs, I have chosen to include three to verify that the image sensor is receiving the signals: (1) reset, (2) shutter, and (3) stand-by. Each use the same n-type MOSFETs used on the processor schematic.

To make the module vacuum-compatible, I replaced the electrolytic decoupling capacitors that were recommended with ceramic capacitors so I wouldn't encounter any issues during vacuum testing.

After consulting with the community (see [here](#)), I decided not to split the copper ground into AGND

and DGND. Instead, they are connected to GND/DGND.

Use pg. 59 of sensor datasheet for startup sequence.

I chose to leave the flash pin as a NC since the camera would never need to operate with a flash, so its function is redundant. Moreover, the SLVS pins (serial four-lane HiSPi interface) was left as NC since that interface was not being used.

Unlike traditional DCMI naming conventions, the AR0141 uses its own pin names:

AR0141 Pin	DCMI Name
PIXCLK	DCMI_PIXLCK
LINE_VALID	DCMI_HSYNC
FRAME_VALID	DCMI_VSYNC

Table 9: AR0141 DCMI pin naming convention

The camera interface has a configurable parallel data interface from 8 to 14 data lines, together with a pixel-clock line DCMI_PIXCLK ... DCMI_PIXCLK and AHB clocks must respect the minimum ratio AHB/PIXCLK of 2.5.

In other words: the AHB bus clock must be at least 2.5 times faster than the PIXCLK input to the DCMI [24].

Lastly, the decoupling capacitors for the power rails use the 0402 package instead of 0603 so they can fit inside the M12 mount.

4.2.3 MCU

There were four STM32 families to pick from that met the requirements of the system; (1) F4; (2) F7; (3) H7; (4) U5G.

	STM32F4	STM32F7	STM32H7	STM32U5G
Core	Cortex-M4, 168 MHz	Cortex-M7, 216 MHz	Cortex-M7, up to 480 MHz	Cortex-M33, up to 160 MHz
RAM	2 MB / 256 KB	512 KB / 128 KB	2 MB / 1 MB	2 MB Flash / 786 KB SRAM
FMC	Yes	Yes	Yes	Yes
DCMI	Yes	Yes	Yes, higher speed + parallel capture	Yes (limited frame rate)
DMA	Moderate	High	Very High (AXI + AHB buses)	High (AHB + low-power DMA)
Max Pixel Rate	24 MP/s	48 MP/s	80 MP/s	20-24 MP/s
FPU / DSP	Single-precision FPU	Single-precision FPU	Single-precision FPU + Chrom-accel (optional)	Single-precision FPU + DSP instructions
Use Case	Simple camera + SDRAM buffering	Medium-res camera + basic image processing	High-res camera + advanced image processing and graphics	Low-power image capture + ML processing
Power Consumption	100-120 mA core, +20-30 mA with DCMI/FMC	120-160 mA core, +30-50 mA with DCMI/FMC	200-300 mA core, +50-80 mA with DCMI/FMC	45-80 mA typical (low-power modes available)
Price	Moderate	Moderate / higher	Higher-end	Moderate / low-power premium

Table 10: Comparison of STM32F4, F7, H7, and U5G (144-pin) for Image Processing Applications

Although the H7 offers more features and higher processing efficiency, its power consumption and cost make it excessive for the current application. The F4 is slightly older, but the F7 provides an excellent middle ground, delivering many of the H7's advantages while remaining comparable to the F4 in price. Despite this, the U5 posed as the most promising candidate. As an ultra-low power processor, the U5 still delivers the same (or even more) high-performance features as the F7 at a low price. Naturally this comes with its drawbacks like the lower pixel and frame rate, but even with these limitations, the U5 still sits within the requirements of the processor. I selected the 144-pin U5 since the 100-pin option did not have enough GPIO pins.

I followed [20] and [25] to connect all the pins (Table 12). Page 41 of [25] included a very handy reference guide which made the pin mapping process swifter.

I was able to recycle the same LED, HSE and LSE configuration from v0, as well as include an audio buzzer for further debugging purposes.

Just to slightly increase the customisation of the board, I included two extra push-buttons for a miscellaneous purpose like starting a timelapse, or a timed-photo. The STM's software would need to be re-uploaded everytime the button's purpose is changed, but I doubt this would need to occur often throughout testing (maybe for outreach purposes).

We can estimate the power consumption of the STM using STM32CubeMX software integrates a tool called Power Consumption Calculator (PCC) [19].

Three standard transmission protocols were selected for testing. These standard protocols were selected over others because they are widely supported, easy to implement and are reliable.

- **UART**

If the UART protocol is a good candidate for the prototype, it will require little shielding because of its low speed, and between transmissions it can double as a serial port for communicating to the flight computer.

- **SPI**

A de facto standard for synchronous serial communication. SPI is significantly faster than UART, so the elapsed transmission time will be much shorter. It is only being used to transmit data, so in the STM32CubeIDE, the SPI mode has been set to 'Transmit Only Master'

- **I2C**

Although it is slower than SPI, it uses fewer wires, which could be beneficial in a tight-PCB.

4.2.4 Memory

Unlike the STM32N6, the U5 does not have a large enough internal memory to buffer an image. I selected 256Mb of memory compared to a value closer to the actual RGB image size (32Mb) just so I did not have to worry about running out of memory. The increased memory size only result in a \$3AUD increase in price, so the price change was justified by the increased memory threshold. Throughout testing, I expect to measure the consumed memory so the prototype uses only as much memory as it needs.

I opted for QSPI over alternative options like SDRAM, and OSPI because of pin count and package issues:

- Initially, I proceeded with SDRAM, but this required nearly 30 lines all length matched, which quickly became far too difficult to achieve in a somewhat compact board.
- OSPI (OctoSPI) was the next best option, however, all the components I had come across all used BGA packages, so they were not further considered.

Although QSPI isn't the fastest memory option, the selected component has a maximum clock speed of 133 MHz which should be fast enough (given the DCMI_PIXCLK is configured slower than the QSPI speed [21]).

Of the available options, the AT25SF2561C-MWUB-T was the cheapest.

The QSPI was configured according to [23] (section 3.2.3). Section 4.3 was helpful for developing the schematic [23], as well as using [10] as a reference. Although the STM32U5G doesn't have an explicit QSPI or xSPI interface, it seems possible to configure OCTOSPI1 to work with the QSPI protocol [6] [22]. It should be noted that the QSPI SS and DQ lines should all be length matched to less than $\pm 50\text{ps}$, and all lines should be length matched within $\pm 250\text{ps}$. [15] [2].

4.2.5

USB-C

The USB connector acts as a sink of power flow [11]. The schematic was mostly copied over from v0. I selected the [USB4105-GF-A-060](#) component over others simply because that was the connector used on FISH and ADCS Integration board. I was able to remove the power sequencer (since the U5 does not require a specific startup sequence) and replace the buck converter with an LDO (because of the lower maximum power consumption). In addition, I removed the differential data pair (D+, D-) because Robert Howie suggested the COM port functionality was not very useful compared to using the ST-Link.

As outlined in pg. 54 of [11], the sink requires $5.1\text{k}\Omega$ R_d resistors on the two CC lines. Since the total capacitance of VBUS does not exceed $10\mu\text{F}$, I do not need a power path or load switch. Moreover, I will not be exceeding 15W of power consumption, so, I do not require a USB PD controller.

Using the logic outlined in [subsection A.3](#), I can estimate the power consumed by the circuit by placing a low-value shunt resistor before the LDO and measuring the voltage drop across them from V1 to V2 with a current sense amplifier. Any voltage losses created by the resistor will be resolved by the LDO since it will regulate the voltage to +3V3. This will provide an accurate power consumption value; it will infer what the maximum current carrying capacitor of traces are for the prototype; and it will provide the Systems Engineering Lead with vital power requirements for the CubeSat. It must be noted that since the shunt resistor is in series with the high-side of the circuit, it must meet current carrying capabilities - this was an issue that flew under the radar until late in the PCB development cycle. As per [14], for a maximum offset voltage of $15\mu\text{V}$, we require a minimum impedance of $150\mu\Omega$. of $2\text{m}\Omega$

The component appears to have no absolute maximum characteristics listed. Since it's a connector and not an integrated circuit, I have assumed that it should not be a concern for thermal and vacuum testing. I will need to consider the maximum ratings of the connected battery/powerbank for stress testing however.

The bi-directional TVS diode acts as surge protection by clamping the voltage in both directions (copied over from ADCS Integration Board). For more information about the pi-filter network, please see [subsubsection 3.2.8](#).

4.2.6

uSD

The uSD schematic was copied from v0. Unfortunately, the filter used in the original schematic sold out across most stores three days before v0.1 started, so, the component was removed from this schematic. Since the filter did act as EMI protection (as suggested in the N6 getting started guide to reduce noise and protect signal integrity) there may be slightly higher EMI which can be mitigated by proper ground referencing. I will also make sure not to prematurely remove the uSD whilst the circuit is active just to mitigate any potential risks.

The component appears to have no absolute maximum characteristics listed. Since it's a connector and not an integrated circuit, I have assumed that it should not be a concern for thermal and vacuum testing. I will need to consider the maximum ratings for the actual uSD card used however.

I followed [26] and [4] for developing the uSD schematic.

4.2.7 LPS

According to the datasheets of the U5 and AR0141, the only power domains required are: 3V3, 2V8 and 1V8. The U5 only requires 3V3 and 1V8 to operate, so 2V8 was only required for the image sensor. According to Table 27 of the AR0141 datasheet (page 54), the maximum operating current consumption in parallel output is 160mA. Because of this, I was able to re-use the same LDO package from v0 since their maximum current output is 200mA. As per the datasheet, I included decoupling capacitors on VCC and VOUT.

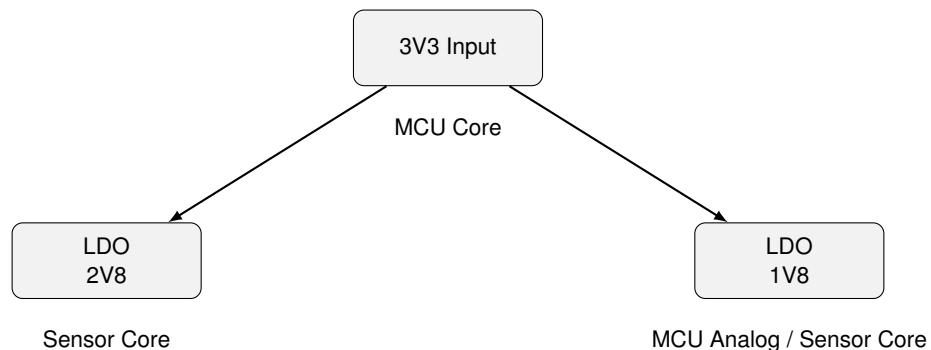


Figure 3: Simple power tree diagram of C2Sv0.1

The prototype should not require local power supplies, so any type of thermal testing is not required.

4.2.8 TAG

The TAG connector schematic was copied from v0 ([subsubsection 3.2.7](#)). No changes were made. Robert Howie had advised me that Jacob, electrical engineer from Binar, has had issues in the past with TAG connectors, so the COM port should act as a failsafe in-case the TAG connector becomes unreliable after a while.

Since the component is just a series of pads and the connector will not be attached to the board during any tests, the maximum characteristics are not a concern (even though they aren't listed).

4.3 PCB Layout and Review

4.3.1 Overview

Both the processor and sensor PCBs use the 4-layer, 1oz, 1.6mm stackup and rules from [JLCPCB](#).

4.3.2 Component Placement

I based the processor board around the STM chip. After my mistakes from v0, I learnt that it's more important to place the most dense components (trace wise) in the centre over everything else. I placed the processor on a 45deg angle so I could easily route all pins on the left and right side to the left and right sides of the PCB. Moreover, the decoupling capacitors and some resistors were also placed on angles, but this was purely to save space and let traces pass between them.

The sensor board barely had any components, so I just placed the image sensor in the centre for simplicity and kept the PLL clock close to the IC.

I kept the USB-C, FPC and uSD connectors each on their own side of the board to avoid unnecessary trace overlap, and placed the SDRAM slightly far away from the F7 to give me enough space to route to each pin without being farther away than needed.

I kept each 'room' (silkscreen box) near each other and avoided spreading components out. This

helped me keep track of each schematic sheet and should speed up the pick-and-place process since each set of components (LDOs, buttons, etc.) belong to one area of the PCB.

Lastly, I tried keeping all the decoupling capacitors as close to their ICs as possible. Some capacitors are a little further away than other since there just wasn't space directly next to the pin, but they're all within $\pm 5\text{mm}$ of where they need to be, so I doubt there will be any issues during testing.

4.3.3 Routing

The critical, higher-speed signals, like DCMI, clocks, memory and the uSD card were all routed first. This helped prioritise signal integrity on these critical traces since I could utilise the entire PCB. Moreover, this helped with the placement of components, since I was able to identify early on where free space exists and where I can potentially route traces. The DCMI signals were length matched to within $\pm 5\text{mm}$ and QSPI was delay matched within 50ps (DCMI did not require length matching at this speed according to [the community](#), but it was included just as a precaution). Moreover, [the community](#) also suggested that Altium's default accordion/sawtooth pattern for length matching is okay for the current frequency application ($\leq 160\text{MHz}$).

90 degree bends were avoided at all costs to prioritise signal integrity and avoid signal reflections, instead, two 45 degree turns were used. Where applicable, traces were set to exit and enter pads at 90 degrees to avoid any potential manufacturing issues. Teardrops were used to ease transitions between thin traces and larger pads. This was accomplished using Altium Designer's built-in teardrop generator, where the teardrop style was set to 'curved' and only 'Adjust teardrop size' was selected.

Where applicable, I avoided layer transitions to prioritise signal integrity. This helped keep as many traces as possible on the same layer, and avoid needing transfer vias which take up extra space.

Lastly, the analog signals (TEMP and CSA_OUT) were both kept away from digital signals to minimise the impact of coupling. Although the TEMP trace is kept adjacent to STANDBY, STANDBY is a signal that will not often switch between high and low states so it should not have any significant impact on the TEMP trace (especially if an average is taken on the temperature values).

4.3.4 Power Distribution

The six-layer PCB stackup uses two ground layers and one power layer. The two grounds were stitched using Altium Designer's stitching tool, a via diameter of 1.0/0.5mm and grid spacing of 5mm were selected. A spacing of 10mm was selected because using [subsection A.4](#), the required spacing for up to 160MHz (QSPI speeds) is 60mm, but the board was too small to have a 60mm spacing so 10mm was a good compromise.

The power layer used one large 3V3 polygon pour with the other voltage domains poured over local areas. To ensure that the 3V3 pour was uniformly connected, any long traces (like those reaching from the LDOs to the sensor connector) were placed on the signal layers.

4.3.5 Vias

Vias were a difficult aspect of the PCB. As discovered in [section 3](#), vias were one of the main culprits to the spike in production cost - the dimension requirements inside the BGA packages meant it was not possible to produce the board within the manufacturer's standard precision price range. As found in an early release of C2Sv0.1, the larger package sizes allowed for larger via dimensions, hence a significant reduction in production cost (approximately from \$150AUD to \$40AUD on the processor). The increased via size introduced some clearance issues that were discovered throughout the PCB development cycle, but this was not a major hurdle.

Transfer vias were included to maintain continuous return current paths and improve signal integrity [3]. One transfer via was shared between 1 or 2 through hole vias (with one or two cases sharing three). These were placed as close as reasonably possible to reduce the loop inductance without introducing voiding [16].

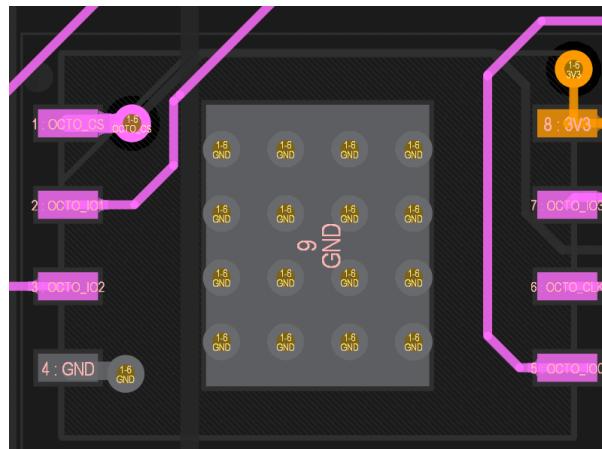


Figure 4: Power pad used on QSPI memory chip.

I did not add stitching or transfer vias on test points since those typically expose the copper on the same layer as the signal. Where test points have been used as through hole vias, transfer or stitching vias had been included.

On the sensor board, you may find that transfer vias have a trace connecting them to another standard via on the ground layer. This has no purpose since the ground pour will blanket the entire layer, it was included just to avoid net antennae violations from the DRC.

4.3.6 Ground Fill

Typically, ground fills may be helpful on lower-layer boards (1 or 2 layers) where good ground return paths are not available (slide 10 [13]). Since the processor board has ground layers adjacent to every signal layer, and no signal layers are adjacent to each other, it was deemed not necessary to include ground filling on the processor board [13]. If ground filling was included, it would require several calculations to determine the required stitching via spacing, as well as a reconstruction of the signal layout to accommodate pours and fills inbetween each signal - a resource-costly step for minimal improvements.

Additionally, it was decided that the sensor board did not require ground filling. Because of the several parallel routes in such close quarters, stitching vias could not be added, so any ground filling that wasn't correctly stitched would couple parallel lines. If the minimum neck length of the pour was increased to avoid long single-ended regions, then the ground fill would only pour on areas outside of the BGA, so including it would not have much effect.

4.3.7 Power Pad

For thermal management on the QSPI memory chip, I used vias of 0.6/0.3mm diameter spaced at 1mm to provide thermal relief to the component (see [Figure 4](#)) [9].

4.3.8 Coupling

Coupling, or cross-talk, refers to the injection of electromagnetic (EM) energy from one trace to another [18] [13]. In PCBs like C2Sv0.1, coupling is a factor that can greatly impact the performance of the board [13]. Although its impacts are more apparent in analog circuits, neglecting its presence could lead to unexpected behaviour.

Coupling was found to be an issue when routing the DCMI and other sensor traces. Since the traces were tightly packed together on the STM chip and FFC connector, early versions of the board were susceptible to coupling issues as seen in [Figure 5](#). Using the Saturn PCB Toolkit, with an original trace spacing of 0.5mm, coupled distance of 40mm, rise/fall time of 4.2ns (according to a maximum 64Mhz DCMI clock speed on the U5 and V_{DDIO2} voltage of 3.3V, with a minimum load



Figure 5: Example of bad spacing on DCMI traces in an early version of C2Sv0.1.

capacitance), the coupled voltage would be 2.3V. Increasing this spacing to 0.75mm changes the coupling voltage to 1.67V. According to Table 95 of [25], the minimum voltage to register a high-level input is 1.85V, greater than the maximum coupling voltage of 1.67V, so there should not be any coupling issues. Additionally, the calculated 1.67V only corresponds to the maximum coupling voltage. If the DCMI clock frequency was reduced, then the coupling voltage would be around 0.9V, safe from triggering high-level inputs.

All other routes were able to adhere to adequate trace spacing.

4.3.9 Silkscreen

Since this PCB houses several circuits, it was important to use the silk-screen to promote good labelling. Circuits like uSD connector were enclosed in a chamfered rectangle to show which components contribute to the circuit. All test points used the 'Arial Black' font instead of the 'Arial Narrow' so I can easily identify where the testpoints are during manufacturing and testing.

You may find jokes on the silkscreen like "BRAIN GOES HERE", these hold no importance towards the actual PCB itself, they have just been included for entertainment purposes Once the components are placed, these jokes are hidden, so they will not obstruct any other silkscreens.

Silkscreen preparation was the last step of the PCB, so if you explore previous saves of this PCB, you will most likely see lots of silkscreen collisions.

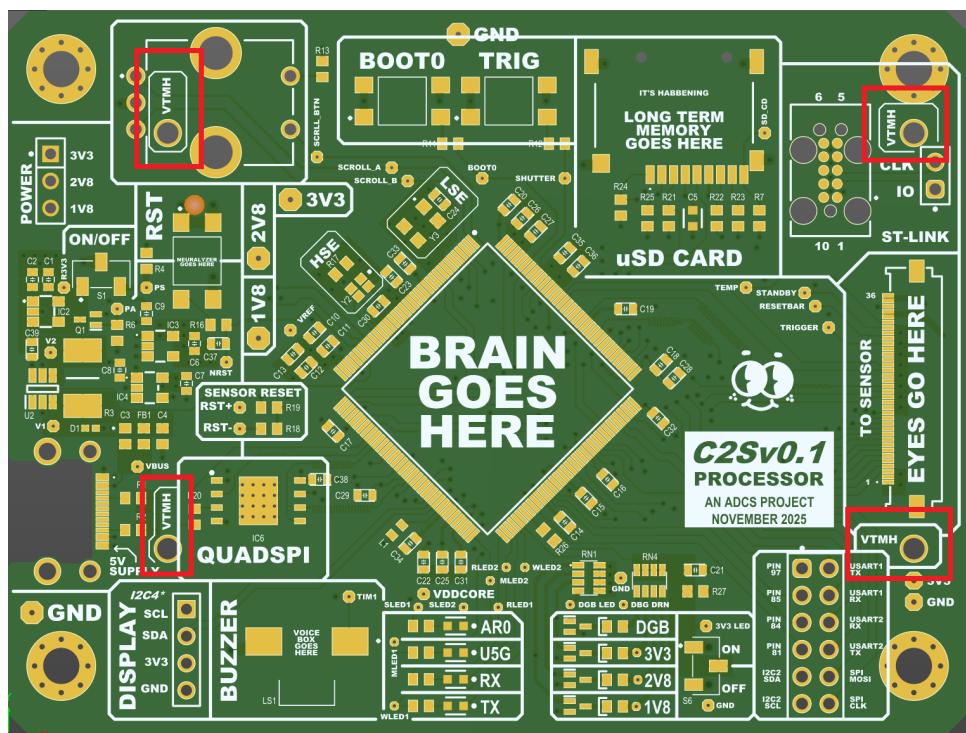
To save space on the PCB, some test point labels were shortened. Additionally, the test points inside the power LED array (i.e., next to 3V3, 2V8 and 1V8) are the drain testpoints (there was not enough space to include the silkscreen).

4.3.10 Mounting

After the schematic and PCB review, Tristan Davies suggested that a mounting pattern should be included to fit the PCB to the Spacecraft Lab (AERO3001) vibration table. As of November 21st, 2025, the shaker bracket specifications can be found [here](#) in the PAST SharePoint (alternatively, search through the Mechanical folder if it has been moved). The mounting bracket has two mounting patterns: M3 and M2 mounts. The M3 mounts are larger and appear in the corners of the bracket, whereas the M2 mounts are located further inside the shape. The M2 mounting bracket pattern was selected for C2Sv0.1 because the M3 pattern exceeds the board dimensions. I deemed it was not worth allocating so much physical space for a single feature of the board. As discussed in [subsubsection 4.3.9](#), the vibration table mounting holes are labelled as "VTMH" on the PCB.

PCB Name	Full Name
SCRL_BTN	Scroll Button
R3V3	Regulated 3V3
PA	Power Action
PS	Power State
DBG	Debug
DRN	Drain
SLED	Sensor LED
MLED	MCU LED
RLED	Read LED
WLED	Write LED
VTMH	Vibration Table Mounting Hole

Table 11: Explanation of truncated silkscreen labels



then checked by comparing it to the MIRA220 reference PCB to see if it looked okay from a surface level. It **MUST** be noted that the mount type refers to the thread and not the mount itself (see literature review for more). The MIRA220 uses a 22mm hole pitch spacing, whereas ours uses 20mm. The 20mm spacing requires M2 screws which PAST already has, whereas the 22mm spacing would require M2.2 which is harder to obtain. Since the 20mm spacing is correct, the lens should fit even if the border is misaligned because the only parts of the mount that need to be correct are the hole pitch and whether the sensor is in the centre:

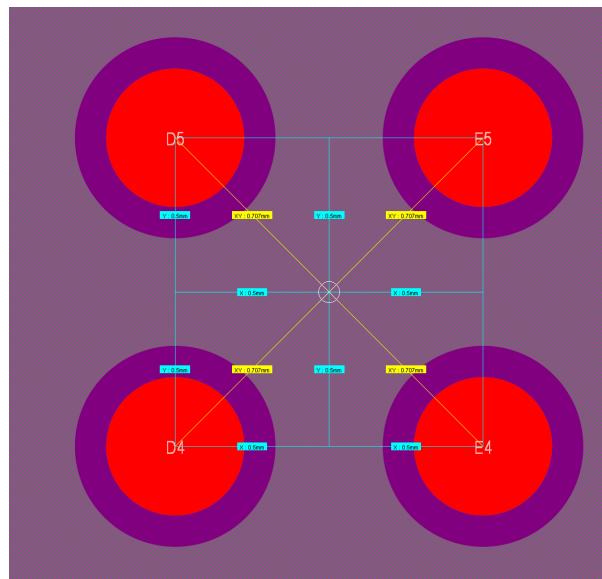


Figure 7: Measurements in Altium Designer confirming that the sensor is centered relative to the component origin. If the middle four balls are centered, all others must be since they are evenly spaced, therefore the sensor is centered.

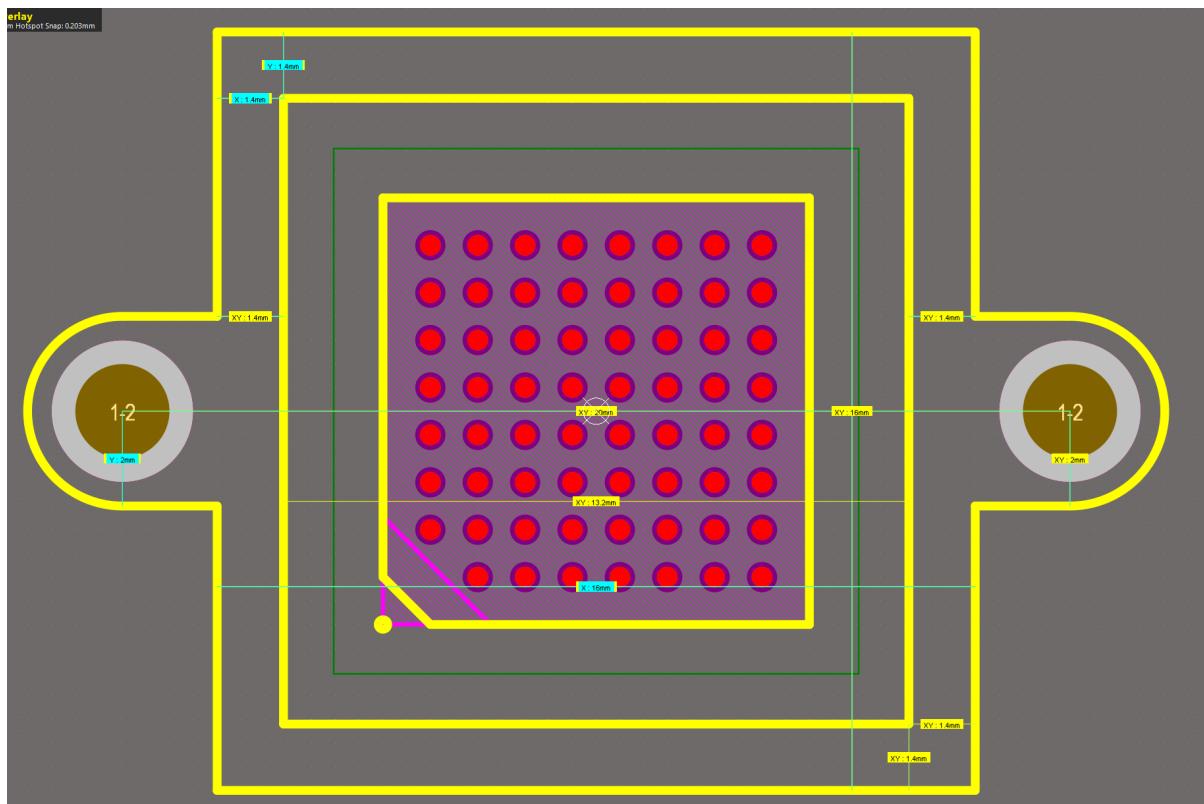


Figure 8: Measurements in Altium Designer confirming the measurements to [these specifications](#).

4.3.11**Fiducial Markers**

Fiducial markers were added to the sensor board (labelled "FM"). Three global markers were added near the corners of the board, and one local marker was added next to pin 1 of the BGA. Although a pick and place machine is not used, it may help AT&MWA when they attempt to place the BGA component.

Fiducial markers were also added to the corners of the processor PCB, although they will probably remain unused (no silkscreen label).

5 Manufacturing v0.1

The components for C2Sv0.1 were received on the following:

- **Element14**: This contained only the U5G processor. This was received on Tuesday, December 2nd 2025.
- **Digikey**: This included the image sensor and Raspberry Pi lens, received Thursday, December 4th 2025.
- **Mouser**: The rest of the project components were in the Mouser order. This was received on Monday, December 8th 2025.

Components were checked off and quantities were verified on Monday, December 8th 2025.

Unfortunately, the 3V3 LDO was not procured and the buttons PAST had in its inventory did not match those used in the PCBs (implications discussed later). To avoid waiting for another procurement round, I purchased the components, as well as my own STLINK/V2 so I did not have to borrow PAST's.

Additionally, I had realised I accidentally selected the wrong LEDs when designing the board. For [these LEDs](#) had mistakenly interpreted bicolor meaning two individual colours i.e., one colour when current goes in one direction and vice versa. Once I began testing, I quickly realised this was not the case.

5.1 Miscellaneous

The [knob](#) I had that was meant to be used on the rotary encoder did not fit, so, I 3D modelled one in Fusion360. I added some grip so it would be easier to rotate. The 3D print request was submitted on December 11th, accepted on the ..., printed on the ..., and received on the ...

UPLOAD .STL FILE TO GITHUB AND HREF IT.

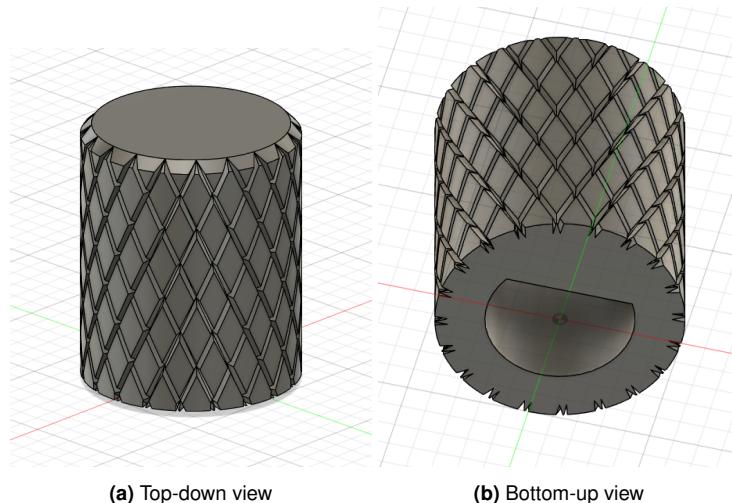


Figure 9: Knob modelled in Fusion360

5.2 Component Placement

It should be noted that 1x $0.1\mu\text{F}$ 0603 capacitor and 3x $10\text{k}\Omega$ 0603 resistors were accidentally wasted whilst making the sensor board as I used the values from the PCB instead of the commented values on the schematic - generic components with commented values were used instead of the exact components. This was an oversight during the design stage and should be improved on in further versions to make the project more readable and host a lower risk of confusion or misinterpretation.

Additionally, I had misinterpreted the product page for the [M12 lens mount](#). I had thought that for \$22.51AUD, I would get the Factory Pack Quantity (20), but I had blanked that the factory pack quantity and order quantity are independent from each other. If more mounts are required in the future, they should be sourced from cheaper suppliers like AliExpress (since at the end of the day it's just plastic).

5.2.1 Sensor Board

The sensor boards were partially manufactured on December 8th, 2025, directly after the Mouser components were received. All components except for the image sensor were successfully placed without any reflow issues.

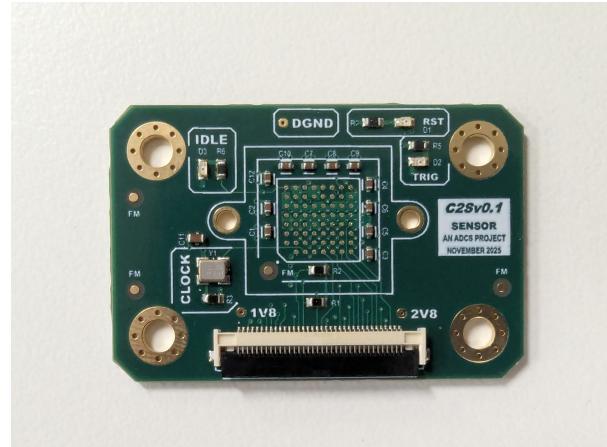
Prior to placing the components, I called Peter Larkin from AT&MWA to ensure that I was following the proper procedure to present them with usable PCBs. It was his recommendation to avoid putting solder paste on the BGA pads and continue normally with all other components. Fortunately, the sensor being the centre of the board made it easy to avoid the BGA pads. I applied solder paste to the edge of the spreader and completed motions from the the edge of the BGA pads to the edge of the PCB (opposed to one large side to side swipe). Repeating this four times for the four sides of the BGA allowed me to spread the paste evenly across all pads whilst avoiding the BGA.

The PCBs were placed on the hotplate (set at 240C) and were removed a couple seconds after the solder had reflowed. After completing a surface level examination of the boards, I noticed no immediate connection issues between components and pads. Unfortunately, I had realised I used the wrong LEDs on the sensor board (bicolour instead of single colour). However, since both have similar pad sizes this was not a large issue.

I then fit the M12 lens mount to one of the PCBs and screwed on the M12 Raspberry Pi Lens to ensure that it fit properly (see [Figure 10a](#)). To fit the mount to the PCB, the screws must be fed from below the PCB instead of above and down through the mount.



(a) C2Sv0.1 sensor board after M12 mount and lens fitted.



(b) Second view or stage of the sensor board.

Figure 10: Birds eye view of the C2Sv0.1 sensor board after intial manufacturing.

5.2.2 Processor Board

The first processor board was manufactured on December 9th, 2025, before the Tuesday project session. I placed all components except for the 3V3 LDO. I did try hand soldering the larger buttons onto the pads, however this did not work and I tore the pads off in the process of removing them.

I did encounter bridging issues with the U5G and FFC connector. I was no expert at removing bridges though, so it took a while, but they were eventually removed. I also had mistakenly used the wrong value capacitors, so a few 100nF 0603 capacitors were wasted.

Initial testing showed that there were no shorts before or after placement. Since I could not power the board by USB-C, I connected the 3V3 output of my Arduino Uno to the 3V3 pin on the PCB as a temporary power source (an intentional design feature). Probing each voltage showed that the LDOs were operating and outputting expected voltages.

Since this board was incomplete, I decided that this will be the board used for vibration table testing.

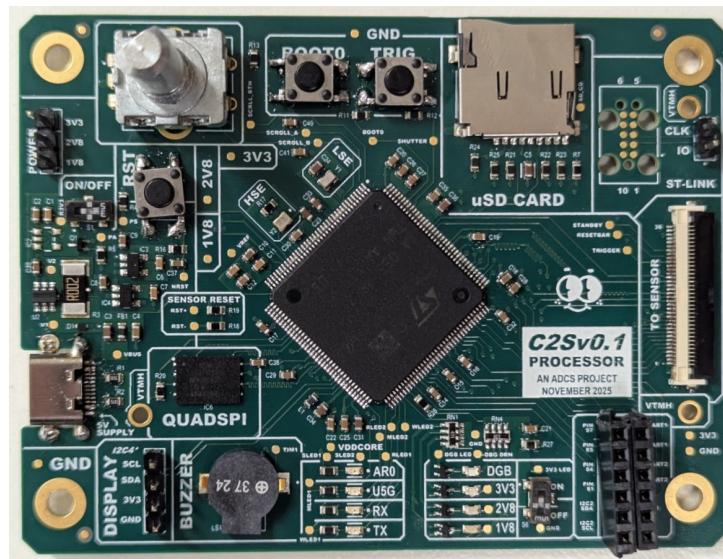


Figure 11: First C2Sv0.1 processor board after manufacturing with the large buttons still included.

5.3 AT&MWA Placement

Peter Larkin stated that they charge \$70AUD+GST per hour and assumed it shouldn't take longer than an hour.

On Thursday 11th December, 2025, I prepared the handover documents, and at 3pm on Friday 12th December, I handed over the two image sensors and sensor boards.

On Thursday 18th December, 2025, I was notified that the image sensors were placed and a payment of \$70AUD+GST was required.

6 Programming v0.1

banana

6.1 Initial Testing

6.1.1 LEDs

As mentioned in [subsubsection 5.2.2](#), the wrong LEDs were used to have one colour when current was passed one way and vice versa. Since current can only flow one way, all the anodes in STM32CubeIDE have been set to LOW:

```
// Set Anode of LEDs to LOW (GND)
HAL_GPIO_WritePin(MCU_LED_1_GPIO_Port, MCU_LED_1_Pin, 0);
HAL_GPIO_WritePin(READ_LED_1_GPIO_Port, READ_LED_1_Pin, 0);
HAL_GPIO_WritePin(WRITE_LED_1_GPIO_Port, WRITE_LED_1_Pin, 0);
HAL_GPIO_WritePin(SENSOR_LED_1_GPIO_Port, SENSOR_LED_1_Pin, 0);
```

6.1.2 Audio Buzzer

Moreover, it was discovered that there was an error with the audio buzzer circuit. I had not realised there was a recommended circuit at the bottom of page 2 in the [datasheet \[5\]](#). So, even though the U5G was outputting a correct PWM signal, the driving current was not enough to turn the buzzer on. Page 1 of the datasheet says that the component consumes a maximum of 110mA, whereas the [U5G datasheet](#) says that the maximum output current sourced by an I/O pin is 20mA, far lower than 110mA (so it is expected the buzzer cannot be heard) [\[25\]](#) [\[5\]](#).

6.1.3 Rotary Encoder

The rotary encoder, or dial, was essential for changing the sensor's configuration parameters without re-flashing the code. Unfortunately, its button was connected to a VSS pin on the U5 (for unknown reasons), making the feature unusable. To restore functionality, I repurposed the BOOT0 button to trigger the encoder's button logic within its ISR.

A struct was used to store all the variables for the encoder, just to keep the software organised. The configuration (cfg) settings (values to be changed) are nested within another struct just to keep the cfg variables in one place.

```
#define MAX_SETTINGS 3
#define STARTING_SETTING 0

typedef struct {
    int num;      // Number of settings
    int arr[MAX_SETTINGS]; // Array of settings containing values
    volatile int idx; // Settings index
} Settings_HandleTypeDef;

// Encoder handle structure
typedef struct {
    Settings_HandleTypeDef cfg;
    volatile uint8_t enc_a;
    volatile uint8_t enc_b;
    volatile uint8_t step;
    GPIO_TypeDef* pinA_port;
    uint16_t pinA;
    GPIO_TypeDef* pinB_port;
    uint16_t pinB;
} Encoder_HandleTypeDef;
```

Debouncing was a critical step to make the encoder work properly. Its mechanical design causes each conducting pad to vibrate and bounce, causing the encoder's natural behaviour to be unpredictable. [This video](#) suggested using pattern recognition to debounce the encoder. Assuming 'A' is going high and 'a' is going low, the encoder must follow 'A-B-a-b' to go clockwise and 'B-A-b-a' for anti-clockwise:

```

// Update encoder state
void Encoder_Update(Encoder_HandleTypeDef *encoder, uint16_t GPIO_Pin, uint8_t rising)
{
    // Read instantaneous state
    encoder->enc_a = HAL_GPIO_ReadPin(encoder->pinA_port, encoder->pinA);
    encoder->enc_b = HAL_GPIO_ReadPin(encoder->pinB_port, encoder->pinB);

    // Clockwise expected sequence (example)
    static const uint8_t CW_sequence[4][2] = {
        {0, 1}, {1, 1}, {0, 0}, {1, 0}
    };

    // Counterclockwise expected sequence
    static const uint8_t CCW_sequence[4][2] = {
        {1, 1}, {0, 1}, {1, 0}, {0, 0}
    };

    // Check CW
    if (GPIO_Pin == encoder->pinA && rising == CW_sequence[encoder->step][1]) {
        encoder->step++;
        if (encoder->step >= 4) {
            if ((encoder->cfg.arr[encoder->cfg.idx]) < 15) {
                (encoder->cfg.arr[encoder->cfg.idx])++; // one CW detent
            }
        }
        encoder->step = 0;
    }
    return;
}

// Check CCW
if (GPIO_Pin == encoder->pinB && rising == CCW_sequence[encoder->step][1]) {
    encoder->step++;
    if (encoder->step >= 4) {
        if ((encoder->cfg.arr[encoder->cfg.idx]) > 0) {
            (encoder->cfg.arr[encoder->cfg.idx])--; // one CCW detent
        }
    }
    encoder->step = 0;
}
return;
}

// Invalid transition → reset step
encoder->step = 0;
}

```

Testing showed that there was no debouncing and I could spin the dial quickly with minimal tracking error.

6.1.4

Current Sense Amplifier

The current sense amplifier (CSA) outputs an analog value dependent on the voltage drop across the connected shunt resistor.

In the STM32CubeIDE, a struct was used to organise the various attributes of the IC:

```

typedef struct {
    float Rshunt;           // Shunt resistor value [Ohms]
    float Vcc;              // ADC reference / supply voltage [V]
    float Vbus;             // Bus voltage if needed [V]
    int gain;               // CSA gain
    volatile float adc;     // ADC value read from pin
    volatile float voltage; // Measured voltage across shunt [V]
    volatile float current; // Calculated current [A]
    volatile float power;   // Calculated power across load [W]
    volatile float sys_power; // System power [W]
} CSA_HandleTypeDef;

```

Moreover, various functions were constructed to generate these values, all called inside a single update function:

```

void CSA_Update(CSA_HandleTypeDef *csa, ADC_HandleTypeDef *hadc)
{
    CSA_ADC(hadc, csa);
    CSA_VOLTAGE(csa);
    CSA_CURRENT(csa);
    CSA_POWER(csa);
    CSA_SYSTEM_POWER(csa);
}

```

Initial testing showed that the shunt resistor procured was the wrong value. The maximum value was calculated from the [CSA datasheet](#) on page 24 (equation 2) and I selected an arbitrary value based on examples throughout the datasheet [27]. This assumption failed to consider how the output analog value related to the ADC. The U5 can accept analog values from 0-3.3V on its ADC pins, which means for a gain of 50 with the selected CSA (Table 8-1, page 26 of datasheet) the maximum voltage drop across the shunt resistor can be $3.3/50 = 66\text{mV}$ [25] [27]. The board is not expected to draw more than 1A on 3V3 logic (based on component power estimations and current carrying capacity of traces), which equates to 660mA at 5V logic. Using Ohm's law, we can find the maximum resistance of the shunt is $0.066/0.66 = 100\text{m}\Omega$ (significantly larger than the selected $2\text{m}\Omega$). To combat the noise issue with the lower value shunt resistor, the CSA was sampled 100 times to obtain consistent readings.

```

powerSum += csa.sys_power;
uint32_t now = HAL_GetTick();
if ((now - prevTime) >= POWER_AVG_WINDOW_MS) {
    uint32_t samples = POWER_AVG_WINDOW_MS / SAMPLE_PERIOD_MS;
    powerAvg = powerSum / samples * 1000; // Take average and convert to mW

    powerSum = 0;
    prevTime = now;
}

```

6.1.5 S-R Latch

As shown in [Figure 12a](#), the PCB included a set-reset latch for the power switch to ensure the processor and other components were powered down correctly. Unfortunately, the latch was included in the wrong location for the circuit to perform as expected. The MOSFET was meant to act as a secondary conducting line (the reset) once the switch was opened. Since the circuit was implemented after the 3V3 LDO, the voltage drop across the MOSFET would cause the U5 to lose power and eventually reset the base of the MOSFET. As you can see in [Figure 12b](#), after the switch is opened, there is single instance where VCC is equal to 3.3 minus the voltage drop across the MOSFET. Since this new VCC value falls outside the operating range of VCC for the U5, the processor powers down and the decoupling and bulk capacitors create the decay behaviour seen afterwards. If this was implemented before the 3V3 LDO, the voltage drop across the MOSFET would not affect VCC since the LDO regulates the voltage down to 3V3 regardless of whether the voltage before it is 5V or 4.3V.

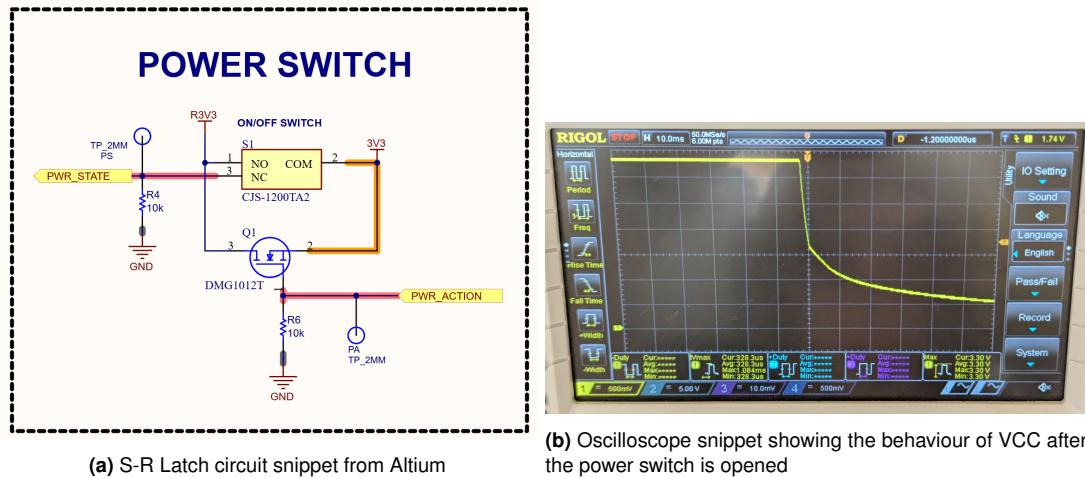


Figure 12: Set-Reset (S-R) latch from C2Sv0.1

6.1.6 Buttons

Using the STM32CubeIDE, I set each button's pin to be an interrupt and enabled the interrupt in the NVIC tab. I then used a basic software debounce:

```
const uint32_t debounce_ms = 100; // debounce period in milliseconds
uint32_t now = HAL_GetTick();
if ((now - last_button_time) >= debounce_ms)
{
    HAL_GPIO_WritePin(TRIGGER_GPIO_Port, TRIGGER_Pin, 1);
    last_button_time = now;
}
```

6.1.7 uSD

Since the U5 can do internal pull-ups on the data lines, the resistors were not required.

References

- [1] *A guide to technical report writing*, 2015. [Online]. Available: https://ias.ieee.org/wp-content/uploads/2023/06/2020-01-16_IET_Technical_Report_Writing_Guidelines.pdf
- [2] AMD, *Qspi — ultrascale architecture pcb design user guide (ug583)*, version 1.28, Released 19 December 2024, 2024. [Online]. Available: <https://docs.amd.com/r/en-US/ug583-ultrascale-pcb-design/QSPI>
- [3] *Anonymous Forum Contributor*. “Importance of transfer vias.” Accessed 2025-11-08; discussion thread on board-level via and coupling issues. [Online]. Available: <https://www.eevblog.com/forum/eda/importance-of-transfer-vias/>
- [4] S. Community. “Schematics for microsd on sdmmc.” Accessed: 2025-10-16. [Online]. Available: <https://community.st.com/t5/stm32-mcus-products/schematics-for-microsd-on-sdmmc/td-p/75062>
- [5] C. Devices, *Cbt-09427-smt-tr magnetic buzzer datasheet*, https://au.mouser.com/datasheet/3/6118/1/cbt_09427_smt_tr.pdf, Accessed: 2025-12-12, 2024.
- [6] S. C. Forum. “Quad spi memory interfacing on stm32u5g9vjtxq microcontroller.” Accessed: 2025-10-16. [Online]. Available: <https://community.st.com/t5/stm32-mcus-products/quad-spi-memory-interfacing-on-stm32u5g9vjtxq-microcontroller/td-p/801941>
- [7] *Guidelines for aluminium electrolytic capacitors*. [Online]. Available: https://cougarelectronics.com/wp-content/uploads/2019/09/guidelines_for_aluminium_electrolytic_capacitors.pdf
- [8] *Ieee editorial style manual for authors*, 445 Hoes Lane, Piscataway, NJ 08854 USA, 2024. [Online]. Available: <https://journals.ieeeauthorcenter.ieee.org/wp-content/uploads/sites/7/IEEE-Editorial-Style-Manual-for-Authors.pdf>
- [9] T. I. Incorporated, “Powerpad layout guidelines,” Texas Instruments Incorporated, Tech. Rep. SLOA120, 2006, <https://www.ti.com/lit/an/sloa120/sloa120.pdf> (accessed 2025-11-08).
- [10] T. Instruments, “Am437x starterkit evm — qspi nor flash memory reference design,” Texas Instruments Inc., Reference Design TIDR808, 2014, Accessed: 2025-10-16. [Online]. Available: <https://www.ti.com/lit/df/tidr808/tidr808.pdf>
- [11] T. Instruments, “An engineer’s guide to usb type-c®,” Texas Instruments Incorporated, Engineering Bulletin SLYY228, 2020, Accessed: 2025-10-16. [Online]. Available: <https://www.ti.com/lit/eb/slyy228/slyy228.pdf>
- [12] T. Instruments, *Analog and digital grounding (part 1 and part 2)*, <https://www.ti.com/lit/an/slyt499/slyt499.pdf> and <https://www.ti.com/lit/an/slyt512/slyt512.pdf>, Application Note SLYT499 and SLYT512, 2011.
- [13] A. Kay, *Crosstalk on pcb layouts*, TI Precision Labs - ADCs Presentation Quiz (PDF), Available at: <https://www.ti.com/content/dam/videos/external-videos/ja-jp/9/3816841626001/6307563213112.mp4/subassets/crosstalk-on-pcb-layouts-presentation-quiz.pdf> (accessed 2025-11-08), 2022.
- [14] R. Manchukonda, *How to choose a shunt resistor - presentation and quiz*, TI Precision Labs - Current Sense Amplifiers (PDF), Available at: <https://www.ti.com/content/dam/videos/external-videos/en-us/4/3816841626001/6076324596001.mp4/subassets/current-sense-amplifiers-how-to-choose-a-shunt-resistor-presentation-quiz.pdf> (accessed 2025-11-08), 2019.
- [15] T. Ogo and T. E. Community. “K2g pcb guidelines for qspi.” Thread: “K2G PCB Guidelines for QSPI” (locked). [Online]. Available: <https://e2e.ti.com/support/processors-group/processors/f/processors-forum/541174/k2g-pcb-guidelines-for-qspi>
- [16] Z. Peterson. “Everything you need to know about stitching vias.” Accessed 2025-11-08. [Online]. Available: <https://resources.altium.com/p/everything-you-need-know-about-stitching-vias>
- [17] *Reference guide*, 445 Hoes Lane, Piscataway, NJ 08854 USA, 2025. [Online]. Available: <https://docs.google.com/document/d/1j1L96U2NagwWI9MEVDNVKt9pXxRzTH7h3krI3Mb6wZE/edit?tab=t.0#heading=h.b2e0set9htjw>
- [18] L. Ritchey. “Crosstalk or coupling in high-speed pcb design.” Accessed 2025-11-08. [Online]. Available: <https://resources.altium.com/p/crosstalk-or-coupling>

- [19] STMicroelectronics. "Basics of power supply design for mcu." Accessed: 2025-10-16. [Online]. Available: https://wiki.st.com/stm32mcu/wiki/Basics_of_power_supply_design_for MCU#SMPS
- [20] STMicroelectronics, "Getting started with stm32u5 mcu hardware development," STMicroelectronics, Application Note AN5373, version Rev 7, Nov. 2023, Accessed: 2025-10-13. [Online]. Available: https://www.st.com/resource/en/application_note/an5373-getting-started-with-stm32u5-mcu-hardware-development-stmicroelectronics.pdf
- [21] STMicroelectronics, "Introduction to digital camera interface (dcmi) for stm32 mcus," STMicroelectronics, Application Note AN5020, 2024, Revision 4, November 2024. [Online]. Available: https://www.st.com/resource/en/application_note/an5020-introduction-to-digital-camera-interface-dcni-for-stm32-mcus-stmicroelectronics.pdf
- [22] STMicroelectronics. "Introduction to external serial memories (xspi) interoperability for stm32." Accessed: 2025-10-16. [Online]. Available: https://wiki.st.com/stm32mcu/wiki/Introduction_to_external_serial_memories_XSPI_interoperability_for_STM32
- [23] STMicroelectronics, "Quadspi interface on stm32 microcontrollers and microprocessors," STMicroelectronics, Application Note AN4760, 2018, Accessed: 2025-10-16. [Online]. Available: https://www.st.com/resource/en/application_note/an4760-quadspi-interface-on-stm32-microcontrollers-and-microprocessors-stmicroelectronics.pdf
- [24] STMicroelectronics. "Stm32f7 peripheral dcmi." Accessed: 2025-10-18. [Online]. Available: https://www.st.com/content/ccc/resource/training/technical/product_training/group0/90/5a/91/24/1a/14/4b/40/STM32F7_Peripheral_DCMI/files/STM32F7_Peripheral_DCMI.pdf/jcr:content/translations/en. STM32F7_Peripheral_DCMI.pdf
- [25] STMicroelectronics, *Stm32u5g7vj datasheet*, version Rev 3, Accessed: 2025-10-13, STMicroelectronics, Geneva, Switzerland, Mar. 2025, p. 353. [Online]. Available: <https://www.st.com/resource/en/datasheet/stm32u5g7vj.pdf>
- [26] M. Technology, "En562805 application note," Microchip Technology Inc., Application Note EN562805, Accessed: 2025-10-16. [Online]. Available: <https://ww1.microchip.com/downloads/en/Appnotes/en562805.pdf>
- [27] Texas Instruments, *Ina181 bidirectional, low- and high-side voltage output current-sense amplifier datasheet*, Rev. H, 62 pages, 2023. [Online]. Available: <https://www.ti.com/lit/ds/symlink/ina181.pdf>

Appendices

A Supporting Calculations

A.1 C-L-C Values

Model the ferrite bead as a series inductance L at low-mid frequencies. The two capacitors C_1 (input side) and C_2 (output side) form a pi filter with that series inductance. The pi network has a resonance frequency f where the L and the effective capacitance interact:

$$C_{eq} = \frac{C_1 \times C_2}{C_1 + C_2} \quad (1)$$

The resonance / cutoff frequency of the $L-C_{eq}$ combination is:

$$f = \frac{1}{2\pi\sqrt{L \times C_{eq}}} \quad (2)$$

At frequencies well above f_r the filter attenuates; below f_r the caps and ferrite appear largely as bypasses so DC and low frequency pass through.

For ferrite beads, its resonant frequency is typically 100MHz (unless stated otherwise). At frequencies well below resonance, the bead behaves approximately like an inductor:

$$Z \approx j2\pi f L \quad (3)$$

Rearranging:

$$L \approx \frac{Z}{2\pi f} \quad (4)$$

A.2 Transconductance of Oscillator

The gain margin ratio is determined by the formula $\text{gain}_{\text{margin}} = g_m/g_{\text{mcrit}}$, where g_m is the oscillator transconductance specified in the STM32 datasheet. The HSE oscillator transconductance is in the range of a dozen of mA/V, while the LSE oscillator transconductance ranges from a few to a few dozens of $\mu\text{A}/\text{V}$, depending upon the product. g_{mcrit} is defined as the minimal transconductance required to maintain a stable oscillation when it is a part of the oscillation loop for which this parameter is relevant. g_{mcrit} is computed from oscillation loop passive components parameters.

Assuming $C_{L1} = C_{L2}$, and that the crystal sees the same C_L on its pads as the value given by the crystal manufacturer, g_{mcrit} is expressed as follows:

$$g_{\text{mcrit}} = 4 \times \text{ESR} \times (2\pi F)^2 \times (C_0 + C_L)^2 \quad (5)$$

where:

- ESR is the equivalent series resistance.
- C_0 is the crystal shunt capacitance.
- C_L is the crystal nominal load capacitance.
- F is the crystal nominal oscillation frequency.

If the oscillator transconductance parameter (g_m) is specified, make sure that the gain margin ratio ($\text{gain}_{\text{margin}}$) is bigger than 5.

A.3 Power Draw Estimation

If we have a series current-sense resistor of resistance R (Figure 13), the current flowing through the resistor is $I = V_d/R$, where V_d is the voltage drop across the resistor. Multiplying V_{IN} to both sides gives

$$V_{IN}I = \frac{V_d V_{IN}}{R} \quad (6)$$

Since $P = VI$, the power going into the circuit is $P_{IN} = \frac{V_d V_{IN}}{R}$.

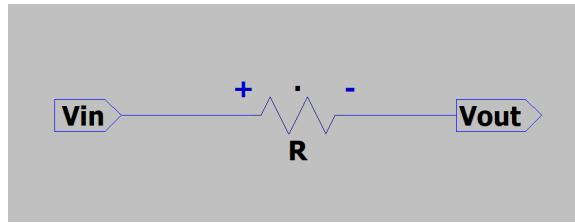
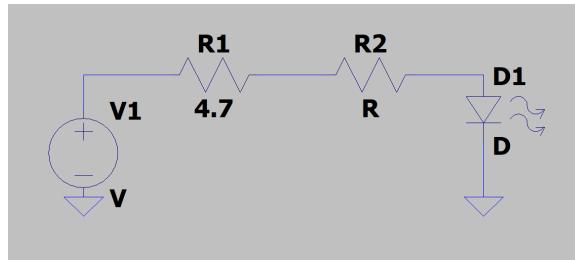


Figure 13: Resistor in series with a voltage source VIN with output voltage VOUT.

This estimation was tested on an LED circuit to gauge the accuracy of the process (Figure 14). Using Equation 6, I was able to successfully identify the power draw within 5% of the approximate power (where the approximate power was the current and voltage drop measured from a multimeter). These promising results show that the resistor circuit, coupled with the STM Power Consumption Calculator, will provide a strong estimation of the camera's power consumption (although it should be noted that the camera's circuit is far more complex than a simple LED set-up, so it is not guaranteed the recorded value will be within 5%).



(a) Power estimation circuit.

Voltage (V)	Voltage Drop Resistor (mV)	Resistance (Ω)	LED Voltage Drop (V)	Bulk Resistor (Ω)	Current (mA)	Estimated Power (mW)	Approximate Power (mW)	Error (%)
3.3	13.2	4.75	1.95	467	2.70	9.17	8.91	3%
5	28	4.75	2	467	6.00	29.47	30.00	-2%
3.3	0.08	4.75	1.64	100000	0.02	0.06	0.06	-1%
5	0.155	4.75	1.68	100000	0.03	0.16	0.17	-1%
5	0.5	4.75	1.74	30000	0.10	0.53	0.50	5%
3.3	0.25	4.75	1.7	30000	0.05	0.17	0.18	-3%
5	6.8	4.75	1.9	950	1.51	7.16	7.55	-5%
3.3	14.5	4.75	1.94	950	3.15	10.07	10.40	-3%

(b) Results gathered from LED power estimation. Calculated power was within 5% of approximate power.

Figure 14: Overview of the LED power estimation: (a) the circuit used, (b) measured results.

A.4 Stitching Via Calculations

For a given frequency, the center-to-center pitch between the vias should be less than approximately:

$$L < \frac{c}{8f\sqrt{Dk}} \quad (7)$$

B C2Sv0 Breakout Board

B.1 PCB Layout

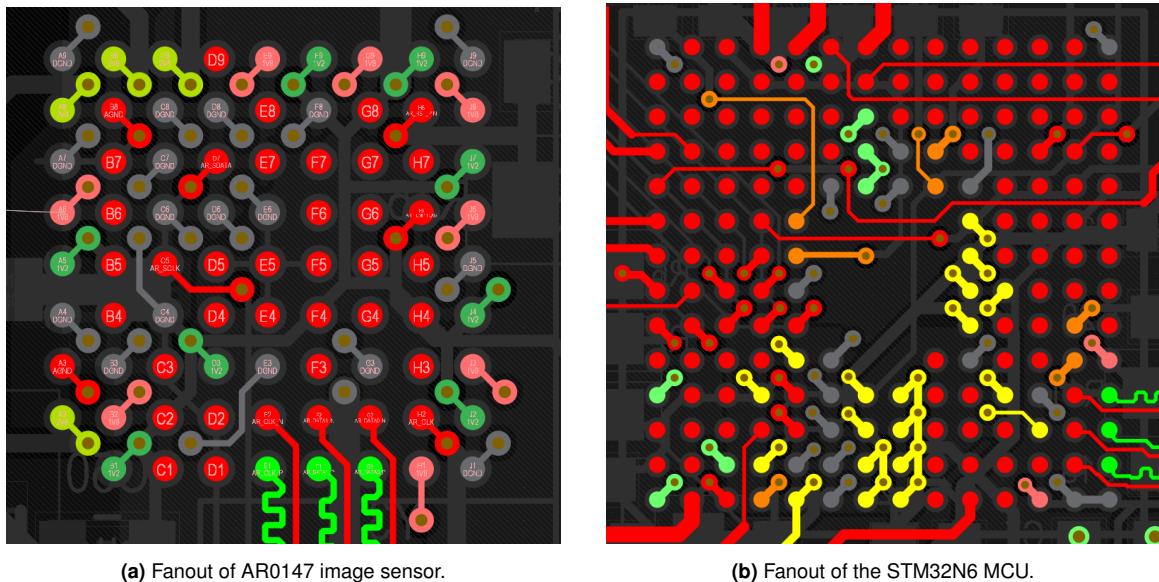


Figure 15: BGA fanout methods used in the C2S Breakout Board PCB.

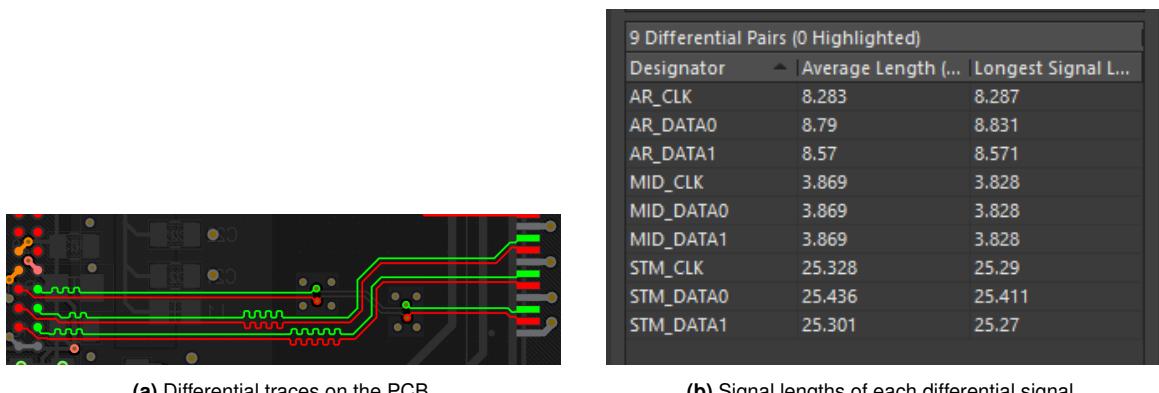


Figure 16: MIPI CSI-2 differential pairs on the C2S Breakout Board.

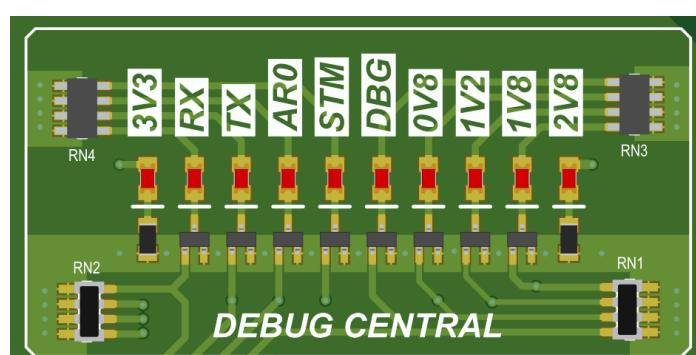


Figure 17: Debug LEDs on C2S Breakout Board.

C C2Sv0.1 Breakout Board

C.1 STM32U5G Power Management

C.1.1 Power Supplies

The STM32U5 requires various independent supplies for specific peripherals. The following breakdown is a justification of the list from page 3 of the getting started guide.

- **$V_{DD} = 3.3V$**
 V_{DD} serves as the primary digital power domain for the STM32U5, supplying the processor core's internal regulator, general-purpose I/O banks, and most on-chip peripherals. A 3.3V rail was selected as it represents the most common logic level for mixed-signal embedded systems and is natively supported by all external components in the design, including the SPI flash memory, sensor control lines, and voltage-level interfaces. Selecting 3.3V simplifies the power architecture by allowing a single shared rail across the majority of devices, avoiding the need for intermediate level shifters and reducing regulator count. It also provides greater noise margin and signal integrity on high-speed digital traces compared to lower-voltage domains, at a modest increase in power consumption. The choice therefore balances component compatibility, design simplicity, and system robustness.
- **$V_{DDA} = 1.8V$**
 V_{DDA} is the external-analog power supply for A/D converters, D/A converters, voltage reference buffer, operational amplifiers, and comparators [20]. It must be independent from V_{DD} when the mentioned peripherals are used [20]. Selecting 1.8V helps reduce the number of different voltage sources, and subsequently number of components, on the board since 1.8V is already a power supply for the image sensor. Moreover, the lower voltage helps reduce overall power consumption.
- **$V_{DDSMPS} = 3.3V$**
 V_{DDSMPS} is the external power supply for the SMPS step-down converter [20]. When in use, it must be connected to the same pin as V_{DD} [20]. So, its voltage is the same as V_{DD} , 3.3V.
- **$V_{LXSMPS} = V_{DDCORE}$**
This is the output pin from the internal SMPS step-down converter and has a typical voltage of 1.21V (pg. 166 [25]) [20].
- **$V_{DD11} = V_{DDCORE}$**
This is the digital core supply for the processor and is supplied by V_{LXSMPS} [20]. It requires a total of $4.7\mu F$ external capacitors, two $2.2\mu F$ bulk capacitors and $100nF$ decoupling capacitor for each pin (in this case) [25] [20]
- **$V_{CAP} = \text{UNAVAILABLE}$**
These pins are not offered on the selected U5G processor.
- **$V_{DDUSB} = 3.3V$**
This is the external-independent power supply for USB transceivers [20]. Since the USB-C connector is only being used to supply power, the transceiver functionality was not applicable, so, the pin was connected to V_{DD} as specified by [20].
- **$V_{DD11USB} = \text{UNAVAILABLE}$**
This pin is not offered on the selected U5G processor.
- **$V_{DDIO2} = V_{DDCORE}$**
 V_{DDIO12} is the external power supply for 14 I/Os (port G[15:2]) and its voltage level is independent from the V_{DD} voltage [20]. It was connected to V_{DDA} since we require the I2C lines on PG13 and PG14 to run at 3V3 logic. If these I2C lines were not connected, V_{DDIO2} would be connected to V_{DDCORE} : The 1.8V and 2.8V voltage levels are produced by the local LDOs, whereas the V_{DDCORE} voltage is produced by a much more efficient SMPS step-down converter [19], which means there are less losses between regulating from SMPS than an LDO. This also allows the pin to still be powered if there are any issues with the 2.8V LDO (if the 1.8V LDO had issues, VREF+ would not be in its operating range and the processor would not function).
- **$V_{BAT} = 3.3V$**
This is the power supply when V_{DD} is not present for RTC, TAMP, LSE, backup registers and

optionally backup SRAM [20]. The pin has been shorted to V_{DD} since the camera will not be placed in any situation where it still needs to operate without V_{DD} (if the main power goes out on the CubeSat of HAB, the camera cannot do anything until the main power comes back).

- $V_{REF-} = 1.8V$

V_{REF-} is the ground reference voltage for the processor. When the VREF+ pin is double-bonded to V_{DDA} in a package, the internal VREFBUF is not available, and must be kept disabled, V_{REF-} must always be equal to V_{SSA} [20].

- $V_{REF+} = 1.8V$

As outlined in [19], the simplest solution to provide V_{REF} is to use V_{DDA} , 1.8V. Since it is connected to V_{DDA} , it admits to the 0.4V voltage difference between V_{REF} and V_{DDA} (pg. 163 [25]).

- $V_{DDDSI} = 3.3V$

This is the external power supply for the DSI controller [20]. It is provided externally through the V_{DDDSI} supply pin, and must be connected to the same supply as V_{DD} pin [20].

- $V_{DD11DSI} = V_{DDCORE}$

This is the external power supply for the DSI transceiver and must be connected to V_{DD11} [20].

C.1.2 Decoupling Capacitors

The decoupling capacitors followed the getting started guide and datasheet power supply schemes (pg. 14 [20] and pg. 161 [25]). The 0603 package was used for all decoupling and capacitors and inductors to comply with the PAST Avionics standard.

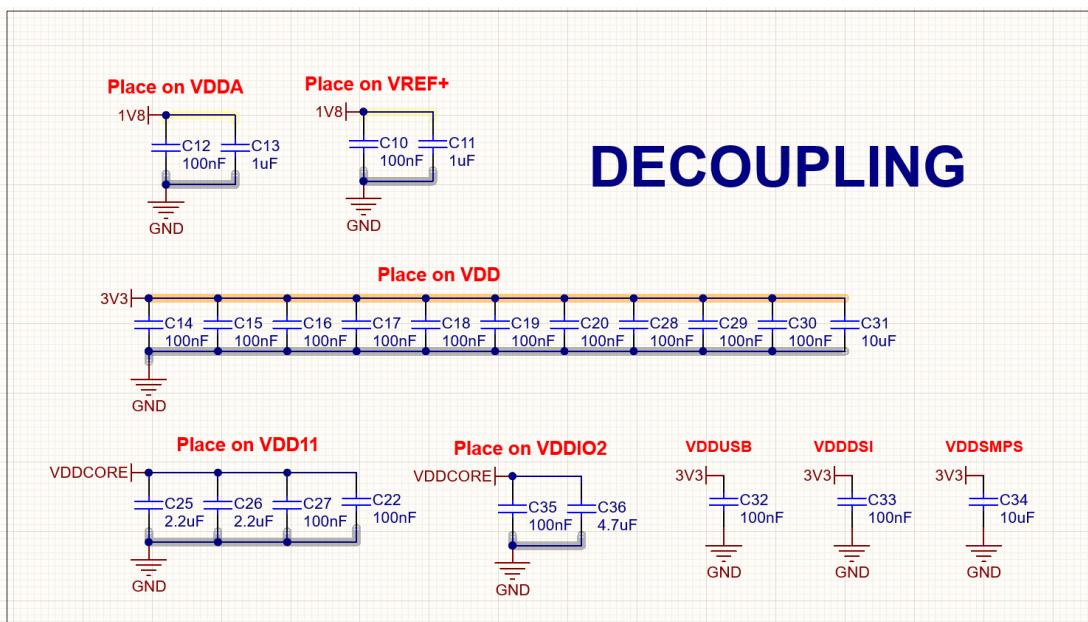


Figure 18: Decoupling capacitors for STM32U5G

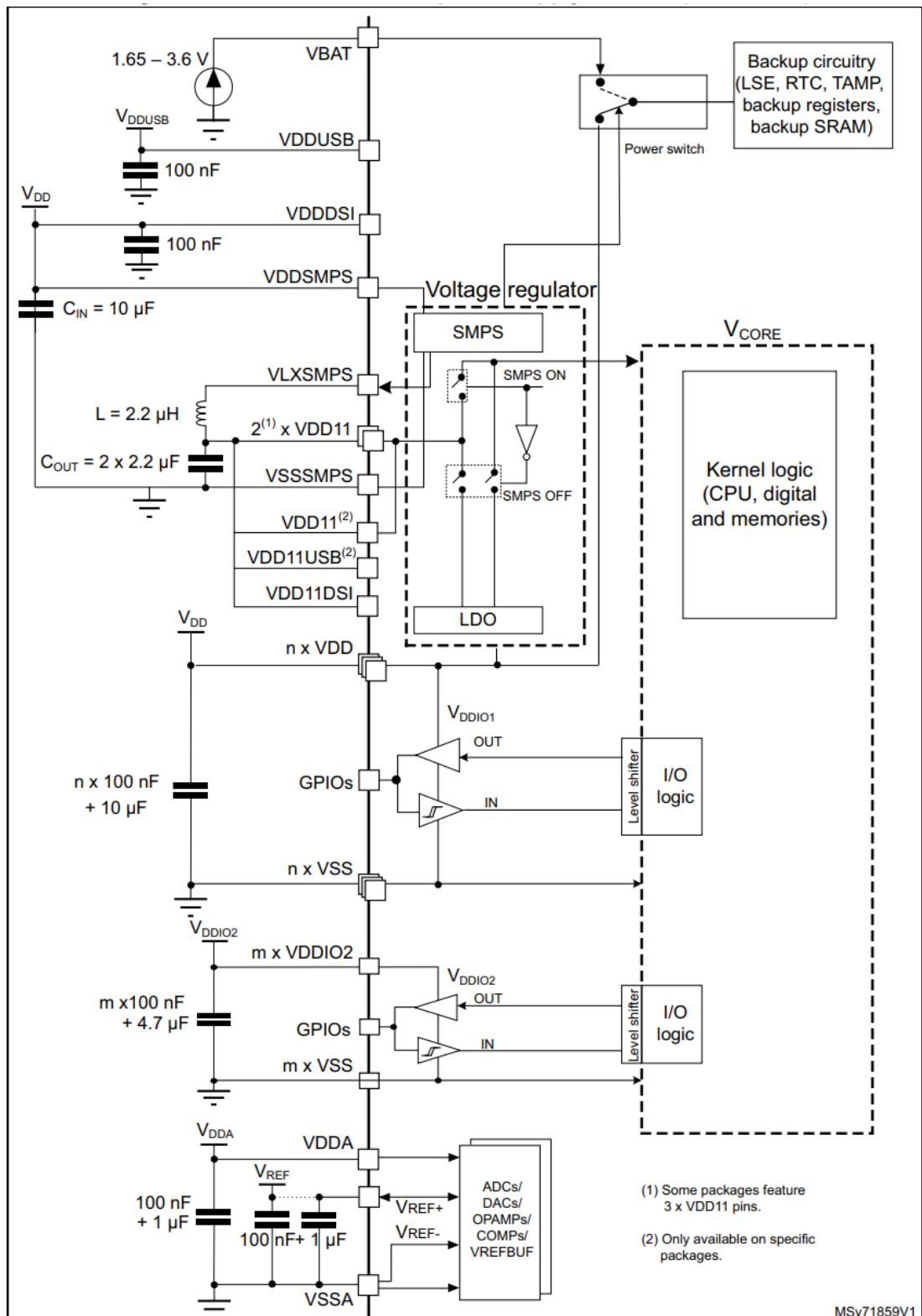


Figure 19: STM32U5GxxxxQ power supply scheme (with SMPS)

C.2 Supplementary Circuits**C.3 Pin Justification Tables**

Pin	Function	Reason	Requirements
1	SCROLL A	Close to where rotary encoder was placed	Requires an interrupt pin (all GPIOs on this processor can be set as interrupts)
2	OCTOSPI M P1 DQS	Auto assigned by STM32CubeIDE after selecting 'PORT1 DQS' Data Strobe	nil.
3	DCMI D4	Auto assigned by STM32CubeIDE after selecting 'Slave 12 bits External Synchro' on DCMI [1]	nil.
4	DCMI D6	1	nil.
5	DCMI D7	1	nil.
6	3V3	No external battery used	Tie to VCC if no external battery.
7	SCROLL B	Close to where rotary encoder was placed	Requires an interrupt pin (all GPIOs on this processor can be set as interrupts)
8	LSE IN	Input pin for low speed clock	32.768kHz
9	NC	Output of LSE	Leave NC if using a single output oscillator (see Table 3 of getting started guide)
10	I2C2 SDA	Auto assigned by STM32CubeIDE after selecting 'I2C2'	nil.
11	I2C2 SCL	Auto assigned by STM32CubeIDE after selecting 'I2C2'	nil.
12	NC	nil.	nil.
13	3V3	Same as VCC	Must be connected to same supply as VCC
14	VDDCORE	Same voltage as VDDCORE	Must be connected to same supply as VDD11
15	NC	nil.	nil.
16	NC	nil.	nil.
17	GND	VSS pin = GND	nil.
18	NC	nil.	nil.
19	NC	nil.	nil.
20	VDDCORE	Same voltage as VDDCORE	Must be connected to same supply as VDD11
21	NC	nil.	nil.
22	NC	nil.	nil.
23	GND	VSS pin = GND	nil.
24	GND	VSS pin = GND	nil.
25	3V3	Common power domain	1.71-3.6V
26	HSE IN	Input pin for HSE	4-50MHz
27	NC	Output of HSE	Leave NC if using a single output oscillator (see Table 3 of getting started guide)
28	NRST	nil.	Active low
29	PWR ACTION	Close to MOSFET	nil.
30	PWR STATE	Close to MOSFET	nil.

Table 12: Pin justification for STM32U5 (Pins 1-30).

Pin	Function	Reason	Requirements
31	CSA OUT	Close to Current Sense Amplifier	Must be analog GPIO
32	NC	nil.	nil.
33	GND	VSS pin = GND	nil.
34	1V8	Same supply as VDDA	Connect to VDDA
35	3V3	Common power domain	1.62-3.6V
36	NC	nil.	nil.
37	SPI1 CLK	Auto assigned by STM32CubeIDE after selecting 'Half-Duplex Master' on Mode	nil.
38	USART2 TX	Auto assigned by STM32CubeIDE after selecting 'Asynchronous' on Mode	nil.
39	USART2 RX	Auto assigned by STM32CubeIDE after selecting 'Asynchronous' on Mode	nil.
40	GND	VSS pin = GND	nil.
41	3V3	Common power domain	1.71-3.6V
42	DCMI HSYNC	1	nil.
43	NC	nil.	nil.
44	DCMI CLK	1	nil.
45	OCTOSPIM P1 IO2	Auto assigned by STM32CubeIDE after selecting 'PORT1 IO[3:0]' on Data [3:0] [2]	.
46	OCTOSPIM P1 IO1	Auto assigned by STM32CubeIDE after selecting 'PORT1 IO[3:0]' on Data [3:0] [3]	nil.
47	OCTOSPIM P1 IO0	3	nil.
48	NC	nil.	nil.
49	NC	nil.	nil.
50	SENSOR RST TP	Near available space for test points	GPIO
51	NC	nil.	nil.
52	NC	nil.	nil.
53	TIM1 CH1 (Buzzer)	Auto assigned by STM32CubeIDE after selecting 'PWM Generation CH1' on Channel1	nil.
54	GND	VSS pin = GND	nil.
55	3V3	Common power domain	1.71-3.6V
56	OCTOSPIM P1 CLK	Auto assigned by STM32CubeIDE after selecting 'Port1 CLK' on Clock	nil.
57	OCTOSPIM P1 NCS	Auto assigned by STM32CubeIDE after selecting 'Port1 NCS' on 'Chip Select'	nil.
58	NC	nil.	nil.
59	SENSOR LED 2	Near LED array	GPIO
60	SENSOR LED 1	Near LED array	GPIO
61	OCTOSPIM P1 IO3	Auto assigned by STM32CubeIDE after selecting 'PORT1 IO[3:0]' on Data [3:0] [2]	.
62	I2C4 SDA	Auto assigned by STM32CubeIDE after selecting 'I2C4'	nil.
63	I2C4 SCL	Auto assigned by STM32CubeIDE after selecting 'I2C4'	nil.
64	VDDCORE	Source of VDDCORE	nil.
65	3V3	Same source as VDD	Connect to VDD
66	GND	VSS pin = GND	nil.
67	VDDCORE	Same source as VLXSMPS	Connect to VLXSMPS
68	GND	VSS pin = GND	nil.
69	3V3	Common power domain	1.71-3.6V

Table 12: Pin justification for STM32U5 (continued, Pins 31-69).

Pin	Function	Reason	Requirements
70	NC	nil.	nil.
71	MCU LED 1	Near LED array	GPIO
71	MCU LED 2	Near LED array	GPIO
73	READ LED 1	Near LED array	GPIO
74	READ LED 2	Near LED array	GPIO
75	WRITE LED 1	Near LED array	GPIO
76	WRITE LED 2	Near LED array	GPIO
77	DEBUG LED	Near LED array	GPIO
78	NC	nil.	nil.
79	3V3 LED	Near LED array	GPIO
80	NC	nil.	nil.
81	81	Originally a HSPI pin, repurposed as a broken out GPIO	GPIO
82	GND	VSS pin = GND	nil.
83	3V3	Common power domain	1.71-3.6V
84	84	Originally a HSPI pin, repurposed as a broken out GPIO	GPIO
85	85	Originally a HSPI pin, repurposed as a broken out GPIO	GPIO
86	DCMI D3	1	nil.
87	NC	nil.	nil.
88	GND	VSS pin = GND	nil.
89	3V3	Common power domain	1.71-3.6V
90	NC	nil.	nil.
91	DCMI D11	1	nil.
92	NC	nil.	nil.
93	DCMI D8	1	nil.
94	GND	VSS pin = GND	nil.
95	3V3	Common power domain	1.71-3.6V
96	DCMI D9	1	nil.
97	97	Originally a HSPI pin, repurposed as a broken out GPIO	GPIO
98	DCMI D0	1	nil.
99	DCMI D1	1	nil.
100	SDMMC1 DAT0	Auto assigned after selecting 'SD 4 bits Wide with auto dir voltage converter' on Mode [5]	nil.
101	SDMMC1 DAT1	5	nil.
102	NC	nil.	nil.
103	USART1 TX	Auto assigned by STM32CubeIDE after selecting 'Asynchronous' on Mode	nil.
104	USART1 RX	Auto assigned by STM32CubeIDE after selecting 'Asynchronous' on Mode	nil.
105	NC	nil.	nil.
106	SPI1 MOSI	Auto assigned by STM32CubeIDE after selecting 'Half-Duplex Master' on Mode	nil.
107	SWDIO	Auto assigned by STM32CubeIDE after selecting 'Serial Wire' on Debug	nil.
108	3V3	USB not used as a transceiver	Must be connected to VDD if transceiver not used
109	3V3	Common power domain	1.71-3.6V
110	GND	VSS pin = GND	nil.

Table 12: Pin justification for STM32U5 (continued, Pins 70-110).

Pin	Function	Reason	Requirements
111	SWDCLK	Auto assigned by STM32CubeIDE after selecting 'Serial Wire' on Debug	nil.
112	NC	nil.	nil.
113	SDMMC1 DAT2	5	nil.
114	SDMMC1 DAT3	5	nil.
115	SDMMC1 CLK	5	nil.
116	NC	nil.	nil.
117	NC	nil.	nil.
118	SDMMC1 CMD	5	nil.
119	DCMI D5	1	nil.
120	NC	nil.	nil.
121	NC	nil.	nil.
122	3V3	Common power domain	1.71-3.6V
123	DMCI D10	1	nil.
124	NC	nil.	nil.
125	NC	nil.	nil.
126	NC	nil.	nil.
127	NC	nil.	nil.
128	I2C1 SDA	Auto assigned by STM32CubeIDE after selecting 'I2C1'	nil.
129	I2C1 SCL	Auto assigned by STM32CubeIDE after selecting 'I2C1'	nil.
130	GND	VSS pin = GND	nil.
131	3V3	PG[15:2] doesn't require a different power domain, I2C commonly runs on 3V3 logic	1.08-3.6V. Independent from VDD when PG[15:2] in use, tie to VDD when not in use
132	STANDBY	Close to sensor connector	GPIO
133	RESET	Close to sensor connector	GPIO
134	TRIGGER	Close to sensor connector	GPIO
135	NC	nil.	nil.
136	SHUTTER	close to sensor connector	GPIO
137	DCMI VSYNC	1	nil.
138	BOOT0	nil.	Active high
139	SDMMC1 CLKIN	5	nil.
140	NC	nil.	nil.
141	DCMI D2	1	nil.
142	VDDCORE	Same voltage as VDDCORE	Must be connected to same supply as VDD11
143	SCROLL BUTTON	Close to where rotary encoder was placed	Requires an interrupt pin (all GPIOs on this processor can be set as interrupts)
144	3V3	Common power domain	1.71-3.6V

Table 12: Pin justification for STM32U5 (continued, Pins 111-144).

C.4 PCB Layout

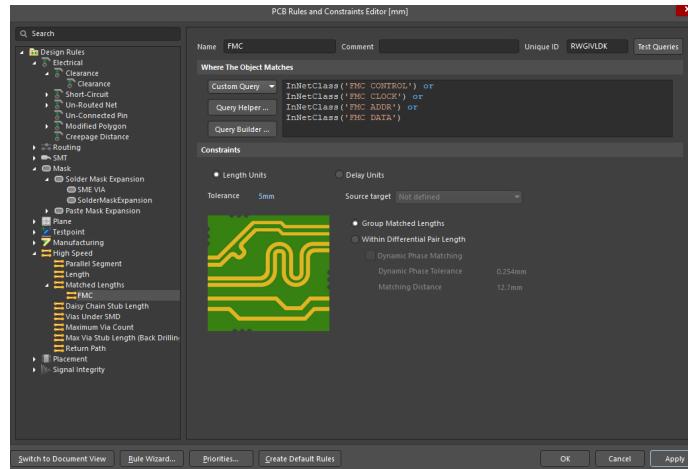


Figure 20: FMC length match design rule.

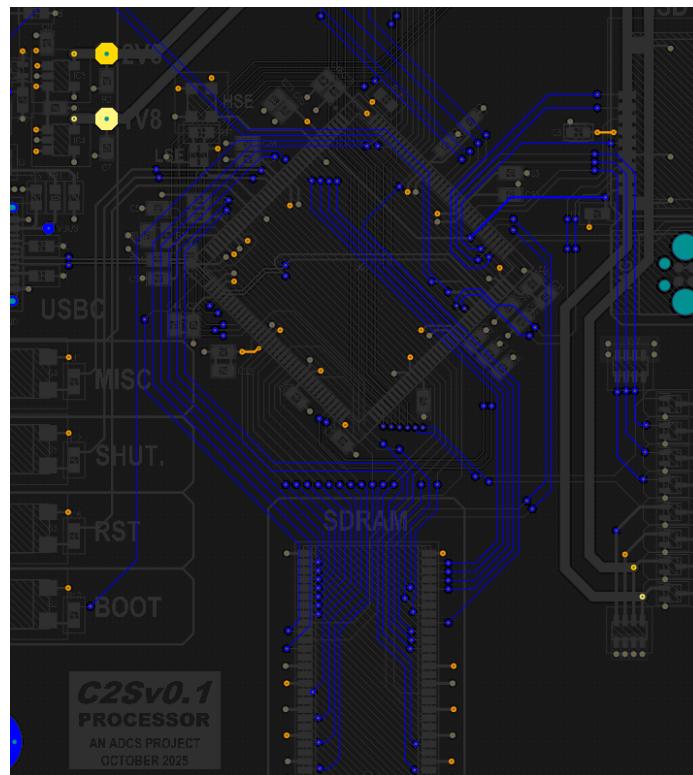


Figure 21: Parallel routes from C2Sv0.1 PCB.

D Datasheet Extracts and Ratings

D.1 Processors

D.1.1 STM32N6

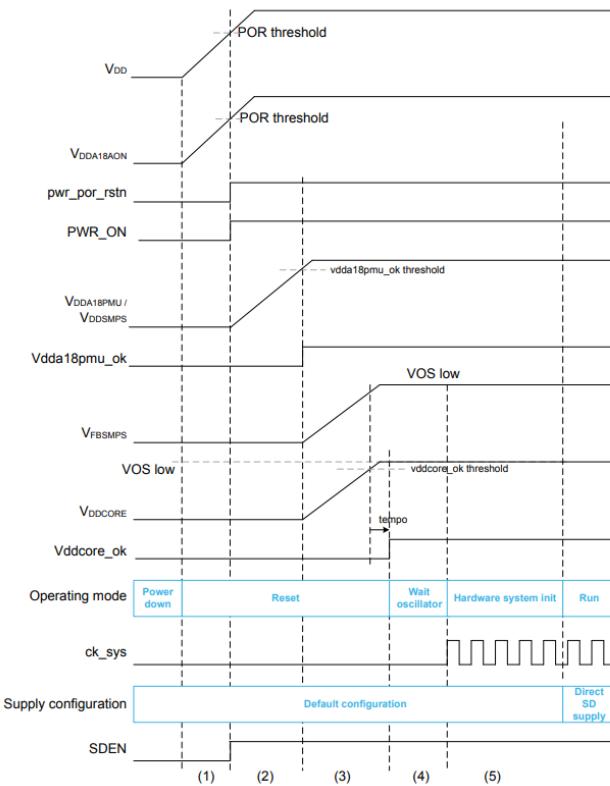


Figure 22: STM32N6 startup with VDDCORE supplied directly from internal SMPS step-down converter

D.1.2 STM32F7

Symbol	Ratings	Min	Max	Unit
$V_{DD}-V_{SS}$	External main supply voltage (including V_{DDA} , V_{DD} , V_{BAT} and V_{DDUSB}) ⁽¹⁾	- 0.3	4.0	
V_{IN}	Input voltage on FT pins ⁽²⁾	$V_{SS} - 0.3$	$V_{DD} + 4.0$	V
	Input voltage on TTa pins	$V_{SS} - 0.3$	4.0	
	Input voltage on any other pin	$V_{SS} - 0.3$	4.0	
	Input voltage on BOOT pin	V_{SS}	9.0	
$ \Delta V_{DDx} $	Variations between different V_{DD} power pins	-	50	mV
$ V_{SSx}-V_{SSL} $	Variations between all the different ground pins ⁽³⁾	-	50	
$V_{ESD(HBM)}$	Electrostatic discharge voltage (human body model)	see Section 6.3.15: Absolute maximum ratings (electrical sensitivity)		-

1. All main power (V_{DD} , V_{DDA} , V_{DDUSB}) and ground (V_{SS} , V_{SSA}) pins must always be connected to the external power supply, in the permitted range.

2. V_{IN} maximum value must always be respected. Refer to [Table 14](#) for the values of the maximum allowed injected current.

3. Include VREF- pin.

(a) Voltage characteristics.

Symbol	Ratings	Max.	Unit
ΣI_{VDD}	Total current into sum of all V_{DD_x} power lines (source) ⁽¹⁾	320	
ΣI_{VSS}	Total current out of sum of all V_{SS_x} ground lines (sink) ⁽¹⁾	- 320	
ΣI_{VDDUSB}	Total current into V_{DDUSB} power line (source)	25	
I_{VDD}	Maximum current into each V_{DD_x} power line (source) ⁽¹⁾	100	
I_{VSS}	Maximum current out of each V_{SS_x} ground line (sink) ⁽¹⁾	- 100	
I_{IO}	Output current sunk by any I/O and control pin	25	mA
	Output current sourced by any I/Os and control pin	- 25	
ΣI_{IO}	Total output current sunk by sum of all I/O and control pins ⁽²⁾	120	
	Total output current sunk by sum of all USB I/Os	25	
	Total output current sourced by sum of all I/Os and control pins ⁽²⁾	- 120	
$I_{INJ(PIN)}$	Injected current on FT, FTf, RST and B pins ⁽³⁾	- 5/+0	
	Injected current on TTa pins ⁽⁴⁾	± 5	
$\Sigma I_{INJ(PIN)}$ ⁽⁴⁾	Total injected current (sum of all I/O and control pins) ⁽⁵⁾	± 25	

1. All main power (V_{DD} , V_{DDA}) and ground (V_{SS} , V_{SSA}) pins must always be connected to the external power supply, in the permitted range.

2. This current consumption must be correctly distributed over all I/Os and control pins. The total output current must not be sunk/sourced between two consecutive power supply pins referring to high pin count LQFP packages.

3. Positive injection is not possible on these I/Os and does not occur for input voltages lower than the specified maximum value.

4. A positive injection is induced by $V_{IN} > V_{DDA}$ while a negative injection is induced by $V_{IN} < V_{SS}$. $I_{INJ(PIN)}$ must never be exceeded. Refer to [Table 13: Voltage characteristics](#) for the values of the maximum allowed input voltage.

5. When several inputs are submitted to a current injection, the maximum $\Sigma I_{INJ(PIN)}$ is the absolute sum of the positive and negative injected currents (instantaneous values).

(b) Current characteristics.

Symbol	Ratings	Value	Unit
T_{STG}	Storage temperature range	- 65 to +150	°C
T_J	Maximum junction temperature	125	

(c) Thermal characteristics.

Figure 23: Absolute maximum ratings from STM32F750Z8T6 datasheet, Table 26. Please note that these are snippets, so any hyperlinks or references are invalid.

Port		AF0	AF1	AF2	AF3	AF4	AF5	AF6	AF7	AF8	AF9	AF10	AF11	AF12	AF13	AF14	AF15
		SYS	TIM1/2	TIM3/4/5	TIM8/9/10/ 11/LPTIM 1/CEC	I2C1/2/3/ 4/CEC	SPI1/2/3/ 4/5/6	SPI3/ SAI1	SPI2/3/U ART6/UA RT4/5/7/8 /SPDIFRX	SAI2/US ART6/UA RT4/5/7/8 /SPDIFRX	CAN1/2/T IM12/13/ 14/QUAD SPI/LCD	SAI2/QU ADSPPIO TG2_HS/ OTG1_FS	ETH/ OTG1_FS	FMC/SD MMC1/O TG2_FS	DCMI	LCD	SYS
Port C	PC4	-	-	-	-	-	I2S1_M CK	-	-	SPDIFRX _IN2	-	-	ETH_MII RXD0/ET H_RMIIL RXD0	FMC_SD NE0	-	-	EVEN TOUT
	PC5	-	-	-	-	-	-	-	-	SPDIFRX _IN3	-	-	ETH_MII RXD1/ET H_RMIIL RXD1	FMC_SD CKE0	-	-	EVEN TOUT
	PC6	-	-	TIM3_C H1	TIM8_CH 1	-	I2S2_M CK	-	-	USART6 _TX	-	-	-	SDMMC 1_D6	DCMI_D 0	LCD_HS YNC	EVEN TOUT
	PC7	-	-	TIM3_C H2	TIM8_CH 2	-	-	I2S3_M CK	-	USART6 _RX	-	-	-	SDMMC 1_D7	DCMI_D 1	LCD_G6	EVEN TOUT
	PC8	TRACE D1	-	TIM3_C H3	TIM8_CH 3	-	-	-	UART5_ RTS	USART6 _CK	-	-	-	SDMMC 1_D0	DCMI_D 2	-	EVEN TOUT
	PC9	MCO2	-	TIM3_C H4	TIM8_ CH4	I2C3_SD A	I2S_CK N	-	UART5_ CTS	-	QUADSP _BK1_IO 0	-	-	SDMMC 1_D1	DCMI_D 3	-	EVEN TOUT
	PC10	-	-	-	-	-	-	SPI3_SC K12S3_ CK	USART3 _TX	UART4_T X	QUADSP _BK1_IO 1	-	-	SDMMC 1_D2	DCMI_D 8	LCD_R2	EVEN TOUT
	PC11	-	-	-	-	-	-	SPI3_MI SO	USART3 _RX	UART4_R X	QUADSP _BK2_N CS	-	-	SDMMC 1_D3	DCMI_D 4	-	EVEN TOUT
	PC12	TRACE D3	-	-	-	-	-	SPI3_M OSI/12S3 _SD	USART3 _CK	UART5_T X	-	-	-	SDMMC 1_CK	DCMI_D 9	-	EVEN TOUT
	PC13	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	EVEN TOUT
	PC14	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	EVEN TOUT
	PC15	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	EVEN TOUT

Figure 24: Example of overlap between SDMMC1 and DCMI pins from STM32F750Z8T6 datasheet (Table 12, page 71).

D.1.3 STM32U5

D.2 Image Sensors

D.2.1 AR0141 Sensor

Symbol	Definition	Condition	Min	Max	Unit
VDD_MAX	Core digital voltage		-0.3	2.4	V
VDD_IO_MAX	I/O digital voltage		-0.3	4	V
VAA_MAX	Analog voltage		-0.3	4	V
VAA_PIX	Pixel supply voltage		-0.3	4	V
VDD_PLL	PLL supply voltage		-0.3	4	V
VDD_SLVS_MAX	HiSPi I/O digital voltage		-0.3	2.4	V
t _{ST}	Storage temperature		-40	150	°C

Note: Exposure to absolute maximum rating conditions for extended periods may affect reliability.

Figure 25: Absolute maximum ratings of AR0141CS2M00SUEA0-DPBR found on the component datasheet, Table 26

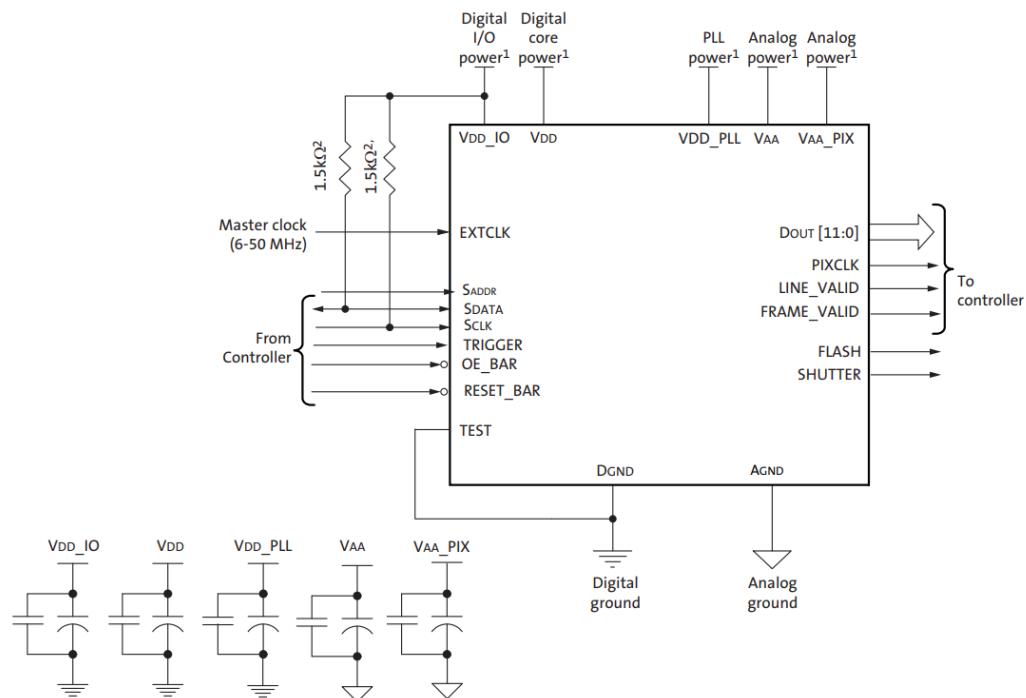


Figure 26: Typical configuration for parallel pixel data interface on AR0141CS2M00SUEA0-DPBR.

D.3 Memory

D.3.1 IS42S16160J Memory

ABSOLUTE MAXIMUM RATINGS⁽¹⁾

Symbol	Parameters	Rating	Unit
V _{DD} MAX	Maximum Supply Voltage	-0.5 to +4.6	V
V _{DDQ} MAX	Maximum Supply Voltage for Output Buffer	-0.5 to +4.6	V
V _{IN}	Input Voltage	-0.5 to V _{DD} + 0.5	V
V _{OUT}	Output Voltage	-1.0 to V _{DDQ} + 0.5	V
P _D MAX	Allowable Power Dissipation	1	W
I _{CS}	Output Shorted Current	50	mA
T _{OPR}	Operating Temperature	Com. Ind. A1 A2	°C
T _{STG}	Storage Temperature	-65 to +150	°C

Notes:

1. Stress greater than those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect reliability.
2. All voltages are referenced to V_{SS}.

Figure 27: Absolute maximum ratings for IS42S16160J-6TL found on page 16 of the datasheet.

D.4 Power Components

D.4.1 TCR2EF LDO

Characteristics	Symbol	Rating		Unit
Input voltage	V_{IN}	6.0		V
Control voltage	V_{CT}	-0.3 to 6.0		V
Output voltage	V_{OUT}	-0.3 to $V_{IN} + 0.3$		V
Power dissipation	P_D	SMV	200	(Note 1)
			580	(Note 2)
		ESV	150	(Note 1)
			320	(Note 3)
Junction temperature	T_j	150		°C
Storage temperature range	T_{STG}	-55 to 150		°C

Figure 28: Absolute maximum ratings of TCR2EF series (SMV), found on the component datasheet, page 2.

List of Figures

1	Expected image coverage and resolution for the MIRA220 sensor, binned once with a focal length of 8.9mm.	6
2	Draft PCBs of the C2S Breakout Board.	9
3	Simple power tree diagram of C2Sv0.1	22
4	Power pad used on QSPI memory chip.	24
5	Example of bad spacing on DCMI traces in an early version of C2Sv0.1.	25
6	Mounting pattern for AERO3001 Spacecraft Project Simple Shaker Bracket	26
7	Measurements in Altium Designer confirming that the sensor is centered relative to the component origin. If the middle four balls are centered, all others must be since they are evenly spaced, therefore the sensor is centered.	27
8	Measurements in Altium Designer confirming the measurements to these specifications.	27
9	Knob modelled in Fusion360	29
10	Birds eye view of the C2Sv0.1 sensor board after initial manufacturing.	30
11	First C2Sv0.1 processor board after manufacturing with the large buttons still included.	31
12	Set-Reset (S-R) latch from C2Sv0.1	35
13	Resistor in series with a voltage source VIN with output voltage VOUT.	39
14	Overview of the LED power estimation: (a) the circuit used, (b) measured results.	39
15	BGA fanout methods used in the C2S Breakout Board PCB.	40
16	MIPI CSI-2 differential pairs on the C2S Breakout Board.	40
17	Debug LEDs on C2S Breakout Board.	40
18	Decoupling capacitors for STM32U5G	42
19	STM32U5GxxxxQ power supply scheme (with SMPS)	43
20	FMC length match design rule.	48
21	Parallel routes from C2Sv0.1 PCB.	48
22	STM32N6 startup with VDDCORE supplied directly from internal SMPS step-down converter	49
23	Absolute maximum ratings from STM32F750Z8T6 datasheet, Table 26. Please note that these are snippets, so any hyperlinks or references are invalid.	50
24	Example of overlap between SDMMC1 and DCMI pins from STM32F750Z8T6 datasheet (Table 12, page 71).	51
25	Absolute maximum ratings of AR0141CS2M00SUEA0-DPBR found on the component datasheet, Table 26	51
26	Typical configuration for parallel pixel data interface on AR0141CS2M00SUEA0-DPBR.	52
27	Absolute maximum ratings for IS42S16160J-6TL found on page 16 of the datasheet.	52
28	Absolute maximum ratings of TCR2EF series (SMV), found on the component datasheet, page 2.	53

List of Tables

1	Milestone objectives for the duration of the project	2
2	Breakdown of project responsibilities	2
3	Project schedule and key tasks	3
4	C2S system requirements as of August '25, Phase 1.	4
5	Comparison of STM32N6, STM32H7, and STM32F4 Microcontroller Families	7
6	Breakdown of STM32N657I0H3Q Part Number	7
7	Current carrying capacity estimation of BGA traces in C2S Breakout Board.	14
8	Cost breakdown of C2Sv0.1.	17
9	AR0141 DCMI pin naming convention	19
10	Comparison of STM32F4, F7, H7, and U5G (144-pin) for Image Processing Applications	19
11	Explanation of truncated silkscreen labels	26
12	Pin justification for STM32U5 (Pins 1-30).	44
12	Pin justification for STM32U5 (continued, Pins 31-69).	45
12	Pin justification for STM32U5 (continued, Pins 70-110).	46
12	Pin justification for STM32U5 (continued, Pins 111-144).	47