Design Concepts and Overview

CubeSat Camera System (C2S)

Felix Abbott — Perth Aerospace Student Team

ADCS Department — CubeSat 1 Mission

July 2025

Contents

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 SD 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^2C/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^2C/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 & 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, we 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 section 3

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 we 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 we 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 the our image are the file size and CubeSat size. We cannot be transmitting 2.2MP images down to Earth, so we must bin down or crop the image. If we bin down by n times, then the pixel size and focal length increases by n^2. If we place a hard limit of 1MP on the pixel count, then we only need a focal length of $<$10mm. If the sensor is binned once (2.224MP$\rightarrow$0.5MP, 2.79μm$\rightarrow$11.16μm) 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. We 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, we may adjust the scope towards the 750m spectral resolution range to comfortably sit within size contraints (depending on the results of testing).

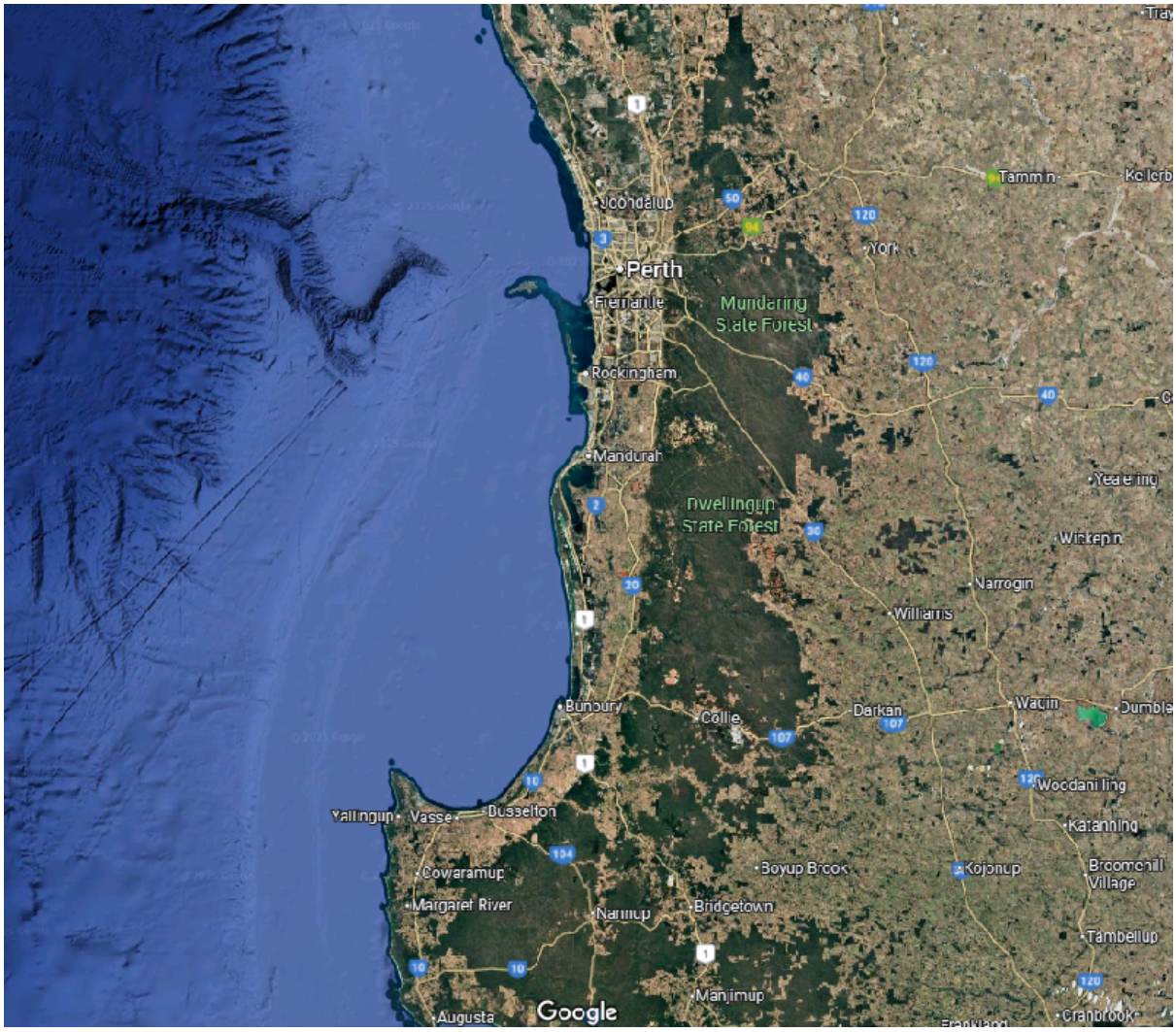


Figure 1: Expected image coverage and resolution for the MIRA220 sensor, binned once with a focal length of 8.9mm.

2.2.4 Image Processor

A lot of consideration was placed into the selection of the image processor. Initially, we investigated ASICs suited for image processing. We 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 senors 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 ameteur aerospace applications. Unlike FPGAs, which can compute parallel processing, STM32s use high-level languages like 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, we 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

We selected the STM32N657I0H3Q as the image processor for the breakout board. We 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, we 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℃ to +85℃ 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 SDRAM like the IS45S16320D-7TLA2.

2.2.5 SD Card Reader

Since we need a short and effective method of checking the files in a remote environment without tethered access (USB connection), we opted for saving the files on an SD 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 SD card directly, we chose the 254 Adafruit breakout board because it can be easily connected via standard PCB headers. Although it appears this step has been shortcut, the prototype will not use an SD card reader so it suffices as an effective solution without having to deal with the techincal aspects. As the prototype is expected to offload images to another device, this breakout-based approach 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.

2.3 Block Diagram

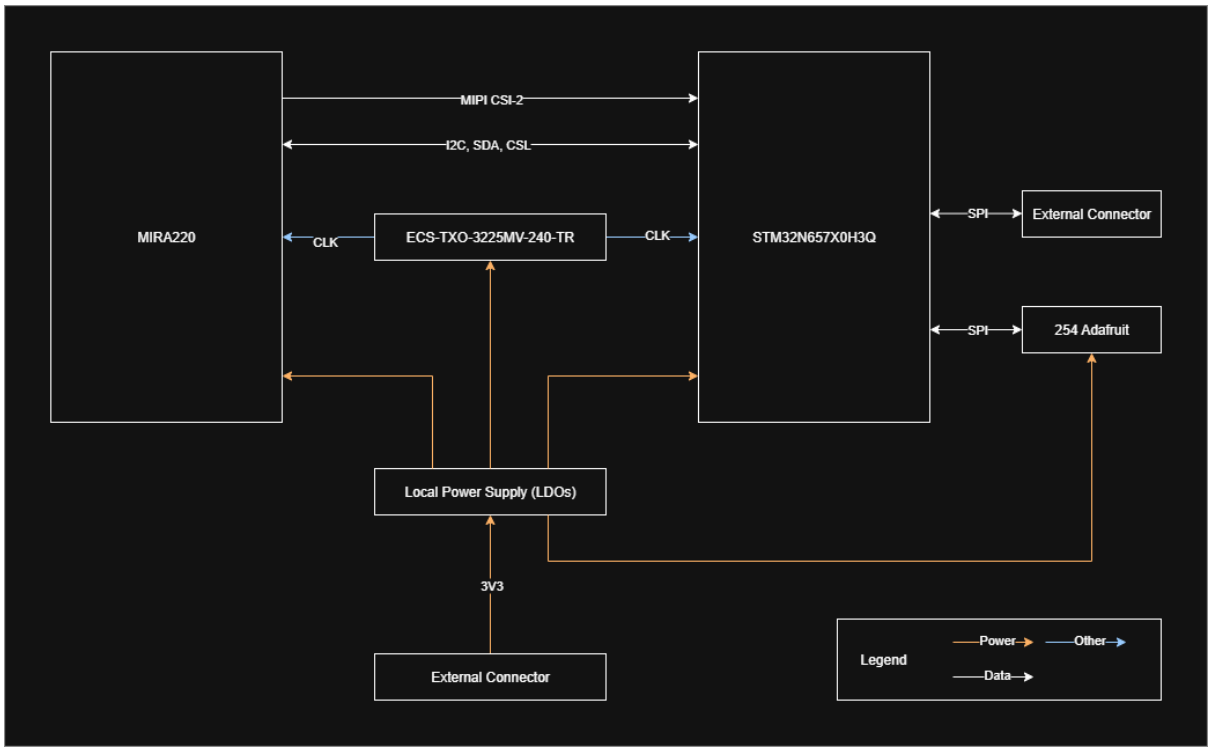


Figure 2: General block diagram of C2S breakout board. Components like resistors, capacitors and suggested circuits have been omitted from the diagram. This is meant to show the core functionality between the core components.

3 Schematic Design & Component Selection

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. Clock
4. SD Card
5. LDOs
6. SWD Connector
7. USBC Connector
8. Power Sequencer

Only critical paths were included on the Top-Level schematic. Routes like general power and ground were not included.

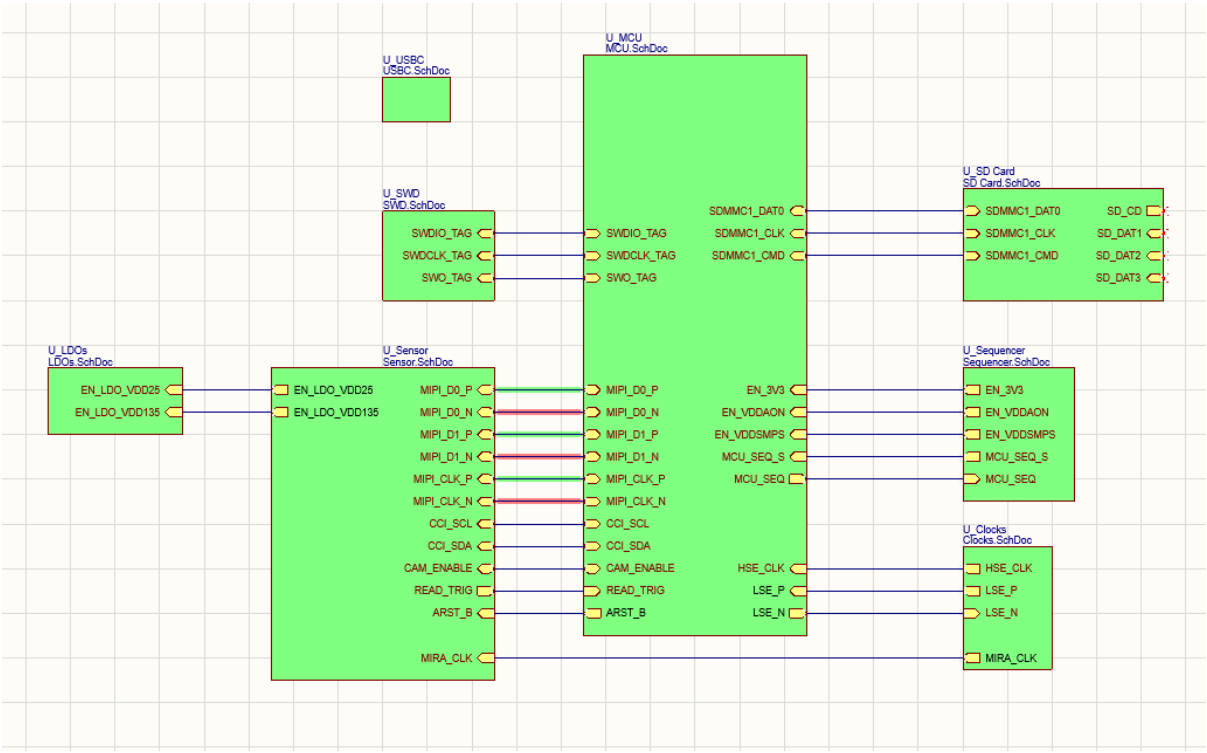


Figure 3: Top-Level schematic sheet from C2S Breakout Board Project.

3.1 Processor

.

The STM datasheet and getting started handbook were used to identify pins and their purpose. For example, pg. 7 of the getting started guide details the decoupling power figures required on most of the supply pins. For pins that were not explicitly described in the guide, we used the datasheet to find the pins.

The schematic symbol was divided in two to make it easier to route (opposed to the BGA order it was previously in: A1, A2, ..., B1, B2, ..., which made it difficult to route).

Additionally, debug LEDs were added to the schematic to make it easier to know which lines are active and which aren't. If we test one of the power lines once and it provides us with the correct voltage, then whenever the corresponding LED is active, we know it has the correct voltage. Similarly, an LED was added for reading from the sensor and writing to the SD card.

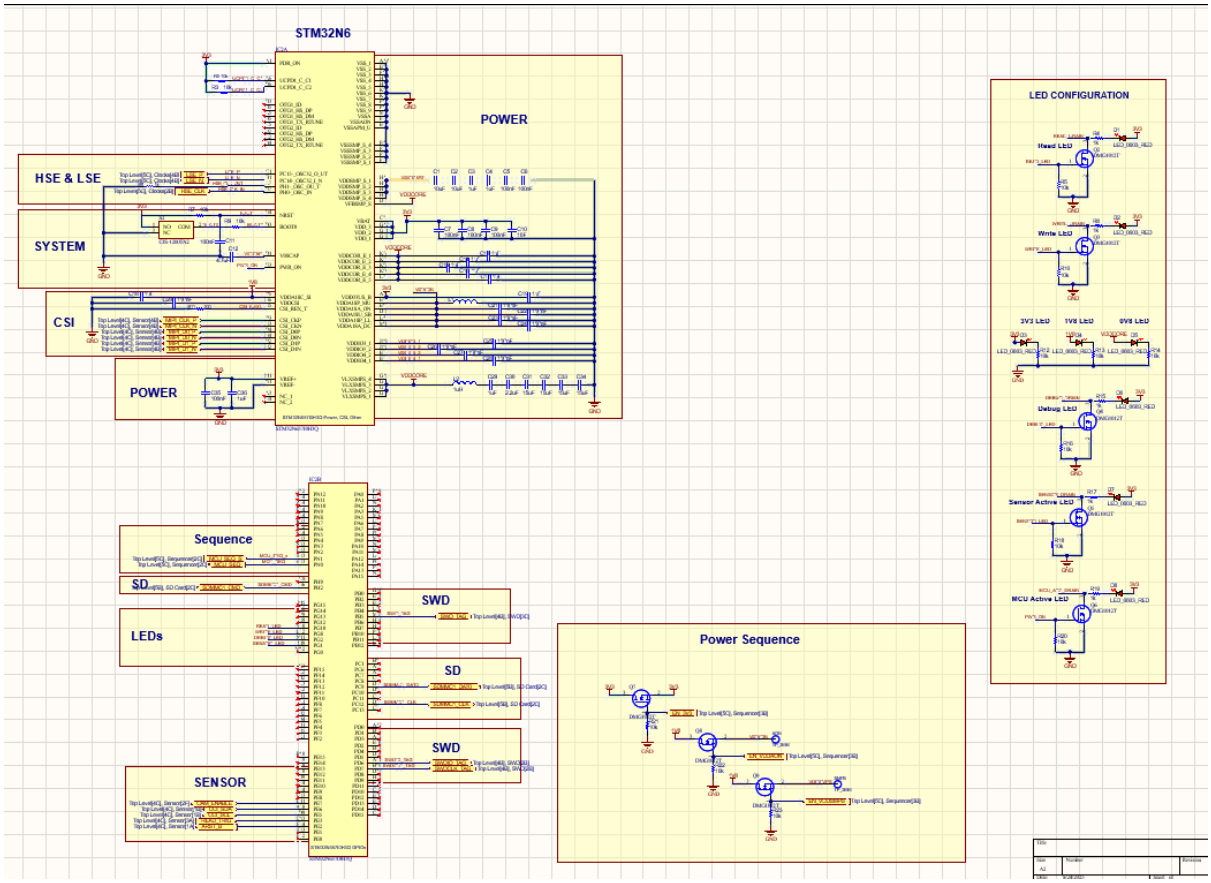


Figure 4: MCU schematic sheet from C2S Breakout Board Project.

3.2 Image Sensor

Most of the schematic was copied from the MIRA220 image sensor reference project. The schematic was not a copy-paste however, minor modifications were required to make it compatible with the STM32. The original project had the sensor itself as a breakout with connectors to an external processor. These connectors needed to be re-routed to the appropriate STM pins. The illumination trigger functionality was removed from the sensor since the project scope did not require this.

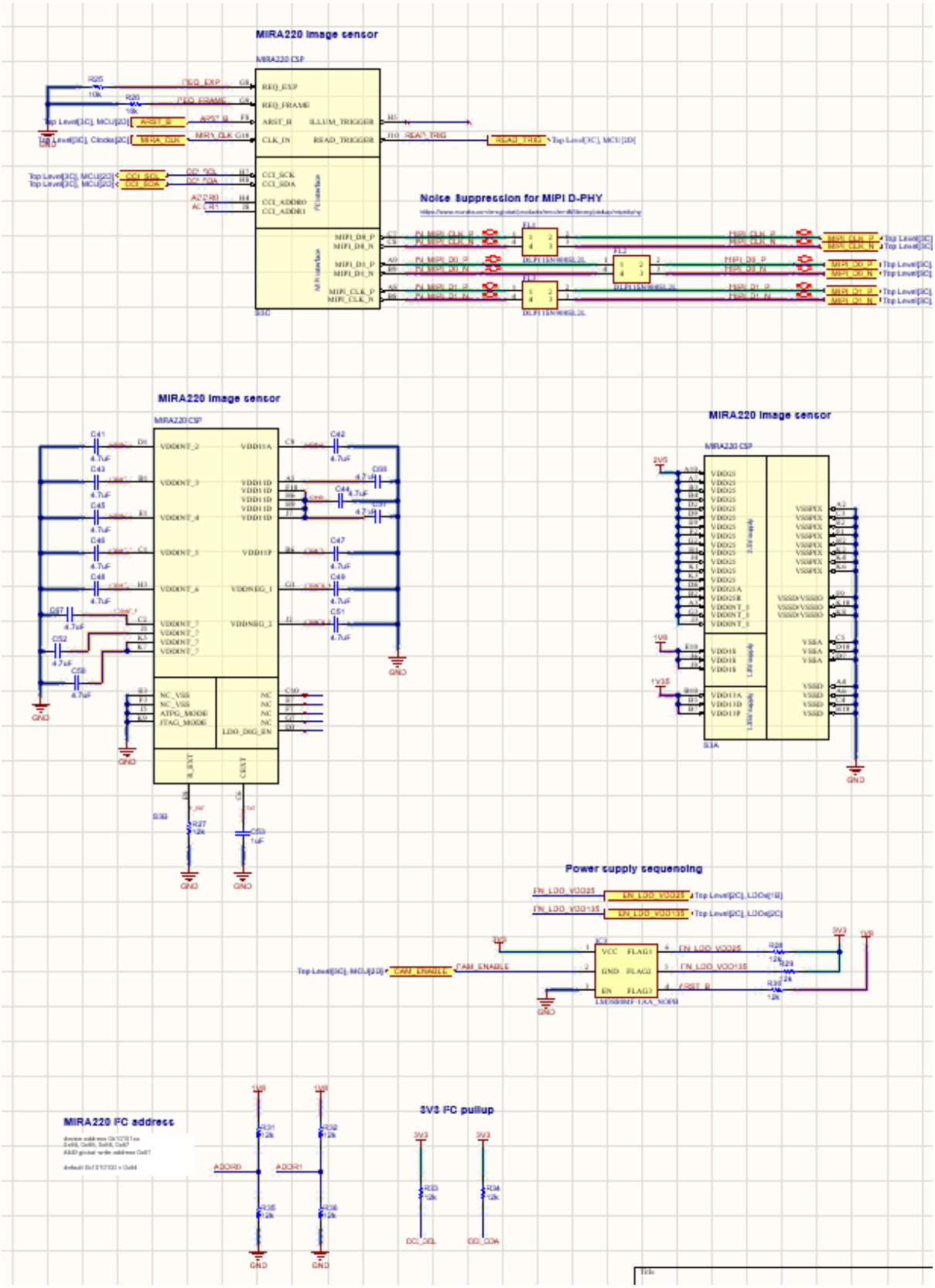


Figure 5: Sensor schematic sheet from C2S Breakout Board Project.

3.3 Clock

The C2S Breakout Board required three individual clocks for operation:

- High-speed 24MHz (HSE) clock for the STM32.
- Low-speed 32.768KHz (LSE) clock for the STM32.
- Hish speed 38.4MHZ external clock for the MIRA220 image sensor.

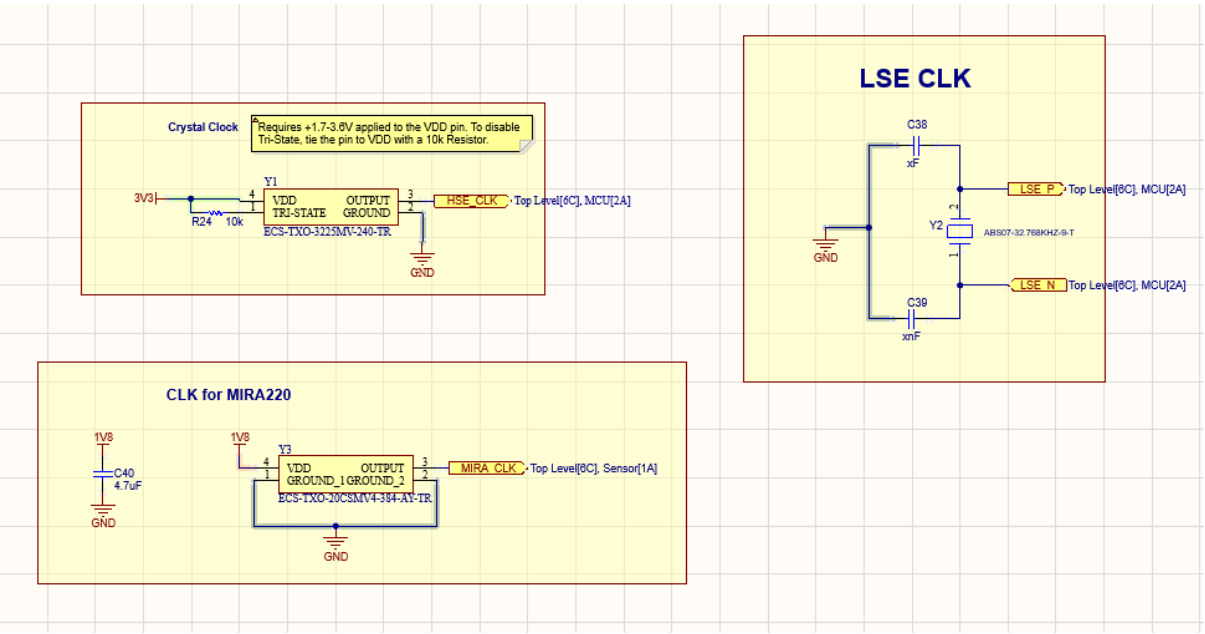


Figure 6: External clocks schematic sheet from C2S Breakout Board Project.

3.4 SD Card

Like the USB-C connector, the SD Card schematic was referenced from the ADCS Integration Board. SDMMC1 is a Secure Digital MultiMediaCard interface on certain microcontrollers, including STM32 series. It provides a dedicated hardware peripheral to communicate with SD cards, eMMC devices, or MMC cards. SDMMC1 implements a native SD protocol in hardware, allowing higher speeds (e.g., up to 50 MHz in SDIO mode vs 25 MHz in SPI). For faster readout speeds, we have connected the component in 4-bit mode, opposed to the simpler, but slower, 1-bit mode (connecting only DAT0).

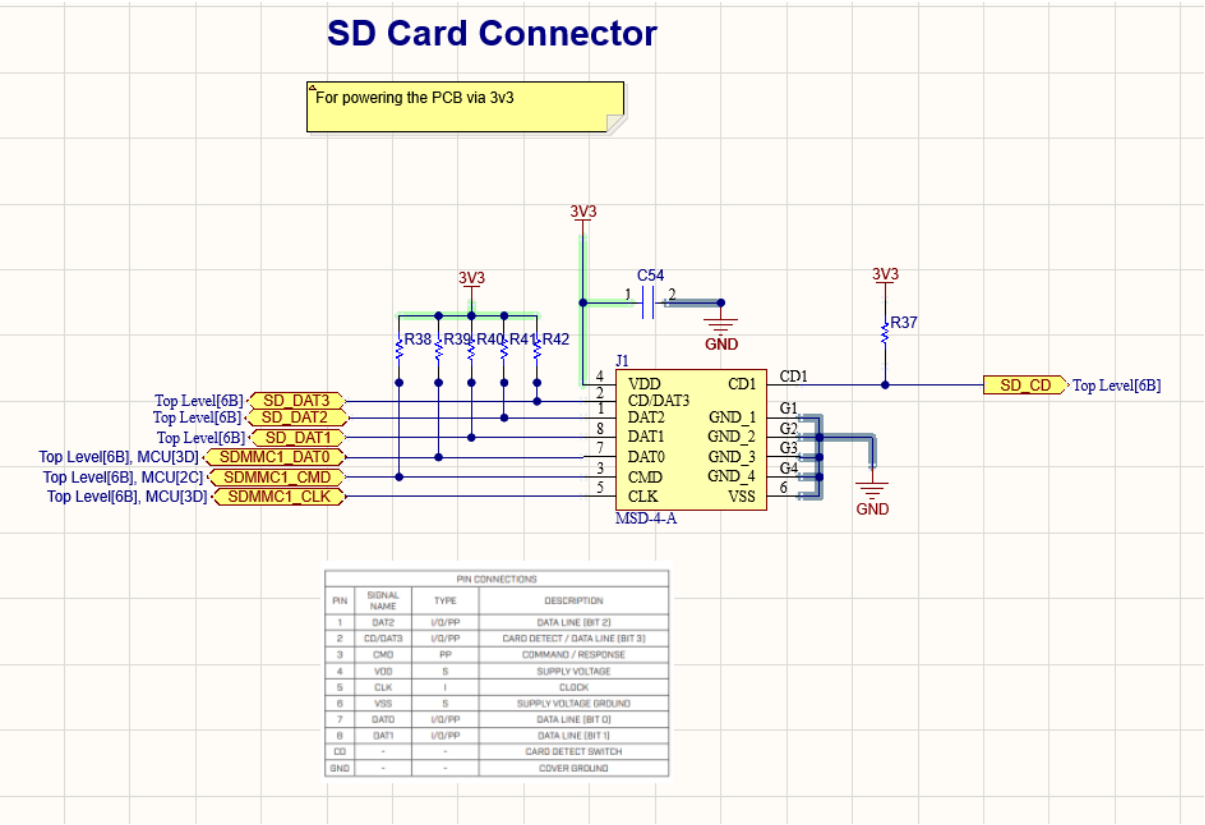


Figure 7: SD Card schematic sheet from C2S Breakout Board Project.

3.5 LDOs

The LDO schematic was mostly copied from the MIRA220 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. The decoupling capacitor values were not calculated and were copied over. Test points were added to the outputs of the LDOs as well as on the enable lines.

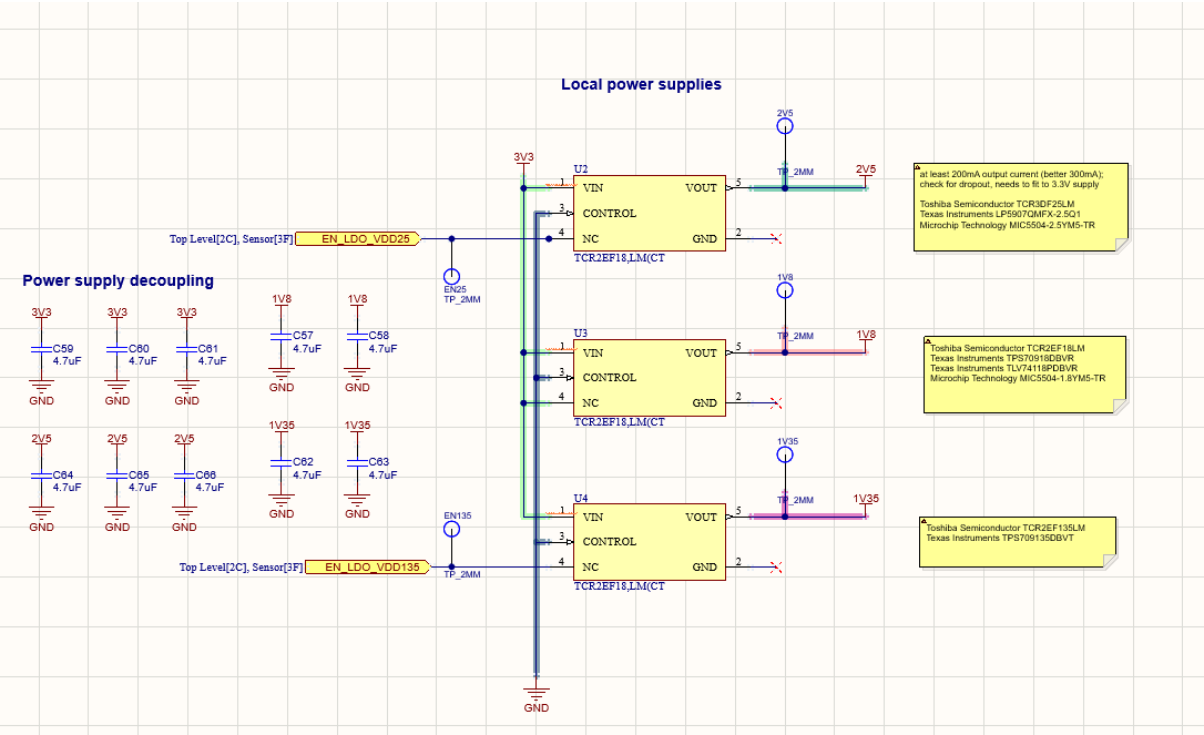


Figure 8: LDO schematic sheet from C2S Breakout Board Project.

3.6 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.

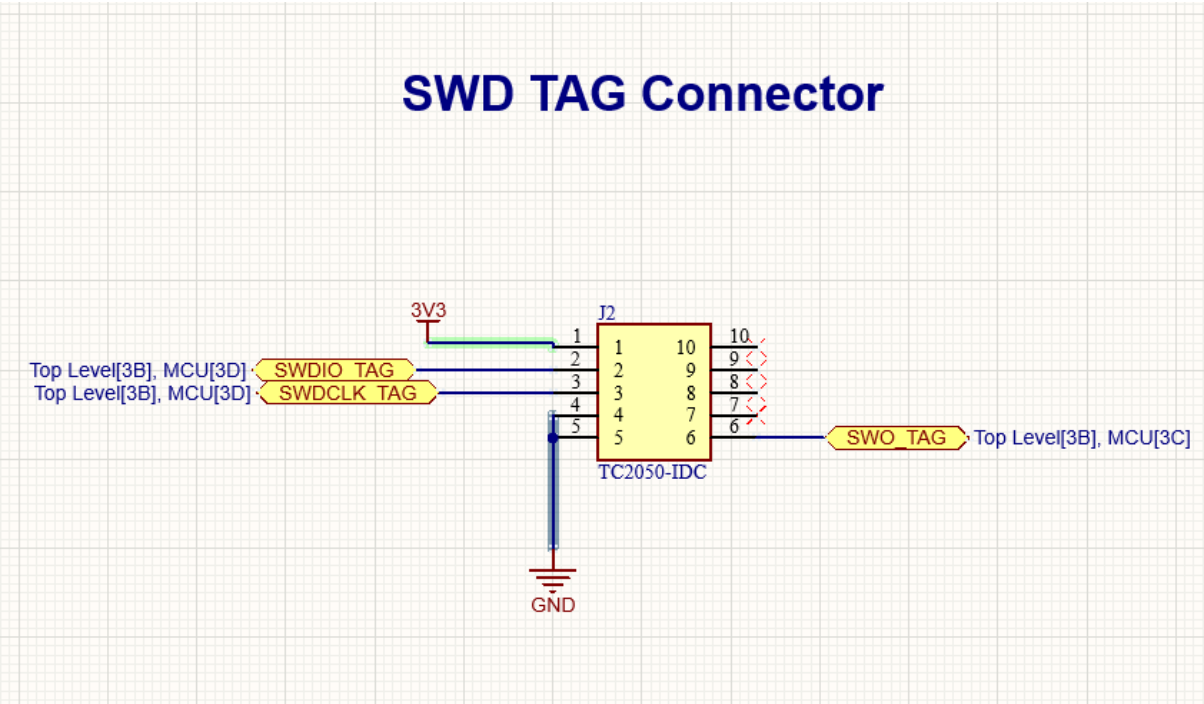


Figure 9: SWD TAG schematic sheet from C2S Breakout Board Project.

3.7 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, we 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. However, unlike the Integration Board, on the power pin (A4) was connected. The other 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 supressor 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. For a resonant frequency cut-off of 100kHz with an inductor value of $L = 190\text{nH}$, we chose capacitor values of $C_1 = 80\mu\text{F}$ and $C_2 = 157\mu\text{F}$ (subsection A.1) to meet the required $C_{eq} = 53\mu\text{F}$.

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, we introduced an LDO as a cost effective and simple method to maintain a +3V3 potential. If the powerbank supplies +3V3, then there is no to little power losses. If supply is higher, like +5V, the potential will be regulated down to +3V3. By connecting the VIN route to the VIN and CONTROL pins of the LDO, the component will turn and and begin regulating the signal whenever it is present - i.e., turns on only when it needs to.

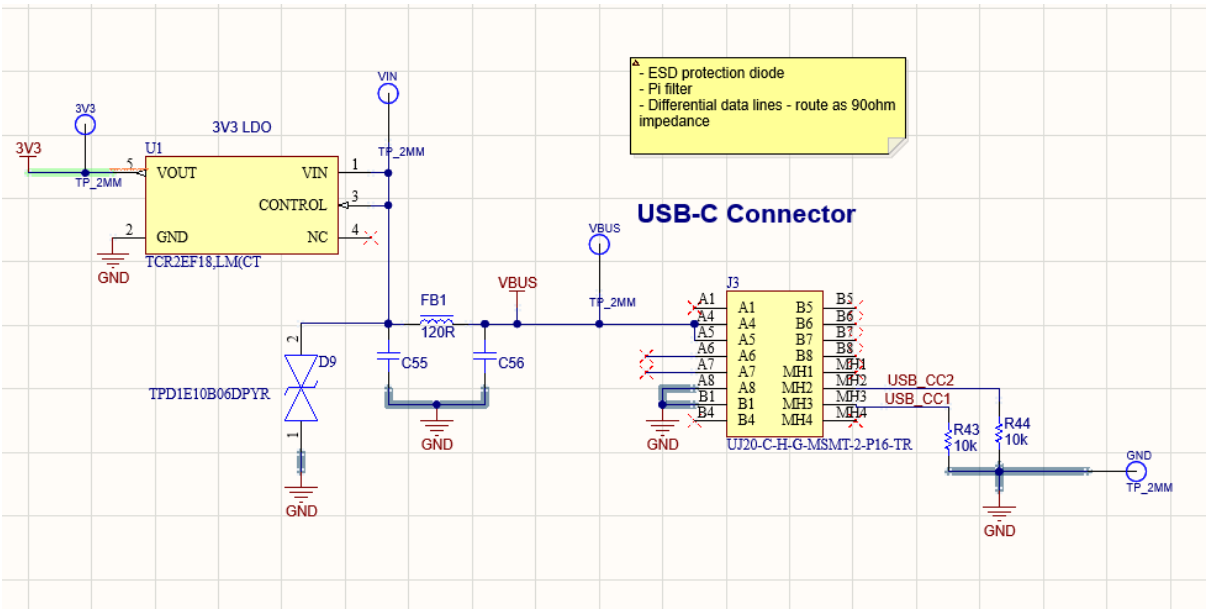


Figure 10: USB-C Connector schematic sheet from C2S Breakout Board Project.

3.8 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 (Figure 12), the component sequentially enables the FLAG1, FLAG2, FLAG3 pins. These enable pins are connected to the base of MOSFETs (see MCU schematic, Figure 4), which allow the sequencer to turn the power lines on and off in the required pattern.

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.

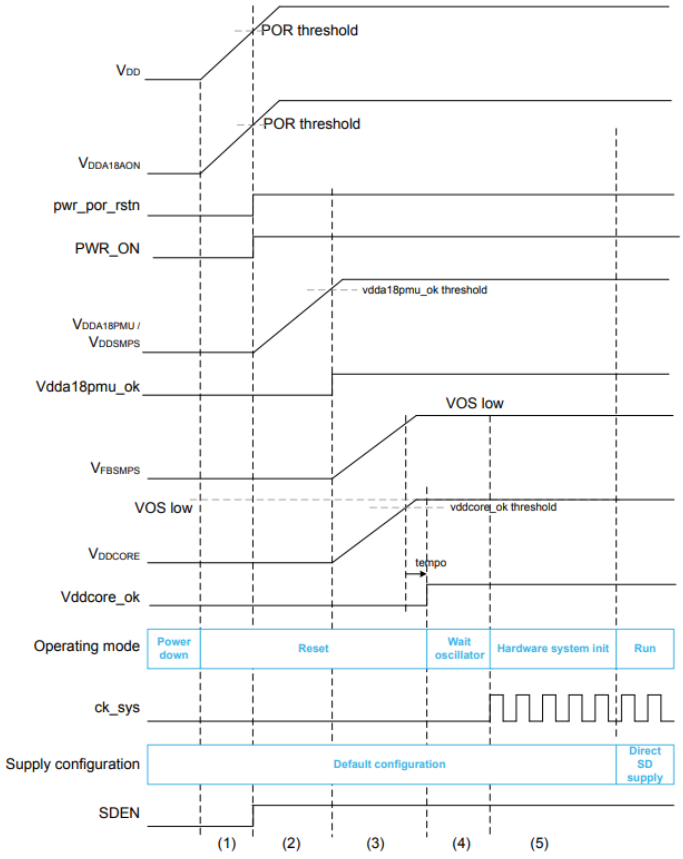


Figure 11: STM32N6 startup with VDDCORE supplied directly from internal SMPS step-down converter

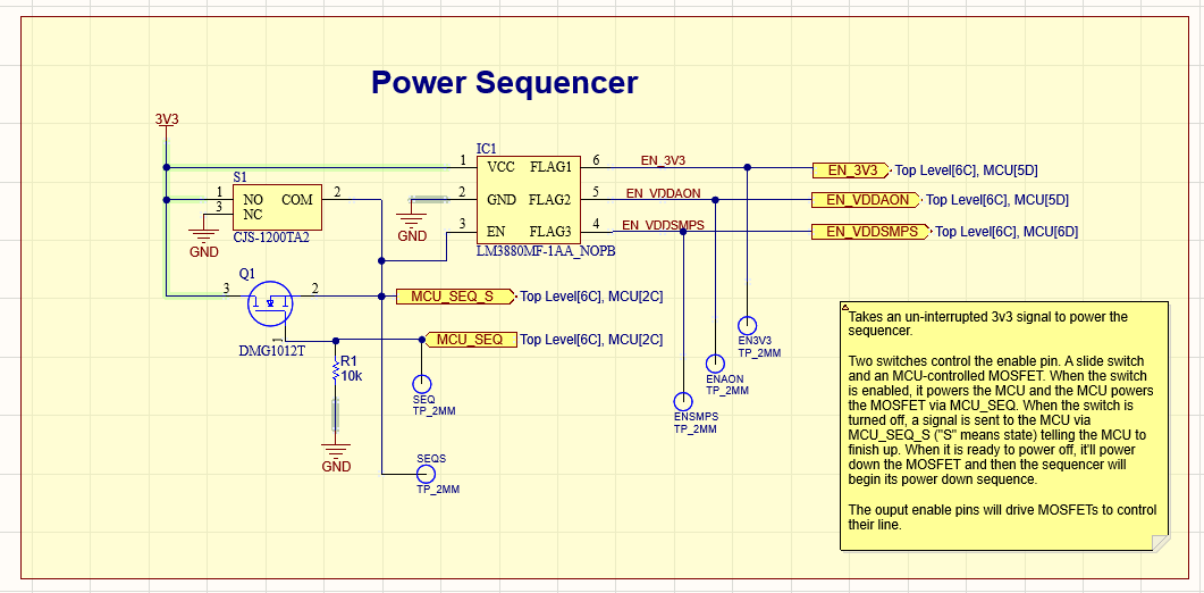


Figure 12: Power Sequencer schematic sheet from C2S Breakout Board Project.

4 PCB Layout and Review

4.1 Selecting Manufacturers

Because of the small BGA package, our traditional PCB manufacturer, JLCPCB, was no longer a viable option since their capabilities did not permit the MIRA220's ball diameter. Other manufacturers like PCBWAY are capable of producing the board, however, they have a high price point (PCBWAY is $350AUD+). Naturally, small quantities at a small scale governs an increased cost, so quotes of $100AUD+ were not unusual. We eventually came across NextPCB, a Chinese manufacturer that can create the boards at 3.5mil trace width and clearance for $110AUD. After consulting on reddit, we investigated potential methods to being the cost of the PCB down.

Again, because of the BGA package size, we concluded that it was not feasible to solder those two components by hand. To place the BGAs, we consulted with Peter from ATMWA to discuss their capabilities and whether they can place the two components or not. We found that:

- They are capable of placing BGAs of 0.2mm diameter and 0.5mm pitch.
- We should aim for a solder mask dam of 4mil (0.102mm) so the board can be fabricated with solder mask around each pad. We can use 2mil if required.
- We should produce a few spare boards in case we 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 stentil themselves, and they will recommend the best way to panelise the boards to suit their machines.
- We are able to place the rest of the components first with the BGAs being last.

4.2 Current Carrying Capacity

Using the Saturn PCB Toolkit, we can estimate the current carrying capacity of the BGA traces with the following parameters:

Parameter	Value	Unit
Conductor Width	3.5	mil
Conductor Length	30	mm
PCB Thickness	62	mil
Plane Present	Yes	-
Distance to Plane	10	mil
Conductor Current	1.02	A

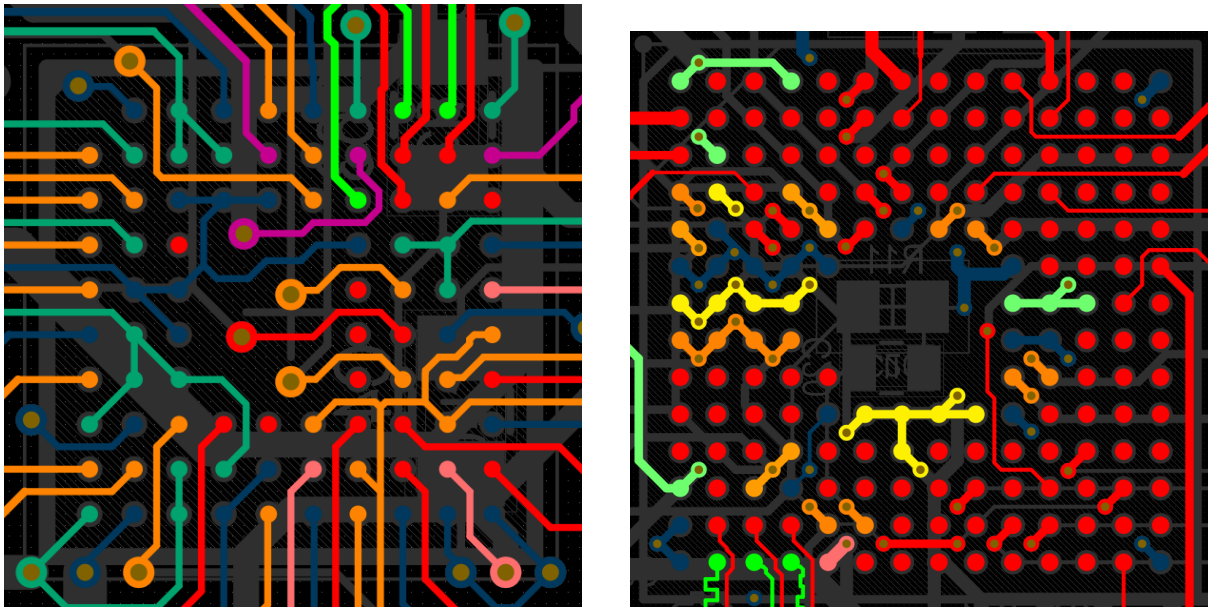
Table 7: Current carrying capacity estimation of BGA traces in C2S Breakout Board.

As per Table 7, the maximum current on a 3.5mil trace is 1A, which is less than the 350mA requested by the MIRA220. The current carrying capacity is not a concern for the STM32N6 since its power traces are 0.2mm, which can carry 1.8A, greater than the maximum 1.1A drawn by the MCU.

4.3 BGA Fanout

We encountered clearance issues with the MIRA220 BGA fanout. Unlike most BGAs which have a ball radius of 0.5-1.0mm with large pitches, the MIRA220 sensor uses a small package profile, one of the smallest commercially available (0.2mm ball diameter, 0.5mm pitch). This package made it difficult to properly route the component. Traditional fanout methods like Dog Bone and Via-in-Pad were not possible as such a small scale, simply because the PCB capabilities didn't permit small enough vias to utilise these methods. 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. Additionally, dog bone was not possible since there was not enough clearance between the via and trace. Although the vias could be placed without errors, it was not possible to fanout on the bottom layer since it would require a trace width of 0.5-0.35-0.17=-0.02mm, which is less than the 3.5mil clearance set by the PCB manufacturer. We then opted for the radial fanout method which admit to the manufacturer's trace width and clearance of 3.5mil (see Figure 13a). Unlike the MIRA220 sensor, the STM32N6 has a more coarse BGA package

which allowed us to use the previously mentioned methods. In this case, we 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.



(a) Fanout of MIRA220 image sensor.

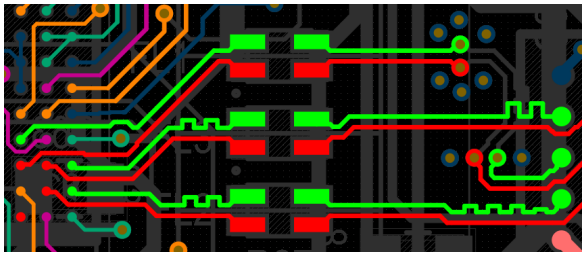
(b) Dog Bone fanout of the STM32N6 MCU.

Figure 13: BGA fanout methods used in the C2S Breakout Board PCB.

4.4 CSI-2 Routing

We used TI's high-speed guidelines for routing the differential signals. From Table A-7 of the guide (pg. 19), we 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, we found the required separation distance between the 3.5mil differential traces to be 3.542mil, or approximately the trace width. We found that there shouldn't be an issue with delay matching at such small distances (<5mm), since $T_{pd} \approx 6.1\text{ps/mm}$ on the microstrip, the trace difference would need to be approx 6.5mm before the lane skew is reached. Regardless, we attempted matching the lengths to achieve a low delay ($\pm 2\text{mil}$).

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, we made sure to place sitching vias around the layer changes.



(a) Differential traces on the PCB.

Name	Signal Length (mil)
IN_MIPI_CLK_N	210.578
IN_MIPI_CLK_P	207.519
IN_MIPI_D0_N	163.697
IN_MIPI_D0_P	164.844
IN_MIPI_D1_N	172.883
IN_MIPI_D1_P	173.204
MIPI_CLK_N	474.231
MIPI_CLK_P	473.437
MIPI_D0_N	234.193
MIPI_D0_P	233.67
MIPI_D1_N	235.476
MIPI_D1_P	236.559

(b) Signal lengths of each differential signal.

Figure 14: MIPI CSI-2 differential pairs on the C2S Breakout Board.

4.5 Debug LEDs

The TX and RX pins are **not representative of UART**. Since there was not enough space to add ”READ” and ”WRITE”, we 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 SD card.

The power LEDs (3V3, 2V5, 1V8, 1V35, 0V8) show whether each line is active or not.

The ”MIRA” 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.

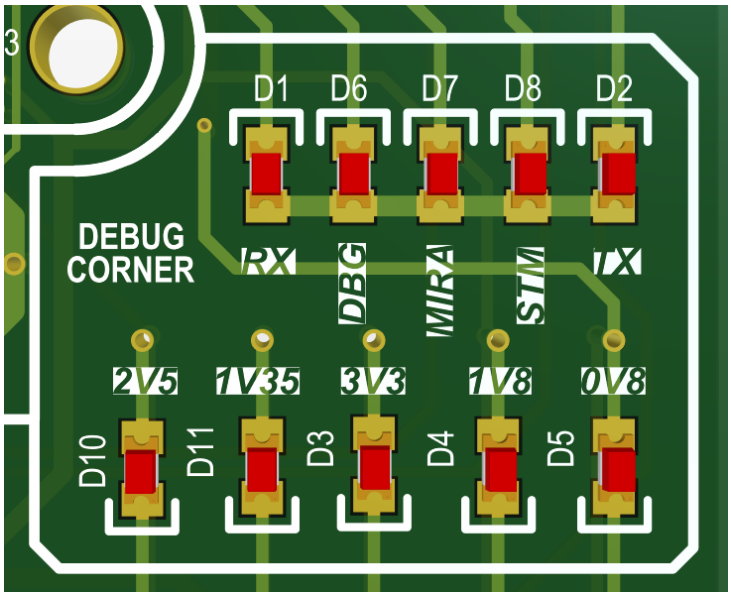
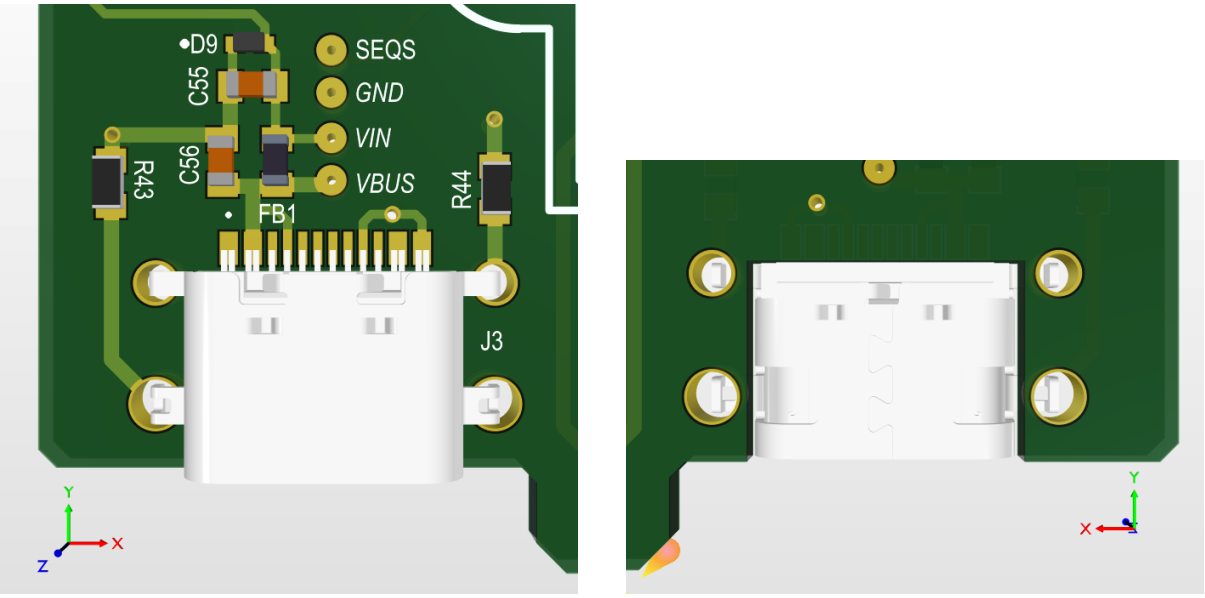


Figure 15: Debug LEDs on C2S Breakout Board.

4.6 USB-C Connector

A cut-out was required to fit the USB-C connector to the PCB. This was made around the mechanical outline and extended outward by 2mm to account for any padding on the cable. After the cutout was made, the tab left of the connector was removed to prevent potential snapping.



(a) Bird eye view.

(b) Bottom side view.

Figure 16: USB-C connector on C2S Breakout Board.

5 PCB Fabrication and Assembly

A 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:

$$C_{eq} = \frac{C_1 \times C_2}{C_1 + C_2} \tag{1}$$

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

$$f = \frac{1}{2\pi\sqrt{L \times C_{eq}}} \tag{2}$$

At frequencies well above fr the filter attenuates; below fr 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:

$$Z \approx j2\pi fL \tag{3}$$

Rearranging:

$$L \approx \frac{Z}{2\pi f} \tag{4}$$