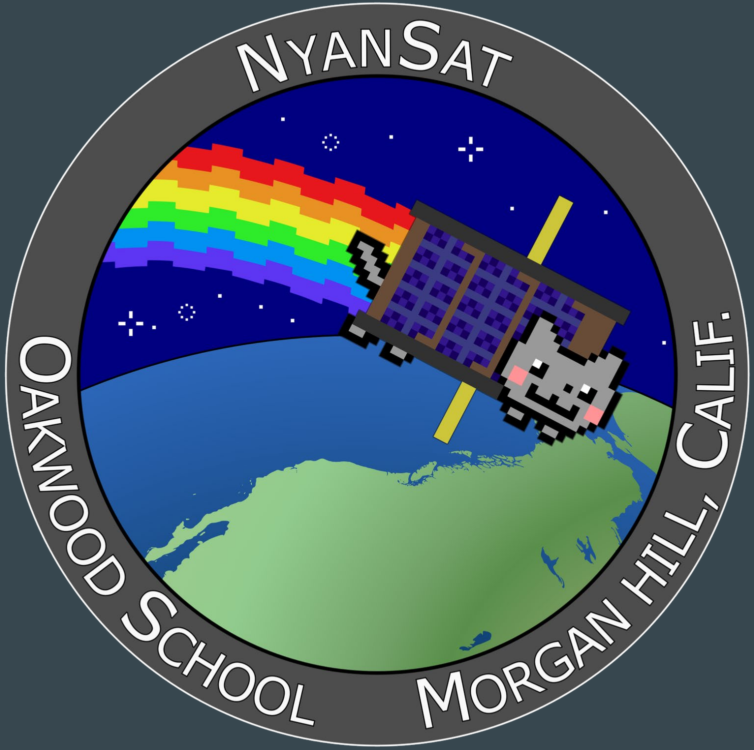
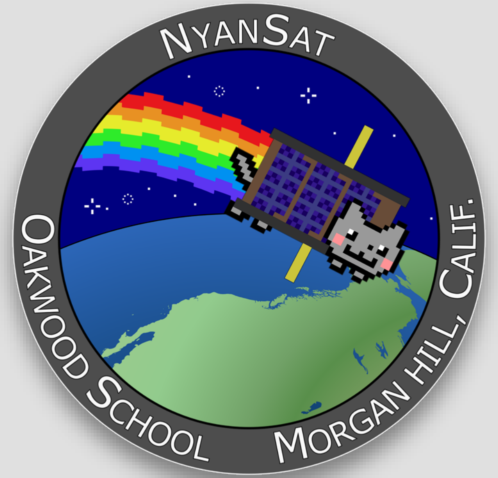
Oakwood High School's NyanSat Mission



				
Lauren Sorci ’25 Project Lead	Porter Banks ’27 Avionics Lead	Dillon Hall ’26 Structural Lead	Brendan Lyle ’27 Analysis Lead	Kayden Wang ’26 Software Lead



- Oakwood High School; Morgan Hill, CA
- 2U CubeSat; 7 Payloads
- Inspire students and provide learning opportunities
- Contribute to the development of new and innovative space technologies

CSLI Call 15 Selections

Payloads

Technology Demonstration

- Acoustic Spacecraft Mapping and Sounding
- Orbit Determination of Other SmallSats
- Cryptographic Ledgers in Space
- LCVR Waveform Generator

Educational Outreach

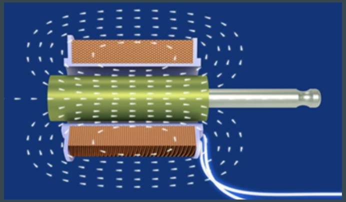
- Drawing “Shooting Star” Dispenser
- Camera

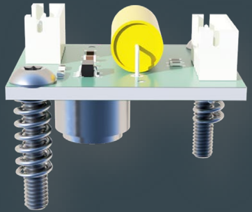
Science / Outreach

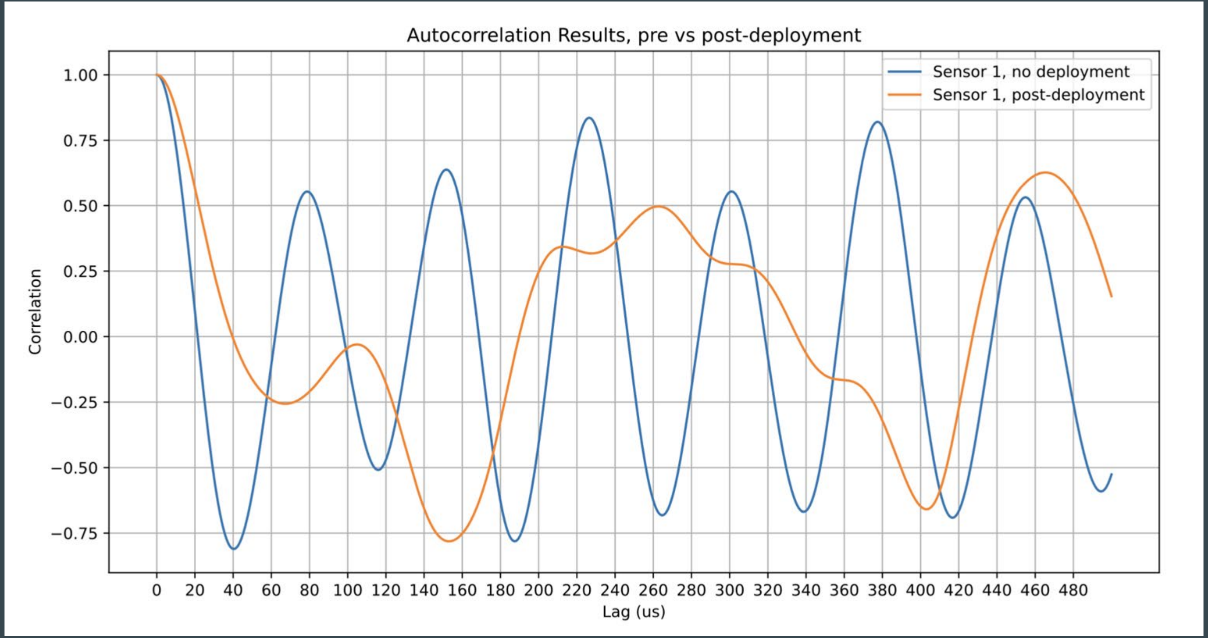
Material Science Experiment

Acoustic Spacecraft Mapping and Sounding

- Observes vibrations sent through spacecraft structure
  - Before, during, after deployment
- Used as alternative to switches and other sensors to determine state of satellite







Development of Acoustic Systems for Structural Monitoring in CubeSats

Background & Problem

Conventional spacecraft use mechanical switches to confirm successful deployment of their components. These switches are cumbersome, unreliable, and prone to failure, which can compromise the mission's success by breaking and giving inaccurate readings.

Solution & Approach

NyanSat addresses this issue by introducing an acoustic monitoring system that uses sound waves to detect mechanical stresses and verify deployment status. The technology will ensure that deployment has been successful without the risk of harming the Cubesat or other nearby spacecraft. This approach reduces moving parts, enhances reliability, and makes the system lighter and more efficient for CubeSats and other future aerospace applications.

Implementation

NyanSat's acoustic verification system will be implemented by mounting solenoids and ultrasonic transducers in strategic points in our CubeSat. Solenoids generate mechanical impulses and transducers capture and analyze the acoustic reflections to find the structural changes. Ground testing will be conducted to find a baseline acoustic signature that we can use to compare to the signature received while in orbit. A board on the satellite will use this information to indicate the status of our deployable mechanisms.

Development

The NyanSat acoustic verification system will be developed through an iterative design process. We will use both CAD models and experimental testing for development. Our first prototype has been tested in a simple environment to analyze the wave propagation through beams. Our next tests will take place with our CubeSat structure in a vacuum. These tests will help refine where we place the solenoid and how sensitive our transducers are. The data we will get can be used to make hardware and software changes so we can be more optimized. We are iterating our designs making sure they can meet all the requirements as we approach our CDR.

Ultrasonic transducers and sensor

Next Steps

Moving forward, we will focus on refining our prototype through additional ground testing and software optimization, making sure it is sustainable for space and any possible encounters. We will finalize the baseline acoustic signatures of successful deployment, ensuring accurate detection in space. Integration and testing will be conducted under simulated space conditions, including vibration and thermal vacuum tests, to validate system performance.

About Us

We are a team of fifteen high school students attending Oakwood School in Morgan Hill, California. This is the team's fourth year working on the project. We were chosen by NASA to take part in their launch initiative to send a CubeSat into Earth's low orbit. We are highly appreciative for the offer and are committed to making the most of this opportunity. Our team has dedicated countless hours to designing, testing, and refining our CubeSat to meet NASA's standards.

Switches are bulky and can be **unreliable** in space, causing **inaccurate** deployment readings and requiring redundancy.

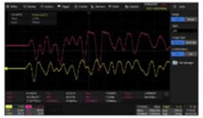
Switches provide a simple method for detecting changes in position or state, making them an important option. However, adding redundancy and hitting reliability targets is often difficult.

NyanSat introduces a novel **sound-based** system to verify spacecraft **deployment** and **structural status.**

NyanSat explores acoustic technology with potential applications in larger spacecraft, aerospace structural analysis, and beyond.

NyanSat's **ultrasonic** tech ensures **accuracy**, reducing mechanical parts through use of a small number of **solenoids and transducers.**

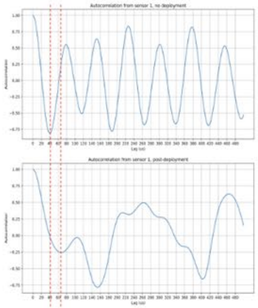
Katerina Starodub, Adhrit Sinha, Efa Anaïs Cannieux, Dhruvil Patel, Michael Lyle



Acoustic waves from deployed configuration

Test for deployed configuration of satellite

Test for closed configuration of satellite



View of "Mic" Board

Tools

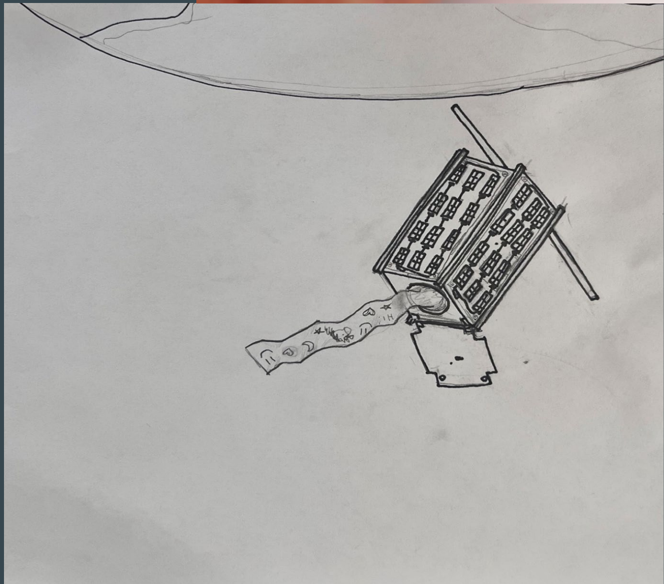
- **Python, Pandas, NumPy, SciPy, Jupyter Notebooks, Plotly -** coding / debugging
- **Discord & Google Colab** - collaboration
- **Oscilloscope SDS2104X Plus -** reading waveforms
- **Wrench, Fishing Line, Ultrasonic Transducers** - model used for testing and producing waves

Material Science Experiment

- Observes properties of various semiconductors in space
- Kits to disadvantaged schools around the country
  - Students to perform control experiments and compare results

Drawing “Shooting Star” Dispenser

- Drawings from local 5th graders
- Active topic of design- orbital debris concerns
- Camera



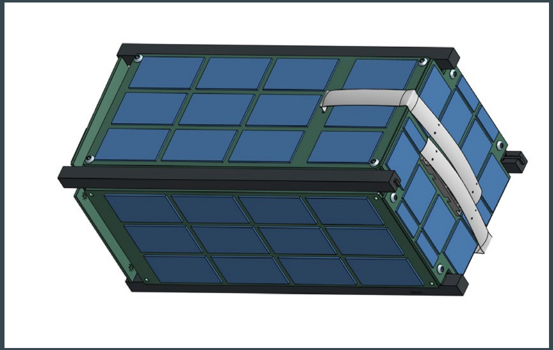
CONOPS: Early Mission Phase

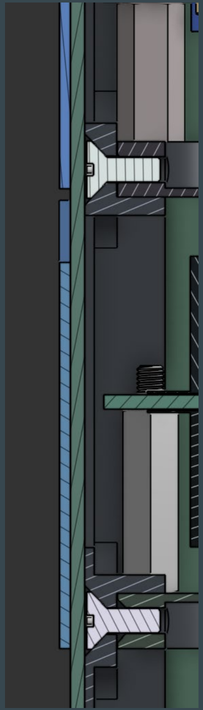


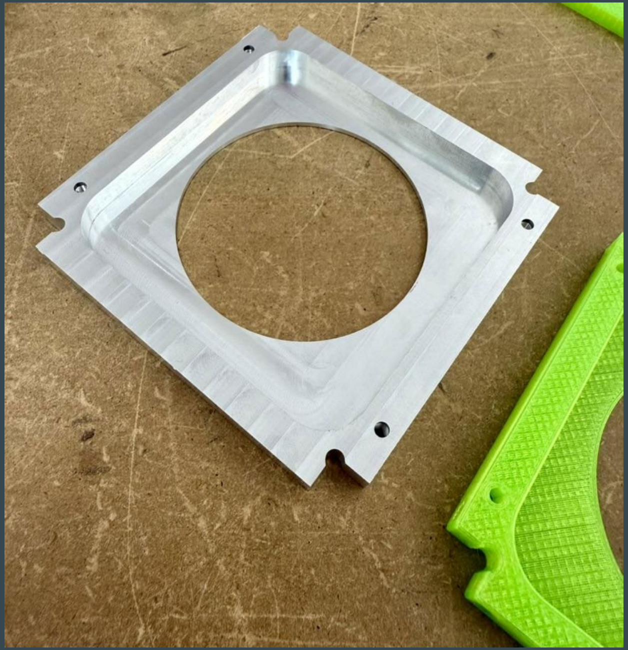


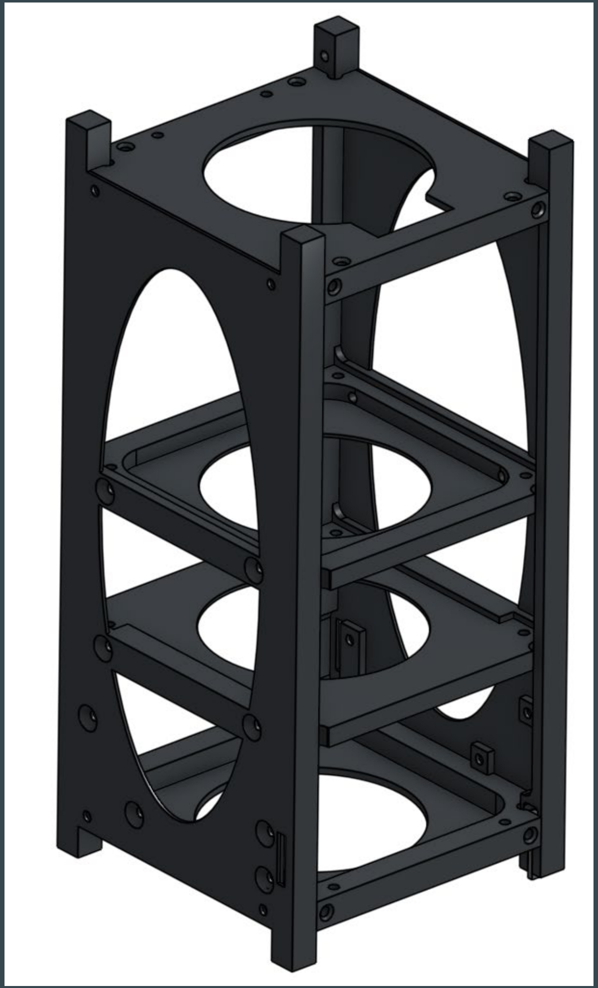
1. Released into space & waits 45 mins
2. Deploys antennas, starts telemetering
3. Attempts detumble
4. Awaits ground commands
5. Before further deployment: obtain control acoustics observations

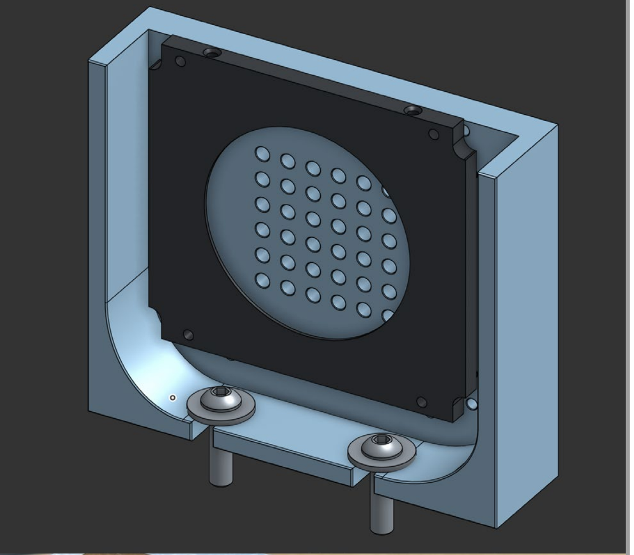
Structure

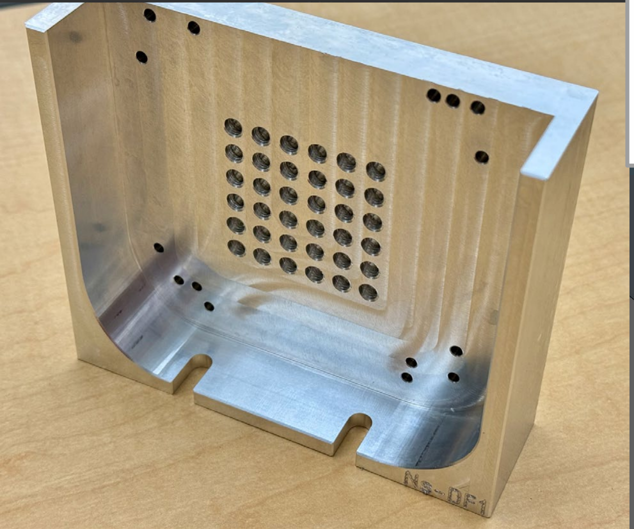


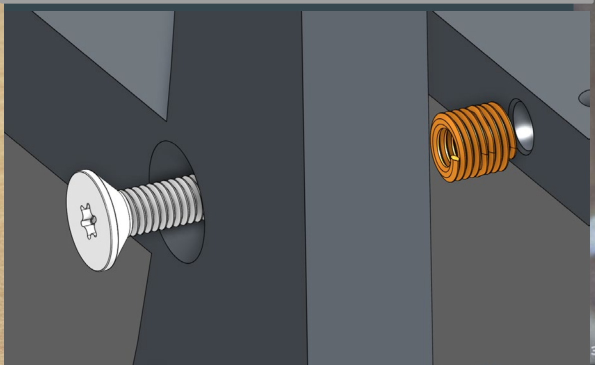


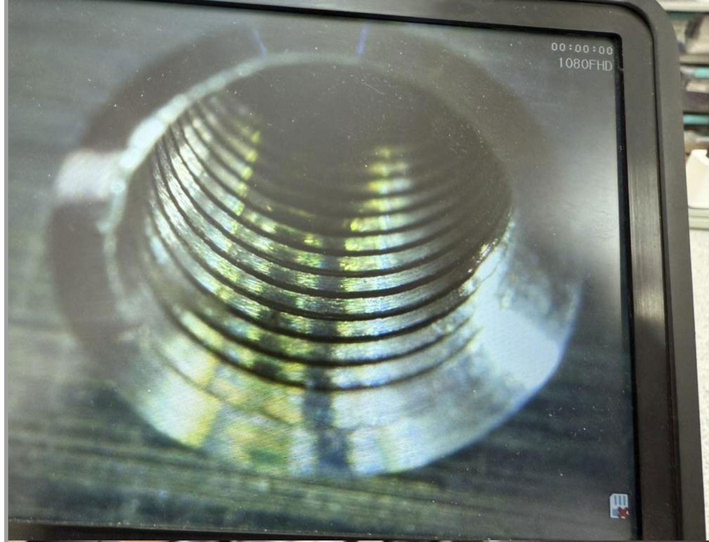







Ns-DF1

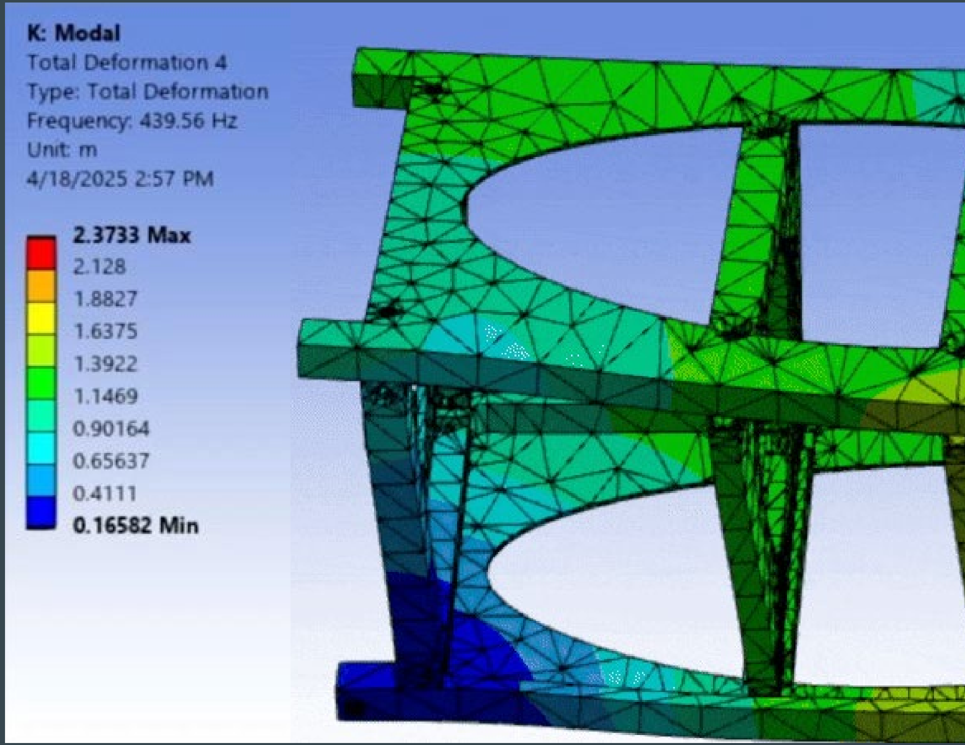


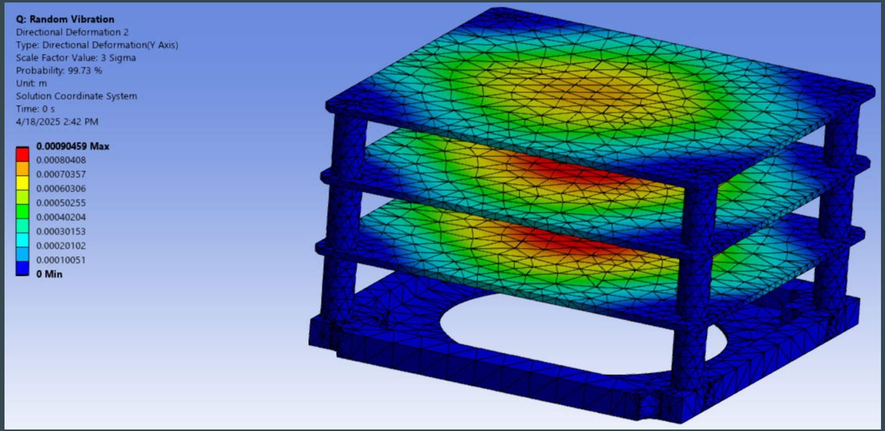

00:00:00
1080FHD


00:00:00
1080FHD
3:28

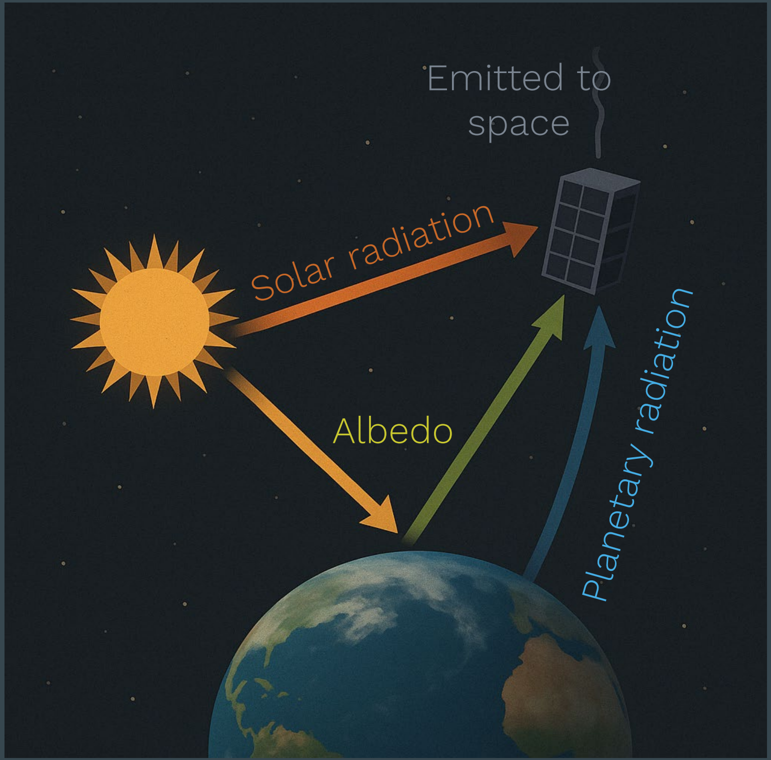
Structure Analyses

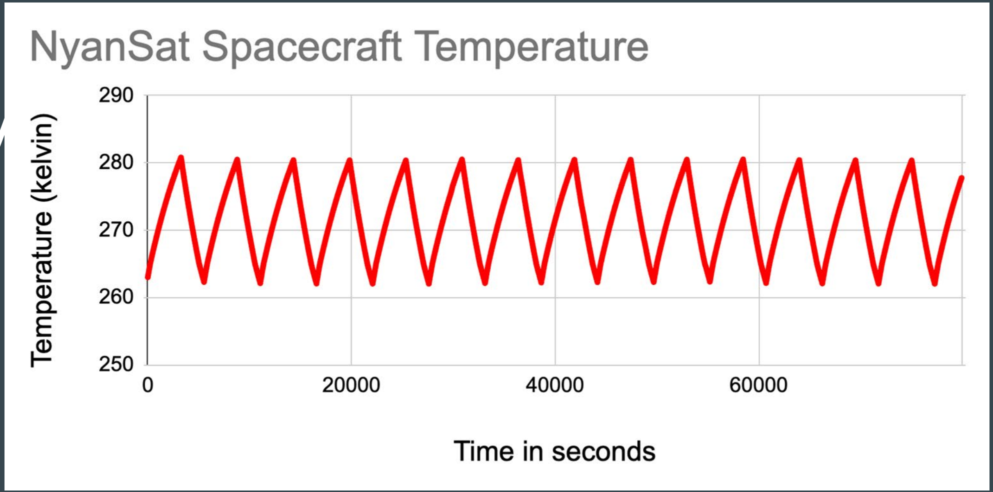
- Static Structural
- Modal (frame and board stack)
- Random Vibration
- Upcoming: Internal Heat Transfer

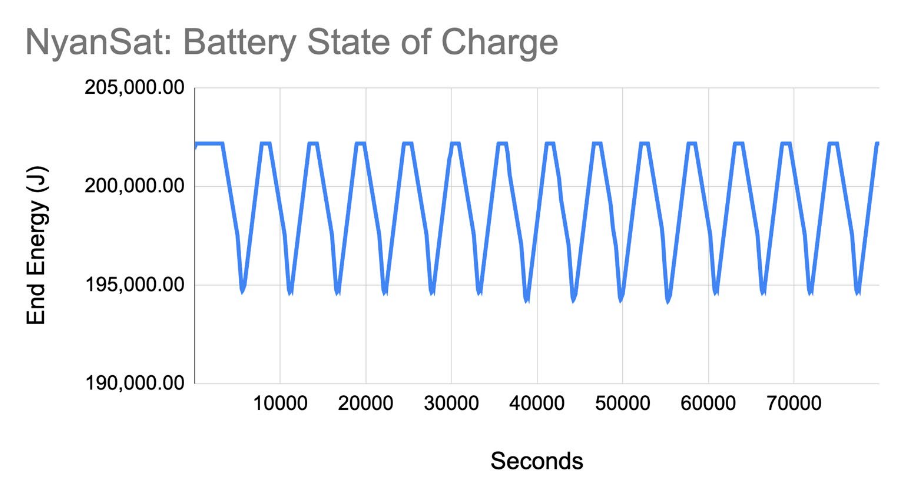




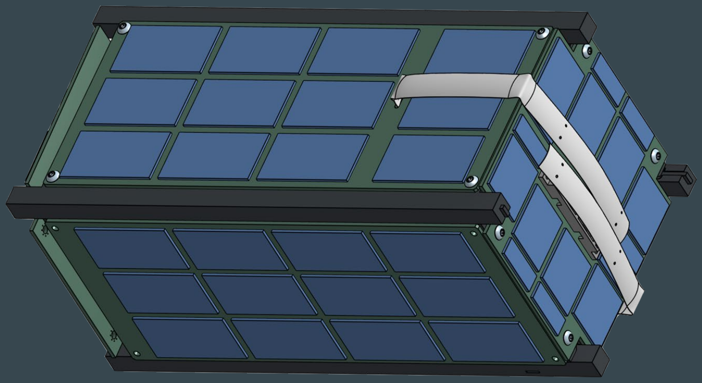
Thermal / Electrical Overv

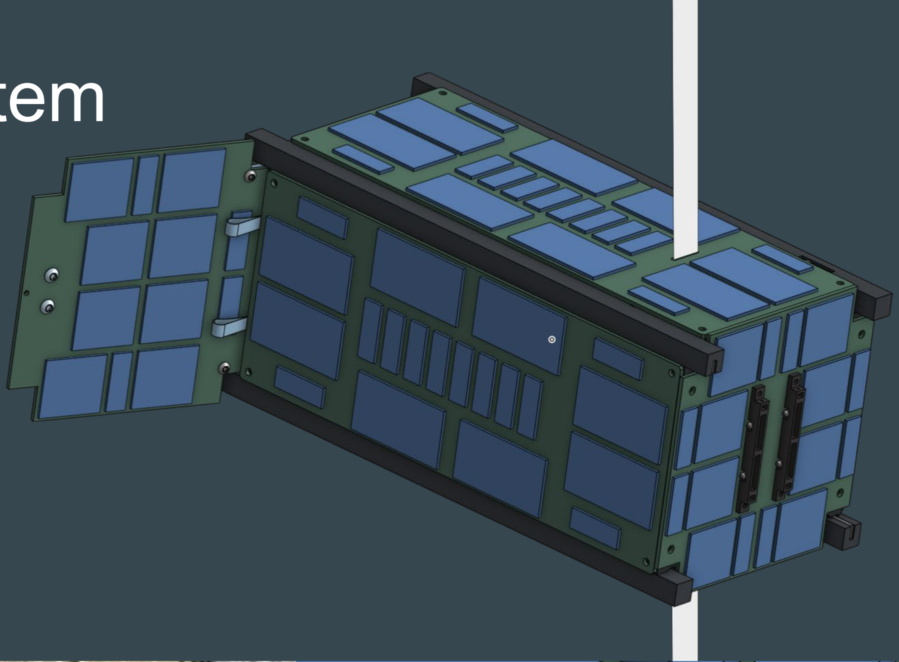


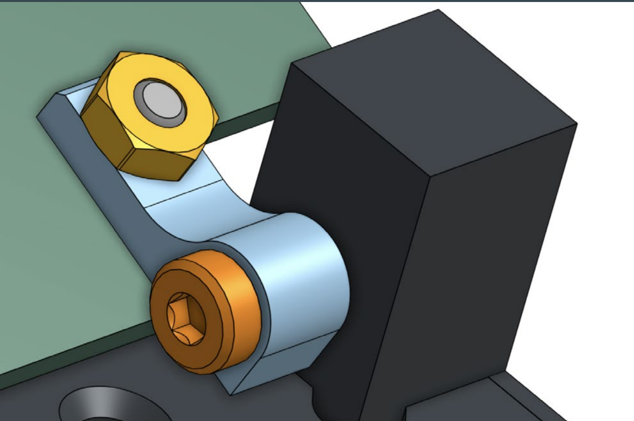


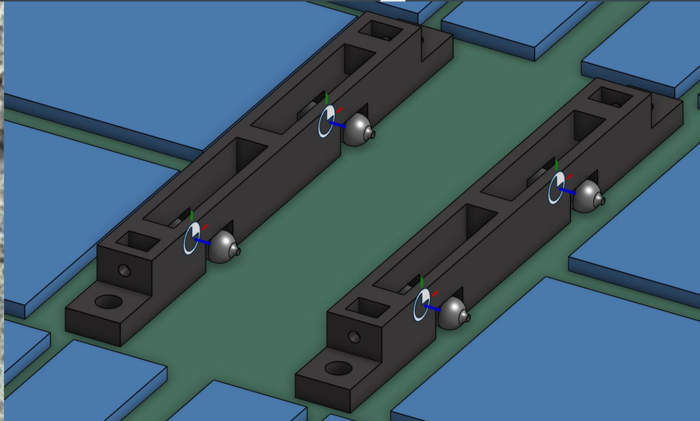


Burn Wire Deployment System









Avionics

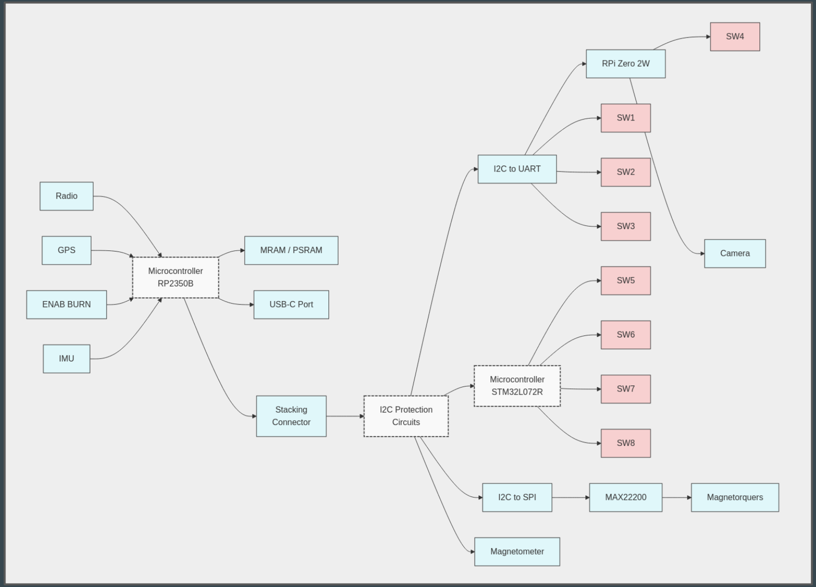
Board Stack & Structure

On-Board Computer

Combined Payload Board (+ Camera & Acoustic Sounding boards)

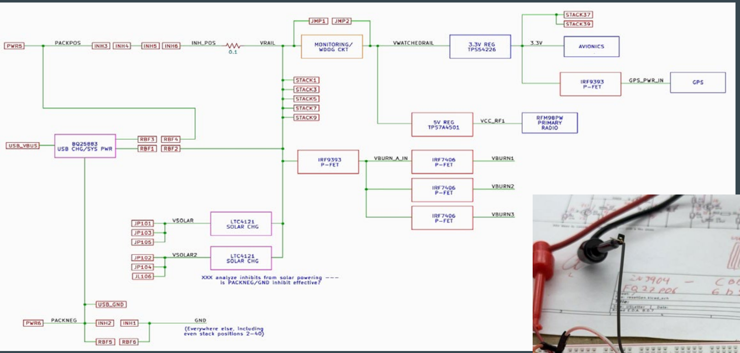
Board Stack

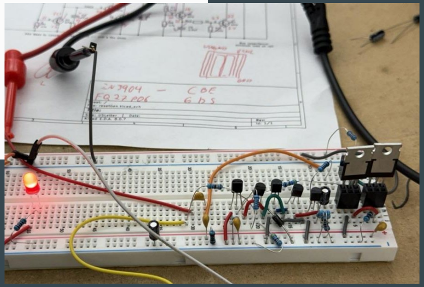
- PC-104 Form Factor
- Based on PyCubed by Holliday

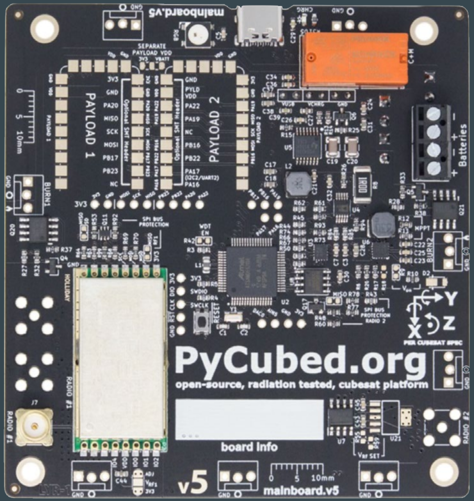


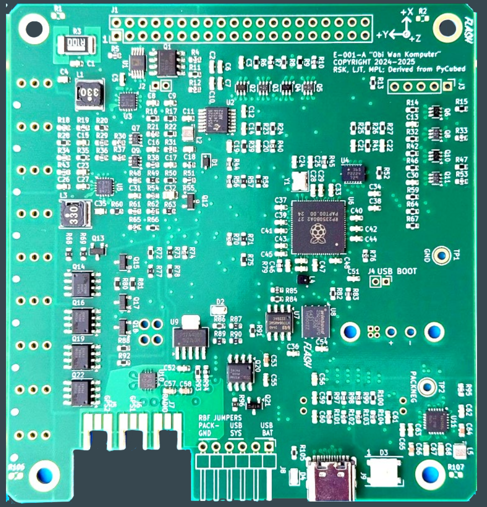
On Board Computer

- Recent discovery of ATSAMD51 microcontroller’s incompatibility with space environment (and positive radiation testing of RP2350)
- Stacking, connectors, access port. Most components on top; radio/GPS/connectors on bottom
- Novel discrete analog watchdog/supervisory circuit cycles power upon request, power anomaly, or failure of microcontroller to toggle a signal at correct rate



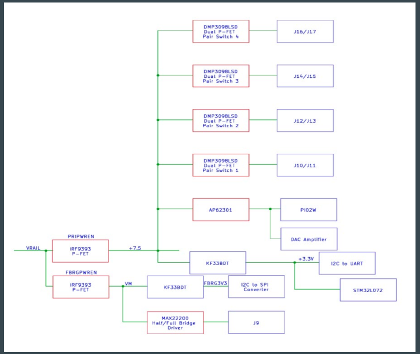


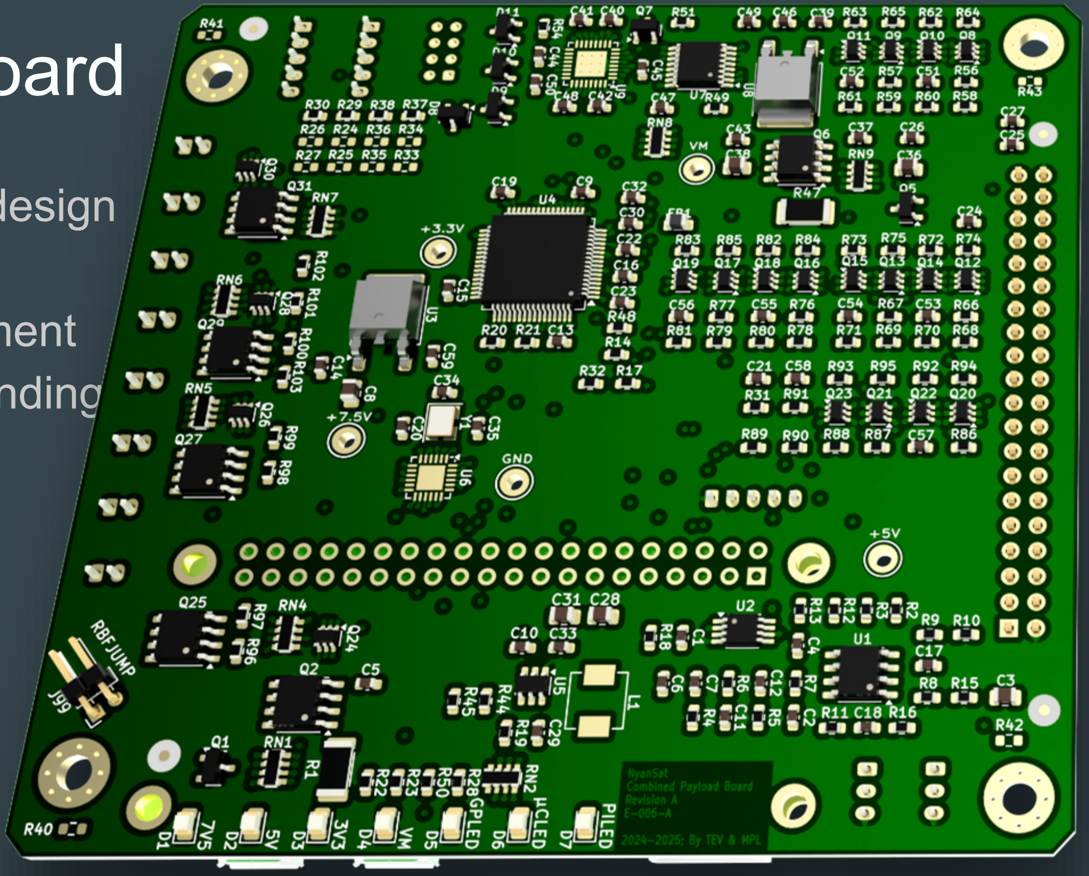




Combined Payload Board

- Original schematic and PCB design
- Subsystems
  - Material Science Experiment
  - Acoustic Spacecraft Sounding
  - Camera
  - Deployables
  - ADCS





Versatile Flight Computer and Payload Board for CubeSats

Derived from PyCubed

Open-Source CubeSat Technology: Developing the Obi Wan Computer and Combined Payload Board

Background

- Oakwood's NyanSat was selected by the NASA CubeSat Launch Initiative (CSLI Call 15)[1]; NyanSat is slated for launch in 2026.
- Based on PyCubed[2], we designed these boards to allow for easier integration into larger and more complex satellites.
- We also targeted easier manufacturing and sourcing; mostly single-sided with readily available components.
- The Combined Payload Board (CPB) is designed to act as a versatile payload interface, suitable for many mission designs.

OBC Design Process

Derived from PyCubed
- Developed stacking connector protocol for board interoperability
- Transitioned memory component from to large MRAM & PSRAM
- Converted from 4-layer to 6-layer PCB
  - Removed routing from top/bottom component layers
  - 2 designated signal layers
  - Alternating with designated power & ground planes (two ground regions, six power regions).
- Added more solar charging and simplified burn-wire circuitry
- Added high-reliability connectors for power, deployment switches
- Moved towards using bipolar and P-FET circuit for power gating to maximize radiation tolerance

Change of microcontroller
- Removed the ATSAMD51 microcontroller due to vulnerability to upsets from solar radiation[3]; converted to the RP2350B.

Watch Dog and Supervisor Circuit
- Novel circuit that cycles power upon request, power anomaly, or failure of microcontroller to toggle a signal at correct rate.

CPB Design
- Initial thoughts about separate cards for each payload
  - Decided we wanted to consolidate most payload functions onto a single versatile board
  - Many IO and power switches that can be used for various purposes
- Off by default
  - 5 gated power domains
  - Power gated by pins on stacking connector; subdomains gated by I2C peripherals
- Enabled and commanded by OBC
  - Designed to remain safe and maintain partial functionality even if program memories are corrupted.

Technical Specifications

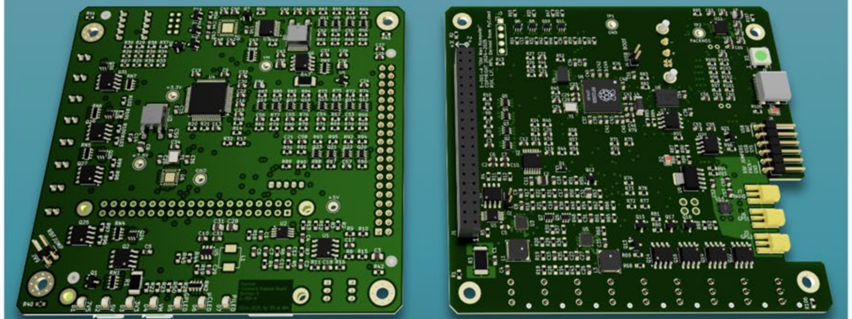
Both Boards:
- 6 layer PCB board with 2 power layers and 2 signal layers
- No traces on top or bottom with vias in pad
- Stacking Connector (in PC/104 location, with our own pinout)

OBC	CPB

• Radio 440 MHz; LoRA or GFSK GPS L1 1.5GHz (with combiner)
• MRAM (16mbit) and PSRAM this (64mbit)
• Designed for 2S Lithium-Ion; baseline ~20V OC solar
• 6 solar charger ports (2 solar domains with MPPT, 6W charging total)
• Copious sensing: IMU, voltages/currents, temperatures, etc.
• 3 high-side switched burn wires (dual-redundant series switched)
• Novel power supervisory circuit to mitigate upsets and latchups
• Runs CircuitPython
• USB, charging, RBF, RF connectors, inhibit cabling, and status LED accessible via access port
• 130 mW base power consumption (may improve with software changes)
	• Uses STM32L072 microcontroller with 128KB flash
	• Many functions independent of microcontroller; presents 4 distinct I2C targets to OBC with protection circuits[4]
	• 8 high side switches (PWM-capable) for high current payloads, deployables, heaters
	• 4 full bridges/8 half bridges (for ADCS or Solenoids) in voltage or current control mode.
	  ○ All switches are dual-redundant (series) switched
	• 2 buffered DACs with feedback
	• 4 ADCs; versatile filtering and coupling modes for all analog.
	• Protected I2C sub-interface
	• Raspberry Pi Zero 2 W for imaging and larger tasks.
	  ○ Using WiFi during development process

Applications & Next Steps

- As two primary circuit boards for the NyanSat mission, they will manage everything from communication to power and control systems for the CubeSat.
- All hardware is open sourced; flight software will be open-sourced, providing a foundation for the broader CubeSat community to build upon.
- OBC Rev AF bringup in progress and looking good after successful OBC Rev 1
- CPB Rev A bringup successful with no significant errata.
- We intend to continue building open avionics and increasing our capabilities.



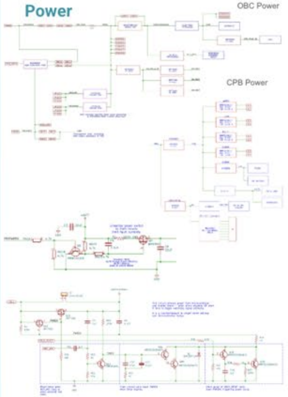
CPB OBC

Open Source

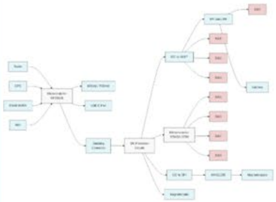
Easily fabbed and assembled by commodity manufacturing

Luke Titi, Ethan Vo, Ruchir Kavulli,
Ryan Beaulieu, Robin Klingauf, Michael Lyle

Power



Hardware



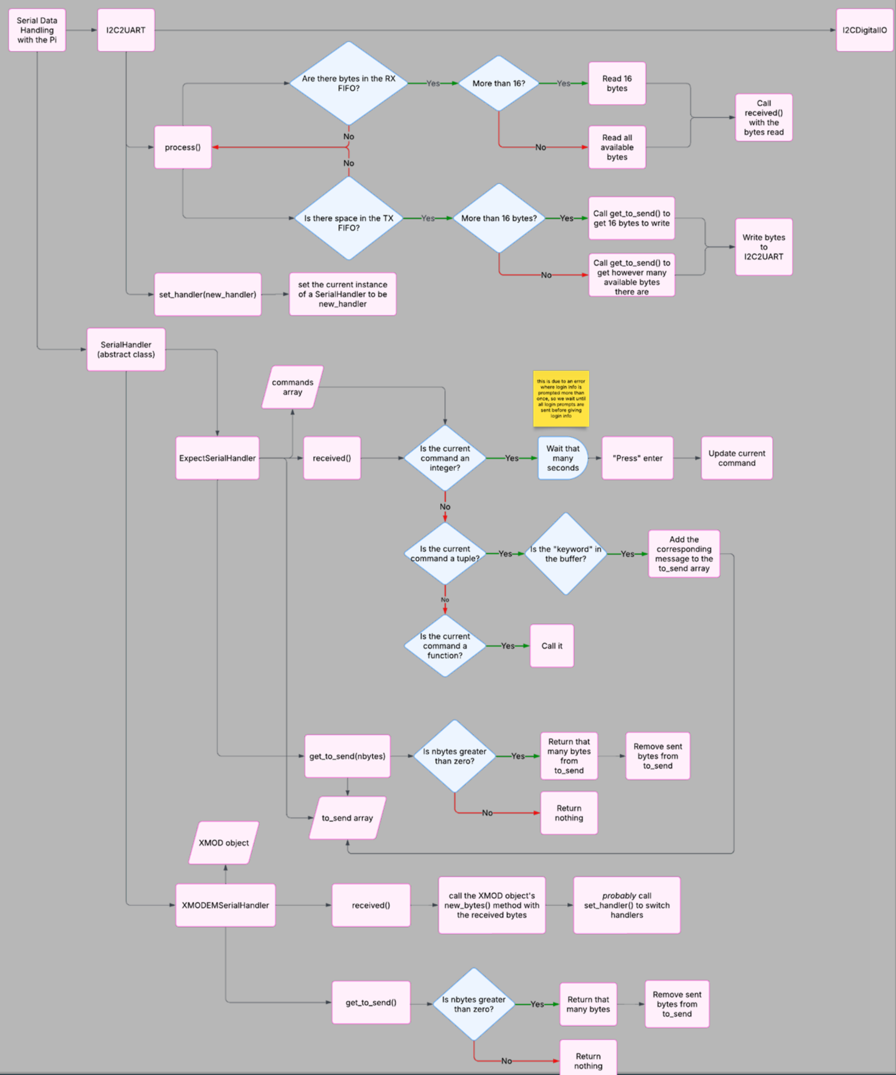
Software

Citations

1. NASA Selects New Round of Candidates for CubeSat Missions. National Aeronautics and Space Administration, Washington D.C., USA https://www.nasa.gov/centers-and-facilities/kennedy/nasa-selects-new-round-of-candidates-for-cubesat-missions-to-station/
2. Maximilian Holliday, et al. PyCubed: An Open-Source, Radiation-Tested SmallSat Framework Programmable Entirely in Python. https://pycubed.org/
3. PyCubed: NOTICE - AUGUST 1 2024 https://github.com/pycubed/forums/discussions/87
4. Maximillian Holliday , Gabriel Buckmaster, Zachary Manchester, et al. On-Orbit Implementation of Discrete Isolation Schemes for Improved Reliability of Serial Communication Buses. TechRxiv. August 13, 2021. https://www.techrxiv.org/doi/full/10.36227/techrxiv.15166620.v1

Board Bringup

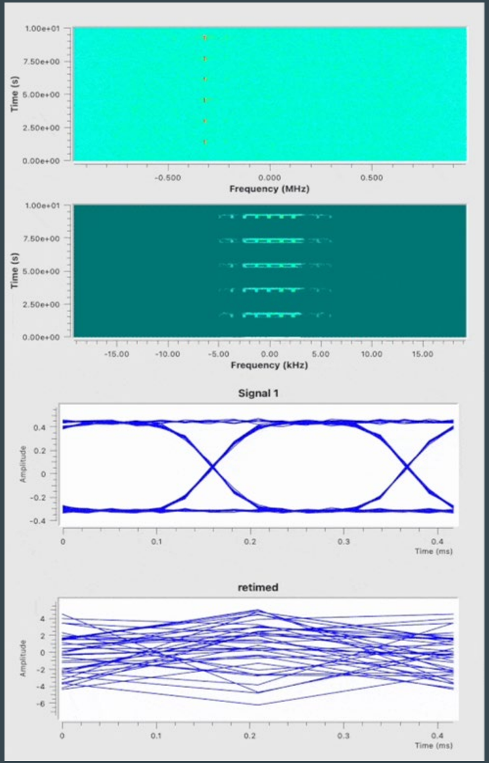
- Testing components of our custom boards
- Software Modules developed from scratch
- Raspberry Pi 02W to handle imaging



Radio/Telemetry

- GFSK-4800, 440MHz
- File transfers scheduled when over SatNOGS scheduled observations
- Sensor data is stored to the Telemetry Data Store (TDS)
- Telemetry frames
  - composed of critical parameters
  - followed by least recently telemetered parameters

NyanSat camera's first photo



Power management

- Approximately twenty power domains in satellite with complex dependencies
- Power Supervisor subsystem keeps critical systems active when power is low
- Understands dependencies and properly sequences for user commands or scheduled activities
- Monitors and reports on battery and solar charging status

powersupervisor.py

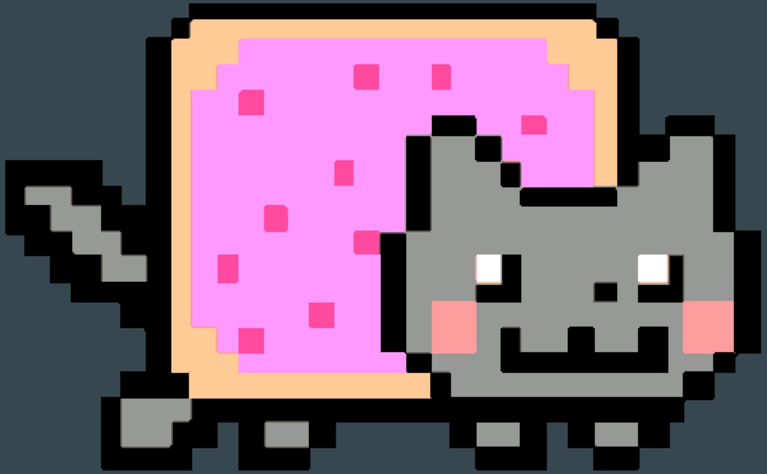
```python
class PowerSupervisor():
    def process(self):
        print("beginning to process power")
        # Manage power for processes, turning off everything that should
        # be off, and turning on at most one thing per cycle.

        turned_on = False # have we turned on anything yet this cycle?
        turned_off = True

        #Iterate all self.services.
        for dev_id in self.devices:
            this_dev = self.devices[dev_id]

            dev_timeout = self.timeouts.get(dev_id, 0)

            if dev_timeout < self.get_time():
                # Timeout is in the past, device should not be on.
                if this_dev.enabled == True and turned_off == True:
                    # It is on, so turn it off.  We may not want this ultimately,
                    # instead preferring to explicitly turn it off each cycle.
                    turned_off = self._turn_off(dev_id)
            else:
                if this_dev.enabled == False and turned_on == False:
                    # It is not on yet, and we haven't turned on anything,
                    # so try to turn it on
                    turned_on = self._turn_on(dev_id)

    def checklist(self):
        reverselist = self.orderlist[::-1]
        print("this is the order we turned it on:", self.orderlist)
        print("this is the order we should turn it off:", reverselist)

psup = PowerSupervisor(system_timeout=0.2 * 1000000000)
n = 0

psup.add_device(PoweredDevice("a"))
psup.add_device(PoweredDevice("b", ["a", "c", "d", "e", "f"]))
psup.add_device(PoweredDevice("c", ["a", "d", "e", "f"]))
psup.add_device(PoweredDevice("d", ["a", "e", "f"]))
psup.add_device(PoweredDevice("e", ["a", "f"]))
psup.add_device(PoweredDevice("f", ["a"]))

n = 0  # Ensure n is defined before use
```

Power Supervisor Module

Thank you!

Questions?

nyansat.org



@OakwoodEngineering