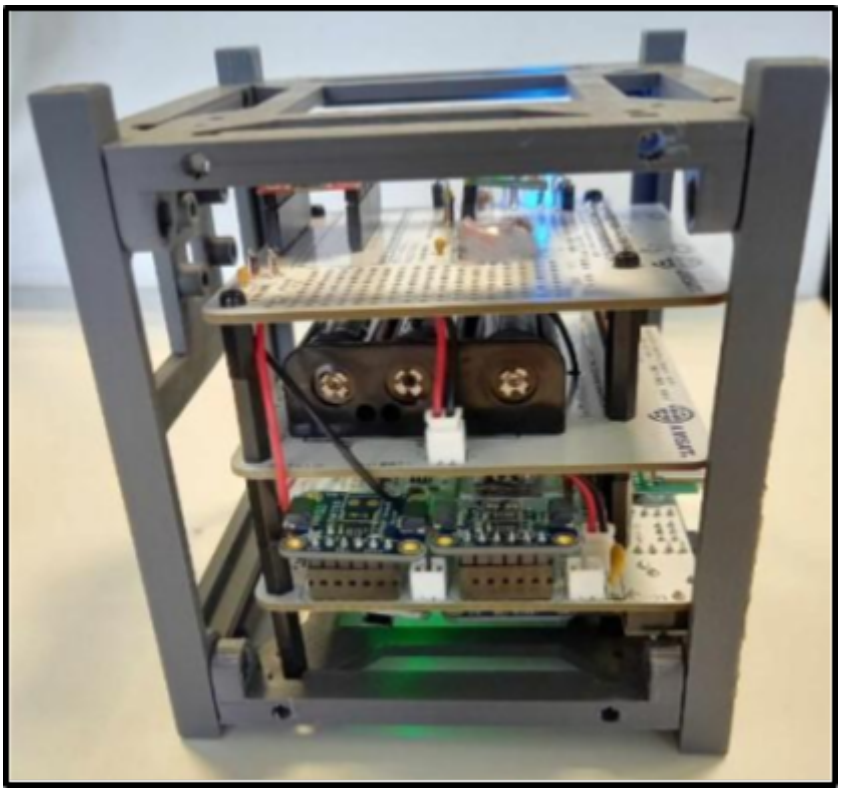
Nimbus Research Centre

Report on CubeSat development Issues and Solutions.

Nimbus Research Center.

Prepared by: JOURDAN Thomas and CHETOUANE M’hamed

Date: 19/08/2025



Abstract:

This technical report outlines the progress, challenges, and solutions developed during the continuation of the CubeSat project. Building on the 2023 report, it addresses issues encountered in hardware and software integration, communications, and system stability. For each challenge, the root cause is analyzed and solutions are described, with the aim of improving CubeSat performance and providing a reference for future development.

Table of contents:

1. Introduction

This document extends the 2023 CubeSat project report, focusing on issues identified in the previous phase and the measures taken to resolve them. It also documents new problems encountered, along with practical insights gained during development and testing. Readers are advised to review the 2023 report before proceeding, as it contains the foundational context for the work presented here.

---

2. Solved issues

We have solved some of the issues raised in the 2023 report:

- **Camera**

Previously, the CubeSatSim failed to detect the Camera Module 3 NoIR. This was due to the original code being written for the Module 2 camera and relying on raspistill, which is incompatible with newer models. The solution involved modifying the code to use the libcamera framework, restoring full camera functionality.

- **SSTV transmissions**

- Testing revealed that not all ground station antennas were capable of receiving SSTV signals. Only the tuner labeled “R820T2 & SDR” on the casing proved reliable. Using this antenna, SSTV reception through QSSTV was successful, enabling images to be received from all CubeSats.

---

3. Current issues

- **Crashing**

During extended testing, the CubeSats occasionally experienced sudden software crashes in the middle of data transmissions. These failures are intermittent, making them challenging to reproduce consistently, and they occur regardless of whether the transmission is SSTV, telemetry, or image data.

One identified contributing factor is GPU memory allocation, particularly when operating the NoIR camera at its highest resolution and quality settings. By default, the Raspberry Pi firmware reserves a limited amount of memory for the GPU, which is insufficient for high-quality image capture and processing. When memory demands exceed this allocation, the camera subsystem fails, leading to an abrupt program termination and, in some cases, a complete system hang.

---

4. Tips and lessons learned

Over the course of development, integration, and testing, a number of practical techniques and considerations emerged that significantly improved workflow efficiency and reduced downtime. These lessons stem from repeated trial-and-error, debugging sessions, and unexpected issues encountered while working with the CubeSat and ground station systems.

- **Remote desktop and SSH**

You should use realVNC to connect to the ground station. It makes it a lot easier to exchange files between your computer and the ground station. Another possibility is to open a github via chromium.

To connect to wifi and activate SSH on a cubeSat do “sudo raspi-config”.

- **Raspberry pi imaging**

It is better to use balena etcher instead of Pi imager for custom images. Sometimes Pi imager can force use to input a username a password, this would cause the program on the cubesats and the ground station to not function properly.

- **Check READMEs**

If you are stuck or one of the programs made by another intern doesn’t work, check the readmes, you might have forgotten to do one step. Sometimes the issue is that the packages are not installed or you might need to reboot the Pi

- **Check running software**

Sometimes one of the programs might not work. Often it is the case that it is because another software is running in the background and already using one of the packages or one of the tuners. You can use “killall <process_name>” functions to stop processes.
Freeing resources before starting a new task significantly improves reliability.

To address this, identify and terminate conflicting processes using:
ps aux | grep <process_name>
kill <PID>
or to terminate all instances:
killall <process_name>

- **Don’t return to factory settings.**

While reverting a CubeSat to factory settings might seem like a quick way to restore stability, it comes with a major downside: all additional software, configurations, and optimizations will be lost. This includes:

- Visualization software
- Custom telemetry tools
- Project-specific configuration files

Reinstalling these components is often time-consuming and may require re-implementing changes that are not fully documented. Factory resets should only be used as a **last resort**, after other recovery strategies have been exhausted.

---

5. Projects in progress.

- Hardware simulation project.

Due to issues on the 3D printer we couldn’t make it work in time to print a simulation environment.

- Distributed Computing project.

It could be interesting to work on new function to improve the Distributed computing project.

I have started to work on the project.

---

6. Potential future projects.

- Create a MQTT interface for communication between CubeSats.

Could be interesting and a lot more optimized for communication

- Create a telecommunication module based on LoRaWan

LoRaWan would be more optimised for telecommunication compared to wifi.

- Create a visualisation software to know the power received by each solar panel. Thus knowing how the cubesat is positioned in space

- Create a visualisation software to receive SSTV from different CubeSats.

This report serves as a comprehensive documentation of the challenges faced and the innovative solutions applied during the CubeSat's construction and operational phases.

Nimbus Center
Cork Institute of Technology Campus, Bishopstown, Cork